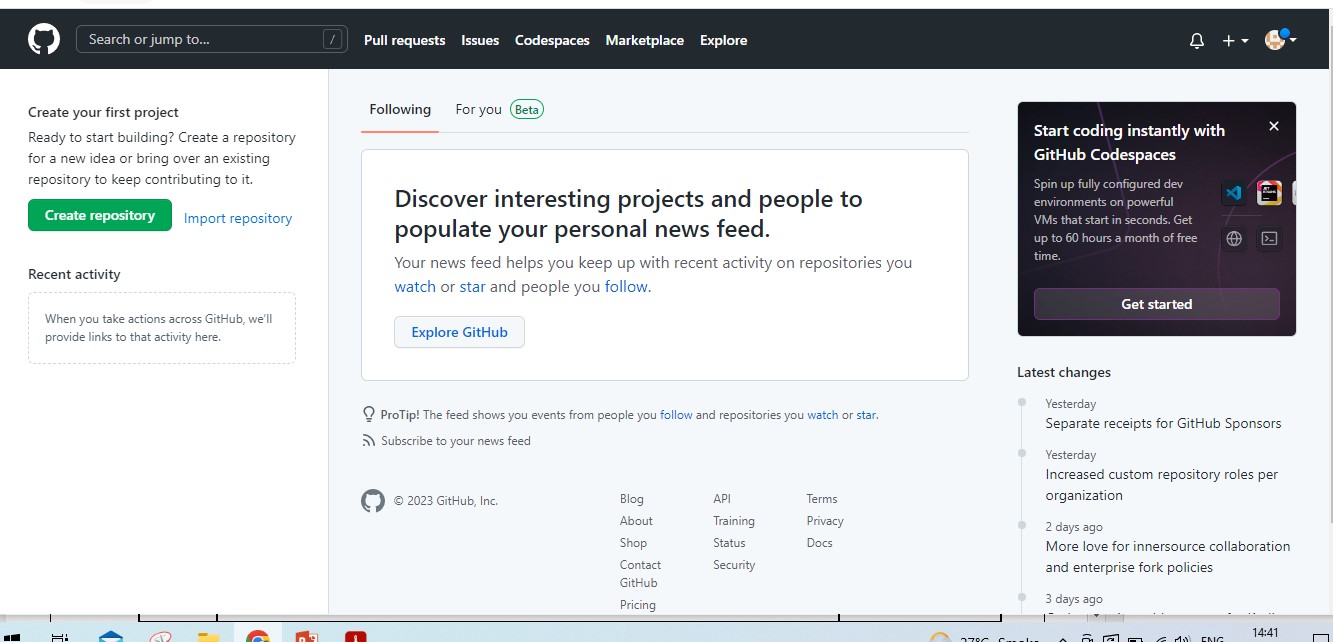
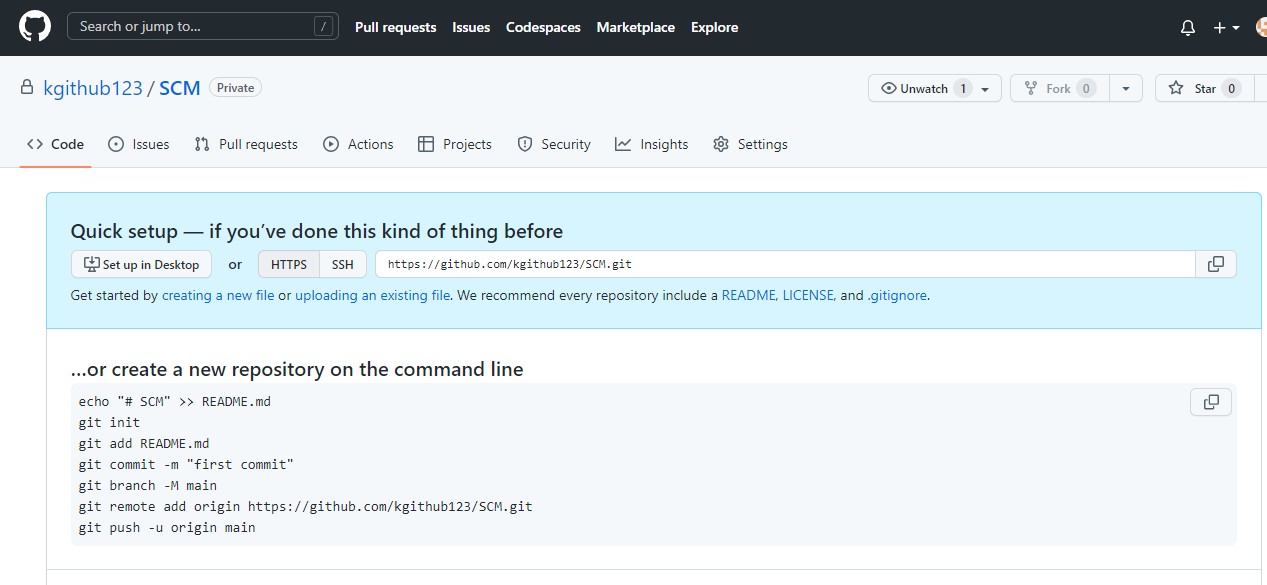
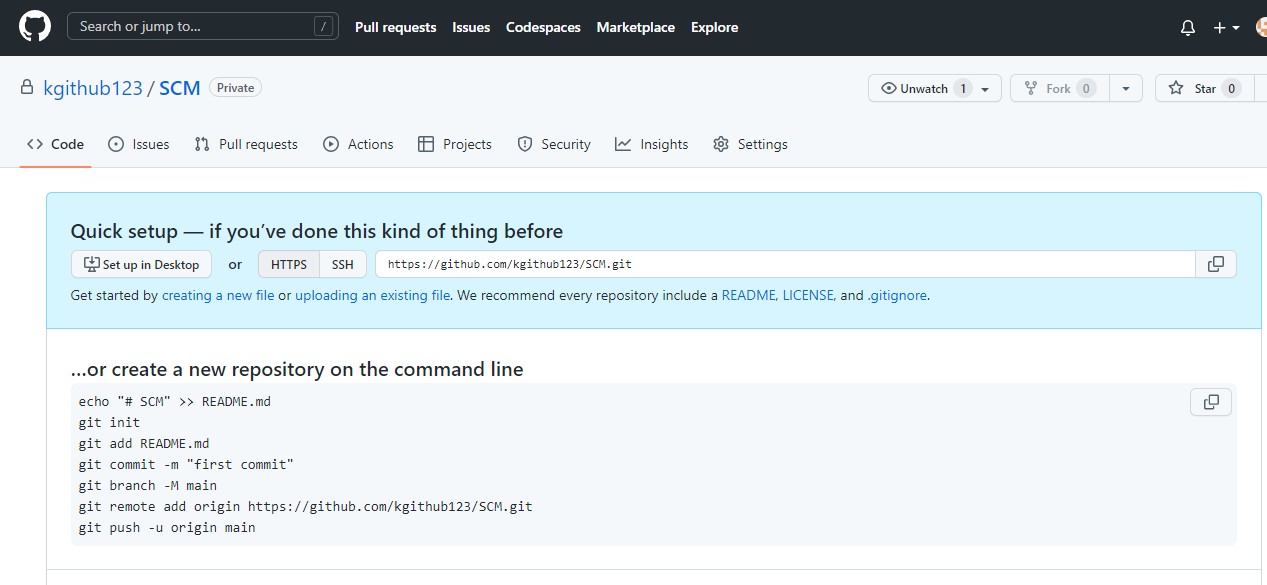
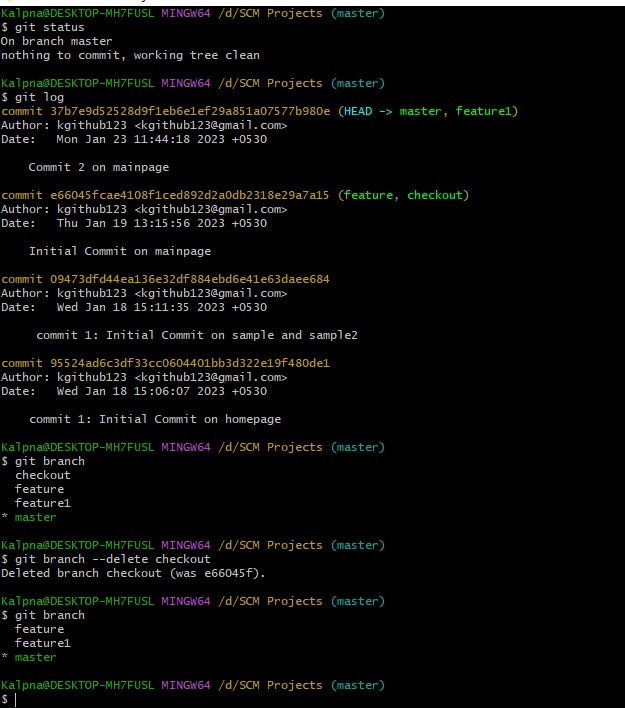
# EXPERIMENT NO. 06

