

Source Code Lab

Name: Tejashwini S

Branch: B.Tech

Section: CSE

Course: Source Code Management

SEN: A86605223035

Submitted To: Dr. Monit Kapoor

Date of Submission: 06 June 2025

INDEX

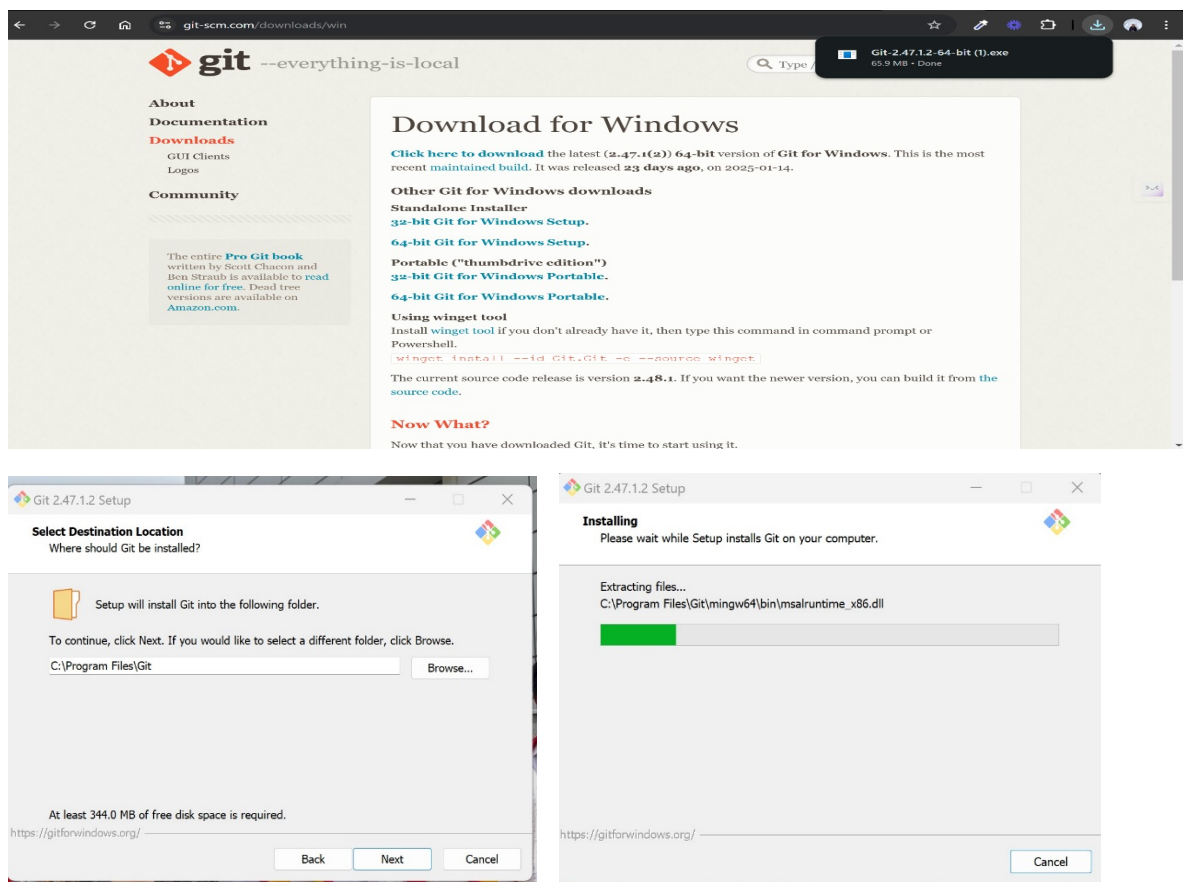
<u>Sl</u>	<u>Topic</u>	<u>Date</u>	<u>Pg No</u>
1	Git & GitHub Fundamentals- Install Git & GitHub- Basic Linux Commands	31-01- 2025	3-6
2	Committing & Viewing History- Understanding Commits- Viewing Git Logs	14-02- 2025	7-9
3	Branching & Version Comparison- Creating & Managing Branches- Using git diff	07-03- 2025	10-12
4	Remote Repositories- Pushing Local Repos to GitHub	16-05- 2025	13-14
5	Collaboration & Conflict Resolution- Inducing & Resolving Conflicts- Opening & Closing Pull Requests	21-05- 2025	15-16

