

2022/2023(2)

IF184504 Web Programming

Lecture #8

System Demonstrations:

PHP, JSP & ASP.NET

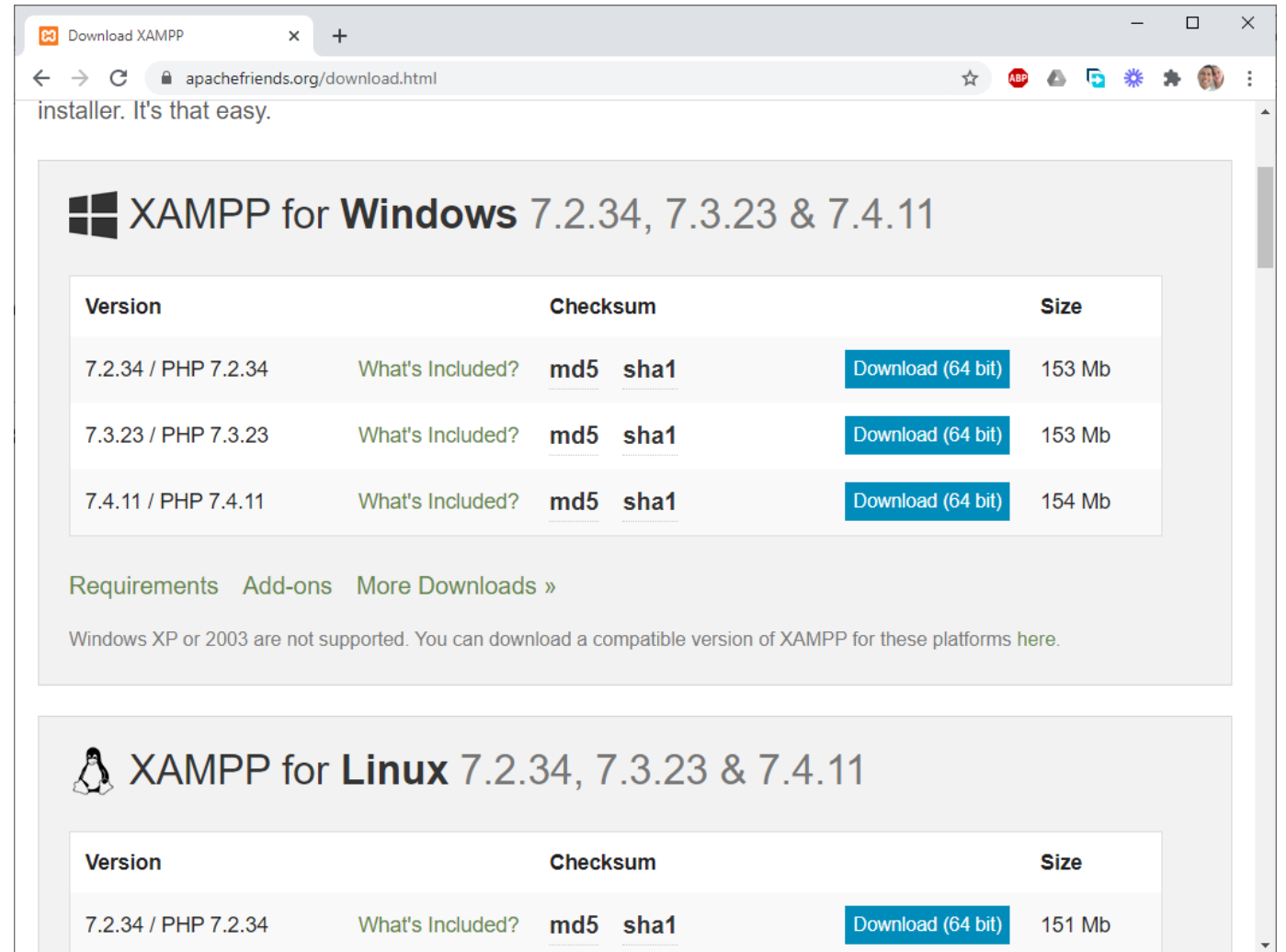
Misbakhul Munir **IRFAN SUBAKTI**

司馬伊凡

Мисбакхул Мунир **Ирфан Субакти**

XAMPP: installation

- Don't install XAMPP on C:\Program Files (x86) → right access problems!



The screenshot shows a web browser window with the URL `apachefriends.org/download.html`. The page content includes a header for "XAMPP for Windows 7.2.34, 7.3.23 & 7.4.11" and a table of download links. Below this, there are links for "Requirements", "Add-ons", and "More Downloads". A note states "Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here." The bottom section of the screenshot shows the header for "XAMPP for Linux 7.2.34, 7.3.23 & 7.4.11" and a table with one row for version 7.2.34 / PHP 7.2.34.

Version	Checksum	Size
7.2.34 / PHP 7.2.34	What's Included? md5 sha1	153 Mb
7.3.23 / PHP 7.3.23	What's Included? md5 sha1	153 Mb
7.4.11 / PHP 7.4.11	What's Included? md5 sha1	154 Mb

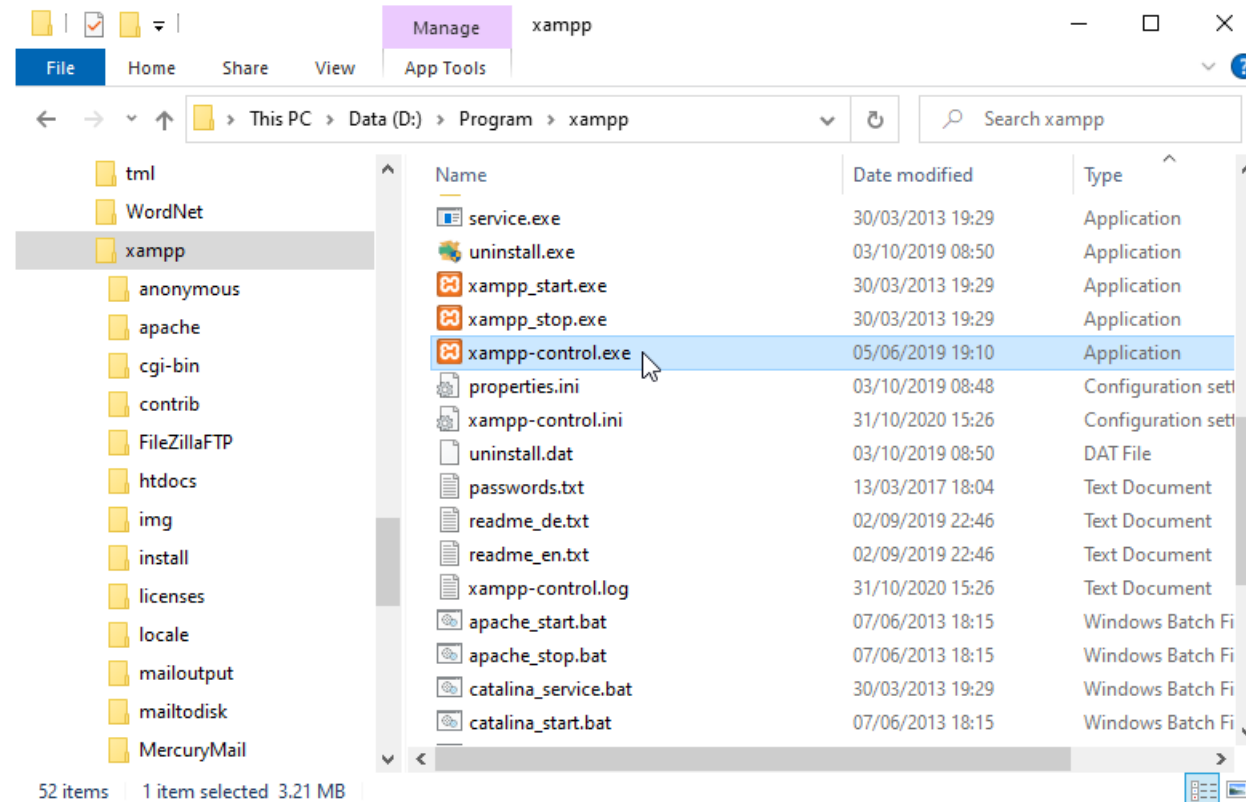
[Requirements](#) [Add-ons](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Version	Checksum	Size
7.2.34 / PHP 7.2.34	What's Included? md5 sha1	151 Mb

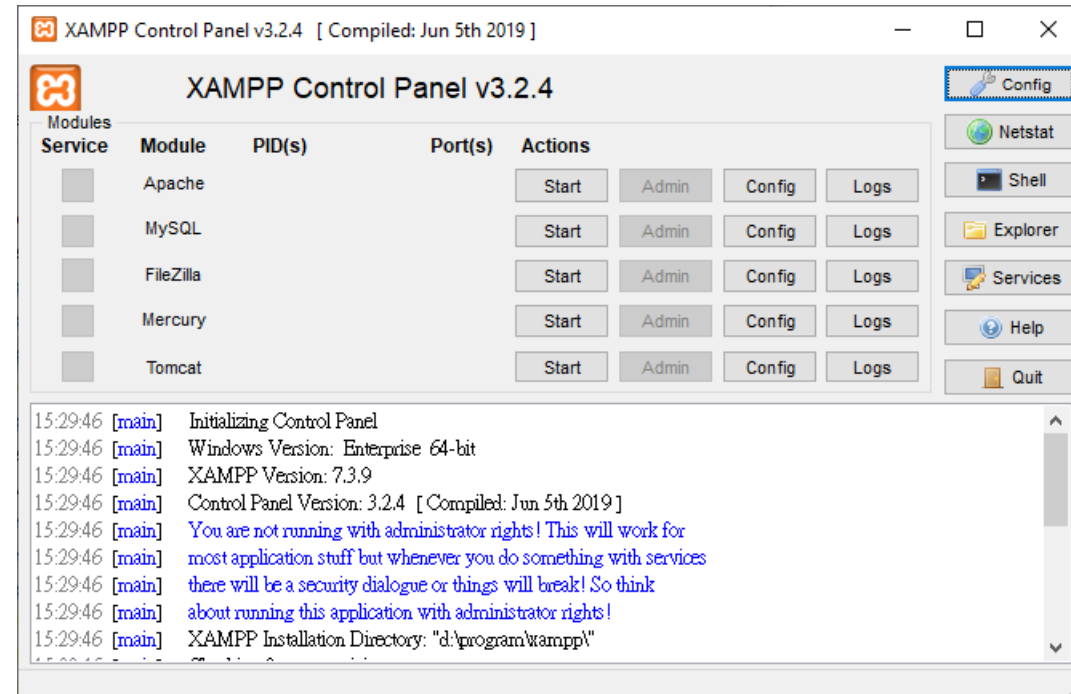
XAMPP: installed

- E.g., install on D:\Program\xampp



XAMPP: running

- Apache → Web server, PHP
- MySQL → DB server
- FileZilla → FTP server
- Mercury → Email server
- Tomcat → Java Servlet, Java Server Pages, Java Expression Language & Java WebSocket technologies.



