

**ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU**



**VEB APLIKACIJA ZA IZBOR RESTORANA**

– Diplomski rad –

Kandidat:

Snežana Lilić 2004/0251

Mentor:

doc. dr Zoran Čiča

Beograd, Septembar 2015.

# SADRŽAJ

<b>SADRŽAJ</b> .....	<b>2</b>
<b>1. UVOD</b> .....	<b>3</b>
<b>2. OSNOVNI ALATI I PROGRAMSKI JEZICI KOJI SE KORISTE U TEZI</b> .....	<b>4</b>
<b>3. UPUTSTVO ZA KORIŠĆENJE SAJTA I STRUKTURA BAZE PODATAKA</b> .....	<b>8</b>
3.1. UPUTSTVO ZA KORISNIKA .....	8
3.2. UPUTSTVO ZA ADMINISTRATORA.....	11
3.3. STRUKTURA BAZE PODATAKA.....	15
<b>4. OPIS IMPLEMENTACIJE</b> .....	<b>18</b>
4.1. STRUKTURA SAJTA KROZ FAJLOVE .....	18
4.2. NAJBITNIJI DELOVI KODA.....	19
<b>5. ZAKLJUČAK</b> .....	<b>27</b>
<b>LITERATURA</b> .....	<b>28</b>

# 1. UVOD

Internet je redefinisao granice modernog sveta[1]. Male i velike kompanije, grupe i neprofitne organizacije pružaju beskonačan niz informacija rasprostranjenoj publici. Mogućnosti veb dizajna su beskrajne. Veb dizajn je umetnost prezentacije sadržaja za krajnjeg korisnika na *World Wide Web* servisu.

U digitalnom svetu, veb stranice pojavljuju se kao značajan alat koji ne samo da pomaže kompanijama kako bi pronašli svoju ciljnu grupu, već i kako bi privukli nove klijente iz celog sveta. Veb stranica je postala medij komunikacije i interakcije za poslovne kuće.

Oblast veb dizajna ima veliko učešće u svakodnevnom životu ljudi. Teško je zamisliti Internet bez animiranih slika, različitih stilova tipografije, šarenih pozadinskih slika i muzike.

Veb dizajn je opšte prihvaćen pojam koji obuhvata različite veštine, standarde i discipline koji se koriste u izradi veb stranica. Veb dizajn je planiranje i izrada veb stranica. To uključuje informacionu arhitekturu, korisničko okruženje, strukturu stranica, navigaciju, izgled, boje, fontove i slike. Sve ovo u kombinaciji sa načelima dizajna za stvaranje veb stranice koja zadovoljava ciljeve vlasnika i dizajnera.

Termin veb dizajn se obično koristi da opiše proces dizajniranja koji se odnosi na prednji deo (izgled stranice) i projektovanje veb sajta uključujući pisanje koda, koje spada u kodiranje, jer to je oblast takođe pokrivena veb razvojem.

Cilj ove teze, odnosno sajta, je da se na brz i lak način dođe do podataka o restoranima u gradu i njihovih ponuda jela. Sa jedne strane korisniku se omogućava da uz pomoć pretrage dobije informaciju u kojim sve restoranima ima određenog jela sa opisom samog jela i cenom ili da za određeni restoran vidi celu ponudu. Sa druge strane administratoru se daje mogućnost da lako dodaje nove podatke u bazu, menja ili briše postojeće.

U narednom, drugom poglavlju, biće reči o osnovnim alatima i programskim jezicima koji su korišćeni u samoj tezi. U trećem poglavlju će biti dat opis sajta sa korisničkog stanovišta, kako korisnika tako i administratora, i opis strukture baze podataka. U četvrtom će biti objašnjen sam kod sajta. I za kraj, u zaključku će biti reči o rezultatima teze, potencijalnim poboljšanjima i primenama.

## 2. OSNOVNI ALATI I PROGRAMSKI JEZICI KOJI SE KORISTE U DIPLOMSKOJ TEZI

Osnovni alati i programski jezici koji se koriste u tezi su HTML (*Hiper Text Markup Language*), CSS (*Cascading Style Sheets*), JavaScript, PHP (*Hypertext Preprocessor*) i MySQL (*My Structured Query Language*). Korišćen je i WAMP server (*Windows Apache MySQL PHP*). HTML i JavaScript se izvršavaju na klijentskoj strani, odnosno na korisničkom računaru[2]. PHP je jedan od programskih jezika za izradu dinamičkih veb strana koji se izvršava na serverskoj strani. PHP može da memoriše podatke i koristi ih u kasnijem trenutku. MySQL je najpoznatija i najviše korišćena open-sors baza podataka zasnovana na SQL jeziku.

HTML, jezik za označavanje hiperteksta, je opisni jezik specijalno namenjen opisu veb stranica. Pomoću njega se jednostavno mogu odvojiti elementi kao što su naslovi, paragrafi, citati i slično[3]. Pored toga, u HTML standard su ugrađeni elementi koji opisuju sam dokument kao što su kratak opis dokumenta, ključne reči, podaci o autoru i slično. Ovi podaci su opštepoznati kao meta podaci i jasno su odvojeni od sadržaja dokumenta.

HTML je nastao uprošćavanjem SGML (*Standard Generalized Markup Language*, standardizovani uopšteni jezik za označavanje) standarda sa svrhom opisa dokumenta koji se objavljuju na vebu. U početku je bio prilično ograničen što se označavanja sadržaja tiče i pružao je uglavnom elementarne stvari za označavanje i formatiranje teksta (paragrafi, naslovi, citati itd.). Kako je veb rastao tako je rasla i potreba za bogatijim sadržajem te je u tom smeru razvijan i HTML standard. Tada su standardu dodati elementi za opis tabela, slika, slojeva, napredno formatiranje teksta itd.

Aktuelna verzija standarda je HTML 4.01, a sam standard održava Konzorcijum za Veb (W3C, *World Wide Web Consortium*).

CSS je jezik formatiranja pomoću kog se definiše izgled elemenata veb stranice[4]. Prvobitno, HTML je služio da definiše kompletan izgled, strukturu i sadržaj veb stranice, ali je od verzije 4.0 HTML-a uveden CSS koji bi definisao konkretan izgled, dok je HTML ostao u funkciji definisanja strukture i sadržaja.

CSS je u određenoj formi postojao još u začecima SGML-a 1970ih godina. Kako je HTML postajao komplikovaniji, davao je sve više mogućnosti za definiciju izgleda elemenata, ali je istovremeno postajao nečitljiviji i teži za održavanje. Različiti brauzeri su prikazivali dokumente na različite načine, i postojala je potreba za doslednom tehnikom definisanja prikaza elemenata na stranici. Da bi se ovo postilgo, devet različitih metoda je predloženo na zvaničnom forumu W3C-a. Od devet, dve metode su izabrane kao temelj onoga što je kasnije postalo CSS: CHSS (*Cascading HTML Style Sheets*) i SSP (*Stream-based Style Sheet Proposal*).

Prvo je Hakon Vium Li (koji je sada šef tehničke službe kompanije Opera) predložio CHSS u oktobru 1994. godine, jezik koji je imao dosta sličnosti sa današnjim CSS-om. Bert Bos je radio na brauzeru po nazivu Argo, koji je imao sopstveni način definisanja stilova, SSP. Li i Bos su radili zajedno da bi osnovali CSS standard (slovo H je izbačeno iz skraćenice CHSS jer se CSS mogao odnositi i na druge jezike pored HTML-a). Za razliku od postojećih jezika, CSS

je dozvoljavao da više opisa utiče na dokument, tj. jedna definicija stilova je mogla naslediti osobine od druge. Lijev predlog je postavljen na konferenciji „Veb mozaik“ u Čikagu 1994. godine, i ponovo sa Bosovim predlogom 1995. Otprilike u ovo vreme je osnovan W3C, koji je preuzeo funkciju razvoja CSS-a. Do kraja 1996., CSS je bio spreman da se objavi kao standard, i CSS1 je objavljen u decembru.

Razvoj HTML-a, CSS-a i DOM-a (*Document Object Model*) se odvijao u jednoj istoj grupi, *HTML Editorial Review Board* (ERB). Početkom 1997. grupa ERB se podelila na tri radne grupe: radna grupa za HTML, kojom je upravljao Den Konoli iz W3C-a, radna grupa za DOM, kojom je upravljao Loren Vud iz kompanije Softkvod, i radna grupa za CSS, kojom je upravljao Kris Lili iz W3C-a.

Radna grupa za CSS je počela da radi na problemima koji nisu bili obuhvaćeni CSS-om verzije 1, koji se tako razvio u CSS2, 4. novembra 1997. Objavljen je kao zvanična verzija 12. maja 1998. CSS3, čiji je razvoj započeo 1998. se još uvijek razvija.

