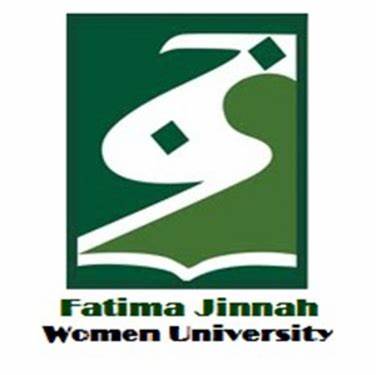
**CLOUD COMPUTING LAB**



**SUBMITTED TO**  
SIR WAQAS SALEEM  
  
**SUBMITTED BY**  
HAMNA MAHMOOD  
2023-BSE-025

BSE V-A

**LAB 03**

**Working with Git History, Stashing, and Reverting Commits**

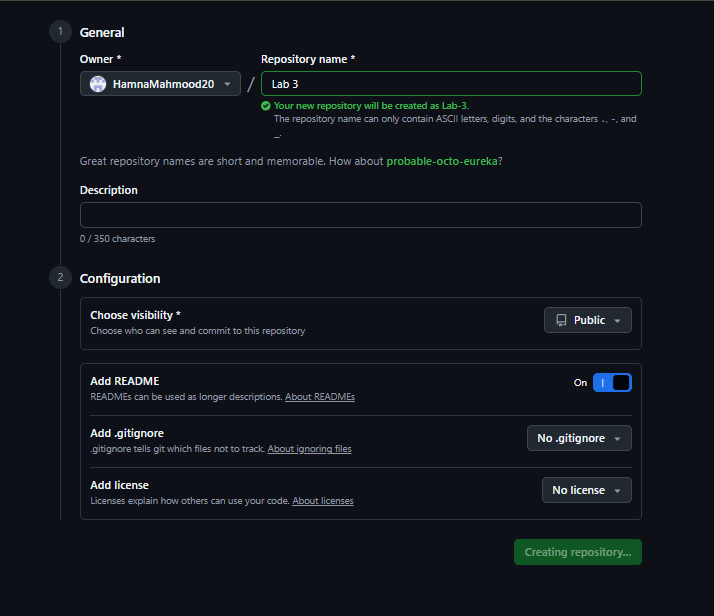
**Repository Link:** https://github.com/HamnaMahmood20/CC-Hamna-Mahmood-25-BSE-VA

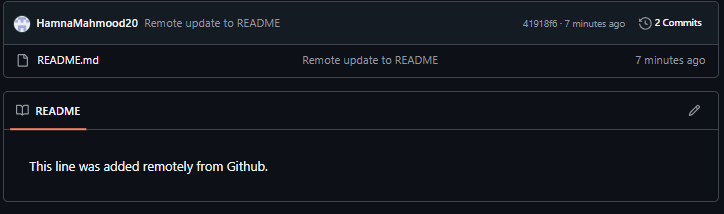
**Task 1**

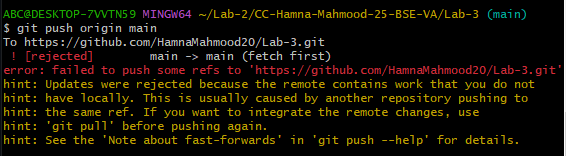
**Handling Local and Remote Commit Conflicts (Pull vs Pull --rebase)**

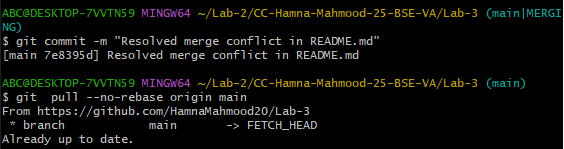
**Save a screenshot of your terminal as local\_commit.png.**

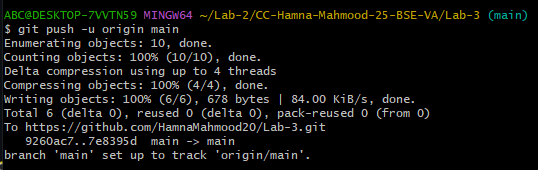
First making a new repository of Lab 3,

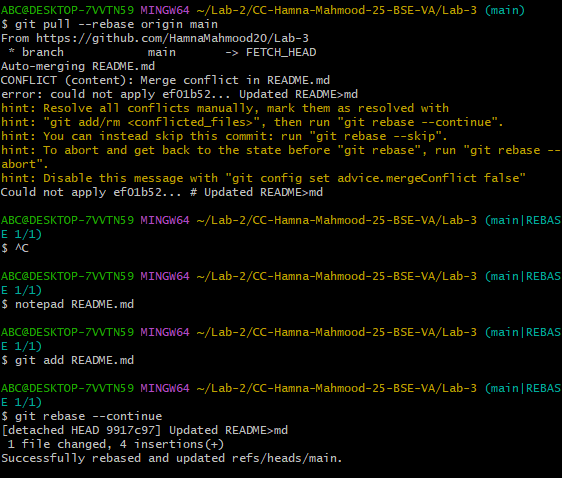
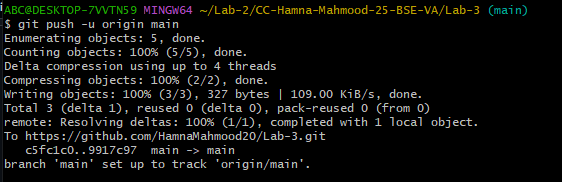
****





