**Publicis Sapient PJPAssignment\_1**

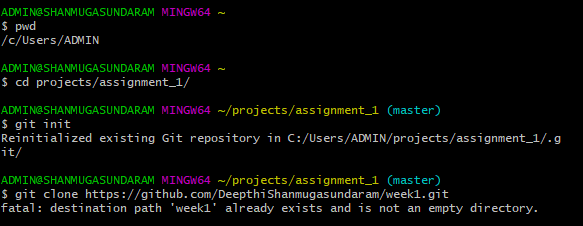
**GIT**

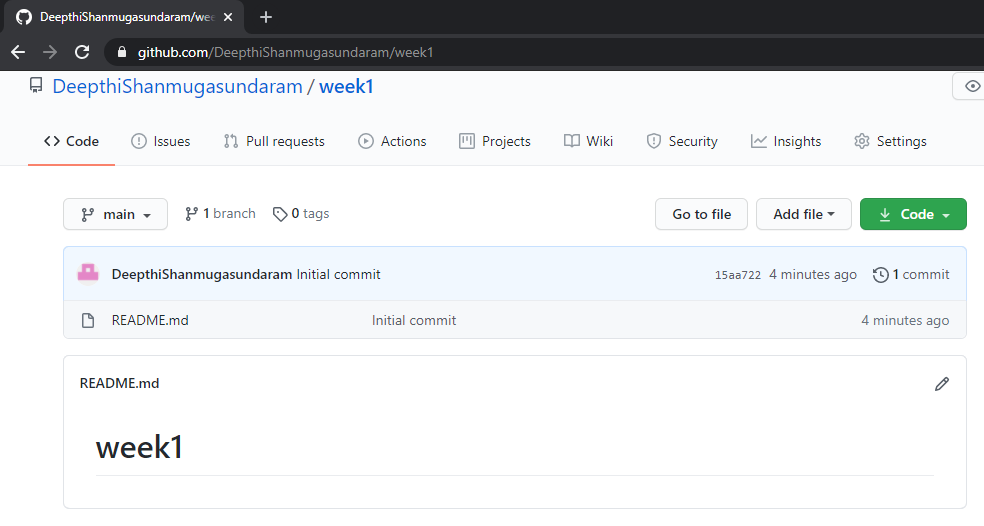
**By** Deepthi Shanmugasundaram

1.a.i. Local repository:

$git init

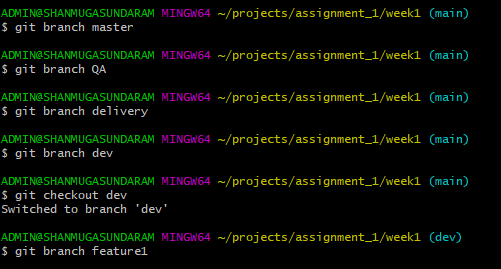
$git clone <url>

ii.Remote Repository:

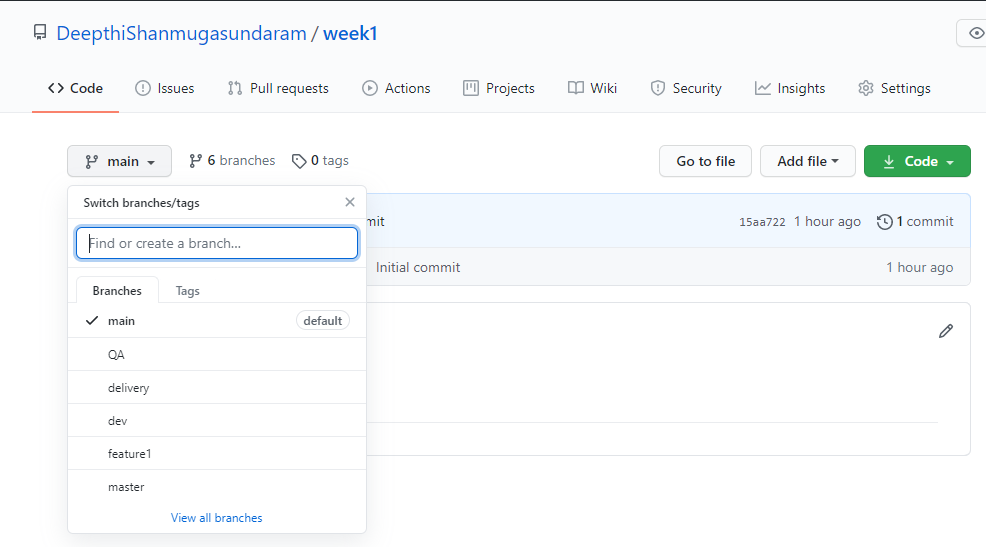


iii.local branches:

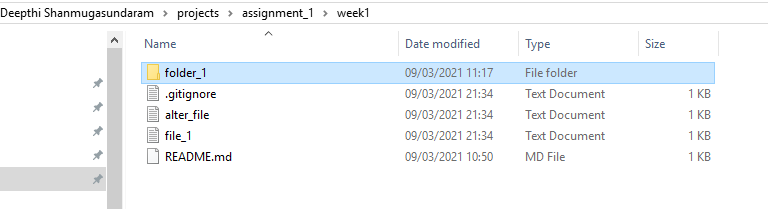
$git branch <branch name>



iv.remote branches:



v.Add files and folders:



vi.Check-in ,Stage, Commit,Push into feature1:

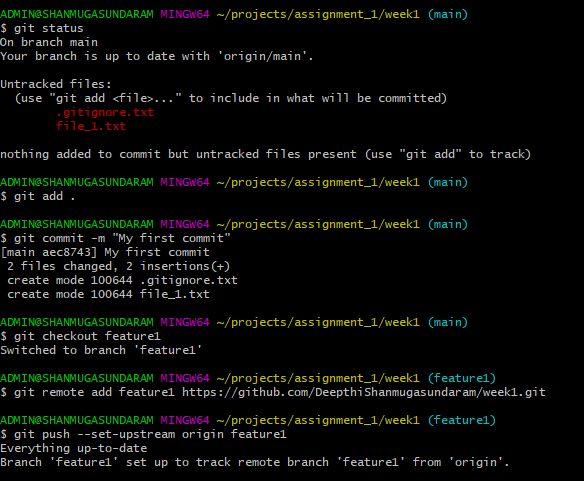
$git status

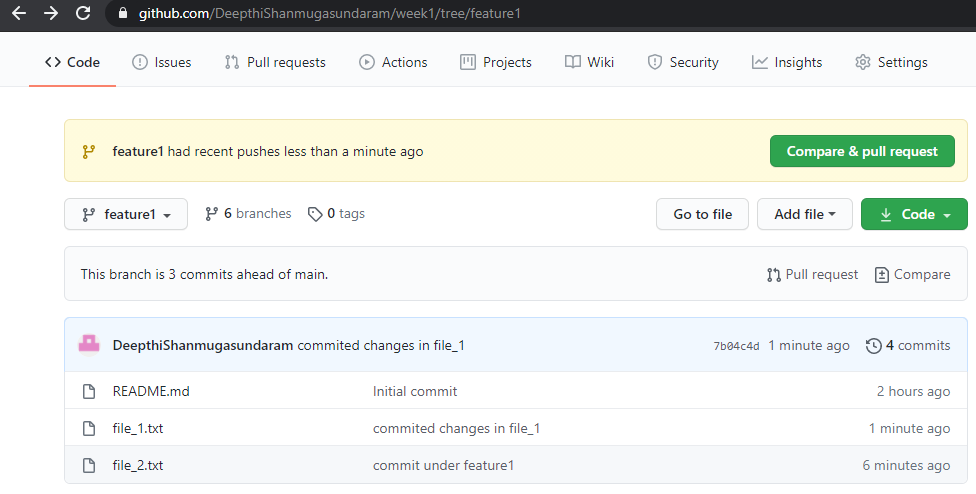
$git add

$git commit –m “message”

