

Web aplikacija za restoran brze hrane

Horvat, Robert

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:575417>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-08-04**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Stručni studij

WEB APLIKACIJA ZA RESTORAN BRZE HRANE

Završni rad

Robert Horvat

Osijek, 2020.

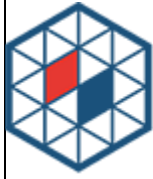
**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju**

Osijek, 14.09.2020.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za završni ispit
na preddiplomskom stručnom studiju**

Ime i prezime studenta:	Robert Horvat
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	A14545, 25.09.2019.
OIB studenta:	72749100780
Mentor:	Robert Šojo
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Marina Peko
Član Povjerenstva 1:	Robert Šojo
Član Povjerenstva 2:	dr.sc. Ivana Hartmann-Tolić
Naslov završnog rada:	Web aplikacija za restoran brze hrane
Znanstvena grana rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rada	Kreirati web aplikaciju za vođenje restorana brze hrane s voditeljem i korisnicima koji naručuju hranu. Voditelj treba biti glavni administrator koji postavlja jelovnik, s potpunim uvidom u narudžbe korisnika, vrijeme narudžbe, cijena, posjećenost aktivnih korisnika, vođenje statistike pojedinog korisnika i hrane. Posjetitelji stranice imaju uvid u jelovnik, ali samo registrirani posjetitelji mogu raditi narudžbe. Korisnik treba imati uvid u jelovnik, mogućnost kreiranja narudžbe, komentari s ocjenom pojedinog jela i usluge koja im se pruža, npr. brzina dostave hrane.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	14.09.2020.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis: Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 29.09.2020.

Ime i prezime studenta:

Robert Horvat

Studij:

Prediplomski stručni studij Elektrotehnika, smjer Informatika

Mat. br. studenta, godina upisa:

AI4545, 25.09.2019.

Turnitin podudaranje [%]:

5

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za restoran brze hrane**

izrađen pod vodstvom mentora Robert Šojo

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD	1
1.1 Zadatak završnog rada	1
2. PRIMIJENJENE TEHNOLOGIJE I ALATI.....	3
2.1 HTML i CSS.....	3
2.2 JavaScript.....	3
2.3 PHP.....	4
2.4 XAMPP	4
2.5 SQL i phpMyAdmin.....	5
2.6 Visual Studio Code.....	5
3. PROCES IZRADE APLIKACIJE	7
3.1 Dizajn web aplikacije	7
3.1.1 Naslovna traka.....	7
3.1.2 Podnožje	9
3.1.3 Prijava i registracija - dizajn.....	10
3.1.4 Jelovnik	11
3.2 Funkcionalnosti web stranice	13
3.2.1 Baza podataka	13
3.2.2 Spajanje s bazom podataka	14
3.2.3 Prijava i registracija – funkcionalnost.....	16
3.2.4 Jelovnik i narudžba.....	19
3.2.5 Unos jela u jelovnik.....	26
3.2.6 Komentari.....	28
3.2.7 Statistika	31
4. UPUTE ZA KORIŠTENJE.....	39
4.1 Registracija	39
4.2 Prijava.....	40
4.3 Naručivanje jela.....	40
4.4 Forum i komentari	41
4.5 Statistika	41
5. ZAKLJUČAK	42
LITERATURA.....	43
SAŽETAK.....	44

ABSTRACT	45
ŽIVOTOPIS	46
PRILOZI.....	47

1. UVOD

U današnjem svijetu jako puno ljudi naručuje brzu hranu tako što u samom restoranu brze hrane naruče određeno jelo ili tako što nazovu dostavu hrane. Da bi se olakšao takav pristup naručivanju brze hrane, danas se sve više koriste web aplikacije. Stvorila se velika potreba za boljom i funkcionalnijom, ali opet jednostavnom web aplikacijom koja će biti prihvatljiva korisniku. Kako u Hrvatskoj ima puno restorana brze hrane, mnoge nemaju mogućnost naručivanja jela online, te bi im ovakva stranica dobro došla. Ta potreba za naručivanjem preko interneta došla je zbog toga što je životni standard postao sve brži, naručivanje preko telefona jednostavno više nije opcija. Zbog toga bi web aplikacija za naručivanje brze hrane bila dobra opcija jer bi se korisniku omogućio bolji uvid u ponudu i odabir, te jednostavnost naručivanja, a administratoru te web aplikacije od samog restorana uvid u korisnikovu narudžbu, kao i ostale narudžbe od ostalih korisnika. Isto tako korisnik bi imao mogućnost uvida u svoje narudžbe. Administrator bi imao uvid u statistiku restorana, dok bi korisnik imao uvid u vlastitu statistiku. Sama web aplikacija nudi uvid korisnicima mišljenje ostalih korisnika o pojedinom odabiru hrane.

Web aplikacija će imati razne mogućnosti od registracije i prijave korisnika, mogućnost uvida u jelovnik, izbor i naručivanje hrane, te na kraju sekcija komentara i uvida u komentare na hranu i sama ocjena hrane i usluge. Za kreiranje aplikacije, koriste se različite web tehnologije. Aplikacija ima bazu podataka kako bi se pohranile bitne informacije nužne za rad same aplikacije. Web aplikacija koristit će sljedeće tehnologije za web programiranje: SQL jezik, te stranica *PHPmyadmin* za bazu podataka, HTML i CSS za *frontend* dio odnosno izgled same aplikacije, *JavaScript* za određene funkcionalnosti vezane uz izgledi aplikacije odnosno samu aplikaciju, te PHP kao *backend* dio odnosno za komuniciranje s bazom podataka. Ovaj završni rad razrađen je u tri cjeline.

U prvom poglavlju pod nazivom „*Primijenjene tehnologije i alati*“ ukratko su opisane tehnologije i alati korišteni za izradu aplikacije. U drugom poglavlju pod nazivom „*Proces izrade aplikacije*“ opisan je način izrade aplikacije, izgled dizajna, te opis korištenoga koda. U trećem poglavlju pod nazivom „*Upute za korištenje*“ ukratko je upisano kako se koristi web aplikacija.

1.1 Zadatak završnog rada

Kreirati web aplikaciju za vođenje restorana brze hrane s voditeljem i korisnicima koji naručuju hranu. Voditelj treba biti glavni administrator koji postavlja jelovnik, s potpunim uvidom

u narudžbe korisnika, vrijeme narudžbe, cijena, posjećenost aktivnih korisnika, vođenje statistike pojedinog korisnika i hrane. Posjetitelji stranice imaju uvid u jelovnik, ali samo registrirani posjetitelji mogu raditi narudžbe. Korisnik treba imati uvid u jelovnik, mogućnost kreiranja narudžbe, komentari s ocjenom pojedinog jela i usluge koja im se pruža, npr. brzina dostave hrane.

2. PRIMIJENJENE TEHNOLOGIJE I ALATI

Kako bi se kreirala ovakva stranica moraju se koristiti razne tehnologije za izradu web stranice. Neke od primijenjenih web tehnologija koje će se koristiti su HTML, CSS, JavaScript, SQL i PHP. Za pomoć u korištenju tih web tehnologija koriste se alate poput *Visual Studio Code*, XAMPP i phpMyAdmin. Stranica ima svoj korisnički dio *frontend*, te svoj pozadinski dio *backend*. Korisnički dio web stranice će biti *responzivan*, što znači da će imati prilagodljivu širinu sadržaja. Stranica mora biti dizajnerski dovoljno jednostavna i pristupačna za korisnika. *Backend* dio će morati imati svoju bazu podataka za samog korisnika i za narudžbu korisnika, a isto tako i za meni.

2.1 HTML i CSS

HTML (engl. *HyperText Markup Language*) je prezentacijski jezik za izradu web stranica. HTML je opće prihvaćen kao jezik, te se gotovo svugdje koristi, a razlog toga je njegova jednostavnost i lakoća učenja. HTML nije programski jezik, te ga ne koriste samo programeri. Prikaz samog HTML odnosno *hiperteksta* omogućuje web preglednik. Prema [1] sama zadaća HTML-a je prikaz *hiperteksta* i sadržaja u web pregledniku. U HTML-u ne može se izvršiti nikakve matematičke operacije i tome slično. HTML datoteka je obična tekstualna datoteka sa ekstenzijom `.html`.

CSS (engl. *Cascading Style Sheets*) služi kao stilski jezik za uređivanje HTML dokumenta. Prema [2] CSS opisuje kako su HTML elementi prikazani na ekranu. Pomoću CSS može se uređivati sami tekst i izgled web stranice. Tekst se može uređivati tako da mu se mijenja veličina, font, boja, boja pozadine teksta i uređivanje margina i unutarnja margina. Isto tako se može uređivati veličina slike i videa koja su postavljena u web stranici.

2.2 JavaScript

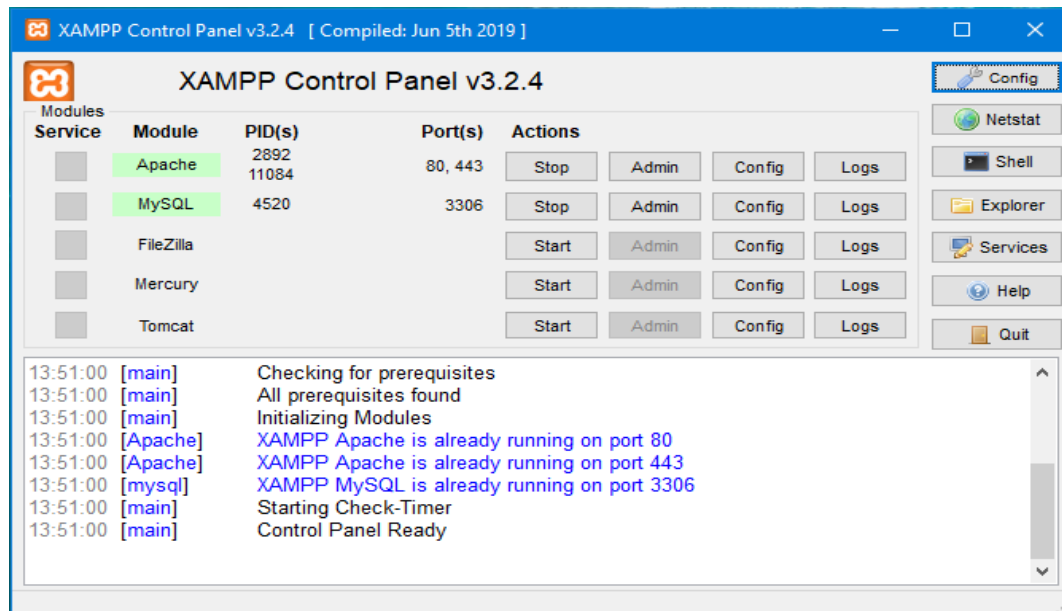
Prema [3] JavaScript predstavlja skriptni programski jezik koji se izvršava u samom web pregledniku na strani korisnika. JavaScript se koristi za izvršavanje same funkcionalnosti unutar samog web preglednika poput raznih provjera prilikom prijave i registracije, razne izračune što ne može sam HTML odraditi poput matematičkih operacija.

2.3 PHP

PHP (engl. *Hypertext Preprocessor*) predstavlja programski jezik koji je namijenjen prvenstveno programiranjem dinamičkih web stranica. Prema [4] PHP ima sličnu sintaksu kao C i Pearl, ali se razlikuje po što tome PHP ima HTML s ugrađenim kodom koji radi određeni posao. PHP se koristi za rad sa bazom podatak, odnosno da bi se povezala bazu podataka sa samom web stranicom. Jedna od namjena korištenja PHP-a je funkcionalnost prijave i registracije korisnika preko baze podataka.

2.4 XAMPP

Prema [5] XAMPP je besplatna više-platformska aplikacija s kojom se instaliraju i omogućuju korištenje Apache, MySQL, PHP, phpMyAdmin koje se koristi pri izradi dinamičke web stranice. Može se koristiti na više različitih platformi poput MAC-a, Windowsa i Linuxa. Slikom 2.1. prikazana je kontrolna ploča programa XAMPP. Na kontrolnoj ploči se preko tipke start pokreće Apache i MySQL.

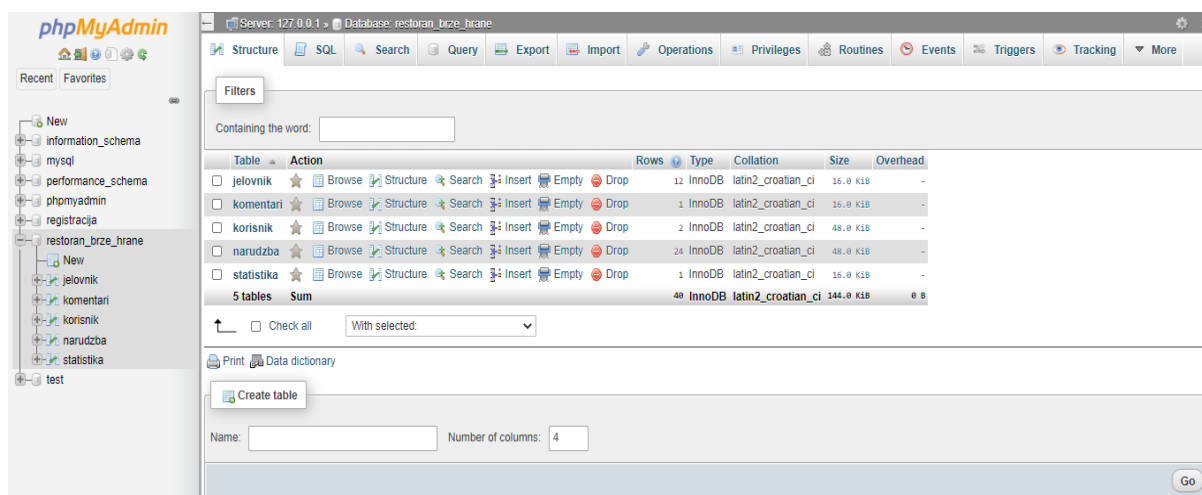


Sl. 2.1. Prikaz kontrolne ploče programa XAMPP.

