

Akademija tehničko-vaspitačkih strukovnih studija

Copyright © 2022 by Zoran Veličković



.NET tehnologije

Prof. dr Zoran Veličković, dipl. inž. el.

2022/23.

Prof. dr Zoran Veličković, dipl. inž. el.

.NET tehnologije



MS Web tehnologije - ASP.NET

(13)



Sadržaj

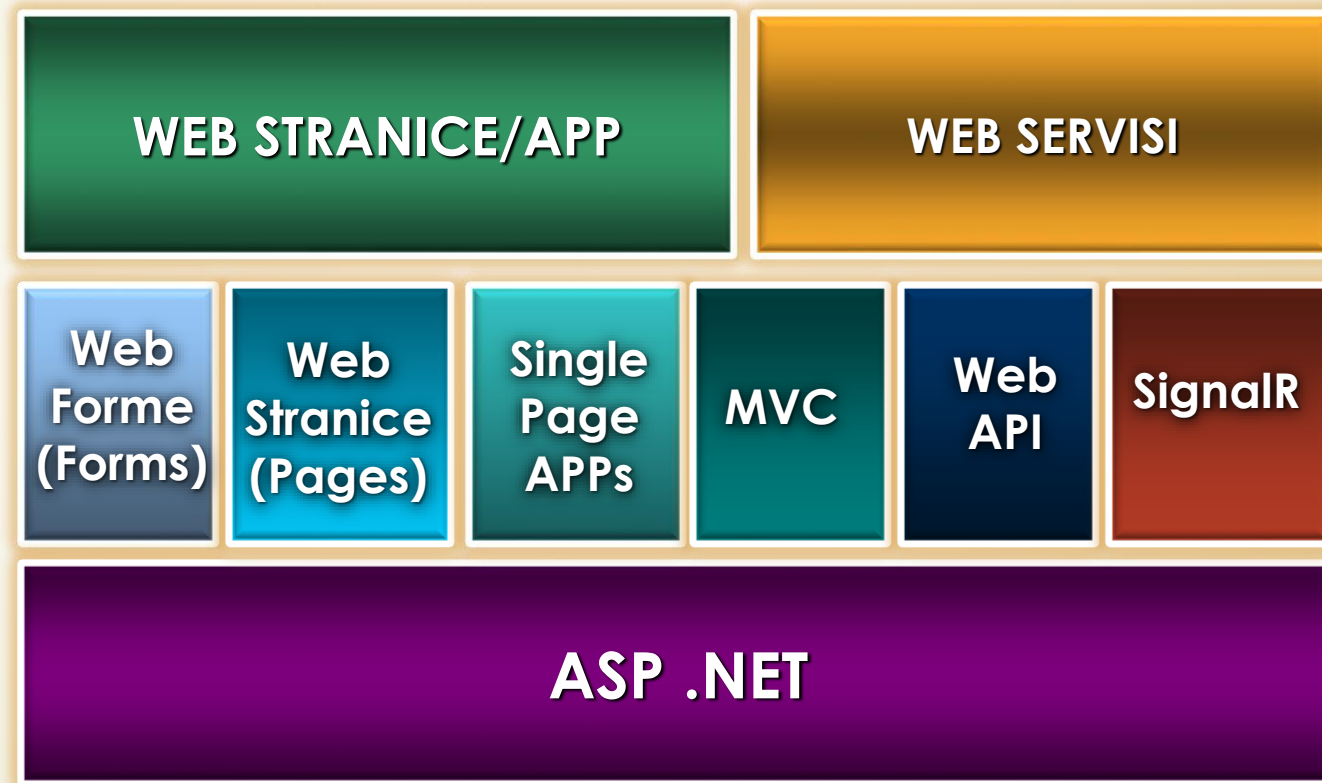


- ▶ PREGLED MS WEB TEHNOLOGIJA
 - ▶ ASP .NET Web tehnologije
 - ▶ Arhitektura ASP .NET-a
- ▶ OSNOVE ASP .NET-A
 - ▶ Moderan Web alat
 - ▶ Multiplatformski .NET Core i AoT
 - ▶ Open source
 - ▶ Visual Studio Code
- ▶ ASP .NET WEB FORMS
 - ▶ Serverske kontrole, atribut runat
 - ▶ Web toolbox
 - ▶ RAZOR sintaksa
- ▶ ASP .NET MVC
 - ▶ Istorija MVC-a
 - ▶ Zahtev za procesiranjem MVC-u
 - ▶ MVC kontroler
 - ▶ MVC View
 - ▶ MVC Model
- ▶ ŠABLONI WEB APP U VS 2019

Pregled MS Web tehnologija

- **ASP .NET** je **BAZNA** Microsoft-ova tehnologija **ORIJENTISANA** ka **WEB-u** zasnovana na **OTVORENIM** standardima kao što su: **XML**, **XHTML**, **HTTP**, **SOAP**, **JSON** i **WSDL**.
- Iako je Microsoft najčešće razvijao **SOPSTVENE** programske tehnologije, uvideo je značaj podrške **OTVORENIH PROGRAMERSKIH TEHNOLOGIJA** u svojim proizvodima!
- **ASP .NET** je **BAZNA PLATFORMA** na kojoj funkcionišu **SVI TIPOVI** ASP .NET aplikacija.
- Microsoft je razvio puno tehnologija za izgradnju **Web APLIKACIJA** na platformi ASP .NET-a, a najznačajnije su:
 - **HTML SERVERSKE KONTROLE** (engl. *HTML Server Controls*), najstarija MS Web tehnologija zasnovana na atributu "**runat**" u HTML oznakama,
 - **ASP .NET Web Forms** je originalni programski model sličan **WINDOWS FORMAMA** – zbog zastarelosti se više ne koristi za razvoj novih Web stranica,
 - **ASP .NET Web Pages**, polako odlazi u istoriju,
 - **ASP .NET MVC**, aktuelna Web tehnologija
 - **BLAZOR WebAssembly**, najnovija MS tehnologija.

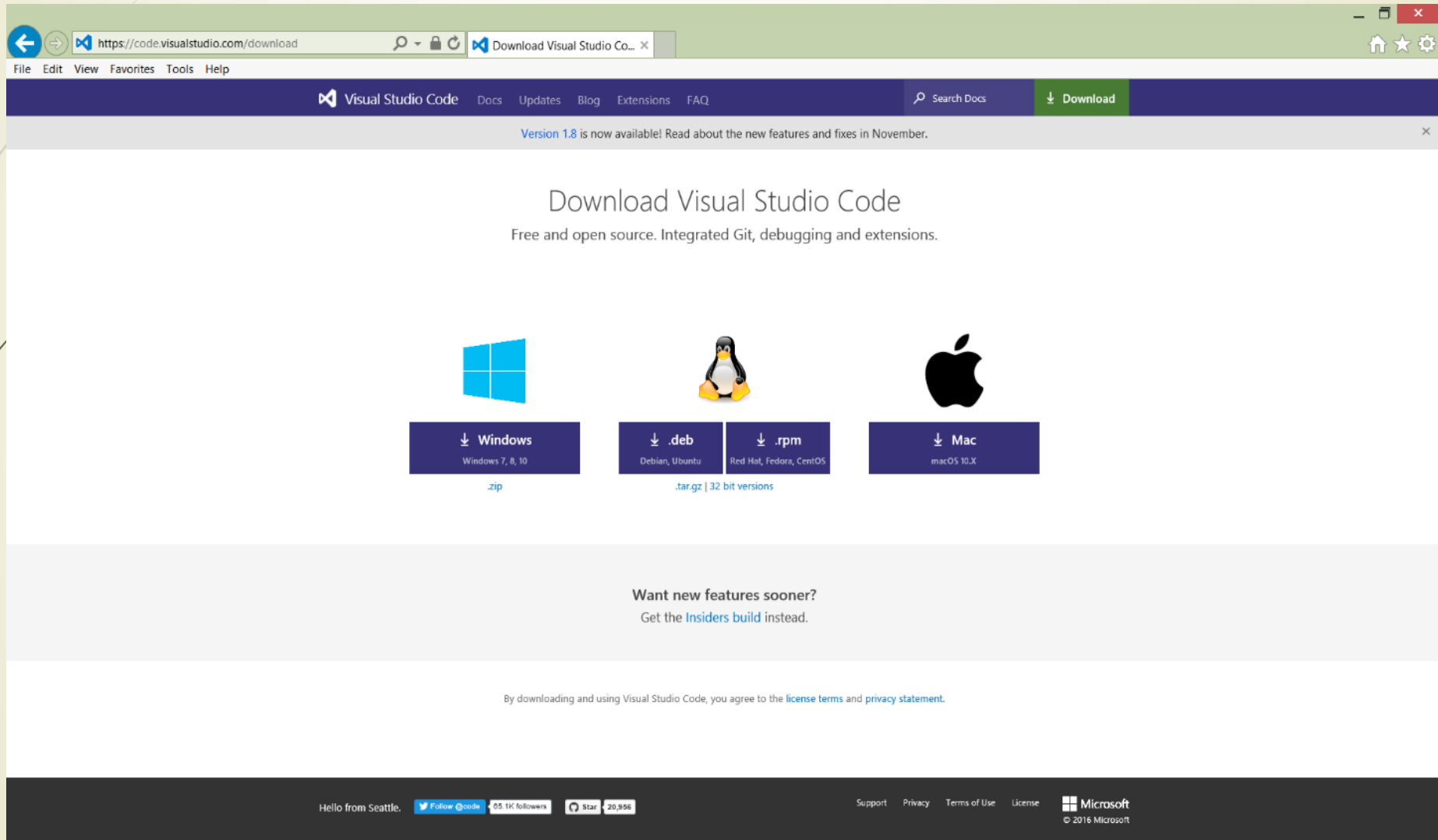
Arhitektura ASP .NET-a



Osnove ASP .NET-a

- **ASP .NET** je Web platforma koja obezbeđuje sve neophodne servise za izgradnju **SERVERSKI ORIJENTISANIH WEB APLIKACIJA**.
- ASP .NET je izgrađen na **.NET FRAMEWORK**-u (sada i na .NET Core-u), tako da su **SVI SERVISI** .NET-a **DOSTUPNI** ASP.NET aplikacijama.
- **WEB APLIKACIJE** mogu biti pisane na **BILO KOM** programskom jeziku koji je kompatibilan sa **CLR**-om (engl. *Common Language Runtime*), što uključuje Visual Basic .NET kao i C#.
- Iako nije neophodno, za kreiranje ASP .NET Web aplikacija najčešće se koristi **Visual Studio**.
- Programerski alati koje poseduje Visual Studio su **POSEBNO KREIRANI** za razvoj Web aplikacija i poznati su pod imenom **VISUAL WEB DEVELOPER**.
- **VISUAL WEB DEVELOPER EXPRESS** (**COMMUNITY** od VS 2015) je **BESPLATNA RAZVOJNA PLATFORMA** koja uključuje osnovni skup alata za dizajn Web stranica iz Visual Studio-a.
- Godine 2019 MS objavljuje stabilnu verziju **Visual Studio Code**-a koji predstavlja besplatan alat za razvoj (Web) aplikacija.

