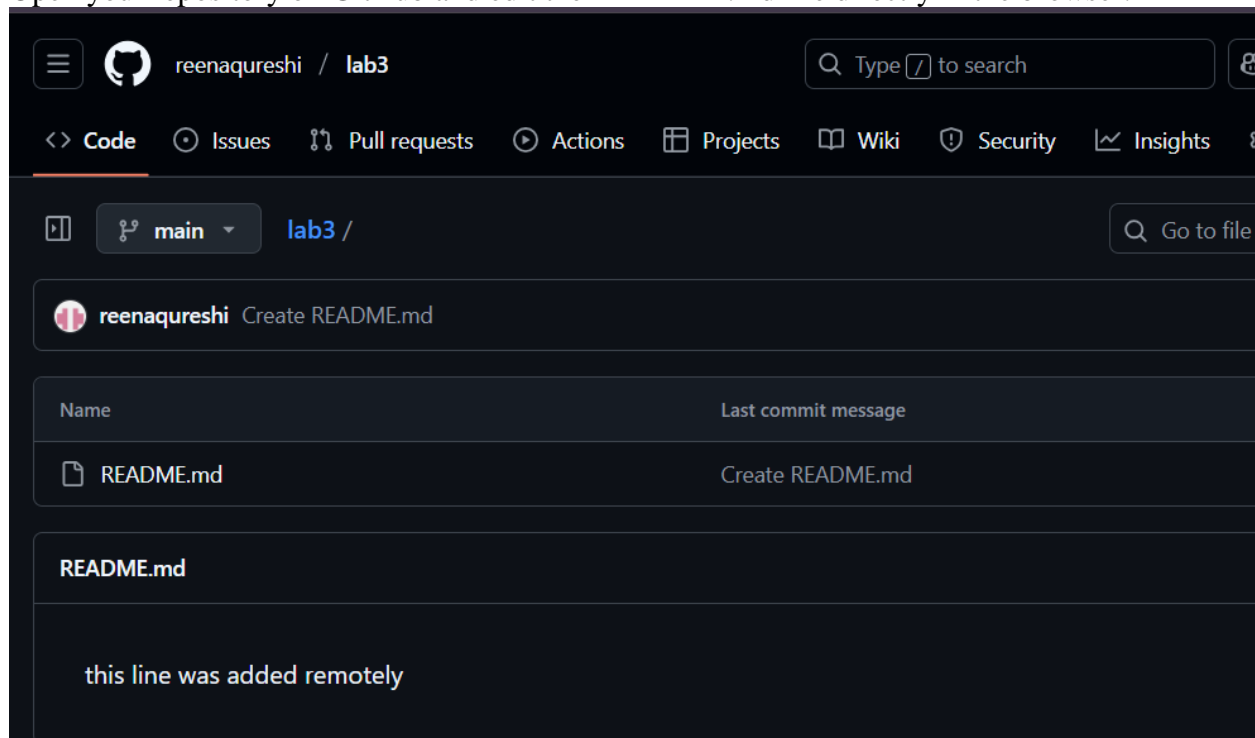


Name : Reena Qureshi
Reg No: 2023-BSE-052
Section: V-B

Cloud Computing: Lab 3

Task 1 – Handling Local and Remote Commit Conflicts (Pull vs Pull --rebase)

Open your repository on GitHub and edit the README.md file directly in the browser.



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

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ echo "This line was added locally." >> README.md
git add README.md
git commit -m "Local update to README"
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[main (root-commit) 9774f30] Local update to README
1 file changed, 1 insertion(+)
create mode 100644 README.md

```

Try to push:
git push origin main
You'll see an error message

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git push origin main
To github.com:reenaqaureshi/lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:reenaqaureshi/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.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)

```

To fix this, pull the latest changes from remote:
git pull --no-rebase origin main

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git pull --no-rebase origin main
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)
Unpacking objects: 100% (3/3), 887 bytes | 8.00 KiB/s, done.
From github.com:reenaqaureshi/lab3
 * branch                main          -> FETCH_HEAD
 * [new branch]          main          -> origin/main
fatal: refusing to merge unrelated histories

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)

```

Push again:

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git push -u origin main
To github.com:reenaqaureshi/lab3.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'github.com:reenaqaureshi/lab3.git'
hint: Updates were rejected because the tip of your current branch is behind
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.

