Assignment 3 Submission Report

Git Repository Link:- [Github Link]

1. Project Setup:

- Open a terminal or command prompt and navigate to the directory where you want to create your project.
- Initialize a new Git repository with the command: `git init`
- Create your Python project files.
- Commit your changes using:

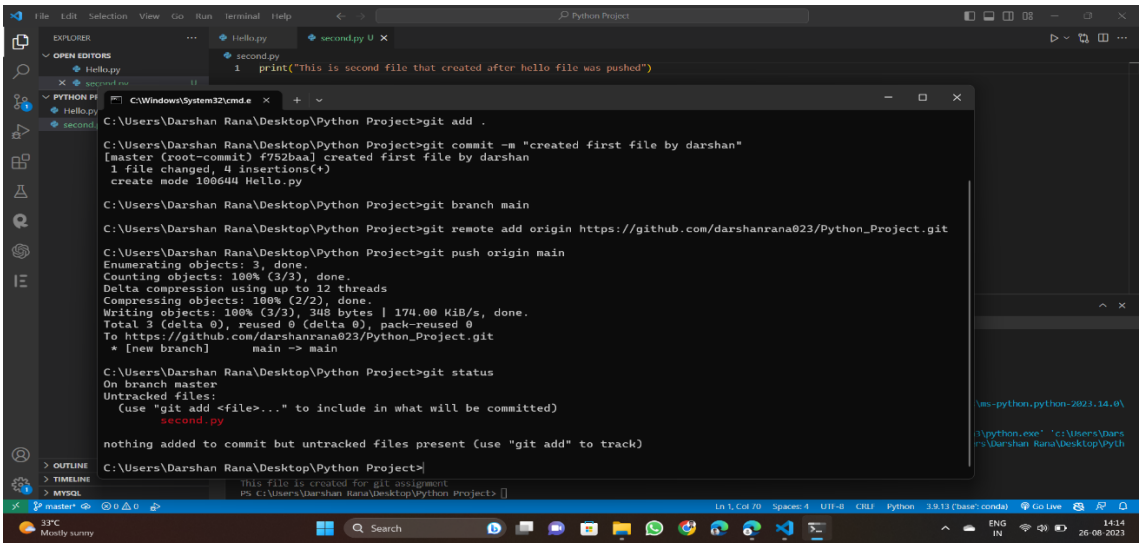
➢ git add . (to stage all changes)

➢ git commit -m "Initial commit"

- Create a new repository on GitHub.
- Link your local repository to the remote GitHub repository:
- git remote add origin "GitHub repository URL"

➢ git branch -M main

➢ git push -origin main



2. Branching and Development:

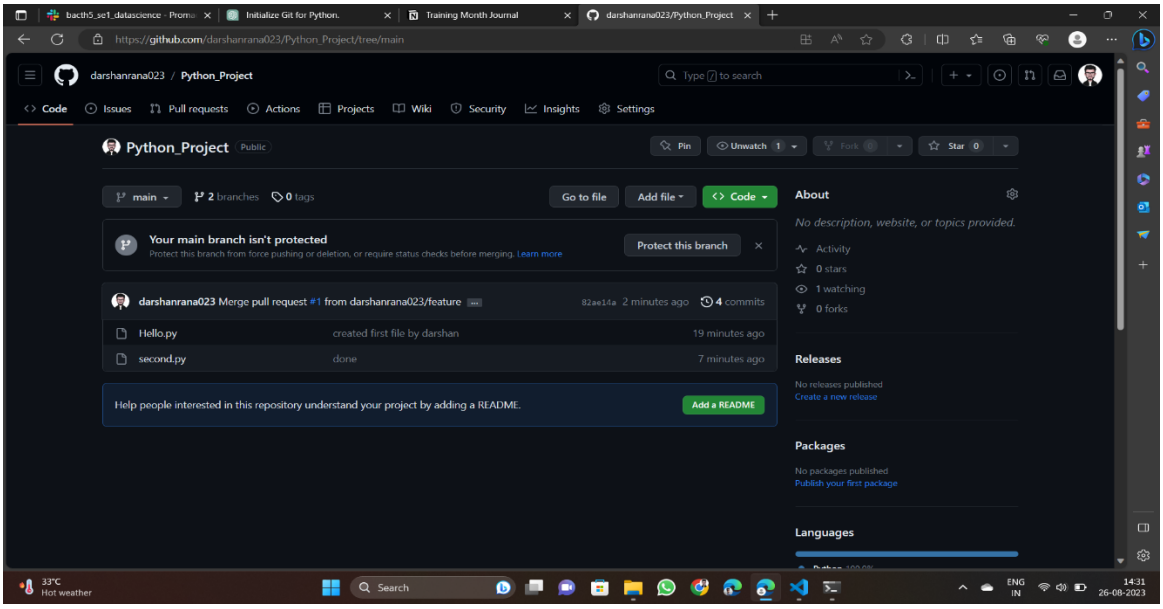
- Create a new branch for a feature:

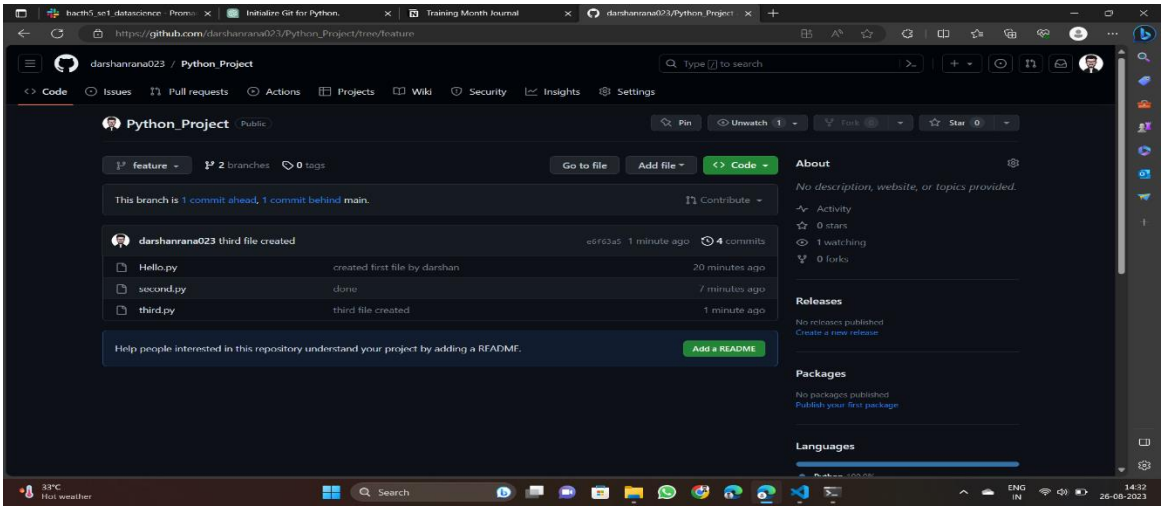
➢ git checkout feature

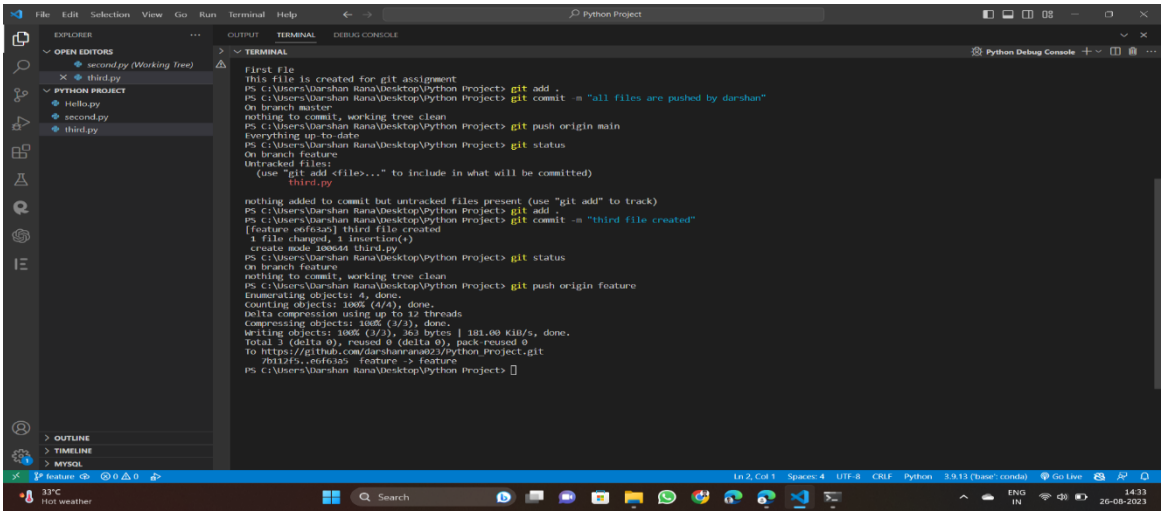
- Develop your feature and commit changes:

➢ git add .

➢ git commit -m "Implement feature: your feature description"

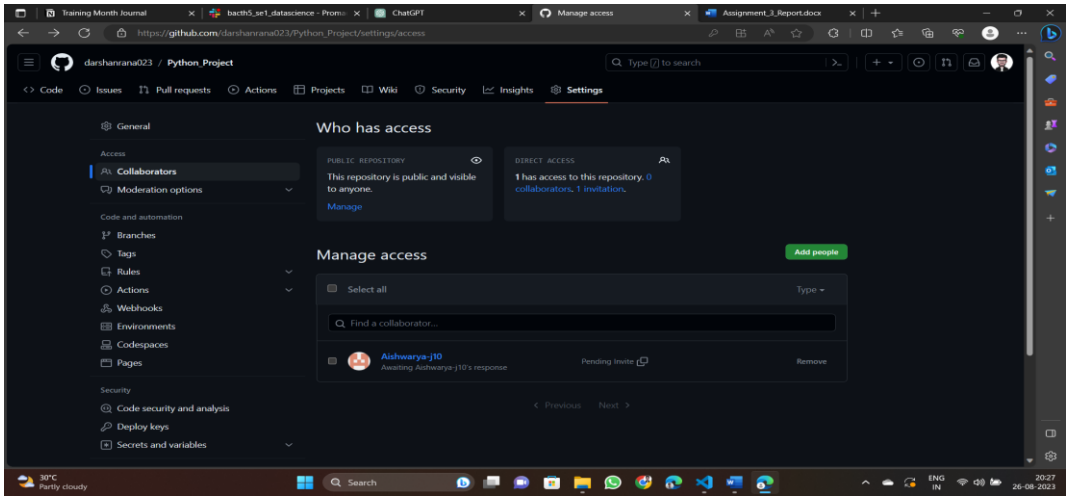






3. Collaboration Simulation:

- Peer should clone the repository:
  - ➢ git clone <GitHub repository URL>
- Peer creates a new branch, makes changes, and commits:
  - ➢ git checkout -b feature/peer-feature
- Peer pushes changes to their branch:
  - ➢ git push origin feature/peer-feature
- Create a pull request on GitHub. Review, comment, and merge the pull request.



4. Handling Merge Conflicts:

- Introduce a merge conflict by editing the same lines in the same file on different branches.
- Commit changes on one branch.
- Attempt to merge the other branch:

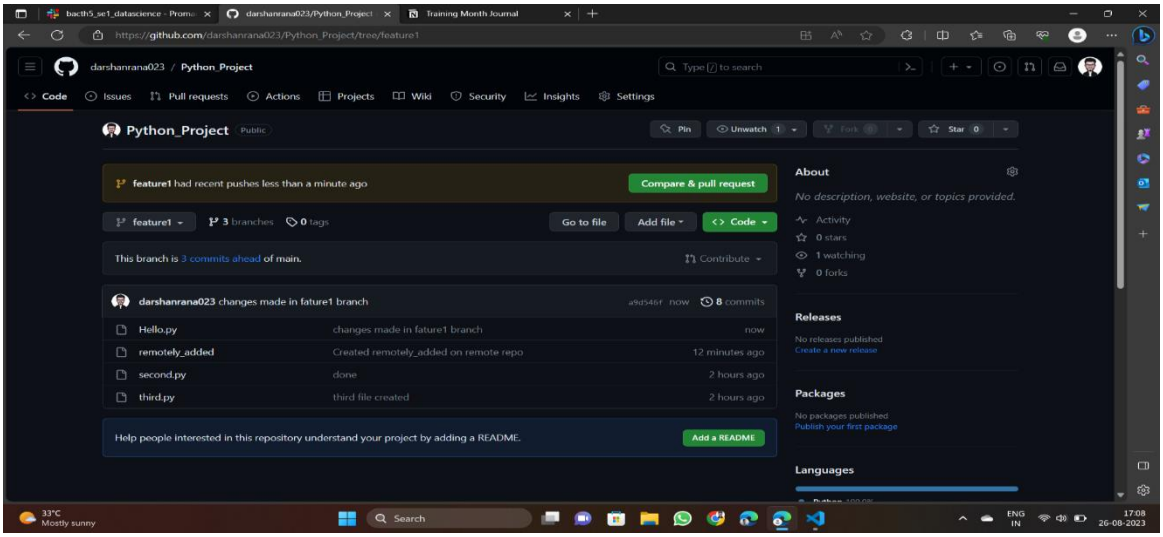
➢ git checkout main

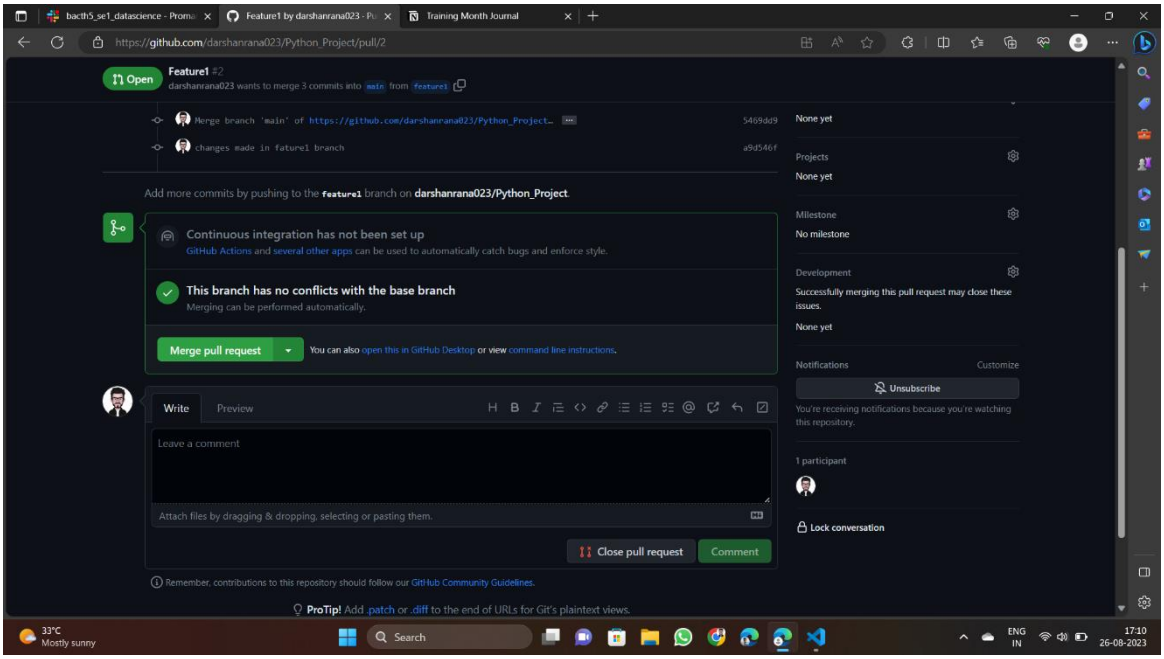
➢ git merge feature/other-branch

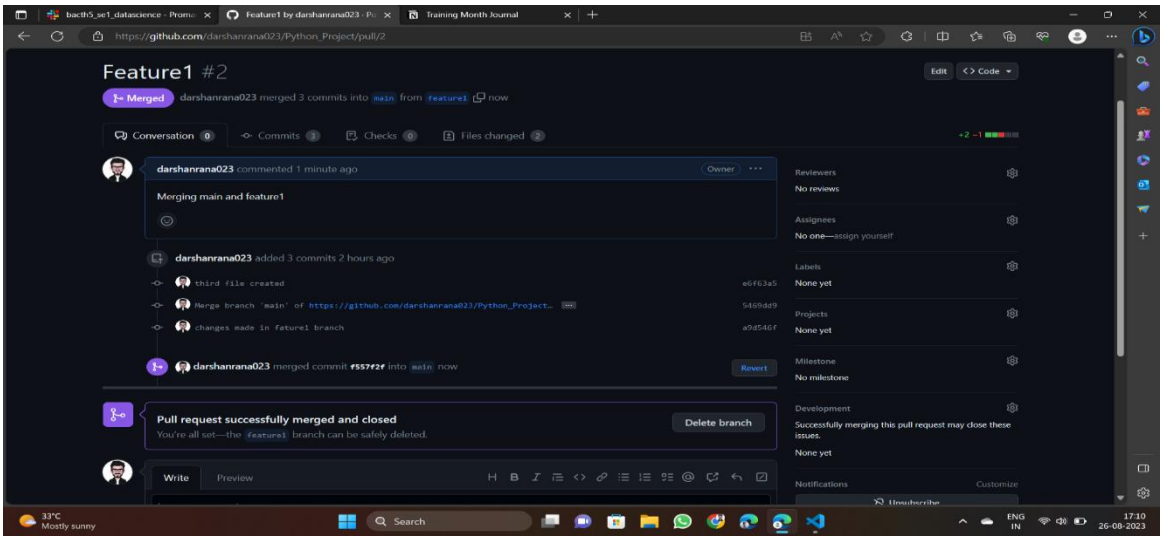
- Resolve the conflict by editing the conflicted file. Then:

➢ git add .

➢ git commit -m "Resolve merge conflict"







5. Rebasing and History Cleanup:

- Create a new branch and make several commits.
- Rebase the branch interactively:

➢ git checkout feature/rebase-branch

➢ git rebase -i HEAD~<number of commits>

- Squash, reword, or rearrange commits in the interactive rebase menu.
- Merge the branch back to main using a fast-forward merge:
  - ➢ git checkout main
  - ➢ git merge feature/rebase-branch