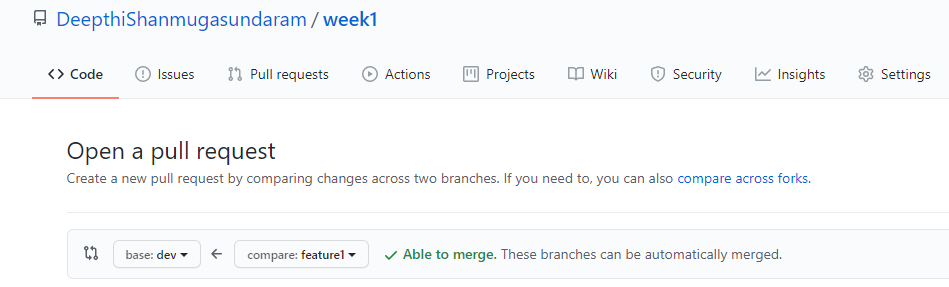
$git remote add feature1 <url>

$git push –set-upstream feature1





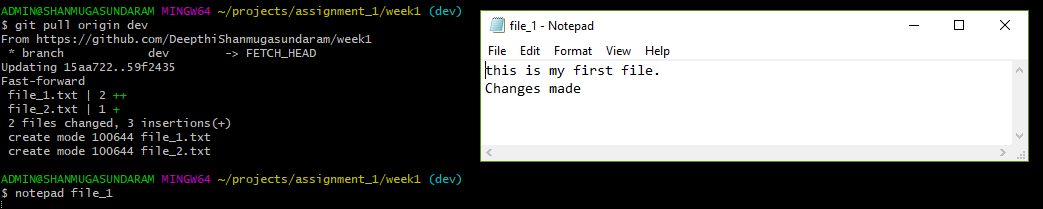
vii.Promote code from feature1 to dev:



viii.Latest code from remote to local:

Pull updated local dev branch from remote repository.

$git pull origin <branch name>



ix.Checkout: **Fetches the** latest changes into already downloaded repository. It wont merge those new changes but makes your working directory reflect them.

 Pull:Fetches and merges in local branch.

x.Handle Merge Conflict:

Same file(alter\_file) is modified at two different branches and pushed into remote repository.When we try to pull them into main branch using cli,merge conflicts occurs. Resolve this merge conflict with a new commit after changing the text under the branch in which conflict occured.

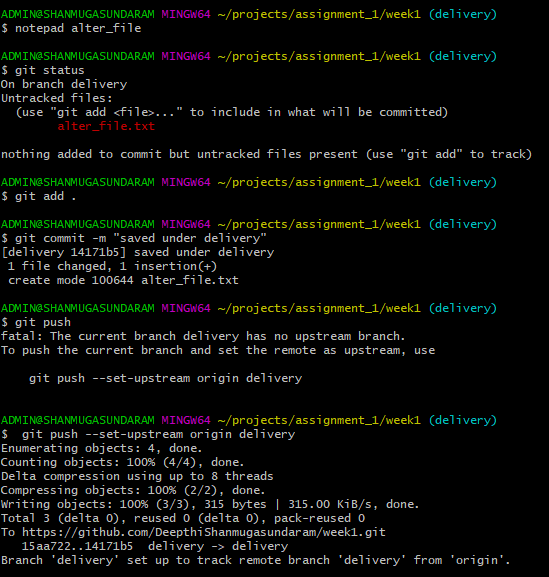
$notepad alter\_file //add a line in the file

$git status

$git add .

$git commit –m “message”

$gitpush –set-upstream origin delivery



Repeat the same steps under another branch containing the same file and add some other lines into the file and push.

Now checkout to main branch.

$git checkout main

$git merge delivery

$git merge QA

Now conflict occurs

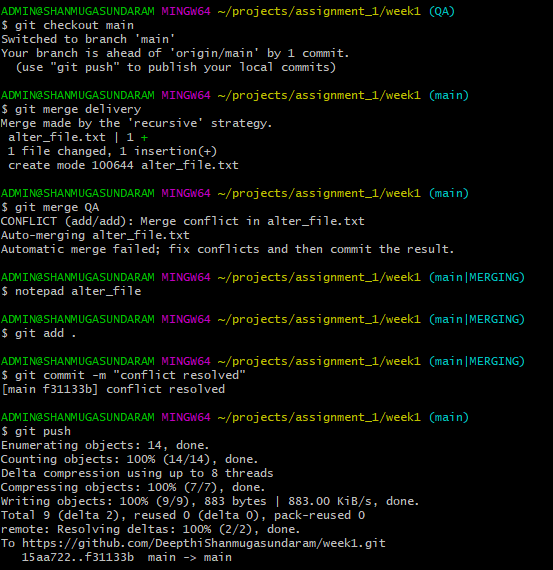
Open the file containing the conflict and resolve it

$notepad alter\_file

$git add .

$git commit –m “message”

$git push

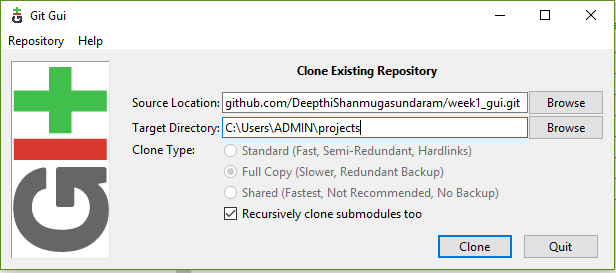


xi.Ensure the code is in sync .with latest changes across all branches:

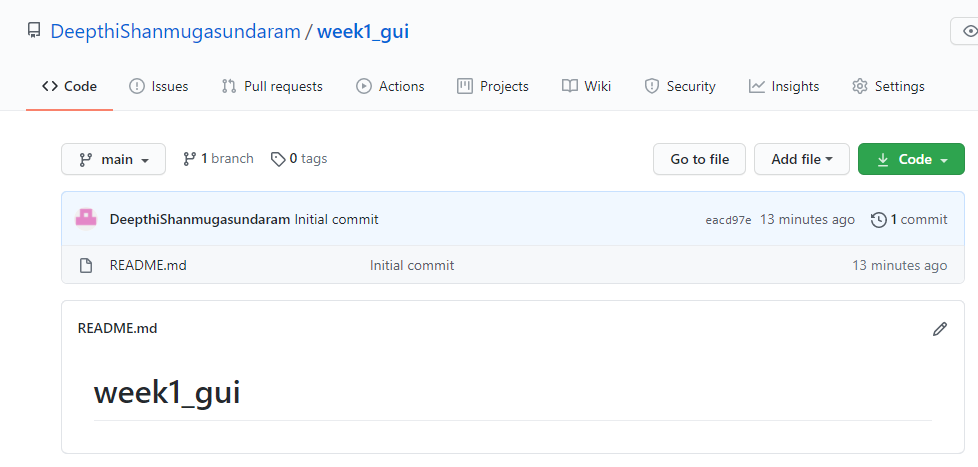
Recent changes made under local branches are pushed into remote branch and pulled to other branches.

GUI

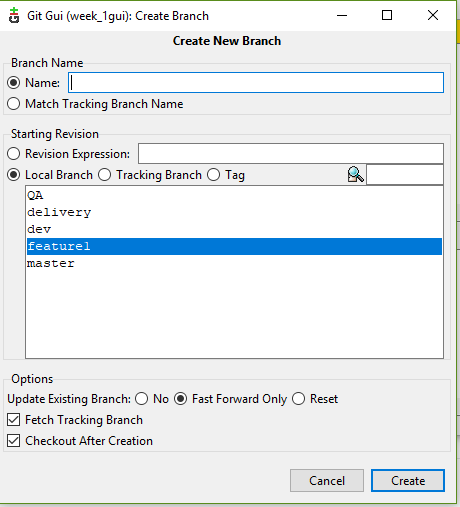
1.b i.Local repository:



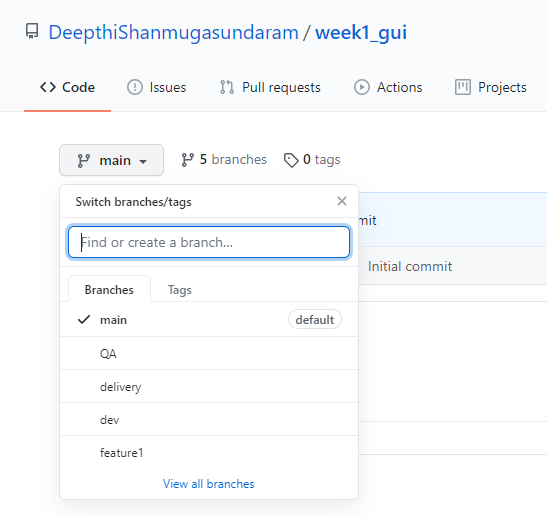
ii.Remote Repository:



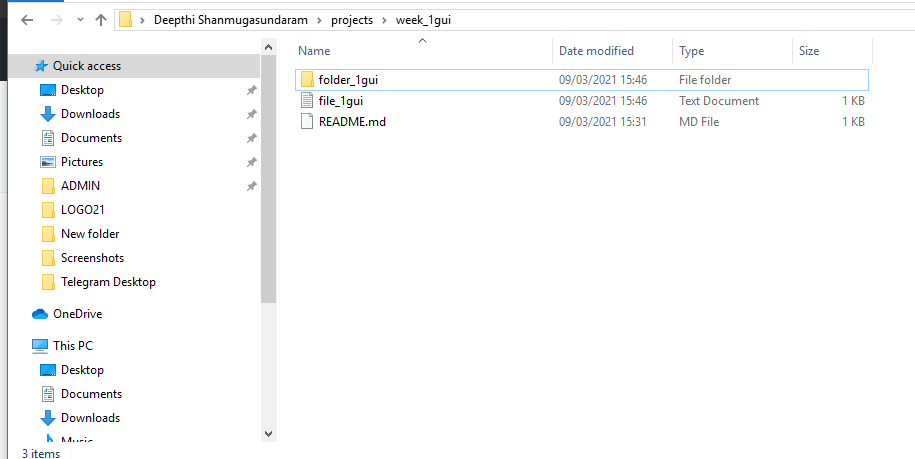
iii.local branches:



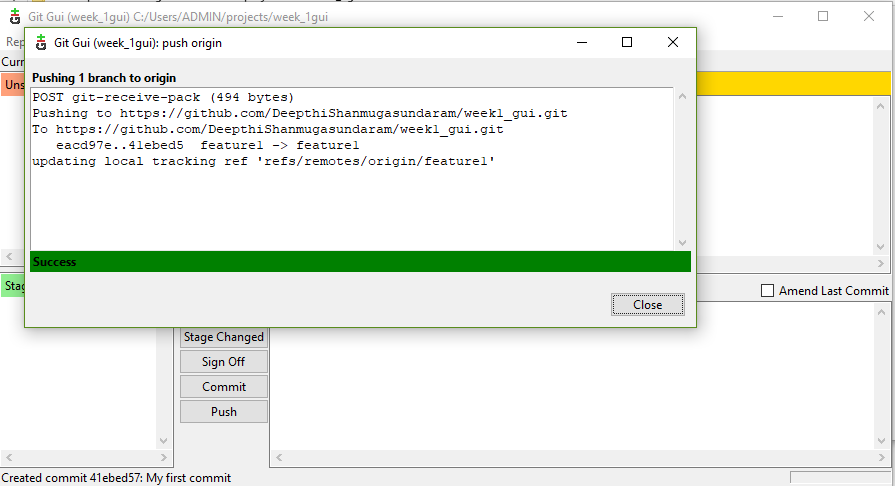
iv.remote branches:



v.add files and folders:

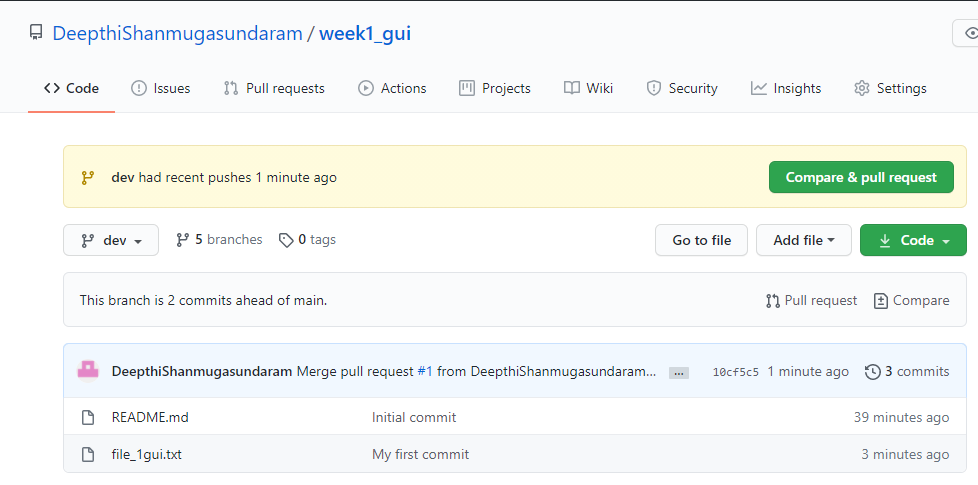


vi.Check-in ,Stage, Commit,Push into feature:



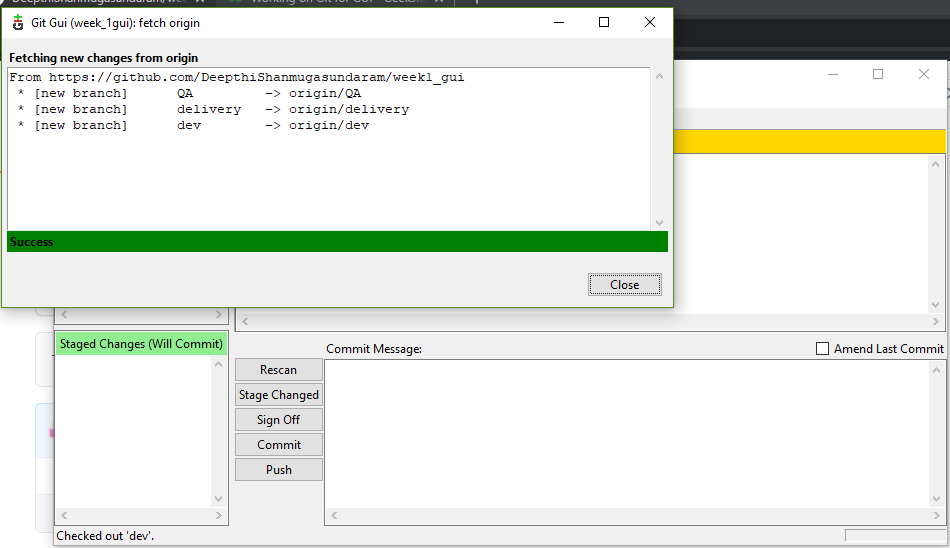
vii.Promote code from feature1 to dev:

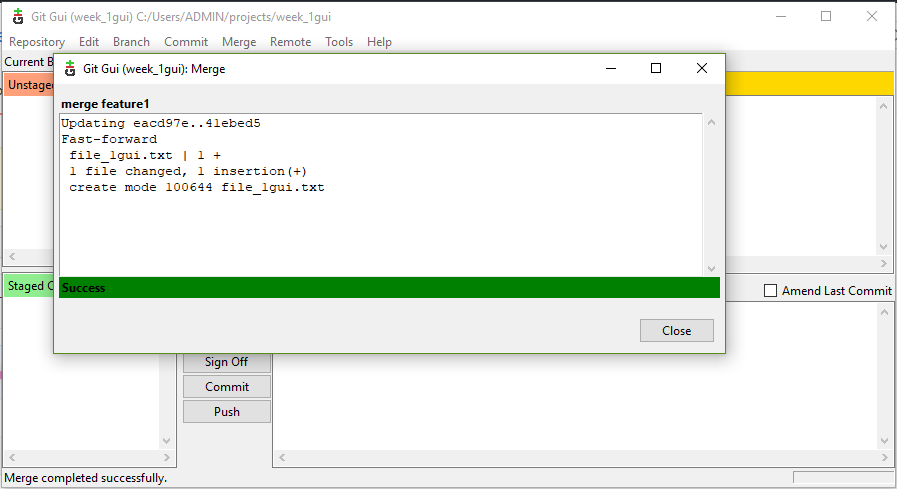
Pull request from feature1 to dev is done.



viii.Latest code from remote to local:

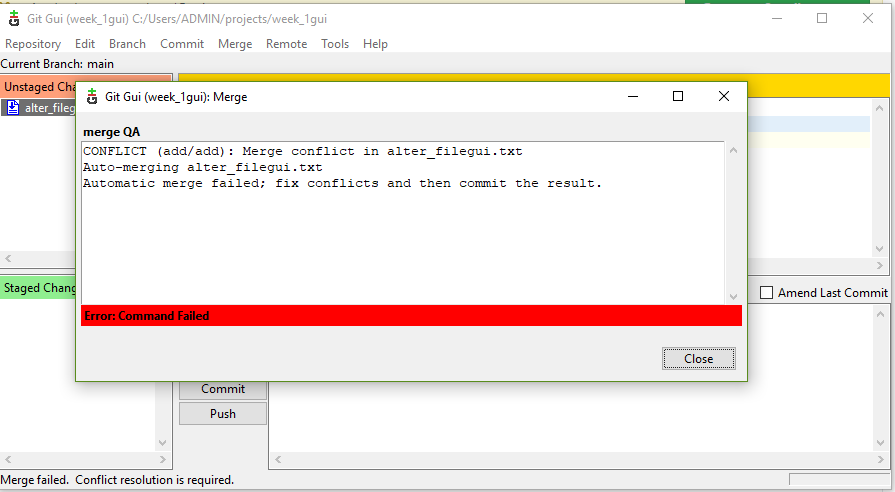
Fetch and merge into local repository



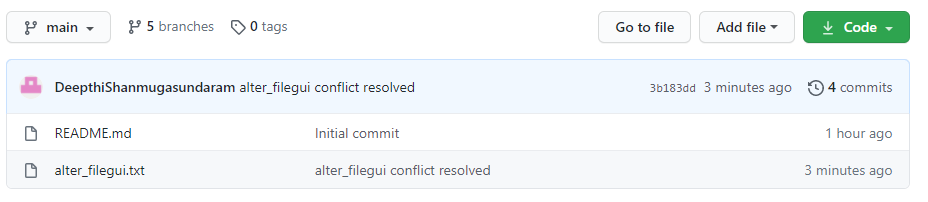


x.Handle Merge Conflict:

Merge conflict occurs when alter\_filegui is merged into main branch as they have been changed under delivery and QA branches .

