

# **Maturski rad**

iz

Web programiranja

Tema: Izrada dinamičke web aplikacije – Restoran

Mentor:

Prof. Jasna Kopic

Učenik:

Faris Lindov IV-5

Sarajevo, maj 2022. godine

Datum predaje maturskog rada: \_\_\_\_\_

Komentar mentora o radu:

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Ocjena maturskog rada: \_\_\_\_\_ (    )

Datum odbrane maturskog rada: \_\_\_\_\_

Pitanja na odbrani maturskog rada:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

Ocjena iz usmene odbrane rada: \_\_\_\_\_ (    )

Komisija u sastavu:

1. \_\_\_\_\_, predsjednik
2. \_\_\_\_\_, ispitivač
3. \_\_\_\_\_, stalni član

Potpis članova komisije:

---

---

---

## Sadržaj:

1. Uvod .....	3
2. Dinamička Web aplikacija – Restoran .....	4
2.1. HTML(hypertext markup language) .....	4
2.1.1 Klasa i id .....	5
2.1.2 Ostali tagovi .....	5
2.1.3 Divajderi .....	6
2.2 CSS .....	6
2.2.1 CSS selektori .....	7
2.2.1.1 Pseudoklase .....	8
2.3 JavaScript .....	9
2.4 PHP .....	10
2.4 Izrada i rješavanje problema .....	12
2.4.1 Početna stranica .....	12
2.4.2 Meni .....	13
2.4.3 Kuhari .....	15
2.4.4 Lokacija i o nama stranica .....	16
2.4.6 Komentari .....	16
2.4.7 Kontakt .....	18
3. Zaključak .....	19
4. Literatura .....	20
7. Popis slika .....	21

# 1.Uvod

---

Odabrana tema ovog maturskog rada je „Izrada dinamičke web aplikacije - Restoran“.

Svakodnevno se susrećemo sa web sajtovima. Na prvi izgled oni izgledaju one izgledaju jednostavnu, ali zapravo se u pozadini odvija mnogo funkcija koje korisnik ne može vidjeti. Stranice koje su spojene sa bazom podataka se nazivaju dinamičke.

Dinamičke web aplikacije su stranice koje imaju sposobnost da se mijenjaju same na osnovu podataka i upita koje primaju od korisnika koji gleda web sajt. Koriste se najčešće prilikom izrade zahtjevnih web stranica koji moraju, apsolutno, da budu prilagođeni zahtjevima korisnika koji pregleda web sajt. Također se koriste za povezivanje s bazom podataka i njenom manipulacijom. Upotrebom ove tehnike je moguće ostvariti mnogo različitih prikaza na web sajtu kreiranjem i programiranjem samo jedne dinamičke strane.

U ovom maturskom radu bit će primijenjeno nekoliko tehnika kao što su: HTML, CSS, PHP, JavaScript, MySQL. Cilj maturskog rada je upoznavanje sa kompleksnim kreiranjem web aplikacija i spajanje više programskih jezika u jedan projekat.

U ovom radu će biti objašnjeni svi korišteni programski jezici i tehnike. U drugom dijelu će biti objašnjene funkcionalnosti na stranici i rješavanje problema koji su vezani za izradu dinamičke web aplikacije.

Ova tema je izabrana, jer je spoj više programskih tehnika. Baza podataka je prethodno projektirana, dizajnirana i kreirana. U njoj se unose podatci koji će se prikazivati na web stranicama. Cilj je bio prilagoditi i spojiti bazu podataka sa web aplikacijom. Izrada maturskog rada sadrži više poglavlja koji su razdijeljeni u pod poglavlja od kojih su glavni korišteni programski jezici i njihove funkcionalnosti i sama izrada web aplikacije uz objašnjene funkcije i probleme.

## 2. Dinamička Web aplikacija – Restoran

---

Statičke web stranice su one koje su fiksne i prikazuju isti sadržaj za svakog korisnika, obično napisan isključivo u HTML-u. S druge strane, dinamična web stranica je ona koja može prikazati različite sadržaje i omogućiti interakciju korisnika, koristeći napredno programiranje i baze podataka uz HTML. Dinamičke stranice su ugrađeni programi koji omogućavaju automatske promjene nekih podataka na stranicama bez intervencije autora stranice. Kreiranje dinamičke web aplikacije, pored HTML-a i CSS-a, potreban je PHP, JavaScript, JQuery, AJAX... Kombinacijom ovih programskih jezika možemo napraviti dinamičku web stranicu koja je interaktivna. Korisnik ima mogućnost da unosi podatke, bira koje podatke da website prikazuje. Dinamička web stranica je stranica koja je spojena sa bazom podataka za taj website. Svaki put kada se podatak u bazi podataka promjeni, promjeni se i na web stranici.

Kako radi web aplikacija:

1. Klijent (korisnik) šalje zahtjev web serveru, putem interneta u web pregledniku
2. Web server prosljeđuje zahtjev aplikaciji na serveru
3. Aplikacija izvodi zadatak, generira rezultat i šalje ga web serveru
4. Web server prosljeđuje rezultat klijentu (korisniku) u web preglednik

„HTTP je u upotrebi od 1990. godine, HTML, kao standard, od 1996. Uvođenje JavaScripta (tada poznatog kao LiveScript) omogućilo je proizvodnju dinamičkih web stranica na strani klijenta, sa JavaScript kodom koji se izvršava u klijentovom pretraživaču. Slovo "J" u terminu AJAX prvobitno je označavalo upotrebu JavaScripta, kao i XML-a. Sa porastom JavaScript obrade na strani servera, na primjer, Node.js, koji je prvobitno razvijen 2009. godine, JavaScript se također koristi za dinamičko kreiranje stranica na serveru koje se šalju u potpunosti formiranim klijentima. Execusite je u junu 1997. predstavio prvo dinamičko rješenje za web stranicu za profesionalno tržište.“

