Academia International College

Affiliated to Tribhuvan University

Lab No. 5

Lab Report on

ASP.NET Core

Submitted by:

Sagar Timalsena

Roll No. 15847

6th semester

Submitted to:

Chandan Bhagat Gupta

Theory

ASP.NET:

- ASP.NET is a fundamental web development platform used to create websites, applications, and web services.
- It works on HTTP (Hypertext Transfer Protocol) and uses the HTTP commands and polices to set a browser to server bilateral communication.
- ASP.NET provides three development style for creating web applications: ASP.NET Web Forms, ASP.NET MVC and ASP.NET Web pages.
- The new version of ASP.NET is ASP.NET Core which was released by Microsoft to make application open-source and cross-platform compatible.

ASP.NET Core:

- ASP.NET Core is a free, open-source, and cross platform framework for building cloud-based applications, such as web apps, IOT apps, and mobile backend.
- It is designed to run on the cloud as well as on-premises.
- It includes the MVC Framework, which now combines the features of MVC and Web API into a single web programming framework.
- ASP.NET Core apps can run on .NET Core or on the full .NET Framework.

Procedure

ASP.NET Core can be used to create both static and dynamic web pages. It is integrated with modern UI Frameworks which makes it easier to use and manage UI frameworks such as AngularJS, ReactJS, Bootstrap, etc. We used Visual Studio 2019 to create and develop an ASP.NET Core web application. Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft which is used to develop computer-programs, as well as websites, web apps, web services and mobile apps.

ASP.NET Core web app development contains following steps:

Creating a Web Application in Visual Studio

1. Start Visual Studio and select "Create a new project". In the Create a new project dialog, select ASP.NET Core Web Application and click Next button.
2. In "Configure your new project" dialog, enter the project name. For e.g. **Lab5**. It is important to name the project with capitalization because it represents the namespace of entire project.
3. Select Create button.
4. In "Create an new ASP.NET Core web application" dialog, select:
    - .NET Core and ASP.NET Core 5.0 in the dropdowns.
    - ASP.NET Core Web App (Model-View-Controller).
    - Create.

The above steps creates an ASP.NET Core MVC web app in Visual Studio which contains different in-built files which helps to configure and run default web pages of ASP.NET Core web app.

Building a Web Application in Visual Studio

1. Add a Controller. Controller is a class that handles browser request, retrieves model data and call view templates that return a response.
    - Right-click on the Controller Folder and select Add > Controller.
    - In the Add Scaffold dialog box, select "MVC Controller – Empty".

- In the Add New Item, enter Controller name and select Add. For e.g. **ProfileController.cs**

2. Create three different action method (i.e. Home, About and Contact) in ProfileController.cs controller file.

```
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace SAGARTIMALLSENA.Controllers
{
    public class ProfileController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
        public IActionResult Home()
        {
            return View();
        }
        public IActionResult About()
        {
            return View();
        }
        public IActionResult Contact()
        {
            return View();
        }
    }
}
```

3. Add a View. View is a component that display that app's UI.
    - Right click on the View folder, and then select Add > New Folder and name the folder Profile.
    - Right click on Views/Profile folder, then Add > New Item.

- In New Item dialog, select Razor View – Empty and name the file as Home.cshtml and select Add.
- Repeat same process for About.cshtml and Contact.cshtml.

4. Update Home, About and Contact pages.

For Home.cshtml

```
<div class="box">
    <div class="head">
        <h2>ASP.NET MVC</h2>
    </div>
    <div class="body">
        <p>
            <b>Model View Controller (MVC)</b>
            MVC is a design pattern used to decouple user-interface (view),
            data (model), and application logic (controller).
            This pattern helps to achieve separation of concerns. <br>

            Using the MVC pattern for websites, requests are routed to
            a Controller that is responsible for working with the Model to
            perform actions and/or retrieve data. The Controller chooses the
            View to display and provides it with the Model. The View renders the
            final page, based on the data in the Model.
        </p>
        <p>
            <h4><strong> Models & Data </strong></h4>
            Create clean model classes and easily bind them to your database.
            Declaratively define validation rules, using C# attributes, which are
            applied on the client and server.<br>

            ASP.NET supports many database engines including SQLite,
            SQL Server, MySQL, PostgreSQL, DB2 and more, as well as
            non-relational stores such as MongoDB, Redis, and Azure Cosmos DB.
        </p>
        <p>
            <h4><strong>Controllers</strong></h4>
            Simply route requests to controller actions, implemented as normal
            C# methods. Data from the request path, query string, and request body
            are automatically bound to method parameters.
        </p>
        <p>
            <h4><strong> Views with Razor </strong></h4>
            The Razor syntax provides a simple, clean, and lightweight way to render HTML
            content based on your view. Razor lets you render a page using C#, producing
            fully HTML5 compliant web pages.
        </p>
    </div>
</div>
```

