

**CLOUD COMPUTING**

**LAB:03**

**Submitted By:**

Seerat Fatima

**Registration.no:**

2023-BSE-060

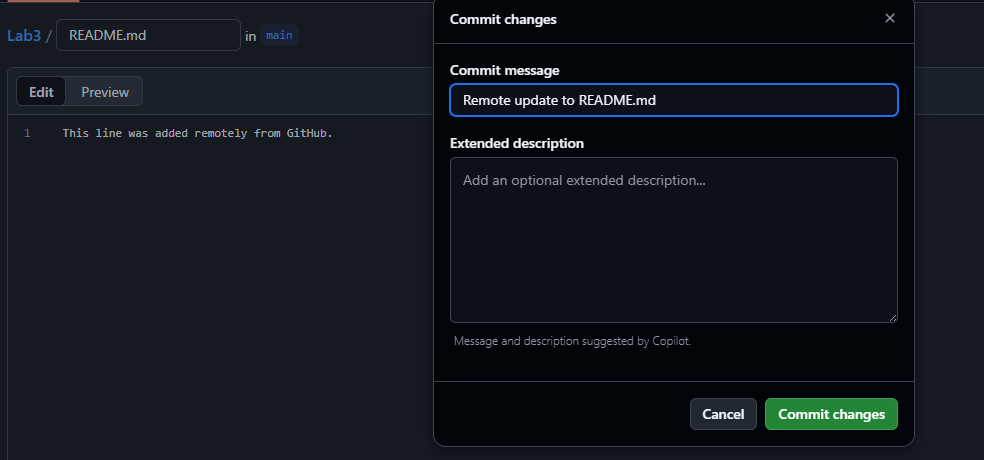
**Section:**

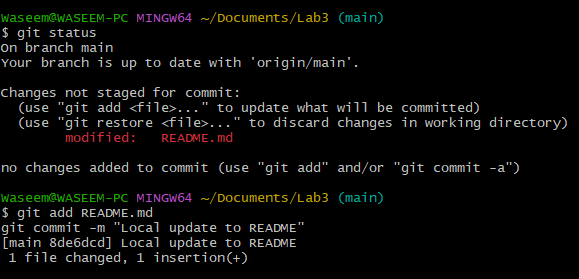
5B

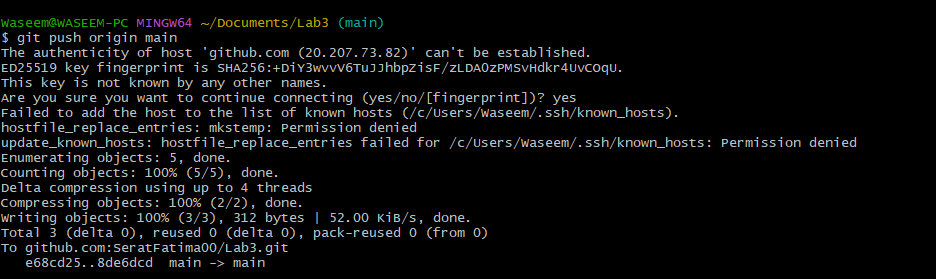
**Submitted To:**

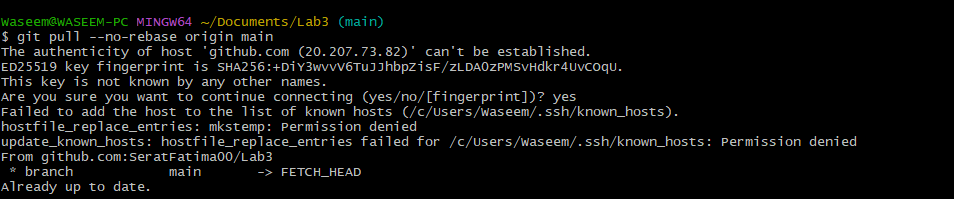
Engr.Shoaib

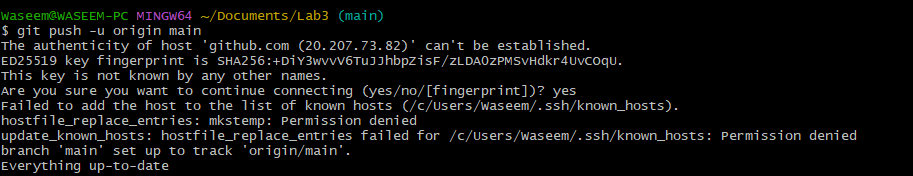
**Task 1 – Handling Local and Remote Commit Conflicts (Pull vs Pull --rebase)**

