**NAME :** ABHAYKUMAR PADAMANI

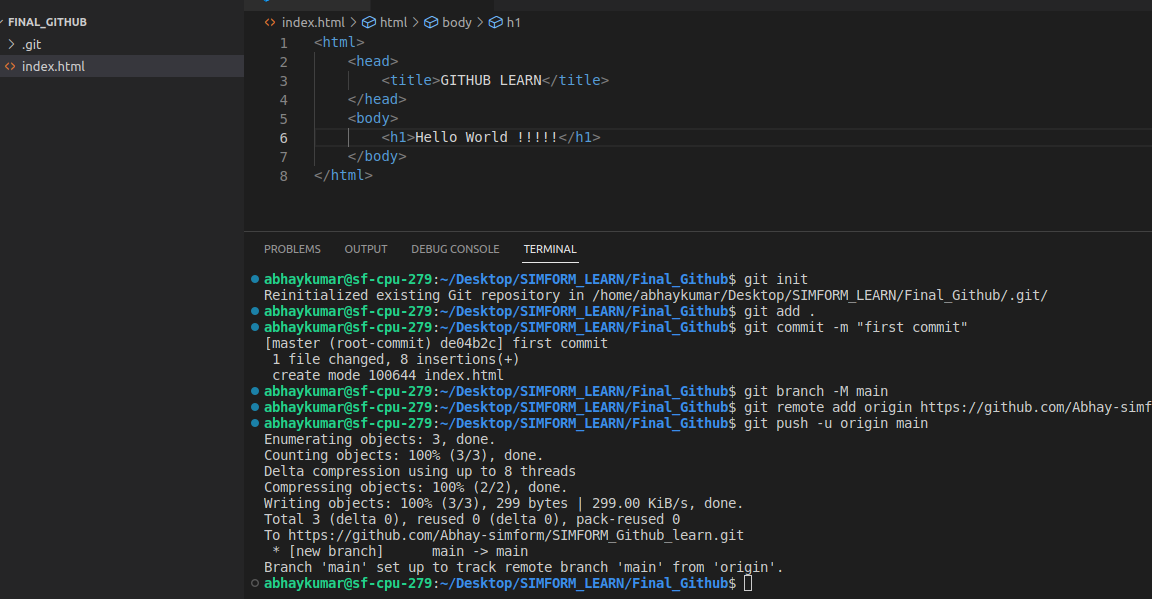
**TOPIC:** GIT

**Mentor:** Jay Motka

**Coordinator :** Kishan Das

**Basic commands:-**

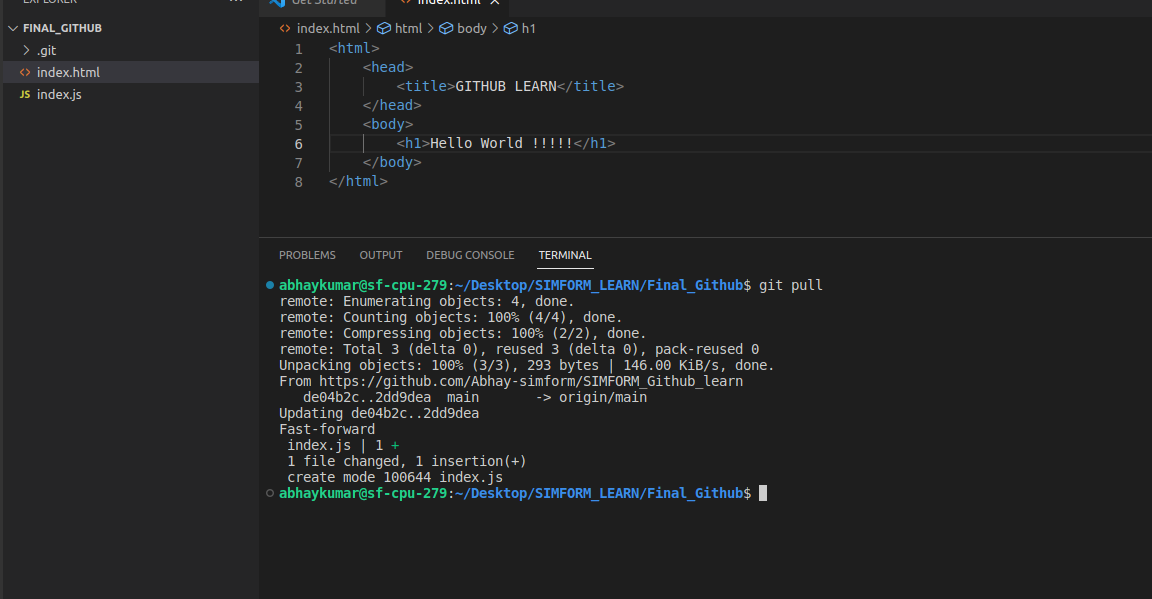
* **Git init**
* **Git commit –m “commit message”**
* **Git push**
* **Git pull**
* **Git status**