2.5 SQL i phpMyAdmin

Prema [6] SQL (engl. *Structured Query Language*) predstavlja upitni jezik koji se koristi za izradu i rad s podacima iz relacijske baze podataka. SQL je programski jezik visoke razine. Sa SQL jezikom možemo izvršiti izradu, ažuriranje, traženje i brisanje podataka iz relacijske baze podataka. Prilikom izrade web aplikacije SQL će se koristiti za izradu baze podataka koja će imati bazu podataka za korisnika, te bazu podataka za narudžbu i ponudu.

Prema [7] *phpMyAdmin* je besplatni programski alat napisan u PHP-u, koji će se koristiti za izradu baze podataka. Dosta je jednostavan za korištenje, te se uglavnom koristi za izradu web stranica. Prema slici 2.2 vidi se prikaz web stranice i programskog alata *phpMyAdmin*.



SI. 2.2. Prikaz web stranice i programskog alata *phpMyAdmin*.

2.6 Visual Studio Code

Prema [8] *Visual Studio Code* je program za pisanje koda za razne programske jezike. On je napravljen od strane Microsofta, te ga koristi Windows, Linux i macOS. *Visual Studio Code* će se koristiti prilikom izrade naše web stranice, tako što će se u njemu pisati kod za HTML, CSS, *JavaScript* te PHP. Prema slici 3.3. prikazan je program *Visual Studio Code*.

```
1 <?php
2 // Start the session
3 session_start();
4 include 'statistika.php';
5 require('spoj.php');
6 require('spojZaPrijavuReg.php');
7 statistikaKupnoKupnoPosjetaWebStraniciUnos($conn);
8 statistikaKupnoZaradaUnos($conn);
9 ?>
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <title id = "title">Restoran Feast the beast</title>
14 <link rel="stylesheet" type="text/css" href="still.css">
15 <meta charset="utf-8">
16 <meta name="viewport" content="width=device-width, initial-scale=1">
17 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
18 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
19 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
20 <div id = "naslovni_dio" class = "container-fluid">
21 <h1 id="naslov">Restoran Feast the beast</h1>
22 <?php
23 if(isset($_SESSION['Korisnicko_ime'])){
24     echo "<a class='btn btn-primary btn' id ='btn2' href='odjava.php' role='button'>Odjava";
25     echo "<a class='btn btn-primary btn' id ='btn2' href='profil.php' role='button'>Profil";
26     echo "<p id ='btn2'>".$_SESSION['Korisnicko_ime'];</p>";
27 }
28 ?>
29 </div>
30 <style>
31     body {
32         background-color: #d6ffb7;
33     }
```

Sl. 2.3. Prikaz programa Visual Studio Code.

3. PROCES IZRADE APLIKACIJE

Kada bi se izradila ovakva jedna web aplikacija morati će se krenuti od izrade dizajna same stranice, šta će ona sadržavati, kako će izgledati i na kraju kako će se zvati. Stranica mora imati svoju navigacijsku ploču na kojoj će imati poveznice na ostale stranice. Taj dio dizajna radit će se raditi u HTML-u i CSS-u. Kako bi omogućili naručivanje hrane, aplikacija će morati imati svoju prijavu i registraciju. Kako bi sam unos e-maila i lozinke omogućili, potrebno je koristiti *JavaScript*. Kako bi se napravila prijava i registracija morati će se kreirati baza podataka za korisnika, a isto tako i za narudžbu. Bazu podataka će se kreirati u phpMyAdmin-u, a rad sa tom bazom podataka omogućeno je preko PHP-a. Isto tako kako bi se omogućili komentari mora se napraviti baza podataka i za taj dio.

Prilikom izrade web aplikacije korišteni su sljedeći alati i programi:

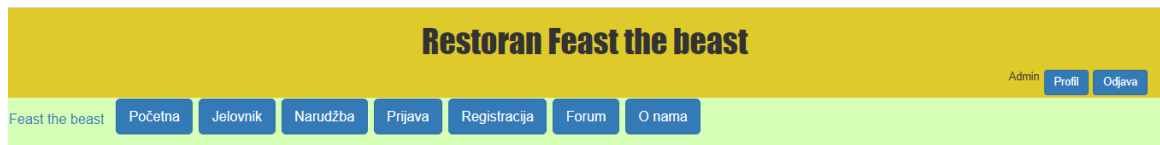
- *XAMPP control panel* – stvaranje lokalnog servera (Apache) i virtualnog servera za povezivanje na bazu podataka
- *phpMyAdmin* – izrada baze podataka zajedno sa tablicama, te unos podataka u tablice
- *Visual Studio Code* -program za pisanje koda u kojemu je pisan kod za HTML, CSS, PHP, JavaScript
- Web preglednik – Google Chrome web preglednik korišten za pregled izvođenja same web stranice i njezine funkcionalnosti.

3.1 Dizajn web aplikacije

Dizajn web aplikacije predstavlja izgled same web aplikacije. Dizajn se postigao preko HTML-a, CSS-a, *JavaScript-a*.

3.1.1 Naslovna traka

Kao i svaka druga web stranica i ova web stranica mora sadržavati naslovnu traku s izbornikom, u naslovnoj traci se nalazi ime web stranice i izbornik. Prema slici 3.1. vidi se prikaz naslovne trake s izbornikom.



Sl. 3.1. Prikaz naslovne trake s menijem.

Prilikom izrade naslovne koristila se opcija *container-fluid*, te se preko CSS-a uredila pozicija i font naslova i boja naslovne trake. U naslovnu traku je ubačeno korisničko ime, tipka za profil i odjavu koji su pomaknuti u desnu stranu. Slikom 3.2. prikazan je HTML kod za izradu naslovne trake. Unutar naslovne trake se nalazi PHP kod koji provjerava da li je prijavljen korisnik, te ako je ispisuje se tipka za profil, odjava i korisničko ime.

```
<div id = "naslovni_dio" class = "container-fluid">
  <h1 id="naslov">Restoran Feast the beast</h1>
  <?php
    if(isset($_SESSION['Korisnicko_ime'])){
      echo "<a class='btn btn-primary btn' id ='btn2' href='odjava.php' role='button'>Odjava</a>";
      echo "<a class='btn btn-primary btn' id ='btn2' href='profil.php' role='button'>Profil</a>";
      echo "<p id ='btn2'>{$_SESSION['Korisnicko_ime']}<p>";
    }
  ?>
</div>
```

Sl. 3.2. Prikaz HTML koda za izradu naslovne trake.

Slikom 3.3. prikazan je CSS kod za izradu naslovne trake. Odabrala se boja za naslovnu traku, te kako će biti postavljen naslov i kojeg će fonta biti naslov.

```
# stil1.css > #naslov
1 /*IZGLED STRANICE*/
2
3 #naslovni_dio{
4   background-color: #dfca2c;
5 }
6
7 #naslov{
8   text-align: center;
9   font-family: "Impact";
10  font-size: 300%;
11 }
```

Sl. 3.3. Prikaz CSS koda za izradu naslovne trake.

Navigacijska traka se radila preko opcije *navbar-light* u kojoj su se nalazile tipke s poveznicama za ostale dijelove web stranice. Slikom 3.4. u navigacijskoj traci poveznica za narudžbu će se pokazati onda ako je korisnik prijavljen. Da bi to bilo odrađeno mora se izvršiti

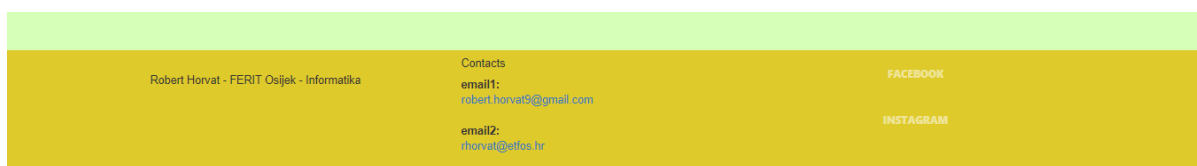
dio PHP koda gdje je postavljen uvjet ako je prijavljen korisnik onda se pokazuje poveznica za narudžbu.

```
<body>
<div id="container">
  <div class="row">
    <nav class="navbar navbar-light bg-light">
      <div class="container-fluid">
        <div class="navbar-header">
          <a class="navbar-brand" href="index.php">Feast the beast</a>
        </div>
        <ul class="nav navbar-nav">
          <a class="btn btn-primary btn-lg" href="index.php" role="button">Početna</a>&nbsp;
          <a class="btn btn-primary btn-lg" href="jelovnik.php" role="button">Jelovnik</a>&nbsp;
          <?php
            if(isset($_SESSION['Korisnicko_ime'])){
          >
          <a class="btn btn-primary btn-lg" href="narudzba.php" role="button">Narudžba</a>&nbsp;
          <?php
            }
          >
          <a class="btn btn-primary btn-lg" href="prijava.php" role="button">Prijava</a>&nbsp;
          <a class="btn btn-primary btn-lg" href="registracija.php" role="button">Registracija</a>&nbsp;
          <a class="btn btn-primary btn-lg" href="forum.php" role="button">Forum</a>&nbsp;
          <a class="btn btn-primary btn-lg" href="o_nama.php" role="button">O nama</a>&nbsp;
        </ul>
      </div>
    </nav>
  </div>
</div>
```

Sl. 3.4. Prikaz HTML koda za navigacijsku traku.

3.1.2 Podnožje

Kako aplikacija mora imati naslovnu traku, tako mora imati i podnožje koje će sadržavati osnovne podatke web stranice. Prema slici 3.5. u podnožju se mogu vidjeti osnovne podatke i kontakt. Raspoređeni su tako da budu u 3 stupca. Boja je ista kao i kod naslovne trake.



Sl. 3.5. Prikaz podnožja.

Slikom 3.6. prikazuje se HTML kod za podnožje. Boja podnožja se odredila u HTML kodu, na početku u oznaci *footer*, preko atributa *style*. Isto tako se i odredila veličina svake sekcije.

```

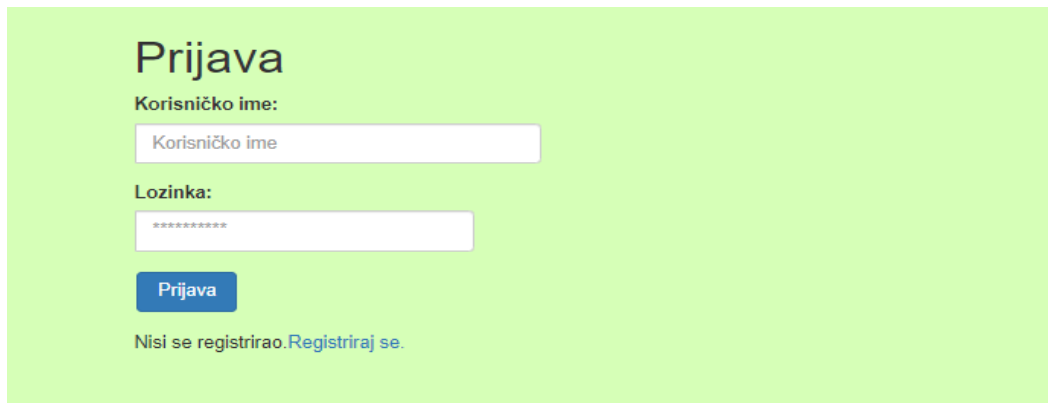
<footer class="section footer-classic context-dark bg-image" style="background: #dfca2c;">
  <div class="container">
    <div class="row row-30">
      <div class="col-md-4 col-xl-5">
        <p><br></p>
        <p>Robert Horvat - FERIT Osijek - Informatika</p>
        <p><br></p>
      </div>
      <div class="col-md-4">
        <h5>Contacts</h5>
        <dl class="contact-list">
          <dt>email1:</dt>
          <dd><a href="mailto:#">ime.prezime1@gmail.com</a></dd>
        </dl>
        <dl class="contact-list">
          <dt>email2:</dt>
          <dd><a href="mailto:#">ime.prezime2@gmail.com</a></dd>
        </dl>
      </div>
      <div class="col-md-4">
        <div class="col">
          <a class="social-inner" href="https://www.facebook.com/robert.horvat.146">
            <span class="icon mdi mdi-facebook"></span><span>Facebook</span></a>
        </div>
        <div class="col">
          <a class="social-inner" href="https://www.instagram.com/robert.horvat3/">
            <span class="icon mdi mdi-instagram"></span><span>instagram</span></a>
        </div>
      </div>
    </div>
  </div>
</footer>

```

Sl. 3.6. Prikaz HTML koda za podnožje.

