

Lab # 02

Name: Rafia Fiaz

Reg No: 2023-BSE-051

Class & Section: BSE 5-B

Submitted To: Sir Shoaib

Task 1:

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * Repository name *

 rafiafiaz-051 / cc_rafiafiaz_051_lab02
 cc_rafiafiaz_051_lab02 is available.

Great repository names are short and memorable. How about [animated-doodle](#)?

Description

0 / 350 characters

2 Configuration

Choose visibility *  Private

Choose who can see and commit to this repository

Add README Off

READMEs can be used as longer descriptions. [About READMEs](#)

Task 2:

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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Mejhish> ssh-keygen -t ed25519 -C "23-22411-051@se.fjwu.edu.pk"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\Mejhish/.ssh/id_ed25519):
Created directory 'C:\Users\Mejhish\.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Mejhish/.ssh/id_ed25519
Your public key has been saved in C:\Users\Mejhish/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:6YreAGZo0XpA8RJn6auVYArMHqIr3ZX2pmb1/NRk5aY 23-22411-051@se.fjwu.edu.pk
The key's randomart image is:
+--[ED25519 256]--+
| +.o.
| . *.
| o+.o .
| oB=. . o
| Oo*.o .S o o
| ++.= +.. + o
| oo..o o.o . E
| o... .+o.o o.
| . .oo+o ..
+---[SHA256]---+
PS C:\Users\Mejhish> |
```

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

The screenshot shows the GitHub Settings page for the user 'rafiatiaz-051'. The left sidebar is visible with options like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and licensing, Emails, Password and authentication, SSH and GPG keys (which is selected), and Organizations.

The main content area is titled 'SSH keys' and contains the following text: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Below this, there is a table for the 'My Laptop Key' entry:

	My Laptop Key	SHA256:6YreAGZo0XpA8RJn6auVYArMHqIr3ZX2pmb1/NRk5aY	Added on Oct 12, 2023	Never used — Read/write	Delete
--	---------------	--	-----------------------	-------------------------	------------------------

Below the table, there is a link: 'Check out our guide to [connecting to GitHub using SSH keys](#) or [troubleshoot common SSH problems](#)'.

The 'GPG keys' section below shows the message: 'There are no GPG keys associated with your account.'

3. Clone your Lab2 repo using SSH:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Mehwish> git clone git@github.com:rafiafiaz-051/Lab2.git
Cloning into 'Lab2'...
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbhpZisF/zLDA0zPMSvHdkr4UvC0qU.
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.
PS C:\Users\Mehwish>
```

Task 3:

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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Mehwish> git config --global user.name "Rafia Fiaz"
PS C:\Users\Mehwish> git config --global user.email "23-22411-051@se.fjwu.edu.pk"
PS C:\Users\Mehwish>
```

2. Verify your configuration:

```
PS C:\Users\Mehwish> 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=Rafia Fiaz
user.email=23-22411-051@se.fjwu.edu.pk
PS C:\Users\Mehwish>
```

Task 4:

Explore the .git Folder:

```
MINGW64:/c/Users/Mehwish/Lab2

Mehwish@Home MINGW64 ~
$ cd Lab2

Mehwish@Home MINGW64 ~/Lab2 (main)
$ ls -a
./ ../ .git/

Mehwish@Home MINGW64 ~/Lab2 (main)
$ ls -a .git
./ ../ HEAD config description hooks/ info/ objects/ refs/

Mehwish@Home MINGW64 ~/Lab2 (main)
$
```

Task 5:

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

```
MINGW64:/c/Users/Mehwish/Lab2

Mehwish@Home MINGW64 ~/Lab2
$ rm -rf .git

Mehwish@Home MINGW64 ~/Lab2
$ ls -a #Confirm .git is gone
./ ../
```

2. Re-initialize the local git repository:

```
Mehwish@Home MINGW64 ~/Lab2
$ git init
Initialized empty Git repository in c:/users/Mehwish/Lab2/.git/