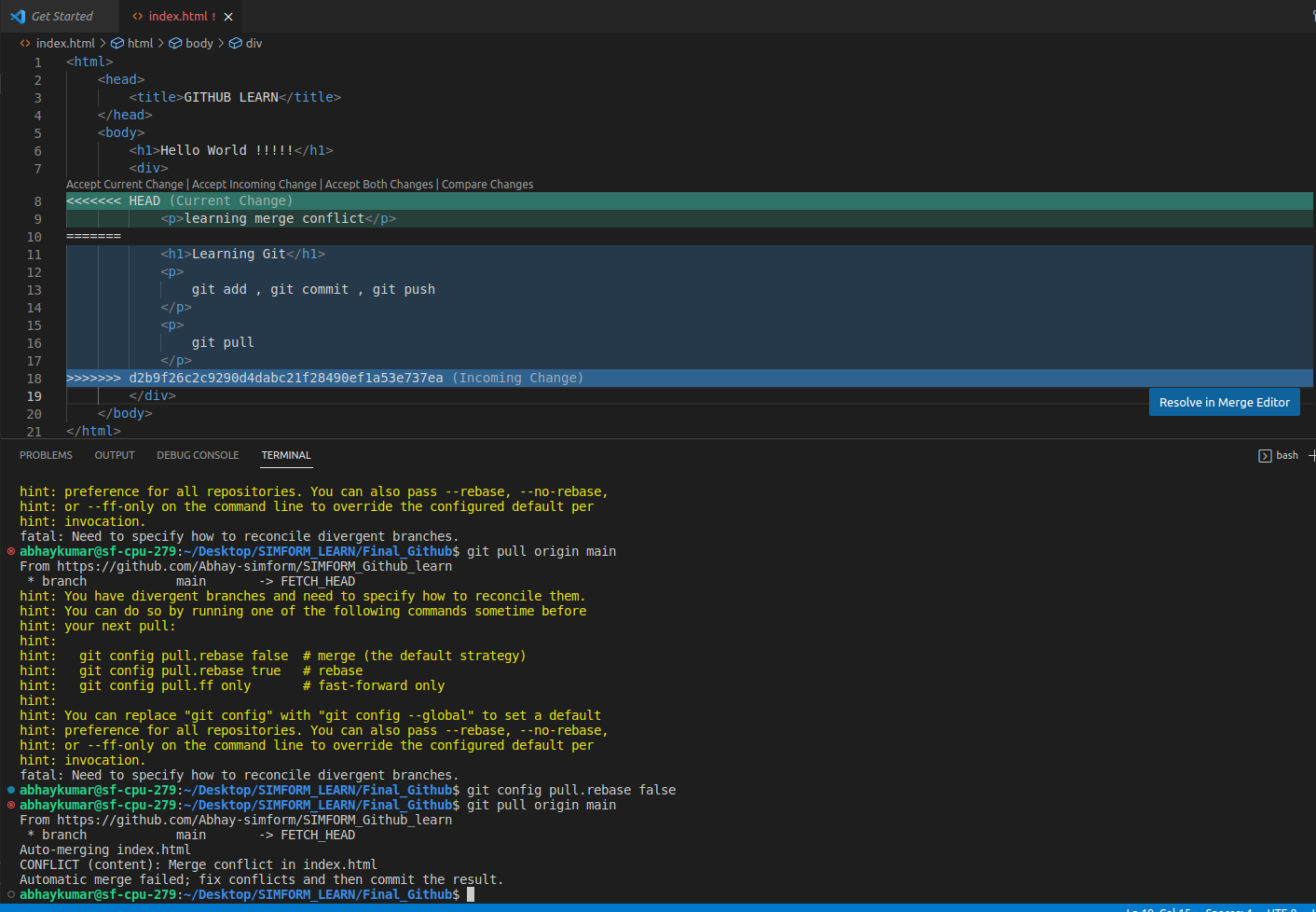
**1. Pull and Merge difference**

**- Make example of pull request and two branch merge event.**

* Git pull

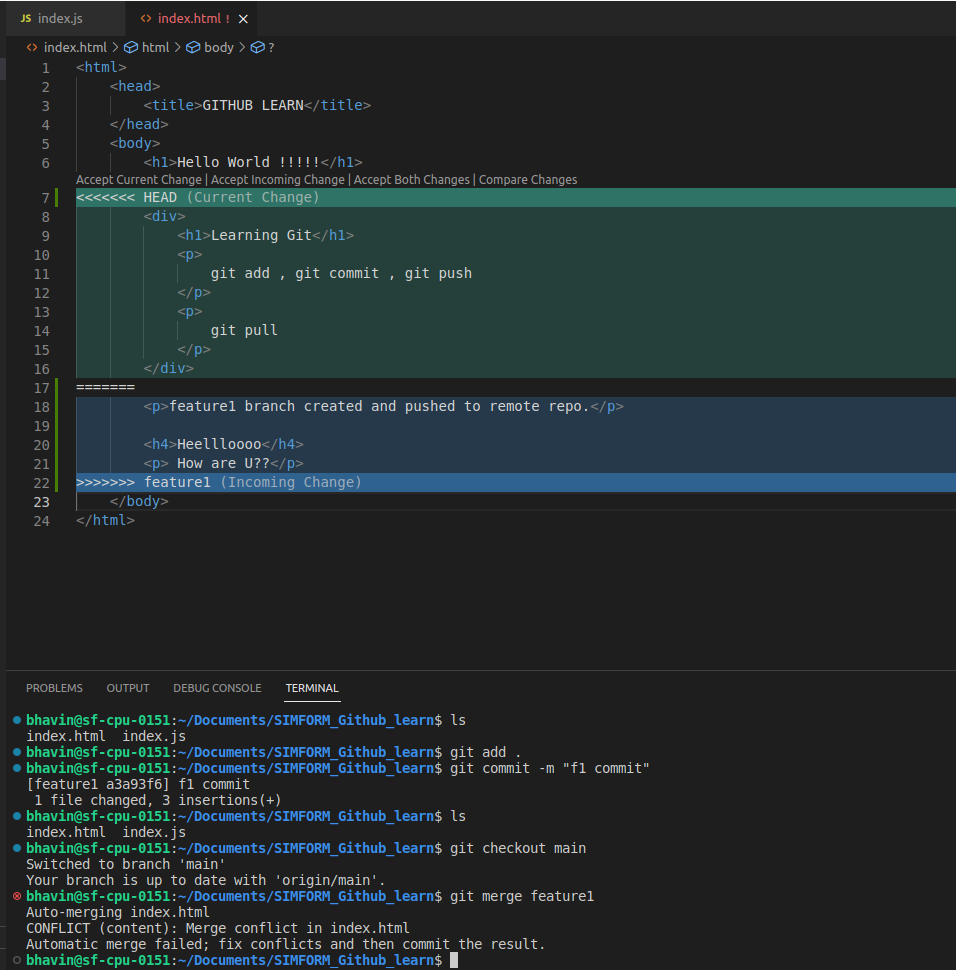