3.1.3 Prijava i registracija - dizajn

Izgled prijave i registracije je identičan, razlika je u tome što registracija ima više polja za unos podataka koje se unose u bazu podataka. Slikom 3.7 prikazan je izgled prijave. Izgled same prijave je vrlo jednostavan i uklapa se uz samu pozadinu. Sadrži dva tekstualna okvira gdje se u prvi tekstualna okvir piše korisničko ime, a u drugi tekstualni okvir se piše lozinka koja je sakrivena. Ispod tekstualnih okvira nalazi se tipka za prijavu i poveznica za registraciju u slučaju da se korisnik još nije registrirao.



Sl. 3.7. Prikaz prijave.

Prema slici 3.8 može se vidjeti dio PHP koda koji pokazuje ukoliko se u sesiji nalazi korisničko ime od korisnika koji je prijavljen prikazati će se samo njegovo ime i poveznica za odjavu. Tekstualni okviri napravljeni su u tablici po odvojenim grupama.

```

<div class="container">
  <h1>Prijava</h1>
  <?php
    if(isset($_SESSION['Korisnicko_ime'])){
      echo "Pozdrav: {$_SESSION['Korisnicko_ime']}";
      echo "<br/><a href = 'odjava.php'>Odjava</a>";
    }else{
  ?>
  <form action="prijava.php" method="post">
    <table>
      <div class="form-group">
        <div class="input-group">
          <label for="name">Korisničko ime: </label>
          <input id="name" name="Korisnicko_ime" placeholder="Korisničko ime" type="text" class="form-control">
        </div>
      </div>
      <div class="form-group">
        <div class="input-group">
          <label for="password">Lozinka: </label>
          <input id="password" name="Lozinka" placeholder="*****" type="password" class="form-control">
        </div>
      </div>
      <div class="form-group">
        <div class="input-group">
          <button type="submit" class="btn btn-block btn-primary" id="prijavaBtn" name="btn-signup">Prijava</button>
        </div>
      </div>
    </table>
  </form>
  <p>Nisi se registrirao.<a href="registracija.php">Registriraj se.</a></p>
  <?php
    echo $_SESSION
  
```

Sl. 3.8. Prikaz HTML koda za prijavu.

3.1.4 Jelovnik

Jelovnik je napravljen po kategorijama tako da svaka kategorija ima svoj padajući izbornik u kojemu se nalaze jela zajedno s cijenom. Jedino registrirani korisnik može vidjeti tipku za

dodavanje u košaricu, te može unesti količinu pojedinog jela koje želi naručiti. Prema slici 3.9. prikazan je jelovnik.



Sl. 3.9. Prikaz jelovnika.

Svako jelo zajedno s cijenom u svakoj kategoriji je dohvaćeno iz baze podataka po kategoriji u kojoj pripada. Svaka kategorija je odvojena zasebno. Funkcija padajućeg izbornik je napravljen preko *JavaScript-a*, a sami izgled padajućeg izbornika je odrađen preko HTML-a i CSS-a. Prema slici 3.10. može se vidjeti da su se jela dohvatila pomoću PHP koda.

```
<div class="dropdown">
  <button onClick="myFunction1()" class="dropbtn">Burgeri</button>
  <div id="myDropdown1" class="dropdown-content">
    <ul class="list-group">
      <?php
        $product_array1 = $db_handle->runQuery("SELECT * FROM `jelovnik` WHERE `Kategorija` = 'Burgeri'");
        if (empty($product_array1)) {
          foreach($product_array1 as $key=>$value){
            <li class="list-group-item">
              <form method="post" action="jelovnik2.php?action=add&ID=<?php echo $product_array1[$key][\"ID\"]; ?>">
                <?php echo $product_array1[$key][\"Naziv_jela\"]; ?>.....<?php echo "$".$product_array1[$key][\"Cijena\"]; ?>
                <?php
                  if(isset($_SESSION['Korisnicko_ime']))
                    echo " <div class='cart-action'><input type='text' class='product-quantity' name='quantity' value='1' size='2' />
                    <form action='jelovnik2.php?action=add' method='post'><input name='addBtn' type='submit' value='Add'></form></div>";
                }
            }
          }
        }
      </li>
    </ul>
  </div>
</div>
```

Sl. 3.10. Prikaz HTML koda za jelovnik.

Prema slici 3.11. vidi se prikaz *JavaScript* koda za padajući izbornik gdje se nalaze 5 funkcija za padajući izbornik. Svaka od tih funkcija dohvaća drugačiji *id* od pojedinog odjeljka gdje se prikazuju pojedine kategorije s jelima.

```

<script>
function myFunction1() {
    document.getElementById("myDropdown1").classList.toggle("show");
}
function myFunction2() {
    document.getElementById("myDropdown2").classList.toggle("show");
} f
function myFunction3() {
    document.getElementById("myDropdown3").classList.toggle("show");
}
function myFunction4() {
    document.getElementById("myDropdown4").classList.toggle("show");
}
function myFunction5() {
    document.getElementById("myDropdown5").classList.toggle("show");
}
</script>

```

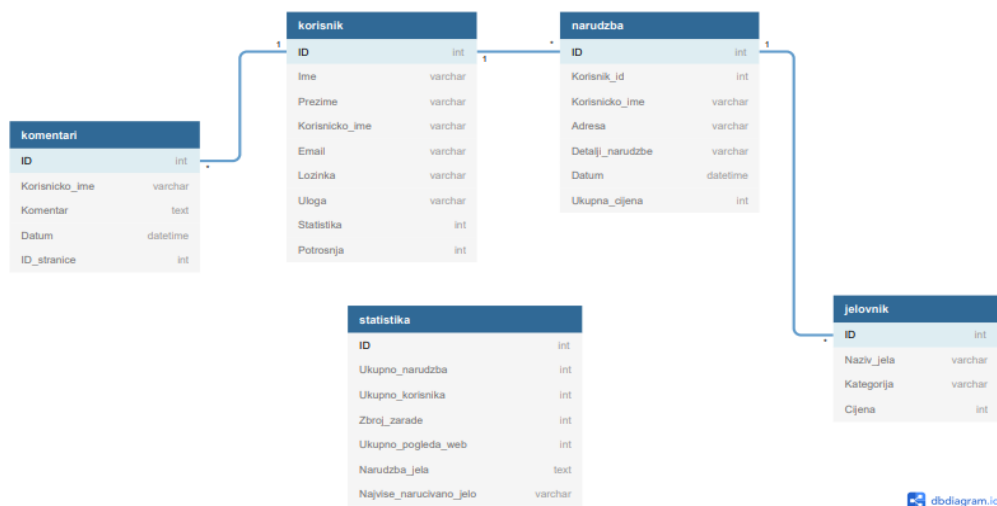
Sl. 3.11. Prikaz JavaScript koda za padajući izbornik.

3.2 Funkcionalnosti web stranice

Da bi sama web aplikacija imala smisla potrebno je bilo napraviti funkcionalnost koja je odrađena u PHP-u. Dio manje funkcionalnosti je prikazan pod dizajnom web stranice. Prije same funkcionalnosti potrebno je bilo napraviti i bazu podataka s tablicama. Pod funkcionalnost ove web stranice smatra se prijava, registracija, odjava, jelovnik s narudžbom i košaricom, dohvaćanje narudžbe, ubacivanje jela u jelovnik od strane administratora, komentari u forumu, te statistika restorana i korisnika.

3.2.1 Baza podataka

Da bi sama funkcionalnost radila ispravno, te imala smisla potrebno je imati bazu podataka. Baza podataka predstavlja spremište u koje se unose podatci iz stvarnog svijeta, u ovom slučaju preko web aplikacije. Slikom 3.12. prikazan je ER dijagram za bazu podataka koji je napravljen preko web aplikacije *dbdijagram.io*.



Sl. 3.12. Prikaz baze podataka preko ER dijagrama.

Slikom 3.13. prikazuje se baza podataka s imenima tablica koja se koristila prilikom izrade web aplikacije.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> jelovnik	☆ Browse Structure Search Insert Empty Drop	12	InnoDB	latin2_croatian_ci	16.0 KiB	-
<input type="checkbox"/> komentari	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	latin2_croatian_ci	16.0 KiB	-
<input type="checkbox"/> korisnik	☆ Browse Structure Search Insert Empty Drop	2	InnoDB	latin2_croatian_ci	48.0 KiB	-
<input type="checkbox"/> narudzba	☆ Browse Structure Search Insert Empty Drop	24	InnoDB	latin2_croatian_ci	48.0 KiB	-
<input type="checkbox"/> statistika	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	latin2_croatian_ci	16.0 KiB	-
5 tables	Sum	40	InnoDB	latin2_croatian_ci	144.0 KiB	0 B

Sl. 3.13. Prikaz baze podataka preko phpMyAdmin.

3.2.2 Spajanje s bazom podataka

Nakon napravljene baze podataka za pojedinu funkcionalnost potrebno je bilo spojiti aplikaciju sa bazom podataka. Spajanje s bazom podataka se odradio preko PHP-a. Prilikom izrade web aplikacije bila su napravljena dva spoja s bazom podataka, jedno spajanje je napravljeno na proceduralan način, dok je drugo spajanje napravljeno na objektno-orijentirani način. Proceduralni način se koristio prilikom prijave i registracije, dok se objektno-orijentirani način koristio za jelovnik, narudžbu s košaricom, dohvaćanje narudžbe i statistike. Slikom 3.14. prikazuje se PHP kod prilikom spajanja odrađen proceduralnim načinom. Da bi se odradilo spajanje, potrebo je inicijalizirati varijable koje su nam potrebne prilikom spajanja. Varijable koje se inicijaliziraju su

ime servera, korisničko ime koje se koristi u serveru, lozinku i najvažnije ime baze podataka. Naziv „localhost“ kod imena servera znači da se web aplikacija pokreće lokalno. Preko varijable „conn“ se povezujemo na samu bazu podataka koja se koristi.

```
spojZaPrijavuReg.php
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "restoran_brze_hrane";
6
7
8  $conn = mysqli_connect($servername, $username, $password, $dbname);
9
10 if (!$conn) {
11     die("Connection failed: " . mysqli_connect_error());
12 }
13
14
```

Sl. 3.14. Prikaz proceduralnog načina spajanja na bazu podataka.

Kao što se može vidjeti na slici 3.15. kod objektno-orijentiranog načina spajanja, inicijalizacija varijabli je identična uz dodatak privatnog prava pristupa. Nakon toga se nalazi funkcija konstruktora koja je bitna zbog stvaranja objekata na funkciju spoja. Poslije toga ide funkcija koja prima zahtjev, te šalje zahtjev za podacima iz baze podataka te ih sprema u varijable koje će se koristiti. Sljedeća funkcija služi za prebrojavanje redova u tablici iz baze podataka.

```
spoj.php
1  <?php
2  class DBController {
3      private $servername = "localhost";
4      private $username = "root";
5      private $password = "";
6      private $dbname = "restoran_brze_hrane";
7      private $conn;
8
9      function __construct() {
10         $this->conn = $this->connectDB();
11     }
12
13     function connectDB() {
14         $conn = mysqli_connect($this->servername,$this->username,$this->password,$this->dbname);
15         return $conn;
16     }
17
18     function runQuery($query) {
19         $result = mysqli_query($this->conn,$query);
20         while($row=mysqli_fetch_assoc($result)) {
21             $resultset[] = $row;
22         }
23         if(!empty($resultset))
24             return $resultset;
25     }
26
27
28     function numRows($query) {
29         $result = mysqli_query($this->conn,$query);
30         $rowcount = mysqli_num_rows($result);
31         return $rowcount;
32     }
33 }
```

Sl. 3.15. Prikaz objektno-orijentiranog načina spajanja na bazu podataka.

3.2.3 Prijava i registracija – funkcionalnost

Prije izrade prijave i registracije bilo je potrebno napraviti u bazi podataka tablicu za korisnika koja će sadržavati sve moguće podatke da bi se isti mogao prijaviti i koristiti samu web aplikaciju. Tablica se zove *korisnik* koja je prikazana na slici 3.16.. Tablica korisnik sadrži svoj primarni ključ ID koji se povećava svaki put kada se registrira novi korisnik. Entitet Uloga ima zadanu vrijednost „Korisnik“ da bi se mogla definirati koja prava ima korisnik, a koja administrator. Administrator je posebno naveden u bazi podataka i zadana mu je vrijednost „Administrator“. Dodatni entiteti su Statistika i Potrošnja koji su naknadno ubačeni, te predstavljaju ukupan broj narudžbi korisnika, te koliko ima ukupnu potrošnju od tih narudžbi. Oni su postavljeni na zadanu vrijednost 0, zbog računanja samih vrijednosti.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 ID	int(250)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 Ime	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	3 Prezime	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	4 Korisnicko_ime	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	5 Email	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	6 Lozinka	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/>	7 Uloga	varchar(250)	latin2_croatian_ci		No	Korisnik			Change Drop More
<input type="checkbox"/>	8 Statistika	int(250)			No	0			Change Drop More
<input type="checkbox"/>	9 Potrosnja	int(250)			No	0			Change Drop More

Sl. 3.16. Prikaz tablice korisnika u bazi podataka preko phpMyAdmin.

