

Univerzitet u Novom Sadu

Tehnički fakultet ”Mihajlo Pupin” Zrenjanin

SEMINARSKI RAD

iz predmeta Razvoj softvera otvorenog koda

Tema: Šlep Služba

|  |  |
| --- | --- |
| Predmetni nastavik: | Autori: |
| Doc. dr Kazi Zoltan | Stefan Zagorac SI 8/16 |
|  | Emanuel Madaras Suču SI 2/16  Ilija Agbaba SI 22/16 |

Zrenjanin 2022.

**Sadržaj**

[1. Uvod 3](#_Toc114015809)

[2. Izrada navigacionog dela 8](#_Toc114015810)

[3. Izrada dela “O nama” 13](#_Toc114015811)

[4. Izrada Loyalty sekcije 16](#_Toc114015812)

[5. Izrada sekcije usluga 18](#_Toc114015813)

[6. Izrada Kontakt sekcije 20](#_Toc114015814)

[7. Kreiranje Footera 22](#_Toc114015815)

[8. Logika stranice za registraciju i logovanje 23](#_Toc114015816)

[9. Logika komentara 26](#_Toc114015817)

[10. Kreiranje Baze podataka 27](#_Toc114015818)

[11. Povezivanje sa Bazom podataka 28](#_Toc114015819)

[12. Zaključak 30](#_Toc114015820)

[13. Literatura 31](#_Toc114015821)

# 1. Uvod

Seminarski rad iz predmeta Razvoj softvera otvorenog koda pod nazivom “Šlep služba” je slojevita aplikacija koja obuhvata osnovne funkcionalnosti sajta za pomoć na putu tj. šlep službe.

Aplikacija sadrži mogućnost registracije i logina koji kreiranjem naloga dobijaju mogućnost da ostave komentar vezan za poslovanje firme iz njihovog ličnog iskustva.

Osnovna funkcionalnost jeste reklama firme koja se bavi pomoći na putu u slučaju nezgode ili kvara, informacije o uslugama koja firma pruža, informacije o loyalty programu i kontakt informacije koje su i najvažnije za ovakvu vrstu firme.



*Slika 1. Logo firme*

Kada se otvori, websajt u gornjem delu sajta se nalazi tab navigacije sa dugmadima koje vode na određene sekcije sajta:

* Početna stranica
* O nama
* Loyalty program
* Usluge
* Kontakt
* Login
* Register

Svako dugme vodi na drugačiji deo sajta. Sajt je modernog izgleda i izrađen je delom kao one page site, gde je korisnicima sve osim login-a i registracije dostupno u jednom prozoru sajta, a skrolovanjem se stiže do željenih odeljaka.

Prvo što se prikazuje kada se uđe na sajt je logo, naziv i slogan firme što je ustvari i početna stranica.

U delu „O nama” detaljnije je opisano poslovanje i usluge koje firma pruža sa slikom vozila koja se koriste.

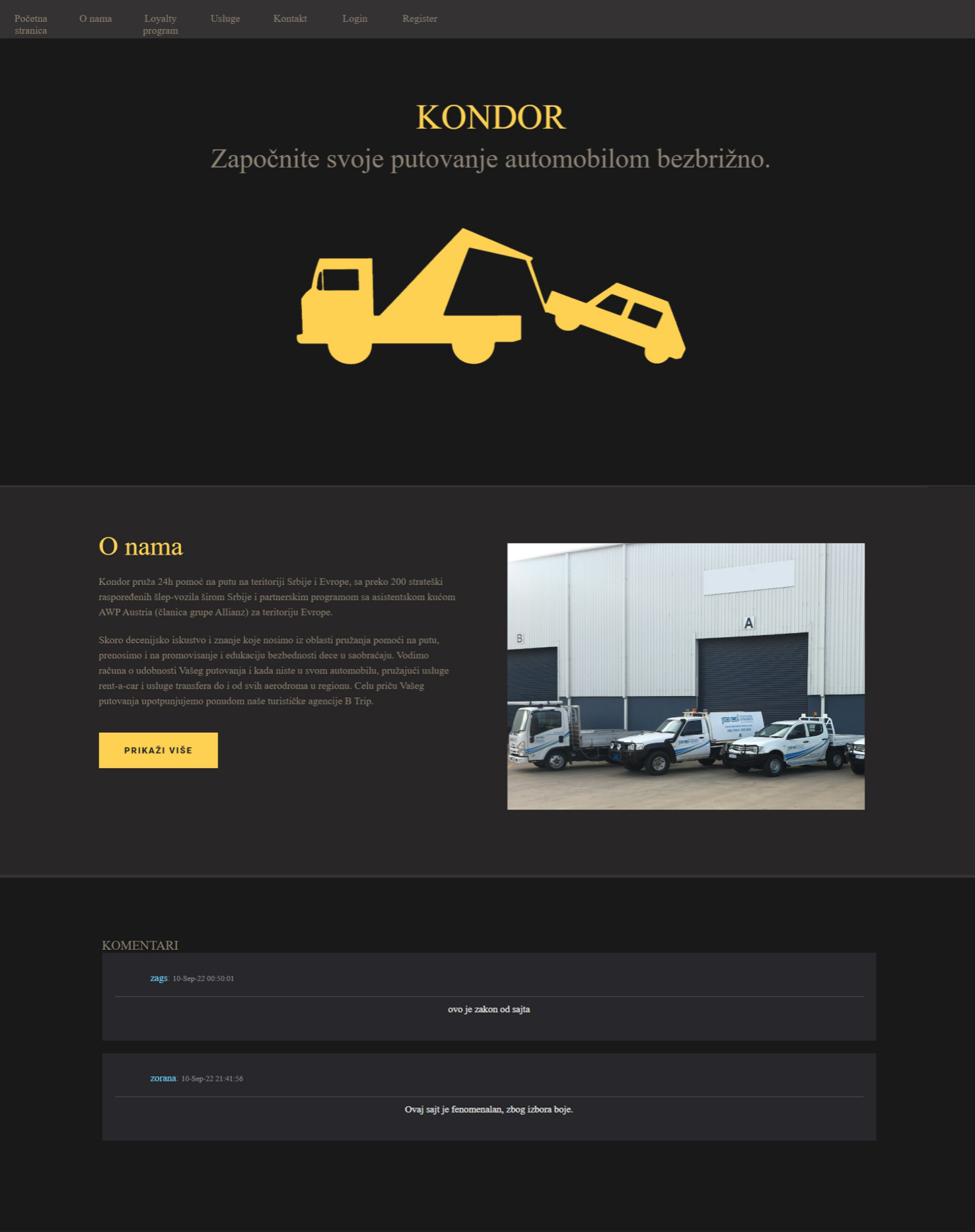
Zatim sledi deo vezan za loyalty program firme u kome u zavisnosti od toga da li je posetilac registrovan i ulogovan razlikuje u tome da ulogovani korisnik ima mogućnost postavljanja komentara dok posetilac koji nije ulogovan ima samo uvid u komentare

