**VISOKA ŠKOLA ZA INFORMACIJSKE TEHNOLOGIJE**

**STRUČNI STUDIJ INFORMACIJSKE TEHNOLOGIJE**

**ZAGREB**

**DIPLOMSKI RAD**

**IZRADA OGLASNIKA U PHP-U**

**Larisa Čemeljić**

**Zagreb, svibanj 2018. godine**

**SADRŽAJ**

[POPIS SLIKA 3](#_Toc58857443)

[POPIS TABLICA 4](#_Toc58857444)

[1. UVOD 5](#_Toc58857445)

[2. INTERNETSKE TEHNOLOGIJE 7](#_Toc58857446)

[2.1. Hyper Text Transfer Protocol – HTTP 7](#_Toc58857447)

[2.2. Hypertext Markup Language - HTML 8](#_Toc58857448)

[2.2.1. Osnovni elementi i atributi HTML-a 9](#_Toc58857449)

[2.2.2. Poveznice, tablice i ostali netekstualni sadržaji 11](#_Toc58857450)

[2.2.3. Forma HTML-a i njezini osnovni elementi 13](#_Toc58857451)

[2.3. Cascading Style Sheets – CSS 16](#_Toc58857452)

[2.3.1. Cascading Style Sheets selektori 17](#_Toc58857453)

[2.3.2. Cascading Style Sheets svojstva 19](#_Toc58857454)

[3. HIPERTEXT PREPROCESSOR – PHP i BAZA PODATAKA 22](#_Toc58857455)

[3.1. Razvoj Hypertext Preprocessor – PHP-a 22](#_Toc58857456)

[3.2. Usporedba Hypertext Preprocessor-a i  Active Server Pages – ASP.NET 24](#_Toc58857457)

[3.3. Osnove Hypertext Preprocessor-a 25](#_Toc58857458)

[3.3.1. Naredba if, petlje i polja 26](#_Toc58857459)

[3.3.2. Funkcije i objekti 31](#_Toc58857460)

[3.4. Baza podataka 32](#_Toc58857461)

[4. PRAKTIČNI DIO – WEB STRANICA OGLASNIK 34](#_Toc58857462)

[4.1. Korišteni alati za izradu praktičnog dijela 35](#_Toc58857463)

[4.2. Baza podataka Oglasnik 36](#_Toc58857464)

[4.3. Elementi web stranice 38](#_Toc58857465)

[4.3.1. Zaglavlje web stranice 38](#_Toc58857466)

[4.3.1.1. Elementi zaglavlja za anonimnog korisnika 42](#_Toc58857467)

[4.3.1.1. Elementi zaglavlja za registriranog korisnika 44](#_Toc58857468)

[4.3.2. Tijelo web stranice 53](#_Toc58857469)

[4.3.3. Podnožje web stanice 61](#_Toc58857470)

[5. ZAKLJUČAK 63](#_Toc58857471)

[LITERATURA 65](#_Toc58857472)

[SAŽETAK 66](#_Toc58857473)

[SUMMARY 67](#_Toc58857474)

# POPIS SLIKA

[Slika 1: Prikaz forme u pretraživaču 15](#_Toc57983341)

[Slika 2: Struktura CSS-a 17](#_Toc57983342)

[Slika 3: Prikaz dijagrama koji prikazuje statistiku korištenja verzija PHP-a 23](#_Toc57983343)

[Slika 4: Prikaz dijagrama korištenja PHP-a u WordPress web stranicama 24](#_Toc57983344)

[Slika 5: Prikaz strukture baze podataka Oglasnika 37](#_Toc57983345)

[Slika 6: Prikaz zaglavlja Oglasnika za anonimnog korisnika 39](#_Toc57983346)

[Slika 7: Prikaz zaglavlja Oglasnika za registriranog korisnika 39](#_Toc57983347)

[Slika 8: Selektiranje kategorije u kojoj nije odabrana potkategorija 46](#_Toc57983348)

[Slika 9: Selektiranje kategorije u kojoj je odabran zadnji nivo potkategorije. 46](#_Toc57983349)

[Slika 10: Tablica uspješnih transakcija u Stripe-u 51](#_Toc57983350)

[Slika 11: Administrativno uređivanje korisnika 52](#_Toc57983351)

[Slika 12: Prikaz skripte uredi\_kategoriju.php i njene mogućnosti 54](#_Toc57983352)

[Slika 13: Prikaz svih aktivnih oglasa 55](#_Toc57983353)

[Slika 14: Administrativni prikaz odobrenog oglasa 59](#_Toc57983354)

[Slika 15: Podnožje Oglasnika 62](#_Toc57983355)

# POPIS TABLICA

[Tablica 1: Jedinice za definiranje veličine fonta 19](#_Toc58857507)

[Tablica 2: Oracle-ova usporedba PHP-a 5 i ASP.NET-a (izvor: www.php.com.hr) 25](#_Toc58857508)

# UVOD

U današnje vrijeme internet je dio svakodnevnog života. Postoje potrošači koji očekuju da na internetu nađu sve što žele i prodavači koji se trude što više toga prodati. Svakodnevica je da ljudi prodaju i kupuju putem interneta i za to koriste razne oglasnike. Takvi oglasnici omogućuju prezentaciju prodavača, komunikaciju između potrošača i prodavača te prodaju raznih proizvoda ili predmeta, čak i usluga. Zbog toga današnji kreatori web stranica imaju mogućnosti iskoristiti taj trend i razviti moćne dinamičke web lokacije koje imaju različite mogućnosti. Isti kreatori jednog takvog oglasnog prostora na svojoj web stranici imaju velike mogućnosti zarade, jer zarada putem interneta je danas sve veća.

U ovom radu je upravo takav jedan primjer web stranice obrađen, oglasnik za oglašavanje i pregledavanje oglasa putem interneta. Cilj diplomskog rada je istraživanje, planiranje i izrada prototipa web oglasnika. Oglasnik je jednostavan, pregledan i prije svega funkcionalan web oglasnik koji je prilagođen svim korisnicima interneta bez obzira na njihov nivo znanja korištenja interneta.

Drugo poglavlje (Internetske tehnologije) opisuje vrste tehnologija koje su korištene za izradu web oglasnika. Svaka pojedina tehnologija je opisana i sadrži primjer u praksi. Neke od tih tehnologija su HTML i CSS. Navedene tehnologije čine korisničko sučelje web stranice i zaslužne su za dizajn same web stranice.

Zatim, 3. poglavlje opisuje poslužiteljsku stranu web stranice, a to uključuje programski jezik PHP i MySQL što je ujedno i funkcionalnost web stranice. Programski jezik PHP korišten je za razvoj same web stranice. Za pohranu podataka korištena je relacijska baza podataka MySQL.

U praktičnom dijelu rada, tj. u 4. poglavlju najprije se opisuju alati korišteni za izradu web stranice kao što su Visual Studio Code uređivač teksta i WAMP paket koji je korišten za prikaz i interakciju web stranice na računalu. Opisana je struktura baze podataka kao i podjela tablica ovisno o njihovoj ulozi. Nakon toga su opisani elementi web stranice i koje funkcionalnosti sadrži pojedini element. Objašnjen je princip rada i mogućnosti web stranice s priloženim slikama.

Korisnici web stranice podijeljeni su u tri razine pa sukladno tomu korisnik može biti anonimni, registrirani i administrator. Anonimni korisnik može pregledavati, pretraživati oglase, kontaktirati pojedinog korisnika preko kontakt forme i, naravno, registrirati se. Kada se anonimni korisnik registrira upisom svojih podataka omogućeno mu je i dodavati oglase, promovirati ih, tj. platiti, spremati tuđe oglase i ostale funkcionalnosti nevezane za oglase. Administrator ima mogućnosti odobravanja oglasa koji su dodani, kao i njihovog brisanja. Također može dodavati i brisati kategorije oglasa, i uređivati uloge korisnika.

# INTERNETSKE TEHNOLOGIJE

Za razvoj web stranica najčešće se koriste iste internetske tehnologije. Kod klijentskog dijela koriste se tehnologije kao što su HTML i CSS, dok se kod poslužiteljskog dijela koriste PHP, SQL i slično.

## 2.1. Hyper Text Transfer Protocol – HTTP

HTTP (Hyper Text Transfer Protocol) je protokol, tj. metoda prijenosa hiperteksta (informacija) na web. Hipertekst su razni dokumenti koji u sebi sadrže veze na druge dokumente, a veze koje sadrže su hipertekstualne veze (engl. hyperlinks). Hypertext Markup Language (HTML) primjenjuje se samom pojavom usluge interneta 1990. godine. Omogućuje prezentaciju i objavu HTML dokumenta, odnosno upravlja načinom na koji preglednici zahtijevaju dokumente od poslužitelja i na koji način im je poslužitelj poslao istu. Naravno, Hypertext Markup Language samo je jedan od takvih protokola, ima i drugih također značajnih koji su na istoj razini.

Temelj ovog protokola je klijent – poslužitelj, te zahtijeva pouzdanu vezu na prijenosnoj razini za prijenos podataka, ali to ne mora biti nužno uz određeni protokol. Može biti definirano i drugačije, ali standard je priključna točka na kojoj sluša poslužitelj 80. Za prijenos svakog dokumenta HTTP 1.0 protokol definira otvaranje odvojenih TCP[[1]](#footnote-1) veza.

HTTP poslužitelj posjeduje dokumente i prima zahtjeve klijenata. Svi dokumenti su opisani: identifikatorom (URI – Unform Recourse Identifier), adresom (URL – Uniform Recourse Locator) i nazivom (URN – Uniform Recourse Name). Na osnovu navedenih parametara HTTP poslužitelj odlučuje kako odgovoriti na poslani zahtjev. Zahtjev sadrži naredbu (još se naziva metodom) koja definira akciju kao što su GET, POST, DELETE itd. Također ima adresu željenog dokumenta, verziju protokola i zaglavlja koja su definirana kroz parametre klijenta. Poslužitelj daje odgovor, ali najprije odlučuje hoće li prihvatiti komunikaciju s klijentom i uspostaviti vezu. U slučaju prihvaćanja zahtjeva za podacima (GET), HTTP poslužitelj šalje odgovor klijentu koji se sastoji od zaglavlja koji prethodi, informaciji, odnosno podacima koji su namijenjeni korisniku. Naravno, to zaglavlje sadrži parametre o poslužitelju, podacima i klijentu. Klijent prihvaća primljene podatke i prezentira informacije koje je prethodno izdvojio za korisnika. Također, klijent može dobiti obavijest o greški, umjesto podataka, kojoj uzrok može biti na strani poslužitelja ili na strani klijenta. Najčešće poruke o greški su 404 – File not found, što znači da datoteka nije pronađena ili 403 – Forbidden, što znači da pristup dokumentu nije dozvoljen.

HTTP protokol omogućava prijenos podataka između različitih protokola, neki od njih su NNTP koji je za prenošenje Usenet članaka (net news – mrežne vijesti), SMTP koji je za razmjenu elektronične pošte, FTP koji je za prijenos datoteka te ostali. Na ovaj način moguće je korištenje mrežnih usluga korištenjem samo mrežnih preglednika (npr. Chrome). Ako se HTTP protokolom prenose podaci drugih protokola, primjenjuje se jedan od dva postupka: tuneliranje, gdje se podatak prosljeđuje bez analize o kojem je protokolu riječ ili postupak prevođenja koji obavljaju poveznici (engl. gateway[[2]](#footnote-2)). U tom slučaju posrednik je veza klijenta i poslužitelja, koji može biti proxy poslužitelj, a njegova uloga je rasterećivanje prometa od poslužitelja na LAN mreži prema internetu. Klijent šalje zahtjev proxy poslužitelju koji nakon toga provjerava sadrži li taj podatak u međuspremniku (engl. cache), ako ima, vraća ga natrag klijentu. U slučaju da nema traženi podatak, proxy poslužitelj umjesto klijenta šalje upit web poslužitelju. Dobiveni odgovor web poslužitelja sprema u svoj međuspremnik kako bi pri sljedećem upitu bio brži i odgovor prosljeđuje korisniku koji je poslao zahtjev. Jedinstvena adresa URL definira protokol za pristup podatku, tj. poslužitelju i mjesto podatka na poslužitelju.

## 2.2. Hypertext Markup Language - HTML

HTML (Hypertext Markup Language) je jezik za opisivanje strukture hipertekst dokumenta koji je prošao kroz nekoliko verzija od svog nastanka 1989. godine. Danas je sadržan u brojnim standardnim opisima od strane World Wide Weba, prva specifikacija izdana je 1992. godine, dok se verzija 4.01 iz 1999. godine zadržala više od desetljeća kao glavni jezik weba. Jezik Hypertext Markup Language zadužen je za izgled stranice i predstavlja sve što korisnici vide kad pristupaju web stranici.

Jezik Hypertext Markup Language ima dobru stranu jer omogućuje proširivanje, odnosno dodavanje novih komandi s tim da ne poremeti izvršavanje ranije napisanih dokumenta u svojoj formi. Koncept je vezan za interpretiranje napisanog teksta što znači da svaki preglednik ne prevodi cijeli tekst napisan u HTML-u, već interpretira liniju po liniju i izvršava komande koje je protumačio iz teksta. Pravac izvršavanja instrukcija u HTML-u je isključivo odozgo prema dolje.

Sastoji se od oznaka (engl. tag) koji su osnovni elementi HTML-a i pomoću tih oznaka sve se obavlja, odnosno označavaju se dijelovi sadržaja dokumenta kao što su naslov, odlomak ili tablica. Također postoje oznake za uključivanje slika, filmova te hipertekstualnih veza (hiperlinkova). Hiperlinkovi omogućuju korisnicima kretanje kroz web jer povezuju trenutnu stranicu s drugim stranicama. Sadrže URL adresu željene stranice koju preglednik čita i preko nje dolazi do te željene stranice.

### 2.2.1. Osnovni elementi i atributi HTML-a

Osnovni elementi svake HTML stranice su oznake koje opisuju način na koji se nešto prikazuje. Sve oznake pišu se između znakova manje i veće (<…>). Neki imaju i svoju završnu oznaku i tako djeluju kao cjelokupan sadržaj, a drugi ga nemaju, već su sami za sebe. Sve završne oznake počinju s „</“, a završavaju s „>“ znakom. Slijedi popis osnovnih elemenata web stranice kojima se definira osnovna struktura HTML dokumenta, kao i primjer koda jedne jednostavne strukture. Osnovni elementi web stranice:

* doctype deklaracija,
* html,
* title,
* head i
* body.

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=“Content-Type“ content=“text/html;charset=UTF-8“ />

<title>Naziv web stranice</title>

</head>

<body>

<p>Ovdje se unosi sadržaj stranice.</p>

</body>

</html>

Doctype (Document Type Definition - DTD) deklaracija određuje pravila strukture XML dokumenta i mora biti prvi element web stranice. Ona govori pregledniku u kojoj je verziji HTML-a pisana web stranica i preglednik na osnovu toga odluči na koji način će prikazati stranicu.

Oznake <html> i </html>govore web pregledniku gdje započinje html dokument, a gdje završava. Sadržaj dokumenta mora biti unutar ovih oznaka, izuzev DTD deklaracije.

Dokument se dijeli na dva dijela unutar samo html elementa: zaglavlje i tijelo dokumenta, tj. <head> i <body>. Unutar zaglavlja dokumenta najčešće se navode naslov stranice (<title>) kao i jezične značajke HTML dokumenta.

Naime u HTML-u nisu uključeni hrvatski znakovi i zato je potrebna jedna od kodnih stranica koja podržava hrvatske znakove. U izradi web stranica za hrvatske znakove koristile su se tri kodne stranice (raspored znakova): Win-1250, ISO-8859-2 i Unicode UTF-8. U današnje vrijeme najrašireniji je UTF-8 raspored znakova. Za definiranje takve kodne stranice potrebno je dodati element <meta> unutar <head>odjeljka.

Dakle, element <meta> može se pojaviti unutar elementa <head> i on služi za specifikaciju metapodataka, tj. informacija o dokumentu kao što su autor, lista ključnih riječi, datum objava, jezik sadržaja dokumenta kako je već ranije spomenuto. Svi navedeni podaci su jako korisni kod pretraživanja.

