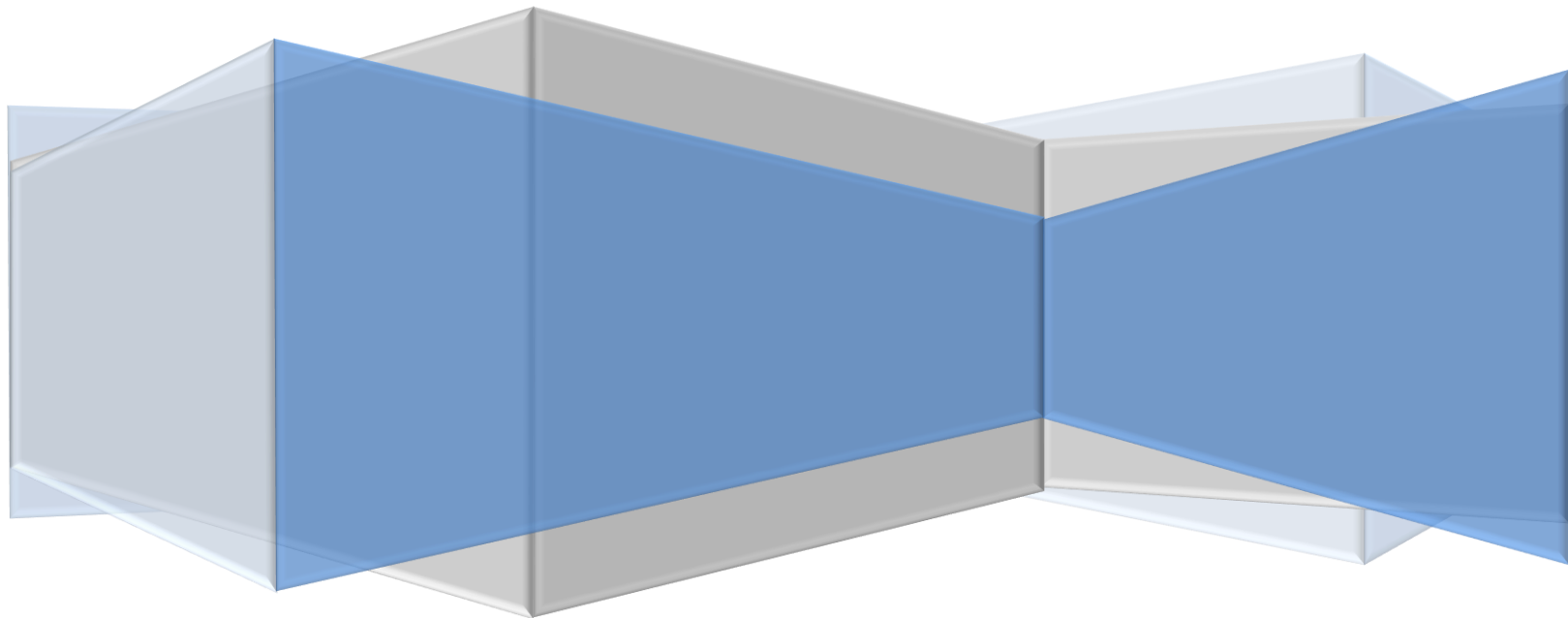


ASP.NET MVC

Registar saobraćajnih prekršaja



Prirodno-matematički fakultet Sarajevo
Juni, 2018.

Contents

• Opis projekta	2
• ASP.NET	2
• MVC	3
• Entity Framework	4
• Realizacija projekta	5

- **Opis projekta**

Ovim projektom imamo za cilj da se upoznamo sa MVC obrascom programske arhitekture na ASP.NET razvojnoj platformi, a koji služi za razvoj web aplikacija. Da bi smještali i dobavljali podatke, koristit ćemo .NET razvojno okruženje koje se naziva Entity Framework za definisanje i rad sa modelima, a koje će dalje koristiti Code First tehniku koja.