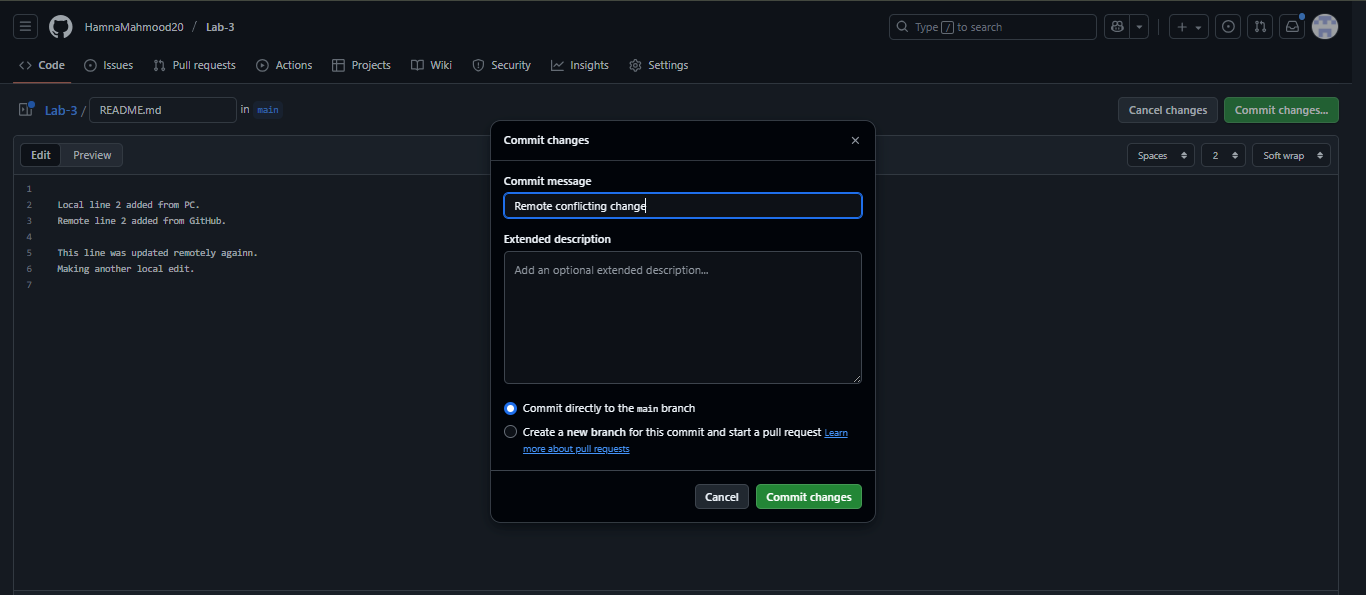


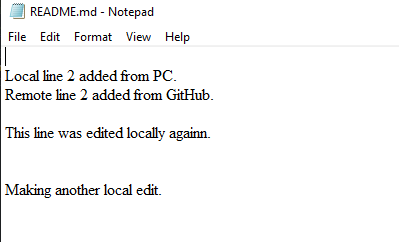


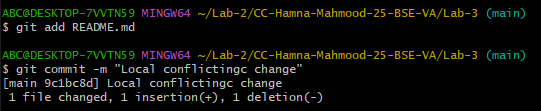
****

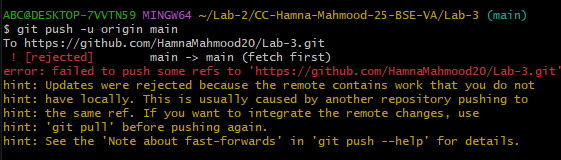
**Task 2**

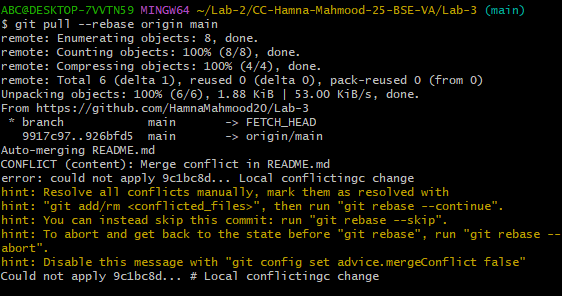
**Creating and Resolving Merge Conflicts Manually**

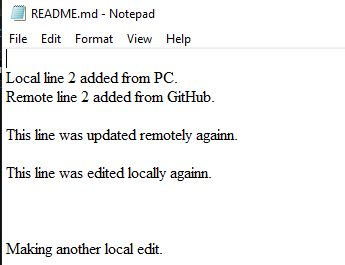


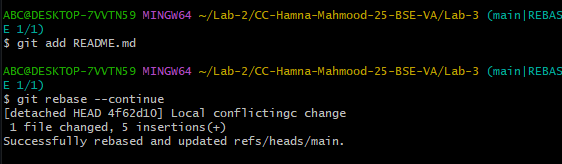
****

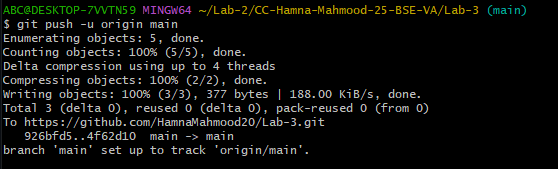
****

****

****

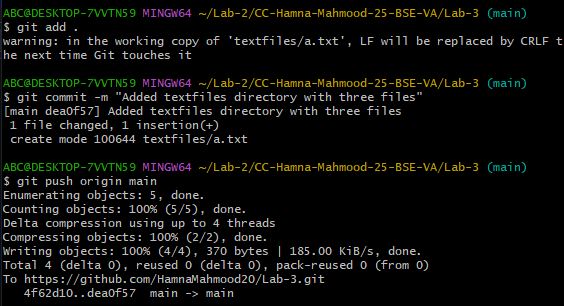
****

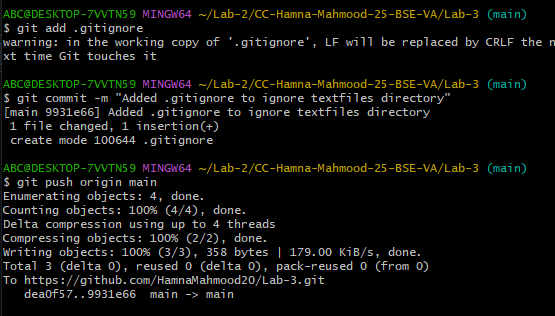
****

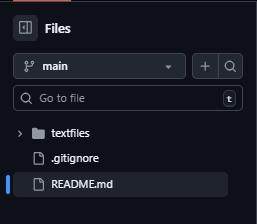
****

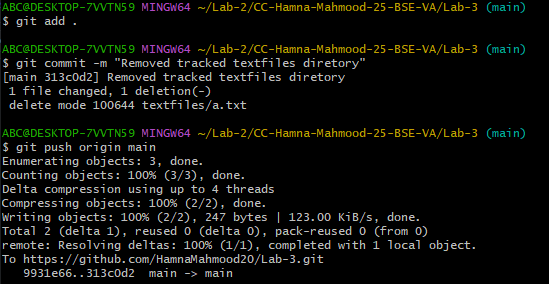
**Task 3**

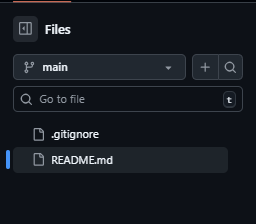
**Managing Ignored Files with .gitignore and Removing Tracked Files**

****

****

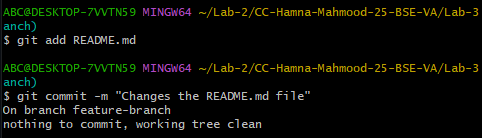
****

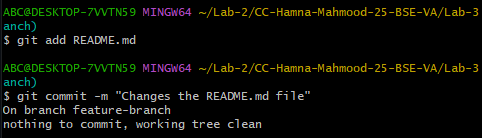
****

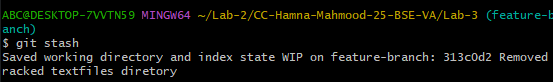
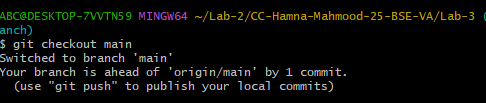
****

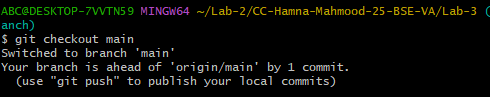
**Task 4**

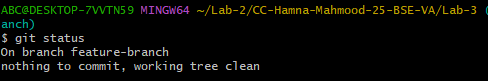
**Create Temporary Changes and Use git stash**

