2.2 Various ActionResults

This section will guide you to:

* Create an ASP.NET MVC project to show different **ActionResults**.

This guide has eleven subsections, namely:

2.2.1 Creating an ASP.NET MVC project

2.2.2 Adding **NewtonSoft.JSON** using **Nuget**

2.2.3 Adding two data files – **StudentData.txt** and **StudentData.json**

2.2.4 Creating a View \_**studentinfo.cshtml**

2.2.5 Creating a View **StudentAsView.cshtml**

2.2.6 Creating a View **StudentList.cshtml**

2.2.7 Changing **Index.cshtml** to show various **ActionResult** options

2.2.8 Making changes to **HomeController** to work with the **ActionResults**

2.2.9 Building the Project

2.2.10 Publishing and running the Project

2.2.11 Pushing the code to your GitHub repositories

**Step 2.2.1:** Creating an ASP.NET MVC project to show different ActionResults to display Student data

* Open Visual Studio.
* From the top menu select **File->New->Project**
* In **Create A New Project** screen, select **ASP.NET Web Application(.Net Framework)** from the list of available project types and click on **Next.**
* Please select the project type where the **language** of the project is **C#.**
* Enter **Project Name** as Phase3Section2.5 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 2.2.2:** Adding NewtonSoft.JSON using Nuget

* From the top menu go to **Tools->Nuget Package Manager->Package Manager Console**
* Type install-package newtonsoft.json and press Enter
* This will install NewtonSoft.JSON as part of the project

**Step 2.2.3:** Adding two data files – StudentData.txt and StudentData.json

* In **Solution Explorer,** expand **wwwroot.**  Right click **wwwroot** and choose **Add->New Item**
* From the file types choose **Text File.** Enter **Name** as StudentData.txt and click **Add**
* Enter the following data:

Charlie Smith, Ralph Stevens, Anne Hathway, Jim Crosby

* Right click **wwwroot** and choose **Add->New Item**
* From the file types choose **Text File.** Enter **Name** as StudentData.json and click **Add**
* Enter the following data:

{

"name": "Walter Isaacs",

"Class": "7A",

"Subjects": [ "English", "Maths", "Geography" ]

}

**Step 2.2.4:** Creating a View \_studentinfo.cshtml

* In the **Solution Explorer,** expand **Views->Home.** Right click **Home** and choose **Add->View**
* Enter **View Name** as \_studentinfo and check **Create as Partial View.**
* Click **Add**
* Enter the following script:

<**h4**>List Of Students</**h4**>

<**ul**>

<**li**>Adam Oak</**li**>

<**li**>Phil Collins</**li**>

<**li**>Mary Thatcher</**li**>

<**li**>Pat Single</**li**>

</**ul**>

**Step 2.2.5:** Creating a View StudentAsView.cshtml

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

@{

ViewData["Title"] = "StudentAsView";

}

<**h2**>StudentAsView</**h2**>

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

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

<**div** class="col-sm-12">

@Html.Raw(ViewData["message"])

</**div**>

</**div**>

</**div**>

**Step 2.2.6:** Creating a View StudentList.cshtml

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

@{

ViewData["Title"] = "StudentList";

}

<**h2**>StudentAsView</**h2**>

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

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

<**div** class="col-sm-12">

@Html.Partial("\_studentInfo");

</**div**>

</**div**>

</**div**>

**Step 2.2.7:** Changing Index.cshtml to show various ActionResult options

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

@{

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

}

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

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

<**br** />

<**div** class="col-sm-12">

<**a** href="Home/StudentAsString">Student Data as String using ContentResult</**a**><**br** />

<**a** href="Home/StudentAsView">Student Data as ContentView</**a**><**br** />

<**a** href="Home/StudentAsRedirectResult">Student Search using RedirectResult</**a**><**br** />

<**a** href="Home/StudentAsRedirectToRouteResult">Student Data using RedirectToRouteResult</**a**><**br** />

<**a** href="Home/StudentAsFileResult">Student Data using FileResult</**a**><**br** />

<**a** href="Home/StudentAsJSONResult">Student Data using JSONResult</**a**><**br** />

<**a** href="Home/StudentList">Student Data Using PartialView</**a**><**br** />

</**div**>

</**div**>

</**div**>

**Step 2.2.8:** Making changes to HomeController to work with the ActionResults

* In the **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** Newtonsoft.Json;

**using** Newtonsoft.Json.Linq;

**using** Phase3Section2.\_4.Models;

**namespace** Phase3Section2.\_4.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** ContentResult StudentAsString()

{

**return** Content("The name of the student is Walter Smith and he is in Grade 7.");

}

**public** ViewResult StudentAsView()

{

ViewData["message"] = "The name of the student is <b>Walter Smith</b> and he is in Grade 7.";

**return** View();

}

**public** RedirectResult StudentAsRedirectResult()

{

**return** **new** RedirectResult("https://www.bing.com/search?q=walter+smith");

}

**public** RedirectToRouteResult StudentAsRedirectToRouteResult()

{

**return** **new** RedirectToRouteResult(**new** { Controller = "Home", action = "StudentAsView" });

}

**public** FileResult StudentAsFileResult()

{

**byte**[] fileBytes = System.IO.File.ReadAllBytes(@"./wwwroot/StudentData.txt");

**string** fileName = "StudentData.txt";

**return** File(fileBytes, System.Net.Mime.MediaTypeNames.Application.Octet, fileName);

}

**public** JsonResult StudentAsJSONResult()

{

String data = System.IO.File.ReadAllText(@"./wwwroot/StudentData.json");

JObject json = JObject.Parse(data);

**return** Json(json);

}

**public** ViewResult StudentList()

{

**return** View();

}

}

}

**Step 2.2.9:** Building the project

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

**Step 2.2.10:** Publishing and running the Project

* From the top menu select **Debug->Start Without Debugging**
* This will execute the program in the default browser

**Step 2.2.11:** 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