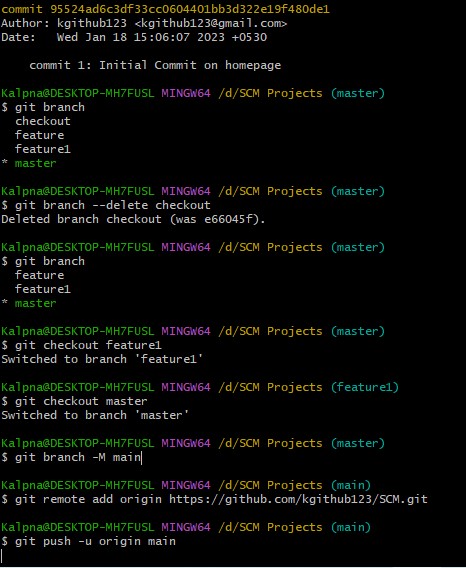
**Aim: Add collaborators on GitHub repository**

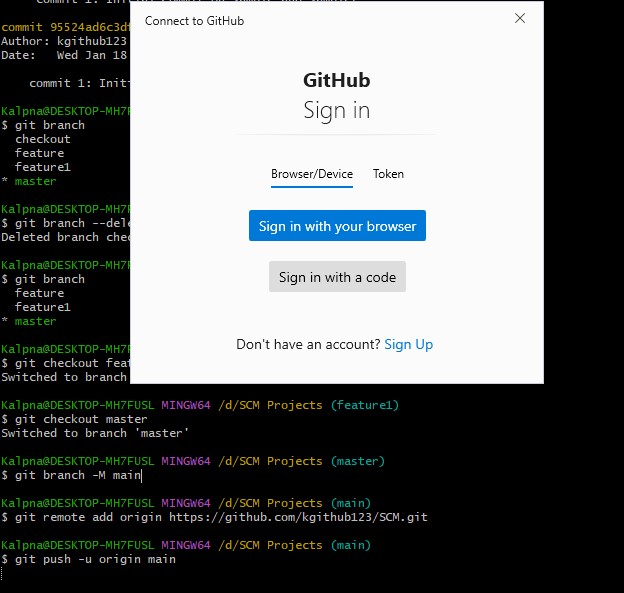
## LOGIN AND CREATE A REPOSITORY IN GITHUB

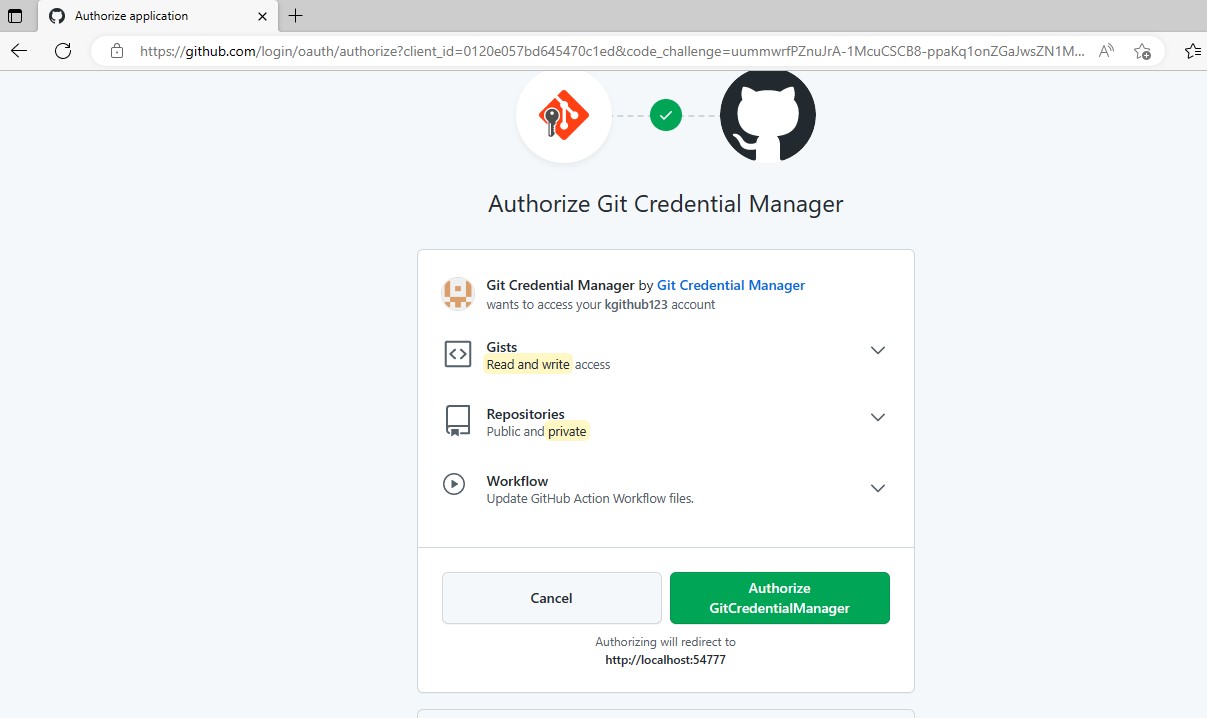


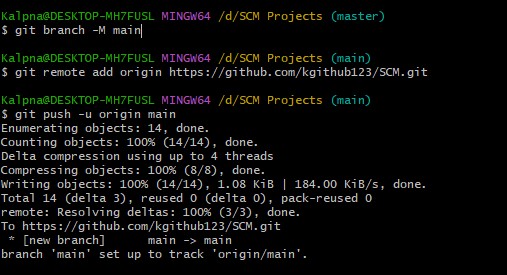




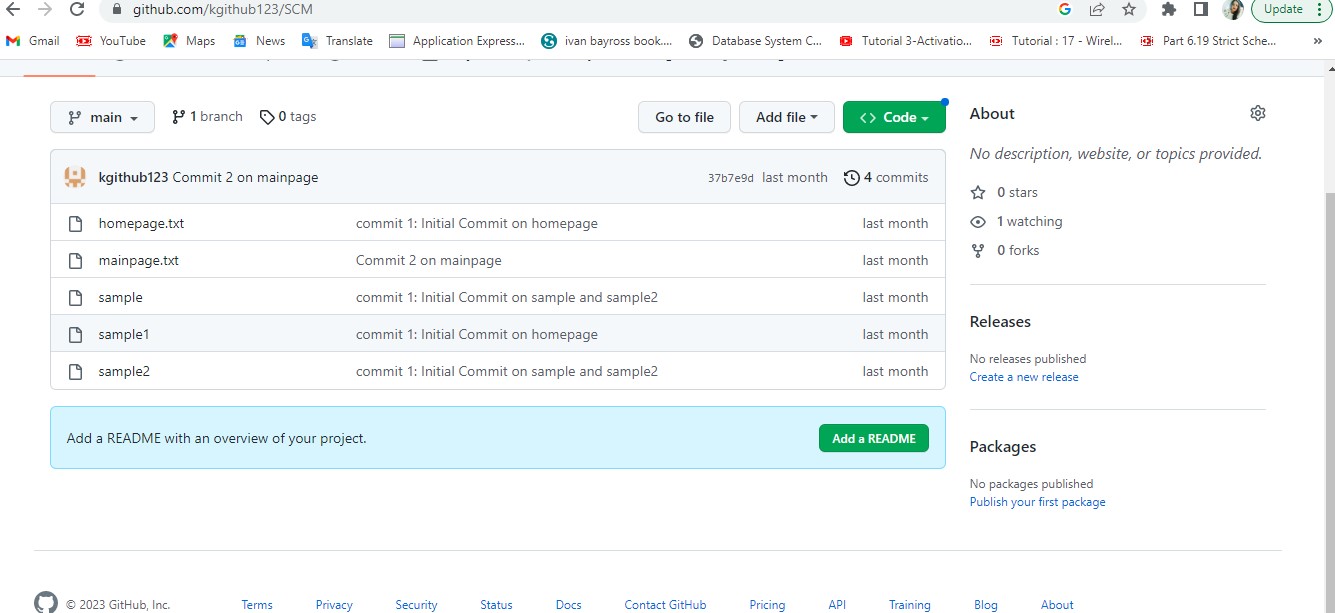
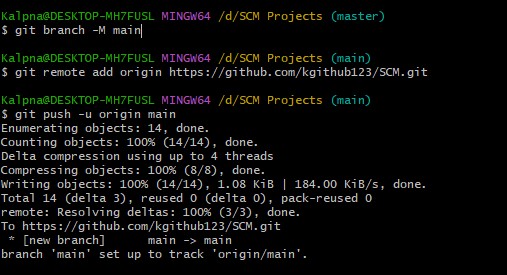