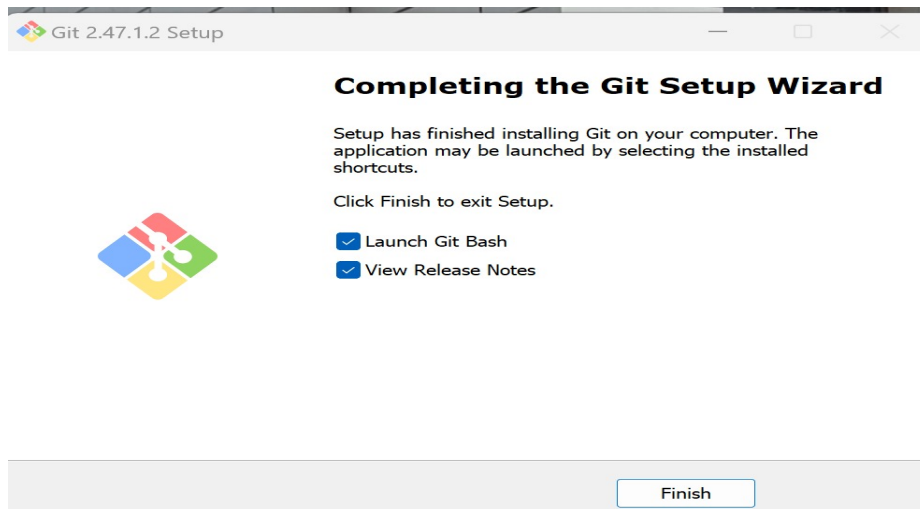
SOURCE CODE MANAGEMENT

LAB - 1

- INSTALLING GIT

1. <https://git-scm.com/downloads/win>





COMMANDS

1. *pwd*

- *Present Working directory*
- *Used to view the current directory working on*

2. *cd*

- *change directory*
- *used to change the directory*

3. *ls*

- *used to list all folders*

```

MINGW64:/c/Users/DELL
DELL@DESKTOP-OCHS1GT MINGW64 ~
$ cd

DELL@DESKTOP-OCHS1GT MINGW64 ~
$ pwd
/c/Users/DELL

DELL@DESKTOP-OCHS1GT MINGW64 ~
$ ls
'3D Objects'/'
AppData/'
'Application Data'@
Contacts/'
Cookies@
Documents/'
Downloads/'
Favorites/'
IntelGraphicsProfiles/'
Links/'
'Local Settings'@
Music/'
'My Documents'@
NTUSER.DAT
NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.0.regtrans-ms
NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.1.regtrans-ms
NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.2.regtrans-ms
NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.b1f
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.b1f
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000000000000
1.regtrans-ms
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000000000000
2.regtrans-ms
NetHood@
OneDrive/'
PrintHood@
Recent@
'Saved Games'/'
Searches/'
SendTo@
'Start Menu'@
Templates@
Videos/'
'WPS Cloud Files'/'
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini

```

4. vi

- vim mode / vi editor
- used to create a file, edit it
- Syntax: vi nameoffile (Ex: vi hello.txt)

click the key *i* to insert or edit your file. Then *esc* to return from insert mode :wq (then enter) to return

5. cat

- Syntax: `cat filename` (`cat hello.txt`)
- to view the file contents.

```
DELL@DESKTOP-0CHS1GT MINGW64 ~  
$ vi hello.txt  
  
DELL@DESKTOP-0CHS1GT MINGW64 ~  
$ cat hello.txt  
hello, this is a word file.!!!
```

