**Cloud Computing Lab**

**Lab 02 tasks**

# **Hands-On Git & Version Control Lab**

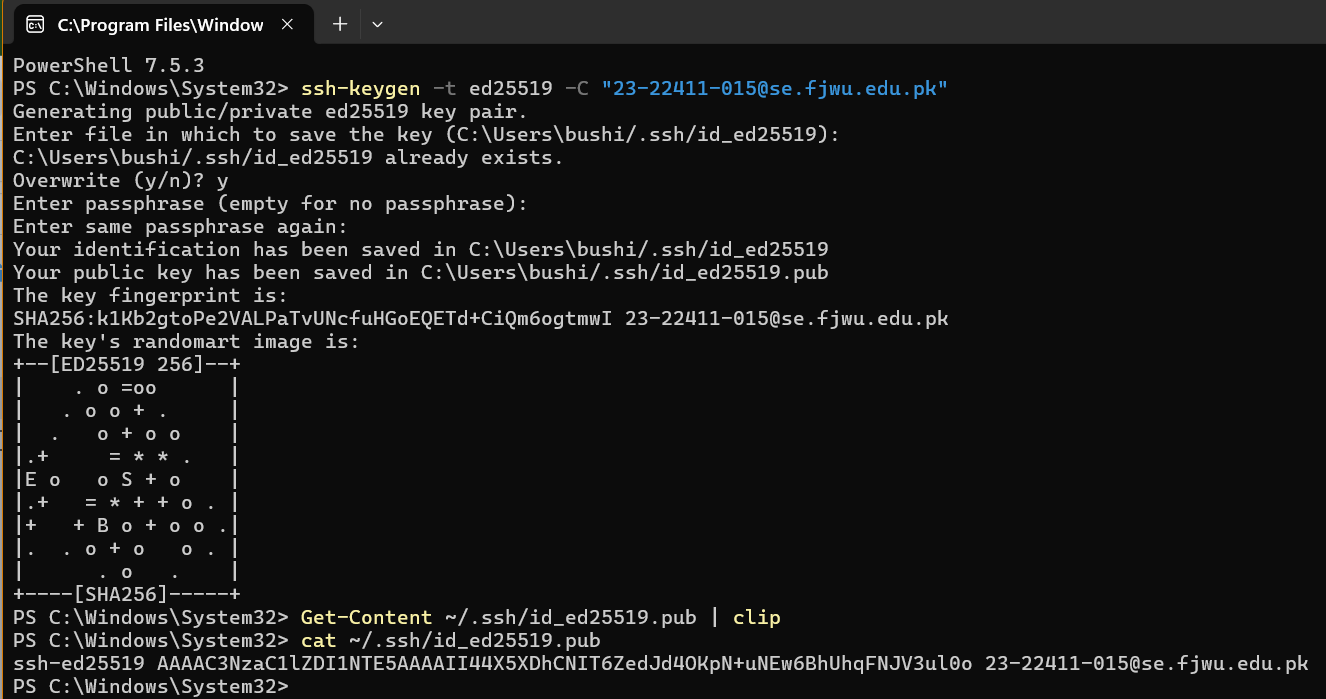
**Submitted To:** Muhammad Shoaib

**Submitted From:** Bushra Ashraf Bhatti

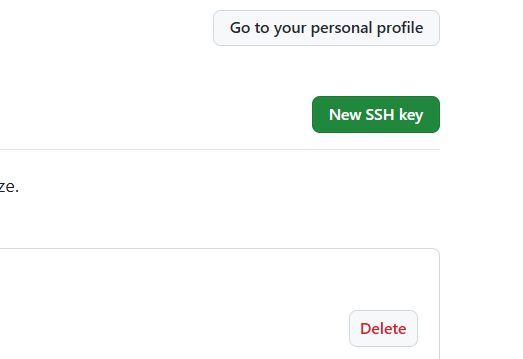
**Registration Number:** 2023-BSE-015

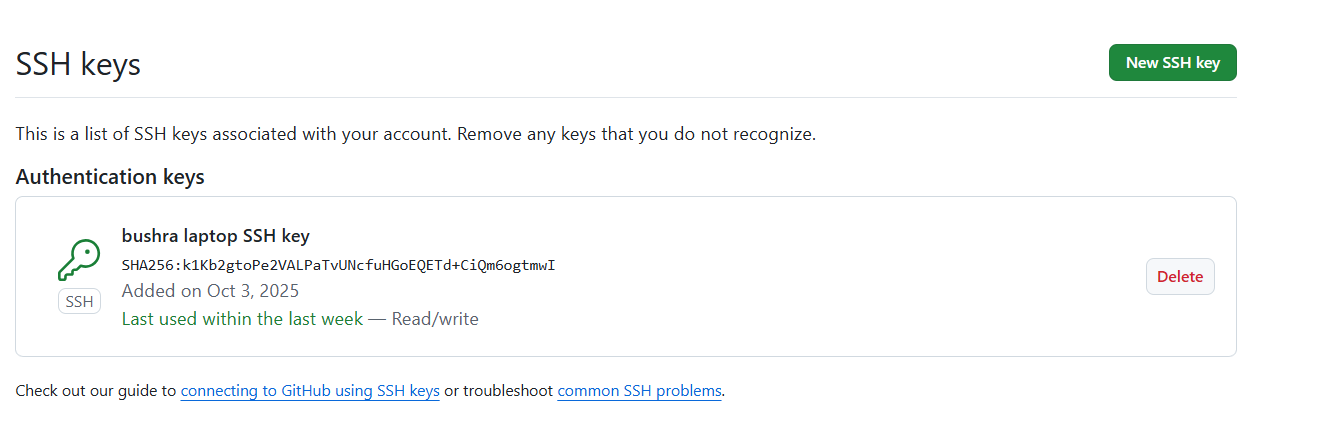
**Task 2: Connect Repository via SSH**

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

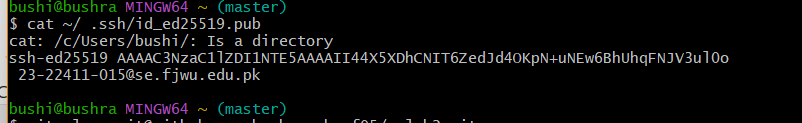


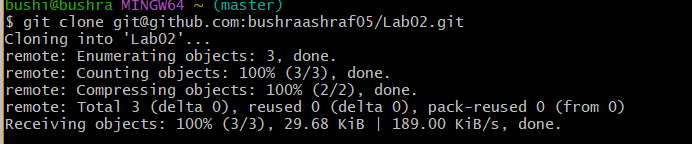
**Add your SSH public key to GitHub (Settings > SSH and GPG keys).**