For About.cshtml

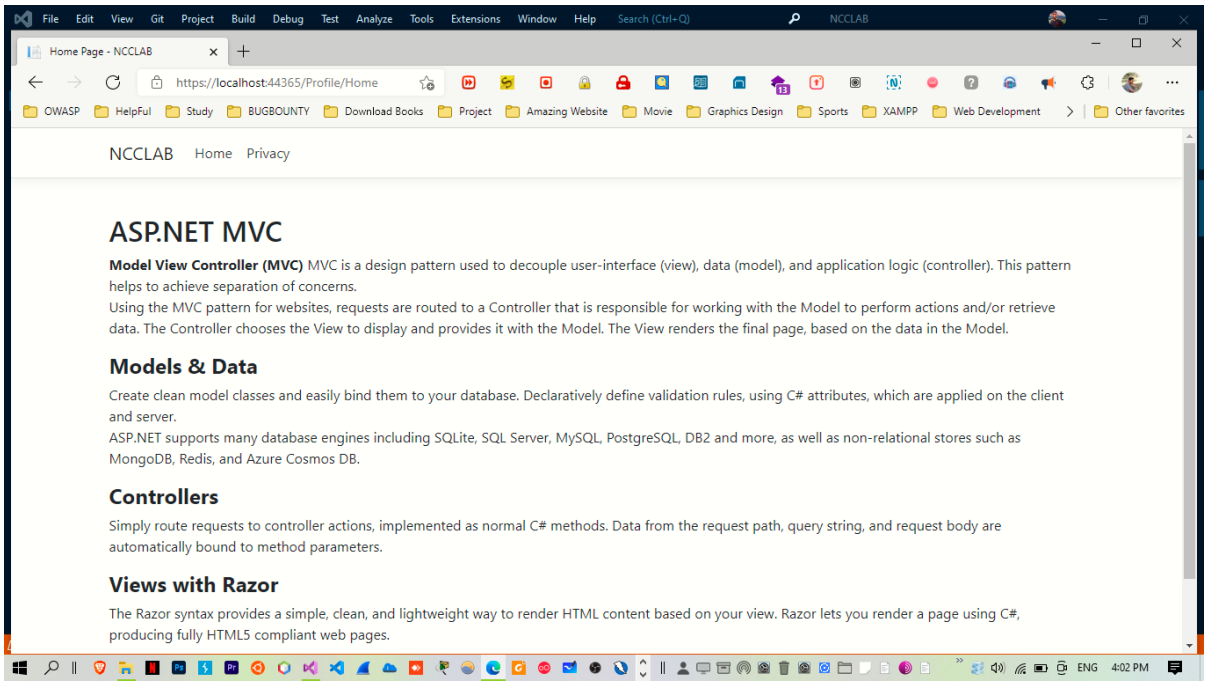
```
@*
    For more information on enabling MVC for empty projects,
    visit https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
}
<div class="text-center">
    <h1 class="display-4">About Me</h1>
    <p>
        I'm an aspiring learner of ethical hacking and technology and try to
        share whatever I learn from the Internet. I am currently a student of
        Computer Science at Academia International College.
        I am a newbie to the world of technology but try to learn anything
        and everything
    </p>


</div>
```

