**LAB 2**

**NAME:** Hajra Sarwar **REG NO:** 2023-BSE-022  
**CLASS:** BSE-5A

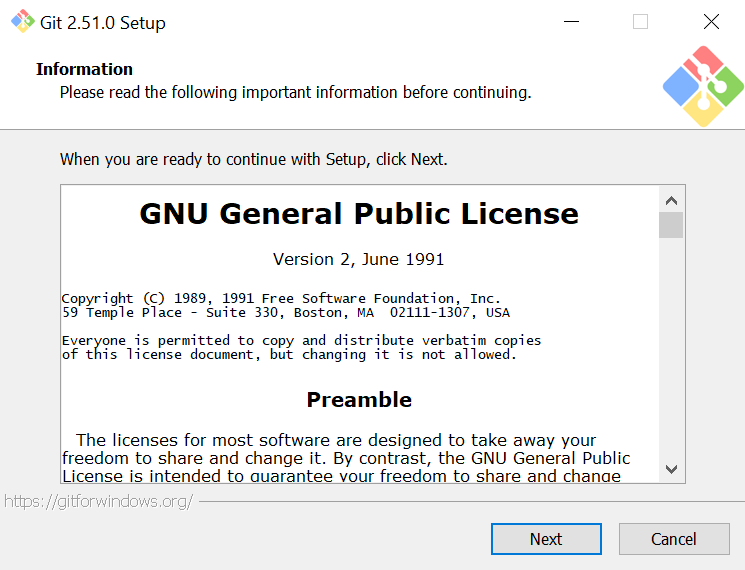
**SUBJECT:** CC

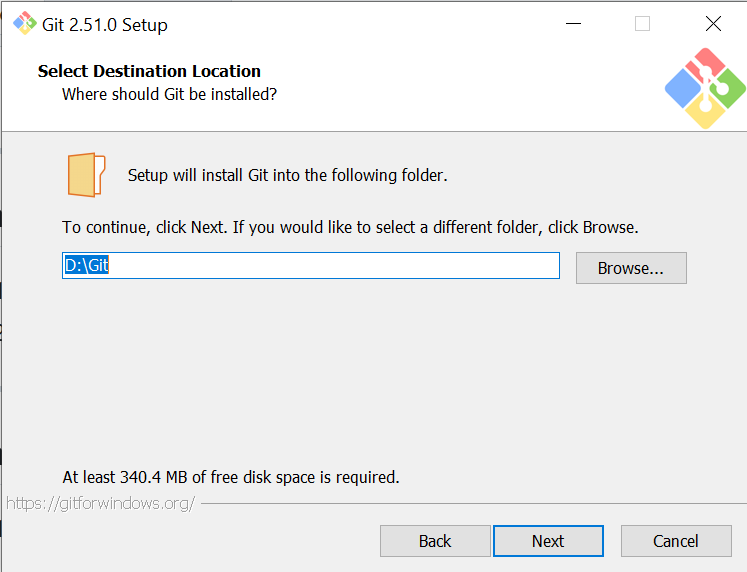
**LAB TITLE**: Hands-On Git & Version Control Lab

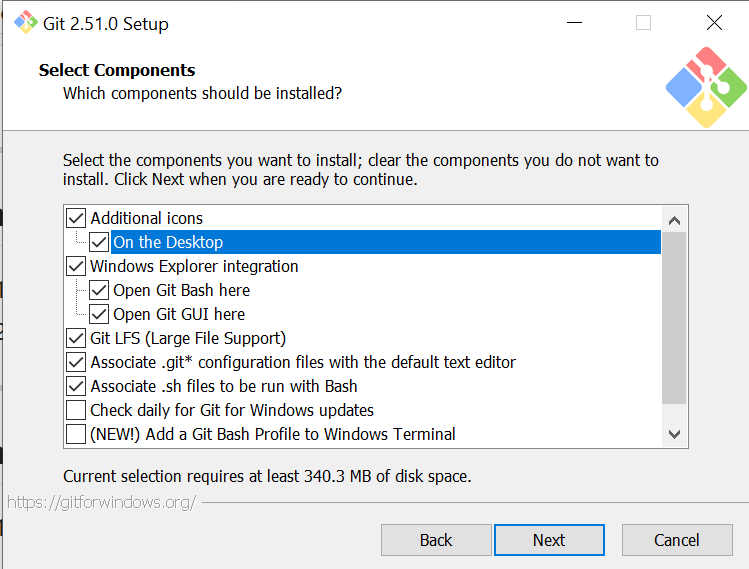
**TASK:**

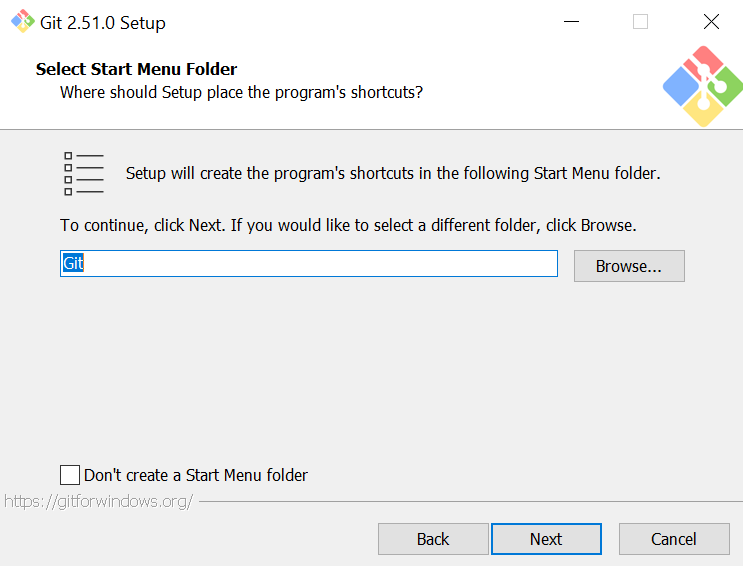
1. **Install Git on your PC.**

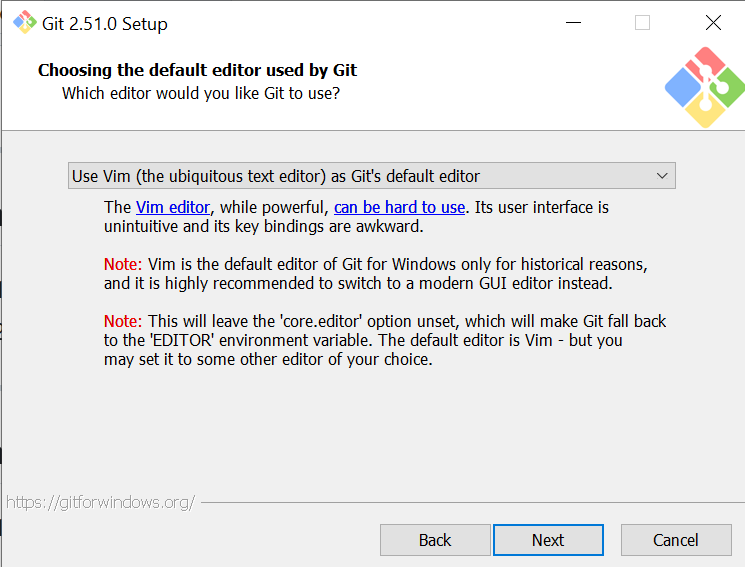
You can download this from: [Download Git for Windows, Mac, or Linux](https://git-scm.com/downloads)