Ispod njega se nalazi deo koji pruža informacije o uslugama i cene za određene pakete sa cenama

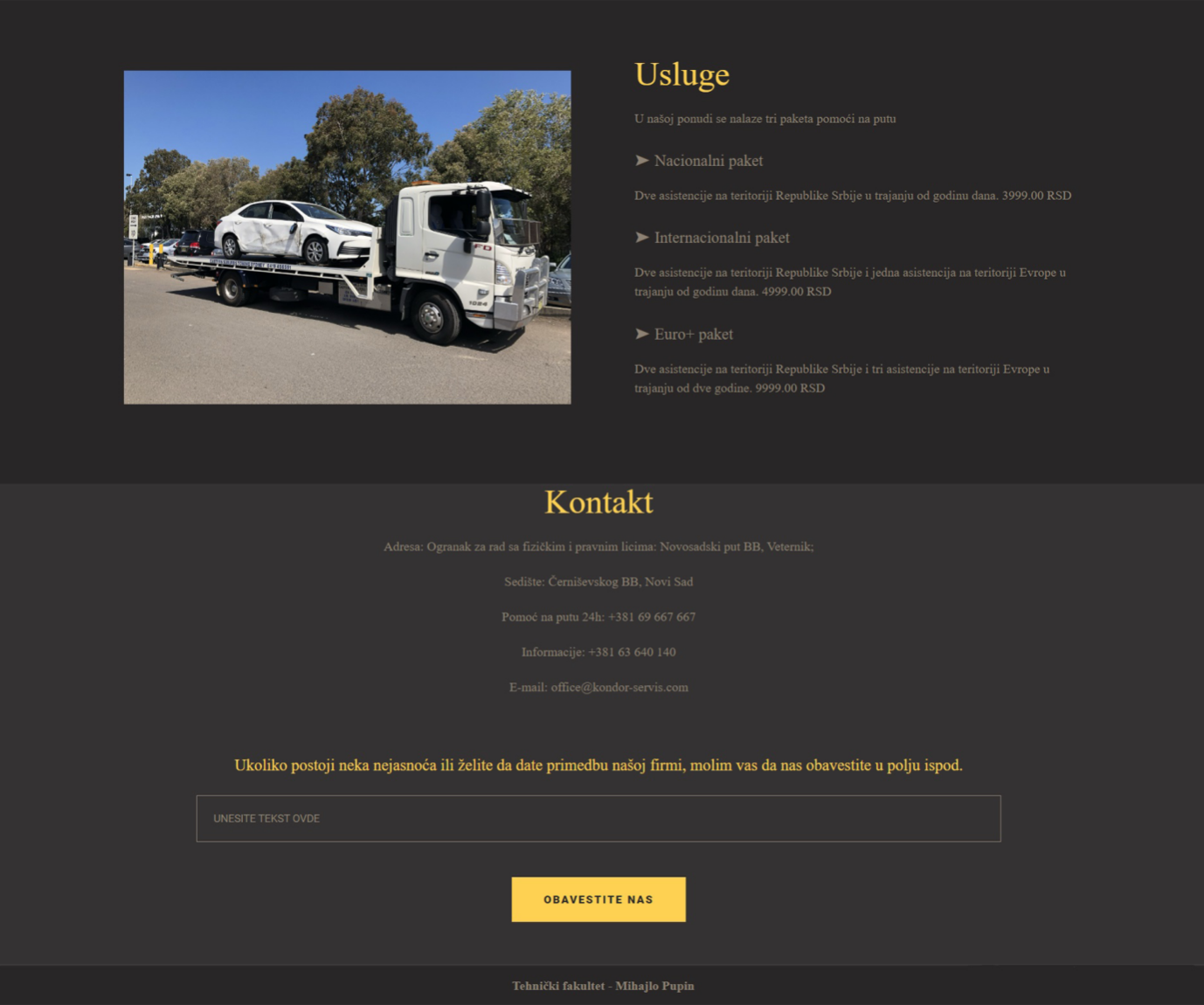
Sledeći deo sajta predstavlja Kontakt deo, u kome se nalaze: Adresa, sedište firme, kontakt telefon za 24h pomoć na putu, kontakt telefon za informacije, e-mail adresa i textbox u kome se može direktno postaviti pitanje dostupnom operateru.

