Name:Rakshitha Shetty

B.Tech CSE(AIML)

A86617512408

SI. NO.   INDEX

- Branches Creation

- Merge Request

- Open and Close Pull Request

- Complete Git Process

- EXPERIMENT-01:

- Basics Of Linux:

- 1.What is Linux?

- Linux is a family of free and open-source operating systems based on the Linux kernel. Operating systems based on Linux are known as Linux distributions or distros. Examples include Debian, Ubuntu, Fedora, CentOS, Gentoo, Arch Linux, and many others.

- 2. Linux Distributions (Distros)

- Different versions of Linux tailored for various needs:
- Ubuntu – User-friendly, great for beginners.
- Debian – Stable and well-tested.
- Fedora – Latest features, cutting-edge.
- Arch Linux – Lightweight and customizable.
- CentOS/RHEL – Used in enterprise environments.

- 3. Linux File System Structure
- Linux follows a hierarchical structure:
- bin → essential binary programs
- etc → configuration files
- home → user directories
- root → root user’s home directory

- var variable data (logs, etc.)

- tmp → temporary files

- usr → user-installed software

- 4. Basic Linux Commands

- Command Description

- Is. – List files in a directory

- Cd - Change directory

- Pwd -Print working directory

- Mkdir. Make a new directory

- Rm - Remove files/directories

- Cp. – Copy files/directories

- Mv. - Move or rename files

- Cat.-   Display file content

- Sudo. – Execute a command as superuser

- man

- Show manual/help for a command

- 5. File Permissions

- Linux controls file access with permissions.

- Use ls -l to view them (e.g., -rwxr-xr--).

- Modify with chmod, chown, or chgrp.

- 6. Package Management
- Install/update software using package managers:
- Debian/Ubuntu: apt (e.g., sudo apt install package-name)
- Red Hat/Fedora: dnf or yum
- Arch Linux: pacman
- 7. Shell and Terminal
- The shell interprets commands (e.g., bash, zsh).
- Terminal is where you type the commands.
- 8. Users and Permissions

- Regular users vs. root (superuser).

- Manage users with adduser, usermod, and passwd

- Experiment-02:

- Introduction -Git Bash

- What Is Git Bash?

- Git Bash is a command-line tool for Windows that provides:

- Git command-line tools

- A Bash (Unix-style) shell

- . A way to run shell commands similar to those used in Linux/macOS

- It’s especially useful for developers using Git on Windows who want a Unix-like experience.

- Key Features:

- Bash Emulation:
-

 Bash (Bourne Again Shell) lets you run Linux-style commands (e.g., ls, pwd, rm).

 This makes it easier to follow tutorials written for Linux/macOS.

- Git Integration:
-

 다 You can run Git commands like git clone, git status, git commit directly in Git Bash.

 Useful for version control and managing code repositories.

- Cross-Platform Compatibility:
-

 Lets Windows users interact with remote Linux servers or repositories more easily.

 How to Install Git Bash:

 Go to https://git-scm.com

 Download the installer for your OS (choose Windows).

 Run the installer and select “Git Bash” when prompted about default terminal.

 After installation, right-click anywhere and choose “Git Bash Here” to open the terminal.

 Common Commands in Git Bash:

Command.	Description
Ls.	List files and directories
Pwd	Print current directory path
Cd.	Change directory
Mkdir myfolder	Create a new folder
Rm filename	Remove a file
Git init	Initialize a new Git repositor

Command.	Description

Git clone URL.	Clone a repository from GitHub
Git status.	Show current Git status
Git add.	Stage all changes
Git commit -m “message”	Commit staged changes

- Use Cases:

- Managing Git repositories

- Running shell scripts

- Navigating your project with Unix-style commands

- Automating development tasks

- Experiment-03:

- Git Bash And GitHub

- What is Git?

- Before connecting GitHub and Git Bash, it’s important to know:

- Git is a version control system to track changes in code.

- GitHub is a hosting service for Git repositories.

- • Git Bash lets you use Git on Windows with Linux-style

- commands.

- 🔗How They Work Together

- Here’s the typical workflow:

- You install Git and Git Bash on your Windows

- machine.

- You create or clone a Git repository using Git Bash.

- You make changes to your files locally.

- You use Git commands in Git Bash to:

Stage changes: git add

- commit changes: git commit

- push changes: git push (to GitHub)

You use GitHub to:

- store your code in the cloud

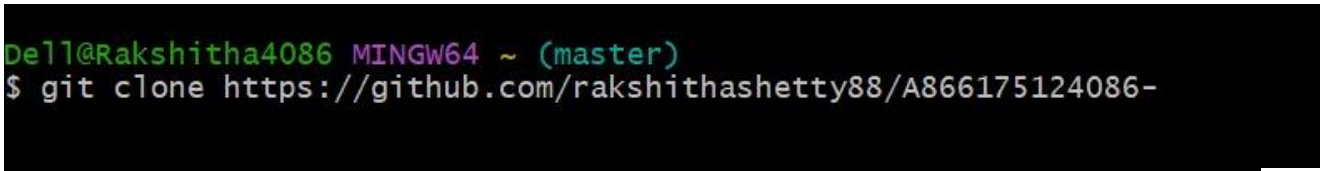
- collaborate with others

- view project history, pull requests, issues, etc.

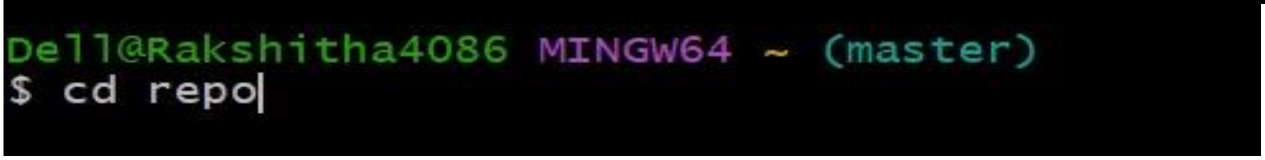
Example: Basic Workflow

Step-by-step using Git Bash & GitHub:

- Clone a GitHub repo:



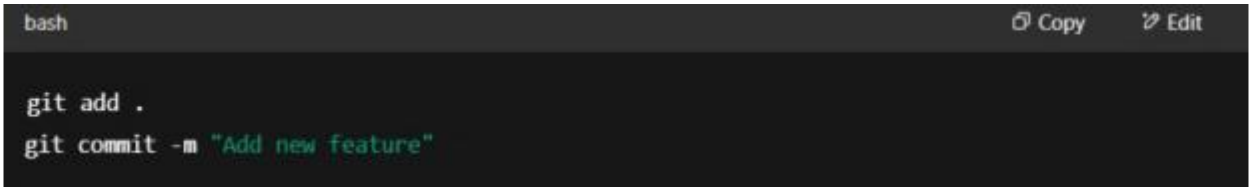
```
Dell@Rakshitha4086 MINGW64 ~ (master)
$ git clone https://github.com/rakshithashetty88/A866175124086-
```



```
Dell@Rakshitha4086 MINGW64 ~ (master)
$ cd repo
```

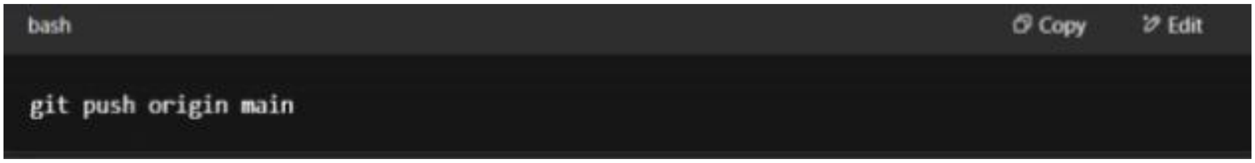
Make changes to our project files.