- **ASP.NET**

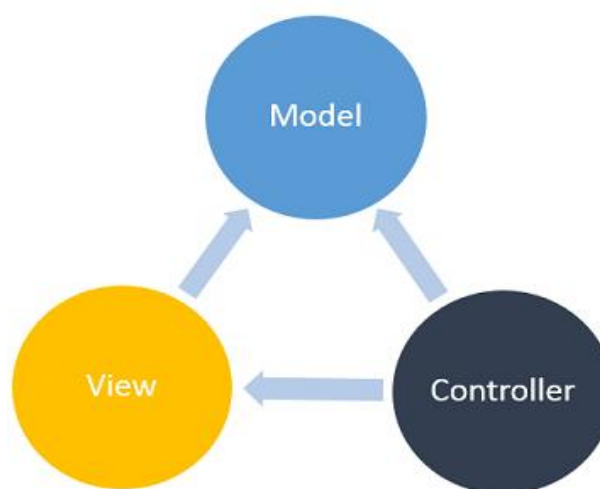
ASP.NET je razvojna platforma za razvijanje web aplikacija (dinamičkih web stranica), prva verzija ASP.NET 1.0 je izašla u januaru 2002. godine zamjenjujući klasični ASP kao Microsoftov poslužiteljski skriptni jezik, trenutna stabilna verzija je ASP.NET 4.5. Više je razloga za prelazak na novu platformu, tj. sa ASP-a na ASP.NET, među kojima su najbitniji brzina i performanse.

Najveća prednost ASP.NET-a je što se kôd prevodi, poboljšavajući brzinu. Kako je baziran na .NET frameworku omogućava razvijanje web aplikacija na način sličan razvijanju desktop aplikacija sa svim mogućnostima objektno orijentiranog programiranja.

- **MVC**

MVC (*Model-View-Controller*) je obrazac programske arhitekture široko prihvaćen za razvoj web aplikacija. Prednost MVC-a je što odvađa bilo kakav prikaz informacija od korisničke interakcije s istom, te omogućava nezavisan razvoj, testiranje i održavanje aplikacije. Sastavni dijelovi MVC-a su:

- **Model** (*Model*)
- **Upravljač** (*Controller*)
- **Pogled** (*View*)



Model je skup klasa koje opisuju podatke sa kojima radimo kao i poslovnu logiku.

Pogled prikazuje kako će izgledati interfejs naše aplikacije. Odnosi se na čisti HTML, koji odlučuje kako će da izgleda korisnički interfejs.

Upravljač sadrži prezentacijsku logiku, obrađuje sve unose korisnika, obavlja izmjenu podataka s modelom i određuje pogled. Upravljač sadrži više akcija koje se izvode ovisno o unosu korisnika.

- **Entity Framework**

Entity Framework (EF) je objektno/relacijski programski okvir, te skup tehnologija koji podržava razvoj aplikacija koje su orijentirane na podatke. Pri tome predstavlja objektno-relacijski (engl. object-relational) alat za pridruživanje, odnosno mapiranje koji omogućava .NET programerima da rade s relacijskim podacima koristeći specifične objekte domene.

EF eliminira potrebu stvaranja većine programskog koda za pristup podacima koje bez njegove primjene programer mora stvoriti. Postoje dva općenita načina primjene EF-a:

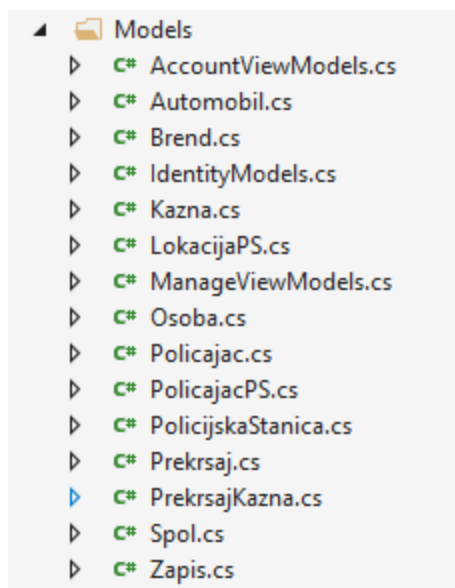
- **Database first** - Kad već postoji baza podataka ili se ona oblikuje u početnom koraku, te se tek nakon toga stvaraju ostali dijelovi aplikacije.
- **Code first** - Kad se želi fokusirati na razrede domene i tek nakon toga stvoriti baza podataka prema razredima domene. Ovaj pristup je korišten u našem projektu.

- **Realizacija projekta**

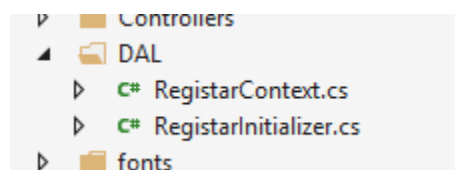
Za izradu projekta je korišten Visual Studio 2017. Rješenje je tipa ASP.NET WEB aplikacije.

Projekat se sastoji od cjelina Model, View i Controller koji imaju zasebne foldere unutar njega.

Prva cjelina koja je kreirana je Model, koja sadrži klase pomoću kojih je kreirana baza.