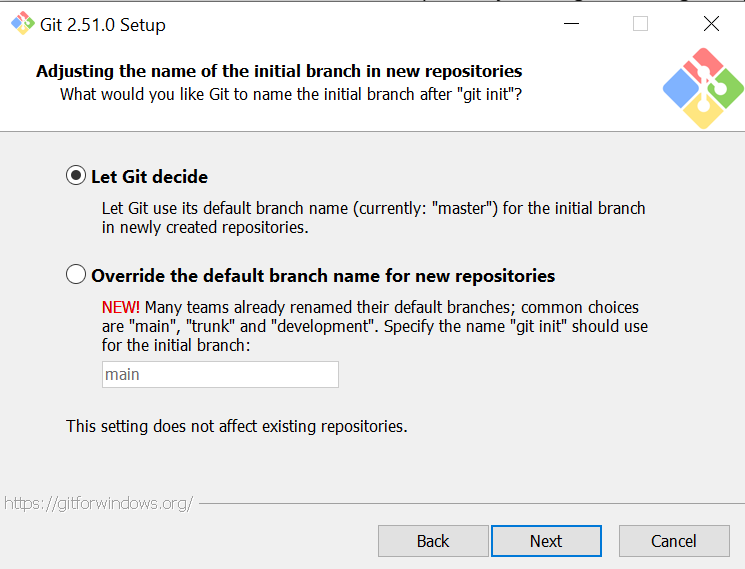


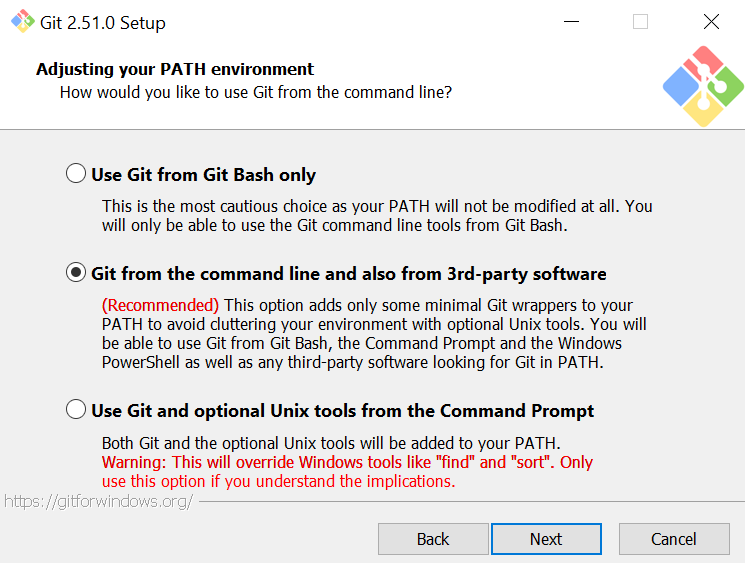


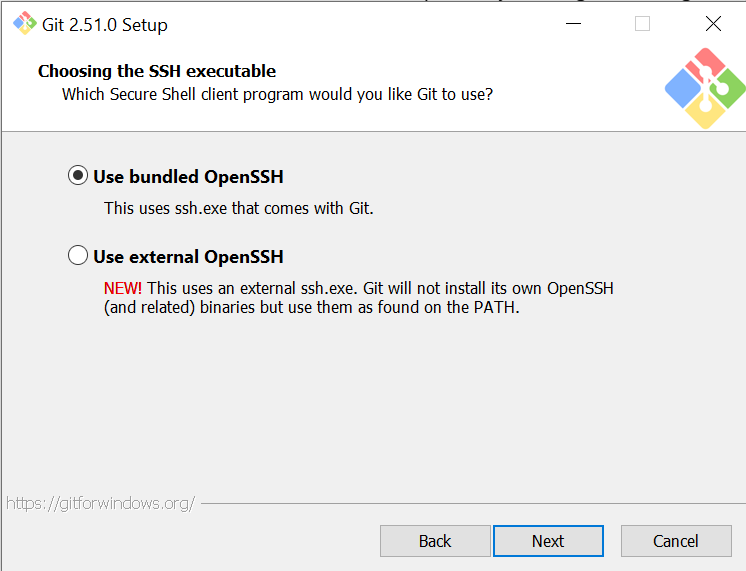


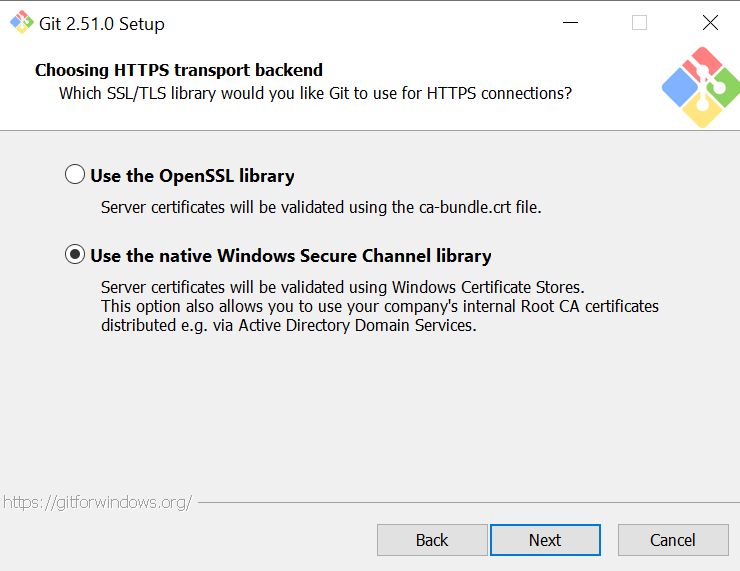


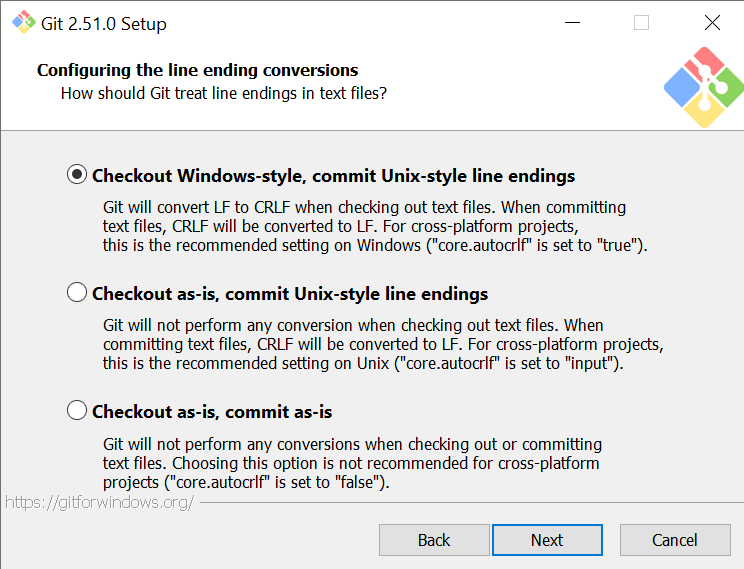


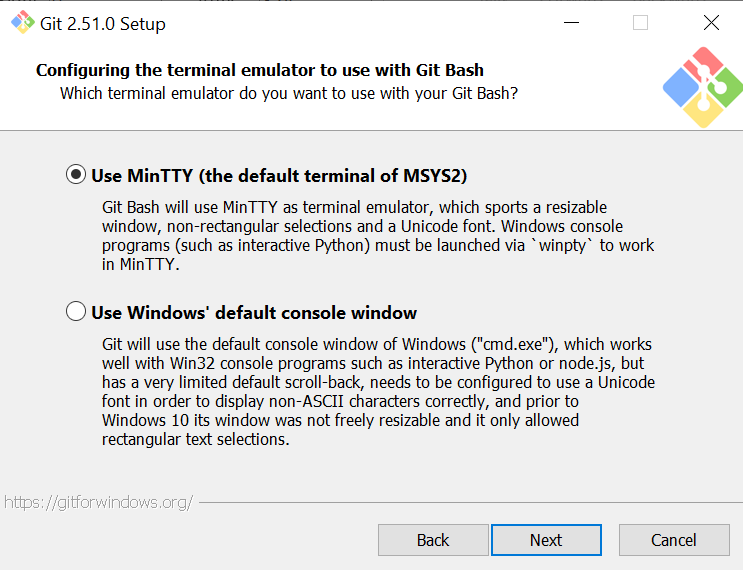


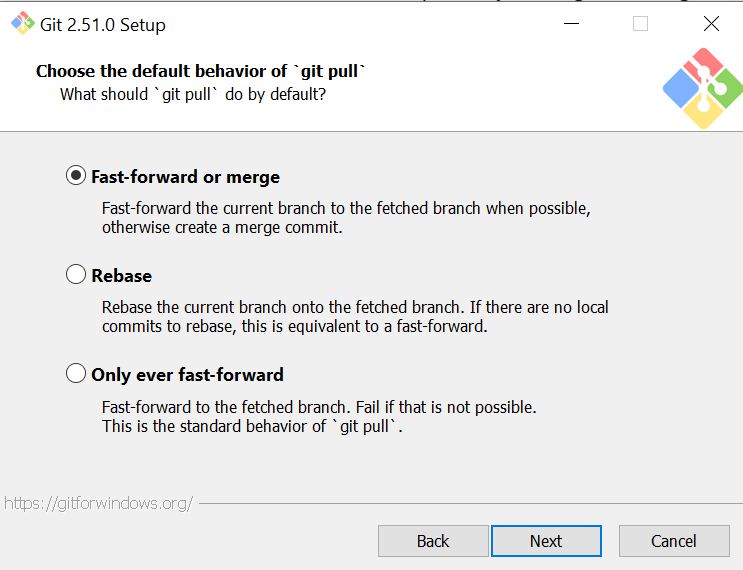


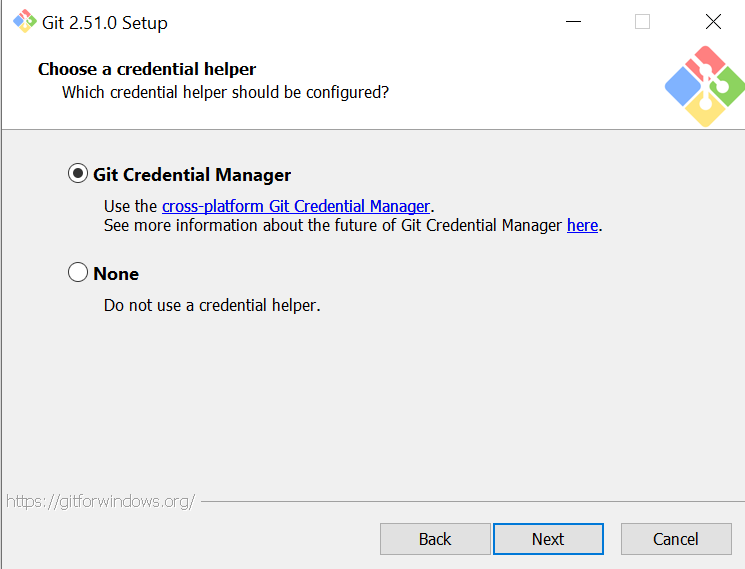




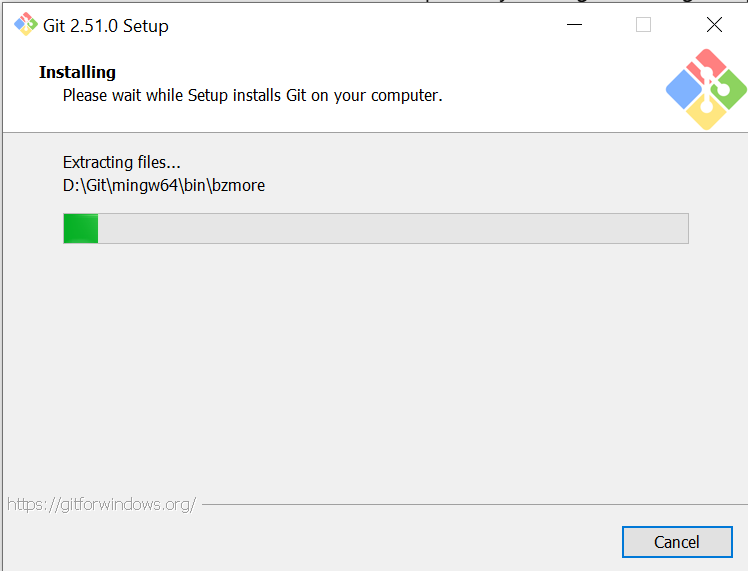


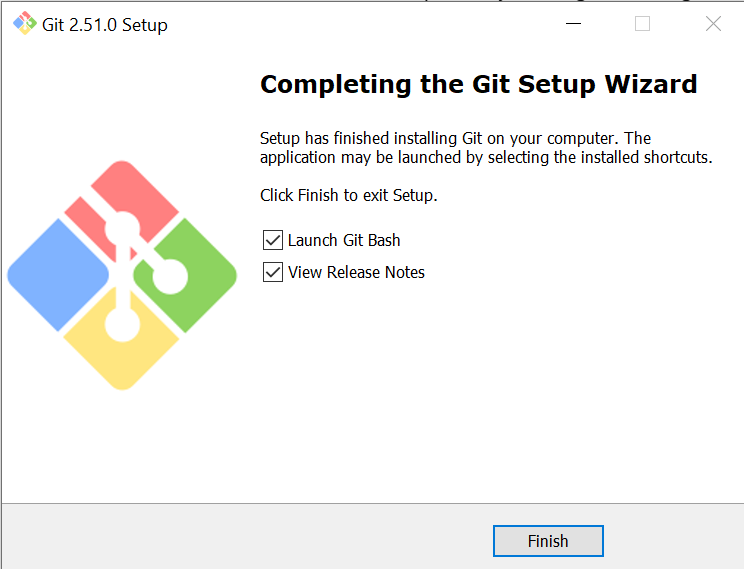






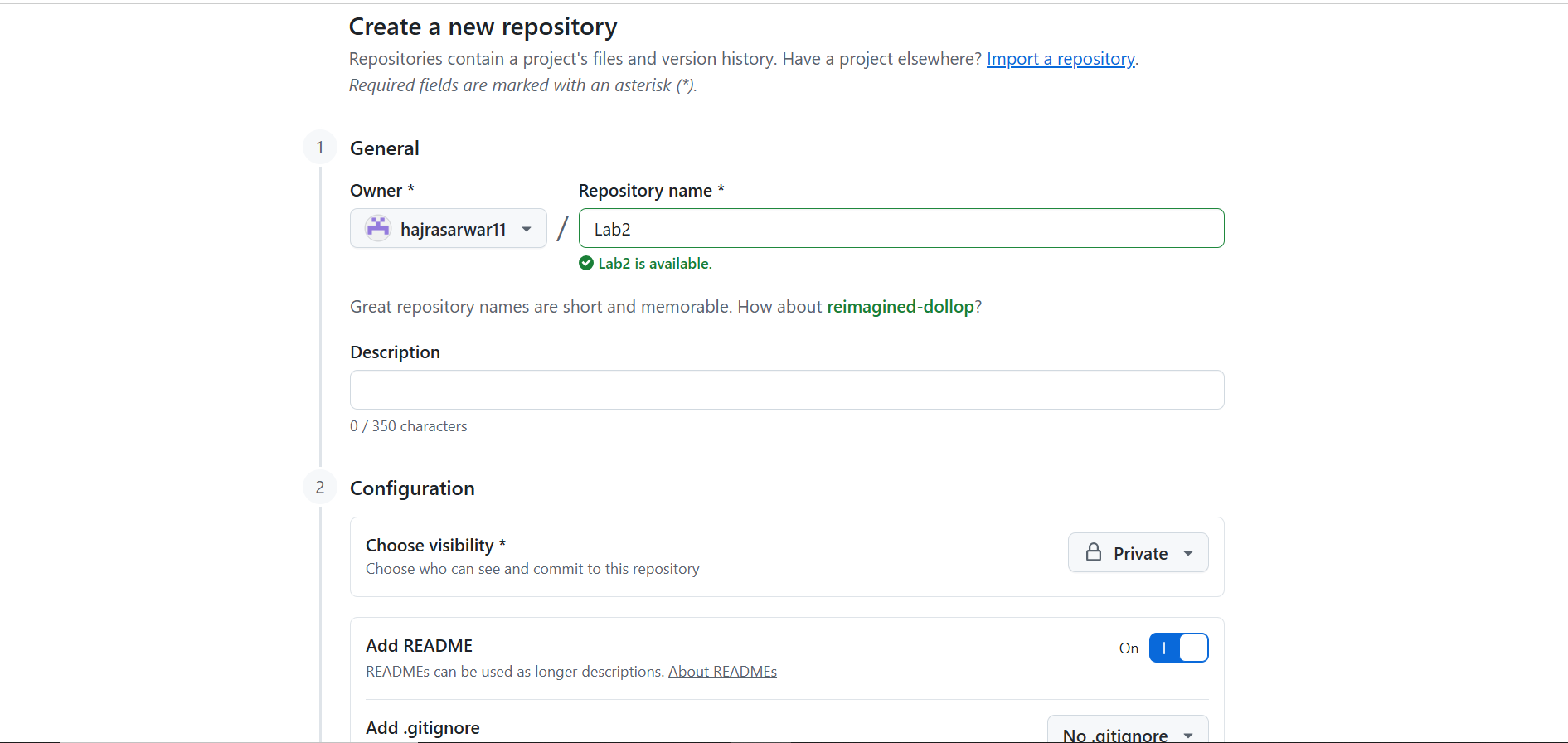






**Task 1: Create Private GitHub Repository**

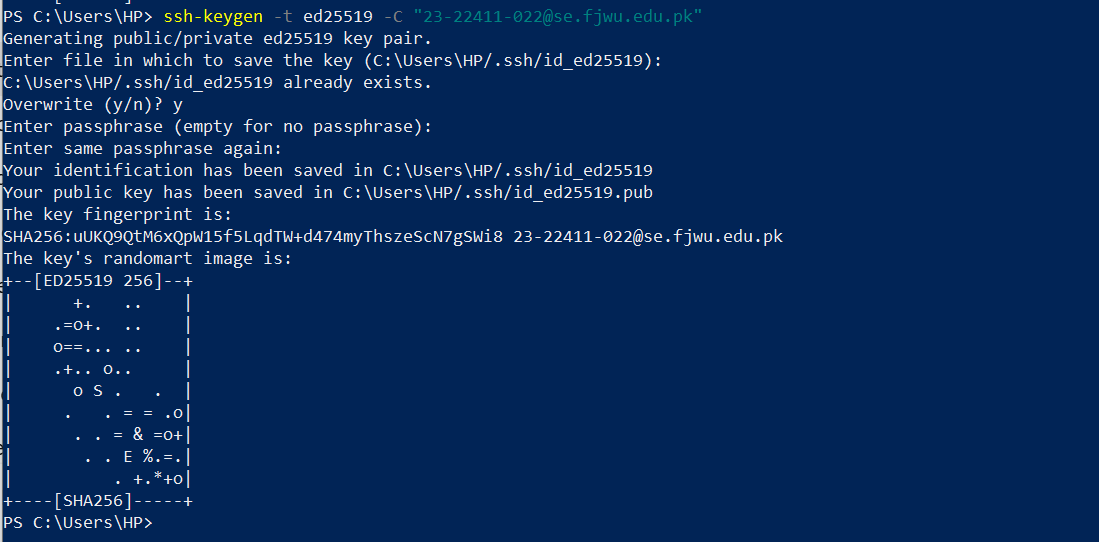
Create a new private repository named Lab2 on GitHub.

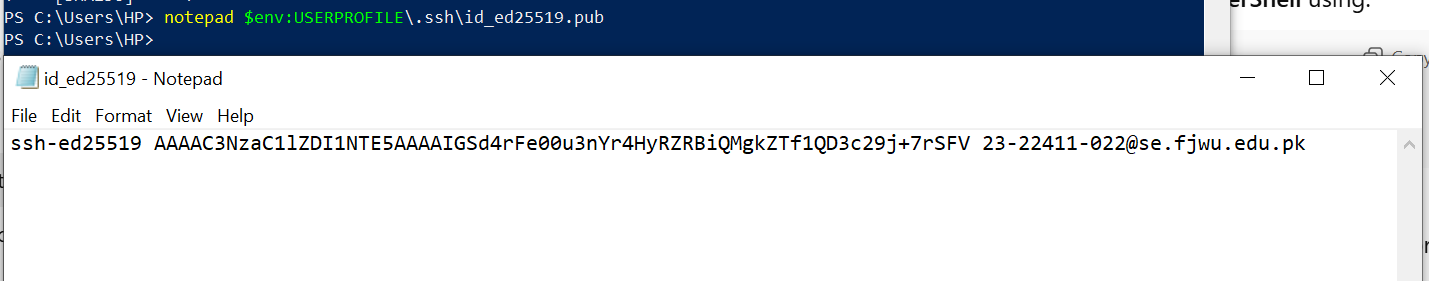


**Task 2: Connect Repository via SSH**

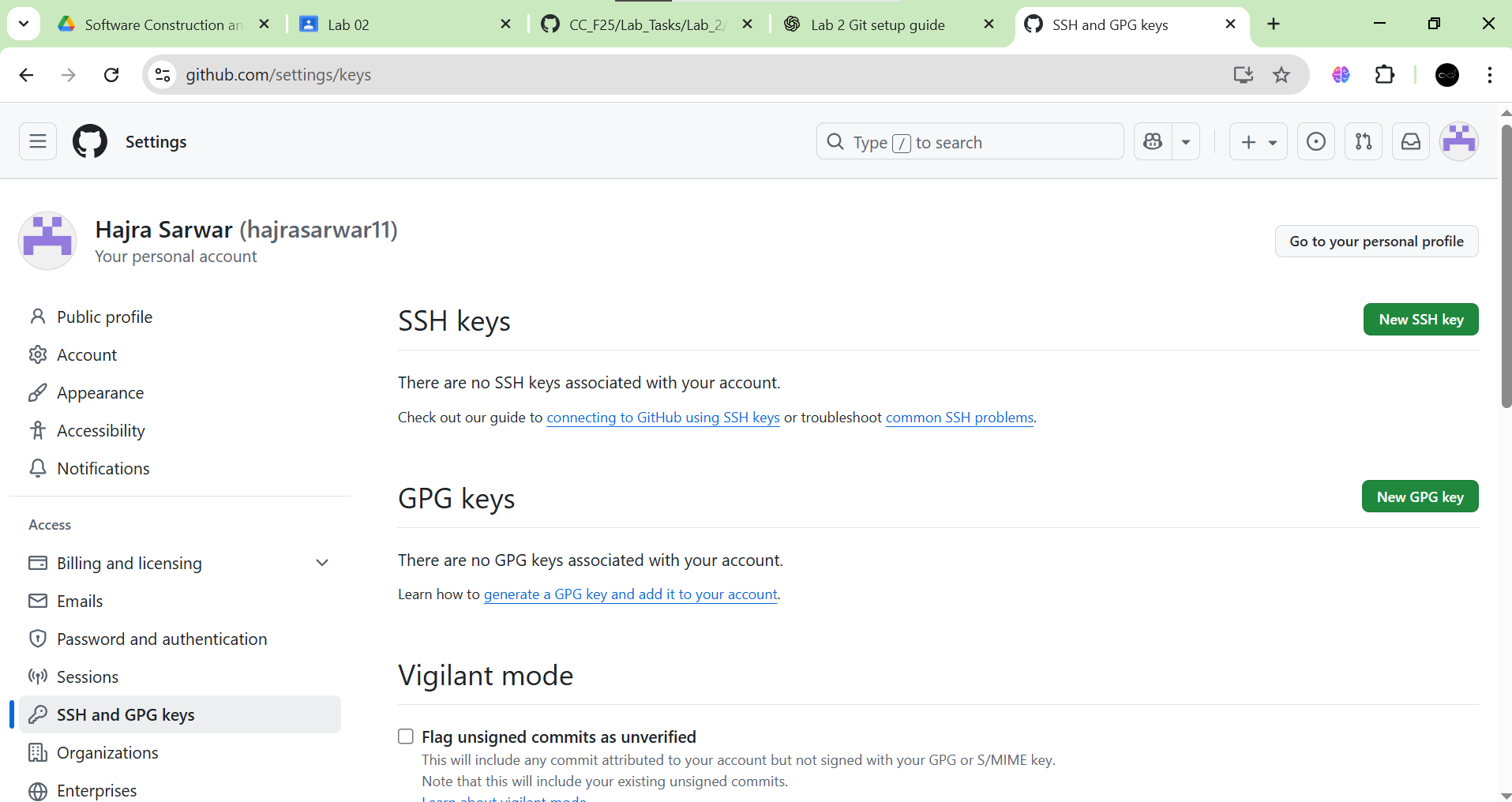
1. **Generate a new SSH key using PowerShell:**

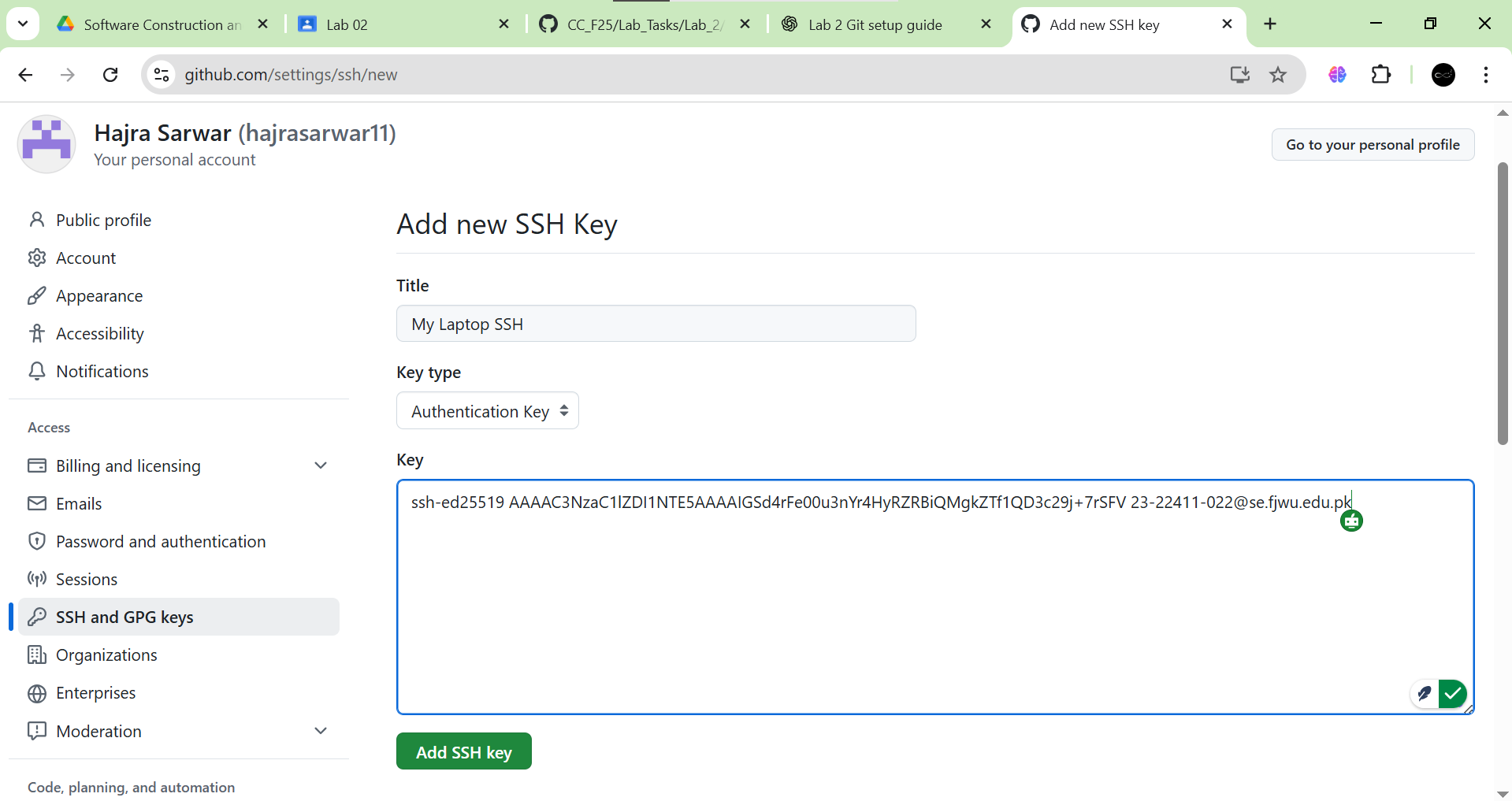
ssh-keygen -t ed25519 -C [your\_email@example.com](mailto:your_email@example.com)

