

**Name : Reena Qureshi**

**Sem : V-B**

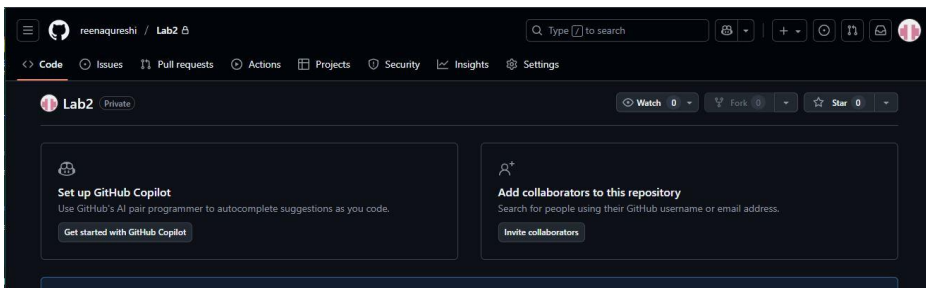
**Reg No : 2033-BSE-052**

## Lab 2

### Task 1: Create Private GitHub Repository

Create a new private repository named Lab2 on GitHub.

Take a screenshot of your repo settings showing it's private.



### Task 2: Connect Repository via SSH

Generate a new SSH key using PowerShell:

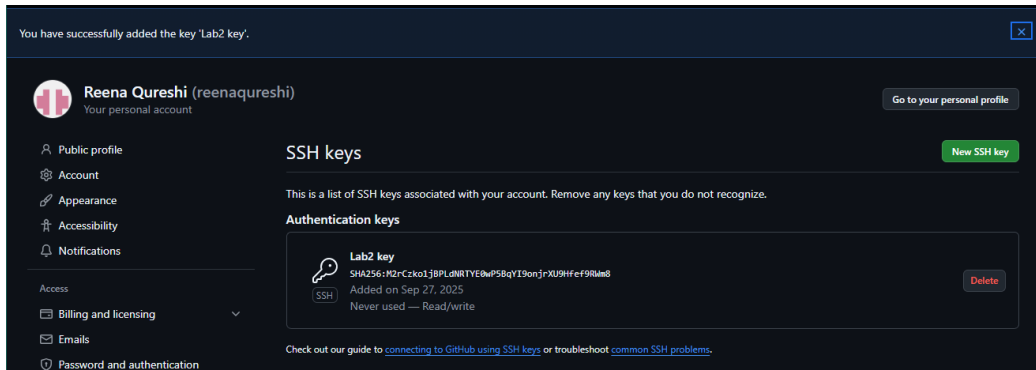
`ssh-keygen -t ed25519 -C "your_email@example.com"`

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ ssh-keygen -t ed25519 -C "23-22411-052@se.fjwu.edu.pk"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Reena Qureshi/.ssh/id_ed25519):
Created directory '/c/Users/Reena Qureshi/.ssh'.
Enter passphrase for "/c/Users/Reena Qureshi/.ssh/id_ed25519" (empty for no pass
phrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Reena Qureshi/.ssh/id_ed25519
Your public key has been saved in /c/Users/Reena Qureshi/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:M2rCzko1jBPLdNRTYE0wP5BqYI9onjrXU9Hfef9RWm8 23-22411-052@se.fjwu.edu.pk
The key's randomart image is:
+--[ED25519 256]--+
|  ..*Bo          |
|  o. .++         |
| oo+.. oo        |
| .+. *+ . ..     |
| o *+ .S. . . o  |
| o + ... o. o .oo |
| . ..o.o        ..oE|
| o..ooo         .o |
| o..o.          o  |
+-----[SHA256]-----+

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDU0NN8XgxxwFA6Aneh4Fp6goJbdmn5wbW1aR0vLQTaN 23-22411-052@se.fjwu.edu.pk

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ |
```

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



## CLONE YOUR LAB2 REPO USING SSH.

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

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ git clone git@github.com:reenaqureshi/Lab2.git
Cloning into 'Lab2'...
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
warning: You appear to have cloned an empty repository.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ |
```

## Task 3: Configure Git Username and Email

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
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ git config --global user.name "Reena Qureshi"
git config --global user.email "23-22411-052@se.fjwu.edu.pk"

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ |
```

## Verify your configuration:

```
git config --list
```

```

Reena Qureshi@DESKTOP-TC5MOFP 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
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
user.name=Reena Qureshi
user.email=23-22411-052@se.fjwu.edu.pk
gui.recentrepo=C:/Users/Reena Qureshi/Lab2