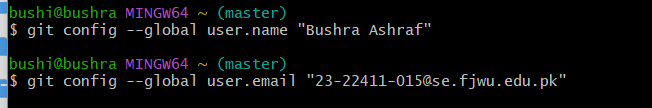
**Clone your Lab2 repo using SSH.**

****

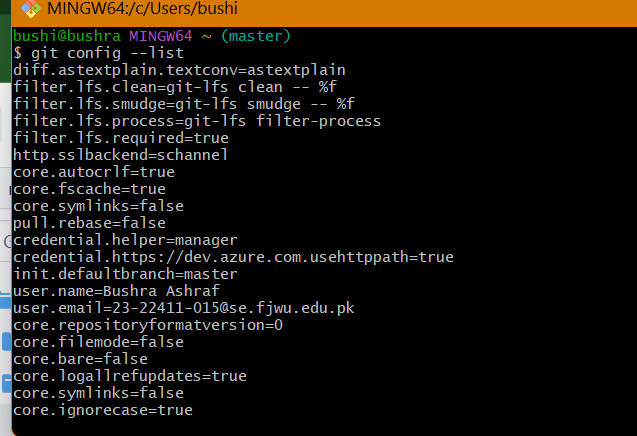
****

## Task 3: Configure Git Username and Email

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

****

1. **Verify your configuration:**

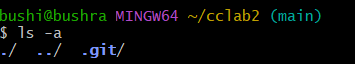
****

## Task 4: Explore the .git Folder

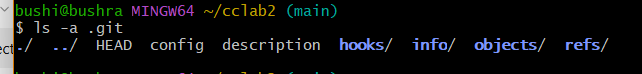
1. Navigate into your cloned repository folder.

****

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

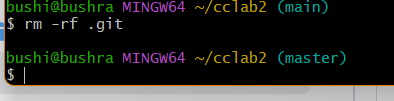


1. Explore what’s inside using:

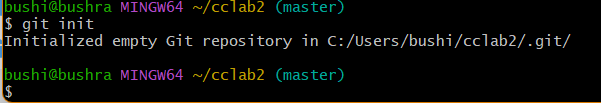


## Task 5: Local Repository Management

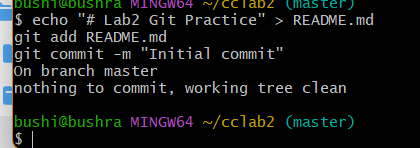
1. **Delete the existing .git folder from your cloned repo using Git Bash**



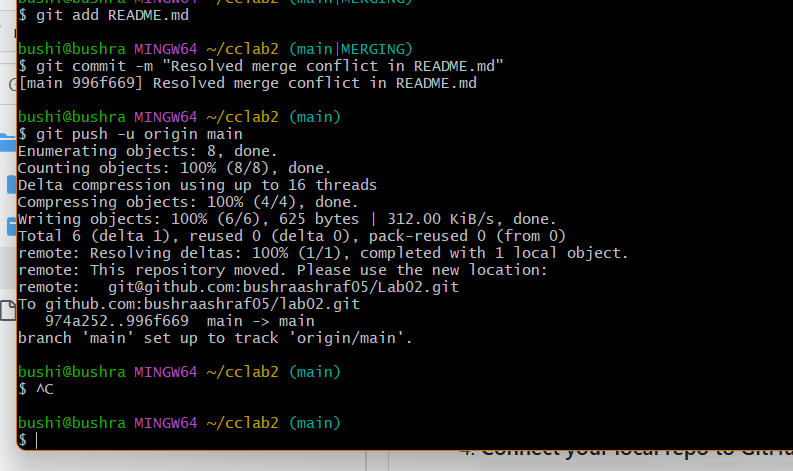
1. **Re-initialize the local git repository**



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

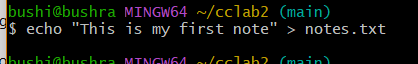


1. **Connect your local repo to GitHub and push:**

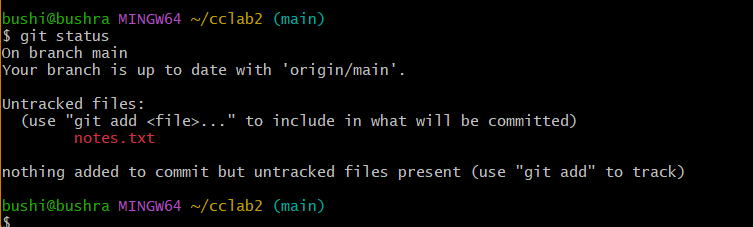


## Task 6: File Status & Staging

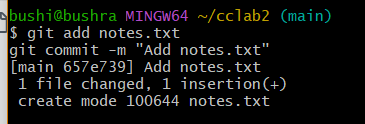
1. **Create a new file notes.txt and write a note.**

****

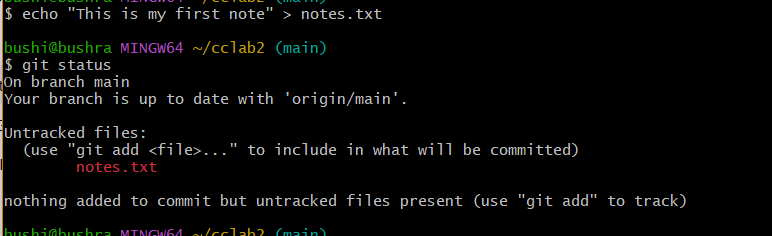
1. **Check status:**

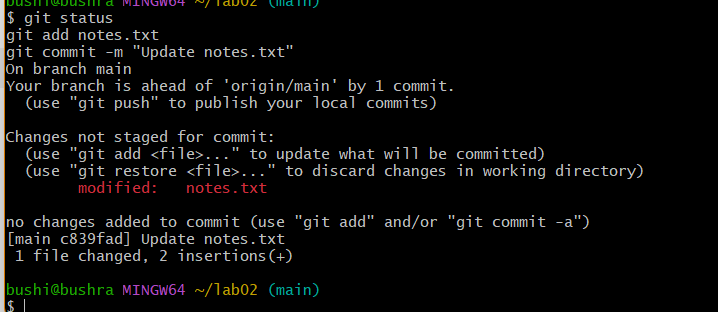
****

1. **Stage and commit:**

****

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.

