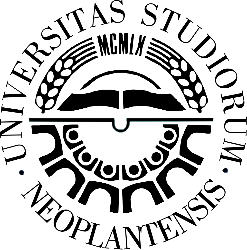
**Univerzitet u Novom Sadu   
 Tehnički fakultet ”Mihajlo Pupin”   
Zrenjanin**

**SEMINARSKI RAD**

**Predmet:** Razvoj softvera otvorenog koda

**Tema:** Poslastičarnica (Web aplikacija za recepte)

Predmetni profesor: Studenti:

Doc. dr. Zoltan Kazi Milica Babić SI 3/16

Miloš Petrović SI 39/17

Zrenjanin, 2020. godina.

Sadržaj:

[1. UVOD 4](#_Toc41993142)

[2. SPECIFIKACIJA ZAHTEVA KORISNIKA 5](#_Toc41993143)

[3. FAZE RAZVOJA SOFTVERA 6](#_Toc41993144)

[4. PRIKAZ SOFTVERA 7](#_Toc41993145)

[4.1 POKRETANJA SOFTVERA I EKRANSKI PRIKAZ 7](#_Toc41993146)

[5. PRIKAZ REALIZACIJE I IMPLEMENTACIJA 12](#_Toc41993147)

[5.1 BAZA PODATAKA (SQL) 12](#_Toc41993148)

[5.2 PRIKAZ IZVORNOG KODA APLIKACIJE 17](#_Toc41993149)

[5.2.1 PREZENTACIONI SLOJ 17](#_Toc41993150)

[5.2.2 SLOJ PODATAKA 32](#_Toc41993151)

[5.3 TESTIRANJE APLIKACIJE 40](#_Toc41993152)

[6. OPIS PRIMENJENIH TEHNOLOGIJA 41](#_Toc41993153)

[6.1 TABELARNI PRIKAZ SLOJEVA I PODSLOJEVA APLIKACIJE 41](#_Toc41993154)

[6.2 GITHUB 41](#_Toc41993155)

[O GITHUB-U 41](#_Toc41993156)

[7. LITERATURA: 46](#_Toc41993157)

# UVOD

Seminarski rad iz predmeta Razvoj softvera otvorenog koda obuhvata izradu web aplikacije za poslasticarnicu (odnosno, aplikacija na kojoj će biti prikazana receptura istih).

Poslastičarnica je Web aplikacija preko koje posetilac može da se registruje, odnosno može da kreira svoj nalog, I svojom registracijom, odnosno prijavom omogućava sebi da preglede tabelu recepata takođe posetilac može da uđe i na ostale stranice kao što su: „Početna“, „Poslastice“, „Recepti“, „O nama“ i „Kontakt“.

Posetilac ima mogućnost da doda neki od recepata.

Posetilac može da se registruje kao Administrator ili kao običan posetilac. Ukoliko je korisnik administrator on ima mogućnost da doda i određenu poslasticu, tako što se nakon njegove prijave na stranici poslastice pojavi dugme „Kreiraj novu poslasticu“.

Ukoliko je posetilac običan korisnik on to dugme ne može da vidi jer nema tu mogućnost. Običan posetilac ima mogućnost da doda recept, što naravno može da uradi i administrator.

U listingu 1 je prikazan način prikrivanja dugmeta „Kreiraj novu poslasticu“.

if (!IsPostBack)

{

Baza baza = new Baza();

List<Poslastice> lista = baza.DajSvePoslastice();

gvTabelaPoslastica.DataSource = lista;

gvTabelaPoslastica.DataBind();

if (!string.IsNullOrWhiteSpace(User.Identity.Name))

{

Posetilac posetilac = baza.DajPosetiocaPremaKorisnickomImenu(User.Identity.Name);

btKreiraj.Visible = posetilac.PosetilacJeAdministrator;

}

else

{

btKreiraj.Visible = false;

}

}

*Listing 1: Prikaz prikrivanja dugmeta „Kreiraj novu poslasticu“.*

# SPECIFIKACIJA ZAHTEVA KORISNIKA

Profil administratora:

* Pristup svakoj stranici („Početna“, „Poslastice“, „Recepti“, „O nama“, „Kontakt“ i „Registracija“);
* Pregledanje tabele poslastica;
* Kreiranje poslastice;
* Pretraga poslastice po nazivu;
* Pregledanje tabele recepata;
* Dodavanje recepta;

Profil prijavljenog posetioca:

* Pristup svakoj stranici („Početna“, „Poslastice“, „Recepti“, „O nama“, „Kontakt“ i „Registracija“);
* Pregledanje tabele poslastica;
* Pretraga poslastice po nazivu;
* Pregledanje tabele recepata;
* Dodavanje recepta;

Profil neprijavljenog posetioca:

* Kreiranje naloga;
* Pristup svakoj stranici („Početna“, „Poslastice“, „Recepti“, „O nama“, „Kontakt“ i „Registracija“);
* Pregledanje tabele poslastica;
* Pretraga poslastice po nazivu;
* Pregledanje tabele recepata;

# FAZE RAZVOJA SOFTVERA

* Analiza i definisanje zahteva;
* Kreiranje repozitorijuma na GitHab-u;
* Kreiranje baze podataka (SQL Server);
* Postavka projekta:
  + Front-end – uspostavljanje veze sa bazom,
  + Back-end – kreiranje projekta
* Kreiranje glavne stranice (Master page – GlavnaStranica.master);
* Kreiranje ostalih stranica;
* Front-end (dizajn aplikacije i samih stranica);
* Back-and ( kreiranje fajla Baza, u kojem projekat povezujemo sa bazom podataka);
* Implementacija
* Testiranje softvera;
* Izrada aplikacije

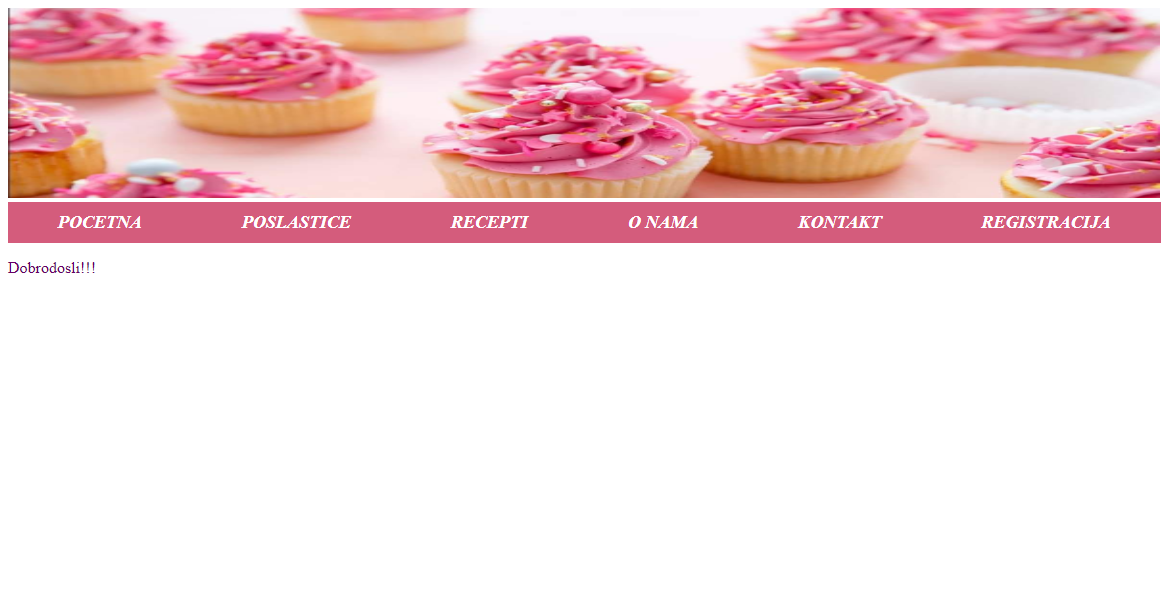
# PRIKAZ SOFTVERA

## 4.1 POKRETANJA SOFTVERA I EKRANSKI PRIKAZ

Aplikacija se pokreće u Visual Studio-u, neophodno je pokrenuti celokupan projekat, takođe potrebno je pokrenuti Microsoft SQL Server Managment Studio da bi aplikacija uspostavila vezu sa bazom.

Nakon toga, pokreće se aplikacija.

Na slici 1 je prikazan izgled sajta za neimenovanog korisnika.



*Slika 1: Prikaz početnog ekrana.*

Na slici 2 se može videti izgleda stranice kad se pritisne na „Registracija“. Gde ako posetilac nema nalog prvo mora da kreira nalog, odnosno da se registriju, i to izgleda ovako:



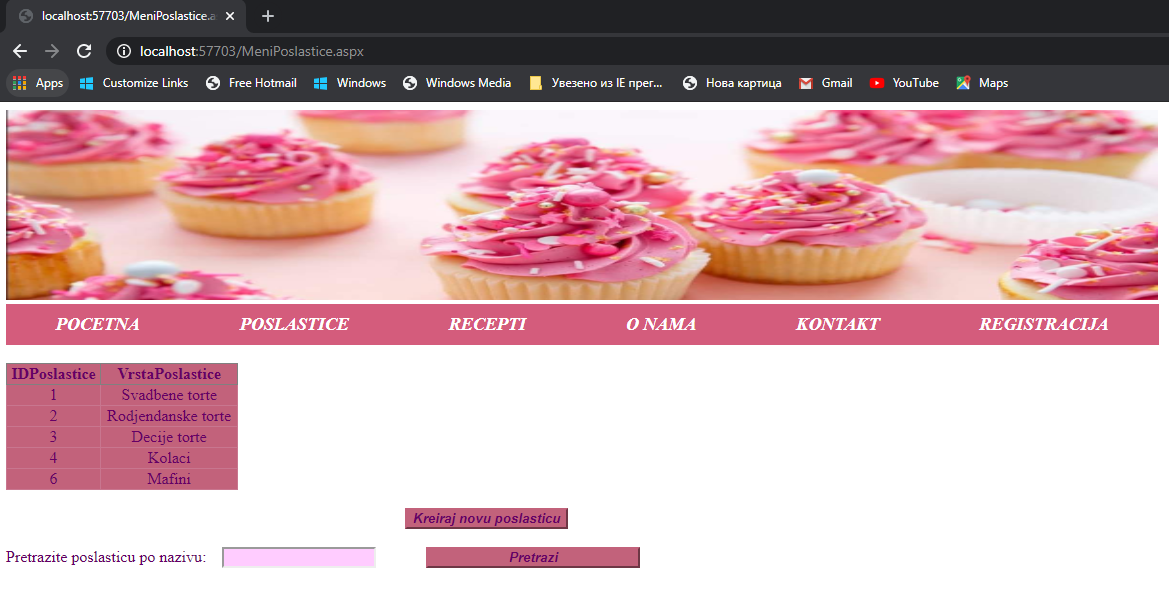
*Slika 2: Izgled stranice registracija.*

Ukoliko posetilac već ima svoj nalog, on je u mogućnosti samo da ide na dugme „Prijavi se“, i kad se klikne na dugme, sledeća stranica izgleda ovako:



*Slika 3: Izgled kad stranice Prijava.*

Ukoliko je korisnik ulogovan kao admin stranica poslastice ima sledeći izgled:



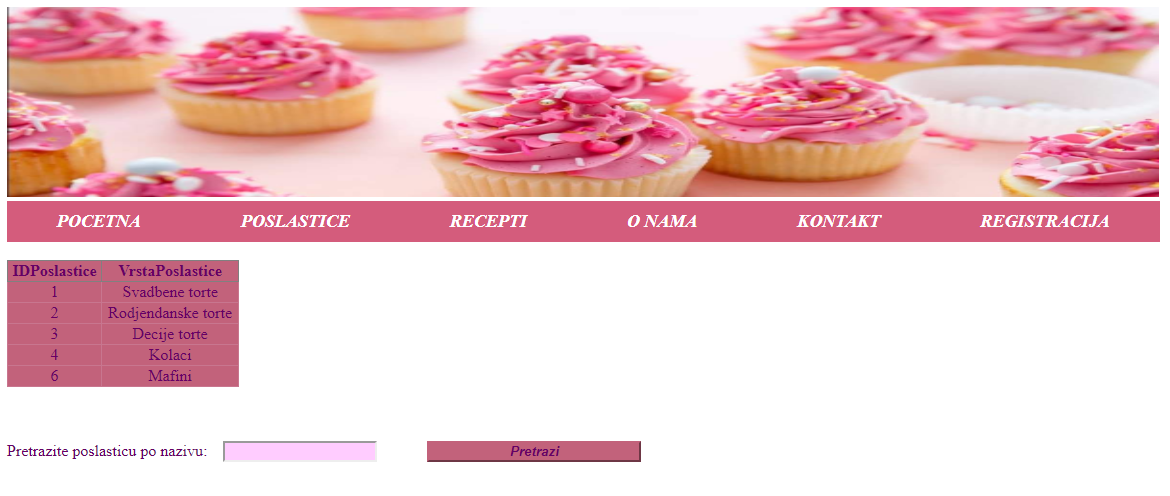
*Slika 4: Izgled stranice „Poslastice“ ako je posetilac ulogovan kao administrator.*

