Jonathan Tweedle

2019/07/24

CSHP 230 B Su 19: Web Applications In C#

Assignment 05

Styling with jQuery and Bootstrap

# Introduction

For this assignment, I am taking what I built in the previous assignment and enhance it using the Bootstrap and jQuery frameworks. Bootstrap is a great library for styling web sites and jQuery helps with some common features in JavaScript to manipulate the DOM elements of an HTML page

# Getting Started

While there are many ways to introduce both BootStrap and jQuery in your project that range from simply referencing the files on other CDN’s, I chose the simpler method by including them using the NPM package manager. Once the install has finished, two new folders appear in the solution tree (Figure 1). CSS3 styling related files are located under the “Content” folder while the JavaScript files are located under the aptly named “Scripts” folder.

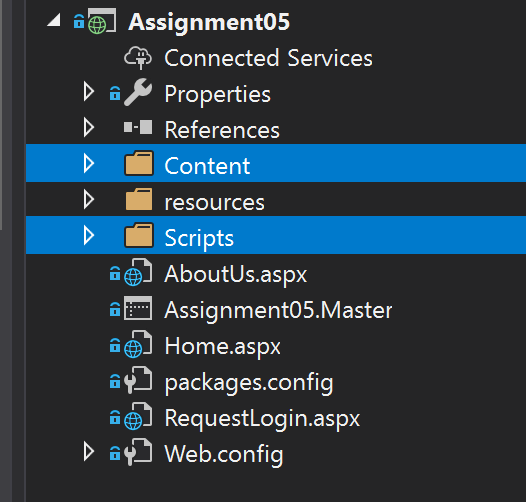


Figure 1: Content and Scripts folders

# Better Navigation

The current navigation had a very basic layout (Figure 2) that I managed to style with some basic CSS3 styling in the master page (Figure 3). This provided some basic navigation buttons (Figure 5) but bootstrap has some great styling for building navigation menus.

To start I first need to strip out the original CSS3 styling (Figure 3) because I will no longer need it and replace it with references to the BootStrap CSS3 style library and supporting JavaScript libraries (Figure 4) in the master file.

A quick test run shows a very bland navigation menu but the page formatting has already changed slightly in its styling (Figure 6).

I then make some changes to the navigation menu design (Figure 7) to define the container for the menu system, some branding that will always show and some additional features that help with some reactive design depending on the size of the screen used to view the page.

The end result is a more professional navigation bar that spans the top of my website pages (Figure 8).

