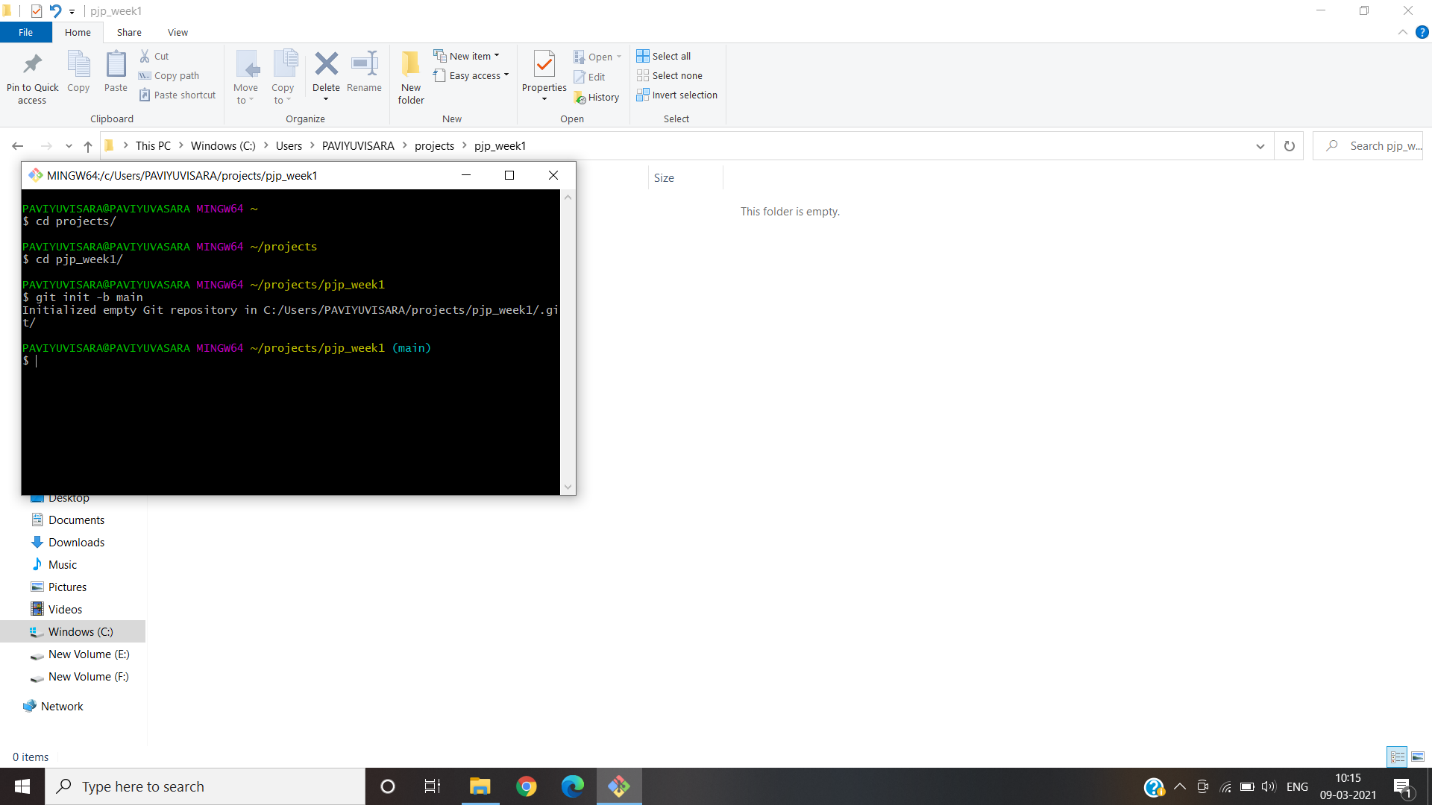
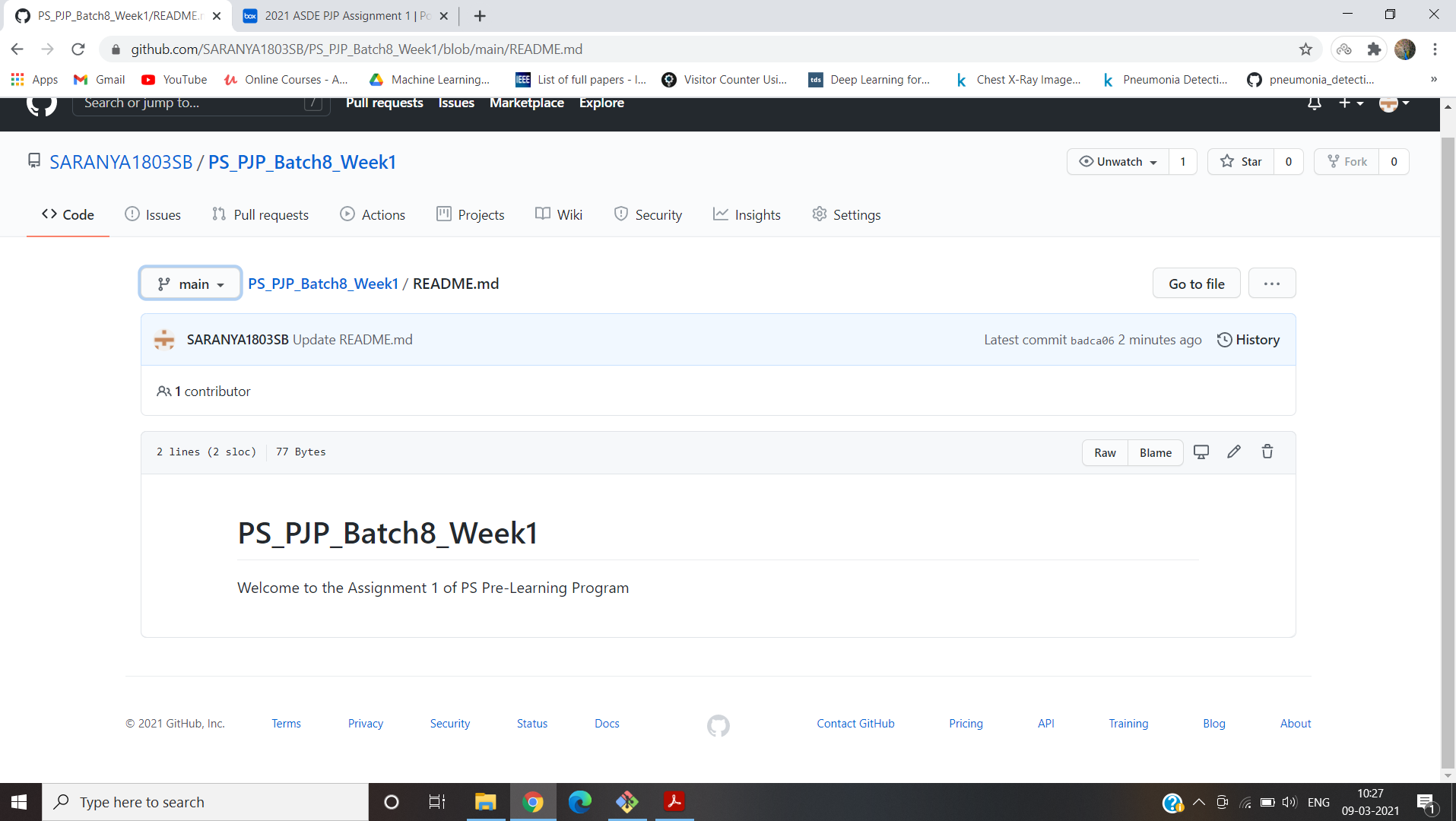
**PS PJP Week1 – Assignment**

