Name : Zainab Bibi

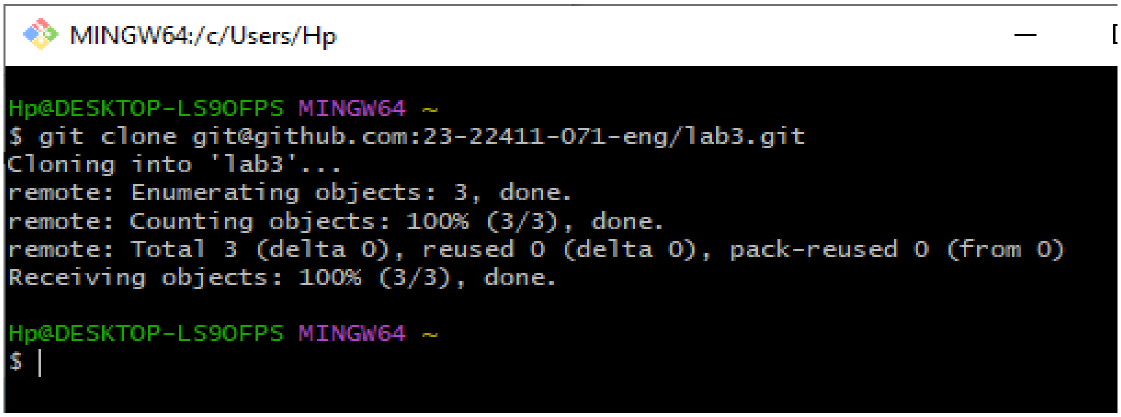
Reg. No : 2023-BSE-071

Class: BSE- 5B

Lab 03

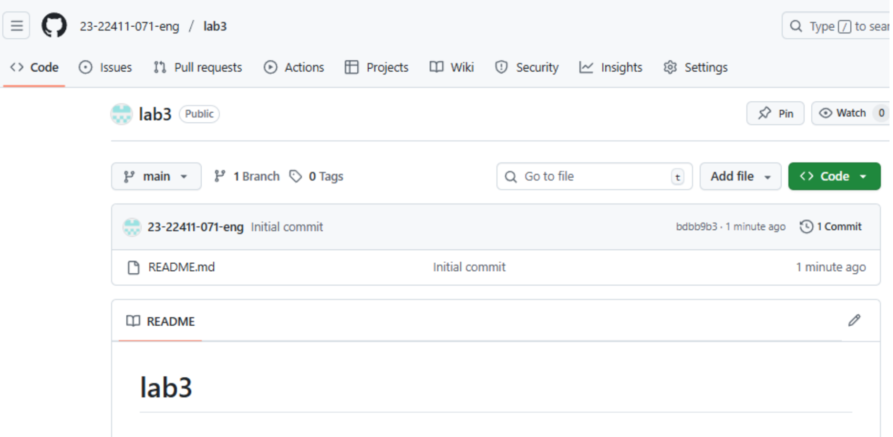
Task 1 :

This task demonstrates what happens when your local repository and remote repository both have new commits — and how git pull and git pull --rebase behave differently.

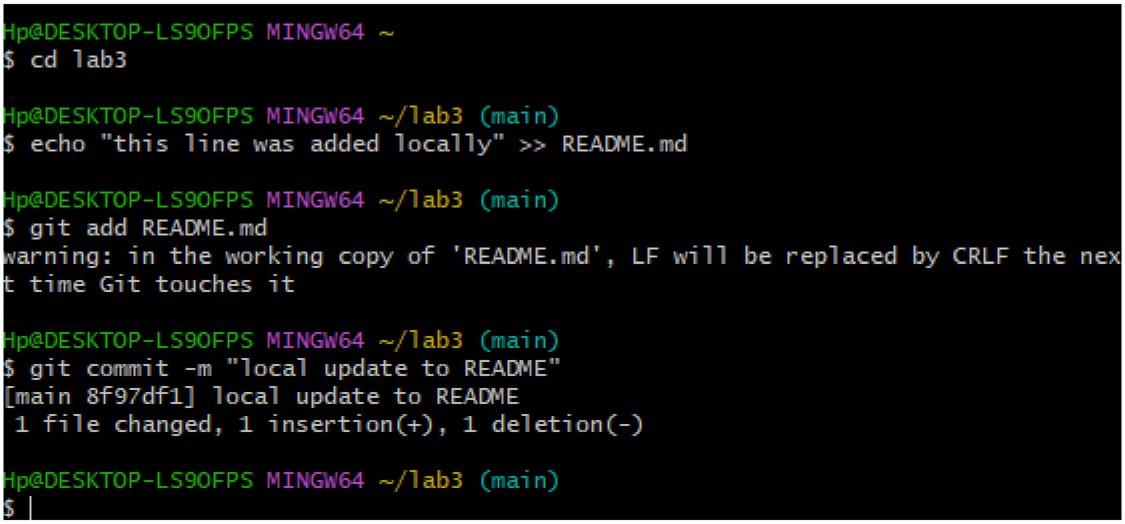


```
Hp@DESKTOP-LS9OFPS MINGW64 ~
$ git clone git@github.com:23-22411-071-eng/lab3.git
Cloning into 'lab3'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

Hp@DESKTOP-LS9OFPS MINGW64 ~
$ |
```

Open your repository on GitHub and edit the README.md file directly in the browser. Save a screenshot of your browser showing the change as remote_edit.png.

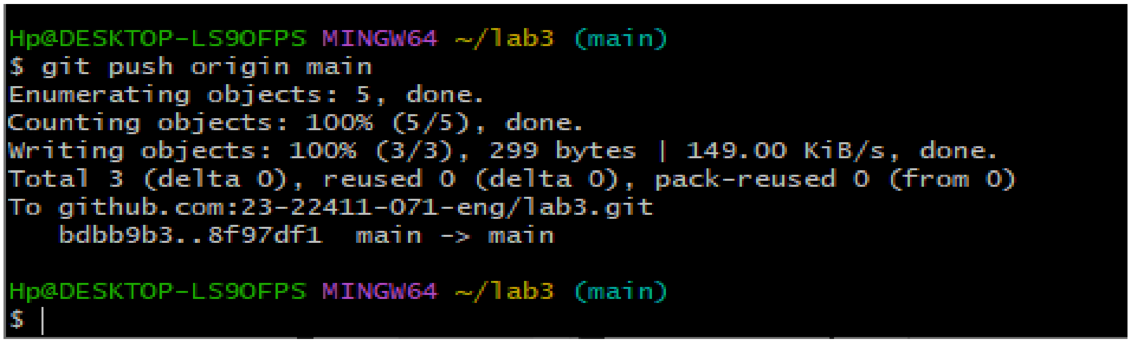


On your local machine, open the same repository folder. Edit the same README.md file locally.

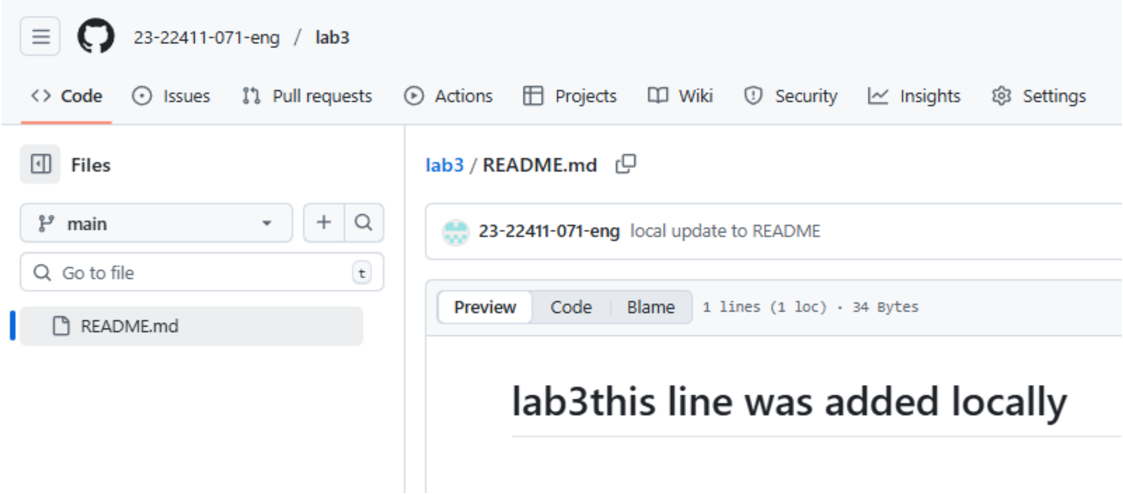


```
Hp@DESKTOP-LS9OFPS MINGW64 ~
$ cd lab3

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ echo "this line was added locally" >> README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ 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

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git commit -m "local update to README"
[main 8f97df1] local update to README
 1 file changed, 1 insertion(+), 1 deletion(-)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ |
```

Try to push:



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 299 bytes | 149.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:23-22411-071-eng/lab3.git
   bdbb9b3..8f97df1  main -> main

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ |
```



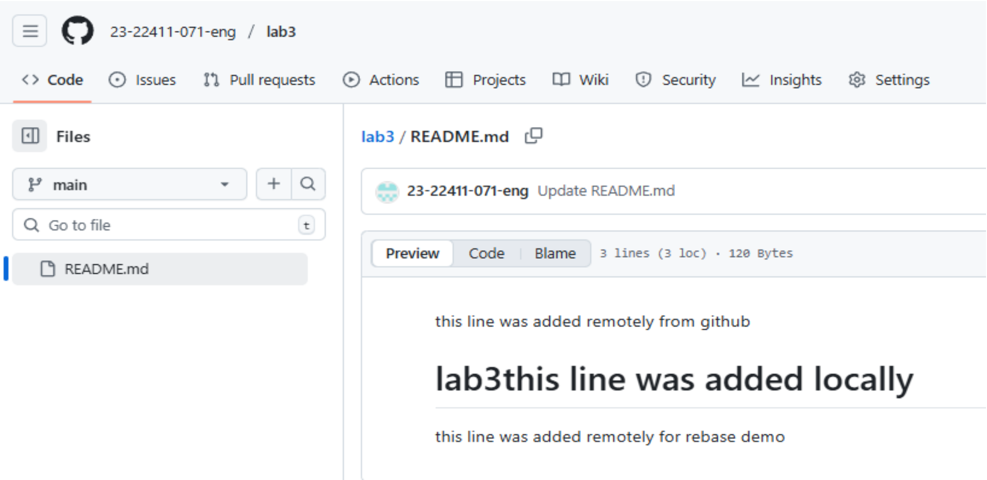
lab3 / README.md

23-22411-071-eng local update to README

1 lines (1 loc) · 34 Bytes

lab3this line was added locally

Now repeat the same process, but this time use rebase instead of merge.Make another remote edit on GitHub in README.md

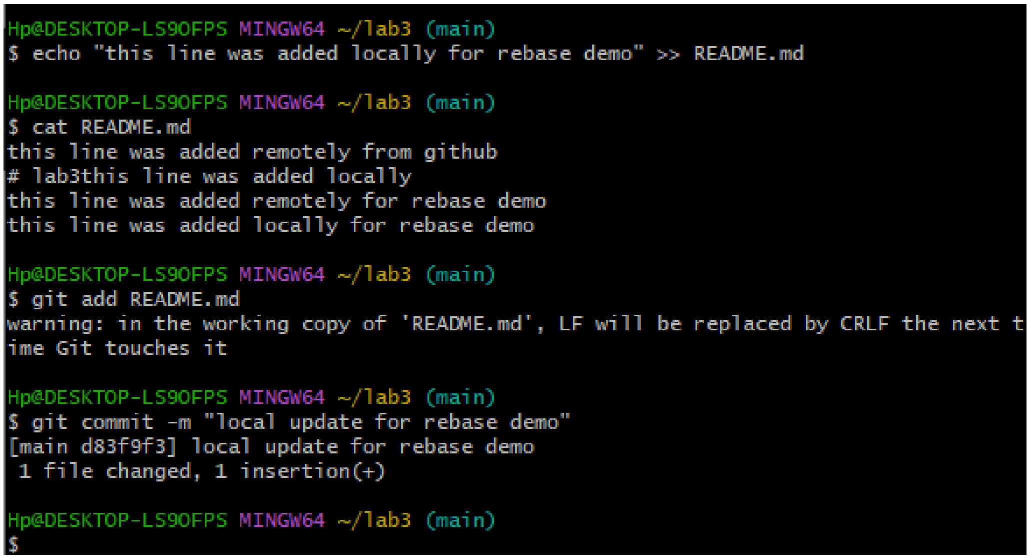


this line was added remotely from github

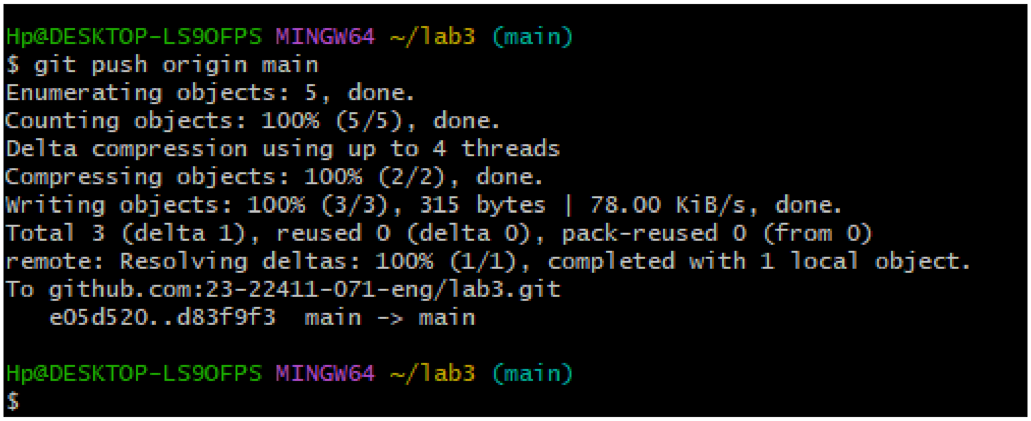
lab3this line was added locally

this line was added remotely for rebase demo

Then make another local edit and commit it.

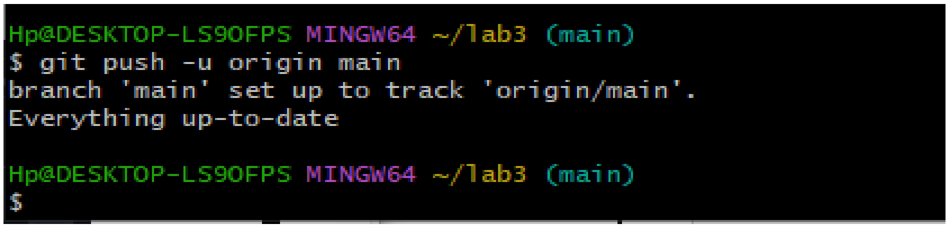


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ echo "this line was added locally for rebase demo" >> README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ cat README.md
this line was added remotely from github
# lab3this line was added locally
this line was added remotely for rebase demo
this line was added locally for rebase demo

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next t
ime Git touches it

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git commit -m "local update for rebase demo"
[main d83f9f3] local update for rebase demo
 1 file changed, 1 insertion(+)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$
```

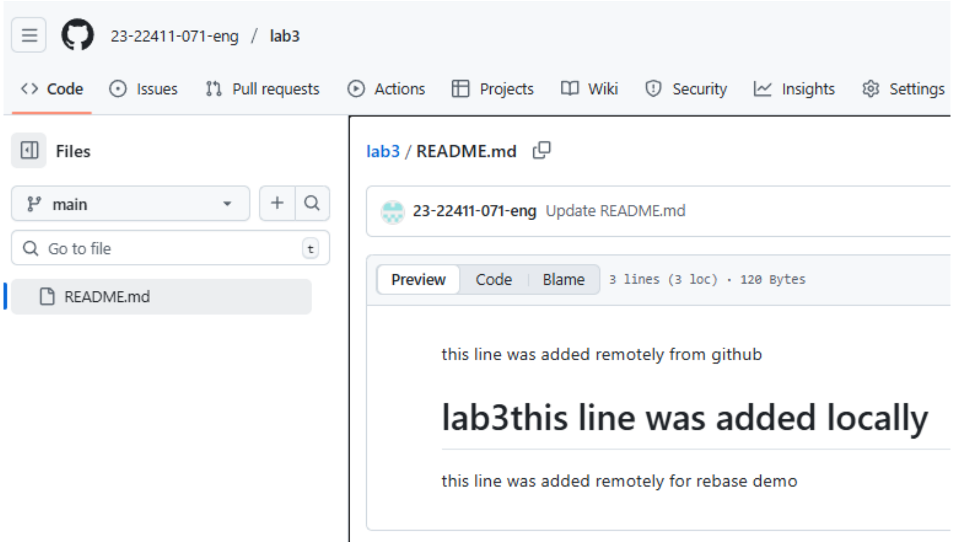


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 315 bytes | 78.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:23-22411-071-eng/lab3.git
   e05d520..d83f9f3  main -> main

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$
```

Then push



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$
```



23-22411-071-eng Update README.md

3 lines (3 loc) · 120 Bytes

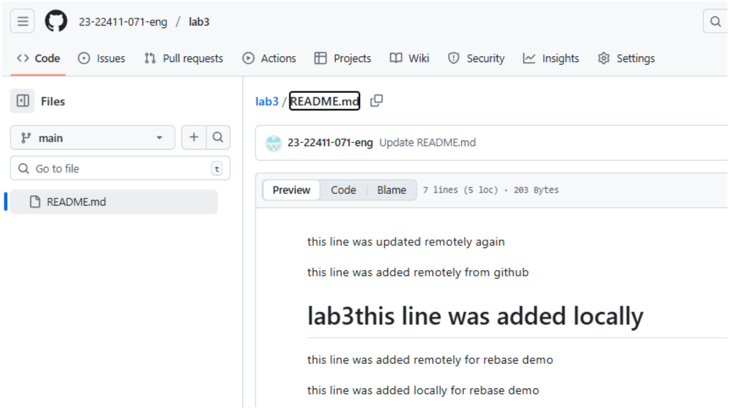
this line was added remotely from github

lab3this line was added locally

this line was added remotely for rebase demo

Task 2 :

This task will help you understand how Git handles file conflicts when two people modify the same line of code differently. On GitHub (remot e), open your README.md file and change an existing line.



23-22411-071-eng Update README.md

7 lines (5 loc) · 203 Bytes

this line was updated remotely again

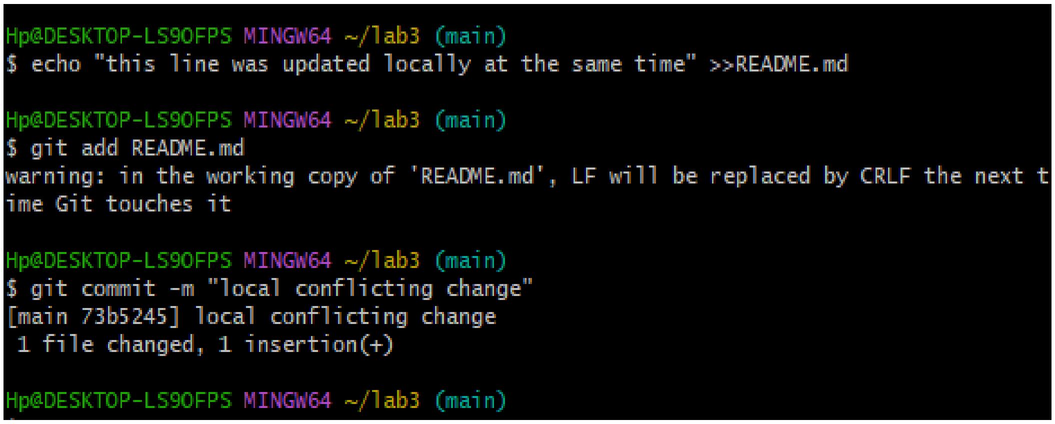
this line was added remotely from github

lab3this line was added locally

this line was added remotely for rebase demo

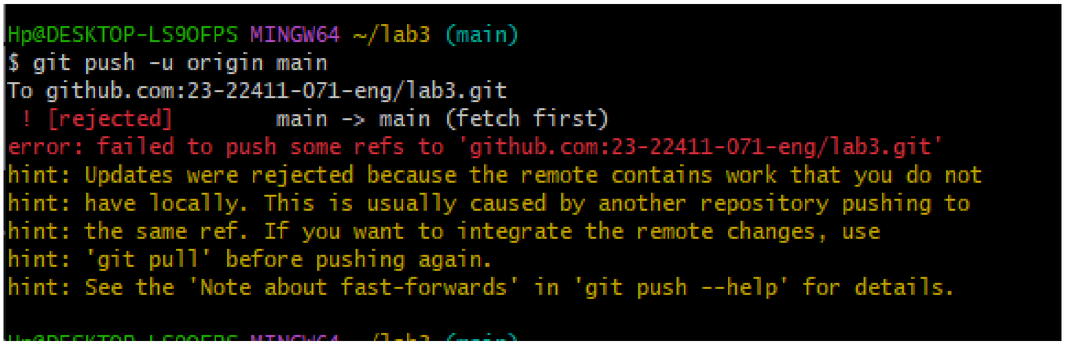
this line was added locally for rebase demo

On your local machine, edit the same line in the same file but make a different change.