*JavaScript* je skriptni programski jezik koji se prvenstveno koristi za definisanje funkcionalnosti veb stranica na klijentskoj strani[5]. Dinamičan, slabo tipiziran jezik, sa skromnom podrškom za objektno orijentisano programiranje, on zapravo predstavlja implementaciju standarda *ECMAScript*-a.

Iako je prvobitna ideja bila da *JavaScript* liči na *Javu*, on ne sadrži ni jedan ključni element ovog programskog jezika osim onih koje su oba nasledili iz C programskog jezika. Jezik je najpoznatiji po programiranju klijentske funkcionalnosti veb stranica, ali se može koristiti i kao skriptni jezik za pristup objektima u drugim aplikacijama.

Jezik se prvo zvao *LiveScript*, ali je ime promenjeno u *JavaScript*, zbog marketinškog dogovora između kompanija *Netscape* i *Sun*. Ključni elementi dizajna *JavaScript*-a su nasleđeni iz programskog jezika *Self*.

PHP je specijalizovani skriptni jezik prvenstveno namenjen za izradu dinamičkog veb sadržaja i izvodi se na strani servera[6]. PHP je stekao popularnost zbog svoje jednostavnosti i sintakse nasleđene iz programskog jezika C. Tokom vremena jezik se proširivao i sticao mogućnosti za objektno orijentisano programiranje, naročito od verzije 5.0. Nalikuje jeziku C++ u smislu što dozvoljava i čisto-proceduralno programiranje, ali istovremeno omogućava i korišćenje klasa i drugih koncepata objektno orijentisanog programiranja (nasleđivanje, apstraktne klase i metode, interfejsi itd.).

PHP predstavlja naslednika alata po nazivu PHP/FI (*Personal Home Page Tools*), napisanog 1995. godine od strane Rasmus Lerdorfa. PHP/FI je predstavljao skup alata napisanih u programskom jeziku *Perl*, i autor ga je koristio za sopstvene potrebe. Skup alata je dobio ime „alati za ličnu prezentaciju“, odakle i skraćeni naziv PHP. Kako su rasle potrebe na sajtu, autor je prepisao kompletan projekat u programskom jeziku C i omogućio da može da komunicira sa bazama podataka, a korisnicima svog sajta da naprave sopstvene prezentacije pomoću njega. Rasmus je potom objavio svoje alate i učinio izvorni kod dostupan svima da bi se projekat brže razvijao i da bi se greške (bagovi) brže ispravljale.

PHP/FI, čija je puna definicija glasila „alati za ličnu prezentaciju/prevodilac obrazaca“ (*Personal Home Page Tools/Forms Interpreter*) je imao samo neke stvari zajedničke sa današnjim PHP-om - promenljive kao u *Perl*-u, automatsko raščlanjivanje promenljivih iz zahteva i ugrađeni HTML. PHP/FI 2.0 je konačno i zvanično objavljen 1997. godine, da bi ga ubrzo zamenio PHP 3.0.

PHP 3.0 je predstavljao prvu verziju koja dosta liči na današnje verzije PHP-a. Kreirali su ga izraelski programeri Zeev Suraski i Endi Gutmans sa fakulteta Tehnikon, kao potpunu preradu projekta PHP/FI. Otkrili su da je on imao ozbiljne nedostatke i da nije mogao zadovoljiti njihov univerzitetski projekat elektronske trgovine. U saradnji sa Ledorfom su objavili PHP 3.0 kao zvaničnog naslednika jezika PHP/FI, a njegov razvoj je zaustavljen. Glavna osobina nove verzije jezika je bila njegova proširivost, koja je privukla brojne programere širom sveta da dodaju nove mogućnosti. Pored kvalitetne podrške za razne vrste baza podataka, verzija 3.0 je uvela i mogućnost objektno orijentisanog programiranja i veću konzistentnost u jezičkoj sintaksi. Ime je promenjeno u PHP (bez dodatka "FI"), čije se značenje izmenilo da predstavlja rekurzivni akronim "PHP: preprocesor hiperteksta" (*PHP: Hypertext Preprocessor*). PHP 3.0 je zvanično objavljen u junu 1998. nakon devet meseci korišćenja u probnoj fazi.

U zimu 1998. Zeev i Gutmans su započeli ponovno pisanje PHP-a ispočetka, pokušavajući da poprave jezgro tako da bi bolje radilo sa većim aplikacijama i da bi poboljšali modularnost. Novo jezgro su nazvali Zend mašina (*Zend engine*), po delovima svojih imena Zeev i Endi, da bi kasnije formirali i *Zend Technologies* u Ramat Ganu, Izrael, koja nadgleda unapređenje PHP-a. Novo jezgro je uspešno dovršeno i pojavilo se u javnosti sredinom 1999., a nova verzija PHP-a, 4.0, opremljena novim jezgrom je zvanično objavljena u maju, 2000. Pored boljih performansi, nova verzija je uključivala i keširanje standardnog izlaza, sesije, bolju podršku za različite veb-servere, sigurniji prenos korisničkih podataka i nekoliko novih kontrolnih struktura.

Verzija 5.0 je izašla u julu 2004. godine, nakon dugog razvoja i nekoliko probnih verzija. Koristi jezgro Zend mašina 2.0 sa novim objektnim modelom i mnogo novih mogućnosti. Verzija 5 je takođe donela mnoštvo izmena u objektno-orijentisanom programiranju.

Iako se PHP može koristiti za programiranje konzolnih aplikacija i grafičkih interfejsa, njegova osnovna i glavna upotreba je u programiranju dinamičnih stranica na Internetu.

MySQL je višenitni, višekorisnički SQL sistem za upravljanje bazama podataka[7]. Sistem radi kao server, obezbeđujući višekorisnički interfejs za pristup bazi podataka.

MySQL je prvi put objavljen 23. maja, 1995. godine. Verzija za *Windows* je objavljena 8. januara 1998., za *Windows95* i *WindowsNT*. Verzija 3.23, beta verzija, objavljena je u junu 2000., a završna verzija u januaru 2001. Verzija 4.0, beta verzija, objavljena je u avgustu 2002., a završna verzija u martu 2003. U verziji 4.0 su dodate unije. Verzija 4.1, beta verzija, objavljena je u junu 2004., a završna verzija u oktobru 2004. U ovoj verziji dodata su r-drвета, pod-upiti i priprema upita unapred. Verzija 5.0, beta verzija, objavljena je u martu 2005., a završna verzija u oktobru 2005. U verziji 5.0 dodati su kursori, procedure, trigeri, pregledi, XA transakcije. Verzija 5.1 je trenutno u pripremi završne verzije (od novembra 2005.). Njoj se dodaju se particije, interfejs za plag-ine, replikacije na nivou pojedinačnih zapisa, tabele sa logovima servera i izvršavanje unaprijed zakazanih događaja.

MySQL je popularan u razvoju veb aplikacija. Njegova popularnost se veže za popularnost PHP-a, koji se obično kombinuje sa MySQL-om.

WAMP (*Windows Apache MySQL PHP*) je softverski paket. P se može odnositi na PHP, *Python* ili *Perl*. Pored verzije za *Windows*, postoji i LAMP (*Linux*), MAMP (*Macintosh*) i WIMP, gde I označava IIS (*Internet Information Services*) veb server. WAMP je najzastupljeniji i najčešće se koristi od svih. WAMP omogućava korisniku da na svom računaru simulira rad veb servera, i da za svoj sajt, prezentaciju ili šta god namerava da okači na Internet, proveri sve

