



Cloud Computing Lab 2

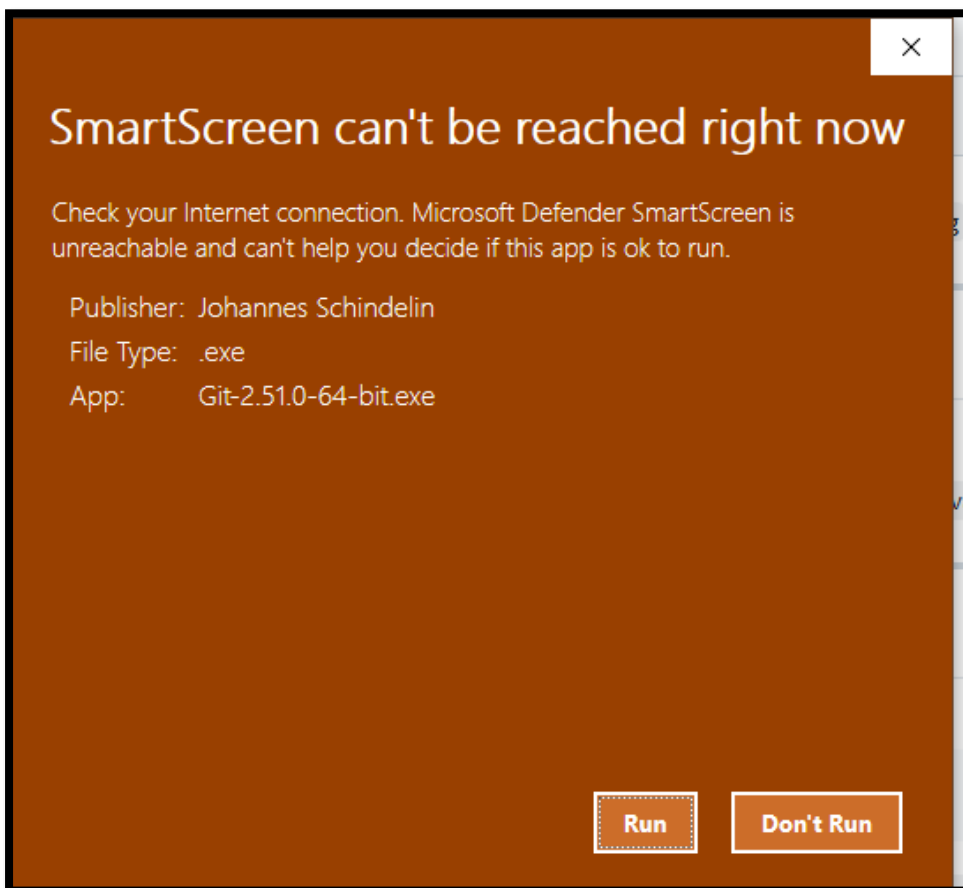
Submitted to: Engr. Muhammad Shoaib

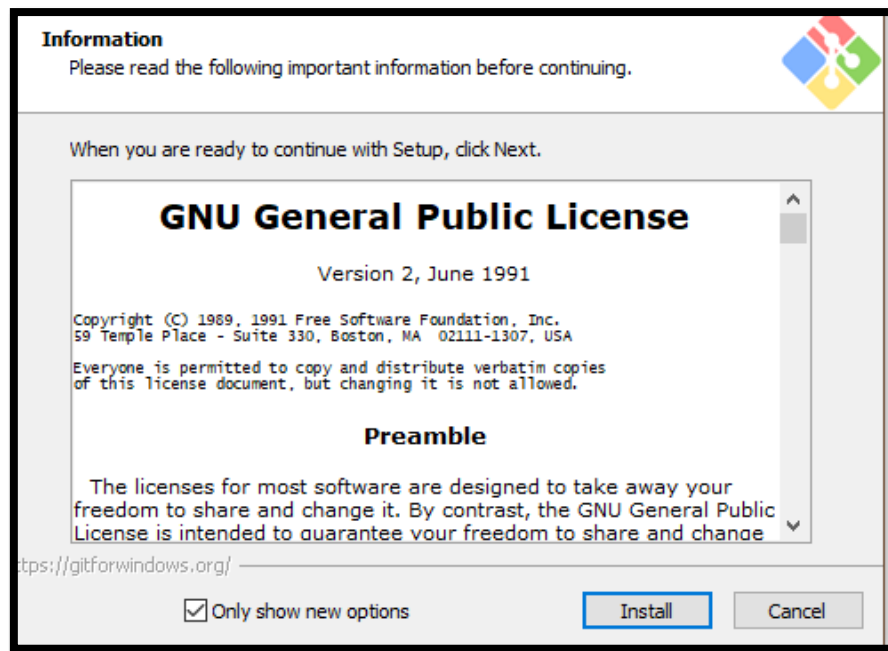
Submitted by: Anara Hayat

Reg#No: 2023-BSE-008

Lab Title: Hands-On Git & Version Control Lab

Install Git on PC





```
MINGW64:/c/Users/anara
anara@DESKTOP-FCGAHUV MINGW64 ~
$ git --version
git version 2.51.0.windows.1
anara@DESKTOP-FCGAHUV MINGW64 ~
$
```

Task 1: Create Private GitHub Repository

Required fields are marked with an asterisk (*).

1 General

Owner * Anara-hayat / Repository name * lab 2

☒ Your new repository will be created as lab-2.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. How about **cuddly-telegram**?

Description

0 / 350 characters

2 Configuration

Choose visibility * Private

Choose who can see and commit to this repository

Task 2: Connect Repository via SSH

1. Generate a new SSH key using PowerShell

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\anara> ssh-keygen -t ed25519 -C "23-22411-008@se.fjwu.edu.pk"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\anara\.ssh\id_ed25519): ssh_keygen.png
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh_keygen.png
Your public key has been saved in ssh_keygen.png.pub
The key fingerprint is:
SHA256:P7B+V+oRtO10ggHHgX+SRy2GYAuHLCdhj5zVC4ChTbA 23-22411-008@se.fjwu.edu.pk
The key's randomart image is:
+--[ED25519 256]--+