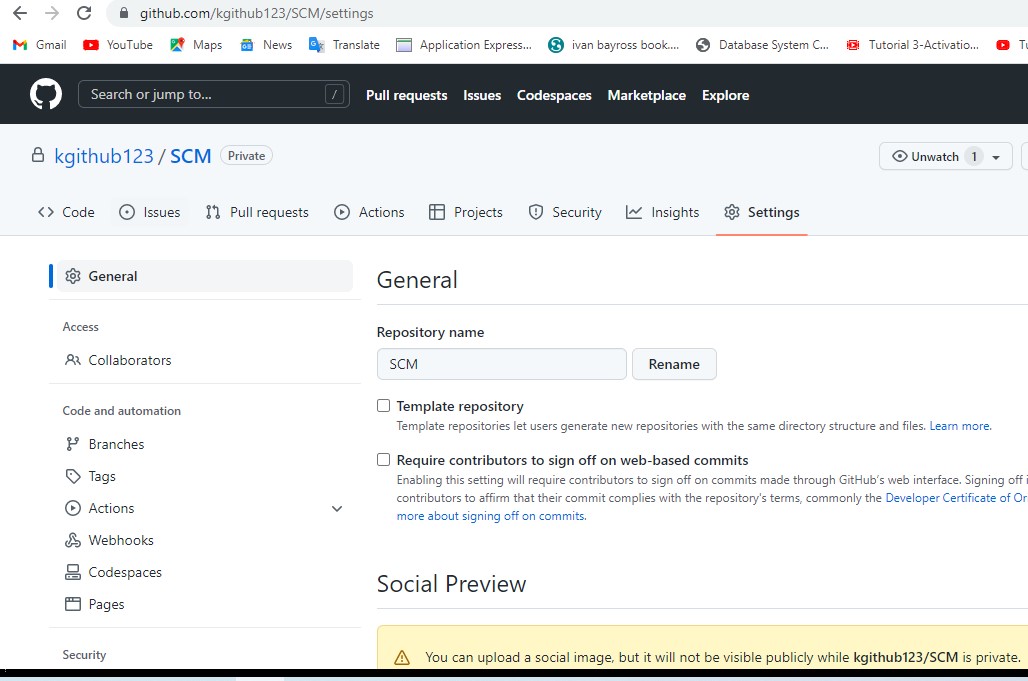


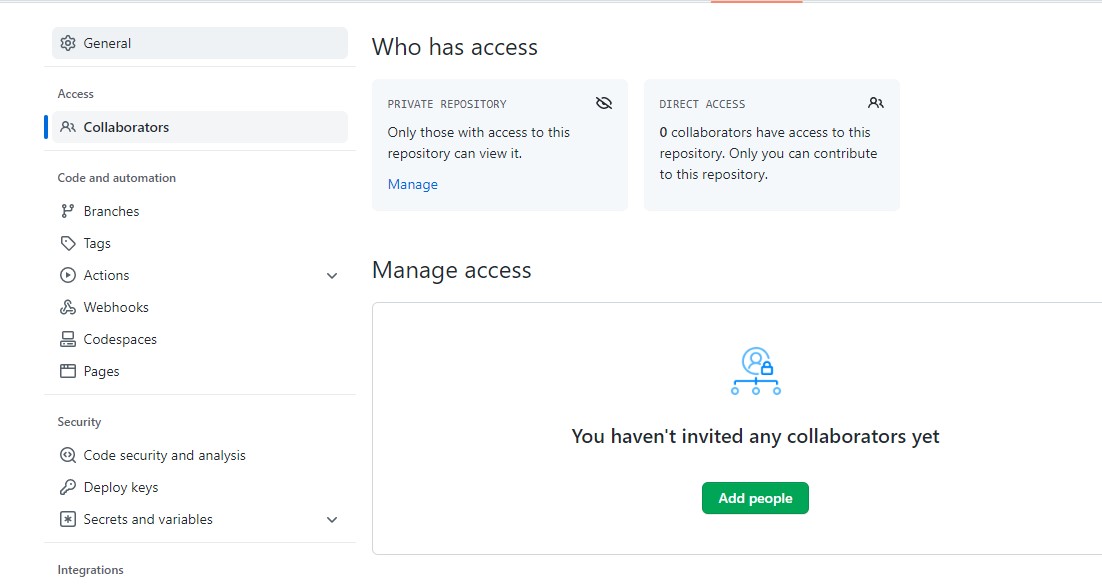


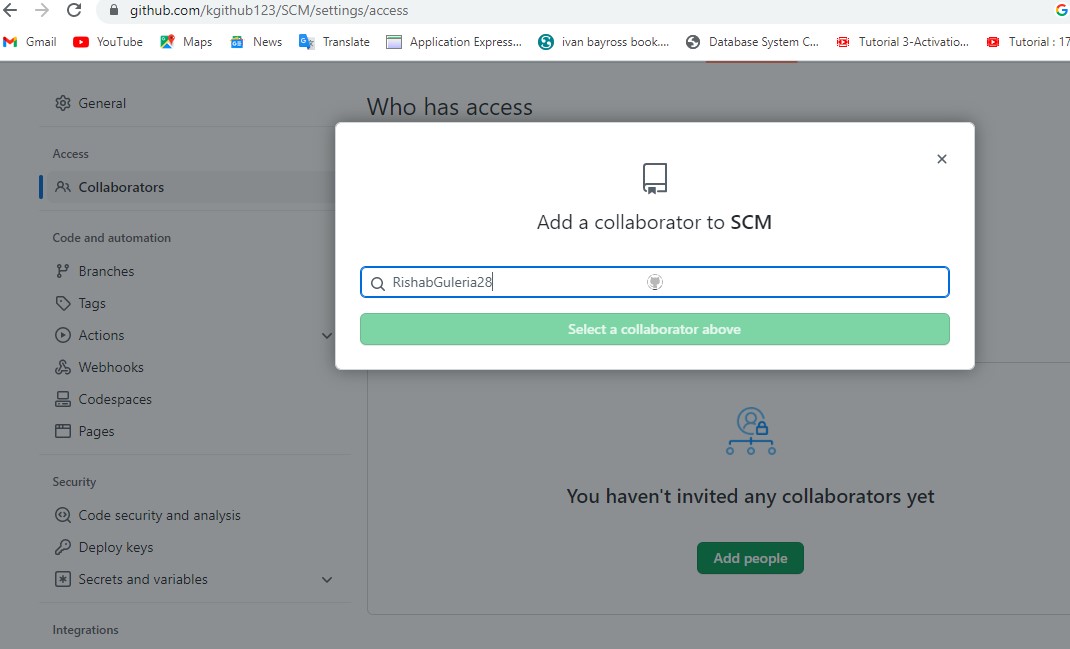


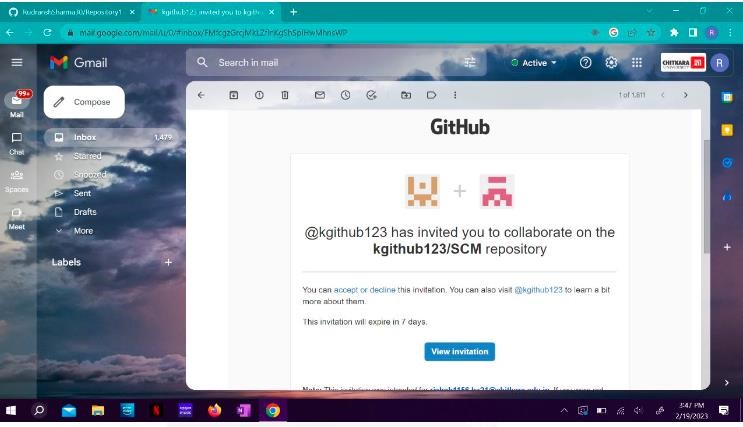