```

this time use rebase instead of merge:

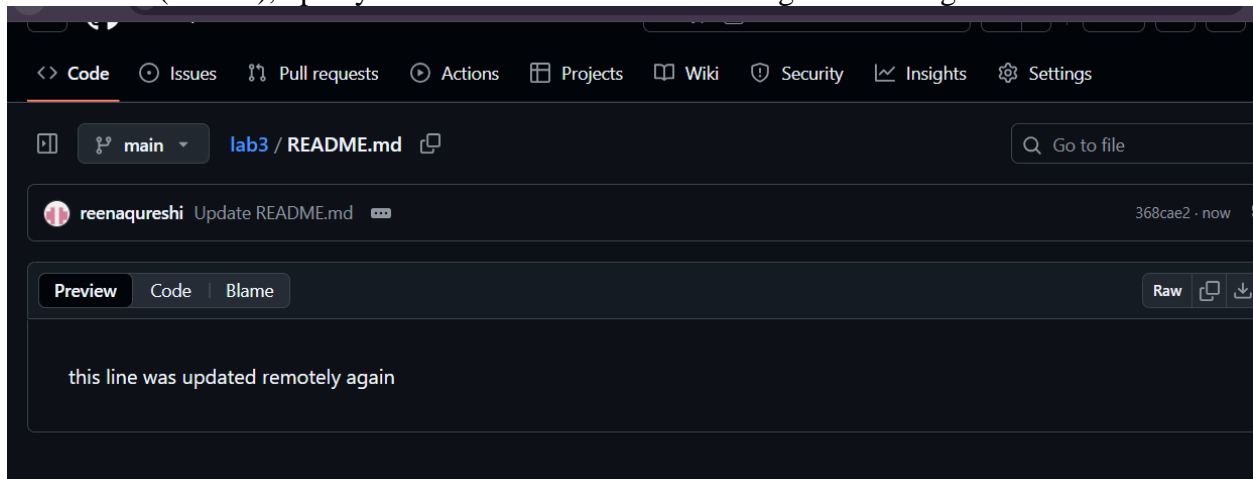
```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git pull --rebase origin main
git push -u origin main
From github.com:reenaqureshi/lab3
* branch          main      -> FETCH_HEAD
Auto-merging README.md
CONFLICT (add/add): Merge conflict in README.md
error: could not apply 9774f30... Local update to README
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config set advice.mergeConflict false"
Could not apply 9774f30... # Local update to README
To github.com:reenaqureshi/lab3.git
 ! [rejected]        main -> main (non-fast-forward)

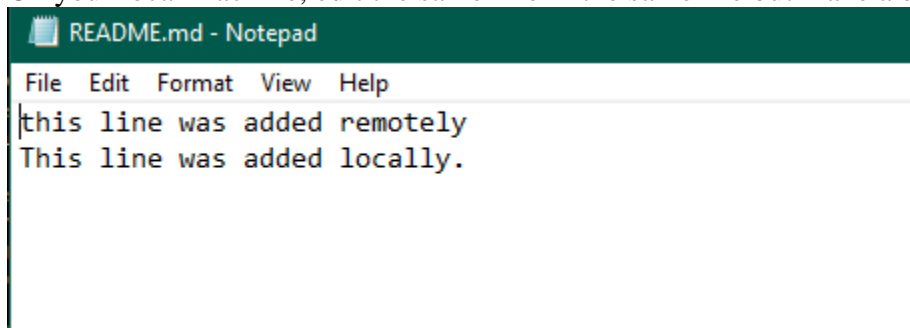
```

Task 2 – Creating and Resolving Merge Conflicts Manually

On GitHub (remote), open your README.md file and change an existing line.



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



Stage and commit your local change:

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main|REBASE 1/1)
$ git add README.md
git commit -m "Local conflicting change"
[detached HEAD ef3db9f] Local conflicting change
1 file changed, 2 insertions(+)

```

Try to push:

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main|REBASE 1/1)
$ git push -u origin main
To github.com:reenaqaureshi/lab3.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:reenaqaureshi/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:

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main|REBASE 1/1)
$ git pull --rebase origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 950 bytes | 32.00 KiB/s, done.
From github.com:reenaqaureshi/lab3
 * branch            main      -> FETCH_HEAD
 6fcb35a..368cae2    main      -> origin/main
fatal: It seems that there is already a rebase-merge directory, and
I wonder if you are in the middle of another rebase. If that is the
case, please try
    git rebase (--continue | --abort | --skip)
If that is not the case, please
    rm -fr ".git/rebase-merge"
and run me again. I am stopping in case you still have something
valuable there.

```

Open the README.md file in your editor — you'll see conflict markers like this:

```
<<<<<<< HEAD
```

This line was updated locally at the same time.

```
=====
```

This line was updated remotely again.

```
>>>>>>> origin/main
```

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main|REBASE 1/1)
$ cat README.md
this line was added remotely
This line was added locally.

```

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

```
git add README.md
```

```
git rebase --continue
```

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main|REBASE 1/1)
$ git add README.md
git rebase --continue
Successfully rebased and updated refs/heads/main.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)

```

Finally, push your changes:

```

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3 (main)
$ git push origin main --force
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), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:reenaqureshi/lab3.git
+ 368cae2...ef3db9f main -> main (forced update)
Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3 (main)

```

Task 3 – Managing Ignored Files with .gitignore and Removing Tracked Files

Create a new folder named textfiles inside your repository:

mkdir textfiles

Inside the textfiles folder, create three text files:

Add and commit the new directory

```

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3 (main)
$ mkdir textfiles

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3 (main)
$ cd textfiles

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3/textfiles (main)
$ echo "File A content" > textfiles/a.txt
$ echo "File B content" > textfiles/b.txt
$ echo "File C content" > textfiles/c.txt
bash: textfiles/a.txt: No such file or directory
bash: textfiles/b.txt: No such file or directory
bash: textfiles/c.txt: No such file or directory

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3/textfiles (main)
$ git add .
git commit -m "Added textfiles directory with three files"
git push origin main
On branch main
Your branch is up to date with 'origin/main'.

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

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3/textfiles (main)
$ |

```

Now, create a .gitignore file in the root of your repository

```

Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3/textfiles (main)
$ echo "textfiles/*" > .gitignore
Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/lab3/textfiles (main)

```

and commit the .gitignore file:

Add

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3/textfiles (main)
$ git add .gitignore
git commit -m "Added .gitignore to ignore textfiles directory"
git push origin main
warning: in the working copy of 'textfiles/.gitignore', LF will be replaced by CRLF the next time Git touches it
[main d9a5204] Added .gitignore to ignore textfiles directory
1 file changed, 1 insertion(+)
 create mode 100644 textfiles/.gitignore
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% (4/4), 370 bytes | 370.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:reenaqureshi/lab3.git
 ef3db9f..d9a5204  main -> main

```

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

reenaqureshi / lab3

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main lab3 /

reenaqureshi Added .gitignore to ignore textfiles directory d9a5204 · 7 minutes ago History

Name	Last commit message	Last commit date
textfiles	Added .gitignore to ignore textfiles directory	7 minutes ago
README.md	Local conflicting change	26 minutes ago

README.md

this line was added remotely This line was added locally.

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

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git rm -r --cached textfiles
rm 'textfiles/.gitignore'

```

Commit and push again:

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git add .
git commit -m "Removed tracked textfiles directory"
git push origin main
warning: in the working copy of 'textfiles/.gitignore', LF will be replaced by CRLF the next time Git touches it
On branch main
Your branch is up to date with 'origin/main'.

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

```

Task 4 – Create Temporary Changes and Use git stash

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
$ |

```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ git add README.md
```

```
git commit -m "Changes the README.md file"
```

```
[feature-branch 3a04a91] Changes the README.md file  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ notepad README.md
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ git checkout main
```

```
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.
```

```
Aborting
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ |
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ git stash
```

```
Saved working directory and index state WIP on feature-branch: 3a04a91 Changes the README.md file
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ git checkout main
```

```
Switched to branch 'main'
```

```
Your branch is up to date with 'origin/main'.
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

```
$ git checkout feature-branch
```

```
Switched to branch 'feature-branch'
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ git status
```

```
On branch feature-branch
```

```
nothing to commit, working tree clean
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

```
$ |
```

```
on(1); 1 detection( )
DFF M1 README.md - Notepad
File Edit Format View Help
DFF M1 this line was added remotely
This line was added locally.
DFF M1 heyyyyyyyyyyyy task 4
to the Temporary test line for stash demo.
or st
DFF M1
d inde
DFF M1
with '
```

Task 5 – Checkout a Specific Commit Using git log

View commit history:

`git log --oneline`

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
$ git log --oneline
3a04a91 (HEAD -> feature-branch) Changes the README.md file
d9a5204 (origin/main, main) Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
```

Copy any previous commit hash (e.g., a1b2c3d).

Checkout that commit:

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (feature-branch)
$ git checkout ef3db9f
Note: switching to 'ef3db9f'.

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 ef3db9f Local conflicting change
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 ((ef3db9f...))
```

To return to your main branch:

git checkout main

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 ((ef3db9f...))
$ git checkout main
Previous HEAD position was ef3db9f Local conflicting change
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

Task 6 – Resetting Commits (Soft vs Hard Reset) (With Verification Steps)

Add a new line in any file and commit it:

Edit any file (e.g., README.md), add a line

git add .

git commit -m "Added test line"

Add another change and commit again:

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ notepad README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git add .
git commit -m "Added test line"
[main 5d719e3] Added test line
1 file changed, 1 insertion(+)

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ notepad README.md

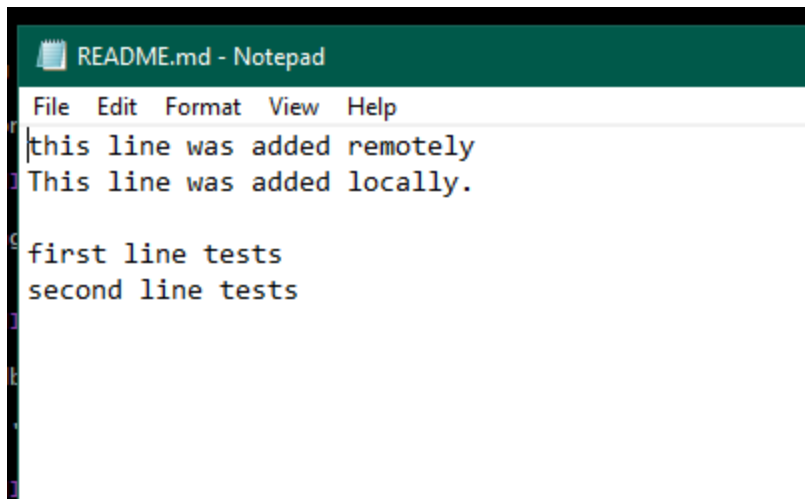
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git add .
git commit -m "Second test commit"
[main fa2db2b] Second test commit
1 file changed, 2 insertions(+), 1 deletion(-)
```