|..o++o.o+o.o.o|
|=...o=oo...+ |
|E . . o.+o.+ |
| . = . ++.o |
| + . S .oo . |
| + o o . |
| . + ooo |
| . ..o= |
| .. o . |
+-----[SHA256]-----+
PS C:\Users\anara>
```

2. Add your SSH public key to GitHub



Add new SSH Key

Title:

Key type: Authentication Key

Key:

3. Clone your Lab2 repo using SSH

```
anara@DESKTOP-FCGAHUV MINGW64 ~ (master)
$ git remote add origin git@github.com:Anara-hayat/lab2.git
```

```
anara@DESKTOP-FCGAHUV MINGW64 ~ (master)
$ git clone git@github.com:Anara-hayat/lab2.git
Cloning into 'lab2'...
```

Task 3: Configure Git Username and Email

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

```
anara@DESKTOP-FCGAHUV MINGW64 ~ (master)
$ git config --global user.name "Anara-hayat"

anara@DESKTOP-FCGAHUV MINGW64 ~ (master)
$ git config --global user.email "23-22411-008@se.fjwu.edu.pk"
```

2. Verify your configuration:

```
credential.https://dev.azure.com.usenttpath=true
init.defaultbranch=master
core.editor="C:\Users\anara\AppData\Local\Programs
-wait
user.name=Anara-hayat
user.email=23-22411-008@se.fjwu.edu.pk
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
```

Task 4: Explore the .git Folder

```
anara@DESKTOP-FCGAHUV MINGW64 ~ (master)
$ ls -la .git
./ ../ HEAD config description hooks/ info/ objects/ refs/
```

Task 5: Local Repository Management

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