For Contact.cshtml

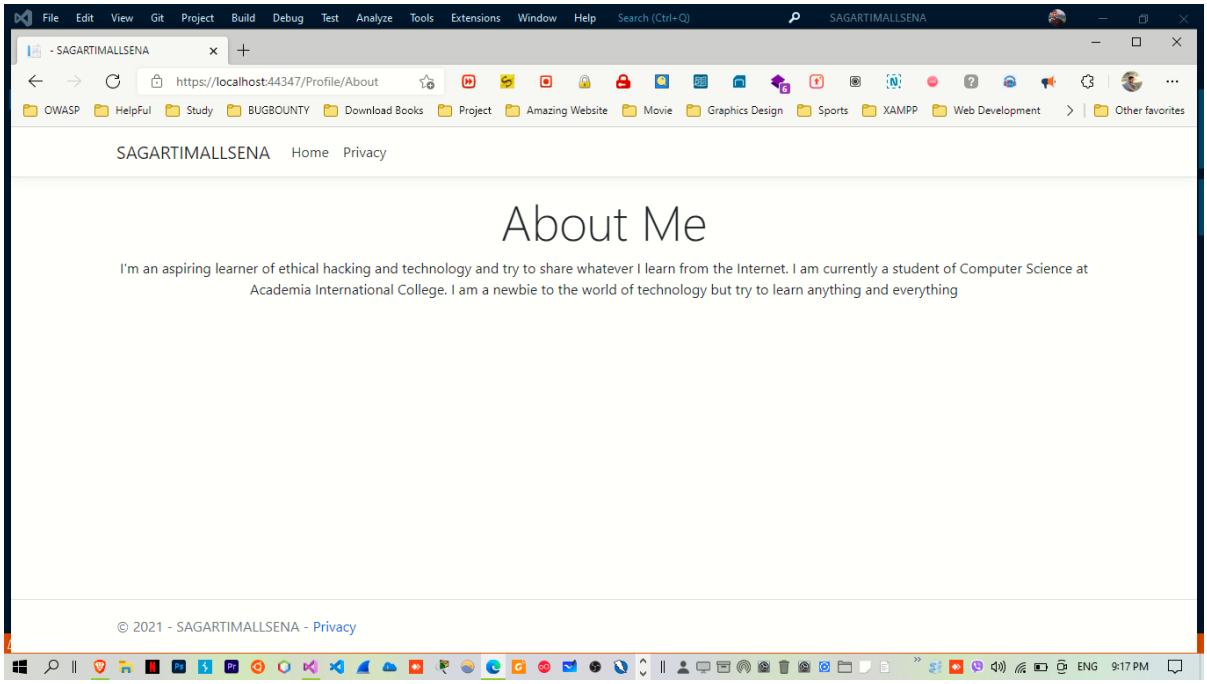
```
@*

*@
@{
}
<div class="text-center">
    <h1 class="display-4">Contact Page </h1>
        <p>
            Contact Page Not Available.<br />
            For More Detail click <a href="www.timalsensagar.com.np"> Here</a>
        </p>
</div>
```

Output

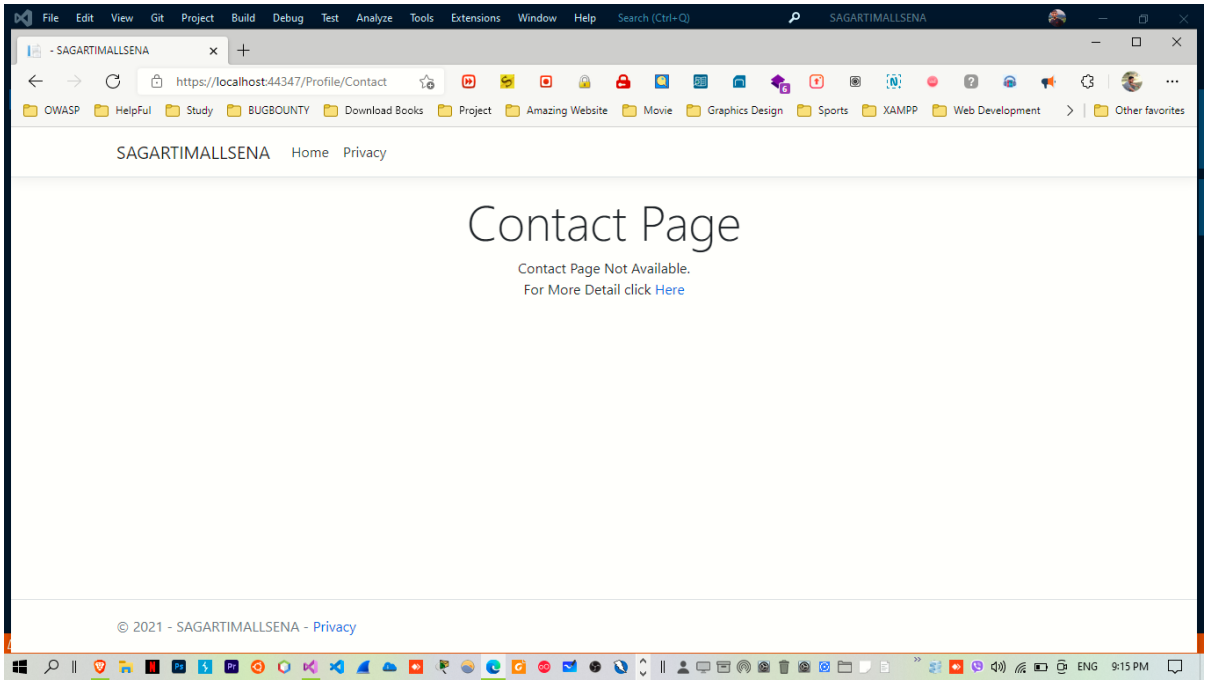
1. Home Page



2. About Page



3. Contact Page



Conclusion

Hence, we implemented ASP.NET Core Framework to create and run web application which contains different web pages like Home, About and Contact Page.

GitHub Repository

All the above codes used in this report are uploaded in GitHub Repository https://github.com/Sagar1555/ncclab