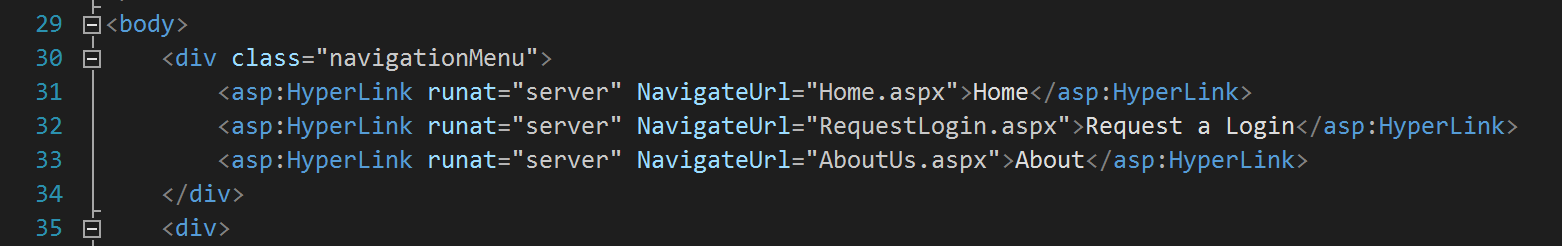


Figure 2: Original navigation design

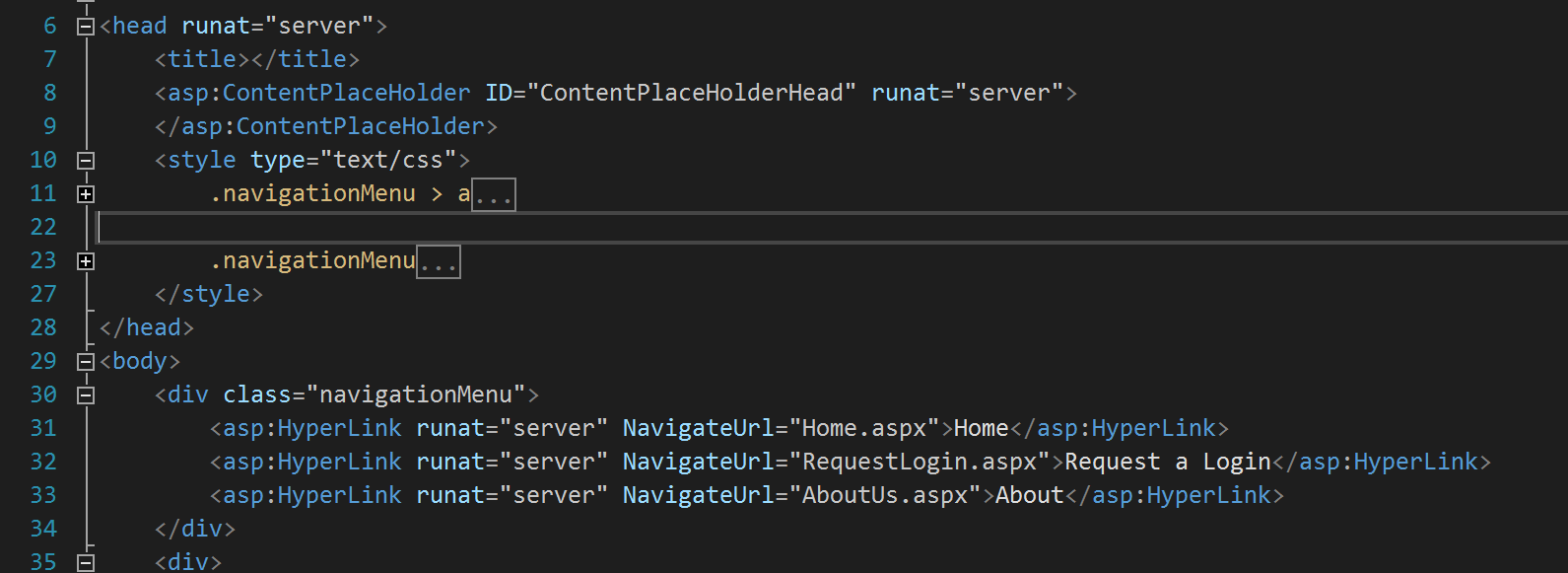


Figure 3: CSS3 styling and Navigation design together

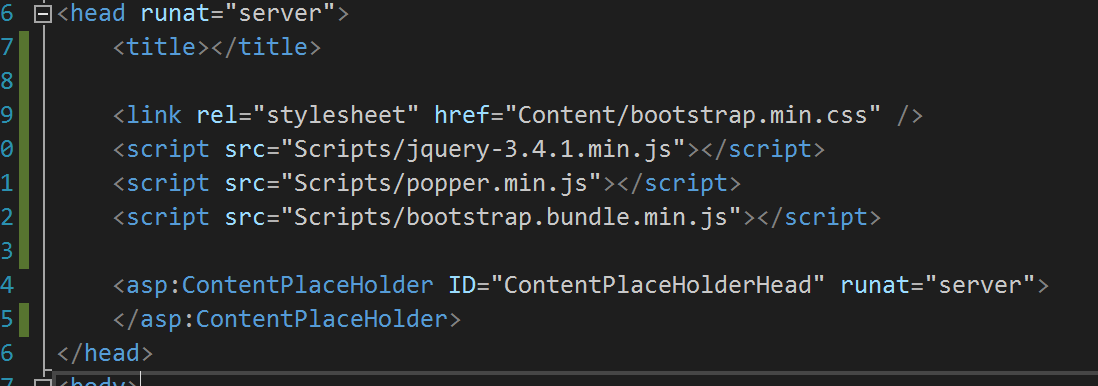


Figure 4: Bootstrap and supporting JavaScript references



Figure 5: Original Navigation design

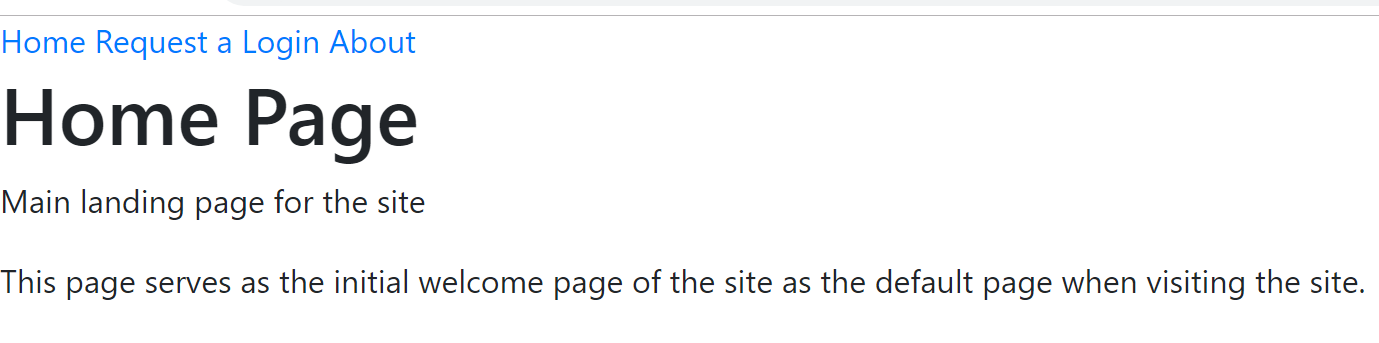