Da bi baza bila kreirana na serveru bili su potrebni i file-ovi RegistrarContext.cs i RegistrarInitializer.cs koji su smješteni u folderu DAL(Data Access Layer) .

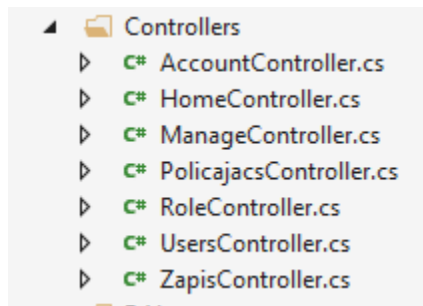


Takođe, primarnu ulogu u vezi s serverom ima Web.config. Pomoću connectionString-a smo definisali vezu sa serverom.

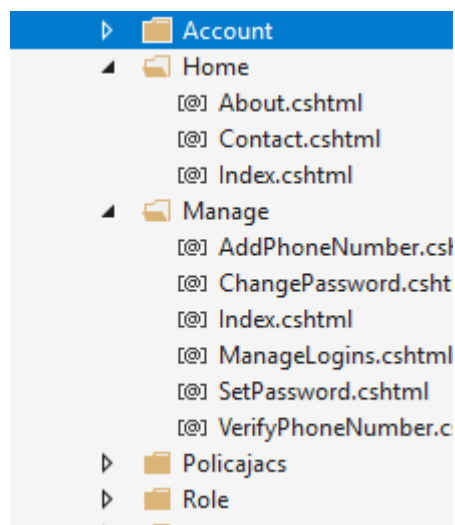
```
<connectionStrings>
!-- 195.130.32.34,1032 172.16.24.3,1032-->
  <add name="RegistarMVC" connectionString="Data Source=195.130.32.34,1032;
    Initial Catalog=RegistarMVC;
    Integrated Security=False;
    User ID=sa;Password=pmf911;
    Pooling=false;
    Connect Timeout=30;
    Encrypt=False;
    TrustServerCertificate=False;
    ApplicationIntent=ReadWrite;
    MultiSubnetFailover=False" providerName="System.Data.SqlClient" />
</connectionStrings>
```

Vidimo da je naziv baze na serveru RegistarMVC.

Nakon uspešnog kreiranja baze, kreirani su Controller-i koji su pomagali u manipulisanju sa bazom.