## 

## Branch created:

## 

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

## 

## (This will bring the branch from GitHub to your local repo.)

## Verify branch list locally

## 

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

## ****Create a branch named****feature/db-connection****using Git Bash****

## 

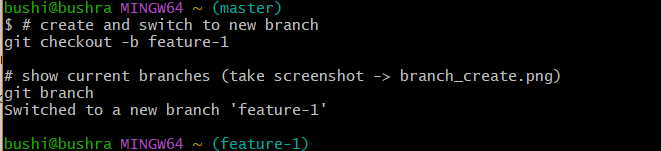
## ****Push the branch to the remote repository:****

## 

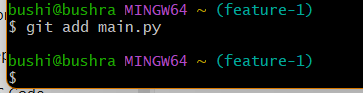
## 

## Task 9: Branching & Merging

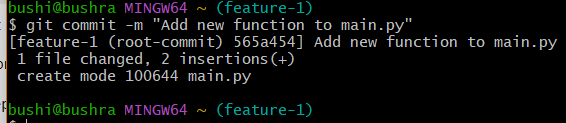
1. **Create & switch to feature-1**

****

1. **Stage your changes for commit**

****

1. **Commit your changes**

****

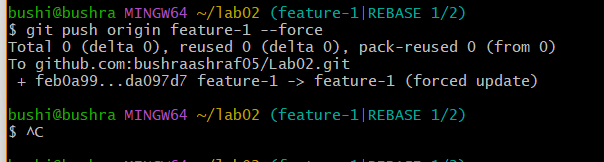
1. **Switch back to main branch**

****

1. **Merge feature-1 into main**

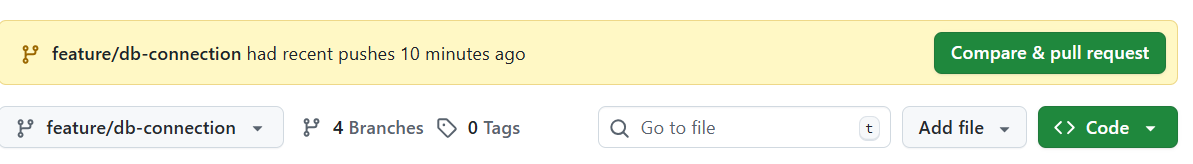
****

1. **Push branches to GitHub**

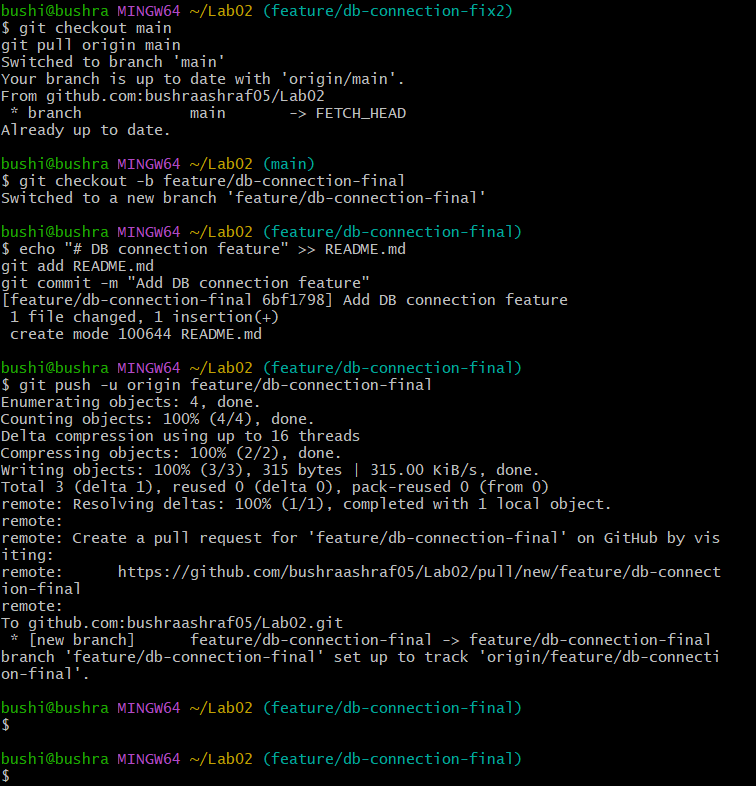
****

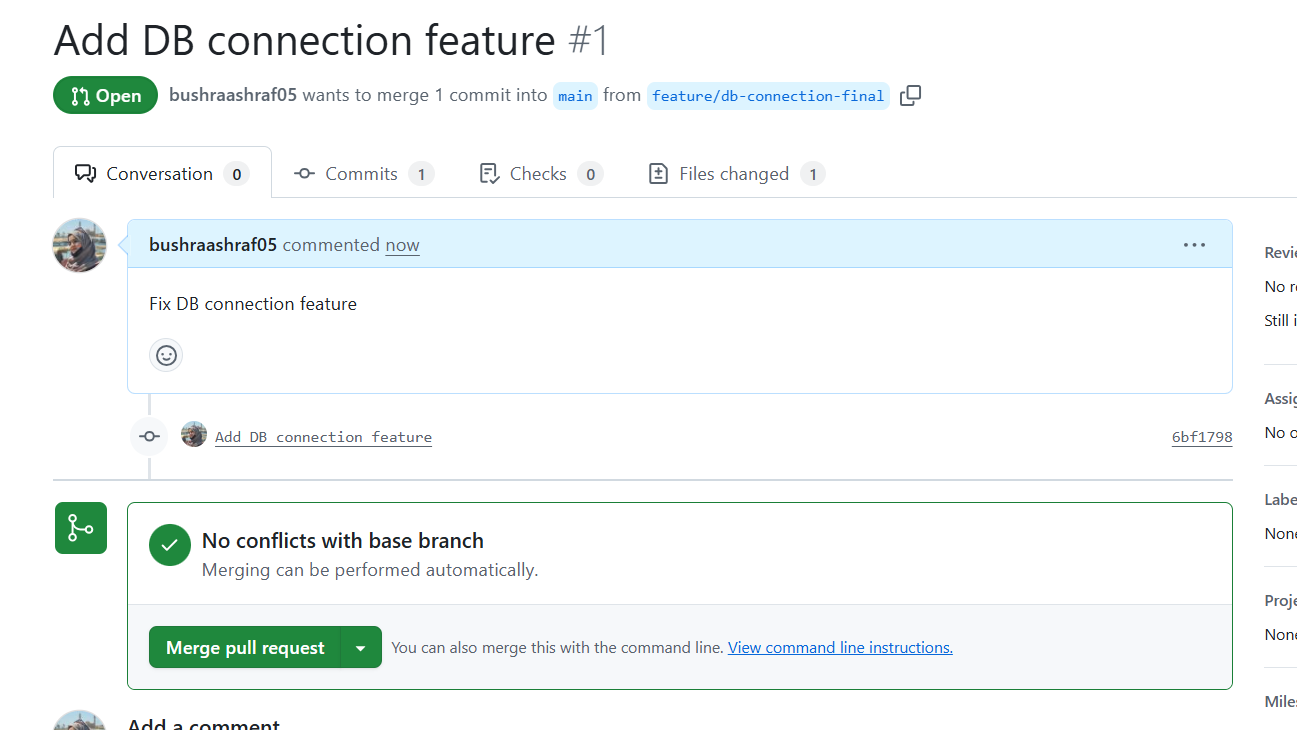
## Task 10: Pull Request and Branch Review (GitHub GUI)