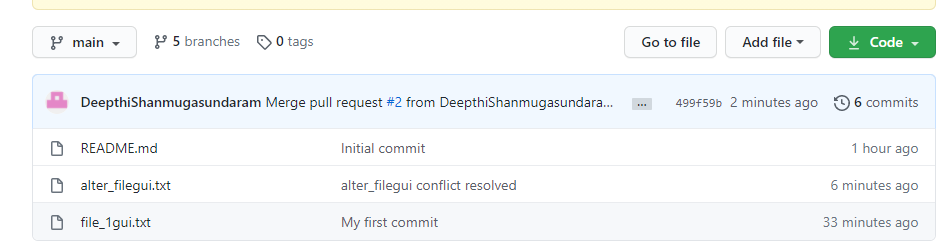


This is resolved by doing a new commit with required changes being done in the conflict file.



xi.Ensure the code is in sync with latest changes across all branches:

Recent changes are pushed into remote branch and pulled to other branches from there.



c.i.Reset or reverting the changes

$git log

$git reset HEAD~1

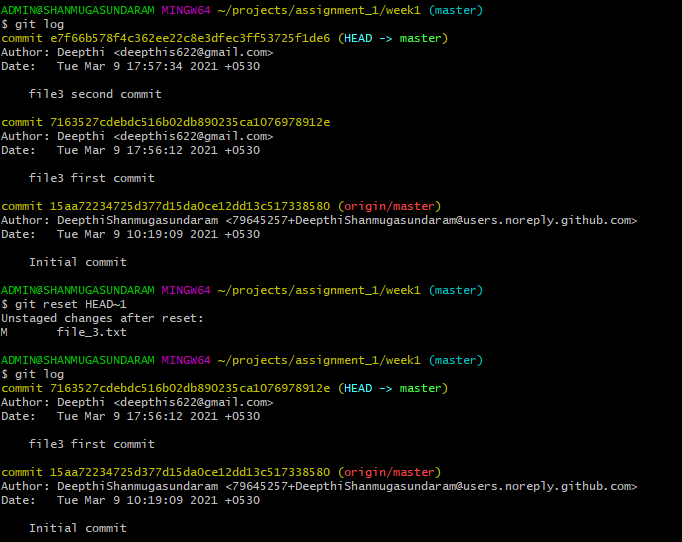
$git reset –soft HEAD~1 //brings commited changes to staging area,so add and commit again.

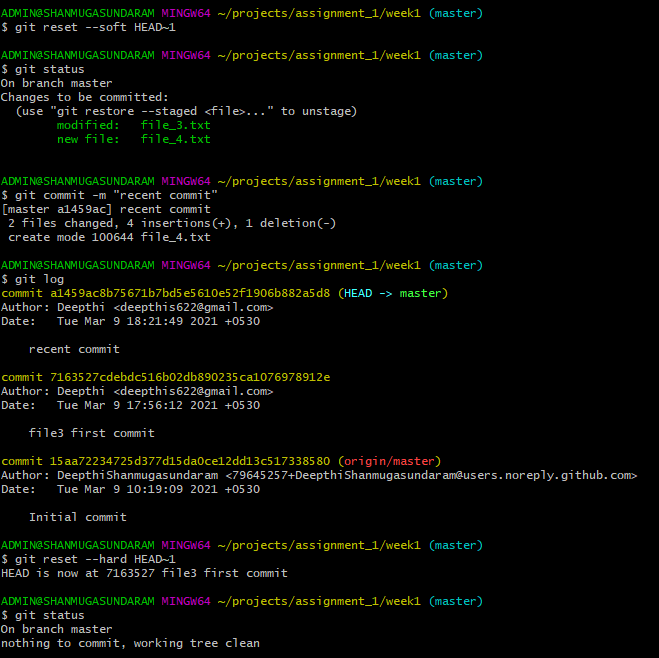
$git status

$git add .

$git commit –m “message”

