

**UNIVERZITET SINGIDUNUM
FAKULTET ZA POSLOVNU INFORMATIKU**

Darko Dragić

**PROJEKAT WEB APLIKACIJE
-INTERNET ONLINE OGLASI-**

- Diplomski rad -

Beograd, 2007.



FAKULTET ZA POSLOVNU INFORMATIKU

PROJEKAT WEB APLIKACIJE -INTERNET ONLINE OGLASI-

- Diplomski rad -

Mentor:
Prof dr Mladen Veinović

Student:
Darko Dragić

Br. indeksa:
2s/2003



FAKULTET ZA POSLOVNU INFORMATIKU

UNIVERZITET SINGIDUNUM
FAKULTET ZA POSLOVNU INFORMATIKU
Beograd, Danijelova 32

Broj: _____/2007

Kandidat: Darko Dragić

Broj indeksa: 2s/2003

Smer: Projektovanje i programiranje

Tema: PROJEKAT WEB APLIKACIJE - INTERNET ONLINE OGLASI-

Zadatak: Predstaviti i upoznati se sa osnovnim tehnikama u realizaciji web aplikacije - internet online oglasi. U ovom radu predstavljeni su HTML , PHP kao i sistem za upravljanje relacionim bazama podataka MySQL za skladištenje podataka. Omogućiti unos i pregled oglasa, registraciju korisnika, brisanje i izmenu istih, administratorska prava kao i ostale propratne funkcije za uspešan rad aplikacije .Realizovati i opisati web aplikaciju za internet online oglase. Unos oglasa u bazu realizovati korišćenjem standardnih HTML formi a za rešenje zadatka primeniti programski jezik PHP i MySQL za skladištenje podataka.

MENTOR

Prof. dr Mladen Veinović

Datum odobrenja teme:

03.06.2007.

Beograd

DEKAN

Prof. dr Milan Milosavljević

SAŽETAK

U ovom radu prezentirana je ideja o razvijanju novog, jednostavnog, preglednog i pre svega funkcionalnog web site-a, dizajniranog tako da bude prilagođen svim strukturama korisnika Internet usluga, odn. funkcionalno prilagođenog svim korisnicima Internet usluga bez obzira na nivo znanja korištenja Internet-a.

ABSTRACT

This paper presents an idea on developing a new, simple, straightforward and, above all, functional web site, designed so as to suit all profiles of Internet service users, i.e. functionally adapted to all users of Internet facilities, regardless of their Internet competence levels .

1	UVOD.....	- 1 -
2	HTML.....	- 2 -
2.1	TAGOV.....	- 2 -
2.2	HTML FORME	- 3 -
2.3	PRVI KORAK.....	- 4 -
2.3.1	TEKST FORMA (TEXT).....	- 4 -
2.3.2	TEKSTUALNO POLJE (TEXT AREA)	- 5 -
2.3.3	POLJE ZA GLASANJE (RADIO BUTTON).....	- 6 -
2.3.4	POLJE ZA OBELEŽAVANJE, ČEKIRANJE (CHECKBOX).....	- 6 -
2.3.5	PADAJUĆI MENI (POP-UP BOX).....	- 7 -
2.3.6	POŠALJI I BRIŠI DUGMAD (Submit and reset buttons).....	- 8 -
2.3.7	PRIMER JEDNE HTML FORM	- 9 -
3	CSS.....	- 11 -
3.1	ŠTA JE CSS	- 11 -
3.2	CSS SYNTAX-e.....	- 12 -
3.2.1	SYNTAX-e ZA TEKST	- 12 -
3.2.2	CSS MARGINE	- 13 -
3.2.3	CSS BOJE - COLORS	- 14 -
3.2.4	BACKGROUND - POZADINA	- 14 -
3.2.5	BORDER - OKVIR.....	- 15 -
3.2.6	UPOTREBA CSS-a.....	- 17 -
3.2.7	UPOTREBA CSS-a U HTML-u	- 17 -
4	PHP.....	- 18 -
4.1	UVOD U PHP	- 18 -
4.2	PHP U POREĐENJU SA DRUGIM JEZICIMA	- 19 -
4.3	RAZLIKA IZMEĐU S-S i C-S SKRIPTNIH JEZIKA	- 20 -
4.4	SINTAKSA PHP-a.....	- 21 -
4.5	PRELAŽENJE IZ PHP-a U HTML MOD	- 21 -
4.6	POVEZIVANJE HTML OBRAZCA SA PHP SKRIPTOM	- 22 -
4.7	NEKE OD OSNOVNIH PHP TEHNIKA.....	- 23 -
4.7.1	IF iskaz:	- 23 -
4.7.2	WHILE petlja:	- 24 -
4.7.3	FOR petlja:	- 25 -
4.8	UPOTREBA KOLAČIĆA (cookies)	- 27 -
5	BAZE PODATAKA.....	- 29 -
5.1	UPOTREBA MYSQL BAZA PODATAKA	- 30 -
6	PROJEKAT	- 31 -
6.1	INDEX.....	- 31 -
6.2	MENI.....	- 32 -
7	RAZVOJ.....	- 33 -
7.1	SPISAK MySQL TABELA	- 33 -
8	FUNKCIJE	- 35 -
8.1	INSTALACIJA.....	- 36 -
8.2	REGISTRACIJA NOVOG KORISNIKA.....	- 42 -
8.3	LOGIN.....	- 45 -
8.4	PREGLED OGLASA.....	- 47 -
8.5	PREDAJA OGLASA	- 49 -
8.6	PRETRAGA OGLASA.....	- 52 -
8.7	PODEŠAVANJA	- 53 -
8.8	BRISANJE OGLASA	- 54 -

8.9	ADMINISTRACIJA – CMS	- 56 -
8.9.1	USERS.....	- 56 -
8.9.2	GRADOVI I KATEGORIJE.....	- 57 -
8.9.3	VESTI – NEWS	- 58 -
8.10	CMS – LISTING KODA.....	- 59 -
9	ZAKLJUČAK.....	- 66 -
10	LITERATURA	- 67 -

1 UVOD

Upoznavajući se sa statistikama određenih web pretraživača uvideo sam da je reč „oglasi“ jedna od vodećih po broju izvršenih pretraga. Oslonivši se na ovaj podatak, odlučio sam da razvijem jednu web aplikaciju koja bi omogućavala da se na brz i jednostavan način omogući internet korisnicima da uspešno vrše razmenu oglasa putem interneta.

Prva zamisao jeste da upotreba ove aplikacije bude prilagođena svim nivoima internet korisnika , jer aplikacija nije usmerena usko jednoj grupi korisnika nego očekujem širu grupu posetilaca. Zbog ovoga kao prvo web aplikacija (u daljem tekstu web stranica) treba da bude pojednostavljena do krajnjih granica ali da u istom trenutku ostane krajnje funkcionalna. Takođe da bi zadovoljio gore navedene uslove i sam dizajn web stranice treba da bude jednostavan kako zbog lakse preglednosti tako i zbog bržeg učitavanja (s obzirom da je u Srbiji i dalje jedna od najzastupljenijih internet konekcija – dial-up).

Posmatrajući neke od već postojećih aplikacija za internet online oglase uvideo sam određene nedostatke koje bi trebalo da ispravim u ovom projektu:

- Određeni deo sadržaja zaštititi autorizacijom (dozvoliti pristup samo registrovanim korisnicima) kao npr. predaja oglasa , pretraga oglasa ...
- Oglase predavati i pregledavati po kategorijama i po gradovima
- Web sajt treba da ima određenu vrstu kontrole kako sadržaja tako i ostalih funkcija sajta kao npr. dodavanje/brisanje kategorija , gradova , oglasa ... uz pomoć CMS-a („Content Management System“)
- Svakom registrovanom korisniku omogućiti da u istom trenutku može imati ukupno 5 oglasa s tim da brisanjem jednog od već postavljenih oslobađa prostor za postavljanje novog oglasa (sprečavanje nagomilavanja nepotrebnih i zastarelih oglasa)

Dakle ideja jeste napraviti jednostavan , pregledan i pre svega funkcionalan web sajt za online oglase prilagođen svim korisnicima interneta bez obzira na njihov nivo znanja korištenja interneta.

2 HTML

Velika raznovrsnost korišćenih tipova računara, operativnih sistema i softvera na internetu prouzrokovala je potrebu za markap jezicima. Markap jezici predstavljaju jezike koji samo opisno prikazuju svoj sadržaj, dok se pravi prikaz sadržaja vrši na korisničkom računaru, gde se nalazi interpreter za dati jezik. Na ovaj način prevaziđeno je pisanje gotovih prikaza za svaki tip računara, operativnog sistema ili korisničkog softvera. Sve www stranice su pisane markap jezicima. Interpreteri na korisničkim računarima se nazivaju brauzeri ili korisnički agenti.

HTML jezik zadužen je za izgled stranice, i on predstavlja “lice” koje vide korisnici koji pristupaju datom web sajtu, i zato je vrlo bitno poznavati HTML jezik prilikom dizajniranja web stranica. HTML jezik se sastoji od komandi koji su osnovni elementi jezika, a najčešće se nazivaju **tag**-ovi. Pomoću tagova se sve obavlja, formatira se tekst kao naslov, paragraf ili tabela. Postoje tagovi za uključenje mnogih drugih formata fajlova, kao što su slike filmovi. Takođe postoje tagovi za uključenje hiperlinkova. Hiperlinkovi su najvažniji deo stakog web sajta jer oni povezuju tekuću web stranicu sa drugim stranicama i tako obezbeđuju korisniku da se kreće kroz web. Hiperlinkovi poseduju URL adresu željene stranice koju brauzer čita i preko nje dolazi do željene stranice.

Dobra osobina HTML jezika je da ima mogućnost proširivanja, odnosno, mogu mu se dodavati nove komande, a da to ne poremeti izvršenje ranje napisanih dokumenata u HTML formatu.

Koncept markup jezika vezan je za interpretiranje napisanog teksta. Svaki brauzer na prevodi celokupan tekst napisan u nekom markup jeziku, npr. HTML jeziku, već interpretira, tumači, liniju po liniju i izvršava komande koje je protumačio iz teksta. U HTML je pravac izvršavanja instrukcija strogo odozgo na dole.

2.1 TAGOVI

Tagovi odnosno komande u HTML jeziku imaju specifičan način pisanja. Svi tagovi se pišu između znakova manje i veće <...>. Kao primer <H1> tag označava početak naslova prvog nivoa, dok tag </H1> označava kraj naslova prvog nivoa. Sledeća linija u HTML programu prikaće jedan naslov u kome piše “Ovo je naslov”.

```
<H1>Ovo je naslov</H1>
```

Primećujemo da postoje tag na početku koji označava startni tag i tag na kraju koji označava krajnji tag. Sam tekst naslova ograničen je početnim i krajnjim tagom. Postoje tagovi koji nemaju krajnji tag, kao što je
 koji vrši prekid linije teksta

```
<H1>Ovo je <br> nalsov</H1>
```

Neosetljivost na velika i mala slova. Svi tagovi su neosetljivi na pisanje malih slova tako da je tag `<strong>` isto što i `<StROng>`. Preporuka je da dizajner web stranice piše tagove na onaj način na koji mu to odgovara najviše, a najčešće se pišu velikim slovima jer su tada najčitljiviji.

Atributi Tagova. Ponekad tagovi poseduju neke attribute ili karakteristike koji preciznije opisuju način prikaza teksta, slike, tabele. Atributi su kao neke promenljive koje mogu dobiti jednu od konačnog broja unapred poznatih vrednosti. Na primer sledeći tag

```
<H1 ALIGN="center"> Ovo je nalsov </H1>
```

H1 poseduje atribut ALIGN koji dobija vrednost "center". Na isti način tag `<IMG>` poseduje atribut SRC koji određuje ime fajla gde se nalazi slika.

```
<IMG SRC="imeslike.gif">
```

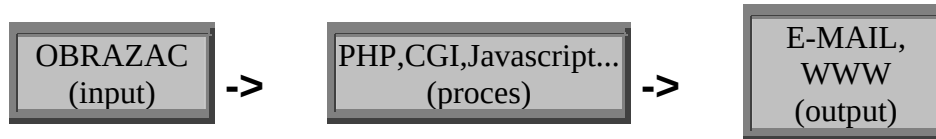
Atributi su takođe neosetljivi na velika i mala slova.

2.2 HTML FORME

Možda najvažniji deo HTML jezika u dizajniranju web sajtova za online oglase jesu obrazci ondosno **forme**. Krstareći internetom često nailazimo na web stranice gde treba da upišemo svoju E-mail adresu, ime, prezime ili neki drugi podatak. Ovo je omogućeno uz pomoć HTML formi.

Web stranice za online oglase takođe sadrže ove forme ili više formi za unos teksta. Ako je potrebno da se logujemo na sajt radi autorizacije ili ako želimo da se registrujemo na nekom sajtu sigurno ćemo morati unositi neke podatke kao što su username ili password ...

U načelu može se reći da je HTML obrazac web stranica ili njezin dio koji sadrži polja za unos podataka. Uneseni podaci se dalje prosljeđuju nekakvoj skripti (PHP, JavaScript, ASP, CGI ...) koja podatke obrađuje te na osnovu toga generira neki izlaz (slanje na e-mail adresu, stvaranje nove web stranice ...). Prema tome naš obrazac predstavlja input podataka, skripta je proces obrade tih podataka, a output je nekakav finalni proizvod samog procesa. Takva struktura prikazana je na slici2.



Slika1: prikaz odnosa HTML form – > neka skripta

NAPOMENA Forme koje ćemo ovde naučiti da pravimo su osnovne. Postoji još puno komandi koje možemo prikazati formama.

2.3 PRVI KORAK

Prvo moramo reći browseru da započinjemo formu, i šta želimo da se desi sa podacima upisanim u formi. Komanda kojom govorimo browseru da treba da prikaže formu izgleda ovako:

```
<FORM METHOD="POST" ACTION="mailto:dragic@gmail.com" ENCTYPE="text/plain">
```

Primećujemo da ova komanda radi četiri stvari:

1. Obaveštava browser da počinje FORMA.
2. Obaveštava browser da METODA kojom će forma tretirati unete podatke biti slanje e-mailom (POST=Pošta).
3. Da podaci uneti u formu treba da budu poslani na upisanu e-mail adresu pomoću ACTION="mailto:dragic@gmail.com" komande.
4. Da unete podatke treba da se pošalju kao tekst. Ovo radi ENCTYPE="text/plain" komanda.

Napomena. Treba da stavimo svoju e-mail adresu odmah posle "mailto:" bez praznog polja! Na adresu koju ovde napišemo, biće poslani podaci iz forme.

Sada kada browser zna da počinje forma treba da napišemo kakav tip forme želimo da se prikaže. Ovde ću prikazati pet tipova; TEXT (tekst), TEXT AREA (tekstualno polje), RADIO BUTTON (polje za glasanje), CHECKBOX (polje za obeležavanje, čekiranje), i POP-UP BOX (Padajući meni). Ove forme se najviše koriste na internetu. Takođe, pokazaću kako da napravimo dugmiće koji će kada se na njih klikne poslati podatke iz forme na "mailto:" adresu ili obrisati sadržaj forme.

2.3.1 TEKST FORMA (TEXT)

Ovo je osnovna forma koja omogućava upisivanje jednog reda teksta. Kada je postavimo na stranici, surferi će moći da upisu podatke kao što su ime i prezime, e-mail adresu i sl. Evo kako tekst forma (box) izgleda:

Slika2 text forma

Treba da kliknete mišem na formu da bi je aktivirali. Ako nikada do sada niste pravili forme možda ćete pomisliti da je to što vidite .gif ili .jpg. E, nije. Forma je ubačena pomoću jedne HTML komande koja izgleda ovako:

```
<INPUT TYPE="text" NAME="naziv" SIZE="30">
```

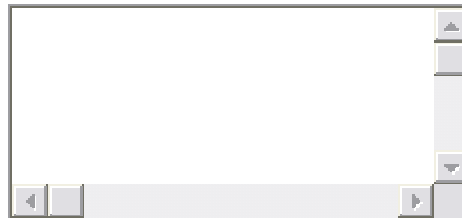
Ova komanda je uradila tri stvari

- **INPUT TYPE** govori browseru da će tu biti ubačen element (neki tip) forme. Gore smo rekli browseru da treba da prikaze formu a sada mu govorimo koji tip forme treba da prikaže. Tip ove forme je "text".
- **NAME** je naziv koji dodeljujemo box-u. Treba da znamo da ne moramo box nazvati "naziv". Nazovimo ga kako god hoćemo i sadržaj će nam stići pod tim imenom. Ako box služi za upisivanje imena surfera, nazovimo ga "ime". Ako box služi za upisivanje e-mail adrese surfera, nazovimo ga "e-mail". itd., itd.
- **SIZE** označava koliko će karektera (polja) box imati. Možeo upisati 10, 60, a možemo i 100 ako želimo.

