FATIMA JINNAH WOMEN UNIVERSITY
RAWALPINDI

Department of Software Engineering

SUBJECT: CLOUD COMPUTING LAB

Name: Saira Ejaz

Reg No: 2023-BSE-057

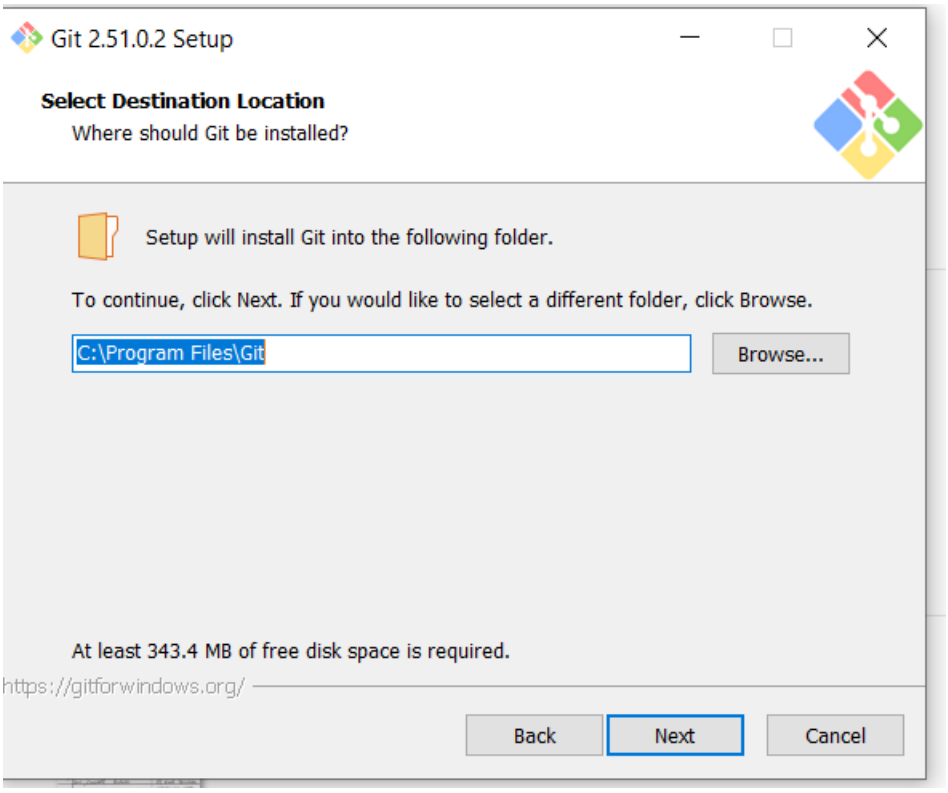
Section: 5-B

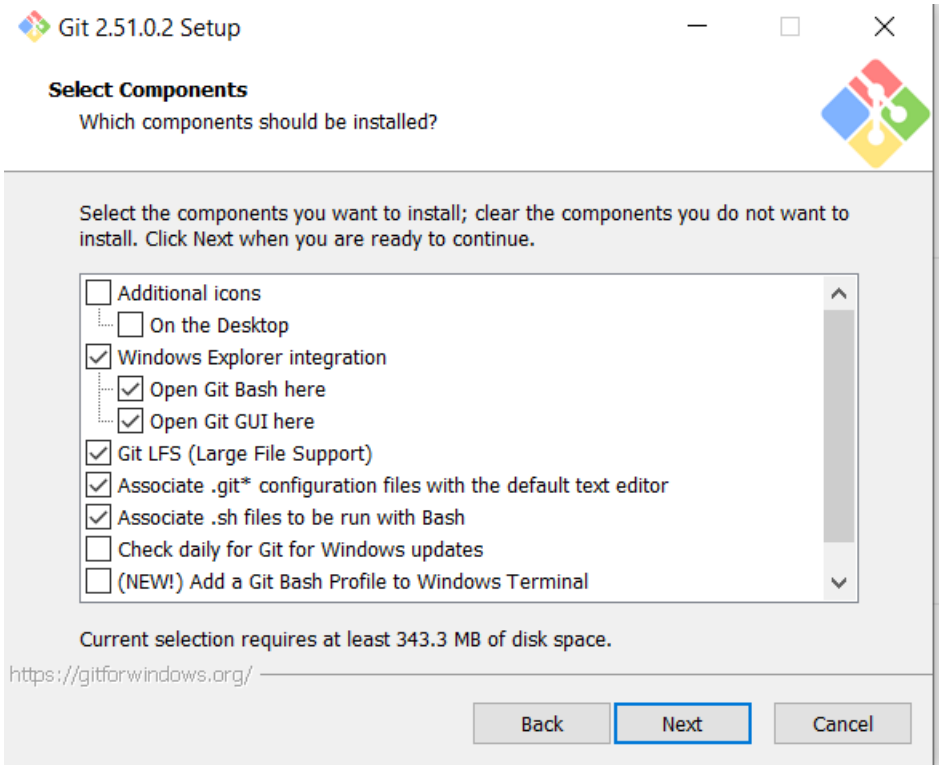
LAB NO 2:

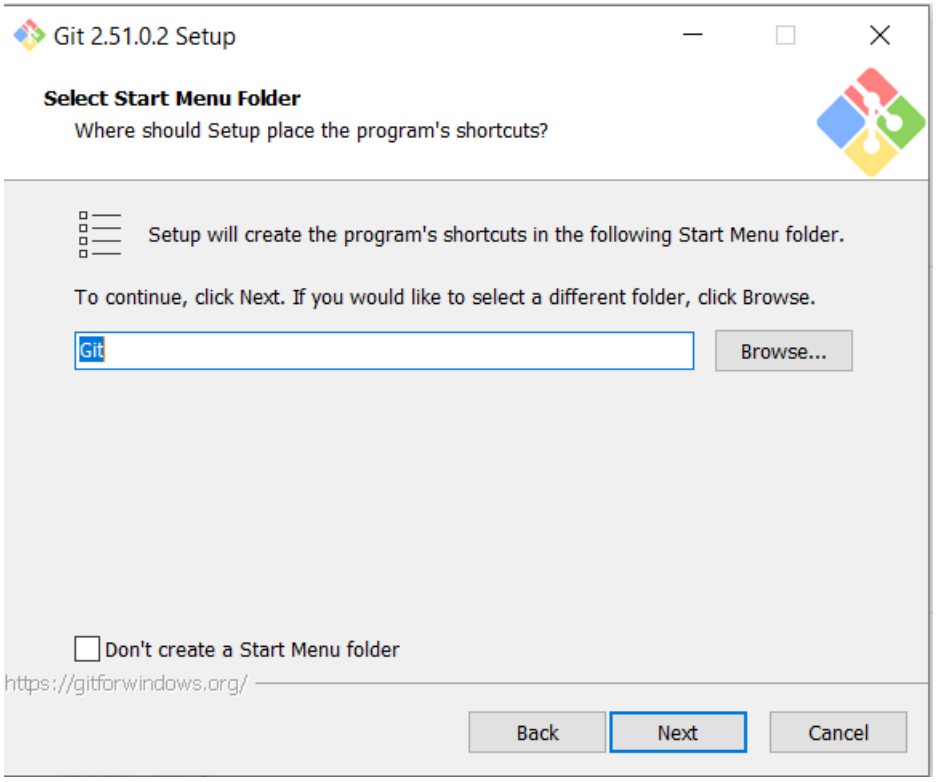
Git Installation

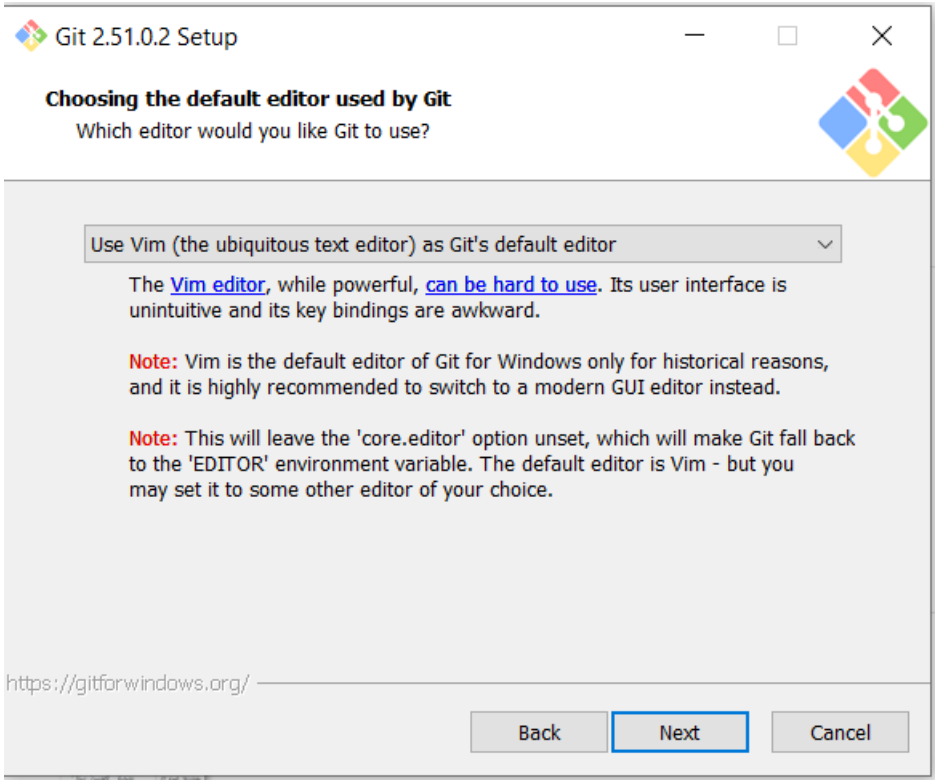
1. Install Git on your PC.

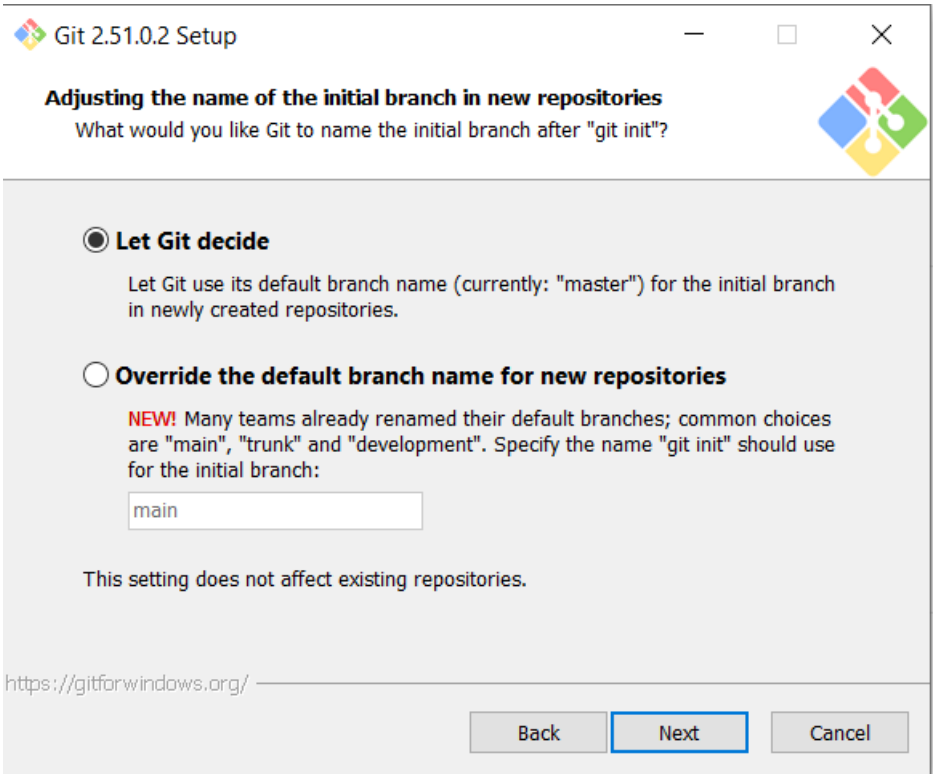
- Download Git for Windows, Mac, or Linux

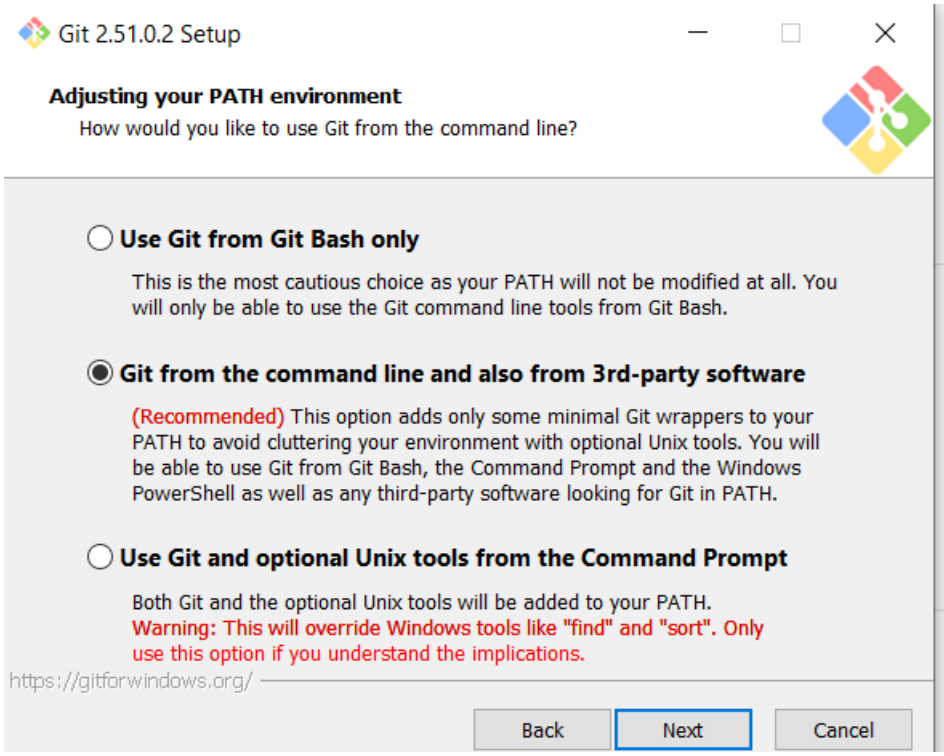


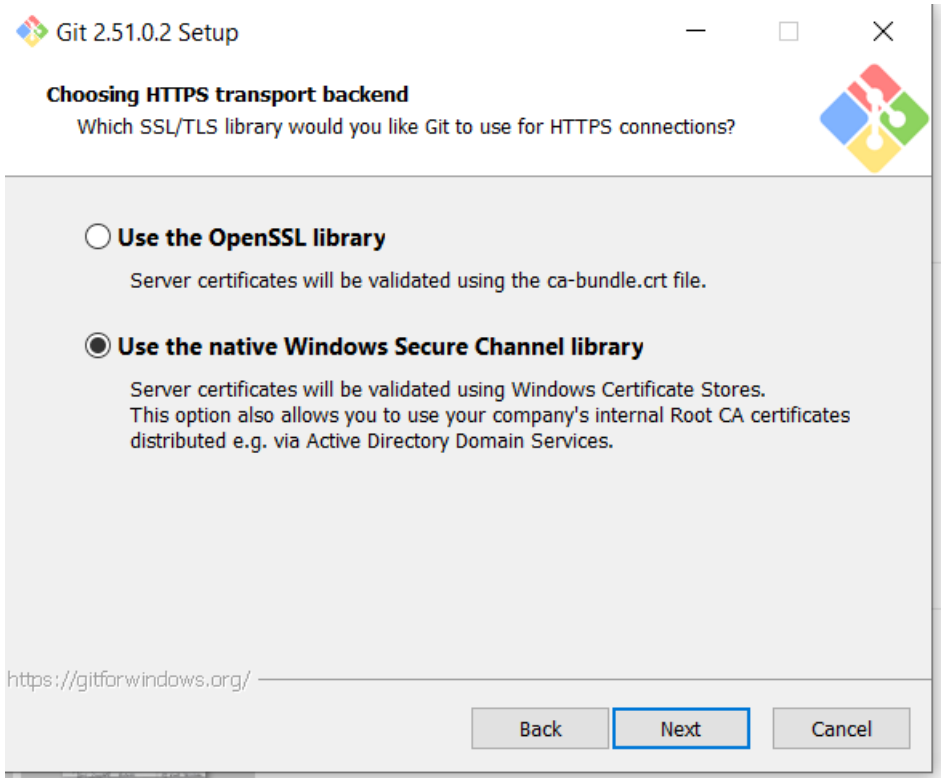

Git 2.51.0.2 Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Open Git Bash here
Open Git GUI here
Git LFS (Large File Support)
Associate .git* configuration files with the default text editor
Associate .sh files to be run with Bash
Check daily for Git for Windows updates
(NEW!) Add a Git Bash Profile to Windows Terminal
Current selection requires at least 343.3 MB of disk space.
https://gitforwindows.org/
Back
Next
Cancel

