**Resolving merge conflicts in Github**

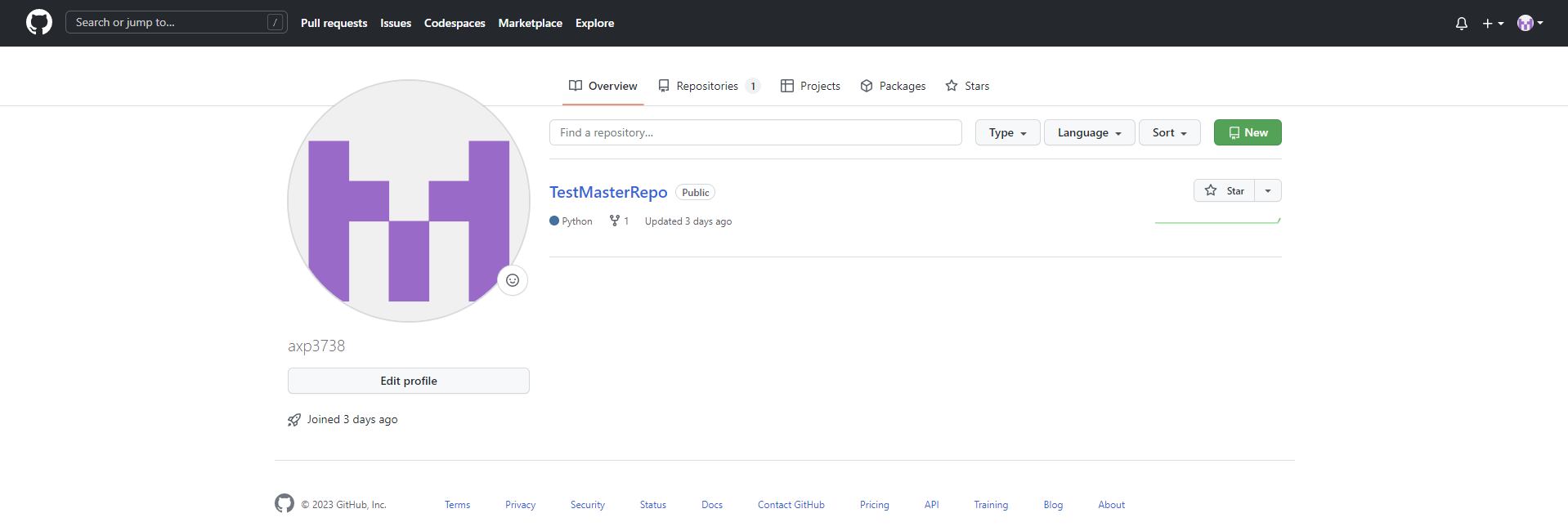
Steps followed:

User 1 – axp3738

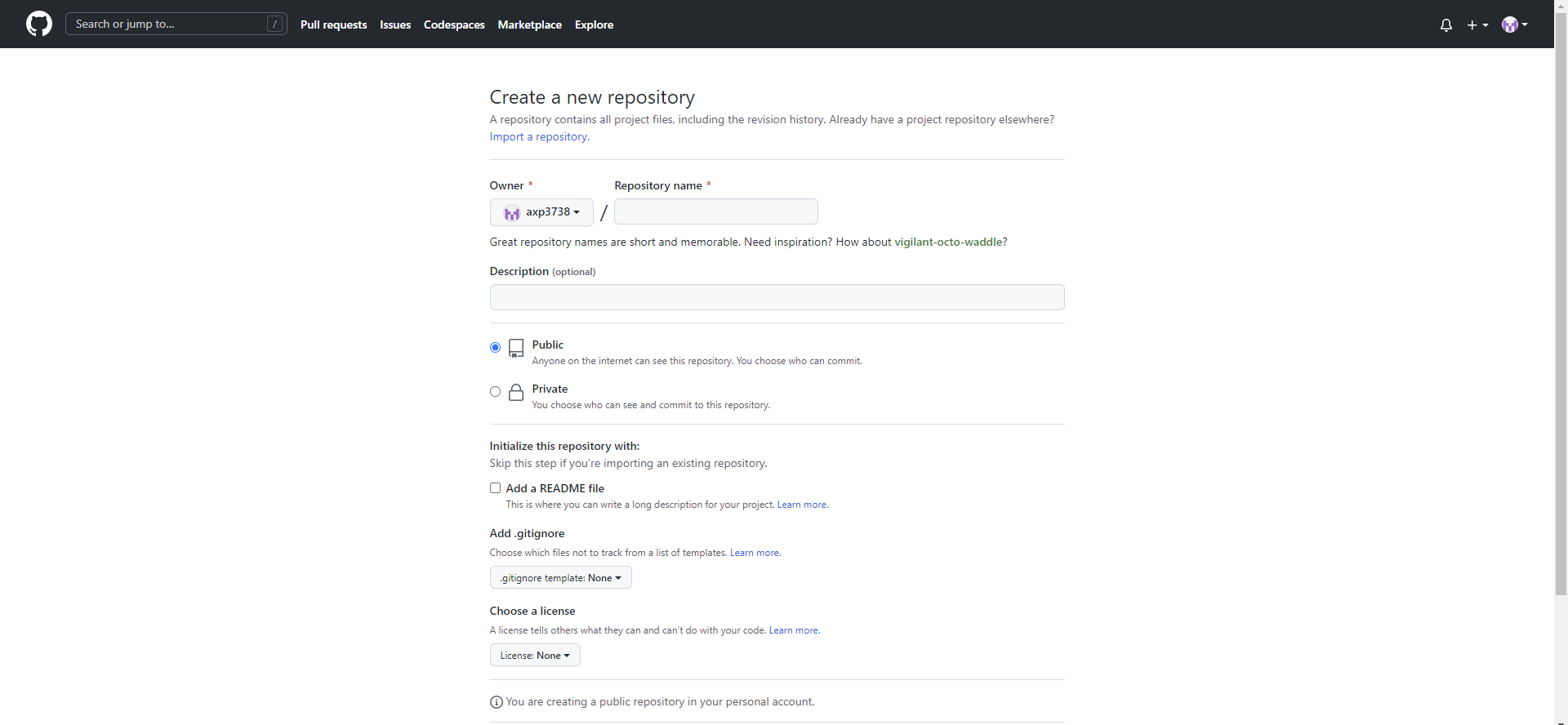
1. Create 1 github account, follow steps to sign up and create a new user ID - axp3738

<https://github.com/join>

1. Go to github account and create a public repository named - TestMasterRepo, under that create one python file named – function1.py



