**PS PJP Week1 – Assignment**

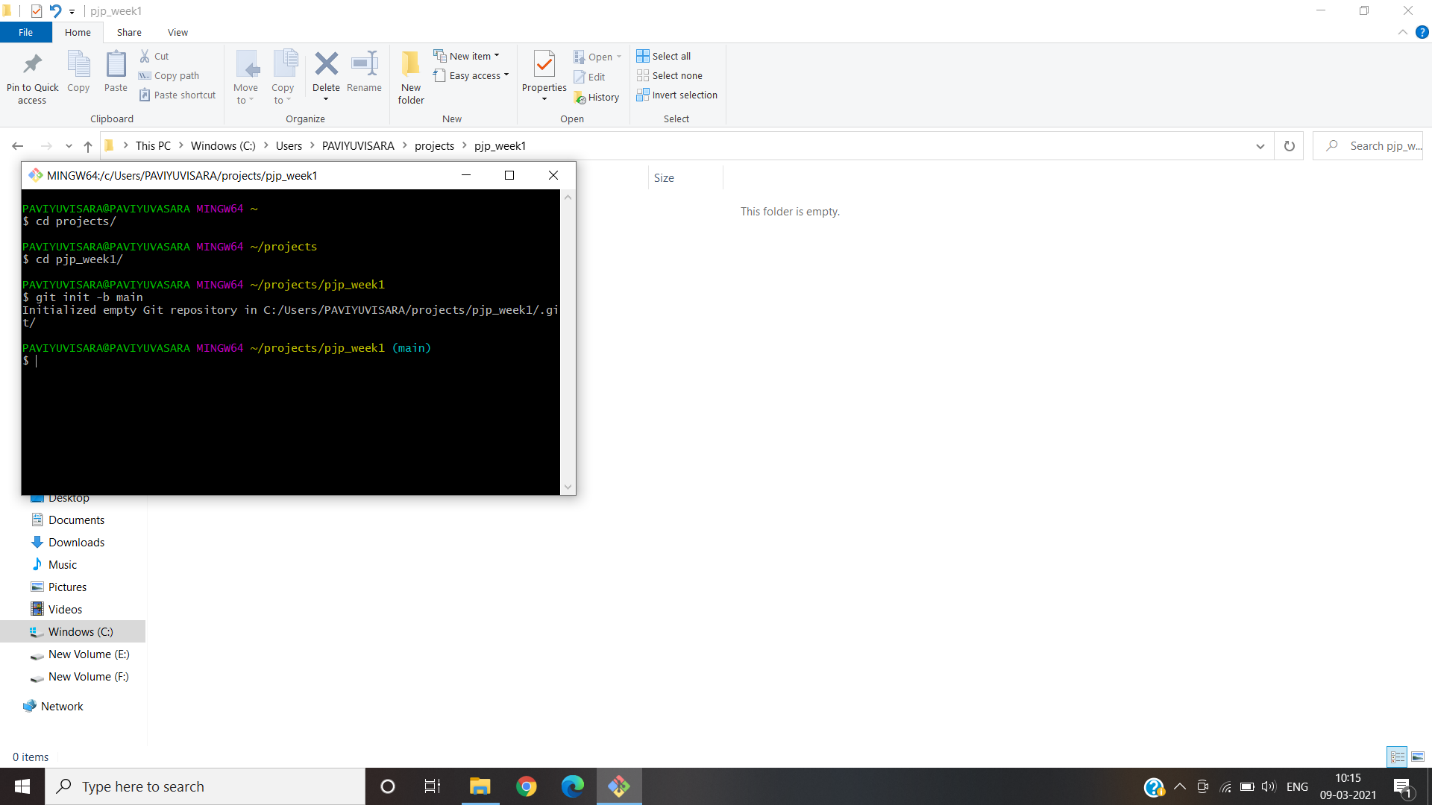
-Saranya.S.B,

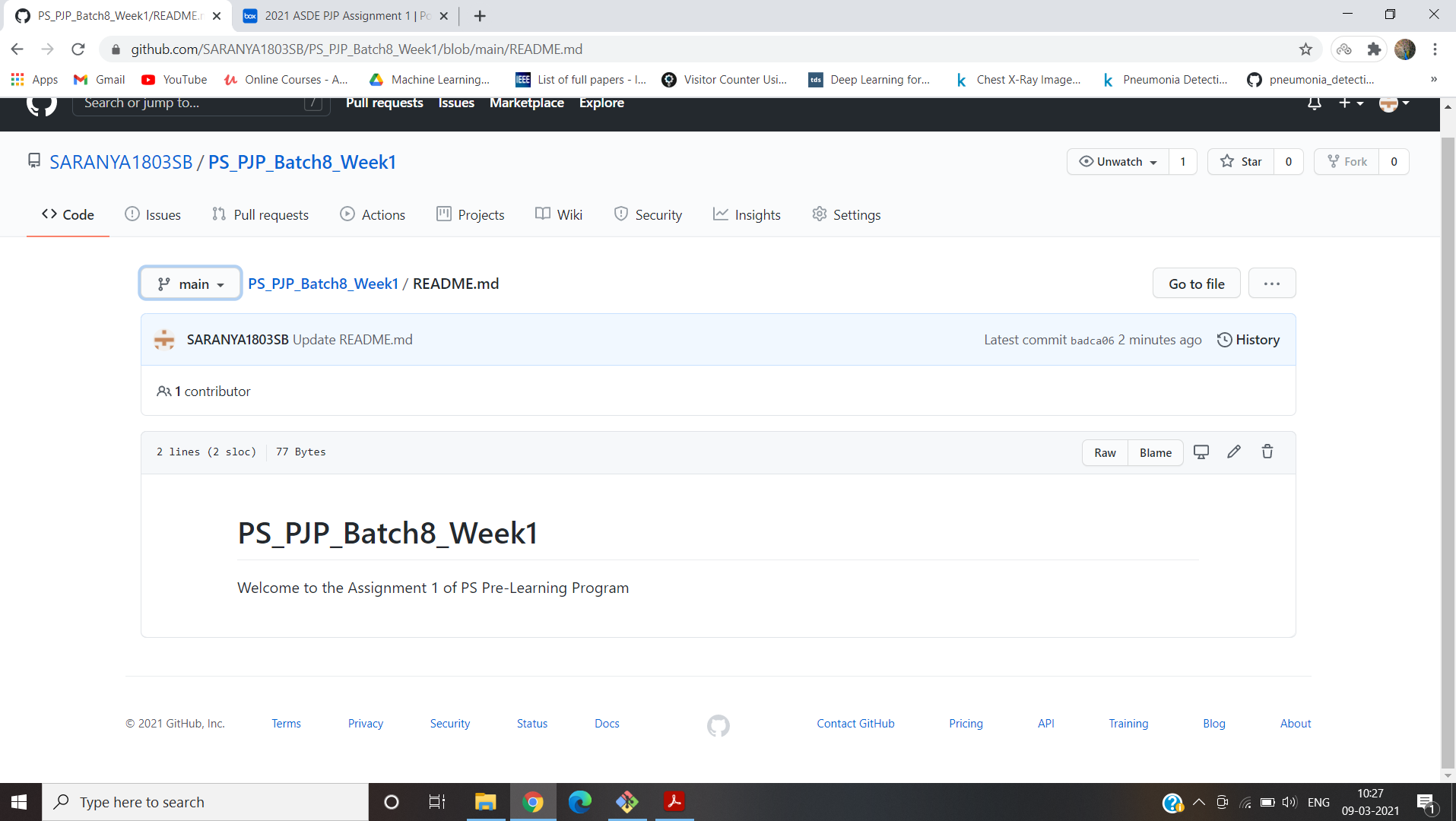
Anna University

**1.Github - a. Basic usage using the CLI**

i. Setup a Local repository:

$ git init -b main



ii. Setup a Remote Repository: 

iii. Create local branches:

$ git clone <https://github.com/SARANYA1803SB/PS_PJP_B8_week1.git>

$ git branch develop

$ git branch master

$ git branch QA

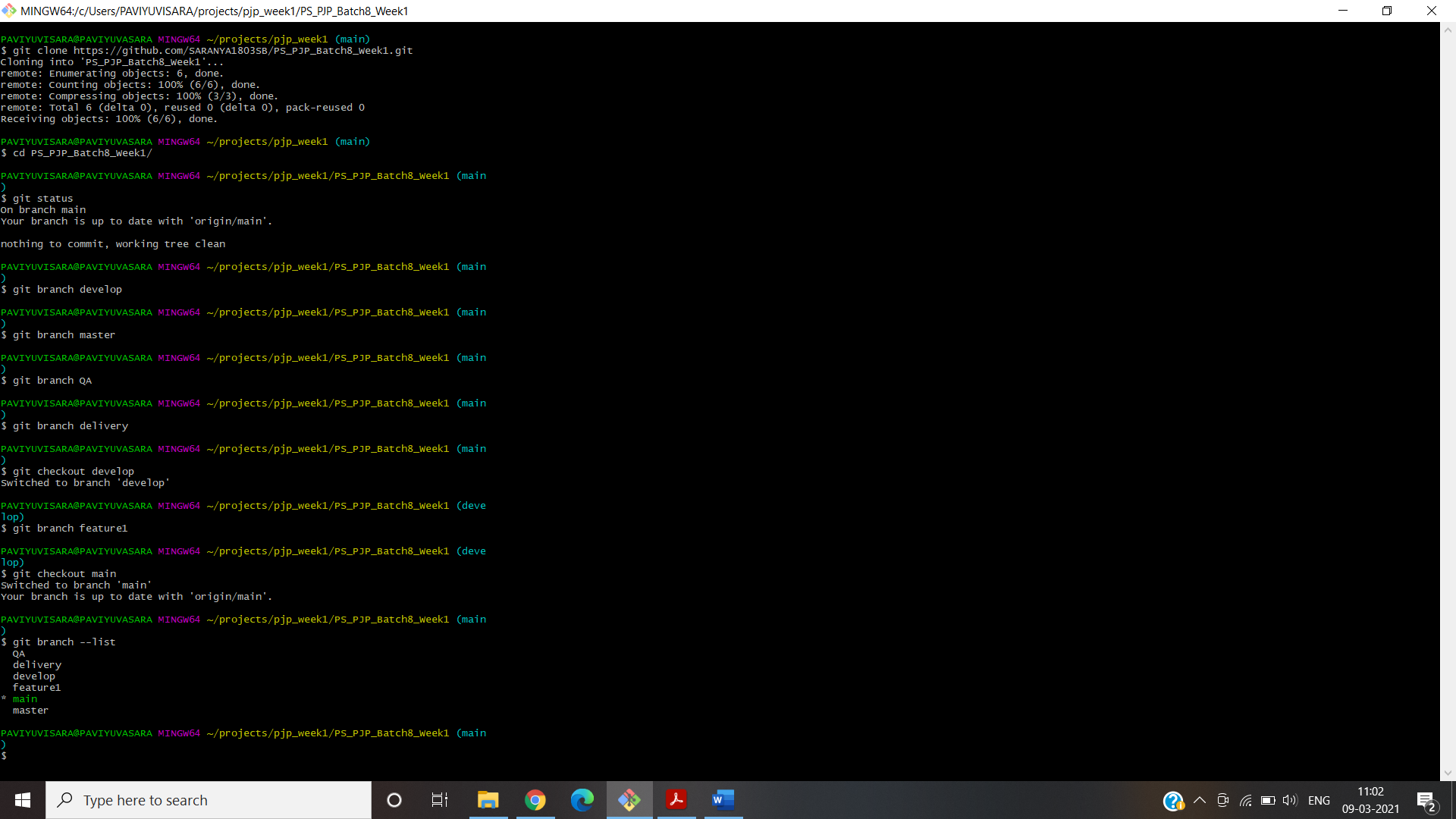
$ git branch delivery

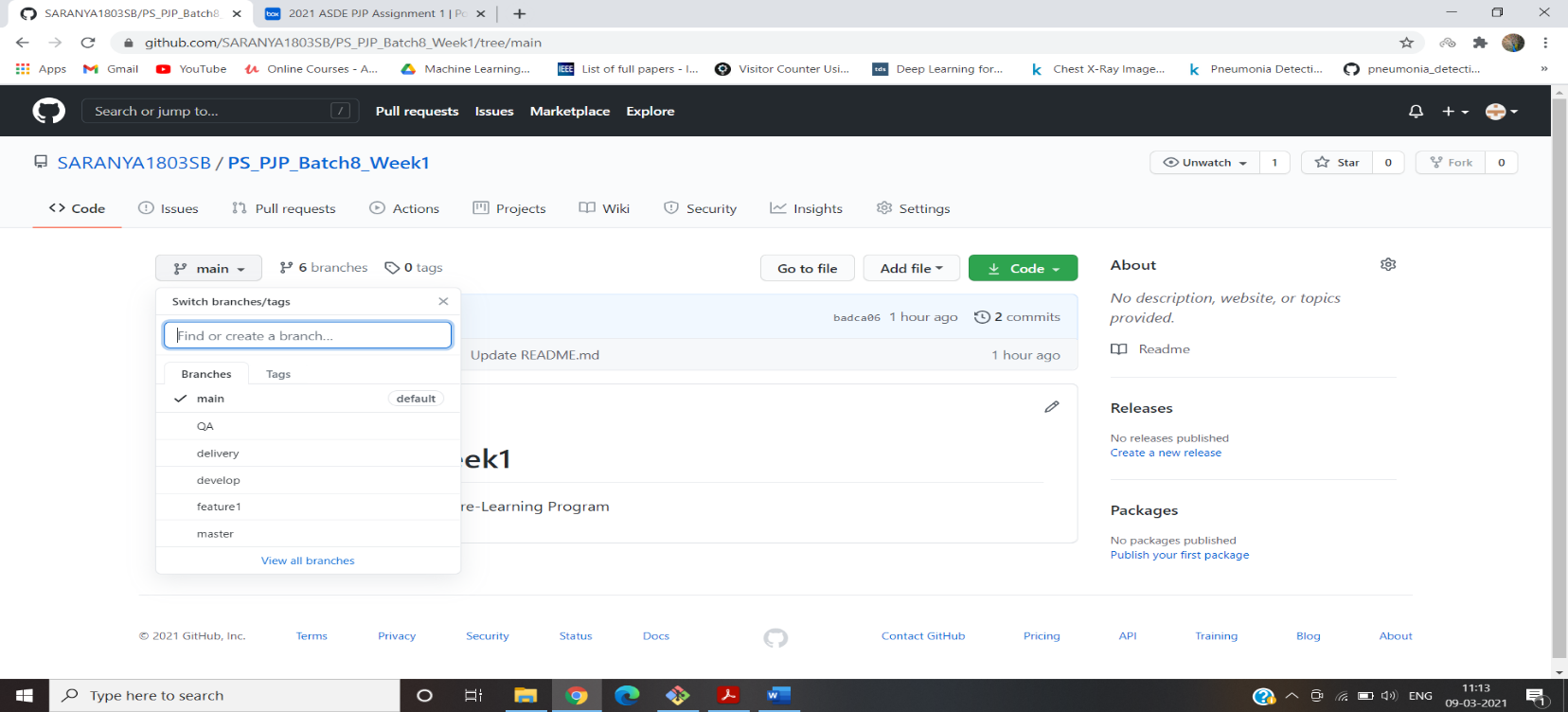
$ git checkout develop

$ git branch feature1

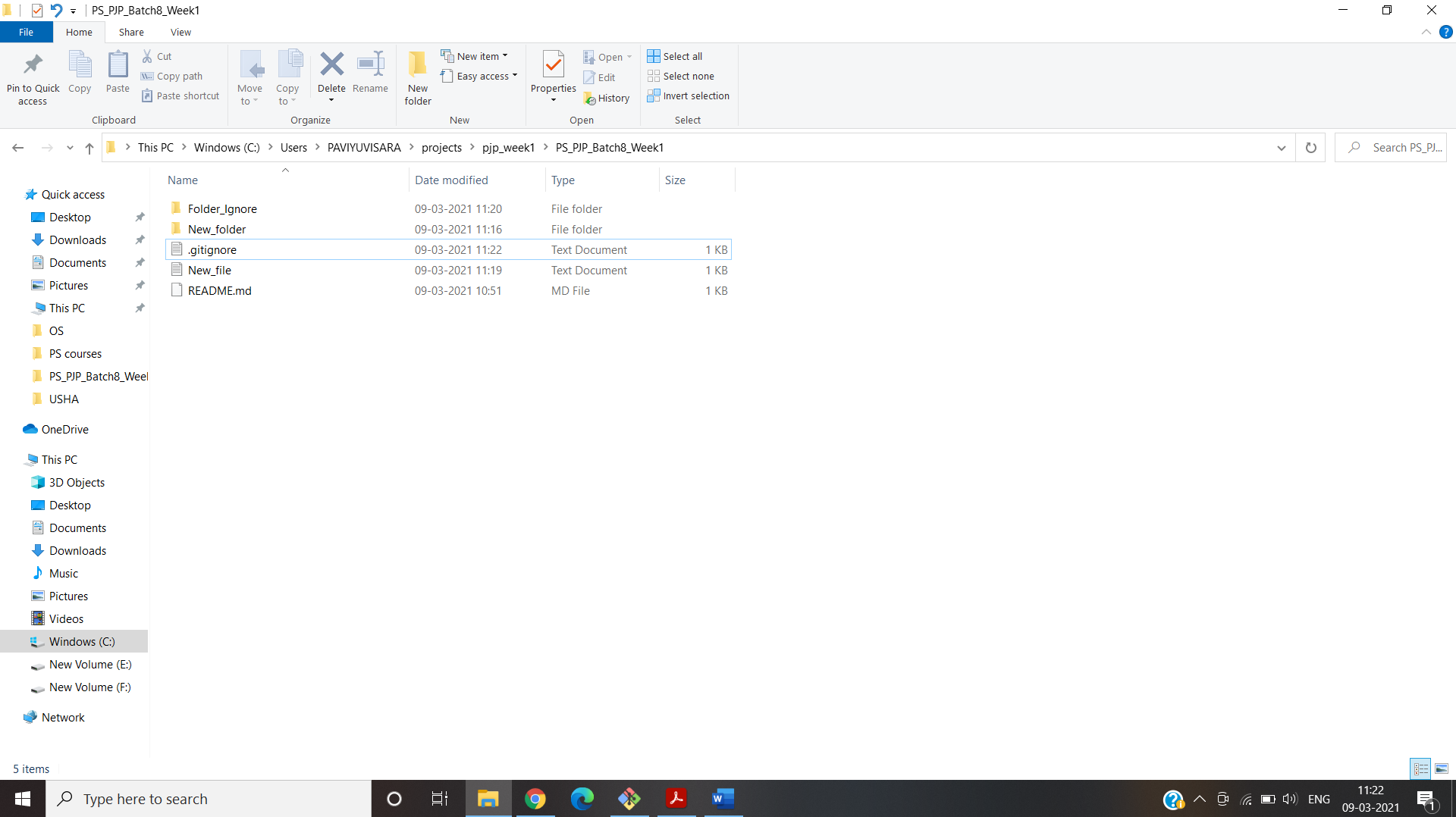
$ git checkout main

$ git branch –list ---------------------------🡪to create the branches created



iv.Create remote branches: 

v. Adding files and folders:



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

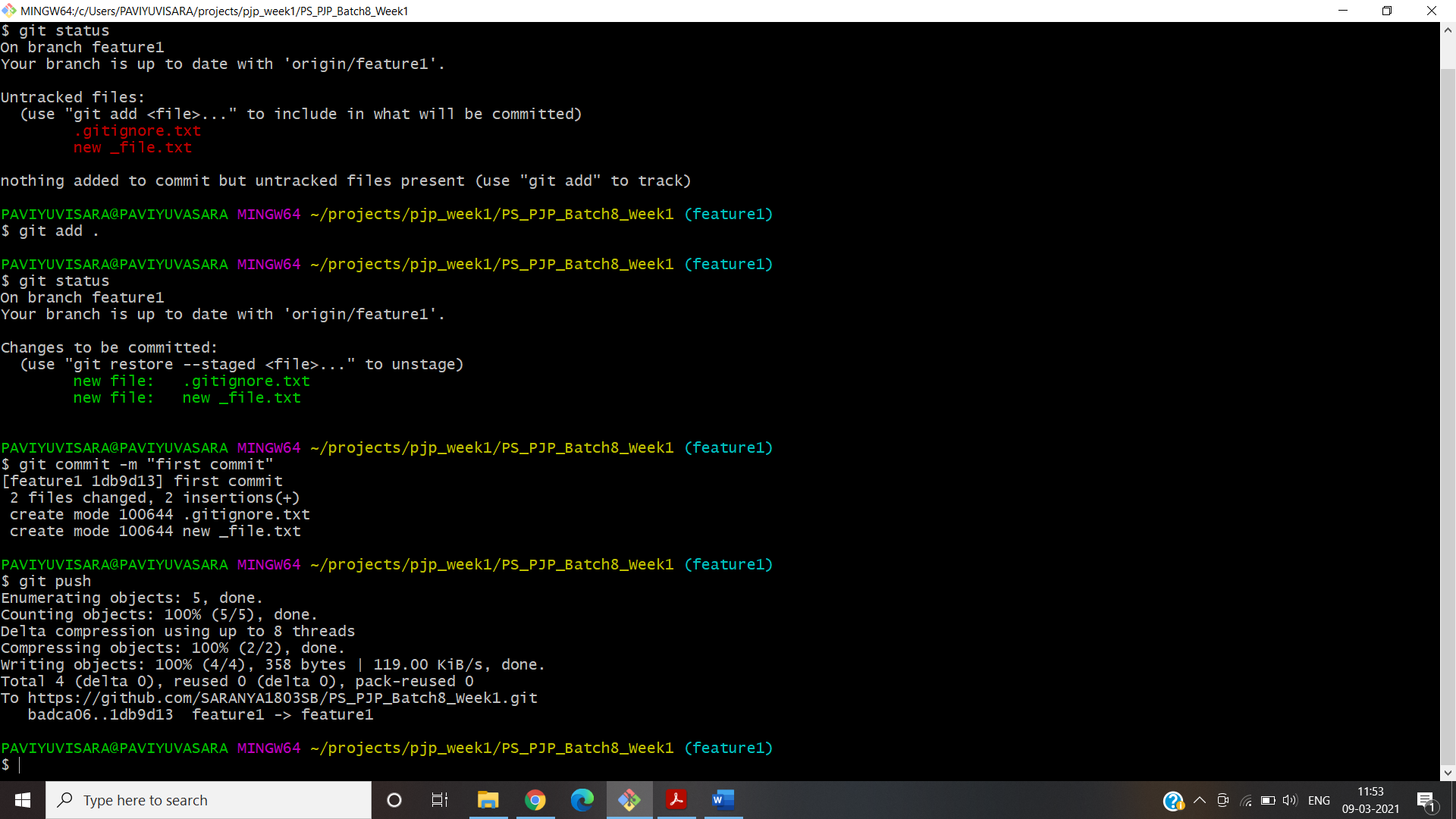
$ git status

$ git add .

$ git status

$ git commit -m “first commit”

$ git push



vii. Promote code from feature to develop:

Click “Compare & Pull request”:



Click “Create pull request”:

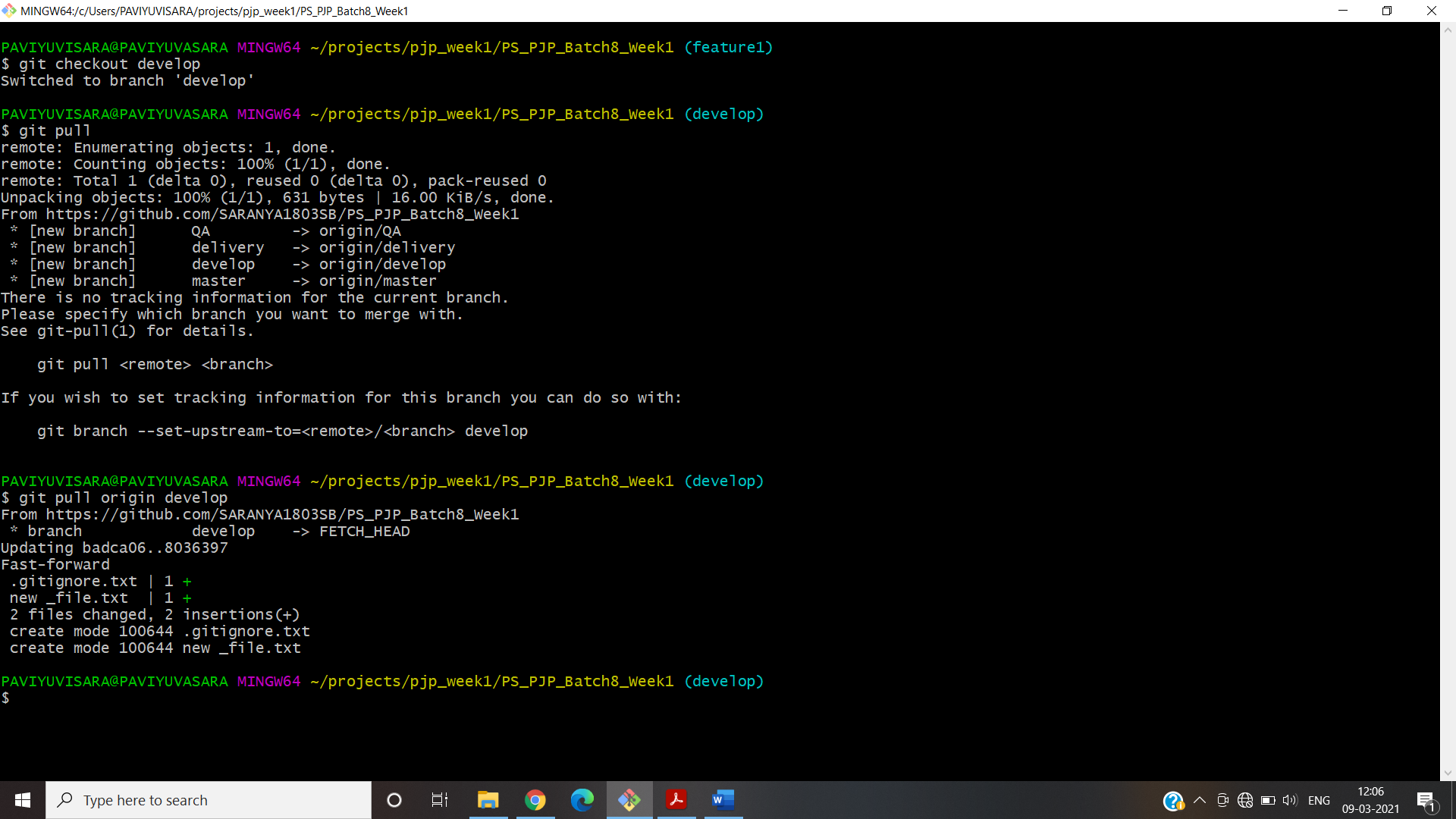


viii. Latest code from remote to local:

In remote repository , We created a pull request from feature to develop but it is not updated in local repository.So we update in local repository using pull command line:

$git checkout develop

$ git pull origin develop



ix.Checkout Vs Pull:

Checkout: It checks for the update in a branch but does not merge the updates.

Pull: It check for the updates and merges in local branch.

x. Handle Merge Conflict:

Merge conflict is created by adding same line in two branches in Branch QA and branch delivery

* In delivery branch , created a text document “merge\_file”.
* Then I created a commit in this branch and pushed to remote repository.
* In QA branch , created tha same a text document “merge\_file” with different content.
* Then I created a commit in this branch and pushed to remote repository.
* I merged delivery branch in main branch.
* After that when I try to merge qa branch to main “It shows merge conflict error”.