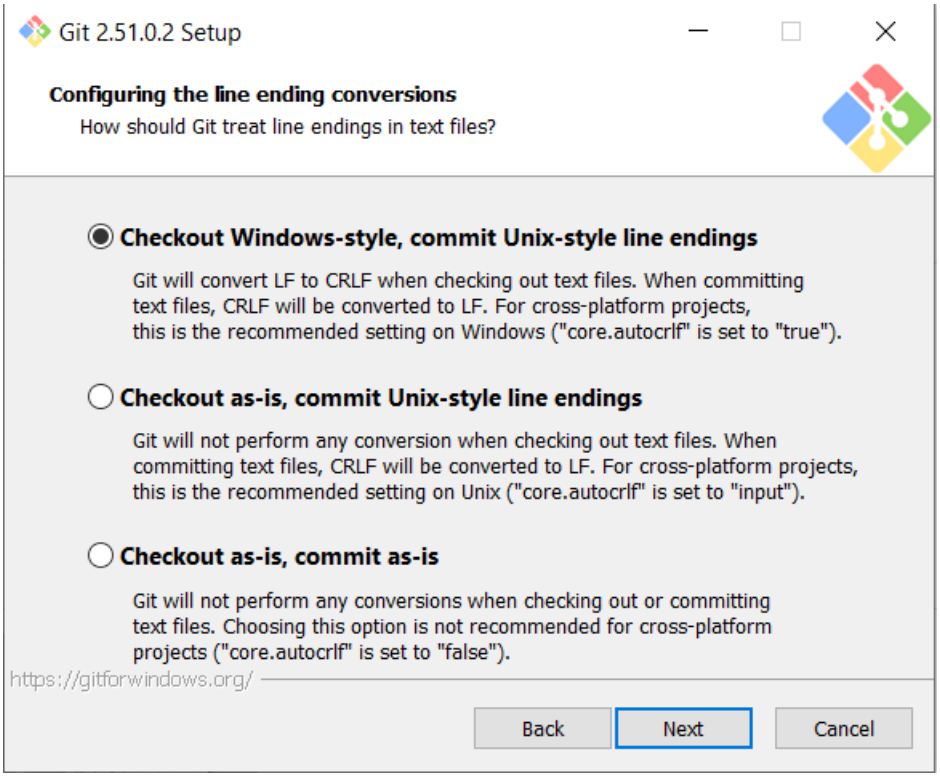

Git 2.51.0.2 Setup
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
Git
Browse...
Don't create a Start Menu folder
https://gitforwindows.org/
Back
Next
Cancel

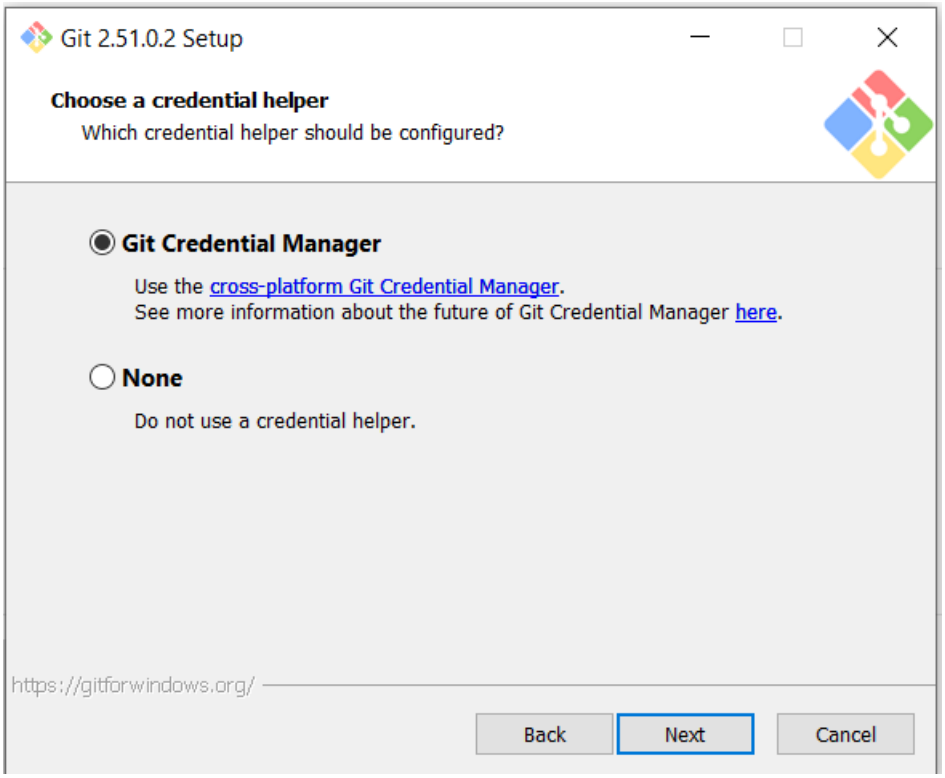

Git 2.51.0.2 Setup
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.
Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.
https://gitforwindows.org/
Back
Next
Cancel

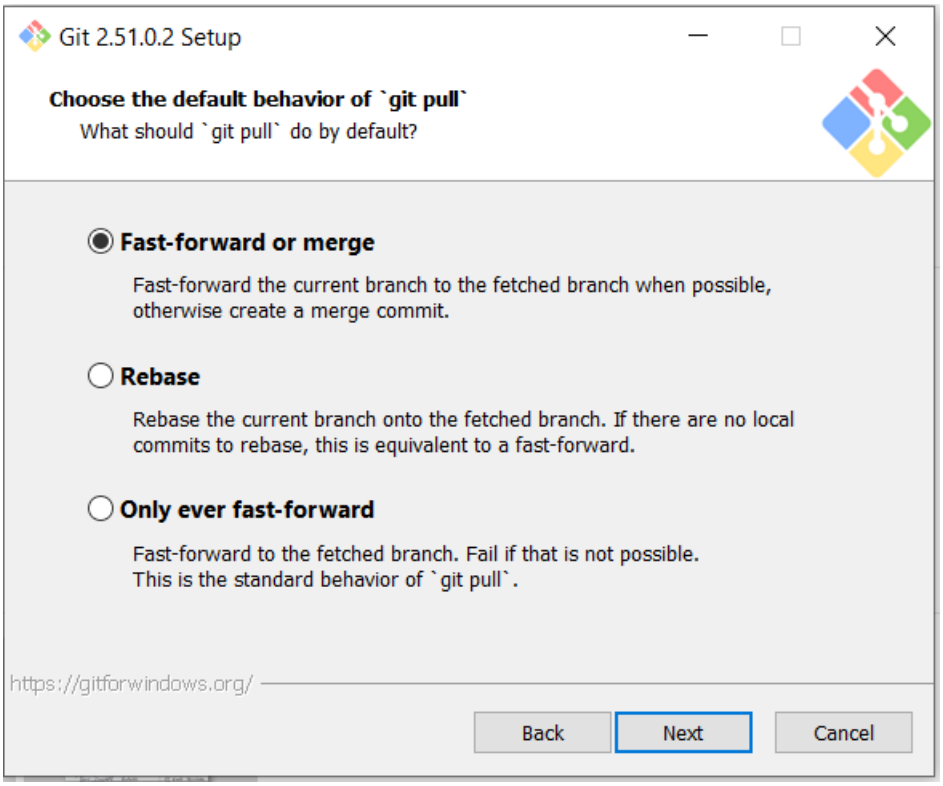

Git 2.51.0.2 Setup
Adjusting the name of the initial branch in new repositories
What would you like Git to name the initial branch after "git init"?
Let Git decide
Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories.
Override the default branch name for new repositories
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:
main
This setting does not affect existing repositories.
https://gitforwindows.org/
Back
Next
Cancel

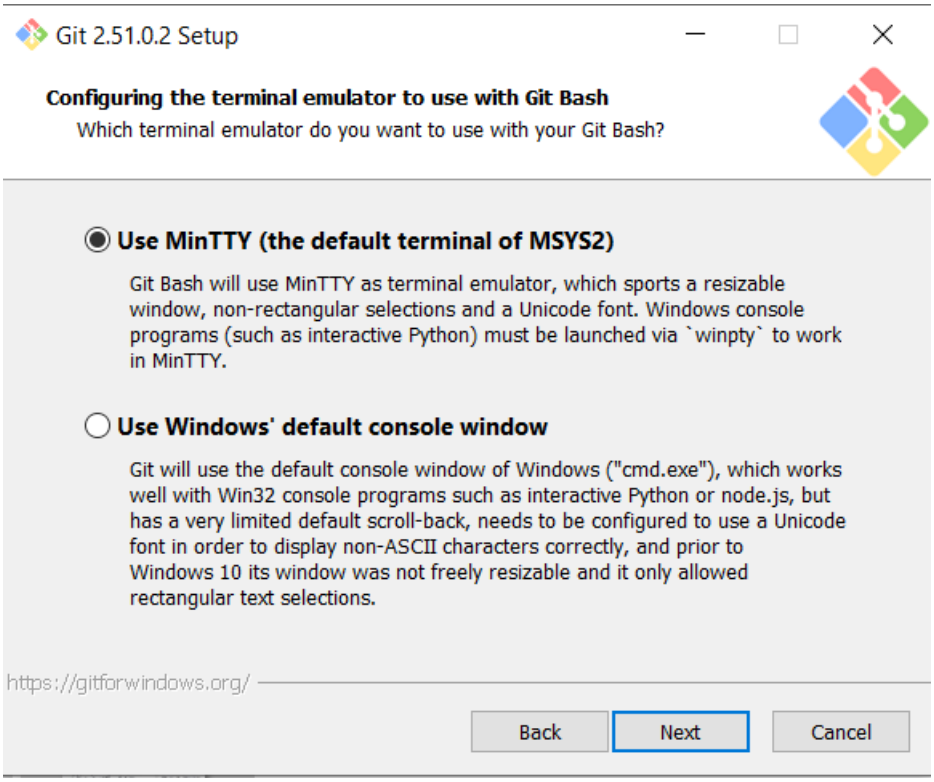

Git 2.51.0.2 Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Git from the command line and also from 3rd-party software
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.
Use Git and optional Unix tools from the Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
https://gitforwindows.org/
Back
Next
Cancel

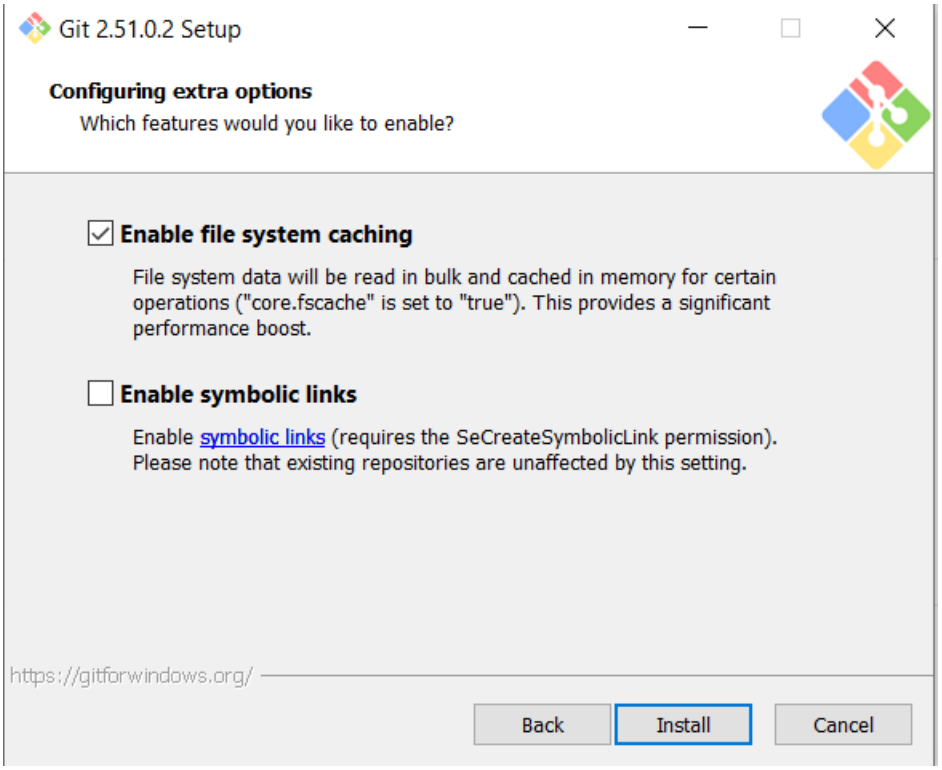

Git 2.51.0.2 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.
https://gitforwindows.org/
Back
Next
Cancel

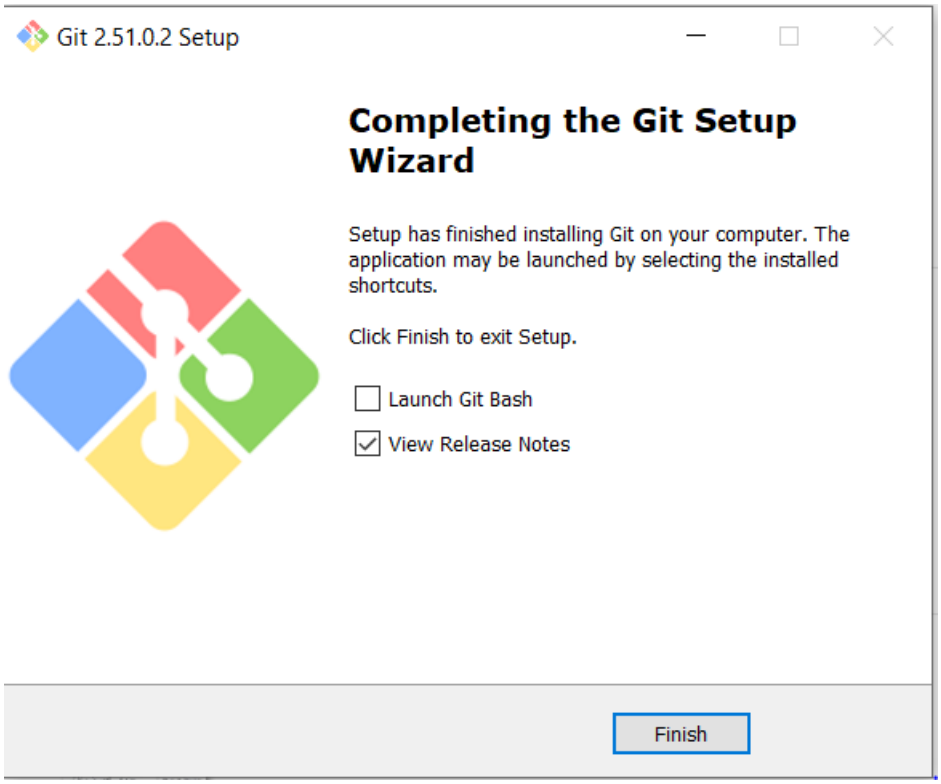

Git 2.51.0.2 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
Back
Next
Cancel


