Cloud Computing Lab

Submitted To:

Sir Shoaib & Sir Waqas

Submitted By:

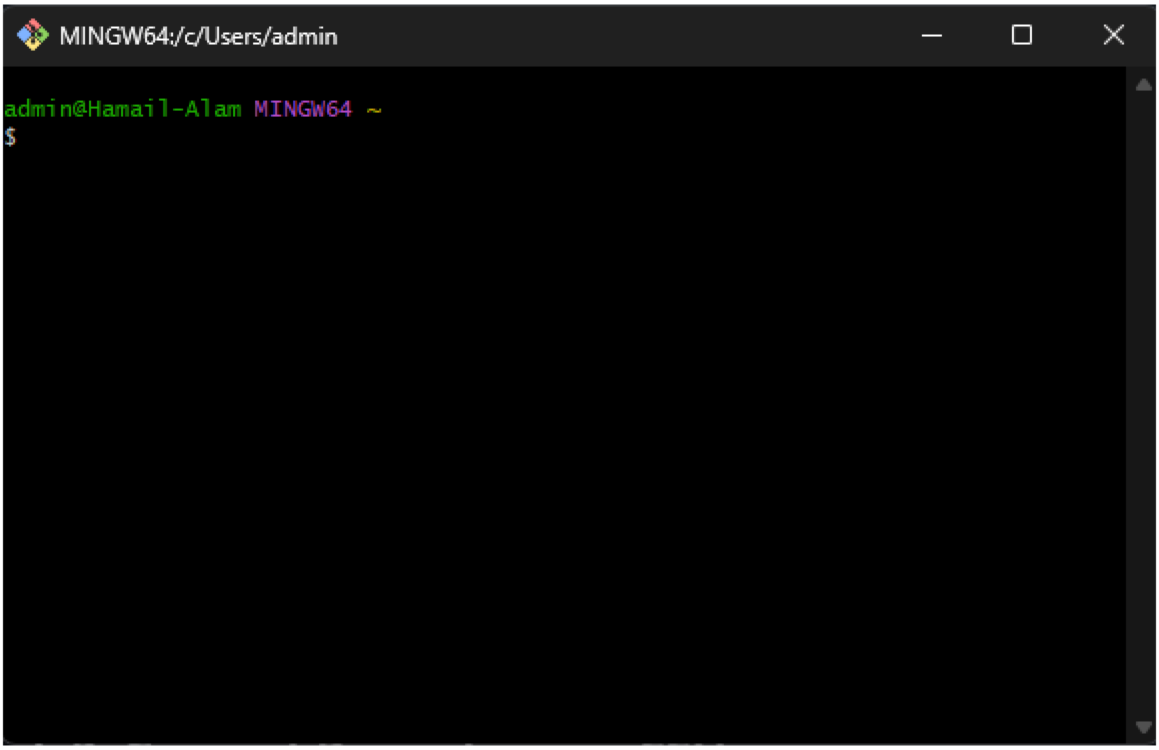
Hamail Fatima

Roll No:

2023-BSE-023

Section:

1. **Install Git on your PC.**
    - Download Git for Windows, Mac, or Linux
    - Take a screenshot of your installation and save it as git_installation.png in your Lab2 repo.

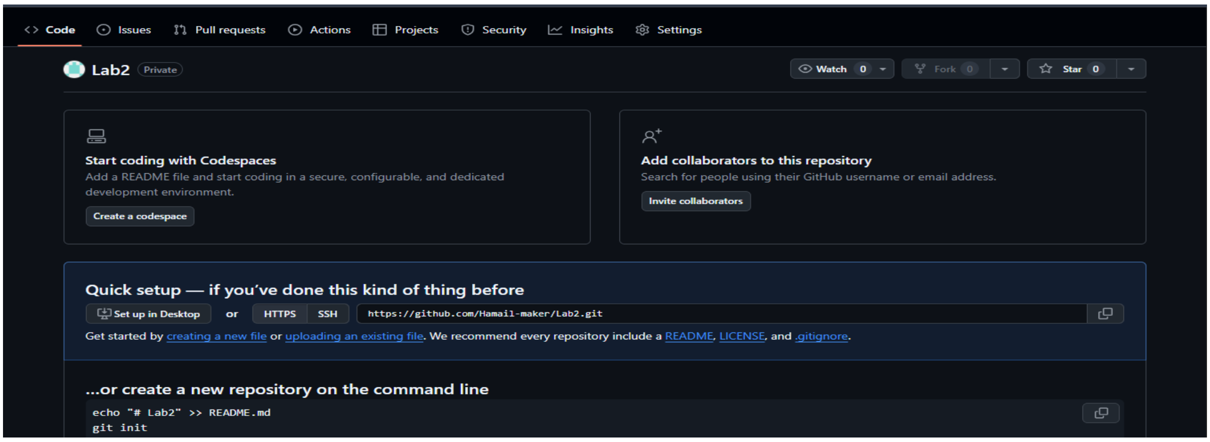


```
admin@Hamail-Alam MINGW64 ~
$
```

Task 1: Create Private GitHub Repository

1.Create a new private repository named Lab2 on GitHub.

2.Take a screenshot of your repo settings showing it's private. Save as repo_private.png in your Lab2 repo.



Start coding with Codespaces

Add a README file and start coding in a secure, configurable, and dedicated development environment.

Add collaborators to this repository

Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

https://github.com/Hamail-maker/Lab2.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# Lab2" >> README.md
git init
```

Task 2: Connect Repository via SSH

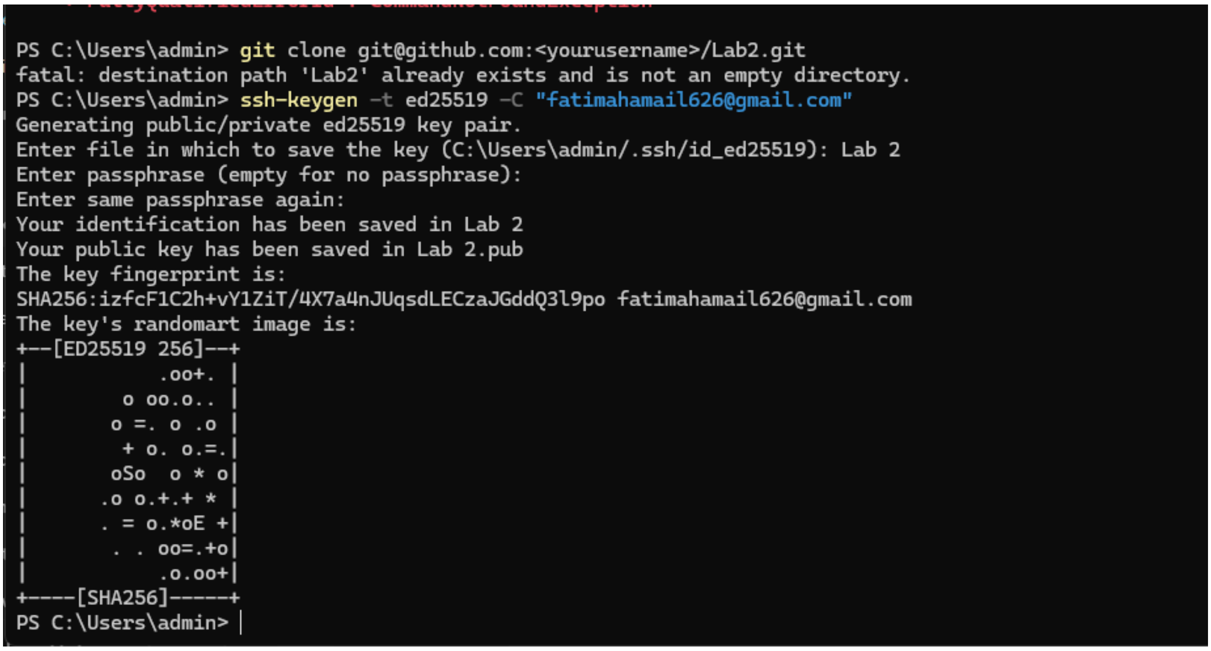
1.Generate a new SSH key using PowerShell:

ssh-keygen -t ed25519 -C your_email@example.com

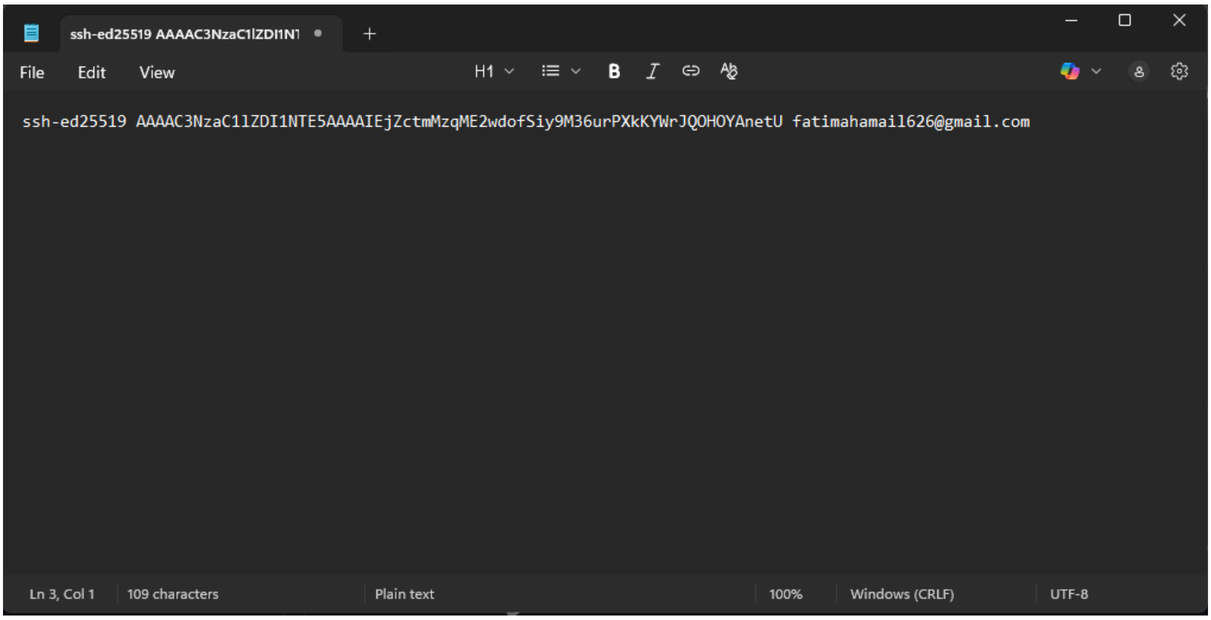
Save a screenshot of your terminal after key generation as ssh_keygen.png.

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

Take a screenshot of the added key as github_sshkey.png.



```
   + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\admin> git clone git@github.com:<yourusername>/Lab2.git
fatal: destination path 'Lab2' already exists and is not an empty directory.
PS C:\Users\admin> ssh-keygen -t ed25519 -C "fatimahamail626@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\admin/.ssh/id_ed25519): Lab 2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in Lab 2
Your public key has been saved in Lab 2.pub
The key fingerprint is:
SHA256:izfcF1C2h+vY1ZiT/4X7a4nJUqsdLECzaJGddQ3l9po fatimahamail626@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|           .oo+. |
|        o oo.o.. |
|       o =. o .o |
|        + o. o.=.|
|       oSo  o * o|
|      .o o.+.+ * |
|      . = o.*oE +|
|       . . oo=.+o|
|           .o.oo+|
+----[SHA256]-----+
PS C:\Users\admin> |
```

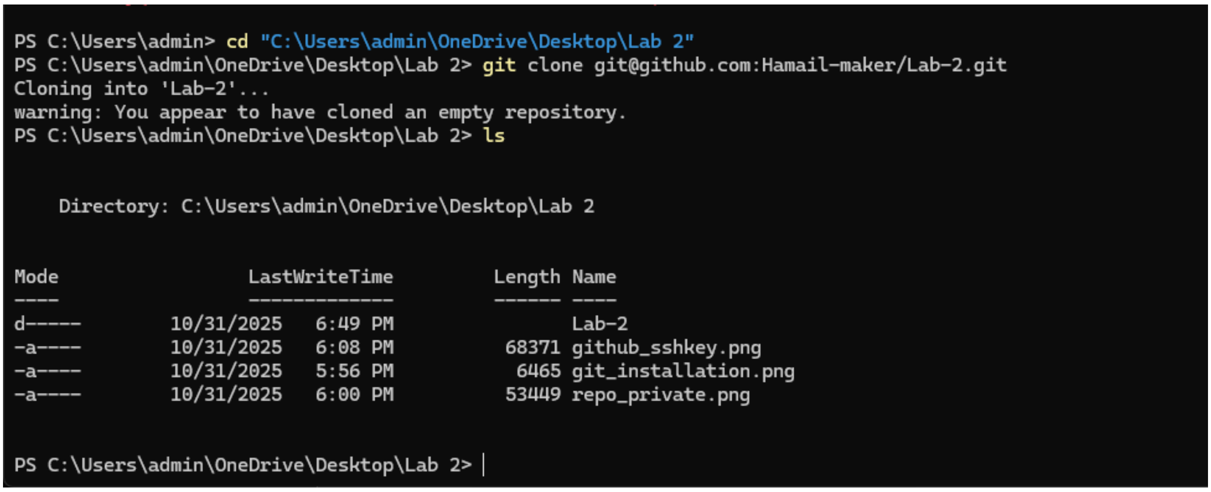


```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEjZctmMzqME2wdofSiy9M36urPXkKYWrJQOHOYAnetU fatimahamail626@gmail.com
```

2. Clone your Lab2 repo using SSH.

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

Save a screenshot as ssh_clone.png.



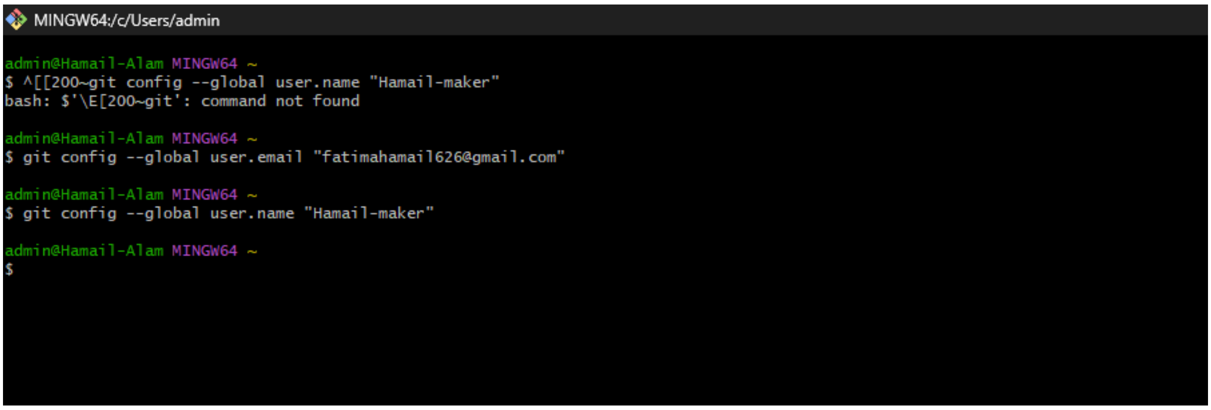
```
PS C:\Users\admin> cd "C:\Users\admin\OneDrive\Desktop\Lab 2"
PS C:\Users\admin\OneDrive\Desktop\Lab 2> git clone git@github.com:Hamail-maker/Lab-2.git
Cloning into 'Lab-2'...
warning: You appear to have cloned an empty repository.
PS C:\Users\admin\OneDrive\Desktop\Lab 2> ls


    Directory: C:\Users\admin\OneDrive\Desktop\Lab 2


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/31/2025   6:49 PM                Lab-2
-a----        10/31/2025   6:08 PM          68371 github_sshkey.png
-a----        10/31/2025   5:56 PM           6465 git_installation.png
-a----        10/31/2025   6:00 PM          53449 repo_private.png