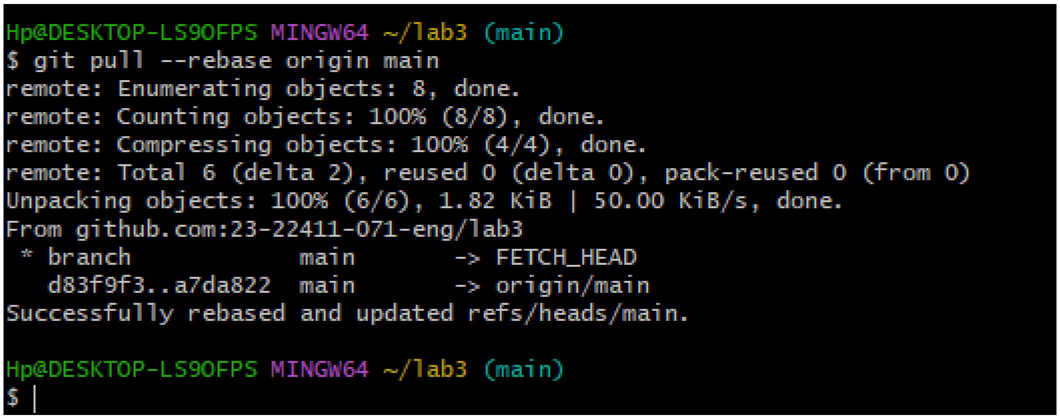
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ echo "this line was updated locally at the same time" >>README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next t
ime Git touches it

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git commit -m "local conflicting change"
[main 73b5245] local conflicting change
 1 file changed, 1 insertion(+)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
```

Try to push



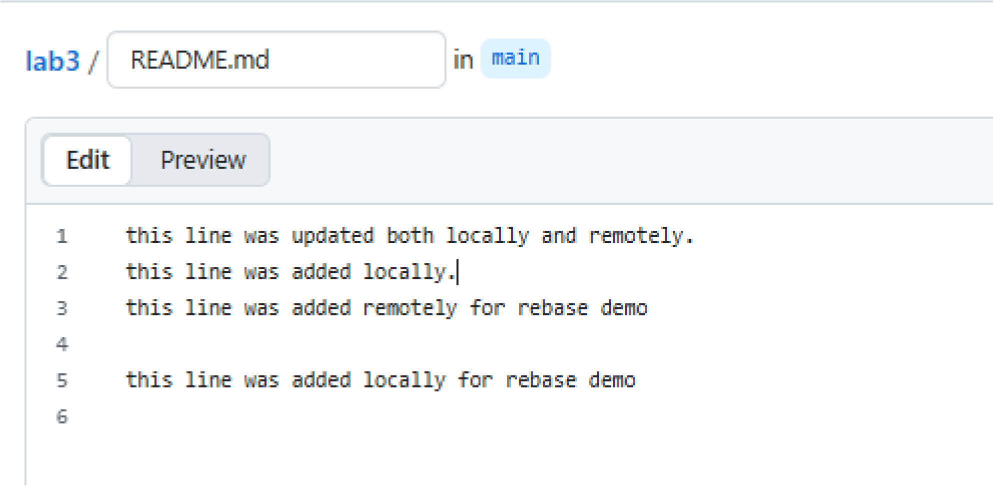
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push -u origin main
To github.com:23-22411-071-eng/lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:23-22411-071-eng/lab3.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Pull with rebase to bring in remote changes



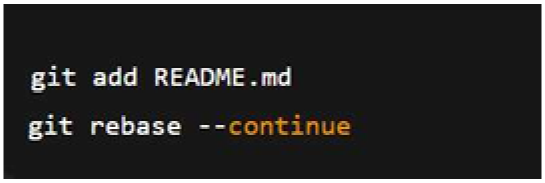
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git pull --rebase origin main
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 1.82 KiB | 50.00 KiB/s, done.
From github.com:23-22411-071-eng/lab3
 * branch            main       -> FETCH_HEAD
   d83f9f3..a7da822  main       -> origin/main
Successfully rebased and updated refs/heads/main.

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ |
```

Open the README.md file in your editor. Manually edit the file to keep the correct line (for example, merge both ideas).



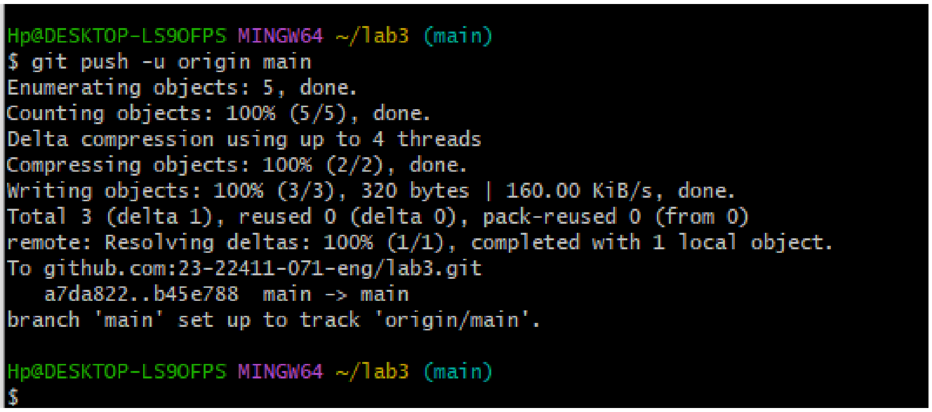
```
this line was updated both locally and remotely.
this line was added locally.
this line was added remotely for rebase demo

this line was added locally for rebase demo

```

After fixing the file, mark the conflict as resolved and continue the rebase.

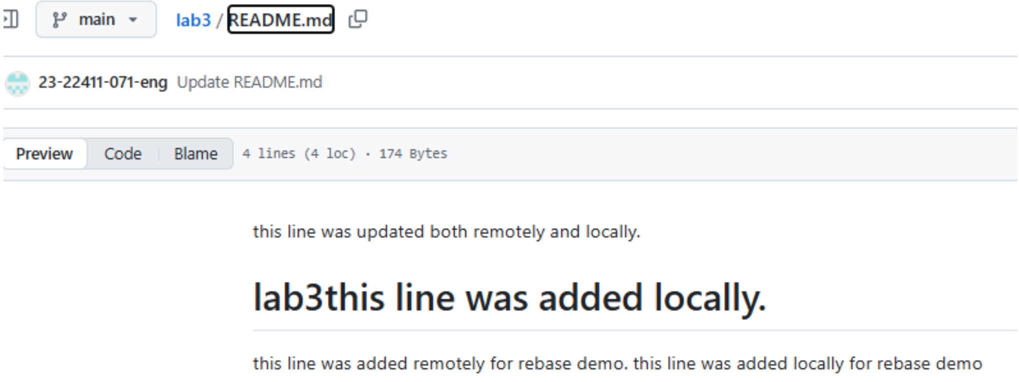


```
git add README.md
git rebase --continue
```

Finally, push your changes.



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 160.00 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:23-22411-071-eng/lab3.git
   a7da822..b45e788  main -> main
branch 'main' set up to track 'origin/main'.

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$
```



23-22411-071-eng Update README.md

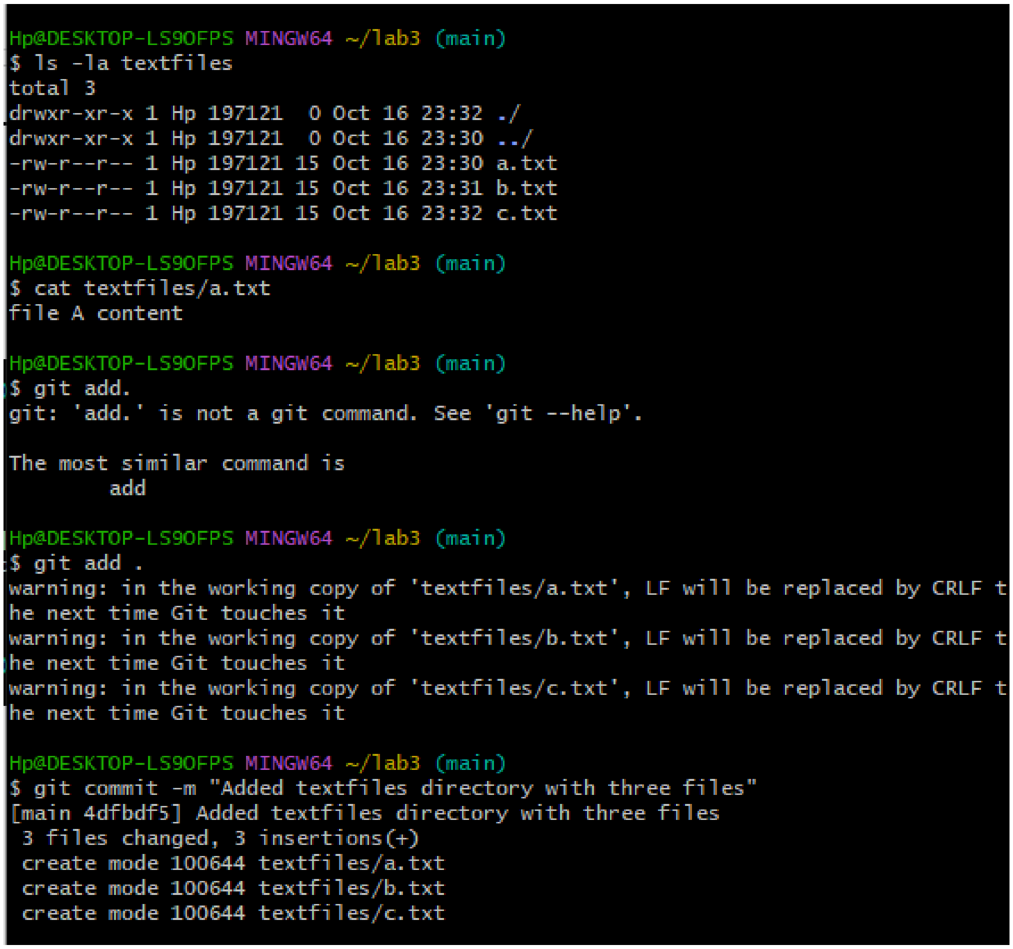
this line was updated both remotely and locally.

lab3this line was added locally.

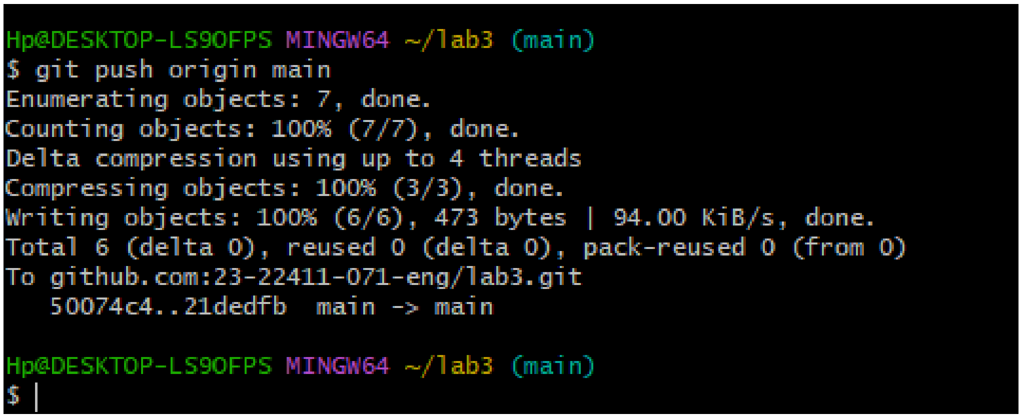
this line was added remotely for rebase demo. this line was added locally for rebase demo

Task 3 :

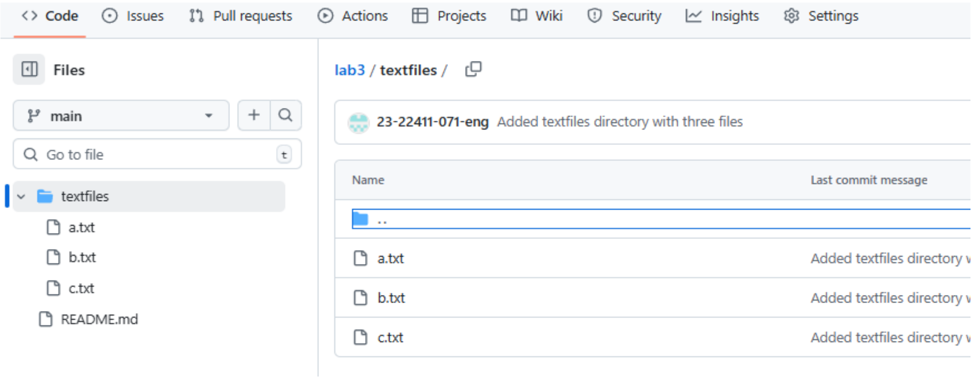
In this task, you will learn how to use a .gitignore file to exclude files or directories from version control and how to remove them from Git's tracking while keeping them locally. Create a new folder named textfiles inside your repository.



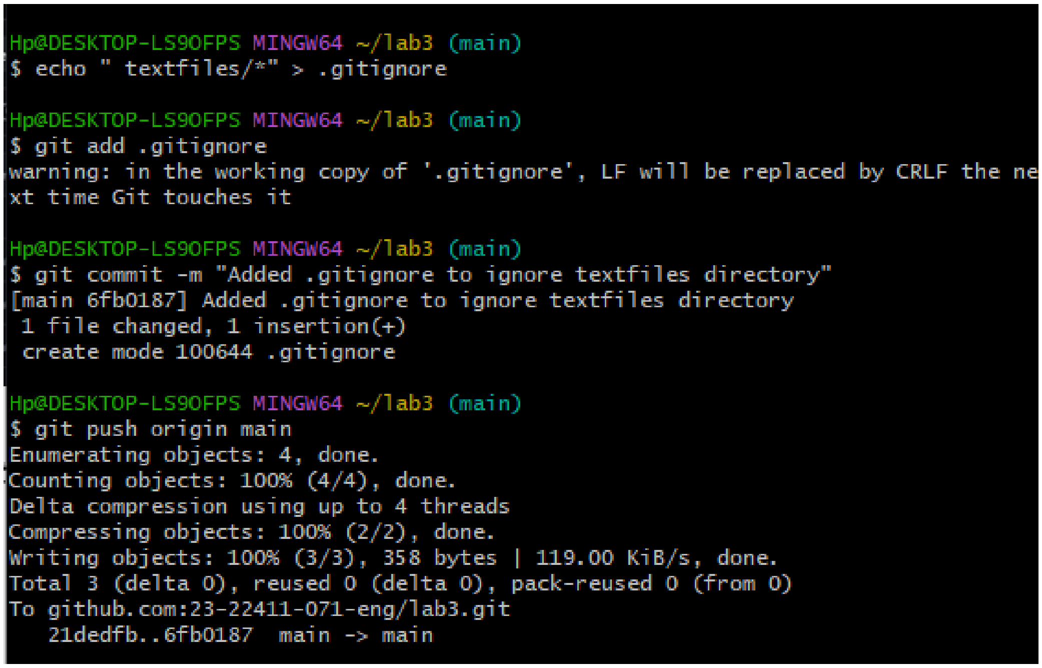
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ ls -la textfiles
total 3
drwxr-xr-x 1 Hp 197121  0 Oct 16 23:32 ./
drwxr-xr-x 1 Hp 197121  0 Oct 16 23:30 ../
-rw-r--r-- 1 Hp 197121 15 Oct 16 23:30 a.txt
-rw-r--r-- 1 Hp 197121 15 Oct 16 23:31 b.txt
-rw-r--r-- 1 Hp 197121 15 Oct 16 23:32 c.txt

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ cat textfiles/a.txt
file A content

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
        add

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git add .
warning: in the working copy of 'textfiles/a.txt', LF will be replaced by CRLF t
he next time Git touches it
warning: in the working copy of 'textfiles/b.txt', LF will be replaced by CRLF t
he next time Git touches it
warning: in the working copy of 'textfiles/c.txt', LF will be replaced by CRLF t
he next time Git touches it

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git commit -m "Added textfiles directory with three files"
[main 4dfbdf5] Added textfiles directory with three files
 3 files changed, 3 insertions(+)
 create mode 100644 textfiles/a.txt
 create mode 100644 textfiles/b.txt
 create mode 100644 textfiles/c.txt
```



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 473 bytes | 94.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:23-22411-071-eng/lab3.git
   50074c4..21dedfb  main -> main

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ |
```

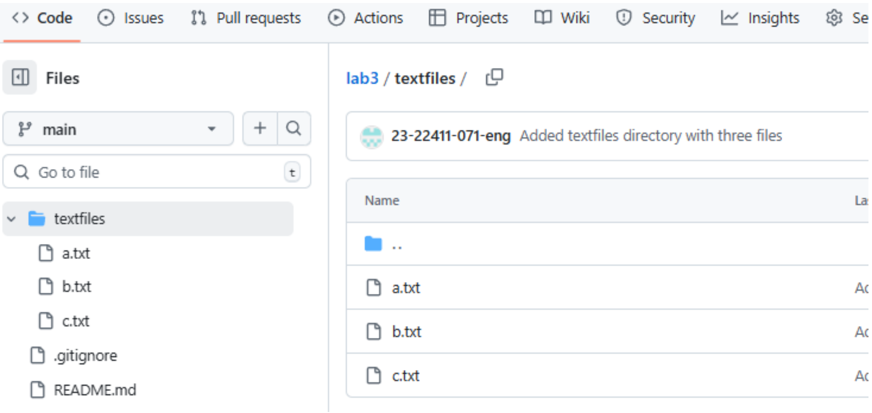


Create a .gitignore file in the root of your repository. Add textfiles to .gitignore. Commit and push.

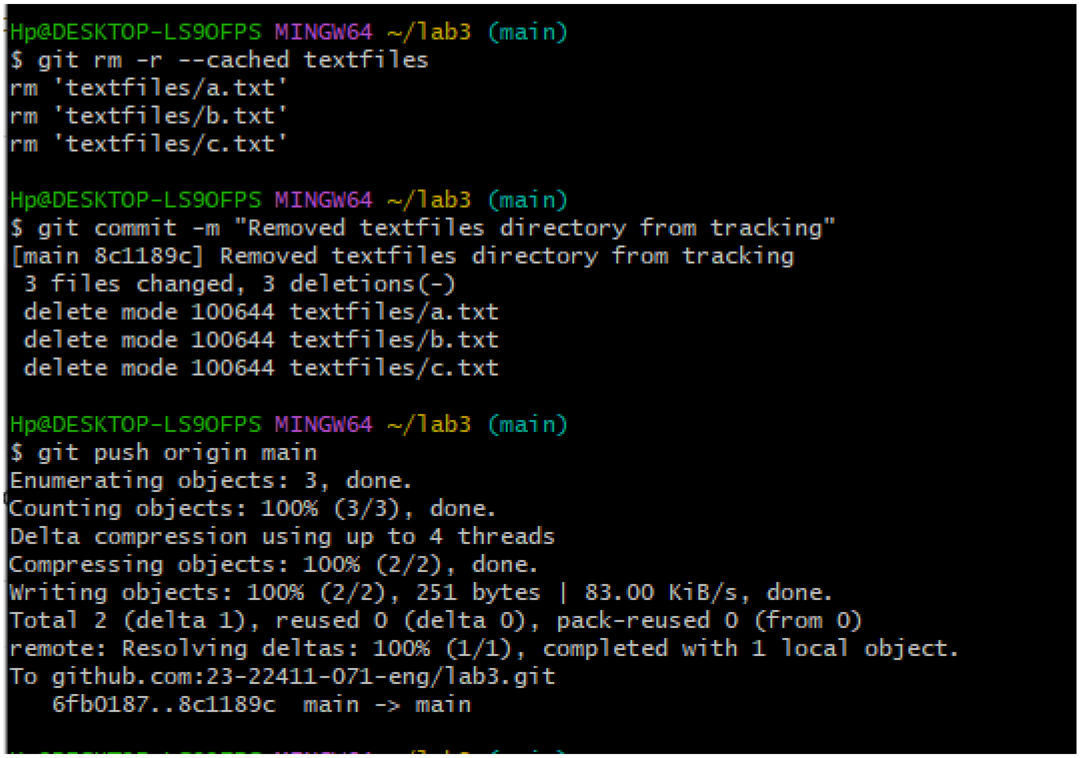


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ echo " textfiles/*" > .gitignore

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git commit -m "Added .gitignore to ignore textfiles directory"
[main 6fb0187] Added .gitignore to ignore textfiles directory
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 358 bytes | 119.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:23-22411-071-eng/lab3.git
   21dedfb..6fb0187  main -> main
```

Go to GitHub and check your repository — notice that the textfiles directory is still visible on the remote.

