MVC Architecture

The main aim of the MVC architecture is to separate the business logic and application data from the presentation data to the user.

Here are the reasons why we should use the MVC design pattern.

1. They are **resuable** : When the problems recurs, there is no need to invent a new solution, we just have to follow the pattern and adapt it as necessary.
2. They are **expressive**: By using the MVC design pattern our application becomes more expressive.

1). **Model:** The model object knows about all the data that need to be displayed. It is model who is aware about all the operations that can be applied to transform that object. It only represents the data of an application. The model represents enterprise data and the business rules that govern access to and updates of this data. Model is not aware about the presentation data and how that data will be displayed to the browser.

2). **View :** The view represents the presentation of the application. The view object refers to the model. It uses the query methods of the model to obtain the contents and renders it. The view is not dependent on the application logic. It remains same if there is any modification in the business logic. In other words, we can say that it is the responsibility of the of the view's to maintain the consistency in its presentation when the model changes.

3). **Controller:** Whenever the user sends a request for something then it always go through the controller. The controller is responsible for intercepting the requests from view and passes it to the model for the appropriate action. After the action has been taken on the data, the controller is responsible for directing the appropriate view to the user. In GUIs, the views and the controllers often work very closely together.

Features of MVC2:

1. The MVC2 architecture removes the page centric property of MVC1 architecture by separating Presentation, control logic and the application state.
2. In MVC2 architecture there is only one controller which receives all the request for the application and is responsible for taking appropriate action in response to each request.