2.8 Form Submission Methods



This section will guide you to:

* Implement various methods to submit and read form data

This guide has eight subsections, namely:

2.8.1 Create an ASP.NET MVC project to show different types of form submits

2.8.2 Create FormGet.cshmtl to show results of a GET form

2.8.3 Create FormPost.cshmtl to show results of a POST form

2.8.4 Change Index.cshtml to show two forms for student profile

2.8.5 Make changes to HomeController to handle the new views

2.8.6 Build the project

2.8.7 Publish and run the project

2.8.8 Push the code to your GitHub repositories

**Step 2.8.1:** Create an ASP.NET MVC project to show different types of form submits

* 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 **Phase3FormSubmit** 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 2.8.2:** Create FormGet.cshmtl to show results of a GET form

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

@{

ViewData["Title"] = "FormGet";

}

<**h2**>FormGet</**h2**>

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

<**div** class="col-sm-7">

Name @ViewData["name"]<**br** />

Email @ViewData["email"]<**br** />

Class @ViewData["class"]<**br** />

Address @ViewData["address"]<**br** />

</**div**>

</**div**>

**Step 2.8.3:** Create FormPost.cshmtl to show results of a POST form

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

@{

ViewData["Title"] = "FormPost";

}

<**h2**>FormPost</**h2**>

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

<**div** class="col-sm-7">

Name @ViewData["name"]<**br** />

Email @ViewData["email"]<**br** />

Class @ViewData["class"]<**br** />

Address @ViewData["address"]<**br** />

</**div**>

</**div**>

**Step 2.8.4:** Change Index.cshtml to show two forms for student profile

* 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-6">

<**h4**>Form Submit using GET</**h4**>

<**form** method="get" action="Home/FormGet">

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

<**div** class="col-sm-4">Student Name</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="name" /></**div**>

</**div**>

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

<**div** class="col-sm-4">Student Email</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="email" /></**div**>

</**div**>

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

<**div** class="col-sm-4">Student Class</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="class" /></**div**>

</**div**>

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

<**div** class="col-sm-4">Student Address</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="address" /></**div**>

</**div**>

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

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

<**button** class="btn btn-default">Submit</**button**>

</**div**>

</**div**>

</**form**>

</**div**>

<**div** class="col-sm-6">

<**h4**>Form Submit using POST</**h4**>

<**form** method="post" action="Home/FormPost">

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

<**div** class="col-sm-4">Student Name</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="name" /></**div**>

</**div**>

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

<**div** class="col-sm-4">Student Email</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="email" /></**div**>

</**div**>

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

<**div** class="col-sm-4">Student Class</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="whichclass" /></**div**>

</**div**>

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

<**div** class="col-sm-4">Student Address</**div**>

<**div** class="col-sm-8"><**input** class="form-control" name="address" /></**div**>

</**div**>

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

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

<**button** class="btn btn-default">Submit</**button**>

</**div**>

</**div**>

</**form**>

</**div**>

</**div**>

</**div**>"

**Step 2.8.5:** 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** Phase3FormSubmit.Models;

**namespace** Phase3FormSubmit.Controllers

{

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

{

**public** IActionResult Index()

{

**return** View();

}

**public** IActionResult FormGet()

{

ViewData["name"] = Request.Query["name"].ToString();

ViewData["email"] = Request.Query["email"].ToString();

ViewData["class"] = Request.Query["class"].ToString();

ViewData["address"] = Request.Query["address"].ToString();

**return** View();

}

[HttpPost]

**public** IActionResult FormPost(**string** name, **string** address, **string** email, **string** whichclass)

{

ViewData["name"] = name;

ViewData["email"] = email;

ViewData["class"] = whichclass;

ViewData["address"] = address;

**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 });

}

}

}

**Step 2.8.6:** Build the project

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

**Step 2.8.7:** 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](about:blank).

**Step 2.8.8:** 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