Mehwish@Home MINGW64 ~/Lab2 (master)
$
```

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

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ echo "# Lab2 Git Practice" > README.md

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git commit -m "Initial commit"
[master (root-commit) 391af1b] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

4. Connect your local repo to GitHub and push:

```
MINGW64:/c/Users/Mehwish/Lab2

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git remote add origin git@github.com:rafiafiaz-051/Lab2.git

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 243 bytes | 81.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:rafiafiaz-051/Lab2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Task 6:

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

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ echo "This is my note for Git Task 6." > notes.txt

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

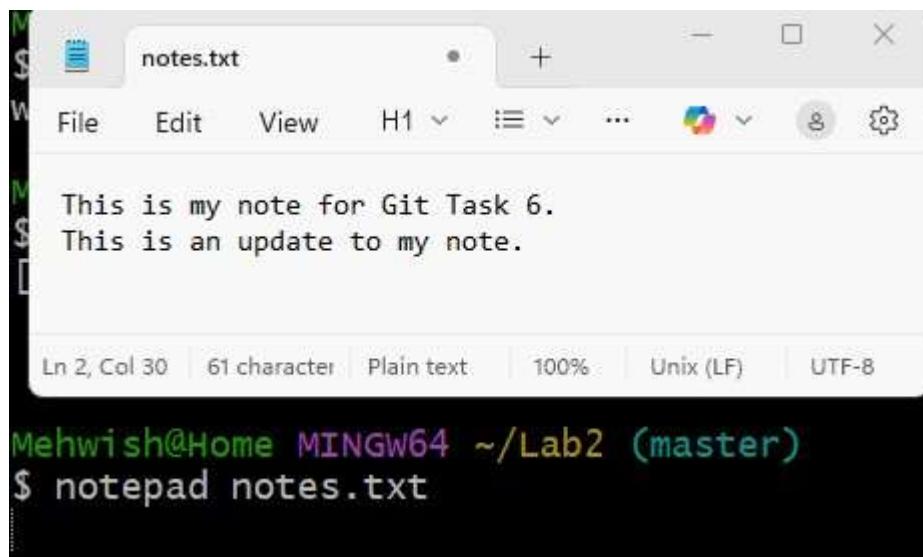
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md~
    notes.txt

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

2. Stage and commit:

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git add notes.txt
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the next time Git touches it

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git commit -m "Add notes.txt"
[master ec4a2da] Add notes.txt
 1 file changed, 1 insertion(+)
 create mode 100644 notes.txt
```



3. Edit notes.txt and repeat status/commit steps:

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git add notes.txt
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the next time Git touches it

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git commit -m "Update notes.txt"
[master e317f6d] Update notes.txt
 1 file changed, 1 insertion(+)

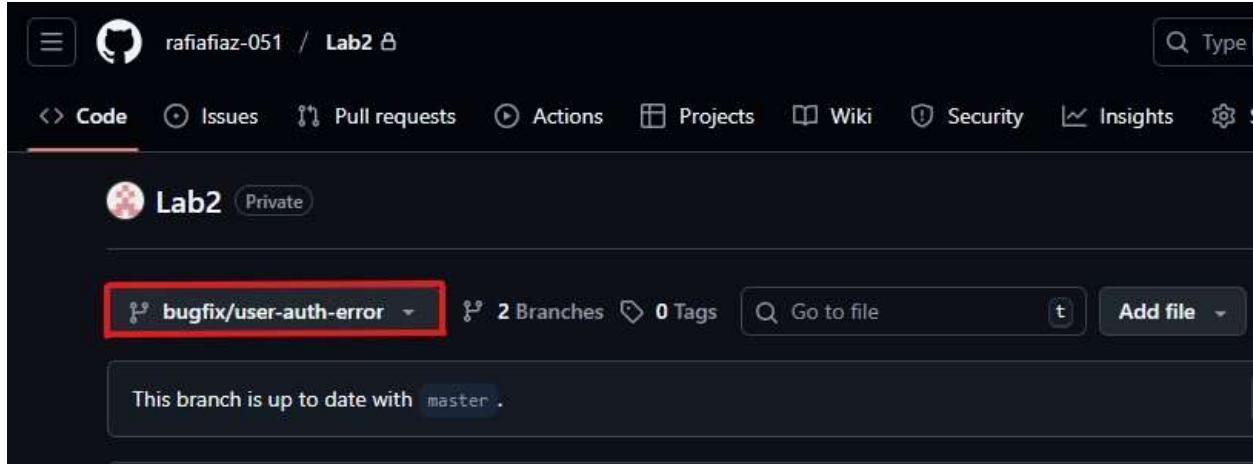
Mehwish@Home MINGW64 ~/Lab2 (master)
$ rm README.md

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Task 7:

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



2. Pull the branch to your local repository to sync.

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git pull origin bugfix/user-auth-error
From github.com:rafiafiaz-051/Lab2
 * branch            bugfix/user-auth-error -> FETCH_HEAD
 * [new branch]      bugfix/user-auth-error -> origin/bugfix/user-auth-error
Already up to date.

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git branch -a
* master
  remotes/origin/bugfix/user-auth-error
  remotes/origin/master

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git checkout bugfix/user-auth-error
branch 'bugfix/user-auth-error' set up to track 'origin/bugfix/user-auth-error'.
Switched to a new branch 'bugfix/user-auth-error'
```

