

*Fatima Jinnah Women University*



*Cloud Computing  
Lab 3*

*Submitted By :*

*Tushal Nasir*

*Section :*

*A*

*Roll No:*

*38*

*Submitted To:*

*Sir Shoail*

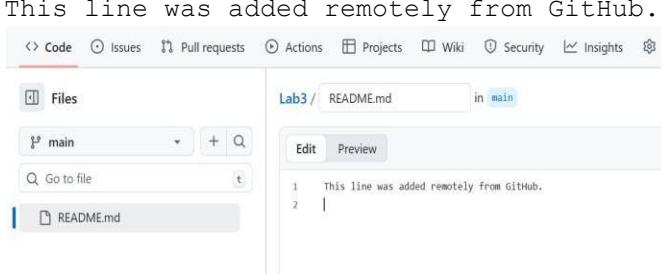
## LAB TITLE: Working with Git History, Stashing, and Reverting Commits

### TASK:

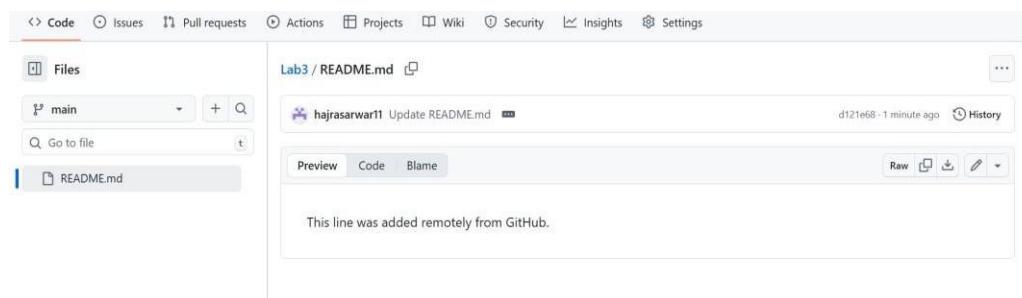
#### **Task 1: Handling Local and Remote Commit Conflicts (Pull vs Pull --rebase)**

##### Procedure

1. **Create a Remote Commit**
  - Open your repository on GitHub.
  - Edit the **README.md** file directly in the browser.
  - Add the following line:  
This line was added remotely from GitHub.



Remote update to README



- Save a screenshot showing this change as **remote\_edit.png**.
2. **Create a Local Commit**
  - On your **local machine**, open the same repository folder.

```

HP@DESKTOP-JFEH5TK MINGW64 /d
$ cd "D:/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3"

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3 (main)
$ git clone https://github.com/hajrasarwar11/Lab3.git
Cloning into 'Lab3'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.

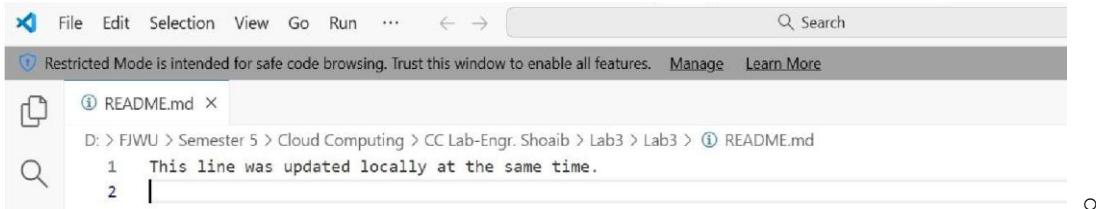
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3 (main)
$ cd Lab3

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git remote -v
origin https://github.com/hajrasarwar11/Lab3.git (fetch)
origin https://github.com/hajrasarwar11/Lab3.git (push)

```

- Edit the **README.md** file and add:

This line was added locally.



Stage and commit the change:

```

git add README.md
git commit -m "Local update to README"

```

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git add README.md

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git commit -m "Local update to README"
[main e2f594f] Local update to README
1 file changed, 1 insertion(+), 1 deletion(-)

```

- Save a screenshot of your terminal as **local\_commit.png**.

### 3. Attempt to Push Local Changes

- Try to push the changes:

```
git push origin main
```

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main|MERGING)
$ git push -u origin main
!> https://github.com/hajrasarwar11/Lab3.git
! [rejected]          main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/hajrasarwar11/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.

```

- Save a screenshot of this error as **push\_error.png**.

### 4. Resolve Using Git Merge

- Pull the latest changes using **merge**:

```
git pull --no-rebase origin main
```

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git pull --no-rebase origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 959 bytes | 20.00 KiB/s, done.
From https://github.com/hajrasarwar11/Lab3
 * branch      main       -> FETCH_HEAD
   e2f594f..22fa12a  main       -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

```

Git merges your local and remote commits, creating a **merge commit**. ○ Save a screenshot of this as **merge\_commit.png**.

```

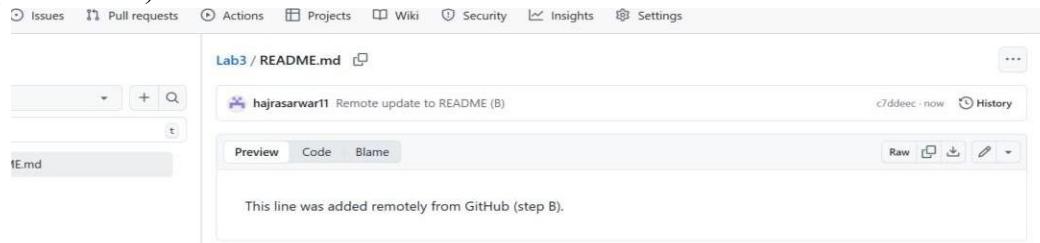
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 654 bytes | 50.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hajrasarwar11/Lab3.git
  22fa12a..9339f5f  main -> main

