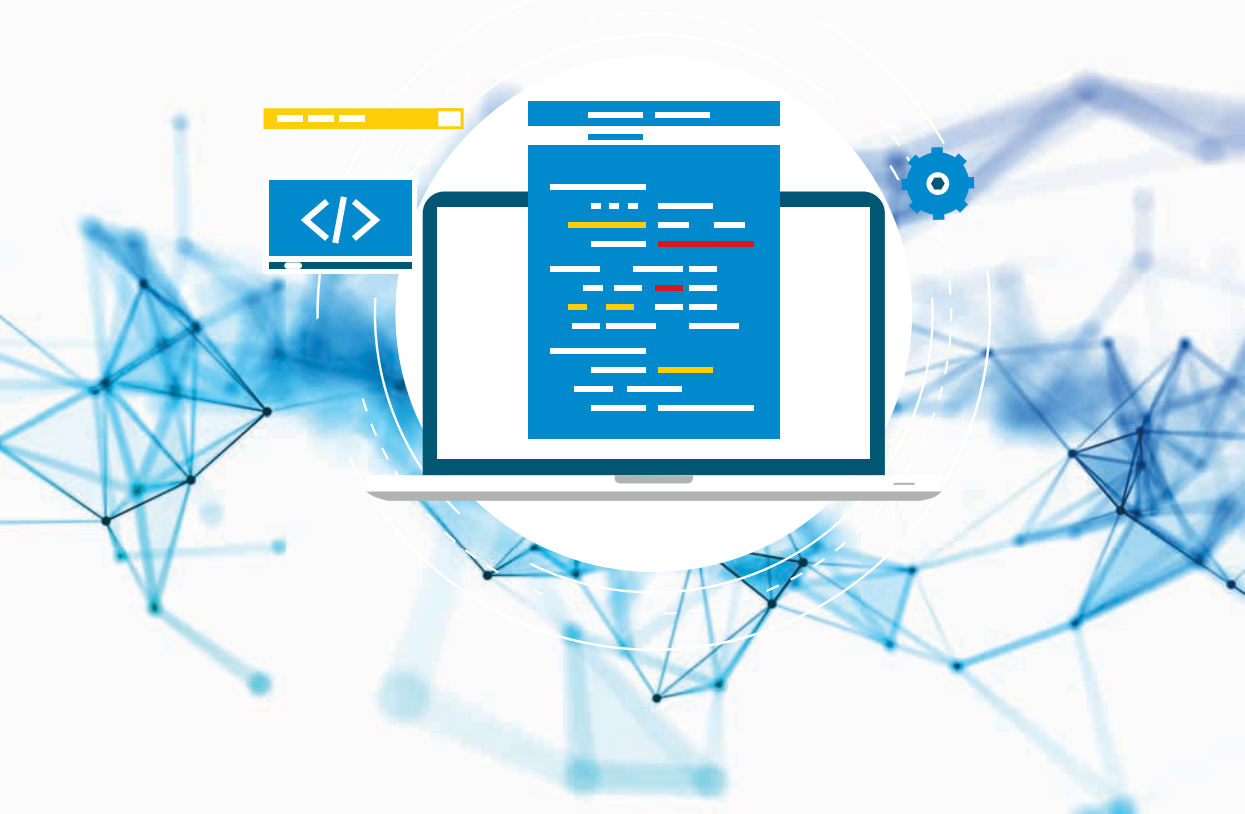
Univerzitet Singidunum

VEB INFORMACIONI SISTEMI

Miloš Dobrojević



Beograd, 2024.

UNIVERZITET SINGIDUNUM

Miloš Dobrojević

VEB INFORMACIONI SISTEMI

Prvo izdanje

Beograd, 2024.

VEB INFORMACIONI SISTEMI

Autor:
dr Miloš Dobrojević

Recenzenti:
dr. Mladen Veinović
dr. Valentino Vranić
dr. Petar Spalević

Izdavač:
UNIVERZITET SINGIDUNUM
Beograd, Danijelova 32
www.singidunum.ac.rs

Za izdavača:
dr Milovan Stanišić

Tehnička obrada:
dr Miloš Dobrojević

Dizajn korica:
Aleksandar Mihajlović, MA

Godina izdanja:
2024.

Tiraž:
600 primeraka

Štampa:
Caligraph, Beograd

ISBN: 978-86-7912-827-0

Copyright:
©2024. Univerzitet Singidunum
Izdavač zadržava sva prava.

Reprodukcija pojedinih delova ili celine ove publikacije nije dozvoljena.

Sadržaj

PREDGOVOR 9

1. OSNOVNI POJMOVI 15

1.1 Sistem 15
1.1.1 Podsistem 16
1.1.2 Komponenta sistema 20
1.2 Podaci 21
1.2.1 Struktura podataka 22
1.2.2 Informacija 23
1.2.3 Analiza 24
1.3 Ljudski faktor 25
1.4 Znanje 27
1.5 Odluka 28
1.6 Informisane poslovne odluke 29

2. INTERNET I VEB APLIKACIJE 33

2.1 Veb serveri 34
2.2 OSI Model, transportni protokoli i HTTP protokol 35
2.3 Veb aplikacije 40
2.4 Multi-platformska kompatibilnost 41
2.5 Troškovi razvoja i održavanja 41
2.6 Arhitektura veb aplikacije 42
2.6.1 MVC arhitektura 44
2.6.2 Modularnost 45
2.6.3 Programiranje na strani servera 46
2.6.4 Programiranje na klijentskoj strani 48
2.6.5 Integracija sa bazom podataka 49
2.7 Kukiji 52
2.8 Komunikacija sa korisnikom 53
2.9 Rutiranje 54
2.10 Upravljanje HTTP zahtevima i odgovorima 56
2.11 Serijalizacija podataka 58
2.12 Rad sa fajlovima 59
2.12.1 Preuzimanje (*upload*) fajlova sa klijentske strane 60

2.12.2 Manipulacija fajlovima na serverskoj strani 60
2.12.3 Strimovanje fajlova 63
2.13 Asihorni pozivi 63
2.14 Ažuriranje u realnom vremenu 65

3. KORISNIČKI INTERFEJS 71

3.1 Osnovni principi 71
3.2 Dizajn 74
3.3 Gradivni elementi veb stranice 75
3.3.1 Navigacija 77
3.3.2 Mrvice hleba 77
3.3.3 Paginacija 78
3.3.4 Slajderi 79
3.3.5 Tagovi 80
3.4 Informacije, obaveštenja i poruke 81
3.5 Unos podataka 82
3.5.1 Obeležavanje elemenata u veb obrascu 85
3.6 Modalni prozori 88
3.7 Praćenje napretka 90
3.8 Interaktivnost 90
3.8.1 Metodičnost ili efikasnost 90
3.8.2 Predvidljivost 91
3.8.3 Sugestije 91
3.9 Prilagodljivi dizajn 92

4. BEZBEDNOST VEB APLIKACIJA 97

4.1 Identifikacija i autorizacija korisnika 97
4.2 Sesije 98
4.3 Sanitacija i validacija ulaznih podataka 102
4.3.1 Sanitacija podataka 102
4.3.2 Validacija podataka 103
4.4 Sigurnosni testovi 104
4.5 Zaštita od SQL Injekcija 105
4.6 Šifrovanje podataka 109
4.7 Zaštita od Cross-Site Scripting (XSS) 110
4.8 Redovno ažuriranje i zakrpe 110
4.9 Monitoring i detekcija pretnji 111

5. INFORMACIONI SISTEMI 115

5.1 Komponente informacionog sistema 115
5.2 Klasifikacija informacionih sistema 116

5.3 Tipovi informacionih sistema ... 117
5.3.1 Poslovni informacioni sistemi i upravljanje resursima ... 117
5.3.2 Sportski informacioni sistemi ... 118
5.3.3 Ekspertski sistemi ... 119
5.3.4 Veb pretraživači ... 119
5.3.5 Geografski informacioni sistemi ... 121
5.3.6 Globalni informacioni sistemi ... 124
5.4 Automatizacija informacionih sistema ... 126
5.5 Namena informacionih sistema ... 127
5.5.1 Upravljački informacioni sistemi ... 127
5.5.2 Izvršni informacioni sistemi ... 127
5.5.3 Sistemi za podršku odlučivanju ... 128
5.6 Informacioni sistemi kao veb aplikacije ... 128
5.7 Trendovi u razvoju informacionih sistema ... 129
5.7.1 Inteligentni informacioni sistemi ... 130
5.7.2 Simulacije ... 131
5.7.3 Virtuelna stvarnost ... 132
5.7.4 Proširena stvarnost ... 133

6. OSNOVNE FUNKCIJE INFORMACIONIH SISTEMA ... 139

6.1 Prikupljanje podataka ... 139
6.2 Skladištenje podataka ... 139
6.2.1 Relacione baze podataka ... 140
6.2.2 NoSQL baze podataka ... 140
6.2.3 Objektno-orijentisane baze podataka ... 140
6.2.4 Ostale metode skladištenja podataka ... 141
6.2.5 Problemi ... 141
6.3 Obrada podataka ... 144
6.4 Distribucija podataka ... 145
6.5 Analiza podataka ... 146

7. MODELIRANJE INFORMACIONIH SISTEMA ... 153

7.1 Upravljanje kvalitetom ... 154
7.1.1 ISO 9000 standard ... 155
7.1.2 Dokumenti kvaliteta ... 156
7.2 Modeliranje organizacione strukture ... 158
7.2.1 Tokogrami aktivnosti ... 160
7.2.2 Distribucija ovlašćenja i autorizacija ... 166

8. IZVEŠTAVANJE U INFORMACIONIM SISTEMIMA ... 175

8.1 Poslovno obaveštavanje ... 176

8.2 Filtriranje podataka....176
8.2.1 Grupisanje podataka....178
8.2.2 Upotreba boja i grafičkih simbola u izveštavanju....180
8.3 Tipovi izveštaja....181
8.4 Kreiranje i personalizacija izveštaja....183
8.5 Vizuelizacija podataka....184
8.5.1 Alati za vizuelizaciju....188

9. PRIMERI INFORMACIONIH SISTEMA....195

9.1 Menadžment maloprodaje....195
9.1.1 eCommerce izveštaji....197
9.2 Proizvodni sektor....199
9.3 Loyalty programi....200
9.4 Sportski informacioni sistemi....201

10. PRIMER PROJEKTNOG ZADATKA....209

10.1 Struktura baze podataka....209
10.2 Generisanje test podataka....211
10.3 Izveštavanje....216
10.4 API....221
10.5 Grafički prikaz, grafikoni....223
10.5.1 Broj pregleda po članku....223
10.5.2 Broj pregleda za izabrani članak....226
10.5.3 Presek stanja....230

11. LITERATURA....241

INDEKS SLIKA....245

INDEKS LISTINGA....249

INDEKS TABELA....251

PREDGOVOR

Veb informacioni sistemi (IS) predstavljaju ključnu komponentu u složenom i dinamičnom upravljanju podacima i optimizaciji procesa, kako za organizacije, tako i za pojedince.

Ova knjiga je namenjena studentima koji prate predmet *Veb informacioni sistemi* na trećoj godini studija *Tehničkog fakulteta Univerziteta Singidunum*, na studijskom programu *Softversko i informaciono inženjerstvo*. Od studenata se očekuje da su prethodno odslušali predmete *Veb dizajn*, *Veb programiranje* i *Baze podataka*, odnosno da su upoznati sa *HTML* markirnim jezikom, *CSS*-om, osnovama *SQL* jezika, *PHP* i *JavaScript* skriptnim jezicima, kao i sa osnovnim principima objektno orijentisanog programiranja.

Tehnologija za razvoj veb IS sa kojom će se studenti upoznati na ovom predmetu je bazirana na *WAMP/LAMP/XAMPP* platformi koju čine *Apache* veb server, skriptni jezici *PHP* na serverskoj strani i *JavaScript* na klijentskoj strani, uz upotrebu *MySQL* ili *MariaDB* baze podataka.

Početak knjige fokusira se na osnovne pojmove koji su temelj razumevanja IS. Ovi pojmovi uključuju osnovne definicije i principe koji omogućavaju bolje shvatanje kako se podaci i informacije integrišu u različite sisteme. Razumevanje ovih elemenata je preduslov za dalje proučavanje složenijih aspekata IS kao veb aplikacija.

Nakon uvodnih pojmova, knjiga se bavi internetom i osnovnim postavkama tehnologije izrade veb aplikacija. U delu posvećenom dizajnu korisničkog interfejsa (UI), istražuju se principi koji omogućavaju kreiranje intuitivnih i za korisnike prijatnih aplikacija.

Bezbednost je ključna komponenta svakog IS, i zato se njen značaj analizira u posebnom poglavlju. Razumevanje bezbednosnih praksi, uključujući identifikaciju i autorizaciju korisnika, kontrolu sesija, sanitaciju i validaciju podataka, predstavlja osnovu za očuvanje integriteta i poverljivosti podataka u digitalnom okruženju.

Osnovne komponente, funkcije, tipovi, namena i klasifikacija IS su detaljno obrađeni u narednim poglavljima, sa ciljem da čitaoca usmeri ka prepoznavanju ključnih uloga IS u različitim kontekstima i organizacijama.

Knjiga takođe obrađuje temu izveštavanja i vizuelizacije podataka, naglašavajući značaj efikasnog prikazivanja informacija kako bi se olakšalo donošenje odluka.

Na kraju, knjiga se oslanja na praktične primere veb IS koji pružaju uvid u stvarne aplikacije i implementacije. Analizom ovih primera, čitaoci mogu bolje da razumeju kako se teorijski koncepti primenjuju u praksi, čime se povezuje teorija sa stvarnim svetom.

Ova knjiga predstavlja uvod u svet IS zasnovanih na veb tehnologijama, sa ciljem da čitaocima obezbedi osnovu za razumevanje koncepta razvoja i primene ovog tipa aplikacija u različitim kontekstima i situacijama.

Iako je knjiga namenjena studentima kao udžbenik, može je koristiti svako ko želi da se dublje upozna sa tehnologijama i principima razvoja IS kao veb aplikacija.

Primedbe, komentare, sugestije i informacije o uočenim greškama možete poslati na email adresu mdobrojevic@singidunum.ac.rs

Autor

Beograd, 2024.

OSNOVNI POJMOVI

1. OSNOVNI POJMOVI

1.1 Sistem

Sistem je koncept koji prožima gotovo svaki aspekt naših života, od bioloških procesa koji nas održavaju do složenih mehanizama koji pokreću naš tehnološki napredak. Pojam sistema može da se predstavi kao grupa međusobno povezanih elemenata koji rade zajedno da bi postigli zajednički cilj ili svrhu. Sistemi se mogu uočiti u različitim kontekstima, uključujući biološke sisteme kao što su ekosistemi ili ljudsko telo, mehaničke sisteme poput vozila ili kompjutera, društvene sisteme kao što su vlada ili ekonomija neke države, i sl.

U suštini, sistem se može definisati pomoću njegovih komponenti, odnosa između ovih komponenti i načina na koji one funkcionišu zajedno kao jedinstvena celina. Ovaj koncept se široko koristi u različitim oblastima kao što su nauka, inženjering, menadžment i sociologija, za razumevanje složenih pojava i rešavanje problema.

U oblasti biologije, pojam sistema može da se prepozna već na ćelijskom nivou gde organele sarađuju unutar ćelija kako bi izvršile specifične funkcije, pa sve do ekosistema u kojima različiti organizmi komuniciraju jedni sa drugima i svojim okruženjem u delikatnoj ravnoteži. Samo ljudsko telo je skup međusobno povezanih sistema, kao što su kardiovaskularni sistem koji isporučuje kiseonik i hranljive materije ćelijama, nervni sistem koji koordinira telesne funkcije i reakcije, i imuni sistem koji brani organizam od patogena.

Izvan fizičkog područja, društveni sistemi upravljaju ljudskim interakcijama i organizacijama. Vlade funkcionišu kao sistemi sa ograncima i agencijama koje sarađuju sa ciljem upravljanja i pružanja usluga. Ekonomski sistemi

uključuju složene mreže proizvođača , potrošača i regulatora koji pokreću proizvodnju, distribuciju i potrošnju dobara i usluga.

Razumevanje sistema uključuje analizu njihovih komponenti, njihovih međusobnih veza i ponašanja koje proizilazi iz ovih interakcija . Sistemsko razmišljanje naglašava holističke perspektive , uviđajući da promene u jednom delu sistema mogu uticati na sistem u celini. Ovakav pristup je ključan za rešavanje problema i donošenje odluka , jer omogućava da se shvati složenost sistema, i da se u širem kontekstu razume kako akcije lokalnog karaktera mogu da utiču na sistem u celini, odnosno da se predvide posledice.

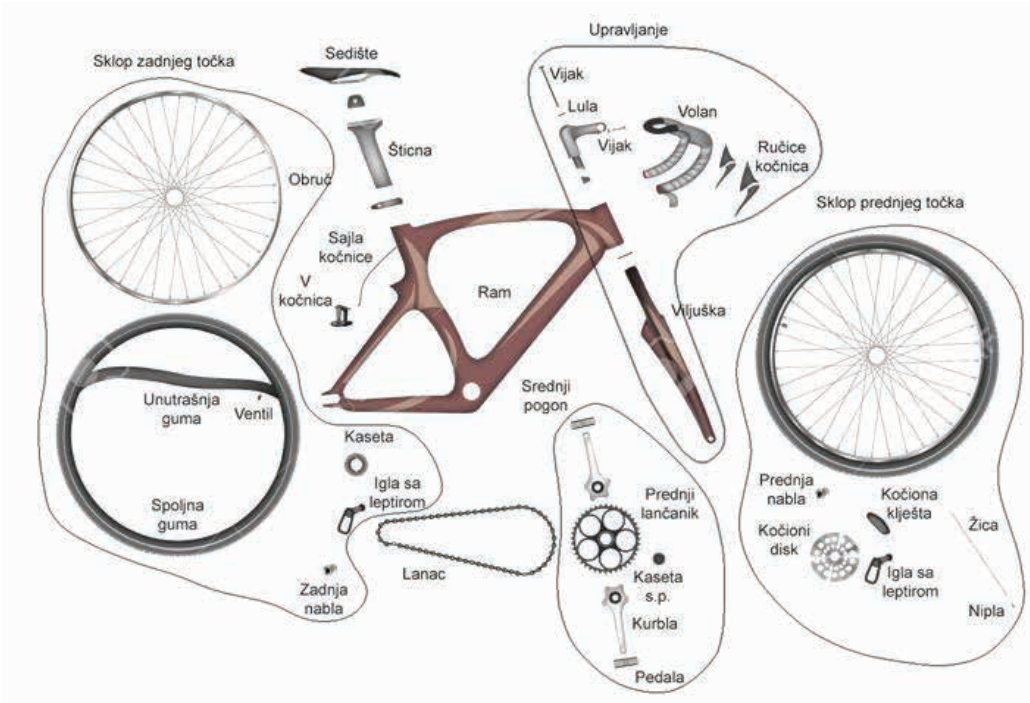
Sistemi su dinamični i prilagodljivi, sposobni za samoregulaciju i evoluciju . Petlje za povratne informacije unutar sistema omogućavaju samoispravljanje i prilagođavanje , održavanje stabilnosti ili prilagođavanje promenljivim uslovima . Međutim, sistemi takođe mogu pokazati ranjivosti , gde se poremećaji ili kvarovi u jednom delu sistema mogu proširiti i uticati na sistem u celini.

Koncept sistema je osnova našeg razumevanja sveta , pružajući okvire za analizu, projektovanje i upravljanje u različitim domenima. Bilo da se radi o složenom biološkom organizmu, optimizaciji tehnoloških sistema, ili složenim društvenim strukturama, upotreba potencijala sistemskog razmišljanja je od suštinskog značaja za rešavanje izazova i podsticanje inovacija.

1.1.1 Podsistem

Podsistem je dio šireg sistema koji ima specifičnu funkciju ili svrhu unutar tog sistema, i kao takav doprinosi ostvarivanju krajnjeg cilja ili svrhe sistema u celini. Primeri podsistema mogu biti pogonski sistem u automobilu, koji obuhvata motor, prenosni sistem i diferencijal, IS u kompaniji koji uključuje hardver, softver, mrežu i ljude, ili čak biološki sistem u organizmu koji uključuje organe, tkiva i ćelije.

U inženjerstvu, sistemi se precizno projektuju sa ciljem da ispunjavaju određenu svrhu. Mehanički sistem kao što je bicikl, sadrži nekoliko podsistema bez kojih ne bi mogao da ispuni svoju osnovnu svrhu, slika 1:



Slika 1 - Podsistemi mehaničkog sistema (bicikl)

- *Pogonski sistem* obuhvata srednji pogon koji čine pedale, kurble, lančanike, kaseta, lanac i zadnji točak. Pedale se koriste za pokretanje bicikla, a prenos snage se ostvaruje preko lanca koji povezuje zupčanike na pedalama i zadnjem točku. Prednji i zadnji menjač omogućavaju vozaču da promeni prenosni odnos između srednjeg pogona i zadnjeg točka, u skladu sa terenom ili preferencijama.
- *Kočioni sistem* omogućava zaustavljanje bicikla i čine ga ručice kočnica, sajle kočnice i kočnice na prednjem i zadnjem točku.
- *Upravljački sistem* uključuje prednju viljušku i volan.

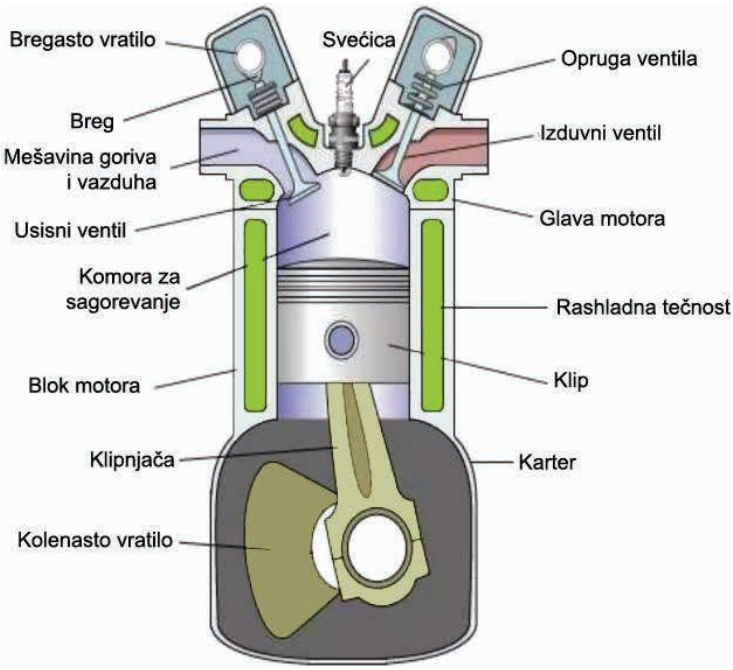
Sa druge strane, ukoliko se mala porodična kuća posmatra kao sistem,

u podsisteme tog sistema koji će moći da pruže potrebne uslove za život i boravak stanara, mogu se uvrstiti:

- *Električni sistem* obuhvata električne instalacije, utičnice, prekidače, osvetljenje, sigurnosne uređaje poput osigurača, kao i električne aparate i uređaje u kući.
- *Vodovodni sistem* uključuje vodovodne cevi, slavine, bojlere, odvode za otpadnu vodu, sudoperu, umivaonik, kadu ili tuš kabinu i WC šolju.
- *Sistem za grejanje i klimatizaciju* obuhvata grejne uređaje kao što su TA peći, radijatori, podno ili zidno grejanje, klima uređaje i uređaje za provetravanje prostorija.
- *Sigurnosni sistem* obuhvata alarm za slučaj provale, video nadzor, senzore dima i ugljen-monoksida, kao i druge sigurnosne uređaje za zaštitu objekta i stanara.
- *Komunikacioni sistem* uključuje fiksne i bežične telefone, priključke za internet, kablovsku i/ili satelitsku televiziju, kao i druge uređaje za komunikaciju unutar kuće i sa spoljnim svetom.
- *Kuhinjski sistem* obuhvata kuhinjske aparate poput šporeta, frižidera, sudoperu i mašinu za pranje sudova, kuhinjske elemente, ormariće i radne površine.

Automobilski motor, odnosno motor sa unutrašnjim sagorevanjem, takođe može da se posmatra kao mehanički sistem koji sadrži nekoliko podsistema bez kojih ne bi mogao da ispuni svoju osnovnu svrhu, a to je pokretanje motornog vozila:

- *Sistem za dovod i pripremu goriva* se sastoji od rezervoara za gorivo, pumpe za gorivo, sistema za ubrizgavanje ili karburatora, filtera goriva i drugih komponenti koje su odgovorne za isporuku goriva u motor.
- *Sistem paljenja* je odgovoran za stvaranje varnice u svećici koja inicira sagorevanje smese goriva i vazduha u cilindru motora. Ovaj sistem obuhvata bobine, svećice, senzore položaja radilice i kompjuter motora.



Slika 2 - Komponente benzinskog motora u automobilu



Slika 3 - Lični računar, podsistemi

- *Sistem za upravljanje vazduhom* reguliše protok vazduha u motor radi priprem optimalne smese goriva i vazduha za sagorijevanje, i obuhvata usisnu granu, filter vazduha, bregastu osovinu, a opciono turbo punjač ili kompresor.
- *Sistem izduvnih gasova* vodi izduvne gasove iz motora kroz izduvnu granu, katalizator i izduvni lonac izvan vozila.
- *Sistem za hlađenje i podmazivanje* obuhvata hladnjak, rashladnu tečnosti, termostat, filter i pumpu za ulje. Namena sistema je da održava optimalnu temperaturu motora kako bi se sprečilo pregevanje, i osiguralo podmazivanje vitalnih delova motora radi smanjenja trenja.

Ukoliko bi se lični računar (engl. *Personal Computer, PC*) posmatrao kao sistem, podsistemi bi mogli da se podele na hardverske, softverske i mrežne, slika 3.

1.1.2 Komponenta sistema

Komponenta sistema je osnovni element ili deo sistema koji sistem čini funkcionalnim. Svaka komponenta u sistemu ima svoju konkretnu ulogu, može da bude direktno ili indirektno povezana sa drugim komponentama i doprinosi integrisanom funkcionisanju sistema kao celine.

To može biti fizička komponenta, kao što su hardver, uređaji ili mehanički deo. Takođe, komponenta sistema može biti softverski entitet, kao što su operativni sistemi (OS), drajveri, aplikacije i drugi programi koji omogućavaju rad sistema.

Na primer, u kontekstu ličnog računara kao kompjuterskog sistema, komponente sistema obuhvataju hardverske komponente i softver koji zajedno čine računar funkcionalnim, slika 3:

- *Hardverski podsistem* čine komponente kao što su procesor (CPU) i matična ploča, RAM memorija, hard disk (HDD ili SSD), grafička kartica i napajanje računara.

- *U softverski podsistem* se ubrajaju *operativni sistem* koji omogućava interakciju između korisnika i hardvera, upravlja resursima računara i pokreće aplikacije, *aplikacije i programi* koje korisnici koriste za obavljanje različitih zadataka, *drajveri* koji omogućavaju komunikaciju između hardverskih komponenti i operativnog sistema i *sigurnosni softver* koji uključuje antivirus programe, firewall i druge alate za zaštitu računara od virusa, malvera i neovlašćenog pristupa.
- *Mrežni podsistem* obuhvata *mrežnu karticu* koja omogućava povezivanje računara sa lokalnom mrežom (LAN) ili internetom, *rutiranje i povezivanje* koji predstavljaju softverske i hardverske mehanizme koji omogućavaju prenos podataka između računara u mreži, i napokon protokole koji predstavljaju standardizovane metode za prenos podataka, kao što je npr. TCP/IP protokol.

U širem smislu, komponenta sistema može da bude bilo koji deo ili resurs koji je neophodan za funkcionisanje podsistema, ili sistema u celini, bez obzira da li se radi o računarskom sistemu, mehaničkom sistemu, ekosistemu, organizacionom sistemu ili bilo kojem drugom tipu sistema.

Razlika između podsistema i komponente sistema je u tome što su komponente osnovni delovi sistema, dok su podsistemi složenije strukture koje integrišu više komponenti sa ciljem realizacije određene funkcije sistema. Podsistem može biti hijerarhijski organizovan, što znači da može imati više nivoa komponenti unutar sebe.

1.2 Podaci

Podaci su osnovni ulazni element IS i predstavljaju neobrađene informacije u izvornom obliku, koje pojedinačno posmatrano nemaju posebnu svrhu, odnosno značaj. To mogu biti imena i prezimena osoba u adresaru, njihove adrese stanovanja, brojevi telefona ili datumi rođenja. Podaci mogu biti finansijske transakcije u veb prodavnici, temperature vazduha, vazdušnog pritiska ili vlažnosti vazduha kao rezultat merenja senzora u meteorološkoj stanici. Podaci mogu da budu i rezultati merenja senzora u industrijskom okruženju, kao i tekstualni dokumenti, slike, video zapisi, ili bilo koji drugi tip dokumenta.

Pojedinačno posmatrano, podaci ne moraju da imaju konkretan smisao ili kontekst. Oni postaju korisni tek kada se obrade, organizuju ili analiziraju u odgovarajućem okruženju, čime dobijaju značenje i postaju korisni za donošenje odluka, pružanje informacija ili podršku poslovnim procesima u IS.

1.2.1 Struktura podataka

Podaci mogu da budu zabeleženi u različitim formatima, kao što su tekstuelni, numerički, grafički, zvučni ili video sadržaji, a način na koji su podaci organizovani i uskladišteni u računarskom sistemu predstavlja strukturu podataka.

U IS se često koriste različite strukture podataka, u zavisnosti od tipa podataka, konkretnih potreba i drugih faktora koje sistem treba da zadovolji. Izbor adekvatne strukture podataka je bitan za efikasnost operacija pristupa, pretrage, kreiranja i brisanja podataka unutar sistema, a u nastavku su navedene neke od najčešće korišćenih struktura podataka:

- *Relaciona struktura podataka* u bazama podataka koristi tabele koje su povezane prema određenim pravilima (ključevima) kako bi se organizovali podaci. Ovakav pristup se koristi u u sistemima za upravljanje bazama podataka (engl. *Database Management Systems,* DBMS) kao što su MySQL, Oracle, Microsoft SQL Server i sl. Nasuprot relacionim strukturama podataka postoji i niz drugih modela baza podataka koji se koriste u specifičnim situacijama, i to su hijerarhijski modeli, mrežni modeli, objektno-relacioni modeli, grafički modeli i sl.
- *Sistem falova* predstavlja organizuju podataka u fajlove i direktorijume na disku. Fajlovi mogu biti struktuirani, poput tekstualnih datoteka, ili nestruktuirani, kao što su slike ili video zapisi. *Hijerarhijski fajl sistem* predstavlja organizaciju podataka u obliku stabla.
- *JSON* (engl. *JavaScript Object Notation*) i *XML* (engl. *eXtensible Markup Language*) predstavljaju formate za razmenu podataka koji se često koriste u IS za prenos i skladištenje strukturiranih podataka.

Obe strukture omogućavaju organizaciju podataka u hijerarhijskom formatu.

- *Grafički modeli* se koriste za predstavljanje složenih relacija i entiteta u IS. Ovo može uključivati usmerene i neusmerene grafove koji povezuju entitete i njihove veze.
- *Heš tabele* se koriste za brz pristup podacima na osnovu ključeva. Ovakav pristup je efikasan za primenu u relacionim bazama podataka, za indeksiranje u pretraživačima, autentifikaciju korisnika, za usmeravanje zahteva u distribuiranim sistemima, ili za brzi pristup keširanim podacima.

1.2.2 Informacija

Informacija se generiše kroz procese analize, interpretacije, filtriranja i prezentacije podataka kako bi se došlo do korisnih rezultata koji su od suštinskog značaja za efikasan rad IS. Dok su podaci sirove činjenice ili detalji koji sami po sebi ne moraju da imaju značenje ili svrhu, informacija se dobija obradom tih podataka kako bi se dobilo korisno znanje ili rezultat. Kao ključne karakteristike informacije u IS mogu se navesti:

- Informacija ima *smisao i značenje* koje omogućava razumevanje situacije, problema ili trendova. Ona pruža korisne uvide i omogućava donošenje informisanih odluka.
- Informacija je *korisna* zato što može biti upotrebljena za određene svrhe, kao što su analiza, predviđanje, komunikacija, planiranje, donošenje odluka ili rešavanje problema.
- Informacija je *relevantna* ako je relevantna za trenutnu situaciju, potrebe ili ciljeve korisnika IS. Relevantnost informacije je ključna za njenu efikasnu upotrebu.
- Informacija treba da bude *precizna i tačna* kako bi se izbegle netačne interpretacije, ili donošenje pogrešnih zaključaka prilikom korišćenja informacija u procesima donošenja odluka.
- Informacija treba da bude *pristupačna*, odnosno dostupna i razumljiva za korišćenje korisnicima IS kada im je potrebna.

- Informacija je *aktuelna* ako je relevantna u datom trenutku i situaciji, a aktuelnost informacija je suštinski važna za donošenje odluka.

1.2.3 Analiza

Proces analize podataka u IS ima primenu u različitim oblastima, uključujući poslovnu i finansijsku analizu, medicinsku dijagnostiku, istraživanje tržišta, optimizaciju procesa, predviđanje trendova i sl. Ovaj proces omogućava donošenje odluka zasnovanih na činjenicama i podacima, što može dovesti do efikasnijeg poslovanja, a samim tim i boljih poslovnih rezultata.

Analiza podataka u IS obuhvata procese obrade, interpretacije i izvlačenja korisnih informacija iz skupova podataka radi donošenja odluka, otkrivanja uzoraka, identifikacije trendova i rešavanja problema. Analiza se može sprovoditi primenom različitih tehnika, alata i metoda, u zavisnosti od specifičnih ciljeva i zahteva sistema.

Da bi se podaci pripremili za analizu, neophodno je da se prvo prikupe, a potom i obrade, što podrazumeva njihovo čišćenje i transformaciju u pogodan oblik. Moguće je da proces obuhvati aktivnosti kao što su filtriranje nepotpunih ili nevažećih podataka, integraciju podataka iz različitih izvora, normalizaciju podataka i sl.

Uvid u osnovne karakteristike podataka kao što su srednja vrednost, raspodela, trendovi i sl. se dobija *deskriptivnom analizom*, čiji je fokus na opisu karakteristika podataka kroz statističke metode, grafikone, tabele i druge metode vizualizacije podataka.

Prediktivna analiza primenom odgovarajućih modela i algoritama, kao što su regresiona analiza, analize vremenskih serija i mašinsko učenje, se koristi da bi se na osnovu istorijskih podataka predvideli budući trendovi, ponašanja ili događaji.

Identifikacija uzoraka, grupisanje podataka u srodne kategorije, ili otkrivanje nepravilnosti u skupovima podataka može da se obavi primenom tehnike *pronalaženja uzoraka* (engl. *Pattern Recognition*), a metode koje se

najčešće koriste su klasterovanje (engl. *Clustering*) i klasifikacija (engl. *Classification*).

Dubinska analiza (engl. *Deep Analysis*) se nadovezuje na prediktivnu analizu i koristi kompleksne algoritme za dublje razumevanje podataka, otkrivanje skrivenih veza i zaključaka koji nisu očigledni na prvi pogled.

1.3 Ljudski faktor

Kvalitet podataka ima značajan uticaj na rad IS zato što utiče na tačnost i pouzdanost informacija na osnovu kojih se donose odluke. Takođe, utiče na efikasnost poslovnih procesa, tačnost analiza i izveštaja, kao i na pouzdanost informacija koje se razmenjuju unutar i van sistema.



Slika 4 - Spremnost zaposlenih da prihvate uvođenje IS na osnovu učinka [1]

Iz perspektive menadžmenta, donošenje poslovnih odluka na osnovu analize prikupljenih podataka je svakako značajna prednost. Međutim, to ne znači da će svi zaposleni sa odobravanjem prihvatiti uvođenje IS u svakodnevni rad, dok se u pojedinim grupama može očekivati i opstrukcija,.

Loš kvalitet podataka može da implicira donošenje pogrešnih odluka, gubitak poverenja u IS, dodatne troškove za ispravke grešaka i smanjenje produktivnosti zbog nedostatka pouzdanih informacija. Zato je važno da

organizacije imaju jasne procese, procedure i politike koje podržavaju visok nivo kvaliteta podataka u IS, kao i da pruže odgovarajuću obuku i podršku korisnicima kako bi se smanjile ljudske greške i poboljšao kvalitet podataka.

Ljudski faktor i urođeni otpor prema promenama može značajno da utiče na kvalitet podataka u IS u različitim tačkama procesa. Zaposleni često percipiraju nove sisteme kao pretnju svojoj radnoj rutini ili kao dodatni teret, što može da uspori i oteža implementaciju sistema, slika 4 [1].

- *Unos podataka.* Kvalitet podataka zavisi od tačnosti i preciznosti prilikom unosa podataka u sistem. Ljudske greške poput nenamernih grešaka u unosu, tipografskih grešaka ili nedostatka pažnje mogu dovesti do loših podataka.
- *Validacija i provera podataka.* Ljudi su odgovorni za proveru i validaciju podataka kako bi se osiguralo da su informacije tačne, relevantne i potpune pre nego što budu upotrebljene u IS. Nedostatak pažnje u procesu validacije može dovesti do loših podataka.
- *Odgovornost za integritet podataka.* Ljudi su odgovorni za održavanje integriteta podataka, što uključuje ažuriranje, brisanje ili izmenu podataka kada je to potrebno. Neodgovorno ili neprecizno upravljanje podacima može uticati na njihovu kvalitet.
- *Kontrola pristupa.* Ljudi su odgovorni za kontrolu pristupa podacima i osiguravanje da samo ovlašćeni korisnici imaju pristup određenim podacima. Nepoštovanje pravila i procedura može dovesti do neovlašćenog pristupa i oštećenja podataka.
- *Korišćenje podataka.* Način na koji ljudi koriste podatke takođe može uticati na kvalitet podataka. Na primer, nepravilna interpretacija podataka, netačne analize ili donošenje odluka na osnovu nepotpunih informacija može dovesti do loših posledica.
- *Nivo obuke i svest o važnosti kvaliteta podataka* među korisnicima IS može imati veliki uticaj na kvalitet podataka. Dobro obučeni korisnici će biti više svesni važnosti tačnosti, preciznosti i potpunosti podataka.

1.4 Znanje

Znanje je sposobnost razumevanja i primene informacija, ideja, činjenica ili veština, i obuhvata sve što čovek uči, shvata ili otkriva o svetu oko sebe.

Znanje može da se stekne na različite načine, kroz iskustvo, ili putem obrazovanja, istraživanja ili posmatranja. Omogućava nam da donosimo kvalitetne odluke, rešavamo probleme i razvijamo nove ideje. Postoje različite vrste znanja koje se mogu svrstati u sledeće kategorije:

- *Eksplicitno znanje* je znanje koje je lako prenosivo i može se formalno izraziti u rečima ili brojevima. Ovo znanje može biti dokumentovano i deljivo putem pisanih ili elektronskih medija.
- *Implicitno znanje* je znanje koje je teže artikulisati ili formalizovati. To obuhvata veštine, intuiciju, iskustvo i instinkte koji nisu lako prenosivi ili opisivi.
- *Individualno znanje* je znanje koje pojedinac poseduje na osnovu svog iskustva, obrazovanja, veština i sposobnosti. To može biti implicitno ili eksplicitno i često je specifično za pojedinca.
- *Kolektivno znanje* je znanje koje je zajedničko grupi ili organizaciji. To može biti rezultat saradnje, komunikacije i deljenja informacija unutar organizacije.
- *Tehničko znanje* je znanje koje se odnosi na tehničke veštine, kao što su programiranje, inženjering, dizajn ili upravljanje sistemima.
- *Procesno znanje* je znanje o procesima i procedurama koji se koriste za obavljanje određenih zadataka ili aktivnosti.

Ove kategorije se često preklapaju i mogu se koristiti za bolje razumevanje i upravljanje znanjem u organizacijama.

U kontekstu IS, znanje obuhvata skup informacija, veština i procesa koji se koriste za analizu, interpretaciju i primenu podataka radi donošenja odluka i rešavanja problema. To može uključivati koncepte kao što su baze podataka, alati za analizu podataka, veštačka inteligencija (engl. *Artificial Intelligence*, AI) i mašinsko učenje (engl. *Machine Learning*, ML), a sve sa

ciljem da pomogne organizacijama da bolje razumeju svoje poslovanje, identifikuju trendove, predviđaju buduće događaje i optimizuju procese.

1.5 Odluka

Odluka je proces izbora između dve ili više mogućih alternativa, rezultat razmišljanja, analize ili procene situacije na osnovu čega se bira kurs sledeće akcije.

Odluke se donose u svim aspektima života, od svakodnevnih situacija do poslovnih strategija. Mogu da budu jednostavne, kao što su izbor između dva različita jela za večeru, ili složene, npr. izbor strategije kompanije za ulaganje u nove tehnologije.

Odluke mogu da budu zasnovane na intuiciji, iskustvu, analizi podataka ili kombinaciji različitih faktora, ali je važno da se napomene da od donete odluke zavise i posledice, zbog čega je važno da se pre donošenja odluke pažljivo razmotre sve relevantne informacije.

U poslovnom okruženju, IS imaju ključnu ulogu u podršci procesu donošenja odluka na svim nivoima organizacije:

- *Prikupljanje podataka*. IS automatski prikupljaju, obrađuju i skladište velike količine podataka iz različitih izvora, što pruža osnovu za informisane odluke.
- *Analiza podataka*. Informacioni sistemi omogućavaju analizu podataka putem različitih tehnika, kao što su statistička analiza ili alati za vizualizaciju podataka, sa ciljem da pomognu korisnicima da identifikuju obrasce, trendove i ključne informacije relevantne za donošenje odluka.
- *Modeliranje i predviđanje*. Napredne funkcije IS omogućavaju modeliranje i predviđanje budućih događaja na osnovu dostupnih podataka. Ovo omogućava organizacijama da bolje planiraju i pripreme se za buduće izazove.
- *Podrška procesu donošenja odluka*. IS obezbeđuju alate i tehnike koji podržavaju korisnike u procesu donošenja odluka, kao što su

interaktivni paneli, simulacije, scenariji ili automatizovani sistemi za donošenje odluka.

- *Distribucija informacija*. IS omogućavaju brzu i efikasnu distribuciju relevantnih informacija korisnicima na svim nivoima organizacije, što omogućava da odluke budu zasnovane na aktuelnim i tačnim informacijama.

1.6 Informisane poslovne odluke

Informisane poslovne odluke (engl. *Informed Business Decisions*) su odluke koje se donose na osnovu detaljne analize relevantnih, tačnih i aktuelnih podataka. Ove odluke se ne oslanjaju na intuiciju ili nasumične pretpostavke, već na konkretne podatke koji pružaju realan uvid u stanje i performanse poslovanja, čime se minimiziraju potencijalne nesigurnosti i greške.

Upotreba analitičkih tehnika kao što su statistička analiza i modeliranje podataka, omogućava efikasnije upravljanje resursima, optimizaciju procesa i planiranje strategije.

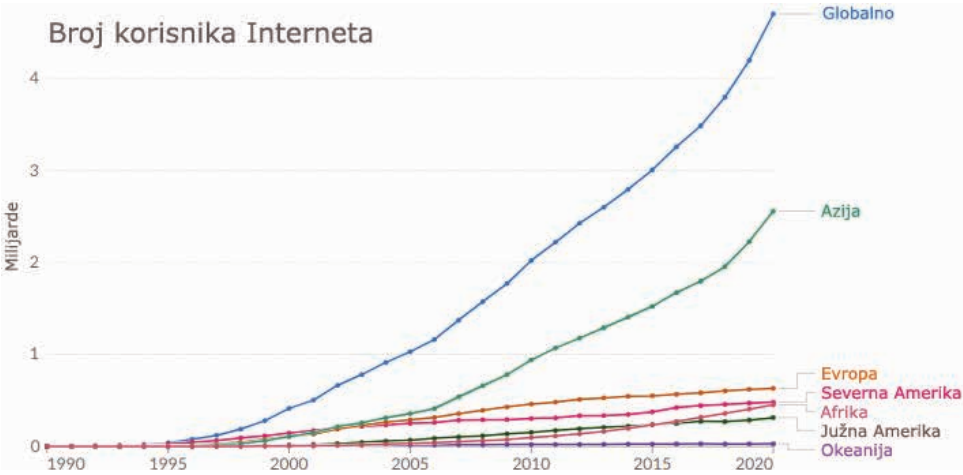
Takođe, informisane poslovne odluke omogućavaju bržu reakciju na promene u tržišnim uslovima i akcije konkurencije, pružajući jasniji uvid u buduće potrebe i prilike. Na taj način, poslovna inteligencija pomaže u donošenju odluka koje su preciznije, bolje osmišljene i efikasnije, koristeći podatke kao osnovu za kreiranje strategije i akcije.

INTERNET I VEB APLIKACIJE

2. INTERNET I VEB APLIKACIJE

Prema trenutnim podacima, 4.53 milijarde ljudi ili ~59% svetske populacije ima pristup Internetu, slika 5. [2] Internet je značajno promenio svakodnevni život ljudi širom sveta. Omogućio je trenutnu globalnu komunikaciju bez obzira na udaljenost.

Informacije su postale dostupne, što je olakšalo procese učenja, istraživanja i profesionalnog razvoja. Internet je pokrenuo nove industrije poput e-trgovine, onlajn edukacije i digitalnog marketinga, pružajući ljudima mogućnosti za rad i poslovni napredak. Značajno je promenio i način zabave i provođenja slobodnog vremena kroz *streaming* servise, onlajn igre i druge digitalne platforme.



Slika 5 - Broj korisnika Interneta

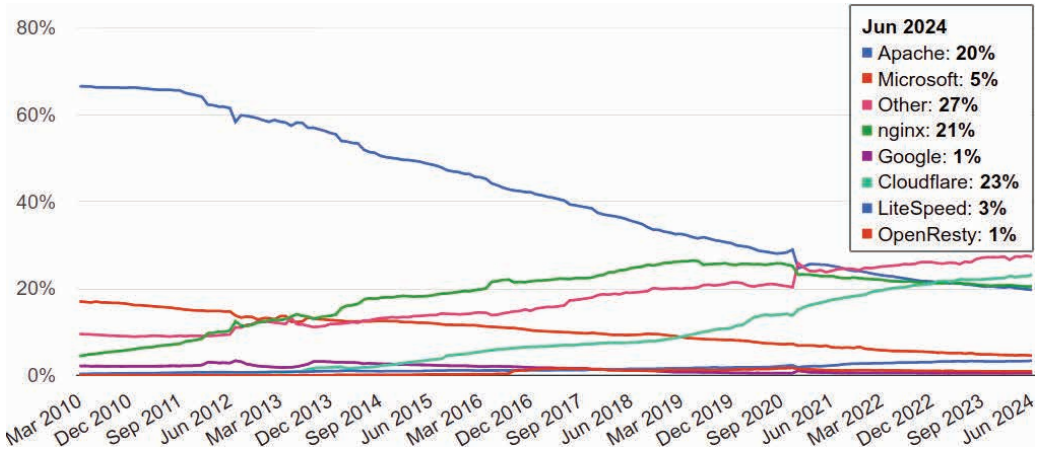
Internet je transformisao industriju i poslovno okruženje na globalnom nivou. Kompanijama je omogućio da prošire svoje tržište putem e-trgovine, što je otvorilo nove mogućnosti za mala i srednja preduzeća.

Takođe, Internet je unapredio način na koji kompanije komuniciraju sa svojim kupcima i partnerima. Društvene mreže, digitalni marketing i personalizovane kampanje omogućavaju direktnu interakciju sa potrošačima i bolje razumevanje njihovih potreba i želja, što je omogućilo efikasniji marketinški pristup i bolju uslugu kupcima.

Dalje, Internet je ubrzao procese poslovanja kroz automatizaciju i digitalizaciju. Kompanije mogu efikasnije da upravljaju podacima, što dovodi do optimizacije operativnih troškova i povećanja produktivnosti, dok zaposleni mogu da rade zajedno na projektima bez obzira na njihovu geografsku lokaciju.

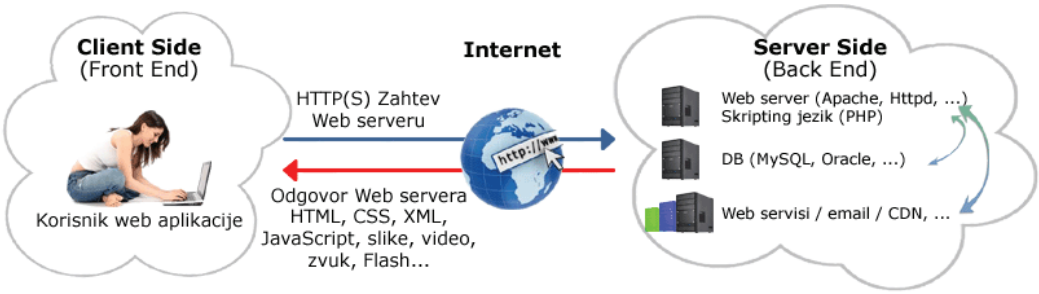
2.1 Veb serveri

Veb server je softverski sistem ili fizički uređaj koji upravlja i isporučuje veb sadržaj korisnicima putem interneta.



Slika 6 - Tržišni udeo veb servera, maj 2024.

Kada korisnik putem pretraživača pošalje zahtev za određenom veb stranicom, veb server prima taj zahtev, obrađuje ga i vraća odgovarajući sadržaj, odnosno odgovor, slika 7. To može da bude HTML dokument, slika, JavaScript ili CSS fajl, ili drugi dokument bilo kog tipa.



Slika 7 - Slanje zahteva ka veb serveru i prijem odgovora

Osim ovih osnovnih funkcija, veb serveri takođe igraju ključnu ulogu u sigurnosti i optimizaciji performansi, kroz implementaciju mera kao što su SSL/TLS šifrovanje, keširanje sadržaja za poboljšanje brzine rada, i zaštita podataka.

Pojam veb server može da se odnosi na softver namenski napravljen da prima i odgovara na ovakve zahteve, ali i na računar na kome je instaliran ovakav softver. Fizički mogu da se nalaze na ličnom računaru, u prostorijama kompanije, u specijalizovanim data centrima, mogu da budu virtuelne instance na fizičkom hardveru (engl. *Virtual Web Server*), ili mogu da budu virtuelni serveri u oblaku (engl. *Cloud Web Server*).

Najpopularnija softverska rešenja u ovoj oblasti u produkcionom okruženju su Cloudflare (23%), Nginx (21%) i Apache HTTP Server (20%), slika 6. Kada je u pitanju razvojno okruženje, dva najpopularnija servera su Apache HTTP Server i Nginx i dostupni su i za Windows i za Linux operativne sisteme, dok korisnici Windows OS imaju mogućnost upotrebe i Microsoft Internet Information Services (IIS).

2.2 OSI Model, transportni protokoli i HTTP protokol

OSI model (engl. *Open Systems Interconnection*) je konceptualni okvir koji sadrži sedam slojeva. Pomaže u razumevanju kako različiti slojevi mrežnog komunikacijskog sistema, od fizičkih veza do aplikacija, funkcionišu zajedno kako bi omogućili pristup veb sadržajima, pri čemu svaki sloj obavlja specifične funkcije i komunicira sa slojevima iznad i ispod sebe.

Slojevi	Jedinica	Protokoli
Aplikacija Mrežni procesi vezani za aplikaciju	Podatak	HTTP, HTTPS, FTP, SSH, DNS, SMTP
Prezentacija Enkripcija i kodiranje podataka	Podatak	MIME SSL TLS
Sesija Uspostavljanje sesije krajnjih korisnika	Podatak	RPC, PAP, SCP
Transport Veza, pouzdanost, transport	Segment Datagram	TCP, UDP
Mreža Logičko adresiranje i rutiranje	Paket	IP, ICMP, ARP, RARP
Sloj veze Fizičko adresiranje, pristup medijumu	Frejm (Okvir)	PPP, HDLC, Frame Relay
Fizički sloj Transmisija signala	Bit	Token Ring IEEE 802.11

Slika 8 - Slojevi OSI

- *Fizički sloj* se bavi prenosom sirovih bitova preko fizičkog medija, kao što su kablovi ili bežični signali.
- *Sloj veze.* Podaci se grupišu u okvire, uz detekciju i korekciju grešaka na fizičkom mediju. Ovaj sloj takođe upravlja pristupom mediju i osigurava da podaci stignu do odredišta bez grešaka.
- *Mrežni sloj* je odgovoran za usmeravanje paketa između različitih mreža. Koristi se za određivanje puta kojim će podaci putovati kroz mrežu, a ključni protokol u ovom sloju je IP (engl. *Internet Protocol*).
- *Transportni sloj.* Ovde se odvija segmentacija i ponovno sastavljanje podataka, kao i kontrola toka i grešaka. Transportni sloj osigurava da podaci budu dostavljeni u ispravnom redosledu. Najpoznatiji protokoli u ovom sloju su TCP (engl. *Transmission Control Protocol*) i UDP (engl. *User Datagram Protocol*).
- *Sesijski sloj* upravlja sesijama ili vezama između aplikacija. Organizuje i usklađuje komunikaciju između aplikacija na različitim uređajima.
- *Prezentacijski sloj*. Ovde se obavlja konverzija podataka, odnosno kodiranje i dekodiranje. Ovaj sloj obezbeđuje da podaci koji se šalju između aplikacija budu u razumljivom formatu.
- *Aplikativni sloj* je najviši sloj u OSI modelu, koji je direktno povezan sa aplikacijama korisnika. Ovde se odvija komunikacija između aplikacija, koristeći različite aplikacione protokole kao što su *HTTP*

(engl. *HyperText Transfer Protocol*), *FTP* (engl. *File Transfer Protocol*) i *SMTP* (engl. *Simple Mail Transfer Protocol*).