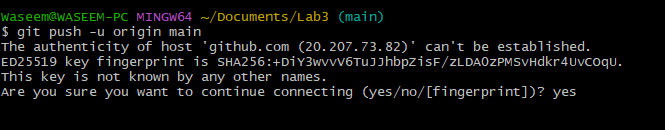




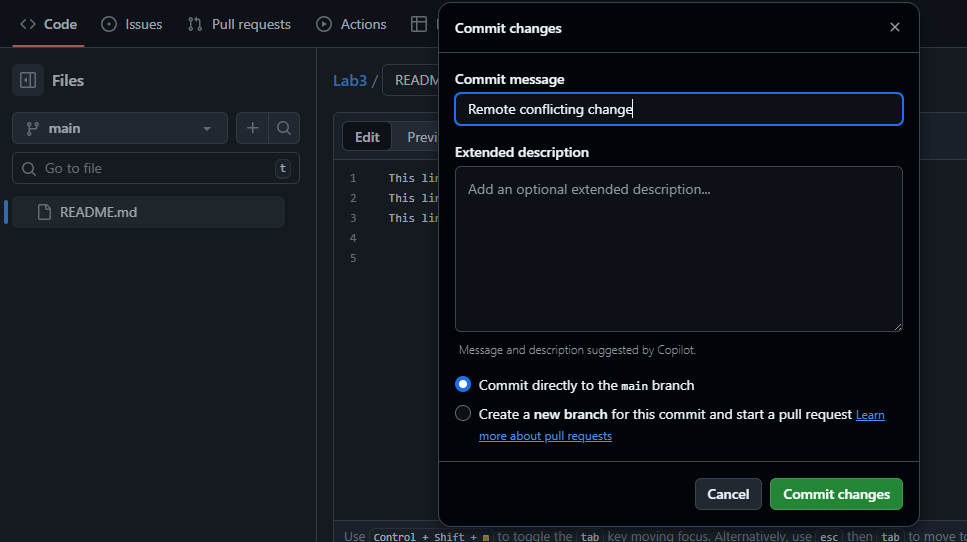


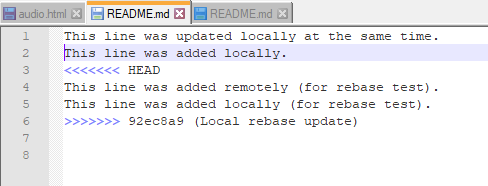


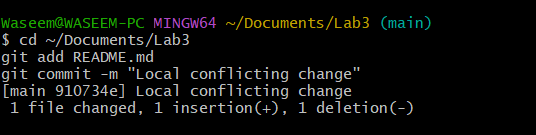


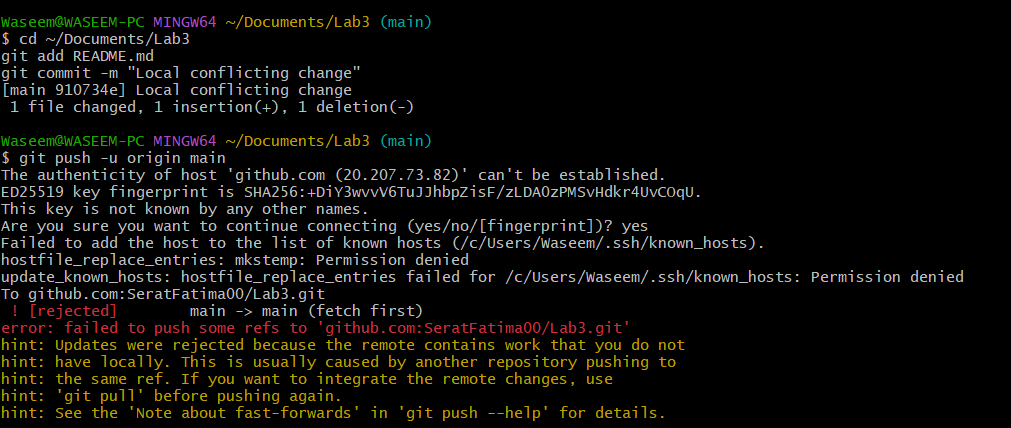


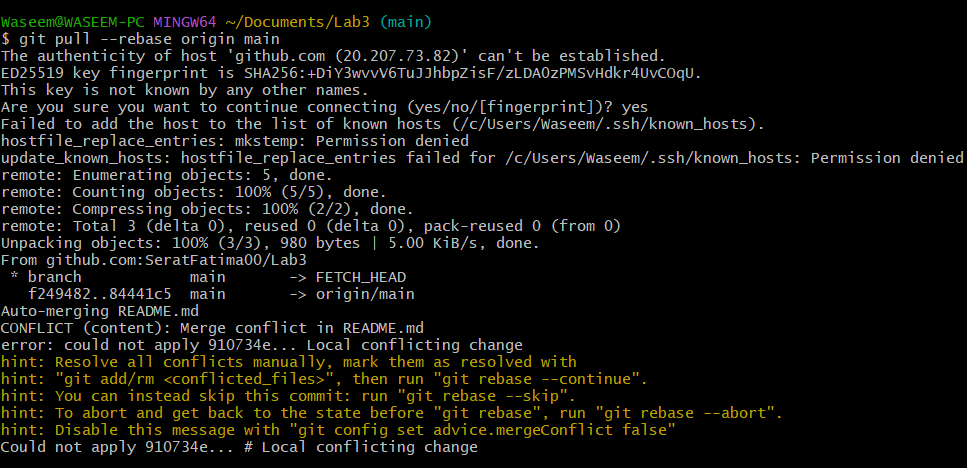
**Task 2 – Creating and Resolving Merge Conflicts Manually**