2.3.2 TEKSTUALNO POLJE (TEXT AREA)

Ovaj box takođe služi za upisivanje teksta, samo što je veći i za razliku od prethodnog možemo upistati više redova teksta.

Evo kako ovaj box izgleda:



Slika3 : text area

Ako želimo, možemo upisati koliko god redova hoćemo. Samo kliknemo mišem da aktiviramo formu.

Evo komande koja je napravila ovakav box:

```
<TEXTAREA NAME="komentar" ROWS=6 COLS=40></TEXTAREA>
```

Primećujemo da TEXTAREA zahteva završnu </TEXTAREA> komandu, dok prethodni primer na zahteva.

Evo opisa delova komande:

- **TEXTAREA** (jedna reč) govori browseru da treba da prikaže tekstualno polje kakvo vidimo gore.
- **NAME** ima istu funkciju kao i prethodni primer. Poruka koju surfer upise u polje će stići na naš e-mail označena bilo kojim imenom koji ovde ubacimo. U ovom slučaju, ono što je napisano u box-u će nam stići na e-mail sa rečima, "komentar=Tekst koji je surfer upisao u formu".

- **ROWS** govori browseru koliko će redova box imati.
- **COLS** govori browseru koliko će karaktera (polja) box imati. Gornji primer ima 6 redova i 40 polja.

2.3.3 POLJE ZA GLASANJE (RADIO BUTTON)

Ovo je maleno okruglo polje. Kružić je aktivan i surfer može mišem kliknuti na njega. Kada neko klikne unutrašnjost kružića se ispuni crnom tačkom. Evo tri ovakva polja:



Probajmo da obeležimo dva kružića i... naravno nećemo uspeti zato što ova polja tako funkcionišu, kada kliknemo na jedno pa zatim na drugo polje, ovo prvo će biti deselktovano.

Ali zašto se zovu **radio** buttons (buttons=dugmici)?! To je zato što se ponašaju kao dugmići na starim radio uređajima za kola. Kada uključimo nešto dugme se uvuče. Kada pritisnete drugo dugme, ovo prvo se isključi i vrati u prethodno stanje. Evo komande za ubacivanje ovakvog polja:

```
<INPUT TYPE="radio" NAME="Grupa_kruzica" VALUE="Za_koga_se_glasa">
```

Pogledajte, tip forme u TYPE komandi je "radio". Ova komanda je malo dugačka, ali nije nimalo komplikovana. Evo njenih tri dela sa objašnjenjima:

- **INPUT TYPE** isto kao u gornjim primerima označava tip forme koja će se prikazati. U ovom slučaju to je "radio"
- **NAME** ovde treba da označimo kojoj grupi kružića ovo polje pripada, npr. ako na sajtu imamo dve ankete, na jednoj recimo pitamo koja je naša najlepša pevečica i u toj anketi stoji nekoliko predloga, a na drugoj, ko je naš najbolji političar, i tu imamo nekoliko predloga. Za prvu anketu, mogli bi da stavimo NAME="Naj pevacica je", a za drugu NAME="Naj politicar je".
- **VALUE** ovde treba da u našem slučaju za prvu anketu napišemo imena i prezimena predloženih pevačica, a za drugu imena i prezimena nekih političara kako bi smo lakše protumačili za koga je surfer glasao.

2.3.4 POLJE ZA OBELEŽAVANJE, ČEKIRANJE (CHECKBOX)

Polje za obeležavanje je klon polja za glasanje, a razlikuje se u dve stvari:

- Kada ubacimo kod na stranici se prikazuje mali kvadrat, a kada kliknete, u kvadraticu se pojavljuje znak za čekiranje.
- Možemo obeležiti (čekirati) koliko god kvadratica želite.

Evo nekoliko ovakvih polja:



Evo komande koja ubacuje ovaj checkbox na našu stranicu:

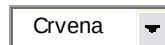
```
<INPUT TYPE="checkbox" NAME="Polje_za_cekiranje" VALUE="Neki tekst">
```

Svaka komanda u ovom kodu je ista kao i u prethodnom primeru pa nema potrebe da ih ponovo opisujemo. Jedino je novo TYPE="checkbox" umesto TYPE="radio" iz prethodnog primera.

Zapamtimo da nam iz ovakvih formi na e-mail mogu stići više obeleženih kvadratića. Ovakve forme se najčešće koriste kada želimo da surferima ponudimo izbor, recimo za primanje e-maila samo iz oblasti koje ih interesuju, i koje će izabrati pomoću ovih simpatičnih kvadratića. Za ankete i glasanja je ipak mnogo bolje koristiti radio polja.

2.3.5 PADAJUĆI MENI (POP-UP BOX)

Ova stvar, nam može uštedeti puno prostora na stranici. Evo jednog padajućeg menija, a da biste videli sve opcije morate da kliknuti na njega. Pomoću ovog padajućeg menija možemo pitati korisnike za omiljenu boju:



Evo koda:

```
<SELECT NAME="Omiljena_boja" SIZE="1">  
<OPTION SELECTED>Braon  
<OPTION>Crvena  
<OPTION>Plava  
<OPTION>Zelena  
<OPTION>Crna  
<OPTION>Braon  
</SELECT>
```

Iako ovo možda izgleda malo komplikovano, zaista nije. Komande se u stvari stalno ponavljaju a neke smo već upoznali. Evo delova koda uz objašnjenje:

- **SELECT** govori browseru da treba da prikaze formu . Ovoga puta to je SELECT (select=izaberi) tj. padajući meni.
- **NAME** isto kao u gornjim primerima označava kako će izgledati tekst koji će stići na naš e-mail. A izgledaće ovako: "Omiljena_boja=" a zatim, izbor surfera.
- **SIZE** označava veličinu box-a. U gornjem primeru je 1, što znači da će samo jedan red teksta biti vidljiv.
- **OPTION SELECTED** označava koja će opcija biti vidljiva. U ovom slučaju to je "Crvena" boja.
- **OPTION** označava druge izbore koji će biti vidljivi kada surfer klikne na padajući meni
- **/SELECT** govori browseru da se forma završava.

2.3.6 POŠALJI I BRIŠI DUGMAD (Submit and reset buttons)

Sada kada smo ubacili sve elemente formi na stranicu, treba da pronađemo način da omogućimo surferu da nam pošalje upisane podatke na e-mail adresu. Ovi dugmići su najlakši za pravljenje od svih elemenata forme. Evo kako izgledaju:



A, evo komandi kojima se dugmići ubacuju na stranicu:

```
<INPUT TYPE="submit" VALUE="Posalji"><INPUT TYPE="reset" VALUE="Brisi">
```

Sve komande su poznate, samo ću nešto razjasniti:

Submit znači izvršiti. Kada surfer klikne "izvršiće" se slanje sadržaja forme na upisanu e-mail adresu.

Reset znači dovesti u prvobitno stanje tj. kada surfer klikne na ovo dugme, sve što je upisao u gore prikazane elemente formi će biti izbrisano.

VALUE u ovom slučaju označava šta će biti napisano na dugmetu. Ne moramo staviti "Posalji" i "Brisi", možemo i nešto drugo.

Konačno!

Nemojmo zaboraviti da sve što smo ispisali zatvorimo sa:

```
</FORM>
```

2.3.7 PRIMER JEDNE HTML FORM

Sledi jedan jednostavan primer (Listing1) HTML forme koja zahteva od korisika da unese Ime ,Prezime , Grad , izabere jednu tri ponuđene države , username , password.

```
<html>

<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="NotePad">
<meta name=" " content=" ">
<title>Registracija</title>
</head>

<body>

<form method="POST" action="registar.php">

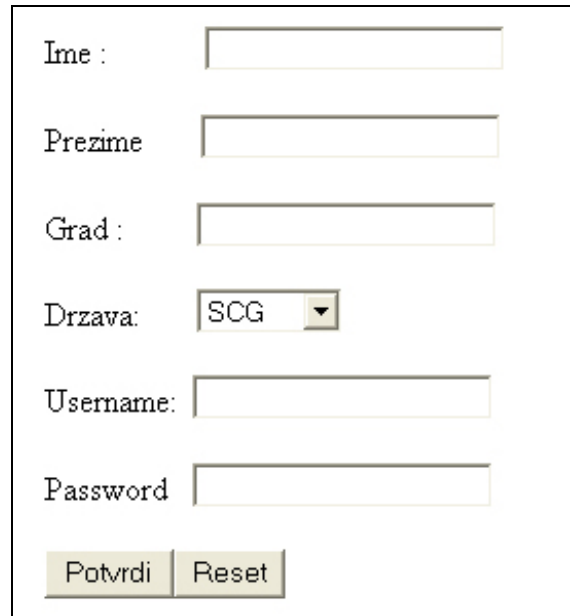
  <p>Ime :<input type="text" name="ime" size="20"></p>
  <p>Prezime <input type="text" name="prezime" size="20"></p>
  <p>Grad :<input type="text" name="grad" size="20"></p>
  <p>Drzava:<select size="1" name="država">
    <option selected>SRB</option>
    <option>Austrija</option>
    <option>Italija</option>
  </select></p>
  <p>Username:<input type="text" name="username" size="20"></p>
  <p>Password <input type="text" name="password" size="20"></p>
  <p><input type="submit" value="Potvrđi" name="B1"><input type="reset"
value="Reset" name="B2"></p>
</form>

</body>

</html>
```

Listing1

Kada gore navedeni kod obradi web browser npr. FireFox ... dobićemo sledeći izlaz. Slika3.



Ime :

Prezime

Grad :

Drzava:

Username:

Password

Potvrdi Reset

Slika3 izgled forme

Iz listinga1 se može zaključiti da navedeni kod predstavlja samo prazan obrazac za unos podatak od strane posetioca – korisnika. Da bi se ovakav obrazac stavio u funkciju povezaćemo ga sa nekom PHP skriptom.

U HTML formi moramo navesti putanju i ime PHP skripte koju ćemo povezati sa našim HTML obrazcem. U listingu1 može se videti sledeća linija koda :

```
<form method="POST" action="registar.php">
```

Deo koda “action=”registar.php”“ nam govori da će uneti podaci biti prosleđeni do PHP skripte pod imenom registrar.php koja se nalazi u istom direktorijumu.

Podaci se šalju u sledećim varijablama :

Ime :<input type="text" name="ime" size="20"

-- ime varijable = **name**

Prezime <input type="text" name="prezime" size="20"

-- ime varijable = **prezime**

3 CSS

3.1 ŠTA JE CSS

CSS (**Cascading Style Sheets**) je jezik koji može sadržati sve one podatke ili bolje rečeno sve syntax-e koje imaju zadatak da odrede stil (style) nekog elementa jedne stranice. Ti elementi mogu biti od običnog texta do jedne veoma kompleksno "stajlovane" tabele.

Svaki sajt koji ne sadrži u sebi neke od syntax-i a koristi CSS za definisanje stila pojedinog ili pojedinih delova stranice mora imati deo HTML (HyperText Markup Language) koda koji poziva neki CSS fajl sa syntax-ama. Mi ćemo krenuti baš od početka a to je kako i kojom komandom pozvati neki CSS dokument.

Recimo da se naš CSS dokument zove "stil" tada u <HEAD> našeg HTML-a pišemo sledeći kod:

```
<link href="stil.css" rel="stylesheet" type="text/css">
```

Ako ne želimo praviti poseban fajl za CSS syntax-e onda ih moramo pisati u naš HTML dokument i to na sledeći način:

```
<style type="text/css">
<!--
OVDE DOLAZE NAŠE SYNTAX-E
-->
</style>
```

U zadnjem kodu imamo sledeće <!-- ... -->, to možemo izostaviti, a ono služi samo da se sakriju syntax-e od browsera koji ne podržavaju CSS.

S ovim smo rešili pozivanje nekog CSS fajla u HTML dokument. Sad je vreme da saznamo kako pisati CSS dokument i pre svega koje syntax-e postoje u CSS-u.

NAPOMENA : Objašnjenje kompletnog CSS bi bilo preobimno za ovaj rad pa sam odlučio da predstavim samo one delove koje smatram potrebnim i interesantnim za realizaciju mog projekta.

3.2 CSS SYNTAX-e

3.2.1 SYNTAX-e ZA TEKST

Syntax-a {font-size}

Koristi se za definisanje veličine slova, a može se definisati na četiri načina: apsolutno, relativno, po dužini, ili procentualno. Možda najjednostavnije a i ono što većina koristi jeste definisanje veličine slova po dužini.

```
{ font-size: 30px; }
```

Objašnjenje: syntax-a je definisana preko dužine od 30px.

Sadržaj

Syntax-a {font-weight}

font-weight definiše debljinu odnosno koliko će naša slova biti boldirana. Ovde možemo koristiti ključne reči kao npr. *bolder* ili *lighter* ali i cele brojeve od 100-900 i primećujemo da ako stavimo veći broj naše slovo će biti deblje.

```
{ font-weight: bolder; }
```

Objašnjenje: debljina slova je postavljena na *bolder*.

Sadržaj

CSS Syntax-a: { font-weight: normal|bolder|lighter|100... 900; }

- normal ova ključna reč je defaultna
- lighter slova se udebljavaju ali malo
- bolder slova se udebljavaju ali više nego kod *lighter*

Syntax-a {font-style}

Ova syntax-a podešava slova na *italic* ili ih vraća u *normal* (prvobitni) oblik.

```
{ font-style: italic; }
```

Objašnjenje: slova će biti ukošena .

Sadržaj

- italic ukošena slova
- normal normalna slova

CSS Syntax-a: { font-style: normal|italic; }

Syntax-a {text-decoration}

Ova syntax-a dekorira naš tekst sa nekim od sledećih vrednosti: *underline*, *line-through*, ili defaultna vrijednost *none*:

```
{ text-decoration: line-through; }
```

Objašnjenje: tekst koji ima ovu syntax-u izgleda ovako:

Sadržaj

- *underline* podvlači text
- *line-through* linija kroz sredinu texta
- *none* nema nikakvih dekoracija

CSS Syntax-a: { text-decoration: none|underline|line-through; }

Syntax-a {text-align}

text-align se koristi za poravnanje teksta. Poravnanje se vrši sa četiri vrednosti. Defaultna vrijednost je *left*.

- *left* poravnava tekst sa leve strane
- *right* poravnava tekst sa desne strane
- *center* centrira tekst
- *justify* poravnava tekst sa obe strane

```
{ text-align: justify; }
```

Objašnjenje: tekst je poravnat s obe strane .

Sadržaj ovog teksta je
poravnat s obe strane.

3.2.2 CSS MARGINE

Margine su nevidljivi vanjski prostor između elemenata i u ovom delu ćemo saznati kako se definišu margine. Pri definisanju margina možemo postaviti da sve margine imaju istu vrednost ili da svaka ima različitu ali ono što je važno, pri davanju vrednosti jednoj margini ne smemo nikada koristiti negativne vrednosti jer može doći do nekih nesporazuma. Kada koristimo margine onda ih možemo podesiti sve od jednne ili pojedinačno svaku posebno. Definisanje margina se može izvršiti na četiri načina:

- *margin* ovde se upisuje samo jedna cifra ili ključna reč *auto* koja pravi isti prazni prostor oko elementa sa sve četiri strane
- *margin-top* podešavanje gornje margine
- *margin-right* podešavanje desne margine
- *margin-bottom* podešavanje donje margine
- *margin-left* podešavanje leve margine

Syntax-a {margin}

Ovu syntax-u koristimo samo ako želimo imati isti prazni prostor sa sve četiri strane elementa.

{ margin: 40px; } pravi 40px praznog prostora oko elementa

CSS Syntax-a: {margin: dužina|procentualno; }

Syntax-a za element koji ima različite margine izgleda ovako: { margin 11px 22px 33px 44px;}

Iz primera vidimo da je: gornja margina 11px, desna margina 22px, donja margina 33px i leva margina 44px. Redosled margina je uvijek isti i piše/čita se u smeru kazaljke na satu, gornja margina je prva a zatim sledi desna itd.

3.2.3 CSS BOJE - COLORS

Pre nego što krenemo sa syntax-ama kojime se definišu boje jednog elementa trebamo znati koje sisteme definisanja boja imamo. Postoje tri različita sistema od kojeg je najinteresantniji HexaDecimalni sistem ali mi ćemo obraditi sve.