PS C:\Users\admin\OneDrive\Desktop\Lab 2> |
```

Task 3: Configure Git Username and Email

1. **Set up your Git identity (this ensures all commits are linked to you):**
2. git config --global user.name "Your Name"

git config --global user.email "your_email@example.com"

- ○ Save a screenshot of your terminal as git_identity.png.

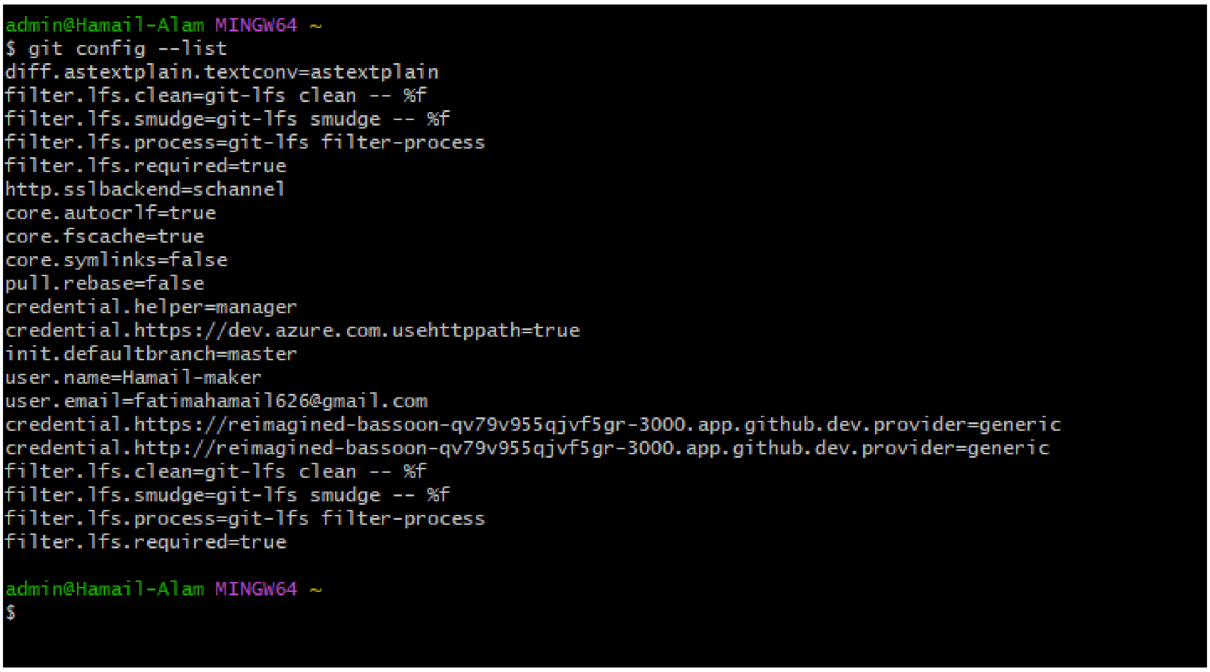


```
admin@Hamail-Alam MINGW64 ~
$ ^[[200~git config --global user.name "Hamail-maker"
bash: $'\E[200~git': command not found

admin@Hamail-Alam MINGW64 ~
$ git config --global user.email "fatimahamail626@gmail.com"

admin@Hamail-Alam MINGW64 ~
$ git config --global user.name "Hamail-maker"

admin@Hamail-Alam MINGW64 ~
$
```

1. **Verify your configuration:**

   git config --list

- ○ Screenshot as git_config_list.png.



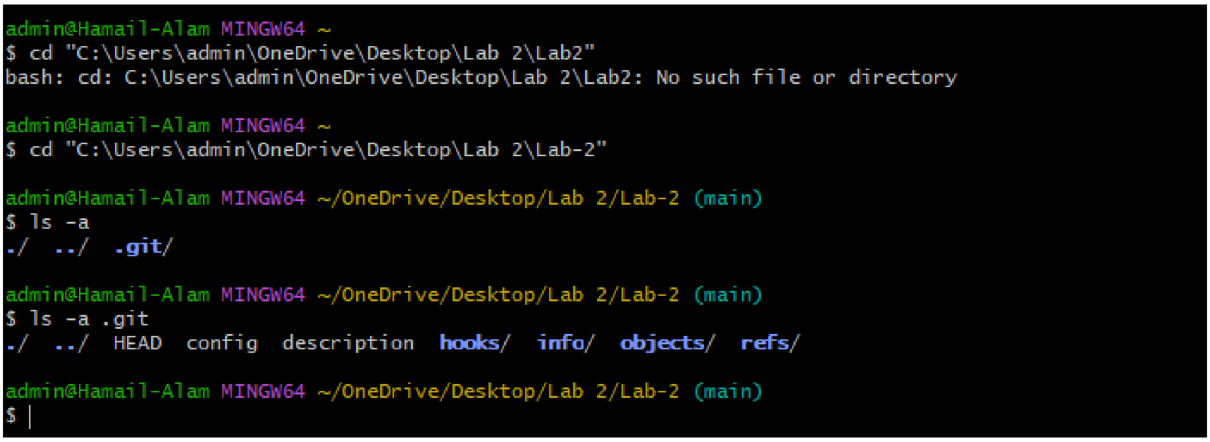
```
admin@Hamail-Alam MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Hamail-maker
user.email=fatimahamail626@gmail.com
credential.https://reimagined-bassoon-qv79v955qjvf5gr-3000.app.github.dev.provider=generic
credential.http://reimagined-bassoon-qv79v955qjvf5gr-3000.app.github.dev.provider=generic
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true

admin@Hamail-Alam MINGW64 ~
$
```

Task 4: Explore the .git Folder

1. Navigate into your cloned repository folder.
2. Show hidden files and locate the .git directory.
3. Explore what's inside using:

ls -a .git

- ○ Take a screenshot as git_folder.png.



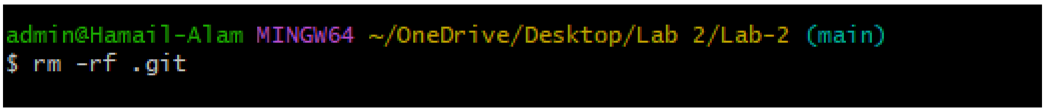
```
admin@Hamail-Alam MINGW64 ~
$ cd "C:\Users\admin\OneDrive\Desktop\Lab 2\Lab2"
bash: cd: C:\Users\admin\OneDrive\Desktop\Lab 2\Lab2: No such file or directory

admin@Hamail-Alam MINGW64 ~
$ cd "C:\Users\admin\OneDrive\Desktop\Lab 2\Lab-2"

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ ls -a
./  ../  .git/

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ ls -a .git
./  ../  HEAD  config  description  hooks/  info/  objects/  refs/

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ |
```

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

Task 5: Local Repository Management

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

rm -rf .git

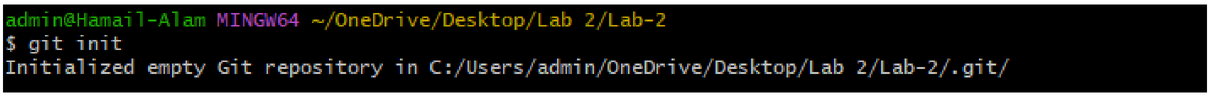


```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ rm -rf .git
```

○ Take a screenshot as delete_git.png.

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

git init



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2
$ git init
Initialized empty Git repository in C:/Users/admin/OneDrive/Desktop/Lab 2/Lab-2/.git/
```

○ Save a screenshot as git_init.png.

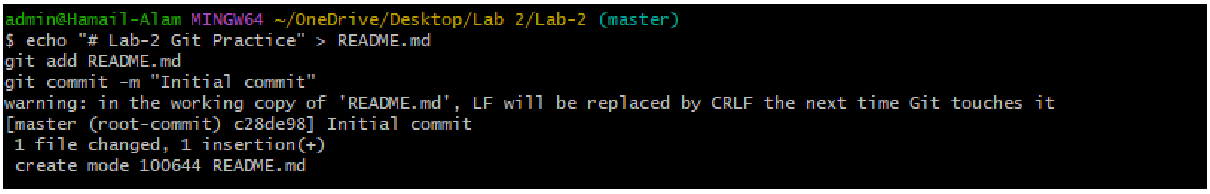
3. **Add a file named README.md and commit it:**

4. echo "# Lab2 Git Practice" > README.md

5. git add README.md

git commit -m "Initial commit"

○ Save a screenshot as first_commit.png.



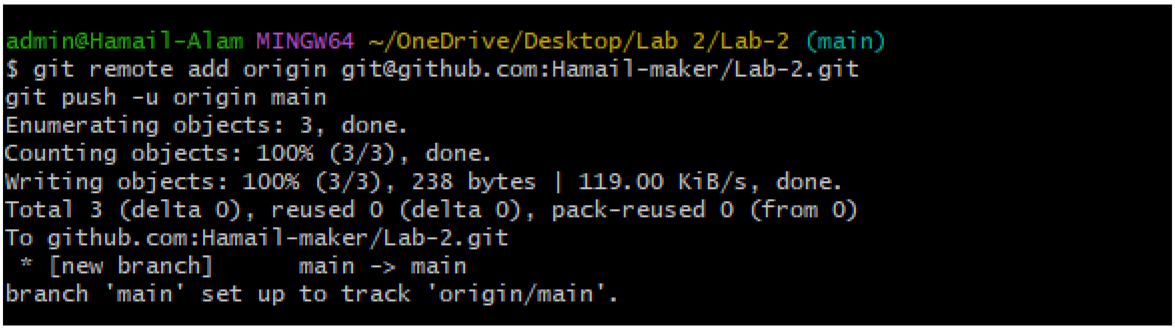
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (master)
$ echo "# Lab-2 Git Practice" > README.md
git add README.md
git commit -m "Initial commit"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[master (root-commit) c28de98] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

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

7. git remote add origin git@github.com:<yourusername>/Lab2.git

git push -u origin main

○ Save a screenshot as first_push.png.



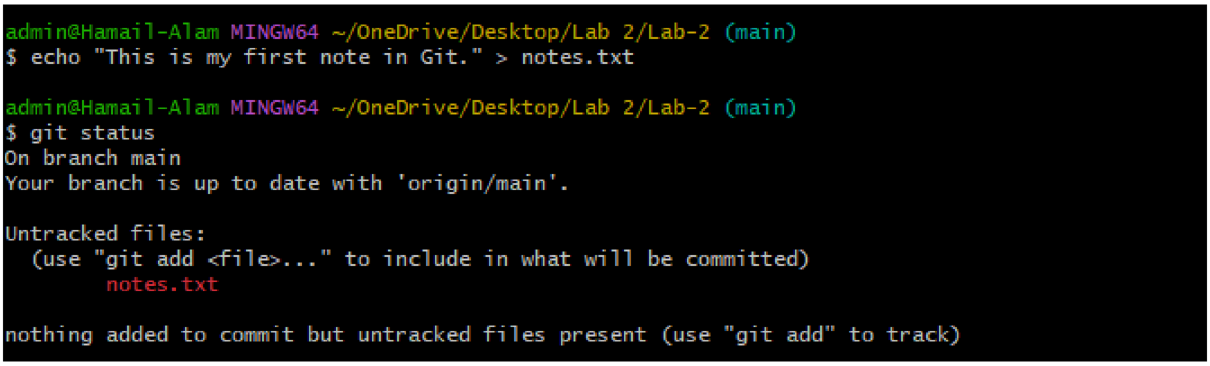
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git remote add origin git@github.com:Hamail-maker/Lab-2.git
git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 238 bytes | 119.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Task 6: File Status & Staging

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

git status

- Screenshot as status1.png.

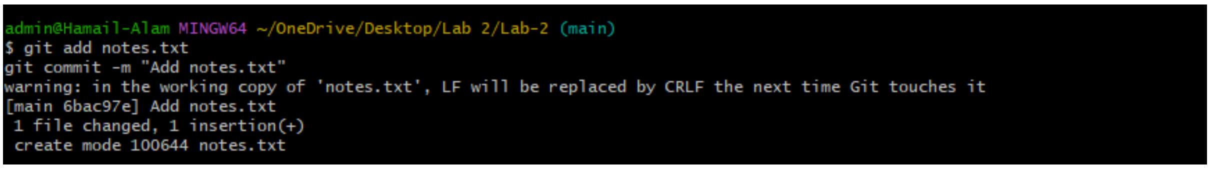


```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ echo "This is my first note in Git." > notes.txt

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        notes.txt

nothing added to commit but untracked files present (use "git add" to track)
```

3. Stage and commit:
4. git add notes.txt

git commit -m "Add notes.txt"

- Screenshot as commit_notes.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git add notes.txt
git commit -m "Add notes.txt"
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the next time Git touches it
[main 6bac97e] Add notes.txt
 1 file changed, 1 insertion(+)
 create mode 100644 notes.txt
```

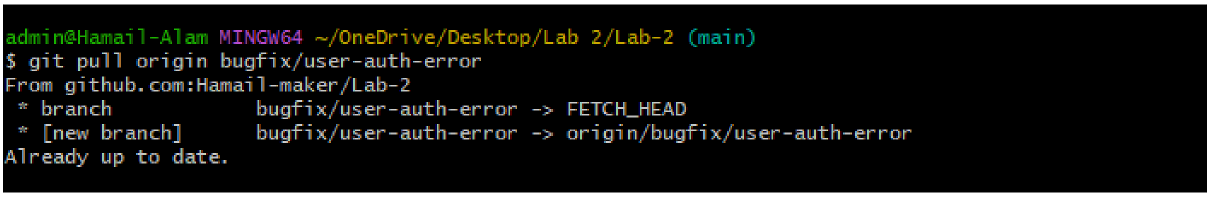
5.Edit notes.txt and repeat status/commit steps.

Task 7: Branch Creation Using GitHub GUI

1. **On GitHub (web interface), create a branch named bugfix/user-auth-error.**
2. **Pull the branch to your local repository to sync.**

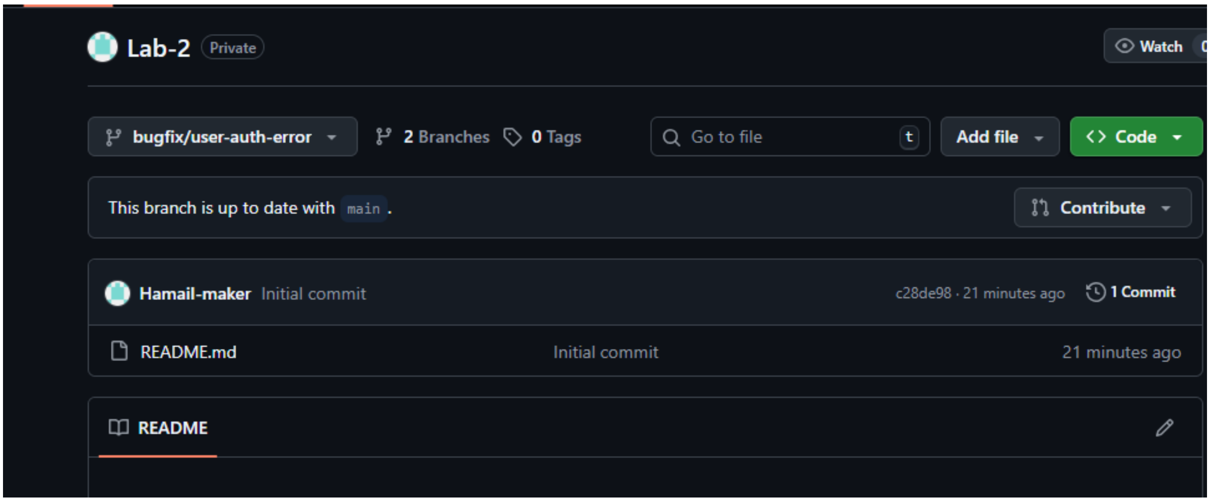
git pull origin bugfix/user-auth-error

- Take a screenshot of branch creation in the GUI as bugfix_branch_gui.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git pull origin bugfix/user-auth-error
From github.com:Hamail-maker/Lab-2
 * branch            bugfix/user-auth-error -> FETCH_HEAD
 * [new branch]      bugfix/user-auth-error -> origin/bugfix/user-auth-error
Already up to date.
```

- Take a screenshot of your local git branch list after pulling as bugfix_branch_local.png.



This branch is up to date with main .