Task 8:

Create a branch named feature/db-connection using Git Bash & Push the branch to the remote repository.

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

Mehwish@Home 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/rafiafiaz-051/Lab2/pull/new/feature/db-connection
remote:
To github.com:rafiafiaz-051/Lab2.git
 * [new branch]      feature/db-connection -> feature/db-connection
```

Task 9:

1. Create and switch to a new branch:

```
Mehwish@Home MINGW64 ~/Lab2 (feature/db-connection)
$ git checkout -b feature-1
Switched to a new branch 'feature-1'

Mehwish@Home MINGW64 ~/Lab2 (feature-1)
$
```

2. Modify main.py:

```
Mehwish@Home MINGW64 ~/Lab2 (feature-1)
$ git add main.py

Mehwish@Home MINGW64 ~/Lab2 (feature-1)
$ git commit -m "Add new function to main.py"
[feature-1 dae6b8a] Add new function to main.py
 1 file changed, 2 insertions(+)
 create mode 100644 main.py
```

3. Switch back to main and merge:

```
Mehwish@Home MINGW64 ~/Lab2 (feature-1)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 4 commits.
  (use "git push" to publish your local commits)

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git merge feature-1
Already up to date.
```

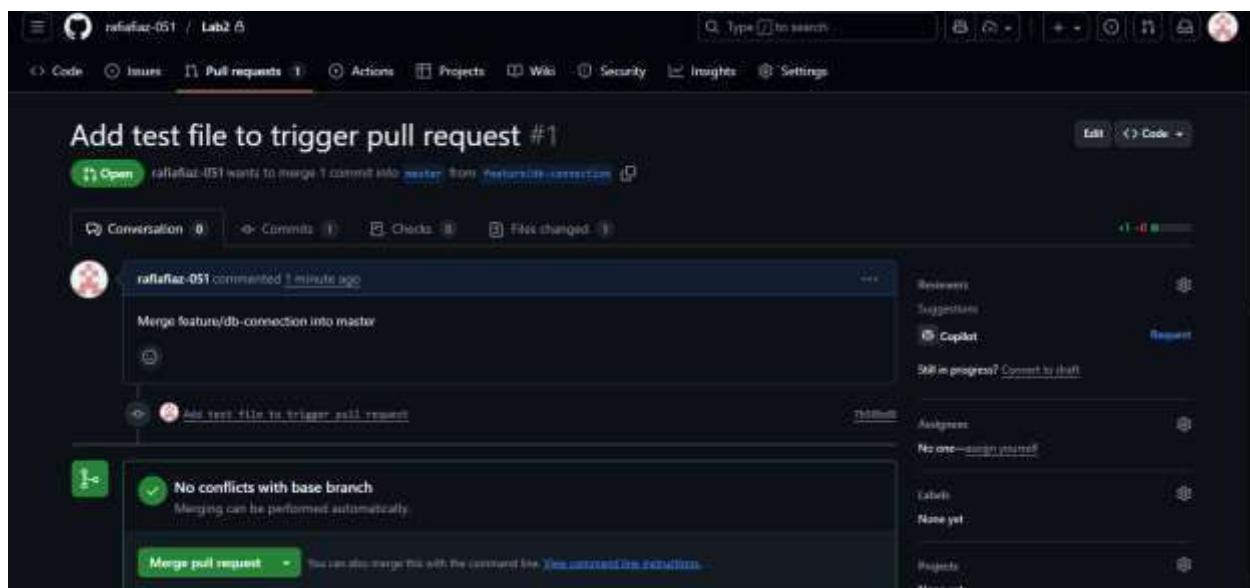
4. Push all branches:

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git push origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 1.09 KiB | 139.00 KiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:rafiafiaiz-051/Lab2.git
  391af1b..7d3fa15  master -> master

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git push origin feature-1
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/rafiafiaiz-051/Lab2/pull/new/feature-1
remote:
To github.com:rafiafiaiz-051/Lab2.git
 * [new branch]      feature-1 -> feature-1
```