Prije same prijave potrebno je napraviti registraciju kako bi se podatci od korisnika postavili u bazu podataka. Slikom 3.17. prikazuje se PHP kod za registraciju. Kod je napisan prema [9]. Prvo što se mora napraviti je zahtjev za PHP dokument gdje se nalazi spajanje koje će se koristiti za registraciju. Nakon toga u uvjetu *if* uvrštava se provjera da li je upisao korisničko ime u polju za korisničko ime. Onda se u varijable unosi vrijednosti koje su upisane u polja. Na kraju slijedi SQL zahtjev za unos u bazu podataka gdje se unose te vrijednosti. Kada se pošalje SQL zahtjev slijedi provjera da li je ispravno poslan zahtjev, te ako je ispisat će se korisnik uspješno registrirao i ispisati će se poveznica za prijavu.

```

registracija.php
91 <?php
92 require('spojzaPrijavuReg.php');
93
94 if(isset($_REQUEST['Korisnicko_ime'])){}
95
96     $Ime = stripslashes($_REQUEST['Ime']);
97     $Ime = mysqli_real_escape_string($conn, $Ime);
98
99     $Prezime = stripslashes($_REQUEST['Prezime']);
100    $Prezime = mysqli_real_escape_string($conn, $Prezime);
101
102    $Korisnicko_ime = stripslashes($_REQUEST['Korisnicko_ime']);
103    $Korisnicko_ime = mysqli_real_escape_string($conn, $Korisnicko_ime);
104
105    $Email = stripslashes($_REQUEST['Email']);
106    $Email = mysqli_real_escape_string($conn, $Email);
107
108    $Lozinka = stripslashes($_REQUEST['Lozinka']);
109    $Lozinka = mysqli_real_escape_string($conn, $Lozinka);
110
111    $query = "INSERT into korisnik (Ime, Prezime, Korisnicko_ime, Email, Lozinka)
112    VALUES ('$Ime', '$Prezime', '$Korisnicko_ime', '$Email', '$Lozinka')";
113    $result = mysqli_query($conn, $query);
114    if($result==true){
115        $Korisnik_id = $conn->insert_id;
116    }
117    if($result){
118        statistikaUkupnoKorisnikaUnos($conn);
119        echo "<div class = 'form'>
120        <h3>Uspješno si se registrirao.</h3>
121        <br/>Click here to <a href='prijava.php'>Prijava</a></div>";
122    }
123 }else{

```

Sl. 3.17. Prikaz PHP koda za registraciju.

Nakon registracije slijedi prijava gdje korisnik treba unijeti svoje korisničko ime i lozinku da bi se prijavio. Prilikom prijave potrebno je odrediti da li se prijavio korisnik ili administrator. Prema slici 3.18. i slici 3.19. može se vidjeti način na koji se provjerava da li se prijavio korisnik ili administrator. Prema slici 3.18. prikazuje se način kako se pri prijavi provjerava da li je prijavljen administrator. Ako je prijavljen administrator onda se u varijable unose podatci od administratora koji su izvučeni iz SQL zahtjeva. Nakon toga se u sesiju upisuju ti podatci da bi se kasnije lakše moglo provjeriti da li je korisnik ili administrator.

```

<?php
$error = "";

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

    $success=false;

    $Korisnicko_ime = stripslashes($_REQUEST['Korisnicko_ime']);
    $Korisnicko_ime = mysqli_real_escape_string($conn,$Korisnicko_ime);
    $Lozinka = stripslashes($_REQUEST['Lozinka']);
    $Lozinka = mysqli_real_escape_string($conn,$Lozinka);

$result = mysqli_query($conn, "SELECT * FROM korisnik WHERE Korisnicko_ime='$Korisnicko_ime' and Lozinka='$Lozinka' and Uloga='Administrator'");
$rows = mysqli_num_rows($result);

    while($row = mysqli_fetch_array($result)){
        $success = true;
        $Korisnik_id = $row['ID'];
        $Uloga= $row['Uloga'];
        $Korisnicko_ime = $row['Korisnicko_ime'];
    }
    if($success == true){
        if($rows == 1){
            $_SESSION['admin_id']=session_id();
            $_SESSION['Korisnik_id'] = $Korisnik_id;
            $_SESSION['Uloga'] = $Uloga;
            $_SESSION['Korisnicko_ime'] = $Korisnicko_ime;
            $_SESSION['Lozinka'] = $Lozinka;
            header("Location: index.php");
        }else{

```

Sl. 3.18. Prikaz PHP koda za prijavu -1.dio.

Prema slici 3.19. prikazuje se način kako se pri prijavi provjerava da li je prijavljen korisnik. Ako je prijavljen korisnik onda se ponavlja isti postupak kao i za administratora. Nakon prijave, korisnik mora imati mogućnost da se odjavi. Odjava se vrši preko poveznice koja je postavljena korisniku.

```

prijava.php
88         }else{
89             echo "<div class='form'>
90                 <h3>Korisničko ime i lozinka su netočni.</h3>
91                 <br/>Klikni ovdje za <a href='prijava.php'>Prijavu</a></div>";
92         }
93     }
94     if($success == false){
95         $query = "SELECT * FROM korisnik WHERE Korisnicko_ime = '$Korisnicko_ime' and Lozinka = '$Lozinka' and Uloga='Korisnik'";
96     }
97     $result = mysqli_query($conn,$query);
98     $rows = mysqli_num_rows($result);
99
100     while($row = mysqli_fetch_array($result)){
101         $success = true;
102         $Korisnik_id = $row['ID'];
103         $Uloga= $row['Uloga'];
104         $Korisnicko_ime = $row['Korisnicko_ime'];
105     }
106     if($success == true){
107         if($rows == 1){
108             $row = mysqli_fetch_assoc($result);
109             $_SESSION['kupač_id']=session_id();
110             $_SESSION['Korisnik_id'] = $Korisnik_id;
111             $_SESSION['Uloga'] = $Uloga;
112             $_SESSION['Korisnicko_ime'] = $Korisnicko_ime;
113             $_SESSION['Lozinka'] = $Lozinka;
114             header("Location: index.php");
115         }else{
116             echo "<div class='form'>
117                 <h3>Korisničko ime i lozinka su netočni.</h3>
118                 <br/>Klikni ovdje za <a href='prijava.php'>Prijavu</a></div>";
119         }

```

Sl. 3.19. Prikaz PHP koda za prijavu - 2.dio.

Slikom 3.20. prikazuje se PHP kod za odjavu. Sesija se mora uništiti, te se nakon toga korisnik vraća na početnu stranicu.

```
odjava.php
1  <?php
2  session_start();
3  unset($_POST);
4  session_destroy();
5  header('Location: index.php');
6  ?>
```

Sl. 3.20. Prikaz PHP koda za odjavu.

3.2.4 Jelovnik i narudžba

Nakon prijave i registracije korisnik treba imati mogućnost uvida u jelovnik i naručivanja jela iz jelovnika. Kao što je već prethodno prikazan izgled jelovnika gdje se jela iz jelovnika dohvaćaju iz baze podataka, ta ista jela se mogu naručiti. Da bi korisnik mogao vidjeti koje jelo želi naručiti i koliko će biti ukupna cijena, mora se prikazati košarica za kupnju (engl. *shopping cart*). Isto tako korisnik mora imati mogućnost uklanjanja pojedinog jela, te uklanjanja svih jela iz košarice. Slikom 3.21. prikazuje se izgled košarice za naručivanje. Košarica se izgledom uklapa uz samu pozadinu, te je napravljena u tablici.

Shopping Cart					
Naziv jela	ID	Količina	Cijena	Ukupna cijena	Ukloni
Tortilja s piletinom	3	1	18 kn	18.00 kn	Ukloni
Restani krumpir	6	1	12 kn	12.00 kn	Ukloni
Hot dog	8	1	15 kn	15.00 kn	Ukloni
Ukupna količina:		3			
Ukupna cijena:		45.00kn			
Isprazni					
Unesi adresu:					
<input type="text" value="Adresa"/>					
Naruči					

Sl. 3.21. Prikaz izgleda košarice.

Da bi se mogla prikazati košarica za svako to jelo mora postojati funkcionalnost dodavanja u košaricu. Sama košarica predstavlja polje u koje je uneseno svako odabrano jelo. Prilikom izrade PHP koda za dodavanje u košaricu koristi se objektno-orijentirani način programiranja.

Prema slici 3.22. svaka pojedina funkcionalnost za košaricu napravljena je preko *switch-case* uvjetovanja prema [10]. Za dodavanje u košaricu prvo se mora provjeriti da li je unesena količina nekog jela, ako je potrebno je poslati SQL zahtjev preko ID-a pojedinog jela koje je odabrano. Nakon toga se pojedini atributi poput naziva jela i cijene unose u polje s jelima. Kada se unesu atributi u polje onda se provjerava da li već postoje jela u košarici, ako postoje, onda se za svako jelo u košarici provjerava količina pojedinog jela. Ako jelo nije uneseno odnosno nema unesene količine, onda se količina postavlja na nulu, u suprotnom se količina povećava. Poslije funkcionalnosti za dodavanje u košaricu dolazi funkcionalnost za brisanje pojedinog jela iz košarice i brisanje svih jela iz košarice.

```
<?php
require_once('spoj.php');
$db_handle = new DBController();
if(!empty($_GET["action"])) {
    switch($_GET["action"]) {
        case "add":
            if(!empty($_POST["quantity"])) {
                $productByID = $db_handle->runQuery("SELECT * FROM jelovnik WHERE ID = '".$_GET["ID"]."'");
                $itemArray = array($productByID[0]["ID"]->array('Naziv_jela'=>$productByID[0]['Naziv_jela'], 'ID'=>$productByID[0]['ID'],
                if(!empty($_SESSION["cart_item"])) {
                    if(in_array($productByID[0]["ID"],array_keys($_SESSION["cart_item"]))) {
                        foreach($_SESSION["cart_item"] as $k => $v) {
                            if($productByID[0]["ID"] == $k) {
                                if(empty($_SESSION["cart_item"][$k]["quantity"])) {
                                    $_SESSION["cart_item"][$k]["quantity"] = 0;
                                }
                                $_SESSION["cart_item"][$k]["quantity"] += $_POST["quantity"];
                            }
                        }
                    }else{
                        $_SESSION["cart_item"] = array_merge($_SESSION["cart_item"],$itemArray);
                    }
                }else{
                    $_SESSION["cart_item"] = $itemArray;
                }
            }
        }
    }
    break;
}
```

SI. 3.22. Prikaz PHP koda za dodavanje u košaricu.

Slikom 3.23. je prikazana funkcionalnost za brisanje iz košarice i pražnjenje košarice. Da bi se pravilno izbrisalo pojedino jelo, mora sadržavati neki indeks po kojemu će se lakše dohvatiti to jelo koje je potrebno obrisati. Postavlja se varijabla u koju se unosi indeks po kome će se brisati pojedino jelo. Prvo se provjerava da li je samo jedno jelo postavljeno u košarici, ako je, onda ga izbacuje preko funkcije *unset* koja se koristi za uklanjanje sesije , ako je više jela u košarici onda mora dohvatiti jelo po indeksu za brisanje te ga ukloniti, te nakon toga sortirati košaricu. Prilikom pražnjenja cijela košarice dovoljno je samo izbaciti sva jela preko funkcije *unset* za uklanjanje

sesije. Kod za izgled košarice i funkcije unutar košarice poput zbrajanja ukupne cijene odrađuje se preko HTML-a i PHP-a.

```
case "remove":
    if(isset($_POST['index_to_remove']) && $_POST['index_to_remove'] != ""){
        $key_to_remove = $_POST['index_to_remove'];
        if(count($_SESSION["cart_item"]) <= 1){
            unset($_SESSION["cart_item"]);
        }else{
            unset($_SESSION["cart_item"][$key_to_remove]);
            sort($_SESSION["cart_item"]);
        }
    }
    break;
case "empty":
    unset($_SESSION["cart_item"]);
    break;
}
```

Sl. 3.23. Prikaz PHP koda za brisanje iz košarice.

Slikom 3.24. je prikazan početak sekcije za košaricu. Unutar PHP koda nalazi se *if* uvjetovanje ako postoji jelo u košarici onda ukupna količina i ukupna cijena postavlja na nulu, te se izvodi ostatak HTML koda. Nakon toga dolazi tablica gdje su prvo unesena imena stupaca.

```
<div id="shopping-cart">
<?php
    if(isset($_SESSION["cart_item"])){
        $total_quantity = 0;
        $Ukupna_cijena = 0;
    }
?>
<div class="txt-heading">Shopping Cart</div>
<table class="tbl-cart" cellpadding="10" cellspacing="1">
<tbody>
<tr>
<th style="text-align:left;"><b>Naziv jela</th>
<th style="text-align:left;"><b>ID</th>
<th style="text-align:right;" width="5%"><b>Količina</th>
<th style="text-align:right;" width="10%"><b>Cijena</th>
<th style="text-align:right;" width="10%"><b>Ukupna cijena </th>
<th style="text-align:center;" width="5%"><b>Ukloni</th>
</tr>
```

Sl. 3.24. Prikaz HTML i PHP koda za izradu košarice -1.dio.

