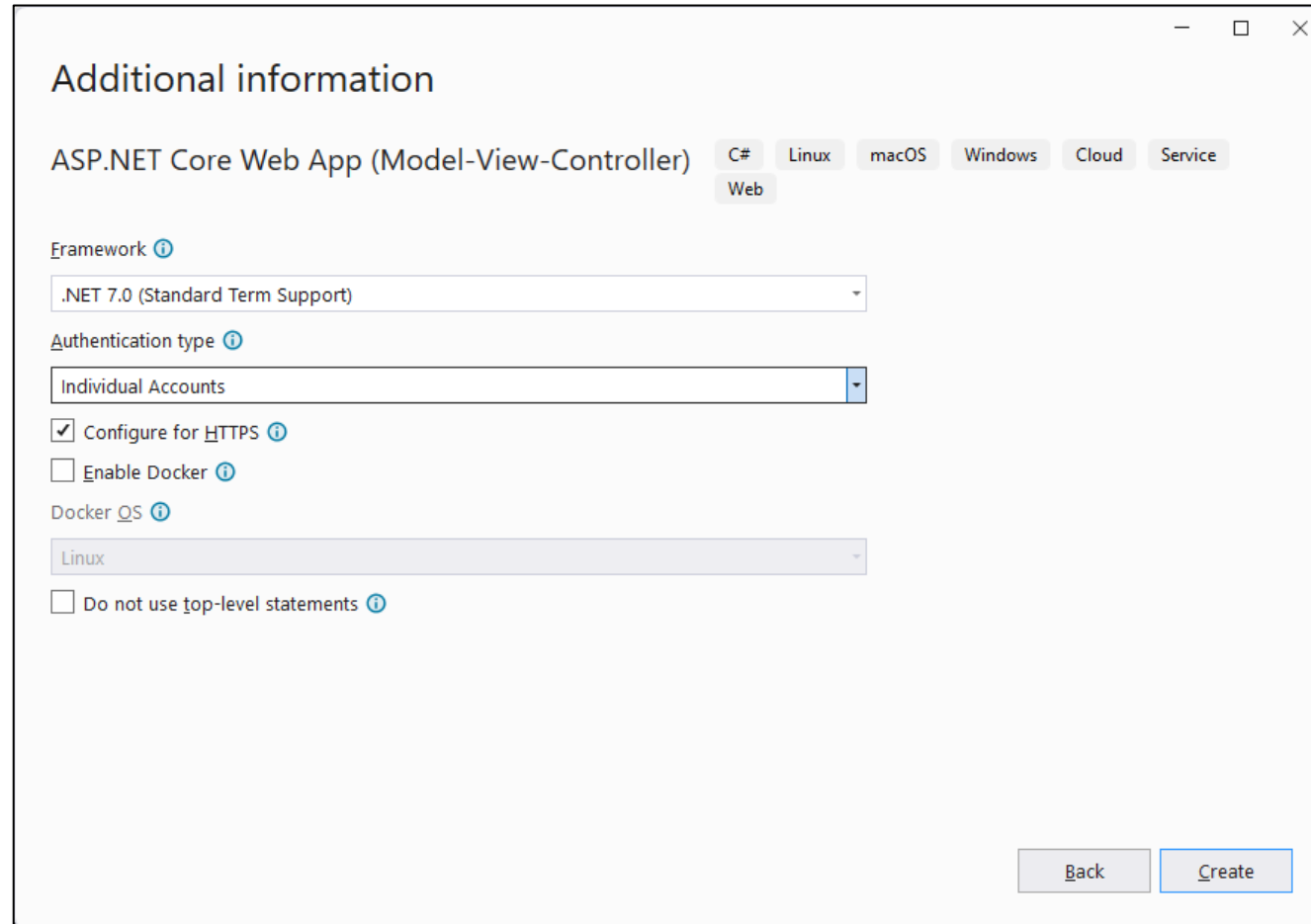


Sigurnost ASP.NET Core MVC web aplikacija

Kreiranje ASP.NET Core web aplikacije



The screenshot shows a dialog box titled "Additional information" for creating an ASP.NET Core Web App. The dialog is titled "ASP.NET Core Web App (Model-View-Controller)" and has several tabs: C#, Linux, macOS, Windows, Cloud, Service, and Web. The "Web" tab is selected. The dialog contains the following options:

- Framework:** A dropdown menu set to ".NET 7.0 (Standard Term Support)".
- Authentication type:** A dropdown menu set to "Individual Accounts".
- Configure for HTTPS:** A checkbox that is checked.
- Enable Docker:** A checkbox that is unchecked.
- Docker OS:** A dropdown menu set to "Linux".
- Do not use top-level statements:** A checkbox that is unchecked.

At the bottom right of the dialog, there are two buttons: "Back" and "Create".

ASP.NET Core Identity sistem

- Visual Studio 2022 implementira Identity sistem za logovanje kao biblioteku baziranu na tehnologiji web pages (ne ASP.NET Core MVC tehnologiji)
- Implementacija koristi code-first pristup pri čemu je definisana migracija na osnovu koje se kreira baza za logovanje
- Koristi se biblioteka **Microsoft.AspNetCore.Identity**;
- Korisnik sistema predstavlja se klasom **IdentityUser** (koja se obično proširuje kada se sistem prilagođava)
- Podrazumevano se korisnik prijavljuje na sistem korišćenjem email adrese (koja je istovremeno njegovo korisničko ime) i lozinke
- Pri registraciji, korisnik navodi samo email adresu i lozinku

Kustomizacija Identity sistema

- Korisnik web aplikacije biće predstavljen klasom `ApplicationUser` koja je izvedena iz sistemski klase **IdentityUser** (biblioteka `Microsoft.AspNetCore.Identity`)
- `DbContext` klasa koja se koristi za komunikaciju sa bazom biće izvedena iz generičke klase **IdentityDbContext**<ApplicationUser> koja pripada biblioteci `Microsoft.AspNetCore.Identity.EntityFrameworkCore`
- Sistemsku klasu **SignInManager**<ApplicationUser> koristi se za kreiranje i uništavanje autentifikacionog kolačića, ili proveru da li korisnik postoji u bazi
- Za kreiranje novih korisnika i dobijanje informacija o njima koristićemo generičku sistemsku klasu **userManager**<ApplicationUser>

Fajl appsettings.json

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=GORAN-HP;Database=aspnet-WebApp09-0305;
      Trusted_Connection=True;MultipleActiveResultSets=true"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

Dodavanje servisa u DI kontejner

```
using Microsoft.AspNetCore.Identity;
using Microsoft.EntityFrameworkCore;
using WebApp09.Data;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
var connectionString =
builder.Configuration.GetConnectionString("DefaultConnection");
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(connectionString));
builder.Services.AddDatabaseDeveloperPageExceptionFilter();

builder.Services.AddDefaultIdentity<IdentityUser>()
    .AddEntityFrameworkStores<ApplicationDbContext>();
builder.Services.AddControllersWithViews();
```

Identity servis postaje raspoloživ kroz DI kontejner

Modifikacija pravila za lozinku i pravila za korisnika

```
builder.Services.Configure<IdentityOptions>(opcije => {  
  
    // Podesavanje lozinke  
    opcije.Password.RequireDigit = false;  
    opcije.Password.RequiredLength = 3;  
    opcije.Password.RequireNonAlphanumeric = false;  
    opcije.Password.RequireUppercase = false;  
    opcije.Password.RequireLowercase = false;  
    opcije.Password.RequiredUniqueChars = 1;  
    // Podesavanje korisnika  
    opcije.User.RequireUniqueEmail = false;  
  
});
```

Modifikacija pravila za lozinku i pravila za korisnika

- Definisana pravila za lozinku definisana u prethodnom slajdu služe samo za lakše testiranje sistema
- Narušena su pravila za kreiranje jake lozinke
- Podrazumevano sistem je podešen da lozinka
 - ima dužinu od najmanje 6 karaktera
 - mora imati veliko slovo
 - mora imati malo slovo
 - mora imati karakter koji nije niti slovo niti broj itd.

Podešavanje protočne obrade zahteva

```
app.UseHttpsRedirection();
app.UseStaticFiles();

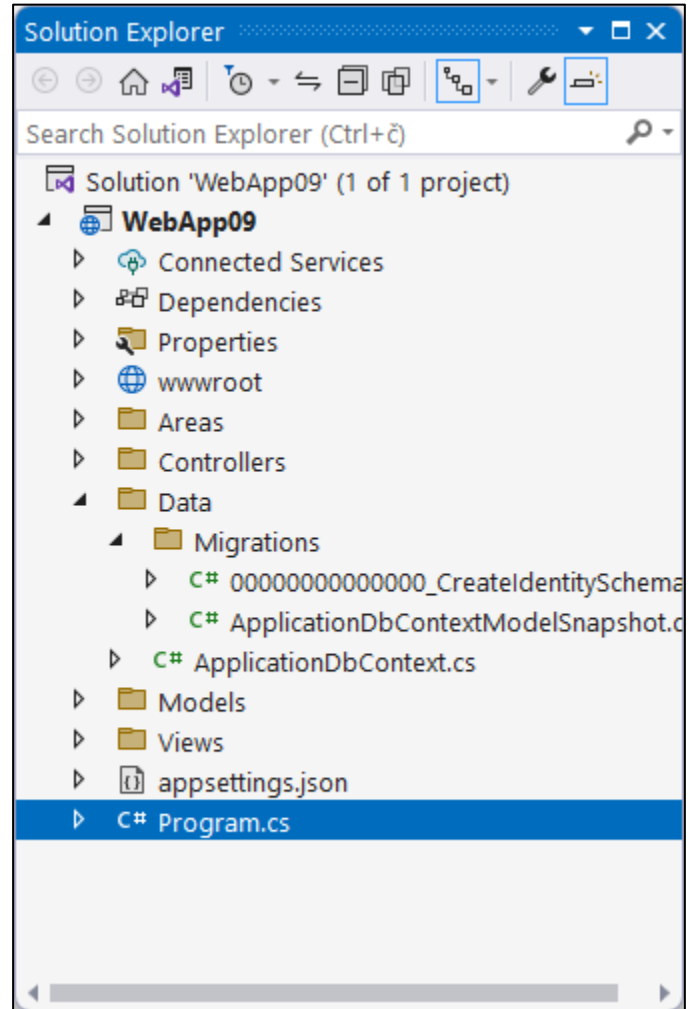
app.UseRouting();
app.UseAuthentication();
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
app.MapRazorPages();

app.Run();
```