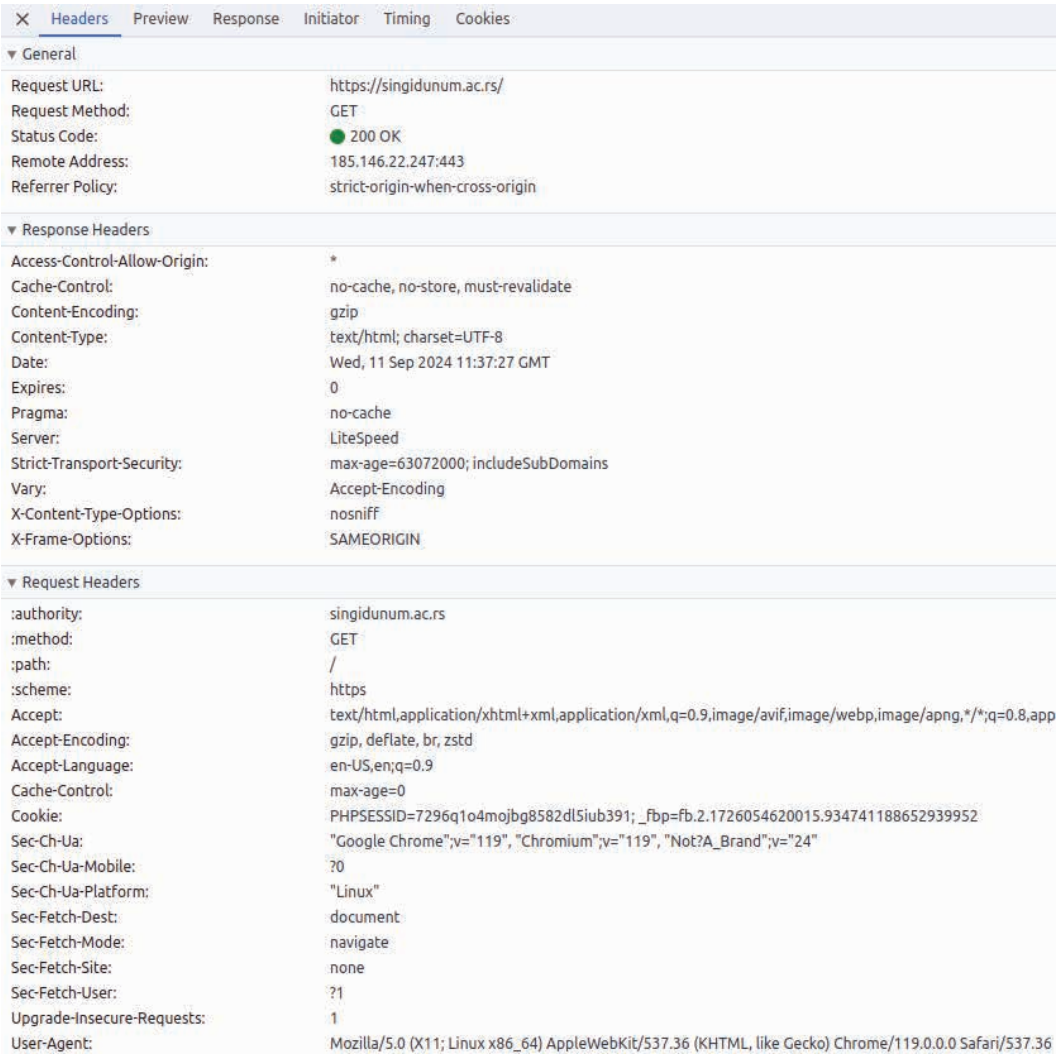
Transportni protokoli kao što su *TCP* (engl. *Transmission Control Protocol*) i *UDP* (engl. *User Datagram Protocol*), obezbeđuju pouzdan i efikasan prenos podataka kroz mreže između veb servera i veb aplikacija. TCP je usmeren na uspostavljanje pouzdane veze između dva uređaja, i koristi mehanizme kao što su potvrde prijema, retransmisije izgubljenih paketa i kontrola protoka kako bi obezbedio da svi podaci stignu na odredište ispravno i u ispravnom redosledu. Za razliku od TCP-a, UDP je jednostavniji i ne obezbeđuje mehanizme za pouzdanost, odnosno ne garantuje da će svi podaci stići na odredište ili da će stići u ispravnom redosledu. UDP je često korišćen za aplikacije gde je brzina važnija od pouzdanosti, kao što su *video streaming* i igre u realnom vremenu.

HTTP protokol (engl. *HyperText Transfer Protocol*) je aplikativni protokol koji omogućava komunikaciju između veb brauzera i veb servera slanjem zahteva i prijemom odgovora, obično koristeći TCP kao transportni protokol. HTTP/2 je uveden 2015. godine sa ciljem da poboljša performanse omogućavajući višestruke zahteve i odgovore na jednoj vezi, čime je smanjena latencija. HTTP/3 je treća revizija HTTP protokola, koji koristi QUIC, dodatno povećava brzinu i efikasnost tako što transport baziran na UDP protokolu, čineći ga izuzetno brzim, posebno za aplikacije u realnom vremenu i mobilne uređaje.

Standardni HTTP zahtev čine sledeće komponente, slika 9:

- *Metod.* Precizira akciju koju klijent želi da izvrši na resursu. Najčešće se koriste metode GET za preuzimanje resursa sa servera, POST za slanje podataka ka serveru i kreiranje resursa, PUT za ažuriranje postojećeg resursa i DELETE za brisanje resursa.
- *Putanja.* Deo zahteva koji označava resurs koji se zahvata. Na primer, /index.html označava da se traži fajl index.html u korenskom direktorijumu.
- *Verzija* HTTP protokola koji se koristi, kao što je HTTP/1.1 ili HTTP/2.

- *Zaglavlje zahteva* (engl. *Headers*) sadrže dodatne informacije o zahtevu, kao što su tip sadržaja, jezik, kodiranje i druge metapodatke.
- *Telo zahteva* je prisutno samo kod metoda kao što su POST ili PUT i sadrži podatke koji se šalju serveru. Na primer, kod POST zahteva, telo može sadržati podatke koji se šalju putem veb obrasca ili JSON podatke.



Slika 9 - HTTP zahtev i odgovor servera.

HTTP odgovor servera ima sličnu strukturu, slika 9:

- *Statusni kod*. Broj koji označava rezultat obrade zahteva.
- *Statusna poruka.* Kratka poruka koja prati statusni kod, kao što su *OK* ili *Not Found*.
- *Zaglavlje odgovora.* Slično kao kod zahteva, pruža dodatne informacije o odgovoru i resursu. Na primer, *Content-Type* precizira tip sadržaja u odgovoru.
- *Telo odgovora*. Deo odgovora koji sadrži stvarni sadržaj koji se šalje klijentu. Ovo može biti HTML stranica, JSON podatak, slika ili bilo koji drugi tip dokumenta.

Tabela 1 - HTTP odgovor servera, najčešći statusni kodovi

Kod	Poruka	Objašnjenje
200	OK	Zahtev je uspešno obrađen i odgovor sadrži tražene podatke.
201	Created	Resurs je uspešno kreiran kao rezultat zahteva, najčešće za POST metodu.
202	Accepted	Zahtev je prihvaćen za obradu, ali obrada još nije završena.
204	No Content	Zahtev je uspešno obrađen, ali nema sadržaja za vraćanje u odgovoru.
400	Bad Request	Zahtev nije ispravan ili nije mogao da bude obrađen zbog sintaktičkih grešaka.
401	Unauthorized	Zahtev zahteva autentifikaciju korisnika.
403	Forbidden	Pristup resursu je zabranjen, bez obzira na autentifikaciju.
404	Not Found	Traženi resurs nije pronađen na serveru.
405	Method Not Allowed	HTTP metoda korišćena u zahtevu nije dozvoljena za traženi resurs.
500	Internal Server Error	Došlo je do greške na serveru koja sprečava obradu zahteva.
501	Not Implemented	Server ne prepoznaje metodu zahteva ili je ne podržava.
502	Bad Gateway	Server je dobio neispravan odgovor od upstream servera.
503	Service Unavailable	Server trenutno nije dostupan zbog preopterećenja ili održavanja.
504	Gateway Timeout	Upstream server nije odgovorio na vreme, uzrokovalo je tajming problem u serveru.

2.3 Veb aplikacije

Veb aplikacije su softverske aplikacije koje se nalaze na veb serverima, a kojima korisnici pristupaju putem veb brauzera (engl. *web browser*) na svojim računarima, tablet uređajima ili pametnim telefonima, bez potrebe da se aplikacija prethodno instalira na samom uređaju. Korisnici mogu da pristupe aplikaciji sa bilo kog uređaja koji ima internet konekciju i instaliran veb brauzer. Postale su ključni deo svakodnevnog života, pružajući korisnicima i kompanijama alate za efikasno upravljanje informacijama, komunikaciju i pružanje usluga.

Važna osobina veb aplikacija je njihovo centralizovano održavanje i izvršavanje na veb serveru. To znači da se poslovna logika aplikacije nalazi na serveru, dok se korisnički interfejs i logika izvršavanja prenose i izvršavaju na korisničkom uređaju u okviru veb brauzera. Ažuriranjem aplikacije na serveru, najnovija verzija softvera odmah postaje dostupna svim korisnicima.

Ove aplikacije koriste tehnologije kao što su HTML, CSS i JavaScript (JS) za formiranje i stilizovanje veb stranica, i za kontrolu interakcije između veb stranice i korisnika. Korisnici mogu da komuniciraju sa aplikacijom, unose podatke, prikazuju informacije i izvršavaju druge akcije.

Veb aplikacije su takođe poznate po svojoj skalabilnosti. Zbog toga što se izvršavaju na jednom ili više servera, potrebni resursi lako mogu da se prilagode kako bi podržali povećani saobraćaj u kontesktu broja korisnika ili zahteva za obradom podataka, ili ugradnju novih funkcionalnosti. Kapaciteti veb aplikacije mogu da se skaliraju dodavanjem resursa ili optimizacijom koda, što je posebno važno u okruženjima sa velikim brojem korisnika.

Još jedna ključna karakteristika veb aplikacija je mogućnost integracije sa različitim servisima i API-jima (engl. *Application Programming Interface*, API). Ovo omogućava aplikaciji pristup dodatnim funkcionalnostima ili podacima iz drugih sistema, što može dovesti do efikasnijeg funkcionisanja sistema, i svakako može da obogatiti korisničko iskustvo.

2.4 Multi-platformska kompatibilnost

Pojam multi-platformska kompatibilnost odnosi se na sposobnost softverskog rešenja da funkcioniše na različitim operativnim sistemima, uređajima i okruženjima.

U kontekstu veb aplikacija, ova karakteristika omogućava da aplikacija bude dostupna i funkcionalna na različitim platformama, uključujući različite operativne sisteme, pretraživače i tipove uređaja, kao što su lični računari, tablet uređaji i pametni telefoni.

Prilikom projektovanja veb aplikacije i izbora platforme na kojoj će se aplikacija razvijati, obavezno mora da se uzme u obzir i činjenica da će aplikacija biti korišćena na različitim OS, uključujući Windows, macOS, Linux, kao i mobilne sisteme poput iOS i Android.

2.5 Troškovi razvoja i održavanja

Izbor tehnologije, platforme i tipa aplikacije koja se razvija zavisi od specifičnih potreba projekta. Pojednostavljeno posmatrano, razvoj veb aplikacija se najčešće zasniva na besplatnim veb tehnologijama kao što su PHP i MySQL/MariaDB na strani servera, i HTML, CSS i JS na klijentskoj strani, uz veliki broj dostupnih i takođe besplatnih radnih okvira i uz mnogobrojnu zajednicu koja pruža podršku i resurse.

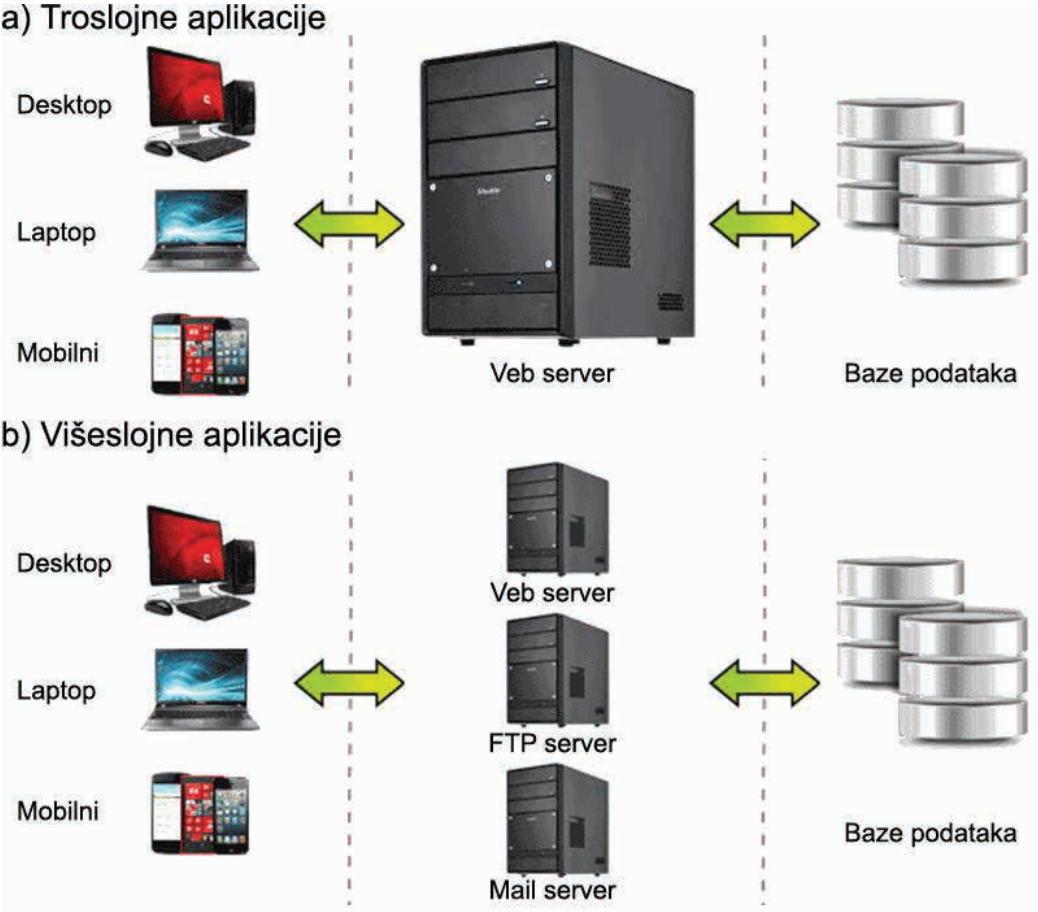
Troškovi razvoja i održavanja veb aplikacija su obično manji u poređenju sa ekvivalentnim desktop, odnosno izvornim (engl. *native*) aplikacijama razvijanim za konkretan OS ili hardversku arhitekturu, čiji razvoj često zahvata specifične tehnologije koje mogu uključivati dodatne troškove zbog upotrebe specifičnih alata i radnih okvira, kao i potrebe za razvojem različitih verzija aplikacije za svaki OS.

Troškovi održavanja veb aplikacija su takođe manji, zato što se ažuriranja primenjuju centralizovano na serveru, bez potrebe da se interveniše na uređajima samih korisnika.

Multi-platformska kompatibilnost se pokazala kao ključna karakteristika modernih veb aplikacija, zato što omogućava širem spektru korisnika da pristupe i koriste aplikaciju bez obzira na njihove uređaje i okruženja.

2.6 Arhitektura veb aplikacije

Arhitektura veb aplikacije predstavlja internu strukturu i organizaciju komponenti aplikacije, i obuhvata kako fizičke, tako i logičke slojeve koji zajedno omogućavaju njeno efikasno funkcionisanje.



Slika 10 - Fizički slojevi veb aplikacije

Fizički slojevi aplikacije se odnose na stvarne, fizičke komponente i infrastrukturu potrebnu za funkcionisanje veb aplikacije i uključuju korisničke uređaje, veb servere, servere baze podataka i infrastrukturne komponente, slika 10. U ovu grupu se ubrajaju:

- *Korisnički uređaji.* Korisnici pristupaju veb aplikaciji putem različitih uređaja, uključujući desktop računare, laptop, tablete i pametne telefone, gde pomoću veb pretraživača šalju zahteve i primaju odgovore od veb servera.
- *Veb serveri* su fizički ili virtuelni računari koji hostuju veb aplikaciju. Oni obrađuju primljene HTTP/HTTPS zahteve, izvršavaju aplikacijski kod, pristupaju podacima i šalju odgovore nazad korisnicima.
- *Baze podataka* skladište podatke koje veb aplikacija koristi. Moguće su različite konfiguracije, gde server baze podataka može fizički da se nalazi na istoj mašini na kojoj je i veb server, na zasebnom računaru, u oblaku i sl.
- *Aplikacijski serveri* u složenijim arhitekturama obrađuju poslovnu logiku i upravljaju interakcijama između veb servera i baza podataka. Mogu da budu deo iste mašine kao veb server ili distribuirani u skladu sa potrebama projekta.
- *Infrastruktura u oblaku* omogućava skaliranje resursa u skladu sa konkretnim potrebama projekta.

Logički slojevi veb aplikacije definišu kako su funkcionalnosti aplikacije međusobno organizovane i povezane:

- *Prezentacioni Sloj* (klijentska strana, engl. *Frontend*). Ovaj sloj je odgovoran za interfejs aplikacije sa korisnicima. Sadrži HTML, CSS, i JavaScript kod koji se koristi za prikazivanje sadržaja i omogućavanje interakcije korisnika sa aplikacijom. Frontend komponente uključuju forme, dugmadi, i druge vizuelne elemente.
- *Poslovna logika* (serverska strana, engl. *Backend*). Poslovna logika se bavi obradom podataka i izvršavanjem poslovnih pravila aplikacije. Ovaj sloj obuhvata server-side skripte i kod koji upravlja

funkcionalnostima aplikacije kao što su korisničke sesije, autentifikacija, i manipulacija podacima.
- *Sloj Podataka* je odgovoran za pristup i manipulaciju podacima u bazama podataka. Ovaj sloj uključuje upite, transakcije, i druge operacije koje omogućavaju aplikaciji da čuva i preuzima podatke prema potrebama korisnika.

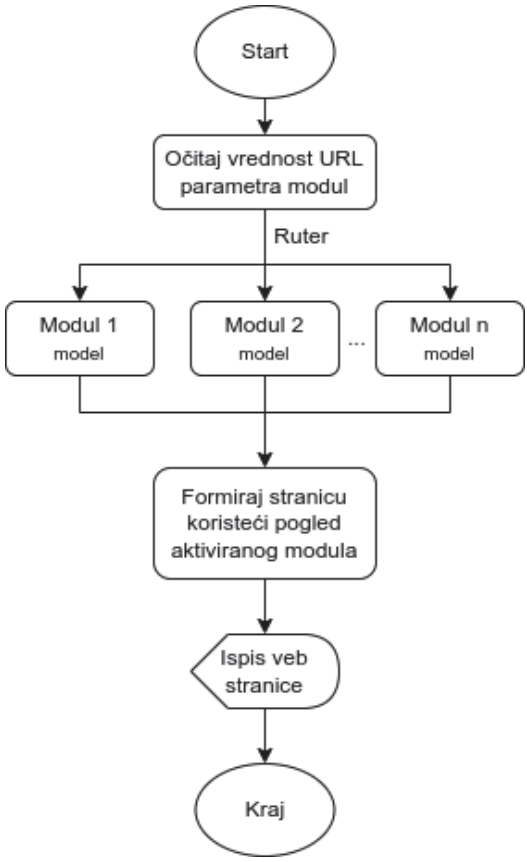
Ovakav pristup omogućava razvoj fleksibilnih aplikacija, čiji se delovi zajedno ili nezavisno mogu iznova koristiti u različitim projektima. Izmene u jednom sloju su izolovane u odnosu na ostale slojeve, dok je protok informacija između slojeva deo arhitekture zasnovan na jednom ili više protokola. Međutim, sa porastom broja slojeva struktura aplikacije postaje sve složenija, što otežava praćenje protoka podataka kroz slojeve.

2.6.1 MVC arhitektura

MVC arhitektura, koju čine *model*, *pogled* i *kontroler* (engl. *Model-View-Controller*, *MVC*), slika 11, omogućava međusobno razdvajanje delova koda koji se odnose na poslovnu logiku na serverskoj strani (model), rezultat zahteva koji je vidljiv na klijentskoj strani u veb brauzeru (pogled) i prepoznavanje, tumačenje i prosleđivanje zahteva odgovarajućim segmentima aplikacije (kontroler).

Klasičan MVC pristup smešta svu logiku aplikacije na veb server, dok tehnologije kao što su *Ajax*, *AngularJS*, *EmberJS*, *JavaScriptMVC*, *Backbone* i sl. omogućavaju da se MVC komponente delimično izvršavaju i na strani klijenta.

Primenom MVC-a se pruža mogućnosti prikaza modela na različite načine, a olakšani su procesi dodavanja novih ili izmene postojećih izveštaja, izmene interakcije sa korisnikom kao i izmene poslovne logike.



Slika 11 - MVC arhitektura veb aplikacije

2.6.2 Modularnost

Jednako važan segment arhitekture veb aplikacije predstavlja modularnost, odnosno pristup u dizajnu softverskog sistema gde se aplikacija deli na manje, nezavisne i samostalne komponente, tzv. module. Svaki modul ima jasno definisanu svrhu i funkcionalnost, što pojednostavljuje razvoj i preglednost koda, koji postaje organizovaniji i lakši za razumevanje.

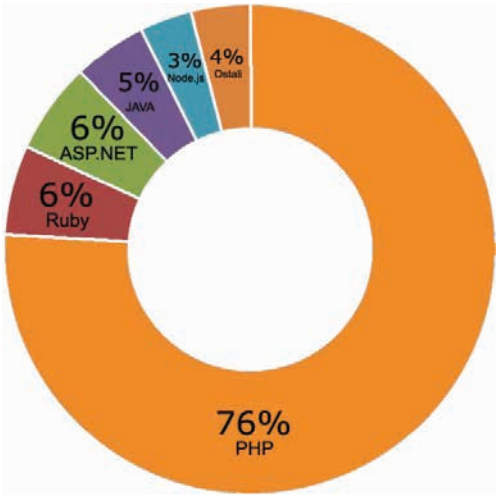
Moduli formirani za specifične funkcionalnosti mogu da se ponovo koriste u različitim delovima aplikacije ili čak u različitim projektima. Modularni dizajn omogućava različitim članovima tima da rade na različitim modulima istovremeno, što ubrzava proces razvoja i poboljšava saradnju članova tima

uz smanjenje rizika. Ako jedan modul uzrokuje probleme ili greške, ostali moduli mogu da funkcionišu neometano, što povećava stabilnost aplikacije.

Modularnost omogućava aplikaciji da se lako skalira dodavanjem novih modula ili proširivanjem postojećih, i omogućava razvijanje aplikacije u fazama. Ovakvim pristupom se smanjuju količina i redudencija koda, što doprinosi jednostavnijem održavanju aplikacije i efikasnijoj upotrebi raspoloživih resursa.

2.6.3 Programiranje na strani servera

Izbor jezika za razvoj veb aplikacije na strani servera zavisi od konkretnih projektnih zahteva, stručnosti tima i raspoloživih resursa, a ima ključnu ulogu u oblikovanju arhitekture, performansi i skalabilnosti veb aplikacije. U trenutno najpopularnije jezike za razvoj veb aplikacija na serverskoj strani mogu da se uvrste *PHP*, *Ruby*, *ASP.NET*, *Java* i *Node.js*.



Slika 12 - Tržišni udeo programskih jezika na strani servera
Avgust 2024

PHP je i dalje dominantan sa tržišnim udelom koji se zavisno od procene kreće između 70% i 80%, slika 12 [3, 4], i to najviše zahvaljujući popularnim sistemima za upravljanje sadržaja (engl. *Content Management Sistem*, CMS) kao što su *Wordpress* i *Drupal*, i aplikacijama za online prodavnice

kao što su *Magento*, *PrestaShop*, *OpenCart* i *WooCommerce*. *Node.js* je u brzom porastu i sve više se koristi za savremene aplikacije koje zahtevaju visoke performanse i fleksibilnost.

Tabela 2 - Kompanije koje koriste PHP programski jezik

Kompanija	Opis
Facebook	Facebook koristi PHP kao deo svoje tehnologije, iako su razvili *HipHop* Virtual Machine (HHVM) za optimizaciju PHP-a.
Wikipedia	Wikipedia koristi PHP za upravljanje sadržajem preko *MediaWiki* platforme.
WordPress	Najpopularniji CMS razvijen u PHP-u, koristi ga veliki broj veb sajtova za upravljanje sadržajem.
Yahoo	Yahoo koristi PHP za različite veb servise i platforme
Tumblr	Tumblr koristi PHP za delove sistema za upravljanje sadržajem.
Slack	Koristi PHP u kombinaciji sa drugim tehnologijama za razvoj svojih komunikacijskih usluga.
Etsy	Koristi PHP za upravljanje tržištem i interakciju sa korisnicima na svojoj platformi.
Mailchimp	Koristi PHP za deo svoje infrastrukture i aplikacija za e-mail marketing.
Spotify	Koristi PHP za razne veb aplikacije i alate, posebno za administrativne funkcionalnosti.
Drupal	Još jedan popularan CMS, koristi PHP kao osnovni jezik za razvoj i upravljanje sadržajem.

Jednostavnosti i pristupačnosti za početnike, kao i široka podrška i kompatibilnosti sa različitim operativnim sistemima i hosting uslugama su mu doneli globalnu popularnost, a dugi niz godina je bio osnovni pokretač razvoja dinamičkih veb aplikacija. Razvijen ekosistem uključuje brojne ekstenzije, biblioteke i pre svega radne okvire kao što su *Laravel* i *Symfony*, koji nude napredne funkcije i alate za pojednostavljenje razvoja.

Mogućnost integracije sa različitim bazama podataka, kao i fleksibilnost u kombinovanju dinamičkog i statičkog sadržaja doprinose njegovoj svestranosti. Osim toga, PHP ima veliku i aktivnu zajednicu koja pruža resurse, podršku i neprekidan razvoj savremenih alata, čineći ga ključnim jezikom za veb razvoj.

Ruby, posebno sa radnim okvirom *Ruby on Rails*, poznat je po svojoj elegantnoj sintaksi i prijateljskom okruženju za programere. Fokus je na

jednostavnosti i produktivnosti, što ga čini privlačnim izborom kada je prioritet na čistom i intuitivnom kodiranju.

C# je razvijen od strane kompanije *Microsoft* i obično se koristi sa *.NET* radnim okvirom. Integracija sa *Microsoft* tehnologijama i obimne biblioteke ga čine korektnim izborom za izgradnju aplikacija i servisa na nivou preduzeća. Sa radnim okvirima kao što je *ASP.NET Core*, C# omogućava programerima da kreiraju veb aplikacije koje mogu da funkcionišu na različitim platformama.

Java ima dugogodišnju pozitivnu reputaciju u korporativnom svetu. Kao statički tipizirani, objektno-orijentisani jezik, Java je dizajnirana da se nosi sa velikim aplikacijama i složenim sistemima. Radni okviri kao što je Spring Boot pojednostavljuju proces razvoja pružajući sveobuhvatan set alata i biblioteka za izgradnju aplikacija. Snažno tipiziranje i razvijen ekosistem čine je dobrim izborom za rešenja na nivou preduzeća koja zahtevaju pouzdanost i performanse.

*Node.*js je omogućio da *JavaScript*, inače tradicionalno poznat kao jezik za programiranje na klijentskoj strani, pronađe novu ulogu i na serverskoj strani, čime se stvara jedinstveno jezičko okruženje preko celog steka. Pogodan je za izgradnju skalabilnih aplikacija koje funkcionišu u realnom vremenu.

Python je poznat po svojoj jednostavnosti i čitljivosti. Njegova jasna sintaksa omogućava brzo i efikasno pisanje koda. Dva popularna radna okvira su napisana u ovom programskom jeziku, *Django* koji podstiče brzi razvoj i čist, pragmatičan dizajn, i *Flask* koji nudi lakši i fleksibilniji izbor za programere koji preferiraju da sami izgrade komponente.

2.6.4 Programiranje na klijentskoj strani

Tehnologije koje se koriste na klijentskoj strani veb aplikacije se koriste za formiranje korisničkog interfejsa i kontrolu njegove interaktivnosti. HTML, CSS i JavaScript čine osnovu za strukturiranje, stilizovanje i kontrolu elemenata veb stranica. *Javascript* radni okviri i biblioteke kao što su *React*,

Angular i *Vue.js* dodatno unapređuju razvoj omogućavajući bržu izgradnju složenih korisničkih interfejsa i lakše upravljanje podacima.

Tehnike kao što je *AJAX* omogućavaj asinhrono učitavanje podataka, čime se obezbeđuje brža i efikasnija interakcija sa veb aplikacijom. *WebAssembly* omogućava izvršavanja složenog koda direktno u pretraživaču brzinom uporedivom sa desktop aplikacijama.

Konačno, *Web API*-ji igraju vitalnu ulogu u integraciji veb aplikacija sa spoljnim servisima, proširujući njihove mogućnosti i omogućavajući pristup različitim podacima i funkcionalnostima.

Zajedno, ove tehnologije omogućavaju stvaranje veb aplikacija koje su po brzini izvršavanja i kvalitetu korisničkog iskustva u potpunosti ravnopravne desktop aplikacijama.

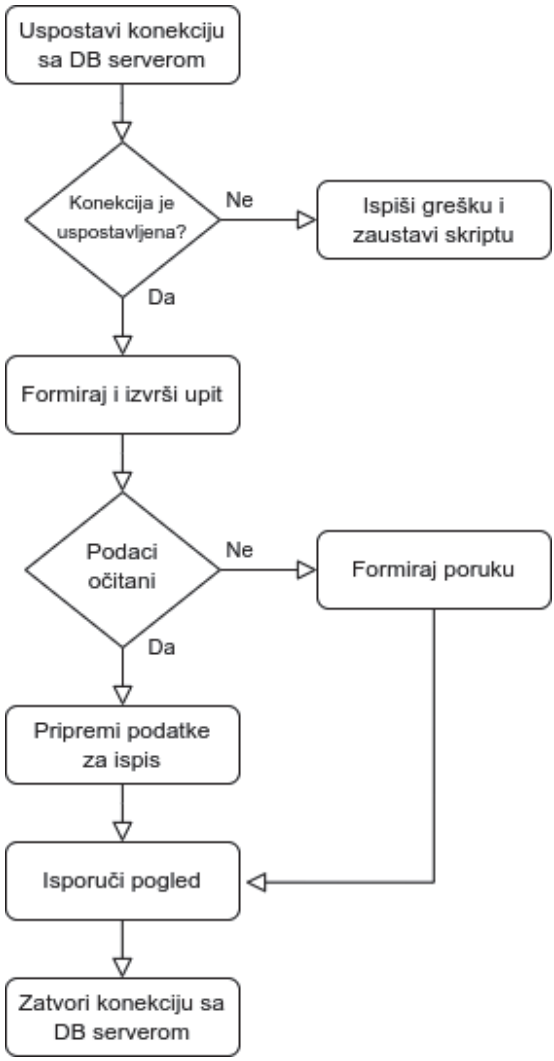
2.6.5 Integracija sa bazom podataka

Veb aplikacije najčešće koriste relacione baze podataka za skladištenje, upravljanje i pristup podacima, čime se obezbeđuje efikasno i organizovano čuvanje velikih količina informacija, kao i mogućnost brze pretrage i manipulacije podacima. Integracija veb aplikacije sa bazom podataka omogućava dinamičko upravljanje, i ovaj proces obuhvata povezivanje aplikacije sa bazom podataka, izvršavanje upita, i na kraju prekid konekcije aplikacije sa bazom podataka, slika 13.

Prvi korak u integraciji je izbor odgovarajućeg servera baze podataka, u skladu sa potrebama aplikacije. Baze podataka mogu biti *relacione baze* kao što su *MySQL*, *PostgreSQL* ili *Microsoft SQL Server*, i nerelacione baze podataka poput *MongoDB* ili *Apache Cassandra*. U sledećem koraku, baza podataka se instalira na serveru ili u klaudu, i konfiguriše u skladu sa parametrima projekta. U poslednjem koraku se formiraju tabele, odnosno struktura podataka i vrši se ubacivanje podataka u bazu.

Veb aplikacija se povezuje sa bazom podataka putem odgovarajuće ekstenzije ili drajvera, a svaka baza podataka ima specifične drajvere koji podržavaju različite programske jezike. Na primer, *PHP* koristi *MySQLi* ili

PDO ekstenzije za rad sa *MySQL* serverom, dok *PostgreSQL* koristi *pg-promise* za *Node.js*.



Slika 13 - Redosled koraka u radu aplikacije sa bazom podataka

ORM (engl. *Object-Relational Mapping*) je tehnologija koja omogućava programerima da komuniciraju sa relacionim bazama podataka koristeći objekte i klase samog programskog jezika, umesto direktnog pisanja SQL upita. ORM vrši automatsko mapiranje baze podataka, pojednostavljujući

rad sa podacima kroz automatsko generisanje SQL upita i upravljanje relacijama između objekata. Ovo ne samo da ubrzava razvoj aplikacija i smanjuje potrebu za ručnim pisanjem SQL upita, već i poboljšava sigurnost kroz zaštitu od SQL injekcija i validaciju podataka.

Veb aplikacije mogu da koriste i API-je, kao što su *RESTful* i *GraphQL* za komunikaciju sa bazom podataka. Ovo omogućava lakšu integraciju, posebno u distribuiranim sistemima i mikroservisnim arhitekturama.

MySQL je jedna od najpopularnijih relacionih baza podataka otvorenog koda. Poznata je po svojoj brzini, stabilnosti i širokoj podršci zajednice. *MariaDB* je takođe baza podataka otvorenog koda, koja je razvijena kao fork *MySQL*-a, sa ciljem da se obezbedi slobodan i otvoren alternativni razvojni put za *MySQL*. U velikoj meri su međusobno kompatibilni.

NoSQL baze podataka ne prate striktne relacione modele podataka, a dizajnirane su da bolje odgovore na potrebe za skalabilnošću i fleksibilnošću u radom sa velikim količinama podataka koji mogu biti nestrukturirani ili polu-strukturirani. *MongoDB* i *Apache Cassandra* spadaju među najpopularnije *NoSQL* baze podataka.

Izbor tipa baze podataka zavisi od količine i strukture podataka sa kojima će IS raditi. Relacione baze podataka su optimalno rešenje kada je su potrebne složena veza između podataka, transakcije i visoka konsistentnost, dok se NoSQL baze podataka koriste pri radu sa velikim količinama neuređenih podataka i kada je potrebna horizontalna skalabilnost. U mnogim slučajevima, aplikacije mogu da koriste oba tipa baza podataka, gde će svaka služiti specifičnim potrebama u okviru iste aplikacije.

PHP programski jezik poseduje ekstenzije za rad sa različitim vrstama baza podataka, pri čemu se najčešće koriste MySQLi (engl. *MySQL improved*) i *PDO* (engl. *PHP Data Object*). *MySQLi* nudi proceduralni i objektno orijentisani pristup, podržava više upita i upravljanje transakcijama. Sa druge strane, *PDO* nudi isključivo objektno orijentisani pristup, nema

podršku za višestruke upite, ali omogućava rad sa različitim serverima baze podataka.

MySQLi i *PDO* pružaju podršku za pripremljene upite, koji predstavljaju tehniku koja unapređuje sigurnost i performanse *SQL* upita. Ovaj proces uključuje dva koraka, u kojima se prvo upit priprema na serveru baze podataka sa označenim mestima za parametre, a zatim se izvršava sa konkretnim vrednostima za te parametre. Ova metoda štiti od SQL injekcija, zato što se parametri obrađuju odvojeno od SQL koda, što sprečava napadače da injektuju maliciozan kod.

2.7 Kukiji

Kuki (engl. *cookie*) je mali dokument, odnosno fajl, koji se čuva na korisnikovom računaru ili uređaju kada poseti veb sajt. Služi za skladištenje informacija koje potom veb aplikacija može da koristi za personalizaciju korisničkog iskustva, kao što su izbor teme, id sesije, podešavanja u obrascu za pretragu, javno dostupni podaci o korisničkom nalogu i sl. Kuki omogućava veb aplikaciji da *zapamti* korisnika između različitih sesija i pružaj kontinuitet u interakciji s korisnikom.

Tabela 3 - PHP funkcije za rad sa kukijima

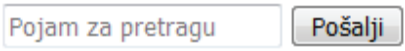
Naziv	Opis
$_COOKIE	Superglobalna promenljiva tipa niz koja sadrži podatke o kukiju poslate sa klijentske strane
setcookie()	Postavlja novi kuki ili ažurira postojeći. Mora biti pozvana pre slanja bilo kakvog HTML sadržaja.
setrawcookie()	Postavlja novi kuki ili ažurira postojeći, ali ne enkodira vrednost kukija.
session_set_cookie_params()	Postavlja parametre za kuki sesije, kao što su trajanje, putanja i domen, mora biti pozvana pre funkcije *session_start()*.
session_get_cookie_params()	Vraća trenutne parametre kukija sesije kao asocijativni niz.

Svaki kuki karakterišu vreme trajanja, putanja i domen na kome je validan. *Vreme trajanja* određuje koliko dugo će kuki važiti pre isteka. Vreme može biti postavljeno na određeni datum ili na osnovu broja sekundi od trenutka

postavljanja. *Putanja* definiše koje URL putanje će imati pristup kukiju, dok *domen* definiše na kom domenu i poddomenima kuki može biti dostupan, čime se kontroliše njegov opseg upotrebe. U tabeli 3 je dat kratak pregled PHP funkcija za rad sa kukijima.

2.8 Komunikacija sa korisnikom

Veb obrazac je komponenta veb stranice koja omogućava korisnicima da unose podatke koji se zatim mogu poslati serveru za obradu. Imaju raznovrsnu namenu, kao što su obrasci za registraciju korisnika, filtriranje sadržaja, pretragu informacija, slanje komentara ili naručivanje robe. Na slici 14 je prikazan jednostavan veb obrazac za pretragu sadržaja veb sajta, dok je na listingu 1 prikazan pratećih HTML kod.



Slika 14 - Veb obrazac, pretraga sadržaja

```
<form>
  <input type="text" name="pojam" placeholder="Pojam">
  <button type="submit">Pošalji</button>
</form>
```

Listing 1 - Veb obrazac, HTML kod

Veb obrazac se formira primenom HTML taga *form*, unutar koga se potom ugnježđuju potrebni input i kontrolni elementi koji omogućavaju korisnicima da unesu podatke. Opcioni parametri *action* i *method* definišu fajl koji će biti aktiviran prilikom slanja podataka i metodu kojom će podaci biti poslati. Ukoliko parametar *method* nije naveden, kao na listingu 1

Uneti podaci mogu da se pošalju ka veb serveru metodama *GET* i *POST*, pri čemu svaka ima svoje specifične karakteristike, pa se zbog toga koriste u različitim situacijama.

GET metoda podatke šalje kao URL parametre (engl. *query parameters*). Ukoliko se u obrazac sa slike 14 unese pojam *ispitni rok*, URL će biti formiran na sledeći način:

```
https://www.singidunum.ac.rs/pretraga.php?pojam=ispitni%20rok
```

Podaci su vidljivi u URL-u, takva adresa može biti sačuvana kao *bookmark*, i zato je pogodna za primenu u obrascima za pretragu, filtriranje podataka i upravljanje aplikacijom, dok se za slanje podataka koji treba da budu uneti u bazu podataka, ili slanje poverljivih podataka ne koristi. Drugi ograničavajući faktor jeste ukupan broj karaktera u URL-u, koji utiče na količinu podataka koja može da se pošalje preko veb obrasca. Iako ovo ograničenje zavisi od pretraživača i servera, u praksi je obično oko 2000 karaktera.

POST metoda podatke šalje u telu HTTP zahteva. Ovo omogućava slanje većih količina podataka, koji nisu javno prikazani kao u slučaju *GET metode*. U praksi, konfiguracije veb servera i aplikacije utiču na količinu podataka koji se mogu obraditi. Metoda se koristi kada se podaci šalju serveru da bi se stvorio novi resurs ili izmenio postojeći.

Na strani servera, PHP očitava vrednosti poslate putem veb obrasca i smešta ih u superglobalne promenljive *$_GET* i *$_POST* u skladu sa metodom kojom su podaci poslati. Podaci dobijeni na ovakav način moraju prvo da se provere po tipu i sadržaju, odnosno da se izvrši njihova sanitacija i validacija, posle čega mogu bezbedno da se obrade i ubace u bazu podataka.

2.9 Rutiranje

Rutiranje je ključna komponenta za organizaciju i upravljanje sadržajem veb aplikacije i odnosi se na proces usmeravanja korisničkih zahteva ili URL-ova ka odgovarajućim delovima aplikacije. Na taj način se jasno razdvajaju različiti delovi aplikacije kroz čitljiviji kod, čime se olakšava i proces održavanja.

URL adrese se koriste za rutiranje u veb aplikacijama jer omogućavaju jednostavnu navigaciju između različitih stranica aplikacije, a korisnici mogu da sačuvaju željene stranice za kasniji pristup (engl. *bookmarking*). Takođe, ovakav pristup olakšava deljenje sadržaja putem veb linkova.

Efikasan pristup u formiranju arhitekture aplikacije obično podrazumeva da se fajl *index.php* koristi kao centralni kontroler za rukovanje svim URL-ovima, a komande se aplikaciji saopštavaju putem URL parametara, tabela 4. Vrednosti parametara potom prolaze kroz *switch* izjavu gde se odlučuje koji fajl, odnosno modul će se dalje uključiti u tok izvršavanja aplikacije, slika 13.

Tabela 4 - Upotreba URL parametara za rutiranje

Listing studenata upisanih 2018. godine
https://www.singidunum.ac.rs/studenti.php?generacija=2018
Listing studenata upisanih 2018. godine, sortiraj po prezimenu
https://www.singidunum.ac.rs/studenti.php?generacija=2018&sort=prezime
Početna strana profila studenta čiji je broj indeksa 2018123456
https://www.singidunum.ac.rs/studenti.php?indeks=2018123456
Profil studenta čiji je broj indeksa 2018123456, tab sa ličnim podacima
https://www.singidunum.ac.rs/studenti.php?indeks=2018123456&tab=licni-podaci

Po potrebi, URL parametri mogu da se maskiraju primenom čistih URL-a (engl. *user-friendly URLs* ili *pretty URLs*), tabela 5, ali je u tom slučaju neophodna upotreba *.htaccess* fajla na veb serveru, koji omogućava prilagođeno upravljanje URL preusmeravanjima, pristupom i drugim postavkama za direktorijume. U kontekstu rutiranja, koristi se za preusmeravanje svih zahteva na glavni *index.php* fajl radi centralizovanog upravljanja aplikacijom, listing 2.

Tabela 5 - Upotreba čistih URL-a za rutiranje

Listing studenata upisanih 2018. godine
https://www.singidunum.ac.rs/studenti/2018
Listing studenata upisanih 2018. godine, sortiraj po prezimenu
https://www.singidunum.ac.rs/studenti/2018/sort/prezime
Početna strana profila studenta čiji je broj indeksa 2018123456
https://www.singidunum.ac.rs/studenti/student/2018123456
Profil studenta čiji je broj indeksa 2018123456, tab sa ličnim podacima
https://www.singidunum.ac.rs/studenti/student/2018123456/licni-podaci

```
1  RewriteEngine On
2  RewriteCond %{REQUEST_FILENAME} !-f
3  RewriteCond %{REQUEST_FILENAME} !-d
4  RewriteRule ^(.*)$ index.php?url=$1 [QSA,L]
```

Listing 2 - Apache .htaccess preusmeravanje dolaznog saobraćaja na *index.php*

2.10 Upravljanje HTTP zahtevima i odgovorima

Veb aplikacija može da upravlja HTTP zahtevima i odgovorima na strani servera, što dalje omogućava prilagođavanje korisničkog iskustva, kontrolu bezbednosti i validaciju zahteva, upravljanje keširanjem radi boljih performansi, kao i precizno rukovanje redirekcijama i greškama. Podešavanjem HTTP zaglavlja i statusnih kodova, server može da pruži dinamične odgovore uz optimizaciju rada aplikacije i lakšu dijagnostiku.

Tabela 6 - PHP funkcije za rad sa zaglavljima

Ime funkcije	Opis
Čitanje zaglavlja	
getallheaders()	Vraća sva HTTP zaglavlja kao asocijativni niz. Može biti onemogućena na nekim serverima.
apache_request_headers()	Vraća HTTP zaglavlja zahteva kao asocijativni niz. Specifično za Apache servere.
Manipulacija zaglavljima	
header()	Postavlja HTTP zaglavlja koja se šalju klijentu. Koristi se za preusmeravanje, kontrolu keširanja itd.
header_remove()	Uklanja zaglavlje sa liste zaglavlja koja će biti poslana.
headers_list()	Vraća niz svih zaglavlja koja su trenutno zakazana za slanje klijentu.
Podešavanje statusa odgovora	
http_response_code()	Postavlja ili vraća trenutni HTTP statusni kod. Koristi se za eksplicitno postavljanje statusnog koda.
headers_sent()	Vraća true ako su zaglavlja već poslata, ili false ako nisu. Korisno za dijagnostiku.

Prvi korak u ovom procesu je čitanje zaglavlja zahteva, a PHP ima nekoliko ugrađenih funkcija koje mogu da se iskoriste za ovu svrhu, tabela 6. Ukoliko je potreban pristup specifičnim zaglavljima, može se iskoristiti PHP superglobalna promenljiva *$_SERVER* koja sadrži ove informacije.

```php
<?php
header("Content-type:application/pdf");
header("Content-Disposition:attachment;"
       "filename='knjiga.pdf'");
readfile("knjiga-original.pdf");
?>
```

Listing 3 - PHP manipulacija zaglavljem, slanje pdf dokumenta

Manipulacija zaglavljima je sledeći važan deo upravljanja HTTP komunikacijom. Na ovaj način moguće je da se izvrši preusmeravanje korisnika na drugu stranicu, ili postavljanje HTTP zaglavlja koja kontrolišu keširanje. Na listingu 3 je prikazan primer upravljanja zaglavljem radi slanja pdf dokumenta. Treba napomenuti da se zaglavlja moraju poslati pre nego što se klijentu pošalje bilo koji drugi deo HTML sadržaja ili izlaza. U suprotnom, PHP će generisati grešku zato što ne može da menja zaglavlja nakon što je sadržaj već poslat.

```php
<?php
header("Status: 404 Not Found");
?>
```

Listing 4 - PHP slanje zaglavlja sa statusnim kodom 404

Statusni kodovi HTTP odgovora obaveštavaju klijenta o rezultatu zahteva. PHP omogućava slanje statusnih kodova kao što je 404, listing 4, što znači da tražena stranica ne postoji. Precizno upravljanje statusnim kodovima omogućava slanje klijentu tačne informacije o stanju aplikacije i kontrolu ponašanja veb brauzera.

2.11 Serijalizacija podataka

Serijalizacija je proces pretvaranja objekta ili složene strukture podataka, listing 5, u format koji može da se lako sačuva ili prenese, kao što su stringovi, *JSON* ili *XML*.

```php
$student = [
    'prezime' => 'Đorđević',
    'ime' => 'Đorđe',
    'broj_indeksa' => '2021201123',
    'godine_upisa' => [
        2021 => 1,
        2022 => 2,
        2023 => 2,
        2024 => 3
    ]
];
```

Listing 5 - Podaci u izvornom obliku, PHP niz

```
a:4:{s:7:"prezime";s:11:"Đorđević";s:3:"ime";s:7:"Đorđe";
s:12:"broj_indeksa";s:10:"2021201123";s:12:"godine_upisa";
a:4:{i:2021;i:1;i:2022;i:2;i:2023;i:2;i:2024;i:3;}}
```

Slika 15 - Serijalizovani podaci

Ova tehnika omogućava skladištenje podataka u fajlovima ili bazama podataka i olakšava prenos podataka između servera i klijenta ili između različitih sistema. Serijalizacija je posebno korisna u scenarijima kao što su rad sa velikim količinama podataka, komunikacija između različitih aplikacija ili sistema, i čuvanje stanja objekata u dugotrajnom skladištu.

Deserijalizacija je suprotan proces koji vraća serijalizovane podatke u izvorni oblik, radi obrade ili manipulacije. Koristi se prilikom prenosa podataka između različitih aplikacija, kada se obnavlja stanje objekata iz baze podataka, ili kada se obrađuju podaci primljeni putem API poziva.

PHP poseduje ugrađene funkcije koje se koriste za ovu svrhu, *serialize()* i *unserialize()*. Važno je da se napomene da deserijalizacija predstavlja potencijalni sigurnosni rizik ako se podaci ne validiraju pravilno.

2.12 Rad sa fajlovima

Mehanizmi za kreiranje, preuzimanje i upload fajlova u veb aplikaciji su slični u većini programskih jezika, uključujući PHP, Python i Javu. Ovi mehanizmi obuhvataju procese kao što su kreiranje novih fajlova i direktorijuma, preuzimanje sadržaja sa udaljenih servera ili resursa i *upload* fajlova sa korisničkih uređaja na server. U osnovi, obavljaju se pomoću ugrađenih funkcija za manipulaciju fajlovima, bez obzira na programski jezik koji se koristi, tabela 7 za PHP, dok se tzv. *strimovanje* (engl. *streaming*) može odnositi na nekoliko različitih operacija.

Tabela 7 - PHP funkcije za rad sa fajlovima i direktorijumima

Ime funkcije	Opis
Rad sa fajlovima na disku	
rename()	Menja ime fajla ili premešta fajl.
unlink()	Briše fajl.
Rad sa direktorijumima	
mkdir()	Kreira novi direktorijum.
rmdir()	Briše direktorijum.
opendir()	Otvara direktorijum za čitanje.
readdir()	Čita sledeći fajl ili direktorijum u otvorenom direktorijumu.
closedir()	Zatvara otvoreni direktorijum.
Manipulacija dozvolama i vlasništvom	
chmod()	Menja na nivou servera dozvole za pristup fajlu ili direktorijumu.
chown()	Menja na nivou servera vlasnika fajla ili direktorijuma.
chgrp()	Menja na nivou servera grupu fajla ili direktorijuma.
Rad sa putanjama	
pathinfo()	Vraća kao niz informacije o putanji fajla.
Informacije o fajlovima	
fstat()	Dobija informacije o fajlu u obliku niza.
file_exists()	Proverava da li fajl ili direktorijum postoji.
filesize()	Vraća veličinu fajla u bajtovima.
is_file()	Proverava da li je navedena putanja fajl.
is_dir()	Proverava da li je navedena putanja direktorijum.

2.12.1 Preuzimanje (*upload*) fajlova sa klijentske strane

Kada se preuzima dokument sa klijentske strane (engl. *document upload*), neophodno je da se prvo proveri da li je proces obavljen bez greške. Potom se proveravaju tip i veličina dokumenta, a u poslednjem koraku, pre postavljanja na trajnu lokaciju, mašinski se generiše ime dokumenta, čime se izbegavaju potencijalni sigurnosni propusti, npr. injekcija skripte na server, ili mogućnost postavljanja fajla preko postojećeg dokumenta sa istim imenom.

2.12.2 Manipulacija fajlovima na serverskoj strani

PHP funkcija *file_get_contents*, tabela 8, se koristi za čitanje celokupnog sadržaja fajla i njegovo smeštanje u promenljivu tipa *string*, što je korisno za jednostavno preuzimanje podataka iz fajlova ili sa URL adresa. Nasuprot tome, funkcija *file_put_contents* omogućava pisanje podataka u fajl. Ukoliko fajl sa navedinim imenom ne postoji, ona će ga automatski kreirati, ili će postojeći fajl zameniti novim, osim ukoliko nije podešen atribut za dodavanje sadržaja na kraj fajla. Oba mehanizma omogućavaju lako i brzo rukovanje fajlovima, pružajući jednostavne metode za čitanje i zapisivanje podataka, bez obzira na to da li se radi o lokalnim fajlovima ili udaljenim resursima, listing 6.

