Task:

Add a text file called version01.txt inside the Devops/Git/ with “This is first feature of our application” written inside. This should be in a branch coming from master, [hint try git checkout -b dev], swithch to dev branch ( Make sure your commit message will reflect as "Added new feature"). [Hint use your knowledge of creating branches and Git commit command]

* version01.txt should reflect at local repo first followed by Remote repo for review. [Hint use your knowledge of Git push and git pull commands here]

Add new commit in dev branch after adding below mentioned content in Devops/Git/version01.txt: While writing the file make sure you write these lines

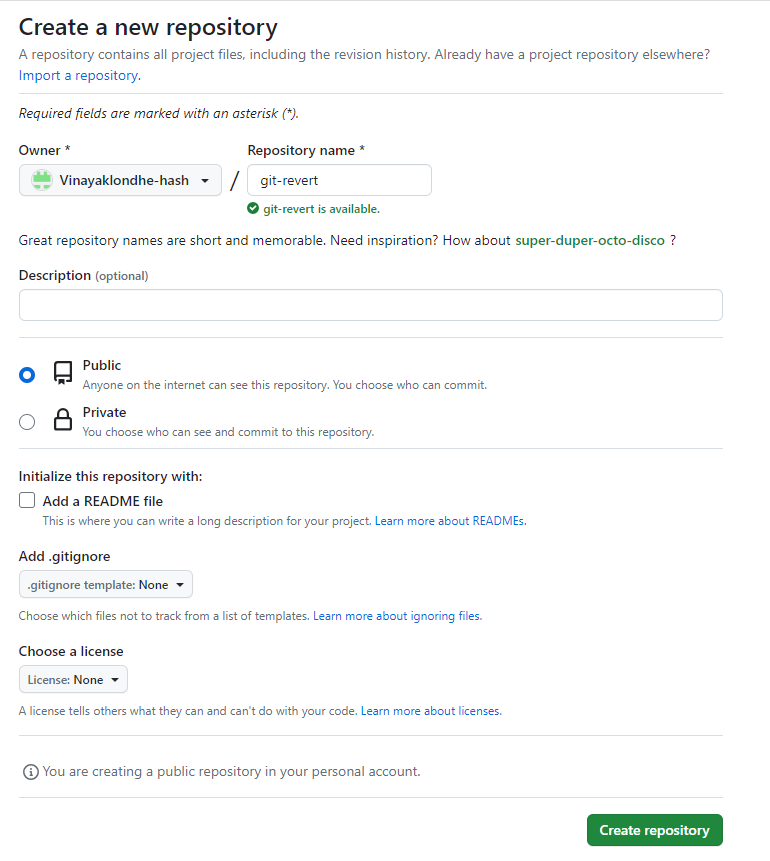
* 1st line>> This is the bug fix in development branch
* Commit this with message “ Added feature2 in development branch”
* 2nd line>> This is gadbad code
* Commit this with message “ Added feature3 in development branch
* 3rd line>> This feature will gadbad everything from now.
* Commit with message “ Added feature4 in development branch

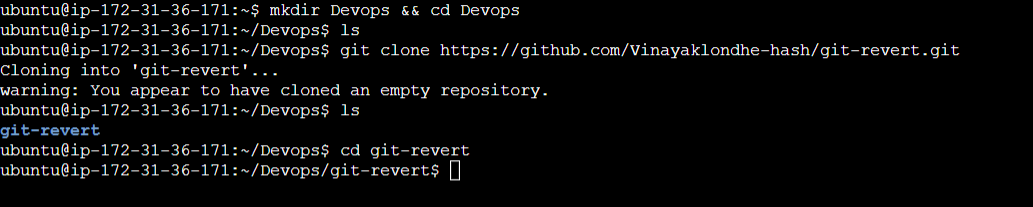
Restore the file to a previous version where the content should be “This is the bug fix in development branch” [Hint use git revert or reset according to your knowledge]

Solution:

**Lets start with git revert,**

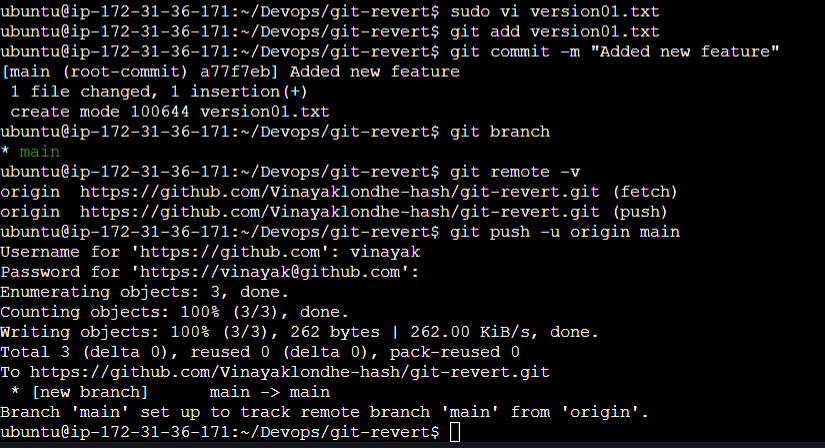
Lets create a github repo with git-revert as a name and clone it under our local git directory called Devops

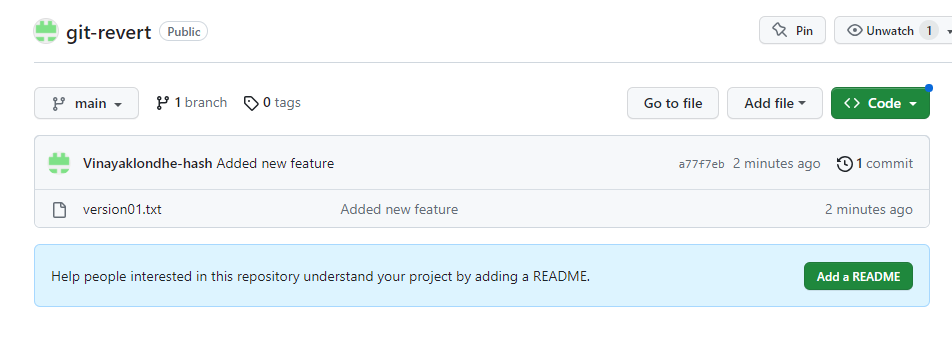




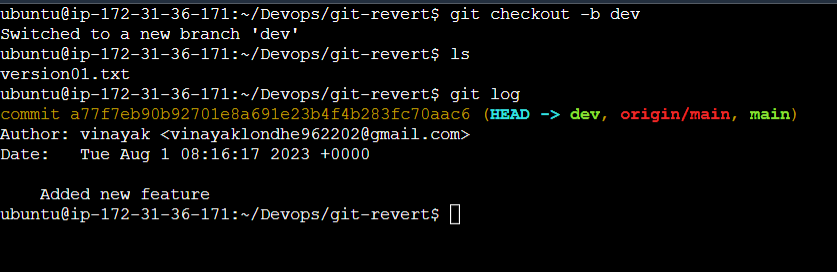
Now lets create version01.txt file & add content “This is first feature of our application”

