2.7 Validations with Data Annotations



This section will guide you to:

* Work with data annotations for validating inputs

This guide has seven subsections, namely:

2.7.1 Create an ASP.NET MVC project to do validation using data annotations

2.7.2 Create a Student Model with data annotations

2.7.3 Change Index.cshtml to show Student Profile form

2.7.4 Make changes to HomeController to do form validation

2.7.5 Build the project

2.7.6 Publish and run the project

2.7.7 Push the code to your GitHub repositories

**Step 2.7.1:** Create an ASP.NET MVC project to do validation using data annotations

* 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 **Phase3Section2.14** 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 2.7.2:** Create a Student Model with data annotations

* In **Solution Explorer**,right click **Models** and choose **Add->Class.**
* Enter **Name** as Student.cs and click **Add.**
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Threading.Tasks;

**using** System.ComponentModel.DataAnnotations;

**namespace** Phase3Section2.\_14.Models

{

**public** **class** Student

{

[Required]

[StringLength(100, ErrorMessage = "Name is required")]

**public** **string** Name { **get**; **set**; }

[Required]

[StringLength(255)]

**public** **string** Address { **get**; **set**; }

[Required]

[StringLength(10)]

**public** **string** Class { **get**; **set**; }

[Required]

[StringLength(100)]

[EmailAddress]

**public** **string** ContactEmail { **get**; **set**; }

[Required]

[Range (5,14)]

**public** **int** Age { **get**; **set**; }

}

}

**Step 2.7.3:** Change Index.cshtml to show Student Profile form

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

@model Student

@{

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

}

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

<**br** />

<**h4**>Student Details Form</**h4**>

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

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

<**div** class="col-sm-4">

Name

</**div**>

<**div** class="col-sm-8">

<**input** class="form-control" name="name" />

</**div**>

</**div**>

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

<**div** class="col-sm-4">

Address

</**div**>

<**div** class="col-sm-8">

<**input** class="form-control" name="address" />

</**div**>

</**div**>

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

<**div** class="col-sm-4">

Class

</**div**>

<**div** class="col-sm-8">

<**input** class="form-control" name="class" />

</**div**>

</**div**>

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

<**div** class="col-sm-4">

Contact Email

</**div**>

<**div** class="col-sm-8">

<**input** class="form-control" name="contactEmail" />

</**div**>

</**div**>

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

<**div** class="col-sm-4">

Age

</**div**>

<**div** class="col-sm-8">

<**input** class="form-control" type="number" name="age" />

</**div**>

</**div**>

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

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

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

</**div**>

</**div**>

</**form**>

</**div**>

**Step 2.7.4:** Make changes to HomeController to do form validation

* 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.Text;

**using** System.Threading.Tasks;

**using** Microsoft.AspNetCore.Mvc;

**using** Microsoft.AspNetCore.Mvc.ModelBinding;

**using** Phase3Section2.\_14.Models;

**namespace** Phase3Section2.\_14.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 });

}

[HttpPost]

**public** IActionResult FormAction(Student student)

{

**if** (ModelState.IsValid)

**return** Content("Form data is valid.");

**else**

{

StringBuilder sb = **new** StringBuilder("");

**foreach** (ModelStateEntry value **in** ViewData.ModelState.Values)

{

**if** (value.Errors.Count > 0)

{

**for** (**int** i = 0; i < value.Errors.Count; i++)

sb.Append(value.Errors[i].ErrorMessage + "\n");

}

}

**return** Content("Form data is invalid with " + ModelState.ErrorCount.ToString() + " errors:\n " + sb.ToString());

}

}

}

}

**Step 2.7.5:** Build the project

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

**Step 2.7.6:** 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 2.7.7:** 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