Figure 6: Navigation after removing the original styling and adding BootStrap

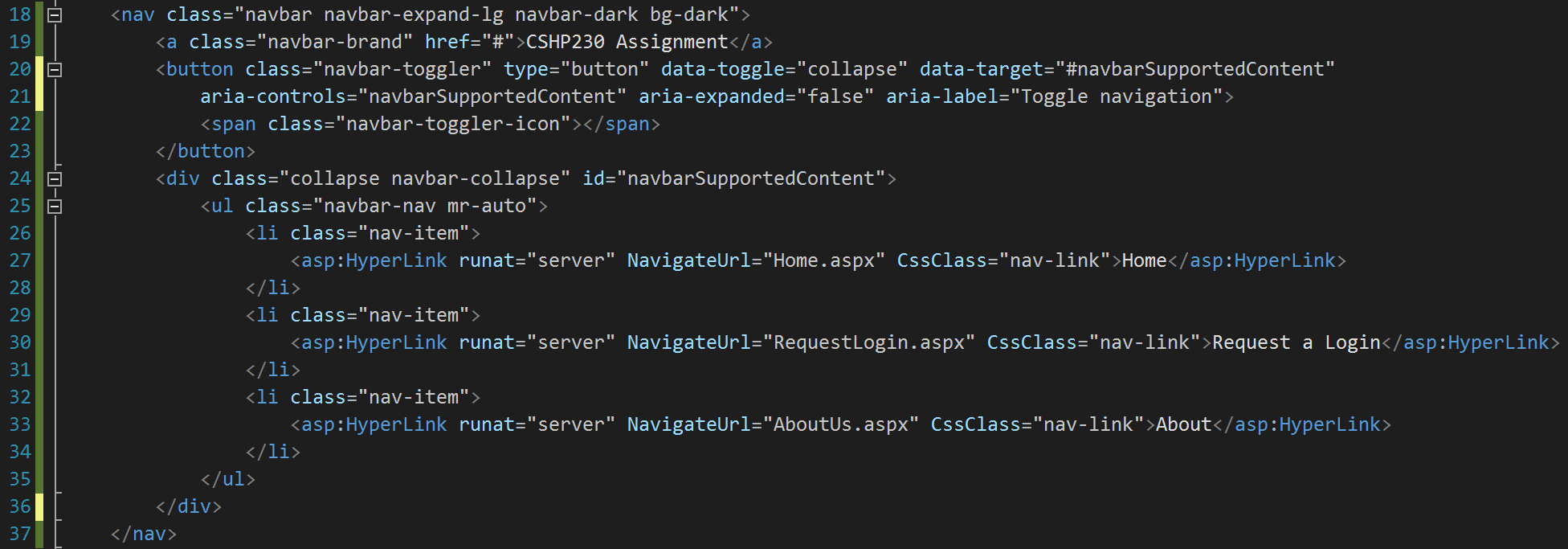


Figure 7: Changes in the navigation design

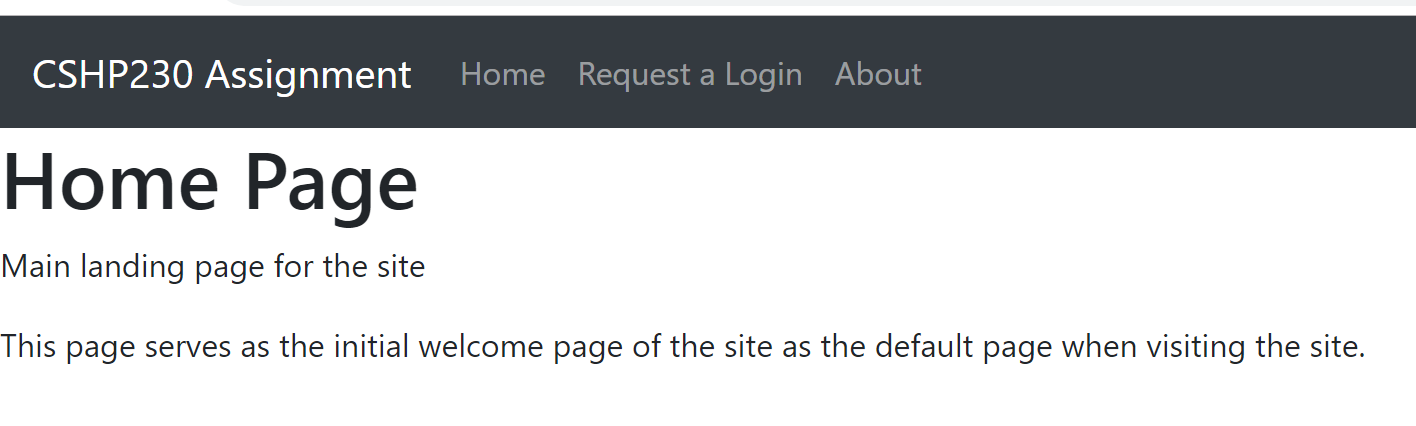


Figure 8: New navigation look

# Cleaner Form Design

My previous assignment relied on a bunch of CSS3 styling (Figure 9) to make the request login form look well formatted (Figure 10). Bootstrap has a bunch of styling to make professional looking forms that also react to the size of the device screen for a smoother experience across multiple devices.

So I took out my original CSS3 styling and reworked the underlying HTML design in order to benefit from the Bootstrap formatting. While my original design was close, it ultimately required redesigning the page.