U nastavku prema slici 3.25. u PHP kodu na početku je inicijalizirana vrijednost za detalje narudžbe prema [11]. Nakon toga se izvršava *foreach* petlja za svako jelo koje je ubačeno u košaricu kako bi se svako to jelo moglo prikazati u tablici. Svakoj vrijednosti se postavlja zadana

vrijednost iz polja gdje se nalaze jela, te se izračunava ukupna cijena za svako jelo po količini. Poslije toga se nastavlja tablica gdje se za svako ubačeno jelo prikazuje novi red u tablici sa zadani vrijednostima poput naziva jela, cijene, količine i opcije za brisanje jela iz tablice. Onda se izračunava ukupna količina i ukupna cijena. Detalji narudžbe se unose preko *stringa*, te se spajaju u novi *string* preko operatora za pridruživanje i funkcije za novi red. Detalji narudžbe se unose u bazu podataka, ali se ne prikazuju u tablici jer su već prethodno zasebno prikazani.

```

<?php
$i=0;
$Detalji_narudzbe="";
$Narudzba_jela="";
foreach ($SESSION["cart_item"] as $item){
    $Kolicina = $item["quantity"];
    $Cijena=$item["Cijena"];
    $Cijena_uk = $item["Cijena"];
    $Naziv_jela=$item["Naziv_jela"];
    $Jelo_id=$item["ID"];
    $Cijena_uk = $Kolicina*$Cijena_uk;
    ?>
    <tr>
    <td><?php echo "<p id='btn1' name='Naziv_jela' >{$Naziv_jela}<p>"; ?></td>
    <td><?php echo "<p id='btn1' name='Jelo_id' >{$Jelo_id}<p>"; ?></td>
    <td style="text-align:right;"><?php echo "<p id='btn1' name='Kolicina' >{$Kolicina}<p>"; ?></td>
    <td style="text-align:right;"><?php echo "<p id='btn1' name='Cijena' >{$Cijena} kn <p>"; ?></td>
    <td style="text-align:right;"><?php echo number_format($Cijena_uk,2)." kn "; ?></td>
    <td style="text-align:center;"><form action="jelovnik.php?action=remove" method="post"><input id="remove" name="deleteBtn" <?php ec
    </tr>
    <?php
    $i++;
    $total_quantity += $Kolicina;
    $Ukupna_cijena += ($Cijena_uk*$Kolicina);
    $Detalji_narudzbe .= nl2br("ID: {$Jelo_id} - Naziv: {$Naziv_jela} - Kolicina: {$Kolicina} - Cijena: {$Cijena} kn - Ukupna cijena jela: {$Cijena_uk} kn");
    $Narudzba_jela .= "{$Naziv_jela},";
}
?>

```

Sl. 3.25. Prikaz HTML i PHP koda za izradu košarice -2.dio.

U nastavku prema slici 3.26. prikazana su još dva reda u tablici za ukupnu količinu i ukupnu cijenu. Nakon tablice nalazi se tipka preko koje se cijela košarica isprazni. Poslije košarice nalazi se forma za unos adrese gdje će se dostaviti narudžba. U formi su još upisani na skriveni način ukupna cijena i detalji narudžbe. Na kraju se u formi nalazi tipka za naručivanje gdje se šalje cijela narudžba.

```

<tr>
  <td colspan="2" align="right"><b>Ukupna količina:</td>
  <td align="right"><?php echo $total_quantity; ?></td>
</tr>
<tr>
  <td colspan="2" align="right"><b>Ukupna cijena:</td>
  <td align="right" id="ukupno" colspan="2"><strong><?php echo number_format($Ukupna_cijena, 2)."kn"; ?></strong></td>
</tr>
</tbody>
</table>
<br>
<a id="btnEmpty" class="btn btn-primary" href="jelovnik.php?action=empty">Isprazni</a>
<br>
<div class="form" id="">
  <br>
  <p><b>Unesi adresu:</p>
  <form name="jelovnik2" action="" method="post">
    <input type="text" name="Adresa" id="adresa" placeholder="Adresa" required/>
    <input type="hidden" name="Ukupna_cijena" value="<?php echo $Ukupna_cijena;?>">
    <input type="hidden" name="Detalji_narudzbe" value="<?php echo $Detalji_narudzbe;?>">
    <input type="hidden" name="Narudzba_jela" value="<?php echo $Narudzba_jela;?>">
    <br><br>
    <button type="submit" class="btn btn-primary" id="naruciBtn" name="action">Naruči</button>
  </form>
  <?php
  } else {
  ?>
</div>

```

Sl. 3.26. Prikaz HTML i PHP koda za izradu košarice -3.dio.

Slikom 3.27. prikazuje se prvi dio PHP koda za narudžbu. Princip je vrlo sličan kao i kod registracije gdje se moraju inicijalizirati vrijednosti prije svih zahtjeva za PHP dokument gdje se nalazi spajanje na bazu. U uvjetu *if* nalazi se provjera da li je unesena adresa u formi. Unutar uvjeta *if* nalaze se inicijalizirane vrijednosti za unos u bazu podataka. Ukupna cijena se prvo morala postaviti na nulu da bi se naknadno dohvatila preko POST metode. Korisničko ime i ID se dohvatio preko sesije koja je unesena prilikom prijave. Adresa je izvučena iz forme, kao i detalji narudžbe. Zbog statistike su dodani dodatni dijelovi poput SQL zahtjeva i unosa u reda u varijablu koja će se naknadno koristiti u statistici za jela. Nakon svega toga dolazi SQL zahtjev za unos svih tih podataka u narudžbu.

```

jelovnik.php
256 <?php
257 require ('spojzaPrijavuReg.php');
258
259 if(isset($_POST['Adresa'])){
260     $Ukupna_cijena = 0;
261     $Korisnik_id=$_SESSION['Korisnik_id'];
262     $Korisnicko_ime=$_SESSION['Korisnicko_ime'];
263     $Adresa = htmlspecialchars($_POST['Adresa']);
264     $Adresa = mysqli_real_escape_string($conn, $Adresa);
265     $Ukupna_cijena = $_POST['Ukupna_cijena'];
266     $Detalji_narudzbe = htmlspecialchars($_POST['Detalji_narudzbe']);
267     $Detalji_narudzbe = stripslashes($_POST['Detalji_narudzbe']);
268     $Detalji_narudzbe = mysqli_real_escape_string($conn,$Detalji_narudzbe);
269     $queryJelo1 = "SELECT Narudzba_jela FROM statistika";
270     $rezultat = $conn->query($queryJelo1);
271     if(mysqli_query($conn, $queryJelo1)) {
272         while($row = $rezultat->fetch_assoc()){
273             $Narudzba_jela1 = $row['Narudzba_jela'];
274         }
275     }
276     $Narudzba_jela = htmlspecialchars($_POST['Narudzba_jela']);
277     $Narudzba_jela = stripslashes($_POST['Narudzba_jela']);
278     $Narudzba_jela = mysqli_real_escape_string($conn,$Narudzba_jela);
279     $Narudzba_jela .= $Narudzba_jela1;
280
281
282     $query = "INSERT INTO narudzba (Korisnik_id, Korisnicko_ime, Adresa, Detalji_narudzbe, Ukupna_cijena)
283     VALUES ('$Korisnik_id', '$Korisnicko_ime', '$Adresa', '$Detalji_narudzbe', '$Ukupna_cijena')";

```

Sl. 3.27. Prikaz PHP koda za narudžbu- 1.dio.

Slikom 3.28. prikazuje se drugi dio PHP koda za narudžbu gdje se provjerava da li je narudžba unesena u bazu podataka odnosno da li je zahtjev odrađen. Ako se košarica briše, te se ispisuje da je korisnik uspješno naručio. U slučaju da se zahtjev nije dobro uneso ispisuje se greška. Dodatno se nalaze pozivi na funkcije za statistiku, te SQL zahtjev koji se koristi za statistiku. Kada se napravila narudžba, onda administrator dohvaća tu narudžbu zajedno s svim ostalim narudžbama.

```

if($conn->query($query) === TRUE){
    unset($_SESSION["cart_item"]);
    statistikaKorisnikNarudzbaUnos($conn, $Korisnicko_ime);
    statistikaUkupnoPotrosnjaKorisnikUnos($conn, $Korisnicko_ime);
    statistikaUkupnoNarudzbiUnos($conn);
    statistikaUkupnoZaradaUnos($conn);
    $queryJelo2="UPDATE statistika SET Narudzba_jela='$Narudzba_jela'";
    $rezJelo = $conn->query($queryJelo2);
    statistikaNajJeloUnos($conn);
    echo "<div class = 'form'>
    <h3>Uspješno si naručio.</h3>";
}else{
    echo "Greška";
}

```

Sl. 3.28. Prikaz PHP koda za narudžbu- 2.dio.

Slikom 3.29. prikazuje se narudžbe sa svim podacima koji dohvaćeni iz baze podataka. Na kraju stranice se nalazi dio za straničenje koji prikazuje pet narudžbi po stranici. Izgled tablice s narudžbama se uklapa uz samu pozadinu.

Narudžbe	
ID:	115
Korisničko ime:	Admin
Adresa:	Adresa 1
Datum:	2020-07-10 20:51:57
Detalji narudžbe:	ID: 2 - Naziv: Cheeseburger - Kolicina: 1 - Cijena: 17 kn - Ukupna cijena jela:17 kn ID: 10 - Naziv: Mineralna voda - Kolicina: 1 - Cijena: 10 kn - Ukupna cijena jela:10 kn
Ukupna cijena:	27
ID:	114
Korisničko ime:	Admin
Adresa:	Adresa 2
Datum:	2020-07-10 20:50:02
Detalji narudžbe:	ID: 1 - Naziv: Hamburger - Kolicina: 1 - Cijena: 15 kn - Ukupna cijena jela:15 kn ID: 5 - Naziv: Pomfrit - Kolicina: 1 - Cijena: 10 kn - Ukupna cijena jela:10 kn
Ukupna cijena:	25

Sl. 3.29. Prikaz narudžbe.

Slikom 3.30. prikazuje se HTML i PHP koda za narudžbu. U kodu se nalazi i dio koda za straničenje, na način da se po svakoj stranici prikazuje pet narudžbi. Narudžbe se dohvaćaju preko SQL zahtjeva SELECT, te se unose u polje, a na kraju se ispisuju u tablice.

```

narudzba.php
69 <div class="txt-heading">Narudžbe</div>
70 <?php
71 require_once('spoj.php');
72 require_once('spojZaPrijavuReg.php');
73 $db_handle = new DBController();
74 if (!isset($_GET['page'])) {
75     $page = 1;
76 }else{
77     $page = $_GET['page'];
78 }
79 $num_per_page=5;
80 $start_from = ($page - 1)*5;
81 $narudzba = $db_handle->runQuery("SELECT * FROM `narudzba` ORDER BY `Datum` DESC LIMIT $start_from, $num_per_page");
82 if(mysqli_query($conn, $narudzba)) {
83     $narudzbaArray = array('ID'=>$narudzba[0]["ID"], 'Korisnicko_ime'=>$narudzba[0]["Korisnicko_ime"], 'Adresa'=>$narudzba[0]["Adresa"],
84 }else{
85     echo "";
86 }
87 if (!empty($narudzba)) {
88     foreach($narudzba as $key=>$value){
89
90 <div id="table-narudzba">
91 <p></p>
92 <table class="tbl-narudzba" style="width:100%" cellpadding="4" cellspacing="4">
93 <tr>
94 <th>ID:</th>
95 <td><?php echo $narudzba[$key]["ID"]; ?></td>
96 </tr>
97 <tr>
98 <th>Korisničko ime:</th>
99 <td><?php echo $narudzba[$key]["Korisnicko_ime"]; ?></td>
100 </tr>

```

Sl. 3.30. Prikaz HTML i PHP koda za dohvaćanje narudžbe s straničenjem 1.dio.

Slikom 3.31. prikazuje se dio koda za straničenje gdje se dohvaća narudžba preko SQL zahtjeva da bi se mogli izbrojati redovi, da bi na kraju mogao odrediti koliko će stranica biti. U ovisnosti na kojoj smo stranici prikazuju se tipke za određenu stranicu.

```

<?php
$page_query = "SELECT * FROM `narudzba`";
$page_result = mysqli_query($conn, $page_query);
$total_record = mysqli_num_rows($page_result);
$total_page=ceil($total_record/$num_per_page);

echo "<div id='pagging'>";
if($page>1){
    echo "<a href='narudzba.php?page=" . ($page-1) . "' class='btn btn-danger'>Nazad</a>";
}

for($i=1; $i<$total_page; $i++){
    echo "<a href='narudzba.php?page=" . $i . "' class='btn btn-primary'>$i</a>";
}

if($i>$page){
    echo "<a href='narudzba.php?page=" . ($page+1) . "' class='btn btn-danger'>Naprijed</a>";
}
echo "</div>";
?>

```

Sl. 3.31. Prikaz HTML i PHP koda za dohvaćanje narudžbe s straničenjem 2.dio.

