

WEB/DEVICE DEVELOPMENT – CLIENT SIDE MIS/CIT 310

Project #4

Updating your class project to be more mobile friendly Build 'include' files to reduce repetitive code

To gain a fuller appreciate for Responsive Design and bootstrap, please review Pages 672 to 77. Another great resource is: <https://www.w3schools.com/bootstrap/>

Overall responsive design is your html page 'responding' to the size of the device that is displaying the page and changing the order/width of the items to be best viewed on that page. Thus, a page may look good on a monitor, tablet or smart device.

Keys to responsive design are style sheets as well as a new term 'media queries'. Think of a media query is a question to your browser that checks the size of the screen that revises the size/appearance/order of items based on the size of the screen

"Bootstrap" is an enhancement on the basic responsive design concept as it is a series of jQuery libraries and style pages that make your job even easier, you would link to the bootstrap.js and .css files like we have done in prior projects. Bootstrap was originally created by Twitter to work easier on mobile devices

A. **Check Entropy for any comments** that would correct/enhance your existing web site. Please correct any missing items, and consider adding any suggestions made in Entropy as it refers to your site.

B. Adding bootstrap to your current project

Overall, the concepts are the same as what we have done in the past, build links to the style sheets for the bootstrap styles, build links to the javascript files, and then build the proper tags <....> that use the classes and styles in the .css files.

- 1) First, make a copy of your entire Project 3 folder (in case you need to go back and restart)
- 2) Before opening Visual Studio, let's add some files to your project that will support Bootstrap/Responsive Design
 - a. Go to [\\miscapstone\mis310](https://www.miscapstone.com/mis310) and copy the contents of the A-BootstrapFiles to your project folder

- b. Open Visual Studio, and you should see the new folders/files in your project
- 3) Implement the new styles and scripts by sliding the following css and js files to your project HEAD segment (BEFORE your styles) into your default file:
 - a. Bootstrap.css (from styles folder)
 - b. Site.css (from styles folder)
- 4) Implement the scripts needed for bootstrap, find your `</BODY>` tag (near end) and slide these two SCRIPTS into this area
 - a. Jquery-3.3.1.min.js
 - b. Bootstrap.min.js
- 5) After the `<body>` tag, copy the contents of NewNav.txt into your default.html file
- 6) Save your work, and bring up your default.html page, you should notice a black bar at the top which will become your new navigation menu. Also if you 'shrink' the window, you will notice the links will become a drop down menu as the screen gets smaller.

C. Modifying the NAVIGATION (Black bar) to be your menu,

1. Slowly replace your current `<nav>` section into the new menu.
2. Replace Link Button 1, with HOME and the path should be do your default.html or index.html or whatever the name of your home page is
3. Replace Link Button 2 with the name of your second page and the href to that page
4. Do Likewise for Link Button 3 (or more if you had additional)
5. If you don't like the black background, you could go to the bootstrap.css and search for navbar-inverse style and change the background color to another choice (do at the end if you desire).
6. Save and test and you should have a menu of your web site, and the links in the bar should take you to your other pages
7. Notice as you reduce the height of your window the nav bar always stays at the top (very useful on phones)

D. Other modifications to make your site more responsive.

1. Check your main.css, and if you have a body tag with a width:980px in it, remove the 980px; and the margin-left / margin-right tags, as bootstrap will center and modify the width of your page based on the size of the screen
2. Modify your <body> tag to include the bootstrap style as <body class="container">
3. Save your work and refresh in the browser and your page should be centered and change its width as your shrink the window to represent a tablet or phone.

E. Update pages 2 and 3.

1. Make the following change to your 2nd and 3rd pages (you do not need to update you feature pages unless you desire.
2. NOTE: do not update the nav at this point on pages 2/3, we will do later
3. Implement the new styles and scripts by sliding the following css and js files to your project HEAD segment (BEFORE your styles) into your other pages file:
 - a. Bootstrap.css (from styles folder)
 - b. Site.css (from styles folder)
4. Implement the scripts needed for bootstrap, find your </BODY> tag (near end) and slide these two SCRIPTS into this area
 - a. JQuery-3.3.1.min.js
 - b. Bootstrap.min.js
5. Modify your <body> tag to include the bootstrap style as <body class="container">
6. Save your work and refresh in the browser and your page should be centered and change its width as your shrink the window to represent a tablet or phone.
7. All your main pages should be using the bootstrap.

F. Demonstrate one new bootstrap feature

1. This page has a ton of bootstrap features, <https://www.w3schools.com/bootstrap4/default.asp>
2. Add one new page to demonstrate a new feature, make sure you put on the page the name of the feature you are demonstrating. (the easier ones are at the bottom of the left menu.

3. Don't forget you will need to slide the styles/scripts to this page
4. Add this page to your footer

G. Build INCLUDE FILES to reduce repetitive code

This feature will only run if you execute the page from the miscapstone server as it requires a web server to 'include' the file. This is known as server side includes.

*****Please note that the server side includes only work if you are using an HTTP call to the server and will not work in design/or locally on your desktop**

Two typical uses of server side include:

- 1) If you have items that are repeated on many pages, it is easier to build a server side include and have pages needing that HTML to 'include' it on that pages. The most comment examples are:
 - a. Footers
 - b. Navigation Menus
 - c. Headers on pages
- 2) If you have items that change frequently (a list of items for a calendar) you could pull the calendar items from a database and have a small program create the 'snippet' of code for the calendar items.
- 3) Build an includes folder in your site
- 4) Add a web page to the includes folder (right click), name this new web page: **footer.html**
- 5) Delete any HTML tags in this new footer.html, it should now be a blank page.
- 6) Copy your <footer> to..... </footer> code (including the <footer> tags) to this new page in the includes folder, and save.
- 7) Back on your default.html page, replace the <footer> </footer> with the following:
`<!--#include file="includes/footer.html" -->`
- 8) Save
- 9) To test, move your project to your miscapstone/mis310 folder
- 10) To see from the server, do the following, start with this path:

<http://miscapstone.uncw.edu/mis310/S19xxxxx/>

To that path you will need to add any subfolder and finally your default.html page

As in

<http://miscapstone.uncw.edu/mis310/S19xxx/xxxProject4/Project4/Project4/default.html>

- 11) Do the same includes for pages 2 and 3 of your site
- 12) Now add a navigation.html page to your includes area
- 13) Copy the `<div class=navbar navbar.....>` to the third `</div>` to this navigation.html page
- 14) Replace the code above on your default page, with an include statement.
- 15) Save and test on the sever
- 16) Do the same for the other two pages
- 17) Save and Test
- 18) DONE!