

## ASP.NET (Active Server Pages)

- It is a development **framework** for building web pages.
- **server side** tech.
- ASP.NET 4.6 → latest official version of ASP.NET.

ASP supports many different development models:

- Classic.ASP
- **ASP.NET web forms** —→ **event driven** app. model.  
not a part of the new ASP.NET Core.
- **ASP.NET MVC** —→ (Model-View-Controller)  
being merged into the new ASP.NET Core.
- **ASP.NET web pages** —→ SPA app. (**single page app.**)  
SPA model is similar to PHP & Classic ASP.
- **ASP.NET Web Pages** is being merged into the new ASP.NET Core.
- **ASP.NET API** —→ (Application Programming Interface)  
being merged into the new ASP.NET Core.
- **ASP.NET Core** (2016) —→ merges MVC, Web API,.NET Web Pages into one app. framework.

### What is ASP.NET?

ASP.NET is a server-side technology used for developing dynamic web applications. It was released in 2002 and had an extension of .aspx. It produces interactive, data-driven web applications on the internet

Examples:



The image shows three social media icons: Twitter (blue bird), Instagram (camera icon), and Facebook (blue square with white 'f').



The illustration depicts a person sitting cross-legged on a stack of books, with various programming language icons (Python, CSS, PHP, C++, JS, etc.) floating around them.

See the picture this shows; at present, the world deals with a huge number of applications

Using ASP.NET, we can build applications faster and more efficiently compare to other technologies

ASP.NET help in reducing issues related to



Speed



Cost



Language



## Architecture

IIS passes the requests to the ASP.NET engine

ASP.NET engine reads the file and executes the script



Then ASP.NET returns to the browser as Html

## Components

- **Language:-** This can help in developing the web application. Some of them are **VB.net** and **C#**
- **Library:-** The most common library used in ASP.NET is the web library
- **Common Language Runtime:-** This is the platform that is used to execute the programs



## Tools and frameworks



Microsoft Web  
Platform Installer

A free **tool** that can be **easier to run web applications**

## Tools and frameworks



Microsoft Web  
Platform Installer



SQL Complete

A customizable tool that **improves the quality of the code** and  
**simplifies the creation of SQL queries**

## Tools and frameworks



Microsoft Web Platform Installer



SQL Complete



Web Essentials for Visual Studio

A tool that opens up the inventory of CSS, Html, JavaScript, and some others

simplilearn

## Tools and frameworks



Microsoft Web Platform Installer



SQL Complete



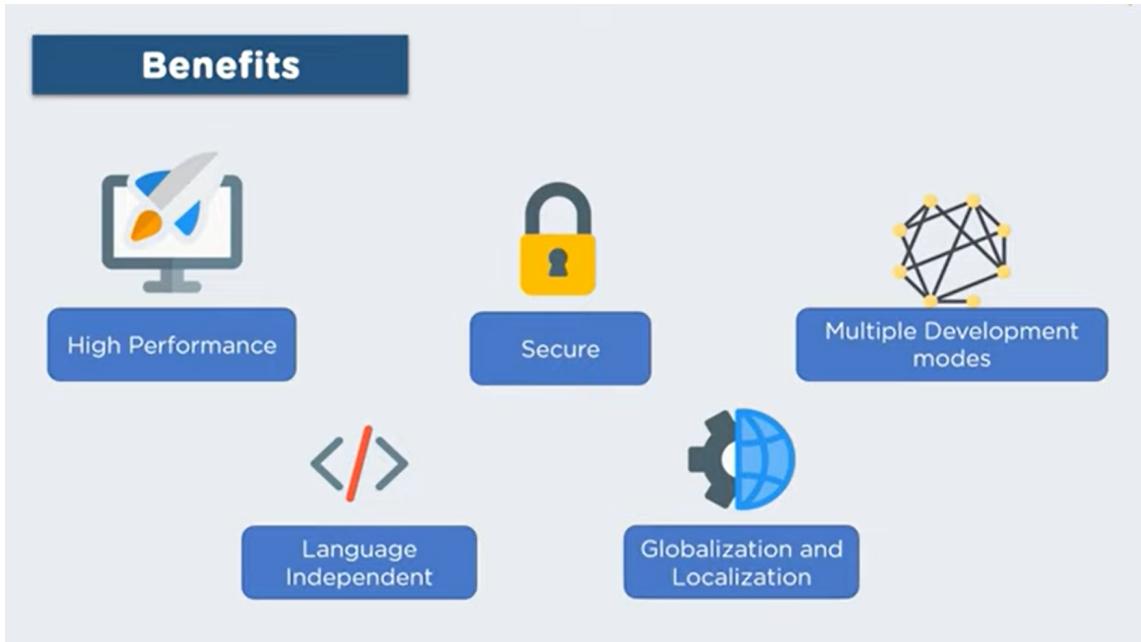
Web Essentials for Visual Studio



ReSharper

A tool that provides an absolute productivity boost in code quality analysis reduces time by identifying the errors.