- Stage and commit changes:



```bash
git add .
git commit -m "Add new feature"
```

- Push changes to GitHub:



```
git push origin main
```

Experiment-04:

File Creation With Commit And Push

Command.

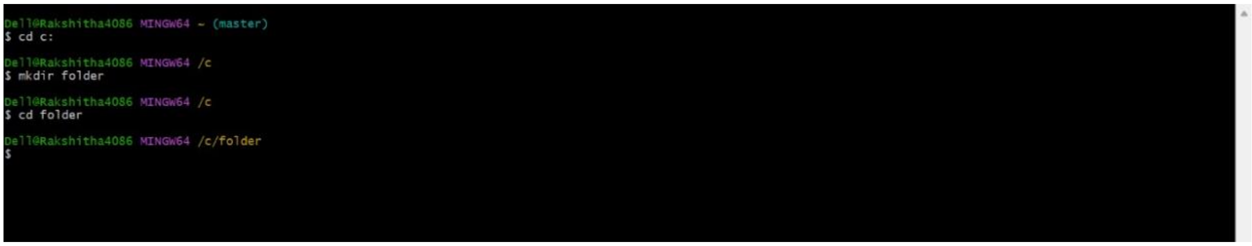
- To Create a file,commit it and push it to a remote

Repository using Git Bash

1.Create a File.

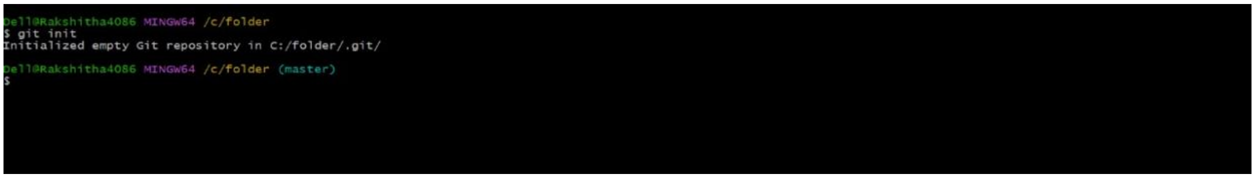
- Open Git Bash and create a folder :

- Using cd and mkdir create a folder:



```
Dell@Rakshitha4086 MINGW64 ~ (master)
$ cd c:

Dell@Rakshitha4086 MINGW64 /c
$ mkdir folder

Dell@Rakshitha4086 MINGW64 /c
$ cd folder

Dell@Rakshitha4086 MINGW64 /c/folder
$
```

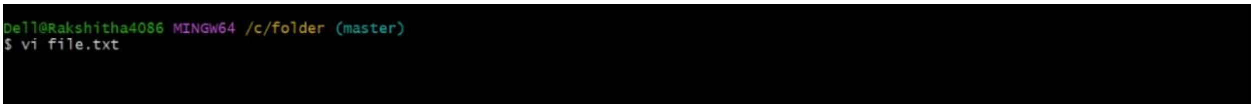
- Then git init command to create a new Git repository



```
Dell@Rakshitha4086 MINGW64 /c/folder
$ git init
Initialized empty Git repository in C:/folder/.git/

Dell@Rakshitha4086 MINGW64 /c/folder (master)
$
```

⍰ Create a new file using vi command in Linux is used to open

And edit files using the vi editor, a powerful text editor available on most Unix-based systems:



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ vi file.txt
```

Now open the file and write:

Basic vi commands;

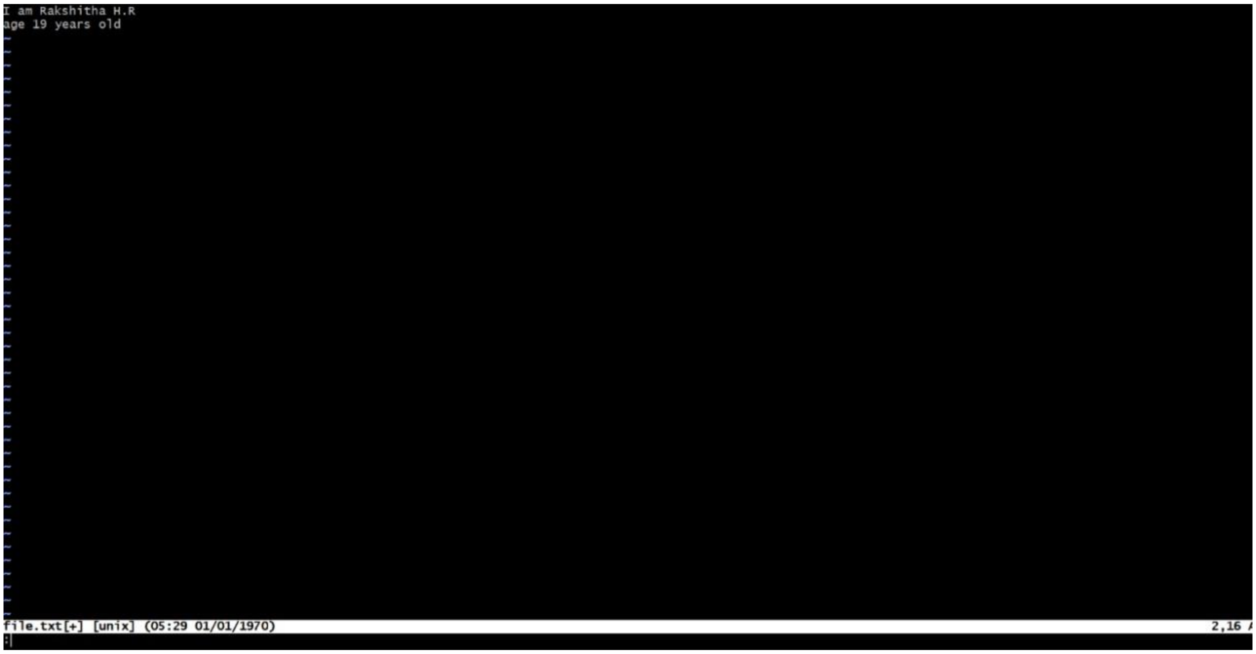
• Insert Mode:

O Press I to start editing.

• Save and Exit:

O Press ESC then type :wq to save and exit.

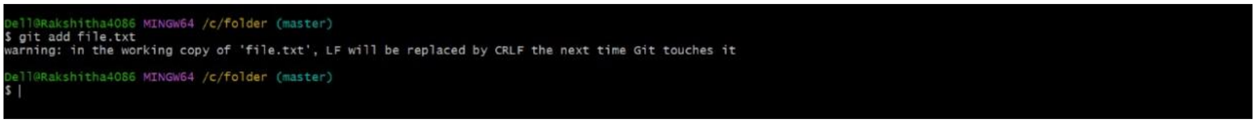
O Use :q! to exit without saving.