MINGW64:/c/Users/DELL

```
hello, this is a word file.!!!  
~  
~  
~  
~  
~  
~  
~  
~
```

LAB - 2

6. ls -l

- Used to list all files

```
DELL@DESKTOP-0CHS1GT MINGW64 ~
$ ls -l
total 18118
drwxr-xr-x 1 DELL 197121      0 Nov  2  2024 '3D objects'/
drwxr-xr-x 1 DELL 197121      0 Nov  2  2024 AppData/
lrwxrwxrwx 1 DELL 197121     29 Nov  2  2024 'Application Data' -> /c/Users/DELL/AppData/Roaming/
drwxr-xr-x 1 DELL 197121      0 Nov  2  2024 Contacts/
lrwxrwxrwx 1 DELL 197121     57 Nov  2  2024 Cookies -> /c/
/Users/DELL/AppData/Local/Microsoft/Windows/INetCookies/
drwxr-xr-x 1 DELL 197121      0 Nov 15  2024 Documents/
drwxr-xr-x 1 DELL 197121      0 Jun  7 21:27 Downloads/
drwxr-xr-x 1 DELL 197121      0 Nov  2  2024 Favorites/
drwxr-xr-x 1 DELL 197121      0 May 21 21:15 IntelGraphics
Profiles/
drwxr-xr-x 1 DELL 197121      0 Nov  2  2024 Links/
lrwxrwxrwx 1 DELL 197121     27 Nov  2  2024 'Local Settings' -> /c/Users/DELL/AppData/Local/
drwxr-xr-x 1 DELL 197121      0 Nov  2  2024 Music/
lrwxrwxrwx 1 DELL 197121     23 Nov  2  2024 'My Documents'
-> /c/Users/DELL/Documents/
-rw-r--r-- 1 DELL 197121 10223616 May 21 21:15 NTUSER.DAT
-rw-r--r-- 1 DELL 197121 1048576 Jun  7 21:41 NTUSER.DAT{53
b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.0.regtrans-ms
-rw-r--r-- 1 DELL 197121 1048576 May 21 21:15 NTUSER.DAT{53
b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.1.regtrans-ms
-rw-r--r-- 1 DELL 197121 1048576 May 21 21:15 NTUSER.DAT{53
b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.2.regtrans-ms
```

7. ls -ah

8. history

- used to view history of commands

```
DELL@DESKTOP-0CHS1GT MINGW64 ~
$ history
 1  pwd
 2  cs
 3  cd
 4  pwd
 5  ls
 6  vi hello.txt
 7  vi hello.txt
 8  cat hello.txt
 9  ls -l
10  ls -ah
11  ls -ah
12  history
```

9. mkdir

- used to create folder
- Syntax: mkdir nameoffolder

➤ ls nameoffolder

- confirmation to check if folder is created

➤ cd folder/

- to change the present directory as the folder

```
DELL@DESKTOP-0CHS1GT MINGW64 ~  
$ mkdir firstproject  
  
DELL@DESKTOP-0CHS1GT MINGW64 ~  
$ cd firstproject/
```

- also after this create one new file by using vi (then insert i)

10. git config

- Used to configure username and email
- Syntax:
git config --global user.mail "mail"
git config --global user.name "username"

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject  
$ git config --global user.mail "shikkukoushikyadav@gmail.com"  
  
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject  
$ git config --global user.name "Koushik-0211"
```

11. git init

- initialize the repository (important step then only folder will get watched)
- working directory to empty repo.

12. *git add file*

- to add the file
- Syntax: *git add filename*
- After this edit the file (using vi, insert i)

13. *git commit*

- Used to commit a change
- Syntax: *git commit -m "Message"*

14. *git log*

- Used to view status / entries of commits in that branch

15. *git status*

- To view status of the branch

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git commit -m "ADDED TXT FILE"
[master (root-commit) 189609e] ADDED TXT FILE
1 file changed, 1 insertion(+)
create mode 100644 hello.txt

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git status
On branch master
nothing to commit, working tree clean

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git log
commit 189609e0254f509de63f3b67516e05ddafd99f56 (HEAD -> master)
Author: Koushik-0211 <shikkukoushikyadav@gmail.com>
Date: Sat Jun 7 21:52:08 2025 +0530

    ADDED TXT FILE
```