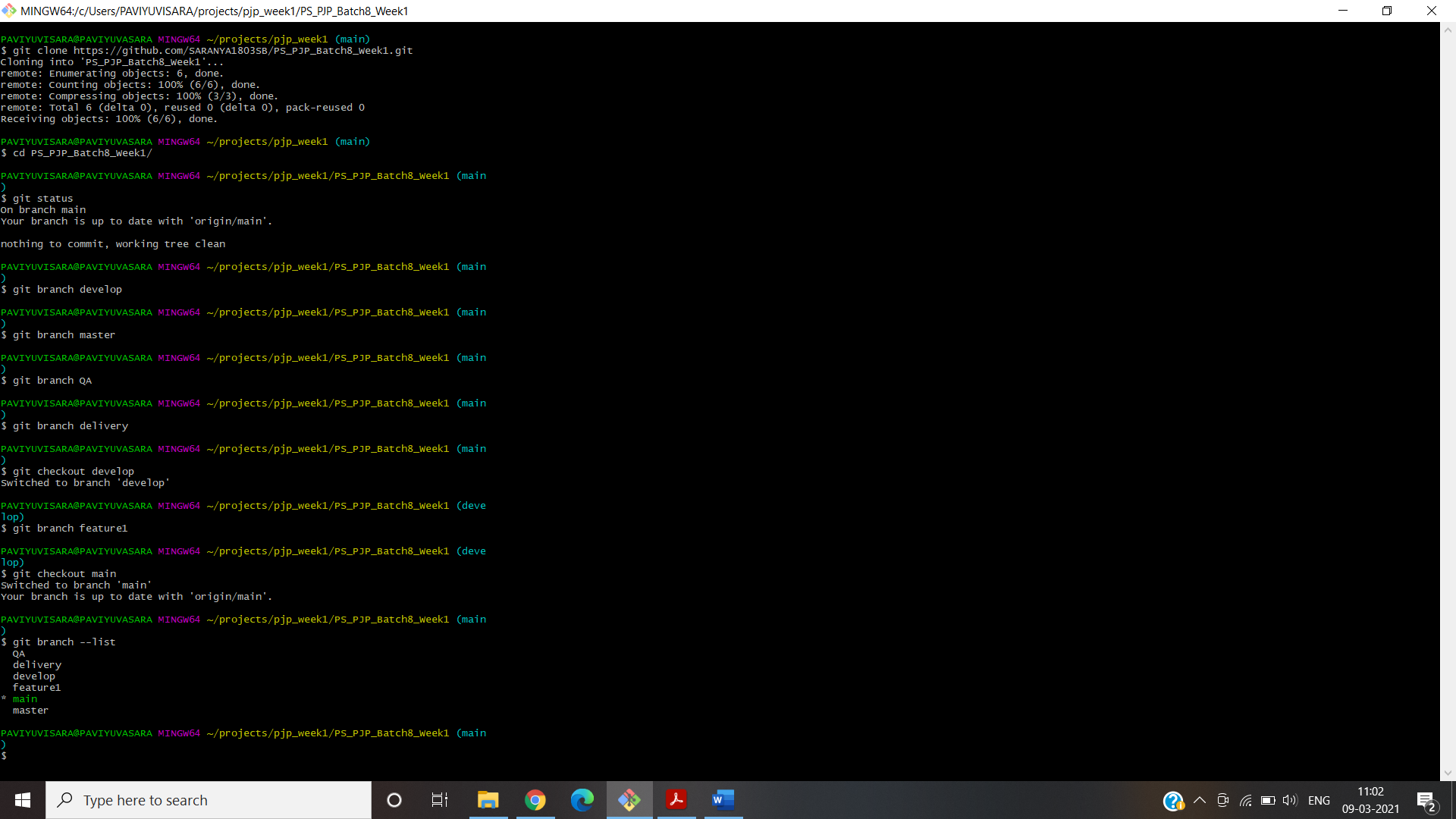
-Saranya.S.B,

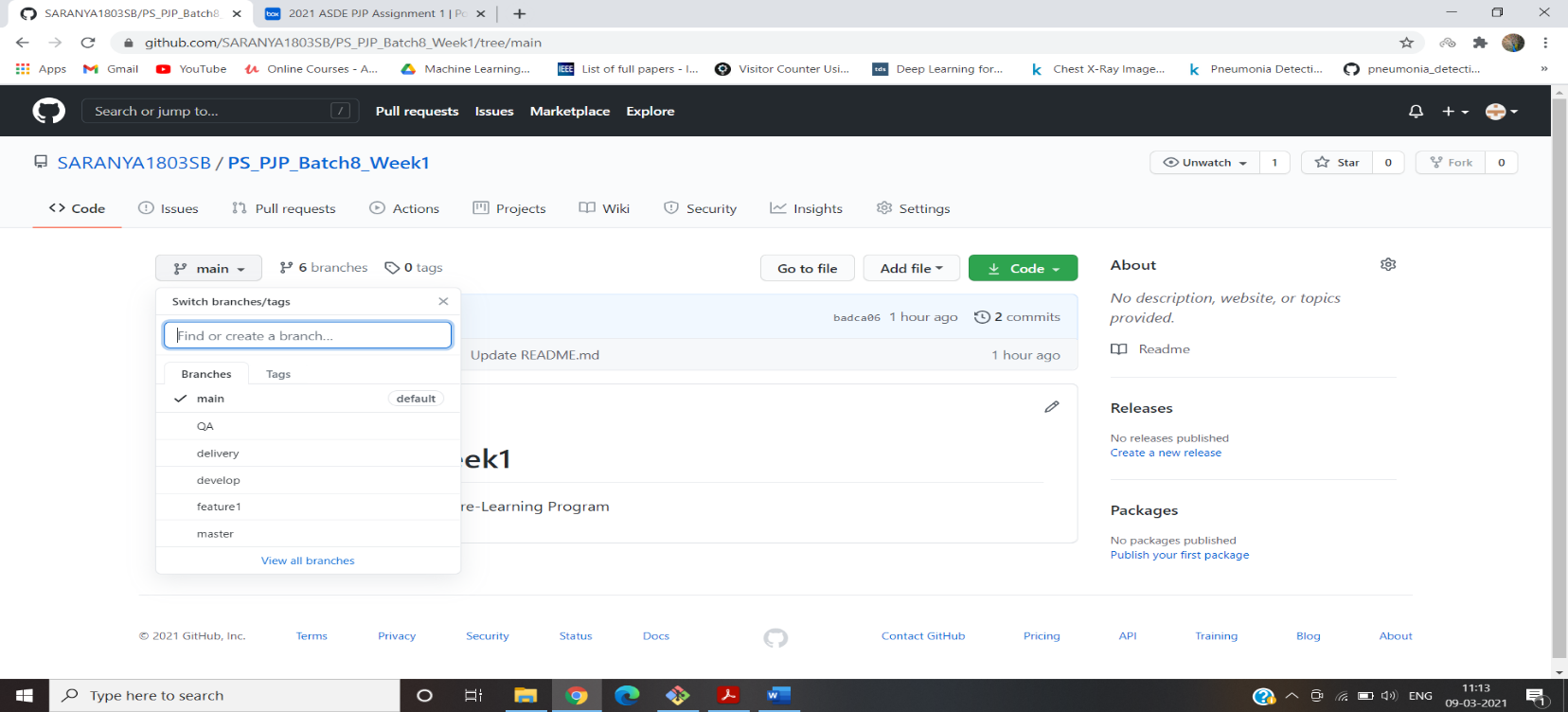
Anna University

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

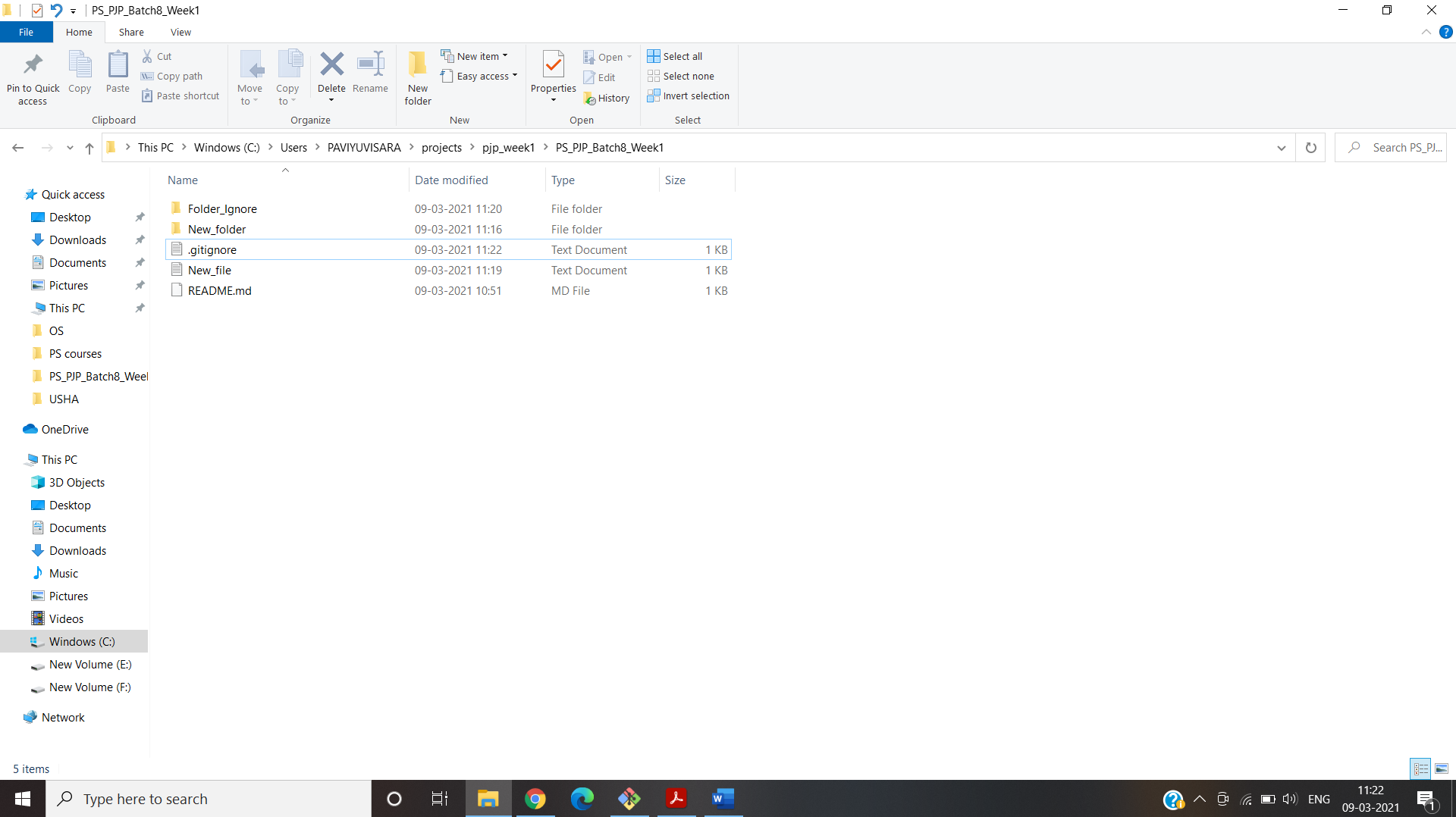
i. Setup a Local repository: 

ii. Setup a Remote Repository: 

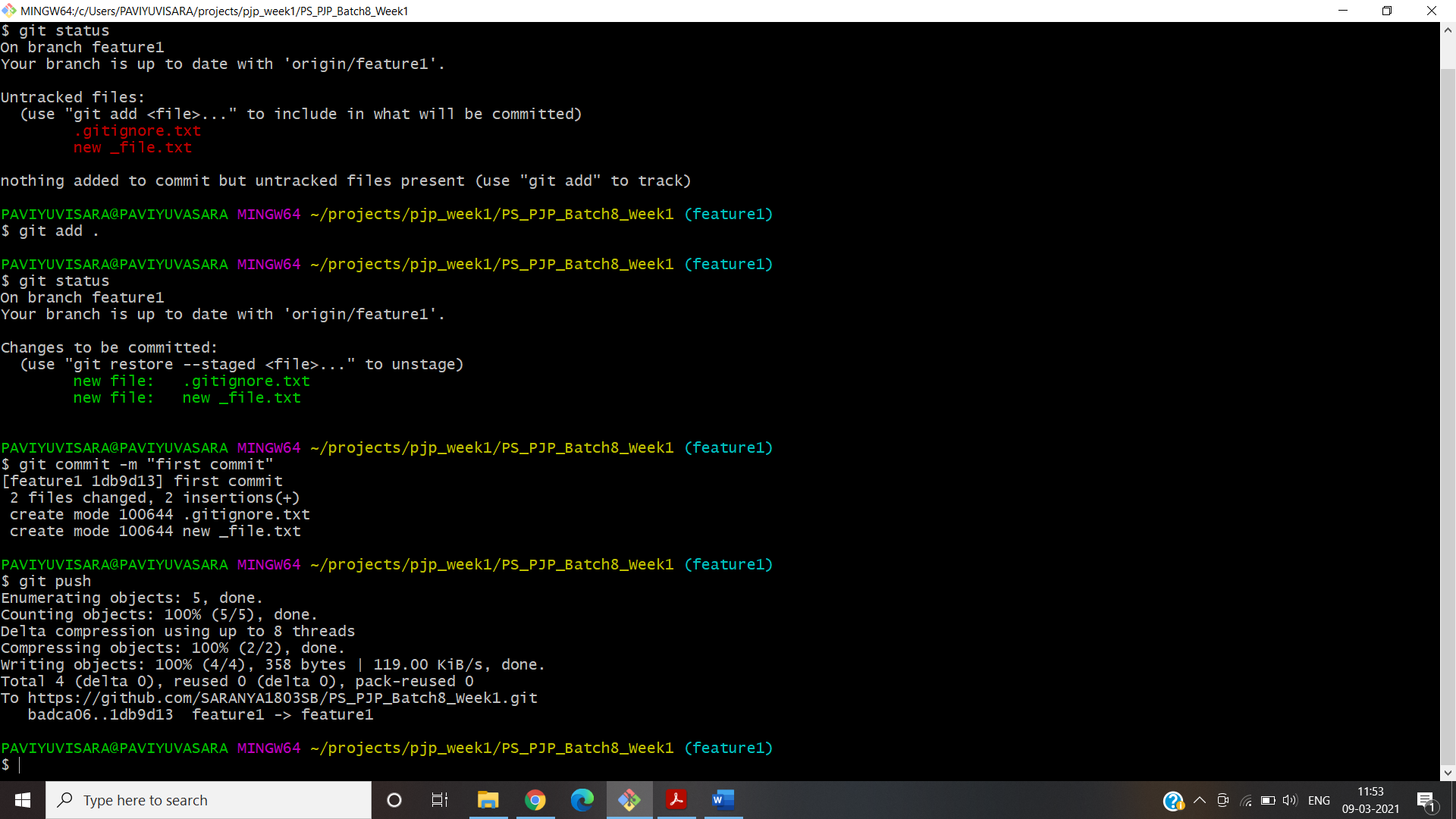
iii. Create local branches: 

iv.Create remote branches: 

v. Adding files and folders:



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



vii. Promote code from feature to dev:

Click “Compare & Pull request”:

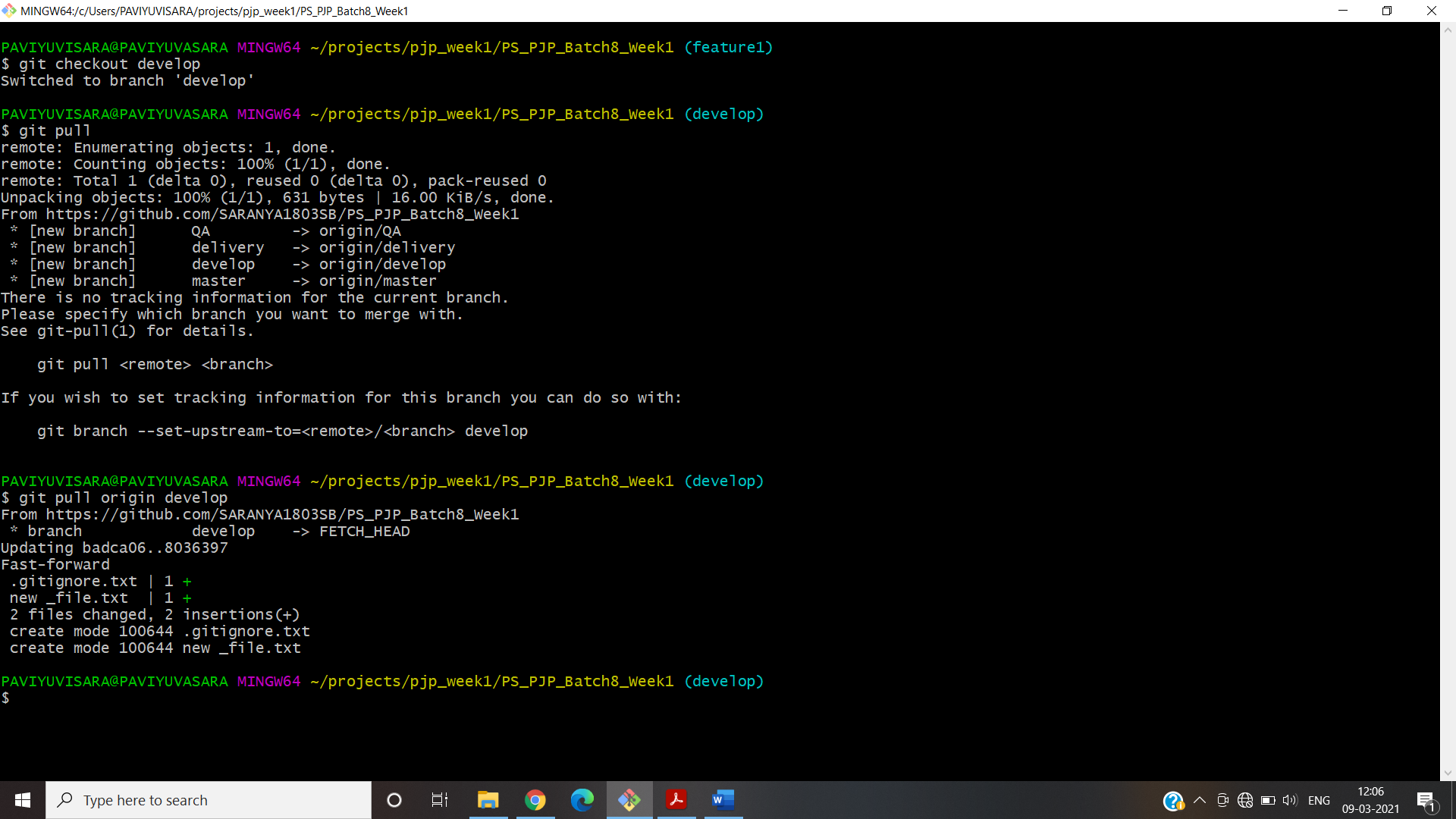


Click “Create pull request”:



viii. Latest code from remote to local:

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



ix.Checkout Vs Pull:

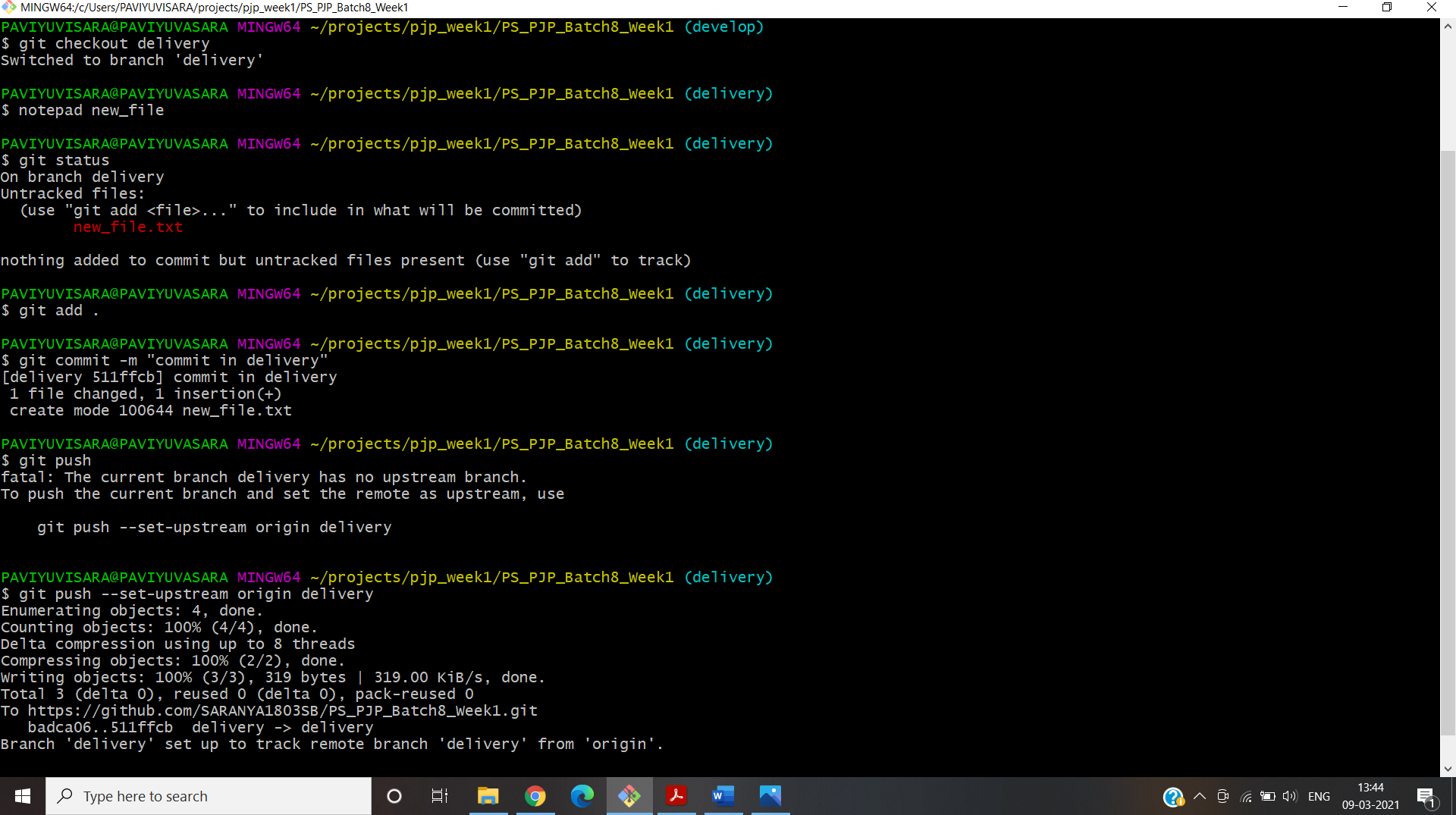
Checkout: It checks for the update in a branch but does not merge the updates.

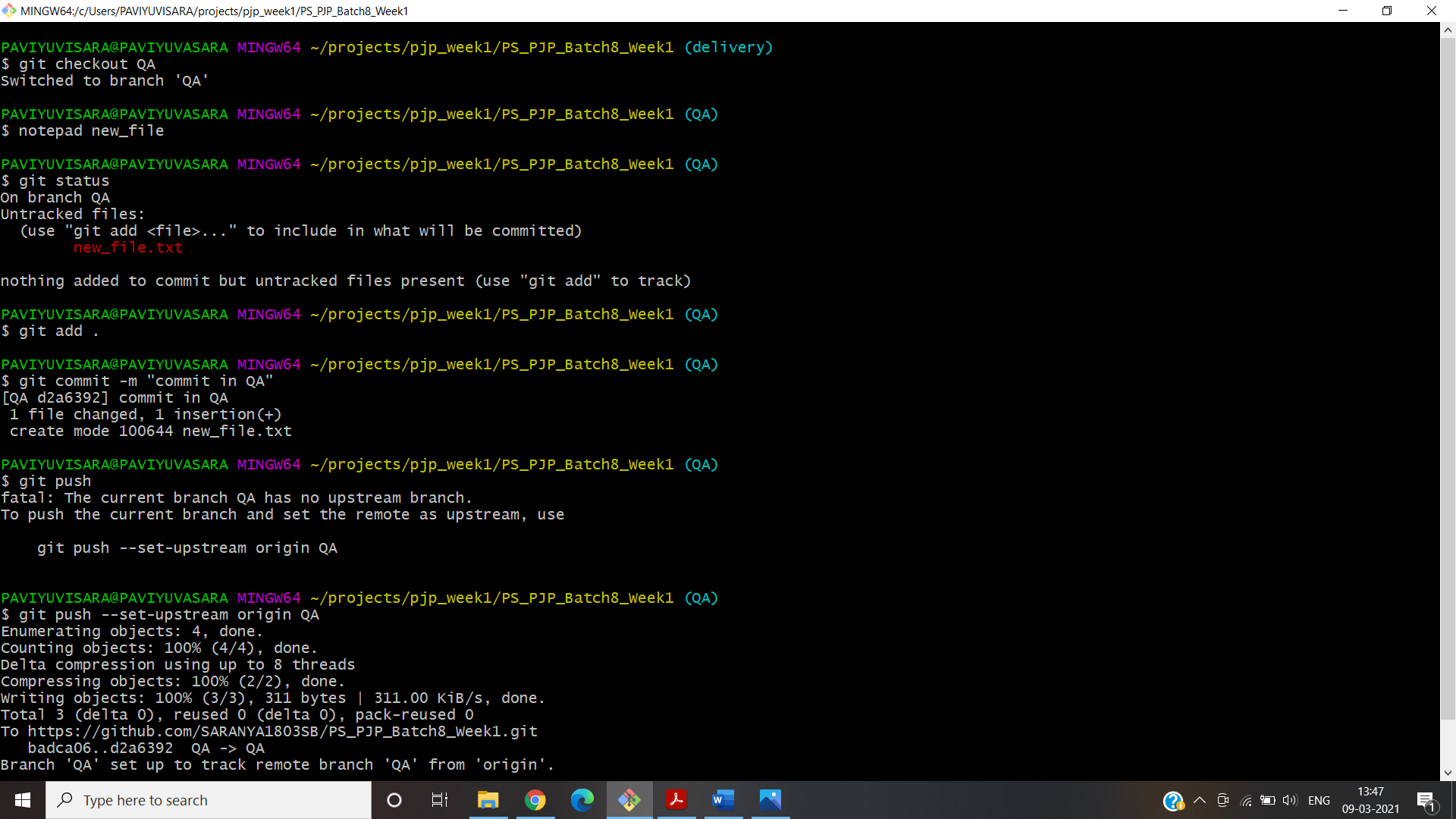
Pull: It check for the updates and merges in local branch.

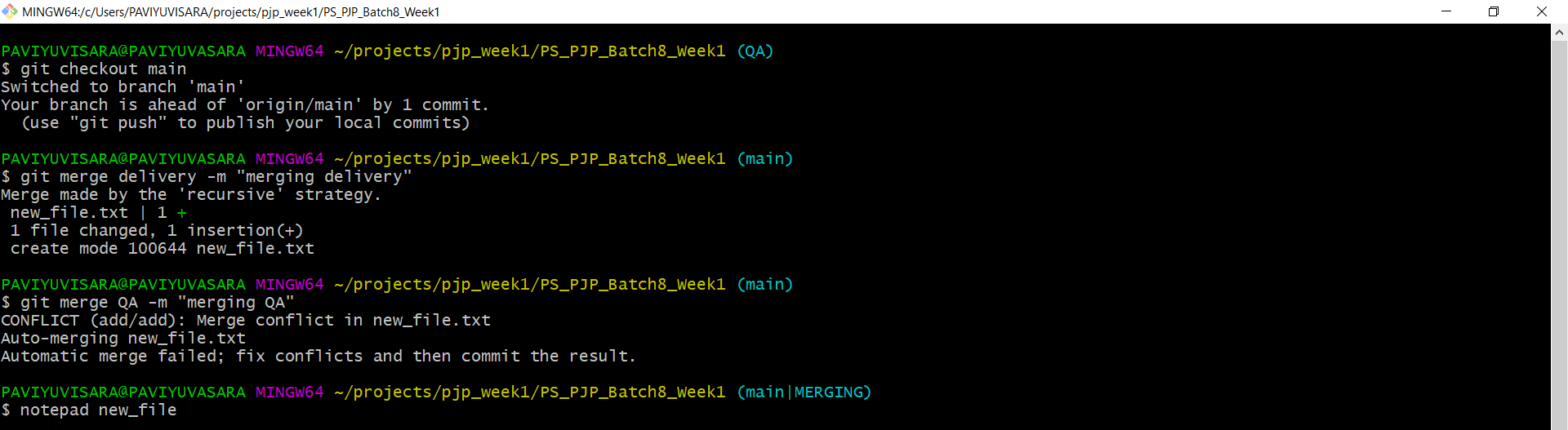
x. Handle Merge Conflict:

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”.

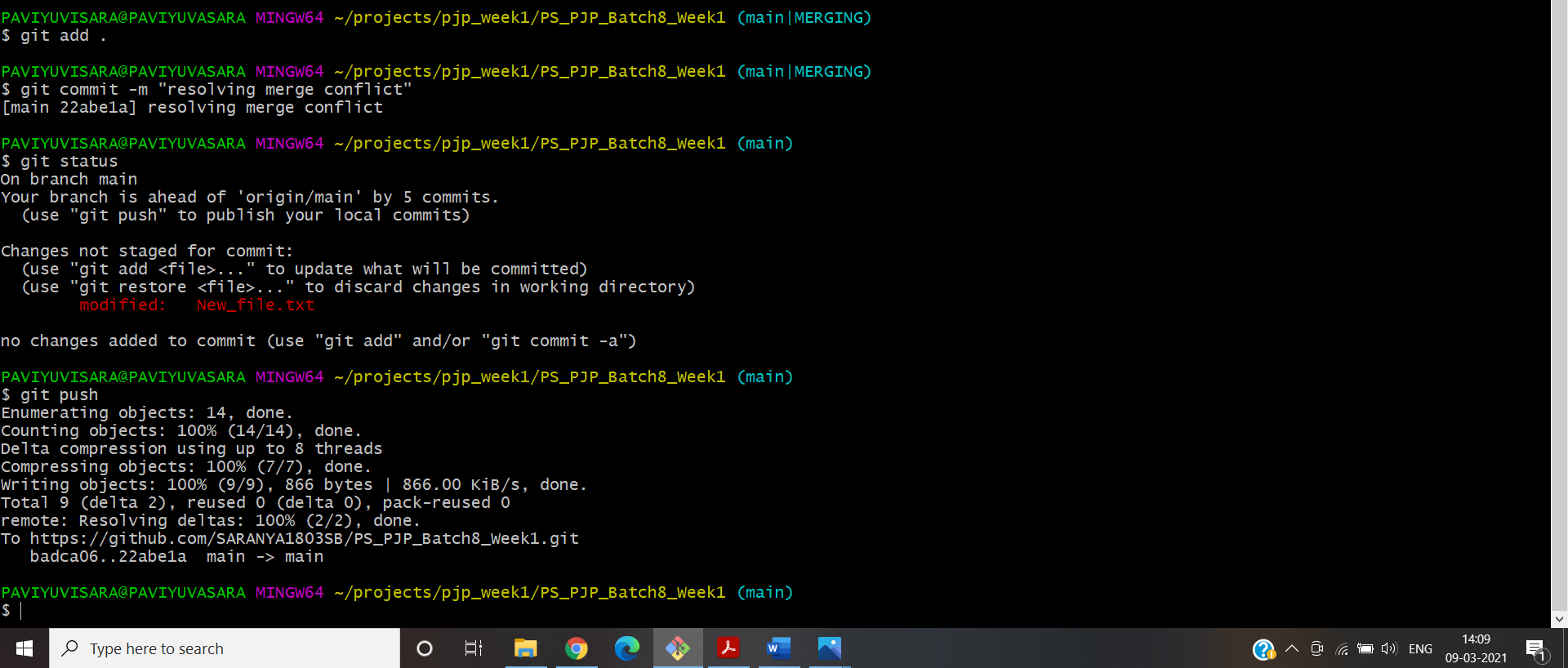






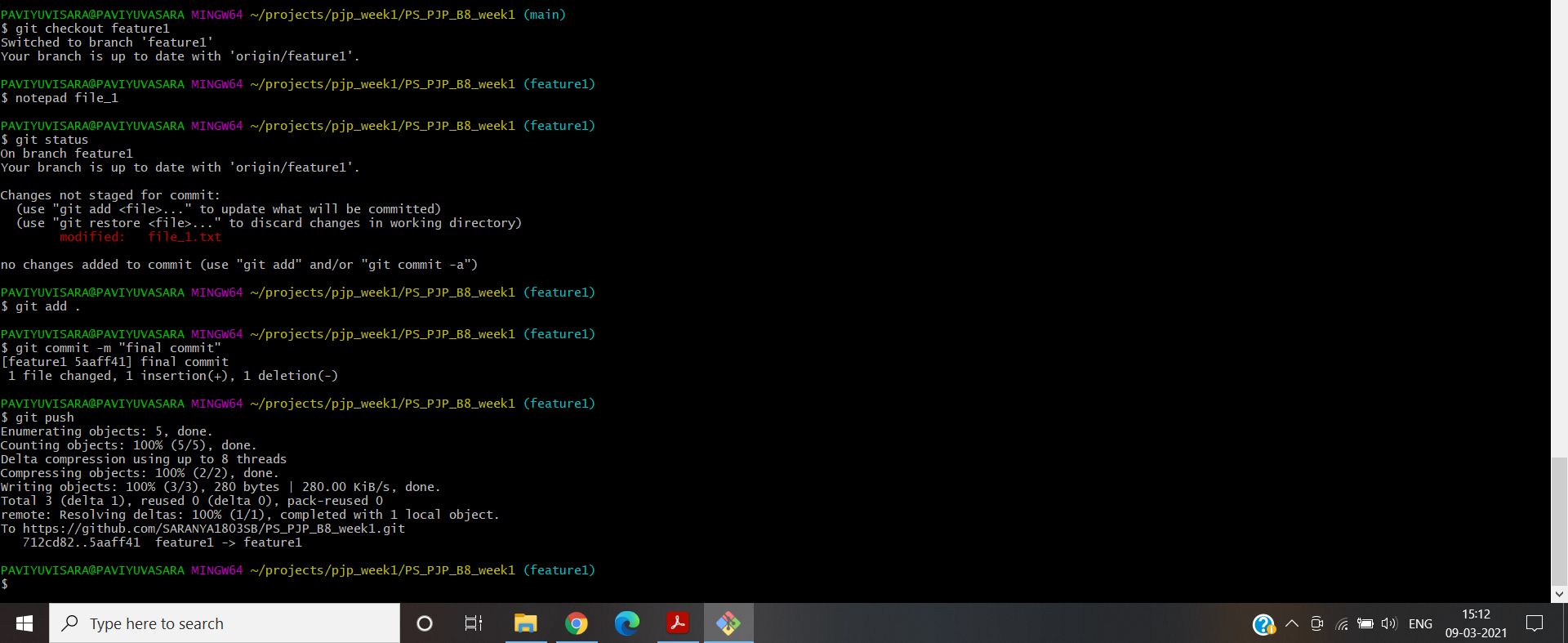
Handling merge Conflicts:

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



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

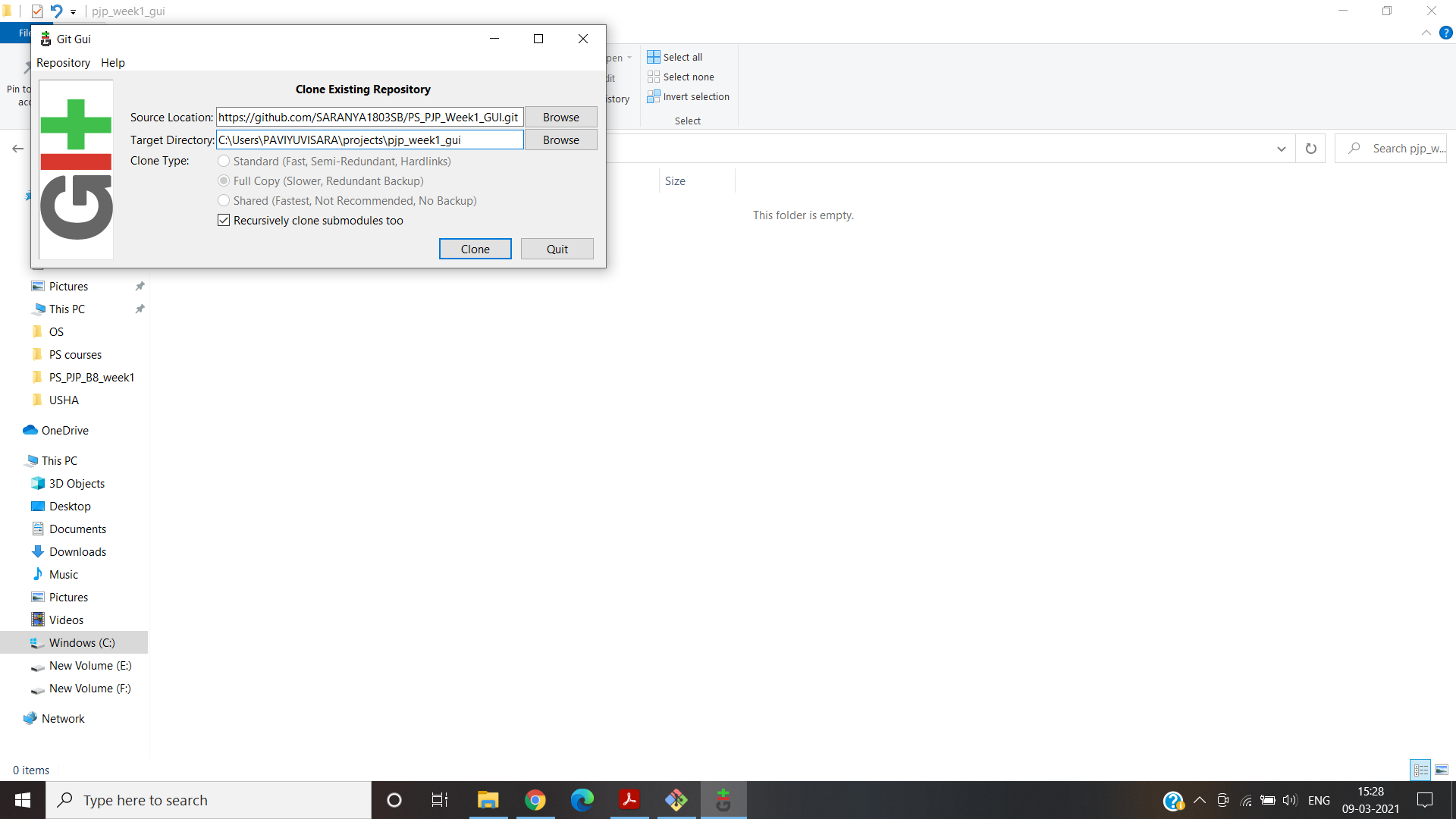
Made some changes in the text file in feature1 branch and it is commited & pushed to remote.



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

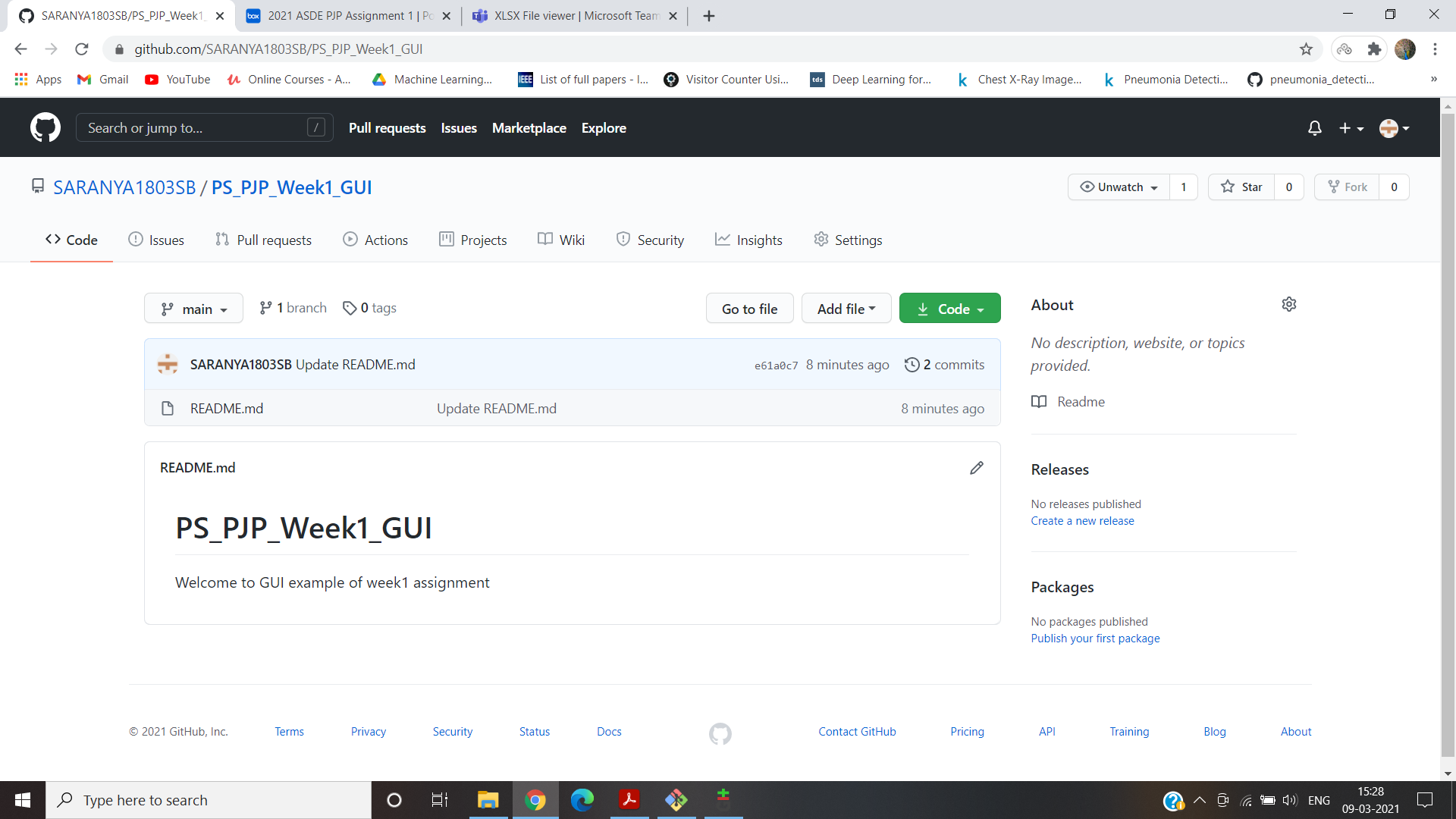
**1.Github -b. Basic usage using GUI client**

i.Local repository:

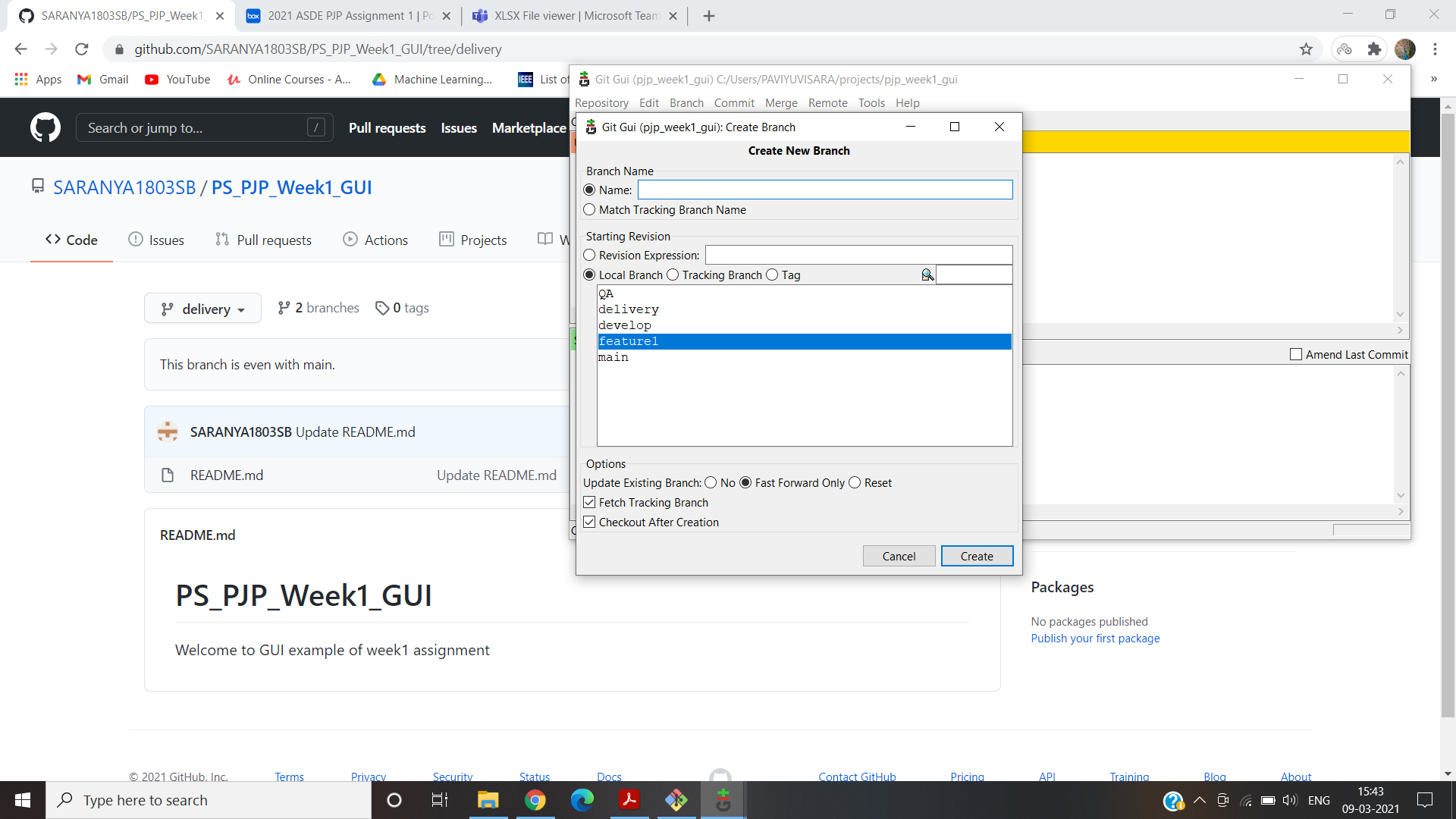


ii.Remote Repository:

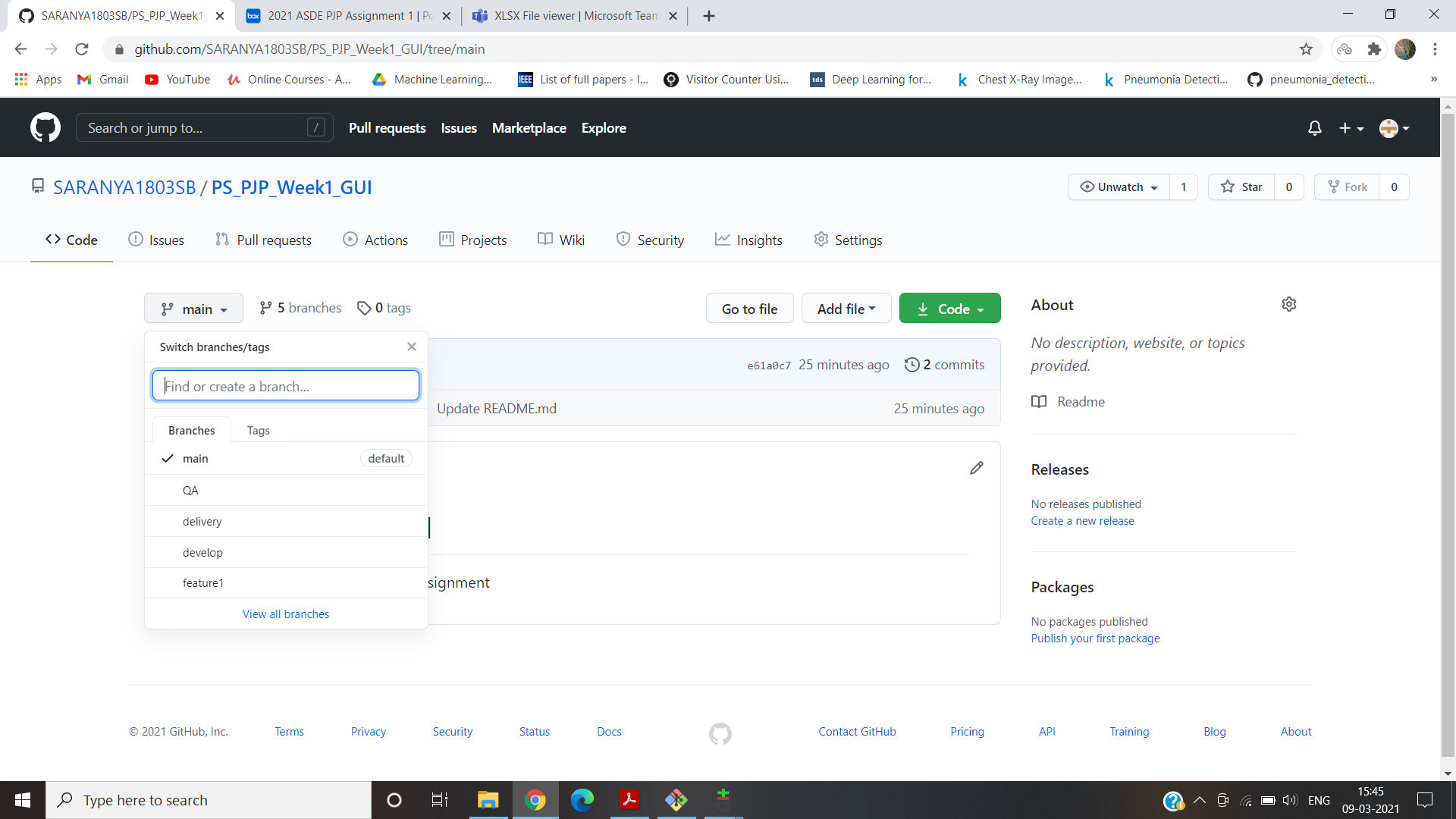
Created a new remote repository named GUI PS\_PJP\_Week1\_GUI.



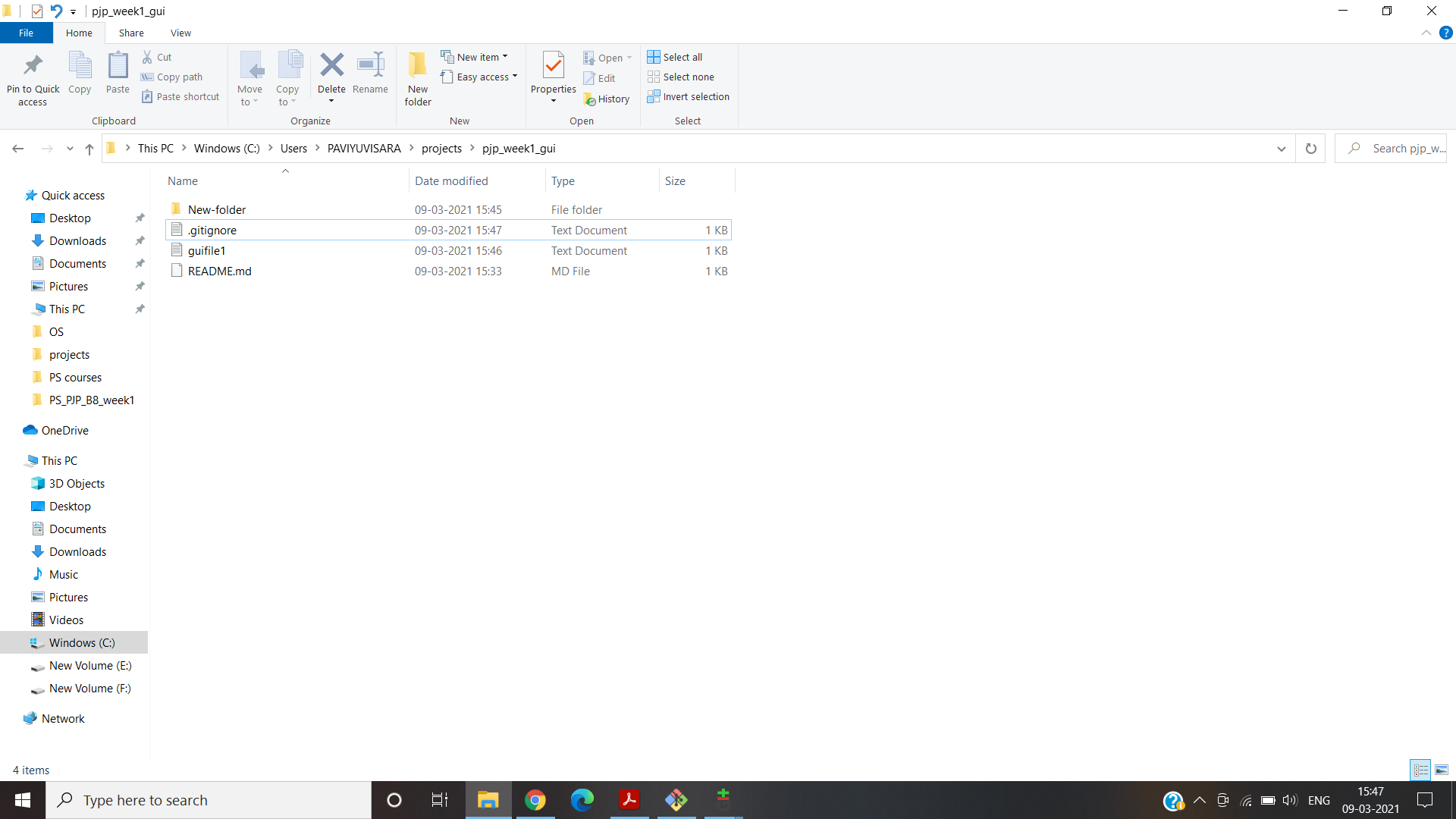
iii.Local branches:



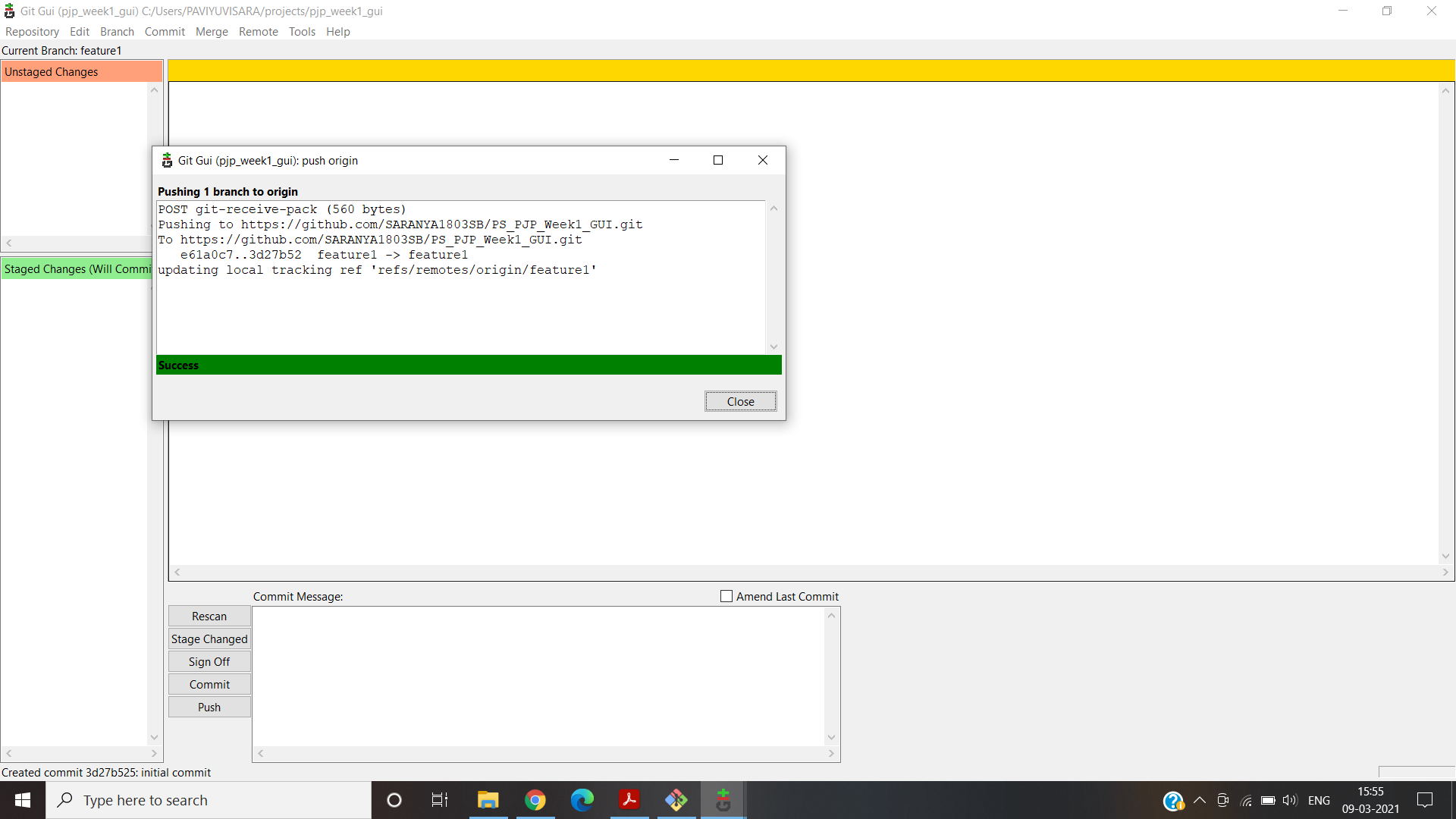
iv.Remote branches:



v.Adding files & folders:

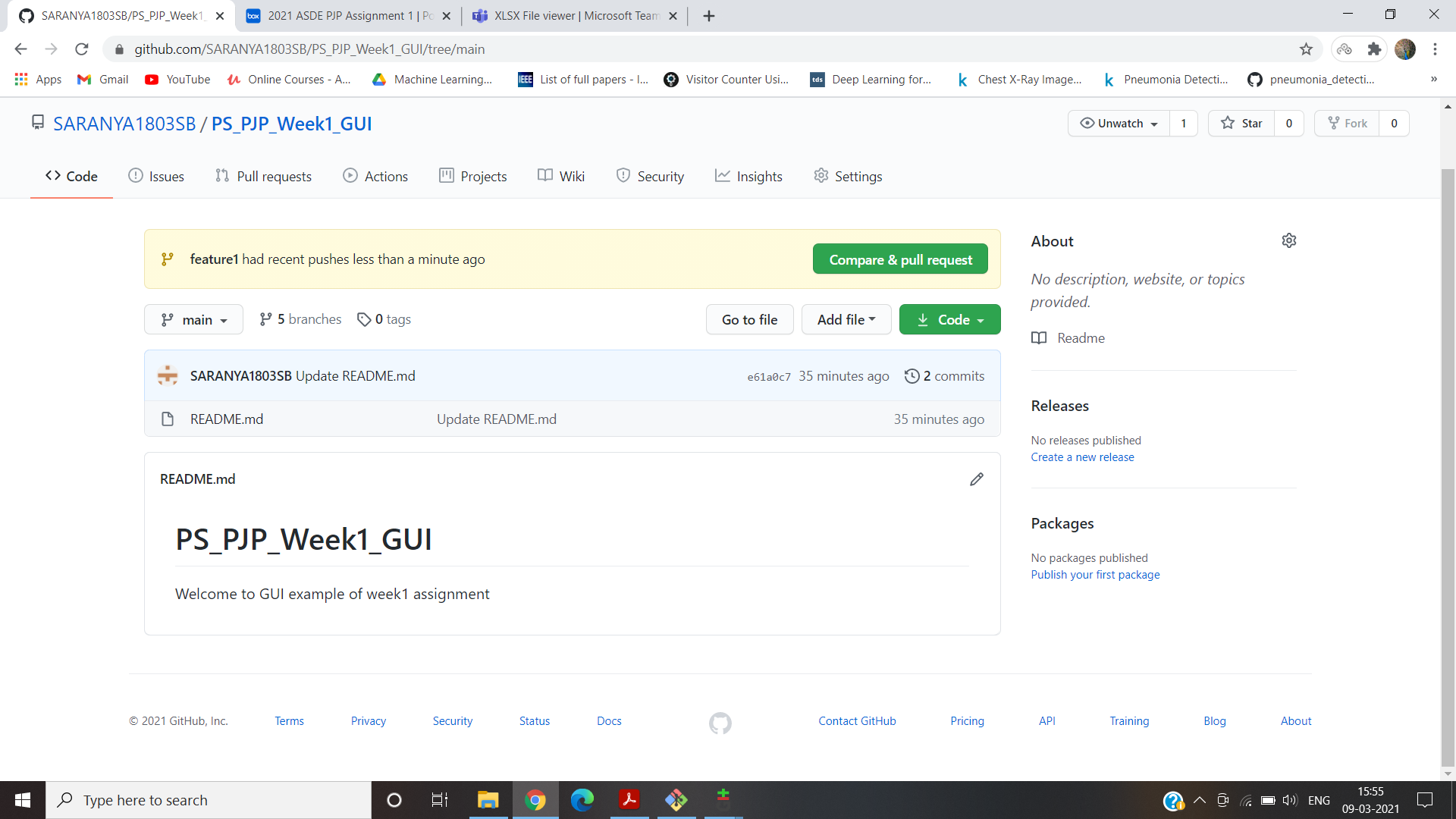


vi.Check-in ,Stage, Commit,Push into feature:

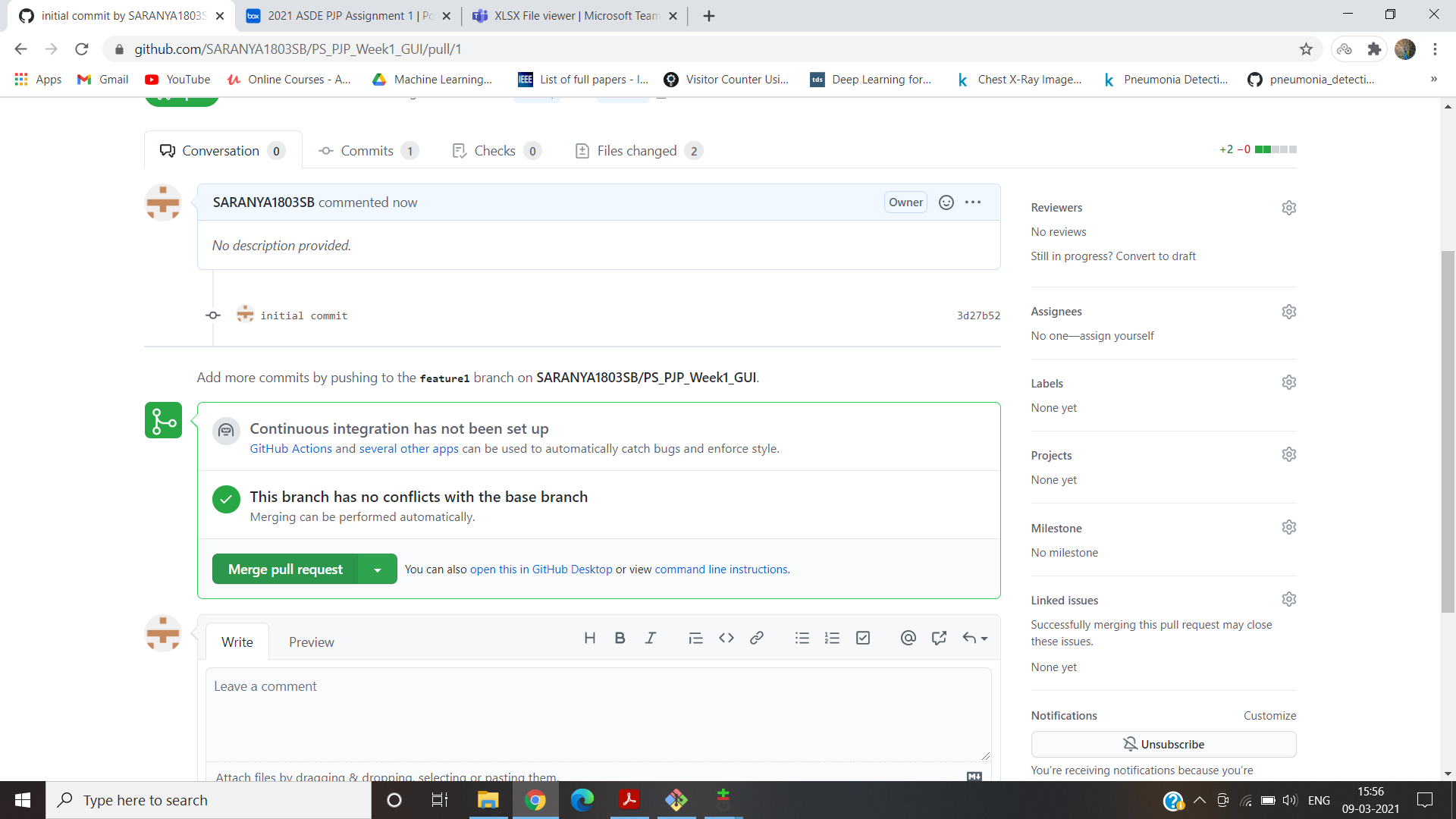


vii.Promote code from feature to develop:

Click “Compare & pull request”:

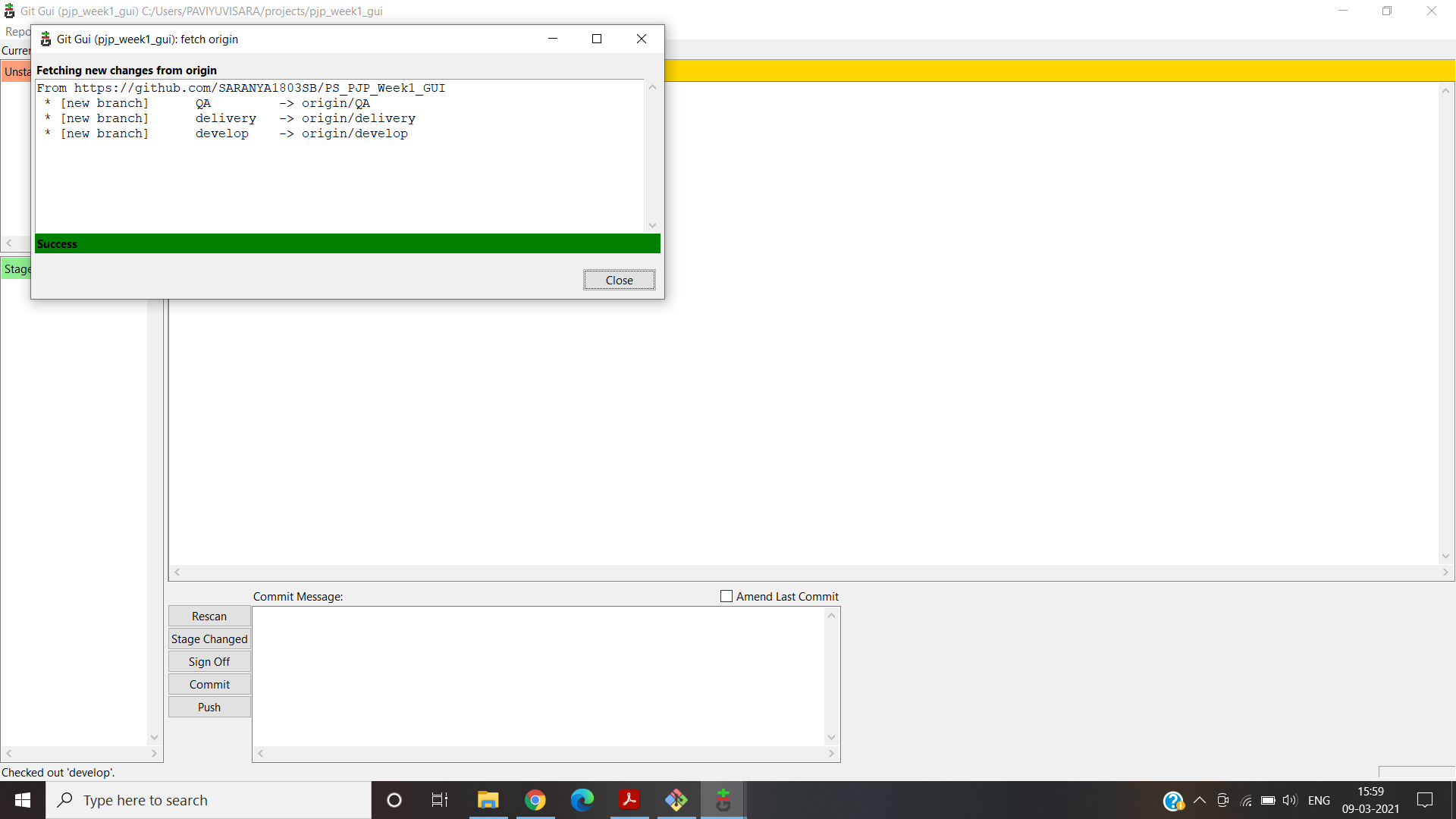


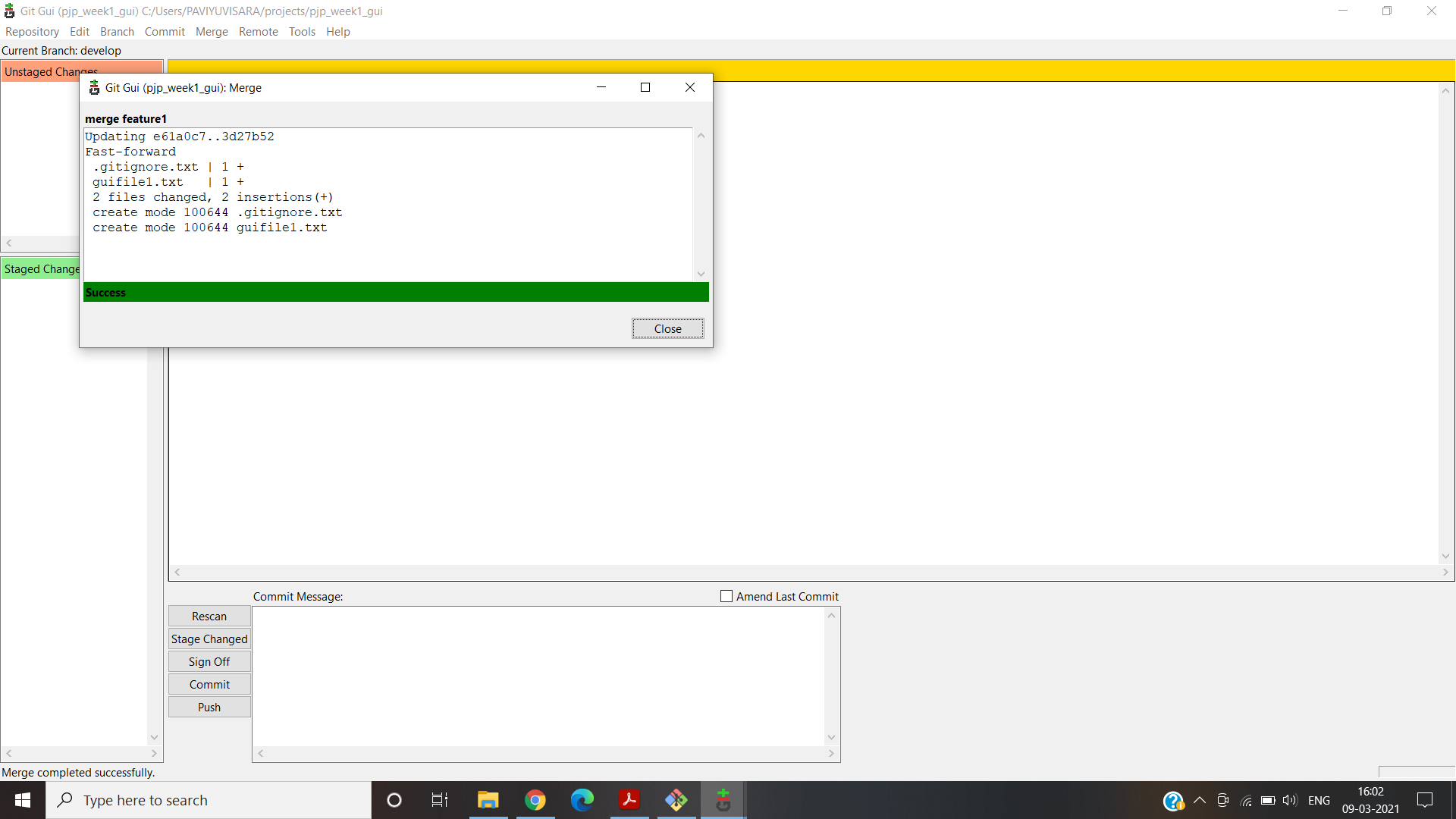
Click “merge pull request”:



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.

