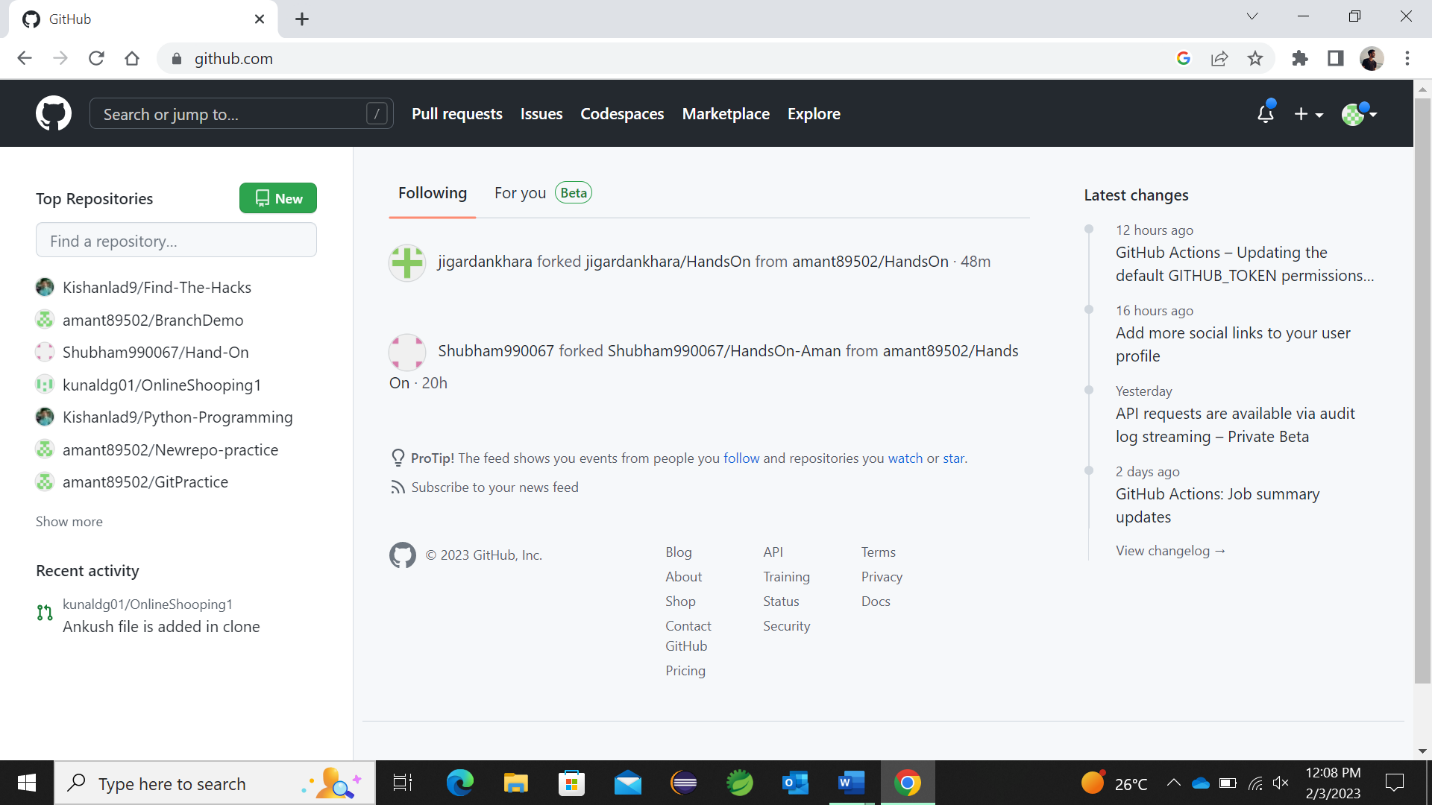
**GIT TEST**

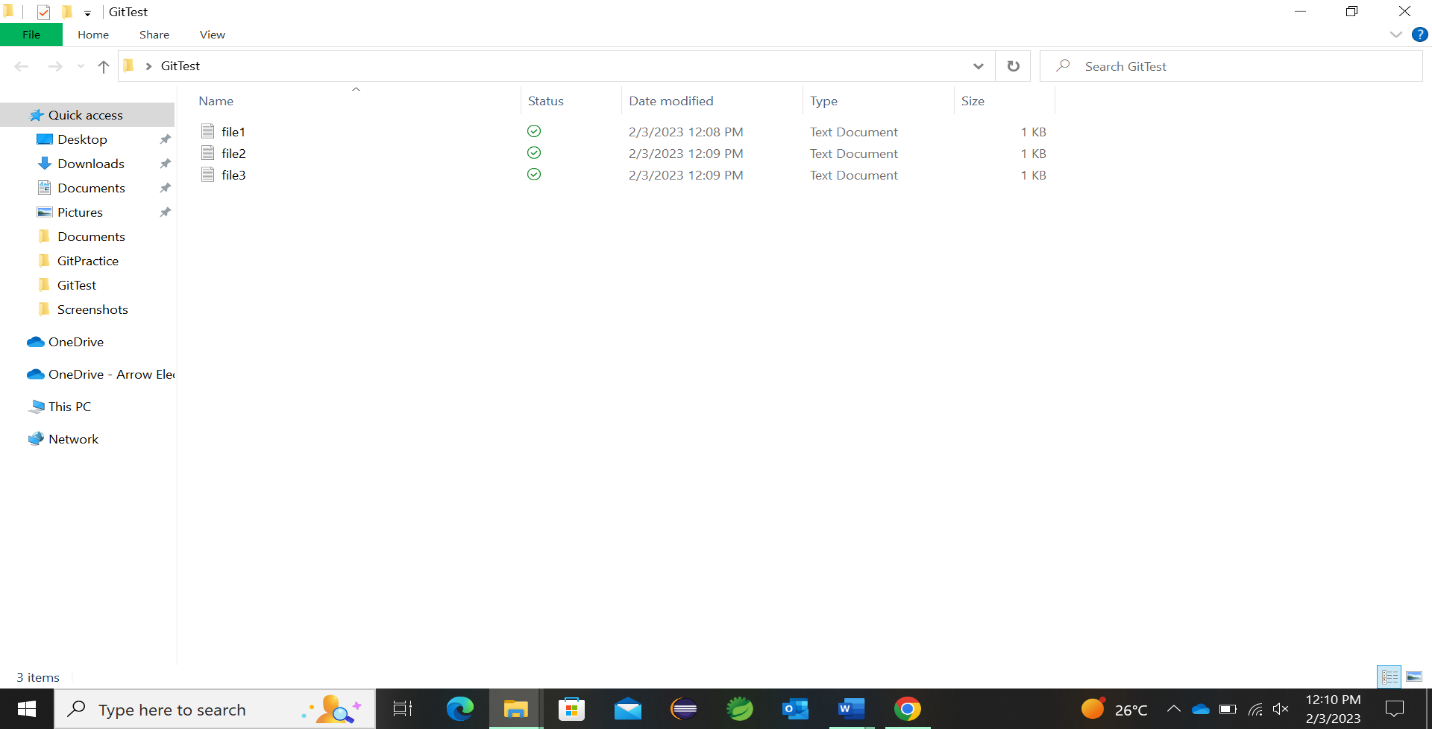
1. Git hub account:

This is how the github interface looks like where we can see and manage all the repo.