### 2.1. HTML(hypertext markup language)

HTML je skraćenica za HyperText Markup Language, što znači prezentacijski jezik za izradu web stranica. Hipertekst dokument stvara se pomoću HTML jezika. HTML jezikom oblikuje se sadržaj i stvaraju se hiperveze hipertekst dokumenta. Prikaz hipertekst dokumenta omogućuje web preglednik. Temeljna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertekst dokument. Svaki HTML dokument sastoji se od osnovnih građevnih blokova - HTML elemenata. Svaki, HTML element sastoji se od para HTML oznaka. Također, svaki element može imati i atribut kojim se definiraju svojstva tog elementa. Na samom početku HTML dokumenta preporučljivo je postaviti !DOCTYPE? element, kojim se označava DTD čime se definira točna verzija standarda koja se koristi za izradu HTML dokumenta. Nakon <!DOCTYPE> elementa,

<html> elementom označava se početak HTML dokumenta. Unutar <html> elementa nalaze se i <head> element te <body> element. <head> element predstavlja zaglavlje HTML dokumenta u kojem se najčešće specificiraju jezične značajke HTML dokumenta kao i sam naslov stranice. Pomoću određenih HTML elemenata unutar zaglavlja dodaju se i stilska obilježja stranice, bila ona direktno ugrađena ili dodana kao referenca na vanjsku CSS datoteku. Često se unutar zaglavlja definišu i skripte kreirane u JavaScript jeziku. U <body> elementu kreira se sadržaj HTML dokumenta, odnosno , stranice koju on predstavlja.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <link rel="stylesheet" type="text/css" href="style.css">
5    <title>Faris site</title>
6  </head>
7  <body class="MainBackground">
8    <header>
9      
10     <nav>
11       <ul class="nav-links">
12         <li><a class="Links" href="Meni.html">Meni</a></li>
13         <li><a class="Links" href="Kuhari.html">Kuhari</a></li>
14         <li><a class="Links" href="Projekat.html">Lokacija</a></li>
15         <li><a class="Links" href="Onama.html">O nama</a></li>
16       </ul>
17     </nav>
18     <a class="cta" href="Contact.html"><button>Kontakt</button></a>
19   </header>
20   <div class="NaslovGlavni">
21     <span class="Naslov1">Restoran</span>
22     <span class="Naslov2">Faris Lindov</span>
23   </div>
24   <br>
25 </body>
26 </html>

```

Slika 1: „Primjer HTML stranice“<sup>1</sup>

### 2.1.1 Klasa i id

Svaki element na stranici ima svoju oznaku, ali nam je i potrebno da posebno obilježavamo elemente ili grupe elemenata. Ove parametre ima smisla dodjeljivati samo HTML elementima unutar body odjeljka. Za obilježavanje elemenata koristimo klase (class) i id-ove (id).

- <div class="contact-adress"><h1>- Faris Lindov - </h1></div>

### 2.1.2 Ostali tagovi

U HTML dokumentu nalazi se „kostur“ web stranice. Sadrži tagove od kojih je građena stranica. Za naslove se koristi tagovi od <h1> do <h6>. U strukturu web stranice ulazi tag <header> kao poglavlje stranice. Navigacijski meni je smješten u <nav> tagu koji je namijenjen za taj dio. U

<sup>1</sup> Slika 1: „Primjer HTML stranice“ – izvor: samostalan rad

tome se nalaze navigacijski linkovi koji se obilježavaju listom tagovima <ul> (unordered list<sup>2</sup>) kao kontenjer za elemente liste, koji se obilježavaju sa <li>(list item). Element liste je u obliku linka <a> koji povezuje glavnu stranicu sa svim ostalim.

- ```
<ul class="nav-links">
  <li><a class="Links" href="Meni.html">Meni</a></li>
  <li><a class="Links" href="Kuhari.html">Kuhari</a></li>
  <li><a class="Links" href="Projekat.html">Lokacija</a></li>
  <li><a class="Links" href="Onama.html">O nama</a></li>
</ul>
```

### 2.1.3 Divajderi

Divajderi <div> su jedan od osnovnih elemenata HTML-a. Oznaka <div> definira podjelu ili odjeljak u HTML dokumentu. Koristi se kao kontejner za HTML elemente - koji se zatim stilizira pomoću CSS-a ili se njime manipulira JavaScript. Div-u je najčešće dodijeljeno ime pomoću klasa i id-ova radi lakšeg manipulisanja. Div može biti parent-div ili child-div. Child-divajderi se nalaze unutar jednog parent-divajdera koji služi kao kontenjer za manje odnosno child-div. Divajderi također mogu biti zvani <section>, <aside>, <footer>, <nav>... u zavisnosti za koju funkciju služe.