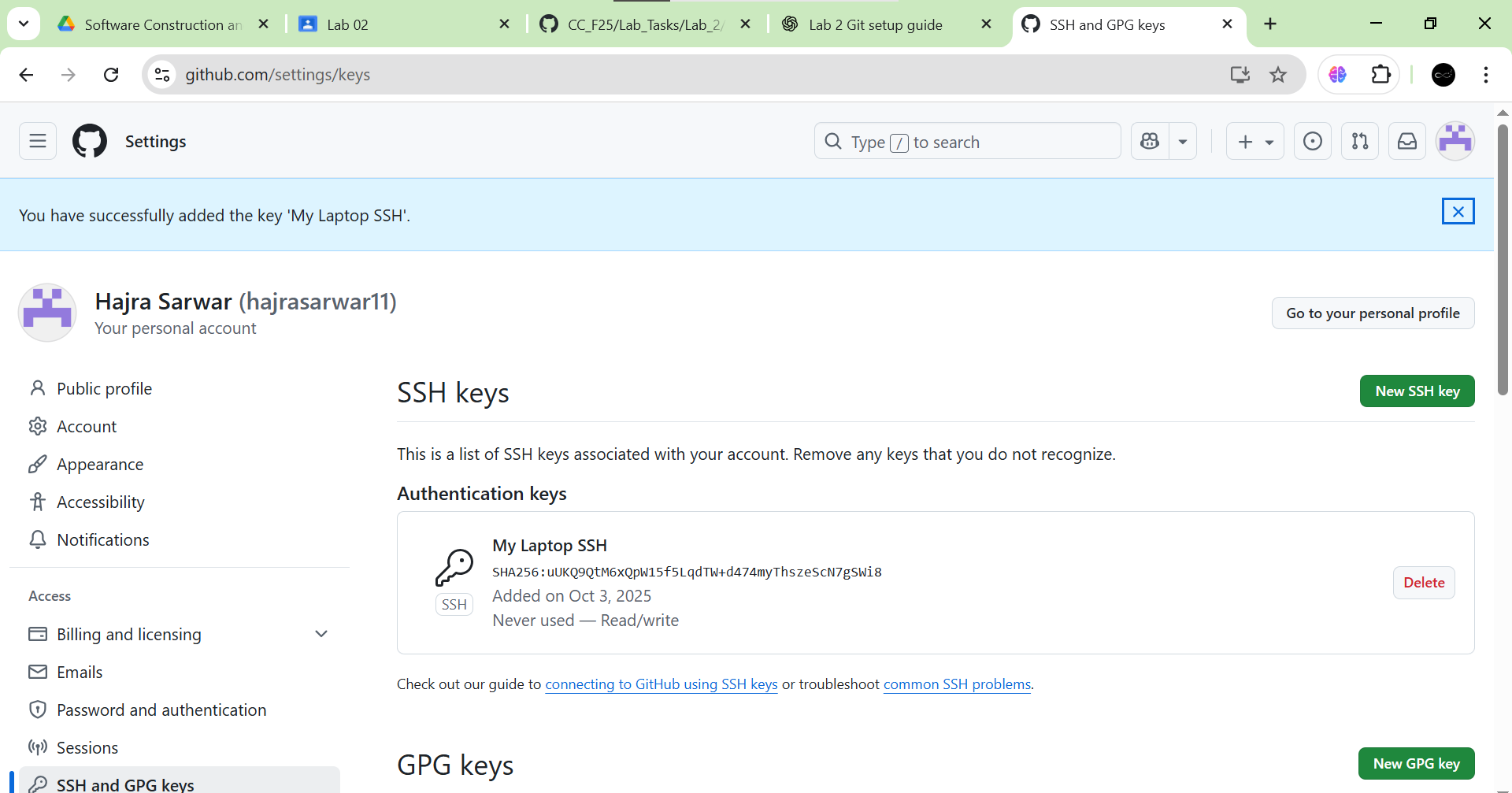




1. **Add your SSH public key to GitHub (Settings > SSH and GPG keys).**

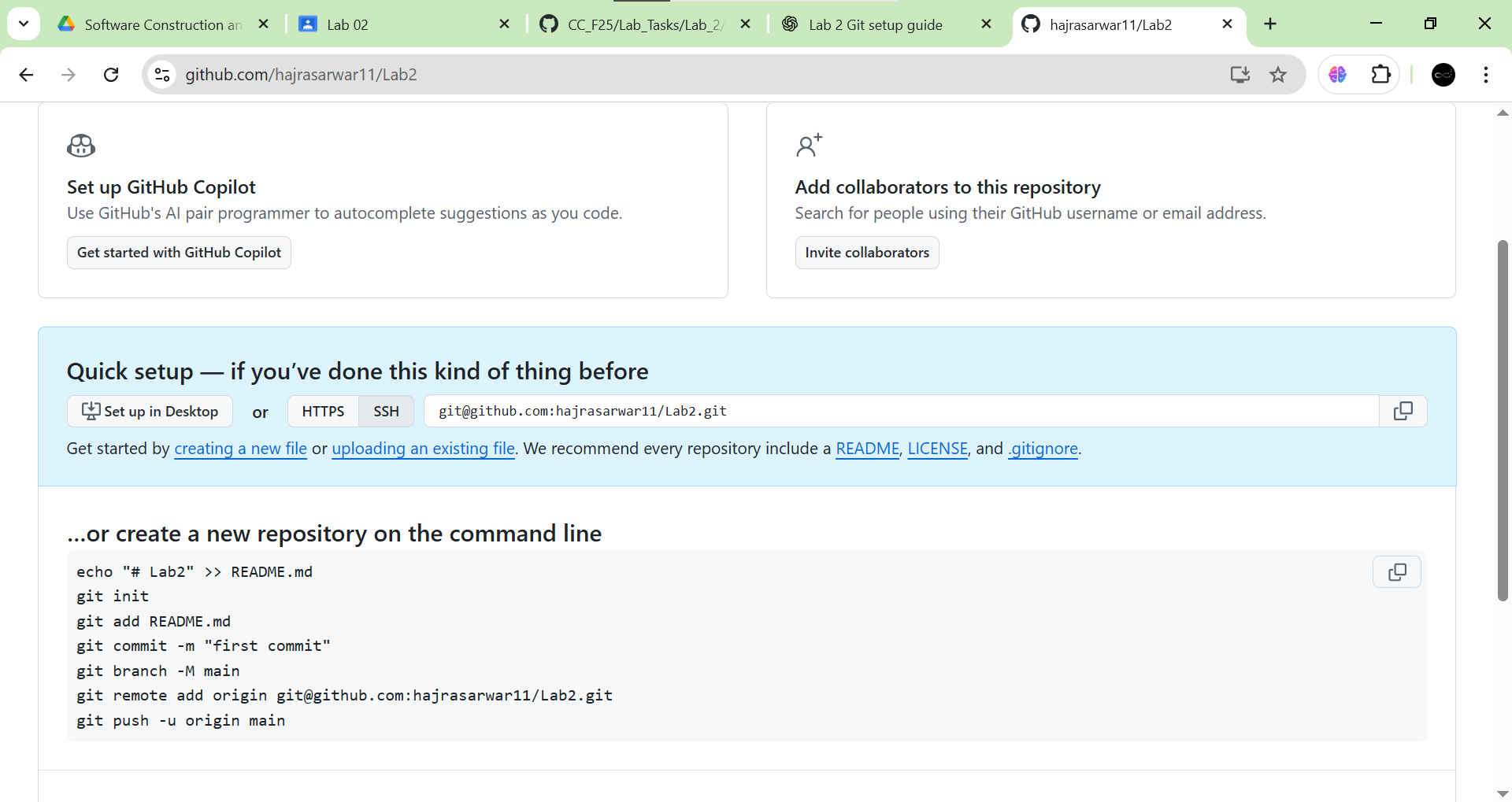


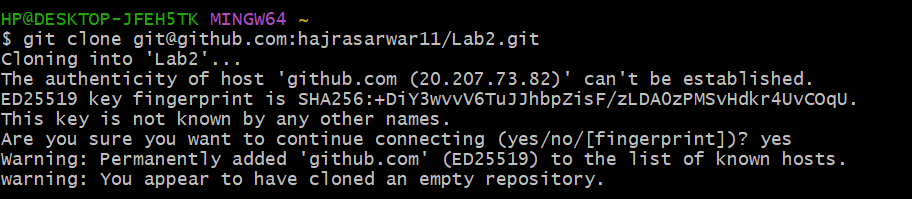




1. **Clone your Lab2 repo using SSH.**

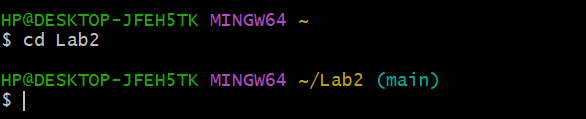
git clone git@github.com:<yourusername>/Lab2.git



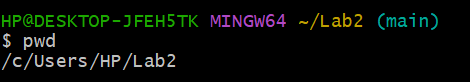


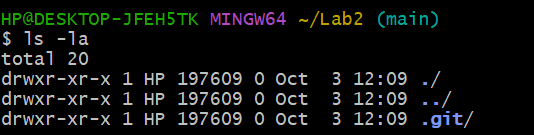
**Task 3: Explore the .git Folder**

1. Navigate into your cloned repository folder.

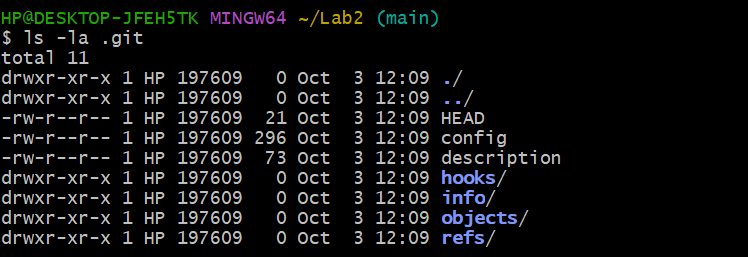


1. Show hidden files and locate the .git directory.





1. Explore what’s inside using: ls -a .git



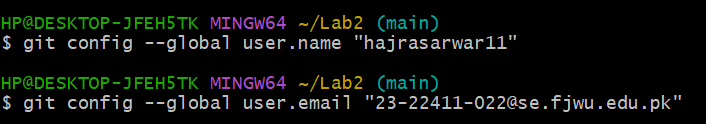
Note: This folder contains history, logs, and config for your repository. Do **not** modify these files manually.

**Task 4: Configure Git Username and Email**

1. **Set up your Git identity (this ensures all commits are linked to you):**

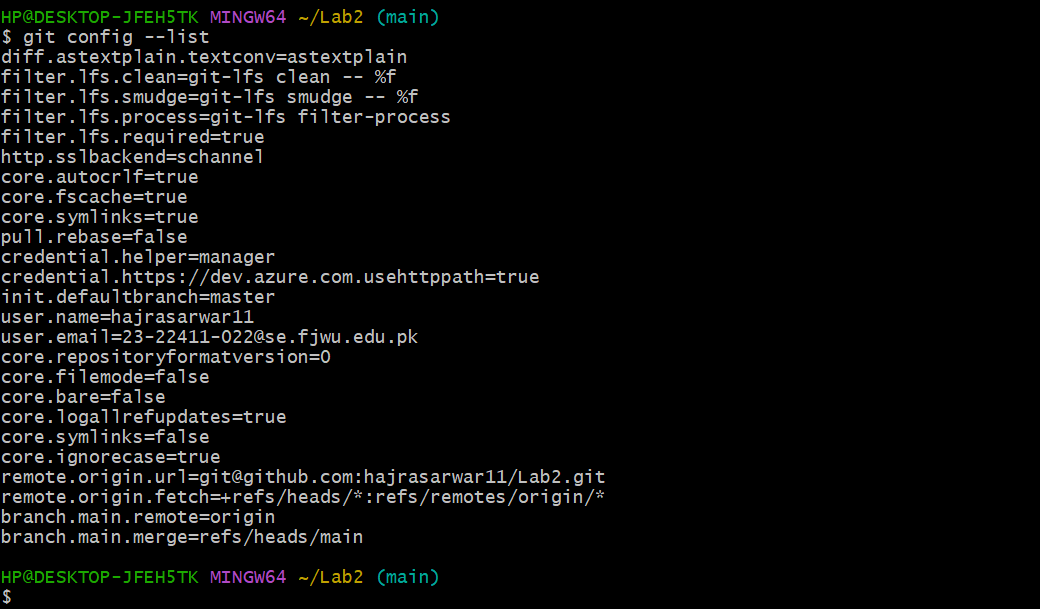
git config --global user.name "Your Name"

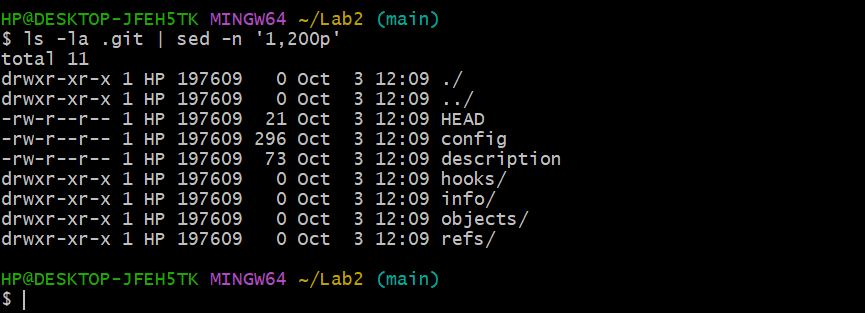
git config --global user.email "your\_email@example.com"

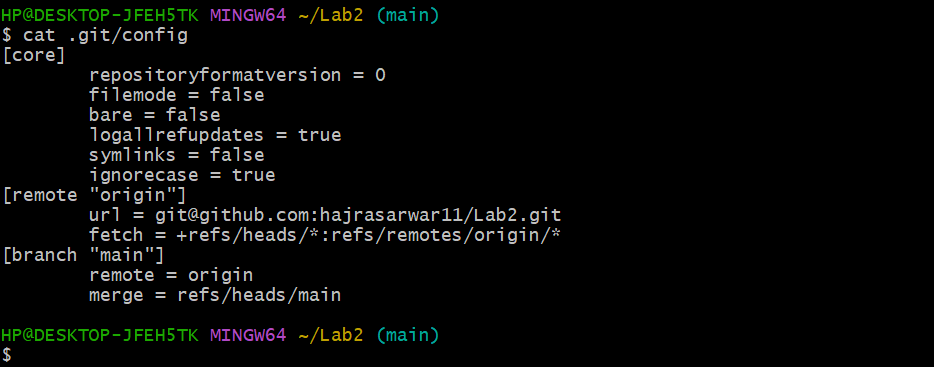


1. **Verify your configuration:**

git config --list



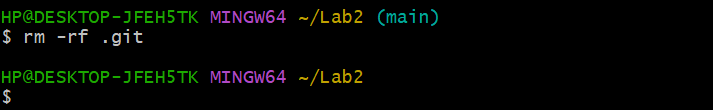




**Task 5: Local Repository Management**

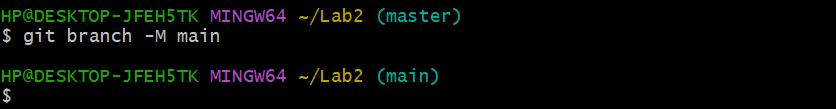
1. **Delete the existing .git folder from your cloned repo using Git Bash:**

rm -rf .git



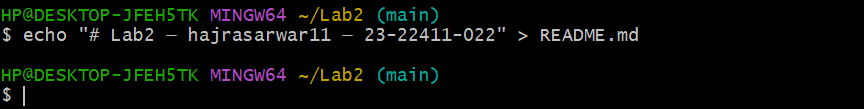
1. **Re-initialize the local git repository:**

git init

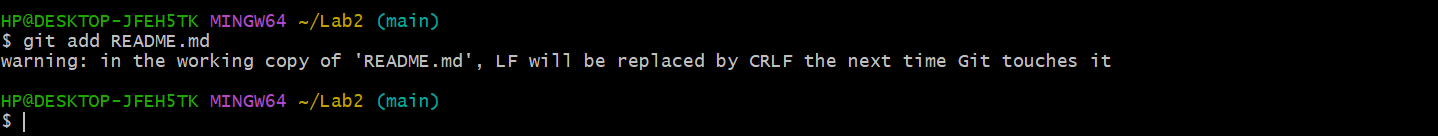


1. **Add a file named README.md and commit it:**

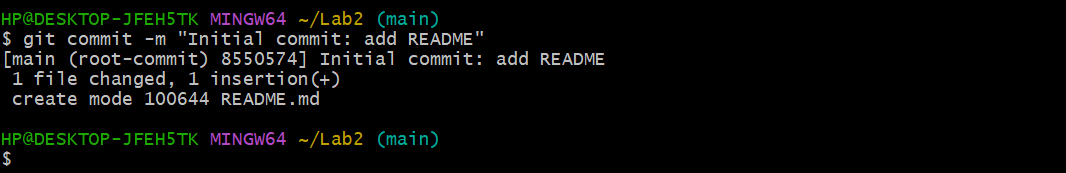
echo "# Lab2 Git Practice" > README.md



git add README.md

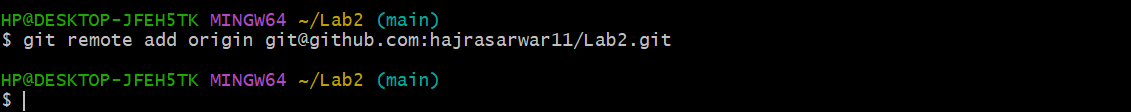


git commit -m "Initial commit"

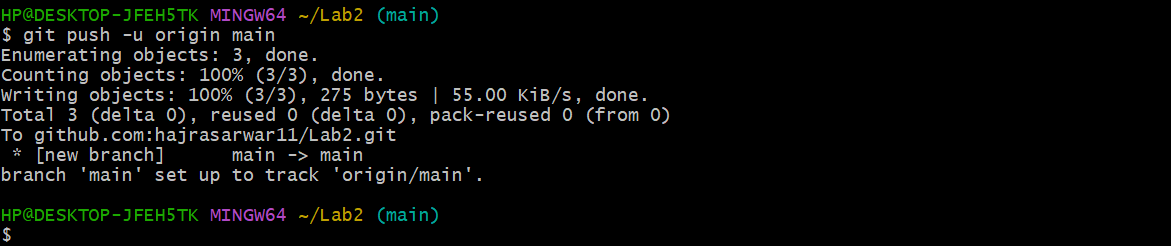


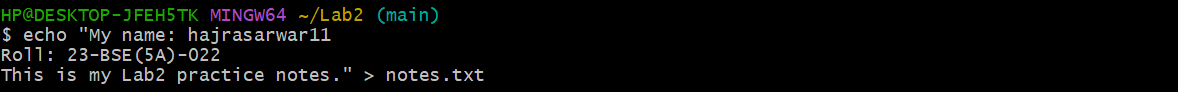
1. **Connect your local repo to GitHub and push:**

git remote add origin [git@github.com:<yourusername>/Lab2.git](mailto:git@github.com:%3cyourusername%3e/Lab2.git)



git push -u origin main

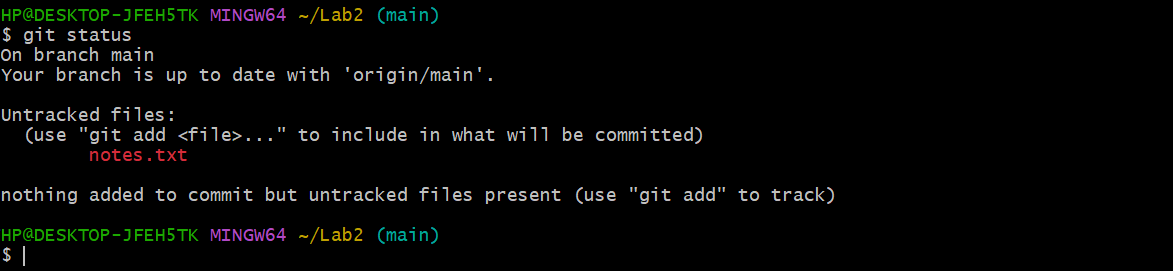




**Task 6: File Status & Staging**

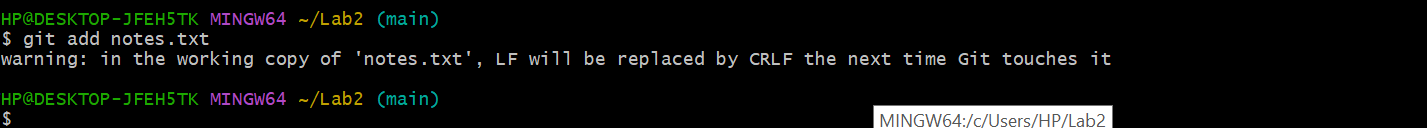
1. Create a new file notes.txt and write a note.
2. Check status:

git status

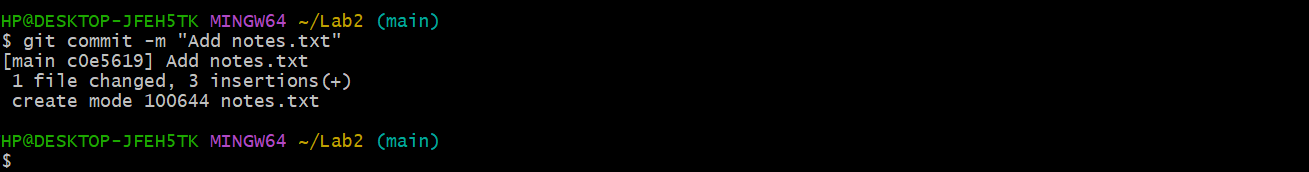


1. Stage and commit:

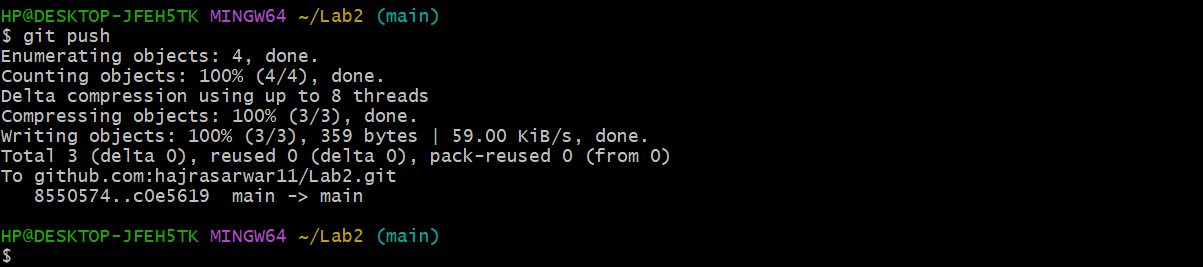
git add notes.txt



git commit -m "Add notes.txt"