Git 2.51.0.2 Setup
Choose a credential helper
Which credential helper should be configured?
Git Credential Manager
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.
None
Do not use a credential helper.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.51.0.2 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Fast-forward or merge
Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible. This is the standard behavior of `git pull`.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.51.0.2 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://gitforwindows.org/
Back
Next
Cancel

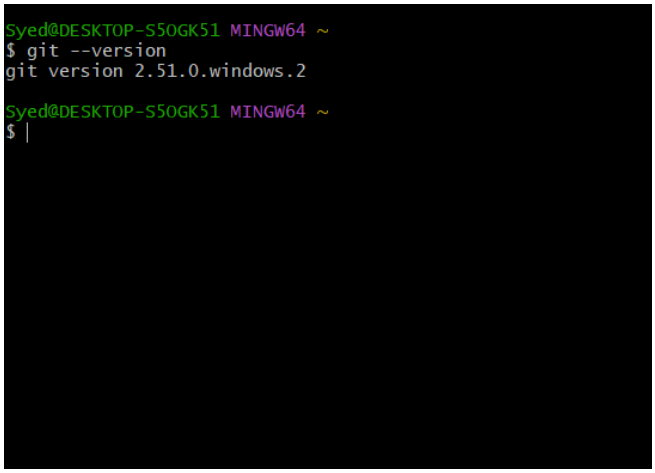


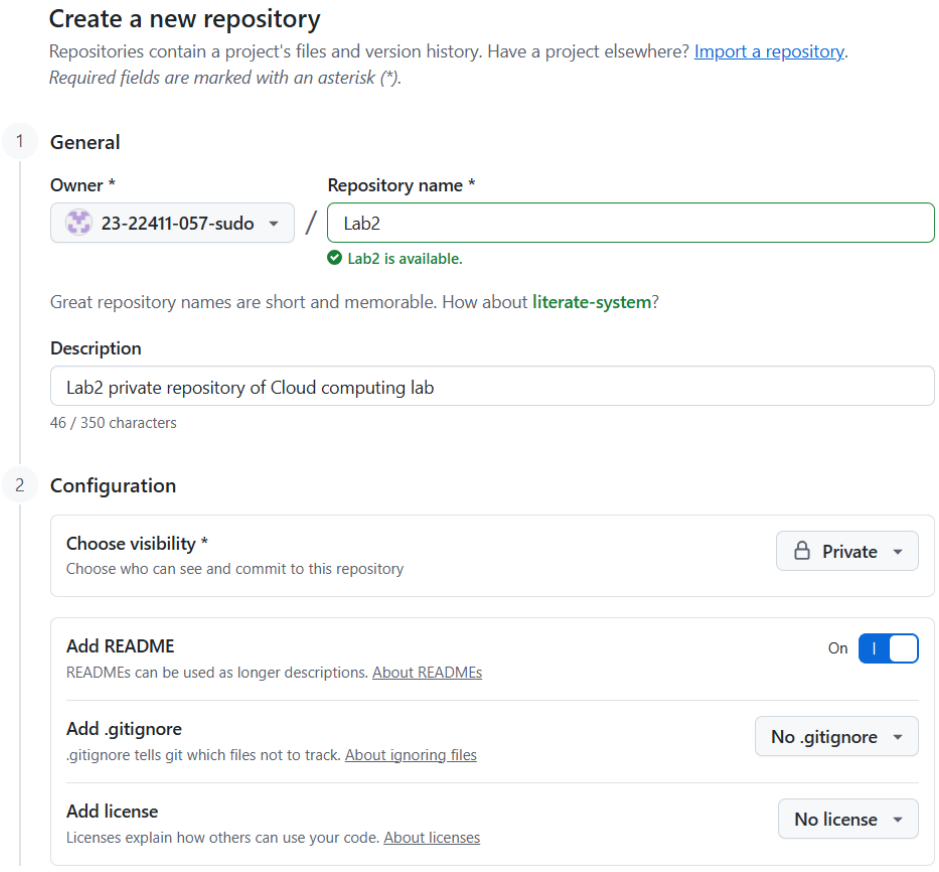


➢ Task 1:

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.

Danger Zone

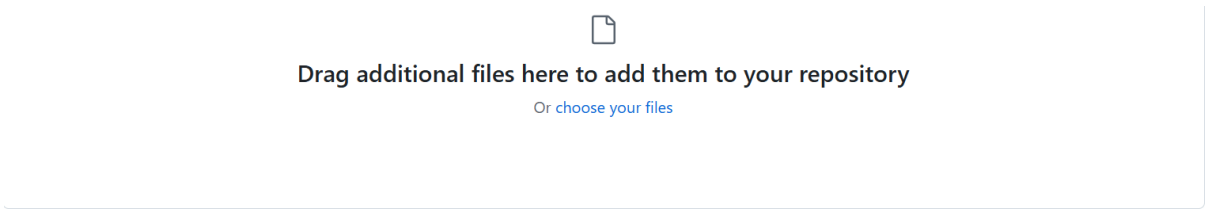
Change repository visibility
This repository is currently private. — Change visibility

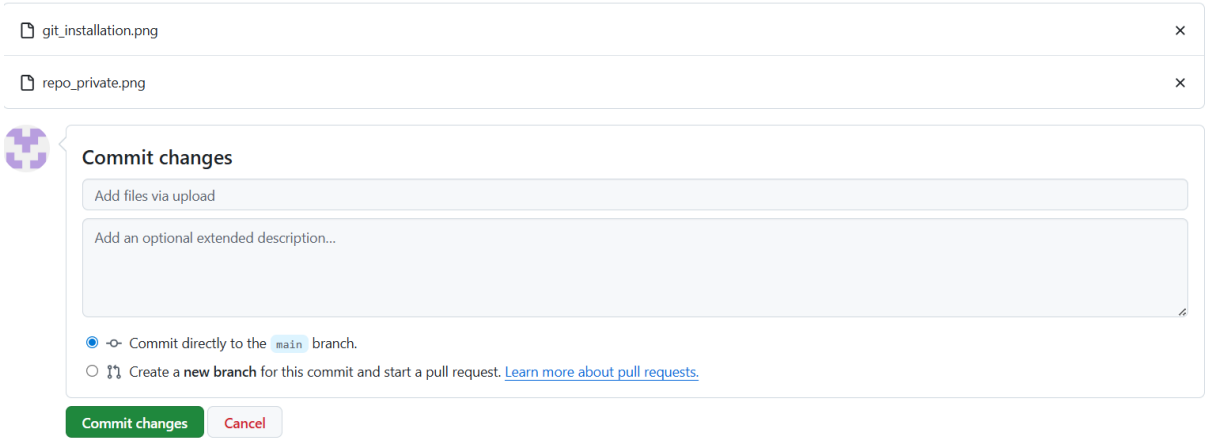
Disable branch protection rules
Disable branch protection rules enforcement and APIs — Disable branch protection rules

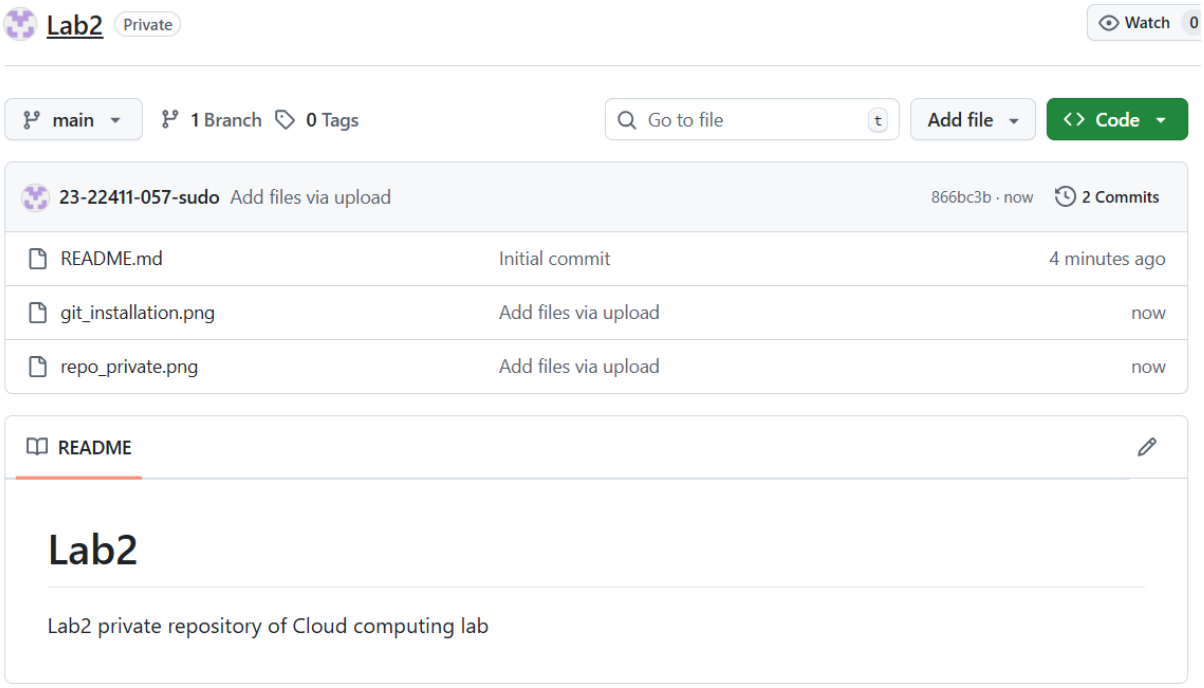
Transfer ownership
Transfer this repository to another user or to an organization where you have the ability to create repositories. — Transfer

Archive this repository
Mark this repository as archived and read-only. — Archive this repository

Delete this repository
Once you delete a repository, there is no going back. Please be certain. — Delete this repository



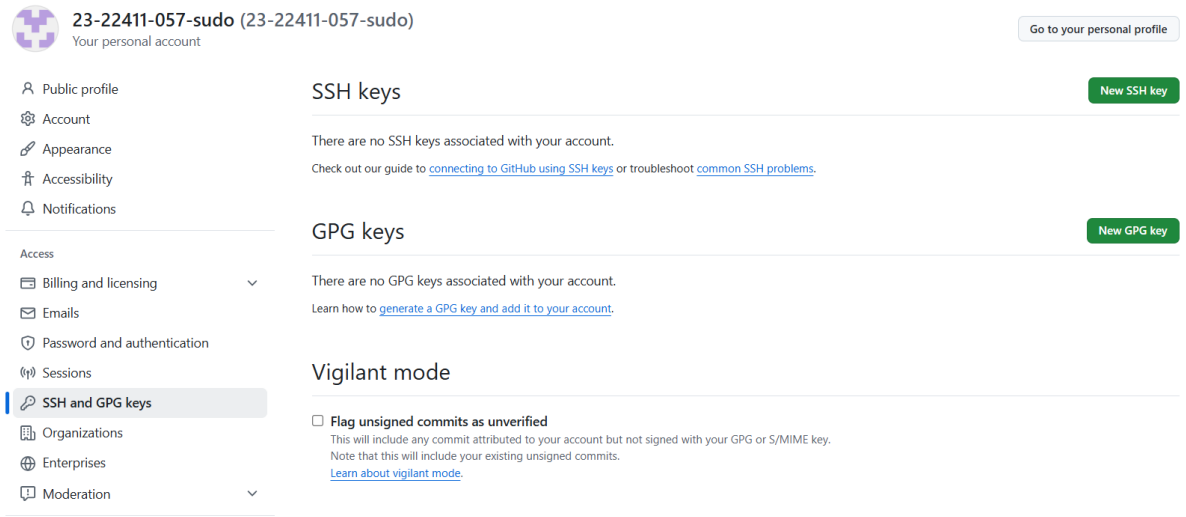


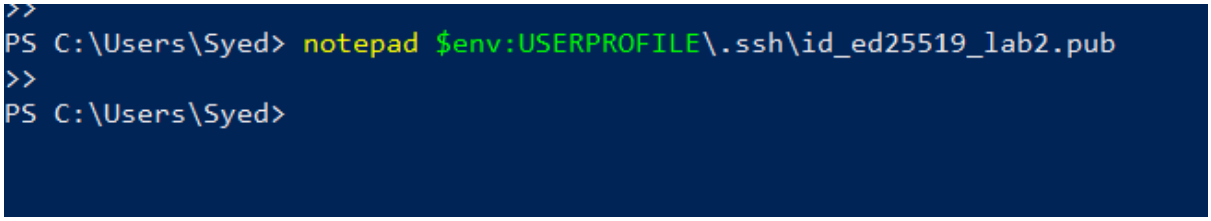


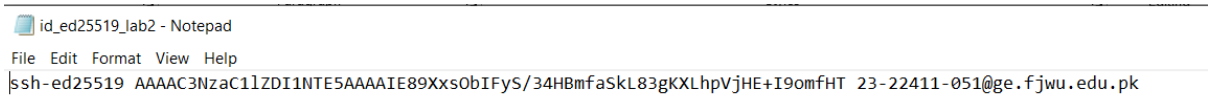
➢ Task 2:

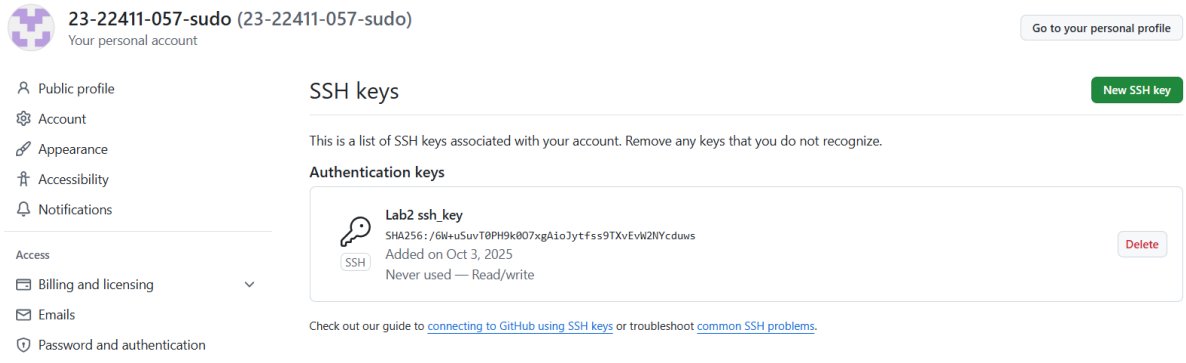
Connect Repository via SSH

```
PS C:\Users\Syed> ssh-keygen -t ed25519 -C "23-22411-051@ge.fjwu.edu.pk"
>>
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\Syed/.ssh/id_ed25519):
C:\Users\Syed/.ssh/id_ed25519 already exists.
Overwrite (y/n)?
PS C:\Users\Syed> ssh-keygen -t ed25519 -C "23-22411-051@ge.fjwu.edu.pk" -f "$env:USERPROFILE\.ssh\id_ed25519_lab2"
>>
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\Syed\.ssh\id_ed25519_lab2
Your public key has been saved in C:\Users\Syed\.ssh\id_ed25519_lab2.pub
The key fingerprint is:
SHA256:/6W+uSuvT0PH9k0O7xgAioJytfss9TXvEvW2NYcduws 23-22411-051@ge.fjwu.edu.pk
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|                 |
|    .     .      |
|   o . . . ... . |
|. o o . S  .o.=o+|
| o   o.  .+. ++O=|
|    .. . ..+oEooO|
|    .o  . oo.=o= |
|     .o   .O%+..o|
+----[SHA256]-----+
PS C:\Users\Syed> notepad $env:USERPROFILE\.ssh\id_ed25519_lab2.pub
>>
```


23-22411-057-sudo (23-22411-057-sudo)
Your personal account
Go to your personal profile
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and licensing
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation
SSH keys
New SSH key
There are no SSH keys associated with your account.
Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.
GPG keys
New GPG key
There are no GPG keys associated with your account.
Learn how to generate a GPG key and add it to your account.
Vigilant mode
Flag unsigned commits as unverified
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
Learn about vigilant mode.