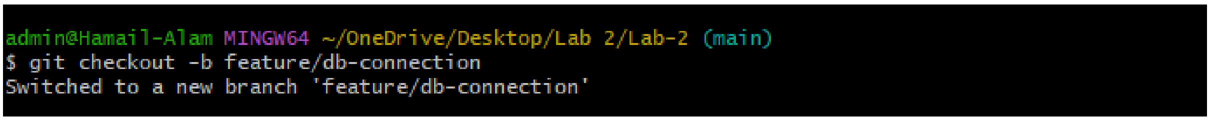
Hamail-maker Initial commit c28de98 · 21 minutes ago 1 Commit

README.md Initial commit 21 minutes ago

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

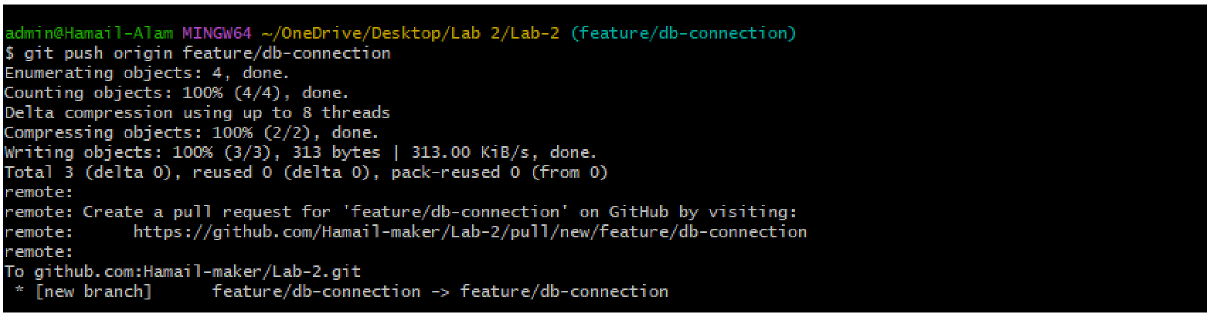


```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git checkout -b feature/db-connection
Switched to a new branch 'feature/db-connection'
```

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

   git push origin feature/db-connection

- ○ Take a screenshot of the branch creation and push as feature_db_branch.png.



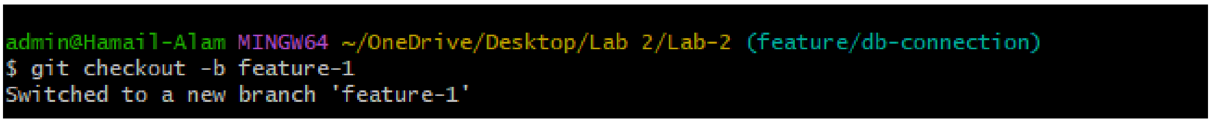
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/db-connection)
$ git push origin feature/db-connection
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 313.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/db-connection' on GitHub by visiting:
remote:      https://github.com/Hamail-maker/Lab-2/pull/new/feature/db-connection
remote:
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      feature/db-connection -> feature/db-connection
```

Task 9: Branching & Merging

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

   git checkout -b feature-1

- ○ Screenshot as branch_create.png.



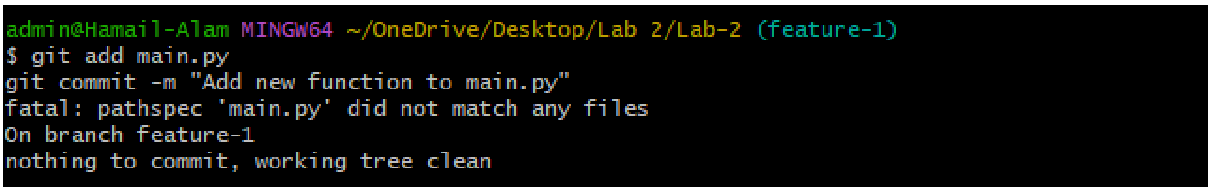
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/db-connection)
$ git checkout -b feature-1
Switched to a new branch 'feature-1'
```

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

3. git add main.py

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

- ○ Screenshot as feature_commit.png.



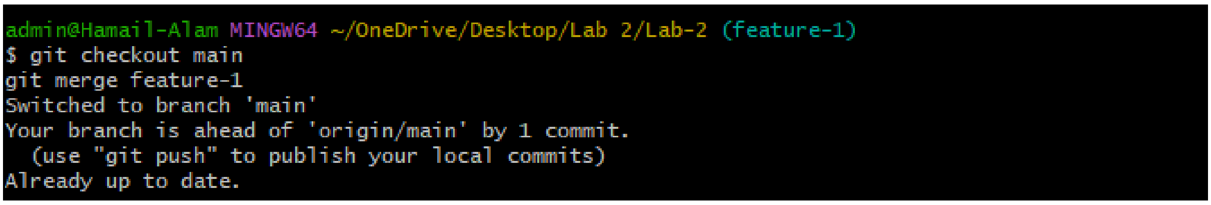
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature-1)
$ git add main.py
git commit -m "Add new function to main.py"
fatal: pathspec 'main.py' did not match any files
On branch feature-1
nothing to commit, working tree clean
```

4. Switch back to main and merge:

5. git checkout main

   git merge feature-1

- ○ Screenshot as merge.png.



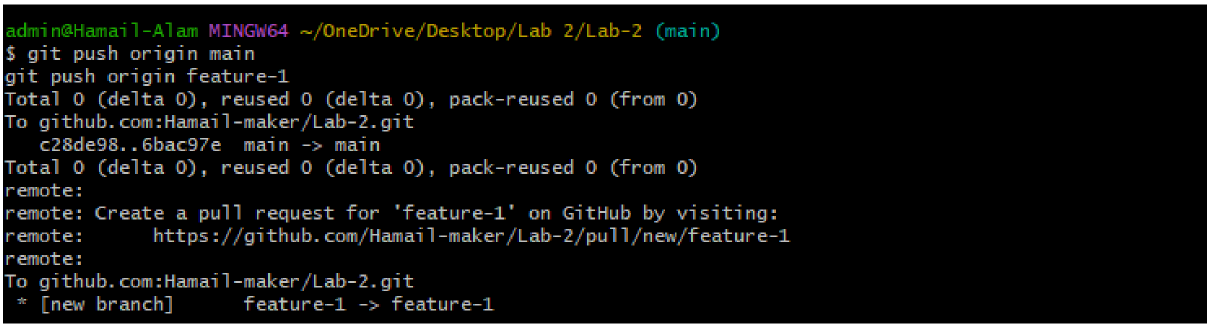
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature-1)
$ git checkout main
git merge feature-1
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
Already up to date.
```

6. Push all branches:

7. git push origin main

   git push origin feature-1

- ○ Screenshot as push_branches.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git push origin main
git push origin feature-1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   c28de98..6bac97e  main -> main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-1' on GitHub by visiting:
remote:      https://github.com/Hamail-maker/Lab-2/pull/new/feature-1
remote:
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      feature-1 -> feature-1
```

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

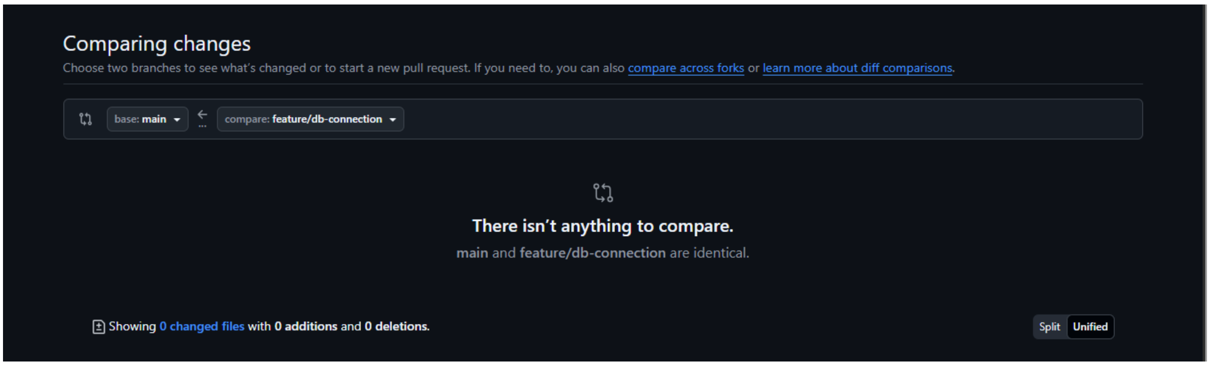
1. **On GitHub, create a Pull Request from the branch feature/db-connection to main.**

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

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

- ○ Take a screenshot of:

- The Pull Request creation (pr_creation.png)



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

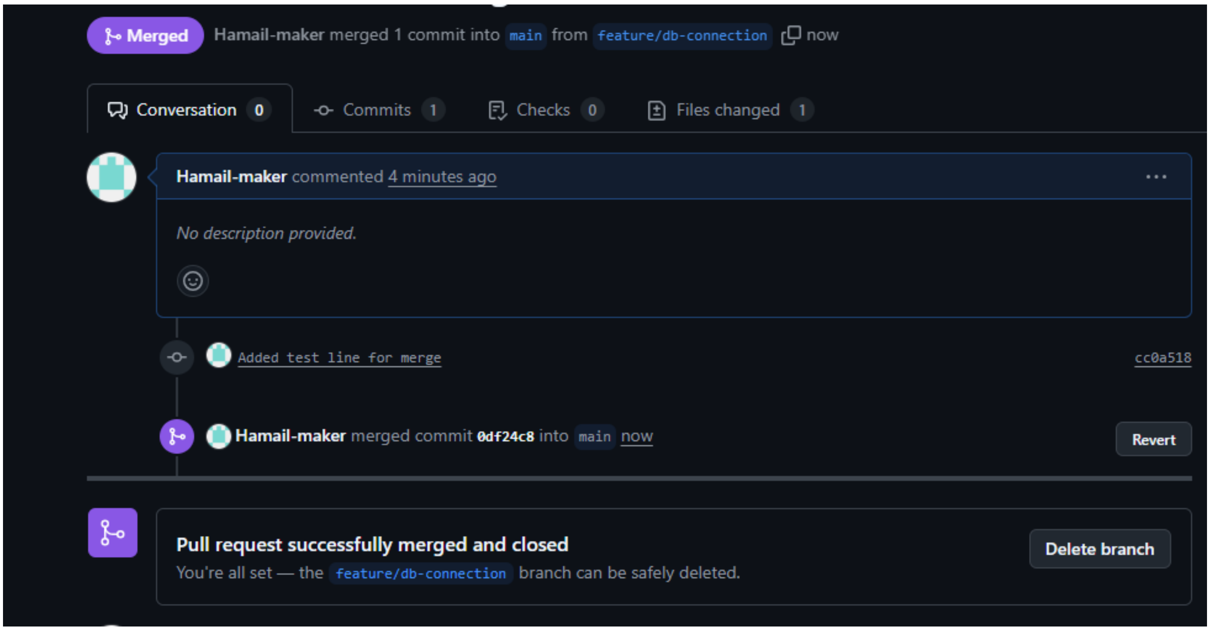
base: main ← compare: feature/db-connection

There isn't anything to compare.

main and feature/db-connection are identical.

Showing 0 changed files with 0 additions and 0 deletions.

- The Pull Request review and merge (pr_merge.png)



Merged Hamail-maker merged 1 commit into main from feature/db-connection now

Hamail-maker commented 4 minutes ago

No description provided.

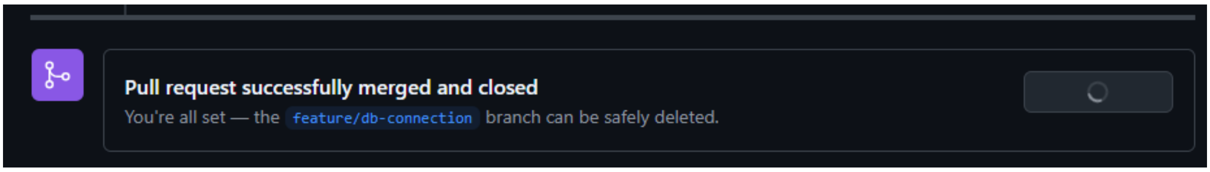
Added test line for merge cc0a518

Hamail-maker merged commit 0df24c8 into main now

Pull request successfully merged and closed

You're all set — the feature/db-connection branch can be safely deleted.

- The branch deletion (branch_delete.png)



Pull request successfully merged and closed

You're all set — the feature/db-connection branch can be safely deleted.

Task 11: Detailed Branch Strategy (Develop/Staging)

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

- develop
- staging
- feature/*

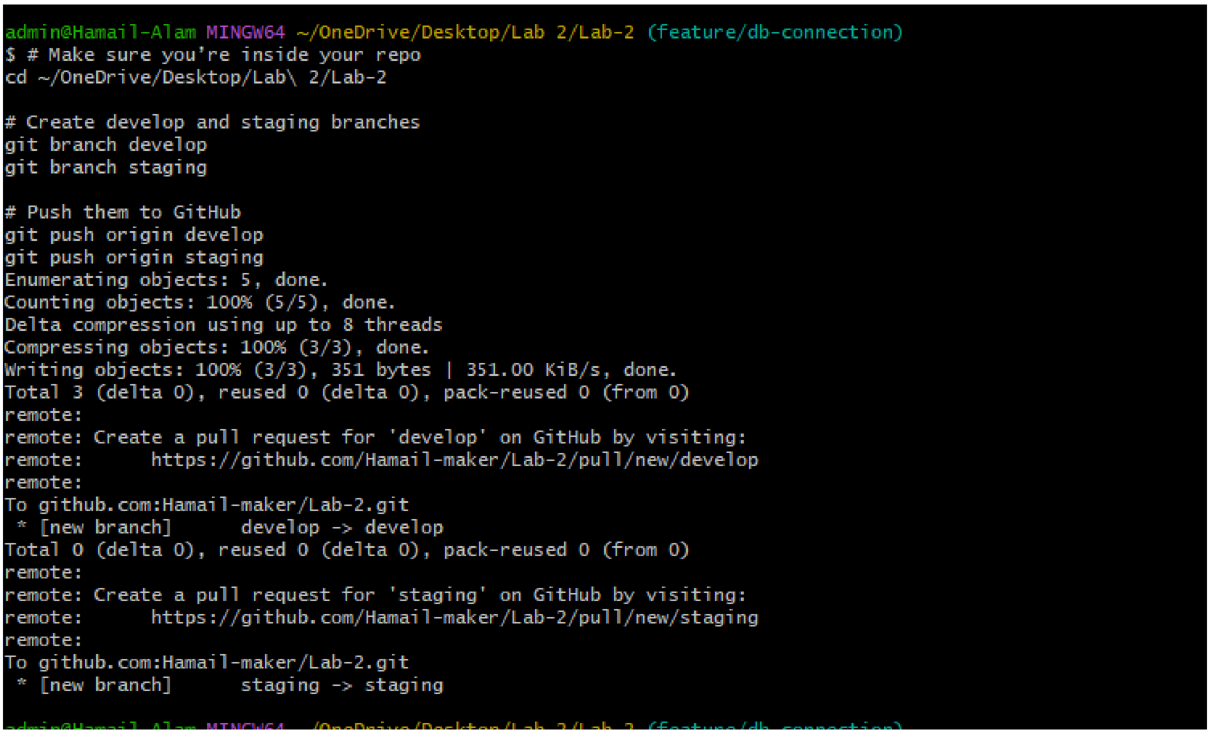
- bugfix/*

2. Example workflow:

- Developers work on feature/* and bugfix/* branches.
- Merge into develop after completion.
- develop is merged into staging for testing.
- Finally, staging is merged into main (production).

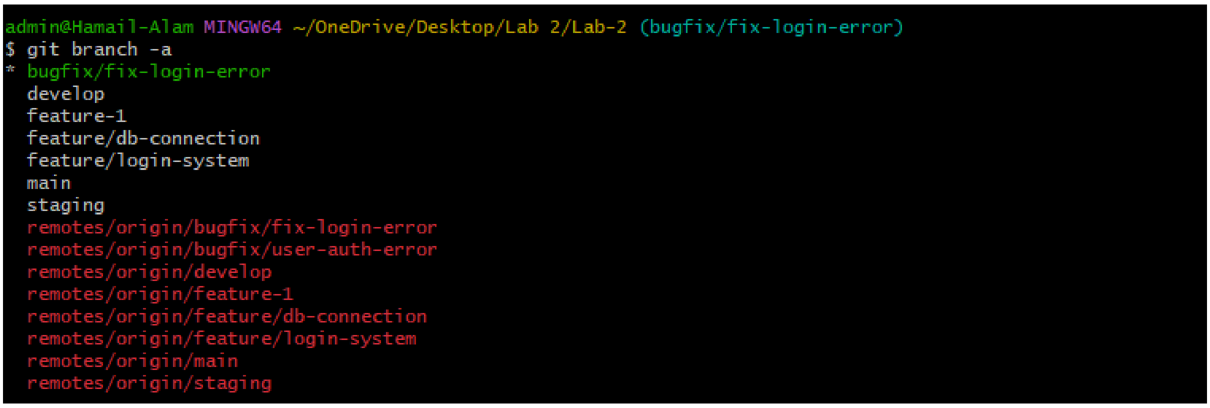
3. Document the workflow with screenshots:

- Branch list (branch_strategy.png)

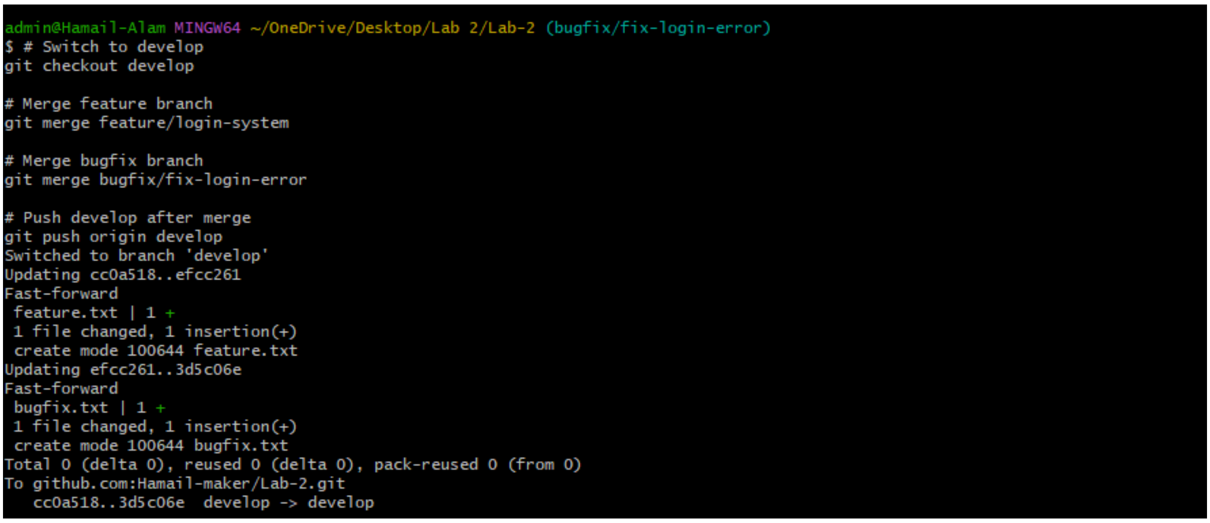