Moguće je:

{ color: rgb(crvena%, zelena%, plava%); } <<< RGB% sistem

{ color: rgb(0-255, 0-255, 0-255); } <<< RGB sistem

{ color: #HexaDecimalni kod; } <<< HexaDecimalni sistem

HexaDecimalni sistem { color: #HexaDecimalni kod; }

U HTML-u ili CSS-u boje možemo definisati preko HexaDecimalnih kodova koji počinju sa znakom "rešetka" (#) i iza rešetke dolazi šest simbola "HexaDecimalni kod"(prilog 1). HexaDecimalni sistem sačinjavaju slova A, B, C.... F i brojevi 0, 1, 2, 3... 9. Neki od ovih simbola se poredaju u red od šest znakova i to čini određenu boju. Ovaj sistem koristi 99% webmastera.

3.2.4 BACKGROUND - POZADINA

background syntax-a ima zadatak da odredi izgled pozadine, ona je određuje na dva načina. Prvi način je boja pozadine a drugi je slika pozadine.

Syntax-a {background-color}, vrši podešavanje boje pozadine elementa.

Primeri:

{ background-color: green; }

{ background-color:#008000; }

{ background-color: rgb(0, 255, 0); }

{ background-color: rgb(0%, 100%, 0%); }

CSS Syntax-a: { background-color: ključna reč|#HexaDecimalni kod|rgb(0-255,0-255, 0-255)|rgb(0-100%,0-100%,0-100%); }

Syntax-a {background-image}

Postavlja sliku u pozadinu elementa. Ako imamo sliku koju želimo postaviti u pozadinu neke tabele onda to možemo uraditi preko sledeće syntax-e:

Predpostavimo da se slika zove logo.jpg: { background-image: url(folder do slike/logo.jpg); }

CSS Syntax-a: { **background-image: url(URLSlike);** }

3.2.5 BORDER - OKVIR

Borderi su oviri koji se nalaze oko nekog elementa.

Syntax-a {border-width}

Podešavanje debljine bordera.

```
{ border-width: 15px; }
```

Obajšnjenje: postavlja debljinu bordera na 15px .
Sadržaj

CSS Syntax-a: { border-style: dužina|procentualno; }

Syntax-a {border-style}

Podešavanje stila odnosno kako će okvir izgledati, moguće su *none*, *solid*, *dashed*, *dotted*, *inherit*, *double*, *inset*, *outset*, *groove*, *ridge* ključne reči. Da bi se border sa određenim stilom mogao prikazati potrebna mu je određena debljina, tu debljinu određujemo syntax-om **border-width**.

```
{ border-style: dotted; }
```

Obajšnjenje: ključna reč u syntax-i čini da okvir oko elementa bude isprekidan.
Sadržaj

CSS Syntax-a: { **border-style: none|solid|dashed|dotted|inherit|double|inset|outset|groove|ridge;** }

Syntax-a {border-color}

Podešavanje boje bordera. Da bi se border sa određenom bojom mogao prikazati potrebna mu je minimalna debljina od 1px, tu debljinu određujemo syntax-om **border-width**.

```
{ border-color: blue; }
```

Obajšnjenje: border je plav.

Sadržaj

{ **border-color: ključna reč| #Hexadecimalni kod|rgb(0-255,0-255,0-255)|rgb(0-100%,0-100%,0-100%);** }

Syntax-a {border-top-width}

Podešavanje debljine gornjeg dela okvira, tj. podešavanje debljine gornje linije.

```
{ border-top-width: 5px; }
```

Obajšnjenje: gornji border ima vrednost 5px.

Sadržaj

Syntax-a {border-right-width}

Podešavanje debljine desnog dela okvira, tj. podešavanje debljine desne linije.

```
{ border-right-width: 5px; }
```

Obajšnjenje: desni border ima vrednost 5px.

Sadržaj

Syntax-a {border-bottom-width}

Podešavanje debljine donjeg dela okvira, tj. podešavanje debljine donje linije.

```
{ border-bottom-width: 5px; }
```

Obajšnjenje: donji border ima vrednost 5px.

Sadržaj

Syntax-a {border-left-width}

Podešavanje debljine levog dela okvira, tj. podešavanje debljine leve linije.

```
{ border-left-width: 5px; }
```

Obajšnjenje: levi border ima vrednost 5px.

Sadržaj

Syntax-a {width}

Ova syntax-a podešava dužinu nekog elementa. **width** može biti izražen procentualno, apsolutno ili sa ključnom reči *auto* koja je i defaultna vrednost.

Neke od mogućih vrednosti syntax-e:

```
{ width: 500px; }
```

```
{ width: 7in; }
```

```
{ width: 100%; }
```

```
{ width: auto; }
```

```
{ width: 160px; }
```

Obajšnjenje: prikazana je tabela koja je zavisna od **width** syntax-e, pošto je syntax-a podešena na 160px i tabela će biti te iste dužine.

3.2.6 UPOTREBA CSS-a

Primer: napraviti CSS sa pet elemenata:

1. element 1 treba imati okvir "border" debljine 3px samo sa desne strane, ostale strane su 1 px
2. element 2 crvena boja slova sa crvenim isprekidanim "dotted" borderom debljine 1px
3. element 3 ima plavu pozadinu "background" sa crnim slovima
4. element 4 je tabela crvene boje i dužine "width" 30%
5. element 5 ima razmak "padding" od 10px sa svih strana, vrsta slova Verdana crne boje

Rešenje:

1. Otvoriti novi dokument u Notpad-u
2. U prazan dokument upisati sledeće CSS syntax-e:

```
.element1 { border-style: solid; border: 1px; border-right-width: 3px ; }
.element2 { color: red; border: 1px; border-style: dotted; border-color: red; }
.element3 { color: #000000; background-color: blue; }
.element4 { background-color: blue; width: 30%; }
.element5 { padding: 10px; font-family: verdana; color: black; }
```

3. Snimiti dokument sa nekim imenom npr. style.css

3.2.7 UPOTREBA CSS-a U HTML-u

Ako već imamo napisane CSS syntax-e onda je vreme da ih iskoristimo, to se radi ovako:

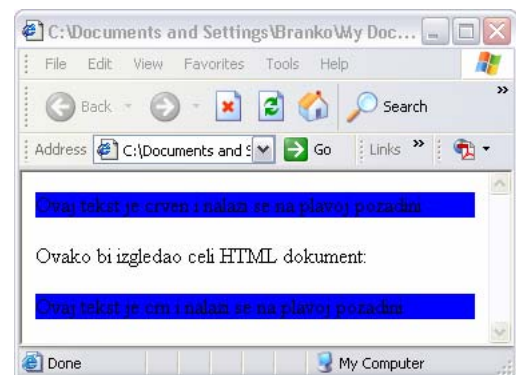
Primjer: imamo tekst koji uzima syntax-u elementa 3 odnosno tekst će biti crn sa plavom pozadinom.

Kod:

```
<p class="element3">Ovaj tekst je crven i nalazi se na plavoj pozadini</p>
```

Ovako bi izgledao celi HTML dokument:

```
<html>
<style type="text/css">
.element1 { border-style: solid; border: 1px; border-right-width: 3px ; }
.element2 { color: red; border: 1px; border-style: dotted; border-color: red; }
.element3 { color: #000000; background-color: blue; }
.element4 { background-color: blue; width: 30%; }
.element5 { padding: 10px; font-family: verdana; color: black; }
</style>
<body>
<p class="element3">Ovaj tekst je crn i nalazi se na plavoj pozadini</p>
</body>
</html>
```



Slika4: CSS u akciji

4 PHP

4.1 UVOD U PHP

Php je open-source server-side skriptni programski jezik za dinamičko generisanje HTML koda.

Drugim rečima, PHP je skriptni programski jezik pomoću kojeg možete kreirati HTML stranicu na serveru , pre nego što je ona poslana klijentu, popunjenu dinamičkim sadržajem. Ovim načinom generisanja sadržaja klijent ne može vidjeti kod (skriptu) koji je generisao sadržaj koji gleda, već ima pristup čistom HTML kodu.

Open-source u gornjoj definiciji znači da svako ko želi može skinuti izvorne PHP kodove pisane u C-u i, ukoliko ih razume, može ih mijenjati po svojoj volji te dodavati nove funkcije PHP-u. Štoviše, svi su pozvani da sudjeluju u razvoju novih verzija PHP-a. Izvorne kodove i instalacijske datoteke možete skinuti sa službenog PHP sajta.

PHP je jedna od najnaprednijih i najkorištenijih server-side skriptnih tehnologija danas u upotrebi. On je svojom sintaksom sličan mnogim drugim sličnim jezicima, čak i ima istoznačne (iste po sintaksi i funkcionalnosti) funkcije kao i neki drugi jezici kao što su **C** ili **Perl**. To znači da jednu radnju možete izvesti korištenjem više različitih funkcija. Ova osobina u mnogome može da olakša rad programerima.

Recimo ova dva primera(Listing2) rezultiraju istim prikazom:

```
<?
echo 'Pozdrav svima';
?>

i

<?
printr( 'Pozdrav svima');
?>
```

Listing2

Rezultat oba primera iz listinga2 imaće oblik :

Pozdrav svima

Još jedna važna stvar je ta da je PHP bogat funkcijama za manipuliranje mnogo različitih tipova sadržaja. Od manipuliranja grafikom (png, jpg, flash...) do loadanja .NET modula i rada sa XML-om.

Ono što PHP stavlja još više ispred ostalih web skriptnih tehnologija je njegova podrška za baratanje širokom paletom baza podataka. Podržava sve popularnije baze podatak kao MySQL, PostgreSQL, dBase, Oracle, ODBC...

Isto tako njegova neovisnost o operativnom sistemu i pristupačne cijene (besplatan je) ga čini među prvim izborom velikih i malih kompanija za izradu vlastitih mrežnih sistema.

4.2 PHP U POREĐENJU SA DRUGIM JEZICIMA

PHP vs ASP

Prva razlika je ta što ASP sam po sebi nije jezik već skupina povezanih objekata kojima možemo pristupiti pomoću VB ili Java Scripta. Druga stvar koja stavlja ASP u drugi plan je ta što je ASP podržan na Win32 sistemima sa IIS-om, a morali bi izdvojiti poveću sumu da bi ga pokrenuli na drugim platformama ili serverima. Veliki problem kod ASP-a je to što je programer ograničen samim jezikom koji je podeljen na 'komponente' i ukoliko želimo šire mogućnosti morali bismo nadograđivati ASP dodatnim komponentama, što u Microsoft žargonu znači dodatni troškovi!

PHP vs Perl

Ova dva jezika imaju dosta sličnosti, ali više razlika. Glavna razlika je ta što je PHP namenjen prvenstveno za web skriptanje a Perl ima mnogo širu upotrebu. Samim time ima i složeniju sintaksu od PHP-a što ga čini teže za naučiti i manje 'shvatljivim' od PHP-a. Unatoč široj uporabi Perla, PHP koristi mnoge 'dobre' osobine Perla poput konstruktora i nekih sintaktičkih osobina .

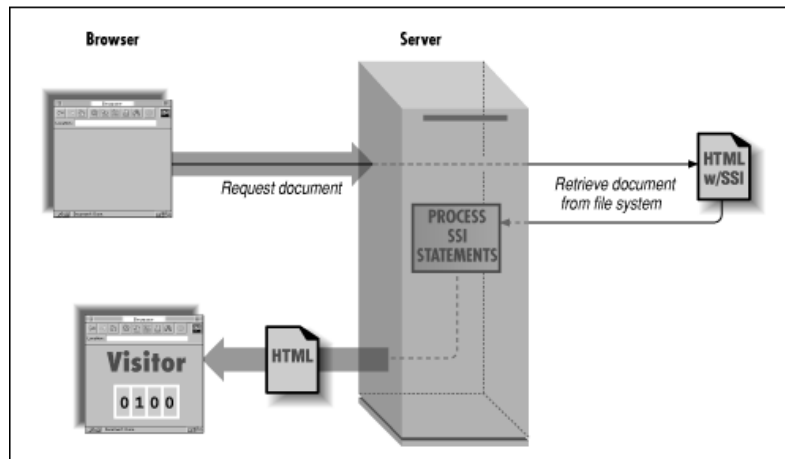
PHP vs Cold Fusion

Osnovna razlika između ova dva jezika je ta što je PHP stvoren sa misli na pravog programera sa iskustvom u nekom C stil jeziku, dok su stvoritelji Cold Fusiona imali u prvom planu neprogramere i njima su prilagodili sintaksu. Isto tako, PHP je pouzdaniji i otvoreniji inovacijama i programerskim trikovima od CF-a.

4.3 RAZLIKA IZMEĐU S-S i C-S SKRIPTNIH JEZIKA

Server-side

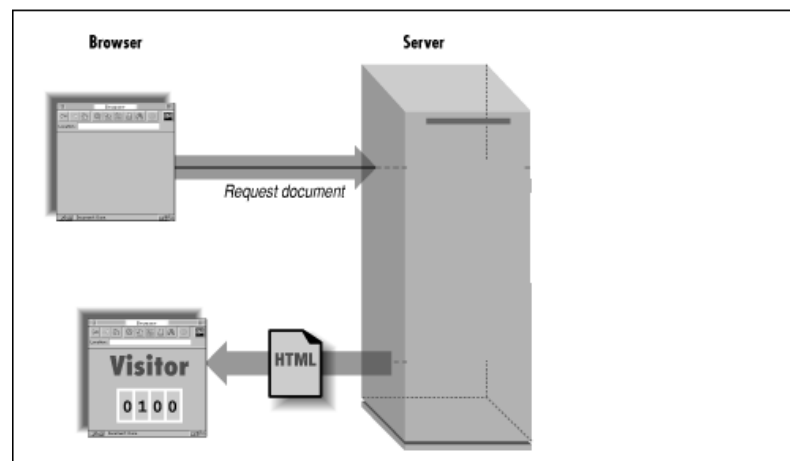
Server **side skripte** se izvršavaju na serveru kada server primi zahtev za PHP dokumentom. Nakon prihvatanja zahteva sa PHP dokumentom server izvršava PHP kod i na osnovu njega generiše HTML kod i šalje ga klijentu. To znači da stranica koja se prikazuje u pretraživaču klijenta ne postoji u tom obliku nigde na serveru odakle ju je klijent primio.



Slika5: Server-side skriptni jezik

Client-side

Glavni i najpoznatiji predstavnik ove skupine jezika je JavaScript. Kod pisan u JavaScriptu je obično umetnut u HTML stranicu i izvršava se tek u klijentovom pretraživaču. Ovakav kod vidljiv je svima, osim ako nemate malo iskustva i spremite svoje kodove u nekakav include file te time sakrijete svoje kodove.



Slika6: Client-side skriptni jezik

4.4 SINTAKSA PHP-a

Već smo u prijašnjem primeru mogli videti neke bitne stvari. Npr. da se sav PHP kod nalazi između `<? i ?>` kvačica. Mali dodatak ovom pravilu bi bio korišćenje `<?php ... ?>` kvačica radi razlikovanja između PHP i XML koda (naime i XML koristi iste ove kvačice). Druga stvar koja je očita iz tog primera je da varijable prije svog imena imaju znak \$.

4.5 PRELAŽENJE IZ PHP-a U HTML MOD

Vrlo bitna karakteristika PHP-a i bilo kojeg drugog jezika je razdvajanje server side koda od statičkog HTML-a. Tako u PHP-u korišćenjem `<? i ?>` govorimo serveru da se između njih nalazi PHP kod i da je potrebno prvo njega izvršiti i tek nakon toga poslati HTML output tog koda skupa sa ostatkom statičkog HTML koda klijentu. Ovo je osnova server-client mrežne komunikacije putem TCP/IP protokola.

Unutar koda se možemo u bilo kojem trenutku prebaciti iz HTML moda u PHP mode. Čak i unutar IF, FOR i ostalih kontrolnih struktura. Ovo nam omogućuje rad sa templateima u kojima se na isti način prikazuju različiti podaci iste strukture. Da ilustriram jednim vrlo jednostavnim primerom. Listing3.

```
<html>
<head>
<title>TEST</title>
</head>
<body>
<?
$time="Marko";
$prezime="Markic";
?>
<table width="300" border="1" cellspacing="0" cellpadding="0">
<tr>
<td>
<?
echo $time;
?>
</td>
<td><?=$time?></td>
</tr>
</table>
</body>
</html>
```

Rezultat Listinga3 bi izgledao

Marko	Markić
-------	--------

Uočite da za ispis sadržaja varijable koristimo dva različita načina koji su rezultirali istim krajnjim ispisom. Naime unutar `<? i ?>` se nalazi php kod koji će se izvršiti ukoliko se u njemu ne nalaze neke sintaktičke greške. Druga metoda ispisa je izgledala ovako :

```
<?=$prezime?>
```

Ovu metodu koristimo kada želimo ispisati neku varijablu ili string. Ova linija je identična

```
<? echo $prezime ?>
```

samo što smo umjesto echo naredbe koristili znak = koji PHP-u govori da ispiše ono što se nalazi između `<? i ?>` kvačica.