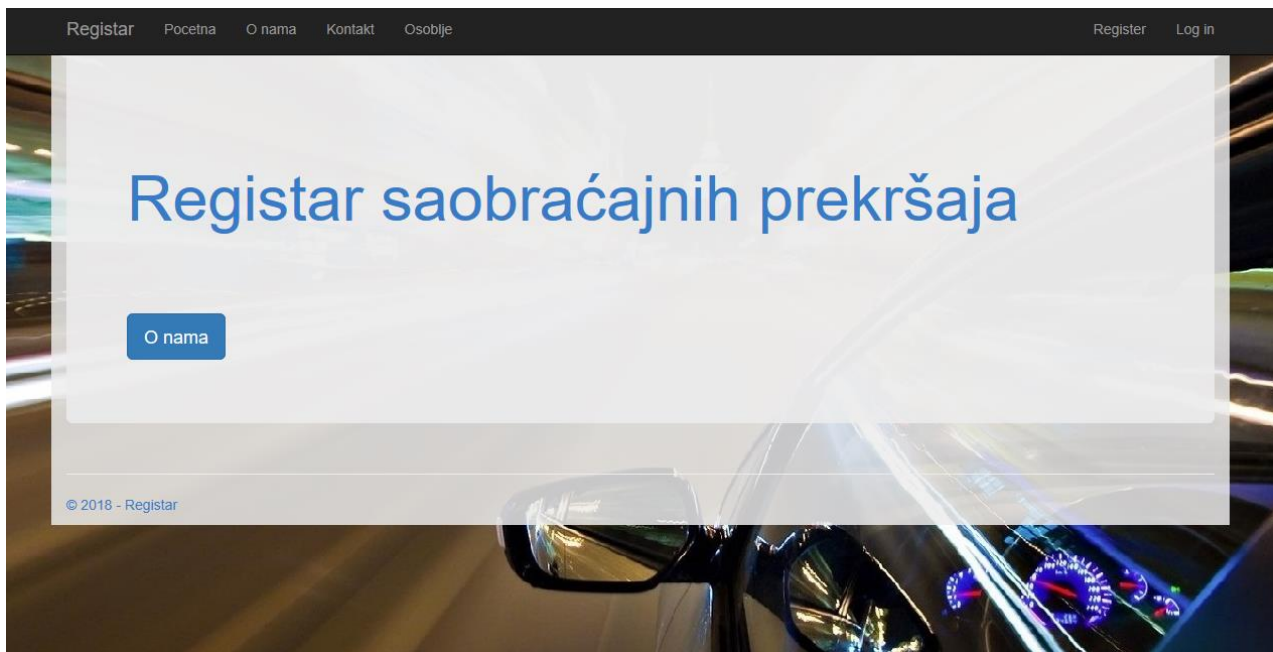


Prilikom kreiranja Controller-a, kreirani su i View-ovi u skladu s njima. Većini funkcija u Controller-ovi tipa „cshtml“ koji funkcije.



ima su pridruženi file-prikazuju aktivnost te

Za početnu stranu zadužen je HomeController. Vidimo predložene tabove „O nama“ i „Kontakt“, koji su takođe dio HomeController-a.



„O nama“ daje kratki uvod o ideji projekta.



„Kontakt“ sadrži listu bitnih institucija koje su povezane sa samim projektom.

Kontakt

Sarajevo
La Benevolencije 16
Centrala
Tel: [+387\(0\)33 286-700](tel:+387033286700)
ID BR: Ministarstvo 4200220350006
Uprava policije 4200220350146

UPRAVA ADMINISTRACIJE
Pomoćnik ministra
Tel: [+387\(0\)33 286-732](tel:+387033286732)
[+387\(0\)33 286-731](tel:+387033286731)
upravaadministracije@mup.ks.gov.ba

UPRAVA PODRŠKE
Tel: [+387\(0\)33 286-770](tel:+387033286770)
[+387\(0\)33 286-771](tel:+387033286771)
upravapodrske@mup.ks.gov.ba

Realised by:

<i>Emina Salkanović</i> eminasalkanovic@outlook.com	<i>Amra Fočo</i> focoamra@gmail.com	<i>Sadina Rašidović</i> sadinarasidovic@outlook.com	<i>Amela Pračić</i> amela_pracic@hotmail.com	<i>Amina Safić</i> aminasafic61@gmail.com
---	--	---	---	--

© 2018 - Registar

AccountController sadrži veoma bitne funkcije za korisnički dio sistema. Pod tim podrazumijevamo mogućnosto registracije i prijave na sistem.

Register.

Create a new account.

Email

UserName

Password

Confirm password

UserRoles

Prijava

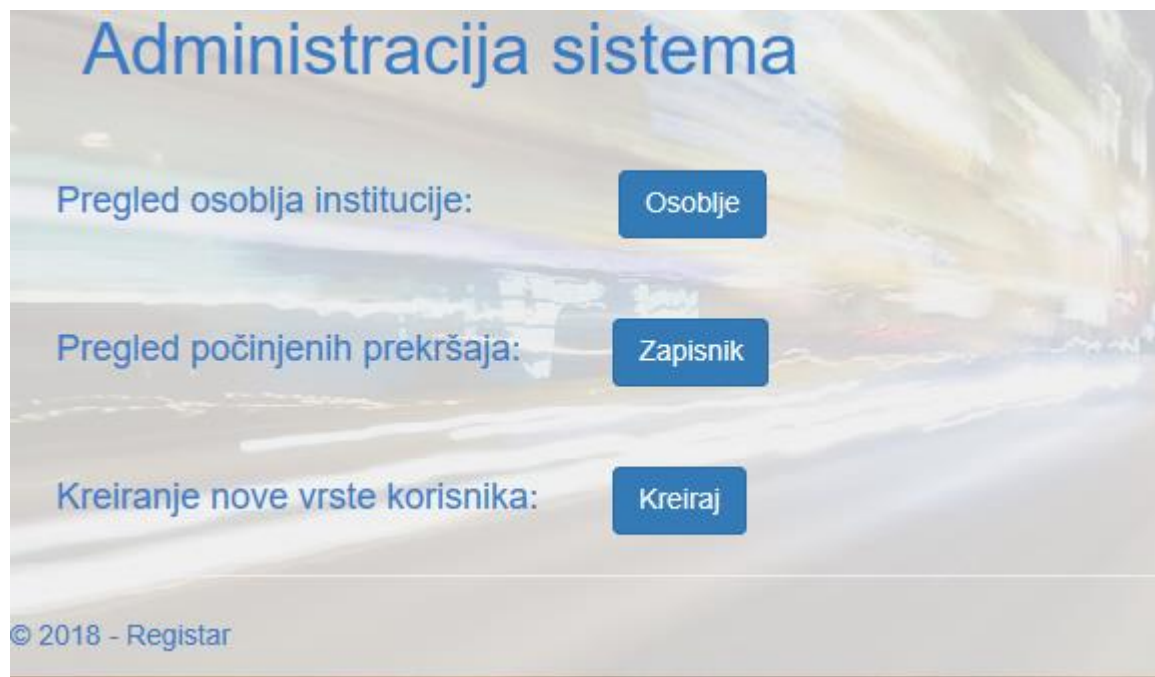
UserName

Password

☐ Zapamti me

Nemate korisnički račun? Kliknite na registraciju.