```
anara@DESKTOP-FCGAHUV MINGW64 ~ (master)
$ rm -rf .git
```

2. Re-initialize the local git repository

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git init
Reinitialized existing Git repository in C:/Users/anara/lab2/.git/
```

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

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ echo "# lab 2 git practice" > README.md

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will
t time Git touches it

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git commit -m "Initial commit"
[master 852a6af] Initial commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

4. Connect your local repo to GitHub and push:

```

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git remote add origin git@github.com: Anara-hayat/lab2.git
usage: git remote add [<options>] <name> <url>

    -f, --[no-]fetch          fetch the remote branches
    --[no-]tags              import all tags and associated objects when fetching
                             or do not fetch any tag at all (--no-tags)
    -t, --[no-]track <branch>
                             branch(es) to track
    -m, --[no-]master <branch>
                             master branch
    --[no-]mirror[=(push|fetch)]
                             set up remote as a mirror to push to or fetch from

```

```

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 271 bytes | 135.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Anara-hayat/lab2.git
   d2a1f2a..852a6af  master -> master
branch 'master' set up to track 'origin/master'.

```

Task 6: File Status & Staging

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

```

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ touch notes.txt

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ echo "Hello Hithub!" >notes.txt

anara@DESKTOP-FCGAHUV 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)
    lab2/
    notes.txt

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

```

3. Stage and commit

```
anara@DESKTOP-FCGAHUV 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

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git commit -m "Add notes.txt"
[master b70689e] Add notes.txt
1 file changed, 1 insertion(+)
create mode 100644 notes.txt
```

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

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git commit -m "second commit"
[master 23a92e2] second commit
1 file changed, 1 insertion(+), 1 deletion(-)

anara@DESKTOP-FCGAHUV 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)


Untracked files:
  (use "git add <file>..." to include in what will be committed)
    lab2/

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

Task 7: Branch Creation Using GitHub GUI

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


Rulesets / New branch ruleset

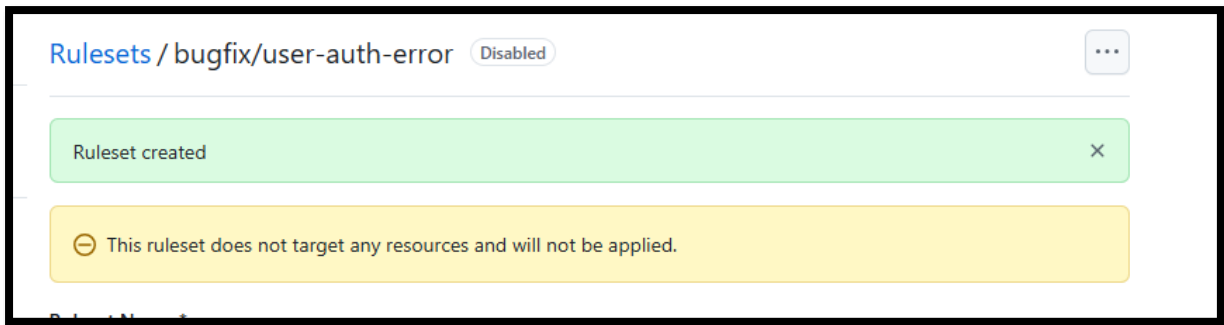
 **Protect your most important branches**
[Rulesets](#) define whether collaborators can delete or force push and set requirements for any pushes, such as passing status checks or a linear commit history.

Ruleset Name *

bugfix/user-auth-error

Enforcement status

 Disabled



2. Pull the branch to your local repository to sync

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git checkout -b bugfix/user-auth-error origin/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'

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bugfix/user-auth-error)
$ git pull origin bugfix/user-auth-error
From github.com:Anara-hayat/lab2
* branch                bugfix/user-auth-error -> FETCH_HEAD
Already up to date.