XAMPP: PHP

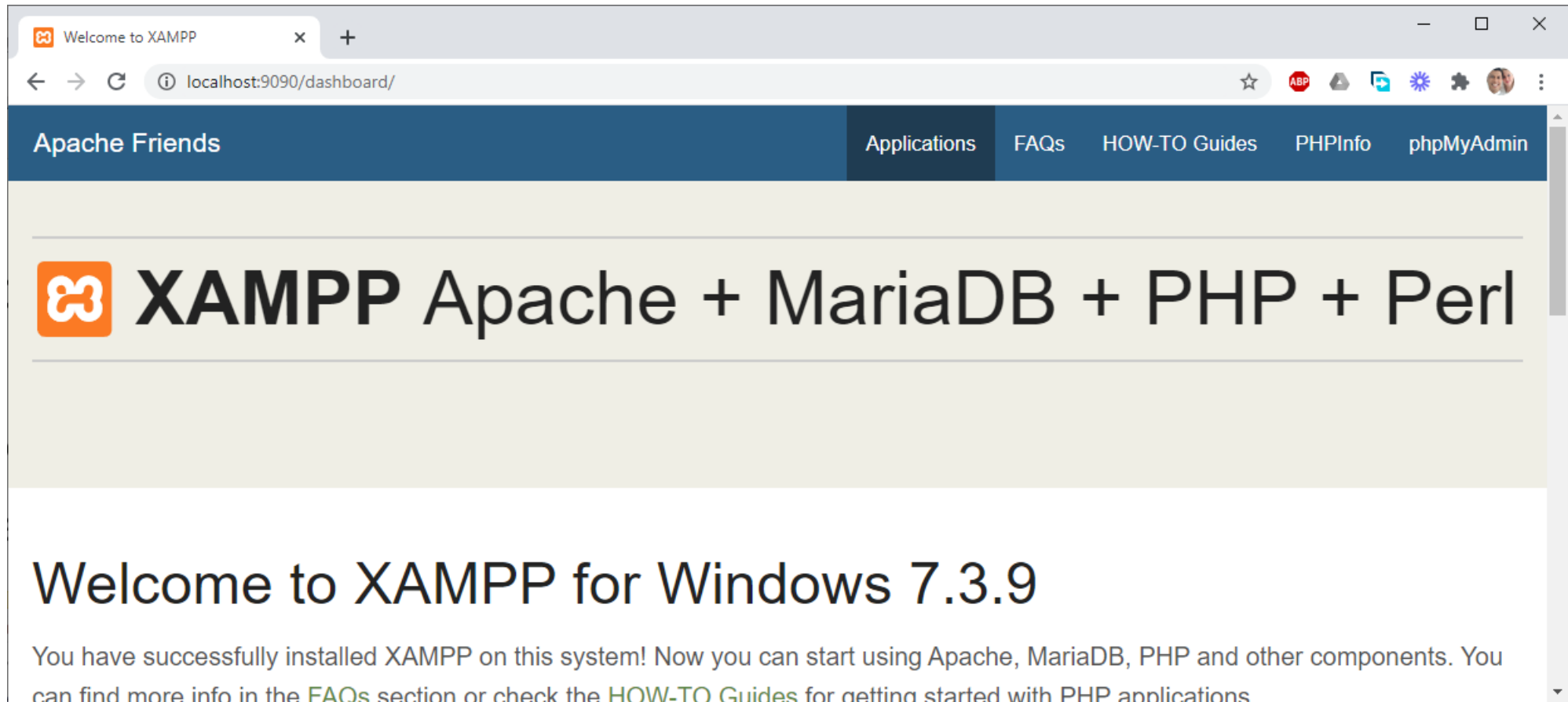
- XAMPP folder, e.g., D:\Program\xampp
- Default ROOT FOLDER: D:\Program\xampp\htdocs

The image shows two windows side-by-side. On the left is a Windows File Explorer window titled 'htdocs' showing the directory structure of the XAMPP installation. The 'htdocs' folder is selected in the left sidebar. The main pane shows a list of files and folders: dashboard, img, piktiweb, public_html, webalizer, webpro, xampp, applications.html, bitnami.css, favicon.ico, and index.php. A blue arrow points from the 'index.php' file to a blue callout box on the right. On the right is the XAMPP Control Panel v3.2.4 window, which shows a table of running services. The Apache service is listed as running on port 443 and 9090. A blue arrow points from the 'index.php' file in the File Explorer to the 'Stop' button in the XAMPP Control Panel.

Service	Module	PID(s)	Port(s)	Actions
Apache	Apache	4800 17572	443, 9090	Stop

Browser will load index.php → redirecting to /dashboard

XAMPP: PHP (continued)



XAMPP: PHP + MySQL/MariaDB

The image displays two overlapping screenshots from a Windows desktop. The background screenshot shows the XAMPP Control Panel v3.2.4 interface. At the top, it says 'XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]'. Below that, there's a table of modules:

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	4800 17572	443, 9090	Stop
<input type="checkbox"/>	MySQL	20132	3306	Stop

The foreground screenshot shows the phpMyAdmin interface in a browser window. The address bar shows 'localhost:9090/phpmyadmin/'. The main content area displays 'XAMPP Apache MariaDB + PHP + P...'. On the left, there's a tree view of databases:

- New
- f_irfan_db
- information_schema
- mysql
- performance_schema
- phpmyadmin
- test

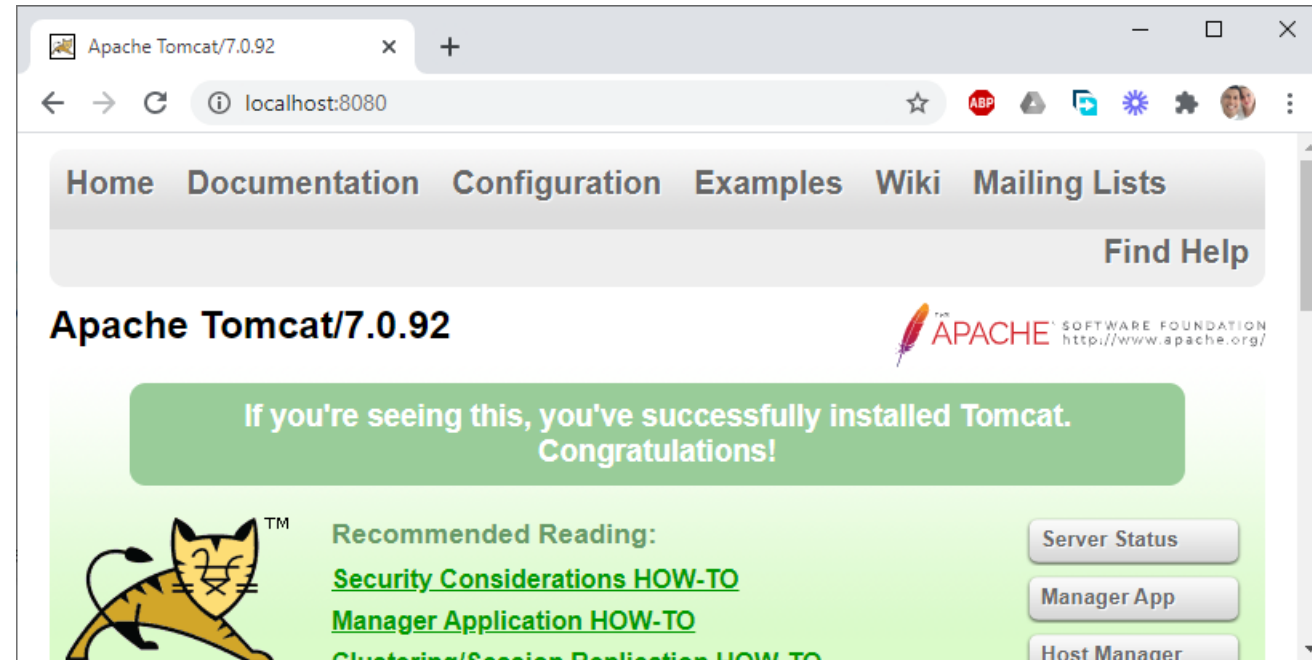
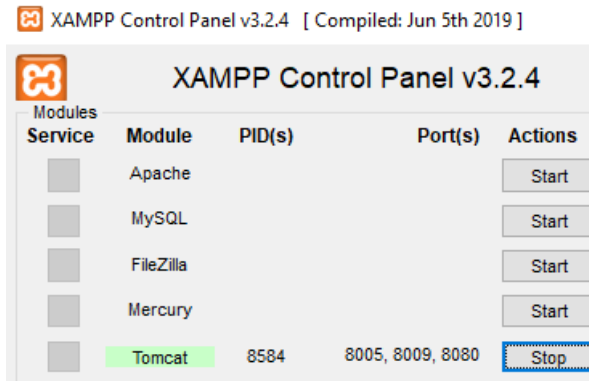
The right side of the interface shows 'Server: 127.0.0.1' with tabs for 'Databases', 'SQL', 'Status', and 'More'. The 'Databases' tab is active, showing 'General settings' with 'Server connection collation' set to 'utf8mb4_unicode_ci'. Below that is 'Appearance settings'. On the far right, a 'Database server' section lists:

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version:

- MySQL folder: D:\Program\xampp\phpMyAdmin

XAMPP: JSP

- Run Tomcat



- Default ROOT FOLDER: D:\Program\xampp\tomcat\webapps
- After XAMPP installation → there is no user!
 - Edit: D:\Program\xampp\tomcat\conf\tomcat-users.xml

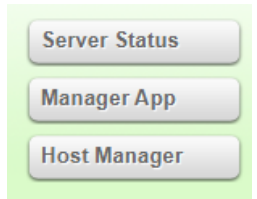
XAMPP: JSP (continued)

- Add the users:

```
tomcat-users.xml
22  you must define such a user - the username and password are arbitrary. It is
23  strongly recommended that you do NOT use one of the users in the commented out
24  section below since they are intended for use with the examples web
25  application.
26  -->
27  <!--
28  NOTE: The sample user and role entries below are intended for use with the
29  examples web application. They are wrapped in a comment and thus are ignored
30  when reading this file. If you wish to configure these users for use with the
31  examples web application, do not forget to remove the <!-- ..> that surrounds
32  them. You will also need to set the passwords to something appropriate.
33  -->
34
35  <role rolename="tomcat"/>
36  <role rolename="role1"/>
37  <user username="tomcat" password="tomcat" roles="tomcat"/>
38  <user username="both" password="tomcat" roles="tomcat,role1"/>
39  <user username="role1" password="tomcat" roles="role1"/>
40
41  <!-- Add this user -->
42  <role rolename="manager-gui"/>
43  <role rolename="admin-gui"/>
44  <role rolename="manager-script"/>
45  <role rolename="manager-jmx"/>
46  <user username="admin" password="admin" roles="tomcat,manager-gui,admin-gui,manager-script,manager-jmx"/>
47
48
49  </tomcat-users>
```

XAMPP: JSP (continued)

