**GIT PROJECT FILE:**

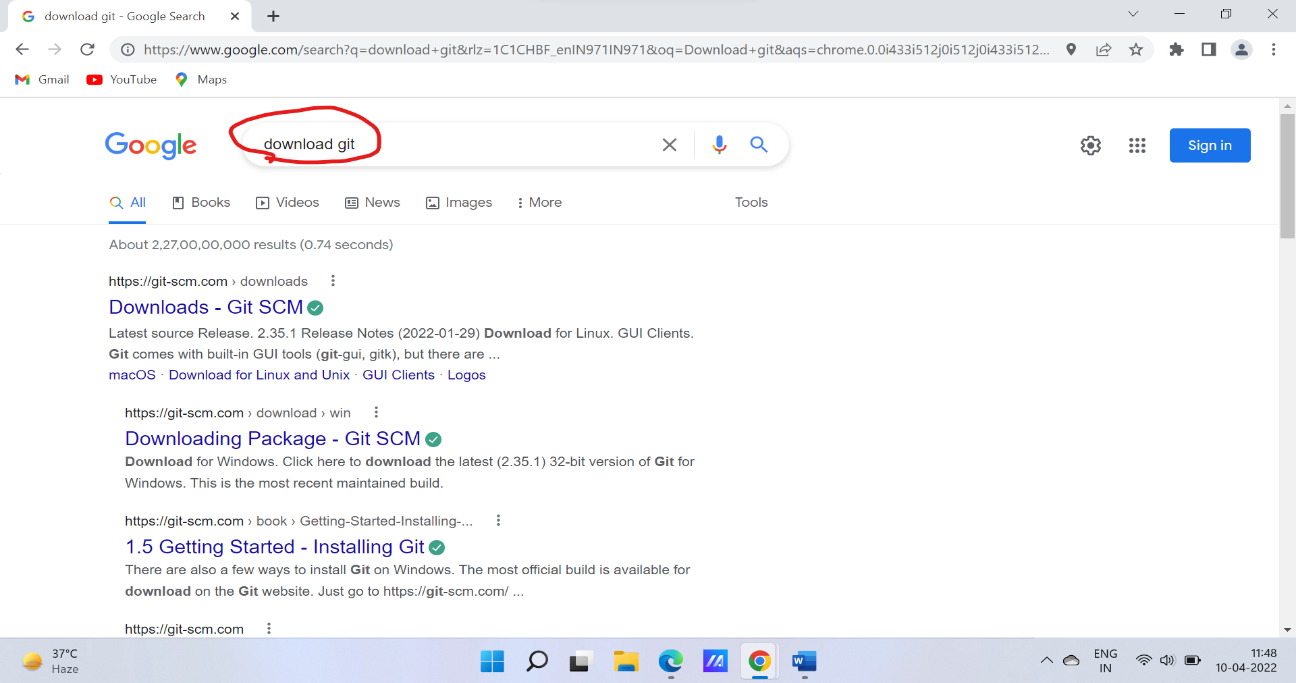
**By Arsh kumar(2110990268)**

**Pratical-1**

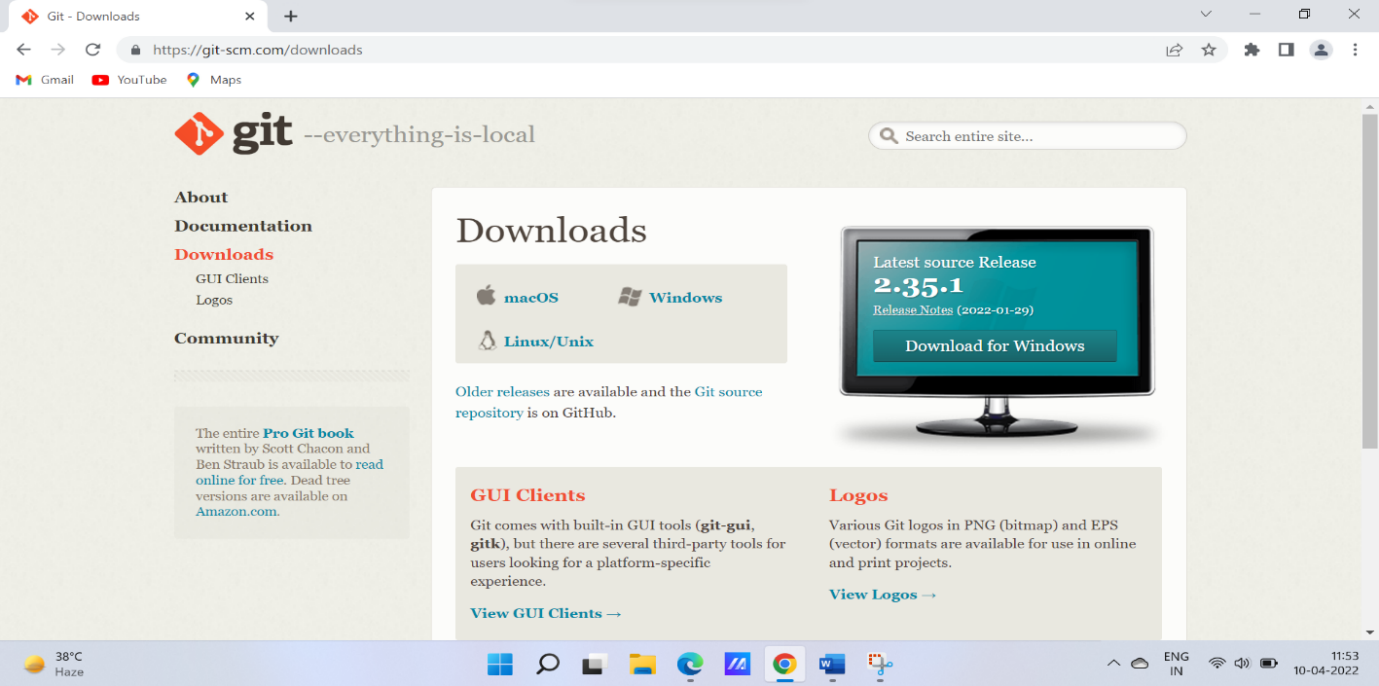
**Aim:** Setting up the git client.

**Procedure:**

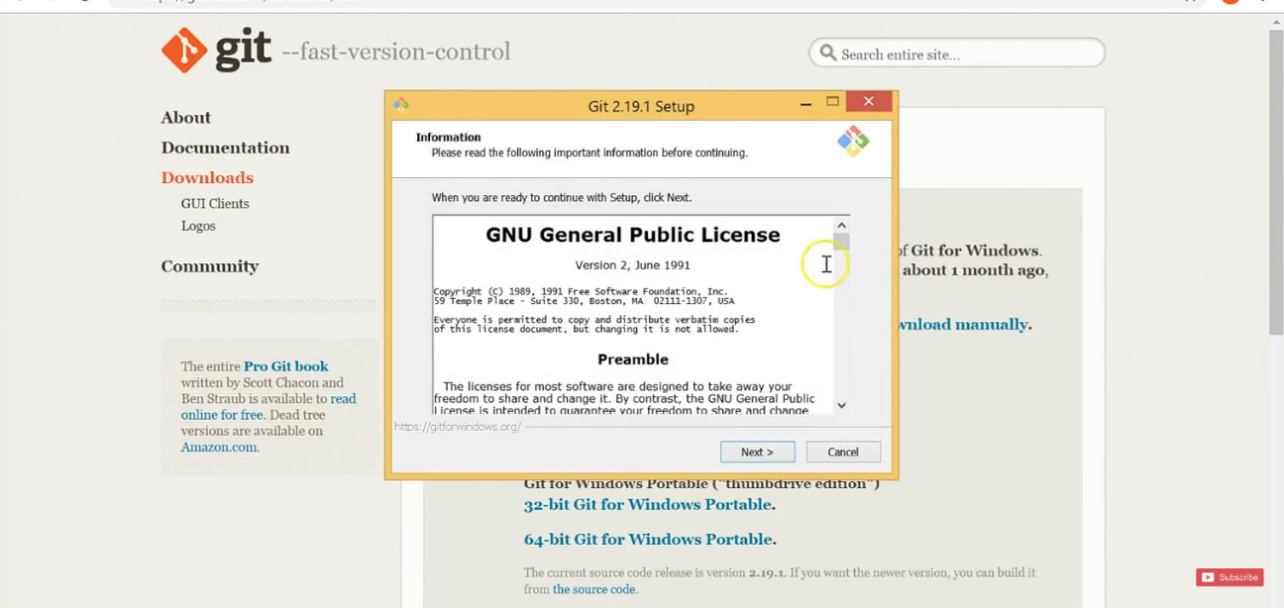
* Search download git on chrome or any search engine available to you.



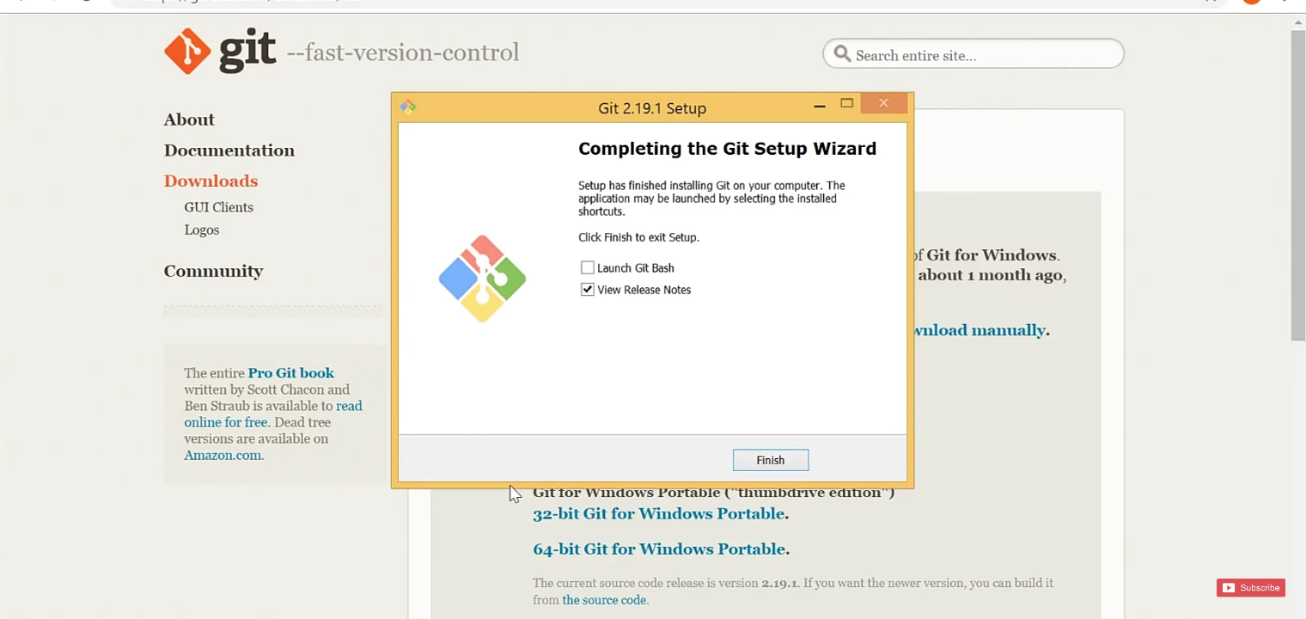
* Click on https://git-scm.com › downloads
* Chose a version you have to download and the download will start

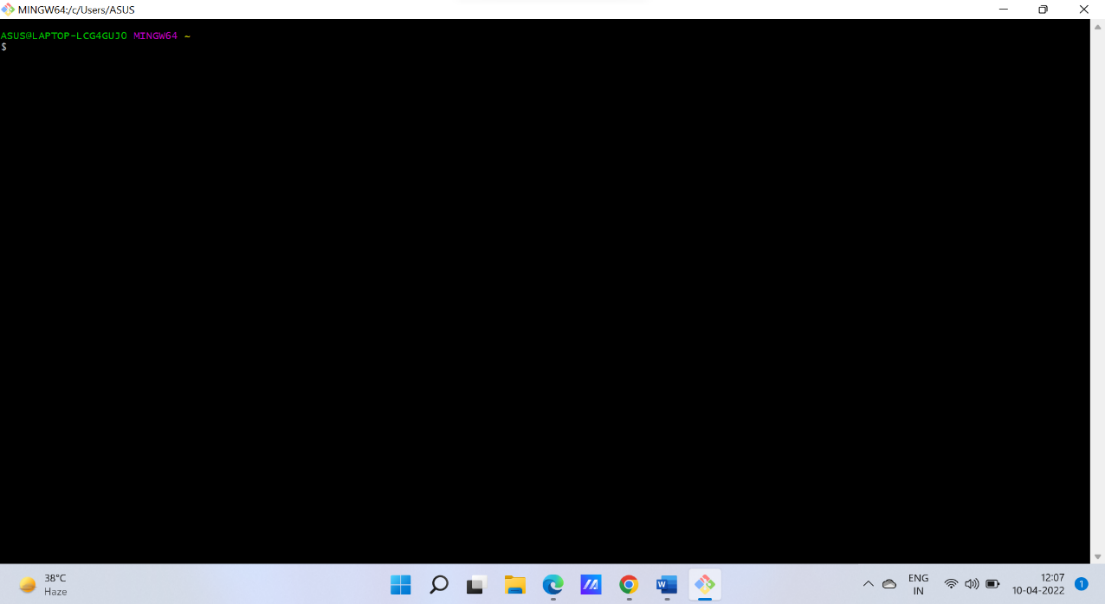


* After downloading a pop-up will open and click next and keep the default setting as it is but if you want to change you can .