10



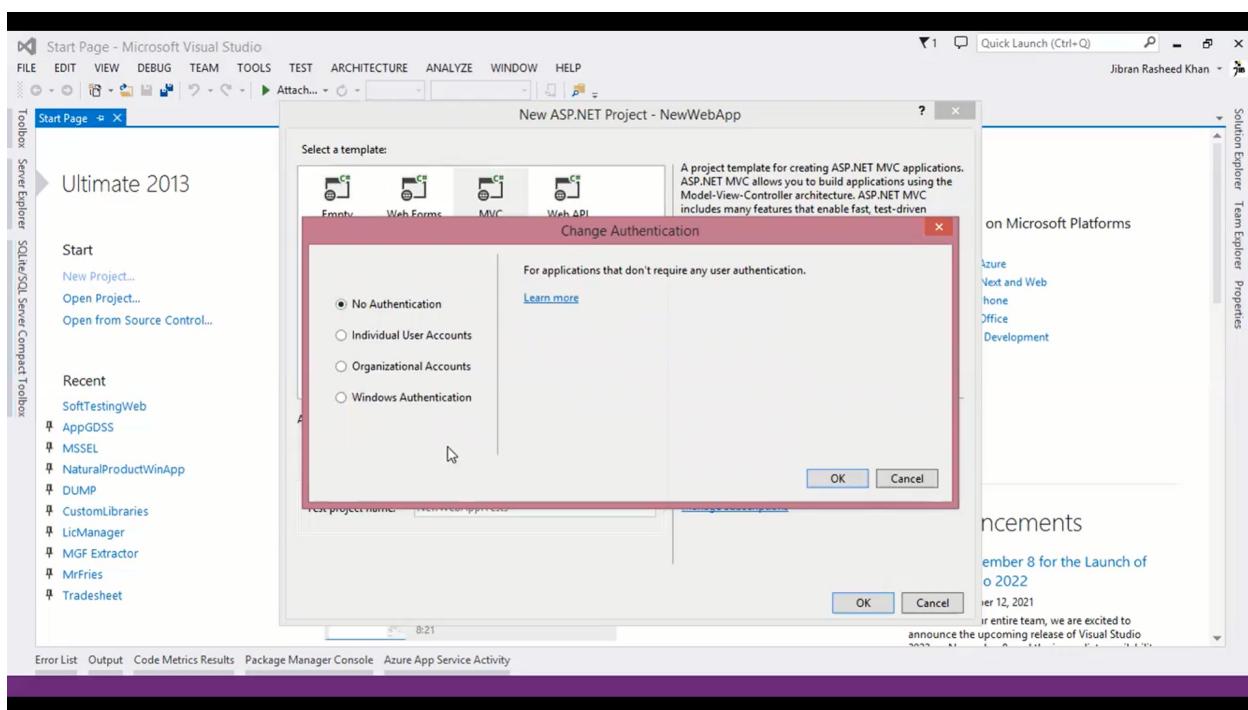
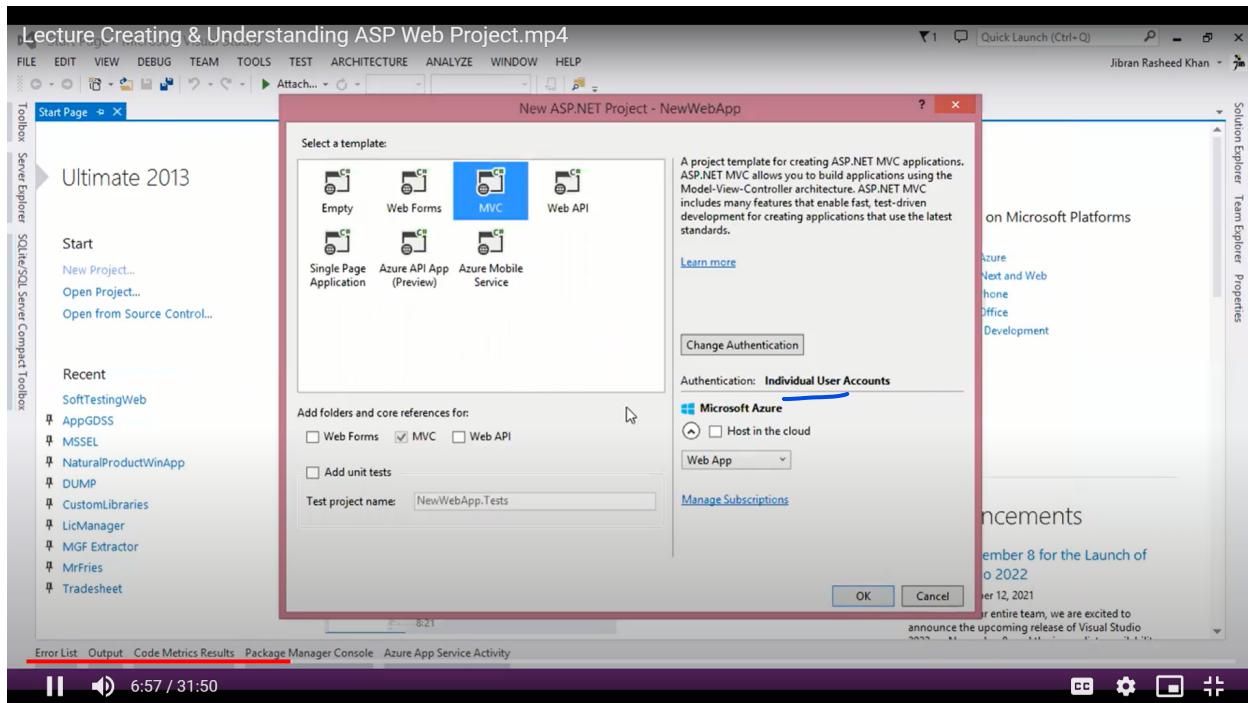
These are some things that ASP.NET adds to the .NET platform:

- **Authentication system** that includes libraries, a database, and template pages for **handling logins**, **including multi-factor authentication** and **external authentication with Google, Twitter, and more.**

## Pages, APIs, real-time, and microservices

ASP.NET allows you to build many types of web app., including web pages, REST APIs, microservices, and hubs that push real-time content to connected clients.