Nakon registracije i prijave na sistem, bitno je auditiranje. Sistem razlikuje korisnika tipa admin od ostalih i daje mu više mogućnosti.



Korisnik tipa admin može kreirati novi tip korisnika.



Controller-i PolicajacController.cs i ZapisController.cs omogućuju prikaz i manipulaciju sa podacima iz tabela Policajac i Zapis. Moguće je pregledati zapis, izmijeniti ga, izbirisati ili eventualno dodati novi.

Datum prekršaja	Počinioc	Nadležni policajac			Opis prekršaja	
10/10/2017 12:00:00 AM	Darko	Kantic	Alma	Dizdar	Oduzimanje prvenstva	Edit Delete
12/11/2017 12:00:00 AM	Zeljko	Dizdar	Elmir	Duranovic	Prolazak kroz crveno	Edit Delete
12/24/2017 12:00:00 AM	Dina	Vasic	Selma	Lekic	Prekoracenje brzine	Edit Delete
12/31/2017 12:00:00 AM	Darko	Kantic	Aleksandar	Mesic	Odsustvo obavezne opreme	Edit Delete
12/31/2017 12:00:00 AM	Ana	Jelic	Semir	Pozder	Alkoholizirano stanje	Edit Delete
2/2/2018 12:00:00 AM	Edin	Mesic	Adnan	Bosnic	Koristenje mobilnog telefona	Edit Delete

PolicajacIme	PolicajacPrezime	
Elmir	Duranovic	Edit Details Delete
Irfan	Nefic	Edit Details Delete
Damir	Hastor	Edit Details Delete
Denis	Sabljić	Edit Details Delete
Aleksandar	Mesic	Edit Details Delete
Ana	Vatic	Edit Details Delete
Adnan	Bosnic	Edit Details Delete
Semir	Pozder	Edit Details Delete
Alma	Dizdar	Edit Details Delete

Controller može, ali ne mora imati uvid u sadržaj tabele za koju je kreiran.

Primjer toga je ZapisController i tabela Zapis. U bazi podataka tabela Zapis sadrži par ID-ijeva, odnosno brojeva, koji nam na prvi pogled ništa puno ne govore, međutim postoji mogućnost napraviti smislene podatke.

	ZapisID	Datum	OsobaID_Osob	PolicajacID_Po...	PrekršajID_Pre...
▶	37	10/10/2017 12:0...	3	9	10
	38	12/11/2017 12:0...	5	1	1
	39	12/24/2017 12:0...	1	10	3
	42	12/31/2017 12:0...	3	5	2
	44	12/31/2017 12:0...	4	8	4
	45	2/2/2018 12:00:...	2	7	6
	46	4/4/2018 12:00:...	1	3	5

Kako tabela sadrži ID za tip Osoba, Policajac i Prekršaj, u funkciji Index unutar ZapisController.cs smo uključili tabele za navedene tipove.

```
public ActionResult Index()
{
    var zapis = db.Zapis.Include(c => c.OsobaID);
    var zapis2 = zapis.Include(c => c.PolicajacID);
    var zapis3 = zapis2.Include(c => c.PrekršajID);
    return View(zapis3.ToList());
}
```

Za odgovarajući ID možemo pozvati ostale pripadne attribute tog tipa.

```
@foreach (var item in Model)
{
    <tr>
        <td> @Html.DisplayFor(modelItem => item.Datum) </td>
        <td> @Html.DisplayFor(modelItem => item.OsobaID.Ime) </td>
        <td> @Html.DisplayFor(modelItem => item.OsobaID.Prezime) </td>
        <td> @Html.DisplayFor(modelItem => item.PolicajacID.PolicajacIme) </td>
        <td> @Html.DisplayFor(modelItem => item.PolicajacID.PolicajacPrezime) </td>
        <td> @Html.DisplayFor(modelItem => item.PrekršajID.OpisPrekršaja) </td>
    </tr>
}
```