- ```
<div class="contact-help">
  <div class="contact-adress"><h1>- Faris Lindov -</h1></div>
  <div class="contact-adress"><h1>- flindov001@gmail.com -</h1></div>
  <div class="contact-adress"><h1>- exe_linda -</h1></div>
</div>
```

## 2.2 CSS

CSS je fronthand programerski jezik i skraćenica je od "Cascading Style Sheets", a služi za definisanje stilova koji određuju izgled HTML elemenata, kao što je manipulacija teksta, boja pozadine, slike, divajdera... CSS može biti inline i vanjski CSS koji je napravljen kao posebn dokument zvan „Style sheet“. Inline CSS se nalazi u html dokumentu i može definisat samo jedan element dokumenta koristeći <style> „<“ tag u koji dodajemo attribute poput boje, fonta, stila i sl. Vanjski CSS možemo povezati sa više HTML dokumenata da bi primijenili stilove na više spojenih dokumenata odjednom.

---

<sup>2</sup> Unordered list je html lista čiji elementi nisu definisani brojevima, već tačkama ili posebnim znakovima.

## 2.2.1 CSS selektori

Selektor identificira elemente na koje se primjenjuju stilovi. Postoji čitav niz selektora pa potpuno upoznavanje primjene svakog od njih nije nimalo banalan zadatak. U CSS-u imamo više načina imenovanja selektora, a oni mogu biti:

- jednostavni selektori
- klasni i id selektori
- pseudoklase

Jednostavni selektor odgovaraju imenu HTML oznake i primjenjuju se na svaki istovrsni element u dokumentu.

- `.li {`  
`font-family: "Montserrat", sans-serif;`  
`font-weight: 500;}`

Klasni i id selektori su najčešći selektori. Odgovaraju imenu klase ili id-a u HTML dokumentu koje je dodijeljeno. Koristi se zbog preglednosti i efikasnosti.

Klasni selektor:

```
.lokacijaNaslov{  
    font-size: 200%;  
    color: #ffffff;  
    margin-left: 120px;  
}
```

Id selektor:

```
#ProduktImage{  
    height: 100%;  
    background-color: black;  
    width: 300px;  
}
```

```
#nav-links li{  
    display: inline-block;  
    padding: 0px 20px;  
}  
.links{  
    transition: font-size 0.2s ease-in-out 0s;  
}  
button{  
    margin-left: 20px;  
    padding: 9px 25px;  
    background-color: rgba(255, 136, 0, 1);  
    border: none;  
    border-radius: 50px;  
    cursor: pointer;  
    transition: all 0.3 ease 0s;  
    color: rgb(0, 0, 0);  
    transition: font-size 0.2s ease 0s;  
}
```

Slika 2: „Primjer svih selektora“<sup>3</sup>

<sup>3</sup> Slika 2: “Primjer svih selektora” – izvor: samostalan rad



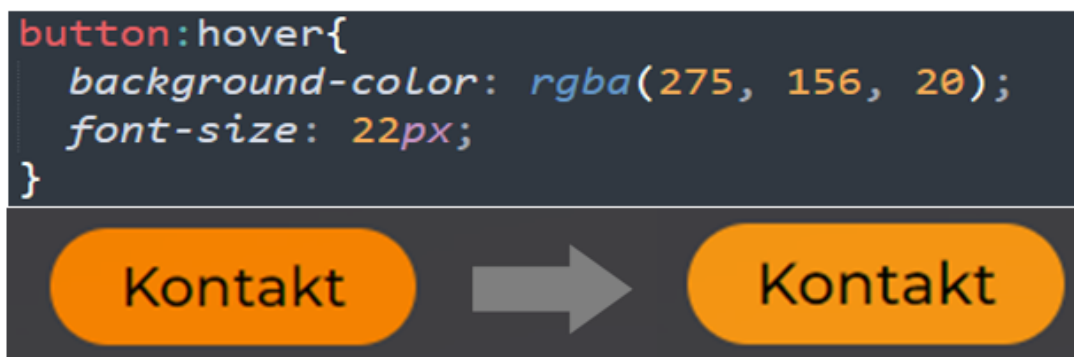
### 2.2.1.1 Pseudoklase

Pseudo-klasa se koristi za definiranje posebnog stanja elementa.

Na primjer, može se koristiti za:

- Stilizirajte element kada korisnik pređe kursorom preko njega
- Različito stilizirajte posjećenog i neposjećenog linka
- Stilizirajte element kada je aktivan

Za stiliziranje elemenata kada je kursor iznad određenog elementa elementa koristimo funkciju *.ImeElementa: hover*. Za ovu web stranicu funkcija hover je korištena za linkove i dugmadi.



Slika 3: „Izgled tranzicije“<sup>4</sup>

Pomoću *:hover* funkcije možemo praviti tranzicije. CSS tranzicije omogućuju promjenu vrijednosti svojstava tijekom zadanog trajanja. Tranziciju pravimo sa funkcijom *transition* u CSS dokumentu. Ova funkcija prima više atributa kao što je vrijeme trajanja, zakašnjenje, vrstu tranzicije itd.

- *transition: all 1s ease-in-out;*

Ovo možemo posebno koristiti sa funkcijama:

- *transition*
- *transition-delay* – zakašnjenje tranzicije
- *transition-duration* – vrijeme trajanja tranzicije
- *transition-property* – koji dio objekta dobija nove vrijednosti
- *transition-timing-function* – vrsta tranzicije. Postoji više vrsta kao što su :
  - *ease* - specificira efekt prijelaza sa sporim početkom, zatim brzim, a zatim polako završava
  - *linear* - specificira efekt tranzicije s istom brzinom od početka do kraja
  - *ease-in* - specificira efekt tranzicije sa sporim startom

<sup>4</sup> Slika 3: „Izgled tranzicije“ – izvor: samostalan rad

- *ease-out* - specificira efekt tranzicije sa sporim krajem
- *ease-in-out* - specificira efekt tranzicije sa sporim početkom i krajem

Za različito stiliziranje linkova koristimo funkciju *.ImeElementa:link* za neposjećene linkove, a *.ImeElementa:visited* za posjećene linkove.