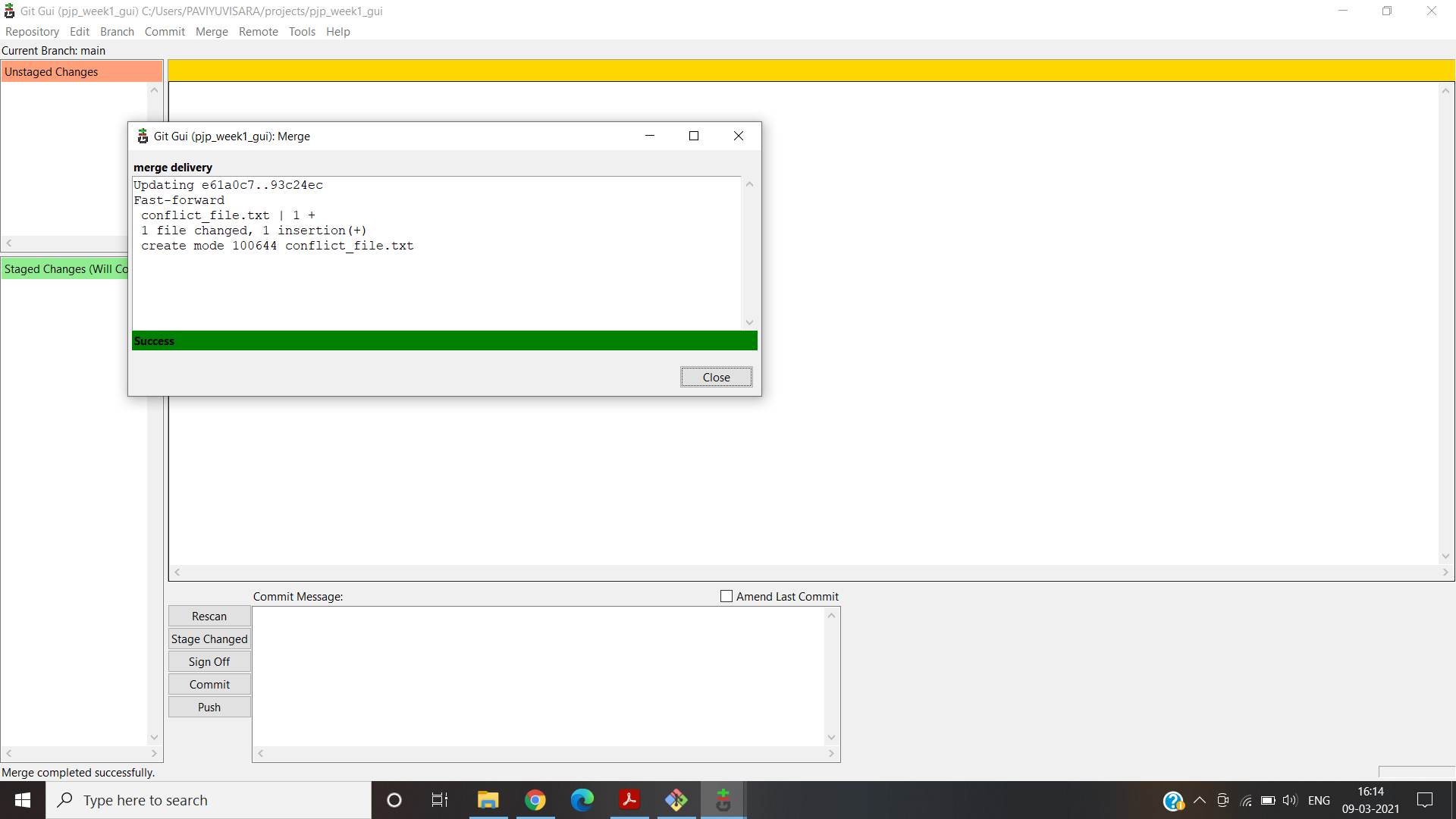




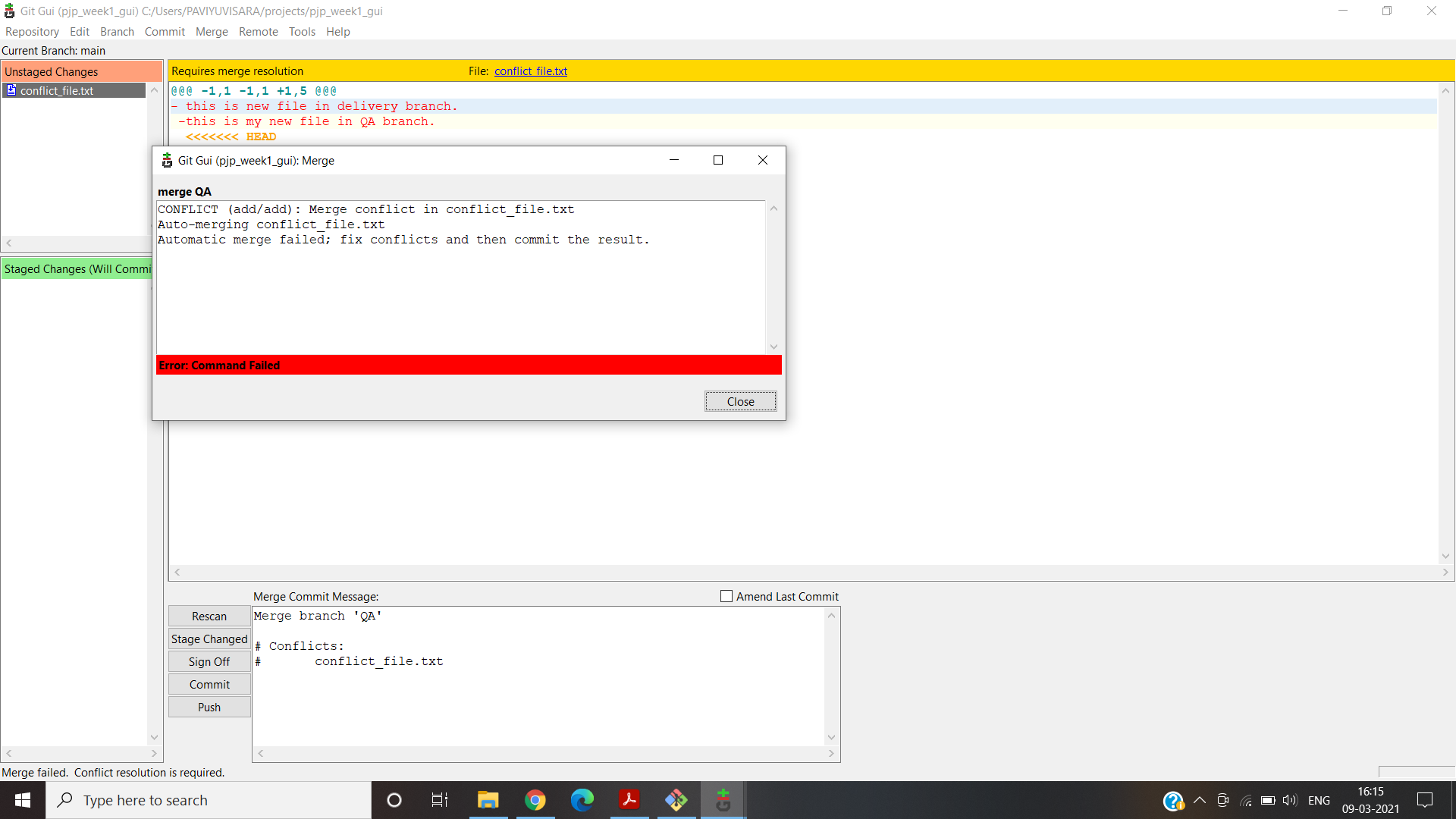
x.Handle Merge Conflict:

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.

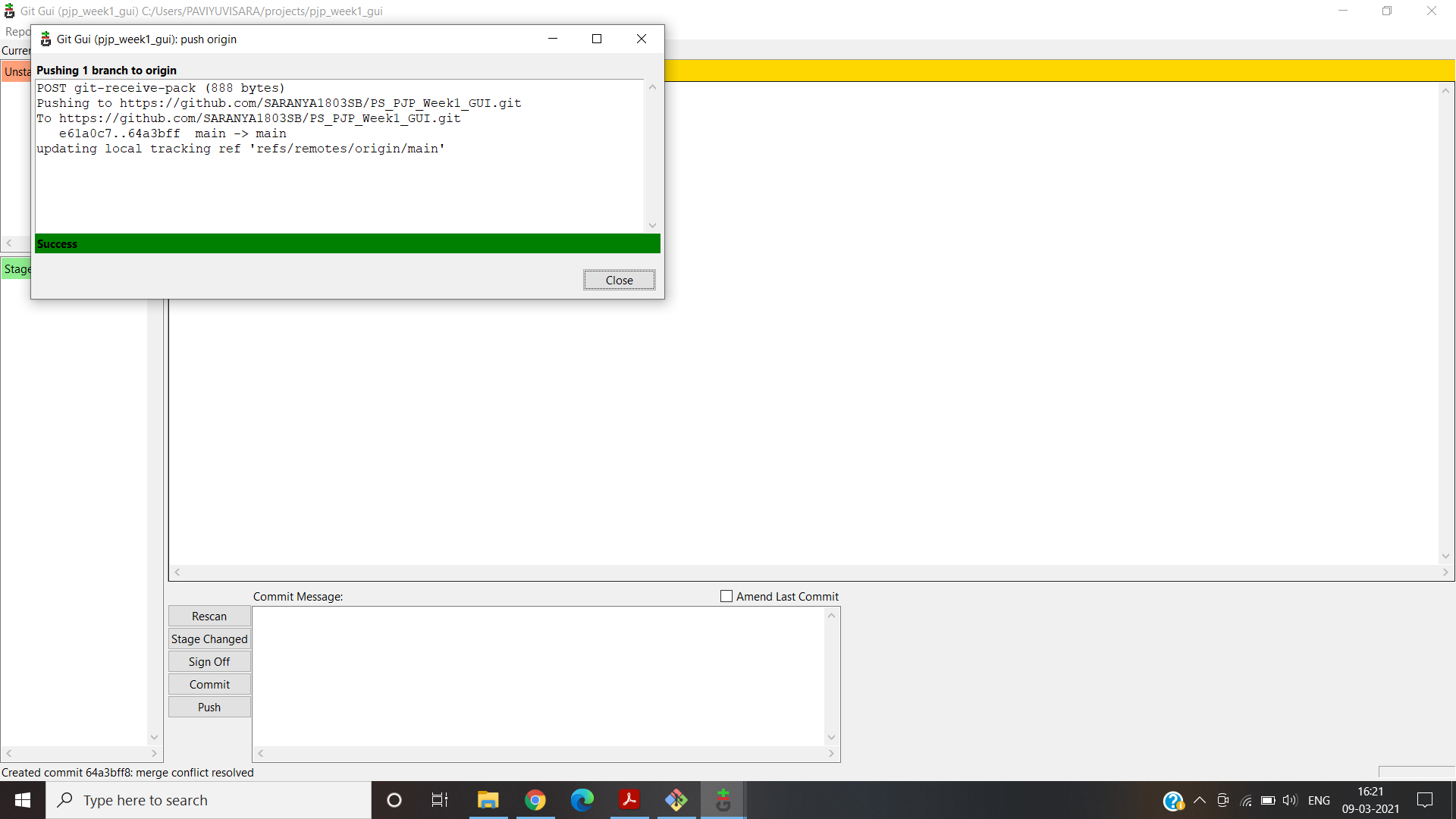
First merge to main with branch delivery