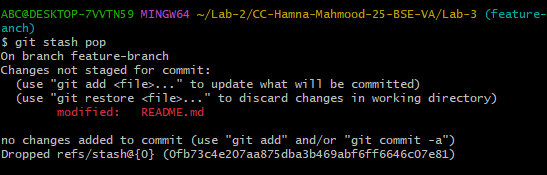
****

****

****

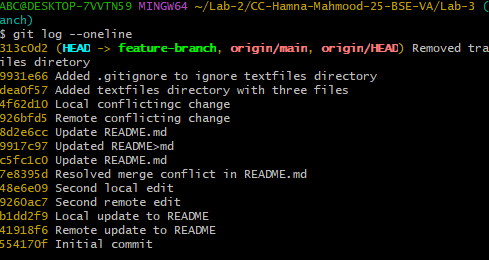
****

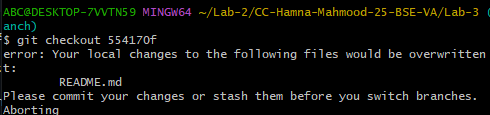
****

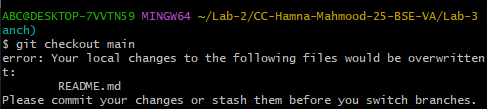
****

**Task 5**

**Checkout a Specific Commit Using git log**

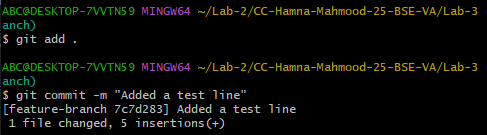
****

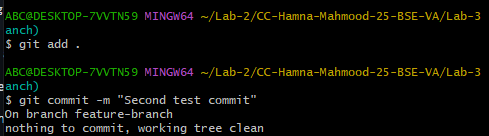
****

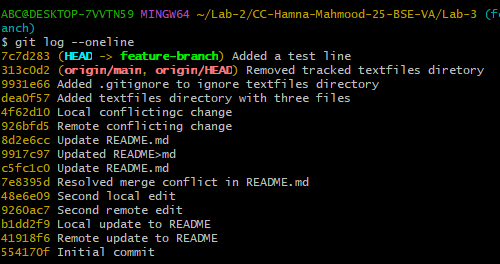
****

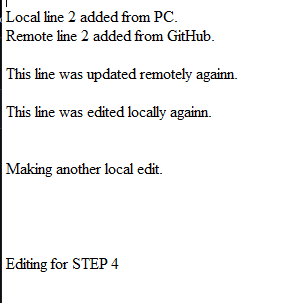
**Task 6**

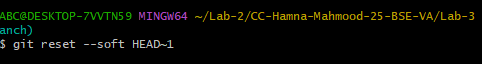
**Resetting Commits (Soft vs Hard Reset) (With Verification Steps)**

