OpenFoam Tutorials:

Task 1: Pipe Flow with varying Re (rho\*u\*d/mu) for constant diameter pipe (d) and constant physical properties (rho, mu)

Prerequisites:

* FreeCad
* Salome
* OpenFoam
* Paraview

Introduction:

FreeCad?

* A 3D parametric modeler open source software
* It can be downloaded from official website: <https://www.freecad.org/>

What is GitHub?

* A web-based graphical user interface (GUI) Git repository that helps developer to collaborate and work together from ANYWHERE
* GitHub is a place where a project MANAGER and DEVELOPERS coordinate, track, and update their work so project stay TRANSPARENT and ON SCHEDULE

Note: Git and GitHub both are different. Git does not need GitHub but GitHub requires Git to work.

Example: You could be working on a website’s landing page and find that you do not like the navigation bar. But at the same time, you might not want to start altering its components because it might get worse.

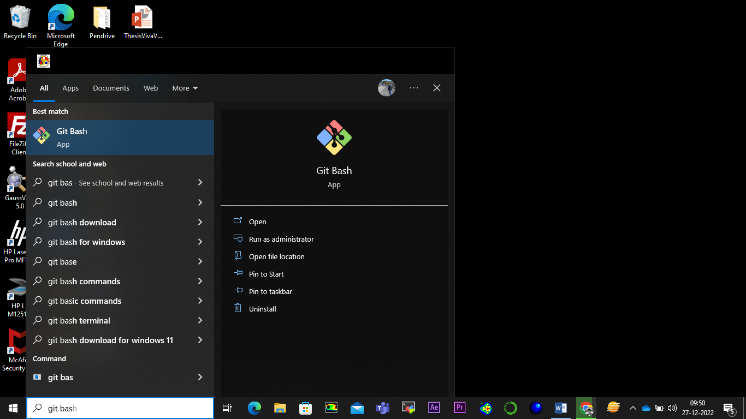
With Git, you can create an identical copy of that file and play around with the navigation bar. Then, when you are satisfied with your changes, you can merge the copy to the original file.

Git Installation:

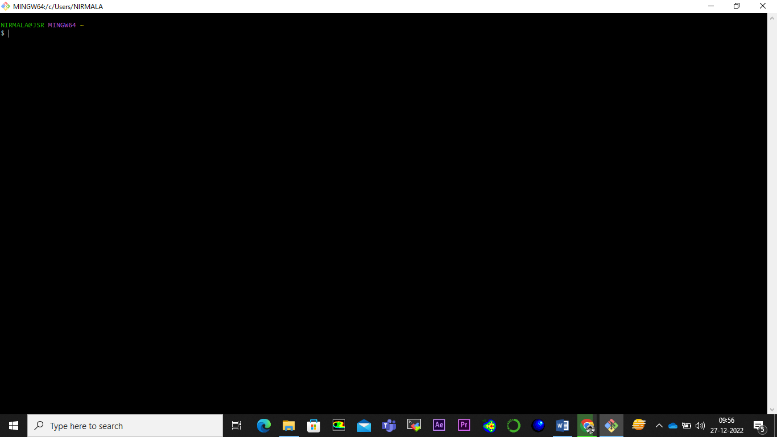
GitHub account:

Now working with Git:

* Start 🡪 Type “Git Bash” to search 🡪

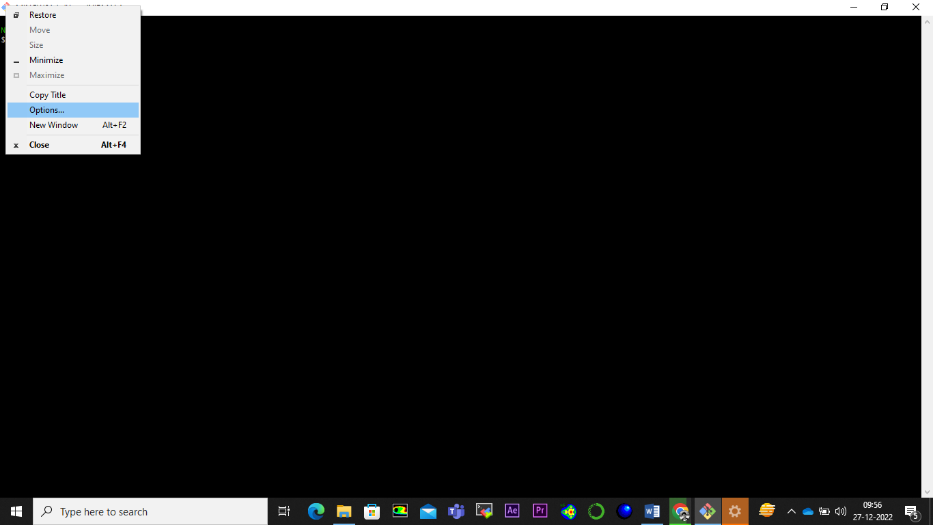


* After the Git window pop up, it will look like this

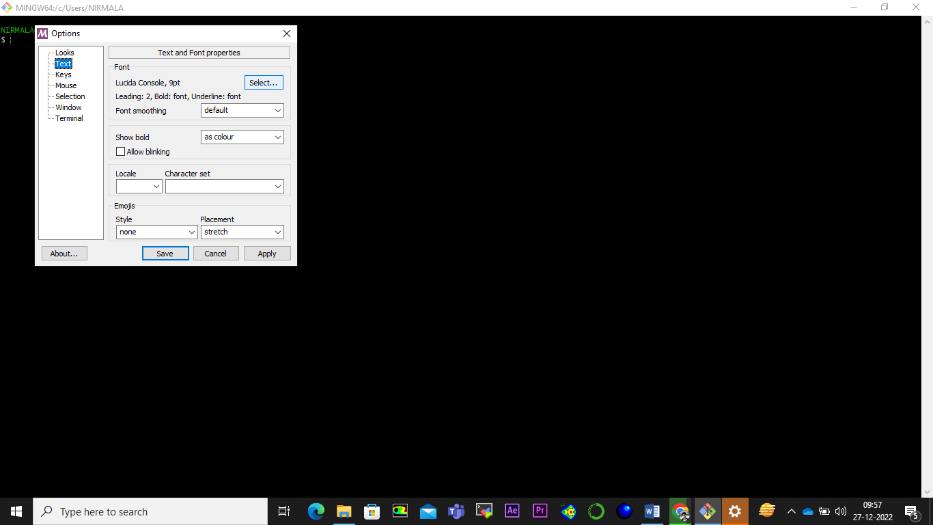


* To customize the Theme, Fonts and appearance follow the procedure given below:

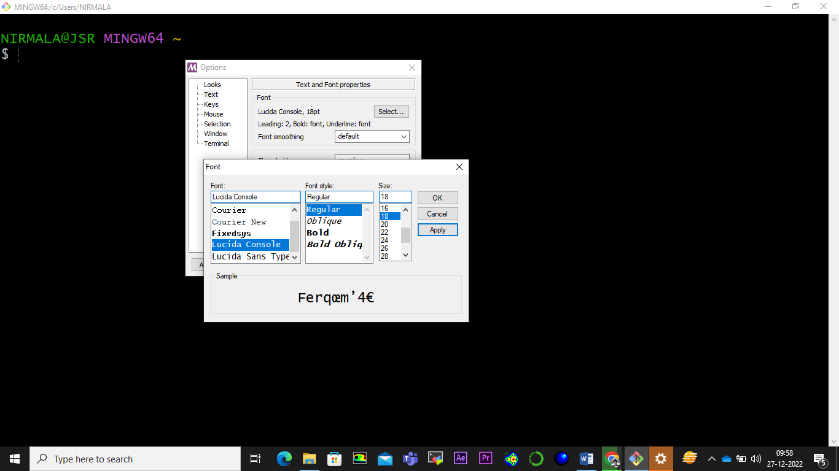
1. Right Click on the Git icon



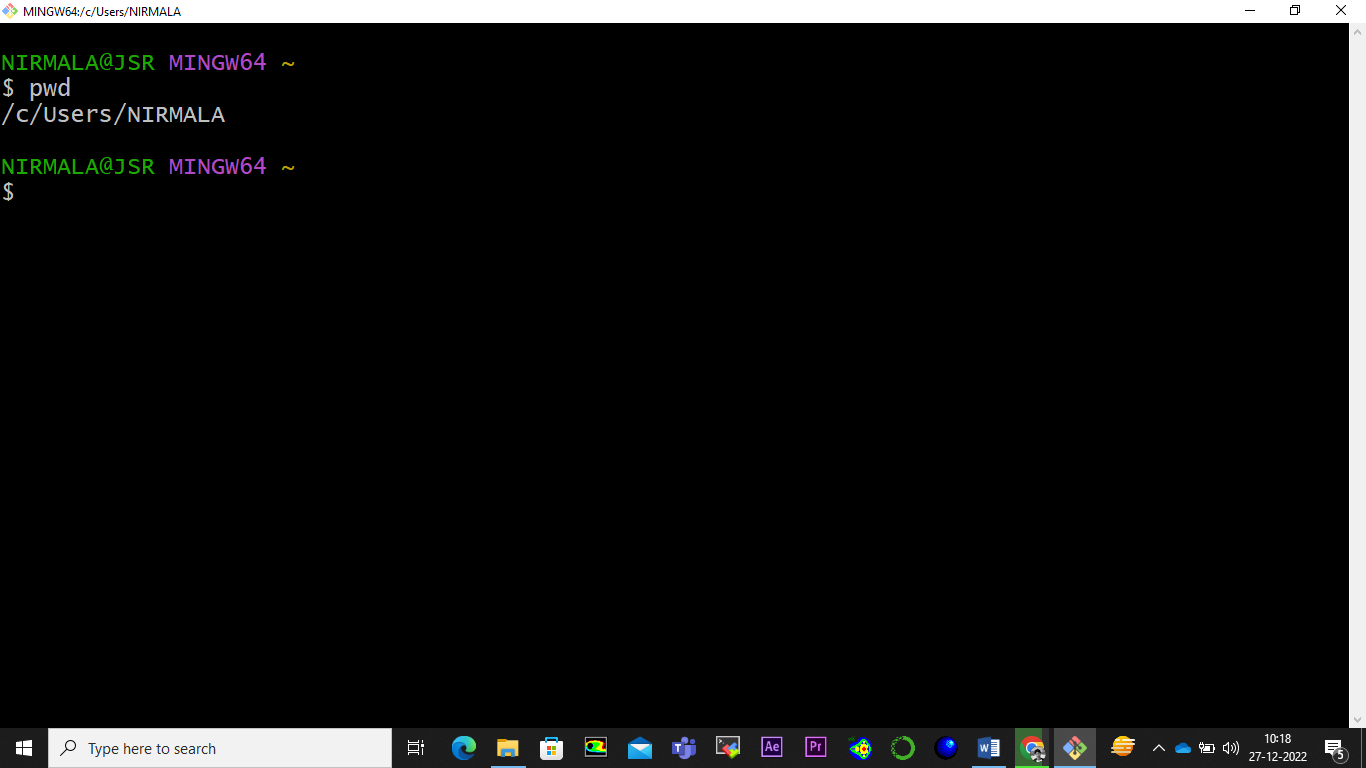
1. Select “Option” from dropdown menu



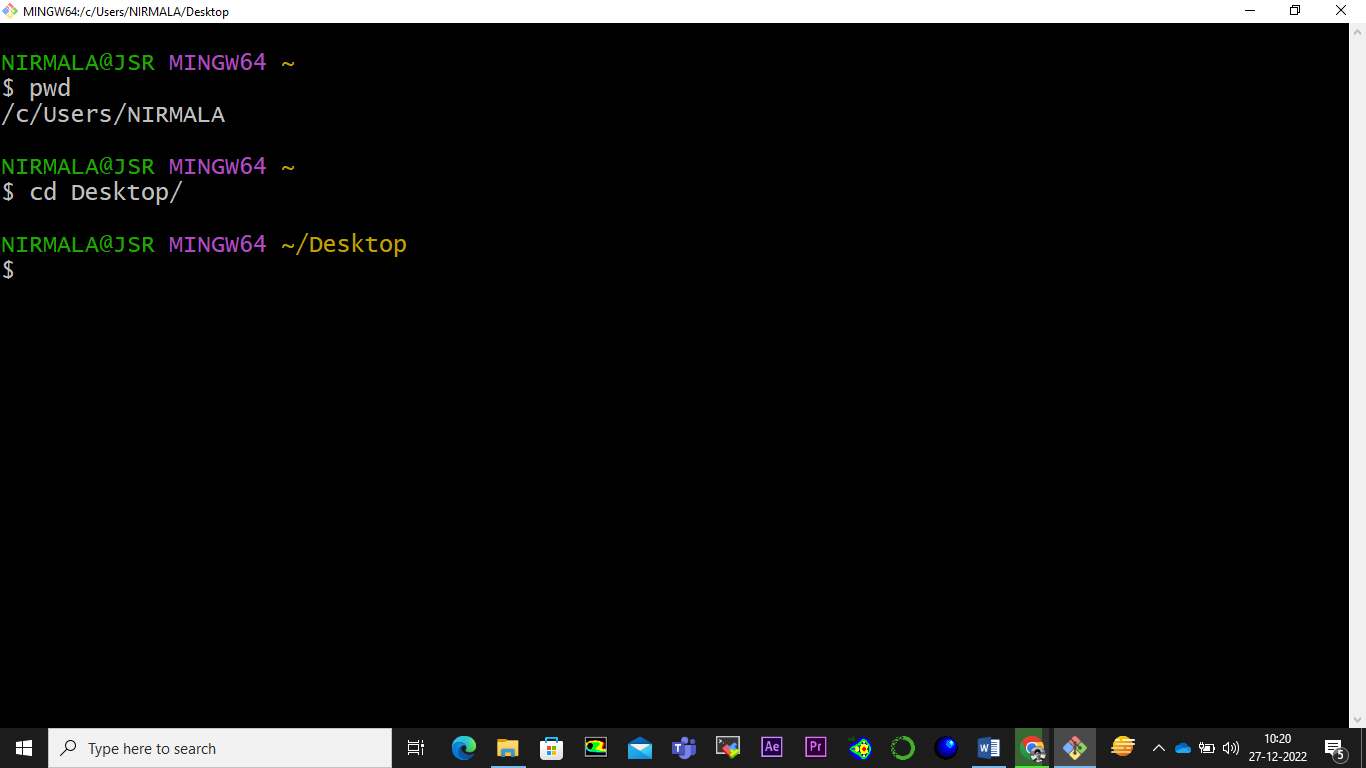
1. Select the “Text” option from Left panel, and click on the “Select” button under Font panel. Select the appropriate options from the selection panel and click on the “Apply” button.



