MVC stands for Model, View and Controller. MVC separates application into three components - Model, View and Controller.

**Model**: Model represents shape of the data and business logic. It maintains the data of the application. Model objects retrieve and store model state in a database.

Model is a data and business logic.

**View**: View is a user interface. View display data using model to the user and also enables them to modify the data.

View is a User Interface.

**Controller**: Controller handles the user request. Typically, user interact with View, which in-tern raises appropriate URL request, this request will be handled by a controller. The controller renders the appropriate view with the model data as a response.

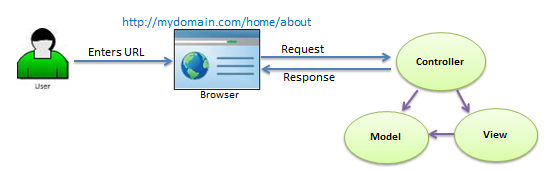
Controller is a request handler.

The following figure illustrates the interaction between Model, View and Controller.

[](http://www.tutorialsteacher.com/Content/images/mvc/mvc-architecture.png)

MVC Architecture

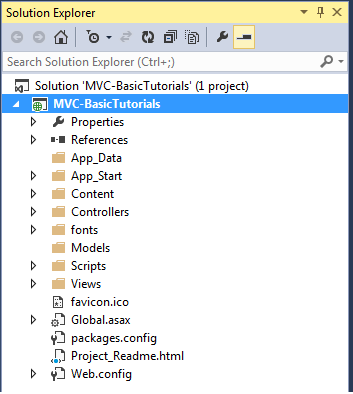
The following figure illustrates the flow of the user's request in ASP.NET MVC.

[](http://www.tutorialsteacher.com/Content/images/mvc/request-handling-in-mvc.png)

Request/Response in MVC Architecture

ASP.NET MVC Folder Structure:

We have created our first MVC 5 application in the previous section. Visual Studio creates the following folder structure for MVC application by default.

[](http://www.tutorialsteacher.com/Content/images/mvc/mvc-folder-structure.png)MVC Folder Structure

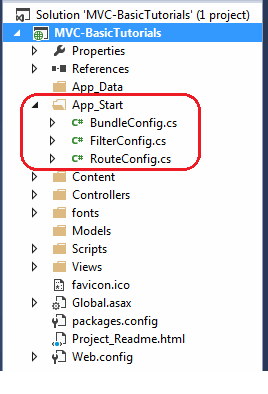
Let's see significance of each folder.

App\_Data:

App\_Data folder can contain application data files like LocalDB, .mdf files, xml files and other data related files. IIS will never serve files from App\_Data folder.

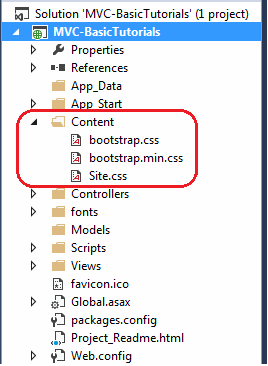
App\_Start:

App\_Start folder can contain class files which will be executed when the application starts. Typically, these would be config files like AuthConfig.cs, BundleConfig.cs, FilterConfig.cs, RouteConfig.cs etc. MVC 5 includes BundleConfig.cs, FilterConfig.cs and RouteConfig.cs by default. We will see significance of these files later.

[](http://www.tutorialsteacher.com/Content/images/mvc/appstart.png)App\_Start Folder

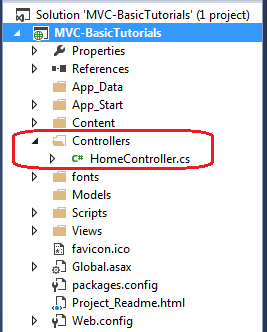
Content:

Content folder contains static files like css files, images and icons files. MVC 5 application includes bootstrap.css, bootstrap.min.css and Site.css by default.

[](http://www.tutorialsteacher.com/Content/images/mvc/content-folder.png)Content Folder

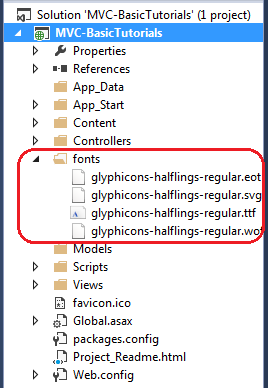
Controllers:

Controllers folder contains class files for the controllers. Controllers handles users' request and returns a response. MVC requires the name of all controller files to end with "Controller". You will learn about the controller in the next section.

[](http://www.tutorialsteacher.com/Content/images/mvc/controller-folder.png)Controller Folder

fonts:

Fonts folder contains custom font files for your application.

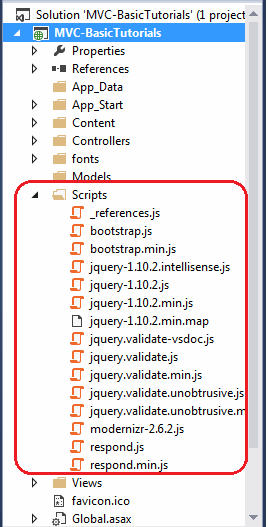
[](http://www.tutorialsteacher.com/content/images/mvc/fonts-folder.png)Fonts folder

Models:

Models folder contains model class files. Typically model class includes public properties, which will be used by application to hold and manipulate application data.

Scripts:

Scripts folder contains JavaScript or VBScript files for the application. MVC 5 includes javascript files for bootstrap, jquery 1.10 and modernizer by default.

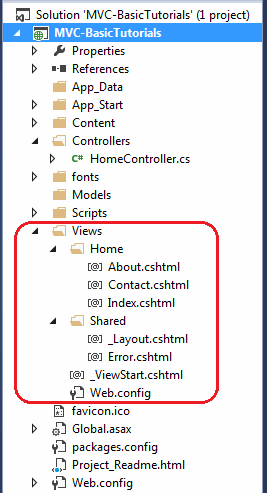
[](http://www.tutorialsteacher.com/Content/images/mvc/scripts-folder.png)Scripts Folder

Views:

Views folder contains html files for the application. Typically view file is a .cshtml file where you write html and C# or VB.NET code.

Views folder includes separate folder for each controllers. For example, all the .cshtml files, which will be rendered by HomeController will be in View > Home folder.

Shared folder under View folder contains all the views which will be shared among different controllers e.g. layout files.

[](http://www.tutorialsteacher.com/Content/images/mvc/view-folder.png)View Folder

Additionally, MVC project also includes following configuration files:

Global.asax:

Global.asax allows you to write code that runs in response to application level events, such as Application\_BeginRequest, application\_start, application\_error, session\_start, session\_end etc.

Packages.config:

Packages.config file is managed by NuGet to keep track of what packages and versions you have installed in the application.

Web.config:

Web.config file contains application level configurations.