

Branch Workflow Guidelines

Our workflow uses two main branches: **main** and **dev**. Here's how they work:

- **main Branch**
 - Changes to the **main** branch need **approval from all team members** before merging.
 - This ensures that only thoroughly reviewed and agreed-upon changes are added to the final project.
 - **dev Branch**
 - Changes to the **dev** branch **do not** need approval.
 - You are responsible for reviewing your own work before submitting changes here.
-

How to Contribute

Follow these steps to contribute to the project:

1. Get the Project

- Download the project by cloning the repository:

```
git clone https://github.com/erdenezaya/gc_2_app.git
```

- Move into the project folder:

```
cd gc_2_app
```

2. Start Your Work

- Create a new branch for your task:

```
git checkout -b my-task
```

3. Make Changes

- Edit or add files as needed.
- Save your changes and prepare them for submission:

```
git add .  
git commit -m "Briefly describe what you changed"
```

4. Push Your Work

- Upload your branch to GitHub:

```
git push origin my-task
```

5. Submit Your Work

- **Navigate to the Repository:**
 - Open your web browser and go to the GitHub repository:
https://github.com/erdenezaya/gc_2_app.
- **Create a Pull Request (PR):**
 - Click on the "Pull requests" tab in the repository.
 - Click the "New pull request" button.
 - Select your branch as the source branch (e.g., **my-task**) and the target branch (**dev** or **main**).
- **Provide Details for the PR:**
 - Add a descriptive title for your pull request.
 - Write a detailed description of the changes you made, including the purpose and any relevant context.
- **Submit the PR:**
 - Click the "Create pull request" button to submit your PR.
- **Follow the Approval Process:**
 - If targeting the **dev** branch, ensure your changes are self-reviewed and ready for integration.
 - If targeting the **main** branch, wait for all team members to review and approve your PR before merging.
- **Address Feedback:**
 - If reviewers leave comments or request changes, make the necessary updates to your branch.
 - Push the updated changes to your branch:

```
git add .  
git commit -m "Addressed review feedback"  
git push origin my-task
```

- The PR will automatically update with your new changes.
 - **Merge the PR:**
 - Once all approvals are received (if required), merge the PR into the target branch.
 - If you do not have permission to merge, request a team member with the necessary access to do so.
-

Helpful Tips

- **Stay Updated:** Regularly update your branch with the latest changes from **dev** or **main** to avoid conflicts:

```
git pull origin <branch-name>
```

- **Write Clear Commit Messages:** Use short and meaningful messages to explain your changes.
- **Communicate:** Keep in touch with your team to ensure smooth collaboration.
- **Clean Up:** After your branch is merged, delete it to keep the repository organized.

Create an Issue After One Iteration

After completing an iteration, follow these steps to create an issue for the next task:

1. Navigate to the Repository

- Open your web browser and go to the GitHub repository:
https://github.com/erdenezaya/gc_2_app.

2. Open the Issues Tab

- Click on the "Issues" tab in the repository.

3. Create a New Issue

- Click the "New issue" button.

4. Provide Details for the Issue

- Add a clear and descriptive title for the issue.
- Write a detailed description, including:
 - The purpose of the task.
 - Any relevant context or background information.
 - Specific steps or requirements for completing the task.

5. Assign and Label the Issue

- Assign the issue to the appropriate team member(s).
- Add relevant labels (e.g., **enhancement**, **bug**, **documentation**) to categorize the issue.

6. **Submit the Issue**

- Click the "Submit new issue" button to create the issue.

By creating issues for each task, you can ensure better tracking and organization of the project's progress.