Task 10:

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



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

Add test file to trigger pull request #1

Merged rafiafiaz-051 merged 1 commit into `master` from `feature/db-connection` 1 minute ago

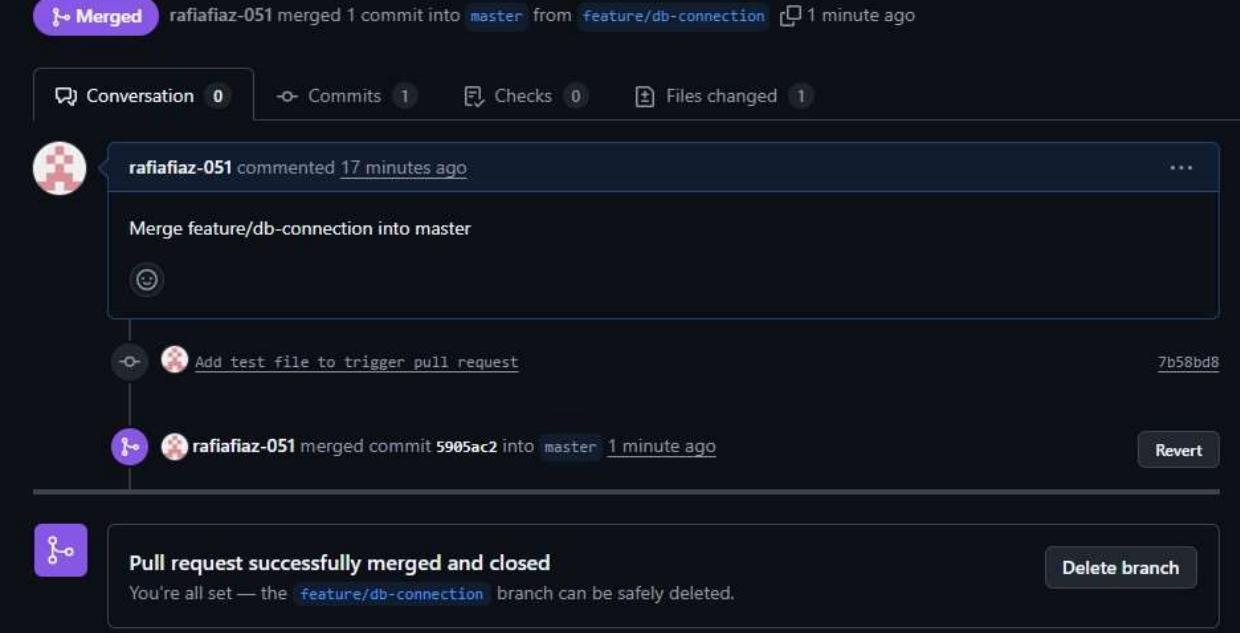
Conversation 0 Commits 1 Checks 0 Files changed 1