VS Studio Code



The screenshot shows the Visual Studio Code download page in a web browser. The browser's address bar displays the URL <https://code.visualstudio.com/download>. The page features a dark blue navigation bar with the Visual Studio Code logo, menu items (File, Edit, View, Favorites, Tools, Help), and a 'Download' button. A notification banner at the top states 'Version 1.8 is now available! Read about the new features and fixes in November.' The main content area is white and contains the heading 'Download Visual Studio Code' and the tagline 'Free and open source. Integrated Git, debugging and extensions.' Below this, three operating system icons are shown: Windows, Linux (Tux penguin), and Mac. Each icon has a corresponding download button with a dropdown arrow. The Windows button is labeled 'Windows' and 'Windows 7, 8, 10' with a '.zip' link below. The Linux button is split into two sections: '.deb' for 'Debian, Ubuntu' and '.rpm' for 'Red Hat, Fedora, CentOS' with a '.tar.gz | 32 bit versions' link below. The Mac button is labeled 'Mac' and 'macOS 10.X'. A section below the download buttons asks 'Want new features sooner?' and suggests getting the 'Insiders build' instead. At the bottom, a footer contains social media links for Twitter (@code, 55.1K followers) and GitHub (20,956 stars), along with links for Support, Privacy, Terms of Use, License, and the Microsoft logo with '© 2016 Microsoft'.

https://code.visualstudio.com/download




Visual Studio Code Docs Updates Blog Extensions FAQ

Search Docs Download

Version 1.8 is now available! Read about the new features and fixes in November.

Download Visual Studio Code

Free and open source. Integrated Git, debugging and extensions.

↓ Windows
Windows 7, 8, 10
.zip

↓ .deb
Debian, Ubuntu


↓ .rpm
Red Hat, Fedora, CentOS
.tar.gz | 32 bit versions

↓ Mac
macOS 10.X

Want new features sooner?
Get the [Insiders build](#) instead.

By downloading and using Visual Studio Code, you agree to the [license terms](#) and [privacy statement](#).

Hello from Seattle. [Follow @code](#) 55.1K followers [Star](#) 20,956

Support Privacy Terms of Use License  Microsoft
© 2016 Microsoft

Međuplatformski .NET Core i AoT

- Da bi se omogućio razvoj Web aplikacija u .NET-u za **RAZLIČITE PLATFORME**, MS je promovisao tzv. **.NET Core**.
- **.NET Core** poseduje **MEĐUPLATFORMSKI CLR** koji se naziva **CoreCLR**.
- Modernizovana **BIBLIOTEKA KLASA** u .NET Core-u se naziva **CoreFX**.
- **.NET Native** je nova inicijativa koja **PREVODI** C# kod u **IZVORNE CPU INSTRUKCIJE PRE VREMENA AoT** (engl. *Ahead-Of-Time*) - umesto da se koristi **JIT** (engl. *Just In Time*) kompajler.
- **AoT** kompajler **POBOLJŠAVA BRZINU IZVRŠAVANJA**, posebno **STARTUP TIME**, i **REDUKUJE MEMORIJSKE ZAHTEVE** u odnosu na **JIT** kompajler.
- **AoT** unapređuje proces sakupljanja đubreta.
- Ideja **AoT**-a je da se **ČITAV KOD PREKOMPAJLIRA** u **VREME INSTALACIJE** aplikacije.
- **AoT** podrazumeva da se aplikacije kompajliraju u **NATIVE KOD** u **JEDNOM PROLAZU** za razliku od **JIT**-a koji podrazumeva "**byte-code**".

ASP.NET – moderan Web alat

Modern Web Tooling

Visual Studio has the open tools and the flexibility you need to create and deploy modern web applications.

[Download Visual Studio](#) 

[Documentation](#) >



```
[Authorize]
1 reference
public class AccountController
{
0 references
public AccountController(
    UserManager userManager,
    SignInManager signInManager)
{
4 references
public UserManager userManager;
    if m.group('sign') == "-":
        self._sign
        self._is_special = False
    else:
        self._compare_check_nans
        self._is_special
        self._is_seven
        self._is_infinity
        self._is_integer
        self._is_logical
        self._is_nan
        self._rescale
        self._sign
        self._is_special = False
    intpart = 0
    if intpart:
        # fin
        fracp
        exp = 0
        self._rescale
        self._sign
        self._is_special = False
```

ASP.NET...and more

Develop for the web with ASP.NET, Node.js, Python, JavaScript

Use the language that makes you happy and productive. Move between languages and project-types with ease using the best editor for HTML5, CSS3, and JavaScript in the world. Tap into the power of LESS, and SASS so front-end developers can join in the fun.

[Supported languages](#) >

ASP.NET - multiplatformska tehnologija

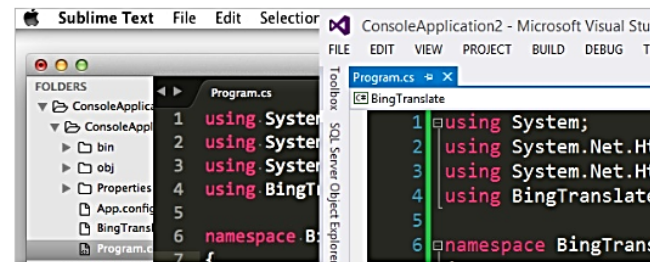
Modern web

Angular, jQuery, Bootstrap, Django, Backbone.js, Express

Increase productivity with powerful web frameworks. Visual Studio includes IntelliSense for your [client-side](#) JavaScript and also lights up with advanced support for today's most popular web frameworks like Angular and Bootstrap.

```
<div ng-c</div>
  A ng-change
  A ng-class
  A ng-class-even
  A ng-class-odd
  A ng-click
  A ng-cloak
  A ng-controller
  A ng-text-change

<div class="container">
  <a class="btn">
    B btn
    B btn-block
    B btn-danger
    B btn-default
    B btn-group
    B btn-group-justified
    B btn-group-lg
    B btn-group-sm
    B btn-group-vertical
  </a>
</div>
```



```
using System;
using System.Net;
using System.Net.Http;
using BingTranslate;
namespace BingTrans
```

Many platforms

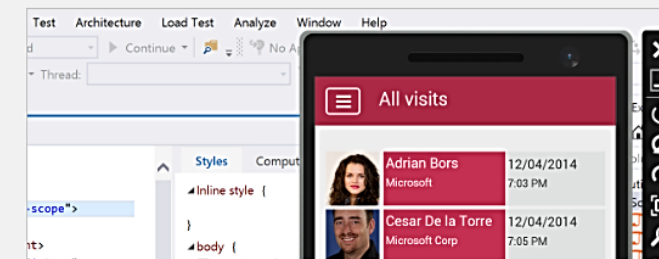
ASP.NET 5 and the .NET Core CLR run on Windows, Mac, and Linux

Write your web applications and service with ASP.NET 5 in C# on the Core CLR using Visual Studio on Windows. Deploy to any web server, or graduate to cloud scale in Microsoft Azure! If you like, code in your editor and OS of choice with [omnisharp.net](#).

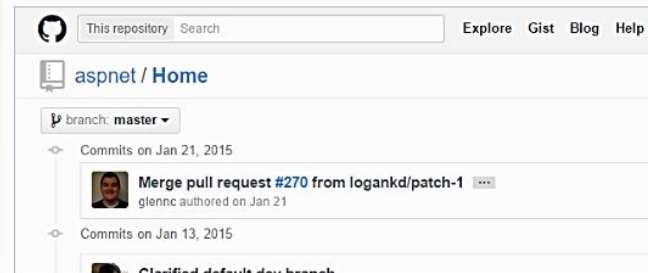
Mobile web apps

Target iOS, Android, and Windows

With [Tools for Apache Cordova](#) you can build a single app for iOS, Android, and Windows using your skillset in web standards: HTML, CSS, and JavaScript. Use the fast and reliable Android Emulator for Hyper-V as well!



ASP .NET – Open source



Open source

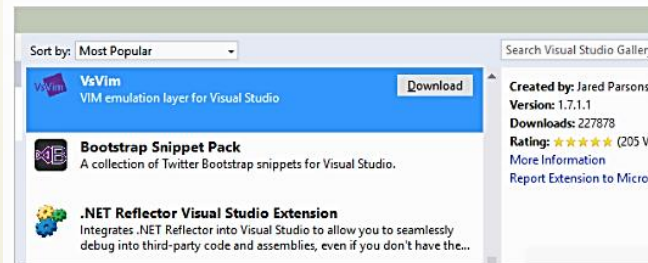
ASP.NET and .NET Core are Apache- and MIT-licensed code on GitHub

Found a bug or got a new idea? Open an issue or send a pull request. All of the ASP.NET and .NET Core CLR runtime components and the Azure SDKs for any language are available on GitHub and taking contributions. Get involved today!

Package managers

NuGet, npm, bower

Join the Open Web and pull in libraries from all over. [NuGet](#) offers rich .NET server-side libraries, the [Bower](#) package manager connects you to the latest thinking in client side JavaScript technology, and [npm](#) pulls in great tools and utilities.



Extensible ecosystem

Extend Visual Studio, explore our community, and make it your own!