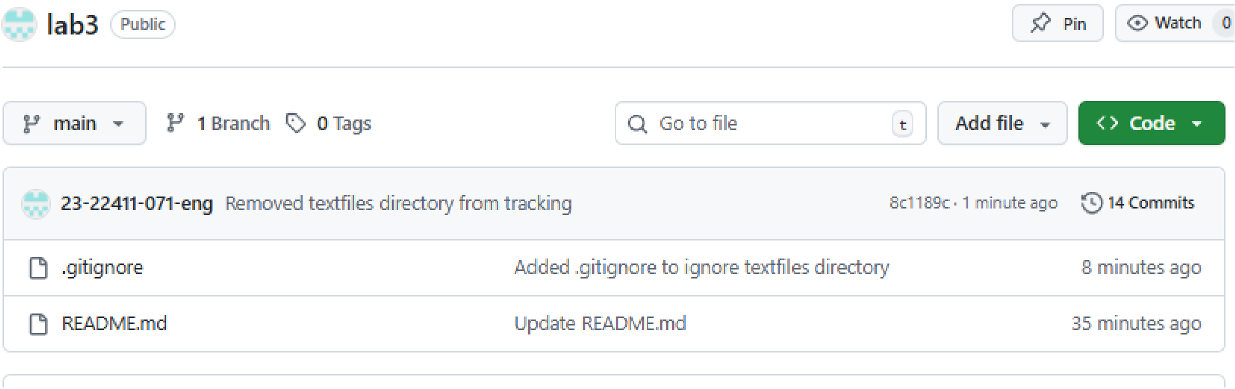


To remove already tracked files from Git (but not from your local system), run:



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git rm -r --cached textfiles
rm 'textfiles/a.txt'
rm 'textfiles/b.txt'
rm 'textfiles/c.txt'

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git commit -m "Removed textfiles directory from tracking"
[main 8c1189c] Removed textfiles directory from tracking
 3 files changed, 3 deletions(-)
 delete mode 100644 textfiles/a.txt
 delete mode 100644 textfiles/b.txt
 delete mode 100644 textfiles/c.txt

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 251 bytes | 83.00 KiB/s, done.
Total 2 (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:23-22411-071-eng/lab3.git
   6fb0187..8c1189c  main -> main
```

Check your GitHub repository again — the textfiles folder should now be deleted remotely.



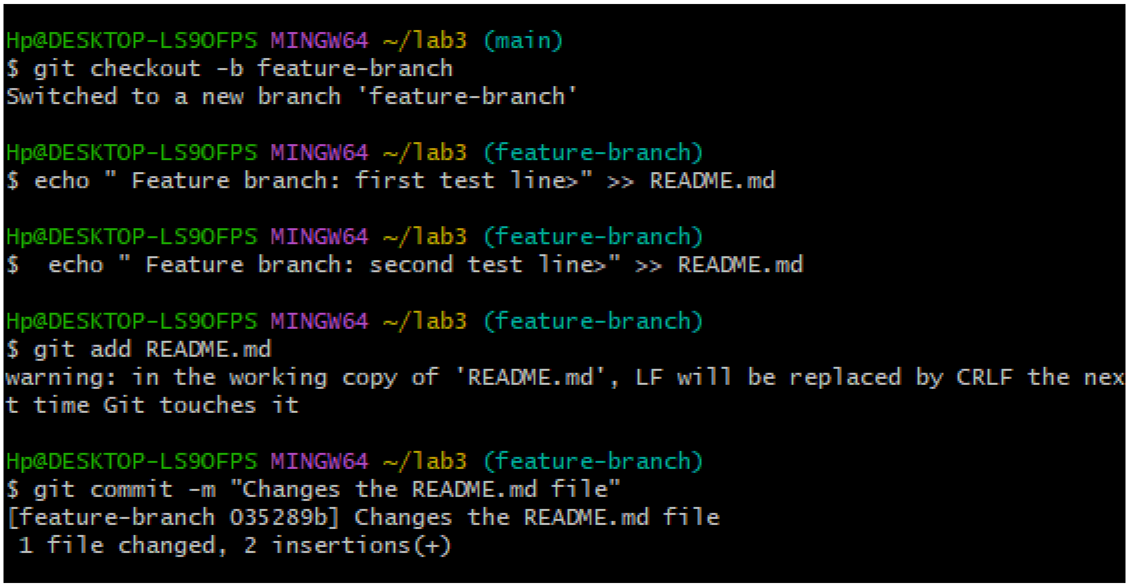
23-22411-071-eng Removed textfiles directory from tracking 8c1189c · 1 minute ago 14 Commits

.gitignore	Added .gitignore to ignore textfiles directory	8 minutes ago
README.md	Update README.md	35 minutes ago

Task 4 :

This task demonstrates how git stash allows you to temporarily save uncommitted changes so that you can safely switch branches without losing work.

Create a new branch: git checkout -b feature-branch.



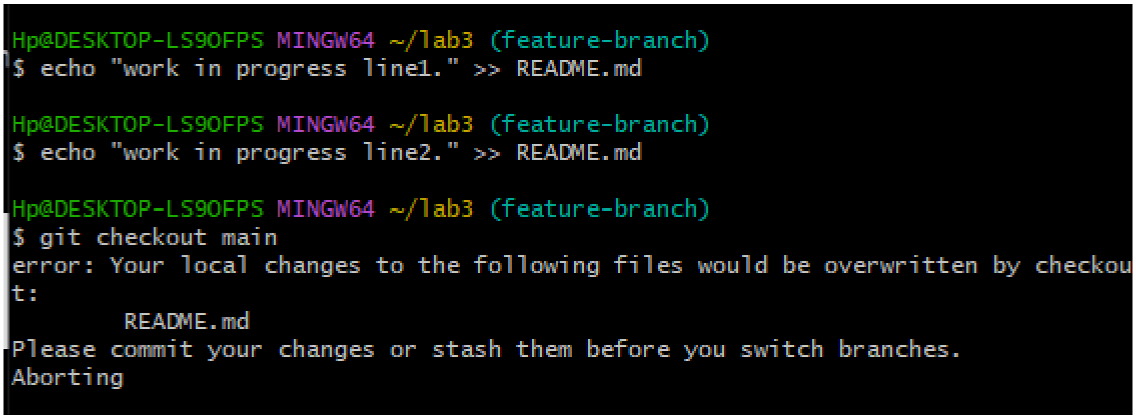
```
Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (feature-branch)
$ echo " Feature branch: first test line>" >> README.md

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (feature-branch)
$  echo " Feature branch: second test line>" >> README.md

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (feature-branch)
$ 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

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (feature-branch)
$ git commit -m "Changes the README.md file"
[feature-branch 035289b] Changes the README.md file
 1 file changed, 2 insertions(+)
```

Without committing or stashing the changes, try to switch to another branch: git checkout main

You'll see an error similar to:

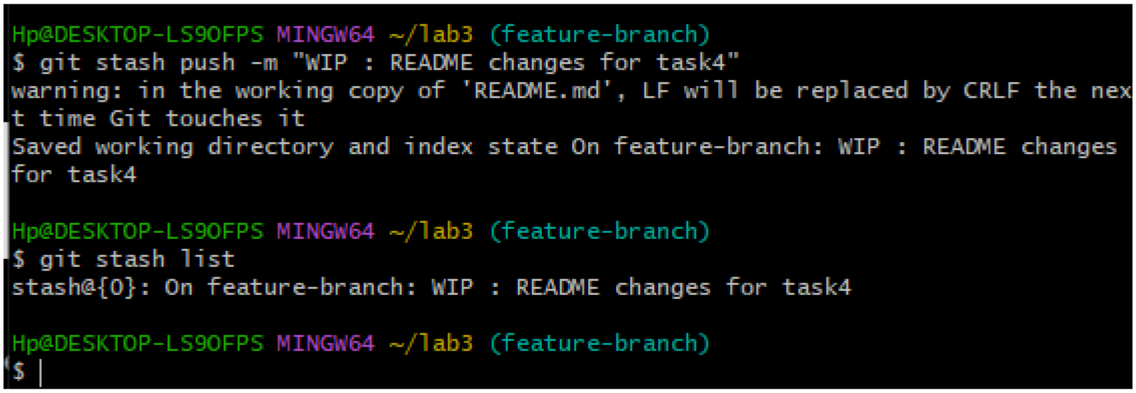
error: Your local changes to the following files would be overwritten by checkout:

README.md, Please commit your changes or stash them before you switch branches.



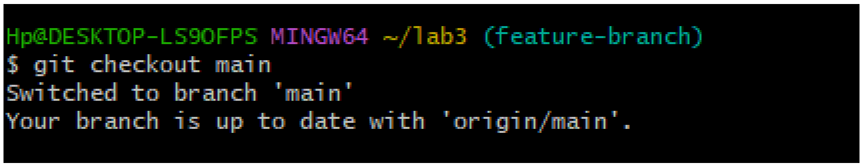
```
Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (feature-branch)
$ echo "work in progress line1." >> README.md

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (feature-branch)
$ echo "work in progress line2." >> README.md

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (feature-branch)
$ git checkout main
error: Your local changes to the following files would be overwritten by checkou
t:
        README.md
Please commit your changes or stash them before you switch branches.
Aborting
```

To fix this, stash your changes: git stash



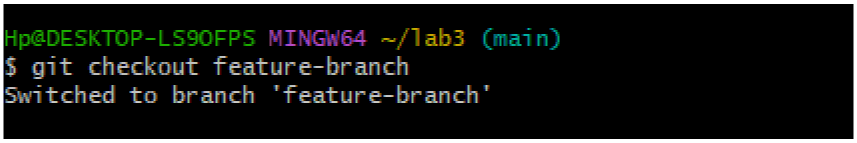
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git stash push -m "WIP : README changes for task4"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it
Saved working directory and index state On feature-branch: WIP : README changes
for task4

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git stash list
stash@{0}: On feature-branch: WIP : README changes for task4

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ |
```

Try switching branches again — it should now work.

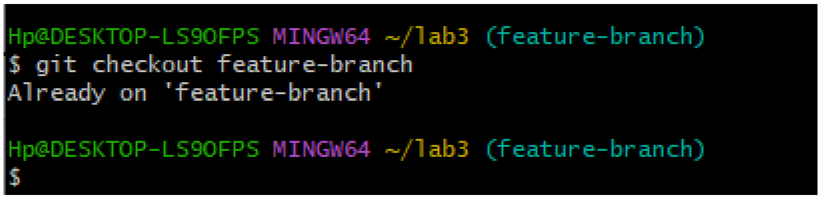


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Return to your previous branch: git checkout feature-branch

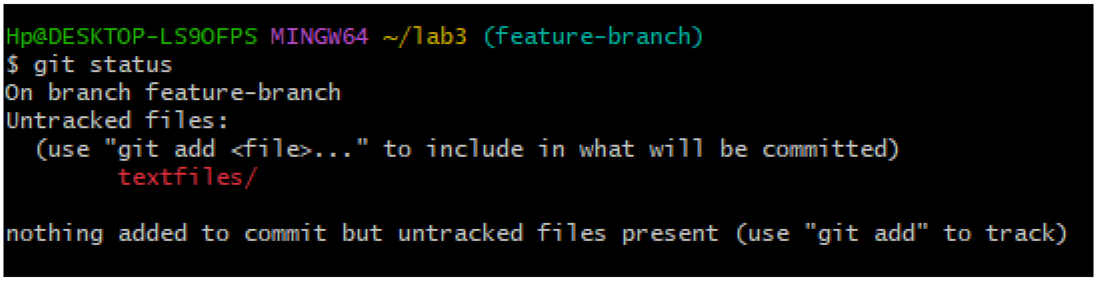


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git checkout feature-branch
Switched to branch 'feature-branch'
```

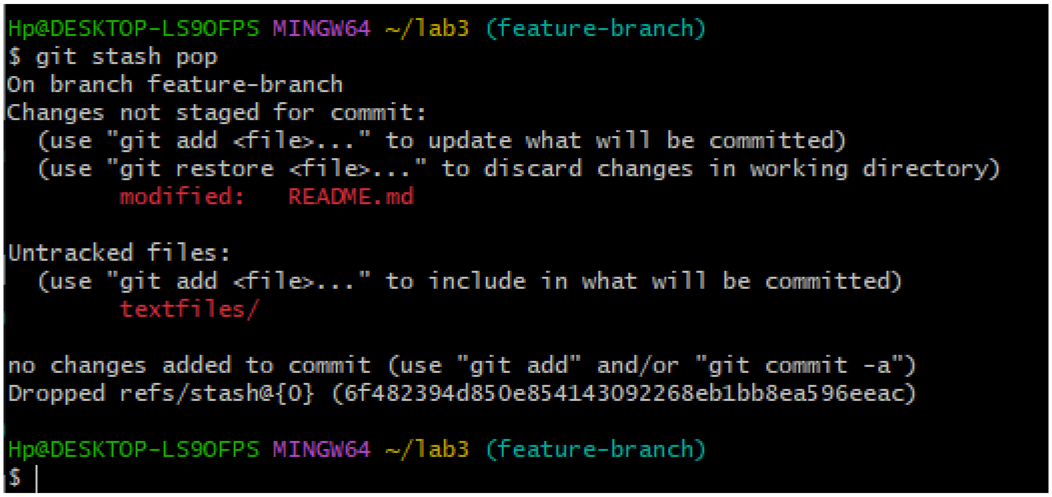


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git checkout feature-branch
Already on 'feature-branch'

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$
```

Now restore your stashed changes: git stash pop .You’ll see a message indicating that changes were reapplied.

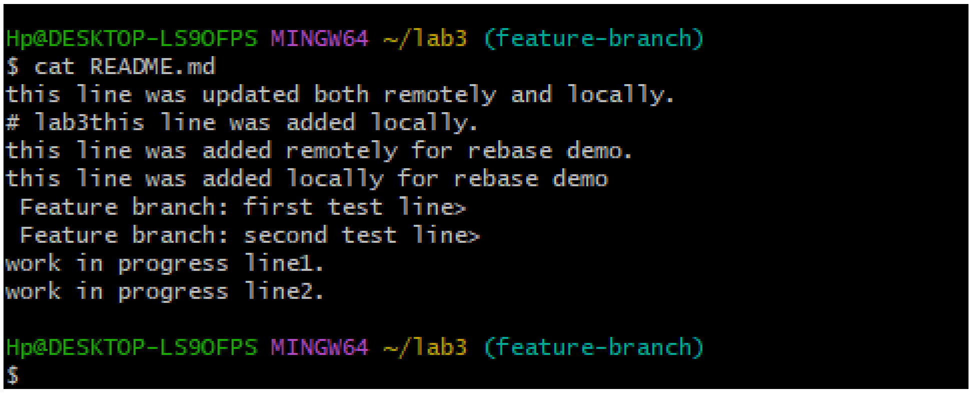


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git status
On branch feature-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        textfiles/

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



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git stash pop
On branch feature-branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

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

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (6f482394d850e854143092268eb1bb8ea596eeac)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ |
```

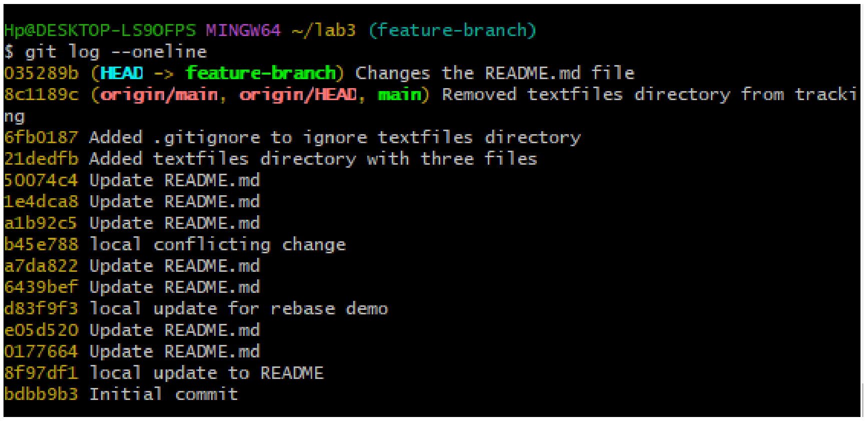
Confirm that your previous edits are back in the file.



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ cat README.md
this line was updated both remotely and locally.
# lab3this line was added locally.
this line was added remotely for rebase demo.
this line was added locally for rebase demo
 Feature branch: first test line>
 Feature branch: second test line>
work in progress line1.
work in progress line2.

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$
```

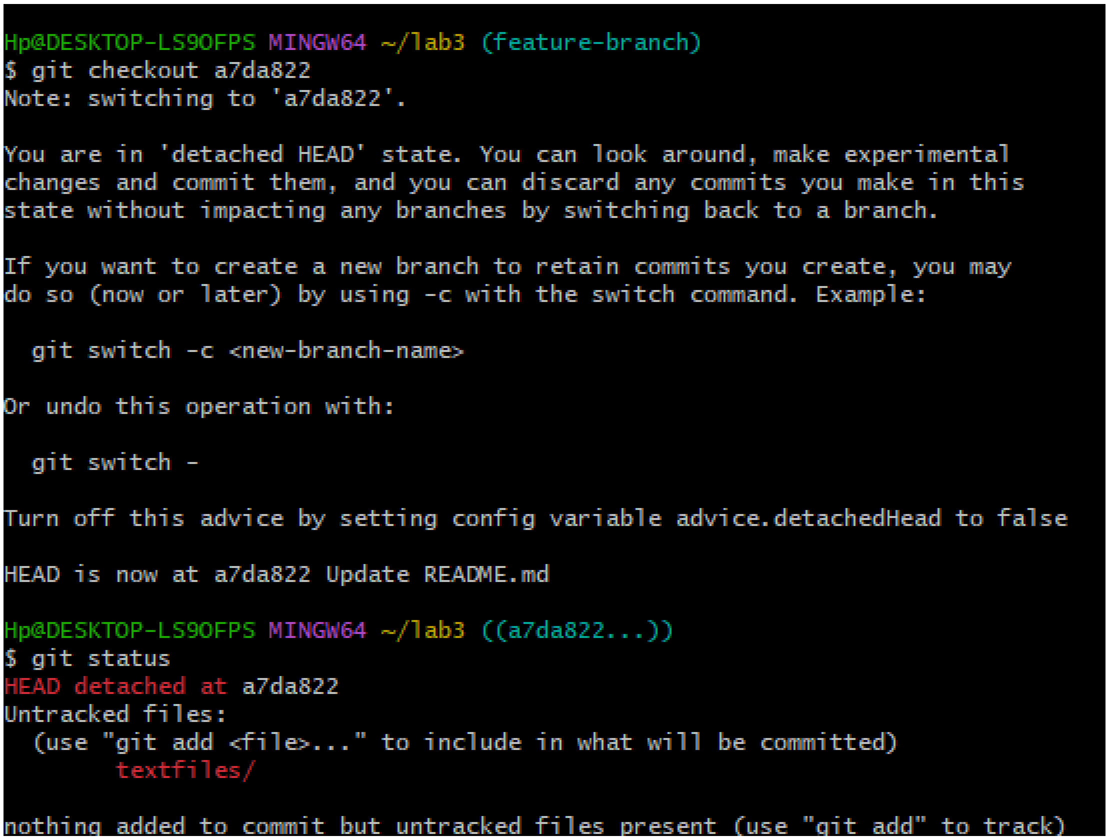
Task 5 :

