Session: 1

Methods and Access Specifiers

Objectives

- Define and describe the layers of Web application
- Explain the structure of an ASP.NET MVC application
- Explain the evaluation of Web application
- Explain and describe how to create Web application in Visual Studio 2013

Overview of Web Application Development

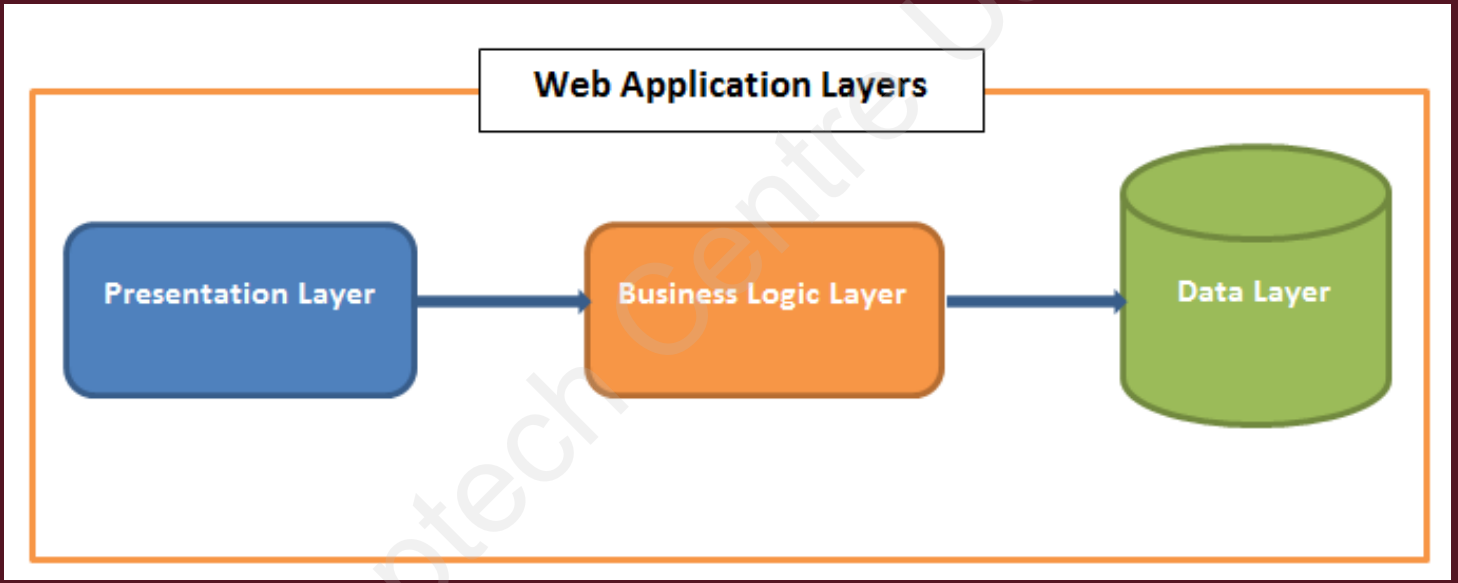
- **Web applications**:
  - Are programs that are executed on a Web server and accessed from a Web browser.
  - Allows you to share and access information over the Internet that can be accessed globally at any time.
  - Enables you to perform commercial transactions known as E-commerce application.

Web Application Layers 1-2

- Web applications are typically divided into three layers:
  - **Presentation layer**: Enable users to interact with the application.
  - **Business logic layer**: Enables to control the flow of execution and communication between the presentation layer and data layer.
  - **Data layer**: Enables to provide the application data stored in databases to the business logic layer.

Web Application Layers 2-2

- Following figure shows the three layers of a Web application:



Web Application Architectures

- Architecture of an application depends on the system in which the layers of the application are distributed and communicated to each other.
- An application can be based on one of the following types of architectures:
  - **Single-tier**: In this architecture, all the three layers are integrated together and installed on a single computer.
  - **Two-tier**: In this architecture, the three layers are distributed over two tiers, a client and a server.
  - **Three-tier**: In this architecture, the three layers of the application are distributed across different computers.
  - **N-tier**: In this architecture, the components of the three-tier are further separated.

Types of Web Pages

- A Web application consists of Web pages.
- A Web page can be categorized on the following two types:
  - **Static Web page**: Consists of only Hyper Text Markup Language (HTML) to present content to users.
  - **Dynamic Web page**: Consists of HTML in combination with server-side and client-side scripts to respond to user actions.

Evolution of ASP.NET MVC

- ASP.NET MVC is a framework for developing dynamic Web applications using the .NET Framework.
- Prior to ASP.NET MVC, dynamic Web applications based on the .NET Framework were developed using the following:
  - ASP.NET Web Forms
  - ASP.NET Web pages

ASP.NET Applications 1-3

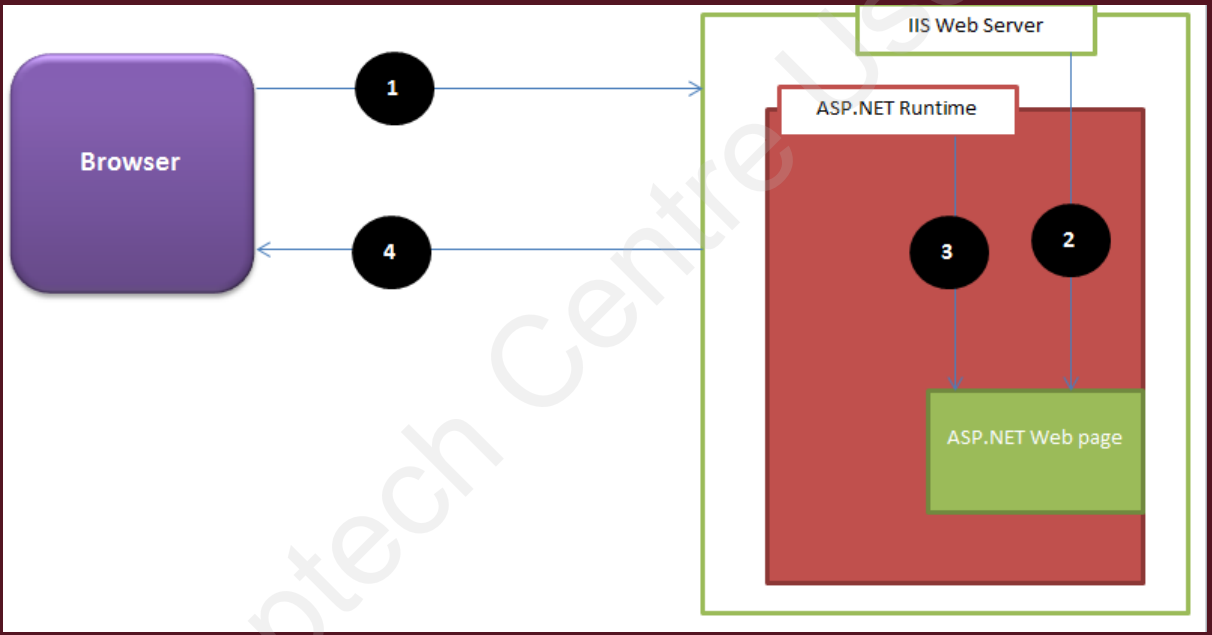
- ASP.NET:
  - Is a server-side technology that enables you to create dynamic Web applications using advanced features.
  - Are comprised of .aspx Web pages that combine both client-side and server-side scripts to build dynamic Websites.
  - Application can be deployed on a Web server such as Internet Information Services (IIS), which is the Web server for the Windows platform.