View the history before reset

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git log --oneline
fa2db2b (HEAD -> main) Second test commit
5d719e3 Added test line
d9a5204 (origin/main) Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

Check file contents for both changes:



```
File Edit Format View Help
this line was added remotely
This line was added locally.

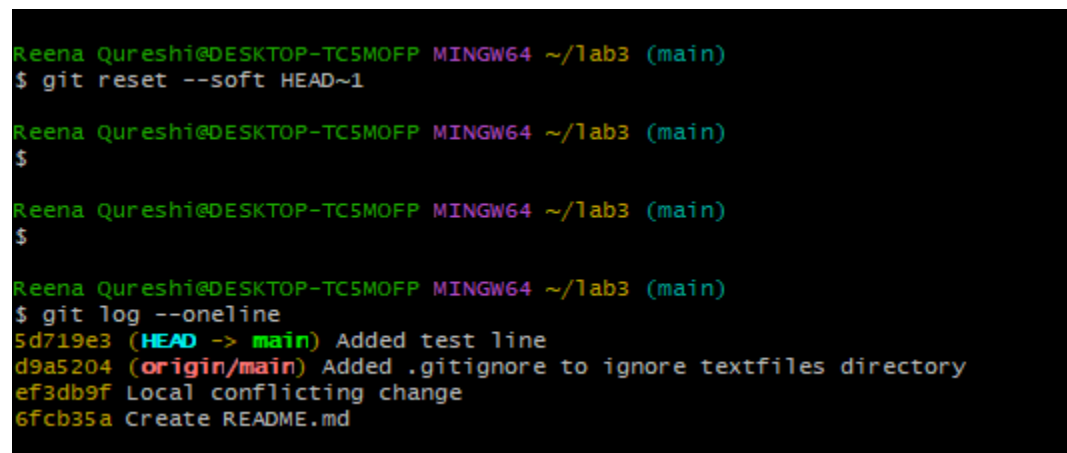
first line tests
second line tests
```

Perform a soft reset (keeps changes in working directory):

`git reset --soft HEAD~1`

check commit history after soft reset:

`git log --oneline`



```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git reset --soft HEAD~1

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git log --oneline
5d719e3 (HEAD -> main) Added test line
d9a5204 (origin/main) Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md
```

Verify changes in the file

```
OP-TC5MOFP M3 README.md - Notepad
File Edit Format View Help
d test line" |this line was added remotely
d test line |This line was added locally.
insertion(+)
OP-TC5MOFP M3
first line tests
OP-TC5MOFP M3 second line tests
nd test commi
nd test commi
insertions(+
OP-TC5MOFP M3
in) Second te
line
n) Added .git
icting change
ME.md
```

Check git status:

git status

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
```

Perform commit

git commit -m "Second Test commit"

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git commit -m "Second test commit"
[main b76dd37] Second test commit
 1 file changed, 2 insertions(+), 1 deletion(-)
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

Perform a hard reset (discards changes completely):

git reset --hard HEAD~

heck commit history after hard reset:

git log --oneline

Verify changes in the file:

```
SMOFP M1 README.md - Notepad
File Edit Format View Help
SMOFP M1 |this line was added remotely
Added test This line was added locally.
Added .git
git change first line tests
SMOFP M1
```

Check git status:

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ |
```

Task 7 – Amending the Last Commit

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ notepad README.md
git status

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
Fix log message
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git log --oneline
6571d6c (HEAD -> main) Fix log message
5d719e3 Added test line
d9a5204 (origin/main) Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

Task 8 – Reverting a Commit (Safe Undo on Remote Branch)

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ echo "temporary text" >> temp.txt
git add .
git commit -m "Added temporary text file"
git push origin main
warning: in the working copy of 'temp.txt', LF will be replaced by CRLF the next time
[main 32c9b9f] Added temporary text file
1 file changed, 1 insertion(+)
create mode 100644 temp.txt
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 947 bytes | 118.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:reenaquareshi/lab3.git
d9a5204..32c9b9f main -> main