$ git checkout delivery

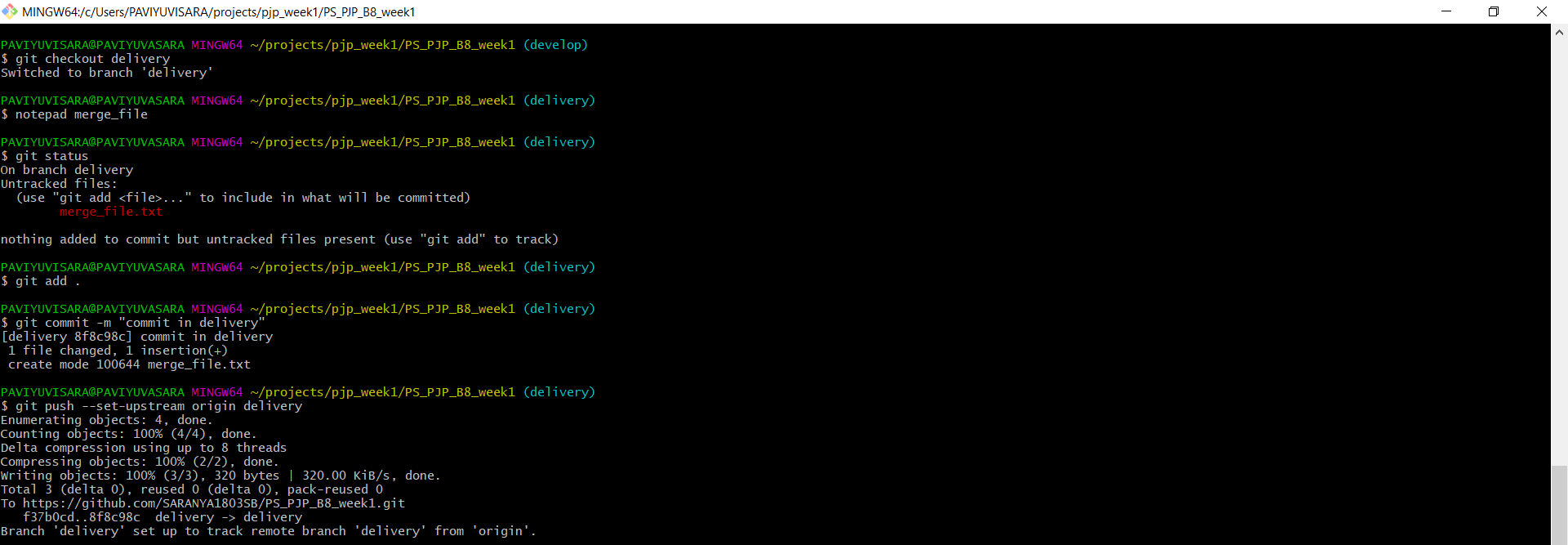
$ notepad merge\_file

$ git status

$ git add .

$ git commit -m “commit in delivey”

$ git push –set-upstream origin delivery



$ git checkout QA

$ notepad merge\_file

$ git status

$ git add .

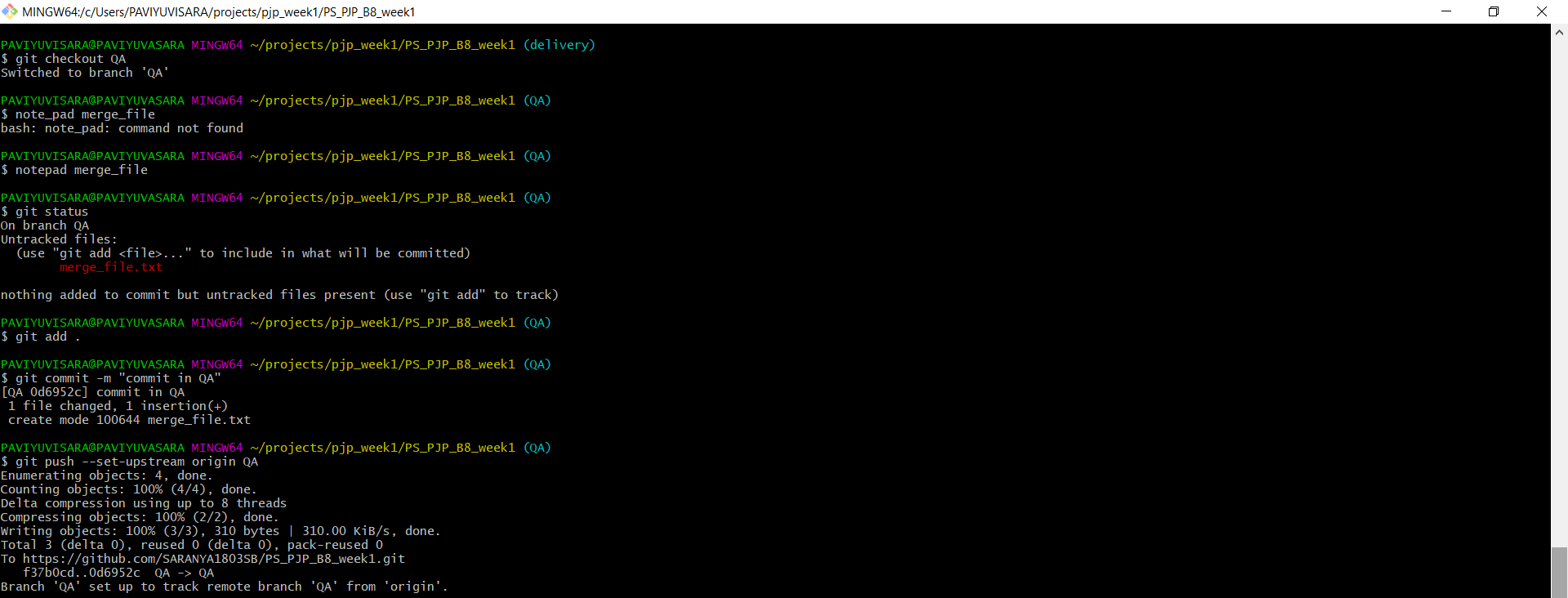
$ git commit -m “commit in QA”

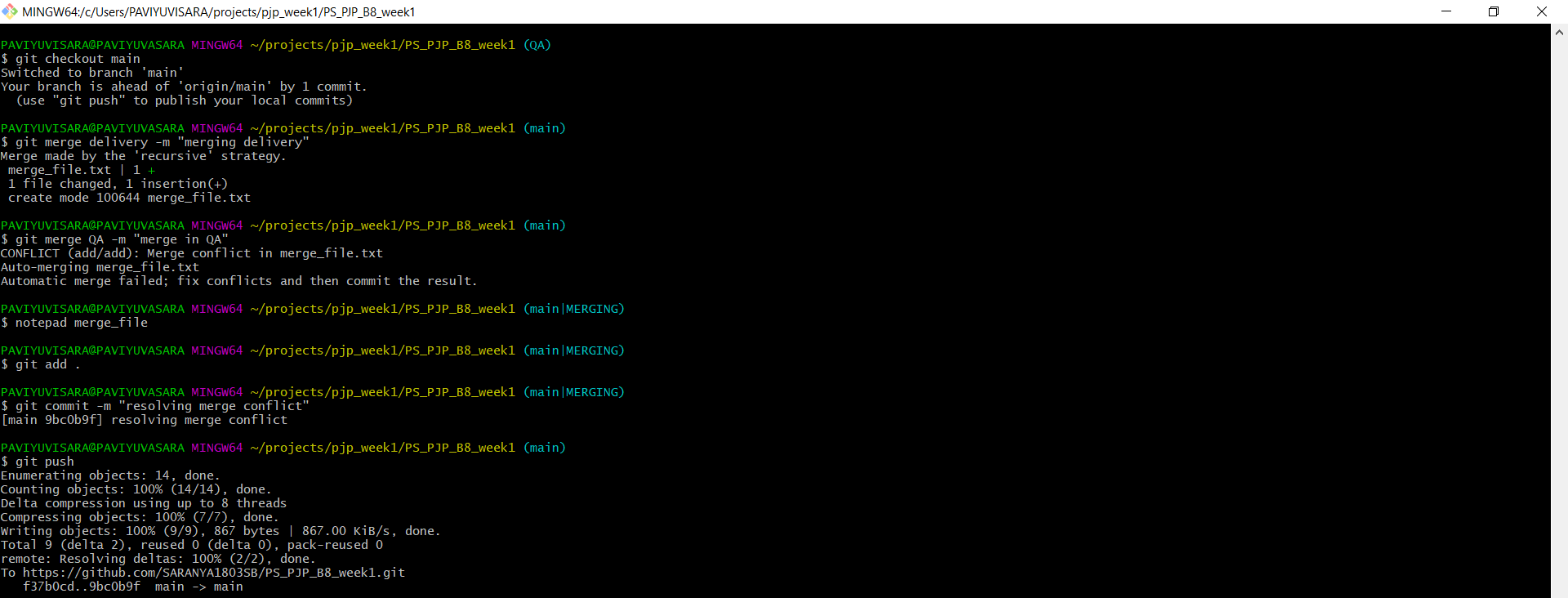
$ git push –set-upstream origin QA

$ git checkout main

$ git merge delivery -m “merging delivery”

$ git merge QA -m “merging QA”





Handling merge Conflicts:

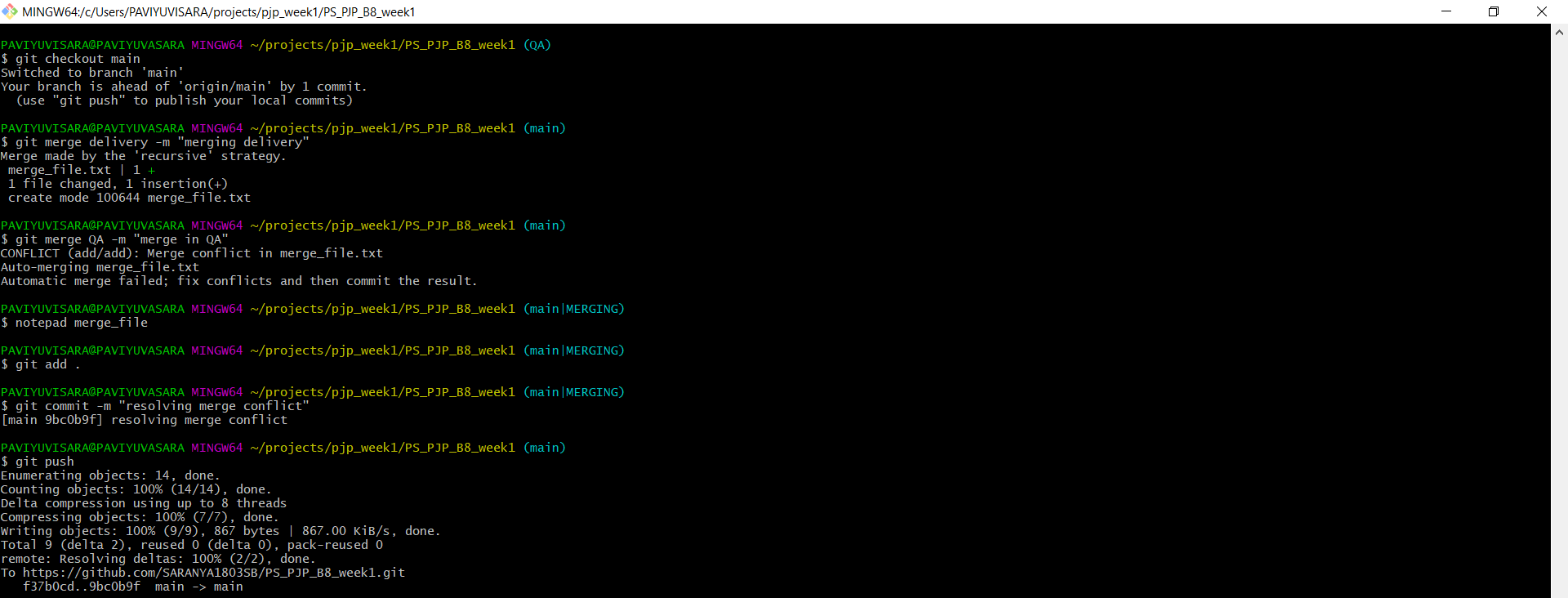
* I changed the content of the file “merge\_file” in main branch.
* Then added and commited and finally pushed the same file to remote.

$ notepad merge\_file

$ git add .

$ git commit -m “resolving merge conflict”

$ git push



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

Made some changes in the text file in feature1 branch and it is commited & pushed to remote.

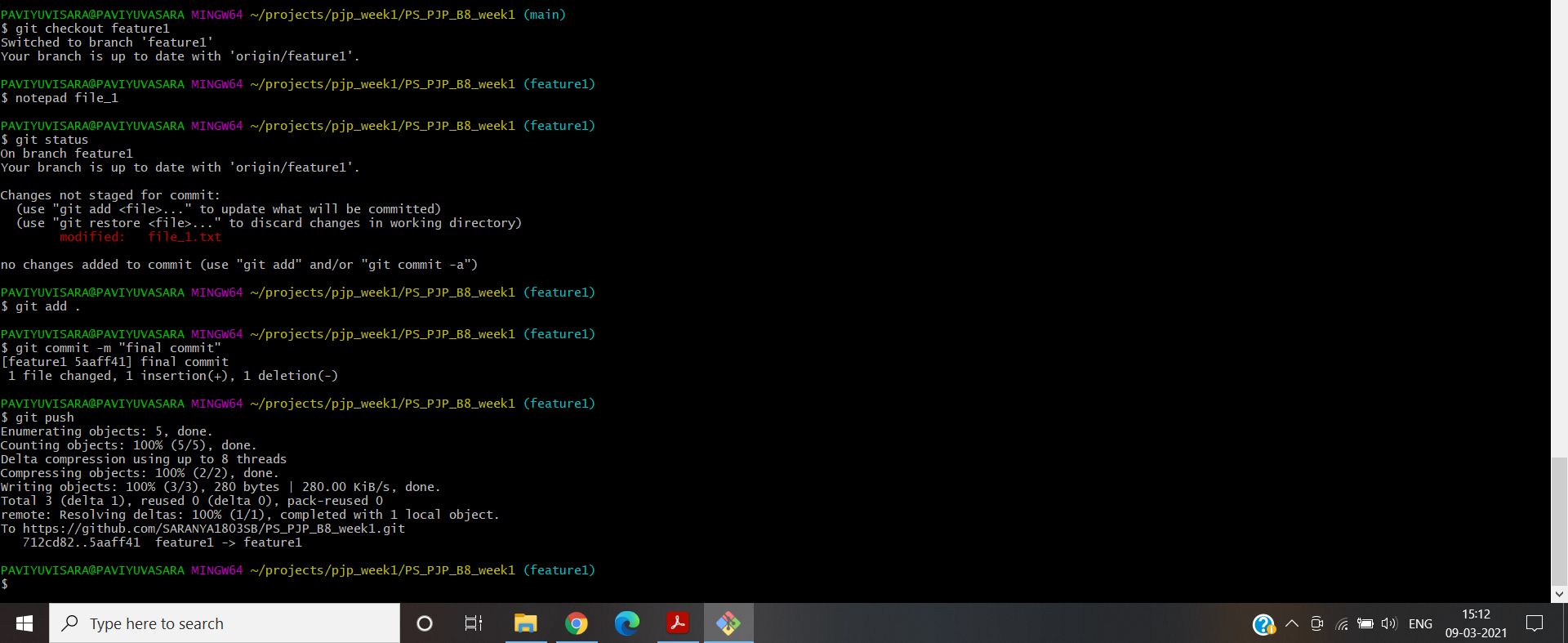
$ git checkout feature1

$ notepad file\_1

$ git add .