ASP.NET Applications 2-3

- The request-response process for an ASP.NET Web page contains the following steps:
  - Browser sends a request for an ASP.NET Web page.
  - When the request arrives, the IIS server intercepts the request, loads the requested file, and forwards it to the ASP.NET runtime for processing.
  - The ASP.NET runtime that contains the ASP.NET script engine processes the requested ASP.NET page and generates the response.
  - The IIS server sends back the response to the Web server that requested the page.

ASP.NET Applications 3-3

- Following figure shows the request-response flow for an ASP.NET Web page:



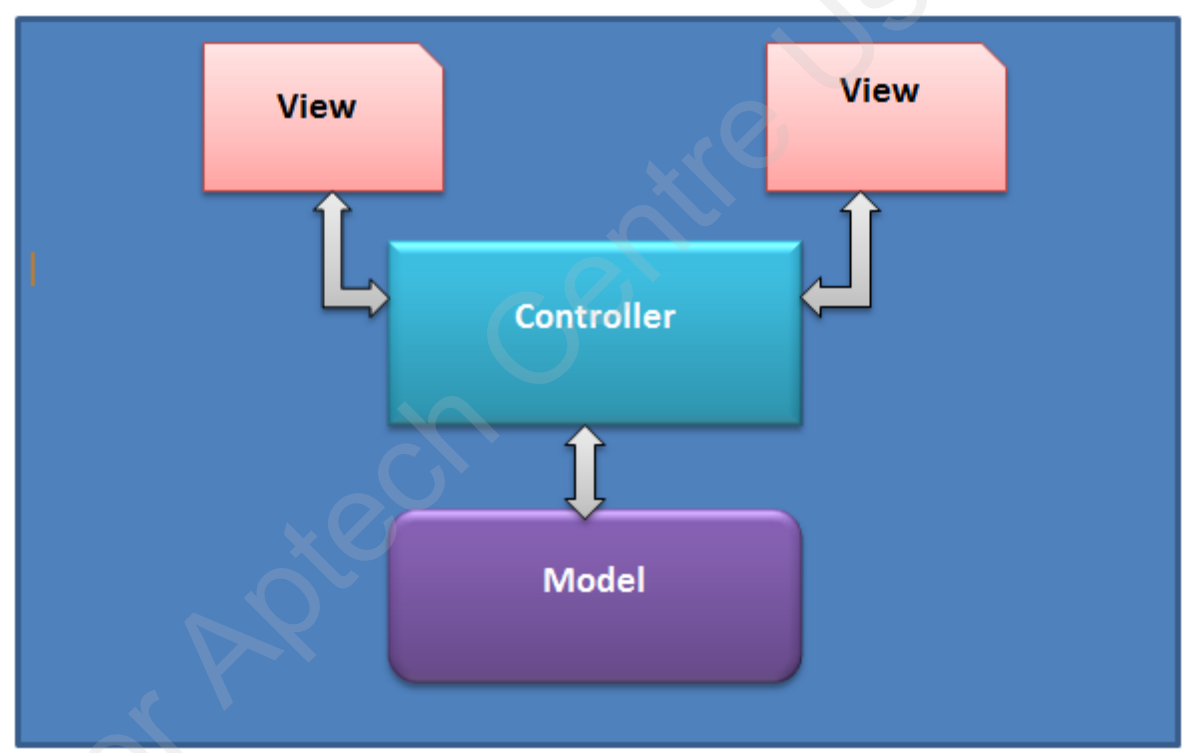
ASP.NET Web Forms

- The traditional ASP.NET Web applications gradually evolved to ASP.NET Web Forms to simplify development of dynamic Web applications.
- In ASP.NET Web Forms:
  - You can drag and drop User Interface (UI) controls to design your UI.
  - You can specify how the form and its control should respond at runtime.
- ASP.NET Web Forms:
  - Uses a combination of HTML, server controls, server code, and allows users to request through browsers.
  - Does not require you to have a hardcore developer background.
  - Allows you to use CSS, generate semantically correct markup, and handle the development environment created for HTML elements easily.

ASP.NET MVC 1-3

- ASP.NET MVC is based on the MVC design pattern that allows you to develop software solutions.
- MVC design pattern:
  - Allows you to develop Web application with loosely coupled components.
  - Enables separating data access, business, and presentation logic from each other.
- While using the MVC design pattern, a Web application can be divided into following types:
  - **Model**: Represents information about a domain that can be the application data of a Web application.
  - **View**: Represents the presentation logic to provide the data of the model.
  - **Controller**: Represents the logic responsible for coordination between the view and model classes.
  -

- Following figure shows the communications between the model, view, and controller components:



ASP.NET MVC 3-3

- As ASP.NET MVC is based on the MVC design pattern, it provides the following benefits:
  - **Separation of concerns**: Enables you to ensure that various application concerns into different and independent software components.
  - **Simplified testing and maintenance**: Enables you to test each component independently. Thus, it allows you to ensure that it is working as per the requirement of the application and then, simplifies the process of testing, maintenance, and troubleshooting procedures
  - **Extensibility**: Enables the model to include a set of independent components that you can easily modify or replace based on the application requirement.

History of MVC 1-6

- The versions that depict the history of MVC are as follows:

History of MVC 2-6

ASP.NET MVC 1

- First version of ASP.NET MVC. ASP.NET MVC 1 was released on March 13, 2009.
- Targets .NET Framework 3.5.
- Supports Visual Studio 2008 and Visual Studio 2008 SP1 to develop ASP.NET MVC 1 applications.

History of MVC 3-6

ASP.NET MVC 2

- Released on March 10, 2010 and targets .NET Framework 3.5 and 4.0.
- Supports Visual Studio 2008 and 2010 to develop ASP.NET MVC 2 applications.
- ASP.NET MVC 2 included the following key features:
  - Strongly typed HTML helpers means
  - Data Annotations Attribute
  - Client-side validation
  - Automatic scaffolding
  - Segregating an application into modules
  - Asynchronous controllers

History of MVC 4-6

ASP.NET MVC 3

- This version of ASP.NET MVC was released on Jan 13, 2011 and targets .NET Framework 4.0.
- Supports Visual Studio 2010 to develop ASP.NET MVC 3 applications.
- This version includes the following key features:
  - Supports Razor view engine
  - Improved Support for Data Annotations
  - Dependency Resolver
  - Supports Entity Framework Code First
  - Supports ViewBag to pass data from controller to view
  - Supports Action Filters that can be applied globally
  - Support for Unobtrusive JavaScript
  - Supports jQuery Validation

History of MVC 5-6

ASP.NET MVC 4

- This version of ASP.NET MVC was released on Aug 15, 2012 and targets .NET Framework 4.0 and 4.5.
- Supports Visual Studio 2010 SP1 and Visual Studio 2012 to develop ASP.NET MVC 4 applications.
- This version introduced the following key features:
  - ASP.NET Web API
  - Asynchronous Controllers with Task support
  - Bundling and minification
  - Support for the Windows Azure SDK

History of MVC 6-6

ASP.NET MVC 5

- This version of ASP.NET MVC was released on 17 October 2013 and targets .NET Framework 4.5 and 4.5.1.
- Supports Visual Studio 2013 to develop ASP.NET MVC 5 applications.
- This version introduced the following new features:
  - ASP.NET Identity
  - ASP.NET Web API 2