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

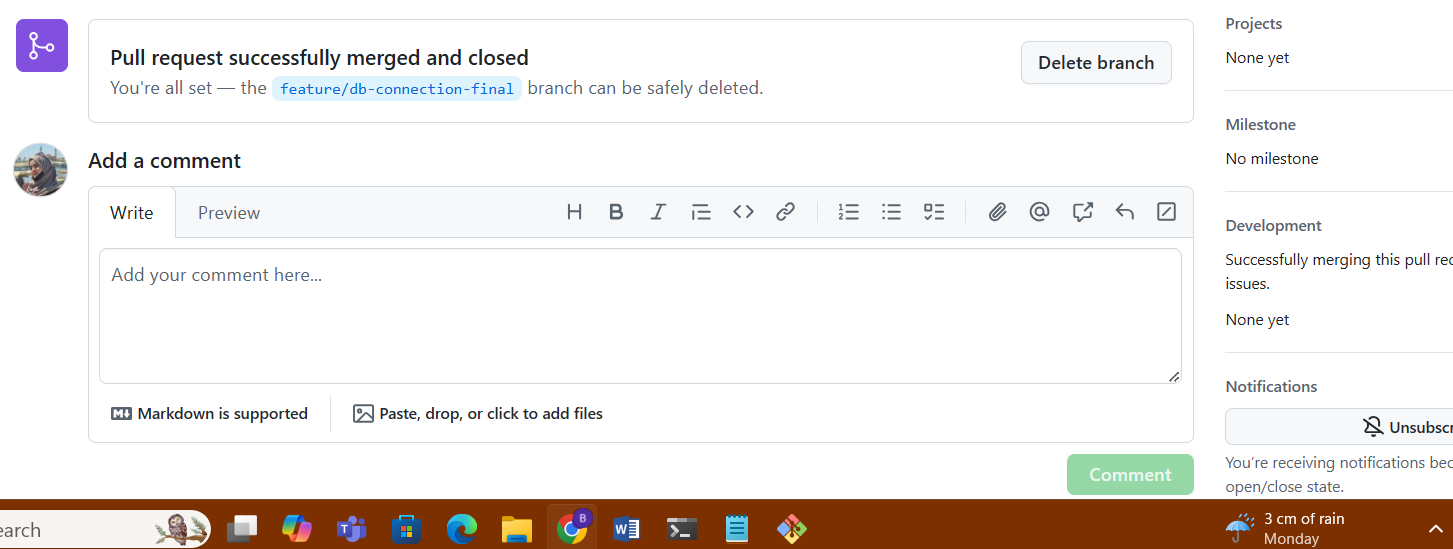


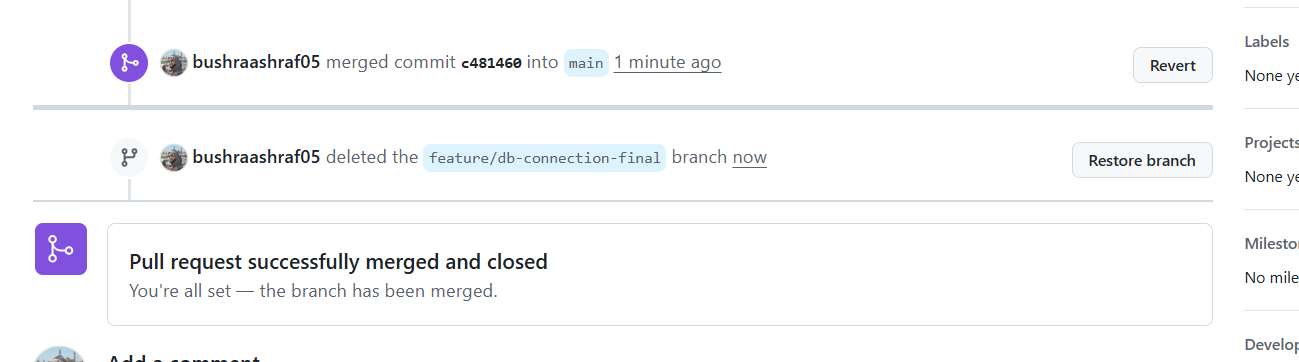
1. **Review the Pull Request and merge it using the GitHub GUI.**





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

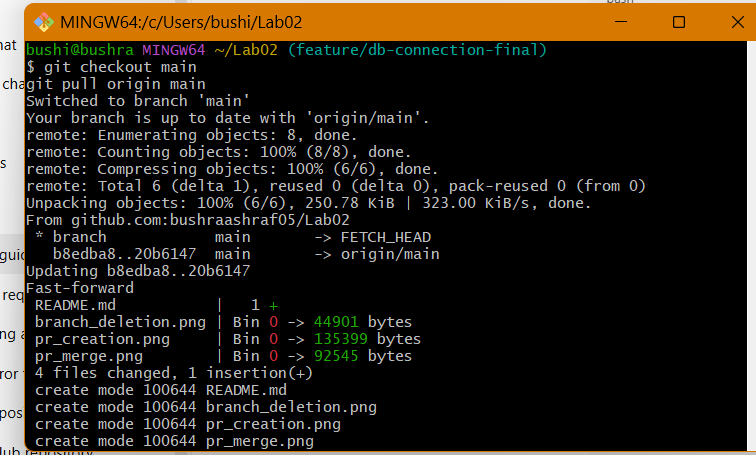




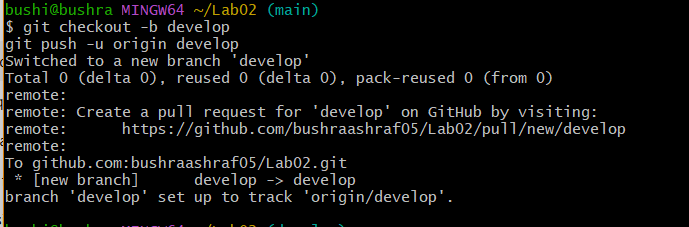
**Task 11: Detailed Branch Strategy (Develop/Staging)**

* 1. **Create the following branches to simulate a professional branching strategy:**
     + **develop**
     + **staging**
     + **feature/\***
     + **bugfix/\***

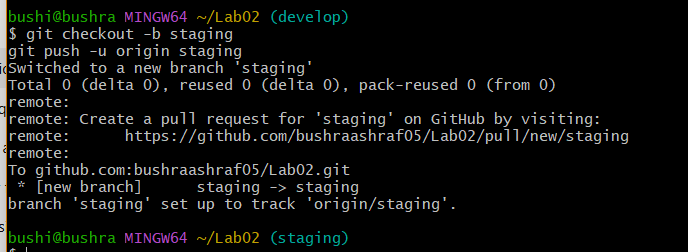
1. **Example workflow:**
   1. **Developers work on feature/\* and bugfix/\* branches.**
   2. **Merge into develop after completion.**
   3. **develop is merged into staging for testing.**
   4. **Finally, staging is merged into main (production).**
2. **Document the workflow with screenshots:**
   1. **Branch list (branch\_strategy.png)**
   2. **Merges into develop and staging (branch\_merges.png)**
   3. **Final merge into main (final\_merge.png)**