rafiafiaz-051 commented 17 minutes ago
Merge feature/db-connection into master

Add test file to trigger pull request 7b58bd8

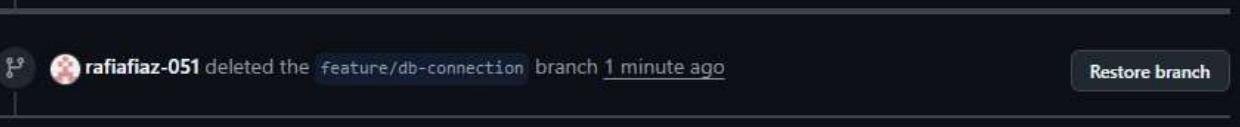
rafiafiaz-051 merged commit 5905ac2 into `master` 1 minute ago Revert

Pull request successfully merged and closed Delete branch
You're all set — the `feature/db-connection` branch can be safely deleted.



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

rafiafiaz-051 deleted the `feature/db-connection` branch 1 minute ago Restore branch



Task 11:

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).

- Merges into develop and staging
(branch_merges.png)
- Final merge into main
(final_merge.png)

```
Mehwish@Home MINGW64 ~/Lab2 (feature/db-connection)
$ git checkout master
git pull origin master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (2/2), 995 bytes | 45.00 KiB/s, done.
From github.com:rafiafiafiaz-051/Lab2
 * branch           master      -> FETCH_HEAD
   7d3fa15..5905ac2  master      -> origin/master
Updating 7d3fa15..5905ac2
Fast-forward
 test.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 test.md

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git branch develop
git branch staging

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git branch -a
bugfix/user-auth-error
develop
feature-1
feature/db-connection
* master
staging
remotes/origin/bugfix/user-auth-error
remotes/origin/feature-1
remotes/origin/feature/db-connection
remotes/origin/master
```

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git checkout -b feature/add-login
Switched to a new branch 'feature/add-login'

Mehwish@Home MINGW64 ~/Lab2 (feature/add-login)
$ echo "# Added login feature" >> README.md

Mehwish@Home MINGW64 ~/Lab2 (feature/add-login)
$ git add .
git commit -m "Add login feature"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[feature/add-login e70278a] Add login feature
 1 file changed, 1 insertion(+)

Mehwish@Home MINGW64 ~/Lab2 (feature/add-login)
$ git checkout develop
git merge feature/add-login
Switched to branch 'develop'
Updating 5905ac2..e70278a
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)

Mehwish@Home MINGW64 ~/Lab2 (develop)
$ git checkout -b bugfix/fix-typo
Switched to a new branch 'bugfix/fix-typo'

Mehwish@Home MINGW64 ~/Lab2 (bugfix/fix-typo)
$ git add .
git commit -m "Fix typo"
On branch bugfix/fix-typo
nothing to commit, working tree clean

Mehwish@Home MINGW64 ~/Lab2 (bugfix/fix-typo)
$ git checkout develop
git merge bugfix/fix-typo
Switched to branch 'develop'
Already up to date.
```

```
Mehwish@Home MINGW64 ~/Lab2 (staging)
$ git checkout develop
git merge feature/add-login
Switched to branch 'develop'
Already up to date.
```