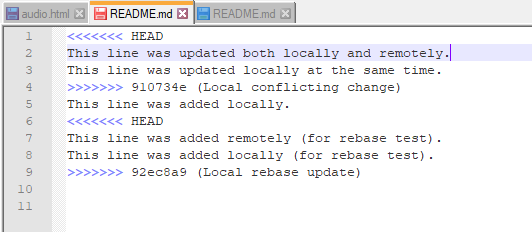


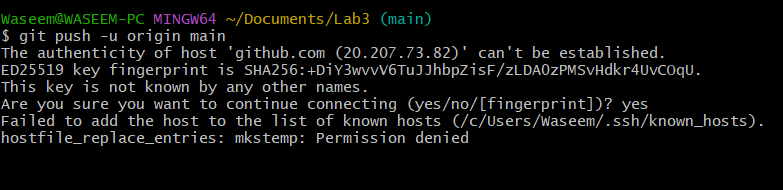


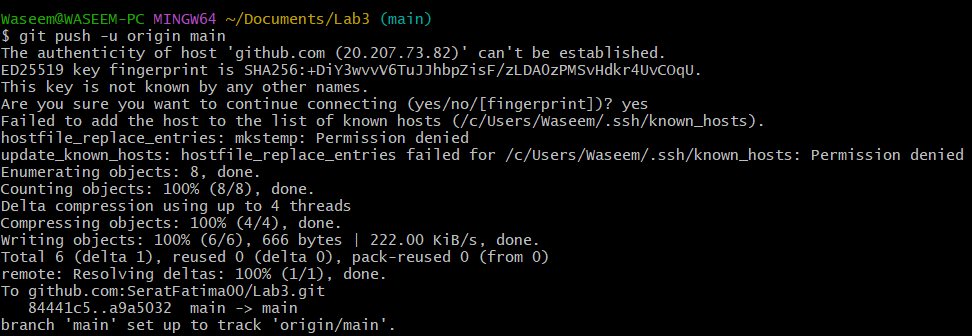




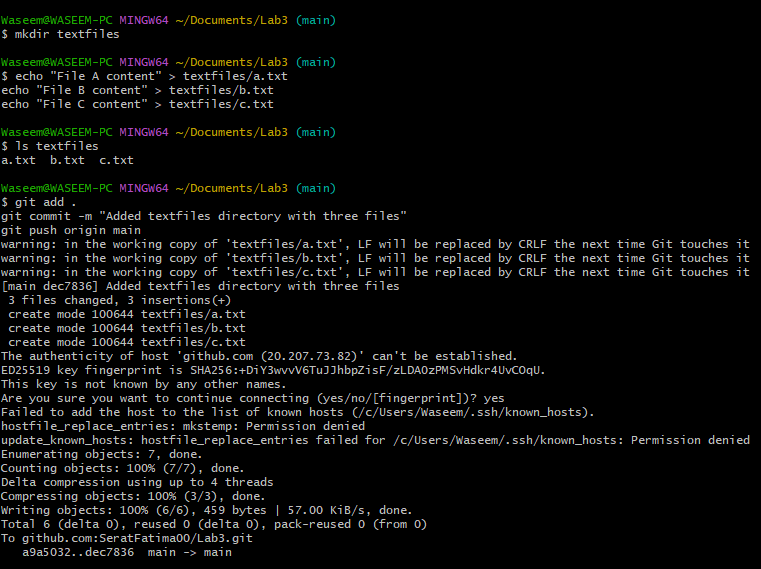


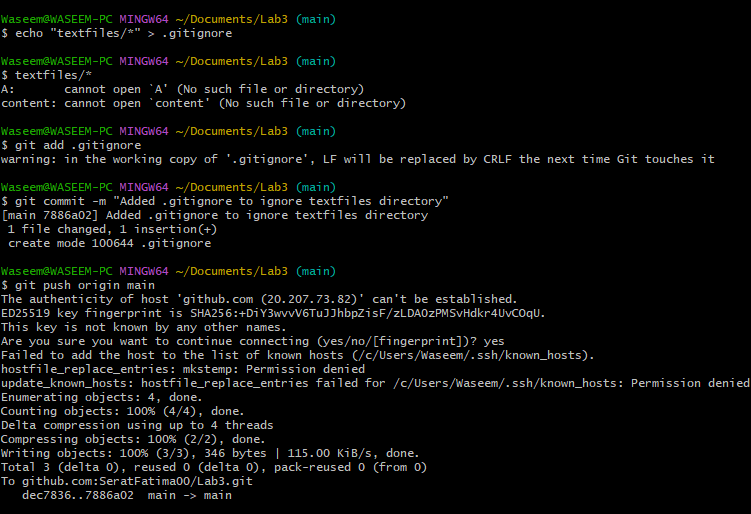


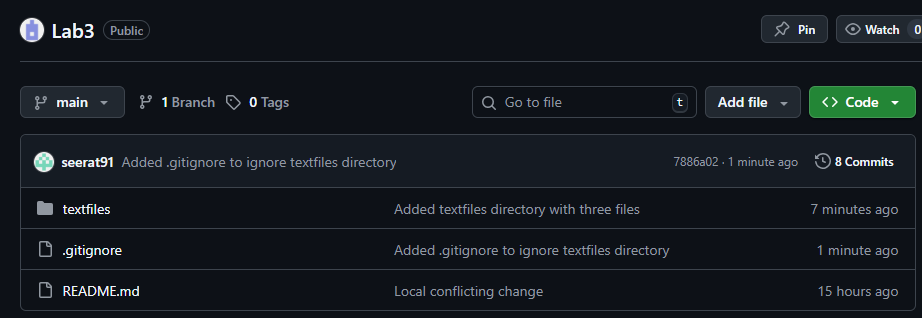


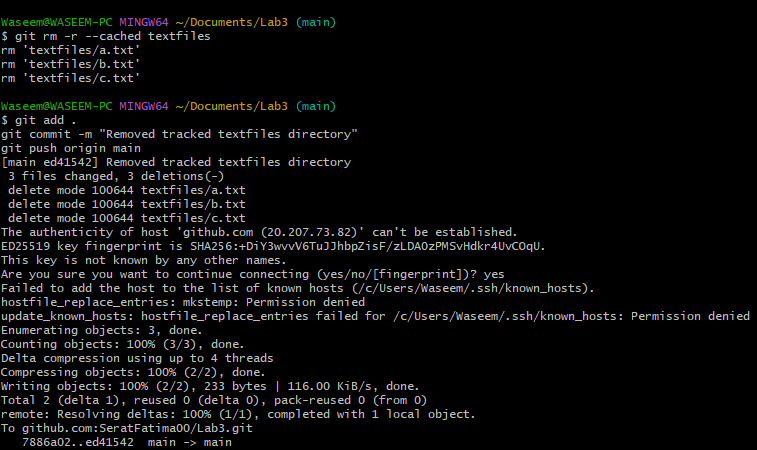


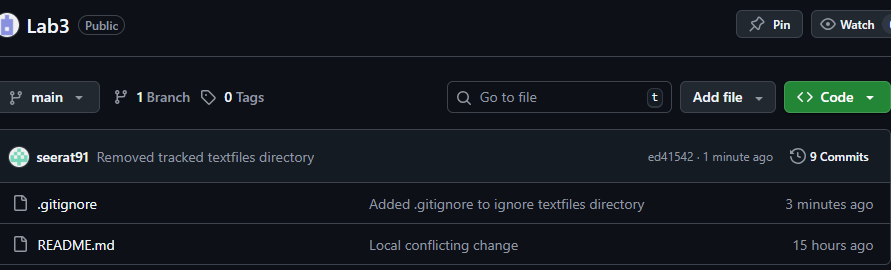
**Task 3 – Managing Ignored Files with .gitignore and Removing Tracked Files**



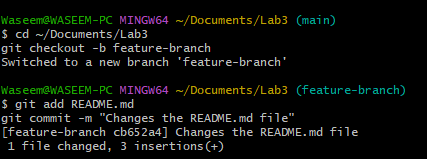


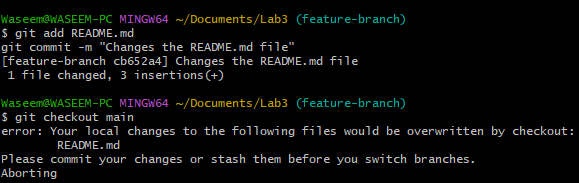




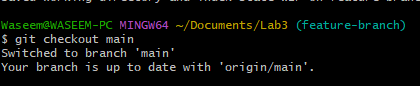


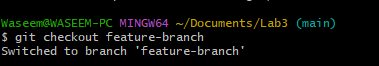
**Task 4 – Create Temporary Changes and Use git stash**

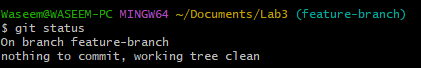


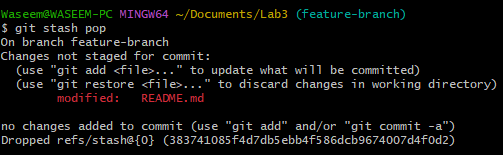




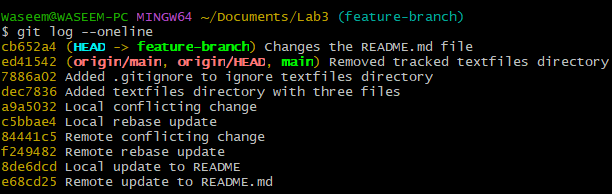


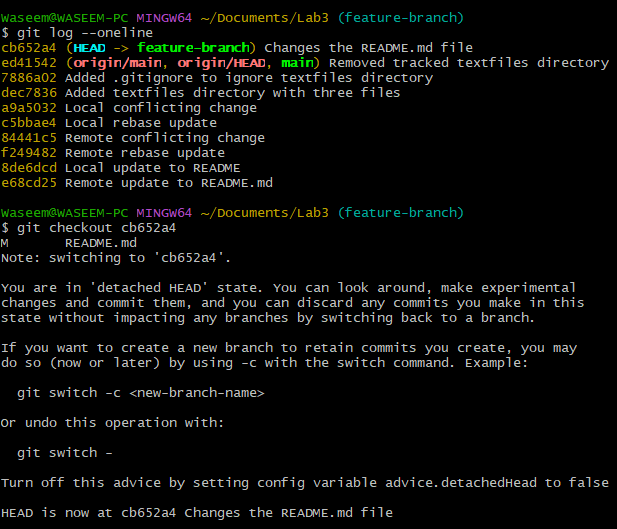


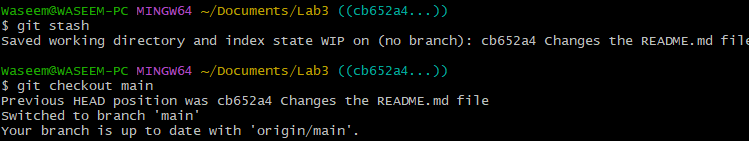




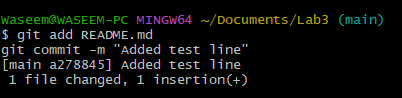
**Task 5 – Checkout a Specific Commit Using git log**

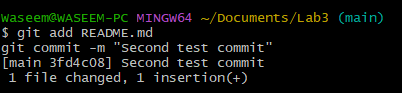


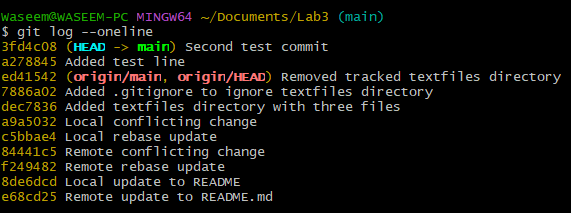




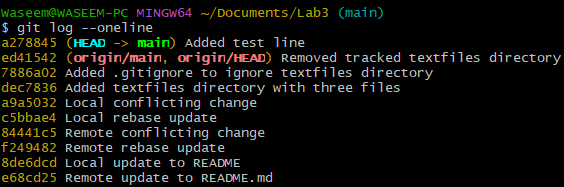
**Task 6 – Resetting Commits (Soft vs Hard Reset) (With Verification Steps)**

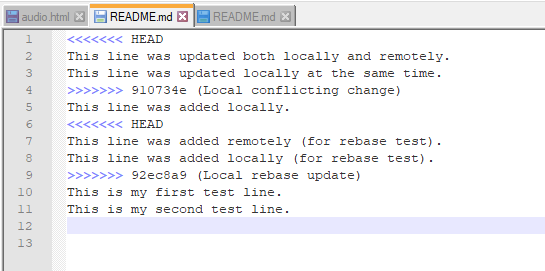


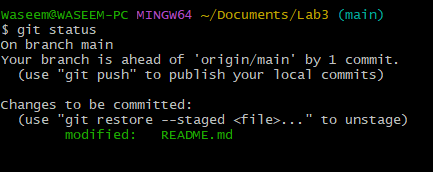




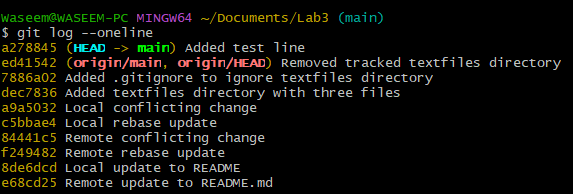


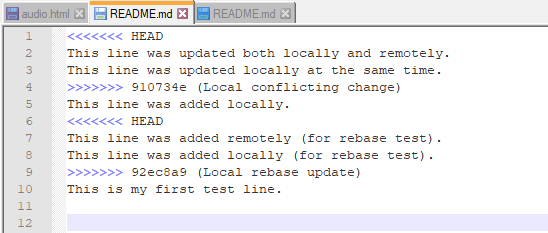




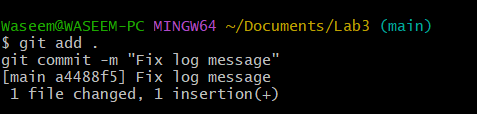


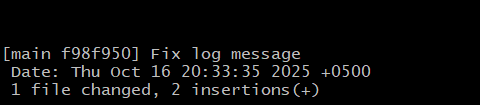




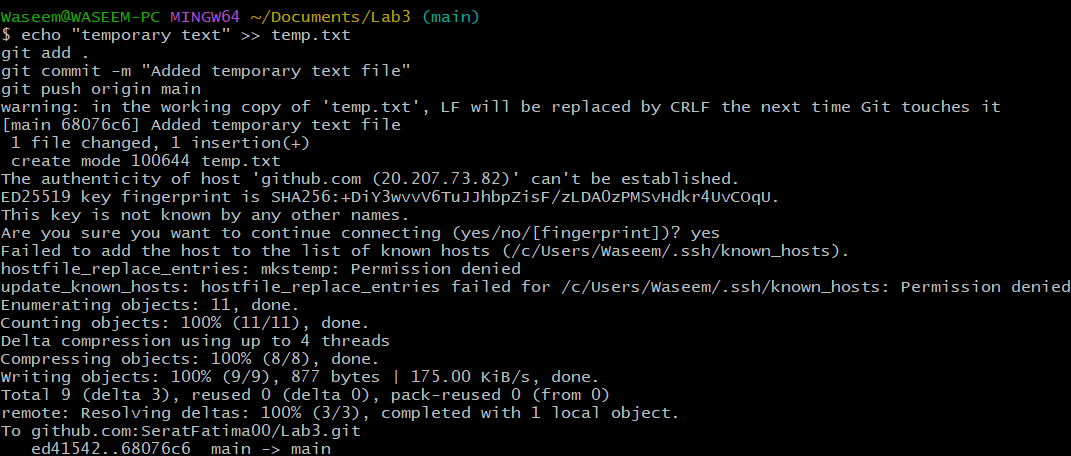


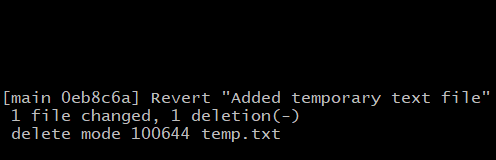
**Task 7 – Amending the Last Commit**

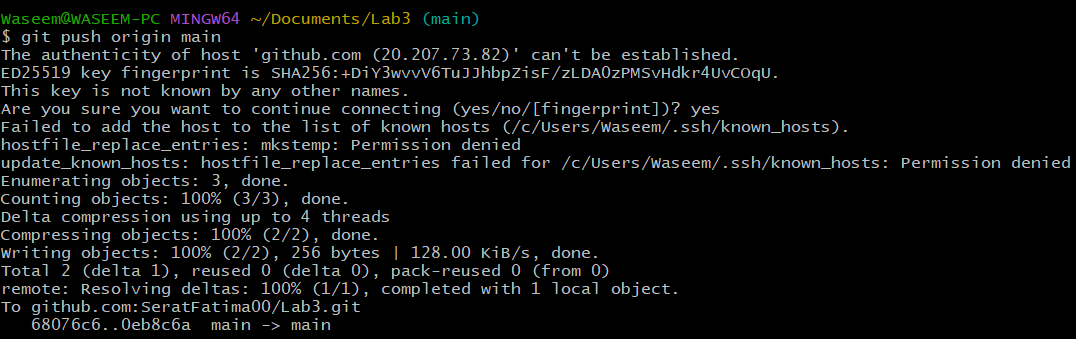




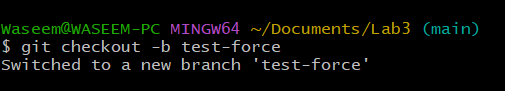
**Task 8 – Reverting a Commit (Safe Undo on Remote Branch)**

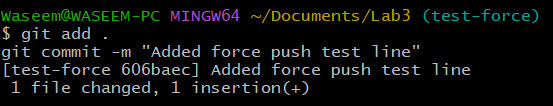


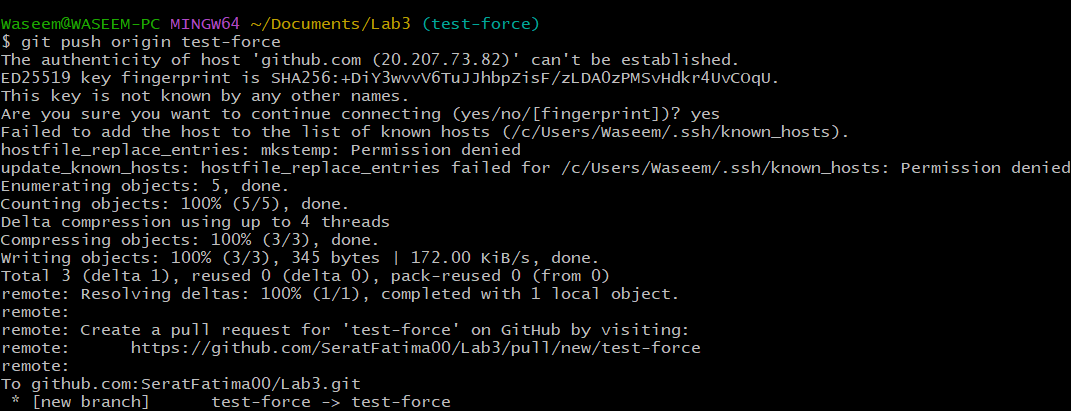


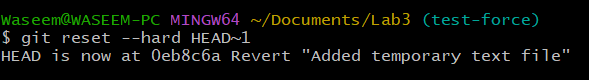


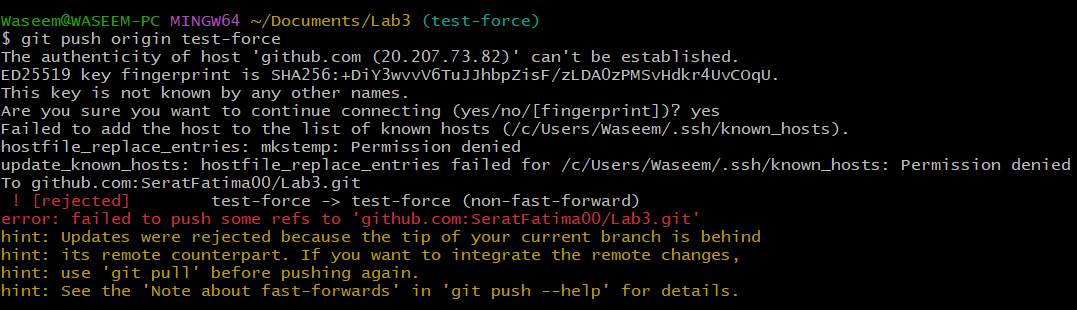
**Task 9 – Force Push (With Caution)**

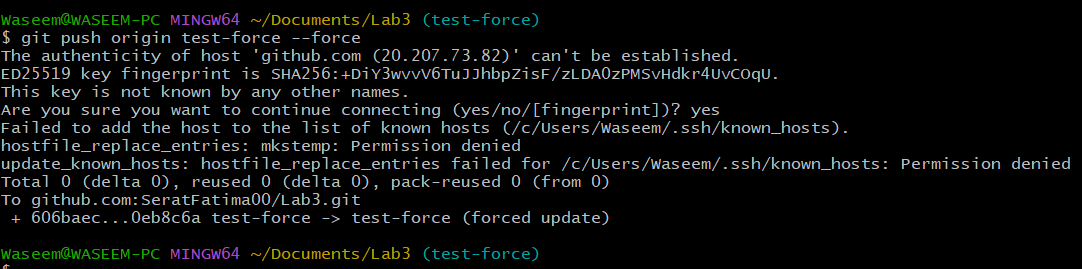




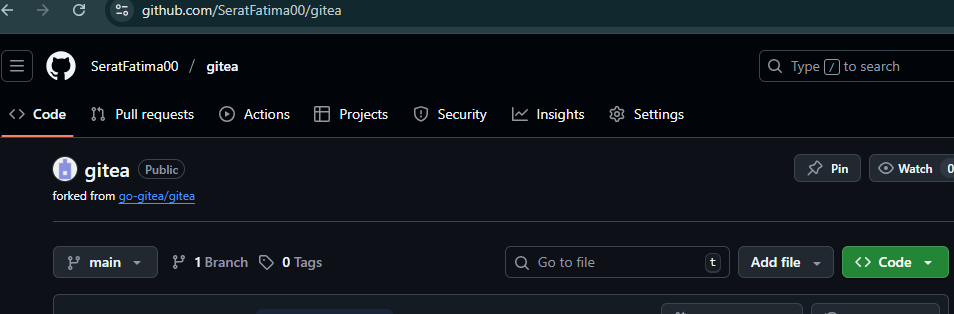


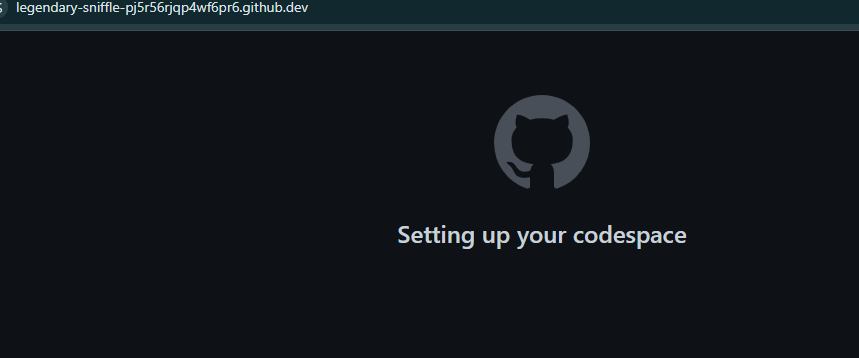


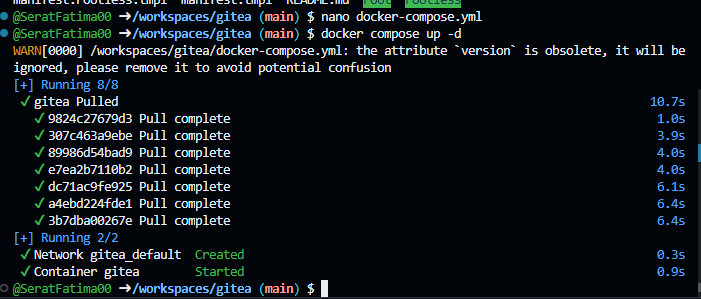


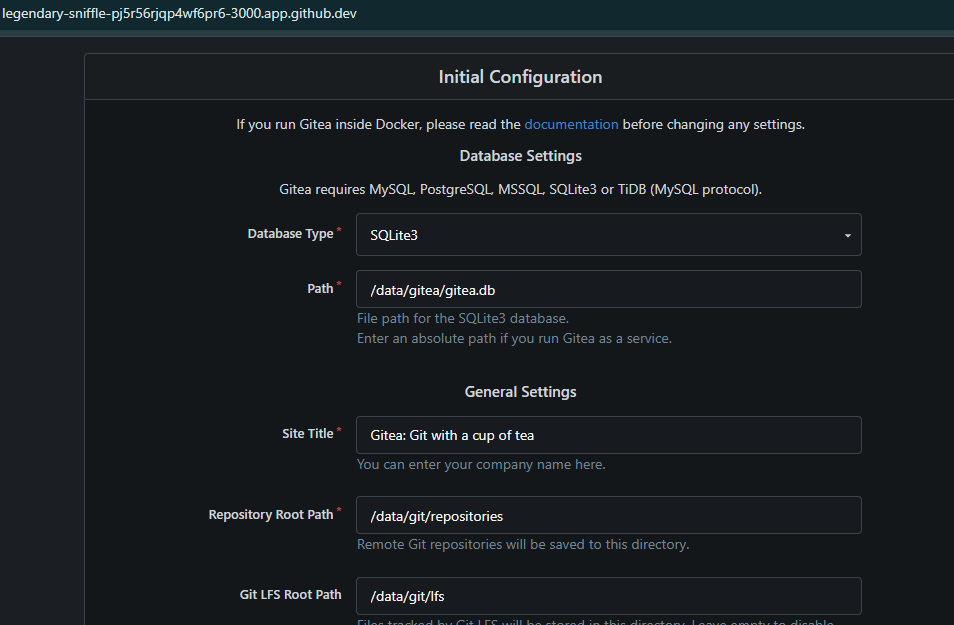


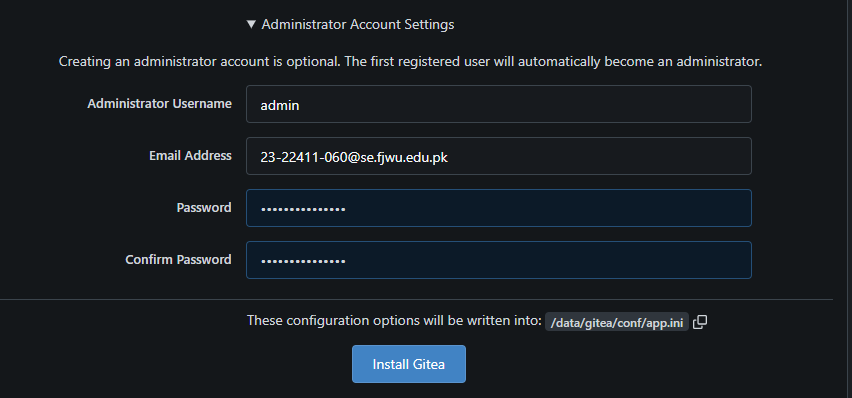
**Task 10 – Running Gitea in GitHub Codespaces via Docker Compose**

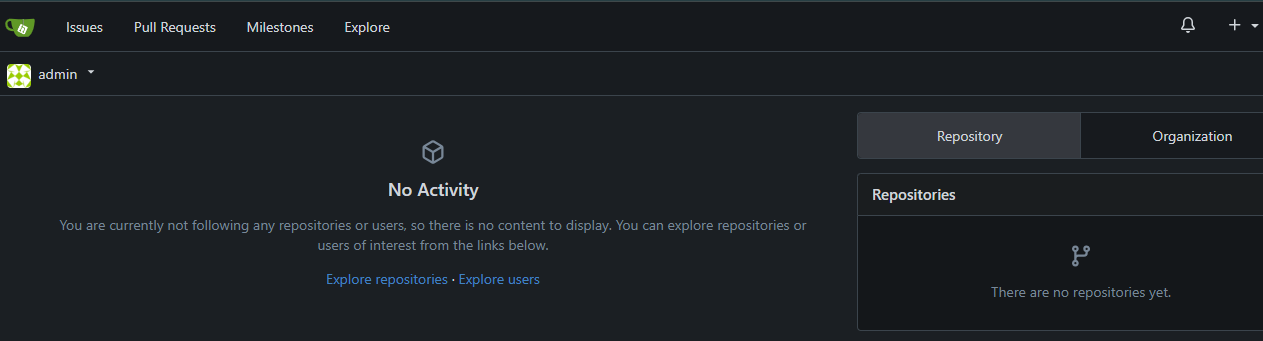


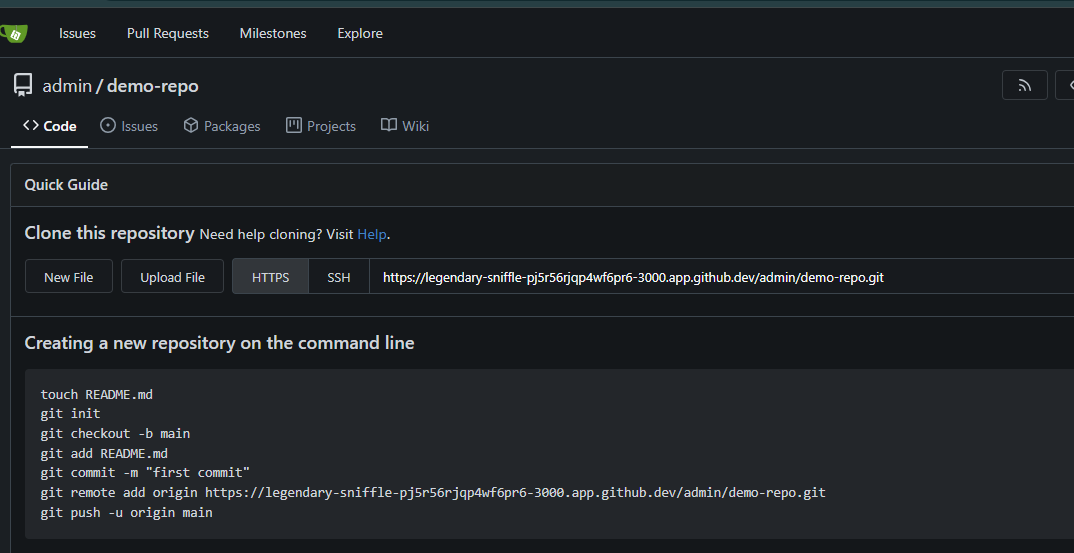






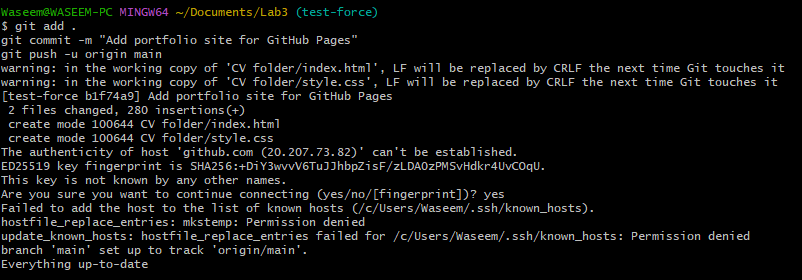


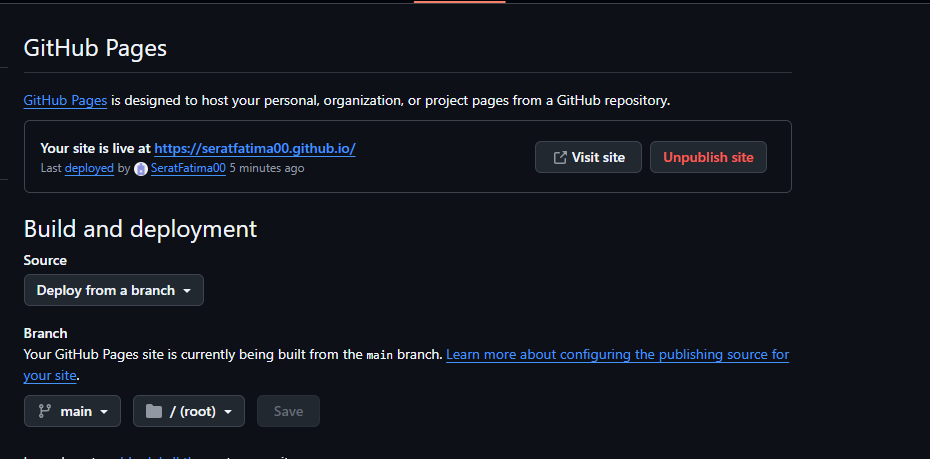


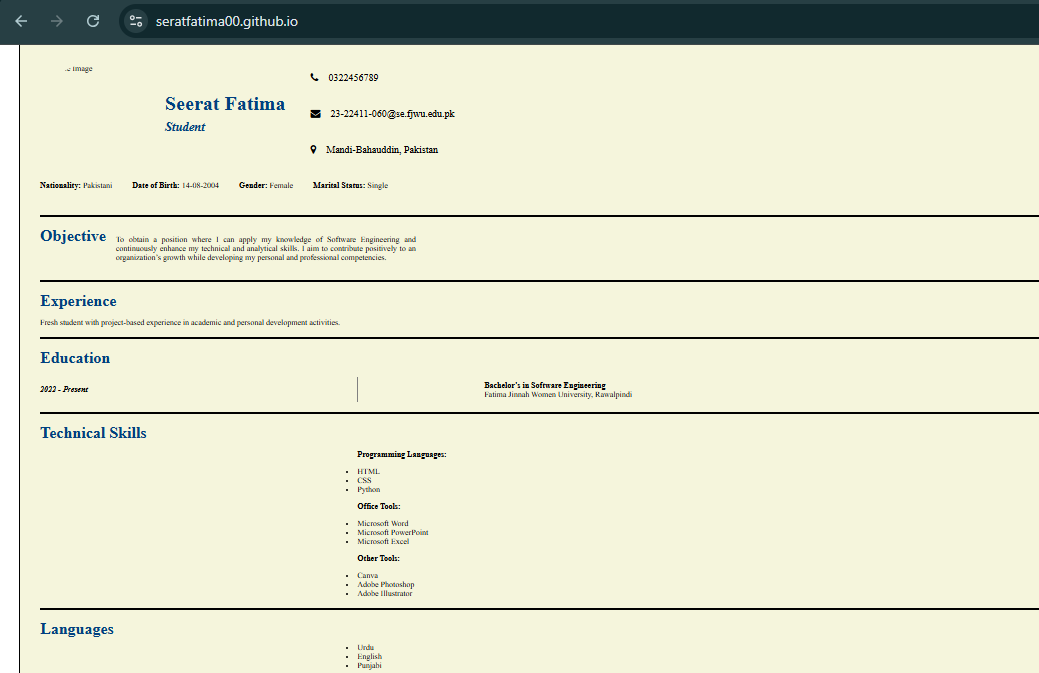


**Task 11 – Creating a GitHub Pages Portfolio Site**







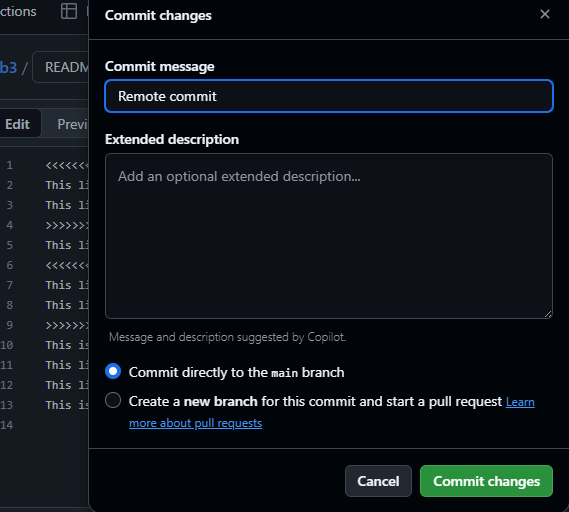


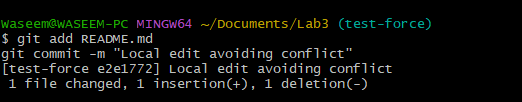
**Exam Evaluation Questions**

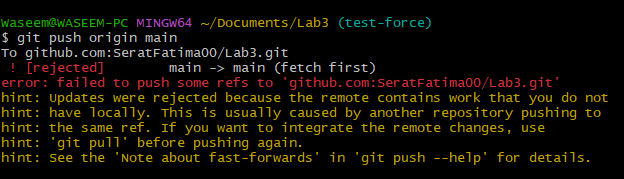
**1. Local vs Remote Conflict Resolution**

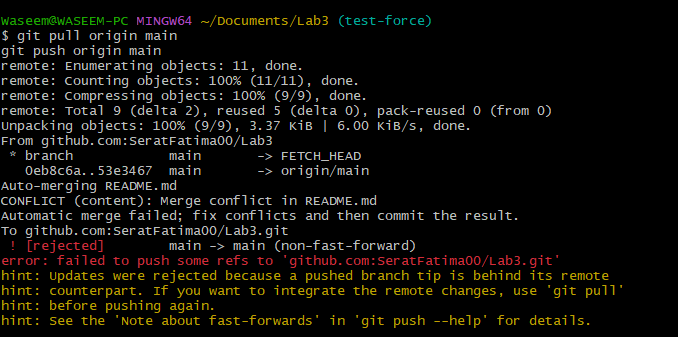
**Steps:**

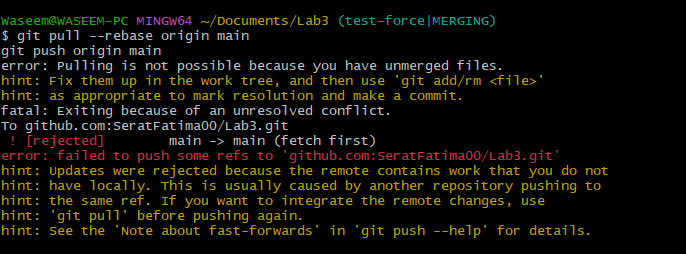
1. On GitHub, edit a file (e.g., README.md) and commit the change.
   * Screenshot as Q1\_remote\_edit.png
2. On your local machine, edit the same file differently (Avoid Conflict) and commit.
   * Screenshot as Q1\_local\_edit.png
3. Try to push your local commit and observe the error.
   * Screenshot as Q1\_push\_error.png
4. Resolve the conflict by running git pull (merge), then push.
   * Screenshot as Q1\_merge\_resolution.png
5. Repeat with another remote/local change, but resolve using git pull --rebase, then push.
   * Screenshot as Q1\_rebase\_resolution.png