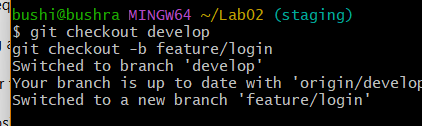
**Create develop:**

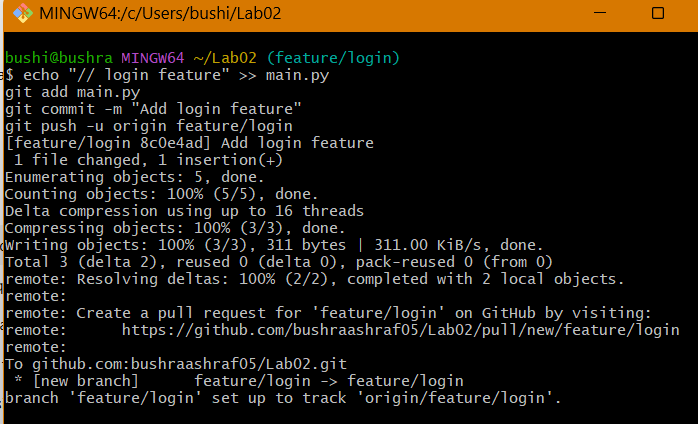
****

**Create staging :**

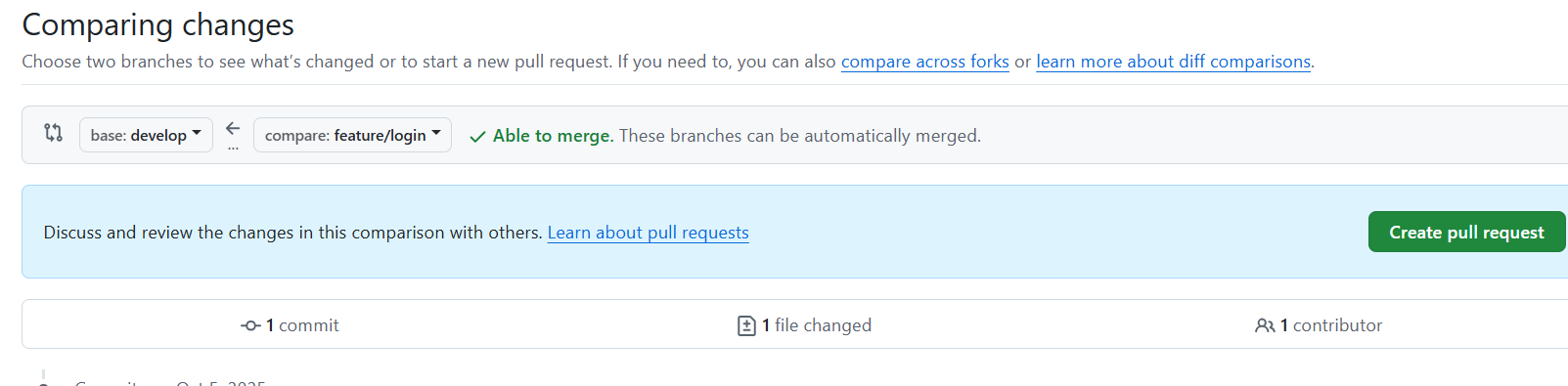
****

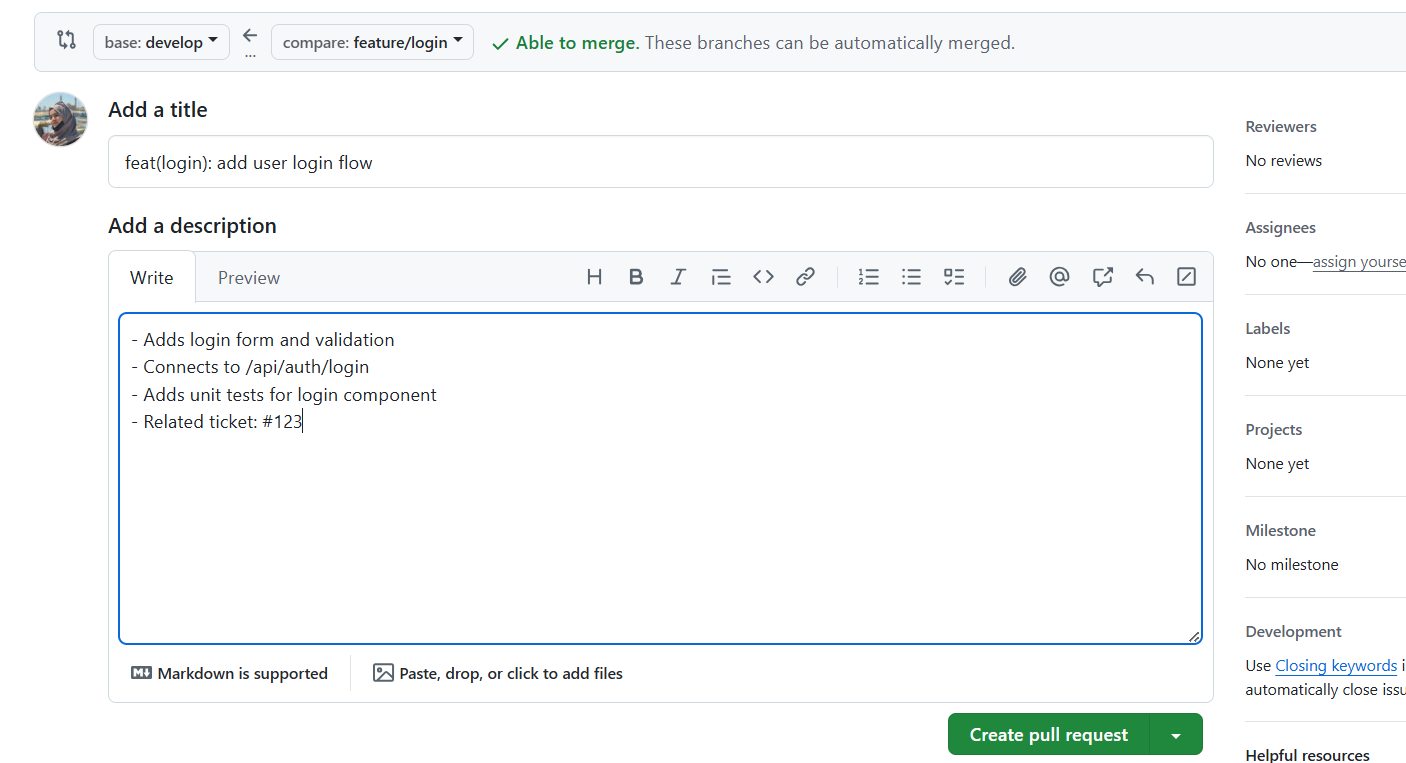
**Create a feature branch:**

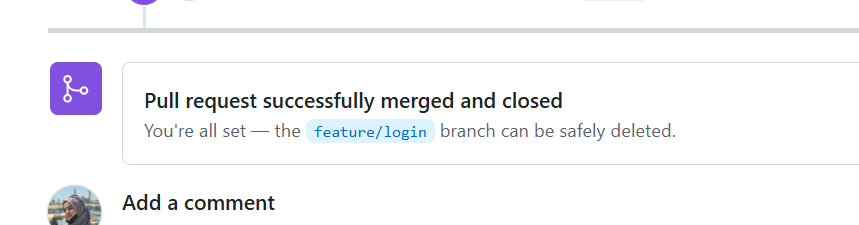
****

****

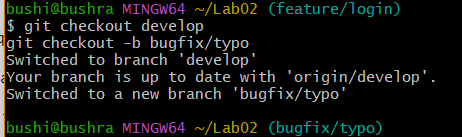
**Merge feature → develop**

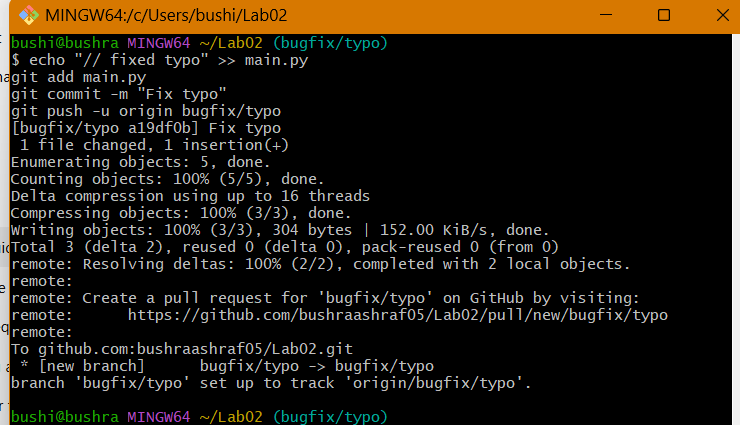
****

****

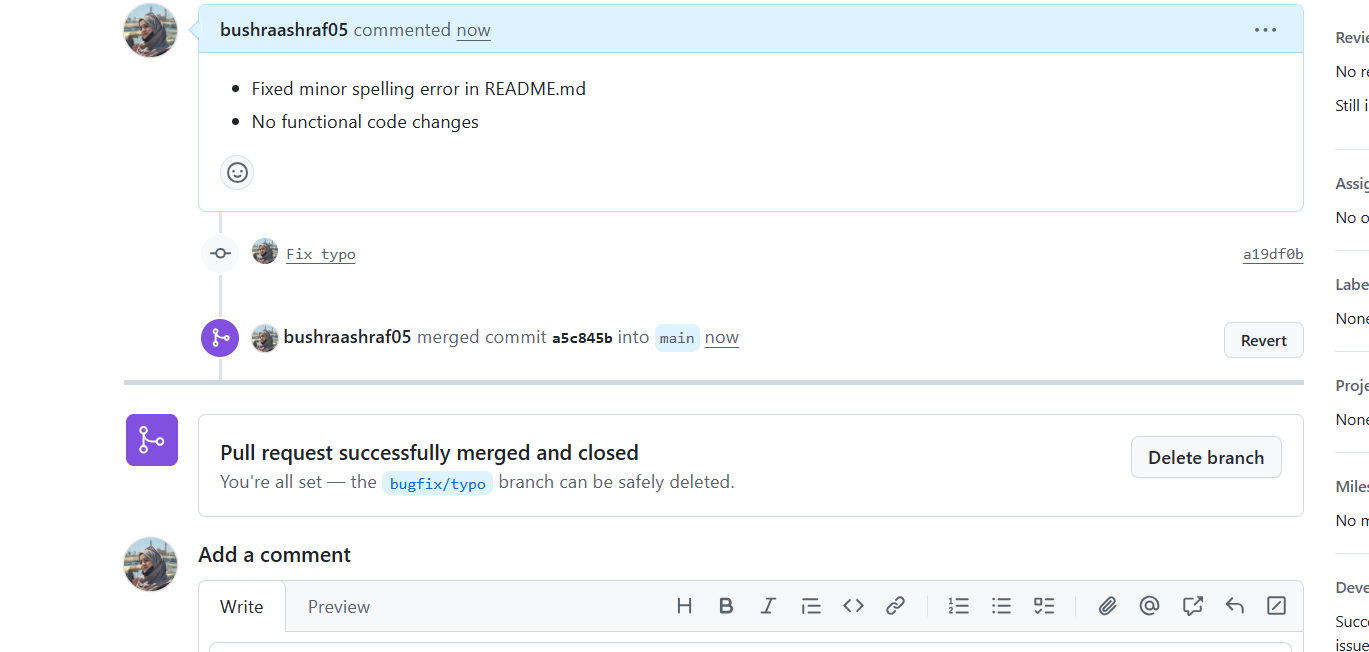