HTML oznake nisu osjetljive na pisanje velikim ili malim slovima, tako da je oznaka <strong> isto što i <StRoNG>. Preporučljivo je da se pišu na način koji najviše odgovara dizajneru web stranice, a najčešće se pišu velikim slovima radi čitljivosti.

Kao i u drugim jezicima, tako i u HTML-u postoje komentari. Oni se ne prikazuju na samoj stranici, a omogućuju lakše snalaženje unutar koda. HTML komentari počinju sa znakom <!--, a završavaju sa -->. Mogu sadržati više linija i u njih se može zapisati bilo što. Korisni su kada neki dio koda treba privremeno izbrisati da ga preglednik ignorira, postavljaju se znakovi komentara i nije potrebno trajno brisanje samog koda.

U nekim slučajevima oznake posjeduju neke atribute s vrijednostima koji preciznije opisuju način prikaza teksta, slike, tablice itd. Atributi u HTML-u ne sadrže podatke, već informacije o podacima u dokumentu. Vrijednosti atributa u se mogu izostaviti iz navodnika, ali se to ne preporuča. Neki atributi mogu prihvatiti bilo kakvu vrijednost dok su drugi ograničeni. Mnogi atributi iziskuju za vrijednost da se unese neki broj ili postotak, pogotovo ako opisuju veličine i dužine. Te brojčane vrijednosti nikad ne sadrže mjerne jedinice, za visinu ili širinu primjenjuju se pikseli. Atributi HTML-a su također neosjetljivi na mala i velika slova.

Slijedi primjer oznake IMG koji postavlja neku sliku na stranicu:

Kod primjera pojavljuju se i atributi src, width, height i alt:

img src=“slika.jpg“ width=“150“ height=“100“ alt=“Moja slika“>

* atribut src (od engl. source) obavezno se mora koristiti; njegova vrijednost je URL slike koja se prikazuje; ako je pak slika smještena u istom direktoriju kao i web stranica, onda se piše kao u primjeru gore, ako je slika u drugom direktoriju, postavlja se „ime-direktorija/slika.jpg“
* atributi width i height koriste se kada se žele postaviti određene dimenzije slike; definiranjem tih dimenzija mijenja se veličina slike, bez ovih atributa slika se učitava u svojoj pravoj veličini
* atribut alt koristi se da bi se izbjegli određeni problemi kod učitavanje slike; vrijednost atributa alt je tekst koji želimo ispisati umjesto same slike, u slučaju da se slika ne učita istog trenutka ili kao informacija korisniku ako se slika ne učita.

### 2.2.2. Poveznice, tablice i ostali netekstualni sadržaji

Osim učitavanje raznih slika, web stranice ne bi imale danas puno smisla bez poveznica. Poveznicama se povezuje web stranica s nekom drugom stranicom. Dakle, kada korisnik klikne na poveznicu, tj. link, njegov preglednik će učitati onu stranicu na koju ga poveznica vodi.

Postoje tri vrste poveznica: ona koja vodi na neko mjesto na trenutnoj stranici, ona koja vodi na stranicu unutar istih web stranica i ona koja vodi na neki vanjski URL. Za postavljanje poveznice koristi se oznaka A, a njezin završetak potrebno je označiti završnom oznakom gdje završava tekst koji predstavlja poveznica. Primjer prethodne poveznice je poveznica na drugu stranicu istih web stranica, atribut href upisan je samo naziv stranice koja se nalazi u trenutnom direktoriju. Ako se pak korisnika želi odvesti na neku drugu web stranicu, samo se napiše njegova potpuna URL adresa. URL adresa sadrži informaciju gdje se datoteka nalazi i što pretraživač treba učiniti s njom. Svaka datoteka na internetu ima svoju jedinstvenu URL adresu.

Osim teksta kao oblika poveznice, što je najčešći slučaj, i slika može služiti kao poveznica. Tako se postavljanje slike kao oblika poveznice postavi slika unutar oznaka A. Kada korisnik prijeđe preko takve slike, pojavit će se rub u istoj, kao što su definirane i tekstualne poveznice. Da bi se to uklonilo, jednostavno se sve vrijednosti ruba slike postave na nulu.

Ponekad je korisnije da slika bude poveznica na više stranica, tj. da nije cijela slika poveznica nego u dijelovima, tada se izrađuju slikovne mape. Slikovna mapa je jedna slika čija su područja definirana da budu poveznice i tada se uvode koordinate dijelova slike u oznaku. Najprije se definira mapa poveznica za određenu sliku, za to je potrebno znati dimenzije slike i područja koja će biti poveznice. Naravno, umjesto x1,y1 i ostalih koordinata unose se prave vrijednosti u pikselima za područje za koje vrijede. Kao što je vidljivo iz primjera, postoje tri osnovna oblika mapiranja slika, a to su pravokutnik, krug i poligon. Poligon se može definirati proizvoljnim brojem koordinata između kojih će redom biti povučeni pravci i poveznica će djelovati na unutrašnje područje. Slika koja predstavlja slikovnu mapu ne mora biti unutar oznake A, ključan je samo atribut usemap.

Valja još spomenuti liste i tablice. Što se tiče lista, svaka mora imati svoj početak i kraj (<ul> i </ul>), a unutar njenih oznaka nalaze se elementi liste (<li>). Postoje poredane i neporedane, kod poredanih lista koristi se oznaka <ol>. Za tip točke kojom se biti označen svaki element koristi se atribut TYPE (npr. circle), također se može definirati hoće li elementi liste biti označeni brojem ili slovom koji se povećava. Tako se može dodati i atribut koji govori od kuda počinje nabrajanje (start).

Tablice su vjerojatno najčešće korišteni element stranica HTML-a. Osim samog prikazivanja tablica popunjenih sadržajem, korisne su i pri dizajnu za raspored elemenata na stranici. Korištenjem nevidljivih tablica, bez rubova, moguće je podijeliti stranicu na više dijelova. Osnovna struktura tablica (oznaka <table>) sastoji se od redaka (oznaka <tr>) i ćelija (oznaka <td>). Naravno, svaka tablica može imati određeni broj redaka i svaki redak određeni broj ćelija. Samo se unutar ćelija može postaviti sadržaj. Osim širine i visine tablice (width i height) postoje još neki atributi koji se mogu definirati unutar svake tablice:

* align – poravnatost tablice (npr. „left“, „center“, „right“),
* background – naziv datoteke za pozadinu tablice,
* bgcolor – pozadinska boja tablice,
* border – širina ruba u pikselima,
* cellpadding – udaljenost od ruba ćelije do njenog sadržaja u pikselima i
* cellspacing – razmak između susjednih ćelija u pikselima.

Tablica koja se najčešće koristi za slaganje elemenata na stranici, nema prostora između ćelija, od ruba ćelije do njenog sadržaja također nema prostora, nema rub i zauzima cijelu širinu ekrana.

### 2.2.3. Forma HTML-a i njezini osnovni elementi

Forme HTML-a služe za unos podataka na stranici. Jako se često koriste za slanje elektroničke pošte, registraciju na nekoj stranici itd. i zato su važan dio dizajna web stranica pogotovo za online oglase. Sadržaj forme šalje se nekoj poslužiteljskoj skripti (PHP, JavaScript, ASP itd.) koja podatke obrađuje i na osnovu toga generira neki izlaz. Unutar oznake form navode se sva polja forme. Oznaka form ima dva atributa: metode i action. U action unosi se skripta na koju se sadržaj forme šalje, a u method unosi se korištena metoda slanja. Dvije su moguće metode: POST i GET, a više o njima je objašnjeno u podnaslovu HTTP protokol.

Forma HTML-a bez polja i sadržaja nije forma, dakle svaka mora sadržati neka polja ili gumbe. Svaki element forme trebao bi imati svoje ime definirano atributom name, bez tog atributa sadržaj se neće poslati jer se svaki sadržaj u formi šalje upareno s nazivom polja.

Polje za unos jednog retka teksta je najjednostavnije polje i radi se o oznaci input tipa „text“. Njemu se još može dodati veličina, odnosno broj upisanih vidljivih znakova, a to se definira korištenjem atributa size. Osim atributa size može se definirati i maksimalan broj znakova koje korisnik može upisati atributom maxlenght. Također je uobičajeno definirati što će u polje biti upisano kada se stranica učita korištenjem atributa value.

Drugi tip polja za unos teksta je textarea, a ono služi za unos dužeg teksta. Predstavlja područje koje ima određeni broj redaka (rows atribut) i u svaki redak stane određeni broj znakova. Ovaj tip polja ima završnu oznaku, a između oznaka može se upisati tekst koji je njegov sadržaj.

Lista definiranih opcija je idealna u slučaju da korisnik ne upisuje podatke. Takva lista se još zove padajući izbornik (engl. pop-up box), s njom se uštedi na prostoru na web stranici. Za to služi oznaka SELECT koja može u sebi sadržati atribut value. Ukoliko se definira atribut value, tada će, u slučaju da je ona odabrana, biti poslan tekst te opcije.

Korisniku se, osim padajućeg izbornika, kao izbor više opcija može ponuditi i polje za glasanje, tj. radio gumb. Princip je jednostavan, više različitih radio gumbi imenuje se istim imenom i korisnik će moći odabrati samo jednog od njih. Automatski može biti odabran prvi od navedenih, a to se definira oznakom checked.

Za odabir više odgovora od korisnika koristi se polje za obilježavanje, tj. checkbox. Ovo polje sliči radio gumbu, ali svaki checkbox stoji sam za sebe, neovisan je o drugim elementima. Ima dva moguća stanja: uključen ili isključen. Razlikuje se još od radio gumba što se prikazuje kao mali kvadrat i korisnik može obilježiti koliko god malih kvadrata želi.

Još jedan poseban element forme je skriveno polje. Njega korisnik ne vidi i najčešće se koristi za unos zaporke.

Ostao je obavezan element za slanje forme. Za to se koristi gumb submit koji kompletni sadržaj šalje na skriptu. Osim gumba submit koji znači izvršiti, odnosno da je popunjavanje forme gotovo, još se obično ponudi i gumb reset. Gumb reset napravi baš to i samo ime gumba znači, resetira formu i omogućuje ponovno popunjavanje forme. On postavlja sve u prvobitno stanje, tj. postavlja sve na početne vrijednosti. Naime, atributi nisu potrebni za ovu akciju jer se ne prenose podaci, već se pokreće akcija.

U nastavku je primjer jedne forme i na slici 1 rezultat forme sa svim elementima gore navedenim:

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=“Content-Type“ content=“text/html;charset=UTF-8“ />

<title>Registracija</title>

</head>

<body>

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

<p>Vaše ime i prezime: <input type=“text“ name=“ime“ size=“50“></p>

<p>Vaša adresa: <textarea rows=“8“ cols=“50“ name=“adresa“>Unesite adresu.

</textarea></p>

<p>Vaš grad stanovanja:<select name=“Izbornik“>

<option value=“ZG“>Zagreb

<option value=“ST“>Split

<option>Zadar

</select></p>

<p>Jeste li već koristili online oglasnik?</br>

<input type=“radio“ name=“izbor“ value=“DA“ checked>Jesam

<input type=“radio“ name=“izbor“ value=“NE“>Nisam</p>

<p><input type=“checkbox“ name=“obavijesti“ value=“OB“>

Želim primati obavijesti.</br>

<input type=“checkbox“ name=“igra“ value=“IG“>

Želim sudjelovati u nagradnoj igri.</p>

<p>Vaša zaporka:<input type=“password“ name=“zaporka“ value=“zaporka“></p>

<input type=“submit“ value=“POŠALJI“>

<input type=“reset“ value=“OTKAŽI“>

</body>

</html>



Slika 1: Prikaz forme u pretraživaču

## 2.3. Cascading Style Sheets – CSS

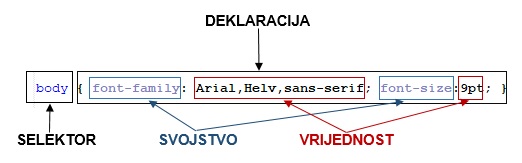
Cascading Style Sheets na jednostavan način omogućuje da se razdvoji izgled web stranice od njenog sadržaja. Kao standard donio ga je World Wide Web Consortium (W3C). Uloga Cascading Style Sheets-a je da sve informacije sa web stranica budu pospremljene u HTML datotekama, a da ne sadrže podatke kako se te informacije prikazuju. Dakle, podaci o prikazu stranica trebali bi ići u Cascading Style Sheets datoteke. Korištenjem Cascading Style Sheets-a definira se kako se pojedini element web stranice prikazuje. On omogućuje još i korištenje klasa koje određuju svojstva (stil) pojedinog elementa. Tako se mogu različiti elementi označiti istom klasom i prikazati na isti način. Ukratko, CSS pojednostavljuje da, primjerice, svi naslovi na web stranici imaju određeni stil i kad se taj stil želi promijeniti, mijenja se u jednoj jedinoj datoteci. Nema potrebe na svakoj HTML datoteci svaki naslov pojedinačno mijenjati. Naravno, tu CSS datoteku treba povezati sa svakom stranicom na čiji se izgled želi utjecati. Povezivanje CSS datoteke sa HTML stranica definira se na sljedeći način:

<link href=“style.css“ rel=“stylesheet“ type=“text/css“>

Navedeni kod upisuje se unutar zaglavlja stranice (head), a style.css je ime css datoteke koja u sebi sadrži stilove za sve elemente na web stranici. Isto se ponovi na svim stranicama weba i isti će stil biti definiran za sve njih.

Stil se također može definirati u samom kodu HTML stranice. Ovakav pristup može poslužiti za neke specifične elemente za neku stranicu, ali time se gubi smisao CSS-a i odvajanja prezentacijskih podataka u posebnu datoteku.

Poznavanje CSS sintakse ključno je kako bi se definirao stil za sve elemente. Zato je potrebno slijediti određena pravila. Na slici 2 je prikaz osnovne strukture CSS-a.



Slika 2: Struktura CSS-a

Kao što je vidljivo sa slike 2, svaka CSS definicija mora imati selektor i deklaraciju. Deklaracija dolazi odmah iza selektora i sadržana je unutar vitičastih zagrada. Sastoji se od jednog ili više svojstava odvojenih točka-zarezom. Svojstvo se sastoji od naziva, dvotočke i vrijednosti. Može imati više vrijednosti koje se odvajaju zarezom. Uz vrijednost može stajati i

jedinica kao u gore navedenoj strukturi. Između vrijednosti i jedinice ne smije biti razmak.

### 2.3.1. Cascading Style Sheets selektori

Selektori u CSS-a su elementi u HTML-u i preglednici ih koriste da odrede na koje elemente stranice utječe naredba CSS. Osim što selektori mogu odrediti neki HTML element, oni mogu puno više. Postoji više tipova selektora s više mogućnosti, a to su:

* selektor HTML elementa,
* klase,
* ID selektori,
* kontekstualni selektori i
* pseudo klase linkova.

Najjednostavniji je onaj kojim se identificira neki element. Da bi se napisao selektorHTMLelementa, jednostavno se napiše naziv oznake bez znakova veće i manje (< i >). Ovakvi selektori pogodni su za izmjenjivanje prikaza svih elemenata na stranici. Ovim je jednostavnim načinom vrlo lako promijeniti svaki odlomak teksta. Ako se žele promijeniti samo specifični odlomci, već nastupaju komplikacije. Rješenje ovog problema su klase. Definiranjem stila za pojedinu klasu i pridodavanju klase određenom elementu daje im njihov jedinstveni izgled. Dva su tipa klasa: klase koje određuju sve elemente i klase koje određuju određene elemente. Za isti način prikaza ćelije i odlomka koriste se klase koje određuju sve elemente. Kod ovih klasa selektor počinje s točkom.

U HTML dokumentu,klase se pridodaju elementima korištenjem class, i kao što je već spomenuto, gornji primjer klase može se pridijeliti svom elementu.

Klase koje određuju određene elemente, ograničene su s primjenom samo na elemente određenog tipa. Selektor, ovog tipa klase, počinje s elementom na koji se primjenjuje, zatim točka i na kraju naziv klase.

IDselektori imaju sličnu sintaksu i način rada kao i klase. Razlika je što se ID selektori definiraju isključivo samo na jedan element. Selektor započinje znakom #, te ID elementa kojeg određuje. Navedeni selektor neće utjecati niti na jedan drugi element osim na onog koji ima definirano ID svojstvo koje odgovara onom iz selektora.