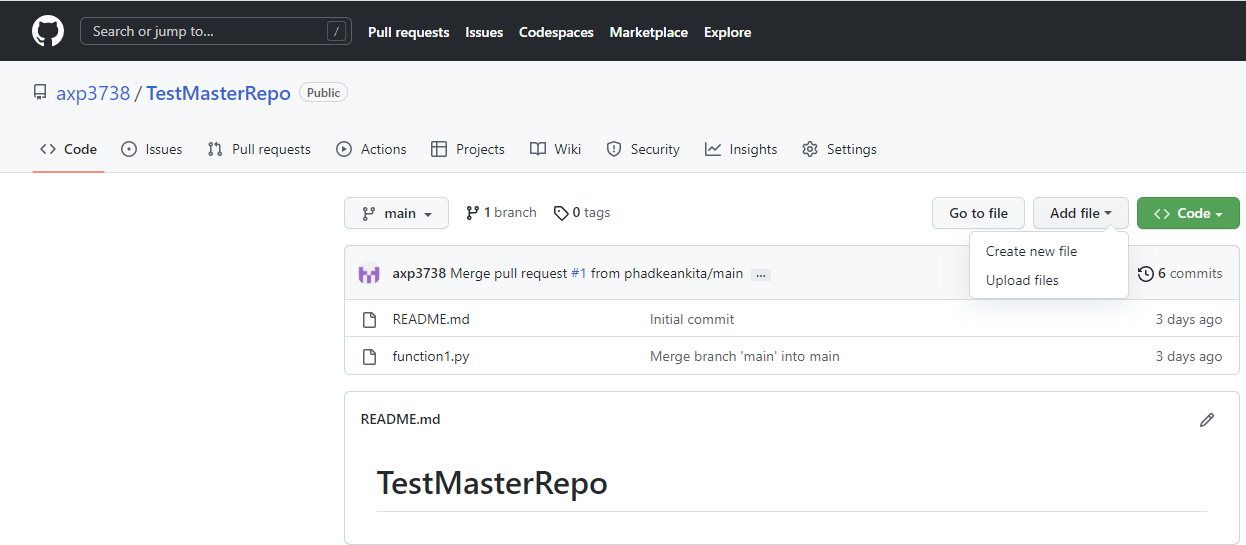




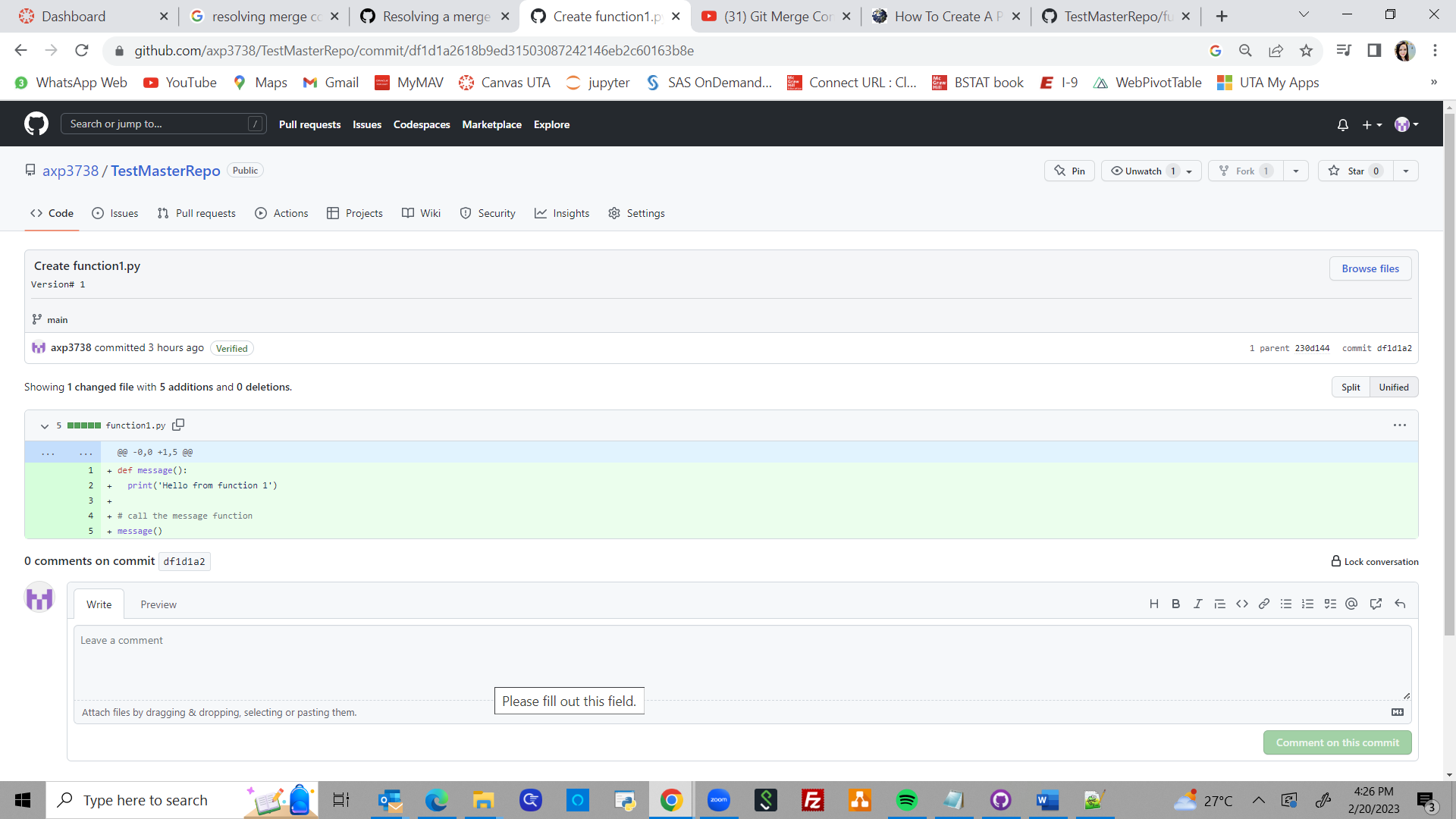


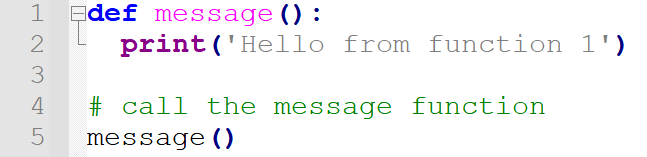
1. The code tab shows 2 files, namely: README.md and function1.py:

function1.py shows below code initially:











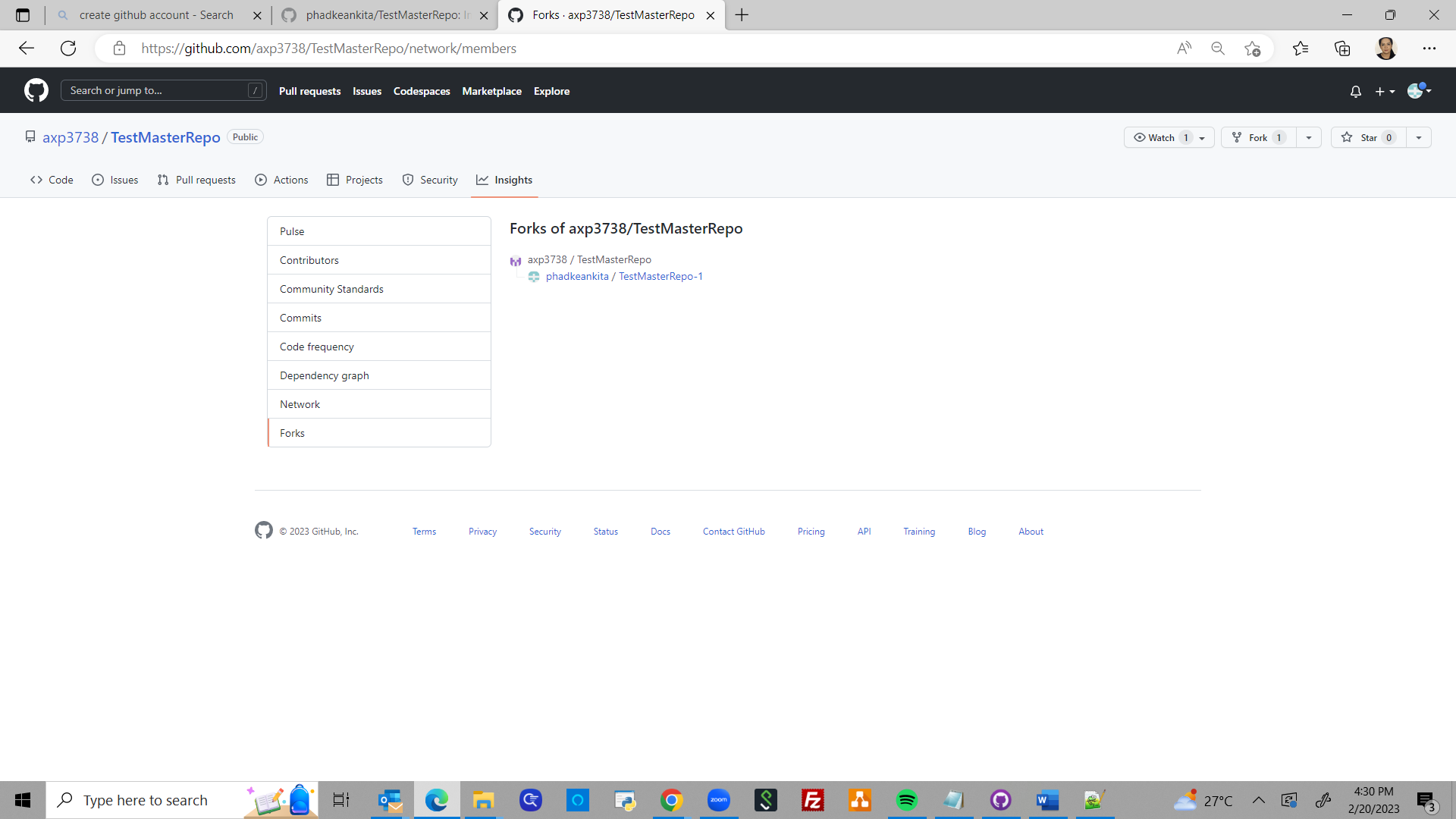
User 2 - phadkeankita

1. Create another github account, follow steps to sign up and create a new user ID – phadkeankita
2. Open the public repository created by user axp3738 – i.e., repository - TestMasterRepo, and create a fork – option available on right top of the page, name the fork as TestMasterRepo-1