231099

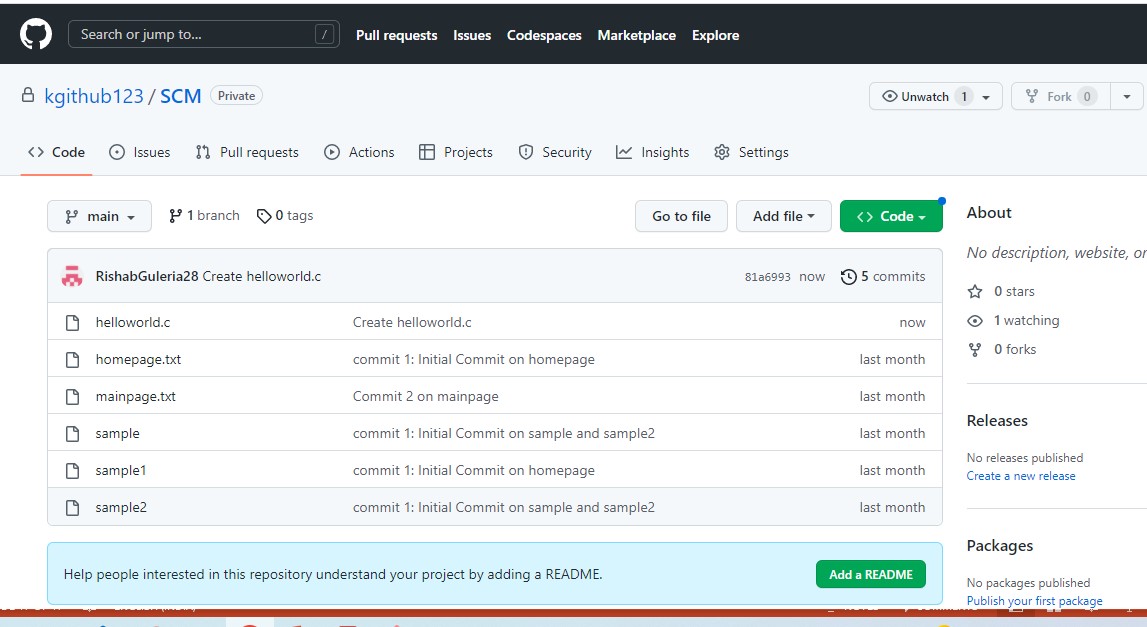
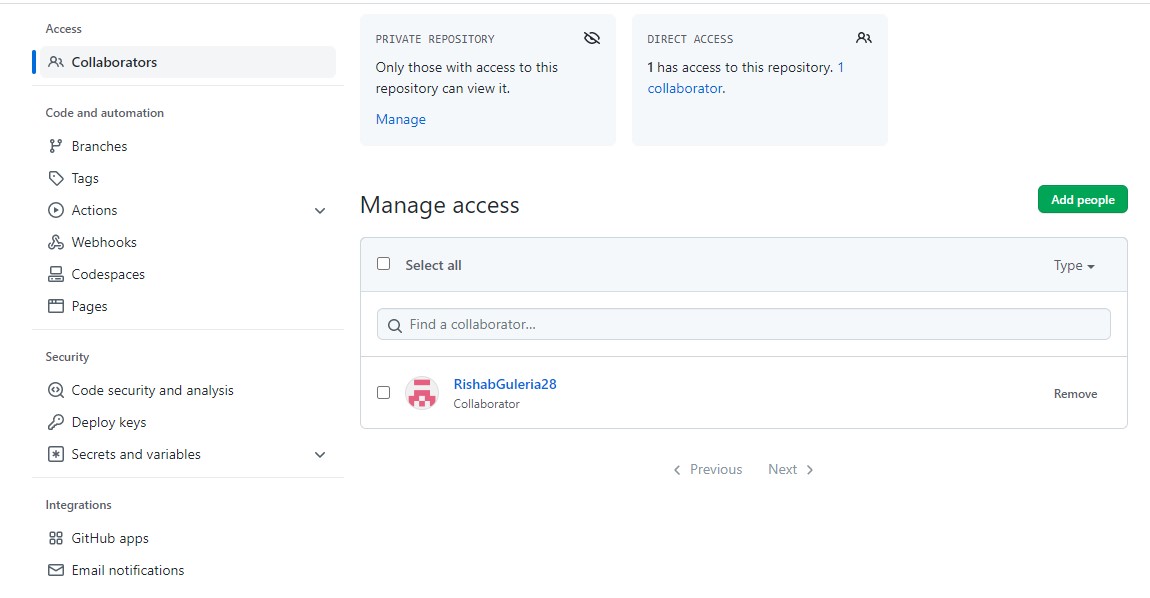
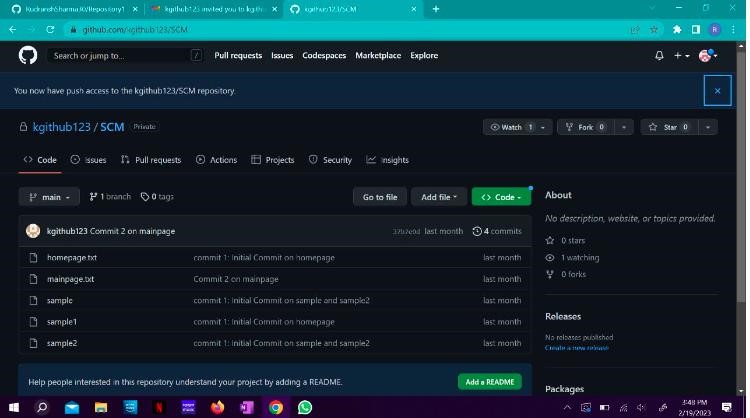
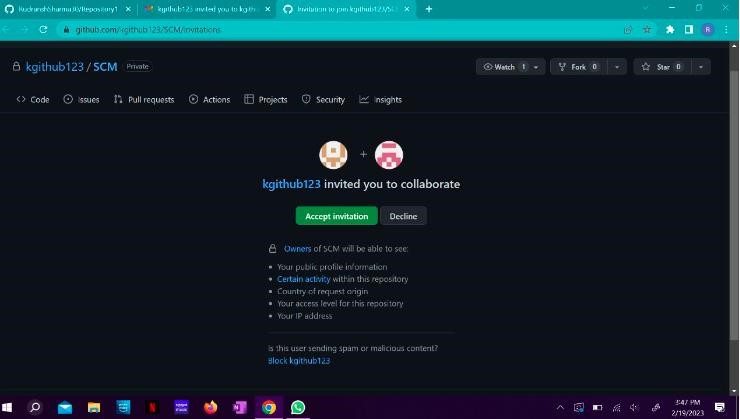