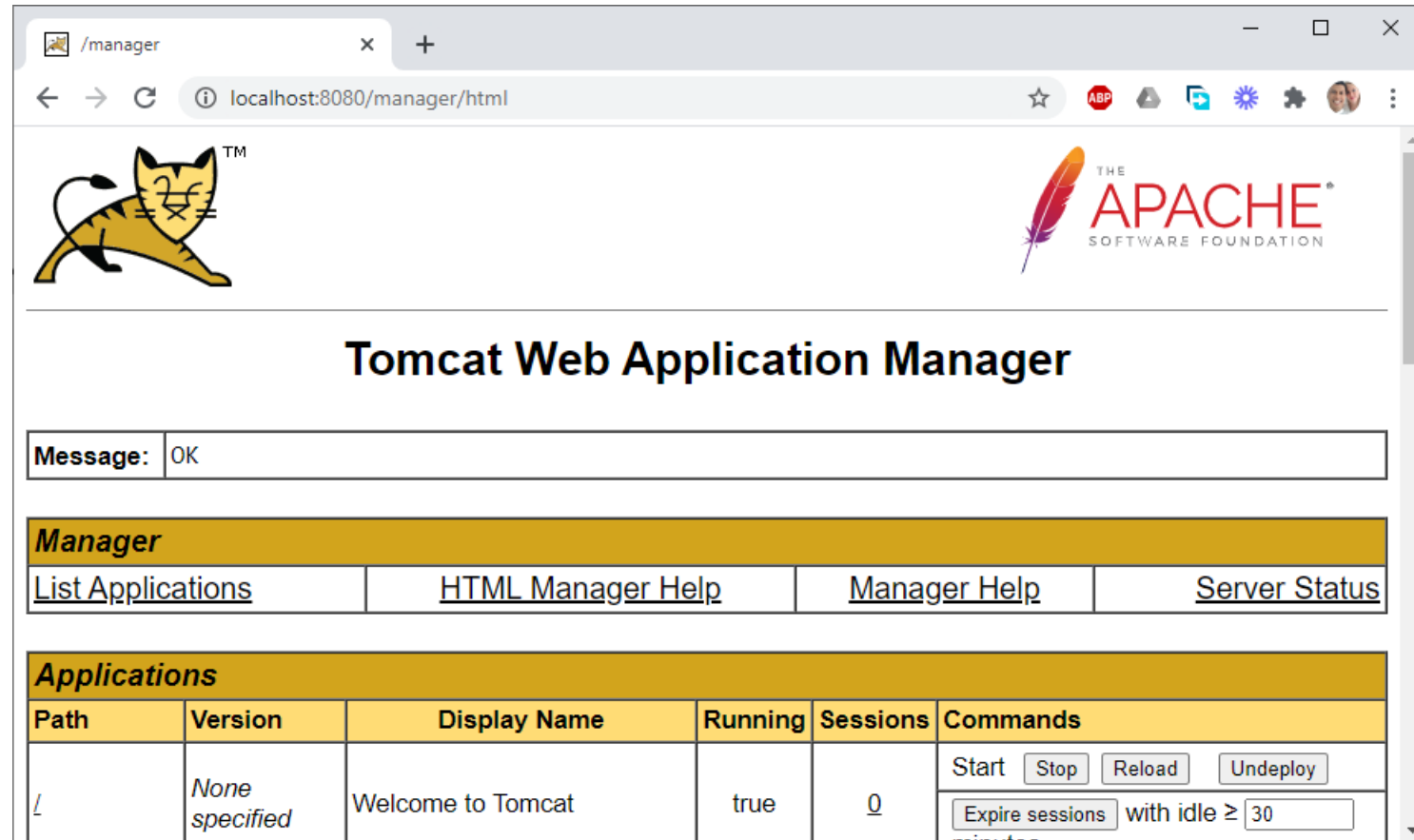
- Now we can access: Server Status, Manager App, Host Manager

A screenshot of a web browser window showing the Apache Manager interface. The browser's address bar displays 'localhost:8080/manager/status'. The page features the Tomcat logo on the left and the Apache Software Foundation logo on the right. The main heading is 'Server Status'. Below this, there is a yellow bar with the word 'Manager' and four links: 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Complete Server Status'. A section titled 'Server Information' contains a table with server details.

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.92	1.8.0_271-b09	Oracle Corporation	Windows 10	10.0	amd64	IrfanLaptop	192.168.1.4

XAMPP: JSP (continued)

- Manager App

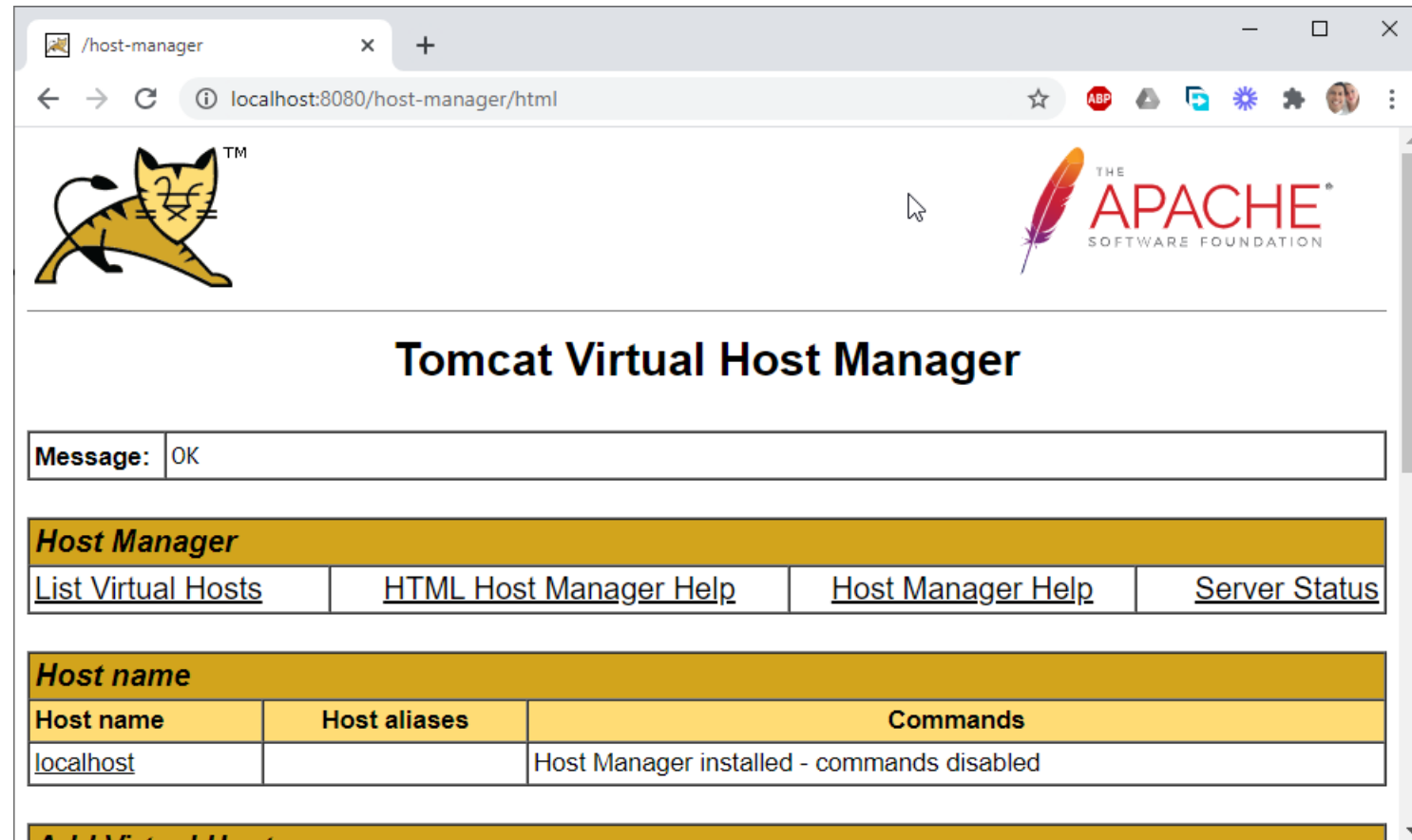


The screenshot shows a web browser window with the URL `localhost:8080/manager/html`. The page features the Tomcat logo (a yellow cat) and the Apache Software Foundation logo. The main heading is "Tomcat Web Application Manager". Below the heading, there is a "Message: OK" box. A navigation bar contains links for "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The "Applications" section is highlighted in yellow and contains a table with the following data:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ <input type="text" value="30"/> minutes

XAMPP: JSP (continued)

- Host Manager



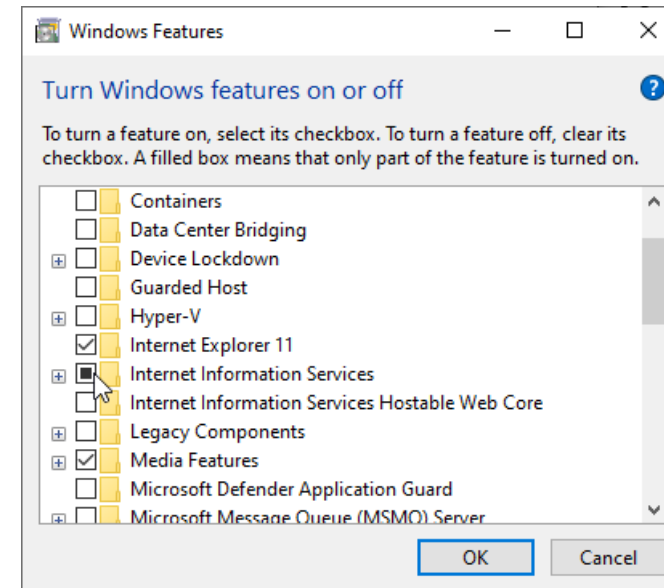
The screenshot shows a web browser window with the address bar displaying `localhost:8080/host-manager/html`. The page features the Tomcat logo (a yellow cat) and the Apache Software Foundation logo. The main heading is "Tomcat Virtual Host Manager". Below the heading, there is a "Message: OK" box. A navigation bar contains links for "List Virtual Hosts", "HTML Host Manager Help", "Host Manager Help", and "Server Status". A table displays the current host configuration:

Host name		
Host name	Host aliases	Commands
localhost		Host Manager installed - commands disabled

At the bottom of the page, there is a partially visible link for "Add Virtual Host".

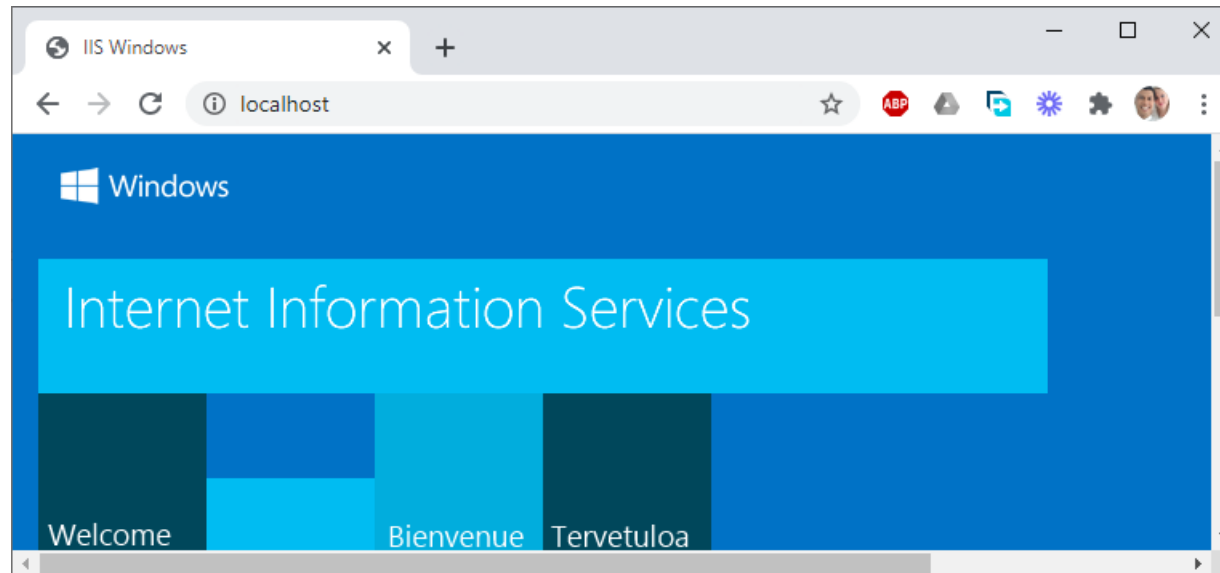
Internet Information Service (IIS)

- It needs to be activated
 - Menu > Control Panel > Program > Turn Windows features on or off > Windows Features > Internet Information Services > Check it!
 - Once it's checked, IIS will be activated!