4.6 POVEZIVANJE HTML OBRAZCA SA PHP SKRIPTOM

Često ćete na stranicama automatizovanih PHP-om, imati formulare. Najčešće će se raditi o anketama, formularima za komentare, glasanje i slično. Na svu sreću, kada se jedan ovakav formular prosledi PHP skriptu, sve varijable definisane na formi biće automatski dostupne PHP skriptu. Ovo znači da skript može da prihvati i varijable koje nisu definisane u samom PHP skriptu. Na primer, imamo jednostavan formular (Listing4), koji se metodom "POST" šalje na obradu PHP skriptu pod nazivom "primer.php":

```
<FORM ACTION="primer.php" METHOD="POST">  
  IME: <INPUT TYPE="text" NAME="ime"><BR>  
  <INPUT TYPE="submit" VALUE="Posalji">  
</FORM>
```

Listing4 : kreiranje HTML forme za unos imena

Kada ovaj formular popunite i kliknete na dugme "Posalji", metodom "POST" elementi formulara će biti prosleđeni PHP stranici "primer.php" na dalju obradu. Stranica "primer.php" će automatski primiti sadržaj iz prosleđenog polja "ime" u varijablu "\$ime". Evo kako bi mogao da izgleda skript "primer.php" Listing4:

```
<?php  
echo "Vase ime je:<BR>";  
echo $ime;  
?>
```

Listing4 : primer PHP skripte za primanje I obradu podataka iz HTML obrazca

Znači, kada popunimo polje "IME" na stranici sa formularom, i kliknemo na dugme za slanje, sadržaj ovog polja će primiti varijabla "\$ime" u PHP skriptu na stranici "primer.php", te će se uz pomoć instrukcije "echo" prikazati vrednost date varijable.



Slika6: Izlaz listinga4

Vase ime je :
Dragoslav

Slika5 : Izlaz PHP skripte primer.php

4.7 NEKE OD OSNOVNIH PHP TEHNIKA

4.7.1 IF iskaz:

IF iskazi su neophodni u programiranju funkcionalnih sajtova. Ako zelimo da proverimo odnosno uporedimo dve ili više vrednosti koristićemo IF iskaz. Sintaksa IF iskaza je jednostavna I najbolje bi bilo da se opiše uz pomoć jednostavnog primera :

Recimo da korisnik unosi jedan broj u varijablu \$broj i da nasa skripta proverava.php proverava da li je taj broj veći od 10 ili ne . Evo koda ovog primera :

```
$broj = 14;  
  
if ( $broj > 10 ) {  
    echo "Uneti broj je veći od 10!<br />";  
}  
echo "Uneti broj nije veći od 10!";
```

Listing5 : primer koda IF iskaza

4.7.2 WHILE petlja:

Nakon "If" strukture prelazimo na obradu najjednostavnijeg tipa petlji u PHP-u, a to je "While" petlja. Struktura i sintaksa u PHP-u je vrlo slična sintaksi ove petlje u programskom jeziku C:

```
while (izraz)
    naredbe
```

Semantika same petlje je vrlo jednostavna, ona "kaže" PHP-u da ponavlja izvršavanje ugrađenih naredbi, sve dok je ispunjen uslov, tj. dok je vrednost "izraza" TRUE. Vrednost izraza se proverava samo na svakom početku ponavljanja petlje, tako da ako se ova vrednost promeni tokom izvršavanja ugneždenih instrukcija, izvršenje se neće prekinuti do kraja date iteracije. U slučaju da je vrednost "izraz"-a FALSE, neće se izvršiti ugneždene naredbe, a izvršenje programa će se prebaciti na sledeću instrukciju koja dolazi nakon petlje.

Kao i kod "IF" grananja, i ovde možete grupisati više instrukcija unutar jedne "While" petlje, i to uz pomoć vitičastih zagrada, ili uz korišćenje alternativne sintakse:

```
while (izraz) :
    ...naredbe...
endwhile;
```

U sledećim redovima možete videti dve "While" petlje, napisane na dva načina, koje "štampaju" cifre od 1 do 10:

```
<?php
/* 1. primer */

$a= 1;
while ($a <= 10) {
    print $a++;
}

/* 2. primer */

$a = 1;
while ($i <= 10):
    print $a;
    $a++;
endwhile;
?>
```

Listing6 : Primer koda WHILE petlje

Kao što vidite, ovde smo upotreбили inkrementalni operator (++) koji uvećava vrednost varijable "\$a", i to nakon njenog štampanja.

Slična prethodnoj "While" petlji je "Do...While" petlja, a razlika je samo u poziciji proveravanja "izraz"-a. Kod ove petlje proverava istinosne vrednosti izraza se vrši na kraju petlje svake iteracije. Glavna razlika je u tome što će se kod ove druge petlje (do...while) prva iteracija svakako izvršiti, za razliku od "While" gde se možda neće izvršiti ni jedna iteracija (u slučaju da je vrednost izraza FALSE).

```
<?php
$a = 0;
do {
    print $a;
} while ($a>0);
?>
```

Listing7 : Primer koda DO WHILE petlje

Prethodna petlja će se izvršiti tačno jedanput, jer se nakon prve iteracije proverava izraz (\$a>0), koji će imati vrednost FALSE. Zbog ove vrednosti izraza petlja će prekinuti dalje izvršavanje

4.7.3 FOR petlja:

Za razliku od "jednostavne" sintakse "While" petlje, "For" petlja ima najsloženiju sintaksu u PHP-u (vrlo sličnu C-u). Struktura i sintaksa je ove petlje je sledeća:

```
for (izraz1; izraz2; izraz3)
    naredba
```

Prvi izraz (izraz1) se proverava (izvršava) samo jednom (bezuslovno), i to na početku petlje. Na početku svake iteracije proverava se drugi izraz (izraz2). Ukoliko je istinosa vrednost ovog izraza TRUE, petlja se nastavlja i izvršava se ugneždena instrukcija(e). Ukoliko je vrednost izraza FALSE, prekinuće se izvršavanje petlje. Takođe, na kraju svake iteracije petlje proverava se (izvršava se) treći izraz (izraz3). Svaki od nabrojanih izraza može biti "prazan", tj. ne mora sadržati nikakav izraz. Ako je na primer, drugi izraz "prazan" petlja će se ponavljati nedefinisan broj puta. Da pogledamo par primera za FOR petlju.

```
<?php
/* 1. primer */
for ($a = 1; $a <= 10; $a++) {
    print $a;
}
/* 2. primer */
for ($b = 1;; $b++) {
    if ($b > 10) {
        break;
    }
    print $b;
}
/* 3. primer */
$c = 1;
for (;;) {
    if ($c > 10) {
        break;
    }
    print $c;
    $c++;
}
/* 4. primer */
for ($d = 1; $d <= 10; print $d,
    $d++);
?>
```

Listing8 : Primer koda FOR petlje

Od ova četiri primera, prvi je svakako najjasniji. U drugom primeru je prikazan "prazan" drugi izraz, pa imamo nedefinisan broj iteracija petlje. Ali, tu smo upotrebili "If" grananje i instrukciju "break", kojom izlazimo iz petlje (petlja se prekida) kada vrednost varijable "\$b" bude 11 (nakon toga vršimo štampanje vrednosti varijable). Treći primer je specifičan jer su sva tri izraza "prazna" (ndefinisana). Tako da će se data petlja izvršavati u zavisnosti od prirode ugneždenih naredbi. PHP naravno, obezbeđuje i alternativno definisanje "For" petlje (sa dvotačkom):

```
for (izraz1; izraz2; izraz3):
    ...naredba; ...;
endfor;
```

4.8 UPOTREBA KOLAČIĆA (cookies)

Vrednosti većine promenljivih nestaju nakon što PHP skript, u kome se nalaze, završi rad. Kolačići su , za razliku od toga , promenljive koje mogu da postoje neograničeno dugo. Pošto njihove vrednosti mogu da ostanu stalne , onda pretraživač korisnika čuva kolačiće da korisnikovom lokalnom disku.

Kolačići mogu biti od koristi na različite načine. Na primer, mnogi sajtovi koriste kolačiće za čuvanje identiteta korisnika i njegovih osobina . Kada se korisnik vrati na taj sajt , kolačić omogućava pretraživaču da prepozna korisnika i da obnovi opcije koje je korisnik podesio.

Recimo da na sajtu za elektronsku trgovinu koja prodaje razne artikule , od odeće , knjiga pa sve do kuća i brodova, omogućimo korisniku da izabere koja ga oblas najviše interesuje. Sledećiput kada korisnik poseti sajt automatski će dobiti novosti u oblasti koja ga najviše interesuje.

Na žalost, kolačići nisu savršeni način za stalno skladištenje podataka. Kod njih postoje različita ograničenja. Na prime ;

- Korisnik može da isključi kolačiće ako podesi odgovarajuću opciju u svom pretraživaču.
- Pod izvesnim okolnostima, kolačiće mogu da vide i drugi korisnici , a ne samo oni kojima kolačići pripadaju
- Sajt može da na disku korisnika smesti samo 20 kolačića i 4 KB inafomacija
- Neke verzije popularnih pretraživača imaju greške i ne mogu da koriste kolačiće na pravilan način

Uprkos pomenutim ograničenjima , kolačići su i dalje najpopularnija tehnika za stalno skladištenje podataka. Zbog toga ih je važno spomenuti

Primeri upotrebe cookie-a :

- **Programski kod koji kreira kolačiće**, iz razloga efikasnosti, trebao bi biti **pozicioniran na početku web stranice** pre bilo kog drugog koda.
- **Cookie fajlove potrebno je smestiti u root direktorijum sajta** kako bi ga mogli koristiti sa bilo koje pozicije vašeg web sajta.

Sledi programski kod kojim kreiramo varijablu *cookie*:

```
<?
setcookie ("cookie", "personalmag kolacic");
//kreiranje varijable $cookie sa vrednošću "kolach"
?>
```

Kreiranjem i postavljanjem ovog koda, kada posetioc ponovo dođe na tekuću stranicu ili bilo koju drugu PHP stranicu u tekućem folderu (ili bilo kom podfolderu), postavljeni kolačić biće dostupan za "konzumiranje". Ukoliko bi sada komandom ECHO ispisali vrednost varijable - `$_COOKIE["cookie"]`, dobili bi nazad "kolach".

Nadalje, ukoliko želite da postavite *expire* vremenski rok kolačića, dodaćete samo još jedno polje sa količinom vremena (u sekundama) koja treba da protekne do "isteka važnosti":

```
<?
setcookie ("cookie", "personalmag kolacic", time()+3600);
//isti kod kao prethodni, s tim da će kolačić ovaj "trajati"3600 sekundi (1 čas) u odnosu
na trenutno vreme
?>
```

Broj kolačić-varijabli nije ograničen, tako da uvek možete postaviti više od jendog kolačića:

```
<?
setcookie ("cookie1", "personalmag kolacic 1 ");
setcookie ("cookie2", "personalmag kolacic 2 ");
setcookie ("cookie3", "personalmag kolacic 3 ");

//kolačiće možete isčitati sa:

echo $_COOKIE["cookie1"];
echo $_COOKIE["cookie2"];
echo $_COOKIE["cookie3"];

// a ako želite da prikazete sadržaj svih varijabli
//upotrebite onda sledeću instrukciju:

print_r($_COOKIE);
?>
```

I konačno, kolačiće je moguće i izbrisati, ukoliko vam više nisu potrebni, upotrebom gorenavedene instrukcije **setcookie**, samo što ovaj put varijabli nećemo dodeliti nikakvu vrednost. Postavićemo još jednu **setcookie** funkciju samo sa nazivom varijable:

```
<?
setcookie ("cookie");
//brisanje "kolačića"
?>
```

5 BAZE PODATAKA

Jedan od možda najvažnijih delova sajtova za elektronsku trgovinu jesu baze podataka. U njima se skladište podaci o svim proizvodima , kao i njihovim karakteristikama , količini , ceni i još mnogo drugih važnih informacija.

Pošto se najčešće sa PHP-om koristi MySQL baza podataka u narednom delu ću predstaviti tehniku upotrebe PHP i MySQL tehnologije.

Podsetimo se samo da PHP podržava i ostal popularne baze podatak kao PostgreSQL, dBase, Oracle, ODBC...



5.1 UPOTREBA MYSQL BAZA PODATAKA

PHP ima celu biblioteku funkcija koje čine interfejs prema MySQL sistemu za upravljanje bazama podataka. Preko ovih funkcija, PHP program može da pristupi i da menja podatke, koji se nalaze u MySQL bazi podataka.

Interakcije sa bazama podataka se uglavnom se odvijaju po sledećem obrascu :

- Upostavlja se veza sa MySQL serverom
- Definiše se baza podataka kojoj se pristupa
- Izvršavaju se SQL upiti, pristupas e rezultatima tih upita i obavlja ju se druge operacije koje nisu u direktnoj vezi sa SQL-om
- Prekida se veza sa MySQL-om

Povezivanje sa MySQL serverom

Za povezivanje sa MySQL serverom koristićemo funkciju `mysql_connect()` koja ima sledeći oblik:

```
mysql_connect(hostname, username, password)
```

Hostname predstavljaj adresu servera na kome se nalazi baza podataka, username predstavlja MySQL korisnika a password predstavlja lozinku koja je pridružena tom korisniku.

Izbor baze podataka

Nakon što smo uspostavili vezu sa MySQL serverom, program može da definiše bazu podataka kojoj treba da se pristupi. Ovo ćemo uraditi uz pomoć funkcije `mysql_select_db()` koja ima sledeći oblik

```
mysql_select_db(baza podataka)
```

Gde je „baza podataka“ ime baze kojoj pristupamo. Funkcija vraća TRUE ako je u stanju da pristupi bazi podataka, u suprotnom vraća FALSE.

Prekidanje veze sa MySQL serverom

Ako želite da prekinete vezu sa MySQL serverom, možete da upotrebite funkciju `mysql_close()`, koja ima sledeći oblik :

```
mysql_close()
```

NAPOMENA : korišćenje MySQL-a je veoma opširan pojam kao i upotreba PHP, tako da sam ja samo spomenuo neke osnovne stvari da bi vam dočarao moć ovih tehnologija.

6 PROJEKAT

6.1 INDEX

Izgled same web stranice je od velikog značaja jer pri poseti korisnik prvo što upčava jeste dizajn stranice. Pod ovim se podrazumeva da je sadržaj stranice pregledan , raspored sadržaja da je adekvatno pozicioniran i da je odvojen od drugih delova stranice. Većina stranica na internetu se drže proverenog šablona , tako da je gotovo uvek sigurno pridržavati se njega.

To izgleda ovako :

- Banner
- Meni
- Sadržaj
- Footer (tekst na dnu stranice)

Sledeća slika pokazuje kako sam zamislio raspored i dizajn ovove web aplikacije :



Slika8 : Index stranica web aplikacije

1. Banner
2. Meni
3. Glavni prozor (sadržaj
4. Footer (u ovom slučaju iskorišćen i kao reklamni prostor)

6.2 MENI

Ono što svaki web sajt treba da poseduje jeste meni za navigaciju. Obimnost ovih menija za navigaciju zavisi i od obimnosti web sajtova na kojima se nalaze kao i broja funkcija koje web sajt izvršava. Tako da krstareći po internetu nailazimo na menije koji se sadrže od svega dva ili tri linka pa do onih koji imaju i više desetina linkova.

Što se tiče menija za web aplikaciju za internet online oglase, o kojoj je reč u ovom radu, obimnost mu je svedena na minimum koji zadovoljava sve potrebe za uspešno korišćenje ove aplikacije.

Pošto na ovom sajtu postoji registracija korisnika poželjno je bilo da se naprave dva menija, jedan za ne registrovane korisnike i jedan, malo opširniji za registrovane korisnike.