---



Hierarchical of project

```

1 // General Information about an assembly is controlled through the following
2 // set of attributes. Change these attribute values to modify the information
3 // associated with an assembly.
4 [assembly: AssemblyTitle("NewWebApp")]
5 [assembly: AssemblyDescription("")]
6 [assembly: AssemblyConfiguration("")]
7 [assembly: AssemblyCompany("")]
8 [assembly: AssemblyProduct("NewWebApp")]
9 [assembly: AssemblyCopyright("Copyright © 2021")]
10 [assembly: AssemblyTrademark("")]
11 [assembly: AssemblyCulture("")]
12
13 // Version information for an assembly consists of the following four values:
14 // Major Version
15 // Minor Version
16 // Build Number
17 // Revision
18
19 // You can specify all the values or you can default the Revision and Build Numbers
20 // by using the '*' as shown below:
21
22 // The following GUID is for the ID of the typelib if this project is exposed to COM
23 [assembly: Guid("3e1dc560-c0cf-472d-ac8c-91d574165c71")]
24
25 // Setting ComVisible to false makes the types in this assembly not visible
26 // to COM components. If you need to access a type in this assembly from
27 // COM, set the ComVisible attribute to true on that type.
28 [assembly: ComVisible(false)]
29
30 // Version information for an assembly consists of the following four values:
31 // Major Version
32 // Minor Version
33 // Build Number
34 // Revision
35
36 // You can specify all the values or you can default the Revision and Build Numbers
37 // by using the '*' as shown below:
38 [assembly: AssemblyVersion("1.0.0.0")]
39 [assembly: AssemblyFileVersion("1.0.0.0")]

```

```

1 // General Information about an assembly is controlled through the following
2 // set of attributes. Change these attribute values to modify the information
3 // associated with an assembly.
4 [assembly: AssemblyTitle("NewWebApp")]
5 [assembly: AssemblyDescription("")]
6 [assembly: AssemblyConfiguration("")]
7 [assembly: AssemblyCompany("")]
8 [assembly: AssemblyProduct("NewWebApp")]
9 [assembly: AssemblyCopyright("Copyright © 2021")]
10 [assembly: AssemblyTrademark("")]
11 [assembly: AssemblyCulture("")]
12
13 // Version information for an assembly consists of the following four values:
14 // Major Version
15 // Minor Version
16 // Build Number
17 // Revision
18
19 // You can specify all the values or you can default the Revision and Build Numbers
20 // by using the '*' as shown below:
21
22 // The following GUID is for the ID of the typelib if this project is exposed to COM
23 [assembly: Guid("3e1dc560-c0cf-472d-ac8c-91d574165c71")]
24
25 // Version information for an assembly consists of the following four values:
26 // Major Version
27 // Minor Version
28 // Build Number
29 // Revision
30
31 // You can specify all the values or you can default the Revision and Build Numbers
32 // by using the '*' as shown below:
33 [assembly: AssemblyVersion("1.0.0.0")]
34 [assembly: AssemblyFileVersion("1.0.0.0")]

```

The screenshot shows the Microsoft Visual Studio interface for a new web application named 'NewWebApp'. The main window displays the 'HomeController.cs' file under the 'NewWebApp' project. The code defines three actions: Index, About, and Contact, each returning a View result. The Solution Explorer on the right shows the project structure, including the Controllers folder which contains the HomeController.cs file.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6
7 namespace NewWebApp.Controllers
8 {
9     public class HomeController : Controller
10    {
11        public ActionResult Index()
12        {
13            return View();
14        }
15
16        public ActionResult About()
17        {
18            ViewBag.Message = "Your application description page.";
19
20            return View();
21        }
22
23        public ActionResult Contact()
24        {
25            ViewBag.Message = "Your contact page.";
26
27            return View();
28        }
29    }
30 }
```

This screenshot is identical to the one above, but it highlights the 'Scripts' folder in the Solution Explorer. The 'Scripts' folder is expanded, showing files such as \_reference.js, bootstrap.js, bootstrap.min.js, jquery-1.10.2.intellisense.js, jquery-1.10.2.js, jquery-1.10.2.min.js, jquery-1.10.2.min.map, jquery.validate-vsdoc.js, jquery.validate.js, jquery.validate.min.js, jquery.validate.unobtrusive.js, jquery.validate.unobtrusive.min.js, modernizr-2.6.2.js, respond.js, and respond.min.js.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6
7 namespace NewWebApp.Controllers
8 {
9     public class HomeController : Controller
10    {
11        public ActionResult Index()
12        {
13            return View();
14        }
15
16        public ActionResult About()
17        {
18            ViewBag.Message = "Your application description page.";
19
20            return View();
21        }
22
23        public ActionResult Contact()
24        {
25            ViewBag.Message = "Your contact page.";
26
27            return View();
28        }
29    }
30 }
```