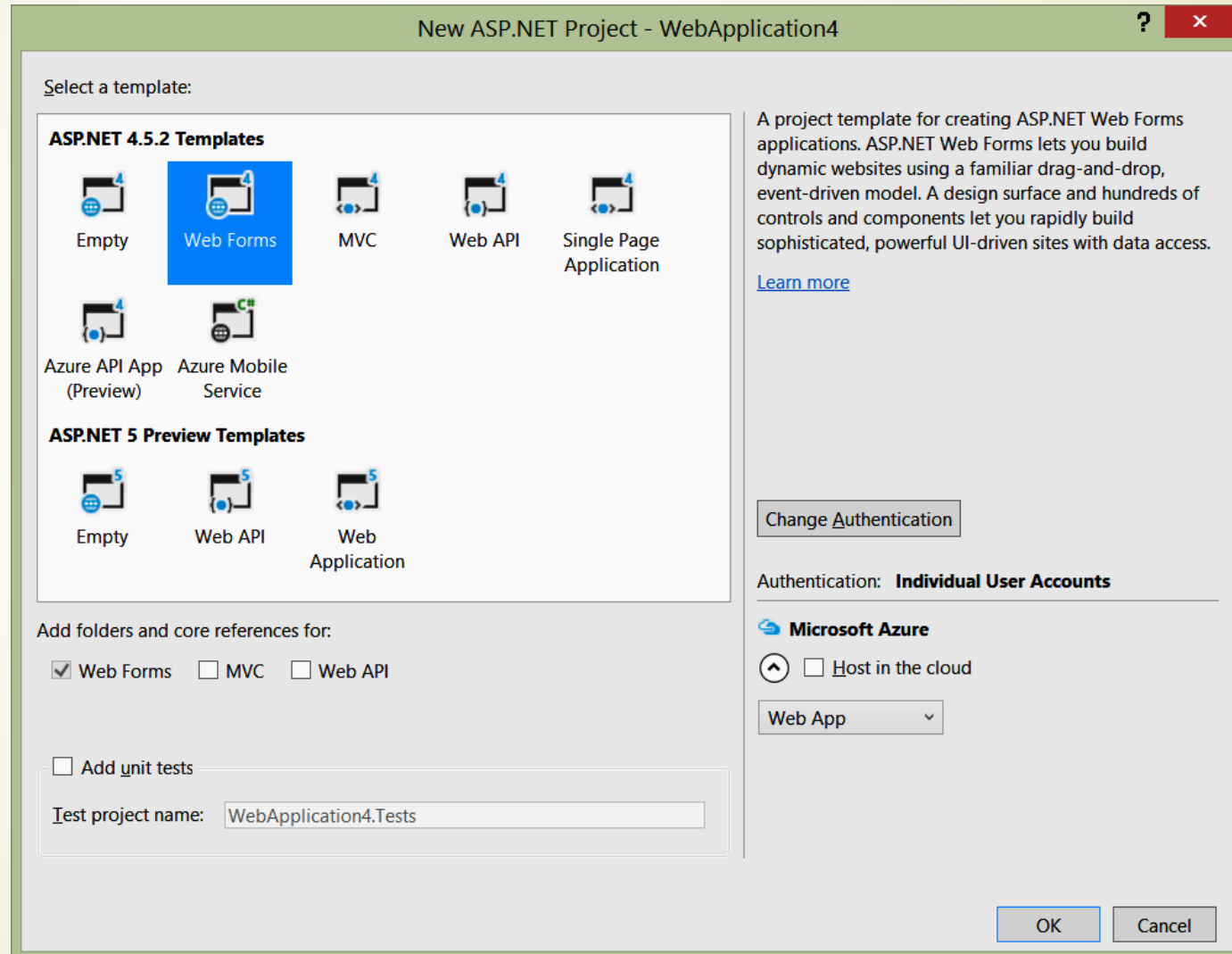
Customize Visual Studio by leveraging the thousands of extensions available from the constantly growing ecosystem or even build your own.

[Extensions for web developers >](#)

ASP .NET Web forms

- MS tehnologija ASP .NET **Web Forms** omogućava razvoj Web aplikacija **NA ISTI NAČIN** na koji se kreiraju **KLASIČNE WINDOWS FORMS APLIKACIJE** (drugi ciklus laboratorijskih vežbi je realizovan u ovoj MS tehnologiji).
- **NA SLIČAN NAČIN**, kao prilikom razvoja klasičnih **DESKTOP** WINDOWS APLIKACIJA (do VS 2013), iz **Web Toolbox-a** Visual Studio-a se na radnu površini “prevlače” **SERVERSKE KONTROLE** koje se postavljaju na Web stranicu.
- Ovaj koncept Web aplikacija podrazumeva **INTERFEJS** baziran na **SERVERSKIM KONTROLAMA**, odnosno, fajlovima koji ih prate:
 - **USER INTERFACE (UI)** fajl: uključuje jezik za označavanje Web dokumenta kojim se definiše **IZGLED WEB STRANICE** i koje se Web kontrole koriste (ekstenzija ovog fajla je **.aspx**).
 - **CODE-BEHIND** fajl: uključuje programski kod koji **OPSLUŽUJE DOGAĐAJE** i **INTERAKCIJE** između objekata na Web stranici. Ova vrsta fajla ima ekstenziju **.aspx.cs** ako se razvoj aplikacije obavlja u programskom jeziku C#.

Izbor tipa Web App u VS 2015



ASP .NET Web forms u VS 2015

The screenshot displays the Visual Studio 2015 environment for a web application project named 'WebApplication2'. The interface is divided into several panes:

- Toolbox (Web Toolbox):** Located on the left, it contains a list of server controls such as Pointer, AdRotator, BulletedList, Button, Calendar, CheckBox, DropDownList, FileUpload, HiddenField, HyperLink, Image, ImageButton, ImageMap, Label, LinkButton, ListBox, Literal, Localize, MultiView, Panel, Placeholder, RadioButton, RadioButtonList, Substitution, and Table. The 'HyperLink' control is currently selected.
- Code - behind file:** The central pane shows the code-behind file for the selected control, containing HTML and C# code. A red label 'Code - behind file' points to this area.
- Design view (Dizajnerski prozor):** The bottom pane shows a visual representation of the page, including a calendar for April 2014. A red label 'Dizajnerski prozor' points to this area.
- Solution Explorer:** Located on the right, it shows the project structure, including folders like Properties, References, Account, App_Data, App_Start, Content, Images, Scripts, and files like About.aspx, Bundle.config, Contact.aspx, Default.aspx, and favicon.ico. A red label 'Solution prozor' points to this area.
- Properties prozor:** The bottom right pane shows the Properties window for the selected control, displaying attributes like (id), accesskey, class, contenteditable, and contextmenu. A red label 'Properties prozor' points to this area.

Additional red text labels include 'Serverske kontrole' pointing to the Toolbox and 'Web' pointing to the 'Web Toolbox' title.

VS 2015 atribut runat = "server"

The screenshot displays the Visual Studio 2015 interface with a code editor showing the source code of a Site.Master page. The code is annotated with red text and arrows pointing to specific attributes.

Annotations:

- Skriptovi na serveru** (Scripts on server) points to the `runat="server"` attribute on the `<asp:ScriptManager>` tag.
- Serverske kontrole** (Server controls) points to the `runat="server"` attribute on the `<asp:ScriptManager>` tag.
- Atributi runat = "server"** (Attributes runat = "server") points to the `runat="server"` attribute on the `<form>` tag.

Code Snippets:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="WebApplication4.Site.Master" %>
<!DOCTYPE html>
<html lang="en">
  <head runat="server">
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title><%: Page.Title %> - My ASP.NET Application</title>
    <asp:PlaceHolder runat="server">
      <%: Scripts.Render("~/bundles/modernizr") %>
    </asp:PlaceHolder>
    <webopt:bundlereference runat="server" path="~/Content/css" />
    <link href="~/favicon.ico" rel="shortcut icon" type="image/x-icon" />
  </head>
  <body>
    <form runat="server">
      <asp:ScriptManager runat="server">
        <Scripts>
          <%-- To learn more about bundling scripts in ScriptManager see http://go.microsoft.com/fwlink/?LinkId=254262 -->
        </Scripts>
      </asp:ScriptManager>
    </form>
  </body>
</html>
```

ASP .NET WebPages i RAZOR

- ASP .NET **WEBPAGES** predstavlja **RADNI OKVIR** (engl. *framework*) za izgradnju Web aplikacija koji koristi tzv. **RAZOR SINTAKSU** za dizajniranje izgleda Web stranice.
- **ASP.NET RAZOR** koristi jednostavnu programsku sintaksu koja omogućava ugradnju **SERVERSKI-BAZIRANOG** koda u Web stranice.
- **RAZOR STRANICE** imaju ekstenziju **.cshtml** kada se za razvoj aplikacije koristi programski jezik C# (mi ćemo koristiti ovu verziju), odnosno, **.vbhtml** ako se koristi Visual Basic .NET.
- Zbog toga što je kôd ugrađen u samu Web stranicu, koncept „**CODE-BEHIND FILE**“ **NIJE PRIMENJIV** u RAZOR-u!
- **RAZOR STRANICE** su sačinjene od **HTML KODA** i **NE KORISTE** se **SERVERSKE KONTROLE** kao u Web Formama (primer: `<asp:Button />`).
- Za kreiranje ASP .NET **WebPages** se može koristiti **Visual Studio**, ali je osnovni alat za to M.S. **WEBMATRIX 2**.
- I ova tehnologija se smatra zastarelom, ali ima podršku do današnjih dana.

RAZOR U VS 2015

WebApplication5 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help

Debug Any CPU Internet Explorer

AccountController.cs HomeController.cs Login.cshtml About.cshtml Contact.cshtml Index.cshtml Lockout.cshtml _LoginPartial.cshtml

```
ViewBag.Title = "Home Page";  
  
<div class="jumbotron">  
  <h1>ASP.NET</h1>  
  <p class="lead">ASP.NET is a free web framework for building great Web sites and Web applicati</p>  
  <p><a href="http://asp.net" class="btn btn-primary btn-lg">Learn more &raquo;</a></p>  
</div>  
  
<div class="row">  
  <div class="col-md-4">  
    <h2>Getting started</h2>  
    <p>  
      ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.  
    </p>  
    <p><a class="btn btn-default" href="http://go.microsoft.com/fwlink/?LinkId=301865">Learn m</p>  
  </div>  
  <div class="col-md-4">  
    <h2>Get more libraries</h2>  
    <p>NuGet is a free Visual Studio extension that makes it easy to add, remove, and update l</p>  
  </div>  
</div>
```

RAZOR sintaksa i „Helper“ metode

Solution Explorer

- App_Start
- Content
- Controllers
- fonts
- Models
- Scripts
- Views
 - Account
 - _ExternalLoginsListPartial.cshtml
 - ConfirmEmail.cshtml
 - ExternalLoginConfirmation.cshtml
 - ExternalLoginFailure.cshtml
 - ForgotPassword.cshtml
 - ForgotPasswordConfirmation.cshtml
 - Login.cshtml
 - Register.cshtml
 - ResetPassword.cshtml
 - ResetPasswordConfirmation.cshtml
 - SendCode.cshtml
 - VerifyCode.cshtml
 - Home
 - About.cshtml
 - Contact.cshtml
 - Index.cshtml
 - Manage

Properties

Output

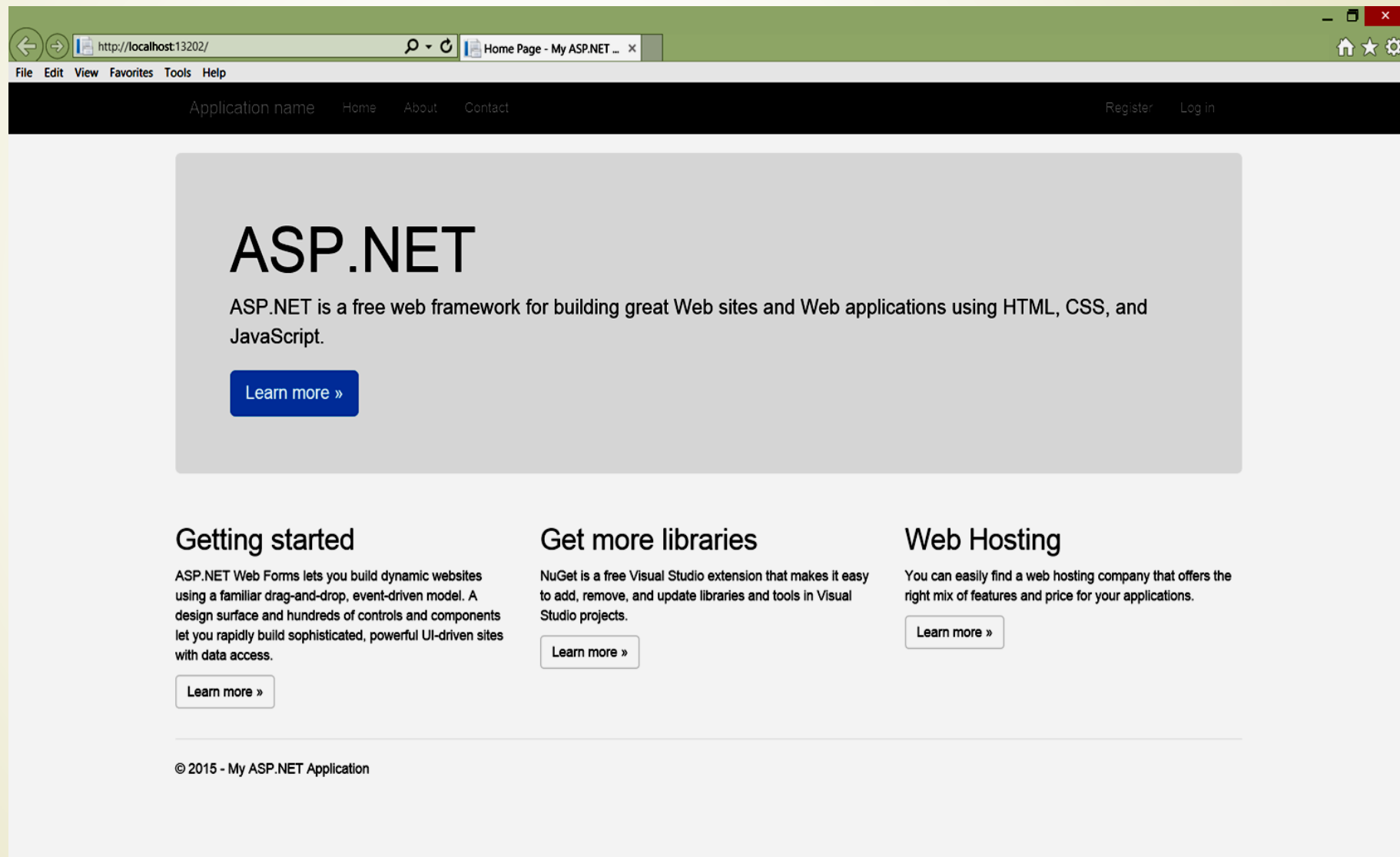
Show output from: Package Manager

Type 'get-help EntityFramework' to see all available Entity Framework commands.

Task Runner Explorer Output

Ready Ln 1 Col 1 Ch 1 INS

Izgled šablona WebPages App, VS 2015



http://localhost:13202/ Home Page - My ASP.NET ... x

File Edit View Favorites Tools Help

Application name Home About Contact Register Log in

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2015 - My ASP.NET Application

ASP .NET MVC

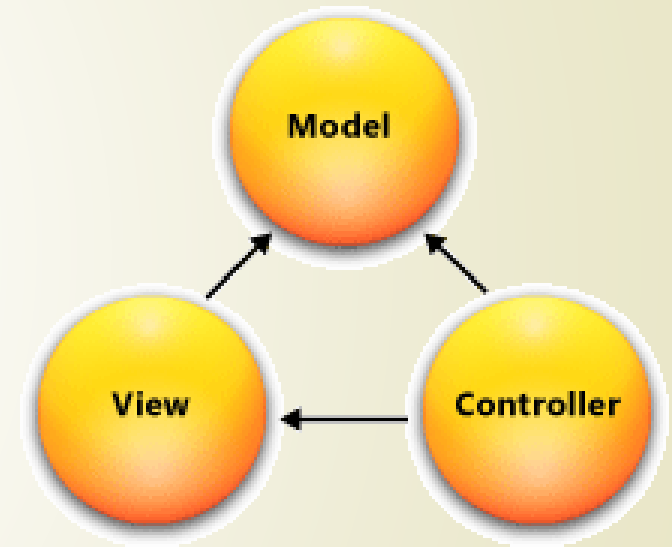
- **ASP.NET MVC** je slobodano i potpuno podržano **RADNO OKRUŽENJE** (okvir) za izgradnju Web stranica zasnovanih na **ŠABLONU** (engl. *pattern*) **Model-View-Controller**.
- Kao i ASP.NET Web Forms, **ASP.NET MVC** je izgrađen **NA VRHU** ASP.NET framework-a (.NET Core-a).
- Koncept .NET-a omogućava da mnogi API-i koji su korišćeni u tradicionalnim ASP .NET Web Forms aplikacijama su **SADA DOSTUPNI** i ASP .NET MVC-u.
- Osnovni benefit koji donosi **MVC** je "**PATERN DIZAJN**" su:
 - **IZOLACIJA** programske logike od
 - **INTERAKCIJE** korisnika.
- **PATERN DIZAJN** uspostavlja **JASNU ARHITEKTURU APLIKACIJE** što ima uticaja na:
 - Poboľšano **ODRŽAVANJE** APLIKACIJE i
 - Njeno **TESTIRANJE**

Istorija ASP .NET MVC-a

- ▶ Prva verzija **ASP.NET MVC**-a se pojavljuje 13.03.2009. sa potpuno implementiranim MVC konceptom na bazi ASP.NET-a. Ova verzija poseduje „routing engine“, „Helper“ metode, „data binding“, ...
- ▶ **ASP.NET MVC 2** se pojavio u martu 2010. kada je **DODAT SKUP** „UI Helpers“ metoda, „strongly typed HTML“ Helper metode, „asynchronous controllers“, validacija na klijentu i serveru ...
- ▶ Deset meseci kasnije pojavljuje se **ASP.NET MVC 3**, koji uključuje .NET 4, „data annotations“ (meta podaci) i „RAZOR“ engine, JavaScript dobija podršku, jQuery validaciju i JSON binding.
- ▶ ASP.NET **MVC 4** dolazi u avgustu 2012., kada se uključuje u Visual Studio 2012.
- ▶ Oktobra 2013., pojavljuje se verzija ASP **.NET MVC 5** koja je podržana u Visual Studiu 2013.
- ▶ MVC 6 u Visual Studiu 2015.

Osnove ASP .NET MVC arhitekture

- **Model-View-Controller** (MVC) je arhitektura koja Web aplikaciju deli u **TRI** osnovne komponente:
 - **Model**,
 - **View** i
 - **Controller**.
- ASP.NET MVC framework pruža **ALTERNATIVU** ASP.NET Web Formama za kreiranje Web aplikacija.
- ASP.NET MVC framework je jednostavno, testibilno okruženje integrisano sa postojećim ASP .NET mogućnostima (kao što su master pages i membership bazirana autentifikacija).
- **MVC** framework je definisan u **System.Web.Mvc** sklopu.



Zahtev za procesiranjem MVC-u

- Najvažni koncepta u ASP .NET MVC je da **NE POSTOJI** relacija između **ZAHTEVA** za nekom Web stranicom i **FIZIČKOG FAJLA** na Web serveru!
- Kod tradicionalnih Web aplikacija (*Web Forms* i *Web Pages*) **SVAKI ZAHTEV** za stranicom se preslikava u **POZIV FIZIČKOM FAJLU** na serveru.
- Tako, ako se traži `http://myapp/mypage.aspx`, Web server interpretira zahtev tako što u svom **FAJL-SISTEMU** traži Web stranicu sa zahtevanim imenom, tj. `mypage.aspx`.
- Po pronalaženju traženog fajla, SERVER **VRAĆA KLIJENTU FAJL** sa HTML kodom.
- Potom, KLIJENT **INTERPRETIRA** WEB SADRŽAJ određen HTML kodom same stranice i **PRIKAZUJE** GA u prozoru Web čitača.
- U kontekstu MVC aplikacije, MVC komponenta koja se naziva „**ROUTING ENGINE**“, vrši proveru **SLAGANJA ZAHTEVA** sa **SPECIFICIRANOM RUTOM**.

Rute u MVC-u

- **RUTA** se u MVC konceptu definiše kao **STRING UZORK** koji određuje **NAČIN PRONALAZENJA** odgovarajućeg **KONTROLERA**, odnosno, klasa u kontroleru čije će **METODE** konačno **OBRADITI ZAHTEV KORISNIKA** za resursom.
- Kada se **RUTA IDENTIFIKUJE**, „*routing engine*“ kreira **POZIV HENDLERU** čiji je zadatak da **KREIRA OBJEKT KONTROLERA** sa metodama koje **PROCESIRAJU KORISNIČKI ZAHTEV**.
- Metode koje obrađuju korisničke zahteve se nazivaju „**AKCIONE METODE**“.
- Kada se završi **OBRADA ZAHTEVA**, akcione metode **GENERIŠU REZULTAT** koji se **VRAĆA NAZAD** klijentu.
- Najčešće je rezultat rada akcionih metoda **HTML kôd** (videćete kasnije i neke druge povratne rezultate) koji se potom prikazuje na klijentu.

Blok šema zahtev za procesiranjem u MVC-u



MVC kontroler

- **SVAKI** pojedinačni HTTP ZAHEV se opslužuje **SPECIFIČNIM KONTROLEROM**.
- U ASP .NET MVC-u je implementiran **KONCEPT KONTROLERA** pomoću **.NET KLASA** koje poseduju metode za procesiranje HTTP zahteva.
- Već je rečeno da se te metode nazivaju **AKCIONIM METODAMA** i one najčešće vraćaju objekte tipa **Actionresult**.
- Procesiranje HTTP zahteva započinje **MAPIRANJEM ZAHTEVA U MVC KONTROLER** i neku od njegovih **AKCIONIH METODA**.
- Posle mapiranja odgovarajuće akcione metode **ONA SE IZVRŠAVA**, a pozivaocu se **VRAĆA OBJEKT TIPAA "Actionresult"** kao rezultat rada akcione metode.
- Jedan od najčešćih povratnih tipova akcionih metoda su **OBJEKTI TIPAA View** koji sadrže **HTML KOD** koji se kasnije renderuje u klijentskom browseru.

Klasa Controller u MVC-u

The screenshot displays the Microsoft Visual Studio IDE with a project named 'WebApplication5'. The main editor window shows the code for 'AccountController.cs'. The code includes the following elements:

```
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using WebApplication5.Models;

namespace WebApplication5.Controllers
{
    [Authorize]
    public class AccountController : Controller
    {
        private ApplicationSignInManager _signInManager;
        private ApplicationUserManager _userManager;

        public AccountController()
        {
        }

        public AccountController(ApplicationUserManager userManager, ApplicationSignInManager signInManager )
        {
            UserManager = userManager;
            SignInManager = signInManager;
        }
    }
}
```

The Solution Explorer on the right shows the project structure, with 'AccountController' selected under the 'Controllers' folder. The Properties window at the bottom right is empty. The Output window at the bottom shows the message: 'Type 'get-help EntityFramework' to see all available Entity Framework commands.'

Kreiranje MVC kontrolera

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace HaveYouSeenMe.Controllers
{
    public class PetController : Controller
    {
        // Izvorni kod akcionih metoda
        . . .
    }
}
```

PetController klasa se
dobija nasleđivanjem
klase **Controller**

Podrazumvano ime kontrolera



ActionResult metoda

```
public ActionResult Display()
{
    var name = (string)RouteData.Values["id"];
    var model = PetManagement.GetByName(name);
    if (model == null)
        return RedirectToAction("NotFound");
    return View(model);
}

public ActionResult NotFound()
{
    return View();
}
```

MVC View komponenta

- Da bi se Web sadržaju **UČINILI DINAMIČNIM**, renderovanje Web stranica se **PREPUŠTA WEB SERVERU**.
- Za potrebe dizajniranja dinamičkih Web stranica neophodno je načiniti "**VIEW ENGINES**" koje manipulišu podacima na serveru.
- **VIEW ENGINES** poseduju specifičnu sintaksu za rad sa **SERVER-SIDE ELEMENTIMA** i renderovanje HTML-a u Web čitačima.
- Najčešće **POVRATNA VREDNOST** iz akcionih metoda je tipa **ViewResult**.
- Ovaj povratni tip indicira da bi ASP.NET MVC trebalo da **POZOVE STRANICU NA RENDEROVANJE** i vrati HTML kod nazad KLIJENTU.
- Ove stranice se smeštaju u **Views DIREKTORIJUMU** unutar aplikacije i podrazumevaju SERVER-SIDE KÔD koji proizvodi finalni HTML.
- Prilikom kreiranja **View**-a u VS, neophodno je specificirati "**view engine**" koja će se koristiti za PROCESIRANJE SERVER-SIDE KODA.

Sekcije u Home page Internet App VS 2015

The screenshot shows a web browser window with the URL `http://localhost:60650/`. The page content is annotated with red boxes and labels:

- HEADER:** A pink box labeled "HEADER" is positioned over the top navigation area, which includes "your logo here", "Register", "Log in", and a menu with "Home", "About", and "Contact".
- FEATURED:** A pink box labeled "FEATURED" is positioned over the main content area, which contains the heading "Home Page. Modify this template to jump-start your ASP.NET MVC application." and a paragraph of introductory text. A red annotation "@RenderSection" points to this area.
- BODY:** A pink box labeled "BODY" is positioned over the list of suggestions. A red annotation "Neimenovana sekcija" (Unnamed section) points to this area, and another red annotation "@RenderBody" points to the list items.
- FOOTER:** A pink box labeled "FOOTER" is positioned over the bottom of the page, which contains the copyright notice "© 2014 - My ASP.NET MVC Application".

Additional red annotations include "Tittle Web stranice" (Web page title) pointing to the browser title bar and "Register" / "Log in" buttons in the header.

Layout.cshtml (1)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>@ViewBag.Title - My ASP.NET MVC Application</title>
  <link href="~/favicon.ico" rel="shortcut icon" type="image/x-icon" />
  <meta name="viewport" content="width=device-width" />
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
</head>
<body>
  <header>
    <div class = "content-wrapper">
      <div class = "float-left">
        <p class = "site-title">
          @Html.ActionLink("your logo here", "Index", "Home")
        </p>
      </div>
```

Layout.cshtml (2)

```
<div class = "float-right">
  <section id = "login">
    @Html.Partial("_LoginPartial")
  </section>
  <nav>
    <ul id = "menu">
      <li>@Html.ActionLink("Home", "Index", "Home")</li>
      <li>@Html.ActionLink("About", "About", "Home")</li>
      <li>@Html.ActionLink("Contact Us", "Contact", "Home")</li>
    </ul>
  </nav>
</div>
</div>
</header>
```


Layout.cshtml (3)

```
<div id = "body">
    @RenderSection("featured", required: false)
    <section class = "content-wrapper main-content clear-fix">
        @RenderBody()
    </section>
</div>
<footer>
    <div class = "content-wrapper">
        <div class = "float-left">
            <p>&copy; @DateTime.Now.Year - My ASP.NET MVC Application</p>
        </div>
    </div>
</footer>
    @Scripts.Render("~/bundles/jquery")
    @RenderSection("scripts", required: false)
</body>
</html>
```







MVC Modeli

- Termin **MODEL** u ASP .NET MVC-u se odnosi na **SKUP OBJEKATA** koji implementiraju **FUNKCIONALNOST** WEB aplikacije.
- Najznačajniji **TIPOVI MODELA** u MVC-u su:
 - Data model,
 - Business model i
 - View model.
- **DATA MODEL** je značajan za interakciju sa **BAZOM PODATAKA**.
- **BUSINESS MODEL** se koristi da obavi **GENERALNA IZRAČUNAVANJA**, i u većini slučajeva on saraduje sa **DATA MODELOM** da bi preuzeo podatke iz baze i kasnije ih smestio u nju.
- **VIEW MODELI** su klase koje prosleđuju podatke iz kontrolera na **PRIKAZ**.
- **DATA** i **BUSINESS** modeli se nazivaju **DOMAIN MODELIMA**.
- **VIEW MODELI** nisu deo domain modela i njihova uloga je da proslede podatke iz **KONTROLERA** na **PRIKAZ** i **OBRATNO**.

Šabloni Web App u VS 2019 (1)

Create a new project

Recent project templates

 Console Application	C#
 Console App (.NET Framework)	C#
 ASP.NET Web Application (.NET Framework)	C#
 ASP.NET Web Application (.NET Framework)	Visual Basic
 Dependency Validation	
 Windows Forms App (.NET Framework)	C#
 Mobile App (Xamarin.Forms)	C#

Search for templates (Alt+S)

Clear all

C#

Windows

Web



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

C# Linux macOS Windows Cloud Service Web



Blazor WebAssembly App

A project template for creating a Blazor app that runs on WebAssembly and is optionally hosted by an ASP.NET Core app. This template can be used for web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

C# Linux macOS Windows Cloud Service Web



ASP.NET Core Web App (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

C# Linux macOS Windows Cloud Service Web



Blazor Server App

A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web



ASP.NET Core Web API

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

C# Linux macOS Windows Cloud Service Web



ASP.NET Core gRPC Service

A project template for creating a gRPC ASP.NET Core service.


Back

Next

Web App u VS 2019 (2)








Create a new project

Recent project templates

-  Console Application C#
-  Console App (.NET Framework) C#
-  ASP.NET Web Application (.NET Framework) C#
-  ASP.NET Web Application (.NET Framework) Visual Basic
-  Dependency Validation
-  Windows Forms App (.NET Framework) C#
-  Mobile App (Xamarin.Forms) C#

Search for templates (Alt+S) Clear all

C# All platforms Web

-  **ASP.NET Core Web App (Model-view-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
C# Linux macOS Windows Cloud Service Web
-  **Blazor Server App**
A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Cloud Web
-  **ASP.NET Core Web API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# Linux macOS Windows Cloud Service Web
-  **ASP.NET Core gRPC Service**
A project template for creating a gRPC ASP.NET Core service.
C# Linux macOS Windows Cloud Service Web
-  **ASP.NET Core with Angular**
A project template for creating an ASP.NET Core application with Angular
C# Linux macOS Windows Cloud Service Web
-  **ASP.NET Core with Reactjs**
A project template for creating an ASP.NET Core application with Reactjs
C# Linux macOS Windows Cloud Service Web
-  **ASP.NET Core with Reactjs and Redux**
A project template for creating an ASP.NET Core application with Reactjs and Redux
C# Linux macOS Windows Cloud Service Web

Back Next

ASP .NET Framework MVC Web App u VS 2019

Create a new project

Recent project templates

- ASP.NET Core Web App (C#)
- Console Application (C#)
- Console App (.NET Framework) (C#)
- ASP.NET Web Application (.NET Framework) (C#)
- ASP.NET Web Application (.NET Framework) (Visual Basic)
- Dependency Validation
- Windows Forms App (.NET Framework) (C#)
- Mobile App (Xamarin.Forms) (C#)

Search: Clear all

C# All platforms Web

ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
C# Linux macOS Windows Cloud Service Web

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# Linux macOS Windows Cloud Service Web

ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
C# Windows Cloud Web

Other results based on your search

ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
Visual Basic Windows Cloud Web

ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
F# Linux macOS Windows Cloud Service Web

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Back Next

ASP .NET Framework MVC Web App

Create a new ASP.NET Web Application



Empty

An empty project template for creating ASP.NET applications. This template does not have any content in it.



Web Forms

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.



MVC

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.



Web API

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.



Single Page Application

A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication

No Authentication
[Change](#)

Add folders & core references

- Web Forms
- MVC
- Web API

Advanced

- Configure for HTTPS
- Docker support
(Requires Docker Desktop)
- Also create a project for unit tests

WebApplication2.Tests

Back

Create

ASP .NET Framework MVC Web App

The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the `RouteConfig.cs` file with the following code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 using System.Web.Routing;
7
8 namespace WebApplication4
9 {
10     1 reference
11     public class RouteConfig
12     {
13         1 reference
14         public static void RegisterRoutes(RouteCollection routes)
15         {
16             routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
17
18             routes.MapRoute(
19                 name: "Default",
20                 url: "{controller}/{action}/{id}",
21                 defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
22             );
23         }
24     }
```
- Solution Explorer:** Shows the project structure for `WebApplication4`, including folders like `App_Start`, `Content`, `Controllers`, `Views`, and files like `RouteConfig.cs`, `BundleConfig.cs`, and `FilterConfig.cs`.
- Properties Window:** Shows the file properties for `RouteConfig.cs`.
- Status Bar:** Indicates 129% zoom, no issues found, and the current cursor position (Ln: 1, Ch: 1, SPC, CRLF).