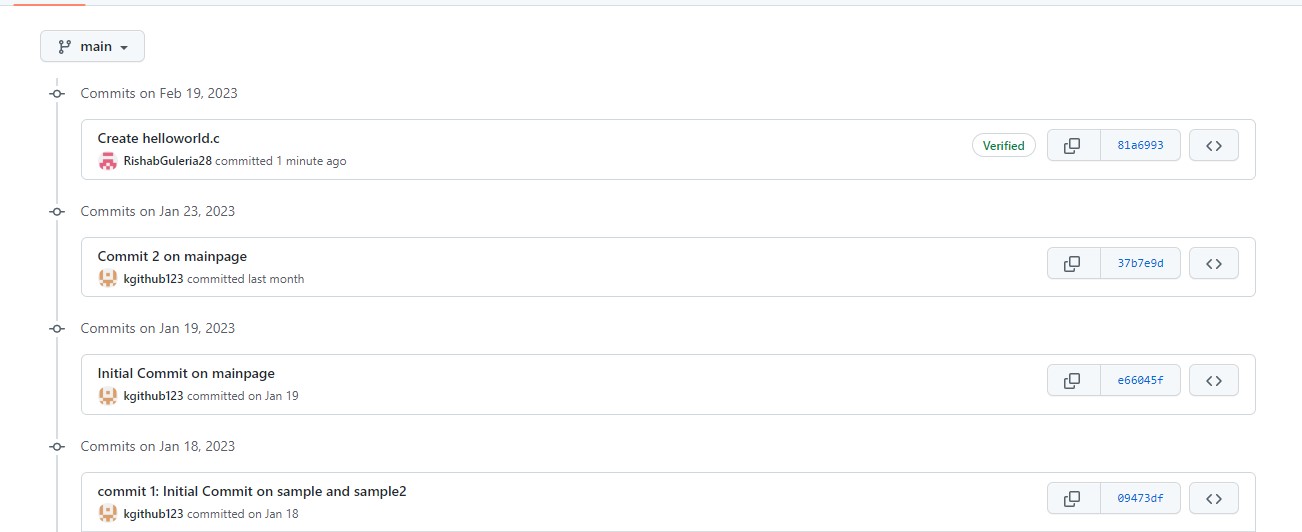








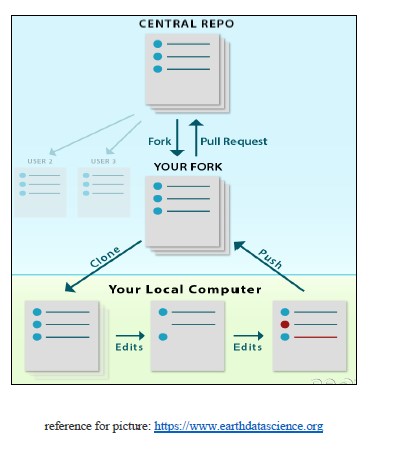




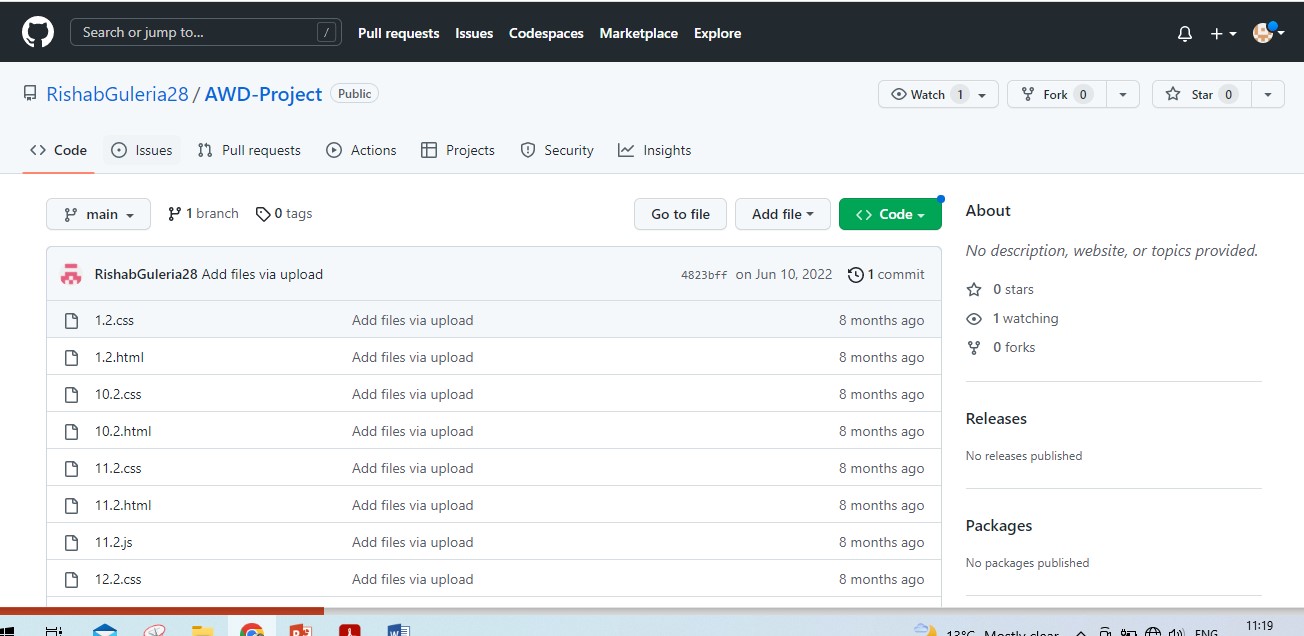
# EXPERIMENT NO. 07

## Aim: Fork and commit

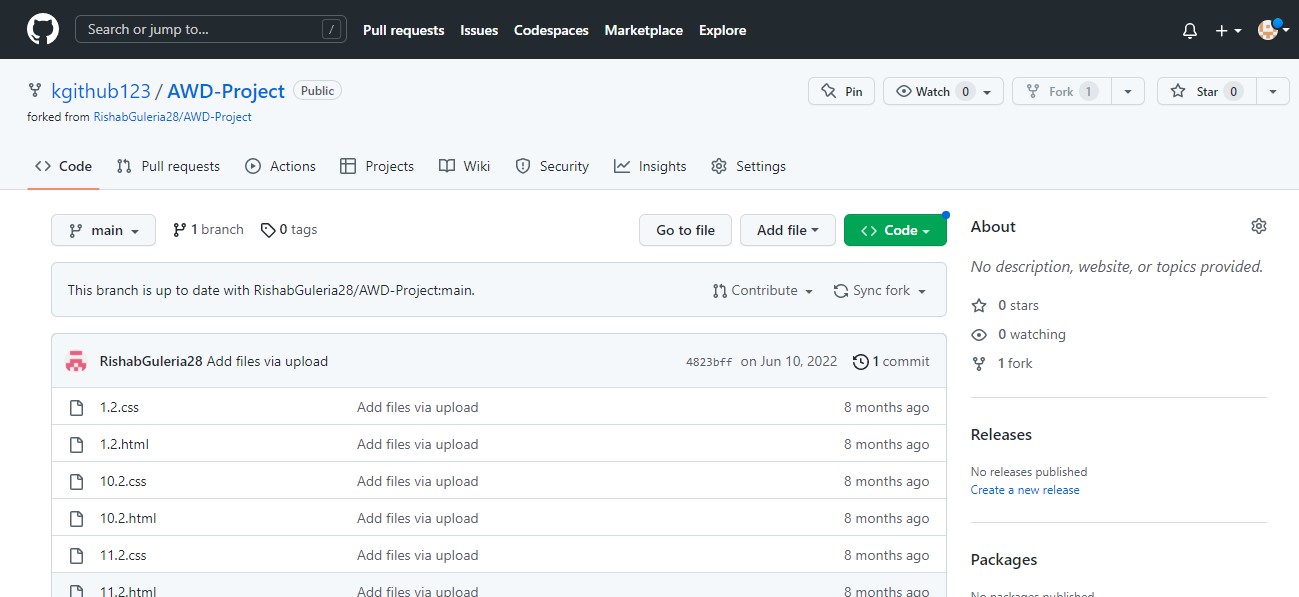
* A fork is a copy of a repository that you manage.
* It allows us to freely experiment with the data.
* After creating a fork, we can make any desired change like adding collaborators, rename files, generate GitHub pages but all these changes won’t be reflected in the original repository