Lecture Creating & Understanding ASP Web Project.mp4

The screenshot shows the Microsoft Visual Studio interface. The top menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, TEST, ARCHITECTURE, ANALYZE, WINDOW, and HELP. The toolbar below has icons for file operations like Open, Save, and Print, along with browser and build controls. The main window has tabs for 'Index' and 'HomeController.cs'. The code editor displays the following C# code for the HomeController:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6 
7 namespace NewWebApp.Controllers
8 {
9     public class HomeController : Controller
10    {
11        public ActionResult Index()
12        {
13            return View();
14        }
15 
16        public ActionResult About()
17        {
18            ViewBag.Message = "Your application description page.";
19 
20            return View();
21        }
22 
23        public ActionResult Contact()
24        {
25            ViewBag.Message = "Your contact page.";
26 
27            return View();
28        }
29    }
30 }
```

The Solution Explorer on the right lists the project structure for 'NewWebApp':

- Solution 'NewWebApp' (1 project)
  - NewWebApp
    - Properties
    - References
    - App\_Data
    - App\_Start
    - Content
    - Controllers
      - HomeController.cs
    - fonts
    - Models
    - Scripts
    - Views
      - Home
      - Shared
        - \_Layout.cshtml
        - Error.cshtml
        - \_ViewStart.cshtml
  - Global.asax
  - packages.config
  - Project\_Readme.html
  - Web.config

A blue bracket highlights the 'Views' folder under 'Controllers'.

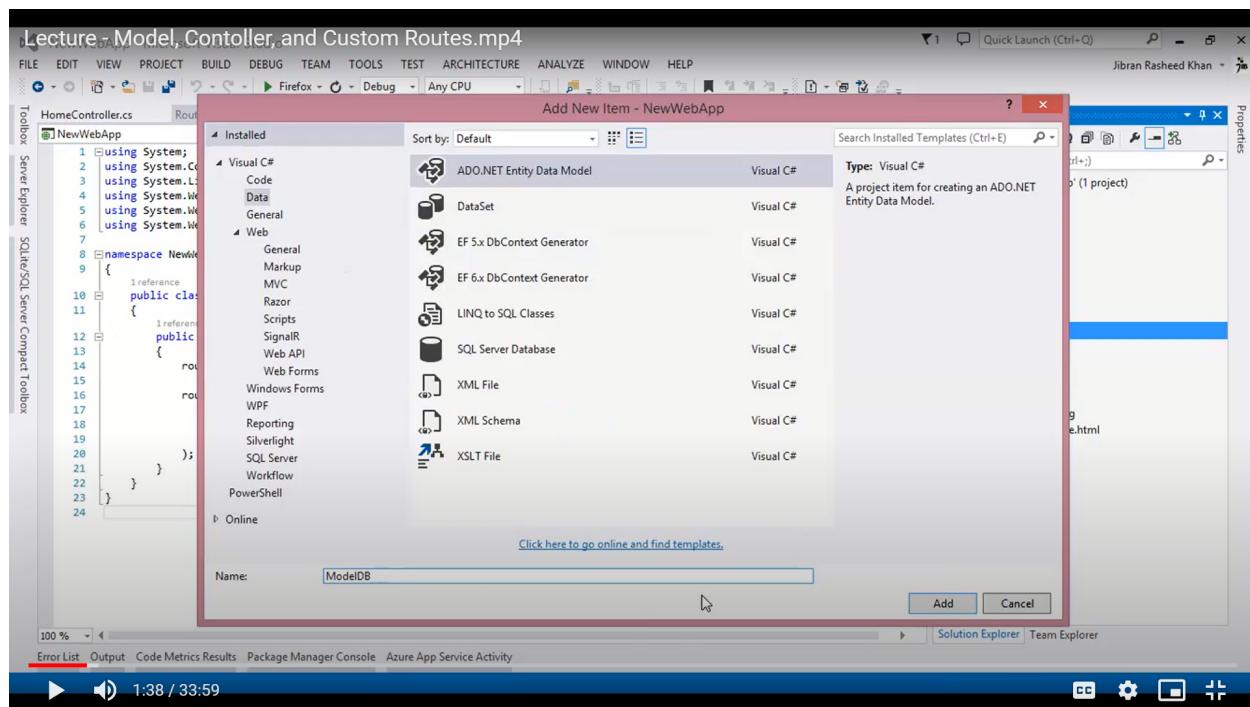
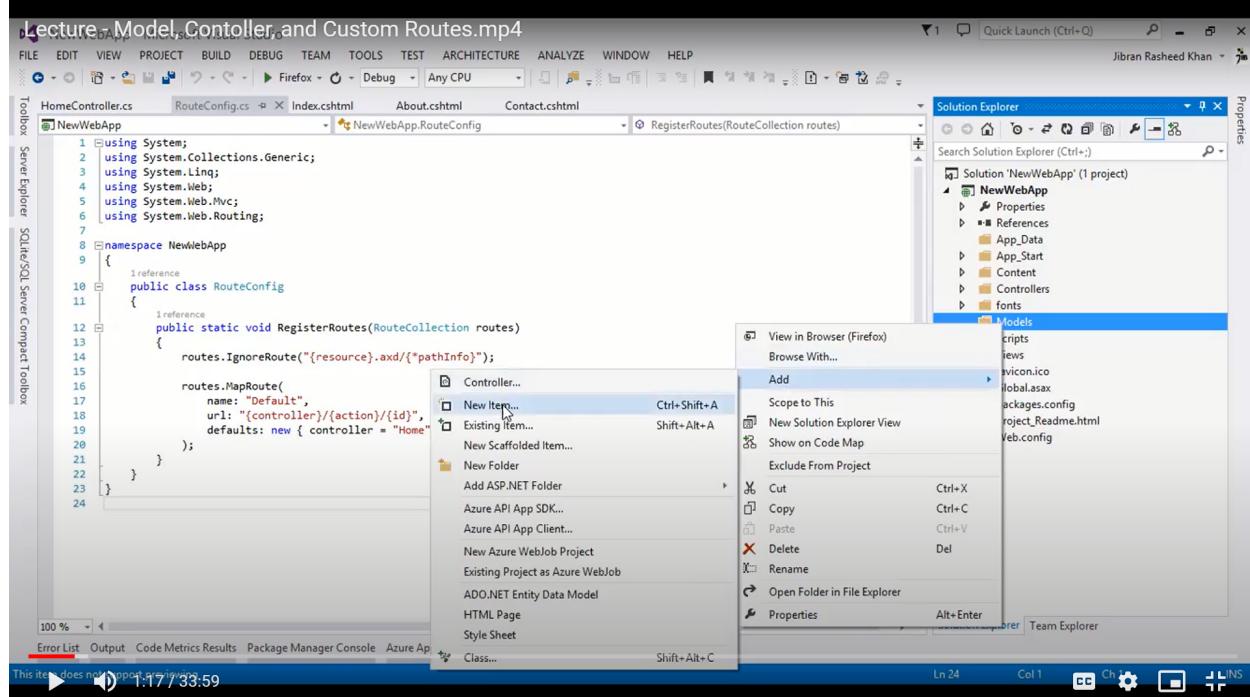