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



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

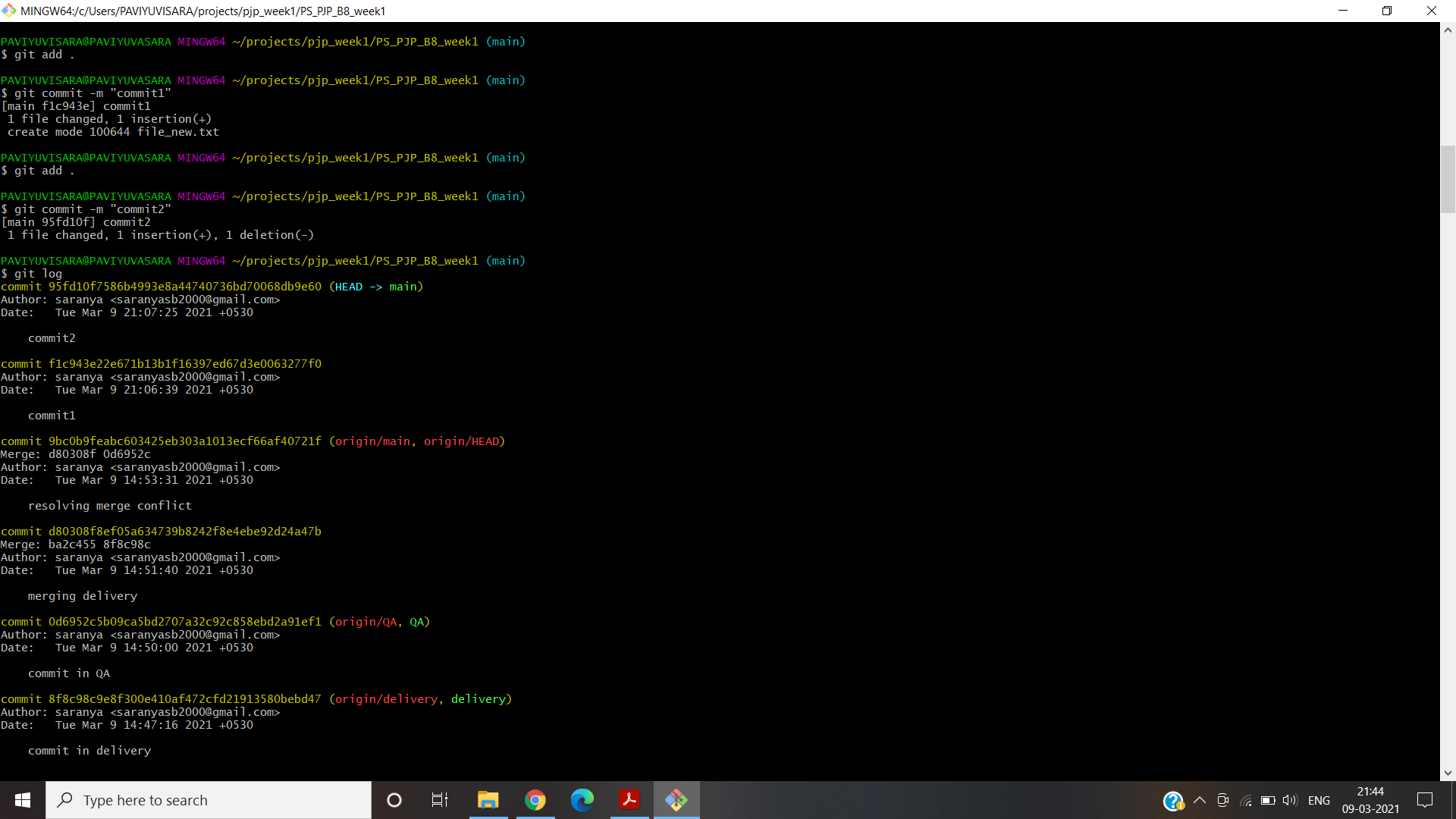


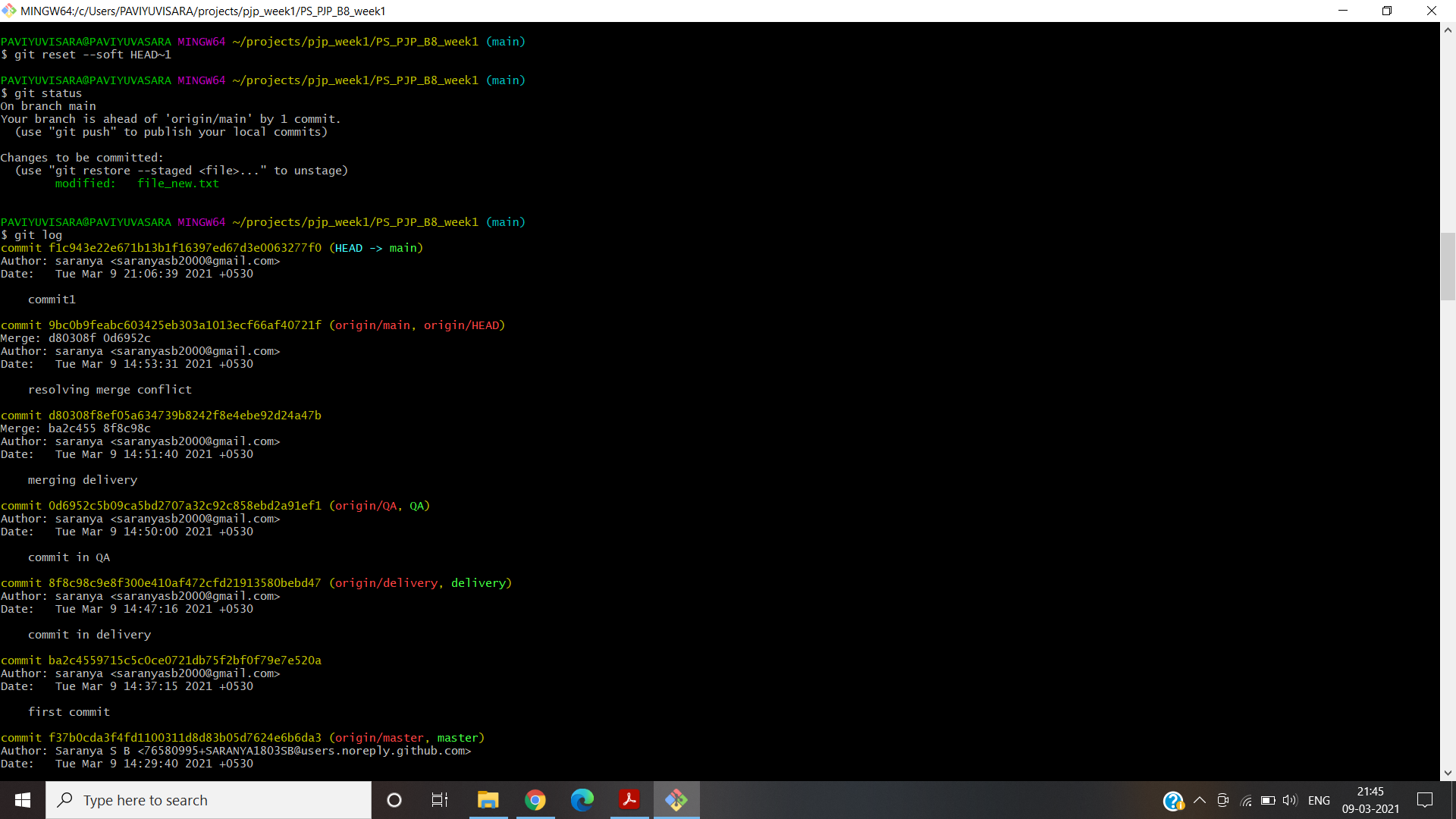
xi.Ensure the code is in sync .with latest changes across all branches:

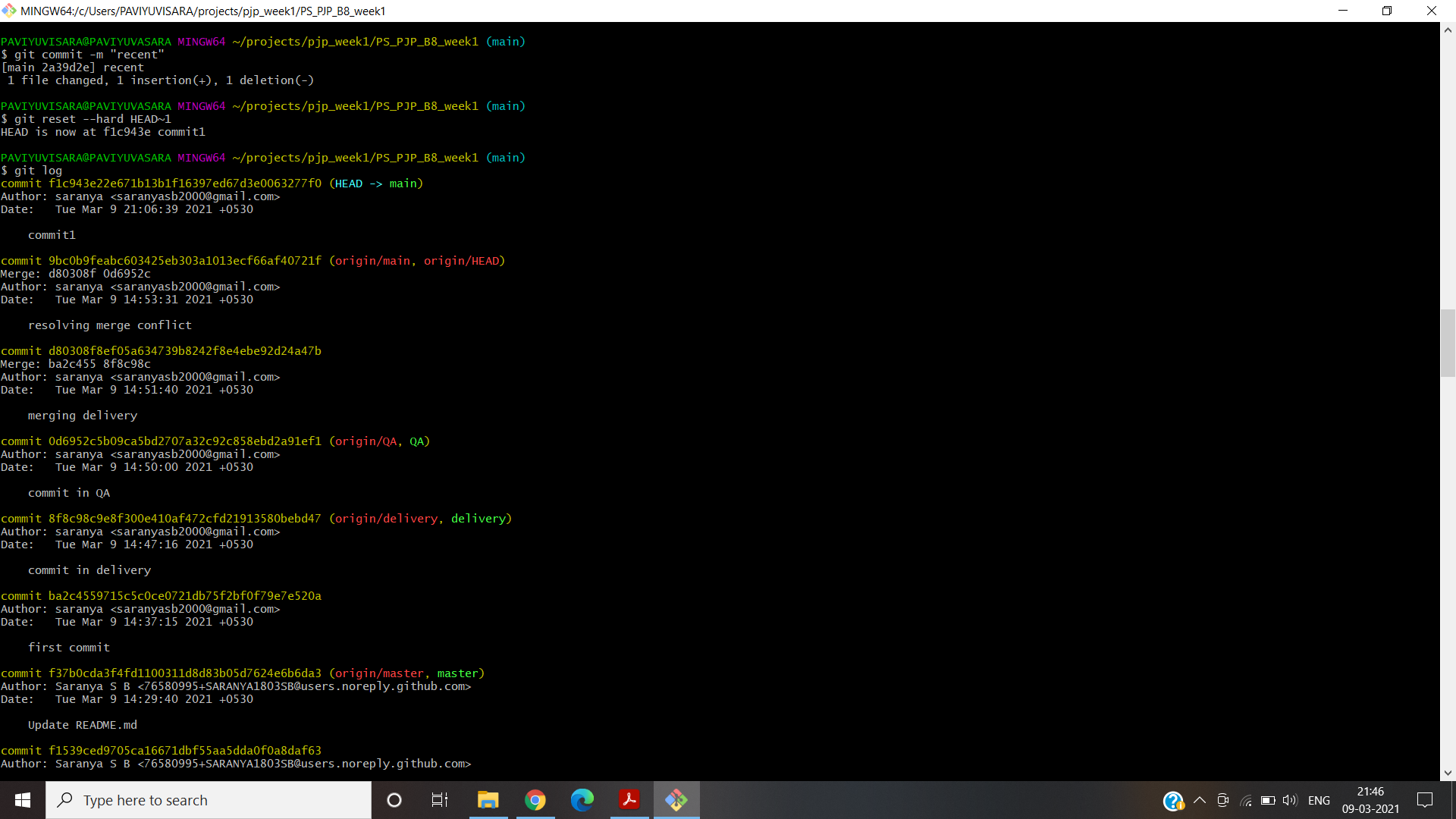
All File are updated to main.

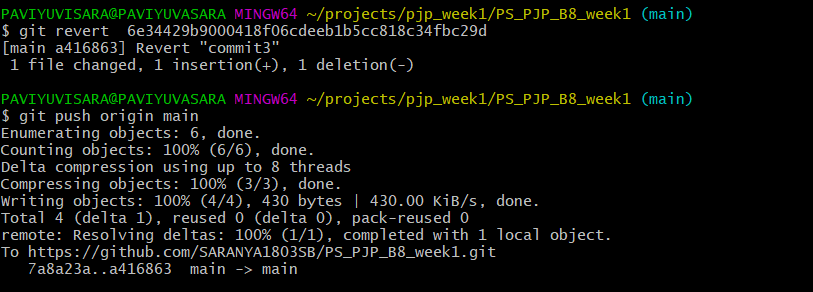
**1.c.On the ground day to day scenario:**

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

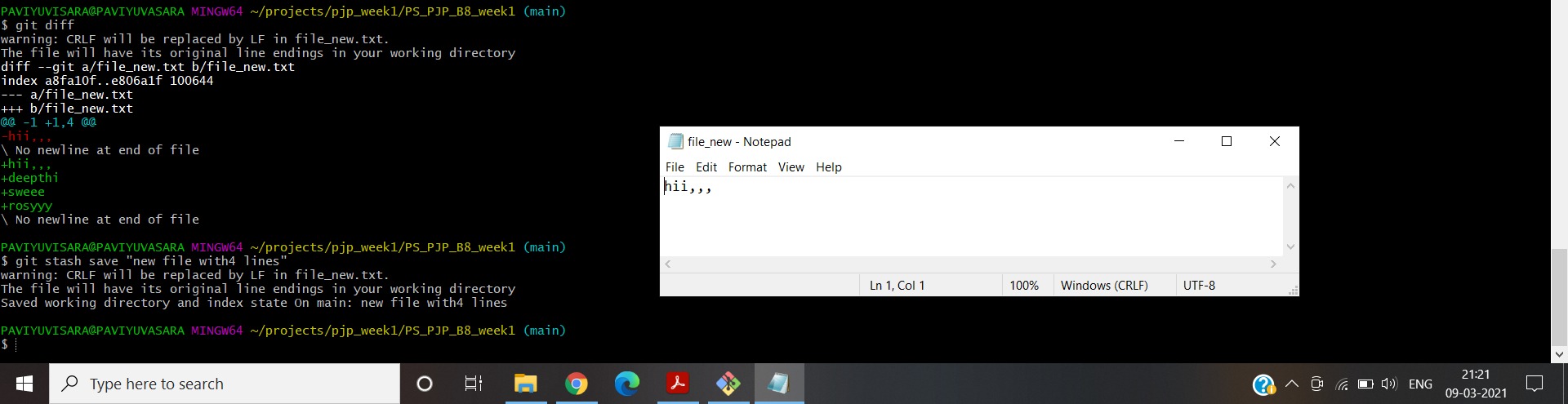


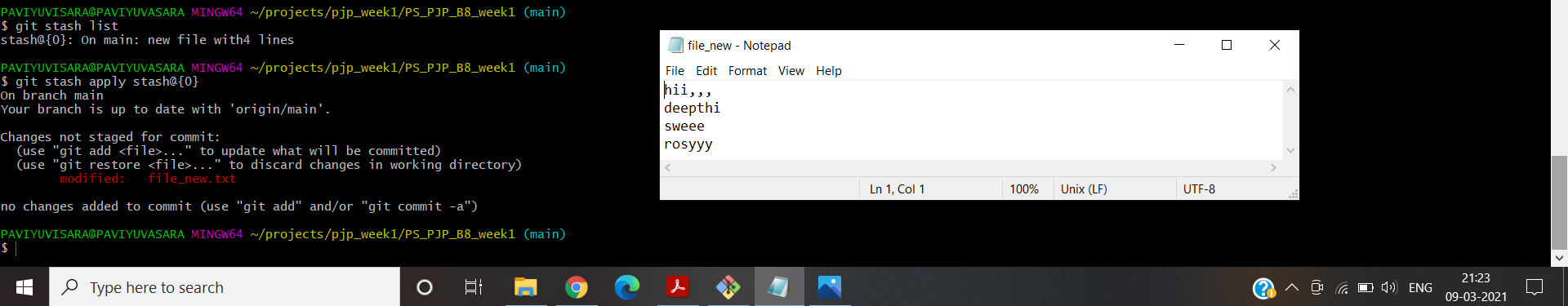




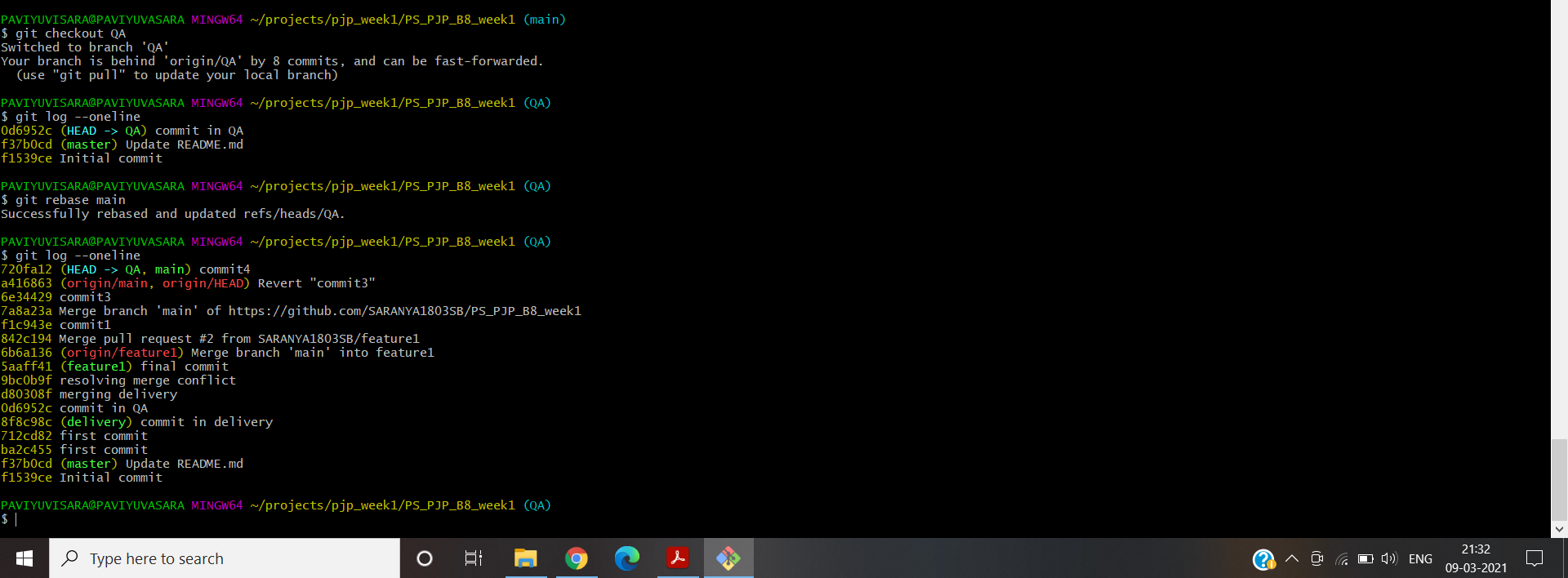


ii.Stash local changes during merge conflicts:





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



iv.git log,status & reflog:

