COM741 Web Applications Development

Practical 4 (Introduction to MVC) - Solutions

1. Create the default template mvc project as outlined.

2. The updated view file `SMS.Web\Views\Home\index.cshtml` should look as follows

   ```
   <div class="text-center">
     <h1>Home Page</h1>
     <p>This is the default page displayed when the application loads</
   </div>
   ```

3. The updated view file `SMS.Web\Views\Home\index.cshtml` should now look as follows

   ```
   <div class="text-center">
     <h1>Home Page</h1>
     <p>This is the default page displayed when the application loads</
     <p>Today is <b>@DateTime.Now.ToLongDateString()</b></p>
   </div>
   ```

4. The SMS.Web\Controllers\HomeController.cs file should contain an `Index()` action method as follows

   ```
   public IActionResult Index()
   {
       ViewBag.LongTime = DateTime.Now.ToLongTimeString();
       ViewBag.Message = "Time Now";
       return View();
   }
   ```

   Then the view file file `SMS.Web\Views\Home\index.cshtml` should now look as follows

```
  <div class="text-center">
    <h1>Home Page</h1>
    <p>This is the default page displayed when the application loads</
    <p>Today is <b>@DateTime.Now.ToLongDateString()</b></p>
</div>

<div class="text-center">
    @ViewBag.Message <b>@ViewBag.LongTime</b>
</div>
```

5. Ensure you added the AboutViewModel.cs file contained in the question to the Models folder.

```
using System;

namespace SMS.Web.Models
{
    public class AboutViewModel
    {
        public string Title { get; set; }
        public string Message { get; set; }
        public DateTime Formed { get; set; } = DateTime.Now;
        public string FormedString => Formed.ToLongDateString();
        public int Days => (DateTime.Now - Formed).Days;
    }
}
```

Next, the SMS.Web\Controllers\HomeController file should have following action added, where we create an instance of our viewmodel and set its properties. We then return the View and pass the viewmodel as a parameter.

```
public IActionResult About()
{
    var about = new AboutViewModel {
        Title = "About",
        Message = "Our mission is to develop great solutions for educ
        Formed = new DateTime(2000,10,1)
    };
    return View(about);
}
```

Next, we create a view file SMS.Web\Views\Home\About.cshtml as follows

```
@model AboutViewModel

<h2>@Model.Title</h2>

<p>@Model.Message</p>
<p>We were formed on @Model.FormedString</p>
<p>And are now operating for @Model.Days days</p>
```

Edit the SMS.Web\Views\Shared_Layout.cshtml file and add a new Bootstrap nav link so that a menu option is created to navigate to the about page. See partial nav below with additional `li` at end containing About anchor link

```
<nav>
    ...
    <ul class="navbar-nav flex-grow-1">
        <li class="nav-item">
            <a class="nav-link text-dark" asp-controller="Home" asp-a
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark" asp-controller="Home" asp-a
        </li>

        <li class="nav-item">
            <a class="nav-link text-dark" asp-controller="Home" asp-a
        </li>

    </ul>
    ...
</nav>
```

6. Using a bootstrap card style to display about view

```
<div class="card text-center shadow">
    <div class="card-header bg-warning">
        <h2>@Model.Title</h2>
    </div>
    <div class="card-body">
        <div class="card-body">
            <h5 class="card-title">Mission Statement</h5>
            <p><em>@Model.Message</em></p>
        </div>
        <div class="card-footer text-muted">
            <div>Formed on @Model.FormedString <cite title="Source T:
        </div>
    </div>
</div>
```

To style the Index.cshtml file to use our new custom css class it should look as follows

```
    <div class="callout callout-info">
        <h1>Home Page</h1>
        <p>This is the default page displayed when the application lo
        <p class="text-center"> Today is <b> @DateTime.Now.ToLongDate
    </div>

    <div class="callout callout-warning text-center">
        @ViewBag.Message <b>@ViewBag.Time</b>
    </div>
```

The SMS.Web\wwwroot\css\site.css file should have following custom css styles added to end of the file. Note the additional callout-info callout-warning etc are optional.

```css
/* ---------- Custom Styles --------- */
.callout {
  padding: 1.25rem;
  margin-top: 1.25rem;
  margin-bottom: 1.25rem;
  border: 1px solid lightgray;
  border-left-color: silver;
  border-left-width: .75rem;
  border-radius: .25rem;
}

.callout-primary {
  border-left-color: cornflowerblue;
}

.callout-info {
  border-left-color: deepskyblue;
}

.callout-warning {
  border-left-color: orange;
}

.callout-danger {
  border-left-color: red;
}

.callout-secondary {
  border-left-color: silver;
}
```