```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/db-connection)
$ # Make sure you're inside your repo
cd ~/OneDrive/Desktop/Lab\ 2/Lab-2

# Create develop and staging branches
git branch develop
git branch staging

# Push them to GitHub
git push origin develop
git push origin staging
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 351 bytes | 351.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/Hamail-maker/Lab-2/pull/new/develop
remote:
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      develop -> develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'staging' on GitHub by visiting:
remote:      https://github.com/Hamail-maker/Lab-2/pull/new/staging
remote:
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      staging -> staging

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/db-connection)
```

- Merges into develop and staging (branch_merges.png)

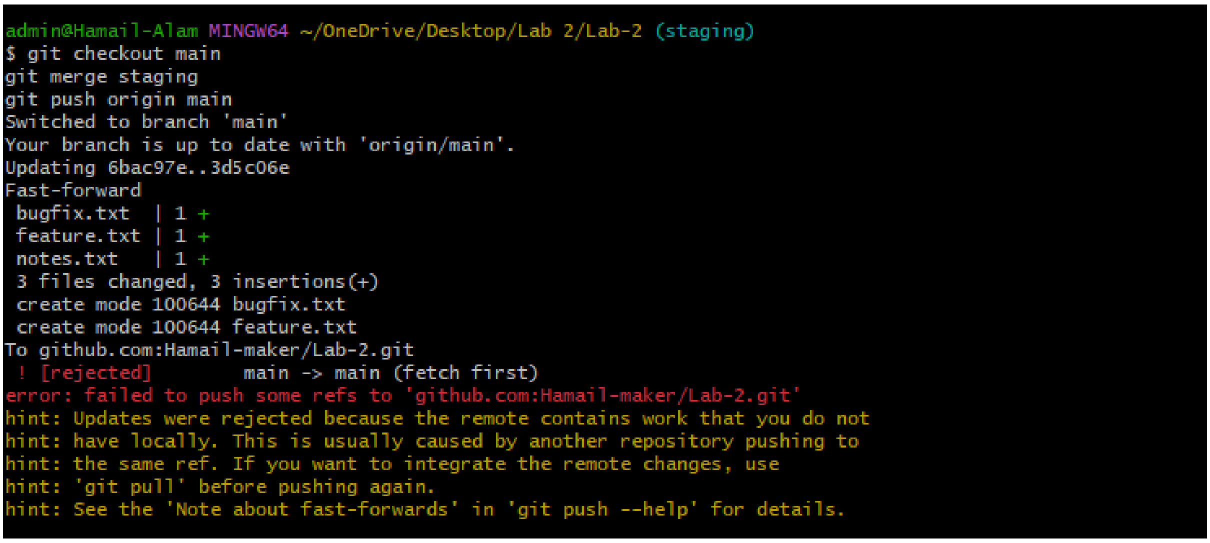


```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (bugfix/fix-login-error)
$ git branch -a
* bugfix/fix-login-error
  develop
  feature-1
  feature/db-connection
  feature/login-system
  main
  staging
  remotes/origin/bugfix/fix-login-error
  remotes/origin/bugfix/user-auth-error
  remotes/origin/develop
  remotes/origin/feature-1
  remotes/origin/feature/db-connection
  remotes/origin/feature/login-system
  remotes/origin/main
  remotes/origin/staging
```



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (bugfix/fix-login-error)
$ # Switch to develop
git checkout develop

# Merge feature branch
git merge feature/login-system

# Merge bugfix branch
git merge bugfix/fix-login-error

# Push develop after merge
git push origin develop
Switched to branch 'develop'
Updating cc0a518..efcc261
Fast-forward
 feature.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt
Updating efcc261..3d5c06e
Fast-forward
 bugfix.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 bugfix.txt
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   cc0a518..3d5c06e  develop -> develop
```

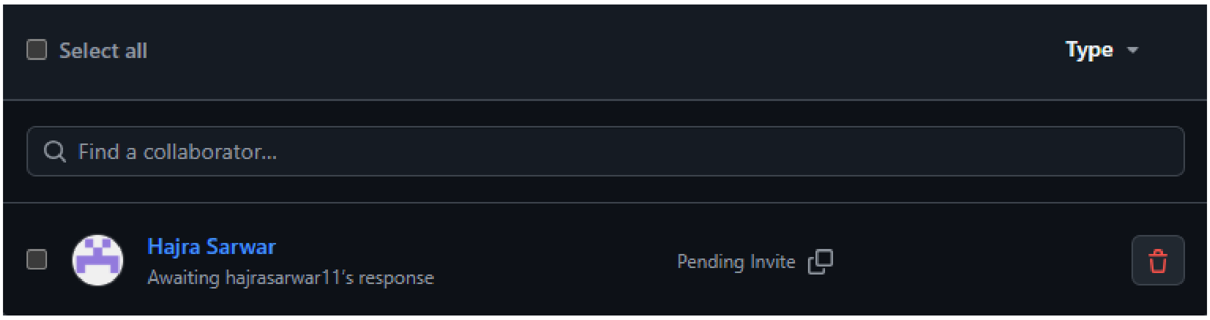
- Final merge into main (final_merge.png)



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (staging)
$ git checkout main
git merge staging
git push origin main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Updating 6bac97e..3d5c06e
Fast-forward
 bugfix.txt  | 1 +
 feature.txt | 1 +
 notes.txt   | 1 +
 3 files changed, 3 insertions(+)
 create mode 100644 bugfix.txt
 create mode 100644 feature.txt
To github.com:Hamail-maker/Lab-2.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:Hamail-maker/Lab-2.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Bonus Task: Simulated Team Collaboration

1. Create a branch collab.
2. Add a collaborator on your GitHub repo.

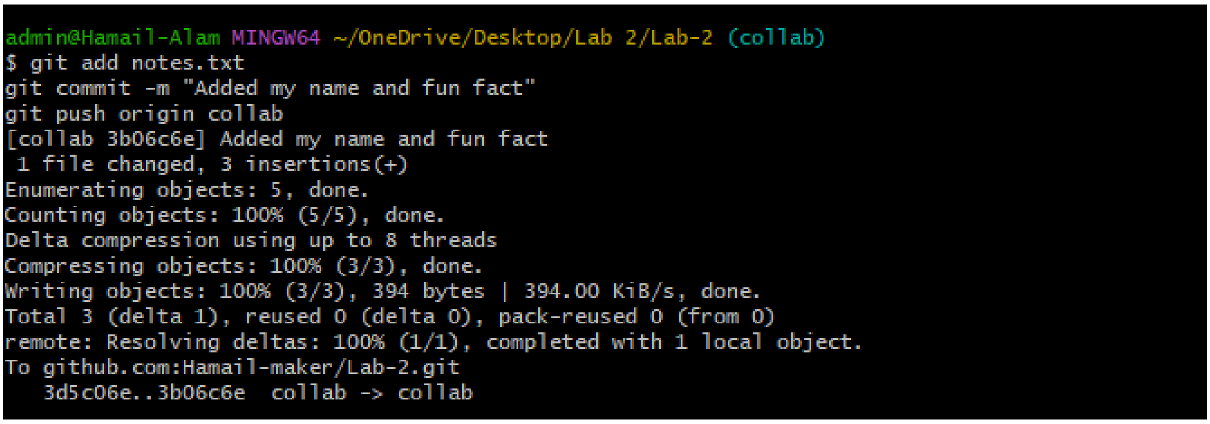


Hajra Sarwar
Awaiting hajrasarwar11's response

3. Both users:

- Pull latest changes.

- Add their name and fun fact to notes.txt.
- Commit and push to the collab branch.
- Screenshot as collab_commit.png.

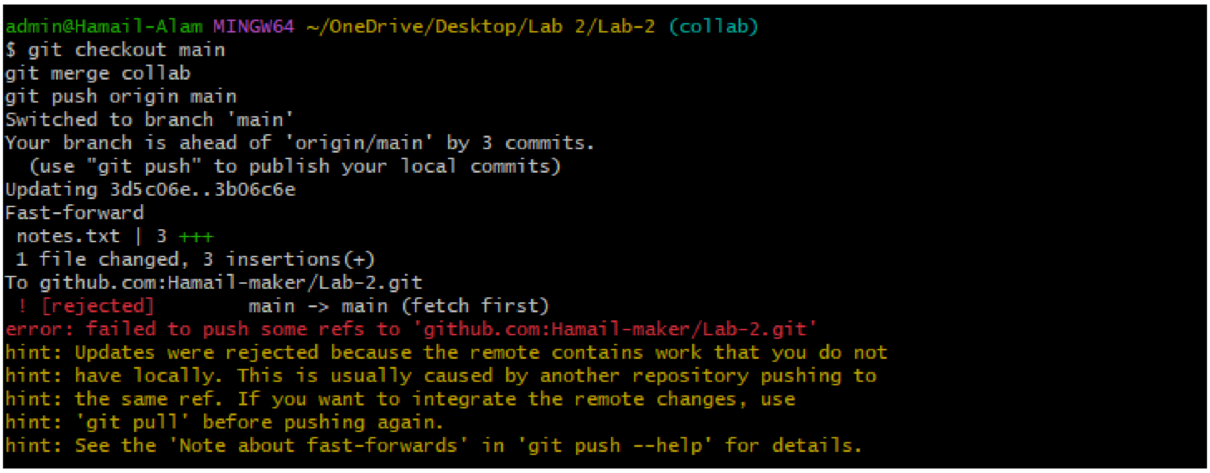


```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (collab)
$ git add notes.txt
git commit -m "Added my name and fun fact"
git push origin collab
[collab 3b06c6e] Added my name and fun fact
 1 file changed, 3 insertions(+)
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 394 bytes | 394.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Hamail-maker/Lab-2.git
   3d5c06e..3b06c6e  collab -> collab
```

4. Merge collab into main and push.
5. git checkout main
6. git merge collab

git push origin main

- Screenshot as collab_merge.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (collab)
$ git checkout main
git merge collab
git push origin main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)
Updating 3d5c06e..3b06c6e
Fast-forward
 notes.txt | 3 +++
 1 file changed, 3 insertions(+)
To github.com:Hamail-maker/Lab-2.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:Hamail-maker/Lab-2.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Task 11: Detailed Branch Strategy (Develop/Staging)

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

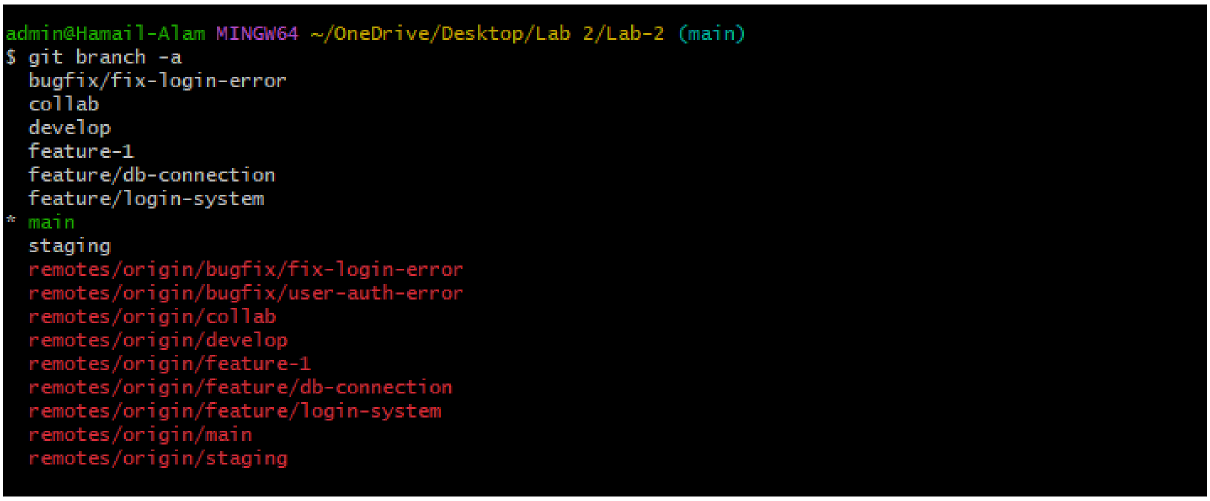
- develop
- staging
- feature/*
- bugfix/*

2. **Example Workflow:**

- Developers work on feature/* and bugfix/* branches.
- Merge completed work into develop.
- Merge develop into staging for testing.
- Finally, merge staging into main (production).