```
I am Rakshitha H.R
age 19 years old
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
file.txt[+] [unix] (05:29 01/01/1970)                                                                   2,16
:
```

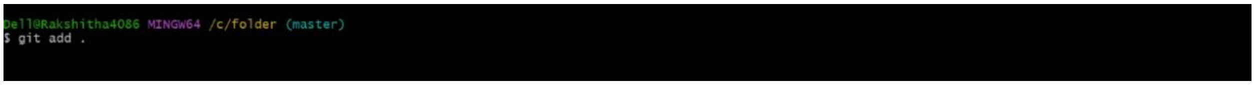
2.Add the File to Git:

O Stage the file for commit :



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it

Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ |
```

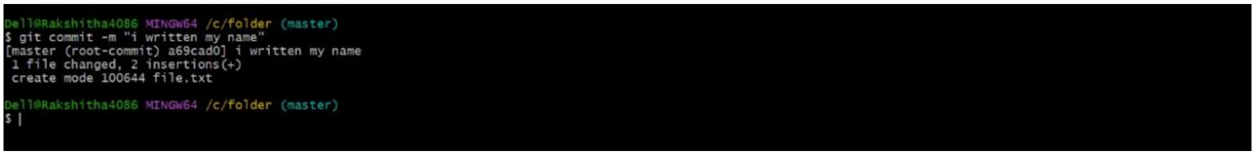
O To add all files in the directory:



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git add .
```

3.Commit the File:

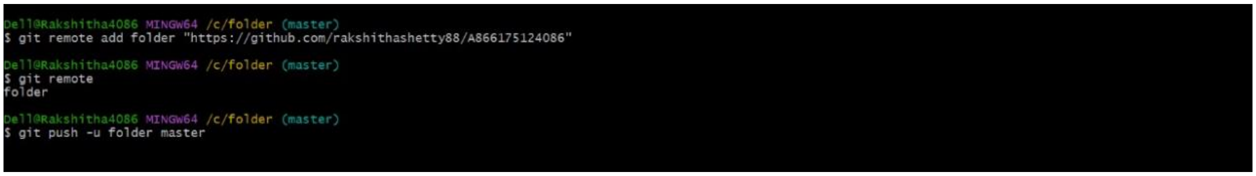
O Commit the changes with a message.



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git commit -m "i written my name"
[master (root-commit) a69cad0] i written my name
 1 file changed, 2 insertions(+)
 create mode 100644 file.txt

Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ |
```

4.Add and push to remote repository:

Push the changes to Git Hub (assuming origin is the remote and main is the branch)



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git remote add folder "https://github.com/rakshithashetty88/A866175124086"

Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git remote
folder

Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git push -u folder master
```

Experiment-05:

Branches Creation

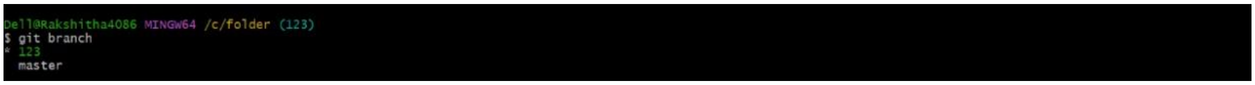
In Git branches allow developers to work on different

Features or fixes without affecting the main

Codebase. Here’s how we can create and manage branches.

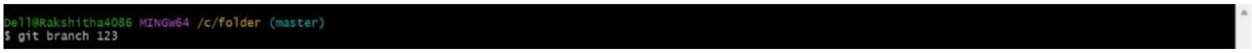
O Check the branches:

Check the branches using git branch command:



```
Dell@Rakshitha4086 MINGW64 /c/folder (123)
$ git branch
* 123
  master
```

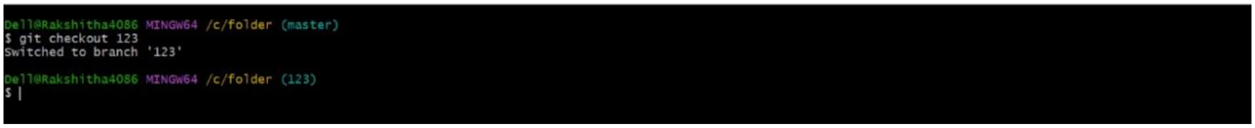
To create new branch ;



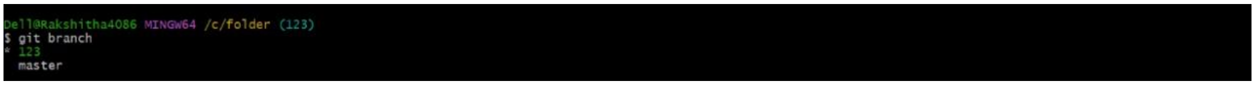
```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git branch 123
```

O Switching to a Branch:

To switch to the newly created branch:



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git checkout 123
Switched to branch '123'

Dell@Rakshitha4086 MINGW64 /c/folder (123)
$ |
```



```
Dell@Rakshitha4086 MINGW64 /c/folder (123)
$ git branch
* 123
  master
```

Experiment-06:

Merge Request

O A Merge Request(MR) is a feature used in Gitbased platforms like GitLab to propose and review

Changes before merging them into the main

Branch.It is similar to a Pull Request(PR) in

GitHub.

How Merge Request Works:

1. Create a Branch

A developer creates a separate branch from the main

Branch (often called main or master) to work on a specific

Feature, bug fix, or enhancement.

2. Make Changes

The developer writes code, commits changes to their

Branch, and pushes the branch to the remote repository.

3.Open a Merge Request

In the Git platform (e.g., GitLab, GitHub):

• The developer opens a merge request from their

Feature branch into the target branch (usually main).

• They add a description, possibly link related issues,

And assign reviewers.

3. Code Review

• Reviewers (team members or maintainers) review

The code.

• Automated checks (e.g., tests, linters) are run via

CI/CD pipelines.

• Feedback might be given; the author can make

Additional commits to address it.

6. Merge

The MR is merged into the target branch:

• Options include merge, squash and merge, or

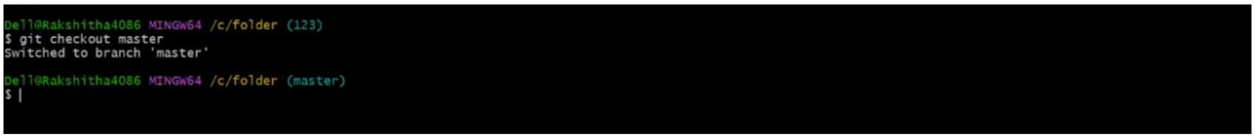
Rebase and merge.

• Once merged, the feature branch can be deleted if no

Longer needed.