$ git commit -m “final commit”

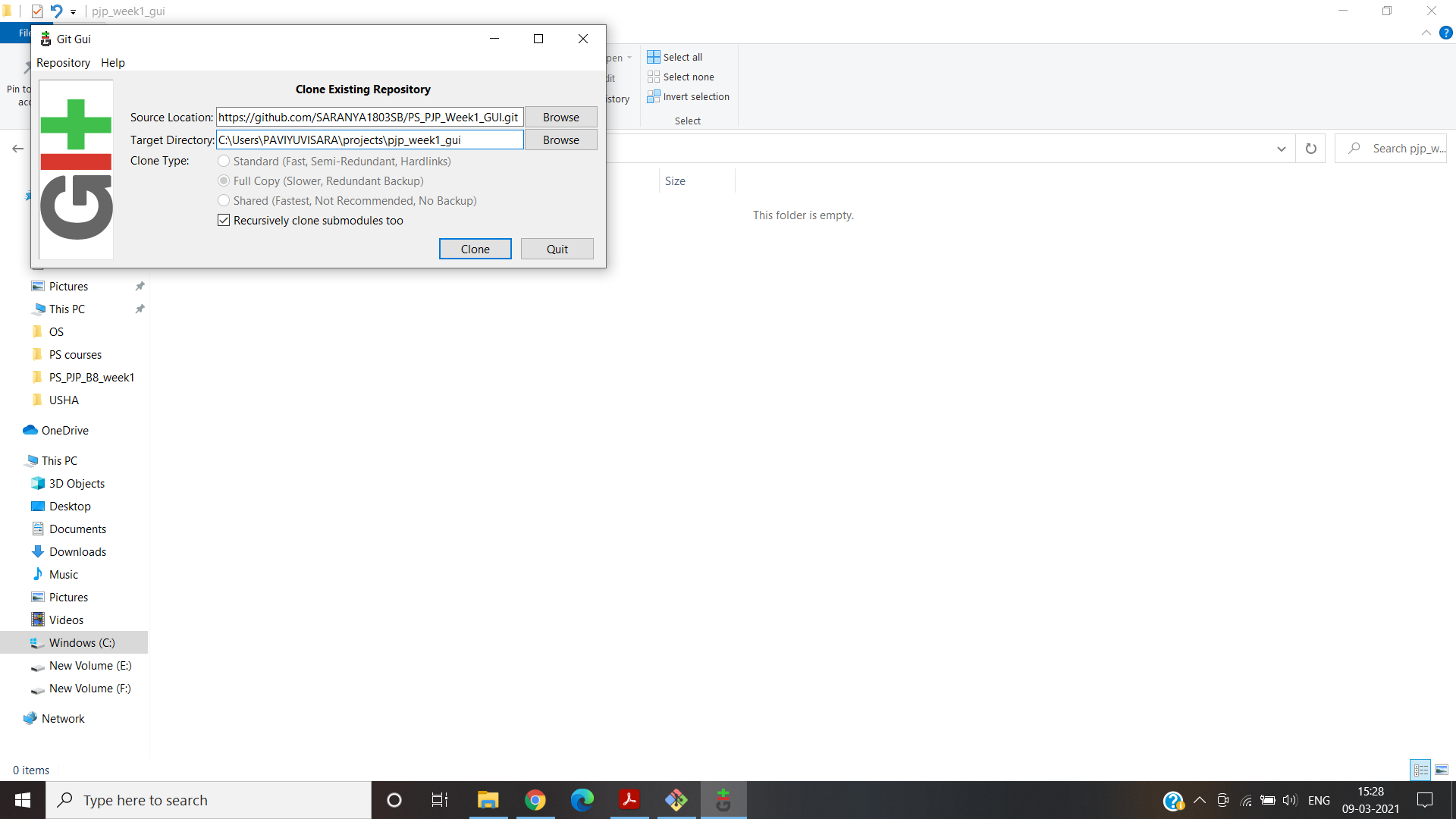
$ git push



I created “pull request” for all the branches from feature1 branch.This will add all the changed folders to all the branches.

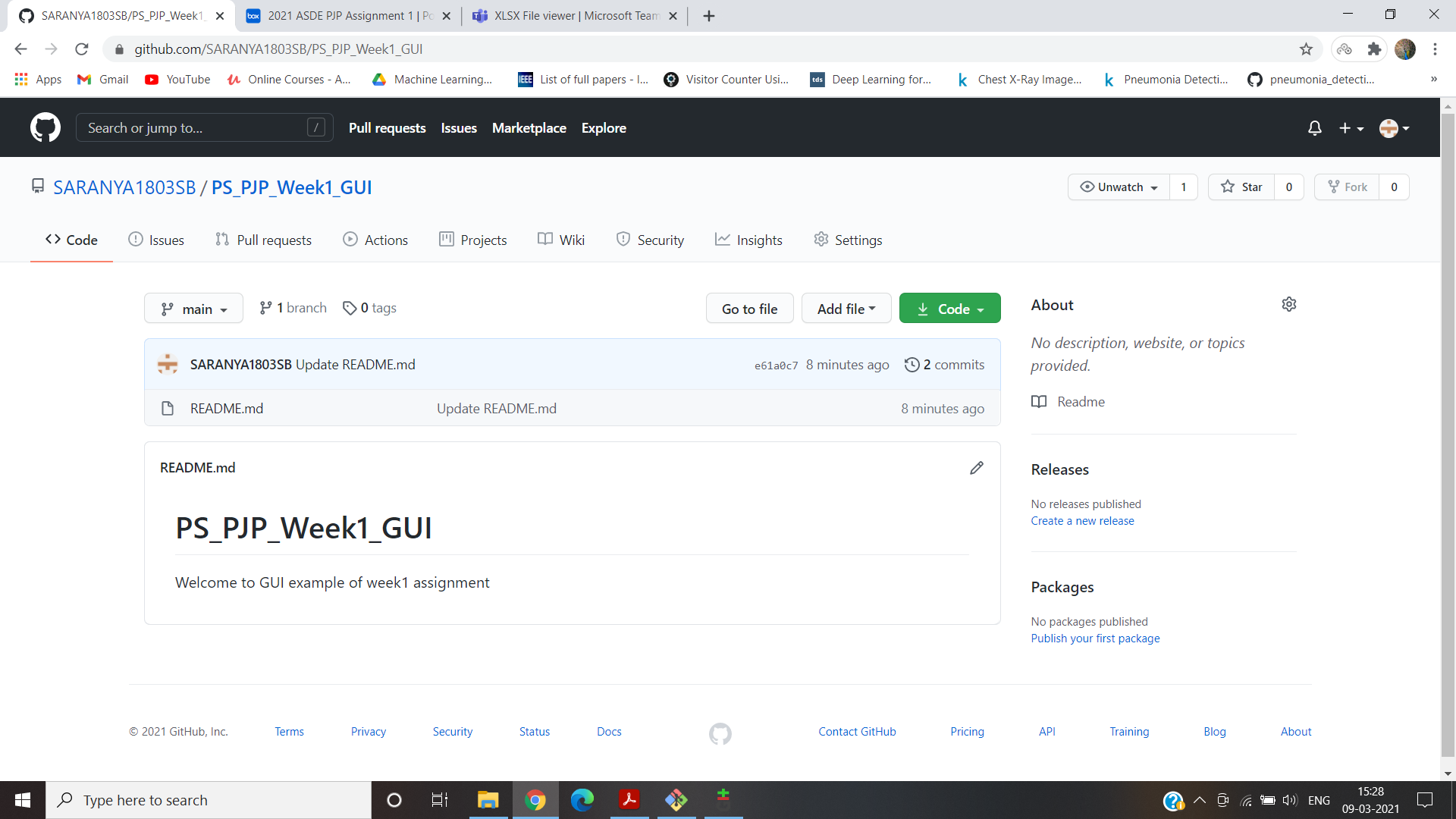
**1.Github -b. Basic usage using GUI client**

i.Local repository:

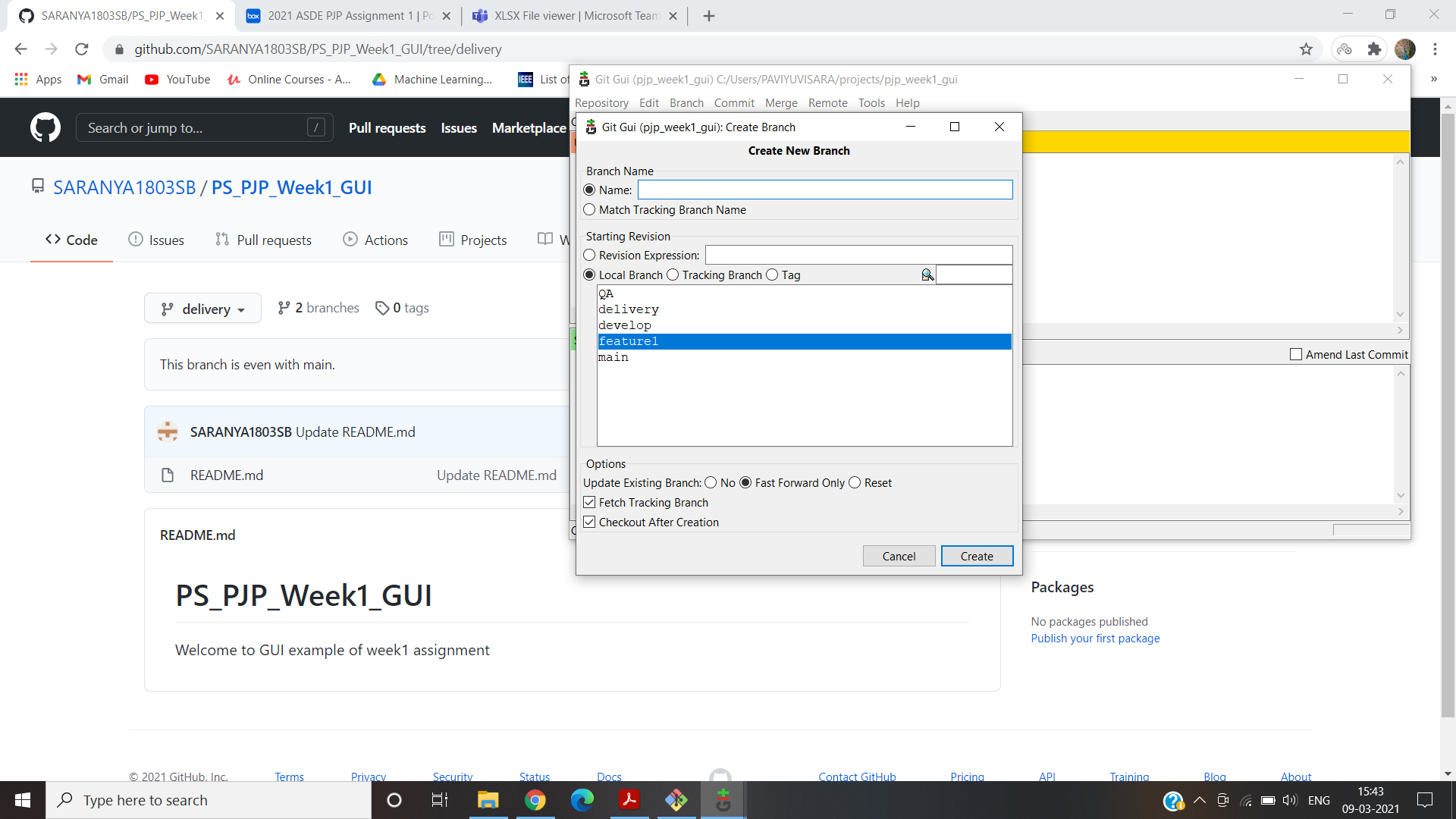


ii.Remote Repository:

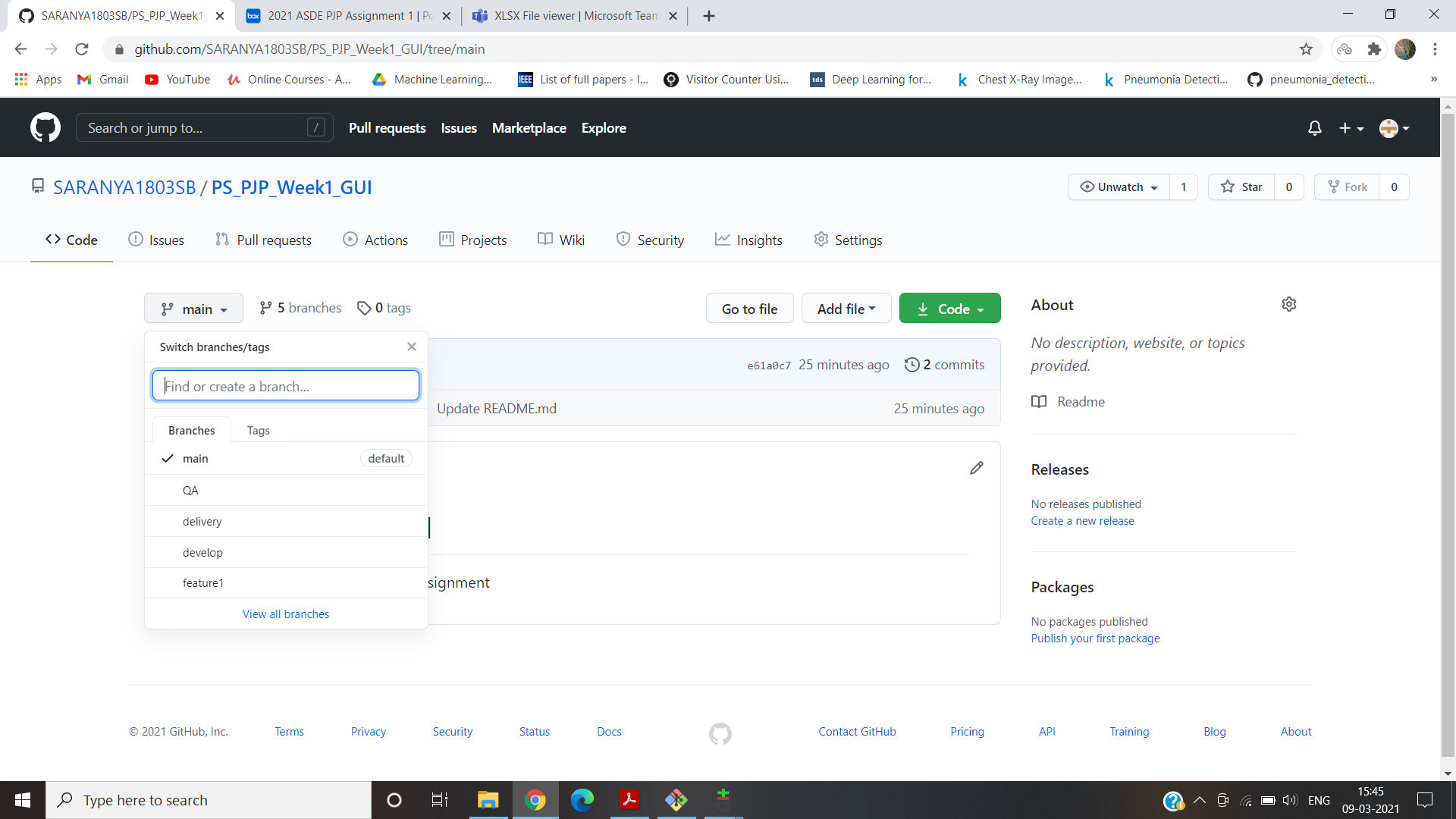
Created a new remote repository named GUI PS\_PJP\_Week1\_GUI.



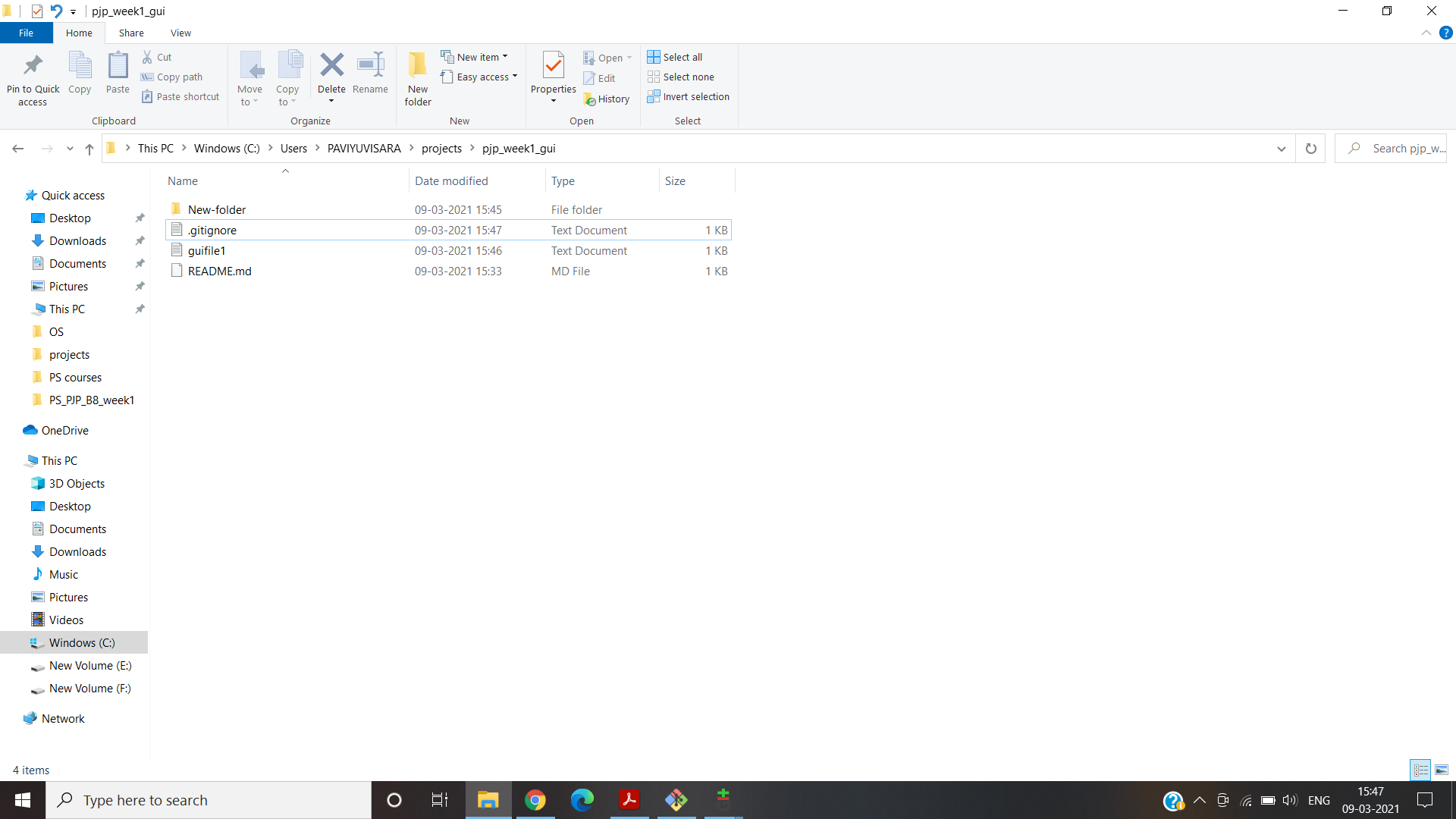
iii.Local branches:



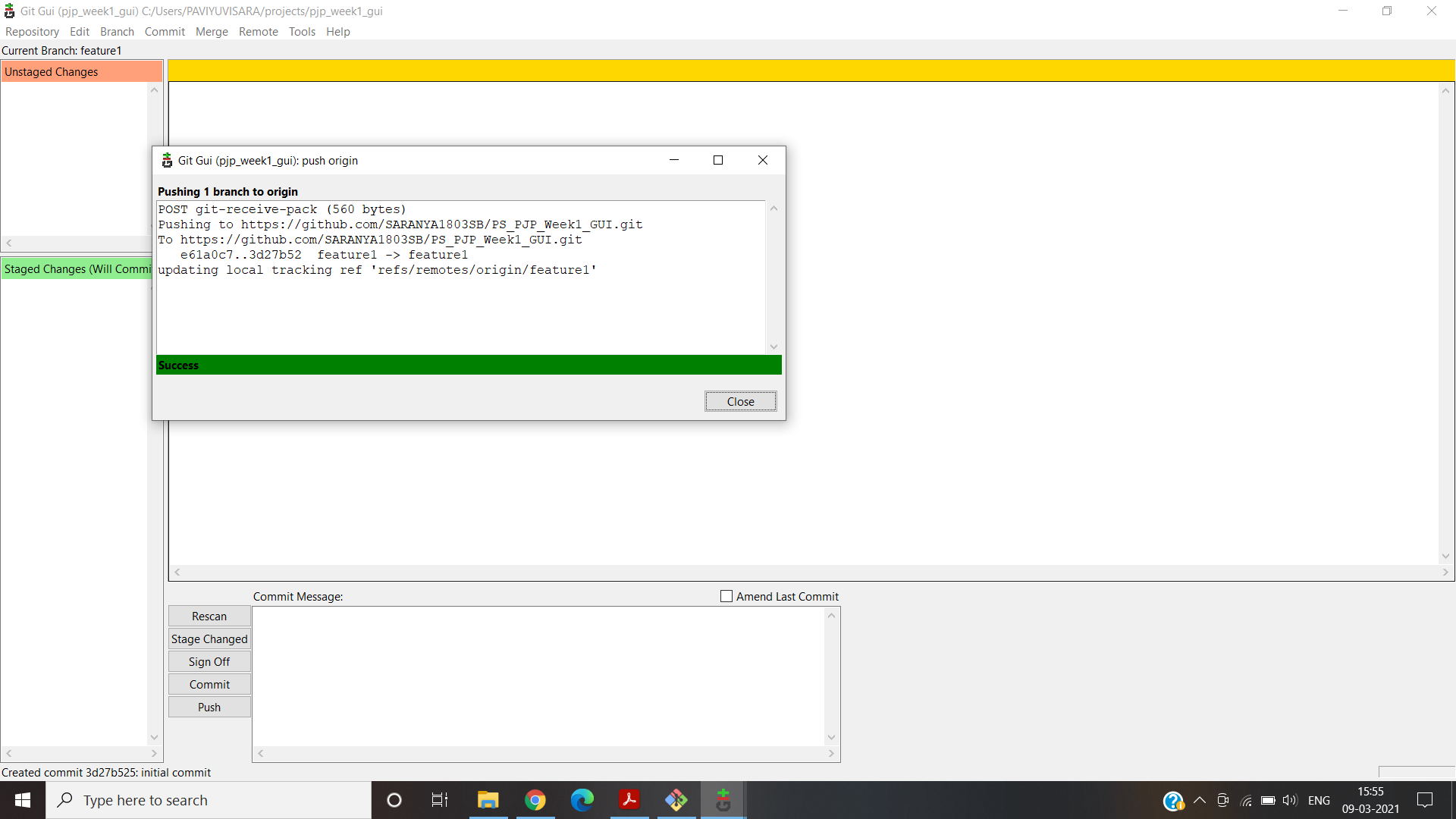
iv.Remote branches:



v.Adding files & folders:

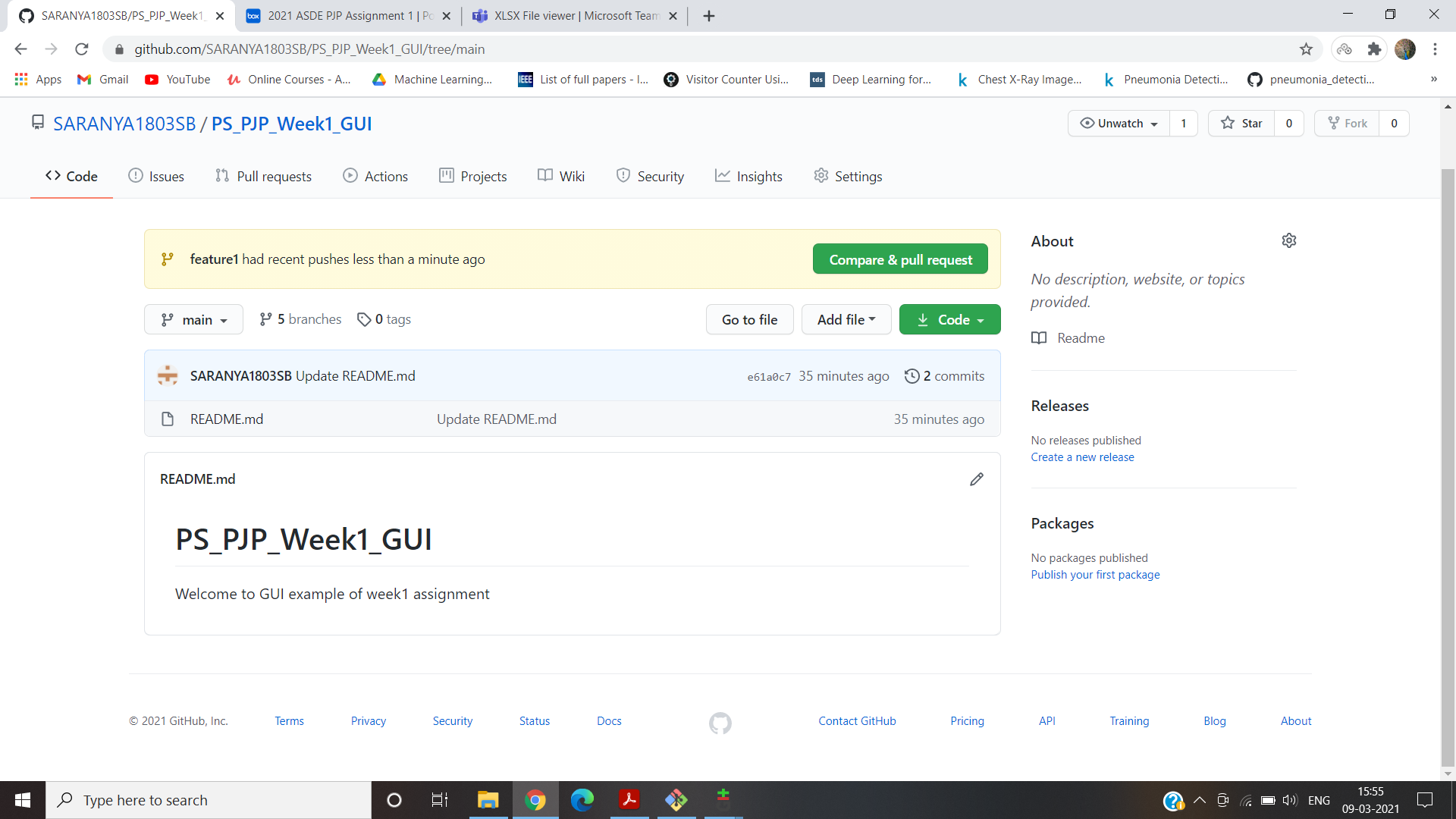


vi.Check-in ,Stage, Commit,Push into feature:

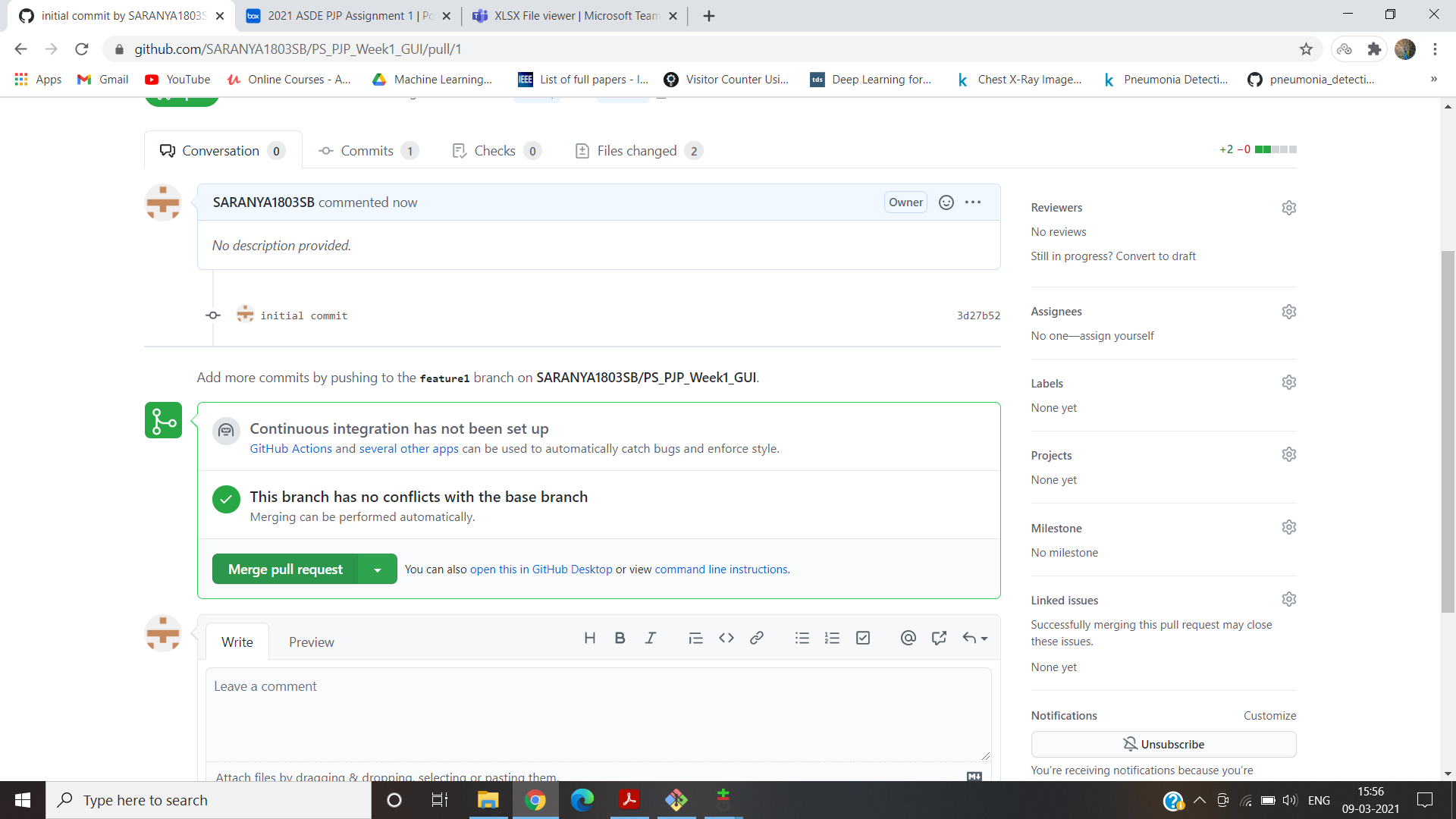


vii.Promote code from feature to develop:

Click “Compare & pull request”:

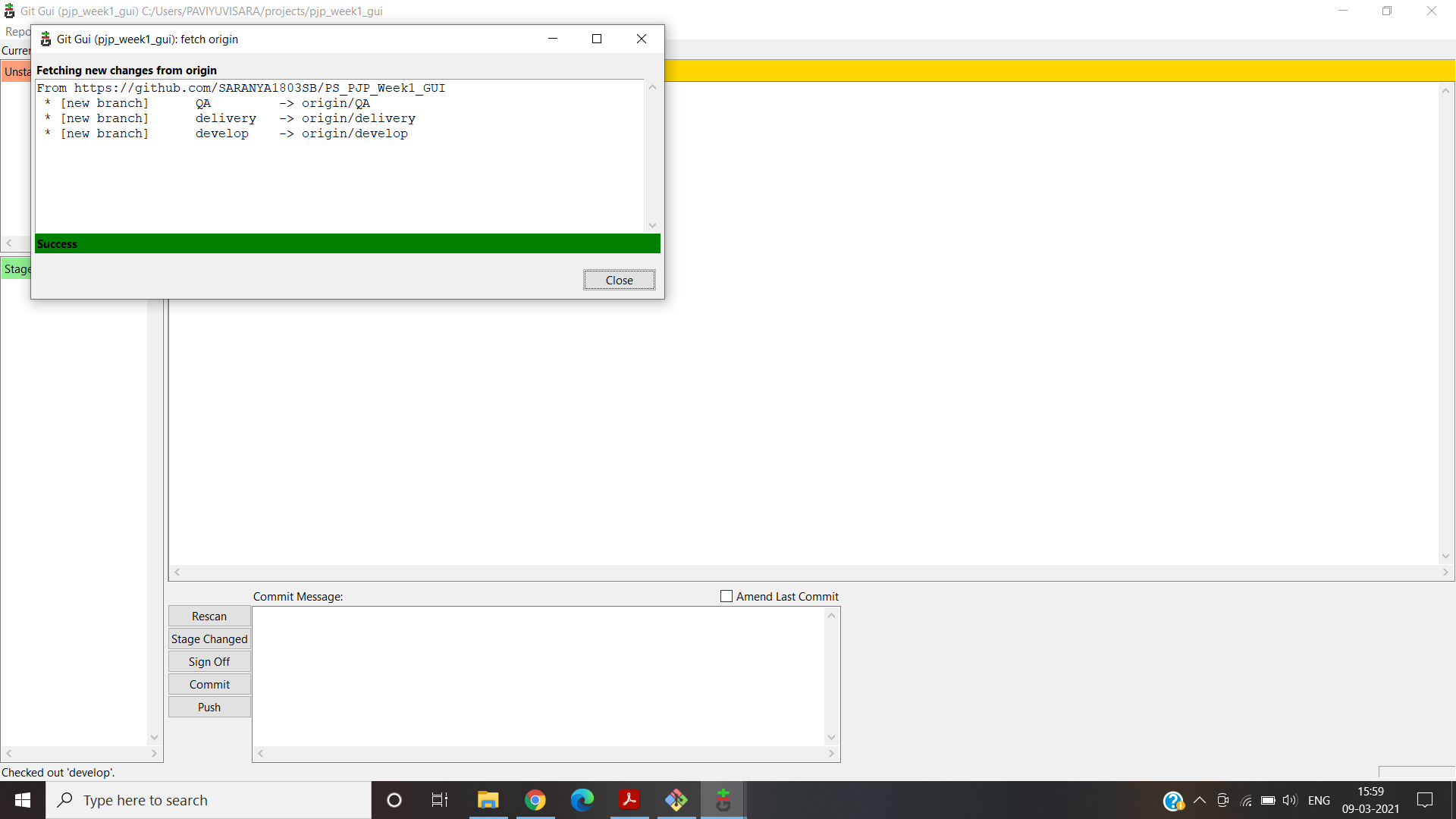


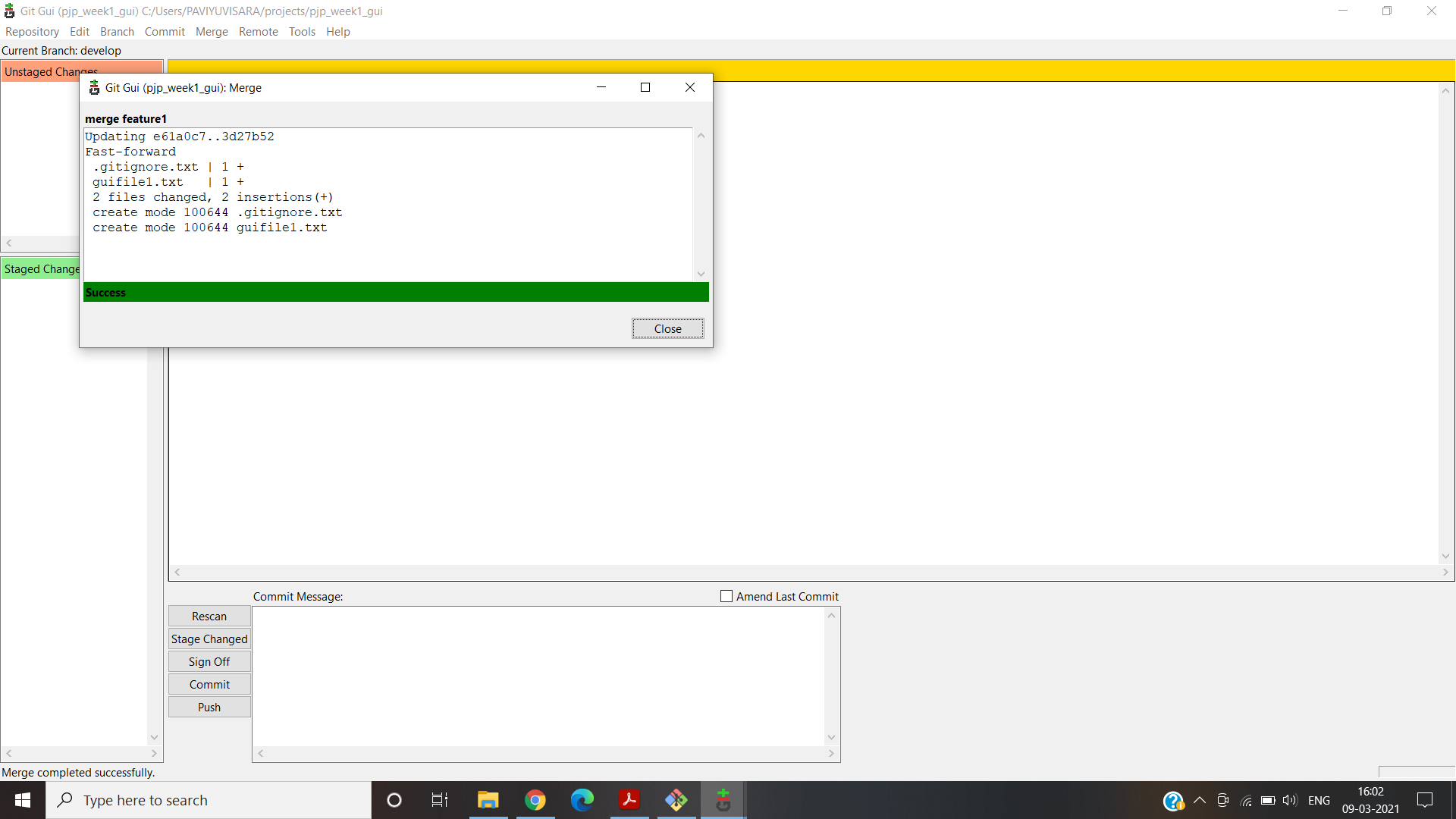
Click “merge pull request”:



viii.Latest code from remote to local:

In remote repository , We created a pull request from feature to develop but it is not updated in local repository. So we update in local repository using GUI. We have to create a fetch and merge in local develop branch.