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git log --oneline
32c9b9f (HEAD -> main, origin/main) Added temporary text file
6571d6c Fix log message
5d719e3 Added test line
d9a5204 Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git log --oneline
32c9b9f (HEAD -> main, origin/main) Added temporary text file
6571d6c Fix log message
5d719e3 Added test line
d9a5204 Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git revert 32c9b9f
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/Reena Qureshi/lab3/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/Reena Qureshi/lab3/.git/COMMIT_EDITMSG to Unix format...
[main b38b64b] Revert "Added temporary text file"
1 file changed, 1 deletion(-)
delete mode 100644 temp.txt

Reena Qureshi@DESKTOP-TC5MOFP 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), 317 bytes | 105.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:reenaquareshi/lab3.git
32c9b9f..b38b64b main -> main

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)

```

Task 9 – Force Push (With Caution)

Create a new branch:

git checkout -b test-force

Save a screenshot as new_branch.png.

Make and commit a small change.

Save a screenshot as force_commit.png.

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ notepad README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ notepad README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ git add .

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ git commit -m "force psuh"
[test-force 57ab8fc] force psuh
1 file changed, 2 insertions(+), 1 deletion(-)

```

Push it:

git push origin test-force

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ git push origin test-force
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 323 bytes | 323.00 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:reenaquareshi/lab3.git
    b38b64b..57ab8fc test-force -> test-force

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ |

```

Perform a hard reset:

git reset --hard HEAD~1

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ git reset --hard HEAD~1
HEAD is now at b38b64b Revert "Added temporary text file"

```

Push again

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ git push origin test-force
To github.com:reenaquareshi/lab3.git
! [rejected]        test-force -> test-force (non-fast-forward)
error: failed to push some refs to 'github.com:reenaquareshi/lab3.git'
hint: Updates were rejected because the tip of your current branch is behind
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.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (test-force)
$ |

```

Now force-push to remote:

```

Reena Qureshi@DESKTOP-TC5MOFP 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:reenaquareshi/lab3.git
+ 57ab8fc...b38b64b test-force -> test-force (forced update)

```

Task 10 – Running Gitea in GitHub Codespaces via Docker Compose

The image shows a GitHub repository named `forkedGitea` by user `reenaquareshi`, which is a fork of `WaqasSaleem97/Gitea`. The repository is public and has 1 branch (`main`) and 0 tags. The `main` branch is up to date with `WaqasSaleem97/Gitea:main`. The repository has 2 commits, with the latest commit by `WaqasSaleem97` titled "giving the names to databases" 3 weeks ago. The repository contains a `compose.yaml` file. The repository is also featured in the "Gitea Tutorial" section.

The screenshot also shows a VS Code Codespace environment. The Explorer panel shows the `FORKEDGITEA [CODESPACES: URBAN SPACE FIESTA]` workspace with the `compose.yaml` file. The Welcome screen is visible, showing options to start a new file, open a file, or clone a Git repository. The terminal panel shows the command `docker compose up -d` being executed, resulting in the following output:

```
@reenaquareshi →/workspaces/forkedGitea (main) $ docker compose up -d
[+] Running 3/3
 ✓ Network webnet      Created
 ✓ Container gitea      Started
 ✓ Container gitea_db   Started
```

Sign In

Username or Email Address *

reena

Password

[Forgot password?](#)

•••••

☐ Remember This Device

Sign In

or



Sign in with OpenID

Sign in with a passkey
Need an account? [Register now.](#)



Issues

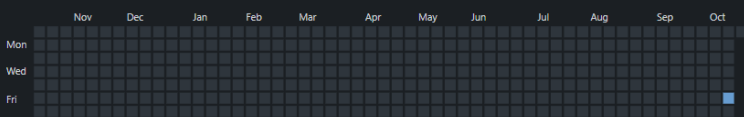
Pull Requests

Milestones

Explore



reena



1 contributions in the last 12 months

Less More



reena created repository [reena/initial_repo](#) 2 days ago



Repository

Organization

Repositories

1



Search repos...

All

1

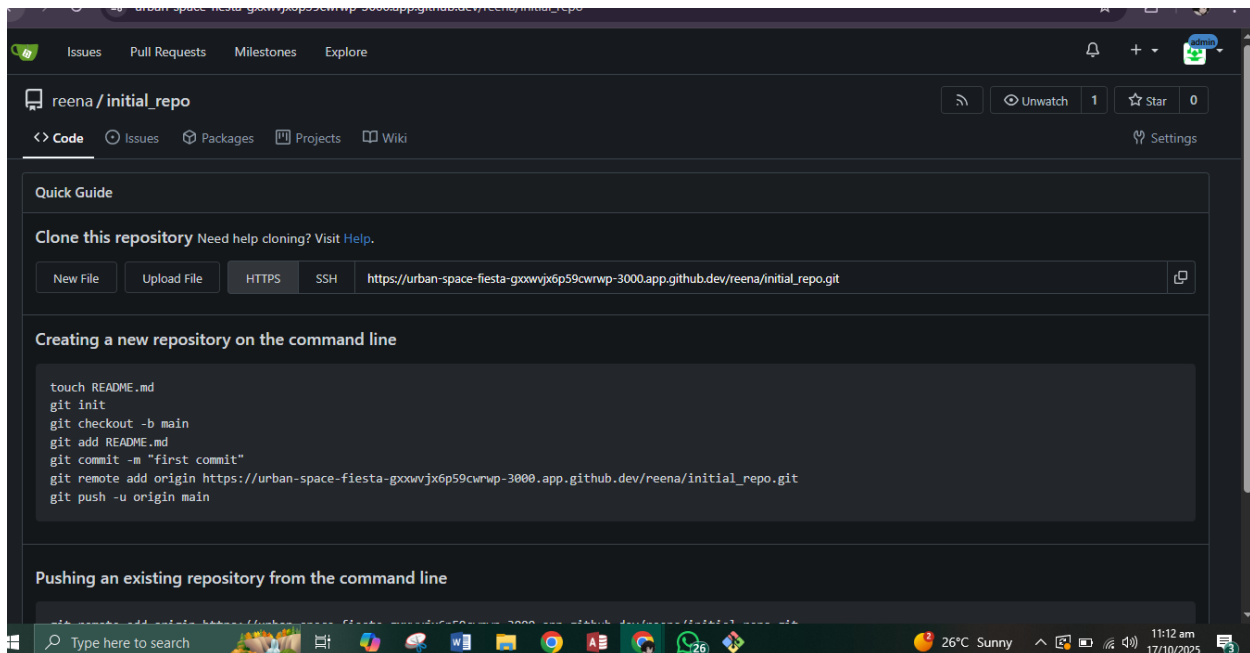
Sources

Forks

Mirrors

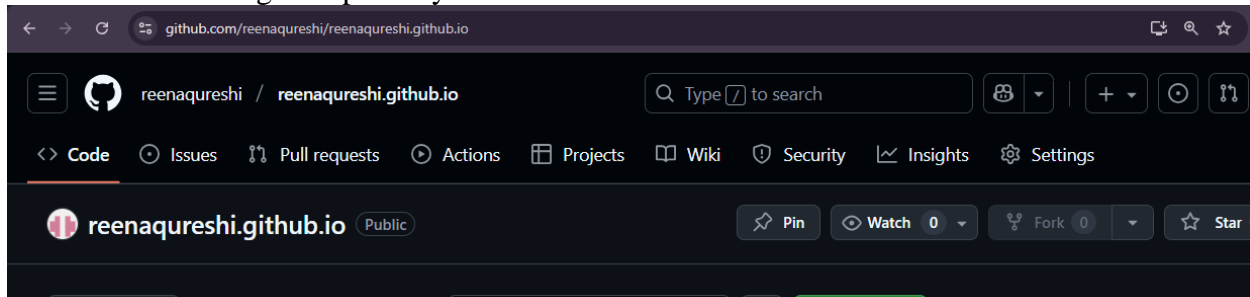
Collaborative

reena/initial_repo



Task 11 – Creating a GitHub Pages Portfolio Site

Create a GitHub Pages Repository



Add Static Website Code

```
git clone git@github.com:reenaqureshi/reenaqureshi.github.io.git
cd reenaqureshi.github.io
Cloning into 'reenaqureshi.github.io'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Push the Files to GitHub

```
Reena Qureshi@DESKTOP-TC5MOFF MINGW64 ~/reenaqureshi.github.io (main)
$ git add .
git commit -m "Add portfolio site for GitHub Pages"
git push -u origin main
On branch main
Your branch is up to date with 'origin/main'.
```

Check GitHub Pages Settings

reenaquareshi / reenaquareshi.github.io

Type to search

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models

GitHub Pages

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

Your site is live at <https://reenaquareshi.github.io/>

Last deployed by reenaquareshi 5 days ago

Visit site

Unpublish site

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.](#)

Visit Your Live Site

reenaquareshi.github.io

REENA QURESHI

Graphic Designer | Software Engineering Student

Rawalpindi, Pakistan • reenaquareshi444@gmail.com • +92 343 6442906

Objective

Creative and detail-oriented Graphic Designer with strong proficiency in Adobe Creative Suite and a working knowledge of UI/UX principles. Experienced in branding, editorial design, and digital illustrations. Eager to contribute creative expertise in a fast-paced, collaborative environment.