Steps to Merge the Branch:

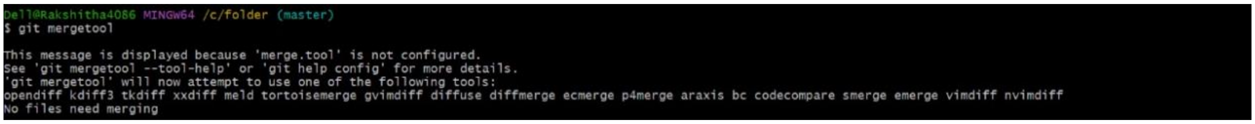
1.Switch to the main branch (master):



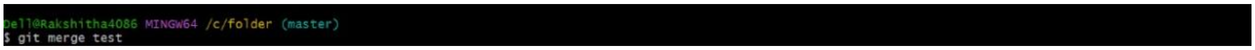
```
Dell@Rakshitha4086 MINGW64 /c/folder (123)
$ git checkout master
Switched to branch 'master'

Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ |
```

2.Merge the branch:

Using git mergetool and git merge main command:



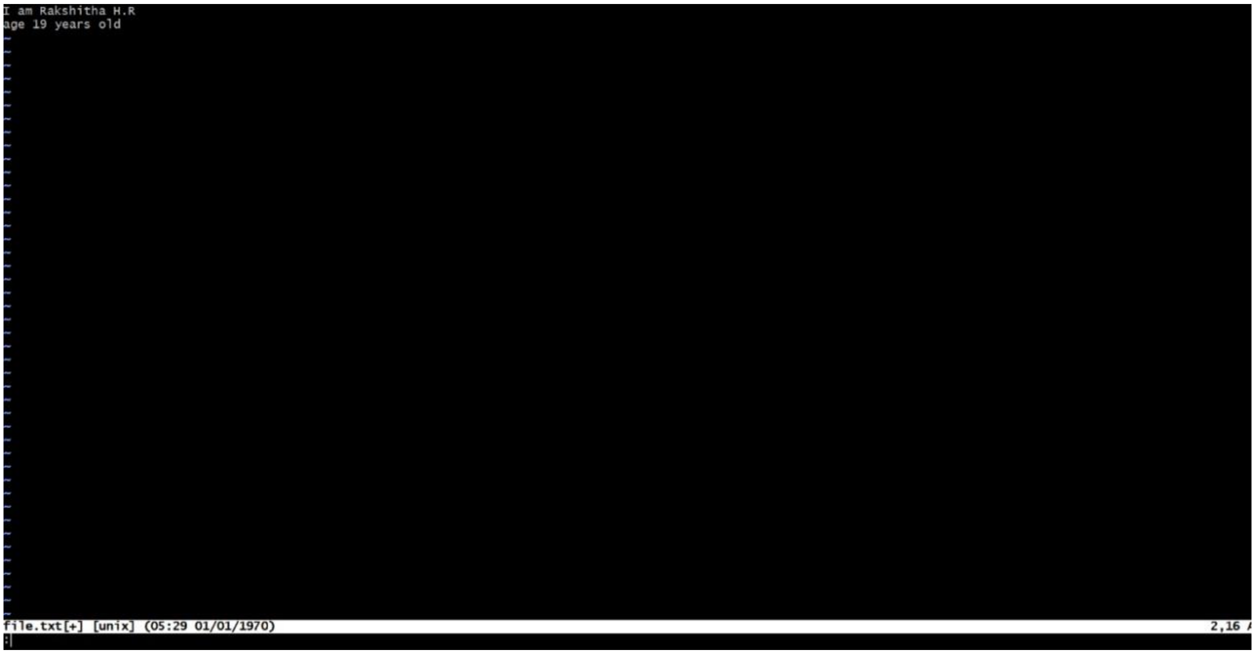
```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
No files need merging
```



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git merge test
```

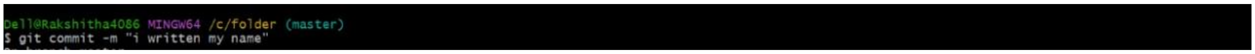
Now Press enter to edit the files and windows are opening

,then remove some red lines and add some lines:



```
I am Rakshitha H.R
age 19 years old
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
file.txt[+] [unix] (05:29 01/01/1970)                                                                                  2,16
:
```

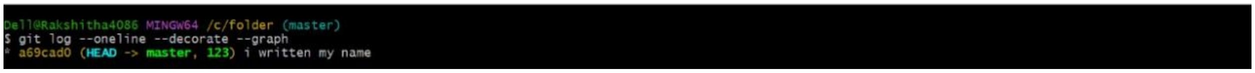
To remove merging commit one line:



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git commit -m "i written my name"
```

Using the command graph we can see the graph of

Commits:



```
Dell@Rakshitha4086 MINGW64 /c/folder (master)
$ git log --oneline --decorate --graph
* a69cad0 (HEAD -> master, 123) i written my name
```

Benefits:

- Facilitates code review

- Triggers automated tests

- Maintains a clear change history

- Encourages collaborative development

Experiment-07:

- Open and Close Pull Request

1. Open a Pull Request

1. Push your changes to a branch on your fork or the

Same repository.

2. Go to GitHub, navigate to the repository.

3. You’ll see a “Compare & pull request” button — click

It.

4. Add a title and description for your PR.

5. Click “Create pull request”.

2. Close a Pull Request

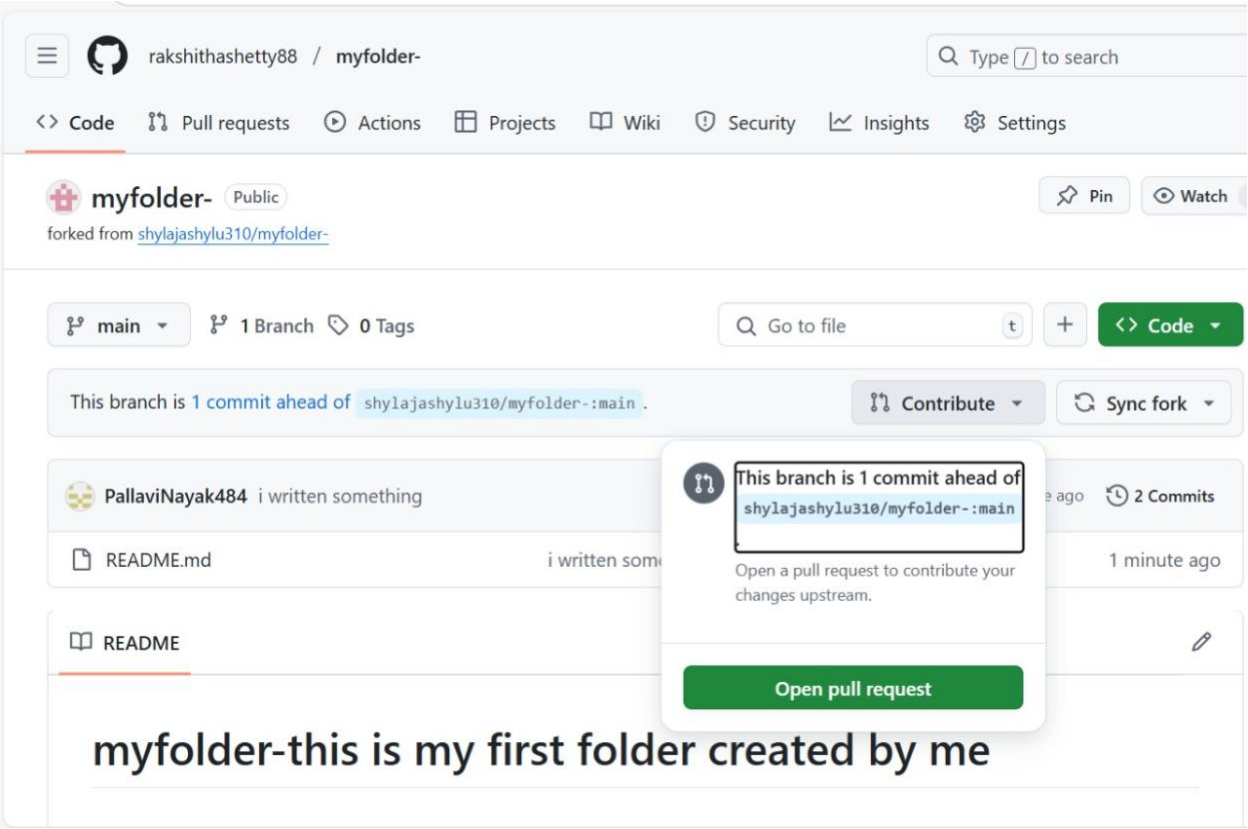
1. Click “Merge pull request”.

