**MODULE 4 [ MVC Architecture]**

1. **What is model?**

* The Model component corresponds to all the data-related logic that the user works with.
* This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data.
* For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

1. **What is routing in MVC?**

* In MVC, routing is a process of mapping the browser request to the controller action and return response back. Each MVC application has default routing for the default HomeController.

1. **What is the difference between Temp data, View, and View Bag?**

|  |  |  |
| --- | --- | --- |
| ViewData | ViewBag | TempData |
| It is Key-Value Dictionary collection | It is a type object | It is Key-Value Dictionary collection |
| ViewData is a dictionary object and it is property of ControllerBase class | ViewBag is Dynamic property of ControllerBase class. | TempData is a dictionary object and it is property of controllerBase class. |
| ViewData is Faster than ViewBag | ViewBag is slower than ViewData | NA |
| ViewData is introduced in MVC 1.0 and available in MVC 1.0 and above | ViewBag is introduced in MVC 3.0 and available in MVC 3.0 and above | TempData is also introduced in MVC1.0 and available in MVC 1.0 and above. |
| ViewData also works with .net framework 3.5 and above | ViewBag only works with .net framework 4.0 and above | TempData also works with .net framework 3.5 and above |
| Type Conversion code is required while enumerating | In depth, ViewBag is used dynamic, so there is no need to type conversion while enumerating. | Type Conversion code is required while enumerating |
| Its value becomes null if redirection has occurred. | Same as ViewData | TempData is used to pass data between two consecutive requests. |
| It lies only during the current request. | Same as ViewData | TempData only works during the current and subsequent request |

1. **What is difference between MVC and Web Forms?**

* MVC or ASP.NET MVC is a web application framework developed by Microsoft, which implements the model–view–controller (MVC) pattern.  ASP.NET Web Forms is a web application framework and one of several programming models supported by the Microsoft ASP.NET technology.

1. **What is session? What is the default time for session?**

* Session is a server side state management technique.
* It is store user information on server.
* ASP.NET MVC Session state enables you to store and retrieve values for a user when user navigates other view in an ASP.NET MVC application.
* A session is defined as the period of time that a unique user interacts with a Web application.
* Session variable we can access in whole application the default time of session will be 20 min.

1. **What is Partial View in MVC? With example**

* In ASP.NET MVC, a partial view is analogous to user controls in ASP.NET Web Forms.
* A partial view is a chunk of HTML that can be safely inserted into an existing DOM.
* Most commonly, partial views are used to componentized Razor views and make them easier to build and update
* Partial view is a reusable view, which can be used as a child view in multiple other views. It eliminates duplicate coding by reusing same partial view in multiple places. You can use the partial view in the layout view, as well as other content views.
* @Html.Partial() helper method renders the specified partial view. It accept partial view name as a string parameter and returns MvcHtmlString. It returns html string so you have a chance of modifingthe html before rendering

1. **What is the difference between View and Partial View?**

* **View -**

1.      View contains the layout page

2.      Before any view is rendered, viewstart page is rendered

3.      View might have markup tags like body, html, head, title, Meta etc.

4.      View is not lightweight as compare to Partial View

* **Partial View-**

1.      Partial View does not contain the layout page

2.      Partial view does not verify for a viewstart.cshtml page.

3.      Partial view is designed specially to render within the view and just because of that it does not consist any mark up

4.      We can pass a regular view to the RenderPartial method

1. **Explain the concept of MVC Scaffolding?**

* Scaffolding is used to define the code-generation framework used in web applications. It uses T4 templates to generate basic controllers and views for the models. It generates instances for the mapped domain model and code for all CRUD operations. It also reduces the amount of time for developing a standard data operation in the application.

1. **How to change time of session?**

* You can increase the time out value in minutes using the timeout attribute of sessionState element in web.config.

1. **What is query string? What are disadvantages of query string?**

* A Query String is information sent to the server appended to the end of a page URL which means that Query String is way to transfer information from one page to another through the URL. Query String is attached to the URL with "?".
* **Disadvantages:**

1.    There is a limit to URL length of 255 characters.

2.    Query String data is directly visible to user thus leading to security problems

1. **What is cookie? What are limitations for cookie?**

* A cookie is a small piece of data stored on a user's computer by a web browser while the user is browsing a website. Cookies are commonly used to remember information about the user or their preferences, allowing websites to provide a more personalized experience. They can store various types of information, such as login credentials, site preferences, shopping cart contents, and more.
* However, cookies come with certain limitations:
* Size Limitation, Security Concerns, Cross-Domain Limitations, Privacy Concerns..