Konstruktivni selektori služe za selektiranje pojedinih elemenata, unutar nekih drugih elemenata. Primjerice, potrebno je selektirati sve B oznake unutar DIV oznake.

Koristan primjer iz prakse ID selektora je utjecaj na prikaz slike kao poveznice. Naime, slike koje se koriste kao poveznica imaju svoj rub, a jedini način da se rub izbjegne je unutar HTML koda, upisivanjem kod svake slike border=“0“. Da bi se takvo što izbjeglo i uštedjelo na vremenu, primjenjuju se ID selektori i na taj se način određuje da sve slike unutar poveznice nemaju rub.

Preostale su još pseudo klase poveznica, one pak određuju izgled poveznica u različitim situacijama, kao što su:

* a:link – obična poveznica,
* a:visited – već posjećena poveznica,
* a:hover – poveznica pri prelasku miša preko njega i
* a:active – poveznica kada korisnik klikne na njega.

Pri korištenju ovih klasa i pri pisanju poveznice ne treba koristiti nikakav poseban HTML kod, već se poveznica piše na standardan način. Pseudo klase mogu se kombinirati i s drugim selektorima.

### 2.3.2. Cascading Style Sheets svojstva

Najvažniji dio sadržaja svake stranice na koji korisnici obraćaju pozornost je tekst i web dizajner se svakako mora potruditi oko njegovog prikaza. Njegova boja može se definirati korištenjem jednog od 16 tekstualnih naziva ili heksadecimalno. Također, može se mijenjati veličina fonta teksta, ali valja pripaziti na jedinice. Dostupne jedinice prikazane su u tablici 1. uz koje se, naravno, moraju definirati i brojčane vrijednosti.

Tablica 1: Jedinice za definiranje veličine fonta

|  |  |  |
| --- | --- | --- |
| Jedinica | Kratica | Opis |
| em | em | Visina fonta |
| ex | ex | Visina slova x u fontu |
| pica | pc | 1 pica je 12 pointa |
| point | pt | 1/72 inča |
| piksel | px | Jedna točka na ekranu |
| milimetar | mm | Veličina pri ispisu |
| centimetar | cm | Veličina pri ispisu |
| inch | in | Veličina pri ispisu |

Atribut koji utječe direktno na čitljivost dokumenta je font. Najlakši za čitanje je font Times New Roman, osim njega često korišteni su još Arial i Verdana. Preporučljivo je koristiti one fontove koji dolaze uz operacijske sustave, da bi se sa sigurnošću učitali korisnicima web stranica. Također se u sklopu svojstva font-family mogu navesti različiti fontovi odvojeni zarezom, što znači da će se prvi dostupan font prikazati korisniku. Ako se pak dogodi da niti jedan od navedenih nije dostupan (ne postoji na softveru računala ili se ne može prikazati), korisnikov preglednik će automatski prikazati font Times New Roman (vrijedi za Windows platformu). Za dodatno uređivanje fonta poslužit će text-decoration i font-weight kojim je moguće tekst podcrtati, prekrižiti ili koristiti stil bold.

Pozadine je moguće definirati za svaki HTML element. Za definiranje pozadinske boje koristi se svojstvo background-color, a za pozadinske slike svojstvo background-image. U slučaju da je pozadinska slika nedostupna ili se još učitava, može se definirati pozadinska boja koja će sličiti pozadinskoj slici i olakšati korisniku čitanje teksta. Ovo je korisno ako se koristi tamna pozadinska slika i bijela boja teksta. Dok se slika učitava, postavi se tamna boja pozadine da se izbjegne situacija s automatskom bijelom pozadinom i zadanim bijelim tekstom.

Naravno, adresa slike koja predstavlja pozadinsku sliku smješta se unutar url() jer je to standardni način definiranja vanjskih komponenti koje se koriste na web stranici u sklopu CSS-a. Za statičnu pozadinsku sliku koristi se svojstvo background-attachment, a vrijednost se postavlja na fixed. Kod definiranja pozicije same pozadine koristi se svojstvo background-position. Još se može spriječiti ponavljanje pozadine korištenjem svojstva background-repet.

Svojstvo koje objedinjuje sva navedena svojstva prima sve vrijednosti odvojene zarezom. Rubovi su također nužni kod izrade web stranica. Da bi se tablicama dao njihov specifični izgled, postoji: širina ruba (engl. border-width), boja ruba (engl. border-color) i stil ruba (engl. border-style). Svojstvo širine ruba može se definirati vrijednosti u nekoj jedinici (npr. „1px“) ili kao tanak (engl. thin), srednja debljina (engl. medium) ili debeo (engl. thick). Boja se definira na standardni način za sve boje u HTML-u. Stil ruba ima mnoge opcije, kao što su nijedan (engl. none), točkast (engl. dotted), iscrtkan (engl. dashed), puna crta (engl. solid), dupla crta (engl. double) itd. Najčešće korišteni je solid, i slično kao i kod definiranja pozadine, koristi se svojstvo ruba (engl. border), u njemu se navode sva svojstva rubova.

Osim definiranja ruba sa svake strane nekog elementa, postoji mogućnost definiranja svojstva svakog ruba (lijevog, desnog, gornjeg ili donjeg). Za takvo definiranje postoje sljedeća svojstva: border-top-width, border-top-color i border-top-style. Slični nazivi bi se definirali za ostale rubove, samo se umjesto top upisuje bottom, left ili right.

Marginesu nevidljivi prostor između svakog elementa i elemenata između njega. Definiraju se na sličan način kao i pozadine, korištenjem svojstva margin. Definira sa za sve strane oko elemenata, ali mogu se koristiti i sljedeća svojstva:

* margin-top – definiranje gornje margine,
* margin-right – definiranje desne margine,
* margin-bottom – definiranje donje margine i
* margin-left – definiranje lijeve margine.

Također, može se definirati padding, a to je prostor između ruba elementa i njegovog sadržaja. Naravno, može se definirati za svaki rub elementa ili za sve njih.

Ostala su još brojna svojstva, ali od korisnih još se može definirati izgled kursora kada se nalazi nad nekim elementom. Postoji nekoliko korisnih vrijednosti za svojstvo cursor:

* hand – ako se želi definirati da kursor ima izgled ruke
* wait – izgled pješčanog sata, što znači da se čeka na neku akciju
* crosshair – izgled tzv. ciljnika.

# 3. HYPERTEXT PREPROCESSOR – PHP I BAZA PODATAKA

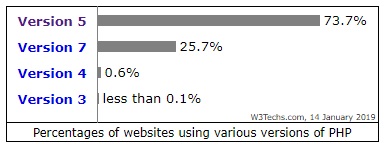
PHP (engl. Hypertext Preprocessor prije Personal Home Page Tools) je poslužiteljski skriptni jezik koji se izvršava na poslužitelju i omogućava sve standardne poslužiteljske akcije kao što su: komunikacija s bazom podataka, slanje elektroničke pošte, čitanje i pisanje u datoteke na poslužitelju i slično. Pozitivna strana PHP-a je u velikom broju biblioteka koje dolaze uz njega, uz pomoć kojih se mogu generirati slike, gotovi PDF dokumenti ili pak Flash animacije i sve to dinamički na poslužiteljskoj strani. Samo korištenje PHP-a vezano je uz poslužiteljski bazirane na Unix-u, ali se može koristiti i na Windowsima. Gotovo je nemoguće naći komercijalni poslužitelj koji nudi PHP i da ga pokreće operacijski sustav Microsoft.

## 3.1. Razvoj Hypertext Preprocessor – PHP-a

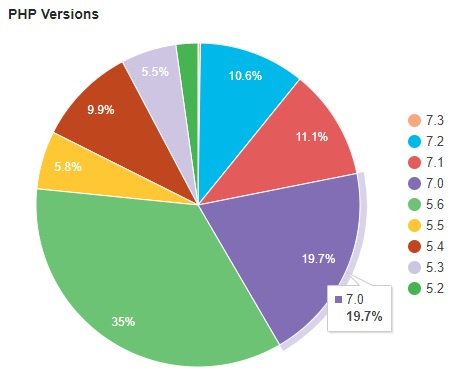
Rasmus Lardorf je 1994. godine dobio ideju o razvoju PHP-a. Radio je na njemu i 1995. godine objavio je prvu verziju pod nazivom Personal Home Page Tools. Verzija je bila vrlo jednostavna, imala je nekoliko funkcija i osnovni **raščlanjivač sintakse** za izvršavanje skripta. Već 1995. godine objavljuje se druga verzija pod nazivom PHP/FI u kojoj je **raščlanjivač sintakse** dorađen i ugrađen je drugi projekt Form Interpreter (od kojeg dolazi kratica FI). Značajno za ovu verziju je podržavanje rada s bazom podataka, što je ključno za prihvaćanje PHP-a i uključivanje mnogih na razvoj. Nadalje, bitna godina je 1997., kada se na PHP-u započelo raditi timski. Zeev Suraski i Andi Gutmans stvaraju treću verziju s potpuno novim **raščlanjivačem sintakse**. Tada je PHP počeo pružati jednostavnost i mnoge mogućnosti programiranja.

PHP 4 objavljen je 2001. godine s novim **raščlanjivačem sintakse** zvanim Zend, dodane su nove biblioteke koje ponovo povećavaju mogućnosti PHP-a. Popularizaciji PHP-a pridodala je mogućnost instalacije kao modula za većinu web poslužitelja, uključujući IIS [[3]](#footnote-3)(engl. Internet Information Services). Tri godine nakon toga uvedena je verzija 5 koja se dugo koristila jer verzija 6 nikad nije objavljena. Naime, Unicode ju je trebao uvesti, to je standard za razmjenu podataka usmjeren na prikaz slova na način neovisan o jeziku, računalnom programu ili računalnoj platformi. Sama implementacija verzije je kompleksna i donesene su neke loše odluke zbog kojih je povećana potrošnja memorije itd. Nakon ankete donesena je odluka da se verzija jednostavno povlači. Početkom 2015. godine objavljena je prva alfa verzija PHP-a 7, a prva stabilna verzija u prosincu iste godine. Godinu dana nakon toga, u prosincu 2016. godine, objavljena je verzija 7.1. Objavljena je lista uklonjenih funkcija i lista uklonjenih API-a i ekstenzija kako bi se što bezbolnije prešlo s verzije 5 na verziju 7.

Prema statistici korištenje verzije 5 postupno pada. U ožujku 2017. godine [www.plus.hr](http://www.plus.hr) objavljuje da verziju 5 koristi 95.6 % web stranica baziranih na PHP-u, a samo 3.4 % verziju 7. U siječnju 2019. godine W3Techs objavljuje drugačiju statistiku koja je prikazana na slici 3.



Slika 3: Prikaz dijagrama koji prikazuje statistiku korištenja verzija PHP-a

Najzastupljeniji CMS [[4]](#footnote-4) na internetu WordPress u svojim statistikama korištenja u siječnju 2019. godine dovodi podatak da 19.7 % WordPress web stranica koristi verziju 7 PHP-a što je prikazano na slici 4.

Slika 4: Prikaz dijagrama korištenja PHP-a u WordPress web stranicama

Svaka nova verzija PHP-a donosi poboljšanja, sigurnosne zakrpe i ponekad promjene u strukturi koda. Pojedini dijelovi prethodnih verzija postaju zastarjeli, a u nekim slučajevima cijele verzije nisu podržane kao što su PHP 4 i starije.

Dakle, ako se koriste dodaci, teme ili skripta koji se oslanjaju na zastarjeli PHP kod, tada kod valja nadograditi na noviju verziju PHP-a. Neke promjene mogu uzrokovati nekompatibilnosti starog koda i web stranica ne može raditi. Stoga je bitno informirati se o promjenama koje donose nove verzije PHP-a i uskladiti kod web stranice s najnovijom verzijom.

Najnovija verzija PHP 7.3 objavljena u prosincu 2018. godine radi mnogo brže od PHP 5. Web stranice na istom hardveru troše manje resursa i omogućuju brži rad korisnicima na web stranicama.

Osim statistike korištenja verzija, tu je i popis nekih popularnih web stranica koje podižu popularnost PHP-a, a koriste ga sljedeće web stranice: facebook.com, wikipedia.com, vk.com,

wordpress.com, pinterest.com.

## 3.2. Usporedba PHP-a i   ASP.NET-a

Prednost PHP-a je u ugrađenim bibliotekama pomoću kojih se jednostavno mogu generirati razni sadržaji. Vrlo bitna stvar je da ASP.NET zahtijeva veću upotrebu memorije i vrijeme izvršavanja samog koda, što je kod izgradnji web aplikacija vrlo bitno. Također, ASP.NET skripte su vrlo vezane uz Microsoftovu platformu i ona zahtijeva korištenje IIS-a (engl. Internet Information Services). IIS pak ima dugačak period slabosti, što administratorima zadaje poteškoće u razvoju web stranica i takav sustav ima povijest hakerskih napada. S druge strane Apache[[5]](#footnote-5) se može koristiti i na Linuxu, Mac OS X-u, Solarisu i mnogim drugim platformama.

I na kraju, sve se svodi na cijenu, pa je tako PHP besplatan i može se instalirati na bilo koje računalo bez potrebe za registracijom. ASP.NET je također besplatan, ali ako se prelazi na profesionalno razvojno okruženje, Windows-platforma s IIS-om košta. Slična je situacija i s bazom podataka. Baza MySQL, koju najčešće koristi PHP, besplatna je, a kupovina baze podataka MySQL koja ide uz ASP.NET košta desetak puta skuplje od najma.

Kvalitetan poslužitelj za ASP.NET s bazom MySQL teško je pronaći unutar Hrvatske.

Slijedi Tablica 2, usporedbe PHP-a i ASP.NET-a prije desetak godina od strane Oracle-a:

Tablica 2: Oracle-ova usporedba PHP-a 5 i ASP.NET-a (izvor: www.php.com.hr)

|  |  |  |
| --- | --- | --- |
|  | PHP 5 | ASP.NET |
| Cijena softvera | Besplatan | Besplatan |
| Cijena platforme | Besplatna | $$ |
| Brzina | Dobra | Loša |
| Efikasnost | Dobra | Loša |
| Sigurnost | Dobra | Dobra |
| Kvaliteta platforme | Dobra | Loša (samo IIS) |
| Izbor platforme | Bilo koja | Windows (samo IIS) |
| Open Source[[6]](#footnote-6) | Da | Ne |
| Iznimke | Da | Da |
| Objektno orijentirano programiranje | Dobro | dobro |

Čitajući razne članke i procjene, svatko ima svoje mišljenje koji je jezik bolji, ali neke činjenice mogu pomoći kod izbora je li to programski jezik poželjan za korištenje.

## 3.3. Osnove PHP-a

PHP, kao skriptni jezik, vrlo je sličan skriptama ASP, kombinira se s HTML-om, može se pisati u Notepad ++-u i označava se oznakama <?php kao početak php koda i ?> za kraj, naravno, sve između tih oznaka smatrat će se kodom PHP-a i izvršit će se prije slanja stranice korisniku. Slijedi jednostavni primjer skripte PHP-a. Skripta ispisuje riječ Pozdrav uz pomoć naredbe echo. Na kraju svake naredbe obavezno je stavljati točku-zarez (;). Ispisivanje varijabli moguće je već spomenutom naredbom echo ili istovjetnom naredbom print\_r. U PHP-u postoji nekoliko varijabli, no on je loosly typed programski jezik, što znači da varijablama nije potrebno odrediti tip, niti ih je potrebno deklarirati.

<?php

echo 'Pozdrav';

?>

Varijable je potrebno označiti prefiksom „$“. Nazivi varijabli osjetljivi su na velika i mala slova, pa tako $neka-Varijabla nije isto što i $Neka-varijabla. Kod naziva varijable dozvoljeno je da sadrži slova, brojke, podvlake, ali ne smije započeti brojkom.

Komentari u PHP-u, ukoliko se protežu kroz samo jednu liniju, na početak linije postavljaju znak „//“ ili „#“, ako se pak protežu kroz više linija, koriste se znakovi „/\*“ za početak komentara i „\*/“ za kraj komentara.