Architecture of ASP.NET MVC Application 1-3

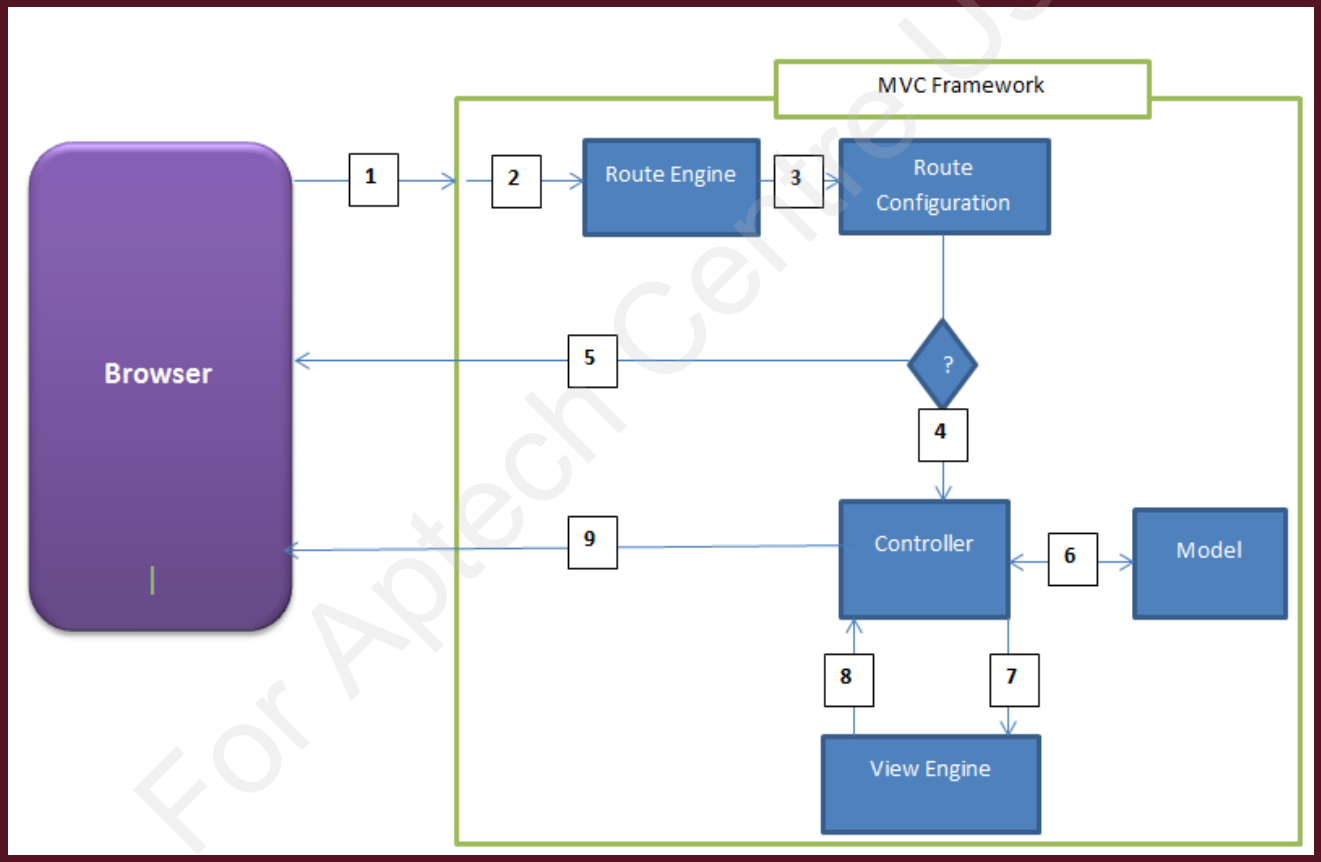
- The basic architecture of an ASP.NET MVC application involves the following components:
  - MVC Framework
  - Route engine
  - Route configuration
  - Controller
  - Model
  - View engine
  - View
- Each of these preceding components communicates to process requests coming to an ASP.NET MVC application.
- The process of handling an incoming request involves a series of steps that these components perform.

Architecture of ASP.NET MVC Application 2-3

- The steps that the components of the ASP.NET MVC Framework performs while handling an incoming request includes:
  - The browser sends a request to an ASP.NET MVC application.
  - The MVC Framework forwards the request to the routing engine.
  - The route engine checks the route configuration of the application for an appropriate controller to handle the request.
  - When a controller is found it is invoked.
  - When a controller is not found the route engine indicates that the controller has not been found and the MVC Framework communicates this as an error to the browser.
  - The controller communicates with the model.
  - The controller requests a view engine for a view based on the data of the model.
  - The view engine returns the result to the controller.
  - The controller sends back the result as an HTTP response to the browser.

Architecture of ASP.NET MVC Application 3-3

- Following figure shows the steps that the components of the ASP.NET MVC Framework performs while handling an incoming request:



Supporting Technologies

- ASP.NET MVC application supports various technologies to create dynamic and responsive Web application.
- Some of the supporting technologies that you can use while creating an ASP.NET MVC application are as follows:

JavaScript

JQuery

Asynchronous JavaScript and XML (AJAX)

IIS

Windows Azure

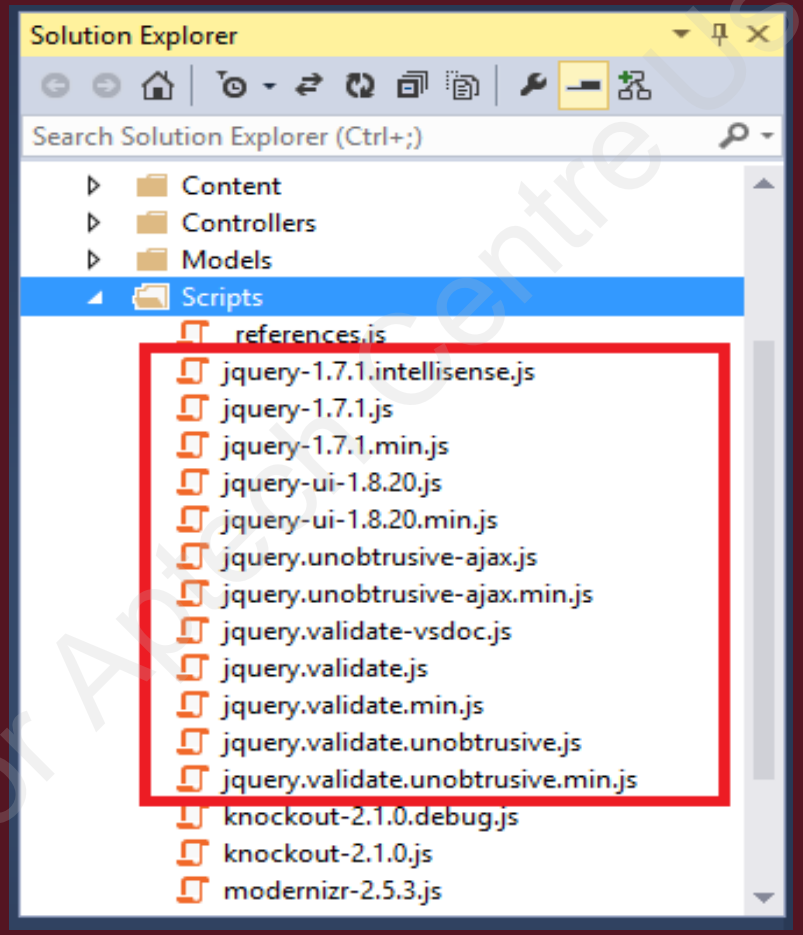
JavaScript

- Is a client-side scripting language that enables a Web application to respond to user requests without interacting with a Web server.
- Allows creating responsive Web applications that enhances user experience.
- Allows implementing functionalities, such as an easy to use UI, quick response to the user's request, and run in all available browsers.
- Allows your Web applications to respond to user requests without interacting with a Web server.