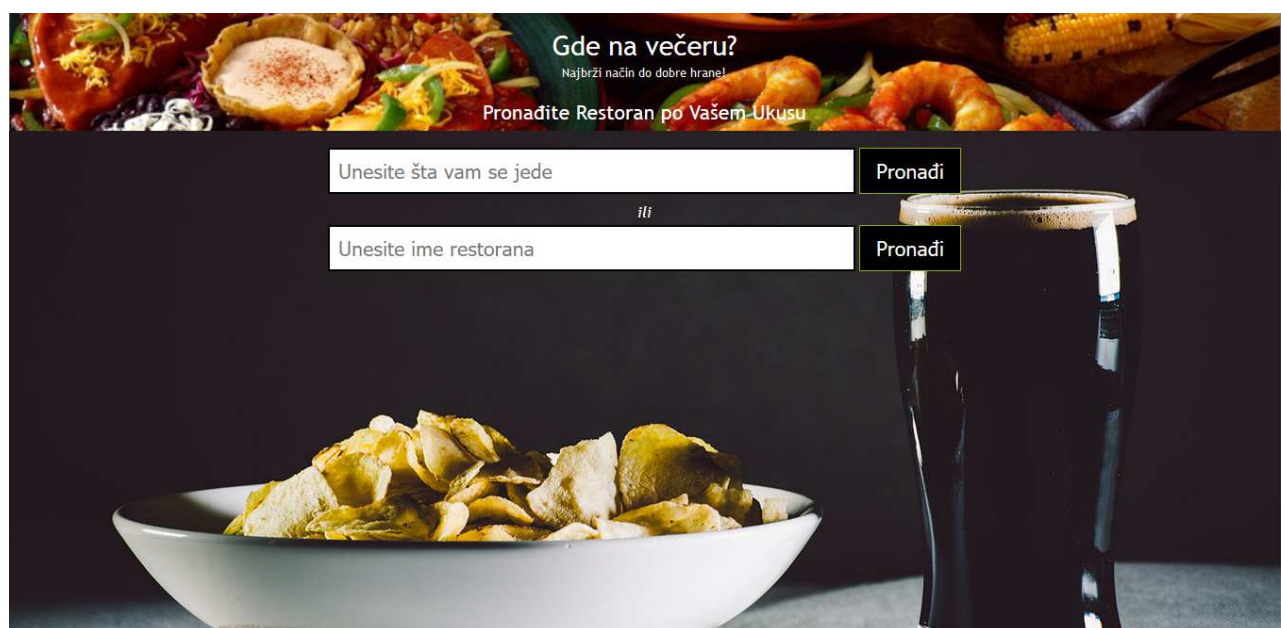
funkcionalnosti koje je u njega implementirao, i uveri se da sve radi kako treba. *Apache* veb server omogućava da se na korisnikovom računaru nalazi istovremeno i korisnik i lokalni server. MySQL je sistem za upravljanje relacionim bazama podataka, dok je PHP skriptni jezik koji služi za izradu dinamičnog veb sadržaja, i koji generiše novu veb stranu svaki put kada korisnik pristupi serveru. Takođe, u paketu mogu biti uključeni i dodatni programi, kao što je phpMyAdmin, kreiran u PHP-u, koji između ostalog služi za kreiranje, modifikovanje i brisanje baza podataka, tabela, kolona i redova, koristeći SQL programski jezik.

### 3. UPUTSTVO ZA KORIŠĆENJE SAJTA I STRUKTURA BAZE PODATAKA

Postoje dva uputstva za korišćenje sajta. Jedno je namenjeno korisniku kome su potrebne informacije o jelima i restoranima, a drugo administratoru koji ažurira podatke samog sajta.

#### 3.1. UPUTSTVO ZA KORISNIKA

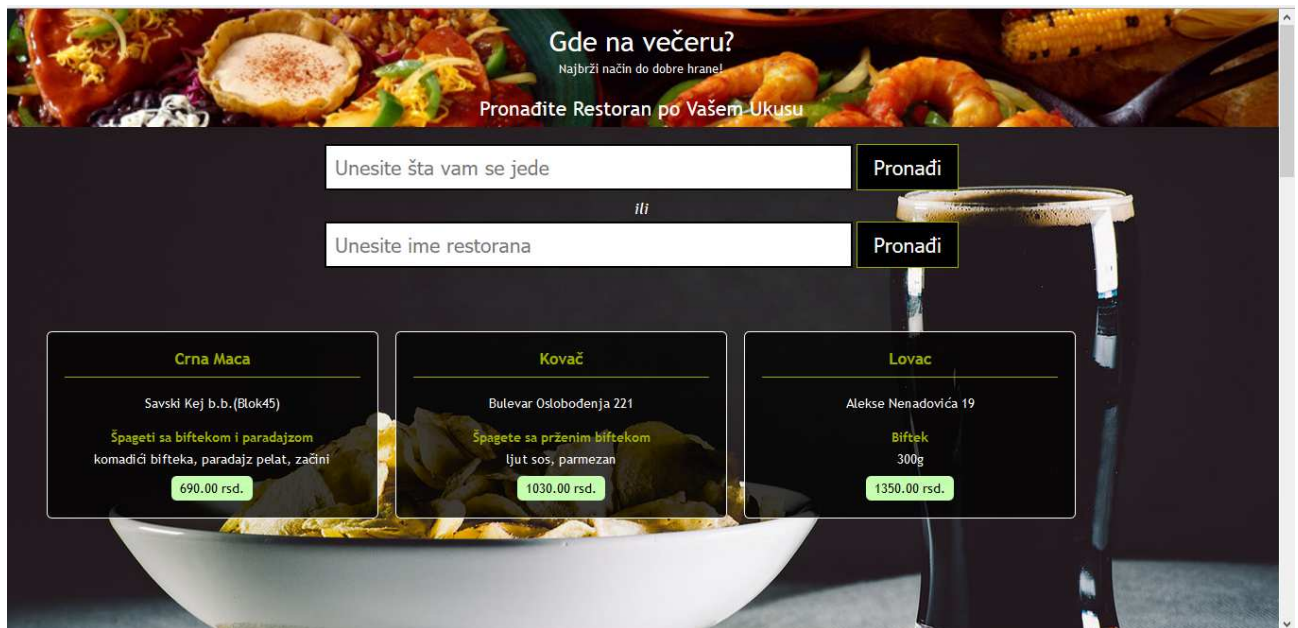
Korišćenje sajta od strane korisnika je veoma jednostavno. Kada korisnik ukuca adresu sajta **gdenaveceru.com** u pretraživač, pojavljuje se naslovna strana sajta. Na naslovnoj strani postoje dva polja za pretragu. Jedno polje je za pretragu jela, a drugo za pretragu restorana (slika 3.1).



Slika 3.1. Naslovna strana sajta

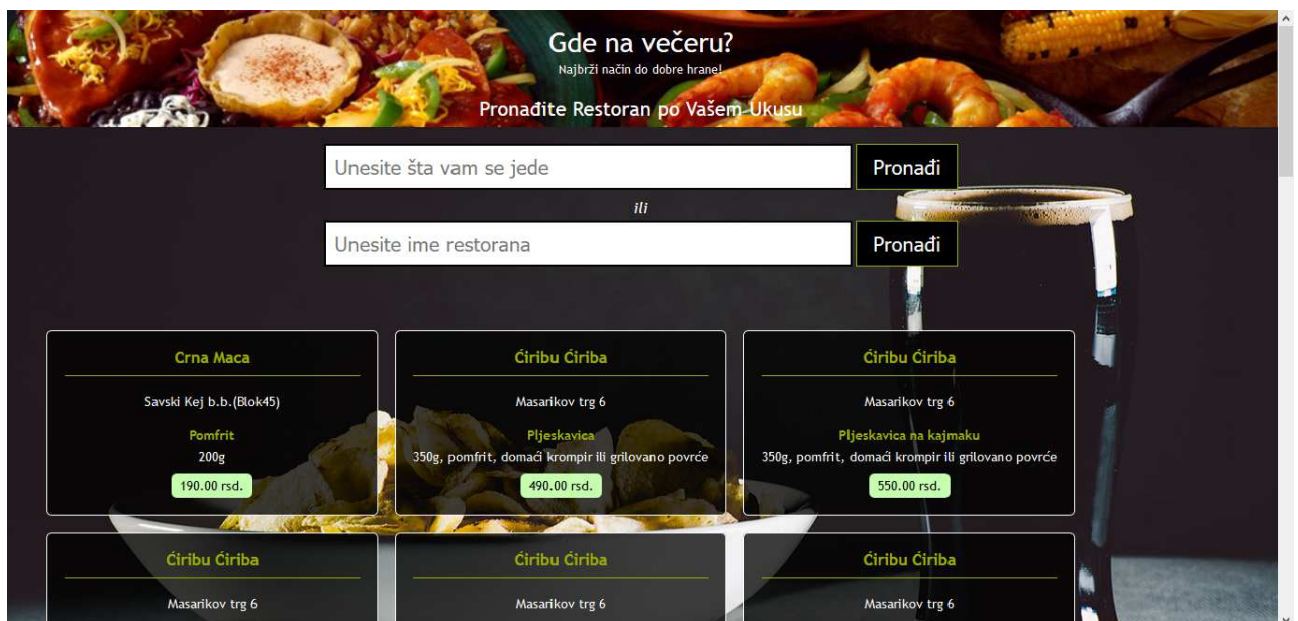
Kada se u polje za pretragu jela ukuca naziv željenog jela i klikne na dugme **Pronađi**, na istoj stranici se pojavljuju polja sa svim restoranima koji u svom jelovniku sadrže traženo jelo. Polja se sastoje od naziva restorana, njegove adrese, naziva i opisa jela i cene jela (slika 3.2).





