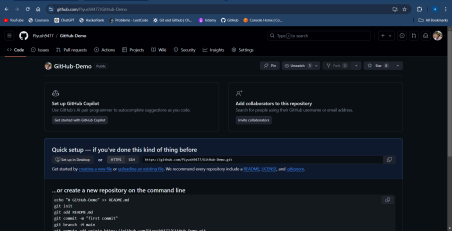
**Assignment 2**

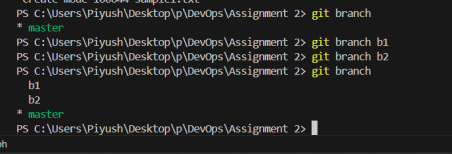
Aim: Perform the following operations on Git and GitHub

● Create a Repository:



● Create Branches:

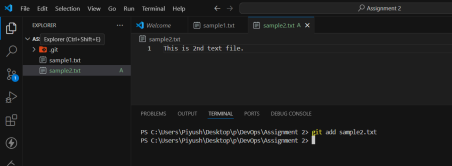
git branch - This command lists all the branches in the current repository. git branch <branch-name> - This command creates a new branch called <branch-name>.



● Perform Add, Commit, Push, Pull:

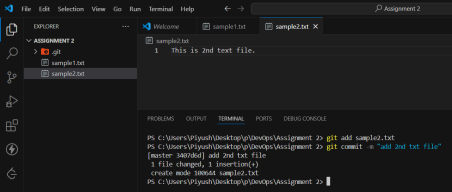
○ Add:

git add - This command adds a change in the working directory to the staging area



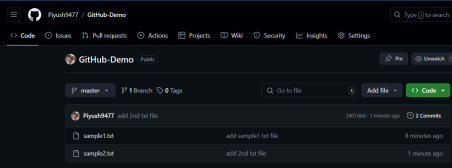
○ Commit:

git commit -m "commit message" - This command is used to record changes, which are staged, to the repository.



○ Push:

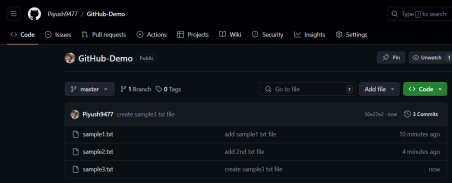
git push <remote-name> <branch-name> - This command is used to push the files from local repository to remote repository.

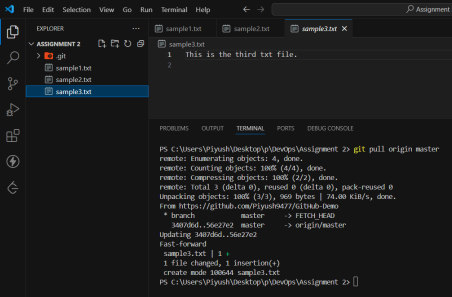


○ Pull:

git pull<remote-name> <branch-name> - This is used to pull the changes from remote repository to local repository.

Adding a new file to remote repository:

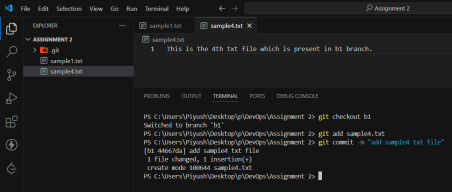


Using git pull command to get the newly added file to local repository: 

● Merge branches:

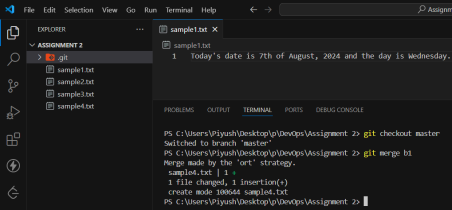
Move to another branch and add a file to it:

git checkout <branch-name> - This command switches to the <branch-name> branch.



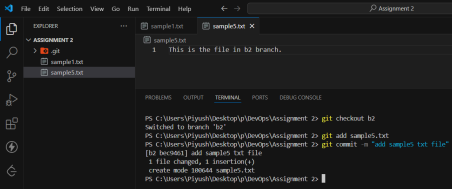
Move to master branch and merge b1 branch to it:

git merge <branch-name> - This command merges the <branch-name> branch into the master branch.

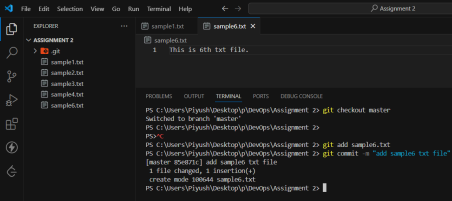


● Rebase:

Move to another branch and add a file to it:

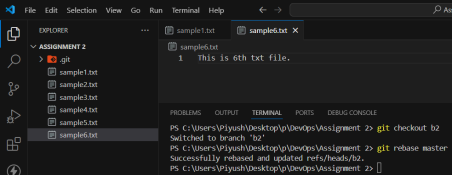


Again move to master branch and create a new file:



Now rebase b2 branch to master branch:

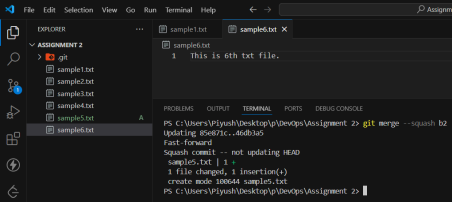
git rebase master - This command will rebase the current branch to master branch



● Squashing Commits:

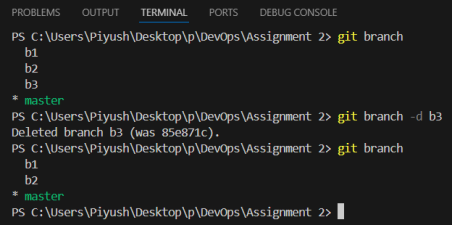
Git squash is a feature that allows developers to simplify the tree structure of a Git repository by merging multiple sequential commits.

git merge --squash - The git merge command allows users to incorporate changes from independent development lines and integrate them into a single branch. Squash commits with git merge by specifying the --squash flag to keep the master branch graph clean.



● Delete branches:

git branch -d <branch-name> - This command will delete the branch <branch-name>.



● Undo commits:

git reset --soft HEAD~<number> - This command is used to reset the current branch's head to a specific commit while preserving the changes made after that commit in the working directory and staging area.



**Conclusion:**

In this experiment, we initialised a local Git repository and linked it to a GitHub remote. We then created new branches and added some files to it. We then used different git commands to add and commit the changes and finally to push those files to our remote repository. We then merged two branches and also we saw how to rebase branches. We squashed commit using merge and also deleted some branches. We also learnt how to undo a commit using git reset command.