The result is a nice vertical layout that renders well across multiple devices (Figure 11).

|  |  |
| --- | --- |
| Figure 9: Original CSS3 styling for the form | Figure 10: Original form layout |

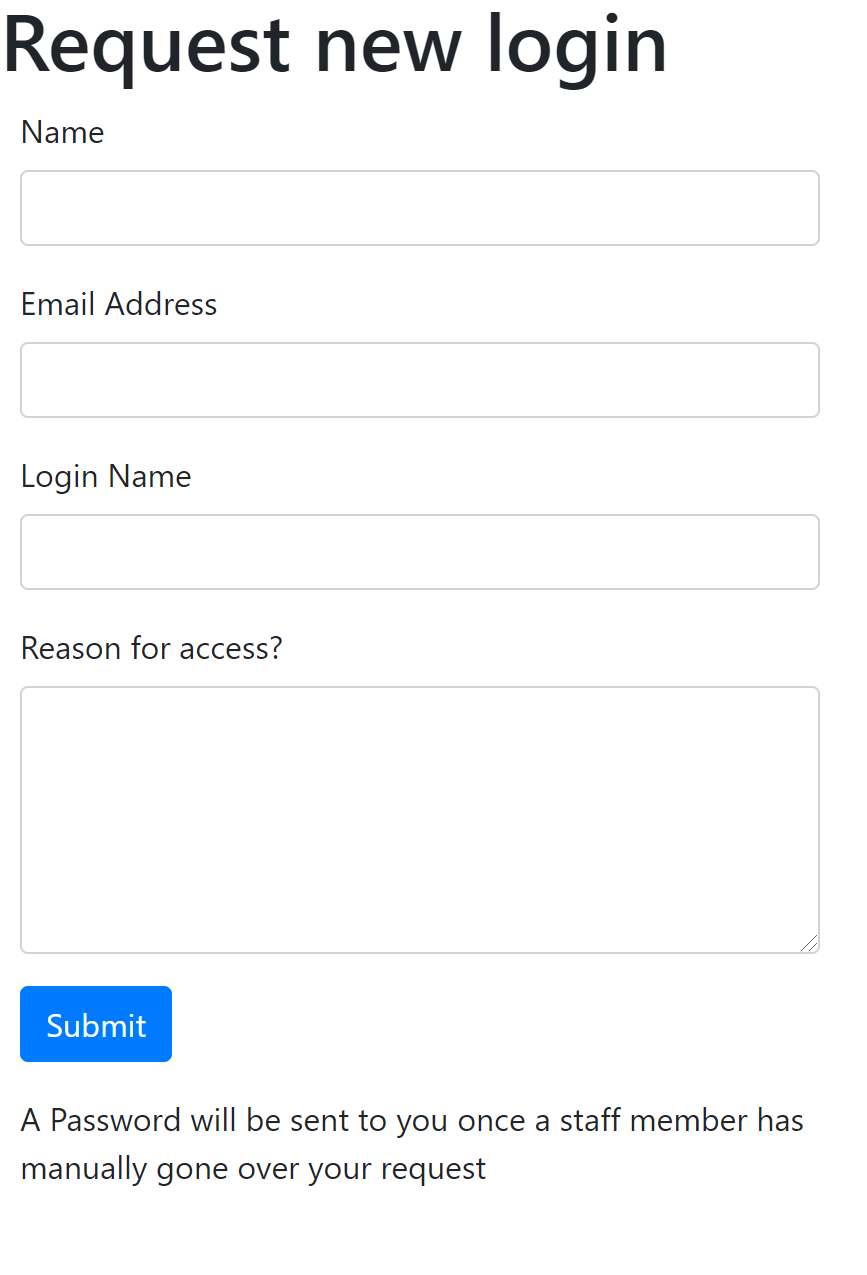


Figure 11: New request login page design

# Client-Side form validation

Without some kind of client-side control, the form can be submitted without any values filled in. To prevent this behaviour, I used jQuery style JavaScript (Figure 12) to monitor the input field values.

If one of the fields is empty then the submit button will be disabled (Figure 13), preventing the submission of the form. Once all the input fields have values in them, the submit button is enabled (Figure 14) allowing the form to be submitted.