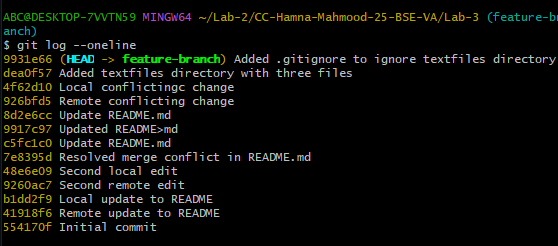
****

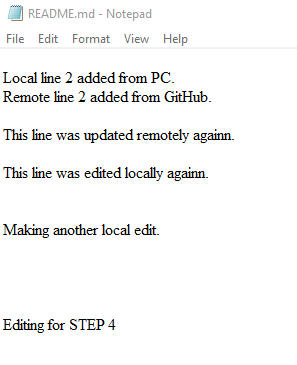
****

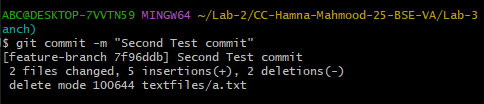


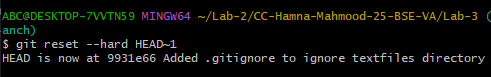
****

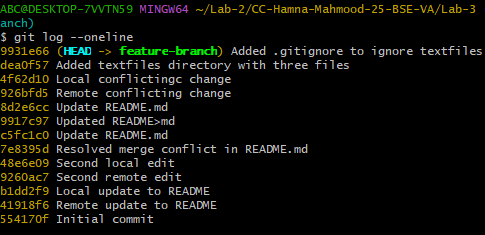
****

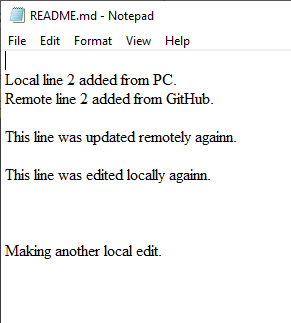
****

****

****

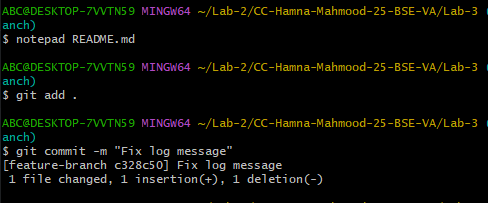
****

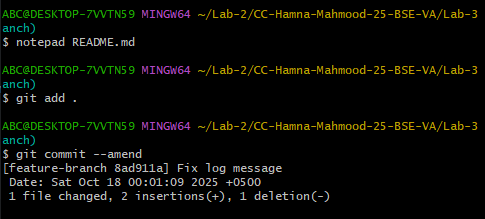
****

****

**Task 7**

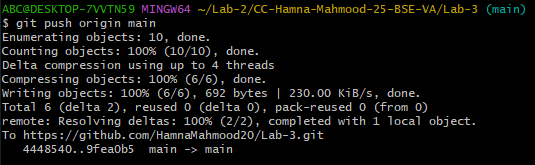
**Amending the Last Commit**

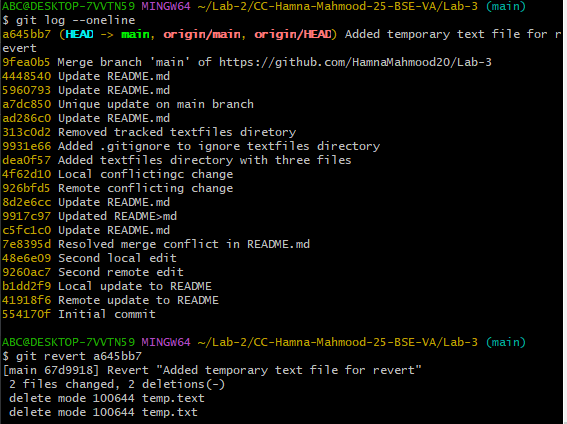
****

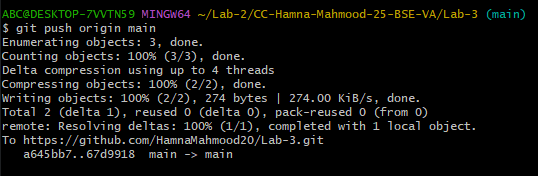
****

**Task 8**

**Reverting a Commit (Safe Undo on Remote Branch)**

****

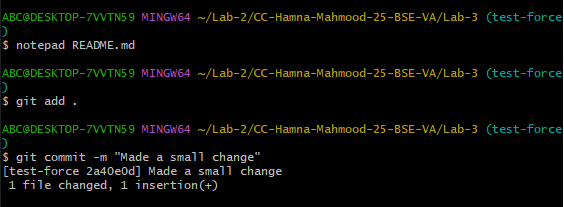
****

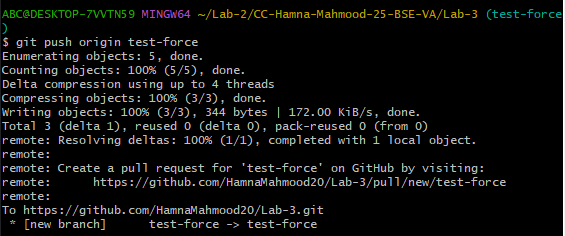
****

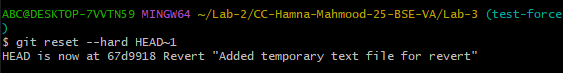
**Task 9**

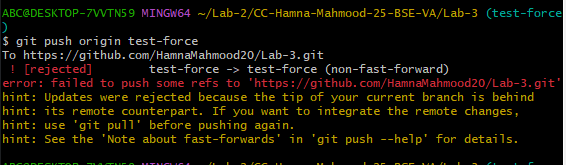
**Force Push (With Caution)**

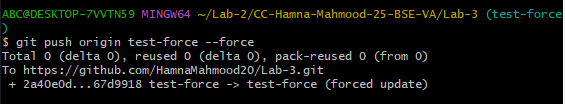






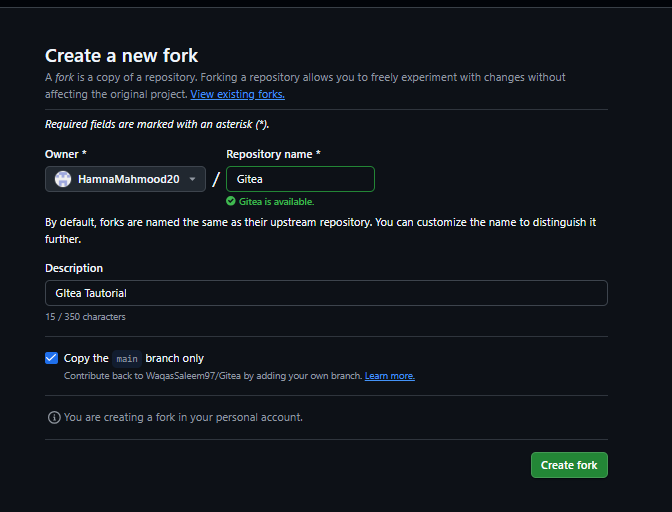


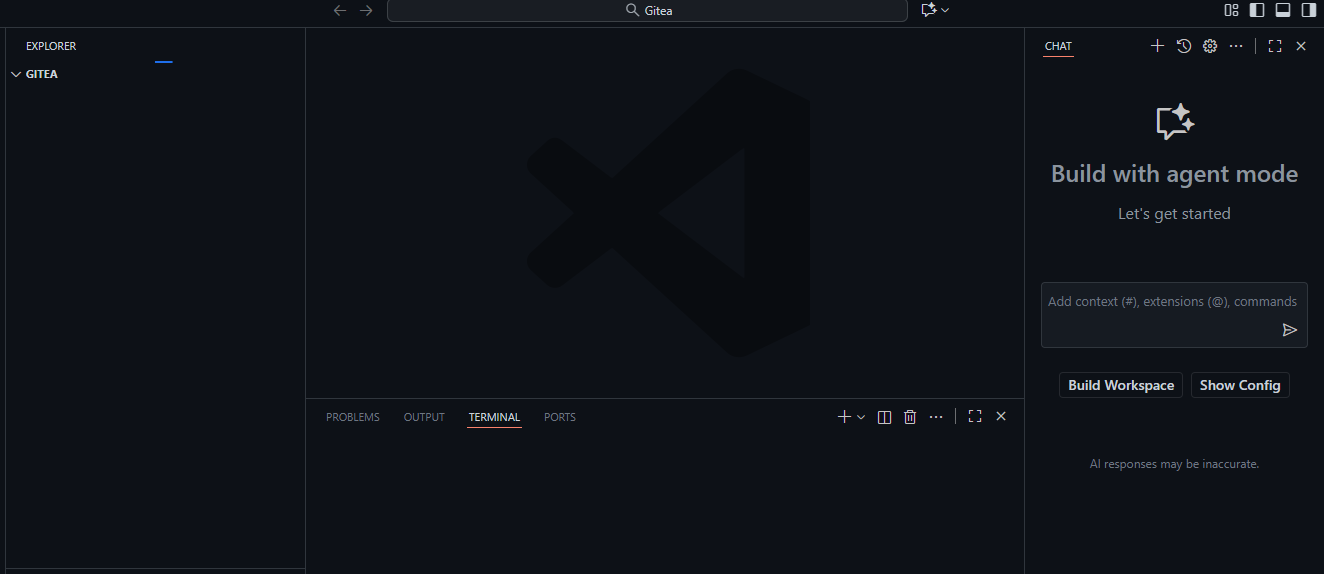




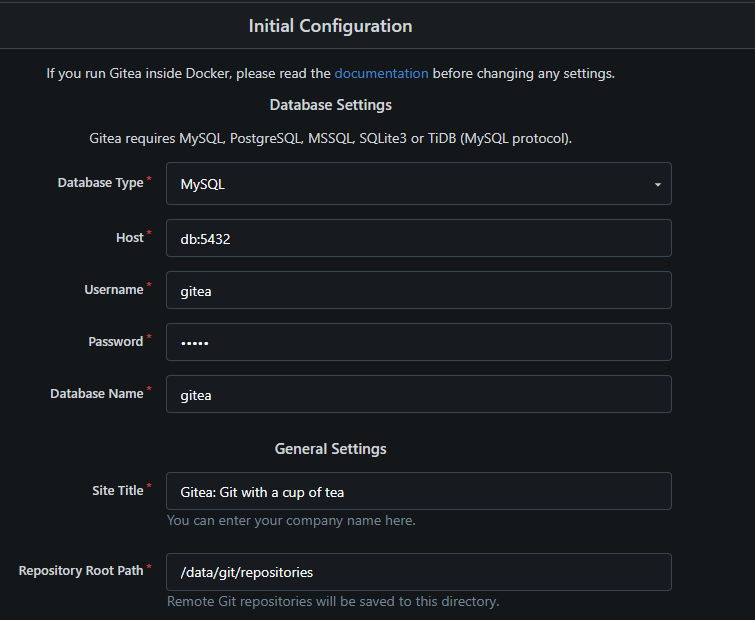
**Task 10**

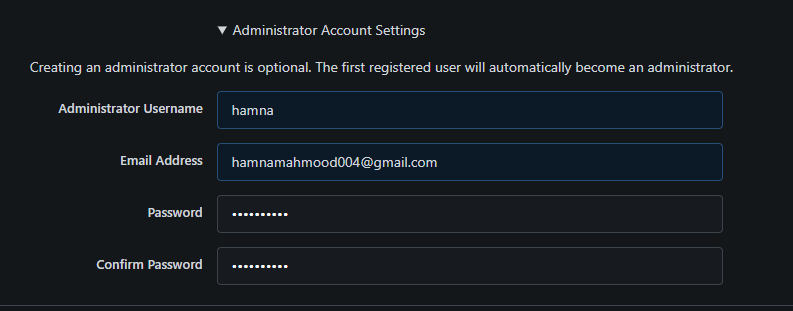
**Running Gitea in GitHub Codespaces via Docker Compose**