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bugfix/user-auth-error)
$ git branch
* bugfix/user-auth-error
master
```

8: Branch Creation and Push Using Git Bash

Create a branch named feature/db-connection using Git Bash and push to remote repository

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bugfix/user-auth-error)
$ git checkout -b feature/db-connection
Switched to a new branch 'feature/db-connection'

anara@DESKTOP-FCGAHUV 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/Anara-hayat/lab2/pull/new/feature/db-connection
remote:
To github.com:Anara-hayat/lab2.git
* [new branch]      feature/db-connection -> feature/db-connection
```

9: Branching & Merging

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

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (feature/db-connection)
$ git checkout -b feature-1
Switched to a new branch 'feature-1'
```

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


```

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (feature-1)
$ touch main.py

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (feature-1)
$ git add main.py

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (feature-1)
$ git commit -m "Add new function to main.py"
[feature-1 ff76a40] Add new function to main.py
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 main.py

```

3. Switch back to main and merge:

```

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (feature-1)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git merge feature-1
Merge made by the 'ort' strategy.
main.py | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 main.py

```

4. Push all branches

```

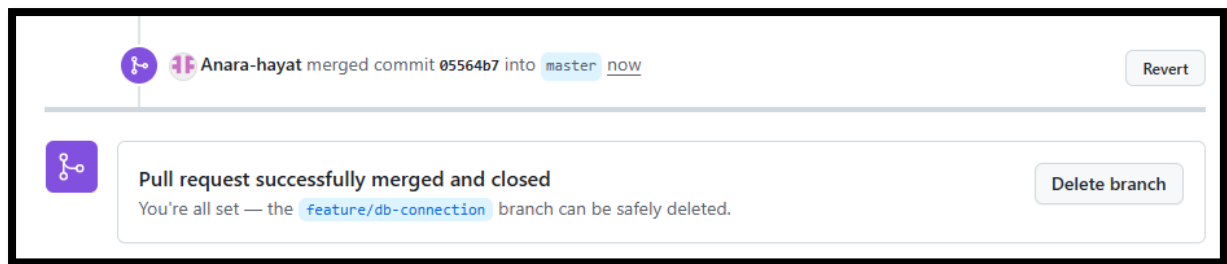
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git push origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 1002 bytes | 250.00 KiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:Anara-hayat/lab2.git
852a6af..781fb86 master -> master

anara@DESKTOP-FCGAHUV 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/Anara-hayat/lab2/pull/new/feature-1
remote:
To github.com:Anara-hayat/lab2.git
* [new branch]   feature-1 -> feature-1

```

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

The Pull Request review and merge



The branch deletion



Branch deleted

Task 11: Detailed Branch Strategy (Develop/Staging)

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

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

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git checkout -b develop
Switched to a new branch 'develop'

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (develop)
$ git checkout -b staging
Switched to a new branch 'staging'

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (staging)
$ git checkout -b feature/*
fatal: 'feature/*' is not a valid branch name
hint: See 'man git check-ref-format'
hint: Disable this message with "git config set advice.refSyntax false"

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (staging)
$ git checkout -b feature/feature-new
Switched to a new branch 'feature/feature-new'

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (feature/feature-new)
$ git checkout -b bugfix/bugfix-new
Switched to a new branch 'bugfix/bugfix-new'
```

```

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bigfix/bugfix-new)
$ git branch
* bigfix/bugfix-new
  bugfix/user-auth-error
  develop
  feature-1
  feature/db-connection
  feature/feature-new
  master
  staging

```

2. Pushed the local branches to the remote git

```

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bigfix/bugfix-new)
$ git push -u origin bigfix/bugfix-new
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 862 bytes | 173.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'bigfix/bugfix-new' on GitHub by visiting:
remote:   https://github.com/Anara-hayat/lab2/pull/new/bigfix/bugfix-new
remote:
To github.com:Anara-hayat/lab2.git
 * [new branch]      bigfix/bugfix-new -> bigfix/bugfix-new
branch 'bigfix/bugfix-new' set up to track 'origin/bigfix/bugfix-new'.

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bigfix/bugfix-new)
$ git push -u origin develop
Total 0 (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/Anara-hayat/lab2/pull/new/develop
remote:
To github.com:Anara-hayat/lab2.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bigfix/bugfix-new)
$ git push -u origin 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/Anara-hayat/lab2/pull/new/staging
remote:
To github.com:Anara-hayat/lab2.git
 * [new branch]      staging -> staging
branch 'staging' set up to track 'origin/staging'.

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (bigfix/bugfix-new)
$ git push -u origin feature/feature-new
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/feature-new' on GitHub by visiting:
remote:   https://github.com/Anara-hayat/lab2/pull/new/feature/feature-new
remote:

```

3. Merging feature/feature-new and bigfix/bugfix-new into develop, and develop into staging

Anara-hayat commented 1 minute ago Author ...

Merged feature/feature-new into develop

😊

Anara-hayat commented 1 minute ago ...

merging bugfix/bugfix-new into develop branch

😊

🔗 Create bugfix-new.txt Verified [79f1a63](#)

🔗 Anara-hayat merged commit 66b6026 into develop 1 minute ago Revert

🔗 Merged Develop #4
Anara-hayat merged 5 commits into staging from develop 1 minute ago

😊

🔗 Anara-hayat added 5 commits 14 minutes ago

- 🔗 Create feature-new Verified [0f88d89](#)
- 🔗 Create develop.txt Verified [de8ff38](#)
- 🔗 Create bugfix-new.txt Verified [79f1a63](#)
- 🔗 Merge pull request #2 from Anara-hayat/feature/feature-new Verified [4dbaa8a](#)
- 🔗 Merge pull request #3 from Anara-hayat/bugfix/bugfix-new Verified [66b6026](#)

🔗 Anara-hayat merged commit 52640a7 into staging 1 minute ago Revert

4. Final merge

🔗 Merged Staging #5
Anara-hayat merged 7 commits into master from staging now

😊

🔗 Anara-hayat added 7 commits 19 minutes ago

- 🔗 Create feature-new Verified [0f88d89](#)
- 🔗 Create staging.txt Verified [cc5c3e4](#)
- 🔗 Create develop.txt Verified [de8ff38](#)
- 🔗 Create bugfix-new.txt Verified [79f1a63](#)
- 🔗 Merge pull request #2 from Anara-hayat/feature/feature-new Verified [4dbaa8a](#)
- 🔗 Merge pull request #3 from Anara-hayat/bugfix/bugfix-new Verified [66b6026](#)
- 🔗 Merge pull request #4 from Anara-hayat/develop Verified [52640a7](#)

🔗 Anara-hayat merged commit a3f311c into master now Revert

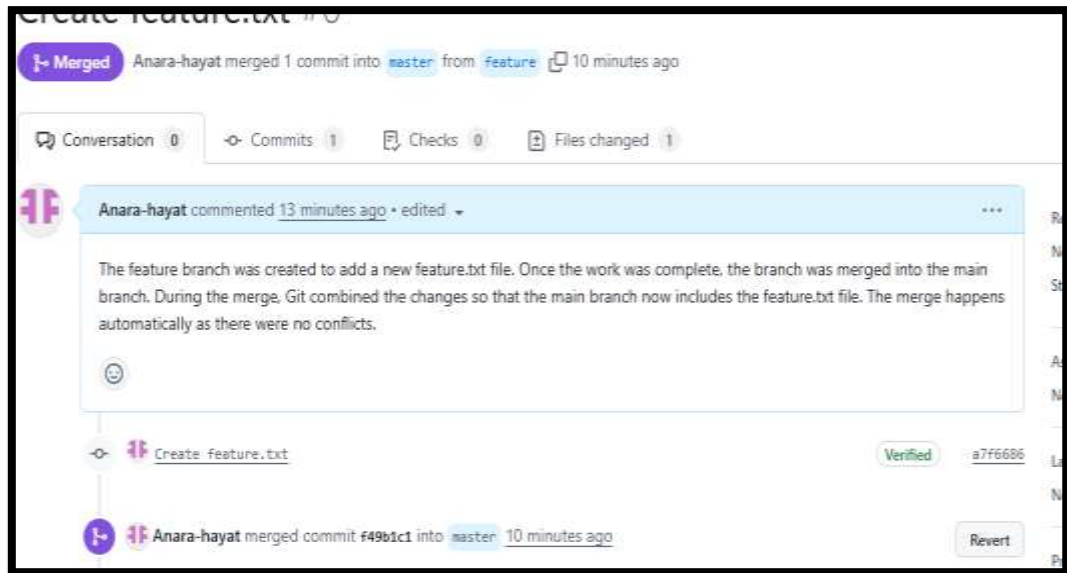
🔗 Anara-hayat commented now Author ...