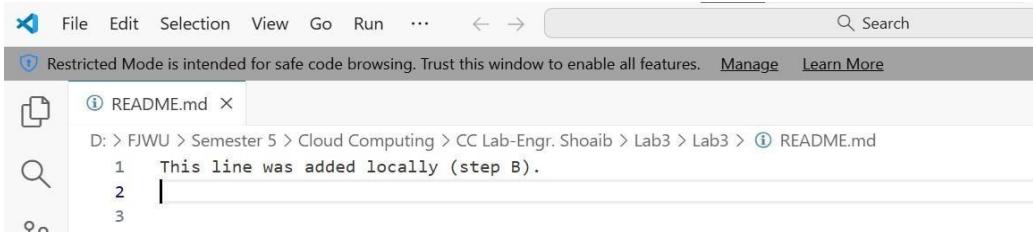
```

- Save a screenshot of successful push as **push\_after\_merge.png**.
- Push the merged changes: `git push -u origin main`

## 5. Resolve Using Git Rebase

- Again, edit **README.md** on GitHub to add a new line (remote edit).





A screenshot of a code editor window titled "README.md". The window shows a single line of text: "1 This line was added locally (step B.)". The status bar at the bottom indicates the path: "D: > FJWU > Semester 5 > Cloud Computing > CC Lab-Engr. Shoaib > Lab3 > Lab3 > README.md". There are icons for file operations like copy, search, and refresh.

This time, pull with **rebase**: `git pull --rebase origin main`

- Git will **replay local commits** on top of remote commits, keeping the history **linear**. ○ Save a screenshot as **rebase\_pull.png**.
- Push the changes again:

`git push -u origin main`



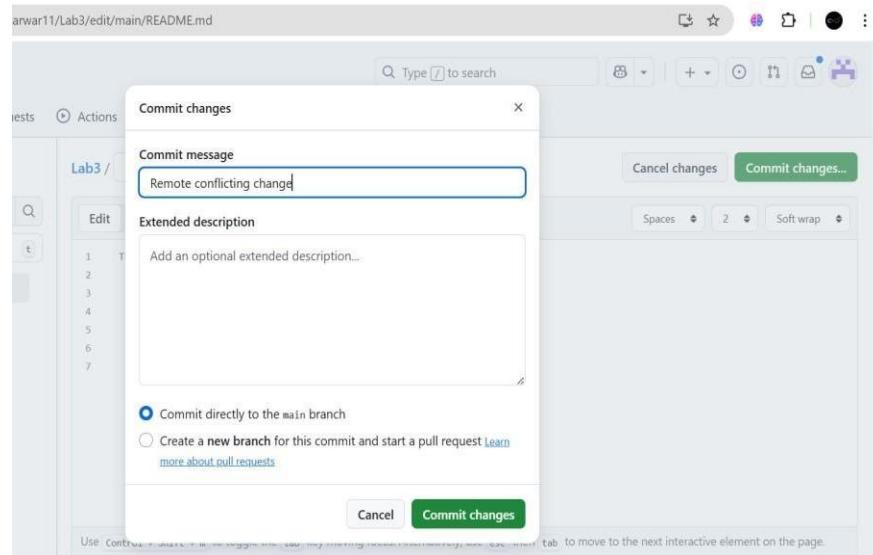
```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 67.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hajrasarwar11/Lab3.git
  df97ab4..ec04ce7 main -> main
branch 'main' set up to track 'origin/main'.
```

○ Save a screenshot as **push\_after\_rebase.png**.

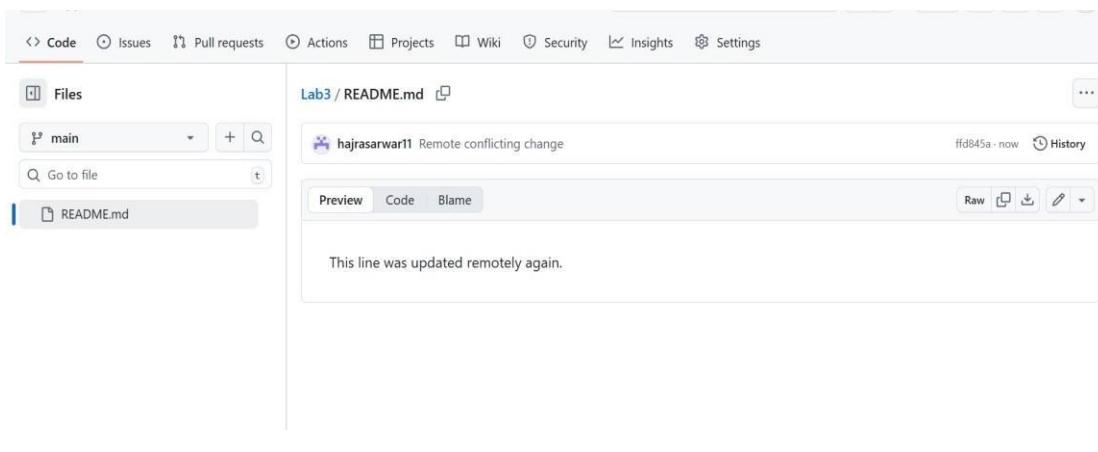
## **Task 2: Creating and Resolving Merge Conflicts Manually**

### **Procedure**

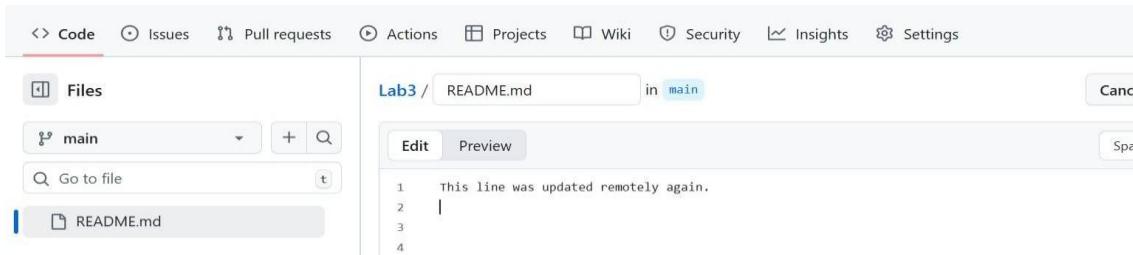
1. **Create a Remote Conflict** ○ On **GitHub**, open your **README.md** file.



- Save a screenshot as **remote\_conflict\_edit.png**.



- Change an existing line from: **o This line was added remotely from GitHub.** to: **This line was updated remotely again.**



- Commit the change with the message:

Remote conflicting change

## 2. Create a Local Conflict

- On your **local machine**, edit the same line in **README.md** but make a different change:

This line was updated locally at the same time.

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git add README.md
git commit -m "Local conflicting change"
[main 6a6ab10] Local conflicting change
 1 file changed, 1 insertion(+), 2 deletions(-)
```

- Save a screenshot as **local\_conflict\_commit.png**.

### 3. Attempt to Push Local Changes

- Try pushing your commit:  
git push -u origin main
- The push will be **rejected** because of conflicting remote changes.

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git push -u origin main
To https://github.com/hajrasarwar11/Lab3.git
 ! [rejected]          main -> main (fetch first)
error: failed to push some refs to 'https://github.com/hajrasarwar11/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.
```

- Save a screenshot of this error as **conflict\_push\_error.png**.



Save a screenshot of your VS Code edit as **local\_conflict\_edit.png**.

- Stage and commit your local change:  
git add README.md  
git commit -m "Local conflicting change"

### 4. Pull with Rebase to Detect Conflict

- Pull remote changes using rebase: git pull --rebase origin main

- Git will detect a conflict and pause the rebase

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git pull --rebase origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 940 bytes | 21.00 KiB/s, done.
From https://github.com/hajrasarwar11/Lab3
 * branch      main      -> FETCH_HEAD
   ec04ce7..ffd845a  main      -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 6a6ab10... Local conflicting change
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 6a6ab10... # Local conflicting change
```

- Save a screenshot of this conflict message as **conflict\_message.png**.

5. **Manually Resolve the Conflict** ○ Open README.md in your editor. You will see conflict markers like:

```
<<<<< HEAD
This line was updated locally at the same time.
=====
This line was updated remotely again. >>>>>
origin/main
```

The screenshot shows a code editor window with a toolbar at the top. The main area displays a file named 'README.md'. A conflict is present between two changes. The first change, 'HEAD (Current Change)', is highlighted in green and contains the line 'This line was updated locally at the same time.'. The second change, 'origin/main', is highlighted in blue and contains the line 'This line was updated remotely again.'. Between these lines are conflict markers: '<<<<< HEAD' on the left and '>>>>> origin/main' on the right. A status bar at the bottom of the editor window indicates 'Restricted Mode is intended for safe code browsing. Trust this window to enable all features.'

Manually edit the file to resolve the conflict (for example, merge both ideas into one correct line).

The screenshot shows the same code editor window after the conflict has been resolved. The file 'README.md' now contains the following text:  
1  
2 This line was updated remotely again.  
3 This line was updated locally at the same time.  
4  
5  
6

- Save a screenshot of your resolved file as **resolved\_readme.png**.

6. **Mark Conflict as Resolved and Continue** ○ After resolving, mark the file as resolved and continue the rebase:

```

git add README.md git
rebase -continue
MINGW64:/d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3
local conflicting change
# Conflicts:
#   README.md

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# interactive rebase in progress; onto ffd845a
# Last command done (1 command done):
#   pick 6a6ab10 # Local conflicting change
# No commands remaining.
# You are currently rebasing branch 'main' on 'ffd845a'.
#
# Changes to be committed:
#   modified:   README.md
#
~
~

```

:wq then enter. ○ Save a screenshot as **rebase\_continue.png**.