### 3.3.1. Naredba if, petlje i polja

If, odnosno naredba ako, provjerava neki uvjet, te ako je taj uvjet istinit, izvršava se kod. U protivnom, se ništa nije ni izvršilo. Operatori za usporedbe su sljedeći:

* < - manje
* <= - manje ili jednako
* == - jednako
* >= - veće ili jednako
* > - veće
* != - različito.

Operatori vraćaju ili istinitu (engl. true) ili lažnu (engl. false) vrijednost. Primjer korištenja naredbe if je u sljedećem kodu:

$godine = 17;

if ($godine < 18){

echo 'Ne možete glasati!';

}else{

echo'Možete glasati!';

}

U primjeru provjerava se uvjet koji uvijek mora biti unutar zagrada, je li varijabla $godine manja od 18. Dakle, ako je uvjet istinit, izvršit će se prvi kod unutar vitičastih zagrada, inače (engl. else) će se izvršiti drugi kod.

Dok PHP provjerava s operatorima je li neka tvrdnja istinita, uvijek će vratiti false vrijednosti samo za one varijable koje su jednake nuli ili za nedefinirane i nepostojeće objekte. Sve ostalo će smatrati istinitim. Tako se može provjeriti sadržaj nekih varijabli ili postoji li uopće neki objekt. Na primjer, kod provjere postoji li objekt dokument.all ispisuje se:

if (dokument.all)…

Ako provjera pokaže da postoji navedeni objekt, smatra se istinitim uvjetom i, naravno, vraća se nulta vrijednost. Na sličan se način provjerava je li neka varijabla jednaka nuli, primjer kod tog slučaja ispisuje se:

if ($godine)…

Ova naredba također može provjeravati više uvjeta odjednom. Operatori s kojima naredba if provjerava:

* jesu li su oba uvjeta istinita – koristi se I (engl. AND) operator „&&“,
* je li bar jedan uvjet istinit – koristi se ILI (engl. OR) operator „||“ i
* je li neki uvjet neispunjen – prije uvjeta postavi se znak negacije „!“.

U situaciji s mnogo mogućih ishoda nije praktično koristiti naredbu if, u tom se slučaju koristi switch struktura. Switch prima kao argument određeni izraz i provjerava je li on jednak jednom od zadanih slučaja. Idealan alat za izradu višenamjenskih stranica je upravo alat switch. Izgled strukture switch prikazan je u nastavku:

switch ( uvjet ){

case slučaj1

// naredbe koje se izvršavaju ukoliko je uvjet jednak slučaju 1

break;

case slučaj2

// naredbe koje se izvršavaju ukoliko je uvjet jednak slučaju 2

break;

case slučaj3

// naredbe koje se izvršavaju ukoliko je uvjet jednak slučaju 3

break;

default:

// naredbe koje se izvršavaju ukoliko uvjet nije jednak niti jednom slučaju

// njega se može izostaviti ukoliko se niti jedan naredba ne treba izvršiti u tom slučaju

}

Ključna riječ break označava završetak grane. Ukoliko se izostavi između dviju grana, naredbe obje grane će se izvršiti ukoliko je viša, tj. ona koja slijedi, prije aktivirana. Naredbe će se izvršavati sve dok se ne pojavi break, odnosno završetak strukture switch.

Petlje se koriste za ponavljanje nekog dijela koda određeni broj puta ili dok je neki uvjet istinit. U svakodnevnom životu petlje se mogu primijeniti na puno toga. Petlje u PHP-u imaju sličnu sintaksu kao i petlje u C-u, samo što je ispred imena varijabli, naravno, znak „$“.

Petlja for koristi brojač petlje koji se kod svakog izvršavanja nekih naredbi petlje smanji ili uveća. Ona se koristi kada se točno zna koliko je puta potrebno ponavljati blok petlje. Brojač petlje je bilo koja varijabla koja je već definirana ili se definira nova samo za potrebe petlje. Ukoliko se koristi nova varijabla, uobičajeno ime tih varijabli je $i, $j ili $k i one se najviše susreću u aplikacijama. Slijedi struktura i sintaksa za petlju for:

for (izraz1; izraz2; izraz3){

//naredbe …

}

Struktura petlje for sadrži:

* izraz1 – definira se koja će se varijabla koristiti kao brojač i postavlja se njena početna vrijednost,
* izraz2 – postavlja se uvjet koji treba biti ispunjen da bi se blok naredbi unutar vitičastih zagrada izvršio,
* izraz3 – operator uvećanja ili smanjenja (-- ili ++) i
* naredbe – koje se izvršavaju svaki put dok je uvjet istinit.

Konkretan primjer izvođenja naredbi točno određeni broj puta prikazan je na sljedećem primjeru koristeći petlju for:

<?php

$zbroj=0;

for ($i=1: $i <= 100; $i++){

$zbroj += $i;

}

echo 'Zbroj prvih sto brojeva je: ' .$zbroj;

?>

Rezultat primjera ispisuje: Zbroj prvih sto brojeva je: 5050. Naredba koja se izvodi unutar vitičastih zagrada napisana je skraćeno. Može se napisati i $zbroj = $zbroj + $i. Ovaj skraćeni izraz vrijedi i za ostale operacije. Skraćeni izraz daje osjećaj uštede vremena tek kod rada s objektima i puno većim nazivima varijabli, što u ovom primjeru nije vidljivo. Za ostale operacije, samo se umjesto plusa stavlja ona operacija koja je potreba, sve ostalo je isto.

Ponekad nije poznat broj ponavljanja izvođenja bloka naredbi, u tom se slučaju koristi petlja dowhile. Osnovna razlika između petlja for i do while je u tome što se u petlji for automatski povećava brojač. Kod petlje do while brojač se mora postavljati ručno, za to se koristi jedna linija u bloku naredbi ove petlje. Sintaksa petlje do while izgleda ovako:

do{

naredbe…

}while(uvjet)

Struktura ove petlje sadrži prvo naredbe koje se izvršavaju i nakon toga se ispituje uvjet koji treba biti istinit da bi se ponovilo izvršavanje bloka koda. Kod ove petlje je garantirano da će se blok naredbi izvršiti bar jednom, čak iako je uvjet neistinit, tako da bi se postigao isti rezultat kao i u prethodnom primjeru, s istim varijablama. Uvjet se mora postaviti malo drugačije, varijabla mora biti manja ili jednaka sto.

Ukoliko se želi izbjeći izvršavanje bloka naredbi prije nego se provjeri je li uvjet istinit, koristi se petlja while. Petlja while izvršava blok naredbi dokle god je izraz u uvjetu istinit. Taj uvjet se ispituje prije izvođenja bloka naredbi. Zato je moguće da se blok naredbi ne izvrši niti jednom ukoliko je uvjet odmah na početku neistinit. Sintaksa petlje while sadrži uvjet koji se provjerava na samom početku izvršavanja petlje i naredbe koje se izvršavaju dokle god je uvjet istinit.

Polja ili nizovisu složeni tip podataka i u članove polja mogu se spremiti varijable raznih tipova. To je uređena zbirka podataka, organizirana u obliku parova ključ – vrijednost. U PHP-u nizovi mogu biti indeksirani ili asocijativni. Ključevi indeksiranog niza počinju vrijednošću nula i to su cijeli brojevi, dok kod asocijativnog niza ključevi mogu biti nizovi (engl. string). Asocira na tablicu koja u prvom stupcu sprema ključeve, preko kojih pristupa vrijednostima u drugom stupcu. Obje vrste nizova se u PHP-u pohranjuju kao asocijativni niz.

Elementi indeksnog polja razlikuju se po ključu, odnosno indeksu koji je neki niz vrijednosti. Svakom elementu pristupa se tako da se iza naziva polja u uglatim zagradama navede indeks. Kao na primjer:

$mjeseci [0] = "Siječanj";

Kod asocijativnog niza, elementi niza za ključ, tj. indeksi imaju string vrijednost. Također kao i kod indeksnog niza, svakom elementu pristupa se preko indeksa i u uglate se zagrade navodi njegova indeksna vrijednost. Preporučljivo je da se unutar uglatih zagrada naziv indeksa stavlja unutar navodnika. Kao na primjer:

$obitelj ["brat"] = "Marko";

Kod nizova je vrlo korisna petlja foreach, ona pomaže kod prolaska kroz svaki element nekog niza i obavlja blok naredbi. Petlja sprema ključ i njegovu vrijednost elementa niza u varijable u kojem se za svako ponavljanje petlje nalaze ključ i vrijednost elementa niza na kojem se trenutno nalazi unutarnji pokazivač. Prije ulaska u petlju unutarnji se pokazivač nalazi na nuli i svakim ponavljanjem petlje povećava se za jedan. Ponovnim pozivanjem petlje foreach taj pokazivač se resetira, a petlja se ponavlja sve dok ne ostane bez elemenata niza.

### 3.3.2. Funkcije i objekti

Izradom web stranice i nakupljanjem linija koda, često je potrebno ponavljati neki dio koda na više mjesta unutar skripte. Rješenje tog problema su funkcije. Svrha svake funkcije je sadržavanje i izvršavanje koda. Jednom napisana funkcija, može se pozvati s bilo kojeg mjesta u skripti i to nebrojno mnogo puta. Funkcije se mogu pozivati iz drugih funkcija ili iz njih samih (tzv. rekurzivne funkcije). Ključna riječ kod definiranja funkcije je function i s njom se označava početak funkcije, a kod koji funkcija treba izvršiti nalazi se unutar vitičastih zagrada ({ i }).

Funkcija može primati parametre, dakle mogu se proslijediti neki podaci funkciji. Ti parametri se prosljeđuju unutar zagrade i odvojeni su zarezom. Jedna je od osnovnih mogućnosti funkcije da vraća određenu vrijednost, ali postoje i one koje ne vraćaju vrijednost. Funkcija u PHP-u vraća vrijednost s naredbom return i ova naredba trenutno prekida izvršavanje funkcije i vraća vrijednost neke varijable. Funkcije u PHP-u definiraju se sljedećom sintaksom:

function ime\_funkcije($parametar1,$ parametar2…$ parametarN){

// … blok naredbi funkcije

}

Postoje lokalne i globalne varijable. S obzirom na to da su funkcije samostalni dio koda, u njima se mogu definirati varijable s nazivima koje već postoje u kodu izvan funkcije i takve se varijable nazivaju lokalnima. Ove varijable se ponašaju neovisno o istoimenim varijablama u drugim dijelovima skripte i one ne mogu biti korištene izvan dijela u kojem su definirane. U funkcijama se mogu koristiti i globalne varijable, tj. varijable definirane u glavnom kodu izvan funkcije, ali njih je potrebno posebno deklarirati unutar funkcije i to se radi ključnom riječi global. Inače ih se ne bi mogle koristiti i one ne bi bile vidljive unutar funkcije.

PHP ima podršku rada s objektima. Funkcije se mogu grupirati u klase i stvoriti instance tog objekta kroz cijelu web stranicu. Tri su ključna principa objektno orijentiranih jezika, a to su: enkapsulacija (učahurivanje), nasljeđivanje i polimorfizam (višeobličje). Dakle, da bi se kreirao objekt, prvo treba stvoriti klasu, za to se koristi ključna riječ class, s nekim funkcijama i varijablama u toj klasi, te ju pozvati unutar koda. Funkcije unutar klase se zamišljaju kao metode istog objekta. Varijablama i funkcijama unutar objekta pristupa se pomoću pokazivača „->“.

Ukoliko postoji potreba da se pri instanciranju objekta automatski izvrše neke radnje kao što je spajanje na bazu podataka ili neku sličnu operaciju koja je potreba za daljnji rad objekta, koriste se konstruktori. Konstruktori su funkcije objekata koji se izvršavaju zajedno sa stvaranjem instance objekta. Oni imaju isto ime kao i sam objekt.

## 3.4. Baza podataka

Baza podataka je skup pohranjenih i međusobno povezanih podataka. Skoro svaka malo složenija web stranica svoj rad temelji na nekoj bazi podataka. Korištenje baze podataka može riješiti razne probleme i ostvariti svakakve ideje bez pretjerano kompliciranih podataka u skriptama. Jedna od najvećih prednosti PHP-a kao poslužiteljskog skriptnog jezika je to što omogućuje jednostavan način korištenja veliki broj baza podataka.

Kod programiranja u PHP-u, na raspolaganju su mnoge baze kao što su Access ili SQL Server, koje se koriste u slučaju korištenja operacijskog sustava Windows-a. Ipak, zbog svoje brzine i svojih mogućnosti, kao i cijenu, MySQL je najbolje rješenje. MySQL je poslužitelj baza podataka, odnosno to je softver kojem se može pristupiti preko mreže na sličan način kao i web poslužiteljima. Na poslužitelju može postojati veći broj baza podataka koje su samostalne i unutar projekta može se manipulirati podacima iz više baza na poslužitelju. Svakom korisničkom računu na poslužitelju mogu se dodijeliti razna administrativna prava na cijeli poslužitelj ili na određene baze. Neka od tih prava su izrada novih baza, pravo pristupa bazama, pravo uređivanja i slično.

Većina poznatih baza podataka podržava SQL (Structured Query Language), to je opisni jezik kojim se zadaju naredbe bazi i upravlja se s njenim podacima. Sve to obavlja se koristeći njezin DBMS (Database Management System), tj. sustav za upravljanje baze podataka. SQL koriste gotovo svi sustavi na tržištu jer ga podržavaju svi proizvođači baza podataka. SQL se sastoji od naredbi za dodavanje, čitanje, izmjenu i brisanje podataka. Pravila jezika odredio je ANSI (American National Standards Institute), što znači da je SQL otvoren i ne kontrolira ga niti jedna tvrtka.

Glavna namjena naredbeSELECT je dohvaćanje zapisa i ona omogućuje mnogo opcija, ali je, također, jedna od najkompleksnijih vrsta upita. Osnovni oblik, tj. upit je u sljedećem primjeru, u kojem se pribavljaju svi redci iz jedne tablice imenom Novosti:

SELECT \* FROM novosti;

Upit počinje nazivom vrste upita, nakon imena operacije slijedi lista polja koja će se pojaviti u tablici rezultata upita. Zvjezdica (\*) je skraćenica kojom se pribavljaju sva polja iz navedenih tablica. U ovom se primjeru također mogu navesti jednostavno sva polja, odvojena zarezom, iz željene tablice. I nakon liste polja slijedi FROM koji govori da slijedi lista tablica iz kojih se podaci pribavljaju, u gornjem primjeru to je samo jedna tablica.

Naredba SELECT ima razne mogućnosti, a neke od njih su:

* dohvaćene zapise poredati po određenom kriteriju, ključna riječ ORDER BY,
* sortirati dohvaćene zapise silazno (DESC) ili uzlazno (ASC),
* dohvaćenim zapisima odrediti broj (TOP) ili postotak (PERCENT),
* pobrojati zapise u određenoj tablici, ključnom riječi COUNT,
* spajanje više tablica, ključna riječ JOIN,
* uz pomoć naredbe DISTINCT izbaciti ponavljajuće rezultate upita i
* korištenje operatora veće (>), manje (<), jednako (=), različito (<>) uz pomoć WHERE uvjeta.

Osnovna namjena naredbe INSERT je dodavanje novih zapisa u neku tablicu. Za razliku od naredbe SELECT koja vraća tražene zapise, naredba INSERT izvršava dodavanje zapisa i ne vraća ništa. Osnovna je sintaksa ove naredbe:

INSERT INTO Tablica (popis-polja) VALUES (popis-vrijednosti);

Što znači da se umjesto popis-polja unose željena polja u koja se žele ubaciti neke vrijednosti, odvojena zarezom, a popis-vrijednosti su željene vrijednosti koje poprimaju navedena polja. Kod korištenja naredbe INSERT treba pripaziti da vrijednosti koje se nalaze unutar dijela VALUES prate redoslijed polja, odnosno da u prvo polje ide prva vrijednost, u drugo polje druga vrijednost itd.

Za izmjenu već postojećih zapisa u tablici koristi se naredba UPDATE. Isto kao i naredba INSERT, naredba UPDATE ne vraća nikakvu vrijednost već izmjenjuje vrijednosti zapisa. Kod postavljanja uvjeta u naredbi UPDATE koristi se polje ID. Sintaksa ove naredbe glasi:

UPDATE Tablice SET polje1 = vrijed.1, polje2 = vrijed.2,… WHERE uvjet7;

Bez uvjeta u naredbi zamijenit će se svi zapisi u tablici, a to je rijetka pojava u skriptama.