finally merged staging into main

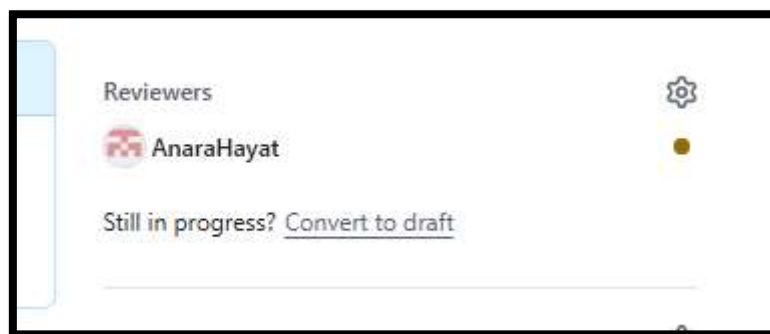
Task 12: Code Review Workflow

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

From a feature branch into main.



2. Assigned reviewer



3. Reviewer Actions (capture screenshots for each)

- Approve the PR → pr_approved.png

@Anara-hayat has invited you to collaborate on the [Anara-hayat/lab2](#) repository

You can [accept or decline](#) this invitation. You can also visit [@Anara-hayat](#) to learn a bit more about them.

This invitation will expire in 7 days.

[View invitation](#)

Note: This invitation was intended for [anarahayat555@gmail.com](#). If you were not expecting this invitation, you can ignore this email. If @Anara-hayat is sending you too many emails, you can [block them](#) or [report abuse](#).