```
a:link {
    color: #FF0000;
}
a:visited {
    color: #00FF00;
}
```

## 2.3 JavaScript

JavaScript je interpretiran objektno orijentisan skriptovan jezik visokog nivoa. On se dominantno koristi pri izradi web stranica uz HTML i CSS. omogućava pravljenje interfejsa koji omogućavaju korisnicima visoku funkcionalnost bez potrebe za novim učitavanjem stranice. Treba znati da JavaScript nije vezan samo za browsere. Brzina i mala memorijska potrošnja JavaScripta u poređenju sa ostalim programskim jezicima doprinosi tome da on ima sve više različitih primjena. JavaScript omogućava kreiranje stranica sa dinamičkim sadržajem. Jedna od najvažnijih mogućnosti jeste čuvanje podataka na uređaju korisnika. To znači da će skripta raditi čak i bez interneta jer su svi podaci smješteni na lokalni kompjuter.

JavaScript koristi nekoliko osnovnih podataka koji omogućavaju rad. To su:

- Brojevi (Numbers) – brojevi kao tip podataka;
- Logičke vrijednosti (Boolean) – postoje samo dvije mogućnosti: tačno (true) ili netačno (false); ove vrijednosti se obično koriste prilikom neke kontrole i testiranja;
- Znakovni nizovi (String) – služi za predstavljanje teksta, tj. za skladištenje slova, brojeva i drugih znakova;
- Null specijalna vrijednost – ukazuje na nepostojanje bilo kakve druge konkretne vrijednosti; to znači da vrijednost postoji, ali da ona nije ni broj, ni tekst, ali ni logička vrijednost;
- Undefined specijalna vrijednost – koristi se pri pokušaju pristupanja vrijednosti neke promjenljive koja je definisana, ali joj nije dodijeljena nikakva vrijednost.

## 2.4 PHP

PHP je jedan programski jezik koji se orijentira po C i Perl sintaksi, namijenjen prvenstveno programiranju dinamičnih web stranica. To je skraćenica za PHP: Hypertext preprocessor. Iako je svoj razvoj započeo kao “hypertext preprocessor” danas je PHP uveliko proširio svoje mogućnosti. Osim kao podrška web aplikacijama danas se PHP koristi i za konzolne aplikacije ali je moguće pisati i aplikacije u PHP-u sa punim grafičkim sučeljem, upotrebom OpenGL biblioteke za trodimenzionalni prikaz i slično. Danas je PHP vrlo moderan programski jezik koji se najčešće koristi na webu. Jedan od razloga vrlo dobre prihvaćenosti PHP-a je vrlo slična sintaksa programskom jeziku C. Veliki broj osnovnih funkcija ima istu sintaksu. Početnicima je također veoma dobar jer nije ograničavajući kao neki drugi programski jezici budući da nema deklaracija tipova varijabli, a moguće je i istu varijablu koristiti za pohranu različitih vrsta vrijednosti. Iako ove karakteristike programski kod čini manje osjetljivim na greške jednostavnije ih je napraviti budući da kompajler na njih neće upozoriti. PHP stranica sadrži tekst i HTML tagove baš kao i standardna HTML stranica, ali uz to PHP stranica može sadržavati i serverske skripte (niz naredbi u nekom od skriptnih jezika koje podržava PHP). Datoteka koja sadrži PHP skripte mora imati ekstenziju **.php** da bi se naredbe u PHP skriptama izvršile. Korištenjem PHP-a možemo generirati cijelu HTML stranicu ili samo jedan njezin dio. Programe pisane u PHP-u nije potrebno prevoditi u izvršni oblik, već se oni izvode prilikom pokretanja u programu koji u realnom vremenu izvršava kod . Programi se pišu kao dio HTML stranice. Sam program se nalazi unutar HTML oznake koji počinje sa `<?php` ,a završava sa `?>` .Sve unutar oznake se smatra PHP programom.

Varijable u PHP-u su nizovi znakova i brojeva koje počinju s oznakom \$. Naziv varijable mora počinjati slovom ili oznakom\_. Ostali znakovi mogu sadržavati brojeve, slova i oznaku. Varijable u PHP-u razlikuju velika i mala slova.

U programskom jeziku PHP nije potrebno deklarirati varijable i na taj način zadati vrstu podataka koja se u njih može smjestiti. Osim toga, moguće je i mijenjati vrstu podataka koji se smješta u neku varijablu. PHP omogućava unutar istog bloka upisivanje numeričke vrijednosti u neku varijablu, da bi se odmah zatim u nju upisao niz znakova. Vrste podataka koje su na raspolaganju su : Boolean, Float, Integer, String. Varijable imaju isti princip rada kao u JavaScriptu. Složenije varijable u PHP su nizovi. Niz definiramo tako što mu dajemo ime i PHP funkciju niza (array).

- `$ImeNiza = array('clan1', 'clan2', 'clan3');`

Funkcija u programiranju je blok koda kome se dodjeljuje ime. Takav kod može biti upotrebljen (pozvan) po potrebi pomoću dodijeljenog imena. Prilikom definisanja funkcije, mogu se zadati ulazni parametri odnosno argumenti. Funkcija će te argumente obraditi, a kao rezultat njenog

rada dobija se određena izlazna vrijednost ili podatak, koji se još nazivaju rezultat funkcije. Funkciju u PHP-u deklariramo pomoću function.

- ```
function Ime_PHPFunkcije ( ){  
    kod...  
}
```

Prilikom poziva, funkciji se mogu predati određeni podaci koje će ona upotrijebiti. Ti podaci se nazivaju argumentima ili parametrima funkcije i moraju se navesti prilikom definicije funkcije. Prilikom navođenja argumenata navodi se samo njihovo ime, a ne navodi se tip podatka.

- ```
function ImeFunkcije($argument1, $argument2) {  
    naredba1;  
    naredba2;  
}
```

```
function IspisiComments($conn){  
    $sql = "SELECT * FROM comments";  
    $result = $conn->query($sql);  
    while($row = $result->fetch_assoc()){  
        echo "<div class='KomentarKutija'><p>";  
        echo $row ['uid']."<br>";  
        echo $row ['date']."<br>";  
        echo $row ['message'];  
        echo "</p>";  
        <form class='UkloniPoruku' method='POST' action='".deleteComments($conn)."'>  
            <input type='hidden' name='cid' value='".$row['cid']."'>  
            <button type='submit' name='commentDelete'>Ukloni</button>  
        </form>  
        </div>";  
    }  
}
```

```
IspisiComments($conn);
```

Slika 4.: „Primjer funkcije i korištenje“<sup>5</sup>

Da bi funkciju pozvali u drugom PHP dokumentu, moramo imati poveznicu ta dva dokumenta. Uključivanje PHP dokumenta u drugi PHP dokument se vrši pomoću funkcije **include**.