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~
$ |

```

## Task 4: Explore the .git Folder

Navigate into your cloned repository folder.

Show hidden files and locate the .git directory.

Explore what's inside using:

**ls -a .git**

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (master)
$ ls -a .git
./ ../ COMMIT_EDITMSG FETCH_HEAD HEAD ORIG_HEAD config description hooks/ index info/ logs/ objects/ refs/

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (master)
$ |

```

## Task 5: Local Repository Management

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

**rm -rf .git**

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~  
$ rm -rf .git  
  
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~  
$ |
```

Re-initialize the local git repository

**git init**

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~  
$ git init  
Initialized empty Git repository in C:/Users/Reena Qureshi/.git/  
  
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)  
$ |
```

Add a file named README.md and commit it:

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

**git add README.md**

**git commit -m "Initial commit"**

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)  
$ ^[[200~echo "# Lab2 Git Practice" > README.md  
bash: $'\E[200~echo': command not found  
  
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)  
$ git add README.md  
  
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)  
$ git commit -m "Initial commit"  
[master (root-commit) b431fe2] Initial commit  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 README.md  
  
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)  
$ |
```

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

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

**git push -u origin main**

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)
$ git remote add origin git@github.com:reenaqureshi/Lab2.git
git push -u origin master

Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 77.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:reenaqureshi/Lab2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)
$

```

## Task 6: File Status & Staging

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

Check status

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)
$ echo "This is my first note." > notes.txt

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)
$

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)
$ git status
warning: could not open directory 'Application Data/': Permission denied
warning: could not open directory 'Cookies/': Permission denied
warning: could not open directory 'Local Settings/': Permission denied
warning: could not open directory 'My Documents/': Permission denied
warning: could not open directory 'NetHood/': Permission denied
warning: could not open directory 'PrintHood/': Permission denied
warning: could not open directory 'Recent/': Permission denied
warning: could not open directory 'SendTo/': Permission denied
warning: could not open directory 'Start Menu/': Permission denied
warning: could not open directory 'Templates/': Permission denied
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .android/
  .dbus-keyrings/
  .gitconfig
  .idea/
  .nuget/
  .openshot_qt/
  .packetracer
  .ssh/
  .templateengine/
  3D Objects/
  AppData/
  Contacts/
  Downloads/
  Favorites/
  IntelGraphicsProfiles/
  Lab2/
  Music/
  NTUSER.DAT
  NTUSER.DAT{09159599-738b-11ee-b4dd-7c5cf82c0aa6}.TM.b1f
  NTUSER.DAT{09159599-738b-11ee-b4dd-7c5cf82c0aa6}.TMContainer00000000000000000001.regtrans-ms
  NTUSER.DAT{09159599-738b-11ee-b4dd-7c5cf82c0aa6}.TMContainer00000000000000000002.regtrans-ms

```

Stage and commit:

git add notes.txt

git commit -m "Add notes.txt"

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~ (master)
$ 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
[master 8621ce5] Add notes.txt
 1 file changed, 1 insertion(+)
 create mode 100644 notes.txt

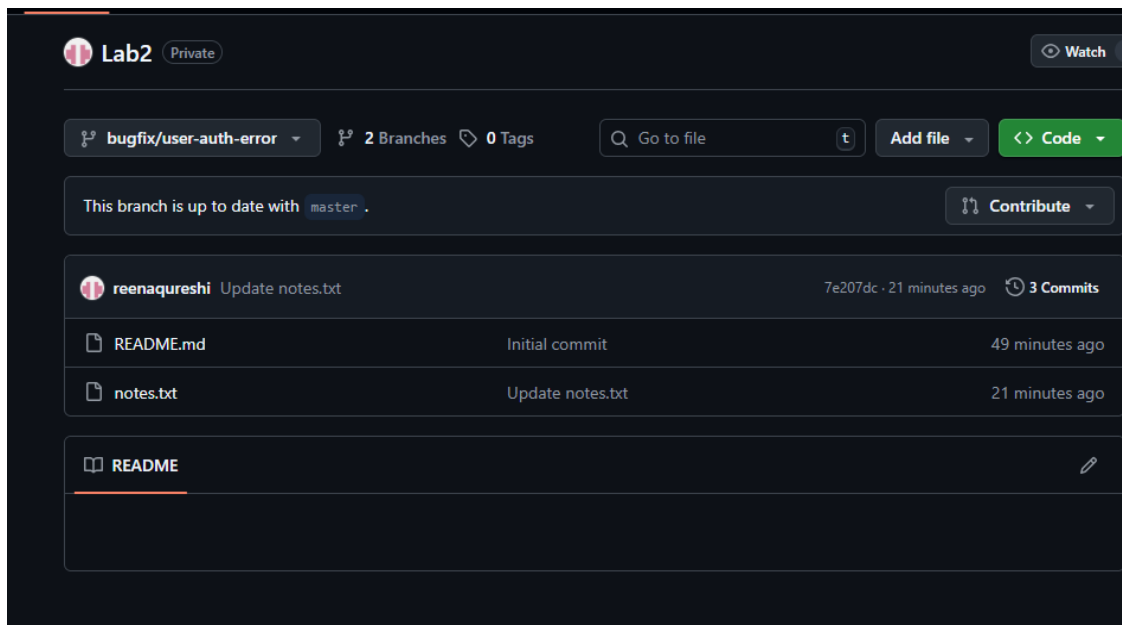
```