Tabela 8 - PHP funkcije za rad sa fajlovima, čitanje i upis podataka

Ime funkcije	Opis
Rad sa sadržajem fajlova	
fopen()	Otvara fajl ili URL za čitanje ili pisanje.
fclose()	Zatvara otvoreni fajl.
fread()	Čita sadržaj fajla u određenom broju bajtova.
fwrite()	Upisuje podatke u otvoreni fajl.
fgets()	Čita liniju iz fajla.
fgetcsv()	Čita liniju iz fajla i parsira je kao CSV
file_get_contents()	Čita sadržaj fajla u string.
file_put_contents()	Upisuje podatke u fajl. Ukoliko fajl ne postoji, biće kreiran.

```php
<?php

$url = "https://repository.singidunum.ac.rs/" .
       "ispitni_rokovi/redovni/2024/tf-sii-sep2-2024.pdf";
$raspored = file_get_contents($url);

if ($contents !== false) {
    file_put_contents(
            "./raspored/tf-sii-sep2-2024.pdf",
            $raspored
        );
    echo "Fajl je uspešno preuzet.";
} else {
    echo "Ne mogu da preuzmem fajl.";
}

?>
```

Listing 6 - PHP reuzimanje sadržaja lokalnog fajla ili sa URL adrese

Tabela 9 - PHP funkcije za rad sa FTP serverom

Funkcija	Opis
ftp_connect	Usmerava PHP da se poveže na FTP server.
ftp_login	Prijavljuje se na FTP server koristeći korisničko ime i lozinku.
ftp_get	Preuzima fajl sa FTP servera na lokalni sistem.
ftp_put	Postavlja lokalni fajl na FTP server.
ftp_delete	Briše fajl sa FTP servera.
ftp_mkdir	Kreira novi direktorijum na FTP serveru.
ftp_rmdir	Briše direktorijum sa FTP servera.
ftp_chdir	Menja trenutni radni direktorijum na FTP serveru.
ftp_pwd	Vraća trenutni radni direktorijum sa FTP servera.
ftp_nlist	Vraća listu fajlova i direktorijuma u trenutnom direktorijumu.
ftp_size	Vraća veličinu fajla na FTP serveru.
ftp_rename	Preimenuje fajl ili direktorijum na FTP serveru.
ftp_raw	Izvršava raw FTP komande na serveru.
ftp_rawlist	Vraća raw listu fajlova i direktorijuma, uključujući detaljne informacije.
ftp_close	Zatvara FTP vezu.

```php
<?php

$ftp_server = "ftp.example.com";
$ftp_user_name = "ime";
$ftp_user_pass = "lozinka";
$remote_file = "/path/to/remote/fajl.txt";
$local_file = "lokalna_putanja/fajl.txt";

// Postavljanje FTP konekcije
$conn_id = ftp_connect($ftp_server);

if ($conn_id) {
    // Prijava na FTP server
    $login_result = ftp_login(
                        $conn_id,
                        $ftp_user_name,
                        $ftp_user_pass
                    );
    if ($login_result) {
        // Preuzimanje fajla
        if (ftp_get(
                $conn_id,
                $local_file,
                $remote_file,
                FTP_BINARY)
            ) {
              echo "Fajl je uspešno preuzet.";
        } else {
            echo "Ne mogu preuzeti fajl.";
        }
    } else {
        echo "Ne mogu da se prijavim na FTP server.";
    }
    // Zatvori FTP konekciju
    ftp_close($conn_id);
} else {
    echo "Ne mogu da se povežem sa FTP serverom.";
}

?>
```

Listing 7 - FTP preuzimanje fajla

Uprkos jednostavnosti, ovakav pristup ima ozbiljna ograničenja, pre svega prilikom rada sa velikim datotekama zato što učitava ceo sadržaj u memoriju, što može da izazove probleme sa performansama ili prekid rada aplikacije. Kada se koristi za čitanje podataka sa udaljenih adresa, mogući su problemi sa HTTPS sertifikatom ili sa zahtevima za autentifikaciju. Pored toga, *file_get_contents* ne podržava direktno čitanje delova datoteke ili *strim* podataka, pa su za takve namene bolje opcije funkcije poput *fopen*, *fread* i *fgets* koje pružaju veću kontrolu nad procesom čitanja.

Upotreba FTP-a za preuzimanje fajlova u PHP-u je moguća primenom ugrađenih funkcija, tabela 9 i listing 7, ukoliko potrebe projekta nalažu takav pristup.

2.12.3 Strimovanje fajlova

Strimovanje predstavlja proces prenosa podataka iz fajla u delovima, tzv. blokovima. Ovo je posebno korisno za rad sa velikim fajlovima koji mogu prelaziti kapacitet dostupne RAM memorije, čime se omogućava efikasnije upravljanje resursima i unapređenje performansi aplikacije.

Tokom strimovanja, podaci mogu da se obrađuju dok se čitaju, što je korisno u aplikacijama koje zahtevaju obradu podataka u realnom vremenu, kao što su video strimovi ili veb aplikacije koje obrađuju velike log fajlove.

Strimovanje može da poboljša efikasnost prenosa podataka kada se koristi u mrežnim okruženjima, kao što je preuzimanje ili učitavanje velikih datoteka sa interneta, jer omogućava korisnicima da započnu rad sa podacima pre nego što se ceo fajl preuzme.

U ovu svrhu u PHP-u mogu da se koriste ugrađene funkcije *fopen*, *fread* i *fgets*.

2.13 Asihorni pozivi

Asinhroni pozivi u veb aplikacijama omogućavaju slanje i primanje podataka sa servera, bez potrebe za ponovnim učitavanjem cele veb stranice. Ova tehnika koristi AJAX (engl. *Asynchronous JavaScript and*

XML), koji omogućava veb stranici da ažurira ili učitava podatke u pozadini dok korisnik nastavlja sa interakcijom.

Asinhroni pozivi mogu da se koriste u različitim situacijama, npr. kada je potrebno da se automatski učitaju novi podaci bez ponovnog učitavanja cele stranice, ili za validaciju podataka unetih u veb obrazac. Primenom ove tehnike veb stranica je interaktivnija i sa bržim odzivom, dok se istovremeno smanjuje opterećenje veb servera i mreže usled smanjene količine podataka koji se šalju i primaju.

Iako pružaju brojne prednosti, implementacija asinhronih poziva donosi i određene nedostatke:

- *Složenost arhitekture.* Implementacija asinhronih poziva može da poveća složenost koda. Upravljanje asinhronim operacijama, posebno kada ih ima mnogo, može da bude izazovno i dovede do problema kao što su prekomerno ugnježdeni povratni pozivi, što otežava praćenje tokova podataka (engl. *callback hell*).
- *Problemi sa sinhronizacijom.* Ako više asinhronih poziva zavise jedan od drugog, upravljanje njihovim redosledom i međusobna sinhronizacija može biti otežano i dovesti do problema ukoliko podaci nisu učitani ili obrađeni u očekivanom redosledu.
- *Povećano opterećenje na klijentskoj strani.* Asinhroni pozivi mogu dovesti do povećanog opterećenja na klijentskoj strani, naročito ako se koriste za često slanje zahteva, ili kada se mnogo podataka prenosi, što može da utiče na performanse veb stranice.
- *Bezbednosni izazovi.* Iako asinhroni pozivi pomažu u sprečavanju određenih vrsta napada kao što su CSRF (engl. *Cross-Site Request Forgery*), mogu da predstavljaju bezbednosni rizik ako nisu pravilno zaštićeni. Na primer, ukoliko nisu implementirani adekvatni mehanizmi za autentifikaciju i autorizaciju, javlja se rizik od neovlašćenog pristupa podacima.
- *Poteškoće u debagovanju.* Asinhroni kod može da oteža testiranje aplikacije i otkrivanje grešaka, zato što su greške u asinhronim pozivima teže uočljive.

XMLHttpRequest je *JavaScript* objekat koji se koristi za asihrono slanje i primanje *HTTP* zahteva. Iako je bio standard za asinhronu komunikaciju na vebu, danas se sve više koristi *Fetch API*, koji nudi jednostavniji i čitljiviji način za rad sa *HTTP* zahtevima.

Fetch API koristi tzv. *obećanja* (engl. *Promises*) za upravljanje asinhronim operacijama, što olakšava rad sa odgovorima i greškama, i pruža više mogućnosti za rad sa različitim formatima podataka. Za razliku od *XMLHttpRequest*-a, omogućava čistiji i fleksibilniji pristup za dodavanje opcija kao što su *HTTP* metode i zaglavlja. Takođe, podržava različite vrste odgovora i omogućava lakše rukovanje podacima koje veb server šalje, čime poboljšava efikasnost i preglednost u radu sa mrežnim zahtevima.

2.14 Ažuriranje u realnom vremenu

WebSocket je protokol koji omogućava uspostavljanje stalne, dugotrajne veze i dvosmernu komunikaciju između veb brauzera i veb servera preko jedne ostvarene TCP veze. Ovo omogućava stvaranje interaktivnih aplikacija koje zahtevaju brzu razmenu podataka u realnom vremenu.

Nakon što se veza uspostavi putem inicijalnog HTTP zahteva (engl. *handshake*), komunikacija prelazi na *WebSocket* protokol. Veza ostaje otvorena sve dok je potrebno, što omogućava klijentu i serveru da šalju i primaju podatke bez potrebe za ponovnim uspostavljanjem veze.

Ovakav pristup donosi sledeće prednosti:

- *Smanjena latencija*. Budući da veza ostaje stalno otvorena, *WebSocket* smanjuje kašnjenje koje je često povezano sa stalnim uspostavljanjem i zatvaranjem veza, što je uobičajeno kod tradicionalnih *HTTP* zahteva.
- *Manja potrošnja resursa*. *WebSocket* protokol koristi manje resursa u poređenju sa *HTTP* protokolom zato što eliminiše potrebu za višestrukim *HTTP* zahtevima i odgovorima za svaku interakciju.

Najčešće se koristi za *online igre* koje zahtevaju brzu komunikaciju između igrača u stvarnom vremenu, *chat aplikacije* u kojima se podrazumevaju

trenutna isporuka i prijem poruka, *finansijske aplikacije* koje prate cene akcija ili druge finansijske podatke u realnom vremenu, kao i na drugim veb stranicama koje takođe prikazuju podatake u realnom vremenu, kao što su vesti i sportski rezultati.

PHP nema ugrađenu podršku za *WebSocket*, ali može da koristi *WebSocket* servere pomoću dodatnih biblioteka kao što su *Ratchet*, *Wrench* ili *Swoole* koje omogućavaju *PHP*-u da upravlja *WebSocket* vezama i razmenjuje poruke u realnom vremenu.

Na strani klijenta, *WebSocket* se obično implementira u *JavaScript*-u, koristeći ugrađeni *WebSocket API* u savremenim veb brauzerima.

KORISNIČKI INTERFEJS

3. KORISNIČKI INTERFEJS

Korisnički interfejs (engl. *User Interface*, UI) veb aplikacije predstavlja vezu između korisnika i same aplikacije, i obuhvata sve vizualne i interaktivne komponente koje omogućavaju korisnicima da komuniciraju sa aplikacijom, izvrše zadatke i dobiju informacije. Osim što obezbeđuje pozitivno korisničko iskustvo, efikasno dizajniran korisnički interfejs može značajno da doprinese uspehu aplikacije.

Korisnički interfejs treba da obezbedi radno okruženje koje je jednostavno i intuitivno za upotrebu i da omogući korisnicima da na što efikasniji način koriste funkcionalnosti aplikacije, uključujući brzo i precizno izvršenje zadataka.

Korisnički interfejs takođe treba da podržava timski rad i koordinaciju aktivnosti, poboljšavajući poslovnu komunikaciju i kolaboraciju među korisnicima. Osim toga, treba da omogući efikasnu integraciju sa drugim sistemima, obezbeđujući da korisnik postigne željeni rezultat uz minimalan unos podataka.

3.1 Osnovni principi

Kreiranje korisničkog interfejsa se zasniva na nekoliko ključnih principa čiji je zadatak ne samo da se ispune funkcionalni zahtevi, već i da se obezbede efikasnost i intuitivnost u upotrebi aplikacije, kao i zadovoljstvo korisnika u radu.

Korisnički interfejs treba da bude *jasan*. Da bi se to postiglo, moraju se eliminisati sve dvosmislenosti tako što će svi koraci, tok operacija, hijerarhija, poruke i vizuelni elementi biti jasno definisani. Ovo omogućava korisnicima da intuitivno razumeju kako da koriste aplikaciju bez nepotrebne konfuzije.

Korisnički interfejs treba da bude *sažet*. Previše elemenata na ekranu može da zbuni korisnike, dok premalo elemenata može učiniti tok rada nejasnim ili konfuznim. Prava ravnoteža između ovih ekstremnih tačaka osigurava da je korisnički interfejs čist i fokusiran, što će svakako dovesti do poboljšanog korisničkog iskustva.

Korisnički interfejs treba da bude *efikasan*. IS treba da služi kao alat zaposlenima za što brže i jednostavnije obavljanje svakodnevnih poslovnih aktivnosti. Efikasan interfejs omogućava korisnicima da brzo postignu svoje ciljeve uz minimum uloženog vremena.

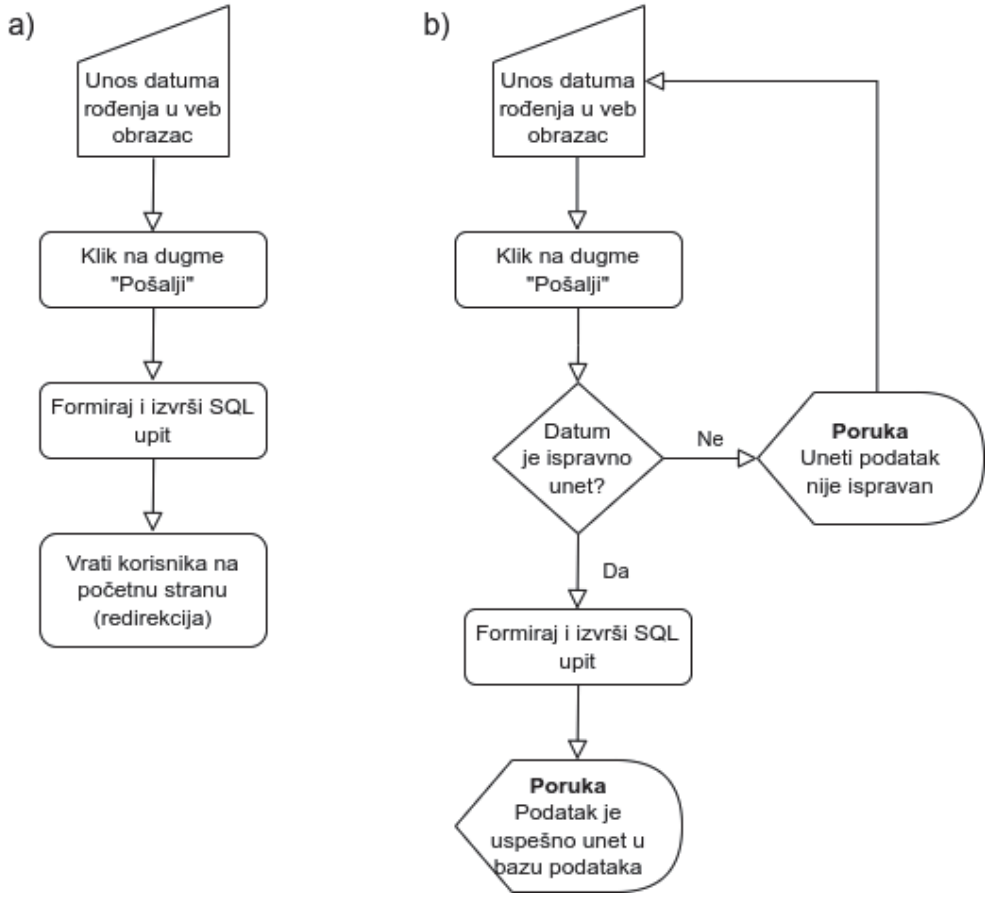
Korisnički interfejs treba da bude *poznat*. Ovaj princip osigurava da, iako korisnici možda prvi put pristupaju aplikaciji, većinu elemenata mogu da prepoznaju po izgledu i poziciji na ekranu.

Korisnički interfejs treba da bude *agilan*, odnosno reaktivan i da precizno obaveštava korisnike o trenutnom statusu softvera ili pokrenutog zadatka. To uključuje obaveštavanje o uspešnom prijemu podataka, napretku obrade i eventualnim greškama, što doprinosi transparentnosti i korisničkoj sigurnosti.

Korisnički interfejs treba da bude *dosledan*, odnosno svi segmenti ili moduli aplikacije moraju da imaju usklađen korisnički interfejs. To će pomoći u održavanju koherentnog iskustva kroz različite delove aplikacije, olakšati korisnicima navigaciju i korišćenje, i napokon, skratiti vreme neophodno za obuku korisnika.

Korisnički interfejs treba da bude *prijatan* za upotrebu. Iako o ukusima ne vredi raspravljati, dobro dizajniran korisnički interfejs olakšava izvršavanje svakodnevnih zadataka i doprinosi opštem zadovoljstvu korisnika.

Korisnički interfejs treba efikasno da *upravlja greškama*. *Tolerancija na greške* je ključna za pružanje fleksibilnog i prilagodljivog korisničkog iskustva. Interfejs treba da bude dizajniran tako da očekivani rezultati budu postignuti uz minimalnu intervenciju korisnika, čak i u slučaju grešaka prilikom unosa podataka, ili da pruži jasne informacije o tome šta se desilo i kako da se ispravi greška, slika 16.



Slika 16 - Korisnički interfejs, tolerancija na greške
a) Bez tolerancije, SQL upit će biti izvršen bez obzira na to da li je uneti podatak ispravan
b) Sa tolerancijom, prvo se vrši provera da li je uneti podatak ispravan.

Na slici 16 su prikazani primeri unosa datuma rođenja bez tolerancije na grešku (a) i sa tolerancijom (b). U prvom slučaju, bez obzira da li je korisnik uneo ispravan podatak, aplikacija će prihvatiti taj podatak, izvršiti SQL upit i vratiti korisnika na početnu stranu bez povratne informacije o tome da li je operacija uspešno izvršena ili je došlo do greške. U varijanti (b), ubačeni su dodatni koraci sa ciljem da korisnik bude obavešten o ishodu operacije. U slučaju neispravno unetog podatka (prazno polje, neodgovarajući format, ispravan datum van očekivanog opsega), ispisaće se odgovarajuća poruka i korisnički interfejs će korisnika ponovo uputiti na veb obrazac radi

korekcije unosa. Kada se unese ispravan podatak, aplikacija će formirati i izvršiti SQL upit i prikazati poruku o uspešno izvršenoj operaciji.

3.2 Dizajn

IS služi zaposlenima kao alat za što je moguće brže i jednostavnije obavljanje svakodnevnih poslovnih aktivnosti. Dizajn korisničkog interfejsa je složen proces koji se obavlja kroz više iteracija i zahteva pažljivo planiranje, testiranje i prilagođavanje kako bi se obezbedilo najbolje moguće iskustvo za korisnike.

Prvi korak u izradi korisničkog interfejsa je *razumevanje potreba i želja korisnika*. Važno je da se precizno definiše ko će koristiti sistem i koje zadatke će obavljati. Ovo uključuje razumevanje broja korisnika, njihovih prava pristupa sistemu, i nivoa njihove obučenosti. Na primer, da li će sistem koristiti samo interni tim ili i eksterni korisnici? Koliko su korisnici obučeni za rad sa sličnim sistemima? Da li će pristupati sistemu redovno ili samo povremeno? Kvalitetna priprema obuhvata prikupljanje podataka putem anketa, intervjua i analiza korisničkog ponašanja, na osnovu čega se potom prepoznaju i definišu ciljevi i zahtevi koje korisnički interfejs treba da ispuni. Razumevanje ovih faktora pomaže u oblikovanju interfejsa koji zadovoljava specifične potrebe korisnika i njihove radne navike.

Nakon što se definišu korisnici i njihovi zadaci, u sledećem koraku se vrše *priprema i konceptualizacija*. Ovo uključuje izradu spiska funkcionalnosti koje su potrebne da bi sistem uspešno ostvario postavljene ciljeve. Kako korisnici obavljaju svoje zadatke u realnom životu? Da li su postojeći sistemi već pružili uvid u to kako bi korisnici želeli da funkcioniše novi sistem? Intervjui sa korisnicima mogu pružiti dragocene uvide u njihove zamisli o budućem IS.

Arhitektura informacija je ključna za organizaciju i protok informacija kroz sistem. Ovaj korak podrazumeva planiranje kako će informacije biti strukturirane i kako će prolaziti kroz različite module aplikacije.

Dizajn interfejsa treba da bude *intuitivan*, odnosno svaki zahtev i funkcionalnost treba da bude jasno razumljiv korisniku bez potrebe za dodatnim objašnjenjima.

Provera upotrebljivosti je ključna faza u kojoj budući korisnici testiraju prototip. Ovaj proces omogućava evaluaciju da li korisnici razumeju interfejs, da li sistem ispunjava postavljene zadatke i da li su identifikovani bilo kakvi problemi ili nedostaci, a testira se kroz izradu prototipa zato što omogućava vizualizaciju osnovnih funkcionalnosti i omogućava korisnicima da testiraju i ocene dizajn pre nego što se razvije konačna verzija.

IS često obrađuju velike količine podataka, a ljudski mozak lakše razume grafički predstavljene informacije u odnosu na tabelarne prikaze numeričkih podataka. Efikasna vizualizacija podataka može značajno da poboljša korisničko iskustvo i olakša interpretaciju složenih podataka.

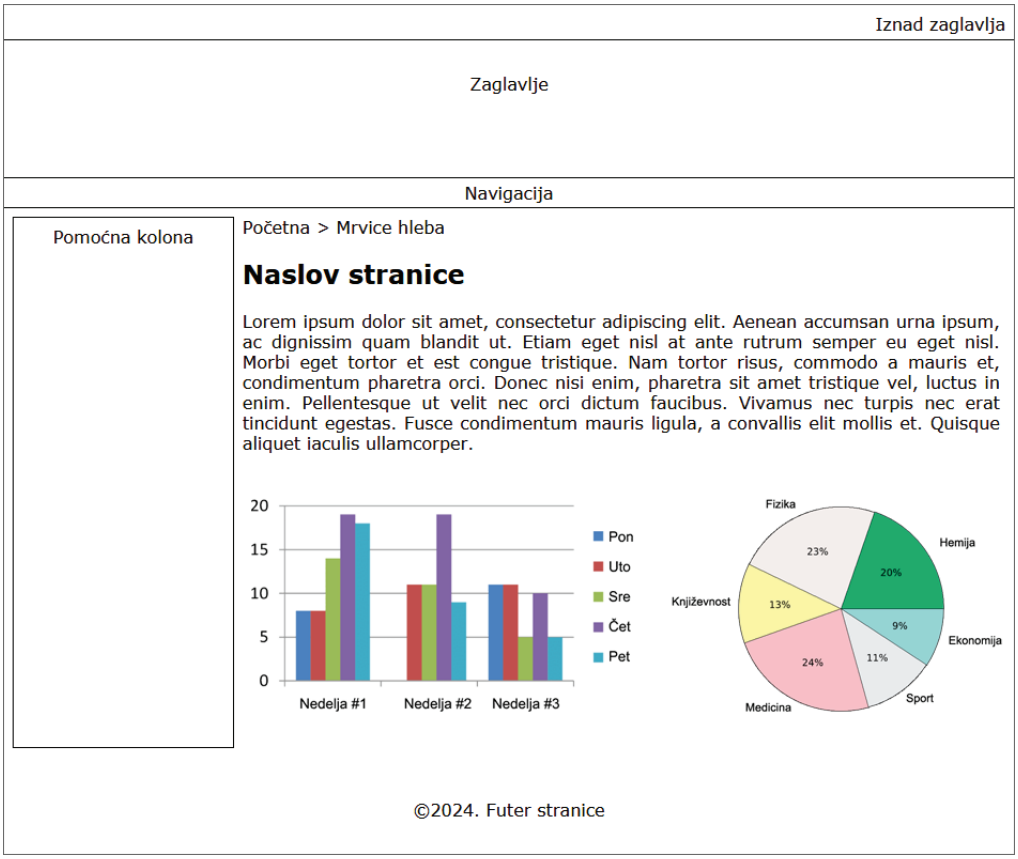
Takođe, mogućnost *personalizacije*, odnosno modifikacije korisničkog interfejsa u skladu sa individualnim željama i navikama korisnika doprinosi njegovom većem zadovoljstvu i produktivnosti.

Nakon implementacije, *održavanje* uključuje ispravljanje uočenih grešaka i razvoj novih funkcionalnosti. Dizajn korisničkog interfejsa je dinamičan proces koji zahteva kontinuirano prilagođavanje i poboljšanje kako bi se zadovoljili evoluirajući zahtevi korisnika i tehnologije.

3.3 Gradivni elementi veb stranice

Gradivni elementi veb stranice su komponente koje čine osnovu svakog korisničkog interfejsa, slika 17, i organizuju se na takav način da korisnicima omoguće da lako pronađu informacije i izvrše željene akcije. Ključne komponente gotovo svake veb stranice su zaglavlje, navigaciona lenta, glavni i sekundarni sadržaj i futer.

Zaglavlje se nalazi na vrhu veb stranice i u njemu obično nalaze naziv veb sajta ili aplikacije, logo i glavna navigaciona lenta. Zaglavlje služi korisniku kao stalna tačka orijentacije.



Slika 17 - Gradivni elementi veb stranice

Traka *iznad zaglavlja* se obično koristi za postavljanje sporednih navigacionih elemenata kao što su linkovi za prijavljivanje, odjavu i registraciju, korpa u slučaju e-prodavnice i polje za pretragu sadržaja.

Glavni sadržaj je centralni deo stranice koji prikazuje najvažnije informacije i funkcionalnosti. Obuhvata elemente kao što su naslov stranice, tekst, slike, tabele, video materijal ili bilo koji drugi sadržaj relevantan za korisnike. Glavni sadržaj je obično najistaknutiji deo stranice i treba da bude organizovan na način koji olakšava korisnicima da pronađu i koriste informacije.

Sekundarni sadržaj pruža dodatne informacije koje podržavaju glavni sadržaj, ali nisu centralne za osnovnu funkciju stranice. To može uključivati dodatne resurse, linkove ka srodnim temama, promotivne sadržaje ili pomoćne informacije.

Futer se nalazi na dnu stranice i obično sadrži informacije kao što su kontakt podaci, linkovi ka politici privatnosti, uslovima korišćenja i drugim relevantnim informacijama. Takođe može uključivati dodatne navigacione linkove ili linkove ka društvenim mrežama.

3.3.1 Navigacija

Navigacija je ključna komponenta koja omogućava korisnicima da se kreću kroz različite sekcije veb aplikacije. Najčešće se stilizuje kao navigaciona lenta, odnosno glavni navigacioni meni ispod zaglavlja stranice, ali se neretko postavlja i na drugim mestima, npr. u pomoćnoj koloni. Ukoliko sadrži veći broj opcija, može da sadrži padajuće menije.

Navigacija mora da bide logički strukturirana, tako da korisnik u svakom trenutku zna odakle je došao, gde je stigao i šta sledi. Efikasna navigacija doprinosi opštem korisničkom iskustvu i omogućava brzu adaptaciju na novi sistem.

Radi uštede prostora na ekranu, navigacija može da sadrži padajuće menije (engl. *dropdown menu*)

3.3.2 Mrvice hleba

Pomoćni navigacioni element su tzv. *mrvice hleba* (engl. *breadcrumbs*). Obično se pozicioniraju iznad naslova stranice, element *<h1>*, a korisnicima pružaju daje jasnu predstavu o strukturi sadržaja veb aplikacije i trenutnom položaju u toj strukturi, slika 18.



Slika 18 - Pomoćni navigacioni element, tzv. mrvice hleba

Ukoliko je ovaj element dobro struktuiran, pomoći će korisnicima da lakše savladaju složenu organizaciju stranica, a time će unaprediti i korisničko iskustvo.

3.3.3 Paginacija

Paginacija u veb aplikaciji je pomoćni navigacioni element i tehnika koja omogućava korisnicima da pregledaju velike količine podataka ili sadržaja podeljeno na više stranica, umesto da prikaže sve podatke na jednom mestu. Primenom ove metode se optimizuju performanse aplikacije, sadržaj se efikasnije organizuje, a navigacija i pretraživanje podataka su olakšani.

Paginacija može značajno da skrati vreme učitavanja stranica i smanji opterećenje servera, tako što će po stranici prikazati samo određeni broj stavki, odnosno podskup većeg skupa.



Slika 19 - Paginacija, najjednostavniji oblik

Može da se stilizuje na različite načine, u zavisnosti od tipa sadržaja koji se prikazuje i konkretnih potreba aplikacije. Najjednostavniji oblik paginacije može da se formuliše pomoću dva dugmeta, za prethodnu i sledeću stranicu, slika 19.



Slika 20 - Paginaciona lenta sa brojevima stranica

Klasična paginacija prikazuje niz brojeva stranica sa linkovima ka svakoj, osim tekućoj stranici, kao i dugmad za prelazak na sledeću ili prethodnu stranicu, slika 20.

U slučaju velikh skupova nije optimalno da se prikazuju brojevi svih stranica, već se primenjuje skraćeni prikaz kombinacijom podskupova, slika 21. Ukoliko je aktivna stranica na početku ili kraju skupa, prikazaće se samo početni i krajnji podskup, slika 21a. U ostalim slučajevima se prikazuje i podskup u neposrednoj blizini stranice (±1 stranica), slika 21b.



Slika 21 - Paginaciona lenta sa veb obrascem, phpBB

S druge strane, beskonačno skrolovanje omogućava automatsko učitavanje novih podataka dok korisnik skroluje prema dnu stranice, čime se stvara neprekidan tok sadržaja, što se često koristi na društvenim mrežama. Alternativno, moguća je primena dugmeta *Učitaj više*, koje korisnik može da klikne za učitavanje dodatnog sadržaja.

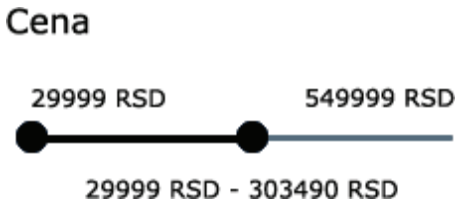
Beskonačno skrolovanje kao vid paginacije nije prikaladan za primenu u IS zato što otežava pristup željenim informacijama zbog nedostatka jasne podele na stranice. Takođe može da izazove probleme sa performansama veb brauzera zbog velike količine podataka na veb stranici, uz otežano indeksiranje i analitiku. Za poslovne aplikacije koje zahtevaju jasnu strukturu podataka i jednostavnu navigaciju između sekcija, tradicionalna paginacija je efikasniji pristup.

3.3.4 Slajderi

Slajderi mogu da budu važan element korisničkog interfejsa IS, zato što omogućavaju intuitivno i dinamično podešavanje parametara i vrednosti. Na primer, slajderi mogu da se upotrebe za podešavanje raspona vrednosti, kao što su budžetske granice, procene ulaganja ili vremenski intervali, što će potom omogućiti korisnicima da bolje analiziraju i interpretiraju podatke.

U alatima za upravljanje projektima, slajderi mogu pomoći u raspodeli resursa, kao što su radni sati ili budžet, kako bi se efikasnije planirali i pratili projekti, dok u istraživačkim alatima za naučne ili tehničke analize, slajderi omogućavaju prilagođavanje parametara eksperimenta, kao što su temperature, pritisci ili koncentracije.

Pored podešavanja parametara, slajderi su korisni i za filtriranje i sortiranje podataka. U filterima za pretragu, slajderi pomažu u pretrazi proizvoda prema ocenama, ili prema rasponu cena, slika 22. Takođe, omogućavaju istraživanje podataka kroz vremenske periode ili dimenzije, dok u edukativnim aplikacijama pomažu u simulacijama kako bi se prikazale posledice promena usled različitih faktora.



Slika 22 - Slajder za podešavanje raspona cene u filteru za pretragu

Za efektivno korišćenje slajdera u dizajnu korisničkog interfejsa, važno je obezbediti jasne oznake i vrednosti, kao i vizuelnu povratnu informaciju koja omogućava praćenje promena. Pristupačnost slajdera takođe mora da bude optimalna svim korisnicima, bez obzira na tip uređaja koji koriste za rad sa aplikacijom.

3.3.5 Tagovi

Tag u korisničkom interfejsu je ključna reč koja se koristi za organizaciju, filtriranje i pretragu sadržaja. Tagovi pomažu u kategorizaciji informacija, omogućavaju lakšu navigaciju i personalizaciju korisničkog iskustva.

Na primer, u blogu ili na forumu, tagovi mogu da označavaju teme kao što su *tehnologija*, *putovanja* ili *hrana*, omogućavajući korisnicima da lako pretražuju sadržaj, slika 23.



Slika 23 - Tagovi za filtriranje vesti sa vremenskom prognozom

U e-prodavnici, tagovi omogućavaju dinamičko filtriranje rezultata prema kriterijumima kao što su *patike za košarku*, *akcija* ili *majice*, čime se poboljšava relevantnost pretrage.

Osim što olakšavaju pretragu i filtriranje, tagovi doprinose personalizaciji korisničkog iskustva. U aplikacijama za preporuke, tagovi se koriste za predlaganje sadržaja na osnovu prethodnih interakcija i interesa korisnika.

Takođe, tagovi mogu pomoći u praćenju korisničkih preferencija i analizi njihovih interakcija, što omogućava kreiranje prilagođenih preporuka. Tagovi mogu da se iskoriste da označe ključne delove sadržaja ili da prikažu meta-informacije o autoru i datumu objave.

Tagovi su takođe korisni za upravljanje zadacima i projektima. U alatima za upravljanje projektima, tagovi omogućavaju organizaciju zadataka prema vrstama aktivnosti, prioritetima ili statusima, kao što su *hitno* ili *završeno*. Time se olakšava praćenje radnog procesa u različitim projektima, fazama ili aktivnostima. U alatima za timsku saradnju, tagovi mogu označiti članove tima ili specifične grupe, što može da olakša komunikaciju i koordinaciju između članova tima, dok u analitičkim alatima, tagovi omogućavaju filtriranje i sortiranje podataka prema specifičnim kriterijumima, što doprinosi boljoj analizi i interpretaciji informacija.

3.4 Informacije, obaveštenja i poruke

Pravovremeno pružanje informacija korisnicima je od suštinskog značaja za uspešno korišćenje sistema.

- *Obaveštavanje o rezultatu akcija.* Korisnici treba da budu obavešteni o rezultatima svojih akcija. Na primer, nakon slanja forme, korisnik treba da dobije potvrdu da je podatak uspešno poslat.
- *Obaveštavanje o posledicama akcija.* Pre nego što korisnik izvrši akciju koja može da ima značajne posledice, npr. brisanje beleške iz baze podataka, treba ga unapred obavestiti o mogućim posledicama.

- *Pravovremene informacije.* Informacije o napretku ili statusu operacija treba da budu ažurne i precizne. Korisnika ne treba obaveštavati da je operacija završena ukoliko proces još uvek traje, jer to može dovesti do konfuzije.

Informacije na veb stranici mogu biti predstavljene na različite načine, uključujući tekst, tabelarne izveštaje i grafičke informacije. Tekstualne informacije pružaju detaljne opise i objašnjenja, dok tabelarni izveštaji organizuju podatke u strukturu koja je pregledna. Grafičke informacije, kao što su dijagrami i grafikoni, slika 17, pomažu u vizualizaciji podataka i olakšavaju njihovo razumevanje.

Obaveštenja i poruke prenose korisnicima informacije o važnim događajima ili statusima unutar aplikacije, kao što su obaveštenja o uspešnim akcijama, greškama i upozorenjima, slike 24 i 25, ili sistemskim porukama.



Slika 24 - Obaveštenje o uspešno izvršenoj operaciji

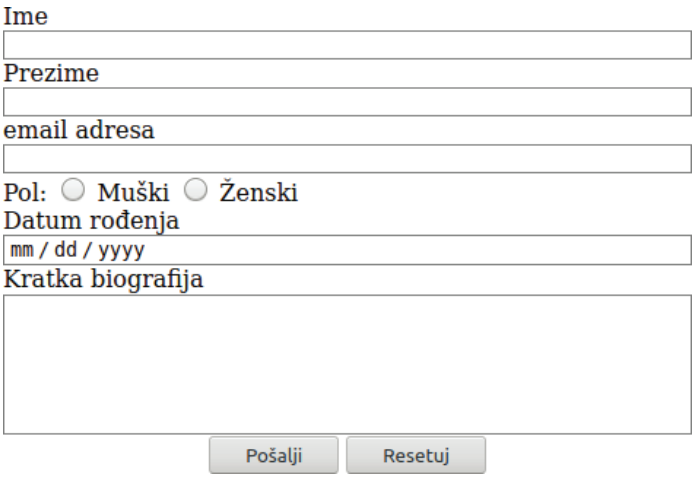


Slika 25 - Upozorenje da operacija nije uspešno izvršena

Jasna i pravovremena obaveštenja poboljšavaju komunikaciju i pomažu korisnicima da pravilno reaguju na situacije koje se javljaju tokom korišćenja sistema.

3.5 Unos podataka

Veb obrazac, slika 26, omogućava komunikaciju između korisnika i veb aplikacije kroz unos i slanje podataka veb serveru. Uneti podaci se koriste za dalju obradu ili se skladište u bazi podataka.



Slika 26 - Veb obrazac za unos osnovnih ličnih podataka

Elementi za unos podataka omogućavaju korisnicima da unose informacije. Pravilno dizajnirana polja olakšavaju unos tekstuelnih i numeričkih informacija, smanjuju mogućnosti pojave greške i omogućavaju korisnicima da efikasno završe svoje zadatke. Svaki tip input elementa ima svoju specifičnu ulogu, a razumevanje i pravilna primena ovih osnovnih elemenata omogućava kreiranje korisničkog interfejsa koji će zadovoljiti postavljene funkcionalne zahteve, i pružiti prijatno i intuitivno korisničko iskustvo.

Input element tipa tekst, *<input type="text">*, jedan je od osnovnih elemenata i omogućava korisnicima da unose tekstualne podatke kao što su imena, adrese, ili druge informacije koje su potrebne za dalji rad sistema. Koristi se kada je očekivana informacija kratka. U slučaju dužeg teksta, koristi se višelinijsko polje koje se označava HTML tag *<textarea>*.

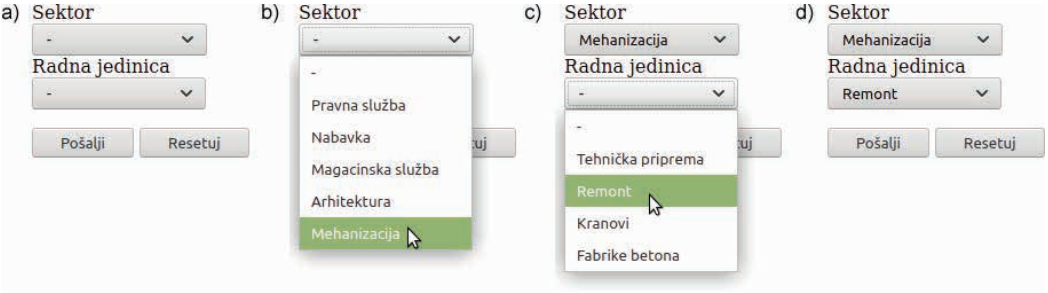
Input element tipa *radio* omogućaju korisnicima da izaberu jednu od više ponuđenih opcija. Ova funkcionalnost je korisna kada korisnik treba da donese preciznu odluku koja je isključiva, što znači da može biti izabrana samo jedna opcija u datom trenutku.

Nasuprot tome, input elementi tipa *checkbox* omogućavaju korisnicima da izaberu više opcija iz skupa ponuđenih opcija. Ovi elementi su idealni za

situacije u kojima korisnik treba da označi više stavki koje nisu međusobno isključive, kao što su interesovanja, preference ili dodatne opcije u formularima. Checkbox elementi pružaju fleksibilnost i omogućavaju korisnicima da prilagode svoje izbore u skladu sa potrebama.

Padajuće liste predstavljaju element koji zavisno od konfiguracije, omogućava korisnicima da izaberu samo jednu, ili više opcija iz ponuđene liste koja se pojavljuje kada korisnik klikne na element, slika 27. Mogu da se kombinuju sa asinhronim pozivima, tako da promena izabrane opcije u jednoj listi (*Sektor*), slika 27 b), vodi do promene sadržaja u povezanoj listi (*Radna jedinica*), slika 27 c).

Padajuće liste su efikasne u optimizaciji prostora na ekranu zato što omogućavaju prikazivanje dugog popisa opcija u kompaktnom formatu. Korisnici mogu lako pregledati i izabrati željenu opciju, a lista se automatski zatvara nakon izbora, čime se održava urednost i preglednost korisničkog interfejsa.



Slika 27 - Upotreba padajućih lista u veb obrascu

Liste su slične padajućim listama, osim što se razlikuju po tome da prikazuju više stavki koje korisnici mogu da pregledaju i izaberu.

Dugmeta su osnovni kontrolni element veb obrasca i tačke za iniciranje akcije prema sistemu. Omogućavaju korisnicima da pokrenu određene akcije, kao što su slanje podataka ili resetovanje elemenata u veb obrascu na početne vrednosti, čuvanje podataka ili pokretanje procesa.



Slika 28 - Upotreba prekidača u veb obrascu

Prekidač (engl. *toggle switch*) je tip dugmeta koji korisniku omogućava da uključi ili isključi određenu opciju ili funkcionalnost, slika 28. Ovakvi elementi su posebno korisni za binarne odluke (*uključeno/isključeno*). Važno je da se naglasiti da se prekidači formiraju kombinacijom standardnih HTML elemenata i CSS-a.

Input element tipa *date*, slika 29, omogućava korisnicima da izaberu datum iz grafičkog prikaza kalendara. Ovaj element pojednostavljuje proces unosa datuma, smanjujući mogućnost greške i ubrzavajući unos podataka.

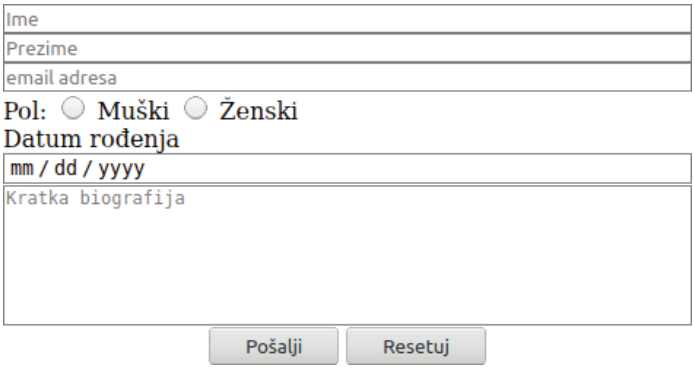


Slika 29 - Input element tipa *date*

Kalendar omogućava korisnicima da lako pregledaju i odaberu tačan datum, što je posebno korisno u aplikacijama koje zahtevaju precizan unos vremenskih informacija.

3.5.1 Obeležavanje elemenata u veb obrascu

Elementi u veb obrascima mogu da se obeležavaju na različite načine, sa ciljem da korisnicima pomognu da razumeju svrhu svakog polja i u skladu sa tim unesu očekivane podatke.



Slika 30 - Obeležavanje elemenata u veb obrascu, atribut *placeholder*

Najčešći način za obeležavanje elemenata je pomoću HTML taga *label*, koji pruža tekstualnu oznaku za svaki element obrasca, slika 26. Ovaj element koristi atribut *for* kako bi se povezao sa odgovarajućim HTML elementom preko njegovog *id* atributa.

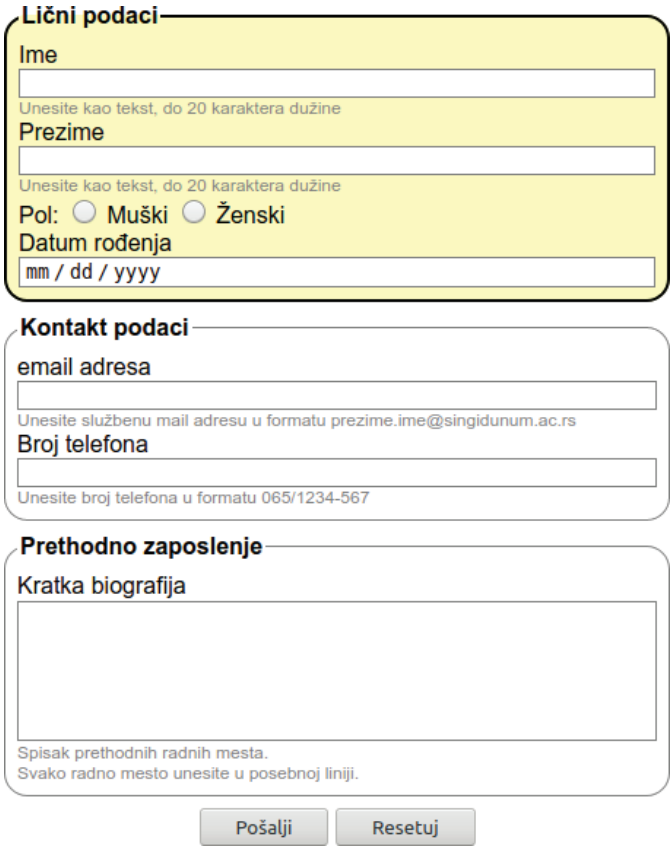
```
18      <!-- Izvorni input element tipa 'date'
19           označen tagom 'label' -->
20      <label>Datum rođenja</label>
21      <input type="date" name="datum_rodjenja">
22      <!-- Maskiranje input elementa tipa 'date'
23           radi upotrebe atributa 'placeholder' -->
24      <input
25          type="text"
26          placeholder="Datum rođenja"
27          name="datum_rodjenja"
28          onfocus="this.type='date';">
```

Listing 8 - Maskiranje input elementa tipa date radi upotrebe atributa *placeholder*

Atribut *placeholder* prikazuje privremeni tekst unutar polja za unos, koji daje korisnicima smernice o tome šta treba da unesu. Ovo može biti korisno za pružanje kratkih instrukcija ili primera podataka, ali se ne može uvek koristiti. Na slici 30 je prikazana varijacija veb obrasca sa slike 26, gde je za obeležavanje elemenata umesto taga *label* korišćen atribut *placeholder*.



Slika 31 - Dodatni opis elementa za unos teksta



Slika 32 - Grupisanje elemenata u veb obrascu

Upotreba atributa placeholder nije moguća na input elementima tipa *radio* i *checkbox*. Da bi se simulirao željeni efekat na input elementu tipa *date*, izvorno se postavlja input element tipa *text* nad kojim se osluškuje JavaScript događaj *onfocus*, kada se tip elementa menja u *date*, listing 8, linije 24-28.

Input elementi mogu dodatno da se opišu porukama koje se nalaze pored ili ispod elemenata, i koje treba da budu stilizovane na takav način da se vizuelno razlikuju od ostatka obrasca. Po potrebi može da se upotrebi tzv. *tool-tip* pomoću atributa *title*, koji će biti prikazan kada korisnik dođe mišem iznad željenog elementa, slika 31.

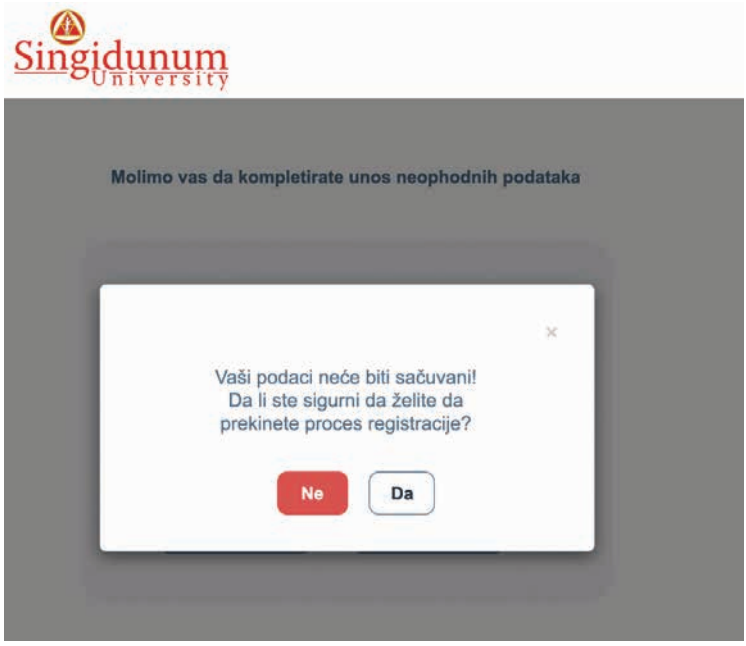
U slučajevima kada postoji grupa povezanih elemenata, korišćenje *fieldset* i *legend* HTML tagova omogućava organizovanje elemenata u logičke grupe, slika 32. Po potrebi, aktivni set elemenata može da se stilizuje pomoću CSS pseudo klase *focus-within*.

Za poboljšanje pristupačnosti, *ARIA* (engl. *Accessible Rich Internet Applications*) atributi mogu da se koriste kako bi se dodale dodatne informacije za korisnike čitača ekrana i drugih pomoćnih tehnologija.

3.6 Modalni prozori

Modalni prozor, alternativno *modalni dijalog* ili *popup*, je tip prozora u veb aplikacijama koji se pojavljuje iznad glavnog sadržaja stranice. Formiraju se korišćenjem HTML-a, CSS-a i JavaScript-a, a dizajnirani su na takav način da privuku pažnju korisnika i često služe za prikazivanje dodatnih informacija ili za unos podataka bez napuštanja trenutne stranice. Obično se koriste u sledećim scenarijima:

- *Važna obaveštenja i poruke*, kao što su greške, upozorenja ili potvrde, slika 33.
- *Prikaz dodatnih informacija*, kao što su detaljni listing podataka, odabir opcija pre izvršenja neke radnje, ili dodatne instrukcije.
- *Formulari za unos podataka.* Modalni prozori omogućavaju korisnicima da popune forme ili unesu podatke bez napuštanja trenutne stranice.



Slika 33 - Modalni prozor za potvrdu prekida procesa registracije



Slika 34 - Praćenje napretka u okviru operacije iz više koraka

3.7 Praćenje napretka

Često se dešava u IS da započeti proces usled svoje složenosti mora da se izvrši u više koraka, slika 34.

Traka za napredovanje (engl. *progress bar*) je vizuelni element koji prikazuje napredak u izvršavanju tog procesa. Može biti prikazan kao

- horizontalna ili vertikalna traka sa ključnim koracima, pri čemu su završeni koraci jasno označeni.
- horizontalna ili vertikalna traka koja se ispunjava dok se proces odvija. Često se koristi za označavanje trenutno aktivnog koraka, ili za prikaz procenta koliko je započeta operacija kompletirana, što omogućava korisnicima da u realnom vremenu vizualno prate koliko je posla obavljeno i koliko još preostaje. Najčešće se koristi prilikom preuzimanja velikih dokumenata, u obradi velikih količina podataka kao što je importovanje baze podataka, ili prilikom učitavanja veb stranice.

3.8 Interaktivnost

Interaktivnost korisničkog interfejsa igra ključnu ulogu u oblikovanju pozitivnog korisničkog iskustva. Kako bi korisnici mogli efikasno da koriste IS, dizajneri i programeri moraju da se fokusiraju na nekoliko ključnih aspekata, kao što su metodčnost ili efikasnost, predvidljivost, sugestije, informacije, i obaveštenja. Svaki od ovih aspekata doprinosi stvaranju interfejsa koji je funkcionalan i lako razumljiv.

3.8.1 Metodičnost ili efikasnost

Prilikom dizajniranja korisničkog interfejsa, često se postavlja pitanje da li treba težiti metodičnosti ili efikasnosti. Metodičnost podrazumeva da se do cilja dolazi kroz korake koji su korisniku poznati i prijatni, dok efikasnost teži smanjenju broja koraka potrebnih za postizanje istog cilja.

Kada korak zahteva unos velikog broja podataka, skrivanje elemenata radi preglednosti veb obrasca može da učini proces naizgled preglednijim, ali

može da izazove zabunu ukoliko korisnici nisu sigurni šta se dešava ili gde da pronađu potrebne informacije.

Za korisnike koji su navikli na određeni način rada, prelazak na efikasniji, ali nepoznat način rada može izazvati konfuziju i frustraciju. Na primer, ako sistem omogućava postizanje cilja u samo jednom koraku, ali korisnici preferiraju da koriste tri koraka zbog svoje prethodne navike, promena može imati suprotan efekat. Ključ je u balansu između metodičnosti i efikasnosti, omogućavajući korisnicima da izaberu put koji im najviše odgovara.

3.8.2 Predvidljivost

Predvidljivost je ključna za intuitivno korišćenje IS. Korisnici očekuju da elementi na ekranu budu postavljeni na mesta gde su navikli da ih vide. Na primer, dugmeta *Pošalji* ili *Sačuvaj* se obično nalaze na dnu veb obrasca, dok je navigacioni meni najčešće na vrhu stranice. Ako su elementi raspoređeni u skladu sa očekivanjima, korisnici će ih brže pronaći i lakše će koristiti sistem.

Da bi se postigla ovakva predvidljivost, moraju da se sagledaju najbolje prakse u industriji, kao i da se analizira korisničko ponašanje pre nego što se počne sa projektovanjem samog IS.

3.8.3 Sugestije

Interfejs treba da bude proaktivan u pogledu pomoći korisnicima kroz sugestije. Ovo uključuje:

- *Predlaganje sledeće akcije.* Nakon što korisnik završi sa određenim zadatkom, korisnički interfejs treba da predloži sledeću logičnu akciju ili korak, čime se olakšava nastavak rada i povećava efikasnost.
- *Inicijalno popunjavanje polja.* Ako su neki podaci već dostupni, npr. iz baze podataka, prethodnih sesija ili profila korisnika, aplikacija može unapred da popuni polja za unos, što potom može da dovede do uštede vremena i smanjiti potrebu za ponovnim unosom istih informacija.

- *Minimiziranje nepotrebnih operacija.* Interfejs treba da svede na minimum nepotrebne operacije ili korake kako bi se omogućila brža i efikasnija interakcija. Ovo uključuje automatsko popunjavanje veb obrazaca, pamćenje prethodnih podešavanja i izbora, kao i smanjenje potrebnog broja klikova i podataka za unos.

3.9 Prilagodljivi dizajn

Iako je pristup Internetu preko desktop ili laptop računara dugo bio dominantan, poslednjih godina primat preuzimaju mobilni uređaji [5], ali to nikako ne znači da se računari napuštaju kao platforma zato što svaki uređaj ima svoju ulogu u tehnološkom ekosistemu.

U kontekstu IS, izbor između mobilnih uređaja i klasičnih računara zavisi od funkcionalnih zahteva aplikacije, radnog okruženja i konkretnih potreba korisnika u datom trenutku. Postoje situacije u kojima podjednako efikasno mogu da se koriste oba tipa uređaja, ali i pored toga prilikom razvoja korisničkog interfejsa moraju da se uzmu u obzir specifične osobine svakog tipa uređaja koji biti korišćen za pristup IS.

Prilagodljivi veb dizajn (engl. *Responsive design*) omogućava aplikaciji da se automatski prilagodi različitim veličinama i rezolucijama ekrana. To uključuje korišćenje fluidnih rasporeda elemenata na veb stranici, čije će se veličine i raspored prilagoditi karakteristikama ekrana uređaja preko koga se vrši pristup aplikaciji.