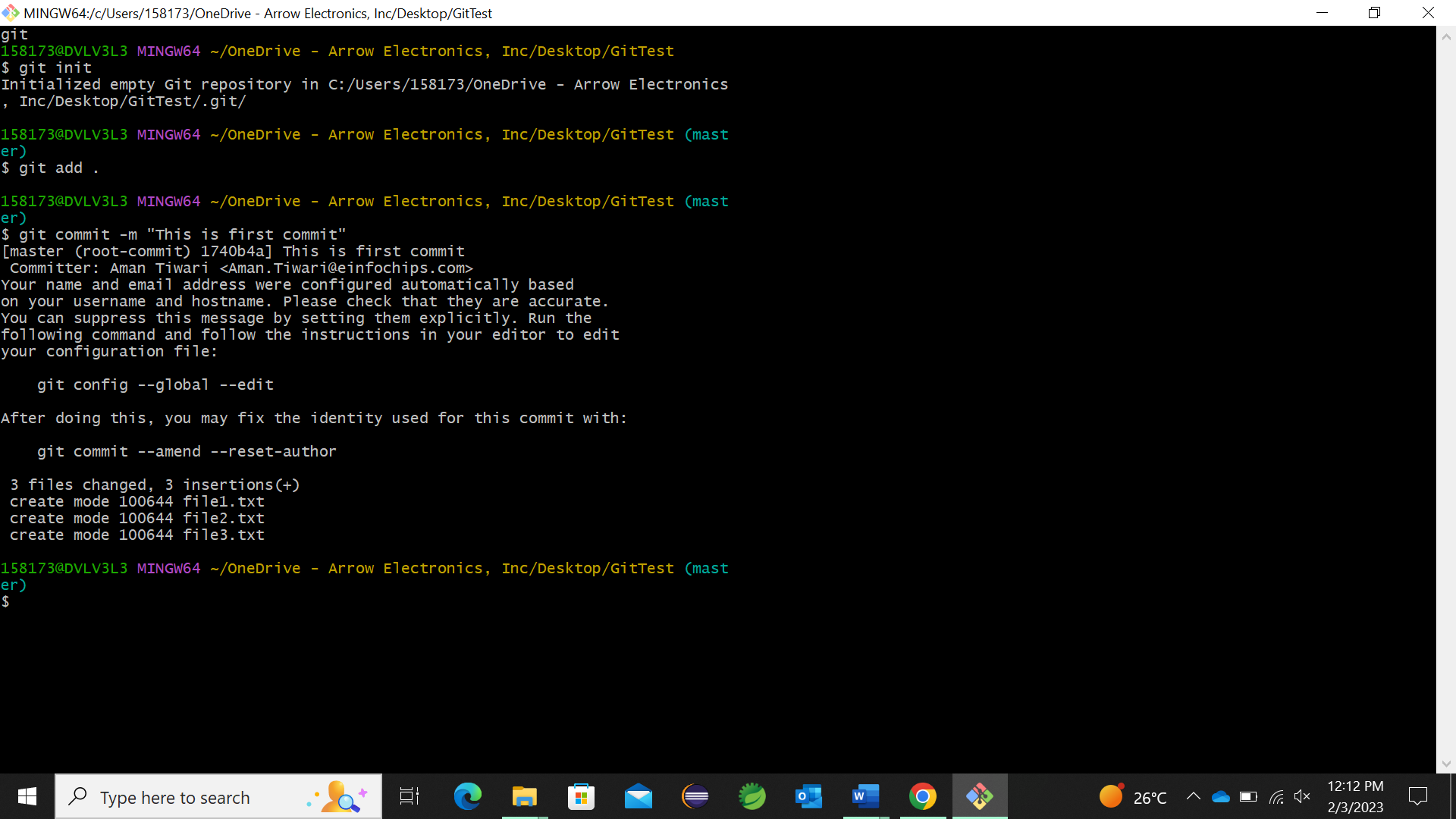
1. Create and add new files to local git repo.

In local PC, we can create and add files as per our project and can then initialize git bash there.



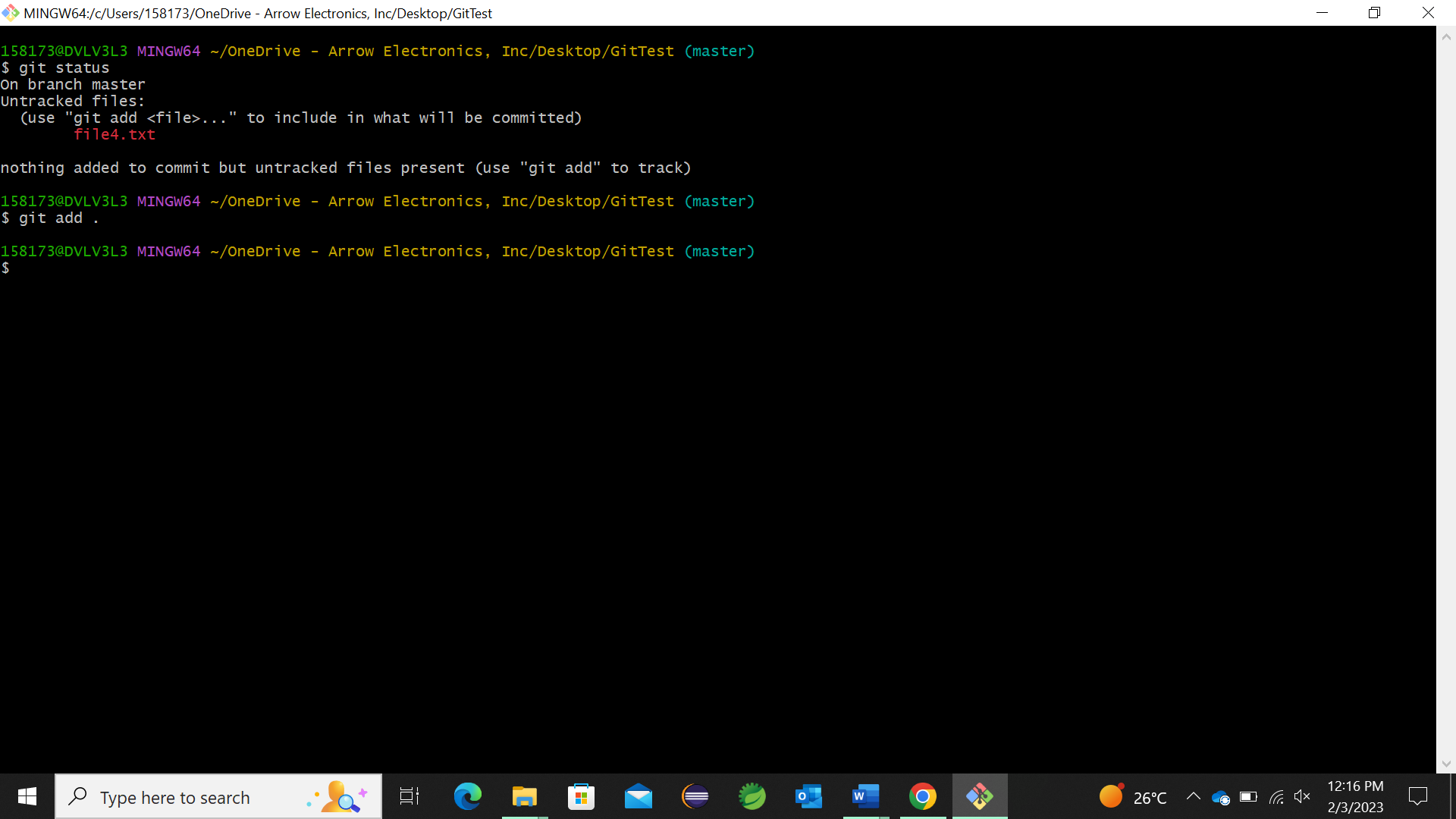
1. Demonstrate the staging environment and various commits

