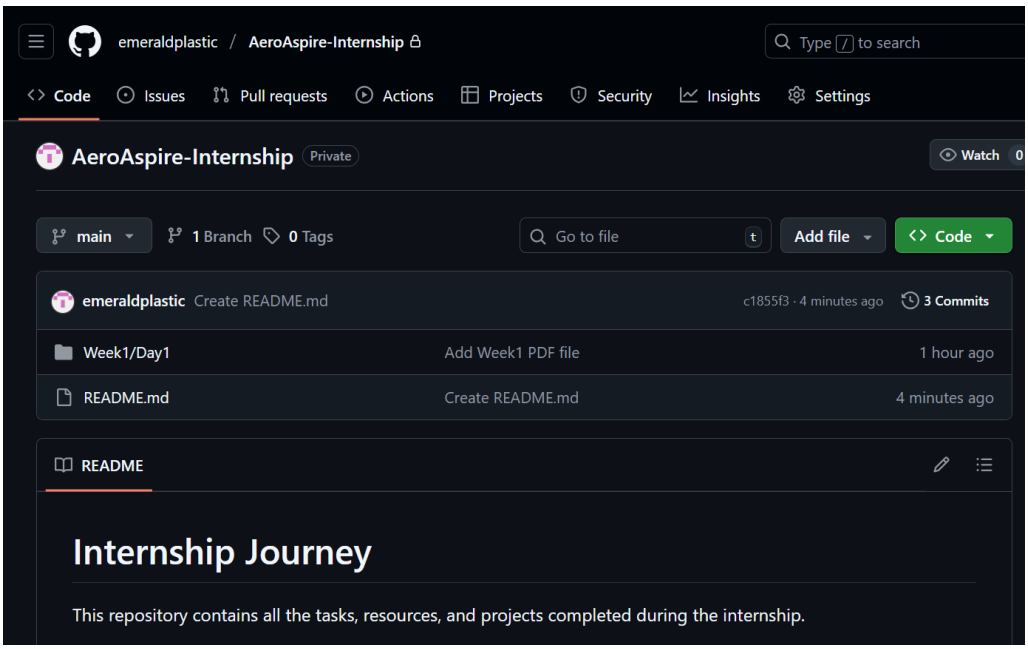
AeroAspire-SDE Intern
John Nikhil G

Week 1 – Day 3 (September 24)

Task/Assignment :

- Initialize repo; commit daily work; create feature branch; merge after review



Steps: For Commit in a new repository:

- Start by creating your repository.
- Give your repository a meaningful name.
- Initialize the repository locally with git init.
- Stage the files you want to include by using git add ./filename.
- Commit your changes with a descriptive message using git commit -m "Your message".
- Link your local repository to a remote one by adding its URL via git remote add origin https://github.com/yourusername/repositoryname.git.
- Finally, upload your commits to the remote repository with git push -u origin main.

QUESTIONS:

1.What is the workflow from making changes → staging → commit → push?

The process usually starts in the working directory where you make changes by editing, adding, or removing files. Next, these changes are prepared for recording by using git add to stage them. After staging, the changes are permanently recorded in the local repository through a commit using git commit. Finally, these committed changes are shared with the remote repository when you run git push. This sequence, often called the "GitHub Flow," underpins most teamwork using Git.

2.What is a merge conflict, what causes it, and how do you resolve it?

A merge conflict happens when different branches contain changes to the same file that overlap or contradict each other.

- The conflict arises because Git cannot automatically merge sections where the code is similar, duplicated, or includes new additions.
- In this situation, Git pauses and requires manual intervention to decide which parts of the code should remain.
- The resolution involves reviewing both the incoming changes and the existing code to determine how to best integrate them.
- You then select the code segments to keep in your local version of the file.
- After resolving these conflicts, the file is saved, staged, and the merge completes by pushing the final version.
- Notably, merge conflicts can occur not only when working with multiple branches but also when several collaborators push changes simultaneously to the repository.

3.Describe what happens under the hood with git commit: what objects are stored?

When a commit occurs, Git captures a snapshot of all the files that were staged, creating a new commit object in the repository history. This commit points to tree objects representing the project's directory structure and blob objects that hold the file data. Each commit also references its parent commit(s), linking the project’s history together and forming the backbone of branches and version tracking. Visual tools like "Learn Git Branching" effectively illustrate these relationships and deepen understanding of Git’s inner workings.