3.2.5 Unos jela u jelovnik

Unos jela u jelovnik se jedino može omogućiti administratoru web stranice. U bazi podataka za korisnika nalazi se atribut uloga koji određuje tko je korisnik, a tko administrator. Svako ubačeno jelo se sprema u tablici jelovnik u bazi podataka. Slikom 3.32. prikazana je tablica jelovnika u bazi podataka gdje je ID predstavlja primarni ključ, te s povećava prilikom svakog unesenog jela. Naziv jela i cijena se prikazuju u jelovniku, dok se kategorija dohvaća i unosi po zadanoj kategoriji u jelovniku.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	ID	int(250)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	Naziv_jela	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/> 3	Kategorija	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/> 4	Cijena	int(250)			No	None			Change Drop More

Sl. 3.32. Prikaz tablice jelovnik u bazi podataka preko phpMyAdmin.

Slikom 3.33. prikazana je poveznica za dodavanje jela koja se prikazuje samo ako je prijavljen administrator. Provjera se izvodi preko *if* uvjeta, a ispis preko *echo* naredbe.


```

<?php
if(isset($_SESSION['Korisnicko_ime'])){
    if($_Uloga=="Administrator"){
        echo "<a class='btn btn-primary btn' id ='btn3' href='dodavanje_jela.php' role='button'>Dodaj jelo</a>";
    }
}
?>

```

Sl. 3.33. Prikaz PHP koda za omogućavanje administratoru ubacivanje jela.

Slikom 3.34. prikazuje se forma za ubacivanje jela. Kako bi se moglo pravilno upisati podatci mora se napraviti forma gdje će se administratoru omogućiti unos naziva jela, kategorije i cijene. Kod kategorije se mora omogućiti padajući izbornik s kategorijama koje se nalaze u bazi podataka.

Sl. 3.34. Prikaz forme za ubacivanje jela.

Slikom 3.35. prikazuje se odjeljak gdje se nalazi forma za ubacivanje jela. Svaki dio za upisivanje atributa za jelovnik ima svoj naziv. Kod kategorije nalazi se opcija za odabir pojedine kategorije. Kod cijene jedino se može upisati brojka, da ne bi došlo do pogreške ako se slovo unese.

```

<div id="hhh" class="container-fluid">
  <div class="form">
    <form name="jelovnik2" action="" method="post">
      <label for="Naziv_jela">Naziv jela: </label>
      <input type="text" name="Naziv_jela" placeholder="Naziv_jela" required/>
      <br>
      <label for="Kategorija">Kategorija: </label>
      <select id="Kategorija" name="Kategorija">
        <option value="Burgeri">Burgeri</option>
        <option value="Tortilje i kebabi">Tortilje i kebabi</option>
        <option value="Prilozi">Prilozi</option>
        <option value="Ostalo">Ostalo</option>
        <option value="Pica">Pića</option>
      </select>
      <br>
      <label for="Cijena">Cijena: </label>
      <input type="number" name="Cijena" value="">
      <br>
      <button type="submit" class="btn btn-block btn-primary" id="btnEmpty" name="action">Naruči</button>
    </form>
  </div>
</div>
<?php } ?>
</div>

```

Sl. 3.35. Prikaz HTML koda za ubacivanje jela.

Slikom 3.36. prikazuje se PHP kod za ubacivanje jela. Proces kod ubacivanja pojedinog jela je identičan kao registracija i narudžba. Provjerava se da li je unesen naziv jela, ako je onda se iz forme uzimaju podatci i unose se u varijable. Funkcija *stripslashes* izbacuje navodnike iz *stringa*, a funkcija *mysqli_real_escape_string* izbacuje specijalne znakove prilikom unosa u SQL i bazu podataka. Ubacuje se u SQL zahtjev INSERT INTO, te se šalje u bazu podataka. Ako je sve ispravno ubačeno, ispisuje se poruka.

```
<?php
require('spojZaPrijavuReg.php');

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

    $Naziv_jela = stripslashes($_REQUEST['Naziv_jela']);
    $Naziv_jela = mysqli_real_escape_string($conn, $Naziv_jela);

    $Kategorija = stripslashes($_REQUEST['Kategorija']);
    $Kategorija = mysqli_real_escape_string($conn, $Kategorija);

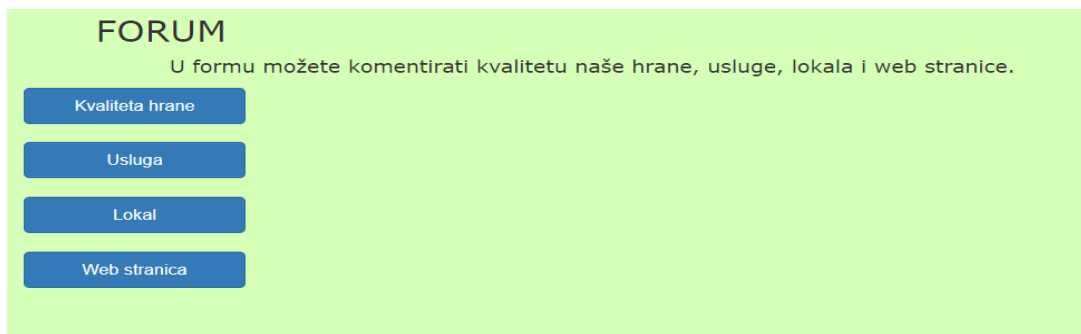
    $Cijena = stripslashes($_REQUEST['Cijena']);
    $Cijena = mysqli_real_escape_string($conn, $Cijena);

    $query = "INSERT into jelovnik (Naziv_jela, Kategorija, Cijena) VALUES ('$Naziv_jela', '$Kategorija', '$Cijena')";
    $result = mysqli_query($conn, $query);
    if($result){
        echo "<div class = 'form'>
        <h3>Uspješno si postavio jelo.</h3>
        </div>";
    }
}
}else{
?>
```

Sl. 3.36. Prikaz PHP koda za ubacivanje jela.

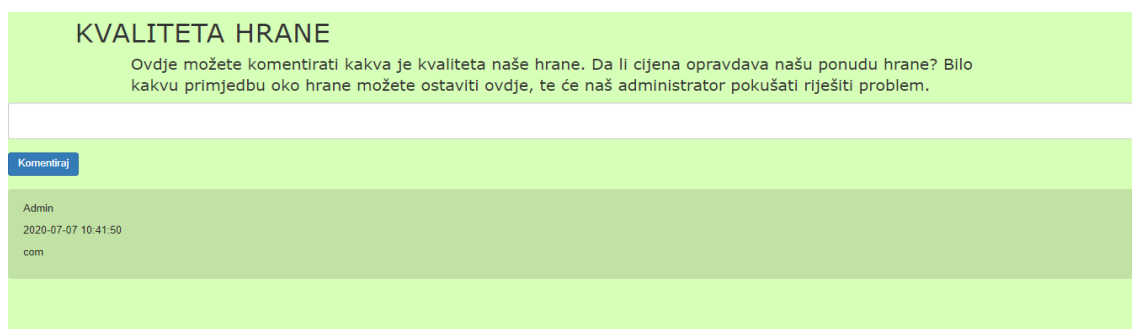
3.2.6 Komentari

Da bi korisnik imao mogućnost komentiranja, mora biti registriran i prijavljen. Komentiranje se vrši preko foruma gdje korisnik može komentirati kvalitetu hrane, uslugu lokal i web stranicu. Slikom 3.37. prikazane su tipke koje otvaraju poveznicu za jednu od tih sekcija.



Sl. 3.37. Prikaz foruma za odabir sekcija za komentiranje.

Slikom 3.38. prikazana je sekcija za komentiranje i jedan komentar. Komentar mogu vidjeti i neregistrirani korisnici. Sam komentar prikazuje korisničko ime osobe koja je komentirala, datum i vrijeme, te sami komentar. Da bi se komentar mogao unijeti i prikazati potrebna je baza podataka koja će u tablici komentara sadržavati sve komentare od svih korisnika za svaku sekciju.



Sl. 3.38. Prikaz sekcije za komentiranje s komentarom.

Slikom 3.39. prikazana je tablica komentara u bazi podataka gdje je ID primarni ključ koji se povećava svakim unesenim komentarom, korisničko ime, datum i komentar se prikazuju u sekciji za komentiranje, dok ID stranice služi da bi se svaki komentar prikazao točno na stranici za komentiranje kojoj pripada. Poslije baze podataka mora se napraviti funkcije za upis komentara, te za ispisivanje komentara.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	ID	int(250)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	Korisnicko_ime	varchar(250)	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/> 3	Datum	datetime			No	current_timestamp()			Change Drop More
<input type="checkbox"/> 4	Komentar	text	latin2_croatian_ci		No	None			Change Drop More
<input type="checkbox"/> 5	ID_stranice	int(250)			No	None			Change Drop More

Sl. 3.39. Prikaz tablice komentari u bazi podataka preko phpMyAdmin.

Prema slici 3.40. prikazana je PHP funkcija za unos komentara prema [12]. Funkcija mora primiti parametar za povezivanje s bazom podataka. Ako je komentar upisan u dijelu za upisivanje komentara onda u varijable unosi već upisane vrijednosti. Preko SQL zahtjeva INSERT INTO se vrijednosti upisuju u tablici za komentare u bazi podataka.

```
function unesiKomentar($conn){
    if(isset($_POST['komentarSubmit'])){
        $Korisnicko_ime = $_POST['Korisnicko_ime'];
        $Komentar = $_POST['Komentar'];
        $id_stranice = $_POST['id_stranice'];

        $sql = "INSERT INTO komentari (Korisnicko_ime, Komentar, ID_stranice)
VALUES ('$Korisnicko_ime', '$Komentar', '$id_stranice')";
        $resultat = $conn->query($sql);
    }
}
```

Sl. 3.40. Prikaz PHP funkcije za unos komentara.

Prema slici 3.41. prikazana je PHP funkcija za dohvaćanje komentara odnosno ispis komentara. Funkcija prima parametar za spoj, te parametar ID stranice. Iz SQL zahtjeva se prema ID stranice dohvaćaju komentari, te se unose u rezultat. U *if* uvjetu se provjerava da li se izvršio zahtjev, ako se u *while* petlji dohvatio rezultat iz baze, onda će se dohvatiti redovi iz tablice komentari u bazi podataka, te će se ispisati preko *echo* naredbe.

```
function dohvatiKomentar($conn, $id_stranice){
    //$id_stranice = $_POST['id_stranice'];
    $sql = "SELECT * FROM komentari WHERE ID_stranice='$id_stranice'";
    $resultat = $conn->query($sql);
    $error="";
    if(mysqli_query($conn, $sql)) {
        while($row = $resultat->fetch_assoc()){
            echo "<div class='komentar-box'><p>";
            echo $row['Korisnicko_ime']."<br><br>";
            echo $row['Datum']."<br><br>";
            echo nl2br($row['Komentar']);
            echo "</div>";
        }
    } else {
        $error = "Error: " . $sql . "<br>" . mysqli_error($conn);
        echo $error;
    }
}
```

Sl. 3.41. Prikaz PHP funkcije za dohvaćanje komentara.

Slikom 3.42. prikazan je dio koda gdje se prikazuju komentari i gdje se komentira. U PHP kodu se provjerava da li je korisnik prijavljen, ako je onda se omogućuje komentiranje. Vidljivo je kako se u *echo* naredbi nalazi HTML kod koji se prikazuje, te se unutar njega poziva funkcija *unesiKomentar* za unos komentara. Nakon toga se poziva funkcija *dohvatiKomentar* za dohvaćanje odnosno ispis komentara.

```
<div id="hhh" class="container-fluid">
<h1 id="ново">KVALITETA HRANE</h1>
<section>
  <p id="tekst1">Ovdje možete komentirati kakva je kvaliteta naše hrane.
  Da li cijena opravdava našu ponudu hrane? Bilo kakvu primjedbu oko hrane možete ostaviti ovdje,
  te će naš administrator pokušati riješiti problem.</p>
</section>
</div>
<?php
if(isset($_SESSION['Korisnicko_ime'])){
  echo"<form method='POST' action='unesiKomentar($conn).">
    <input type='hidden' name='Korisnicko_ime' value='".$_SESSION['Korisnicko_ime']."'>
    <input type='hidden' name='id_stranice' value='1'>
    <textarea id='komentar' class='form-control' name='komentar' ></textarea><br>
    <button id='btnKomentar' class='btn btn-primary' type='submit' name='komentarSubmit'>Komentiraj</button><br><br>
  </form>";
}else{
  echo "Da bi komentirao moraš se prijaviti!";
}
dohvatiKomentar($conn, $id_stranice);
?>
</div>
```

Sl. 3.42. Prikaz HTML i PHP funkcije za prikaz komentara.

