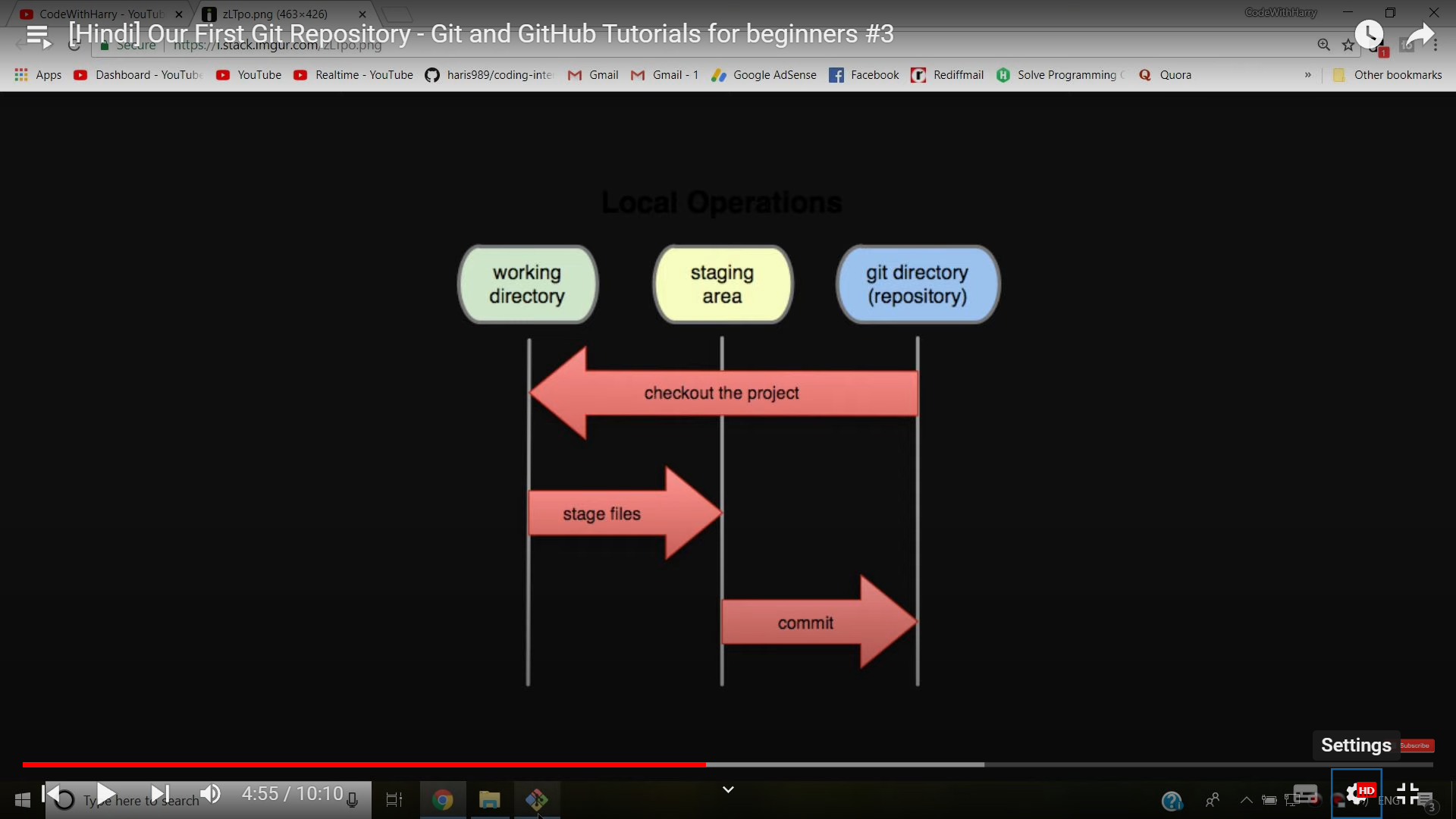
**VERSION CONTROL SYSTEM :**



git config --global user.name “<username>” => to set user name shown in commit log

git config --global user.email “<useremail>” => to set user email shown in commit log