* Click finish and launch Git bash





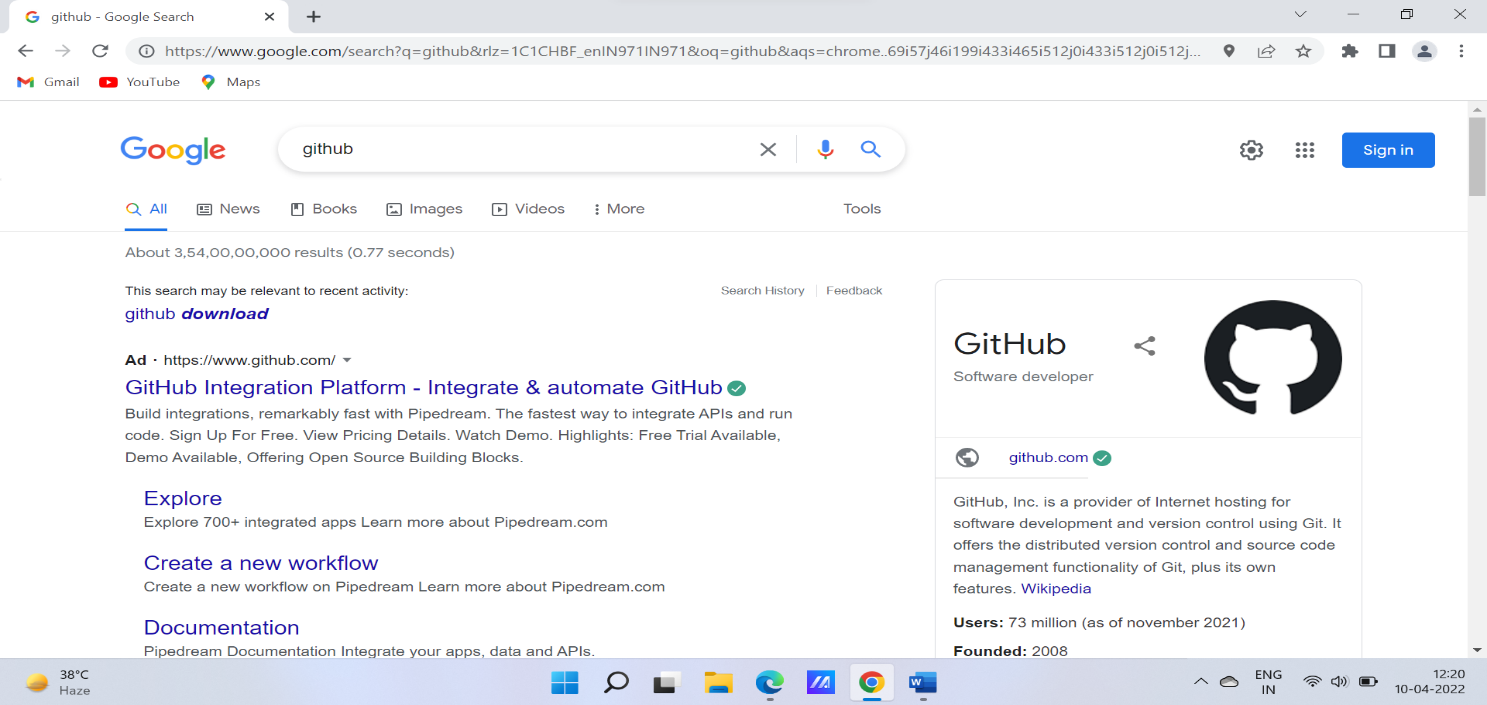
**Practical-2**

# Aim

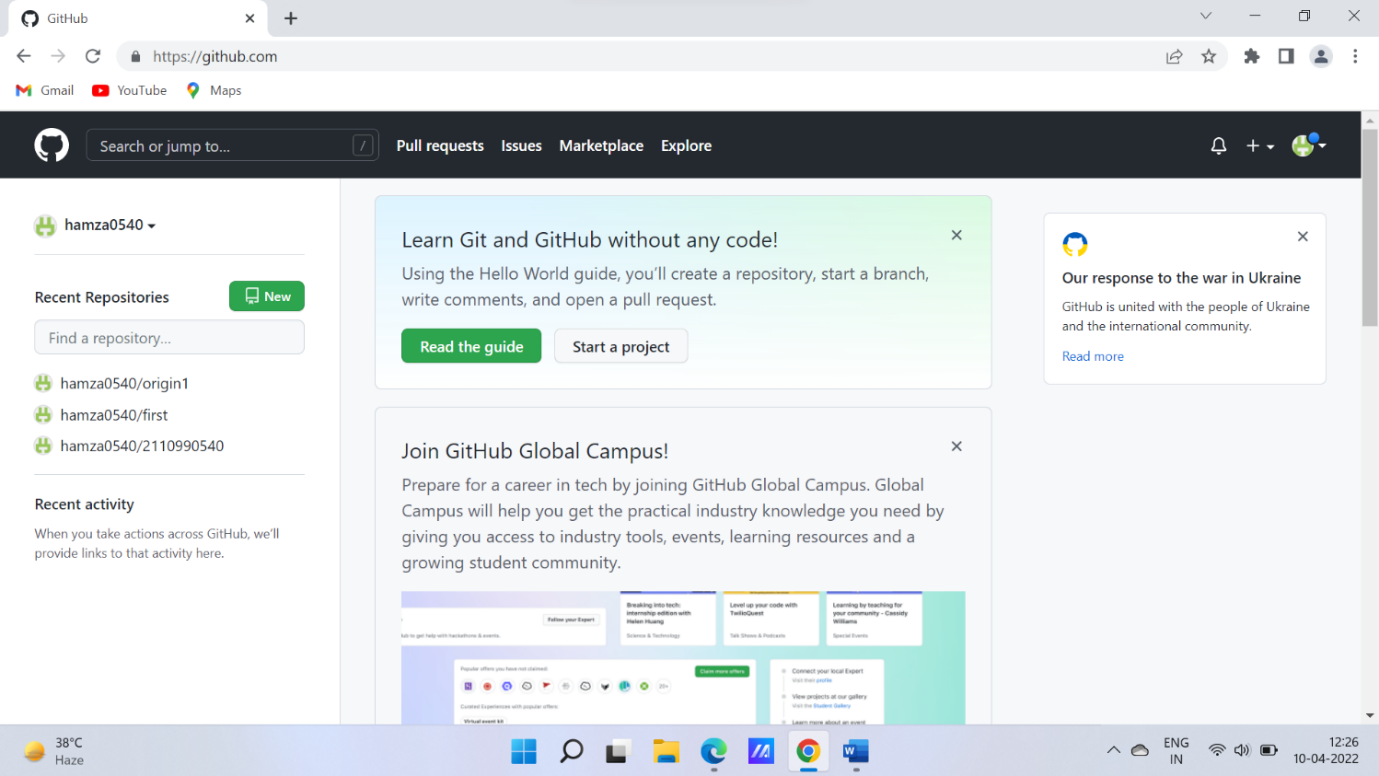
Setting up GitHub Account

The steps for setting up a github account are:

* Search Github on any search engine available to you and click on <https://github.com>.



* Click on sign-up and enter you email, password and username.
* Enter the code that has been mailed to you.
* Select the members and your designation.
* Select your plan
* And your Github is set up



**GIT Basic commands:**

1. **ls: All the files printed on the directory where we working.**
2. **clear: Used to clear the screen.**
3. **pwd: stands for(present working directory)- It tells us in directory we are working.**
4. **cd : stands for(change directory)-used to change directory.**
5. **Cd .. : used to get out from directory/file.**
6. **mkdir : It is used for making a new directory and you have to named it.**

**PRACTICAL-3**

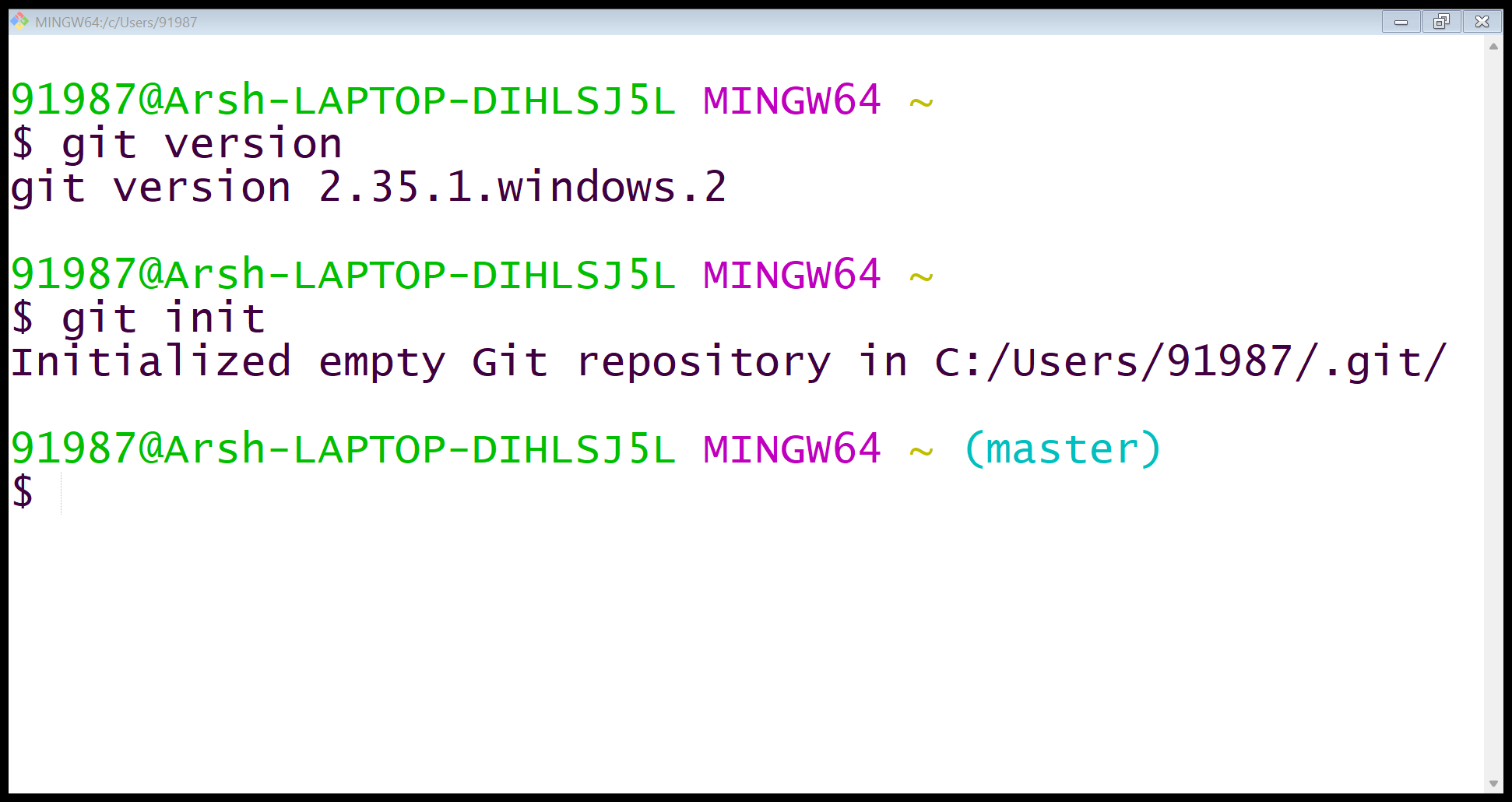
* git version:

It is a GIT command used to check thew version of GIT having in your device.



