**CS-321: Server-Side Web Development**
**Spring 2021**
**Northeastern Illinois University**
**Homework #6: Bootstrap**
**Due Date: Thursday, 03/04/21 by 6:00 p.m. CDT**

**Homework Goals:**

1. Make all your HTML pages mobile responsive using Bootstrap.
2. Have consistent navigation from page to page.
3. Note that every new HTML page from this point on must be mobile-responsive and styled appropriately.
4. Push your code changes to GitHub using GitHub Desktop.

---

**Part I:** Setting up Bootstrap

- Include Bootstrap in your project by incorporating the correct dependency in the pom.xml file.
- Include the Bootstrap CSS file in the head tags of your base fragment page (make sure it goes inside a fragment!)
- Include the jQuery, Popper, and Bootstrap JavaScript files in a fragment in the base fragment page. These should go in a block element outside of the body tags at the end of the file (see the lecture). If you don't have a fragment page, now is the time to create one!
- Make sure to include the fragments in all of your HTML pages.
- Run the project and look at your pages - they should look slightly different in terms of the font (if they don't, something is wrong - check all your spellings and paths).
- Push to GitHub.

---

**Part II:** Creating Navigation

- Create a navigation bar for your HTML pages:
  - This should go in the base fragment page so that you can include it all of your pages.
  - Think about the color and the styling of the navigation bar. What kind of tone does your site convey? Friendly? Modern? Professional? Classic? Upscale? Cutting-edge? High-tech? Elite? Powerful? Artistic? Eclectic? Edgy? Youthful (though not childish)? You will be expected to justify your design choice as part of your final project submission.
  - Your navigation bar must have a link to each of the pages in your site (and these should be named appropriately).

  - The navigation bar must be collapsible - in computer browsers, the navigation bar should be fully expanded with all the links visible. In mobile view, the navigation bar should be collapsed with the hamburger icon visible to toggle to see the links. You can choose whether these links are right, left, or center-aligned.
  - The links on the navigation bar should indicate whether that link is currently active (i.e. you are on that page) or not. See the lecture video - I went over how to do this!

- Include the navigation bar on each of your pages using the fragment.
- Check each page in mobile and desktop view.
- Push your code to GitHub.

---

**Part III:** Styling the form page

- Add Bootstrap styling elements to the elements on your form page.
- Your form should be centered in both mobile and desktop view.
- Textboxes should **not** extend the width of the screen in desktop view - they should be restricted to the center of the screen.
- Apply Bootstrap styling to your buttons and labels.
- Check each page in mobile and desktop view.
- Push your code to GitHub.

---

**Part IV:** Additional styling

- The name of your application must be on every page in some form. You can decide how to display your application name (as a header, as part of an image, as part of the navigation, etc), but it must be on every HTML page.
- Your `index.html` page must have an image that is related to your website and it should be centered in the page (underneath your navigation).
- Create a small logo - this should be part of your navigation bar.
- A centered copyright footer at the bottom of each page with your name: Apply Bootstrap styling to the footer as shown in lecture. This should be on every page, so put it in your base fragment page.
- Check each page in mobile and desktop view.
- Push your code to GitHub.

---

**Part III:** GitHub!!

- Commit and push your code changes to GitHub.
- A good rule of thumb is to commit and push your code every time you get a small amount of it working.
- Do not create a new or separate project for this - the whole point of GitHub is that it provides version control - you can go back to older versions of your code and you can see the whole history of your code changes (this is why committing and pushing frequently is a good idea).
- Note: I will begin opening issues on your GitHub repo when fixes are needed.

---

**Submitting your code to D2L:**

1. Zip up the project folder and submit to the D2L assignment folder. I look at your code in GitHub and also use D2L as a way to know when your GitHub repo is ready to be reviewed.