**MVC**

* The Model-View-Controller (MVC) it is design pattern assigns objects in an application one of three roles: model, view, or controller.
* MVC It is Standard design pattern it is used by all the language like java,c#

to develop web based or mobile application. It is standard Architecture.

* In iOS We also we use same design pattern used for developing iphone Applications.
* By Using MVC We Can write Code Separately Business Logic Separate, Presentation Logic Separate ,Database Connection Separate.
* Anything We want to Change You Change Particulary that only it will not affect others classes.
* **Model**: Represents the business logic of your application.
* **View**: Represents what the user sees in the device.
* **Controller:** Acts as a mediator between the Model and View.
* There should not be any direct conversation between the View and the Model.
* The Controller updates the View based on any changes in the underlying Model.
* If the user enters or updates any information in the View, the changes are reflected in the Model with the help of the Controller.
* The pattern defines not only the roles objects play in the application, it defines the way objects communicate with each other.
* Each of the three types of objects is separated from the others by abstract boundaries and communicates with objects of the other types across those boundaries.
* Controller handles Database records.



**Fig: Model View Controller Architecture**

**Model Objects:**

* The model is responsible for managing the data of the application.
* It responds to the request from the view and it also responds to instructions from the controller to update itself.
* We write Business Logic for Application.
* Model We store User Data in Database.

**Communication:**: User will interact with in the view layer Through controller.

* The View Layer that create or modify data are communicated through a controller object and result in the creation or updating of a model object.
* When a model object changes (for example, new data is received over a network connection), it notifies a controller object, which updates the appropriate view objects.

**View Objects:**

* View Object in an application the user can see
* It is used for Presentation.
* view Object it is used for display the data from application model Objects.
* There are Many Other technology are there for view like html, jsp, jsf,

php these all are used to work with user Only.

**Communication**: There is no interaction directly through Model and view.

* If User make any changes in View the changes are reflected in the Model with the help of the Controller.

**Controller Objects:**

* The controller is responsible for responding to user input. and Controller perform interactions on the data model objects.
* The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

**Communication**:

* A controller object interprets user entered value made in view objects and communicates new or changed data to the model layer.
* When model objects change, a controller object communicates that new model data to the view objects so that they can display it.

**Advantages**: By Using MVC Architecture We write All the Code in Separate classes if anything you want to change you can change easily it will not affect others classes and Objects.

**There are Many Advantages using MVC**

* Navigation control is centralized Now only controller contains the logic to determine the next page.
* Easy to maintain
* Easy to extend
* Easy to test
* Better separation of concerns.