Team Information

- Team number: 14
- Team name: Bozosort
- GitHub repository link: https://github.com/cse110-sp24-group14/warmup-exercise
- Youtube link: https://youtu.be/Q1QmeVGHyQM

Summary

We created a task list app, where data was pulled from a premade JSON file. The app does not have any CRUD functionality.

Cindy (Huimeng) Lu created designs in Figma.

Dev workflow

- Tasks details were put in a GitHub issue
- Issues were then assigned to each member of the team
- Devs created branches to work on a task, and made a PR when finished
    - Used slack to communicate when a PR has been created
- Code reviews were conducted by Brenton and Advaith
    - Only leads conducted code reviews due to time constraints

Devops

- Tasks tracked via GitHub issues
- Branch protections were put on the main branch so that changes must be made through a PR
    - Leads took care of any merge conflicts that emerged
- PR template was made for consistent PRs

SWOT Analysis

Strengths

- Organization
    - Compartmentalization of tasks allowed for increased parallelism, increasing efficiency
- User-Friendly Interface
    - The interface is easy to use and easy to understand. There are not a lot of moving parts and all the functions are intuitive. Everything is clearly organized for the user to see all on one page.
- Proper Code Management
    - The main code can't be merged without proper appraisal and approval
    - Used GitHub Issues to create individual tasks and disperse them equally among team members.
- Well-Paced
    - Issues were created during the weekend so they could be completed at members' preferred time throughout the week.
    - There were set deadlines but they were adjustable to account for the unaccountable.

Weaknesses

- Task priorities?
    - Some tasks required other tasks to be completed beforehand. A task priority system would allow members to understand which tasks should be prioritized first so the following tasks could be worked on.
- Limited functionality
    -
- Reliability on JS
    - To accommodate all devices and cases, we must also consider the case where JS is not available on the user's device. The program relies on JS to populate the table from a JSON file which will show a blank table when JS is removed.
- Time constraints
    - The short 1-week timeframe meant that there was not a lot of time to develop and test features.
    - Possibly have overlooked key features, or have implemented unnecessary ones.
- Lack of user privacy

    - Users do not have any privacy on what tasks are on the website as anyone can access the website and see them as well.

Opportunities

- Non-js fallback
- Rotating PR reviewers
    - For the warm-up, only Brenton and Advaith managed PR requests to save time.
- Mobile Development
    - As the task list is something users will be constantly checking to know what they need to do, having quicker mobile access to the site would benefit the users' needs.
- Adding new features
    - Being able to add tasks or delete tasks would allow users to customize tasks to pertain to their own lives.

Threats

- Competition/Individuality
    - There are a lot of other task lists on the internet making it difficult to stand out and draw in users.