## Task 7: Branch Creation Using GitHub GUI

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

Pull the branch to your local repository to sync.

```
git pull origin bugfix/user-auth-error
```



```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main)
$ git fetch origin
git checkout bugfix/user-auth-error
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 8 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (9/9), 762 bytes | 8.00 KiB/s, done.
From github.com:reenaquareshi/Lab2
 * [new branch]      bugfix/user-auth-error -> origin/bugfix/user-auth-error
 * [new branch]      master -> origin/master
branch 'bugfix/user-auth-error' set up to track 'origin/bugfix/user-auth-error'.
Switched to a new branch 'bugfix/user-auth-error'

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (bugfix/user-auth-error)
$

```

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

Create a branch named `feature/db-connection` using Git Bash:

**git checkout -b feature/db-connection**

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

**git push origin feature/db-connection**

```
$
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (bugfix/user-auth-error)
$ git checkout -b feature/db-connection
Switched to a new branch 'feature/db-connection'

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature/db-connection)
$ git push origin feature/db-connection
Total 0 (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/reenaqureshi/Lab2/pull/new/feature/db-connection
remote:
To github.com:reenaqureshi/Lab2.git
 * [new branch]      feature/db-connection -> feature/db-connection

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature/db-connection)
$
```

## Task 9: Branching & Merging

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

**git checkout -b feature-1**

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature/db-connection)
$ git checkout -b feature-1
Switched to a new branch 'feature-1'
```

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

**git add main.py**

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

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature-1)
$ echo "def hello():" > main.py
echo "    print('Hello from feature-1 branch')\" >> main.py

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature-1)
$

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature-1)
$ git add main.py
git commit -m "Add new function to main.py"
warning: in the working copy of 'main.py', LF will be replaced by CRLF the next time Git touches it
[feature-1 2088b9c] Add new function to main.py
 1 file changed, 2 insertions(+)
 create mode 100644 main.py

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature-1)
$ |
```

Switch back to main and merge:

`git checkout main`

`git merge feature-1`

```
Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/Lab2 (feature-1)
$ git checkout master
git merge feature-1
branch 'master' set up to track 'origin/master'.
Switched to a new branch 'master'
Updating 7e207dc..2088b9c
Fast-forward
 main.py | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 main.py
```

Push all branches:

`git push origin main`

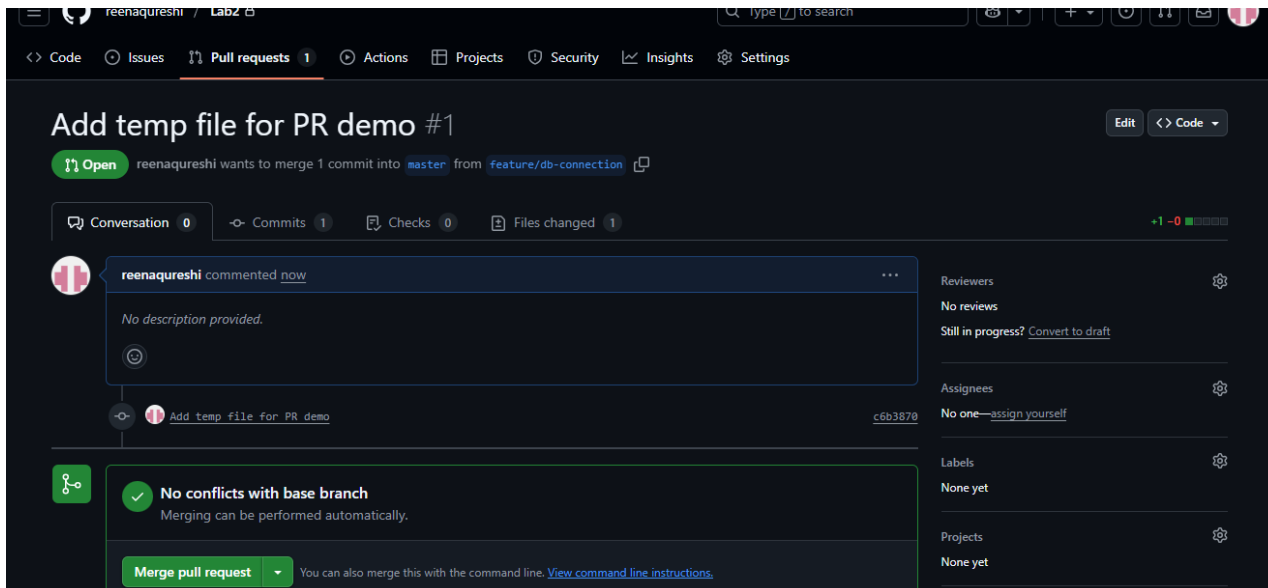
`git push origin feature-1`

```
Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/Lab2 (master)
$ git push origin master
git push origin feature-1
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 376 bytes | 376.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:reenaquareshi/Lab2.git
   7e207dc..2088b9c  master -> master
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/reenaquareshi/Lab2/pull/new/feature-1
remote:
To github.com:reenaquareshi/Lab2.git
 * [new branch]      feature-1 -> feature-1
```