After creation of fork, this is what the screen looks like

For axp3738/TestMasterRepo repository – fork is created as phadkeankita/TestMasterRepo-1

Public link to access repository: <https://github.com/axp3738/TestMasterRepo>



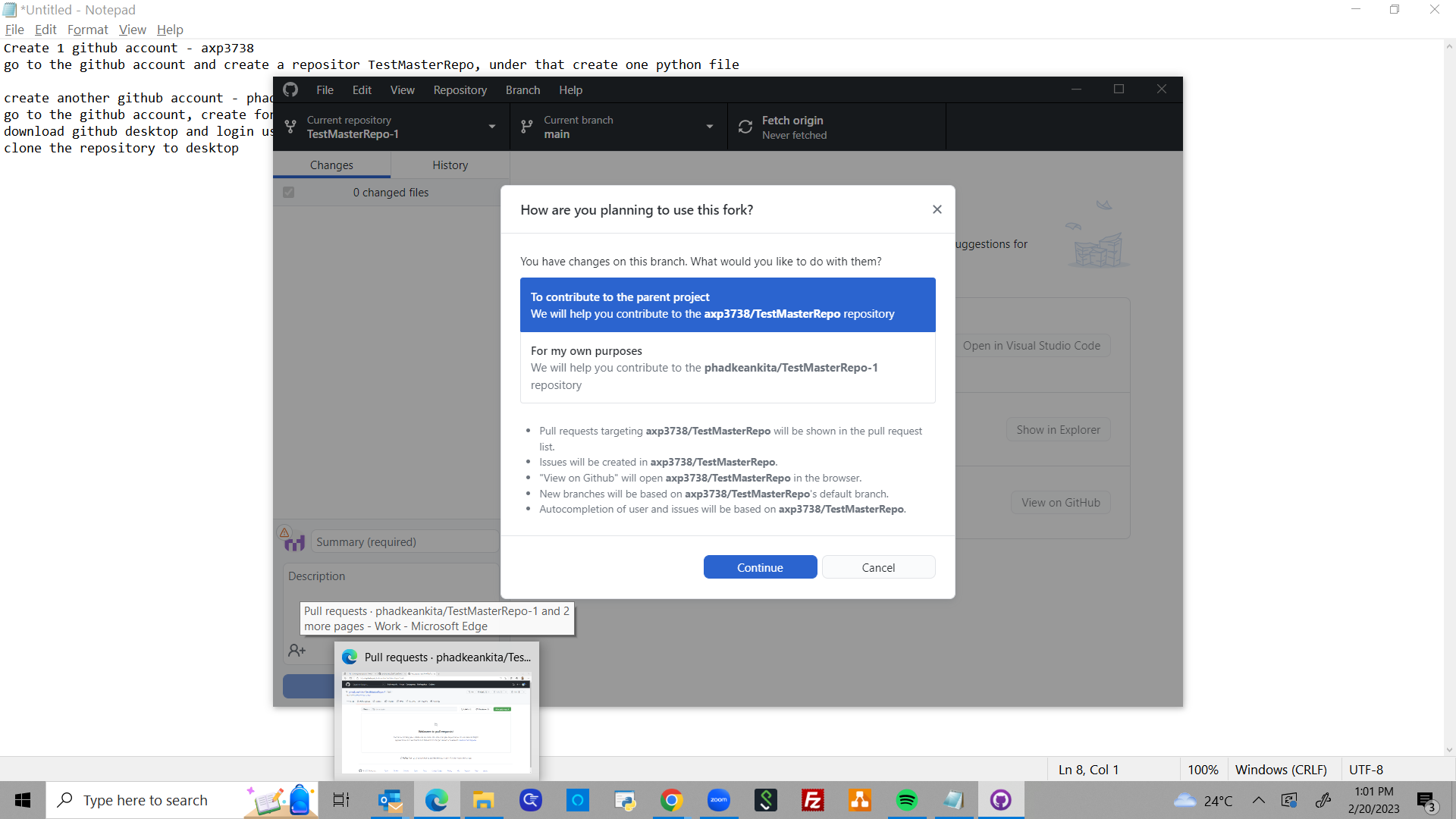


Steps on how merge conflict gets created and how to resolve the merge conflict:

* The phadkeankita user has created a fork to repository of user axp3738. The phadkankita user has all the latest code from axp3738’s repository. Now, phadkeankita user tries to make a change to the code file and commit the change in main repository, TestMasterRepo.
* However, while this is being done, another user (say axp3738) comes and commits a code change in same file/same function in same file
* In such case, when phadkeankita user tries to commit the code change, he/she runs into a merge conflict as the code changed by him/her has already been changed in original repository

Steps to resolve the merge conflict:

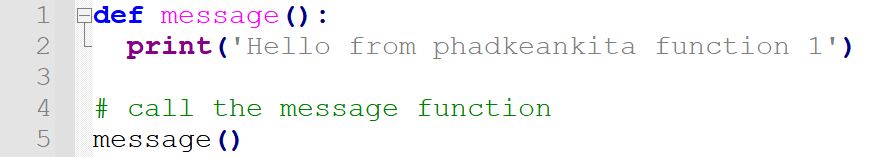
1. When creating the fork from phadkeankita’s login, select the option to: contribute changes to parent project as below





1. Make changes to the python file and save. This change is currently only there in phadkeankita’s local repository – TestMasterRepo-1

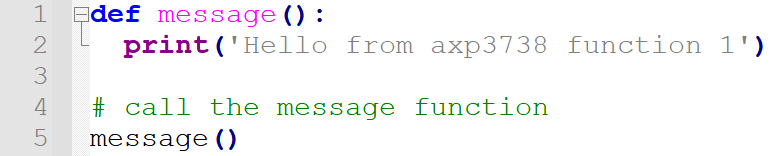
This is how the code change looks like:





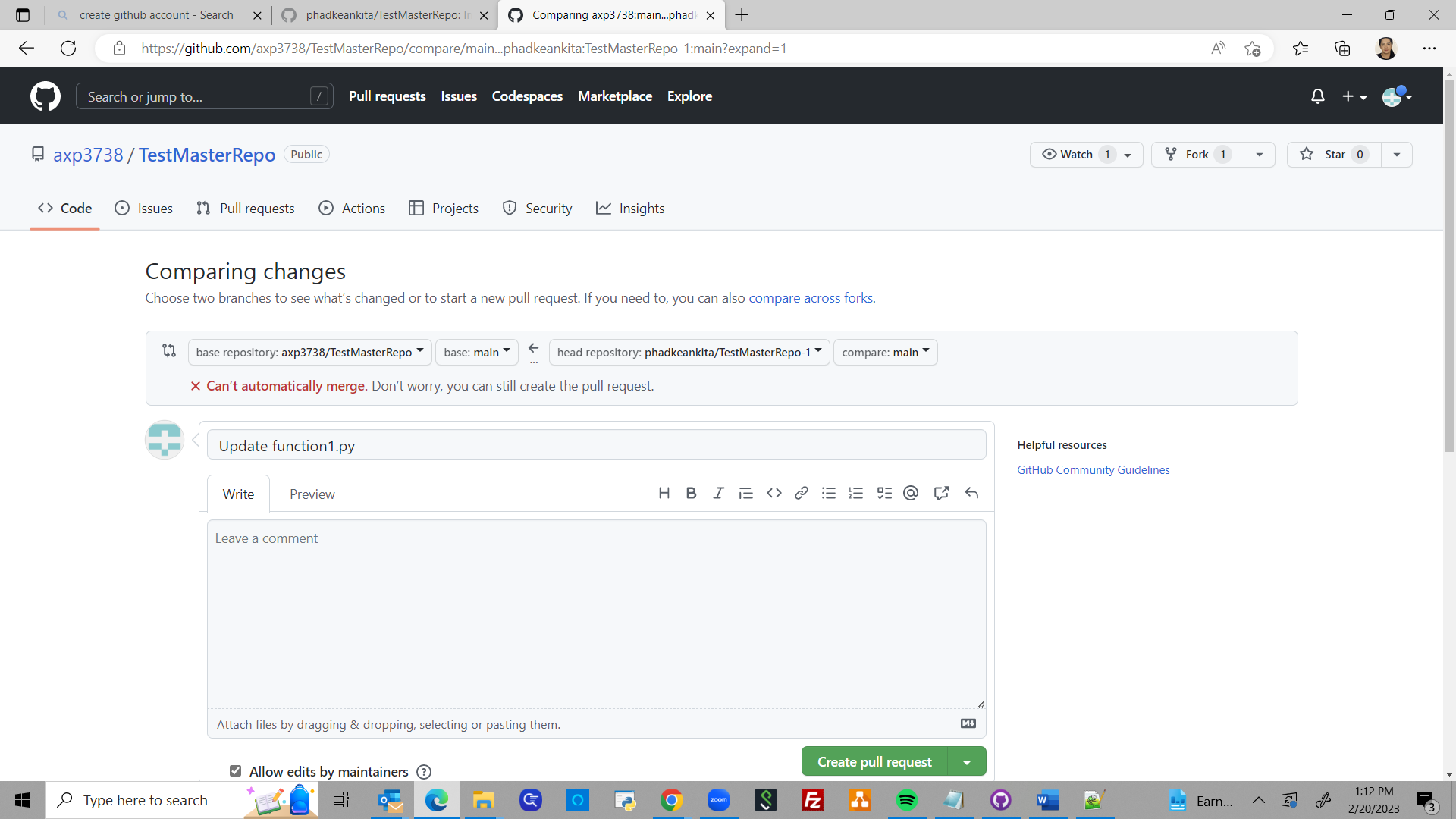
1. Now, after this code change has been saved in the local repository TestMasterRepo-1, the other user axp3738 has made a code change in the main repository TestMasterRepo and saves this code

This is how the code change looks like:



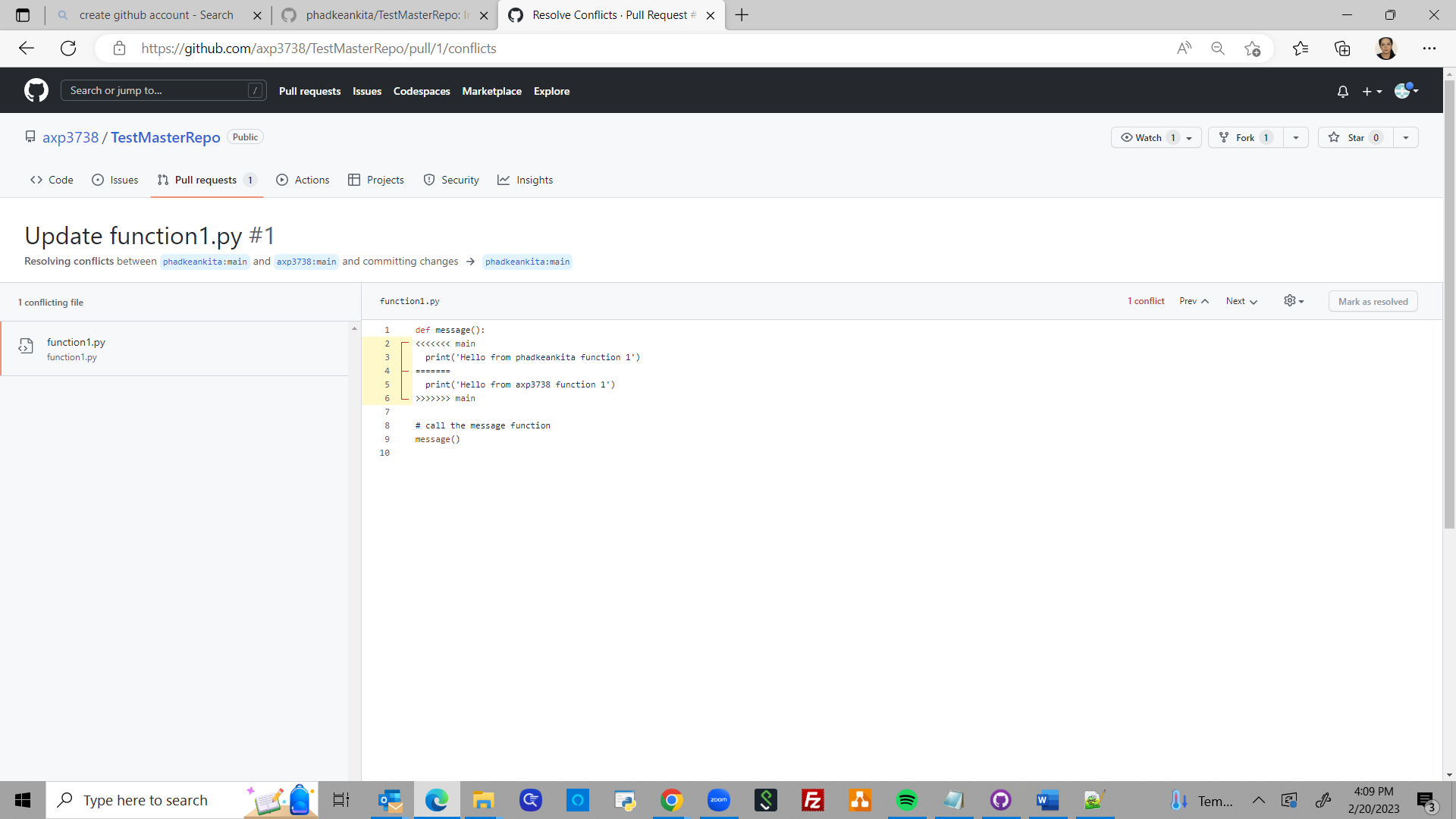


1. Now, unaware of this change, when phadkeankita user tries to commit their code change, below screen is shown, which shows that the code change can’t be merged