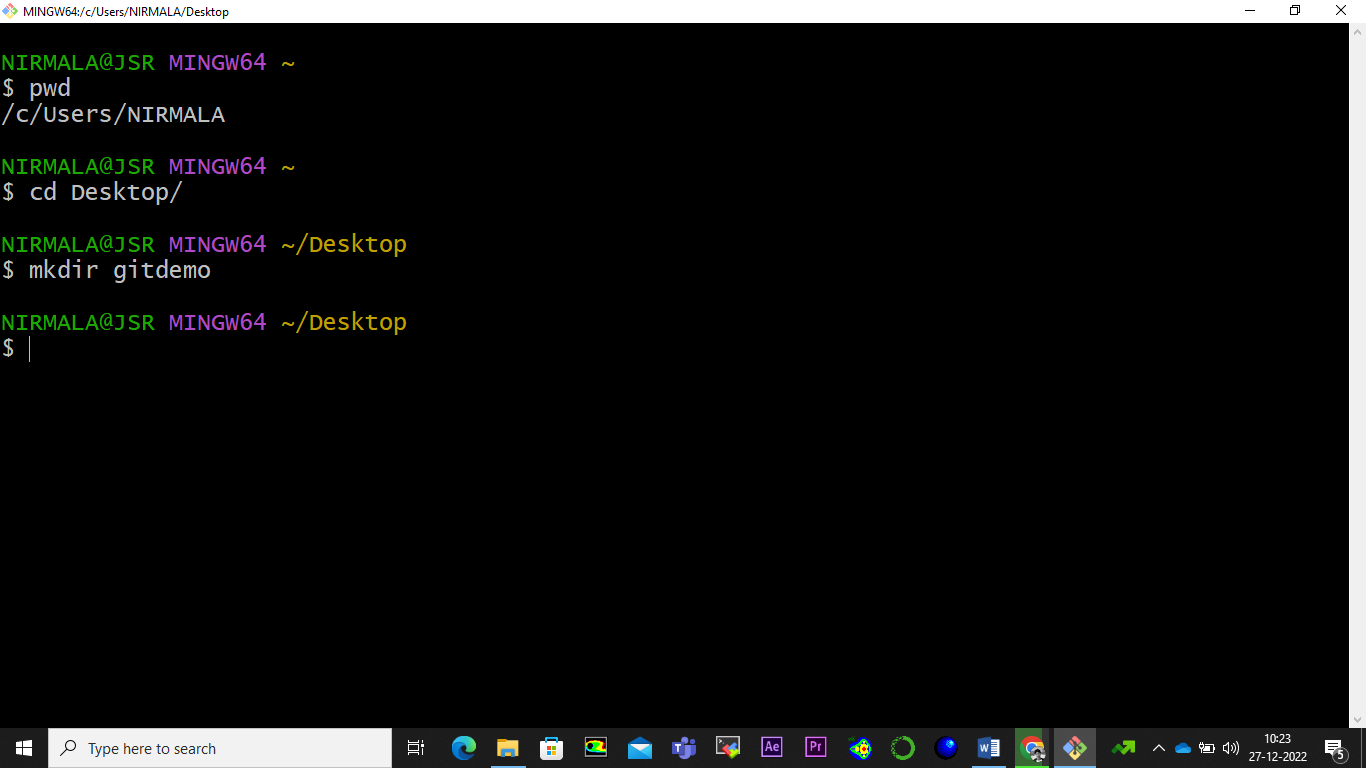
* Now let’s start working on Git. To get the path of present directory, use command “pwd”



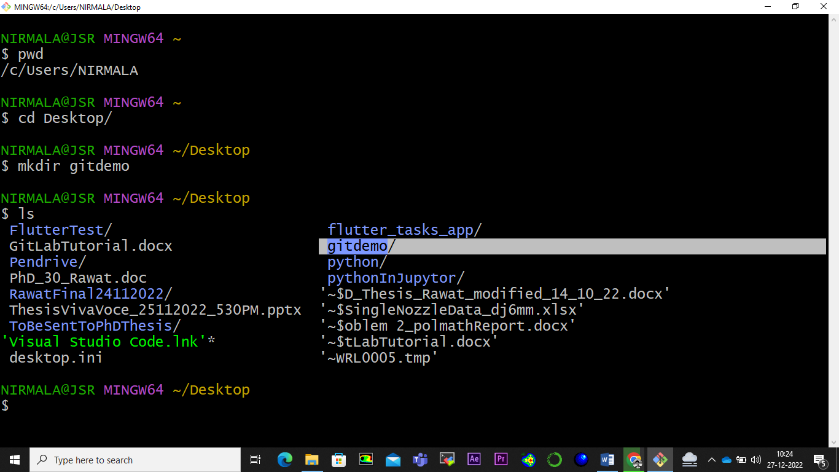
* Here we are in root directory. To navigate to other directory, use command “cd”



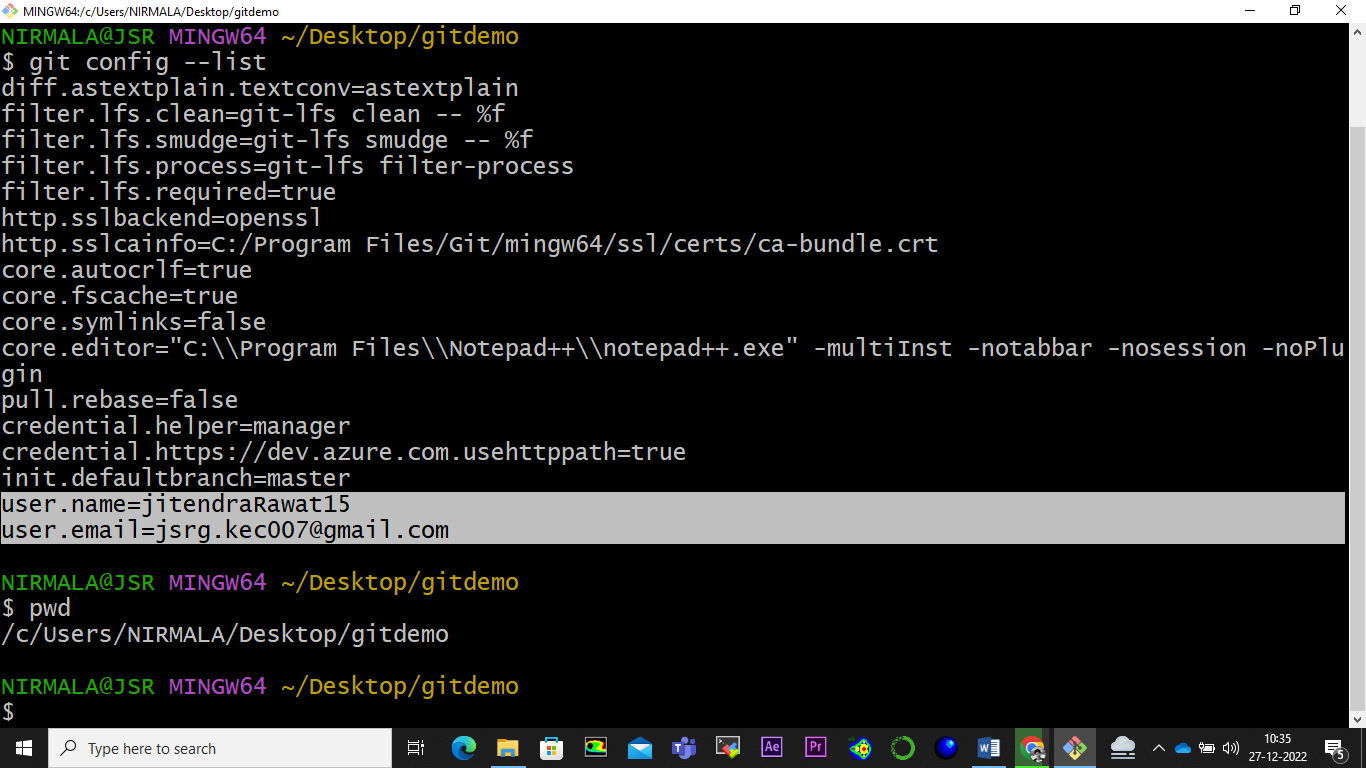
* Let create a new directory “gitdemo” on the desktop. Use command “mkdir”.



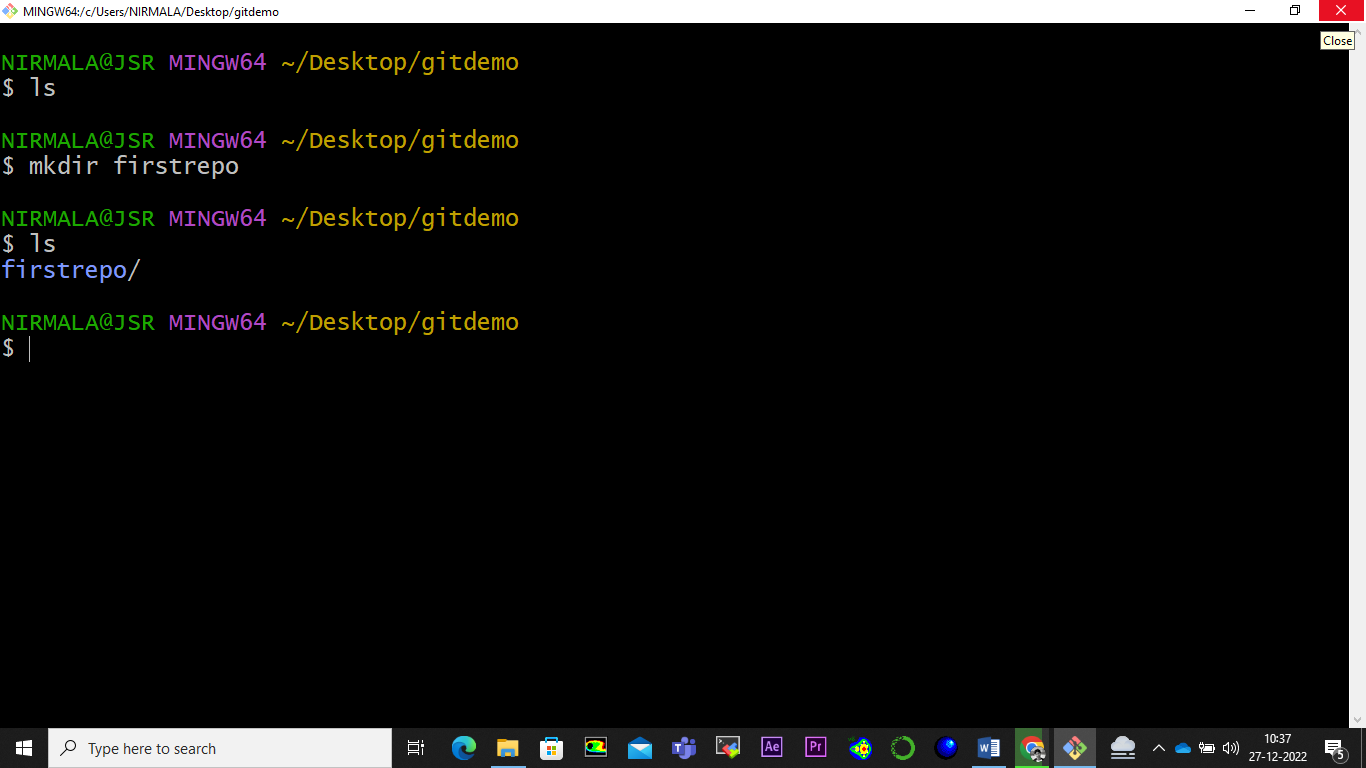
* To list out all the contents of the directory, use command “ls”.



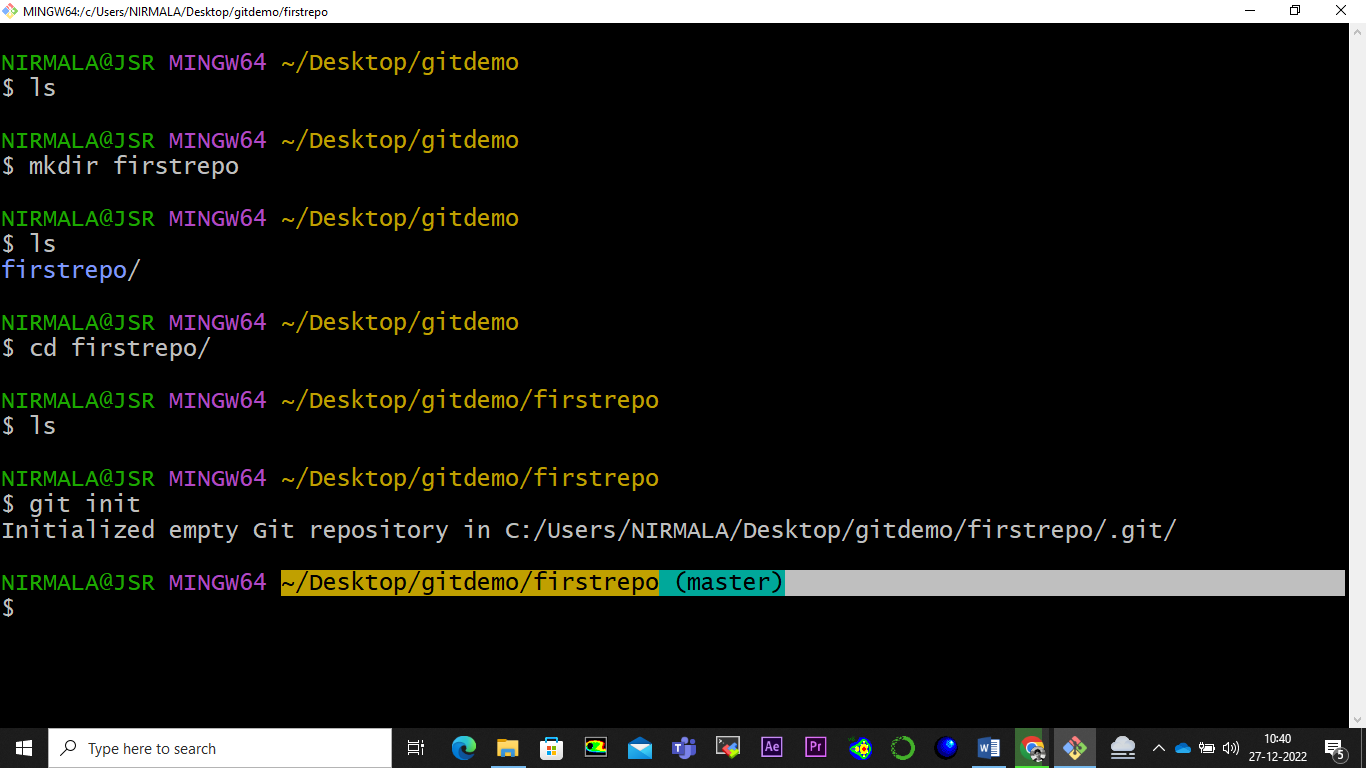
* Navigate to “gitdemo” directory.



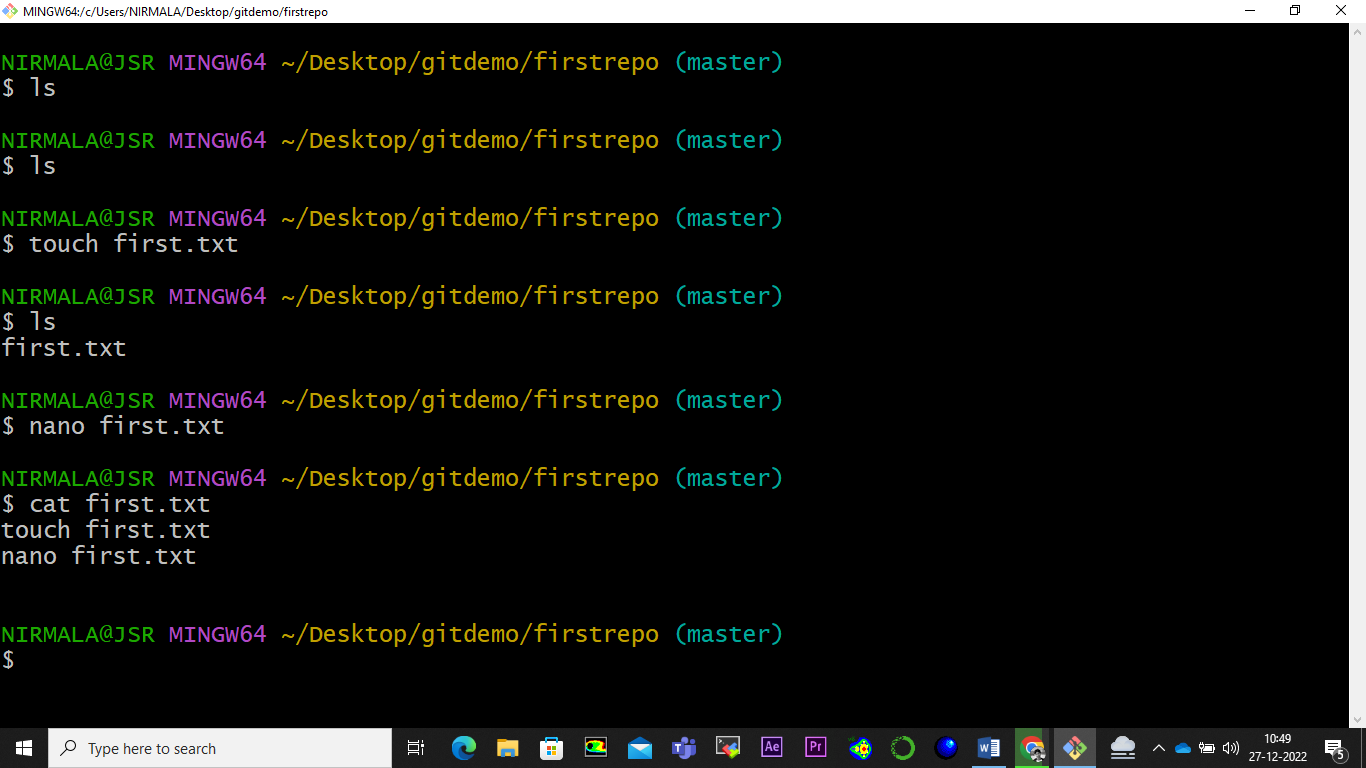
* Create a local git repository : Create a new directory “firstrepo” and navigate to it