2. Confirm by clicking “Confirm merge”.

3. Optionally, delete the branch.

In the git hub account select the user which whom you

Want to merge and select the repo and fork it:



myfolder- Public

forked from shylajashylu310/myfolder-

This branch is 1 commit ahead of `shylajashylu310/myfolder-:main` .

PallaviNayak484 i written something ... e ago

README.md i written som... 1 minute ago

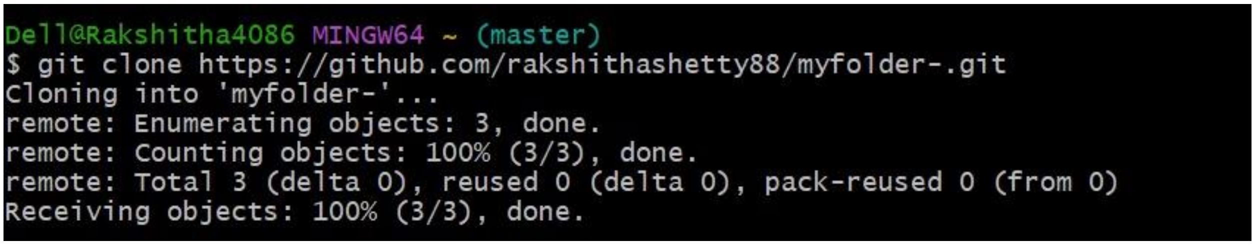
This branch is 1 commit ahead of `shylajashylu310/myfolder-:main` .

Open a pull request to contribute your changes upstream.

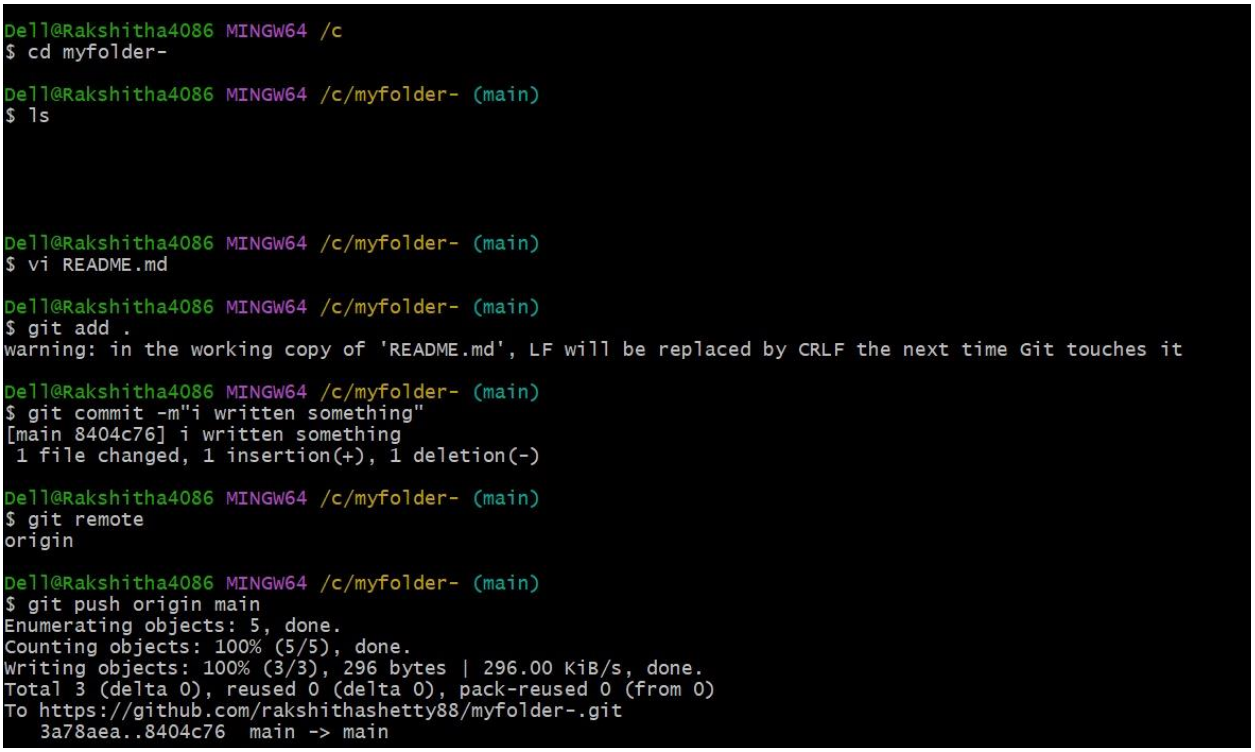
myfolder-this is my first folder created by me

Now copy the link and using git clone command open that

File:

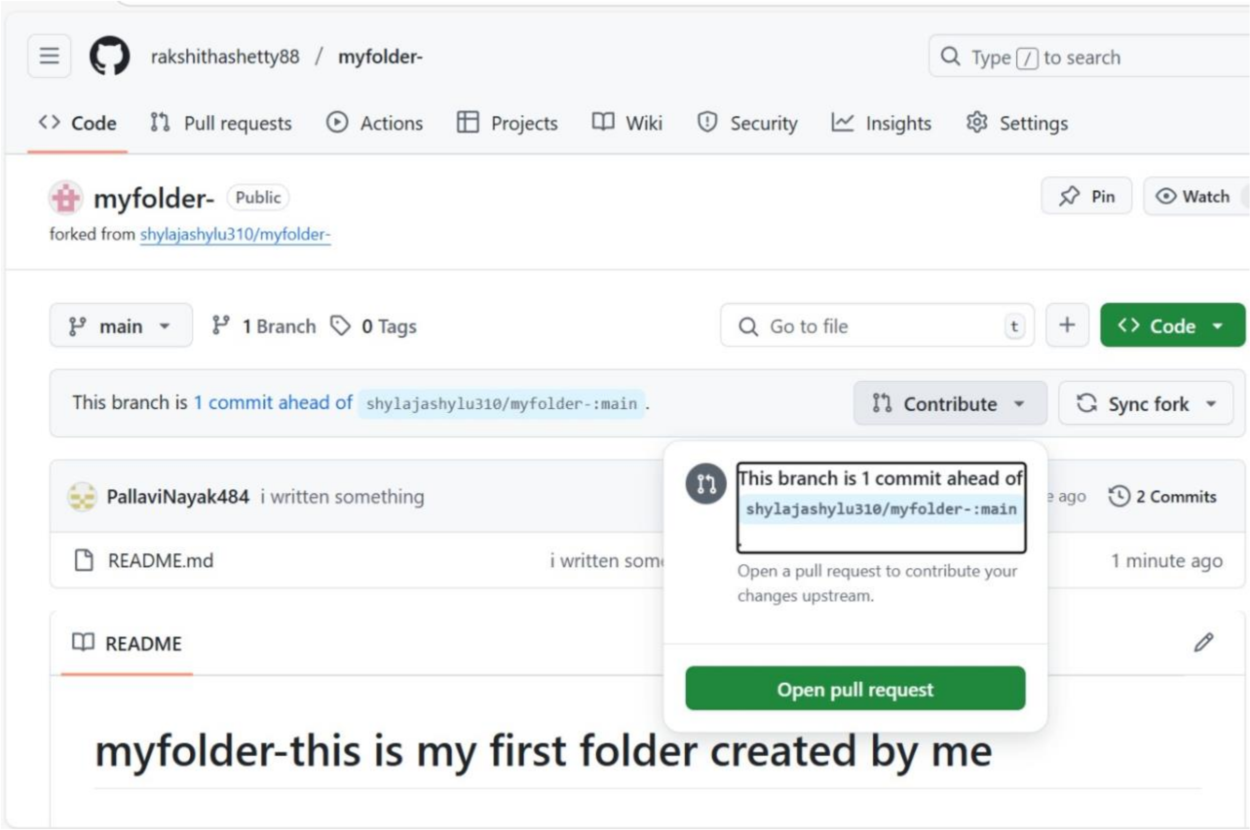


```
Dell@Rakshitha4086 MINGW64 ~ (master)
$ git clone https://github.com/rakshithashetty88/myfolder-.git
Cloning into 'myfolder-'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```



```
Dell@Rakshitha4086 MINGW64 /c
$ cd myfolder-

Dell@Rakshitha4086 MINGW64 /c/myfolder- (main)
$ ls




Dell@Rakshitha4086 MINGW64 /c/myfolder- (main)
$ vi README.md

Dell@Rakshitha4086 MINGW64 /c/myfolder- (main)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

Dell@Rakshitha4086 MINGW64 /c/myfolder- (main)
$ git commit -m"i written something"
[main 8404c76] i written something
 1 file changed, 1 insertion(+), 1 deletion(-)

Dell@Rakshitha4086 MINGW64 /c/myfolder- (main)
$ git remote
origin

Dell@Rakshitha4086 MINGW64 /c/myfolder- (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/rakshithashetty88/myfolder-.git
   3a78aea..8404c76  main -> main
```

In the Git Hub account contribute to open the pull request:



forked from shylajashylu310/myfolder-

This branch is 1 commit ahead of shylajashylu310/myfolder-:main .

PallaviNayak484 i written something

This branch is 1 commit ahead of shylajashylu310/myfolder-:main

Open a pull request to contribute your changes upstream.

myfolder-this is my first folder created by me

Steps to Close a Pull Request on GitHub:

1. Go to the repository on GitHub.

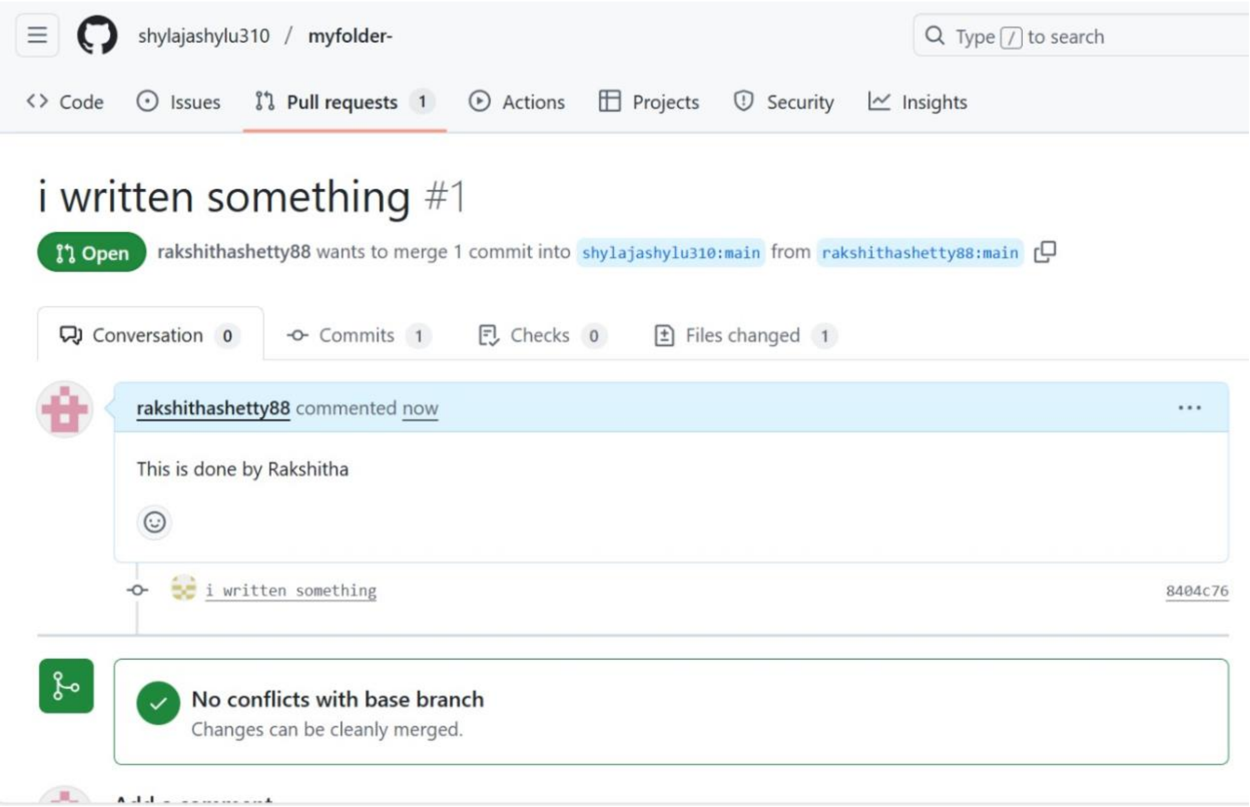
2. Click on the “Pull requests” tab

3. Find the pull request you want to close and click on

It.

4. Scroll to the bottom of the PR page.

5. Click the “Close pull request” button.



i written something #1

Open rakshithashetty88 wants to merge 1 commit into shylajashylu310:main from rakshithashetty88:main

rakshithashetty88 commented now

This is done by Rakshitha

i written something

No conflicts with base branch
Changes can be cleanly merged.

Add a comment

Experiment 08:

- Complete Git Process.

1. Install Git Bash

- Download Git for Windows from Git’s official

Website.

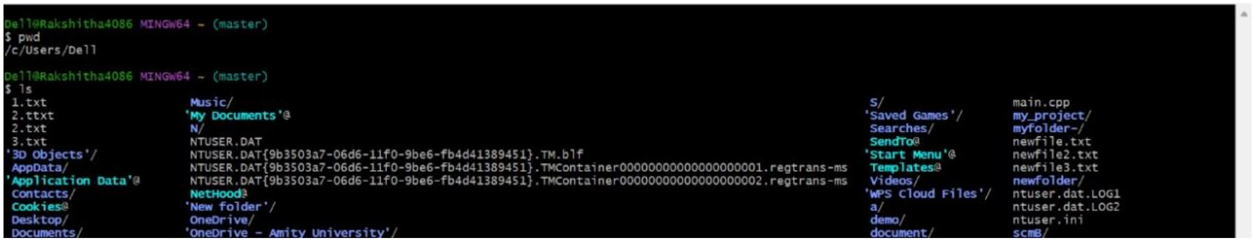
• Run the installer and follow the setup

Instructions.

• Choose Git Bash as the default terminal option.

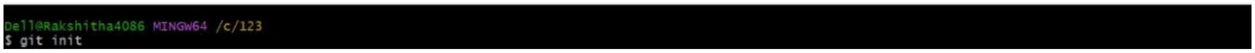
2.Initialize a Repository

• Open Git Bash and navigation to our project folder.