3.2.7 Statistika

Statistika je potrebna da bi administrator stranice mogao vidjeti bitne podatke o svom restoranu odnosno web stranici restorana. Isto tako korisnik ima mogućnost vidjeti statistiku koja se veže za njega. Slikom 3.43 prikazana je statistika za administratora gdje administrator može vidjeti koliko ima ukupno narudžbi i korisnika, kolika je ukupna zarada, ukupni broj pregleda web stranice, najviše naručivano jelo, te koji korisnik ima najviše narudžbi i najveću potrošnju. Isto tako administrator ima uvid u potrošnju i količinu narudžbi kod svakog korisnika.

Profil i statistika restorana

Pozdrav: Admin

ID:	1
Ukupno narudžbi:	30
Ukupno korisnika:	1
Zbroj ukupne zarade:	703
Ukupni broj pogleda web stranice:	5
Najviše naručivano jelo:	Hamburger

Korisnik s najviše narudžbi

Korisničko ime:	robi007
Ukupno narudžbi:	16

Korisnik s najvećom potrošnjom

Korisničko ime:	robi007
Ukupna potrošnja:	3415

[Statistika za sve korisnike](#)

Sl. 3.43. *Prikaz statistike za administratora.*

Slikom 3.44. prikazan je dio PHP koda koji po ulozi za administratora koji se dohvaća pomoću sesije prikazuje i izvodi naredni HTML i PHP kod.

```

profil.php
1  <?php
2  session_start();
3  error_reporting(E_ERROR | E_PARSE);
4  include 'spojZaPrijavuReg.php';
5  include 'spoj.php';
6  if(isset($_SESSION['Korisnicko_ime'])){
7      $Korisnicko_ime=$_SESSION['Korisnicko_ime'];
8      $Korisnik_id=$_SESSION['Korisnik_id'];
9      $Uloga=$_SESSION['Uloga'];
10 }
11 if($Uloga=="Administrator")
12 {
13  ?>

```

Sl. 3.44. *Prikaz HTML i PHP koda za prikaz statistike 1. dio.*

Slikom 3.45. prikazan je dio HTML i PHP koda koji se radi preko objektno-orijentiranog načina. Dohvaća se ukupna statistika iz baze podataka i ubacuje se u polje.

```

<div id="hhh" class="container-fluid">
  <h1 id="novo">Profil i statistika restorana</h1>
  <?php
  if(isset($_SESSION['Korisnicko_ime'])){
    echo "Pozdrav: ".$_SESSION['Korisnicko_ime'];
  }
  ?>
  <?php
  require_once('spoj.php');
  $db_handle = new DBController();
  $statistikaAdmin = $db_handle->runQuery("SELECT * FROM `statistika`");
  if(mysqli_query($conn, $statistikaAdmin)) {
    $statistikaArray = array('ID'=>$statistikaAdmin[0]['ID'], 'Ukupno_narudzba'=>$statistikaAdmin[0]['Ukupno_narudzba'],
    'Ukupno_korisnika'=>$statistikaAdmin[0]['Ukupno_korisnika'], 'Zbroj_zarade'=>$statistikaAdmin[0]['Zbroj_zarade'],
    'Ukupno_pogleda_web'=>$statistikaAdmin[0]['Ukupno_pogleda_web'],
    'Najvise_narucivano_jelo'=>$statistikaAdmin[0]['Najvise_narucivano_jelo']);
  }
  if (!empty($statistikaAdmin)) {
    foreach($statistikaAdmin as $key=>$value){
  ?>

```

Sl. 3.45. Prikaz HTML i PHP koda za prikaz statistike 2. dio.

Prema slici 3.46 u nastavku koda dohvaćena statistika se ubacuje u tablicu po zadanom ključu i vrijednosti. Na identičan način se odrađuje ostatak HTML i PHP koda. U statistici za korisnika, korisnik može vidjeti svoje podatke, koliko ima ukupno narudžbi i koliko ima ukupnu potrošnju. Za statistiku je potrebna baza podataka gdje će se unositi potrebni podatci koji će se izračunati pomoću funkcija. Tablica za korisnika u bazi podataka sadrži dva atributa statistiku i potrošnju koji će nam biti potrebni za odrađivanje statistike. Statistika za korisnika je odrađena na identičan način kao i statistika za administratora.

```

<div id="table-narudzba">
<p></p>
<table class="tbl-narudzba" style="width:100%" cellpadding="4" cellspacing="4">
  <tr>
    <th>ID:</th>
    <td><?php echo $statistikaAdmin[$key]["ID"]; ?></td>
  </tr>
  <tr>
    <th>Ukupno narudžbi:</th>
    <td><?php echo $statistikaAdmin[$key]["Ukupno_narudzba"]; ?></td>
  </tr>
  <tr>
    <th>Ukupno korisnika:</th>
    <td><?php echo $statistikaAdmin[$key]["Ukupno_korisnika"]; ?></td>
  </tr>
  <tr>
    <th>Zbroj ukupne zarade:</th>
    <td><?php echo $statistikaAdmin[$key]["Zbroj_zarade"]; ?></td>
  </tr>
  <tr>
    <th>Ukupni broj pogleda web stranice:</th>
    <td><?php echo $statistikaAdmin[$key]["Ukupno_pogleda_web"]; ?></td>
  </tr>
  <tr>
    <th>Najviše naručivano jelo:</th>
    <td><?php echo $statistikaAdmin[$key]["Najvise_narucivano_jelo"]; ?></td>
  </tr>
</table>
<p><br></p>
</div>
<?php
}

```

Sl. 3.46. Prikaz HTML i PHP koda za prikaz statistike 3. dio.

Slikom 3.47. prikazan je tablica za statistiku, a od toga se mora izdvojiti atribut *Narudzba_jela* za sva jela koja su naručena, te se iz tog atributa u funkciji izdvaja najviše naručeno jelo.. Nakon baze podataka potrebno je bilo napraviti funkcije koje će obrađivati potrebne podatke i unositi ih u statistiku.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 ID	int(250)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 Ukupno_narudzba	int(250)			No	0			Change Drop More
<input type="checkbox"/>	3 Ukupno_korisnika	int(250)			No	0			Change Drop More
<input type="checkbox"/>	4 Zbroj_zarade	int(250)			No	0			Change Drop More
<input type="checkbox"/>	5 Ukupno_pogleda_web	int(250)			No	0			Change Drop More
<input type="checkbox"/>	6 Narudzba_jela	text	latin2_croatian_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	7 Najvise_narucivano_jelo	varchar(250)	latin2_croatian_ci		Yes	NULL			Change Drop More

Sl. 3.47 Prikaz tablice statistika u bazi podataka preko phpMyAdmin.

Slikom 3.48. prikazuju se dvije PHP funkcije koje odrađuju identičan posao. Jedan služi za izračun ukupnog broja korisnika, a druga za izračun ukupnog broja narudžbi. U bazi podataka atribut *Ukupno_korisnika* i atribut *Ukupno_narudzba* su postavljeni na nulu. Preko SQL zahtjeva se dohvaća potreban atribut. Provjerava se zahtjev te se u petlji *while* dohvaća potreban red iz baze podataka i ubacuje u varijablu. Potrebna varijabla se povećava za jedan, te se preko SQL zahtjeva i SQL naredbe UPDATE u bazi podataka izmjenjuje potrebna vrijednost.

```
statistika.php
1  <?php
2  function statistikaUkupnoKorisnikaUnos($conn){
3      $sql3 = "SELECT Ukupno_korisnika FROM statistika";
4      $rezultat = $conn->query($sql3);
5      if(mysqli_query($conn, $sql3)) {
6          while($row = $rezultat->fetch_assoc()){
7              $Ukupno_korisnika = $row['Ukupno_korisnika'];
8          }
9      }
10     $Ukupno_korisnika = $Ukupno_korisnika + 1;
11     $sql4 = "UPDATE statistika SET Ukupno_korisnika='$Ukupno_korisnika'";
12     $rez = $conn->query($sql4);
13 }
14
15 function statistikaUkupnoNarudzbiUnos($conn){
16     $sql5 = "SELECT Ukupno_narudzba FROM statistika";
17     $rezultat = $conn->query($sql5);
18     if(mysqli_query($conn, $sql5)) {
19         while($row = $rezultat->fetch_assoc()){
20             $Ukupno_narudzba = $row['Ukupno_narudzba'];
21         }
22     }
23     $Ukupno_narudzba = $Ukupno_narudzba + 1;
24     $sql6 = "UPDATE statistika SET Ukupno_narudzba='$Ukupno_narudzba'";
25     $rez = $conn->query($sql6);
26 }
```

Sl. 3.48. Prikaz PHP funkcija za određivanje ukupnog broja korisnika i narudžbe.

Prema slici 3.49. prikazane su dvije PHP funkcije. Prva funkcija prikazuje izračun ukupne zarade, te se koristio objektno-orijentirani način. Varijabla za ukupni zbroj se postavila na nulu zbog računanja. Preko SQL zahtjeva SELECT dohvaćaju se ukupne cijene iz tablice za narudžbu. U polje se unosi svaka od tih vrijednosti, te se preko *foreach* petlje izračunao ukupan zbroj. Na kraju se preko SQL zahtjeva UPDATE unosi u tablicu statistika i bazu podataka, odnosno izmjenjuju se vrijednosti. Funkcija za ukupan broj posjeta na web stranici radi na identičan način kao i kod određivanja ukupnog broja narudžbi i korisnika, te se funkcija poziva na početnoj web stranici.

```

function statistikaUkupnoZaradaUnos($conn){
    $db_handle = new DBController();
    $ukupni_zbroj=0;
    $zbroj = $db_handle->runQuery("SELECT `Ukupna_cijena` FROM `narudzba`");
    if($zbroj==TRUE) {
        $zbrojArray = array('Ukupna_cijena'=>$zbroj[0]["Ukupna_cijena"]);
    }
    if (!empty($zbroj)) {
        foreach($zbroj as $key=>$value){
            $ukupni_zbroj += $zbroj[$key]["Ukupna_cijena"];
        }
    }
    $sql7 = "UPDATE statistika SET Zbroj_zarade='$ukupni_zbroj'";
    $rez = $conn->query($sql7);
}

function statistikaUkupnoUkupnoPosjetaWebStraniciiUnos($conn){
    $sql8 = "SELECT Ukupno_pogleda_web FROM statistika";
    $rezultat = $conn->query($sql8);
    if(mysqli_query($conn, $sql8)) {
        while($row = $rezultat->fetch_assoc()){
            $Ukupno_pogleda_web = $row['Ukupno_pogleda_web'];
        }
    }
    $Ukupno_pogleda_web += 1;
    $sql9 = "UPDATE statistika SET Ukupno_pogleda_web='$Ukupno_pogleda_web'";
    $rez = $conn->query($sql9);
}
?>

```

Sl. 3.49. Prikaz PHP funkcija za određivanje ukupne zarade i ukupnog broja posjeta web stranice.

Slikom 3.50. prikazuje se funkcija za ukupnu potrošnju korisnika koja prima parametar korisničko ime. Funkcija implementira objektno-orijentirani način. Varijabla za ukupni zbroj se postavlja na nulu, te se nakon toga odradio SQL zahtjev SELECT gdje se dohvaća ukupna cijena iz tablice za narudžbu preko korisničkog imena. Ukupna cijena se unosi u polje, a nakon toga se ukupna cijena zbraja. Poslije toga se iz SQL zahtjeva dohvaća potrošnja od korisnika, te se na tu potrošnju zbrojio već ukupni zbroj od narudžbe i preko SQL zahtjeva UPDATE se uneslo u tablicu.

```

function statistikaUkupnoPotrosnjaKorisnikUnos($conn, $Korisnicko_ime){
    $db_handle = new DBController();
    $ukupni_zbroj=0;
    $zbroj = $db_handle->runQuery(["SELECT `Ukupna_cijena` FROM `narudzba`
    WHERE Korisnicko_ime='$Korisnicko_ime'"]);
    if($zbroj==TRUE) {
        $zbrojArray = array('Ukupna_cijena'=>$zbroj[0]["Ukupna_cijena"]);
    }else{
        echo "Ne postoji narudžba";
    }
    if (!empty($zbroj)) {
        foreach($zbroj as $key=>$value){
            $ukupni_zbroj += $zbroj[$key]["Ukupna_cijena"];
        }
    }
    $sql3 = "SELECT Potrosnja FROM korisnik WHERE Korisnicko_ime='$Korisnicko_ime'";
    $rezultat = $conn->query($sql3);
    if(mysqli_query($conn, $sql3)) {
        while($row = $rezultat->fetch_assoc()){
            $Potrosnja = $row['Potrosnja'];
        }
    }
    $Potrosnja = $Potrosnja + $ukupni_zbroj;
    $sql4 = "UPDATE korisnik SET Potrosnja='$Potrosnja' WHERE Korisnicko_ime='$Korisnicko_ime'";
    $rez = $conn->query($sql4);
}

```

Sl. 3.50. Prikaz PHP funkcije za ukupnu potrošnju korisnika.