Upotreba HTML5 i CSS3 standarda omogućavaju implementaciju funkcionalnosti i prikaz sadržaja na nivou tradicionalnih desktop aplikacija, dok JavaScript biblioteke i radni okviri, kao što su *React*, *Angular* i *Vue.js*, mogu da pomognu u kreiranju dinamičnih aplikacija koje optimalno funkcionišu na različitim veb brauzerima.

Opsežno testiranje aplikacije na različitim pretraživačima i uređajima omogućava identifikaciju i ispravljanje problema sa kompatibilnošću. Automatizovani testovi olakšavaju ovaj proces i pomažu u otkrivanju potencijalnih problema.

BEZBEDNOST VEB APLIKACIJA

4. BEZBEDNOST VEB APLIKACIJA

Bezbednost veb aplikacije je kompleksan i višeslojan proces koji uključuje različite aspekte kako bi se obezbedila zaštita podataka i resursa od potencijalnih pretnji. Efikasna zaštita zahteva sveobuhvatan pristup koji uključuje prevenciju, detekciju, i odgovor na bezbednosne incidente.

Veb aplikacije mogu da poseduju različite mehanizme i sigurnosne mere za zaštitu podataka i korisničkih informacija, kao što su provera identiteta korisnika (identifikacija), kontrola pristupa (autorizacija), kao i zaštita od različitih vrsta napada, poput SQL injekcija i XSS (engl. *Cross-Site scripting*).

4.1 Identifikacija i autorizacija korisnika

Identifikacija i autorizacija korisnika su ključni aspekti bezbednosti IS, i od suštinskog značaja za zaštitu podataka i resursa. Dok identifikacija omogućava prepoznavanje korisnika, autorizacija osigurava da samo ovlašćeni korisnici imaju pristup određenim komponentama sistema i podacima. Zajedno, ove dve komponente pomažu u zaštiti sistema od neovlašćenog pristupa, očuvanju integriteta i privatnosti podataka, i osiguranju da aktivnosti korisnika budu praćene i pravilno kontrolisane.

Identifikacija korisnika se obično izvodi primenom jedinstvenih identifikatora, kao što su *korisničko ime*, *ID broj*, *broj telefona* ili drugi prepoznatljivi podaci u kombinaciji sa lozinkom. Iako je ova funkcionalnost prisutna u većini veb aplikacija u kojima se sprovodi dinamičko upravljanje sadržajem, njen značaj u kontekstu IS je neuporedivo veći.

- *Praćenje aktivnosti korisnika u IS* pomaže u nadzoru i analizi korišćenja sistema. Ovako prikupljeni podaci su od neprocenjive važnosti za otkrivanje i istraživanje sigurnosnih incidenata.

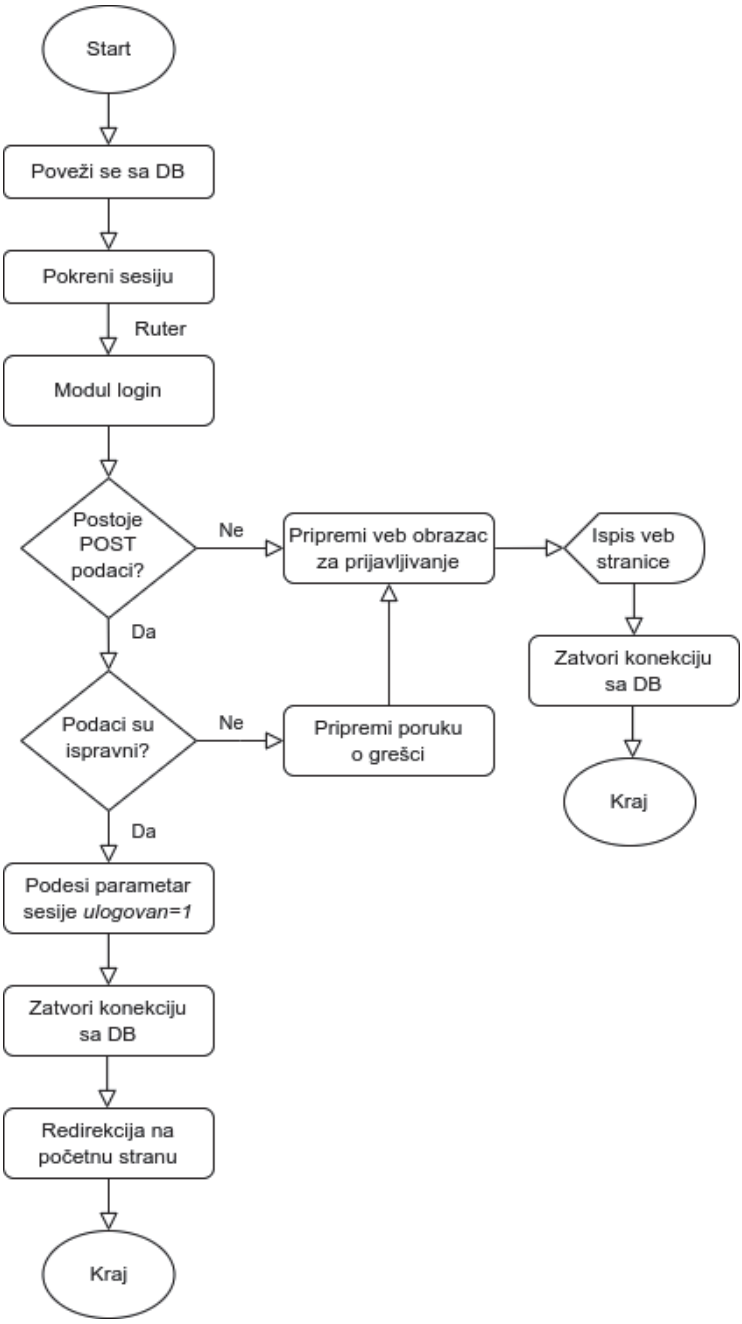
- *Identifikacija* je preduslov za *autorizaciju*, odnosno da svaka radnja u sistemu može da bude precizno povezana sa konkretnim korisnikom.
- *Personalizacija* omogućava distribuciju sadržaja, pristup funkcionalnostima i prilagođavanje korisničkog iskustva na osnovu podešavanja dodeljenih konkretnom korisničkom nalogu.

Nakon identifikacije korisnika, vrši se njegova autorizacija, odnosno određivanja prava pristupa određenim funkcionalnostima u okviru IS. Značaj autorizacije se ogleda u sledećem:

- Kontrola pristupa obezbeđuje da samo ovlašćeni korisnici mogu pristupiti određenim funkcionalnostima, podacima ili resursima, što je ključno za zaštitu osetljivih informacija i sprečavanje neovlašćenog pristupa.
- Pomaže u zaštiti podataka od neovlašćenog pristupa i manipulacije. Na primer, samo administratori mogu da pristupe funkcijama za upravljanje korisničkim nalozima, dok obični korisnici mogu samo da pregledaju svoje podatke.
- Omogućava organizacijama da ispune zahteve propisane standardima i regulativama u pogledu zaštite podataka i bezbednosti u različitim kontekstima. Tu spadaju GDPR (engl. *General Data Protection Regulation*) regulativa Evropske unije koja se bavi zaštitom ličnih podataka i privatnošću građana EU, američki zakon HIPAA (engl. *Health Insurance Portability and Accountability Act*) koji se bavi zaštitom zdravstvenih informacija i privatnošću pacijenata, ili standard PCI-DSS (engl. *Payment Card Industry Data Security Standard*) razvijen kako bi zaštitio informacije o platnim karticama i sprečio prevare i krađe. Zajednička osobina im je da zahtevaju preciznu kontrolu pristupa i zaštite podataka.

4.2 Sesije

Sesije se koriste za praćenje korisnika tokom interakcije sa veb aplikacijom. Prvi korak je inicijalizacija nove sesije, ili nastavak rada sa već započetom sesijom.



Slika 35 - Redosled koraka u procesu prijave korisnika

Kada se sesija pokrene, PHP generiše jedinstveni *id* sesije koji se čuva na dva mesta - na veb serveru i u kolačiću u korisnikovom veb brauzeru, i služi kao ključ za povezivanje korisnika sa podacima sesije na serveru. Tokom sesije, korisnički podaci, kao što su informacije o prijavi ili preferencijama, mogu da se sačuvaju u superglobalnoj promenljivoj *$_SESSION*, listing 9. Na slici 35 je prikazan redosled koraka u aplikaciji prilikom aktivacije modula za prijavljivanje korisnika. U tabeli 10 je dat pregled PHP funkcija za kontrolu sesija.

```
$_SESSION['user_id'] = $user_id;
$_SESSION['ulogovan'] = 1;
```

Listing 9 - Podešavanje parametara sesije, PHP

Sigurnost sesija uključuje generisanje sigurnih sesijskih identifikatora i obezbeđivanje da ovi identifikatori budu zaštićeni od krađe i manipulacije. To podrazumeva korišćenje sigurnih kolačića (engl. *cookies*) sa atributima kao što su *HttpOnly* i *Secure*, kao i implementaciju mehanizama za automatski prekid sesije nakon konfigurisanog perioda neaktivnosti.

Iako je tehnički moguće, nije dobra praksa da se podaci o sesiji čuvaju u kukiju, zato što takvim pristupom njihova sigurnost i privatnost mogu da budu kompromitovani. Umesto toga, podaci se čuvaju na serverskoj strani, a jedini podatak koji ostaje u kolačiću jeste identifikator sesije.

Tabela 10 - PHP funkcije koje se koriste za kontrolu sesije

Funkcija	Opis
session_start()	Pokreće novu sesiju ili nastavlja postojeću
session_destroy()	Uništava sve podatke o sesiji
session_unset()	Briše sve varijable sesije, ali ne uništava samu sesiju.
session_regenerate_id()	Generiše novi ID sesije, čime se sprečava sesijska krađa identiteta.
session_id()	Vraća ili postavlja ID sesije.
session_name()	Vraća ili postavlja ime sesije.
session_get_cookie_params()	Vraća trenutne parametre kolačića sesije.
session_set_cookie_params()	Postavlja parametre kolačića sesije.
session_write_close()	Zatvara trenutnu sesiju i odmah upisuje sve podatke sesije.
session_cache_expire()	Vraća ili postavlja vreme isteka sesije u minutama.
session_cache_limiter()	Vraća ili postavlja ograničivač keširanja za sesije

Tabela 11 - PHP funkcije primenjive u sanitaciji i validaciji podataka

Funkcija	Opis
Escaping i sanitacija stringova	
htmlspecialchars()	Konvertuje specijalne karaktere u HTML entitete kako bi se sprečili XSS napadi.
htmlentities()	Konvertuje sve primenljive karaktere u HTML entitete.
strip_tags()	Uklanja HTML i PHP tagove iz stringa.
filter_var()	Validacija i sanitacija podataka primenom različitih filtera.
addslashes()	Dodaje escape karaktere za specijalne karaktere u stringu za korišćenje u SQL upitima.
mysqli_real_escape_string()	Dodaje escape karaktere za specijalne karaktere u stringu za korišćenje u SQL upitima kada se koristi MySQLi.
Funkcije za validaciju	
filter_var()	Može se koristiti za validaciju kao i za sanitizaciju.
preg_match()	Izvršava regularne izraze za validaciju formata (npr. brojevi telefona).
ctype_* funkcije	Validira tipove karaktera (npr. ctype_alpha(), ctype_digit()).
Trimovanje i filtriranje	
trim()	Uklanja beline ili druge preddefinisane karaktere sa obe strane stringa.
ltrim()	Uklanja beline ili druge preddefinisane karaktere sa leve strane stringa.
rtrim()	Uklanja beline ili druge preddefinisane karaktere sa desne strane stringa.
Filtriranje i validacija podataka	
filter_input()	Uzima promenljivu iz ulaznih podataka i filtrira je.
filter_input_array()	Uzima više promenljivih iz ulaznih podataka i filtrira ih prema zadatim filterima.
Numerička sanitacija	
intval()	Konvertuje promenljivu u ceo broj.
floatval()	Konvertuje promenljivu u decimalni broj.
number_format()	Formatira broj sa grupisanjem hiljada.
Ostale Funkcije	
preg_replace()	Izvodi zamenu baziranu na regularnim izrazima kako bi se sanitizovali ili modifikovali stringovi.
basename()	Uklanja informacije o direktorijumu iz putanje

CSRF napadi manipulišu korisničkim sesijama kako bi izvršili neautorizovane akcije u ime korisnika. Za zaštitu od CSRF napada, veb aplikacije treba da koriste tokene u svim veb obrascima. Tokeni se generišu

i proveravaju za svaki zahtev koji menja podatke, čime se obezbeđuje da samo legitimni zahtevi mogu da budu obrađeni.

4.3 Sanitacija i validacija ulaznih podataka

Sanitacija i validacija podataka su ključni procesi u dizajnu sigurnih i pouzdanih IS. Ovi procesi osiguravaju integritet, tačnost i sigurnost podataka, a takođe pomažu u zaštiti od različitih bezbednosnih pretnji kao što su SQL injekcije, cross-site scripting i druge vrste napada. Ukoliko se efikasno koriste, rizik od sigurnosnih pretnji može značajno da se smanji, a samo funkcionisanje aplikacija će biti pouzdanije. U tabeli 11 su navedene osnovne PHP funkcije koje mogu da se koriste u ovu svrhu.

4.3.1 Sanitacija podataka

Pojam *sanitacije podataka* odnosi se na proces čišćenja ili modifikacije podataka koji ulaze u sistem kako bi se uklonili potencijalno štetni sadržaji. Obuhvata uklanjanje specijalnih karaktera, kodiranje URL parametara ili upotrebu tehnika za normalizaciju podataka.

```php
<?php

$username = $mysqli->real_escape_string($_POST['username']);
$query = "SELECT * FROM users WHERE username = '$username'";

?>
```

Listing 10 - PHP priprema korisničkog imena za ubacivanje u MySQL upit

Cilj sanitacije je da se podaci pripreme za obradu ili prikaz na takav način da ne mogu da izazovu sigurnosnu pretnju ili grešku u radu aplikacije. U tu svrhu se obično koriste sledeće tehnike:

- *Uklanjanje* nepoželjnih karaktera iz stringa se vrši primenom regularnih izraza.
- *Eskapiranje specijalnih karaktera* (engl. *HTML Escaping*) podrazumeva konverziju specijalnih HTML karaktera u HTML entitete, npr. karakter < se menja u *<*.

- *Eskapiranje SQL upita (engl. SQL Escaping*) se koristi za sprečavanje SQL injection napada tako što se specijalni karakteri u SQL upitima tretiraju na takav način da ne mogu biti interpretirani kao deo SQL komande, listing 10.
- *Kodiranje URL parametara* (engl. *URL Encoding*) pretvara specijalne karaktere u URL-u u odgovarajuće *escape* sekvence, npr. razmak postaje *%20*.
- *Base64 enkodiranje* se koristi za kodiranje binarnih podataka kao tekstualnih stringova, čime se osigurava da podaci budu bezbedni za transport preko različitih protokola.
- *Normalizacija podataka* podrazumeva *a) trimovanje stringa*, odnosno uklanjanje praznih karaktera (engl. *whitespace characters*) sa početka i kraja stringa, i *b)* konverziju malih i velikih slova čime se osigurava doslednost u podacima, npr. pretvaranje svih unosa u mala slova radi upoređivanja stringova.

4.3.2 Validacija podataka

Validacija podataka je proces provere da li podaci ispunjavaju određene kriterijume pre nego što budu prihvaćeni i obrađeni. Cilj validacije je da osigura da su podaci tačni, potpuni i u skladu sa očekivanim formatom.

Validacija podrazumeva provere da li su obavezna polja popunjena, da li su vrednosti unutar očekivanog opsega, i da li su podaci u očekivanom tipu (celobrojna vrednost, string, ...) i formatu (email adresa, broj telefona, ...).

Tehnike validacije podataka obuhvataju sledeće korake:

- *Provera formata stringa*, odnosno provera da li je sadržaj stringa u skladu sa određenim šablonom, kao što su email adrese, brojevi telefona ili postanski brojevi, vrši se primenom regularnih izraza.
- *Provera opsega i tipa* podrazumeva proveru da li su numeričke vrednosti unutar očekivanog opsega, npr. starost između 1 i 120 godina, kao i proveru tipa podataka čime se osigurava da su podaci odgovarajuće vrste, npr. stringovi, brojevi, datumi i sl.

- *Provera obaveznih polja o*sigurava da su neophodne vrednosti obavezno unete pre dalje obrade.
- *Sofisticirane validacije* obuhvataju *konzistentnost podatak*a, npr. da li je datum početka pre datuma završetka, i r*eferencijalni integritet* kojim se proverava da li podaci odgovaraju postojećim zapisima u drugim tabelama ili sistemima, kao što je npr. *ID* korisnika koji postoji u sistemu.

Validacija podataka može da se vrši na klijentskoj strani putem JavaScript-a, čime se unapređuje korisničko iskustvo i smanjuje broj zahteva koji se šalju ka veb serveru. Međutim, oslanjanje isključivo na ovakav vid validacije predstavlja veliki bezbednosni propust, zato što potencijalni napadač može lako da zaobiđe ovakve mehanizme.

Validacija podataka na strani servera ključna za bezbednost aplikacije, zato što omogućava proveru podataka u slučajevima ako klijentska strana nije uspešno validirala podatke.

Popularni radni okviri kao što su *Django* (Python), Laravel (*PHP*) ili Express (*Node.js*), često imaju ugrađene mehanizme za sanitaciju i validaciju podataka. Osim toga, za većinu popularnih programskih jezika postoje namenske biblioteke koje mogu da se uključe u projekat, kao što su *Cerberus* (*Python*), *Validator* (*PHP*) ili *Joi* (*Node.js*).

4.4 Sigurnosni testovi

Sigurnosni testovi veb aplikacije igraju ključnu ulogu u zaštiti od različitih pretnji i napada. Koriste se za identifikaciju i evaluaciju ranjivosti koje bi mogle omogućiti neovlašćeni pristup, manipulaciju podacima ili oštećenje sistema.

Penetracijski testovi (engl. *Pen testing*) simuliraju stvarne napade na aplikaciju. Testeri koriste različite tehnike i alate kako bi otkrili ranjivosti koje bi mogle biti iskorišćene od strane napadača. Proces uključuje planiranje, skeniranje, eksploataciju ranjivosti i izveštavanje o nalazima. Penetracijski testovi mogu da se sprovode u tri scenarija, tzv. *Black Box* kada tester

nema nikakve informacije o aplikaciji osim javno dostupnih, *White Box* kada tester ima na raspolaganju sve detalje o aplikaciji, uključujući izvorni kod i konfiguracije, i *Gray Box* koji predstavlja kombinaciju prethodna dva scenarija, a tester ima delimične informacije. Pen testovi mogu da budu

- *Manuelni.* Ručno testiranje aplikacije, često u kombinaciji sa automatizovanim alatima.
- *Automatizovani.* Korišćenje alata za automatsko pretraživanje ranjivosti. Proces je znatno brži, ali ostaje rizik da možda neće biti otkrivene sve ranjivosti.

Za skeniranje ranjivosti se koriste specijalizovani alati za automatsko pretraživanje aplikacija. Ovi alati analiziraju softverske komponente i konfiguracije kako bi otkrili poznate slabosti.. Alati poput *Nessus*, *OpenVAS* i *Qualys* nude različite pristupe i mogućnosti za automatsko skeniranje.

- *Nessus.* Jedan od najpoznatijih skenera ranjivosti, pruža detaljne izveštaje o sigurnosnim slabostima u mrežnim i aplikacijskim komponentama.
- *OpenVAS.* Open-source alat za skeniranje ranjivosti koji se često koristi kao alternativa komercijalnim rešenjima.
- *Qualys*. Nudi sveobuhvatno rešenje za skeniranje ranjivosti sa pristupom baziranim u oblaku, uz odgovarajuće analize i izveštaje.
- *Rapid7 Nexpose.* Omogućava uvid u sigurnosne rizike i daje preporuke za poboljšanje zaštite.

Da bi integritet veb aplikacije bio sačuvan od potencijalnih pretnji, neophodna je primena oba pristupa, zajedno sa neprekidnim praćenjem i ažuriranjem sigurnosnih mera.

4.5 Zaštita od SQL Injekcija

SQL injekcija (eng. *SQL injection*) je vrsta napada na veb aplikaciju u kome napadač pokušava da injektiranjem podataka, izvorni SQL upit transformiše u maliciozni i neovlašćeno pristupi podacima. Postoje različite tehnike

zaštite od SQL injekcija, od kojih se najčešće koriste sanitacija i validacija ulaznih podataka, pripremljeni i parametrizovani upiti.

```
$user = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT *
        FROM korisnici
        WHERE
          korisnicko_ime = '{$user}'
          AND lozinka = '{$password}'";
```

Listing 11 - Upit za selektovanje korisničkog naloga, PHP/MySQL

Na listingu 11 je prikazan ispravan, ali loše formiran upit za selektovanje podataka o korisničkom nalogu iz tabele *korisnici*. Ukoliko bi se u njega direktno ubacili korisničko ime i lozinka koje je poslao korisnik putem veb obrasca za prijavu, SQL server će izvršiti upit i vratiti odgovarajuću belešku ukoliko postoji nalog sa takvim podacima, što je i očekivano ponašanje.

```
SELECT *
FROM korisnici
WHERE
  korisnicko_ime = '' OR '1'='1'
  AND lozinka = '' OR '1'='1'
```

Listing 12 - SQL upit sa injekcijom

Problem nastaje ukoliko se u obrazac unesu stringovi koji mogu da poremete prirodu SQL upita. Na listingu 12 je prikazan isti upit kada su kao korisničko ime i lozinka uneti stringovi *' OR '1'='1*. Upit je i dalje ispravan, ali će kao rezultat vratiti sve beleške iz tabele *korisnici*.

Na listingu 13 se nalazi primer PHP koda koji koristi MySQLi ekstenziju i pripremljene upite (*prepare*, *bind_param*, linije 18 i 24), što osigurava da se korisnički unos ne može modifikovati i izvesti kao deo SQL upita.

Na listingu 14 je prikazan alternativni pristup. Upit je ranjiv na SQL injekcije, ali uz pažljivu obradu dobijenih rezultata njihov efekat se eleminiše. U ovom slučaju, injekcija može da bude string *' OR 1=1 --*, sa sledećim efektom:

```php
<?php
  // Povezivanje sa bazom podataka
  $mysqli = new mysqli(
                  'localhost',
                  'korisnik',
                  'lozinka',
                  'baza'
                );
  // Provera povezivanja
  if ($mysqli->connect_error)
    die("Povezivanje neuspešno: " . $mysqli->connect_error);
  // Priprema SQL upita
  $sql = "SELECT *
            FROM korisnici
            WHERE
              korisnicko_ime = ?
              AND lozinka = ?";
  $stmt = $mysqli->prepare($sql);
  if (!$stmt)
      die("Priprema upita neuspešna: " . $mysqli->error);
  // Dodeljivanje parametara
  $user = $_POST['username'];
  $password = $_POST['password'];
  $stmt->bind_param('ss', $user, $password);
  // Izvršenje upita
  $stmt->execute();
  // Preuzimanje rezultata
  $result = $stmt->get_result();
  while ($row = $result->fetch_assoc()) {
      // Obrada rezultata
  }
  // Zatvaranje izjave i konekciju
  $stmt->close();
  $mysqli->close();
?>
```

Listing 13 - Upotreba pripremljenih upita, PHP/MySQL

- Navodnik zatvara string u upitu
- OR 1=1 je uslov koji je uvek tačan i eliminiše ograničenje za korisničko ime
- -- (dupla srednja crta) označava komentar u SQL-u, čime se iz upita eleminiše ograničenje LIMIT 1

```
$sql = "SELECT *
        FROM korisnici
        WHERE korisnicko_ime = "$username"
        LIMIT 1"
```

Listing 14 - MySQL upit za selektovanje korisničkog naloga

U praksi, ovo bi omogućilo napadaču da dobije podatke o svim korisnicima iz tabele *korisnici*, listing 15.

```
1 $result = $stmt->get_result();
2 if ($result) {
3   while ($row = $result->fetch_assoc()) {
4     // Obrada podataka
5     var_dump($row);
6   }
7 } else
8   echo "Nema korisnika sa unetim korisničkim imenom.";
```

Listing 15 - Očitavanje rezultata SQL upita, injekcija ima efekta

Međutim, ukoliko bi se eleminisala petlja, linije 3-6 na listingu 15 i ubacila provera dobijenog rezultata, linije 3-6 na listingu 16, SQL injekcija gubi efekat, a aplikacija zadržava kontrolu nad podacima.

```
1 $result = $stmt->get_result();
2 if ($result) {
3   if ($row['korisnicko_ime'] == $username) {
4     // Obrada podataka
5     var_dump($row);
6   } else
7     echo "Nema korisnika sa unetim korisničkim imenom.";
```

Listing 16 - Očitavanje rezultata SQL upita, injekcija nema efekta

4.6 Šifrovanje podataka

Šifrovanje podataka je ključna tehnika za njihovu zaštitu na serveru (u mirovanju) i prilikom prenosa na relaciji server-korisnik i obrnuto (u tranzitu).

Šifrovanje podataka u mirovanju obezbeđuje da podaci uskladišteni na serverima budu zaštićeni od neovlašćenog pristupa. Ovo je posebno važno za osetljive informacije kao što su lozinke, brojevi kreditnih kartica, i lični identifikacioni podaci, poslovni ugovori, podaci o tehnološkim procesima i sl. Tehnike koje se primenjuju u veb aplikacijama za ovu svrhu uključuju različite metode i pristupe, svaka sa specifičnim karakteristikama, prednostima i nedostacima.

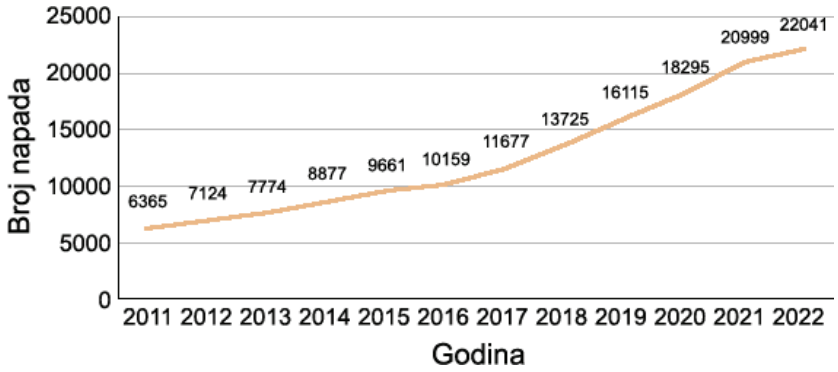
Šifrovanje podataka u tranzitu obezbeđuje da podaci koji se prenose između korisnika i servera budu zaštićeni od neovlašćenog prisluškivanja, što se obično postiže korišćenjem SSL/TLS protokola.

Iako je ovo ključni aspekt zaštite informacija, prilikom projektovanja i razvoja IS moraju da se uzmu u obzir i neizbežni nedostaci kako bi se došlo do optimalnog rešenja uz minimalan uticaj na performanse i funkcionalnost sistema:

- Procesi šifrovanja i dešifrovanja mogu značajno da utiču na performanse sistema jer zahtevaju dodatne resurse, što može usporiti rad aplikacije i povećati vreme obrade podataka.
- Upravljanje šifrovanim ključevima je složeno, jer ključevi moraju biti sigurno generisani, distribuirani, pohranjeni i ažurirani kako bi se izbeglo njihovo kompromitovanje. Kompromitovanje šifrovanog ključa može ugroziti sve podatke zaštićene tim ključem, dok različiti zakonski i regulatorni zahtevi mogu dodatno otežati usklađivanje sa standardima.
- Usled implementacije šifrovanja, struktura IS postaje složenija, što dovodi do povećanja troškova razvoja i održavanja.

4.7 Zaštita od Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) napadi se dešavaju kada napadač ubaci zlonamerni, obično *JavaScript* kod u sadržaj napadnutog veb sajta.



Slika 36 - Trend porasta broja XSS napada

XSS propust može da se iskoristi kako bi se zaobišli sigurnosni mehanizmi veb aplikacije, a time i izgubila kontrola nad istom. Najčešće se koriste kako bi se ukrale informacije korisnika koji pristupaju sajtu. Broj XSS napada se povećava, slika 36, sve češće kroz posebno kreirane XML dokumente [6]. Zaštita od XSS napada podrazumeva validaciju svih ulaznih podataka, kao i upotrebu bezbednih metoda za prikazivanje dinamičkog sadržaja. S obzirom da napadi postaju sve složeniji, sve češće se za njihovu detekciju primenjuju konvolucijske neuronske mreže i AI. Takođe, uvođenje polise bezbednosti sadržaja (engl. *Content Security Policy*, CSP) može pomoći u sprečavanju izvršenja neželjenog *JavaScript* koda.

4.8 Redovno ažuriranje i zakrpe

Redovno ažuriranje softverskih komponenti i implementacija sigurnosnih zakrpa je od suštinskog značaja za održavanje bezbednosti IS. Ovo uključuje ažuriranje operativnog sistema, veb servera, baza podataka, kao i svih biblioteka i radnih okvira koji se koriste u razvoju IS. Nepažnja i neažurnost u implementaciji zakrpa može izazvati ranjivost aplikacije koju potom napadači mogu da iskoriste.

4.9 Monitoring i detekcija pretnji

Monitoring i detekcija pretnji uključuju praćenje aktivnosti na serveru i u aplikaciji kako bi se identifikovali sumnjivi obrasci i potencijalni napadi. Ovaj proces može uključivati logovanje svih ključnih događaja, korišćenje sistema za detekciju upada (IDS) i analiziranje logova radi otkrivanja anomalija i pravovremenog reagovanja na bezbednosne incidente.

INFORMACIONI SISTEMI

5. INFORMACIONI SISTEMI

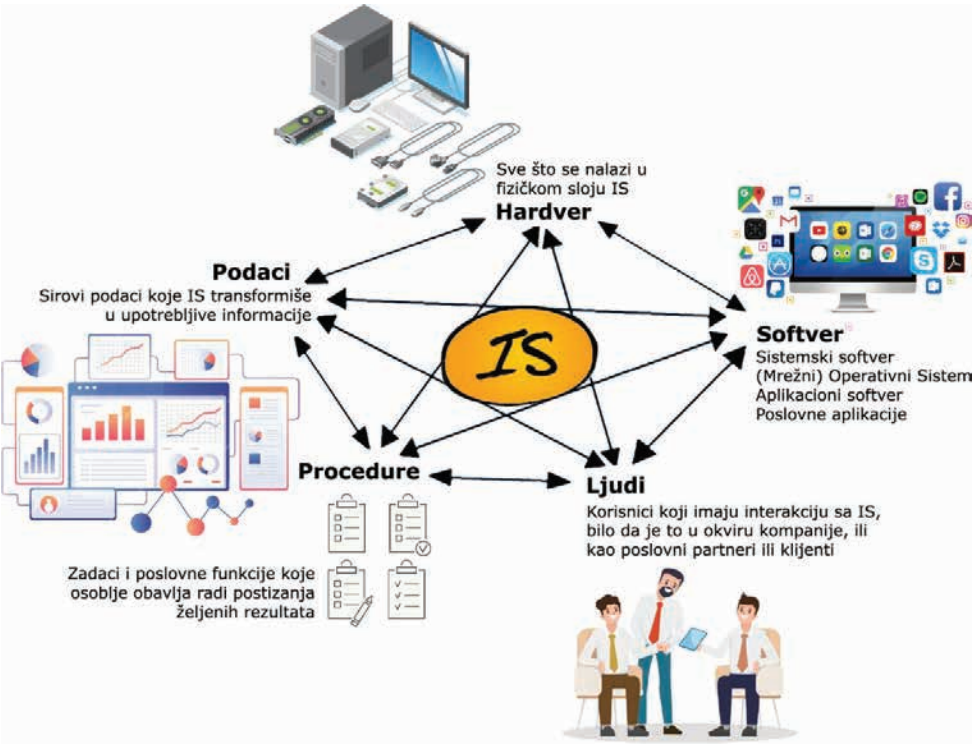
IS je skup povezanih komponenti koje rade zajedno kako bi prikupile, obradile, skladištile i distribuirale informacije radi podrške procesima donošenja odluka, upravljanja i izvršavanja poslovnih aktivnosti unutar organizacije, sa ciljem unapređenja procesa donošenja odluka i efikasnijeg upravljanja organizacijom.

5.1 Komponente informacionog sistema

Komponente IS su različiti delovi sistema koji zajedno čine celinu i omogućavaju efikasno funkcionisanje sistema, slika 37. Svaka komponenta ima svoju ulogu i funkciju u procesu prikupljanja, obrade, skladištenja, analize i distribucije informacija. Osnovne komponente IS obuhvataju:

- *Hardver* su fizičke komponente IS u koje se ubrajaju računari, serveri, mrežna oprema, skladišta podataka i drugi uređaji potrebni za obradu i skladištenje informacija.
- *Softver* čine programi i aplikacije koje koriste korisnici IS kako bi obradili, analizirali i manipulisali podacima. U ovu grupu se mogu uvrstiti operativni sistemi, baze podataka, aplikacije za analizu podataka, alati za upravljanje sadržajem i druge softverske komponente.
- *Podaci* su sirove informacije koje se prikupljaju, skladište i obrađuju unutar informacionog sistema. To mogu biti podaci o transakcijama, korisnicima, proizvodima, finansijama i drugim aspektima poslovanja.
- *Procesi i procedure* koji se koriste za prikupljanje, obradu, analizu i distribuciju informacija unutar organizacije. To uključuje radne tokove, procedure za donošenje odluka, rutine za bezbednost podataka i druge operativne procese.

- *Ljudi.* Osobe koje koriste informacioni sistem za obavljanje svojih poslovnih aktivnosti. To mogu biti menadžeri, zaposleni, korisnici sistema ili IT stručnjaci koji održavaju i podržavaju informacioni sistem.



Slika 37 - Komponente IS

5.2 Klasifikacija informacionih sistema

IS se mogu klasifikovati na različite načine, obično na osnovu različitih kriterijuma kao što su namena, veličina, funkcionalnost ili tehnologija koju koriste. Prema nameni, IS mogu da se podele na sledeći način:.

- *Operativni IS* se koriste za podršku svakodnevnim operacijama organizacije, kao što su upravljanje zalihama ili obrada transakcija.
- *Upravljački IS* pružaju informacije menadžmentu za donošenje odluka, kao što su sistemi za upravljanje odnosima s klijentima (engl.

Client/Customer relationship management, CRM) ili za upravljanje skladištima podataka (engl. *Data warehouse*, DWH).
- *Strateški IS* se koriste za podršku dugoročnom strateškom planiranju, i u ovu grupu mogu da se uvrste sistemi za analizu tržišta ili sistemi za upravljanje odnosima sa dobavljačima.

Prema veličini, IS mogu da se podele u tri osnovne grupe:

- *Mikro IS* predstavljaju male poslovne sisteme, koji se koriste na pojedinačnim računarima, ili u manjim računarskim mrežama.
- *Mezo IS* se odnose na sisteme srednje veličine, koji mogu da podrže nekoliko stotina korisnika, poput sistemska za upravljanje resursima u srednjim preduzećima.
- *Makro IS* uključuju velike sisteme koji podržavaju hiljade korisnika, kao što su korporativne mreže, bankarski sistemi ili državni IS.

Prema tehnologiji IS mogu da budu *centralizovani* ukoliko se podaci i mehanizam obrade nalaze na jednom centralnom mestu, ili distribuirani gde su podaci i obrada podataka raspoređeni na više lokacija, uz primenu mrežne tehnologije.

Prema funkciji, IS mogu da se podele na *finansijske IS* gde je fokus na finansijskim transakcijama, računovodstvu i kontroli troškova, IS za *upravljanje ljudskim resursima* (menadžment osoblja, plate, beneficije i sl) i *proizvodne IS* kojio bezbeđuju podršku za proizvodnju kroz planiranje proizvodnje i upravljanje zalihama.

5.3 Tipovi informacionih sistema

5.3.1 Poslovni informacioni sistemi i upravljanje resursima

Poslovno informacioni sistem (PIS) i *sistemi za upravljanje resursima* (engl. *Enterprise Resource Planning*, ERP) su IS koji se koriste u organizacijama. PIS predstavlja širi koncept koji obuhvata sve informacione tehnologije (IT) i procese koji podržavaju poslovanje, dok je ERP specifičan tip PIS-a sa fokusom na integraciju i upravljanje različitim resursima u organizaciji.

PIS obuhvata širi spektar informacionih tehnologija i procesa usmerenih ka podršci poslovnim funkcijama organizacije. Fokus je na obradi, generisanju, skladištenju i distribuciji informacija koje podržavaju različite poslovne procese. PIS može da obuhvati različite vrste softverskih sistema i tehnologija, uključujući one za upravljanje bazama podataka, dokumentima, analizu podataka i druge, a sa ciljem da se unaprede efikasnost, donošenje kvalitetnijih poslovnih odluka i podrška operativnim procesima u organizaciji.

Cilj ERP-a je da se efikasno upravlja resursima poput novca, materijala, ljudi, opreme i procesa. Fokus je na integraciji i automatizaciji poslovnih procesa unutar organizacije. Osnovna svrha ERP-a je da kroz svoje module integriše različite funkcije i organizacione jedinice (OJ), kao što su finansije, proizvodnja, nabavka, prodaja, ljudski resursi i druge u jedinstvenu platformu čime će omogućiti smanjenje redundentnosti podataka, povećanu transparentnost poslovanja i podršku donošenju odluka zasnovanih na realnim podacima.

5.3.2 Sportski informacioni sistemi

Sportski informacioni sistemi (SIS) se koriste za prikupljanje, obradu, skladištenje i distribuciju informacija vezanih za sportske događaje i aktivnosti. Generalno mogu da se podele u četiri osnovne kategorije [7]:

- *Sistemi za praćenje performansi.* Ovi sistemi prate i analiziraju performanse sportista tokom treninga i takmičenja koristeći različite senzore i tehnologije.
- *Sistemi za upravljanje timovima.* Uključuju alate za planiranje treninga, analizu statistike, i komunikaciju unutar tima.
- *Sistemi za analizu i izveštavanje.* Ovi sistemi obrađuju podatke i generišu izveštaje o rezultatima i statistikama za sportske organizacije i medije.
- *Sistemi za podršku donošenju odluka.* Pomažu trenerima i menadžerima u donošenju odluka na osnovu analize podataka i predikcija.

Obuhvataju širok spektar funkcionalnosti, uključujući praćenje rezultata sportskih takmičenja, vođenje statistike igrača i timova, upravljanje rasporedima događaja, omogućavaju komunikaciju između učesnika i gledalaca, kao i pružanje analitičkih alata za procenu performansi.

5.3.3 Ekspertski sistemi

Ekspertski informacioni sistemi integrišu ekspertsko znanje i tehnologiju AI kako bi pružili stručne savete ili rešenja u određenoj oblasti. Ovi sistemi koriste bazu znanja koja obuhvata informacije i pravila koja su specifična za datu oblast ili problem, na osnovu kojih mogu da donose zaključke, prave predikcije, ili daju preporuke, slično kako bi to uradila osoba koja je ekspert u datoj oblasti. Ova vrsta sistema se često koristi u složenim domenima poput medicine, prava, inženjeringa ili finansija kako bi se identifikovali problemi, pružila stručna podrška korisnicima ili podržalo donošenje odluka.

5.3.4 Veb pretraživači

Veb pretraživači, kao što su npr. *Google*, *Bing* ili *DuckDuckGo*, omogućavaju korisnicima da pretražuju internet kako bi pronašli informacije koje im trebaju. Ovi sistemi koriste algoritme za pretraživanje, ekstrakciju i indeksiranje veb sadržaja, uključujući faktore poput relevantnosti, popularnosti i autoriteta, kao i tehnike analize teksta, veza i strukture veb stranica kako bi pružili korisnicima relevantne rezultate pretrage. U proces mogu da budu uključeni i lični podaci korisnika, kao i istorija pretraživanja čime se obezbeđuju personalizovani rezultati pretrage.

Osim pretraživača, *Google* nudi i alate za analizu podataka poput *Google Analytics* platforme, slika 38, koja omogućava vlasnicima veb sajtova i aplikacija da prate i analiziraju ponašanje posetilaca na svojim sajtovima, uključujući statistiku posete pojedinim stranicama, vreme provedeno na sajtu, geo lokaciju posetilaca, polnu i starosnu strukturu posetilaca. *Google* takođe nudi i druge alate za analizu, kao što su *Google Trends* za analizu informacija o popularnosti pretraga, kao i *Google Keywords Planner* koji pomaže u istraživanju ključnih reči i planiranju marketinških kampanja.



Slika 38 - *Google Analytics*, izveštaj o saobraćaju na veb sajtu
Broj posetilaca, prosečno vreme i zarada za izabrani vremenski period (gore) i ostvareni saobraćaj u poslednjih 30 minuta (dole)

Bing, internet pretraživač kompanije *Microsoft*, takođe nudi analitičke servise pod imenom *Bing Webmaster Tools*. Ova platforma omogućava vlasnicima veb stranica da dobiju uvid u to kako *Bing* indeksira njihov sadržaj, kako se isti prikazuje na rezultatima pretrage, kao i uvid u informacije o saobraćaju na sajtu, linkovima i mnogim drugim aspektima koji su važni za optimizacija za pretraživače (engl. *Search Engine Optimization*, SEO) i unapređenje performansi veb sajta.

5.3.5 Geografski informacioni sistemi

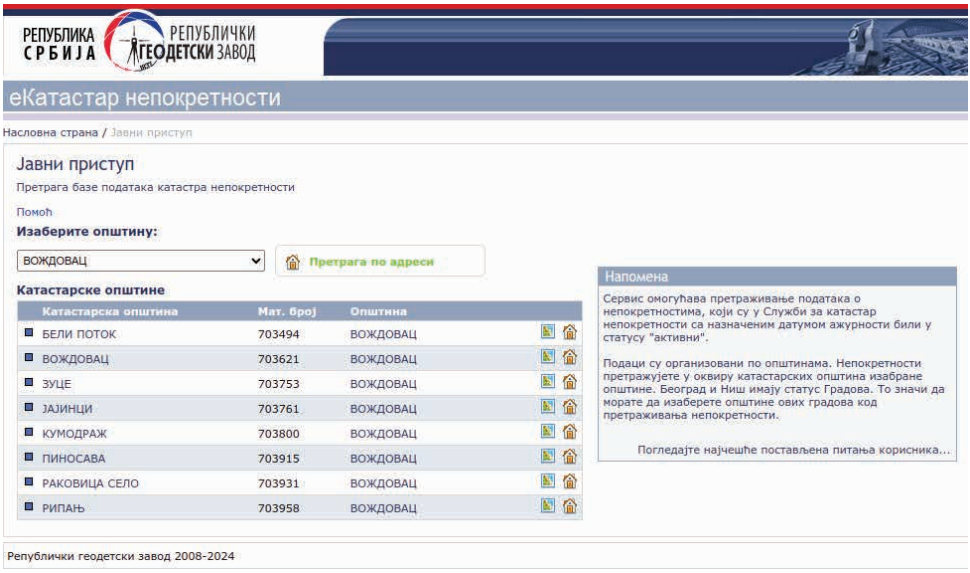
Geografski informacioni sistemi (engl. *Geographic Information System*, GIS) su softverski alati i tehnologije koje se koriste za prikupljanje, skladištenje, analizu, manipulaciju i vizualizaciju prostornih i geografskih podataka, a korisnicima omogućavaju da razumeju i interpretiraju geografske informacije na efikasan način.

Podaci koji mogu da se prikupljaju i čuvaju u okviru GIS-a mogu da budu geografske koordinate, visinske informacije, granice teritorija, topografski podaci, satelitski snimci, i sl. Jedna od ključnih funkcionalnosti GIS-a je mogućnost analize prostornih podataka, npr. putem prostornih upita, analize rastojanja, analize terena, geostatistike i sl. kako bi se otkrili obrasci, trendovi i međusobni odnosi. GIS takođe omogućava korisnicima manipulaciju prostornim podacima na različite načine, uključujući preklapanje, spajanje, razdvajanje, transformaciju i oblikovanje podataka u skladu sa konkretnim potrebama. U Republici Srbiji, ovakav tip IS koriste mnoge vladine i nevladine organizacije.

- *Katastar* koristi GIS za upravljanje informacijama o katastarskim područjima, parcelama, vlasničkim pravima i drugim zemljišnim podacima.
- *Gradske vlade* koriste GIS za upravljanje zemljištem, planiranje infrastrukture, analizu saobraćaja i prostorno planiranje.
- *Telekomunikacione kompanije* koriste GIS za planiranje mrežne infrastrukture i optimizaciju pokrivenosti teritorije signalom.
- U *vodoprivredi* GIS se koristi za upravljanje vodnim resursima, praćenje kvaliteta vode, analizu plavnih područja i planiranje navodnjavanja.
- U *geologiji*, GIS ima primenu prilikom kartiranja i analize geoloških podataka, istraživanja rudnih nalazišta i upravljanja eksploatacijom mineralnih resursa.
- U *zaštiti životne sredine*, GIS može da se koristi za praćenje zagađenja vazduha i vode, upravljanje otpadom, identifikaciju ekoloških problema i planiranje mera zaštite životne sredine.



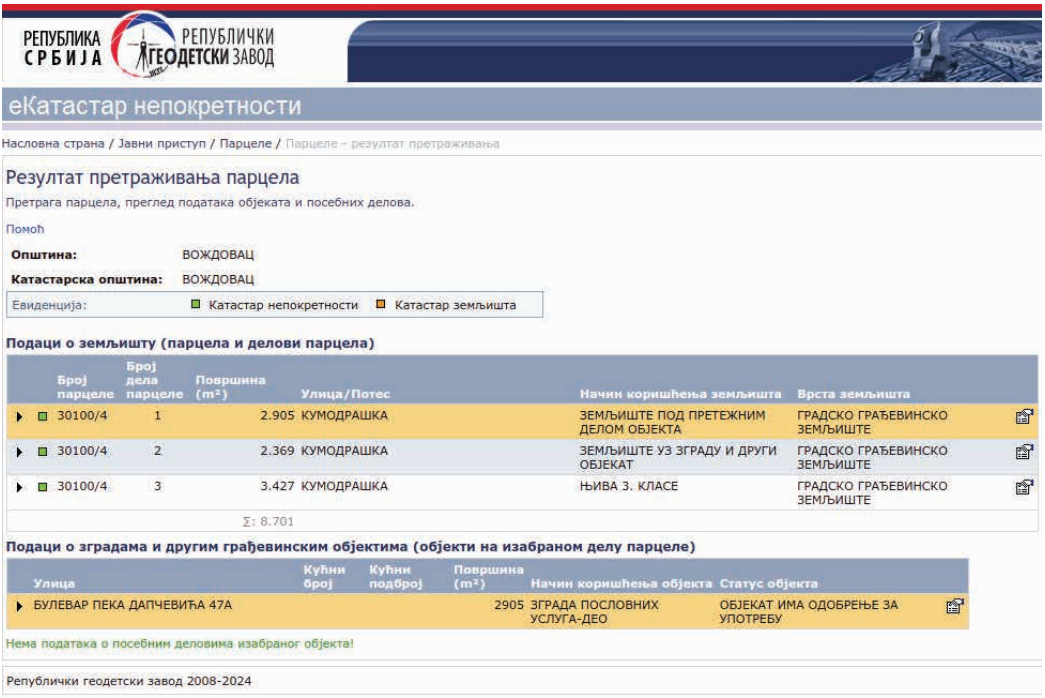
Slika 39 - GIS, satelitski snimak dela teritorije
sa oznakam katastarskih parcela, a3.geosrbija.rs



Slika 40 - GIS, katastar nepokretnosti, izbor opštine i katastarske opštine



Slika 41 - GIS, katastar nepokretnosti, unos broja katastarske parcele



Slika 42 - GIS, katastar nepokretnosti
podaci o zemljištu i objektima na izabranoj katastarskoj parceli

GIS pruža mogućnost vizualizacije prostornih podataka putem mapa, grafikona, dijagrama i drugih vizualnih elemenata kako bi se olakšalo razumevanje i interpretacija informacija.

Geosrbija[1)] je digitalna platforma koja omogućava vizuelni prikaz geoprostornih podataka Republike Srbije, pristup katalogu metapodataka, kao i niz digitalnih usluga. Na slici 39 je prikazan javno dostupni satelitski snimak dela teritorije Grada Beograda sa oznakama katastarskih parcela, dobijen putem aplikacije na veb adresi *a3.geosrbija.rs*. Osim osnovnih operacija kao što su merenje udaljenosti ili površine, aplikacija omogućava pristup i drugim funkcionalnostima.

Republički geodetski zavod omogućava uvid u katastar nepokretnosti putem javno dostupne veb aplikacije [2)], a podaci su organizovani po katastarskim opštinama, slika 40. Nakon izbora opštine i katastarske opštine, u veb obrazac se unosi broj katastarske parcele, slika 41, ili ulica i broj u slučaju pretrage po adresi. Nakon toga, veb aplikacija će na veb stranici prikazati odgovarajući list nepokretnosti, slika 42. U oba slučaja, navedene funkcionalnosti su javno dostupne. Pored njih, specijalizovanim korisnicima su na raspolaganju dodatni setovi funkcionalnost, ali za pristup je neophodan odgovarajući korisnički nalog.

Nasuprot navedenim primerima, platforma *Google Maps* nije GIS iako poseduje relevantne karakteristike. Google Maps pruža korisnicima mogućnost da pregledaju prostorne podatke na mapama putem veb interfejsa ili mobilne aplikacije, ali ne pruža mogućnost analize i manipulacije prostornih podataka.

5.3.6 Globalni informacioni sistemi

Globalni informacioni sistemi (engl. *Global Information System*, GIS) [3)] omogućavaju prikupljanje, skladištenje, analizu i distribuciju informacija na

[1] https://www.geosrbija.rs
[2] https://katastar.rgz.gov.rs/eKatastarPublic/publicaccess.aspx
[3] PAŽNJA! Ista skraćenica (GIS) se koristi i za geografske i za globalne IS.

globalnom nivou. Ova vrsta sistema obuhvata širok spektar tehnologija, resursa i infrastrukture koji omogućavaju komunikaciju i razmenu podataka širom sveta. IS iz ove grupe se koriste u različitim oblastima.

Svetska meteorološka organizacija (engl. *World Meteorological Organization,* WMO) koristi globalni IS za praćenje i analizu vremenskih uslova, klimatskih promena, prirodnih katastrofa i drugih meteoroloških fenomena radi upravljanja rizicima i obezbeđivanja sigurnosti meteoroloških podataka širom sveta.

Svetska zdravstvena organizacija (engl. *World Health Organization*, WHO) koristi globalne IS za praćenje širenja bolesti, vakcinacije, zdravstvene infrastrukture i drugih pokazatelja širom sveta. Ovi sistemi igraju ključnu ulogu u brzom reagovanju na pandemije i druge globalne zdravstvene izazove.

Globalni trgovinski IS omogućavaju praćenje i analizu međunarodne trgovine, uključujući robu, usluge, valute i finansijske tokove širom sveta. Važni su za upravljanje globalnim ekonomskim procesima, trgovinskim sporazumima, tarifama i politikama.

Internet obuhvata kompleksnu mrežu komunikacionih kanala, servera, uređaja i softverskih aplikacija koje omogućavaju razmenu informacija širom sveta. Internet se može smatrati GIS-om, zato što obuhvata ključne karakteristike kao što su globalni pristup, skladištenje i pristup velikim količinama podataka, komunikacija korisnika bez obzira na geografsku lokaciju, distribucija informacija, i konačno integracija različitih IS, aplikacija i servisa.

U grupu globalnih IS može da se svrsta i SAP, koji može da integriše različite poslovne procese i operacije koje se odvijaju u različitim regionima i zemljama, što dalje omogućava organizacijama da centralizuju upravljanje, optimizuju operativne parametre i olakšava donošenje odluka.

5.4 Automatizacija informacionih sistema

Neautomatizovani IS obuhvataju tradicionalne papirne evidencije ili tabele i dokumente koje korisnici moraju ručno da održavaju. U njima se ne koriste sofisticirane tehnologije ili alati za automatizaciju procesa i analize podataka, i kao takvi obično zahtevaju veći stepen ručnog rada i intervencije korisnika u obradi podataka i donošenju odluka, uz povećan rizik od pojave greške i uticaja ljudskog faktora na rezultate.

Iako se u savremenom društvu organizacije uglavnom oslanjaju na tehnologiju za unapređenje poslovnih procesa, neautomatizovani IS nisu u potpunosti potisnuti iz upotrebe, naročito u manjim organizacijama ili u specifičnim industrijama gde automatizacija nije praktična ili ekonomski opravdana.

Kroz automatizaciju, IS mogu da unapređuju poslovanje, omoguće efikasniju komunikaciju, analizu podataka i personalizaciju, čime doprinose povećanju produktivnosti pojedinaca i efikasnijem radu organizacije u celini.

Upotrebom različitih aplikacija i softverskih alata, IS automatizuju rutinske zadatke kao što su obrada narudžbina, fakturisanje ili upravljanje zalihama. Zahvaljujući ovakvoj automatizaciji, zaposleni ne moraju da se bave repetitivnim poslovima, čime se smanjuje mogućnost pojave grešaka izazvanih ljudskim faktorom. Osim toga, elektronska dokumentacija zamenjuje tradicionalnu papirnu dokumentaciju, omogućavajući jednostavnije i efikasnije skladištenje, pretragu i deljenje dokumenata.

Kroz alate za kolaboraciju i komunikaciju, IS olakšavaju interakciju među zaposlenima, bez obzira na njihovu geografsku lokaciju. Ovo podrazumeva korišćenje elektronske pošte, instant poruka, zajedničkih radnih prostora i video konferencija kako bi se olakšala saradnja i razmena informacija.