In this section, we can see the files getting added in the staging area and after addition to staging area, it is committed .



1. Add a file in to the staging environment.

In this we need to create a file in our local git repo and can be added to staging area by the command in the given screenshot.



1. Create commit and new branches(3 features and 1 develop)

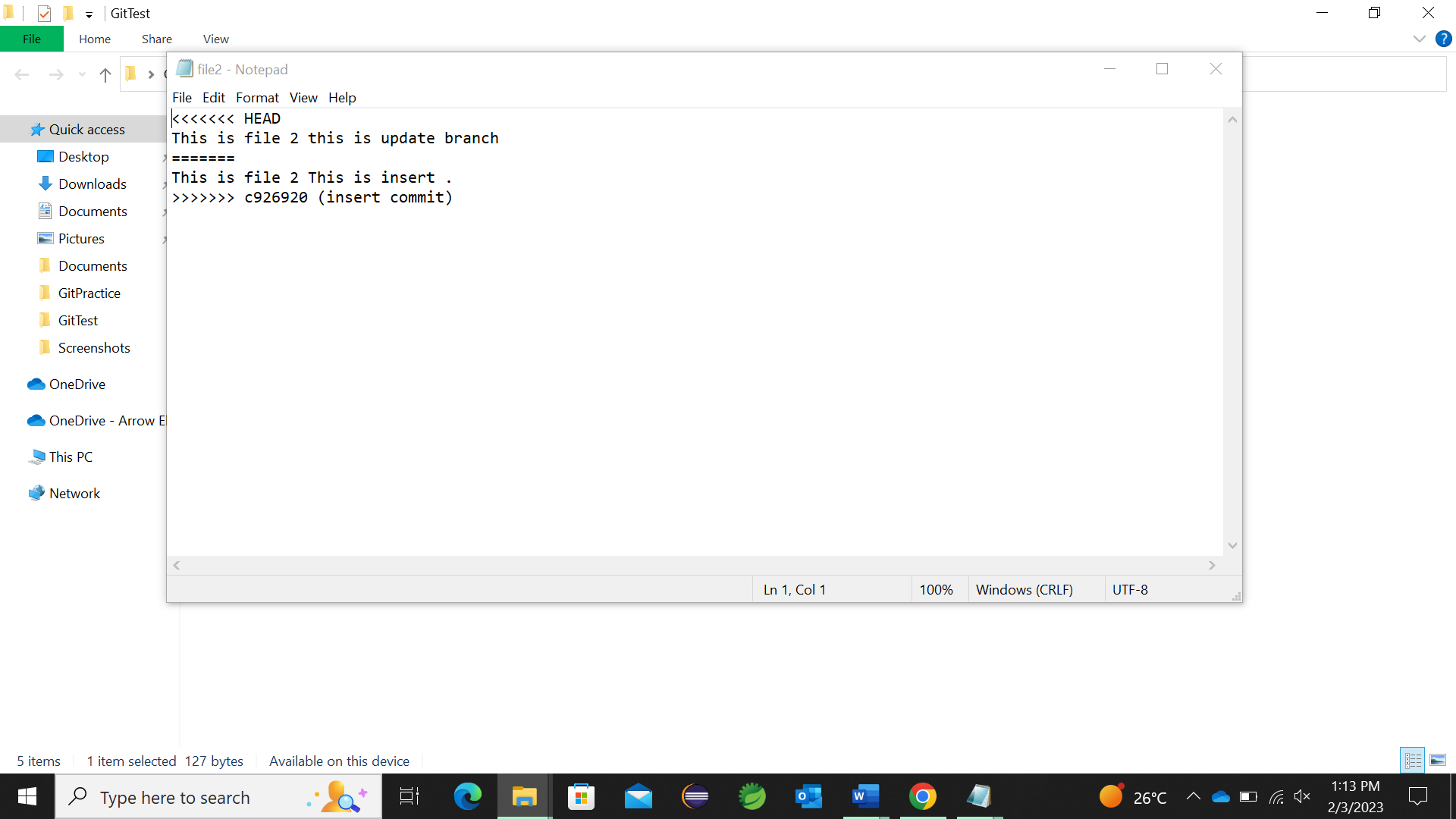
In this section , three feature branch is created namely delete, update, insert and we already have master branch.