>>
PS C:\Users\Syed> notepad $env:USERPROFILE\.ssh\id_ed25519_lab2.pub
>>
PS C:\Users\Syed>

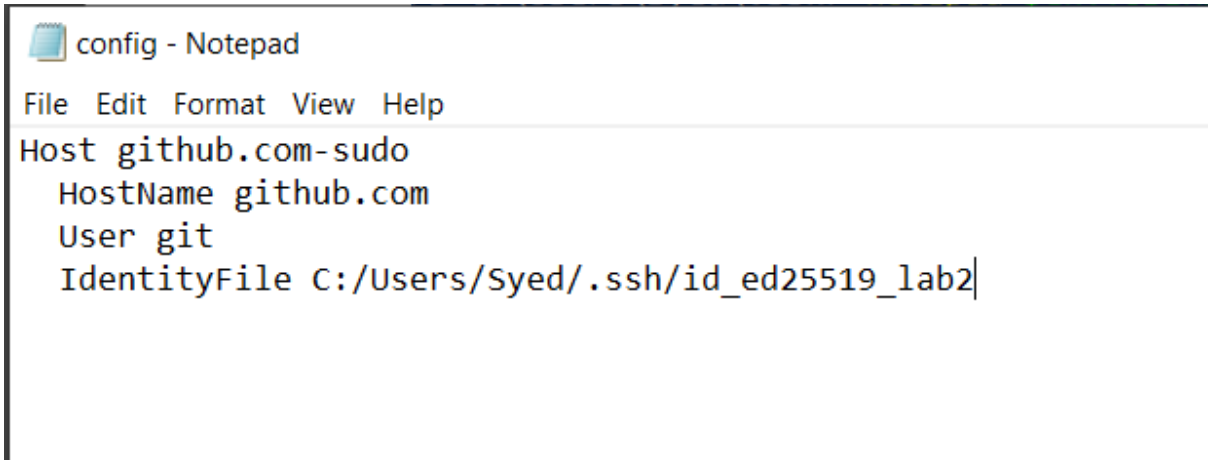

id_ed25519_lab2 - Notepad
File Edit Format View Help
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIE89XxsObIFyS/34HBmfaSkL83gKXLhpVjHE+I9omfHT 23-22411-051@ge.fjwu.edu.pk

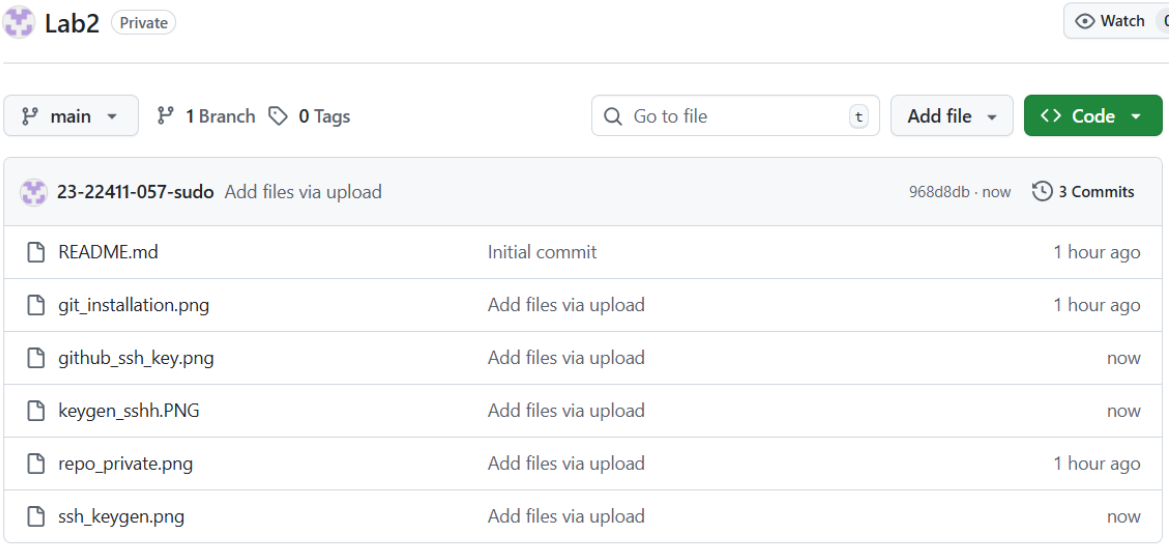

23-22411-057-sudo (23-22411-057-sudo)
Your personal account
Go to your personal profile
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and licensing
Emails
Password and authentication
SSH keys
New SSH key
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.
Authentication keys
Lab2 ssh_key
SHA256:/6W+u5uvT0PH9k0O7xgAioJytfss9TXvEvW2NYcduws
SSH
Added on Oct 3, 2025
Never used — Read/write
Delete
Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

```
PS C:\Users\Syed> cd documents
PS C:\Users\Syed\documents> git clone git@github.com-lab2:yourusername/Lab2.git
>>
Cloning into 'Lab2'...
ssh: Could not resolve hostname github.com-lab2: No such host is known.
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
PS C:\Users\Syed\documents> Get-ChildItem $env:USERPROFILE\.ssh\config*
>>


    Directory: C:\Users\Syed\.ssh


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         10/3/2025   6:14 PM            114 config.txt
-a----         10/3/2025   6:13 PM            102 config1.txt
-a----         10/3/2025   6:14 PM            114 config2.txt


PS C:\Users\Syed\documents> Rename-Item "$env:USERPROFILE\.ssh\config.txt" "$env:USERPROFILE\.ssh\config"
>>
PS C:\Users\Syed\documents> notepad $env:USERPROFILE\.ssh\config
>>
PS C:\Users\Syed\documents> ssh -vT git@github.com-lab2
>>
OpenSSH_for_Windows_9.5p1, LibreSSL 3.8.2
debug1: Reading configuration data C:\\Users\\Syed/.ssh/config
debug1: C:\\Users\\Syed/.ssh/config line 2: Applying options for github.com-lab2
debug1: Connecting to github.com [20.207.73.82] port 22.
debug1: Connection established.
debug1: identity file C:/Users/Syed/.ssh/id_ed25519_lab2 type 3
debug1: identity file C:/Users/Syed/.ssh/id_ed25519_lab2-cert type -1
debug1: Local version string SSH-2.0-OpenSSH_for_Windows_9.5
debug1: Remote protocol version 2.0, remote software version 550beff
debug1: compat_banner: no match: 550beff
debug1: Authenticating to github.com:22 as 'git'
debug1: load_hostkeys: fopen C:\\Users\\Syed/.ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen __PROGRAMDATA__\\ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen __PROGRAMDATA__\\ssh/ssh_known_hosts2: No such file or directory
debug1: SSH2_MSG_KEXINIT sent
```

```
debug1: Found key in C:\\Users\\Syed/.ssh/known_hosts:4
debug1: ssh_packet_send2_wrapped: resetting send seqnr 3
debug1: rekey out after 134217728 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: ssh_packet_read_poll2: resetting read seqnr 3
debug1: SSH2_MSG_NEWKEYS received
debug1: rekey in after 134217728 blocks
debug1: get_agent_identities: ssh_get_authentication_socket: No such file or directory
debug1: Will attempt key: C:/Users/Syed/.ssh/id_ed25519_lab2 ED25519 SHA256:/6W+uSuvT0PH9k0O7xgAioJytfss9TXvEvW2NYcduws explicit
debug1: SSH2_MSG_EXT_INFO received
debug1: kex_input_ext_info: server-sig-algs=<ssh-ed25519-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ecdsa-sha2-nistp384
5519-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256-cert-v01@openssh.com,rsa-sha2-512-cert-v01@openssh.com,rsa-sha2-256-cert-v01@openssh.com,
tp256@openssh.com,ssh-ed25519,ecdsa-sha2-nistp521,ecdsa-sha2-nistp384,ecdsa-sha2-nistp256,rsa-sha2-512,rsa-sha2-256,ssh-rsa>
debug1: kex_ext_info_check_ver: publickey-hostbound@openssh.com=<0>
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey
debug1: Next authentication method: publickey
debug1: Offering public key: C:/Users/Syed/.ssh/id_ed25519_lab2 ED25519 SHA256:/6W+uSuvT0PH9k0O7xgAioJytfss9TXvEvW2NYcduws explicit
debug1: Server accepts key: C:/Users/Syed/.ssh/id_ed25519_lab2 ED25519 SHA256:/6W+uSuvT0PH9k0O7xgAioJytfss9TXvEvW2NYcduws explicit
Authenticated to github.com ([20.207.73.82]:22) using "publickey".
debug1: channel 0: new session [client-session] (inactive timeout: 0)
debug1: Entering interactive session.
debug1: pledge: filesystem
debug1: client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
debug1: client_input_hostkeys: searching C:\\Users\\Syed/.ssh/known_hosts for github.com / (none)
debug1: client_input_hostkeys: searching C:\\Users\\Syed/.ssh/known_hosts2 for github.com / (none)
debug1: client_input_hostkeys: hostkeys file C:\\Users\\Syed/.ssh/known_hosts2 does not exist
debug1: client_input_hostkeys: no new or deprecated keys from server
debug1: pledge: fork
debug1: client_input_channel_req: channel 0 rtype exit-status reply 0
Hi 23-22411-057-sudo! You've successfully authenticated, but GitHub does not provide shell access.
debug1: channel 0: free: client-session, nchannels 1
Transferred: sent 2112, received 2704 bytes, in 0.8 seconds
Bytes per second: sent 2511.3, received 3215.2
debug1: Exit status 1
PS C:\Users\Syed\documents> ssh -T git@github.com-lab2
>>
Hi 23-22411-057-sudo! You've successfully authenticated, but GitHub does not provide shell access.
PS C:\Users\Syed\documents>
```

```
PS C:\Users\Syed> cd $env:USERPROFILE\.ssh
PS C:\Users\Syed\.ssh> ssh -T git@github,com-lab2
ssh: Could not resolve hostname github,com-lab2: No such host is known.
PS C:\Users\Syed\.ssh> ssh -T git@github,com
ssh: Could not resolve hostname github,com: No such host is known.
PS C:\Users\Syed\.ssh> notepad $env:USERPROFILE\.ssh\config
>>
PS C:\Users\Syed\.ssh> ssh -T git@github.com-sudo
>>
Hi 23-22411-057-sudo! You've successfully authenticated, but GitHub does not provide shell access.
PS C:\Users\Syed\.ssh> git clone git@github.com-sudo:23-22411-057-sudo/Lab2.git
>>
Cloning into 'Lab2'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), 45.20 KiB | 110.00 KiB/s, done.
PS C:\Users\Syed\.ssh>
```





➢ Task 3:

Configure Git Username and Email

```
PS C:\Users\Syed\documents> git config --global user.name "Saira Ejaz"
PS C:\Users\Syed\documents> git config --global user.email "23-22411-057@se.fjwu.edu.pk"
PS C:\Users\Syed\documents> get config --list
get : The term 'get' is not recognized as the name of a cmdlet, function, script file, or operable prc
again.
At line:1 char:1
+ get config --list
+ ~~~
    + CategoryInfo          : ObjectNotFound: (get:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
```

```
PS C:\Users\Syed\documents> 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
gui.recentrepo=C:/Users/Syed/xxxxxx
gui.recentrepo=C:/Users/Syed/Lab2
gui.recentrepo=C:/Users/Syed/Pictures/lab 2
user.email=23-22411-057@se.fjwu.edu.pk
user.name=Saira Ejaz
core.sshcommand=C:/Windows/System32/OpenSSH/ssh.exe
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=git@github.com:23-22411-061-rgb/lab-2.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/master
gui.wmstate=normal
gui.geometry=1109x563+32+32 216 255
PS C:\Users\Syed\documents>
```

```
PS C:\Users\Syed\Documents\lab-2> Move-Item "git_installation.png" "$env:USERPROFILE\Documents\lab-2"
>> Move-Item "repo_private.png" "$env:USERPROFILE\Documents\lab-2"
>> Move-Item "ssh_keygen.png" "$env:USERPROFILE\Documents\lab-2"
>> Move-Item "github_sshkey.png" "$env:USERPROFILE\Documents\lab-2"
>> Move-Item "ssh_clone.png" "$env:USERPROFILE\Documents\lab-2"
>> Move-Item "git_identity.png" "$env:USERPROFILE\Documents\lab-2"
>> Move-Item "git_config_list.png" "$env:USERPROFILE\Documents\lab-2"
>>
PS C:\Users\Syed\Documents\lab-2> cd $env:USERPROFILE\Documents\lab-2
>>
PS C:\Users\Syed\Documents\lab-2> git add *.png
>>
PS C:\Users\Syed\Documents\lab-2> git commit -m "Added all lab screenshots"
>>
[main 80c989d] Added all lab screenshots
 7 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 git_config_list.png
 create mode 100644 git_identity.png
 create mode 100644 git_installation.png
 create mode 100644 github_sshkey.png
 create mode 100644 repo_private.png
 create mode 100644 ssh_clone.png
 create mode 100644 ssh_keygen.png
```

```
>>
9c3a1b2 (HEAD -> main, origin/main, origin/HEAD) Added all lab screenshots
6ddbb7e Added all lab screenshots
0ba4493 Add files via upload
d806aa2 Add files via upload
956ef3e Delete ssh_keygen.png
87a2995 Delete ssh_clone.png
46d68ac Delete repo_private.png.PNG
ed009f8 Delete github_sshkey.png
034d803 Delete git_installation.png
75b4d48 Delete git_identity.png..PNG
dfa591f Delete git_config_list.png..PNG
7b2b1fa Add files via upload
a873e46 Add files via upload
3baf9bd Add files via upload
PS C:\Users\Syed\Documents\lab-2> git status
>>
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\Syed\Documents\lab-2> cd $env:USERPROFILE\Documents\lab-2
>> git add git_identity.png git_config_list.png
>> git commit -m "Added Git identity and config screenshots"
>> git push origin main
>>
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
```

```
                        Get-ChildItem "Lab2" -Filter *.png
>> C:\Users\Syed\Documents>