JQuery 1-3

- JQuery:
  - Is a JavaScript library that simplifies the client-side scripting of HTML.
  - Is used in views to make your views responsive and dynamic.
  - Allows you to perform client-side validation of forms.
  - Provides several plugins that enable you to enhance the UI elements of a view.
- When you create an ASP.NET MVC project in Visual Studio 2013, the project automatically includes the jQuery libraries within the **Scripts** folder.

- Following figure shows the jQuery libraries in the **Scripts** folder of an ASP.NET MVC project:



- To use a jQuery library in a view, you need to refer the library from the view.
- You can use the following code snippet to refer to a jQuery library from a view:

Code Snippet:

```
<script src="/Scripts/jquery-1.7.1.min.js"
type="text/javascript"></script>
<!- -JQuery Code - ->
</script>
```

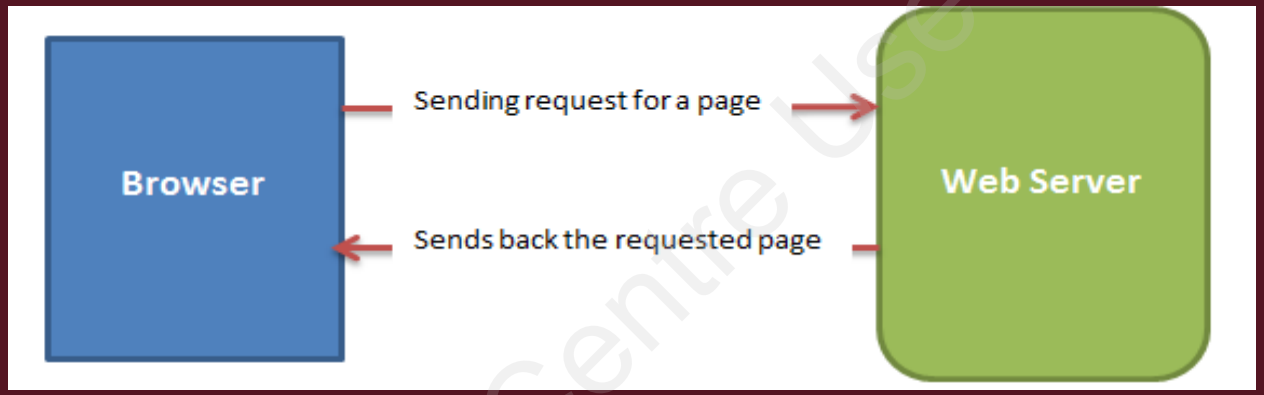
AJAX

- **AJAX**:
  - Is a Web development technique that is used to create interactive applications.
  - Allows you to asynchronously retrieve data in the background of an ASP.NET MVC application without interfering with the display and behavior of the existing view.
  - Enables the users to work continuously on the application without being affected by the responses received from the server.
- JavaScript forms an integral part of the AJAX-enabled Web applications because these applications process most of the requests on the client side, unless there is a need to connect to the Web server.
- The XMLHttpRequest object is one of the primary objects used by JavaScript because it enables asynchronous communication between the client and the server.

IIS 1-5

- An ASP.NET MVC application requires a Web server that enables handling HTTP requests and creates responses.
- When you request for a Web page on a browser, you type a URL on a browser, for example, `http://mvcexample.com/index.html`.
- This URL consists of the following parts:
  - **http**: A protocol to use for exchanging request and response.
  - **mvctest.com**: A domain name that maps to an unique IP address.
  - **index.html**: A file that you are requesting.

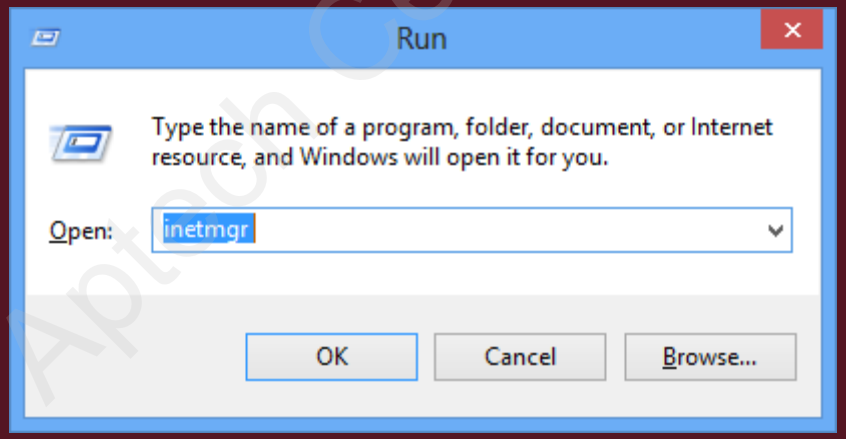
- Following figure shows the communication between a browser and the server:



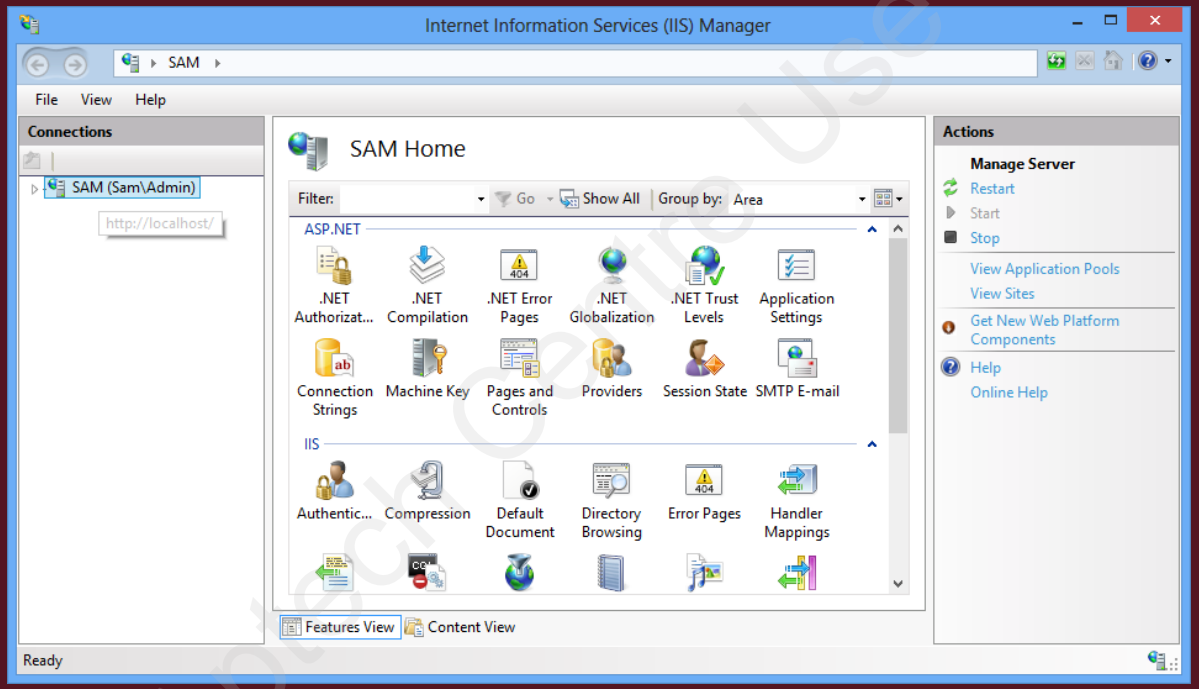
- In this figure:
  - The browser creates a connection with a Web server where the IP address is registered based on the IP address.
  - Next, the browser uses the HTTP protocol to send a request to the server, asking for the file.
  - The server, searches for the file and sends back the content of the file as HTML markup to the browser.
  - The browser parses the HTML markup and displays the output to you.