\*The --global flag is not mandatory, if removed the changes will be limited to the current repository

git config --list => to get all the configuration data

1. Create a local Git repository
2. GUI

VCS(TopBar) -> Enable Version Control Integration -> Git

1. CLI

Open git bash -> Navigate to the directory (pwd , cd , ls , ls -a)

Command:

git init

1. Adding things to staging area

Use a .gitignore file to specify the files that you don’t want to keep track of

CLI Commands

git add . => to add all the files and folders in current directory

git add index.html error.html => to add specified files

To unstage files

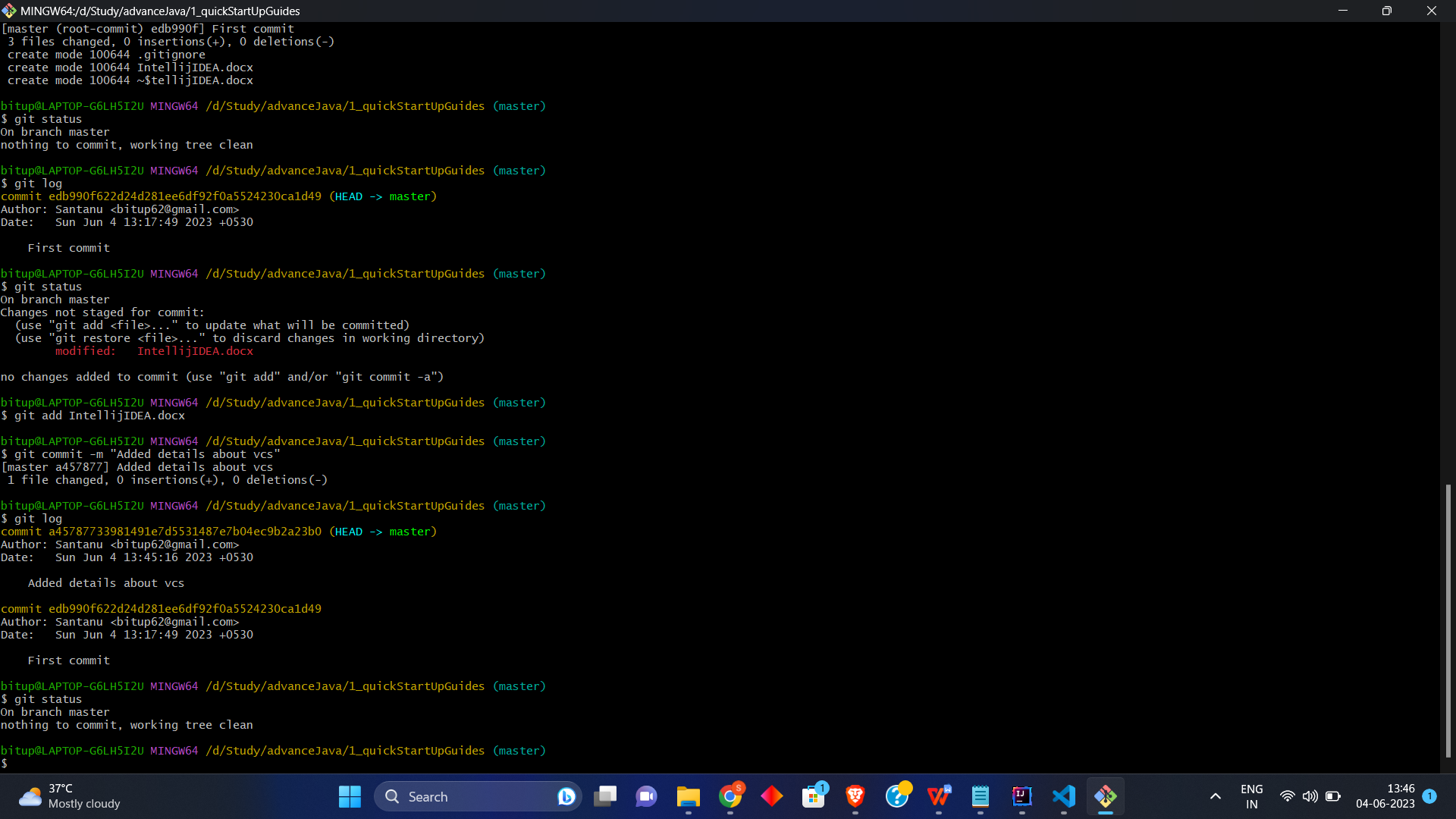
git reset HEAD <file1> <file2> =>to unstage the files but maintain the modifications made on them