Kako ovo izgleda predstavljaju sledeće slike :

Meni za ne registrovane korisnike sadrži svega tri linka i to :

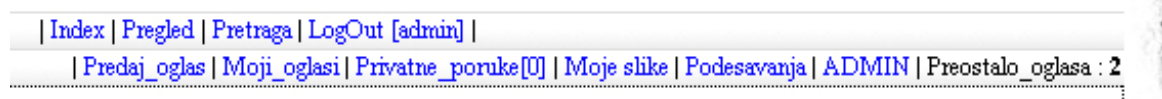
1. Index – za povratak na početnu stranicu
2. Pregled – za pregled već postojećih oglasa
3. Login – odnosno u ovom slučaju registracija novog korisnika
4. Kao i info poruku da je korisnik ne registrovan



Slika9 :Meni za neregistrovane korisnike

Meni za registrovane korisnike sadrži pored menija za ne registrovane korisnike još i :

1. Pretraga – omogućava pretraživanje oglasa
2. Predaj_oglas – predaja novih oglasa
3. Moji_oglas – pregled oglasa čiji je autor korisnik
4. Privatne_poruke – pregled privatnih poruka (u linku se nalazi i broj trenutnih poruka)
5. Podesavanja – izmena već postojećih podataka o korisniku
6. ADMIN – link ka administraciji sajta (prikazuje se samo korisnicima sa ADMIN pravom)
7. Info poruka o broju preostalih oglasa za predaju



Slika10 : Meni za registrovane korisnike

7 RAZVOJ

Posle predstavljanja osnovnih tehnika za razvoj web aplikacije za internet online oglase vreme je da objasnim sam razvoj istog. Zbog preobimnosti objašnjavanja kompletnog postupka kao i samog koda web aplikacije odlučio sam da razvoj predstavim objašnjavanjem određenih funkcija web sajta koje predstavljaju ključni faktor za funkcionisanje. Uz adekvatno objašnjenje funkcionalnosti pokušaću da predstavim i pojedine delove samog koda i da ih objasnim. Ovo smatram neizostavnim jer mislim da će na taj način shvatanje funkcionisanja ove aplikacije biti u mnogome pojednostavljen.

7.1 SPISAK MySQL TABELA

Pre nego što počnemo sa objašnjavanjem funkcionalnosti same aplikacije, treba dati na uvid sve potrebne tabele u bazi koje će nam služiti sa pohranjivanje podataka tokom rada web aplikacije.

U sledećim listinzima prikazane su sve tabele.

grad

Field	Type	Null	Default	Comments
<u>grad_ID</u>	int(11)	No		
grad	varchar(30)	No		

kategorija

Field	Type	Null	Default	Comments
<u>kategorija_ID</u>	int(11)	No		
kategorija	text	No		

news

Field	Type	Null	Default	Comments
<u>news_ID</u>	int(11)	No		
news	text	No		
slika	text	No		

pvm

Field	Type	Null	Default	Comments
<u>pvm_ID</u>	int(11)	No		
za_user	text	No		
od_user	text	No		
od_mail	text	No		
datum	text	No		
poruka	text	No		

oglas

Field	Type	Null	Default	Comments
<u>oglas ID</u>	int(11)	No		
title	text	No		
oglas	text	No		
cena	text	No		
datum	text	No		
link	text	No		
br_pregleda	int(11)	No	0	
tip_oglasa	int(11)	No	0	
grad	text	No		
kategorija	text	No		
user	text	No		
mail	text	No		
tel1	text	No		
tel2	text	No		

users

Field	Type	Null	Default	Comments
<u>users ID</u>	int(11)	No		
username	varchar(30)	No		
password	varchar(50)	No		
mail	varchar(50)	No		
ime	varchar(30)	No		
prezime	varchar(30)	No		
tel1	text	No		
tel2	text	No		
adresa	text	No		
br_dozv_oglasa	int(11)	No	5	
grad	text	No		
admin	int(11)	No	0	
limit2	int(11)	No	20	

8 FUNKCIJE

Za prikazivanje u ovom radu izabrao sam sledeće funkcije :

- Instalacija
- Registracija novog korisnika
- Registracija već postojećeg korisnika (login)
- Pregled oglasa
- Predaja oglasa
- Brisanje oglasa
- Pretraga oglasa
- Podešavanja
- Privatne poruke
- Administracija
 - Korisnici
 - Gradovi
 - Kategorije
 - Vesti

U opisu gore navedenih funkcija moguće je da će se spominjati neke koje nisam naveo ali iz već navedenih razloga ih neću objašnjavati.

8.1 INSTALACIJA

Da bi smo uspešno mogli koristiti ovu aplikaciju na jednom od web servera prvo što treba da uradimo jeste da je instaliramo. Što se tiče web servera on mora da podržava PHP kao i MySQL jeru protivnom aplikacija neće raditi. Ono što se očekuje da pored ovoga na serveru već postoji , ako ne onda bi administrator trebao da obezbedi , jeste baza podataka sa imenom „diplomski2“. Ovako je po standardu , mada se lako može izmeniti na sledeći način:

U root direktorijumu postoji fajl po imenu config.php koji izgleda kao na listingu ispod ,

```
<?php
    $database = 'diplomski2';
    $host      = 'localhost';
    $username  = 'root';
    $password  = 'subotica';
?>
```

Uočavamo 4 promenljive :

- \$database – ime baze podataka na serveru
- \$host – adresa servera na kome se nalazi web aplikacija
- \$username – korisničko ime za pristup bazi podataka
- \$password – lozinka za pristup bazi podataka

U slučaju da se na serveru baza drugačije zove , kao npr. „oglas“ , da nas je administrator registrovao po korisničkim imenom „korisnik“ i sa lozinkom „sifra123“ , u tom slučaju bi trebali izmeniti fajl config.php da izgleda kao na listingu ispod.

```
<?php
    $database = 'oglas';
    $host      = 'localhost';
    $username  = 'korisnik';
    $password  = 'sifra123';
?>
```

Promenljivu \$host nismo menjali jer se u većini slučajeva sve nalazi na jednom serveru , odnosno na računaru gde se i nalazi naša aplikacija pa zbog toga će vrednost „localhost“ omogućiti nesmetan rad web aplikacije.

Ovo se odnosi na prilagođavanje sajta serveru na kom se nalazi. Bez ovoga bi bilo nemoguće izvršavanje aplikacije . Ovo nije potrebno samo za instalaciju već će se veći deo sajta oslanjati na ove podatke koji su potrebni za svaki pristup bazi podataka kao i za čitanje tako i za upis podataka u nju.

Kada smo izvršili pravilnu izmenu config.php fajla sada je potrebno da pređemo na instalaciju same aplikacije. Ovo ćemo uraditi tako što ćemo da pokrenemo install.php fajl.

NAPOMENA : u daljem tekstu biće dosta reči o adresi fajlova kao i same aplikacije. S obzirom da se aplikacija nalazi na računaru za kojim i sedim (na kojem je takođe instaliran i web i mysql server) onda je pristup omogućen preko sledeće adrese <http://localhost/diplomski/> gde „localhost“ predstavlja adresu servera a „diplomski“ predstavlja direktorijum na serveru gde se nalazi aplikacija.

Unošenjem adrese <http://localhost/diplomski/install.php> u naš web browser dolazimo do jednostavne HTML forme preko koje ćemo da unesemo osnovne podatke o administratoru i da potom instaliramo i potrebne tabele u bazi podataka za rad naše aplikacije.

Slika11 : HTML forma za unos podataka o administratoru i instaliranje aplikacije

Iz ove slike zaključujemo da imaju tri tekst polja (text box-a) i to : jedan za unos korisničkog imena (čiji je naziv promenljive „admin_username“) i druga dva za unos lozinke („admin_password“ i „admin_password2“). Nazivi promenljivih za data polja su nam potreba kasnije u kodu za prihvatanje vrednosti koje se unose.

Ovde vidimo da moramo dva puta da unesemo lozinku (identične) pa da razjasnim zašto. Pošto su polja za unos lozinke zaštićena , nije dozvoljen pregled šta se unosi, nego korisnik vidi samo zvezdice (****) i ako bi bilo samo jedno polje postoji mogućnost da korisnik unese slučajno pogrešnu lozinku koja će se uneti u bazu. Zbog ovoga se primenjuje unos dva puta iste lozinke koje se kasnije u kodu proveravaju da li su identične.

```
Admin Password <input type="password" name="admin_password" size="20">
```

HTML kod koji pravi tekst polje za unos lozinke (zaštićeno zvezdicama)

Slika11Izgled zaštićenih polja u web browseru

Da bismo shvatili kako ovo sve radi , prikazaću kod iz install.php fajla i objasniti najbitnije delova za šta služe.

```

1 <form method="POST" action="install.php?action=install">
2
3     <div align="center">
4         <center>
5             <table border="1" width="300" cellspacing="0" cellpadding="0">
6                 <tr>
7                     <td width="296" bgcolor="#5184AE" colspan="2">
8                         <p align="center"><font color="#FFFFFF"><b>Instalacija online oglasa </b></font>
9
10
11                     </td>
12                 </tr>
13                 <tr>
14                     <td width="296" bgcolor="#3596CA" colspan="2">
15                         <p align="center">
16 Admin Username <input type="text" name="admin_username" size="20"><br>
17 Admin Password <input type="password" name="admin_password" size="20"><br>
18 Admin Password <input type="password" name="admin_password2" size="20">
19                         </td>
20                     </tr>
21                     <tr>
22                         <td width="243" bgcolor="#5184AE">
23                             <p align="center"><input type="submit" value="INSTALL" name="B1">
24                             </td>
25                         <td width="51" bgcolor="#5184AE">
26                             <a href="install_help.php" target="_blank"><font color="#FFFF00">pomoc</font></a>
27                         </td>
28                     </tr>
29                 </tr>
30             </table>
31         </center>
32     </div>
33
34
35 </form>

```

HTML kod za kreiranje install forme

Objasnenje po linijama koda :

- 1 – Kreiranje HTML forme za slanje podataka ka install.php stranici kojoj smo jos dodelili vrednost „install“ promenljivoj „action“
- 5 – Kreiranje tabele sa određenim parametrima
- 16,17,18 – Kreiranje polja za unos podataka
- 23 – Kreiranje dugmeta za potvrdu slanja podataka iz forme
- 26 – Kreiranje teksta „pomoc“ sa hyperlinkom ka install_help.php stranici
- 35 – Kraj forme

NAPOMENA : u priloženim listinzima koda brojevi linija su naknadno postavljeni tako da u originalnom kodu nesmeju da budu.

Pošto smo kreirali formu za unos podataka sada ću da predstavim kod za primanje tih podataka kao i za kreiranje tabela u MySQL bazi i na kraju unos podataka ako sve prođe bez problema. Napominjem da se i ovaj kod nalazi u install.php fajlu.

Postupci su sledeći :

- Prihvatanje podataka iz HTML forme
- Provera podataka
- Konektovanje na MySQL bazu
- Kreiranje tabela
- Unos podataka iz HTML forme u bazu
- Kraj

```
1 $admin_username_buff = $_POST['admin_username'];
2 $admin_password_buff = $_POST['admin_password'];
3 $admin_password2_buff = $_POST['admin_password2'];
4
5
6 if($HTTP_GET_VARS['action']=='install' AND $admin_password_buff == $admin_password2_buff
7 AND $admin_username_buff != "" OR $admin_password_buff != "" ) {
8
9 echo "<BR>";
10 mysql_connect($host, $username, $password) or die(mysql_error());
11 echo "<BR>";
12 echo "Uspesno konektovanje na bazu";
13 echo "<BR>";
14
15 mysql_select_db($database) or die(mysql_error());
16
17 echo "<BR>";
18 echo "Uspesna selekcija baze";
19 echo "<BR>";
20 mysql_query("CREATE TABLE `users` (
21             `users_ID` INT NOT NULL AUTO_INCREMENT ,
22             `username` VARCHAR( 30 ) NOT NULL,
23             `password` VARCHAR( 50 ) NOT NULL,
24             `mail` VARCHAR( 50 ) NOT NULL,
25             `ime` VARCHAR( 30 ) NOT NULL ,
26             `prezime` VARCHAR( 30 ) NOT NULL ,
27             `tel1` TEXT NOT NULL ,
28             `tel2` TEXT NOT NULL ,
29             `adresa` TEXT NOT NULL ,
30             `br_dozv_oglasa` INT NOT NULL DEFAULT '5' ,
31             `grad` TEXT NOT NULL ,
32             `admin` INT NOT NULL ,
33             `limit2` INT NOT NULL DEFAULT '20',
34             PRIMARY KEY ( `users_ID` )
35             ) TYPE = MYISAM ;" ) or die(mysql_error());
36
37 echo "<BR>";
38 echo "Uspesno kreirana tabela users";
39
40 // ..... Kreiranje ostalih tabela na isti način kao i tabele "users"
```

PHP kod za prihvatanje podataka i HTML forme kao i za kreiranje tabela u bazi

Objašnjenje po linijama koda :

- 1,2,3 – Definisanje novih promenljivih kojima se dodeljuju vrednosti iz HTML forme
- 6 – Proveravamo da li je vrednost promenljive „action“ jednaka sa „install“ , da li je unesena vrednost za korisničko ime, podudaraju li se unesene lozinke i ako su svi uslovi ispunjeni prelazimo na dalje izvršavanje koda , u suprotnom ne dešava se ništa i očekuje se novi unos podataka.
- 10 – konekcija na MySQL bazu sa određenim parametrima
- 12 – ispis info poruke
- 20 – kreiranje tabele sa nazivom „users“ koja sadrži sledeće kolone : users_ID , username, password, mail, ime, prezime, tel1, tel2, adresa, br_dozv_oglasa, grad, admin, limit2

Postupak iz linije dvadeset se ponavlja jos pet puta za kreiranje ostalih tabela potrebnih za rad web aplikacije.

Ako sve do sada prođe bez grešaka , na kraju nam ostaje još da u tek kreirane tabele (u ovom slučaju će to biti tabela „users“) unesemo podatke koje je korisnik uneo preko HTML forme. Sledeći listing pokazuje kod koji ovo izvršava.

```
1  mysql_query("INSERT INTO users(users_ID,
                                username,
                                password,
                                mail,
                                ime,
                                prezime,
                                tel1,
                                tel2,
                                adresa,
                                br_dozv_oglasa,
                                grad,
                                admin)
2  VALUES(NULL, ".$admin_username_buff.",
            ".md5($admin_password_buff).",
            'localhost@localhost',
            'Administrator',
            'Administrator',
            '127.0.0.1',
            '127.0.0.1',
            'localhost',
            50 ,
            'Internet',1) ") or die(mysql_error());
3  // mysql_close();

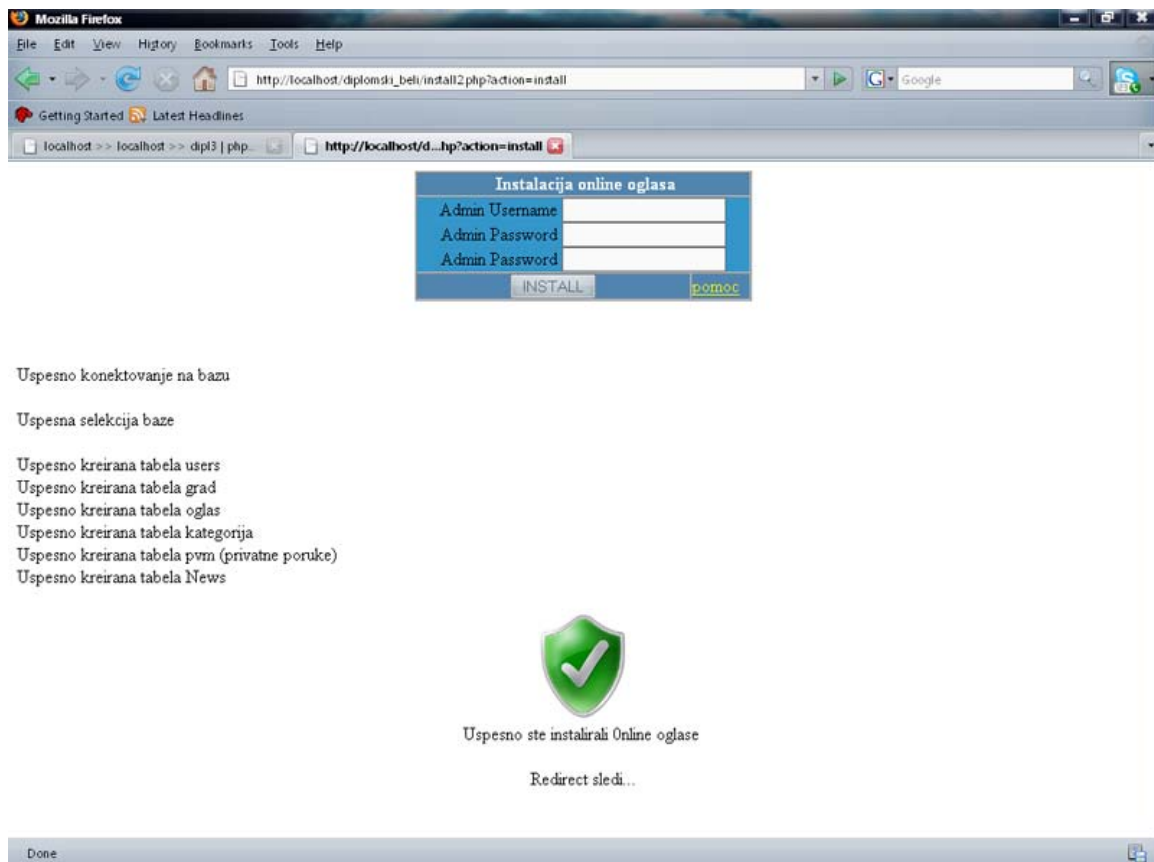
4  mkdir("./usere/".$admin_username_buff, 0777, true);
5  echo '<center> <br> <br>Uspesno ste instalirali Online oglase
6  <br><br> Redirect sledi... </center>';
7  echo '<META HTTP-EQUIV="Refresh" CONTENT="3; URL=index.php">';
```

PHP kod za unos podataka u tabelu users u bazi podataka

Objašnjenje po linijama koda:

- 1 – izvršavamo upit (query) INSERT INTO koji nam govori da unosimo podatke u tabelu users određenim redosledom ...
- 2 – postavljamo vrednosti koje će da se upšu u bazu podataka .Treba obratiti pažnju na redosled podataka kojimora biti identičan kao što je navedeno u liniji 1.
- 5 – ispisujemo info poruku sa propratnom slikom
- 7 – automatski redirekt ka index.php stranici posle tri sekunde.