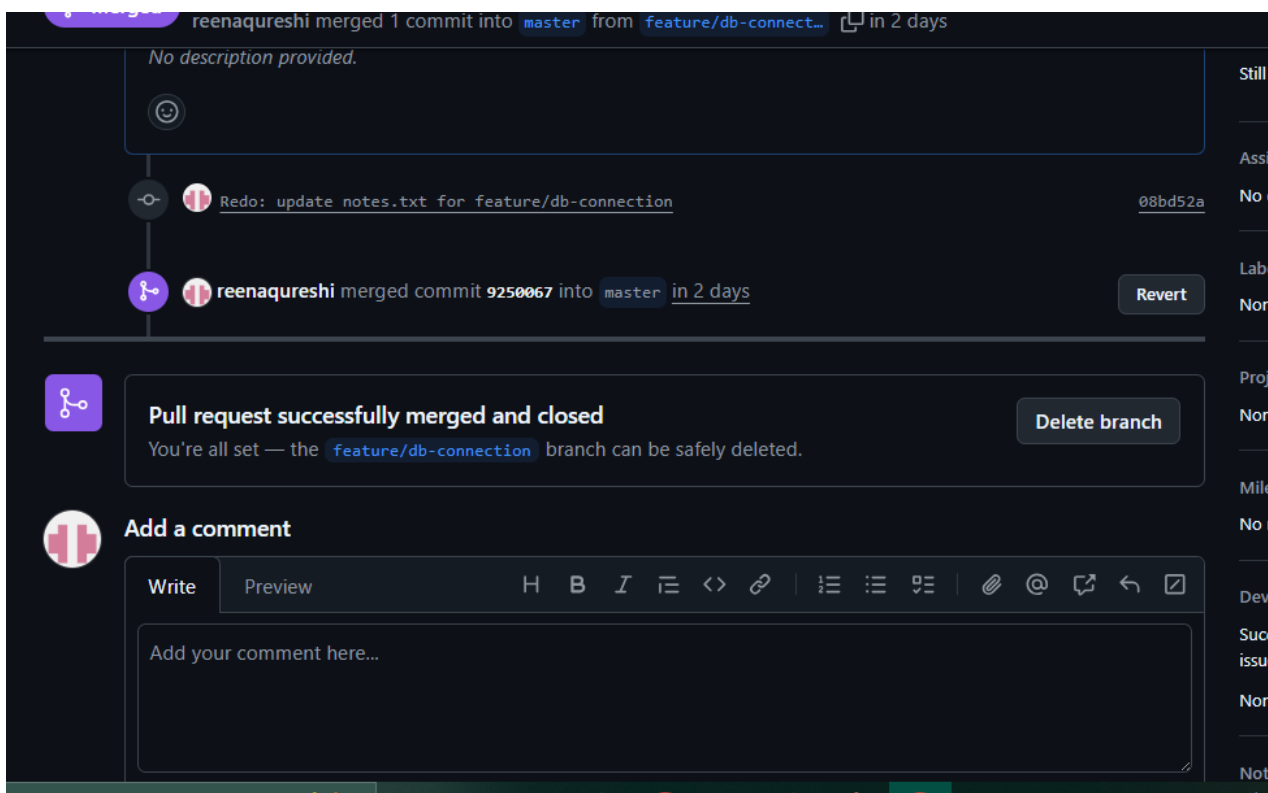
## Task 10: Pull Request and Branch Review (GitHub GUI)