7. **Push Resolved Changes** ○ Finally, push your updated changes to the remote repository:

```

git push -u origin main
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 333 bytes | 55.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hajrasarwar11/Lab3.git
  ffd845a..9d47d7a main -> main
branch 'main' set up to track 'origin/main'.

```

- Save a screenshot of successful push as **push\_after\_resolve.png**.

## **Task3: Managing Ignored Files with .gitignore and Removing Tracked Files**

Steps:

1. Create a new folder named **textfiles** inside your repository: `mkdir textfiles`

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ mkdir textfiles

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ ls -la
total 9
drwxr-xr-x 1 HP 197609 0 Oct 17 10:33 .
drwxr-xr-x 1 HP 197609 0 Oct 17 10:34 ..
drwxr-xr-x 1 HP 197609 0 Oct 17 10:24 .git/
-rw-r--r-- 1 HP 197609 100 Oct 17 10:19 README.md
drwxr-xr-x 1 HP 197609 0 Oct 17 10:33 textfiles/

```

2. Inside the **textfiles** folder, create three text files:

```

echo "File A content" > textfiles/a.txt echo
"File B content" > textfiles/b.txt

```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git add .
git commit -m "Added textfiles directory with three files"
git push origin main
warning: in the working copy of 'textfiles/a.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'textfiles/b.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'textfiles/c.txt', LF will be replaced by CRLF the next time Git touches it
[main 149c401] 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
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 467 bytes | 66.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hajrasarwar11/Lab3.git
 9d47d7a..149c401 main -> main
```

Screenshot: push\_textfiles.png

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ echo "File A content" > textfiles/a.txt

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ echo "File B content" > textfiles/b.txt

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ echo "File C content" > textfiles/c.txt

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ ls textfiles
a.txt b.txt c.txt

echo "File C content" > textfiles/c.txt
```

3. **Add and commit the new directory:** git add . git commit -m "Added textfiles directory with three files"  
git push origin main

4. **Create a .gitignore file in the root of your repository to ignore the textfiles folder:**

```
echo "textfiles/*" > .gitignore
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ echo "textfiles/*" > .gitignore
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ cat .gitignore
textfiles/*
```

5. **Add and commit the .gitignore file:** git add .gitignore git commit -m "Added .gitignore to ignore textfiles directory"

```
git push origin main
```

**8. Commit and push again:** git add . git commit -m "Removed tracked textfiles directory" git push origin main

*Screenshot: rm\_cached\_push.png*

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

The screenshot shows a GitHub repository named 'Lab3'. The repository is public and has 1 branch and 0 tags. The main branch contains three commits:

- Commit fdd2375 by hajrasarwar11: Removed tracked textfiles directory (1 minute ago)
- Commit 9 minutes ago: Added .gitignore to ignore textfiles directory
- Commit 34 minutes ago: Local conflicting change

The README file is visible with the following content:

```
This line was updated remotely again. This line was updated locally at the same time.
```

The repository details on the right show 15 commits, 0 stars, 0 forks, and no releases published.

## Task 4: Create Temporary Changes and Use

```
git checkout -b feature-branch
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoai... (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoai... (feature-branch)
$ git branch
* feature-branch
  main
```

**Modify a file (for example, `README.md`) by adding a few test lines.**

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoai... (main)
$ git add .
$ git commit -m "Removed tracked textfiles directory"
[main fdd2375] Removed tracked textfiles directory
 1 files changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 textfiles/a.txt
 delete mode 100644 textfiles/b.txt
 delete mode 100644 textfiles/c.txt
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 244 bytes | 244.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1) completed with 1 local object.
To d:\FJWU\Semester 5\Cloud Computing\CC Lab-Engr. Shoai... (main)
   b869d8a..fdd2375 main -> main
 3  This line was updated locally at the same time.
 4
 5  This is a test line added in feature-branch.
 6  Another test line for demonstration.
 7
 8
 9
```

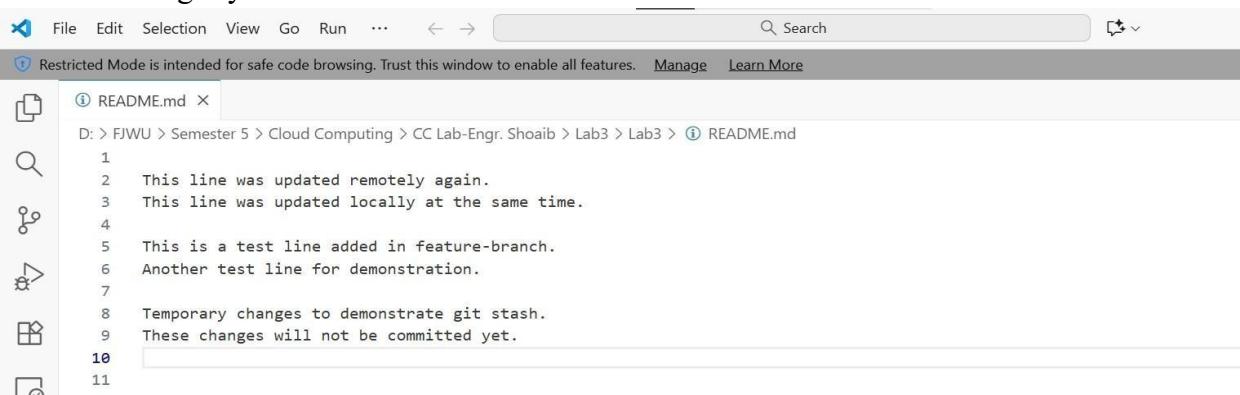
Then stage and commit your changes:

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

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (feature-branch)  
$ git add README.md  
git commit -m "Changes the README.md file"  
[feature-branch c6fbafc] Changes the README.md file  
1 file changed, 4 insertions(+)
```

*Save screenshot as modified\_readme.png*

3. Make another modification to the same file (`README.md`) — add some new lines, but **do not commit or stash** these changes yet.



The screenshot shows a code editor window with the file `README.md` open. The file contains the following content:

```
D: > FJWU > Semester 5 > Cloud Computing > CC Lab-Engr. Shoaib > Lab3 > Lab3 > README.md  
1  
2 This line was updated remotely again.  
3 This line was updated locally at the same time.  
4  
5 This is a test line added in feature-branch.  
6 Another test line for demonstration.  
7  
8 Temporary changes to demonstrate git stash.  
9 These changes will not be committed yet.  
10  
11--
```

4. Try switching to another branch (e.g., `main`): `git checkout main`

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/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
```

*Save  
screenshot as checkout\_error.png*

5. Stash the uncommitted changes to temporarily save them: `git stash`

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (feature-branch)  
$ git stash  
Saved working directory and index state WIP on feature-branch: c6fbafc Changes the README.md file
```

*Save screenshot as stash\_command.png*

6. Switch branches again — now it should work successfully: `git checkout main`

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (feature-branch)  
$ git checkout main  
Switched to branch 'main'  
Your branch is up to date with 'origin/main'.
```

*Save screenshot as branch\_switched.png*

7. Return to your previous branch (`feature-branch`):

```
git checkout feature-branch
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)  
$ git checkout feature-branch  
Switched to branch 'feature-branch'  
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (feature-branch)  
$
```

*Save screenshot as back\_to\_feature.png* 8. Check the current working directory status: `git status`

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (feature-branch)  
$ git status  
On branch feature-branch  
nothing to commit, working tree clean
```

It

should show: nothing to commit, working tree clean *Save screenshot as  
status\_clean.png*

## 9. Restore your stashed changes: git stash pop

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoab/Lab3/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

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (008ccee943c57456c18faaf52900f7d20bedad9f)
```

Save

screenshot as **stash\_pop.png** 10. Confirm that your previous edits are back in the file.