Posle uspešne instalacije uz pomoć install.php fajla web browser bi trebao da izgleda kao na sledećoj slici :



Slika12 :Uspešno izvršena instalacija web aplikacije

8.2 REGISTRACIJA NOVOG KORISNIKA

Radi sprečavanja nekontrolisane predaje oglasa, što bi tokom vremena dovelo do nagomilavanja nepotrebnih stvari a kasnije možda čak i do prestank funkcionisanja web aplikacije, odlučio sam da uvedem registraciju korisnika. Na ovaj način možemo da limitiramo određene korisnike tako što ćemo im odrediti maksimalan broj oglasa koje mogu da objavljuju u istom vremenskom intervalu. Takođe u slučaju da pojedini korisnik se ne pridržava pravila o postavljanju oglasa na web sajtu, jednostavnom akcijom administratora korisnik može da bude izbrisan kao i svi njegovi trenutni online oglasi.

Registracija novog korisnika treba da bude jednostavna, brza i da ne predstavlja veliki problem korisnicima jer u suprotnom može da dovede do neželjenih posledica kao na primer da odvrati korisnika od upotreba aplikacije.

O HTML formama je bilo već reči i mislim da nema potrebe da ponovo objašnjavam iste. Na sledećoj slici je prikazana forma za registraciju novog korisnika sa koje se jasno vide šta se zahteva za uspešnu registraciju.



Polja oznacena sa * su obavezna

Username*	
Password*	
Password*	
E-mail*	
Ime.....	
Prezime.....	
Adresa.....	
Telefon 1* ...[samo brojevi]..	
Telefon 2.....	
Grad.....	Beograd ☒

Slika13 HTML forma za registraciju novog korisnika

Možda treba napomenuti to da se u padajućem meniju za izbor grada nalaze svi gradovi koje je administrator naveo u CMS-u.

Pozivajući „login.php?act=add_user“ fajl (gde promenljivoj „act“ dajemo vrednost „add_user“) preko login.php fajla pokrećemo funkciju html_add_user() koja je definisana u html.php fajlu i služi za prikaz gore navedene forme za registraciju novog korisnika.

Posle unosa potrebnih podataka za registraciju pritiskom na dugme „Registruj me“ saljemo unešene podatke prema fajlu add_user2.php gde će se izvršiti provera i kasnije unos podataka u bazu.

```

1 function registruj() {
2     $add_username_buff = $_POST['add_username'];
3     $add_password_buff = md5($_POST['add_password']);
4     $add_password_buff2 = md5($_POST['add_password2']);
5     $add_email_buff = $_POST['add_mail'];
6
7     $add_ime_buff = $_POST['add_ime'];
8     $add_prezime_buff = $_POST['add_prezime'];
9     $add_gradd_buff = $_POST['add_gradd'];
10    $add_adresa_buff = $_POST['add_adresa'];
11    $add_telefon1_buff = $_POST['add_telefon1'];
12    $add_telefon2_buff = $_POST['add_telefon2'];
13
14    $num1 = chek_username($add_username_buff);
15    $num2 = chek_password($add_password_buff, $add_password_buff2);
16    $num3 = chek_email($add_email_buff);
17    $num4 = chek_tel($add_telefon1_buff);
18
19    if($num1 + $num2 + $num3 + $num4 == 0)
20    {
21        mysql_query("INSERT INTO users(users_ID, username, password,
22                                mail, ime, prezime, tel1, tel2, adresa,
23                                br_dozv_oglasa, grad, admin)
24                                VALUES(NULL, '', $add_username_buff, '',
25                                '$add_password_buff',
26                                '$add_email_buff',
27                                '$add_ime_buff',
28                                '$add_prezime_buff',
29                                '$add_telefon1_buff',
30                                '$add_telefon1_buff',
31                                '$add_adresa_buff',
32                                5,
33                                '$add_gradd_buff', 0) ") or die(mysql_error());
34        // mysql_close();
35    }
36    echo '<center> <br> <br>Uspesno ste se registrovali <br><br>
37    Redirect sledi... </center>';
38    echo '<META HTTP-EQUIV="Refresh" CONTENT="3; URL=login.php?act=login">';
39    }
40    else
41    {
42        echo '<center> <br>  <br>
43        Greska pri registraciji ....Proverite unos podataka !!!!! <br>';
44        echo "<BR>";
45        if($num1==1) echo 'Proverite <b>username</b><br>';
46        if($num2==1) echo 'Proverite <b>password</b><br>';

```

```

24 if($num3==1) echo 'Proverite <b>e-mail</b><br>';
25 if($num4==1) echo 'Proverite <b>telefon 1</b><br>';

27 echo '<br>Pokusajte ponovo <a href="login.php?act=add_user">Login</a></p></center>';
    }
    }

```

Objašnjenje po linijama koda (funkcija registruj()) :

- 1 – definisanje funkcije registruj()
- 2-11 – definisanje varijabli sa vrednostima iz HTML forme za registraciju korisnika
- 12-15 – pozivanje funkcija za proveru validnosti pojedinih podataka kao na primer : korisnicko ime tj. username (da li već postoji u bazi neki drugi korisnik sa istim korisničkim imenom) , lozinka (da li se obe unete lozinke podudaraju) , e-mail adresa (da li je validnog formata) kao i broj telefona.
- 16 – ako se provera podataka sprovede bez greške prelazimo na unos podataka u bazu.
- 17 – unos podataka u bazu
- 18 – ispis info poruke
- 19 – automatski redirekt ka stranici za login
- 20 – u suprotno da provera podataka prođe neuspešno
- 21 – ispis info poruke
- 22-25 – pomoć pri ukazivanju šta je pogrešno uneseno
- 27 – link ka ponovnom pokušaju registracije novog korisnika.

NAPOMENA : lozinka se u bazi čuva hashovana uz pomoć MD5 algoritma što znači da je više nemožemo vratiti u prvobitni. PHP nam omogućava veoma brz način korišćenja MD5 algoritma.

```

3 $add_password_buff = md5($_POST['add_password']);
4 $add_password_buff2 = md5($_POST['add_password2']);

```



Uspesno ste se registrovali

Redirect sledi...



Greska pri registraciji Proverite unos podataka !!!!!

Proverite **password**

Pokusajte ponovo [Login](#)

Poruka posle uspešne registracije

Poruka posle neuspele registracije

8.3 LOGIN

Kada smo se uspešno registrovali sada ostaje još da se prijavimo na sajt unosom registrovanog korisničkog imena i lozinke.

Izborom „Login“ u meniju browser će nam otvoriti login stranicu koja izgleda kao na slici ispod :



Izgled login stranice

Pored HTML forme za unos korisničkog imena i lozinke na ovoj stranici je i link ka registraciji novog korisnika. Ovde možemo primetiti i jednu info poruku koja nas obaveštava da moramo omogućiti rad cookie-a da bi smo se mogli uspešno registrovati. U ovom projektu sam se odlučio samo za cookie kako zbog jednostavnosti izrade tako i objašnjenja rada istih.

NAPOMENA : tehnika koja je zastupljena u ovom projektu a koja se odnosi na upotrebu samo cookie-a nije baš najbolji izbor. Takođe i upotreba cooke-a na način koji je prikazan ovde nije najbolje rešenje jer ovde u kolačićima čuvamo čisto korisničko ime i na taj način saj prepoznaje korisnika. U drugim slučajevima , kada sajt ide direktno online upotreba kolačića je malo drugačija : u kolačićima se čuva ili hashovana lozinka i korisničko ime , ili hashovana random reč koja je u bazi zabeležena za svakog korisnika. Postoji mnogo načina za prepoznavanje korisnika među kojima je i upotreba sesija (session) ali sam se ja odlučio samo za cookie (kolačiće) i to na najjednostavniji način da bi što jasnije prikazao njihovo funkcionisanje i upotrebu istih.

Nakon unosa traženih podataka , HTML forma ih prosleđuje funkcija koja se zove „user_chek()“ koja je definisana u login_chek.php fajlu. Kod ove funkcije izgleda ovako :

```

1 function user_chek() {
2   if($_COOKIE['teste']==='') {

       echo '<p align="center">da bi se uspesno logovali morate omoguciti rad COOKI-a !!! </p>';
       }
       else
       {

3   mysql_konekcija();

4   $user_buff = $_POST['username'];
5   $pass_buff = md5($_POST['password']);

6   $result = mysql_query("SELECT password FROM users WHERE username LIKE '". $user_buff ."'");

7   if(!$result || (mysql_numrows($result) < 1)){
       echo '<p align="center" ><br><br>';
8     echo "Pogresan username ili password";
       echo "<BR>";
9     echo '<a href="login.php?act=login">Pokusajte ponovo</a></p>';
       }
       else {
10    while($row = mysql_fetch_array( $result )) {

           if($pass_buff == $row['password']) {
11      setcookie("usere" , "".$user_buff."");

           echo '<p align ="center"> <br>  <br> Uspesno ste logovani !
</p>';
           echo "<BR>";
12      echo '<META HTTP-EQUIV="Refresh" CONTENT="5; URL=index.php">';
           echo '<p align="center">Ukoliko vas browser ne redirectuje automatski klikni te
           <a href="index.php">ovde</a></p>';
           }

       } } } }

```

Prikaz PHP koda funkcije user_chek() koja se koristi za proveru identiteta korisnika

Objašnjenje po linijama koda :

- 1 – definisanje funkcije user_chek()
- 2 – proveravamo postojanje cookie-a „test“ koji je trebao biti kreiran odmah po učitavanju stranice i u slučaju da ne postoji ispisuje se obaveštenje da se upotreba cookie-a omogući od strane korisnika
- 3 – poziv funkcije za konekciju na bazu podataka
- 4-5 – preuzimanje podataka iz HTML forme
- 6 – provera postojanja korisnika u bazi
- 7,8,9 – ako korisnik nije pronađen u bazi spisuje se info poruka
- 11 – pošto je korisnik pronađen u bazi kreira se cookie (kolač) „usere“ koji sadrži username (korisničko ime) logovanog korisnika. Na osnovu ovog kolačića web aplikacija će u daljem radu tačno znati o kom se korisniku radi.

8.4 PREGLED OGLASA

Pregled oglasa je dozvoljen i ne registrovanim korisnicima. Ovo je jedan od važnijih delova web sajta jer većina korisnika će najvećim delom koristiti ovu funkciju.

Pre svega ovde posebno treba da se obrati pažnja na sam dizajn odnosno na preglednost sadržaja oglasa. Kako je važna preglednost tako je važno i da se sve potrebne informacije nađu u tom oglasu, kako sam tekst oglasa tako i ko ga je postavio, broj telefona, cena ...

Pvoro što se uoči kada se otvori stranica za pregled oglasa jeste forma za izbor grada i kategorije.

Grad	Beograd ▾
Kategorija	Racunari ▾
Prikazi	

forma za izbor grada i kategorije

Po izboru željenog grada i kategorije, pritiskom na dugme „prikaz“ izlistaće se svi oglasi koji ispunjavaju traženi kriterijum. Oglasi su sortirani tako da je prvi (najgornji) oglas najnoviji.

Na slici su prikazani oglasi za kriterijum grad : Beograd i kategorija : Racunari

Grad	Beograd ▾
Kategorija	Racunari ▾
Prikazi	

Grad : Beograd----**Kategorija :** Racunari

Strinice : 1 |

Oglas predao : admin	18. 6. 2007	
prodajem pentium4.64 bit ploca asus,procesor 2800 32 bit,1gb ram memorija,512 mb grafika gforce,hard disk 120gb,dvd rezac,dvd combo,jako malo koristeno original racun.uz punu cenu dajem garanciju 1 god.		: 024111111 : admin@localhost.com : 400e
Oglas predao : admin	18. 6. 2007	
Prodajem laptop benq , 1.6ghz , 512mb ddr2, 60gb hard , grafika do 128 , wairles, bluetoo ...		: 024111111 : admin@localhost.com : 500 eura
Oglas predao : admin	18. 6. 2007	
Prodajem Samsung CRT monitor 17" , dobro ocuvan , slika zadovoljavajua ... vredi pogledati		: 024111111 : admin@localhost.com : 35 eura

Strinice : 1 |

Ono što treba da se primeti jeste da svaki oglas poseduje identične informacije i to sledeće :

- Ko je predao oglas
- Datum predaje oglasa
- Link za slanje privatne poruke korisniku koji je autor oglasa
- Tekst oglasa
- Telefon autora
- E-mail adresa autora
- Opciono cena

NAPOMENA : oglasi se prikazuju u tabelama. Potrebno je samo definisati osnovnu tabelu kojoj će se prikazivati tekst oglasa i informacije vezane za njega. Kasnije se samo uz pomoć WHILE petlje čitaju oglasi iz baze i pravi se onoliko tabela koliko i postoji oglasa u bazi koji zadovoljavaju birani kriterijum.

Da vidimo kod za prikaz oglasa :

```

1 $result4 = mysql_query("SELECT * FROM oglas WHERE kategorija LIKE ".$pregled_kat_buff."
  AND grad LIKE ".$pregled_grad_buff." ORDER BY oglasi_ID DESC LIMIT $set_limit, $limit");

2 while($row4 = mysql_fetch_array($result4)) {

3 echo'<table border="0" width="80%" cellpadding="0" cellspacing="0" align="center" >
  <tr>
    <td width="35%" class="oglas_header">Oglas predao :&nbsp;<b>'. $row4['user'] .'</b></td>
    <td width="32%" class="oglas_header">&nbsp;<b>'. $row4['datum'] .'</td>
    <td width="33%" class="oglas_header"><p align="center" >
      <a href="pvm.php?op=add_pvm&za_user='.$row4['user'].'"></a></p>
    </td>
  </tr>
  <tr>
4 <td width="67%" colspan="2" class="oglas_text"><p class="oglas_text">'. $row4['oglas'] .'</p></td>
    <td width="33%" class="oglas_info" ><p align="right"> : <font
    color="#FFFFFF">'. $row4['tel1'] .' <BR>  : </img>'. $row4['mail'] .'
    <BR> : </img>'. $row4['cena'] .'&nbsp;</font> </p> </td>
  </tr>
  <tr>
    <td class="oglas_header" ><p>&nbsp;</p></td>
    <td class="oglas_header" ><p>&nbsp;</p></td>
    <td class="oglas_header" ><p>&nbsp;</p></td>
  </tr>
</table>';

}

```

Objašnjenje po linijama koda :

- 1 – Izvršavanje upita nad bazom gde se biraju svi oglasi koji odgovaraju zadatom kriterijumu odnosno pripadaju biranom gradu i kategoriji
- 2 – WHILE petlja koja nam omogućava čitanje svih oglasa koji zadovoljavaju zadati kriterijum.
- 3 – kreiranje tabele za prikaz oglasa
- 4 – kreiranje polja u tabeli gde će se prikazati tekst oglasa. Ovde je interesantno i to da se za ovo polje pozivamo na klasu „glas_text“ koja određuje grafički dizajn samog polja i koja je definisana u „design.css“ fajlu.