```
Mehwish@Home MINGW64 ~/Lab2 (develop)
$ git checkout staging
git merge develop
Switched to branch 'staging'
Already up to date.
```

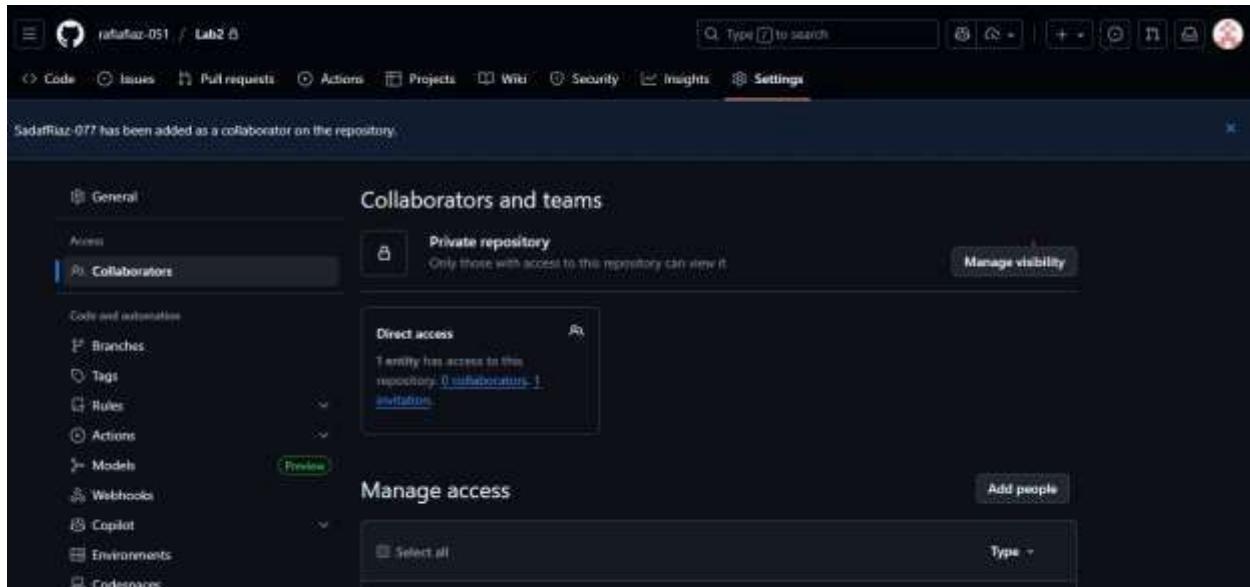
```
Mehwish@Home MINGW64 ~/Lab2 (staging)
$ git checkout master
git merge staging
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Updating 5905ac2..e70278a
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```

Bonus Task:

1. Create a branch collab.

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git checkout -b collab
Switched to a new branch 'collab'
```

2. Add a collaborator on your GitHub repository.



3. Both users:

- o Pull latest changes.
- o Add their name and fun fact to notes.txt.
- o Commit and push to the collab branch.

```
Mehwish@Home MINGW64 ~/Lab2 (collab)
$ echo "Rafia Fiaz - I love coding late at night!" > notes.txt