## Task 5: Checkout a Specific Commit Using `git log`

Steps:

1. View commit history In Git Bash, run:

```
git log --oneline
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoab/Lab3/Lab3 (feature-branch)
$ git log --oneline
c6fbafc (HEAD -> feature-branch) Changes the README.md file
fdd2375 (origin/main, origin/HEAD, main) Removed tracked textfiles directory
b869d8a Added .gitignore to ignore textfiles directory
149c401 Added textfiles directory with three files
9d47d7a Local conflicting change
ffd845a Remote conflicting change
ec04ce7 Local update to README (B)
df97ab4 Remote update to README (B)
c7ddeec Remote update to README (B)
9339f5f Resolve merge conflict in README.md (merge remote changes)
77e6bd2 Local update to README (A)
22fa12a Remote update to README (A)
e2f594f Local update to README
c1da5cc Update README.md
d121e68 Update README.md
28e9511 Initial commit
```

This displays a list of all commits with their short hashes and messages.

Screenshot: **log\_before\_checkout.png**

2. Copy a commit hash

Select any previous commit hash from the log (for example: `a1b2c3d`). **28e9511**

3. Checkout the specific commit Run:

```
git checkout <commit-hash> Replace <commit-hash>
```

with the commit you copied.

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoab/Lab3/Lab3 (feature-branch)
$ git checkout 28e9511
Note: switching to '28e9511'.

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 28e9511 Initial commit
```

You are now in a **detached HEAD** state, meaning changes are not attached to any branch.

Screenshot: **detached\_head.png**

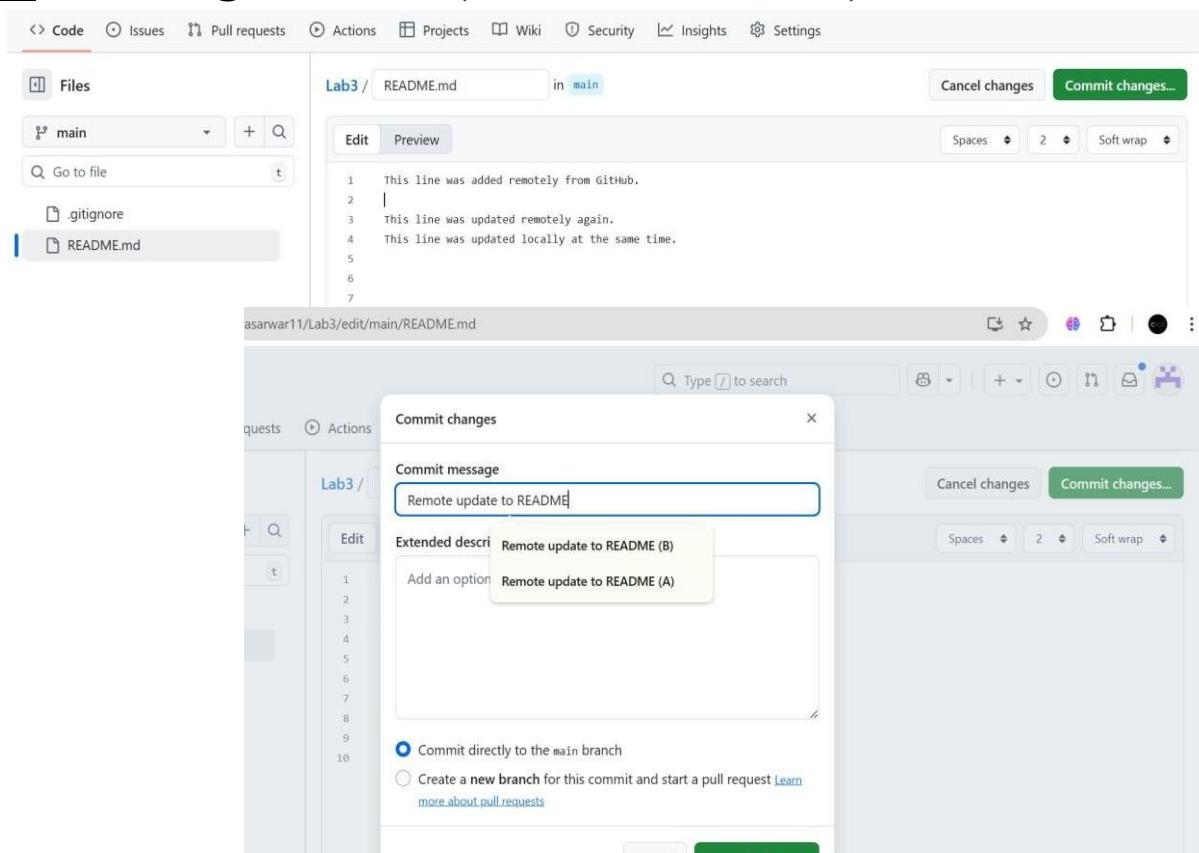
#### 4. Return to the main branch

To go back to the latest branch state, run:

```
git checkout main
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 ((28e9511...))
$ git checkout main
Previous HEAD position was 28e9511 Initial commit
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

*Screenshot: back\_to\_main.png*

### Task 6: Resetting Commits (Soft vs Hard Reset)



The screenshot shows the GitHub interface for a repository named 'Lab3'. On the left, the 'Files' sidebar shows files like '.gitignore' and 'README.md'. The main area displays the contents of 'README.md':

```
1 This line was added remotely from GitHub.
2 |
3 This line was updated remotely again.
4 This line was updated locally at the same time.
5
6
7
```

A modal window titled 'Commit changes' is open over the main view. It contains a 'Commit message' field with the value 'Remote update to README|'. Below it is an 'Extended descri' dropdown menu with two options: 'Remote update to README (B)' and 'Remote update to README (A)'. At the bottom of the modal, there are radio buttons for 'Commit directly to the main branch' (selected) and 'Create a new branch for this commit and start a pull request'. A 'Commit changes...' button is at the bottom right of the modal.

```
git add .
git commit -m "Added test line"
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git add .

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git commit -m "Added test line"
[main 685e2a5] Added test line
 1 file changed, 2 insertions(+)
```

*Screenshot: first\_commit.png*

#### 2. Add and Commit the Second Change

##### Steps and Verification

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git add .
git commit -m "Second test commit"
[main 97772a1] Second test commit
 1 file changed, 2 insertions(+)

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ |

```

*Screenshot: second\_commit.png*

### 3. View Commit History Before Reset