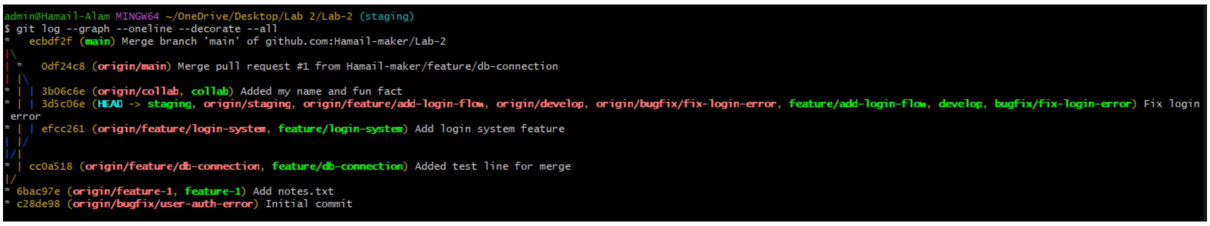
3. **Documentation with Screenshots:**

- Branch list → branch_strategy.png



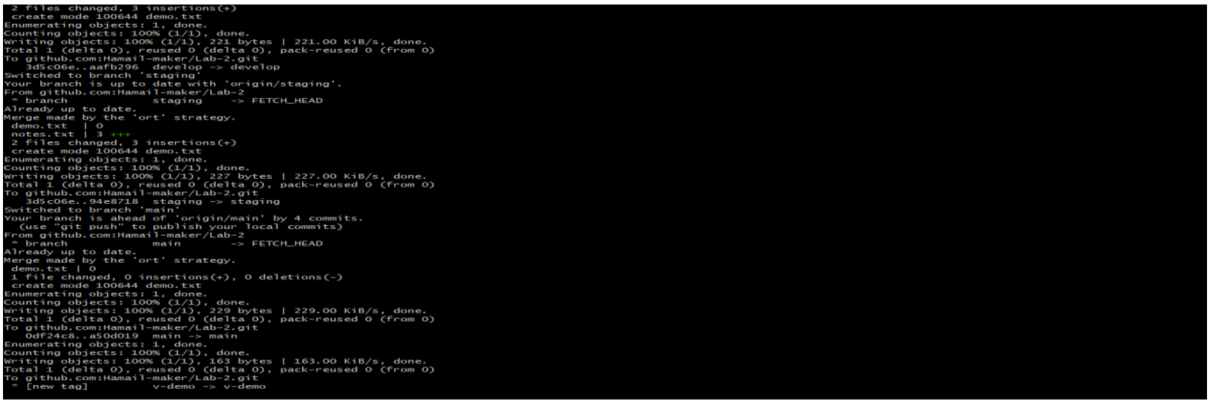
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git branch -a
  bugfix/fix-login-error
  collab
  develop
  feature-1
  feature/db-connection
  feature/login-system
* main
  staging
  remotes/origin/bugfix/fix-login-error
  remotes/origin/bugfix/user-auth-error
  remotes/origin/collab
  remotes/origin/develop
  remotes/origin/feature-1
  remotes/origin/feature/db-connection
  remotes/origin/feature/login-system
  remotes/origin/main
  remotes/origin/staging
```

- Merges into develop and staging → branch_merges.png



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (staging)
$ git log --graph --oneline --decorate --all
*   ecbdf2f (main) Merge branch 'main' of github.com:Hamail-maker/Lab-2
|\
| *   0df24c8 (origin/main) Merge pull request #1 from Hamail-maker/feature/db-connection
| |\
* | | 3b06c6e (origin/collab, collab) Added my name and fun fact
* | | 3d5c06e (HEAD -> staging, origin/staging, origin/feature/add-login-flow, origin/develop, origin/bugfix/fix-login-error, feature/add-login-flow, develop, bugfix/fix-login-error) Fix login
 error
* | | efcc261 (origin/feature/login-system, feature/login-system) Add login system feature
| |/
|/|
* | cc0a518 (origin/feature/db-connection, feature/db-connection) Added test line for merge
|/
* 6bac97e (origin/feature-1, feature-1) Add notes.txt
* c28de98 (origin/bugfix/user-auth-error) Initial commit
```

- Final merge into main → final_merge.png

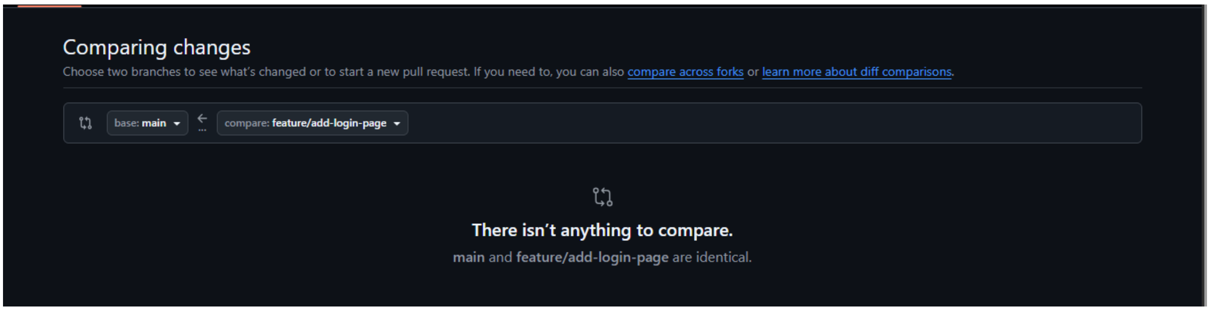


```
 2 files changed, 3 insertions(+)
 create mode 100644 demo.txt
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 221 bytes | 221.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   3d5c06e..aafb296  develop -> develop
Switched to branch 'staging'
Your branch is up to date with 'origin/staging'.
From github.com:Hamail-maker/Lab-2
 * branch            staging    -> FETCH_HEAD
Already up to date.
Merge made by the 'ort' strategy.
 demo.txt  | 0
 notes.txt | 3 +++
 2 files changed, 3 insertions(+)
 create mode 100644 demo.txt
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 227 bytes | 227.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   3d5c06e..94e8718  staging -> staging
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)
From github.com:Hamail-maker/Lab-2
 * branch            main       -> FETCH_HEAD
Already up to date.
Merge made by the 'ort' strategy.
 demo.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo.txt
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 229 bytes | 229.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   0df24c8..a50d019  main -> main
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 163 bytes | 163.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
 * [new tag]         v-demo -> v-demo
```

Task 12: Code Review Workflow

1. **Create a Pull Request (PR) / Merge Request (MR):**

- From a feature branch into main.
- Add a clear title and description summarizing the changes.
- Screenshot as pr_create_details.png.

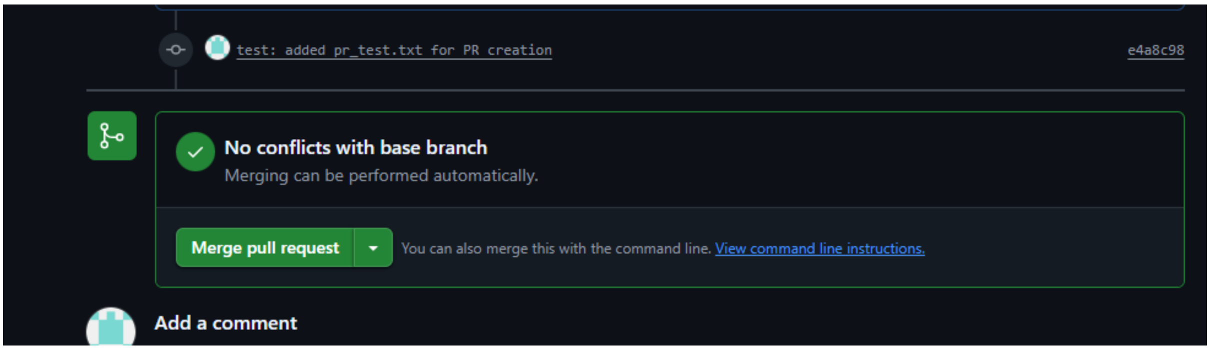


Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

There isn't anything to compare.

main and feature/add-login-page are identical.



test: added pr_test.txt for PR creation e4a8c98

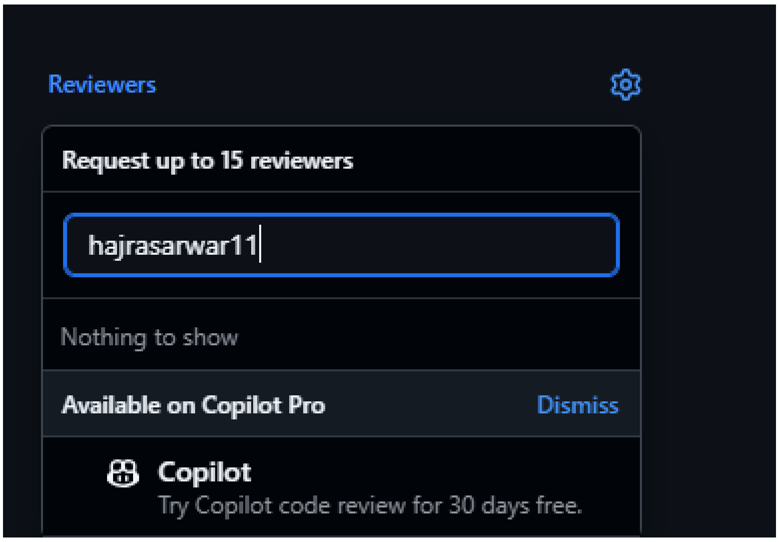
No conflicts with base branch

Merging can be performed automatically.

You can also merge this with the command line. View command line instructions.

2. **Assign a reviewer** (teammate or second account).

- Screenshot as pr_assigned_reviewer.png.



Request up to 15 reviewers

hajrasarwar11

Nothing to show

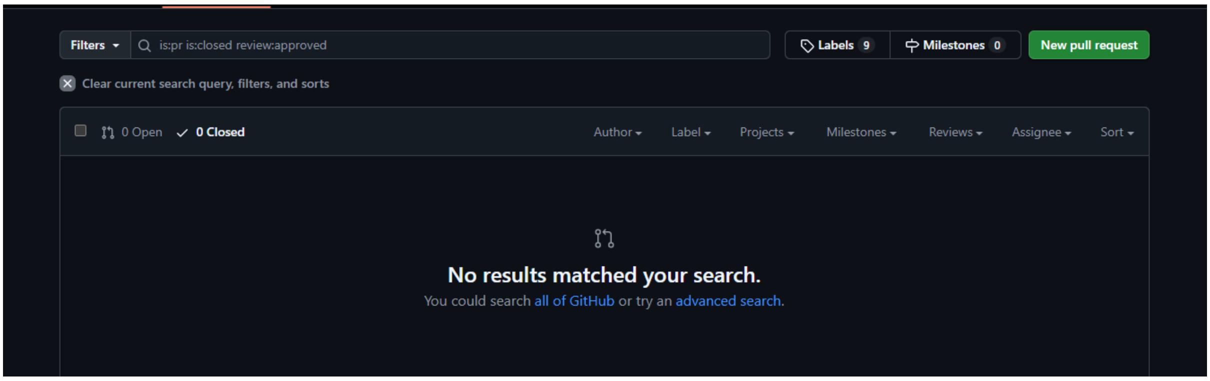
Available on Copilot Pro

Copilot

Try Copilot code review for 30 days free.

3. **Reviewer Actions (capture screenshots for each):**

- Approve the PR → pr_approved.png



is:pr is:closed review:approved

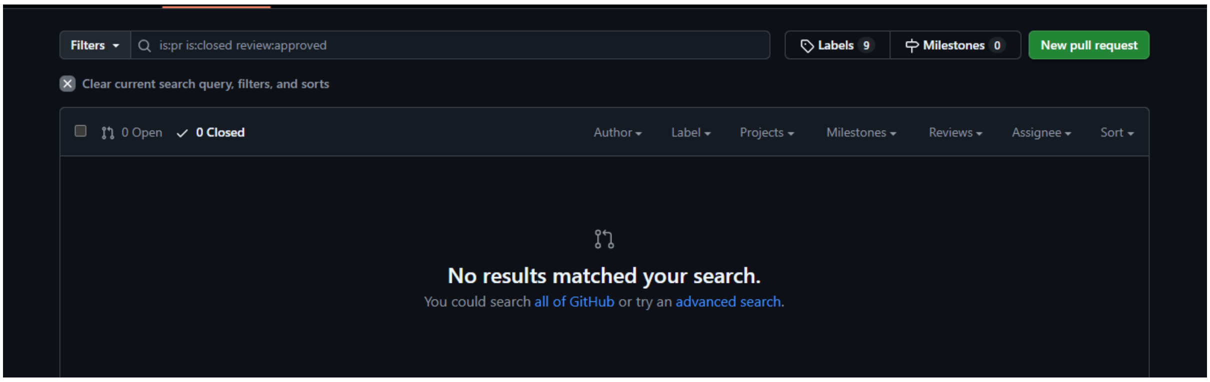
Clear current search query, filters, and sorts

0 Open ✓ 0 Closed

No results matched your search.

You could search all of GitHub or try an advanced search.

- Request changes and add comments → pr_request_changes.png



is:pr is:closed review:approved

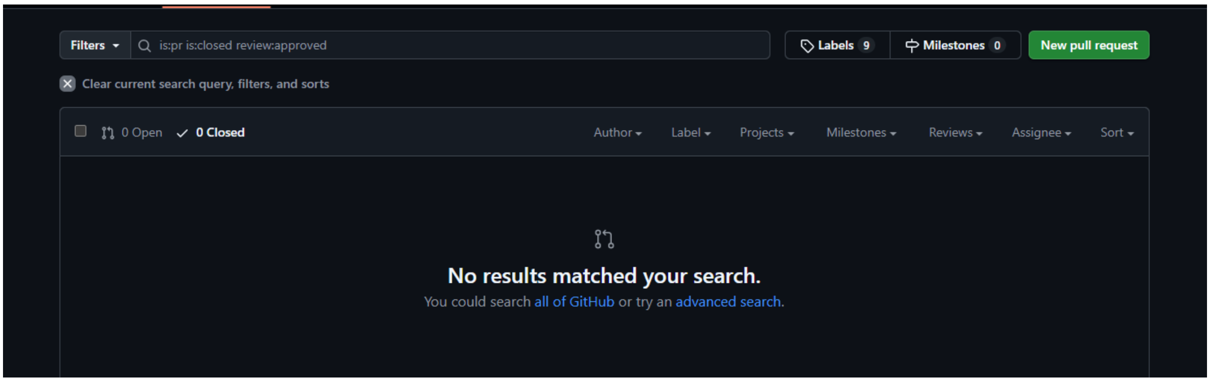
Clear current search query, filters, and sorts

0 Open ✓ 0 Closed

No results matched your search.

You could search all of GitHub or try an advanced search.

- Optionally reject/close the PR → pr_rejected.png



is:pr is:closed review:approved

Clear current search query, filters, and sorts

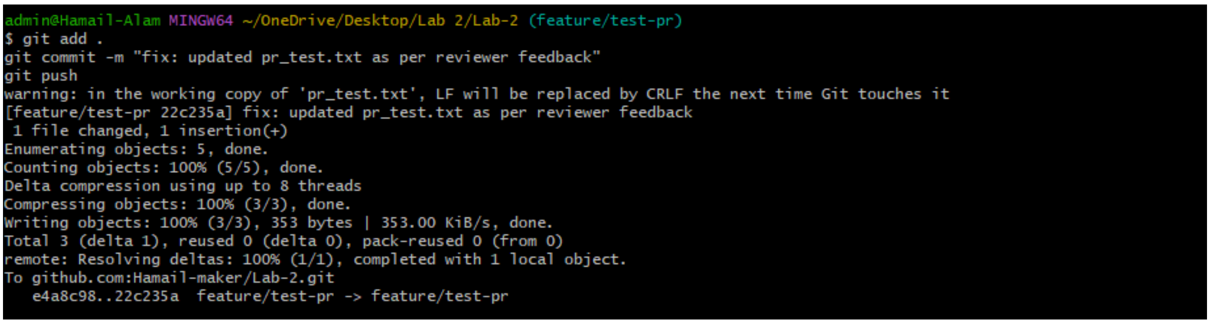
0 Open ✓ 0 Closed

No results matched your search.

You could search all of GitHub or try an advanced search.