Automatizacija analize podataka omogućava organizacijama da identifikuju obrasce, trendove i ključne pokazatelje performansi, što je posebno korisno za donošenje odluka i optimizaciju poslovanja.

Napokon, IS pružaju mogućnost personalizacije kroz prilagođavanje korisničkog interfejsa, funkcionalnosti i podešavanja kako bi se poboljšala efikasnost, a time i zadovoljstvo korisnika.

5.5 Namena informacionih sistema

5.5.1 Upravljački informacioni sistemi

Upravljački informacioni sistemi (UIS) se koriste za podršku procesu donošenja odluka na upravljačkom nivou u organizaciji, a osnovni zadatak im je da menadžmentu pruže relevantne, precizne i pravovremene informacije i izveštaje koji su neophodni za planiranje, kontrolu i vođenje organizacije. Kao primeri UIS mogu da se navedu:

- *Sistemi za upravljanje performansama* prate ključne pokazatelje performansi (KPI) organizacije i pružaju menadžmentu informacije o trenutnom stanju organizacije u odnosu na zadate ciljeve i strategije.
- *Sistemi za poslovnu analitiku* koriste napredne tehnike analize podataka kako bi identifikovali obrasce, trendove i ključne informacije koje su od značaja za upravljanje organizacijom.
- *Sistemi za upravljanje znanjem* omogućavaju organizacijama da efikasno upravljaju znanjem i informacijama unutar organizacije, kao i da olakšaju deljenje znanja i saradnju među zaposlenima.
- *Sistemi za upravljanje odnosima sa klijentima* (CRM) prate interakciju sa klijentima i pružaju informacije o njihovim potrebama, preferencijama i istoriji kupovina kako bi se unapredili nivo usluge i zadovoljstvo kupaca.

5.5.2 Izvršni informacioni sistemi

Svrha *izvršnih IS* (engl. *Executive Information System*, EIS) jeste da obezbede alate i resurse koji će da podrže donošenje odluka na najvišem nivou upravljanja. Izvršnim direktorima i rukovodstvu organizacije će omogućiti pristup ključnim informacijama iz različitih izvora na brz i jednostavan način, što ih čini ključnim alatom za efikasno upravljanje organizacijom na strateškom nivou.

EIS obično kombinuju podatke iz unutrašnjih izvora, kao što su podaci o finansijama, proizvodnji, prodaji i ljudskim resursima, sa spoljnim izvorima kao što su tržišna istraživanja, konkurencija i makroekonomski trendovi. Ti podaci se obično prikazuju u obliku grafova, tabela, i drugih vizuelnih elemenata koji omogućavaju brzo razumevanje ključnih performansi organizacije. Osim toga, EIS mogu da podržavaju alate za analizu podataka i simulaciju scenarija radi procene različitih poslovnih strategija i njihovih mogućih ishoda.

Iako donose mnoge koristi, implementacija EIS može da bude izuzetno složena i skupa u firmama sa složenom organizacionom strukturom, dok integracija podataka iz različitih izvora predstavlja poseban izazov. Upotreba ovakvih sistema donosi neizbežne sigurnosne rizike, a prekomerno oslanjanje menadžment na EIS u donošenju odluka može da dovede do zavisnosti organizacije od tehnologije.

5.5.3 Sistemi za podršku odlučivanju

Iako EIS i *sistemi za podršku odlučivanju* (engl. *Decision Support System*, DSS) imaju slične ciljeve, razlikuju se u svom fokusu i funkcionalnostima. Fokus DSS je na podršci procesu donošenja operativnih i taktičkih odluka na svim nivoima menadžmenta, pružajući alate i tehnike za analizu podataka i donošenje odluka vezanih za konkretne zadatke ili probleme.

5.6 Informacioni sistemi kao veb aplikacije

IS napravljeni u formi veb aplikacije imaju nekoliko osnovnih prednosti u odnosu na desktop aplikacije.

- *Pristupačnost i dostupnost.* Pristup aplikaciji je preko veb brauzera, sa bilo kog uređaja koji ima internet konekciju. Ovo eliminiše potrebu za instalacijom softvera na svakom korisničkom uređaju i omogućava korisnicima da pristupe aplikaciji sa različitih platformi, bez obzira na arhitekturu hardvera ili operativni sistem (*Windows*, *macOS*, *Linux* i sl.). Razvojni tim razvija aplikaciju koja će raditi na

različitim OS bez potrebe za prilagođavanjem za svaku specifičnu platformu.

- *Lakše održavanje i ažuriranje.* Budući da se veb aplikacije izvršavaju na serveru, a ne na korisničkom uređaju, održavanje i ažuriranje softvera se vrši centralizovano. To znači da su ispravke grešaka ili nove funkcionalnosti odmah dostupne svim korisnicima istovremeno, bez potrebe za ručnim ažuriranjem na svakom korisničkom uređaju.
- *Skalabilnost.* Kapaciteti veb aplikacije mogu lako da se skaliraju u odnosu na broj korisnika ili zahteve za obradu podataka. Povećanje kapaciteta servera može da se relativno jednostavno implementira bez potrebe za izmenama na korisničkim uređajima.
- *Cena* može da varira u skladu sa specifičnim potrebama i uslovima organizacije, ali najčešće predstavlja veliku prednost u odnosu na desktop aplikacije, kako u pogledu razvoja tako i prilikom implementacije i održavanja softvera,

Ove prednosti čine veb aplikacije atraktivnim izborom za organizacije imaju veliki broj korisnika (zaposlenih, klijenata, poslovnih partnera i sl), ili organizacione jedinice na različitim geografskim lokacijama koje žele da olakšaju pristup svojim informacionim sistemima korisnicima širom sveta, smanje troškove održavanja i osiguraju brže i efikasnije ažuriranje i skaliranje svojih aplikacija.

5.7 Trendovi u razvoju informacionih sistema

Na trendove u razvoju IS utiču reflektuju dinamičan tempo tehnoloških inovacija i konkretnih poslovnih potreba. AI i ML su ključne komponente koje su sposobne da automatski uče, prilagođavaju se novim uslovima i delimično samostalno funkcionišu, unapređujući efikasnost poslovnih procesa i procesa donošenja odluka.

U isto vreme, edge computing omogućava obradu podataka i izvršavanje aplikacija na samom rubu mreže, što smanjuje latenciju i omogućava brži odgovor na zahteve korisnika. Uvođenje 5G tehnologije dodatno ubrzava inovacije kao tehnološka osnova za npr. proširenu stvarnost i autonomna vozila. Ova tehnologija je ključna za Internet stvari (engl. *Internet of Things*,

IoT) i uređaje koji generišu velike količine podataka, omogućavajući njihovu efikasnu analizu i upotrebu u realnom vremenu. Blockchain tehnologija takođe igra važnu ulogu u osiguravanju sigurne i transparentne razmene podataka među entitetima, bez potrebe za posrednicima, što je ključno za finansijske transakcije i upravljanje lancem snabdevanja.

Napredna analitika i alati u okviru poslovne inteligencije (engl. *Business Intelligence*, BI) su takođe u vrhu liste aktuelnih trendova, omogućavajući organizacijama da detaljnije analiziraju podatke i donose optimalnije poslovne odluke.

Simulacije, virtuelna i proširena stvarnost predstavljaju moćne alate u izradi modernih IS, omogućavajući napredne tehnološke inovacije, poboljšanje efikasnosti procesa i obogaćivanje korisničkog iskustva u različitim industrijama i sektorima.

Svi ovi trendovi zajedno oblikuju novo poglavlje u razvoju IS, uz kontinuirani napredak i transformaciju u različitim industrijama i sektorima.

5.7.1 Inteligentni informacioni sistemi

Tehnologije AI i ML su transformisale način na koji se razvijaju i koriste moderni informacioni sistemi. Ovi napredni tehnološki alati omogućavaju sistemima da prepoznaju i razumeju govorni jezik, što je ključno za intuitivnu interakciju sa korisnicima. Kroz upotrebu tehnika obrade prirodnog jezika (engl. *Natural Language Processing*, NLP), AI sistemi analiziraju tekstualne podatke, interpretiraju ih i odgovaraju na pitanja ili izvršavaju zahteve korisnika. Ova sposobnost omogućava sistemima da pruže personalizovanu uslugu korisnicima i efikasno obrade velike količine podataka.

Inteligentni IS takođe koriste ML i AI kako bi analizirali podatke, identifikovali obrasce i trendove, i izvukli korisne informacije koje bi inače bile teško uočljive. Sposobni su da obavljaju postavljene zadatke, ili da donose odluke na osnovu analize podataka i unapred definisanih pravila ili modela. Adaptiraju se na promenljive uslove okruženja ili postavljene zadatke.

Kroz proces učenja, ovi sistemi mogu vremenom da poboljšavaju svoje performanse bez potrebe za dodatnim programiranjem za svaki novi scenario. Tako npr. u kontekstu pametnih gradova, ovi sistemi mogu da automatski reaguju na promene u saobraćaju [8] ili vremenske i hidrološke uslove [9], prilagođavajući svoje preporuke ili akcije u realnom vremenu.

AI takođe omogućava prepoznavanje analognog ulaza, kao što su slike ili zvukovi [10, 11], koristeći tehnike kompjuterskog vida (engl. *Computer Vision*, CV) [8, 12] i obrade signala [13]. Sistemi mogu automatski klasifikovati slike, prepoznati lica ili interpretirati zvučne signale, što otvara vrata novim aplikacijama u oblastima kao što su bezbednost, medicina i automatizacija industrijskih procesa.

Inteligentne aplikacije mogu automatski da označavaju važne sadržaje, filtriraju informacije ili predlažu akcije na osnovu analize podataka. Ovo je od posebne važnosti u okruženjima gde je potrebno brzo donošenje odluka na osnovu promenljivih uslova ili velike količine podataka.

Konačno, AI doprinosi efikasnijem rešavanju prioriteta zadatka kroz automatsku analizu, rangiranje i raspoređivanje zadataka prema njihovoj važnosti i rokovima, što pak organizacijama omogućava da optimizuju resurse i povećaju produktivnost kroz bolje upravljanje radnim tokovima.

5.7.2 Simulacije

Simulacije su nezamenjiv alat u savremenim IS, pružajući inženjerima mogućnost da kreiraju detaljne digitalne modele stvarnih mašina, uređaja, procesa i kompleksnih sistema, radi detaljne analize performansi i karakteristika sistema pre nego što se započne izrada prototipa.

Osim što olakšavaju razvoj novih, simulacije su ključne i za optimizaciju postojećih proizvoda, omogućavajući inženjerima da identifikuju potencijalne slabosti ili da kreiraju efikasnija rešenja bez potrebe za skupim eksperimentalnim testiranjem.

Kroz simulacije, inženjeri mogu dublje da razumeju uzročno-posledične veze i zakonitosti koje vladaju u složenim sistemima. Analizom podataka

dobijenih iz simulacija, moguće je da se predvidi kako će se sistem ponašati u različitim uslovima rada ili pod opterećenjem, što je od ključnog značaja za optimizaciju performansi i bezbednosti sistema. Precizna kontrola parametara procesa i sistemskih karakteristika koje se postiže kroz simulacije omogućava inženjerima da eksperimentišu sa različitim scenarijima i brzo prilagode dizajn prema potrebama, čime se značajno unapređuje efikasnost procesa i smanjuju operativni troškovi.

U industriji, simulacije se sve više koriste kao sredstvo za optimizaciju resursa i maksimizaciju performansi sistema. Ovo tehnološko oruđe ne samo da doprinosi inovacijama i konkurentnosti na tržištu, već i omogućava organizacijama da budu proaktivne u rešavanju izazova koje donosi složeno poslovno okruženje.

5.7.3 Virtuelna stvarnost

Tehnologija virtuelne stvarnosti (VR) pruža brojne mogućnosti za unapređenje IS u različitim industrijama. VR pre svega omogućava kreiranje realističnih simulacija koje poboljšavaju procese učenja i razvoja veština bez potrebe za tradicionalnim metodama. U zdravstvu, VR se primenjuje za realističnu obuku medicinskog osoblja, simulaciju operacija i terapiju pacijenata. U automobilskoj industriji, VR omogućava inženjerima da dizajniraju vozila sa optimizovanom ergonomijom i performansama. Virtuelno testiranje bezbednosnih karakteristika vozila pre fizičkog prototipa omogućava ranu detekciju potencijalnih problema i brže iteracije u dizajnu. Osim toga, VR se koristi za obuku radnika u proizvodnji, simulirajući radne procese i procedure kako bi se povećala efikasnost i smanjili troškovi. Vojska koristi VR za simulaciju taktičkih operacija i trening vojnika, razvoj vojne strategije i obuke, i pripremu vojnika za realne izazove na terenu. U obrazovanju, VR se koristi za interaktivno učenje i edukaciju, stvarajući angažovano okruženje za studente da istražuju kompleksne koncepte kroz simulacije i virtuelne eksperimente.

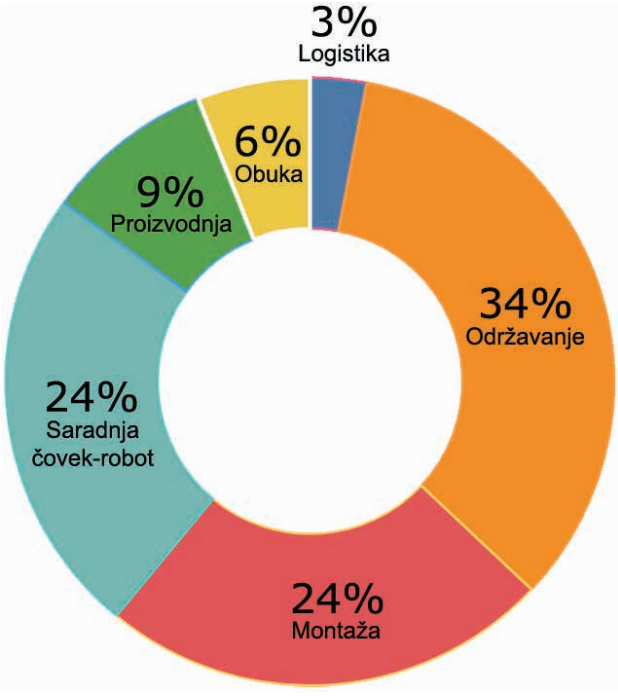
U domenu IS, VR pruža napredne mogućnosti u analizi podataka, omogućavajući vizualizaciju kompleksnih informacija u trodimenzionalnom prostoru. Ovo olakšava otkrivanje skrivenih uzoraka i donošenje odluka na

osnovu podataka i činjenica. Primena VR u IS donosi potencijal za inovaciju i unapređenje konkurentnosti. Integracija ove tehnologije omogućava organizacijama da iskoriste prednosti naprednih simulacija, boljeg korisničkog iskustva, dublje analize podataka i efikasnijeg radnog okruženja.

5.7.4 Proširena stvarnost

Proširena stvarnost (engl. *Augmented reality*, AR) takođe ima široku primenu, naročito u vojsci, medicini i industriji održavanja i popravki. U vojsci, AR se koristi za poboljšanje situacione svesti, navigaciju, prepoznavanje meta i obuku.

U medicini, AR se koristi za navigaciju tokom operacija, prikaz vitalnih podataka pacijenata i obuku medicinskog osoblja. U industriji održavanja i popravki, AR omogućava tehničarima da vizualizuju uputstva za popravke preko realnog sveta, što ubrzava procese održavanja i smanjuje greške.



Slika 43 - Oblasti primene AR u industriji [14]

U prodaji i marketingu, AR se koristi za stvaranje interaktivnih iskustava za potrošače. Na primer, kompanije mogu koristiti AR aplikacije kako bi potencijalnim kupcima omogućile da putem svojih pametnih uređaja "probaju" proizvode ili da vide kako bi određeni komadi nameštaja ili dekoracija izgledali u njihovom prostoru. Ovo direktno angažovanje kupaca može povećati prodaju i poboljšati korisničko iskustvo.

AR se sve više koristi i u domenu IS kako bi se poboljšalo korisničko iskustvo, unapredili procesi i omogućila efikasnija interakcija sa digitalnim informacijama. Na primer, u industriji održavanja i popravki, AR omogućava tehničarima da kroz pametne naočare ili aplikacije na pametnim telefonima vizualizuju uputstva za popravke direktno na fizičkim objektima. Ovo smanjuje potrebu za korišćenjem papirnih uputstava i ubrzava procese održavanja.

OSNOVNE FUNKCIJE INFORMACIONIH SISTEMA

6. OSNOVNE FUNKCIJE INFORMACIONIH SISTEMA

6.1 Prikupljanje podataka

Prikupljanje podataka u IS predstavlja proces sakupljanja informacija iz različitih izvora i njihovo smeštanje u odgovarajuće skladište podataka radi kasnije obrade, analize i upotrebe. Na ovaj način, organizacije mogu da koriste relevantne podatke za podršku operativnim procesima, donošenje odluka i druge aktivnosti.

Proces prikupljanja podataka započinje identifikacijom različitih izvora informacija unutar organizacije, kao što su baze podataka, aplikacije, senzori, veb servisi i drugi informacioni resursi.

Nakon toga, definišu se formati i strukture podataka koji će biti prikupljeni, uključujući vrste podataka, polja, atribute i druge relevantne informacije.

Sledeći korak je implementacija mehanizama za prikupljanje podataka iz identifikovanih izvora, kao što su automatski procesi sinhronizacije, ručni unos podataka ili integracija sa drugim sistemima putem programskih interfejsa aplikacije (engl. *Application Programming Interface*, API).

Napokon, podaci se proveravaju i verifikuju kako bi se osigurala njihova tačnost, potpunost i kvalitet pre skladištenja, npr. u bazi podataka.

6.2 Skladištenje podataka

U IS mogu da se koriste različiti sistemi za skladištenje podataka, što obično zavisi od prirode podataka koji se čuvaju i konkretnih potreba i zahteva koje IS treba da zadovolji.

6.2.1 Relacione baze podataka

Podaci mogu da se skladište u bazi podataka, a najčešći tip koji se koristi u ovu svrhu su relacione baze podataka (engl. *relational database*, RDB). Podaci su tabelarno organizovani, pri čemu su relacije između tabela definisane preko odgovarajućih ključeva. Relacione baze imaju podršku za složene upite i transakcije (engl. *Structured Query Language*, SQL) i omogućavaju efikasno skladištenje i upravljanje podacima. Takođe, pružaju sigurnosne mehanizme za kontrolu pristupa podacima, uključujući autentifikaciju, autorizaciju i enkripciju podataka. Kroz integraciju sa različitim aplikacijama, relacione baze omogućavaju efikasno upravljanje podacima unutar IS.

6.2.2 NoSQL baze podataka

Nerelacione baze podataka (NoSQL) nisu ograničene strogo definisanom strukturom podataka, i stoga omogućavaju fleksibilnost i brzo skaliranje uz odlične performanse za efikasno upravljanje velikim i neuređenim skupovima podataka, kao što su skladišta dokumenata, grafičke baze podataka, ključ-vrednost baze, objektne baze podataka i sl. Korisne su kada je potrebno da se upravlja velikim i promenljivim skupovima heterogenih. Pogodne su za upotrebu u veb aplikacijama, npr. za upravljanje korisničkim podacima, skladištenje i obradu velikih količina podataka, čuvanje multimedijalnog sadržaja, kao i za efikasno upravljanje sesijama korisnika, društvene mreže i analitiku podataka. NoSQL sistemi se ponekad nazivaju "Not only SQL", ističući njihovu sposobnost podrške jezicima upita sličnim SQL-u ili koegzistiranje sa SQL bazama podataka u poliglot-persistantnim (engl. *Polyglot persistence*) arhitekturama.

6.2.3 Objektno-orijentisane baze podataka

Objektno-orijentisane baze podataka čuvaju podatke u obliku objekata, čime omogućavaju direktno skladištenje složenih struktura podataka. Korisne su kada je potrebno očuvati objekte i njihove međusobne odnose, kao što su objektno-orijentisani programski jezici i sistemi za upravljanje sadržajem.

6.2.4 Ostale metode skladištenja podataka

In-memory baze podataka umesto na disku čuvaju podatke u memoriji, što omogućava brži pristup podacima i bolje performanse. Korisne su za aplikacije koje zahtevaju visoku brzinu obrade podataka, kao što su finansijske transakcije, analitika u realnom vremenu i upravljanje keš memorijom.

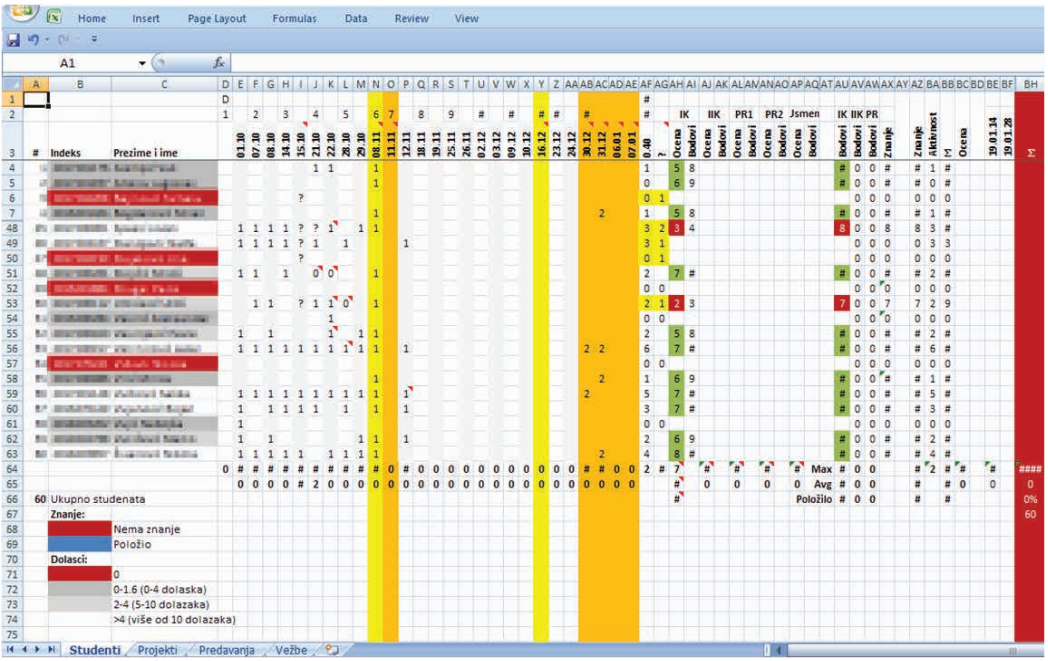
Sistemi fajlova, poput NTFS, FAT32 ili ekt4, takođe omogućavaju čuvanje podataka u fajlovima i direktorijumima (engl. *folder*). Obično se koriste za skladištenje nestrukturiranih podataka poput dokumenata, slika i video zapisa.

Mrežni skladišni prostor (engl. *Network-attached storage*, NAS) je tip uređaja za skladištenje podataka na nivou datoteke, koji omogućava pristup za više korisnika preko mreže. Uglavnom su dostupni kao namenski napravljeni uređaji i obično se koriste za centralizovano skladištenje datoteka u organizacijama.

Mreže za skladištenje podataka (engl. *Storage Area Network*, SAN) je tip arhitekture za skladištenje podataka koji omogućava da više uređaja za skladištenje podataka budu povezani na zajedničku mrežu. Često se koristi u poslovnim okruženjima za potrebe skladištenja velikih količina podataka uz visoku brzinu pristupa podacima.

6.2.5 Problemi

Čuvanje podataka u dokumentima različitih tipova može izazvati probleme u analizi i konsolidaciji informacija. Kada se podaci nalaze u raznim formatima, kao što su Office dokumenti (*Word*, *Excel*, *Power Point*, *PDF, ...*), tehnički dokumenti (*Auto Cad*, *Solid Works*, *Ansys*, *Abaqus*, ...) i sl., postaje teže standardizovati i integrisati te informacije u jedinstvene izveštaje. Različiti tipovi dokumenata često imaju različite strukture i načine organizacije podataka, što može otežati njihovo objedinjavanje i analizu.



Slika 44 - Agregacija podataka u aplikaciji *MS Excel*
Istorijski podaci o prisustvu studenata na predavanjima u toku semestra

Nestandardizovani formati čuvanja podataka dodatno komplikuju proces izveštavanja. Kada podaci nisu pohranjeni u doslednim formatima, postoji rizik od grešaka i neslaganja u interpretaciji informacija.

Na slici 44 je prikazana tabela napravljena u aplikaciji *MS Excel*, sa ciljem prikupljanja podataka o prisustvu studenata na predavanjima u toku semestra, njihovim predispitnim aktivnostima i završnom ocenom, i potom analize tih podataka. Na individualnom nivou (*predavač*), ovakav dokument može da predstavlja koristan instrument za sticanje uvida u efikasnost procesa nastave. Međutim, ukoliko je potrebno da se sagleda šira slika, na nivou samo jednog predmeta gde nastavu vrši više predavača, postavlja se pitanje uticaja ljudskog faktora na unos podataka i centralnog prikupljanja podataka radi analize. Svaki predavač vodi ovakav tip beleški na sebi svojstven način, a ukoliko koriste standardizovan dokument, mogući su tehnički problemi izazvani tipom i verzijom softvera koji koriste (*MS Excel*, *LibreOffice Calc*, *Apache OpenOffice Calc*, ...). Centralno prikupljanje

podataka u najjednostavnijem obliku može da bude manuelno, kombinacijom podataka iz više dokumenata u jedan zajednički dokument uz rizik pojave greške usled ljudskog faktora. Automatizacija je moguća, ali je prvo potrebno da se izradi i testira odgovarajuća aplikacija, što podrazumeva dodatne, često nedostupne resurse. Ukoliko bi bilo potrebno da se podaci analiziraju na nivou fakulteta ili univerziteta (*više predavača i predmeta, različiti studijski programi*), ovakav pristup ostaje teorijski moguć, ali neprimenjiv u praksi.

```
1. Summary
Meeting title   Veb bazirani IS - Predavanja
Attended participants   49
Start time  11/10/23, 1:48:32 PM
End time    11/10/23, 4:54:10 PM
Meeting duration    3h 5m 37s
Average attendance time 2h 5m 34s

2. Participants
Name    First Join  Last Leave  In-Meeting Duration
Stefan Stefanović - 2021200999  11/10/23, 1:48:39 PM    11/10/23, 4:49:53 PM    3h 1m 14s
Marina Marijanović - 2021200999 11/10/23, 1:48:45 PM    11/10/23, 4:50:23 PM    3h 1m 38s
Jovan Jovanović - 2021240999    11/10/23, 1:48:45 PM    11/10/23, 4:49:52 PM    3h 1m 7s
Marko Marković - 2022206999 11/10/23, 1:49:14 PM    11/10/23, 4:50:03 PM    3h 48s
```

Slika 45 - MS Teams, izveštaj sa sastanka, CSV format

```
[
    {
        "ime": "Stefan Stefanović - 2021200999",
        "prvi_pristup": "11/10/23, 1:48:39 PM",
        "poslednji_odlazak": "11/10/23, 4:49:53 PM",
        "trajanje_sastanka": "3h 1m 14s"
    },
    {
        "ime": "Marina Marijanović - 2021200999",
        "prvi_pristup": "11/10/23, 1:48:45 PM",
        "poslednji_odlazak": "11/10/23, 4:50:23 PM",
        "trajanje_sastanka": "3h 1m 38s"
    },
    {
        "ime": "Jovan Jovanović - 2021240999",
        "prvi_pristup": "11/10/23, 1:48:45 PM",
        "poslednji_odlazak": "11/10/23, 4:49:52 PM",
        "trajanje_sastanka": "3h 1m 7s"
    },
    {
        "ime": "Marko Marković - 2022206999",
        "prvi_pristup": "11/10/23, 1:49:14 PM",
        "poslednji_odlazak": "11/10/23, 4:50:03 PM",
        "trajanje_sastanka": "3h 48s"
    }
]
```

Slika 46 - Izveštaj sa sastanka, JSON format

```
Name               | Indeks     | First Join          | Last Leave          | Duration
-------------------+------------|---------------------|---------------------|-----------
Stefan Stefanović  | 2021200099 | 11/10/23, 13:48:39 | 11/10/23, 16:49:53 | 3h 1m 14s
Marina Marijanović | 2021200999 | 11/10/23, 13:48:45 | 11/10/23, 16:50:23 | 3h 1m 38s
Jovan Jovanović    | 2021240999 | 11/10/23, 13:48:45 | 11/10/23, 16:49:52 | 3h 1m 7s
Marko Marković     | 2022206999 | 11/10/23, 13:49:14 | 11/10/23, 16:50:03 | 3h 48s
```

Slika 47 - Izveštaj sa sastanka, tekst format

Prilikom automatizacije prikupljanja podataka, moguć je scenario gde su podaci pohranjeni u različitim formatima, npr. u CSV formatu, slika 45, JSON formatu, slika 46, i u tabelama formatiranim kao običan tekst, slika 47. Problem je jednostavan i tehnički trivijalan za rešavanje izradom odgovarajućeg parsera, ali zahteva fokus na detalje s obzirom da lokalizacija nije standardizovana i da imena polja u JSON formatu odstupaju od imena polja u CSV i tekst formatu. Ova nedoslednost, kao i svaka naknadna promena u formatu sačuvanih podataka može dovesti do problema u radu parsera, a posledično i do nepoverenja u tačnost izveštaja.

Redundencija podataka je još jedan ozbiljan problem. Kada se iste informacije skladište na više mesta, može doći do neusklađenosti i grešaka u izveštajima. Takođe, višestruko skladištenje istih podataka povećava troškove za skladištenje i upravljanje podacima, kao i složenost njihovog održavanja. Efikasno upravljanje podacima uključuje eliminisanje redundancije i obezbeđivanje jedinstvenog za sve informacije.

Različita softverska rešenja u okviru sistema takođe mogu dodatno da otežaju izveštavanje usled različitih formata za čuvanje podataka, kao i problema sa kompatibilnošću, integracijom, importom i eksportom podataka, što može da rezultuje poteškoćama u povezivanju i konsolidaciji informacija iz više izvora.

6.3 Obrada podataka

Obrada podataka je ključni deo poslovnog obaveštavanja i uključuje sve procese koji se odvijaju od trenutka kada se podaci prikupe do trenutka kada su spremni za analizu. Ovo može uključivati prikupljanje podataka iz

različitih izvora kao što su baze podataka, veb servisi ili datoteke, i složene procese kao što su agregacija podataka, normalizacija, i konsolidacija.

U slučaju da se podaci prikupljaju iz više izvora, možda će biti potrebna njihova integracija u jedinstvenu strukturu. Proces podrazumeva spajanje podataka iz različitih izvora i njihovo organizovanje u skladu sa zadatim formatom ili modelom. Transformacija podataka obuhvata primenu različitih operacija ili pravila nad prikupljenim podacima kako bi se dobili željeni rezultati. To može uključivati izračunavanje novih atributa, filtriranje ili grupisanje podataka, ili primenu drugih manipulacija.

Prikupljeni podaci moraju da se očiste kako bi se uklonile greške i neslaganja, da se standardizuju tako da budu u jedinstvenom formatu, i da se transformišu u oblik koji je pogodan za analizu. Proces čišćenja podataka obuhvata identifikaciju i ispravku grešaka, nedoslednosti ili nedostataka u prikupljenim podacima. To može uključivati uklanjanje duplikata, popunjavanje nedostajućih vrednosti ili ispravku pogrešnih informacija, sve sa ciljem da se dobiju tačne i relevantne informacije.

6.4 Distribucija podataka

Segment distribucije podataka u IS se bavi organizacijom podataka radi efikasnog upravljanja i optimizacije performansi sistema. Ovaj proces podrazumeva raspoređivanje podataka na različite fizičke lokacije ili servere radi povećanja dostupnosti podataka ili bržeg pristupa podacima, optimizacije performansi i otpornosti sistema na kvarove, havarije i druge nepredviđene situacije. Izbor optimalnih lokacija za servere je ključan, a faktori kao što su latencija, cena i bezbednost zahtevaju detaljnu procenu pre donošenja odluke.

Na primer, baza podataka kompanije može da se klonira, pri čemu klonovi mogu da se nalaze

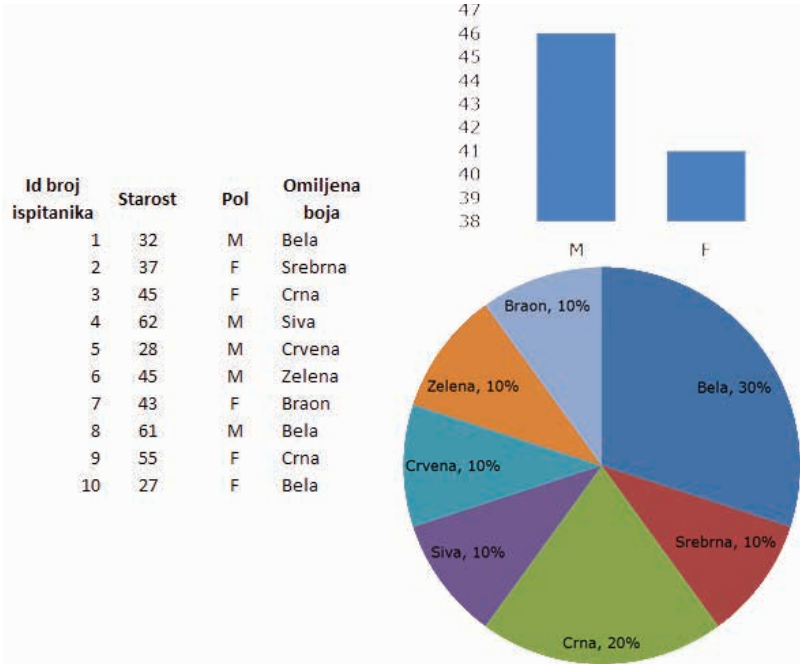
- u istom data centru, ali na različitim serverima, ili
- u više data centara na različitim lokacijama

Dodatno, distribucija podataka u IS uključuje i deljenje velikih skupova podataka na manje delove ili particije (engl. *partitioning*), koji se zatim distribuiraju na različite servere ili čvorove sistema.

Ova praksa omogućava bolje upravljanje velikim količinama podataka i efikasniju obradu upita, smanjujući opterećenje na pojedinim delovima infrastrukture. Velika baza podataka može biti podeljena na segmente prema geografskim lokacijama ili po vrsti podataka (npr. transakcije, analitički podaci), što omogućava optimizovan pristup i bržu obradu podataka u skladu sa specifičnim zahtevima korisnika ili konkretne aplikacije.

6.5 Analiza podataka

Proces analize podataka podrazumeva korišćenje različitih tehnika i alata za analizu prikupljenih i obrađenih podataka radi identifikacije obrazaca, trendova ili ključnih informacija koje mogu biti korisne za organizaciju.



Id broj ispitanika	Starost	Pol	Omiljena boja
1	32	M	Bela
2	37	F	Srebrna
3	45	F	Crna
4	62	M	Siva
5	28	M	Crvena
6	45	M	Zelena
7	43	F	Braon
8	61	M	Bela
9	55	F	Crna
10	27	F	Bela

Slika 48 - Izvorni podaci u Excel tabeli (levo); Deskriptivna i pregledna analiza (desno)

Analiza podataka u IS se odnosi na proces istraživanja, interpretacije i tumačenja prikupljenih podataka kako bi se identifikovali obrasci, trendovi, odnosi ili druge korisne informacije koje mogu da budu od koristi organizaciji, pre svega kroz dublje razumevanje poslovnih operacija, procesa ili tržišta na osnovu čega potom mogu da se identifikju prilike za poboljšanje, smanje rizici, i pre svega donesu kvalitetne odluke i optimizuju aktivnosti.

Analiza podataka u IS može uključivati različite vrste analitičkih tehnika i alata.

- *Deskriptivna analiza* opisuje i sumira osnovne karakteristike i svojstava podataka, kao što su srednje vrednosti, mediani, distribucije ili frekvencije. Na slici 48 su prikazani rezultati ankete, sačuvani u Excel tabeli. Na osnovu tih podataka su potom izračunate prosečna starost ispitanika po polu, kao i procentualna zastupljenost navedenih boja.
- *Pregledna analiza* podrazumeva upotrebu vizualizacije podataka, kao što su grafikoni, dijagrami ili tabele, radi jasnog prikaza informacija i identifikacije trendova ili obrazaca.
- *Prediktivna analiza* koristi statističke modele i tehnike kao što su regresiona analiza, vremenske serije ili mašinsko učenje, za predviđanje budućih događaja ili rezultata na osnovu postojećih podataka.
- *Preskriptivna analiza* se fokusira na identifikaciju najboljih akcionih koraka ili strategija na osnovu analize podataka, npr. kroz upotrebu optimizacionih tehnika ili simulacija za pronalaženje optimalnih rešenja.
- *Otkrivanje anomalija* je usmereno na identifikaciju neobičnih ili nepredviđenih uzoraka ili događaja u podacima koji mogu ukazivati na potencijalne probleme ili prilike.

Analiza podataka može da bude zasnovana na istorijskim podacima, tekućim podacima, ili se može vršiti filtriranjem podataka prema različitim

kriterijumima, pri čemu svaki pristup ima svoje specifične prednosti i primene.

Analiza bazirana na *istorijskim podacima* se fokusira na proučavanje podataka koji su već prikupljeni i arhivirani. Ova vrsta analize omogućava identifikaciju trendova, obrazaca i ciklusa koji se mogu koristiti za predviđanje budućih događaja i donošenje strateških odluka. Korišćenjem istorijskih podataka, organizacije mogu da steknu uvid u to kako su se različiti faktori razvijali tokom vremena i kako su prethodne odluke uticale na rezultate, što se može pokazati korisnim prilikom dugoročnog planiranja i analize performansi.

Sa druge strane, u dinamičnom okruženju gde je bitan brzi odgovor na promene, pogodna je analiza bazirana na *tekućim podacima* koja koristi informacije koje su trenutno dostupne i relevantne za trenutne aktivnosti i operacije. Ova vrsta analize pruža uvid u trenutno stanje poslovanja i omogućava brzo reagovanje na promene u okruženju. Analizom tekućih podataka, organizacije mogu da prate i odmah identifikuju probleme ili prilike, i optimizuju operativne procese u realnom vremenu.

MODELIRANJE INFORMACIONIH SISTEMA

7. MODELIRANJE INFORMACIONIH SISTEMA

Razvoj IS u kompanijama predstavlja složen proces koji može biti pogođen različitim izazovima. uključujući modeliranje procesa, organizacionu strukturu, sistematizaciju radnih mesta, dokumente kvaliteta, tokograme aktivnosti, distribuciju ovlašćenja i ljudski faktor, koji dalje mogu značajno da utiču na efikasnost i uspeh implementacije sistema. Modeliranje realnog procesa je ključni korak u razvoju IS, jer predstavlja osnovu na kojoj će sistem biti izgrađen.

Ukoliko se trenutni procesi u kompaniji ne analiziraju dovoljno detaljno, može doći do implementacije sistema koji ne odgovara stvarnim potrebama. Nedostaci u analizi mogu biti rezultat nedostatka informacija, neadekvatne komunikacije sa korisnicima ili jednostavno nepoznavanja stvarnog toka posla.

Distribucija ovlašćenja i autorizacija predstavljaju značajan izazov. Tačan raspored ovlašćenja i prava pristupa unutar sistema moraju da budu precizno mapirani da bi se obezbedili sigurnost i integritet podataka. Odstupanje od pravila za pristup može dovesti do problema kao što su neautorizovani pristup ili manipulacija podacima.

Ljudski faktor često je nepredvidiv i može značajno da utiče na uspeh implementacije IS. Problemi kao što su otpor prema promenama, nedovoljna obuka korisnika i greške u korišćenju sistema mogu otežati postizanje željenih rezultata. Efikasno upravljanje promenama, pružanje adekvatne obuke i motivacija korisnika su presudni za dugoročno uspešno funkcionisanje IS.

U složenim organizacijama sa mnogo međusobno povezanih procesa, modeliranje može postati veoma komplikovano, dok nepravilno modeliranje

može da dovede do neefikasnog sistema koji ne može da se uklopi u postojeće organizacione i radne procese, ili uzrokuje dodatne probleme.

7.1 Upravljanje kvalitetom

Sistem upravljanja kvalitetom (engl. *Quality Management System*, QMS), predstavlja okvir ili set procedura koje organizacija koristi kako bi osigurala da njeni proizvodi i usluge zadovoljavaju određene standarde i potrebe kupaca. *QMS* obuhvata politike, ciljeve, resurse, procese i procedure koji su usmereni na postizanje i održavanje visokog nivoa kvaliteta u svim aspektima poslovanja.

- *Politike kvaliteta*. Formalno dokumentovane izjave o kvalitetu koje odražavaju ciljeve organizacije.
- *Planiranje*. Ciljevi kvaliteta, kao i resursi i procesi potrebni za postizanje tih ciljeva.
- *Kontrola procesa*. Praćenje i merenje procesa kako bi se obezbedila doslednost i identifikovale oblasti za poboljšanje.
- *Procenjivanje učinka.* Praćenje i analiza performansi sistema i procesa, uključujući interne i eksternalno revizije.
- *Poboljšanje.* Kontinuirano unapređivanje procesa i proizvoda na osnovu povratnih informacija i analize podataka.
- *Dokumentacija.* Pisane procedure i uputstva koja podržavaju i osiguravaju doslednu primenu procesa.

QMS može biti koristan za različite tipove organizacija, bez obzira na njihovu veličinu ili sektor poslovanja. Proizvođači, na primer, koriste *QMS* kako bi osigurali da njihovi proizvodi ispunjavaju standarde bezbednosti, performansi i pouzdanosti. Uslužne organizacije, uključujući finansijske institucije, zdravstvene ustanove, obrazovne ustanove i ugostiteljske firme, implementiraju *QMS* kako bi poboljšale kvalitet svojih usluga i zadovoljstvo svojih korisnika. Takođe, u industrijama sa strožim regulatornim zahtevima, kao što su farmaceutska industrija, automobilska industrija i avijacija, *QMS* je ključan za usklađivanje sa standardima i propisima. Čak i mala i srednja preduzeća mogu imati koristi od *QMS*-a, jer im može pomoći da unaprede

svoje operacije, poboljšaju konkurentnost i izgrade bolje odnose sa kupcima. Velike korporacije često koriste *QMS* kako bi upravljale složenim procesima i velikim brojem dobavljača i kupaca.

Bez obzira na vrstu organizacije, *QMS* može pomoći u sistematskom pristupu upravljanju kvalitetom. U kontekstu razvoja IS, ovaj standard pomaže u uspostavljanju jasnih procesa i procedura koji obuhvataju sve faze od planiranja i dizajna do implementacije i održavanja sistema.

7.1.1 ISO 9000 standard

Porodica standarda *ISO 9000* je skup od inicijalno pet *QMS* standarda (*ISO 9000* do *ISO 9004*) koje je razvila Međunarodna organizacija za standardizaciju (ISO). Ovi standardi pomažu organizacijama da zadovolje potrebe kupaca i drugih zainteresovanih strana u okviru zakonskih i regulatornih zahteva koji se odnose na proizvod ili uslugu.

- *ISO 9000.* Ovaj standard opisuje terminologiju i osnovne koncepte i principe sistema upravljanja kvalitetom.
- *ISO 9001* je najpoznatiji standard u ovoj porodici. Definiše zahteve za sistem upravljanja kvalitetom koji organizacije treba da ispune kako bi obezbedile da njihovi proizvodi i usluge ispunjavaju zahteve kupaca i zakonske regulative. Ovaj standard se koristi za sertifikaciju organizacija.
- *ISO 9002* je bio standard specifičan za proizvodnju i instalaciju. Povučen je i zamenjen sa *ISO 9001* 2000. godine.
- *ISO 9003* je bio standard koji se odnosio na inspekciju i ispitivanje gotovih proizvoda. Takođe je povučen i zamenjen sa ISO 9001 2000. godine.
- *ISO 9004* je standard koji pruža smernice za poboljšanje performansi i dugoročnu uspešnost organizacije kroz sistem upravljanja kvalitetom. Fokusira se na šire aspekte upravljanja kvalitetom i kontinuiranog poboljšanja, i nije namenjen za sertifikaciju, već kao alat za unapređenje poslovanja.

ISO 9001 se fokusira na sledeće ključne oblasti:

- *Upravljanje procesima.* Organizacije trebaju da identifikuju i upravljaju svojim procesima kako bi osigurale doslednost i kvalitet u isporuci proizvoda ili usluga.
- *Zadovoljstvo kupaca.* Standard naglašava važnost razumevanja i zadovoljavanja potreba kupaca, kao i stalnog poboljšanja.
- *Vođstvo i angažovanje zaposlenih.* Efikasno upravljanje kvalitetom zahteva angažman i liderstvo na svim nivoima organizacije.
- *Kontrola i praćenje*. Organizacije moraju da prate i mere performanse svojih procesa i proizvoda kako bi identifikovale mogućnosti za poboljšanje.

Implementacija *ISO 9001* može da ima važnu ulogu u razvoju i implementaciji IS, zato što osigurava da su svi aspekti razvoja sistema dobro dokumentovani i da su postupci u skladu sa definisanim standardima kvaliteta, što uključuje identifikaciju i upravljanje rizicima, kontrolu promena i redovno praćenje performansi sistema. *ISO 9001* podržava obuku i osposobljavanje osoblja, što je ključno za osiguranje da timovi koji razvijaju i održavaju IS poseduju neophodna znanja i veštine.

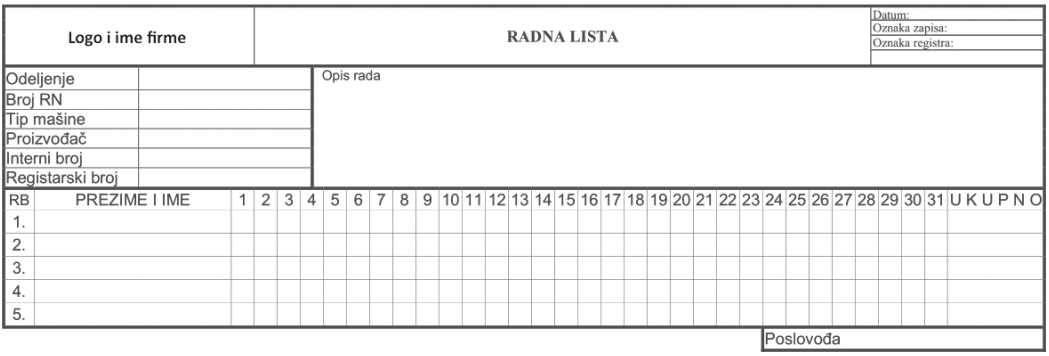
Kroz primenu principa kvaliteta iz ISO 9001 standarda, kompanije mogu da poboljšaju efikasnost procesa u svojim IS, da smanje greške, povećaju zadovoljstvo korisnika i osiguraju da sistem ispunjava sve regulativne i funkcionalne zahteve. Na taj način, QMS kao što je *ISO 9001* direktno doprinosi uspehu i održivosti IS u kompaniji.

7.1.2 Dokumenti kvaliteta

Dokumenti kvaliteta u okviru *QMS* kao što je ISO 9001, igraju ključnu ulogu u uspostavljanju i održavanju standarda kvaliteta unutar organizacije. Ovi dokumenti pomažu u definisanju, implementaciji i praćenju procesa, procedura i standarda koji osiguravaju da organizacija ispunjava zahteve svojih kupaca i regulatorne zahteve. Dokumenti kvaliteta u okviru QMS obuhvataju sledeće elemente:

- *Politika kvaliteta.* Izjava vrhovnog menadžmenta o opredeljenosti organizacije prema kvalitetu.

- *Planovi kvaliteta.* Dokumenti koji detaljno opisuju specifične ciljeve i aktivnosti koje treba sprovesti kako bi se postigli zahtevi kvaliteta za određene projekte ili proizvode.
- *Priručnik kvaliteta.* Dokument koji opisuje sistem upravljanja kvalitetom i kako su ispunjeni zahtevi standarda, iako *ISO 9001:2015* ne zahteva obavezno priručnik kvaliteta.
- *Procedura.* Detaljan opis specifičnih procesa i aktivnosti u organizaciji, uključujući njihove odgovornosti, korake i resurse potrebne za ostvarivanje rezultata.
- *Uputstvo za rad.* Detaljna uputstva o tome kako se izvršavaju određene aktivnosti ili zadaci.
- *Obrasci i zapisnici.* Dokumentacija koja se koristi za beleženje informacija i rezultata procesa, kao što su zapisnici o inspekcijama, kontrolama kvaliteta, korektivnim akcijama, itd. Primer obrasca radne liste je prikazan na slici 49.



Logo i ime firme	RADNA LISTA	Datum: Oznaka zapisa: Oznaka registra:

		Opis rada
Odeljenje		
Broj RN		
Tip mašine		
Proizvođač		
Interni broj		
Registarski broj		

RB	PREZIME I IME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	U	K	U	P	N	O
1.																																						
2.																																						
3.																																						
4.																																						
5.																																						

Poslovođa

Slika 49 - Dokumenti kvaliteta, obrazac radne liste

Prilikom razvoja IS, dokumenti kvaliteta mogu da pomognu na sledeći način:

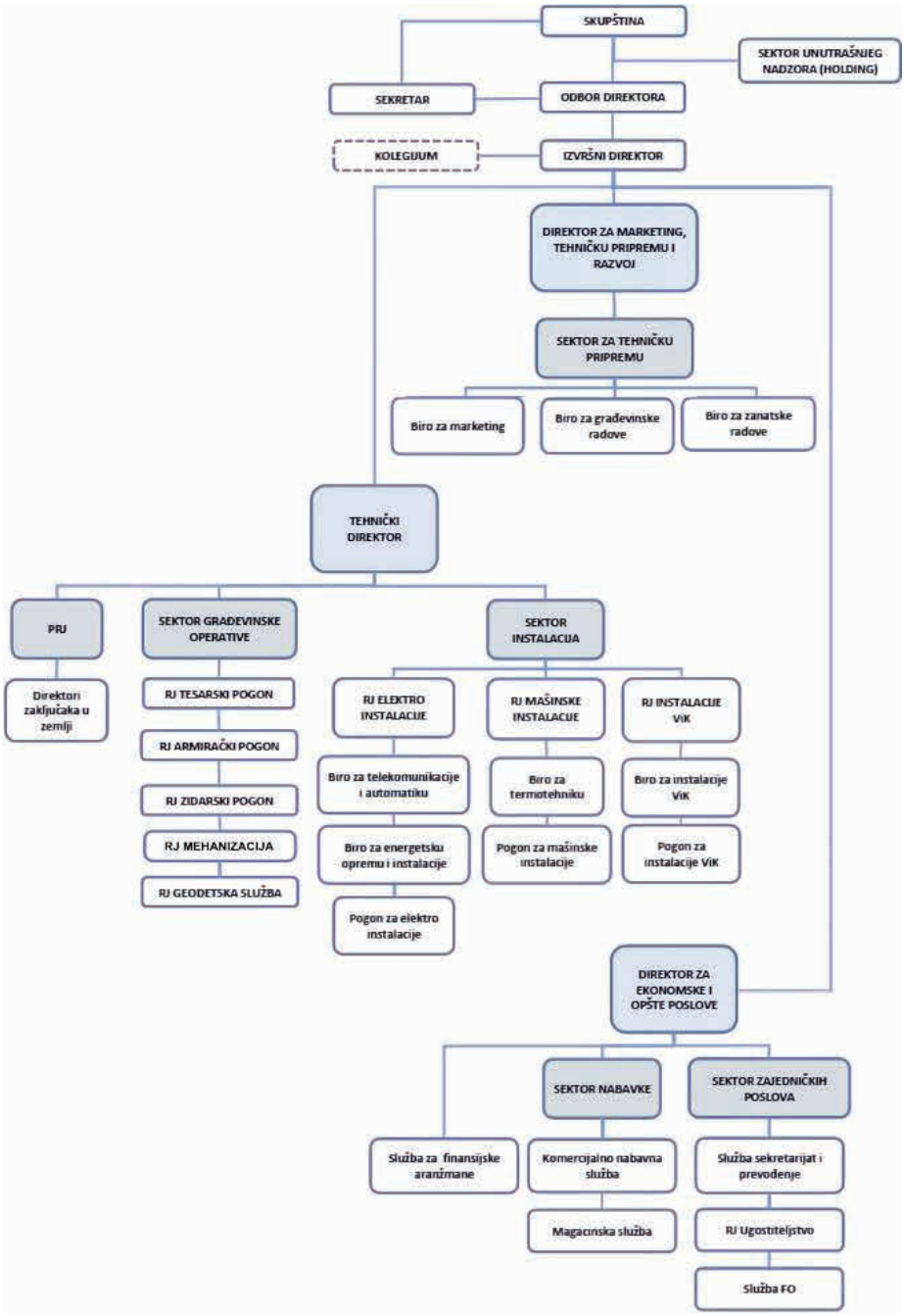
- *Jasna definicija zahteva i očekivanja.* Politike i procedure kvaliteta mogu da pomognu u definisanju zahteva koje IS treba da ispuni, kao što su funkcionalnosti, bezbednost, pouzdanost i usklađenost sa standardima.

- *Mapiranje procesa.* Dokumenti kvaliteta kao što su *procedura* i *uputstva za rad*, mogu da pomognu u mapiranju poslovnih procesa i tokova rada koji se implementiraju u IS. Ovo omogućava jasnu viziju toka informacija i integraciju različitih poslovnih funkcija u sistem.
- *Identifikacija i upravljanje rizicima.* Dokumentacija može da pomogne u identifikaciji eventualnih rizika i problema u postojećim procesima i sistemima, što može biti korisno pri prilagođavanju postojećih i razvoju novih rešenja.
- *Osiguranje usklađenosti.* Dokumenti kvaliteta pomažu u tome da se obezbedi da razvoj IS bude usklađen sa industrijskim standardima, regulativama i zahtevima korisnika.
- *Praćenje i evaluacija performansi.* Kroz zapise i obrasce, organizacija može da prati performanse sistema, identifikuje probleme i implementira poboljšanja. Ovo može da obuhvati procese testiranja sistema, ocenjivanja rezultata i analize povratnih informacija dobijenih od korisnika.