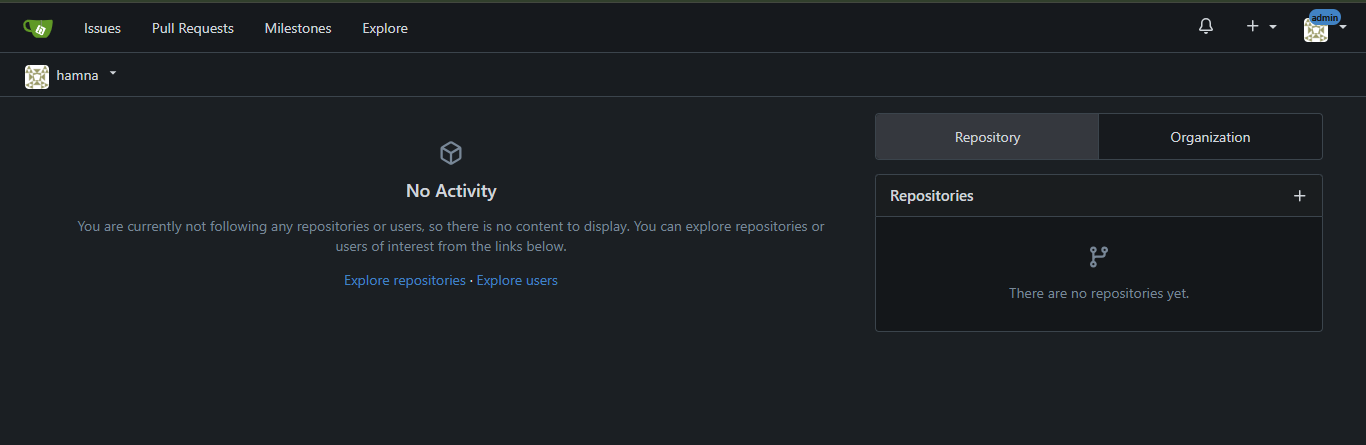
****

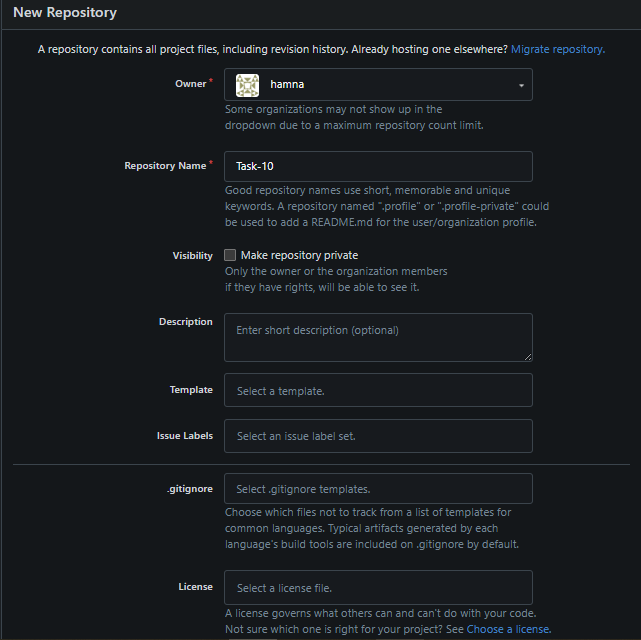
****

****

****

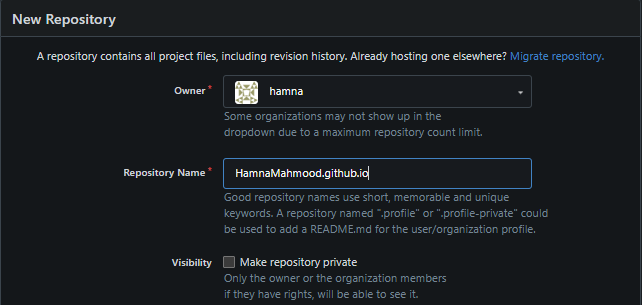
****

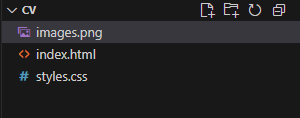
****

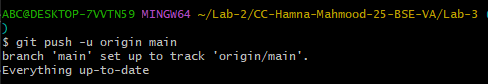
****

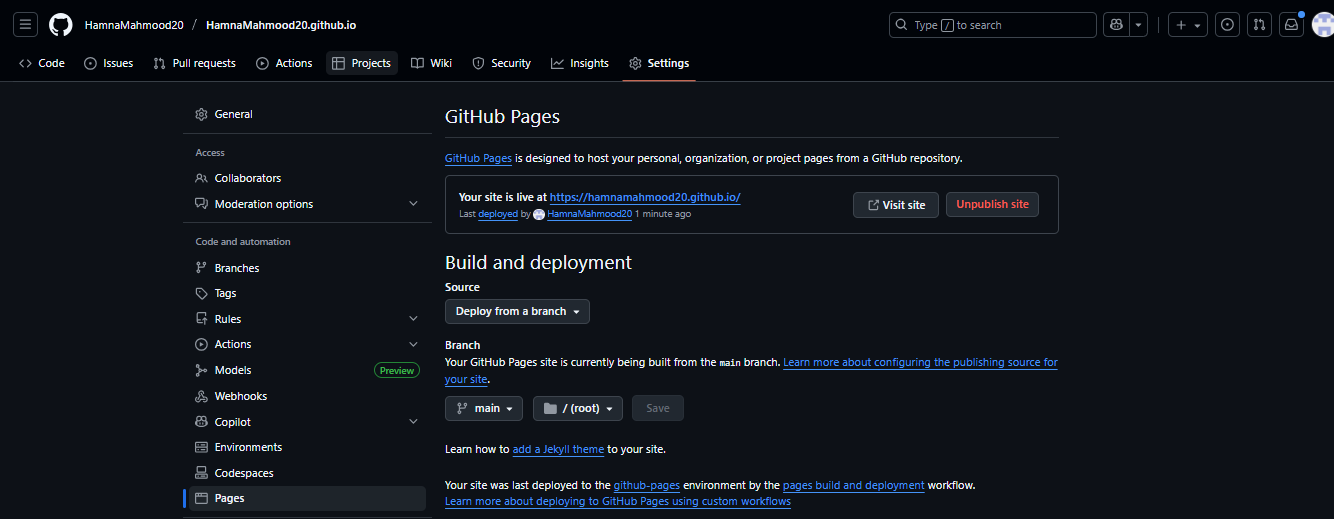
**Task 11**

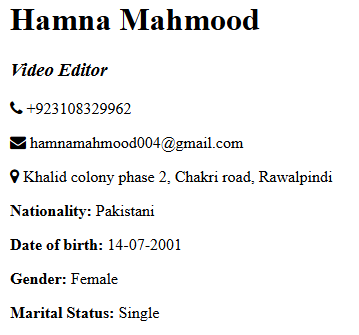
**Creating a GitHub Pages Portfolio Site**

****

****

****

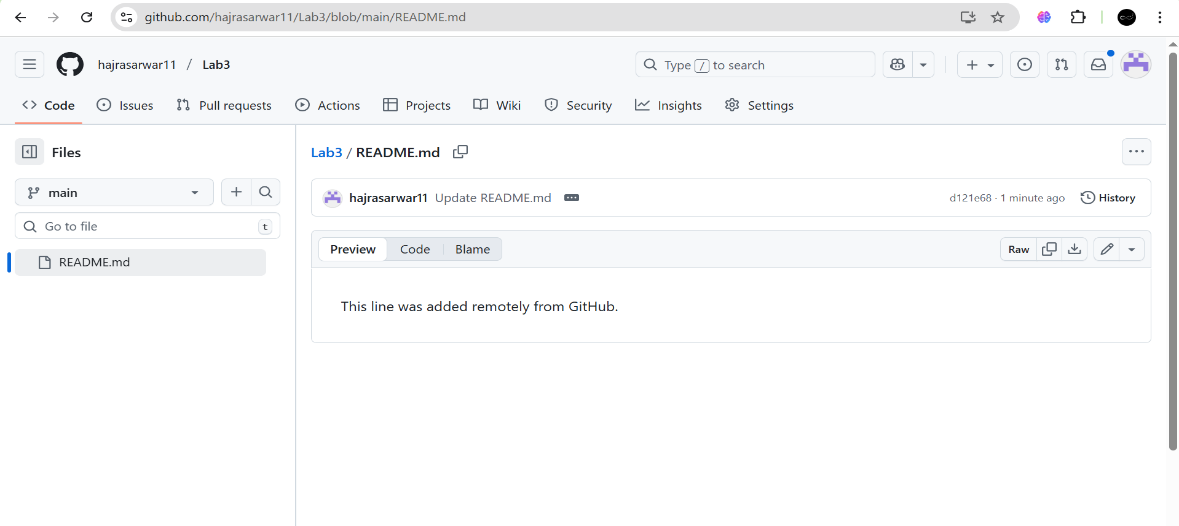
****

****

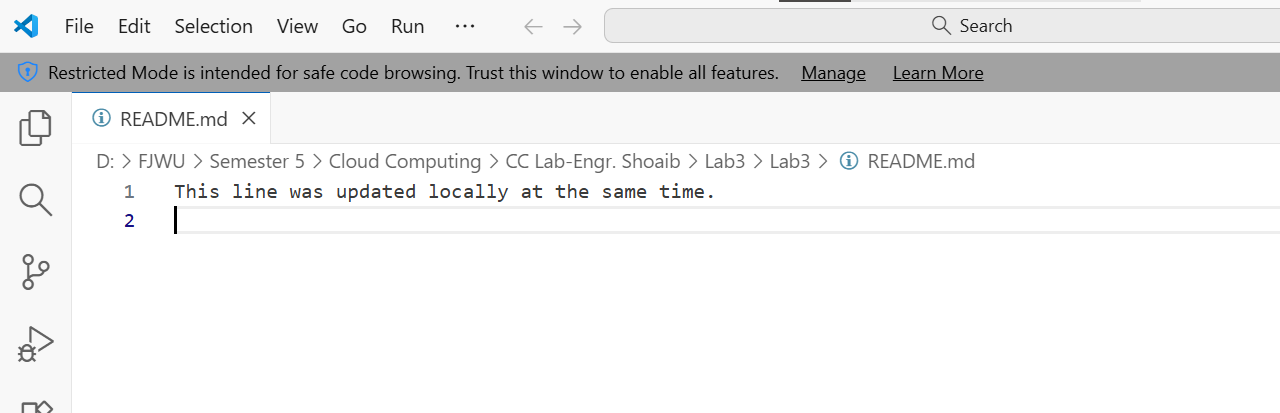
**Exam Evaluation Questions**

**1. Local vs Remote Conflict Resolution**

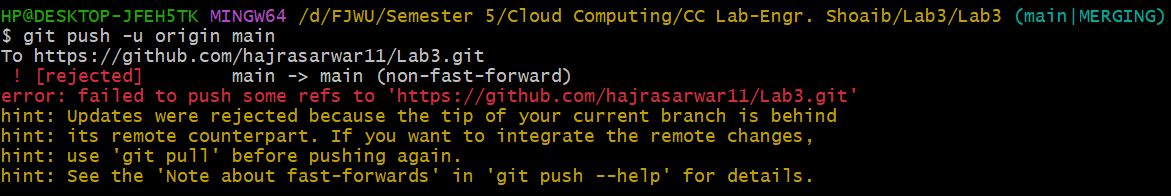
**Step 1: On GitHub, edit a file (e.g., README.md) and commit the change.  
Screenshot:**

****

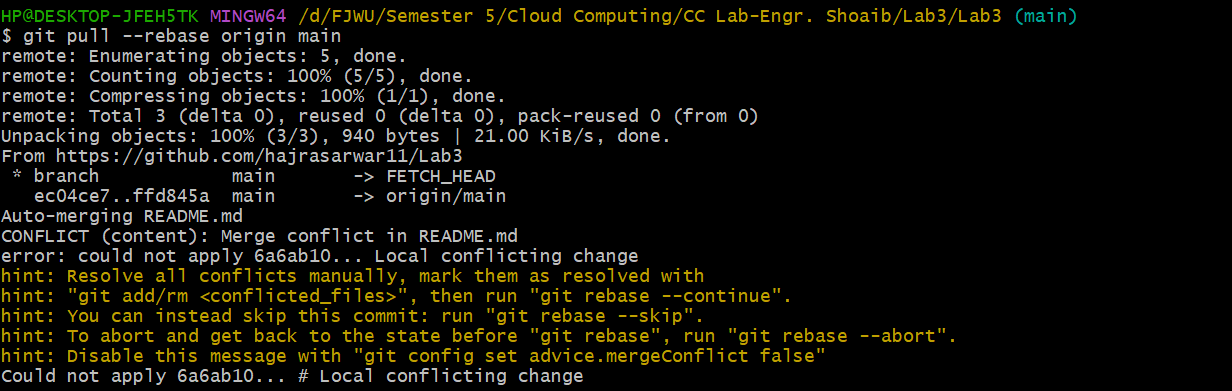
**Step 2: On your local machine, edit the same file differently (avoid conflict) and commit.  
Screenshot:**

****

**Step 3: Try to push your local commit and observe the error.  
Screenshot:**

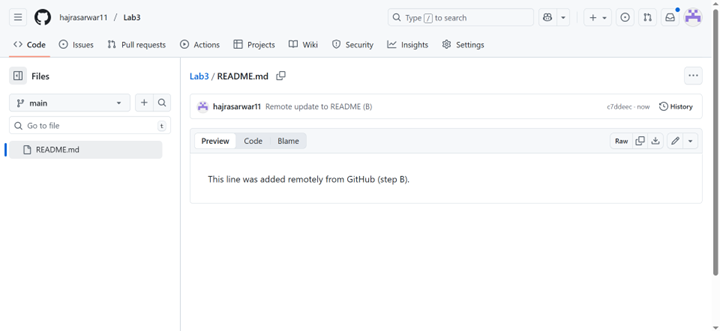
****

**Step 5: Repeat with another remote/local change, but resolve using git pull --rebase, then push.  
Screenshot:**

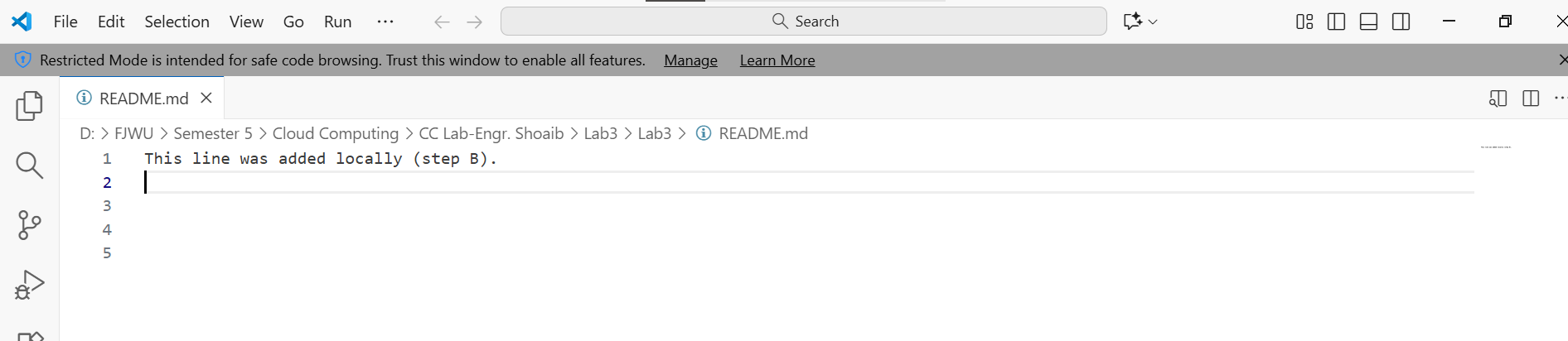
****

**2. Manual Merge Conflict Handling**

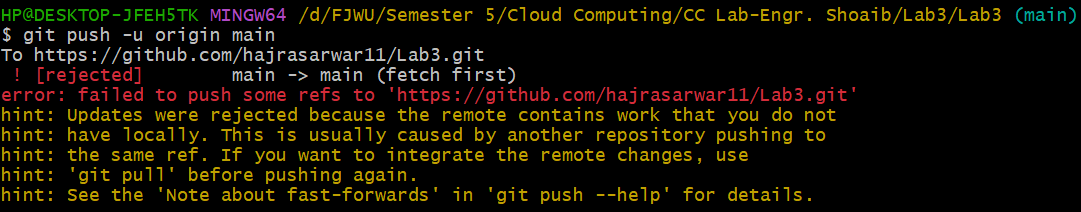
**Step 1: On GitHub, change a specific line in a file and commit.  
Screenshot:**