ASP .NET Framework MVC Web App

Home Page - My ASP.NET Applic x +

localhost:3662

Apps New Tab Академија технич... Uživo prenosi C# Books, .NET Boo... Web Api Resources Consume Web API... 24 Coronavirus CO... Quotes API Docum...

Other bookmarks Reading list

Application name Home About Contact

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.







[Learn more »](#)

© 2021 - My ASP.NET Application

ASP .NET Core MVC Web App u VS 2019


Create a new project


Recent project templates


-  ASP.NET Core Web App C#
-  Console Application C#
-  Console App (.NET Framework) C#
-  ASP.NET Web Application (.NET Framework) C#
-  ASP.NET Web Application (.NET Framework) Visual Basic
-  Dependency Validation
-  Windows Forms App (.NET Framework) C#
-  Mobile App (Xamarin.Forms) C#

mvc Clear all


C# All platforms Web


 **ASP.NET Core Web App (Model-View-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core **MVC** Views and Controllers. This template can also be used for RESTful HTTP services.
C# Linux macOS Windows Cloud Service Web **Core**


 **ASP.NET Core Web API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core **MVC** Views and Controllers.
C# Linux macOS Windows Cloud Service Web

 **ASP.NET Web Application (.NET Framework)**
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, **MVC**, or Web API applications and add many other features in ASP.NET.
C# Windows Cloud Web

Other results based on your search

 **ASP.NET Web Application (.NET Framework)**
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, **MVC**, or Web API applications and add many other features in ASP.NET.
Visual Basic Windows Cloud Web

 **ASP.NET Core Web App (Model-View-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core **MVC** Views and Controllers. This template can also be used for RESTful HTTP services.
F# Linux macOS Windows Cloud Service Web

 **ASP.NET Core Web API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core **MVC** Views and Controllers.

Back Next

ASP .NET Core MVC Web App u VS 2019

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Target Framework ⓘ

.NET Core 3.1 (Long-term support)

Authentication Type ⓘ

Windows

Configure for HTTPS ⓘ

Enable Docker ⓘ

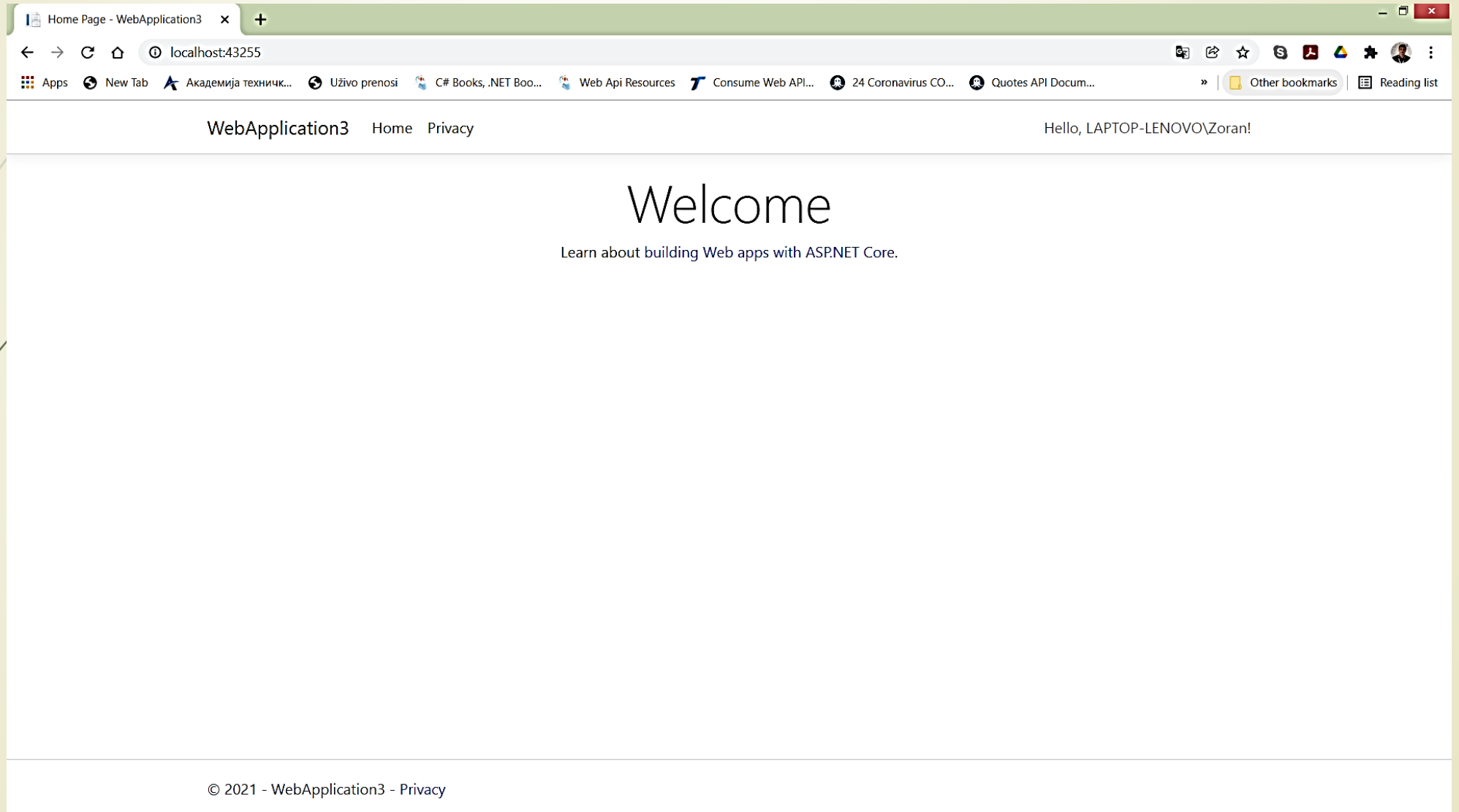
Docker OS ⓘ

Linux

Enable Razor runtime compilation ⓘ

Back Create

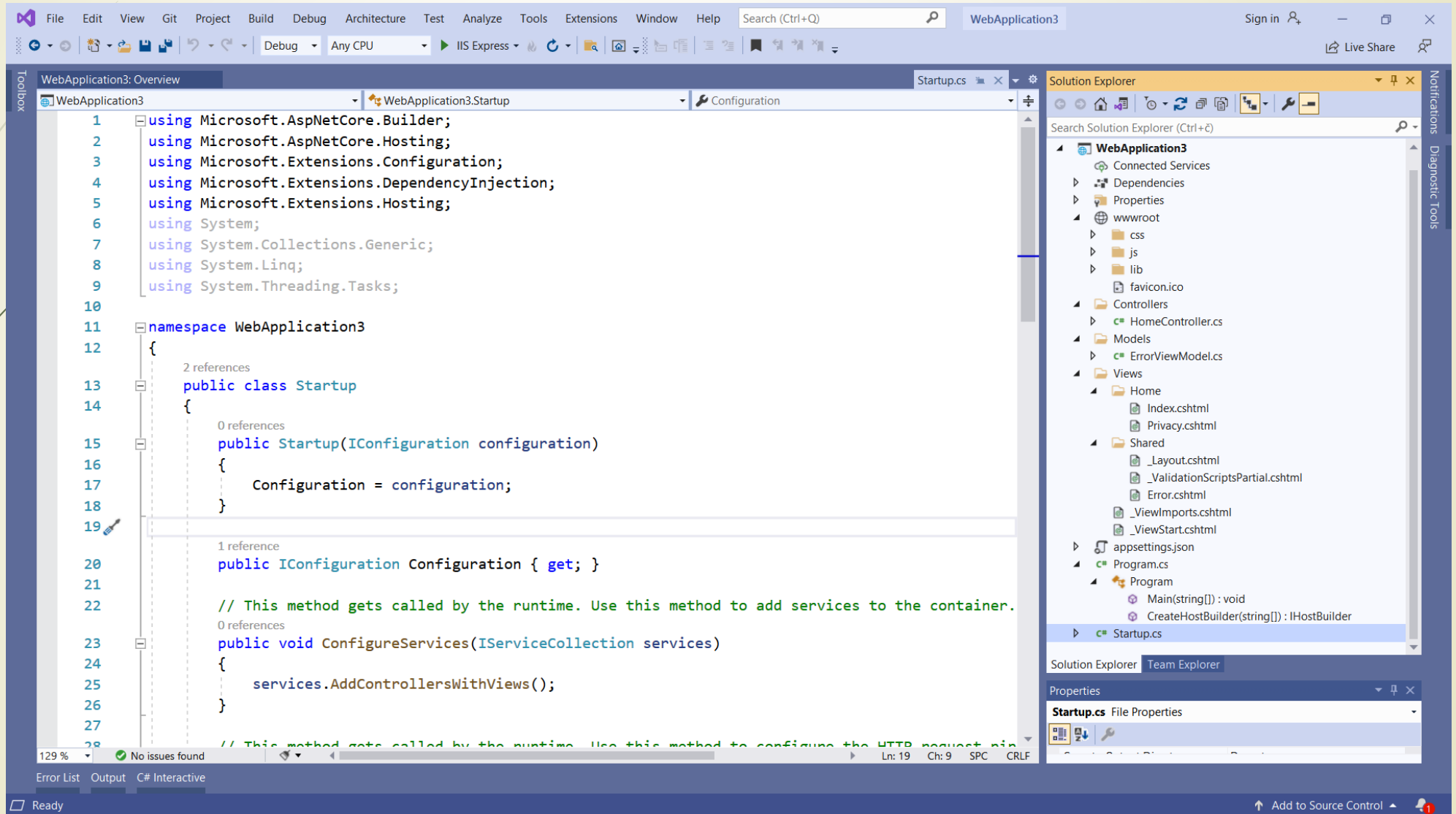
ASP .NET Core MVC Web App u VS 2019



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

- Browser Tab:** Home Page - WebApplication3
- Address Bar:** localhost:43255
- Bookmarks:** Apps, New Tab, Академија технич..., Uživo prenosi, C# Books, .NET Boo..., Web Api Resources, Consume Web API..., 24 Coronavirus CO..., Quotes API Docum..., Other bookmarks, Reading list
- Page Header:** WebApplication3 Home Privacy (left); Hello, LAPTOP-LENOVO\Zoran! (right)
- Main Content:**
 - # Welcome
 - Learn about building Web apps with ASP.NET Core.
- Page Footer:** © 2021 - WebApplication3 - Privacy