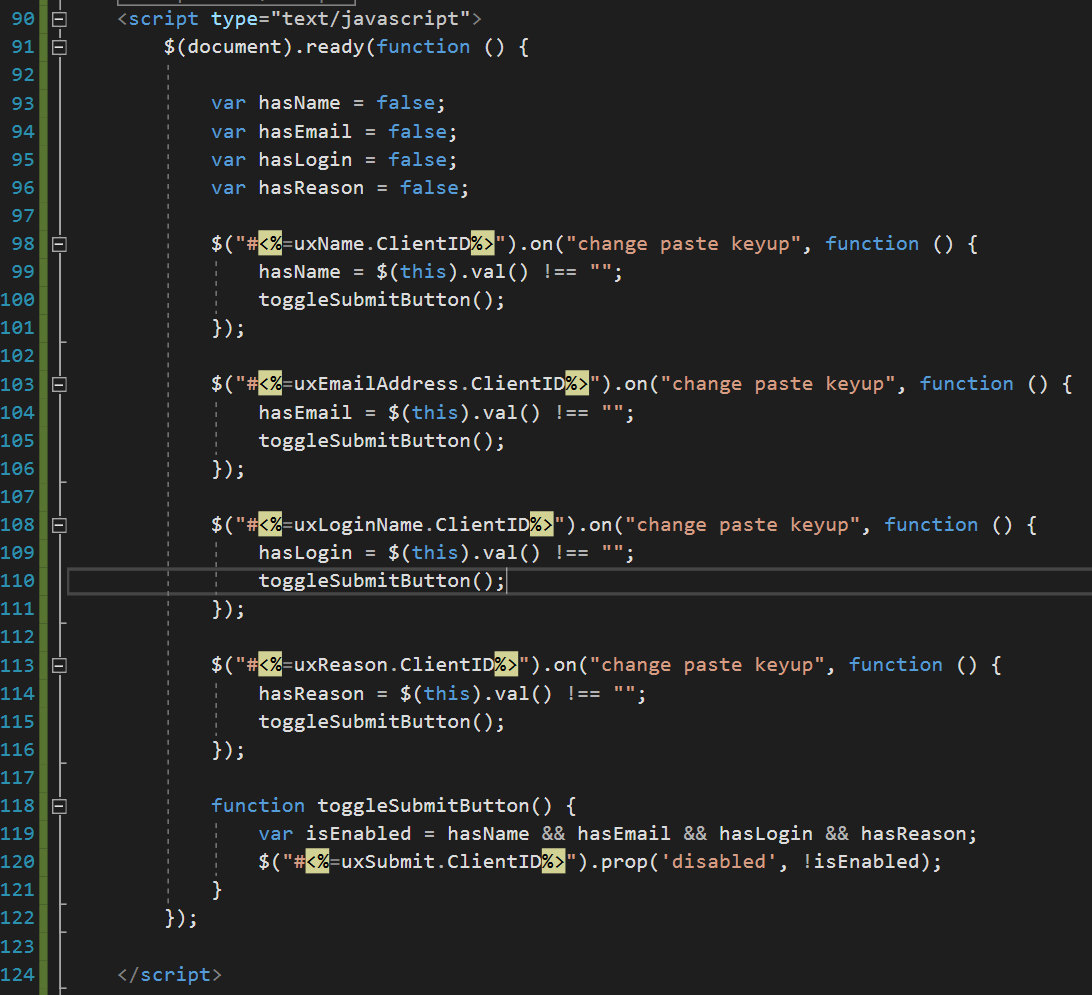


Figure 12: Client scripting using jQuery for validation control

|  |  |
| --- | --- |
| Figure 13: Disabled submit button | Figure 14: Enabled submit button |

# Summary

While it is possible to style your sites and you could write some code to inspect the html DOM to manage the form validation, this can be simplified by taking advantage of libraries that abstracted such common challenges. This leads to a more consistent design. It also makes it easier for other developers to support the design in situations such as shared teams.