* Merge conflict



**Merge to branch**

* Git merge feature1



**2. Rebase**

**- Try to rebase feature branch with master branch**

To create branch

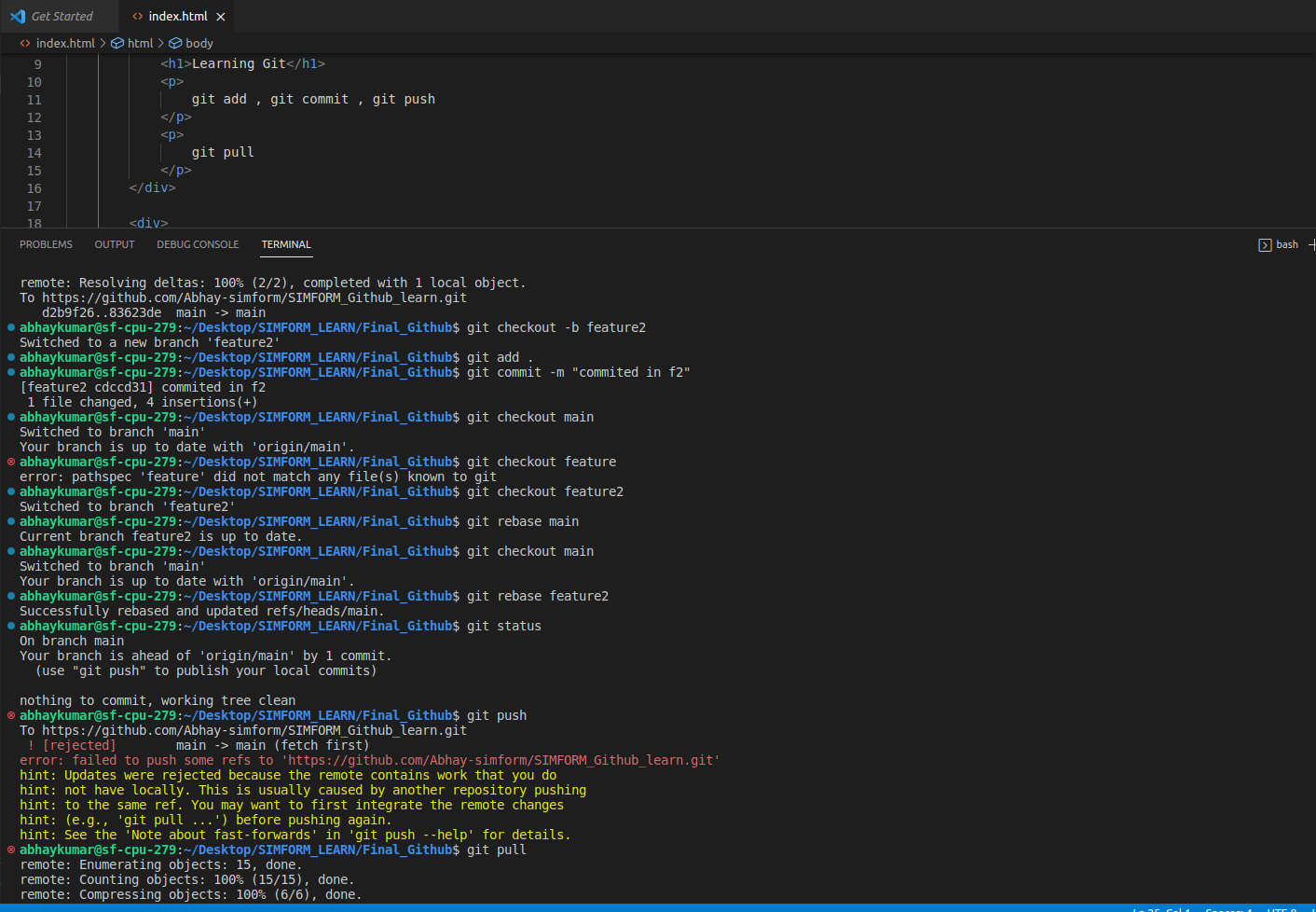