Naredba DELETE koristi se za brisanje zapisa iz određene tablice. Upit je vrlo jednostavan i lako pamtljiv kao što je vidljivo iz sljedećeg primjera:

DELETE FROM Tablica;

Gornja naredba bi izbrisala sve iz tablice imenom Tablica, ali takva situacija je rijetko potrebna u realnim skriptama. Puno češća je potreba za brisanjem jednog podatka ili manje grupe podataka i tu je potrebno koristiti uvjet WHERE kao u sljedećem primjeru:

DELETE FROM novosti WHERE id-novosti = 1;

# 4. PRAKTIČNI DIO – WEB STRANICA OGLASNIK

Web stranica Oglasnik prvenstveno služi za pretraživanje i pregledavanje oglasa. Korisnik registracijom na web stranicu dobiva mogućnost prijave na aplikaciju. Nakon prijave korisnik ostvaruje mogućnost dodavanja, tj. objave oglasa. Objavljeni oglas korisnik može naknadno promovirati, uređivati, deaktivirati kao i obrisati. Osim manipulacije s vlastitim oglasima, korisnik ima mogućnost spremanja tuđih oglasa koje želi pratiti. Ima mogućnost uređivanja svojih podataka i pregledavanje drugih korisnika koji su registrirani na web stranici Oglasnik. Osim anonimnog i registriranog korisnika postoji i administrator koji ima opcije odobravanja objavljenih oglasa ili brisanja istih, uređivanja uloge ostalih korisnika i dodavanje novih kategorija oglasa kao i uređivanje istih.

## 4.1. Korišteni alati za izradu praktičnog dijela

Visual Studio Code je uređivač teksta ili tekstni editor otvorenog koda kreiran i dizajniran kako bi olakšao razvoj web stranica, tj. aplikacija. Uređivač teksta razvija Tvrtka Microsoft, a on sadrži podršku za Windows, Linux i macOS operacijske sustave. Visual Studio Code baziran je na razvojnom okruženju Electron (engl. framework), odnosno alatu otvorenog koda korištenog za izradu aplikacija pomoću web tehnologija. Inače poznat i po kratici VS Code, uređivač teksta sadrži brojne pomoćne alate za pisanje koda:

* integrirano dovršavanje koda (engl. Intelligent code completion),
* isticanje sintakse (engl. Syntax highlighting),
* ugrađeni i prilagođeni predlošci koda (engl. snippets),
* ugrađena Git[[7]](#footnote-7) kontrola i GitHub,
* podudaranje zagrada (engl. Bracket matching),
* moguća promjena teme i prečaca za tipkovnicu i
* instalacija ekstenzija za dodatnu funkcionalnost.

WampServer ili WAMP (engl. Windows, Apache, MySQL, PHP) je skup programskih alata i razvojna okolina za Windows sustave. Instalacijom paketa WampServer dobivaju se alati potrebni za razvoj web aplikacija na lokalnom računalu.

Neke od funkcionalnosti WAMP-a:

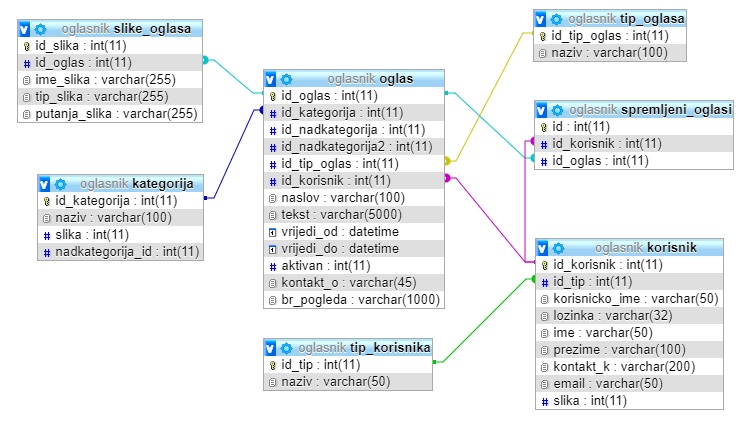
* upravljanje Apache i MySQL poslužiteljom,
* instalacija i promjena Apache, MySQL i PHP verzija,
* pristup zapisniku (engl. log file),
* pristup poslužitelju lokalnim računalom ili unutar lokalne mreže,
* pristup PHP postavkama i
* podešavanje poslužiteljskih postavki.

PhpMyAdmin je besplatna administracijska web aplikacija otvorenog koda za upravljanje bazama podataka MySQL-a preko web preglednika. Jedan je od najpopularnijih administracijskih alata MySQL-a i napisan je u PHP jeziku. Izvršavanje operacija MySQL-a moguće je uz pomoć phpMyAdmin i to preko korisničkog sučelja, kao što su nove baze, izmjena, dodavanje i brisanje tablica, dodavanje novih korisnika i izmjena dozvola pristupa istih, izvršavanje SQL upita, uvoz i izvoz podataka itd. Kreirane baze mogu se izvoziti u datoteke.sql, također, datoteke.sql mogu se uvoziti i ponovo koristiti te nadograđivati. Baza podataka može biti kreirana unosom programskog koda SQL-a ili uz pomoć opcija koje same generiraju upite i kreiraju bazu, tj. tablice.

## 4.2. Baza podataka Oglasnik

Prije same izrade projekta, koji se zove Oglasnik, potrebno je osmisliti bazu podataka, kao i njene tablice i veze između tih tablica. Aplikacija phpMyAdmin koristi relacijski model povezivanja podataka između tablica. Sve veze između tablica u projektu su binarne (1:N), jedan naprama više. Za izradu takvih veza potrebni su primarni i strani ključevi što je sve prikazano na slici 5. Tablice koje se nalaze u strukturi baze podataka su:

* kategorija – sadrži popis kategorija,
* oglas – sadrži popis oglasa,
* slike\_oglasa – sadrži slike od pojedinih oglasa,
* spremljeni\_oglasi – sadrži spremljene oglase pojedinih korisnika,
* tip\_oglasa – tablica daje informaciju koji od oglasa su plaćeni, a koji su besplatni,
* korisnik – sadrži popis korisnika i
* tip\_korisnika – tablica daje podatke o tipu korisnika.



Slika 5: Prikaz strukture baze podataka Oglasnika

Nakon planiranja i kreiranja baze podataka, potrebno je povezati se na bazu. Povezivanje na bazu podataka ostvareno je u jednoj skripti PHP (dbh.inc.php), prikazanoj u sljedećem kodu, koja je potom uključena u skriptu header.php. Na taj način, s obzirom na to da je u svim ostalim skriptama web stranice vidljivo zaglavlje, tj. skripta header.php, ostvarena je konstantna konekcija na kreiranu bazu.

<?php

$imeservera = 'localhost';

$DBkorisnik = 'larisa';

$DBlozinka = '123';

$bazapodataka = 'oglasnik';

$conn = mysqli\_connect($imeservera, $DBkorisnik, $DBlozinka, $bazapodataka);

$conn->set\_charset("utf8");

if (!$conn) {

    die("Povezivanje nije uspjelo: ".msqli\_connect\_error());

}

Kao što je i vidljivo iz samog koda, skripta sadrži varijable:

* $imeservera – koja predstavlja samo ime poslužitelja, a vrijednost joj je „localhost“,
* $DBkorisnik – sadrži vrijednost korisničkog imena, koje je „larisa“,
* $DBlozinka – ima vrijednost zaporke,
* $bazapodataka – govori o kojoj se bazi radi i
* $conn – sadrži funkciju koja prima sve gore navedene elemente kako bi se spajanje na bazu odvilo.

Osim navedenih varijabli skripta sadrži i funkciju set\_charset koja postavlja zadani skup znakova UTF-8[[8]](#footnote-8). Slijedi naredba if koja ispisuje grešku u slučaju da nije uspjelo povezivanje na bazu.

## 4.3. Elementi web stranice

Kako bi web stranica bila informativna, moderna, funkcionalna i korisna za sve vrste korisnika, mora sadržati neke osnovne elemente kao što su: zaglavlje, tijelo i podnožje web stranice. A svi navedeni elementi moraju sadržati elemente tipa:

* logotip,
* navigacija,
* tražilica,
* naslovi i podnaslovi,
* tekst,
* slike i
* forme itd.

### 4.3.1. Zaglavlje web stranice

Već spomenuta skripta header.php čini zaglavlje web stranice. Unutar zaglavlja korisnik na prvu uočava logotip, koji se iz tog razloga najčešće postavlja na gornji lijevi ugao web stranice. Smatra se da to mjesto korisnik prvo pogleda i to je usvojeno „pravilo“ koje vrijedi za zemlje u kojima se čita s lijeva na desno. Logotip je slika na koju je postavljena poveznica tako da se korisnik u svakom trenutku može vratiti na početnu stranicu.

Osim logotipa, zaglavlje sadrži navigacijsku traku, koja je najbitniji dio zaglavlja, a skoro i cijele web stranice. To je skup poveznica koje omogućuju korisniku kretanje kroz web stranicu. S obzirom na to da stranica ima dva prikaza: prikaz za anonimnog korisnika, tj. korisnika koji se nije još registrirao ili prijavio na stranicu i prikaz za korisnika koji je već registriran, tj. prijavljen, shodno tome zaglavlje promijeni svoj izgled.

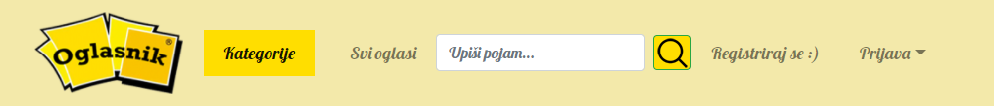
Za anonimnog korisnika zaglavlje nudi:

* poveznicu na listu kategorija, koja je ujedno i početna stranica Oglasnika,
* poveznicu na listu oglasa,
* tražilicu – koja omogućuje pretraživanje oglasa s ključnom riječju,
* poveznicu na stranicu za registraciju korisnika i
* gumb za prijavu korisnika.

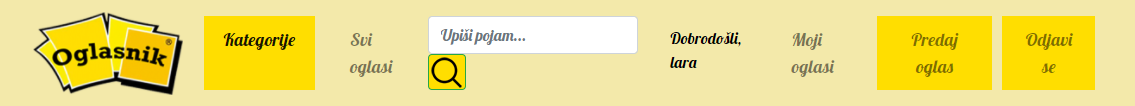
Registrirani korisnik, nakon što se prijavi u stranicu, vidi umjesto poveznice za registraciju i gumba za prijavu:

* običan tekst u kojem piše da je dobro došao i svoje kreirano korisničko ime,
* poveznica na oglase koje je korisnik sam kreirao,
* poveznica na stanicu za kreiranje oglasa i
* poveznica koja ga odjavljuje sa stranice i postaje ponovo anonimni korisnik.

Slijedi prikaz zaglavlja u oba slučaja na slici 6 i slici 7:



Slika 6: Prikaz zaglavlja Oglasnika za anonimnog korisnika



Slika 7: Prikaz zaglavlja Oglasnika za registriranog korisnika

U nastavku slijedi kod koji omogućava ova dva prikaza, glavna funkcionalnost koda je u naredbi if.

<?php

if (isset($\_SESSION['kor\_id']) && isset($\_SESSION['kor\_ime'])){

echo "<li class='nav-item'>";

echo "<a class='nav-link' href='oglasi\_ulogiranog\_korisnika.php'> Moji oglasi</a></li>";

}else{

echo "<li class='nav-item'>";

echo "<a class='nav-link' href='registracija.php'>Registriraj se :)</a></li>";

}

if (!(isset($\_SESSION['kor\_id']) && isset($\_SESSION['kor\_ime']))){

echo"<li class='nav-item dropdown'>

<a class='nav-link dropdown toggle' href='#' id='navbarDropdown' role='button'

data-toggle='dropdown' aria-haspopup='true' aria-expanded='false'> Prijava</a>

<div class='dropdown-menu nav-prijava-wrapper' aria labelledby ='navbarDropdown'>

<div class='dropdown-item nav-prijava'>

              <form action='includes/prijava.inc.php' method='post'>

                      <div class='col-auto'>

<input type='text' name='kor\_ime' class='form-control' placeholder='Korisničko ime/e-mail'>

<input type='password' class='form-control' name='pass' placeholder='Lozinka'>

<a href='zaboravljena\_lozinka.php'>Zaboravili ste lozinku?</a>

<button id='prijava' type='submit' name='prijavaBut'> Prijavi se </button>

                         </div>

                     </form>

                </div>

            </div>";

}else{

echo'<li class="nav-item"><form action=odaberi\_kategoriju.php method="post">

<button id="dodajOglas" class="nav-link" type="submit" name="dodajOglas"> Predaj oglas </button>

        </form></li>';

    echo '<li class="nav-item"><form action=includes/odjava.inc.php method="post">

<button id="odjava" type="submit"  class="nav-link" name="odjavaBut"> Odjavi se </button>

                         </form></li>';

     }

?>

Unutar naredbe if postavljeno je niz poveznica na odgovarajuću skriptu, također kreirana je forma za prijavu, kao i forma koja sadrži gumb za prijavu i za odjavu. Osim naredbe if spominje se $\_SESSION. Sesija (engl. session) jedna je od metoda upravljanja podataka o stanju.

Pretraživač „čuva“ identifikator sesije tako da se vrijednosti varijabli ne bi čuvale lokalno. Sesija se prvo mora započeti, a zatim identificirati i inicijalizirati postojeću sesiju. Ovo je kreirano u skripti za prijavu jer se čuva korisnički jedinstveni broj koji mu je dodijeljen tokom registracije kao i korisničko ime. Ovako to izgleda u kodu:

session\_start();

$\_SESSION['kor\_id'] = $row['id\_korisnik'];

$\_SESSION['kor\_ime'] = $row['korisnicko\_ime'];

Pozivom funkcije session\_start() generira se identifikator nove sesije. Vrijednost identifikatora sesije je grupa od 32 nasumično generirane heksadecimalne šifre. Poslije toga su definirane varijable koje će se kasnije koristiti za provjeru je li korisnik prijavljen ili ne. To su redovi iz tablice korisnik koji su dohvaćeni jednostavnim upitom SQL.

U skripti za prijavu korisnika provodi se niz radnji:

* provjera je li korisnik popunio oba polja (korisničko ime i zaporku),
* provjera je li u bazi podataka postoji korisnik s navedenim podacima,
* ako je korisnik upisao pogrešno korisničko ime ili pak zaporku, javlja mu se greška o istom i
* u slučaju da je sve od provjera u redu, sprema se već spomenuta sesija.

Registrirani korisnici koji se ne mogu prijaviti na web stranicu jer su zaboravili svoju lozinku imaju mogućnost resetiranje lozinke, tj. ponovo postavljanje lozinke korištenjem poveznice „Zaboravili ste lozinku?“. Za postupak resetiranje lozinke korištene su sljedeće skripte:

* zaboravljena\_lozinka.php,
* obavijest.php i
* resetiranje\_lozinke.php.

U prvoj skripti postavljena je forma koja prima jedan podatak korisnika i gumb koji provjerava uneseni podatak korisnika. Provjerava se je li upisana adresa elektroničke pošte koja je korištena prilikom registracije korisnika. Ako je unesena adresa elektroničke pošte ispravna, korisniku se otvara skripta obavijest.php na kojoj je obaviješten da mu je poslana elektronička poruka. Također se dohvaća njegov identifikacijski broj na osnovu preuzete adrese elektroničke pošte. U elektroničkoj poruci prima poveznicu na skriptu resetiranje\_lozinke.php s pripadajućim identifikacijskim brojem u url-u. Navedena skripta nudi mu ponovo postavljanje lozinke, a za ponovo postavljene potrebno je dva put upisati identičnu lozinku. Provjerava se jesu li sva polja popunjena i podudaraju li se lozinke. Ako obje provjere ne ispišu grešku, korisnik je uspješno resetirao lozinku i upisuje se novi podatak u tablicu korisnik pomoću upita UPDATE na bazu podataka.