```

git log --oneline
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git log --oneline
97772a1 (HEAD -> main) Second test commit
685e2a5 Added test line
fdd2375 (origin/main, origin/HEAD) Removed tracked textfiles directory
b869d8a Added .gitignore to ignore textfiles directory
149c401 Added textfiles directory with three files
9d47d7a Local conflicting change
ffd845a Remote conflicting change
ec04ce7 Local update to README (B)
df97ab4 Remote update to README (B)
c7ddeec Remote update to README (B)
9339f5f Resolve merge conflict in README.md (merge remote changes)
77e6bd2 Local update to README (A)
22fa12a Remote update to README (A)
e2f594f Local update to README
c1da5cc Update README.md
d121e68 Update README.md
28e9511 Initial commit

```

*Screenshot: log\_before\_reset.png*

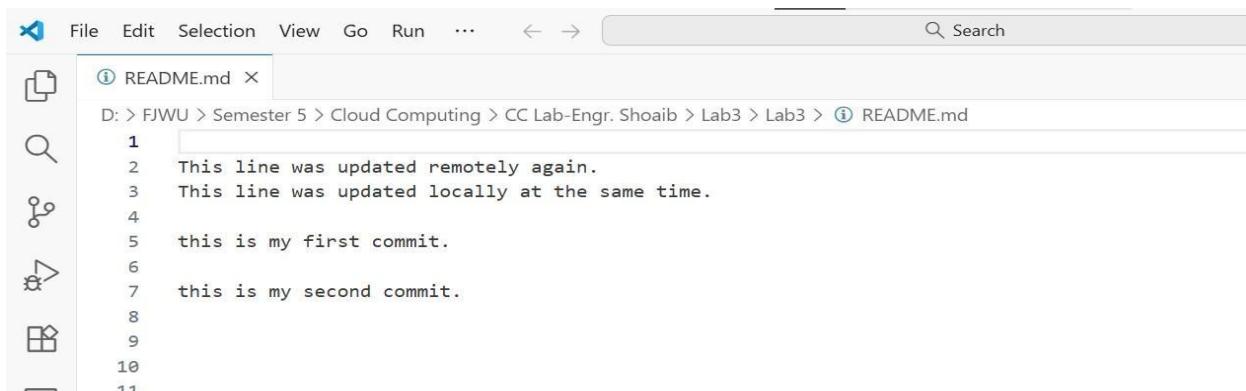
### 1. Add and Commit the First Change

```

# Edit any file (e.g., README.md), add a new line
# Edit the same file again and add another new line git
add . git commit -m "Second test commit"

```

### 5. Verify File Contents Before Reset



### 5. Perform a Soft Reset (Keeps Changes)

```

git reset --soft HEAD~1
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git reset --soft HEAD~1

```

*Screenshot: soft\_reset.png*

### Check Commit History After Soft Reset

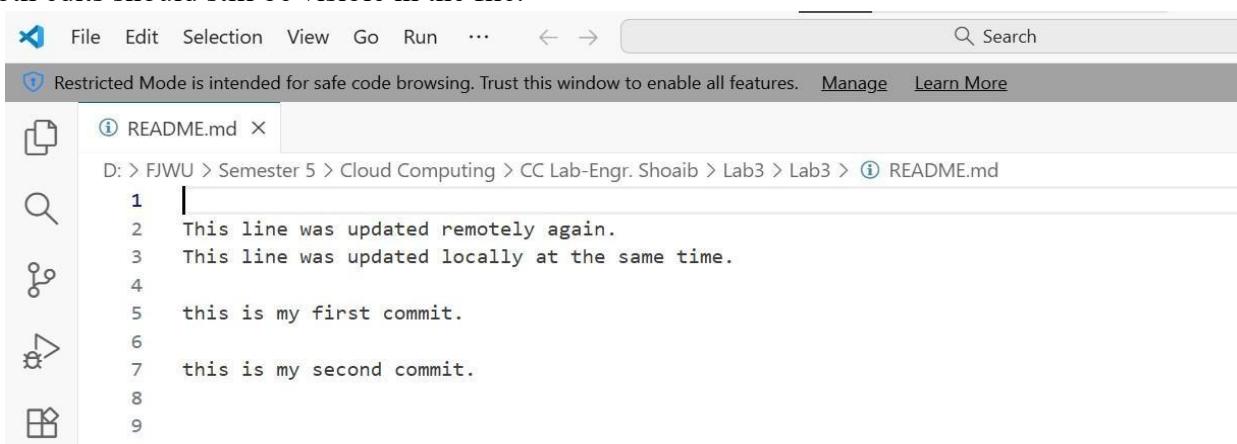
```
git log --oneline
```

Open the edited file (e.g., README.md) and confirm both newly added lines exist.

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git log --oneline
685e2a5 (HEAD -> main) Added test line
fdd2375 (origin/main, origin/HEAD) Removed tracked textfiles directory
b869d8a Added .gitignore to ignore textfiles directory
149c401 Added textfiles directory with three files
9d47d7a Local conflicting change
ffd845a Remote conflicting change
ec04ce7 Local update to README (B)
df97ab4 Remote update to README (B)
c7ddeec Remote update to README (B)
9339f5f Resolve merge conflict in README.md (merge remote changes)
77e6bd2 Local update to README (A)
22fa12a Remote update to README (A)
e2f594f Local update to README
c1da5cc Update README.md
d121e68 Update README.md
28e9511 Initial commit
```

Screenshot: log\_after\_soft\_reset.png Verify File Contents After Soft Reset

Both edits should still be visible in the file.



Screenshot: file\_after\_soft\_reset.png

## Check Git Status

```
git status
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/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
```

You

should see that changes are **staged and ready to commit again**.

Commit again (optional):

```
git commit -m "Second Test commit"
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git commit -m "Second Test commit"
[main 5e5eabd] Second Test commit
 1 file changed, 2 insertions(+)
```

Screenshot: file\_after\_soft\_reset.png

## 6. Perform a Hard Reset (Removes Changes)

```
git reset --hard HEAD~1
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git reset --hard HEAD~1
HEAD is now at 685e2a5 Added test line
```

Screenshot: hard\_reset.png

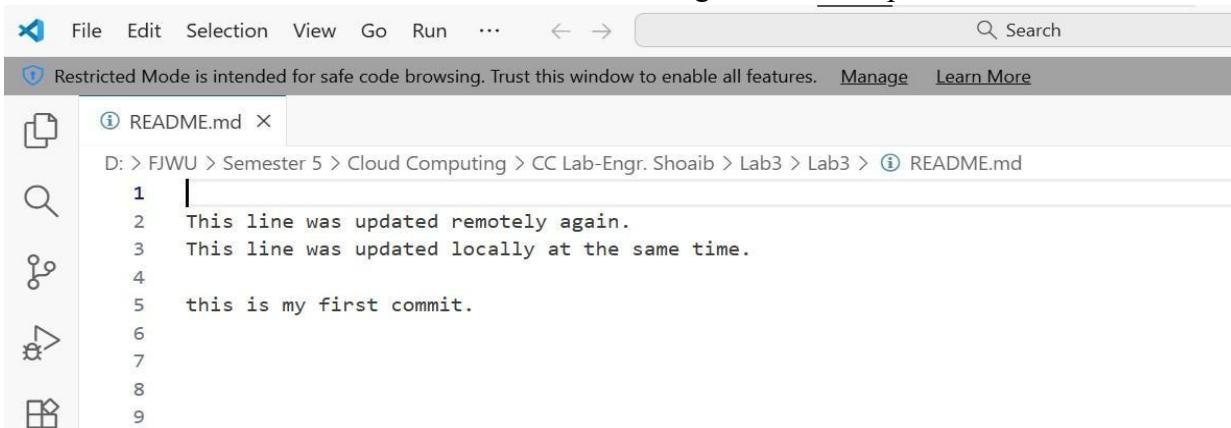
## Check Commit History After Hard Reset

```
git log -oneline
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git log --oneline
685e2a5 (HEAD -> main) Added test line
fdd2375 (origin/main, origin/HEAD) Removed tracked textfiles directory
b869d8a Added .gitignore to ignore textfiles directory
149c401 Added textfiles directory with three files
9d47d7a Local conflicting change
ffd845a Remote conflicting change
ec04ce7 Local update to README (B)
df97ab4 Remote update to README (B)
c7ddec Remote update to README (B)
9339f5f Resolve merge conflict in README.md (merge remote changes)
77e6bd2 Local update to README (A)
22fa12a Remote update to README (A)
e2f594f Local update to README
c1da5cc Update README.md
d121e68 Update README.md
28e9511 Initial commit
```

Screenshot: log\_after\_hard\_reset.png

## Verify File Contents After Hard Reset

The second added line should now be **removed**, reverting the file to its previous state.



