**GitHub Push Guidelines**

To maintain code quality and ensure a collaborative development process, follow these guidelines before submitting a push to GitHub. Code reviews are mandatory for all changes.

**Before You Push**

**1. Write Clear Commit Messages**

* Ensure your commit messages are descriptive and follow this format:
  + **Title:** A short summary (max 50 characters).
  + **Body (Optional):** Detailed explanation of changes, including:
    - Why the change was necessary.
    - What was changed.
    - References to related issues or tasks (e.g., "Fixes #123").

**2. Ensure the related tasks went through the testing phase**

* The task should have related tests that pass.

**3. Check Code Quality**

* Perform a self-review of your code:
  + Ensure consistency with project coding standards.
  + Remove any debugging statements or commented-out code.
  + Optimize for readability and maintainability.

**4. Pull the Latest Changes**

* Sync your branch with the latest changes from the main branch (or equivalent):
* git fetch origin

git rebase origin/main

* + Resolve any merge conflicts locally before pushing.

**5. Create a Pull Request**

* Push your branch to GitHub:

git push origin <branch-name>

* Open a pull request (PR) from your branch to the main branch:
  + Add a clear title and description.
  + Link any related tasks or issues.
  + Assign reviewers.

**Code Review Process**

**1. Assign Reviewers**

* Unless there is a reason to, no reviewer should be assigned. Send a message to the relevant chat and someone will review it.

**2. Respond to Feedback**

* Address comments or suggestions from reviewers promptly.
* Push additional commits to the same branch to reflect changes.

**3. Approval Before Merging**

* Wait for at least one approval (or the required number, if specified) before merging.
* Ensure all conversations in the PR are resolved.

**Merging the Code**

**1. Final Checks**

* Verify that:
  + All tests have passed.
  + All required approvals are received.
  + The branch is up-to-date with the main branch.

**2. Merge the PR**

* Use the "Squash and Merge" option (preferred) to keep the commit history clean.
  + Add a meaningful commit message summarizing the changes.

**3. Delete the Branch**

* After merging, delete the feature branch from GitHub to keep the repository organized.