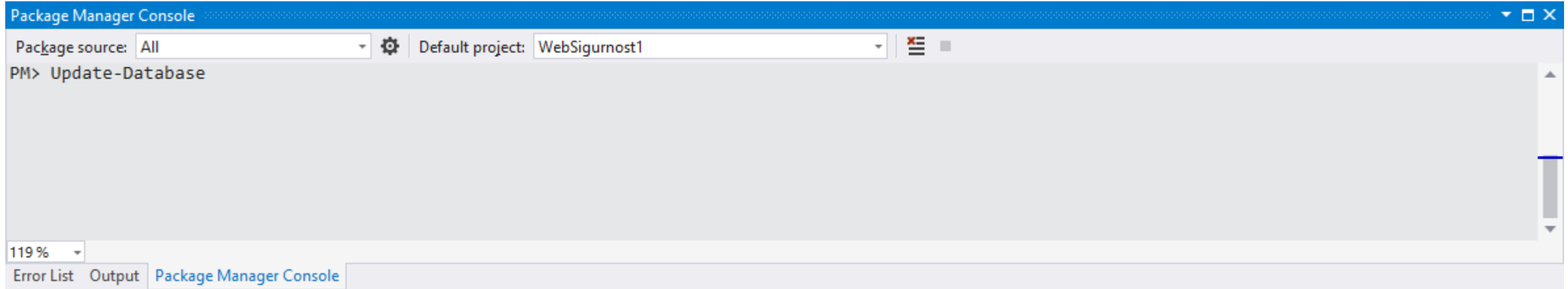
Dodaje se middleware komponenta za autentifikaciju i autorizaciju

Migracije

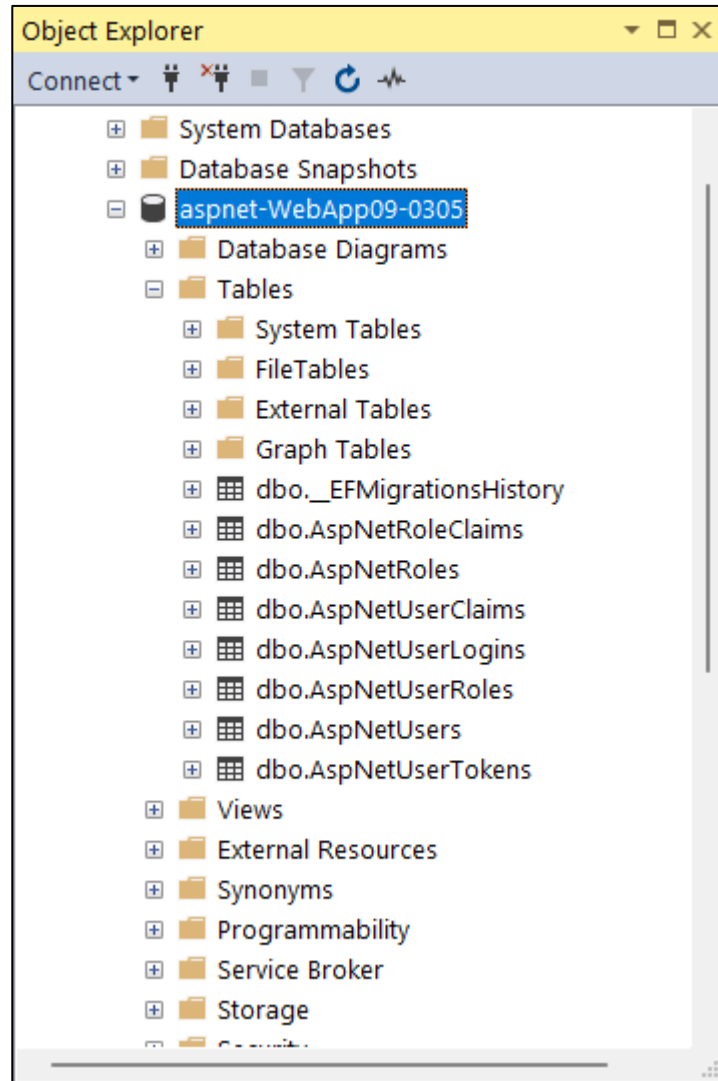


Kreiranje baze za logovanje

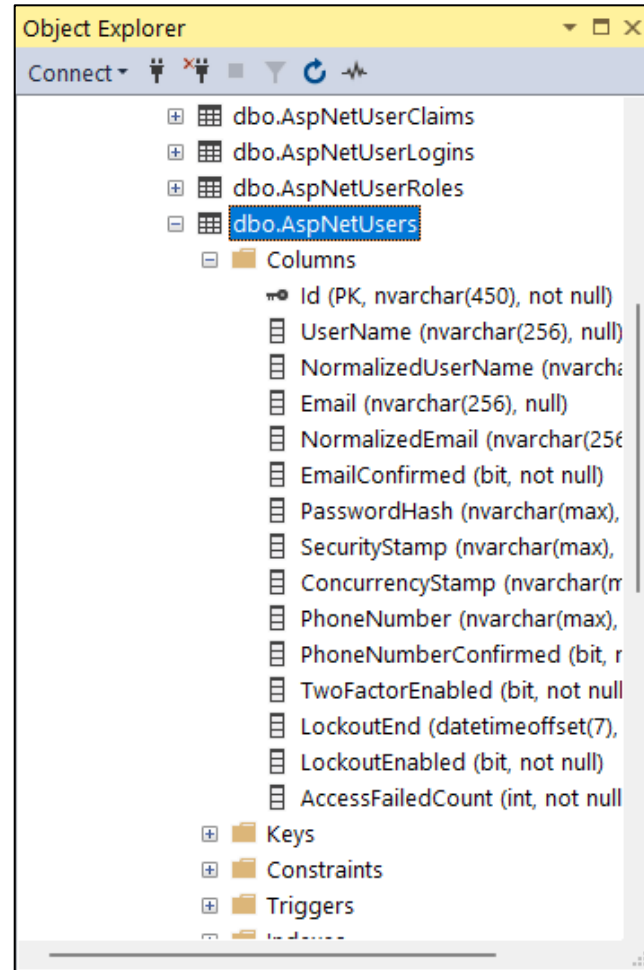
PM> Update-Database



Generisana baza podataka



TabelaAspNetUsers



Klasa ApplicationDbContext folder Data

```
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

namespace WebApp09.Data
{
    public class ApplicationDbContext : IdentityDbContext
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }
    }
}
```

ApplicationDbContext klasa je DbContext klasa koju koristi Identity sistem pri komunikaciji sa bazom podataka. Izvedena je iz klase IdentityDbContext

Klasa ApplicationUser folder Models

```
namespace WebSigurnost.Models
{
    public class ApplicationUser : IdentityUser
    {
        [Required]
        [StringLength(30)]
        public string FirstName { get; set; }

        [Required]
        [StringLength(30)]
        public string LastName { get; set; }
    }
}
```

Ovde se dodaju svi podaci koji su neophodni za registraciju korisnika a ne nalaze se u sistemskoj klasi IdentityUser

Konfigurisanje servisa

```
// Add services to the container.
var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(connectionString));
builder.Services.AddDatabaseDeveloperPageExceptionFilter();

builder.Services.AddDefaultIdentity<ApplicationUser>()
    .AddEntityFrameworkStores<ApplicationDbContext>();
builder.Services.AddControllersWithViews();

builder.Services.Configure<IdentityOptions>(opcije => {

    // Podesavanje lozinke
    opcije.Password.RequireDigit = false;
    opcije.Password.RequiredLength = 3;
    opcije.Password.RequireNonAlphanumeric = false;
    opcije.Password.RequireUppercase = false;
    opcije.Password.RequireLowercase = false;
    opcije.Password.RequiredUniqueChars = 1;
    // Podesavanje korisnika
    opcije.User.RequireUniqueEmail = false;
});
```

Identity servis postaje raspoloživ kroz DI kontejner. Metoda AddDefaultIdentity kao generički tip sadrži klasu ApplicationUser umesto klase IdentityUser .

Isključivanje razor biblioteke za logovanje

```
app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
//app.MapRazorPages();

app.Run();
```

Klasa ApplicationDbContext folder Data

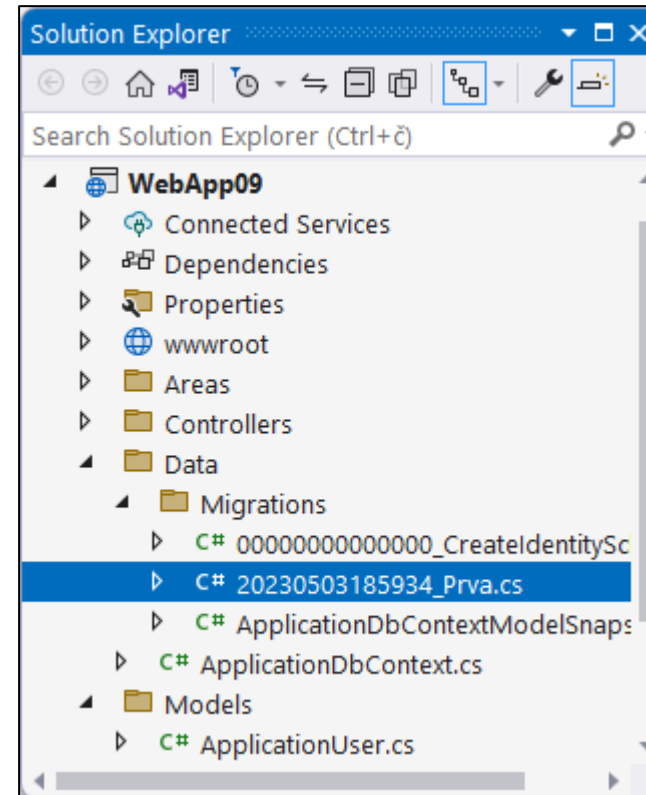
```
namespace WebSigurnost.Data
{
    public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }
    }
}
```

Klasa ApplicationDbContext je izvedena iz generičke klase **IdentityDbContext<ApplicationUser>** umesto iz negeneričke klase IdentityDbContext.