LAB - 3

16. `git log --oneline`

- To view log statements at once

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git log --oneline
1b5acda (HEAD -> master) One more line added
189609e ADDED TXT FILE
```

17. `git diff commit1 commit2`

- By taking numbers of the commit which will be shown in the git log – oneline command, the commits are compared in terms of what is added and what is removed.

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git diff 1b5acda 189609e
diff --git a/hello.txt b/hello.txt
index 41d2353..6d0b012 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,3 +1 @@
 hello, hi!!!!!!
 -how r u doing?
 -

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git diff 189609e 1b5acda
diff --git a/hello.txt b/hello.txt
index 6d0b012..41d2353 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1 +1,3 @@
 hello, hi!!!!!!
 +how r u doing?
 +
```

18. `git add .`

- To add multiple files to repo at once

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git add .
```

19. `git branch <name>`

- It is used to create a new branch other than the master branch
- Syntax: `git branch nameofbranch` (Example: `git branch test`)

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git branch test1
```

20. `git branch`

- Used to see the different branches in the repo

21. `git checkout branchname`

- Used to move to that branch.
-

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git branch test2

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git branch
* master
  test1
  test2

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git checkout test2
Switched to branch 'test2'

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (test2)
$ |
```

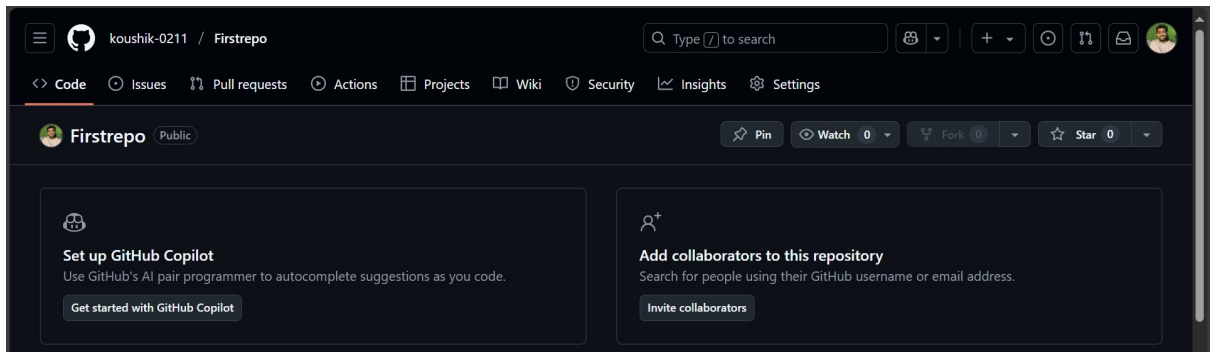
22. `git reflog`

- Used to see all the commits present in all the branches.
- Even `git log` works

- 2 different branches will have 2 different codes for the same file. Changes done in one branch will not be reflected in changes done in the other branch.

LAB - 4

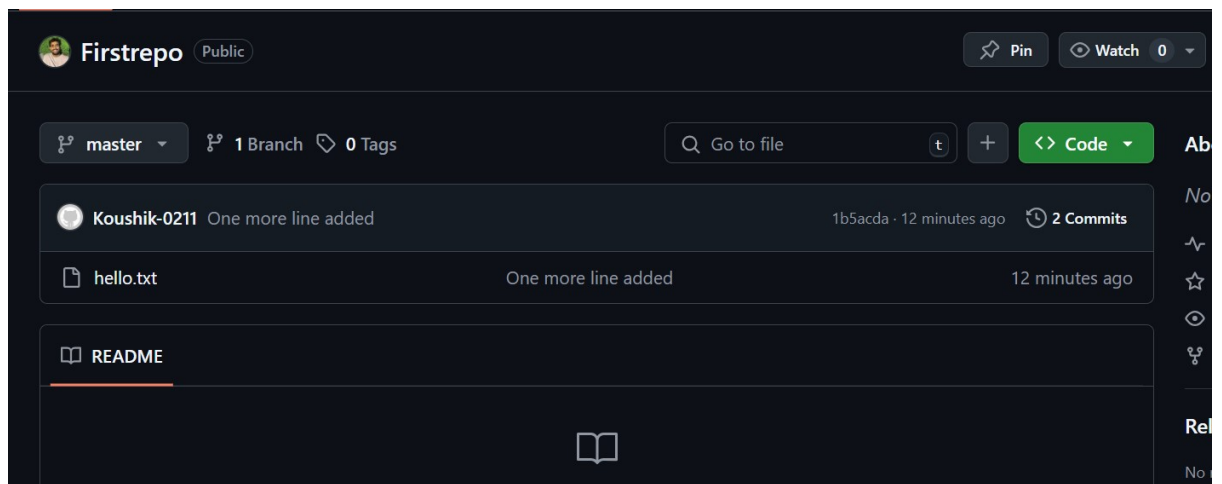
23. *Git remote add origin <url of the repository on github.com>*
- *For that create a repository on github.com after logging on to the website. Use the same id / email address which was used to config in git bash. Create a new repository by giving a name to the repository.*



Copy the url of the repository and using the command `git remote add origin <link of repo>` add the repo on gitbash

24. *git push -u origin <branch>*
- *Used to push the branch to the repo on Github.*
 - *Use the command `git push -u origin <nameofbranch>` to add it, for the first time authentication will be required then when the page is refreshed that branch and all the files in it gets added.*

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 495 bytes | 55.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/koushik-0211/Firstrepo.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```