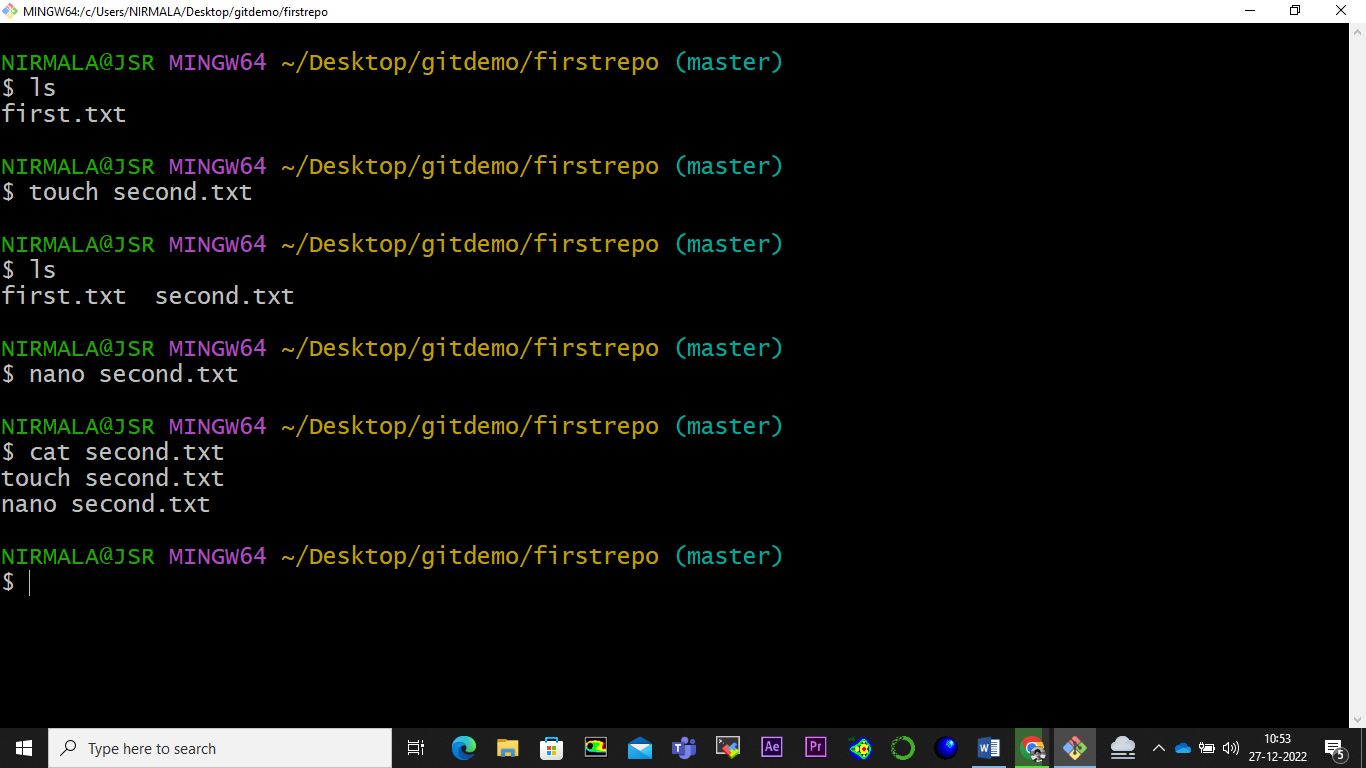
* Initialize the current directory as git repository. Use command “git init”



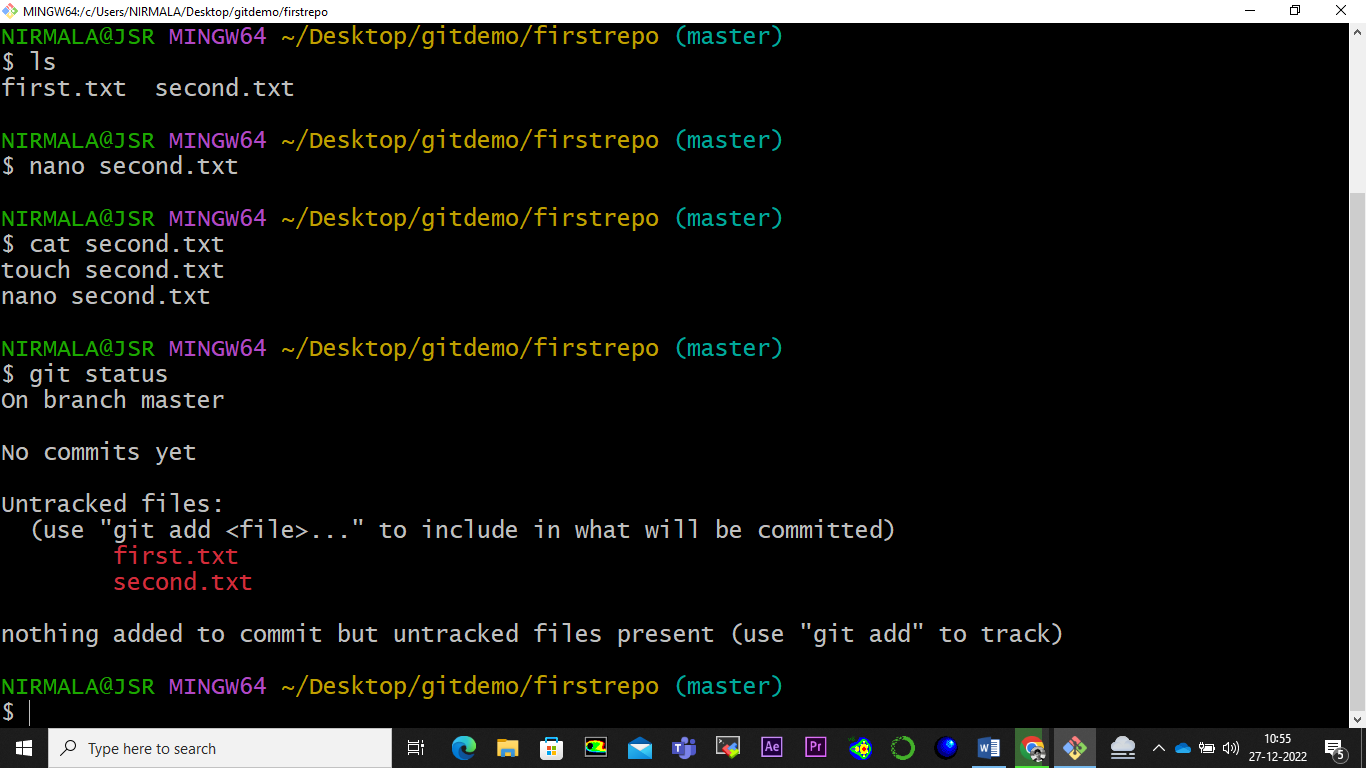
* Add a new file to the repo: Check the contents of the git repository. Create a new text file with “touch” command.



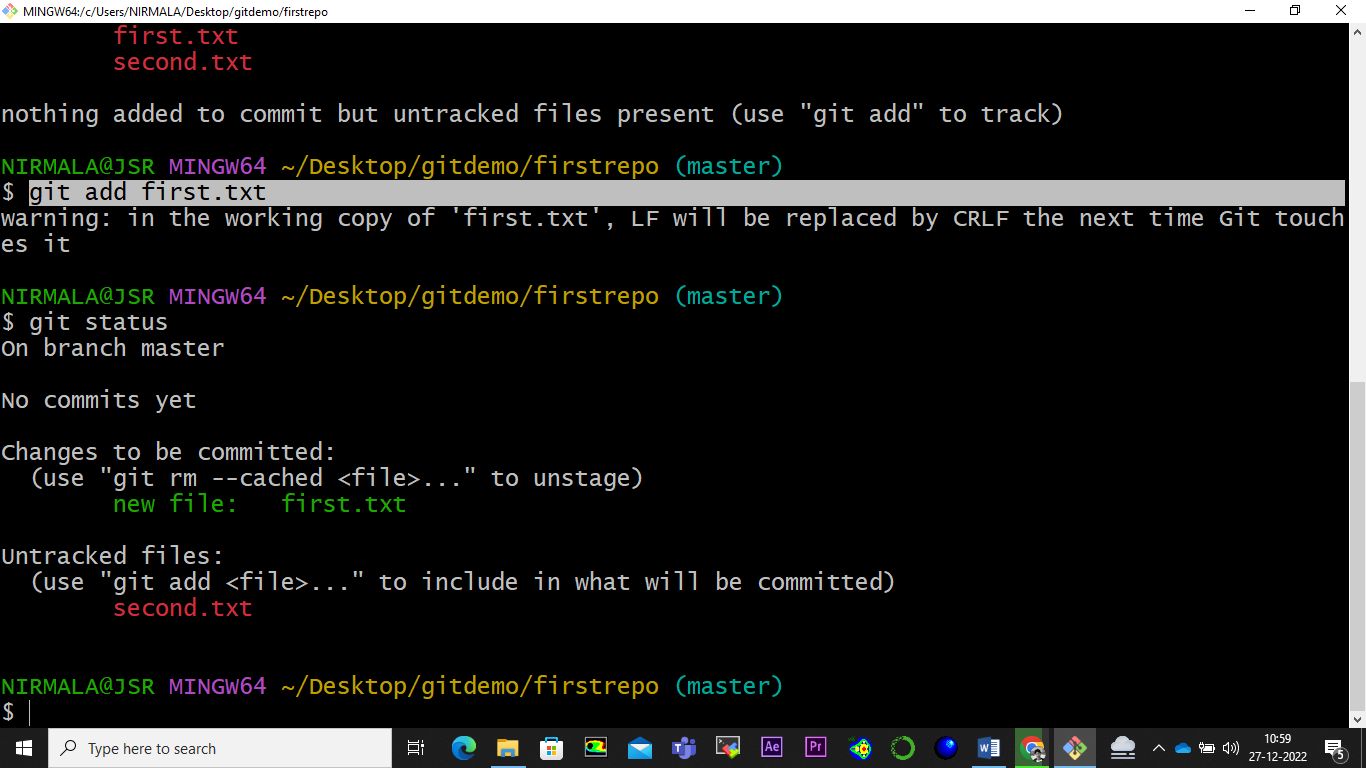
* Create second text file with “touch” command



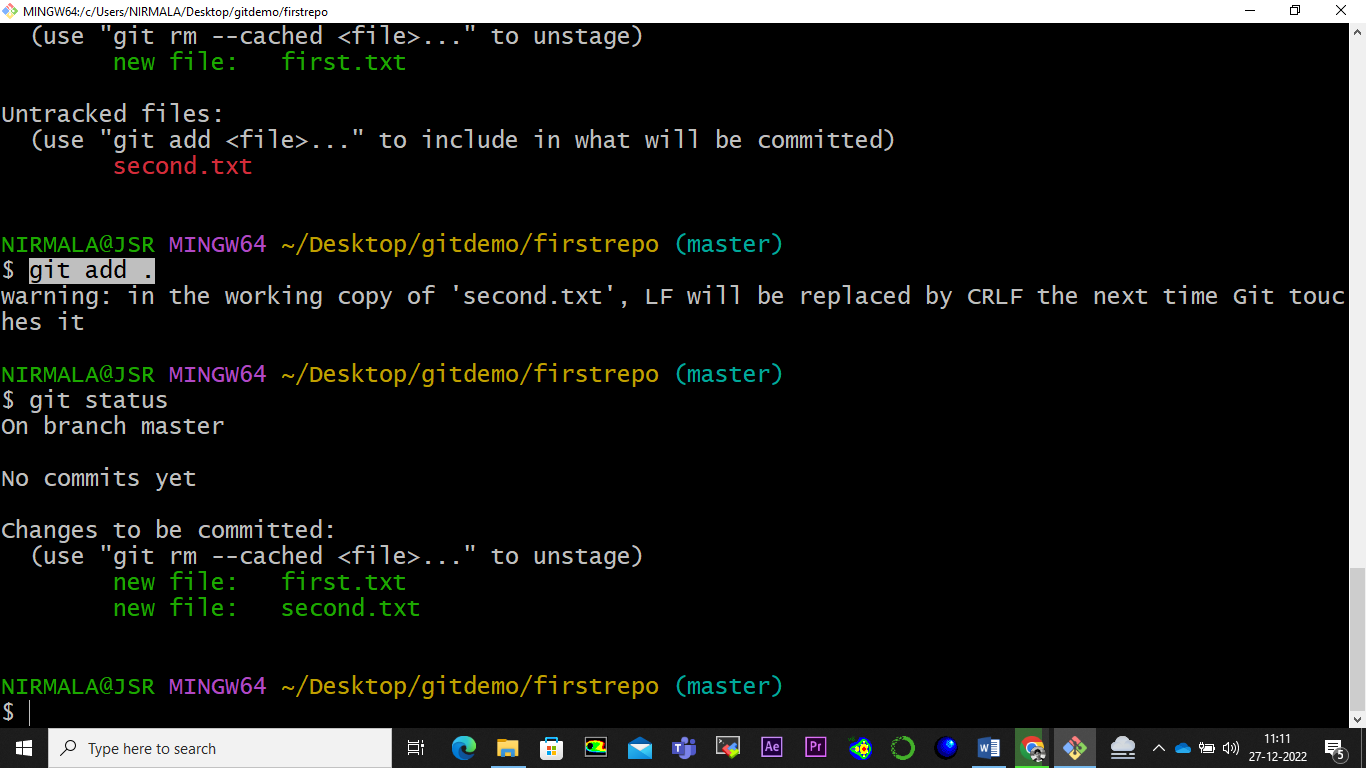
* Let’s check the status of the files created. Use command “git status”.



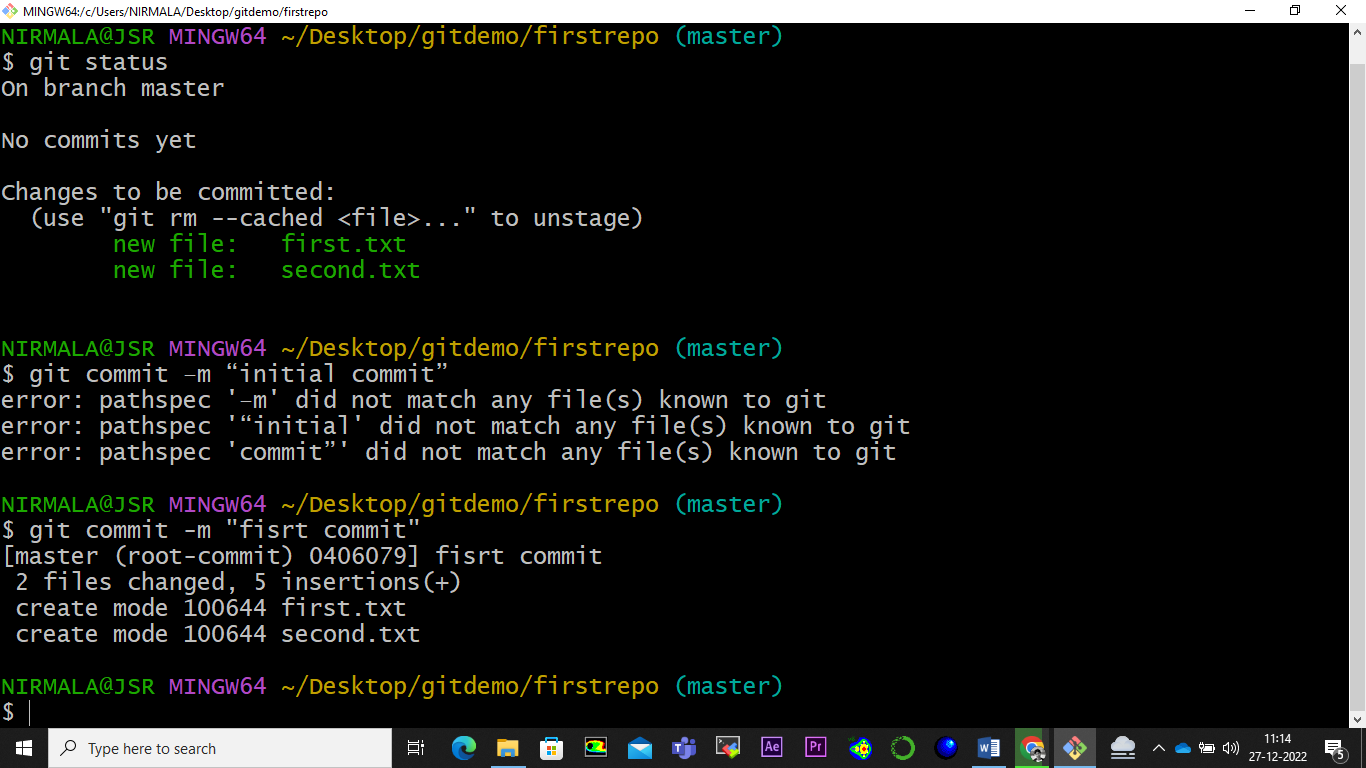
* Add a file to the staging environment: For Git to track the file, use “git add <filename>” command. Once added, check the status with “git status” command.