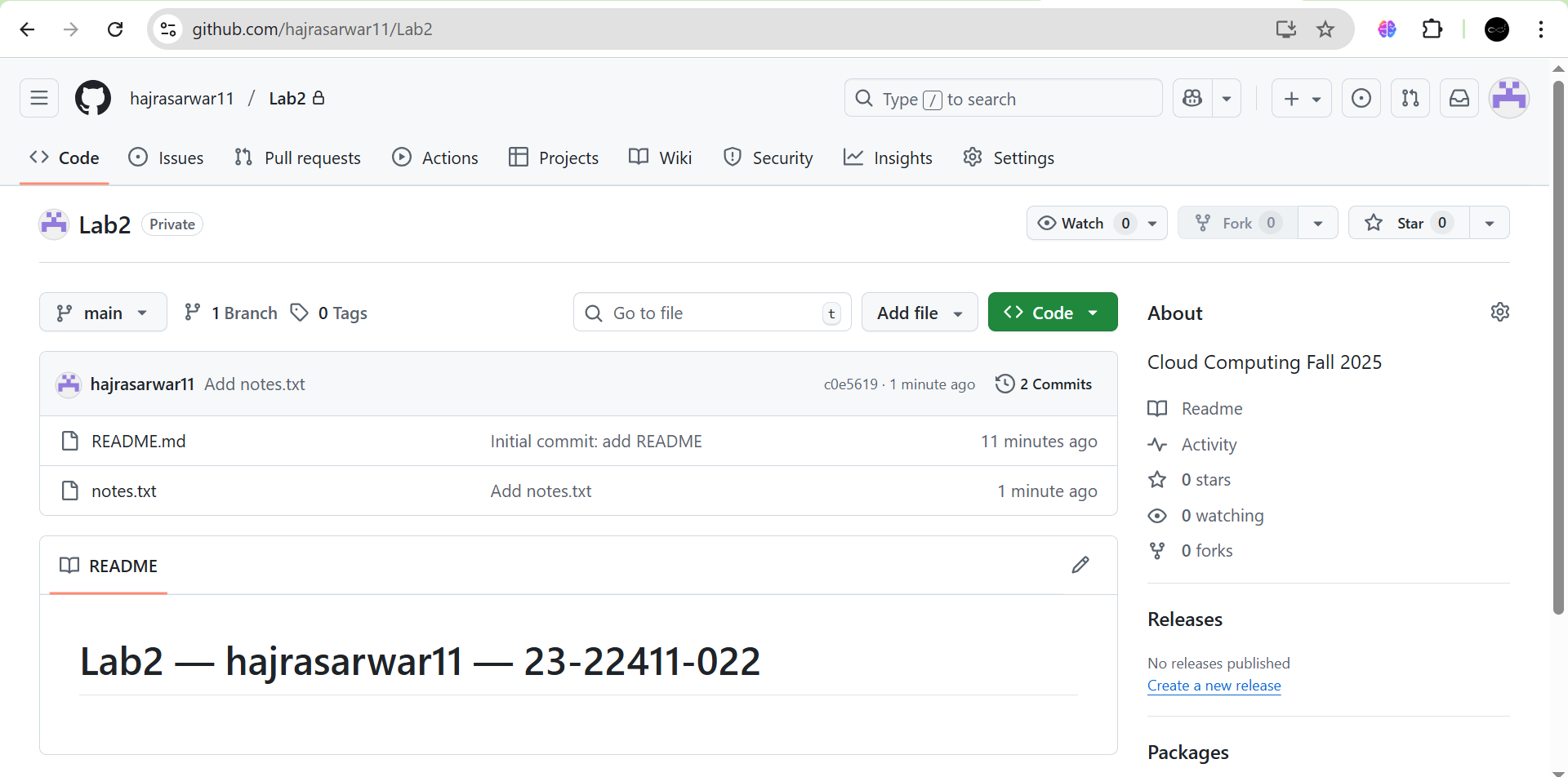


1. Edit notes.txt and repeat status/commit steps.



**Task 7: Branch Creation Using GitHub GUI**

1. **On GitHub (web interface), create a branch named bugfix/user-auth-error.**







1. **Pull the branch to your local repository to sync.**

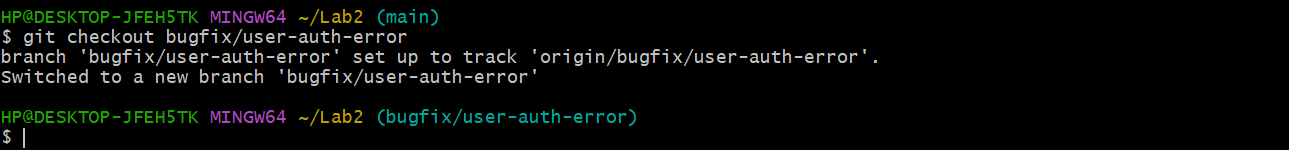
git pull origin bugfix/user-auth-error

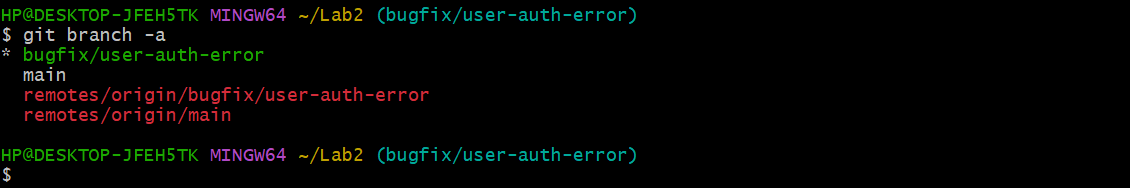


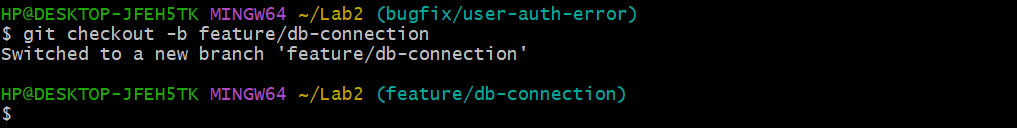
**Task 8: Branch Creation and Push Using Git Bash**

1. **Create a branch named feature/db-connection using Git Bash:**

git checkout -b feature/db-connection

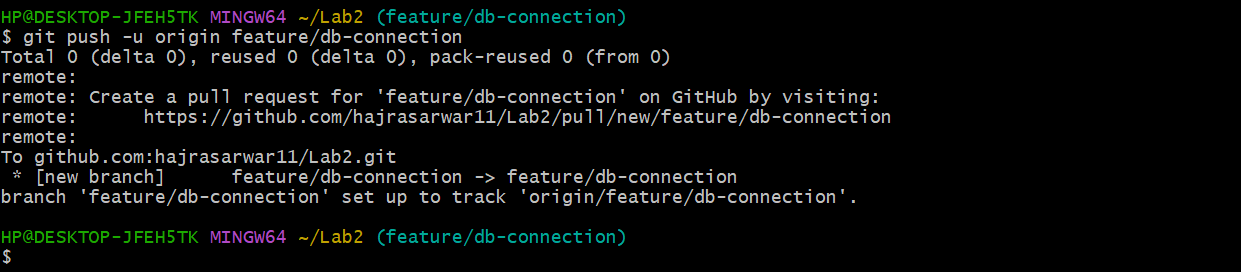


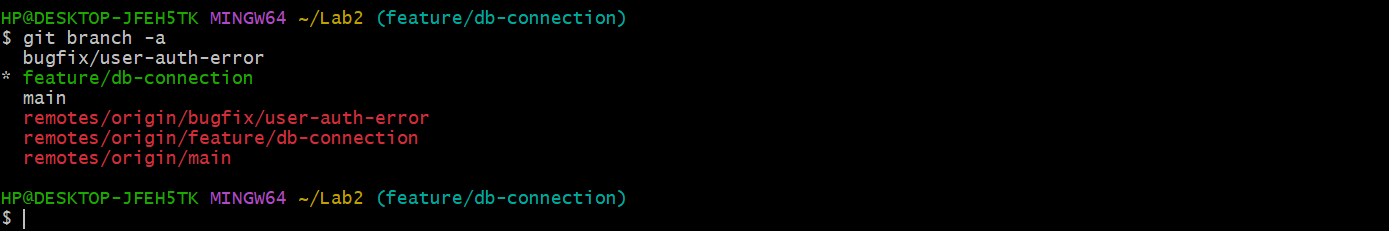


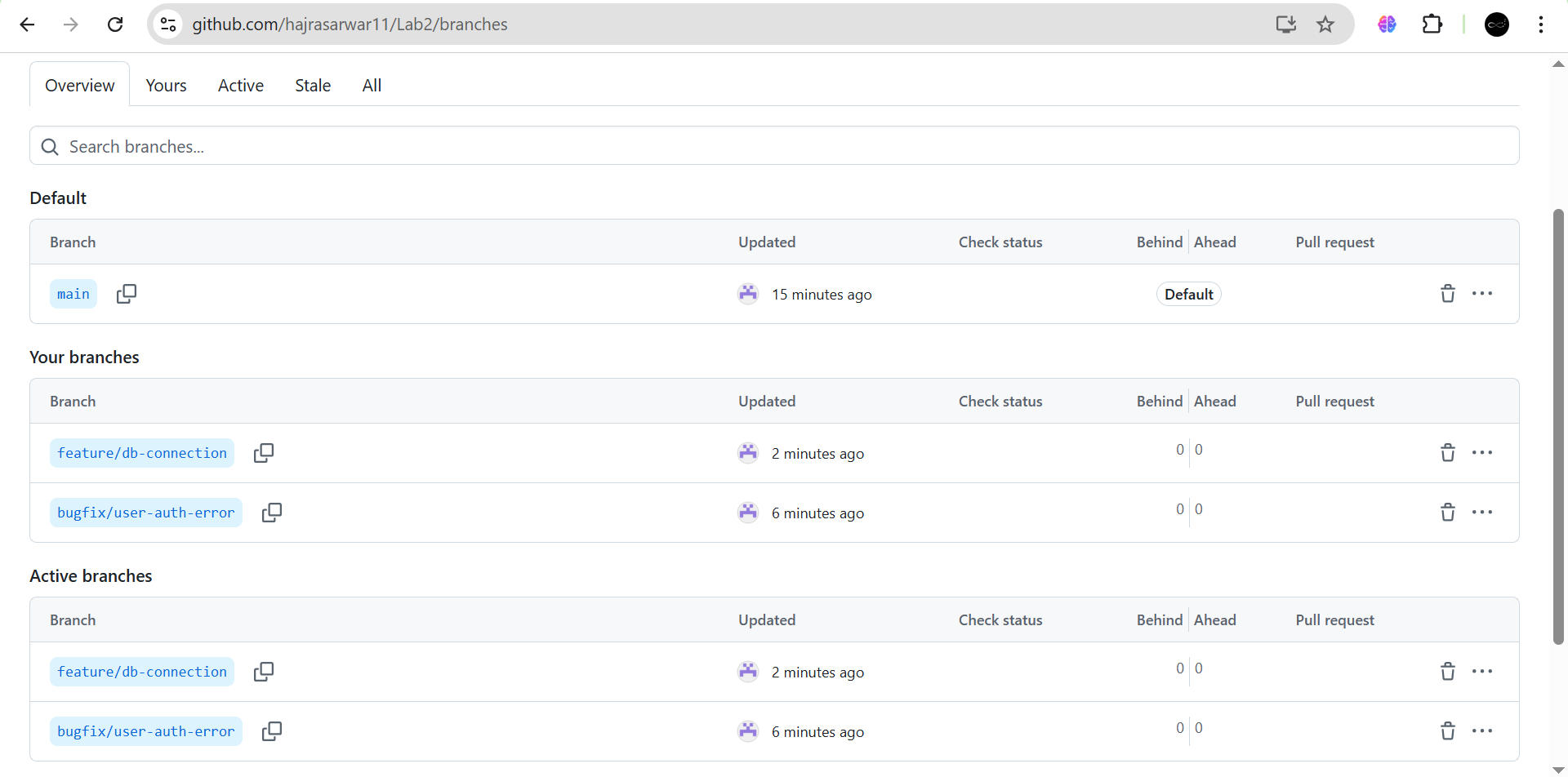


1. **Push the branch to the remote repository:**

git push origin feature/db-connection



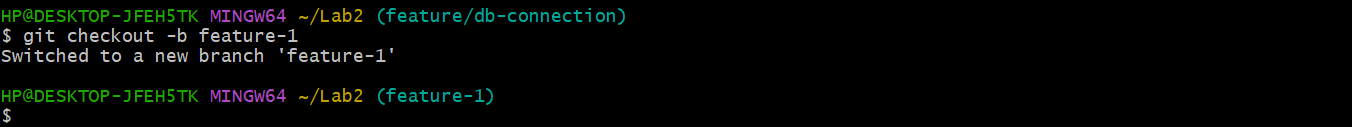


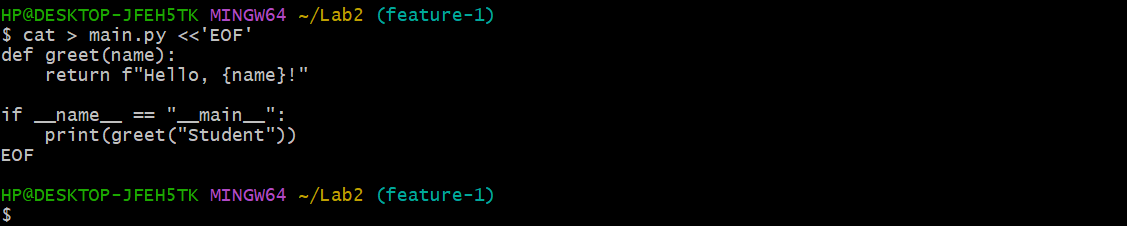


**Task 9: Branching & Merging**

1. Create and switch to a branch feature-1:

git checkout -b feature-1

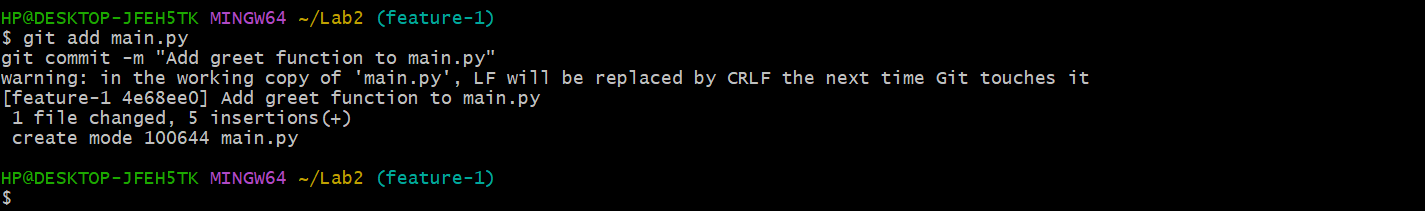




1. Modify main.py (add a function) and commit.

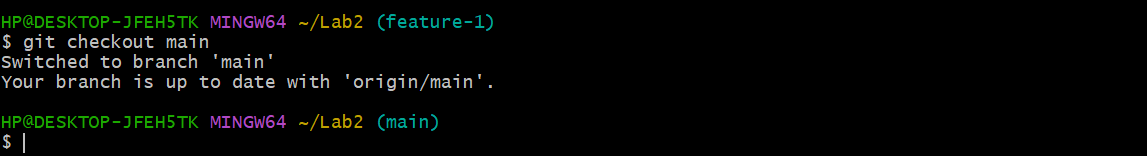
git add main.py

git commit -m "Add new function to main.py"

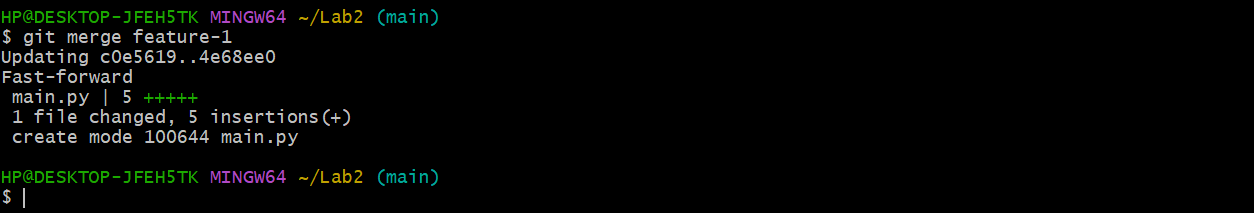


1. Switch back to main and merge:

git checkout main



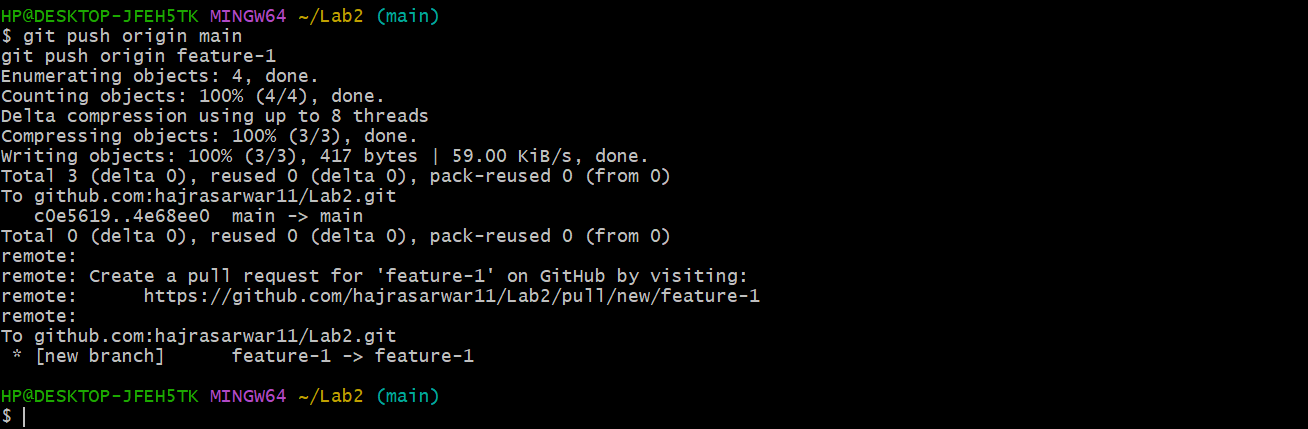
git merge feature-1

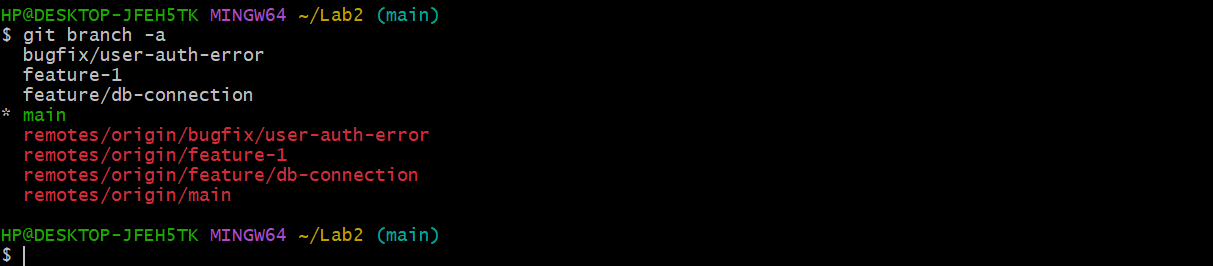


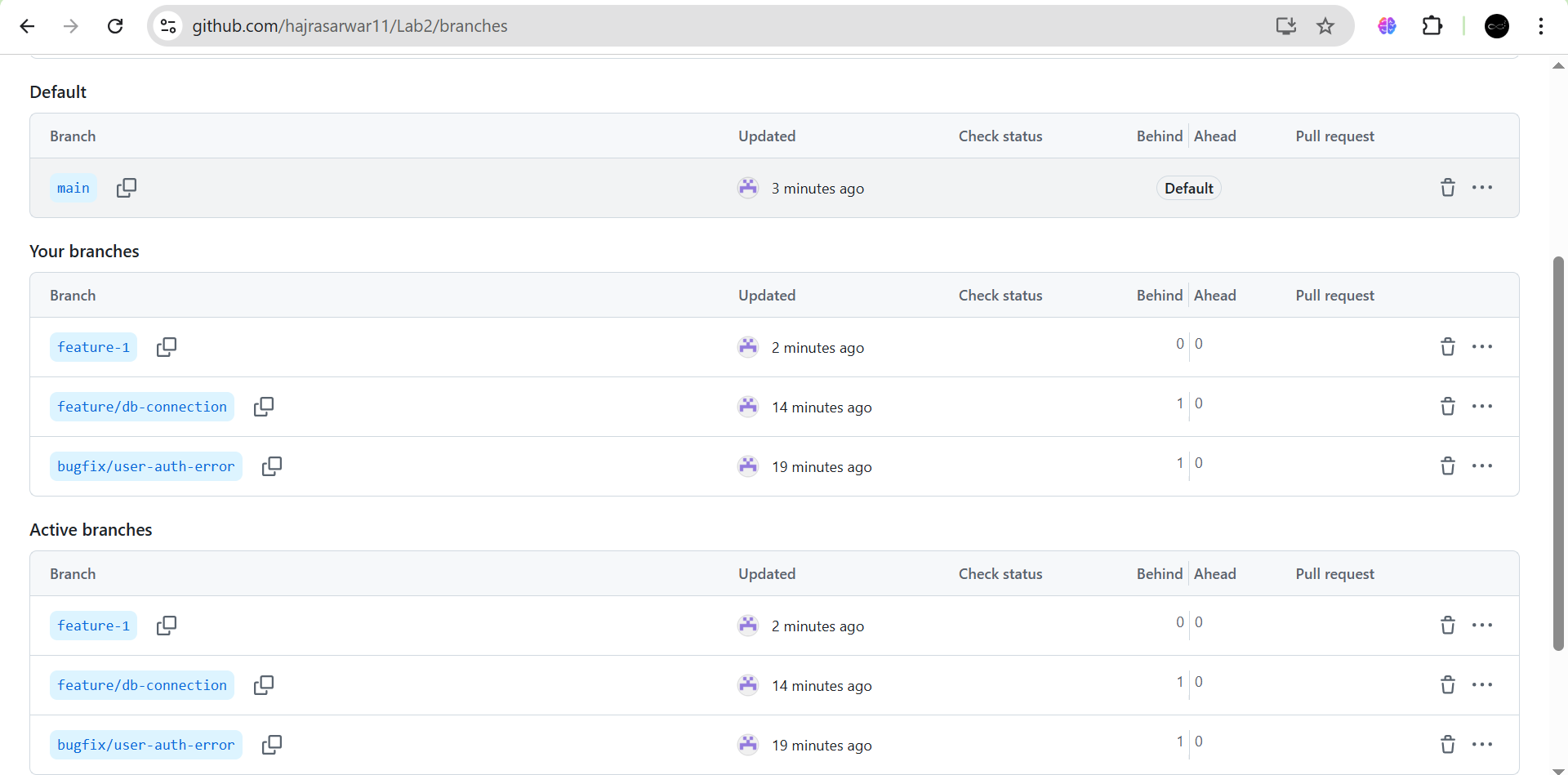
1. Push all branches:

git push origin main

git push origin feature-1

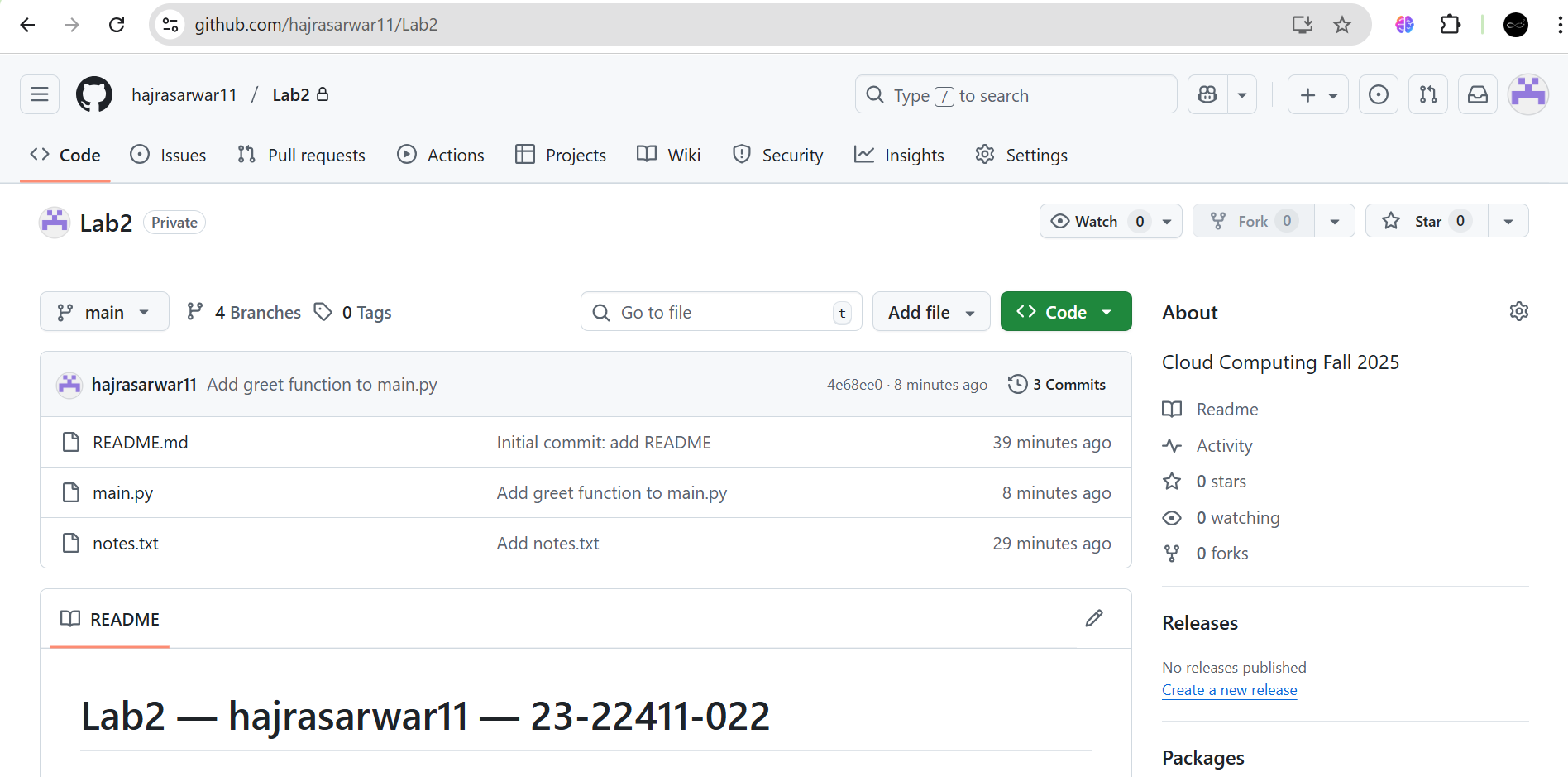


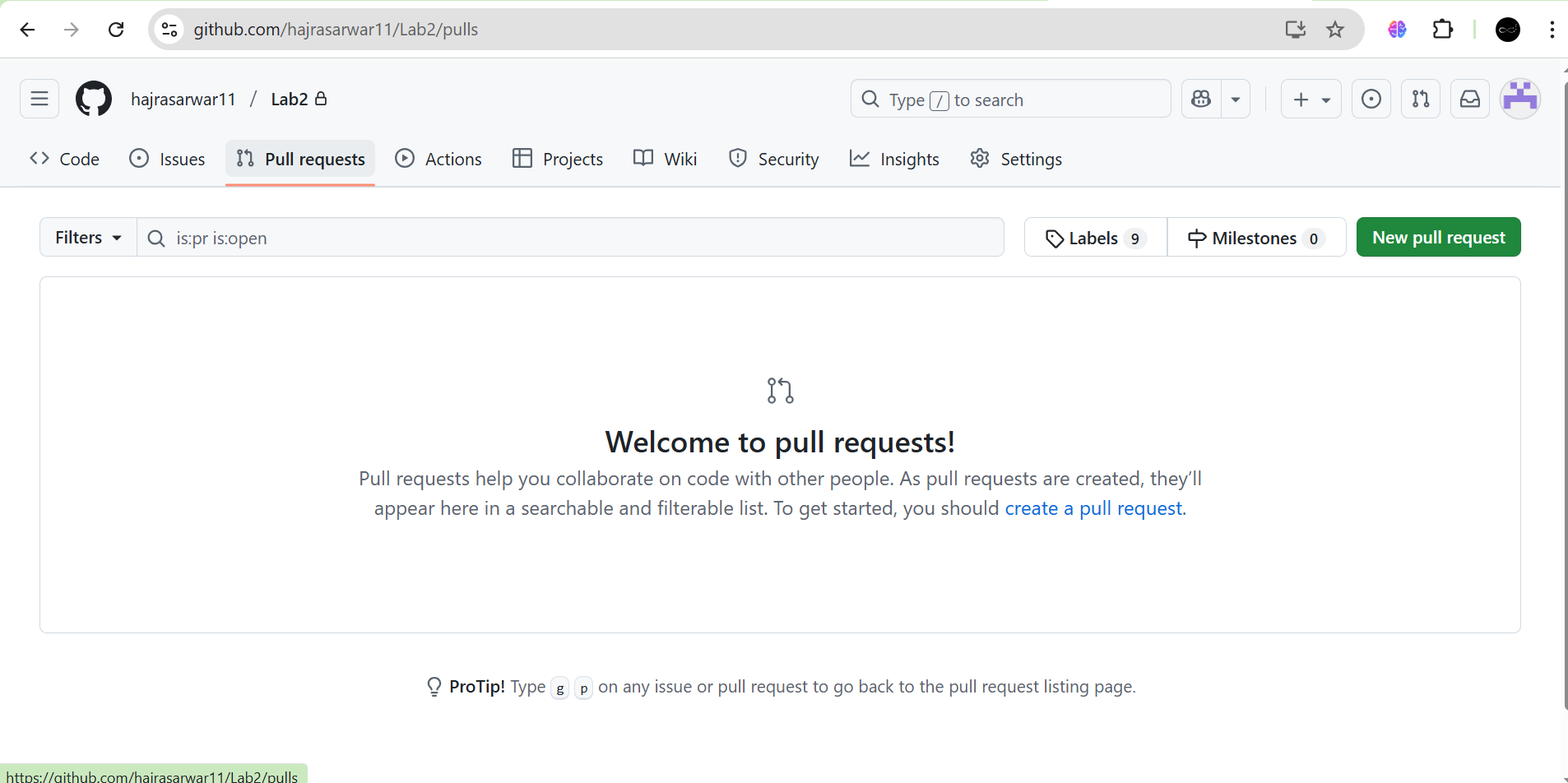


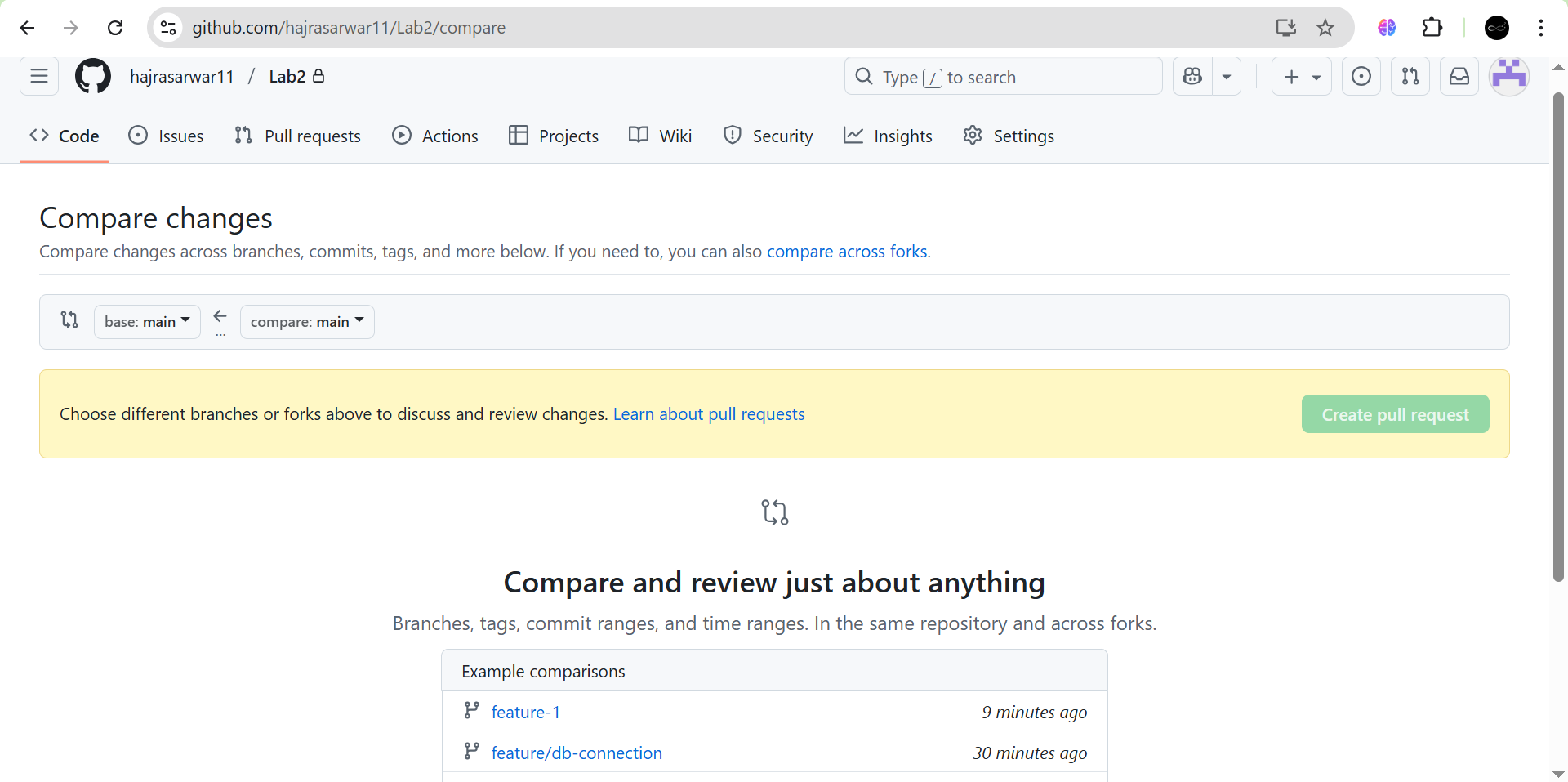


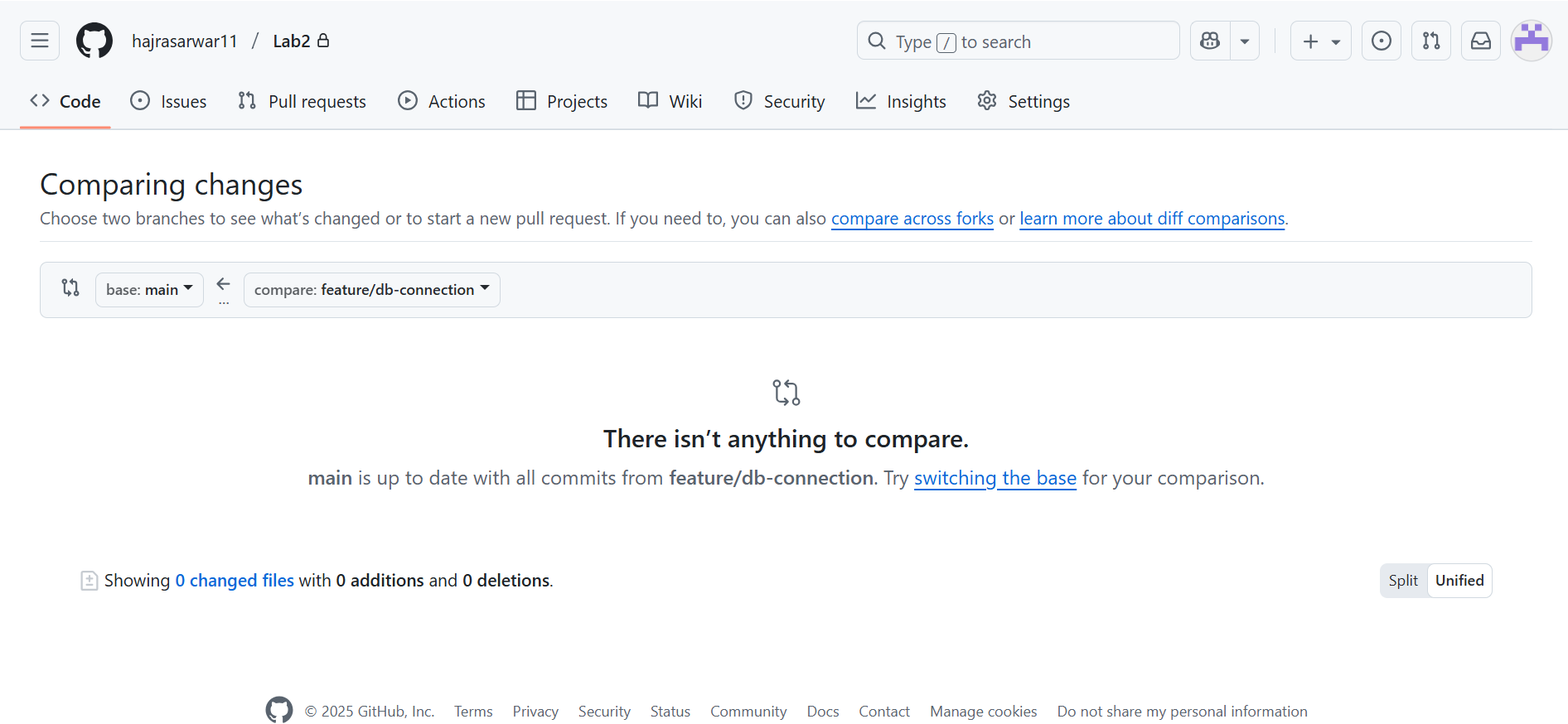
**Task 10: Pull Request and Branch Review (GitHub GUI)**

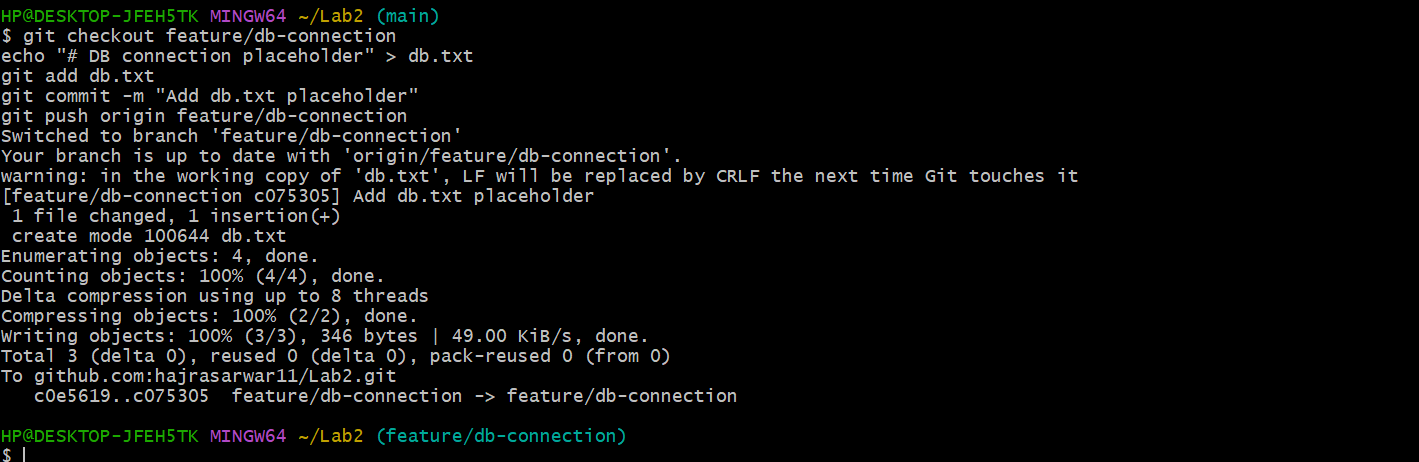
1. **On GitHub, create a Pull Request from the branch feature/db-connection to main.**