Nepotpuna ili neažurirana dokumentacija može uzrokovati nesigurnost u vezi sa funkcionalnostima sistema i greške u njegovoj implementaciji, dok nedostatak standardizacije može dovesti do varijacija u kvalitetu dokumentacije, otežavajući obuku i održavanje sistema.

7.2 Modeliranje organizacione strukture

Organizaciona struktura kompanije, slika 50, predstavlja sistematski raspored i međusobne odnose između različitih organizacionih jedinica, odeljenja i zaposlenih unutar organizacije. Ova struktura definiše hijerarhiju, odgovornosti, autoritet i komunikacijske kanale, omogućavajući efikasno upravljanje resursima i koordinaciju aktivnosti. Služi za jasno određivanje uloga i odgovornosti, efikasniju koordinaciju, optimizaciju procesa donošenja odluka i postizanje strateških ciljeva organizacije. Dobra organizaciona struktura pomaže u upravljanju složenim operacijama, unapređuje produktivnost i omogućava lakše prilagođavanje promenama u poslovnom okruženju.



Slika 50 - Deo organizacione šeme kompanije

Organizaciona struktura kompanije ima značajan uticaj na razvoj IS. Kada odeljenja nisu dobro povezana, ili dovoljno koordinirana, njihove različite potrebe i procesi mogu da stvore izazove u kreiranju jedinstvenog rešenja, odnosno otežaju integraciju IS unutar kompanije.

Promene u organizaciji, kao što su restrukturiranje ili spajanje odeljenja, mogu da učine prethodno funkcionalan sistem neadekvatnim. Prilagođavanje i ažuriranje sistema je neophodno kako bi se osiguralo da IS ostane relevantan i funkcionalan u svetlu novih organizacionih struktura i potreba.

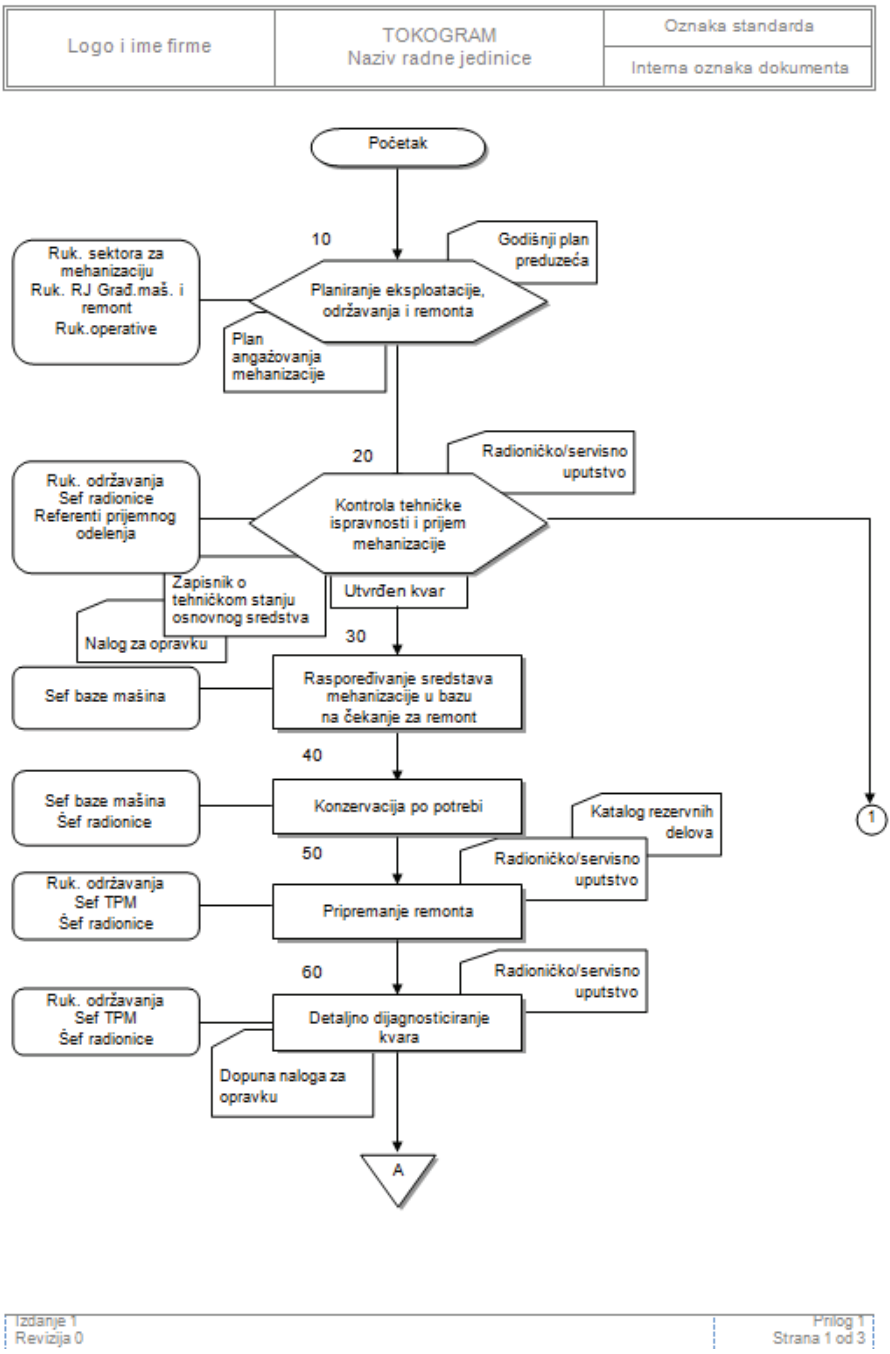
Sistematizacija radnih mesta, koja uključuje definisanje uloga i odgovornosti zaposlenih u vezi sa IS-om, takođe može da predstavlja problem. Nejasne odgovornosti mogu da dovedu do preklapanja uloga, što će otežati efikasnu upotrebu sistema.

7.2.1 Tokogrami aktivnosti

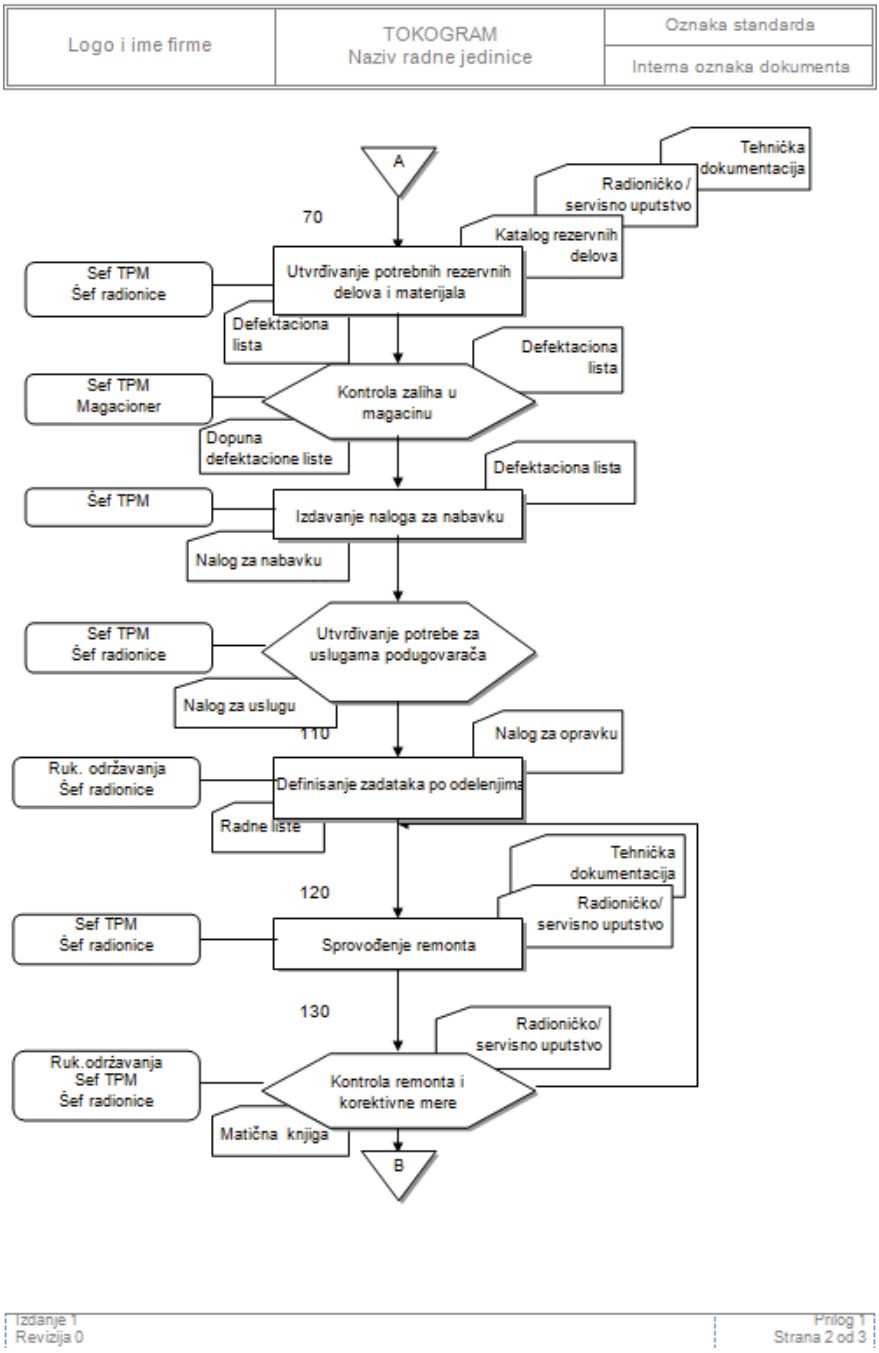
Tokogram aktivnosti je vizuelni alat koji se koristi za modeliranje i analizu poslovnih procesa i IS, sa ciljem da prikaže kako se različite aktivnosti i koraci međusobno povezuju i kako informacija teče kroz proces. Ovaj prikaz omogućava lakše razumevanje procesa, identifikaciju potencijalnih problema, kao i mogućnosti za optimizaciju i unapređenje efikasnosti. Takođe omogućava jasnu komunikaciju o procesu sa članovima tima i pomaže u obuci novih zaposlenih.

Tokogram aktivnosti obuhvata sledeće elemente:

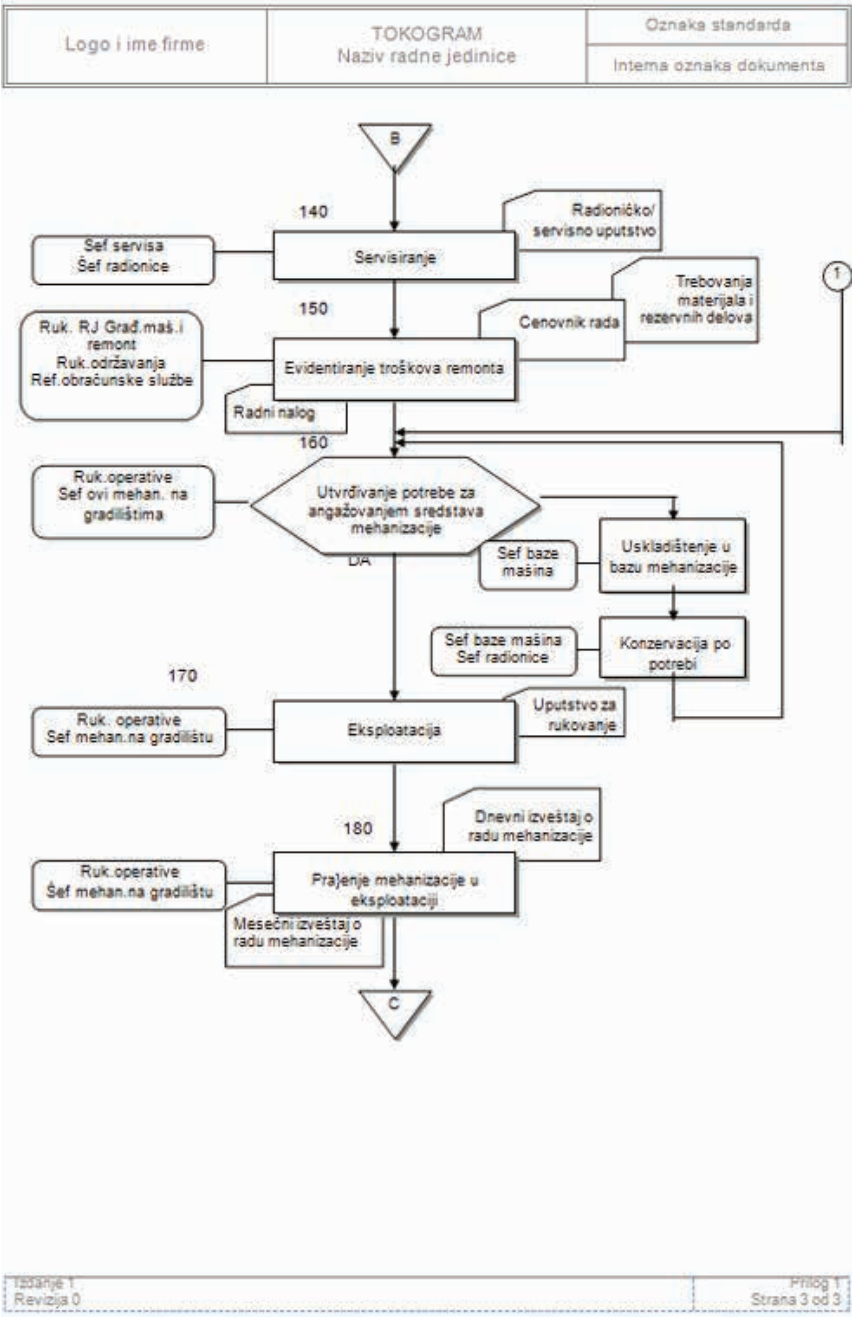
- *početne i završne tačke*, mesta gde proces počinje i gde se završava
- *aktivnosti* ili *koraci*, pojedinačne operacije u procesu
- *tokovi* ili *veze*, linije koje pokazuju pravac i redosled aktivnosti
- *odluke* ili *grananje*, tačke gde se donose odluke koje mogu promeniti tok operacija.
- *uloge i odgovornosti*, prikazuju se kao deo aktivnosti, označavajući ko je odgovoran za određene korake.



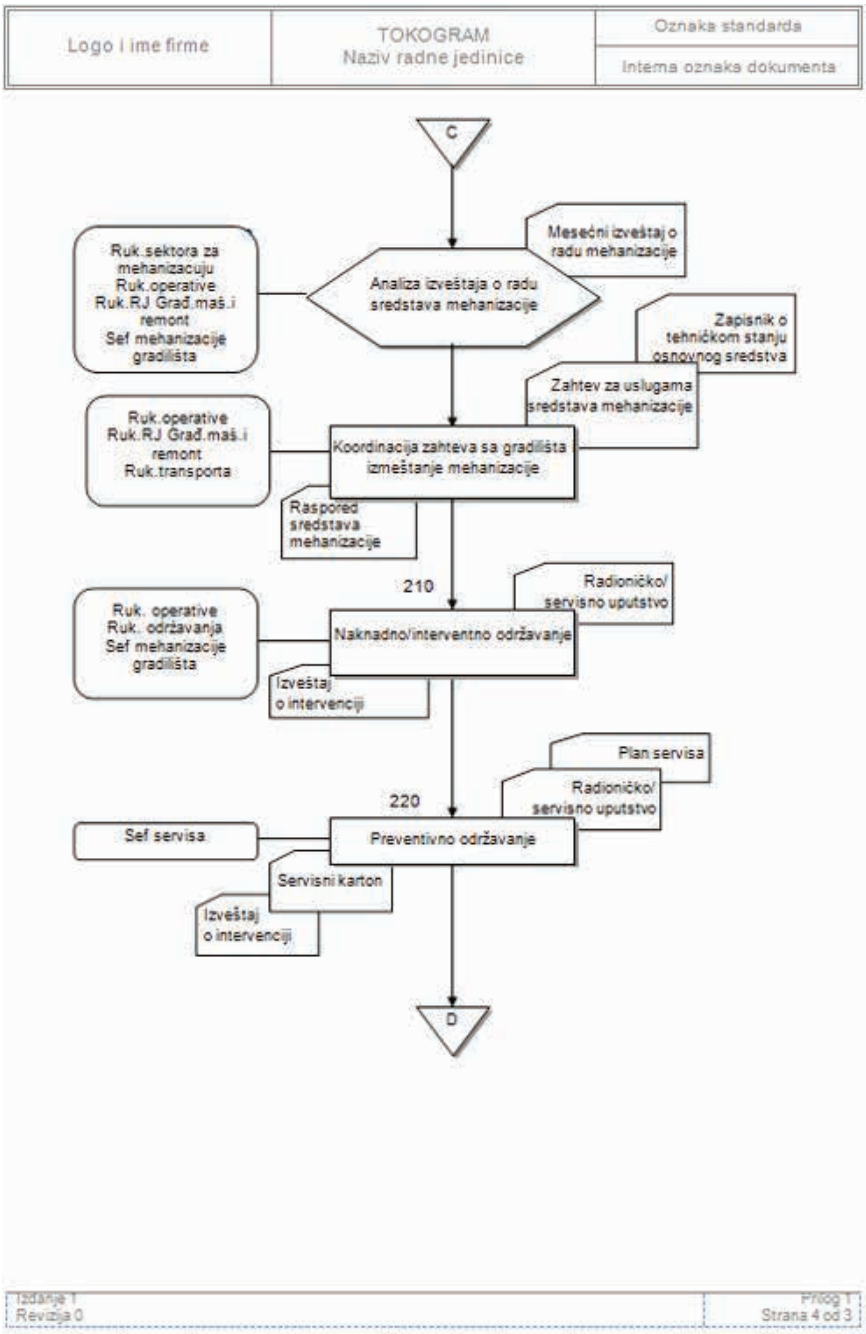
Slika 51 - Tokogram aktivnosti, 1. deo



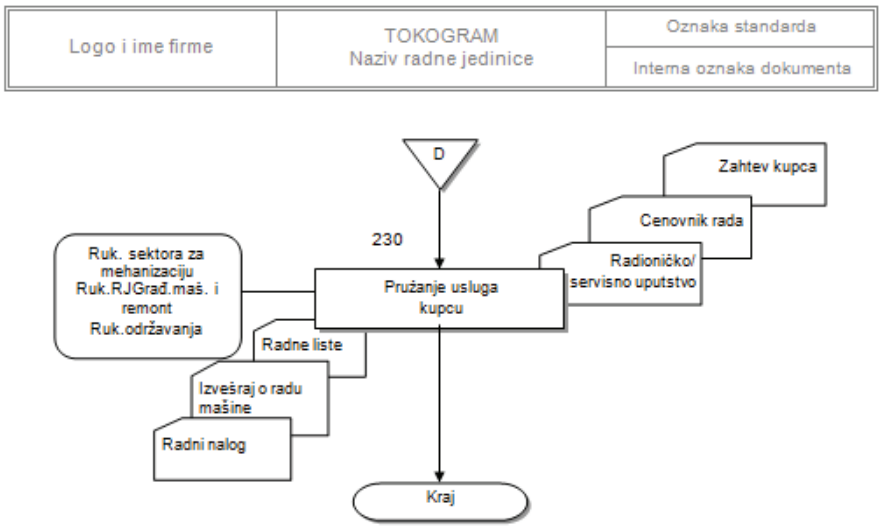
Slika 52 - Tokogram aktivnosti, 2. deo



Slika 53 - Tokogram aktivnosti, 3. deo



Slika 54 - Tokogram aktivnosti, 4. deo



Slika 55 - Tokogram aktivnosti, 5. deo

Na slikama 51-55 je prikazan tokogram aktivnosti u radnoj jedinici zaduženoj za upravljanje resursima mehanizacije u okviru kompanije sa složenom organizacionom strukturom, slika 50.

Tokogram aktivnosti predstavlja važan dokument u razvoju i implementaciji IS, pružajući detaljan uvid u postojeće poslovne procese. Dizajnerima sistema omogućava da precizno razumeju ključne korake i njihove međusobne veze, tok posla i koje funkcionalnosti je potrebno da se implementiraju. Pomaže u identifikaciji specifičnih zahteva koje novi sistem treba da zadovolji, čime se osigurava da krajnji proizvod bude u skladu sa stvarnim poslovnim potrebama.

Pored toga, ovaj alat pomaže u razvoju scenarija za testiranje koji odražavaju stvarne tokove aktivnosti, čime se omogućava precizno i detaljno testiranje sistema pre puštanja u rad.

Ukoliko nisu dobro modelirani, ili ukoliko nisu ažurirani nakon organizacionih promena, tokogrami mogu da izazovu neusklađenost sa stvarnim procesima, što potom može da izazove probleme u vezi sa operativnošću i efikasnošću IS. U praksi je moguće da se ode i u drugu krajnost tako što tokogrami postaju previše složeni i samim tim teški za razumevanje i primenu, što potom može da oteža implementaciju i održavanje IS.

7.2.2 Distribucija ovlašćenja i autorizacija

Distribucija ovlašćenja i autorizacija u IS kontrolišu sigurnost i pristup informacijama i resursima sa ciljem da osiguraju da samo ovlašćeni korisnici imaju pristup određenim funkcionalnostima i podacima unutar sistema. Odnosi se na način na koji se prava i privilegije dodeljuju korisnicima i grupama korisnika unutar IS, i obično je zasnovana na principu najmanjih privilegija, što znači da korisnici dobijaju samo ona ovlašćenja koja su im neophodna za obavljanje njihovih zadataka.

Korisnicima se dodeljuju uloge, ili se organizuju u grupe koje imaju specifična ovlašćenja. Uloga *korisnik* najčešće ima samo osnovne privilegije, dok uloga *administratora* ima sve privilegije u sistemu. *Grupa*

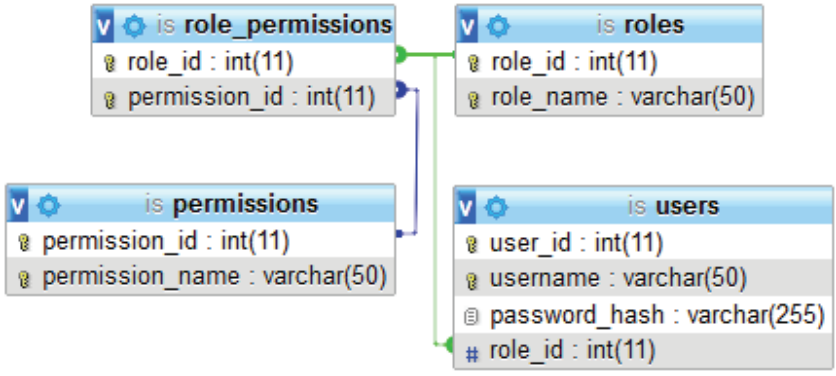
korisnika može biti definisana na osnovu funkcionalnih potreba ili na osnovu organizacione strukture kompanije.

Politike ovlašćenja propisuju način na koji se ovlašćenja dodeljuju i kako se sa njima upravlja. Kontrola pristupa može biti zasnovana na osnovu *uloge* (engl. *Role-based access control*, RBAC), ili na *atributima* korisnika (engl. *Attribute-Based Access Control*, ABAC) i datog konteksta.

Ovlašćenja se postavljaju i ažuriraju u skladu sa promenama u organizaciji, kao što su novi zaposleni, promene u radnim zadacima ili promene u organizacionoj strukturi. Takođe, treba redovno proveravati i ažurirati ovlašćenja kako bi se osiguralo da su u skladu sa aktuelnim potrebama i politikama.

Autorizacija je proces koji prati autentifikaciju i određuje koje resurse i funkcionalnosti korisnik može da koristi nakon što se prijavi u sistem. Nakon autentifikacije, sistem koristi pravila i politike ovlašćenja kako bi odredio koji resursi ili funkcionalnosti su dostupni korisniku. Ovo uključuje kontrolu pristupa modulima aplikacije, podacima, dokumentima i drugim resursima.

Ovlašćenja mogu biti dodeljena na različitim nivoima, uključujući operativne, administrativne i sistemske nivoe. Na primer, dok većina korisnika ima privilegiju samo da čita podatke, određeni korisnici sistema mogu da budu ovlašćeni da te podatke menjaju ili brišu.



Slika 56 - Struktura tabela za podršku sistemu za distribuciju ovlašćenja

Autorizacija korisnika se prati i dokumentuje, kako bi se omogućila revizija i otkrivanje neovlašćenih pristupa ili nepoželjnih aktivnosti. Ovo može uključivati logovanje aktivnosti, kreiranje izveštaja o pristupu i sprovođenje bezbednosnih procena.

Na slici 56 je prikazan jednostavan primer strukture MySQL baze podataka namenjene za podršku sistemu za distribuciju ovlašćenja u veb aplikaciji. U tabeli *users* se nalaza podaci o korisničkim nalozima i ulozi svakog naloga u sistemu. Tabela *roles* definiše uloge u sistemu, dok tabela *permissions* definiše tipove ovlašćenja koja mogu da budu dodeljena. Tabela *role_permissions* povezuje uloge sa dodeljenim ovlašćenjima. Na slici 57 je prikazan sadržaj opisanih tabela.

Tabela *roles*

role_id	role_name
1	Administrator
2	Inženjer VII
3	Šef tehničke pripreme

Tabela *users*

user_id	username	password_hash	role_id
1	petar	hashed_password1	1
2	mirko	hashed_password2	2
3	tamara	hashed_password3	3

Tabela *permissions*

permission_id	permission_name
3	Izmena
4	Brisanje
2	Čitanje
1	Kreiranje

Tabela *role_permissions*

role_id	permission_id
1	1
1	2
1	3
1	4
2	1
2	2
2	3
3	2

Slika 57 - Sadržaj tabela za podršku sistemu za distribuciju ovlašćenja

Zadatak SQL upita prikazanog na listingu 17 jeste da očita osnovne podatke o korisničkom nalogu (*user_id*, *username*) i njegovoj ulozi u IS (*role_id*, *role_name*).

```sql
SELECT
    u.user_id,
    u.username,
    r.role_id,
    r.role_name
FROM
    users u
JOIN
    roles r ON u.role_id = r.role_id
WHERE
    u.user_id = XYZ;
```

Listing 17 - Očitavanje podataka o korisniku i njegovoj ulozi u IS, SQL upit

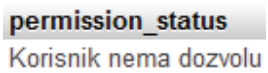
user_id	username	role_id	role_name
1	petar	1	Administrator

Slika 58 - Očitavanje podataka o korisniku i njegovoj ulozi u IS, rezultat SQL upita

Do željenog rezultata se dolazi u jednom upitu, spajanjem tabela *users* i *roles* na osnovu polja *role_id* u tabeli *users* i polja *role_id* u tabeli *roles*. Upit se filtrira tako da vraća podatke samo za željenog korisnika, čiji je *user_id* u upitu simbolično prikazan u *WHERE* klauzuli kao XYZ. Rezultat upita za korisnika čiji user_id ima vrednost 1 je prikazan na slici 58.

```sql
SELECT
    CASE
        WHEN EXISTS (
            SELECT 1
            FROM users u
            JOIN
                roles r
                ON u.role_id = r.role_id
            JOIN
                role_permissions rp
                ON r.role_id = rp.role_id
            WHERE u.user_id = 3
            AND rp.permission_id = 3
        )
        THEN 'Korisnik ima dozvolu'
        ELSE 'Korisnik nema dozvolu'
    END AS permission_status;
```

Listing 18 - Provera ovlašćenja korisnika za izabranu akciju



Slika 59 - Rezultat provere ovlašćenja za korisnika

SQL upit na listingu 16 proverava da li korisnik sa *user_id=3* ima dozvolu za akciju označenu sa *permission_id=3* tako što se koristi EXISTS kako bi se utvrdilo da li postoji odgovarajući red u tabelama *users*, *roles* i *role_permissions*. Upit prvo povezuje tabelu *users* sa tabelom *roles* preko *role_id*, zatim povezuje *roles* sa tabelom *role_permissions* preko *role_id* da bi se proverilo da li je tražena dozvola dodeljena ulozi korisnika. Ako se pronađe barem jedan red koji zadovoljava uslove, upit vraća poruku *Korisnik ima dozvolu*, dok će u suprotnom vratiti poruku *Korisnik nema dozvolu*. Rezultat upita je prikazan na slici 59.

Upit je očigledno tehnički ispravan i daje rezultat, ali taj rezultat možda nije očekivan. Ukoliko se uporede vrednosti navedene u upitu na listingu 16 sa vrednostima u bazi podataka, slika 57, za korisnika *user_id=3* koji ima dodeljenu ulogu *šefa tehničke pripreme*[1] se proverava da li ima dozvolu za akciju *Izmena* na dokumentima. U fizičkom svetu, šef OJ je odgovoran za sva dokumenta koja iz te jedinice idu ka drugim OJ u okviru kompanije, što bi u digitalnom svetu moglo da se opiše ulogom administratora i iz tog razloga dobijeni rezultat upita može da deluje netačno. Međutim, da bi se održao integritet dokumenata u kompaniji, sadržaj dokumenta koji je ušao u proceduru ne sme da se menja, već se takav dokument stornira, a umesto njega se u proceduru pušta novi dokument. Navedenom analizom može da se zaključi da je rezultat upita korektan, ali i da je lista tipova ovlašćenja ubačenih u bazu podataka nepotpuna, odnosno da je potrebno da se ubace podaci za akciju *storniraj*.

[1] Odeljenje u organizacionoj strukturi kompanije

IZVEŠTAVANJE U INFORMACIONIM SISTEMIMA

8. IZVEŠTAVANJE U INFORMACIONIM SISTEMIMA

Izveštavanje u IS je proces kreiranja i distribucije strukturiranih prikaza podataka sa ciljem da se olakša razumevanje velikih količina podataka, praćenje performansi i donošenja odluka. Ovaj proces obuhvata prikupljanje podataka iz različitih izvora unutar sistema, njihovu obradu i transformaciju kako bi se prilagodili specifičnim potrebama izveštaja, na osnovu čega se potom generišu izveštaji u obliku tekstualnih opisa, tabela i grafikona.

Izveštaji mogu biti periodični, kao što su dnevni, nedeljni ili mesečni, ili *ad-hoc*, što znači da se kreiraju prema potrebi radi rešavanja specifičnih pitanja ili analiza. Korisnicima omogućavaju da analiziraju trendove, identifikuju probleme, i prate ključne performanse, pomažući u donošenju informisanih odluka i unapređenju poslovnih procesa.

Proces izveštavanja često uključuje i distribuciju izveštaja relevantnim korisnicima u organizaciji, koji mogu da koriste ove informacije za operativne odluke, ali i strategijsko planiranje. U savremenim IS, izveštaji mogu biti *dinamični* i *interaktivni*, omogućavajući korisnicima da istražuju podatke i prilagođavaju prikaze prema svojim potrebama.

Osnovni preduslovi za kvalitetno izveštavanje u IS podrazumevaju precizno opisane i dokumentovane *poslovne transakcije* kako bi se osiguralo da podaci koji se prikupljaju i obrađuju odražavaju stvarne poslovne aktivnosti i procese. *Agregacija podataka* podrazumeva sakupljanje i sumiranje podataka iz različitih izvora kako bi se obezbedila kompaktnija, ali relevantna informacija za izveštaje. *Ekstrakcija podataka* se odnosi na sposobnost efikasnog *izvlačenja* potrebnih informacija iz baza podataka ili drugih izvora radi kreiranja tačnih i ažurnih izveštaja.

Možda i najvažniji preduslov jeste dobro dizajnirana baza podataka sa jasno definisanim relacijama između tabela, čime se stiču uslovi za precizno povezivanje i analizu podataka. To uključuje pravilno korišćenje primarnih i stranih ključeva (engl. *foreign keys*), kao i optimizovane upite za preuzimanje i obradu potrebnih podataka.

8.1 Poslovno obaveštavanje

Poslovno obaveštavanje (engl. *Business Intelligence*, BI) u kontekstu IS se odnosi na tehnologije, alate i procese koji omogućavaju organizacijama da prikupe, analiziraju i koriste podatke kako bi doneli informisane poslovne odluke. Poslovno obaveštavanje obuhvata niz aktivnosti, uključujući obradu podataka, filtriranje i prikaz u formi izveštaja, sve s ciljem da se sirovi podaci pretvore u korisne informacije.

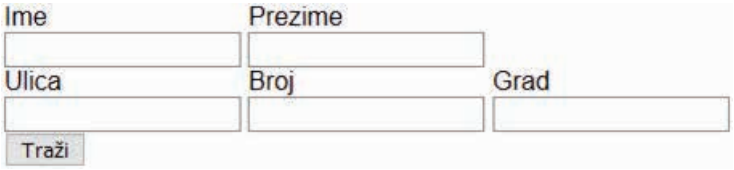
Kombinovanjem obrade podataka, filtriranja i vizualizacije podataka u formi izveštaja, poslovno izveštavanje omogućava organizacijama da efikasno upravljaju podacima i donose poslovne odluke na osnovu tačnih i relevantnih informacija.

8.2 Filtriranje podataka

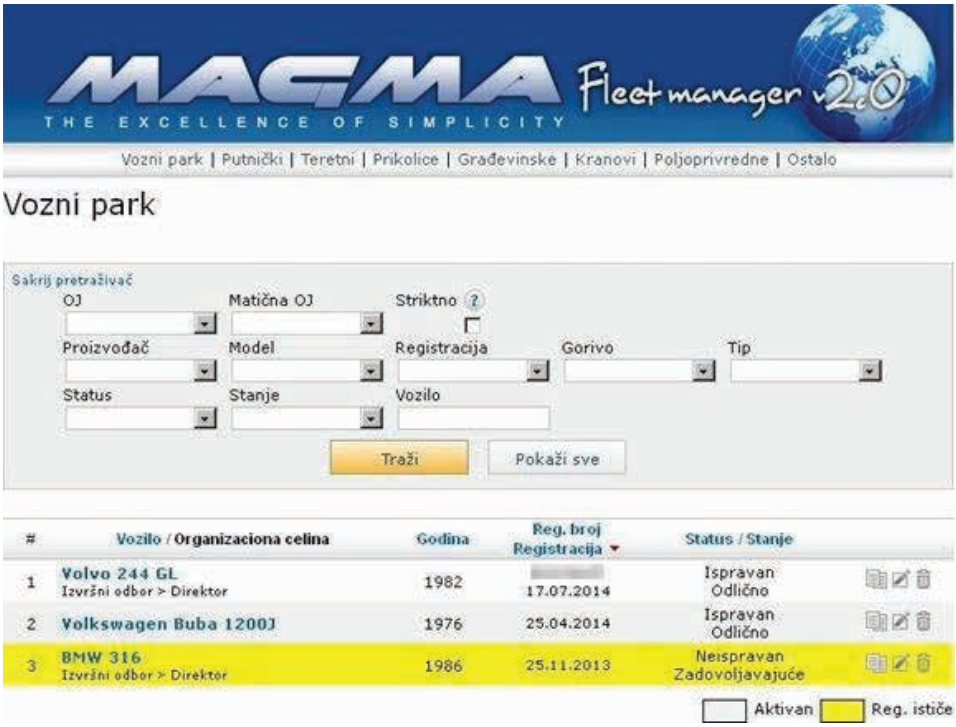
Filtriranje podataka prema različitim kriterijumima omogućava precizno usmeravanje analize na relevantne informacije. Korišćenjem različitih kriterijuma kao što su vremenski periodi, geografske lokacije, ili specifične karakteristike korisnika, analitičari mogu da fokusiraju analizu na specifične aspekte poslovanja. Filtriranje pomaže u otkrivanju skrivenih obrazaca i u trendovima koji bi inače ostali nezapaženi u širem skupu podataka. Na primer, filtriranjem prodajnih podataka prema regionima i vremenskim periodima, može se dobiti uvid u regionalne performanse i sezonske varijacije, što omogućava bolje ciljanje marketinških strategija i optimizaciju resursa.

Proces filtriranja podataka počinje selektovanjem podskupa podataka iz većeg skupa, što se najčešće vrši putem veb obrasca, slika 60 i 61, koji

omogućava korisnicima da unesu specifične informacije ili kriterijume prema kojima žele da vrše pretragu.



Slika 60 - Veb obrazac za pretragu baze korisnika



Slika 61 - Pretraga putničkih vozila filtriranih po datumu isteka registracije Magma Fleet Manager [15]

Nakon što su podaci uneti, definišu se kriterijumi za filtriranje. Struktura baze podataka igra važnu ulogu u ovom procesu, zato što omogućava pravilno definisanje odnosa između različitih entiteta i atributa. Na osnovu punog ili delimičnog seta kriterijuma, formiraju se upiti koji služe za pristup relevantnim podacima. Pre nego što se upiti izvrše, neophodna je

verifikacija kriterijuma kako bi se osiguralo da su svi unosi tačni i da se izbegnu greške prilikom pretrage.

Finalni korak je sortiranje dobijenih podataka. Korisnici mogu odabrati način prikaza podataka, bilo da je reč o rastućem (A-Z, 0-9) ili opadajućem (Z-A, 9-0) redosledu. Na slici 61 je komanda za sortiranje rešena putem *HTML linkova* u naslovu kolona. Ukoliko potrebe nalažu, sistem može da omogućiti i sortiranje podataka u više nivoa, što je korisno kada se podaci grupišu prema više kriterijuma. Na primer, primarno sortiranje može biti po datumu, dok sekundarno može biti po imenu ili kategoriji.

8.2.1 Grupisanje podataka

Grupisanje podataka u *SQL* upitu se koristi za organizovanje i sumarizovanje podataka u skladu sa određenim kriterijumima. To se obično postiže upotrebom klauzule *GROUP BY* koja grupiše redove sa istim vrednostima u ciljanim kolonama u jedinstvene grupe. Ovaj proces omogućava izvršavanje agregatnih funkcija kao što su *COUNT*, *SUM*, *AVG*, *MAX*, ili *MIN* nad svakom grupom.

```
1  SELECT
2      count(id) AS brojac, grad
3  FROM `korisnici`
4  GROUP BY grad
```

```
brojac | grad
-------+---------
10     | Beograd
13     | Niš
16     | Novi Sad
11     | Subotica
```

Listing 19 - Grupisanje podataka po gradu, SQL upit

Na listingu 19 je prikazan primer *SQL* upita koji podatke iz tabele *korisnici* grupiše po koloni *grad* i nad svakom grupom izvršava agregatnu funkciju *COUNT* čime se dobija broj korisnika po gradu.

Na slici 62 [1] je prikazan listing trebovanja ili nabavke goriva za vozilo koje pripada organizacionoj jedinici (OJ) kompanije.

Datum	Tip goriva	Količina	Cena / J.cena
16.06.2015	Benzin 95	14.00	2.000,00RSD 140,90RSD
06.06.2015	Benzin 95	28.00	4.000,00RSD 140,90RSD
02.06.2015	Benzin 95	11.00	1.500,00RSD 140,70RSD
16.05.2015	Benzin 95	21.00	3.000,00RSD 139,90RSD
05.05.2015	Benzin 95	14.00	2.000,00RSD 143,90RSD
05.05.2015	Benzin 95	10.00	1.500,00RSD 143,90RSD
28.04.2015	Benzin 95	11.00	1.500,00RSD 138,90RSD
28.04.2015	Benzin 95	22.00	3.000,00RSD 138,90RSD
28.04.2015	Benzin 95	11.00	1.500,00RSD 138,90RSD
18.04.2015	Benzin 95	18.00	2.500,00RSD 136,90RSD
21.03.2015	Benzin 95	22.00	3.000,00RSD 136,90RSD
18.03.2015	Benzin 95	22.00	3.000,00RSD 136,90RSD
12.03.2015	Benzin 95	18.00	2.500,00RSD 136,90RSD
09.03.2015	Euro Dizel	21.91	3.000,00RSD 136,90RSD
27.02.2015	Benzin 95	22.74	3.000,00RSD 131,90RSD

Slika 62 - Nabavka goriva, rezultat pretrage za izabrano vozilo

Ovakvi izveštaji mogu da se koriste prilikom kontrole regularnosti trebovanja goriva na nivou OJ. U realnim uslovima službena vozila prelaze

veliki broj kilometara što podrazumeva česte nabavke goriva, samim tim i veliki broj stavki [1] u izveštaju za relativno kratke vremenske periode, što je indikator da tabelarni prikaz nije optimalan za brzi pregled. Međutim, ukoliko se uz tabelarni izveštaj dodaju i grupisani podaci, npr. po tipu trebovanog goriva, eventualna odstupanja i greške su lako uočljivi, slika 63 [1].

Benzin 95

337.7l ► 46.010,00RSD

Euro Dizel

21.91l ► 3.000,00RSD

Benzin 98

37.07l ► 70,00KM
22.74l ► 3.000,00RSD

Datum	Tip goriva	Količina	Cena / J.cena
16.06.2015	Benzin 95	14.00	2.000,00RSD 140,90RSD
06.06.2015	Benzin 95	28.00	4.000,00RSD 140,90RSD
02.06.2015	Benzin 95	11.00	1.500,00RSD 140,70RSD
16.05.2015	Benzin 95	21.00	3.000,00RSD 139,90RSD

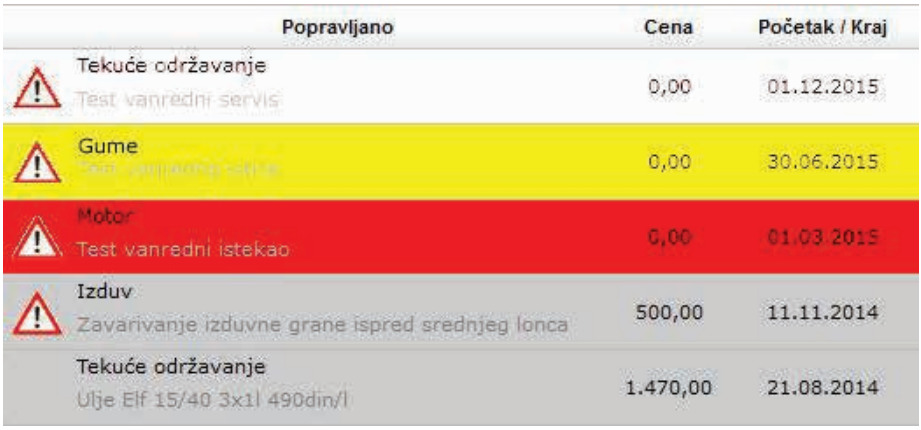
Slika 63 - Nabavka goriva, grupisanje podataka prema tipu goriva

8.2.2 Upotreba boja i grafičkih simbola u izveštavanju

Prikaz velikih količina podataka može se unaprediti upotrebom boja i prigodnih grafičkih simbola, pri čemu treba istaći da izbor boja u vizualizaciji

[1] Pretpostavka: vozilo godišnje prelazi 50,000km, uz prosečnu potrošnju od 8l goriva/100km i prosečno trebovanje od 30l goriva. Na godišnjem nivou, to je ~130 trebovanja goriva.

podataka nije estetske prirode, već ima suštinski značaj za efikasno prenošenje poruka. [16]



Slika 64 - Upotreba boja u izveštaju

Na slici 64 [1] je prikazan pregled liste zadataka na poslovima održavanja za izabrano vozilo [1], gde su podaci obeleženi odgovarajućim bojama pozadine u skladu sa trenutnim statusom, dok grafički simbol označava da li je aktivnost prioritetna:

- *Bela boja*. Aktivnost koja je zabeležena i tek sledi u planu održavanja.
- *Žuta boja*. Aktivnost koja nije izvršena, a zakazani rok izvršenja je manji od 30 dana
- *Crvena boja*. Aktivnost koja nije izvršena, a zakazani rok izvršenja je istekao
- *Siva boja*. Aktivnost koja je kompletirana.

8.3 Tipovi izveštaja

U IS postoji nekoliko osnovnih tipova izveštavanja koji se koriste za prikazivanje i analizu podataka, pri čemu svaki od njih ima svoju specifičnu

[1] Referentni datum izveštaja na slici je 1. jun 2015. godine

svrhu i koristi se u zavisnosti od nivoa menadžmenta u organizaciji i trenutnih konkretnih potreba.

Operativno izveštavanje se koristi za praćenje svakodnevnih operacija i transakcija u organizaciji. Izveštaji obično pružaju detaljne informacije o trenutnom stanju, kao što su prodaja po danu, broj transakcija, stanje zaliha i sl., slika 65. Ovi izveštaji su ključni za operativne menadžere i timove kako bi pratili performanse i donosili brze odluke.

Taktičko izveštavanje se fokusira na srednjoročno planiranje i donošenje odluka. Izveštaji sadrže analize trendova i obrazaca u podacima kako bi se identifikovali problemi ili prilike. Na primer, izveštaji o prodajnim trendovima za poslednjih šest meseci mogu pomoći menadžmentu da planira strategije za sledeći kvartal.



Slika 65 - Operativni izveštaj prodaje

Strateško izveštavanje se koristi za dugoročno planiranje i strategijsko vođenje organizacije. Izveštaji obuhvataju šire analize tržišta, konkurencije, finansijskih performansi i trendova u industriji. Na primer, izveštaji o globalnim tržištima i dugoročnim prognozama mogu pomoći izvršnom menadžmentu da donese ključne strategijske odluke o budućem rastu i razvoju kompanije.

Ad hoc izveštavanje se odnosi na izveštaje koji se kreiraju na zahtev, kako bi se odgovorilo na specifične upite ili potrebe korisnika. Ovi izveštaji su obično privremenog karaktera i fokusirani su na konkretan problem ili pitanje koje zahteva brzo rešenje.

Operativna podrška odlučivanju je tip izveštavanja koji integriše podatke i analize u realnom vremenu kako bi podržao operativne odluke tokom radnog procesa. Kao primer mogu da se navedu alarmni sistemi za praćenje stanja u industrijskim postrojenjima ili izveštaji za praćenje kvaliteta proizvoda tokom proizvodnje.

8.4 Kreiranje i personalizacija izveštaja

Proces kreiranja izveštaja i njihove personalizacije uključuje nekoliko koraka koji osiguravaju da izveštaji ne samo da pružaju tačne informacije, već i da budu prilagođeni specifičnim potrebama korisnika i poslovnim zahtevima.

Prvi zadatak u kreiranju izveštaja jeste da se odredi *poslovni segment za koji se izveštaj kreira*. To znači da izveštaj treba da bude relevantan za određeni deo poslovanja, kao što su *prodaja*, *finansije*, *ljudski resursi* ili *proizvodnja*. Razumevanje poslovnog segmenta omogućava dizajnerima izveštaja da uključe samo relevantne podatke i analize, čime se povećavaju upotrebna vrednost i preciznost izveštaja.

U sledećem koraku je potrebno da se podese *kriterijumi za filtriranje podataka*, kao što su vremenski periodi, geografske lokacije, ili specifične karakteristike korisnika i transakcija.

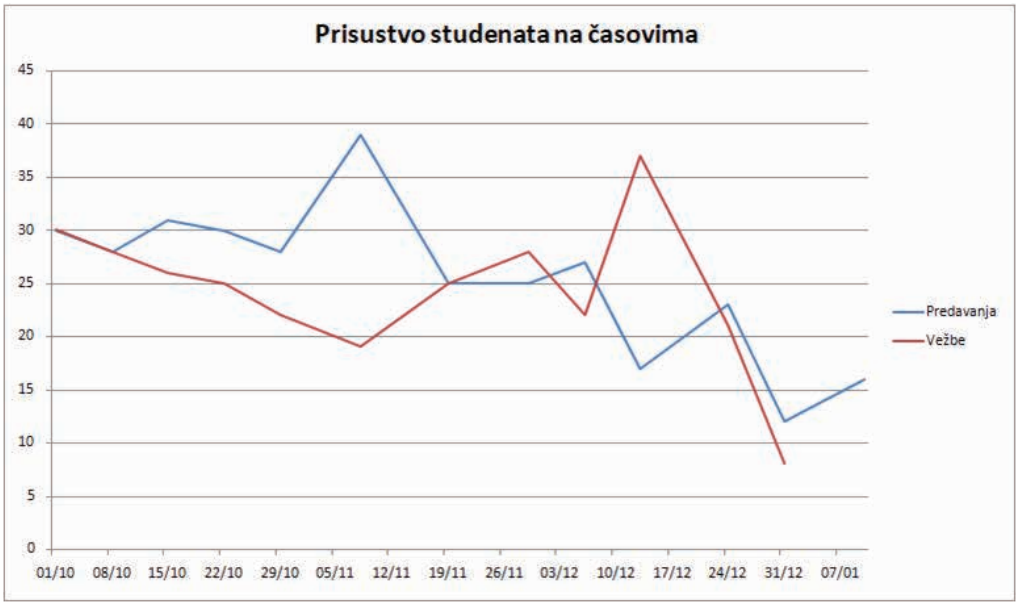
Kada je izveštaj kreiran, prikazuje se na ekranu korisnika, što omogućava brzu analizu i interakciju sa podacima, često uz opciju *štampanja*, pružajući fizičke kopije koje se mogu koristiti za distribuciju, arhiviranje ili prezentacije.

Personalizacija izveštaja obuhvata usklađivanje sa internim procedurama i standardima u okviru sistema kvaliteta, ali i uklapanje u vizuelni identitet

kompanije (*logo*, *boje*, *fontovi*, *veličina slova*, ...), što doprinosi njihovom profesionalnom izgledu i preglednosti.

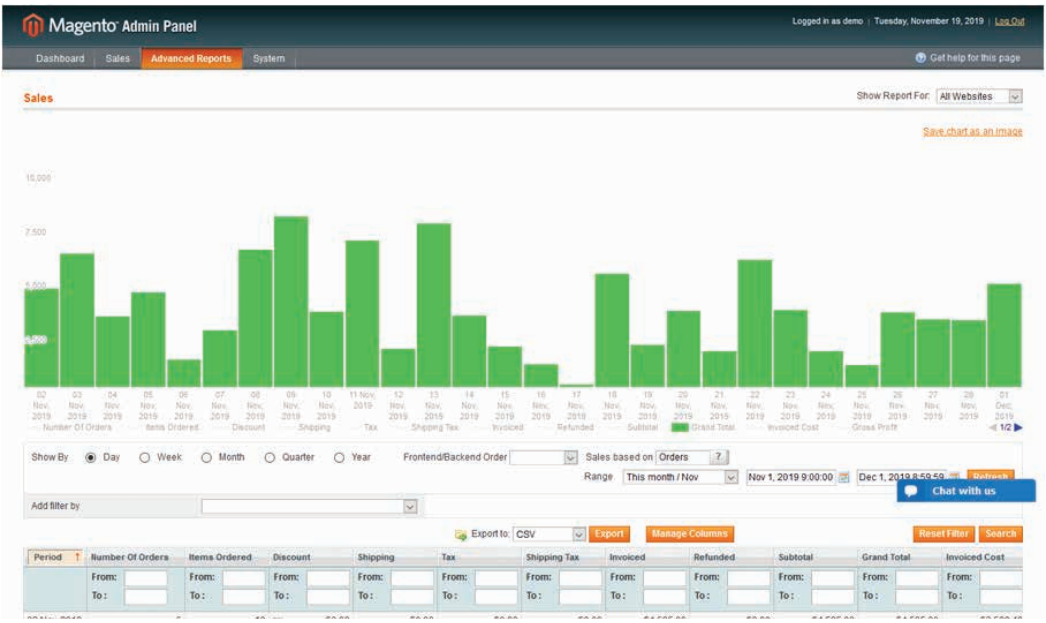
8.5 Vizuelizacija podataka

Vizuelizacija podataka predstavlja grafičku prezentaciju informacija koja omogućava korisnicima da lakše razumeju složne skupove podataka i trenutno prepoznaju ključne informacije, slika 66. Korišćenjem različitih grafičkih prikaza, kao što su grafikoni, dijagrami i mape, vizuelizacija olakšava interpretaciju podataka i identifikaciju obrazaca, trendova i odnosa koji bi ostali neprimećeni posmatranjem podataka u izvornom obliku.

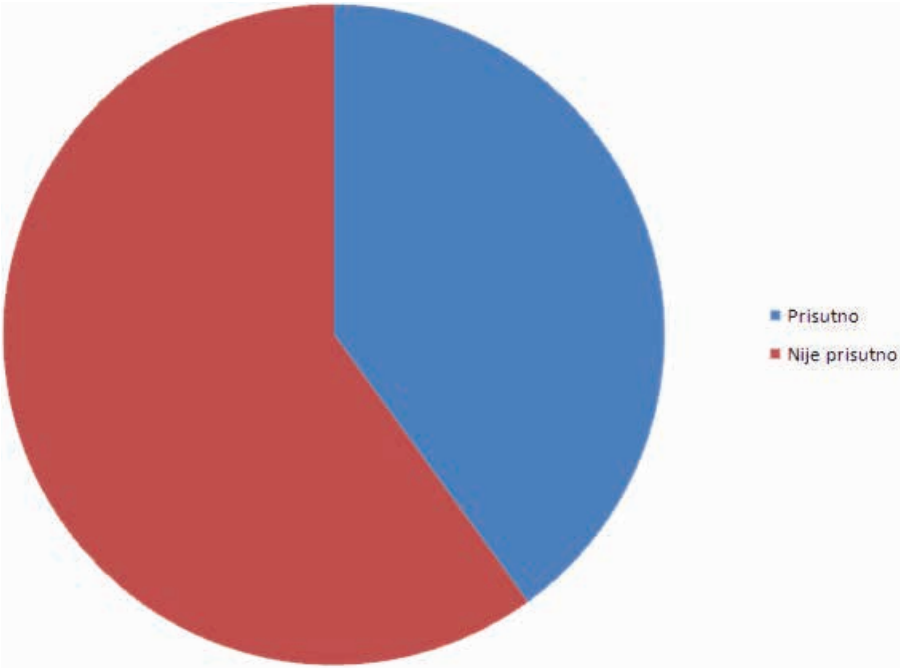


Slika 66 - Vizuelizacija podataka o prisustvu studenata na predavanjima

Tradicionalni poslovni izveštaji često su preplavljeni tabelarnim prikazima i suvišnim detaljima, što može otežati razumevanje osnovnih informacija. Sa druge strane, sposobnost da se na efikasan način interpretiraju ključne informacije je neophodna za donošenje informisanih odluka. Osnovni cilj vizuelizacije podataka je da transformiše složene informacije u razumljive i deljive poruke, čime se olakšava donošenje odluka i unapređuje saradnja.