NAPOMENA : kako je kreirano polje za tekst oglasa tako se kreiraju i polja za ostale informacije vezane za taj oglas.

8.5 PREDAJA OGLASA

Još jedan od bitnijih delova web sajta jeste predaja oglasa. Za razliku od pregleda oglasa koji je dozvoljen svim posetiocima sajta , predaja oglasa je ograničena samo na registrovane korisnike. Iz menija koji je dozvoljen samo registrovanim korisnicima izborom linka „Predaj oglas“ dolazimo do stranice koja sadrži formu za unos novog oglasa. Ta forma je prikazana na sledećoj slici :



Izaberi kategoriju :

Oglas :

Grad : Preostalo karaktera :

Cena :

HTML forma za predaju oglasa

Kao što se vidi na slici , od korisnika se traži da izabere kategoriju iz ponuđene liste, zatim da unese tekst oglasa. Dužina oglasa je ograničena na tri stotine (300) karaktera i ova funkcija je izvršena uz pomoć jednostavne JAVA skripte , koju neću detaljno objašnjavati jer JAVA nije tema ovoga rada. Posle unosa teksta potrebno je izabrati grad za koji će oglas važiti , opciono uneti cenu i pritiskom na dugme „Dodaj oglas“ proslediti ove podatke PHP funkciji koja će ih obraditi i pohraniti u bazu podataka.

Funkcija za predaju oglasa (pohranjivanje podataka u bazu) naziva se „predaja()“ i definisana je u fajlu predaja_add.php

Prihvatanje podataka iz HTML forme vršimo na sledeći način :

```
$predaja_datum_buff = $_POST['predaja_datum'];
$predaja_kat_buff   = $_POST['predaja_kat'];
$predaja_oglas_buff = $_POST['predaja_oglas'];
$predaja_grad_buff  = $_POST['predaja_grad'];
$predaja_cena_buff  = $_POST['predaja_cena'];
$predaja_user_buff  = $_POST['predaja_user'];
```

Provera validnosti podataka :

```
if ($predaja_datum_buff == "" ||
    $predaja_kat_buff   == "" ||
    $predaja_oglas_buff == "" ||
    $predaja_grad_buff  == "" ||
    $predaja_user_buff  == "" )
{
    echo '<p align="center"><br> <br>Oglas nije predat !
    <br><br>
    Uneseni podaci za oglas nisu validni , molimo vas ponovite unos.
    </p>';
}
```

Dobijanje podataka o korisniku koji predaje oglas

```
$resultt = mysql_query("SELECT * FROM users WHERE username LIKE ' ".$predaja_user_buff."") or
die(mysql_error());

while($roww = mysql_fetch_array($resultt)) {
    $broj_dozvoljenih_oglasa = $roww['br_dozv_oglasa'];
    $predaja_tel1_buff = $roww['tel1'];
    $predaja_tel2_buff = $roww['tel2'];
    $predaja_mail_buff = $roww['mail'];
}
```

Provera da li korisnik ima pravo da preda oglas :

```
if($broj_dozvoljenih_oglasa > 0) {
```

Unošenje podatak u bazu :

```
mysql_query("INSERT INTO oglas(kategorija , oglas , grad , cena, user, datum, tel1, tel2, mail )
VALUES('".$predaja_kat_buff."',
                                           ".$predaja_oglas_buff."',
                                           ".$predaja_grad_buff."',
                                           ".$predaja_cena_buff."',
                                           ".$predaja_user_buff."',
                                           ".$predaja_datum_buff."',
                                           ".$predaja_tel1_buff."',
                                           ".$predaja_tel2_buff."',
                                           ".$predaja_mail_buff.'"
                                           ) ") or die(mysql_error());
```

Umanjivanje broja dozvoljenih oglasa za korisnika i izmena tog podatka u bazi :

```
$broj_dozvoljenih_oglasa--;

mysql_query("UPDATE users SET br_dozv_oglasa = '".$broj_dozvoljenih_oglasa.'" WHERE username
= '".$predaja_user_buff.'"");
```

Ispis info poruke :

```
echo '<p align="center">Uspesano ste predali oglas</p>';
echo '<p align="center"><BR><BR>Automatski redirekt za par sekundi !</p>';
echo '<META HTTP-EQUIV="Refresh" CONTENT="5; URL=pregled.php">';
}
else
{
echo '<p align="center"><br> Nimate dovoljno kredita da predate
oglas </p>';

}
```

Predaja oglasa je veoma laka , i brza. Pažnju sam obratio na to da se korisnik niti u jednom trenutku nađe u takvoj situaciji da ne zna šta dalje treba da se radi. Sve je praćeno adekvatnim objašnjenjem i mislim da bilo koji korisnik neovisno o njegovom znanju upotrebe interneta može bez ikakvih problema brzo i jednostavno da preda svoj online oglas.

8.6 PRETRAGA OGLASA

Kako bi vreme prolazilo , rastom broja registrovanih korisnika na web sajtu bi doprineo i povećanje broja online oglasa. Samim tim pregled oglasa bi bio sve teži i teži , kasnije možda čak i nemoguć, pa zbog toga je gotov i neizostavno omogućiti pretragu predatih oglasa.

Pretragu oglasa sam onemogućio ne registrovanim korisnicima da bi ih možda baš to i navelo na registraciju koja bi poboljšala statistiku samog web sajta. Registrovani korisnici izborom „Pretraga“ linka iz menija dolaze do stranice za pretragu oglasa. Kako bi se držali pravila ovoga projekta , da sve bude pojednostavljeno do krajnjih granica, tako je i pretraga osiromašena i kao takva omogućava samo osnovnu vrstu pretrage oglasa po gradovima , kategorijama i zadatim rečima.

Na slici ispod je prikazana forma za pretragu online oglasas :



Izborom željenog grada i kategorije kao i reči koju želimo da traženi oglas sadrži pritiskom na dugme „Traži“ PHP funkcija „pretraga_pregled()“ koja je definisana u fajlu pretraga.php vrši pretaživanje baze podataka za oglasima koji odgovaraju traženom kriterijumu.

Sledeći kod prikazuje način pretrage baze podataka po kriterijum grada , kategorije i tražene reči :

```
$result4 = mysql_query("SELECT * FROM oglas WHERE kategorija LIKE ' ".$pregled_kat_buff." ' AND grad LIKE ' ".$pregled_grad_buff." ' AND oglas LIKE '%$trazena_rec%' ORDER BY oglasi_ID DESC ");
```

Posle završene pretrage , pronađeni oglasi se izlistavaju na identičan način kao što je to objašnjeno u naslovu „PREGLED OGLASA“.

8.7 PODEŠAVANJA

Kada već postoji registracija korisnika i pamćenje nekih osnovnih podataka za njih potrebno je omogućiti i njihovu izmenu u slučaju da se neki podaci vezani za pojedinog korisnika promene.

Svaki registrovani korisnik odlaskom na „Podesavanja“ strnicu ima mogućnost da sam menja svoje podatke. Ono što je bitno, jesto to da obavezno korisniku omogućimo da pre promene vidi svoje trenutne podatke. Korisnik može da menja sve one podatke koje je pri registraciji uneo kao i limit koliko oglasa želi da mu se izlista istovremeno pri pregledu. Takođe je omogućeno da promeni i svoju lozinku.

Slika ispod prikazuje HTML formu za podešavanja registrovanog korisnika.



E-mail*	admin@localhost.com
Ime.....	
Prezime.....	
Adresa.....	
Telefon 1*.....	024111111
Telefon 2.....	024111111

Limit oglasa po stranici...  Vas trenutni limit je :20

Nova lozinka...

Obavezno ostavite prazno ako ne želite da menjate lozinku !!!!

Klikom na dugme „Sacuvaj“ podaci se prosleđuju funkciji „podesavanja2()“ koja je definisana u podesavanja2.php fajlu.

NAPOMENA : mislim da opis koda ove funkcije nije potreban jer je tehnika pohranjivanja podataka u bazu već više puta objašnjavana.

8.8 BRISANJE OGLASA

Pored predaje oglasa potrebo je omogućiti i brisanje istih. Ovo je zamisljeno tako da svaki korisnik ima pravo da briše svoje oglase. Ako se prisetimo pravila da je korisnik limitiran sa brojem oglasa koje može u isto vreme da objavljuje onda je ovo itekako interesantna funkcija. Brisanjem oglasa korisnik pravi sebi prostora za sledeći noviji oglas.

Brisanje je zamišljeno krajnje jednostavno. Izborom linka „Moji oglasi“ iz menija korisniku će se izlistati svi njegovi aktuelni oglasi. Klikom na ikonicu (kanta za smeće) ispod određenog oglasa on prosleđuje ID_oglasa funkciji koja automatski briše taj oglas.



Oglasi koje je predao : **admin**

admin	18. 6. 2007	024111111
Pentium 4.64 bit ploca asus, procesor 2800 32 bit, 1gb ram memorija, 512 mb nvidia geforce, hard disk 120gb, dvd rezac, dvd combo, jako malo koristeno, ucun, uz punu cenu dajem garanciju 1 god.		
		Slike
admin	18. 6. 2007	024111111
Laptop benq , 1.6ghz , 512mb ddr2, 60gb hard , grafika do 128 , wairles, 100e		
		Slike
admin	18. 6. 2007	024111111
Samsung CRT monitor 17" , dobro ocuvan , slika zadovoljavajua ... vred		
		Slike

Listanje ličnih oglasa i njihovo brisanje.

Izlistavanje oglasa za određenog korisnika je isto kao i izlistavanje oglasa po kriterijumu grada i kategorije samo je ovde slučaj da je kriterijum korisničko ime (username) korisnika.

U tabeli za prikaz oglasa sada je potrebno da napravimo jedno dodatno polje u koje ćemo smestiti ikonicu da brisanje sa adekvatnom URL adresom koja će nam omogućiti uspešno brisanje željenog oglasa.

```
<td width="32%" bgcolor="#E2E2E2" align="center">
  <a href="user_cp_func.php?op=brisanje_oglasa&op2='.$row_cp['oglas_ID'].'" >
  </img>
</a></td>
```

Kod za prikazivanje ikonice da brisanje oglasa sa adekvatnom adresom

U gore prikazanom kodu jedna od bitnijih stvari jeste kreiranje URL adrese koja će da pozove određeni php fajl , prenese vrednosti promenljivih koje nam govore šta treba da se radi i nad kojim elementom. U ovom slučaju to je fajl user_cp_func.php , promenljiva „op“ čija je vrednost „brisanje_oglasa“ kao i promenljiva „op2“ koja za vrednost ima ID_oglasa koji treba da se briše.

Sledi prikaz tog dela koda :

```
<a href="user_cp_func.php?op=brisanje_oglasa&op2='.$row_cp['oglas_ID'].'">
```

Kreiranje URL adrese za brisanje određenog oglasa

Prosleđivanjem promenljive „op“ čija je vrednost „brisanje_oglasa“ prema fajlu „user_cp_func.php“ poziva se funkcija „brisanje_oglasa(\$br_oglasa)“ koja je definisana u istoimenom php fajlu. Ono što treba da se primeti jeste da ova funkcija očekuje neku vrednost koju će da smesti u promenljivu „br_oglasa“ i koju će upotrebiti u kasnijem izvršavanju koda.

Sledeći listing prikazuje kod „brisanje_oglasa(\$br_oglasa)“ funkcije :

```
function brisanje_oglasa($br_oglasa) {

    $result_cp = mysql_query("SELECT user FROM oglas WHERE oglas_ID LIKE ' ".$br_oglasa."");

    while($row_cp = mysql_fetch_array( $result_cp )) {
        $buff_user = $row_cp['user'];
    }
    if($buff_user==$_COOKIE['usere']){

        mysql_query("DELETE FROM oglas WHERE oglas_ID = $br_oglasa");

        $br_doza_oglasa;

        $result_cp = mysql_query("SELECT br_doza_oglasa FROM users WHERE username LIKE ' ".$_COOKIE['usere']."");
        while($row_cp = mysql_fetch_array( $result_cp )) { $br_doza_oglasa = $row_cp['br_doza_oglasa']; }
        $br_doza_oglasa++;

        mysql_query("UPDATE users SET br_doza_oglasa = ' ".$br_doza_oglasa." WHERE username = ' ".$_COOKIE['usere']."");

        echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=user_cp_func.php?op=prikazi_oglase">';
    }
}
```

Funkcija „brisanje_oglasa(\$br_oglasa)“

8.9 ADMINISTRACIJA – CMS

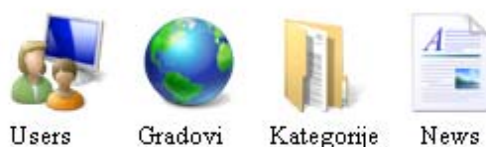
Web aplikacija kao što je i ova za internet online oglase je nezamisliva bez mogućnosti administracije sadržaja. Iz potrebe za četim izmenama , dopunama , brisanjima obavezno se pravi CMS (Content management system) mehanizam koji nam omogućava sve ove radnje.

Ovo jedan od vrlo bitnih delova aplikacije i često predstavlja i najveći zalogaj za programera.

CMS za aplikaciju koja je tema ovog rada je takođe pojednostavljen kako zbog izrade tako i zbog shvatanja istog.

Planiranjem došao sam do zaključka da su mi neophodne sledeće stvari :

- Kontrola nad korisnicima
- Dodavanje – brisanje gradova
- Dodavanje – brisanje kategorija
- Pisanje – izmena – brisanje vesti koje se nalaze na index stranici.



Izgled CMS menija

8.9.1 USERS

U ovom delu se vrši kontrola nad registrovanim korisnicima : Administratorima je dozvoljen pregled svih korisnika, kao i njihovih osnovnih podataka. Takođe je moguće izvršiti brisanje korisnika kao i dodavanje odnosno oduzimanje broja dozvoljenih oglasa .



Username	Email	Ime	Prezime	Grad	O	admin	Obrisi	++	--
admin	admin@localhost.com				2	1	Obrisi	++	--
darko	darko@oglasi.com	Darko	Dragic	Beograd	5	0	Obrisi	++	--

Administracija korisnika uz pomoć CMS-a

8.9.2 GRADOVI I KATEGORIJE

Zbog toga što se oglasi sortiraju prema gradovima za koje važe kao i prema kategorijama kojim pripadaju, izostavljanje administracije gradova i kategorija je nezamisliva. Pomoću ovih funkcija mi kontrolišemo sadržaj baza podataka za gradove i kategorije iz kojih se podaci čitaju na više mesta tokom funkcionisanja aplikacije. Prisetimo se samo predaje oglasa , pregleda i pretrage oglasa , registracije korisnika , na svim ovim mestima imamo mogućnost izbora ili grada ili kategorije a negde oba ova kriterijuma.

Izmenom ovih podataka uz pomoć CMS-a automatski dolazi do izmene tih podataka u bilo kojim izbornim listama.

Sledeća slika pokazuje kako to izgleda :

 Users
  Gradovi
  Kategorije
  News

Beograd	Obrisi grad
Novi Sad	Obrisi grad
Subotica	Obrisi grad

Dodaj grad

Adminsitracija gradova uz pomoć CMS-a

Iz ove slike vidimo da je na sasvim jednostavan i jasan način dodavanja i brisanja određenog grada.