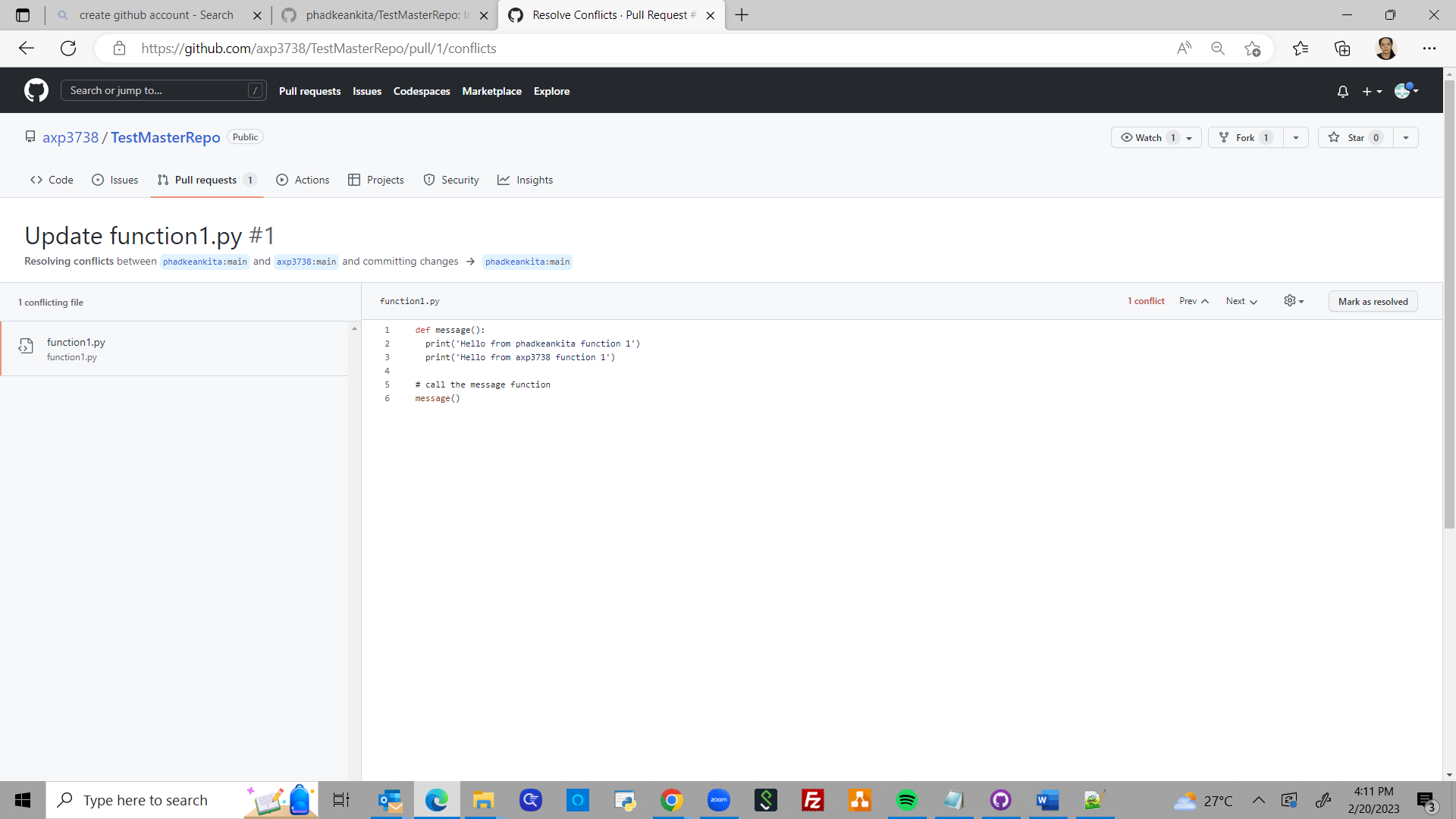


1. Click on Resolve Conflicts option
2. The code window shows where the merge conflict has occurred



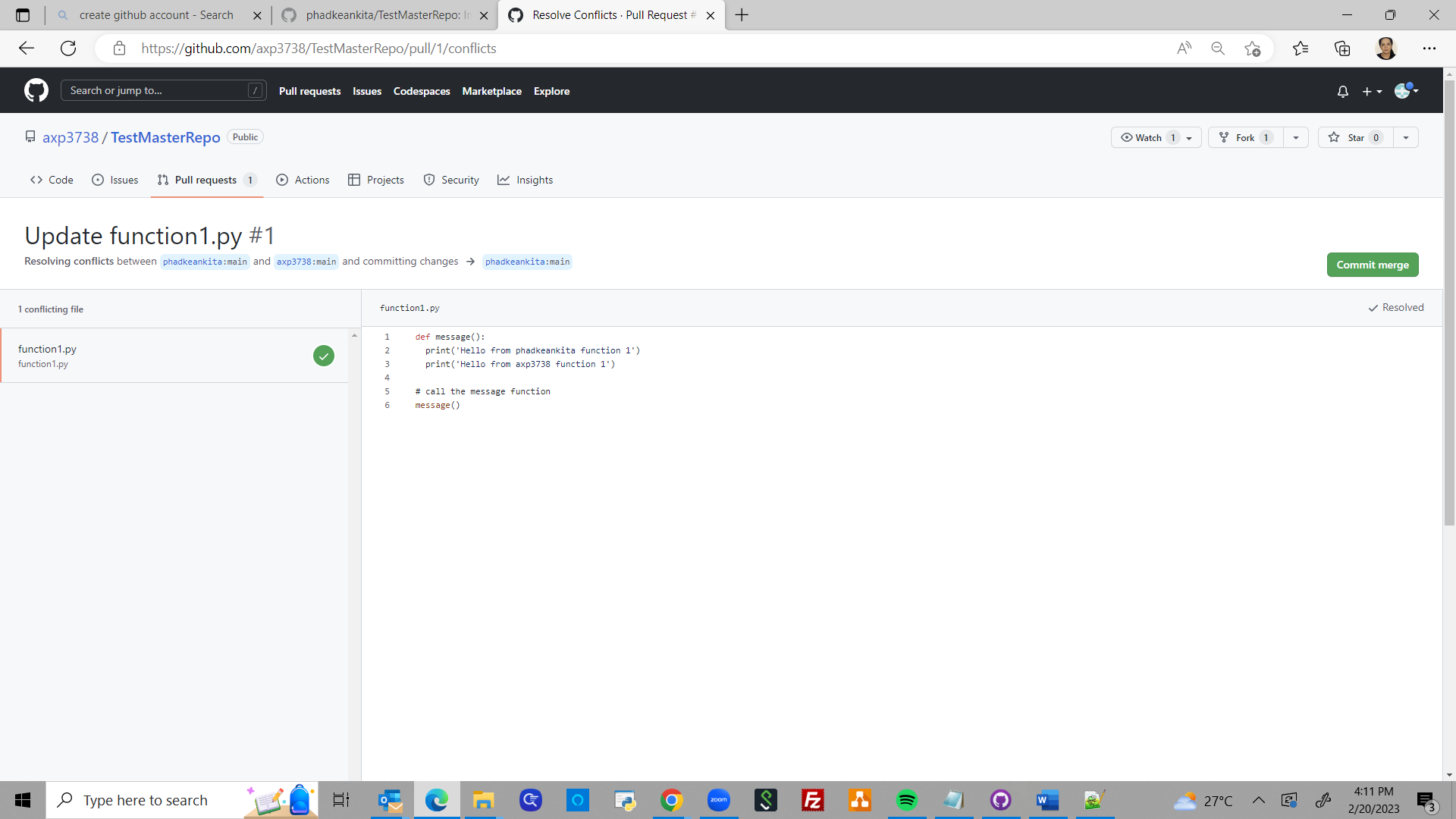


1. The code is corrected by phadkeankita and saved. Click on Mark as Resolved



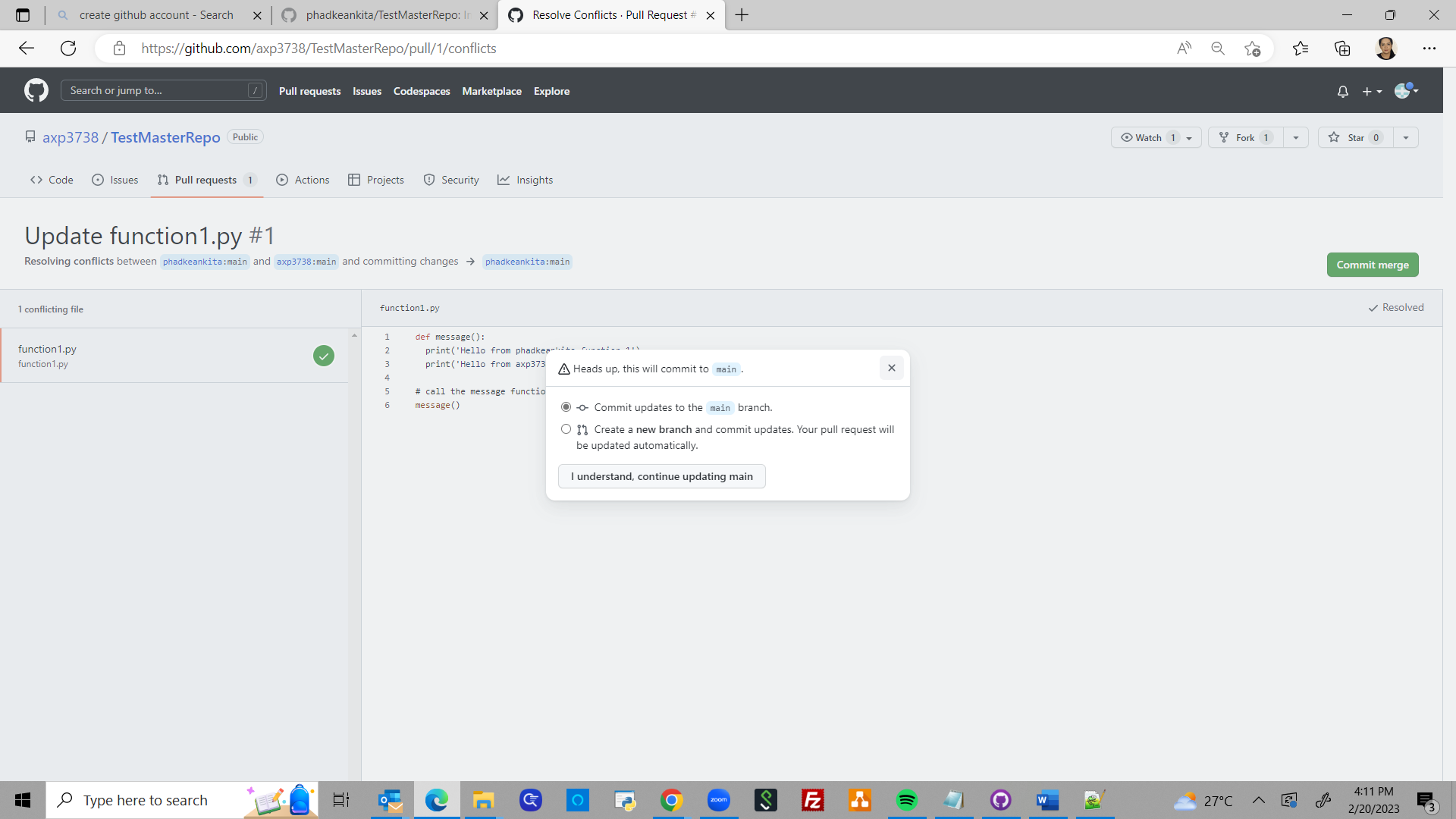