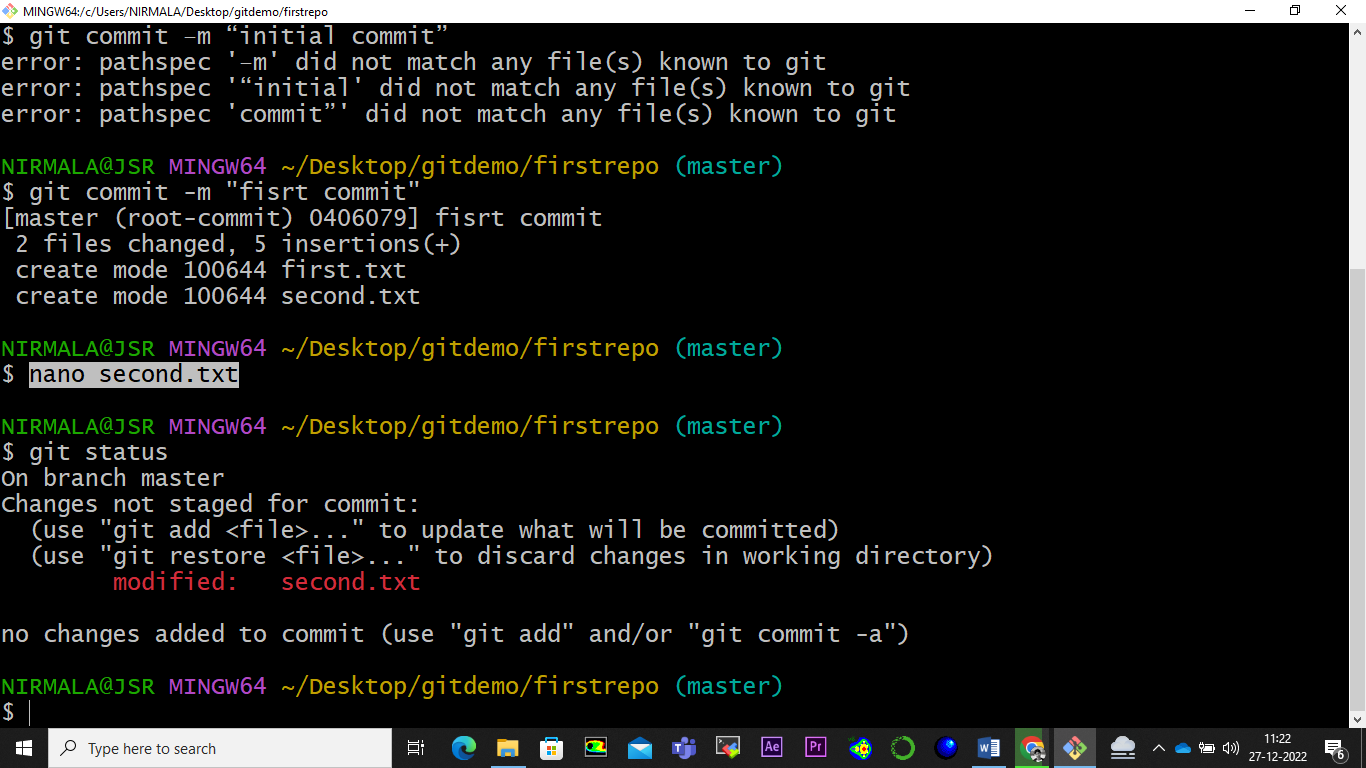


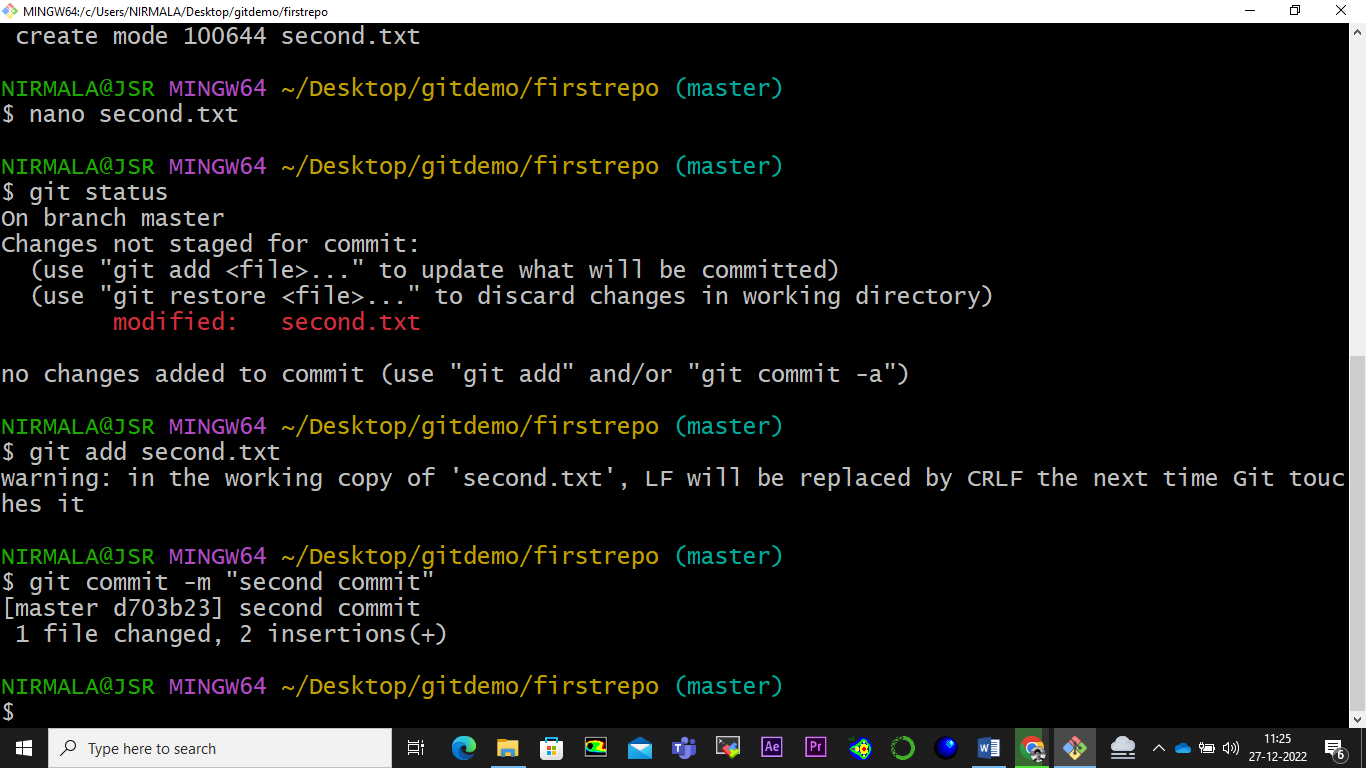
To track second file (or others as well), use command “ git add . ”.



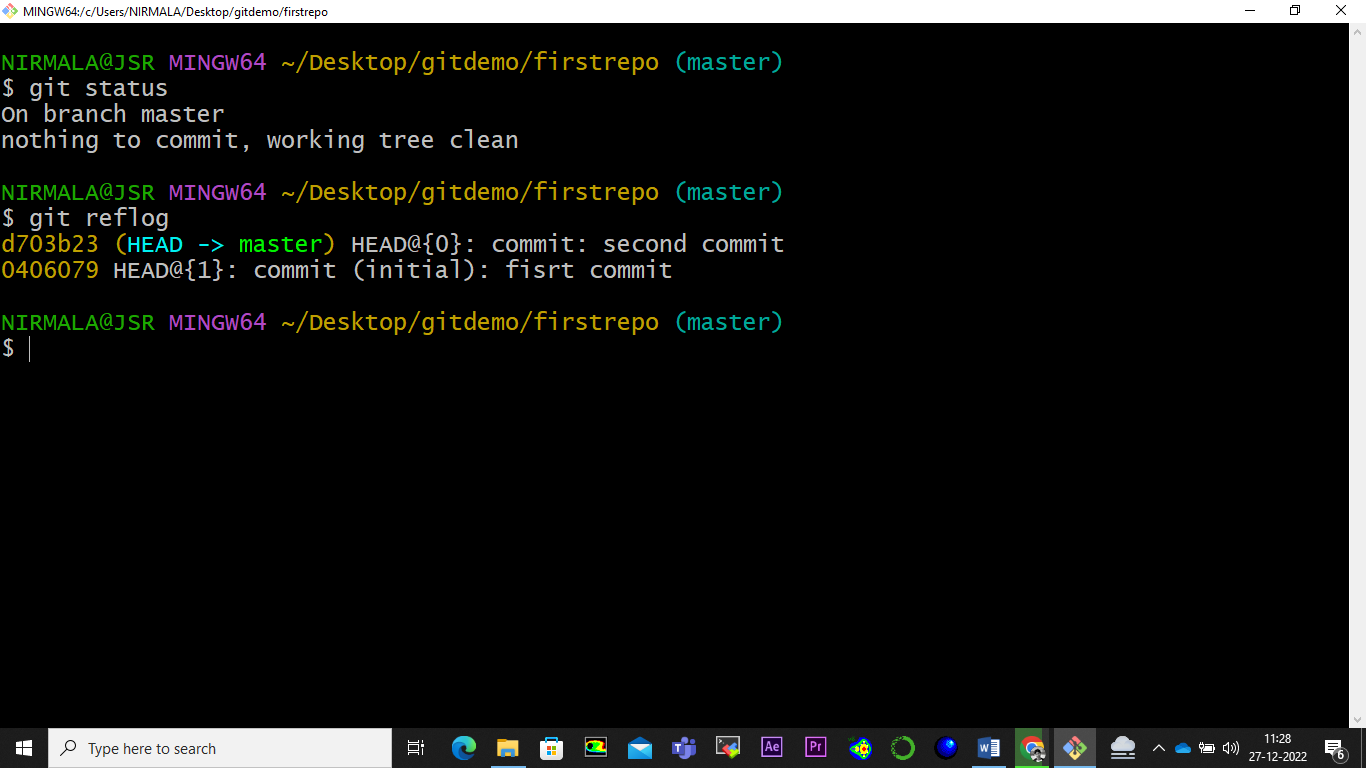
* Create a commit: The next step is to commit the file. Use “ git commit -m "fisrt commit" “.



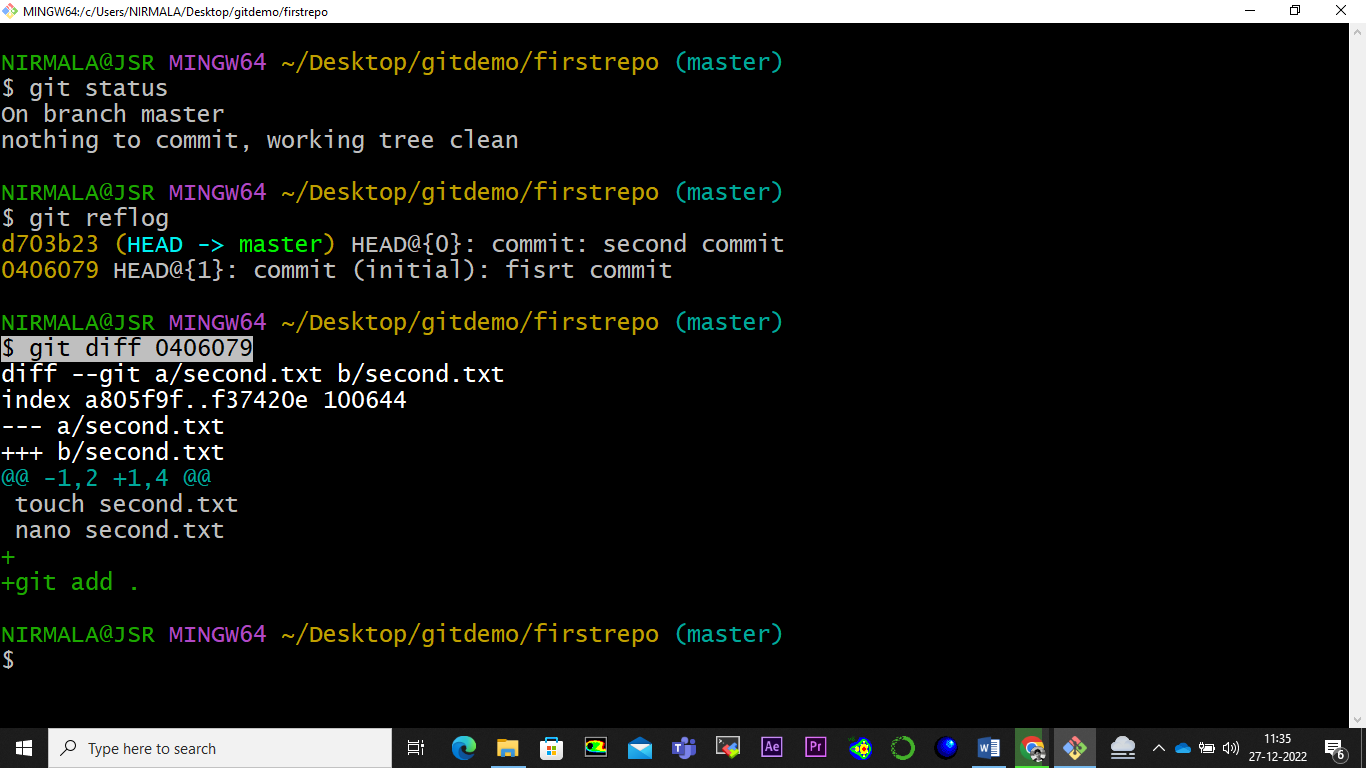




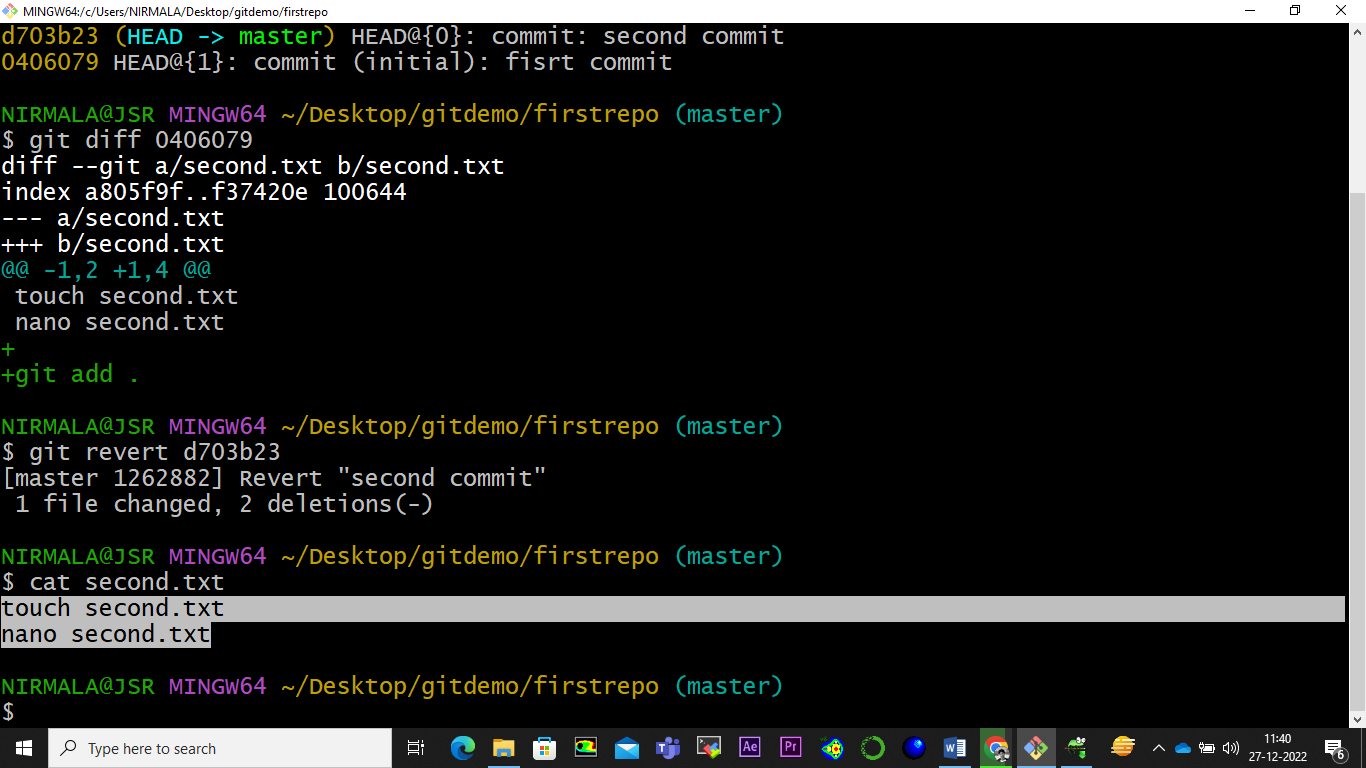
* To track all the commits on the current git repository, use command “git reflog”

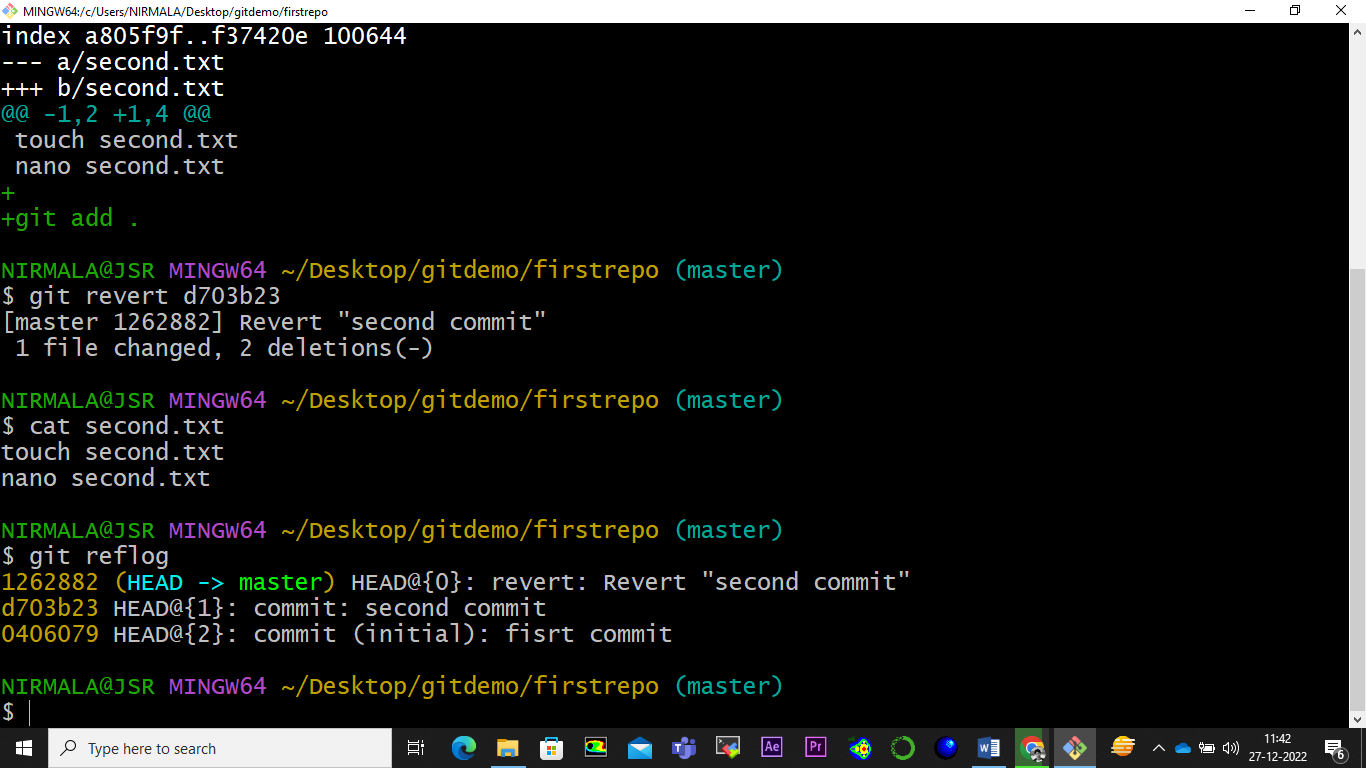


* To track all the commits on the current git repository, use command “git diff <commit\_id>”. Here results shows that we added “git add .” sentence in the second file with new commit.



