**Q3**

MVC is a pattern for the architecture of a software application. It separates an application into these following components:

1: Models is for handling data but it does not deal with any logic about how to present the data.

2: Views is for handling graphical user interface objects and presentation it deals with how to link up with the model's data

3: Controllers is for handling the user interface, application and relation between models and views and performs an appropriate response back to the events. All the logic is the controller eg receive user input and decide what to do with it

**Q4**

The core features of AngularJS are as follows −

**Data-binding** − It is the automatic synchronization of data between model and view components.

**Scope** − These are objects that refer to the model. They act as a glue between controller and view.

**Controller** − These are JavaScript functions bound to a particular scope.

**Services** − AngularJS comes with several built-in services such as $http to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.

**Filters** − These select a subset of items from an array and returns a new array.

**Directives** − Directives are markers on DOM elements such as elements, attributes, css, and more. These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives such as ng-Bind, ngModel, etc.

**Templates** − These are the rendered view with information from the controller and model. These can be a single file (such as index.html) or multiple views in one page using *partials*.

**Routing** − It is concept of switching views.

**Model View Whatever** − MVW is a design pattern for dividing an application into different parts called Model, View, and Controller, each with distinct responsibilities. AngularJS does not implement MVC in the traditional sense, but rather something closer to MVVM (Model-View-ViewModel). The Angular JS team refers it humorously as Model View Whatever.

**Deep Linking** − Deep linking allows to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.

**Dependency Injection** − AngularJS has a built-in dependency injection subsystem that helps the developer to create, understand, and test the applications easily.