- IIS:
  - Is a product of Microsoft and is designed to deliver information in high speed and securely.
  - Acts as a platform that you can target to extend the capabilities of the Internet standards.
  - Has been designed to be modular that allows selecting only those modules that meet your specific requirements while installing IIS.
  - Some of the modules that you need to host ASP.NET MVC applications are:
    - **Content modules**: Enables controlling how content is delivered to client.
    - **Security modules**: Enables implementing security through various authentication mechanisms, authorization based on URLs, and filtering requests.
    - **Compression module**: Enables compressing responses sent to client browser using compression standard, such as Gzip.
    - **Caching modules**: Enables caching content and delivering the cached content in subsequent requests for the same resource.
    - **Logging and Diagnostics modules**: Enables logging request processing information and response status for diagnostic purposes.

- You can deploy and manage ASP.NET MVC applications in IIS using IIS Manager.
- To access IIS Manager, you need to perform the following steps:
  1. Ensure that IIS is installed in your computer.
  2. Press the **Windows+R** key combination. The **Run** dialog box appears.
  3. Type `inetmgr`, as shown in the following figure:



4. Click **OK**. The **IIS Manager** window opens, as shown in the following figure:



- You can use the IIS Manager window to configure the features of IIS, such as granting permissions to hosted applications, managing server security, and managing authentication and authorization of applications.

Windows Azure 1-4

- Prior to the cloud platform, applications were deployed on Internet-accessible servers supported by datacenters.
- So, individuals and organizations find it difficult to set up such infrastructures.
- In addition, as applications scales, the problems to maintain such infrastructures become more obvious.
- As a solution, the cloud platform got introduced where individuals and organizations have the option to host and maintain applications in an infrastructure that they do not have to manage.

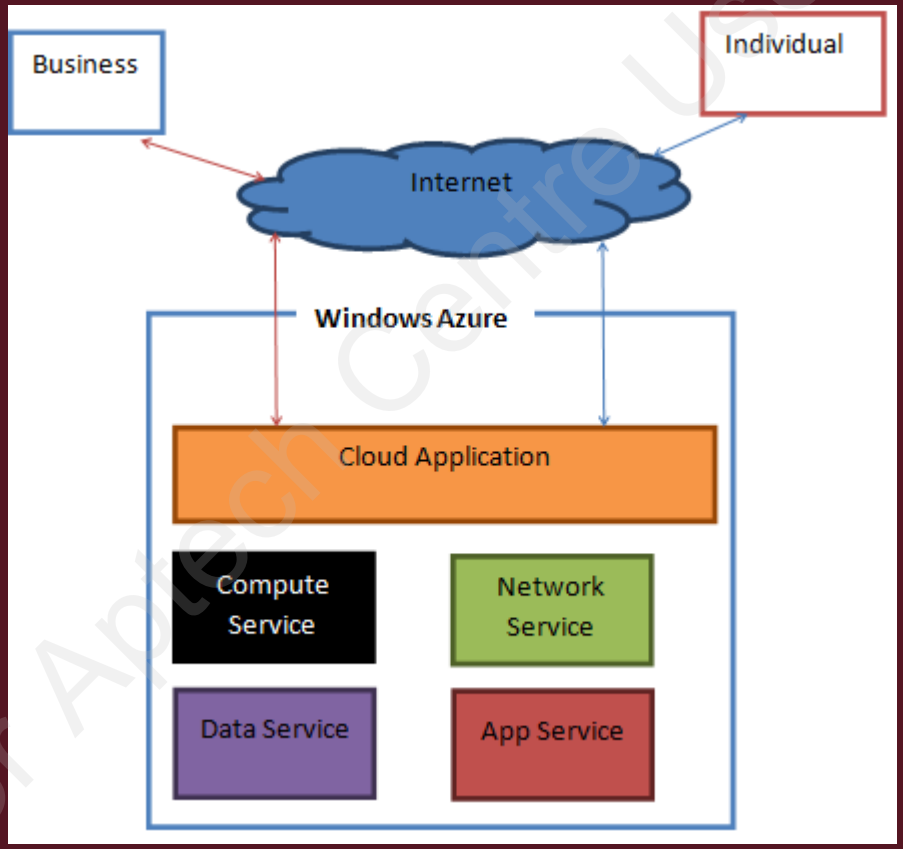
Windows Azure 2-4

- **Windows Azure**:
  - Is a cloud solution provided by Microsoft.
  - Allows you to create Word documents, share them, save them without requiring any software to be installed on your computer, or upgrade the current hardware configurations of your computer.
  - Provides a platform to build applications that can leverage the cloud to meet user needs that can range from a simple task, such as sending an e-mail to managing a complex ASP.NET MVC application that implements an online auction store with global user base.
- Instead of downloading, installing, and using a product on your own computers, you can use Windows Azure as a service to perform the same functions.

Windows Azure 3-4

- To enable cloud computing Windows Azure provides the following key cloud-based services:
  - **Compute services**: Provides the infrastructure to deliver processing power required to run cloud applications through services, such as virtual machines, Web sites, and mobile services.
  - **Network services**: Provides the services to deliver cloud applications to users and data centers.
  - **Data services**: Provides the services to enable cloud applications to efficiently store, manage, and secure application data.
  - **App services**: Provides the services to enable cloud applications to enhance their performance and security.

- Following figure shows the key cloud-based services of Windows Azure:

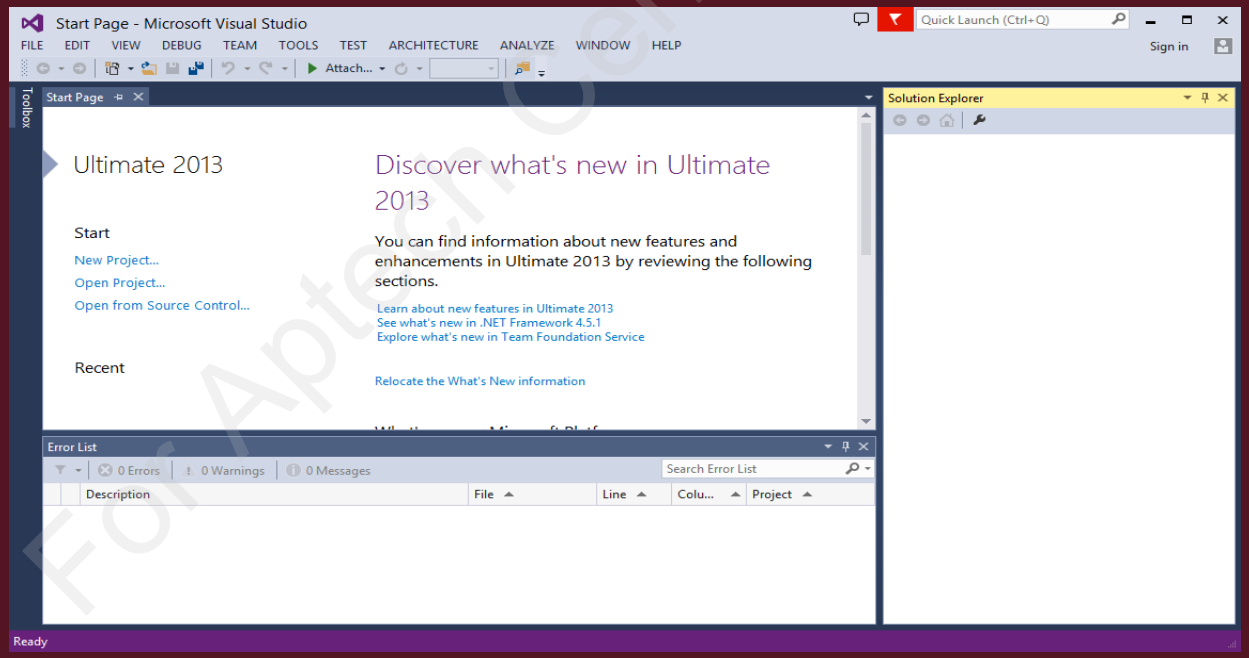


MVC Support in Visual Studio 2013

- Visual Studio 2013:
  - Simplifies the process of creating ASP.NET MVC applications by providing various in-built templates.
  - Provides an MVC template that automatically creates an MVC application structure with the basic files to run the application.

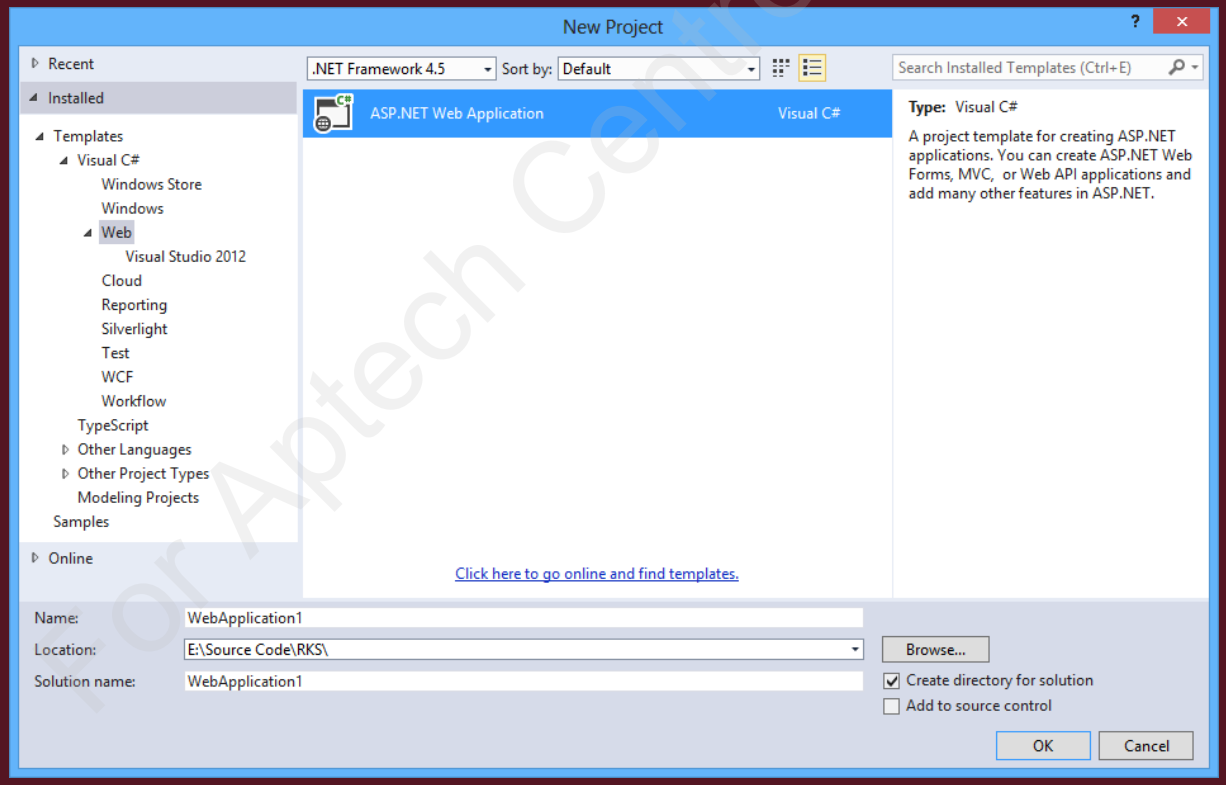
Creating an ASP.NET MVC Project 1-5

- To create an ASP.NET MVC application using Visual Studio 2013, you need to perform the following tasks:
  1. Press the **Windows+Q** key on keyboard.
  2. In the **Search** box that appears, start typing Visual Studio 2013. The **Visual Studio 2013** icon appears under the Apps section.
  3. Double-click the **Visual Studio 2013** icon. The **Start Page** of Visual Studio 2013 appears, as shown in the following figure:



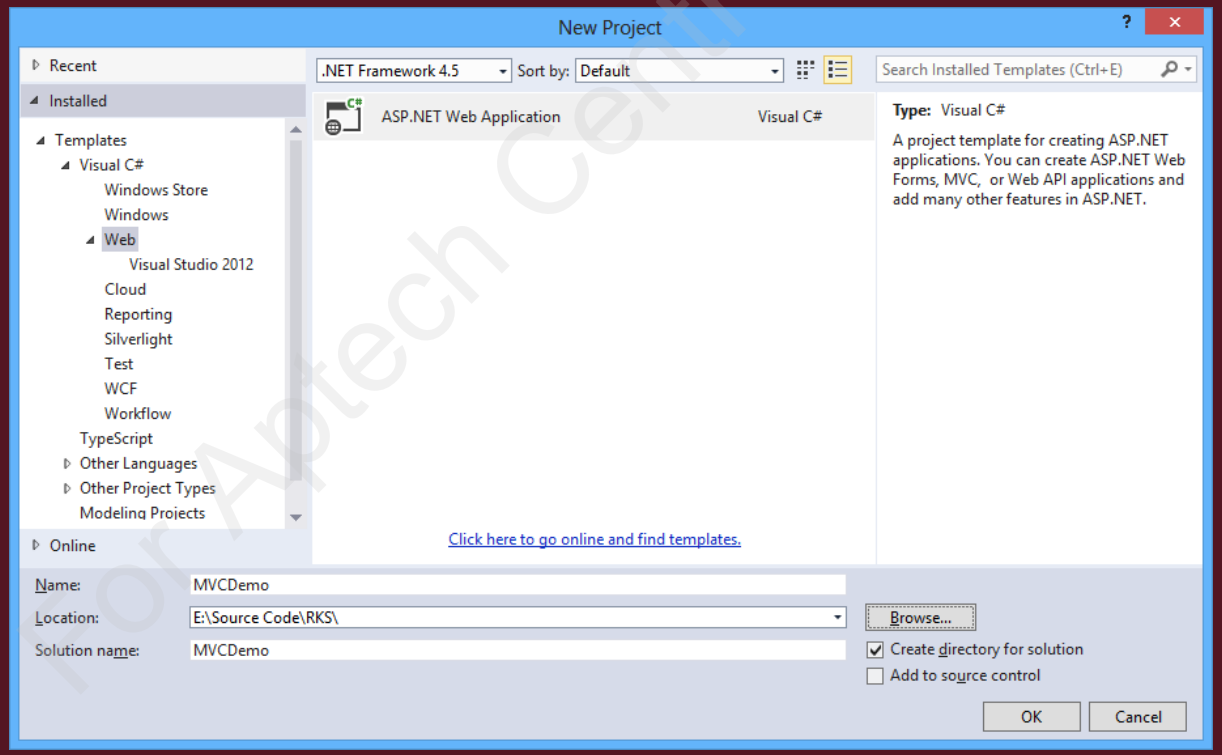
Creating an ASP.NET MVC Project 2-5

4. Click **File→New→Project** menu options in the menu bar of Visual Studio 2013.
5. In the **New Project** dialog box that appears, select **Web** under the **Installed** section and then, select the **ASP.NET Web Application** template, as shown in the following figure:



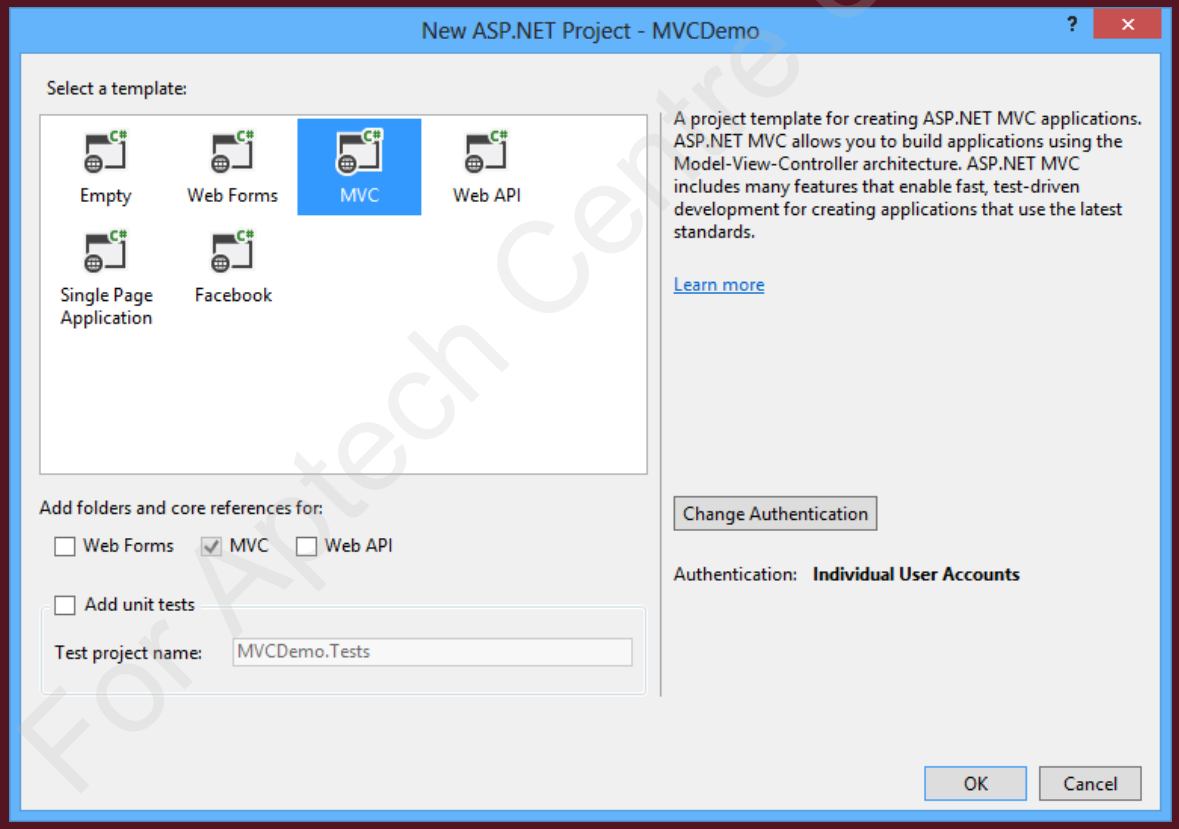
Creating an ASP.NET MVC Project 3-5

6. Type MVCDemo in the **Name** text field.
7. Click **Browse**. The **Project Location** dialog box is displayed. Specify the location where the application has to be created.
8. Click **Select Folder**. The **New Project** dialog box displays the specified location in the Location field, as shown in the following figure:



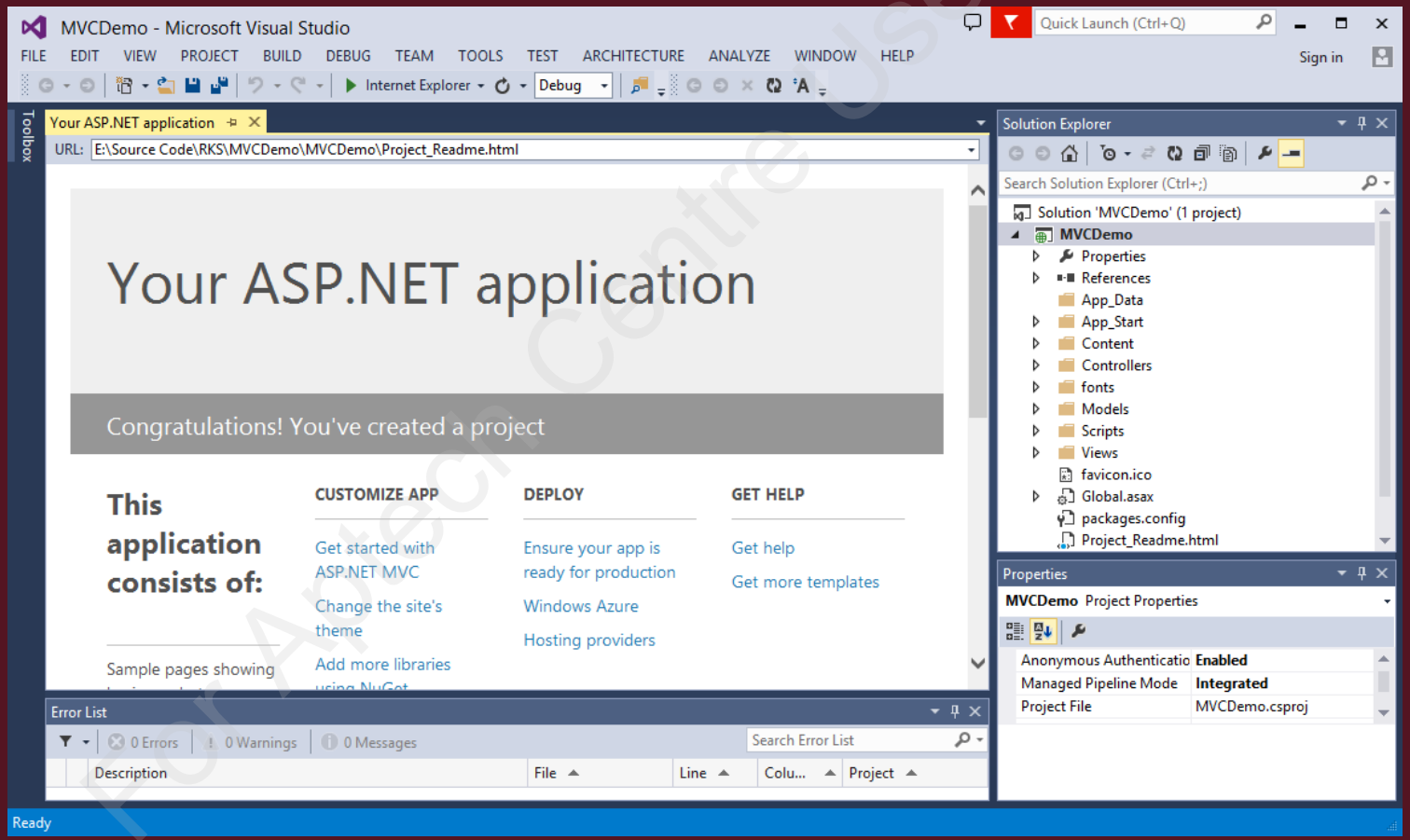
Creating an ASP.NET MVC Project 4-5

9. Click **OK**. The **New ASP.NET Project – MVCDemo** dialog box is displayed.
10. Select **MVC** under the Select a template section of the **New ASP.NET Project – MVCDemo** dialog box , as shown in the following figure:



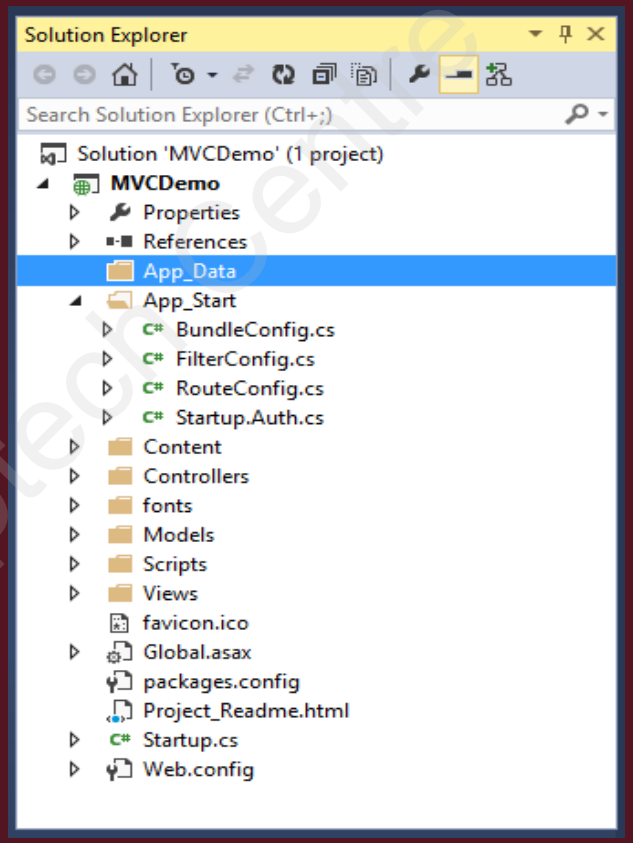
Creating an ASP.NET MVC Project 5-5

11. Click **OK**. Visual Studio 2013 displays the newly created application, as shown in the following figure:



Structure of an ASP.NET MVC Project 1-2

- When you create an ASP.NET MVC Web application in Visual Studio 2013, it automatically adds several files and folders to the project.
- Following figure shows the files and folders that Visual Studio 2013 creates when you create an ASP.NET MVC application:

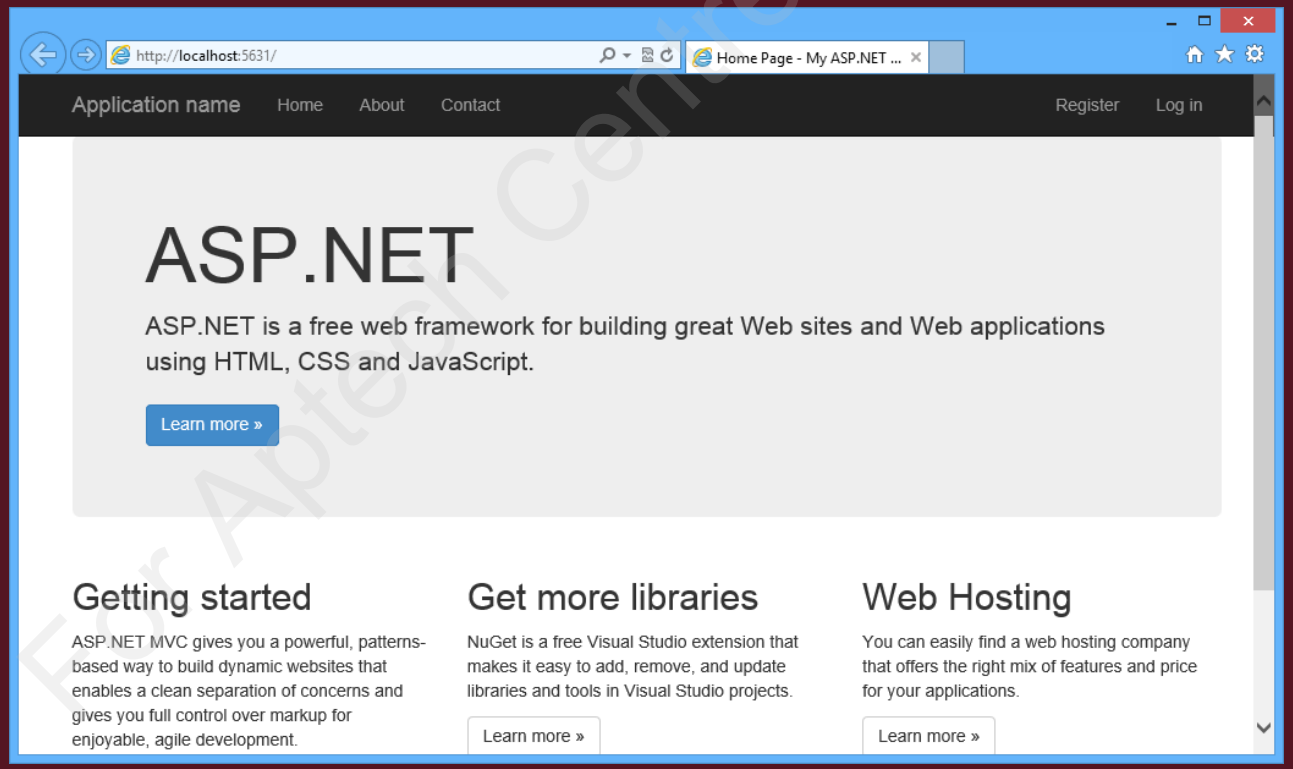


Structure of an ASP.NET MVC Project 2-2

- The top-level directory structure of an ASP.NET MVC application contains the following folders:
  - **Controllers**: Contains the Controller classes that handle URL requests.
  - **Models**: Contains the classes that represent and manipulate data and business objects.
  - **Views**: Contains the UI template files that are responsible for rendering output, such as HTML.
  - **Scripts**: Contains the JavaScript library files.
  - **Images**: Contains the images that you need to use in your application.
  - **Content**: Contains the CSS and other site content, other than scripts and images.
  - **Filters**: Contains the filter code.
  - **App_Data**: Contains data files that you need to read/write.
  - **App_Start**: Contains the files containing configuration code that you can use features like Routing, Bundling, and Web API.

Executing ASP.NET MVC Project

- To execute an application created in Visual Studio 2013 you need to click **Debug→Start** without debugging from the menu bar. The browser will display the output of the default application, as shown in the following figure:



Summary

- Web applications are programs that are executed on a Web server and accessed from a Web browser.
- Web applications are typically divided into three layers, where each layer performs different functionalities.
- The architecture of an application depends on the system in which the layers of the application are distributed and communicated to each other.
- ASP.NET is a server-side technology that enables you to create dynamic Web applications using advanced features, such as simplicity, security, and scalability, which are based on the .NET Framework.
- ASP.NET MVC is based on the MVC design pattern that allows you to develop software solutions.
- An ASP.NET MVC application requires a Web server that enables handling HTTP requests and creates responses.
- Windows Azure is a cloud solution provided by Microsoft.