& push it to github repo ,



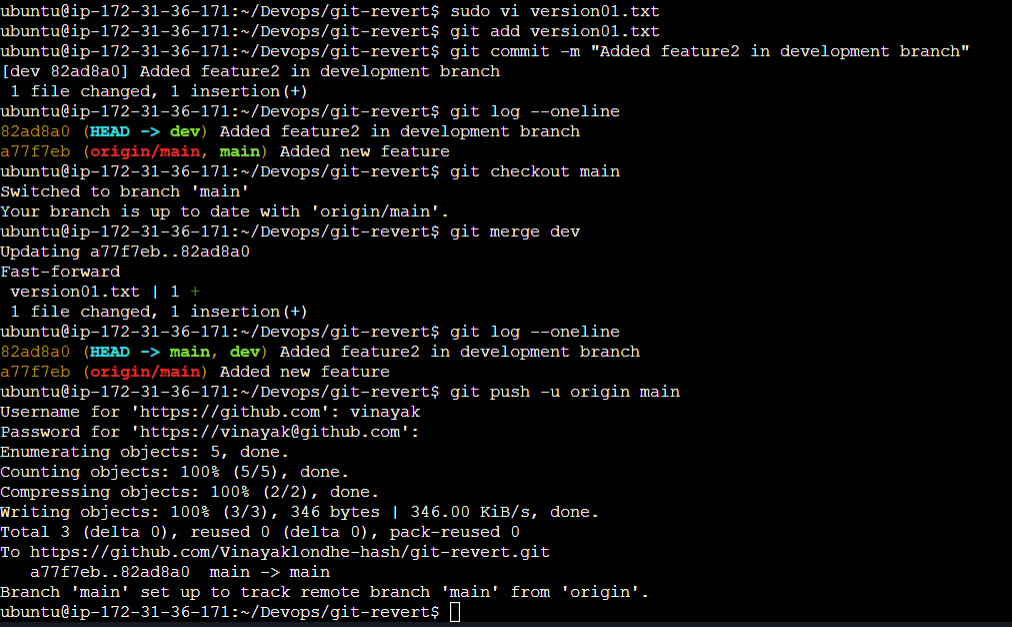


Now lets checkout dev branch ( here you can see Head is pointing to dev ,main that means both branches are in sync)



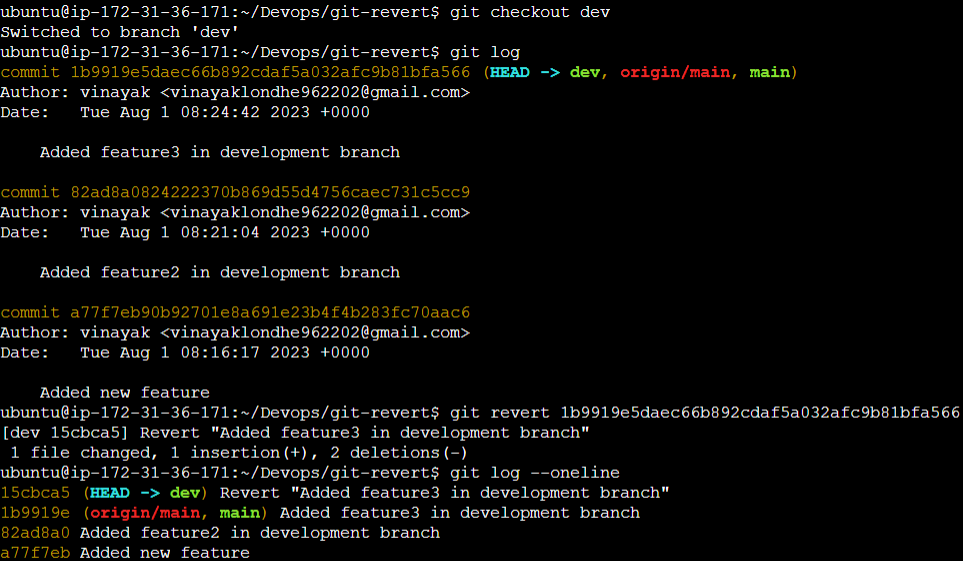
Now add “This is the bug fix in development branch” inside file with commit msg as

“Added feature2 in development branch” on dev branch & merge to main & push it to github repo

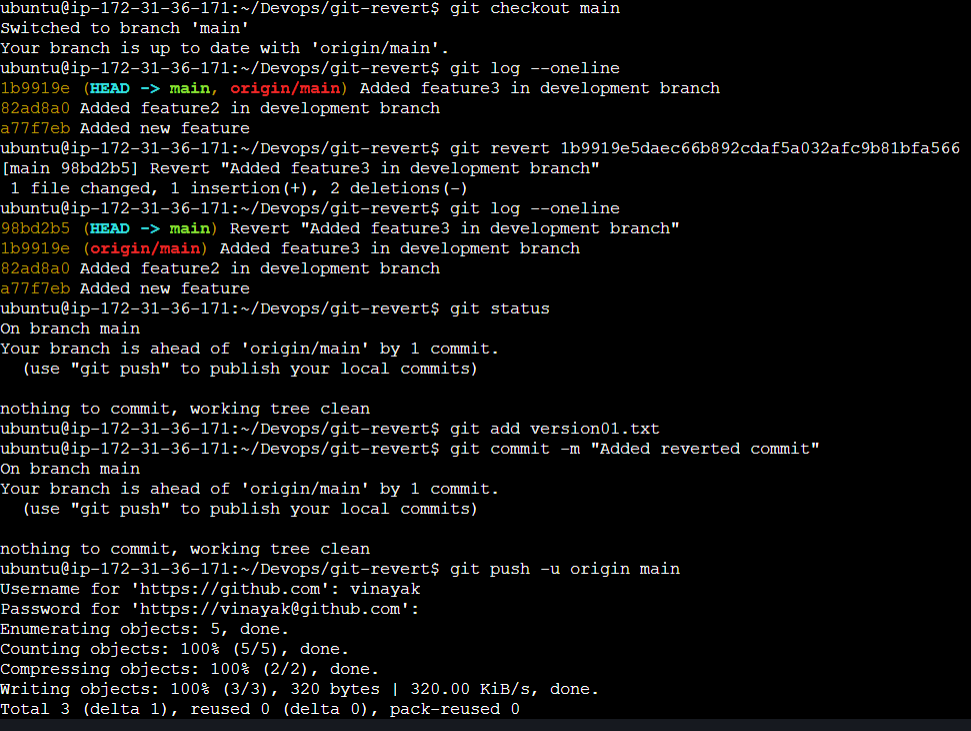


Now push 3rd commit to main as before & since 3rd commit **(check task)** is wrong we need to revert it from main & dev & then push it to github from main branch

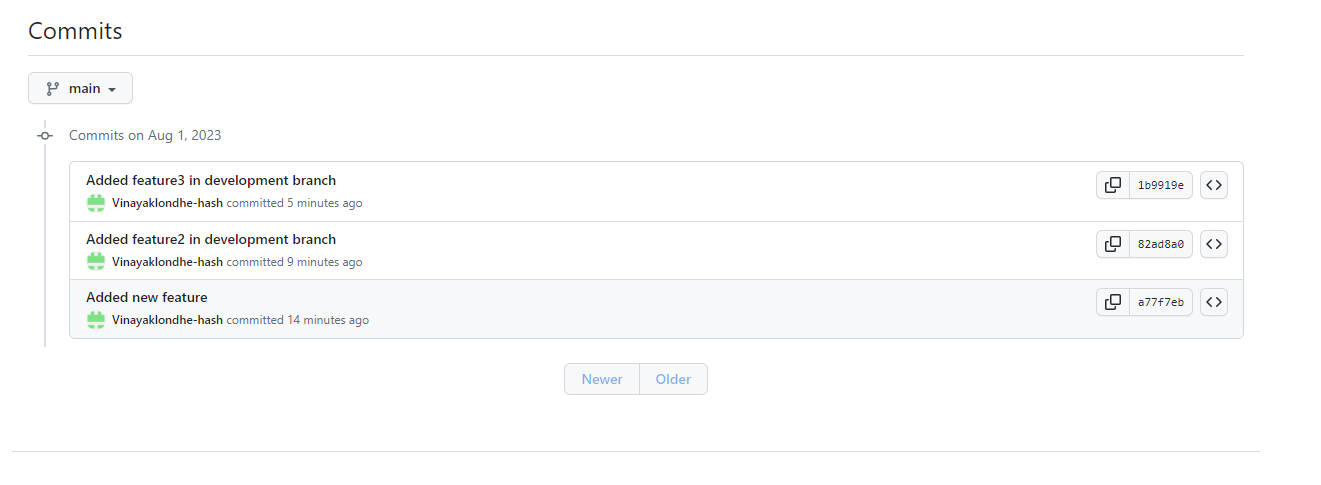
**Dev:**



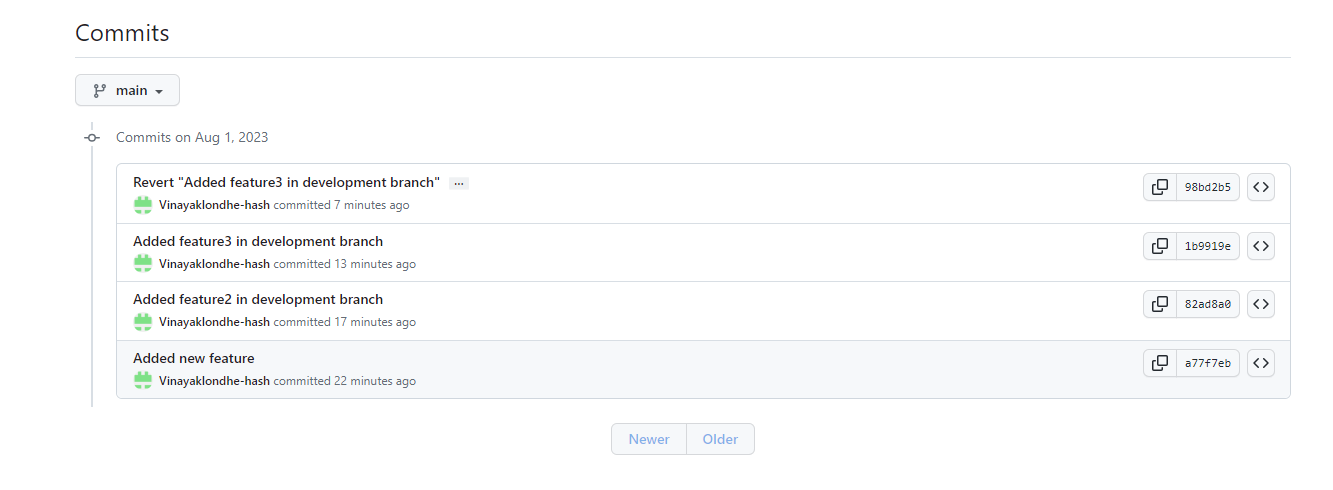
**Main**:



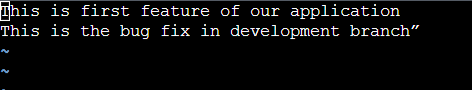
Before pushing changes to github main branch



After pushing changes ,



**Result:**



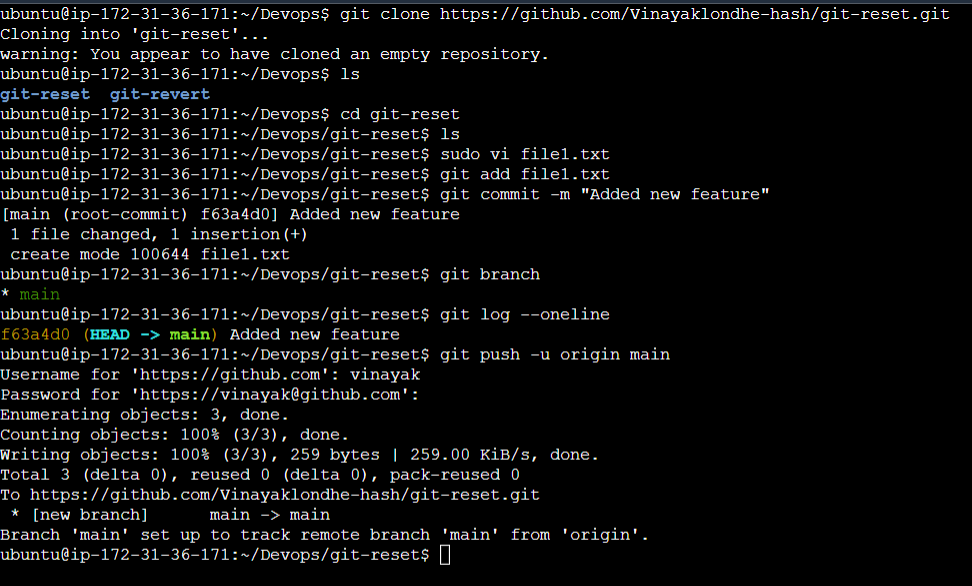
So git revert will add revert commit history.

-----------------------------------------------------------------------------------------

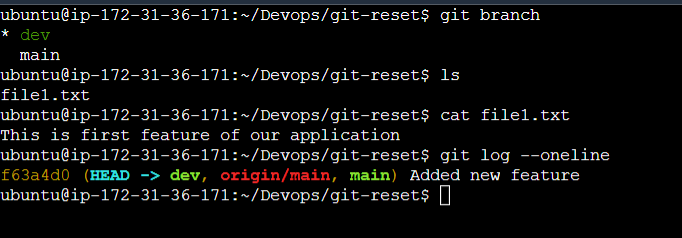
**Now lets move on to Git Reset**:

Basic use of reset is to reset /clear /delete all the commits after correct commit. So ,there will be no history of wrong commits made by us.