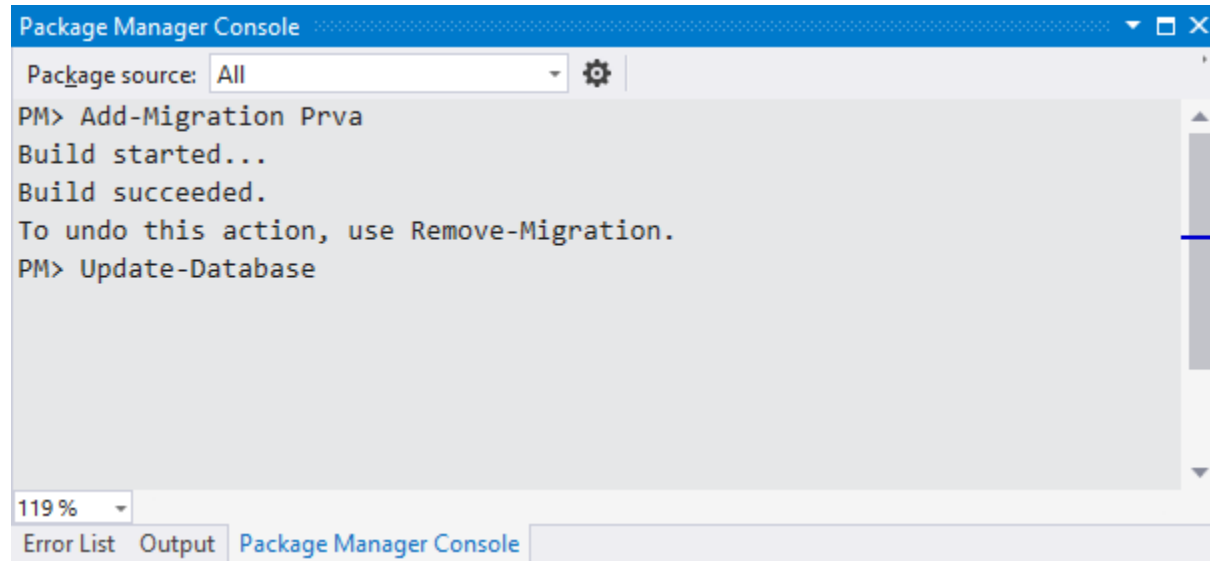
Dodavanje Migracije

```
PM> Add-Migration Prva
```

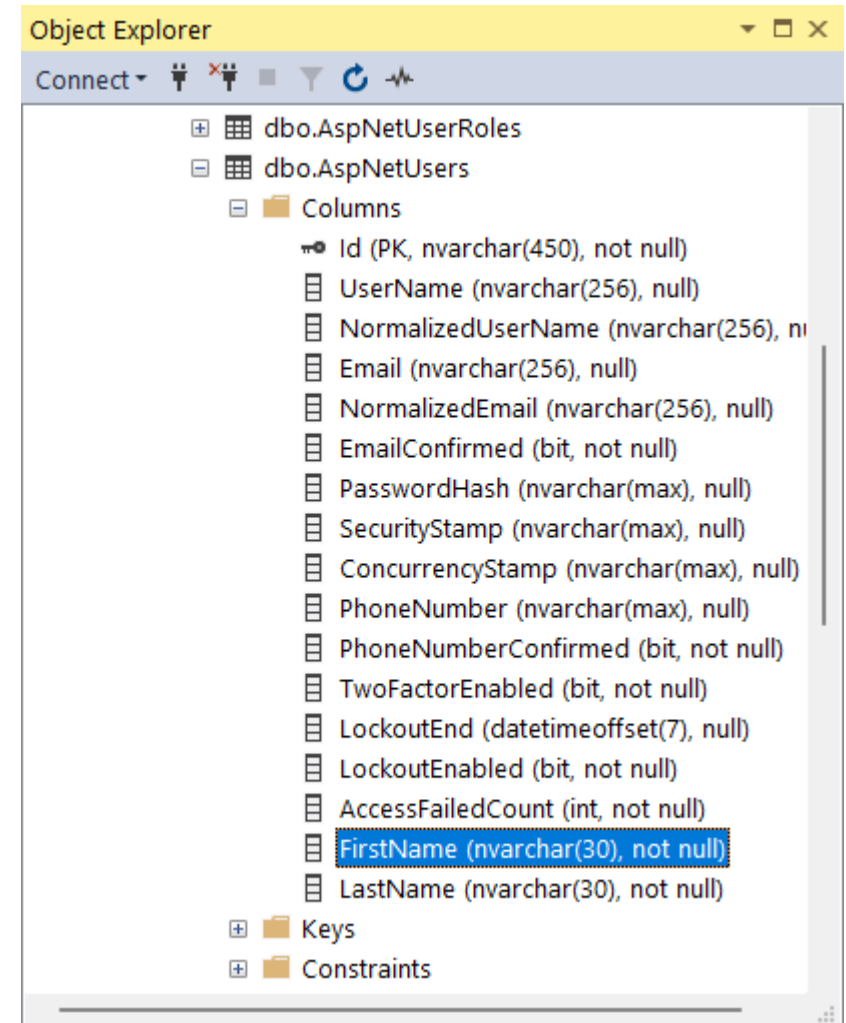
Dodavanjem ove migracije kreira se klasa Prva.cs koja će se koristiti za promenu tabeleAspNetUsers baze za logovanje.



Kreiranje baze za logovanje



```
Package Manager Console
Package source: All
PM> Add-Migration Prva
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> Update-Database
```



```
Object Explorer
Connect
dbo.AspNetUserRoles
dbo.AspNetUsers
  Columns
    Id (PK, nvarchar(450), not null)
    UserName (nvarchar(256), null)
    NormalizedUserName (nvarchar(256), null)
    Email (nvarchar(256), null)
    NormalizedEmail (nvarchar(256), null)
    EmailConfirmed (bit, not null)
    PasswordHash (nvarchar(max), null)
    SecurityStamp (nvarchar(max), null)
    ConcurrencyStamp (nvarchar(max), null)
    PhoneNumber (nvarchar(max), null)
    PhoneNumberConfirmed (bit, not null)
    TwoFactorEnabled (bit, not null)
    LockoutEnd (datetimeoffset(7), null)
    LockoutEnabled (bit, not null)
    AccessFailedCount (int, not null)
    FirstName (nvarchar(30), not null)
    LastName (nvarchar(30), not null)
  Keys
  Constraints
```

Obratiti pažnju da tabela AspNetUsers sada ima dve dodatne kolone FirstName i LastName

Klasa RegisterModel

- Klasa RegisterModel ima svojstva koja se koriste za prikupljanje informacija o novom korisniku
- Objekat klase RegisterModel će se kreirati od strane kontrolera za logovanje na osnovu podataka koji se prosleđuju interfejsom za registraciju

Klasa RegisterModel

```
public class RegisterModel
{
    [Required]
    [StringLength(256)]
    public string UserName { get; set; }

    [Required]
    [StringLength(30)]
    public string FirstName { get; set; }

    [Required]
    [StringLength(30)]
    public string LastName { get; set; }

    [Required(ErrorMessage = "Password is required")]
    [StringLength(100, ErrorMessage = "The password must be at least {2} and at max {1} characters long",
    MinimumLength = 3)]
    [DataType(DataType.Password)]
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
    public string ConfirmPassword { get; set; }
}
```

Kontroler za autentifikaciju

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using WebSigurnost.Models;

namespace WebSigurnost.Controllers
{
    [Authorize]
    public class AccountController : Controller
    {
        private readonly UserManager<ApplicationUser> _userManager;
        private readonly SignInManager<ApplicationUser> _signInManager;

        public AccountController(UserManager<ApplicationUser> userManager,
            SignInManager<ApplicationUser> signInManager)
        {
            _userManager = userManager;
            _signInManager = signInManager;
        }
    }
}
```

Kontroler za autentifikaciju

- Sadrži privatno readonly polje `_signInManager` tipa **`SignInManager<ApplicationUser>`**
- Sadrži privatno readonly polje `_userManager` tipa **`UserManager< ApplicationUser >`**
- Klase `SignInManager` i `userManager` nalaze se u biblioteci `Microsoft.AspNetCore.Identity`
- Klasa **`SignInManager`** služi da proveri postojanje korisnika u bazi i da na osnovu toga napravi autentifikacioni kolačić(metode **`SignInAsync`** i **`PasswordSignInAsync`**)
- Klasa **`SignInManager`** ima metodu za uništavanje autentifikacionog kolačića (metoda `SignInOutAsync`)
- Klasa **`userManager`** služi za komunikaciju sa bazom podataka i ima metodu za kreiranje novog korisnika (metoda `CreateAsync`)

Register GET metoda

```
[AllowAnonymous]  
public ActionResult Register()  
{  
    return View();  
}
```

Metoda poziva pogled koji prikazuje korisnički interfejs za registraciju

Register POST metoda

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser { UserName = model.UserName, FirstName = model.FirstName,
            LastName = model.LastName };

        var result = await _userManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await _signInManager.SignInAsync(user, isPersistent: false);
            return RedirectToAction("Index", "Home");
        }
        foreach (var error in result.Errors)
        {
            ModelState.TryAddModelError(error.Code, error.Description);
        }

        return View(model);
    }
    return View(model);
}
```

Pogled za registraciju -1

```
<div class="row">
  <div class="col-md-4">
    <form asp-action="Register">
      <div asp-validation-summary="All" class="text-danger"></div>

      <div class="form-group">
        <label asp-for="UserName" class="control-label"></label>
        <input asp-for="UserName" class="form-control" />
        <span asp-validation-for="UserName" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="FirstName" class="control-label"></label>
        <input asp-for="FirstName" class="form-control" />
        <span asp-validation-for="FirstName" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="LastName" class="control-label"></label>
        <input asp-for="LastName" class="form-control" />
        <span asp-validation-for="LastName" class="text-danger"></span>
      </div>
    </form>
  </div>
</div>
```