Mehwish@Home MINGW64 ~/Lab2 (collab)
$ git add notes.txt
git commit -m "Added my name and fun fact to notes.txt"
git push origin collab
warning: in the working copy of 'notes.txt', LF will be replaced by CRLF the next time Git touches it
[collab 4a88a99] Added my name and fun fact to notes.txt
 1 file changed, 1 insertion(+), 2 deletions(-)
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), 325 bytes | 325.00 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 'collab' on GitHub by visiting:
remote:     https://github.com/rafiafiaz-051/Lab2/pull/new/collab
remote:
To github.com:rafiafiaz-051/Lab2.git
 * [new branch]      collab -> collab
```

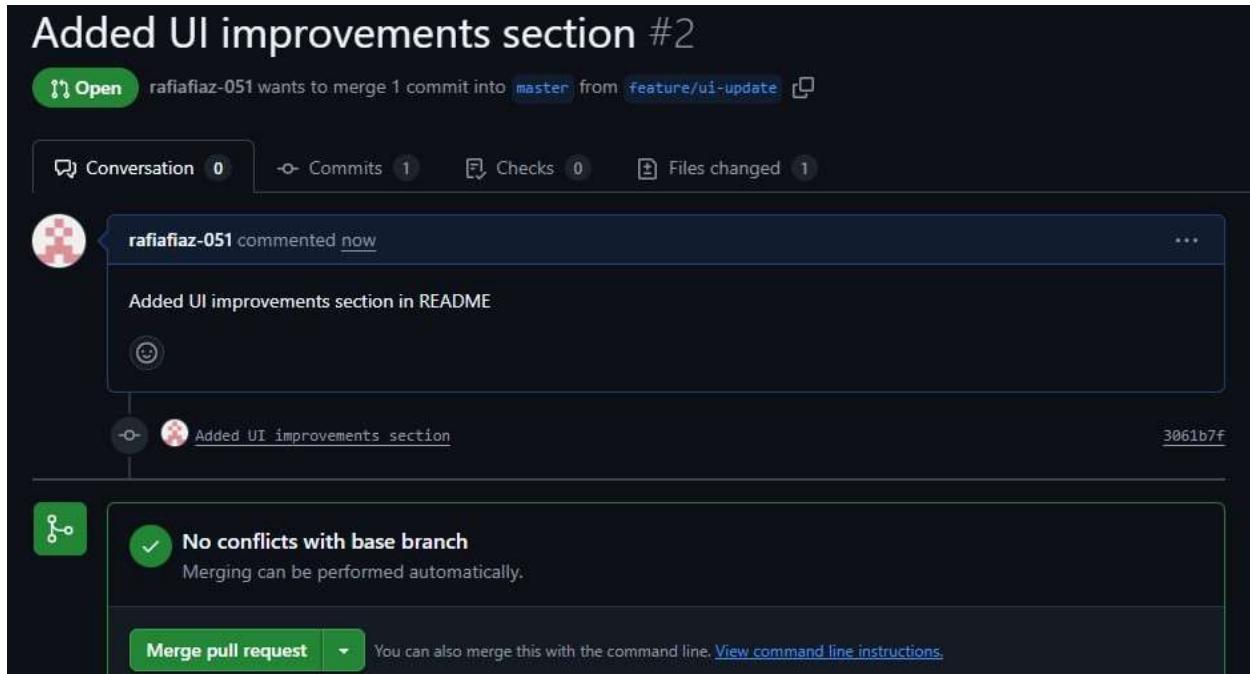
4. Merge collab into main and push.

```
git checkout main  
git merge collab  
push origin main
```

```
Mehwish@Home MINGW64 ~/Lab2 (collab)  
$ git checkout master  
git merge collab  
git push origin master  
Switched to branch 'master'  
Your branch is up to date with 'origin/master'.  
Updating e70278a..4a88a99  
Fast-forward  
 notes.txt | 3 +--  
 1 file changed, 1 insertion(+), 2 deletions(-)  
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To github.com:rafiafiafiaz-051/Lab2.git  
 e70278a..4a88a99 master -> master
```

Task 12:

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



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



3. Reviewer Actions :

- **Approve the PR**

A screenshot of a GitHub pull request review comment. The comment is from a user named "rafa-reviewer" and says:

rafa-reviewer reviewed now

rafa-reviewer left a comment

Everything looks great. Approved ✅

(smiley face)

View reviewed changes

- **Request Changes**

A screenshot of a GitHub pull request review comment. The comment is from a user named "rafa-reviewer" and says:

rafa-reviewer now

Please fix formatting.

(smiley face)

The comment is in response to a code diff in the file "README.md". The diff shows the following changes:

```
diff --git a/README.md b/README.md
index 1234567..8901234 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,3 @@
1 1 # Lab2 Git Practice
2 2 # Added login feature
3 + Added UI improvements section
```

Reply...

Resolve conversation

4. Author Actions:

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git checkout feature/ui-update
Switched to branch 'feature/ui-update'

Mehwish@Home MINGW64 ~/Lab2 (feature/ui-update)
$ git pull origin feature/ui-update
From github.com:rafiafiaz-051/Lab2
 * branch            feature/ui-update -> FETCH_HEAD
Already up to date.

Mehwish@Home MINGW64 ~/Lab2 (feature/ui-update)
$ nano README.md

Mehwish@Home MINGW64 ~/Lab2 (feature/ui-update)
$ git add .
git commit -m "Fixed issues as per review feedback"
[feature/ui-update 9c13720] Fixed issues as per review feedback
 1 file changed, 1 insertion(+)

Mehwish@Home MINGW64 ~/Lab2 (feature/ui-update)
$ git push origin feature/ui-update
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), 441 bytes | 147.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:rafiafiaz-051/Lab2.git
 3061b7f..9c13720  feature/ui-update -> feature/ui-update
```