1. Once the changes are finalized, click on Commit merge



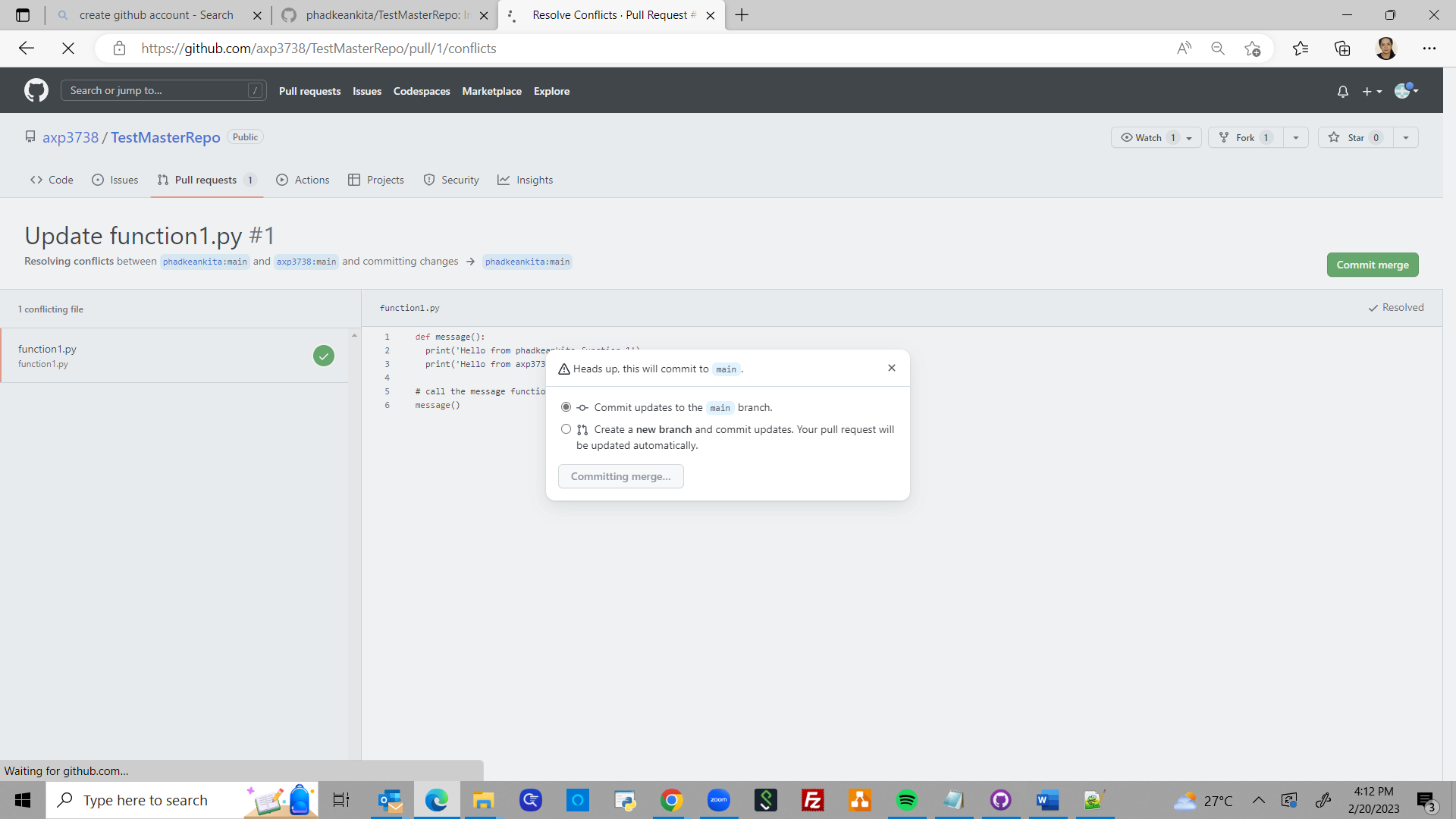


1. Click on I understand, continue updating main, to push the changes into axp3738 user’s original repository



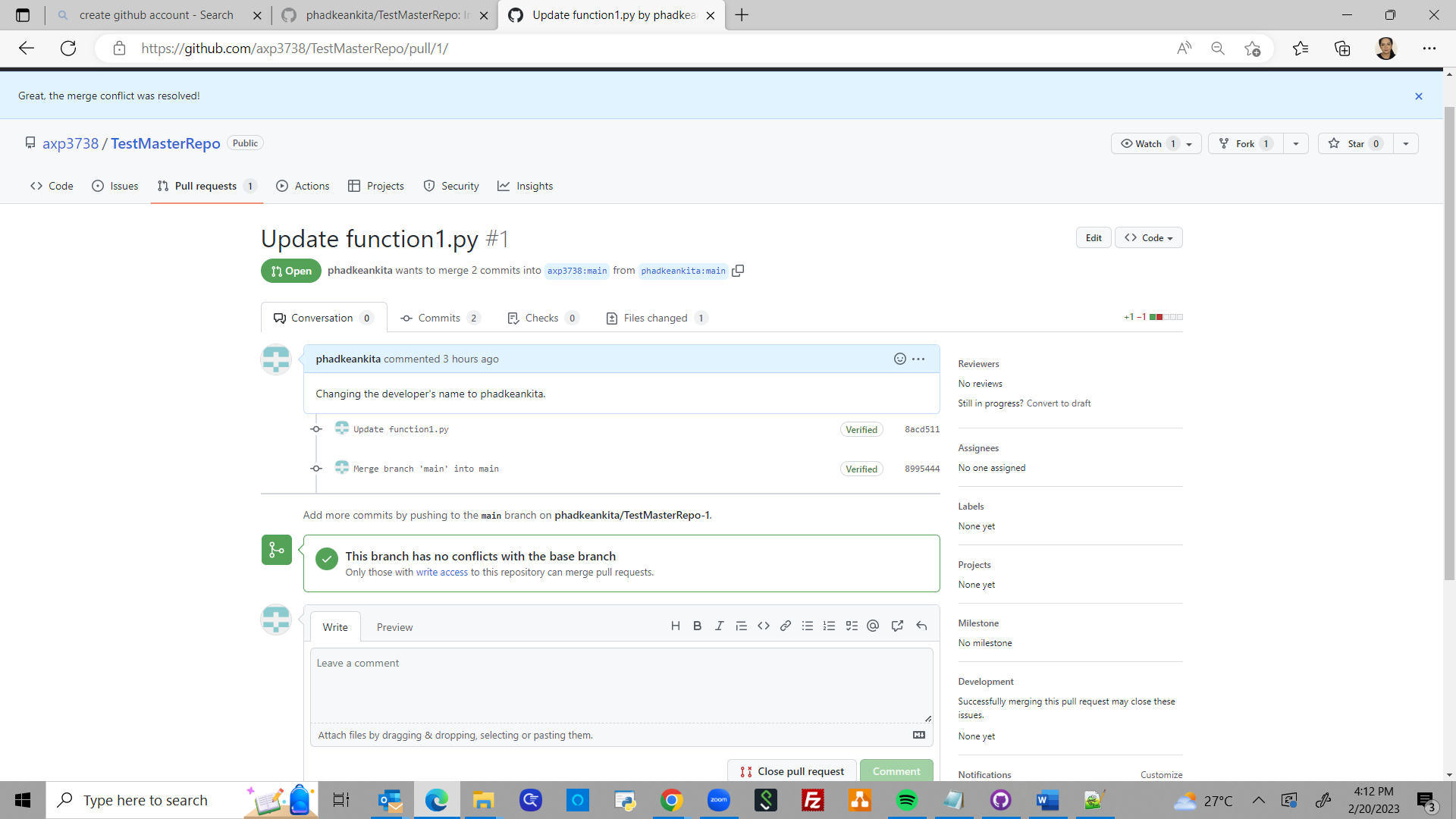


1. The changes are being committed