A screenshot of a computer

Description automatically generated with medium confidence

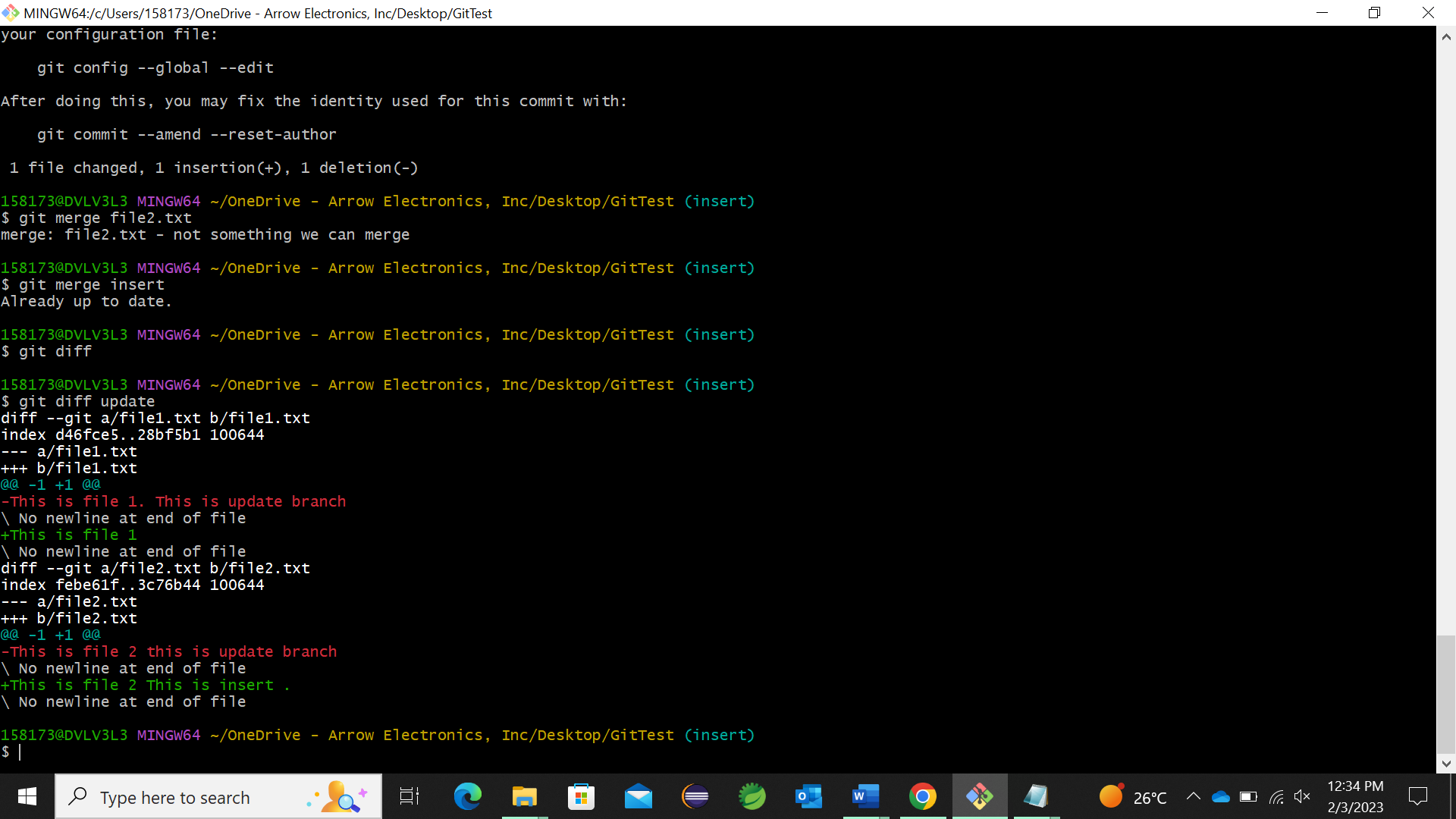
1. Intentionally try to have merge conflict and resolve it manually.

In this we made similar insertion on the same file on same line in different branches. In that case, it will show a conflict message and we can solve it manually.



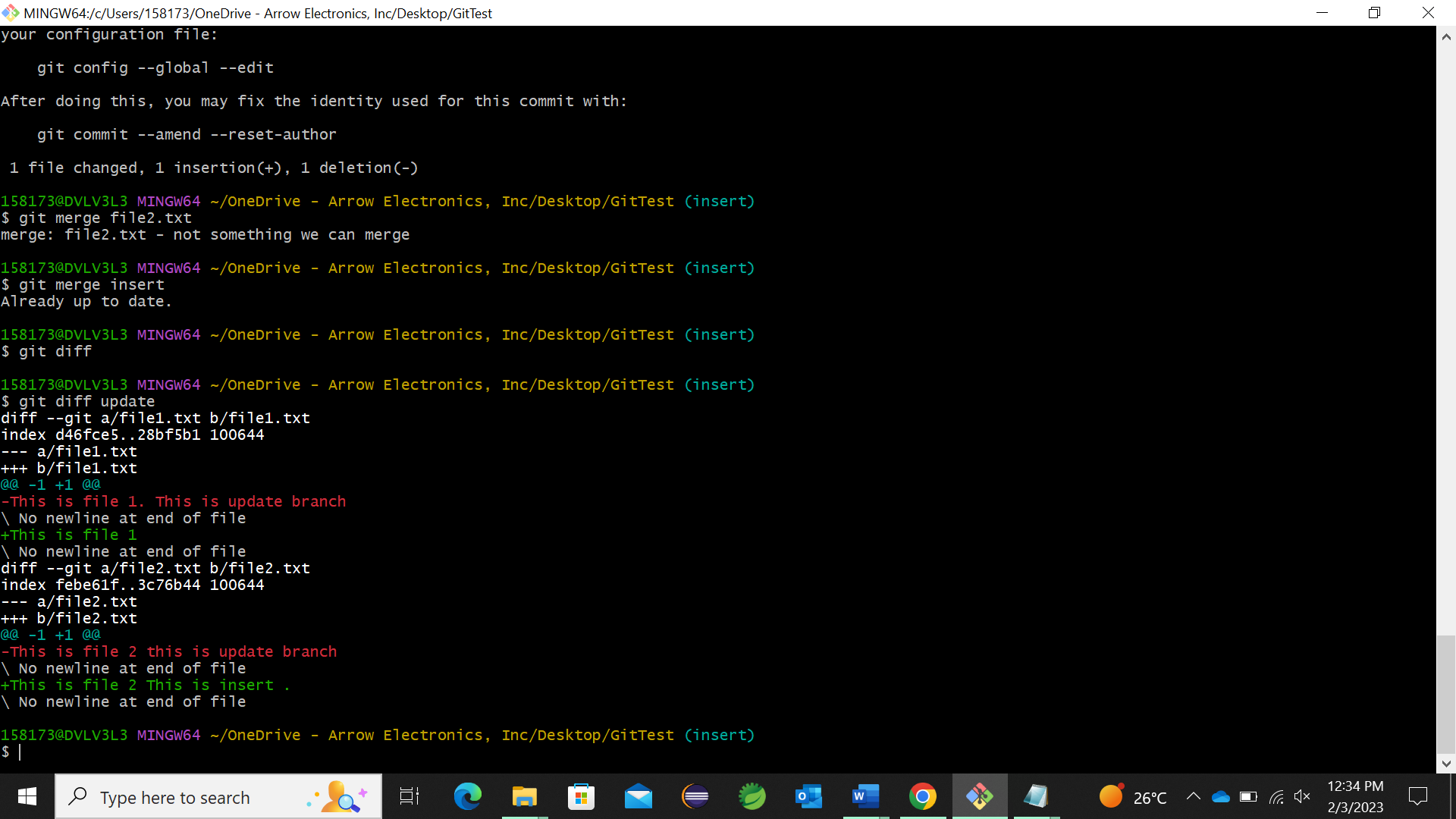
1. Find the difference between branches

In this case, we are on insert branch and by using the command git diff and other branch name, it show the difference between the two branches.