* To undo the change made in the second file, use command “ git revert <commit\_id> “.





* Create a new branch:

GitHub

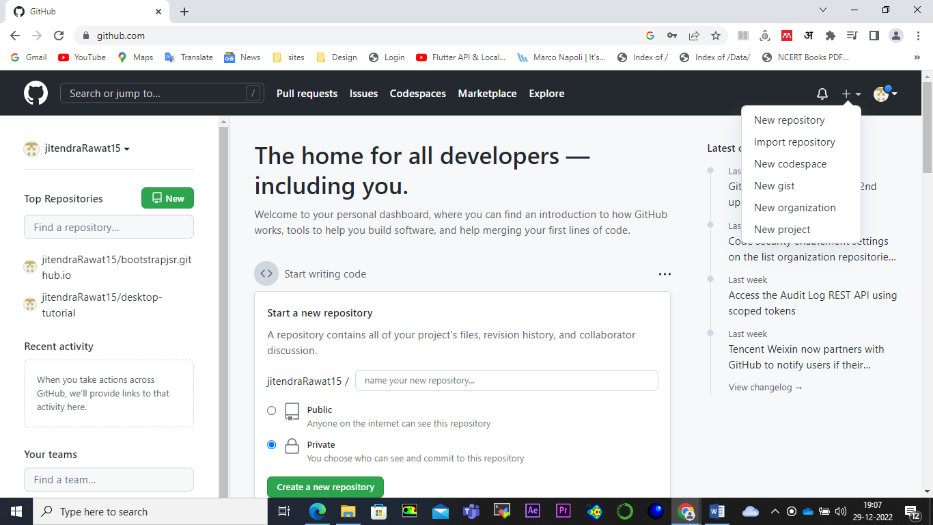
Step 1: Create a GitHub account

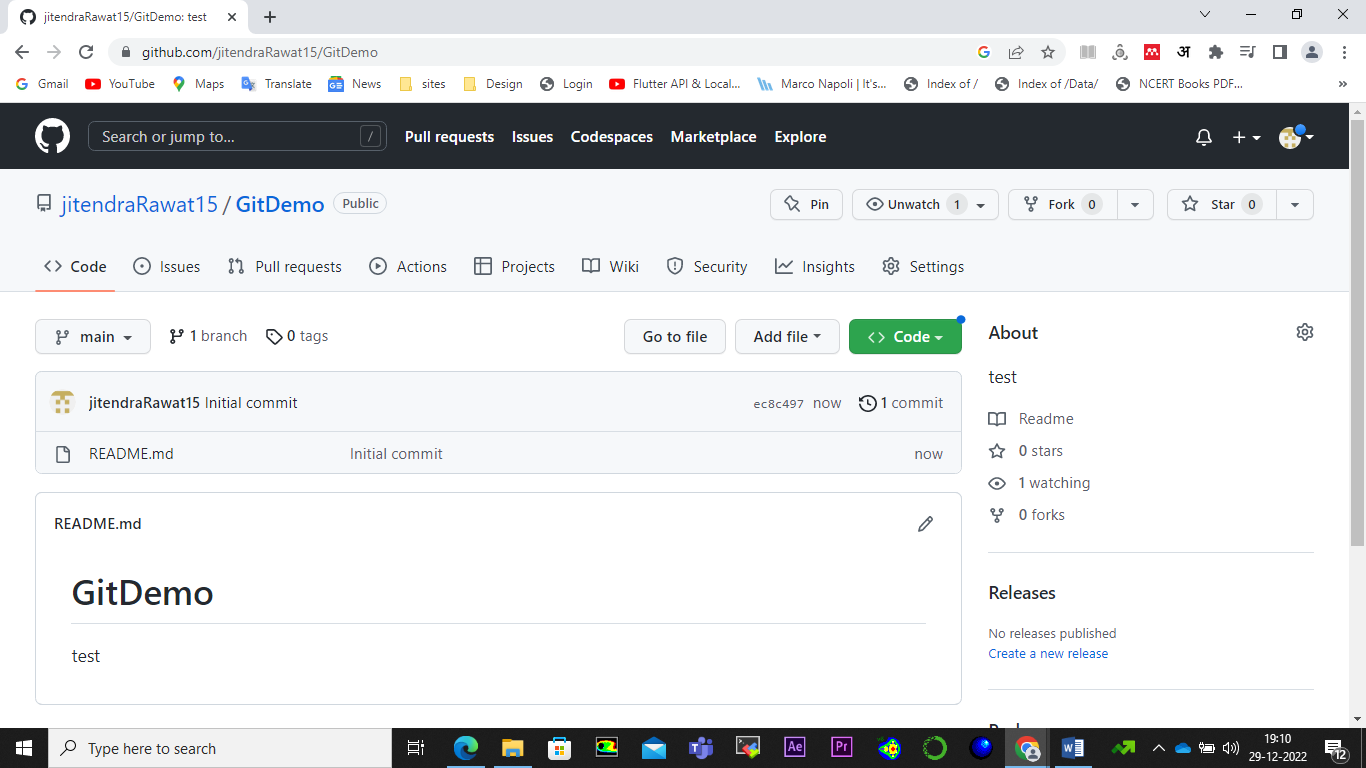
Use <https://github.com/>

Step 2: Create a repository

You can click on the + symbol on the top right corner of the page then choose "New repository"

Give your repo a name and then scroll down and click on “Create repository”

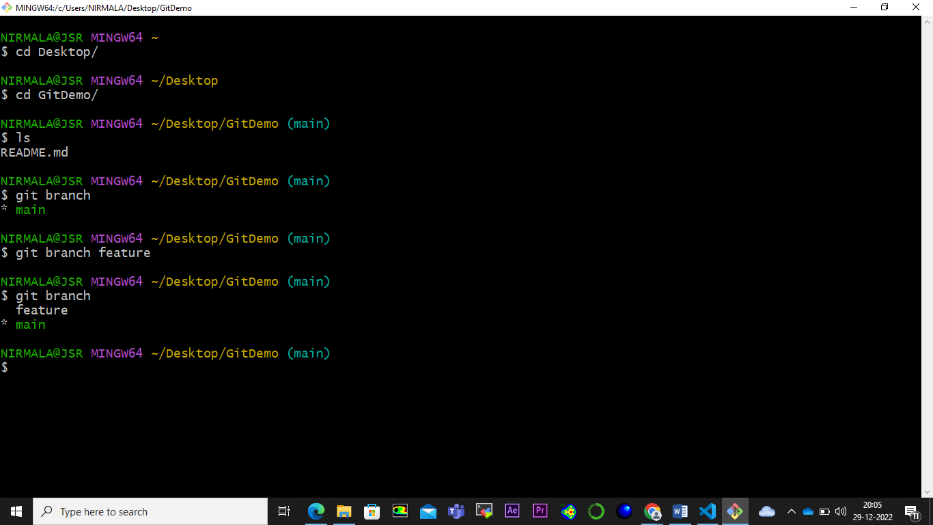




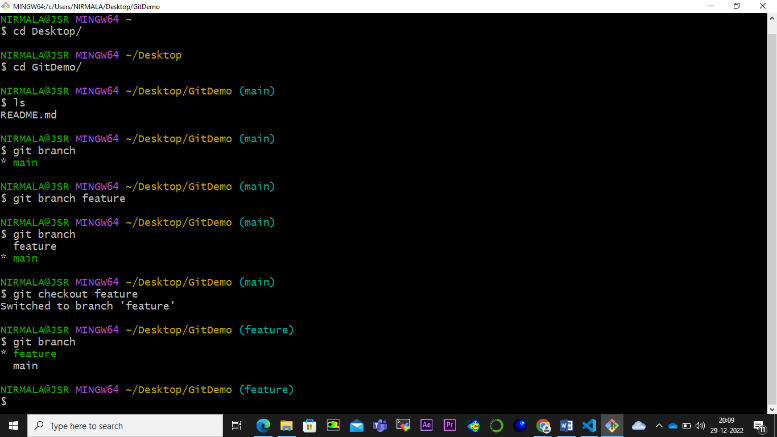
Step 3: Clone the repo to the specific destination (in this case desktop). First navigate to the desktop directory using “cd desktop” command. Now clone the repository to the desktop with following command.

“ git clone <url> “

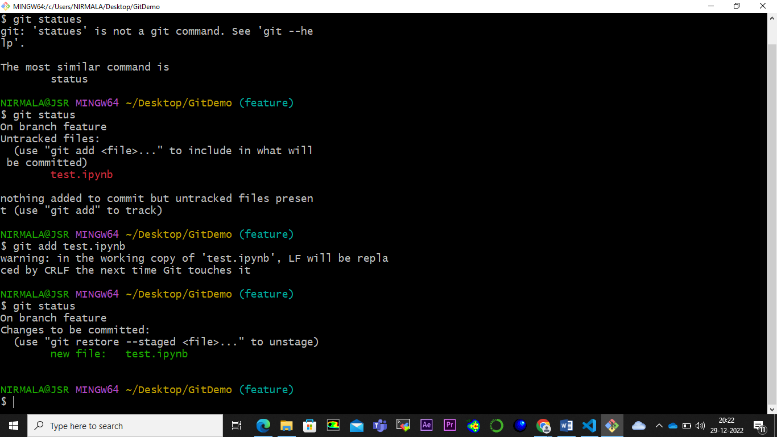
Step 4: Create a new branch using “ git branch <branchname> “

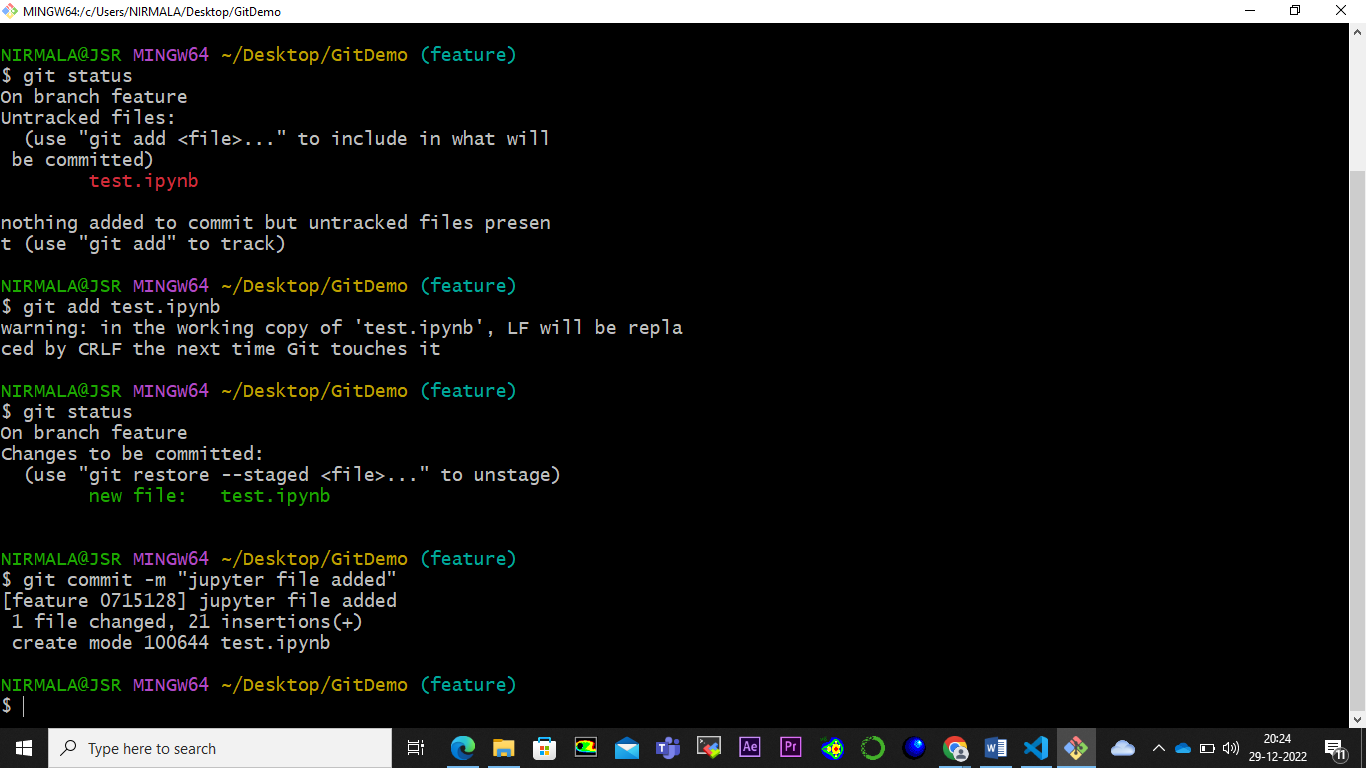


Step 5: Last command shows that main branch is the active branch. To work on feature branch, I must checkout to feature branch. Use “ git checkout feature “.