Open a repo and fork

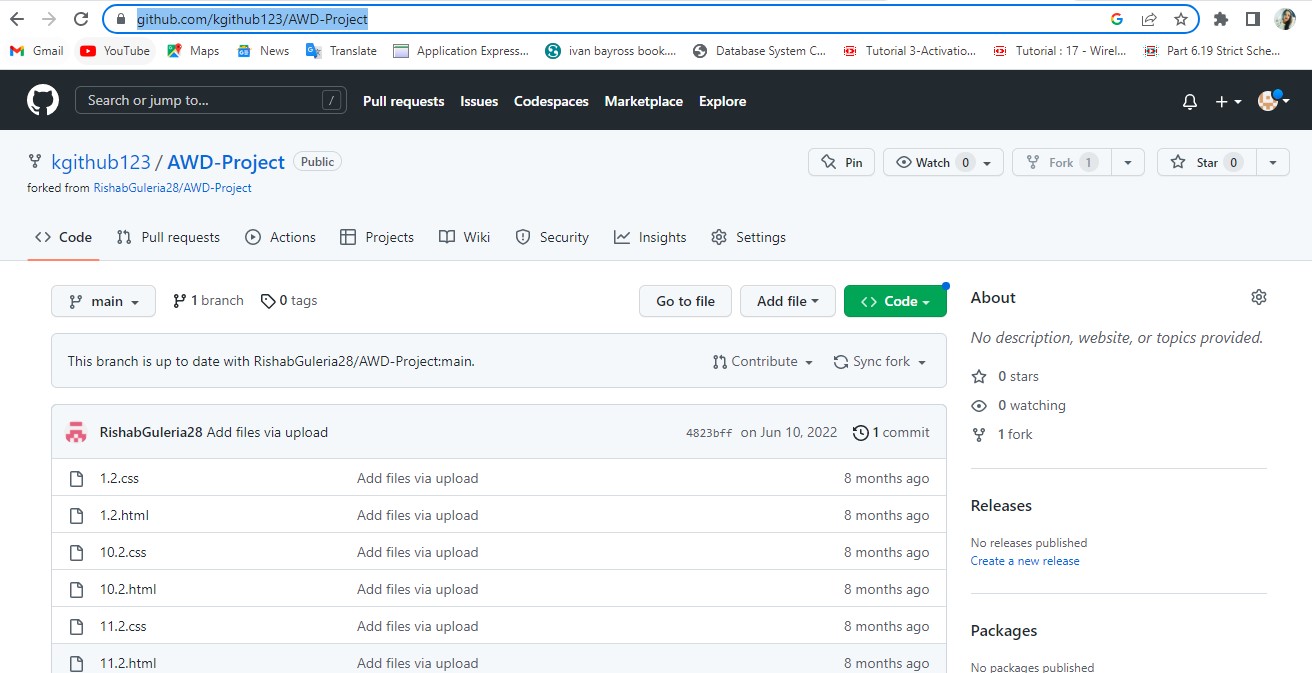


Forked repo in your GitHub account

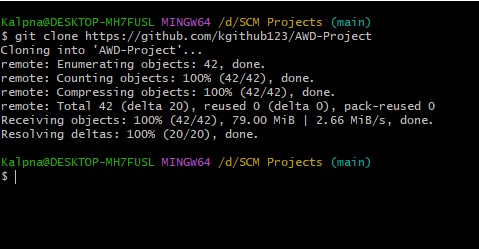
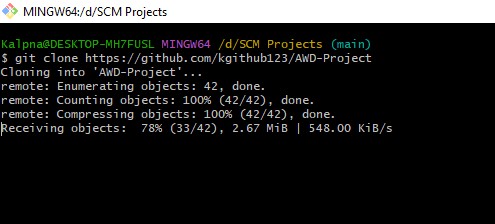