Slično je i za administraciju kategorija i to se može videti iz sledeće slike :

 Users
  Gradovi
  Kategorije
  News

Automobili	Obrisi kategoriju
Racunari	Obrisi kategoriju
Telefoni	Obrisi kategoriju

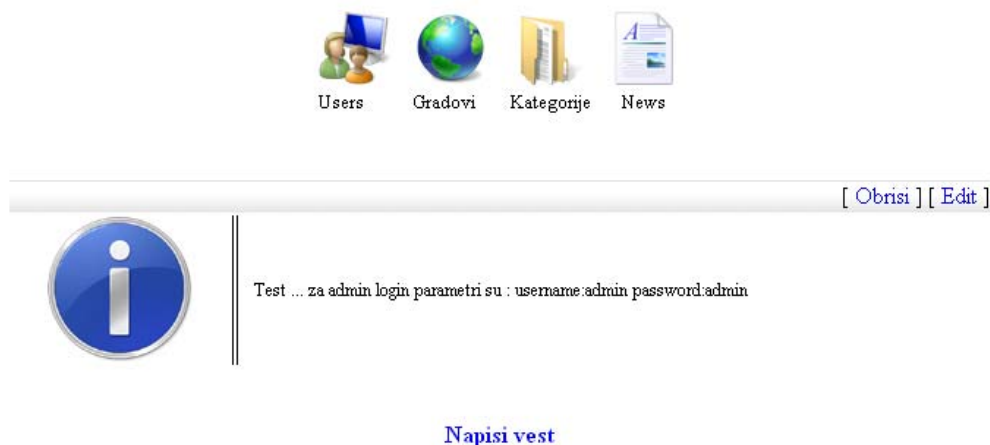
Dodaj kategoriju

Adminsitracija kategorija uz pomoć CMS-a

8.9.3 VESTI – NEWS

Iz potrebe za obaveštavanjem posetilaca sajta o novostima vezanim za njega kao i o nekim drugim obaveštenjima na početnu stranicu (index) postavio sam prikaz vesti (news). Radi lakšeg dodavanja , brisanja ili izmene vesti dodali smo administraciju vestu u CMS. Oдавde je na najjednostavniji način omogućeno izvršavanje svih gore spomenutih akcija.

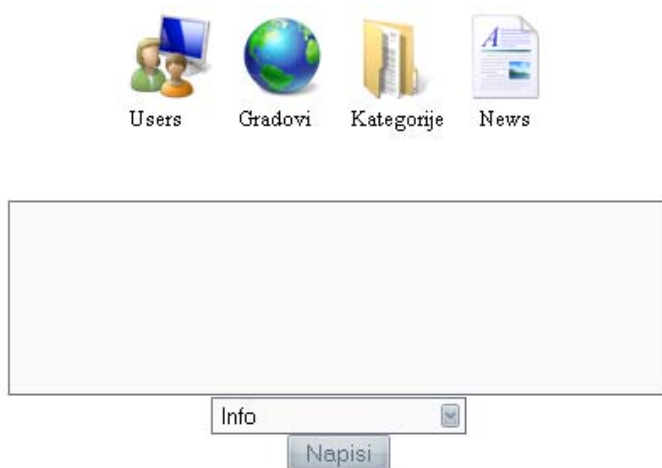
Sledeća slika pokazuje stranicu u CMS-u za administraciju vesti :



Administracija vesti uz pomoć CMS-a

Na slici vidimo da nam se prvo izlistavaju sve postojeće vesti nad kojim možemo da vršimo izmenu ili da ih obrišemo. Takođe na ovoj stranici se nalazi i link „Napisi vest“ koji nas vodi do stranice gde možemo napisati novu vest koja će biti prikazana na početnoj stranici.

Sledeća slika pokazuje stranicu u CMS-u za pisanje novih vesti :



Pisanje novih vesti uz pomoć CMS-a

8.10 CMS – LISTING KODA

Sledeći listing prikazuje fajl gazda.php gde se nalazi kod CMS-a :

```
<?php
include "inc.php";

//$op = $HTTP_GET_VARS[op];

if($_POST['add_news'] == "1")
{
    $komanda = "add_news";
    $_POST['add_news'] = "0";
}
else
{
    $komanda = $HTTP_GET_VARS[op];
    $komanda2 = $HTTP_GET_VARS[op2];
}

function dodaj_broj_oglasa($userr)
{
    $resultt = mysql_query("SELECT * FROM users WHERE username LIKE ''.$userr.'''") or
die(mysql_error());

    while($roww = mysql_fetch_array($resultt)) {
        $broj_dozvoljenih_oglasa = $roww['br_dozv_oglasa'];
    }

    $broj_dozvoljenih_oglasa++;

    mysql_query("UPDATE users SET br_dozv_oglasa = ''.$broj_dozvoljenih_oglasa.'" WHERE username
= ''.$userr.'''");

    echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=users">';
}

//-----

function oduzmi_broj_oglasa($userr1)
{
    $resultt = mysql_query("SELECT * FROM users WHERE username LIKE ''.$userr1.'''") or
die(mysql_error());

    while($roww = mysql_fetch_array($resultt)) {
        $broj_dozvoljenih_oglasa = $roww['br_dozv_oglasa'];
    }

    if($broj_dozvoljenih_oglasa > 0) {

        $broj_dozvoljenih_oglasa--;
        mysql_query("UPDATE users SET br_dozv_oglasa = ''.$broj_dozvoljenih_oglasa.'" WHERE username
```

```

= ".$userr1."");
    echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=users">';
}

}

//-----
// Brisanje usera i svih njegovih oglasa kao i privatnih poruka koje je on nekome i koje je neko njemu poslao
function brisi_usera($userr2){

    $result = mysql_query("SELECT * FROM users WHERE users_ID LIKE $userr2") or
die(mysql_error());
    while($row = mysql_fetch_array($result)) {
        $usern = $row['username'];
    }

    mysql_query("DELETE FROM oglas WHERE user LIKE ".$usern." ");
    mysql_query("DELETE FROM pvm WHERE za_user LIKE ".$usern." ");
    mysql_query("DELETE FROM pvm WHERE od_user LIKE ".$usern." ");

    mysql_query("DELETE FROM users WHERE users_ID = $userr2");
    //mysql_close();

    echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=users">';

}

//=====
function admin_users()
{

    echo '
        <table border="0" width="100%" bgcolor="#C0C0C0" cellpadding="0" cellspacing="0">
            <tr>
                <td align="center" width="14%">Username</td>
                <td align="center" width="24%">Email</td>
                <td align="center" width="14%">Ime</td>
                <td align="center" width="14%">Prezime</td>
                <td align="center" width="14%">Grad</td>
                <td align="center" width="20%">O</td>
                <td align="center" width="4%">admin</td>
                <td>Obrisi</td>
                <td>++</td>
                <td>--</td>
            </tr>
        </table>
    ';

    $result = mysql_query("SELECT * FROM users ORDER BY username ");

    while($row = mysql_fetch_array( $result ))
    {

        echo '

```

```

<table border="1" width="100%" cellpadding="0" cellspacing="0">
<tr>
  <td width="14%" bgcolor="#E2E2E2" class="oglas_header">'. $row['username'].</td>
  <td width="24%">>'. $row['mail'].</td>
  <td width="14%" bgcolor="#E2E2E2" class="oglas_header">'. $row['ime'].</td>
  <td width="14%">>'. $row['prezime'].</td>
  <td width="14%" bgcolor="#E2E2E2" class="oglas_header">'. $row['grad'].</td>
  <td width="20%">>'. $row['br_dozv_oglasa'].</td>
  <td width="4%" bgcolor="#E2E2E2" class="oglas_header">'. $row['admin'].</td>
  <td><a href="gazda.php?op=brisi_usera&op2='. $row['users_ID'].'">Obrisi</a> </td>
  <td><a href="gazda.php?op=dodaj_broj_oglasa&op2='. $row['username'].'">+</a></td>
  <td><a href="gazda.php?op=oduzmi_broj_oglasa&op2='. $row['username'].'">--</a></td>
</tr>
</table>
';
}
}
}
//-----

function admin_kategorije() {

    $result = mysql_query("SELECT * FROM kategorija ORDER BY kategorija ");

    while($row = mysql_fetch_array( $result ))
    {

        echo '
align="center">
<table border="1" width="50%" cellpadding="0" cellspacing="0">
<tr>
  <td width="30%" bgcolor="#E2E2E2" class="oglas_header"
>'. $row['kategorija'].</td>
  <td width="5%">><a
href="gazda.php?op=brisanje_kat&op2='. $row['kategorija_ID'].'">Obrisi kategoriju</a></td>
</table>';

    }

    echo "<BR>";
    echo '<p align="center">Dodaj kategoriju </p> ';
    echo '<form method="POST" action="admin_add.php">
<p align="center"><input type="text" name="add_kat" size="20"> <input
type="hidden" name="admin_op" value="kat"><input type="submit" value="Dodaj" name="B1"></p>
</p>';

    }

}

//-----

function admin_gradovi() {

    $result = mysql_query("SELECT * FROM grad ORDER BY grad ");

    while($row = mysql_fetch_array( $result ))
    {

```

```

        echo '
        <table border="1" width="50%" cellpadding="0" cellspacing="0"
align="center">
        <tr>

        <td width="30%" bgcolor="#E2E2E2"
class="oglas_header">'. $row['grad'].</td>

        <td width="30%"
        ><a
href="gazda.php?op=brisanje_grad&op2='.$row['grad_ID'].'">Obrisi grad</a></td>
        </table>';
    }

    echo "<BR>";
    echo '<p align="center">Dodaj grad </p> ';
    echo '<form method="POST" action="admin_add.php">
    <p align="center"><input type="text" name="add_grad" size="20"> <input
type="hidden" name="admin_op" value ="grad"><input type="submit" value="Dodaj" name="B1"></p>
    </form>';
    }

function obrisi_grad($br_grada){

mysql_query("DELETE FROM grad WHERE grad_ID = $br_grada");
echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=gradovi">';
}

function obrisi_kategoriju($br_kat){

    mysql_query("DELETE FROM kategorija WHERE kategorija_ID = $br_kat");
    echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=kategorije">';
}

function admin_news() {
    $result = mysql_query("SELECT * FROM news ORDER BY news_ID DESC");
    while($row = mysql_fetch_array($result)) {
        $news = $row['news'];
        $slika = $row['slika'];
        $news_id = $row['news_ID'];
        echo '
        <br>
        <table class="news" align="center">
        <tr>
        <td class="news_gore" colspan="2" align="right">[<a
href="gazda.php?op=obrisi_news&op2='.$news_id.'"> Obrisi </a>] [ <a
href="gazda.php?op=edit_news&op2='.$news_id.'"> Edit </a> ] </td>
        </tr>
        <tr>
        <td class="news_levo">;
        //-----SLIKA U NEWS
        echo '<center></center>';
        echo '</td>
        <td class="news_desno"><P class="news">'. $news.</P></td>
        </tr>
        </table>

```

```

';
}

echo '<br><p align="center"> <b><a href="gazda.php?op=napisi_vest">Napisi vest</a></b> </p>';
}

function obrisi_news($news_id) {
mysql_query("DELETE FROM news WHERE news_ID = $news_id");
echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=news">';
}

function napisi_vest() {
echo '
<center>
<form method="POST" action="gazda.php?op=add_news">

<p><textarea rows="6" name="news" cols="46"></textarea><br>';

echo '<select name="news_slika">';
//The path to the style directory
$dirpath = "./pix/news/";
$dh = opendir($dirpath);

while (false !== ($file = readdir($dh))) {
//Don't list subdirectories

if (!is_dir("$dirpath/$file")) {
//Truncate the file extension and capitalize the first letter
echo "<option value='$file'>" . htmlspecialchars(ucfirst(preg_replace('/\..*$/ ', $file))) . "</option>";
}
}

closedir($dh);
//Close Select
echo '</select><br>
<input type="hidden" value="1" name="add_news">
<input type="submit" value="Napisi" name="B1"></p>
</form>
</center>

';
}

function add_news(){
$news = $_POST['news'];
$slika = $_POST['news_slika'];
mysql_query("INSERT INTO news (news_ID, news , slika) VALUES (NULL , '$news.', '$slika.')")
or die(mysql_error());
echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=news">';
}

function edit_news($news_id) {
$result = mysql_query("SELECT * FROM news WHERE news_ID LIKE $news_id");
while($row = mysql_fetch_array($result)) {

```

```
$news = $row['news'];
}
echo' <center>
<form method="POST" action="gazda.php?op=edit_news2&op2='.$news_id.'">

<p><textarea rows="6" name="news" cols="46"> '.$news.' </textarea><br>';

echo'
<input type="submit" value="Napisi" name="B1"></p>
</form>
</center>

';
}

function edit_news2($news_id) {
$news = $_POST['news'];
mysql_query("UPDATE news SET news = '".$news.'" WHERE news_ID = $news_id");
echo '<META HTTP-EQUIV="Refresh" CONTENT="0; URL=gazda.php?op=news">';
}

function gazda(){
global $komanda;
global $komanda2;
html_admin_meni();

switch ($komanda)
{
    case "users":
        admin_users();
        break;

    case "kategorije":
        admin_kategorije();
        break;

    case "gradovi":
        admin_gradovi();
        break;

    case "brisanje_grada":
        obrisi_grad($komanda2);
        break;

    case "brisanje_kat":
        obrisi_kategoriju($komanda2);
        break;

    case "brisi_usera":
        brisi_usera($komanda2);
        break;

    case "dodaj_broj_oglasi":
        dodaj_broj_oglasi($komanda2);
```

```
        break;

        case "oduzmi_broj_oglasa":
            oduzmi_broj_oglasa($komanda2);
            break;

        case "news":
            admin_news();
            break;

        case "obrisi_news":
            obrisi_news($komanda2);
            break;

        case "napisi_vest":
            napisi_vest();
            break;

        case "add_news":
            add_news();
            break;

        case "edit_news":
            edit_news($komanda2);
            break;

        case "edit_news2":
            edit_news2($komanda2);
            break;
    }

}

html(gazda);

?>
```

9 ZAKLJUČAK

HTML je jezik osmišljen radi prezentacije sadržaja. Svako ko je imao nešto da izloži, tražio je univerzalan i svuda razumljiv jezik, a to je upravo ono što je HTML nudio. Kao što svako voli da njegova knjiga bude štampana sa prijatnim pismom za čitanje, kvalitetnim slikama i dobrim kombinacijama boja i nijansi, tako i svaki autor HTML dokumenta želi da njegova stranica ima i umetnički dojam. Pored prava svakog autora da svojim čitaocima ponudi svoj sadržaj sa umetničkim karakteristikama, postoji i obaveza da on svakom čitaocu pruži mogućnost pregleda njegovog sadržaja.

Kako je CCS danas definisani standard možemo biti sigurni da će svi korisnici novije generacije browsera moći korektno da pregledaju naše stranice. To je jedna od retkih stavki oko kojih su se Microsoft i Netscape složili. Sve novije verzije programa za izradu HTML stranica podržavaju CSS. Iako je podržan samo u novijoj generaciji browsera, to ne predstavlja veliki problem jer su oni danas najčešće korišćeni. Kad se uzme u obzir da se pri izradi stranica ne mora voditi računa o kompatibilnosti sa browserom, i odličnu implementaciju u popularnim HTML editorima, onda CSS svako spadaju u stvari koje treba probati ali i koristiti.

PHP programski jezik je jedan od najboljih izbora ako trebate izradu web aplikacije ili nekog web-orijentisanog sistema. Uči se vrlo brzo, jednostavan je za primenu, jedini nedostatak je taj što ne postoje neki ozbiljniji alati za izradu php stranica. Inače, sve se svodi na notepad, a konkurencija (Microsoft ASP.NET, JSP, ...) ima puno bolju razvojnu okolinu. Naravno, još jedna od velikih prednosti PHP-a pred konkurencijom je ta što je besplatan, a u kombinaciji s također besplatnim mySql paketom može poslužiti za izradu vrlo moćnih web aplikacija.

Sve je više korisnika Internet-a koji vremenom postaju sve zahtevniji. Naredne web aplikacije će morati da nude više, da budu što bolje i da odgovaraju na što veći broj korisnikovih potreba. U bližoj budućnosti, kada se poboljšaju veze prema Internet-u, očekuje se i veliki napredak web aplikacija. Vrlo je moguće da u budućnosti web aplikacije zamene čak i pojedine desktop aplikacije (Google-ov odgovor na MS Office).

Što se tiče aplikacija za online oglase, sigurno je da će biti atraktivne i u budućnosti. Mogućnosti samih aplikacija će se proširiti i samim tim privući veliki broj korisnika. Jedno je sigurno a to je da ove web aplikacije treba da budu što jednostavnije, funkcionalne i da omoguće rad svim korisnicima bez obzira na njihov nivo znanja korištenja Internet-a, što predstavlja dodatni izazov programerima.

10 LITERATURA

- [1]. Brže, bolje: HTML & XML Michael Morrison
- [2]. Internet biblija
- [3]. PHP 4 , bill McCarty
- [4]. Neimenovane skripte sa Više tehničke škole –Subotica-
- [5]. <http://www.w3schools.com/>
- [6]. <http://www.mysql.com/>
- [7]. <http://www.php.net>