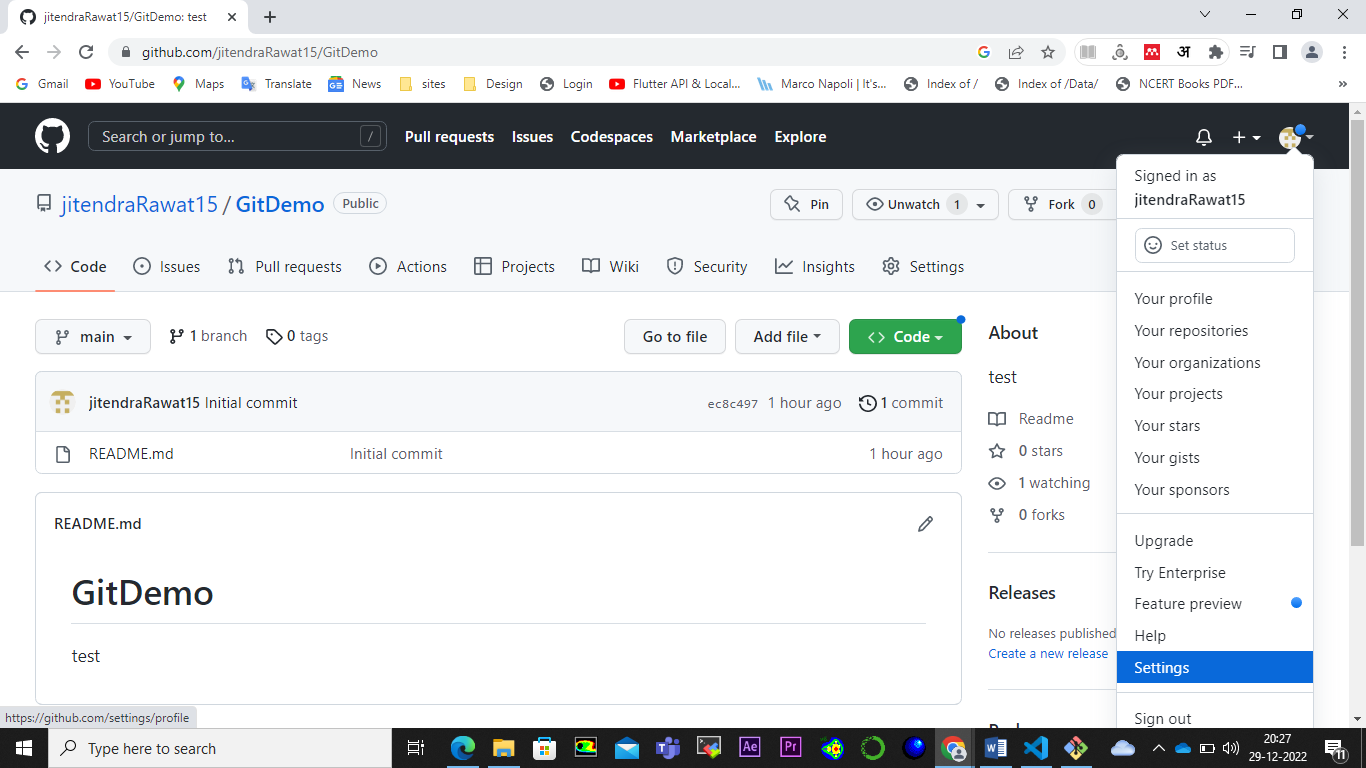


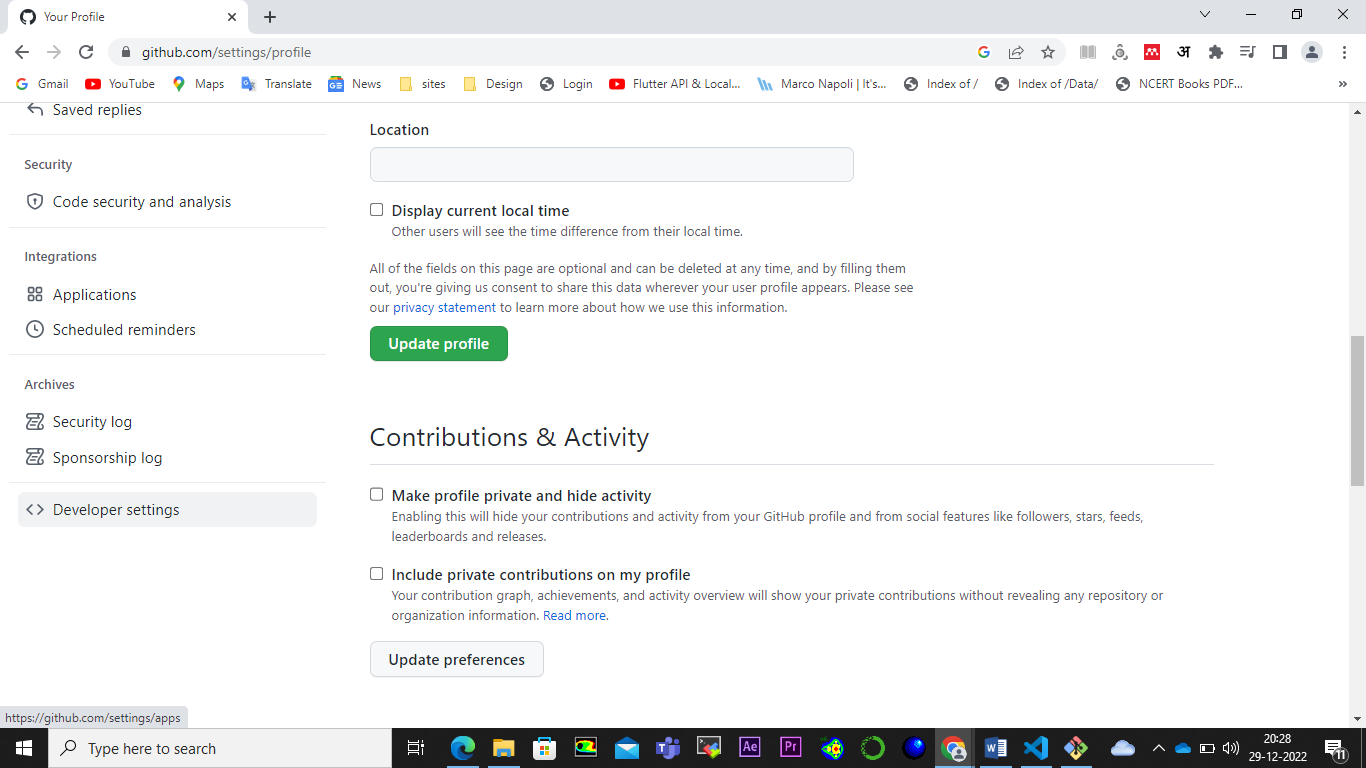
Step 6:

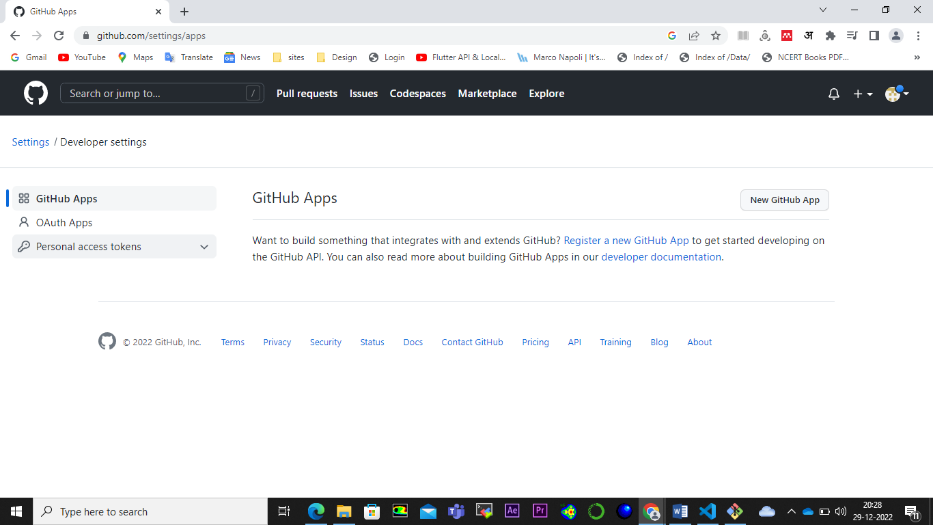


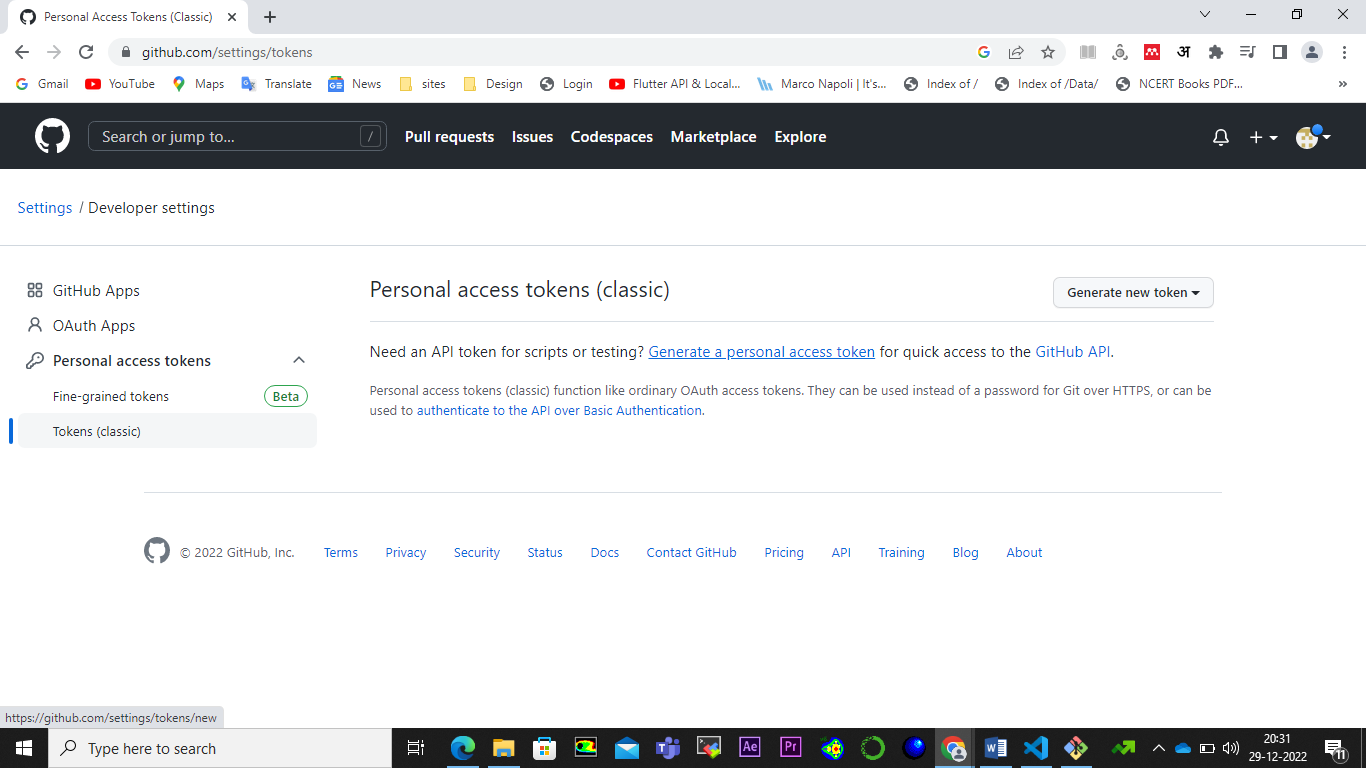


Step 7: Now we have to push the new file into GitHub repo. So we have to use “token”. To generate token go to setting 🡪 Developer setting 🡪 Personal access tokens (PATs) 🡪 Tokens 🡪 Generate a personal access token



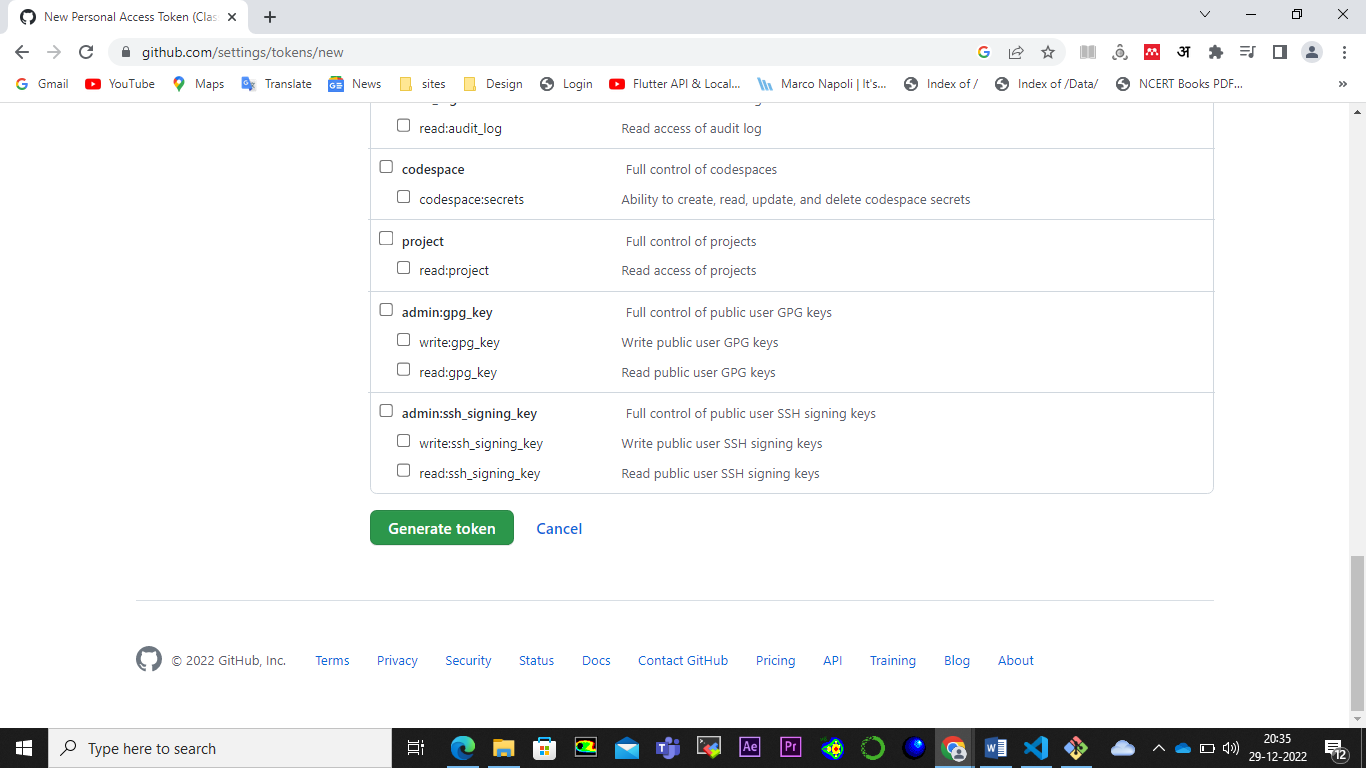


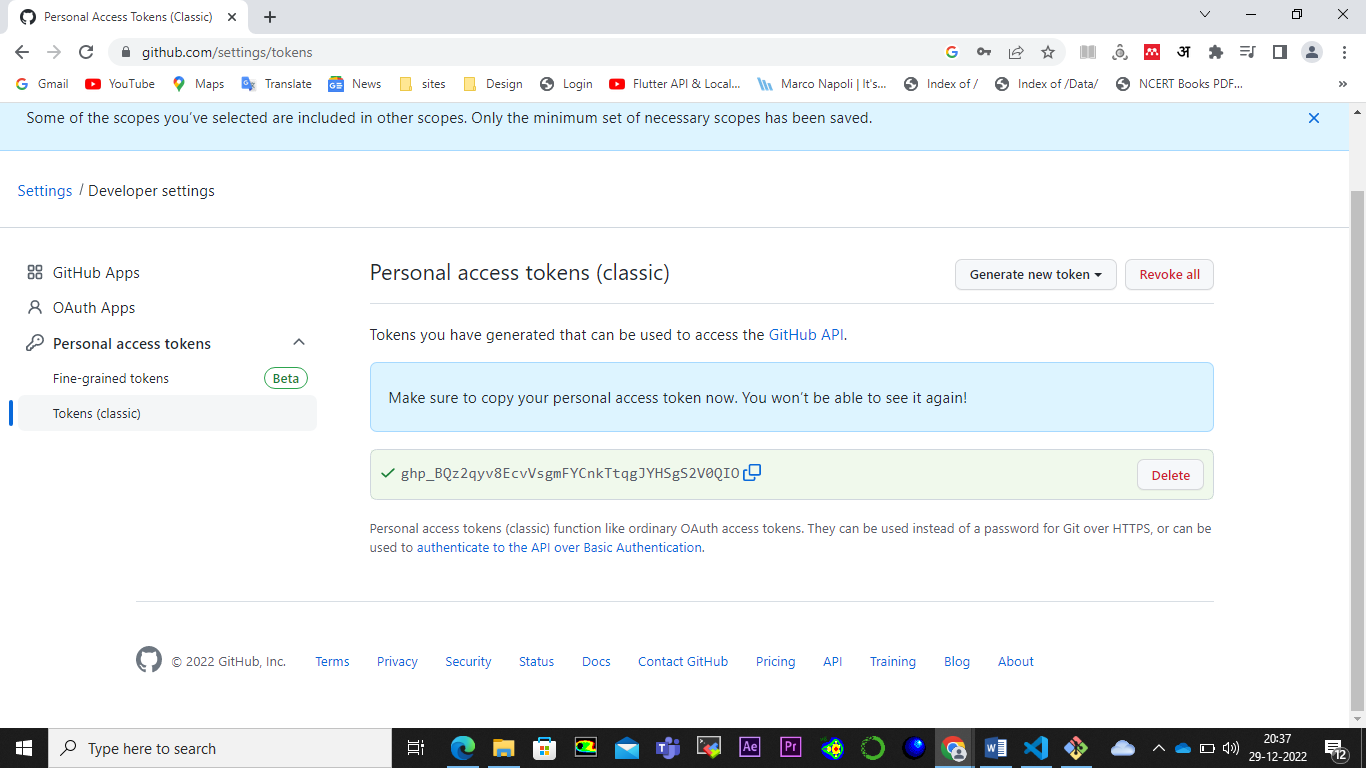




Fill the note as “test” and check the repo, and scroll down to click on Generate token button.