$git reset –hard HEAD~1





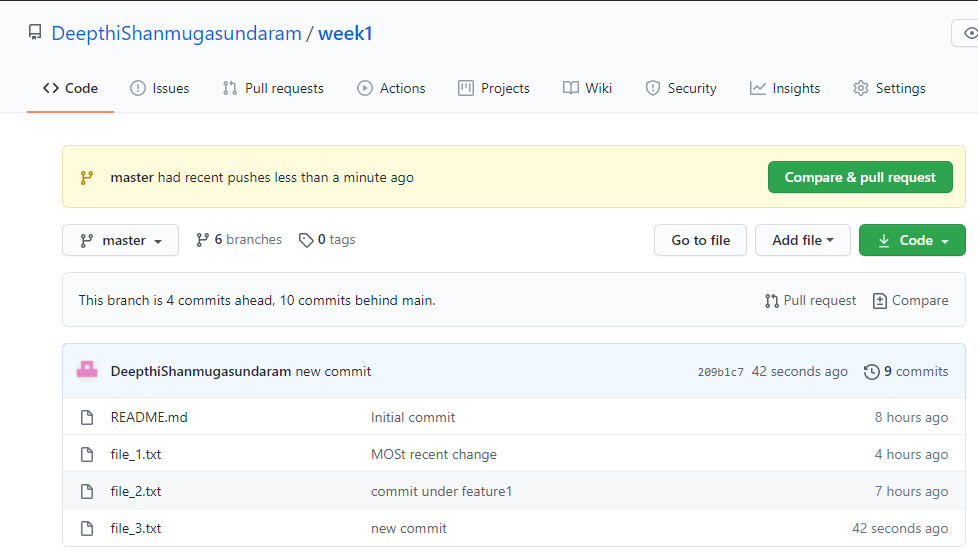
After pushing commits into remote branch,if we have to undo commits,use revert.

$git log

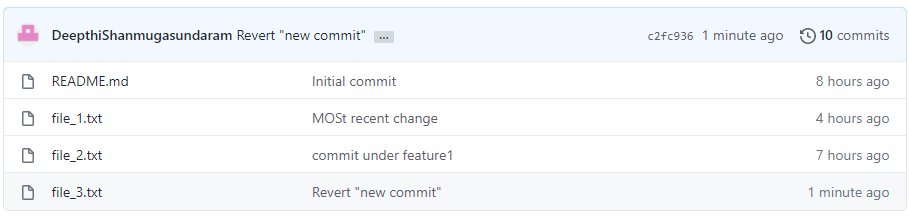
$git pull origin master

$git revert <commit ID>

$git push



After reverting,file\_3



ii.Stash local changes

make some changes into the file

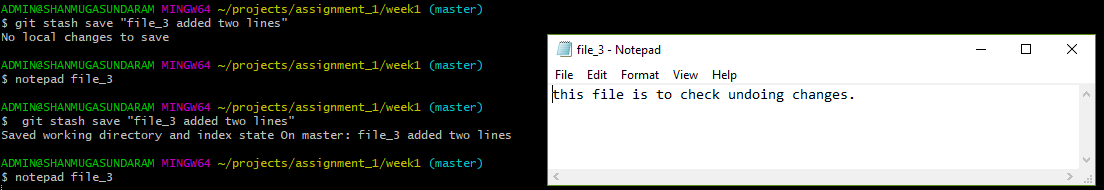
$notepad file\_3

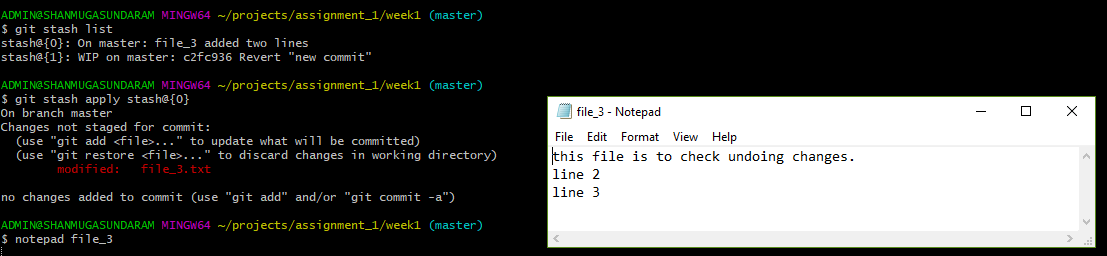
$git diff

$git stash save “message”

$git stash list

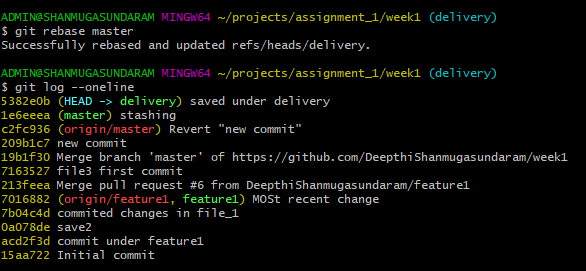
$git stash apply <stash ID>





iii.Rebasing

$git rebase <branch name>



iv.git log,status & reflog:

$git log //shows all commits iin the current branch

$git status //shows the contents under the branch and whether it is in staging area or changes made and not stages and if it is up to date.

$git reflog