Professional Experience

Graphic Designer — Google Developers Group (Rawalpindi)

August 2024 – July 2025

- Designed visuals, social media graphics, and event posters
- Contributed to UI layouts and user flows using Figma and Adobe XD
- Delivered assets that increased engagement by 10%

Freelancer — Remote (Ongoing)

- Designed 30+ real-world inspired projects: logos, magazines, websites
- Developed brand kits and landing pages (HTML/CSS)
- Deployed a personal portfolio using Netlify and GitHub Pages

Education

Exam Evaluation Questions

1. Local vs Remote Conflict Resolution

```

li Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git push -u origin main
To github.com:reenaqureshi/lab3.git
! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'github.com:reenaqureshi/lab3.git'
hint: Updates were rejected because the tip of your current branch is behind
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.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git pull --no-rebase origin main
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)
Unpacking objects: 100% (3/3), 887 bytes | 8.00 KiB/s, done.
From github.com:reenaqureshi/lab3
 * branch          main      -> FETCH_HEAD
 * [new branch]    main      -> origin/main
fatal: refusing to merge unrelated histories

```

Manual Merge Conflict Handling

The screenshot illustrates a manual merge conflict resolution process. It shows a GitHub repository where a pull request has been merged, but the local branch is out of sync. The conflict is resolved by manually editing the file to include both changes, then using git to stage the changes and continue the rebase.

GitHub Interface:

- Repository: lab3 / README.md
- Branch: main
- Commit: Update README.md (368cae2)
- Conflict: this line was updated remotely again

Local File (README.md - Notepad):

```

File Edit Format View Help
this line was added remotely
This line was added locally.

```

Terminal Output:

```

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main|REBASE 1/1)
$ git add README.md
git rebase --continue
Successfully rebased and updated refs/heads/main.

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)

```

Managing Ignored and Tracked Files

Steps:

Create a new folder (e.g., DocFiles) and add several files inside

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ mkdir DocFiles
echo "My first doc" > DocFiles/file1.txt
echo "My second doc" > DocFiles/file2.txt
```

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

Commit and push the folder/files to GitHub.

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git add DocFiles
git commit -m "Added DocFiles folder for Q3"
git push
warning: in the working copy of 'DocFiles/file1.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'DocFiles/file2.txt', LF will be replaced by CRLF the next time Git touches it
[main 6304f43] Added DocFiles folder for Q3
 2 files changed, 2 insertions(+)
 create mode 100644 DocFiles/file1.txt
 create mode 100644 DocFiles/file2.txt
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 443 bytes | 221.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:reenaqureshi/lab3.git
```

Add the folder to your .gitignore file

Commit and push the .gitignore update

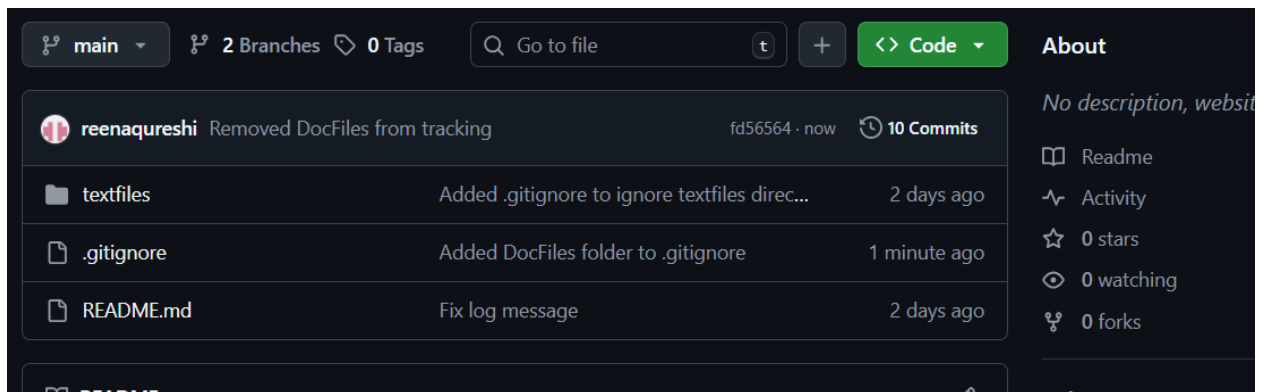
```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ echo "DocFiles/" > .gitignore

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git add .gitignore
git commit -m "Added DocFiles folder to .gitignore"
git push
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
[main 9aefe6b] Added DocFiles folder to .gitignore
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```

Remove the folder from tracking using `git rm -r --cached <folder>`

Commit and push the change, then verify the folder is no longer tracked on GitHub

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git rm -r --cached DocFiles
git commit -m "Removed DocFiles from tracking"
git push
rm 'DocFiles/file1.txt'
rm 'DocFiles/file2.txt'
[main fd56564] Removed DocFiles from tracking
 2 files changed, 2 deletions(-)
 delete mode 100644 DocFiles/file1.txt
 delete mode 100644 DocFiles/file2.txt
```



Commit History Manipulation and Recovery

Hard and Soft reset

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ notepad README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git add .
git commit -m "Added test line"
[main 5d719e3] Added test line
1 file changed, 1 insertion(+)