* Git branch feature2

To change branch

* Git checkout feature2

To rebase

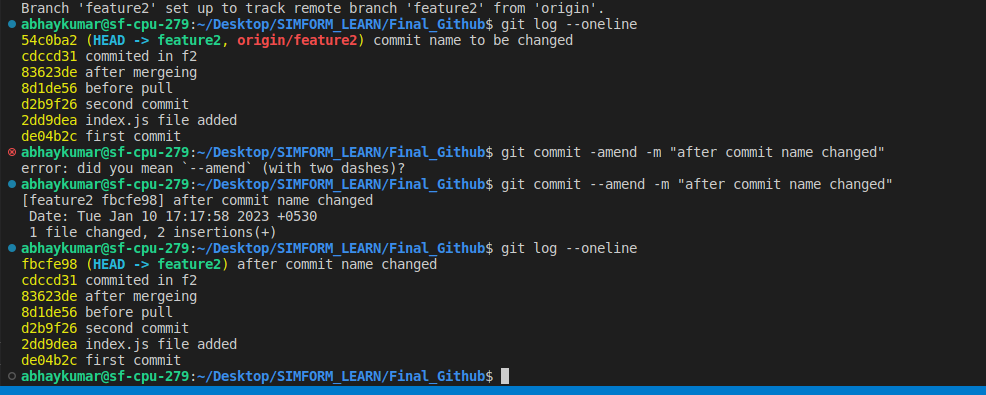
* Git rebase main
* Git rebase feature