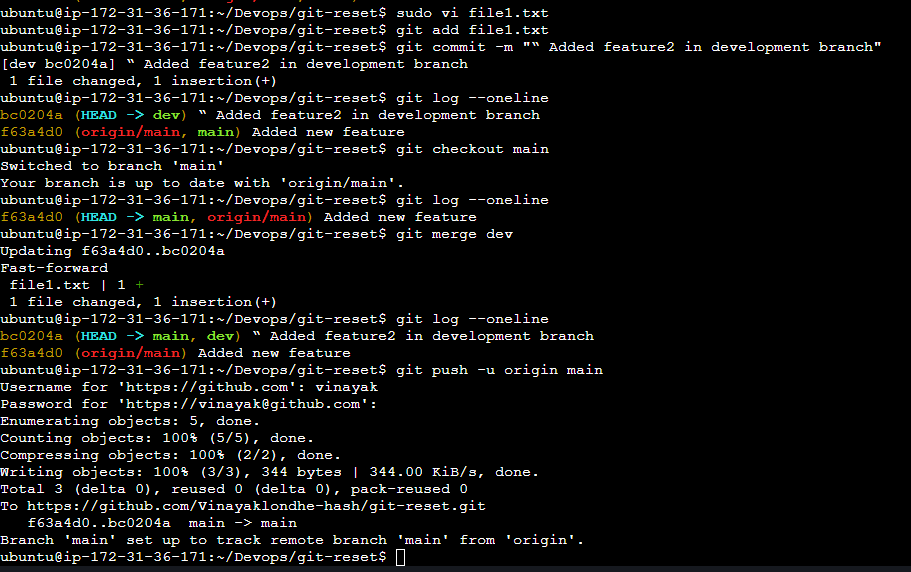
Lets create new github repo named git-reset and clone it on local repo . Lets add new file , file1.txt.



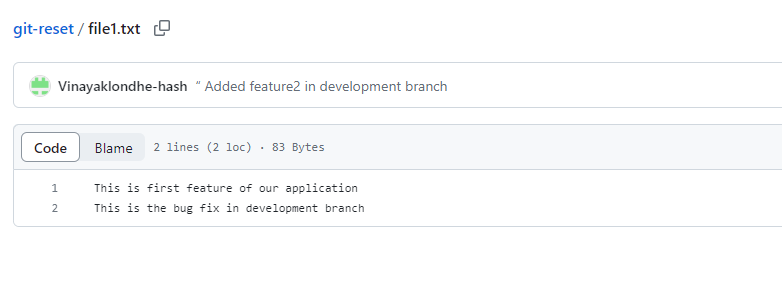
Lets checkout dev branch ,



Lets add one commit (valid commit), and merge to main & push to github

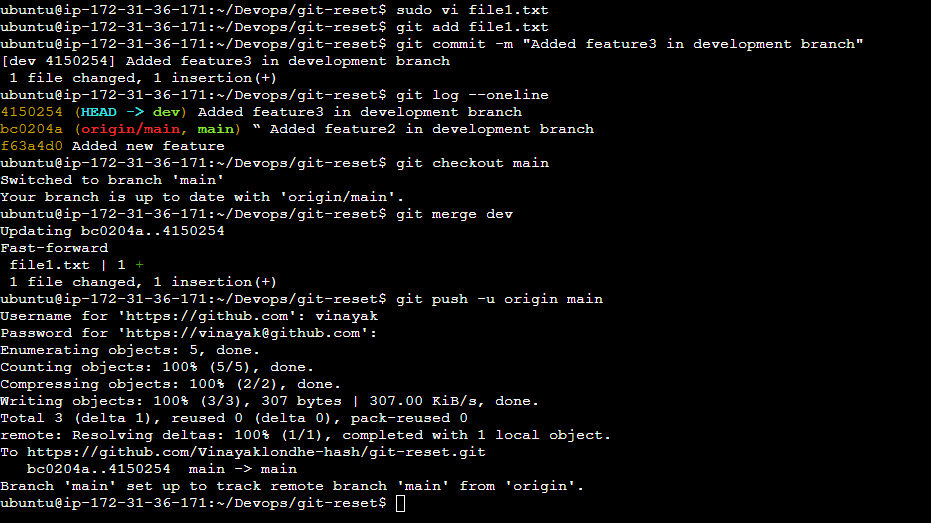


Our github repo after 2 commits,



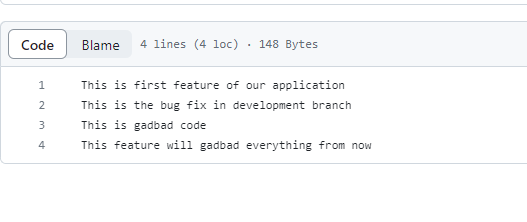
Now lets enter 2 invalid commits on dev branch & merge to main branch & to github (3rd & 4th commits will be invalid and pushed to github)