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

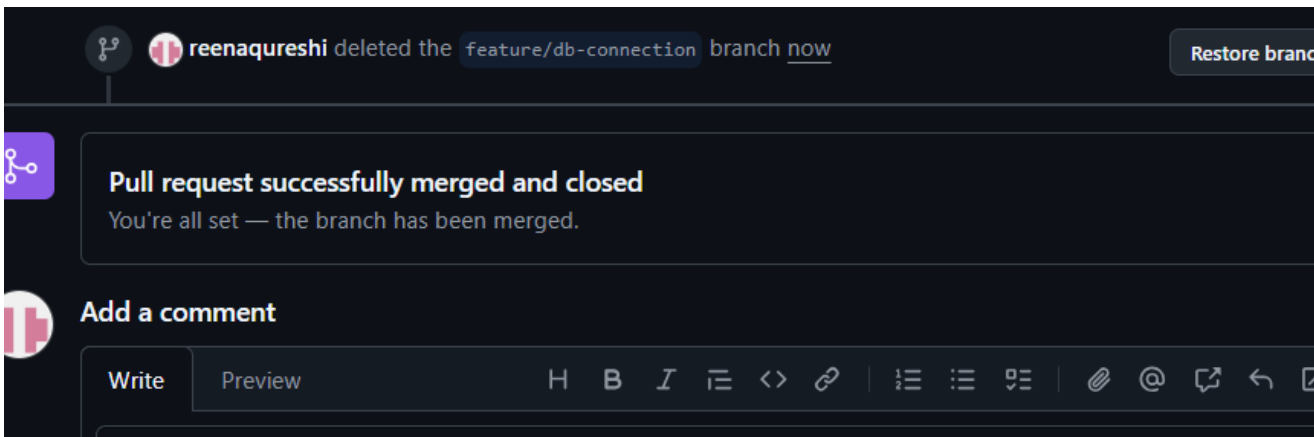




**Review the Pull Request and merge it using the GitHub GUI.**



**AFTER MERGING, DELETE THE FEATURE/DB-CONNECTION BRANCH USING THE GITHUB GUI.**



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

Create the following branches to simulate a professional branching strategy:

develop

staging

feature/\*

bugfix/\*

```
Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/Lab2 (master)
$ git checkout -b develop
git push origin develop
Switched to a new branch 'develop'
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 355 bytes | 88.00 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 'develop' on GitHub by visiting:
remote:   https://github.com/reenaqureshi/Lab2/pull/new/develop
remote:
To github.com:reenaqureshi/Lab2.git
 * [new branch]      develop -> develop
```

```
Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/Lab2 (develop)
$ git checkout -b staging
git push origin staging
Switched to a new branch 'staging'
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/reenaqureshi/Lab2/pull/new/staging
remote:
To github.com:reenaqureshi/Lab2.git
 * [new branch]      staging -> staging

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/Lab2 (staging)
$
```

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (staging)
$ git checkout -b feature/login
git push origin feature/login
Switched to a new branch 'feature/login'
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/login' on GitHub by visiting:
remote:   https://github.com/reenaqureshi/Lab2/pull/new/feature/login
remote:
To github.com:reenaqureshi/Lab2.git
 * [new branch]      feature/login -> feature/login

```

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature/login)
$ git checkout -b bugfix/typo
git push origin bugfix/typo
Switched to a new branch 'bugfix/typo'
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'bugfix/typo' on GitHub by visiting:
remote:   https://github.com/reenaqureshi/Lab2/pull/new/bugfix/typo
remote:
To github.com:reenaqureshi/Lab2.git
 * [new branch]      bugfix/typo -> bugfix/typo

```

## Example Workflow:

Developers work on feature/\* and bugfix/\* branches.

Merge completed work into develop.

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (bugfix/typo)
$ git checkout develop
git merge feature/login
git push origin develop
A      master
Switched to branch 'develop'
Already up to date.
Everything up-to-date

```

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (develop)
$ git merge bugfix/typo
git push origin develop
Already up to date.
Everything up-to-date

```

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (develop)
$ git checkout staging
git merge develop
git push origin staging
A      master
Switched to branch 'staging'
Already up to date.

```

Merge develop into staging for testing.

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (develop)
$ git checkout staging
git merge develop
git push origin staging
A      master
Switched to branch 'staging'
Already up to date.

```

Finally, merge staging into main (production).

