**Assignment #2: Getting Your Site Started**

**Creating the Correct Project Type**

If you created your project using the command "dotnet new web" on the previous assignment, you can skip this step. If you used a different type, such as console or class library, run the command "dotnet new web --force" in order to overwrite / create the correct project type.

**Setting Up MVC**

Your basic web project does little more than display "Hello World" to the user if you run it and browse to the site. This is not very useful.

**Modify Startup.cs** - Refer to pages 386 and 387 in the book for the changes you will need to make in order for your site to start using MVC.

In addition, add the following to the Configure function:

app.UseStaticFiles()

**Create the MVC Folders** - Create the following three folders at the root of your site "Models", "Views", "Controllers"

**Create some Controllers**

HomeController – this will be used to serve your sites home page. You will need to create and associated Index.cshtml view to go with. You can put some dummy content on this page.

AboutController – this will be used to serve information about your project. You can add dummy content on this page for now, since your project has not yet been finalized.

ContactController – this will be used for a contact form. More details on this later in the assignment writeup.

**Creating Your Layout**

Add a new layout file named \_DefaultLayout.cshtml to the ~/Views/Shared folder

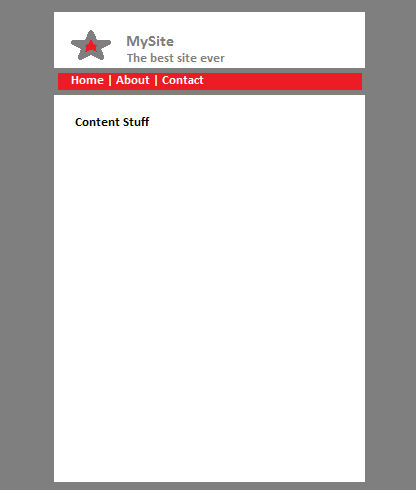
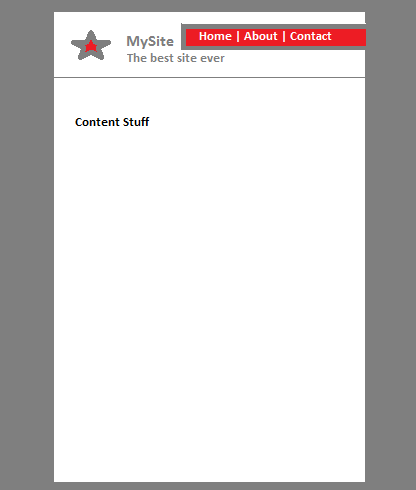
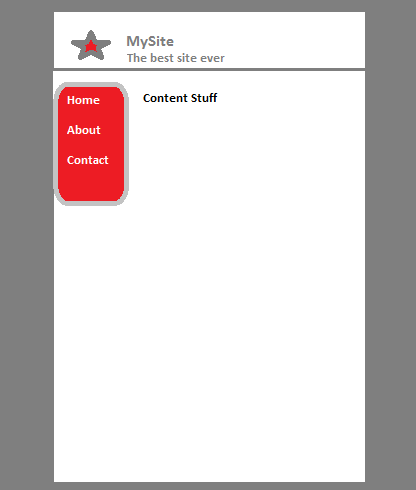
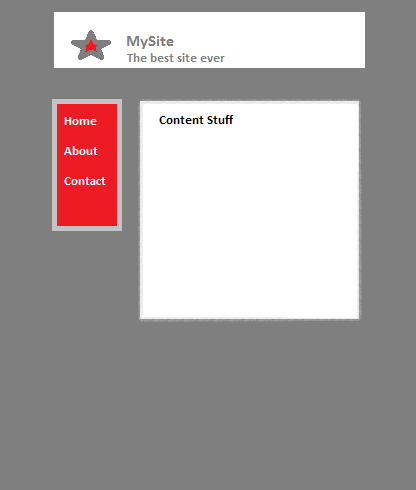
**Make Everything Use the Layout File**

Create the \_ViewStart.cshtml file in ~/Views and set the Layout to be your \_DefaultLayout file

**Building the Layout**

Include Bootstrap in your template. Decide on how you want your navigation laid out. You will want to incorporate the grid system here. Also, use the Bootstrap Navbar component in order to style your nav menu. There are multiple ways to use it. Here are the docs: https://getbootstrap.com/docs/4.0/components/navbar/

I also expect the site to be responsive, meaning that if viewed on a small screen, the main navigation will adjust so that it doesn’t wrap strangely. Here are some example layouts, for ideas, but you are welcome to come up with your own

Your template should contain links to the Home, About and Contact pages. Use the @Html.ActionLink HTML Helper to generate these links.

**Picking a color palette for your site**

Visit http://paletton.com and use it to find a color palette for your site. Once you are done playing around and picking your color palette, click the Share palette link in the upper right. This will provide you with an identifier that you can use to return to the picker should you forget the colors or need to tweak them. Don’t forget to write it down! Use the Tables/Export tab in the bottom right to view the color codes to use.

If you haven’t added a CSS file to your project yet, create a css folder within the wwwroot folder. Then, create a file called DefaultStyles.css and add it to the css folder.

In your CSS file, use at least 3 of the colors from your palette to color things differently on your site. You will want to use the background-color for changing box colors or the color property for changing text color. You can use the examples from Paletton to determine which elements you may want to style.

ONLY IMPLEMENT THE STYLING IN YOUR CSS FILE. Applying classes or IDs to elements of your page will be the best way to accomplish this.

**Contact Form**

**The Model**

Create a model with the following properties. All of the properties should be required.

* Name
* Phone Number
* Email
* Message

**The View**Create a form on the view which has fields for all of the properties of the model

**The Controller**

The controller should have an Index method which displays the form when the users views it. It should have another overload of Index which accepts a POST from the form and takes your model as a parameter. When the form is posted, it will return a view that displays all of the information that was entered to the user (we will deal with sending emails later on).

When you are done, check-in/sync your work to your repository. I will pull the commit from that point in time from your repository. And don’t forget to run your pages through the HTML validator at http://validator.w3.org!