**\* MVC Architecture**

The MVC model defines web applications with 3 logic layer:

* The business logic (Model Logic).
* The display layer (View Logic).
* The input control (Controller Logic).

**MODEL**

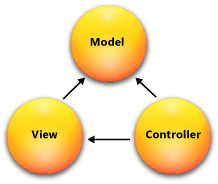
* The Model is responsible for managing the data of application.
* It responds to the request from the view and it also responds to instructions from the controller update itself.
* It is the lowest level of the pattern which is responsible for maintaining data.
* The Model represents the application core (for instance of a list of database records).
* It is also called the domain layer

**VIEW**

* The View display the data (the database records).
* A view requests information from the model, that it needs to generate an output representation.
* It presents data in a particular format like JSP, ASP, PHP, CSHTML.
* MVC is often seen in web applications, where the view is the HTML page.

**CONTROLLER**

* The Controller is the part of application that handles user interaction.
* Typically controllers read data from a view, control user input and send input data to the model.
* It handle the input, typically user actions and may invoke changes on the model and view. For example, the controller handles query-string values, and passes these values to the model, which in turn might use these values to query the database.



**MVC architecture diagram (references to msdn.microsoft.com)**

**Advantages of an MVC-Based Web Application**

The ASP.NET MVC framework offers the following advantages:

* It makes it easier to manage complexity by dividing an application into the model, the view, and the controller.
* It does not use view state or server-based forms. This makes the MVC framework ideal for developers who want full control over the behavior of an application.
* It uses a Front Controller pattern that processes Web application requests through a single controller. This enables you to design an application that supports a rich routing infrastructure.
* It provides better support for test-driven development (TDD).
* It works well for Web applications that are supported by large teams of developers and for Web designers who need a high degree of control over the application behavior.