* git init:

It is a GIT command used to create a GIT repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository. By using this command we will see a .git hidden folder in the folder we are working in.



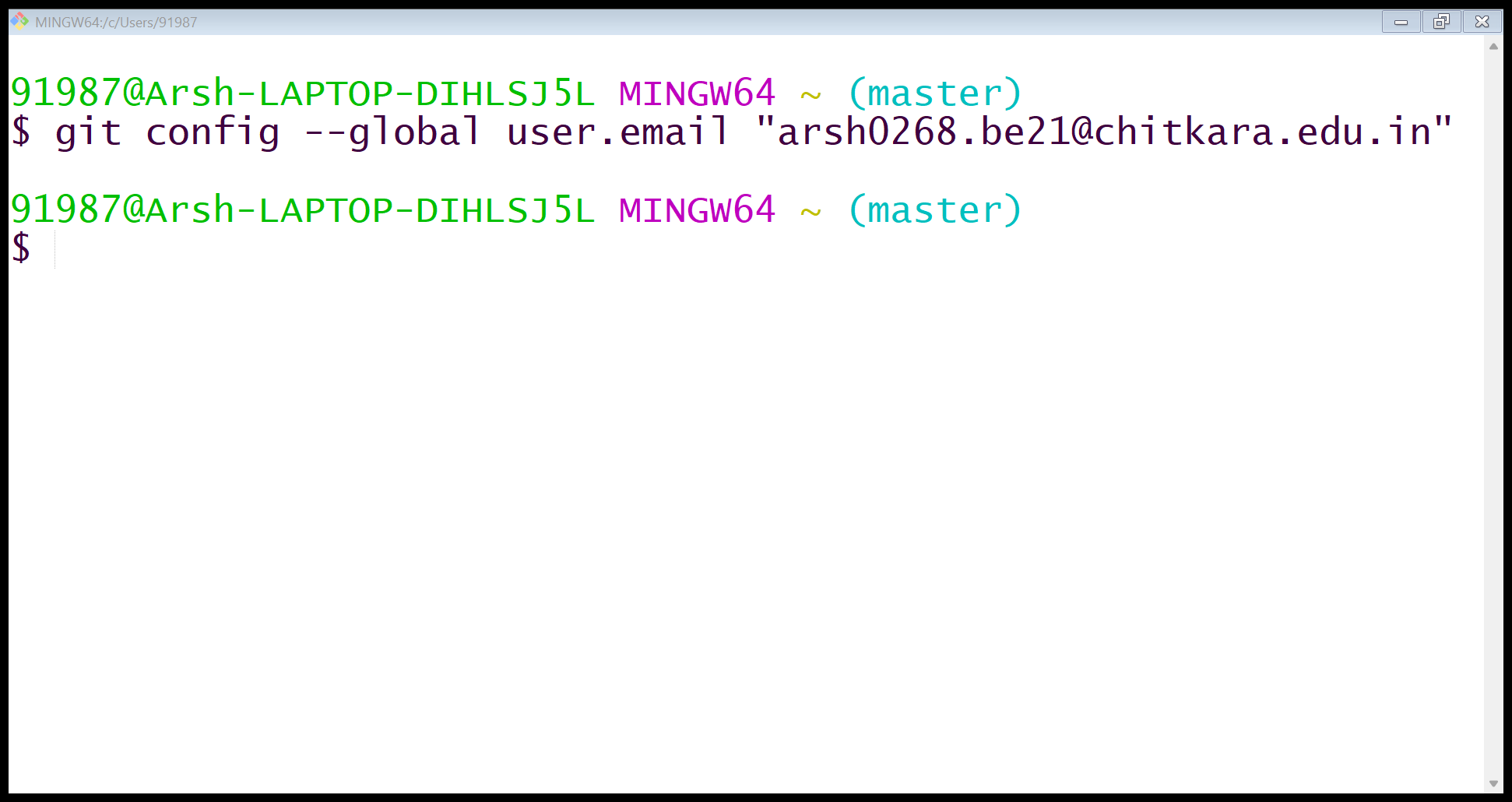
* git config --global user.name “ ” :

It is a GIT command used to config your global username for the GIT. The same name is used in GITHUB account.



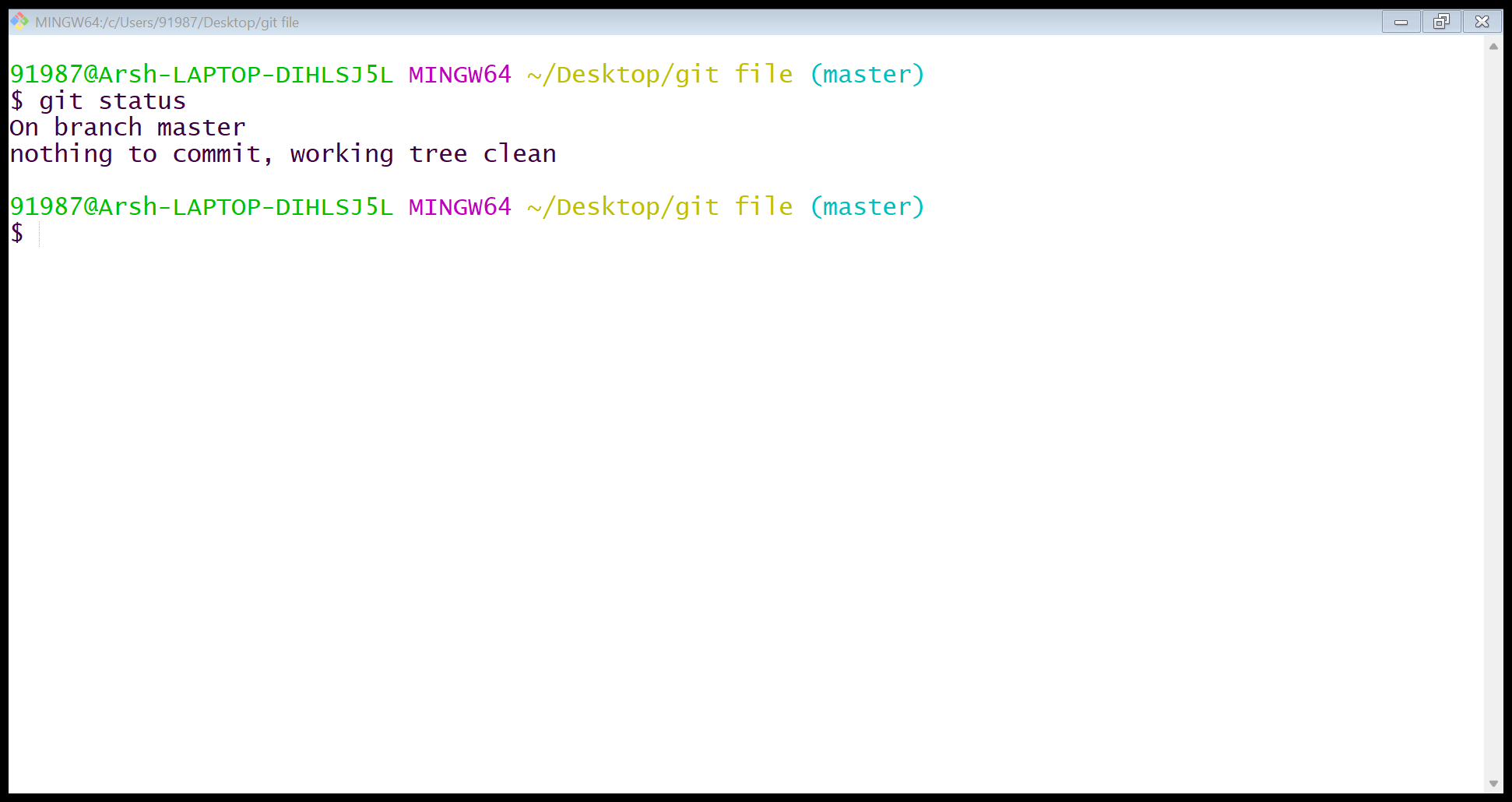
* git config --global user.email “ ” :

It is a GIT command used to config your global email for the GIT. The same email is used in GITHUB account.

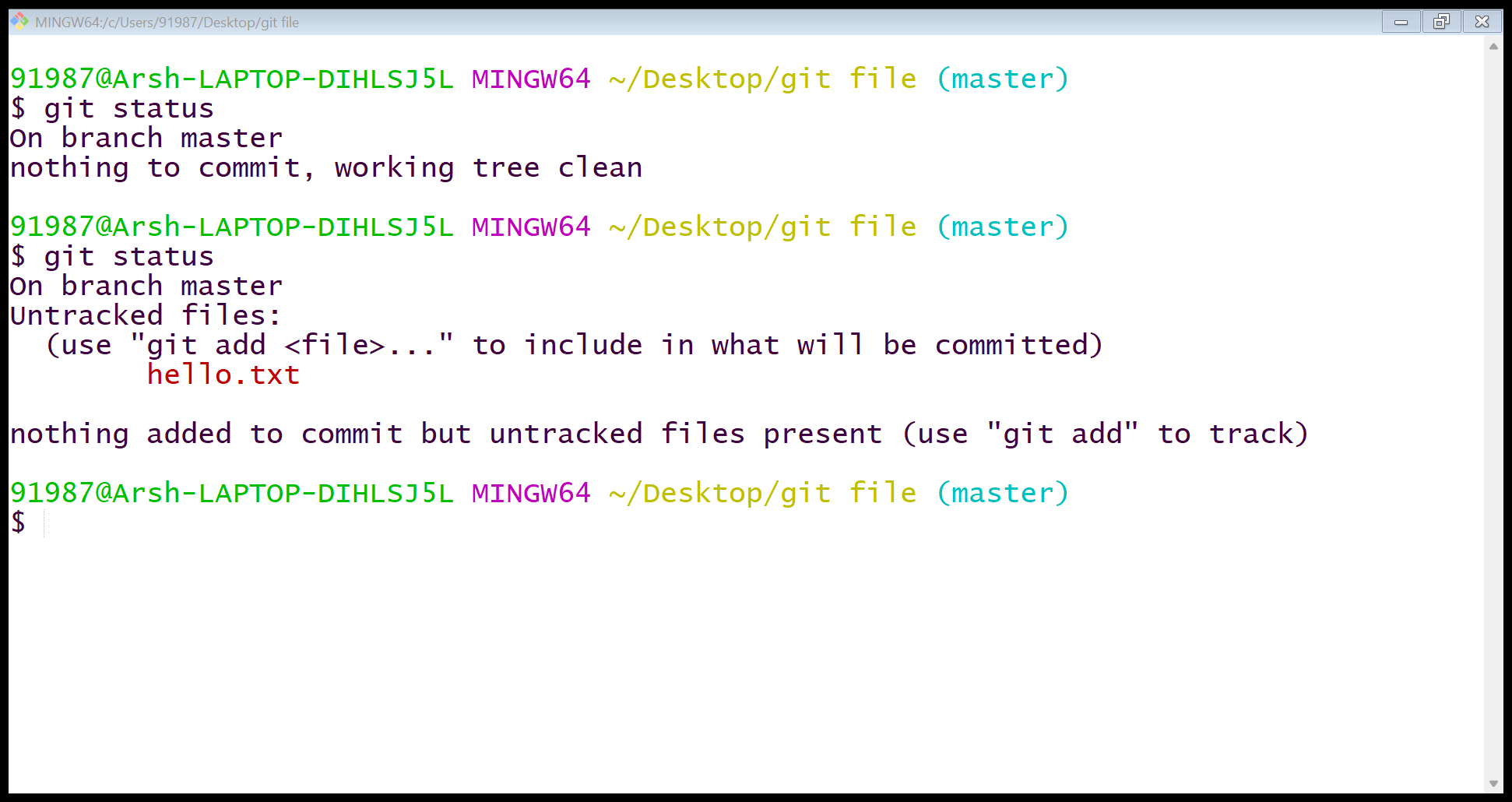


* git status:

It is a GIT command used to check the status of the GIT repository.