- ```
Include 'C:\xampp\htdocs\commentsection\dba.inc.php';
```

<sup>5</sup> Slika 4: “Primjer funkcije i korištenje” – izvor: samostalni rad

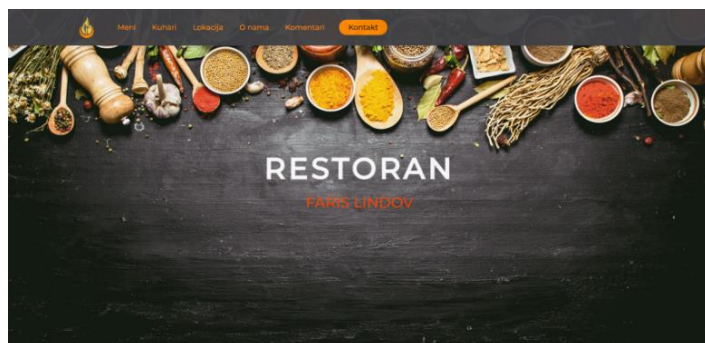
Veza ili konekcija je termin koji se koristi u radu s bazama podataka, a označava korisnikov proces koji se spaja na bazu podataka. Sustav za upravljanje bazama podataka MySQL predefinirano dopušta 100 istovremenih veza što znači da 101. korisnik mora čekati da se zatvori neka od prethodno otvorenih veza, odnosno dobiva poruku o preopterećenju. Veza će se zatvoriti automatski nakon prestanka izvršavanja skripte. Veza sql baze podataka vrši se pomoću funkcije **mysqli\_connect**.

- `$conn=mysqli_connect ('localhost', 'root', 'root', '', 'commentsection');`

## 2.4 Izrada i rješavanje problema

Zadatak je bio napraviti dinamičku web aplikaciju za restoran. Trebalo je upotrijebiti front-end programske jezike za izgled HTML i CSS. Za back-end alate korišten je JavaScript i PHP, koji spaja Web stranicu sa Bazom podataka koja je napravljena u SQL-u za istu temu.

### 2.4.1 Početna stranica



Ovo je izgled početne stranice za website. U header-u se nalazi navigacijski odjelak, a glavni dio stranice pokazuje naslov stranice uz animaciju koja se dešava svaki put kada se otvori web aplikacija. Web stranica ima veoma lagano za koristiti korisničko sučelje.

*Slika 5: „Izgled početne stranice“<sup>6</sup>*

Navigacijski odjeljak spaja svaku stranicu na web aplikaciji. Izgled navigacijskog odjeljka je isti na svakoj stranici. Napravljen je pomoću tagova `<nav>`, lista i `<a>`. Ovako izgleda html za navigacijski odjeljak. Unordered listi je dato ime *nav-links*, jer se unutar nalaze svi ostali linkovi. Linkovima je dato ime *Links*. Href označava putanju linka za web stranice.

---

<sup>6</sup> Slika 5: „Izgled početne stranice“ – izvor: samostalan rad

```

<nav>
  <ul class="nav-links">
    <li><a class="Links" href="Meni.html">Meni</a></li>
    <li><a class="Links" href="Kuhari.html">Kuhari</a></li>
    <li><a class="Links" href="Projekat.html">Lokacija</a></li>
    <li><a class="Links" href="Onama.html">O nama</a></li>
    <li><a class="Links" href="Onama.php">Komentari</a></li>
  </ul>
</nav>

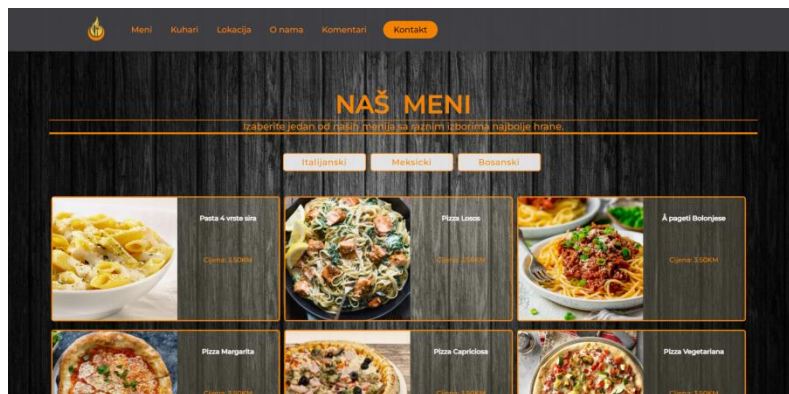
.nav-links li{
  display: inline-block;
  padding: 0px 20px;
}
.Links{
  transition: font-size 0.2s ease-in-out 0s;
}
.Links:hover{
  color: rgb(300, 300, 300);
  font-size: 21px;
}

```

Slika 6: „HTML i CSS za navigacijski odjeljak“<sup>7</sup>

Display: inline-block omogućuje postavljanje širine i visine elementa. Tranzicija za likove se dešava kada se kursor nađe iznad linka. Tada se mijenja boja i veličina fonta linka.

## 2.4.2 Meni



Slika 7: „Izgled stranice za meni“<sup>8</sup>

Ovako izgleda meni restorana. Ispod glavnog naslova i teksta nalazi se odjeljak za odabir vrste menija. Svaki novi ulazak u stranicu, pokazuje svako jelo, ali kada se odabere meni, stranica pokazuje samo jela koja su vezana za taj meni (italijanski, meksički, bosanski).

Za odabir vrste menija korišten je JavaScript. Svakoj vrsti menija dato je ime *list*. *Data-filter* nam govori o kojoj vrsti menija će jelo pripadati.

<sup>7</sup> Slika 6: „HTML i CSS za navigacijski odjeljak“ – izvor: samostalan rad

<sup>8</sup> Slika 7: „Izgled stranice za meni“ – izvor: samostalan rad