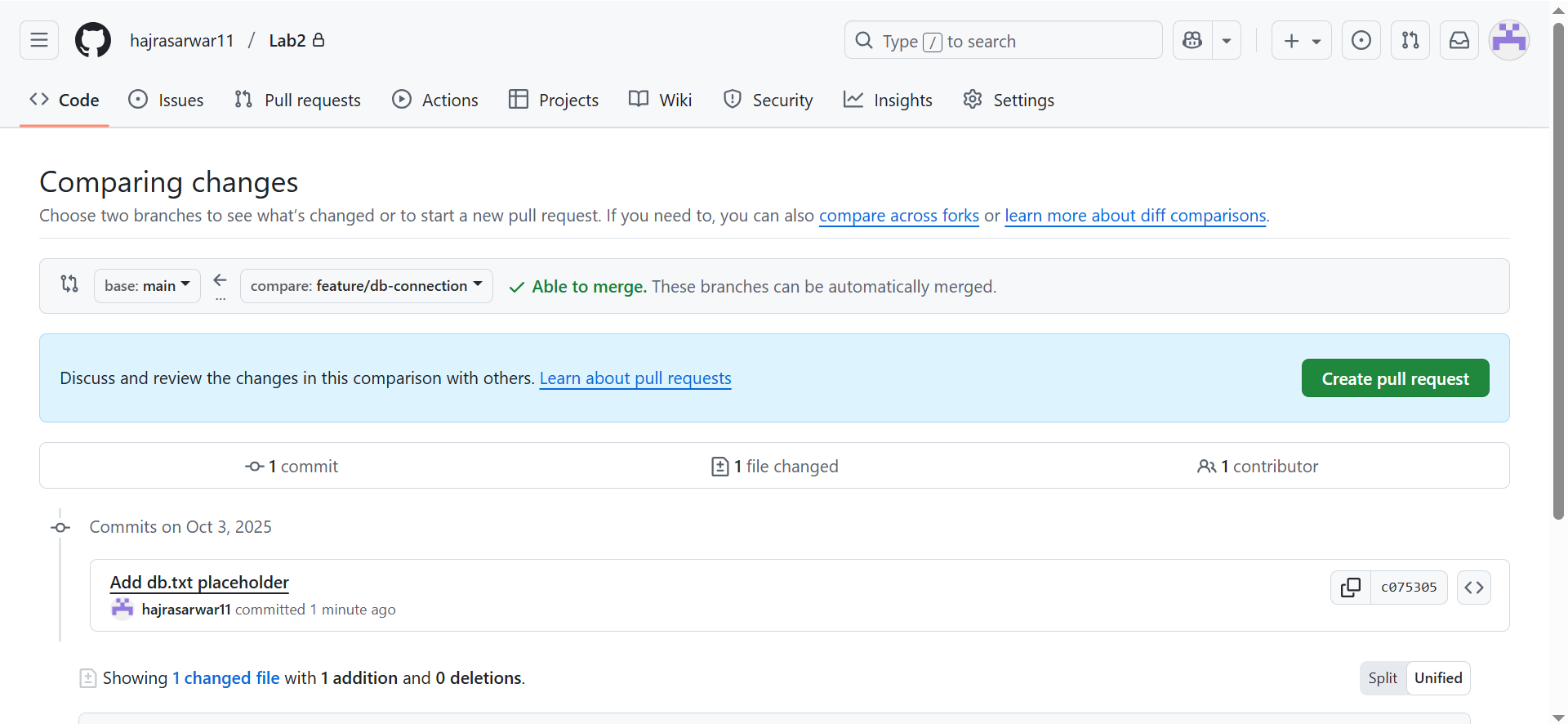


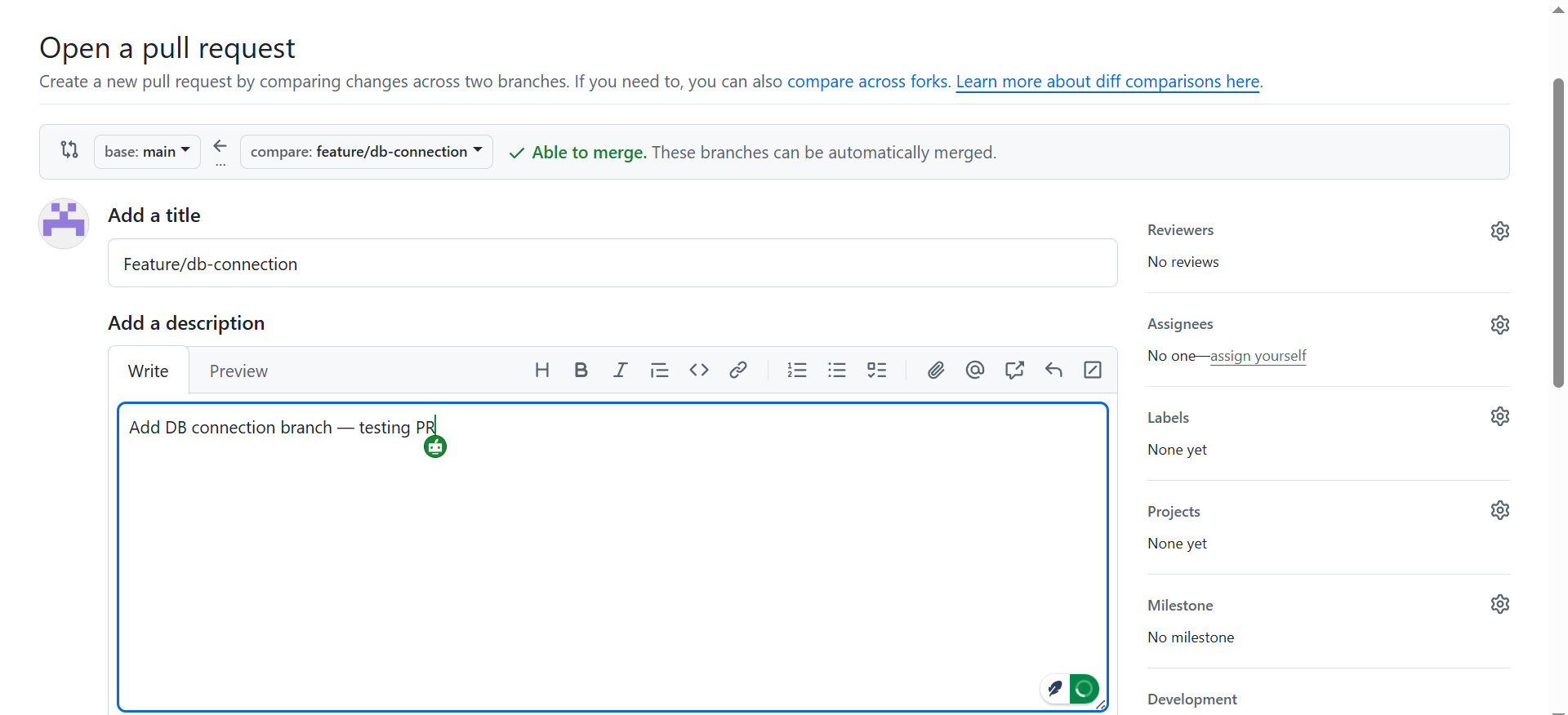


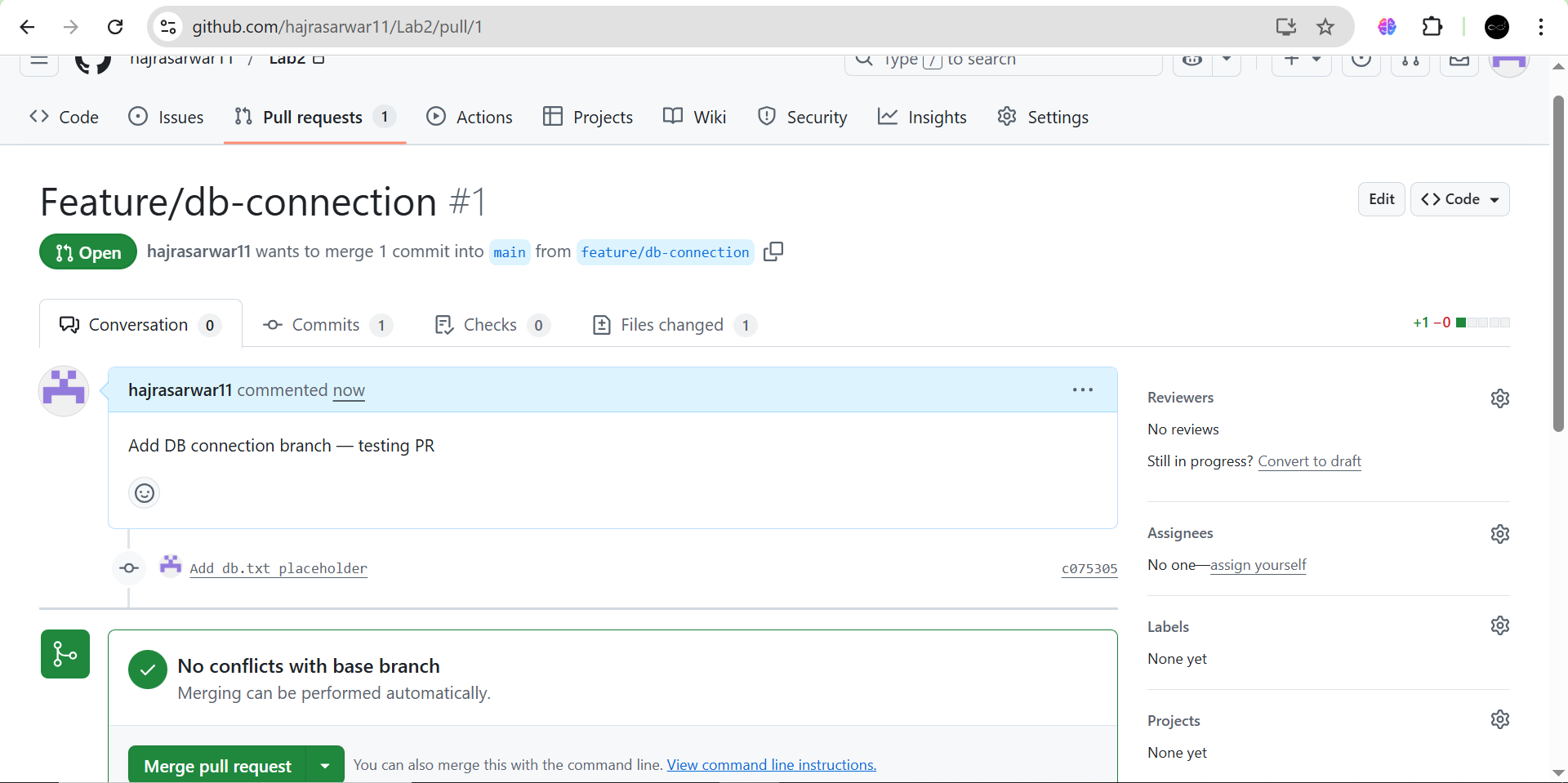


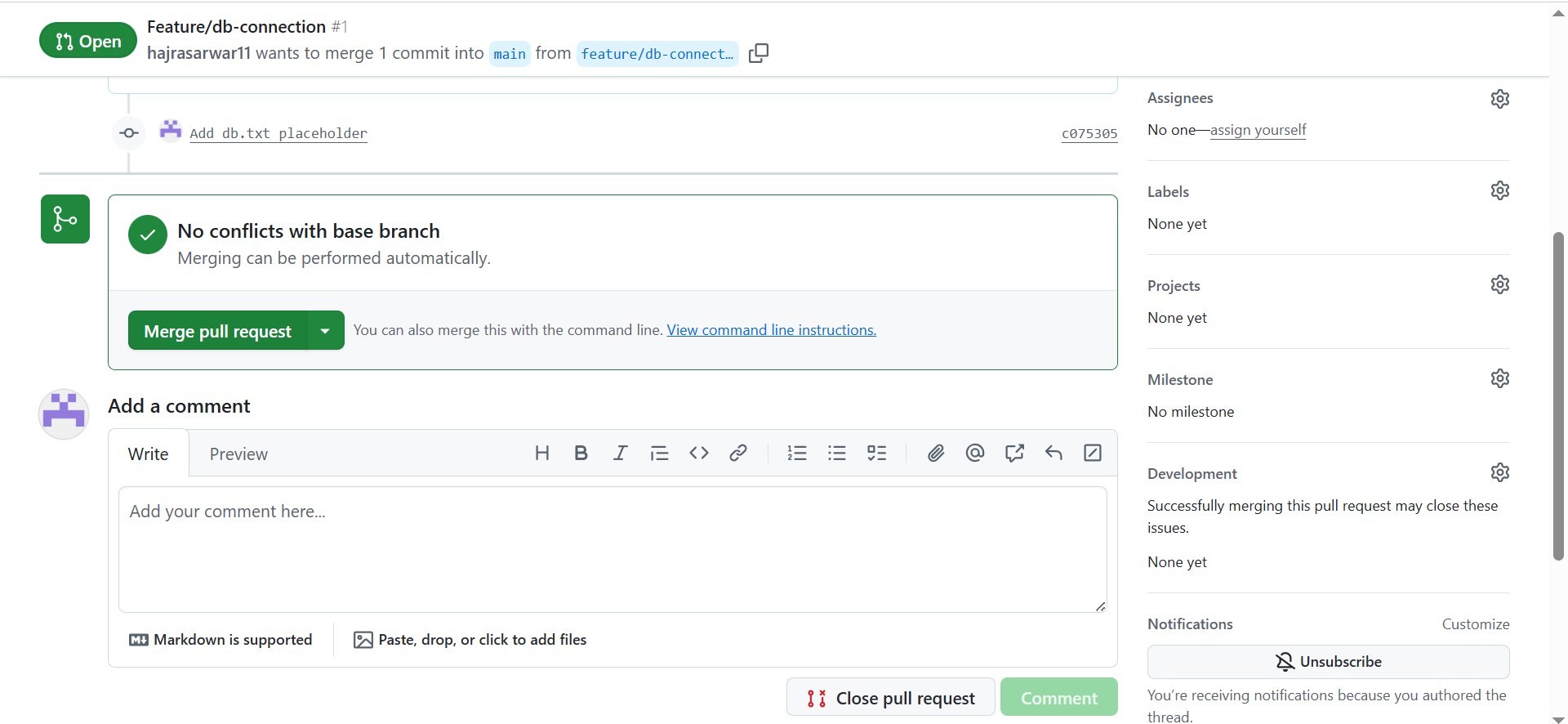


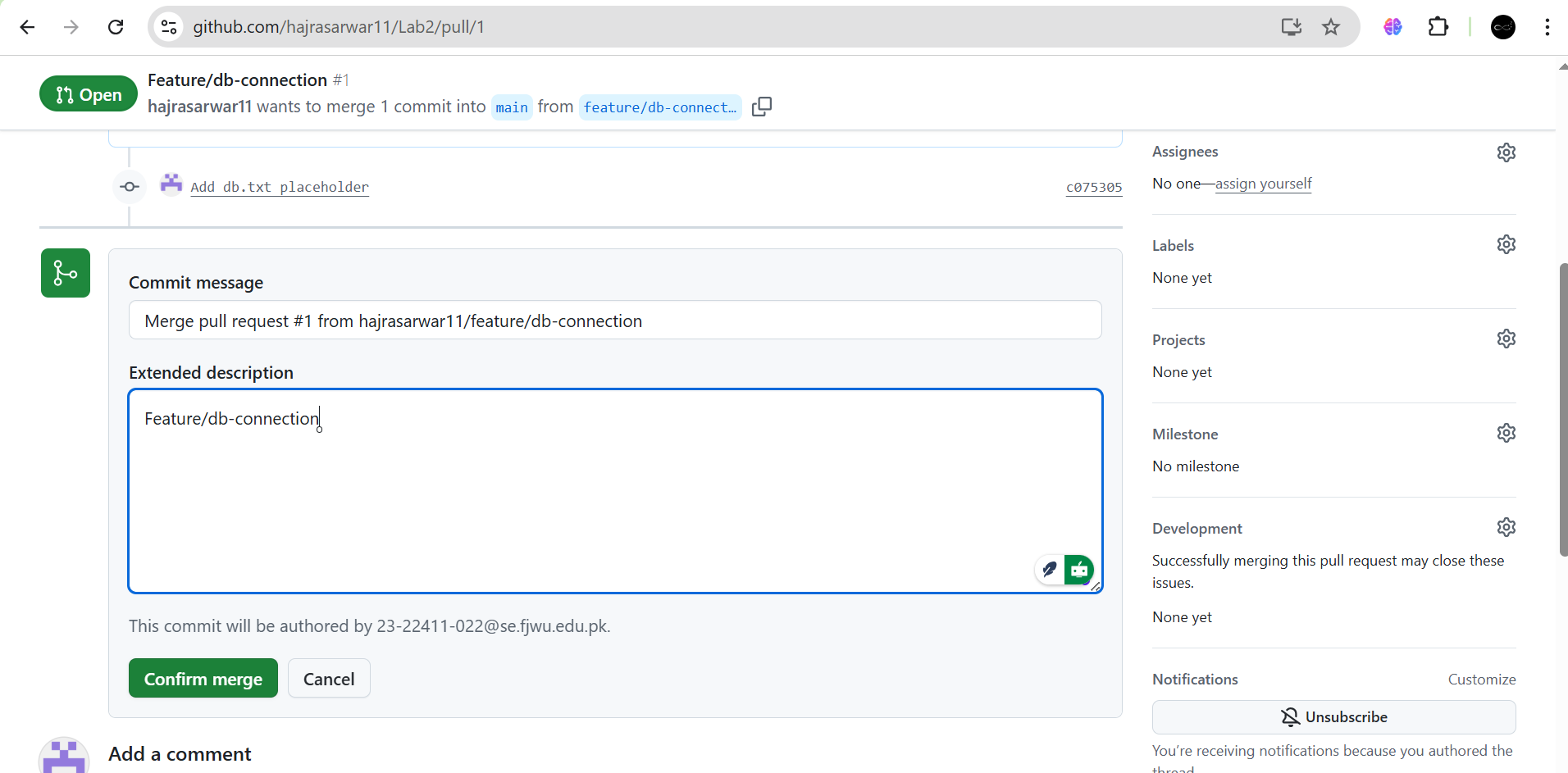
1. **Review the Pull Request and merge it using the GitHub GUI.**