    Directory: C:\Users\Syed\Documents\Lab2


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         10/3/2025   6:47 PM          13844 clone_ssh.PNG
-a----         10/3/2025   8:08 PM          66614 github_ssh_key.png
-a----         10/3/2025   8:08 PM          50735 git_configlist.PNG
-a----         10/3/2025   8:08 PM          12733 git_Identit.PNG
-a----         10/3/2025   6:54 PM           8777 git_installation.png
-a----         10/3/2025   8:08 PM          24984 keygen_sshh.PNG
-a----         10/3/2025   6:54 PM          46191 repo_private.png
-a----         10/3/2025   8:08 PM          44502 ssh_keygen.png


PS C:\Users\Syed\Documents>
```

➢ Task 4:

Explore the .git Folder

```
PS C:\Users\Syed\Documents> cd Lab2
PS C:\Users\Syed\Documents\Lab2> cd $env:USERPROFILE\Documents\Lab2
PS C:\Users\Syed\Documents\Lab2> ls -Force


    Directory: C:\Users\Syed\Documents\Lab2


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--h--         10/3/2025   8:09 PM                .git
-a----         10/3/2025   6:47 PM          13844 clone_ssh.PNG
-a----         10/3/2025   8:08 PM          66614 github_ssh_key.png
-a----         10/3/2025   8:08 PM          50735 git_configlist.PNG
-a----         10/3/2025   8:08 PM          12733 git_Identit.PNG
-a----         10/3/2025   6:54 PM           8777 git_installation.png
-a----         10/3/2025   8:08 PM          24984 keygen_sshh.PNG
-a----         10/3/2025   6:54 PM             56 README.md
-a----         10/3/2025   6:54 PM          46191 repo_private.png
-a----         10/3/2025   8:08 PM          44502 ssh_keygen.png


PS C:\Users\Syed\Documents\Lab2>
```

```
PS C:\Users\Syed\Documents\Lab2> Get-ChildItem -Force .git


    Directory: C:\Users\Syed\Documents\Lab2\.git


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         10/3/2025   6:53 PM                hooks
d-----         10/3/2025   6:53 PM                info
d-----         10/3/2025   6:54 PM                logs
d-----         10/3/2025   8:09 PM                objects
d-----         10/3/2025   6:54 PM                refs
-a----         10/3/2025   8:07 PM             26 COMMIT_EDITMSG
-a----         10/3/2025   6:54 PM            305 config
-a----         10/3/2025   6:53 PM             73 description
-a----         10/3/2025   8:09 PM             98 FETCH_HEAD
-a----         10/3/2025   6:54 PM             21 HEAD
-a----         10/3/2025   8:08 PM            721 index
-a----         10/3/2025   8:09 PM             41 ORIG_HEAD
-a----         10/3/2025   6:54 PM            112 packed-refs


PS C:\Users\Syed\Documents\Lab2> ls -Force .git


    Directory: C:\Users\Syed\Documents\Lab2\.git


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         10/3/2025   6:53 PM                hooks
d-----         10/3/2025   6:53 PM                info
d-----         10/3/2025   6:54 PM                logs
d-----         10/3/2025   8:09 PM                objects
d-----         10/3/2025   6:54 PM                refs
-a----         10/3/2025   8:07 PM             26 COMMIT_EDITMSG
-a----         10/3/2025   6:54 PM            305 config
-a----         10/3/2025   6:53 PM             73 description
-a----         10/3/2025   8:09 PM             98 FETCH_HEAD
-a----         10/3/2025   6:54 PM             21 HEAD
-a----         10/3/2025   8:08 PM            721 index
-a----         10/3/2025   8:09 PM             41 ORIG_HEAD
-a----         10/3/2025   6:54 PM            112 packed-refs
```

In Git Bash:

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Lab2 (master)
$ cd

Syed@DESKTOP-S5OGK51 MINGW64 ~ (main)
$ cd ~/Documents/Lab2

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ ls -a
./     README.md        git_configlist.PNG    keygen_sshh.PNG
../    clone_ssh.PNG    git_installation.png  repo_private.png
.git/  git_Identit.PNG  github_ssh_key.png    ssh_keygen.png

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ ls -a .git
./   COMMIT_EDITMSG  HEAD       config       hooks/  info/  objects/     refs/
../  FETCH_HEAD      ORIG_HEAD  description  index   logs/  packed-refs

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$
```

➢ Task 5:

Local Repository Management

1. Delete existing .git folder

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ cd ~/Documents/Lab2

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ cd

Syed@DESKTOP-S5OGK51 MINGW64 ~ (main)
$ rm -rf .git
```

2. Re-initialize Git

```
Syed@DESKTOP-S5OGK51 MINGW64 ~
$ git init
Initialized empty Git repository in C:/Users/Syed/.git/
```

3. Add README.md and commit

```
Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$ echo "# Lab2 Git Practice" > README.md

Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it

Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$ git commit -m "Initial commit"
[master (root-commit) 4d64cc5] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

4. Connect to GitHub repo

```
Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$ git remote add origin git@github.com:23-22411-057-sudo/Lab2.git

Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'github.com:23-22411-057-sudo/Lab2.git'
```

5. Push to GitHub

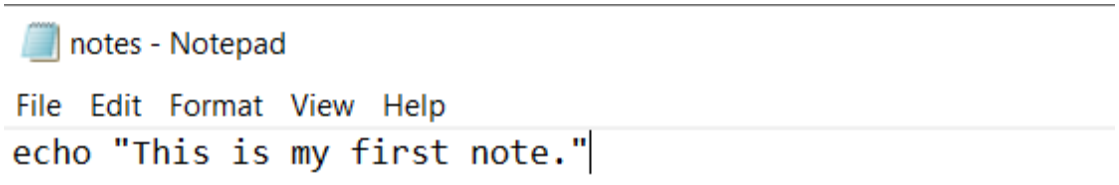
```
Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$ git push -u origin master
ERROR: Repository not found.
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$
```

➢ Task 6:

File Status & Staging

1. Create a new file

```
PS C:\Users\Syed\Documents\Lab2> notepad notes.txt
```



2. Check status

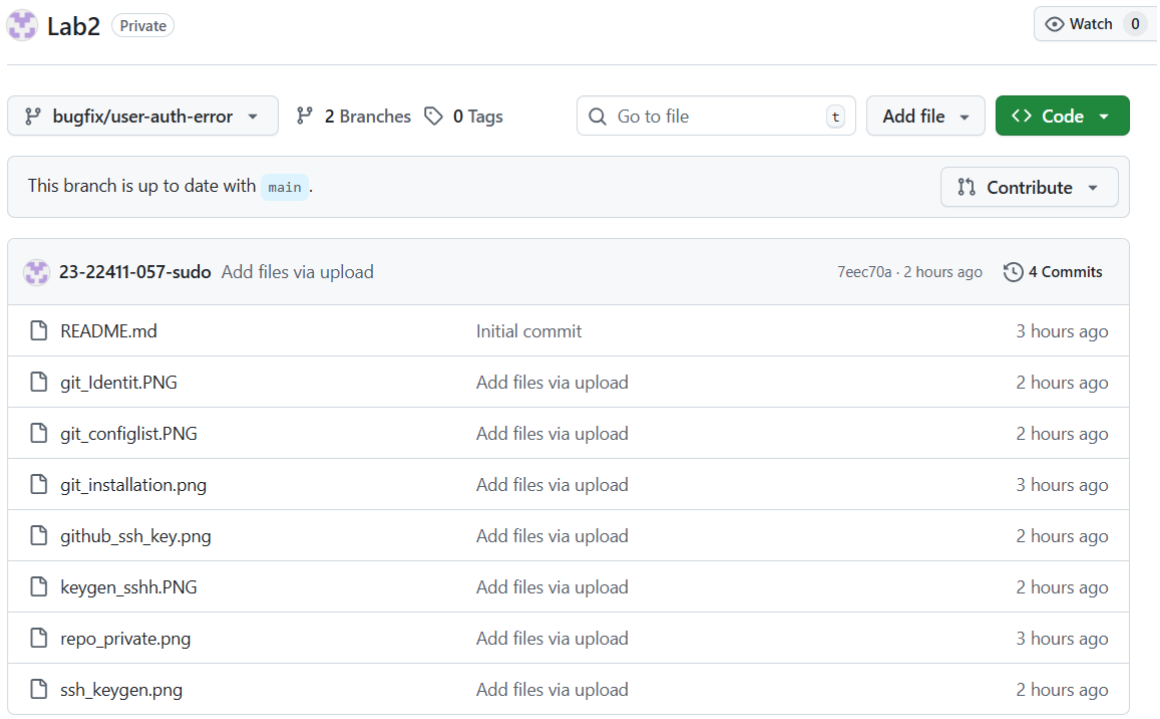
```
PS C:\Users\Syed\Documents\Lab2> git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

3. Stage and commit

```
PS C:\Users\Syed\Documents\Lab2> git add notes.txt
PS C:\Users\Syed\Documents\Lab2> git commit -m "Add notes.txt"
[main f3e8d8f] Add notes.txt
 1 file changed, 1 insertion(+)
 create mode 100644 notes.txt
PS C:\Users\Syed\Documents\Lab2>
```

➢ Task 7:

Branch Creation Using GitHub GUI



```
Syed@DESKTOP-S5OGK51 MINGW64 ~ (master)
$ cd ~/Documents/Lab2

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git pull origin bugfix/user-auth-error
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            bugfix/user-auth-error -> FETCH_HEAD
 * [new branch]      bugfix/user-auth-error -> origin/bugfix/user-auth-error
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git branch
* main

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$
```

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ 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'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (bugfix/user-auth-error)
$ git branch
* bugfix/user-auth-error
  main

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (bugfix/user-auth-error)
$
```

➢ 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

2. Push the branch to the remote repository:

git push origin feature/db-connection

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (bugfix/user-auth-error)
$ git checkout -b feature/db-connection
Switched to a new branch 'feature/db-connection'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/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/23-22411-057-sudo/Lab2/pull/new/feature/db-conne
ction
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      feature/db-connection -> feature/db-connection

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature/db-connection)
$
```

➢ Task 9:

Branching & Merging

1. Create and switch to a new branch
2. Modify `main.py` (add a function) and commit.
3. Switch back to `main` and merge:
4. Push all branches

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature/db-connection)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git checkout -b feature-1
Switched to a new branch 'feature-1'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git add main.py
fatal: pathspec 'main.py' did not match any files

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ ^C

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ ls
README.md        git_configlist.PNG    keygen_sshh.PNG   ssh_keygen.png
clone_ssh.PNG    git_installation.png  notes.txt
git_Identit.PNG  github_ssh_key.png    repo_private.png

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ echo "def hello():\n    print('Hello from feature-1')" > main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git add main.py
warning: in the working copy of 'main.py', LF will be replaced by CRLF the next
time Git touches it

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git commit -m "Add new function to main.py"
[main f160fab] Add new function to main.py
 1 file changed, 1 insertion(+)
 create mode 100644 main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 594 bytes | 84.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com-sudo:23-22411-057-sudo/Lab2.git
   7eec70a..f160fab  main -> main

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
```

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git push main
fatal: 'main' does not appear to be a git repository
fatal: Could not read from remote repository.
Merge branch 'feature-2'
```

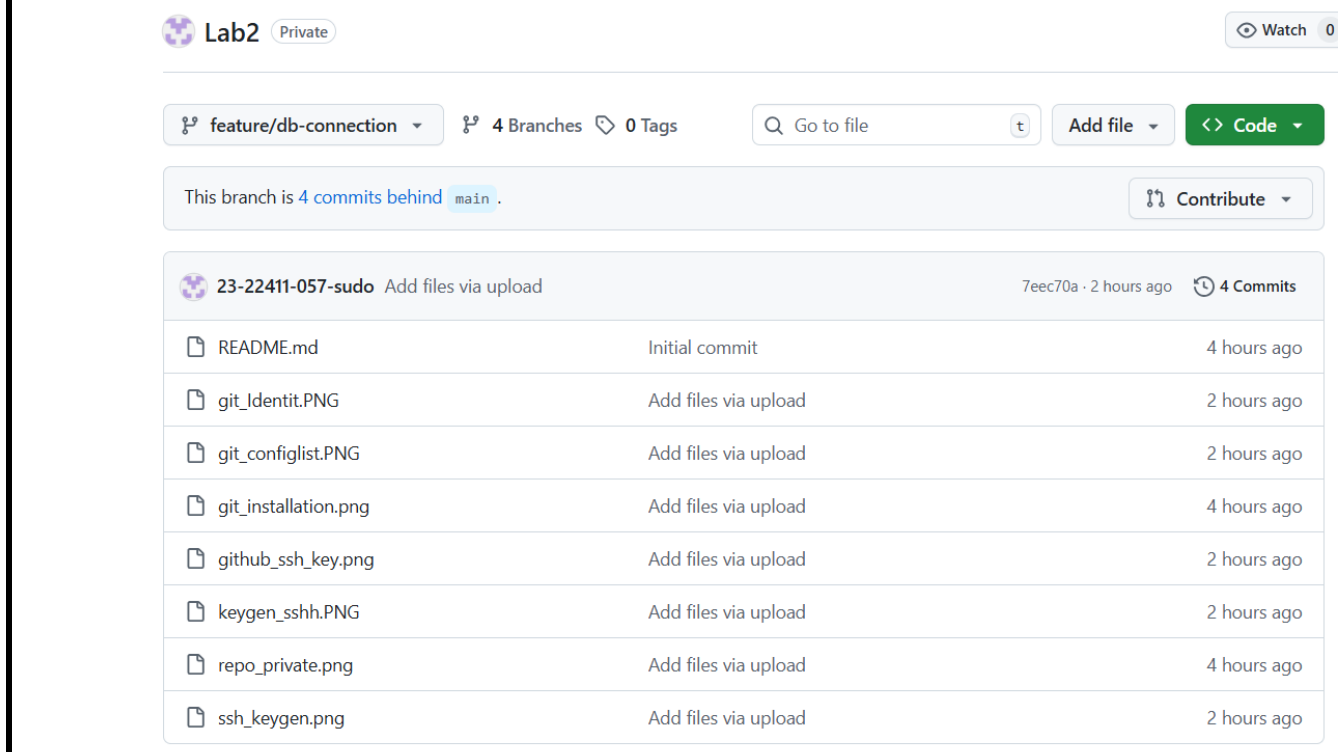
```
Merge made by the 'ort' strategy.
```

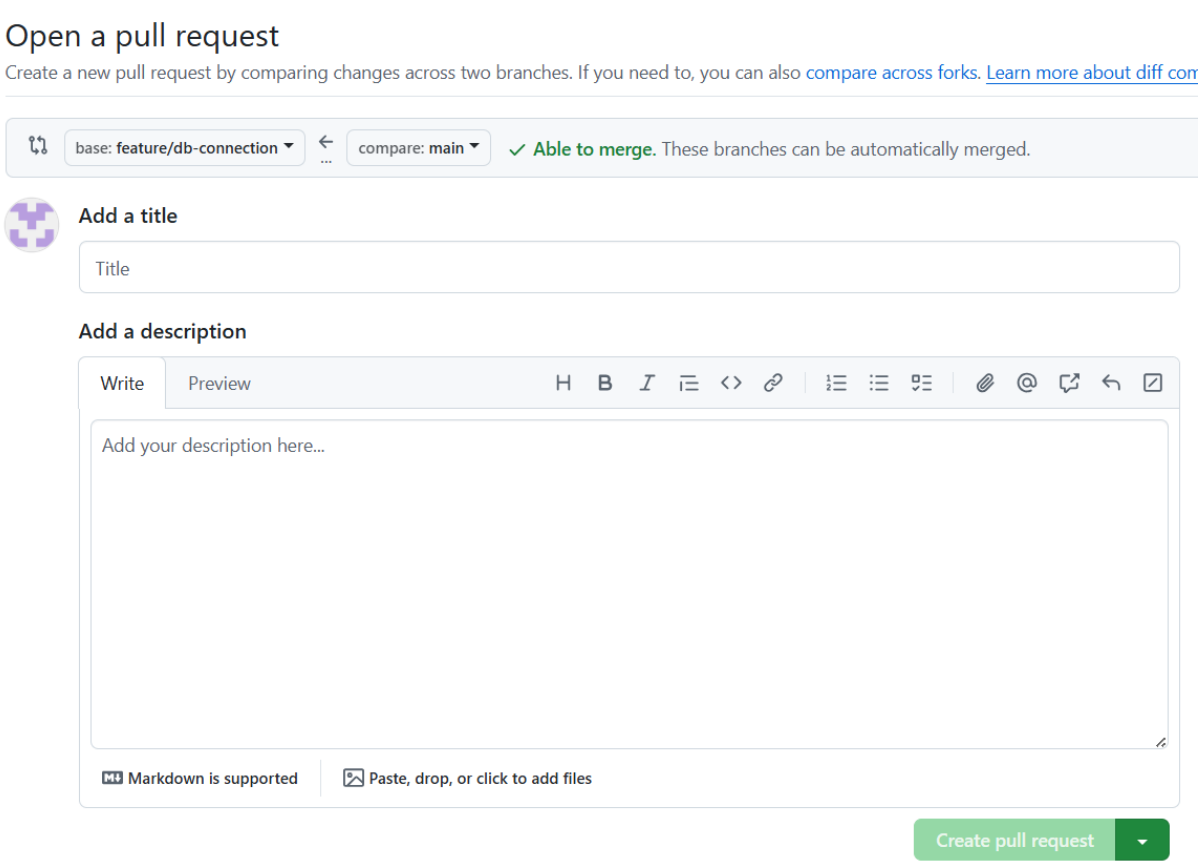
```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git push origin main
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 338 bytes | 112.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com-sudo:23-22411-057-sudo/Lab2.git
   f160fab..a28b864  main -> main