Poslednji deo ovog sajta je footer u kome je napisan naziv našeg fakulteta

Frontend sajta je responzivan što znači da se može koristiti i za mobilne uređaje. Kada se smanji veličina sajta za mobilne uređaje ili tablete, gornji meni sada postaje padajući meni koji se otvara klikom na tri horizontalne linije u gornjem desnom uglu sajta.



*Slika 2. Prvi deo izgleda glavne stranice sajta*



*Slika 3. Drugi deo izgleda glavne stranice sajta*



*Slika 4. Izgled login stranice*



*Slika 5. Izgled stranice za registraciju*

# 2. Izrada navigacionog dela

Prva stvar koja se za ovaj sajt stvara jeste navigacioni deo koji će da sadrži dugmad koja su putem *hyperlink-a* povezana sa delom sajta koje predstavljaju.

Za početak, pravi se fajl *index.html* gde će biti ubačen sav html kod. On se sastoji od *head* i *body* dela.

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="UTF-8">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<meta http-equiv="X-UA-Compatible" content="ie=edge">*

*<title>Seminarski rad - Veb dizajn</title>*

*<!-- referenca za css -->*

*<link rel="stylesheet" href="css/style.css">*

*<!-- referenca za font -->*

*<link href="https://fonts.googleapis.com/css?family=Roboto:100,100i,300,300i,400,400i,500,500i,700,700i,900,900i" rel="stylesheet">*

*<!-- referenca za jquery -->*

*<script src="https://code.jquery.com/jquery-3.3.1.js"></script>*

*<!-- referenca za font ikonica -->*

*<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">*

*</head>*

U head sekciji smo definisali charset, naziv stranice, dodali reference za css, font, jquery i font ikonica.

Sada na red dolazi *body deo,* gde je prva stvar koju smo uradili upravo navigacioni deo sajta.

*<header>*

*<nav class="menu">*

*<ul class="menu">*

*<li>*

*<a href="#pocetna">Početna stranica</a>*

*</li>*

*<li>*

*<a href="#onama">O nama</a>*

*</li>*

*<li>*

*<a href="#loyalty">Loyalty program</a>*

*</li>*

*<li>*

*<a href="#usluge">Usluge</a>*

*</li>*

*<li>*

*<a href="#kontakt">Kontakt</a>*

*</li>*

*<li>*

*@if (Request.IsAuthenticated)*

*{*

*@Html.ActionLink("Log Out", "Logout", "Account")*

*}*

*else*

*{*

*@Html.ActionLink("Login", "Login", "Account")*

*}*

*</li>*

*<li>*

*@if (!Request.IsAuthenticated)*

*{*

*@Html.ActionLink("Register", "Register", "Account")*

*}*

*</li>*

*<li>*

*@if (Request.IsAuthenticated)*

*{*

*<p class="welcome-sign">Dobrodošao, @Html.Encode(User.Identity.Name)</p>*

*}*

*</li>*

*</ul>*

*<a href="#" id="toggle-btn">Meni</a>*

*</nav>*

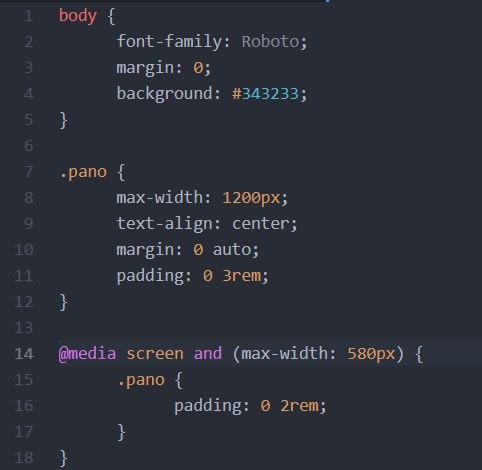
*</header>*

Korišćenjem *ul taga (unordered list)* kreirali pet elemenata: Početna , O nama, Loyalty, Usluge i Kontakt. Za svaki element smo postavili referentnu vrednost na #\_\_\_\_\_(ime elementa), kako bi kasnije kreiranu

klasu povezali sa navigacionim delom i kako bi, kada kliknemo misem na element bili preusmereni da tačno taj deo sajta koji nam treba.

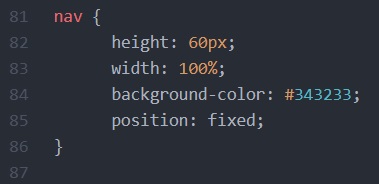
Kako bi navigacioni deo bio vizualno pregledan i moderan, napravili smo novi fajl *style.css* gde ćemo stilizovati svaki deo našeg sajta.

Postavili smo osnovne vrednosti, kako bi resetovali margine i padding kao i odabrali font koji ćemo koristiti tokom celog projekta (*Roboto*).