Kada se klikne na dugme „Kreiraj novu poslasticu“ izbaciće novu stranicu na kojoj se kreira nova stranica i to izgleda ovako:



*Slika 5: Izgled stranice na kojoj se kreira poslastica.*

Ukoliko je korisnik kreiran kao običan posetilac njemu će stranica „Poslastice“ izgledati ovako:



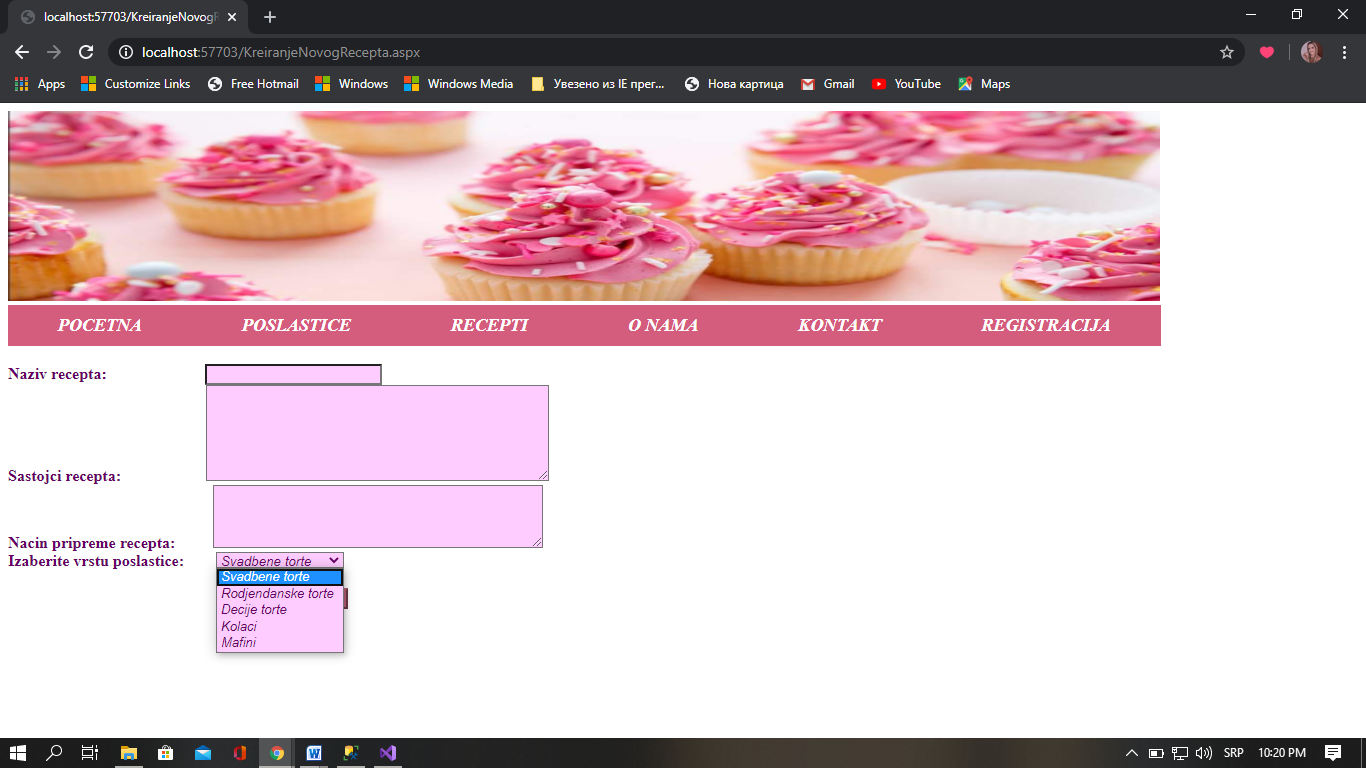
*Slika 6: Izgled stranice „Poslastice“ ako je posetilac registrovan kao običan posetilac.*

Kada se uđe na stranicu „Recepti“ bez obzira da li je u pitanju korisnik ili administor, ona ima sledeći izgled:



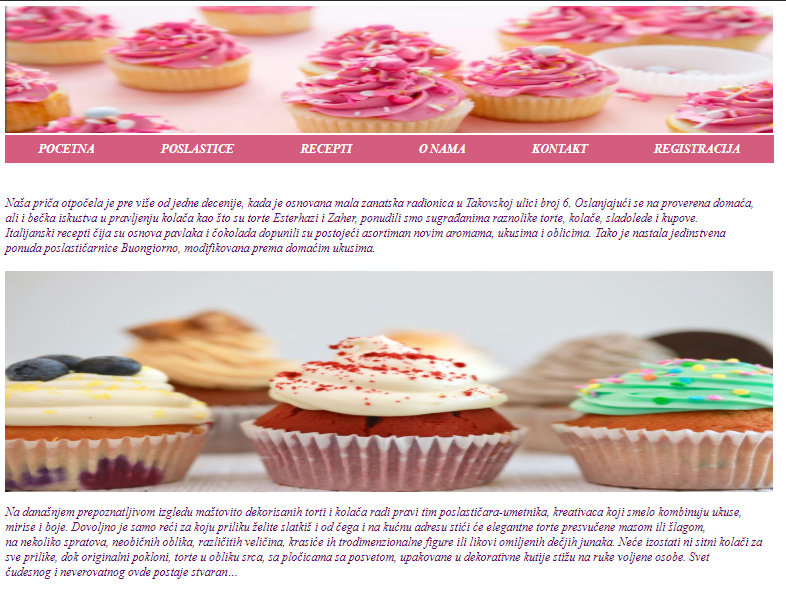
*Slika 7: Izgled stranice „Recepti“.*

Na slici 8. Se može videti kako izgleda stranica za kreiranje recepta:



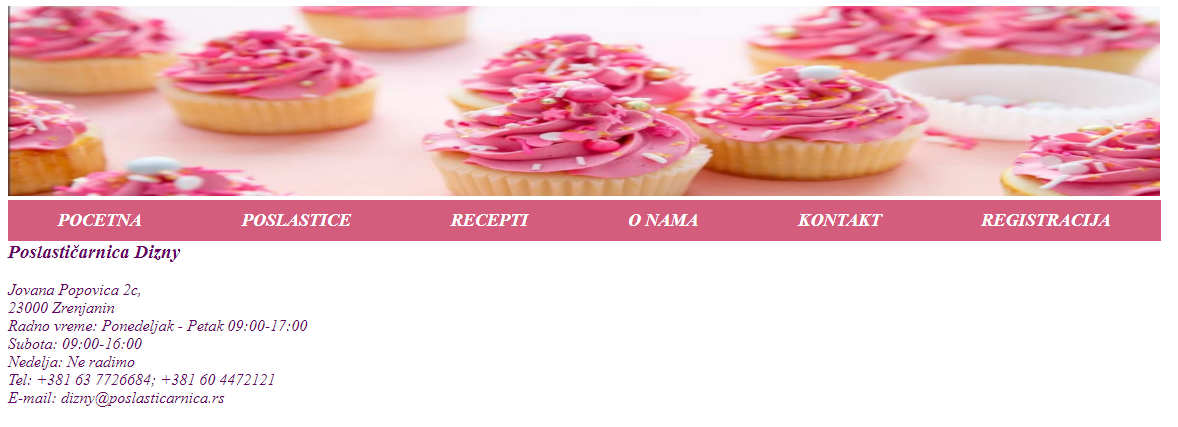
Slika 8:Stranica za kreiranje recepta.

Na sledećoj slici je prikazan izgled stranice „O nama“:



*Slika 9: Izgled strancie „O nama“.*

Na sledećoj slici se može videti izgled stranice „Kontakt“:



*Slika 10: Izgled stranice „Kontakt“.*

# PRIKAZ REALIZACIJE I IMPLEMENTACIJA

## BAZA PODATAKA (SQL)

USE [master]

GO

/\*\*\*\*\*\* Object: Database [Poslasticarnica] Script Date: 6/2/2020 2:53:05 PM \*\*\*\*\*\*/

CREATE DATABASE [Poslasticarnica]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Poslasticarnica', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Poslasticarnica.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'Poslasticarnica\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Poslasticarnica\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

GO

ALTER DATABASE [Poslasticarnica] SET COMPATIBILITY\_LEVEL = 150

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [Poslasticarnica].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [Poslasticarnica] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [Poslasticarnica] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [Poslasticarnica] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [Poslasticarnica] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [Poslasticarnica] SET ARITHABORT OFF

GO

ALTER DATABASE [Poslasticarnica] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [Poslasticarnica] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [Poslasticarnica] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [Poslasticarnica] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [Poslasticarnica] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [Poslasticarnica] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [Poslasticarnica] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [Poslasticarnica] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [Poslasticarnica] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [Poslasticarnica] SET DISABLE\_BROKER

GO

ALTER DATABASE [Poslasticarnica] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [Poslasticarnica] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [Poslasticarnica] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [Poslasticarnica] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [Poslasticarnica] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [Poslasticarnica] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [Poslasticarnica] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [Poslasticarnica] SET RECOVERY SIMPLE

GO

ALTER DATABASE [Poslasticarnica] SET MULTI\_USER

GO

ALTER DATABASE [Poslasticarnica] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [Poslasticarnica] SET DB\_CHAINING OFF

GO

ALTER DATABASE [Poslasticarnica] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [Poslasticarnica] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [Poslasticarnica] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [Poslasticarnica] SET QUERY\_STORE = OFF

GO

USE [Poslasticarnica]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Posetilac] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Posetilac](

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

[Email] [nvarchar](50) NOT NULL,

[Lozinka] [nvarchar](50) NOT NULL,

[KorisnickoIme] [nvarchar](50) NOT NULL,

[PosetilacJeAdministrator] [bit] NOT NULL,

CONSTRAINT [PK\_Posetilac] PRIMARY KEY CLUSTERED

(

[IDPosetioca] 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

/\*\*\*\*\*\* Object: Table [dbo].[Poslastice] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Poslastice](

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

[VrstaPoslastice] [nvarchar](50) NOT NULL,

PRIMARY KEY CLUSTERED

(

[IDPoslastice] 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

/\*\*\*\*\*\* Object: Table [dbo].[Recepti] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Recepti](

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

[NazivRecepta] [nvarchar](50) NOT NULL,

[NacinPripremeRecepta] [nvarchar](max) NOT NULL,

[SastojciRecepta] [nvarchar](max) NOT NULL,

[IDPoslastice] [int] NOT NULL,

[IDPosetioca] [int] NOT NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[Recepti] WITH CHECK ADD FOREIGN KEY([IDPosetioca])

REFERENCES [dbo].[Posetilac] ([IDPosetioca])

GO

ALTER TABLE [dbo].[Recepti] WITH CHECK ADD FOREIGN KEY([IDPoslastice])

REFERENCES [dbo].[Poslastice] ([IDPoslastice])

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DajPosetiocaPremaKorisnickomImenu] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure [dbo].[DajPosetiocaPremaKorisnickomImenu]

@KorisnickoIme nvarchar(50)

as

begin

select \* from Posetilac where KorisnickoIme = @KorisnickoIme

end

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DajSvePosetioce] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure [dbo].[DajSvePosetioce]

as

begin

select \* from Posetilac

end

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DajSvePoslastice] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[DajSvePoslastice]

AS

BEGIN

SELECT \* FROM Poslastice

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DajSvePoslasticePremaVrsti] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[DajSvePoslasticePremaVrsti]

(

@VrstaPoslastice nvarchar(50)

)

AS

select \* from Poslastice where VrstaPoslastice=@VrstaPoslastice

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DajSveRecepte] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure [dbo].[DajSveRecepte]

AS

BEGIN

select \*

from Recepti

inner join Poslastice on Poslastice.IDPoslastice = Recepti.IDPoslastice

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DodajNoviRecept] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[DodajNoviRecept]

(

@NazivRecepta nvarchar(50),

@SastojciRecepta nvarchar(50),

@NacinPripremeRecepta nvarchar(max),

@IDPoslastice int,

@IDPosetioca int

)

AS

BEGIN

Insert into Recepti(NazivRecepta, SastojciRecepta, NacinPripremeRecepta, IDPoslastice, IDPosetioca) values (@NazivRecepta, @SastojciRecepta, @NacinPripremeRecepta, @IDPoslastice, @IDPosetioca)

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DodajNovogPosetioca] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[DodajNovogPosetioca]

@Email nvarchar(50),

@Lozinka nvarchar(50),

@KorisnickoIme nvarchar(50),

@PosetilacJeAdministrator bit

AS

BEGIN

INSERT INTO [dbo].[Posetilac]

([Email]

,[Lozinka]

,[KorisnickoIme]

,[PosetilacJeAdministrator])

VALUES

(@Email

,@Lozinka

,@KorisnickoIme

,@PosetilacJeAdministrator)

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DodajNovuPoslasticu] Script Date: 6/2/2020 2:53:06 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[DodajNovuPoslasticu]