****

**Create a bugfix branch**

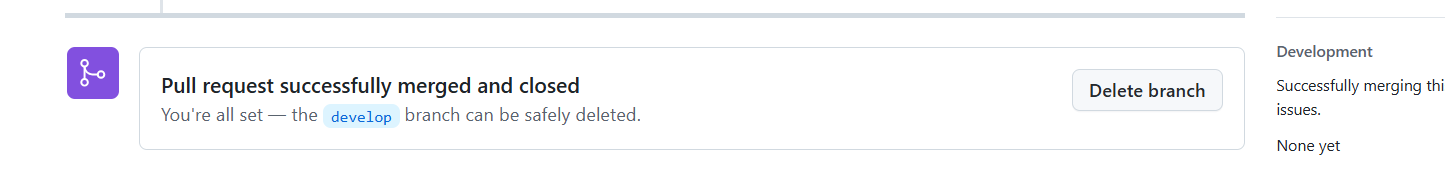
****

****

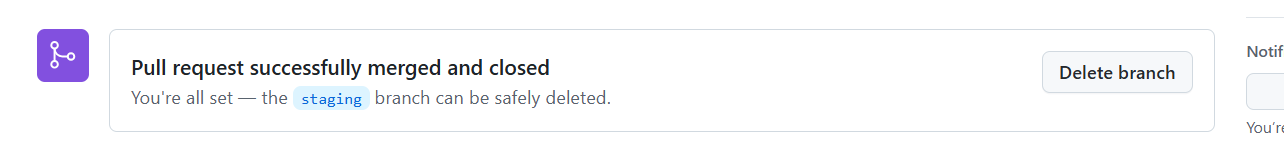
**Merge bugfix → develop**

****

**Merge develop → staging**

****

**Merge staging → main**

****

## Task 12: Code Review Workflow

## Create a Pull Request (PR) — feature → main

## ****Goal:**** open the PR with a clear title & description.

## 

## Assign a reviewer

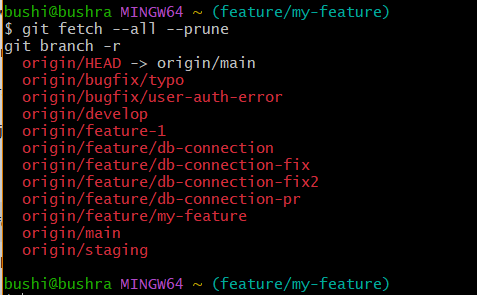
## 

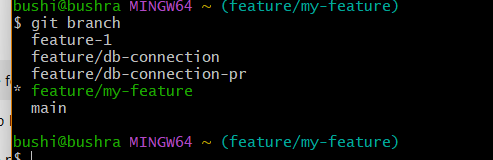