#### **4.3.1.1. Elementi zaglavlja za anonimnog korisnika**

Za registraciju korisnika kreirane su dvije skripte, jedna s jednostavnom formom HTML-a i druga s funkcionalnim kodom PHP-a. U prvoj skripti, osim forme HTML-a, postavljena je naredba if koja ispisuje greške, ovisno koju je grešku korisnik napravio. Greške su proslijeđene pomoću metode slanja GET i metode primanja podataka POST . Prenošenje podataka moguće je samo ako se navede u parametru action oznaka <form> HTML-a. Podaci o greškama poslani su kroz komandnu liniju. Ključna funkcija isset() provjerava je li varijabla postavljena. To u kodu izgleda ovako:

if (isset ($\_GET['error'])){

# provjera koja je greška poslana.

}

U drugoj skripti s kodom PHP-a, osim raznih provjera koje su slične provjerama obrađenim u skripti za prijavu, korištena je funkcija header() kako bi se znalo o kojoj je greški riječ. Navedena funkcija vraća zaglavlje HTTP nazad u pretraživač. Unutar funkcije header() definirana je lokacija na koju treba otići te ime greške koje je korisnik napravio.

Još valja napomenuti i korištenje autentikacije CAPTCHA (engl. Completely Automated Public Turing test to tell Computers and Humans Apart). To je vrsta autentikacije koju koristi računarstvo da bi ustanovilo je li korisnik čovjek ili računalo, s ciljem sprječavanja pristupa zlonamjernim programima. ReCAPTCHA je preuzeta s Google servisa koji generira dva potrebna ključa i predstavlja korake potrebne za postavljanje.

Osim registracije i prijave, zaglavlje stranice sadrži i tražilicu. Tražilica je kreirana u zasebnoj skripti koja dva puta generira upit SQL, jedanput da se prikaže broj rezultata nastao pretraživanjem, a drugi put da se izlistaju oglasi koji su rezultat pretrage. Kod s jednim od upita izgleda na sljedeći način:

if (isset($\_POST['searchBut'])) {

$search = mysqli\_real\_escape\_string($conn, $\_POST['search']);

$sql = "SELECT a.id\_oglas, a.id\_kategorija, a.naslov, b.naziv, CONCAT(SUBSTRING\_INDEX(tekst,' ' ,3), '...') as tekst FROM oglas AS a INNER JOIN( SELECT id\_kategorija, naziv FROM kategorija) AS b ON a.id\_kategorija = b.id\_kategorija WHERE aktivan = 1 AND naslov LIKE '%".$search."%' OR tekst LIKE '%".$search."%';";

$result = mysqli\_query($conn, $sql);

$resultCheck = mysqli\_num\_rows($result);

if ($resultCheck > 0) {

while ($row = mysqli\_fetch\_assoc($result)) {

echo "<li><a href='oglas.php?id=" .$row['id\_oglas']."'>

<i class='pin'></i>

<h2>" .$row['naslov']. "</h2>

<p style='font-style: italic'>" .$row['naziv']."</p>

" .$row['tekst']. "

<p>Pogledaj oglas</p></a></li>";

}

}

echo <h3 id=obav> Nema rezultata!</h3>";

}

Prije samog upita SQL, provjerava se je li korisnik upisao tip podatka string uz pomoć funkcija mysqli\_real\_escape\_string(). Navedena funkcija je standardna funkcija za izbjegavanje znakovnih simbola, što je jednostavan način zaštite od napada umetanjem koda SQL.

Upit sadrži funkciju CONCAT koja vrši spajanje stringova, zatim funkciju SUBSTRING\_INDEX koja u ovom slučaju traži razmak unutar teksta ili naslova oglasa i na kraju dodaje tri točke. Uvjet unutar upita je, naravno, da se izlistaju samo oglasi koji su trenutno aktivni, a drugi je uvjet da naslov ili tekst oglasa sadrži ključnu riječ koju je korisnik upisao u tražilicu.

Upit SQL provodi se kroz funkciju mysqli\_query() koja za parametre prima varijablu koja je spremljena od prije za povezivanje na bazu i varijablu u koju je spremljen upit na bazu podataka. Nakon što je rezultat navedene funkcije spremljen u zasebnu varijablu $result, poziva se funkcija mysqli\_num\_rows() koja kao rezultat daje broj redaka iz tablice. I konačno se u varijablu $row spremaju rezultati funkcije mysqli\_fetch\_assoc() koja vraća asocijativni niz koji odgovara dohvaćenom retku ili vraća NULL, ako više nema redaka. I na taj se način mogu ispisati određeni podaci iz tablice kao što je $row['naziv'] ili $row['tekst'].

#### **4.3.1.1. Elementi zaglavlja za registriranog korisnika**

Nadalje, kod prijavljenog korisnika, poveznica moji oglasi je skripta oglasi\_ulogiranog\_korisnika.php koja ispisuje dvije liste kroz dva SQL-upita. Prva lista je lista aktivnih oglasa koje je kreirao korisnik, a druga su oglasi koji su istekli ili čekaju na odobrenje administratora. Klikom na poveznicu određenog oglasa, korisnika šalje na skriptu uredi\_oglas.php koja će biti obrađena kasnije u tekstu. Osim korisnikovih oglasa, korisnik ima pristup spremljenim oglasima, odnosno listi korisnikovih favorita. Ovakvu listu korisnik sam kreira klikom na poveznicu koja se nalazi na svakom oglasu. Služi mu da spremi oglase koji ga zanimaju, a kreirali su ih drugi korisnici. Lista spremljenih oglasa sprema se u posebnu tablicu u bazi, a podaci koji se spremaju su:

* id – predstavlja primarni ključ tablice,
* id\_korisnik – identifikacijski broj korisnika koji je spremio oglase i
* id\_oglas – identifikacijski broj oglasa koji je korisnik spremio.

Navedenu listu ispisuje skripta favoriti.php uz pomoć upita SQL koji tablici *oglas* pridružuje tablicu spremljeni\_oglasi. Osim identifikacijskog broja oglasa, dohvaća se naslov i datum kad je objavljen oglas. Bilo koji od spremljenih oglasa može se izbrisati iz navedene liste jednostavnim klikom na poveznicu „Ukloni favorita“. Brisanje favorita iz liste obavlja se u skripti ukloni\_favorita.php. Korišteni upit SQL izgleda na sljedeći način:

DELETE FROM spremljeni\_oglasi WHERE id\_korisnik =".$idk." AND id\_oglas =".$ido." ;

Kako bi se izbjeglo brisanje krivog oglasa, ili pak svih oglasa od istog korisnika, osim identifikacijskog broja korisnika potrebno je navesti kao uvjet brisanja i identifikacijski broj oglasa. Između ova dva uvjeta postavlja se funkcija AND što znači da oba uvjeta moraju biti ispunjena.

Korisnik također ima i pristup svom profilu, odnosno uređivanju svojih podataka, što mu omogućuje skripta uredi\_profil.php. Podaci koje prijavljeni korisnik može uređivati na svom profilu su sljedeći:

* korisničko ime,
* ime i prezime,
* kontakt broj,
* elektroničku poštu,
* profilnu sliku,
* resetiranje lozinke i
* brisanje vlastitog profila.

Tekstualni dio podataka ažurira se upitom UPDATE na bazu podataka, a kod manipuliranja profilnom slikom drugačije se postupa. Provjerava se je li korisnik selektirao sliku za učitavanje sljedećim kodom:

if($\_FILES["slika"]["error"] == 4) {

#slijedi UPDATE upit.

}

Navedeni izraz koristi se kod provjere je li određena učitana datoteka u ovom slučaju slika. Ako je rezultat greške broj 4, to znači da datoteka nije učitana i upisuje se samo tekstualni dio koji je korisnik popunio. U slučaju da je korisnik učitao sliku, provodi se još niz provjera, između ostalog i ima li korisnik već od prije spremljenu sliku u bazi i u mapi na poslužitelju. Takva provjera izgleda ovako:

$sqlST = "SELECT slika FROM kategorija;";

$result = mysqli\_query($conn, $sqlST);

$resultCheck = mysqli\_num\_rows($result);

if ($resultCheck > 0) {

while ($row = mysqli\_fetch\_assoc($result)){

if($row['slika'] === 1){

#slijedi brisanje postojeće slike, zatim učitavanje nove.

}

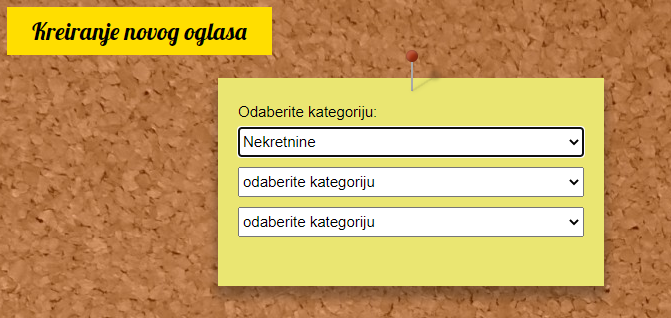
}

}

Ostale provjere su iste kao i kod skripte za dodavanje oglasa koja će biti obrađena u nastavku.

Nakon profilne slike, korisnik ima mogućnost resetiranja lozinke. Ovaj postupak je već objašnjen. Iznimka je da korisnik odmah poveznicom dolazi do skripte resetiranje\_lozinke.php.

Kod korisničkog profila valja još napomenuti poveznicu svi korisnici koja vodi na listu svih korisnika koji su registrirani na Oglasniku. Navedena poveznica je korisna, ako korisnik ima određenog korisnika kojeg prati, odnosno njegove oglase. Lista korisnika je vidljiva isključivo prijavljenim korisnicima. Običan prijavljeni korisnik, selektirajući određenog korisnika, može vidjeti listu njegovih oglasa, dok administrator može uređivati korisnike.

Poslije korisnikovih oglasa slijedi poveznica predaj oglas koja pokreće skriptu odaberi\_kategoriju.php i korisnik prvo ima mogućnosti odabrati kategoriju za koju želi kreirati oglas. To je ostvareno pomoću tri selektivne opcije, jer Oglasnik sadrži tri nivoa kategorija i potkategorija. Ovisno o tome ima li kategorija potkategoriju prikazuje mu se gumb nastavi. Dakle, ako je selektirao kategoriju koja nema potkategorija, može poveznicom nastaviti, ako je pak selektirao kategoriju koja ima potkategoriju, mora dalje selektirati potkategoriju i tako još jednom za treći nivo. Slijedi prikaz odabira kategorija na slici 8 i slici 9.

Slika 8: Selektiranje kategorije u kojoj nije odabrana potkategorija

Kao što je i vidljivo na slici 8, korisniku se nije pojavila poveznica da nastavi s izradom oglasa jer kategorija nekretnine sadrži i potkategorije od kojih se jedna treba odabrati da bi se nastavilo s kreiranjem oglasa.

Slika 9: Selektiranje kategorije u kojoj je odabran zadnji nivo potkategorije.

Na slici 9 vidljivo je da je odabir kategorije završen, jer potkategorija Kuće ne sadrži svoju potkategoriju. U suprotnom, poveznica nastavi koja vodi na skriptu dodaj\_oglas.php ne bi bila aktivna i selektiranje kategorije ne bi bilo izvršeno.

Nakon odabira kategorije moguće je kreirati i oglas što je ostvareno pomoću skripte dodaj\_oglas.php. Skripta za odabir kategorije šalje skripti dodaj\_oglas.php sve identifikacijske brojeve kategorije potrebne za izradu oglasa putem metode $\_GET. U skripti za kreiranje oglasa osim forme za upis podataka i provjere je li korisnik kliknuo gumb zalijepi oglas koji započinje upisivanje svih podataka u tablicu oglas, još je obrađeno nekoliko ključnih stavki.

Potrebna je inicijalizacija varijable $count koja uz pomoć funkcija array\_filter() i count() broji koliko je slika korisnik odabrao za kreiranje svog oglasa. Slijedi prikaz koda u nastavku:

$count = count(array\_filter($\_FILES['files']['name']));

Limit slika postavljen je na pet. U slučaju da je korisnik selektirao više od pet slika, web aplikacija mu ispisuje grešku. Ukoliko korisnik ne označi niti jednu sliku za učitavanje u web stranicu, provjerava se kategorija, odnosno potkategorija za njegov oglas, što je vidljivo u sljedećem kodu:

if ($count > 5) {

header("Location: dodaj\_oglas.php?error = br\_slika&id = ".$id\_kat."&pod\_id = ".$id\_pod."&pod\_id2 = ".$id\_pod2);

}

elseif ($count == 0 || $count == NULL) {

if(($id\_pod2 == 0) && ($id\_pod == 0)){

#kategorija nema niti jednu potkategoriju, slijedi upis tekstualnog sadržaja oglasa i slanje e-mail poruke administratoru.

}elseif($id\_pod2 == 0){

#kategorija nema drugu potkategoriju, slijedi upis tekstualnog sadržaja oglasa i slanje e-mail poruke administratoru.

}else{

# obje potkategorije sadrže vrijednosti, slijedi upis tekstualnog sadržaja oglasa i slanje e-mail poruke administratoru.

}

}else{

#varijabla $count je veća od 1 i manja ili jednaka 5,

#slijede ponovo provjere potkategorija,

#upis tekstualnog i multimedijskog dijela oglasa i slanje e-mail poruke administratoru.

}

Ovisno o broju identifikacijskih brojeva kategorija proslijeđenih skripti izvršava se određena naredba if. Ako je pak broj odabranih slika od korisnika zadovoljavajući, slijedi ponovna provjera s kategorijama, a nakon toga upis u bazu podataka.

Prvo je potrebno upisati samo tekstualni dio oglasa, a nakon toga se upisuju slike s istim identifikacijskim brojem oglasa kao i zadnji upisani oglas. Za učitavanje slika u direktorij i spremanje putanje u bazu podataka potrebne su određene varijable, odnosno podaci kao što je privremeno ime slike, ime slike te tip slike. Postavljen je kriterij da ekstenzija slike može biti samo .jpg, .jpeg ili .png. Za provjeru koju ekstenziju sadrži slika, prvo je potrebna funkcija explode() kojom se odvaja ime slike i njena ekstenzija, koje su razdvojene točkom. Nakon toga postoji mogućnost da je ekstenzija napisana velikim slovima i zato se koristi funkcija strtotower() kako bi ekstenzija sa sigurnošću sadržala mala slova. Lista dozvoljenih ekstenzija sprema se u niz i kreirani niz se provjerava funkcijom in\_array(). Ukoliko slika ima nedozvoljenu ekstenziju, javlja se greška. Ako je ekstenzija u redu, slika dobiva novo unikatno ime pomoću funkcije uniqid() koja kreira jedinstveni niz znakova. Na ovaj je način izbjegnut slučaj istoimene slike što bi uzrokovalo brisanje podataka.

Nakon dodjele imena slici, kreirana je varijabla koja predstavlja putanju na direktoriju gdje će slika biti pohranjena. Slika na kraju, ali prije jedinstvenog imena, sadrži i jedinstveni broj oglasa.

Prije samog spremanja slike u određeni direktorij treba kreirati mapu u kojoj će slika biti spremljena s funkcijom mkdir(), no funkcija file\_exists() provjerava postoji li već takva mapa u direktoriju. Potom slijedi spremanje slike u novo kreiranu mapu pomoću funkcije move\_uploaded\_file().