231099

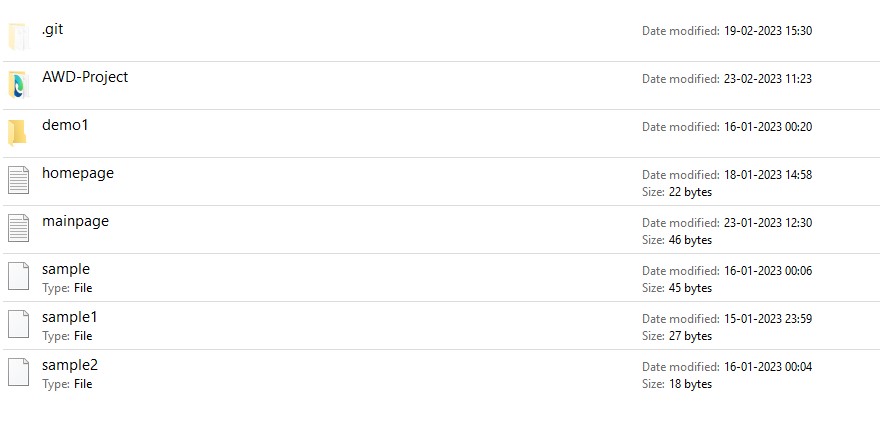
Copy the url of the repo



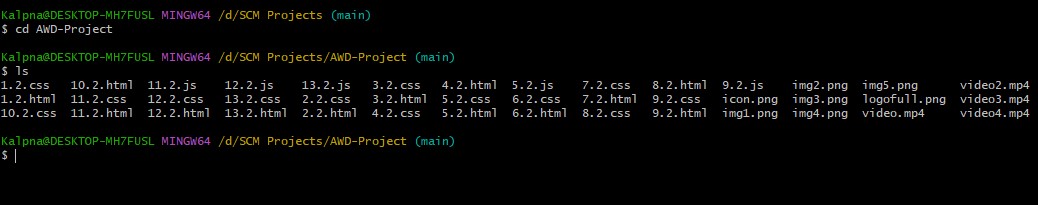
To clone a repo from the Github



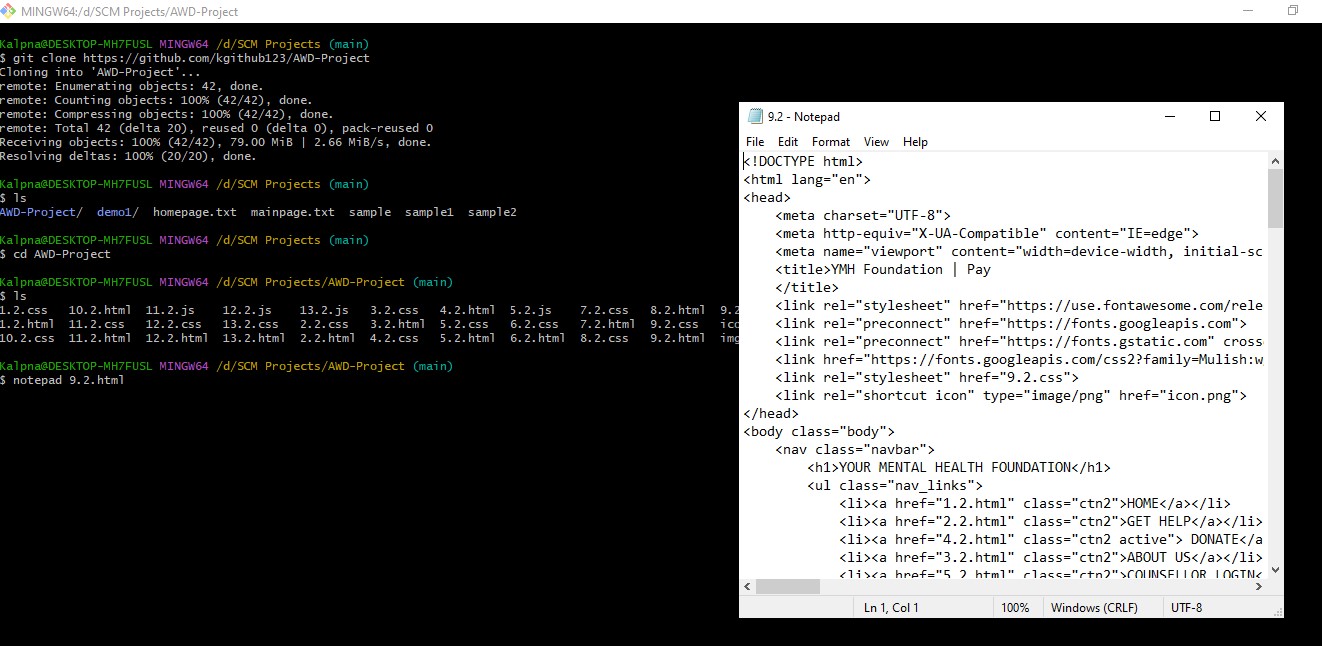
Forked repo in GUI



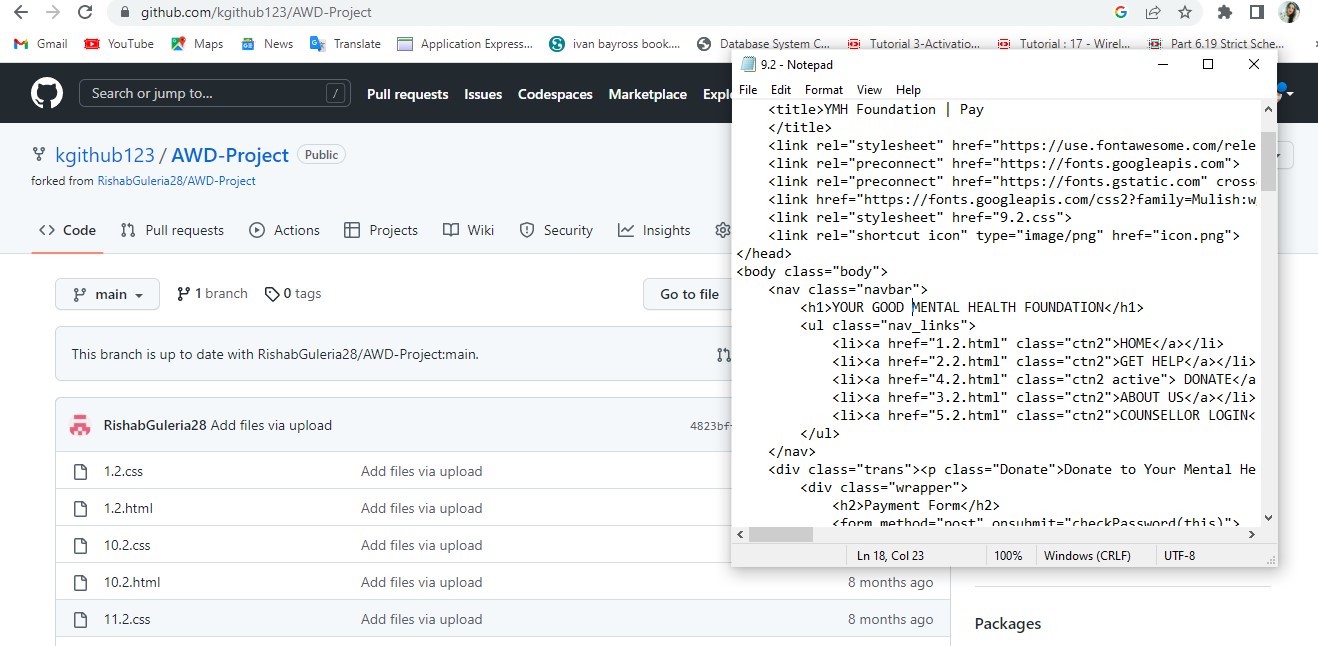
To view files in the forked repo in bash:



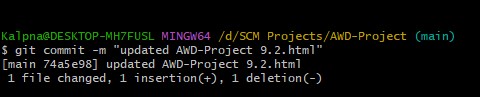
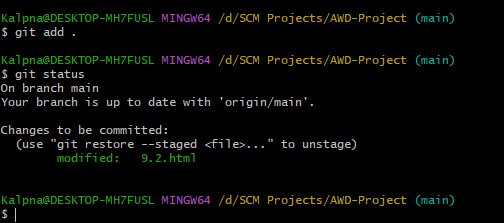
Open a file from the forked repo



Update a file :



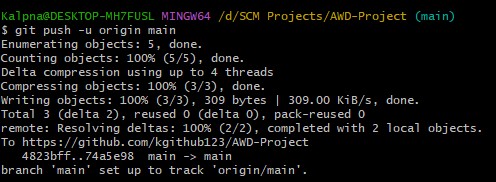
Commit the changes



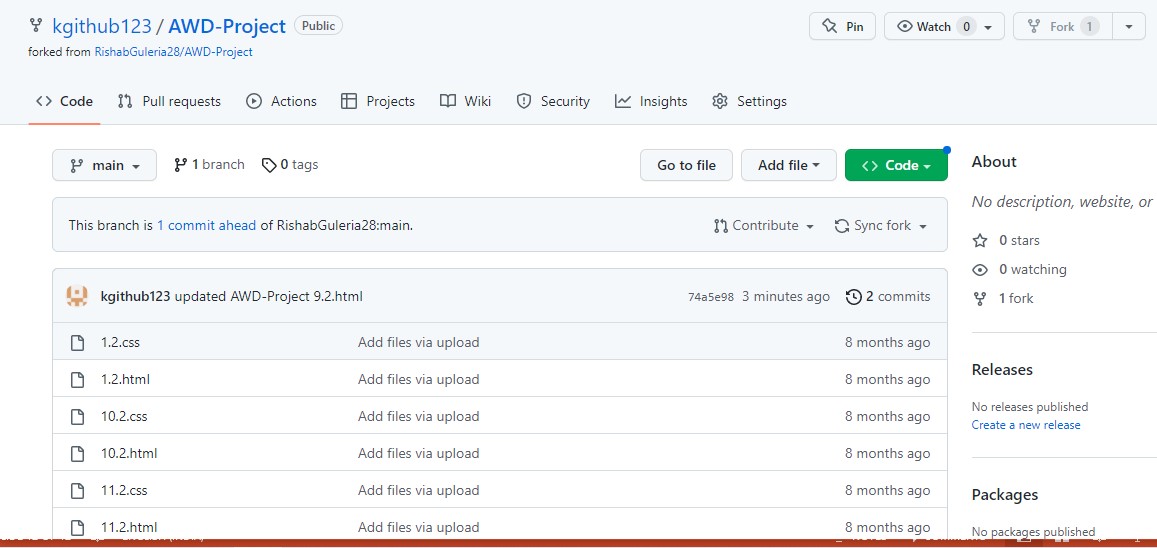
Check the origin created earlier to push the changes to Github



Push the changes to Github



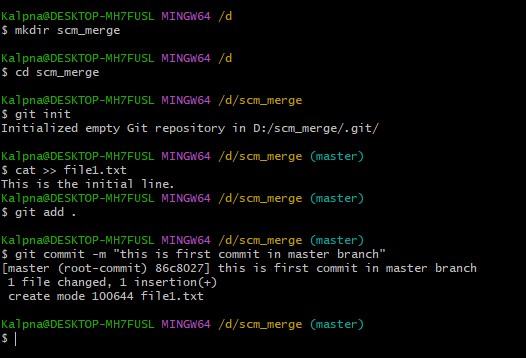
Check the commit on Github



# EXPERIMENT NO. 08

## Aim: Merge and resolve conflicts

Create a file in master branch and commit it.