Slika 67 - *Stubasti dijagram*, izveštaj o maloprodaji, *Magento*



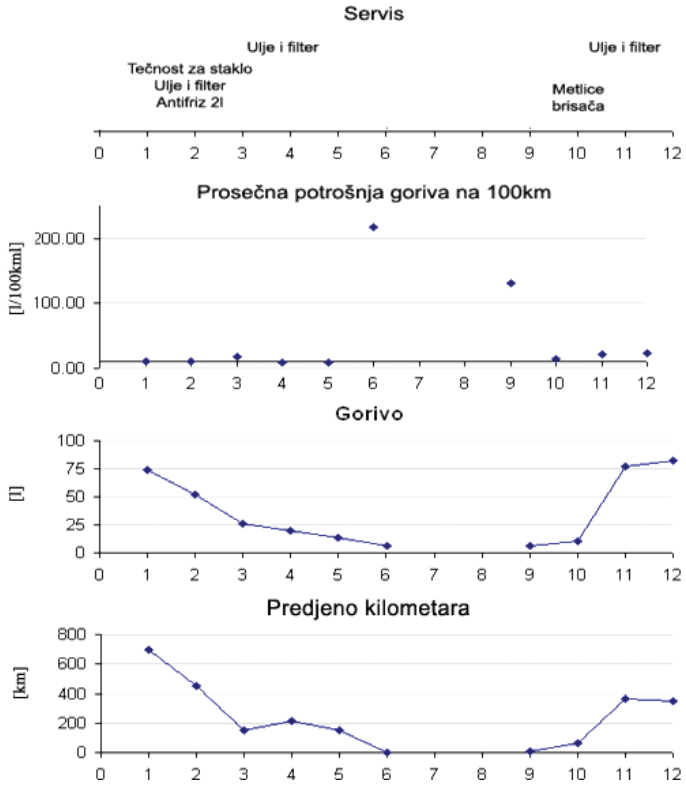
Slika 68 - *Pita dijagram*, prikaz trenutnog preseka stanja

Tip podataka i vrsta izveštaja određuju vrstu dijagrama koji će se koristi u izveštaju. *Stubasti dijagram* (engl. *Bar chart*), slika 67, je optimalan za prikaz kvantitativnih podataka i njihove distribucije. Omogućava lako upoređivanje između različitih kategorija i kada se vrednosti među njima bitno razlikuju. *Bar chart* jasno ističe razlike u veličinama, što ga čini pogodnim za analizu trendova ili distribucije podataka.

Sa druge strane, *kružni dijagram* ili *pita dijagram* (engl. *pie chart*) je pogodan za prikaz delova celine, npr. presek stanja, slika 68, gde se fokus stavlja na udeo svake kategorije u ukupnom zbiru. Ovaj dijagram se koristi kada je važno naglasiti proporcionalne odnose između manjih grupa podataka, poput tržišnih udela ili demografskih informacija. Međutim, *pie chart* može postati nejasan kada ima previše kategorija ili kada su slične vrednosti, zato što se u tom slučaju teško uočavaju razlike.

Vizuelizacija kombinuje komunikaciju, nauku o podacima i dizajn. U pitanju je suštinski važan alat u analizi i interpretaciji informacija u mnogim oblastima. Jedan od ključnih aspekata vizuelizacije podataka je identifikacija ključnih informacija, ili eventualnih odstupanja u podacima što može ukazivati na probleme koje treba istražiti. U IS, vizuelizacija značajno poboljšava proces donošenja odluka, pre svega omogućavajući bržu reakciju na promene i izazove. Ljudski mozak je prirodno predisponiran da brže obrađuje grafički prikazane informacije u odnosu na tekstualne podatke.

Razumevanje veze između radnih operacija i postignutih rezultata takođe je olakšano vizuelizacijom podataka. Kroz grafičke prikaze, korisnici mogu lako da uoče kako određene aktivnosti utiču na poslovne performanse, čime se stvara svest o tome šta funkcioniše, a šta ne. Ukoliko se korisnicima pruži mogućnost da manipulišu podacima, prilagođavaju prikaze i istražuju različite scenarije, to će im pomoći da bolje razumeju složene procese unutar organizacije.



Slika 69 - Uporedni grafički prikaz podataka za vozilo

Na slici 69 [1] se nalazi presek podataka za izabrano vozilo u okviru jedne kalendarske godine, sa uporednim prikazom servisnih podataka, prosečne potrošnje i nabavke goriva, kao i podaci o pređenoj kilometraži. Na grafičkom prikazu su primetna odstupanja od prosečnih vrednosti. Ukoliko se eliminiše mogućnost statističke greške, odstupanja mogu da ukažu na:

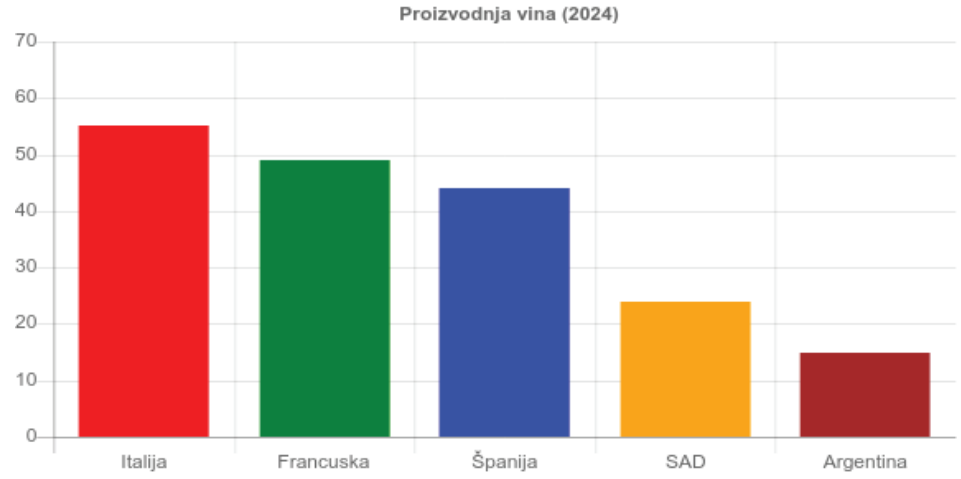
- tehničke probleme
- neispravan, neredovan ili zlonameran unos podataka
- kvalitet goriva i potrošnog materijala
- zloupotrebe prilikom korišćenja ili održavanja

Detaljnijim uvidom u podatake u relevantnom modulu IS, kao i unakrsnom verifikacijom sa podacima iz kompanijskog ERP sistema, moguće je dobiti dublji uvid u suštinu problema.

8.5.1 Alati za vizuelizaciju

Alat za vizuelizaciju podataka u kontekstu veb aplikacija i veb IS su softverske biblioteke koja omogućavaju predstavljanje podataka na grafički način. Ovi alati pomažu u analizi i interpretaciji podataka, olakšavajući korisnicima da brzo razumeju informacije kroz različite vizuelne formate, kao što su dijagrami, grafikon, karte i infografike.

U veb aplikacijama se ovakvi alati najčešće integrišu u korisnički interfejs, omogućavajući korisnicima interakciju kroz filtriranje i sortiranje podataka. U okviru IS, ovi alati mogu podržati donošenje odluka, omogućavajući analitičarima i menadžerima da brzo identifikuju trendove, obrasce i anomalije u velikim skupovima podataka. U tabelama 12 i 13 su navedeni alati koji se najčešće koriste za ovu svrhu.



Slika 70 - Stubasti dijagram (*bar chart*), *Chart.js*

Chart.js je besplatna biblioteka otvorenog koda, napisana u JavaScript programskom jeziku, koja omogućava kreiranje različitih tipova grafova i dijagrama u veb aplikacijama. Osnovna prednost ove biblioteke jeste jednostavnost upotrebe, što je uslov za brzu integraciju u projekte. Biblioteka koristi HTML5 canvas element za renderovanje grafika, uz visok kvalitet vizuelizacije i dobre performanse.

```html
<html>
<script src=
    "https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js"
></script>
<style>
    canvas {
        width: 100%;
        max-width: 600px;
    }
</style>
<body>
<canvas id="bar-chart-vertical"></canvas>
<script>
var xValues = ["Italija", "Francuska", "Španija", "SAD", "Argentina"];
var yValues = [55, 49, 44, 24, 15];
var barColors = ["red", "green", "blue", "orange", "brown"];
new Chart("bar-chart-vertical", {
    type: "bar",
    data: {
        labels: xValues,
        datasets: [{
            backgroundColor: barColors,
            data: yValues
        }]
    },
    options: {
        legend: {
            display: false
        },
        title: {
            display: true,
            text: "Proizvodnja vina (2024)"
        },
        scales: {
            yAxes: [{
                ticks: {
                    beginAtZero: true,
                    min: 0,
                    max: 70
                }
            }]
        }
    }
});
</script>
</body>
</html>
```

Listing 20 - Integracija *Chart.js* stubastog dijagrama u HTML stranicu

Chart.js takođe podržava animacije, interaktivnost i prilagodljive opcije stilizovanja, čime korisnicima omogućava da prilagode izgled grafika prema svojim potrebama. Na slici 70 je prikazan primer stubastog dijagrama generisanog pomoću ove biblioteke, dok je na listingu 20 prikazan prateći kod.

Tabela 12 - Komercijalni alati za vizuelizaciju podataka

Ime alata i tržišni udeo	Licenca i cena
Tableau (~20%)	Plaćena (od $70/mesečno po korisniku) tableau.com
Qlik Sense (~10%)	Plaćena (od $30/mesečno po korisniku) qlik.com
Power BI Pro (~15%)	Plaćena (od $20/mesečno po korisniku) powerbi.microsoft.com
Sisense (~5%)	Plaćena (cena po dogovoru) sisense.com
Looker (~7%)	Plaćena (cena po dogovoru) looker.com
MicroStrategy (~3%)	Plaćena (cena po dogovoru) microstrategy.com
TIBCO Spotfire (~4%)	Plaćena (od $85/mesečno po korisniku) tibco.com

Tabela 13 - Besplatni JavaScript alati za vizuelizaciju podataka

Ime alata i tržišni udeo	Licenca
Chart.js (~8%)	Besplatan (MIT licenca) chartjs.org
D3.js (~10%)	Besplatan (MIT licenca) d3js.org
Plotly.js (~5%)	Besplatan (MIT licenca); Pro verzija dostupna plotly.com/javascript
Highcharts (~15%)	Besplatan za nekomercijalnu upotrebu; Plaćena licenca za komercijalnu upotrebu highcharts.com
ApexCharts (~5%)	Besplatan (MIT licenca) apexcharts.com
Google Charts (~20%)	Besplatan (korisnička licenca) developers.google.com/chart
ECharts (~7%)	Besplatan (Apache 2.0 licenca) echarts.apache.org

PHPChart je biblioteka za vizualizaciju podataka napisana u PHP programskom jeziku. Podržava različite tipove grafikona uključujući stubaste, linijske i kružne. Često se koristi u veb aplikacijama za dinamičko generisanje grafika na osnovu podataka iz baze podataka ili drugih izvora.

Za razliku od *Chart.js* i drugih biblioteka napisanih u *JavaScript-u* koje funkcionišu na *klijentskoj strani*, *PHPChart* se koristi prvenstveno na *serverskoj strani*. Međutim, rezultati koje generiše mogu da se prikažu na *klijentskoj strani* kao deo veb stranice.

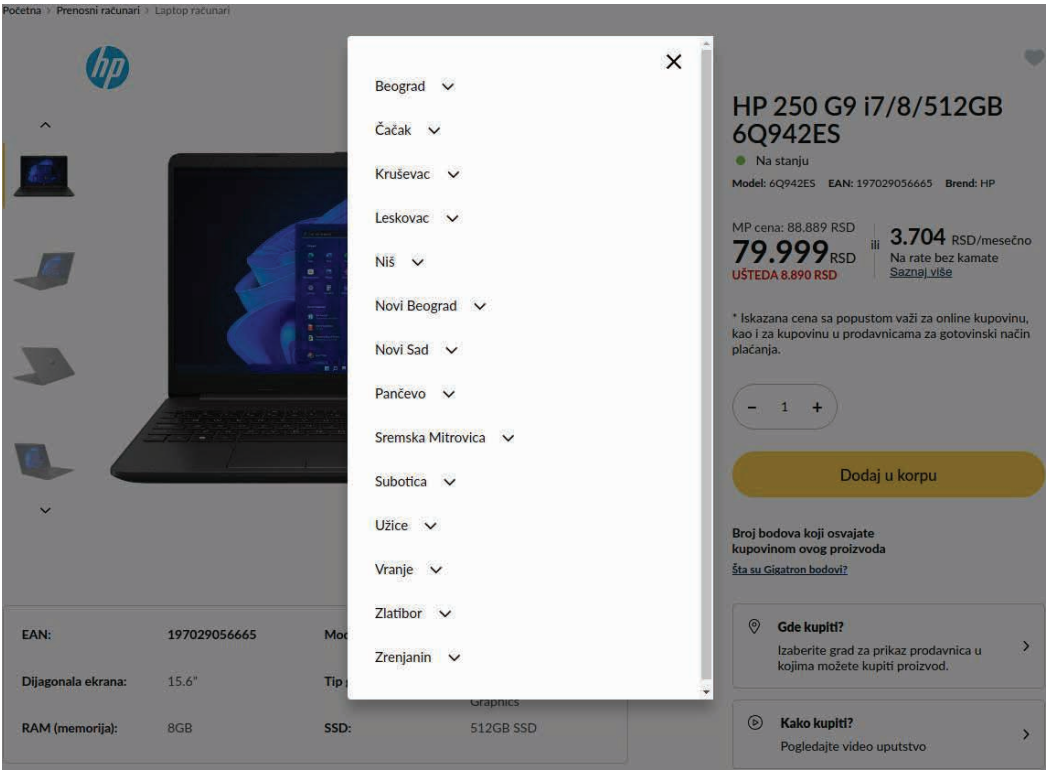
PHPChart se ne ubraja među najpopularnije alate u ovom domenu, ali ima važnu ulogu u PHP projektima gde postoji potreba za jednostavnim rešenjem za vizualizaciju na *serverskoj strani*.

PRIMERI INFORMACIONIH SISTEMA

9. PRIMERI INFORMACIONIH SISTEMA

9.1 Menadžment maloprodaje

Online prodavnice pružaju mogućnost pojedincima i kompanijama da dopru do većeg broja potrošača, na nacionalnom, međunarodnom i globalnom nivou. Mnoge kompanije koje imaju fizičke prodavnice paralelno razvijaju segment online prodaje, uz integraciju sa IS kako bi optimizovale svoje poslovanje i unapredile korisničko iskustvo potrošača.



Slika 71 - Integracija veb prodavnice sa stanjem zaliha
Prikaz spiska fizičkih prodavnica u kojima je proizvod na stanju

Integracija IS u online prodavnicu omogućava centralizovano upravljanje svim aspektima poslovanja, od upravljanja zalihama i obrade narudžbi u realnom vremenu do analize podataka o potrošačima. Time se smanjuje rizik od prekomernih ili nedovoljnih zaliha, kao i troškovi poslovanja, dok menadžmentu omogućava da brzo reaguje na promene u potražnji i trendovima, prilagodi asortiman proizvoda ili organizuje promotivne aktivnosti u skladu sa potrebama tržišta.

IS omogućava integraciju procesa obrade narudžbina iz online prodavnice sa fizičkim prodavnicama, kao i mogućnost analize podataka o prodaji, zalihama i kupcima, slika 71. Tehnologije ML i AI omogućavaju da se posetiocima online prodavnice prikazuju personalizovane preporuke proizvoda, ponude i drugi sadržaji koji odgovaraju njihovim individualnim interesovanjima, čime se povećava šansa za konverziju i jača lojalnost prema kompaniji ili brendu.

IS može da pomogne i u vođenju efikasnih marketinških kampanja. Analitika podataka o kupcima vodi ka dubljem razumevanju ciljnih grupa, identifikaciji ključnih segmenata i pravilnoj evaluaciji rezultata kampanje. U tabeli 14 su navedene popularne veb aplikacije sa elementima IS za upravljanje maloprodajom.

Tabela 14 - Veb bazirana rešenja za upravljanje veb prodavnicama

Ime sistema	Opis
Shopify	Kreiranje i upravljanje veb prodavnicama shopify.com | Ruby
WooCommerce	Veb prodavnica, plugin za WordPress woocommerce.com | PHP
Magento	Napredna platforma za e-trgovinu, idealna za velike maloprodaje. magento.com | PHP
Salesforce Commerce Cloud	Klaud rešenja za menadžment maloprodaje salesforce.com/products/commerce-cloud | Apex, JavaScript
Square	Platforma za online prodaju i upravljanje maloprodajom. squareup.com | Ruby, JavaScript
BigCommerce	Klaud platforma za upravljanje veb prodavnicama. bigcommerce.com | PHP, JavaScript
PrestaShop	e-trgovina prestashop.com | PHP

9.1.1 eCommerce izveštaji

Izveštavanje se u veb prodavnici koristi za analizu poslovnih parametara, odnosno pomažu menadžerima da razumeju ponašanje kupaca, procene kvalitet svoje ponude, identifikuju trendove i optimizuju strategije prodaje, slika 72.

Neto profit predstavlja razliku između ukupnog prihoda i troškova. Ova metrika pomaže u proceni finansijskog zdravlja veb prodavnice i efikasnosti poslovanja.

Bounce rate meri procenat posetilaca koji napuste sajt nakon pregleda samo jedne stranice. Visok bounce rate može ukazivati na loše korisničko iskustvo ili neodgovarajući sadržaj.

Demografija i starosna struktura klijenata. Razumevanje demografskih podataka, uključujući starosnu strukturu, omogućava prilagođavanje marketinških strategija i ponuda ciljnim grupama.

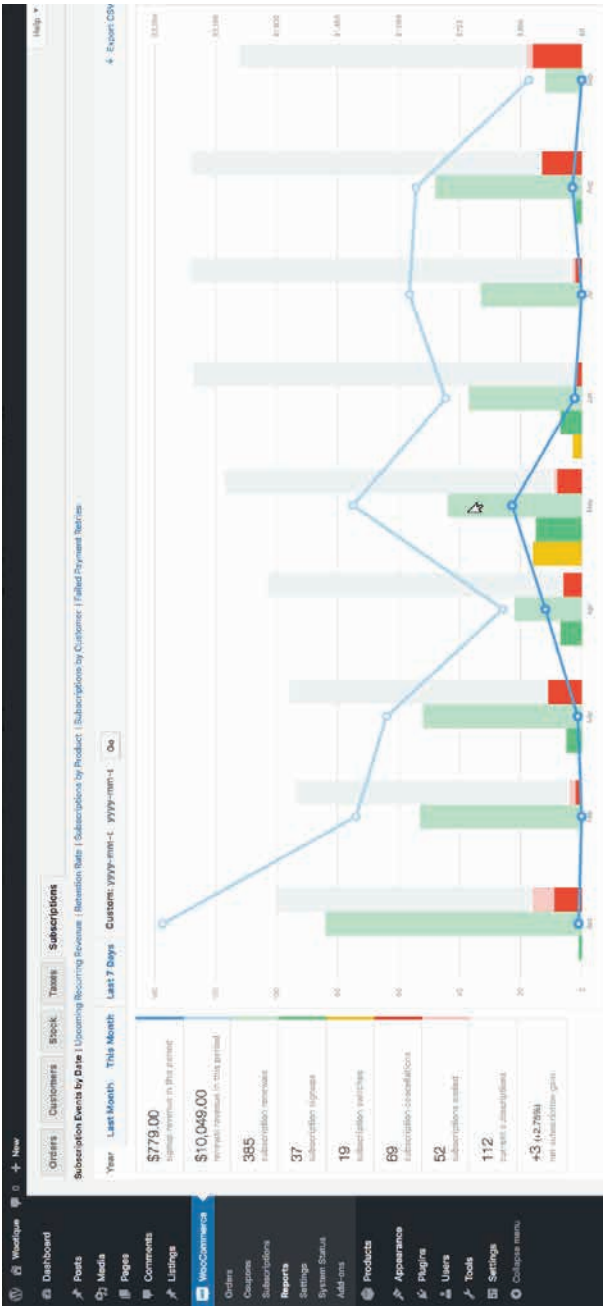
Geografska raspodela klijenata pokazuje gde se nalaze kupci, što može pomoći u optimizaciji dostavnih strategija i marketinških kampanja prema regionima.

Odnos novih klijenata i povratnika. Praćenje proporcija novih klijenata u odnosu na povratnike pomaže u oceni lojalnosti kupaca i uspešnosti strategija zadržavanja.

Upotreba brauzera i tipova uređaja pruža informacije o vrsti brauzera i uređaja (desktop, tablet, mobilni) koje kupci koriste za pristup prodavnici, na osnovu čega potom može da se optimizuje veb sajt.

Referali. Praćenje referalnih sajtova omogućava identifikaciju izvora saobraćaja, što može pomoći u usmeravanju marketinških napora.

Brzina veb sajta. Brzina učitavanja sajta direktno utiče na korisničko iskustvo i stopu konverzije. Sporiji sajtovi mogu dovesti do gubitka posetilaca.



Slika 72 - Izveštaj o pretplatama, WooCommerce
(engl. *Subscription Events by Date*)

Analitika *ključnih reči i termina* koji se pretražuju pomaže u razumevanju šta kupci traže, omogućavajući optimizaciju sadržaja i SEO strategija.

Broj pregledanih strana i vreme provedeno na sajtu. Veći broj pregledanih stranica često ukazuje na interesovanje kupaca, dok duže vreme na sajtu signalizira kvalitetno korisničko iskustvo. Ove informacije mogu da pomognu u identifikaciji popularnih proizvoda i optimizaciji navigacije radi unapređenja konverzije.

Cross-selling strategija uključuje preporučivanje dodatnih proizvoda kupcima na osnovu njihovih trenutnih ili prethodnih kupovina, čime se povećava ukupna vrednost narudžbine.

Prosečna vrednost narudžbine (engl. *average order value*, AOV) pomaže u oceni koliko kupci troše po transakciji, i može potom da se iskoristi prilikom planiranja budžeta i strategije rasta.

Stopa konverzije pokazuje procenat posetilaca koji su izvršili kupovinu, i koristi se za procenu uspešnosti marketinga i dizajna sajta.

$$R_{konverzije} = \frac{n_{prodaja}}{n_{poseta}}$$

Stopa napuštanja korpe meri procenat kupaca koji dodaju proizvode u korpu, ali ne završe kupovinu. Razumevanje razloga za napuštanje korpe može pomoći u smanjenju gubitaka i poboljšanju korisničkog iskustva.

$$R_{napu\,štanja} = \frac{n_{nekompletno}}{n_{zapo\,četo}}$$

9.2 Proizvodni sektor

U proizvodnom sektoru, IS mogu da poboljšaju efikasnost i produktivnost, pojedinih proizvodnih procesa, tako i organizacije u celini. Omogućavaju upravljanje proizvodnjom, uključujući planiranje i kontrolu proizvodnih procesa, kao i optimizaciju resursa i raspoređivanje zadataka. Sistemi za upravljanje inventarom prate nivoe zaliha i automatski naručuju potrebne

materijale, dok ERP sistem integriše različite funkcije, kao što su finansije i ljudski resursi, radi bolje koordinacije.

Pored toga, IS omogućavaju kontrolu kvaliteta proizvoda, analizu podataka i praćenje stanja opreme, što doprinosi smanjenju troškova i povećanju efikasnosti. Automatizacija procesa smanjuje potrebu za ručnim radom, dok analitički alati pomažu menadžmentu u donošenju informisanih odluka. Kao primer, mogu da se navedu izveštaji o proizvodnji, radnim časovima, radu mašine, potrošnji goriva i stanju zaliha.

Izveštaj o proizvodnji pruža pregled proizvodnih aktivnosti, uključujući količine proizvedenih proizvoda, efikasnost proizvodnih linija i postignute ciljeve. Analiza podataka u izveštaju pomaže menadžmentu da optimizuje proizvodne procese.

Izveštaj o radnim časovima prikazuje vreme koje su zaposleni proveli na različitim zadacima ili projektima. Ova informacija omogućava praćenje produktivnosti, planiranje rasporeda rada i identifikaciju potreba za obukom.

Izveštaj o radu mašine beleži učinak mašine, uključujući vreme rada, zastoje i održavanje. Analiza ovih podataka pomaže u proceni efikasnosti opreme i u planiranju preventivnog održavanja.

Izveštaj o potrošnji goriva prikazuje podatke o potrošnji goriva za vozila ili mašine u proizvodnji, što je važno za optimizaciju troškova i identifikaciju mogućnosti za uštede.

Izveštaj o stanju zaliha pruža pregled trenutnih zaliha sirovina i gotovih proizvoda. Praćenje stanja zaliha pomaže u optimizaciji nabavke i smanjenju troškova skladištenja, osiguravajući da proizvodnja ne bude ometena nedostatkom materijala.

9.3 Loyalty programi

Program lojalnosti korisnika (engl. *Loyalty program*) predstavlja strategiju koju kompanije koriste kako bi nagradile i zadržale svoje kupce. Ovi

programi motivišu kupce da se vraćaju i ponovo kupuju proizvode ili usluge, nudeći razne benefite poput bodova, popusta ili ekskluzivnih ponuda.

Da bi proces bio efikasan, kompanija mora da postavi jasne ciljeve i prikupljaju podatke o ponašanju kupaca, fokusirajući se na metrike poput stope zadržavanja, prosečne vrednosti kupovine i učestalost kupovine. Segmentacija kupaca, uz prikupljanje povratnih informacija, omogućava dublje razumevanje njihovih potreba. Redovno izveštavanje o rezultatima analize i prilagođavanje programa na osnovu dobijenih podataka pomaže u kreiranju još privlačnijih i uspešnijih strategija lojalnosti.

9.4 Sportski informacioni sistemi

Osnovna namena sportskih IS je prikupljanje, analiza i distribucija podataka vezanih za performanse sportista, statistiku i vesti. Ovi sistemi omogućavaju korisnicima, uključujući medije, trenere i organizacije, da optimizuju strategije, unaprede treninge i pruže kvalitetnije izveštavanje. Njihove funkcije obuhvataju analizu fizičkih aktivnosti i praćenje rezultata u realnom vremenu.

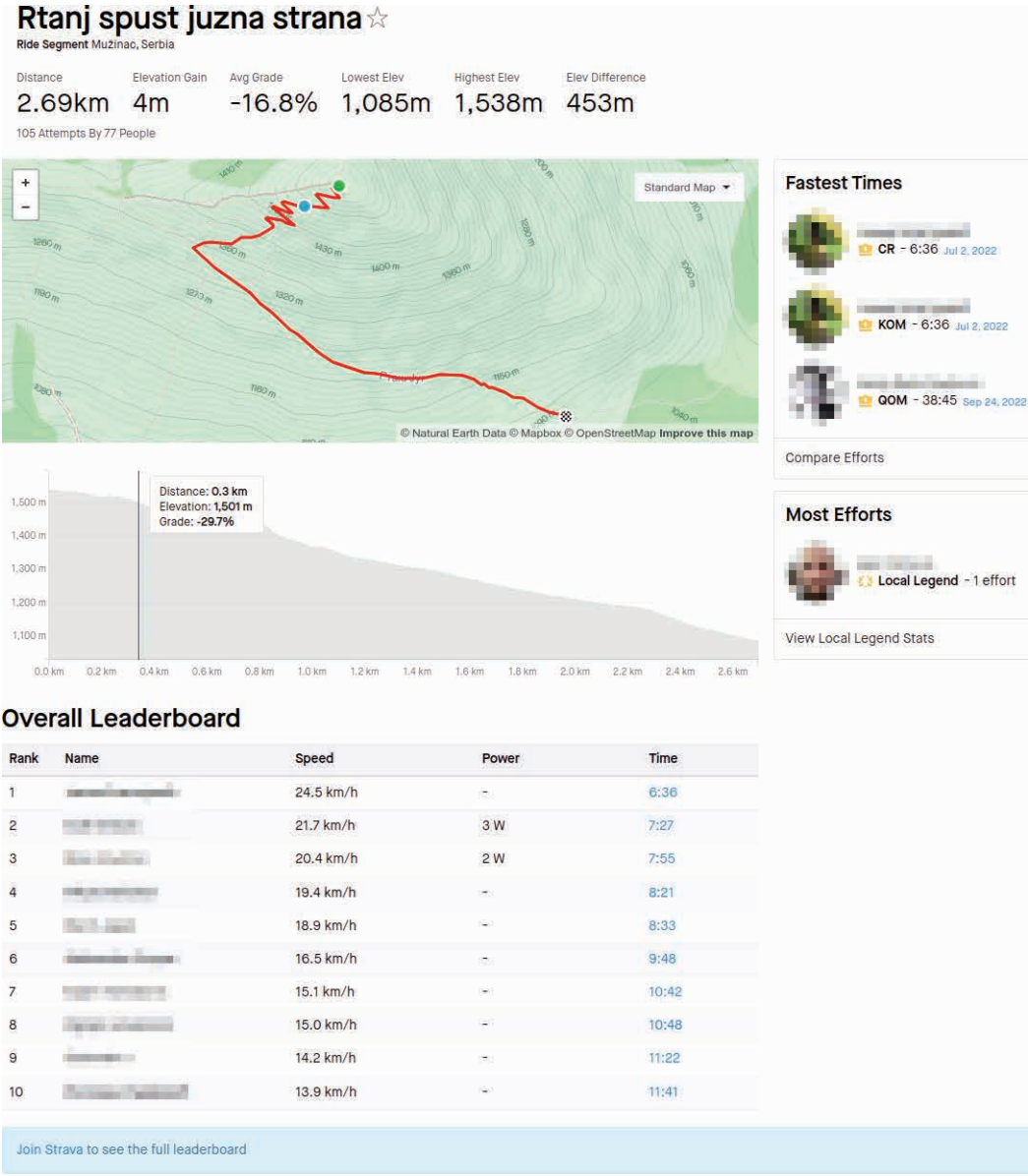
U tabeli 15 su navedena neka od trenutno popularnih rešenja u ovom domenu. Ovakvi sistemi se koriste u sportskim organizacijama, klubovima i savezima, ali ih koriste i mediji radi efikasnog vođenja sportskih aktivnosti, pružanja relevantnih informacija o sportskim događajima, kao i platforme za klađenje sa ciljem automatizacije analitike.

U razvoju ovakvih IS, bitnu ulogu igraju senzori i nosiva tehnologija (engl. *wearable technology*) poput GPS uređaja, akcelerometara i pametnih satova, koji omogućavaju precizno praćenje kretanja i fizičkih performansi sportista u realnom vremenu. Takođe, neizbežna je primena analitičkih tehnika, AI i ML za interpretaciju složenih podataka i predviđanje rezultata, čime se olakšava donošenje odluka i optimizacija strategija.

Metodologije za prikupljanje i analizu podataka, uključujući video analizu i kvantitativne i kvalitativne metode, doprinose sveobuhvatnom razumevanju sportskih performansi i strategija. [7]

Tabela 15 - Veb bazirani sportski IS

Ime Sistema veb adresa	**Opis** Programski jezik	**Ključne Funkcije** Ciljna grupa
TeamSnap teamsnap.com	Upravljanje timovima i komunikacija među igračima PHP, JavaScript	Zakazivanje, evidencija, chat Treneri i sportisti
Sportlyzer sportlyzer.com	Praćenje napretka sportista i organizacija treninga JavaScript, Ruby	Statistika, planiranje treninga Treneri i sportisti
Hudl hudl.com	Analiza video snimaka i statistika Java, JavaScript	Video analiza, statistički izveštaji Treneri i analitičari
Crossover crossover.com	Analiza video snimaka i performansi Python, JavaScript	Video analiza, statistički izveštaji Treneri i analitičari
Catapult Sports catapultsports.com	Praćenje performansi sportista C++, JavaScript	Monitoring fizičke aktivnosti, analize Treneri i fizioterapeuti
AthleteMonitoring athletemonitoring.com	Praćenje fizičkih performansi sportista PHP, JavaScript	Monitoring fizičkog stanja, izveštaji Treneri i fizioterapeuti
OpenSports opensports.net	Platforma za organizaciju sportskih događaja Ruby on Rails	Registracija, praćenje rezultata Organizatori i učesnici
Playpass playpass.com	Upravljanje članarinama i članovima JavaScript, Ruby	Članstvo, finansijsko upravljanje Sportske organizacije
LeagueApps leagueapps.com	Platforma za organizaciju liga i timova Ruby on Rails	Zakazivanje, komunikacija, statistika Organizatori i timovi
SportsEngine sportsengine.com	Upravljanje sportskim organizacijama PHP, JavaScript	Registracija, praćenje rezultata, komunikacija Organizatori i timovi
Fanalytics fanalytics.io	Prikaz podataka o gledanosti i interakciji Java, JavaScript	Analiza podataka, izveštaji o angažmanu Klubovi i organizacije
ESPN espn.com	Vesti, statistika, analize PHP, JavaScript	Vesti, rezultati, analize Navijači i mediji
SportsRadar sportsradar.com	Sportski podaci i statistike Java, Python	Statistika u realnom vremenu, izveštaji Klubovi i kladionice
Opta Sports optasports.com	Sportska statistika i analize Java, Python	Statistika, analize igre Mediji i analitičari
The Athletic theathletic.com	Analiza i vesti PHP, JavaScript	Vesti, analize, intervjui Navijači i mediji



Slika 73 - Vizuelizacija GPS log-a

Na slici 73 je prikazan javno dostupan izveštaj formiran na osnovu *GPS* log-a generisanog prilikom biciklističkog spusta niz planinu Rtanj. Izveštaj sadrži podatke o:

- dužini staze, prosečnom nagibu staze, nadmorskim visinama početne i krajnje tačke spusta i visinskoj razlici,
- mapu sa iscrtanom putanjom
- dijagram sa elevacijom, nagibom u svakoj tački pređenog puta
- tabelu sa podacima o sportisti, prosećnoj brzini tokom spusta i vremenu za koje je spust obavljen.

U pitanju je veb platforma koja se koristi za praćenje fizičkih aktivnosti, posebno trčanja i biciklizma. Korisnici mogu beležiti svoje treninge, analizirati performanse, postavljati ciljeve, i povezivati se s drugim sportistima.

Internet medicinskih stvari (engl. *Internet of Medical Things*, IoMT) podrazumeva medicinske uređaje i senzore, kao što su pametni narukvice i monitori vitalnih telesnih funkcija, koji se povezuju putem interneta za prikupljanje relevantnih podataka u realnom vremenu. [17] IoMT uređaji omogućavaju kontinuirano praćenje vitalnih znakova, nivoa stresa i oporavka, što omogućava trenerima i medicinskom osoblju da bolje razumeju fizičko i mentalno stanje sportista. Ovi podaci pomažu u prilagođavanju treninga individualnim potrebama, kao i u identifikaciji potencijalnih rizika od povreda. IoMT može podržati proces rehabilitacije prateći napredak sportista i omogućavajući prilagođavanje terapija.

Integracija sa sistemima za analizu performansi i upravljanje timom omogućava holistički pristup sportskom menadžmentu, uz poboljšanu komunikaciju između trenera i medicinskog osoblja.

PRIMER PROJEKTNOG ZADATKA

10. PRIMER PROJEKTNOG ZADATKA

Kao osnova za izradu veb IS može da posluži gotovo svaki CMS, uključujući i projekat opisan u udžbeniku za predmet *Veb programiranje* [18]. Zato u ovom poglavlju neće biti opisan proces razvoja cele veb aplikacije, već će fokus biti na primerima izrade delova IS primenom tehnologije i tehnika opisanih u ovom udžbeniku.

Cilj zadatka jeste da se prikažu principi razvoja segmenata IS koji omogućavaju čuvanje i analizu podataka o pregledu članaka veb sajta, omogućavajući uvid u trendove i popularnost sadržaja.

10.1 Struktura baze podataka

U okviru ovog zadatka, podaci o člancima i njihovim pregledima će se čuvati u *MySQL* relacionoj bazi podataka. Broj tabela i njihova struktura treba da omoguće lako upravljanje podacima i brzo generisanje potrebnih izveštaja.

Kako bi se osigurala efikasnost, a rizik od gubitka podataka sveo na minimum, baza će biti normalizovana, što uključuje razdvajanje informacija u različite tabele na način koji smanjuje redundanciju. Ovim pristupom se obezbeđuje da se podaci ažuriraju na samo jednom mestu, umesto u više tabela, što smanjuje mogućnost grešaka.

Za uspešnu realizaciju postavljenog zadatka, dovoljne su dve tabele u bazi podataka, listing 21:

- *clanci* - Osnovne informacije o svakom članku, kao što su ID članka, naslov, opis, datum i vreme objave, slika 74.
- *clanci_brojac_poseta* - Podaci o svakom pregledu članka, uključujući datum i vreme posete, slika 75.

```sql
CREATE TABLE `clanci` (
  `clanak_id` int(10) UNSIGNED NOT NULL,
  `clanak_naslov` varchar(255) NOT NULL,
  `clanak_opis` varchar(1024) NOT NULL,
  `clanak_datum_vreme` datetime NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

ALTER TABLE `clanci`
  ADD PRIMARY KEY (`clanak_id`);

CREATE TABLE `clanci_brojac_poseta` (
  `brojac_id` int(10) UNSIGNED NOT NULL,
  `clanak_id` int(10) UNSIGNED NOT NULL,
  `datum_vreme` datetime NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

ALTER TABLE `clanci_brojac_poseta`
  ADD PRIMARY KEY (`brojac_id`),
  ADD KEY `clanak_id` (`clanak_id`);

ALTER TABLE `clanci_brojac_poseta`
  MODIFY `brojac_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT;
```

Listing 21 - Kreiranje tabela za članke i brojač poseta, SQL upit

Svaki članak može da ima više pregleda, ali svaki pregled se odnosi na samo jedan članak, tzv. relacija *jedan prema mnogima*. Ova veza se ostvaruje preko kolone *clanak_id* u tabeli *clanci_brojac_poseta*, koja je i indeksirana radi veće brzine pretrage i efikasnije analize podataka.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	clanak_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	clanak_naslov	varchar(255)	utf8mb4_general_ci		No	None		
3	clanak_opis	varchar(1024)	utf8mb4_general_ci		No	None		
4	clanak_datum_vreme	datetime			No	None		

Slika 74 - Struktura tabele za skladištenje članaka `*clanci*`

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	brojac_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	clanak_id	int(10)		UNSIGNED	No	None		
3	datum_vreme	datetime			No	None		

Slika 75 - Struktura tabele za skladištenje podataka o poseti `*clanci_brojac_poseta*`

10.2 Generisanje test podataka

Prilikom razvoja IS, neophodno je da se obezbede odgovarajući test podaci kako bi se simulirali realni scenariji radi provere da li sistem i ugrađena analitika funkcionišu ispravno, i da bi korisnici mogli da se na adekvatan način upoznaju sa funkcionalnostima sistema. Bez test podataka, otkrivanje i ispravljanje grešaka postaje gotovo nemoguće, niti mogu da se procene performanse sistema pod različitim opterećenjima.

Takođe se mora voditi računa i o veličini skupa podataka koja igra ključnu ulogu u tačnosti rezultata analitike. Ukoliko je skup podataka isuviše mali, moguće je da nije reprezentativan. To uslovljava pristrasnost i netačne zaključke, i dovodi u putanje pouzdanost analitičkih modela. U takvim slučajevima, modeli mogu razviti lažne obrasce koji nisu realni.

Još jedna od posledica malih skupova podataka je loša generalizacija, kao i otežano otkrivanje anomalija ili neobičnih obrazaca, što može dovesti do propuštanja važnih informacija koje bi mogle da utiču na analizu.

U realnim uslovima, najbolji scenario prilikom razvoja IS jeste kada test podatke obezbeđuje kompanija za koju se IS razvija. Ukoliko to nije moguće, test podaci mogu da se generišu mašinski upotrebom namenski razvijenih skripti. Test podaci mogu i ručno da se unose u IS, ali to obično nije preporučljivo iz više razloga. Prvo, ručni unos podataka povećava rizik od grešaka izazvanih ljudskim faktorom, što može dovesti do netačnih ili nekompletnih informacija. Takođe, ovaj proces može biti vremenski veoma zahtevan, posebno kada je potrebno uneti velike količine podataka, što dodatno usporava testiranje. Koherentnost podataka takođe može biti problem, zato što ručni unos često dovodi do varijacija koje mogu uticati na analizu. Pored toga, kreiranje složenih test scenarija, koji zahtevaju međuzavisne podatke ili različite varijacije, može biti značajno teže bez automatizacije.

Na listingu 22 je prikazan primer skripte za mašinsko generisanje test podataka u tabeli *clanci_brojac_poseta.* Posete se generišu za vremenski period od 1. januara 2018. godine, pa do dana kada je skripta pokrenuta,

linije 9 i 10, pri čemu su granične vrednosti podešene kao *unix timestamp*. Skripta generiše 100 beleški o posetama, ali taj broj može da se promeni u skladu sa potrebama u liniji 14.

```php
<?php

$sql_sablon = "INSERT INTO
                `clanci_brojac_poseta`
                (`clanak_id`, `datum_vreme`)
                VALUES
                %s";
$broj_clanaka = 10;
$datum_pocetak = strtotime('Jan 1 2018');
$datum_kraj = time();
$sql_vrednosti = "(%d, '%s')";
$red = [];

for($i=1; $i<=100; $i++) {
    $clanak_id = rand(1, $broj_clanaka);
    $datum_vreme = date(
            'Y-m-d H:i:s',
            rand(
                $datum_pocetak,
                $datum_kraj
            )
        );
    $red[] = sprintf(
                $sql_vrednosti,
                $clanak_id,
                $datum_vreme
            );
}

$sql = sprintf($sql_sablon, implode(',', $red));

print $sql;
exit;

?>
```

Listing 22 - Generisanje test podataka o posetama za tabelu *clanci_brojac_poseta*

U petlji, linije 15-27, potrebne vrednosti za kolone *clanak_id* i *datum_vreme* se generišu kao slučajni brojevi, koji se potom ubacuju u šablon *$sql_vrednost* i kao string čuvaju u nizu *$red* u linijama 23-27. Napokon, sadržaj niza se imploduje i kao string ubacuje u *$sql* upit, linija 30, koji se potom prikazuje na ekranu, linija 32.



INSERT INTO `clanci_brojac_poseta` (`clanak_id`, `datum_vreme`) VALUES (9, '2019-02-19 12:18:15'),(2, '2023-04-13 11:25:05'),(1, '2019-11-20 07:47:50'),(1, '2022-01-05 19:48:56'),(5, '2019-05-16 20:42:06'),(7, '2018-04-10 12:47:04'),(6, '2020-07-20 11:24:31'),(8, '2020-01-22 23:17:58'),(1, '2023-12-02 03:14:57'),(10, '2019-11-13 13:39:13'),(7, '2024-03-03 11:35:17'),(9, '2020-09-14 14:44:43'),(2, '2020-06-29 14:56:34'),(8, '2020-10-15 04:54:47'),(8, '2023-02-02 23:42:37'),(6, '2024-05-10 09:49:40'),(8, '2024-03-06 20:36:20'),(10, '2020-08-05 02:32:21'),(4, '2023-12-11 20:22:11'),(4, '2024-05-13 01:46:27'),(5, '2022-11-25 06:20:07'),(5, '2018-06-23 06:35:11'),(7, '2021-12-08 03:46:35'),(7, '2023-02-18 06:02:49'),(2, '2021-08-04 01:58:00'),(10, '2022-05-25 14:56:37'),(6, '2023-04-11 07:54:18'),(6, '2022-05-16 21:15:47'),(4, '2018-01-21 21:08:10'),(10, '2019-07-02 04:25:38'),(5, '2018-12-26 16:54:00'),(4, '2022-06-25 11:42:24'),(9, '2024-07-16 12:34:25'),(3, '2020-07-14 01:21:23'),(9, '2024-05-04 10:52:06'),(7, '2022-01-01 01:24:58'),(10, '2023-07-11 04:30:28'),(3, '2023-02-27 19:18:48'),(8, '2020-06-23 18:37:57'),(8, '2020-12-18 15:57:22'),(6, '2024-04-15 13:53:16'),(3, '2018-03-12 01:01:03'),(9, '2021-05-10 23:21:55'),(2, '2023-08-18 16:01:38'),(2, '2023-10-05 18:22:53'),(9, '2024-06-24 19:41:26'),(3, '2018-10-20 01:15:08'),(6, '2019-12-20 13:40:55'),(2, '2021-09-28 17:20:32'),(2, '2023-12-12 23:16:30'),(3, '2018-07-18 22:49:03'),(9, '2023-01-07 08:34:25'),(9, '2020-02-09 07:03:54'),(7, '2021-09-06 10:22:21'),(6, '2021-12-16 21:19:14'),(1, '2020-03-11 00:31:56'),(5, '2020-12-26 06:36:10'),(4, '2020-05-23 03:54:37'),(7, '2020-12-25 10:05:31'),(9, '2020-01-31 17:30:39'),(8, '2019-05-25 06:20:13'),(5, '2020-05-27 05:23:02'),(5, '2019-09-02 07:00:44'),(7, '2020-09-02 02:35:20'),(8, '2024-01-25 15:41:55'),(5, '2018-03-27 08:40:56'),(1, '2021-05-22 08:50:45'),(6, '2022-06-14 23:58:42'),(10, '2018-12-07 06:52:18'),(5, '2021-01-14 00:53:02'),(10, '2020-07-22 19:58:24'),(6, '2021-03-11 00:56:38'),(4, '2022-12-19 06:29:51'),(1, '2022-12-31 03:07:47'),(5, '2019-01-23 08:34:06'),(6, '2021-07-14 14:21:07'),(3, '2022-03-13 03:26:24'),(3, '2023-08-22 21:08:45'),(7, '2022-09-17 10:25:49'),(9, '2021-12-01 23:17:17'),(3, '2018-01-26 20:11:59'),(6, '2022-12-14 11:14:07'),(1, '2023-11-27 16:30:02'),(2, '2021-03-15 15:03:04'),(2, '2021-03-15 12:01:00'),(10, '2024-04-27 20:39:34'),(2, '2022-01-04 15:45:59'),(6, '2018-10-03 04:31:53'),(9, '2022-03-11 16:58:18'),(10, '2021-01-10 14:33:57'),(2, '2020-04-26 07:47:36'),(10, '2023-01-09 10:38:23'),(2, '2020-08-09 16:20:48'),(6, '2020-09-21 12:36:17'),(9, '2023-09-06 02:22:03'),(6, '2018-07-08 09:49:54'),(2, '2024-03-11 21:29:11'),(7, '2020-10-23 20:38:32'),(6, '2018-05-15 22:09:13'),(8, '2023-06-16 09:52:19')

Slika 76 - Generisani test podaci, SQL upit



	brojac_id	clanak_id	datum_vreme
Edit Copy Delete	1	9	2019-02-19 12:18:15
Edit Copy Delete	2	2	2023-04-13 11:25:05
Edit Copy Delete	3	1	2019-11-20 07:47:50
Edit Copy Delete	4	1	2022-01-05 19:48:56
Edit Copy Delete	5	5	2019-05-16 20:42:06
Edit Copy Delete	6	7	2018-04-10 12:47:04
Edit Copy Delete	7	6	2020-07-20 11:24:31
Edit Copy Delete	8	8	2020-01-22 23:17:58
Edit Copy Delete	9	1	2023-12-02 03:14:57
Edit Copy Delete	10	10	2019-11-13 13:39:13
Edit Copy Delete	11	7	2024-03-03 11:35:17

Slika 77 - Generisani test podaci, tabela *clanci_brojac_poseta*, *phpMyAdmin*

```php
<?php

$db = mysqli_connect(
        'localhost',
        'root',
        '',
        'wbis_knjiga'
    );
if (!$db) die('MySQL server ne radi');
$sql_vrednosti = "('%s', '%s', '%s')";
$red = [];

$sql = "TRUNCATE `clanci`";
mysqli_query($db, $sql);

$sql = "SELECT
            `clanak_id`,
            min(`datum_vreme`) AS `datum_vreme`
        FROM `clanci_brojac_poseta`
        GROUP BY `clanak_id`
        ORDER BY `clanak_id`";
$result = mysqli_query($db, $sql);
while ($row = mysqli_fetch_assoc($result)) {
    $red[] = sprintf(
                $sql_vrednosti,
                "Članak {$row['clanak_id']}",
                "Opis članka",
                $row['datum_vreme']
            );
}

$sql = "INSERT INTO `clanci` (
            `clanak_naslov`,
            `clanak_opis`,
            `clanak_datum_vreme`
        ) VALUES" .
        implode(',', $red);
mysqli_query($db, $sql);
mysqli_close($db);

?>
```

Listing 23 - Generisanje test podataka o člancima za tabelu *clanci*

Na slikama 76 i 77 je prikazan rezultat rada opisane skripte, koji je dovoljno kopirati i izvršiti kao upit u aplikaciji za menadžment *MySQL* baze podataka poput *phpMyAdmin*, *Workbench* ili *Navicat*. U slučaju većih ili složenijih setova podataka, konekcija sa bazom podataka i izvršenje upita se vrše u samoj skripti, čime se eliminiše mogućnost greške izazvane ljudskim faktorom.

Na listingu 23 je prikazana skripta čiji je zadatak da generiše test podatke o člancima za tabelu *clanci*, sa donekle izmenjenim pristupom u odnosu na prethodni primer. Prvo se uspostavlja konekcija sa bazom podataka, linije 3-9. Sledi korak koji po potrebi može da se izostavi, a to je uklanjanje postojećeg sadržaja iz tabele *clanci*, linije 13 i 14.