****

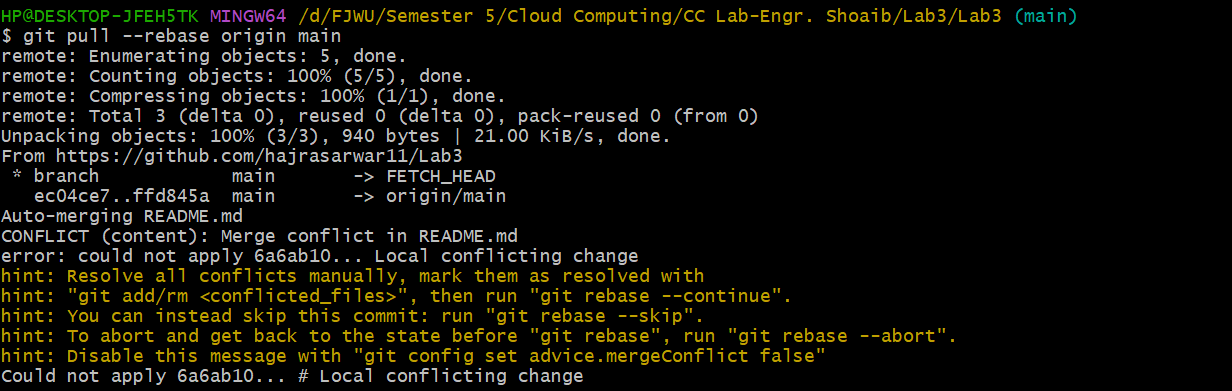
**Step 2: Locally, change the same line differently and commit.  
Screenshot:**

****

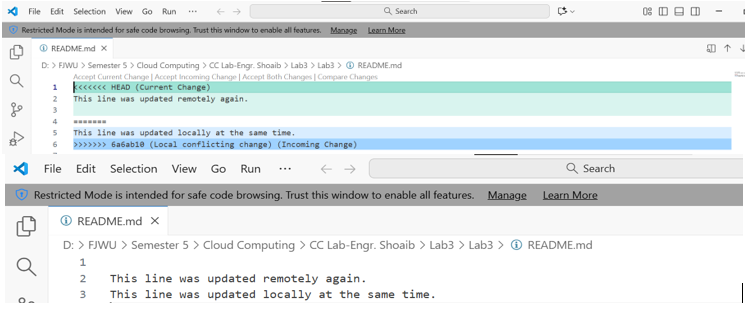
**Step 3: Try to push your local change and observe the conflict error.  
Screenshot:**

****

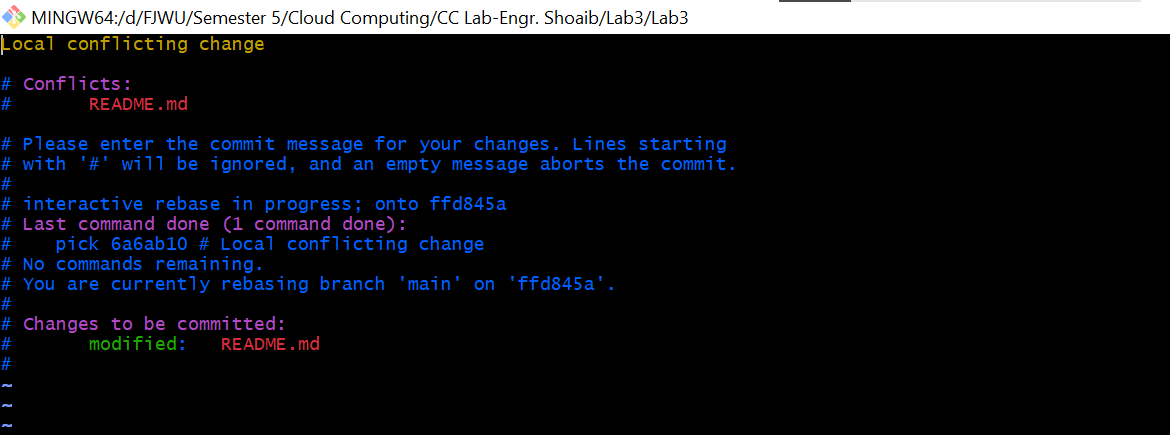
**Step 4: Use git pull --rebase to fetch changes and trigger the conflict.  
Screenshot:**

****

**Step 5: Edit the conflicted file to resolve the conflict manually.  
Screenshot:**

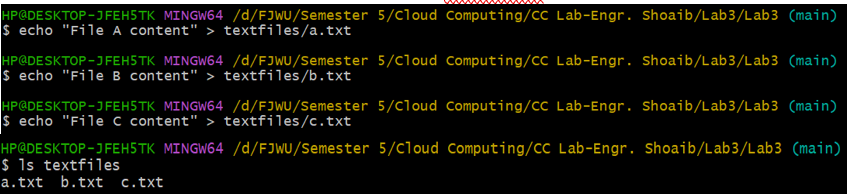
****

**Step 6: Mark the conflict resolved (git add <file>, git rebase --continue) and push.  
Screenshot:**

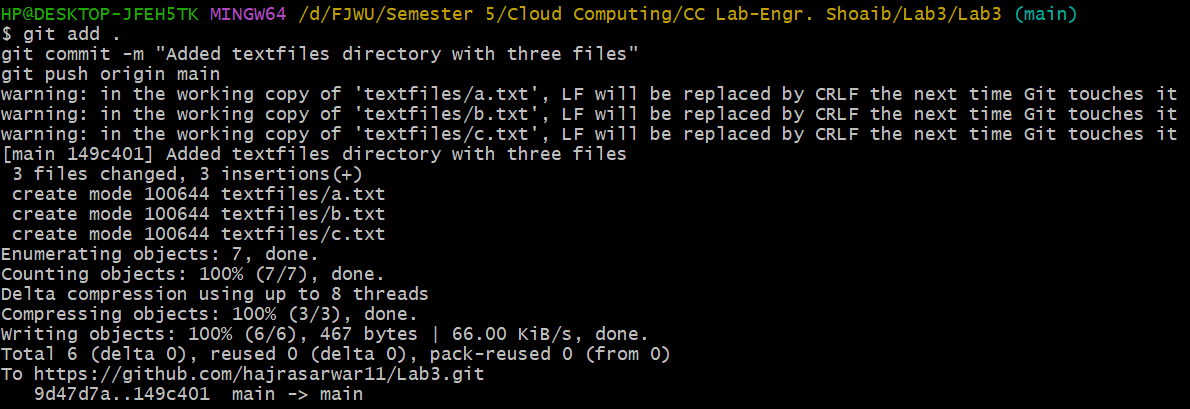
****

**3. Managing Ignored and Tracked Files**

**Step 1: Create a new folder (e.g., DocFiles) and add several files inside.  
Screenshot:**

****

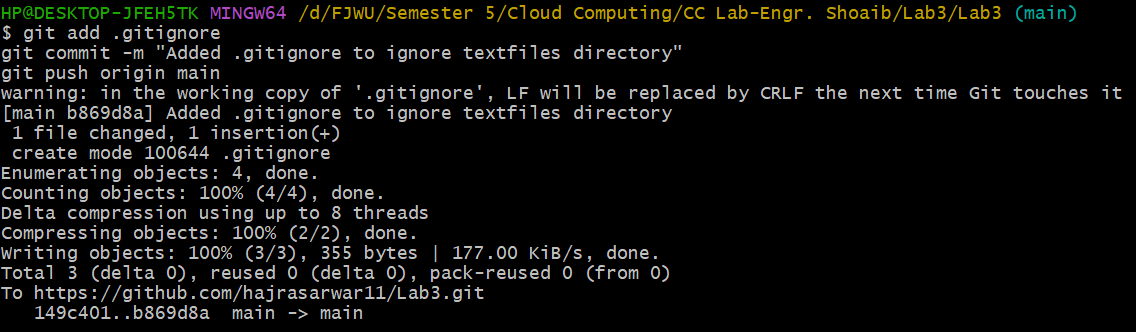
**Step 2: Commit and push the folder/files to GitHub.  
Screenshot:**