Slikom 3.51. prikazuje se PHP funkcija za ukupnu narudžbu od korisnika koja je odrađena na identičan način kao neke prethodno prikazane funkcije i funkcija za najviše naručivano jelo. Funkcija za najviše naručivano jelo radi na način da se preko jelovnika u statistiku tokom narudžbe unesu sva jela koja su se naručila. Ta jela se dohvaćaju preko SQL zahtjeva SELECT, te se u varijablu unosi taj red *Potrosnja* iz tablice. Pošto su jela bila odvojena zarezom s kojim se odvojio pri unosu u bazu podataka, svako to jelo se moralo posebno odvojiti preko funkcije *explode*. Funkcija *explode* pretvara *string* u polje s vrijednostima iz *stringa*, gdje se svaka vrijednost odvaja određenim znakom. Funkcija *array_count_values* broji učestalost ponavljanja pojedinog znaka, a *arsort* sortira polje od najfrekventnijega do najmanje frekventnoga jela. Funkcija *array_slice* zajedno s *array_keys* odvaja prvo jelo po ključu iz polja, a funkcija *implode* pretvara polje u *string*. Na kraju se preko SQL zahtjeva UPDATE unosi u tablicu za statistiku u bazi podataka.

```

function statistikaKorisnikNarudzbaUnos($conn, $Korisnicko_ime){
    $sql1 = "SELECT Statistika FROM korisnik WHERE Korisnicko_ime='$Korisnicko_ime'";
    $rezultat = $conn->query($sql1);
    if(mysqli_query($conn, $sql1)) {
        while($row = $rezultat->fetch_assoc()){
            $Statistika = $row['Statistika'];
        }
    }
    $Statistika = $Statistika + 1;
    $sql2 = "UPDATE korisnik SET Statistika='$Statistika' WHERE Korisnicko_ime='$Korisnicko_ime'";
    $rez = $conn->query($sql2);
}

function statistikaNajJeloUnos($conn){
    $sql5 = "SELECT Narudzba_jela FROM statistika";
    $rezultat = $conn->query($sql5);
    if(mysqli_query($conn, $sql5)) {
        while($row = $rezultat->fetch_assoc()){
            $Narudzba_jela = $row['Narudzba_jela'];
        }
    }
    $Narudzba_jela_dijelovi = explode(",", $Narudzba_jela);
    $Najfrekventnija_jela = array_count_values($Narudzba_jela_dijelovi);
    arsort($Najfrekventnija_jela);
    $Naj_jelo = array_slice(array_keys($Najfrekventnija_jela), 0, true);
    $Najvise_narucivano_jelo = implode( $Naj_jelo );
    $sql6 = "UPDATE statistika SET Najvise_narucivano_jelo='$Najvise_narucivano_jelo'";
    $rez = $conn->query($sql6);
}

```

Sl. 3.51. *Prikaz PHP funkcija za ukupno narudžbi od korisnika i najviše naručivano jelo.*

4. UPUTE ZA KORIŠTENJE

Kada korisnik uđe u web aplikaciju za restoran brze hrane ima mogućnost uvida u jelovnik i osnovnim informacijama o web stranici. Da bi naručio hranu i imao mogućnost komentiranja na forumu, korisnik mora biti registriran i prijavljen. Prema slici 3.1. korisnik u izborniku može odabrati između početne stranice, jelovnika, prijave, registracije, foruma, a kada se registrira, odnosno prijavi može izabrati narudžbu i u desnom dijelu naslovne trake posjetiti svoje profil, te se odjaviti.

4.1 Registracija

Prema slici 4.1. prikazan je izgled registracije gdje korisnik može unijeti podatke koji se traže, te se registrirati pomoću tipke za registraciju.



Registracija

Ime :

Prezime :

Korisničko ime :

E-mail :

Lozinka :

Sl. 4.1. Prikaz izgleda registracije.

Registracija je vrlo jednostavna, potrebno je ostaviti osobne podatke (ime, prezime, korisničko ime, email i lozinka), a nakon toga se potrebno prijaviti uz pomoć korisničkog imena i lozinke koja je zaštićena. Nakon toga se otvaraju ostale mogućnosti za korištenje web aplikacije.

4.2 Prijava

Kada se korisnik registrirao, onda se mora i prijaviti da bi mogao naručivati hranu, te komentirati na forumu. Prema slici 3.7. korisnik se prijavljuje na sličan način kao što se i registrira. U prvo polje mora unijeti korisničko ime, a u drugo polje lozinku, te nakon toga se prijaviti pomoću tipke za prijavu. Kada je korisnik prijavljen pojavljuje mu se poveznica za sve narudžbe i u desnom kutu naslovne trake svoje korisničko ime, profil i tipku za odjavu.

4.3 Naručivanje jela

Prema slici 3.9. se prikazuje se jelovnik gdje korisnik može odabrati pojedinu kategoriju i pojedino jelo unutar pojedine kategorije preko tipke za naručivanje, te se to jelo pojavljuje unutar košarice (engl. *shopping cart*) koja je prikazana na slici 3.21., te se to jelo može ukloniti iz košarice preko tipke za uklanjanje. Isto tako moguće je izbaciti sva jela iz košarice pomoću tipke za izbacivanje svih jela. U košarici se osim jela prikazuju i cijene pojedinih jela zajedno s količinom koja je odabrana, te ukupna cijena narudžbe. Nakon što su jela za naručivanje odabrana, potrebno je ostaviti adresu gdje će narudžba stići, te onda naručiti preko tipke za konačnu narudžbu. Prema slici 3.29. administrator može vidjeti svaku narudžbu koju je korisnik naručio. Isto tako i korisnik može vidjeti samo svoje narudžbe. Prema slici 4.2. prikazuje se straničenje kojim se korisnik i administrator može kretati da vidi sve narudžbe koje su naručene.

ID:	111
Korisničko ime:	Admin
Adresa:	Adresa 1
Datum:	2020-07-09 21:49:52
Detalji narudžbe:	ID: 1 - Naziv: Hamburger - Kolicina: 1 - Cijena: 15 kn - Ukupna cijena jela: 15 kn ID: 5 - Naziv: Pomfrit - Kolicina: 1 - Cijena: 10 kn - Ukupna cijena jela: 10 kn
Ukupna cijena:	25

1 2 3 4 Naprijed

Sl. 4.2. Prikaz izgleda straničenja.

Na slici 3.34. prikazan je izgled ubacivanja jela u jelovnik od strane administratora. Za razliku od korisnika administrator ima mogućnost dodavanja jela u jelovnik, gdje korisnik odabire ime jela, kategoriju jela, te cijenu jela, te unosi jelo u jelovnik.

4.4 Forum i komentari

Korisniku se otvara mogućnost komentiranja na forumu, gdje može komentirati kvalitetu hrane, uslugu, lokal i web stranicu. Na slici 3.37. prikazan je izbornik za forum gdje korisnik može odabrati šta želi komentirati od prethodno spomenutih opcija. Na slici 3.38 prikazuju se komentari i dio za komentiranje za izabranu opciju. Korisnik komentar može ostaviti u okviru za tekst, te odabrati tipku za dodavanje komentara. Taj komentar se prikazuje ispod okvira za tekst zajedno sa svim ostalim komentarima drugih korisnika.

4.5 Statistika

Slika 3.43. prikazuje statistiku za administratora. Administrator u statistici može vidjeti statistiku svih korisnik i statistiku restorana. U statistiku svih korisnika može ući preko poveznice. Prema slici 4.2. prikazana je statistika za korisnika gdje korisnik može vidjeti svoje podatke i svoju statistiku.



Profil i statistika korisnika	
Pozdrav: robi007	
Podatci korisnika	
ID:	10
Ime:	Robi
Prezime:	Horvat
Korisničko ime:	robi007
Email:	robi007@gmail.com
Uloga:	Korisnik
Statistika korisnika	
Ukupno narudžbi:	14
Ukupna potrošnja:	1544

Sl. 4.2. Prikaz statistike za korisnika.

5. ZAKLJUČAK

Web aplikacija za restoran brze hrane služi za korisnika koji na jednostavan način želi naručiti neko jelo iz restorana na način da mu je sve vidljivo i transparentno. Isto tako predstavlja web aplikaciju kod koje administrator može vrlo jednostavno koristiti, te imati uvid u narudžbe korisnika, statistiku restorana i mogućnost ubacivanja jela u jelovnik. Aplikacija je transparentna kako za korisnika tako i za administratora, zbog svoje jednostavnosti i mogućnosti uvida u statistiku i narudžbe. U procesu izrade same web aplikacije u HTML, CSS i PHP programskim jezicima uvidjele su se neke dobre i neke loše strane takvog programiranja, te sama povezanost između baze podataka i same web aplikacije. Dobre strane takvog programiranja su da je jednostavno i lako učenje takvog programiranja uz prethodno znanje nekih programski jezika i baze podataka. Loša strana takvog programiranja je nedostatak otklanjanja pogrešaka. Neke pogreške će se prikazati prilikom pokretanja web preglednika odnosno aplikacije, dok neke pogreške se uopće neće prikazati, te je samim time takvo programiranje otežano.

LITERATURA

- [1] Grupa autora, HTML[online] [21.08.2020] , dostupno na:
https://www.w3schools.com/html/html_intro.asp
- [2] Grupa autora, CSS[online] [21.08.2020] , dostupno na:
https://www.w3schools.com/css/css_intro.asp
- [3] Grupa autora, JavaScript[online] [21.08.2020] , dostupno na:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript
- [4] Grupa autora, PHP[online] [21.08.2020] , dostupno na: <https://www.php.net/manual/en/intro-what-is.php>
- [5] Grupa autora, XAMPP[online] [21.08.2020] , dostupno na:
<https://www.apachefriends.org/index.html>
- [6] Grupa autora, SQL[online] [21.08.2020] , dostupno na:
https://www.w3schools.com/sql/sql_intro.asp
- [7] Grupa autora, phpMyAdmin[online] [21.08.2020] , dostupno na:
<https://code.visualstudio.com/docs>
- [8] Grupa autora, VISUAL STUDIO CODE[online] [21.08.2020] , dostupno na:
<https://code.visualstudio.com/docs>
- [9] Grupa autora, Prijava i registracija[online][20.01.2020] , dostupno na:
<https://www.allphptricks.com/simple-user-registration-login-script-in-php-and-mysql/>
- [10] Grupa autora, Jelovnik i naručivanje[online][20.02.2020] , dostupno na:
<https://phppot.com/php/simple-php-shopping-cart/>
- [11] Grupa autora, Naručivanje[online][10.03.2020] , dostupno na: <https://projectworlds.in/free-projects/php-projects/food-ordering-system-project-in-php/>
- [12] Grupa autora, Komentari[online][15.04.2020] , dostupno na:
<https://www.youtube.com/watch?v=kWOuUkLtQZw>
- [13] Grupa autora, Straničenje[online][15.04.2020] , dostupno na:
<https://www.youtube.com/watch?v=qxNXS5PeID4&feature=youtu.be>

SAŽETAK

Naslov: Web aplikacija za restoran brze hrane

Na području Hrvatske mali je broj postojećih aplikacija za naručivanje brze hrane. Ova web aplikacija će pratiti današnje standarde za izračun cijena, naručivanje pojedinih menija, te ispunjavanja podataka korisnika kako bi se određena ponuda iz menija mogla naručiti. U izradi web aplikacije korištene su sljedeće tehnologije: HTML, CSS, JavaScript, MySQL, PHP, te programski alati poput Visual Studio Code, XAMPP i phpMyAdmin. Aplikacija nudi korisniku izbor ponuda iz jelovnika, količinu narudžbe koja se naručuje iz pojedinog jelovnika, komentar, ocjena ponude i usluge, ukupna cijena narudžbe, te se cijela narudžba sprema u bazu podataka.

Ključne riječi: brza hrana, meni, naručivanje, ponuda, web aplikacija

ABSTRACT

Title: Web application for fast food restaurant

In Croatia, there are a small number of existing applications for ordering fast food. This web application will follow today's standards for calculating the price, ordering individual menus and filling in user data to determine the offer from the menu that could be ordered. The following technologies were used in the development of web applications: HTML, CSS, JavaScript, MySQL, PHP, and software tools such as Visual Studio Code, XAMPP and phpMyAdmin. The application offers a user selection offer from the menu, the amount of orders ordered from each menu, comments, ratings and services, the total price of orders, and saves the entire order in the database.

Keywords: fast food, menu, ordering, offer, web application

ŽIVOTOPIS

Robert Horvat rođen je 10.listopada 1997.godine u Zagrebu. Završio je Osnovnu Školu Julija Kempfa u Požegi. Nakon toga upisuje Tehničku školu u Požegi, smjer tehničar za elektroenergetiku, završava praksu u HEP-u, te maturira 2016. godine. Nakon mature upisuje stručni studije elektrotehnike smjer informatika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Poznaje rad sa programskim jezicima: C, Java, Python, SQL, JavaScript, HTML, CSS, PHP. Također upoznat i sa raznim alatima korištenim pri programiranju poput programa za kodiranje Visual studio, Visual studio code, Android studio, IntelliJ IDEA, XAMPP, GitHub.

PRILOZI

Projektna mapa s izvornim kodom i priloženim word i pdf dokumentom nalazi se na optičkom disku koji je priložen uz printanu verziju završnog rada.