A blue and yellow logo

Description automatically generated

CSES WebDev Main Site

1) Overview

In aim of developing this web application with assured quality from start to finish, we will practice a proper Software Development Life Cycle (**SDLC**), making sure working as a team will both make development more efficient, and decrease the chances our product is faulty in any way.

This entails firstly,

Developers and Designers will work thoroughly with one another:

* Developers should consult the designers for any instruction or clarification on the UI of any task if it’s not provided already.
* Likewise, Designers should consult with the developers to see that the plans they propose are feasible and will take a reasonable amount of time to complete.
* Designers communicate with me for any needed information or clarification, and I’ll communicate with CSES WebDev (for design questions, contact Jheel, the VP of design [jhghandi@ucsd.edu](mailto:jhghandi@ucsd.edu)).
* Developers communicate with me for any questions and especially for any administrative questions or concerns, such as adapting new technologies or practices.

This also entails,

Our development process will be as follows:

* Any major UI/UX design will be opened as a task, visible on our GitHub Org’s Projects.
* After these designs are reviewed and approved, the appropriate frontend and backend tasks will be opened to implement them.
* Developer(s) create a subbranch under the “dev” branch, which has the latest approved features/fixes.
* Once these features/fixes are finished and tests are written to ensure them, a pull request is made to merge the subbranch and “dev” branch, which automatically runs provided tests, after which I will review it and approve for merging.
* Every week, “dev” and “main” branch is merged (main is production that everyone sees)

2) Designers

You will find your tasks on GitHub Projects, tagged specifically as [UI/UX], but grouped with developer tasks if they exist. To complete any task, you pretty much have free will as to how you approach designing any given UI, and the UX to justify it. The only thing you should do is document all of it including the UX, since the developers will rely on these documents. For that reason, try to be concise and clear as possible, and include any resources or aids that you think might help with comprehension. If there is a big issue to address, I will mention it anyways during review.

Also, your development process is slightly different than the technical ones. The documentation repo only has a “main” branch, which is what all our team members will see. However, you will still need to create a subbranch for you to work in, then submit a pull request for me to review your changes and approve them. Though you don’t need a “dev” branch like the developers, all changes still need to be reviewed, so a developer doesn’t accidentally refer to a faulty doc.

3) Developers

Just like the Designers, I will open tasks for development based on newly approved design changes, tagged specifically as [Frontend] or [Backend], but grouped with designer tasks if they exist. To complete any task, feel free to implement it in any way you see fit, but **try** to avoid using new technologies. Sometimes this is unavoidable, such as requiring a MySQL database to store user data. However, for big changes that I anticipate such as these, I’ll likely administer them beforehand, but let me know regardless since I may miss things.

Both Frontend and Backend repo’s contain README’s that thoroughly explain how to set up a local environment to test your branch.

For both, be sure to **document** all relevant changes that are required to understand your changes. For relatively small things, that can simply be a comment in code, but for bigger things, like migrating new technologies or adding API endpoints or such, you should write it down in the documentation repo.

For frontend, there are separate secrets for both “main” and “dev” environments that contain the “prod” and “dev” stages of our API’s base URI, respectively. Thus, for local testing, you can ask me for the “dev” API’s URI and put that in an .env file.

That’s it for now! Make sure you keep up with the Slack channel and DM/email me if needed!