2.11 Partial Views



This section will guide you to:

* Implement partial views in your frontend and handle with controllers

This guide has nine subsections, namely:

2.11.1 Create an ASP.NET MVC project to use partial views

2.11.2 Create a partial view for left bar

2.11.3 Create StudentInfo.cshmtl as a placeholder page

2.11.4 Create TeacherInfo.cshmtl as a placeholder page

2.11.5 Change Index.cshtml to show links to the pages

2.11.6 Make changes to HomeController to handle the new views

2.11.7 Build the project

2.11.8 Publish and run the project

2.11.9 Pushing the code to your GitHub repositories

**Step 2.11.1:** Create an ASP.NET MVC project to use partial views

* Open Visual Studio.
* From the top menu, select **File->New->Project.**
* In **Create A New Project** screen, select **ASP.NET Core Web Application** from the list of available project types and click on **Next.**
* Enter **Project Name** as **Phase3Section2.22** and click on **Create.**
* From the list of project sub-types, choose **Web Application (Model-View-Controller)** and uncheck **Configure for HTTPS.** Click on **Create.**
* This will create the files for an ASP.NET MVC project.

**Step 2.11.2:** Create a partial view for left bar

* In **Solution Explorer**,expand **Views->Home.** Right click **Home** and choose **Add->View.**
* Put **View Name** as LeftBar. Check **Create as Partial View** and click on **Add**.
* Add the following script:

@\*

For more information on enabling MVC for empty projects, visit http://go.microsoft.com/fwlink/?LinkID=397860

\*@

<**ul**>

<**li**><**a** href="/Home/Index">Homepage</**a**></**li**>

<**li**><**a** href="/Home/StudentInfo">Student Info</**a**></**li**>

<**li**><**a** href="/Home/TeacherInfo">Teacher Info</**a**></**li**>

</**ul**>

**Step 2.11.3:** Create StudentInfo.cshmtl as a placeholder page

* In **Solution Explorer**,expand **Views->Home.** Right click **Home** and choose **Add->View.**
* Put **View Name** as StudentInfo and  click on **Add.**
* Add the following script:

@{

ViewData["Title"] = "StudentInfo";

}

<**div** class="container">

<**br** />

<**div** class="row">

<**div** class="col-sm-3">

@Html.Partial("LeftBar");

</**div**>

<**div** class="col-sm-9">

This is the student info page

</**div**>

</**div**>

</**div**>

**Step 2.11.4:** Create TeacherInfo.cshmtl as a placeholder page

* In **Solution Explorer**,expand **Views->Home.** Right click **Home** and choose **Add->View.**
* Put **View Name** as TeacherInfo and  click on **Add.**
* Add the following script:

@{

ViewData["Title"] = "TeacherInfo";

}

<**div** class="container">

<**br** />

<**div** class="row">

<**div** class="col-sm-3">

@Html.Partial("LeftBar");

</**div**>

<**div** class="col-sm-9">

This is the Teacher info page

</**div**>

</**div**>

</**div**>

**Step 2.11.5:** Change Index.cshtml to show links to the pages

* In **Solution Explorer**, expand **Views->Home** and double-click index.cshtml.
* Enter the following script:

@{

ViewData["Title"] = "Home Page";

}

<**div** class="container">

<**br** />

<**div** class="row">

<**div** class="col-sm-3">

@Html.Partial("LeftBar");

</**div**>

<**div** class="col-sm-9">

This is the homepage

</**div**>

</**div**>

</**div**>

**Step 2.11.6:** Make changes to HomeController to handle the new views

* In **Solution Explorer,** expand **Controllers** and double click **HomeController**.
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Diagnostics;

**using** System.Linq;

**using** System.Threading.Tasks;

**using** Microsoft.AspNetCore.Mvc;

**using** Phase3Section2.\_22.Models;

**namespace** Phase3Section2.\_22.Controllers

{

**public** **class** HomeController : Controller

{

**public** IActionResult Index()

{

**return** View();

}

**public** IActionResult About()

{

ViewData["Message"] = "Your application description page.";

**return** View();

}

**public** IActionResult Contact()

{

ViewData["Message"] = "Your contact page.";

**return** View();

}

**public** IActionResult Privacy()

{

**return** View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = **true**)]

**public** IActionResult Error()

{

**return** View(**new** ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

**public** ViewResult StudentInfo()

{

**return** View();

}

**public** ViewResult TeacherInfo()

{

**return** View();

}

}

}

**Step 2.11.7:** Build the project

* From the top menu, choose **Build->Build Solution.**
* If any compile errors are shown, fix them as required.

**Step 2.11.8:** Publish and run the project

* From the top menu, select **Debug->Start Without Debugging.**
* This will execute the program in the default browser.
* To see the student pages, go to the url : http://localhost:xxxx/students.

**Step 2.11.9:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master