Layouts : Template for your views

\_ViewStart.cshtml > It is the first View which is called first

This is the file which contains the path & name of the layout(.cshtml) which has to be used in all the views

\_ViewStart.cshtml >

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

\_Layout.cshtml

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@ViewBag.Title - My ASP.NET Application</title>

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

@Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })

</div>

<div class="navbar-collapse collapse">

<ul class="nav navbar-nav">

<li>@Html.ActionLink("Home", "Index", "Home")</li>

<li>@Html.ActionLink("About", "About", "Home")</li>

<li>@Html.ActionLink("Contact", "Contact", "Home")</li>

</ul>

</div>

</div>

</div>

<div class="container body-content">

@RenderBody()

<hr />

<footer>

<p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>

</footer>

</div>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

</body>

</html>

It will remove Layout from all the Views

@{

Layout = null;

//Layout = "~/Views/Shared/\_Layout.cshtml";

}

Layouts are used to have common look and feel for all the views. So that they look similar

What should we put in Layout

Common contents that you want in every view

<!Doctype html>

<html>

<head>

<title> ASP Application </title>

</head>

<body style="background-color:palegoldenrod">

<center>

<div style="align-items:center">

<h1> ABC Ltd. </h1>

<p>

Attribute Routing gives us more control over the URIs in our MVC web application. The earlier way of routing (convention-based routing) is fully supported by this version of MVC. We can also use both type of routing in the same project.

</p>

<p>

Attribute Routing gives us more control over the URIs in our MVC web application. The earlier way of routing (convention-based routing) is fully supported by this version of MVC. We can also use both type of routing in the same project.

</p>

</div>

</center>

@RenderBody()

</body>

</html>

Because of RenderBody() method, contents of the views will appear. There can be only one RenderBody() method call in the layout. This method is mandatory

RenderSection() method is used to display some particular sections of the views . This is optional. There could be more than one calls to this method. Before we use it in the layout, we have to create sections in the views

Create Section in View

@section A1

{<b>

This is from Index View

</b>

}

Go to Layout

<!Doctype html>

<html>

<head>

<title> ASP Application </title>

</head>

<body style="background-color:palegoldenrod">

<center>

<div style="align-items:center">

<h1> ABC Ltd. </h1>

<p>

Attribute Routing gives us more control over the URIs in our MVC web application. The earlier way of routing (convention-based routing) is fully supported by this version of MVC. We can also use both type of routing in the same project.

</p>

<p>

Attribute Routing gives us more control over the URIs in our MVC web application. The earlier way of routing (convention-based routing) is fully supported by this version of MVC. We can also use both type of routing in the same project.

</p>

</div>

</center>

**@RenderSection("A1", required:false)**

**@RenderSection("A2", required:false)**

@RenderBody()

</body>

</html>

@model WebApplication15.Models.Student

@{

//Layout = null;

Layout = "~/Views/Shared/\_Layout.cshtml";

}

@{

ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Student</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Marks, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Marks, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Marks, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.BatchCode, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.BatchCode, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.BatchCode, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Create" class="btn btn-default" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>