Pogled za registraciju -2

```
<div class="form-group">
    <label asp-for="Password" class="control-label"></label>
    <input asp-for="Password" class="form-control" />
    <span asp-validation-for="Password" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="ConfirmPassword" class="control-label"></label>
    <input asp-for="ConfirmPassword" class="form-control" />
    <span asp-validation-for="ConfirmPassword" class="text-danger"></span>
</div>
<div class="form-group">
    <input type="submit" value="Create" class="btn btn-primary" />
</div>
</form>
</div>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

Klasa LoginModel

```
public class LoginModel
{
    [Required]
    public string UserName { get; set; }

    [Required]
    [DataType(DataType.Password)]
    public string Password { get; set; }

    [Display(Name = "Remember me?")]
    public bool RememberMe { get; set; }
}
```

Objekat klase LoginModel će se kreirati od strane Account kontrolera na osnovu podataka koji se prosleđuju interfejsom za logovanje.

GET metoda za logovanje

```
[AllowAnonymous]
public IActionResult Login(string returnUrl = null)
{
    ViewData["ReturnUrl"] = returnUrl;
    return View();
}
```

Metoda RedirectToLocal

```
private ActionResult RedirectToLocal(string returnUrl)
{
    if (Url.IsLocalUrl(returnUrl))
    {
        return Redirect(returnUrl);
    }
    return RedirectToAction("Index", "Home");
}
```

POST metoda za logovanje

```
[AllowAnonymous]
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Login(LoginModel model, string returnUrl = null)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    var result = await _signInManager.PasswordSignInAsync(model.UserName, model.Password,
model.RememberMe, false);
    if (result.Succeeded)
    {
        return RedirectToLocal(returnUrl);
    }
    else
    {
        ModelState.AddModelError("", "Invalid UserName or Password");
        return View();
    }
}
```


Pogled za Logovanje

```
@model LoginModel
<h1>Login</h1>
<div class="row">
  <div class="col-md-4">
    <form asp-action="Login" asp-route-returnUrl="@ViewData["ReturnUrl"]">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="UserName" class="control-label"></label>
        <input asp-for="UserName" class="form-control" />
        <span asp-validation-for="UserName" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Password" class="control-label"></label>
        <input asp-for="Password" class="form-control" />
        <span asp-validation-for="Password" class="text-danger"></span>
      </div>
      <div class="form-group form-check">
        <label class="form-check-label">
          <input class="form-check-input" asp-for="RememberMe" />
          @Html.DisplayNameFor(model => model.RememberMe)
        </label>
      </div>
      <div class="form-group">
        <input type="submit" value="Log in" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>
@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

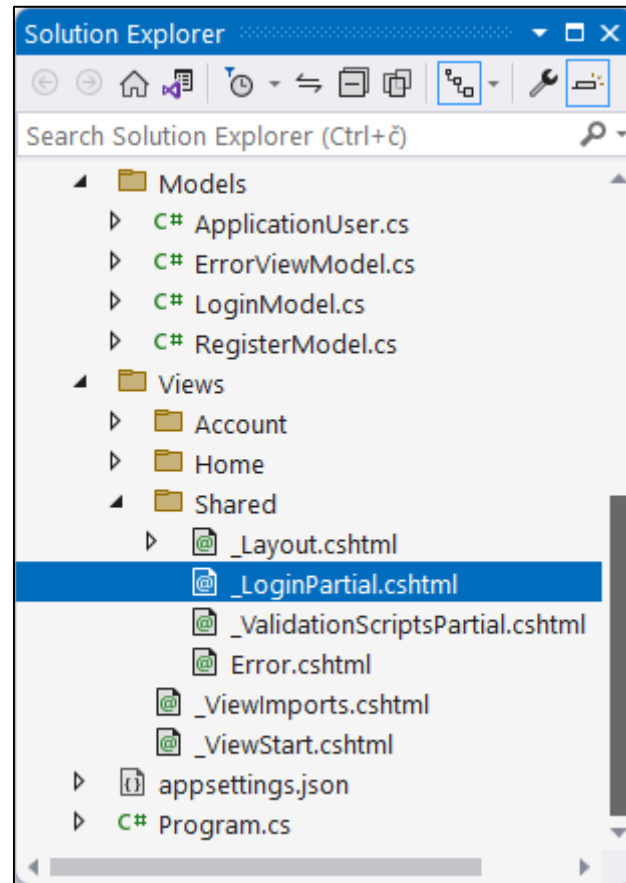
Konfigurisanje autentifikacionog kolačića

Unutar fajla Program.cs konfigurisaćemo autentifikacioni kolačić

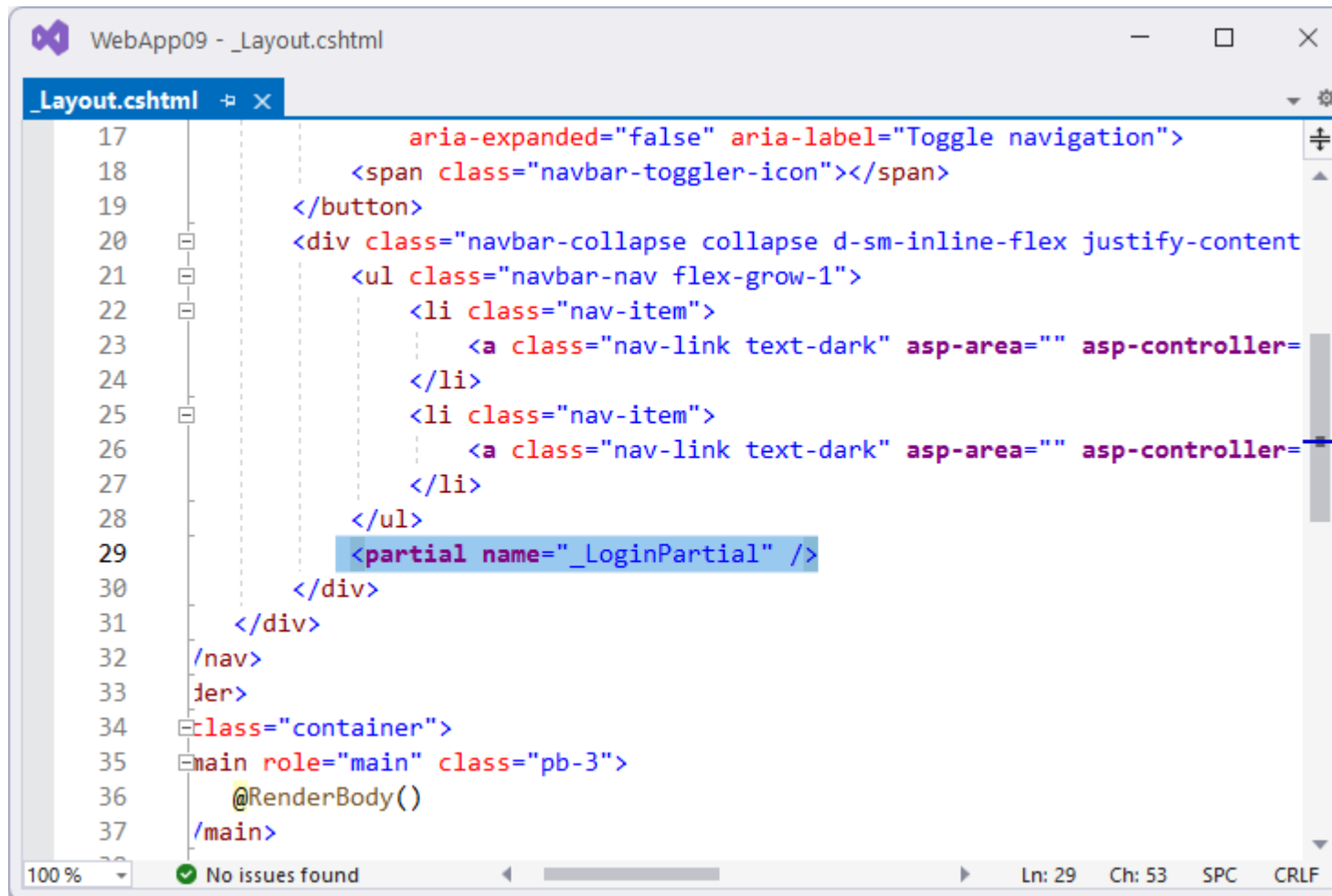
```
builder.Services.ConfigureApplicationCookie(opcije =>
{
    // Cookie settings
    opcije.Cookie.HttpOnly = true;
    opcije.ExpireTimeSpan = TimeSpan.FromMinutes(5);

    opcije.LoginPath = "/Account/Login";
    opcije.AccessDeniedPath = "/Account/AccessDenied";
    opcije.SlidingExpiration = true;
});
```

Parcijalni pogled za logovanje



Ubacivanje parcijalnog pogleda za logovanje u Layout



```
WebApp09 - _Layout.cshtml
_Layout.cshtml
17         aria-expanded="false" aria-label="Toggle navigation">
18         <span class="navbar-toggler-icon"></span>
19     </button>
20     <div class="navbar-collapse collapse d-sm-inline-flex justify-content
21         <ul class="navbar-nav flex-grow-1">
22             <li class="nav-item">
23                 <a class="nav-link text-dark" asp-area="" asp-controller=
24             </li>
25             <li class="nav-item">
26                 <a class="nav-link text-dark" asp-area="" asp-controller=
27             </li>
28         </ul>
29         <partial name="_LoginPartial" />
30     </div>
31 </div>
32 </nav>
33 </div>
34 <div class="container">
35     <main role="main" class="pb-3">
36         @RenderBody()
37     </main>
```