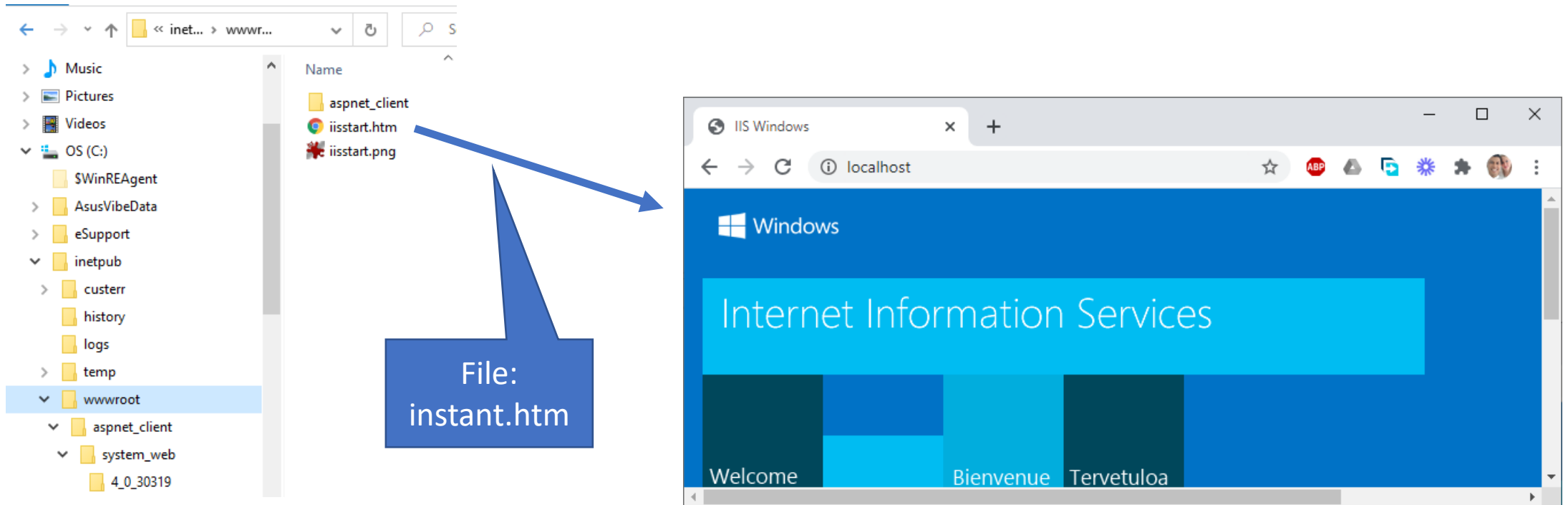
Internet Information Service (IIS) - continued

- Check whether IIS works
 - Open your internet browser
 - Type “localhost” on address bar
 - If you can see this, it’s mean IIS works!



Internet Information Service (IIS) - continued

- Default ROOT FOLDER: C:\inetpub\wwwroot



Internet Information Service (IIS) - continued

- Since drive C:\ usually is used by OS (i.e., access rights issues) then it's better to use other folder for our default ROOT FOLDER
 - E.g., D:\inetpub\wwwroot
- How to change the default ROOT FOLDER?
 - Run Internet Information Service (IIS) Manager
 - Click on Default Web Site > Manage Website > Advanced Settings...

Internet Information Service (IIS) - continued

- Choose the folder for our new default ROOT FOLDER

The image shows the Internet Information Services (IIS) Manager interface. The main window displays the 'Default Web Site Home' with a context menu open over it. The 'Advanced Settings...' option is selected, leading to the 'Advanced Settings' dialog box. A blue arrow points from the 'Advanced Settings...' option in the context menu to the 'Advanced Settings' dialog box.

The 'Advanced Settings' dialog box is open, showing the 'Physical Path' field set to 'D:\inetpub\wwwroot'. The 'Physical Path' field is highlighted, and a blue arrow points to it from the context menu in the IIS Manager window.

Advanced Settings	
(General)	
Application Pool	DefaultAppPool
Bindings	http*:80:
ID	1
Name	Default Web Site
Physical Path	D:\inetpub\wwwroot
Physical Path Credentials	
Physical Path Credentials Logon	ClearText
Preload Enabled	False
Behavior	
Enabled Protocols	http
HSTS	
Limits	

Physical Path
[physicalPath] Physical path to the content for the virtual directory.

OK Cancel

ASP.NET: Tutorial

- There are a lot of tutorials for mastering ASP.NET
 - E.g., can be seen at: <https://www.w3schools.com/asp/default.ASP>

ASP.NET Core: Tutorial

- From its creator, there are tutorials for mastering ASP.NET Core
 - It can be seen at: <https://docs.microsoft.com/en-gb/aspnet/core>



Installation

- .NET Core 3.1 SDK

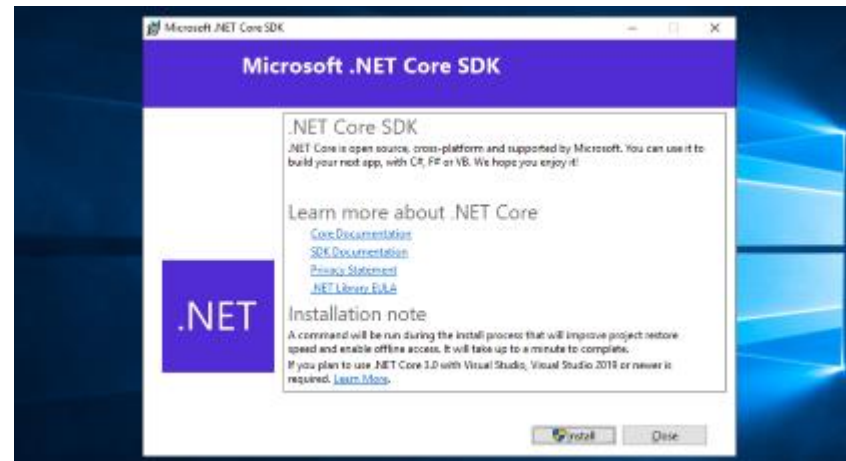


- Visual Studio 2019 with the ASP.NET and web development workload



.NET Core 3.1 SDK

- Step 1: Run Installer
 - When your download completes, run the installer and complete the steps to install .NET on your machine.



.NET Core 3.1 SDK (continued)

- Step 2: Verify Installation
 - When the installer completes, open a new command prompt and run the `dotnet` command. This will verify .NET is correctly installed and ready to use.



```
Microsoft Windows
(c) 2017 Microsoft Corporation. All rights reserved.

C:\>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

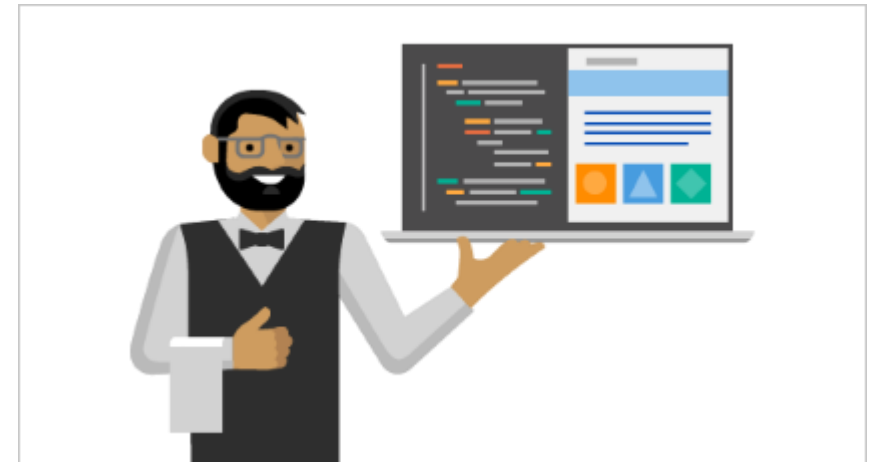
Options:
  -h|--help          Display help.
  --version          Display version.

path-to-application:
  The path to an application .dll file to execute.

C:\>_
```

.NET Core 3.1 SDK (continued)

- Step 3: Get Started
 - Now that you've got .NET installed, let's build your first app!



Hello World

- To start building .NET apps, download and install the .NET SDK (Software Development Kit).



- Check everything installed correctly
 - Once you've installed, open a new command prompt and run the following command:

```
Command prompt Copy  
> dotnet
```

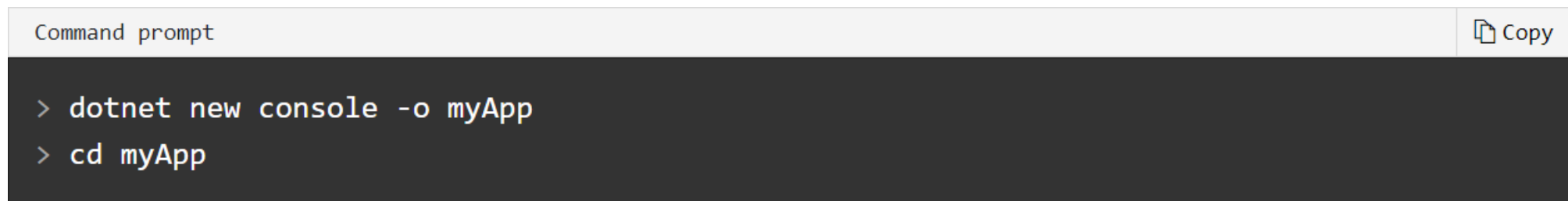
- If the command runs, printing out information about how to use dotnet, you're good to go.

Hello World (continued)

- Got an error?
 - If you receive a 'dotnet' is not recognized as an internal or external command error, make sure you opened a new command prompt.

Hello World (continued)

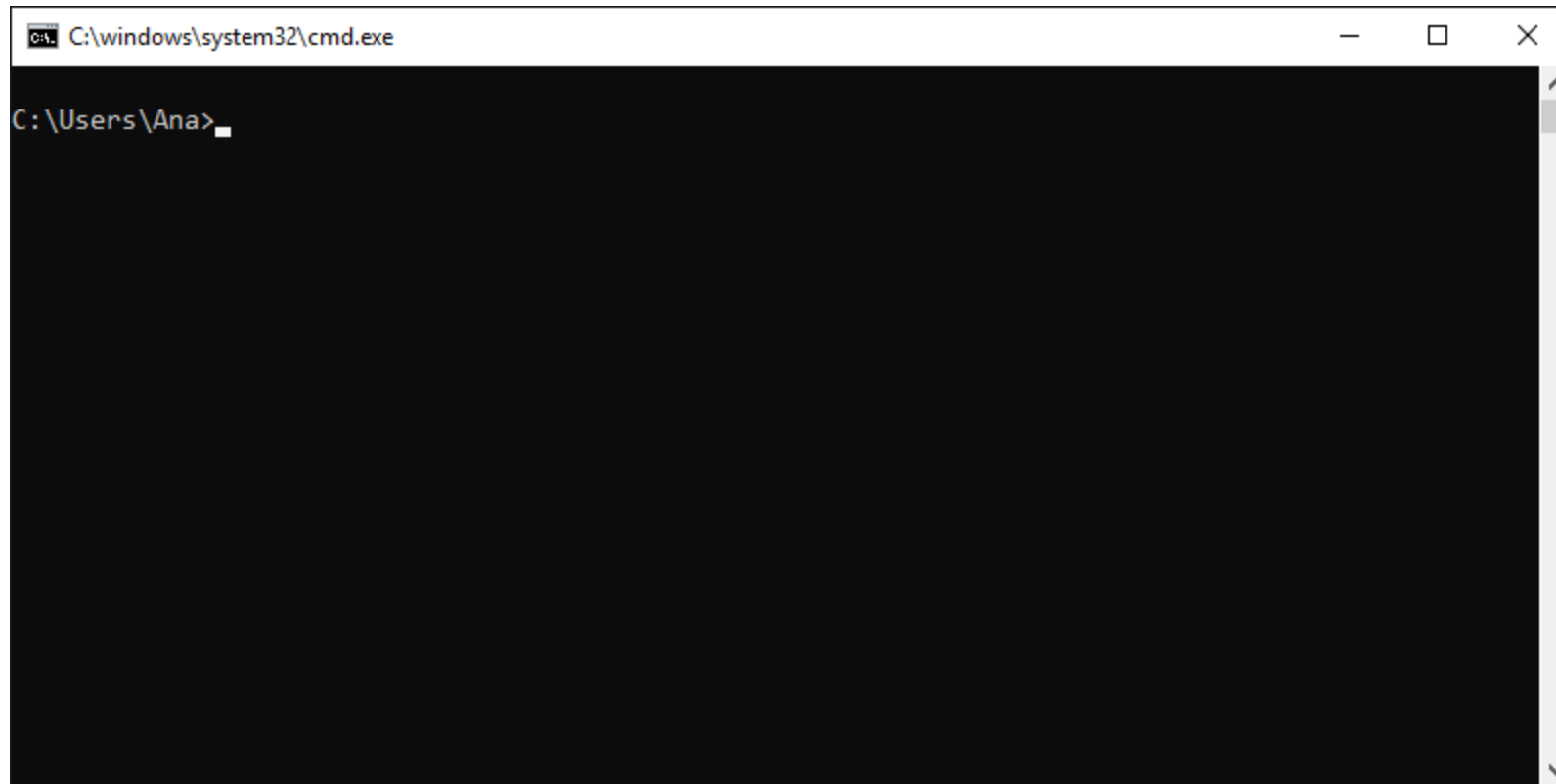
- Create your app
 - In your command prompt, run the following commands:



```
Command prompt Copy  
> dotnet new console -o myApp  
> cd myApp
```

- The `dotnet` command creates a `new` application of type `console` for you. The `-o` parameter creates a directory named `myApp` where your app is stored, and populates it with the required files. The `cd myApp` command puts you into the newly created app directory.