**3. Change commit message**

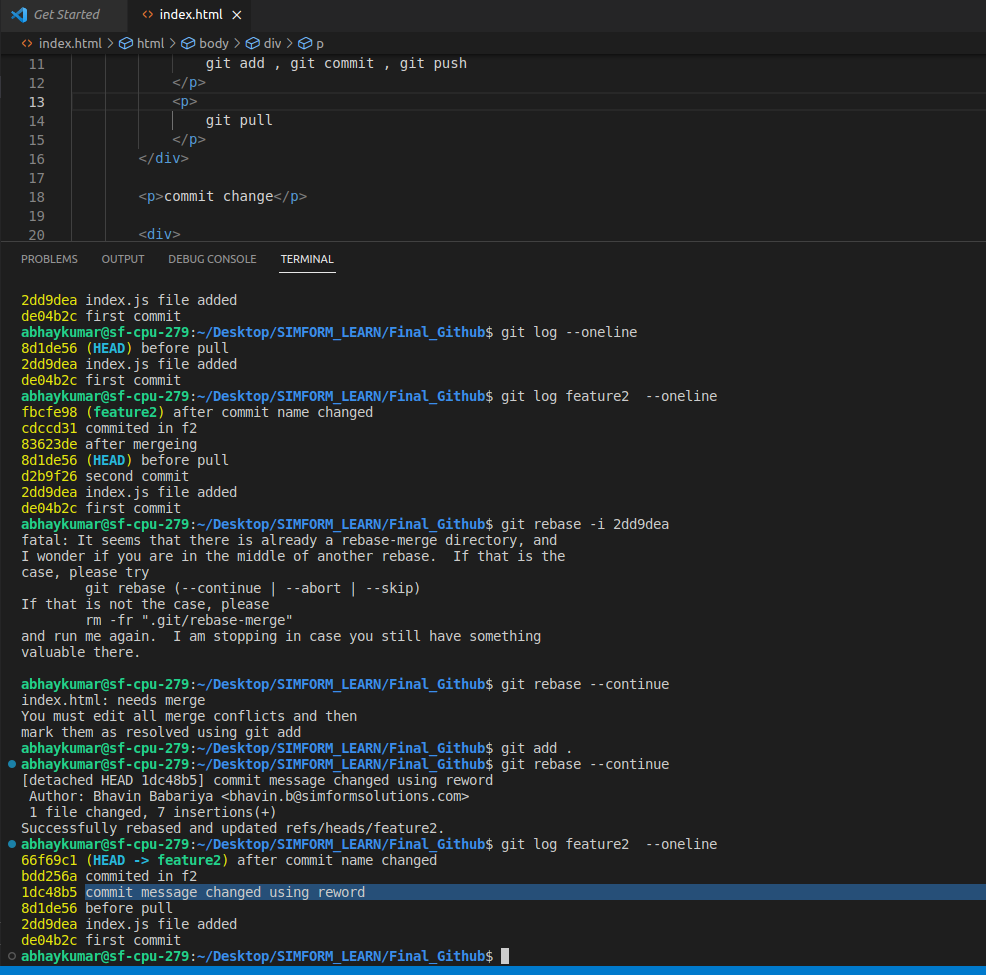
**- Commit push on commit in feature branch and then change commit message**

Git commit -- amend -m “commit message”