Slika 3.2. Rezultat pretrage za izbor jela

Pored polja restorana koji sadrži konkretno jelo u jelovniku, pojavljuju se i restorani kod kojih se u opisu jela pojavljuje reč koja je ukucana u pretragu (slika 3.3).



Slika 3.3. Rezultat pretrage za izbor jela sa traženim pojmom u opisu jela

Ako korisnik slučajno nije otkucao prvih par ili poslednjih par slova u nazivu jela, sajt će to prepoznati, izvršiće pretragu, i restorani će se svakako izlistati.

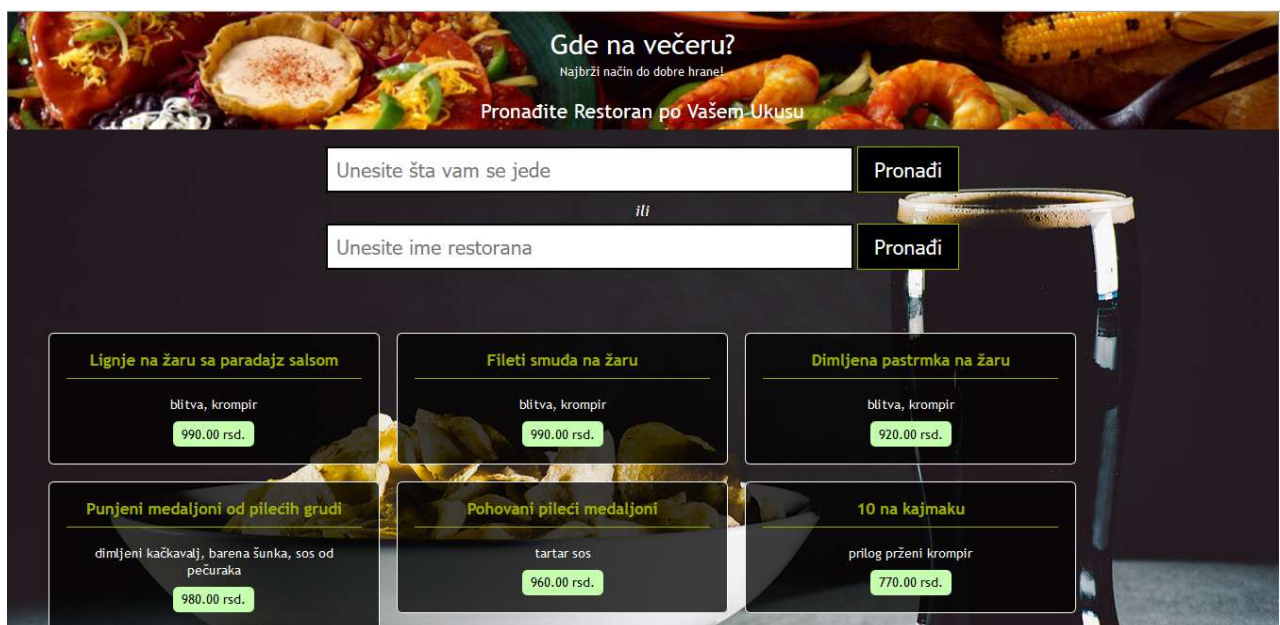
Kada se u polje za pretragu restorana ukuca naziv restorana i klikne na dugme **Pronađi**, na istoj stranici se pojavljuje polje sa nazivom i adresom restorana, i linkom koji vodi ka njegovom jelovniku (slika 3.4).



Slika 3.4. Rezultat pretrage za izbor restorana

Kao i kod pretrage jela, ako korisnik slučajno izostavi prvih par ili poslednjih par slova iz naziva restorana, pretraga će se izvršiti i rezultat će biti prikazan.

Pošto se klikne na link **Prikaži Jelovnik Restorana**, na stranici se pojavljuju polja koja sadrže sva jela tog restorana sa opisima i cenama (slika 3.5).



Slika 3.5. Jelovnik određenog restorana

Ako korisnik u pretragu ukuca naziv jela ili restorana koji ne postoje u bazi podataka i klikne na dugme **Pronađi**, pojaviće se naslovna strana sajta za dalju pretragu.

Na dnu stranice se nalazi ime administratora, **Snežana**, odnosno link koji otvara email administratora, preko koga se korisnik može javiti sa pohvalama, primedbama ili predlozima kako bi se sajt unapredio (slika 3.6).



Slika 3.6. Link za email administratora

### 3.2. UPUTSTVO ZA ADMINISTRATORA

Stranica koju koristi administrator se sastoji iz dva dela. Na vrhu stranice su predviđena polja za unos podataka o restoranima i jelima (slika 3.7).

Slika 3.7. Stranica koju koristi administrator, deo za unos podataka

U drugom delu stranice, ispod polja za unos podataka, nalaze se polja sa već unetim podacima, gde administrator ima mogućnost da menja ili briše podatke (slika 3.8).

Slika 3.8. Stranica koju koristi administrator, deo sa postojećim podacima

U prvom delu, gde administrator unosi nove podatke, postoje dva zasebna polja za unos. U levom polju, u okvire predviđene za to, administrator unosi naziv i adresu restorana, i klikom na dugme **Dodaj** restoran se čuva u bazu podataka (slika 3. 9).

Slika 3.9. Unos novog restorana

U desnom polju, administrator prvo bira jedan od ponuđenih restorana kome hoće da doda novo jelo u jelovnik. Kada izabere restoran, u okvire predviđene za to, unosi naziv, opis jela i cenu. Klikom na dugme **Dodaj**, novo jelo se čuva u bazu podataka (slika 3.10).

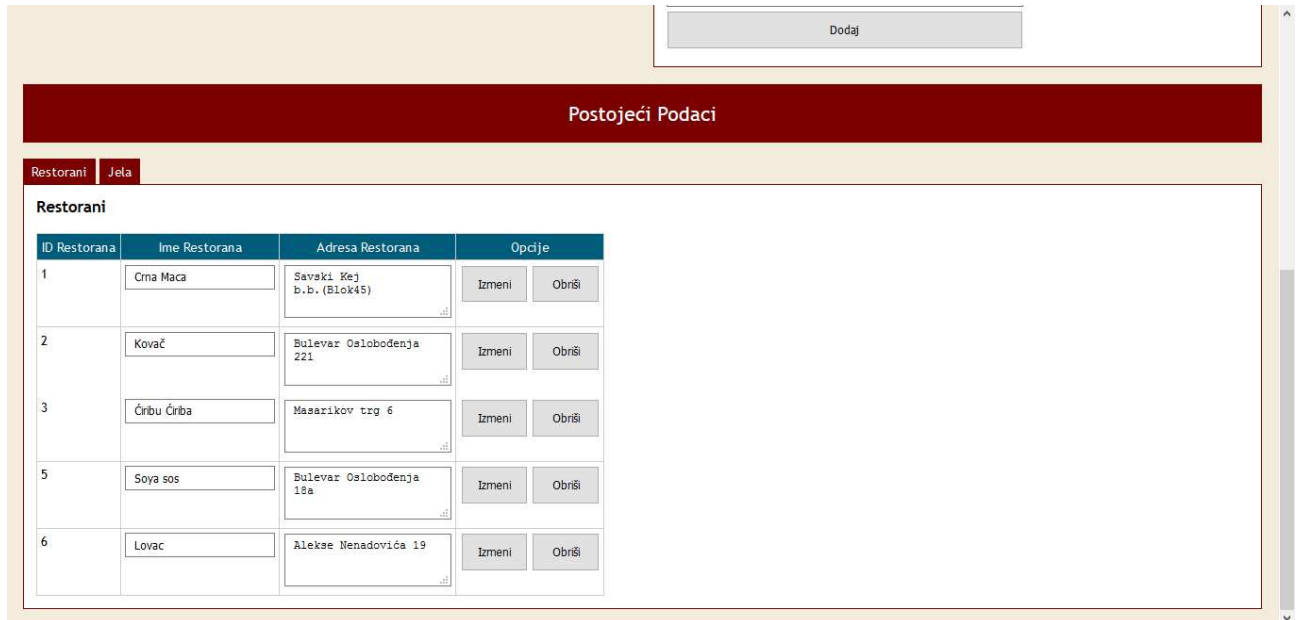
Slika 3.10. Unos novog jela za izabrani restoran

Treba naglasiti da, da bi se restoran ili jelo dodali u bazu, je obavezno popuniti sva polja u unosu. U suprotnom, podaci neće biti uneti.