(@VrstaPoslastice nvarchar (50))

AS

BEGIN

insert into Poslastice (VrstaPoslastice) values (@VrstaPoslastice)

end

GO

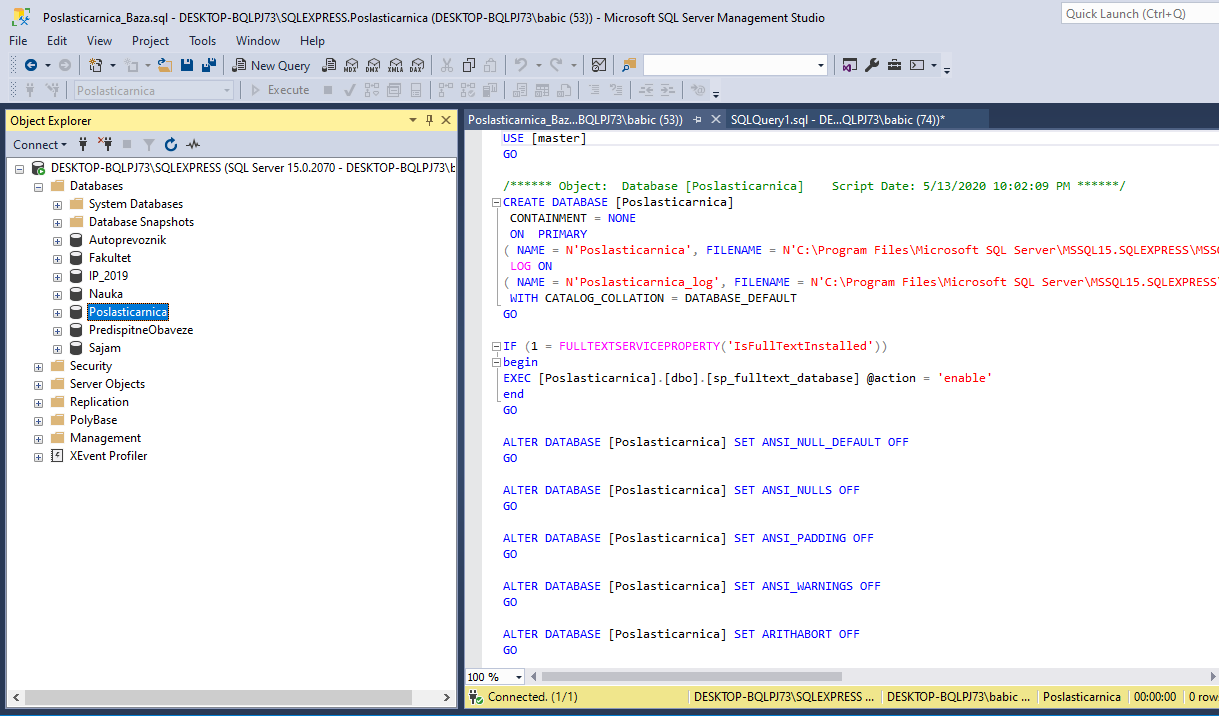
USE [master]

GO

ALTER DATABASE [Poslasticarnica] SET READ\_WRITE

GO

*Listing 2: Prikaz SQL baze podataka.*



*Slika 11: Kreirana baza „Poslastičarnica“ u SQL Server Managment Studio-u.*

## 5.2 PRIKAZ IZVORNOG KODA APLIKACIJE

### 5.2.1 PREZENTACIONI SLOJ

Softver ima Prezentacioni sloj koji ima glavnu stranicu (Master page), koji predstavlja dizajn za stranice (front-end) i ona ima sl. izgled:

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="GlavnaStranica.master.cs" Inherits="PrezentacioniSloj.GlavnaStranica" %>

<!DOCTYPE html>

<html>

<head runat="server">

<title></title>

<%-- <asp:ContentPlaceHolder ID="head" runat="server">

</asp:ContentPlaceHolder>--%>

<style type="text/css">

.auto-style1 {

width: 1152px;

height: 190px;

margin-right: 0px;

}

</style>

</head>

<body>

<form id="form2" runat="server">

<div>

<img alt="" class="auto-style1" src="Slike/slika1.jpg"/>

<br />

<asp:Menu ID="MeniPoslasticarnice" runat="server" Orientation="Horizontal" OnMenuItemClick="MeniPoslasticarnice\_Click" >

<Items>

<asp:MenuItem Text="POCETNA" Value="~/MeniPocetna.aspx"></asp:MenuItem>

<asp:MenuItem Text="POSLASTICE" Value="~/MeniPoslastice.aspx"></asp:MenuItem>

<asp:MenuItem Text="RECEPTI" Value="~/MeniRecepti.aspx"></asp:MenuItem>

<asp:MenuItem Text="O NAMA" Value="~/MeniONama.aspx"></asp:MenuItem>

<asp:MenuItem Text="KONTAKT" Value="~/MeniKontakt.aspx"></asp:MenuItem>

<asp:MenuItem Text="REGISTRACIJA" Value="Registracija.aspx"></asp:MenuItem>

</Items>

<StaticHoverStyle BackColor="#660066" ForeColor="#3366FF" />

<StaticMenuItemStyle BackColor="#D45C7C" Font-Bold="True" Font-Italic="True" Font-Size="Large" Font-Strikeout="False" ForeColor="White" HorizontalPadding="50px" VerticalPadding="10px" />

</asp:Menu>

</div>

<div>

<asp:ContentPlaceHolder ID="Stranica" runat="server">

</asp:ContentPlaceHolder>

</div>

</form>

</body>

</html>

*Listing 3: Glavna stranica.*

Kod za stranicu „GlavnaStranica“ :

* GlavnaStranica : System.Web.UI.MasterPage – glavna stranica je Master stranica.
* MeniPoslasticarnice\_Click – predstavlja aplikacije koji ima 6 stavki.
* Response.Redirect – predstavlja metodu koja očitava stranice.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class GlavnaStranica : System.Web.UI.MasterPage

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void MeniPoslasticarnice\_Click(object sender, MenuEventArgs e)

{

string stranica = e.Item.Value;

Response.Redirect(stranica);

}

}

}

*Listing 4:Prikaz koda za Master page.*

Stranica Prijava.aspx – front-end :

* Svaka stranica ima tipičnu html formu (html, head, body).
* U okviru head-a se nalazi stil stranica.
* U okviru form-a se nalaze:
  + slike (ImageUrl) koje predstavljaju ikonice za data polja (icons8-login-100.png, icons8-password-100.png, icons8-username-100.png), njima je određena visina i širina;
  + Dugme (btnPrijava) – koje je obojeno i unutar sebe ima određeni tekst.

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Prijava.aspx.cs" Inherits="PrezentacioniSloj.Prijava" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<style type="text/css">

.auto-style1 {

width: 299px;

height: 403px;

}

.auto-style2 {

margin-left: 118px;

}

.auto-style5 {

margin-left: 15px;

}

.auto-style6 {

margin-left: 21px;

margin-top: 0px;

}

.auto-style7 {

margin-left: 18px;

}

.auto-style8 {

margin-left: 20px;

}

.auto-style9 {

margin-left: 16px;

}

</style>

</head>

<body>

<form id="form2" runat="server">

<div class="auto-style1" style="background-image: none; background-color: #FF99CC;">

<br />

<br />

<asp:Image ID="Image1" runat="server" CssClass="auto-style2" Height="47px" ImageUrl="~/Slike/icons8-login-100.png" Width="69px" />

<br />

<br />

<br />

<asp:Image ID="Image3" runat="server" CssClass="auto-style5" Height="32px" ImageUrl="~/Slike/icons8-password-100.png" Width="35px" />

<asp:TextBox ID="tbLozinka" runat="server" BackColor="#FF99CC" BorderColor="#FF99CC" BorderStyle="Ridge" CssClass="auto-style6" Font-Italic="True" ForeColor="#660066" Height="20px" Width="151px" TextMode="Password">lozinka</asp:TextBox>

<br />

<br />

<asp:Image ID="Image4" runat="server" CssClass="auto-style7" Height="31px" ImageUrl="~/Slike/icons8-username-100.png" Width="34px" />

<asp:TextBox ID="tbKorisnickoIme" runat="server" BackColor="#FF99CC" BorderColor="#FF99CC" BorderStyle="Ridge" CssClass="auto-style8" Font-Italic="True" ForeColor="#660066" Height="19px" Width="150px">korisnicko ime</asp:TextBox>

<br />

<br />

<br />

<br />

<br />

<asp:Button ID="btnPrijava" runat="server" BackColor="Black" BorderColor="Black" CssClass="auto-style9" Font-Bold="True" Font-Italic="True" ForeColor="White" Height="33px" Text="Prijavi se" Width="264px" OnClick="btnPrijava\_Click" />

</div>

</form>

</body>

</html>

*Listing 5:Front-end za stranicu Prijava.aspx.*

Stranica Prijava.aspx pravila :

* Povezana je sa slojom podataka, zbog baze podataka. (using SlojPodataka);
* Povezana je sa folderom modeli jer se u njima konstruktori koji omogućava čitanje podataka (using SlojPodataka.Modeli);
* btnPrijava\_Click() – predtavlja metodu u okviru koje se realizuje prijava korisnika.
* Posetilac posetilac = baza.DajPosetiocaPremaKorisnickomImenu – preko konstruktora se pristupa bazi I omogućava se prijava korisnika, kada unese svoje podatke.
* Response.Redirect– kada se korisnik prijavi odlazi na stranicu koja je navedena u zagradi.
* SetAuthCookie() – metoda koja provera autentifikaciju za korisnicko ime

using SlojPodataka;

using SlojPodataka.Modeli;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class Prijava : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnPrijava\_Click(object sender, EventArgs e)

{

Baza baza = new Baza();

Posetilac posetilac = baza.DajPosetiocaPremaKorisnickomImenu(tbKorisnickoIme.Text);

if (posetilac.Lozinka == tbLozinka.Text)

{

FormsAuthentication.SetAuthCookie(posetilac.KorisnickoIme, false);

}

Response.Redirect("~/MeniPocetna.aspx");

}

}

}

*Listing 6: „MeniPocetna.aspx“.*

Registracija.aspx – front-end :

* 4 slike (ImageUrl)
* 3 textBox-a – tbEmail, tbLozinka, tbKorisnickoIme – koriste se za unos teksta.
* 2 dugmeta (btnRegistrujSe, btnPrijaviSe) – koji su obojeni sa backgrond-color, border je crne boje, font-podebljan, font-iskošen, boja teksta je bela.

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Registracija.aspx.cs" Inherits="PrezentacioniSloj.Registracija" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<style type="text/css">

.auto-style1 {

width: 299px;

height: 403px;

}

.auto-style2 {

margin-left: 118px;

}

.auto-style3 {

margin-left: 14px;

}

.auto-style4 {

margin-left: 21px;

}

.auto-style5 {

margin-left: 15px;

}

.auto-style6 {

margin-left: 21px;

margin-top: 0px;

}

.auto-style7 {

margin-left: 18px;

}

.auto-style8 {

margin-left: 20px;

}

.auto-style9 {

margin-left: 16px;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div class="auto-style1" style="background-image: none; background-color: #FF99CC;">

<br />

<br />

<asp:Image ID="Image1" runat="server" CssClass="auto-style2" Height="47px" ImageUrl="~/Slike/icons8-login-100.png" Width="69px" />

<br />

<br />

<asp:Image ID="Image2" runat="server" CssClass="auto-style3" Height="33px" ImageUrl="~/Slike/icons8-email-open-100.png" Width="36px" />

<asp:TextBox ID="tbEmail" runat="server" BackColor="#FF99CC" BorderColor="#FF99CC" BorderStyle="Ridge" CssClass="auto-style4" Font-Italic="True" ForeColor="#660066" Height="21px" Width="152px">e-mail</asp:TextBox>

<br />

<br />

<asp:Image ID="Image3" runat="server" CssClass="auto-style5" Height="32px" ImageUrl="~/Slike/icons8-password-100.png" Width="35px" />

<asp:TextBox ID="tbLozinka" runat="server" BackColor="#FF99CC" BorderColor="#FF99CC" BorderStyle="Ridge" CssClass="auto-style6" Font-Italic="True" ForeColor="#660066" Height="20px" Width="151px" TextMode="Password">lozinka</asp:TextBox>

<br />

<br />

<asp:Image ID="Image4" runat="server" CssClass="auto-style7" Height="31px" ImageUrl="~/Slike/icons8-username-100.png" Width="34px" />

<asp:TextBox ID="tbKorisnickoIme" runat="server" BackColor="#FF99CC" BorderColor="#FF99CC" BorderStyle="Ridge" CssClass="auto-style8" Font-Italic="True" ForeColor="#660066" Height="19px" Width="150px">korisnicko ime</asp:TextBox>

<br />

<br />

<br />

<asp:Button ID="btnRegistrujSe" runat="server" BackColor="Black" BorderColor="Black" CssClass="auto-style5" Font-Bold="True" Font-Italic="True" ForeColor="White" Height="33px" Text="Registruj se" Width="264px" OnClick="btnRegistrujSe\_Click"/>

<br />

<br />

<asp:Button ID="btnPrijaviSe" runat="server" BackColor="Black" BorderColor="Black" CssClass="auto-style9" Font-Bold="True" Font-Italic="True" ForeColor="White" Height="33px" OnClick="btnPrijaviSe\_Click" Text="Prijavi se" Width="264px" />

</div>

</form>

</body>

</html>

*Listing 7:Front-end za stranicu Registracija.aspx.*

Stranica Registracija.aspx pravila :

* Ukoliko korisnik nema nalog mora da se registruje, prilikom registracije moraju se uneti sl. podaci : e-mail („tbEmail.Text“), lozinku („tbLozinka.Text“), korisničko ime („tbKorisnickoIme.Text“);
* Prilikom klika na dugme „Registruj se“, podaci se čuvaju u bazi podataka, i korisnik ide na stranicu „Prijava“ i prijavljuje se.
* Povezana je sa slojom podataka, zbog baze podataka. (using SlojPodataka);
* Povezana je sa folderom modeli jer se u njima konstruktori koji omogućava čitanje podataka (using SlojPodataka.Modeli);
* btnPrijaviSe\_Click() – metoda za dugme prijavi se, gde pomoću Response.Redirect() odlazi na svoju stranicu.
* btnRegistrujSe\_Click() – metoda za dugme za registraciju, gde se preko konstruktora pristupa bazi I upisuje se posetilac u bazu pomoću store procedure DodajNovogPosetioca, zatim uz pomoć Response.Redirect() odlazi na stranicu prijave.

using SlojPodataka;

using SlojPodataka.Modeli;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class Registracija : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnPrijaviSe\_Click(object sender, EventArgs e)

{

Response.Redirect("~/Prijava.aspx");

}

protected void btnRegistrujSe\_Click(object sender, EventArgs e)

{

Posetilac posetilac = new Posetilac(tbEmail.Text, tbLozinka.Text, tbKorisnickoIme.Text);

Baza baza = new Baza();

baza.DodajNovogPosetioca(posetilac);

Response.Redirect("~/Prijava.aspx");

}

}

}

*Listing 8: „Registracija.aspx“.*

Stranica „MeniPocetna.aspx“ – front-end :

* ima labelu (Label1) – koja u sebi ima tekst.

<%@ Page Title="" Language="C#" MasterPageFile="~/GlavnaStranica.Master" AutoEventWireup="true" CodeBehind="MeniPocetna.aspx.cs" Inherits="PrezentacioniSloj.MeniPocetna" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Stranica" runat="server">

<p>

<asp:Label ID="Label1" runat="server" ForeColor="#660066" Text="Dobrodosli!!!"></asp:Label>

</p>

</asp:Content>

*Listing 9: „*MeniPocetna*.aspx“ – front-end.*

Što se tiče koda nema nikakvih specijalnih uslova.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class MeniPocetna : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

}

}

*Listing 10: „*MeniPocetna*.aspx“.*

Stranica „MeniPoslastice.aspx“ – front-end :

* sadrži tabelu (GridView-gvTabelaPoslastica) –koja prikazuje poslastice koje su unete u bazu podataka, koja je obojena sa BackColor, tekst unutar tabele je takođe obojen I definisan kako će izgledati, u listingu se može videti kako to izgleda.
* Sadrži 2 dugmeta (btKreiraj, btPretrazi) – koji su obojeni I pozicionirani, kao I tekst u njima.
* Sadrži labelu (lbPretrazi) – čiji je tekst obojen sa ForeColor;
* Sadrži textbox(tbPretraga) – čija je širina određena sa width, I obojen sa BackColor, a tekst je obojen sa ForeColor.

<%@ Page Title="" Language="C#" MasterPageFile="~/GlavnaStranica.Master" AutoEventWireup="true" CodeBehind="MeniPoslastice.aspx.cs" Inherits="PrezentacioniSloj.MeniPoslastice" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Stranica" runat="server">

<br />

<asp:GridView ID="gvTabelaPoslastica" runat="server" Width="232px" BackColor="#C2627B" ForeColor="#660066">

<AlternatingRowStyle BackColor="#C2627B" BorderColor="#C8768F" ForeColor="#660066" HorizontalAlign="Center" />

<EditRowStyle BackColor="#C2627B" BorderColor="#C8768F" ForeColor="White" HorizontalAlign="Center" />

<FooterStyle BackColor="#993399" ForeColor="White" HorizontalAlign="Center" />

<HeaderStyle ForeColor="#660066" Wrap="False" />

<RowStyle BackColor="#C2627B" BorderColor="#C8768F" ForeColor="#660066" HorizontalAlign="Center" />

</asp:GridView>

<br />

<asp:Button ID="btKreiraj" runat="server" Text="Kreiraj novu poslasticu" BackColor="#C2627B" BorderColor="#C2627B" Font-Bold="True" Font-Italic="True" ForeColor="#660066" OnClick="btKreiraj\_Click" style="margin-left: 399px" />

<br />

<br />

<asp:Label ID="lbPretrazi" runat="server" Text="Pretrazite poslasticu po nazivu:" ForeColor="#660066"></asp:Label>

<asp:TextBox ID="tbPretraga" runat="server" style="margin-left: 12px" Width="150px" BackColor="#FFCCFF" ForeColor="#660066"></asp:TextBox>

<asp:Button ID="btPretrazi" runat="server" BackColor="#C2627B" BorderColor="#C2627B" OnClick="btPretrazi\_Click" style="margin-left: 46px" Text="Pretrazi" Width="214px" Font-Bold="True" Font-Italic="True" ForeColor="#660066" />

<br />

<br />

</asp:Content>

*Listing 11: „*MeniPoslastice*.aspx“ – front-end.*

Stranica MeniPoslastice.aspx:

* Povezana je sa slojom podataka, zbog baze podataka. (using SlojPodataka);
* Povezana je sa folderom modeli jer se u njima konstruktori koji omogućava čitanje podataka (using SlojPodataka.Modeli);
* Iz List<Poslastice> lista tabela učitava podatke, I zatim uz pomoć store procedure proverava da li je korisnik admin, ukoliko jeste (btKreiraj.Visible) dugme kreiraj je vidljivo, ukoliko nije dugme nije vidljivo (btKreiraj.Visible = false;)
* btPretrazi\_Click() metoda – isčitava podatke iz baze preko konstruktora, zatim iz liste isčitava poslastice prema vrsti preko store procedure (DajSvePoslasticePremaVrsti()), zatim se u tabelu ispisuju svi ti podaci (gvTabelaPoslastica.DataSource = lista, gvTabelaPoslastica.DataBind();)

using SlojPodataka;

using SlojPodataka.Modeli;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class MeniPoslastice : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

Baza baza = new Baza();

List<Poslastice> lista = baza.DajSvePoslastice();

gvTabelaPoslastica.DataSource = lista;

gvTabelaPoslastica.DataBind();

if (!string.IsNullOrWhiteSpace(User.Identity.Name))

{

Posetilac posetilac = baza.DajPosetiocaPremaKorisnickomImenu(User.Identity.Name);

btKreiraj.Visible = posetilac.PosetilacJeAdministrator;

}

else

{

btKreiraj.Visible = false;

}

}

}

protected void btPretrazi\_Click(object sender, EventArgs e)

{

Baza baza = new Baza();

string stringPretrage = btPretrazi.Text;

List<Poslastice> lista = baza.DajSvePoslasticePremaVrsti(stringPretrage);

gvTabelaPoslastica.DataSource = lista;

gvTabelaPoslastica.DataBind();

}

protected void btKreiraj\_Click(object sender, EventArgs e)

{

Response.Redirect("~/KreiranjePoslastice.aspx");

}

}

}

*Listing 12: „*MeniPoslastice*.aspx“.*

Stranica „KreiranjePoslastice.aspx“ – front-end :

* ima labelu (Label1) – čije su svojstva definisana na isti način kao na prethodnim stranicama.
* Ima textbox (tbKreiranjePoslastice) - čije su svojstva definisana na isti način kao na prethodnim stranicama.
* Ima dugme (btnKreirajPoslasticu) - čije su svojstva definisana na isti način kao na prethodnim stranicama.

<%@ Page Title="" Language="C#" MasterPageFile="~/GlavnaStranica.Master" AutoEventWireup="true" CodeBehind="KreiranjePoslastice.aspx.cs" Inherits="PrezentacioniSloj.KreiranjePoslastice" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Stranica" runat="server">

<asp:Label ID="Label1" runat="server" Font-Bold="True" ForeColor="#660066" Text="Vrsta poslastice:"></asp:Label>

<asp:TextBox ID="tbKreiranjePoslastice" runat="server" BackColor="#FFCCFF" style="margin-left: 14px"></asp:TextBox>

<asp:Button ID="btnKreirajPoslasticu" runat="server" BackColor="#C2627B" BorderColor="#C2627B" Font-Bold="True" Font-Italic="True" ForeColor="#660066" OnClick="btnKreirajPoslasticu\_Click" style="margin-left: 41px" Text="Kreiraj" Width="84px" />

</asp:Content>

*Listing 13: „KreiranjePoslastice.aspx“ – front-end.*

Stranica KreiranjePoslastice.aspx :

* Povezana je sa slojom podataka, zbog baze podataka. (using SlojPodataka);
* Povezana je sa folderom modeli jer se u njima konstruktori koji omogućava čitanje podataka (using SlojPodataka.Modeli);
* Takođe provera da li korisnik admin kao i na prethodnim stranicama na isti način;
* btnKreirajPoslasticu\_Click() metoda – pomoću koje se kreira poslastica, unosi se vrsta poslastice u tbKreiranjePoslastice, zatim se to sačuvava u bazu preko konstruktora.

using SlojPodataka;

using SlojPodataka.Modeli;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class KreiranjePoslastice : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (!string.IsNullOrWhiteSpace(User.Identity.Name))

{

Baza baza = new Baza();

Posetilac posetilac = baza.DajPosetiocaPremaKorisnickomImenu(User.Identity.Name);

if(!posetilac.PosetilacJeAdministrator)

{

Response.Redirect("~/MeniPocetna.aspx");

}

}

else

{

Response.Redirect("~/MeniPocetna.aspx");

}

}

protected void btnKreirajPoslasticu\_Click(object sender, EventArgs e)

{

Baza baza = new Baza();

string VrstaPoslastice = tbKreiranjePoslastice.Text;

Poslastice obj = new Poslastice(VrstaPoslastice);

baza.DodajNovuPoslasticu(obj);

Response.Redirect("~/MeniPoslastice.aspx");

}

}

}

*Listing 14: „KreiranjePoslastice.aspx“*

Stranica „MeniRecepti.aspx“ – front-end:

* sadrži tabelu kao i na stranici MeniPoslastice koja se zove gvTabelaRecepata, ista su svojstva korišćena.
* Sadrži dugme btnKreirajRecept.

<%@ Page Title="" Language="C#" MasterPageFile="~/GlavnaStranica.Master" AutoEventWireup="true" CodeBehind="MeniRecepti.aspx.cs" Inherits="PrezentacioniSloj.MeniRecepti" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Stranica" runat="server">

<br />

<asp:GridView ID="gvTabelaRecepta" runat="server" BackColor="#C1617A">

<EditRowStyle ForeColor="#660066" />

<FooterStyle ForeColor="#660066" />

<HeaderStyle ForeColor="#660066" />

<RowStyle ForeColor="#660066" />

</asp:GridView>

<br />

<asp:Button ID="btnKreirajRecept" runat="server" BackColor="#C1617A" ForeColor="#660066" OnClick="btnKreirajRecept\_Click" Text="Kreirajte novi recept" BorderColor="#C1617A" Font-Bold="True" Font-Italic="True" />

</asp:Content>

*Listing 15: „MeniRecepti.aspx“ – front-end.*

Stranica „MeniRecepti.aspx“:

* Isčitava podatke iz baze preko konstruktora kao i na prethodnim stranicama.
* Prikazuje recepte u tabelu gvTabelaRecepta uz pomoć store procedure DajSveRecepte().
* btnKreirajRecept\_Click() metoda koja vodi na sl stranicu da se kreira recept.

using SlojPodataka;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using SlojPodataka.Modeli;

namespace PrezentacioniSloj

{

public partial class MeniRecepti : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

Baza baza = new Baza();

gvTabelaRecepta.DataSource = baza.DajSveRecepte();

gvTabelaRecepta.DataBind();

if (string.IsNullOrWhiteSpace(User.Identity.Name))

{

btnKreirajRecept.Visible = false;

}

}

protected void btnKreirajRecept\_Click(object sender, EventArgs e)

{

Response.Redirect("~/KreiranjeNovogRecepta.aspx");

}

}

}

*Listing 16: „MeniRecepti.aspx“.*

Stranica „KreiranjeNovogRecepta.aspx“ – front-end :

* sadrži 4 labele (lbNazivRecepta, lbSastojciRecepta, lbPriprema, lbVrstaPoslastice).
* Sadrži 3 textbox-a (tbNazivRecepta, tbSastojciRecepta, tbPriprema).
* Dropdown listu (dplVrstaPoslastice) .

