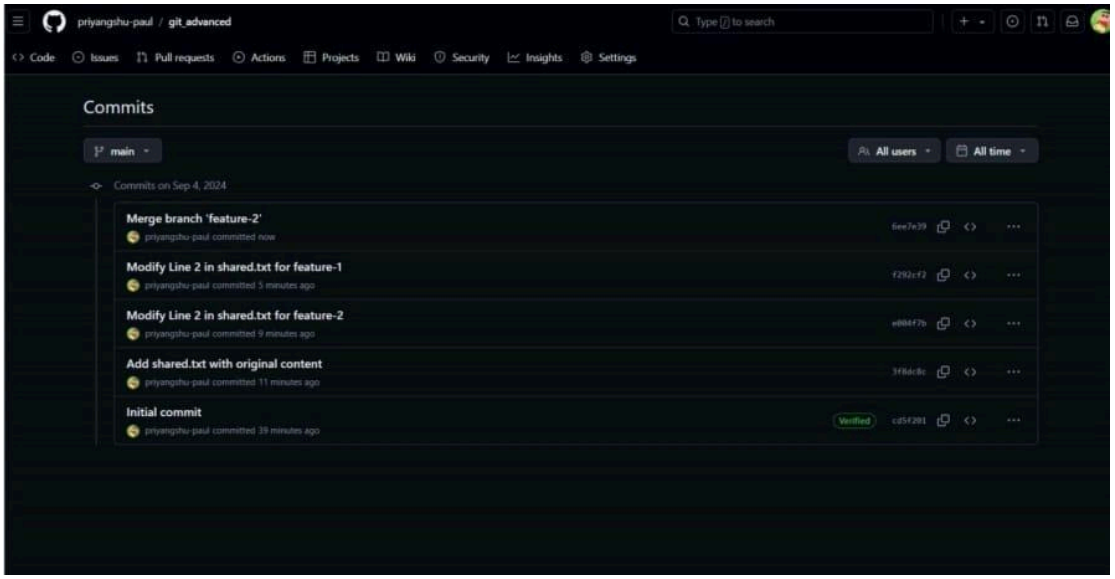
Git Assigment-3

Assignment- Git Branching and Merging

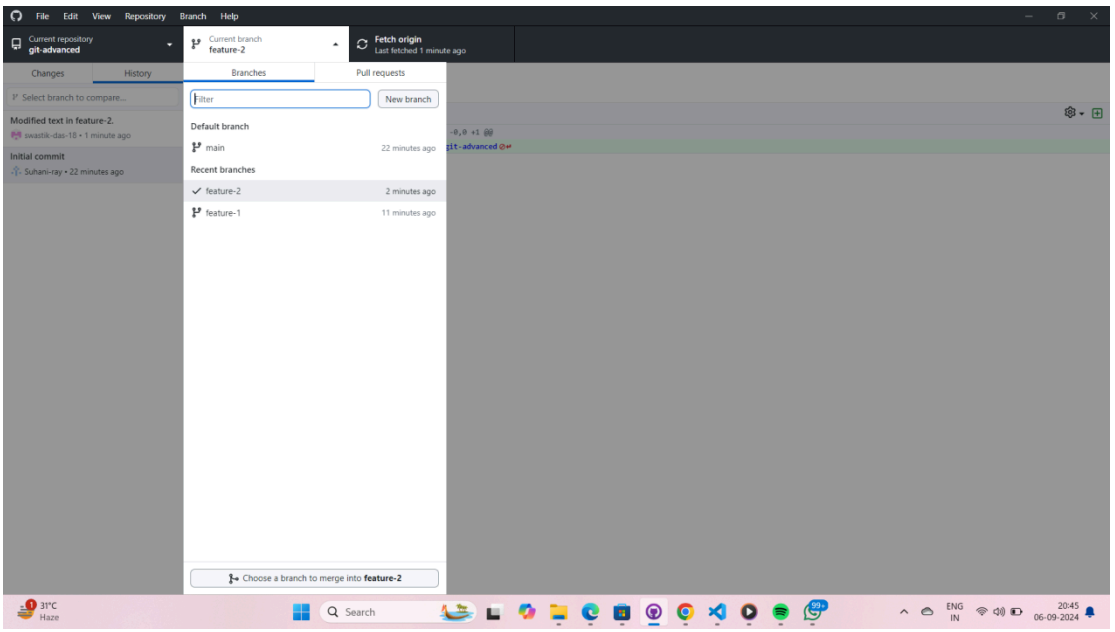
Objective: Demonstrate proficiency in Git branching, merging, and conflict resolution.

Screenshots: -

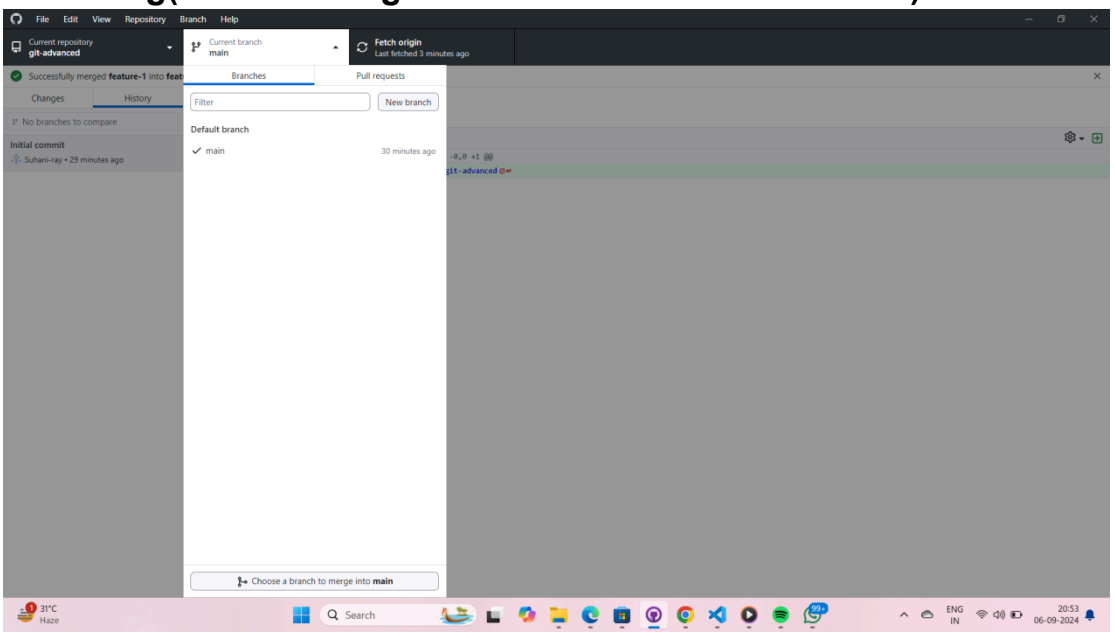
1. A screenshot of my GitHub repository showing the commit history.



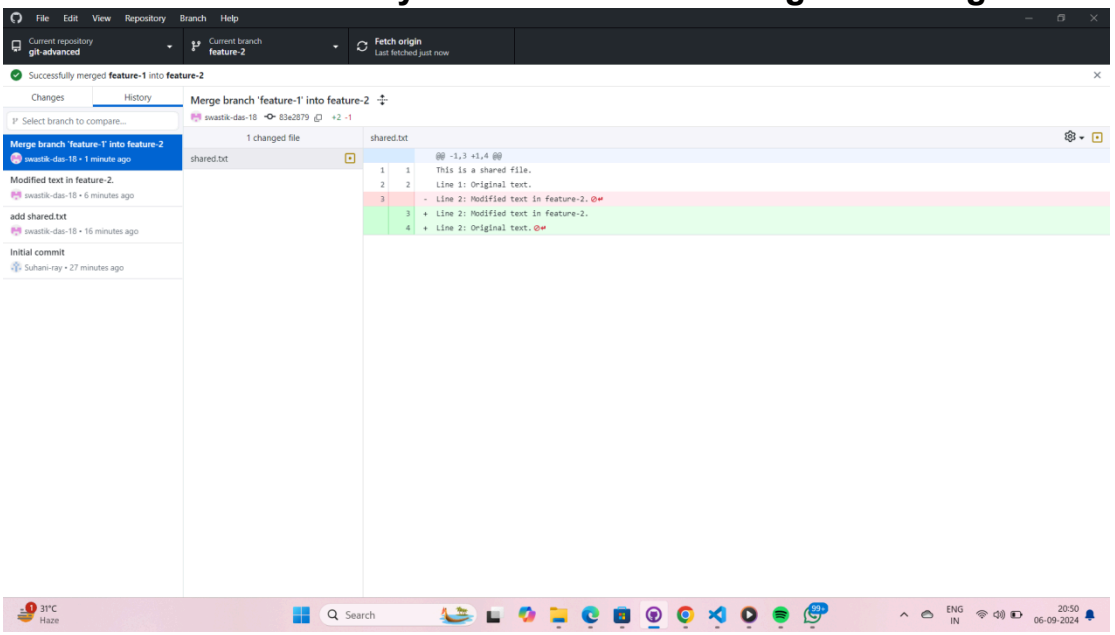
2. A screenshot of my GitHub repository showing the branching



3. A screenshot of my GitHub repository showing the branching(After deleting feature-1 and feature-2 branch):



4. A screenshot of my local machine showing the Git log.



❖ A brief write-up of my experience with Git branching and merging:

My experience with Git branching and merging has been instrumental in managing codebase changes and collaborating eU’ ectively within teams. Branching in Git allows me to create isolated environments for new features, bug fixes, or experiments without aU’ ecting the main codebase. This practice enables parallel
development, where multiple features or tasks can be developed concurrently by diU’ erent team members.

Merging, on the other hand, is the process of integrating these branches back into the main codebase. I often use the merge command to combine changes from diU’ erent branches, ensuring that the work done in isolated environments is smoothly integrated.

Handling merge conflicts—when diU’ erent branches modify the same part of a file—has also taught me to carefully review changes and communicate with team members to resolve issues.

I've used both fast-forward merges and three-way merges, depending on the situation. Fast-forward merges are straightforward when the main branch hasn’t changed since the branch was created. Three-way merges, while more complex, are essential when the branches have diverged significantly