The screenshot shows the Microsoft Visual Studio interface, similar to the previous one but with a different view structure in the Solution Explorer. The code editor and menu bar are identical. The Solution Explorer now shows the following structure:

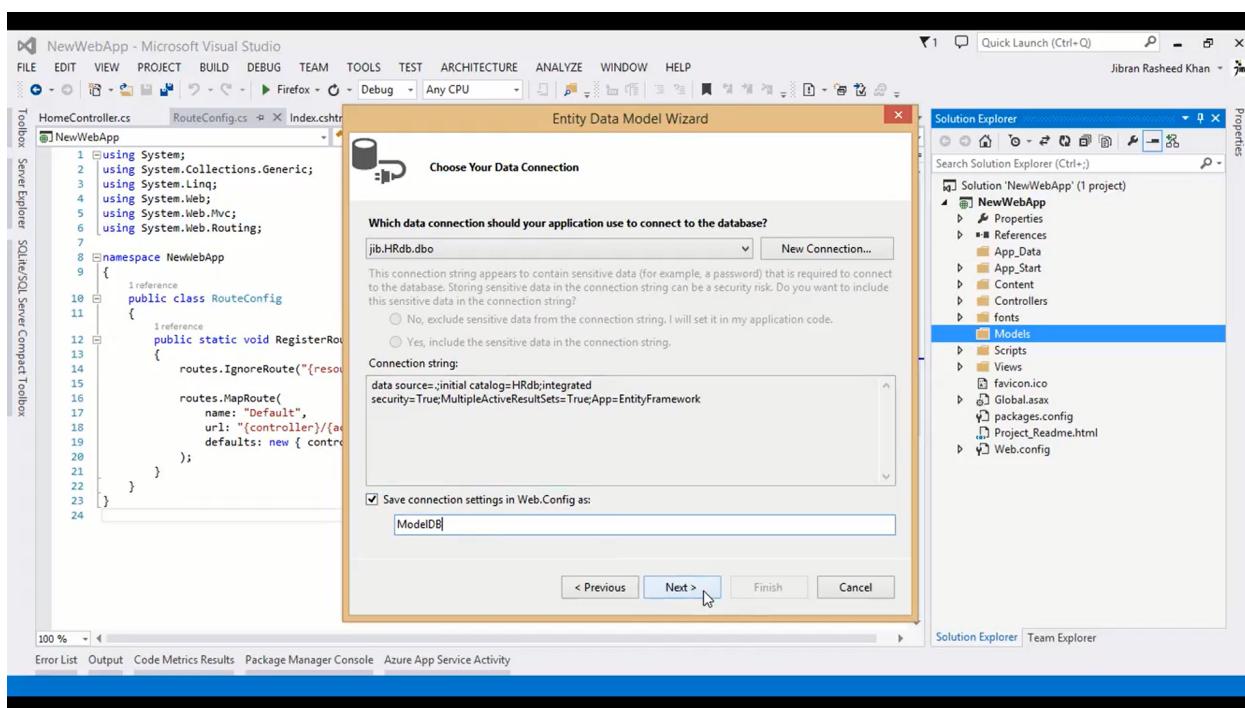
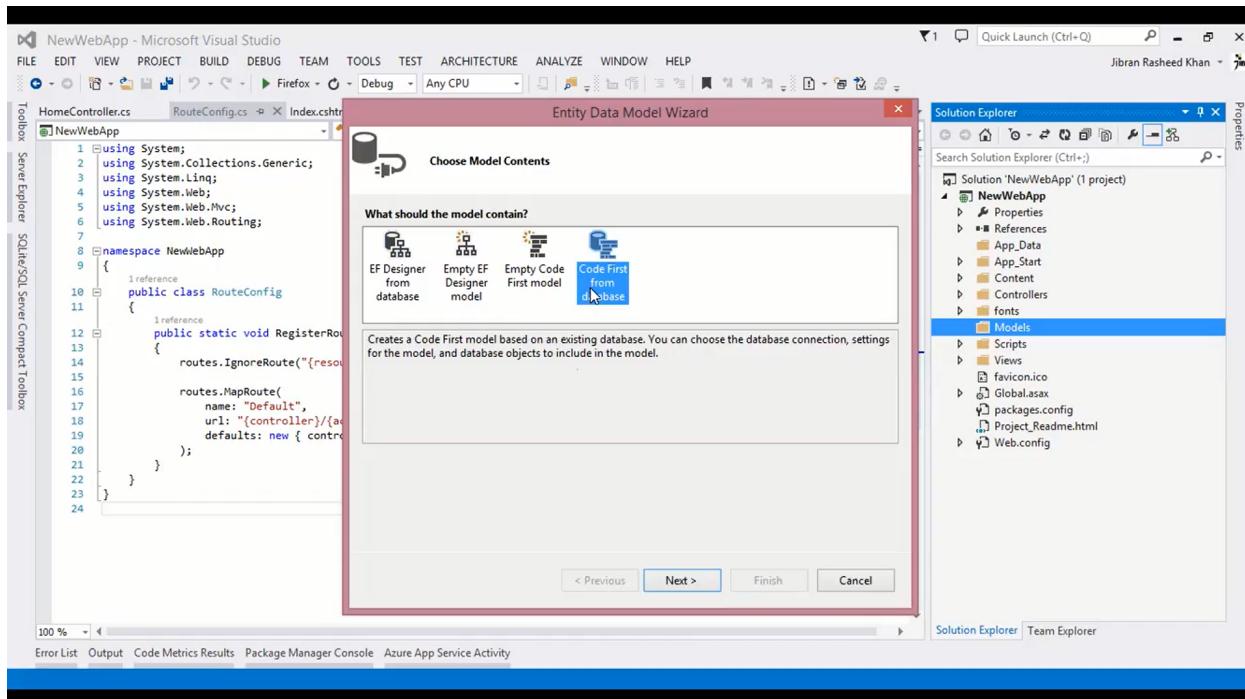
- Solution 'NewWebApp' (1 project)
  - NewWebApp
    - Properties
    - References
    - App\_Data
    - App\_Start
    - Content
    - Controllers
      - HomeController.cs
    - fonts
    - Models
    - Scripts
    - Views
      - Home
      - Shared
        - \_Layout.cshtml