View commit history: git log --oneline



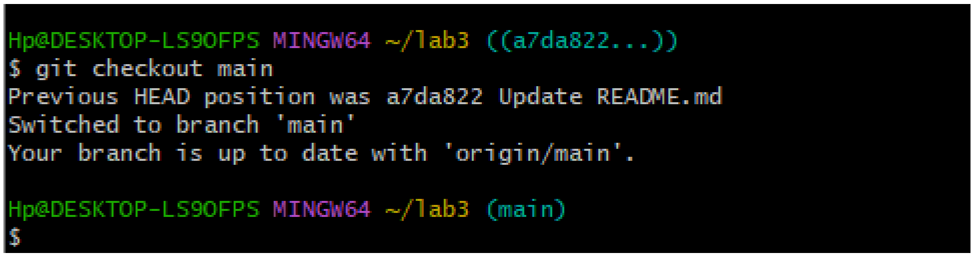
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git log --oneline
035289b (HEAD -> feature-branch) Changes the README.md file
8c1189c (origin/main, origin/HEAD, main) Removed textfiles directory from tracki
ng
6fb0187 Added .gitignore to ignore textfiles directory
21dedfb Added textfiles directory with three files
50074c4 Update README.md
1e4dca8 Update README.md
a1b92c5 Update README.md
b45e788 local conflicting change
a7da822 Update README.md
6439bef Update README.md
d83f9f3 local update for rebase demo
e05d520 Update README.md
0177664 Update README.md
8f97df1 local update to README
bdbb9b3 Initial commit
```

Copy any previous commit hash (e.g., a1b2c3d). Checkout that commit: git checkout <commit-hash>



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (feature-branch)
$ git checkout a7da822
Note: switching to 'a7da822'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at a7da822 Update README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 ((a7da822...))
$ git status
HEAD detached at a7da822
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        textfiles/

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

To return to your main branch: git checkout main

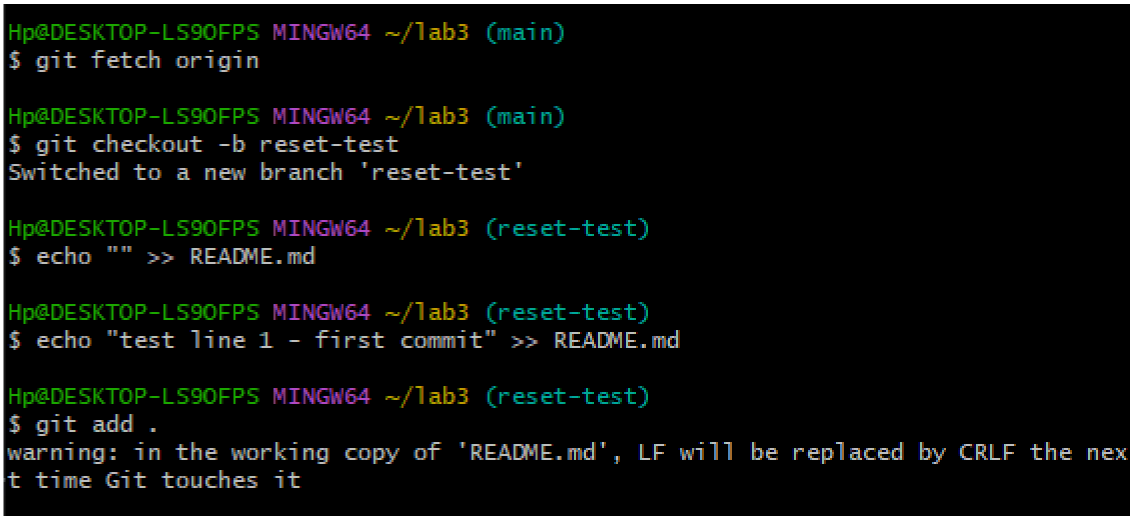


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 ((a7da822...))
$ git checkout main
Previous HEAD position was a7da822 Update README.md
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$
```

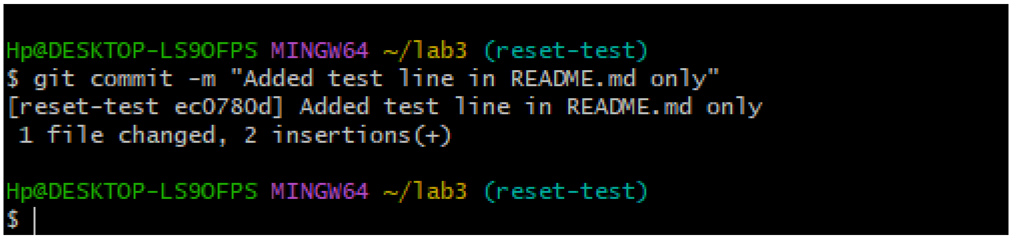
Task 6 :

This task demonstrates the difference between a soft and hard reset in Git. You will verify the presence of changes in files and confirm the existence of commits after each reset.

Add a new line in any file and commit it: git add . git commit -m "Added test line"

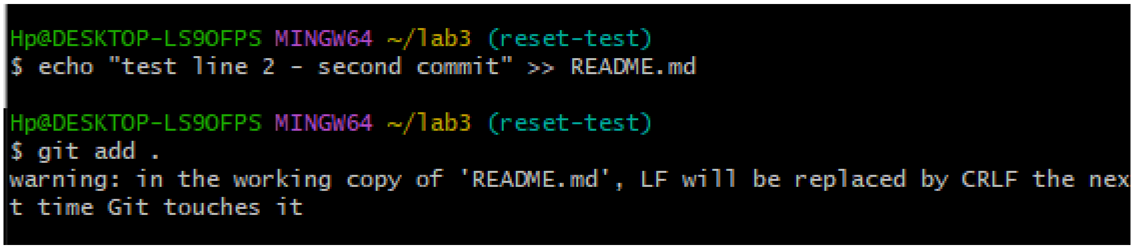


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git fetch origin

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (main)
$ git checkout -b reset-test
Switched to a new branch 'reset-test'

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ echo "" >> README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ echo "test line 1 - first commit" >> README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it
```

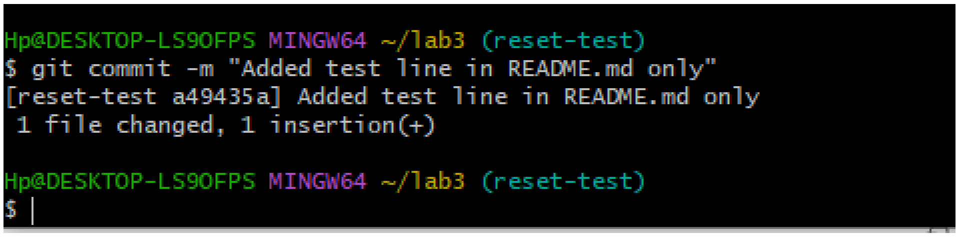


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git commit -m "Added test line in README.md only"
[reset-test ec0780d] Added test line in README.md only
 1 file changed, 2 insertions(+)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ |
```

Add another change and commit again: git add . git commit -m "Second test commit"

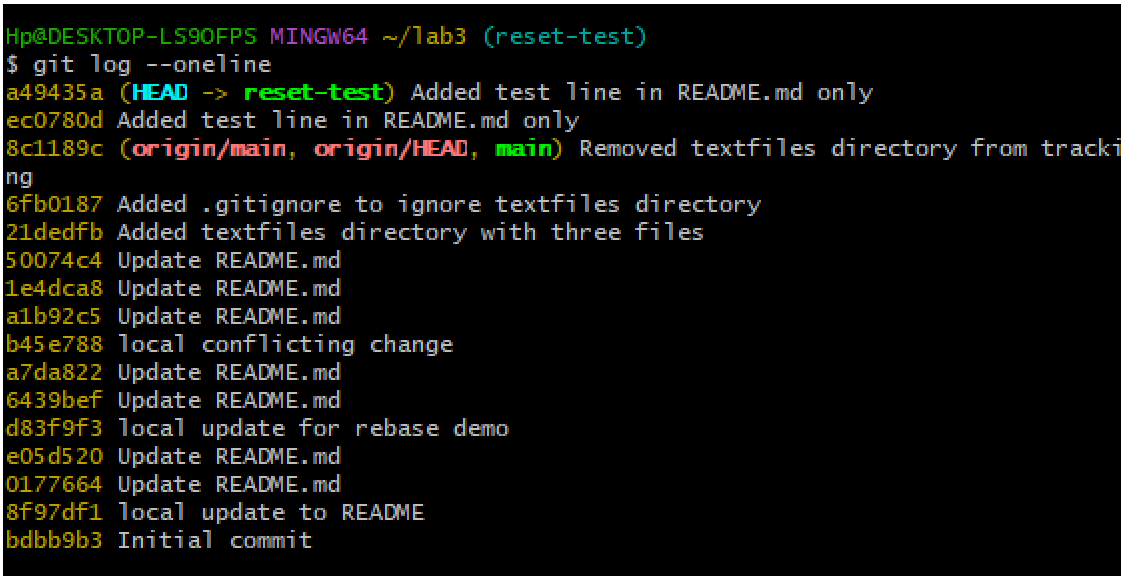


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ echo "test line 2 - second commit" >> README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it
```



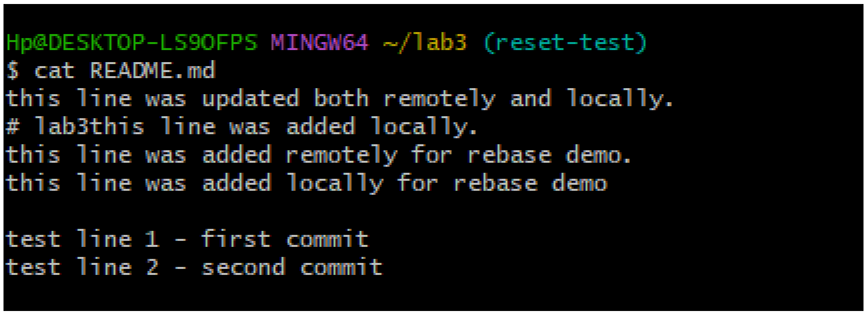
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git commit -m "Added test line in README.md only"
[reset-test a49435a] Added test line in README.md only
 1 file changed, 1 insertion(+)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ |
```

View the log history before reset: git log --oneline



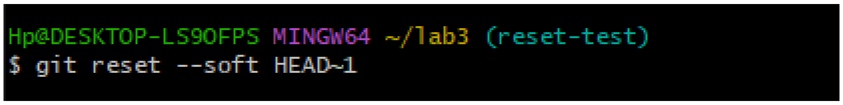
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git log --oneline
a49435a (HEAD -> reset-test) Added test line in README.md only
ec0780d Added test line in README.md only
8c1189c (origin/main, origin/HEAD, main) Removed textfiles directory from tracki
ng
6fb0187 Added .gitignore to ignore textfiles directory
21dedfb Added textfiles directory with three files
50074c4 Update README.md
1e4dca8 Update README.md
a1b92c5 Update README.md
b45e788 local conflicting change
a7da822 Update README.md
6439bef Update README.md
d83f9f3 local update for rebase demo
e05d520 Update README.md
0177664 Update README.md
8f97df1 local update to README
bdbb9b3 Initial commit
```

Open the edited file and visually confirm both added lines exist.



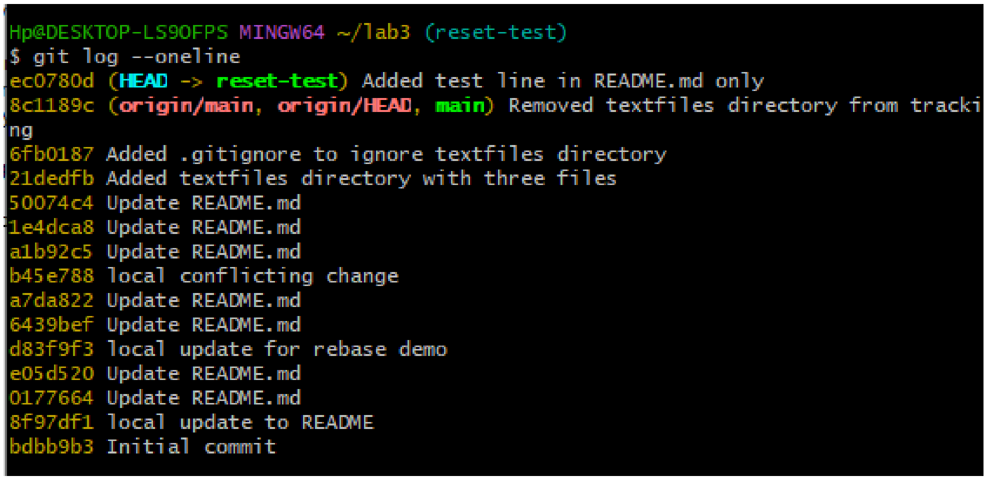
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ cat README.md
this line was updated both remotely and locally.
# lab3this line was added locally.
this line was added remotely for rebase demo.
this line was added locally for rebase demo

test line 1 - first commit
test line 2 - second commit
```

Perform a soft reset: git reset --soft HEAD~1



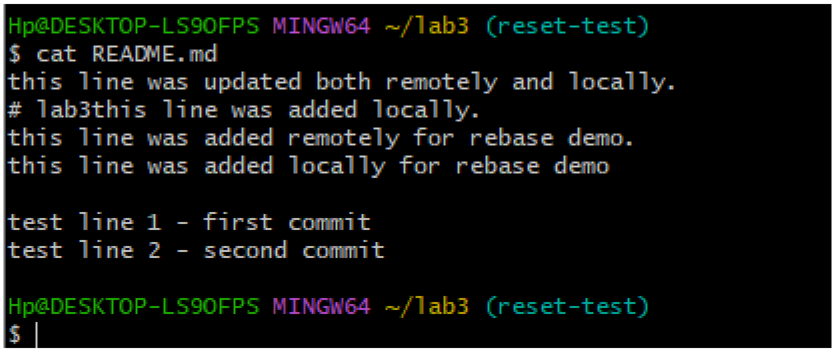
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git reset --soft HEAD~1
```

Check commit history after soft reset.



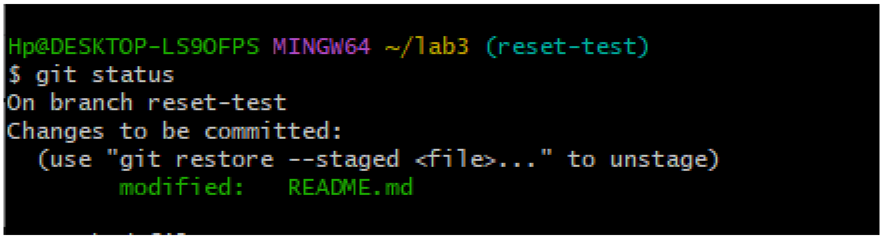
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git log --oneline
ec0780d (HEAD -> reset-test) Added test line in README.md only
8c1189c (origin/main, origin/HEAD, main) Removed textfiles directory from tracki
ng
6fb0187 Added .gitignore to ignore textfiles directory
21dedfb Added textfiles directory with three files
50074c4 Update README.md
1e4dca8 Update README.md
a1b92c5 Update README.md
b45e788 local conflicting change
a7da822 Update README.md
6439bef Update README.md
d83f9f3 local update for rebase demo
e05d520 Update README.md
0177664 Update README.md
8f97df1 local update to README
bdbb9b3 Initial commit
```

Verify changes in the file

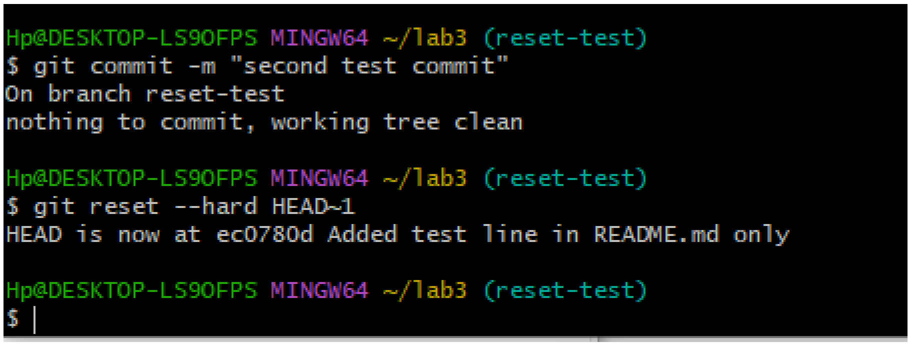


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ cat README.md
this line was updated both remotely and locally.
# lab3this line was added locally.
this line was added remotely for rebase demo.
this line was added locally for rebase demo

test line 1 - first commit
test line 2 - second commit

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ |
```

Check git status: git status .You should see your changes are staged (ready to commit again). Perform commit git commit -m "Second Test commit". Perform a hard reset (discards changes completely): git reset --hard HEAD~1

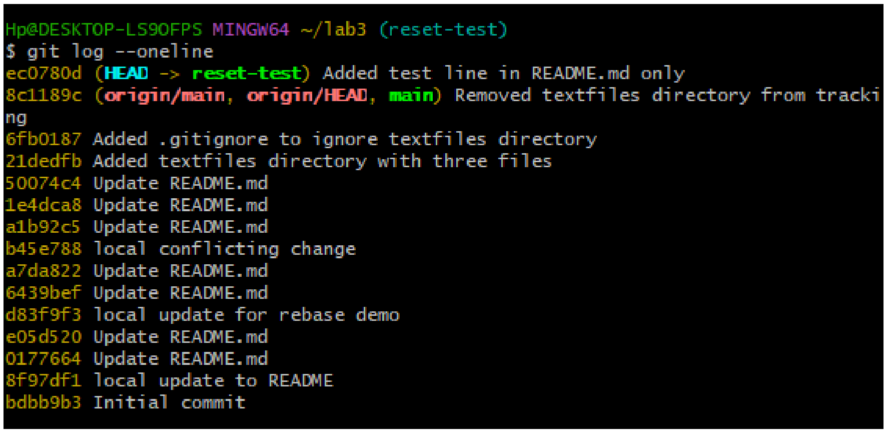


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git status
On branch reset-test
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
```



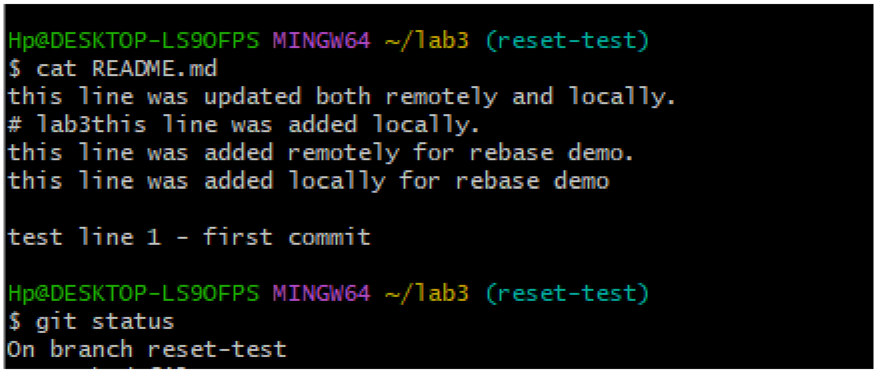
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git commit -m "second test commit"
On branch reset-test
nothing to commit, working tree clean

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git reset --hard HEAD~1
HEAD is now at ec0780d Added test line in README.md only

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ |
```

Check commit history after hard reset: git log --oneline



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git log --oneline
ec0780d (HEAD -> reset-test) Added test line in README.md only
8c1189c (origin/main, origin/HEAD, main) Removed textfiles directory from tracki
ng
6fb0187 Added .gitignore to ignore textfiles directory
21dedfb Added textfiles directory with three files
50074c4 Update README.md
1e4dca8 Update README.md
a1b92c5 Update README.md
b45e788 local conflicting change
a7da822 Update README.md
6439bef Update README.md
d83f9f3 local update for rebase demo
e05d520 Update README.md
0177664 Update README.md
8f97df1 local update to README
bdbb9b3 Initial commit
```

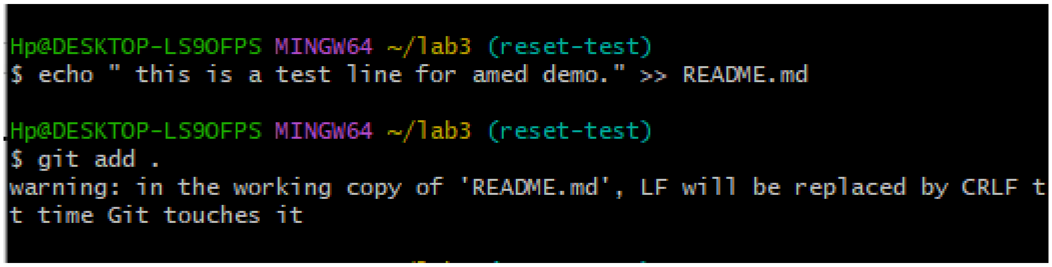
Verify changes in the file: Open the file and confirm the second edit is gone (reverted to previous commit). Check git status: git status .Should report "nothing to commit, working tree clean".