<%@ Page Title="" Language="C#" MasterPageFile="~/GlavnaStranica.Master" AutoEventWireup="true" CodeBehind="KreiranjeNovogRecepta.aspx.cs" Inherits="PrezentacioniSloj.KreiranjeNovogRecepta" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Stranica" runat="server">

<br />

<asp:Label ID="lbNazivRecepta" runat="server" Text="Naziv recepta:" Font-Bold="True" ForeColor="#660066"></asp:Label>

<asp:TextBox ID="tbNazivRecepta" runat="server" style="margin-left: 94px" BackColor="#FFCCFF" ForeColor="#660066"></asp:TextBox>

<br />

<asp:Label ID="lbSastojciRecepta" runat="server" Text="Sastojci recepta:" Font-Bold="True" ForeColor="#660066"></asp:Label>

<asp:TextBox ID="tbSastojciRecepta" runat="server" style="margin-left: 81px" BackColor="#FFCCFF" ForeColor="#660066" Height="90px" TextMode="MultiLine" Width="337px"></asp:TextBox>

<br />

<asp:Label ID="lbPriprema" runat="server" Text="Nacin pripreme recepta:" Font-Bold="True" ForeColor="#660066"></asp:Label>

<asp:TextBox ID="tbPriprema" runat="server" style="margin-left: 34px" Width="324px" BackColor="#FFCCFF" ForeColor="#660066" Height="57px" TextMode="MultiLine"></asp:TextBox>

<br />

<asp:Label ID="lbVrstaPoslastice" runat="server" Font-Bold="True" ForeColor="#660066" Text="Izaberite vrstu poslastice:"></asp:Label>

<asp:DropDownList ID="dplVrstaPoslastice" runat="server" BackColor="#FFCCFF" EnableTheming="True" Font-Italic="True" Font-Overline="False" Font-Underline="False" ForeColor="#660066" Height="16px" style="margin-left: 28px" Width="128px">

</asp:DropDownList>

<br />

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnKreiraj" runat="server" BackColor="#C2627B" ForeColor="#660066" style="margin-left: 39px" Text="Kreiraj" Width="93px" OnClick="btnKreiraj\_Click" BorderColor="#C2627B" Font-Bold="True" Font-Italic="True" Font-Overline="False" />

</asp:Content>

*Listing 17: „KreiranjeNovogRecepta.aspx“ front-end.*

Stranica „KreiranjeNovogRecepta.aspx“ :

* btnKreiraj\_Click() metoda – čitaju se podaci iz baze, moraju se takođe uneti svi podaci koji se zahtevaju (tbNazivRecepta, tbSastojciRecepta, tbPriprema, dplVrstaPoslastice), vrsta poslastice se bira iz ponuđene liste već kreiranih poslastica, idPosetioca se dobija prema korisničkom imenu. Zatim se dodaje u bazu preko store procedure DodajNoviRecept().

using SlojPodataka;

using SlojPodataka.Modeli;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class KreiranjeNovogRecepta : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(User.Identity.Name))

{

Response.Redirect("~/MeniPocetna.aspx");

}

Baza baza = new Baza();

dplVrstaPoslastice.DataSource = baza.DajSvePoslastice();

dplVrstaPoslastice.DataValueField = "IDPoslastice";

dplVrstaPoslastice.DataTextField = "VrstaPoslastice";

dplVrstaPoslastice.DataBind();

}

protected void btnKreiraj\_Click(object sender, EventArgs e)

{

Baza baza = new Baza();

string NazivRecepta = tbNazivRecepta.Text;

string SastojciRecepta = tbSastojciRecepta.Text;

string NacinPripremeRecepta = tbPriprema.Text;

int IDPoslastice = int.Parse(dplVrstaPoslastice.SelectedValue);

string VrstaPoslastice = dplVrstaPoslastice.SelectedItem.Text;

int idPosetioca = baza.DajPosetiocaPremaKorisnickomImenu(User.Identity.Name).IDPosetioca;

Recepti obj = new Recepti(NazivRecepta, SastojciRecepta, NacinPripremeRecepta, IDPoslastice, VrstaPoslastice, idPosetioca);

baza.DodajNoviRecept(obj);

Response.Redirect("~/MeniRecepti.aspx");

}

}

}

*Listing 18: „KreiranjeNovogRecepta.aspx“*

Stranica „MeniONama.aspx“ – front-end :

* sadrži 2 labele;
* i jednu sliku.

<%@ Page Title="" Language="C#" MasterPageFile="~/GlavnaStranica.Master" AutoEventWireup="true" CodeBehind="MeniONama.aspx.cs" Inherits="PrezentacioniSloj.MeniONama" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Stranica" runat="server">

<br/>

<br/>

<br/>

<asp:Label ID="lbTekst" runat="server" Text="Naša priča otpočela je pre više od jedne decenije, kada je osnovana mala zanatska radionica u Takovskoj ulici broj 6. Oslanjajući se na proverena domaća, <br/>

ali i bečka iskustva u pravljenju kolača kao što su torte Esterhazi i Zaher, ponudili smo sugrađanima raznolike torte, kolače, sladolede i kupove. <br/>Italijanski recepti čija

su osnova pavlaka i čokolada dopunili su postojeći asortiman novim aromama, ukusima i oblicima. Tako je nastala jedinstvena <br/>ponuda poslastičarnice Buongiorno, modifikovana

prema domaćim ukusima." Font-Italic="True" Font-Size="Large" ForeColor="#660066"></asp:Label><br/>

<br/>

<img alt="" src="Slike/slika2.jpg" style="width: 1152px; height: 331px; margin-top: 6px" /><br/>

<br/>

<asp:Label ID="lbTekst1" runat="server" Text="Na današnjem prepoznatljivom izgledu maštovito dekorisanih torti i kolača radi pravi tim poslastičara-umetnika, kreativaca koji smelo kombinuju ukuse,<br/> mirise i boje.

Dovoljno je samo reći za koju priliku želite slatkiš i od čega i na kućnu adresu stići će elegantne torte presvučene masom ili šlagom,<br/> na nekoliko spratova,

neobičnih oblika, različitih veličina, krasiće ih trodimenzionalne figure ili likovi omiljenih dečjih junaka. Neće izostati ni sitni kolači za <br/>sve prilike, dok

originalni pokloni, torte u obliku srca, sa pločicama sa posvetom, upakovane u dekorativne kutije stižu na ruke voljene osobe. Svet<br/> čudesnog i neverovatnog ovde postaje stvaran…

" Font-Italic="True" Font-Size="Large" ForeColor="#660066"></asp:Label>

</asp:Content>

*Listing 19: „MeniONama.aspx“ – front-end.*

Što se tiče koda nema nikakvih specijalnih uslova.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class MeniONama : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

}

}

*Listing 20: „MeniONama.aspx“.*

Stranica „MeniKontakt.aspx“ – front-end:

<%@ Page Title=”” Language=”C#” MasterPageFile=”~/GlavnaStranica.Master” AutoEventWireup=”true” CodeBehind=”MeniKontakt.aspx.cs” Inherits=”PrezentacioniSloj.MeniKontakt” %>

<asp:Content ID=”Content1” ContentPlaceHolderID=”Stranica” runat=”server”>

<asp:Label ID=”lbNaslov” runat=”server” Text=”Poslastičarnica Dizny” Font-Bold=”True” Font-Italic=”True” Font-Size=”Larger” ForeColor=”#660066”></asp:Label>

<br />

<br />

<asp:Label ID=”lbOnama” runat=”server” Text=”Jovana Popovica 2c,<br/>

23000 Zrenjanin<br/>

Radno vreme: Ponedeljak – Petak 09:00-17:00<br/>

Subota: 09:00-16:00<br/>

Nedelja: Ne radimo<br/>

Tel: +381 63 7726684; +381 60 4472121<br/>

E-mail: dizny@poslasticarnica.rs” ForeColor=”#660066” Font-Italic=”True”></asp:Label>

</asp:Content>

*Listing 21: „MeniKontakt.aspx“ – front-end.*

Što se tiče koda za stranicu „MeniKontakt.aspx“ nema nikakvih specijalnih uslova.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PrezentacioniSloj

{

public partial class MeniKontakt : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

}

}

*Listing 22: „MeniKontakt.aspx“.*

### 5.2.2 SLOJ PODATAKA

Sloj podataka sadrzi kod u kojem se pozivaju stored procedure, i omogućava se kontakt sa samom SQL bazom podataka.

U okviru ovog sloja postoji folder „Modeli“ u kome se nalaze četiri klase, „Posetilac.cs“, „Poslastice.cs“ i „Recepti.cs“. Svaka od ovih klasa sadrži konstrukore i „get“ i „set“ metode pomoću kojih im se može pristupiti.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SlojPodataka.Modeli

{

public class Posetilac

{

public Posetilac(string email, string lozinka, string korisnickoIme)

{

Email = email;

Lozinka = lozinka;

KorisnickoIme = korisnickoIme;

}

public Posetilac(int iDPosetioca, string email, string lozinka, string korisnickoIme, bool posetilacJeAdministrator)

{

IDPosetioca = iDPosetioca;

Email = email;

Lozinka = lozinka;

KorisnickoIme = korisnickoIme;

PosetilacJeAdministrator = posetilacJeAdministrator;

}

public int IDPosetioca { get; set; }

public string Email { get; set; }

public string Lozinka { get; set; }

public string KorisnickoIme { get; set; }

public bool PosetilacJeAdministrator { get; set; }

}

}

*Listing 23: Kod klase „Posetilac.cs“.*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SlojPodataka.Modeli

{

public class Poslastice

{

public Poslastice(string vrstaPoslastice)

{

VrstaPoslastice = vrstaPoslastice;

}

public Poslastice(int iDPoslastice, string vrstaPoslastice)

{

IDPoslastice = iDPoslastice;

VrstaPoslastice = vrstaPoslastice;

}

public int IDPoslastice { get; set; }

public string VrstaPoslastice { get; set; }

}

}

*Listing 24: Kod klase „Poslastice.cs“.*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SlojPodataka.Modeli

{

public class Recepti

{

public Recepti(int iDRecepta, string nazivRecepta, string sastojciRecepta, string nacinPripremeRecepta, int iDPoslastice, string vrstaPoslastice, int idPosetioca)

{

IDRecepta = iDRecepta;

NazivRecepta = nazivRecepta;

SastojciRecepta = sastojciRecepta;

NacinPripremeRecepta = nacinPripremeRecepta;

IDPoslastice = iDPoslastice;

VrstaPoslastice = vrstaPoslastice;

IDPosetioca = idPosetioca;

}

public Recepti(string nazivRecepta, string sastojciRecepta, string nacinPripremeRecepta, int iDPoslastice, string vrstaPoslastice, int idPosetioca)

{

NazivRecepta = nazivRecepta;

SastojciRecepta = sastojciRecepta;

NacinPripremeRecepta = nacinPripremeRecepta;

IDPoslastice = iDPoslastice;

VrstaPoslastice = vrstaPoslastice;

IDPosetioca = idPosetioca;

}

public int IDRecepta { get; set; }

public string NazivRecepta { get; set; }

public string SastojciRecepta { get; set; }

public string NacinPripremeRecepta { get; set; }

public int IDPoslastice { get; set; }

public string VrstaPoslastice { get; set; }

public int IDPosetioca { get; set; }

}

}

*Listing 25: Kod klase „Recepti.cs“.*

U okviru Sloja podataka nalazi se klasa „Baza.cs“.

Ova klasa služi za uspostavljanje komunikacije sa SQL bazom podataka uz pomoć **„ \_konekcija = new SqlConnection("data source=.\\SQLEXPRESS; database = Poslasticarnica; integrated security=SSPI"); konekcija.Open(); “**.

U okviru ove klase nalaze se stored procedure:

* DajPosetiocaPremaKorisnickomImenu,
* DodajNovuPoslasticu,
* DajSvePoslastice,
* DajSvePoslasticePremaVrsti,
* DajSveRecepte,
* DodajNoviRecept,
* DodajNovogPosetioca,
* DajSvePosetioce.