4. **Author Actions:**

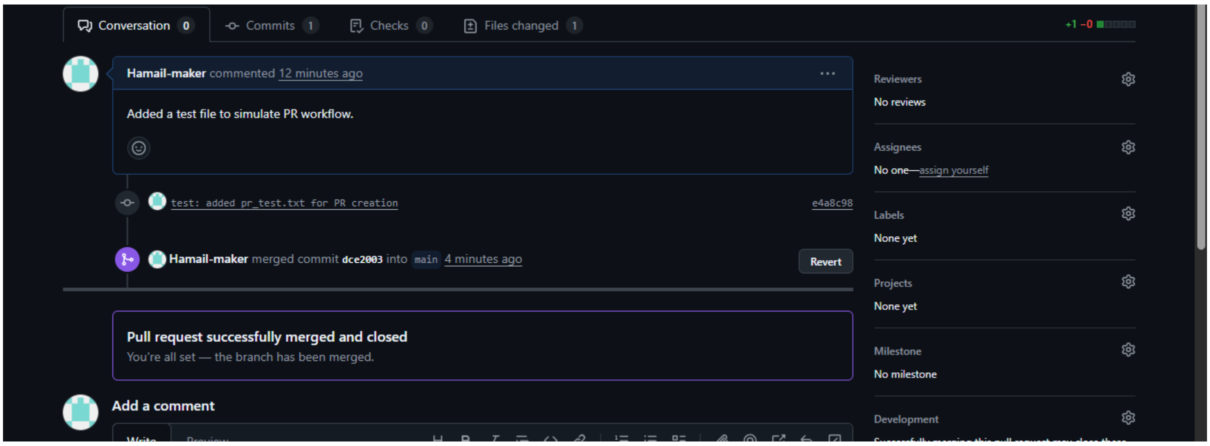
- Fix issues in the same branch and push new commits.
- PR updates automatically.
- Screenshot updated PR with new commits → pr_updated_with_commits.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/test-pr)
$ git add .
git commit -m "fix: updated pr_test.txt as per reviewer feedback"
git push
warning: in the working copy of 'pr_test.txt', LF will be replaced by CRLF the next time Git touches it
[feature/test-pr 22c235a] fix: updated pr_test.txt as per reviewer feedback
 1 file changed, 1 insertion(+)
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 353 bytes | 353.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Hamail-maker/Lab-2.git
   e4a8c98..22c235a  feature/test-pr -> feature/test-pr
```

5. **Merge the PR:**

- Reviewer approves → merge using GitHub/GitLab UI.
- Choose merge style (merge commit, squash, or rebase).
- Screenshot confirmation → pr_merge_confirm.png.
-



Hamail-maker commented 12 minutes ago

Added a test file to simulate PR workflow.

test: added pr_test.txt for PR creation e4a8c98

Hamail-maker merged commit dce2003 into main 4 minutes ago

Pull request successfully merged and closed

You're all set — the branch has been merged.

Add a comment

Reviewers: No reviews

Assignees: No one—assign yourself

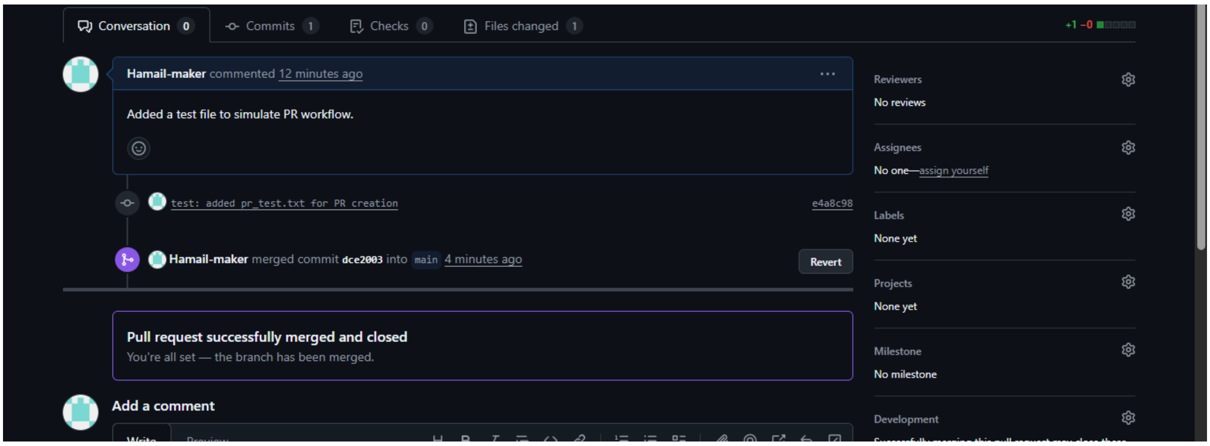
Labels: None yet

Projects: None yet

Milestone: No milestone

Development

- Screenshot result → pr_merged.png.
-



Hamail-maker commented 12 minutes ago

Added a test file to simulate PR workflow.

test: added pr_test.txt for PR creation e4a8c98

Hamail-maker merged commit dce2003 into main 4 minutes ago

Pull request successfully merged and closed

You're all set — the branch has been merged.

Add a comment

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

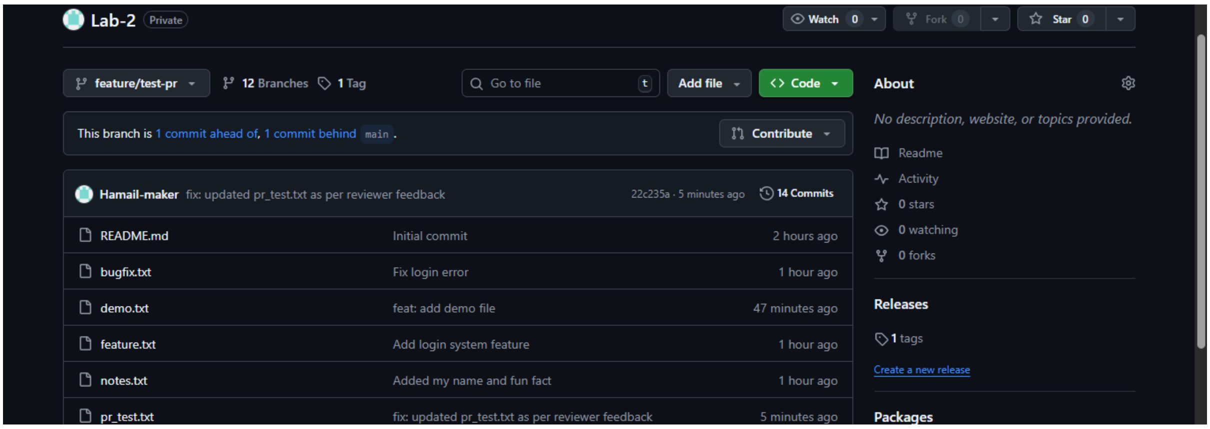
Projects: None yet

Milestone: No milestone

Development

6. **Branch Handling After Merge:**

- If deleted, screenshot → pr_branch_deleted.png.



This branch is 1 commit ahead of, 1 commit behind main .

Hamail-maker fix: updated pr_test.txt as per reviewer feedback 22c235a · 5 minutes ago 14 Commits

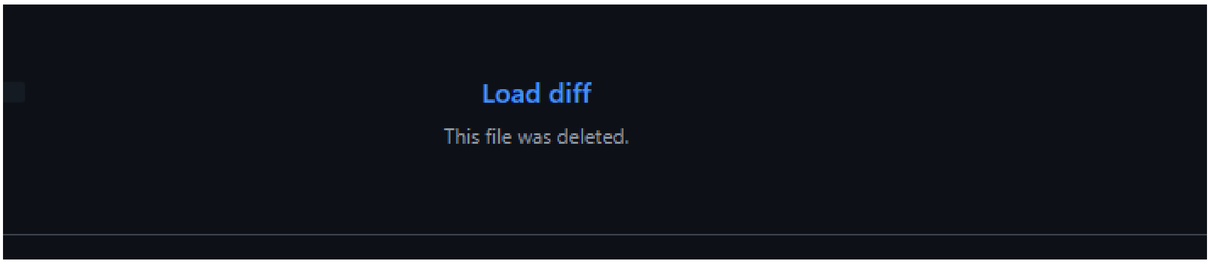
README.md	Initial commit	2 hours ago
bugfix.txt	Fix login error	1 hour ago
demo.txt	feat: add demo file	47 minutes ago
feature.txt	Add login system feature	1 hour ago
notes.txt	Added my name and fun fact	1 hour ago
pr_test.txt	fix: updated pr_test.txt as per reviewer feedback	5 minutes ago

- If kept, add explanation in REVIEW_NOTES.md.

Task 13: Branch Cleanup Best Practices

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

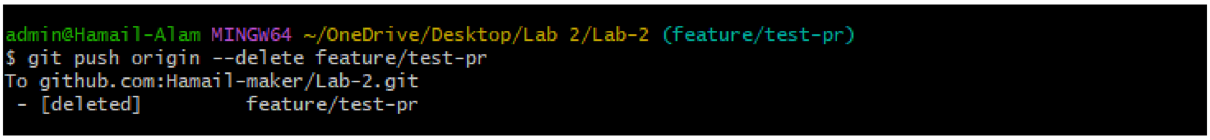
- Use GitHub UI → screenshot remote_branch_deleted.png.



Load diff

This file was deleted.

- Screenshot as remote_branch_delete_cmd.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/test-pr)
$ git push origin --delete feature/test-pr
To github.com:Hamail-maker/Lab-2.git
 - [deleted]         feature/test-pr
```

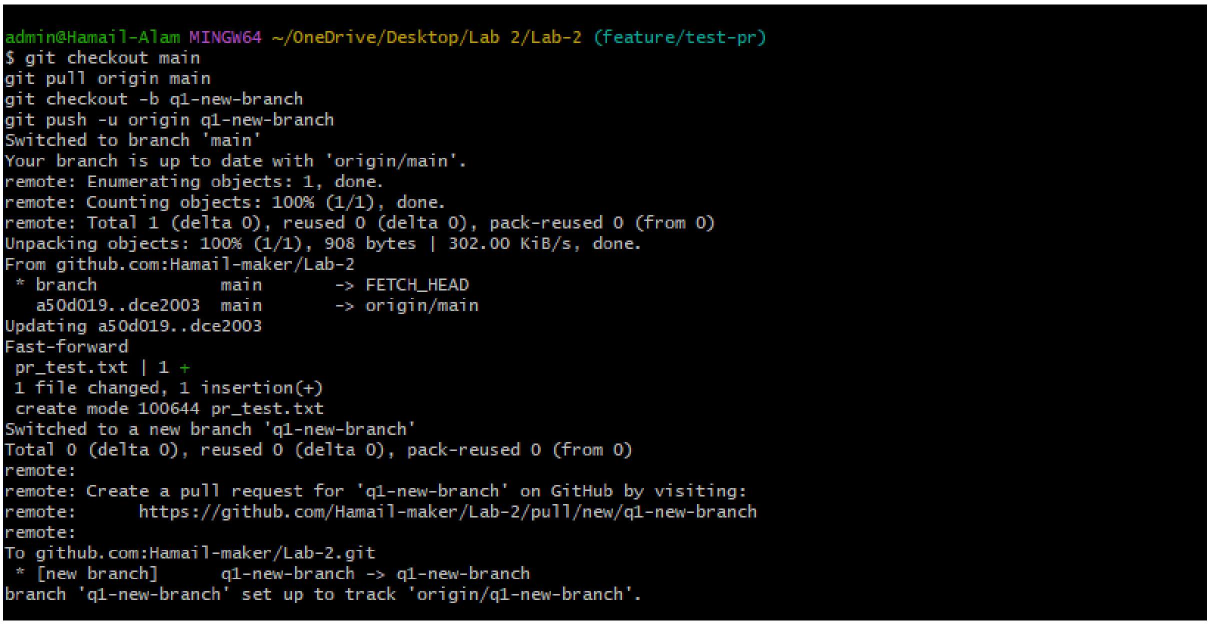
2. **Update Local Repository:**
3. git checkout main
4. git pull origin main
5. git branch -d <branch-name>
6. git branch

Exam Evaluation Questions

1. **Advanced Branching & Merge Verification**

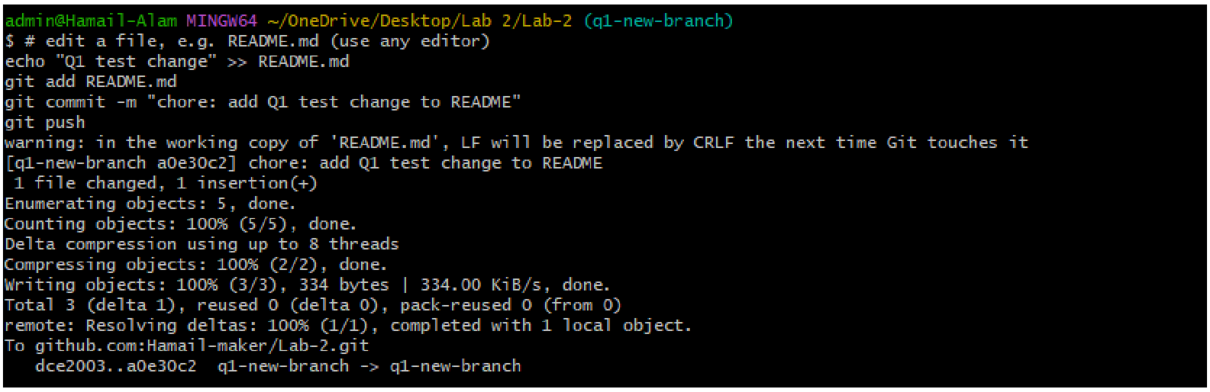
- Create a new branch in your repository.

- Screenshot as Q1_branch_created.png.



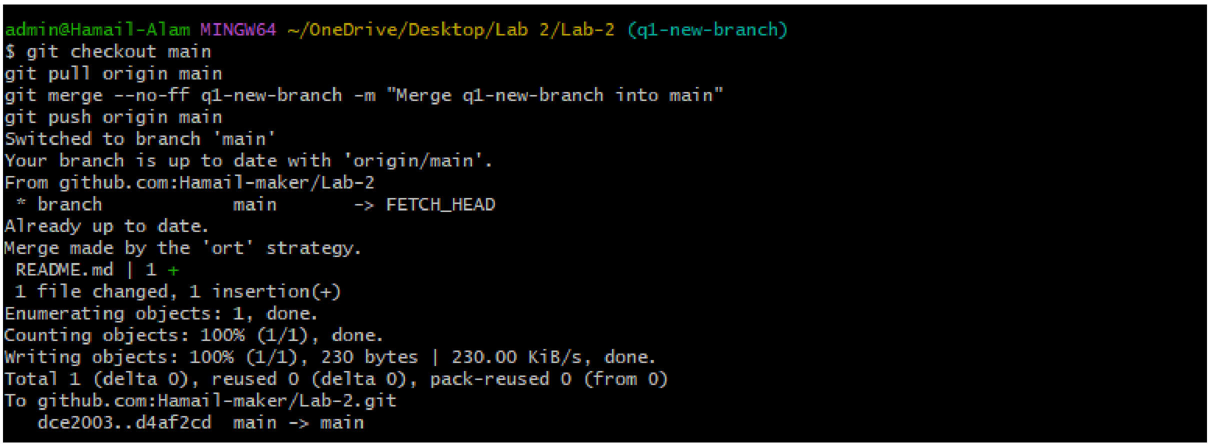
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/test-pr)
$ git checkout main
git pull origin main
git checkout -b q1-new-branch
git push -u origin q1-new-branch
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 908 bytes | 302.00 KiB/s, done.
From github.com:Hamail-maker/Lab-2
 * branch            main       -> FETCH_HEAD
   a50d019..dce2003  main       -> origin/main
Updating a50d019..dce2003
Fast-forward
 pr_test.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 pr_test.txt
Switched to a new branch 'q1-new-branch'
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'q1-new-branch' on GitHub by visiting:
remote:      https://github.com/Hamail-maker/Lab-2/pull/new/q1-new-branch
remote:
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      q1-new-branch -> q1-new-branch
branch 'q1-new-branch' set up to track 'origin/q1-new-branch'.
```

- Make a small change in a file and commit it.
  - Screenshot as Q1_commit_done.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (q1-new-branch)
$ # edit a file, e.g. README.md (use any editor)
echo "Q1 test change" >> README.md
git add README.md
git commit -m "chore: add Q1 test change to README"
git push
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[q1-new-branch a0e30c2] chore: add Q1 test change to README
 1 file changed, 1 insertion(+)
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 334 bytes | 334.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Hamail-maker/Lab-2.git
   dce2003..a0e30c2  q1-new-branch -> q1-new-branch
```