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$
git push origin feature-1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com-sudo:23-22411-057-sudo/Lab2.git
   f3e8d8f..3a2c34d  feature-1 -> feature-1

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$
```

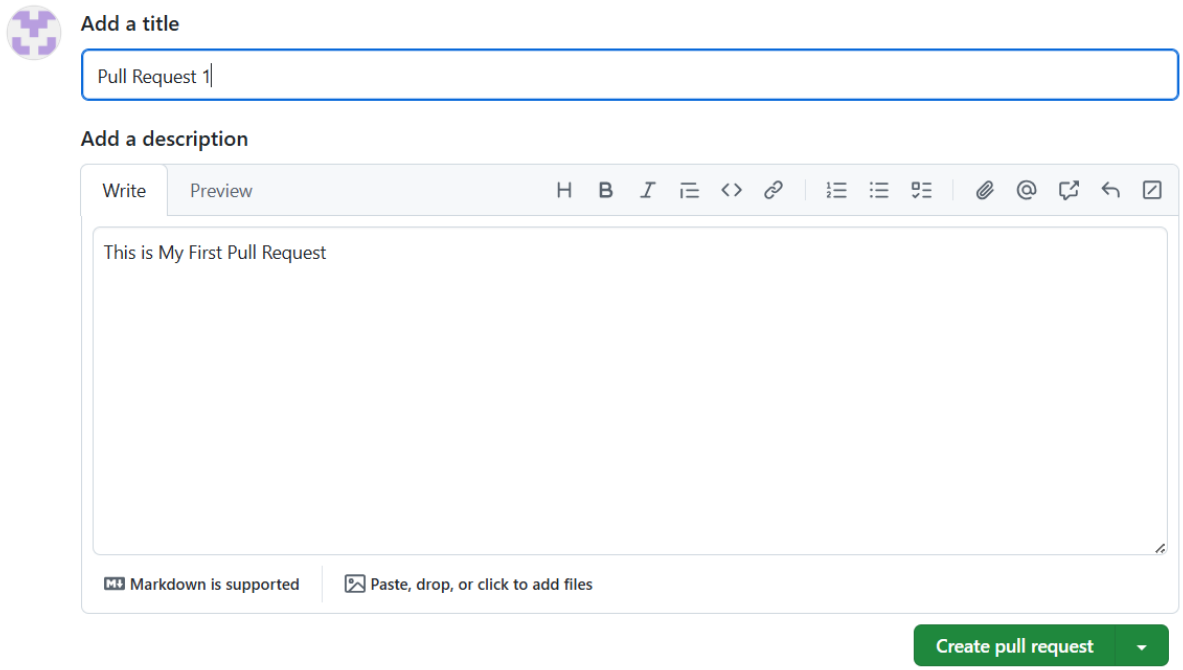
➢ Task 10:

Pull Request and Branch Review (GitHub GUI)

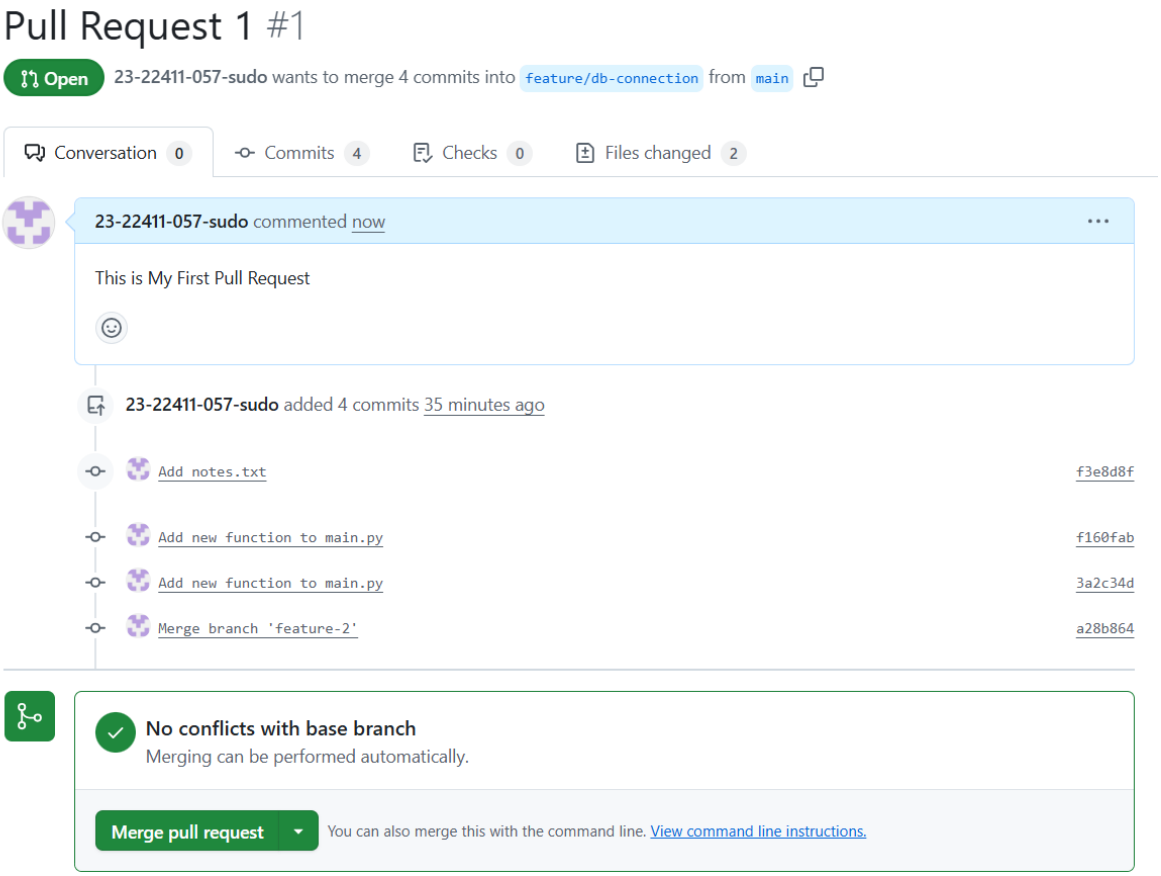
1. Create a Pull Request


Lab2
Private
Watch 0
feature/db-connection
4 Branches
0 Tags
Go to file
t
Add file
Code
This branch is 4 commits behind main.
Contribute
23-22411-057-sudo Add files via upload
7eec70a · 2 hours ago
4 Commits
README.md Initial commit 4 hours ago
git_Identit.PNG Add files via upload 2 hours ago
git_configlist.PNG Add files via upload 2 hours ago
git_installation.png Add files via upload 4 hours ago
github_ssh_key.png Add files via upload 2 hours ago
keygen_sshh.PNG Add files via upload 2 hours ago
repo_private.png Add files via upload 4 hours ago
ssh_keygen.png Add files via upload 2 hours ago

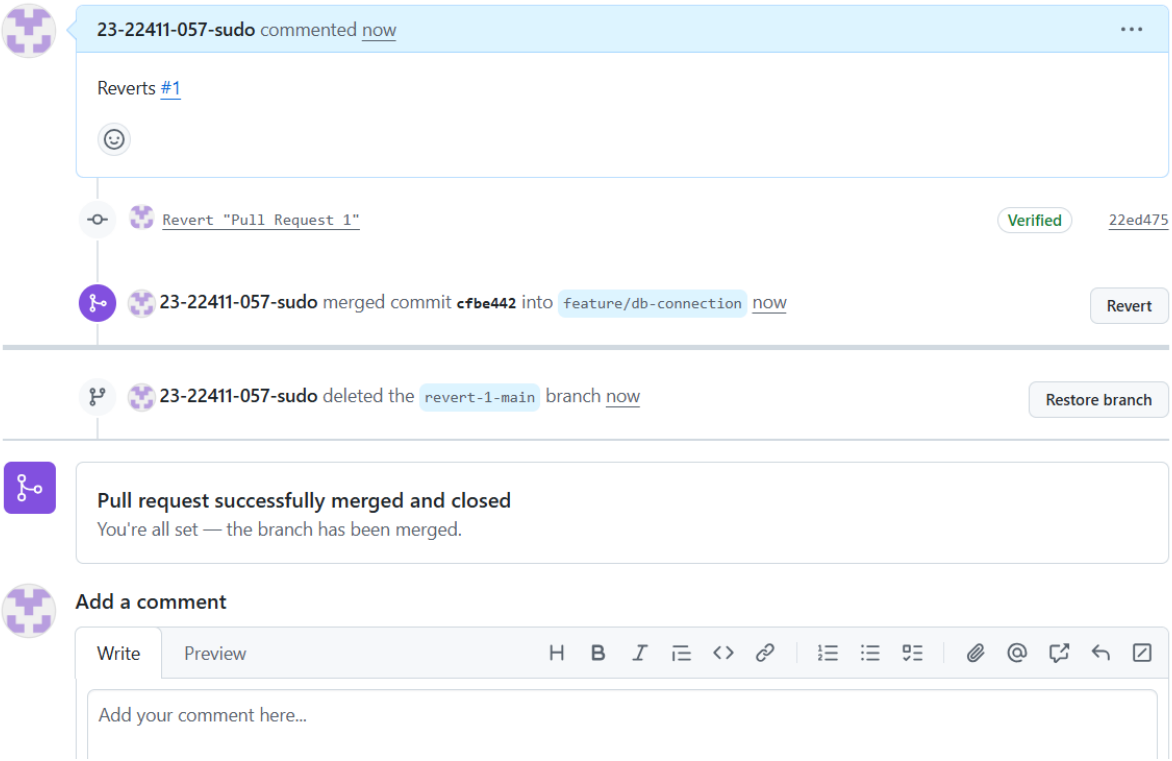

Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff com
base: feature/db-connection
compare: main
Able to merge. These branches can be automatically merged.
Add a title
Title
Add a description
Write
Preview
Add your description here...
Markdown is supported
Paste, drop, or click to add files
Create pull request



2. Review and Merge



3. Delete the branch



➢ Task 11:

Detailed Branch Strategy (Develop/Staging)

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git pull origin main
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            main       -> FETCH_HEAD
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git checkout -b develop
Switched to a new branch 'develop'
```

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git push 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/23-22411-057-sudo/Lab2/pull/new/develop
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      develop -> develop

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git checkout -b staging
Switched to a new branch 'staging'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (staging)
$ git push 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/23-22411-057-sudo/Lab2/pull/new/staging
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      staging -> staging
```

➢ Bonus Task:

Simulated Team Collaboration

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git checkout -b collab
Switched to a new branch 'collab'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (collab)
$ git checkout collab
Already on 'collab'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (collab)
$ git add notes.txt

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (collab)
$ git add notes.txt

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (collab)
$ git commit -m "I am Saira Ejaz"
On branch collab
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        clone_ssh.PNG

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

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (collab)
$ git push origin collab
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'collab' on GitHub by visiting:
remote:      https://github.com/23-22411-057-sudo/Lab2/pull/new/collab
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      collab -> collab
i
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (collab)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git pull origin main
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            main       -> FETCH_HEAD
Already up to date.
cl
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git merge collab
Already up to date.
```

Example Workflow:

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git checkout develop
Already on 'develop'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git pull origin develop
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            develop    -> FETCH_HEAD
Already up to date.
e
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git merge feature/login-page
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git push origin develop
Everything up-to-date

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git push origin staging
Everything up-to-date

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git merge develop
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git push origin staging
Everything up-to-date
m
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git pull origin main
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            main       -> FETCH_HEAD
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git merge staging
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git push origin main
Everything up-to-date

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git checkout staging
Switched to branch 'staging'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (staging)
$ git pull origin staging
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            staging    -> FETCH_HEAD
```

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git pull origin main
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            main       -> FETCH_HEAD
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git merge staging
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git push origin main
Everything up-to-date

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git checkout staging
Switched to branch 'staging'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (staging)
$ git pull origin staging
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            staging    -> FETCH_HEAD
Already up to date.
g
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (staging)
$ git merge develop
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (staging)
$ git push origin staging
Everything up-to-date
gi
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (staging)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git pull origin main
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            main       -> FETCH_HEAD
Already up to date.
s
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git merge staing
merge: staing - not something we can merge

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git merge staging
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git push origin main
Everything up-to-date

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$
```

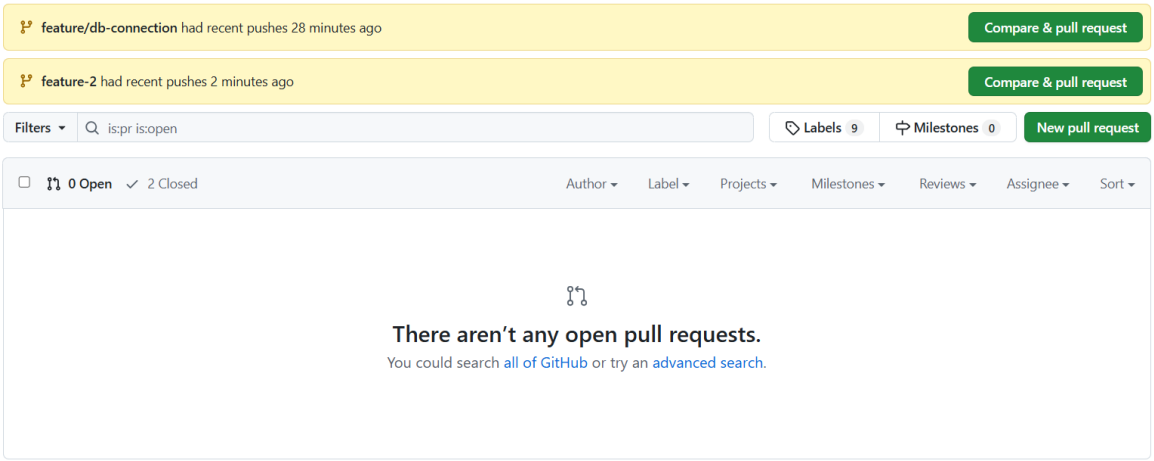
➢ Task 12:

Code Review Workflow

1. Create a Pull Request (PR) from a feature branch

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (collab)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git pull origin main
From github.com-sudo:23-22411-057-sudo/Lab2
 * branch            main       -> FETCH_HEAD
Already up to date.
cl
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git merge collab
Already up to date.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git push origin main
Everything up-to-date

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git checkout -b feature-2
Switched to a new branch 'feature-2'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-2)
$ echo "Print('Feature 2')" > feature2.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-2)
$ git add feature2.py
warning: in the working copy of 'feature2.py', LF will be replaced by CRLF the n
ext time Git touches it

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-2)
$ git commit -m "Add feature2.py"
[feature-2 78df11b] Add feature2.py
 1 file changed, 1 insertion(+)
 create mode 100644 feature2.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-2)
$ git push origin feature-2
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), 299 bytes | 74.00 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-2' on GitHub by visiting:
remote:      https://github.com/23-22411-057-sudo/Lab2/pull/new/feature-2
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      feature-2 -> feature-2
```

2. On GitHub Website:



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff com

base: main ← compare: feature-2 ✓ Able to merge. These branches can be automatically merged.

Add a title

Add feature2.py

Add a description

Write Preview

"This PR adds feature2.py for testing Git workflow"

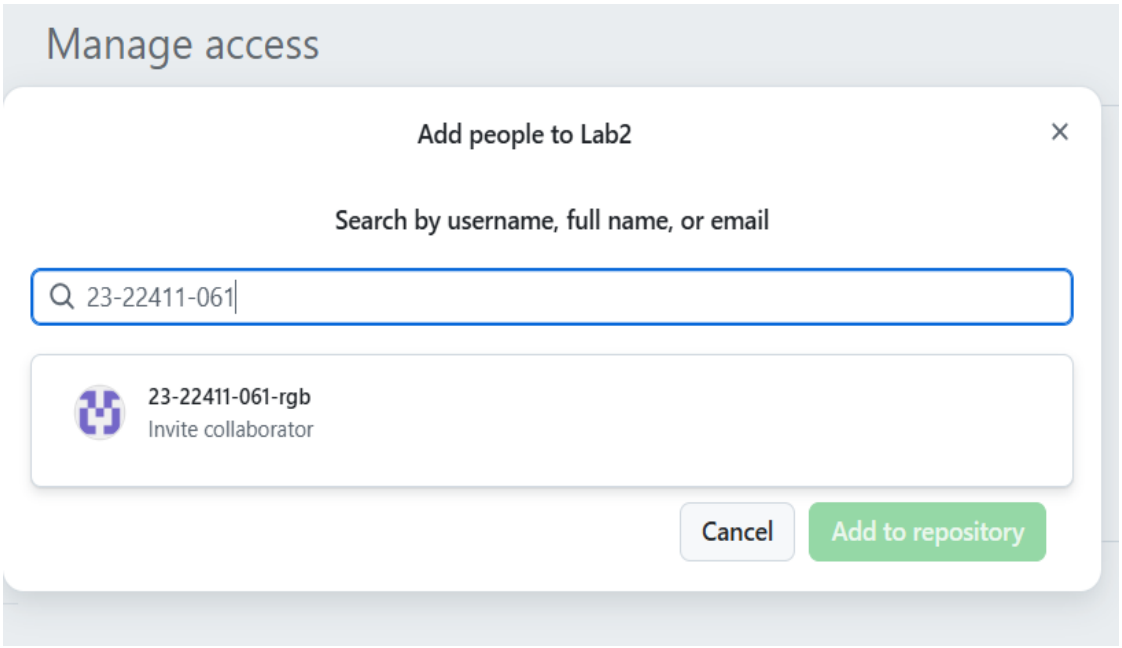
Markdown is supported | Paste, drop, or click to add files

Create pull request

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (staging)
$ git checkout develop
Switched to branch 'develop'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git checkout -b feature/login-page
Switched to a new branch 'feature/login-page'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature/login-page)
$ git push origin feature/login-page
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/login-page' on GitHub by visiting:
remote:      https://github.com/23-22411-057-sudo/Lab2/pull/new/feature/login-pa
ge
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      feature/login-page -> feature/login-page

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature/login-page)
$ git checkout develop
Switched to branch 'develop'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git checkout develop
Already on 'develop'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (develop)
$ git checkout -b bugfix/header-error
Switched to a new branch 'bugfix/header-error'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (bugfix/header-error)
$ git push origin bugfix/header-error
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'bugfix/header-error' on GitHub by visiting:
remote:      https://github.com/23-22411-057-sudo/Lab2/pull/new/bugfix/header-er
ror
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      bugfix/header-error -> bugfix/header-error

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (bugfix/header-error)
$
```

3. Assign a Reviewer

In the PR page, on the right sidebar assign a reviewer (your friend, second GitHub account, or just yourself if practicing).



Direct access

1 entity has access to this repository. 0 collaborators. 1 invitation.

Manage access

Add people

Select all Type

Find a collaborator...

23-22411-061-rgb
Awaiting 23-22411-061-rgb's response
Pending Invite

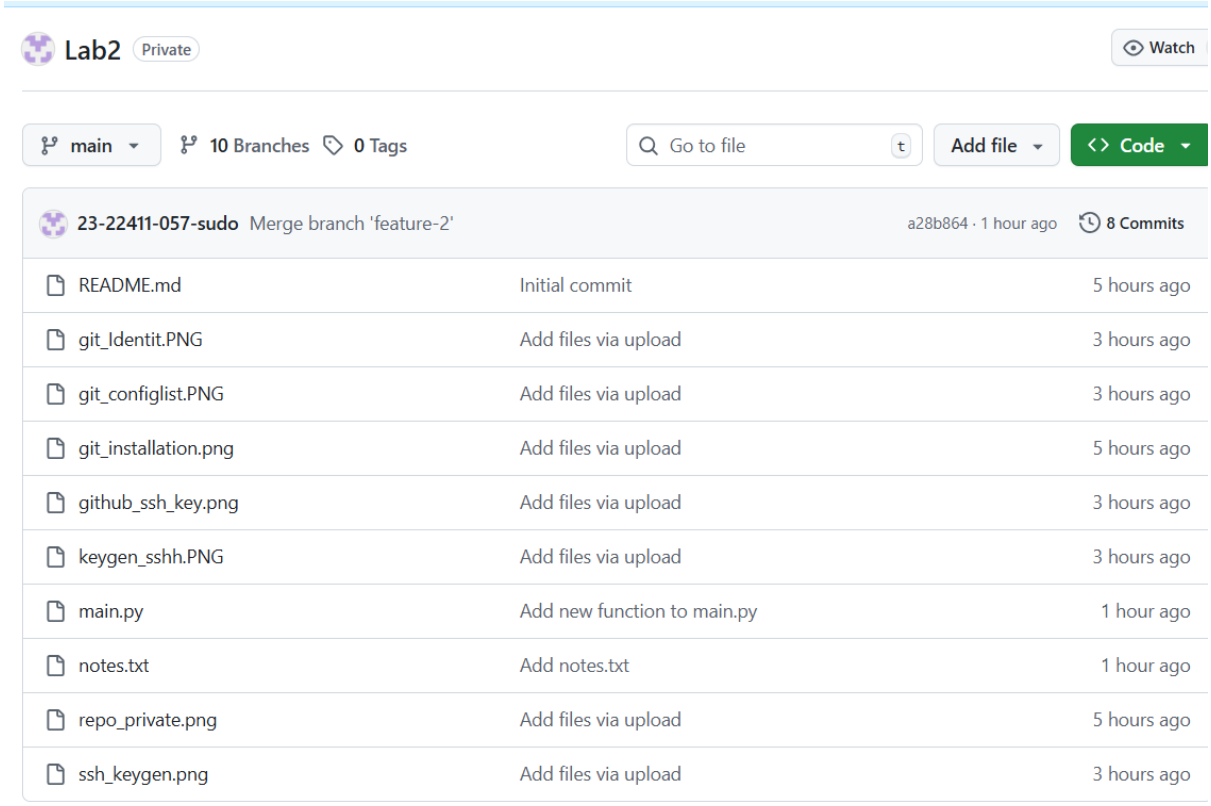
23-22411-057-sudo invited you to collaborate on 23-22411-057-sudo/Lab2

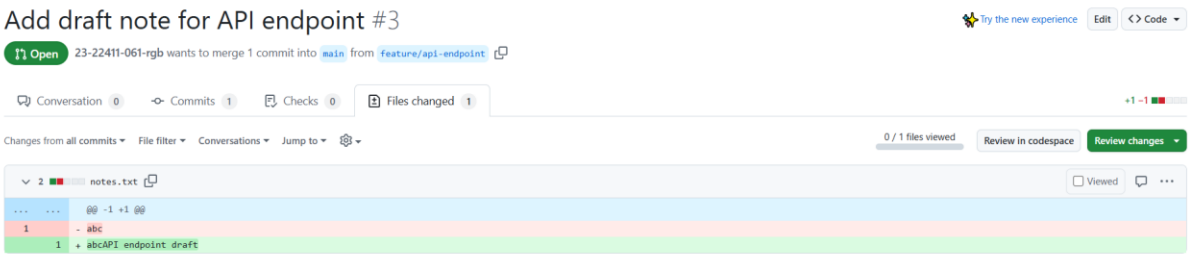
Accept invitation | Decline invitation

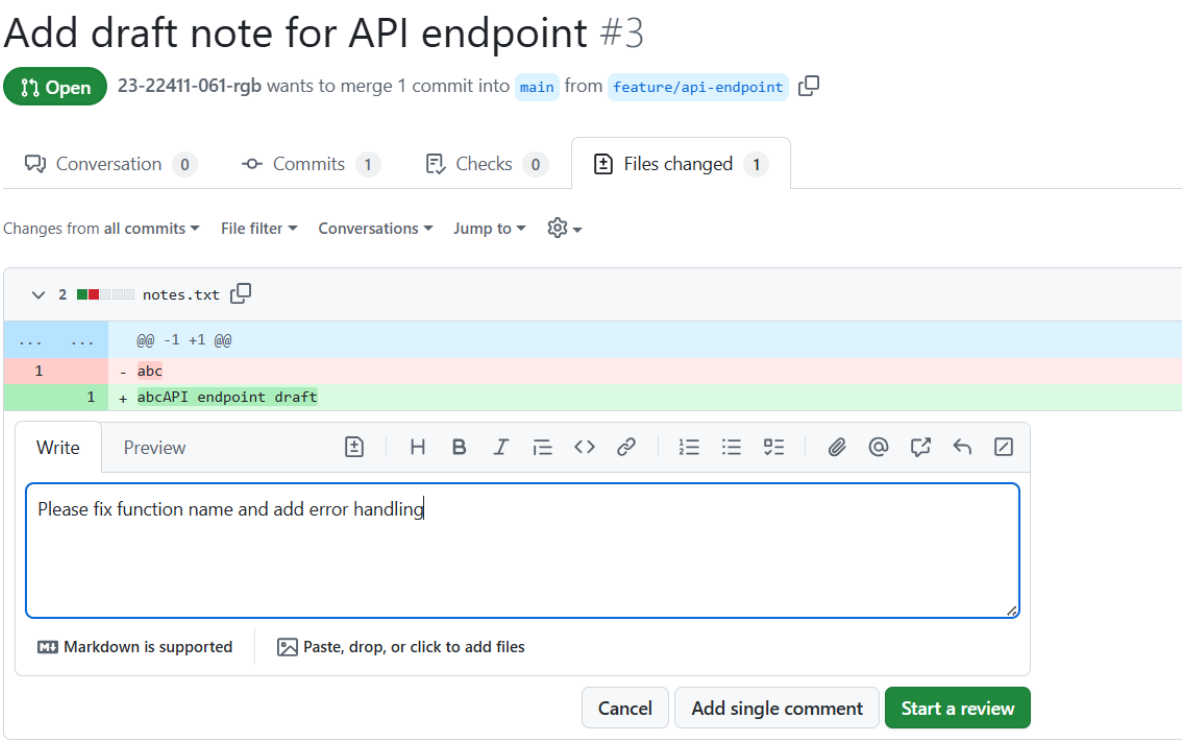
Owners of Lab2 will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?
Block 23-22411-057-sudo







4. Reviewer Actions and Approves:

If Athor approves then ok if not then he or shep made changes himself.

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-2)
$ git checkout feature-hello
error: pathspec 'feature-hello' did not match any file(s) known to git

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-2)
$ ^C

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-2)
$ git checkout -b feature-hello
Switched to a new branch 'feature-hello'

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ echo "def say_hello():\n    print('Hello World')" > main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ cat main.py
def say_hello():\n    print('Hello World')

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git add main.py
warning: in the working copy of 'main.py', LF will be replaced by CRLF the next time Git touches it

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git commit -m "Add say_hello function with fixed message"
[feature-hello 9d1eb44] Add say_hello function with fixed message
 1 file changed, 1 insertion(+), 1 deletion(-)

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git push origin feature-hello
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), 331 bytes | 66.00 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-hello' on GitHub by visiting:
remote:      https://github.com/23-22411-057-sudo/Lab2/pull/new/feature-hello
remote:
To github.com-sudo:23-22411-057-sudo/Lab2.git
 * [new branch]      feature-hello -> feature-hello

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$
```

Add draft note for API endpoint #3

Open 23-22411-061-rgb wants to merge 1 commit into main from feature/api-endpoint

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to

2 notes.txt