        - Error.cshtml
        - \_ViewStart.cshtml
  - Global.asax
  - packages.config
  - Project\_Readme.html
  - Web.config

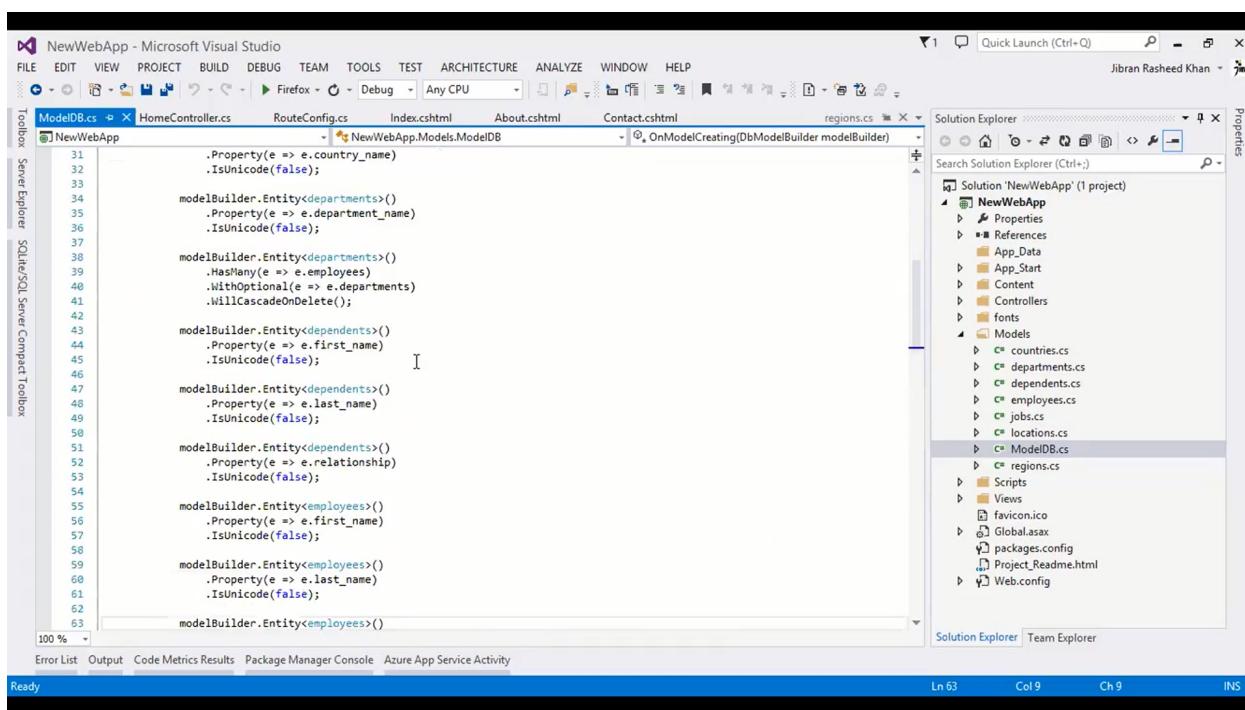
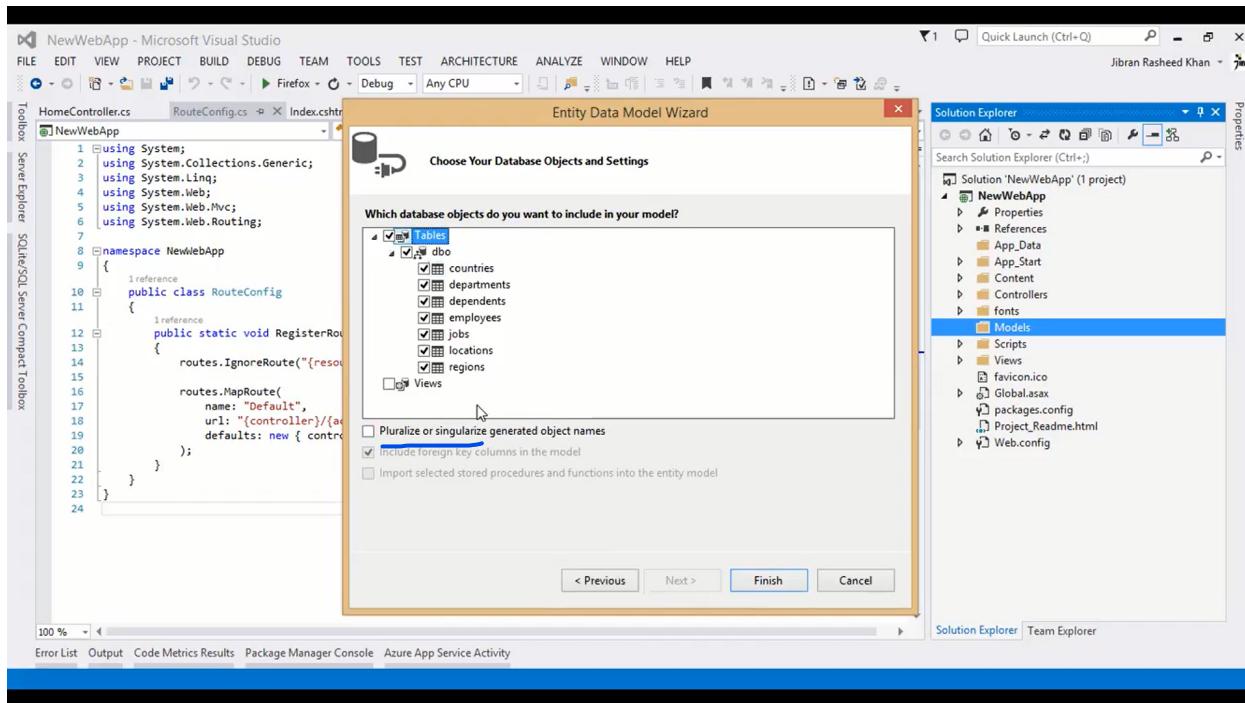
A blue bracket highlights the '\_Layout.cshtml' file under the 'Shared' folder in the 'Views' section.

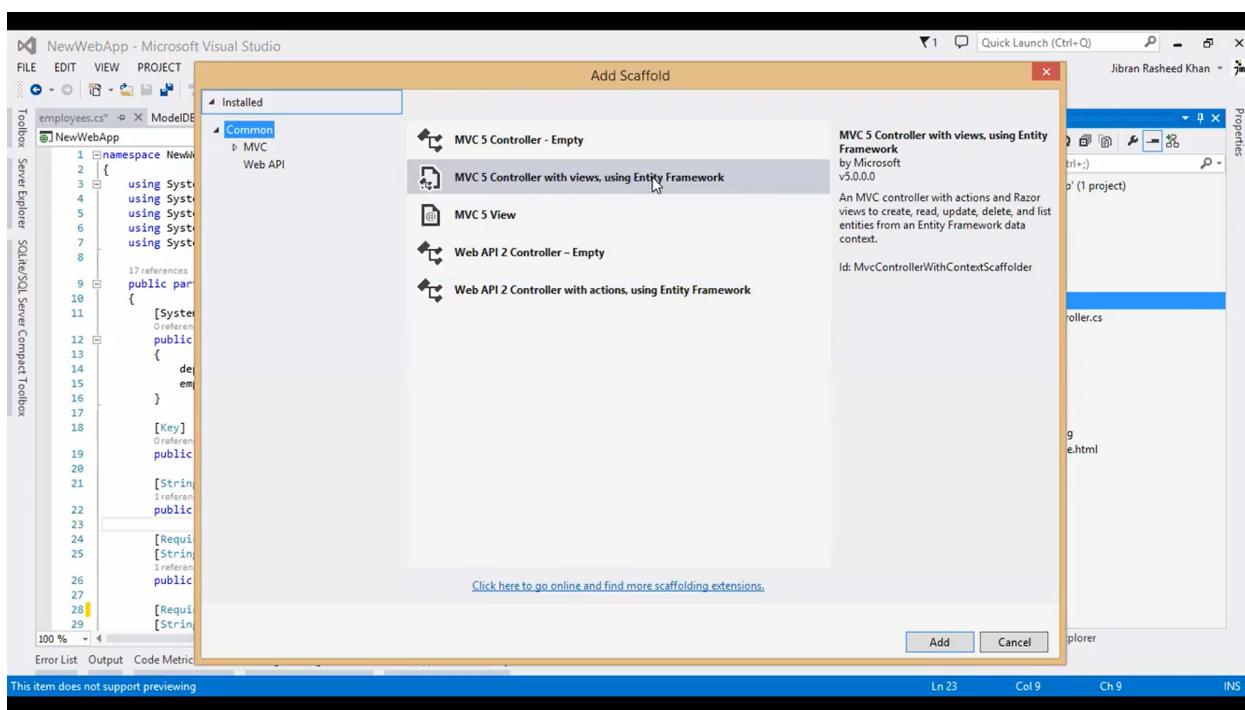
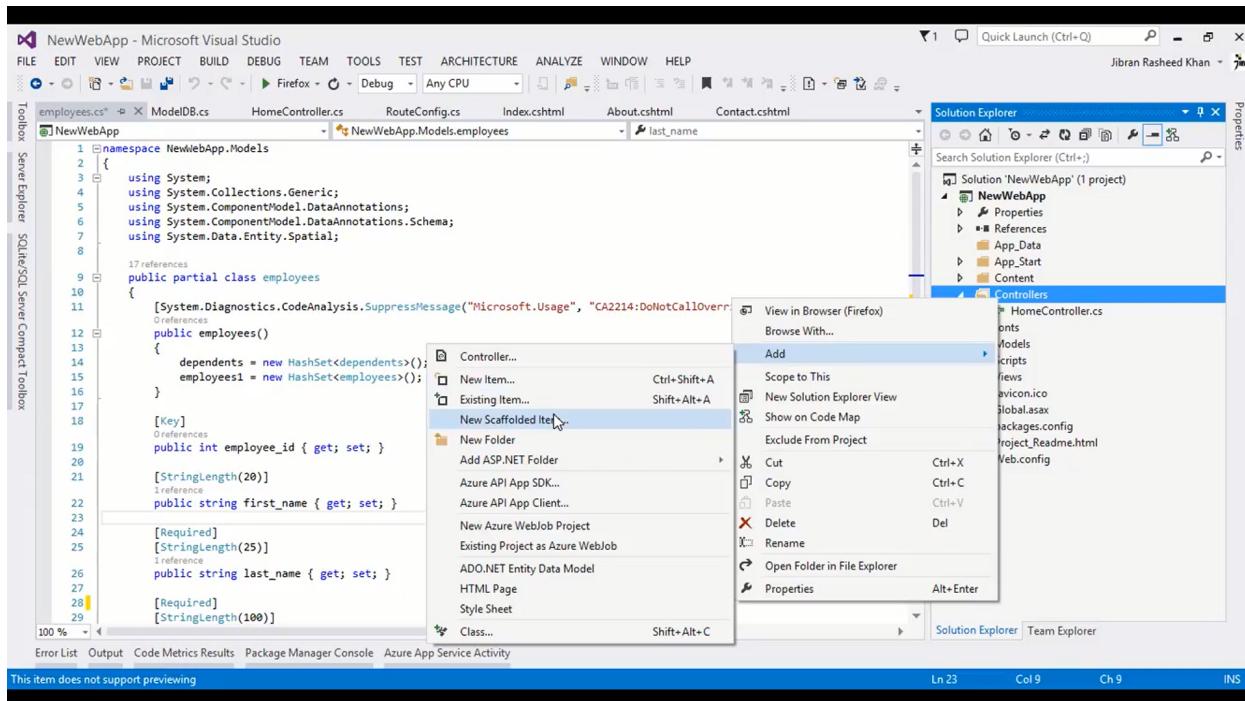
[required]--> compulsory  
?--> optional

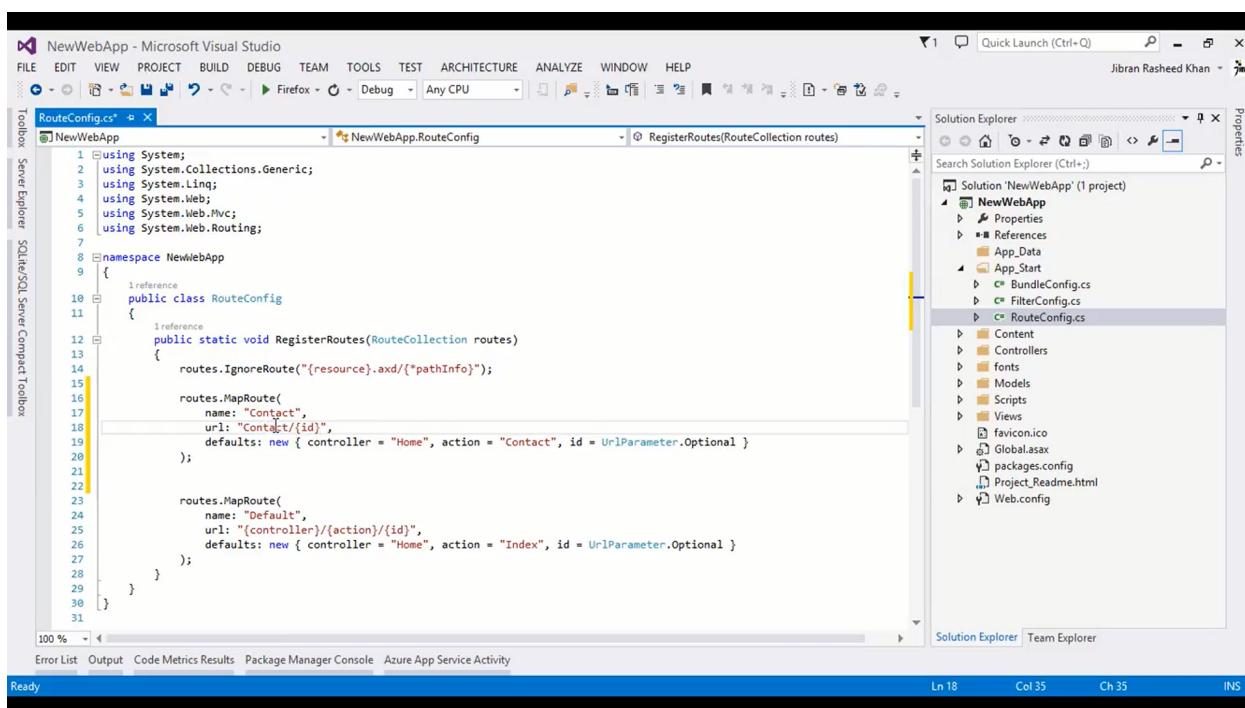
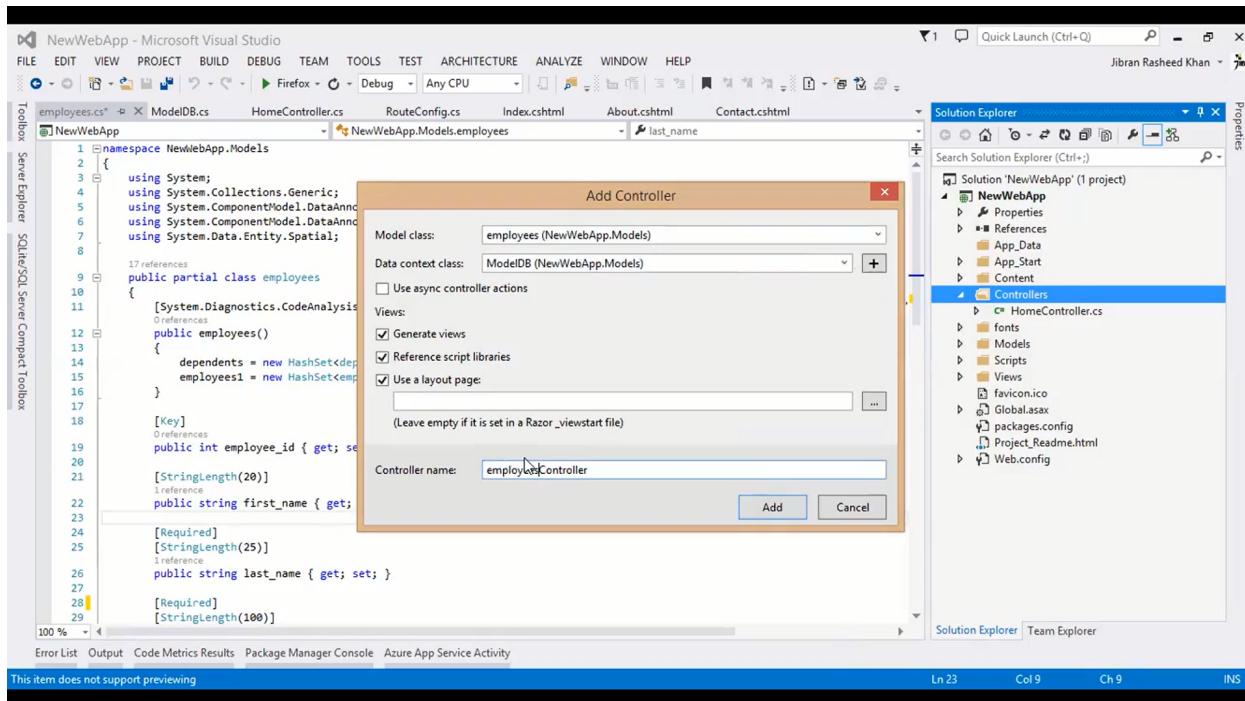
- Model class added (DB)











- customize the URL
-