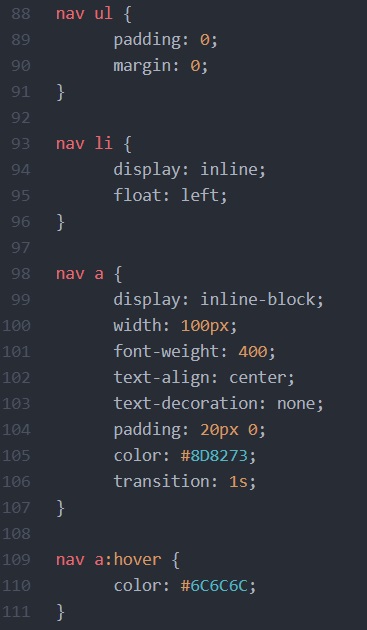
*Slika 6. Osnovne vrednosti u style.css fajlu*

Zatim stilizujemo navigacioni deo. Koristimo naziv nav**{}** kako bi označili celi navigacioni deo



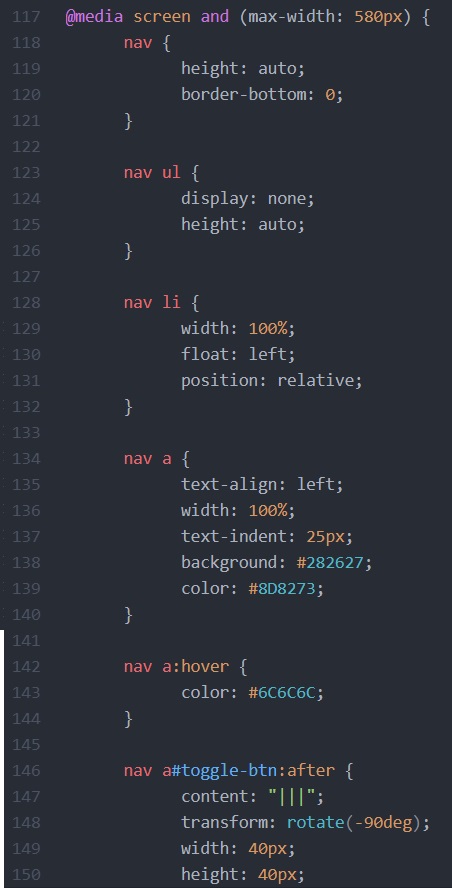
*Slika 7. Vrednosti za avigaciju u css fajlu*

Zatim stilizujemo posebne delove naše navigacije, tj elemente navigacione liste.



*Slika 8. Stilizovanje elemenata navigacionog dela.*

Kada se ovaj websajt koristi putem telefona, navigacioni deo ne može biti isti kao i na računaru zbog rezolucije i veličine ekrana, zato je neophodno napraviti modernu tranziciju, tj. dugme koje nakon klika mišem otvara padajući meni i drugacije stiliyovati tab-ove.

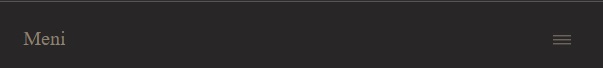


*Slika 9. CSS za mobilnu navigaciju*

Sve gore navedene stvari su neophodne da bi navigacioni deo izgleda tako kako sada izgleda.



*Slika 10. Konačan izgled navigacionog dela za desktop*



*Slika 11. Konačan izgled navigacionog dela za mobile uređaje i tablete*

# 3. Izrada dela “O nama”

U ovom delu sajta doali smo klasu, heading, dva pragrafa i sliku



*Slika 12. Manipulisanje html fajlom kako bi ubacili deo “o nama”*

Nakon toga doterujemo sekciju putem CSS-a



*Slika 13. Css za sekciju o nama*



*Slika 14. Konačan izgled sekcije o nama desktop verzija*

**

*Slika 15. Konačan izgled sekcije o nama mobilna verzija*

# 4. Izrada Loyalty sekcije

U Loyalty sekciji imamo prikaz komentara za posetioce koji nisu ulogovani i mogućnost dodavanja komentara za registrovane i ulogovane korisnike.

@if (Request.IsAuthenticated)

{

<form method="post">

<!--<input name="ime" type="text" class="feedback-input" placeholder="Ime" oninvalid="this.setCustomValidity('Unesite ime')" oninput="this.setCustomValidity('')" />-->

<!-- <input name="email" type="text" class="feedback-input" placeholder="Email" /> -->

<textarea name="tekstKomentara" class="feedback-input" placeholder="Komentar" required oninvalid="this.setCustomValidity('Unesite tekst komentara')" oninput="this.setCustomValidity('')"></textarea>

<input type="submit" value="Komentarisi" />

</form>

}

<div class="comments-section">

<div class="comments">

<h4>KOMENTARI</h4>

<div id="comments-container">

<!--Za svaki komentar se pravi komentar polje u kojem se nalazi ime, datum i tekst komentara-->

@foreach (Komentar k in @Model)

{

<div class="comment">

<div class="comment-user">

<span class="user-details"><span class="username">@k.ime</span><span>: </span><span>@k.datum</span></span>