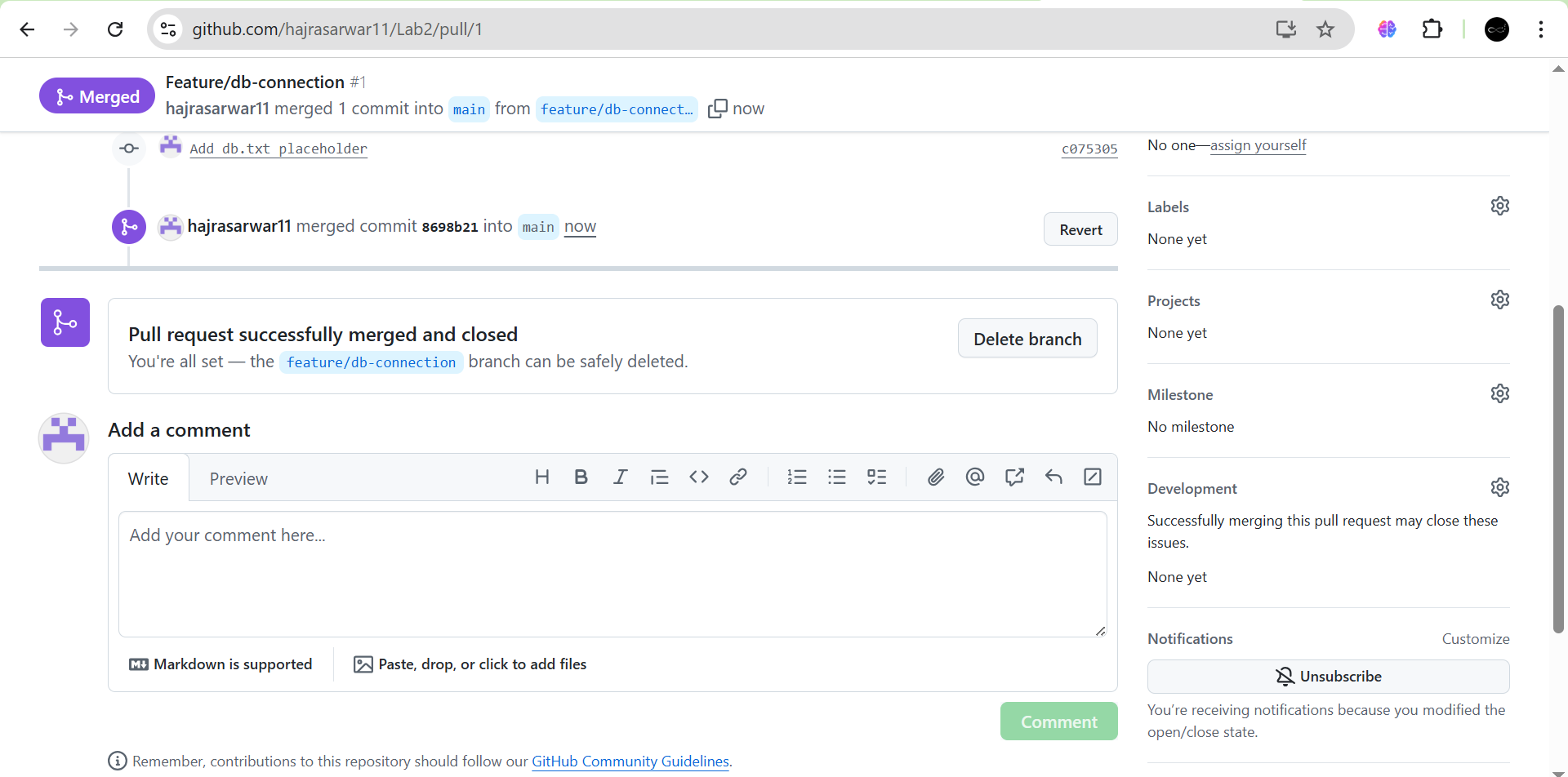




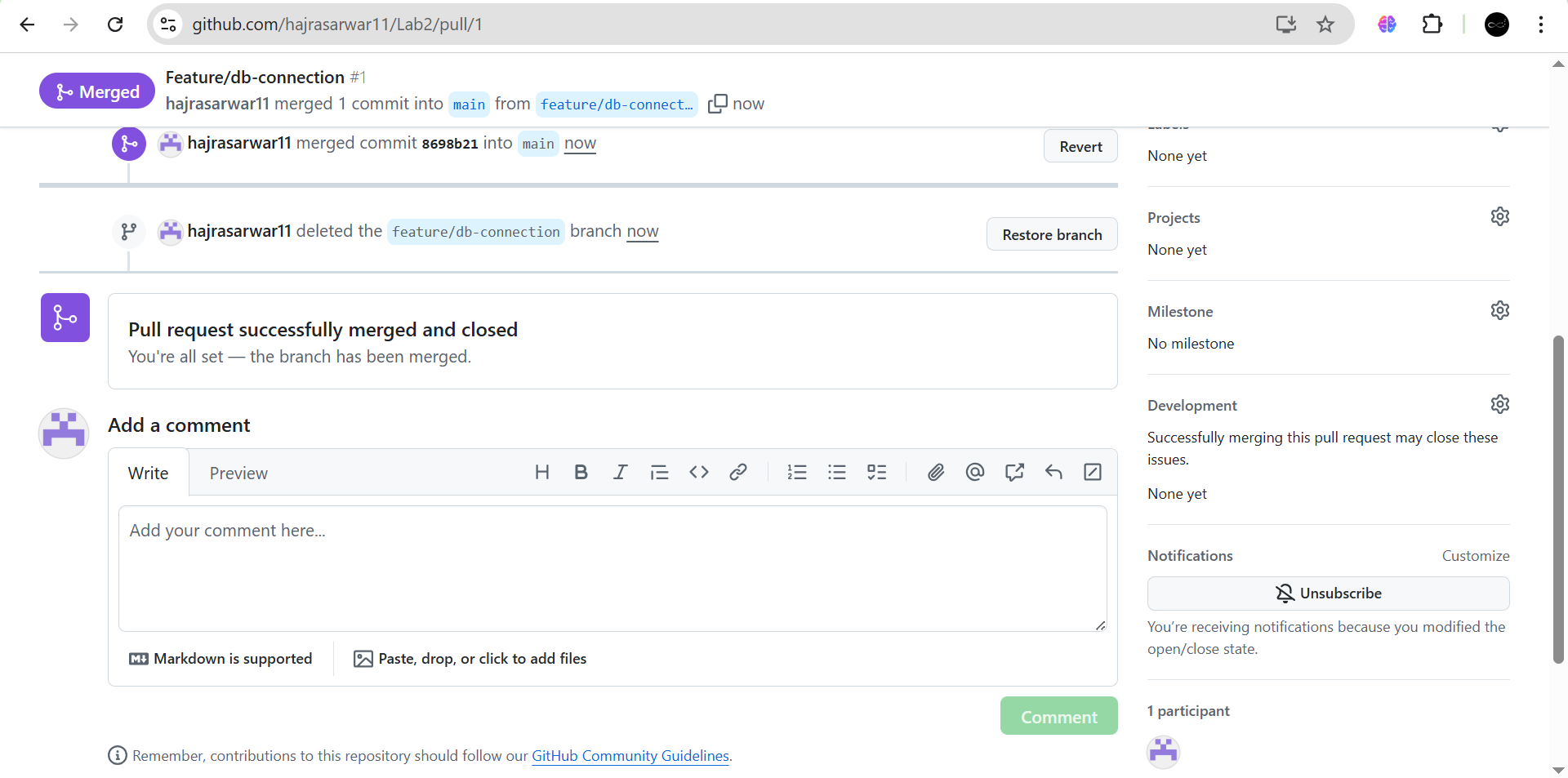


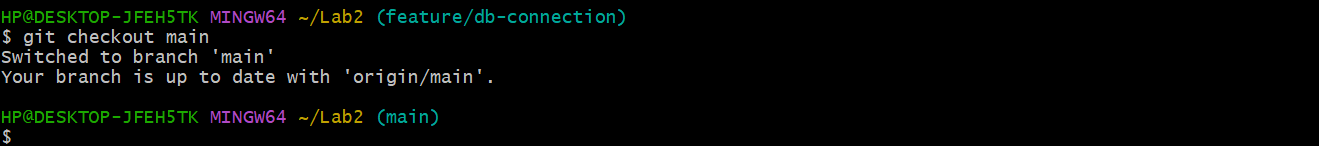


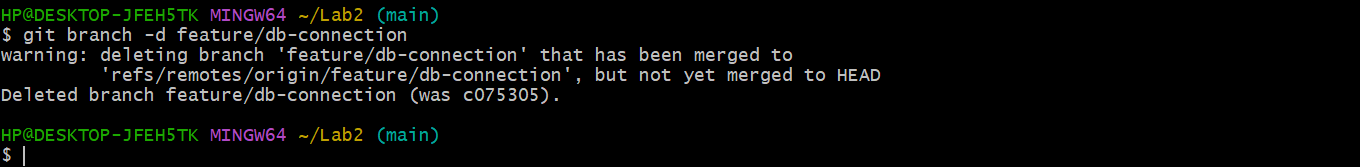


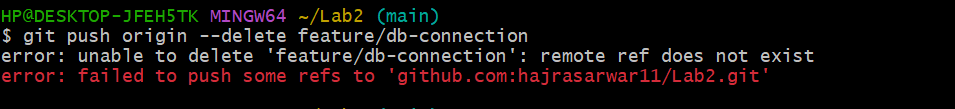


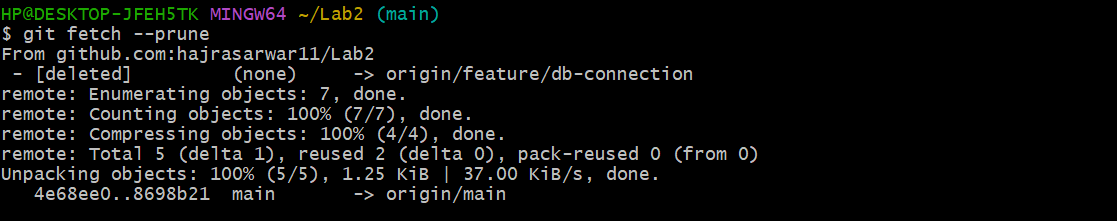
1. **After merging, delete the feature/db-connection branch using the GitHub GUI.**

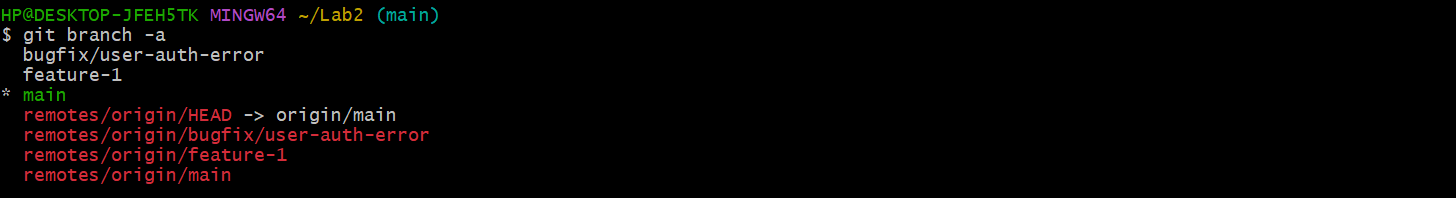








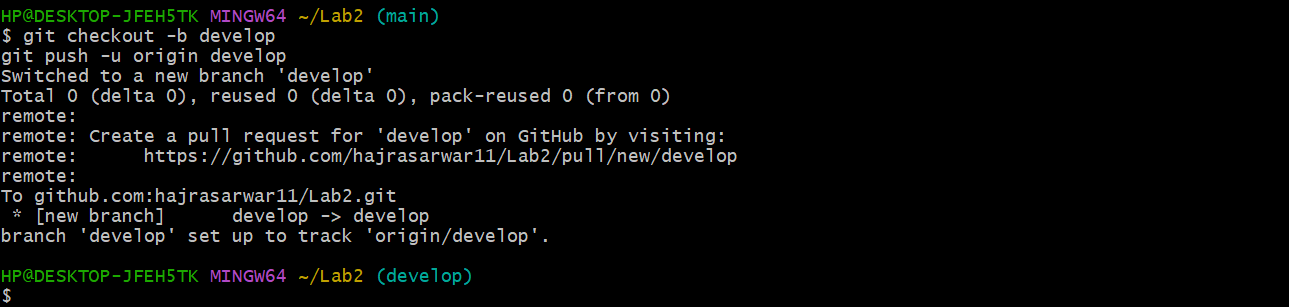


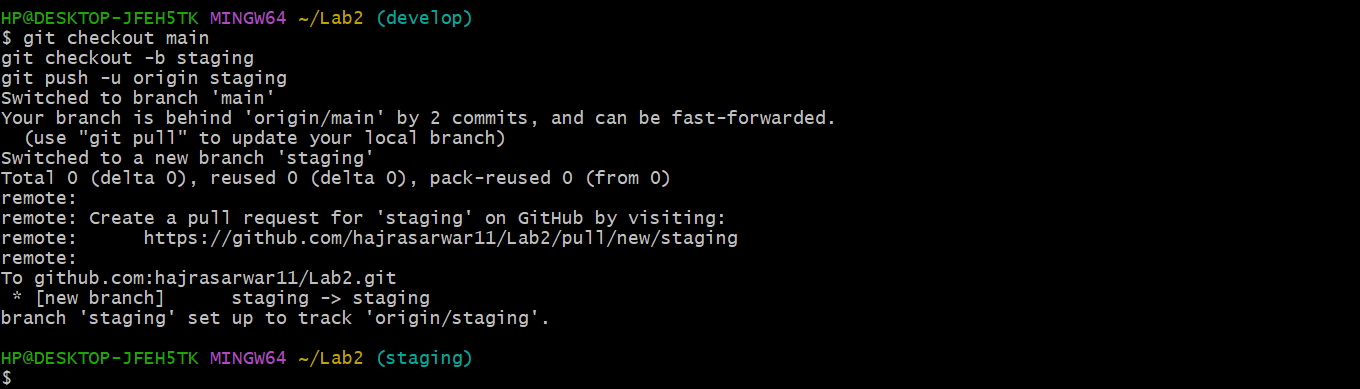


**Task 11: Detailed Branch Strategy (Develop/Staging)**

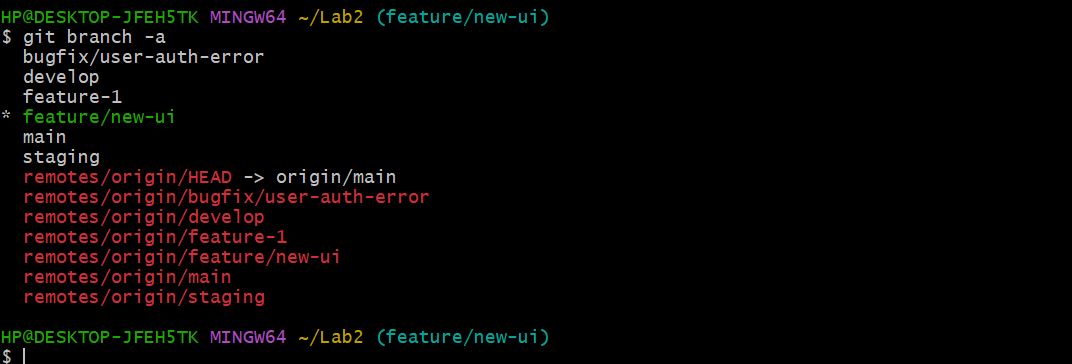
1. Create the following branches to simulate a professional branching strategy:

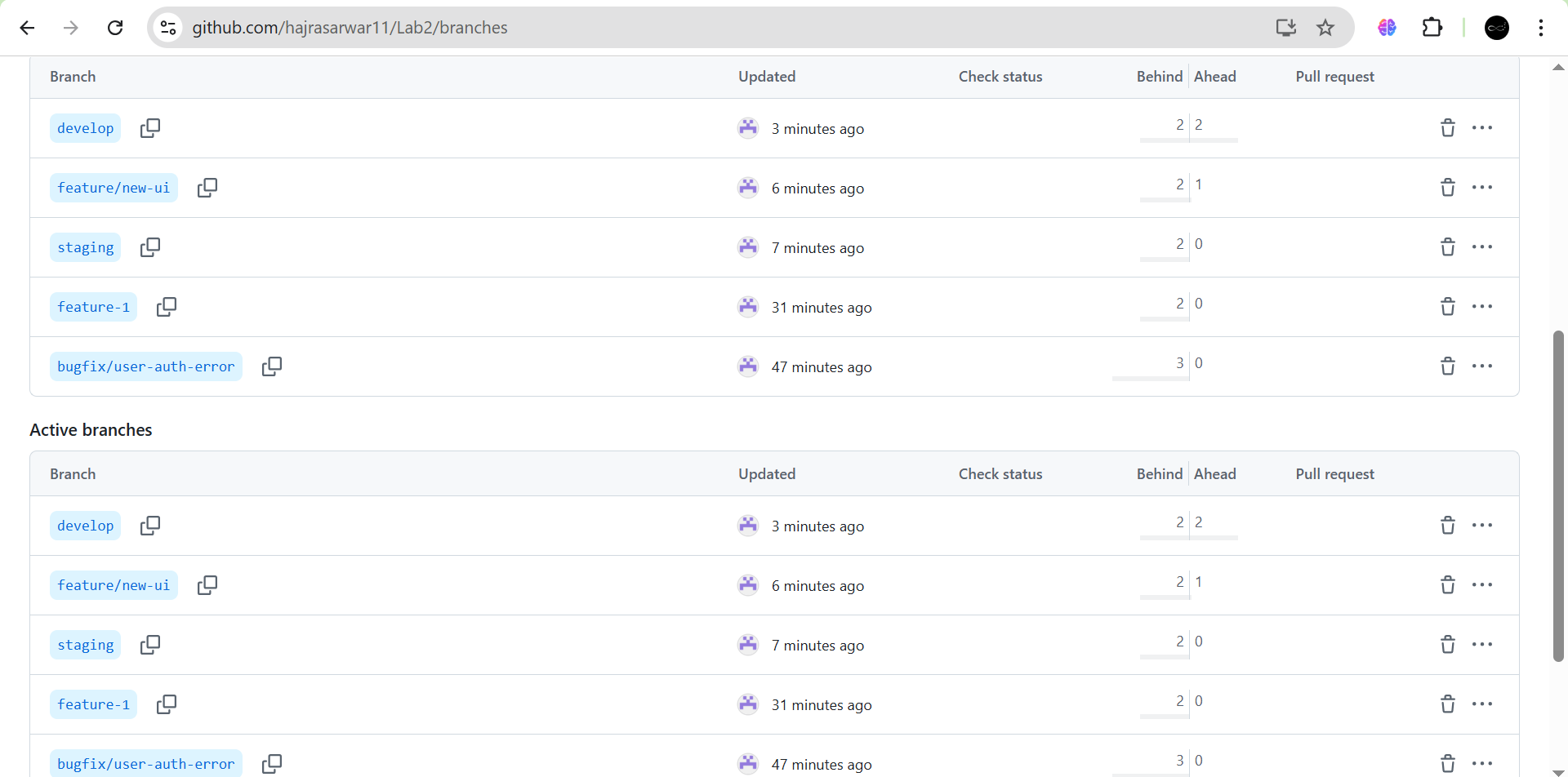
. Develop, Staging, feature/\*, bugfix/\*

