1) To set your Git username, type this in your terminal:

$git config --global user.name "Divyanshu84"

To confirm that you have set your Git username correctly, type this:

$git config --global user.name

You should have "Divyanshu84" as the output.

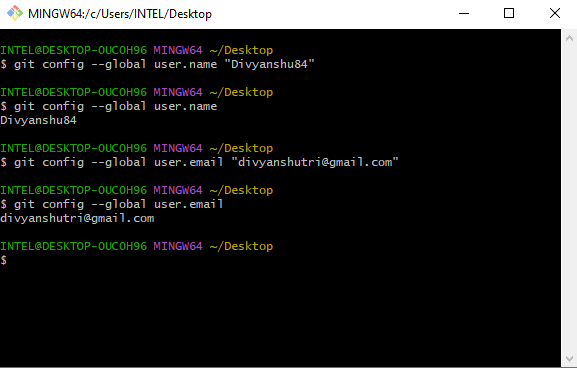
2) To set your Git email, type this in your terminal:

$git config --global user.email "divyanshutri@gmail.com"

To confirm that you have set your Git email correctly, type this:

$git config --global user.email

You should have "youremail@gmail.com" as the output.



3) Create a folder locally

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop

$ mkdir Demodt12

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop

$ cd De

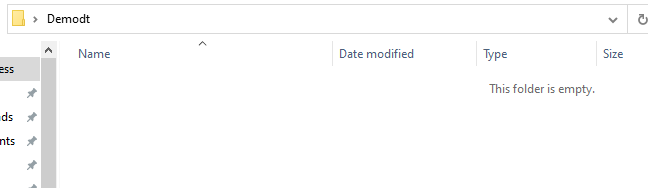
Demodt/

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop

$ cd Demodt

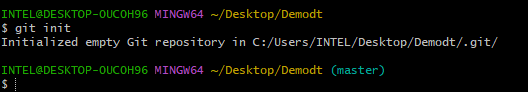
INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop/Demodt





**git init**

The git init command creates a new Git repository.

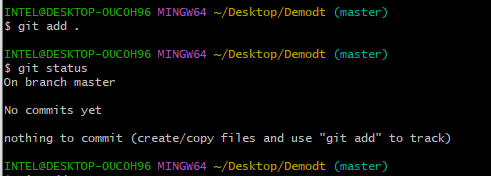


**Git add**

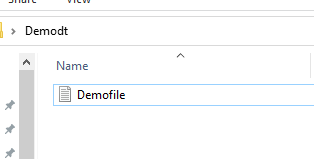
Moves changes from the working directory to the staging area.

**Git Status**

The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven’t, and which files aren’t being tracked by Git. Status output does not show you any information regarding the committed project history. For this, you need to use git log.



Go to your folder and add one file



INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop/Demodt (master)

$ git add .

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop/Demodt (master)

$ git status

On branch master

No commits yet

Changes to be committed:

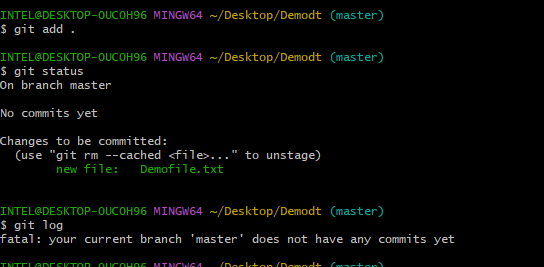
(use "git rm --cached <file>..." to unstage)

new file: Demofile.txt

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop/Demodt (master)

$ git log

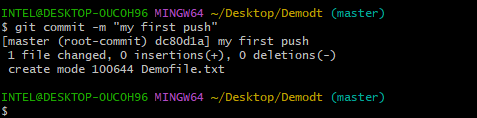
fatal: your current branch 'master' does not have any commits yet



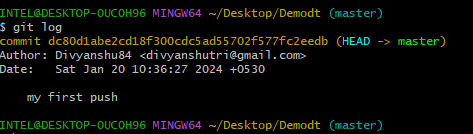
**Git Commit**

The git commit command captures a snapshot of the project's currently staged changes. Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to.