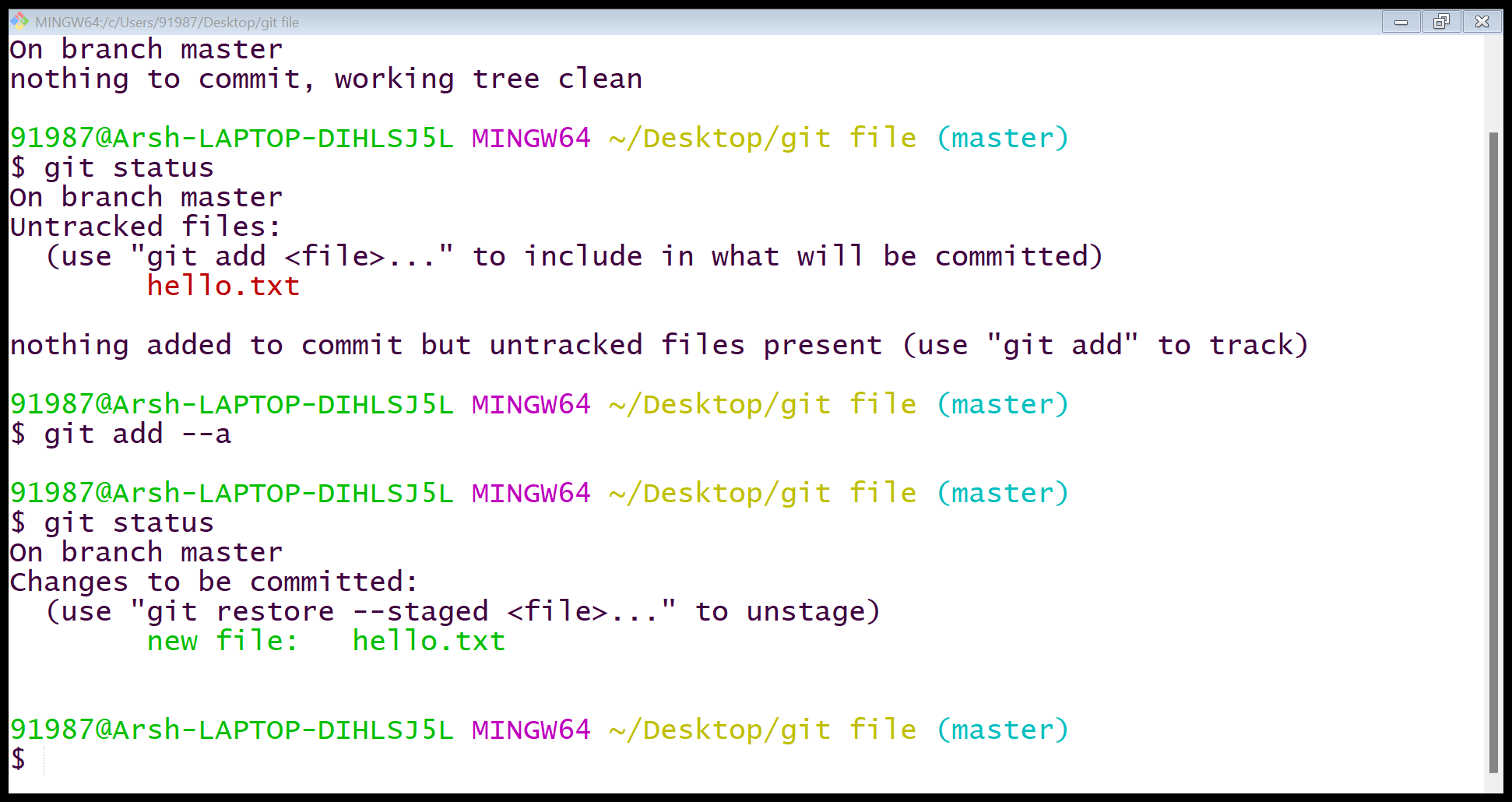
* Git status when the files are untracked:



The Red highlighted text indicates that the files are untracked in the repository but are added in the folder.

* git add --a :

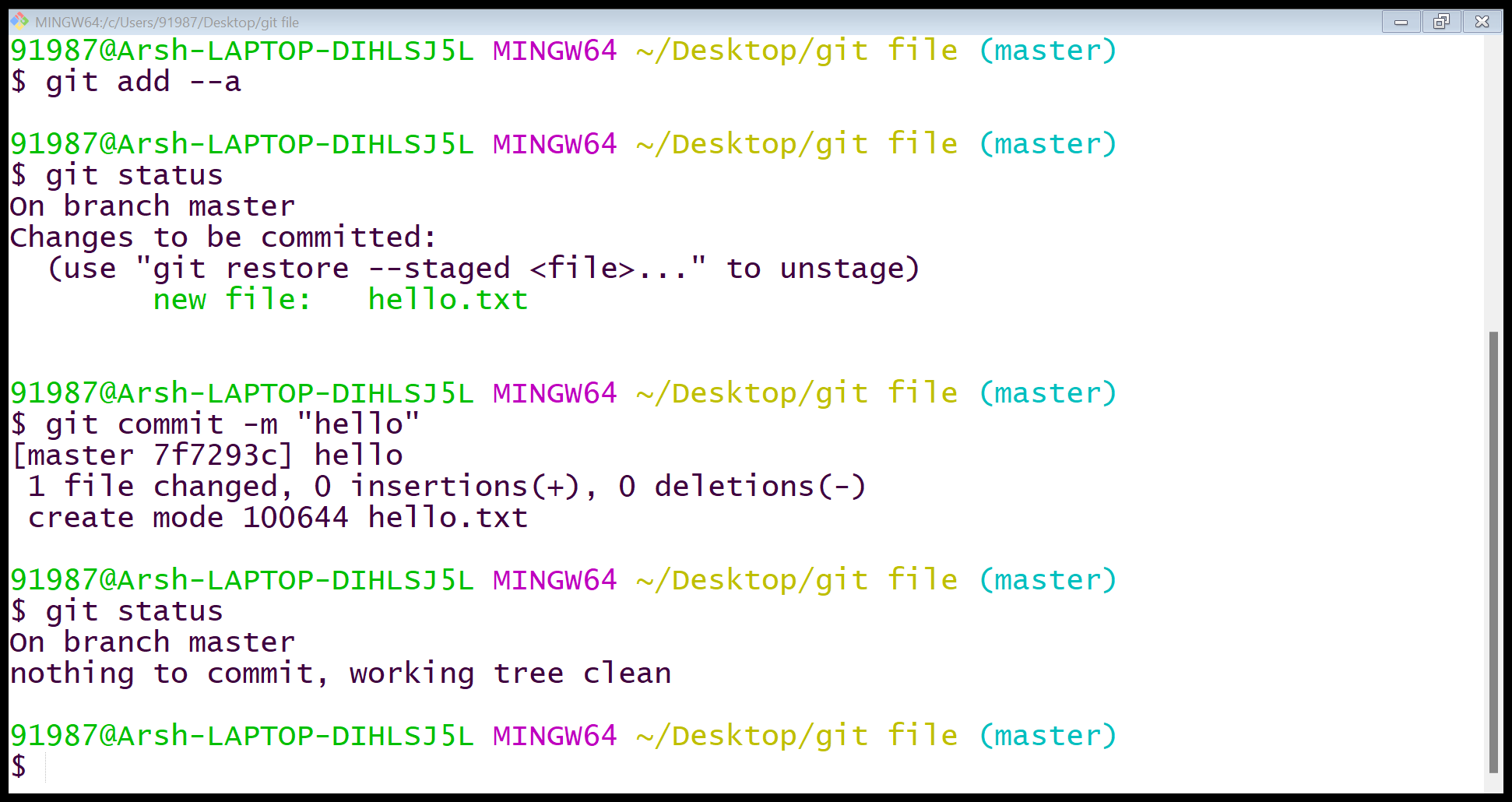
It is a GIT command used to add files to the present repository and keep them on a track.



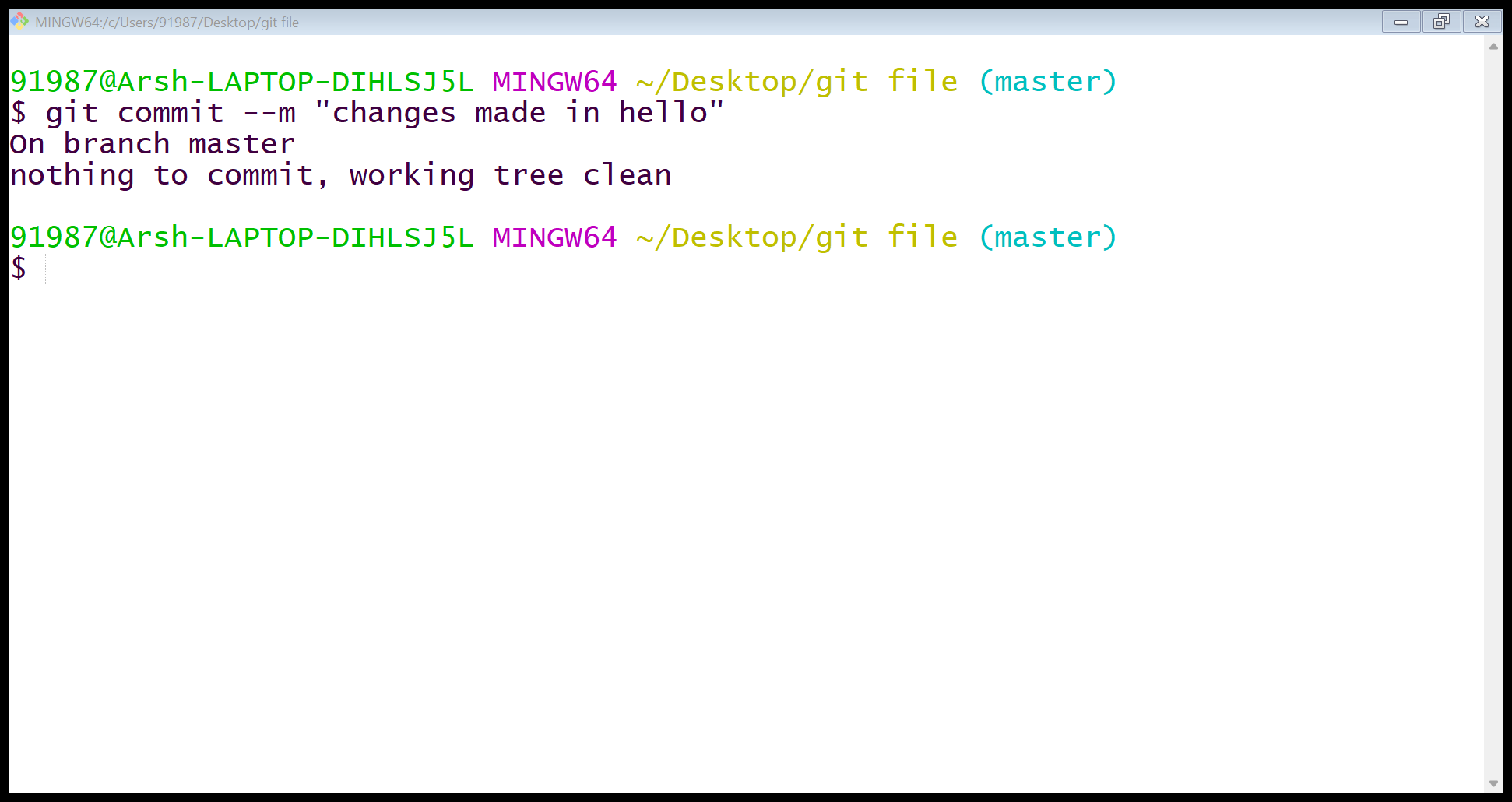
After the command works, the files are tracked in the repository and are shown in green highlighted text.

* git commit –m “ ” :

It is a GIT command to deliver a short account of the information of the changes being committed in the file.



The statement “nothing to commit, working tree clean” tells that all the changes made in GIT repository are committed.



* git branch:

It is a GIT command used to check in which the GIT branch we are staging changes. The MASTER is the main GIT branch.



* git branch name:

It is a GIT command used to create new branch. By this command we create a new branch but we can’t work in the same till the time we don’t use the switch command.