Sve ove procedure su prethodno realizovane u SQL bazi podataka.

using SlojPodataka.Modeli;

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SlojPodataka

{

public class Baza

{

private SqlConnection \_konekcija;

public Baza()

{

\_konekcija = new SqlConnection("data source=.\\SQLEXPRESS; database = Poslasticarnica; integrated security=SSPI");

\_konekcija.Open();

}

private Poslastice KreirajInstancuPoslastice(SqlDataReader reader)

{

if (reader == null)

throw new ArgumentNullException(nameof(reader));

if (!reader.HasRows)

throw new ArgumentOutOfRangeException(nameof(reader));

int IDPoslastice = (int)reader["IDPoslastice"];

string VrstaPoslastice = reader["VrstaPoslastice"] as string;;

Poslastice resenje = new Poslastice(IDPoslastice, VrstaPoslastice)

;

return resenje;

}

public Posetilac DajPosetiocaPremaKorisnickomImenu(string korisnickoIme)

{

using (SqlCommand command = \_konekcija.CreateCommand())

{

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "DajPosetiocaPremaKorisnickomImenu";

command.Parameters.Add("@KorisnickoIme", SqlDbType.NVarChar).Value = korisnickoIme;

using (SqlDataReader reader = command.ExecuteReader())

{

reader.Read();

return KreirajInstancuPosetioca(reader);

}

}

}

public void DodajNovuPoslasticu(Poslastice obj)

{

throw new NotImplementedException();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public bool DodajNovuPoslasticu(Recepti obj)

{

if (obj == null)

throw new ArgumentNullException(nameof(obj));

try

{

using (SqlCommand komanda = \_konekcija.CreateCommand())

{

komanda.CommandText = "DodajNovuPoslasticu";

komanda.CommandType = System.Data.CommandType.StoredProcedure;

komanda.Parameters.Add("@VrstaPoslastice", SqlDbType.NVarChar).Value = obj.VrstaPoslastice;

komanda.ExecuteScalar();

return true;

}

}

catch (Exception)

{

return false;

//throw;

}

}

public List<Poslastice> DajSvePoslastice()

{

List<Poslastice> resenja = new List<Poslastice>();

using (SqlCommand command = \_konekcija.CreateCommand())

{

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "DajSvePoslastice";

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

resenja.Add(KreirajInstancuPoslastice(reader));

}

return resenja;

}

}

}

public List<Poslastice> DajSvePoslasticePremaVrsti(string stringPretrage)

{

List<Poslastice> resenja = new List<Poslastice>();

using (SqlCommand command = \_konekcija.CreateCommand())

{

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "DajSvePoslasticePremaVrsti";

command.Parameters.Add("@VrstaPoslastice", SqlDbType.NVarChar).Value = stringPretrage;

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

resenja.Add(KreirajInstancuPoslastice(reader));

}

return resenja;

}

}

}

private Recepti KreirajInstancu(SqlDataReader reader)

{

if (reader == null)

throw new ArgumentNullException(nameof(reader));

if (!reader.HasRows)

throw new ArgumentOutOfRangeException(nameof(reader));

int IDRecepta = (int)reader["IDRecepta"];

string NazivRecepta = reader["NazivRecepta"] as string;

string SastojciRecepta = reader["SastojciRecepta"] as string;

string NacinPripremeRecepta = reader["NacinPripremeRecepta"] as string;

string VrstaPoslastice = reader["VrstaPoslastice"] as string;

int IDPoslastice = (int)reader["IDPoslastice"];

int IDPosetioca = (int)reader["IDPosetioca"];

Recepti resenje = new Recepti(IDRecepta, NazivRecepta, SastojciRecepta, NacinPripremeRecepta, IDPoslastice, VrstaPoslastice, IDPosetioca)

;

return resenje;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public List<Recepti> DajSveRecepte()

{

List<Recepti> resenja = new List<Recepti>();

using (SqlCommand command = \_konekcija.CreateCommand())

{

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "DajSveRecepte";

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

resenja.Add(KreirajInstancu(reader));

}

return resenja;

}

}

}

public bool DodajNoviRecept(Recepti obj)

{

if (obj == null)

throw new ArgumentNullException(nameof(obj));

try

{

using (SqlCommand komanda = \_konekcija.CreateCommand())

{

komanda.CommandText = "DodajNoviRecept";

komanda.CommandType = System.Data.CommandType.StoredProcedure;

komanda.Parameters.Add("@NazivRecepta", SqlDbType.NVarChar).Value = obj.NazivRecepta;

komanda.Parameters.Add("@SastojciRecepta", SqlDbType.NVarChar).Value = obj.SastojciRecepta;

komanda.Parameters.Add("@NacinPripremeRecepta", SqlDbType.NVarChar).Value = obj.NacinPripremeRecepta;

komanda.Parameters.Add("@IDPosetioca", SqlDbType.NVarChar).Value = obj.IDPosetioca;

komanda.Parameters.Add("@IDPoslastice", SqlDbType.NVarChar).Value = obj.IDPoslastice;

komanda.ExecuteScalar();

return true;

}

}

catch (Exception)

{

return false;

//throw;

}

}

private Posetilac KreirajInstancuPosetioca(SqlDataReader reader)

{

if (reader == null)

throw new ArgumentNullException(nameof(reader));

if (!reader.HasRows)

throw new ArgumentOutOfRangeException(nameof(reader));

int IDPosetioca = (int)reader["IDPosetioca"];

string Email = reader["Email"] as string; ;

string Lozinka = reader["Lozinka"] as string; ;

string KorisnickoIme = reader["KorisnickoIme"] as string; ;

bool PosetilacJeAdministrator = (bool)reader["PosetilacJeAdministrator"];

Posetilac resenje = new Posetilac(IDPosetioca, Email, Lozinka, KorisnickoIme, PosetilacJeAdministrator)

;

return resenje;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public bool DodajNovogPosetioca(Posetilac obj)

{

if (obj == null)

throw new ArgumentNullException(nameof(obj));

try

{

using (SqlCommand komanda = \_konekcija.CreateCommand())

{

komanda.CommandText = "DodajNovogPosetioca";

komanda.CommandType = System.Data.CommandType.StoredProcedure;

komanda.Parameters.Add("@Email", SqlDbType.NVarChar).Value = obj.Email;

komanda.Parameters.Add("@Lozinka", SqlDbType.NVarChar).Value = obj.Lozinka;

komanda.Parameters.Add("@KorisnickoIme", SqlDbType.NVarChar).Value = obj.KorisnickoIme;

komanda.Parameters.Add("@PosetilacJeAdministrator", SqlDbType.Bit).Value = DajSvePosetioce().Count == 0;

komanda.ExecuteScalar();

return true;

}

}

catch (Exception)

{

return false;

//throw;

}

}

private List<Posetilac> DajSvePosetioce()

{

List<Posetilac> posetioci = new List<Posetilac>();

using (SqlCommand command = \_konekcija.CreateCommand())

{

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "DajSvePosetioce";

using (SqlDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

posetioci.Add(KreirajInstancuPosetioca(reader));

}

return posetioci;

}

}

}

}

}

*Listing 26: Klasa “Baza.cs”.*

## 5.3 TESTIRANJE APLIKACIJE

Testiranje aplikacije „Poslastičarnica“ se izvršava nakon završetka povezivanja sa bazom podataka, kada se odradi, aplikacija se može pokrenuti u Visual Studio-u.

Prilikom pokretanja aplikacije, u zavisnosti sa koje stranice pokrećemo na tu stranicu će nas odvesti. Prvo što korisnik treba da uradi jeste da ode na stranicu registracije i da prijavi ukoliko ima nalog, ukoliko nema nalog da kreira i samim tim sačuva svoje podatke u bazu podataka.

Kada je korisnik ulogovan kao običan Administrator treba proveriti sledeće stavke:

* Treba da ima mogućnost kreiranja recepta (na stranici Recepti),
* Treba da ima mogućnost pretrage prema nazivu (na stranici Poslastice),
* Treba da ima mogućnost kreiranja poslastice (na stranici Poslastice).

Za sve ove stavke se podrazumeva da se moraju uneti sva potrebna navedena polja.

Kada je korisnik ulogovan kao običan Običan korisnik treba proveriti sledeće stavke:

* Treba da ima mogućnost kreiranja recepta (na stranici Recepti),
* Treba da ima mogućnost pretrage prema nazivu (na stranici Poslastice).

Za sve ove stavke se podrazumeva da se moraju uneti sva potrebna navedena polja.

Ukoliko aplikacija prikazuje podatke prilikom unosa svih podataka, aplikacija je uspešno testirana, i može početi sa korišćenjem.

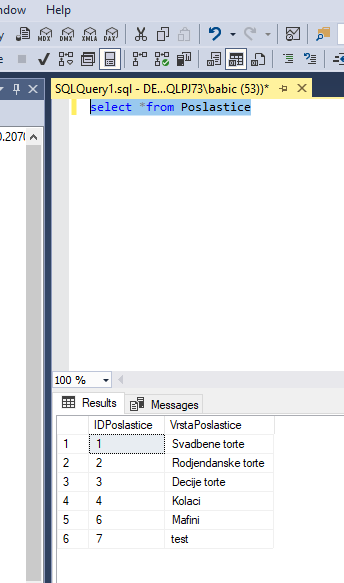
|  |  |  |
| --- | --- | --- |
| **Broj slučaja** | **Opis slučaja** | **Test podaci** |
| 1. | Dodavanje poslastice na stranici „Poslastice“, provera da li će se sačuvati u bazi podataka i da li će se prikaazati u tabeli. (prikaz na slici 12. 13. i 14.) | test |
| 2. | Pretraživanje poslastice po nazivu (slika 15.) | Mafini |
| 3. | Unos recepta na stranici recepti () | test, test, testttt, svadbene torte, korisnik ulogovan kao admin |



*Slika 12: Testiranje unosa poslastica.*

**

*Slika 13: Prikaz dodate poslastice u tabeli.*

**

*Slika 14: Prikaz dodate poslastice u SQL bazi podataka.*

**

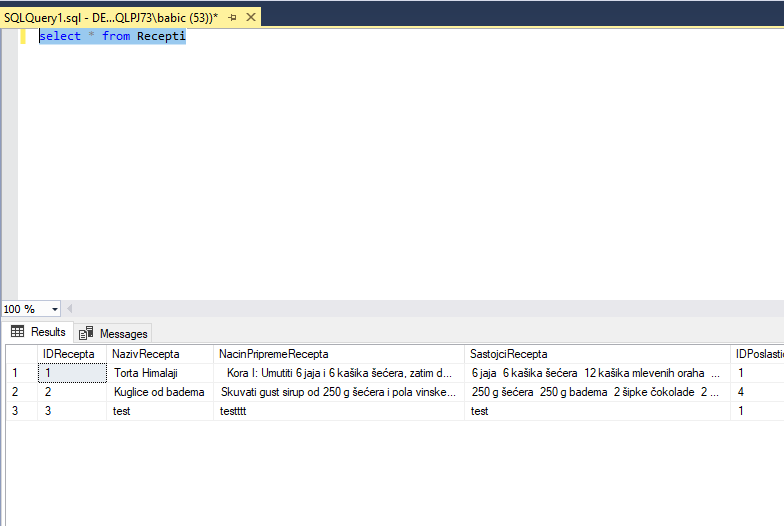
*Slika 15: Pretraga poslastice.*

**

*Slika 16: Unos recepta.*

**

*Slika 17: Prikaz u tabeli.*

**

*Slika 18: Prikaz u SQL bazi.*

# OPIS PRIMENJENIH TEHNOLOGIJA

U ovom seminarskom je korišćen Visual Studio kao tekst editor, IIS Express, SQL Server Management Studio.U okviru realizacije programskog koda:

* Korisnički interfejs ja napravljen pomoću ASPX Web formi.
* Podaci sa korisničkog interfejsa i iz baze podataka su obrađeni pomoću C#.
* Konekcija sa bazom urađena pomoću Sql Client-a**.**

Takođe za čuvanje i smeštanje celog fajla korišćen je GitHub.