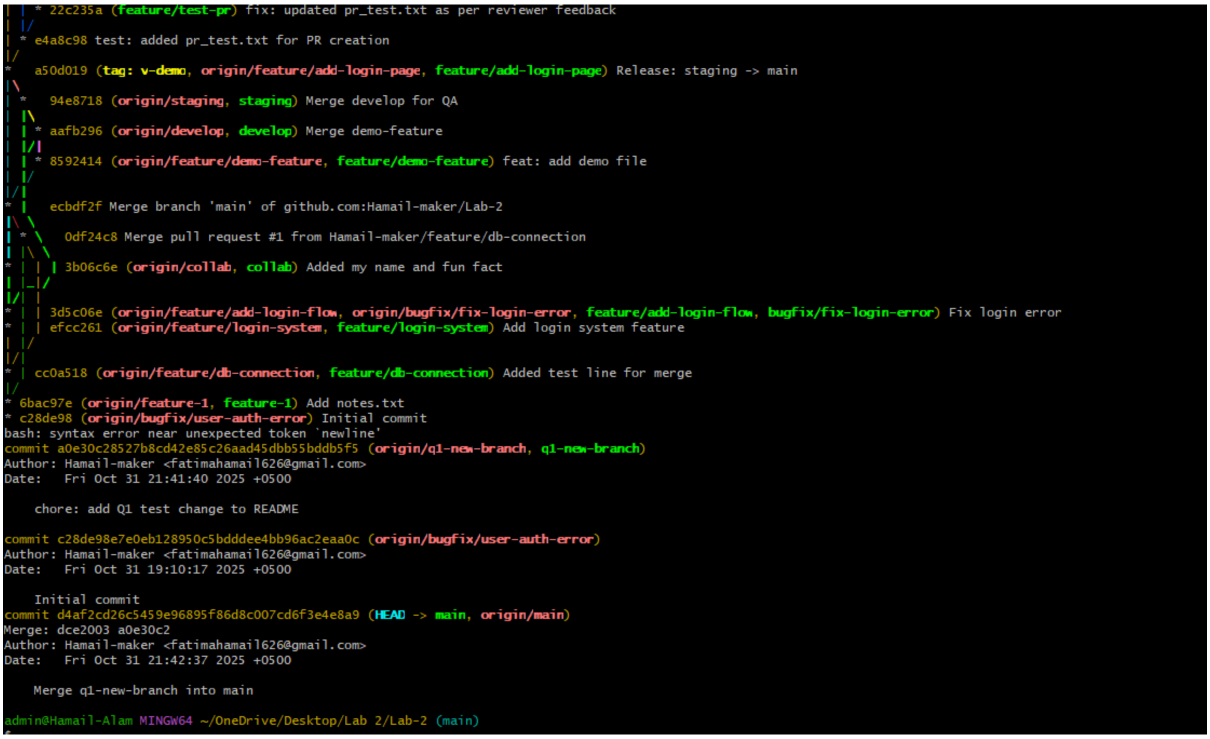
- Merge this branch back into the main branch.
  - Screenshot as Q1_merge_done.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (q1-new-branch)
$ git checkout main
git pull origin main
git merge --no-ff q1-new-branch -m "Merge q1-new-branch into main"
git push origin main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
From github.com:Hamail-maker/Lab-2
 * branch            main       -> FETCH_HEAD
Already up to date.
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 230 bytes | 230.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   dce2003..d4af2cd  main -> main
```

- Show the history of commits in a way that verifies the merge.

- Screenshot as Q1_merge_history.png.

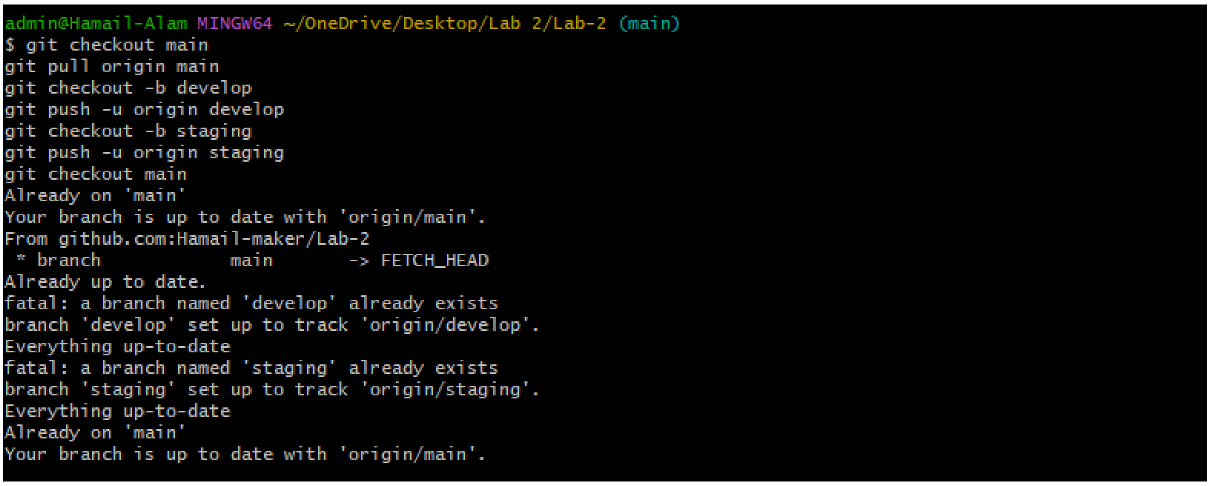


```
| | * 22c235a (feature/test-pr) fix: updated pr_test.txt as per reviewer feedback
| |/
| * e4a8c98 test: added pr_test.txt for PR creation
|/
*   a50d019 (tag: v-demo, origin/feature/add-login-page, feature/add-login-page) Release: staging -> main
|\
| *   94e8718 (origin/staging, staging) Merge develop for QA
| |\
| | * aafb296 (origin/develop, develop) Merge demo-feature
| |/|
| | * 8592414 (origin/feature/demo-feature, feature/demo-feature) feat: add demo file
| |/
|/|
* |   ecbdf2f Merge branch 'main' of github.com:Hamail-maker/Lab-2
|\ \
| * \   0df24c8 Merge pull request #1 from Hamail-maker/feature/db-connection
| |\ \
* | | | 3b06c6e (origin/collab, collab) Added my name and fun fact
| |_|/
|/| |
* | | 3d5c06e (origin/feature/add-login-flow, origin/bugfix/fix-login-error, feature/add-login-flow, bugfix/fix-login-error) Fix login error
* | | efcc261 (origin/feature/login-system, feature/login-system) Add login system feature
| |/
|/|
* | cc0a518 (origin/feature/db-connection, feature/db-connection) Added test line for merge
|/
* 6bac97e (origin/feature-1, feature-1) Add notes.txt
* c28de98 (origin/bugfix/user-auth-error) Initial commit
bash: syntax error near unexpected token `newline'
commit a0e30c28527b8cd42e85c26aad45dbb55bddb5f5 (origin/q1-new-branch, q1-new-branch)
Author: Hamail-maker <fatimahamail626@gmail.com>
Date:   Fri Oct 31 21:41:40 2025 +0500

    chore: add Q1 test change to README

commit c28de98e7e0eb128950c5bdddee4bb96ac2eaa0c (origin/bugfix/user-auth-error)
Author: Hamail-maker <fatimahamail626@gmail.com>
Date:   Fri Oct 31 19:10:17 2025 +0500

    Initial commit
commit d4af2cd26c5459e96895f86d8c007cd6f3e4e8a9 (HEAD -> main, origin/main)
Merge: dce2003 a0e30c2
Author: Hamail-maker <fatimahamail626@gmail.com>
Date:   Fri Oct 31 21:42:37 2025 +0500

    Merge q1-new-branch into main

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
```

2. **Multi-Stage Workflow Simulation**

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

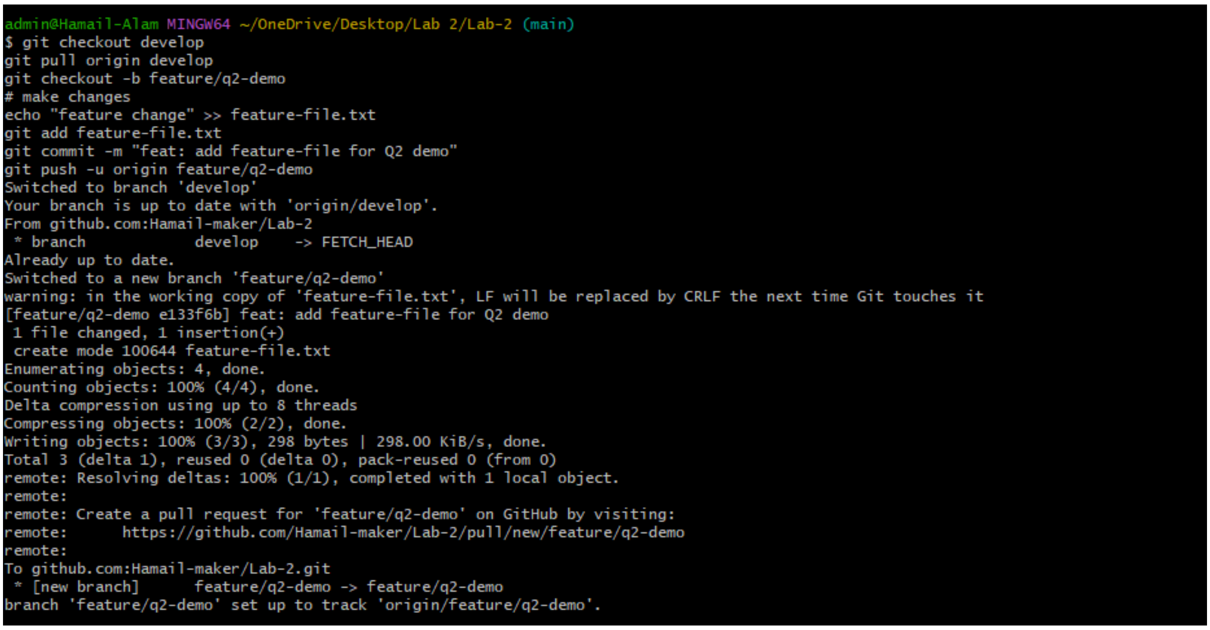
- Screenshot as Q2_branches_created.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git checkout main
git pull origin main
git checkout -b develop
git push -u origin develop
git checkout -b staging
git push -u origin staging
git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
From github.com:Hamail-maker/Lab-2
 * branch            main       -> FETCH_HEAD
Already up to date.
fatal: a branch named 'develop' already exists
branch 'develop' set up to track 'origin/develop'.
Everything up-to-date
fatal: a branch named 'staging' already exists
branch 'staging' set up to track 'origin/staging'.
Everything up-to-date
Already on 'main'
Your branch is up to date with 'origin/main'.
```

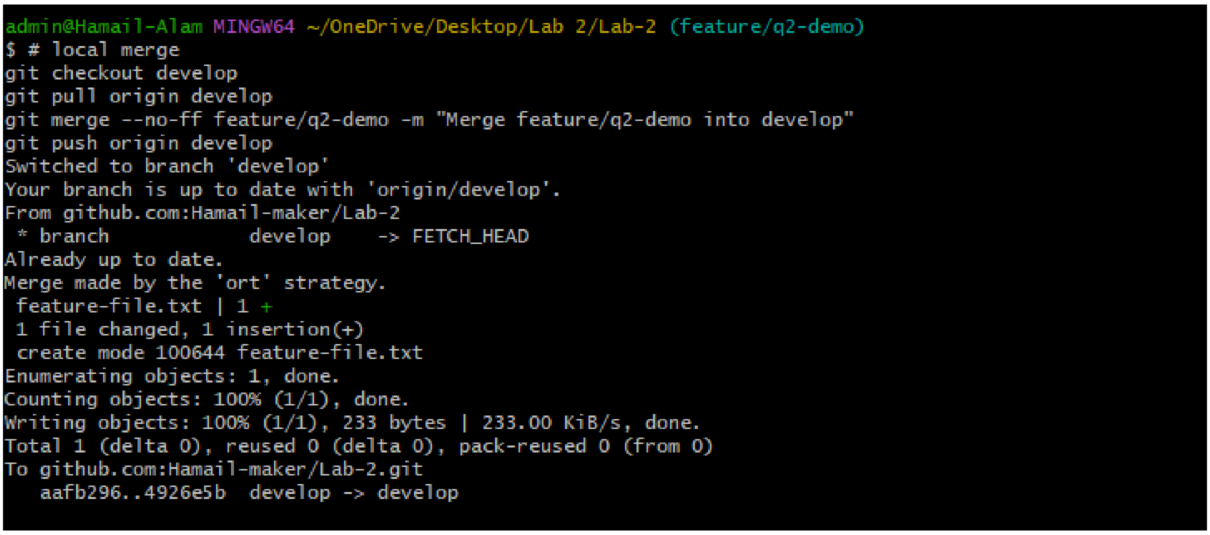
○ Create a feature branch from develop, make changes, and commit them.

- Screenshot as Q2_feature_branch_commit.png.



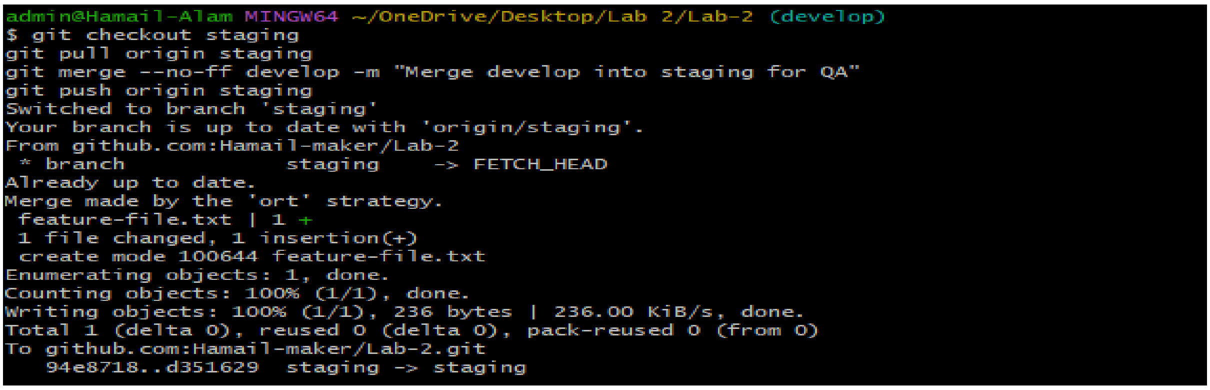
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ git checkout develop
git pull origin develop
git checkout -b feature/q2-demo
# make changes
echo "feature change" >> feature-file.txt
git add feature-file.txt
git commit -m "feat: add feature-file for Q2 demo"
git push -u origin feature/q2-demo
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
From github.com:Hamail-maker/Lab-2
 * branch            develop    -> FETCH_HEAD
Already up to date.
Switched to a new branch 'feature/q2-demo'
warning: in the working copy of 'feature-file.txt', LF will be replaced by CRLF the next time Git touches it
[feature/q2-demo e133f6b] feat: add feature-file for Q2 demo
 1 file changed, 1 insertion(+)
 create mode 100644 feature-file.txt
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/q2-demo' on GitHub by visiting:
remote:      https://github.com/Hamail-maker/Lab-2/pull/new/feature/q2-demo
remote:
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      feature/q2-demo -> feature/q2-demo
branch 'feature/q2-demo' set up to track 'origin/feature/q2-demo'.
```

- Merge the feature branch into develop.
  - Screenshot as Q2_merge_into_develop.png.



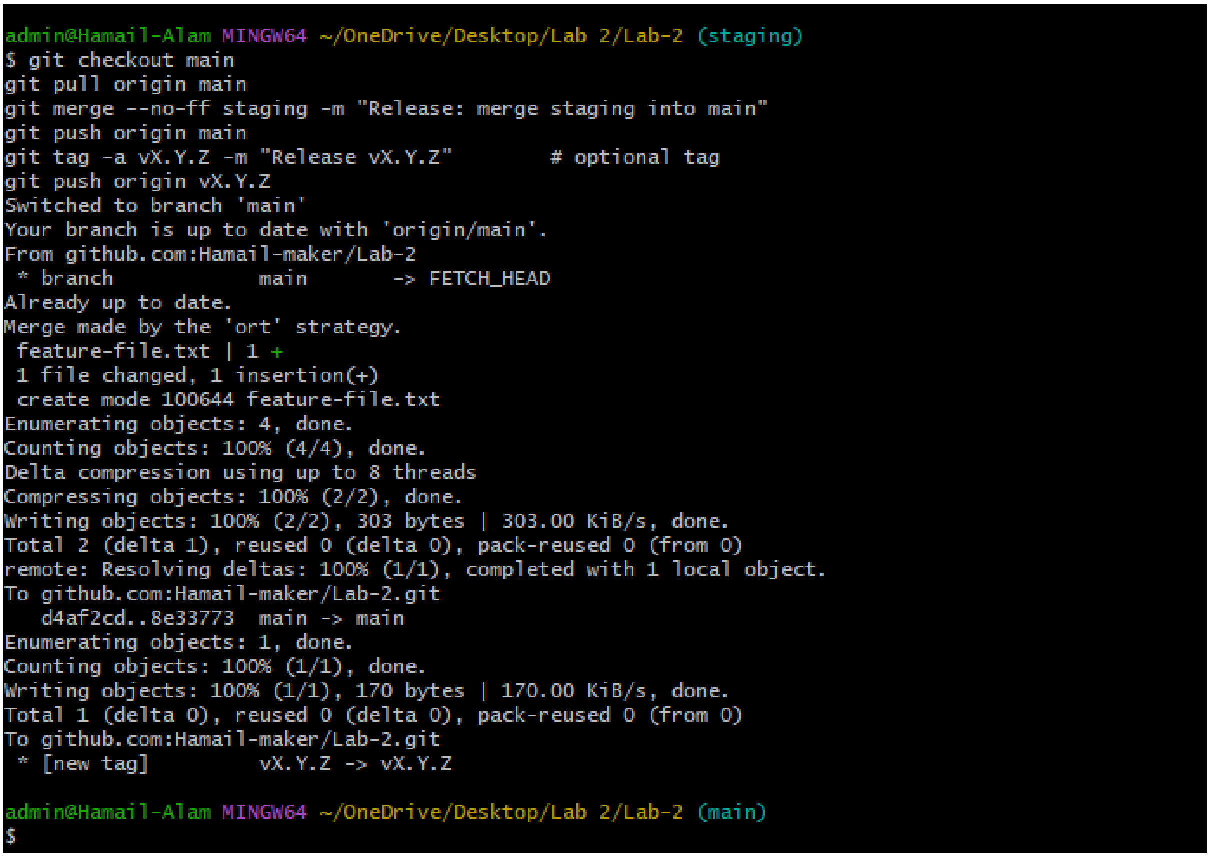
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (feature/q2-demo)
$ # local merge
git checkout develop
git pull origin develop
git merge --no-ff feature/q2-demo -m "Merge feature/q2-demo into develop"
git push origin develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
From github.com:Hamail-maker/Lab-2
 * branch            develop    -> FETCH_HEAD
Already up to date.
Merge made by the 'ort' strategy.
 feature-file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature-file.txt
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 233 bytes | 233.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   aafb296..4926e5b  develop -> develop
```

- Merge develop into staging.
  - Screenshot as Q2_merge_into_staging.png.