Putanja direktorija slike također je spremljena u bazu podataka INSERT SQL upitom. Spremanjem oglasa i slike oglasa u bazu slijedi slanje elektroničke poruke administratoru da postoji novi neodobreni oglas koji treba odobriti. Slanje elektroničke poruke obrađeno je u sljedećem poglavlju. A sve gore navedene funkcije u kodu koriste se na sljedeći način:

$id\_oglas = $row['max(id\_oglas)'];

foreach($\_FILES["files"]["tmp\_name"] as $key=>$tmp\_name) {

$tmp\_slika = $\_FILES["files"]["tmp\_name"][$key];

$ime\_slika = $\_FILES["files"]["name"][$key];

$tip\_slika = $\_FILES["files"]["type"][$key];

$slExt = explode('.', $ime\_slika);

$slActualExt = strtolower(end($slExt));

$allowed = array('jpg', 'jpeg', 'png'

if (in\_array($slActualExt, $allowed)) {

$slNameNew = uniqid('', true) .".".$slActualExt;

$putanja = "uploads/oglasi/slike\_oglasa\_".$id\_oglas;

$putanja\_slikaCheck = "uploads/oglasi/slike\_oglasa\_" .$id\_oglas."/".$slNameNew;

if (!file\_exists($putanja\_slikaCheck)) {

mkdir($putanja, 0777, true);

$putanja\_slika = "uploads/oglasi/slike\_oglasa\_" .$id\_oglas."/".$slNameNew;

move\_uploaded\_file($tmp\_slika, $putanja\_slika);

$sqlSL = "INSERT INTO slike\_oglasa (id\_oglas, ime\_slika, tip\_slika, putanja\_slika) VALUES ('$id\_oglas', '$slNameNew', '$tip\_slika', '$putanja\_slika');";

$resultSl = mysqli\_query($conn, $sqlSL);

header("Location: oglasi\_ulogiranog\_korisnika.php?uploadsuccess");

}

}else{

echo"<h2 class='greska'>Format slike nije dozvoljen!</h1>";

echo count(array\_filter($\_FILES['files']['name']));

}

}

Kao što je ranije spomenuto, korisnik može i uređivati svoj kreirani oglas. Skripta uredi\_oglas.php ima dosta zajedničkog s prijašnjom obrađenom skriptom. Naravno, korisnik u svom oglasu može uređivati:

* naslov oglasa,
* tekst oglasa,
* kontakt broj oglasa,
* dodavati slike, ako oglas sadrži manje od pet slika,
* brisati slike pojedinačno,
* aktivirati svoj oglas, ako je istekao,
* deaktivirati oglas,
* obrisati oglas i
* promovirati oglas.

Rad sa slikama je vrlo sličan kao u skripti dodaj\_oglas.php s tim da se najprije provjerava sadrži li oglas već slike i ako ih sadrži, koji je broj već dodanih slika. Broj već dodanih slika i dodavanje novih slika obrađeno je naredbom switch. U slučaju da oglas nema već dodanih slika, korisnik može selektirati maksimalno pet slika. Ako oglas već sadrži jednu sliku, onda korisnik može selektirati maksimalno četiri slike itd.

Brisanje pojedinačnih slika ostvareno je zasebnom skriptom brisanje\_slika\_oglasa.php. Prosljeđuju se identifikacijski broj slike i oglasa kao i putanja slike već spomenutom metodom GET. Nakon brisanja slike iz baze, tj. putanje SQL upitom DELETE, pozvana je funkcija unlink() koja briše datoteku, a kao parametar prima samo putanju datoteke.

Aktiviranje, deaktiviranje i brisanje oglasa postiže se pozivom upita SQL, izuzev što kod brisanja oglasa treba provesti i poseban upit SQL na tablicu sa slikama oglasa i nakon toga se briše mapa s postavljenom putanjom.

Promoviranje oglasa, ako oglas već nije plaćen, moguće je klikom na gumb promoviraj oglas. Gumb korisnika vodi na izbor načina plaćanja, općom uplatnicom ili kartičnim plaćanjem (Stripe[[9]](#footnote-9)). Odabirom plaćanja općom uplatnicom korisnik dobiva podatke koji su mu potrebni da bi ispunio uplatnicu i uplatio navedeni iznos. Dok odabirom kartičnog plaćanja, korisnik je obvezan upisati potrebne podatke o sebi i o kartici koju koristi.

Naplata sredstava obrađena je u skripti charge.php. Da bi se koristio određeni sustav za naplaćivanje putem interneta kao što je Stripe, potrebno je registrirati se na njihovu stranicu. Registracijom se dobije uvid u tablicu naplata i tablicu kupaca, također je generiran jedinstveni ključ koji je korišten u navedenoj skripti. Ključni podaci za provođenje naplate ovdje su:

* identifikacijski broj oglasa (id\_oglas),
* naslov oglasa, tj. proizvod koji korisnik plaća,
* elektronička pošta korisnika koji vrši uplatu i
* znak (engl. token), tj. broj transakcije.

Ti podaci su grupirani u dva asocijativna niza sukladno tablicama naplata i kupaca kao što je prikazano u sljedećem kodu:

$customer = \Stripe\Customer::create(array(

"email" =>$email,

"source" => $token

));

$charge = \Stripe\Charge::create(array(

"amount" => 1500,

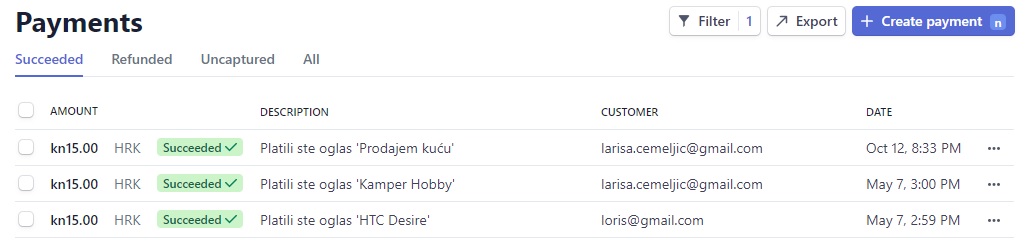
"currency" => "HRK",

"description" => "Platili ste oglas '".$naslov\_ogl."'",

"customer" => $customer->id

));

Prvi niz sadrži podatke potrebne za tablicu kupaca, a drugi za tablicu naplata. Izgled podataka s uspješnim transakcijama prikazan je na Slici 10.



Slika 10: Tablica uspješnih transakcija u Stripe-u

Podaci vidljivi iz tablice naplata su iznos koji je korisnik uplatio, koji je oglas platio, koja je njegova elektronička adresa i datum uplate.

Nakon što je naplata provedena uspješno, korisnik prima elektroničku poruku s identifikacijskim brojem transakcije koji isključivo samo taj korisnik može vidjeti, a služi mu kao dokaz o plaćanju oglasa. To je provedeno u skripti success.php. Osim slanja elektroničke poruke, postavljen je upit UPDATE na bazu gdje oglas s tim identifikacijskim brojem postaje tip oglasa 1, odnosno plaćeni oglas.

Ostaje samo još jedna mogućnost korisnika, a to je odjaviti se s web stranice. Kod skripte odjava.inc.php pozvano je niz funkcija prikazano dalje u kodu:

session\_start();

session\_unset();

session\_destroy();

header("Location: ../kategorije.php");

Dakle, nakon što je postavljena sesija da bi se uklonile sve kreirane globalne varijable i izbrisala sesija, potrebno je pozvati funkcije session\_unset() i session\_destroy();

I posljednje, što ima mogućnost samo administrator web stranice, uređivanje je korisnika. Pod uređivanje korisnika misli se na promjenu uloge korisnika na Oglasniku ili pak brisanje korisnika u potpunosti. Slika 11 prikazuje jedno takvo uređivanje:



Slika 11: Administrativno uređivanje korisnika

Na prethodnoj slici vidljivo je da administrator ima uvid u sve korisnikove podatke. Administrator može urediti podatak je li taj korisnik administrator ili obični registrirani korisnik. Također ima mogućnost brisanja korisnika iz baze podataka i time korisnik više nema pristup web stranici. Za ovakvo ažuriranje korišten je radio gumb i to na sljedeći način:

<form action='uredi\_korisnika?id=" .$row['id\_korisnik']."' method='post'>

<input type='hidden' name='id' value=". $row['id\_korisnik'].">

<input type='radio' name='id\_tip' value='0'". ( $row['id\_tip'] == '0' ? ' checked="checked"':''). "/> Administrator<br>

<input type='radio' name='id\_tip' value='1'". ( $row['id\_tip'] == '1' ? ' checked="checked"':''). "/> Obični korisnik<br>

<button id='urediKor' type='submit' name='urediKor'> Spremi promjene </button>

<button id='delKor' type='submit' name='delKor'> Obriši korisnika </button></form></div>

U prvoj linija koda u kojoj je otvorena oznaka form, vidljivo je da će stranica uredi\_korisnika.php biti ujedno i stranica na koju se šalju podaci. Podatak koji se šalje metodom POST je identifikacijski broj korisnika koji se uređuje. U prvom ulaznom podatku obrasca, identifikacijski broj je skriveno polje koje je potrebno za provođenje forme i nije bitan podatak običnom korisniku. Nakon toga su dva radio gumba od kojih je jedan označen ovisno o kojoj se ulozi korisnika radi. I za kraj postavljena su dva gumba, jedan koji sprema promjene, a drugi briše korisnika iz baze.

### 4.3.2. Tijelo web stranice

Naslovi i podnaslovi stranice pojavljuju se ovisno o tome na koju je poveznicu korisnik kliknuo. Njihov ispis je ostvaren na istom principu kao i lista oglasa, pa i tako početna stranica koja ima naslov kategorije. U listi kategorija ispisana su tri upita koja ovise ponovo o tipu korisnika koji učitava web stranicu. Osim anonimnog i prijavljenog korisnika postoje i administratori koji mogu upravljati dijelovima web stanice kojima obični prijavljeni korisnik ne može, osim ako ne ostvari to pravo.

Dakle, administrator na Oglasniku pod naslovom kategorije može urediti kategoriju i dodati kategoriju. Isto tako, ako uređuje određenu kategoriju, može joj dodati ili urediti njenu potkategoriju i tako za još jedan nivo kao što je vidljivo na slici 12.

A picture containing text, businesscard

Description automatically generated

Slika 12: Prikaz skripte uredi\_kategoriju.php i njene mogućnosti

Pod uređivanjem kategorija ili potkategorija podrazumijeva se naziv i slika. Slika se može u potpunosti izbrisati, onda se automatski postavlja slika koja je zadana kodom, ili se pak učitava sasvim nova slika koja će predstavljati ikonu zadane kategorije ili potkategorije. Slijedi prikaz u kodu:

if($row['slika'] == 0){

echo "<p><img class='slProf1' src='uploads/kategorije/kategorija.png' /></p>";

}else{

$imeSL = "uploads/kategorije/kategorija".$kat."\*";

$infoSL = glob($imeSL);

$extSL = explode(".", $infoSL[0]);

$actualextSL = $extSL[1];

echo"<p><img class='slProf1' src = 'uploads/kategorije/kategorija". $kat .".". $actualextSL ."'/></p>";

}

Prikaz je postignut naredbom if. U prvom slučaju, ako slika nije zadana od strane administratora i ako je u bazi podataka pod zadanom kategorijom u stupcu slika, vrijednost je 0, ispisuje se slika koja je na URL-u definirana atributom src. Dok u drugom slučaju ime slike, odnosno URL ovisi o identifikacijskom broju kategorije i ekstenziji slike, tj. vrsti slike koju je administrator učitao. Sukladno tome, najprije je korištena funkcija glob() koja vraća niz imena datoteka ili direktorija, zatim funkcija explode() koja rezultat prijašnje funkcije pretvara u niz i razdvaja članove niza koji su spojeni točkom. Znači, ako je identifikacijski broj kategorije na primjer 14, rezultat druge funkcije će biti:

Array ( [0] => uploads/kategorije/kategorija14 [1] => png )

Što je zapravo i cilj, odvojiti ekstenziju od ostatka putanje kako bi se ispravna ekstenzija slike dodala na kraju imena slike.

Najčešće prikazana lista je lista oglasa. Iako to može značiti više lista, glavna je lista svih aktivnih oglasa kojoj se može pristupiti pomoću navigacijske trake u zaglavlju web stranice. Izgled takve liste prikazan je na slici 13.

A picture containing timeline

Description automatically generated

Slika 13: Prikaz svih aktivnih oglasa

Primarni prikaz ove skripte prvo je lista plaćenih oglasa koji su označeni žutom pozadinskom bojom u naslovu, a zatim onih koji su besplatno kreirani. Lista je također sortirana prema datumu objave oglasa, dakle oglas koji je najkasnije dodan, bit će prvi na listi. Ovo je postavljeno uz pomoć naredbe SQL ORDER BY koja služi za sortiranje vrijednosti:

* uzlazno – ASC (engl. **Ascedenting**) npr. od 1 do 10 i
* silazno – DESC (engl. Descedenting), npr. od 10 do 1.

U ovom slučaju korišteno je silazno sortiranje vrijednosti (DESC) i to na sljedeći način:

SELECT id\_oglas, b.id\_kategorija, b.naziv, id\_tip\_oglas, naslov, aktivan, vrijedi\_do, DATE\_FORMAT(vrijedi\_od, '%Y-%m-%d') AS datum FROM oglas AS a INNER JOIN(

SELECT id\_kategorija, naziv FROM kategorija) AS b ON a.id\_kategorija=b.id\_kategorija

WHERE aktivan='1' AND id\_tip\_oglas = '2' ORDER BY datum DESC

U kodu je vidljivo da je atribut vrijedi\_do preimenovan u datum zbog lakšeg korištenja u daljnjem kodu. Također se koristila funkcija DATE\_FORMAT koja oblikuje zadani datum. Navedena funkcija korištena je kako bi se lakše izračunalo i ispisalo kada je objavljen oglas. Kod korišten za ispis takve informacije izgleda ovako:

$datum = $row['datum'];

$datum = strtotime($datum);

$danas = strtotime(date("Y-m-d"));

$datediff = $danas - $datum ;

$rezultat = round($datediff / 86400);

if ($rezultat == 1){

echo "jučer</p>";

}elseif($rezultat == 0){

echo "danas</p>";

}else{

echo "prije ". $rezultat." dana</p>";

}

U prvoj liniji koda redak dobiven iz tablice inicijaliziran je na varijablu $datum. Nakon toga je pozvana funkcija strtotime() koja vraća tekstualni datum, što znači ako je 2020-11-10 datum kada je oglas objavljen, rezultat ove funkcije bi vratio vrijednost 1604962800. Ovakav rezultat također je potreban za trenutni datum i to je provedeno na sljedećoj liniji koda. Za razliku od datuma kada je objavljen oglas, sada se traži današnji datum, a to omogućuje funkcija date() koja vraća formirani datum po lokalnom vremenu. Zatim su oduzete te dobivene vrijednosti i podijeljene s brojem 86 400 što je umnožak sekundi, minuta i sati (60\*60\*24). Kako rezultat ne bi sadržao decimale, korištena je funkcija round() koja vraća zaokruženu vrijednost. I na koncu ispisuje se rezultat izračuna pomoću naredbe if. Ako je rezultat jedan dan ispisuje se „jučer“, ako je jednak nuli ispisuje se „danas“ i za ostale rezultate ispisuje se „prije“, zatim vrijednost varijable $rezultat i string „dana“.

Skripta svi\_oglasi sadrži i formu sa selektivnom opcijom kako bi korisnik mogao pogledati samo oglase koji su plaćeni ili samo besplatne oglase i povratak na sve aktivne oglase. Ako je prijavljen administrator, onda mu je vidljiv gumb „odobri oglase“ koji ga vodi do skripte odobri\_oglase.php koja sadrži listu neodobrenih oglasa. Ako su pak svi oglasi odobreni, onda je administratoru vidljiva ta obavijest.

U navedenoj skripti napisana je funkcija koja deaktivira oglase kada oni isteknu i to na sljedeći način:

function deaktivacijaOglasa ( $vrijedi\_do ){

$danas = strtotime(date("Y-m-d H:i:s"));

$razlika = $danas - strtotime($vrijedi\_do);

if($razlika > 1){

return true;

}

}

Funkcija za deaktivaciju oglasa prima jedan parametar, a to je datum do kada pojedini oglas vrijedi, što je zapisano u bazi pod tablicom oglas. Unutar funkcije pozvana je funkcija date() koja vraća formirani niz datuma po lokalnom vremenu. To lokalno vrijeme funkcija strtotime() vraća kao tekstualni datum i vrijeme. Dakle, za trenutno vrijeme koje je sada recimo 2020-10-08 19:10:27 rezultat bi bio sljedeći: 1602177027. Ista funkcija se poziva s parametrom koji je datum do kada vrijedi oglas i ako je razlika između današnjeg datuma i datuma do kad vrijedi oglas veća od 1, onda kreirana funkcija vraća istinu, tj. oglas je istekao. Za taj oglas to znači da je u bazi status aktivan, odnosno mijenja se 1 u 2. Naravno, kada određeni oglas istekne, web aplikacija šalje elektroničku poruku korisniku čiji je oglas s informacijom koji mu je oglas istekao i poveznicom da isti oglas obnovi, tj. produži.

