

Hello team!

To properly work on the project collaboratively we need to set up some rules/conventions for our repo.

First, I will give a basic understanding/overview on the expected flow when usually working:

1. You have an issue that you will work on.
2. You make a branch for that issue using the conventions I will send.
3. You add your issue/ticket in the project section found in our repository. [Set it as in progress]
4. You work on it.
 1. Try to commit as many changes you have done, whether it is adding/deleting/editing. [small commits, do not write huge code then commit a lot, make sure it is small and clear]
 2. When committing you need to follow the format/convention I mention below.
 3. Feel free to push after any amount of commits you want!
5. You finish working on it and it is time to merge your branch with the main branch!
 1. You should make a Pull Request to merge what you have done with the main branch! The naming of the PR has to follow the convention.
 2. You resolve any potential merge conflicts found.
 3. You wait for it to be reviewed.
 4. After successfully reviewed, the owner of the PR must close it. When closing it, the previous branch is deleted. [You have a branch for each issue you work on].
6. Move it to “completed” in the projects section when done.

Conventions:

- Branch naming conventions:
 - General structure:
 - <GitHub-Task-ID-WITHOUT-HASHTAG>/<initials>-<short-description>
 - <short-description>: Short, describe branch purpose, separated by dashes.
 - Ex: 12/YD-navbar-alignment
- PR title naming convention:
 - Type list:
 - <type>: Describes the purpose of a commit.
 - feat/ for a new feature
 - fix/ for a bug fix
 - chore/ for maintenance

- General structure:
 - <GitHub-Task-ID>-<type>: <description>
 - Ex: #39 - feat: Adding user authentication
- PR description:
 - General structure:

Summary

Provide a short description of what changes are introduced

Changes Made

 - Change 1
 - Change 2
 - ...

Testing Instructions

Steps to verify that the changes work as expected

Screenshots (if applicable)

Attach screenshots showing the change, usually for frontend

Additional Notes

Anything else relevant for the reviewers to know
- Commit message conventions:
 - Commits should be small and focused on specific change!
 - Icons, they indicate what you did:
 - [+]: Added
 - [-]: Removed
 - [~]: Fixed
 - General structure:
 - <GitHub-Task-ID> - <Icon> - Clear description of the changes.
 - Ex: #1 - + Added OAuth login support in Login.jsx