```

<ul class="VrsteMenia">
<li class="list" data-filter="Italijanski"> Italijanski</li>
<li class="list" data-filter="Meksički"> Meksicki</li>
<li class="list" data-filter="Bosanski"> Bosanski</li>
</ul>

<div class="Produkti">
  <div class="Produkt Italijanski"><div class="ProduktImage"></div>
  <h4 style="margin-top: 0px;">Pasta 4 vrste sira</h4><p class="CijenaProdukta">Cijena: 3.50KM</p></div>
  <div class="Produkt Italijanski"><div class="ProduktImage"></div>
  <h4 style="margin-top: 0px;">Pizza Losos</h4><p class="CijenaProdukta">Cijena: 3.50KM</p></div>

```

Slika 8: „HTML za odabir i produkte menija“<sup>9</sup>

Sva jela su smještena u jedan parent divajder kojem je data klasa *Produkti*. U njemu se nalaze više child divajdera koja su zapravo blokovi za jelo. Blokovima za jelo je dat naziv *Produkt*. Nakon klase određeno je kojem meniju pripada, odnosno *data-filter*-u. Svaki blok za jelo sadrži sliku, ime jela i cijenu. Cijena je uzeta iz MySQL baze podataka u kojoj se nalaze podatci za restoran. Odabir menija automatski pokreće JavaScript kod koji sakriva ostala jela i prikazuje samo ona koja su odabrana.

- ```

<script type="text/javascript">
$(document).ready(function(){
$('.list').click(function(){
const value = $(this).attr('data-filter');
if (value == 'sve'){
$('.Produkt').show('1000');
}
else{
$('.Produkt').not('.'+value).hide('1000');
$('.Produkt').filter('.'+value).show('1000');
}
})
})
</script>