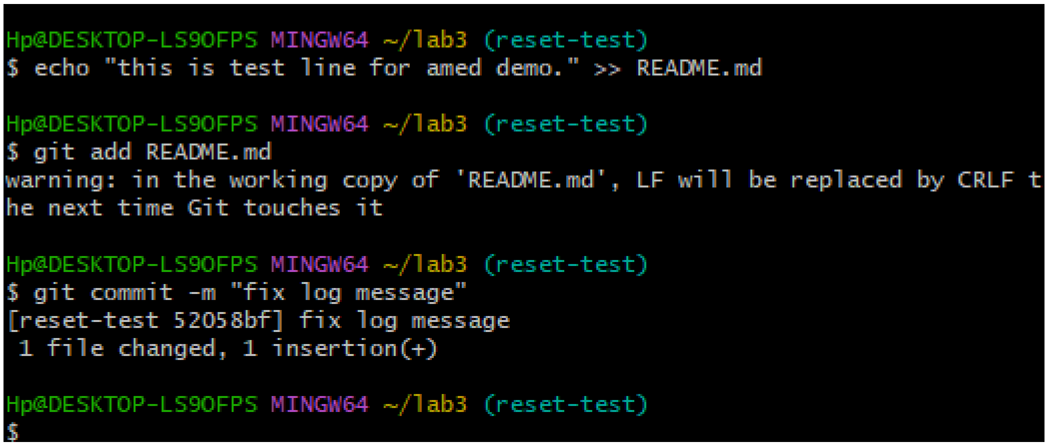
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ cat README.md
this line was updated both remotely and locally.
# lab3this line was added locally.
this line was added remotely for rebase demo.
this line was added locally for rebase demo

test line 1 - first commit

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git status
On branch reset-test
```

Task 7 :

Make a small change in any file. Stage it and commit: git add . git commit -m "Fix log message"

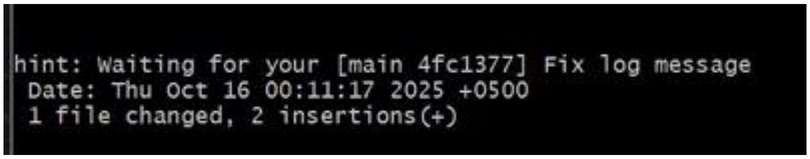


```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ echo " this is a test line for amed demo." >> README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git add .
warning: in the working copy of 'README.md', LF will be replaced by CRLF t
t time Git touches it
```



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ echo "this is test line for amed demo." >> README.md

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF t
he next time Git touches it

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git commit -m "fix log message"
[reset-test 52058bf] fix log message
 1 file changed, 1 insertion(+)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$
```

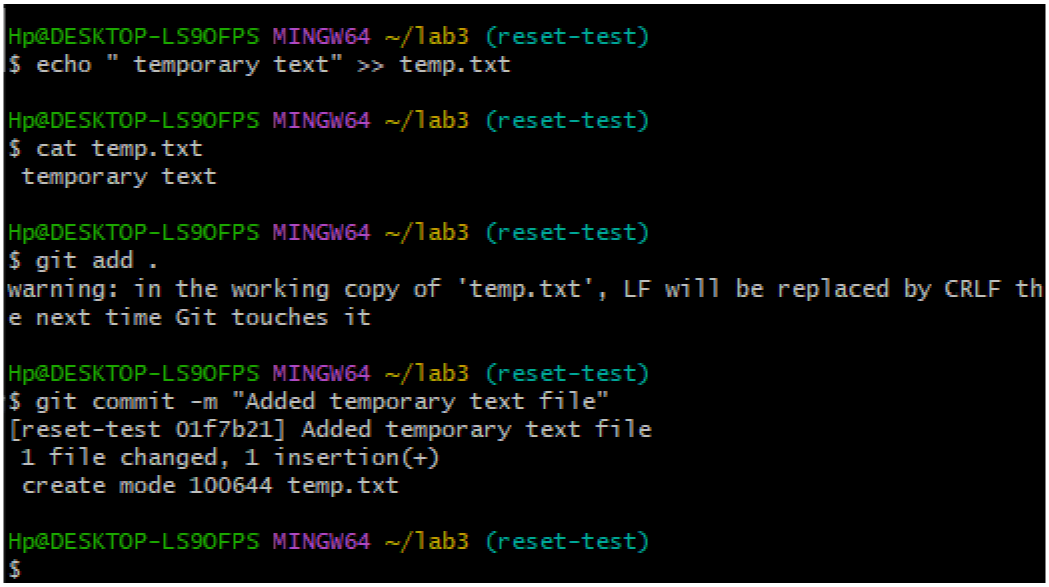
Realize you forgot to change another file. Update it now.Add it and amend the last commit: git add . git commit --amend



```
hint: Waiting for your [main 4fc1377] Fix log message
 Date: Thu Oct 16 00:11:17 2025 +0500
 1 file changed, 2 insertions(+)
```

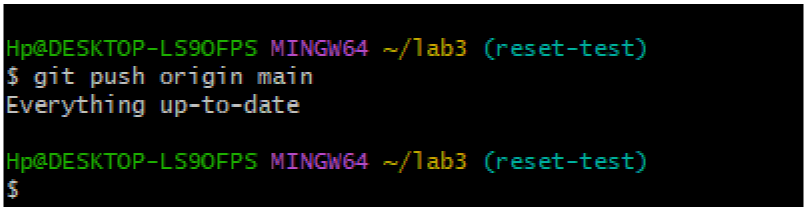
Task 8 :

Make a change and commit it: echo "temporary text" >> temp.txt
git add . git commit -m "Added temporary text file"



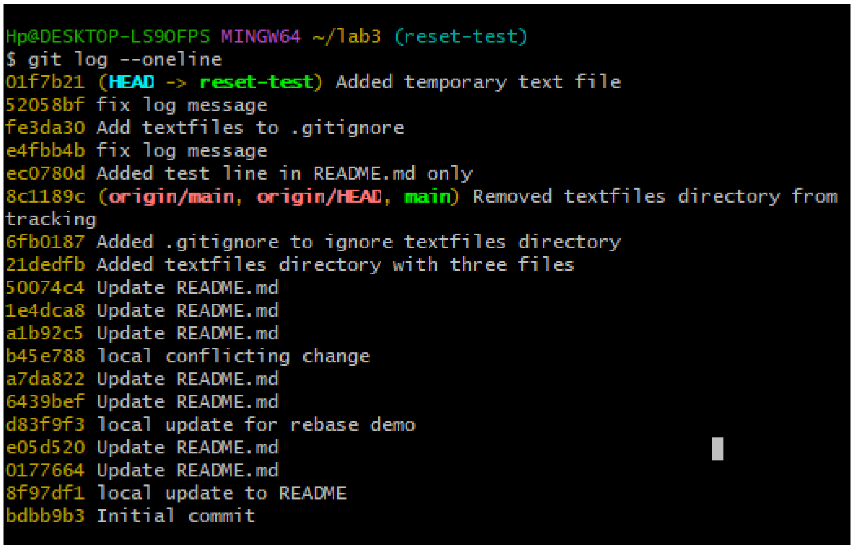
```
Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$ echo " temporary text" >> temp.txt

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$ cat temp.txt
 temporary text

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$ git add .
warning: in the working copy of 'temp.txt', LF will be replaced by CRLF th
e next time Git touches it

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$ git commit -m "Added temporary text file"
[reset-test 01f7b21] Added temporary text file
 1 file changed, 1 insertion(+)
 create mode 100644 temp.txt

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$
```

git push origin main



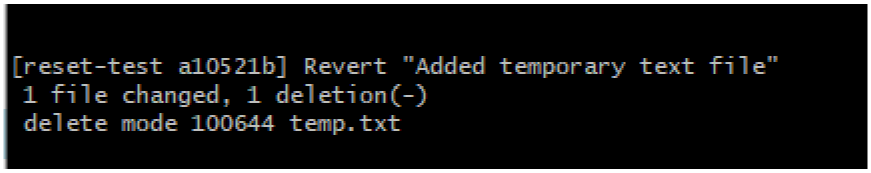
```
Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$ git push origin main
Everything up-to-date

Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$
```

Git log --oneline



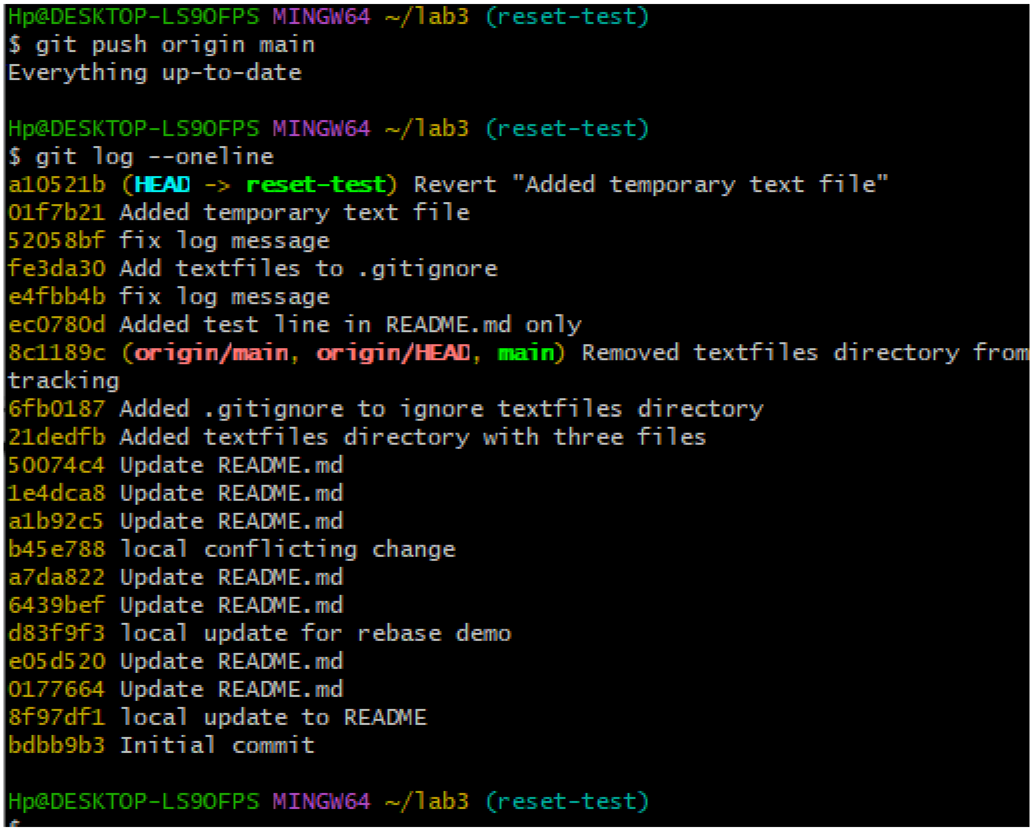
```
Hp@DESKTOP-LS90FPS MINGW64 ~/lab3 (reset-test)
$ git log --oneline
01f7b21 (HEAD -> reset-test) Added temporary text file
52058bf fix log message
fe3da30 Add textfiles to .gitignore
e4fbb4b fix log message
ec0780d Added test line in README.md only
8c1189c (origin/main, origin/HEAD, main) Removed textfiles directory from
tracking
6fb0187 Added .gitignore to ignore textfiles directory
21dedfb Added textfiles directory with three files
50074c4 Update README.md
1e4dca8 Update README.md
a1b92c5 Update README.md
b45e788 local conflicting change
a7da822 Update README.md
6439bef Update README.md
d83f9f3 local update for rebase demo
e05d520 Update README.md
0177664 Update README.md
8f97df1 local update to README
bdbb9b3 Initial commit
```

Now revert that commit safely (do not delete it): git log --oneline
git revert <a10521b>



```
[reset-test a10521b] Revert "Added temporary text file"
 1 file changed, 1 deletion(-)
 delete mode 100644 temp.txt
```

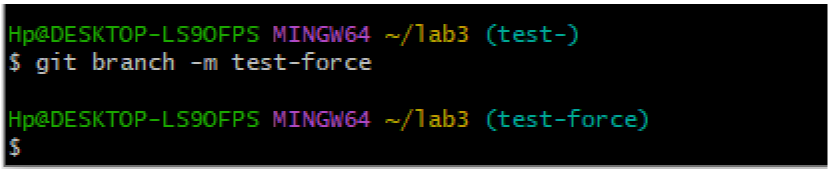
Push the revert commit: git push origin main



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git push origin main
Everything up-to-date

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
$ git log --oneline
a10521b (HEAD -> reset-test) Revert "Added temporary text file"
01f7b21 Added temporary text file
52058bf fix log message
fe3da30 Add textfiles to .gitignore
e4fbb4b fix log message
ec0780d Added test line in README.md only
8c1189c (origin/main, origin/HEAD, main) Removed textfiles directory from
tracking
6fb0187 Added .gitignore to ignore textfiles directory
21dedfb Added textfiles directory with three files
50074c4 Update README.md
1e4dca8 Update README.md
a1b92c5 Update README.md
b45e788 local conflicting change
a7da822 Update README.md
6439bef Update README.md
d83f9f3 local update for rebase demo
e05d520 Update README.md
0177664 Update README.md
8f97df1 local update to README
bdbb9b3 Initial commit

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (reset-test)
```

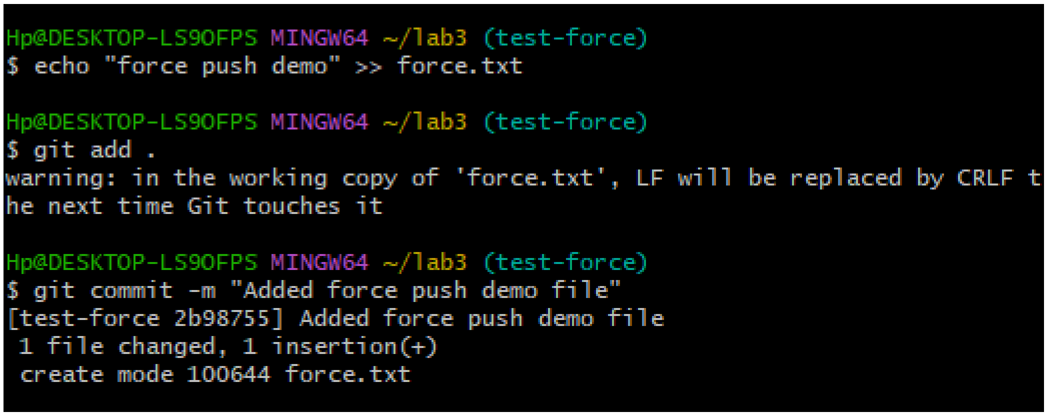
Task 9 :

Use this only in your own branch (never on main or develop).Create a new branch: git checkout -b test-force



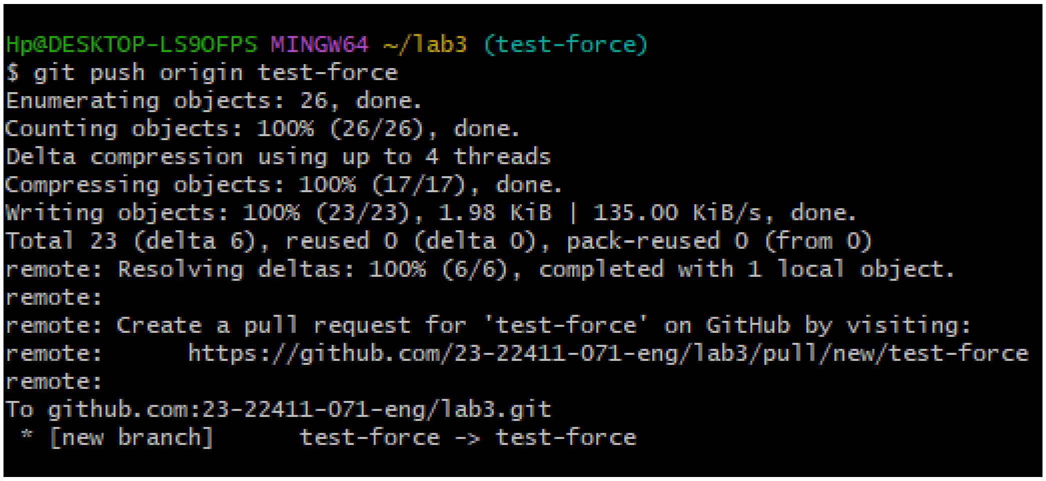
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-)
$ git branch -m test-force

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$
```

Make and commit a small change



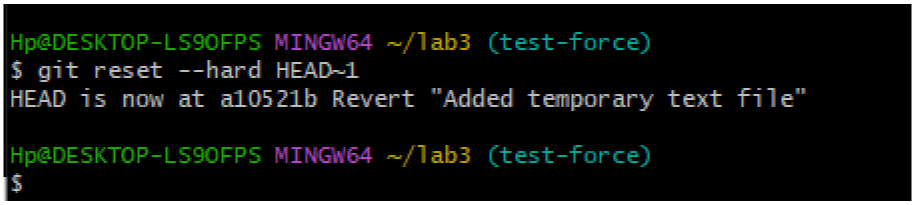
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ echo "force push demo" >> force.txt

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ git add .
warning: in the working copy of 'force.txt', LF will be replaced by CRLF t
he next time Git touches it

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ git commit -m "Added force push demo file"
[test-force 2b98755] Added force push demo file
 1 file changed, 1 insertion(+)
 create mode 100644 force.txt
```

Push it: git push origin test-force



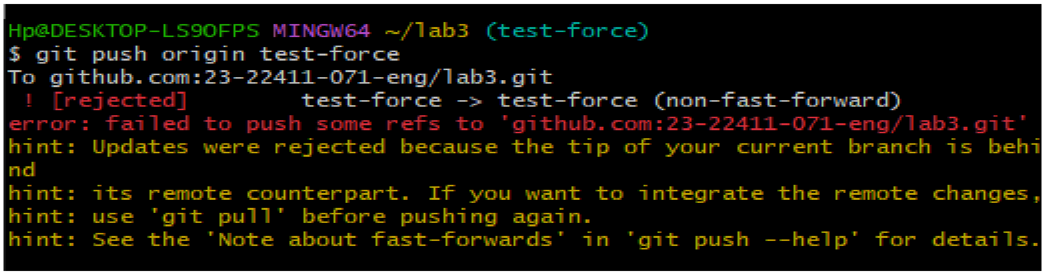
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ git push origin test-force
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 4 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (23/23), 1.98 KiB | 135.00 KiB/s, done.
Total 23 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 1 local object.
remote:
remote: Create a pull request for 'test-force' on GitHub by visiting:
remote:      https://github.com/23-22411-071-eng/lab3/pull/new/test-force
remote:
To github.com:23-22411-071-eng/lab3.git
 * [new branch]      test-force -> test-force
```

Perform a hard reset: git reset --hard HEAD~1



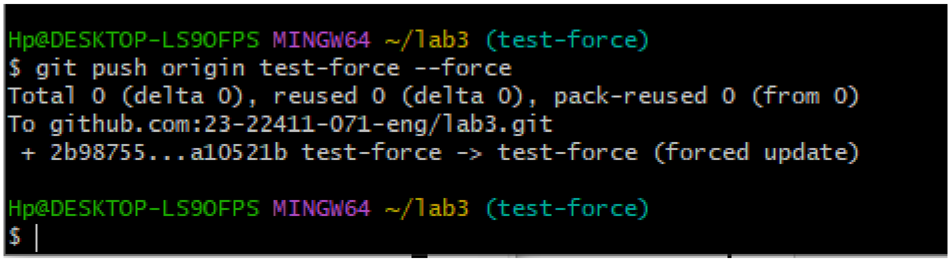
```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ git reset --hard HEAD~1
HEAD is now at a10521b Revert "Added temporary text file"

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$
```

Push again : git push origin test-force .Rejected by remote repository



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ git push origin test-force
To github.com:23-22411-071-eng/lab3.git
 ! [rejected]        test-force -> test-force (non-fast-forward)
error: failed to push some refs to 'github.com:23-22411-071-eng/lab3.git'
hint: Updates were rejected because the tip of your current branch is behi
nd
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