Screenshot: file\_after\_hard\_reset.png

## Check Git Status git status

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/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
```

Output should show: *nothing to commit, working tree clean*

Screenshot: file\_after\_hard\_reset.png

## Task 7: Amending the Last Commit

### Step 1 – Make a small change

1. Open any file in your repository (for example, README.md).
2. Make a small change, such as adding a line or correcting a typo.
3. Save the file.

## Step 2 – Stage and commit the change `git add`

```
git commit -m "Fix log message"
```

- Take a screenshot of your terminal showing the commit.

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git add .

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git commit -m "Fix log message"
[main 599f6f4] Fix log message
 1 file changed, 3 insertions(+)
```

- Save it as `first_amend_commit.png`.

## Step 3 – Update another file

1. Realize you forgot to change another file. Open that file and make the necessary change.
2. Save the file.



## Step 4 – Stage and amend the last commit `git`

```
add . git commit --amend
```

- This updates your previous commit to include the new changes.
- Take a screenshot of your terminal after amending the commit.

```
MINGW64:/d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3
Fix log message

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:     Fri Oct 17 12:49:36 2025 +0500
#
# On branch main
# Your branch is ahead of 'origin/main' by 2 commits.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:   README.md
```

- Save it as `amend_commit.png. Summary`
- `git commit --amend` lets you **modify your last commit**, including its message or any staged files.
- It is useful for **quick fixes before pushing** to the remote repository.
- Once pushed, amending requires a **force-push** to update remote history.

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

### Step 1 – Make a change and commit it

1. In your repository folder, create or modify a file. For example, run in terminal:

```
echo "temporary text" >> temp.txt
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ echo "temporary text" >> temp.txt
```

2. Stage and commit the change: git add . git commit -m "Added temporary text file"

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git add .
git commit -m "Added temporary text file"
warning: in the working copy of 'temp.txt', LF will be replaced by CRLF the next time Git touches it
[main blaac4a] Added temporary text file
 1 file changed, 1 insertion(+)
 create mode 100644 temp.txt
```

3.

Push the commit to the remote branch:

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main|REBASE 1/3)
$ git push origin main
To https://github.com/hajrasarwar11/Lab3.git
  ! [rejected]      main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/hajrasarwar11/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.

File Edit Selection View Go Run ... ↶ ↷ Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
① README.md ×
D: > FJWU > Semester 5 > Cloud Computing > CC Lab-Engr. Shoaib > Lab3 > Lab3 > ① README.md
1 This line was added remotely from GitHub.
2
3 This line was updated remotely again.
4 This line was updated locally at the same time.
5
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
6 <<<<< HEAD (Current Change)
7 This is the first test line.
8
9 This is a test change for amending commit.
10
11 =====
12 this is my first commit.
13 >>>> 685ea5 (Added test line) (Incoming Change)
14
15
File Edit Selection View Go Run ... ↶ ↷ Search
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
① README.md ×
D: > FJWU > Semester 5 > Cloud Computing > CC Lab-Engr. Shoaib > Lab3 > Lab3 > ① README.md
1 This line was added remotely from GitHub.
2
3 This line was updated remotely again.
4 This line was updated locally at the same time.
5
6
7 This is the first test line.
8
9 This is a test change for amending commit.
10
11
12 this is my first commit.
13
14
git push origin
main

```

Take  
a

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 870 bytes | 72.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/hajrasarwar11/Lab3.git
  d01a12f..ebf51d0 main -> main

```

- Save it as **commit\_temp\_file.png**.

screenshot of your terminal showing the commit and push.

## Step 2 – Revert the commit safely

- Check the commit history:

```
git log -oneline
```

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git log --oneline
ebf51d0 (HEAD -> main, origin/main, origin/HEAD) Added temporary text file
7e730e9 Fix log message
afd8fa7 Added test line
d01a12f amending commit.
62cecf9 committing again
4bfe4e6 Remote update to README
ffd2375 Removed tracked textfiles directory
b869d8a Added .gitignore to ignore textfiles directory
149c401 Added textfiles directory with three files
9d47d7a Local conflicting change
ffd845a Remote conflicting change
ec04ce7 Local update to README (B)
df97ab4 Remote update to README (B)
c7ddeec Remote update to README (B)
9339f5f Resolve merge conflict in README.md (merge remote changes)
77e6bd2 Local update to README (A)
22fa12a Remote update to README (A)
e2f594f Local update to README
c1da5cc Update README.md
d121e68 Update README.md
28e9511 Initial commit

```

2. Identify the **commit hash** of the temporary text commit.
  3. Revert the commit: `git revert <commit-hash>`
- This creates a new commit that **undoes the changes** from the specific commit.

MINGW64:/d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3

Revert "Added temporary text file"

This reverts commit `ebf51d00caf1755f38123548020ae8344ecce10f`.

```

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   deleted:    temp.txt

```

```

[main 20e9220] Revert "Added temporary text file"
1 file changed, 1 deletion(-)
 delete mode 100644 temp.txt

```

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ |

```

Save it as `revert_commit.png`.

### Step 3 – Push the revert commit

git push origin main

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 265 bytes | 132.00 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 https://github.com/hajrasarwar11/Lab3.git
 eb51d0..20e9220 main -> main

```

- Take a **screenshot** after pushing.
- Save it as `revert_push.png`.

## Task 9: Force Push (With Caution)

**Warning:** Use this only on your own branch. Never force push on main or develop.

### Steps:

#### 1. Create a new branch:

```
git checkout -b test-force
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (main)
$ git checkout -b test-force
Switched to a new branch 'test-force'

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$
```

Save a screenshot: **new\_branch.png**

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ git branch
  feature-branch
* main
* test-force
```

#### 2. Make and commit a small change.

The screenshot shows a GitHub code editor interface with a file named README.md. The file contains the following content:

```
D: > FJWU > Semester 5 > Cloud Computing > CC Lab-Engr. Shoaib > Lab3 > Lab3 > README.md
1 This line was added remotely from GitHub.
2
3 This line was updated remotely again.
4 This line was updated locally at the same time.
5
6
7 This is the first test line.
8
9 This is a test change for amending commit.
10
11 this is my first commit.
12
13
14 This is a test change for amending commit.
15
16 This is a test change for force push.
17
18
```

Below the code editor, the terminal shows the command to add the file and the commit message:

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ git add .
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ git commit -m "Added test change for force push"
[test-force dbf2e32] Added test change for force push
 1 file changed, 3 insertions(+)
```

Save a screenshot: **force\_commit.png** 3.