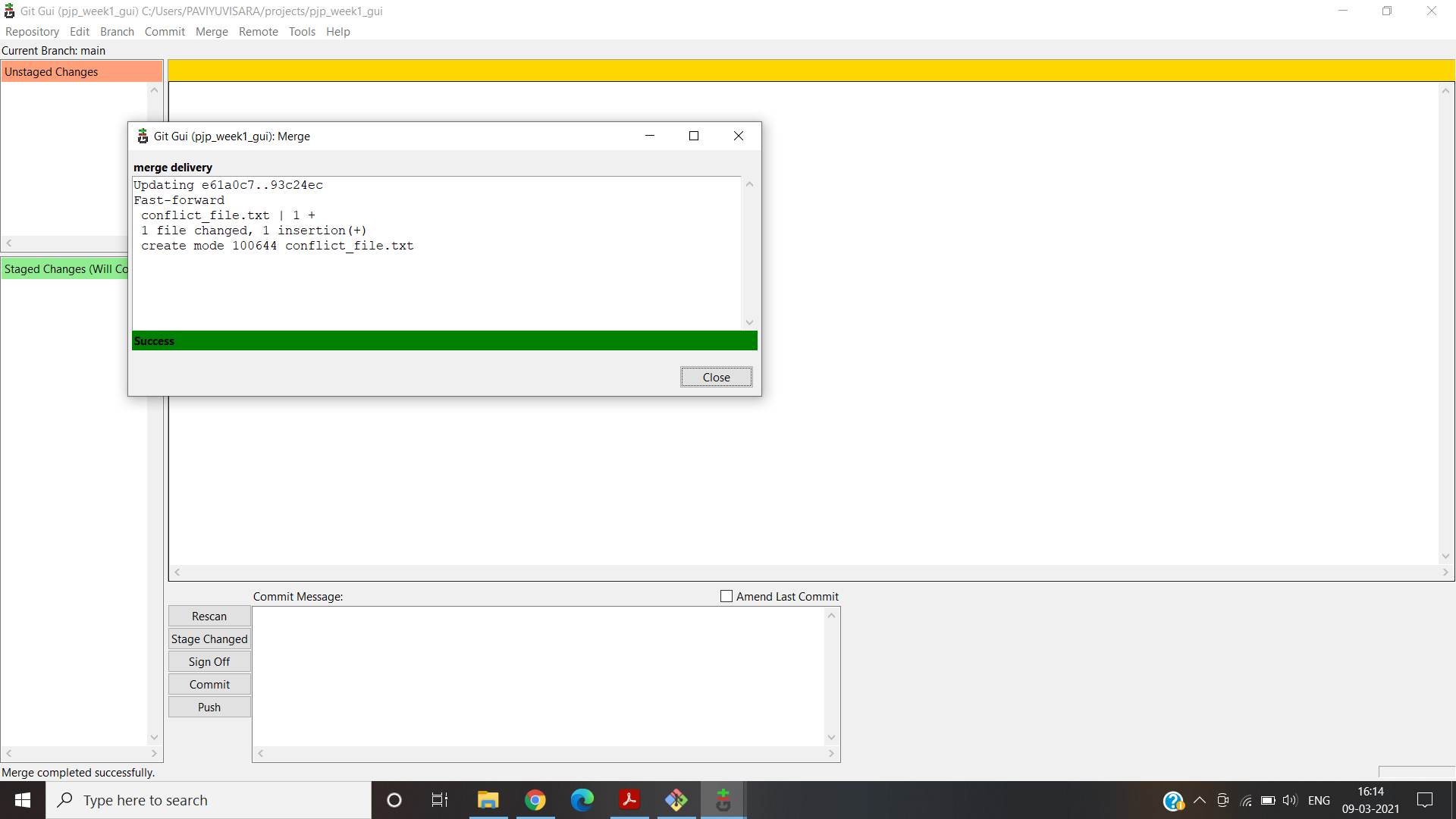




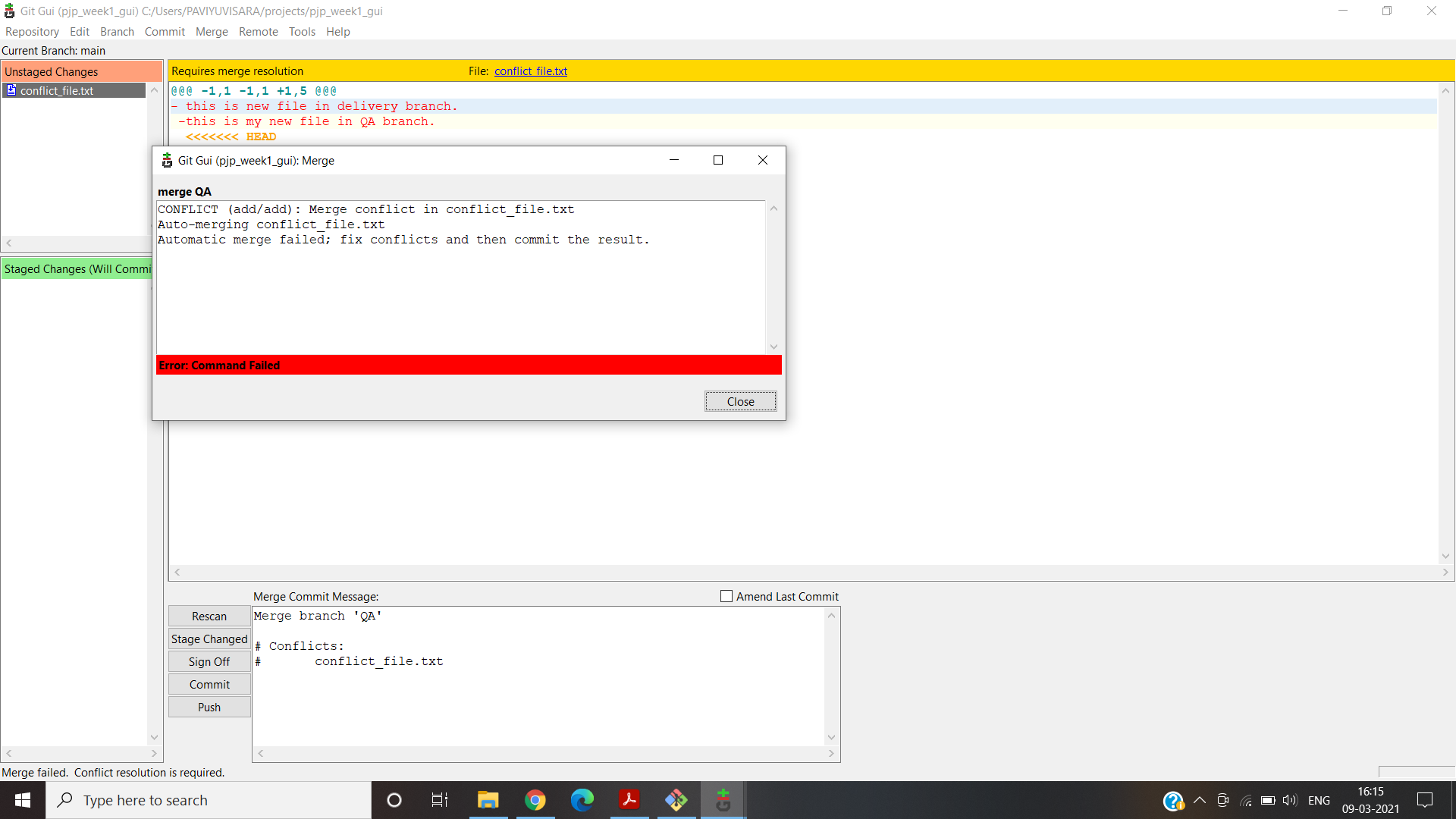
x.Handle Merge Conflict:

The Merge conflict was created by the same Merge Conflict method used the above section ,but here with the help of GUI Tool rather than CLI.

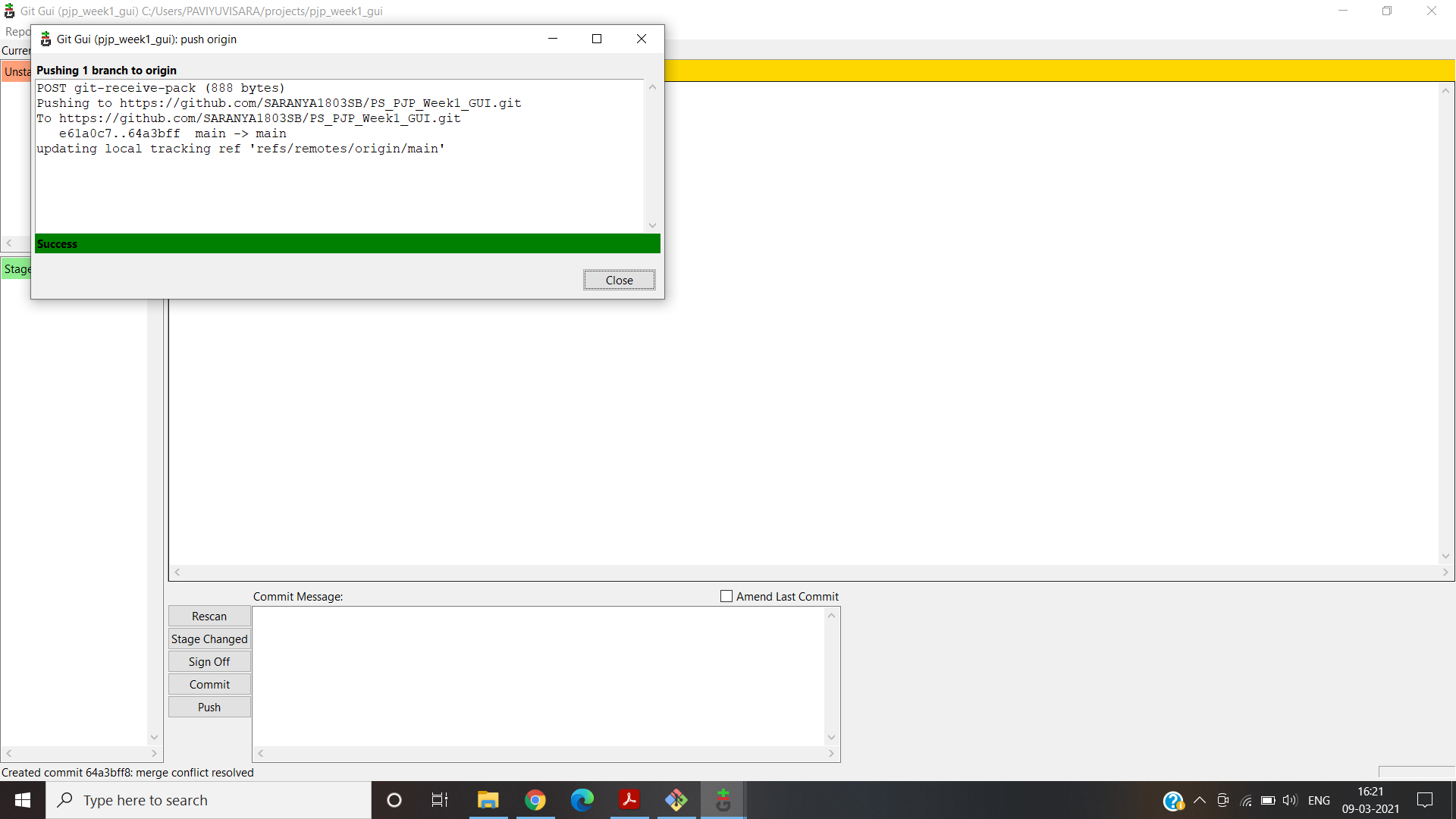
First merge to main with branch delivery



Second merge to main with QA branch – Merge Conflict occurs here



Using Ctrl+T will resolve this issue or commit->stage to commit.



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

All File are updated to main.

**1.c.On the ground day to day scenario:**

i.Reset / Revert (soft reset & hard reset):

Create “new\_file” text document.

$ git add .

$ git commit -m “commit1”

Add some more lines in “new\_file”

$ git add .

$ git commit -m “commit2”

$ git log -------------------------------------------🡪show all the commits done so far

$ git reset –soft HEAD~1 --------------🡪deletes the commit2 & moves “new\_file” to staging area

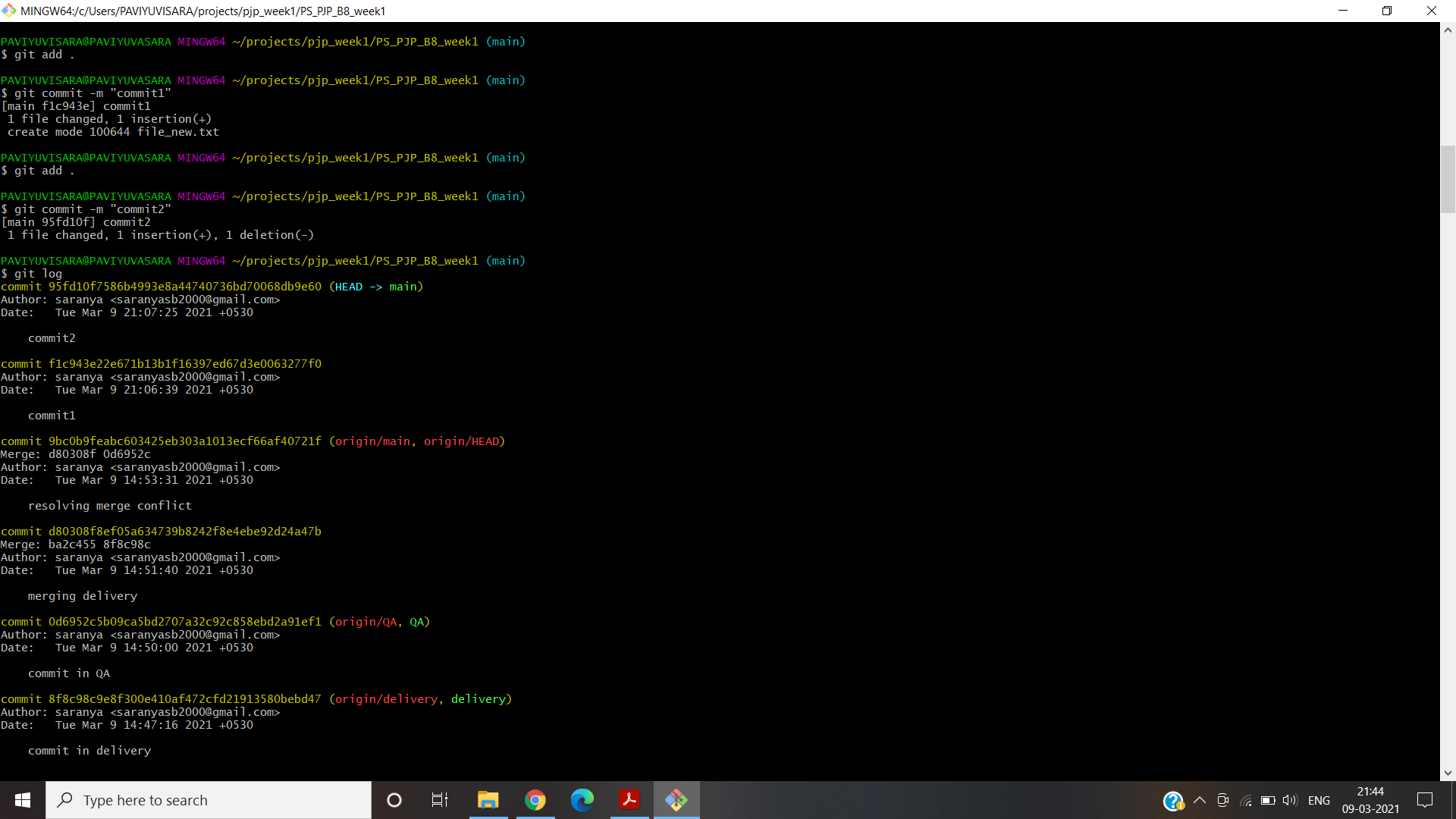
$ git status --------------------------------------------🡪to see that the “new\_file” is in statging area