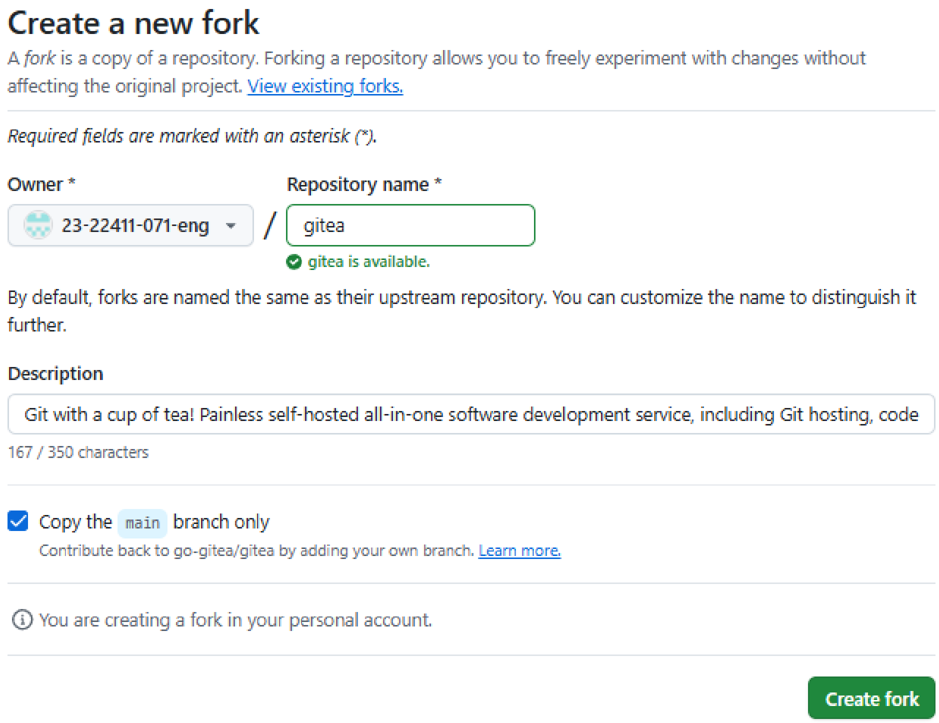
Now force-push to remote: git push origin test-force --force



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ git push origin test-force --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:23-22411-071-eng/lab3.git
 + 2b98755...a10521b test-force -> test-force (forced update)

Hp@DESKTOP-LS9OFPS MINGW64 ~/lab3 (test-force)
$ |
```

Task 10 :

This task will guide you through forking a repository, running a full web application in a GitHub Codespace, and interacting with Gitea (a self-hosted Git service) inside Codespaces.
Fork the Gitea Repository. Go to Gitea. Click the Fork button at the top right and fork it into your own GitHub account.



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.

Required fields are marked with an asterisk (*).

Owner * 23-22411-071-eng / Repository name * gitea

gitea is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description

Git with a cup of tea! Painless self-hosted all-in-one software development service, including Git hosting, code

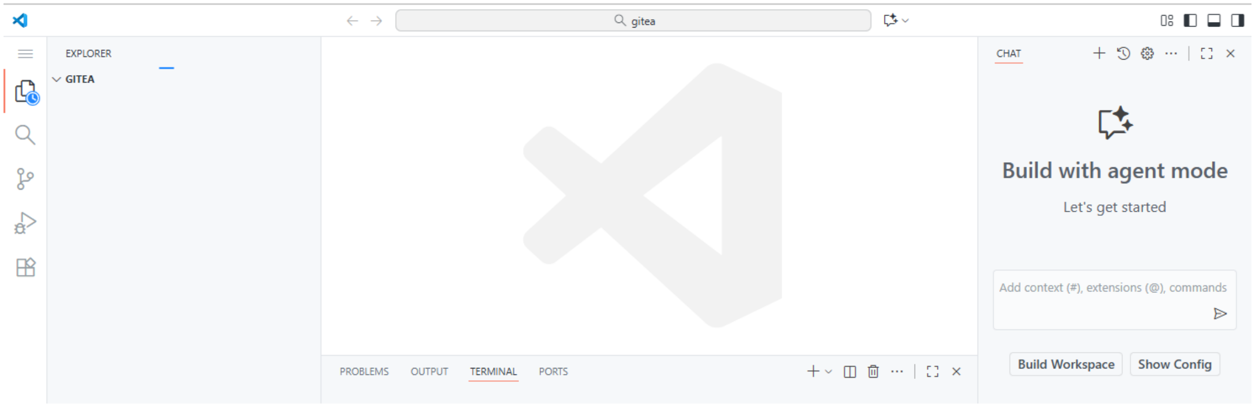
167 / 350 characters

☑ Copy the `main` branch only

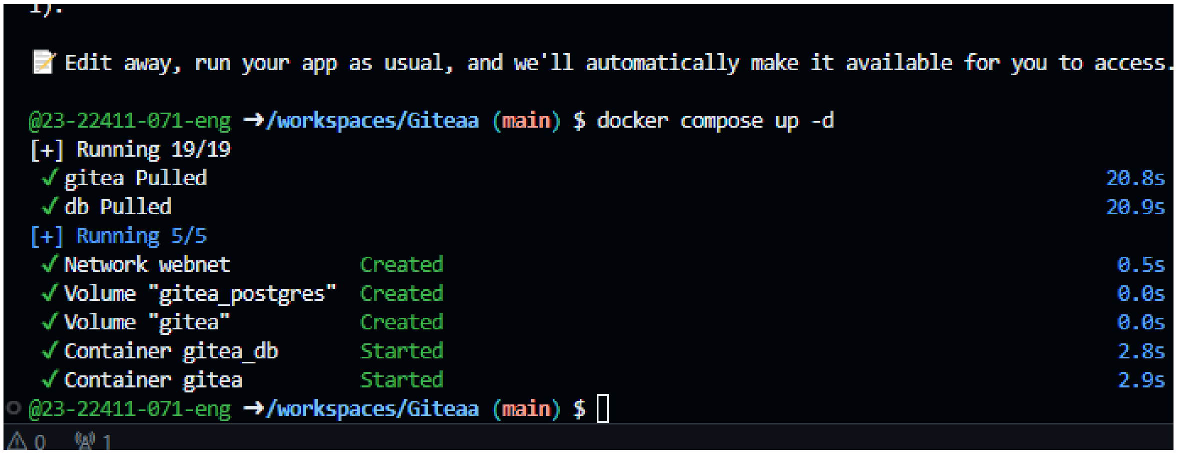
Contribute back to go-gitea/gitea by adding your own branch. Learn more.

You are creating a fork in your personal account.

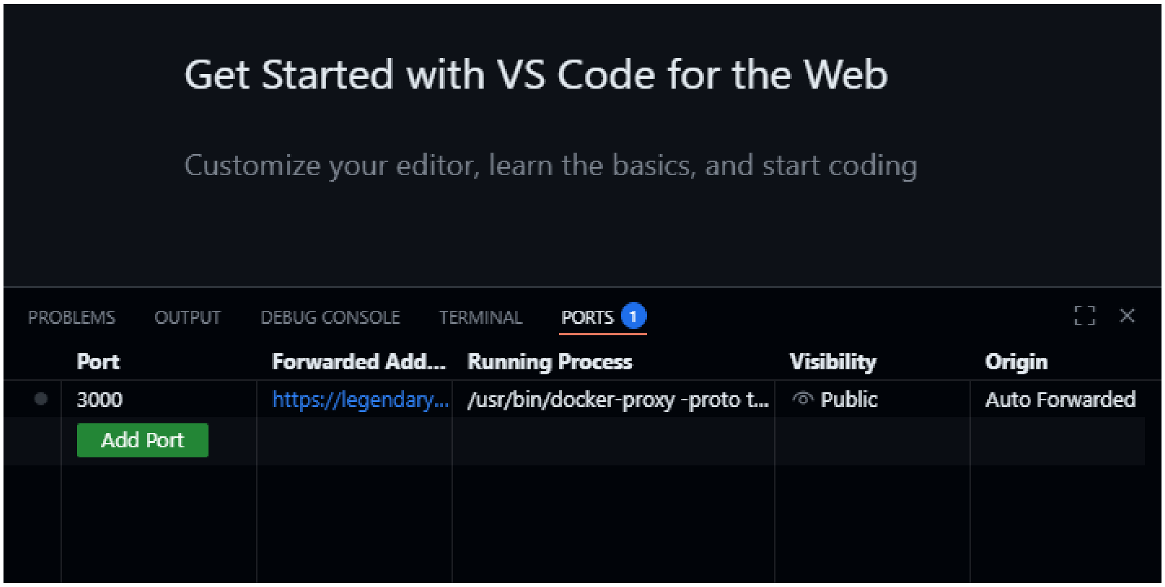
Open the Forked Repo in GitHub Codespaces. On your forked repo, click Code > Codespaces > Create codespace on main.



Start Gitea with Docker Compose .In the Codespace terminal, run: docker compose up –d. Wait for the containers to start.Access Gitea Web Interface. In Codespaces, forward port 3000 (see Codespaces port forwarding UI).Change the Visibility of port 3000 from private to public. Click the forwarded port 3000 link to open Gitea in your browser tab.



```
1):

📝 Edit away, run your app as usual, and we'll automatically make it available for you to access.

@23-22411-071-eng ➜ /workspaces/Giteaa (main) $ docker compose up -d
[+] Running 19/19
 ✓ gitea Pulled                                                              20.8s
 ✓ db Pulled                                                                 20.9s
[+] Running 5/5
 ✓ Network webnet            Created                                          0.5s
 ✓ Volume "gitea_postgres"   Created                                          0.0s
 ✓ Volume "gitea"            Created                                          0.0s
 ✓ Container gitea_db        Started                                          2.8s
 ✓ Container gitea           Started                                          2.9s
@23-22411-071-eng ➜ /workspaces/Giteaa (main) $
```

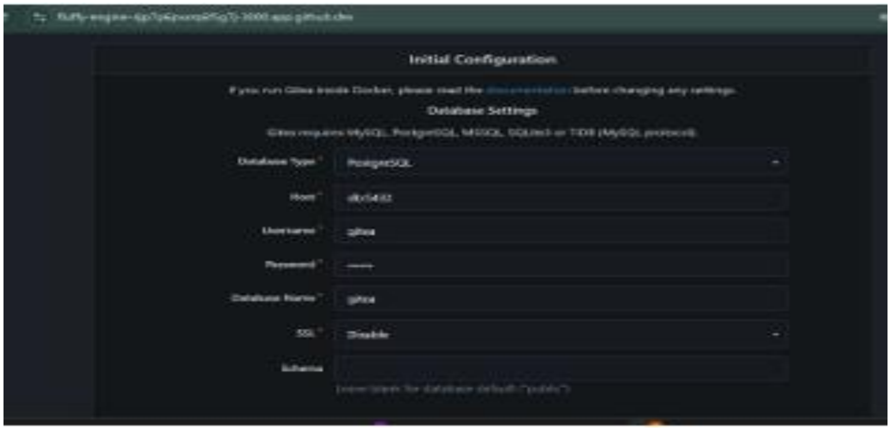


Get Started with VS Code for the Web

Customize your editor, learn the basics, and start coding

Port	Forwarded Add...	Running Process	Visibility	Origin
3000	https://legendary...	/usr/bin/docker-proxy -proto t...	Public	Auto Forwarded
Add Port				

Install Gitea ,Fill out the installation form, providing an admin username and password (e.g., admin / yourpassword)



Initial Configuration

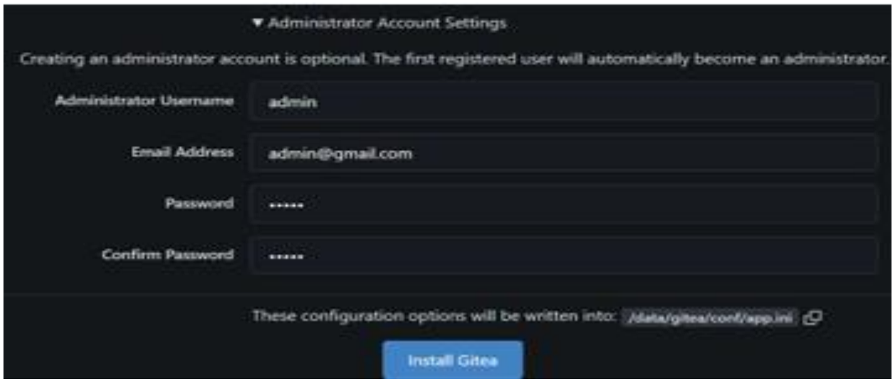
If you run Gitea inside Docker, please read the documentation before changing any settings.

Database Settings

Gitea requires MySQL, PostgreSQL, MSSQL, SQLite3 or TiDB (MySQL protocol).

Database Type *	PostgreSQL
Host *	db:5432
Username *	gitea
Password *	------
Database Name *	gitea
SSL *	Disable
Schema	

Leave blank for database default ("public").



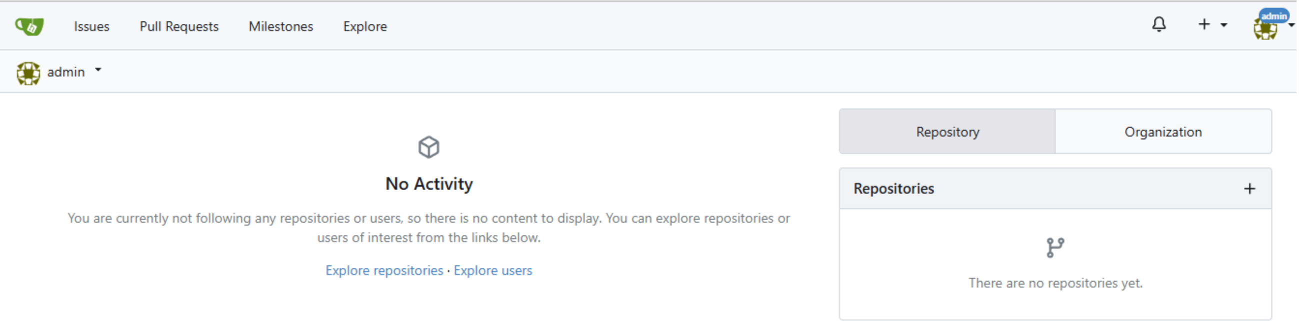
▾ Administrator Account Settings

Creating an administrator account is optional. The first registered user will automatically become an administrator.

Administrator Username	admin
Email Address	admin@gmail.com
Password	••••••
Confirm Password	••••••

These configuration options will be written into: /data/gitea/conf/app.ini

Log In to Gitea. Use your admin credentials to log in.

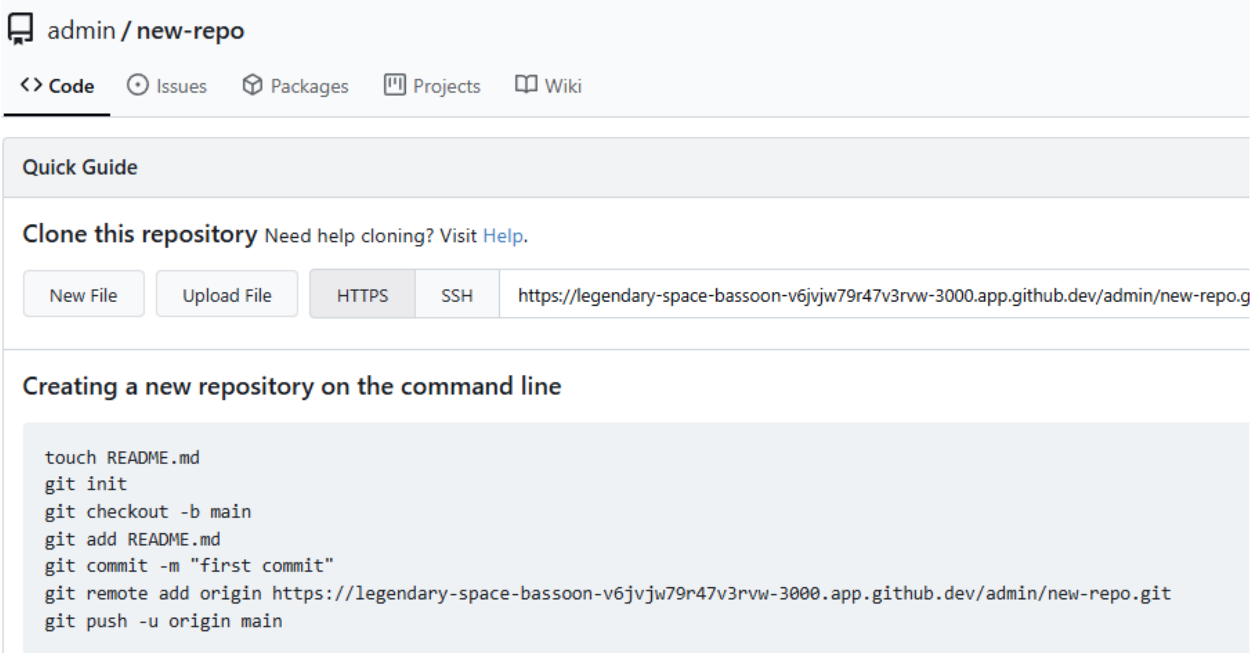


No Activity

You are currently not following any repositories or users, so there is no content to display. You can explore repositories or users of interest from the links below.

There are no repositories yet.

Create a New Repository in Gitea. Click New Repository or + > New Repository in Gitea.



Quick Guide

Clone this repository Need help cloning? Visit Help.

https://legendary-space-bassoon-v6jvjw79r47v3rvw-3000.app.github.dev/admin/new-repo.g

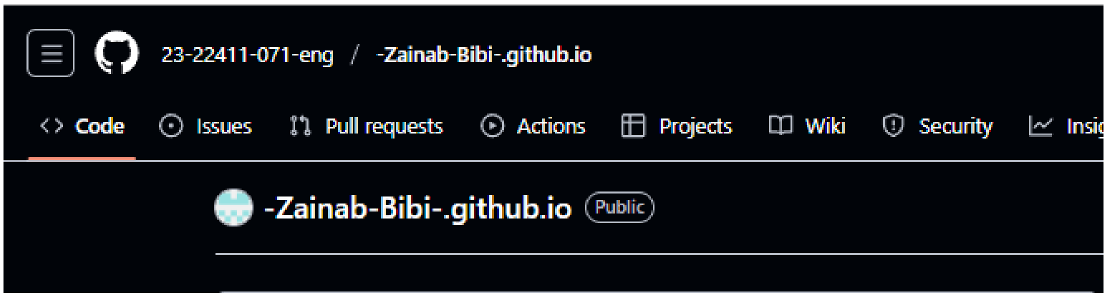
Creating a new repository on the command line

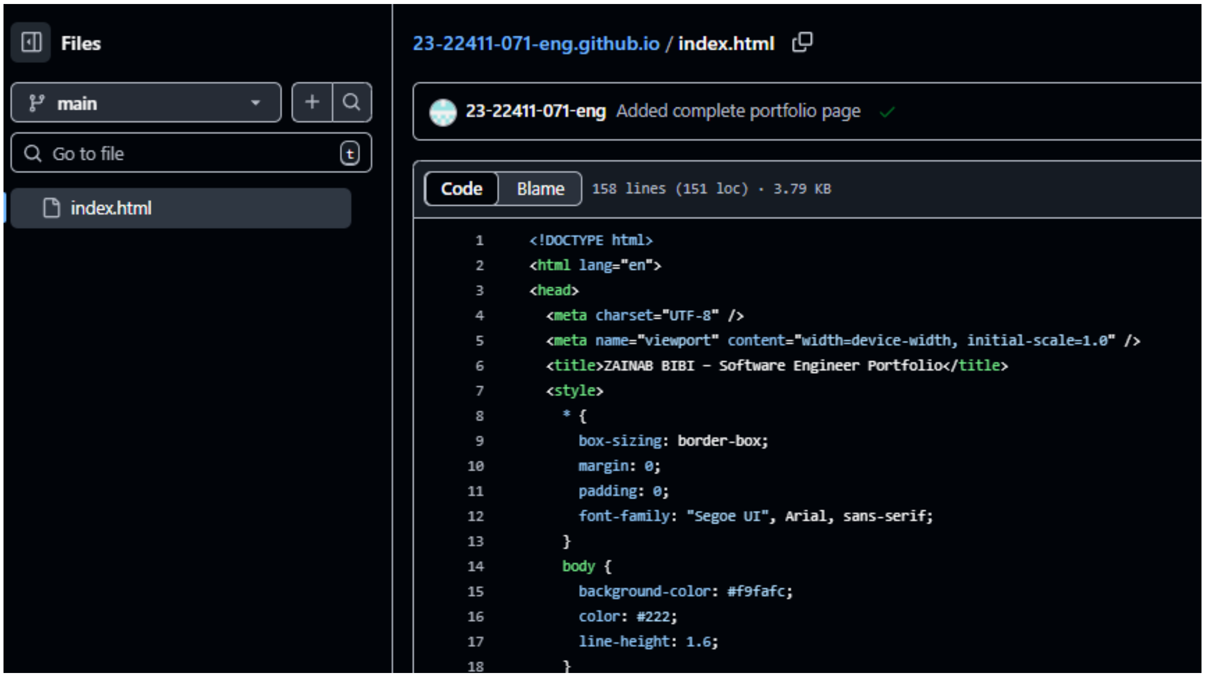
```
touch README.md
git init
git checkout -b main
git add README.md
git commit -m "first commit"
git remote add origin https://legendary-space-bassoon-v6jvjw79r47v3rvw-3000.app.github.dev/admin/new-repo.git
git push -u origin main
```

Task 11: Creating a GitHub Pages Portfolio Site

This task will guide you through creating a personal portfolio/static site using GitHub Pages.

Create a GitHub Pages Repository named <username>.github.io. Make sure the repository is public.



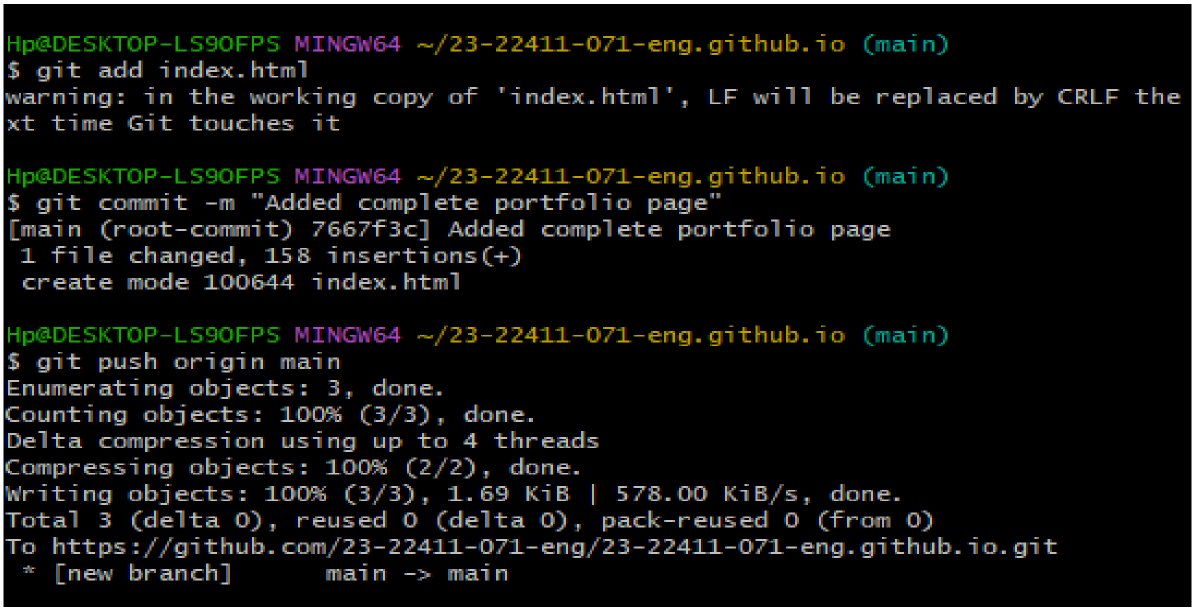


23-22411-071-eng.github.io / index.html

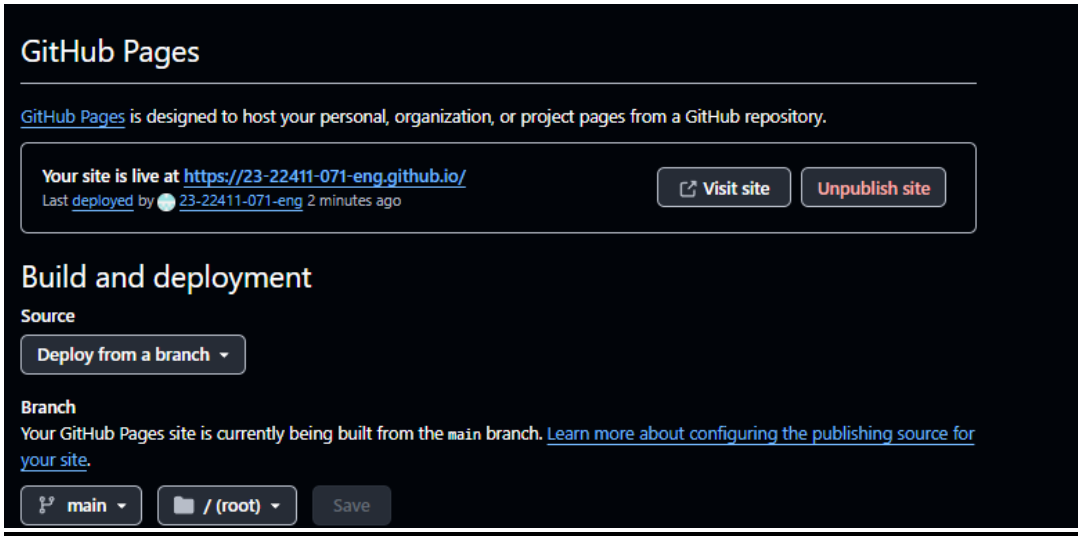
23-22411-071-eng Added complete portfolio page ✓

158 lines (151 loc) · 3.79 KB

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>ZAINAB BIBI - Software Engineer Portfolio</title>
  <style>
    * {
      box-sizing: border-box;
      margin: 0;
      padding: 0;
      font-family: "Segoe UI", Arial, sans-serif;
    }
    body {
      background-color: #f9fafc;
      color: #222;
      line-height: 1.6;
    }
```



```
Hp@DESKTOP-LS9OFPS MINGW64 ~/23-22411-071-eng.github.io (main)
$ git add index.html
warning: in the working copy of 'index.html', LF will be replaced by CRLF the
xt time Git touches it

Hp@DESKTOP-LS9OFPS MINGW64 ~/23-22411-071-eng.github.io (main)
$ git commit -m "Added complete portfolio page"
[main (root-commit) 7667f3c] Added complete portfolio page
 1 file changed, 158 insertions(+)
 create mode 100644 index.html

Hp@DESKTOP-LS9OFPS MINGW64 ~/23-22411-071-eng.github.io (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.69 KiB | 578.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/23-22411-071-eng/23-22411-071-eng.github.io.git
 * [new branch]      main -> main
```