**Another way using rebase**

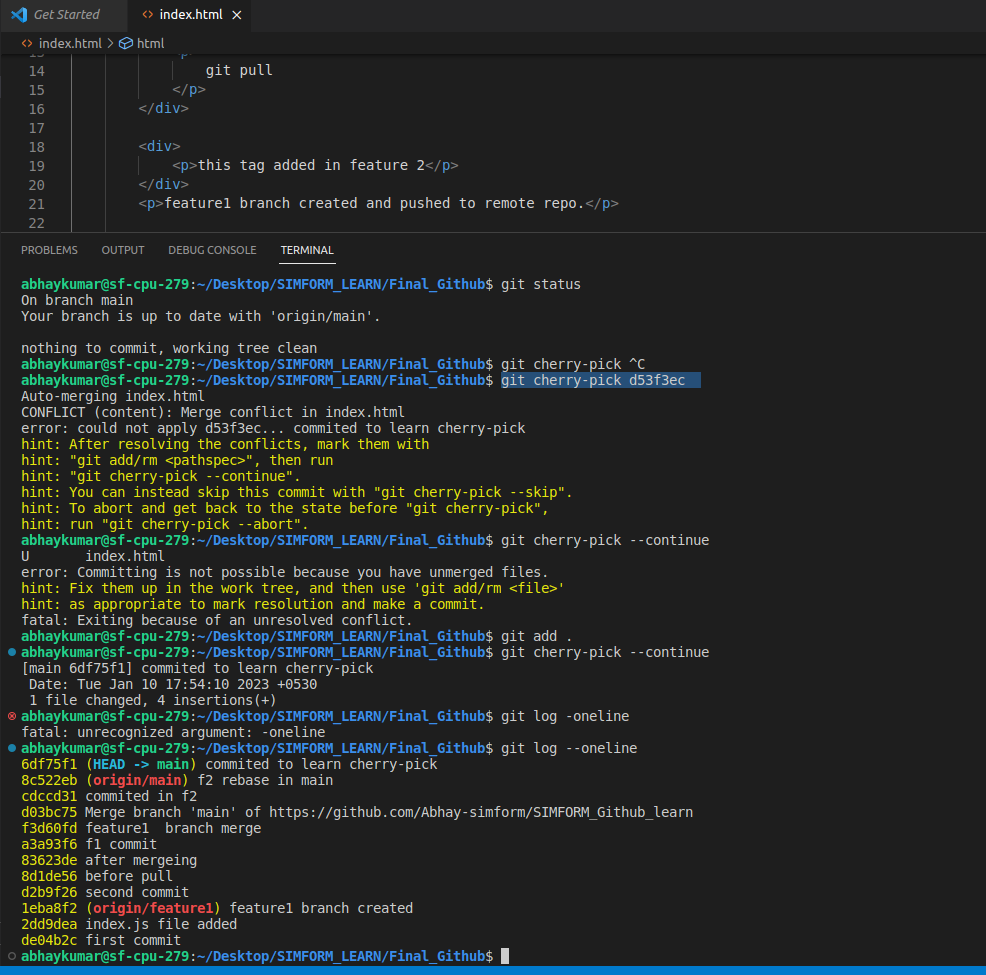
* Git rebase -i [commit hash value]