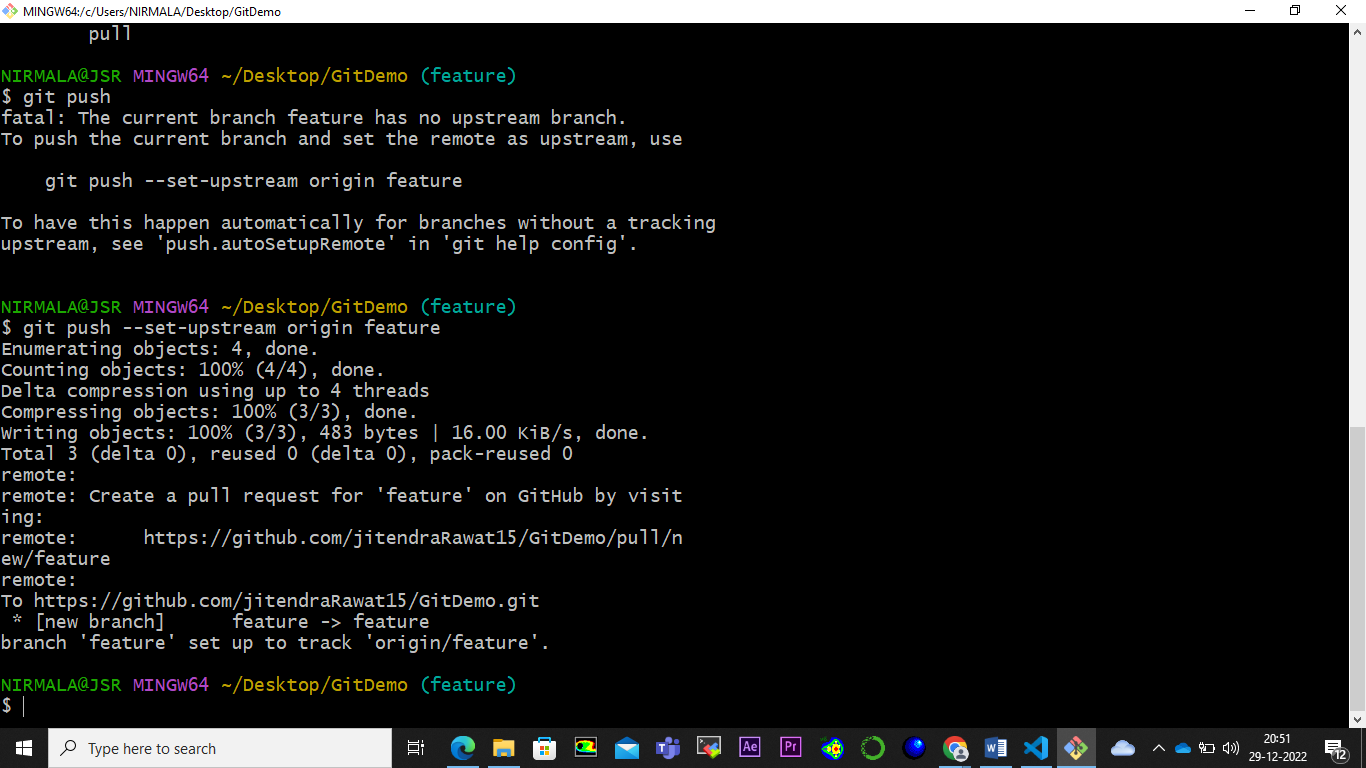


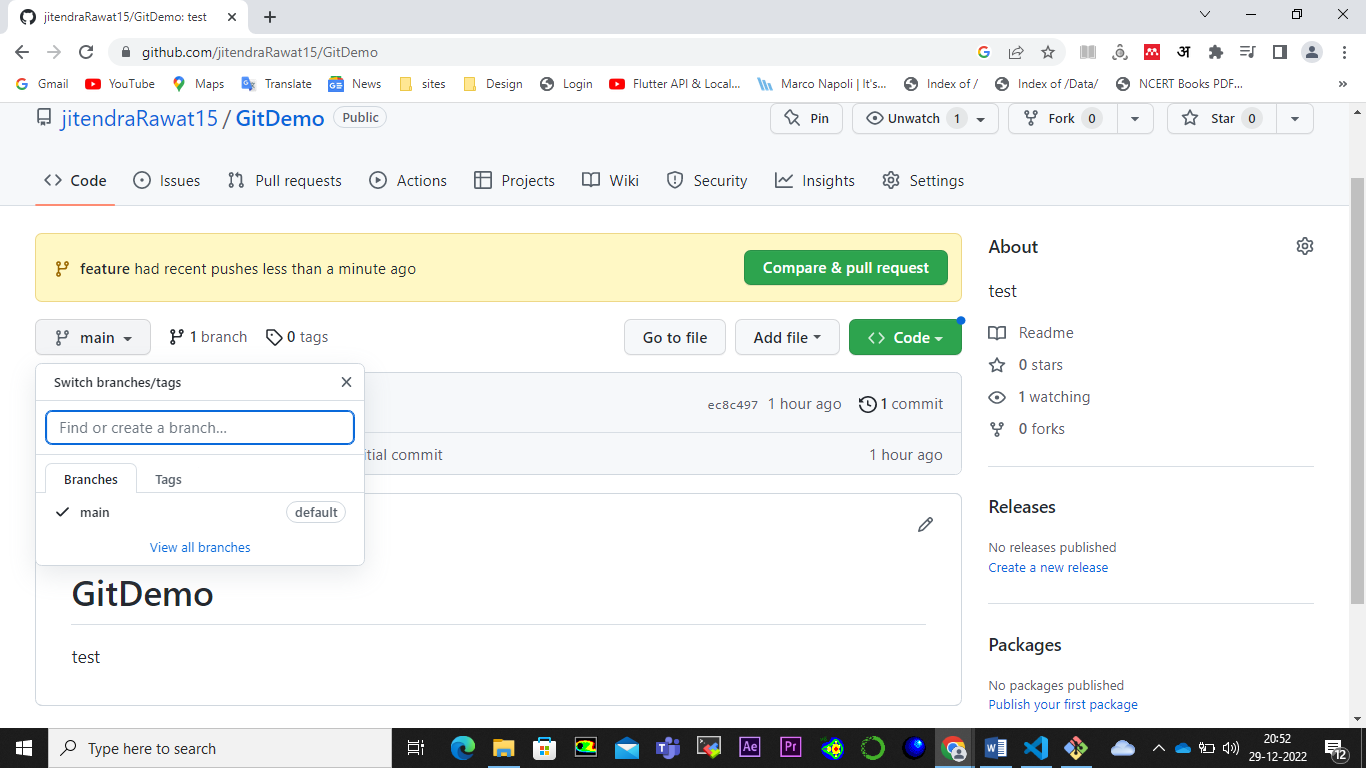


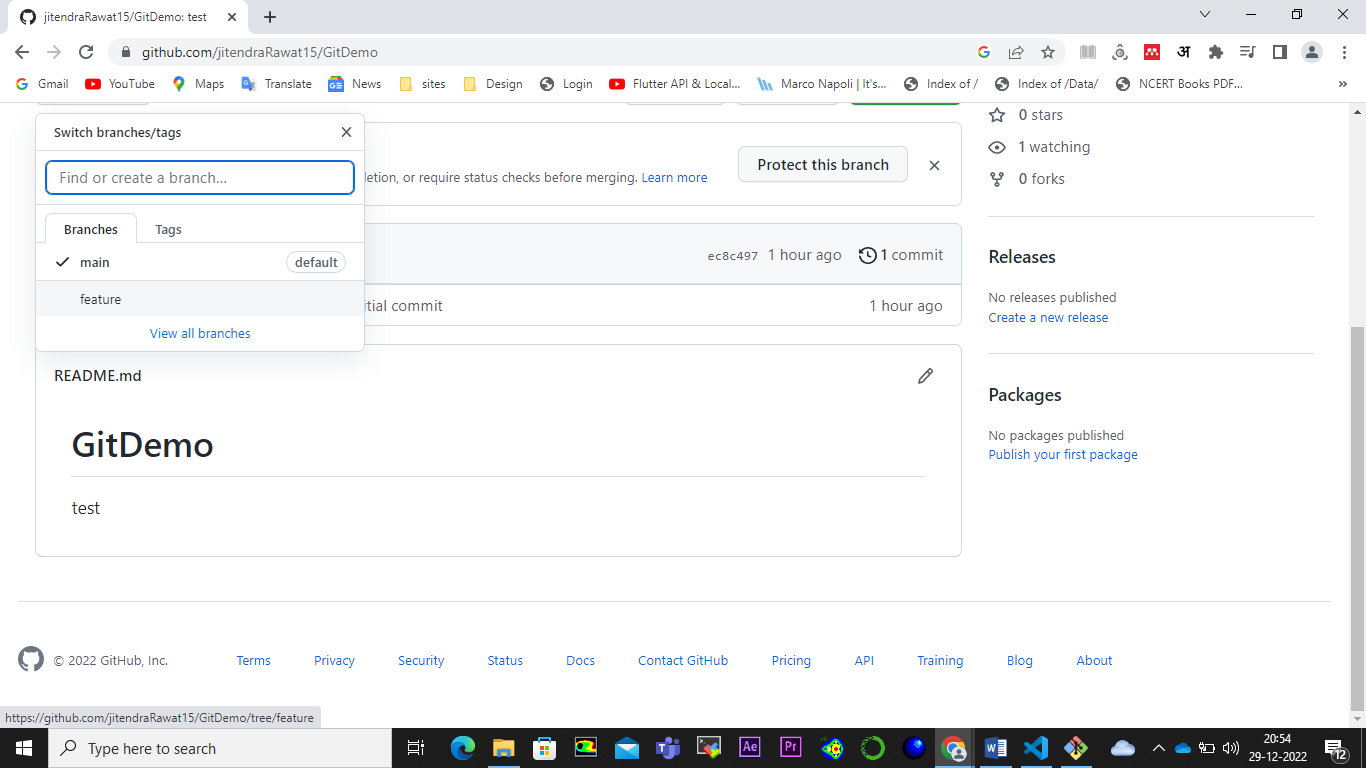
Step 8: After the token is generated. Copy it. To push the new branch in the GitHub, use command “git push “.

It shows tha error. Hence, use the following command:

git push --set-upstream origin feature

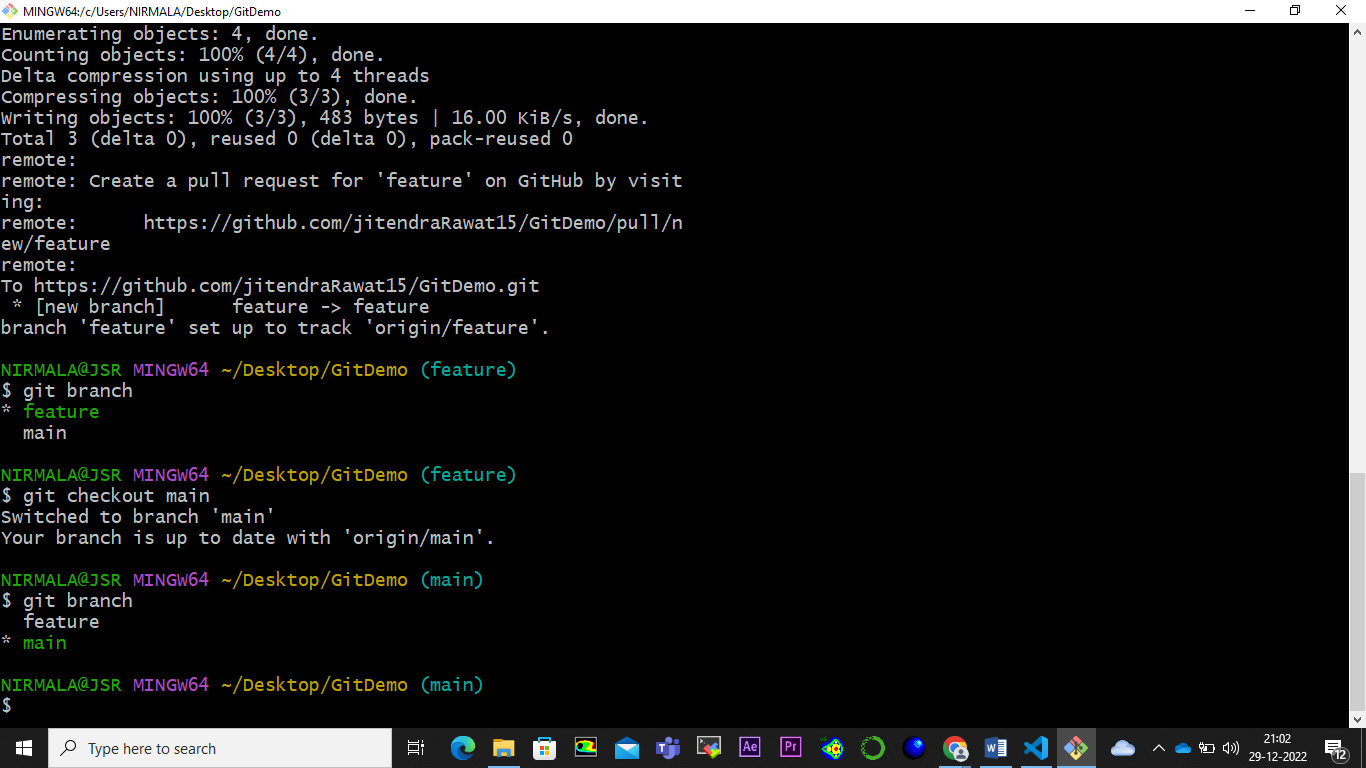




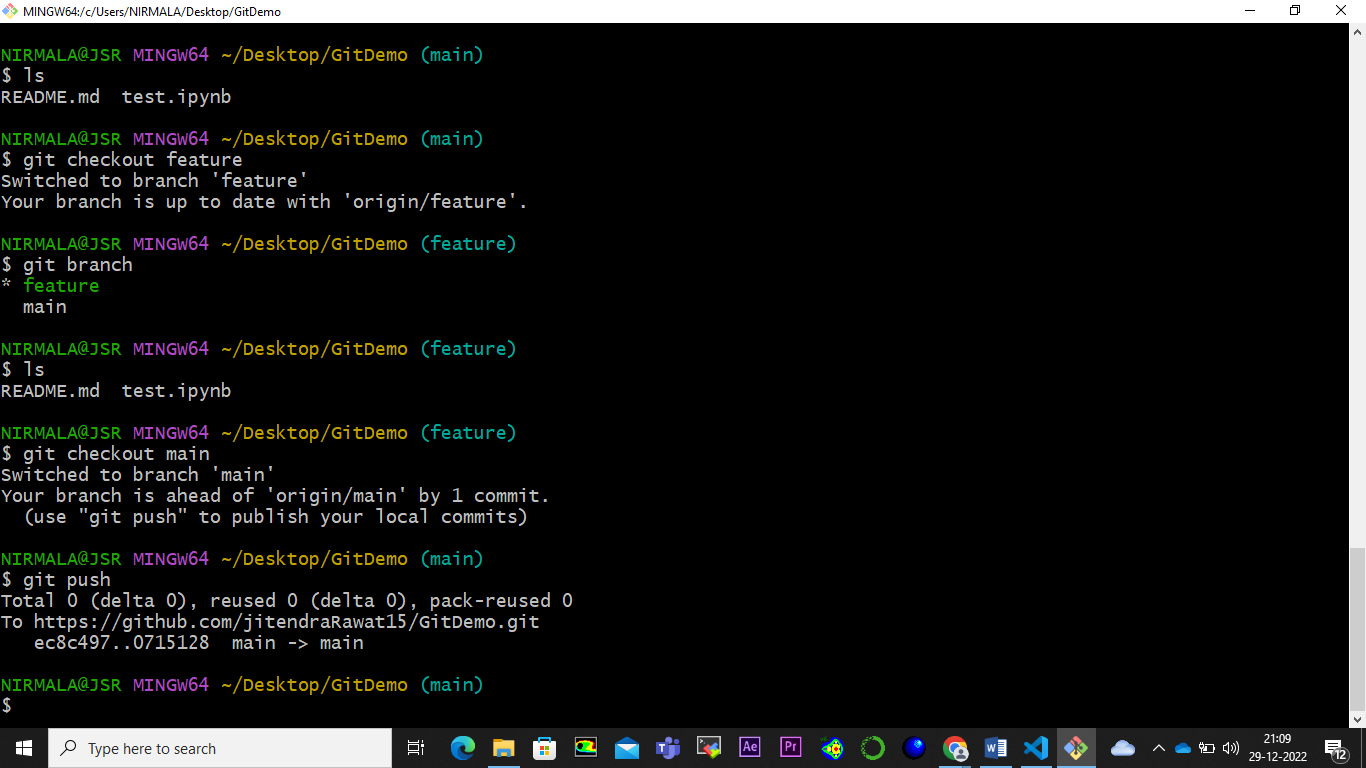


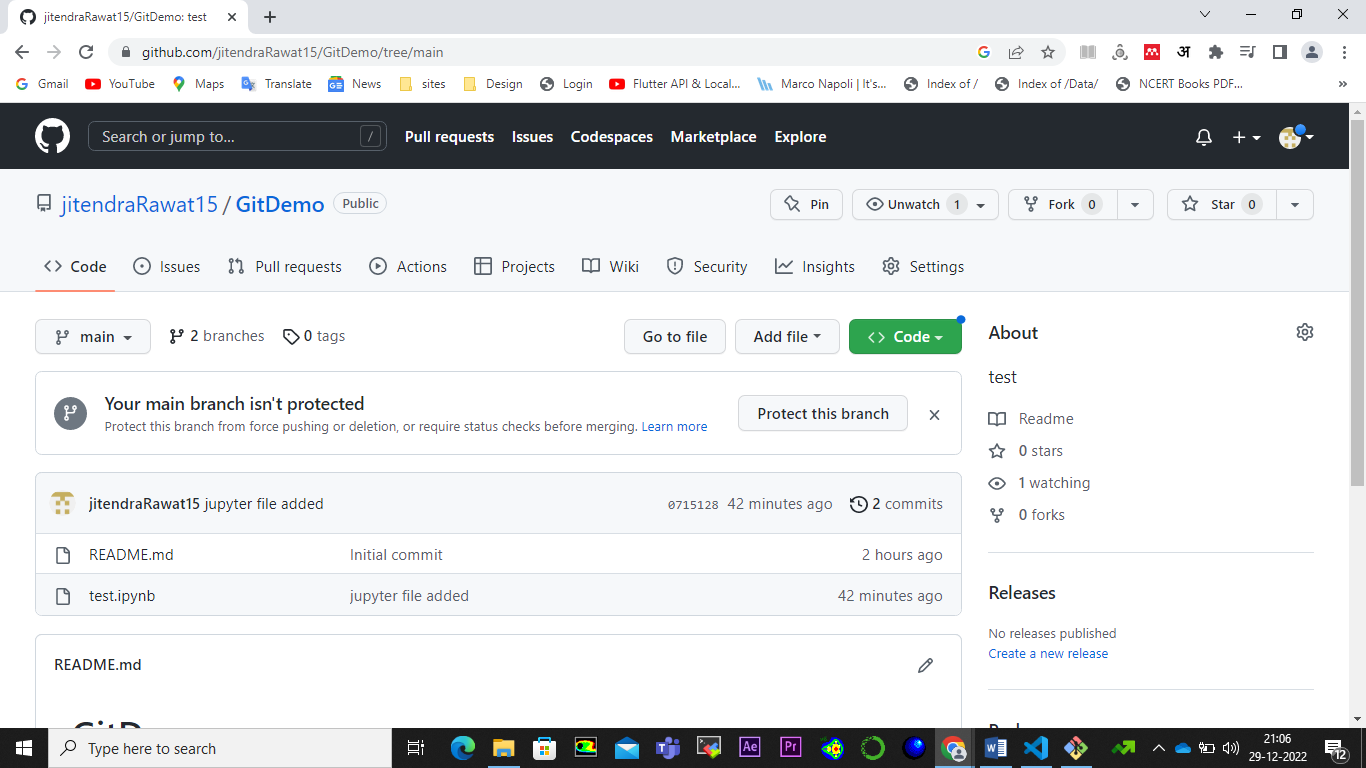
Step 9: Now we added the new branch to the repo. Next, we will merge “feature” branch to the “main” branch. To do so, you should keep in mind that you should be on the SAME branch in which you want to merge the “feature” branch. As we want “feature” branch to be merge with “main” branch so we will use following command:

* git checkout main
* git merge feature
* git push ( to publish in GitHub)





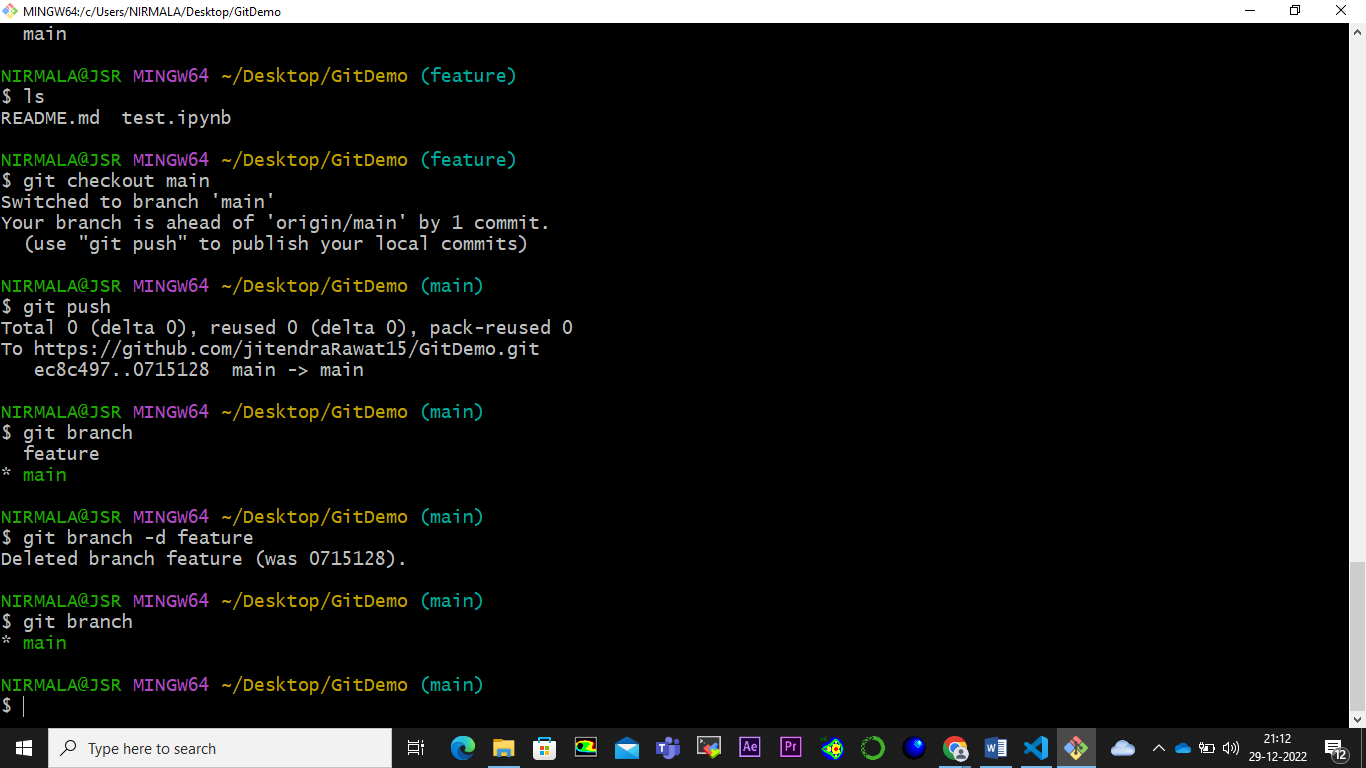




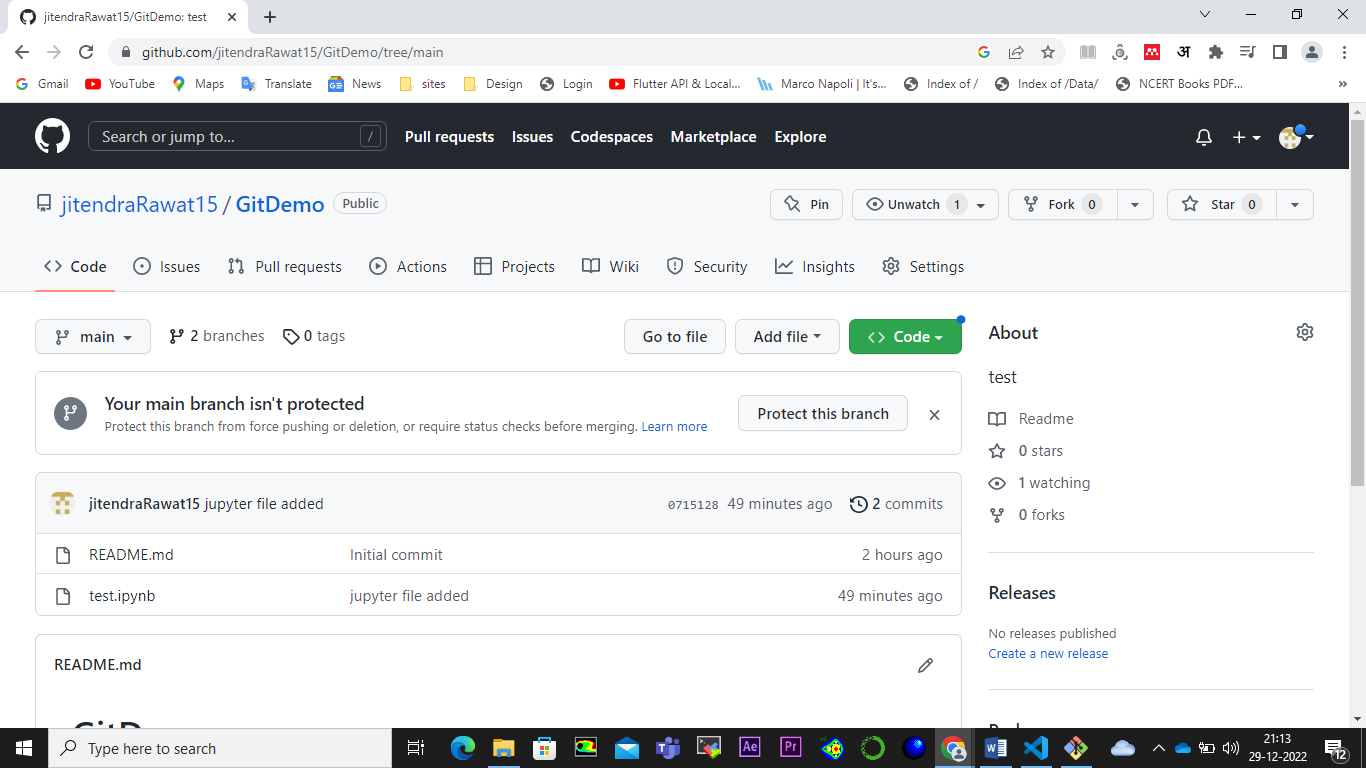
It is now published at GitHub main branch too.

Step 10: Next step, we will delete the feature branch from local repo. Use this command as follows:

* git branch –d feature

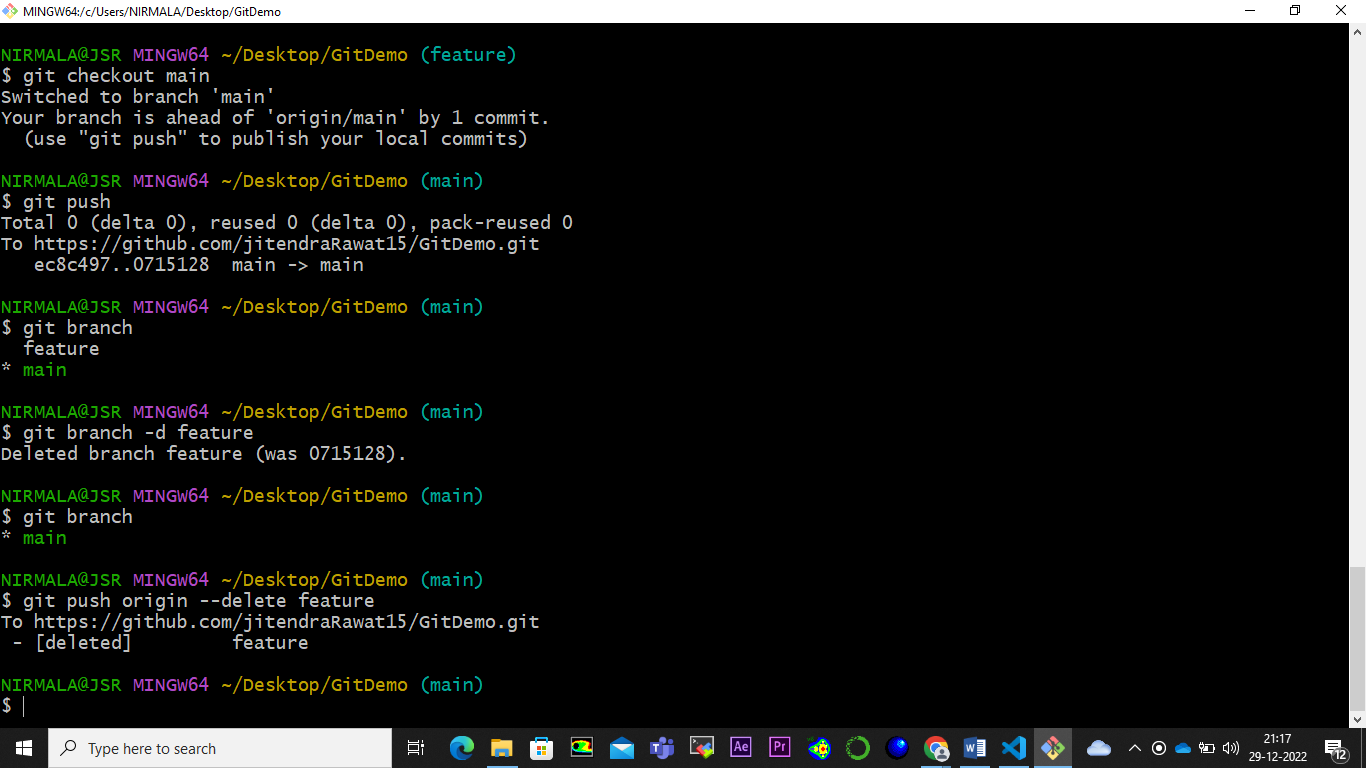


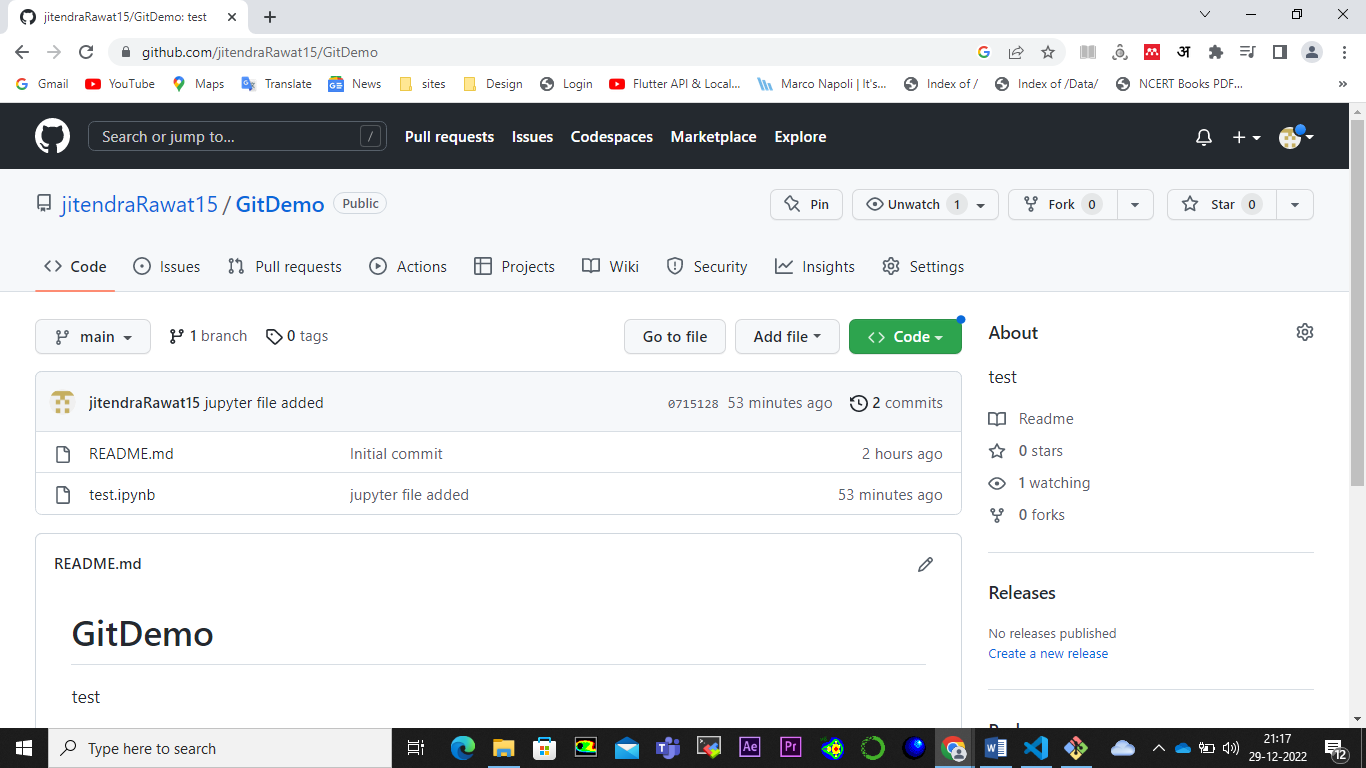
But in GitHub, the feature branch is still there.



Step 11: To delete feature branch from github, use the following command.

* git push origin –delete feature





This is for this session.

Thank you all