1. **Explain MVC application life cycle.**

* The Model-View-Controller (MVC) architectural pattern is commonly used in software development, especially in web applications. The MVC pattern separates an application into three interconnected components: Model, View, and Controller. The life cycle of an MVC application involves the interactions between these three components to handle user requests and generate responses.
* The typical life cycle of an MVC application:
* **Request Handling:**

A user initiates a request by interacting with the application through the user interface (UI), such as clicking a link or submitting a form.

The request is sent to the application's front controller (often the Router in web applications).

* **Routing:**

The front controller (Router) determines which controller should handle the request based on the URL or route defined in the application.

The Router maps the incoming request to a specific Controller action or method.

* **Controller Handling:**

The Controller receives the request and processes it. It contains the application logic to handle the user's input.

The Controller may interact with the Model to retrieve or update data, perform business logic, or prepare data for the View.

* **Model Interaction:**

The Controller communicates with the Model, which represents the application's data and business rules.

The Model interacts with the database or other data sources to fetch, manipulate, or update data as required by the Controller.

* **View Rendering:**

Once the Controller has processed the request and obtained necessary data from the Model, it passes this data to the appropriate View.

The View is responsible for presenting the data to the user in a user-friendly format (HTML, XML, JSON, etc.).

The View may use templates or other rendering mechanisms to generate the final output.

* **Response Generation:**

The rendered View, along with the processed data, is sent back to the user's browser as a response to the initial request.

The browser displays the response to the user, presenting the requested information or performing the requested action.

* **User Interaction and Feedback Loop:**

The user interacts with the UI again, triggering another request, and the cycle continues.

* This cycle repeats for each user interaction, with the Controller handling requests, interacting with the Model to manipulate data, and passing the processed data to the View for presentation to the user. The separation of concerns among these components (Model, View, and Controller) in the MVC pattern helps in maintaining code organization, reusability, and scalability in applications.

1. **List out different return types of a controller action method**

* All return types are use to return results from controller to view.
* base type of all result types is ActionResult.
* **ViewResult (View) :** It is used to return a webpage from an action method.
* **PartialviewResult (Partialview) :** It is used to send a part of a view which will be rendered in another view.
* **RedirectResult (Redirect) :** It is used to redirect to any other controller and action method depending on the URL.
* **RedirectToRouteResult (RedirectToAction, RedirectToRoute) :** It is used when we want to redirect to any other action method.
* **ContentResult (Content) :** It is used to return HTTP content type like text/plain as the result of the action.
* **jsonResult (json) :** It is used when we want to return a JSON message.
* **javascriptResult (javascript) :** It used to return JavaScript code that will run in browser.
* **FileResult (File) :** It is used to send binary output in response.
* **EmptyResult :** It is used to return nothing (void) in the result.

1. **What are filters in MVC?**

* ASP.NET MVC Filter is a custom class where you can write custom logic to execute before or after an action method executes. Filters can be applied to an action method or controller in a declarative or programmatic way.

1. **What are HTML helpers in MVC?**

* HTML Helpers in MVC (Model-View-Controller) frameworks, such as ASP.NET MVC, are methods used to generate HTML markup programmatically within views. They assist developers in creating HTML elements by generating the necessary HTML code dynamically.

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

* RAZOR View Engine
* The Razor view was introduced with ASP.NET MVC 3 and is the default view engine moving forward.
* The file extension for view, partial views are
* .cshtml - with c# language
* .vbhtml - with VB language
* The key transition character in Razor is the “at” sign (@). This character is used to switch from markup to code and sometimes again back to markup.
* **ASPX View Engine**
* This is the default view engine for the ASP.net MVC.
* The syntax for writing views with this engine is the same syntax that the ASP.NET Web Forms uses
* The file extensions are also taken from ASP.NET Web Form (.aspx, .ascx, .master) .It has .aspx extensions for views, .ascxextensions for Partial views,. master for layout and master pages.
* The server side script is wrapped between<% %>,