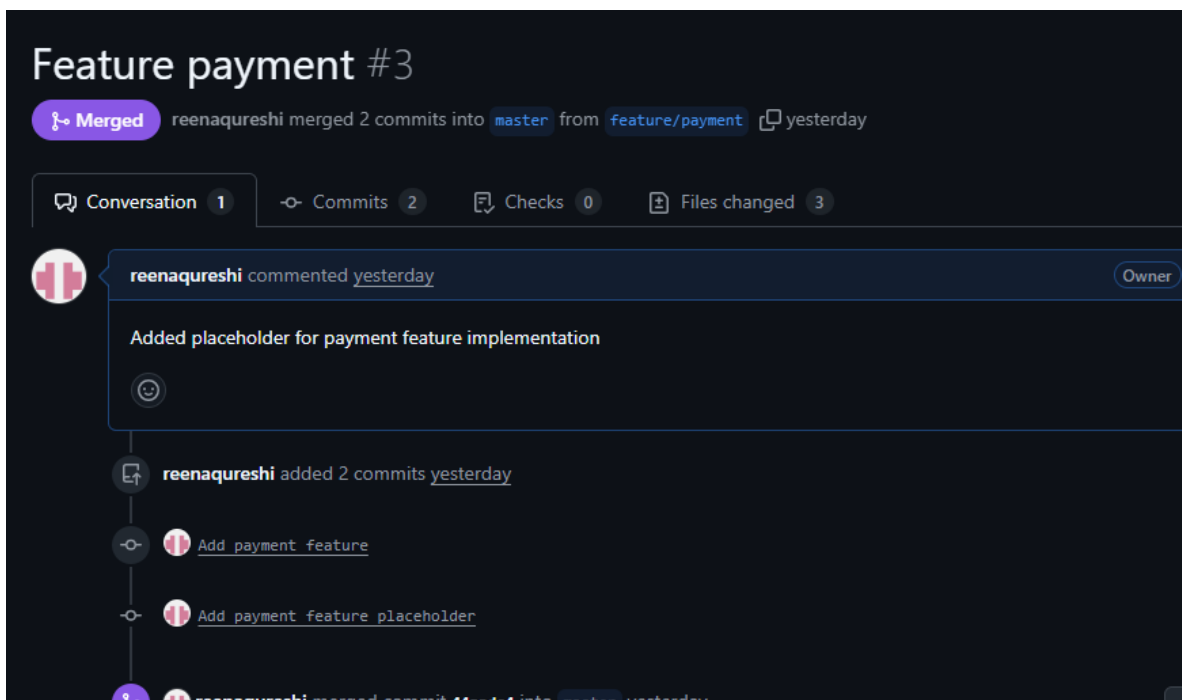
```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (staging)
$ git checkout master
git merge staging
git push origin master
A      master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
Already up to date.
```

## Task 12: Code Review Workflow

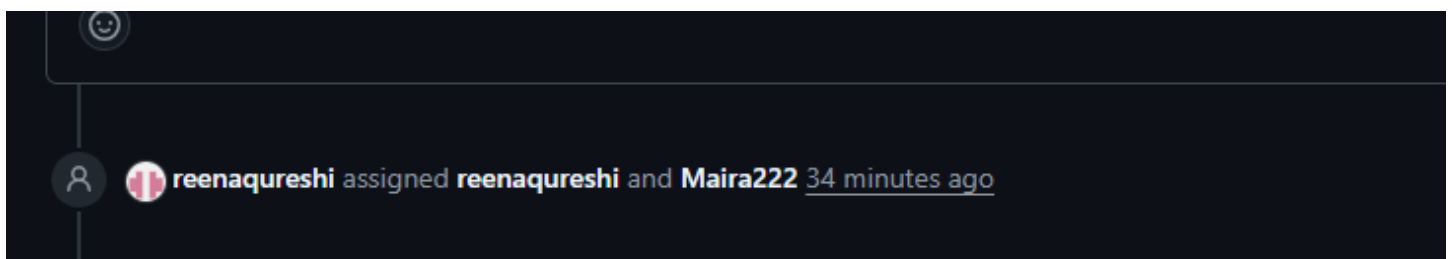
Create a Pull Request (PR) / Merge Request (MR):

From a feature branch into main.

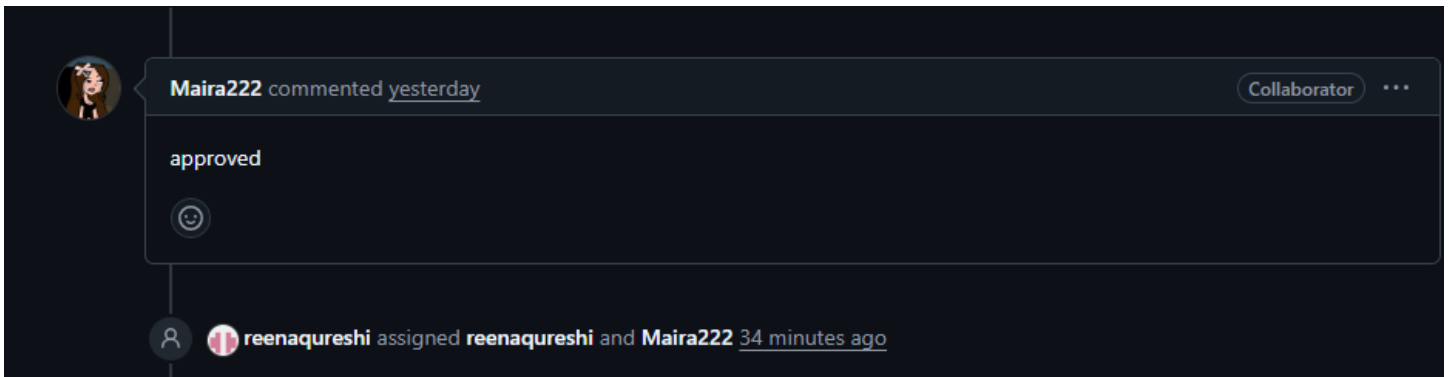
Add a clear title and description summarizing the changes.