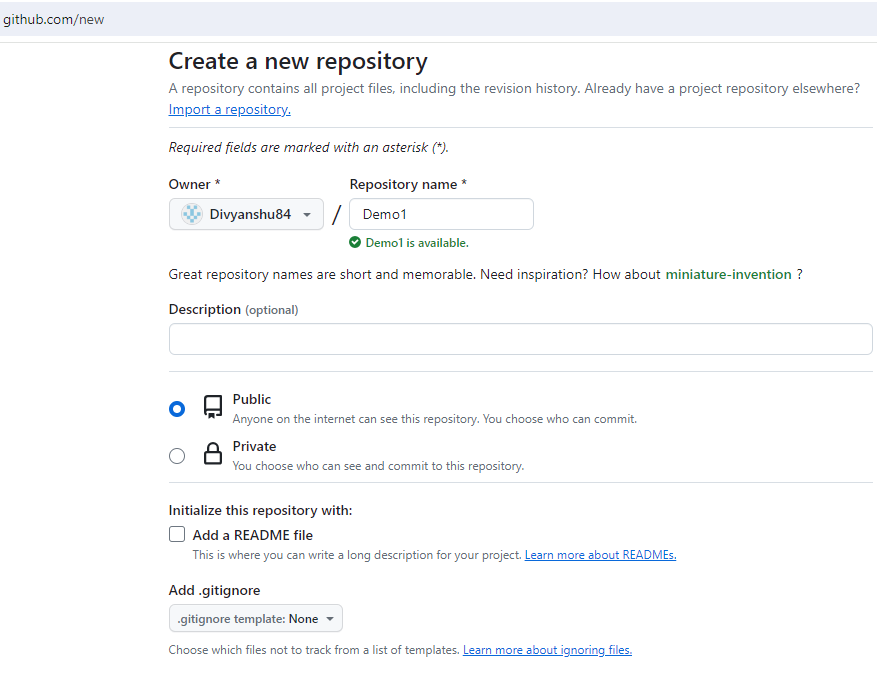
Prior to the execution of git commit, the git add command is used to promote or 'stage' changes to the project that will be stored in a commit. These two commands git commit and git add are two of the most frequently used.