U drugom delu stranice nalaze se već postojeći podaci o restoranima i jelima. Postoje dva dugmeta koja zasebno otvaraju tabelu sa restoranima i tabelu sa jelima određenog restorana. Kada se klikne na dugme **Restorani** otvara se tabela sa svim restoranima koji se trenutno nalaze

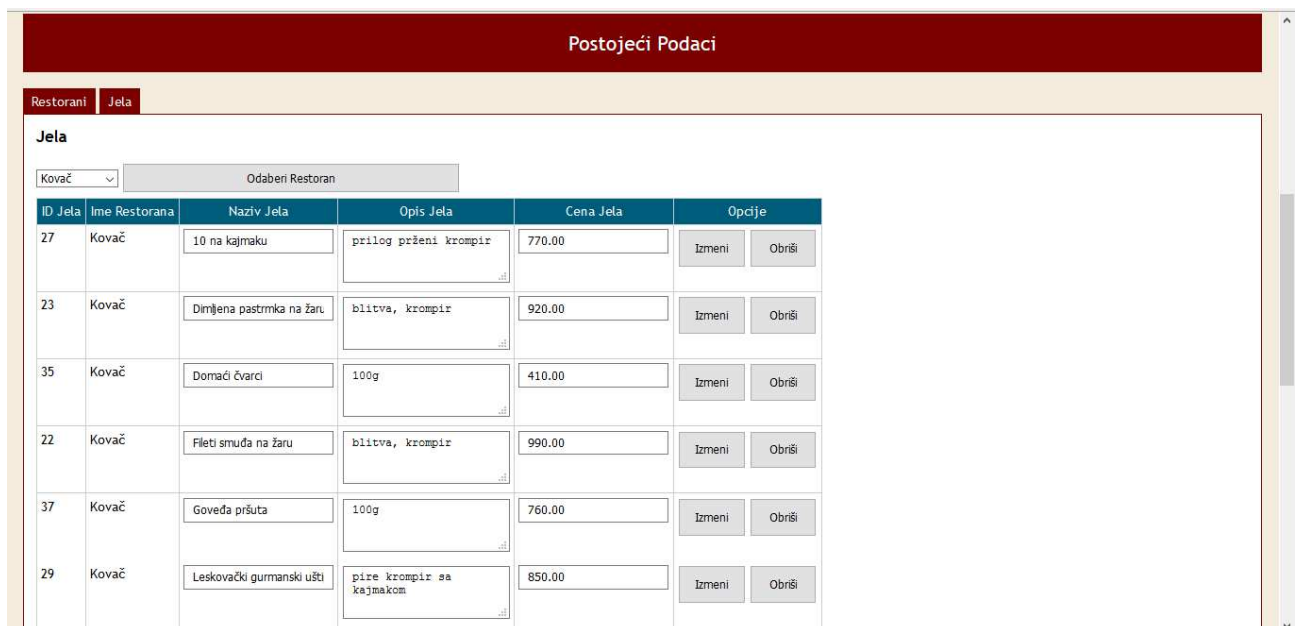


u bazi. Administrator u tom slučaju ima dve opcije. Jedna opcija je da, ako želi, može da u predviđenim poljima promeni naziv i/ili adresu restorana, i klikom na dugme **Izmeni** ti podaci će biti ažurirani. Druga opcija je brisanje restorana iz baze, što se postiže klikom na dugme **Obriši** (slika 3.11).



Slika 3.11. Ažuriranje postojećih restorana

Kada se klikne na dugme **Jela** otvara sa jelima. Administrator prvo bira jedan restoran od ponuđenih, i klikom na dugme **Odaberi Restoran** otvara se tabela sa svim jelima koja su u ponudi izabranog restorana. Administrator i u ovom slučaju ima dve opcije. Jedna je da u predviđenim poljima promeni naziv, opis i/ili cenu jela i klikom na dugme **Izmeni** izmeni podatke u bazi. Druga opcija je brisanje jela iz baze klikom na dugme **Obriši** (slika 3.12).

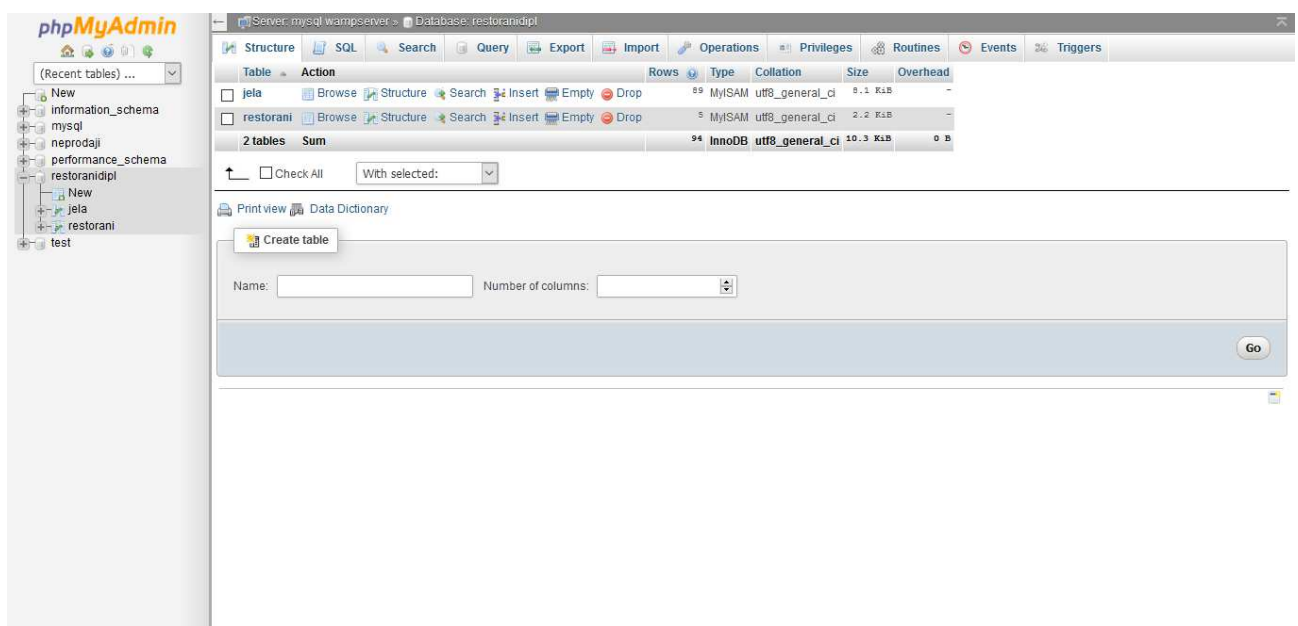


Slika 3.12. Ažuriranje postojećih jela određenog restorana

Brisanjem konkretnog restorana automatski se brišu i sva jela iz ponude tog restorana.

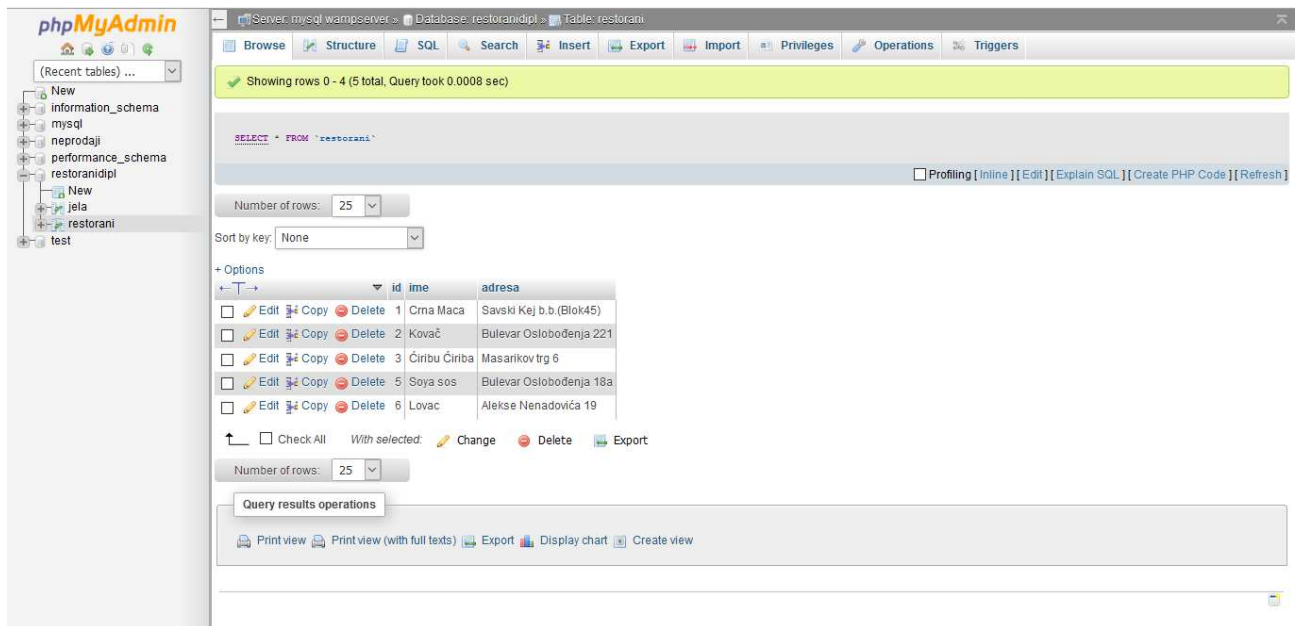
### 3.3. STRUKTURA BAZE PODATAKA

Koristi se MySQL baza podataka. Za rad sa bazom podataka, u ovom slučaju, koristi se phpMyAdmin. Može se koristiti i neki drugi alat za rad sa MySQL bazama. Struktura baze podataka **restoranidipl**, kojoj samo administrator ima pristup, sastoji se iz dve tabele, tabele **jela** i tabele **restorani** (slika 3.13).