Hello World (continued)



```
C:\windows\system32\cmd.exe
C:\Users\Ana>
```

A screenshot of a Windows Command Prompt window. The title bar at the top reads "C:\windows\system32\cmd.exe" and includes standard window control buttons (minimize, maximize, close). The main area of the window is black with white text. The prompt "C:\Users\Ana>" is visible at the top left, followed by a small white cursor. The right side of the window has a vertical scrollbar.

Hello World (continued)

Program.cs

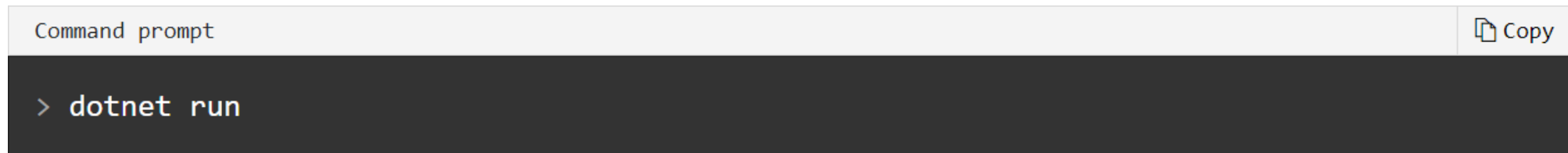
```
using System;

namespace myApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

- The main file in the `myApp` folder is `Program.cs`. By default, it already contains the necessary code to write "Hello World!" to the Console.

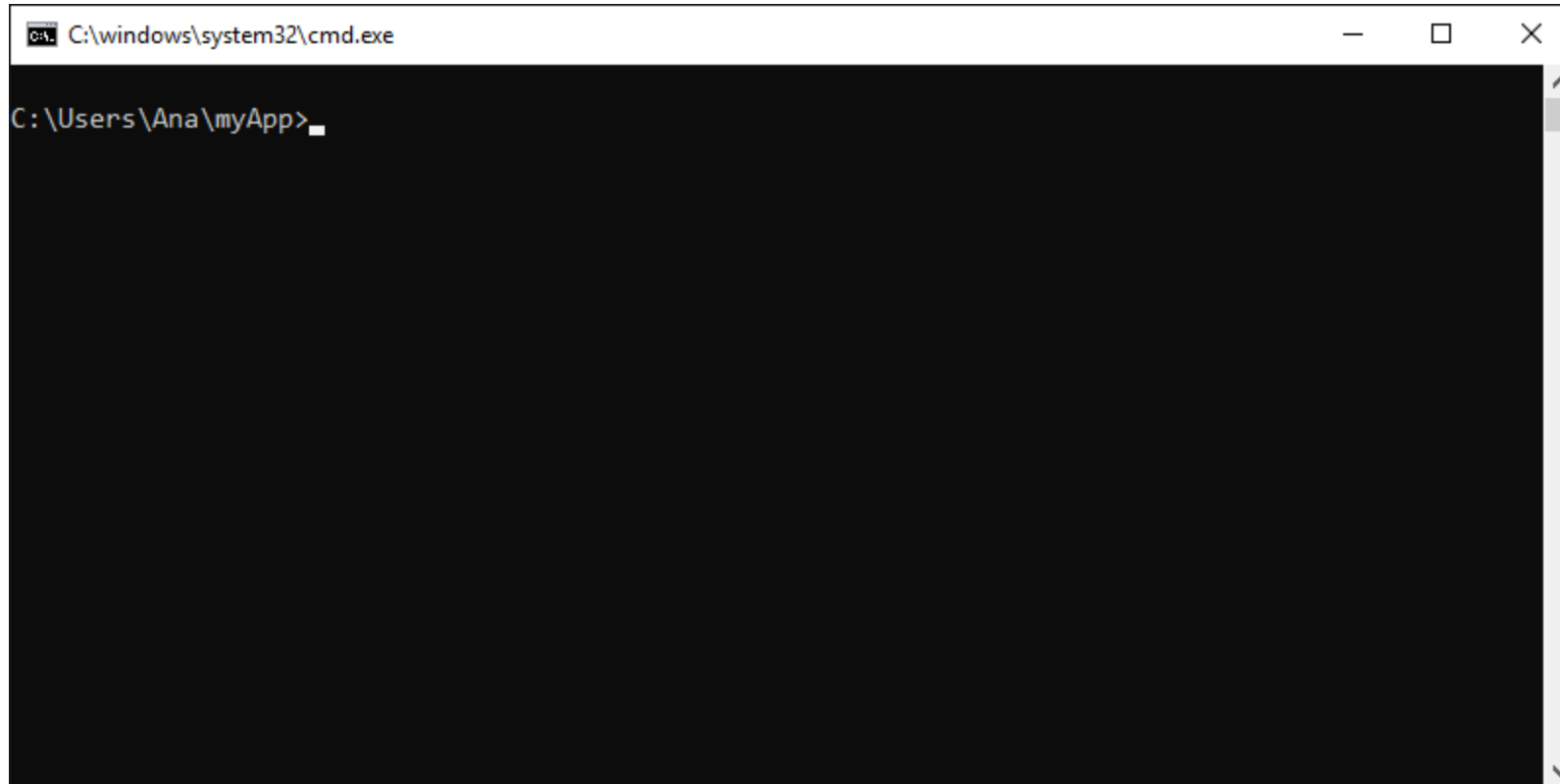
Hello World (continued)

- Run your app
 - In your command prompt, run the following command:

A screenshot of a Windows command prompt window. The title bar at the top reads "Command prompt" and has a "Copy" button on the right. The main area of the window is dark gray and contains the text "> dotnet run" in white, indicating the command being entered.

```
> dotnet run
```

Hello World (continued)

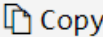


A screenshot of a Windows Command Prompt window. The title bar at the top reads "C:\windows\system32\cmd.exe". The main area of the window is black with white text. The prompt shows the current directory as "C:\Users\Ana\myApp>".

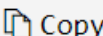
- Congratulations, you've built and run your first .NET app!

Hello World (continued)

- Edit your code
 - Open [Program.cs](#) in any text editor, such as Notepad, and add a new line of code below the one that prints "Hello World!", like the following:

```
Program.cs (shortened for clarity)   
  
Console.WriteLine("Hello World!");  
Console.WriteLine("The current time is " + DateTime.Now);
```

- Save the [Program.cs](#) file, and run your code again.

```
Command prompt   
  
> dotnet run
```

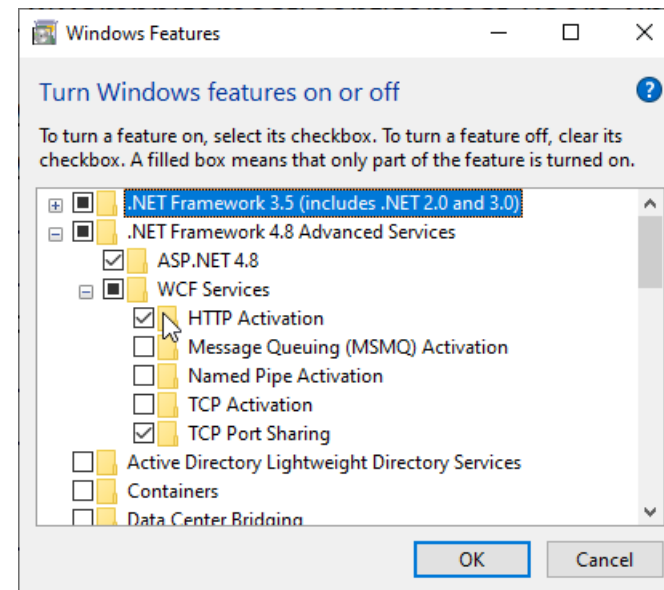
Build a web application

- Create your app
 - In your command prompt, run the following commands:

```
Command prompt Copy  
> dotnet new webApp -o myWebApp --no-https  
> cd myWebApp
```


Windows Communication Foundation (WCF)

- HTTP needs to be activated
 - Menu > Control Panel > Program > Turn Windows features on or off > Windows Features > .NET Framework 4.8 Advanced Services > WCF Services > HTTP Activation > Check it!
 - Once it's checked, HTTP will be activated!



Build a web application (continued)

- What do these commands mean?

