

Group Project 2 - HTML - CSS

Team Members:

- Austin Ng
- Cesar Herrera
- Ken Gu
- Nhan Nguyen

GitHub:

- User ID: nhannguyensf
- Link: [WebDev317-03-Fall2023/group-project-2-website-nhannguyensf](https://github.com/nhannguyensf/WebDev317-03-Fall2023/group-project-2-website-nhannguyensf)

Assignment description:

My team is tasked with coding the website using HTML and CSS base on the design we have from the last assignment. Teams need to develop a fully navigable site, including a homepage, product/resource pages, and intermediate pages.

Approach / What we did:

1. Setup and Collaboration:

- Clone the provided starting repository.
- Set up a GitHub repository for the team.
- Ensure all team members have access to push and pull from the repository.

2. Planning:

- Review the website's design, layout, and content with our team.
- Decide on the navigation structure and the content for each page.
- Assign tasks to each team member based on their strengths and preferences.

3. File Organization:

- Use public directory as the root for all our website files.
- Inside public, create separate directories for images (images), styles (CSS), and product pages (products).
- Ensure all files and directories have meaningful names.

4. Coding:

- Start with the basic structure of the HTML pages.
- Add content to the home page, product/resource pages, and intermediate pages, etc.
- Style the website using CSS. Ensure consistency across all pages.
- Test the website in different browsers to ensure compatibility.

5. Review and Iteration:

- Regularly commit and push changes to the shared repository.
- Review each other's work and provide feedback.
- Make necessary revisions based on feedback and testing.

6. Documentation:

- Write a PDF document following the requirements.

7. Submission:

- Ensure all files and directories are within the public directory in our repository.
- Use the command provided by the instructor to tag the version of our repository we want to submit.
- Submit the PDF to Canvas.

Issues:

1. Need to control the version when collaborating

Description:

_Changes made by one developer can conflict with changes made by another.

Solution:

_Adopt a VCS (Git and GitHub platform). It allows multiple developers to work on the same project without interfering with each other's changes. Each developer can create branches, make changes, and then merge them back into the main codebase.

_Ensure that all team members understand the basics of using GitHub. Conduct workshops or training sessions about doing branches in Git.

_Implement a branching strategy like feature branching, Gitflow, or trunk-based development. This ensures that the main codebase remains stable, while development continues in separate branches.

_Encourage team members to regularly pull the latest changes from the main branch and push their own updates. This reduces the chances of major conflicts.

2. Inconsistencies in the codebase

Description:

_We had different style of writing code, such as indentation levels, varied naming conventions, or diverse ways of structuring code. This made the code hard to read, understand, and maintain. It led to merge conflicts and other collaboration issues.

Solution:

_We had team meetings to agree on a coding standard, which was based on widely-accepted industry standards or tailored to the team's preferences.

_We used Prettier tool to automatically detect and fix style issues. Integrating these tools into development process ensures that code adheres to a consistent style.

- _ Implement a code review process where team members review each other's code before it's merged. This not only catches style inconsistencies but also improves code quality by leveraging multiple sets of eyes.
- _ Hold regular meetings or check-ins to discuss any challenges or discrepancies in coding styles. This helped in addressing any deviations early on.