Implementation

Group 13: Good Draw

Link to Github Repository: https://github.com/CSE-110-Group-13/warm-up
Demo Video Youtube Link: https://youtu.be/wq0vZxtx6-U

SWOT Analysis

Calendar + Team Workflow & AI Usage

Strengths

The calendar has a clean design. The drop down menu and the two navigation buttons look good and are easy to navigate. The calendar itself looks aesthetically pleasing, without an overcrowding of graphics or color. It keeps the user data during the session even if the user switches months. All of the header components do not have break cases. The three buttons below the title all add useful features.

Team meeting was held beforehand to decide on dividing members into two different groups. A deadline of Wednesday was set to get most of the code completed and then the following days for updates and bug-fixes. After integrating usage of GitHub issues, pull requests, and branches, development was able to be done more in parallel after the initial core calendar functionality implementation was completed. Two new channels were created on Slack for the two tasks for better communication.

Weakness

Currently, the UI is not inherently straightforward to the user. The website does not gracefully fail. The mobile experience of the website is much worse due to inconsistencies with using absolute units and relative units for CSS measurements. The information about the months is currently hardcoded. There is no ability to read or write with JSON files, so no legitimate ability to deal with stored data.

A lack of initial designing and features specification led to a slow start for developing the core functionality of the calendar. Project management tools like GitHub issues and task boards were not fully taken advantage of from the start, leading to a poor first implementation with code-first design. A lack of proper testing and debugging led to issues that would arise later when implementing new features. There wasn’t an established process for reviewing and merging pull requests. Merging was often done by the person who created the branch and PR

themselves, rather than having assigned code-reviewers. There should have been more active communication about implementing features and new pull requests.

Opportunities

The calendar looks very clean, so slightly improving UX would make this calendar very pleasing for users overall. Furthermore, users enjoy customization, so the ability for users to change the background via colors or images and then change text and line color would allow this app to feel like a clean modular masterpiece. Could make it more accessible.

Approaching the project with a design-first method would have made the process go a lot smoother. Usage of project design tools like Miro to decide on how the calendar would look, what features it would have, and an established idea of how to implement it would have been useful to commit to from the start. With a clearer idea of the implementation, it would have been easier to create GitHub issues and assign tasks to different people for a more parallel coding process.

e.g. (Our miro board for the core functionality) https://miro.com/app/board/uXjVKPmAAAY=/

Threats

The calendar currently lacks features and the ability to add daily events is pretty unintuitive at the moment, so users may quickly discard the calendar without appreciating the more technical aspects of the app. The UI and UX can cause users to turn away from the app along with the inability to truly save data.

Due to a lack of design-first thinking, the implementation of new features ended up being janky as it would create new bugs sometimes or cause mobile accessibility to fail due to squished elements. Further improving this calendar would most likely require refactoring the code-base or changing some core implementation to add features like storing events in a backend or being able to change the year.

The use of artificial intelligence was both beneficial and harmful. AI was used to generate a base template of the calendar in HTML, but it resulted in a hard-coded approach that didn't allow for different months. However, when it came to learning new syntax in HTML/CSS/JS and helping to identify why a simple bug was happening, ChatGPT served as a useful third-party tool. For more complicated bugs, ChatGPT wasn't good at problem solving and identifying the real reason why a bug was happening. Carelessly using code from ChatGPT would result in janky code that would only appear to work on the surface before more in-depth testing.

Task List

Strengths

The task list is user friendly and makes it easy for users to view their tasks organized by date. The ability to cross out tasks when completed allows users to have a visual indication of their progress. The task list has a simple design and color scheme to effectively separate the different tasks by date without overwhelming the user.

Weakness

The task list relies on JavaScript for the interactive feature of crossing out tasks upon completion. If JavaScript is disabled, users lose the functionality of being able to uncross and cross out tasks. Also, the specific tasks are hard coded into a JSON file. There is no ability to write to a JSON file, so the task items cannot be modified.

Opportunities

The user experience can be enhanced through adding more interactive features such as the ability to add and delete tasks along with customizable options such as themes and different color options. This would improve the overall experience of the user through making the task list more user friendly and personalizable.

Threats

The task list lacks the ability to add and delete tasks from the page, which may lead to user dissatisfaction. The lack of functionality may result in decreased user engagement and cause users to become disinterested in using the task list due to its limited capabilities. Therefore, users may quickly discard it and find alternative solutions that offer more interactive features.