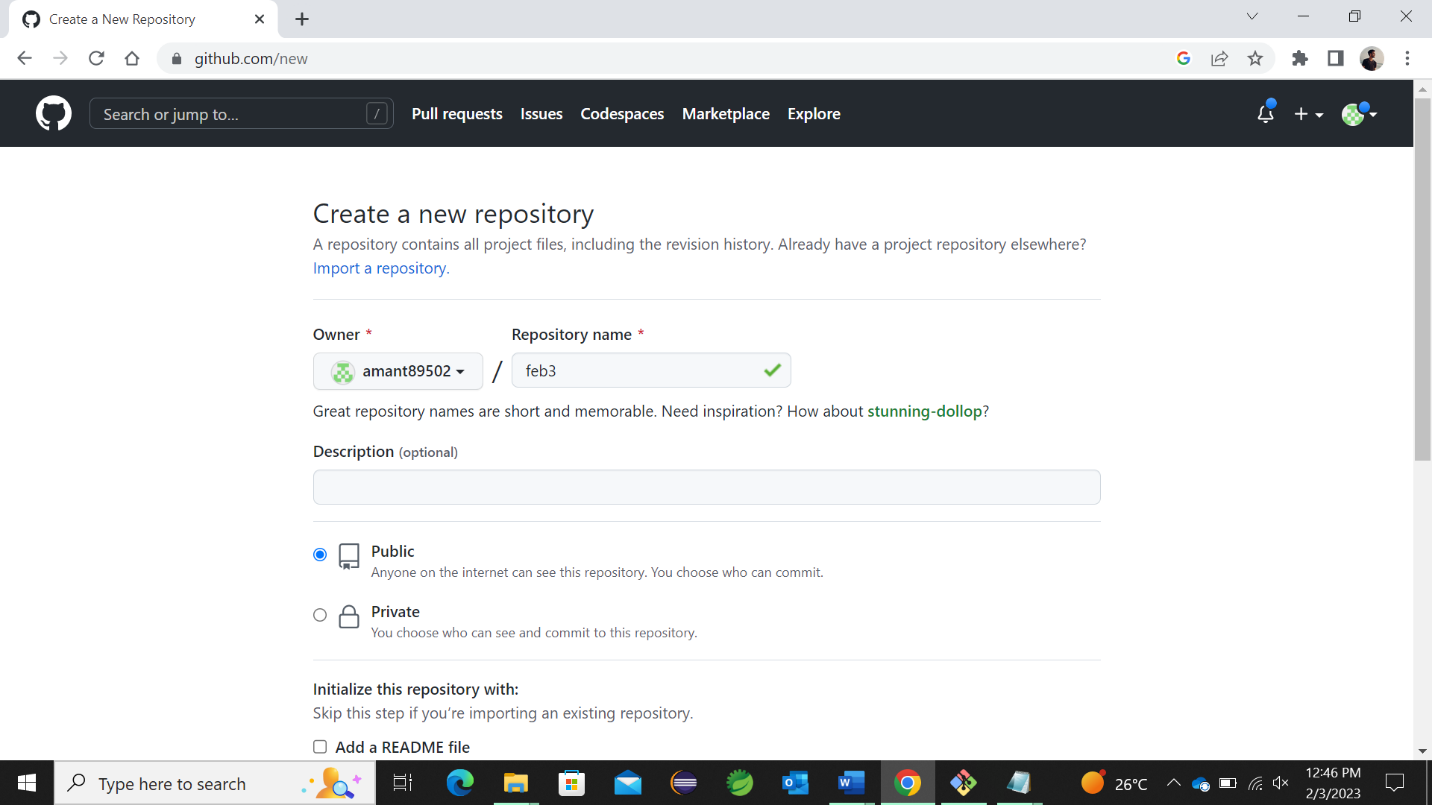
1. Amend the current changes into previous commit.
2. Push the content from feature branch to develop and then develop to master

Here, we are merging the insert branch into update branch.



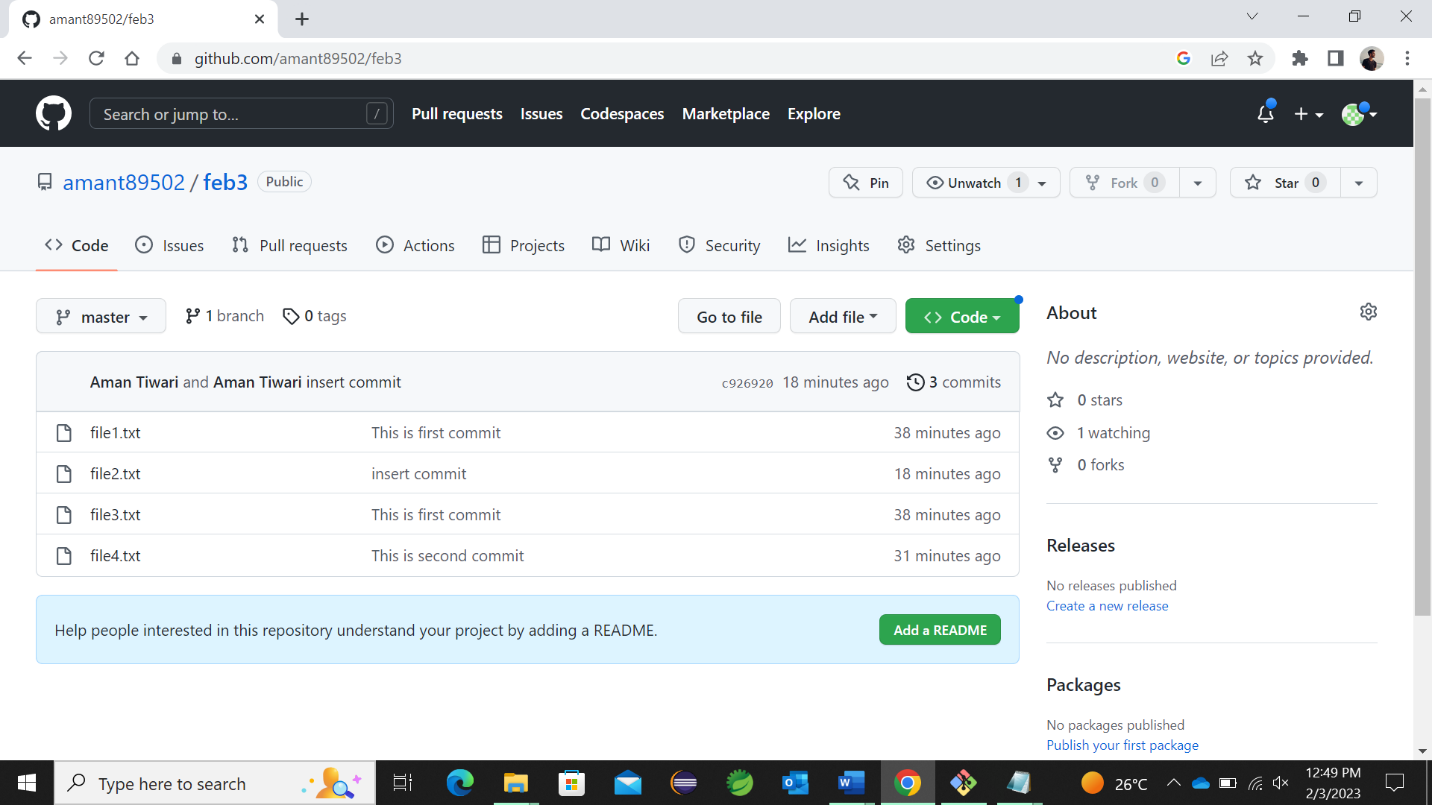
1. Create a new repository on git hub:

We can create new repository by and available repo name and can keep I as public or private.