Slika 3.13. Izgled baze podataka restoranidipl

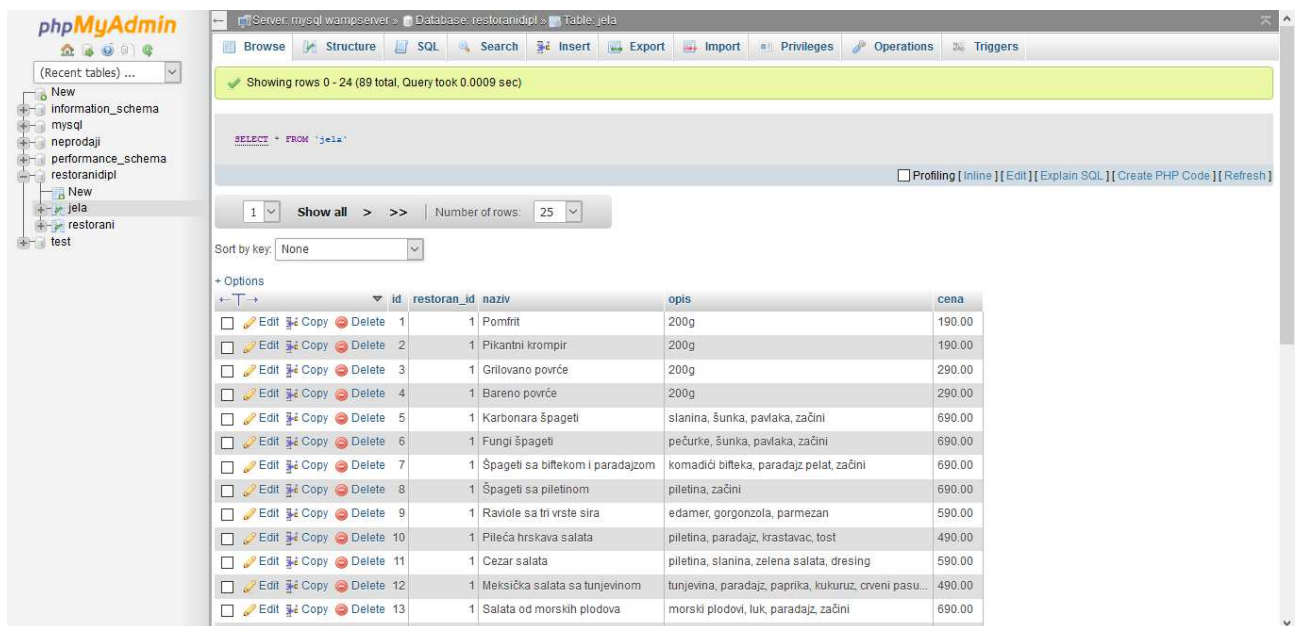
Tabela **restorani** se sastoji iz tri kolone, a to su id restorana, naziv i adresa restorana. Administrator ima opciju da direktno u bazi kopira ili obriše postojeće podatke u poljima sve tri kolone, ili da ih menja i to tako što dva puta klikne na određeno polje, a zatim promeni šta je hteo (slika 3.14).



Slika 3.14. Izgled tabele restorani u bazi

Tabela **jela** se sastoji iz pet kolona. Prve dve kolone su id jela i id\_restorana iz čijeg je jelovnika to jelo, dok su ostale kolone naziv, opis i cena jela. Polje id\_restorana predstavlja poveznicu sa tabelom koja čuva podatke o restoranima, i na taj način se jelo povezuje sa restoranom koje nudi dotično jelo.

Administrator, kao i u tabeli **restorani**, ima opciju da direktno u bazi kopira ili obriše postojeće podatke u poljima svih pet kolona, ili da ih menja tako što dva puta klikne na određeno polje, a zatim promeni šta je teo (3.15).



Slika 3.15. Izgled tabele jela u bazi



Administrator ima mogućnost i da izbriše kompletnu bazu. U tom slučaju će se svi podaci izbrisati jer baza neće postojati.

## 4. OPIS IMPLEMENTACIJE

U ovom poglavlju će biti detaljnije objašnjen programski kod sajta.

### 4.1. STRUKTURA SAJTA KROZ FAJLOVE

Sajt čini devet fajlova zbog kojih tako izgleda i funkcioniše.

Fajl *kreirajbazu.php*, kao što mu i samo ime kaže, služi za kreiranje baze podataka. On se koristi samo jednom, i za njim nema više potrebe kad se baza formira. Prvo se kreira i proverava konekcija, a zatim se kreira konkretna baza **restoranidipl** sledećim kodom:

```
// Kreiranje konekcije
$conn = mysqli_connect($servername, $username, $password);
// Provera konekcije
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
} else {
    mysqli_query($conn, "SET NAMES utf8");
}

// Kreiranje baze
$sql = "CREATE DATABASE IF NOT EXISTS restoranidipl DEFAULT COLLATE
utf8_general_ci";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully <br />";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}

// Selektovanje baze
mysqli_select_db($conn, "restoranidipl");
```

Potom se kreiraju tabele **restorani** i **jela** u bazi sa svim kolonama, i vrši se provera da li je to uspešno izvršeno:

```
// Kreiranje tabele za jela
$sql = "CREATE TABLE IF NOT EXISTS `jela` (
    `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
    `restoran_id` int(10) unsigned NOT NULL,
    `naziv` varchar(256) NOT NULL,
    `opis` text NOT NULL,
    `cena` decimal(6,2) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;";
if (mysqli_query($conn, $sql)) {
    echo "Table jela created successfully <br />";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}
```

```

}

// Kreiranje tabele za restorane
$sql = "CREATE TABLE IF NOT EXISTS `restorani` (
        `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
        `ime` varchar(128) NOT NULL,
        `adresa` text NOT NULL,
        PRIMARY KEY (`id`)
    ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;";
if (mysqli_query($conn, $sql)) {
    echo "Table restorani created successfully <br />";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}

```

Struktura baze podataka je detaljnije objašnjena na kraju prethodnog poglavlja.

U fajlu *baza.php* se kreira i proverava konekcija sa bazom uz pomoć koga će se u svim ostalim fajlovima, sem u fajlu *stil.css*, kreirati i proveravati konekcija sa bazom uz pomoć funkcije `require_once('baza.php');`

Fajl *index.php* se sastoji iz više delova. Prvi deo je PHP deo koji na osnovu pretrage, koja se vrši u izboru restorana ili jela, pristupa podacima iz baze, odnosno tabelama i kolonama vezanim za one podatke koji se ispisuju kao rezultat pretrage. Drugi deo fajla čini HTML deo, koji je odgovoran za izgled i sadržaj stranice sajta koja je dostupna korisniku. Treći, poslednji deo ovog fajla, čini *JavaScript* deo koji proverava da li su sva relevantna polja popunjena. O najbitnijim delovima koda koje sadrži fajl *index.php* će biti više reči malo kasnije.

Fajl *stil.css* služi za definisanje stilova koji određuju izgled HTML elemenata (font, boje pozadine, razmake...).

Fajl *unos.php* se takođe sastoji iz više delova. HTML deo je odgovoran za izgled i sadržaj stranice sajta koju koristi administrator. U okviru koda, PHP deo služi da se pristupi bazi i određenim podacima iz tabela baze i da se izvrše određene funkcije po pitanju unosa novih podataka i ažuriranja postojećih. *JavaScript* deo čini da se sa korišćenjem tabova odvoje sadržaji tabela vezanih za restorane i za jela na samoj stranici, tako da kada je prikazana tabela restorana tabela sa jelima je sakrivena i obrnuto. O najbitnijim delovima koda koje sadrži fajl *unos.php* će biti više reči malo kasnije.