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

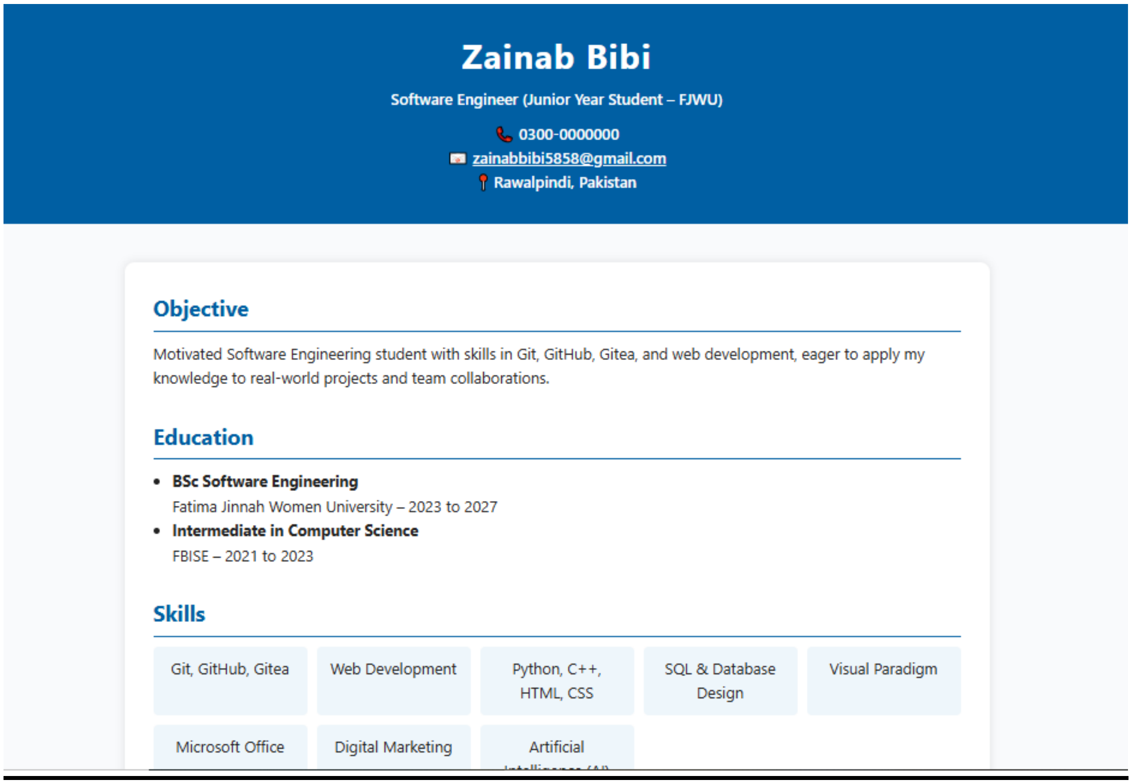
Your site is live at https://23-22411-071-eng.github.io/
Last deployed by 23-22411-071-eng 2 minutes ago

Build and deployment

Source: Deploy from a branch

Branch
Your GitHub Pages site is currently being built from the `main` branch. Learn more about configuring the publishing source for your site.

main / (root)



Zainab Bibi

Software Engineer (Junior Year Student – FJWU)

☎ 0300-0000000
✉ zainabbibi5858@gmail.com
📍 Rawalpindi, Pakistan

Objective

Motivated Software Engineering student with skills in Git, GitHub, Gitea, and web development, eager to apply my knowledge to real-world projects and team collaborations.

Education

- **BSc Software Engineering**
  Fatima Jinnah Women University – 2023 to 2027
- **Intermediate in Computer Science**
  FBISE – 2021 to 2023

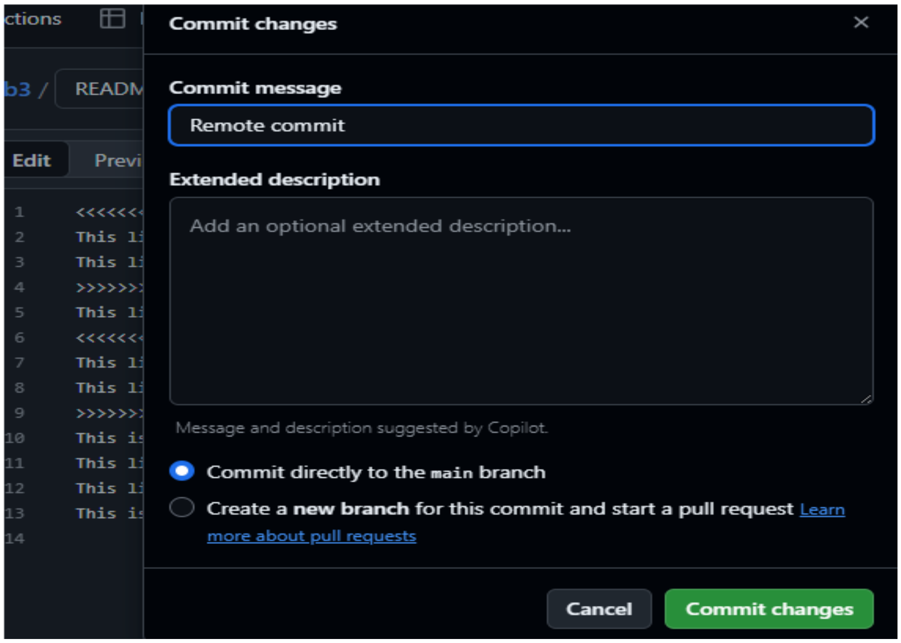
Skills

Git, GitHub, Gitea
Web Development
Python, C++, HTML, CSS
SQL & Database Design
Visual Paradigm
Microsoft Office
Digital Marketing
Artificial

https://23-22411-071-eng.github.io/

Exam Evaluation Questions

1.Local vs Remote Conflict Resolution



Commit changes

Commit message

Remote commit

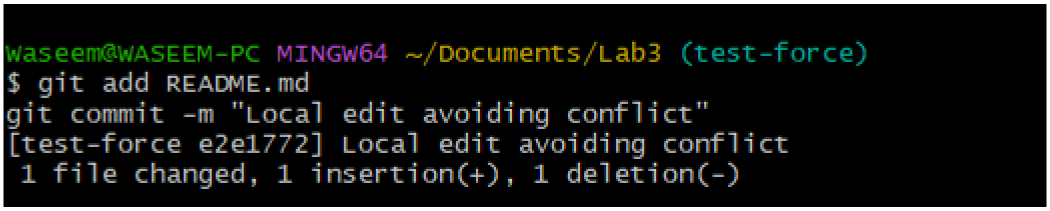
Extended description

Add an optional extended description...

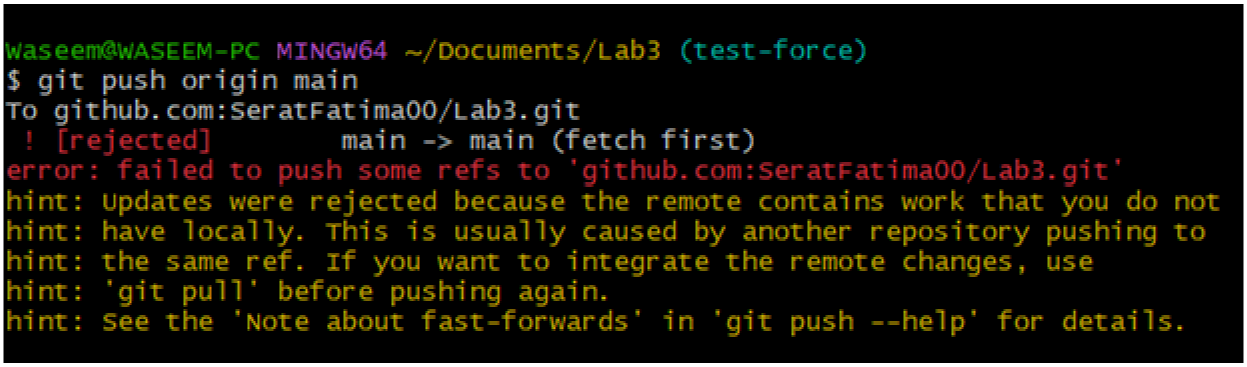
Message and description suggested by Copilot.

● Commit directly to the `main` branch

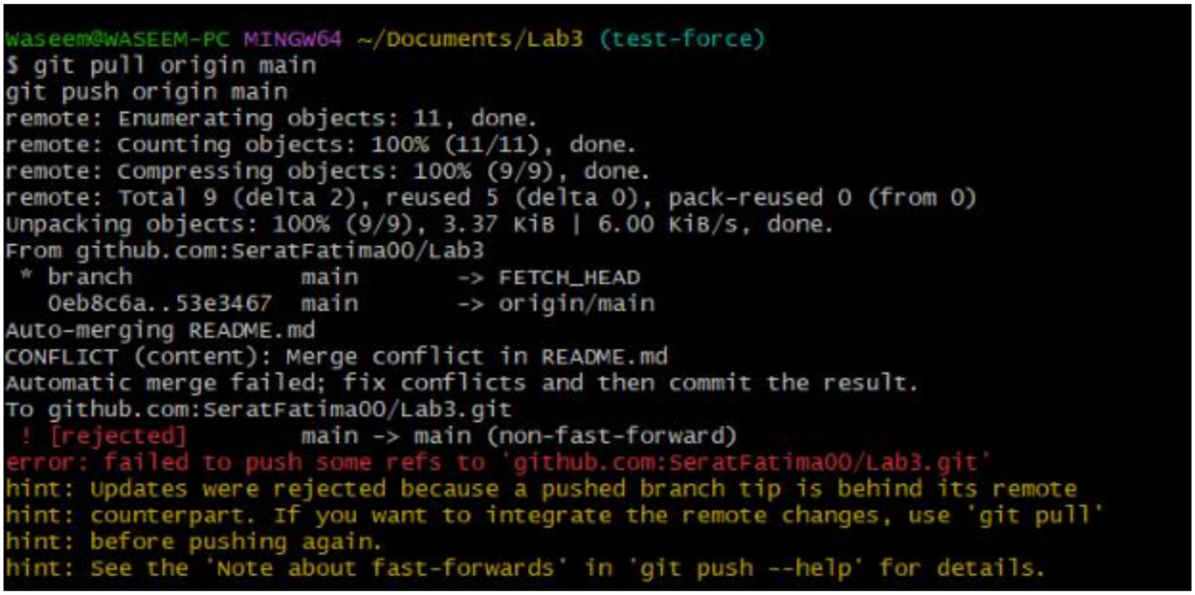
○ Create a **new branch** for this commit and start a pull request Learn more about pull requests



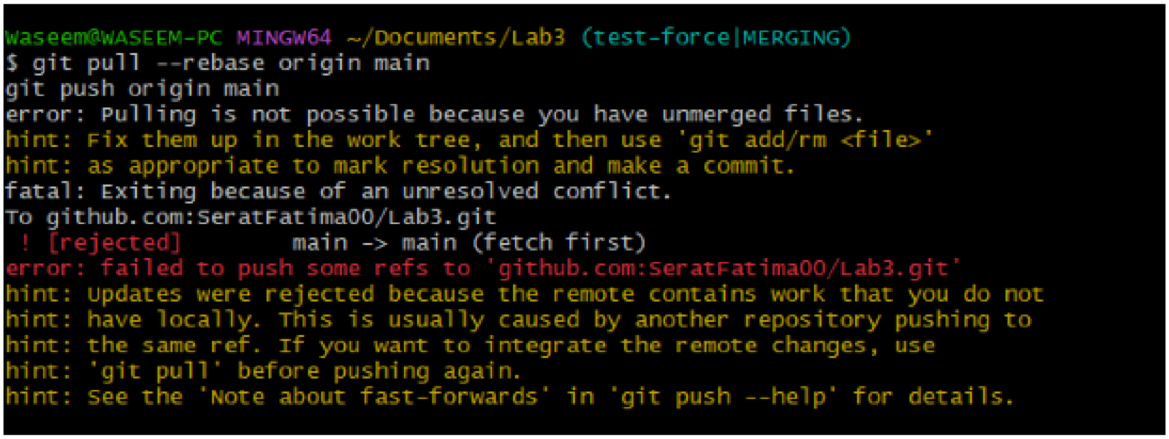
```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git add README.md
git commit -m "Local edit avoiding conflict"
[test-force e2e1772] Local edit avoiding conflict
 1 file changed, 1 insertion(+), 1 deletion(-)
```



```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git push origin main
To github.com:SeratFatima00/Lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:SeratFatima00/Lab3.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

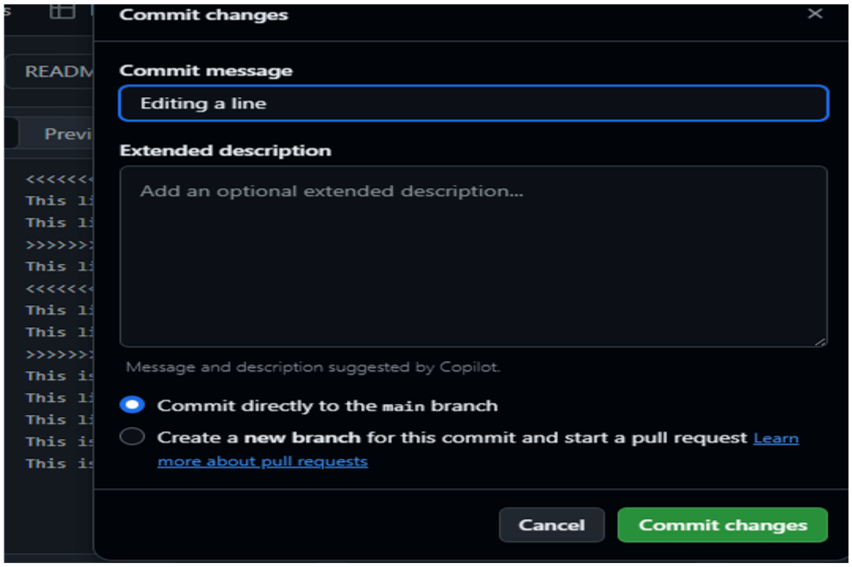