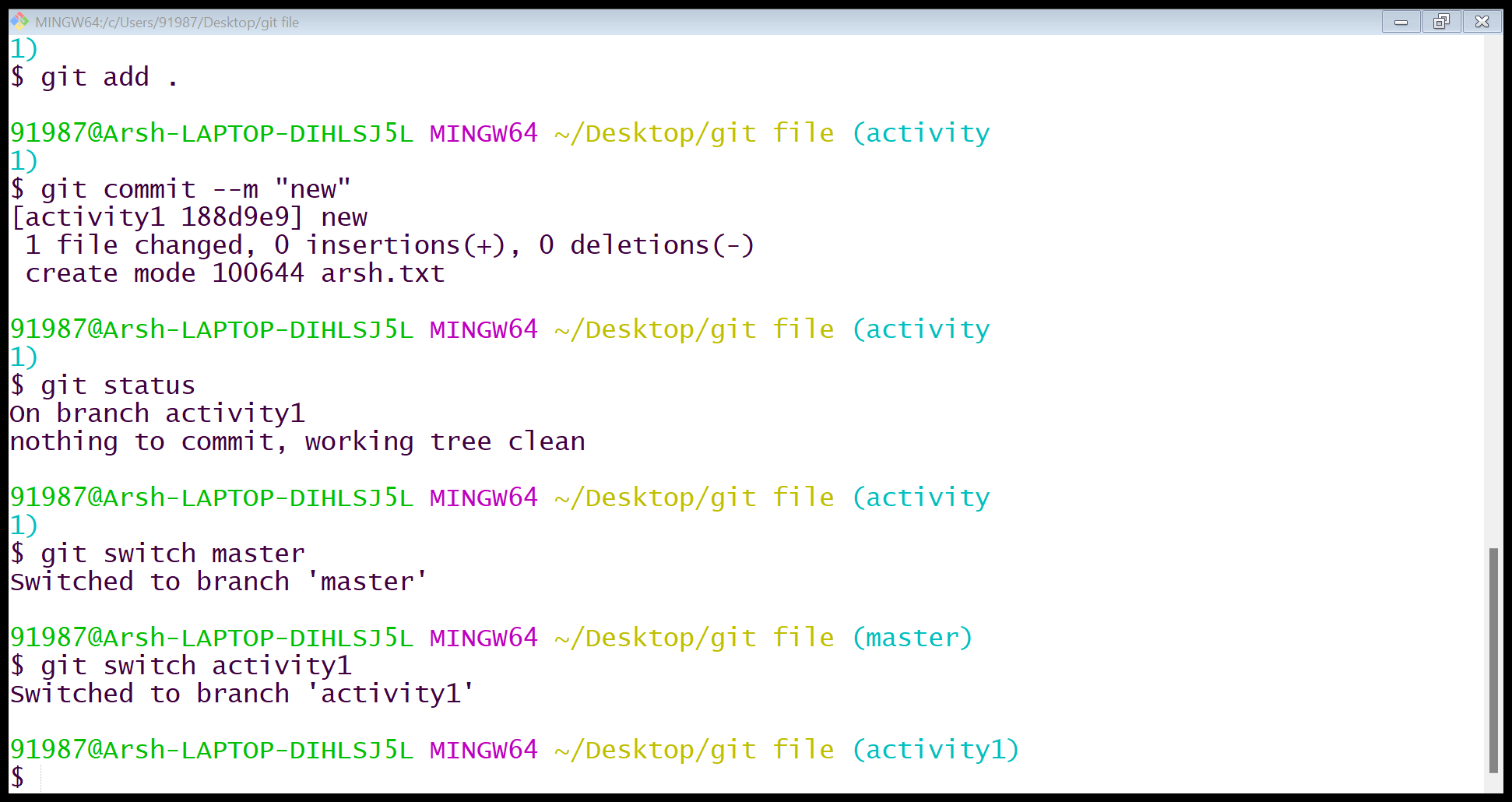
* git switch name:

It is a GIT command used to switch the branch from one to another. When the branch is switched the \* is shown on the name of the branch we’ll be working in.



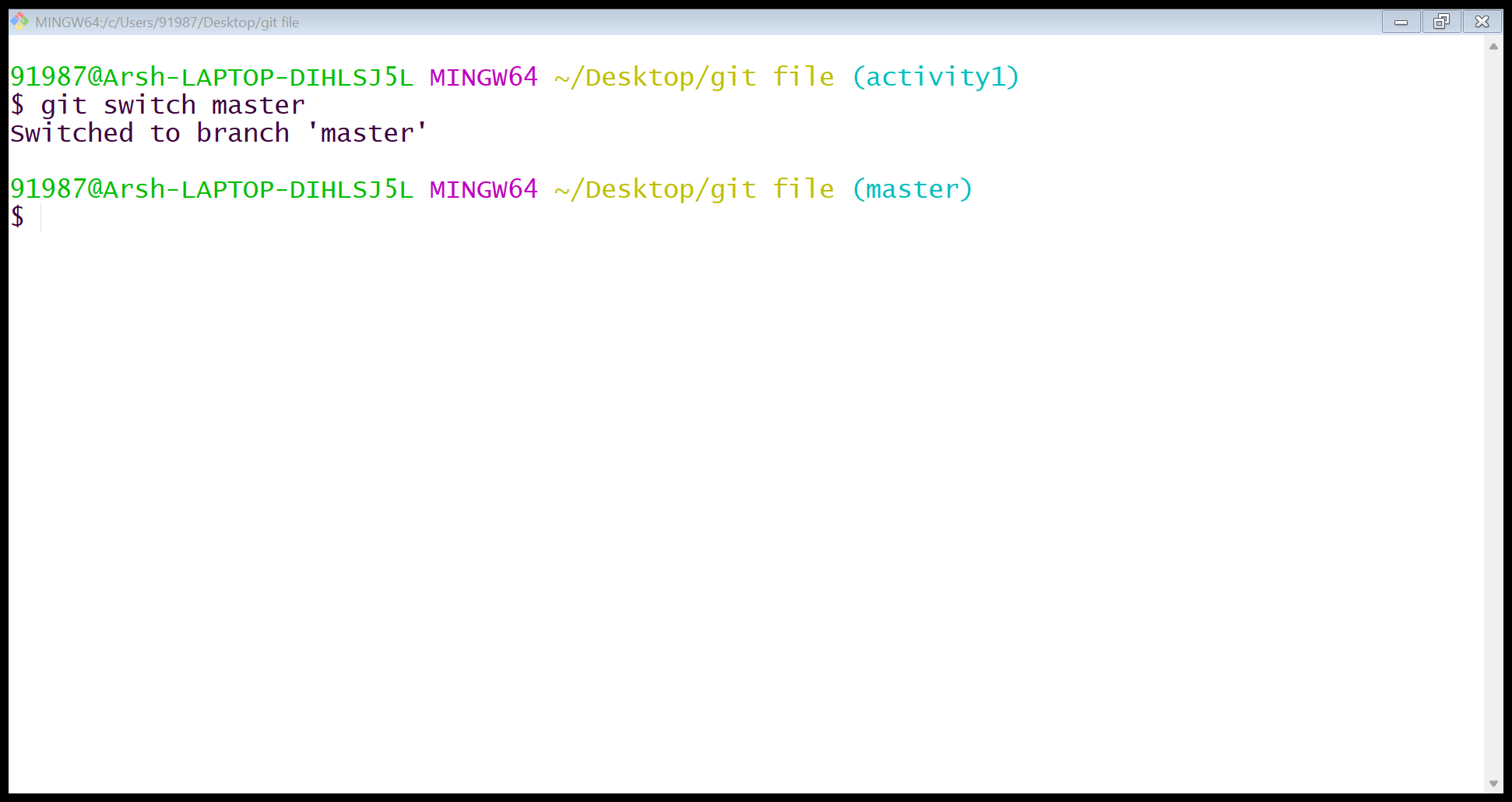
* Addition of a new file to a particular branch:

When we add a new file to the GIT repository while working in the branch, the file will be allocated to the particular branch only and it will be hidden from the repository when we switch branch.



* Switching branch to master:

When we switch the branch to master the files allocated with the activity branch will be hidden from the GIT repository but will exist in the internal database.



* git log:

It is a GIT command which shows a list of all the commits made in the GIT repository. It also shows all the messages committed after the changes are made while working in the project.

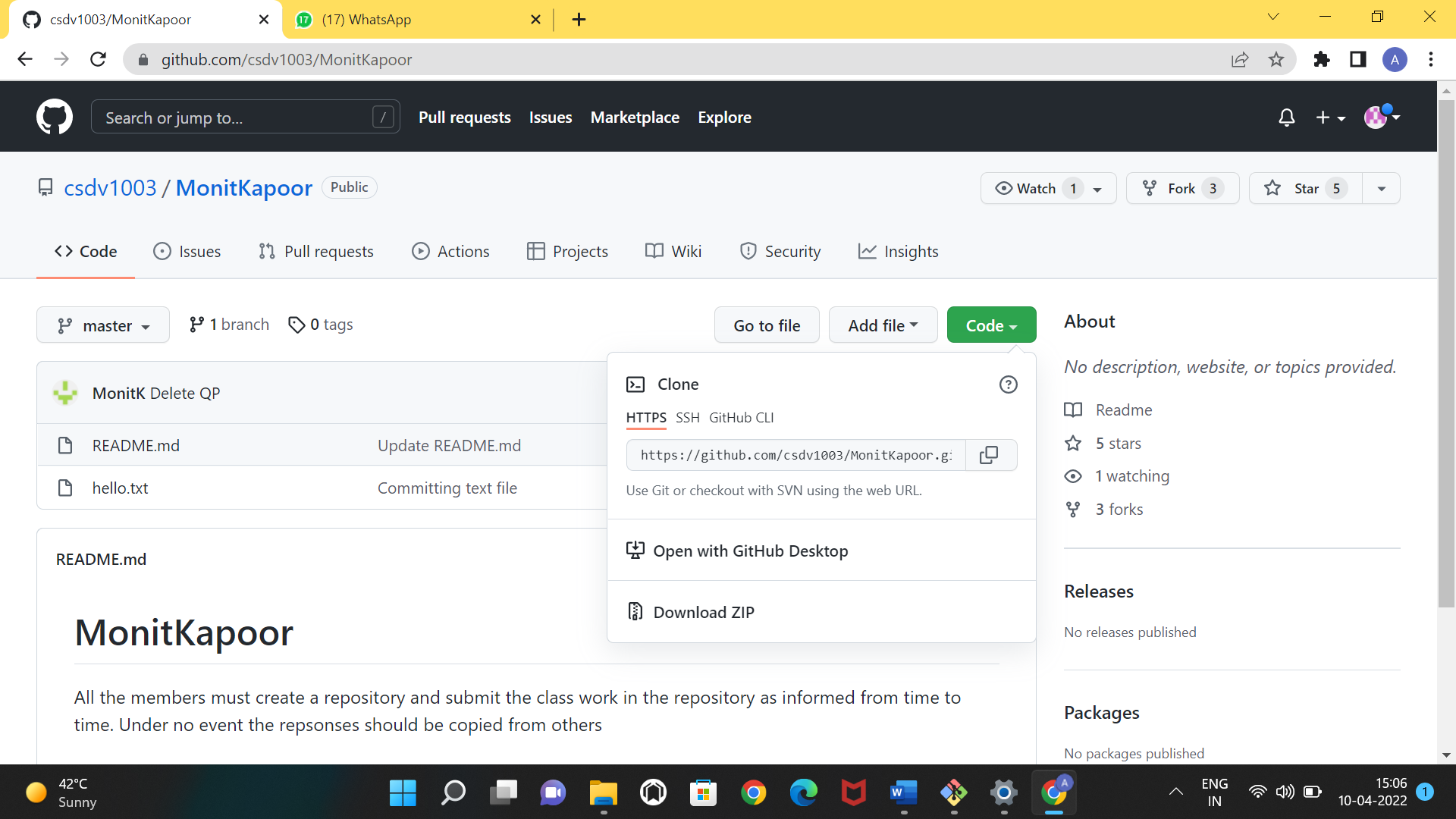


* git clone:

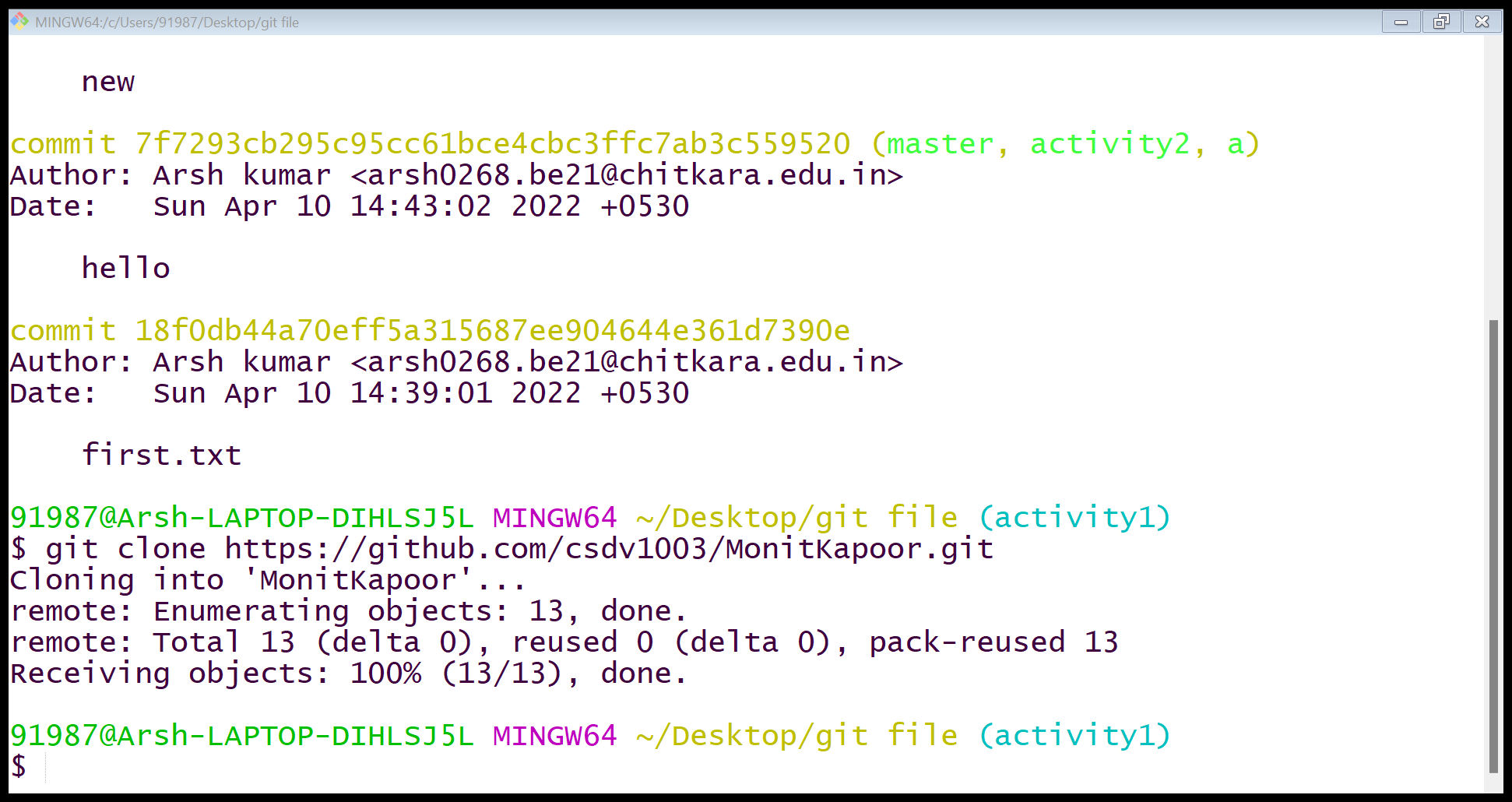
It is a GIT command used to point to an existing repository and make a clone or copy of that repository at in a new directory, at another location.



* Copy the link of the repository from GITHUB.
* Then paste it in the GIT bash using the git clone command.



As now, we’ve downloaded the repository we can see it in the folder we’re working in.



* git branch -a:

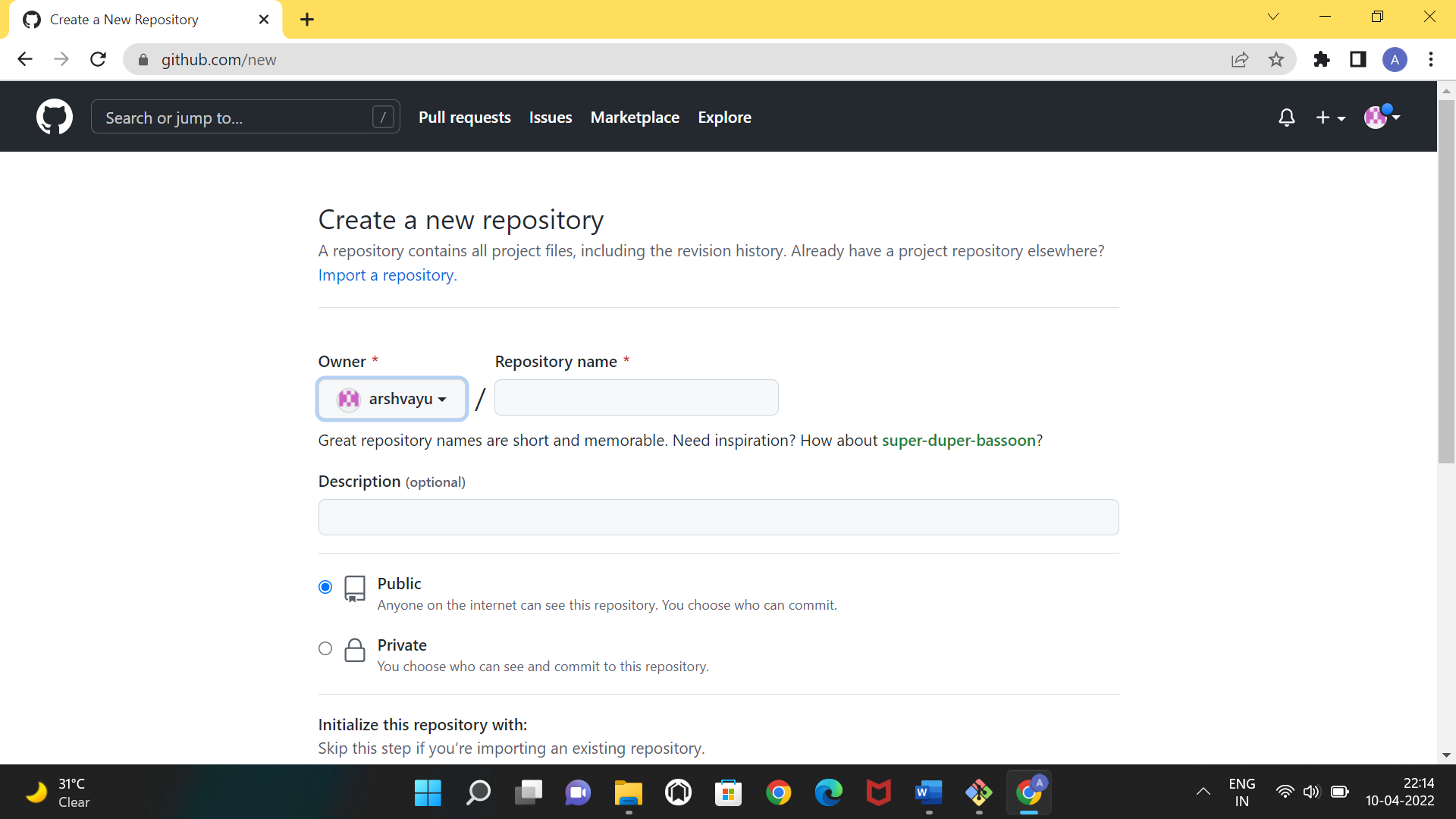
It is a GIT command used to check all the active branches present in the GIT repository.



PRATICAL-4

Working with remote repositories

## First we have to create a new repository on git hub. for new making a repository open your git hub account then click on plus button then click on new repository.

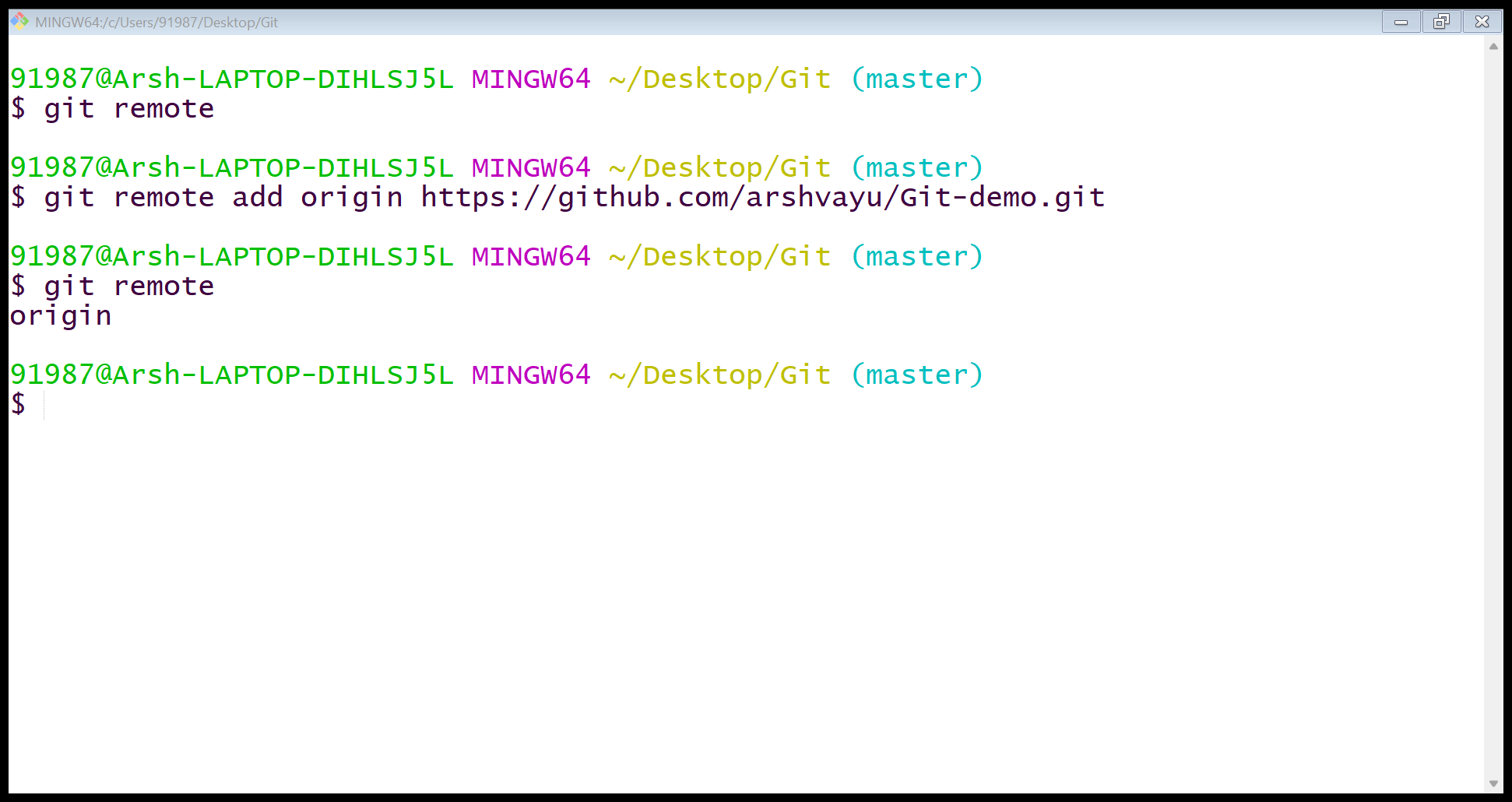


Fill the repository name that you want name it.

After doing this git hub create repository for you then copy the link.

