Assignment 1 MVC

Ques 1. Explain MVC and the benefit of MVC.

Answer: MVC stands for Model View Controller.MVC separates application into three components - Model, View and Controller.

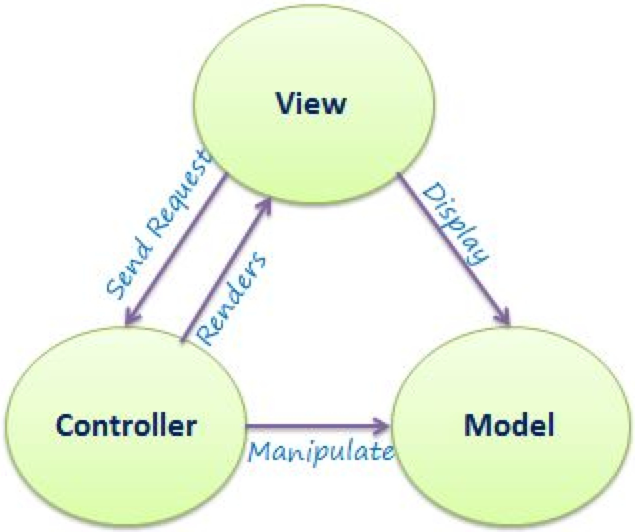
1. Model is a data and business logic.
2. View is a User Interface.
3. Controller is a request handler.

Model: Model represents shape of the data and business logic. It maintains the data of the application. Model objects retrieve and store model state in a database.

View: View is a user interface. View display data using model to the user and also enables them to modify the data.

Controller: Controller handles the user request. when the user enters a URL in the browser, it goes to the server and calls appropriate controller. Then, the Controller uses the appropriate View and Model and creates the response and sends it back to the user.

The following figure illustrates the interaction between Model, View and Controller.



Advantages of the MVC:

- We can easily maintain our application because of separation of concernSeparation of concern means we divide the application Model, Control   and View..

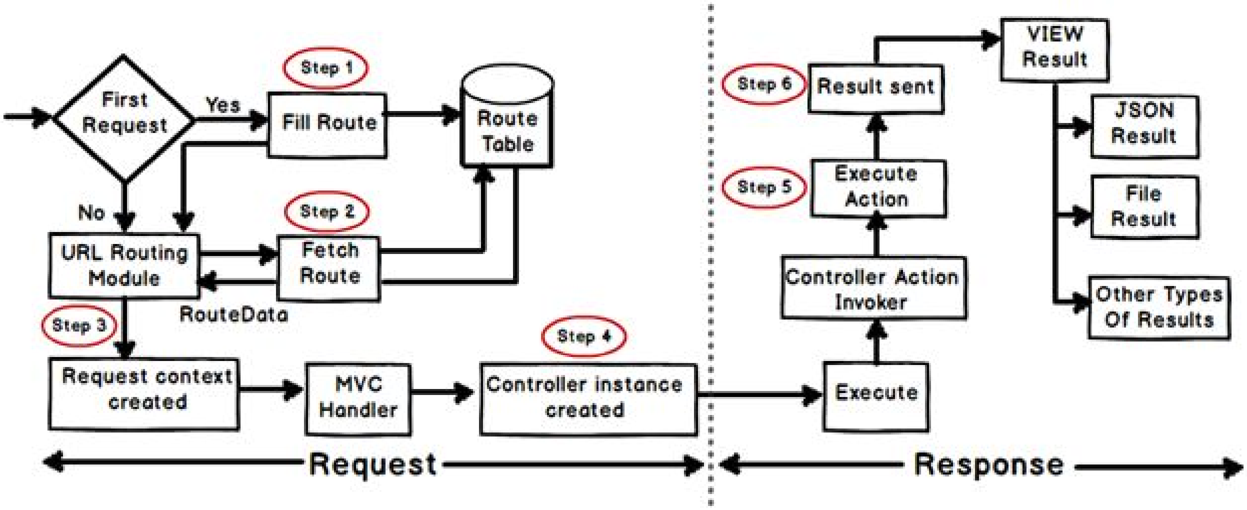
- In the same time we can split many developers work at a time. It will not affects one developer work to another developer work.

- It supports TTD (test-driven development). We can create an application with unit test. We can write own test case.

- Latest version of MVC Support default responsive web site and mobile templates.

Ques2. Explain MVC Life Cycle

Answer:



Step 1 Fill route: - MVC requests are mapped to route tables which in turn specify which controller and action to be invoked. So if the request is the first request the first thing is to fill the route table with routes collection. This filling of route table happens in the global.asax file.

Step 2 Fetch route:- Depending on the URL sent "UrlRoutingModule" searches the route table to create "RouteData" object which has the details of which controller and action to invoke.

Step 3 Request context created: - The "RouteData" object is used to create the "RequestContext" object.

Step 4 Controller instance created: - This request object is sent to "MvcHandler" instance to create the controller class instance. Once the controller class object is created it calls the "Execute" method of the controller class.

Creating Response object: - This phase has two steps executing the action and finally sending the response as a result to the view.

Step 5 Execute Action: - The "ControllerActionInvoker" determines which action to executed and executes the action.

Step 6 Result sent: - The action method executes and creates the type of result which can be a view result , file result , JSON result etc.

Ques4. Why we use App_Start, App_Data and Shared Folder..

Answer:

App_Data- folder can contain application data files like LocalDB, .mdf files, xml files and other data related files.

App_Start- App_Start folder can contain class files which will be executed when the application starts. These would be config files like AuthConfig.cs, BundleConfig.cs, FilterConfig.cs, RouteConfig.cs etc.

Shared Folder-Shared folder under View folder contains all the views which will be shared among different controllers e.g. layout files.

Ques 5.Use of Global. Asax and difference b/w global Web config and View Web Config. (T)

Answer:

Global.asax:

Global.asax allows you to write code that runs in response to application level events, such as Application_BeginRequest, application_start, application_error, session_start, session_end etc.
In this file we register the different config file which are defined in the app_start folder like BundleConfig.cs,RouteConfig.cs etc.

Difference between global Web config and View Web Config.

Global Web config:

- Web.config file contains application level configurations.
- Web.config file is an XML file, which is used to define Connection strings, Session variables, Authentication mechanism,Global variables and so on.

View Web Config:

- Web.config file contains view level configurations.

- The web.config file exists in the Views folders to prevent access to your views by any means other than your controller

Reason to have two web.config in mvc project is:

> The controller are responsible to route your request based on the identified (and valid) pattern, do some operation on the model and return a rendered view based on the operation performed as response.
>
> So logically if the view folder and view (webpages) are accessible or can be requested by user then the basic MVC pattern and practice followed by Microsoft will be violated.
>
> This is prevented by adding a web.config and blocking access to view folder and files directly via user request or through URL.