Module-3

1. What is model?

Model represents the shape of the data. A class in C# is used to describe a model. Model objects store data retrieved from the database.

1. Explain working of MVC Pattern.

-MVC is an architectural pattern consisting of three parts: Model, View, Controller

Model: handles data logic.

View: it displays the information from the model to the user.

Controller: it controls the data flow into a model object and updates the view whenever data changes.

1. what is routing in MVC?

MVC routing is a process of mapping the browser request to the controller action and return response back. Each application has default routing for the default HomeController. we can set custom routing for newly created controller.

1. What is a View?

View in MVC is a user interface. View display model data to the user and also enables them to modify them. View in ASP.NET MVC is HTML, CSS, and some special syntax (Razor syntax) that makes it easy to communicate with the model and the controller.

1. What is the importance of Controller?

A controller is responsible for controlling the way that a user interact with an MVC application. A controller contains the flow control logic for an ASP.NET MVC application. A controller determines what response to send back to a user when a user makes a browser request.

1. What is difference between MVC and Web Forms?

- A web application framework developed by Microsoft, which implements the model-view-controller

- and in mvc is models supported by the Microsoft ASP.NET technology.

- Web forms follows web forms syntax.

- MVC in customizable sysntax(razor as default)

1. Explain what is Model.

The model is the part of MVC which implements the domain logic. In simple terms the logic is used to handle the data passed between the database and the user interface (UI).

How is Model different from DataLayer of WebForm?

1. Explain the concept of MVC Scaffolding?

Scaffolding is used to defin the code-generation framework used in web applications. It generates instance for the mapped domain model and code for all CRUD operations.

1. What is RouteData and how to access its values?

Routedata is a property of the base controller class so routedata can be accessed in any controller. RouteData contains route information of a current request. You can get the controller, action or parameter information using routedata.

1. Differences between Razor and ASPX View Engine in MVC?

|  |  |
| --- | --- |
| **Razor View Engine** | **ASPX View Engine (MVC)** |
| The namespace used by the Razor View Engine is System.Web.Razor | The namespace used by the ASPX View Engine is System.Web.Mvc.WebFormViewEngine |
| The file extensions used by the Razor View Engine are different from a web form view engine. It uses cshtml with C# and vbhtml with vb for views, partial view, templates and layout pages. | The file extensions used by the Web Form View Engines are like ASP.Net web forms. It uses the ASPX extension to view the aspc extension for partial views or User Controls or templates and master extensions for layout/master pages. |
| Razor has a syntax that is very compact and helps us to reduce typing. | The web form view engine has syntax that is the same as an ASP.Net forms application. |
| The Razor View Engine uses @ to render server-side content. | The ASPX/web form view engine uses "<%= %>" or "<%: %>" to render server-side content. |
| By default all text from an @ expression is HTML encoded. | There is a different syntax ("<%: %>") to make text HTML encoded. |
| Razor does not require the code block to be closed, the Razor View Engine parses itself and it is able to decide at runtime which is a content element and which is a code element. | A web form view engine requires the code block to be closed properly otherwise it throws a runtime exception. |
| The Razor View Engine prevents Cross Site Scripting (XSS) attacks by encoding the script or HTML tags before rendering to the view. | A web form View engine does not prevent Cross Site Scripting (XSS) attack. |
| The Razor Engine supports Test Driven Development (TDD). | Web Form view engine does not support Test Driven Development (TDD) because it depends on the System.Web.UI.Page class to make the testing complex. |
| Razor uses "@\* … \*@" for multiline comments. | The ASPX View Engine uses "<!--...-->" for markup and "/\* … \*/" for C# code. |
| There is only three transition characters with the Razor View Engine. | There are only three transition characters with the Razor View Engine. |

1. Explain various return types of an action method.

An ActionResult is a return type of a controller method in MVC. Action method help us to return model to view , file streams , and also redirect to another controller’s Action method.

1. What is Razor in MVC?

Razor is a markup sysntax that lets you embed server-based code into web pages using c# and vb.net. it is not a programming languages. It is a server side markup languages. You can use it anywhere to generate output like HTML.

1. Explain custom middleware and where to add.

The custom middleware component like any other .NET class with Invoke() method. However in order to execute next middleware in a sequence, it should have RequestDelegate type parameter in the constructor.

* Add -> New Item and search middleware. Select middleware class item and give it a name and click on add button. This will add new class for the middleware with extension method.

Create an application to add a Custom Middleware class and call it from the view.

1. Why is Layout page important?

The basic objectives of layout is to ensure a smooth flow of work, material and information through a system. The basic meaning of facility is the space in which a business’s activities take place.

Download a Template from the Internet and integrate it into an application.

1. What are sections?

In ASP.NET MVC a section is a piece of code that we want to render in layout page. This allows us to render a specific view content in any location of a layout. In order to declare a section in a view , we use the @section keyword followed by the name of the section.

1. Create a section in the layout page and call it in the derived views.