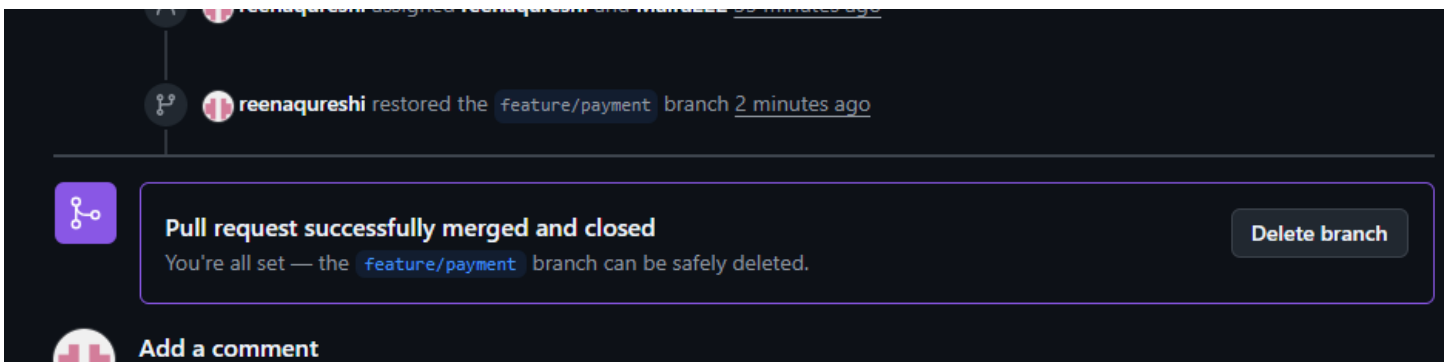
ASSIGN A REVIEWER (TEAMMATE OR SECOND ACCOUNT).



## REVIEWER ACTIONS

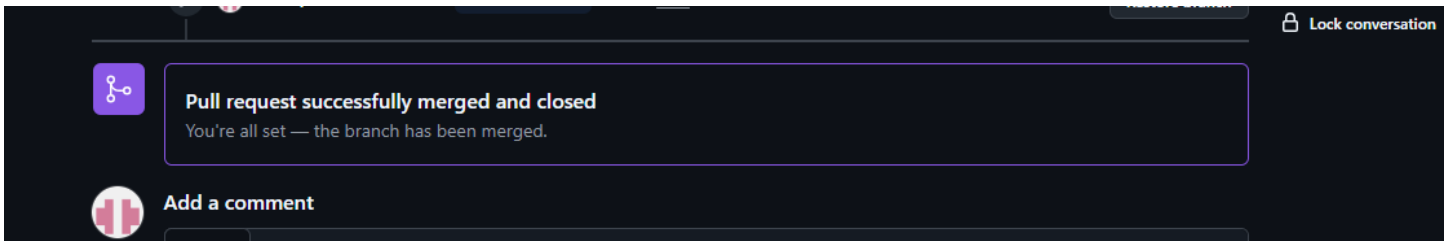


## MERGE THE PR:



## Task 13: Branch Cleanup Best Practices

Delete Remote Branch After Merge:



## Update Local Repository:

`git checkout main`

`git pull origin main`

`git branch -d <branch-name>`

`git branch`

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (master)
$ git branch
  bugfix/typo
  bugfix/user-auth-error
  develop
  feature-1
  feature/login
* master
  staging
```

## Branch “feature/payment deleted”

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (master)
$ git checkout master
git pull origin master
git branch -d feature/payment # delete locally if merged
git branch -D feature/payment # force delete if not merged
Already on 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (2/2), 1.76 KiB | 112.00 KiB/s, done.
From github.com:reenaquareshi/Lab2
 * branch                master      -> FETCH_HEAD
   7836895..44aade4      master      -> origin/master
Updating 08bd52a..44aade4
Fast-forward
 master      | 0
 payment.py  | 1 +
 payment.txt | 1 +
 3 files changed, 2 insertions(+)
 create mode 100644 master
 create mode 100644 payment.py
 create mode 100644 payment.txt
Deleted branch feature/payment (was 2e6b936).
error: branch 'feature/payment' not found

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (master)
```