## ****Author Actions:****

## 

## Task 13: Branch Cleanup Best Practices

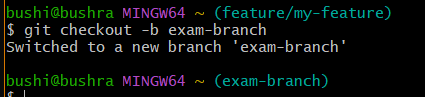
1. **Delete Remote Branch After Merge:**



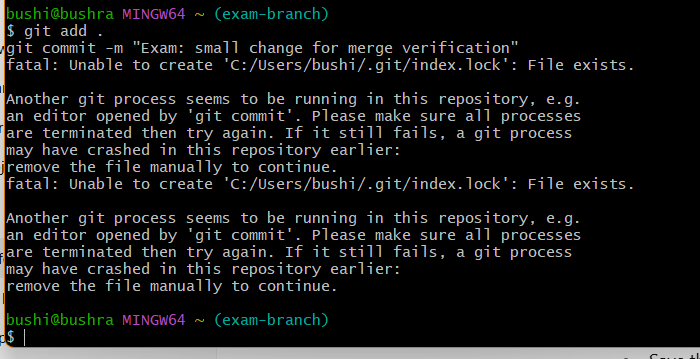
****

**Exam Evaluation Questions**

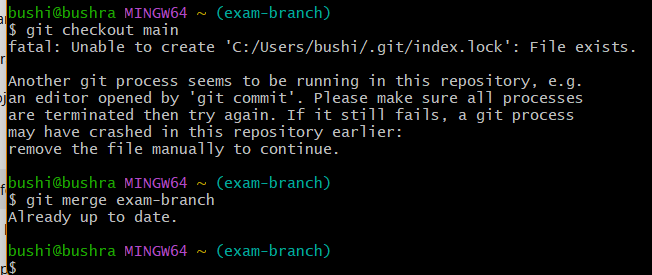
1. **Advanced Branching & Merge Verification**
   * Create a new branch in your repository

****

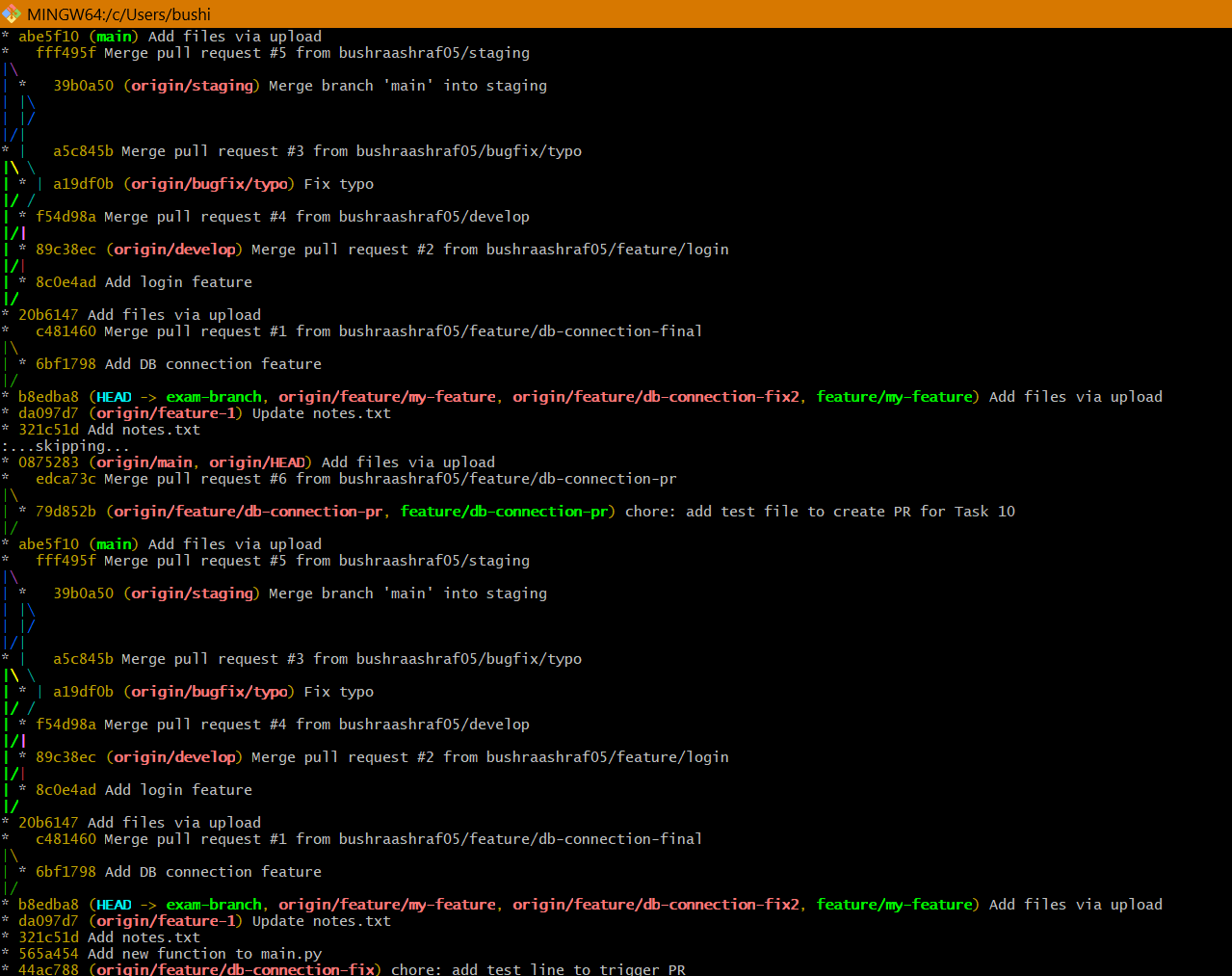
Make a small change in a file and commit it



Merge this branch back into the main branch.