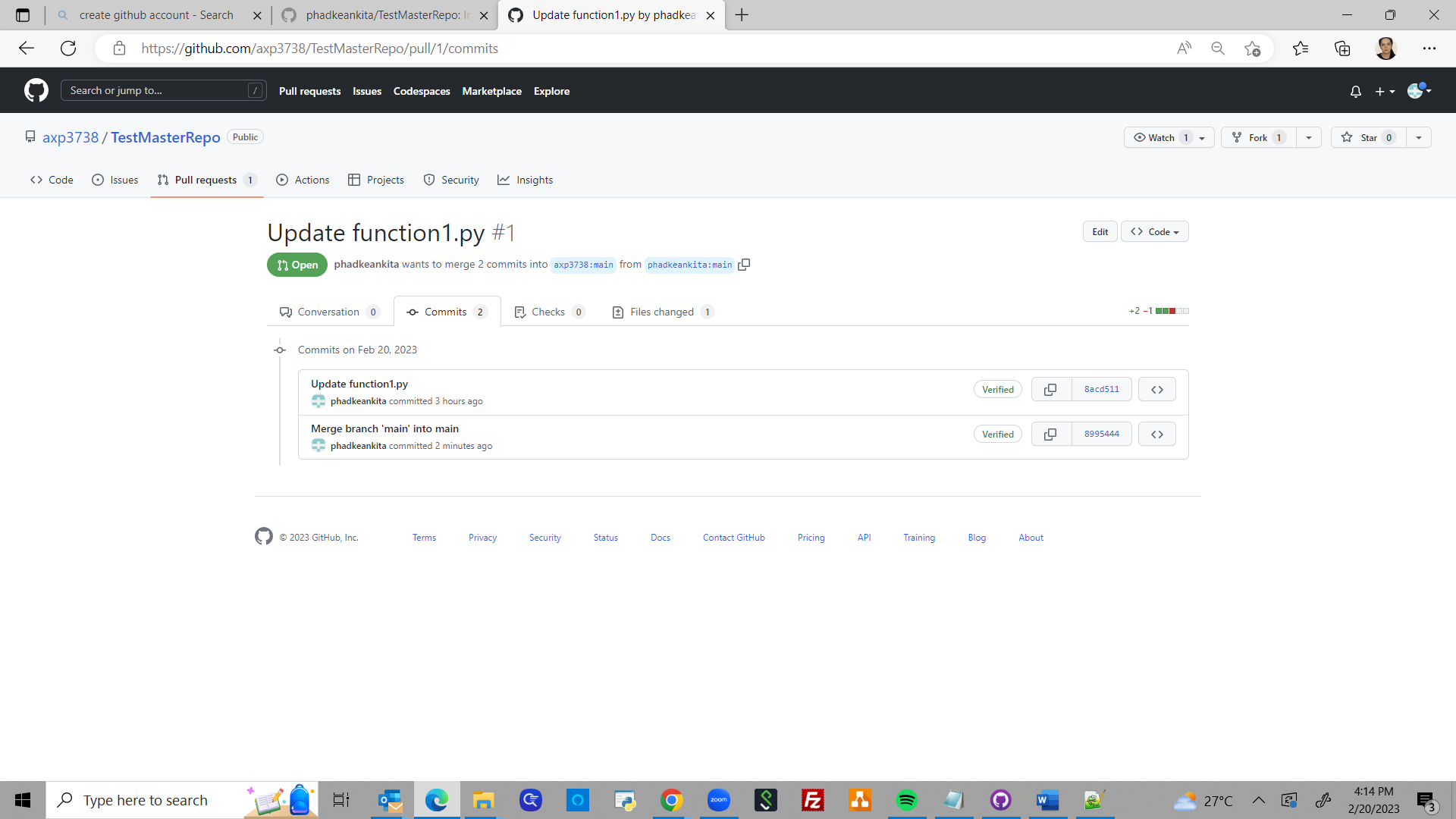


1. Shows confirmation once the changes have been committed



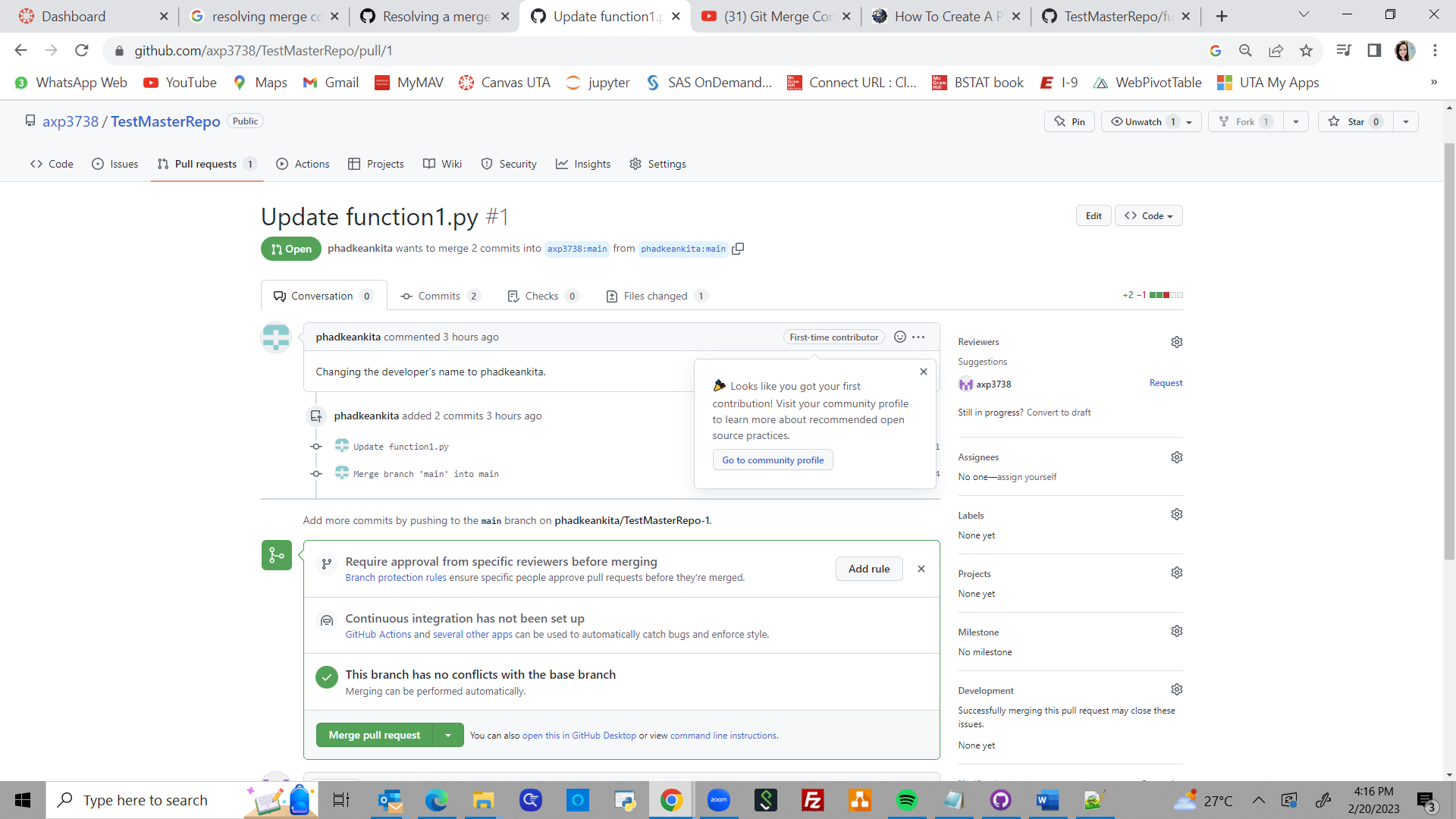


1. Below is the activity happening at axp3738 user’s repository



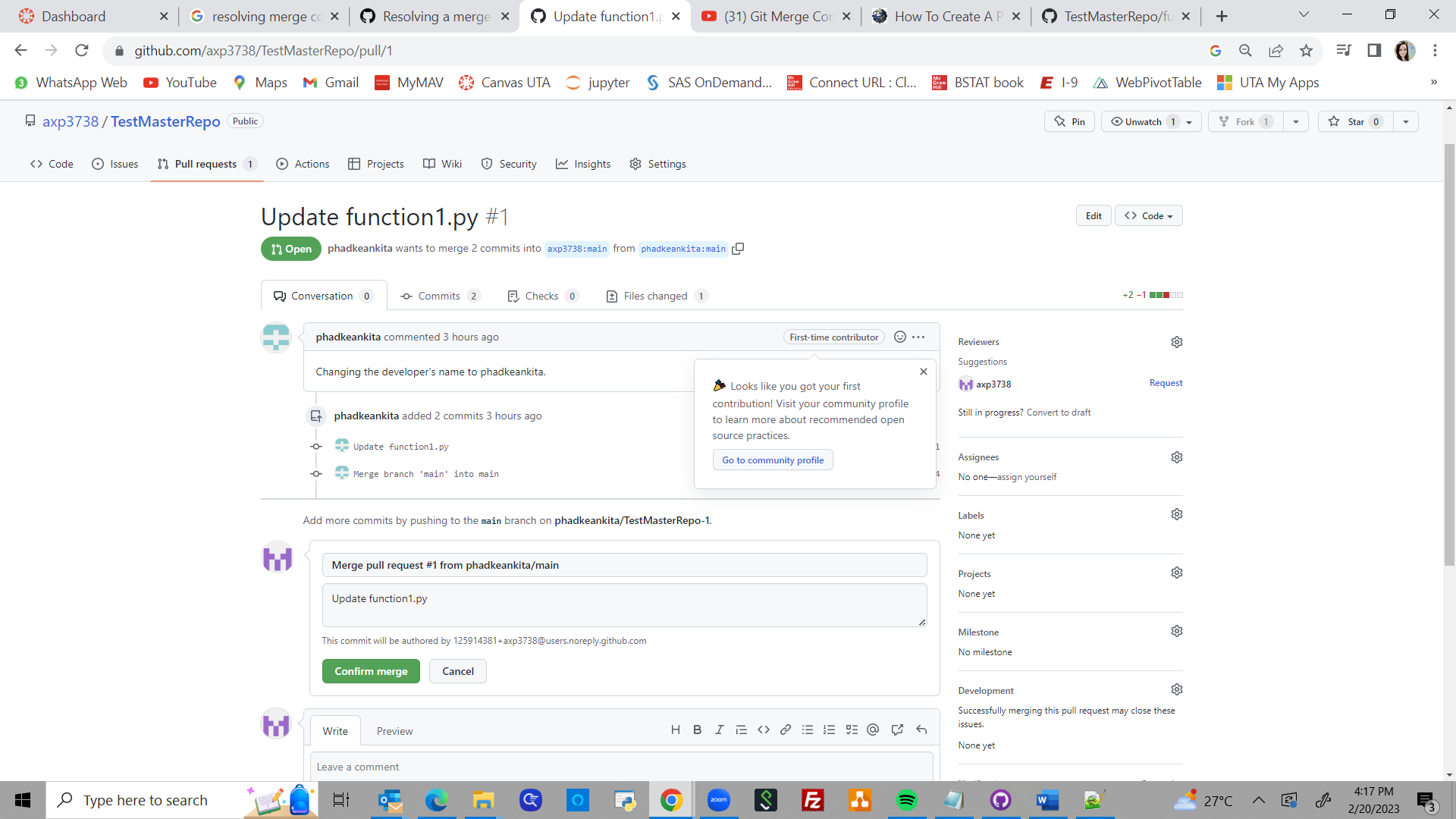


1. In the axp3738 user’s main repository, the commit