Arhitektura ASP .NET Core MVC Web App



```
1 using Microsoft.AspNetCore.Builder;
2 using Microsoft.AspNetCore.Hosting;
3 using Microsoft.Extensions.Configuration;
4 using Microsoft.Extensions.DependencyInjection;
5 using Microsoft.Extensions.Hosting;
6 using System;
7 using System.Collections.Generic;
8 using System.Linq;
9 using System.Threading.Tasks;
10
11 namespace WebApplication3
12 {
13     2 references
14     public class Startup
15     {
16         0 references
17         public Startup(IConfiguration configuration)
18         {
19             Configuration = configuration;
20         }
21         1 reference
22         public IConfiguration Configuration { get; }
23
24         // This method gets called by the runtime. Use this method to add services to the container.
25         0 references
26         public void ConfigureServices(IServiceCollection services)
27         {
28             services.AddControllersWithViews();
29
30         // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
```

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the `Startup.cs` file with line numbers 1 through 29. The code includes using statements for `Microsoft.AspNetCore.Builder`, `Microsoft.AspNetCore.Hosting`, `Microsoft.Extensions.Configuration`, `Microsoft.Extensions.DependencyInjection`, `Microsoft.Extensions.Hosting`, `System`, `System.Collections.Generic`, `System.Linq`, and `System.Threading.Tasks`. It defines a namespace `WebApplication3` and a public class `Startup` with a constructor `Startup(IConfiguration configuration)` and a `ConfigureServices` method that calls `services.AddControllersWithViews()`.
- Solution Explorer:** Shows the project structure for `WebApplication3`, including folders for `css`, `js`, `lib`, `Controllers`, `Models`, `Views`, and `Program.cs`.
- Properties Window:** Shows the file properties for `Startup.cs`.
- Status Bar:** Indicates the current line (Ln: 19), column (Ch: 9), and other details.

Arhitektura ASP .NET FW WebForms App

Create a new ASP.NET Web Application

Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
[Change](#)

Add folders & core references

- Web Forms
- MVC
- Web API

Advanced

- Configure for HTTPS
- Docker support
(Requires [Docker Desktop](#))
- Also create a project for unit tests

WebApplication5.Tests

[Back](#) [Create](#)

Arhitektura ASP .NET FW WebForms App

Home Page - My ASP.NET Applic x +

localhost:35536/Default

Apps New Tab Академија технич... Uživo prenosi C# Books, .NET Boo... Web Api Resources Consume Web API... 24 Coronavirus CO... Quotes API Docum... Other bookmarks Reading list

Application name Home About Contact

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

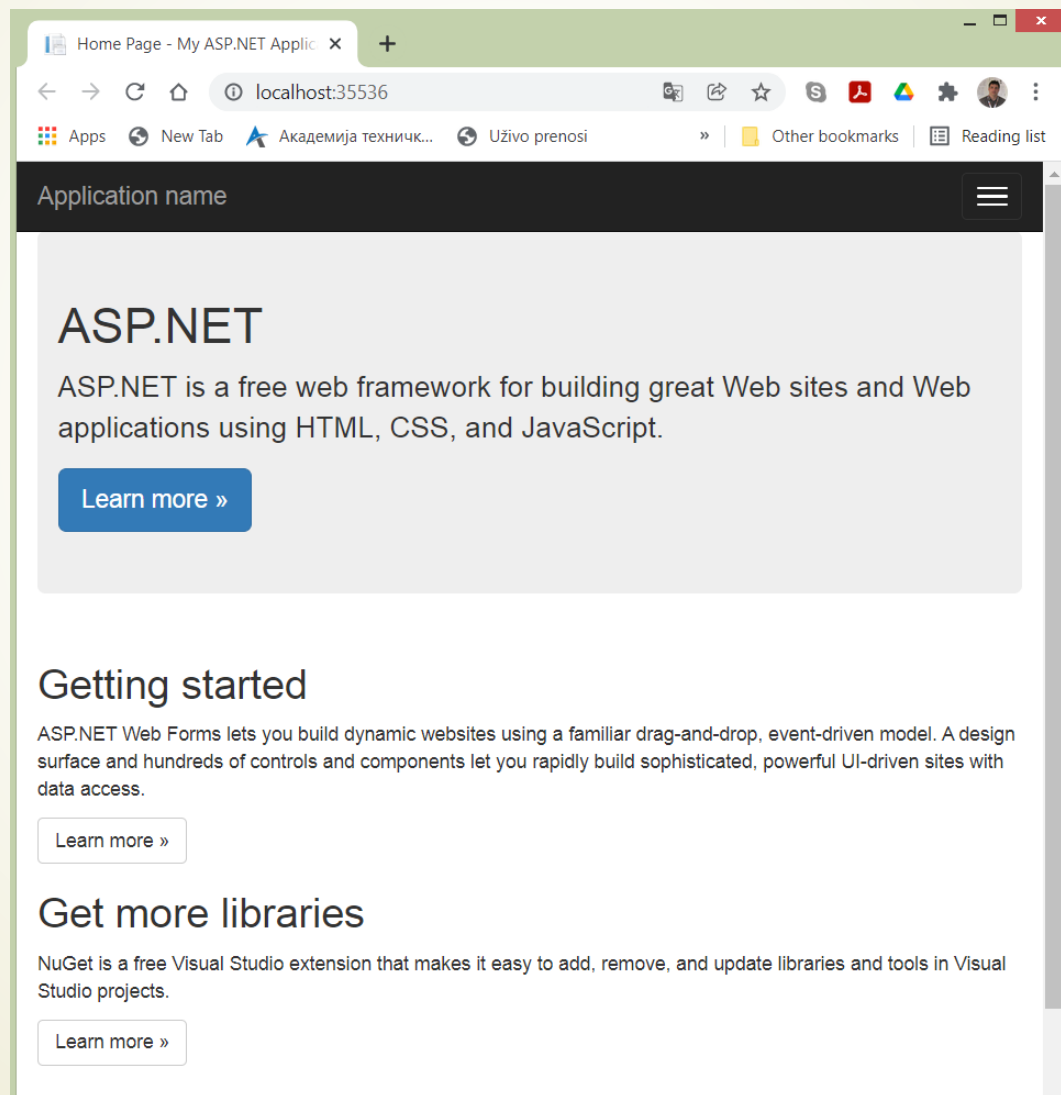
Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2021 - My ASP.NET Application











Arhitektura ASP .NET FW WebForms App



Blazor WebAssembly App

Create a new project

Recent project templates

-  Blazor WebAssembly App C#
-  ASP.NET Web Application (.NET Framework) C#
-  ASP.NET Core Web App (Model-View-Controller) C#
-  ASP.NET Core Web App C#
-  Console Application C#
-  Console App (.NET Framework) C#
-  ASP.NET Web Application (.NET Framework) Visual Basic
-  Dependency Validation
-  Windows Forms App (.NET Framework) C#
-  Mobile App (Xamarin.Forms) C#

Search for templates (ALT+S) 🔍

[Clear all](#)

C# All platforms Web



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

C# Linux macOS Windows Cloud Service Web



Blazor WebAssembly App

A project template for creating a Blazor app that runs on WebAssembly and is optionally hosted by an ASP.NET Core app. This template can be used for web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

C# Linux macOS Windows Cloud Service Web



ASP.NET Core Web App (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

C# Linux macOS Windows Cloud Service Web



Blazor Server App

A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web



ASP.NET Core Web API

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

C# Linux macOS Windows Cloud Service Web



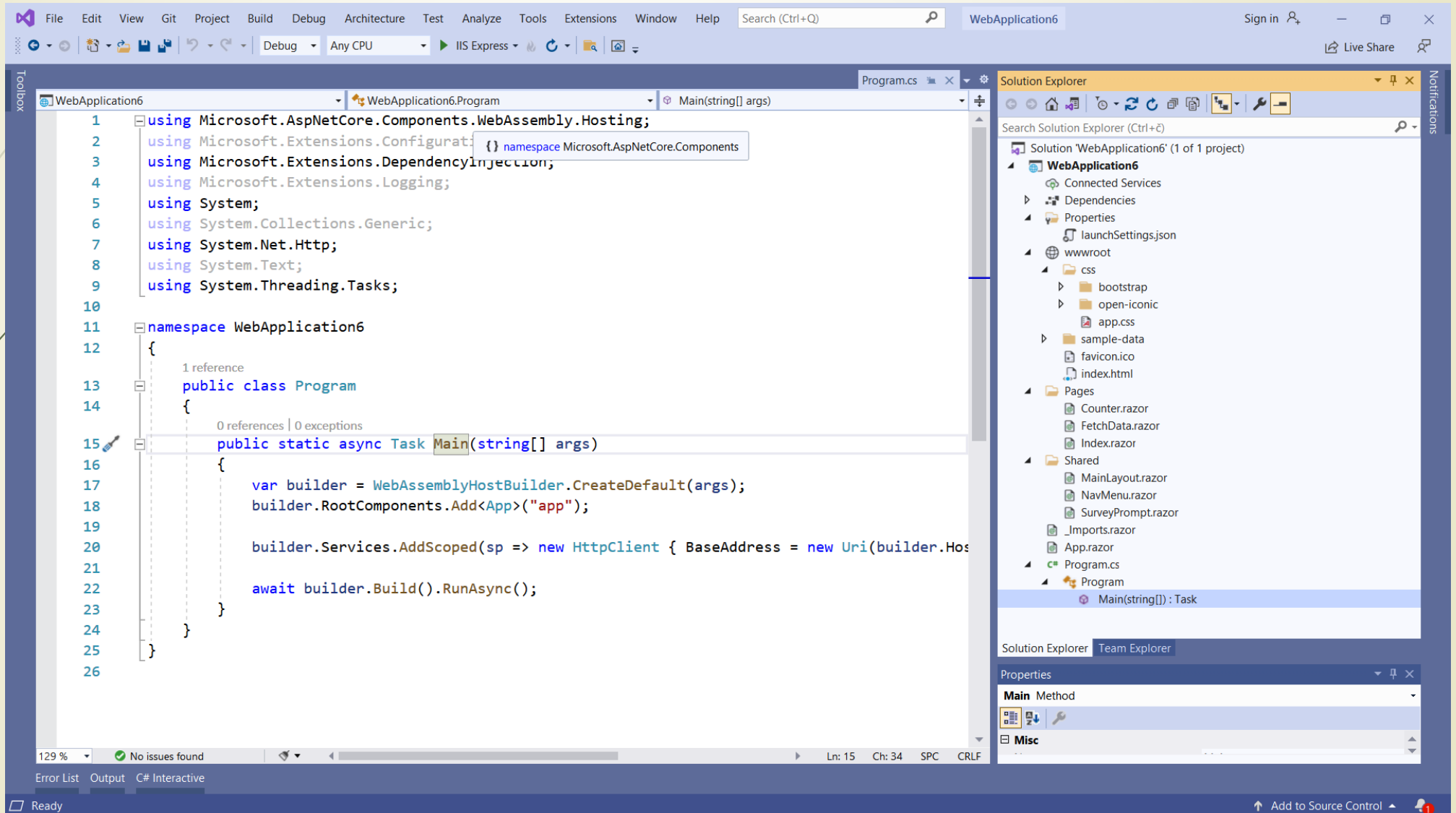
ASP.NET Core gRPC Service

A project template for creating a gRPC ASP.NET Core service.