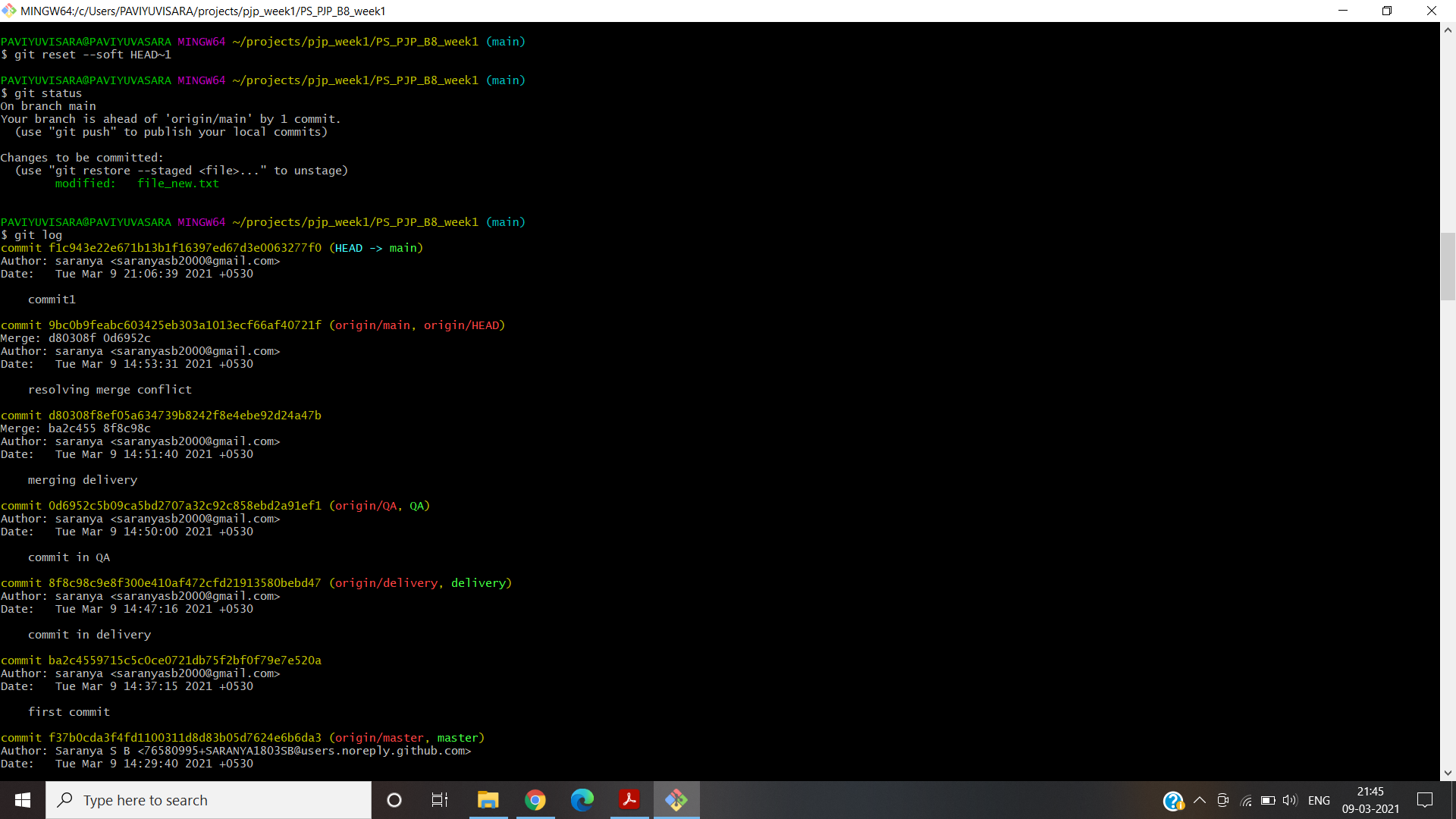
$ git log ------------------🡪to see the absence of “commit2”

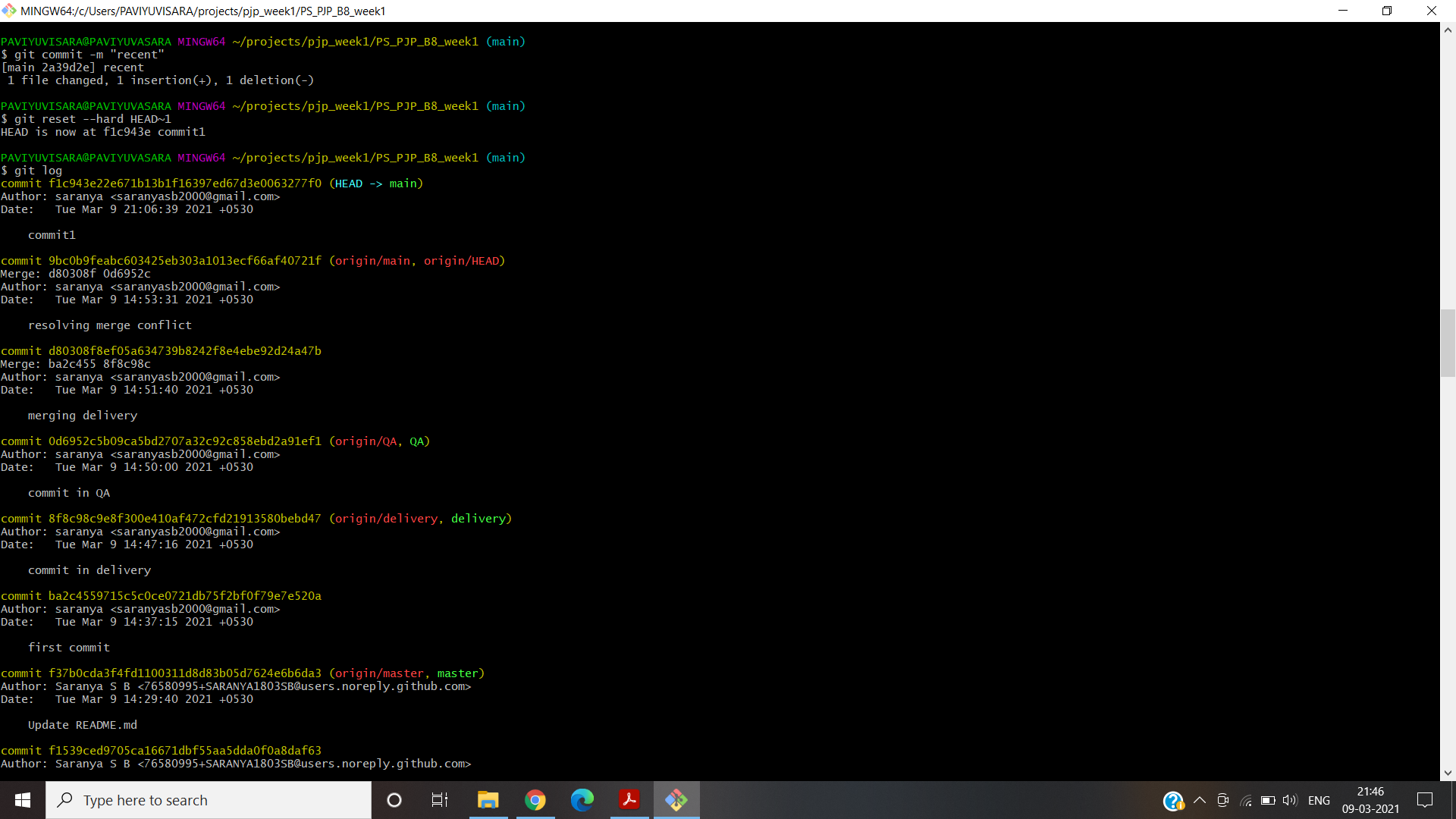
$ git commit -m “recent” --------------------🡪again committing “new\_file”

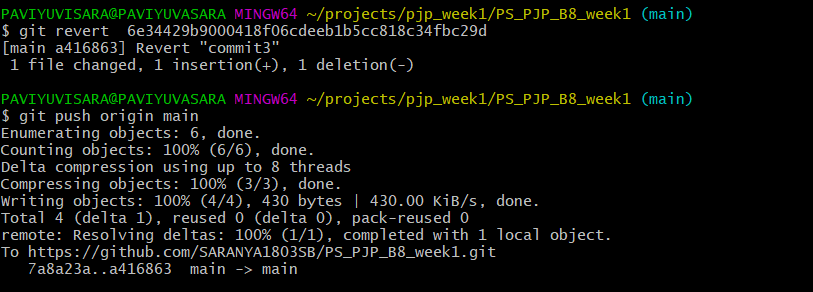
$git reset –hard HEAD~1 ------------------🡪deletes the “recent” commit

$ git log ------------------------------------🡪to see the absence of “recent” commit









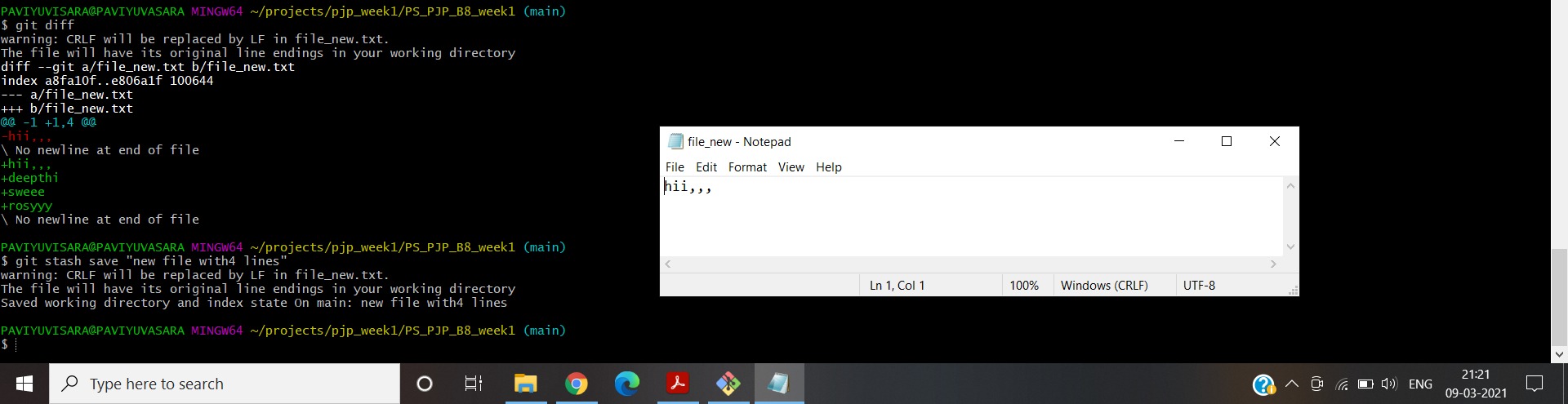
ii.Stash local changes during merge conflicts:

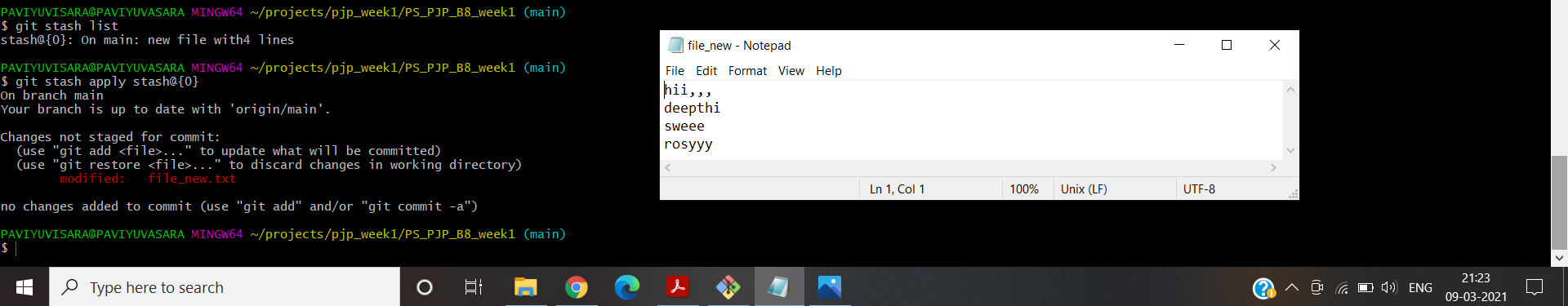
$ git diff

$ git stash save “new file with 4 lines”

$ git stash list

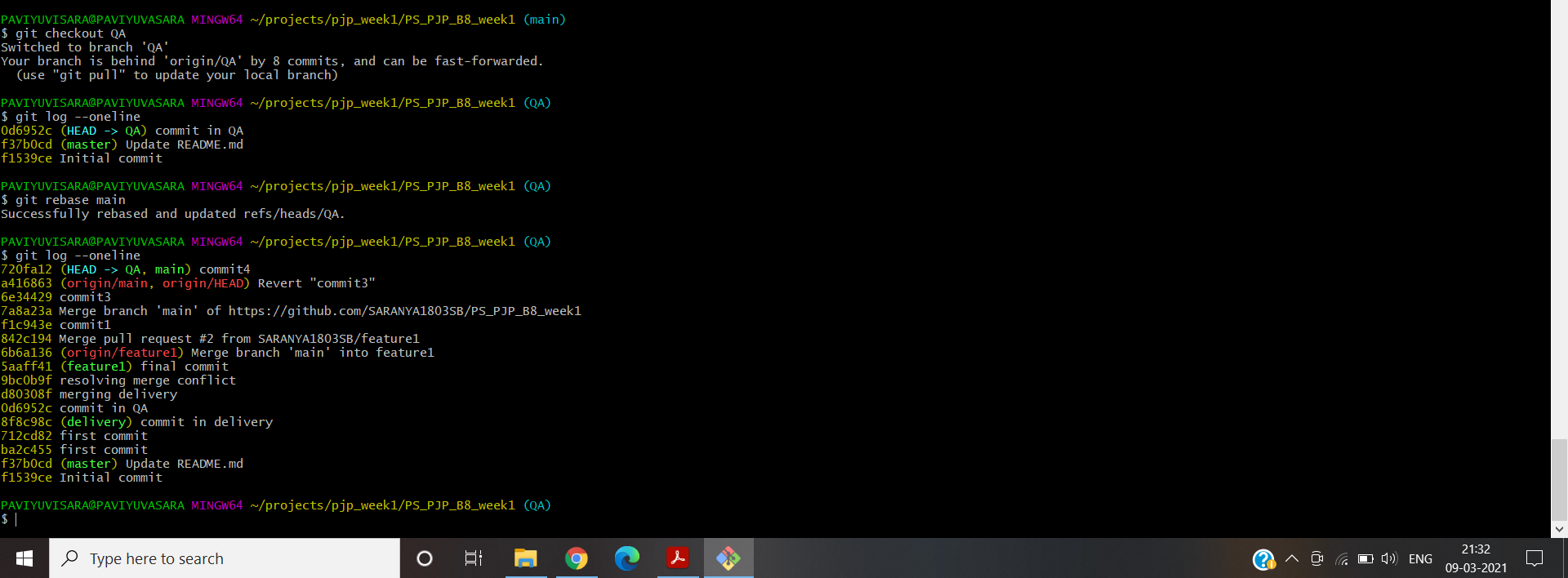
$ git stash apply stash{0}





iii.Rebasing – reword,edit,squash,fixup,exec,drop:

$ git rebase <branch name>



iv.git log,status & reflog:

$ git log –oneline

$ git status

$ git reflog