needs to be approved as this user owns the main branch. After this, click on Merge pull request



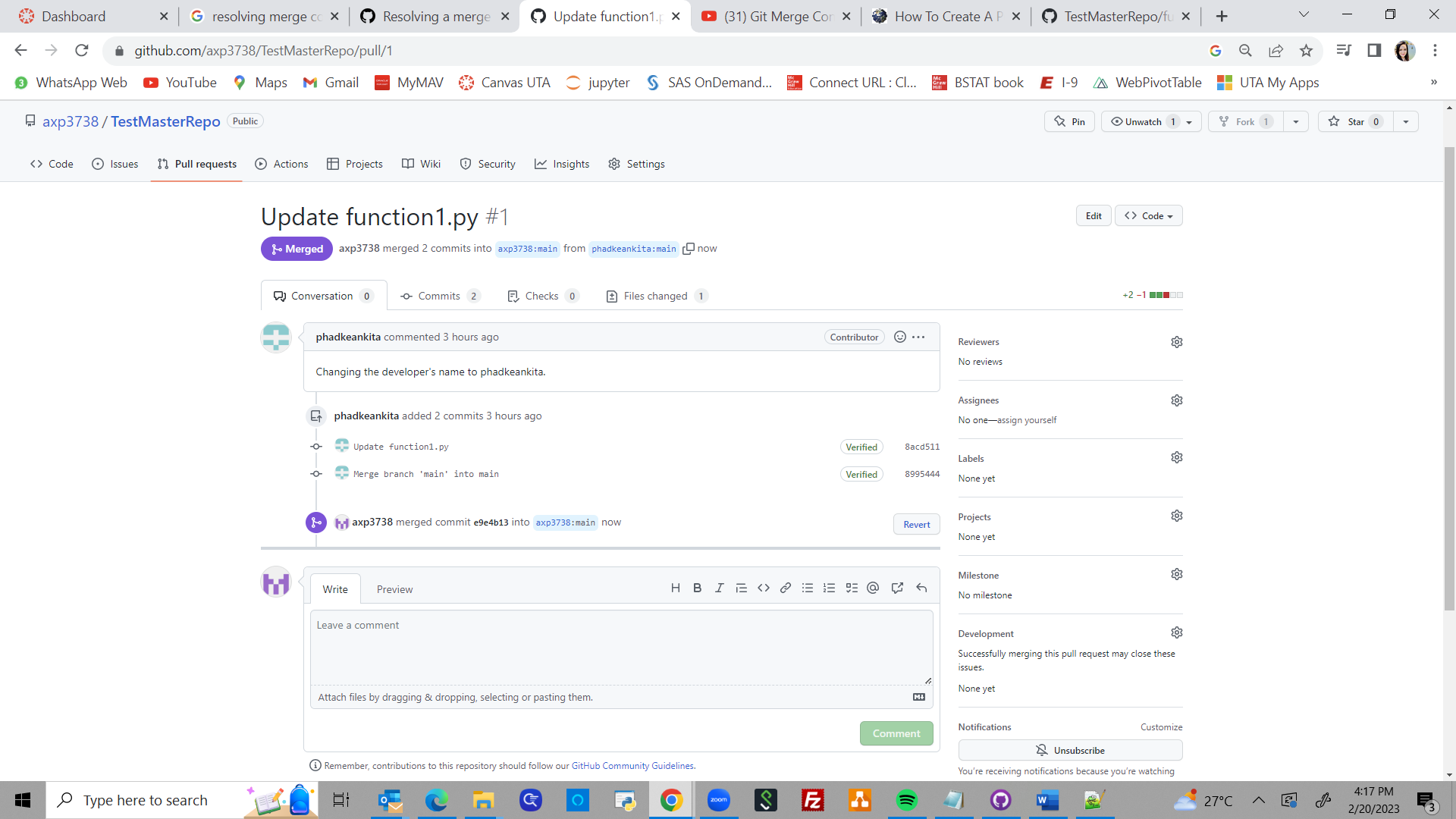


1. Click on Confirm merge to finalize the code change





1. After code merge, this is what the status looks like



1. Below is the final version of this file