#### Push the branch normally:

```
git push origin test-force
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ git push origin test-force
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 345 bytes | 86.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'test-force' on GitHub by visiting:
remote:     https://github.com/hajrasarwar11/Lab3/pull/new/test-force
remote:
To https://github.com/hajrasarwar11/Lab3.git
 * [new branch]      test-force -> test-force
```

Save a screenshot: **push\_force\_branch.png** 4. Perform a hard reset: `git reset --hard HEAD~1`

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ git reset --hard HEAD~1
HEAD is now at 20e9220 Revert "Added temporary text file"
```

Save a screenshot: **hard\_reset\_force.png** 5. Try pushing again normally:

```
git push origin test-force
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ git push origin test-force
To https://github.com/hajrasarwar11/Lab3.git
 ! [rejected]      test-force -> test-force (non-fast-forward)
error: failed to push some refs to 'https://github.com/hajrasarwar11/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.
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ |
```

Expected: Rejected by remote Save a

screenshot: **normal\_push.png** 6.

Force push to remote:

```
git push origin test-force -force
```

```
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/Lab3 (test-force)
$ git push origin test-force --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hajrasarwar11/Lab3.git
 + dbf2e32...20e9220 test-force -> test-force (forced update)
```

Save a screenshot: **force\_push.png**

## Task 10: Running Gitea in GitHub Codespaces via Docker Compose

Steps:

### 1. Fork the Gitea Repository

1. Go to the Gitea repository.
2. Click **Fork** at the top right and fork it to your GitHub account.

The image shows two screenshots of the GitHub interface. The top screenshot is titled 'Create a new fork' and shows the process of forking the 'go-gitea/gitea' repository. It includes fields for 'Owner' (set to 'hajrasarwar11') and 'Repository name' (set to 'gitea'), a description field containing 'Git with a cup of tea! Painless self-hosted all-in-one software development service, including Git hosting, code', and a checked checkbox for 'Copy the main branch only'. The bottom screenshot shows the forked repository 'gitea' (forked from 'go-gitea/gitea'). It displays basic repository stats (1 Branch, 0 Tags), a commit history with recent updates from 'GiteaBot', and a sidebar with repository details like 'About' (description: 'Git with a cup of tea! Painless self-hosted all-in-one software development service, including Git hosting, code review, team collaboration, package registry and CI/CD'), 'gitea.com', and links to 'Readme', 'MIT license', 'Code of conduct', 'Contributing', 'Security policy', and 'Activity'.

### 3. Take a screenshot of your forked repository page.

Save as **forked\_gitea.png**

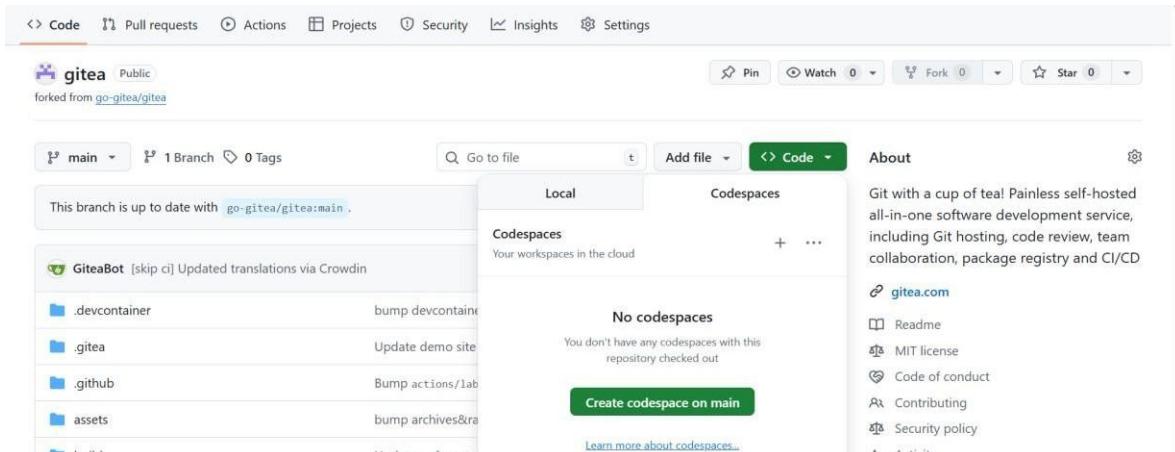
## 2. Open the Forked Repo in GitHub Codespaces

1. On your forked repo, click:

**Code > Codespaces > Create codespace on main**

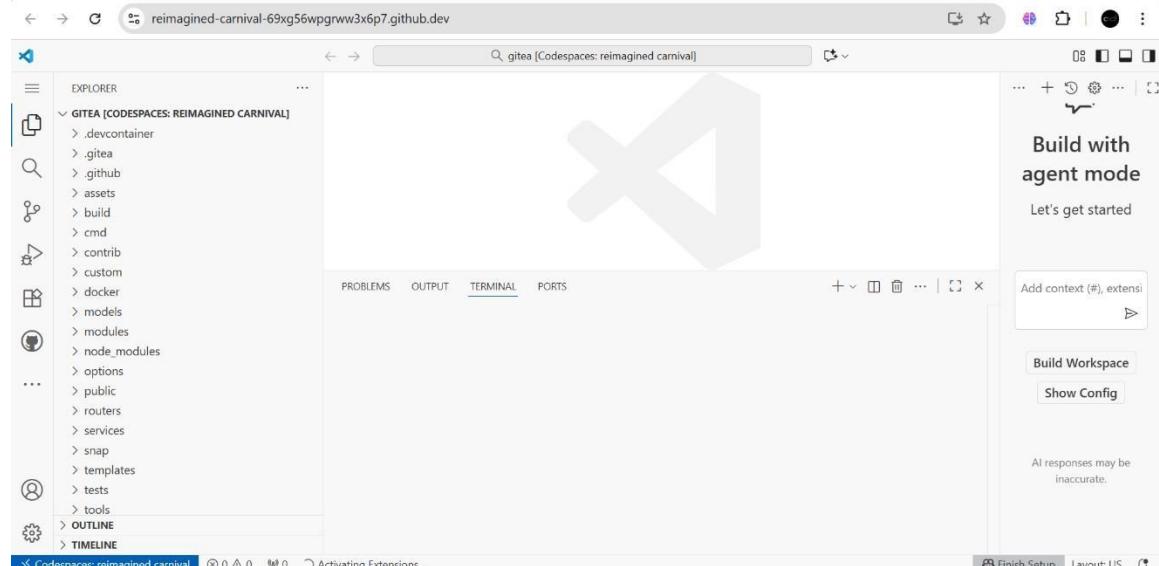
### 3. Start Gitea with Docker Compose

1. In the Codespace terminal, run: `docker compose up -d`



2. Wait for the Codespace to load.

3. Take a screenshot of the Codespace loading.



Save as **codespace\_loading.png**

2. Wait for the containers to start.

3. Take a screenshot of the terminal showing the running containers. Save as **docker\_up.png**

### 4. Access Gitea Web Interface

1. In Codespaces, forward port 3000 (using the port forwarding UI).

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS 5		
Port	Forwarded Addr...	Running Process		Visibility	Origin	
3000	<a href="https://supreme-s...">https://supreme-s...</a>	Process information unavailable		Private	User Forwarded	

2. Change the port visibility from Private → Public.

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS 5		
Port	Forwarded Addr...	Running Process		Visibility	Origin	
3000	<a href="https://supreme-s...">https://supreme-s...</a>	Process information unavailable		Public	User Forwarded	

3. Click the forwarded port link to open Gitea in a browser tab.

The screenshot shows the 'Initial Configuration' page of Gitea. It includes fields for Database Type (MySQL), Host (127.0.0.1:3306), Username (gitea), Password, and Database Name (gitea). Under General Settings, the Site Title is set to 'Gitea: Git with a cup of tea'. Another screenshot below shows the 'Optional Settings' section with fields for Gitea Base URL (https://supreme-space-orbit-wvvj57jxx5g35qqv-3000.app.github.dev) and Log Path (/home/vscode/gitea/log). There is also a checkbox for 'Enable Update Checker'.

Save as **gitea\_install\_page.png**

4. Take a screenshot of the Gitea install page.

## 5. Install Gitea

1. Fill out the installation form:

- Admin username: admin
- Password: yourpassword

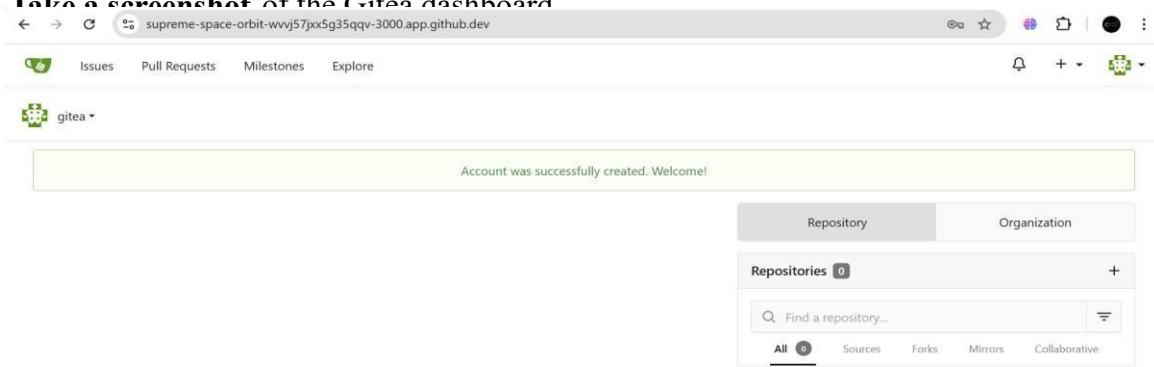
The screenshot shows the 'Register Account' form. It has fields for Password and Confirm Password, both containing masked text. A 'Register Account' button is at the bottom, and a link for existing users to 'Sign in now!' is at the bottom right.

2. Take a screenshot showing the completed admin setup.

Save as **admin\_setup.png**

## 6. Log In to Gitea

1. Use your admin credentials to log in.
2. Take a screenshot of the Gitea dashboard



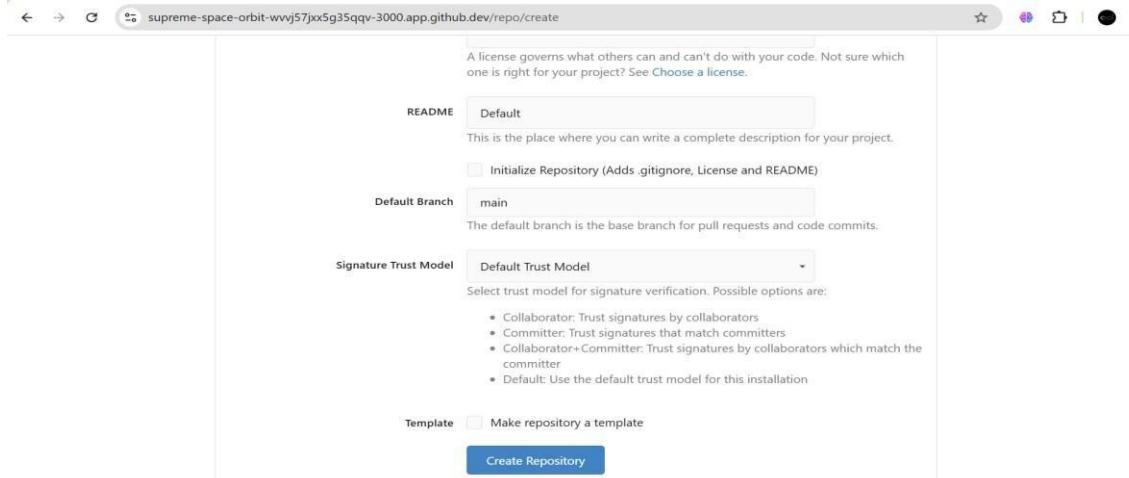
Save as gitea\_dashboard.png

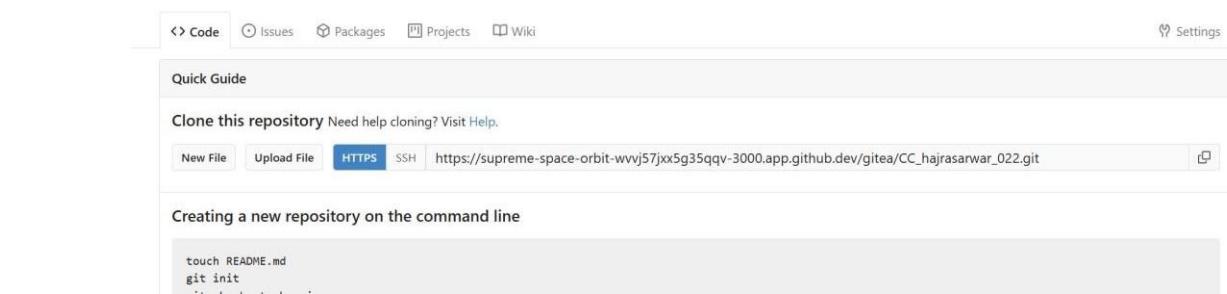
## 7. Create a New Repository in Gitea

1. Click New Repository or + > New Repository in Gitea.



2. Fill out repository details and create it.





## Task 11: Creating a GitHub Pages Portfolio Site Steps

### 1. Create a GitHub Pages Repository

1. Go to github and create a new repository named: <your-username>.github.io  
*Example:* If your username is inshal nasir, the repository should be:  
HajraSarwar.github.io
2. Make sure the repository is **public**.



3. Save a screenshot of the newly created repository as:



### 2. Add Static Website Code

1. Prepare a simple static website (like your CV or portfolio) using **HTML/CSS/JS**.

- You can use sample code such as the Iqra Bashir CV repository.
2. Put your files in a **local folder**.
  3. Save a screenshot of your local files as:

```

index.html # style.css
D: > FJWU > Semester 5 / Cloud Computing > CC Lab-Engr. Shoaib > Lab3 > CV > index.html > html > body > div.container > section > ul > li
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6    <title>Hajra Sarwar - Software Engineer</title>
7    <link rel="stylesheet" href="style.css" />
8  </head>
9  <body>
10   <div class="container">
11     ...

```

local\_static\_site.png

### 3. Push the Files to GitHub

1. Open your terminal and navigate to your local folder.

```

MINGW64:/d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/CV
HP@DESKTOP-JFEH5TK MINGW64 ~ (master)
$ cd "D:\FJWU\Semester 5\Cloud Computing\CC Lab-Engr. Shoaib\Lab3\CV"
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib\Lab3\CV (main)
$ 

```

2. Initialize git (if not already): `git init`

```

git remote add origin git@github.com:<yourusername>/<yourusername>.github.io.git
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib\Lab3\CV (main)
$ git init
Initialized empty Git repository in D:/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/CV/.git/
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/CV (master)
$ git remote add origin https://supreme-space-orbit-wvvj57jxx5g35qqv-3000.app.github.dev/gitea/HajraSarwar.github.io.git

```

3. Add and commit your files:

```

git add .
git commit -m "Add portfolio site for GitHub Pages"
HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/CV (master)
$ git add .
git commit -m "Add portfolio site for GitHub Pages"
[master (root-commit) 42055f3] Add portfolio site for GitHub Pages
 2 files changed, 214 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css

```

git push -u origin main

```

HP@DESKTOP-JFEH5TK MINGW64 /d/FJWU/Semester 5/Cloud Computing/CC Lab-Engr. Shoaib/Lab3/CV (main)
$ git push -u origin main --force
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.77 KiB | 142.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hajrasarwar11/Hajrasarwar.github.io.git
 + 3a75a43...2066045 main -> main (forced update)
branch 'main' set up to track 'origin/main'.

```

push\_static\_site.png

4. Save a screenshot of your terminal showing a successful push:

### 4. Check GitHub Pages Settings

1. Go to your repository on GitHub.

github\_pages\_settings.png

2. Click **Settings > Pages**.
3. Confirm that your site is published and note the link to your live site:  
<https://<yourusername>.github.io>
4. Save a screenshot of the Pages settings:

## 5. Visit Your Live Site

1. Open your GitHub Pages site in the browser.