</div>

<div class="comment-text">@k.tekstKomentara</div>

</div>

}

</div>

</div>

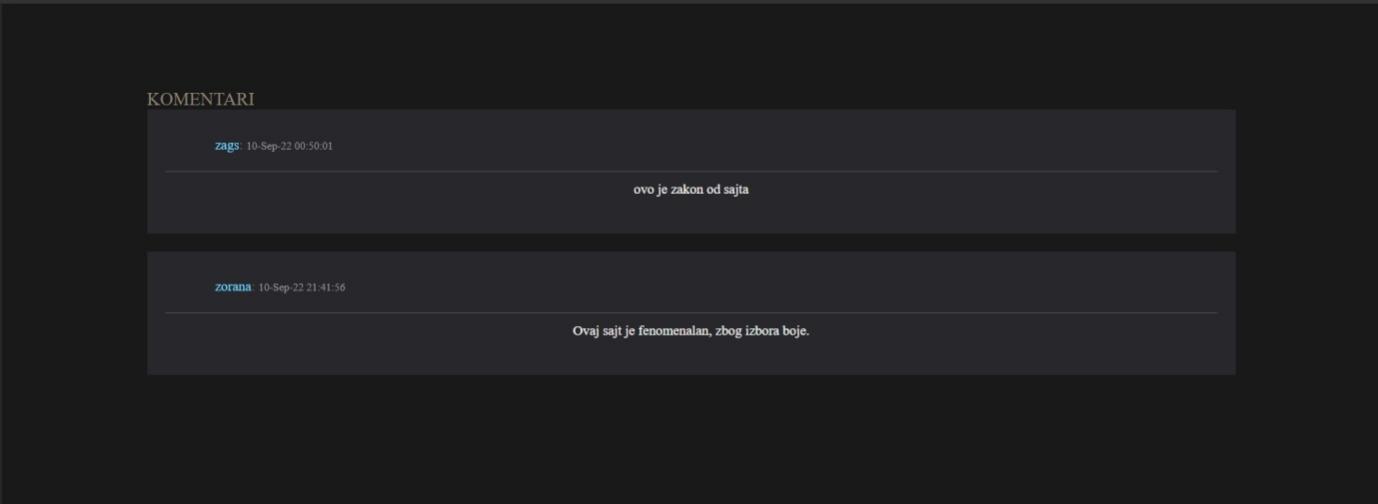
</div>

Opet, da bi ovaj deo websajta izgledao lepo, vraćamo se na manipulisanjem CSS fajla.



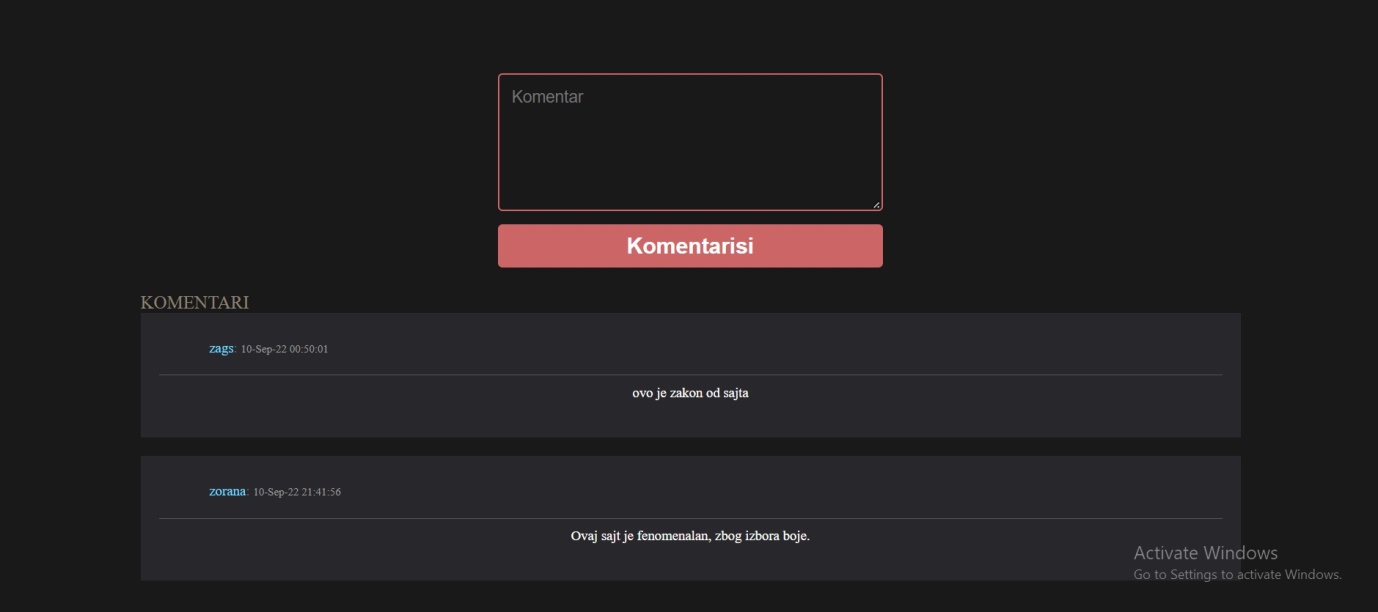
*Slika 16. CSS kod za Loyalty sekciju*

Konačan izgled loyalty sekcije za posetioce koji nisu ulogovani izgleda ovako izgleda ovako:



*Slika 17. Konačan izgled loyalty sekcije za neulogovane*

Za one koji su registrrovani i ulogovani izgleda ovako:



*Slika 18. Konačan izgled loyalty sekcije za ulogovane*

# 5. Izrada sekcije usluga

Ovo predstavlja glavni deo sajta, gde korisnik zapravo prima informacije o uslugama koje firma pruža uz detaljan pis usluge i cenu.

<a name="usluge"></a>

<div id="usluge" class="sekcija">

<div class="pano">

<div class="usluge-sadrzaj">

<div>

<img src="~/Content/Img/usluge.jpg" alt="">

</div>

<div>

<h2>Usluge</h2>

<p>U našoj ponudi se nalaze tri paketa pomoći na putu</p>

<h4>&#10148; Nacionalni paket</h4>

<p>

Dve asistencije na teritoriji Republike Srbije u trajanju od godinu dana.