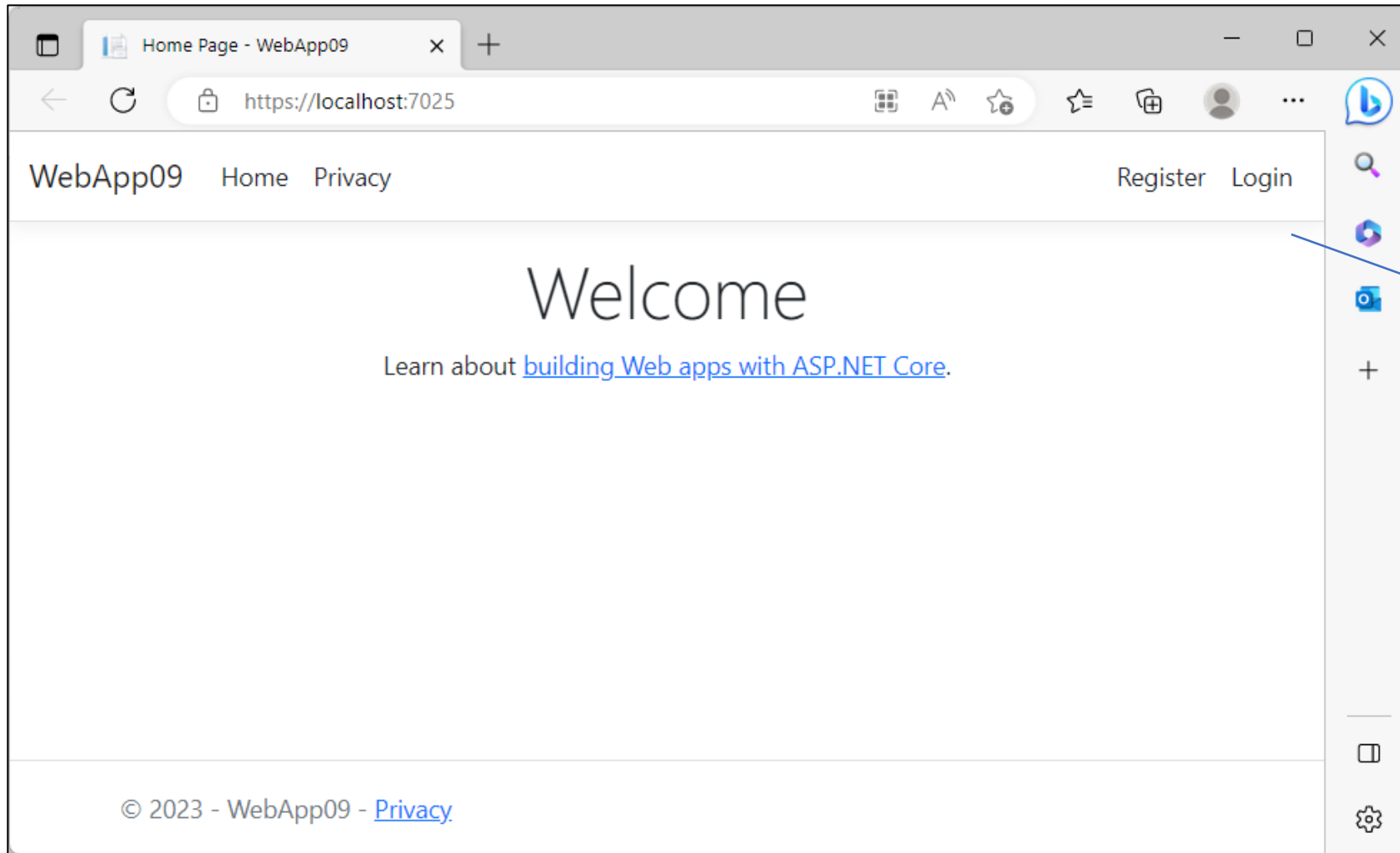
100 % No issues found Ln: 29 Ch: 53 SPC CRLF

Parcijalni pogled za logovanje _LoginPartial.cshtml

```
@using Microsoft.AspNetCore.Identity
@inject SignInManager<ApplicationUser> SignInManager
@inject UserManager<ApplicationUser> UserManager

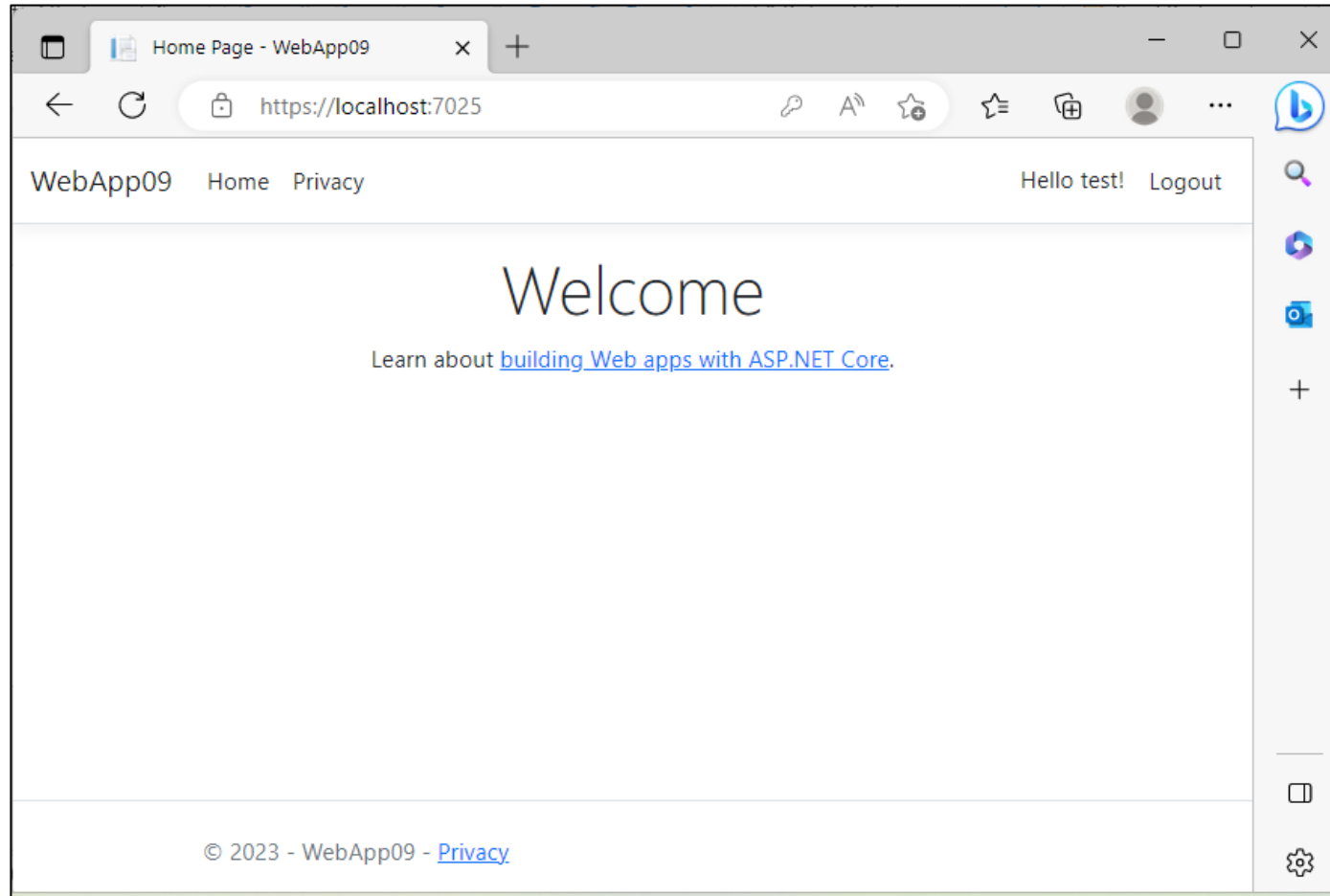
<ul class="navbar-nav">
  @if (SignInManager.IsSignedIn(User))
  {
    <li class="nav-item">
      <a class="nav-link text-dark">Hello @User.Identity?.Name!</a>
    </li>
    <li class="nav-item">
      <form class="form-inline" asp-controller="Account" asp-action="Logout"
        asp-route-returnUrl="@Url.Action("Index", "Home", new { area = "" })">
        <button type="submit" class="nav-link btn btn-link text-dark">Logout</button>
      </form>
    </li>
  }
  else
  {
    <li class="nav-item">
      <a class="nav-link text-dark" asp-controller="Account" asp-action="Register">Register</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" asp-controller="Account" asp-action="Login">Login</a>
    </li>
  }
</ul>
```