## EXAM EVALUATION QUESTIONS

### Advanced Branching & Merge Verification

#### Create a new branch in your repository

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (master)
$ git checkout -b exam-branch
Switched to a new branch 'exam-branch'
```

Make a small change in a file and commit it.

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (exam-branch)
$ echo "Exam branch testing" >> notes.txt

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (exam-branch)
$ git add notes.txt
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the
next time Git touches it

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (exam-branch)
$ git commit -m "Add exam branch test line"
[exam-branch f33f9a6] Add exam branch test line
1 file changed, 1 insertion(+)

```

Merge this branch back into the main branch

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (exam-branch)
$ git checkout master
git merge exam-branch
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Updating 44aade4..f33f9a6
Fast-forward
 notes.txt | 1 +
1 file changed, 1 insertion(+)

```

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

```

* f33f9a6 (HEAD -> master, exam-branch) Add exam branch test line
| * da08c93 (origin/feature/new) add notes.txt
|/
* 44aade4 (origin/master, origin/HEAD) Merge pull request #3 from reenaqureshi/feature/
| \
| * 2e6b936 (origin/feature/payment) Add payment feature placeholder
| * a3ecbe3 Add payment feature
* | 9250067 Merge pull request #2 from reenaqureshi/feature/db-connection
| \
| * 08bd52a (origin/staging, origin/feature/login, origin/feature/db-connection, origin/d
ature/db-connection
|/
* 7836895 (origin/collab) Merge pull request #1 from reenaqureshi/feature/db-connection

```

## Multi-Stage Workflow Simulation

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

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main)
$ git checkout -b develop
git checkout -b staging
git checkout main
Switched to a new branch 'develop'
Switched to a new branch 'staging'
Switched to branch 'main'
Your branch is based on 'origin/main', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main)
$ |

```

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

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature/test)
$ git add .
$ git commit -m "Added a new feature for testing"
[feature/test 6735980] Added a new feature for testing
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 feature-1
```

Merge the feature branch into develop

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (feature/test)
$ git checkout develop
git merge feature/test
Switched to branch 'develop'
Updating f33f9a6..6735980
Fast-forward
 feature-1 | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 feature-1
```

Merge develop into staging

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (develop)
$ git checkout staging
git merge develop
Switched to branch 'staging'
Updating f33f9a6..6735980
Fast-forward
 feature-1 | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 feature-1

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (staging)
$ |
```

Merge staging into main

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (staging)
$ git checkout main
git merge staging
Switched to branch 'main'
Your branch is based on 'origin/main', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)
Updating f33f9a6..6735980
Fast-forward
 feature-1 | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 feature-1
```

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

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main)
$ git log --oneline --graph --all
* 6735980 (HEAD -> main, staging, feature/test, develop) Added a new feature for testing
* f33f9a6 Add exam branch test line
| * da08c93 (origin/feature/new) add notes.txt
|/
* 44aade4 (origin/master, origin/HEAD) Merge pull request #3 from reenaqureshi/feature/payment
|/
```



## Collaboration & Conflict Resolution

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

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (collaborator2)
$ git checkout collaborator1
Switched to branch 'collaborator1'

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (collaborator1)
$ notepad notes.txt

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (collaborator1)
$ git checkout collaborator2
Switched to branch 'collaborator2'

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (collaborator2)
$ notepad notes.txt

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (collaborator2)
$ git add notes.txt

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (collaborator2)
$ git commit -m "Collaborator 2 made a change"
[collaborator2 5f92be7] Collaborator 2 made a change
1 file changed, 1 insertion(+)
```

Attempt to merge the branches and capture the conflict.

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main)
$ git merge collaborator1
Updating 6735980..fda6d47
Fast-forward
 notes.txt | 1 +
1 file changed, 1 insertion(+)

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main)
$ git merge collaborator2
Auto-merging notes.txt
CONFLICT (content): Merge conflict in notes.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Resolve the conflict so that both contributions are preserved

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main|MERGING)
$ notepad notes.txt

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main|MERGING)
$ git add notes.txt
git commit -m "Resolved conflict and kept both contributions"
[main 7c4b610] Resolved conflict and kept both contributions

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/Lab2 (main)
$ |
```

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

File Edit Format View Help

---

This is my first note.

Adding a second line to notes.txt

# Redo Task 8

Exam branch testing

this was added by collaborator 1

this is added by collaborator 2 (and overwrites collaborator 1)