```
Command prompt Copy  
> dotnet new webApp -o myWebApp --no-https  
> cd myWebApp
```

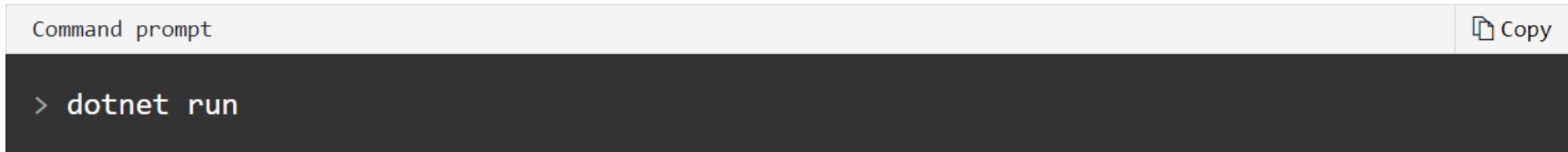
- The `dotnet new` command creates a new application.
 - The `webApp` parameter selects what template to use when creating your app.
 - The `-o` parameter creates a directory named `myWebApp` where your app is stored.
 - The `--no-https` flag specifies not to enable HTTPS.
- The `cd myWebApp` command puts you into the newly created app directory.

- What files were created?

- Several files were created in the `myWebApp` directory, to give you a simple web application that is ready to run.
 - `Startup.cs` contains all the settings and configurations.
 - The `myWebApp/Pages` directory contains some example web pages for the application.
 - `myWebApp.csproj` defines what libraries are referenced etc.

Build a web application (continued)

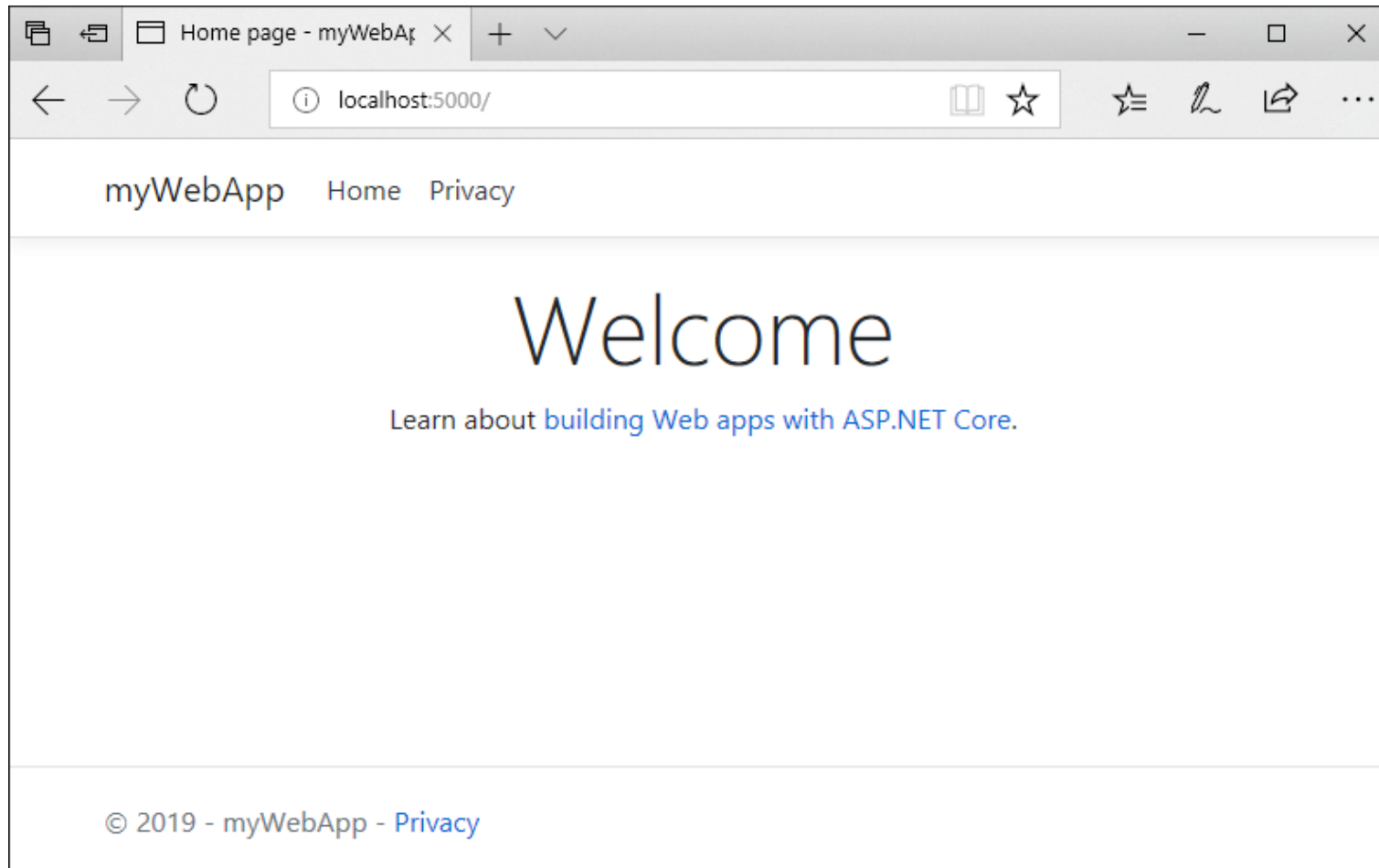
- Run your app
 - In your command prompt, run the following command:

A screenshot of a Windows command prompt window. The title bar reads "Command prompt" and there is a "Copy" button in the top right corner. The command prompt shows a prompt character ">" followed by the command "dotnet run".

```
Command prompt Copy  
> dotnet run
```

- Once the command completes, browse to <http://localhost:5000>

Build a web application (continued)



Congratulations,
you've built and run
your first .NET web
app!

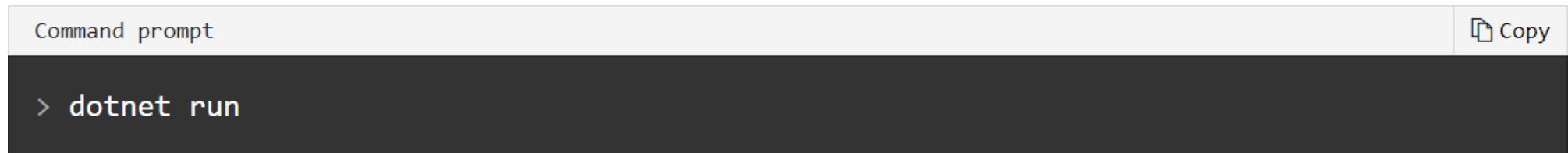
Build a web application (continued)

- Edit your code
 - Open [Pages/Index.cshtml](#) in any text editor and replace all of the code with the following, then save the file.