Kodovi fajlova *unos-jela.php* i *unos-restorana.php* rade po istom principu. Prilikom unosa novog restorana ili jela za određeni restoran, kod proverava da li su sva relevantna polja popunjena, ako jesu, novi podaci se upisuju u bazu.

Kodovi fajlova *azuriraj-jelo.php* i *azuriraj-restoran.php*, kao i u prethodnom slučaju, rade po istom principu. Pomoću njih se omogućava izmena ili brisanje pojedinačnih podataka iz baze vezanih za restorane i jela.

## 4.2. NAJBITNIJI DELOVI KODA

HTML deo u fajlovima *index.php* i *unos.php* služi da opiše same veb stranice. Pomoću njega su odvojeni elementi kao što su naslov i podnaslov stranice. Jasno je naznačeno čemu služe polja u kojima se unosi pretraga. Kod unosa novih i ažuriranja već postojećih podataka na

stranici koju koristi administrator, HTML je odgovoran za naslove samih polja i jasnih opisa šta se u koje polje upisuje ili šta se u kom polju već nalazi. CSS deo služi za definisanje stilova koji određuju izgled HTML elemenata kao što su fontovi, boje, pozadine, razmaci. CSS fajl tačno određuje kojim fontom i kojom veličinom fonta će se određeni tekst ispisati na stranici, kojom bojom, zatim kako će izgledati pozadina stranice, kako će izgledati posebna polja stranice, okviri, rezultati pretrage i još mnogo toga. Ukratko, za vizuelni izgled sajta, kako za korisnika tako i za administratora, zaslužni su HTML i CSS delovi ovog koda. HTML i CSS deo su povezani na sledeći način:

```
<!DOCTYPE html>
<html>
<head>
  <title>Restorani</title>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="css/stil.css" />
</head>
```

Provera da li su polja za pretragu popunjena, na naslovnoj strani, ispituje *JavaScript* uz pomoć funkcije:

```
<script type="text/javascript">
function provera(){
  var pretraga = document.getElementById("pretraga");
  if(pretraga.value == ''){
    alert('Unesite');
    return false;
  }else{
    return true;
  }
}

function provera_restorana(){
  var pretraga = document.getElementById("pretraga_restorana");
  if(pretraga.value == ''){
    alert('Unesite');
    return false;
  }else{
    return true;
  }
}
</script>
```

Već je rečeno da *JavaScript* kod u fajlu *unos.php* služi da se sa korišćenjem tabova odvoje sadržaji tabela vezanih za restorane i za jela na samoj stranici, tako da kada je prikazana tabela sa restoranima tabela sa jelima je sakrivena i obrnuto. To se radi uz pomoć sledeće funkcije:

```
<script type="text/javascript">
function showTab(tabID){
  var divsToHide = document.getElementsByClassName("tab-content");
  for(var i = 0; i < divsToHide.length; i++){
    divsToHide[i].classList.add("hidden");
  }
  var divToShow = document.getElementById(tabID);
  divToShow.classList.remove("hidden");
}
</script>
```

Za pretrage na naslovnoj strani sajta, koja služi korisniku, zaslužni su PHP delovi koda. Kada se unese pretraga u polje za jela, kod će u bazi naći, ako postoje, jela koja imaju zadatu pretragu u nazivu ili u opisu jela:

```
// Ako je uneta pretraga
if(isset($_POST['pretraga'])) {

    $pretraga = $_POST['pretraga'];

    // Nalazjenje jela koja imaju zadatu pretragu u nazivu
    $sql = "SELECT r.*, j.naziv, j.cena, j.opis FROM jela j LEFT JOIN
restorani r ON (r.id = j.restoran_id) WHERE j.naziv LIKE '%$pretraga%'
OR j.opis LIKE '%$pretraga%'";

    if ($pretraga = mysqli_query($conn, $sql)) {
        // Ima Rezultata pretrage
        $rezultati = true;
        $tipPretrage = "jela";
    } else {
        // Greska
        echo "Greška: " . mysqli_error($conn);
        $rezultati = false;
    }
}
```

U suprotnom, ako se unese pretraga u polje za restorane po nazivu ili adresi restorana, kod će naći restoran u bazi, ako postoji, sa podacima vezanim za taj restoran:

```
}else if(isset($_POST['pretraga-restorana'])) {

    $pretraga = $_POST['pretraga-restorana'];

    // Nalazjenje restorana koji imaju zadatu pretragu u nazivu
    $sql = "SELECT * FROM restorani WHERE ime LIKE '%$pretraga%'
OR adresa LIKE '%$pretraga%'";

    if ($pretraga = mysqli_query($conn, $sql)) {
        // Ima Rezultata pretrage
        $rezultati = true;
        $tipPretrage = "restorani";
    } else {
        // Greska
        echo "Greška: " . mysqli_error($conn);
        $rezultati = false;
    }
}
```

U slučaju da se pretragom birao restoran koji postoji u bazi, na naslovnoj stranici će se pojaviti polje sa restoranom i linkom ka jelovniku tog restorana. Kod zadužen za pristupanje podacima vezanim za jela iz jelovnika određenog restorana u bazi je sledeći:

```
// Ako je odabran restoran, prikazivanje jelovnika
if(isset($_GET['jelovnik'])) {
    $restoranId = $_GET['jelovnik'];

    $sql = "SELECT r.*, j.naziv, j.cena, j.opis FROM jela j LEFT JOIN
restorani r ON (r.id = j.restoran_id) WHERE j.restoran_id = $restoranId";

    if ($pretraga = mysqli_query($conn, $sql)) {
        // Ima Rezultata pretrage
        $rezultati = true;
        $tipPretrage = "jelovnik";
    }
}
```

```

    } else {
        // Greska
        echo "Greška: " . mysqli_error($conn);
        $rezultati = false;
    }
}

```

Za prikaz rezultata pretraga koji se pojavljuje na naslovnoj stranici zadužen je sledeći kod:

```

<div class="rezultati">
    <?php
        // Ako je uneta pretraga
        if(isset($tipPretrage)){
            if($tipPretrage == "jela"){
                // Izlistavanje svih jela iz baze
                while($row = mysqli_fetch_assoc($pretraga)) {
                    ?>
                <div class="rezultat">
                    <h3><?php echo $row['ime'] ;?></h3>
                    <?php echo $row['adresa'];?>
                    <h4><?php echo $row['naziv'];?></h4>
                    <?php echo $row['opis']; ?><br/>
                    <span class="cena"><?php echo $row['cena']; ;?> rsd.</span>
                </div>
                    <?php
                }
            }else if($tipPretrage == "restorani"){
                // Izlistavanje svih restorana iz baze
                while($row = mysqli_fetch_assoc($pretraga)) {
                    ?>
                <div class="rezultat">
                    <h3><?php echo $row['ime'] ;?></h3>
                    <?php echo $row['adresa'];?>
                    <a href="index.php?jelovnik=<?php echo $row['id'];?>" class="link">
                        Prikaži Jelovnik Restorana
                    </a>
                </div>
                    <?php
                }
            }else if($tipPretrage == "jelovnik"){
                // Izlistavanje jelovnika restorana iz baze
                while($row = mysqli_fetch_assoc($pretraga)) {
                    ?>
                <div class="rezultat">
                    <h3><?php echo $row['naziv'];?></h3>
                    <?php echo $row['opis']; ?><br/>
                    <span class="cena"><?php echo $row['cena']; ;?> rsd.</span>
                </div>
                    <?php
                }
            }
        }else{
            // Ako nema rezultata
            echo "Unesite traženi pojam.";
        }
    ?>
</div>

```

Promenljiva je simbol koji se može primenjivati za smeštanje određenih vrednosti koje kasnije možemo iskoristiti. `$_POST` promenljiva je superglobalna promenljiva koja služi za prikupljanje podataka koji su poslani putem POST metode. Na stranici `index.php` jedan deo koda omogućava da u polje za pretragu unesemo jelo po želji:

```
<form method="post" action="index.php" onsubmit="return provera();" >
  <input type="text" id="pretraga" name="pretraga" placeholder="Unesite
šta vam se jede" />
  <input type="submit" class="submit" value="Pronadi" />
</form>.
```

U `form` tag je dodan atribut `method`, tu je definisan metod slanja podataka, u ovom slučaju POST, i atribut `action` u koji se upisuje putanja do stranice kojoj će se nakon pritiska na dugme **Pronadi** proslediti podaci putem POST metoda, u ovom slučaju to je ista stranica, `index.php`. Nakon pritiska na dugme **Pronadi** kreira se promenljiva `$_POST['pretraga']` i prosleđuje se istoj stranici putem POST metoda:

```
// Ako je uneta pretraga
if(isset($_POST['pretraga'])) {

    $pretraga = $_POST['pretraga'];

    // Nalazenje jela koja imaju zadatu pretragu u nazivu
    $sql = "SELECT r.*, j.naziv, j.cena, j.opis FROM jela j LEFT JOIN
restorani r ON (r.id = j.restoran_id) WHERE j.naziv LIKE '%$pretraga%'
OR j.opis LIKE '%$pretraga%'";

    if ($pretraga = mysqli_query($conn, $sql)) {
        // Ima Rezultata pretrage
        $rezultati = true;
        $tipPretrage = "jela";
    } else {
        // Greska
        echo "Greška: " . mysqli_error($conn);
        $rezultati = false;
    }
}
```

Kao rezultat dobiće se jelo koje je korisnik uneo na stranici u polje za pretragu.