```
switched to branch 'test2'

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (test2)
$ git push -u origin test2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'test2' on GitHub by visiting:
remote:   https://github.com/koushik-0211/Firstrepo/pull/new/test2
remote:
To https://github.com/koushik-0211/Firstrepo.git
 * [new branch]      test2 -> test2
branch 'test2' set up to track 'origin/test2'.

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (test2)
$ |
```

LAB-5

1. Induce Conflicts, Open and Close Pull Requests

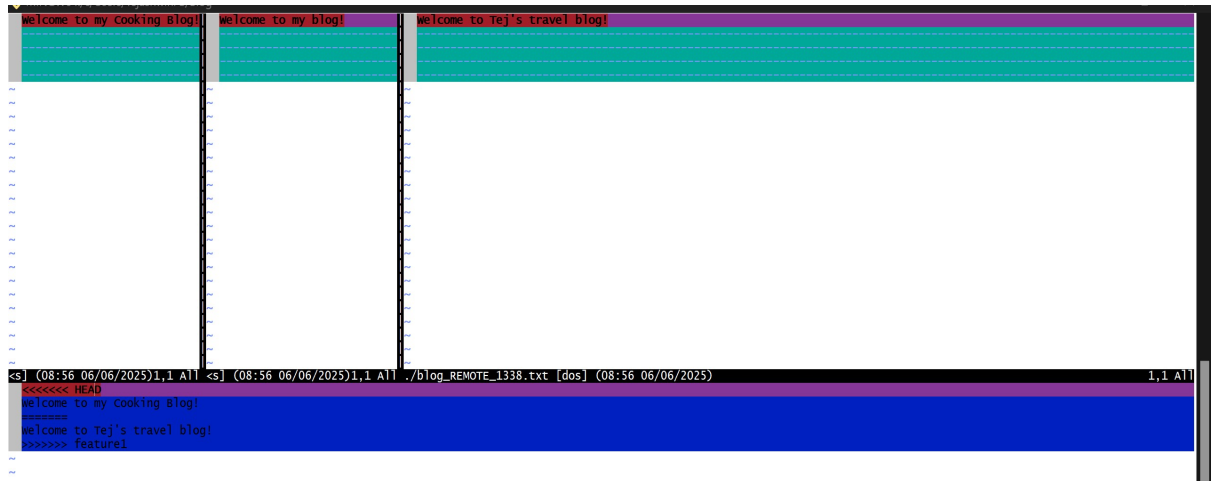
1. Git Merging & Conflicts:

- * `git merge`: Integrate changes from one branch into another.
- * Conflicts arise during merging when Git cannot automatically reconcile diverging changes.
- * Use a merge tool to resolve conflicts.
- * `git mergetool`: Command to launch the configured merge tool.