```
Dell@Rakshitha4086 MINGW64 ~ (master)
$ pwd
/c/Users/Dell

Dell@Rakshitha4086 MINGW64 ~ (master)
$ ls
 1.txt                    Music/                                                                                          S/                  main.cpp
 2.ttxt                  'My Documents'@                                                                                 'Saved Games'/       my_project/
 2.txt                    N/                                                                                              Searches/           myfolder-/
 3.txt                    NTUSER.DAT                                                                                      SendTo@             newfile.txt
'3D Objects'/             NTUSER.DAT{9b3503a7-06d6-11f0-9be6-fb4d41389451}.TM.blf                                        'Start Menu'@        newfile2.txt
 AppData/                 NTUSER.DAT{9b3503a7-06d6-11f0-9be6-fb4d41389451}.TMContainer00000000000000000001.regtrans-ms    Templates@          newfile3.txt
'Application Data'@       NTUSER.DAT{9b3503a7-06d6-11f0-9be6-fb4d41389451}.TMContainer00000000000000000002.regtrans-ms    Videos/             newfolder/
 Contacts/                NetHood@                                                                                       'WPS Cloud Files'/   ntuser.dat.LOG1
 Cookies@                'New folder'/                                                                                    a/                  ntuser.dat.LOG2
 Desktop/                 OneDrive/                                                                                       demo/               ntuser.ini
 Documents/              'OneDrive - Amity University'/                                                                   document/           scmB/
```

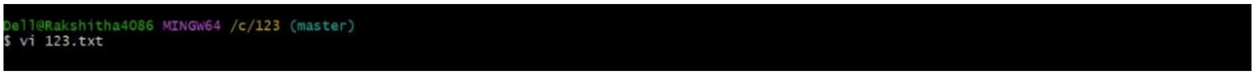
• Initialize a new git repository



```
Dell@Rakshitha4086 MINGW64 /c/123
$ git init
```

3.Create and Modify Files

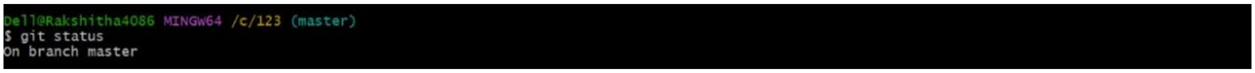
- Create a file using vi command:



```
Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ vi 123.txt
```

4.Check Repository Status

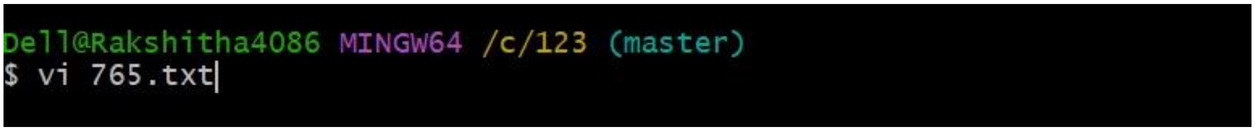
- View changes



```
Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ git status
On branch master
```

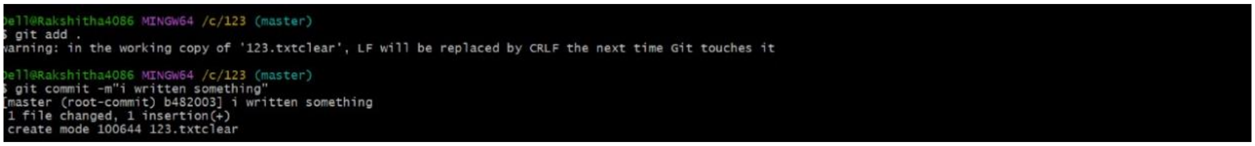
6. Stage and Commit Changes

- Add files to the staging area



```
Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ vi 765.txt
```

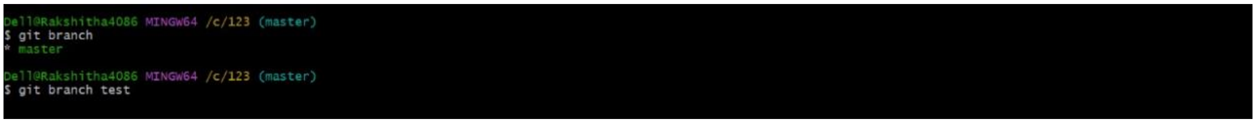
- Commit changes:



```
Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ git add .
warning: in the working copy of '123.txtclear', LF will be replaced by CRLF the next time Git touches it

Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ git commit -m"i written something"
[master (root-commit) b482003] i written something
 1 file changed, 1 insertion(+)
 create mode 100644 123.txtclear
```

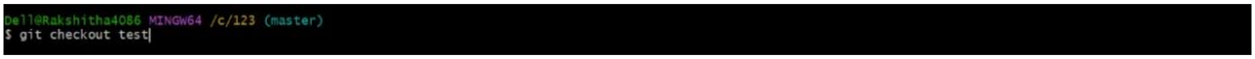
7. Create and Manage Branches

- Create a new branch:

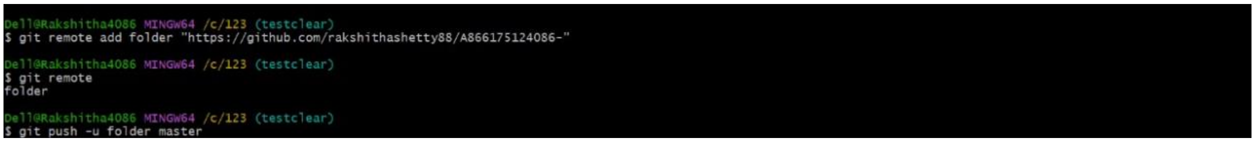