3rd commit:



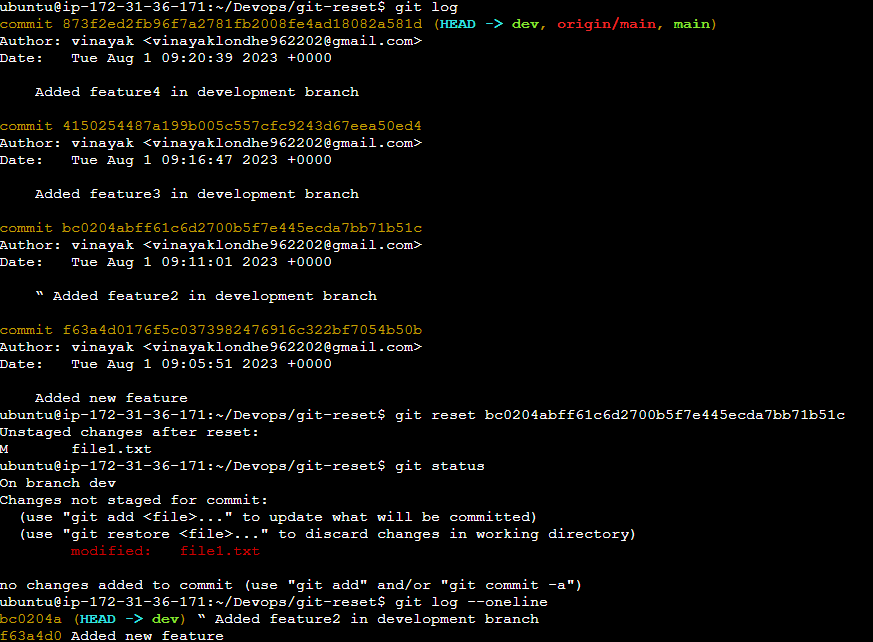
4th Commit :



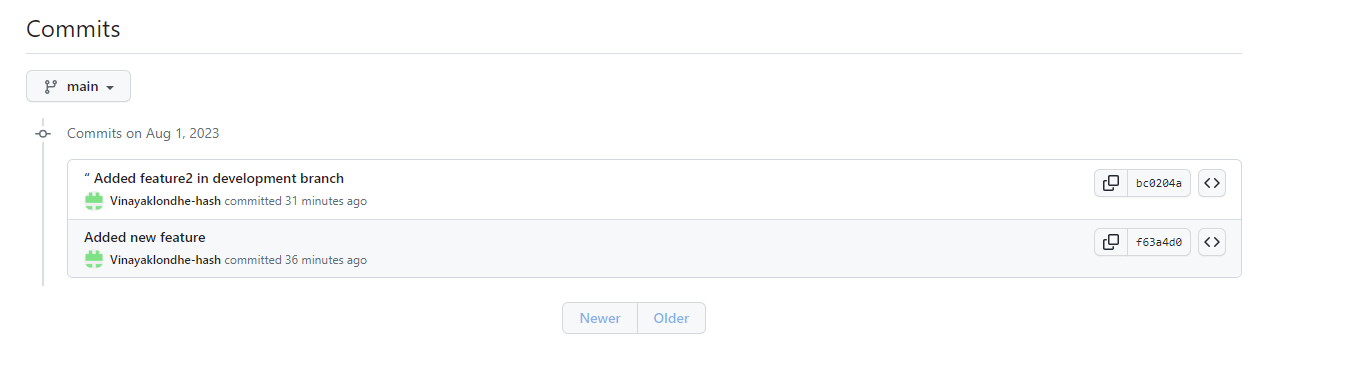


After analysis we got to know that wrong code/file has been pushed to github at 3rd commit & we need to remove it from commit history !!!

We need to use git reset <commit id of 2nd commit > , here we can see that no 3rd & 4th commit is vanished from git log on dev branch & file is modified we can add our changes & commit on dev branch .



**Similar process we need to do on master.** On master we need to do reset and then push changes to github with -ff (fast forward) and refersh your github we will not be able to see any commits after 2nd commit.



Hope That Clears your concept!!!