

MyRoster

// Add a new controller to the Experiment project created on Worksheet 1A

- ☐ Update the **DotNetCompilerPlatform** nuget package to the latest version
- ☐ Create a class named **RosterItem** under the **Models** folder
- ☐ Create a class named **Roster** under the **Models** folder
- ☐ The **RosterItem** class should have the following properties:
 - ☐ **FirstName : string**
 - ☐ **LastName : string**
 - ☐ **Email : string**
- ☐ The **Roster** class should have the following properties:
 - ☐ **Items : List<RosterItem>**
- ☐ Create a **RosterController**
- ☐ Build a page to list all the people currently on the roster (call it Index)
- ☐ Build a page to add a person to the roster (call it Add)
- ☐ Use Bootstrap 4 to style the pages

Experiments with Layout

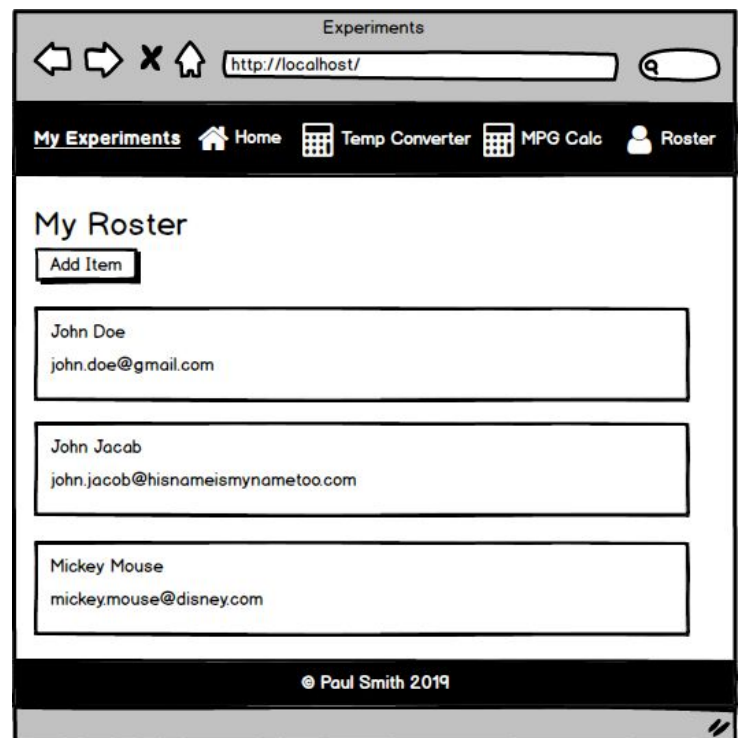
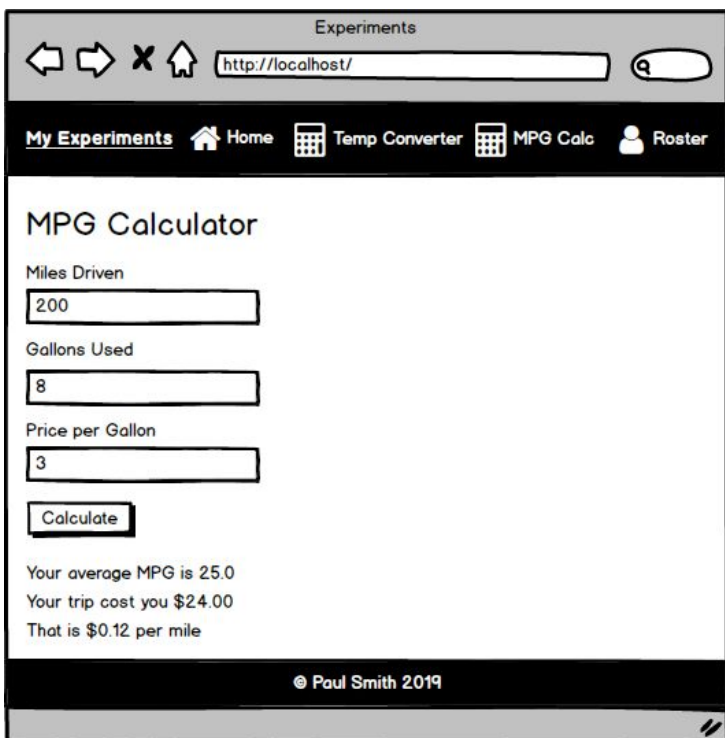
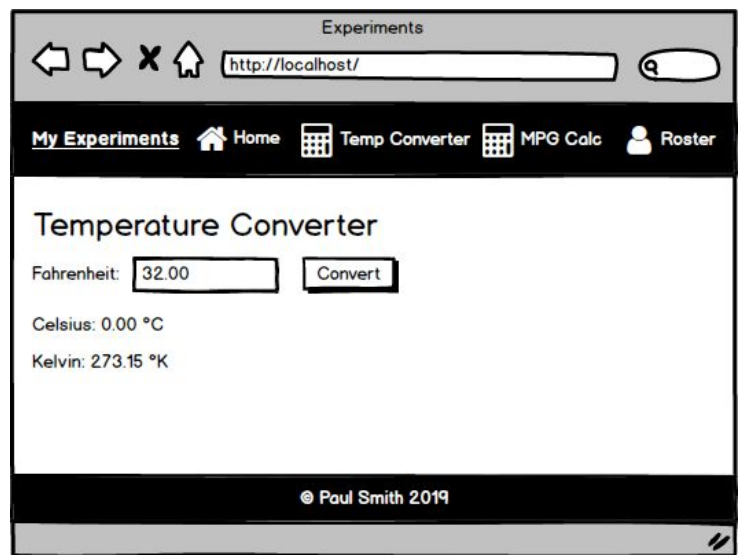
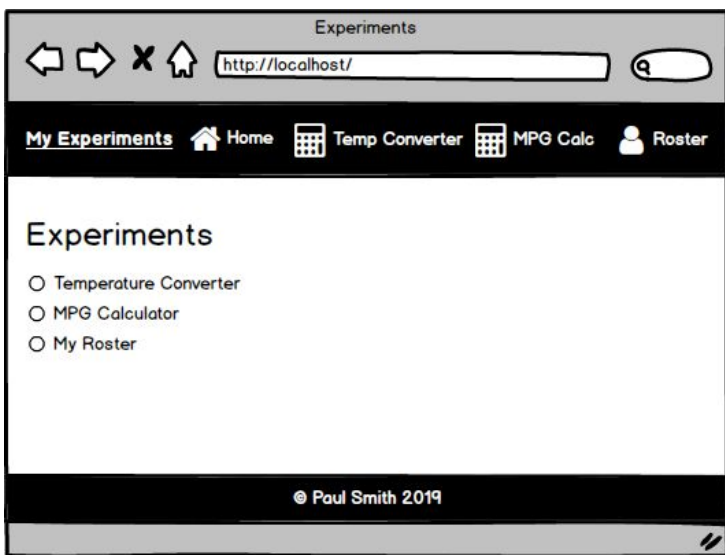
Update your existing **Experiments** project to use a layout file.

- **You must combine your TempConvert, MpgCalc, and Roster labs into one project, if you did not do so already.**
- Create a **_ViewStart.cshtml** file
 - Create it directly under the Views folder (not a subfolder)
 - Remove any previously existing content and replace it with the following code

```
@{ Layout = "~/Views/_Layout.cshtml" }
```
- Create a **_Layout.cshtml** file
 - Create it directly under the Views folder (not a subfolder)
 - Allow views to set the title of the page like so:

```
<title>@ViewBag.Title</title>
```
 - Add the Bootstrap 4.3.1 stylesheet to the head
 - Add the Bootstrap 4.3.1 scripts to the bottom of the body
 - Replace the jQuery script with the minified version, instead of the slim version [found online at <https://code.jquery.com>]
 - Add a navbar with links to: Home, Temp Converter, MPG Calc, Roster

- Add a main tag to hold the content of the views
<main>@RenderBody()</main>
- Add a footer with copyright info like so © **Paul Smith 2019** (use your name instead)
- Refactor all of the views in the project to use the default layout created above
 - Remove the line of code that sets the Layout
 - Add a line of code to set ViewBag.Title
 - Remove all the redundant HTML, leaving just the contents of the main tag
- **Review screen mockups below for guidance**



Portfolio

// Improve your previously created portfolio under labs/UNIT_01_LAB/Portfolio

Improve the portfolio that you created for Lab 01A. For this assignment you will add a gallery of past projects to your portfolio.

- ☐ Update the **DotNetCompilerPlatform** nuget package to the latest version
- ☐ Create a class named **Project** under the **Models** folder
- ☐ The **Project** class should have the following properties:
 - ☐ **Name : string**
 - ☐ **Description : string**
 - ☐ **Photos : string[]**
 - ☐ **Tags : string[]**
- ☐ Create a **List<Project>** of three of your past projects in your **HomeController** using both a collection initializer and object initializers (see chapter 4)
- ☐ In **HomeController.Index()** use **ViewBag** to pass the list of projects to the view
- ☐ Add three **screenshots / images** per project to the **Content/images** folder
(this should be a total of 9 images)
- ☐ Add the filenames for the images to the **Photos** arrays
- ☐ Add a **Projects** section between the **Contact Me** section and the **footer** for your list of projects
- ☐ Add a link in the navbar to the **Projects** section
- ☐ Use a FontAwesome 5 icon in both the navbar and section header
- ☐ Display each of your three projects in a format similar to the one shown below
 - ☐ Use Bootstrap 4 to style each project as a **card**
 - ☐ Use Bootstrap 4 to style each tag as a **badge**
 - ☐ Use Bootstrap 4 to put the photos into a **carousel**
- ☐ Display the list of projects responsively using the Bootstrap Grid System
 - ☐ Show 3 projects per row on extra-large screens
 - ☐ Show 2 projects per row on medium screens
 - ☐ Show 1 project per row on mobile devices