Kada je reč o unosu novih podataka u bazu ili menjanju i brisanju već postojećih od strane administratora, postoji više tipova unosa.

Unos novog restorana u bazu se vrši sledećim kodom:

```
// Preuzimanje podataka iz forme
$ime_restorana = $_POST['ime_restorana'];
$adresa_restorana = $_POST['adresa_restorana'];

// Provera podataka iz forme
if(!empty($ime_restorana) && !empty($adresa_restorana)) {
    // Ubacivanje u bazu
    $sql = "INSERT INTO restorani (ime, adresa) VALUES
('$ime_restorana', '$adresa_restorana')";
    if (mysqli_query($conn, $sql)) {
        echo "Uspešno ažurirano.";
    } else {
        echo "Greška: " . mysqli_error($conn);
    }
}
```

Pre unosa novog jela postoji opcija da se izabere jedan od restorana koji se nalaze u bazi podataka. To se radi uz pomoć koda:

`<h3>Jela</h3>`

```
<form method="get" action="unos.php">
  <select name="restoran">
    <?php
      // Uzimanje podataka o restoranima
      $sql = "SELECT * FROM restorani ORDER BY id ASC";
      $result = mysqli_query($conn, $sql);
      if (mysqli_num_rows($result) > 0) {
        while($row = mysqli_fetch_assoc($result)) {
          ?>
          <option value="<?php echo $row['id'];?>" <?php echo
            ($row['id'] == $odabraniRestoran) ? 'selected' : '';?>
            <?php echo $row['ime'];?></option>
          <?php
        }
      }
    <?php
  </select>
  <input type="submit" value="Odaberi Restoran" />
</form>
```

Unos novog jela u bazu se vrši sledećim kodom:

```
// Preuzimanje podataka iz forme
$restoran = $_POST['id_restorana'];
$naziv_jela = $_POST['naziv_jela'];
$opis_jela = $_POST['opis_jela'];
$cena_jela = $_POST['cena_jela'];

// Provera podataka iz forme
if(!empty($naziv_jela) && !empty($opis_jela) && $restoran != 0){
  // Ubacivanje u bazu
  $sql = "INSERT INTO jela (restoran_id, naziv, opis, cena) VALUES
($restoran, '$naziv_jela', '$opis_jela', '$cena_jela')";
  if (mysqli_query($conn, $sql)) {
    echo "Uspešno ažurirano.";
  } else {
    echo "Greška: " . mysqli_error($conn);
  }
}
```

Pre izmene ili brisanja podataka o postojećem restoranu, PHP kod prvo pristupa podacima u bazi vezanim za restorane. Izmena ili brisanje podataka o postojećem restoranu, kao i brisanje jela tog restorana iz baze, vrši se na sledeći način:

```
if(isset($_POST['izmeni'])){

  // Izmena Podataka
  $id = $_POST['id'];
  $ime_restorana = $_POST['ime_restorana'];
  $adresa_restorana = $_POST['adresa_restorana'];

  $sql = "UPDATE restorani SET ime = '$ime_restorana',
adresa = '$adresa_restorana' WHERE id = $id";
```



```

    if (mysqli_query($conn, $sql)) {
        echo "Uspešno ažurirano.";
    } else {
        echo "Greška: " . mysqli_error($conn);
    }
}

}else if(isset($_POST['obrisi'])) {

    // Brisanje Podataka
    $id = $_POST['id'];

    // Brisanje Jela od Restorana
    $sql = "DELETE FROM jela WHERE restoran_id = $id";
    if (mysqli_query($conn, $sql)) {
        echo "Uspešno ažurirano.";
    } else {
        echo "Greška: " . mysqli_error($conn);
    }

    // Brisanje Restorana
    $sql = "DELETE FROM restorani WHERE id = $id";
    if (mysqli_query($conn, $sql)) {
        echo "Uspešno ažurirano.";
    } else {
        echo "Greška: " . mysqli_error($conn);
    }

}

}else {
    // Nepravilno pozivanje skripte
}

```

Pre izmene ili brisanja podataka o postojećim jelima, postoji opcija da se izabere restoran čija će se ponuda ažurirati. Ova opcija služi da bi se izbegla prenatrpanost prozora velikom količinom podataka koji bi poticali od svih restorana u ponudi. Pre ažuriranja jela iz baze, kod pristupa svim podacima u bazi koji su vezani za jela određenog restorana. Izmena ili brisanje podataka o postojećim jelima u bazi vrši se uz pomoć koda:

```

if(isset($_POST['izmeni'])) {

    // Izmena Podataka
    $id = $_POST['id'];
    $naziv_jela = $_POST['naziv_jela'];
    $opis_jela = $_POST['opis_jela'];
    $cena_jela = $_POST['cena_jela'];

    $sql = "UPDATE jela SET naziv = '$naziv_jela',
    opis = '$opis_jela', cena = '$cena_jela' WHERE id = $id";
    if (mysqli_query($conn, $sql)) {
        echo "Uspešno ažurirano.";
    } else {
        echo "Greška: " . mysqli_error($conn);
    }
}

}else if(isset($_POST['obrisi'])) {

    // Brisanje Podataka
    $id = $_POST['id'];

```

```
$sql = "DELETE FROM jela WHERE id = $id";
if (mysqli_query($conn, $sql)) {
    echo "Uspešno ažurirano.";
} else {
    echo "Greška: " . mysqli_error($conn);
}

}else{
    // Nepravilno pozivanje skripte
}
.
```

## 5. ZAKLJUČAK

U tezi je realizovan sajt za pretragu restorana ili jela po želji korisnika. Korisnik, u zavisnosti šta mu se u datom trenutku jede, dobija kao rezultat restorane koji imaju određeno jelo u ponudi, ili ako postoji određeni restoran u koji bi voleo da ide, nudi mu se meni samog restorana sa jelima, opisima istih i cenama. Administrator, s druge strane, ima potpunu kontrolu nad bazom podataka. Administrator sam bira koje će restorane uvrstiti u ponudu, kao i jela koja će se naći u jelovnicima restorana. Administrator ima mogućnost da menja podatke kao što su naziv i adresa restorana, kao i naziv, opis i cena jela. Takođe ima mogućnost da obriše bilo koji restoran i bilo šta što je u njegovoj ponudi.

Uz pomoć ovog sajta korisnik na brz i lak način može da pronađe restoran koji ispunjava sve njegove kriterijume za više nego prijatan obrok. Sajt svakako ima prostora za unapređenje. Dodavanje karte pića u pretragu, kao i dezerta, bi upotpunilo ponudu restorana. Ubacivanje kontakta, lokacija restorana, zatim fotografija ambijenta i ponude bi dodatno olakšalo korisniku izbor mesta gde bi hteo da provede svoje dragoceno vreme. Najbolje je oslušivati šta je to što se traži među korisnicima, pa uz pomoć tih zahteva krenuti u razvoj i unapređenje osnovne verzije sajta, kako u vizuelnom tako i u dinamičkom delu.

## LITERATURA

- [1] <http://www.popwebdesign.net/web-dizajn.html>
- [2] Materijali sa predavanja predmeta Internet programiranje
- [3] <https://sr.wikipedia.org/wiki/HTML>
- [4] <https://sr.wikipedia.org/wiki/CSS>
- [5] <https://sh.wikipedia.org/wiki/JavaScript>
- [6] <https://sr.wikipedia.org/sr-el/PHP>
- [7] <https://sr.wikipedia.org/sr-el/MySQL>
- [8] <http://www.w3schools.com/>