3999.00 RSD

</p>

<h4>&#10148; Internacionalni paket</h4>

<p>

Dve asistencije na teritoriji Republike Srbije i jedna asistencija na teritoriji Evrope u trajanju od godinu dana.

4999.00 RSD

</p>

<h4>&#10148; Euro+ paket</h4>

<p>

Dve asistencije na teritoriji Republike Srbije i tri asistencije na teritoriji Evrope u trajanju od dve godine.

9999.00 RSD

</p>

</div>

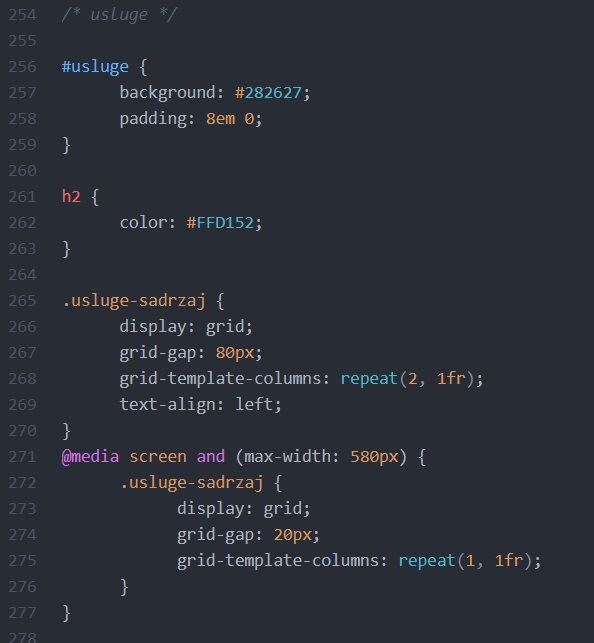
</div>

</div>

</div>

*Html kod za stranicu usluga*

Naravno opet da bi smo doterali izgled sekcije kako za desktop tako i za mobilu verziju maniuišemo CSS fajlom:



*Slika 19. CSS kod za sekciju usluga*

Na kraju izgled sekcije usluga izgleda ovako:



*Slika 20. Izgled sekcije usluga*

# 6. Izrada Kontakt sekcije

Kontakt sekcija, kao najvažnije sekcija 90% sajtova za firme i preduzeća koje se bave uslužnim delatnostima zaslužuje posebnu pažnju pa je na nju osim uobičajenih elemenata dodat i text box za unos pitanja i primedbi.

<a name="kontakt"></a>

<div id="kontakt" class="sekcija">

<div class="pano">

<h2>Kontakt</h2>

<p>Adresa: Ogranak za rad sa fizičkim i pravnim licima: Novosadski put BB, Veternik;</p>

<p>Sedište: Černiševskog BB, Novi Sad</p>

<p>Pomoć na putu 24h: +381 69 667 667</p>

<p>Informacije: +381 63 640 140</p>

<p>E-mail: office@kondor-servis.com</p>

<h4 align="center">Ukoliko postoji neka nejasnoća ili želite da date primedbu našoj firmi, molim vas da nas obavestite u polju ispod.</h4>

<input type="text" placeholder="Unesite tekst ovde">

<br>

<button type="button">Obavestite nas</button>

</div>

</div>

*HTML kod za kontakt sekciju*

CSS je dodat naravno kako za desktop tako i za mobilnu verziju websajta jednim definisanjem parametara.

/\* kontakt sekcija \*/

#kontakt {

background: #343233;

padding: 8em 0;

}

#kontakt h2 {

color: #FFD152;

margin-top: -10vh;

}

#kontakt h4 {

color: #FFD152;

margin-top: 10vh

}

#kontakt input {

width: 80%;

margin: 2em 0;

background: none;

border: 1px solid #8D8273;

padding: 20px;

font-family: Roboto;

font-weight: 400;

color: rgb(211, 165, 100);

}

#kontakt input:focus {

border: none;

outline: none;

border: 2px solid #8D8273;

}

input::placeholder {

color: #8D8273;

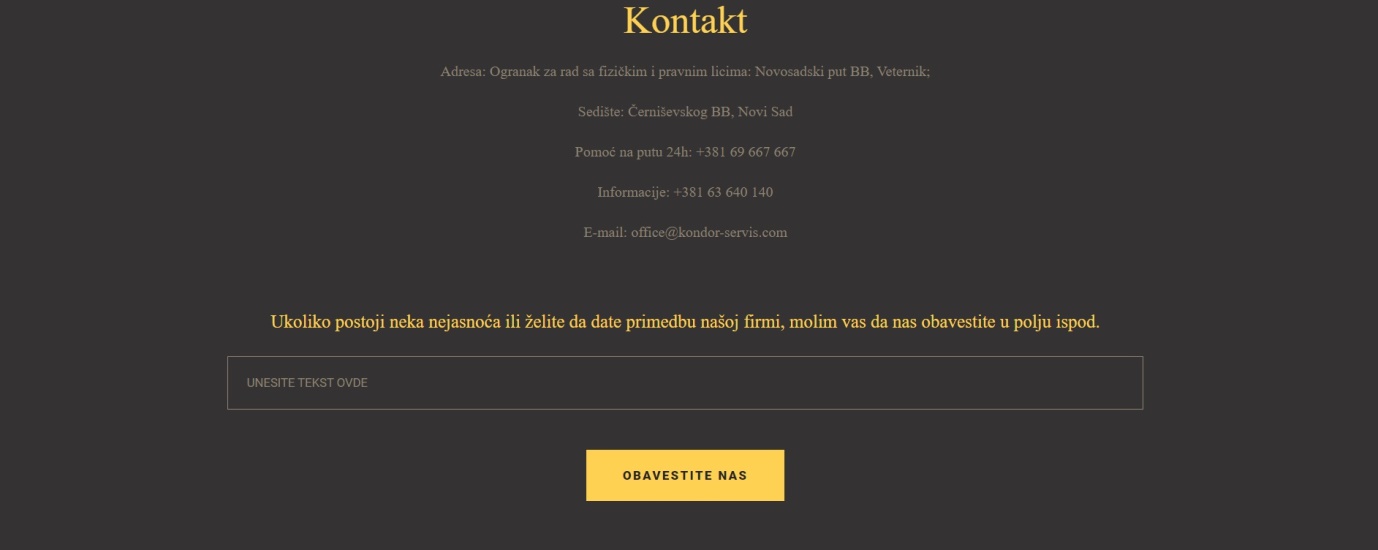
text-transform: uppercase;