```
Index.cshtml Copy  
  
@page  
@model IndexModel  
@{  
    ViewData["Title"] = "Home page";  
}  
  
<div class="text-center">  
    <h1>Hello, world!</h1>  
    <p>The time on the server is @DateTime.Now</p>  
</div>
```

Build a web application (continued)

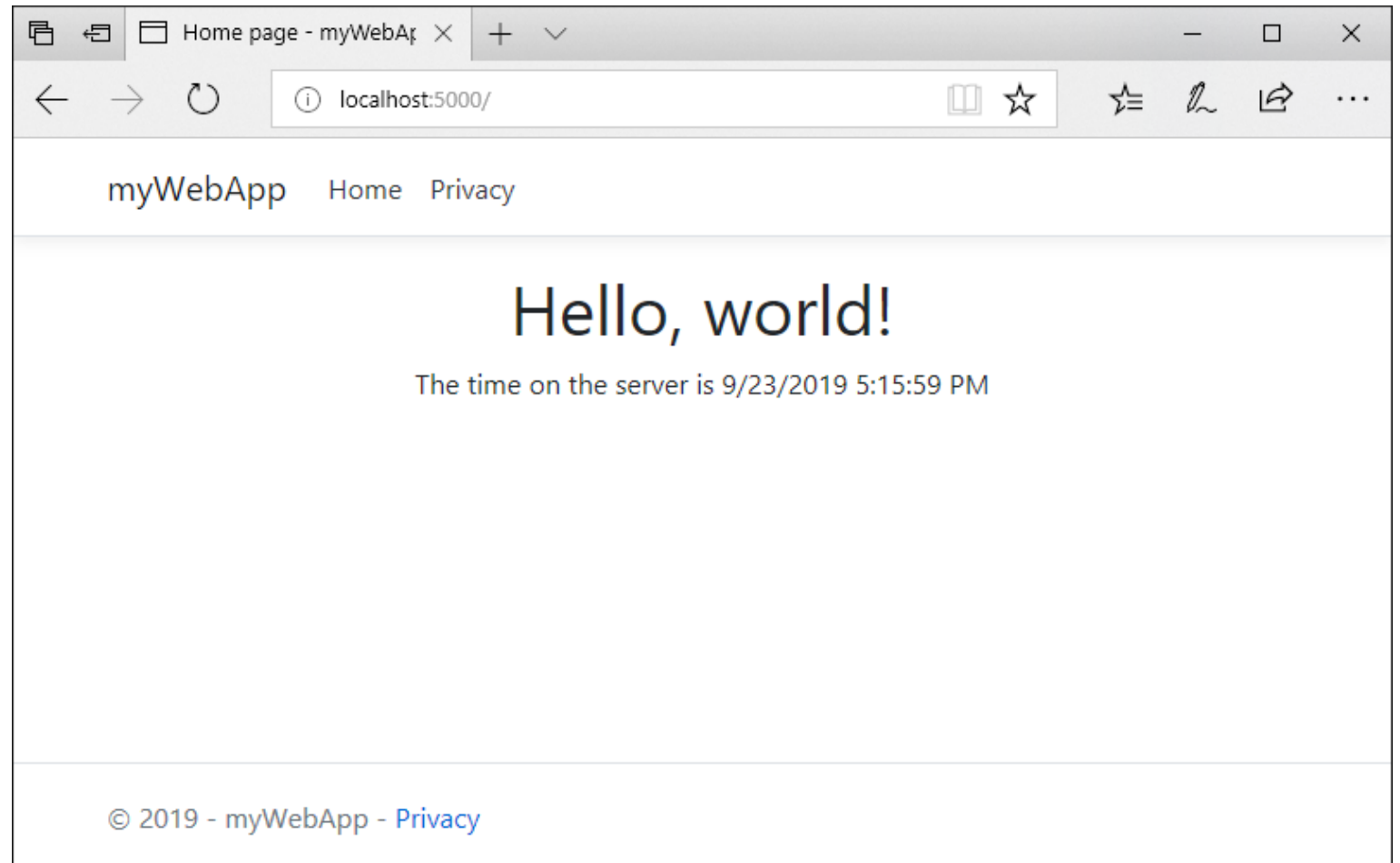
- Re-run your app
 - End the previous `dotnet run` command that is running the site locally, then run the following command to re-launch the site:

A screenshot of a Windows command prompt window. The title bar reads "Command prompt" and there is a "Copy" button in the top right corner. The main area of the window is dark with white text showing a prompt character followed by the command "dotnet run".

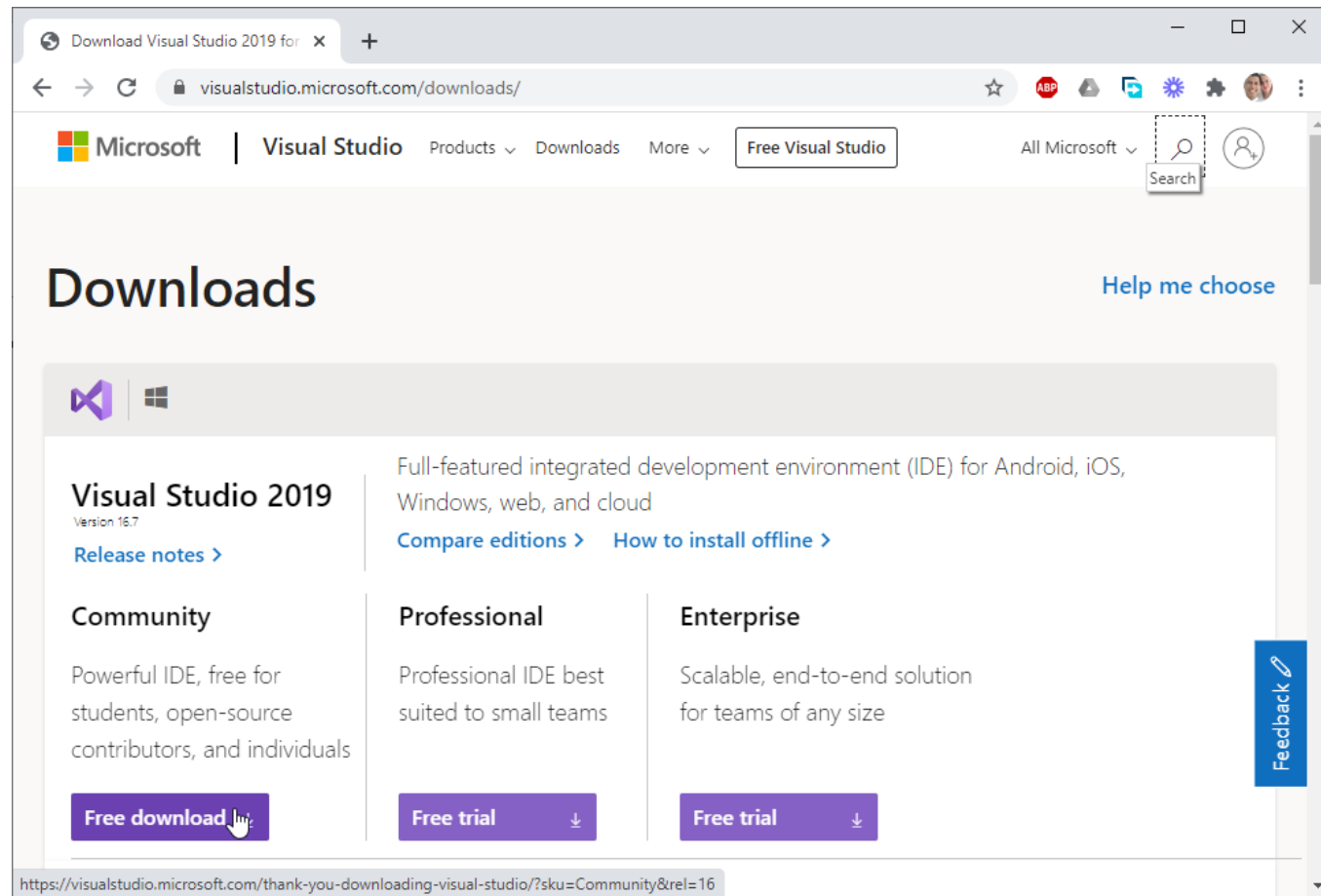
```
> dotnet run
```

Build a web application (continued)

- Refresh the browser to see the change:



Visual Studio 2019: installation

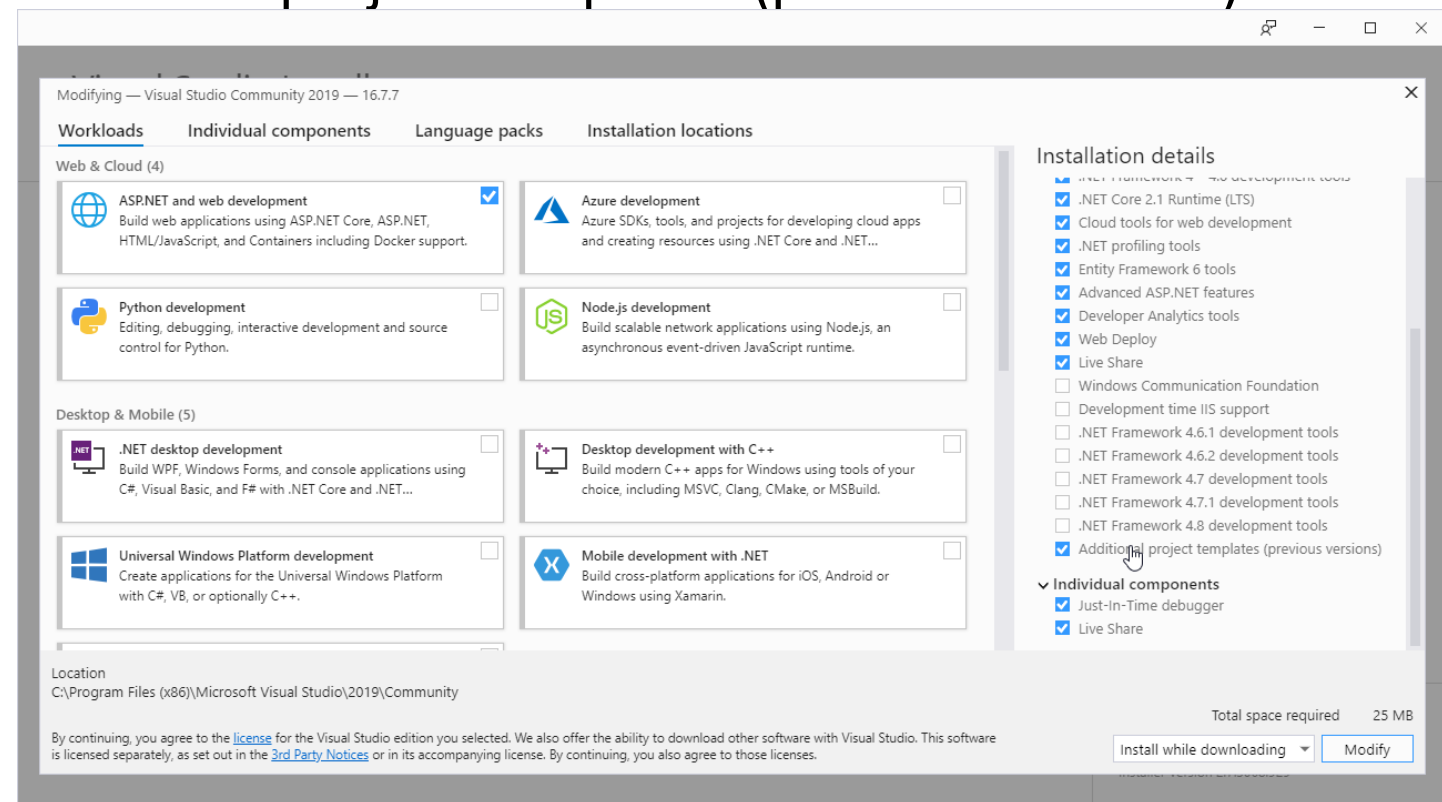


The screenshot shows a web browser window with the URL `visualstudio.microsoft.com/downloads/`. The page features the Microsoft logo and navigation links for 'Visual Studio', 'Products', 'Downloads', and 'More'. A 'Free Visual Studio' button is visible in the top right. The main heading is 'Downloads', with a 'Help me choose' link. Below this, there is a section for 'Visual Studio 2019' (Version 16.7) with a 'Release notes >' link. The page is divided into three columns: 'Community' (described as a powerful IDE free for students, open-source contributors, and individuals, with a 'Free download' button), 'Professional' (described as a professional IDE best suited to small teams, with a 'Free trial' button), and 'Enterprise' (described as a scalable, end-to-end solution for teams of any size, with a 'Free trial' button). A 'Feedback' button is located on the right side. The browser's address bar shows the URL `https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16`.

Visual Studio 2019: installation (continued)



- When install/modify Visual Studio 2019
 - Don't forget to tick Additional project templates (previous versions)



Get started with Razor Pages in ASP.NET Core

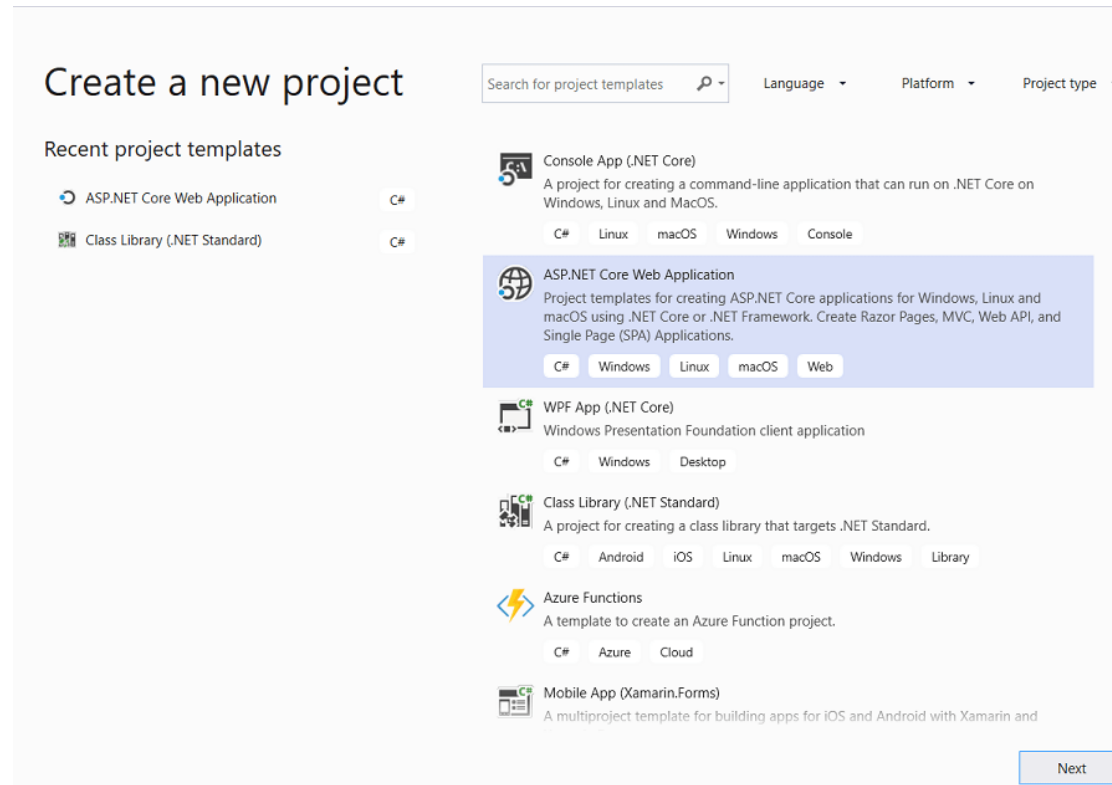
- The basics of building an ASP.NET Core Razor Pages web app.
- You'll have an app that manages a database of movies.
 - Create a Razor Pages web app.
 - Run the app.
 - Examine the project files.

Razor Pages web app

- Prerequisites
 - Visual Studio
 - **Visual Studio 2019** 16.4 or later with the ASP.NET and web development workload
 - **.NET Core 3.1 SDK** or later

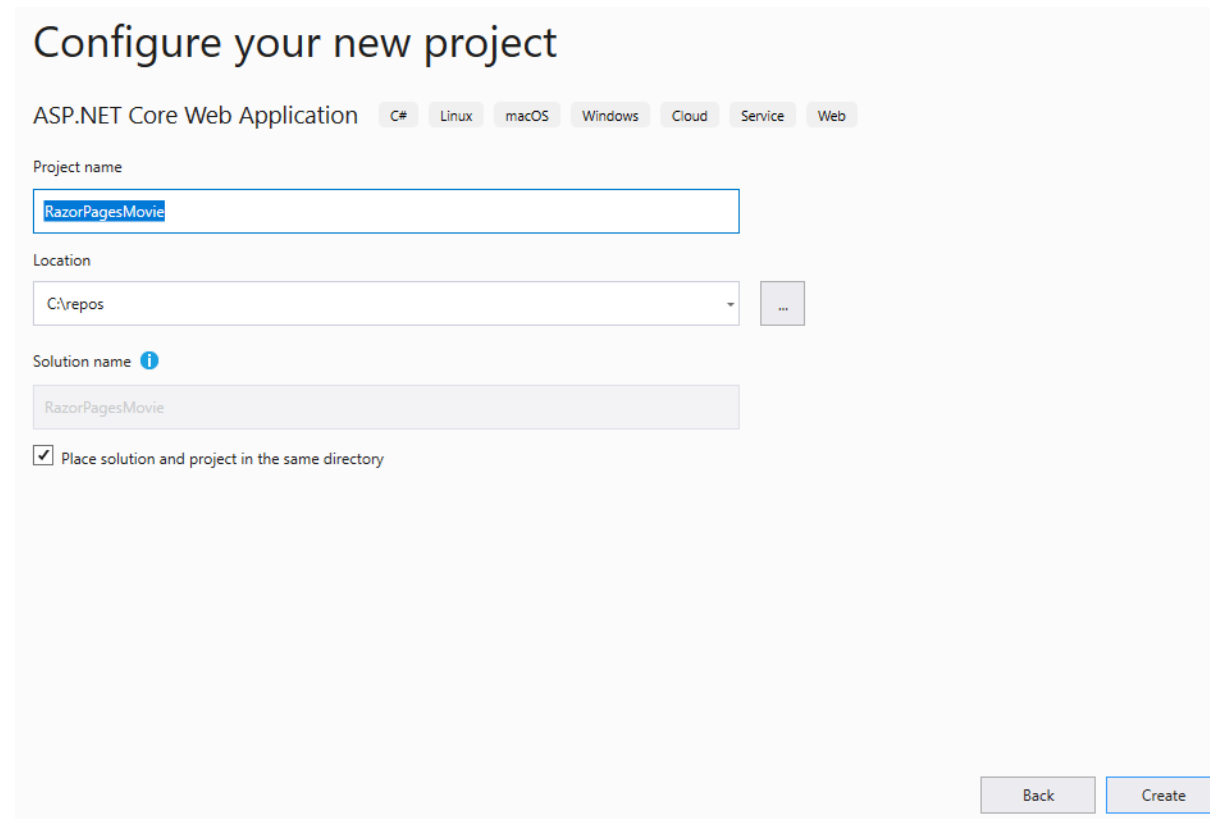
Razor Pages web app (continued)

- From the Visual Studio **File** menu, select **New > Project**.
- Create a new ASP.NET Core Web Application and select **Next**.



Razor Pages web app (continued)

- Name the project **RazorPagesMovie**. It's important to name the project *RazorPagesMovie* so the namespaces will match when you copy and paste code.