[Back](#)

[Next](#)

Blazor WebAssembly App



```
1 using Microsoft.AspNetCore.Components.WebAssembly.Hosting;
2 using Microsoft.Extensions.Configuration;
3 using Microsoft.Extensions.DependencyInjection;
4 using Microsoft.Extensions.Logging;
5 using System;
6 using System.Collections.Generic;
7 using System.Net.Http;
8 using System.Text;
9 using System.Threading.Tasks;
10
11 namespace WebApplication6
12 {
13     public class Program
14     {
15         public static async Task Main(string[] args)
16         {
17             var builder = WebAssemblyHostBuilder.CreateDefault(args);
18             builder.RootComponents.Add<App>("app");
19
20             builder.Services.AddScoped(sp => new HttpClient { BaseAddress = new Uri(builder.HostEnvironment.BaseAddress) });
21
22             await builder.Build().RunAsync();
23         }
24     }
25 }
26
```

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the C# code for `Program.cs`. The `Main` method is highlighted, and a tooltip for `namespace Microsoft.AspNetCore.Components` is visible over the `using` statements.
- Solution Explorer:** Shows the project structure for `WebApplication6`, including folders like `css`, `Pages`, and `Shared`, and files like `Program.cs`.
- Properties Window:** Shows the `Main` method properties.
- Status Bar:** Indicates the current line (Ln: 15), column (Ch: 34), and other details.

Blazor WebAssembly App

WebApplication6 [About](#)

Home
Counter
Fetch data

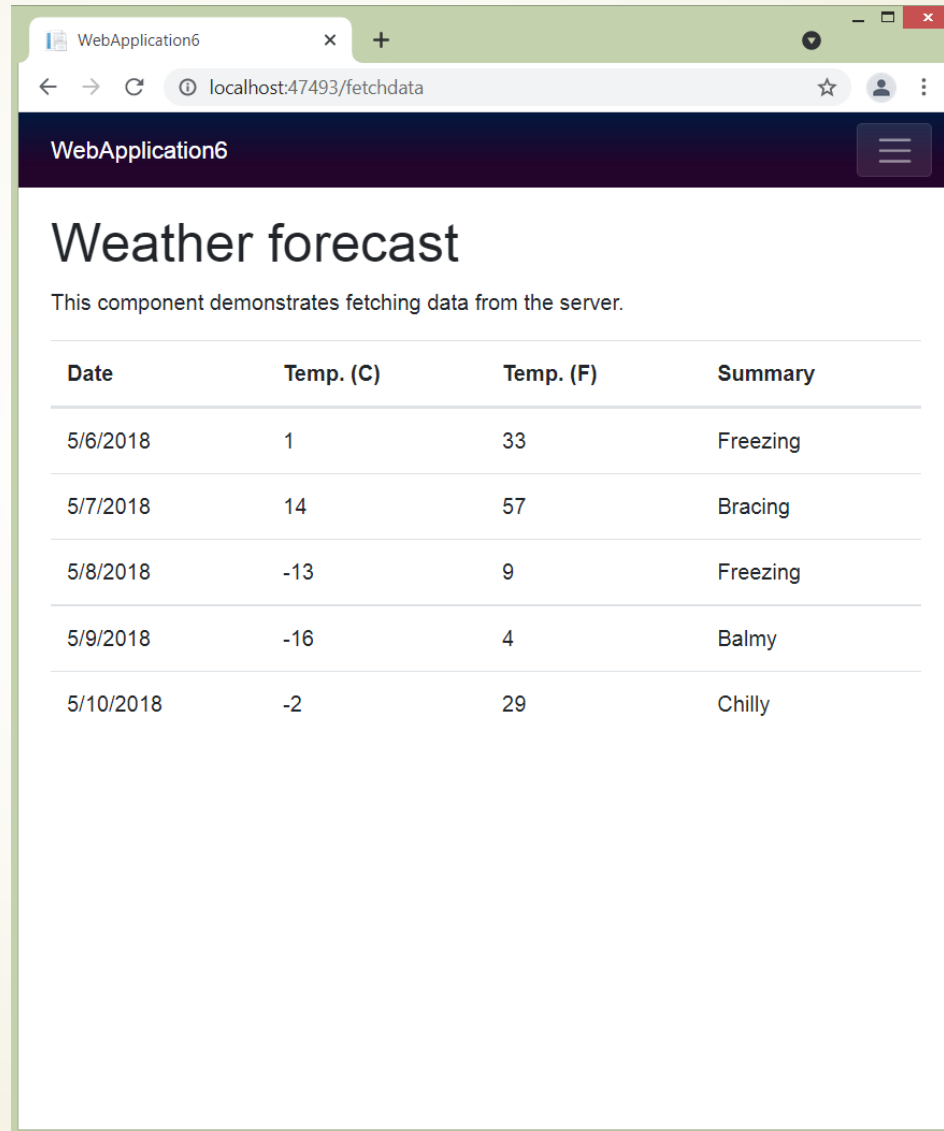
Weather forecast

This component demonstrates fetching data from the server.

Date	Temp. (C)	Temp. (F)	Summary
5/6/2018	1	33	Freezing
5/7/2018	14	57	Bracing
5/8/2018	-13	9	Freezing
5/9/2018	-16	4	Balmy
5/10/2018	-2	29	Chilly

localhost:47493/fetchdata

Blazor WebAssembly App



WebApplication6

Weather forecast

This component demonstrates fetching data from the server.

Date	Temp. (C)	Temp. (F)	Summary
5/6/2018	1	33	Freezing
5/7/2018	14	57	Bracing
5/8/2018	-13	9	Freezing
5/9/2018	-16	4	Balmy
5/10/2018	-2	29	Chilly

Windows WebForms App

The screenshot shows the 'Create a new project' dialog in Visual Studio. On the left, there is a list of 'Recent project templates' including Windows Forms App (.NET Framework), Blazor WebAssembly App, ASP.NET Web Application (.NET Framework), ASP.NET Core Web App (Model-View-Controller), ASP.NET Core Web App, Console Application, Console App (.NET Framework), ASP.NET Web Application (.NET Framework), Dependency Validation, and Mobile App (Xamarin.Forms).

The main search area has a search box containing 'web' and a 'Clear all' button. Below the search box are filters for 'C#' (language), 'All platforms' (platform), and 'Web' (category).

The search results are divided into two sections: 'Other results based on your search' and a list of specific project templates. The first result is 'ASP.NET Web Application (.NET Framework)', which is highlighted. Below it are 'Web Driver Test for Edge (.NET Core)' and 'Web Driver Test for Edge (.NET Framework)'. The second section, 'Other results based on your search', includes 'ASP.NET Web Application (.NET Framework)', 'Django Web Project', and 'Flask Web Project'. Each result includes a description and a list of tags for language, platform, and category.

At the bottom right, there are 'Back' and 'Next' buttons.

Windows WebForms App

The screenshot displays the Visual Studio IDE with a Windows WebForms application. The main editor shows the code for Site.Master, which includes the following HTML and ASP.NET code:

```
1 <%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" Inherits="WebApplication7.SiteMaster" %>
2
3 <!DOCTYPE html>
4
5 <html lang="en">
6 <head runat="server">
7   <meta charset="utf-8" />
8   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
9   <title><%: Page.Title %> - My ASP.NET Application</title>
10
11   <asp:PlaceHolder runat="server">
12     <%: Scripts.Render("~/bundles/modernizr") %>
13   </asp:PlaceHolder>
14
15   <webopt:bundlereference runat="server" path="~/Content/css" />
16   <link href="~/favicon.ico" rel="shortcut icon" type="image/x-icon" />
17
18 </head>
19 <body>
20   <form runat="server">
21     <asp:ScriptManager runat="server">
22       <Scripts>
23         <!--To learn more about bundling scripts in ScriptManager see https://go.microsoft.com/fwlink
24         <!--Framework Scripts-->
25         <asp:ScriptReference Name="MsAjaxBundle" />
26         <asp:ScriptReference Name="jquery" />
27         <asp:ScriptReference Name="bootstrap" />
28         <asp:ScriptReference Name="WebForms.js" Assembly="System.Web" Path="~/Scripts/WebForms/WebFor
29         <asp:ScriptReference Name="WebUIValidation.js" Assembly="System.Web" Path="~/Scripts/WebForms
30         <asp:ScriptReference Name="MenuStandards.js" Assembly="System.Web" Path="~/Scripts/WebForms/M
31         <asp:ScriptReference Name="GridView.js" Assembly="System.Web" Path="~/Scripts/WebForms/GridVi
32         <asp:ScriptReference Name="DetailsView.js" Assembly="System.Web" Path="~/Scripts/WebForms/De
```

The Solution Explorer on the right shows the project structure for 'WebApplication7' (1 of 1 project). The project contains the following files and folders:

- WebApplication7
 - Connected Services
 - Properties
 - References
 - App_Data
 - App_Start
 - Content
 - fonts
 - Scripts
 - About.aspx
 - Bundle.config
 - Contact.aspx
 - Default.aspx
 - favicon.ico
 - Global.asax
 - packages.config
 - Site.Master
 - Site.Mobile.Master
 - ViewSwitcher.ascx
 - Web.config

The Properties window at the bottom right shows the properties for the selected <BODY> element, including:

Property	Value
accesskey	
class	
contenteditable	
contextmenu	

Windows WebForms App

Home Page - My ASP.NET Applic x +

localhost:2839

Apps New Tab Академија техничк... Uživo prenosi C# Books, .NET Boo... Web Api Resources Consume Web API... 24 Coronavirus CO... Quotes API Docum... Other bookmarks Read Zoran

Application name Home About Contact

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2021 - My ASP.NET Application