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ notepad README.md

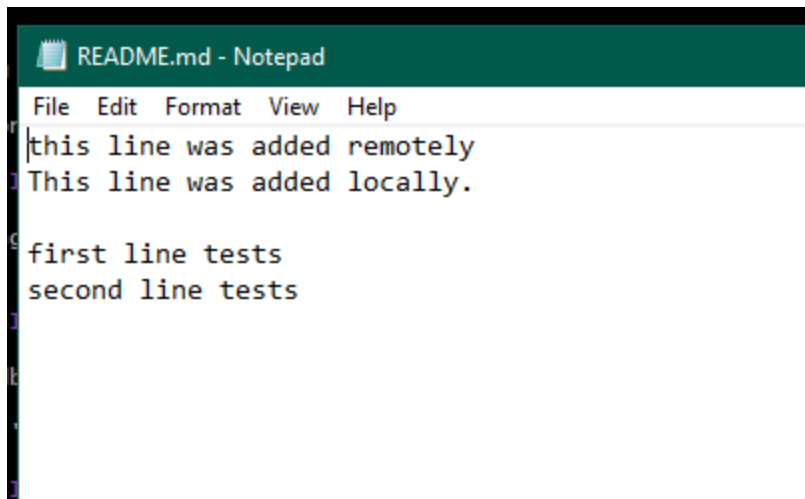
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git add .
git commit -m "Second test commit"
[main fa2db2b] Second test commit
1 file changed, 2 insertions(+), 1 deletion(-)
```

View the history before reset

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git log --oneline
fa2db2b (HEAD -> main) Second test commit
5d719e3 Added test line
d9a5204 (origin/main) Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$
```

Check file contents for both changes:



```
README.md - Notepad
File Edit Format View Help
this line was added remotely
This line was added locally.

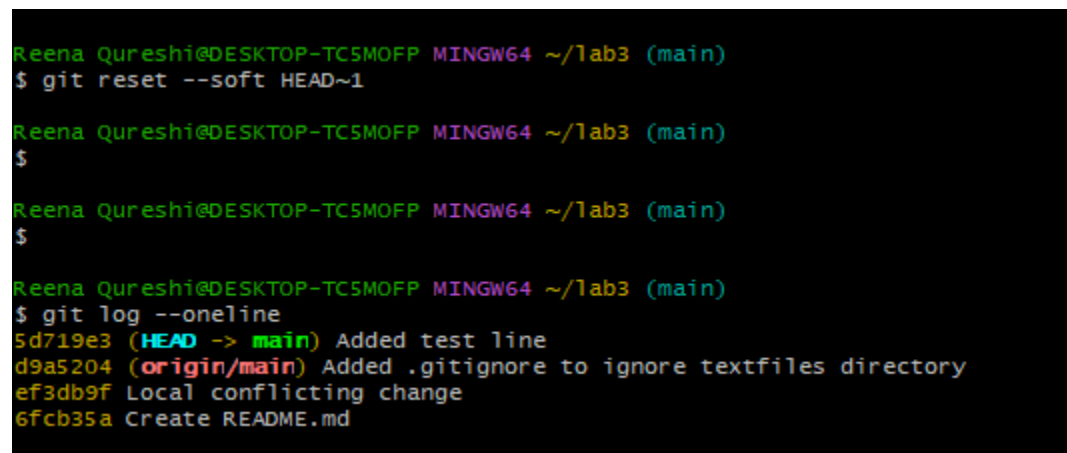
first line tests
second line tests
```

Perform a soft reset (keeps changes in working directory):

```
git reset --soft HEAD~1
```

check commit history after soft reset:

```
git log --oneline
```



```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git reset --soft HEAD~1

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git log --oneline
5d719e3 (HEAD -> main) Added test line
d9a5204 (origin/main) Added .gitignore to ignore textfiles directory
ef3db9f Local conflicting change
6fcb35a Create README.md
```

Verify changes in the file

```
OP-TC5MOFP M3 README.md - Notepad
File Edit Format View Help
d test line" |this line was added remotely
d test line |This line was added locally.
insertion(+)
OP-TC5MOFP M3
first line tests
OP-TC5MOFP M3 second line tests
nd test commi
nd test commi
insertions(+
OP-TC5MOFP M3
in) Second te
line
n) Added .git
icting change
ME.md
```

Check git status:
git status

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
```

Perform commit
git commit -m "Second Test commit"

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git commit -m "Second test commit"
[main b76dd37] Second test commit
 1 file changed, 2 insertions(+), 1 deletion(-)
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
```

Perform a hard reset (discards changes completely):
git reset --hard HEAD~
heck commit history after hard reset:

```
SMOFP M1 README.md - Notepad
File Edit Format View Help
SMOFP M1 |this line was added remotely
added test This line was added locally.
added .git
g change first line tests
SMOFP M1
```

Check git status:

```
Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Reena Qureshi@DESKTOP-TC5MOFP MINGW64 ~/lab3 (main)
$ |
```