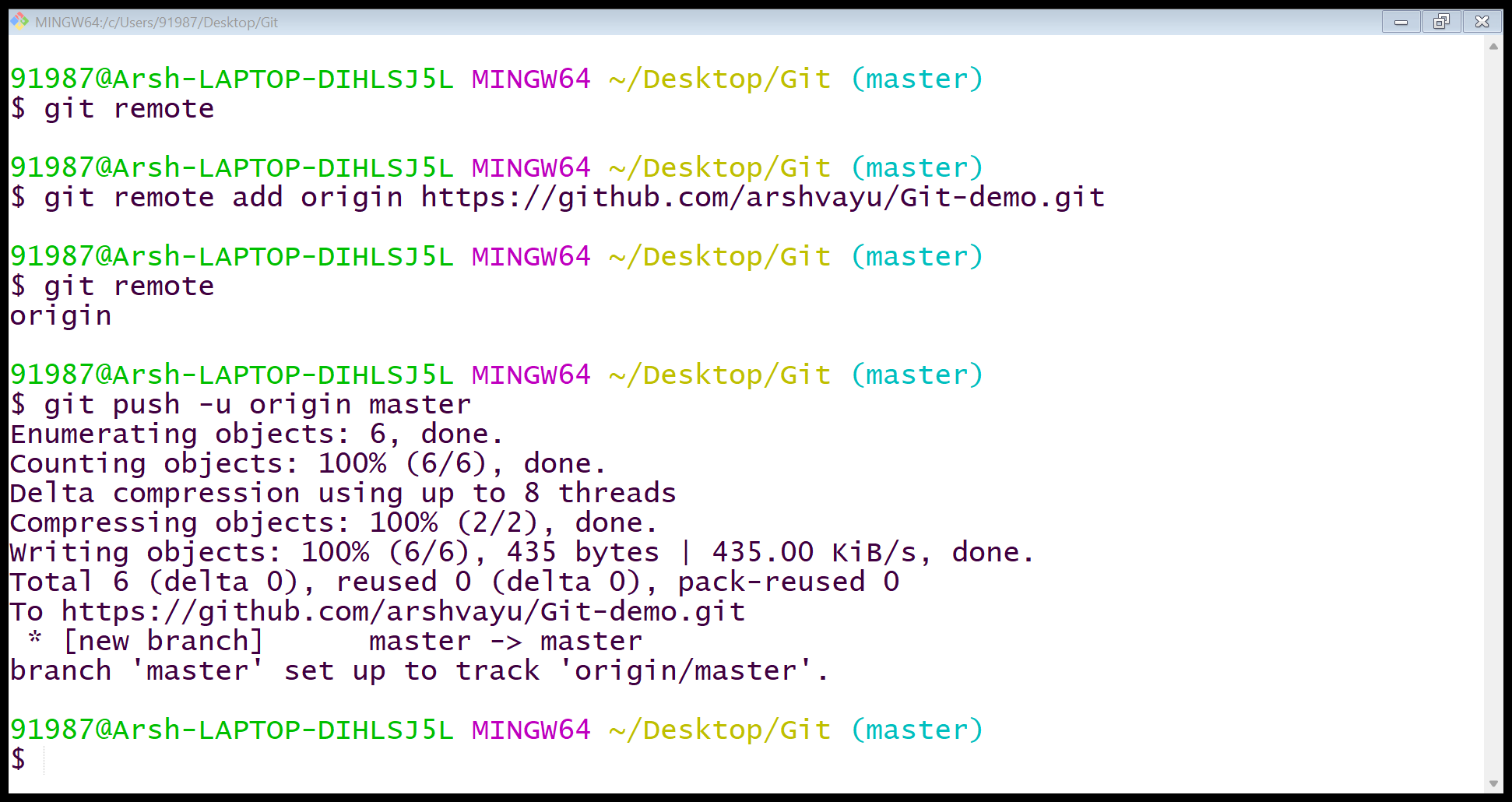
* Git remote add origin https link:

It git command used to add repositories into git hub .



* Git push -u origin master:

It is git command used for push all files to git hub.

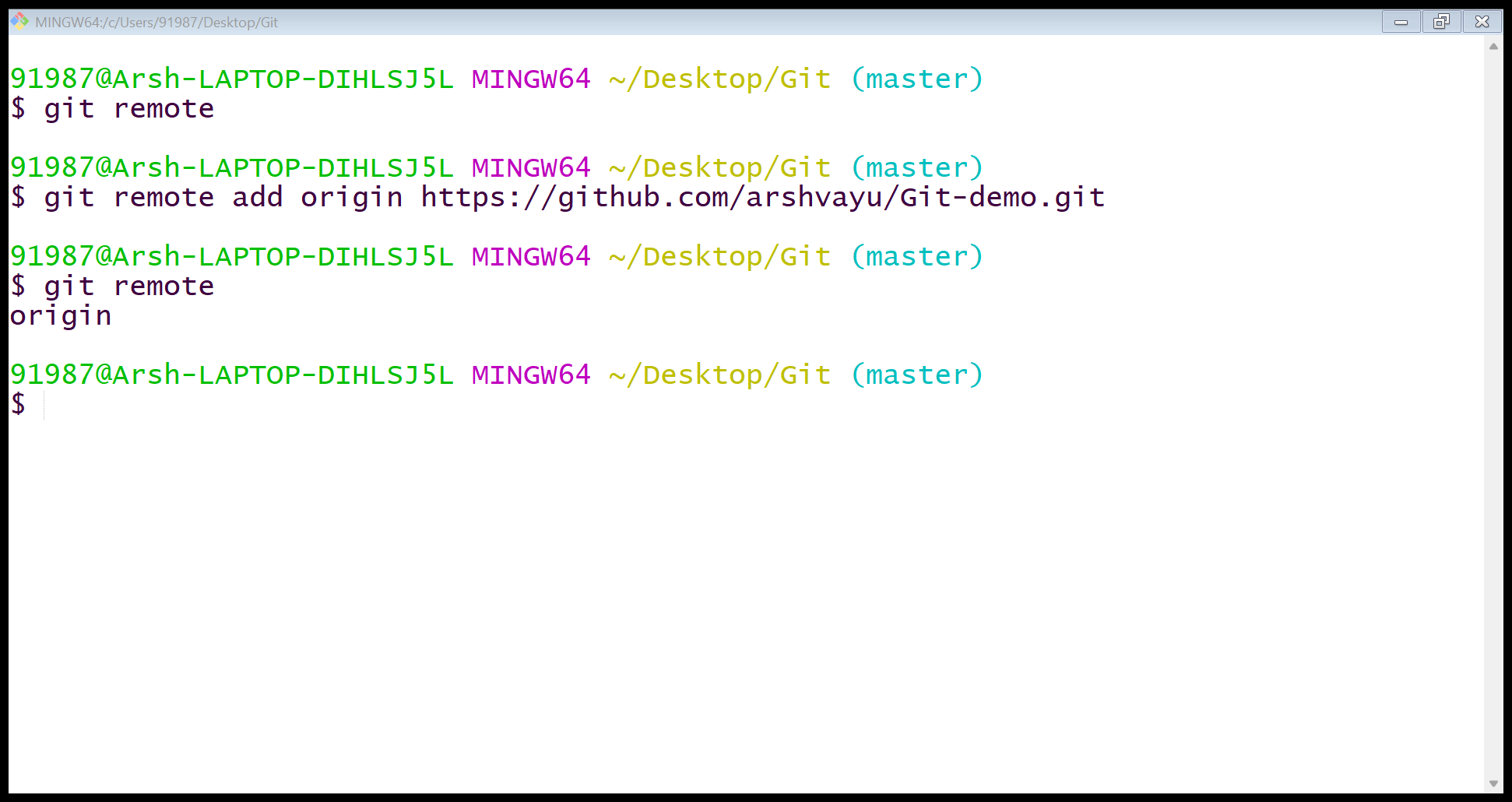


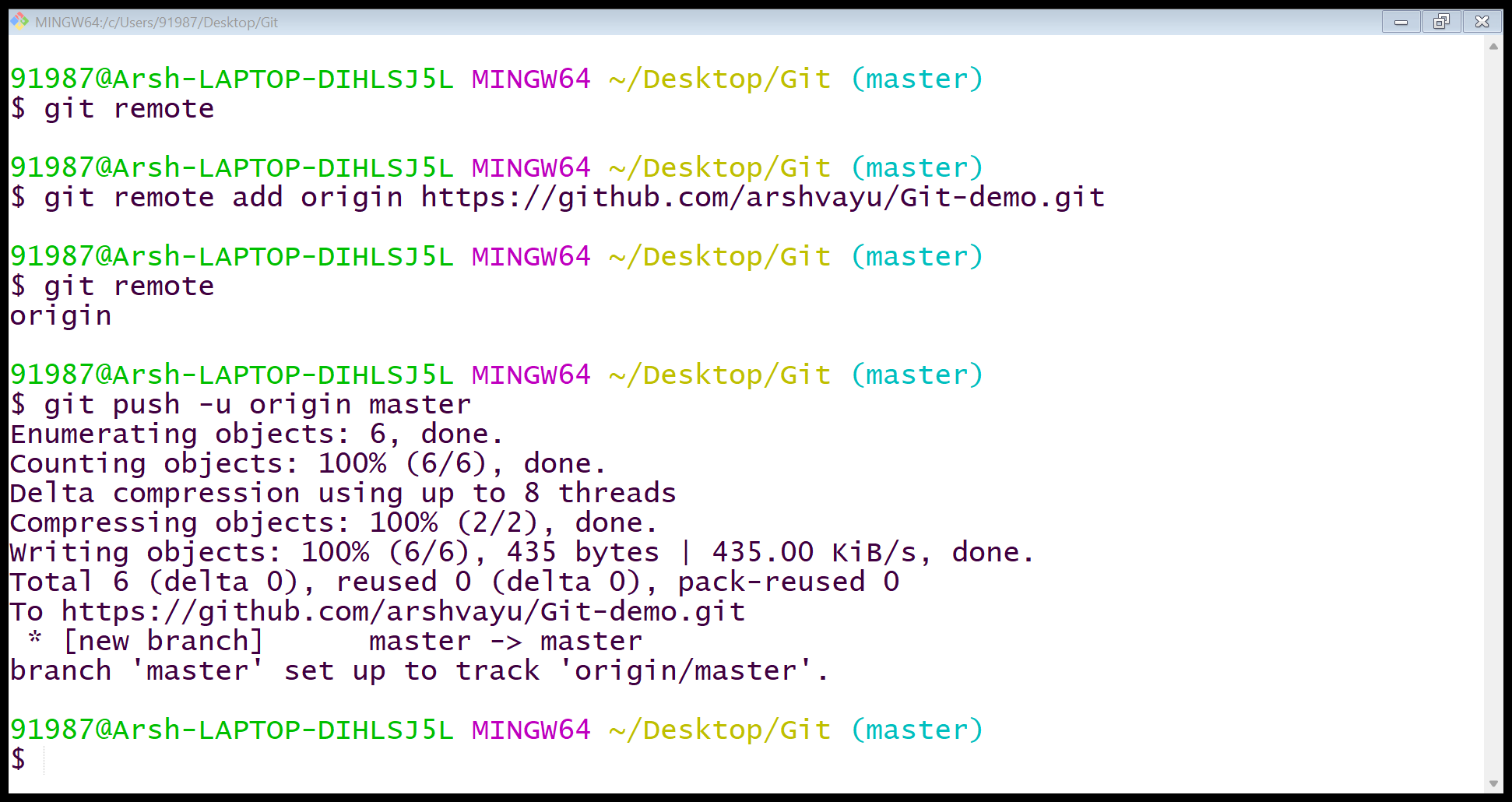
Practical-5

1. Create a new repository.

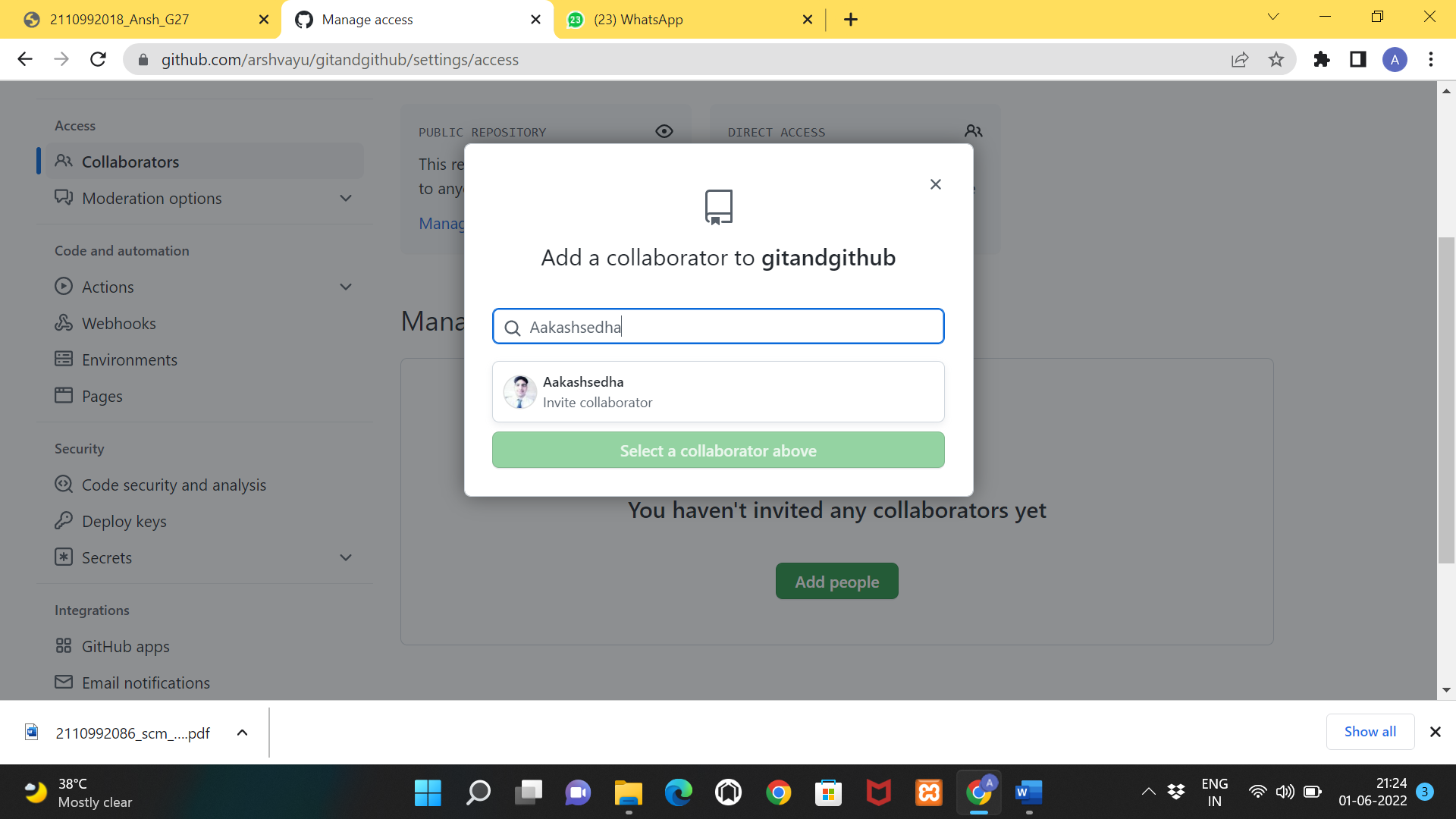


1. Now copy the HTTP link of your repo and paste it on your ‘Git CLI’, and merge the local repo in remote repo (i.e.) temporary.

