

Date: November 12th

Attendees: Emmanuel, Wail, Yesenia, Shichang

Notes:

*Teach the team how to create and use Git branches.

- Demonstrate the workflow for:
 - Creating branches
 - Committing changes
 - Pushing to remote branches
 - Creating pull requests (PRs)
 - Merging changes
 - Resolving merge conflicts

1. Creating and Using Branches

- You **should not** work directly on the `main` branch.
- Each person must create **their own branch** for their tasks.

Example:

```
git checkout -b branchName
```

- Branches appear on GitHub under the repository's branch list.

2. Pushing Local Changes to GitHub

Workflow demonstrated:

Check status

```
git status
```

Add changed files

```
git add <filename>
```

Commit changes

```
git commit -m "Your message"
```

Push to your remote branch

```
git push origin <branchName>
```

3. Creating a Pull Request (PR)

- After pushing, GitHub shows “**Compare & Pull Request**”.
- The PR is a request for teammates to **review and approve** your changes.
- PR steps:

- Provide description/notes if needed.
- Submit the PR.
- Another team member must review and approve.
- Then the PR can be merged into `main`.

4. Reviewing & Merging PRs

- Any team member can review someone else's PR.
- Reviewer can:
 - **Approve**
 - **Request changes**
 - **Comment**
- Once approved, the PR can be **merged** into main:
 - Either by the author or another team member.

5. Pulling Latest Changes - IMPORTANT

Before starting new work and before creating a branch or PR:

```
git pull origin main
```

- Ensures your local project is up to date with team changes.

How conflicts happen:

- When two people change the same section of the same file.
- The person who tries to merge second sees the conflict.

Resolving Merge Conflicts

Steps shown:

1. **Attempt to merge → conflict detected**
 - GitHub shows:

"This branch has conflicts that must be resolved."
2. **Options to resolve:**
 - Use **Eclipse merge conflict tool**
 - Right-click file → **Team** → **Merge Tool / Edit Conflicts**
 - Or resolve manually within GitHub.
3. **In Eclipse Merge Tool:**
 - Left pane = your version
 - Right pane = incoming version
 - Middle pane = final resolved version
 - Arrows allow selecting changes from either side.

- You can also manually combine code for a hybrid result.
- 4. **After resolving:**
 - Save the file
 - `git add <file>`
 - `git commit`
 - `git push`
 - GitHub will then allow the PR to proceed.
- Reminder: Submit work in the next few days.
- Meeting on the 17th will confirm time later this week!