MVC FRAMEWORK - INTRODUCTION

http://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

What is MVC?

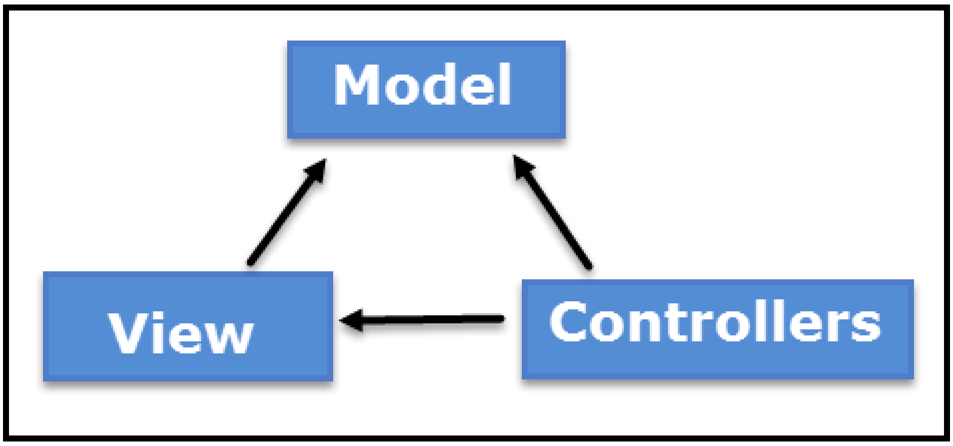
The **Model-View-Controller** *MVC* is an architectural pattern that separates an application into three main logical components: the **model**, the **view**, and the **controller**. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

MVC Components

Model: The Model component corresponds to all the data related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View: The View component is used for all the UI logic of the application. For example, the Customer view would include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller: Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller would handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller would be used to view the Customer data.



ASP.NET MVC

ASP.NET supports three major development models: Web Pages, Web Forms and MVC *ModelViewController*. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that is integrated with existing ASP.NET features, such as master pages, authentication, etc. Within .NET, this framework is defined in the System.Web.Mvc assembly. The latest version of the MVC Framework is 5.0. We use Visual Studio to create ASP.NET MVC applications which can be added as template in Visual Studio.

ASP.NET MVC Features

The ASP.NET MVC provides the following features:

- Ideal for developing complex but light weight applications
- It provides an extensible and pluggable framework which can be easily replaced and customized. For example, if you do not wish to use the in-built Razor or ASPX View Engine,

then you can use any other third-party view engines or even customize the existing ones.

- Utilizes the component-based design of the application by logically dividing it into Model, View and Controller components. This enables the developers to manage the complexity of large-scale projects and work on individual components.
- The MVC structure enhances the test-driven development and testability of the application since all the components can be designed interface-based and tested using mock objects. Hence the ASP.NET MVC Framework is ideal for projects with large team of web developers.
- Supports all the existing vast ASP.NET functionalities such as Authorization and Authentication, Master Pages, Data Binding, User Controls, Memberships, ASP.NET Routing, etc.
- It does not use the concept of View State $whichispresentinASP.NET$. This helps in building applications which are light-weight and gives full control to the developers.

Thus, you can consider MVC Framework as a major framework built on top of ASP.NET providing a large set of added functionality with focus on component-based development and testing.