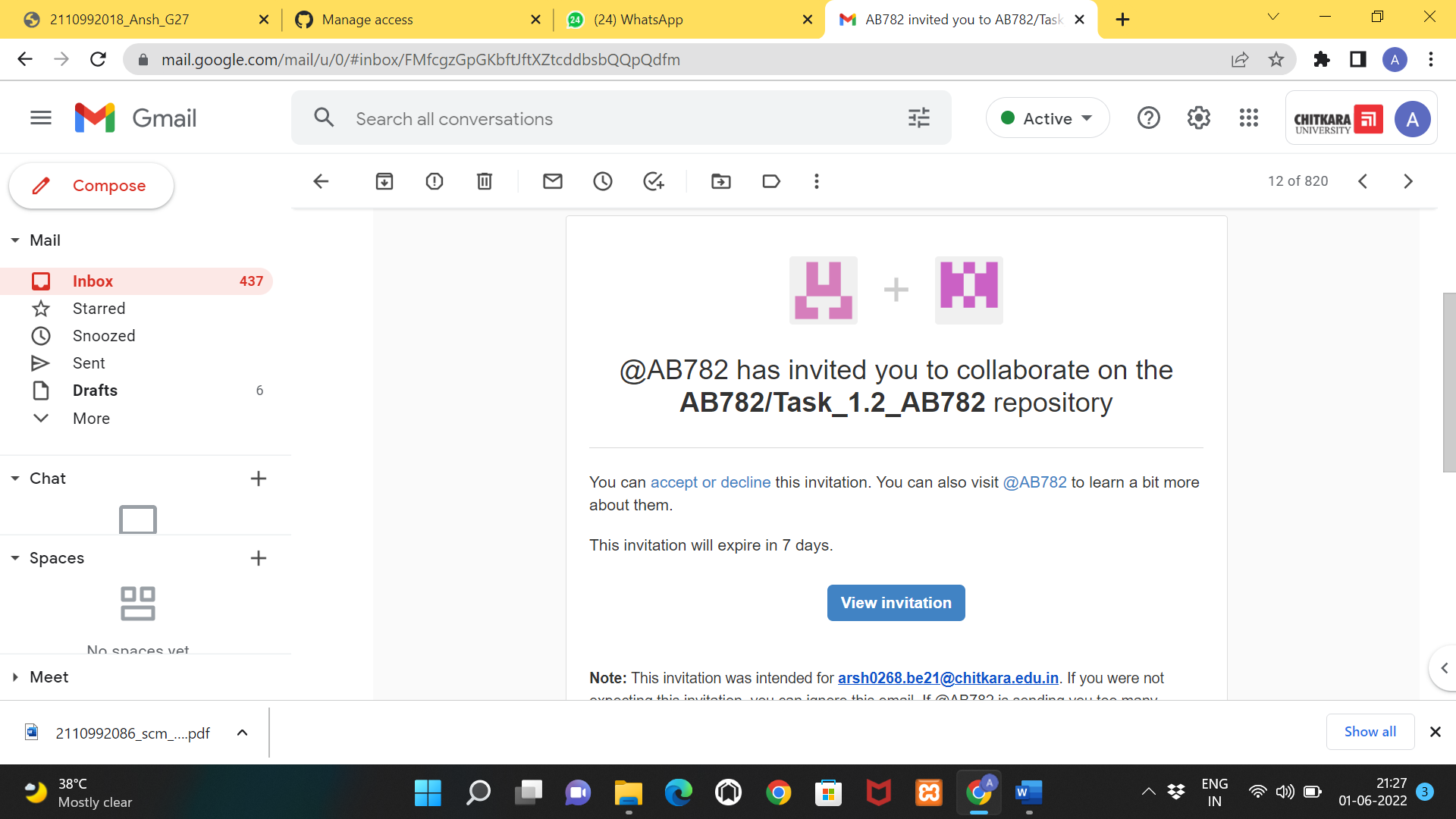




1. Go to Collaborators in repo setting, and the username or email of collaborator you want to add in your repo.



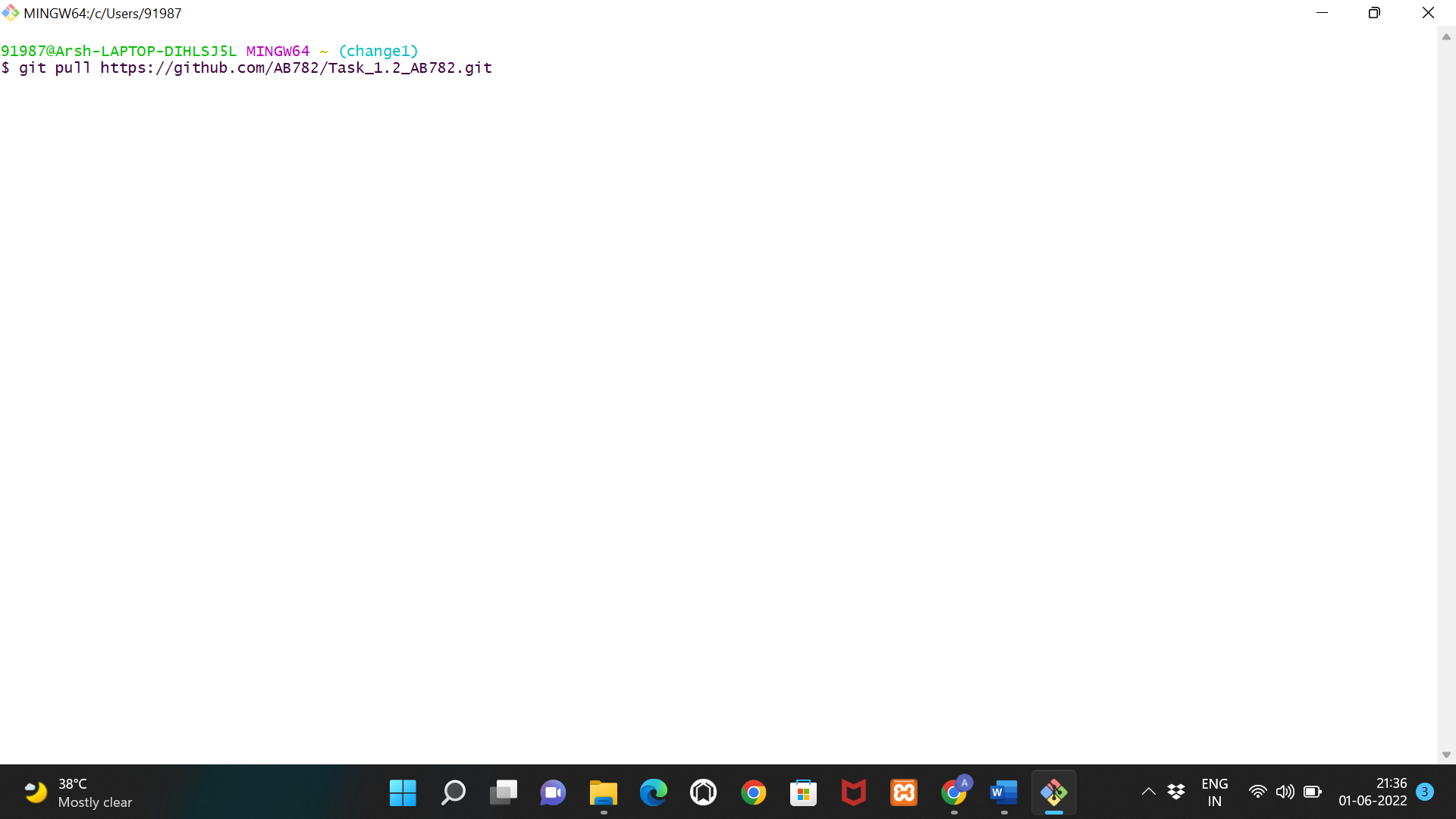
1. Invitation mail is send to collabrator and the collabrator has to accept this invitation.



Fork and Commit

1. Type “Gitpullhttps://github.com/AB782/Task\_1.2\_AB782.git

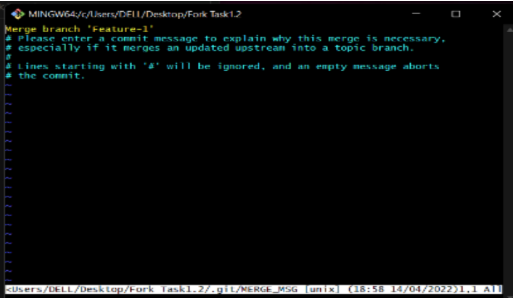
Git pull <url> This command is used to fetch the remote repo or to clone the repo.



2.Create a new feature branch “Feature-1”.Create a file and do changes in it and commit it.

Merge and Resolve conflicts created due to own activity and collaborators activity.

1. Do changes in master branch and commit those change and checkout to feature 1 branch and again do changes and commit it. Now checkout to master branch and merge the feature-1 branch in master.

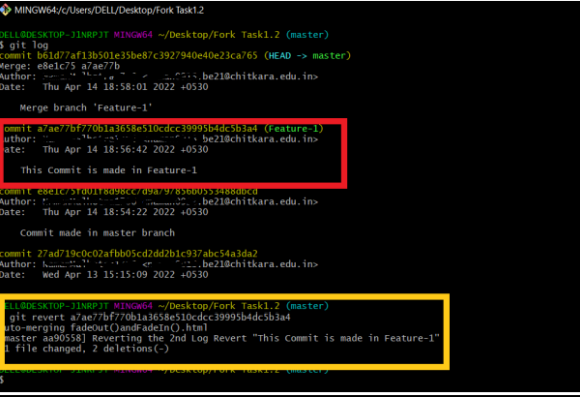




Reset And Revert

Git-revert-Revert some existing commits

1.On Git Bash CLI, Type command “git Command ”. It revert the changes that done before Commit.



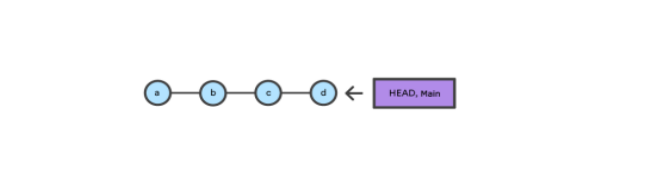
Git revert Head~3;

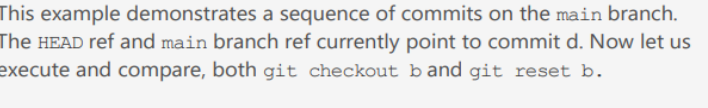
Revert the changes specified by the fourth last commit in head and create a new commit with the revert changes

Git-reset-Reset current HEAD to the specified state

AT the surface level, git reset is similar in behaviour to git checkout. Where git checkout solely operator on the head of ref pointer , git

Reset will move the Head ref pointer and the current branch ref pointer. To better demonstrate this behaviour consider the following the example





Git Reset reset is the command we use when we want to move the repository back to a previous commit, discarding any changes made after that commit.