```
... ...   @@ -1 +1 @@
1         - abc
      1   + abcAPI endpoint draft
```

Write Preview

Please update say_hello() so it accepts a name parameter and prints Hello <name> instead of always "Hello World"

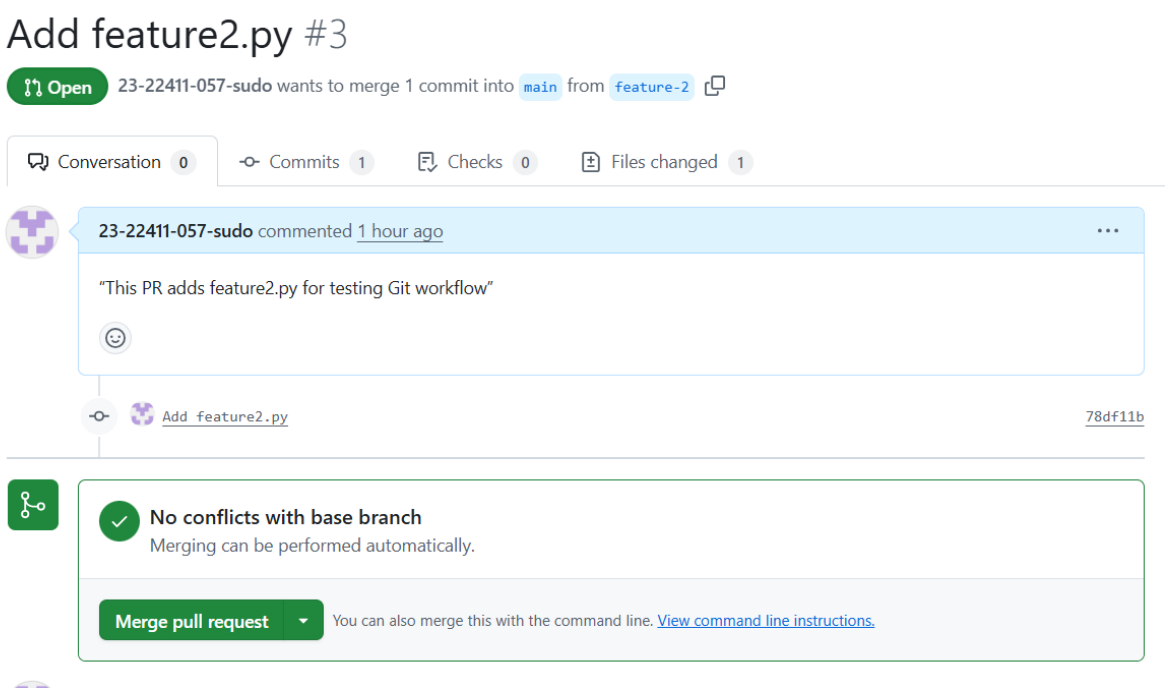
Markdown is supported Paste, drop, or click to add files

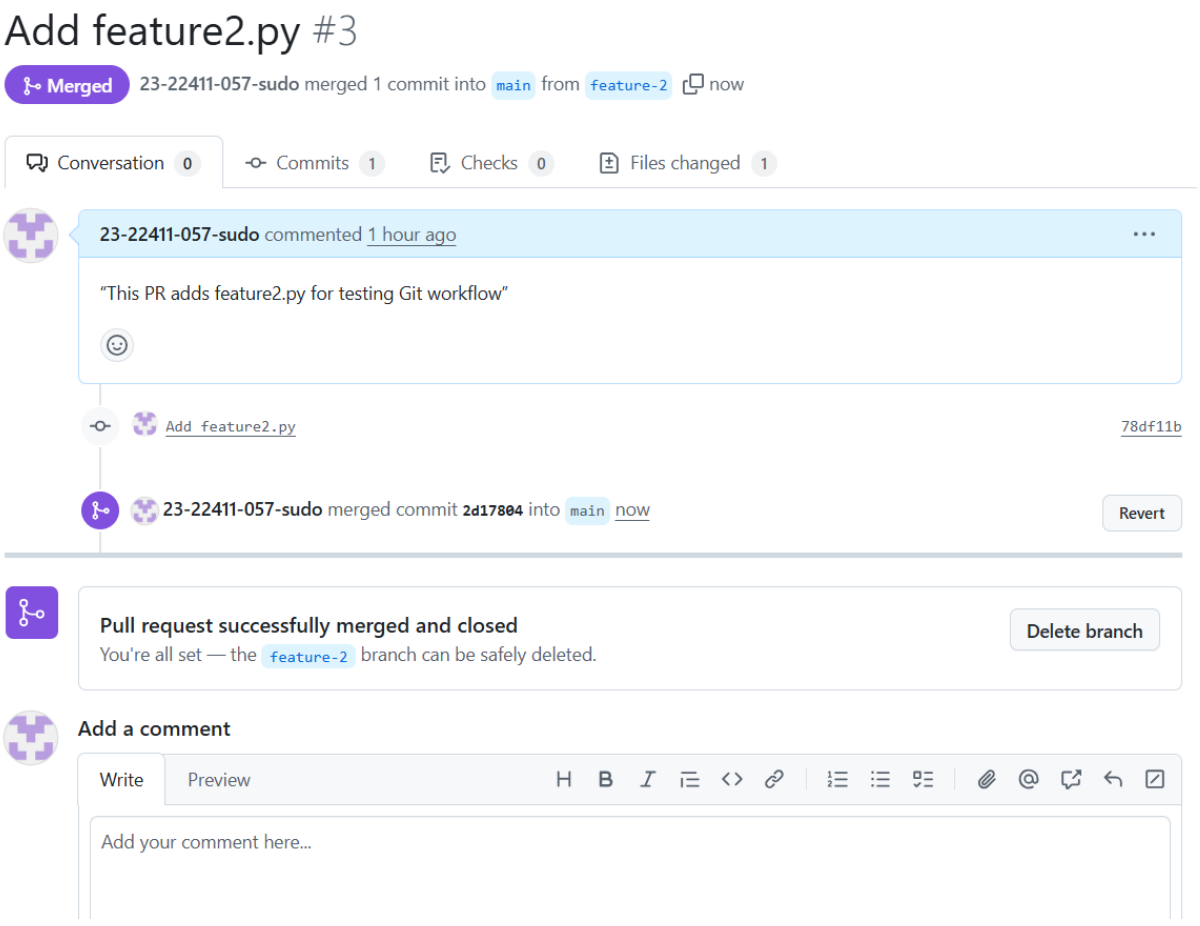
Cancel Add single comment Start a review

5. Update the file main.py

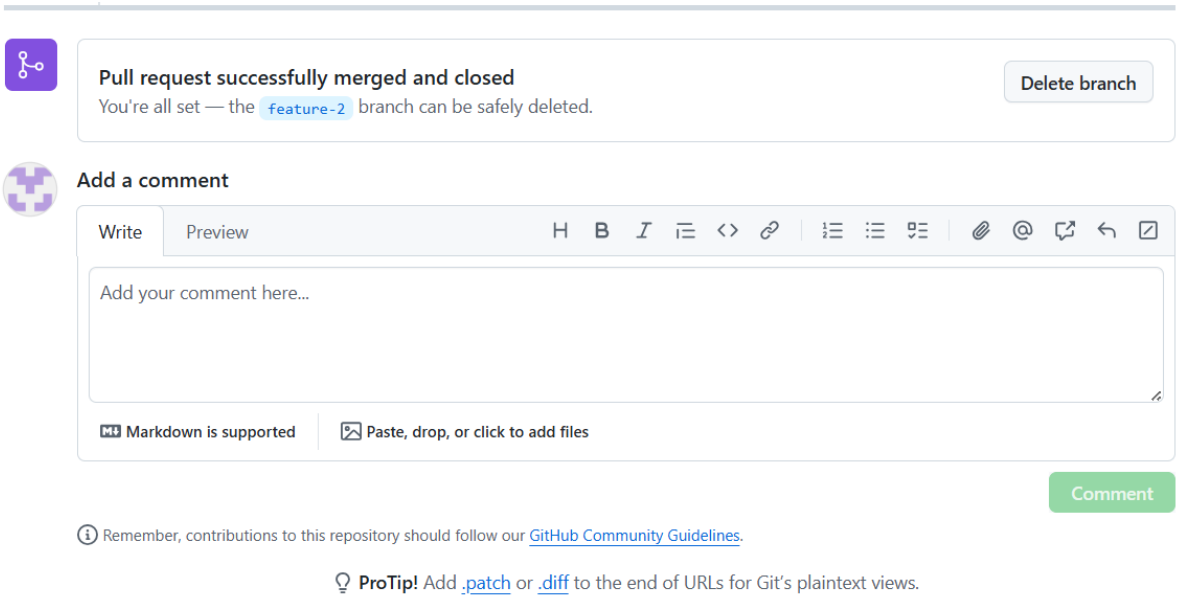
```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ echo "def say_hello(name):" > main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ echo "    print(f\"Hello {name}\")" >> main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ echo "" >> main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ echo "if __name__ == \"__main__\":" >> main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ echo "    say_hello(\"Saira\")" >> main.py

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ cat main.py
def say_hello(name):
    print(f"Hello {name}")

if __name__ == "__main__":
    say_hello("Saira")

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git add main.pay
fatal: pathspec 'main.pay' did not match any files

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git add maain.py
fatal: pathspec 'maain.py' did not match any files

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git main.py
git: 'main.py' is not a git command. See 'git --help'.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git add main.py
warning: in the working copy of 'main.py', LF will be replaced by CRLF the next
time Git touches it

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git commit -m "Update say_hello to accept a name parameter"
[feature-hello 289a0bd] Update say_hello to accept a name parameter
 1 file changed, 5 insertions(+), 1 deletion(-)

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git push origin feature-hello
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), 370 bytes | 185.00 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-sudo:23-22411-057-sudo/Lab2.git
   9d1eb44..289a0bd  feature-hello -> feature-hello

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
```

6. Merge the PR:





7. Branch handling after merge
   Delete the branch



Keep the branch

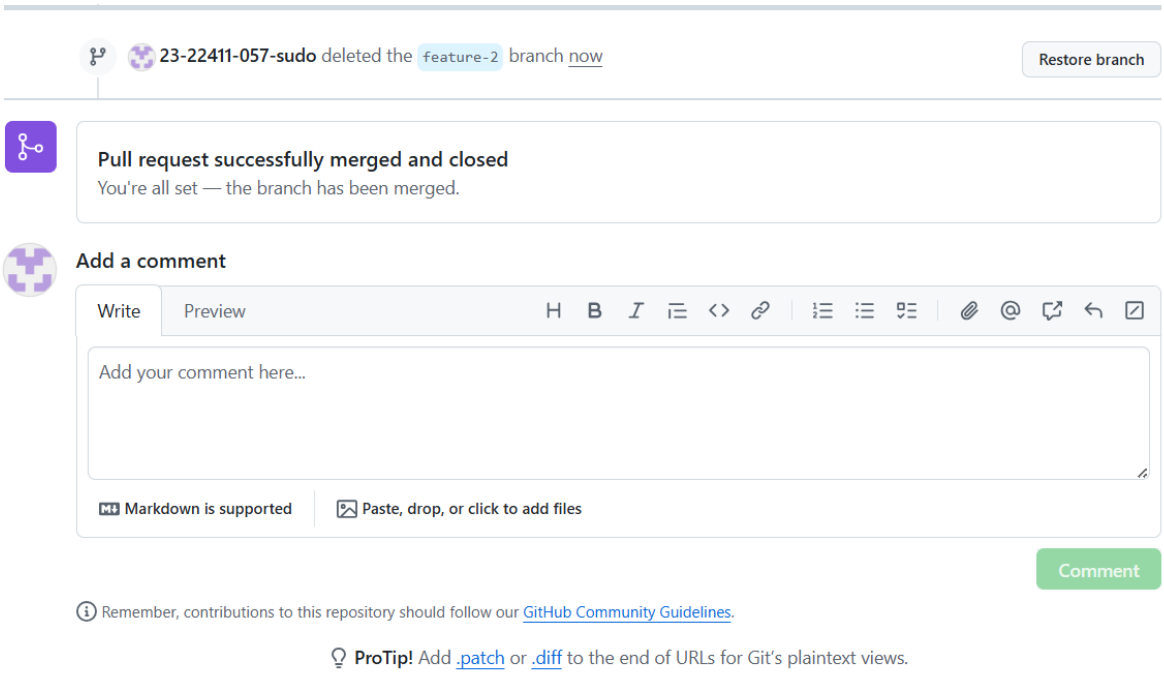
- Keep the branch if you plan more work on it.
- Document the reason in a file called REVIEW_NOTES.md

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ echo "# REVIEW_NOTES.md" > REVIEW_NOTES.md
echo "" >> REVIEW_NOTES.md
echo "## feature-hello branch" >> REVIEW_NOTES.md
echo "- Kept after merge for possible future enhancements." >> REVIEW_NOTES.md
echo "- Original change: Updated say_hello() to accept a name parameter." >> REVIEW_NOTES.md

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ cat REVIEW_NOTES.md
# REVIEW_NOTES.md

## feature-hello branch
- Kept after merge for possible future enhancements.
- Original change: Updated say_hello() to accept a name parameter.

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git add REVIEW_NOTES.md
git commit -m "Add branch handling notes after PR merge"
git push origin feature-hello
warning: in the working copy of 'REVIEW_NOTES.md', LF will be replaced by CRLF the next time Git touches it
[feature-hello 1271070] Add branch handling notes after PR merge
 1 file changed, 5 insertions(+)
 create mode 100644 REVIEW_NOTES.md
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 440 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.
To github.com-sudo:23-22411-057-sudo/Lab2.git
   289a0bd..1271070  feature-hello -> feature-hello

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
```

➢ Task 13:

Branch Cleanup Best Practices

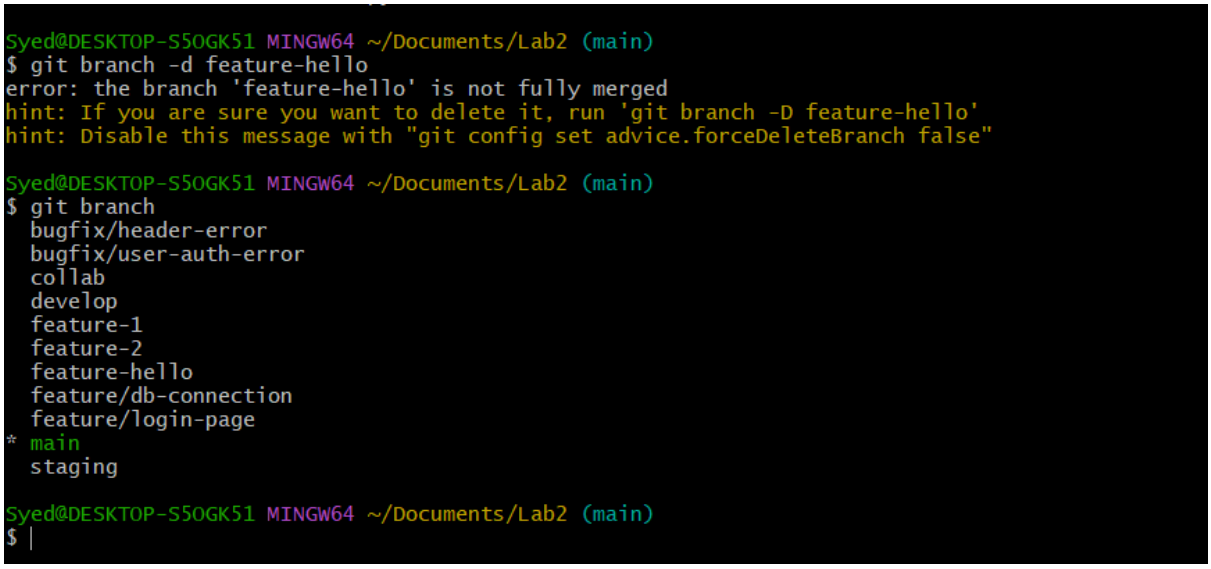
1. Delete the Remote Branch via GitHub UI



2. Git Bash command to verify deletion (optional)

```
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$ git fetch -p
From github.com-sudo:23-22411-057-sudo/Lab2
 - [deleted]         (none)                -> origin/feature-2
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 3.42 KiB | 102.00 KiB/s, done.
   7eec70a..cfbe442  feature/db-connection -> origin/feature/db-connection
   a28b864..2d17804  main                  -> origin/main

Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (feature-hello)
$
```

3. Update your Local Repository


Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git branch -d feature-hello
error: the branch 'feature-hello' is not fully merged
hint: If you are sure you want to delete it, run 'git branch -D feature-hello'
hint: Disable this message with "git config set advice.forceDeleteBranch false"
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$ git branch
bugfix/header-error
bugfix/user-auth-error
collab
develop
feature-1
feature-2
feature-hello
feature/db-connection
feature/login-page
* main
staging
Syed@DESKTOP-S5OGK51 MINGW64 ~/Documents/Lab2 (main)
$