Sljedeća stavka na kojoj administrator ima prostora za manipuliranjem podacima je upravo u skripti svi\_oglasi.php. Njegov prikaz skripte je vrlo sličan anonimnom i prijavljenom korisniku, osim jednog gumba, a taj gumb ga vodi do liste neodobrenih oglasa. Lista neodobrenih oglasa je lista s oglasima koji nisu aktivni, što znači da je određeni korisnik dodao oglas koji još nije odobren ili je korisnik zatražio produljenje oglasa koje mu još nije odobreno.

Kada administrator klikne na određeni neodobreni oglas, prikaz tog oglasa mu je gotovo isti kao i kod anonimnog ili kod prijavljenog korisnika. Jedina razlika je u opciji odobravanja oglasa i brisanja istog u slučaju neprimjerenog sadržaja. Klikom na opciju odobri oglas, status oglasa se ažurira u tablici na vrijednost 1. Dok su u slučaju brisanja oglasa pozvana dva upita SQL sukcesivno i to s funkcijom mysqli\_multi\_query(). Naravno, zbog primarnog i sekundarnog ključa u tablicama oglas i slike oglasa, nije moguće prvo obrisati podatke iz primarne tablice u kojoj je primarni ključ, već se prvo poziva upit sa sekundarnim ključem koji je identifikacijski broj oglasa, potom se brišu podaci iz primarne tablice.

Prikaz oglasa kod anonimnog i prijavljenog korisnika ispisuje im sljedeće informacije i funkcije:

* opcija ispisa oglasa,
* opcija spremanja oglasa u favorite,
* naslov oglasa,
* opis, odnosno tekst oglasa,
* kontakt broj koji je korisnik dodao prilikom dodavanja oglasa,
* kontakt broj koji je korisnik upisao u svoj profil,
* poveznicu na ostale oglase koje je isti korisnik kreirao,
* poveznicu na formu za slanje elektroničke poruke u kojoj se direktno može obratiti korisniku čiji je oglas,
* datum do kada je oglas aktivan i
* broj koliko je puta oglas pogledan.

Opcije koje administrator ima još su uređivanje korisnika i brisanje oglasa. Prikaz jednog oglasa vidljiv je na slici 14.



Slika 14: Administrativni prikaz odobrenog oglasa

Prikaz oglasa podijeljen je u dva bloka. Prvi, s lijeve strane prikazuje tekstualni dio oglasa, a drugi prikazuje slike koje je korisnik učitao za navedeni oglas.

U prvom bloku najprije je poveznica, tj. slika pisača. Kada korisnik na nju klikne, otvara se dodatni prozor u kojem se korisniku nude određene opcije za ispis sadržaja. Sadržaj koji nije potrebno ispisati, definiran je u kodu na sljedeći način:

<style>

@media print{

.printer, .poruka, .ostalo, .favorit, .korisnik, #DelBut, footer, .subfooter {

display:none;

}

.ispis, .ispis\*{

visibility:visible;

}

<style>

Pravilo @media omogućava definiranje različitih stilova za različite medije. U ovom slučaju postavljen je jedan skup stilskih pravila za pisače, a drugi za računalne zaslone. Pa tako kod ispisa nije vidljivo sljedeće:

* slika poveznice na ispis (klasa printer),
* gumb za slanje elektroničke poruke korisniku (klasa poruka),
* poveznica na ostale oglase od istog korisnika (klasa ostalo),
* slika poveznice koja sprema oglas u listu favorita (klasa favorit),
* poveznica za uređivanje korisnika (klasa korisnik),
* gumb za brisanje oglasa (id DelBut),
* podnožje web stranice i
* poveznica na pravila o privatnosti i kolačićima (klasa subfooter).

Svi ostali podaci kao što su naslov oglasa, opis oglasa itd. nalaze se pod klasom ispis i oni su vidljivi kod ispisa oglasa.

Pored naslova oglasa nalazi se slika zvjezdice koja sadrži poveznicu te pokreće skriptu spremanje\_oglasa.php koja sprema identifikacijski broj korisnika i oglasa u tablicu spremljeni\_oglasi. Navedena tablica sadrži liste favorita određenih korisnika.

Većina navedenog su podaci selektirani iz baze podataka, osim poveznice na ostale oglase koje je korisnik kreirao. Ova poveznica vodi do skripte oglasi\_ulogiranog\_korisnika.php i to je jednostavna lista koje je već obrađena prethodno u tekstu. Poveznica na formu za elektroničku poruku služi korisniku za pribavljanje dodatnih informacija o navedenom oglasu od drugog korisnika. Forma automatski popunjava ime primatelja i naslov poruke, tj. o kojem je oglasu riječ. Korisnik koji šalje poruku mora upisati svoje ime, svoju elektroničku adresu, tekst poruke koju šalje i potvrditi da nije robot, što je već spominjana reCAPTCHA u poglavlju Zaglavlje kod registracije korisnika.

Skripta za slanje poruke, osim forme, sadrži i obavijesti o greškama kao što to čini i skripta za registraciju. Za uspješno slanje elektroničke poruke, kreirana je i skripta contact\_send.php u kojoj se provodi slanje elektroničke poruke. Prije svega, potrebno je metodom slanja podataka $\_POST prikupiti podatke koje je korisnik upisao putem forme kao što su ime, elektronička pošta, tekst poruke i ime oglasa. Zatim se iz baze dohvaća elektronička pošta korisnika koji prima poruku, za to se stavlja kriterij identifikacijski broj korisnika.

Za slanje poruke elektroničkom poštom koristi se funkcija mail() koja prima parametre:

* varijabla koja sadrži elektroničku poštu na koju se šalje poruka,
* tema poruke – koji u ovom slučaju sadrži ime pošiljatelja i naslov oglasa u vezi kojeg je poslana poruka elektroničkom poštom,
* tijelo poruke – može se sastojati od više redaka, u ovom slučaju opet ispisuje ime pošiljatelja, njegovu elektroničku poštu i tekst poruke i
* dodatna zaglavlja koja nisu obavezna kao From, Cc i Bcc.

U nastavku slijedi kod za slanje elektroničke poruke korištenjem funkcije mail().

$sql = "SELECT email FROM korisnik WHERE id\_korisnik=" .$idk. ";" ;

$result = mysqli\_query($conn,$sql);

while($row = mysqli\_fetch\_array($result)){

$mail\_to = $row['email'];

$subject = 'Nova poruka od '.$ime.' za oglas "'.$imeOglasa.'"';

$body\_message = 'Poruku poslao: '.$ime. "\n";

$body\_message .= 'E-mail: '.$email."\n";

$body\_message .= 'Poruka: '.$poruka;

$headers = 'From: '.$email."\r\n";

$headers .= 'Reply-To: '.$row['email']."\r\n";

$mail\_status = mail($mail\_to, $subject, $body\_message, $headers);

header("Location: svi\_oglasi.php");

}

Za broj koliko je puta oglas pogledan dodan je poseban stupac u tablici oglas pod nazivom br\_pogleda. Ovaj podatak ažurira se svaki put kad se prikaže oglas s određenim identifikacijskim brojem. Prikaz upita je u sljedećem kodu:

UPDATE `oglas` SET `br\_pogleda` = `br\_pogleda`+ '1' WHERE `id\_oglas` = ".$row['id\_oglas'].";

Kao što je vidljivo u kodu, vrijednosti koja sadrži broj pogleda pribraja se jedan i tako kod svakog otvaranja oglasa pod određenim identifikacijskim brojem oglasa. Naravno, nakon ažuriranja, kako bi se prikazao navedeni broj, treba se selektirati iz baze podataka pomoću upita SELECT.

### 4.3.3. Podnožje web stanice

Podnožje web stranice služi kao dodatna navigacija kako korisnik koji je došao do dna web stranice ne bi morao dužim putem vraćati se na vrh web stranice do bitnih elemenata.

Podnožje Oglasnika sastoji se od tri bloka. Prvi blok sadrži informacije o samoj tvrtki, kao što je korisnička podrška, kontakt broj i adresa elektroničke pošte, kao i radno vrijeme iste. Sljedeći blok sadrži malu navigaciju kako bi korisnik ponovo pregledao kategorije Oglasnika, njegove oglase i kako bi mogao predati oglas. U slučaju da se radi o neregistriranom korisniku, tj. korisniku koji nije prijavljen, klikom na poveznicu predaj oglas, poveznica ga vodi na registraciju. Ako se pak radi o korisniku koji je prijavljen, odmah po kliku može odabrati kategoriju za oglas koji želi objaviti. I, naravno, tu je blok s poveznicama na društvene mreže.

Osim navedenih poveznica, u podnožju se također nalaze pravila o privatnosti i pravila o kolačićima. Ova su pravila potrebna jer prikupljanje podataka bez obavještavanja korisnika o tome je zakonom kažnjivo. Cilj je ovih pravila obavijestiti korisnike o načinu na koji web stranice prikupljaju osobne podatke, kao i o mogućnostima pristanka na njihovu razmjenu.

Izgled podnožja Oglasnika prikazan je na slici 14.

A picture containing graphical user interface

Description automatically generated

Slika 14: Podnožje Oglasnika

# 5. ZAKLJUČAK

Za izradu web stranice potrebno je puno istraživanja i proučavanja kako bi se došlo do idealnih rješenja. Kako bi web stranica bila privlačna i funkcionalna, potrebno je dobro planiranje prije same izrade, od strukture baze podataka pa do elemenata same web stranice koji su obavezni na svakom web mjestu. Također je bitno vršiti testiranje i, naravno, uklanjanje pogrešaka nastalih razvojem web stranice. Svaka web stranica može svakodnevno napredovati, pa tako i ova.

Sve većom potrebom prodaje i kupnje putem interneta povećava se potreba za web oglasnicima. Oglasnik omogućuje internet korisnicima uspješnu razmjenu oglasa te uspostavu direktnog kontakta bez ugrožavanje prava o privatnosti podataka.

Ideja za web dizajn je oglasna ploča s papirićima pričvršćenih pribadačama. Korisničko je sučelje ove web stranice, kao i svake druge, kodiranje vizualnog dijela web stranice. Pristup kodiranju svakog pojedinog elementa potkrijepljen je slikom i kodom koji prikazuje cijeli proces izrade web stranice. Prilikom izrade web stranice stečene su vještine u korištenju internetskih tehnologija kao što su HTML koji je osmišljen radi prezentacije sadržaja i CSS koji je korišten za grafički dizajn web stranice koja ima prilagođen prikaz za različite uređaje kao što su mobiteli, računala, tableti itd.

Poslužiteljski dio web stranice zadužen je za korisniku „nevidljivi“ dio web stranice, odnosno funkcionalnost same stranice. Obuhvaćeni su zahvati na poslužiteljskoj strani, integraciji podataka i bazi podataka. Dakle, stvorene su komponente i značajke kojima korisnik neposredno pristupa. Također su osigurani podaci i usluge koje zahtijevaju korisničko sučelje.

Programski jezik PHP jedan je od najboljih izbora za izradu web stranica i jedan od najpopularnijih jezika poslužiteljskog sučelja koji je potpuno specijaliziran za razvoj na webu. Ovaj moćan jezik za pisanje skripta omogućava brzu izradu složenih web aplikacija.

MySQL je brz i pouzdan sistem za upravljanje bazama podataka koji je u kombinaciji s PHP-om pogodan za dinamičke aplikacije koje rade na internetu. Ova kombinacija internetskih tehnologija pruža široke mogućnosti za kreiranje bilo kakvih web stranica.

Web stranica izrađena je kao potpuno funkcionalna i cilj joj je demonstrirati poznavanje rada web tehnologija. Pretraga, dodavanje, uređivanje i plaćanje oglasa su među najvažnijim funkcijama na web stranici. Dorade su moguće i potrebne da bi se poboljšala učinkovitost, a neka od područja, gdje ima mjesta doradama, su bolja sigurnost pri komunikaciji između klijenta i poslužitelja te provjere korisnika. Da bi web stranica bila ljepša i ugodnija posjetiteljima, potrebne su estetske dorade na grafičkom dizajnu. Kako bi se postiglo veće povjerenje posjetitelja, potrebna je uspostava komunikacije s ostalim posjetiteljima na web stranici, što znači da bi posjetitelji, koji su koristili uslugu web stranice, ostavili svoj dojam i ocjenu kvalitete usluge.

# LITERATURA

[1] Lerderf R., Tatroe K. i Maclntyre P. (2009): Programiranje PHP

[2] Luka Abrus (2003): Izrada weba, abeceda za webmastere

[3] Elizabeth Castro (2004): Brzi vizualni vodič HTML za World Wide Web

[4] HTTP protokol - <https://hr.wikipedia.org/wiki/HTTP> (pristupljeno 08.05.2018.)

[5] <http://mreze.layer-x.com/s050100-0.html> (pristupljeno 08.05.2018.)

[6] HTML - <https://hr.wikipedia.org/wiki/HTML> (pristupljeno 07.01.2019.)

[7] <https://web.math.pmf.unizg.hr/~karaga/mrezev6.html> (pristupljeno 07.01.2019.)

[8] PHP - <https://wiki.php.net/> (pristupljeno 14.01.2019.)

[9]<https://www.plus.hr/blog/2017/03/30/prednosti-koristenja-php-verzije-7/>(pristupljeno 14.01.2019.)

[10] <https://wordpress.org/about/stats/> (pristupljeno 14.01.2019.)

[11] <https://w3techs.com/> (pristupljeno 14.01.2019.)

[12] <https://php.com.hr/88> (pristupljeno 14.01.2019.)

# SAŽETAK

Ovaj diplomski rad temelji se na znanju internetskih tehnologija potrebnih za uspješnu izradu web stranice. U teorijskom dijelu diplomskog rada obrađene su internetske tehnologije koje su potrebne za izradu jedne web stranice, a to su HTML, CSS, MySQL i PHP. U navedenoj obradi postavljeni su primjeri koda u praksi.

Način izrade web stranice objašnjen je u praktičnom dijelu, od strukture baze podataka do elemenata web stranice. Koraci i planiranja kroz izradu web oglasnika potkrijepljeni su slikama i primjerima koda.

**Ključne riječi:** PHP, HTML, CSS, MySQL, izrada web stranice, web oglasnik

# SUMMARY

This paper is based on knowledge of internet technologies needed for successful website design. The theoretical part of the thesis deals with the internet technologies needed to create a website, namely HTML, CSS, MySQL and PHP. In the mentioned processing, examples of code in practice are set.

The method of creating a website is explained in the practical part, from the structure of database, to the elements of the website. Steps and planning through web ad creation are supported by images and code examples.

**Keywords:** PHP, HTML, CSS, MySQL, web design, web advertisement

1. TCP – (Transmission Control Protocol) je prijenosni protokol interneta koji garantira pouzdanu isporuku podataka u kontrolnom redoslijedu od pošiljatelja prema primatelju. Ima mogućnost višestrukih istovremenih povezivanja prema jednoj aplikaciji više klijenata na jednom poslužitelju, najčešći primjer je poslužitelj e-pošte [↑](#footnote-ref-1)
2. Gateway – uređaj koji se nalazi u čvoru računalne mreže, koristi se za komuniciranje s drugom mrežom koja koristi drugačiji protokol [↑](#footnote-ref-2)
3. IIS je web poslužitelj kreiran od strane Microsofta [↑](#footnote-ref-3)
4. CMS (engl. Content management system) je sustav koji omogućuje upravljanje sadržajem [↑](#footnote-ref-4)
5. Apache – besplatan web poslužitelj [↑](#footnote-ref-5)
6. Open Source – znači da svatko može pogledati izvorni kod **raščlanjivača sintakse** zaduženog za izvršavanje PHP aplikacija, dodati neku svoju funkcionalnost i komunicirati s razvojnim timom, te im predložiti ideje za poboljšanja [↑](#footnote-ref-6)
7. Git – distribuirani sustav za upravljanje izvornim kodom [↑](#footnote-ref-7)
8. UTF-8 je način zapisa kodnih točaka pomoću nizova 8-bitnih bajtova u standardu Unicode [↑](#footnote-ref-8)
9. Stripe –omogućuje prihvaćanje različitih načina plaćanja putem interneta [↑](#footnote-ref-9)