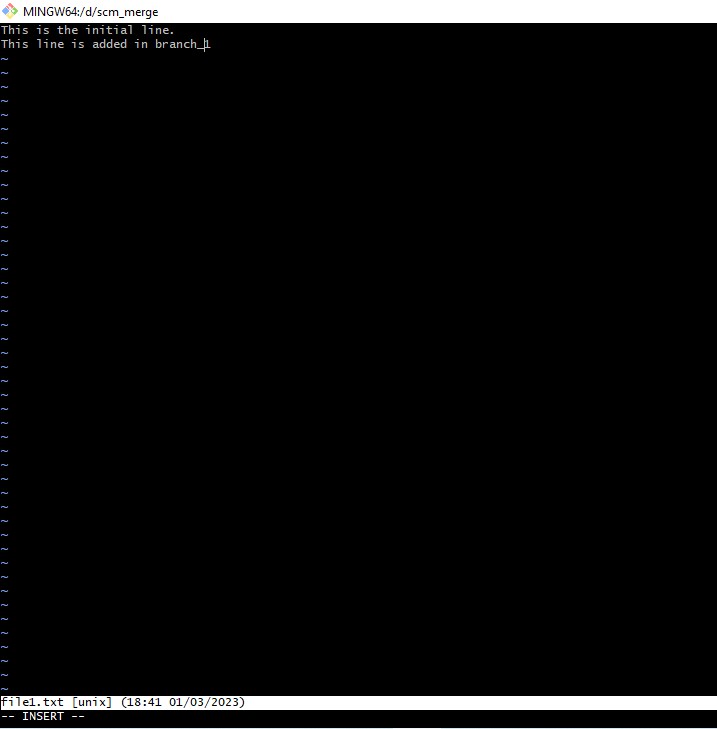


Create a branch and checkout in it

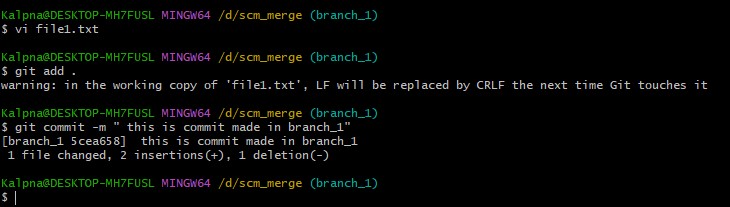


Open vi editor and add a line in the same file

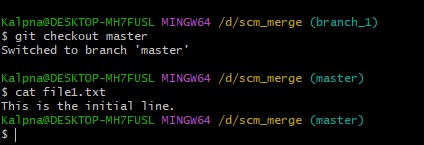
* Open vi editor by command vi filename
* I for insert mode
* Type text you want to add
* Esc:wq to save and exit vi editor.



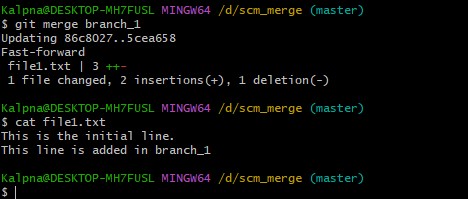
Commit the changes made in branch\_1



Checkout to master and see the contents of file1 , the changes made in branch 1 are not reflected in master

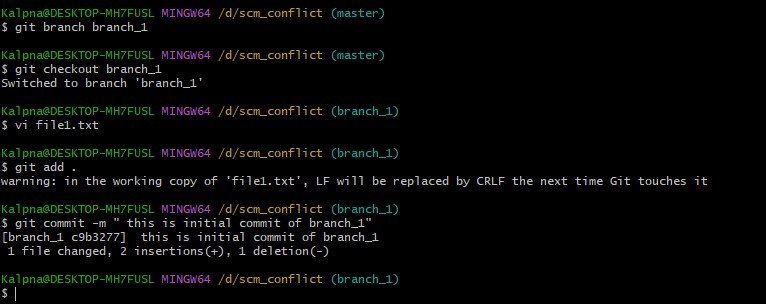
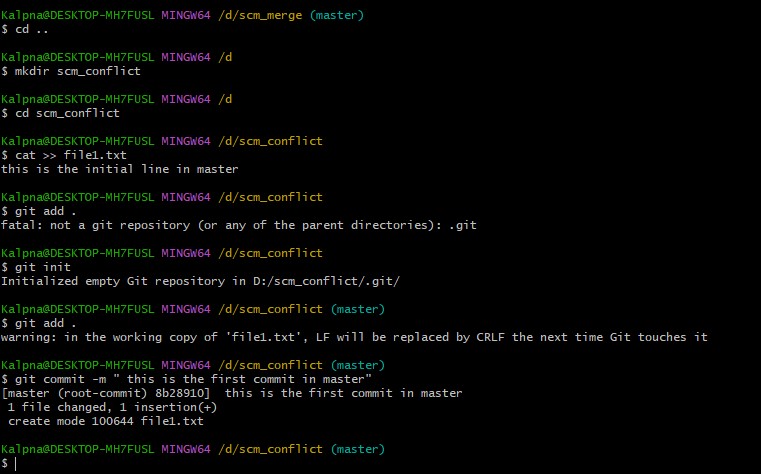


Merge the branch\_1 with master and view the changes being reflected in the file



231099

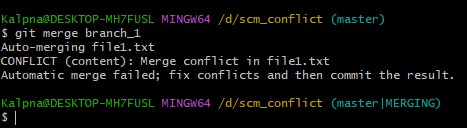
Resolve conflict:



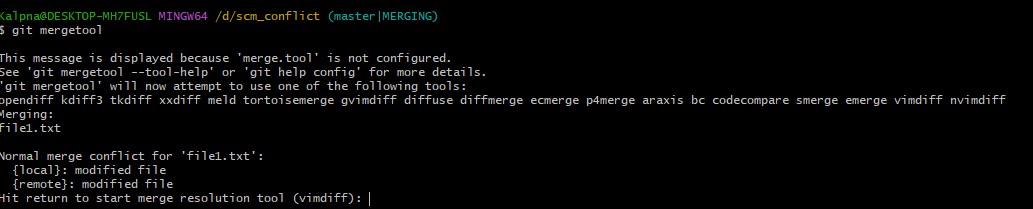
Return to master branch and change the file and commit



Merge branch\_1 with master and view the conflict occurred



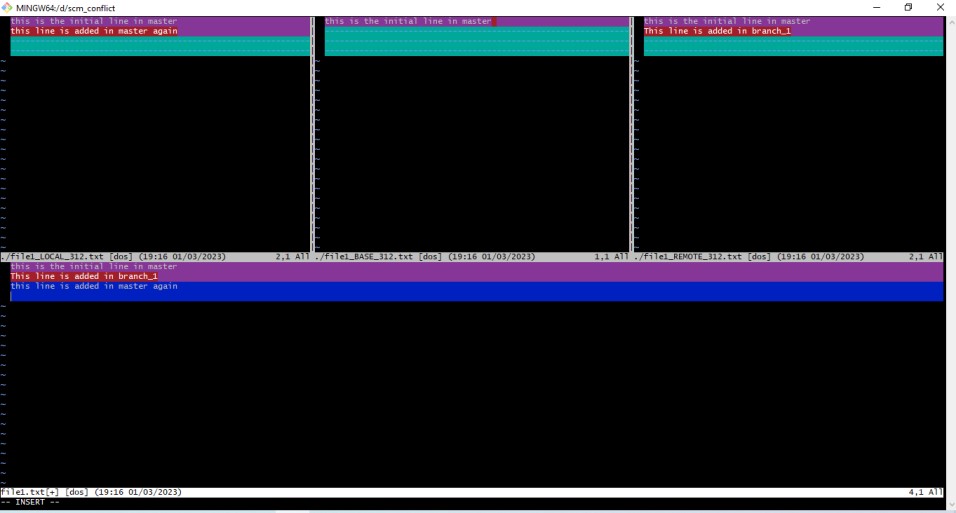
Use merge tool to resolve conflict:



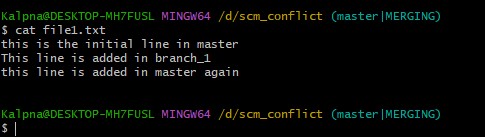
Mergetool opened:



Make the desired changes in the last window



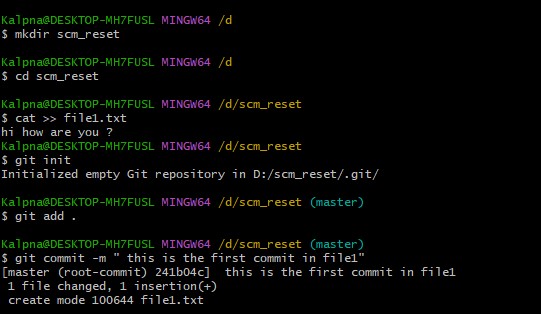
Exit mergetool esc:wq and view the final file in which the conflicts have been resolved:



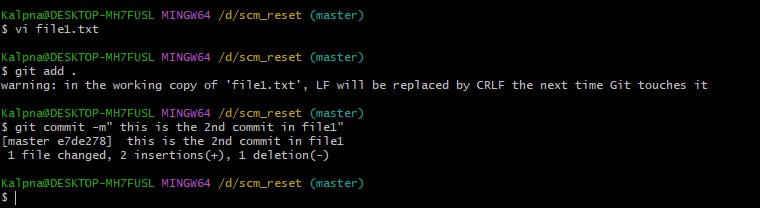
# EXPERIMENT NO. 09

## Aim: Git reset

Gitreset



Edit file1 and commit again



Soft reset done to undo most recent commit , it will bring the committed file back to staging area: you can view by using git status