The screenshot shows the 'Configure your new project' dialog in Visual Studio. The project type is 'ASP.NET Core Web Application' with 'C#' selected. The project name is 'RazorPagesMovie'. The location is 'C:\repos'. The solution name is 'RazorPagesMovie'. The checkbox 'Place solution and project in the same directory' is checked. The 'Create' button is highlighted.

Configure your new project

ASP.NET Core Web Application C# Linux macOS Windows Cloud Service Web

Project name

RazorPagesMovie

Location

C:\repos

Solution name ⓘ

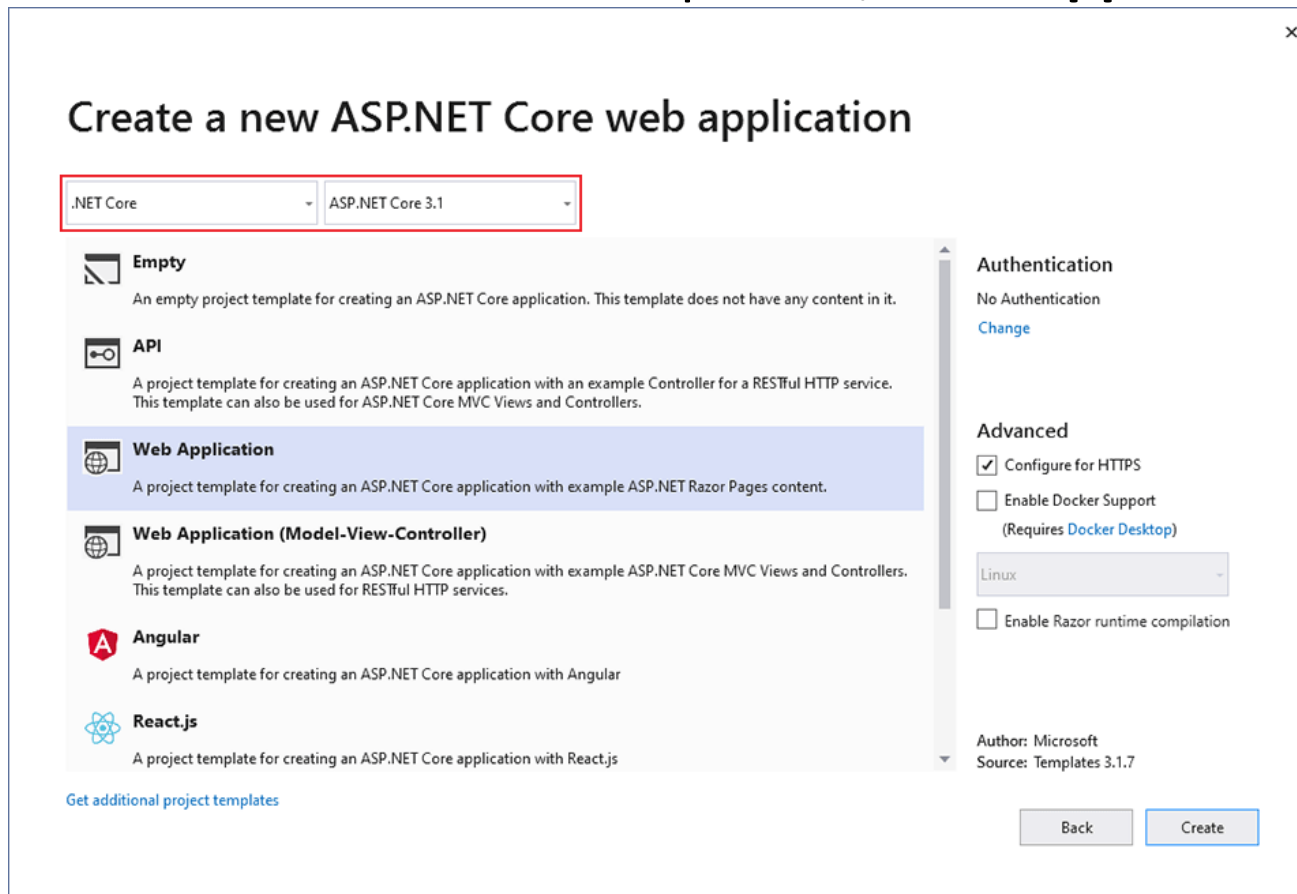
RazorPagesMovie

Place solution and project in the same directory

Back Create

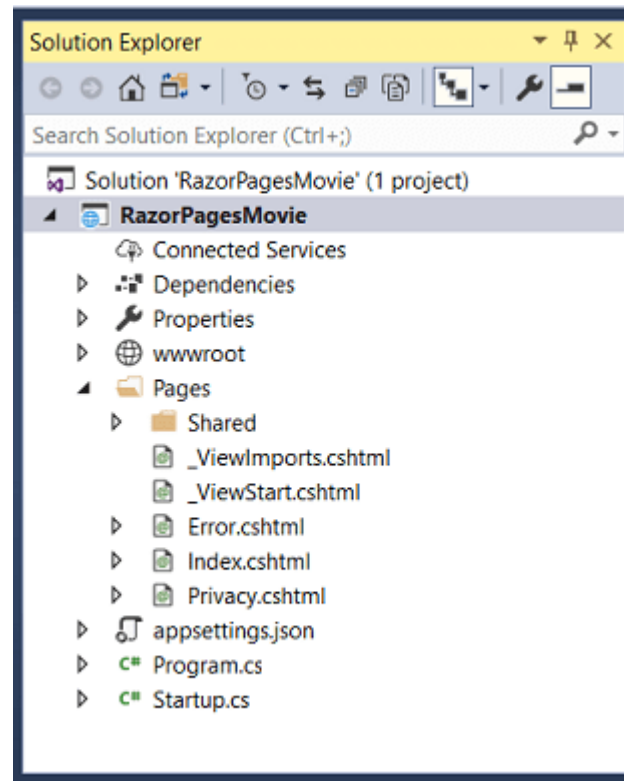
Razor Pages web app (continued)

- Select **ASP.NET Core 3.1** in the dropdown, **Web Application**, and then select **Create**.



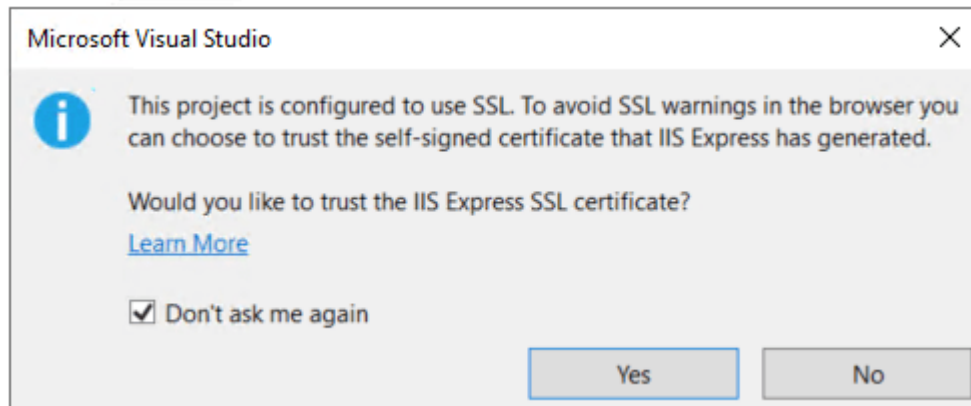
Razor Pages web app (continued)

- The following starter project is created:



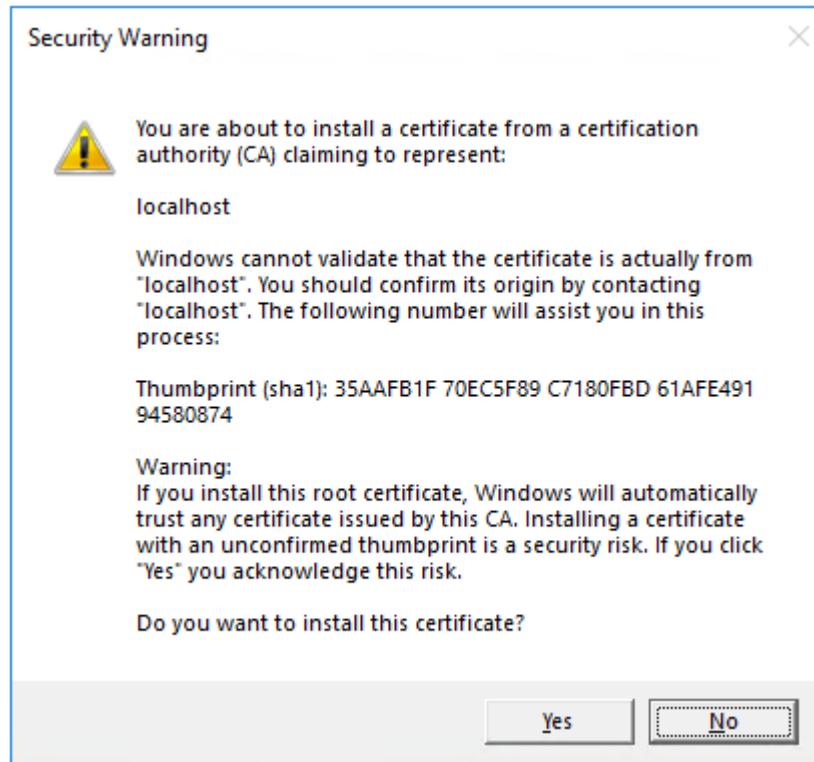
Razor Pages web app (continued)

- Run the app
 - Press Ctrl+F5 to run without the debugger.
 - Visual Studio displays the following dialog:



Razor Pages web app (continued)

- Select Yes if you trust the IIS Express SSL certificate.
- The following dialog is displayed:



Razor Pages web app (continued)

- Select **Yes** if you agree to trust the development certificate.
- Visual Studio starts **IIS Express** and runs the app. The address bar shows **localhost:port#** and not something like **example.com**. That's because **localhost** is the standard hostname for the local computer. Localhost only serves web requests from the local computer. When Visual Studio creates a web project, a random port is used for the web server.

Razor Pages web app (continued)

- Examine the project files
 - Here's an overview of the main project folders and files that you'll work with
- Pages folder
 - Contains Razor pages and supporting files. Each Razor page is a pair of files:
 - A *.cshtml* file that contains HTML markup with C# code using Razor syntax.
 - A *.cshtml.cs* file that contains C# code that handles page events.
 - Supporting files have names that begin with an underscore. For example, the *_Layout.cshtml* file configures UI elements common to all pages. This file sets up the navigation menu at the top of the page and the copyright notice at the bottom of the page.

Razor Pages web app (continued)

- wwwroot folder
 - Contains static files, such as HTML files, JavaScript files, and CSS files.
- appSettings.json
 - Contains configuration data, such as connection strings.
- Program.cs
 - Contains the entry point for the program.
- Startup.cs
 - Contains code that configures app behaviour.

Razor Pages web app (continued)

- The full tutorial can be seen at:

<https://docs.microsoft.com/en-gb/aspnet/core/tutorials/razor-pages>