```
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ vi hello.txt

DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git add hello.txt

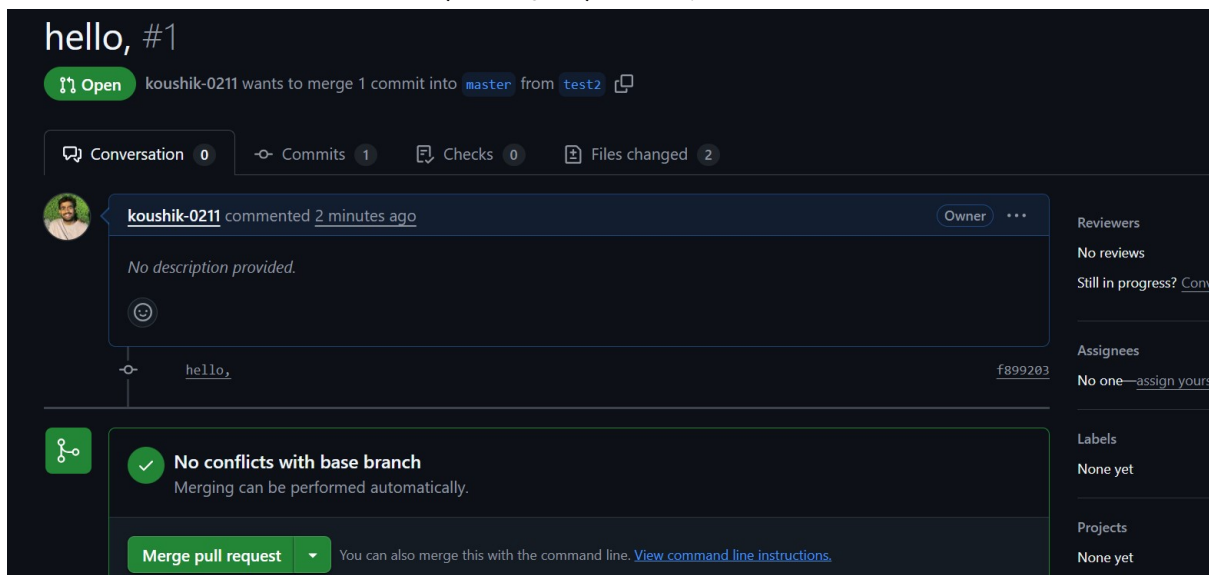
DELL@DESKTOP-0CHS1GT MINGW64 ~/firstproject (master)
$ git commit -m "changed txt file"
[master 5662a16] changed txt file
1 file changed, 1 insertion(+), 1 deletion(-)
```



2. **Gitignore:** *. gitignore file: Specifies intentionally untracked files that Git should ignore.

* Files listed in .gitignore will not be tracked by Git and will not be mirrored on GitHub.

2. Pull Requests: Opening a pull request



2. * Closing a pull request

When we add a new branch or any file, a pull request gets created. We need to merge pull request for it to get added after viewing and checking it.

