

# Journal for CS463 Web Development Final Project

## Project Overview

For this project, I developed a website for my final project for CS463: Intro to Web Development. This journal will document my contributions, steps I took to complete the project, and the external resources I used. This process was incredibly straightforward, and I didn't run into any major issues.

**GitHub Repo:** <https://github.com/hunteriddings/CS463-WebDev-Final>

**GitHub Deployment URL:** <https://hunteriddings.github.io/CS463-WebDev-Final/dist/>

## Session 1: Initial Setup and Planning

I initially created my webpack.config.js and copied my common package.json file I use in all my projects. From here, I used npm install to install all packages, which created a node\_modules folder. From here, I also imported my SCSS folder, containing 5 directories:

- **abstracts/** - containing functions, mixins, and variables
- **base/** - contains animations, base css (html, body, etc.), and typography definitions
- **components/** - definitions for component styles
- **layout/** - style definitions for layout styles
- **pages/** - style definitions for page styles

For the HTML structure, I used Visual Studio Code's shortcut !doc, which automatically generated the basic HTML boilerplate. I created five main sections:

1. Header - <header>
2. About - <section>
3. Experience - <section>
4. Projects - <section>
5. Contact - <section>

I then set up SCSS files for each section in my layout/ folder and styled them as needed.

**Challenges:** None. The setup and structure were seamless, and everything worked perfectly from the start.

## Session 2: Styling with SCSS and Layout

Styled the website using the SCSS structure set up earlier. I focused on laying out the sections and making sure the content was visually appealing. The design was made with mobile-first design in mind, responsive, leveraging SCSS mixins for cleaner code and reusable styling. I utilized Flexbox and CSS grid in some of my sections.

**Challenges:** None. The styling was straightforward, and all the sections aligned perfectly. I used the tools I already had in place, and they worked as expected.

## Session 3: Adding Interactivity and JavaScript

For the bottom mobile navigation menu, I implemented functionality to highlight the active item when clicked. The process involved using JavaScript to add and remove specific CSS classes to the navigation items.

1. **Selecting Elements:** I used `document.querySelectorAll` to select all the navigation list items with the class `mobilenav__item`.
2. **Setting Active State:** I created a function, `setActiveItem`, which:
  - Removes the "active" classes (`mobilenav__item-current`, `mobilenav__icon-current`, and `mobilenav__icon-text-current`) from all navigation items.
  - Adds these active classes to the clicked item, visually highlighting it as the current selection.
3. **Event Listeners:** I added event listeners to each navigation item, so that when a user clicks an item, it triggers the `setActiveItem` function and updates the active state accordingly.

For the header section of the website, I implemented two key features to enhance the user experience: a scroll-based effect for the header and dynamic height adjustment for the "about" section.

**1. Scroll-Based Header Effect:**

- I selected the header element using `document.querySelector(".header")` and created a function `handleScroll()` to detect when the user scrolls down the page.
- When the page is scrolled down more than 50 pixels (`window.scrollY > 50`), I added a class `header--scrolled` to the header element. This class can be used to apply specific styles (like a background color change or shadow) to the header when the user scrolls.
- If the scroll position is less than 50 pixels, I removed the `header--scrolled` class, returning the header to its original state.
- The function is triggered every time the user scrolls by attaching it to the window's scroll event using `window.addEventListener("scroll", handleScroll)`.

**2. Dynamic Height Adjustment:**

- I used jQuery to dynamically adjust the top margin of the `.about` section to ensure it starts below the `.head` section, regardless of screen size.
- The `dynamicHeight()` function sets the top margin of the `.about` section equal to the height of the `.head` section.
- This function is run when the document is ready (`$(document).ready()`) to ensure the elements are fully loaded before executing. Additionally, I attached `dynamicHeight` to the window's resize event (`$(window).resize()`) to handle resizing and keep the layout intact.

**Challenges:** None. The JavaScript features worked flawlessly, and I didn't encounter any issues during implementation.

## Session 4: GitHub Pages Deployment

I deployed the site to GitHub Pages. I made sure to configure the deployment correctly so the content appeared as intended.

**Challenges:** None. The deployment process was smooth, and the website was live within minutes.

## External Resources and Tutorials

I didn't need to refer to many external resources for this project as everything I used was based on personal experience. However, I did rely on the following:

- <https://sass-lang.com/documentation/>
- <https://webpack.js.org/>
- <https://fontawesome.com/>

## Conclusion

This project was incredibly smooth and straightforward. There were no issues with the setup, development, or deployment process. I was able to focus on building the website with minimal hassle and it took roughly 6 hours total to build. It was a great learning experience that helped reinforce the concepts I'm very familiar with.