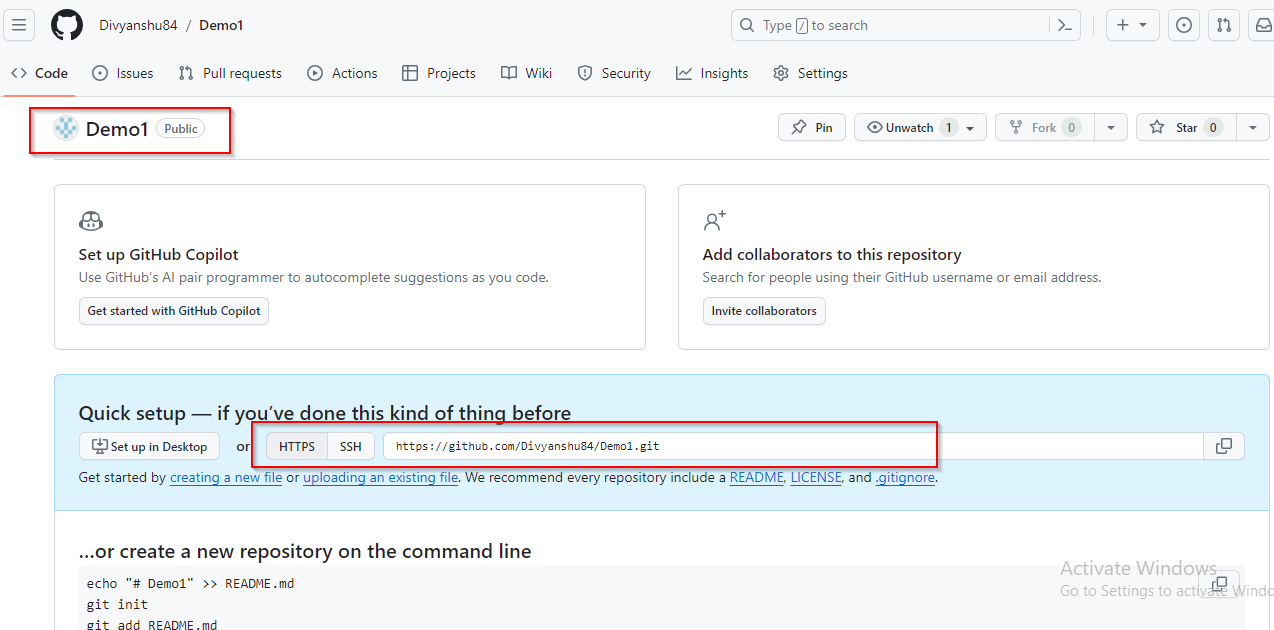


Git Log



Login in Github and create a new repository





Git Remote

The git remote command lets you create, view, and delete connections to other repositories. Remote connections are more like bookmarks rather than direct links into other repositories. Instead of providing real-time access to another repository.

**git remote add origin** finally connects the local folder to the repository on GitHub. It is followed by the repository's link.

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop/Demodt (master)

$ git remote add origin https://github.com/Divyanshu84/Demo1.git

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop/Demodt (master)

$ git branch -m main

INTEL@DESKTOP-0UC0H96 MINGW64 ~/Desktop/Demodt (main)