U sledećem koraku, linije 16-30 se očitavaju podaci o posetama, kako bi se za svaki članak odredilo vreme prve posete, koje će ujedno biti i vreme kreiranja članka. Potom se formira upit koji će generisane podatke ubaciti u tabelu *clanci*, linije 32-38, i na kraju u liniji 39 se zatvara konekcija sa bazom podataka. Generisani podaci su ubačeni u tabelu *clanci*, slika 78.

	clanak_id	clanak_naslov	clanak_opis	clanak_datum_vreme
Edit Copy Delete	1	Članak 1	Opis članka	2019-11-20 07:47:50
Edit Copy Delete	2	Članak 2	Opis članka	2020-04-26 07:47:36
Edit Copy Delete	3	Članak 3	Opis članka	2018-01-26 20:11:59
Edit Copy Delete	4	Članak 4	Opis članka	2018-01-21 21:08:10
Edit Copy Delete	5	Članak 5	Opis članka	2018-03-27 08:40:56
Edit Copy Delete	6	Članak 6	Opis članka	2018-05-15 22:09:13
Edit Copy Delete	7	Članak 7	Opis članka	2018-04-10 12:47:04
Edit Copy Delete	8	Članak 8	Opis članka	2019-05-25 06:20:13
Edit Copy Delete	9	Članak 9	Opis članka	2019-02-19 12:18:15
Edit Copy Delete	10	Članak 10	Opis članka	2018-12-07 06:52:18

Slika 78 - Generisani test podaci, tabela *clanci*, *phpMyAdmin*

10.3 Izveštavanje

Kada su test podaci obezbeđeni, potrebno je da se na osnovu njih generišu izveštaji koji će pomoći u daljoj analizi. Na listingu 24 je prikazan SQL upit čiji je zadatak da iz tabele *clanci_brojac_poseta* pročita spisak članaka, pri čemu će svaki id clanka (*clanak_id*) spojiti sa brojem poseta koji je taj članak do tog trenutka imao (*brojac*).

```
1  SELECT
2      `clanak_id`,
3      COUNT(`datum_vreme`) AS `brojac`
4  FROM `clanci_brojac_poseta`
5  GROUP BY `clanak_id`
6  ORDER BY `clanak_id`
```

Listing 24 - Brojač poseta po članku, SQL upit

clanak_id ▲ 1	brojac
1	7
2	13
3	8
4	6
5	10
6	15
7	10
8	9
9	12
10	10

Slika 79 - Brojač poseta po članku, rezultat upita, *phpMyAdmin*

Rezultat upita u aplikaciji *phpMyAdmin* je prikazan na slici 79. Delovi *PHP* skripte koja generiše ovakav tabelarni prikaz se nalaze na listingu 25 i u MVC arhitekturi predstavljaju *model*, linije 1-23, i *pogled*, linije 25-48. U *modelu*, upit se formira i izvršava u linijama 11-19, a potom se očitani podaci smeštaju u promenljivu *$podaci* tipa niz. Tako pripremljene podatke potom *PHP* u *pogledu* dinamički ubacuje u *HTML* tabelu, linije 41-46.

```php
<?php

$db = mysqli_connect(
        'localhost',
        'root',
        '',
        'wbis_knjiga'
    );
if (!$db) die('MySQL server ne radi');

$podaci = [];

$sql = "SELECT
            `clanak_id`,
            COUNT(`datum_vreme`) AS `brojac`
        FROM `clanci_brojac_poseta`
        GROUP BY `clanak_id`
        ORDER BY `clanak_id`";
$result = mysqli_query($db, $sql);
while ($row = mysqli_fetch_assoc($result))
    $podaci[] = $row;

?>

<style>
    table { border-collapse: collapse; }
    th { background: silver; }
    th, td {
        border: 1px solid black;
        padding: 0.25em;
    }
</style>

<table>
<thead>
<tr>
    <th>Članak ID</th>
    <th>Brojač</th>
</thead>
<tbody>
<?php foreach($podaci AS $podatak): ?>
<tr>
    <td><?= $podatak['clanak_id'] ?></td>
    <td><?= $podatak['brojac'] ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
```

Listing 25 - Očitavanje podataka iz baze i priprema za klijentsku stranu, model

Članak ID	Brojač
1	7
2	13
3	8
4	6
5	10
6	15
7	10
8	9
9	12
10	10

Slika 80 - Rezultati upita prikazani na veb stranici u tabelarnom formatu

Rezultat rada skripte su podaci prikazani u tabelarnom formatu, slika 80. Iako je u pitanju mali skup podataka, prikazani izveštaj je nepregledan, zato što analitičaru prikazuje samo identifikacione brojeve članaka. U realnom slučaju, npr. medijskog portala, gde se na dnevnom nivou generiše oko 70 članaka, u periodu od samo dve godine funkcionisanja takvog portala, u skupu bi se našlo 51,100 članaka.

```
1  SELECT
2      c.`clanak_id`,
3      d.`clanak_naslov`,
4      d.`clanak_datum_vreme`,
5      COUNT(c.`datum_vreme`) AS `brojac`
6  FROM `clanci_brojac_poseta` c
7  LEFT JOIN `clanci` d ON c.clanak_id=d.clanak_id
8  GROUP BY `clanak_id`
9  ORDER BY `clanak_id`
```

Listing 26 - Listing članaka sa brojačem poseta, SQL upit

Da bi se formirao izveštaj koji umesto identifikacionih brojeva članaka prikazuje naslove članaka i njihov datum objave, neophodno je da se izmeni *SQL* upit na takav način da poveže podatke iz obe tabele, *clanci* i *clanci_brojac_poseta*, listing 26. Dalje je trivijalno da se upit ubaci u listing

25, i da se u skladu sa tim izmeni pogled ubacivanjem u tabelu kolona sa naslovom članka i datumom objave, listing 27 i slika 81.

```
<?php foreach($podaci AS $podatak): ?>
<tr>
    <td><?= $podatak['clanak_id'] ?></td>
    <td><?= $podatak['clanak_naslov'] ?></td>
    <td><?= date(
                'd.m.Y',
                strtotime(
                    $podatak['clanak_datum_vreme']
                )
            ) ?></td>
    <td><?= $podatak['brojac'] ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
```

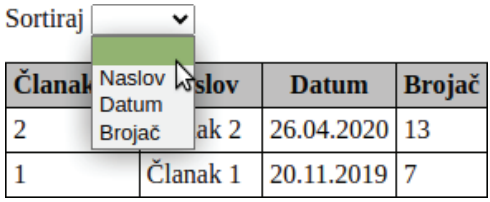
Listing 27 - Adaptacija pogleda za prikaz naslova članka i datuma objave

Članak ID	Naslov	Datum	Brojač
1	Članak 1	20.11.2019	7
2	Članak 2	26.04.2020	13
3	Članak 3	26.01.2018	8
4	Članak 4	21.01.2018	6
5	Članak 5	27.03.2018	10
6	Članak 6	15.05.2018	15
7	Članak 7	10.04.2018	10
8	Članak 8	25.05.2019	9
9	Članak 9	19.02.2019	12
10	Članak 10	07.12.2018	10

Slika 81 - Rezultati izmenjenog upita

Korisnički interfejs može da se unapredi dodavanjem jednostavnog veb obrasca za pretragu, slika 82. U pitanju je HTML element *select*, linije 60-74 na listingu 28, a pomoću *JavaScript*-a se osluškuje promena njegove vrednosti izborom jedne od ponuđenih opcija, linija 54. U tom slučaju se aktivira događaj *submit* na veb obrascu u kome se ovaj element nalazi,

linija 55, a izabrana vrednost se prenosi aplikaciji kao url parametar. Blok koda na listingu 28 može da se ubaci u kod prikazan na listingu 25 neposredno ispred HTML tabele.



Slika 82 - Upravljanje sortiranjem podataka preko *dropdown* elementa

```
<script>
    window.addEventListener('DOMContentLoaded', function () {
        var sortElement = document.getElementsByTagName('select')[0];
        var sortObrazac = document.getElementsByTagName('form')[0];
        sortElement.addEventListener('change', function () {
            sortObrazac.submit();
        });
})
</script>

<form method="get">
    Sortiraj
    <select name="sort">
        <option></option>
        <option value="naslov" <?= $select_naslov ?? '' ?>>
            Naslov
        </option>
        <option value="datum" <?= $select_datum ?? '' ?>>
            Datum
        </option>
        <option value="brojac" <?= $select_brojac ?? '' ?>>
            Brojač
        </option>
    </select>
</form>
```

Listing 28 - Sortiranje, dodavanje veb obrasca

Na listingu 29 je prikazan deo koda koji očitava URL parametar, linija 8, i u skladu sa očitanom vrednošću priprema deo *SQL* upita za sortiranje, linije 9-24, koji se u *SQL* upit ubacuje u liniji 37. Prikazani segment koda može da se ubaci u osnovu datu na listingu 25.

```php
$sql_order = "ORDER BY d.`clanak_datum_vreme` DESC";

if ($sort = ($_GET['sort'] ?? '')) {
    switch ($sort) {
        case 'naslov' :
            $sql_order = "ORDER BY d.`clanak_naslov`";
            $select_naslov = 'selected';
            break;
        case 'brojac' :
            $sql_order = "ORDER BY `brojac` DESC";
            $select_brojac = 'selected';
            break;
        case 'datum' :
        default :
            $sql_order = "ORDER BY
                                d.`clanak_datum_vreme` DESC";
            $select_datum = 'selected';
            break;
    }
}

$podaci = [];

$sql = "SELECT
            c.`clanak_id`,
            d.`clanak_naslov`,
            d.`clanak_datum_vreme`,
            COUNT(c.`datum_vreme`) AS `brojac`
        FROM `clanci_brojac_poseta` c
        LEFT JOIN `clanci` d ON c.clanak_id=d.clanak_id
        GROUP BY `clanak_id`
        {$sql_order}";
```

Listing 29 - Sortiranje, očitavanje *URL* parametra i formiranje *SQL* upita

10.4 API

U veb aplikaciji, *API* može da se iskoristi kao alat za komunikaciju, odnosno prenos podataka sa serverske strane ka klijentskoj strani, a to mogu da budu i podaci potrebni za generisanje dijagrama.

Na listingu 30 je prikazan takav primer. S obzirom da je zadatak ove skripte da obezbedi podatke koji nisu javno dostupni, neophodno je da se prvo proveri da li korisnik ima pravo pristupa.

```php
<?php

start_session();
if ($_SESSION['admin'] !== true) exit;

header('Content-Type: application/json');

$db = mysqli_connect('localhost', 'root', '', 'wbis_knjiga');
if (!$db) die('MySQL server ne radi');

$data = [];

$sql = "SELECT
            c.`clanak_id`,
            d.`clanak_naslov`,
            d.`clanak_datum_vreme`,
            COUNT(c.`datum_vreme`) AS `brojac`
        FROM `clanci_brojac_poseta` c
        LEFT JOIN `clanci` d ON c.clanak_id=d.clanak_id
        GROUP BY `clanak_id`
        ORDER BY `clanak_id`";
$result = mysqli_query($db, $sql);
while ($row = mysqli_fetch_assoc($result)) {
    $data['clanci'][] = $row['clanak_naslov'];
    $data['brojac'][] = $row['brojac'];
}

mysqli_close($db);

echo json_encode($data);

?>
```

Listing 30 - *api.php*, broj pregleda po članku

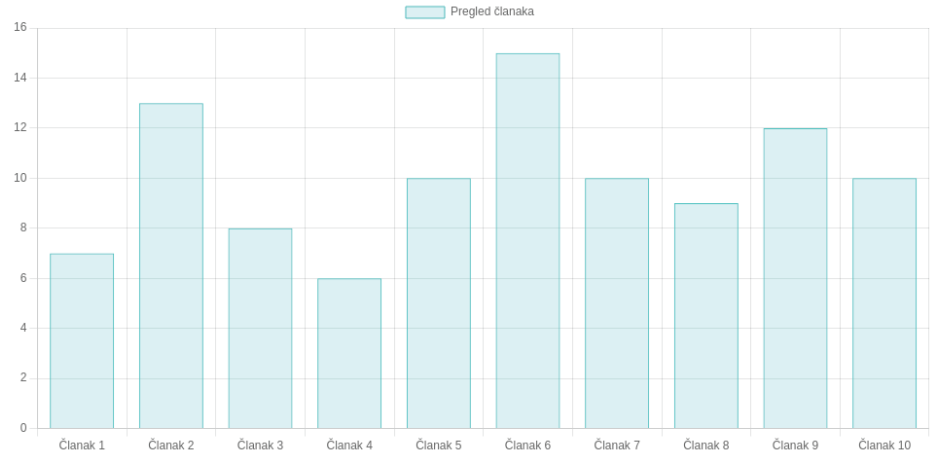
Provera se vrši pokretanjem sesije i proverom ključa *admin* u super globalnoj promenljivoj *$_SESSION*, linije 3 i 4 koji je mogao biti podešen na vrednost *true* prilikom prijave administratora. Ukoliko korisnik nije administrator, skripta se zaustavlja u liniji 4. Nakon toga sledi mehanizam koji je sličan prethodno opisanim primerima u ovom poglavlju - formira se *SQL* upit i izvršava, podaci se očitavaju i smeštaju u niz *$data*. Bitna razlika u odnosu na prethodne primere jeste u liniji 6, gde se označava da će sadržaj poslat klijentu biti u *JSON* formatu, i potom u liniji 30 gde se prikupljeni podaci ispisuju u *JSON* formatu.

10.5 Grafički prikaz, grafikoni

Grafički prikaz, odnosno vizualizacija podataka se u IS koriste prvenstveno sa ciljem da efikasnije prikažu i olakšaju razumevanje složenih podataka, i da pomognu korisnicima da identifikuju obrasce, trendove i anomalije. Ovi vizualni alati omogućuju bržu analizu velikih količina informacija, što je ključno za donošenje informisanih odluka. Mnogi alati, uključujući i *Chart.js* biblioteku koja će biti korišćena u ovom primeru, nude interaktivnost, što korisnicima omogućava interakciju sa dijagramom.

10.5.1 Broj pregleda po članku

Na slici 83 su prikazani podaci o broju pregleda članaka u bazi podataka, i ova slika je zapravo grafički prikaz podataka u tabelarnoj formi sa slike 81. Kod potreban za generisanje dijagrama je prikazan na listingu 31. *Chart.js* biblioteka se integriše u dokument u liniji 5, dok se u liniji 8 postavlja *canvas* HTML element sa jedinstvenim atributom id, na kome će dijagram biti nacrtan. U poslednjoj liniji bloka sa JavaScript kodom, linija 44 se poziva funkcija *createBarChart()*.



Slika 83 - Grafički prikaz podataka u formi stubastog dijagrama (*barchart*)

```
1  <html>
2  <head>
3      <meta charset="utf8">
4      <title>Primer stubastog dijagrama</title>
5      <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
6  </head>
7  <body>
8  <canvas id="BarChartBrojPregleda" width="400" height="200"></canvas>
9
10 <script>
11 async function fetchData() {
12     const response = await fetch('api.php');
13     const data = await response.json();
14     return data;
15 }
16
17 async function createBarChart() {
18     const data = await fetchData();
19
20     const BarChartelement = document.getElementById('BarChartBrojPregleda');
21     const ctx = BarChartelement.getContext('2d');
22     const BarChartBrojPregleda = new Chart(ctx, {
23         type: 'bar',
24         data: {
25             labels: data.clanci,
26             datasets: [{
27                 label: 'Pregled članaka',
28                 data: data.brojac,
29                 backgroundColor: 'rgba(75, 192, 192, 0.2)',
30                 borderColor: 'rgba(75, 192, 192, 1)',
31                 borderWidth: 1
32             }]
33         },
34         options: {
35             scales: {
36                 y: {
37                     beginAtZero: true
38                 }
39             }
40         }
41     });
42 }
43
44 createBarChart();
45 </script>
46 </body>
47 </html>
```

Listing 31 - *brojac-clanci.html*, stubasti dijagram (*barchart*), *Chart.js*

Podaci iz *API*-ja se dobijaju u *JSON* formatu pozivom fajla *api.php*, linije 12 i 13, i smeštaju u konstantu *data*, linija 18. Potom se inicijalizuje dijagram

na *DOM* elementu sa atributom id *BarChartBrojPregleda*, linije 20-21, dok se konfiguracija dijagrama vrši u linijama 22-41.

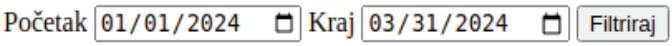
Dobijeni dijagram je pregledan i jasno pokazuje međusobni odnos broja poseta članaka u bazi podataka, ali *API* zadužen da obezbedi podatke poseduje ozbiljan nedostatak, a to je da iščitava sve članke iz baze podataka. U slučaju većeg broja članaka, dijagram će se i dalje uspešno generisati, ali uz nepotrebno opterećenje servera, i grafički prikaz će biti nepregledan usled velikog broja podataka.

Osim toga, iako će prikazani podaci na dijagramu biti tačni, oni posle određenog perioda više neće biti relevantni. Naime, *SQL* upit u *API*-ju nema vremensko ograničenje, tako da se brojač formira na osnovu svih pregleda, od trenutka kada je sistem počeo da radi. Ukoliko aplikacija funkcioniše duži niz godina, moguće je da se među člancima sa najvećim brojem pregleda nađu i članci koji zapravo dugi vremenski period nemaju niti jedan pregled, što će onemogućiti realnu analizu i uočavanje aktuelnih trendova.

```
1  SELECT
2      c.`clanak_id`,
3      d.`clanak_naslov`,
4      d.`clanak_datum_vreme`,
5      COUNT(c.`datum_vreme`) AS `brojac`
6  FROM `clanci_brojac_poseta` c
7  LEFT JOIN `clanci` d ON c.clanak_id=d.clanak_id
8  WHERE
9      c.`datum_vreme`
10         BETWEEN '2024-01-01 00:00:00'
11         AND '2024-03-31 23:59:59'
12 GROUP BY `clanak_id`
13 ORDER BY brojac
14 LIMIT 5
```

Listing 32 - API, ograničenje rezultata na 5 članaka

Opisani problemi se rešavaju jednostavnom izmenom *SQL* upita u fajlu *api.php*, listing 32. Količina podataka koji će biti prikazani na dijagramu može da se ograniči u *SQL* upitu pomoću klauzule *LIMIT*, linija 14, npr. na 5 članaka sa najmanjim (*ASC*, podrazumevana vrednost, može da se izostavi) ili najvećim (*DESC*) brojem pregleda, linija 13.

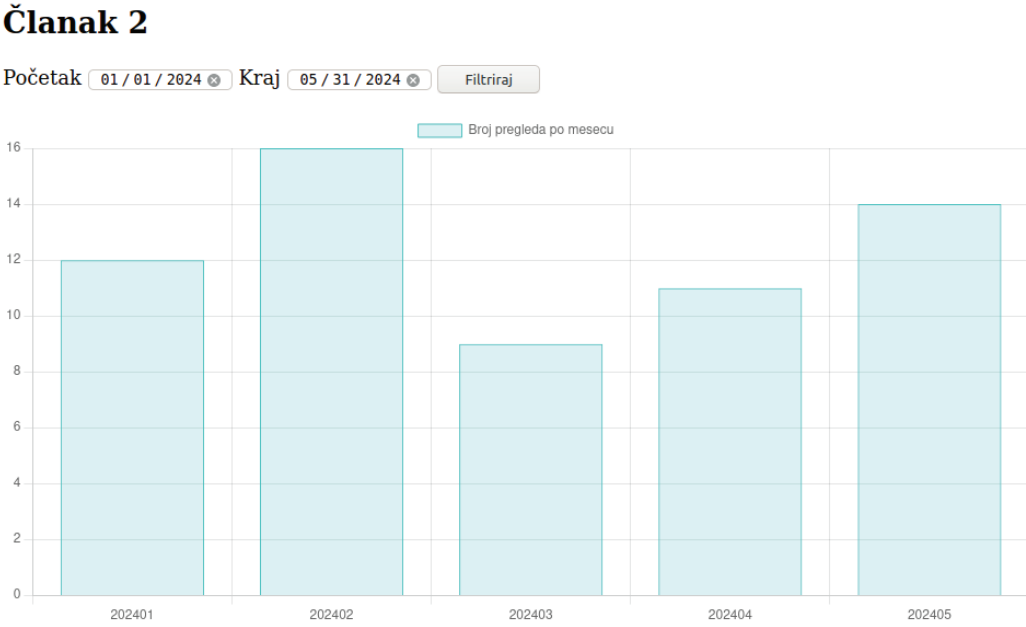


Slika 84 - Veb obrazac za izbor vremenskog perioda za analizu

Vremenski period za koji se analizira broj pregleda, može da se ograniči upotrebom *SQL* operatora *BETWEEN* u *WHERE* klauzuli, linije 8-11. Datumi početka i kraja posmatranog perioda se određuju pomoću *HTML* elemenata *input* tipa *date*, slika 84, a ka serveru se šalju *GET* metodom kao *URL* parametri.

10.5.2 Broj pregleda za izabrani članak

Da bi se generisao izveštaj o broju pregleda za izabrani članak, slika 85, potrebno je da se napravi odgovarajući *API*, listing 33.



Slika 85 - Mesečni broj pregleda za izabrani članak

API se kontroliše pomoću tri *URL parametra* - *clanak* koji odgovara koloni *clanak_id* u tabeli *clanci*, dok *pocetak* i *kraj* predstavljaju početni i krajnji datum perioda za analizu, u formatu *godina-mesec-dan*.

```php
<?php
if (!($clanak_id = (int)($_GET['clanak'] ?? ''))) exit;
$pocetak = $_GET['pocetak'] ?? '';
$kraj = $_GET['kraj'] ?? '';
$data = [];
if (!preg_match('#^[\d]{4}-[\d]{2}-[\d]{2}$#', $pocetak)
    && !preg_match('#^[\d]{4}-[\d]{2}-[\d]{2}$#', $kraj))
        exit;

$db = mysqli_connect('localhost', 'root', '', 'wbis_knjiga');
if (!$db) die('MySQL server ne radi');
$sql = "SET NAMES utf8";
mysqli_query($db, $sql);

$where_sql =
    ($pocetak != '' AND $kraj != '')
    ? "c.`datum_vreme` BETWEEN '{$pocetak}' AND '{$kraj}'"
    : "1";

$sql = "SELECT
            EXTRACT(YEAR_MONTH FROM `datum_vreme`) AS `godina_mesec`,
            COUNT(c.`datum_vreme`) AS `brojac`
        FROM `clanci_brojac_poseta` c
        LEFT JOIN `clanci` d ON c.clanak_id=d.clanak_id
        WHERE c.`clanak_id`={$clanak_id} AND {$where_sql}
        GROUP BY `godina_mesec`
        ORDER BY `godina_mesec`
        LIMIT 5";
$result = mysqli_query($db, $sql);
while ($row = mysqli_fetch_assoc($result)) {
    $data['godina_mesec'][] = $row['godina_mesec'];
    $data['brojac'][] = $row['brojac'];
}

$sql = "SELECT `clanak_naslov`
        FROM `clanci`
        WHERE `clanak_id`=$clanak_id
        LIMIT 1";
$result = mysqli_query($db, $sql);
$clanak = mysqli_fetch_assoc($result);

$data['naslov'] = $clanak['clanak_naslov'];

mysqli_close($db);

header('Content-Type: application/json; charset=utf-8');
echo json_encode($data);exit;
?>
```

Listing 33 - *api-clanak.php*, mesečni broj pregleda za izabrani članak

Provera da li je *id članka* ispravno zadat kao pozitivna celobrojna vrednost se vrši u liniji 2, dok se u linijama 6 i 7 vrši provera pomoću regularnih izraza da li su početni i kranji datumi zadati u očekivanom formatu.

U liniji 10 se uspostavlja veza sa bazom podataka. Ukoliko je veza uspostavljena, komunikacija sa bazom podataka se podešava na karakter set *utf8*, linije 12-13. Potom se u linijama 15-18 formira deo *SQL* upita koji će biti uključen u *WHERE* klauzulu u liniji 25.

Nakon što se *SQL* upit formira i izvrši, očitani podaci o broju poseta po mesecu se smeštaju u višedimenzioni niz *$data*. Podatak koji nedostaje jeste naslov članka, očitava se upotrebom još jednog *SQL* upita, linije 35-40, i takođe se smešta u niz *$data*, linija 42.

Na kraju, zatvara se konekcija sa bazom podataka, linija 44, format odgovora servera se podešava na *JSON* format i *utf8* karakter set, linija 46, i niz *$data* se transformiše u *JSON* format i štampa, linija 47.

Deo izveštaja na klijentskoj strani je prikazan na listinzima 34 i 35. U pitanju je jedan fajl, *brojac-clanak.php*, ali je kod podeljen na dva listinga radi lakšeg razumevanja.

```
1  <?php
2  $clanak_id = (int)($_GET['clanak_id'] ?? '');
3  if (!$clanak_id) exit;
4
5  $pocetak = $_GET['pocetak'];
6  $kraj = $_GET['kraj'];
7  $url =  "api-clanak.php?" .
8          "clanak={$clanak_id}" .
9          "&pocetak={$pocetak}" .
10         "&kraj={$kraj}";
11 ?>
```

Listing 34 - *brojac-clanak.php*, očitavanje URL parametara

Na početku dokumenta u linijama 1-11, listing 34, je smešten *PHP* segment sa zadatkom da na klijentskoj strani očita vrednosti URL parametara i na osnovu njih formira *URL* adresu za poziv *API*-ja, linije 7-10.

```
<html>
<head>
    <meta charset="utf8">
    <title>Primer stubastog dijagrama</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
<h2></h2>
<form method="get">
    Početak
    <input type="date" name="pocetak" value="<?= $_GET['pocetak'] ?? '' ?>">
    Kraj
    <input type="date" name="kraj" value="<?= $_GET['kraj'] ?? '' ?>">
    <input type="hidden" name="clanak_id" value="<?= $clanak_id ?>">
    <button>Filtriraj</button>
</form>
<canvas id="BarChartBrojPregleda" width="400" height="200"></canvas>
<script>
async function fetchData() {
    const response = await fetch('<?= $url ?>');
    const data = await response.json();
    return data;
}
async function createBarChart() {
    const data = await fetchData();
    document.getElementsByTagName('h2')[0].innerHTML = data.naslov;
    const BarChartelement = document.getElementById('BarChartBrojPregleda');
    const ctx = BarChartelement.getContext('2d');
    const BarChartBrojPregleda = new Chart(ctx, {
        type: 'bar',
        data: {
            labels: data.godina_mesec,
            datasets: [{
                label: 'Broj pregleda po mesecu',
                data: data.brojac,
                backgroundColor: 'rgba(75, 192, 192, 0.2)',
                borderColor: 'rgba(75, 192, 192, 1)',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                y: {
                    beginAtZero: true
                }
            }
        }
    });
}
createBarChart();
</script>
</body>
</html>
```

Listing 35 - *brojac-clanak.php*, veb obrazac za filtriranje podataka

U liniji 16, listing 35, se u stranicu integriše *chart.js*, dok se u liniji 28 formira *HTML canvas* element koji će biti iskorišćen za crtanje stubastog dijagrama. U liniji 19 se nalazi prazno *h2* zaglavlje, u koje će pomoću *JavaScript*-a biti unet naslov članka. Veb obrazac koji se koristi za podešavanje vremenskog perioda za izveštaj se nalazi u bloku linija 20-27.

Kada su potrebni DOM elementi formirani, poziva se *JavaScript* funkcija *createBarChart*, linija 61. Ka *API*-ju se šalje asihroni poziv, a odgovor se smešta u konstantu *data*, linija 36. Naslov članka se nalazi u svojstvu *data.naslov* i smešta se u *h2* element, linija 37. Dalje se inicijalizuje stubasti dijagram na isti način kako je opisano u prethodnim primerima.

Alternativno, čitanje *URL* parametara i formiranje *URL* adrese za API poziv umesto na serverskoj strani kao što je prikazano na listingu 34, može da se obavi na klijentskoj strani, listing 36.

```
<script>
const queryString = window.location.search;
const urlParams = new URLSearchParams(queryString);
const clanak = urlParams.get('clanak');
const pocetak = urlParams.get('pocetak');
const kraj = urlParams.get('kraj');
const url = "api-clanak.php?" +
            "clanak=" + clanak +
            "&pocetak=" + pocetak +
            "&kraj=" + kraj;

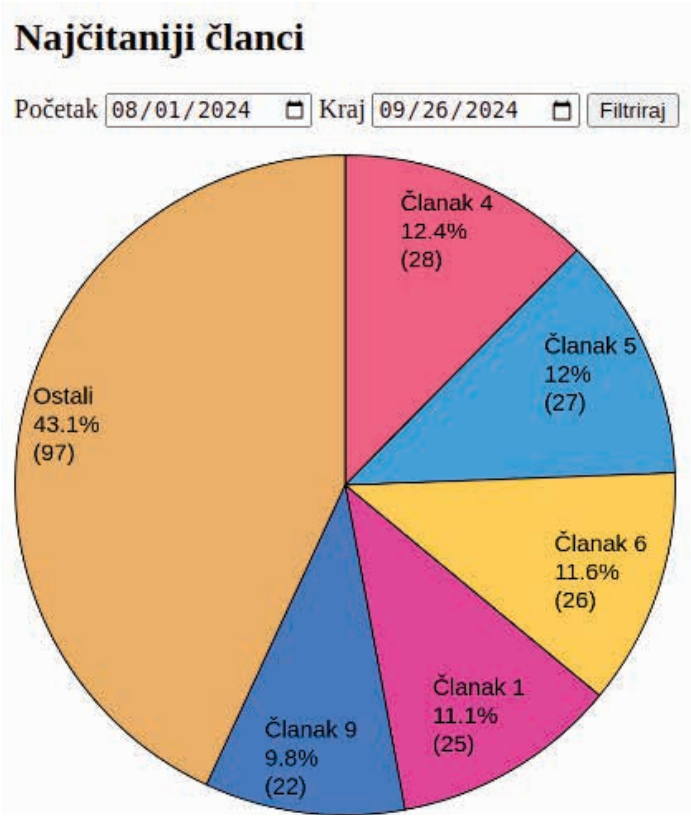
async function fetchData() {
    const response = await fetch(url);
    const data = await response.json();
    return data;
}
```

Listing 36 - Očitavanje URL parametara, *JavaScript*

10.5.3 Presek stanja

Za vizuelizaciju izveštaja sa presekom stanja se obično koristi kružni grafikon, koji proporcionalno ilustruje raspodelu različitih kategorija unutar jednog skupa podataka. Na slici 86 je prikazan međusobni odnos broja poseta 5 najposećenijih članaka u izabranom vremenskom periodu, uz

dodatno šesto polje koje predstavlja zbirni broj pregledanih stranica svih ostalih članaka u bazi podataka u istom periodu.



Slika 86 - Najčitanijih 5 članaka u izabranom periodu
Kružni dijagram (engl *PieChart*)

Podaci za kružni dijagram se dobijaju u *JSON* formatu pozivom na odgovarajući *API*, u ovom slučaju je to dokument *api-procentualno.php*, prikazan radi preglednosti na listinzima 37 i 38.

Arhitektura skripte je slična kao i u prethodnim primerima. Prvo se očitavaju vrednosti *URL parametara pocetak* i *kraj*, linije 2 i 3, a potom se proverava pomoću regularnih izraza, linije 8 i 9, da li su zadate u očekivanom formatu godina-mesec-dan (*YYYY-MM-DD*). Ukoliko su format ili vrednosti neispravne, skripta se prekida, linija 10.

```php
<?php
$pocetak = $_GET['pocetak'] ?? '';
$kraj = $_GET['kraj'] ?? '';
$data = [];
$procenat = 0;
$brojac = 0;

if (!preg_match('#^[\d]{4}-[\d]{2}-[\d]{2}$#', $pocetak)
    && !preg_match('#^[\d]{4}-[\d]{2}-[\d]{2}$#', $kraj))
        exit;

$db = mysqli_connect('localhost', 'root', '', 'wbis_knjiga');
if (!$db) die('MySQL server ne radi');
$sql = "SET NAMES utf8";
mysqli_query($db, $sql);

$sql = "SELECT
            COUNT(`brojac_id`) AS brojac
        FROM `clanci_brojac_poseta`
        WHERE
            `datum_vreme` BETWEEN '{$pocetak}' AND '{$kraj}'";
$result = mysqli_query($db, $sql);
$row = mysqli_fetch_assoc($result);
$brojac_ukupno = (int)$row['brojac'];
if ($brojac_ukupno < 1) exit;
```

Listing 37 - *api-procentualno.php*, inicijalizacija

U sledećem koraku se ostvaruje veza sa bazom podataka, a karakter set i kolacija (engl. *collation*) za tu vezu se podešava na UTF-8 karakter set, linije 14 i 15.

U bloku linija 17-24 se očitava ukupan broj pregleda koji su ostvarili svi članci u zadatom periodu, koji će se u nastavku iskoristiti za izračunavanje procentualnog udela za filtrirane članke. Ukoliko je dobijena vrednost manja od 1, skripta se prekida zbog nedostatka podataka za vizuelizaciju, linija 25.

Glavni *SQL* upit, linije 27-43 na listingu 38, se sastoji iz dva upita. Zadatak ugnežđenog upita je da očita *id članka*, *naslov članka* i *brojač poseta* za 5 najposećenijih članaka u posmatranom periodu, dok roditeljski upit u dobijeni podskup dodaje procentualno učešće svakog članka u ukupnom broju poseta, linija 29. Dobijeni podaci se potom očitavaju i smeštaju u niz *$data*, linije 44-51. Na kraju se u niz dodaje i zbirna vrednost, odnosno učešće svih ostalih članaka u ostvarenom saobraćaju za izabrani period, linije 53-55.

```
27  $sql = "SELECT
28              c.*,
29              c.`brojac`/{$brojac_ukupno} AS `procenat`
30          FROM (
31              SELECT
32                  c.`clanak_id`,
33                  d.`clanak_naslov`,
34                  COUNT(c.`datum_vreme`) AS `brojac`
35              FROM `clanci_brojac_poseta` c
36              LEFT JOIN `clanci` d ON c.clanak_id=d.clanak_id
37              WHERE
38                  c.`datum_vreme` BETWEEN '{$pocetak}' AND '{$kraj}'
39              GROUP BY c.`clanak_id`
40              ORDER BY `brojac` DESC
41              LIMIT 5
42          ) c
43          ORDER BY procenat DESC";
44  $result = mysqli_query($db, $sql);
45  while ($row = mysqli_fetch_assoc($result)) {
46      $data['clanci'][] = $row['clanak_naslov'];
47      $data['brojac'][] = $row['brojac'];
48      $data['procenat'][] = round($row['procenat'] * 100, 1);
49      $procenat += round($row['procenat'] * 100, 1);
50      $brojac += $row['brojac'];
51  }
52
53  $data['clanci'][] = 'Ostali';
54  $data['brojac'][] = $brojac_ukupno - $brojac;
55  $data['procenat'][] = round(100 - $procenat, 1);
56
57  mysqli_close($db);
58
59  header('Content-Type: application/json');
60  echo json_encode($data);
61
62  ?>
```

Listing 38 - *api-procentualno.php*, očitavanje podataka i slanje JSON odgovora

Pita dijagram se generiše u dokumentu *procentualno.html*, listinzi 39-41. Osnovna struktura *HTML* stranice je prikazana na listingu 39. U odnosu na prethodni slučaj, ne koristi se *PHP* za prijem podataka za filtriranje na serverskoj strani, već taj zadatak obavlja *JavaScript* na klijentskoj strani. Osim same *Chart.js* biblioteke koja se ubacuje na stranicu u liniji 6, dodaje se i plugin za kontrolu natpisa (engl. *labels*) u poljima sa podacima, linija 7.

Jednostavan CSS kod se nalazi u linijama 8-13, sa ciljem da odredi širinu i visinu dijagrama koji će biti generisan.

```
1  <html lang="en">
2  <head>
3    <meta charset="UTF-8">
4    <meta name="viewport" content="width=device-width, initial-scale=1.0">
5    <title>Presek stanja za izabrani period</title>
6    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
7    <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-datalabels"></script>
8    <style>
9      #PieChartProcentualno {
10       width: 400px !important;
11       height: 400px !important;
12     }
13   </style>
14 </head>
15 <body>
16 <h2>Najčitaniji članci</h2>
17 <form method="get">
18   Početak
19   <input type="date" name="pocetak">
20   Kraj
21   <input type="date" name="kraj">
22   <button>Filtriraj</button>
23 </form>
24 <canvas id="PieChartProcentualno"></canvas>
```

Listing 39 - *procentualno.html*, kružni (pita) dijagram, *HTML* struktura stranice

Struktura veb obrasca za određivanje vremenskog raspona za koji se vrši vizuelizacija podataka je ista kao u prethodnom primeru i čine je tri elementa:

- *input* element tipa *date*, za unos podatka o početnom datumu
- *input* element tipa *date*, za unos podatka o krajnjem datumu
- *button* element za slanje unetih podataka

HTML element *canvas* u liniji 24 sa jedinstvenim identifikatorom *PieChartProcentualno* će biti iskorišćen za formiranje samog dijagrama.

Za inicijalizaciju i konfiguraciju dijagrama se koristi *JavaScript*, listinzi 40 i 41, a proces započinje pozivom funkcije *createPieChart* u liniji 103. U prvom koraku se očitavaju *URL parametri* sa podacima o početnom i krajnjem datumu, i na osnovu njih se formira *URL* za *API* poziv, linije 27-33. Input elementima se trenutne vrednosti podešavaju na izabrane datume, linije 34-35, i potom se asihrono preuzimaju podaci pozivom dokumenta *api-procentualno.php*, linije 37-39.

```
<script>
  async function fetchData() {
    const queryString = window.location.search;
    const urlParams = new URLSearchParams(queryString);
    const pocetak = urlParams.get('pocetak');
    const kraj = urlParams.get('kraj');
    const url = "api-procentualno.php?" +
                "pocetak=" + pocetak +
                "&kraj=" + kraj;
    document.getElementsByName('pocetak')[0].value = pocetak;
    document.getElementsByName('kraj')[0].value = kraj;

    const response = await fetch(url);
    const data = await response.json();
    return data;
  }
  async function createPieChart() {
    const data = await fetchData();
    const PieChartelement = document.getElementById('PieChartProcentualno');
    const ctx = PieChartelement.getContext('2d');
    const PieChartProcentualno = new Chart(ctx, {
      type: 'pie',
      data: {
        labels: data.clanci,
        datasets: [{
          label: 'Presek stanja za izabrani period',
          data: data.procenat,
          backgroundColor: [
            'rgb(255, 99, 132)',
            'rgb(54, 162, 235)',
            'rgb(255, 206, 86)'
          ],
          borderColor: [
            'rgba(0, 0, 0)',
            'rgba(0, 0, 0)',
            'rgba(0, 0, 0)'
          ],
          borderWidth: 2
        }]
      },
      options: {
        responsive: true,
        plugins: {
          legend: {
            display: false,
            position: 'top',
            labels: {
              font: {
                size: 40
              }
            }
          },
```

Listing 40 - *procentualno.html*, kružni dijagram, konfiguracija

```
77            tooltip: {
78              callbacks: {
79                label: function(tooltipItem) {
80                  return tooltipItem.label + ': ' + tooltipItem.raw;
81                }
82              }
83            },
84            datalabels: {
85              color: '#000000',
86              font: {
87                size: 30
88              },
89              formatter: (value, context) => {
90                const label = context.chart.data.labels[context.dataIndex];
91                const brojac = data.brojac[context.dataIndex];
92                return `${label}\n${value}%\n(${brojac})`;
93              },
94              anchor: 'end',
95              align: 'start',
96              offset: 10
97            }
98          }
99        },
100       plugins: [ChartDataLabels]
101     });
102   };
103   createPieChart();
104 </script>
105 </body>
106 </html>
```

Listing 41 - *procentualno.html*, kružni dijagram, konfiguracija natpisa u poljima

Pita dijagram se inicijalizuje i konfiguriše u linijama 45-102. *Chart.js* dozvoljava veliku slobodu u stilizovanju dijagrama, što omogućava prilagođavanje prikaza u skladu sa konkretnim potrebama projekta. U ovom slučaju su izvršene minimalne izmene, poput podešavanja debljine okvira elemenata dijagrama, linija 62 i veličine fonta, linije 73 i 87.

Chart.js generiše pita dijagram bez posebnih informacija u poljima dijagrama, već samo sa opcionom legendom koja se sastoji iz boje polja i imena kategorije. Ovakav prikaz često nije pregledan, a poseban problem može da predstavlja osobama koje pate od nekog oblika daltonizma.

Plugin datalabels koji nije izvorno uključen u biblioteku je konfigurisan u linijama 84-97, registruje se u liniji 100, i služi za prikaz natpisa o imenu polja, broju poseta i procentualnom udelu u samom polju, linije 89-93.

LITERATURA I INDEKSI

11. LITERATURA

1 Dobrojević, M. (2017). Importance of usage of ERP software and data mining techniques for crime prevention in fleet management. Bezbednost, Beograd, 59(3), 79-100.

2 Ritchie, H., Mathieu, E., Roser, M., & Ortiz-Ospina, E. (2023). Internet. Our World in Data. Dostupno na https://ourworldindata.org/internet, pristup 12.08.2024.

3 Usage statistics of server-side programming languages for websites. W3Techs. (n.d.). dostupno na https://w3techs.com/technologies/overview/programming_language, pristup 14.08.2024.

4 Usage statistics of Node.js, W3Techs, dostupno na https://w3techs.com/technologies/details/ws-nodejs, pristup 14.08.2024.

5 Beckman, J. (2024, May 30). Mobile vs desktop usage statistics for 2023. The Tech Report. https://techreport.com/statistics/hardware-gadgets/mobile-vs-desktop-usage-statistics

6 Gupta, C., Singh, R. K., & Mohapatra, A. K. (2022). GeneMiner: a classification approach for detection of XSS attacks on web services. Computational Intelligence and Neuroscience, 2022(1), 3675821.

7 Blobel, T., Rumo, M., & Lames, M. (2021). Sports information systems: a systematic review. International Journal of Computer Science in Sport, 20(1), 1-22.

8 Radojčić, V., Cvetković, A. S., Dobrojević, M., Spalević, P., & Guider, J. M. E. (2024). Advancements in Computer Vision Applications for Traffic Surveillance Systems.

9 Šarak, E., Dobrojević, M., & Sedmak, S. A. (2020). IoT based early warning system for torrential floods. FME Transactions, 48(3), 511-515.

10 Bacanin, N., Jovanovic, L., Stoean, R., Stoean, C., Zivkovic, M., Antonijevic, M., & Dobrojevic, M. (2024). Respiratory Condition Detection Using Audio Analysis and Convolutional Neural Networks Optimized by Modified Metaheuristics. Axioms, 13(5), 335.

11 Minic, A., Jovanovic, L., Bacanin, N., Stoean, C., Zivkovic, M., Spalevic, P., ... & Stoean, R. (2023). Applying recurrent neural networks for anomaly detection in electrocardiogram sensor data. Sensors, 23(24), 9878.

12 Radojcic V., Cvetkovic, AS., Dobrojević, M. (2023). The Use of Computer Vision in Precision Agriculture. In Sinteza 2023-International Scientific Conference on Information Technology, Computer Science, and Data Science (pp. 60-65). Singidunum University.

13 Todorović, J., Jakšić, B., Spalević, P., Dobrojević, M., & Milovanović, I. (2023). Performance analysis of WSN–FSO system modeled by Gamma–Chi-square channel distribution. Proceedings of the Estonian Academy of Sciences, 72(1), 30-40.

14 Reljić, V., Milenković, I., Dudić, S., Šulc, J., & Bajči, B. (2021). Augmented reality applications in industry 4.0 environment. Applied Sciences, 11(12), 5592.

15 M. Dobrojević, Magma fleet manager v2.0, Zbornik radova XIV Međunarodni naučni skup Sinergija 2013, pp. 263 - 268, Dec, 2013

16 Robert Simmon, “Use of Color in Data Visualization”, pristup 17.07.2024, dostupno na https://earthobservatory.nasa.gov/resources/blogs/intro_to_color_for_visualization.pdf

17 Zhang, Y., Zhang, Y., Zhao, X., Zhang, Z., & Chen, H. (2020). Design and data analysis of sports information acquisition system based on internet of medical things. IEEE Access, 8, 84792-84805.

18 Dobrojević, M., & Bačanin Džakula, N. (2021). Veb programiranje. Univerzitet Singidunum. ISBN 978-86-7912-752-5

Indeks slika

Slika 1 - Podsistemi mehaničkog sistema (bicikl) 17
Slika 2 - Komponente benzinskog motora u automobilu 19
Slika 3 - Lični računar, podsistemi 19
Slika 4 - Spremnost zaposlenih da prihvate uvođenje IS 25
Slika 5 - Broj korisnika Interneta 33
Slika 6 - Tržišni udeo veb servera, maj 2024. 34
Slika 7 - Slanje zahteva ka veb serveru i prijem odgovora 35
Slika 8 - Slojevi OSI 36
Slika 9 - HTTP zahtev i odgovor servera. 38
Slika 10 - Fizički slojevi veb aplikacije 42
Slika 11 - MVC arhitektura veb aplikacije 45
Slika 12 - Tržišni udeo programskih jezika na strani servera 46
Slika 13 - Redosled koraka u radu aplikacije sa bazom podataka 50
Slika 14 - Veb obrazac, pretraga sadržaja 53
Slika 15 - Serijalizovani podaci 58
Slika 16 - Korisnički interfejs, tolerancija na greške 73
Slika 17 - Gradivni elementi veb stranice 76
Slika 18 - Pomoćni navigacioni element, tzv. mrvice hleba 77
Slika 19 - Paginacija, najjednostavniji oblik 78
Slika 20 - Paginaciona lenta sa brojevima stranica 78
Slika 21 - Paginaciona lenta sa veb obrascem, phpBB 79
Slika 22 - Slajder za podešavanje raspona cene u filteru za pretragu 80
Slika 23 - Tagovi za filtriranje vesti sa vremenskom prognozom 80
Slika 24 - Obaveštenje o uspešno izvršenoj operaciji 82
Slika 25 - Upozorenje da operacija nije uspešno izvršena 82
Slika 26 - Veb obrazac za unos osnovnih ličnih podataka 83
Slika 27 - Upotreba padajućih lista u veb obrascu 84
Slika 28 - Upotreba prekidača u veb obrascu 85
Slika 29 - Input element tipa *date* 85
Slika 30 - Obeležavanje elemenata u veb obrascu, atribut *placeholder* 86
Slika 31 - Dodatni opis elementa za unos teksta 87
Slika 32 - Grupisanje elemenata u veb obrascu 87
Slika 33 - Modalni prozor za potvrdu prekida procesa registracije 89
Slika 34 - Praćenje napretka u okviru operacije iz više koraka 89

Slika 35 - Redosled koraka u procesu prijave korisnika 99
Slika 36 - Trend porasta broja XSS napada 110
Slika 37 - Komponente IS 116
Slika 38 - *Google Analytics*, izveštaj o saobraćaju na veb sajtu 120
Slika 39 - GIS, satelitski snimak dela teritorije 122
Slika 40 - GIS, katastar nepokretnosti, izbor opštine i katastarske opštine 122
Slika 41 - GIS, katastar nepokretnosti, unos broja katastarske parcele 123
Slika 42 - GIS, katastar nepokretnosti 123
Slika 43 - Oblasti primene AR u industriji [] 133
Slika 44 - Agregacija podataka u aplikaciji *MS Excel* 142
Slika 45 - MS Teams, izveštaj sa sastanka, CSV format 143
Slika 46 - Izveštaj sa sastanka, JSON format 143
Slika 47 - Izveštaj sa sastanka, tekst format 144
Slika 48 - Izvorni podaci u Excel tabeli (levo); Deskriptivna i pregledna analiza (desno) 146
Slika 49 - Dokumenti kvaliteta, obrazac radne liste 157
Slika 50 - Deo organizacione šeme kompanije 159
Slika 51 - Tokogram aktivnosti, 1. deo 161
Slika 52 - Tokogram aktivnosti, 2. deo 162
Slika 53 - Tokogram aktivnosti, 3. deo 163
Slika 54 - Tokogram aktivnosti, 4. deo 164
Slika 55 - Tokogram aktivnosti, 5. deo 165
Slika 56 - Struktura tabela za podršku sistemu za distribuciju ovlašćenja 167
Slika 57 - Sadržaj tabela za podršku sistemu za distribuciju ovlašćenja 168
Slika 58 - Očitavanje podataka o korisniku i njegovoj ulozi u IS, rezultat SQL upita 169
Slika 59 - Rezultat provere ovlašćenja za korisnika 170
Slika 60 - Veb obrazac za pretragu baze korisnika 177
Slika 61 - Pretraga putničkih vozila filtriranih po datumu isteka registracije 177
Slika 62 - Nabavka goriva, rezultat pretrage za izabrano vozilo 179
Slika 63 - Nabavka goriva, grupisanje podataka prema tipu goriva 180
Slika 64 - Upotreba boja u izveštaju 181
Slika 65 - Operativni izveštaj prodaje 182
Slika 66 - Vizuelizacija podataka o prisustvu studenata na predavanjima 184
Slika 67 - *Stubasti dijagram*, izveštaj o maloprodaji, *Magento* 185
Slika 68 - *Pita dijagram*, prikaz trenutnog preseka stanja 185
Slika 69 - Uporedni grafički prikaz podataka za vozilo 187
Slika 70 - Stubasti dijagram (*bar chart*), *Chart.js* 188
Slika 71 - Integracija veb prodavnice sa stanjem zaliha 195
Slika 72 - Izveštaj o pretplatama, WooCommerce 198
Slika 73 - Vizuelizacija GPS log-a 203
Slika 74 - Struktura tabele za skladištenje članaka `*clanci*` 210
Slika 75 - Struktura tabele za skladištenje podataka o poseti `*clanci_brojac_poseta*` 210
Slika 76 - Generisani test podaci, SQL upit 213

Slika 77 - Generisani test podaci, tabela *clanci_brojac_poseta*, *phpMyAdmin* 213
Slika 78 - Generisani test podaci, tabela *clanci*, *phpMyAdmin* 215
Slika 79 - Brojač poseta po članku, rezultat upita, *phpMyAdmin* 216
Slika 80 - Rezultati upita prikazani na veb stranici u tabelarnom formatu 218
Slika 81 - Rezultati izmenjenog upita 219
Slika 82 - Upravljanje sortiranjem podataka preko *dropdown* elementa 220
Slika 83 - Grafički prikaz podataka u formi stubastog dijagrama (*barchart*) 223
Slika 84 - Veb obrazac za izbor vremenskog perioda za analizu 226
Slika 85 - Mesečni broj pregleda za izabrani članak 226
Slika 86 - Najčitanijih 5 članaka u izabranom periodu 231

Indeks listinga

Listing 1 - Veb obrazac, HTML kod 53
Listing 2 - Apache .htaccess preusmeravanje dolaznog saobraćaja na *index.php* 56
Listing 3 - PHP manipulacija zaglavljem, slanje pdf dokumenta 57
Listing 4 - PHP slanje zaglavlja sa statusnim kodom 404 57
Listing 5 - Podaci u izvornom obliku, PHP niz 58
Listing 6 - PHP reuzimanje sadržaja lokalnog fajla ili sa URL adrese 61
Listing 7 - FTP preuzimanje fajla 62
Listing 8 - Maskiranje input elementa tipa date radi upotrebe atributa *placeholder* 86
Listing 9 - Podešavanje parametara sesije, PHP 100
Listing 10 - PHP priprema korisničkog imena za ubacivanje u MySQL upit 102
Listing 11 - Upit za selektovanje korisničkog naloga, PHP/MySQL 106
Listing 12 - SQL upit sa injekcijom 106
Listing 13 - Upotreba pripremljenih upita, PHP/MySQL 107
Listing 14 - MySQL upit za selektovanje korisničkog naloga 108
Listing 15 - Očitavanje rezultata SQL upita, injekcija ima efekta 108
Listing 16 - Očitavanje rezultata SQL upita, injekcija nema efekta 108
Listing 17 - Očitavanje podataka o korisniku i njegovoj ulozi u IS, SQL upit 169
Listing 18 - Provera ovlašćenja korisnika za izabranu akciju 169
Listing 19 - Grupisanje podataka po gradu, SQL upit 178
Listing 20 - Integracija *Chart.js* stubastog dijagrama u HTML stranicu 189
Listing 21 - Kreiranje tabela za članke i brojač poseta, SQL upit 210
Listing 22 - Generisanje test podataka o posetama za tabelu *clanci_brojac_poseta* 212
Listing 23 - Generisanje test podataka o člancima za tabelu *clanci* 214
Listing 24 - Brojač poseta po članku, SQL upit 216
Listing 25 - Očitavanje podataka iz baze i priprema za klijentsku stranu, model 217
Listing 26 - Listing članaka sa brojačem poseta, SQL upit 218
Listing 27 - Adaptacija pogleda za prikaz naslova članka i datuma objave 219
Listing 28 - Sortiranje, dodavanje veb obrasca 220
Listing 29 - Sortiranje, očitavanje *URL* parametra i formiranje *SQL* upita 221
Listing 30 - *api.php*, broj pregleda po članku 222
Listing 31 - *brojac-clanci.html*, stubasti dijagram (*barchart*), *Chart.js* 224
Listing 32 - API, ograničenje rezultata na 5 članaka 225
Listing 33 - *api-clanak.php*, mesečni broj pregleda za izabrani članak 227
Listing 34 - *brojac-clanak.php*, očitavanje URL parametara 228

Listing 35 - *brojac-clanak.php*, veb obrazac za filtriranje podataka 229
Listing 36 - Očitavanje URL parametara, *JavaScript* 230
Listing 37 - *api-procentualno.php*, inicijalizacija 232
Listing 38 - *api-procentualno.php*, očitavanje podataka i slanje JSON odgovora 233
Listing 39 - *procentualno.html*, kružni (pita) dijagram, *HTML* struktura stranice 234
Listing 40 - *procentualno.html*, kružni dijagram, konfiguracija.....235
Listing 41 - *procentualno.html*, kružni dijagram, konfiguracija natpisa u poljima 236

Indeks tabela

Tabela 1 - HTTP odgovor servera, najčešći statusni kodovi 39
Tabela 2 - Kompanije koje koriste PHP programski jezik 47
Tabela 3 - PHP funkcije za rad sa kukijima 52
Tabela 4 - Upotreba URL parametara za rutiranje 55
Tabela 5 - Upotreba čistih URL-a za rutiranje 55
Tabela 6 - PHP funkcije za rad sa zaglavljima 56
Tabela 7 - PHP funkcije za rad sa fajlovima i direktorijumima 59
Tabela 8 - PHP funkcije za rad sa fajlovima, čitanje i upis podataka 60
Tabela 9 - PHP funkcije za rad sa FTP serverom 61
Tabela 10 - PHP funkcije koje se koriste za kontrolu sesije 100
Tabela 11 - PHP funkcije primenjive u sanitaciji i validaciji podataka 101
Tabela 12 - Komercijalni alati za vizuelizaciju podataka 190
Tabela 13 - Besplatni JavaScript alati za vizuelizaciju podataka 190
Tabela 14 - Veb bazirana rešenja za upravljanje veb prodavnicama 196
Tabela 15 - Veb bazirani sportski IS 202

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

007:004(075.8)
004.42:004.738.5(075.8)

ДОБРОЈЕВИЋ, Милош, 1975-
Veb informacioni sistemi / Miloš Dobrojević. - 1. izd. - Beograd : Univerzitet Singidunum, 2024 (Beograd : Caligraph). - 251 str. : ilustr. ; 25 cm

Tiraž 600. - Bibliografija: str. 241-243. - Registri.

ISBN 978-86-7912-827-0

a) Информациони системи b) Интернет -- Програмирање

COBISS.SR-ID 153200137

© 2024.

Sva prava zadržana. Nijedan deo ove publikacije ne može biti reprodukovan u bilo kom vidu i putem bilo kog medija, u delovima ili celini bez prethodne pismene saglasnosti izdavača.

www.singidunum.ac.rs

9 788679 128270



VEB INFORMACIONI SISTEMI

Miloš Dobrojević

Internet i veb tehnologije su nezaobilazni deo svakodnevnog, poslovnog i privatnog života. Veb aplikacije su pristupačne i mogu da se koriste preko različitih uređaja, omogućavaju saradnju u realnom vremenu, skalabilne su i podrazumevaju niže troškove implementacije i održavanja u odnosu na desktop aplikacije, što ih čini atraktivnom opcijom za mnoge organizacije. Informacioni sistemi, kao alat za unapređenje poslovanja i lične produktivnosti, su takođe obuhvaćeni ovim trendom i sve češće se razvijaju u formi veb aplikacija. Ova knjiga je namenjena studentima koji pohađaju predmet Veb informacioni sistemi na trećoj godini studija Tehničkog fakulteta Univerziteta Singidunum, u okviru studijskog programa Softversko i informaciono inženjerstvo. Očekuje se da studenti imaju prethodno znanje iz predmeta Veb dizajn, Veb programiranje i Baze podataka, odnosno da su upoznati sa osnovnim veb tehnologijama kao što su HTML, CSS, PHP i SQL. U okviru ovog predmeta, studenti će se upoznati sa osnovnim konceptima razvoja informacionih sistema kao veb aplikacija.

Univerzitet Singidunum