font-weight: 400;

}

*CSS kod za kontakt sekciju*

Na kraju kontakt sekcija izgleda ovako:



*Slika 21. Konačan izgled kontakt sekcije.*

# 7. Kreiranje Footera

<footer>

<span>Tehnički fakultet - Mihajlo Pupin</span>

</footer>

*HTML kod footera*

/\* footer \*/

footer {

width: 100%;

margin: 0 auto;

text-align: center;

padding: 1em 0;

background: #282627;

}

footer span {

color: #8D8273;

font-weight: 600;

}

*CSS kod Footera*

Na kraju footer izgleda ovako:



*Slika 22. Footer*

# 8. Logika stranice za registraciju i logovanje

Za početak treba kreirati klasu kontrolera naloga i u njoj definisati metode za formu registracije i čuvanje u bazu i formu za logovanje

public class AccountController : Controller

{

// GET: Account

public ActionResult Index()

{

return View();

}

public ActionResult Register()

{

return View();

}

//Register forma

[HttpPost]

public ActionResult SaveRegisterDetails(Register registerDetails)

{

//Proveravamo da li je stanje modela važeće ili ne. Koristili smo atribute DataAnnotation.

//Ako bilo koja vrednost obrasca ne prođe validaciju DataAnnotation, stanje modela postaje nevažeće.

if (ModelState.IsValid)

{

//kreirati kontekst baze podataka koristeći Entite Framework

using (var databaseContext = new AccountsEntities2())

{

bool userExists = databaseContext.RegisterUsers.FirstOrDefault(x => x.ime == registerDetails.ime) != null;

//Proveravamo da li uneseno ime postoji u bazi

if (userExists)

{

//Ukoliko postoji, korisnik dobija poruku da je ime zauzeto

ModelState.AddModelError("ime", "Ime je zauzeto!");

return View("Register");

}

else

{

//Ukoliko ne postoji, podaci se cuvaju bazi

RegisterUser reglog = new RegisterUser();

reglog.ime = registerDetails.ime;

reglog.lozinka = registerDetails.lozinka;

//Čuvanje podataka

databaseContext.RegisterUsers.Add(reglog);

databaseContext.SaveChanges();

}

}

//Kada se nalog napravi, dobija se poruka i korisnik biva upićen na login

ViewBag.Message = "Nalog je napravljen!";

return View("Login");

}

else

{

//Ako validacija ne uspe, vraćamo objekat modela sa greškama u prikaz, koji će prikazati poruke o grešci.

return View("Register", registerDetails);

}

}

public ActionResult Login()

{

return View();

}

//Login forma

[HttpPost]

public ActionResult Login(LoginViewModel model)

{

//Provera stanja modela

if (ModelState.IsValid)

{

//Validacija korisnika, bez obzira da li je korisnik validan ili ne

var isValidUser = IsValidUser(model);

//Ako je korisnik ispravan i prisutan u bazi podataka, preusmeravamo ga na početnu stranicu

if (isValidUser != null)

{

FormsAuthentication.SetAuthCookie(model.ime, false);

return RedirectToAction("Index","Home");

}

else

{

//Ako korisnik nije prisutan, dobije poruku

ModelState.AddModelError("Failure", "Pogresno ime ili lozinka!");

return View();

}

}

else

{

//Ako stanje modela nije važeće, model sa porukom o grešci se vraća u prikaz

return View(model);

}

}

//Funkcija za proveru da li je korisnik ispravan ili ne

public RegisterUser IsValidUser(LoginViewModel model)

{

using (var dataContext = new AccountsEntities2())

{

//Preuzimanje podataka o korisniku iz baze podataka na osnovu korisničkog imena i lozinke koje je uneo korisnik

RegisterUser user = dataContext.RegisterUsers.Where(query => query.ime.Equals(model.ime) && query.lozinka.Equals(model.lozinka)).SingleOrDefault();

//Ako je korisnik prisutan, onda se vraća true

if (user == null)

return null;

//Ako korisnik nije prisutan, vraća se false

else

return user;

}

}

public ActionResult Logout()

{

FormsAuthentication.SignOut();

Session.Abandon(); //Brisanje sesije na kraju zahteva

return RedirectToAction("Index", "Home");

}

public ActionResult Back()

{

return RedirectToAction("Index", "Home");

}

}

}

*C#(ASP.NET) kod za logiku stranica za registraciju i login*

# 9. Logika komentara

Ovaj deo Backend-a nam služi za prikaz svih komentara na loyalty sekcije koji su uspešno upisani u bazu.