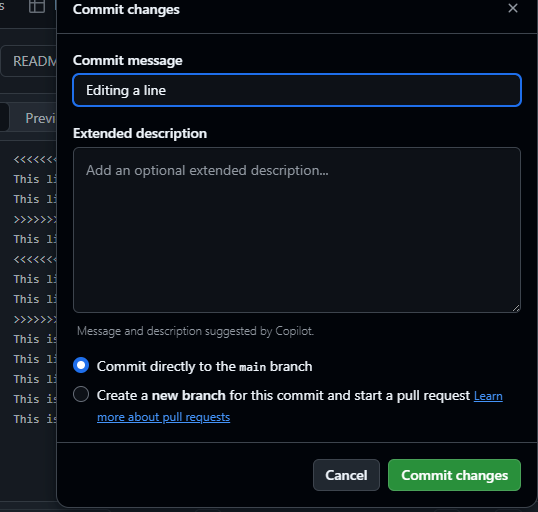


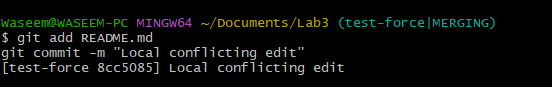


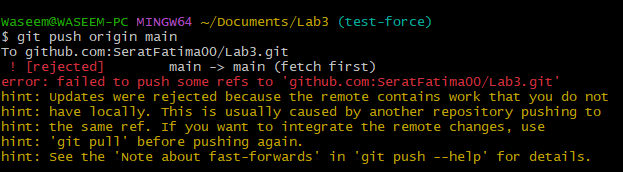
**2. Manual Merge Conflict Handling**

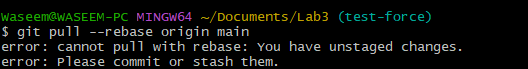
**Steps:**

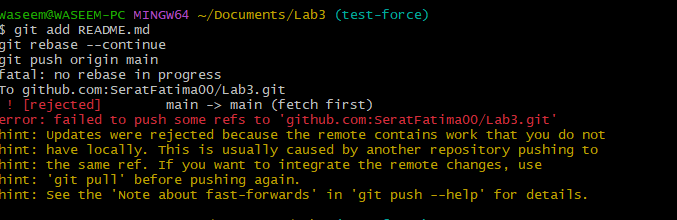
1. On GitHub, change a specific line in a file and commit.
   * Screenshot as Q2\_remote\_conflict\_edit.png
2. Locally, change the same line differently and commit.
   * Screenshot as Q2\_local\_conflict\_edit.png
3. Try to push your local change and observe the conflict error.
   * Screenshot as Q2\_conflict\_push\_error.png
4. Use git pull --rebase to fetch changes and trigger the conflict.
   * Screenshot as Q2\_rebase\_conflict.png
5. Edit the conflicted file to resolve the conflict manually.
   * Screenshot as Q2\_resolved\_file.png
6. Mark the conflict resolved (git add <file>, git rebase --continue) and push.
   * Screenshot as Q2\_resolution\_complete.png









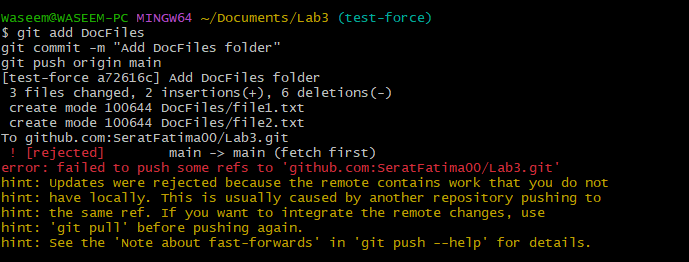


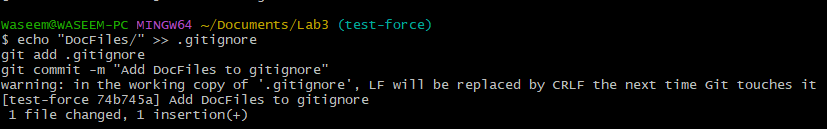
**3. Managing Ignored and Tracked Files**

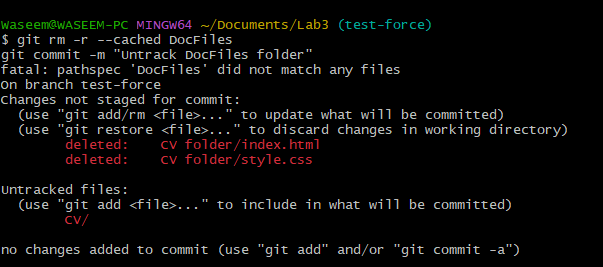
**Steps:**

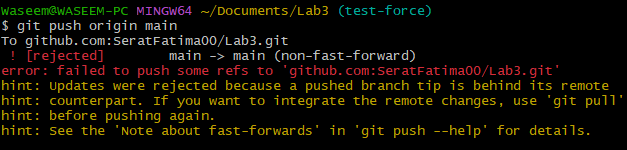
1. Create a new folder (e.g., DocFiles) and add several files inside.
   * Screenshot as Q3\_folder\_created.png
2. Commit and push the folder/files to GitHub.
   * Screenshot as Q3\_files\_pushed.png
3. Add the folder to your .gitignore file.
   * Screenshot as Q3\_gitignore\_added.png
4. Commit and push the .gitignore update.
   * Screenshot as Q3\_gitignore\_pushed.png
5. Remove the folder from tracking using git rm -r --cached <folder>.
   * Screenshot as Q3\_folder\_untracked.png
6. Commit and push the change, then verify the folder is no longer tracked on GitHub.
   * Screenshot as Q3\_folder\_removed\_github.png











**4. Commit History Manipulation and Recovery**

**Steps:**

1. Make a change and commit it.
   * Screenshot as Q4\_first\_commit.png
2. Make another change and commit again.
   * Screenshot as Q4\_second\_commit.png
3. View your commit history.
   * Screenshot as Q4\_commit\_history.png
4. Perform a soft reset (git reset --soft HEAD~1) and observe your file and history.
   * Screenshot as Q4\_soft\_reset.png
5. Make commit again.
   * Screenshot as Q4\_third\_commit.png
6. Perform a hard reset (git reset --hard HEAD~1) and observe the changes.
   * Screenshot as Q4\_hard\_reset.png