1. Pushing a branch to GitHub

This two ss show the pushing of master branch from pc to github. We need to add remote location first and then we can push a particular branch from pc to github.

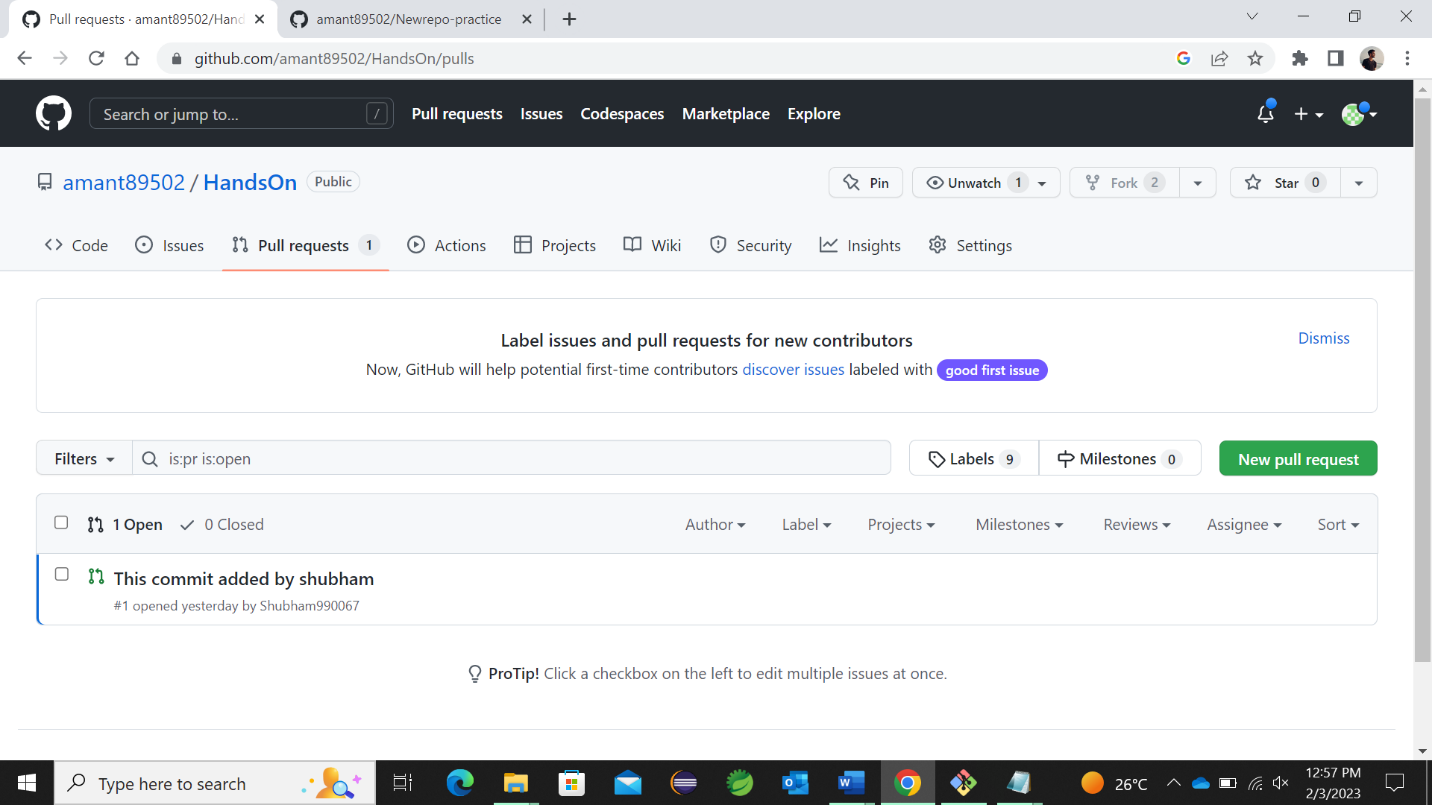


A screenshot of a computer

Description automatically generated with medium confidence

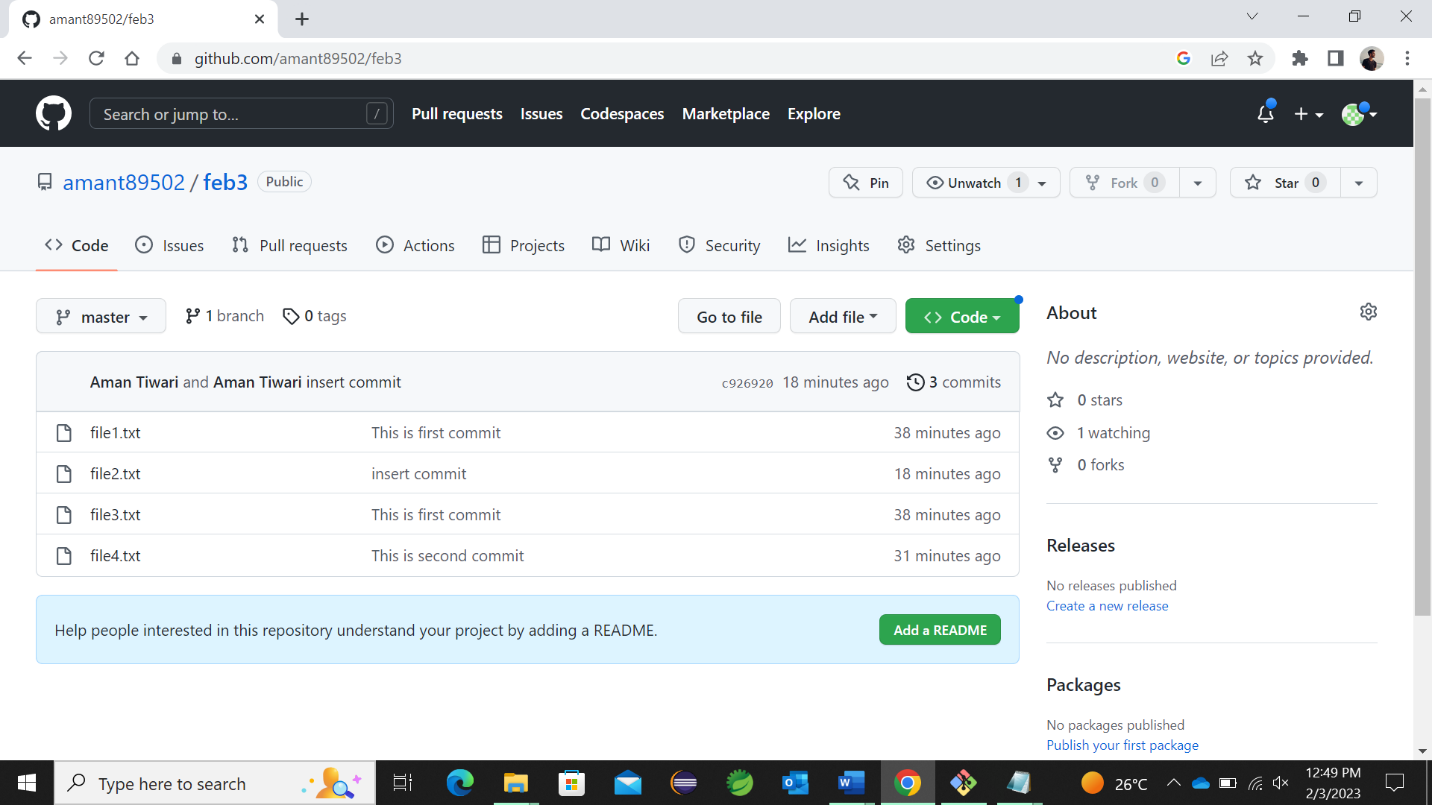
1. Managing and pull request for your projects.