**4. cherry pick**

**- Pick some commits from feature branch to master branch**

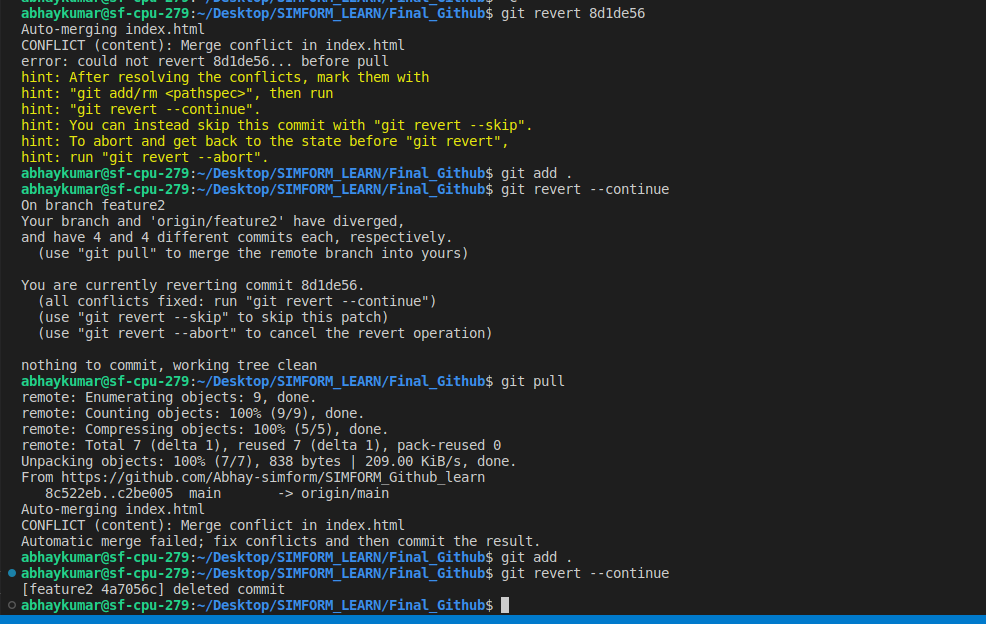
* Git cherry-pick [commit hash value]
* (if merge conflict then we have to continue) git cherry-pick --continue



**5. Drop commit**

**- Remove some commit from feature branch.**

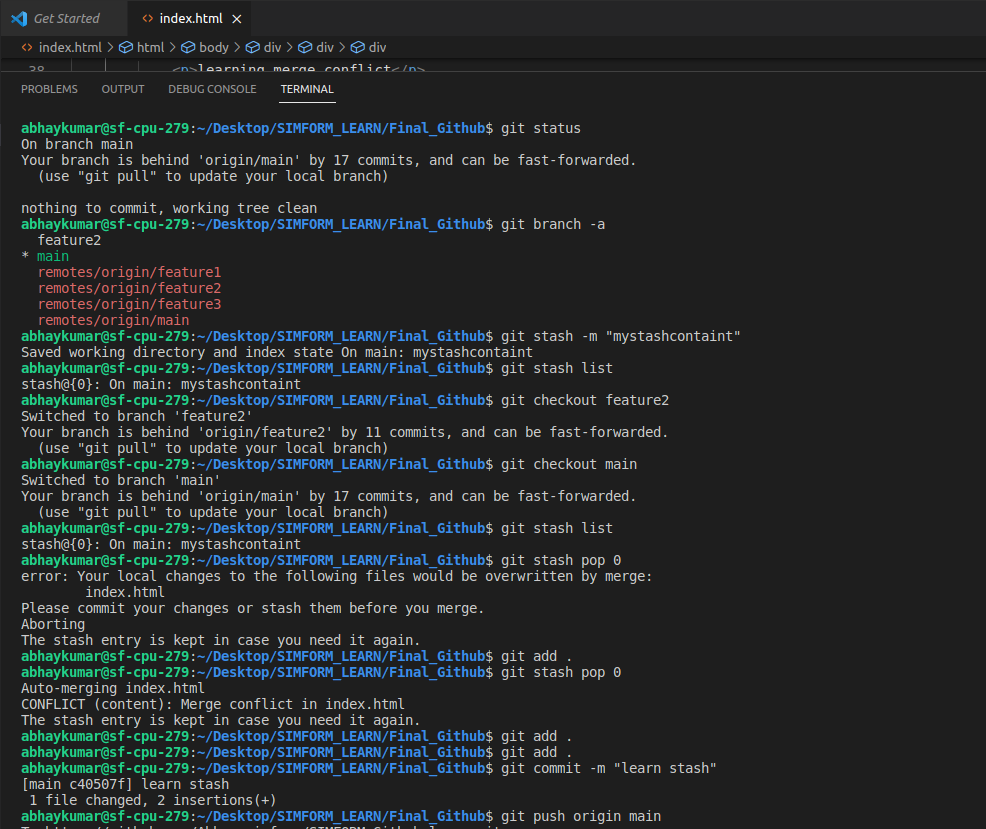
* Git revert [commit hash value]



**change branch without commit using Stash**

* git stash -m “stash name”
* git stash list
* git pop 0(index)

after solve conflict



**combine commit using squash**

* git rebase -i HEAD~3
* then edit in vi editor which we have to mern with last commit
* write **squash** in front of commit
* solve conflict and merge