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

//Listanje svih komentara

Baza baza = new Baza();

List<Komentar> sviKomentari = baza.sviKomentari;

return View(sviKomentari);

}

[HttpPost]

public ActionResult Index(string tekstKomentara)

{

//Čuvanje imena, teksta komentara i datum

var ident = System.Web.HttpContext.Current.User.Identity.Name;

Komentar komentar = new Komentar();

komentar.ime = ident.ToString();

komentar.tekstKomentara = tekstKomentara;

komentar.datum = DateTime.Now;

//Upisivanje komentara u bazu

Baza baza = new Baza();

baza.upisiKomentar(komentar);

return RedirectToAction("Index");

}

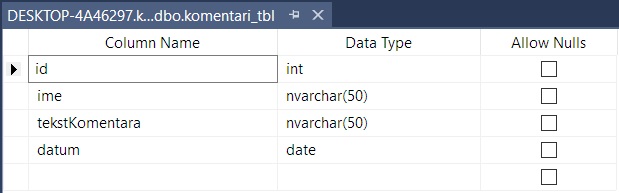
}

}

*C#(ASP.NET) kod za logiku prikaza kometara*

# 10. Kreiranje Baze podataka

Bazu podataka sa potrebnim elementima kreiramo u MySql Menagment Studiu



*Slika 23. Potrebni tipovi podataka u tabeli u bazi podataka*

CREATE DATABASE [Accounts]

GO

USE [Accounts]

GO

/\*\*\*\*\*\* Object: Table [dbo].[komentari\_tbl] Script Date: 9/10/2022 12:25:34 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[komentari\_tbl](

[id] [int] IDENTITY(1,1) NOT NULL,

[ime] [varchar](100) NOT NULL,

[tekstKomentara] [varchar](500) NOT NULL,

[datum] [datetime] NOT NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[RegisterUser] Script Date: 9/10/2022 12:25:34 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[RegisterUser](

[Id] [int] IDENTITY(1,1) NOT NULL,

[ime] [varchar](100) NOT NULL,

[lozinka] [varchar](500) NOT NULL,

CONSTRAINT [PK\_RegisterUser] PRIMARY KEY CLUSTERED

(

[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

*SQL kod za kreiranje potrebne baze podataka*

# 11. Povezivanje sa Bazom podataka

Da bi mogli da iskoristimo bazu koju smo kreirali moramo da kreiramo klasu koja će da poveže bazu sa aplikacijom.

public class Baza

{

//Izlistavanje svih komentara

public List<Komentar> sviKomentari

{

get

{

List<Komentar> komentari = new List<Komentar>();

//Konekcija sa bazom

string cs = "data source=DESKTOP-4A46297; database = Accounts; integrated security=SSPI";

using (SqlConnection con = new SqlConnection(cs))

{

con.Open();

SqlCommand cmd = new SqlCommand("select \* from komentari\_tbl", con);

using (SqlDataReader rdr = cmd.ExecuteReader())

{

while (rdr.Read())

{

Komentar f = new Komentar();

f.id = Convert.ToInt32(rdr["id"]);

f.ime = rdr["ime"].ToString();

f.tekstKomentara = rdr["tekstKomentara"].ToString();

f.datum = (DateTime)rdr["datum"];

komentari.Add(f);

}

}

}

return komentari;

}

}

//Upisivanje komentara u bazu

public void upisiKomentar(Komentar komentar)

{

string cs = "data source=DESKTOP-4A46297; database = Accounts; integrated security=SSPI";

using (SqlConnection con = new SqlConnection(cs))

{

con.Open();

string stringKomentara = "Insert into komentari\_tbl values('"+ komentar.ime + "','" + komentar.tekstKomentara + "','" + komentar.datum + "')";

SqlCommand cmd = new SqlCommand(stringKomentara, con);

cmd.ExecuteNonQuery();

}

}

}

}

*C#(ASP.NET) kod za povezivanje sa bazom*

**NAPOMENA:** Da bi websajt pravilno funkcionisao na ostalim računarima potrebno je promeniti naziv lokalnog servera koji je unikatan za svaki klijent, naznačen je crveno u prethodnom kodu tj. fajlu Baza.cs kao i fajl-u Web.config, naravno sa prethodno kreiranom bazom sa nazivom “Accounts”, tabelom “komentari\_tbl”:

[id] [int] IDENTITY(1,1) NOT NULL,

[ime] [varchar](100) NOT NULL,

[tekstKomentara] [varchar](500) NOT NULL,

[datum] [datetime] NOT NULL () i podacima:

i tabelom “RegisterUser”

[Id] [int] IDENTITY(1,1) NOT NULL,

[ime] [varchar](100) NOT NULL,

[lozinka] [varchar](500) NOT NULL,

CONSTRAINT [PK\_RegisterUser] PRIMARY KEY CLUSTERED

# 12. Zaključak

Korišćenjem tehnologija HTML,CSS, C#(ASP.NET), MySQL i GitHub i implementacijom svih gore navedenih elemenata, dobijamo websajt koji je modernog izgleda, ostvaren tako da je pogodan za desktop, tablet i mobilne uređaje.

# 13. Literatura

1. Jon Duckett, “HTML & CSS: Design and Build Web Sites”, 2020.
2. Lea Verou, “CSS Secrets: Better Solutions to Everyday Web Design Problems”, 2019.
3. Jon Skeet, “C# in Depth”, 2019.
4. Sergey Kuzmichev and Vinicius M. Grippa, “Learning MySQL”, 2020.
5. Phil Haack and Sarah Guthals, “GitHub For Dummies”, 2019.