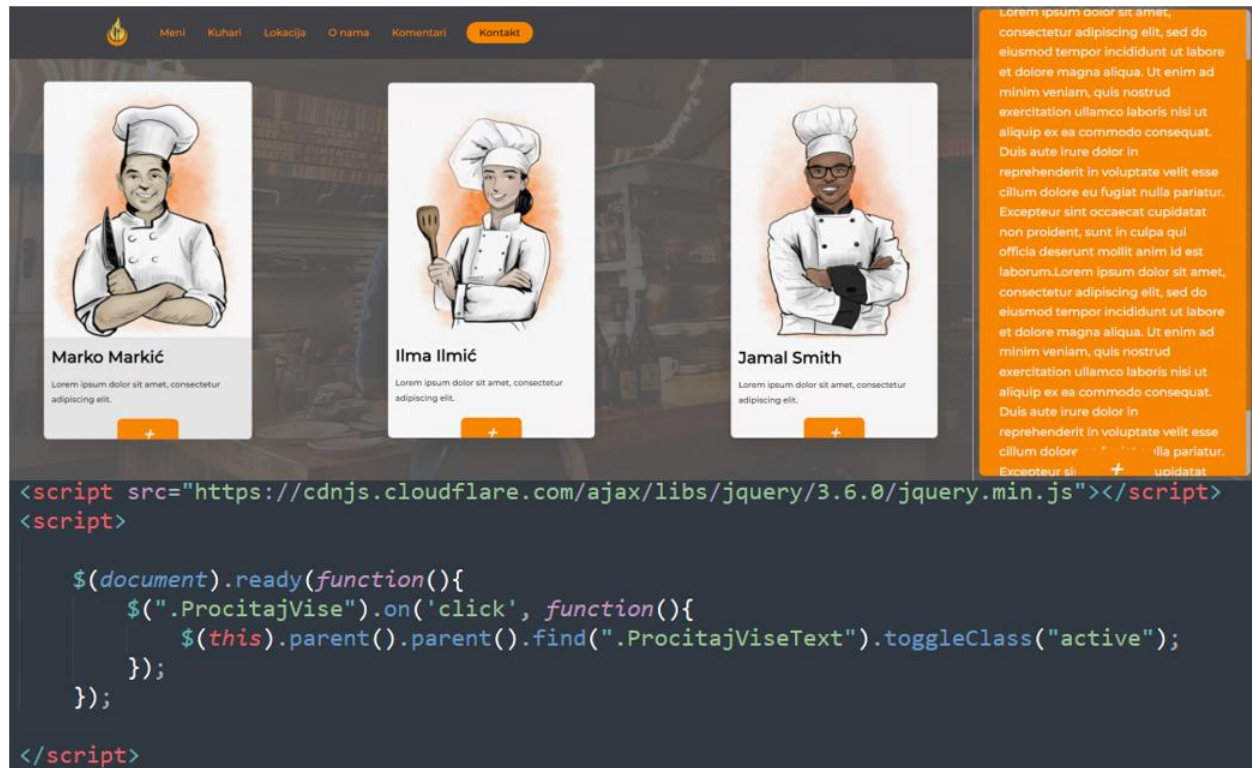
```

Kreirana je konstanta pod imenom *value* koja ima vrijednost istu kao i *data-filter* u divajderu *Produkt*. Funkcijom *if else* uklanjamo blokove koji nisu pod tom klasom (ako se u klasi odabranog produkta ne nalazi ključna riječ Italijanski, Meksički ili Bosanski, funkcija sakriva sve blokove, a blokove u kojima se nalazi prikazuje).

<sup>9</sup> Slika 8: „HTML za odabir i produkte menija“ – izvor: samostalan rad



## 2.4.3 Kuhari



Slika 9: „Izgled i funkcija stranice za kuhare“<sup>10</sup>

Ovo je stranica za pregled kuhara u restoranu. U ova tri bloka idu slike kuhara, njihovo ime i prezime, kratak tekst i pročitaj više plus. Na ovoj stranici je također upotrebljen JavaScript za pročitaj više plus.

Kada se klikne plus on će imati vrijednost **active**. U CSS-u kada je vrijednost active prikazat će više teksta. Kada se ponovo klikne vrijednost **active** će nestati, zatim će se vratiti na početni blok.

<sup>10</sup> Slika 9: „Izgled i funkcija stranice za kuhare“ – izvor: samostalan rad



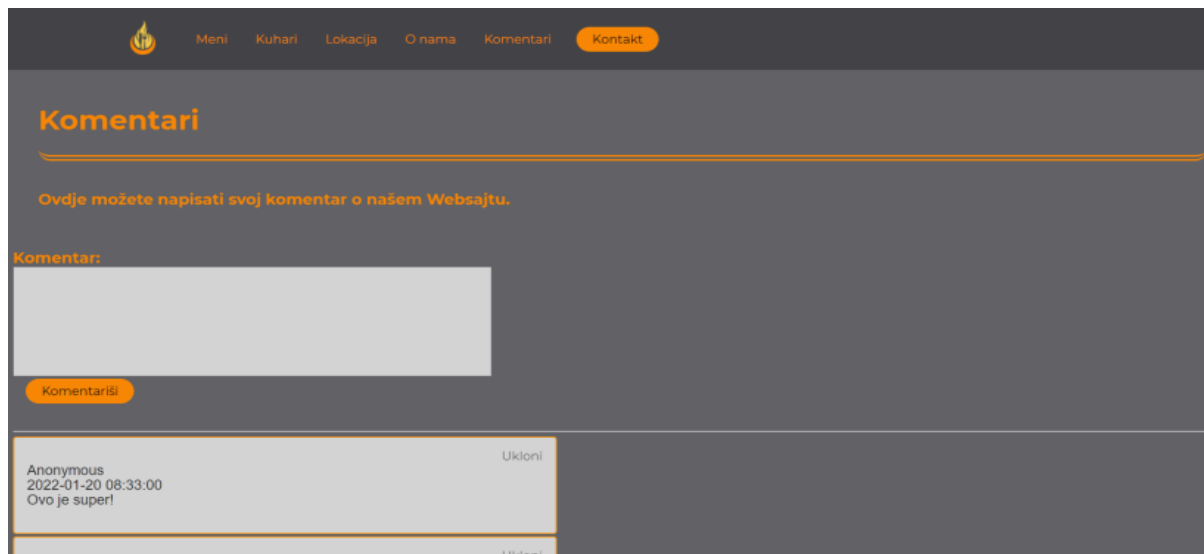
## 2.4.4 Lokacija i o nama stranica



Slika 10: „Izgled stranice o nama i lokaciju“<sup>11</sup>

Ovo je izgled stranice za lokaciju. Na njoj se nalazi lokacija u obliku <iframe> uzeta sa Google Maps. Na desnoj strani se nalazi blok sa radim vremenom radnim danima i vikendom. O nama je stranica koja opisuje restoran. Na ovoj stranici se nalazi slike i tekst o restoranu.

## 2.4.6 Komentari



Slika 11: „Izgled stranice za komentare“<sup>12</sup>

Ovo je izgled stranice za komentare. Ovdje korisnici mogu ostaviti svoj komentar za web aplikaciju. Nakon što korisnik upiše svoj komentar u <textarea> i klikne dugme komentiraj, tekst koji je napisan ide u bazu podataka *commentsection*. Nakon toga komentar iz baze podataka se pokazuje na ovoj stranici u obliku divajdera.

<sup>11</sup> Slika 10: „Izgled stranice o nama i lokaciju“ – izvor: samostalan rad

<sup>12</sup> Slika 11: „Izgled stranice za komentare“ – izvor: samostalan rad

Za ovu stranicu korištena su tri PHP dokumenta. Prvi dokument **Komentari.php** služi za izgled stranice i iskorištavanje funkcija od ostale dvije stranice. Drugi php dokument **dbh.inc.php** služi za spajanje PHP dokumenta o komentarima sa MySQL bazom podataka. Treći dokument **comment.inc.php** služi za funkcije koje su iskorištene u početnom PHP dokumentu. Funkcije služe nakon što korisnik unese komentar i klikne dugme komentariši da se tekst smjesti u MySQL bazu podataka na mjesto *message*. Pored teksta, na stranici se ispisuje vrijeme i datum kada je komentar napisan i korisničko ime kojem je data konstantna vrijednost „Anonymous“. U ovoj stranici također se nalazi funkcija da se ukloni komentar. Nakon što korisnik klikne ukloni, taj komentar će se izbrisati sa stranice i izbrisati iz baze podataka. Za ove funkcije korišten je PHP jezik.

Skripta za spajanje baze podataka i web stranice:

- ```
<?php
$conn = mysqli_connect('localhost', 'root', '', 'commentsection');
if (!$conn){
    die ("Connection failed: ".mysqli_connect_error());
}
?>
```

Funkcija za ispis komentara iz baze podataka na stranicu:

- ```
function IspisiComments($conn){
    $sql = "SELECT * FROM comments";
    $result = $conn->query($sql);
    while($row = $result->fetch_assoc()){
        echo "<div class='KomentarKutija'><p>";
        echo $row ['uid']. "<br>";
        echo $row ['date']. "<br>";
        echo $row ['message'];
        echo "</p>";
        <form class='UkloniPoruku' method='POST' action=''.deleteComments($conn).''>
        <input type='hidden' name='cid' value=''. $row['cid'].''>
        <button type= 'submit' name='commentDelete'>Ukloni</button>
        </form>
        </div>";
    }
}
```

Funkcija za brisanje komentara sa stranice i iz baze podataka:

- ```
function deleteComments($conn){
    if (isset($_POST['commentDelete'])) {
        $cid=$_POST['cid'];
        $sql = "DELETE FROM comments WHERE cid='$cid'";
```

```

$result = $conn->query($sql);
header("Location: Onama.php");
}
}

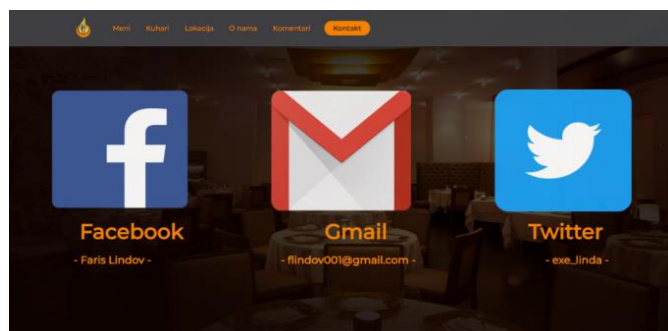
```

Tabela za bazu podataka sadrži četiri atributa (cid, uid, date, message). Cid je broj komentara, uid je ime korisnika koji je komentarisao, date pamti datum i vrijeme komentara i message pamti tekst komentara.

|                          |      |      |        | cid | uid       | date                | message                     |
|--------------------------|------|------|--------|-----|-----------|---------------------|-----------------------------|
| <input type="checkbox"/> | Edit | Copy | Delete | 1   | Anonymous | 2022-01-20 08:33:00 | Ovo je super!               |
| <input type="checkbox"/> | Edit | Copy | Delete | 2   | Anonymous | 2022-01-20 08:33:18 | Odlican restoran!<3         |
| <input type="checkbox"/> | Edit | Copy | Delete | 11  | Anonymous | 2022-01-20 08:53:46 | Ovo je website za resotran. |
| <input type="checkbox"/> | Edit | Copy | Delete | 17  | Anonymous | 2022-01-20 08:54:39 | svidza mi se meni           |

Slika 12: „Baza podataka za komentare“<sup>13</sup>

## 2.4.7 Kontakt



Ovo je izgled stranice za kontakt. Nalaze se tri divajdera sa tranzicijom kada je kursor iznad ikone za Facebook, Gmail ili Twitter. Klikom na ikonu upućuje na stranicu osnivača.

Slika 13: „Izgled stranice za kontakt“<sup>14</sup>

<sup>13</sup> Slika 12: „Baza podataka za komentare“ – izvor: samostalan rad

<sup>14</sup> Slika 13: „Izgled stranice za kontakt“ – izvor: samostalan rad

### **3. Zaključak**

---

Danas su web aplikacije sve češće. Nove tehnologije i alati omogućavaju da kreiranje bude što lakše i efikasnije. Dinamičke stranice su ugrađeni programi koji omogućavaju automatske promjene nekih podataka na stranicama bez intervencije autora stranice i one su interaktivne sa korisnicima.

U ovom maturskom radu smo prikazali elemente za izradu moderne dinamičke web aplikacije i izrada dinamičke web aplikacije - Restoran. Stranica je izrađena uz pomoć PHP-a, HTML-a, CSS-a i JavaScripta. Baza web stranice izrađena je u MySQL alatu. Baza podataka i stranica su spojene sa XAMPP-om. Baza se sastoji od dvije tablice: prva je namjenjena samo za komentare, a druga je za prikazivanje podataka kao što su imena kuhara, cijene jela, datumi... Stranica je rađena pomoću Sublime Text 3 koji pruža mnoge moderne mogućnosti i pametne alate. U web aplikaciji se nalazi sedam stranica: početna, meni, kuhari, lokacija, o nama, komentari i kontakt. Pristup komentarima imaju svi korisnici.

## 4. Literatura

---

1. Designing and Developing a dynamic website using PHP  
[https://www.theseus.fi/bitstream/handle/10024/346960/Kamal\\_lamsal.pdf](https://www.theseus.fi/bitstream/handle/10024/346960/Kamal_lamsal.pdf)
2. Dynamic web sites M. Gabbrielli  
<https://www.w3.org/People/Massimo/papers/rmm.pdf>
3. „Information revolution“(2005) – J.R. Okin
4. PHP toturijal, <https://www.w3schools.com/php/>
5. PHP and MySQL Web Development Fifth Edition – Luke Weiling, Laura Thomson
6. Uvod u PHP i MySQL D350(2014) - Edin Mujadžević  
[https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajeви/d350\\_polaznik.pdf](https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajeви/d350_polaznik.pdf)

## **7. Popis slika**

---

Slika 1 – Primjer HTML stranice

Slika 2 – Primjer svih selektora

Slika 3 – Izgled tranzicije

Slika 4 – Primjer funkcije i korištenje

Slika 5 – Izgled početne stranice

Slika 6 – HTML i CSS za navigacijski odjeljak

Slika 7 – Izgled stranice za meni

Slika 8 – HTML za odabir i proizvode menija

Slika 9 – Izgled i funkcija stranice za kuhare

Slika 10 – Izgled stranice o nama i lokaciju

Slika 11 – Izgled stranice za komentare

Slika 12 – Baza podataka za komentare

Slika 13 – Izgled stranice za kontakt