Here we can see the pull requests in the current working repository.



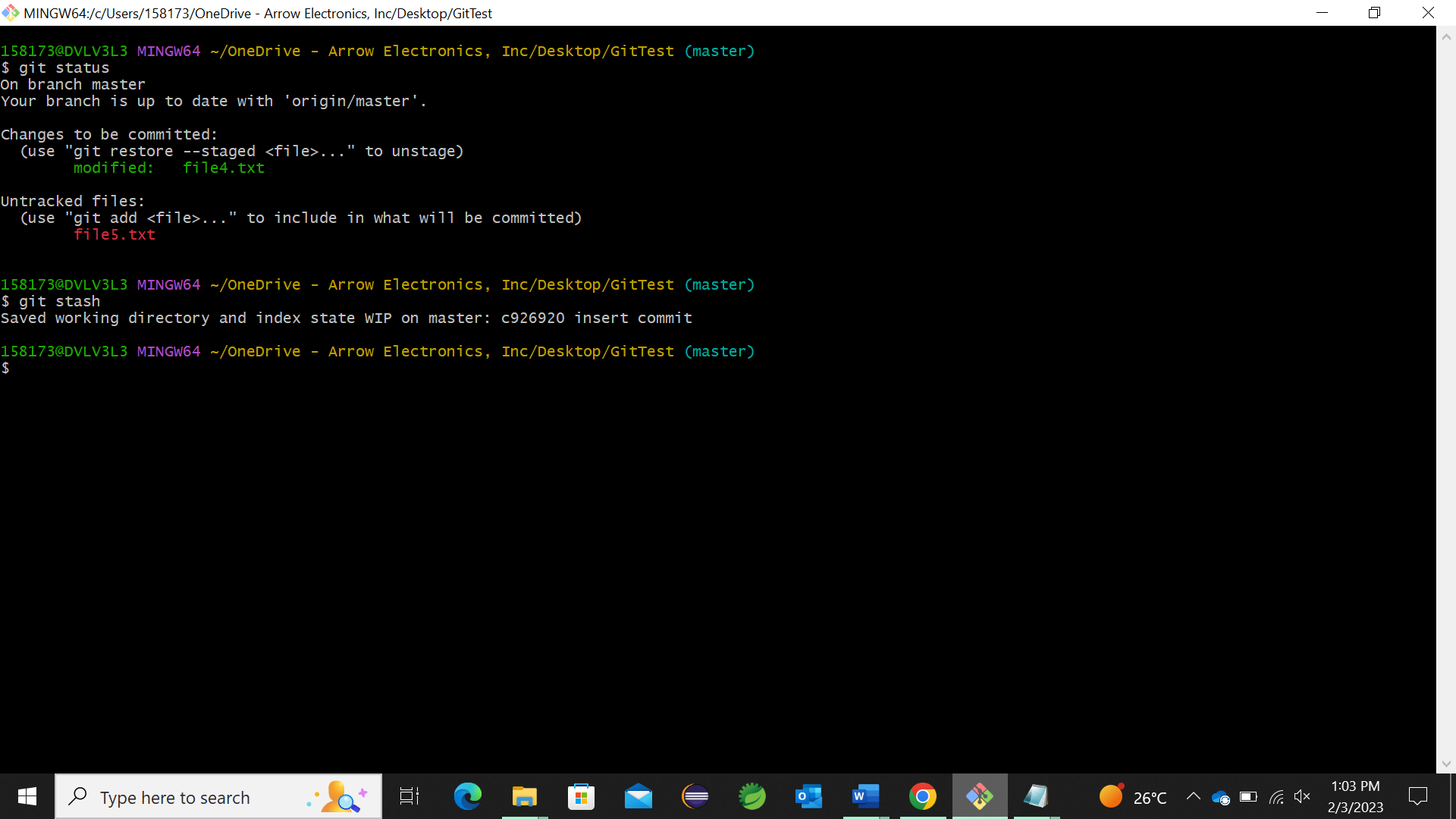
1. Managing changes back to your local PC from github.