```
Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ git branch
* master

Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ git branch test
```

- Switch to the branch:

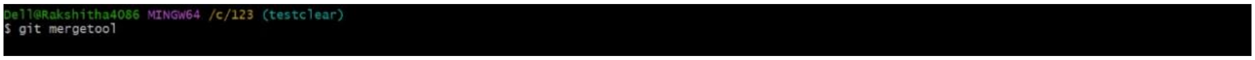


```
Dell@Rakshitha4086 MINGW64 /c/123 (master)
$ git checkout test
```

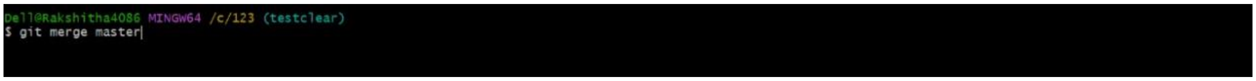


```
Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git remote add folder "https://github.com/rakshithashetty88/A866175124086-"

Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git remote
folder

Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git push -u folder master
```

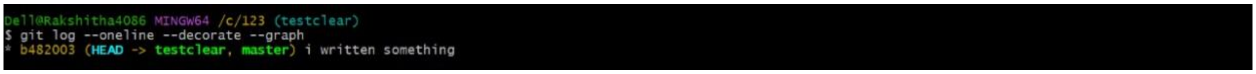
- Merge the branch into the main branch:



```
Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git mergetool
```

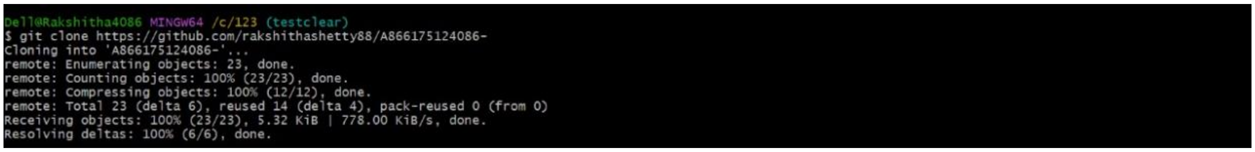


```
Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git merge master
```



```
Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git log --oneline --decorate --graph
* b482003 (HEAD -> testclear, master) i written something
```

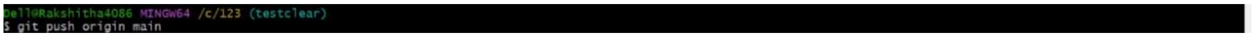
8. Git Clone:



```
Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git clone https://github.com/rakshithashetty88/A866175124086-
Cloning into 'A866175124086-'...
remote: Enumerating objects: 23, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 23 (delta 6), reused 14 (delta 4), pack-reused 0 (from 0)
Receiving objects: 100% (23/23), 5.32 KiB | 778.00 KiB/s, done.
Resolving deltas: 100% (6/6), done.
```

- Update local repository:

9. Pull Changes from Remote Repository

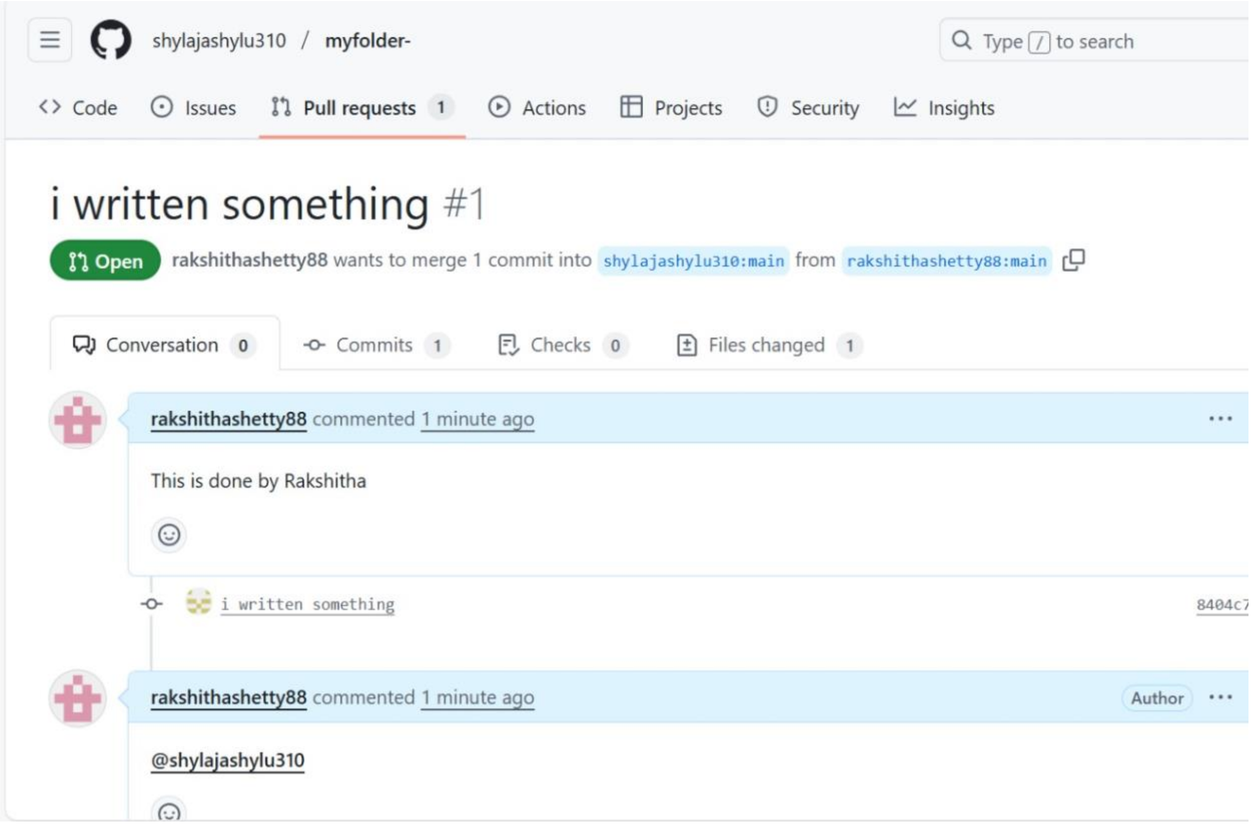


```
Dell@Rakshitha4086 MINGW64 /c/123 (testclear)
$ git push origin main
```

10.Open and Close Pull Requests

- Open a Pull Request on GitHub and merge changes.

- Close the Pull Request if needed.



i written something #1

Open rakshithashetty88 wants to merge 1 commit into `shylajashylu310:main` from `rakshithashetty88:main`

rakshithashetty88 commented 1 minute ago

This is done by Rakshitha

`i written something` 8404c7

rakshithashetty88 commented 1 minute ago

@shylajashylu310