****

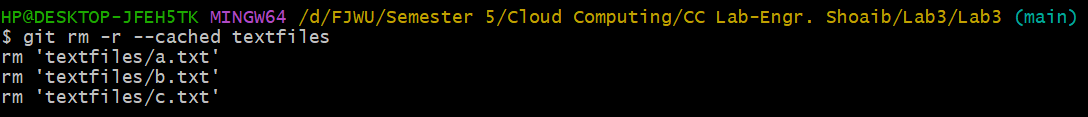
**Step 3: Add the folder to your .gitignore file.  
Screenshot:**

****

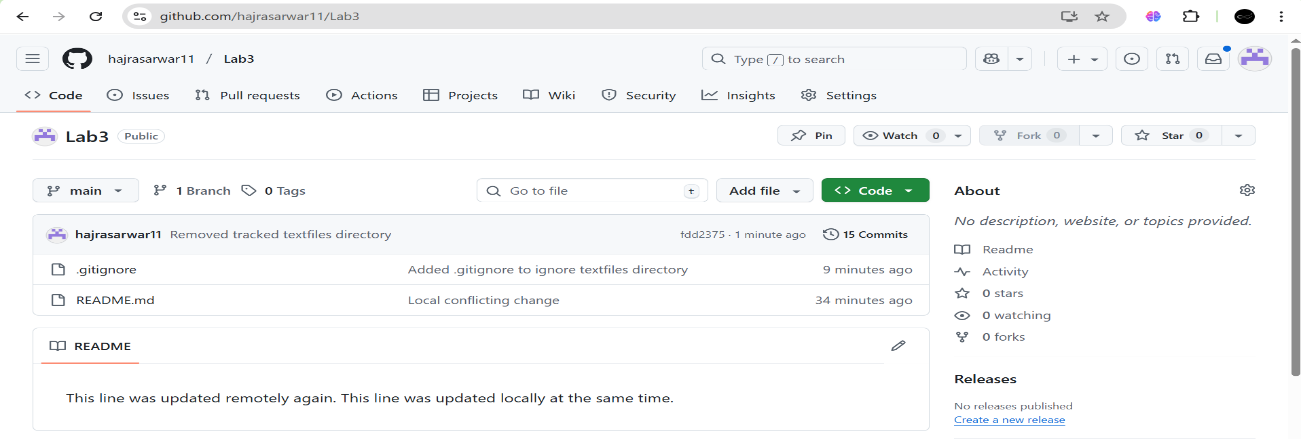
**Step 4: Commit and push the .gitignore update.  
Screenshot:**

****

**Step 5: Remove the folder from tracking using git rm -r --cached <folder>.  
Screenshot:**

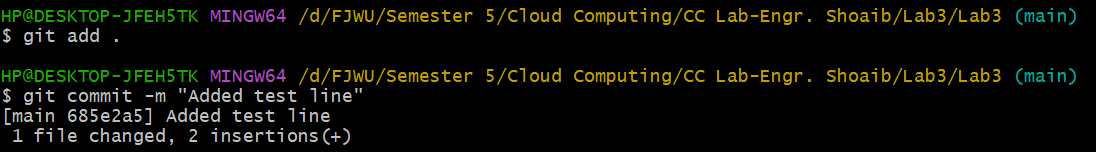
****

**Step 6: Commit and push the change, then verify the folder is no longer tracked on GitHub.  
Screenshot:**

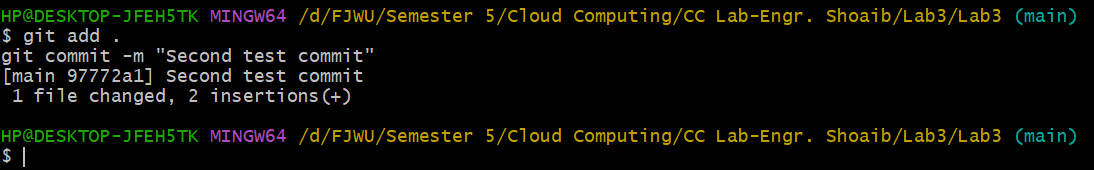
****

**4. Commit History Manipulation and Recovery**

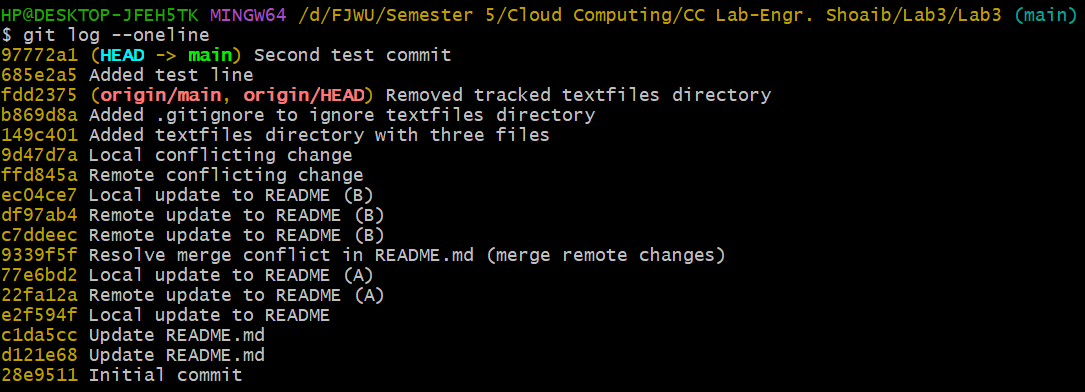
**Step 1: Make a change and commit it.  
Screenshot:**

****

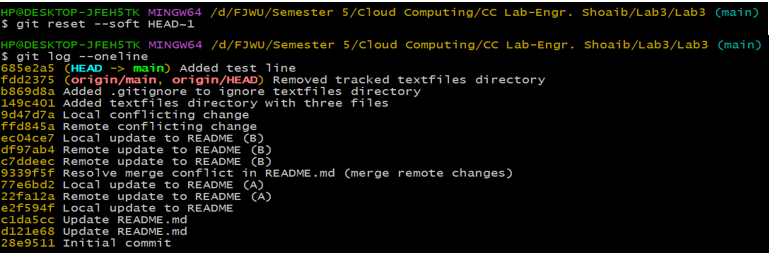
**Step 2: Make another change and commit again.  
Screenshot:**

****

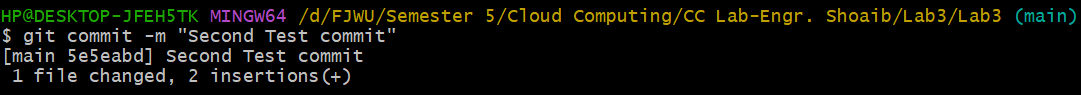
**Step 3: View your commit history.  
Screenshot:**

****

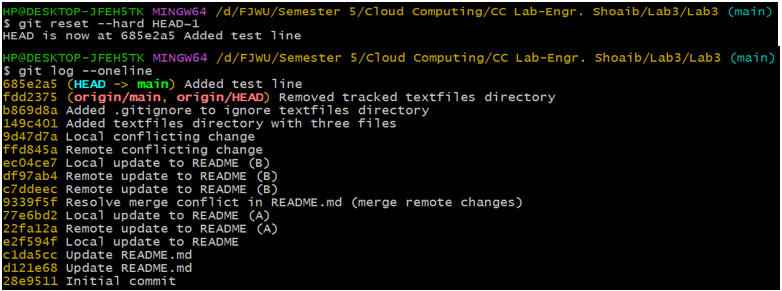
**Step 4: Perform a soft reset (git reset --soft HEAD~1) and observe your file and history.  
Screenshot:**

****

**Step 5: Make commit again.  
Screenshot:**

****

**Step 6: Perform a hard reset (git reset --hard HEAD~1) and observe the changes.  
Screenshot:**

****