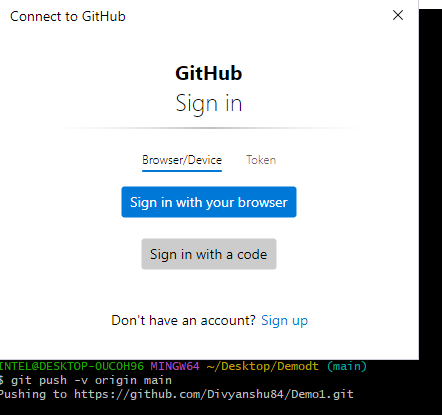
$

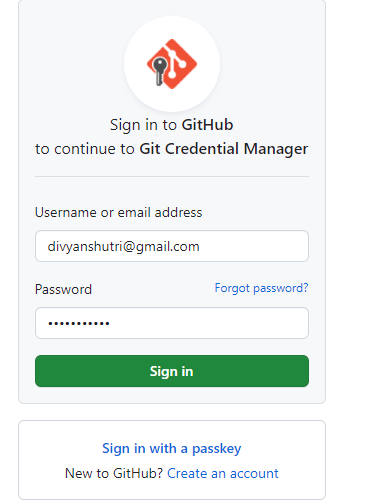
**Git push**

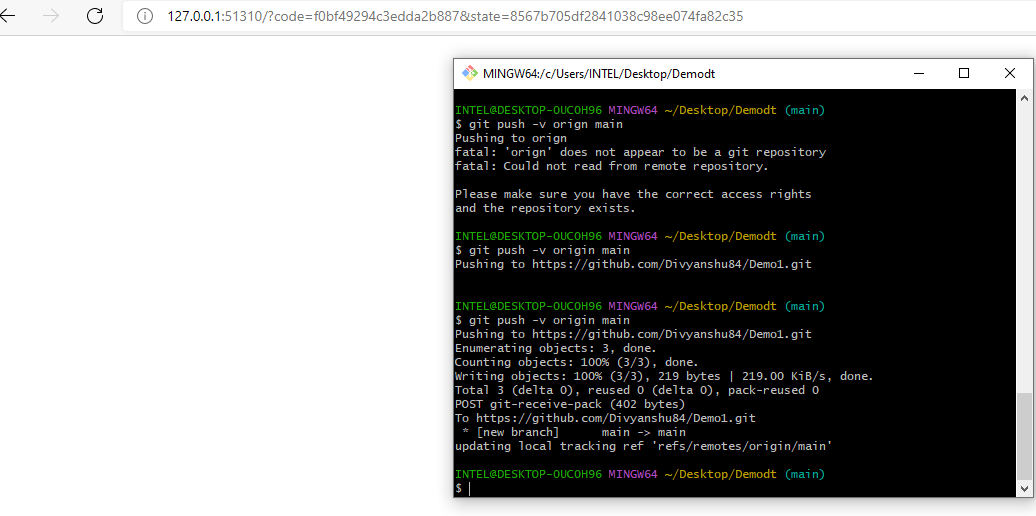
git push -v origin main

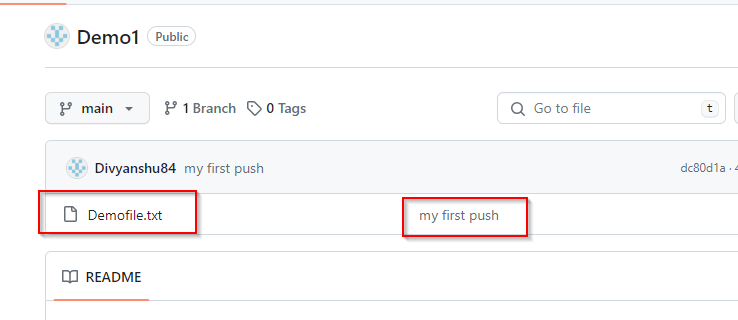
it will ask you for your sing in details for login in git

Click on “Sign with Browser”



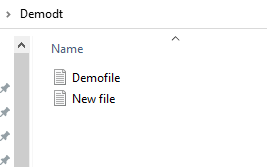




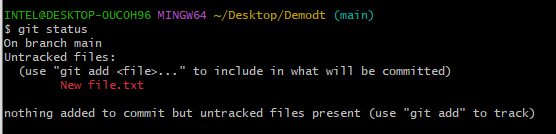


1. Create new file in folder
2. Add the file in git
3. Commit the file
4. Push the file

1)



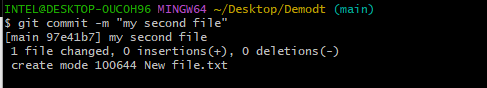
2)



3)

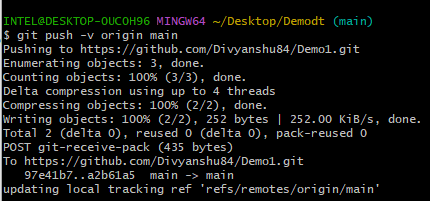


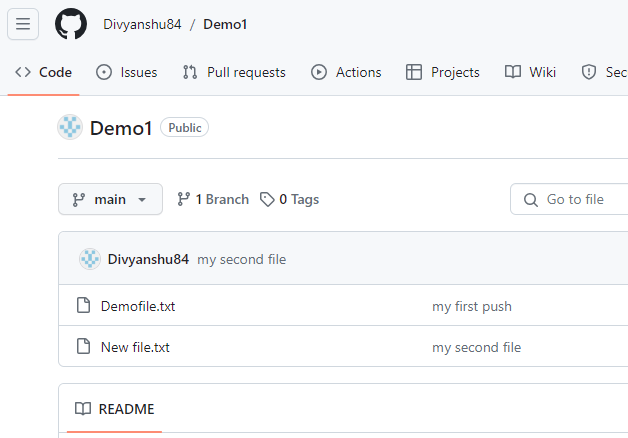
4)



5)

$ git push origin main





Logout from git

git config --unset-all --global user.name

git config --unset-all --global user.email

git config --unset-all --global user.name "nishantchoutele"

git config --unset-all --global user.email "nishantchoutele@gmail.com"