## TABELARNI PRIKAZ SLOJEVA I PODSLOJEVA APLIKACIJE

|  |  |  |
| --- | --- | --- |
| SLOJ | PODSLOJ | REALIZACIJA U SEMINARSKOM RADU |
| PREZENTACIONI SLOJ | MeniPocetna | ASPX Web forma |
|  | MeniPoslastice | ASPX Web forma |
|  | KreiranjePoslastice | ASPX Web forma |
|  | MeniRecepta | ASPX Web forma |
|  | KreiranjeNovogRecepta | ASPX Web forma |
|  | MeniKontakt | ASPX Web forma |
|  | MeniOnama | ASPX Web forma |
| SLOJ PODATAKA | Folder Modeli | Klase |
|  | Baza | Klase |

*Tabela 1: Prikaz slojeva i podslojeva aplikacije „Poslastičarnica“.*

## GITHUB

## O GITHUB-U

**GitHub** (originalno poznat kao — **Logical Awesome LLC**) je veb-baziran [hosting servis](https://sr.wikipedia.org/wiki/Internet_hosting) za kontrolu verzije, [Git](https://sr.wikipedia.org/wiki/%D0%93%D0%B8%D1%82_(%D1%81%D0%BE%D1%84%D1%82%D0%B2%D0%B5%D1%80)). Pruža Git funkcionalnosti: [distribuiranu kontrolu revizija](https://sr.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D1%81%D1%82%D1%80%D0%B8%D0%B1%D1%83%D0%B8%D1%80%D0%B0%D0%BD%D1%83_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%83_%D1%80%D0%B5%D0%B2%D0%B8%D0%B7%D0%B8%D1%98%D0%B0&action=edit&redlink=1) i [menadžment izvornog koda](https://sr.wikipedia.org/w/index.php?title=%D0%9C%D0%B5%D0%BD%D0%B0%D1%9F%D0%BC%D0%B5%D0%BD%D1%82_%D0%B8%D0%B7%D0%B2%D0%BE%D1%80%D0%BD%D0%BE%D0%B3_%D0%BA%D0%BE%D0%B4%D0%B0&action=edit&redlink=1) (eng. Source Control Management — SCM), dodajući dodatne funkcije. Za razliku od Git-a, koji je striktno alat koji se koristi iz [komandne linije](https://sr.wikipedia.org/wiki/Command-line_interface), GitHub pruža [veb grafički interfejs](https://sr.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B0%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%98%D0%B5), radnu površinu i mobilnu integraciju. Takođe pruža [kontrolu pristupa](https://sr.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B0_%D0%BF%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%BF%D0%B0) i nekoliko funkcija za saradnju, kao što su [praćenje grešaka](https://sr.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%B0%D1%9B%D0%B5%D1%9A%D0%B5_%D0%B3%D1%80%D0%B5%D1%88%D0%B0%D0%BA%D0%B0&action=edit&redlink=1) ([eng.](https://sr.wikipedia.org/wiki/%D0%95%D0%BD%D0%B3%D0%BB%D0%B5%D1%81%D0%BA%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA) bug tracking), [zahteve za dodavanje novih karakteristika](https://sr.wikipedia.org/w/index.php?title=%D0%97%D0%B0%D1%85%D1%82%D0%B5%D0%B2_%D0%B7%D0%B0_%D0%B4%D0%BE%D0%B4%D0%B0%D0%B2%D0%B0%D1%9A%D0%B5_%D0%BD%D0%BE%D0%B2%D0%B8%D1%85_%D0%BA%D0%B0%D1%80%D0%B0%D0%BA%D1%82%D0%B5%D1%80%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0&action=edit&redlink=1) (eng. feature request), [upravljanje zadacima](https://sr.wikipedia.org/w/index.php?title=%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D1%99%D0%B0%D1%9A%D0%B5_%D0%B7%D0%B0%D0%B4%D0%B0%D1%86%D0%B8%D0%BC%D0%B0&action=edit&redlink=1) (eng. task management) i mogućnost pravljenja [viki](https://sr.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8)dokumentacije za svaki projekat.

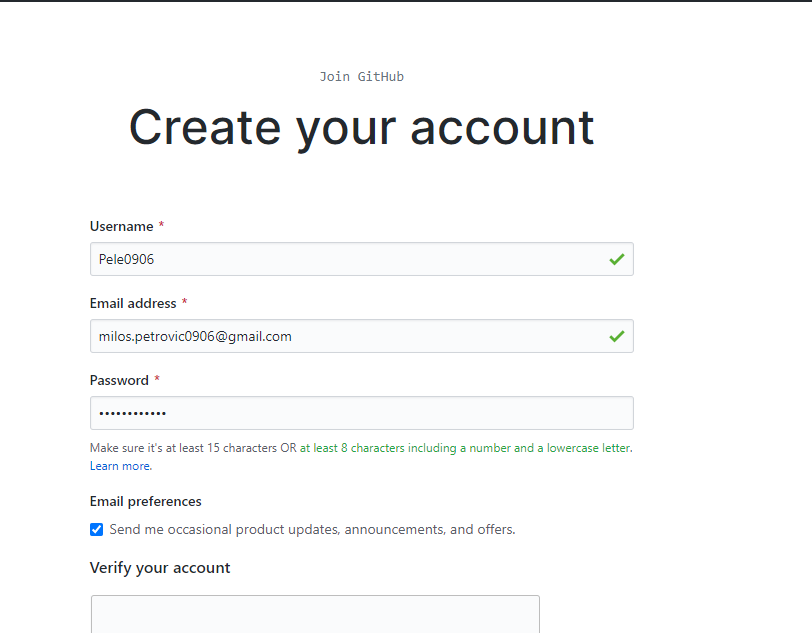
GitHub pruža planove za privatna [skladišta](https://sr.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D1%98%D1%83%D0%BC) kao i besplatne naloge, koji se obično koriste kao hostovi za sofverske projekte [otvorenog koda](https://sr.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D0%B2%D0%B5%D1%80_%D0%BE%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BE%D0%B3_%D0%BA%D0%BE%D0%B4%D0%B0). GitHub izveštaji iz aprila 2016. godine tvrde postojanje više od 14 miliona korisnika i 35 miliona skladišta, što GitHub čini najvećim hostom izvornog koda na svetu.

GitHub se najviše koristi za izvorne kodove.

Osim [izvornog koda](https://sr.wikipedia.org/wiki/Source_code), GitHub podržava sledeće formate i funkcije:

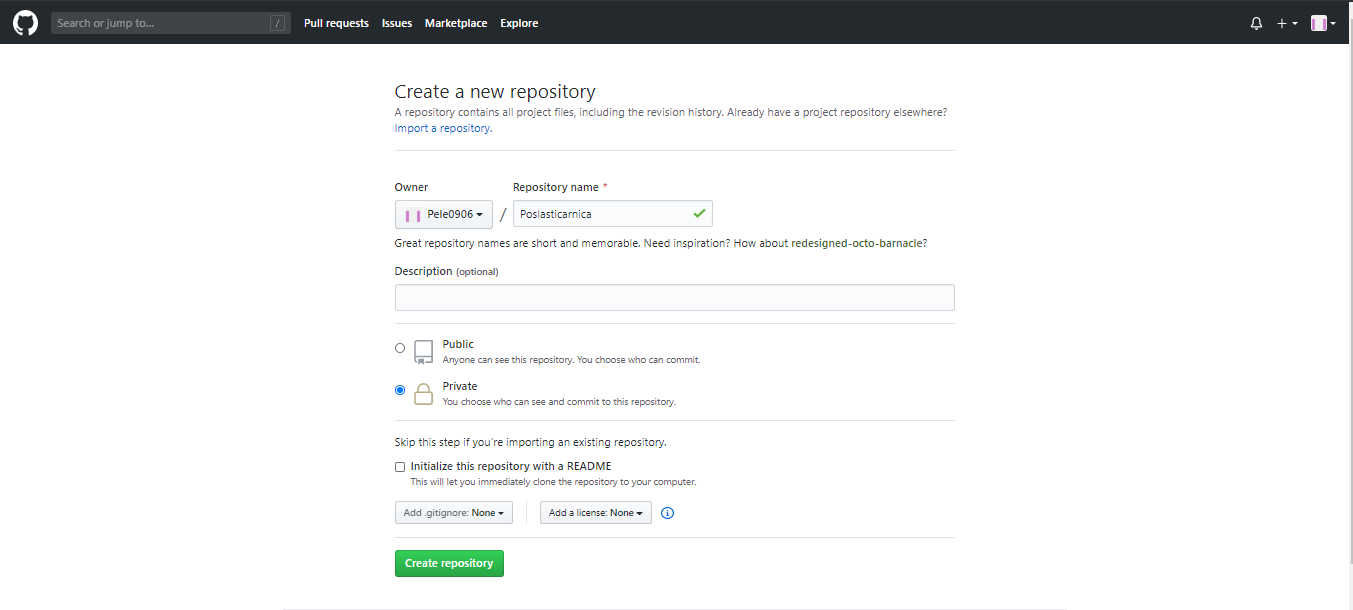
* Dokumentaciju, uključujući automatski generisane [README](https://sr.wikipedia.org/w/index.php?title=README&action=edit&redlink=1) fajlove u raznim [Markdown](https://sr.wikipedia.org/w/index.php?title=Markdown&action=edit&redlink=1) fajl formatima (pogledati [README fajlovi na GitHub-u](https://sr.wikipedia.org/w/index.php?title=README&action=edit&redlink=1)).
* [Praćenje problema](https://sr.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%B0%D1%9B%D0%B5%D1%9A%D0%B5_%D0%BF%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC%D0%B0&action=edit&redlink=1) (uključujući zahteve za novim karakteristikama) sa labelama, prekretnicama, zastupnicima i pretraživačem.
* [Viki](https://sr.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8) podrška.
* Zahteve za pregled izmena sa recenzijama koda i komentarima.
* Istorija izvršavanja.
* Grafikoni: puls, saradnici, frekvencije koda, mreže, članovi.
* Adresar integracija.
* E-mail notifikacije.
* Opcija prijave za notifikacije o nekome, koristeći znak @.
* Emoji podrška.
* Mali [veb-sajtovi](https://sr.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D1%98%D1%82) se mogu hostovati sa javnih skladišta na GitHub-u. URL format je: [http://nalog.github.io](http://nalog.github.io/).
* Ugnježdene [liste zadataka](https://sr.wikipedia.org/w/index.php?title=%D0%9B%D0%B8%D1%81%D1%82%D0%B0_%D0%B7%D0%B0%D0%B4%D0%B0%D1%82%D0%B0%D0%BA%D0%B0&action=edit&redlink=1) unutar fajlova
* Vizuelizacija [geoprostornih](https://sr.wikipedia.org/w/index.php?title=Geospatial_analysis&action=edit&redlink=1) 3D podataka koji se mogu pregledati koristeći novointegrisani STL fajl pregledač koji otvara fajlove na 3D platnu. Pregledač pokreću [WebGL](https://sr.wikipedia.org/w/index.php?title=WebGL&action=edit&redlink=1) i [Three.js](https://sr.wikipedia.org/w/index.php?title=Three.js&action=edit&redlink=1)
* [Photoshop](https://sr.wikipedia.org/wiki/Adobe_Photoshop)-ov nativni PSD format se može pregledati i porediti sa prethodnim verzijama istog fajla

Kreiranje naloga na GitHub-u:

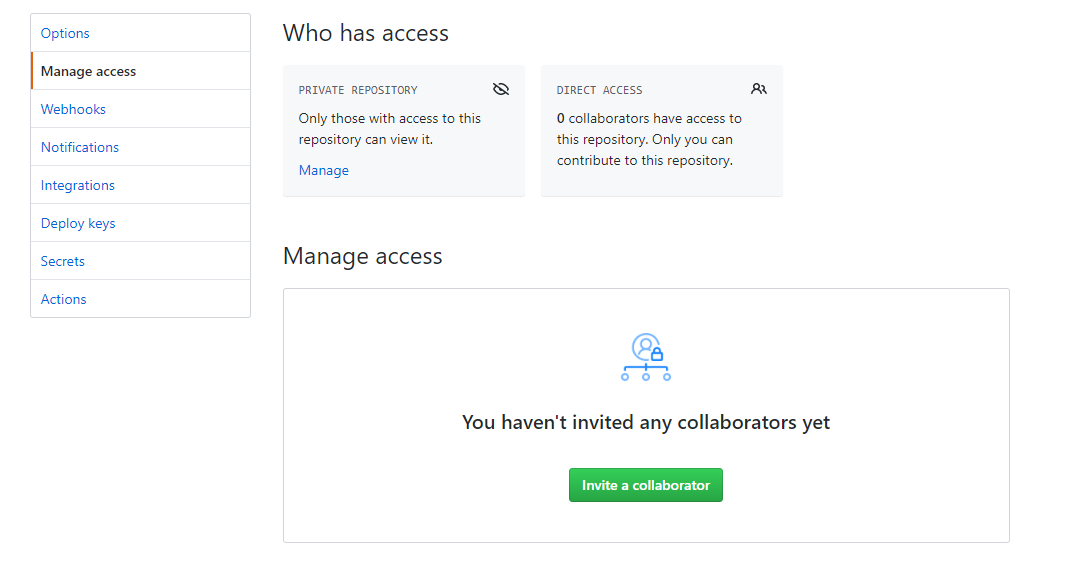


*Slika 12: Kreiranje naloga na GitHub-u.*

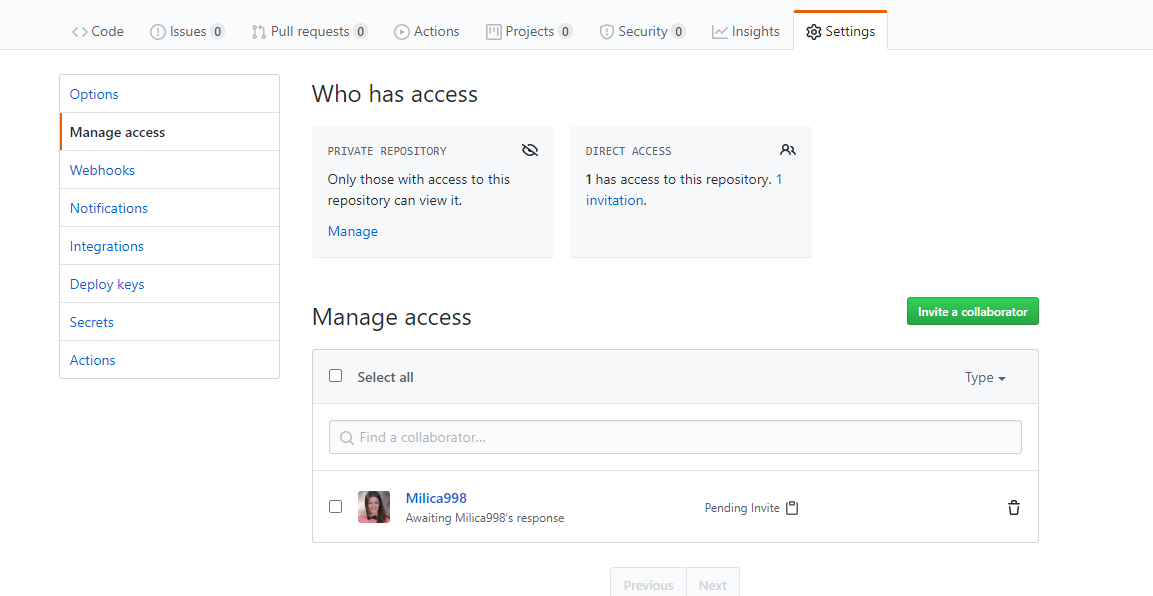
Kreiranje repozitorijuma za aplikaciju:



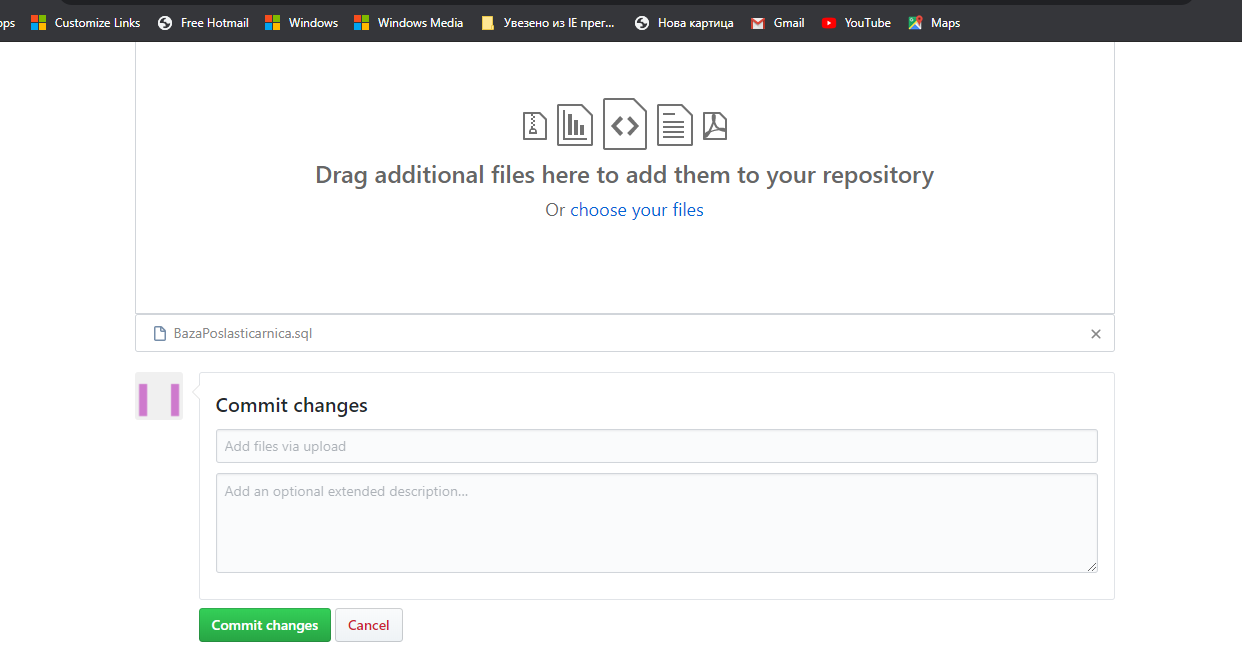
*Slika 13: Kreiranje repozitorijuma.*

**

*Slika 14: Povezivanje sa drugim nalogom.*

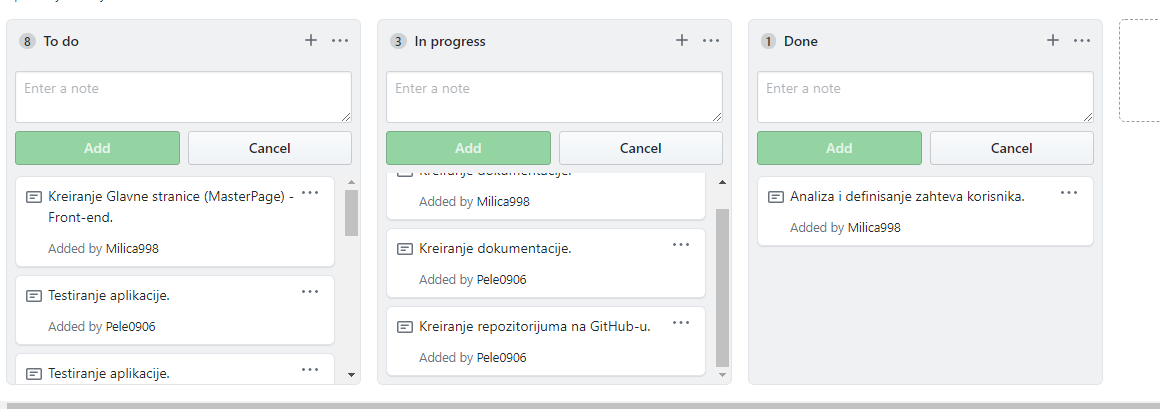
**

*Slika 15: Dodavanje drugog naloga.*

**

*Slika 16: Kreiranja Commit-a.*

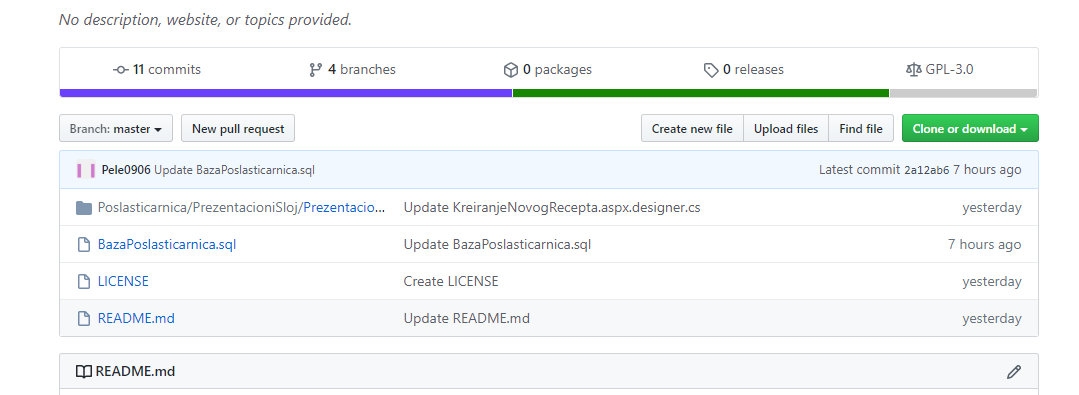
Plan projekta i faze razvoj softvera, prikazano je na slici 17.:



*Slika 17: Plan projekta na GitHub-u.*

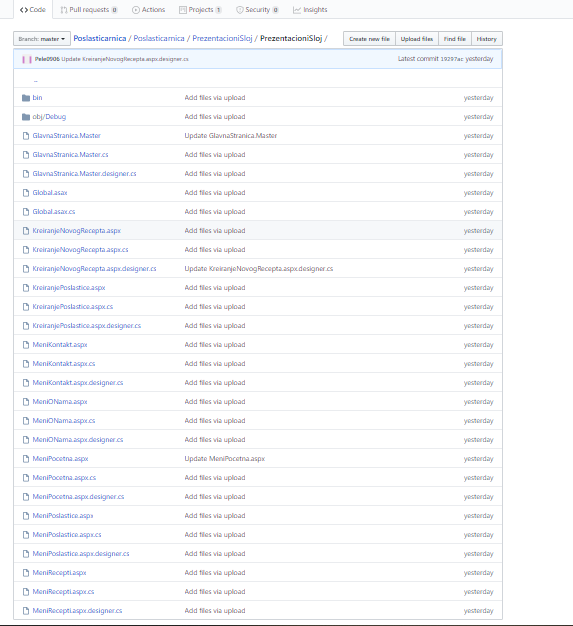
Na slici 18. su prikazani fajlovi koji se nalaze na GitHub-u. Fajl „README.md“ i LICENSE su kreirani fajlovi odnosno napravljeni su putem pritiska na dugme „Create new file“, LICENSE je kreirana po upustvu sa vežbi iz predmeta Razvoj softvera otvorenog koda, fajl „README.md“ predstavlja fajl koji opisuje samu aplikaciju.

Dok su ostali fajlovi dodati, tj dodati su klikom na dugme „Upload file“.



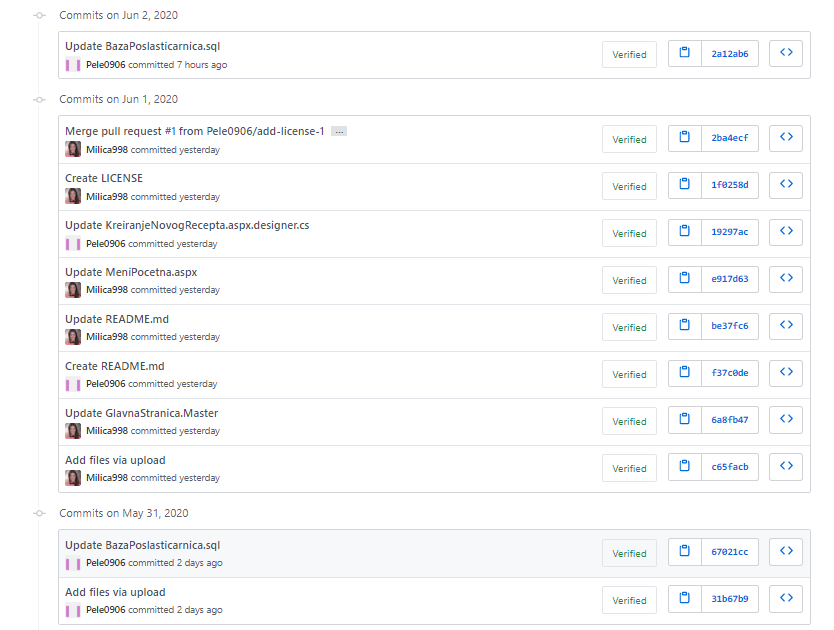
*Slika 18:Prikaz fajlova na GitHub-u.*

Na slici 18. se može videti šta se nalazi u fajlu „Poslasticarnica“, odnosno koji sve slojevi, stranice i klase postoje u okviru ove aplikacije.



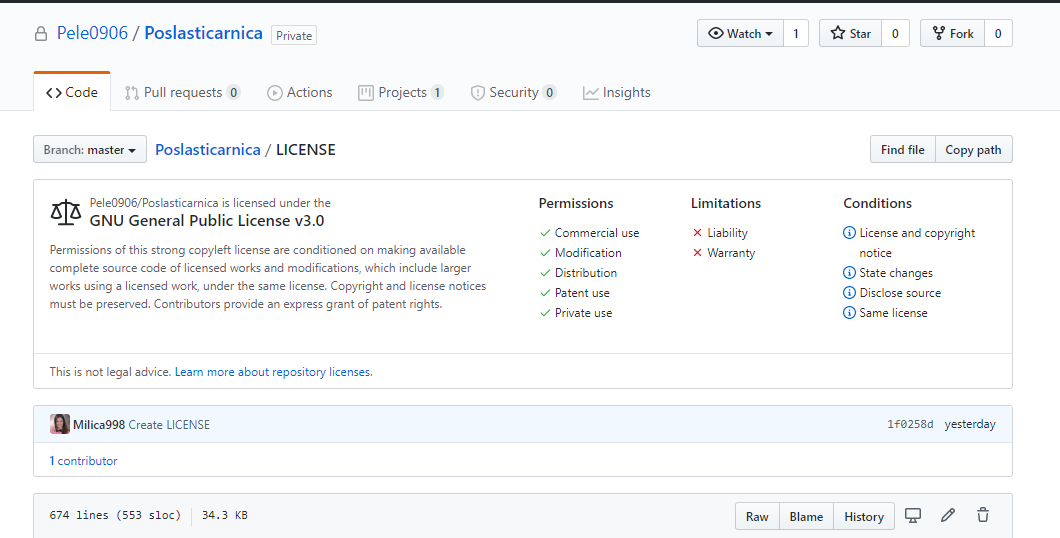
*Slika 19: Fajl „Poslasticarnica“.*

Na slici 19. se može videti ukupan broj commit-ova - 12, od kojih student Miloš Petrović ima 5 commit-ova, a Milica Babić 8.



Slika 20: Commit-ovi.

Na slici 20. se može videti izgled i sadržaj licence.



Slika 21: Licenca.

# LITERATURA:

[1] <https://saveti.kombib.rs/sta-je-github-vikipedija>

[2] <http://www.tfzr.uns.ac.rs/Content/files/0/RSOKVezbe%2020192020%20-%20dokument%20FINALNO.pdf>

[3] <http://www.tfzr.uns.ac.rs/Predmet/razvoj-softvera-otvorenog-koda/sadrzaj-predmeta>

[4] <https://help.github.com/en/github>