Parcijalni pogled za logovanje

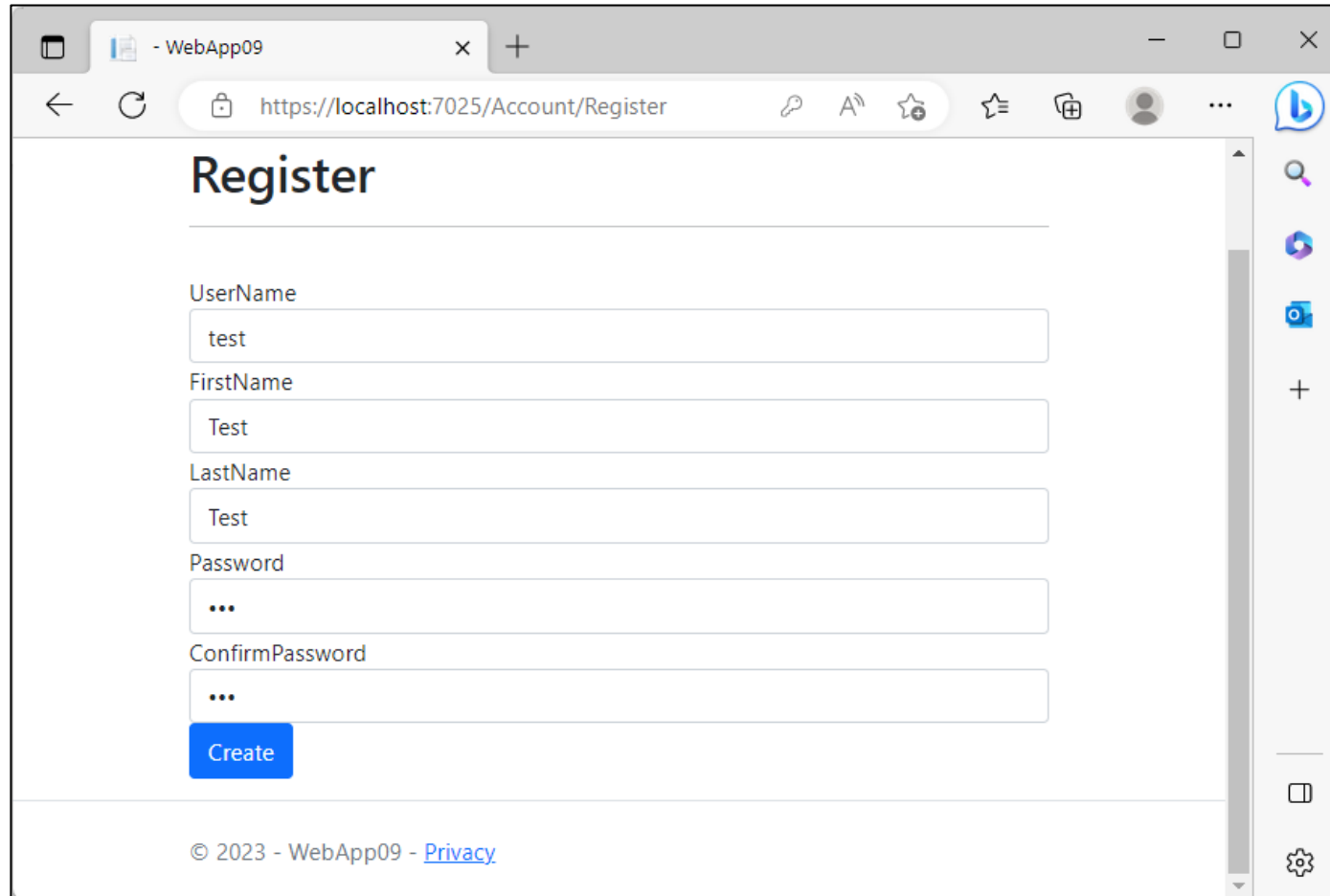


parcijalni pogled

Parcijalni pogled za logovanje

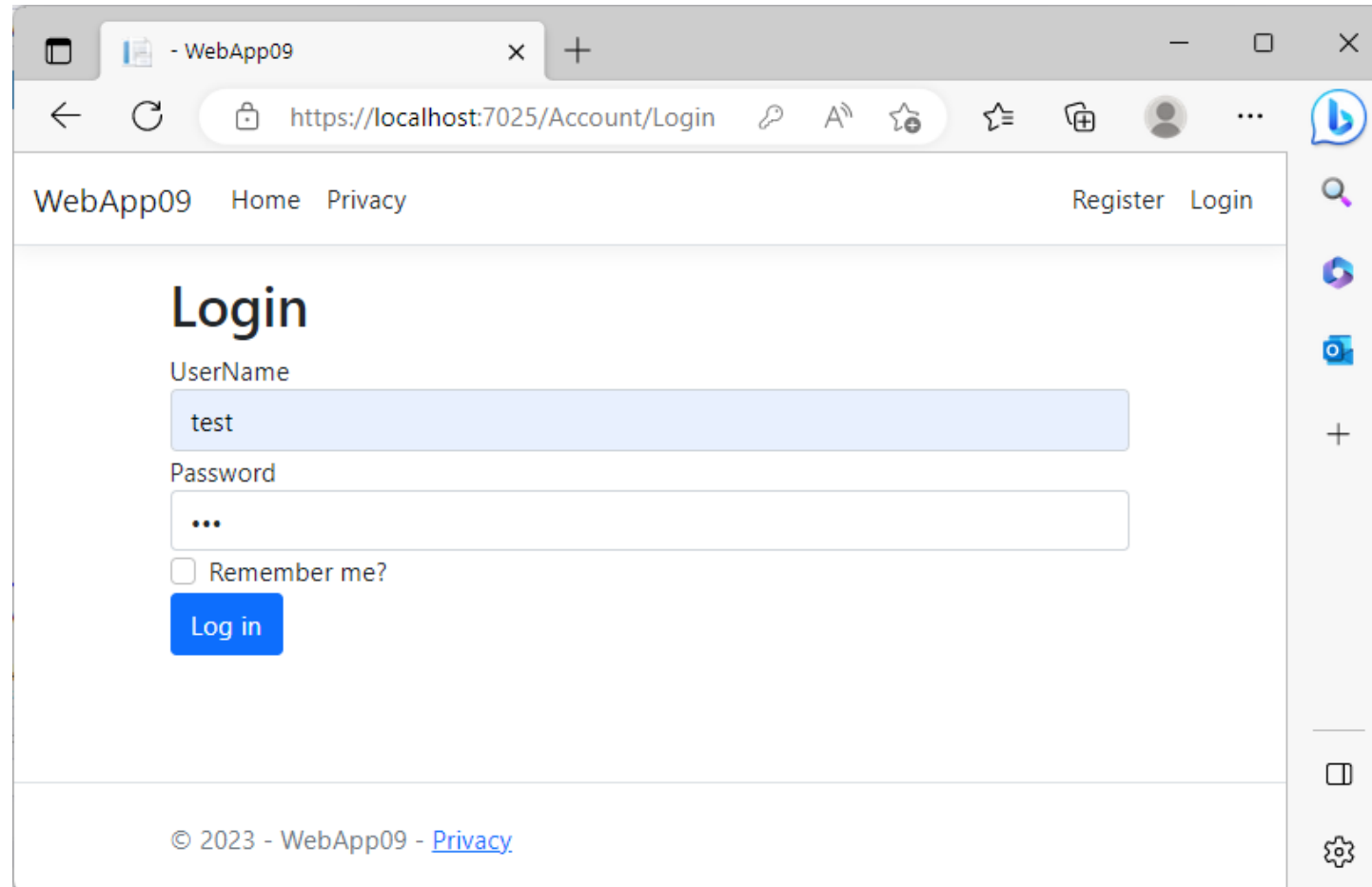


Registracija novog korisnika



The screenshot shows a web browser window with a single tab titled '- WebApp09'. The address bar displays 'https://localhost:7025/Account/Register'. The page content includes a heading 'Register' followed by a horizontal line. Below this are five input fields: 'UserName' (containing 'test'), 'FirstName' (containing 'Test'), 'LastName' (containing 'Test'), 'Password' (containing three dots), and 'ConfirmPassword' (containing three dots). A blue 'Create' button is positioned below the 'ConfirmPassword' field. At the bottom of the page, there is a footer: '© 2023 - WebApp09 - [Privacy](#)'.

Logovanje na aplikaciju



The screenshot shows a web browser window with the following elements:

- Browser Tab:** - WebApp09
- Address Bar:** <https://localhost:7025/Account/Login>
- Page Header:** WebApp09 Home Privacy Register Login
- Section Title:** Login
- Form Fields:**
 - UserName:** A text input field containing the value "test".
 - Password:** A password input field with masked characters "..." and a toggle icon.
- Checkbox:** Remember me?
- Button:** A blue button labeled "Log in".
- Footer:** © 2023 - WebApp09 - [Privacy](#)

Autentifikacioni kolačić

WebApp09

Welcome

Learn about [building Web apps with ASP.NET Core.](#)

© 2023 - WebApp09 -

Identify your project's root folder to open source files in Visual Studio Code and sync changes. [Set root folder](#) [Don't show again](#) [Learn more](#)

Application

Manifest

Service Workers

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
 - https://localhost:7025**
 - Trust Tokens
 - Interest Groups
 - Shared Storage

Filter

Only show cookies with an issue

| Name | Value | Do... | Path | Ex... | Size | Http... | Se... | Sa... | Par... | P... |
|------------------------------------|----------------|--------|------|-------|------|---------|-------|--------|--------|-------|
| .AspNetCore.Identity.Application | CfDJ8IXkRQS... | loc... | / | Se... | 656 | ✓ | ✓ | Lax | | Me... |
| .AspNetCore.Antiforgery.gtOHh77... | CfDJ8IXkRQS... | loc... | / | Se... | 190 | ✓ | | Strict | | Me... |

Cookie Value Show URL decoded

CfDJ8IXkRQSWb91NvnaXz4PxXBiakD9psUmCbeIRF84YJVJeBMGjy9VbU9zisDLzVkl_wlqX6uzA-EqVpJg7MtPJHvxR9Weo95vNwlpC6uGDTnLRYUVsV3w2fFRugUrjn1M8y_SdMqi0etlupTqtMCSmYjBVnZ5eiEQjxN8IVLU8x89ba12hnVIQ6_IgtQ9Fd9tfSKHQf9OxdB8Q2D8IOO6akVCmlOHamrM8Ov3859jE_FoZ3JBs6M7MJRLjmBf6u0zJwpWOAfGB02wimuzvpDWLGjnK3v79Bq6o3J3uSrlcZjhHRCFS1ITIM2Ho3j74AdAxHoRxbJWt4RzuMiMrA6ahyh-dneMhtRxTqOkoKJ1T2wgn0PrW2us8qwkWRssM0uEx803oJYN2aCNmOejD9i9Sih01eJwDtl26gxrot64k4ym3AwExBLS3jjHKmCT5iJl46OL7W8csgYm3QdNT9wlqymHIk0oX_UOI4cYn9c-iG2Wqsy7JiMz0PQXZ6IEZTecdryBAY1mU49Z5em_xm-1tSq2jEgodC9E7GjtUvtdrfCLqit2trT074AT1i0KI2jRUIZ_J-nlXf6hh7pR9UIeM3Acf39UIEuyEeTYU01J8

Uništavanje autentifikacionog kolačića

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Logout()
{
    await _signInManager.SignOutAsync();