Fixed issues as per review feedback #3

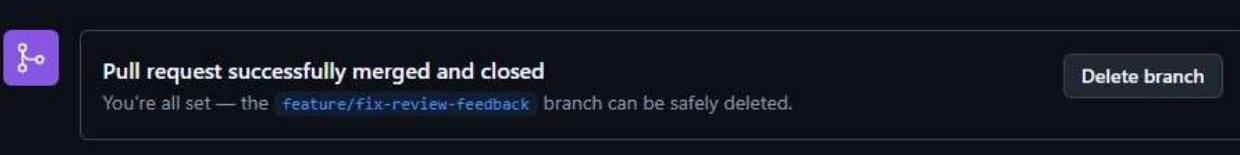
1 Open rafiafiaz-051 wants to merge 3 commits into `master` from `feature/fix-review-feedback`

Conversation 0 · Commits 3 · Checks 0 · Files changed 0

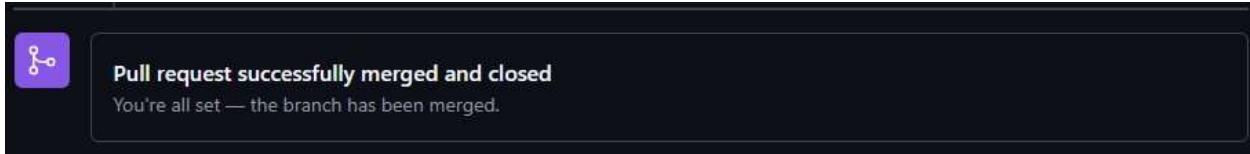
Commits on Oct 17, 2025

Fixed issues as per review feedback rafiafiaz-051 committed 1 hour ago	3727863
Fixed issues as per review feedback rafiafiaz-051 committed 18 minutes ago	3944669
Merge branch 'master' into feature/fix-review-feedback rafiafiaz-051 authored 1 minute ago	Verified 4213457

5. Merge the PR:



6. Branch Handling After Merge:



Task 13:

1. Delete Remote Branch After Merge:

```
Mehwish@Home MINGW64 ~/Lab2 (feature/fix-review-feedback)
$ git checkout master
git pull origin master
git branch -d feature/fix-review-feedback
git branch -d feature/db-connection
Switched to branch 'master'
Your branch is behind 'origin/master' by 6 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)
From github.com:rafiafiaz-051/Lab2
 * branch           master      -> FETCH_HEAD
Updating 3727863..7ab0c99
Fast-forward
 README.md | 6 ++++++
 1 file changed, 6 insertions(+)
Deleted branch feature/fix-review-feedback (was 38a1dc6).
error: branch 'feature/db-connection' not found

Mehwish@Home MINGW64 ~/Lab2 (master)
$ git branch
 bugfix/fix-typo
 bugfix/user-auth-error
 collab
 develop
 feature-1
 feature/add-login
 feature/ui-update
* master
 staging
```

2. Update Local Repository

```
Mehwish@Home MINGW64 ~/Lab2 (master)
$ git checkout master
git pull origin master
git branch -d feature/fix-review-feedback
git branch -d feature/db-connection
Already on 'master'
Your branch is up to date with 'origin/master'.
From github.com:rafiyahfiaz-051/Lab2
 * branch            master    -> FETCH_HEAD
Already up to date.
error: branch 'feature/fix-review-feedback' not found
error: branch 'feature/db-connection' not found
```