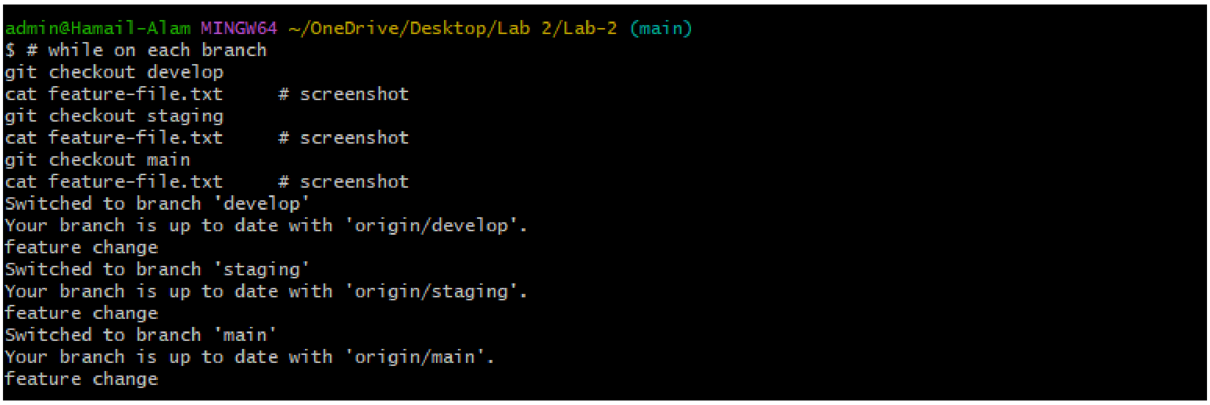
```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (develop)
$ git checkout staging
git pull origin staging
git merge --no-ff develop -m "Merge develop into staging for QA"
git push origin staging
Switched to branch 'staging'
Your branch is up to date with 'origin/staging'.
From github.com:Hamail-maker/Lab-2
 * branch            staging    -> FETCH_HEAD
Already up to date.
Merge made by the 'ort' strategy.
 feature-file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature-file.txt
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 236 bytes | 236.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
   94e8718..d351629  staging -> staging
```

- Merge staging into main.
  - Screenshot as Q2_merge_into_main.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (staging)
$ git checkout main
git pull origin main
git merge --no-ff staging -m "Release: merge staging into main"
git push origin main
git tag -a vX.Y.Z -m "Release vX.Y.Z"        # optional tag
git push origin vX.Y.Z
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
From github.com:Hamail-maker/Lab-2
 * branch            main       -> FETCH_HEAD
Already up to date.
Merge made by the 'ort' strategy.
 feature-file.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature-file.txt
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 303 bytes | 303.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Hamail-maker/Lab-2.git
   d4af2cd..8e33773  main -> main
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 170 bytes | 170.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Hamail-maker/Lab-2.git
 * [new tag]         vX.Y.Z -> vX.Y.Z

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$
```

- Provide proof that each stage contains the updated changes before it reaches main.
  - Screenshot as Q2_stage_verification.png.

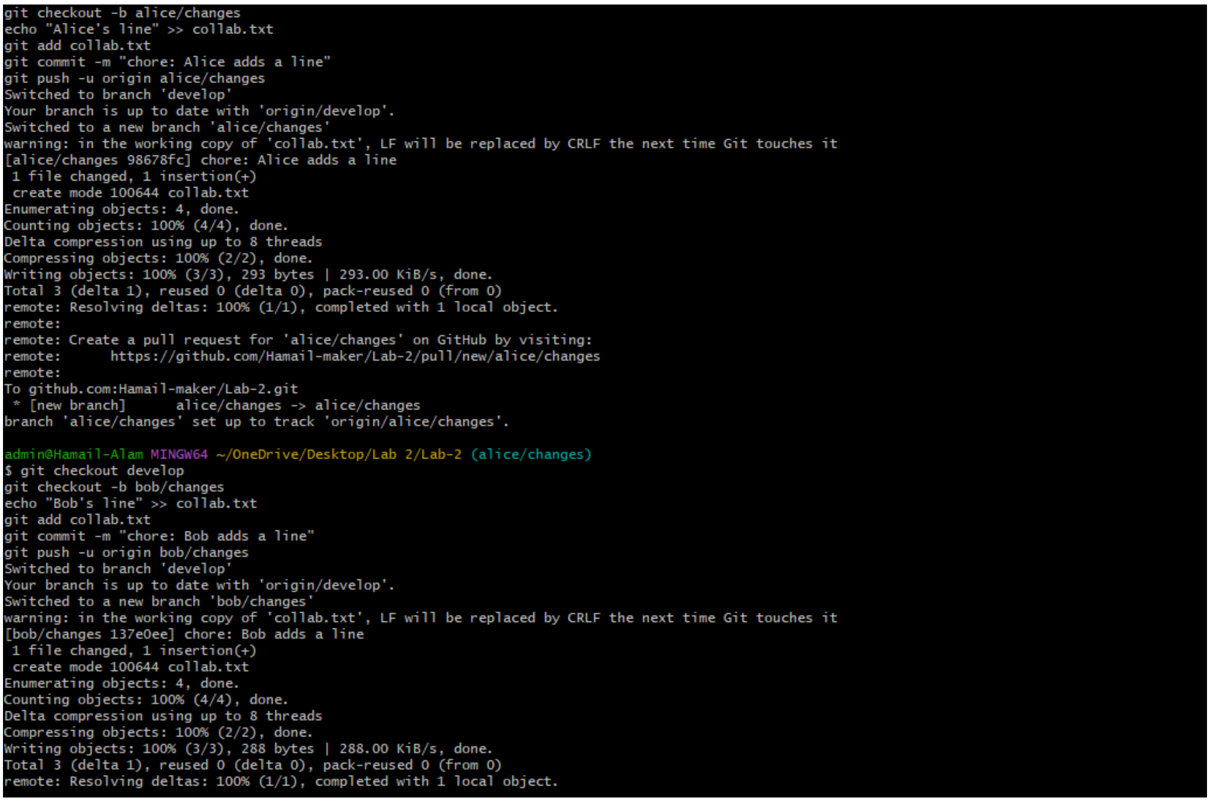


```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (main)
$ # while on each branch
git checkout develop
cat feature-file.txt     # screenshot
git checkout staging
cat feature-file.txt     # screenshot
git checkout main
cat feature-file.txt     # screenshot
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
feature change
Switched to branch 'staging'
Your branch is up to date with 'origin/staging'.
feature change
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
feature change
```

---

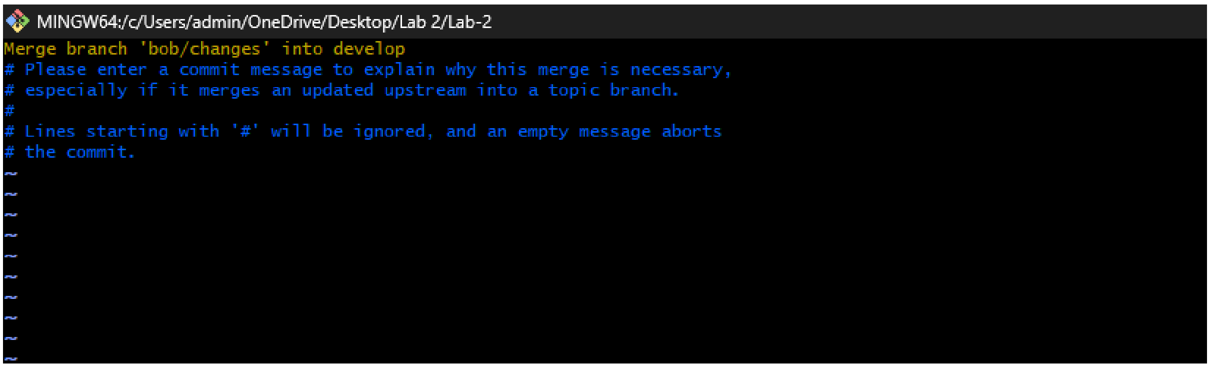
3. **Collaboration & Conflict Resolution**

- Work with a collaborator: both contributors should modify the same file but in separate branches.
  - Screenshot as Q3_branches_modified.png.



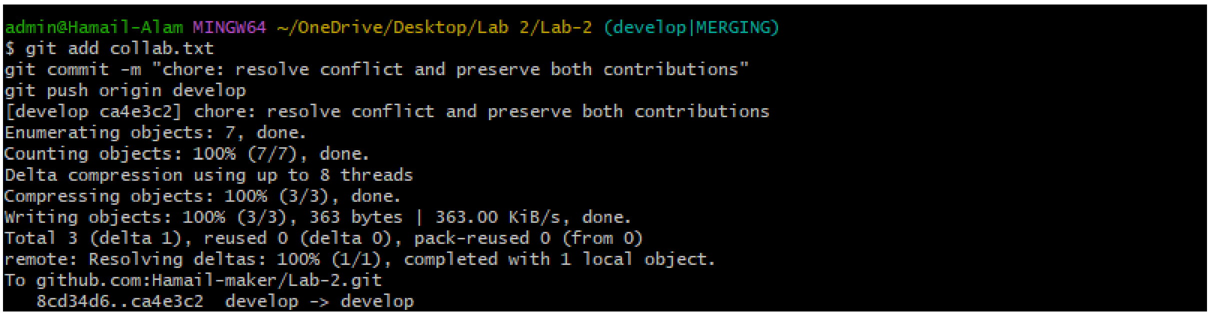
```
git checkout -b alice/changes
echo "Alice's line" >> collab.txt
git add collab.txt
git commit -m "chore: Alice adds a line"
git push -u origin alice/changes
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Switched to a new branch 'alice/changes'
warning: in the working copy of 'collab.txt', LF will be replaced by CRLF the next time Git touches it
[alice/changes 98678fc] chore: Alice adds a line
 1 file changed, 1 insertion(+)
 create mode 100644 collab.txt
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 293 bytes | 293.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'alice/changes' on GitHub by visiting:
remote:      https://github.com/Hamail-maker/Lab-2/pull/new/alice/changes
remote:
To github.com:Hamail-maker/Lab-2.git
 * [new branch]      alice/changes -> alice/changes
branch 'alice/changes' set up to track 'origin/alice/changes'.

admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (alice/changes)
$ git checkout develop
git checkout -b bob/changes
echo "Bob's line" >> collab.txt
git add collab.txt
git commit -m "chore: Bob adds a line"
git push -u origin bob/changes
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Switched to a new branch 'bob/changes'
warning: in the working copy of 'collab.txt', LF will be replaced by CRLF the next time Git touches it
[bob/changes 137e0ee] chore: Bob adds a line
 1 file changed, 1 insertion(+)
 create mode 100644 collab.txt
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 288 bytes | 288.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
```

- ○ Attempt to merge the branches and capture the conflict.
  - ▪ Screenshot as Q3_merge_conflict.png.



```
Merge branch 'bob/changes' into develop
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
~
~
~
~
~
```

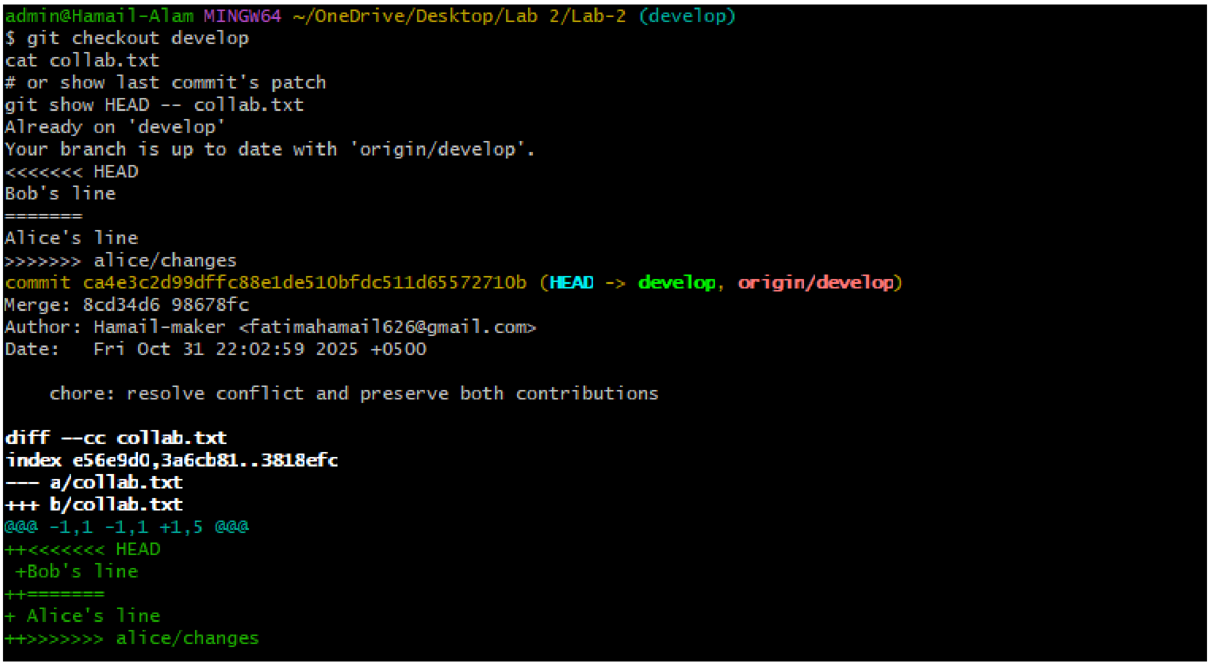
- ○ Resolve the conflict so that both contributions are preserved.
  - ▪ Screenshot as Q3_conflict_resolved.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (develop|MERGING)
$ git add collab.txt
git commit -m "chore: resolve conflict and preserve both contributions"
git push origin develop
[develop ca4e3c2] chore: resolve conflict and preserve both contributions
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Hamail-maker/Lab-2.git
   8cd34d6..ca4e3c2  develop -> develop
```

- ○ Provide evidence that the final version of the file contains both collaborators' changes.

- Screenshot as Q3_final_file.png.



```
admin@Hamail-Alam MINGW64 ~/OneDrive/Desktop/Lab 2/Lab-2 (develop)
$ git checkout develop
cat collab.txt
# or show last commit's patch
git show HEAD -- collab.txt
Already on 'develop'
Your branch is up to date with 'origin/develop'.
<<<<<<< HEAD
Bob's line
=======
Alice's line
>>>>>>> alice/changes
commit ca4e3c2d99dffc88e1de510bfdc511d65572710b (HEAD -> develop, origin/develop)
Merge: 8cd34d6 98678fc
Author: Hamail-maker <fatimahamail626@gmail.com>
Date:   Fri Oct 31 22:02:59 2025 +0500

    chore: resolve conflict and preserve both contributions

diff --cc collab.txt
index e56e9d0,3a6cb81..3818efc
--- a/collab.txt
+++ b/collab.txt
@@@ -1,1 -1,1 +1,5 @@@
++<<<<<<< HEAD
 +Bob's line
++=======
+ Alice's line
++>>>>>>> alice/changes
```