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

Metoda i pogled AccessDenied

```
public IActionResult AccessDenied()
{
    return View();
}
```

```
@{
    ViewData["Title"] = "Access denied";
}

<header>
    <h1 class="text-danger">@ViewData["Title"]</h1>
    <p class="text-danger">You do not have access to this
resource.</p>
</header>
<a asp-controller="Home" asp-action="Index">Index</a>
```

Atributske klase

- Nalaze se u biblioteci `Microsoft.AspNetCore.Authorization`
- Klasa `[Authorize]` služi za opisivanje kontrolera ili akcione metode kontrolera čime se dozvoljava pristup samo autorizovanim korisnicima
- Klasa `[AllowAnonymous]` atributska klasa se obično koristi u autorizovanim kontrolerima da bi se dozvolio pristup nekoj akcionoj metodi anonimnim korisnicima

Model klasa Osoba

```
[Table("Osoba")]
public class Osoba
{
    public int OsobaId { get; set; }

    [Required()]
    [StringLength(30)]
    public string Ime { get; set; }

    [Required()]
    [StringLength(30)]
    public string Prezime { get; set; }

    [Required()]
    [StringLength(20)]
    public string Telefon { get; set; }
}
```

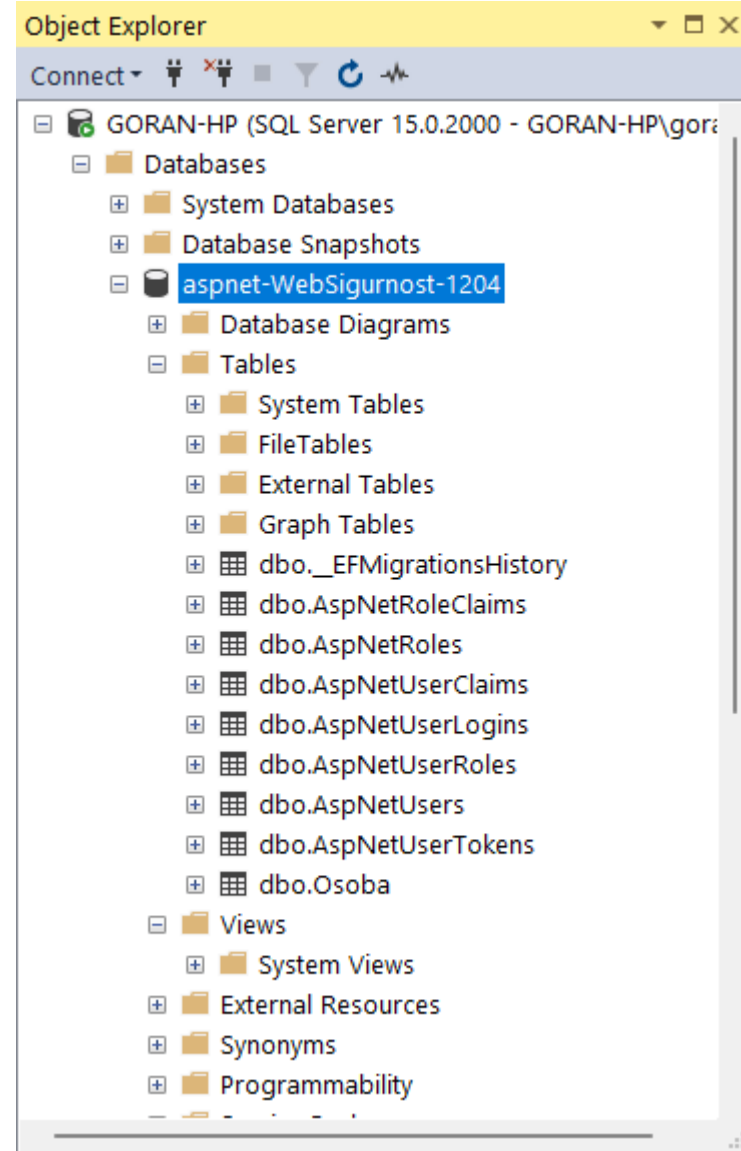
DbContext klasa

```
namespace WebApp09.Data
{
    public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
            : base(options)
        {
        }
        public DbSet<Osoba> Osobe { get; set; }
    }
}
```

Migracije baze

PM> Add-Migration Druga

PM> Update-Database



Kreiranje kontrolera

Add MVC Controller with views, using Entity Framework

Model class: Osoba (WebApp09.Models)

DbContext class: ApplicationDbContext (WebApp09.Data) +

Database provider: SQL Server

Views

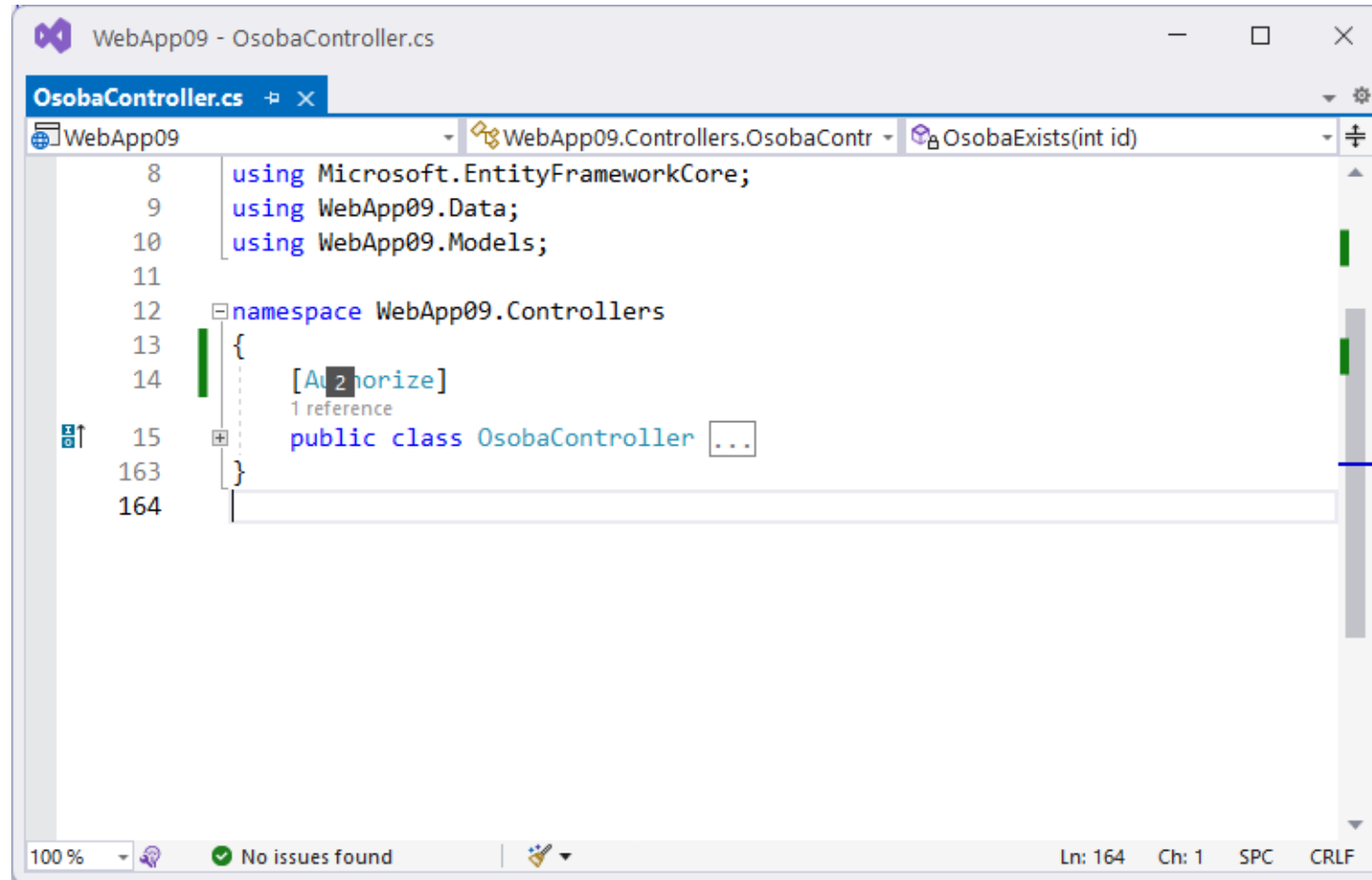
- Generate views
- Reference script libraries
- Use a layout page

...
(Leave empty if it is set in a Razor _viewstart file)

Controller name: OsobaController

Add Cancel

Autorizacija kontrolera



```
8 using Microsoft.EntityFrameworkCore;
9 using WebApp09.Data;
10 using WebApp09.Models;
11
12 namespace WebApp09.Controllers
13 {
14     [Authorize]
15     public class OsobaController
163 }
164
```

The screenshot shows a Visual Studio window titled "WebApp09 - OsobaController.cs". The code editor displays the following C# code:

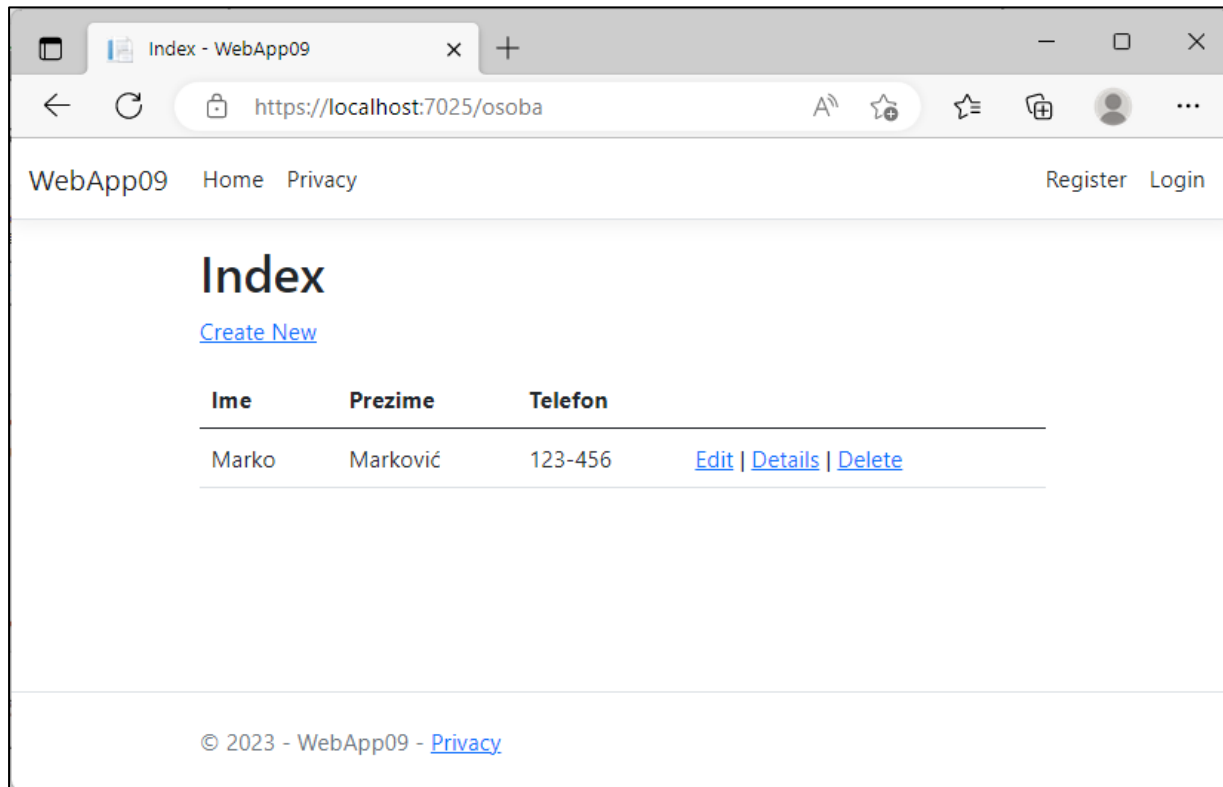
```
8 using Microsoft.EntityFrameworkCore;
9 using WebApp09.Data;
10 using WebApp09.Models;
11
12 namespace WebApp09.Controllers
13 {
14     [Authorize]
15     public class OsobaController
163 }
164
```

The code includes using statements for `Microsoft.EntityFrameworkCore`, `WebApp09.Data`, and `WebApp09.Models`. It defines a namespace `WebApp09.Controllers` containing a public class `OsobaController`. The `[Authorize]` attribute is applied to the class. The status bar at the bottom indicates "100%", "No issues found", and "Ln: 164 Ch: 1 SPC CRLF".

Dozvoljen poziv Index akcije anonimnom(neregistrovanom) korisniku

```
[AllowAnonymous]  
public async Task<IActionResult> Index()  
{  
    return View(await _context.Osobe.ToListAsync());  
}
```

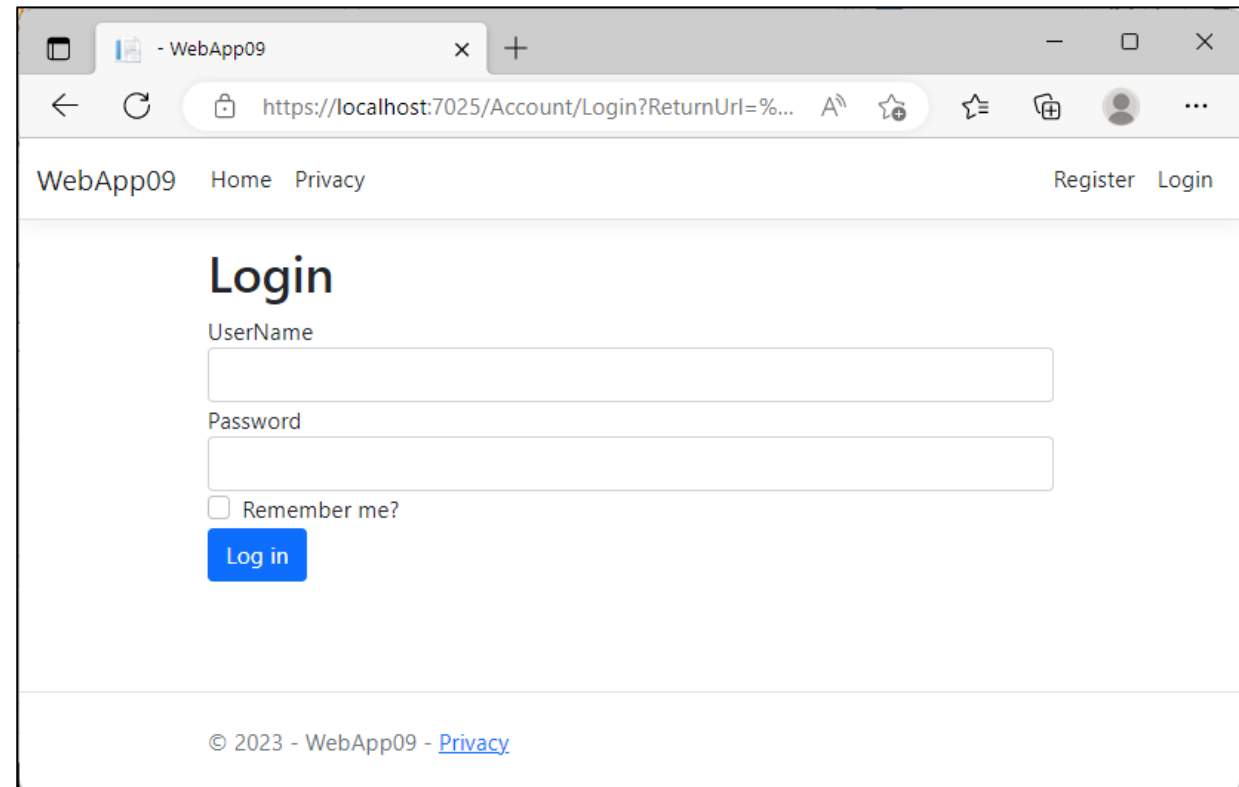
Pokušaj editovanja podataka od strane anonimnog korisnika



The screenshot shows the 'Index' page of a web application. The browser address bar indicates the URL is <https://localhost:7025/osoba>. The page features a navigation menu with 'Home' and 'Privacy' links, and 'Register' and 'Login' buttons. The main content area displays a table with the following data:

| Ime | Prezime | Telefon | |
|-------|----------|---------|---|
| Marko | Marković | 123-456 | Edit Details Delete |

At the top of the table area, there is a link for [Create New](#). The footer contains the text '© 2023 - WebApp09 - [Privacy](#)'.



The screenshot shows the 'Login' page of the web application. The browser address bar indicates the URL is <https://localhost:7025/Account/Login?ReturnUrl=%...>. The page features a navigation menu with 'Home' and 'Privacy' links, and 'Register' and 'Login' buttons. The main content area contains a login form with the following fields and elements:

- UserName**: A text input field.
- Password**: A text input field.
- Remember me?
- Log in**: A blue button.

The footer contains the text '© 2023 - WebApp09 - [Privacy](#)'.

Pitanje 1

Atributska klasa za opis akcione metode kontrolera pomoću koje se u autorizovanim kontrolerima dozvoljava pristup toj akcionoj metodi anonimnim korisnicima je:

- a. [Unauthorized]
- b. [AllowAnonymous]
- c. [NoAuthorize]

Odgovor: b

Pitanje 2

Na koji način se zabranjuje neregistrovanim korisnicima ASP.NET Core MVC web aplikacije, da pozivaju akcione metode nekog kontrolera?

- a. Kontroler se opisuje atributskom klasom [AllowAnonymous]
- b. Kontroler se opisuje atributskom klasom [Authenticate]
- c. Kontroler se opisuje atributskom klasom [Authorize]

Odgovor: c

Pitanje 3

ASP.NET Core Identity sistem za logovanje za rad sa logovanim korisnikom aplikacije kao baznu klasu koristi sledeću klasu:

- a. IdentityUser
- b. User
- c. LoggedUser

Odgovor: a

Pitanje 4

Registrowanje identity servisa kod asp.net core web aplikacije vrši se unutar Program.cs fajla pozivom metode

- a. `services.AddDefaultIdentity<IdentityUser>()`
- b. `services.AddAutentitication<IdentityUser>()`
- c. `services.Add<IdentityUser>()`

Odgovor: a

Pitanje 5

Zaokruži opciju gde je ispravno izvršeno konfigurisanje middleware komponenti za autentifikaciju i autorizaciju kod ASP.NET Core MVC web aplikacije:

- a. `app.UseAuthentication();`
`app.UseAuthorization();`
- b. `services.Authorize();`
- c. `app.UseAuthenticationAuthorization();`

Odgovor: a

Pitanje 6

Kustomizovani Identity system korisnika predstavlja klasom **ApplicationUser**. Za kreiranje novog korisnika ASP.NET Core MVC web aplikacije može se koristiti objekat generičke klase?

- a. UserManager<ApplicationUser>
- b. SignInManager<ApplicationUser>
- c. User <ApplicationUser>

Odgovor: a

Pitanje 7

Konfigurisanje autentifikacionog kolačića vrši se prosleđivanjem anonimne metode sledećoj metodi?

- a. `builder.Services.CookieOptions()`
- b. `builder.Services.Cookie()`
- c. `builder.Services.ConfigureApplicationCookie()`

Odgovor: c

Pitanje 8

Metoda klase UserManager za kreiranje novog korisnika u bazi za logovanje je?

- a. `NewUserAsync()`
- b. `CreateAsync()`
- c. `CreateUser()`

Odgovor: b

Pitanje 9

Kreiranje autentifikacionog kolačića na osnovu podataka iz baze za logovanje vrši se korišćenjem?

- a. Metode SignInAsync() objekta klase SignInManager
- b. Metode SignInAsync() objekta klase UserManager
- c. Metode CreateCookie() objekta klase SignInManager

Odgovor: a

Pitanje 10

Uništavanje autentifikacionog kolačića kod ASP.NET Core MVC web aplikacije

- a. Metode SignOutAsync objekta klase SignInManager
- b. Metode SignOutAsync objekta klase UserManager
- c. Metode SignOut objekta klase Cookie

Odgovor: a