WMDD 4815 Project 2: Intermediate Web Mockup

The purpose of this assignment is to practice the techniques learned in this class to create a moderately complex website using HTML and CSS. This is an **individual** project and all work should be your own.

Your job is to create a mockup of a website; the type of website will be randomly assigned to you based on your student number. However, the constraints are loose and you have a lot of scope to improvise and be creative. In the starter files, you'll find a file called **project.html.** Open this and follow the instructions to get two options for types of websites that you may choose from.

Your mockup site doesn't have to be fully functional; it only has to demonstrate some key aspects of the type of site you've been assigned. That said, it should be intuitive to use, with no broken or non-functional links. Find examples of other websites in the genre to get an idea of what kind of content, structure, and style they often have.

General Requirements

Regardless of the type of site you've been assigned, these are the technical and design requirements:

- The site should have **four pages:**
  - **Three** pages should be of the type one would typically find on the type of website you're creating. They should all represent different areas of the site and have significantly different layouts. For example, if one were making a magazine mockup site, one page could be the home page, another could be a typical article page, and the third could be the contact or subscribe page.
  - The **fourth** page should be called **References.** It should have a **table** to organize references for images, text, code, etc. You can organize the table however you want, but it should be accessible, responsive, and have hyperlinks to all resources.
- The site design should be original, represent an appropriate challenge level, and use the techniques we learned in class. Lower marks will be given for very simple sites that only trivially satisfy the requirements. (Ask me for feedback if you're not sure about this.)
- The home page should be in the root of the project folder and be called **index.html.** Use subfolders to organize resources such as images, pages, and CSS.
- The site should use at least **one form.** You can decide the best way to use a form based on your particular website type and subject. The front-end of the form should be functional, but it doesn't have to be connected to any server-side application.
- Images and other media should either have a Creative Commons license or be in the public domain. All sources must be referenced with hyperlinks to the original source.

You must provide a link to the exact page where you got the media, not just the homepage of the website. If using code from an external resource, provide a link to the source and tell me which files and line numbers in your project use the code.

- All HTML and CSS code must validate without errors or warnings. (Take screenshots of the validation results.)
- DO NOT use Lorem Ipsum text. You can use Wikipedia text if you want, with a link to the source.
- Semantic tags must be used correctly. If you're not sure which tag to use for a particular part of the page, do some research on the web or ask me for suggestions.
  - Avoid presentational markup as much as possible unless you have a compelling reason to use it. For example, avoid using <br> tags for line breaks, <i> tags for italics, etc. You may occasionally find it necessary to use a <div> or <span> as a generic container, but only do so when necessary.
- Code must be formatted to be easily readable. You can use Visual Studio Code's automatic formatting functionality to do this: https://code.visualstudio.com/docs/languages/html#_formatting
- CSS should be organized in a sensible way with comments to label each part of the stylesheet. You can choose whichever organization method you want, as long as it makes sense and is consistent. Use variable names that make sense and are easy to interpret.
  - Make sure styles are repeated as little as possible. Remember that a single stylesheet can be applied to many pages.
- Create an attractive and usable design for your site:
  - Use the graphic design and user experience principles you've learned about in your other classes. You may want to consider making a wireframe or mockup of your site in Figma or a similar program, although this is not a requirement.
  - Make sure the text is easy to read; use appropriate Google fonts. DO NOT use the browser default fonts;
  - Make sure the design is appropriate for the target audience;
  - Make sure images are not squished or stretched; use object-fit if necessary;
  - Use a grid layout for larger screens where necessary.
- Make your site is accessible:
  - Use alt text for images, labels for icon links, and visually-hidden text labels if necessary;
  - Make sure color contrast is adequate for text;
  - Forms and tables should be accessible;
  - In general, follow the POUR principles.
- Make your site responsive:
  - The site should work well and look good on very small screens, very large screens, and everything in between. Use media queries where necessary. Avoid horizontal scrolling except for mobile tables or widgets that are specifically designed for horizontal scrolling, like image sliders.
  - Use mobile-first methodology.

  - Feel free to use Intrinsic design concepts if you want; include comments that explain the technique you're using if you think it may not be obvious.
- Use images that have a low file size but high quality. Process images using Photoshop if necessary. Use the appropriate file format for each type of image. Each page should load on a typical mobile connection in approximately 2 seconds and have a total file size of approximately 3 MB.
- Upload your site to the server so that it's accessible on the web. (See the provided instructions on how to do this.)
- Test your site thoroughly for errors, bugs, and user experience problems:
  - Test in several browsers (at least Chrome and Firefox).
  - Test on several devices. You should at least use the developer tools to simulate various devices, but it's strongly recommended that you test the site on your phone, tablet, the big screens in Langara's Mac labs, etc.
  - Get your friends to test the site. They may notice problems you haven't noticed.
  - Test the site again after uploading it to the server. TEST EVERY PAGE AND EVERY LINK. The server environment may work a bit differently than your development environment, and unexpected problems could arise.
- If you intend to make extensive use of some library, framework, or other tool, please check in with me first.

Milestones

- **November 27th**: **Peer Testing**
  - On this date, your site should be approximately 75% complete. During the lab, we'll do testing**:** you will be assigned to a group of students, and you will all test each other's sites and provide feedback and suggestions for improvement. Your testing documentation will be graded. (More info will be provided on this day.)
- **December 4th, 11:59pm**:
  - This is the due date for the completed project.

Hand In

Follow these steps to submit your work:

- Delete all unnecessary files, such as images that aren't being used. Delete code that has been commented out.
- Take screenshots of your validation results (even if there are errors or warnings) and place them in a folder within your project called **validation.**
- Collect the feedback you received from your peers and the feedback your provided to them and place it all into a text file called **group-testing.txt.**
- Name your project folder **project2-firstname-lastname**.
  - Upload the project to the server. (you can delete the group-testing.txt file if you don't want it to be accessible on the web.)
  - Zip the project folder and hand it in to Brightspace. (Make sure group-testing.txt is included.) In the comments of your Brightspace submission, include the following:
    - A hyperlink to your site URL. (I will not try to guess the URL if you don't provide a link.)
    - Any information you would like me to know when I grade your project. (You can write as much as you want.)

Grading

- [3 marks] All required pages/parts of site have been implemented;
- [1 mark] Semantic HTML;
- [2 marks] Responsive;
- [1 mark] Site has a low file size with high-quality images;
- [2 mark] Accessible with POUR principles;
- [2 marks] Well-designed with good typography;
- [1 mark] group testing feedback (from you and group members)

Total: 12

Marks may be deducted for the following:

- [-2 marks] Very simple, low challenge level (this is somewhat subjective so ask me if in doubt)
- [-2 marks] Significant errors, crashes, bugs; missing validation screenshots or invalid code; site breaks in certain situations
- [-2 mark] Not uploaded to server or some other problem with hand in
- [-2 marks] Poorly formatted code, disorganized, unnecessary files, etc.
- [-100%] Missing references and citations, copied without attribution, etc.