```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git pull origin main
git push origin main
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 9 (delta 2), reused 5 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (9/9), 3.37 KiB | 6.00 KiB/s, done.
From github.com:SeratFatima00/Lab3
 * branch            main       -> FETCH_HEAD
   0eb8c6a..53e3467  main       -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
To github.com:SeratFatima00/Lab3.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'github.com:SeratFatima00/Lab3.git'
hint: Updates were rejected because a pushed branch tip is behind its remote
hint: counterpart. If you want to integrate the remote changes, use 'git pull'
hint: before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```



```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force|MERGING)
$ git pull --rebase origin main
git push origin main
error: Pulling is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
To github.com:SeratFatima00/Lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:SeratFatima00/Lab3.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

2. Manual Merge Conflict Handling



Commit changes ×

Commit message

Editing a line

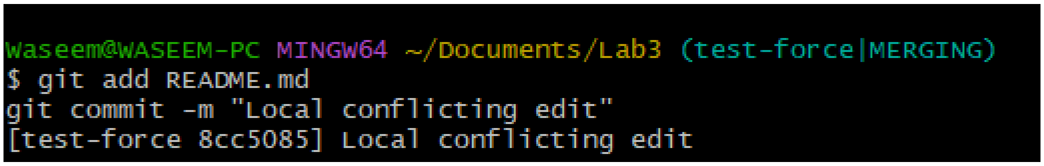
Extended description

Add an optional extended description...

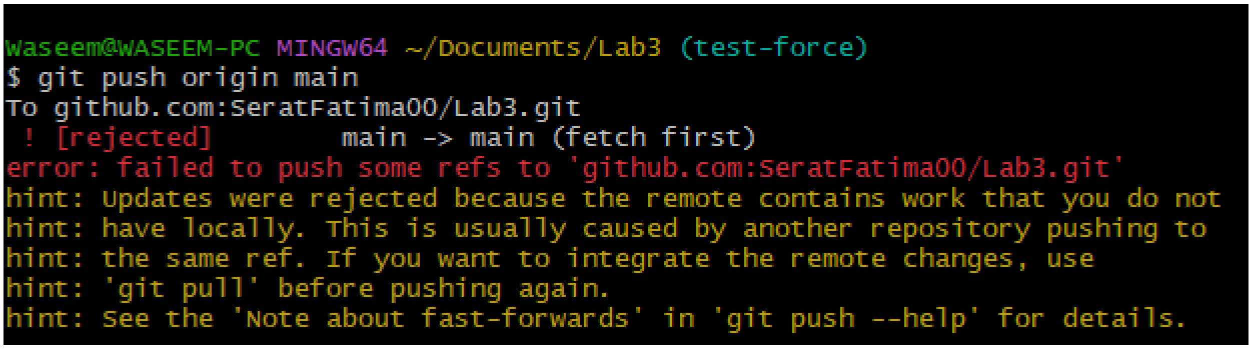
Message and description suggested by Copilot.

● Commit directly to the `main` branch

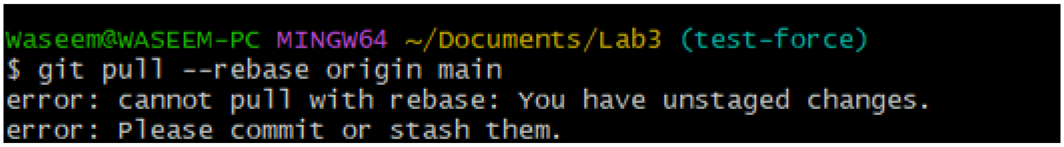
○ Create a **new branch** for this commit and start a pull request Learn more about pull requests



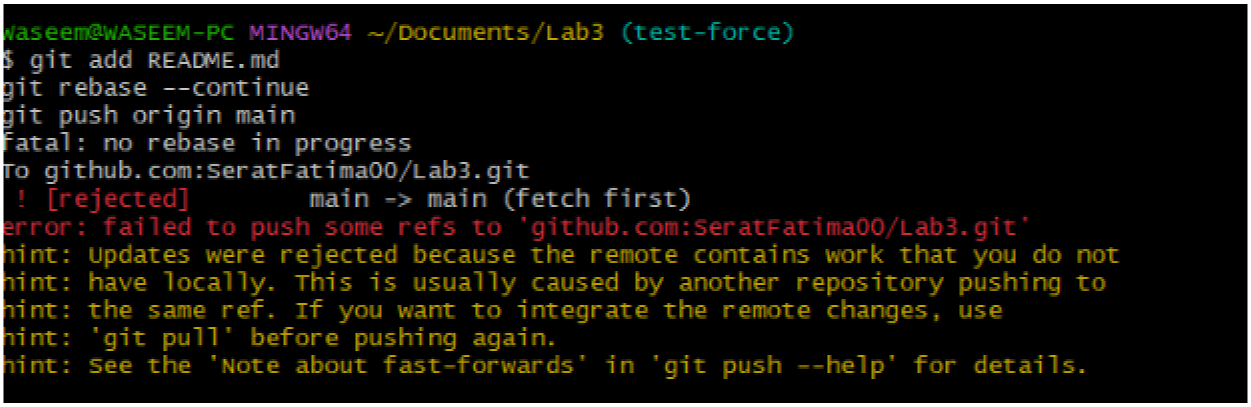
```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force|MERGING)
$ git add README.md
git commit -m "Local conflicting edit"
[test-force 8cc5085] Local conflicting edit
```



```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git push origin main
To github.com:SeratFatima00/Lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:SeratFatima00/Lab3.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

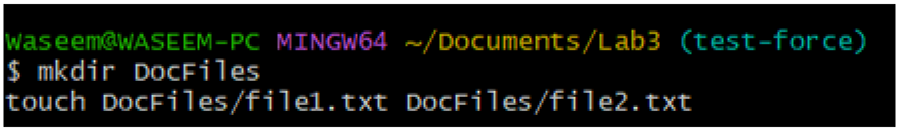


```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git pull --rebase origin main
error: cannot pull with rebase: You have unstaged changes.
error: Please commit or stash them.
```

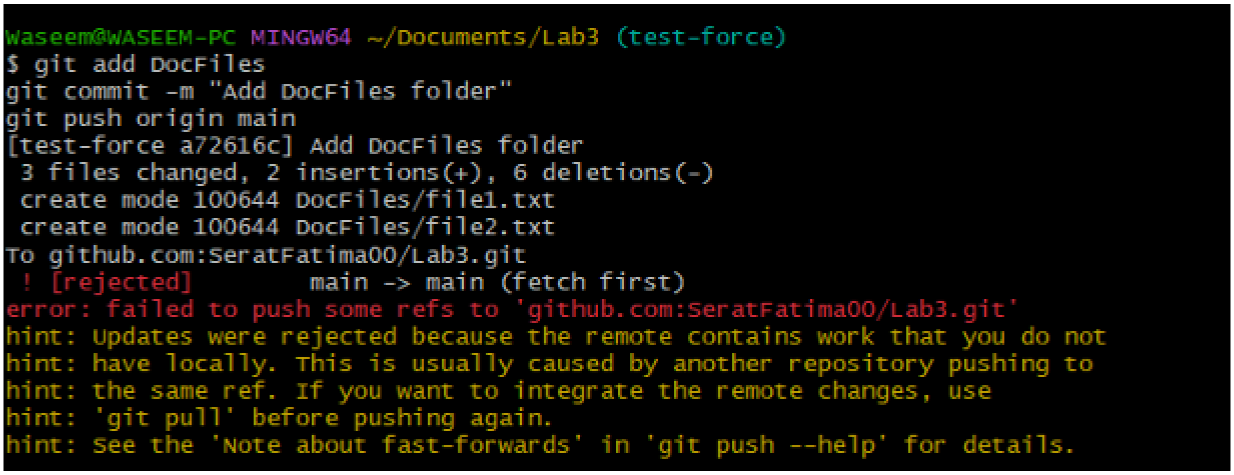


```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git add README.md
git rebase --continue
git push origin main
fatal: no rebase in progress
To github.com:SeratFatima00/Lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:SeratFatima00/Lab3.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

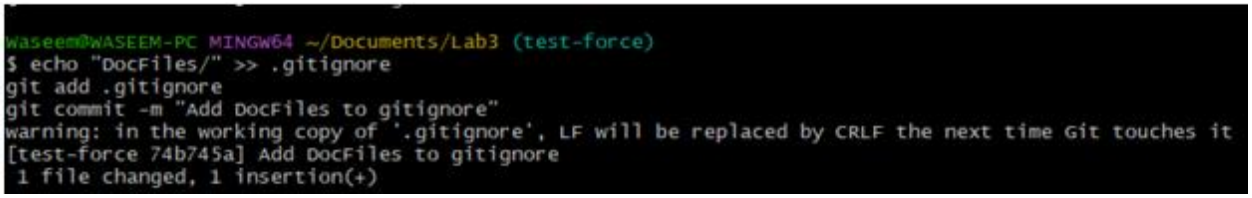
3. Managing Ignored and Tracked Files



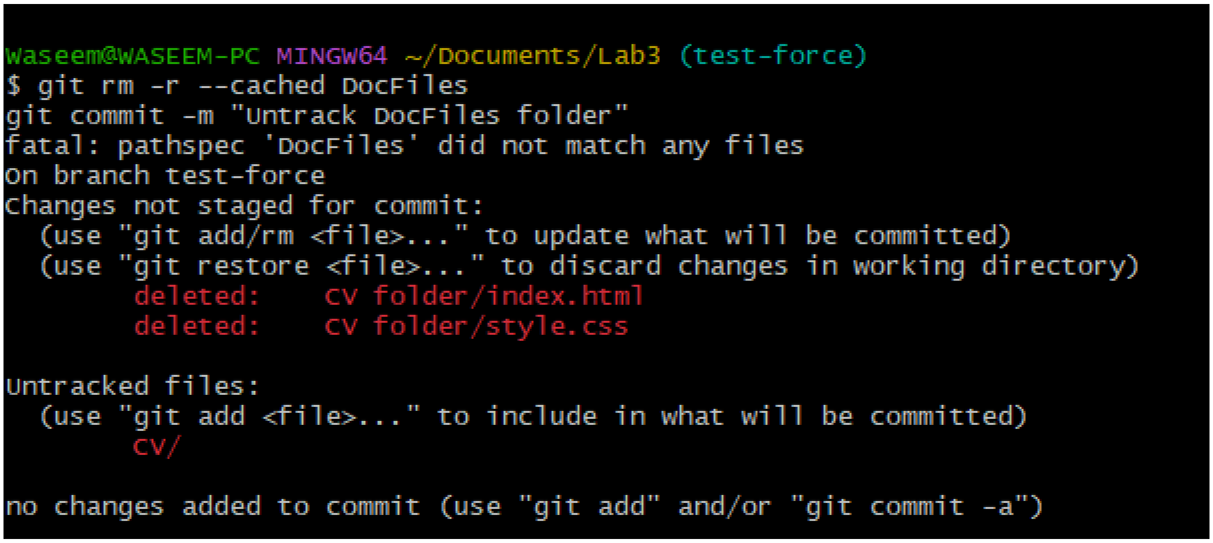
```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ mkdir DocFiles
touch DocFiles/file1.txt DocFiles/file2.txt
```



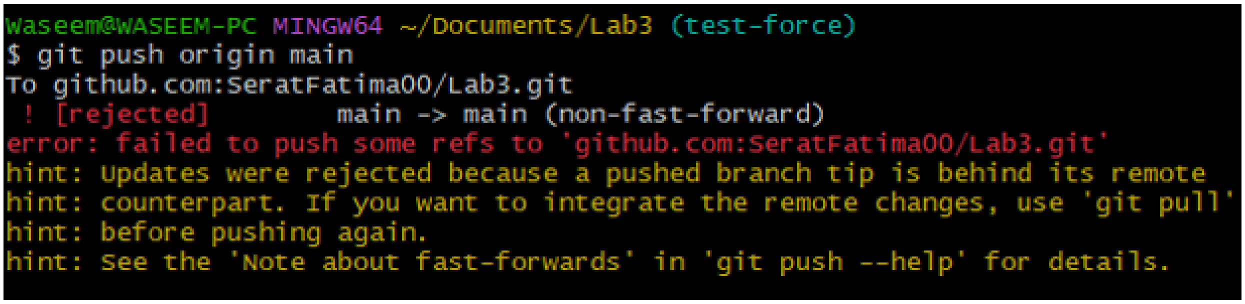
```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git add DocFiles
git commit -m "Add DocFiles folder"
git push origin main
[test-force a72616c] Add DocFiles folder
 3 files changed, 2 insertions(+), 6 deletions(-)
 create mode 100644 DocFiles/file1.txt
 create mode 100644 DocFiles/file2.txt
To github.com:SeratFatima00/Lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:SeratFatima00/Lab3.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```



```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ echo "DocFiles/" >> .gitignore
git add .gitignore
git commit -m "Add DocFiles to gitignore"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
[test-force 74b745a] Add DocFiles to gitignore
 1 file changed, 1 insertion(+)
```

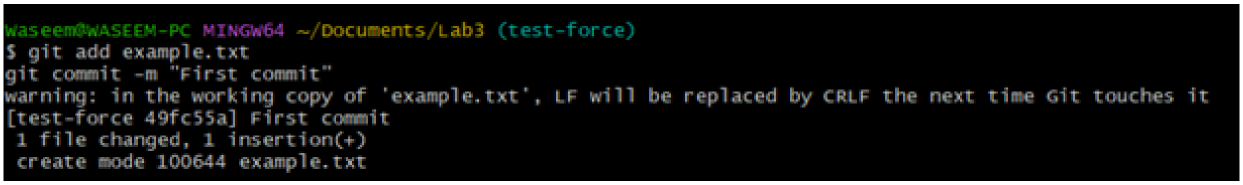


```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git rm -r --cached DocFiles
git commit -m "Untrack DocFiles folder"
fatal: pathspec 'DocFiles' did not match any files
On branch test-force
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    CV folder/index.html
        deleted:    CV folder/style.css

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

no changes added to commit (use "git add" and/or "git commit -a")
```

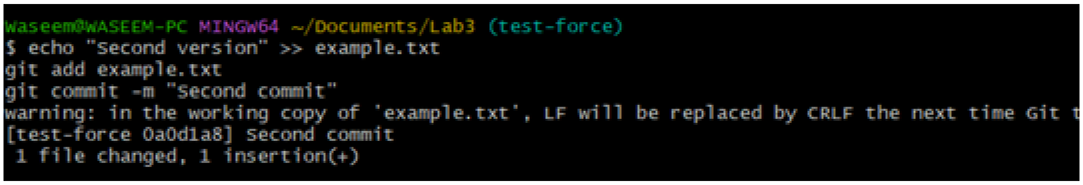


```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git push origin main
To github.com:SeratFatima00/Lab3.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'github.com:SeratFatima00/Lab3.git'
hint: Updates were rejected because a pushed branch tip is behind its remote
hint: counterpart. If you want to integrate the remote changes, use 'git pull'
hint: before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

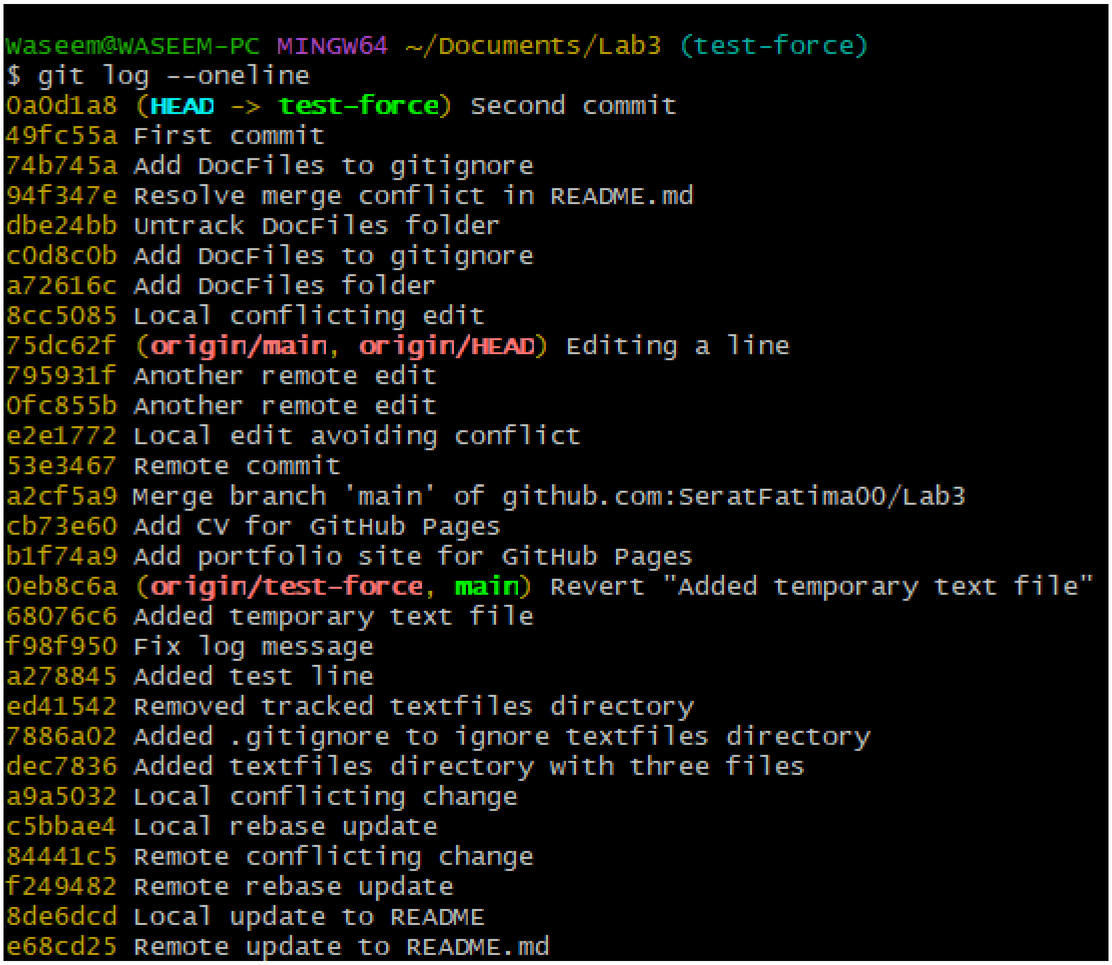
4.Commit History Manipulation and Recovery



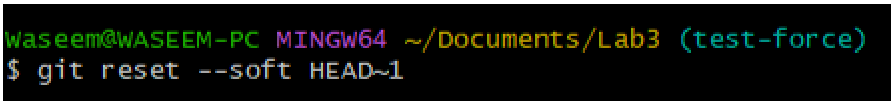
```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git add example.txt
git commit -m "First commit"
warning: in the working copy of 'example.txt', LF will be replaced by CRLF the next time Git touches it
[test-force 49fc55a] First commit
 1 file changed, 1 insertion(+)
 create mode 100644 example.txt
```



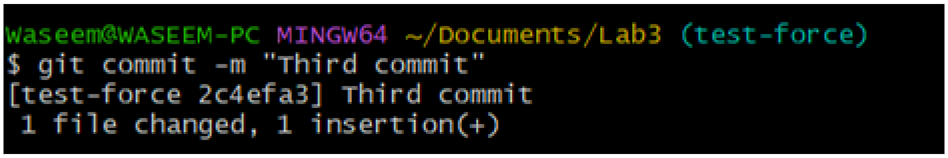
```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ echo "Second version" >> example.txt
git add example.txt
git commit -m "Second commit"
warning: in the working copy of 'example.txt', LF will be replaced by CRLF the next time Git t
[test-force 0a0d1a8] Second commit
 1 file changed, 1 insertion(+)
```



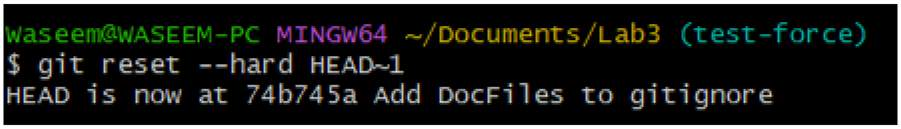
```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git log --oneline
0a0d1a8 (HEAD -> test-force) Second commit
49fc55a First commit
74b745a Add DocFiles to gitignore
94f347e Resolve merge conflict in README.md
dbe24bb Untrack DocFiles folder
c0d8c0b Add DocFiles to gitignore
a72616c Add DocFiles folder
8cc5085 Local conflicting edit
75dc62f (origin/main, origin/HEAD) Editing a line
795931f Another remote edit
0fc855b Another remote edit
e2e1772 Local edit avoiding conflict
53e3467 Remote commit
a2cf5a9 Merge branch 'main' of github.com:SeratFatima00/Lab3
cb73e60 Add CV for GitHub Pages
b1f74a9 Add portfolio site for GitHub Pages
0eb8c6a (origin/test-force, main) Revert "Added temporary text file"
68076c6 Added temporary text file
f98f950 Fix log message
a278845 Added test line
ed41542 Removed tracked textfiles directory
7886a02 Added .gitignore to ignore textfiles directory
dec7836 Added textfiles directory with three files
a9a5032 Local conflicting change
c5bbae4 Local rebase update
84441c5 Remote conflicting change
f249482 Remote rebase update
8de6dcd Local update to README
e68cd25 Remote update to README.md
```



```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git reset --soft HEAD~1
```



```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git commit -m "Third commit"
[test-force 2c4efa3] Third commit
 1 file changed, 1 insertion(+)
```



```
Waseem@WASEEM-PC MINGW64 ~/Documents/Lab3 (test-force)
$ git reset --hard HEAD~1
HEAD is now at 74b745a Add DocFiles to gitignore
```