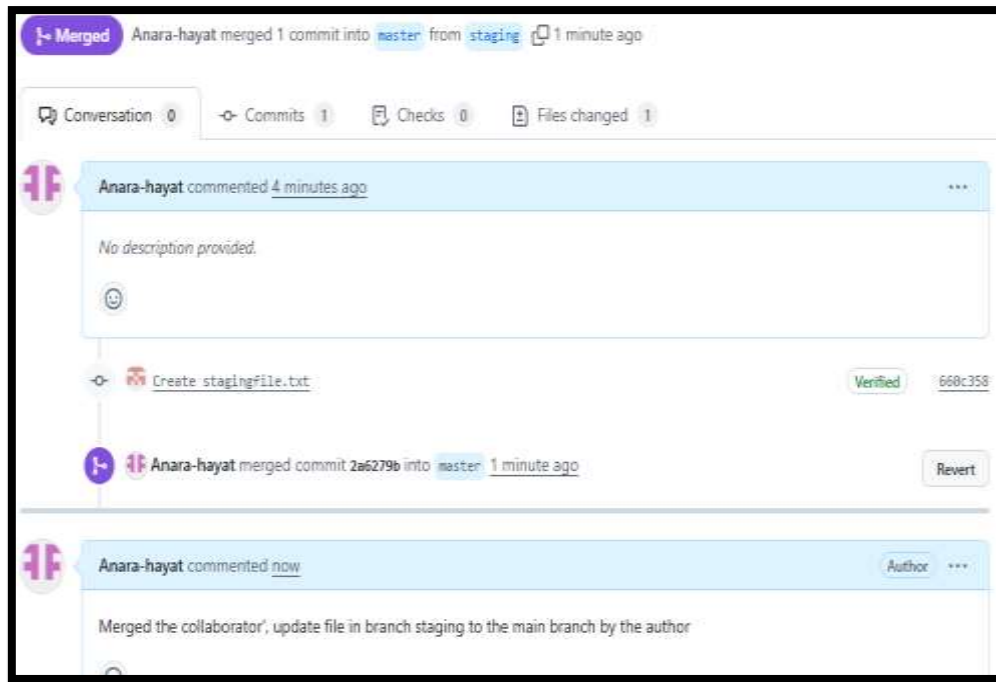
4. Request changes and add comments



Author Actions:

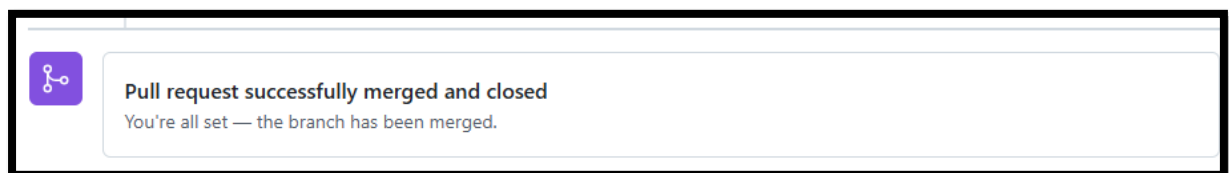
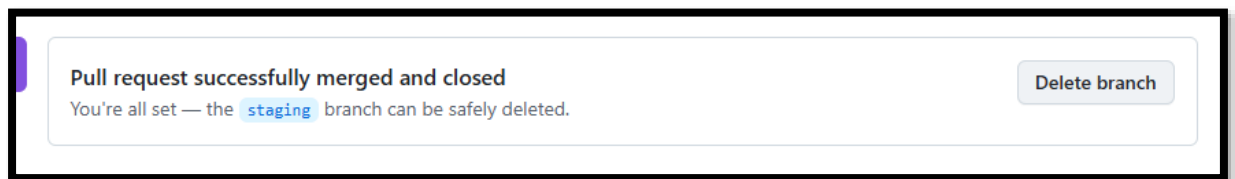


5. Merge the PR:



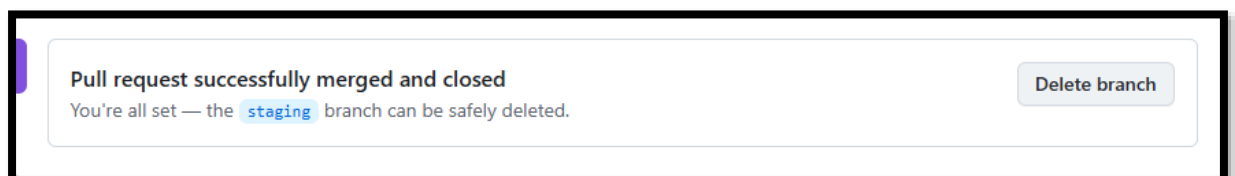
6. Branch Handling After Merge:

- If deleted, screenshot



Task 13: Branch Cleanup Best Practices

1. Delete Remote Branch After Merge:



2. Update Local Repository:

```
anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git branch
  bigfix/bugfix-new
  bugfix/user-auth-error
  feature-1
  feature/db-connection
  feature/feature-new
* master
  staging

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git branch -d staging
Deleted branch staging (was 781fb86).

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git branch -d bigfix/bugfix-new
error: branch 'bigfix/bugfix-new' not found

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git branch -d bigfix/bugfix-new
Deleted branch bigfix/bugfix-new (was 781fb86).

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git branch -d feature/db-connection
Deleted branch feature/db-connection (was 852a6af).

anara@DESKTOP-FCGAHUV MINGW64 ~/lab2 (master)
$ git branch
  bugfix/user-auth-error
  feature-1
  feature/feature-new
* master
```