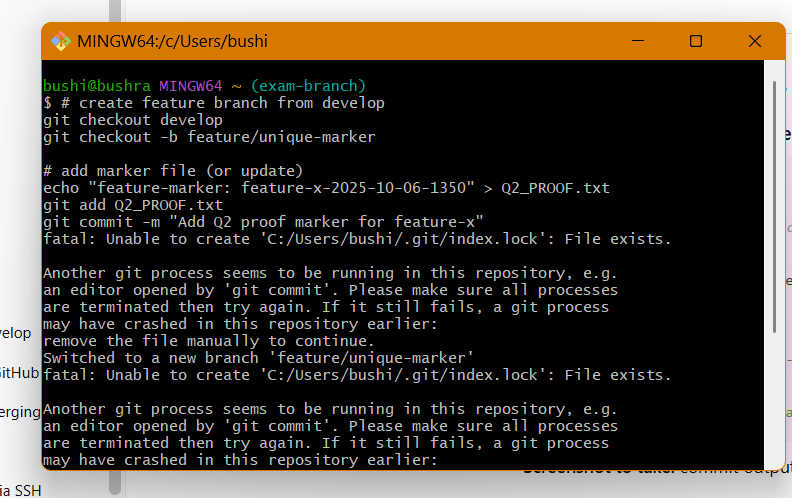


Show the history of commits to verify the merge

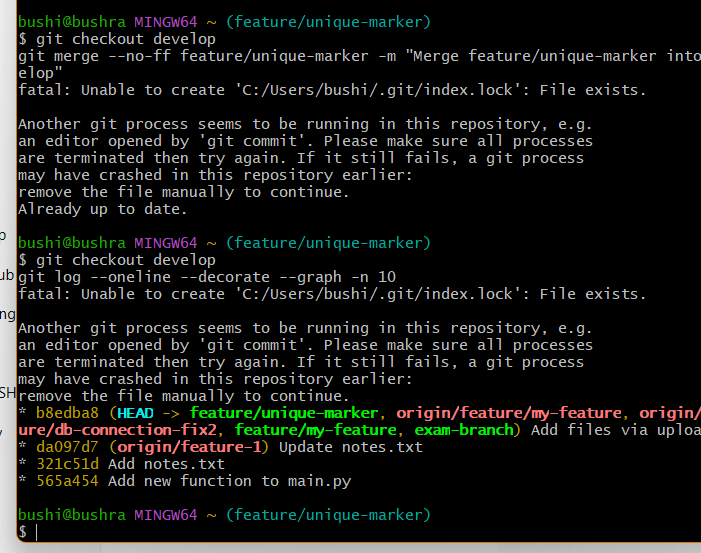


1. **Multi-Stage Workflow Simulation**

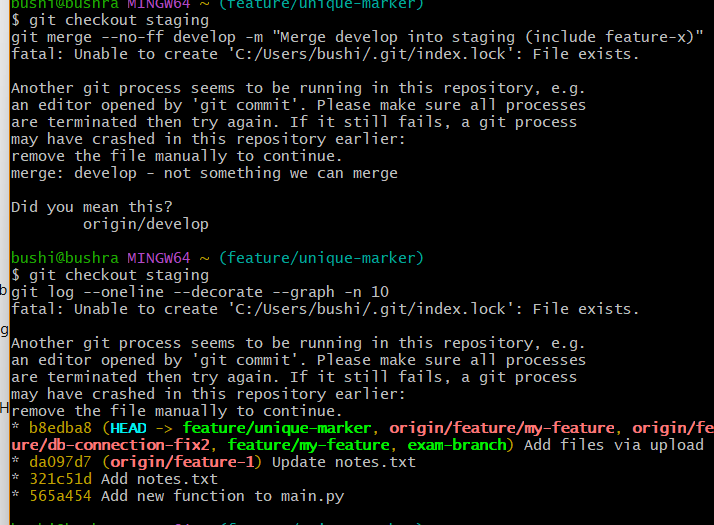
* Set up a branching workflow with three branches: main, develop, and staging



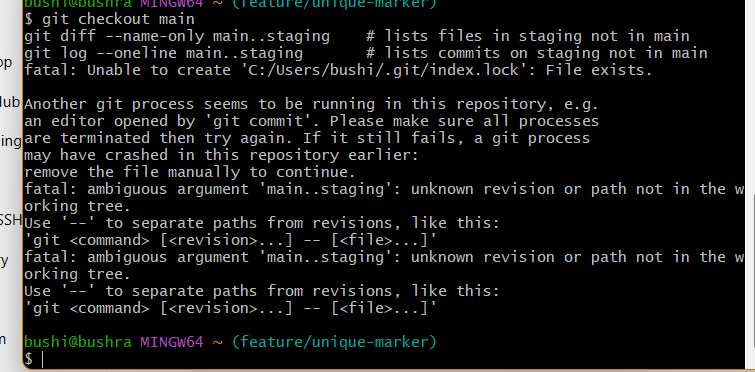
Create a feature branch from develop, make changes, and commit them.



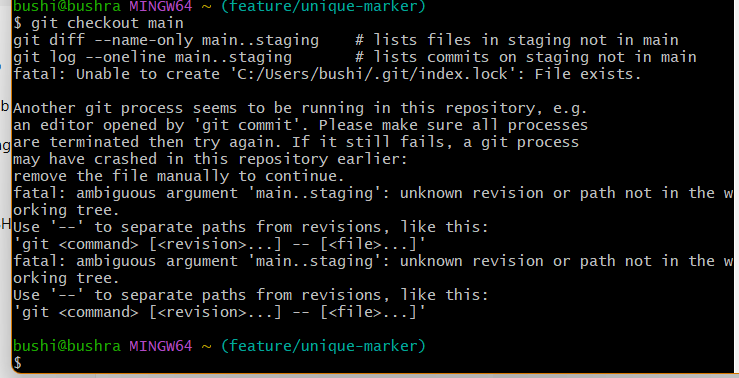
Merge develop into staging



Merge staging into main.

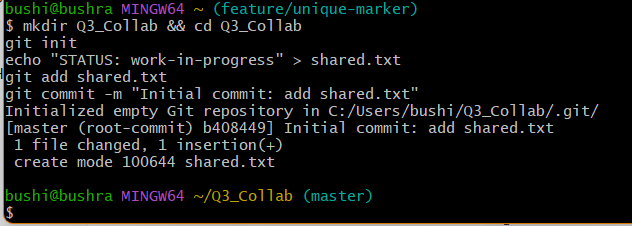


Provide proof that each stage contains the updated changes before it reaches main.

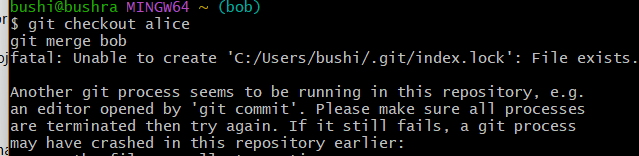


**Collaboration & Conflict Resolution**

* Work with a collaborator: both contributors should modify the same file but in separate branches



Attempt to merge the branches and capture the conflict.



Provide evidence that the final version of the file contains both collaborators’ changes.