Here we can see and manage all the changes back to our local PC to github



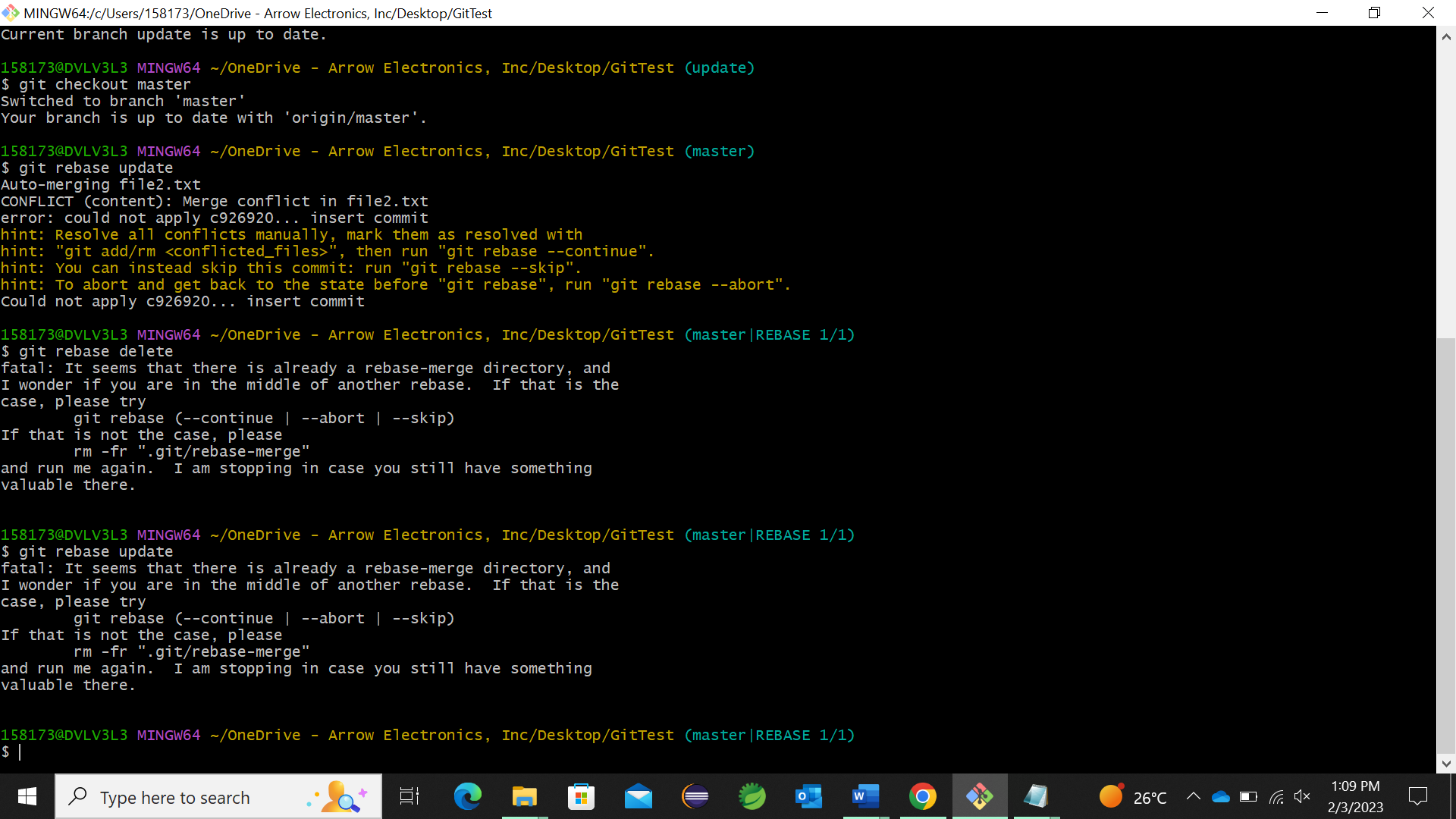
1. Temporary store the working tree.

Here, we can add changes to a particular files and if that changes are not required as of now we can store that with stash in temporary area.



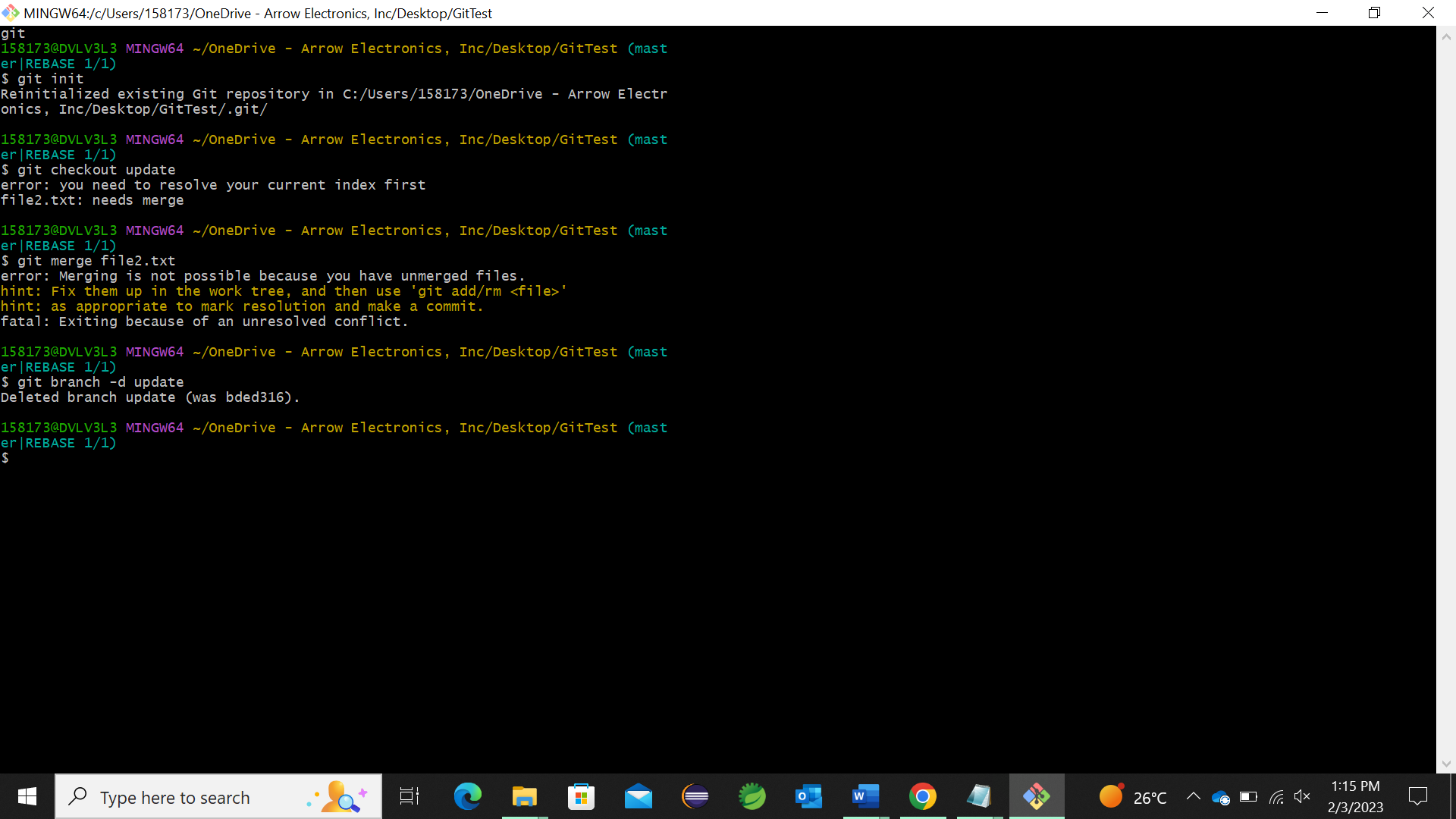
1. Rebase the master branch into sub branch

Rebasing it adding of sub branch to master branch. It will not happen if we have conflicts like in section 6.



1. Remove one branch from repo.

This deletes the branch.



1. Git clone

This is