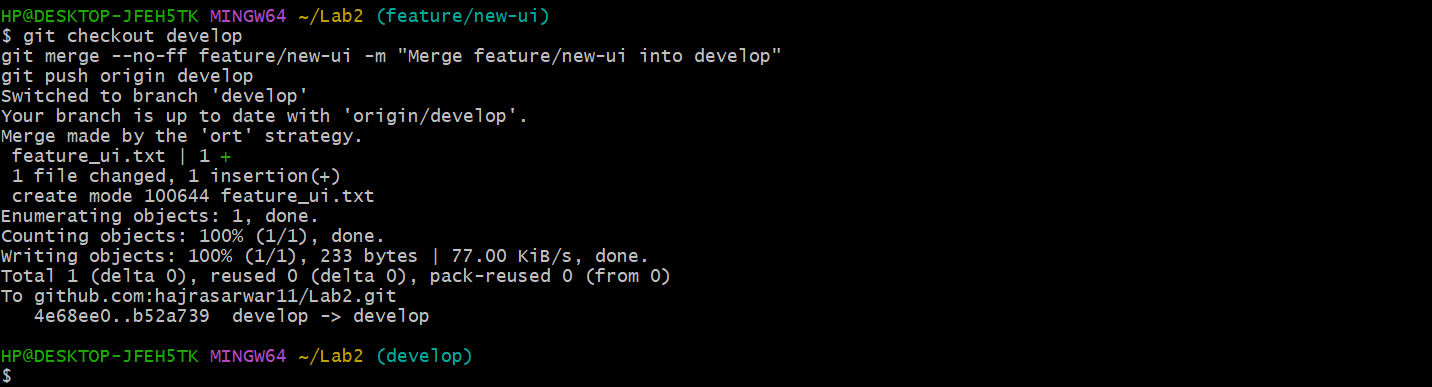


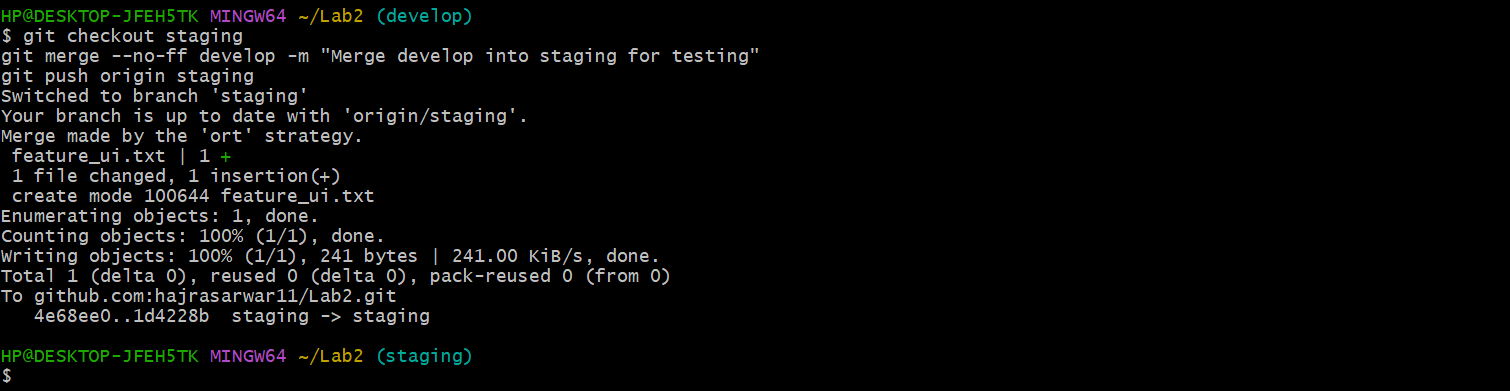


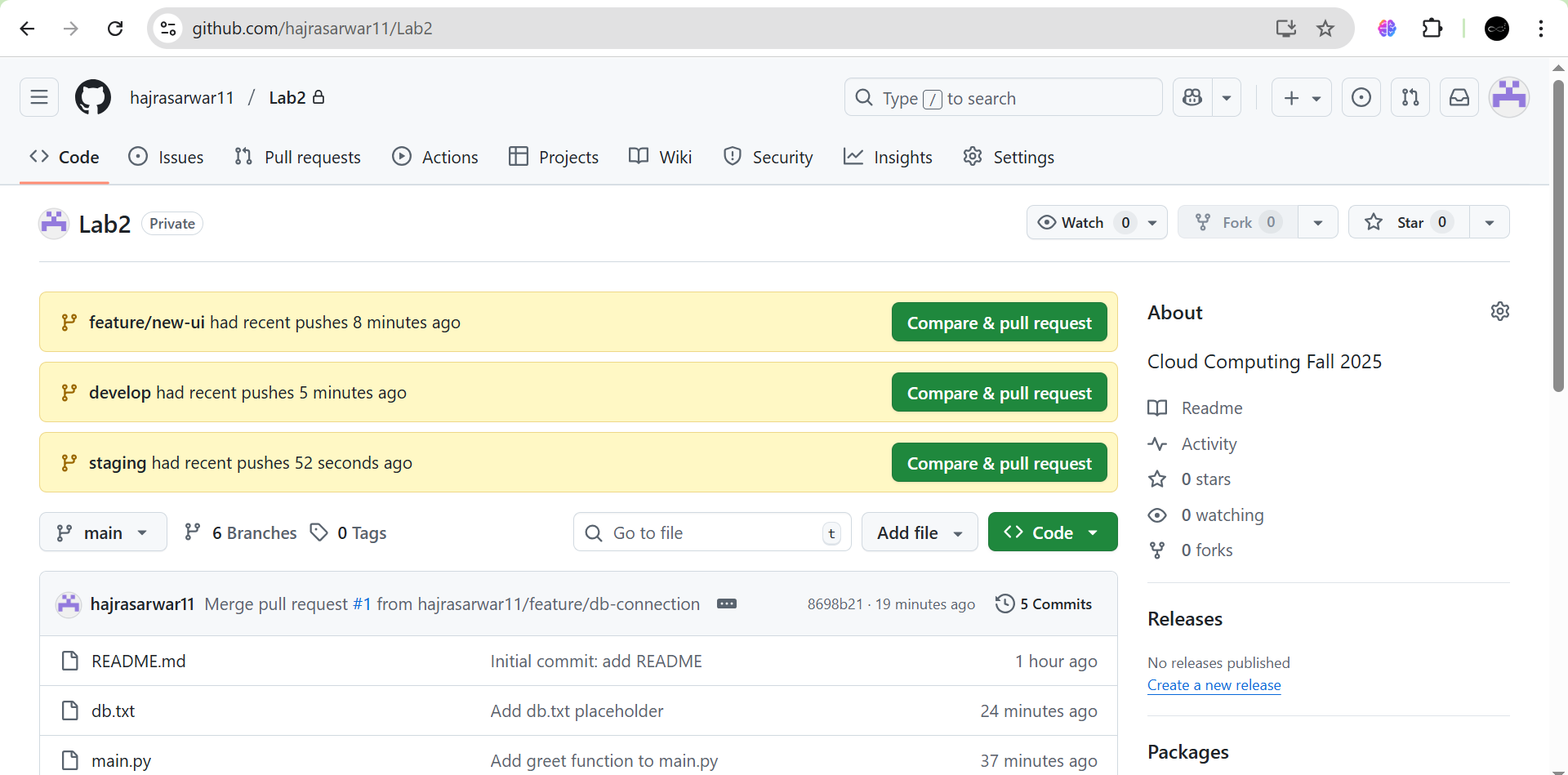


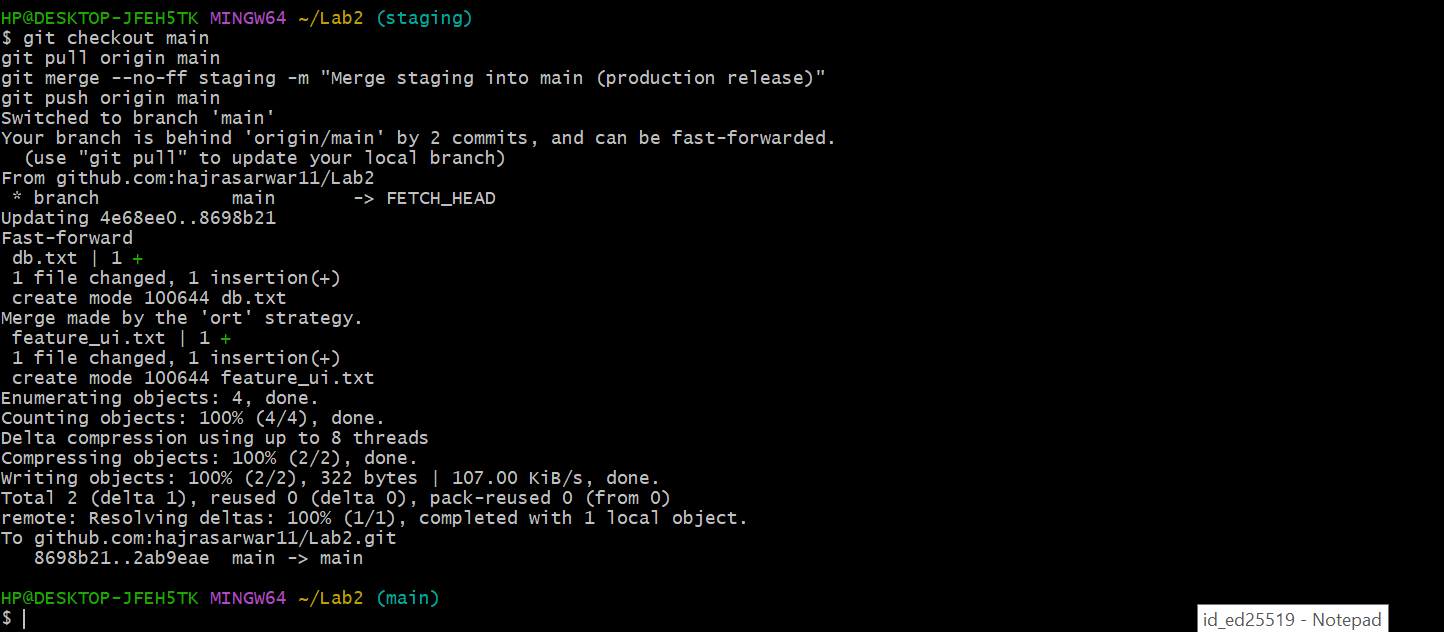


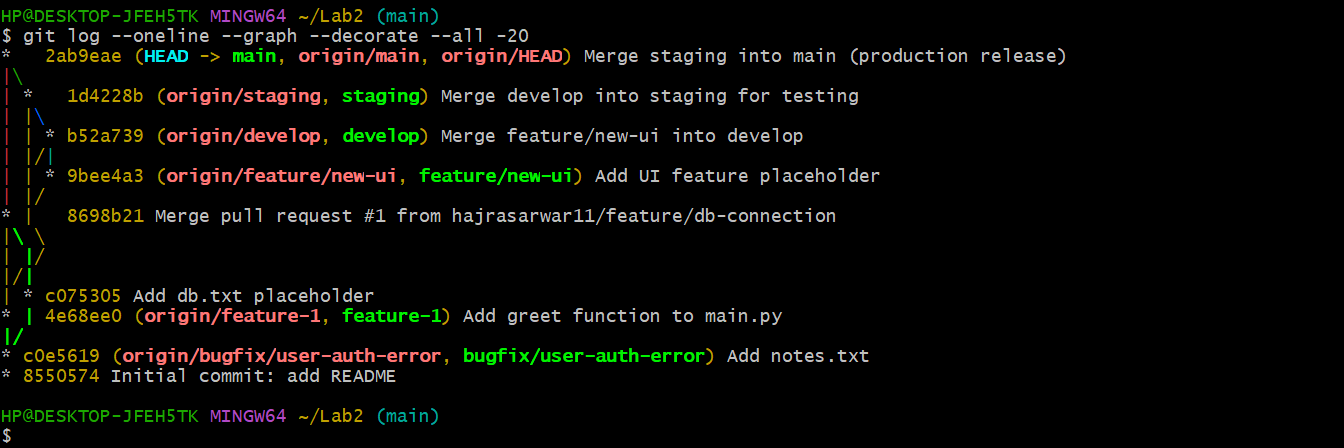


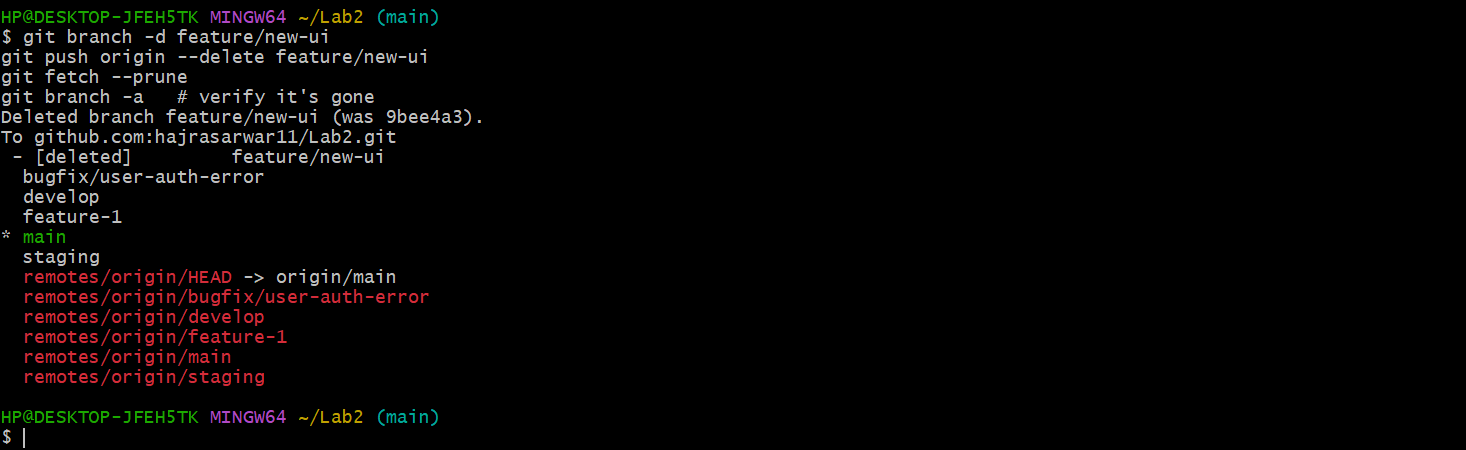










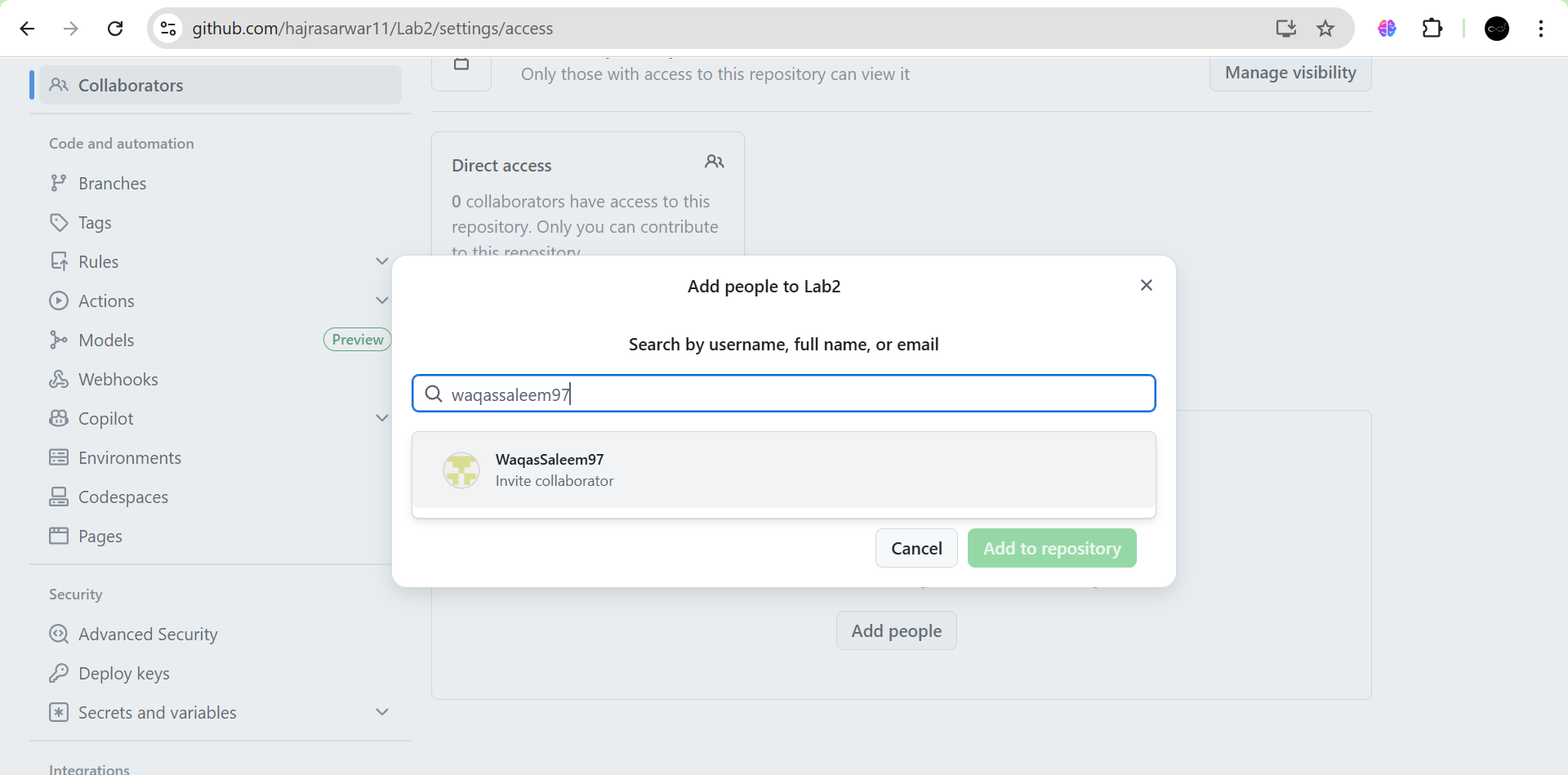


**Bonus Task: Simulated Team Collaboration**

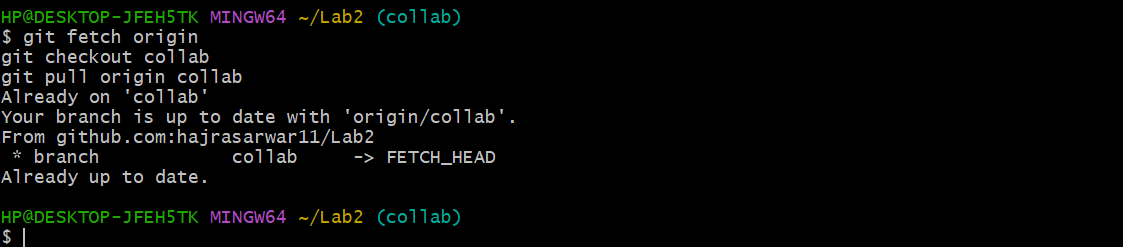
1. Create a branch collab.

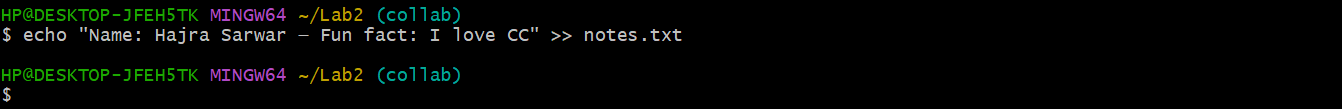


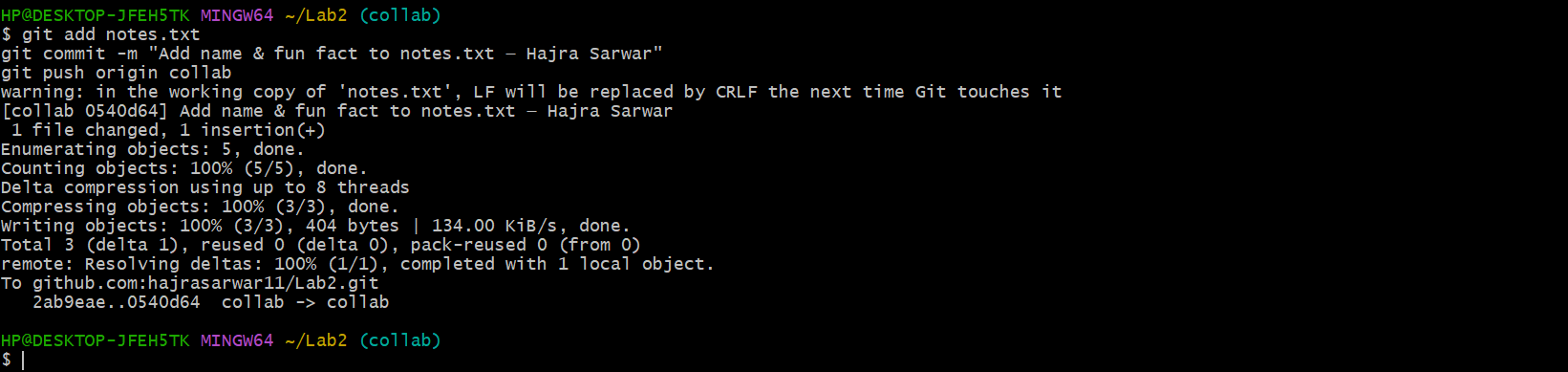
1. Add a collaborator on your GitHub repo.



1. Both users:
   * Pull latest changes.
   * Add their name and fun fact to notes.txt.
   * Commit and push to the collab branch.







1. Merge collab into main and push.

git checkout main

git merge collab

git push origin main

