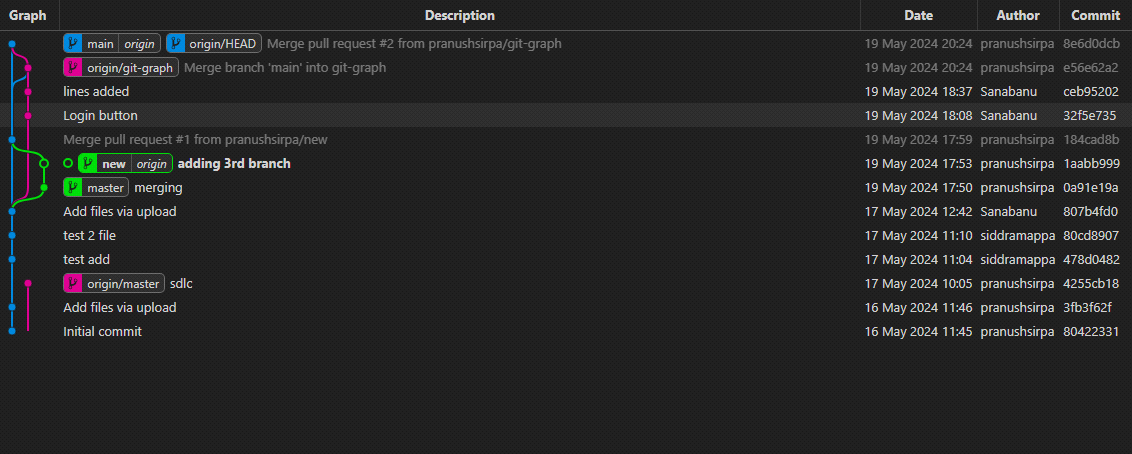
Assignment 2: Version Control with Git - Initialize a new Git repository, simulate making changes to a project by adding text files, and commit those changes.

Create a new branch, make further changes in this branch, and merge it back to the main branch, resolving a simulated merge conflict.

SOLUTION:



Initialize a new Git repository:

bash

Copy code

# Navigate to your project directory

cd /path to our project/ (Changing the directory)

# Initialize a new Git repository

git init

Simulate making changes to a project by adding text files and committing those changes:

bash

Copy code

# Create some text files

echo "This is file1.txt" > file1.txt

echo "This is file2.txt" > file2.txt

# Add the files to the staging area

git add file1.txt file2.txt

# Commit the changes

git commit -m "Initial commit with file1.txt and file2.txt"

Create a new branch and make further changes in this branch:

bash

Copy code

# Create a new branch

git branch new-feature

# Switch to the new branch

git checkout new-feature

# Make changes in the new branch (for example, modify file1.txt)

echo "This is an additional line in file1.txt" >> file1.txt

# Add the modified file to the staging area

git add file1.txt

# Commit the changes

git commit -m "Added an additional line to file1.txt"

Merge the new branch back to the main branch and resolve a simulated merge conflict:

bash

Copy code

# Switch back to the main branch

git checkout main

# Merge the new-feature branch into main

git merge new-feature

# If there's a merge conflict, resolve it by editing the conflicted files

# After resolving conflicts, add the conflicted files to the staging area

git add <conflicted files>

# Then commit the merge

git commit -m "Merged new-feature branch, resolved conflicts"