1. Commiting the files put into git repository

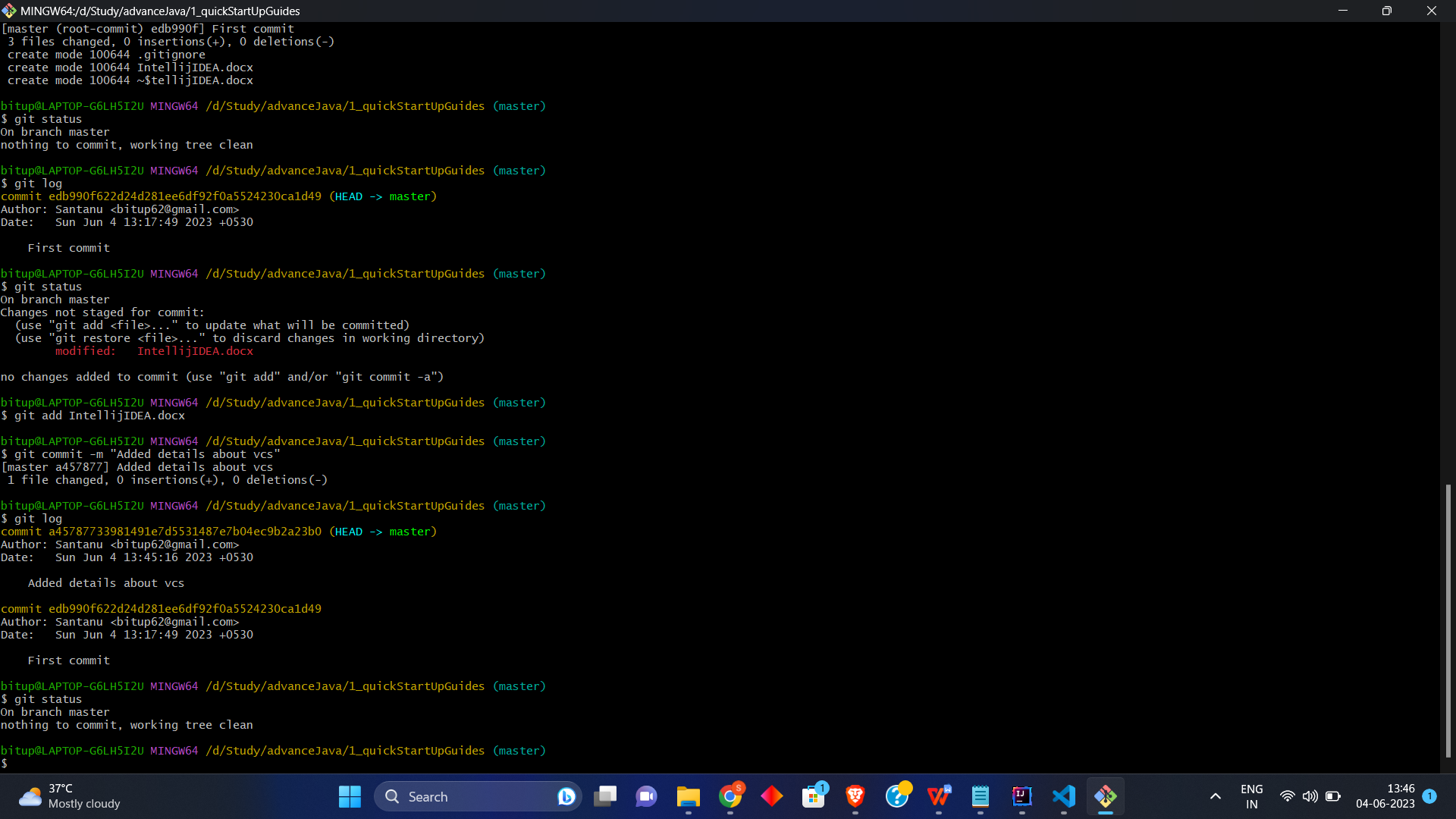
git commit -m “<Commit message>”

1. Checking out Repo details

git status => To check out the files in the staging area



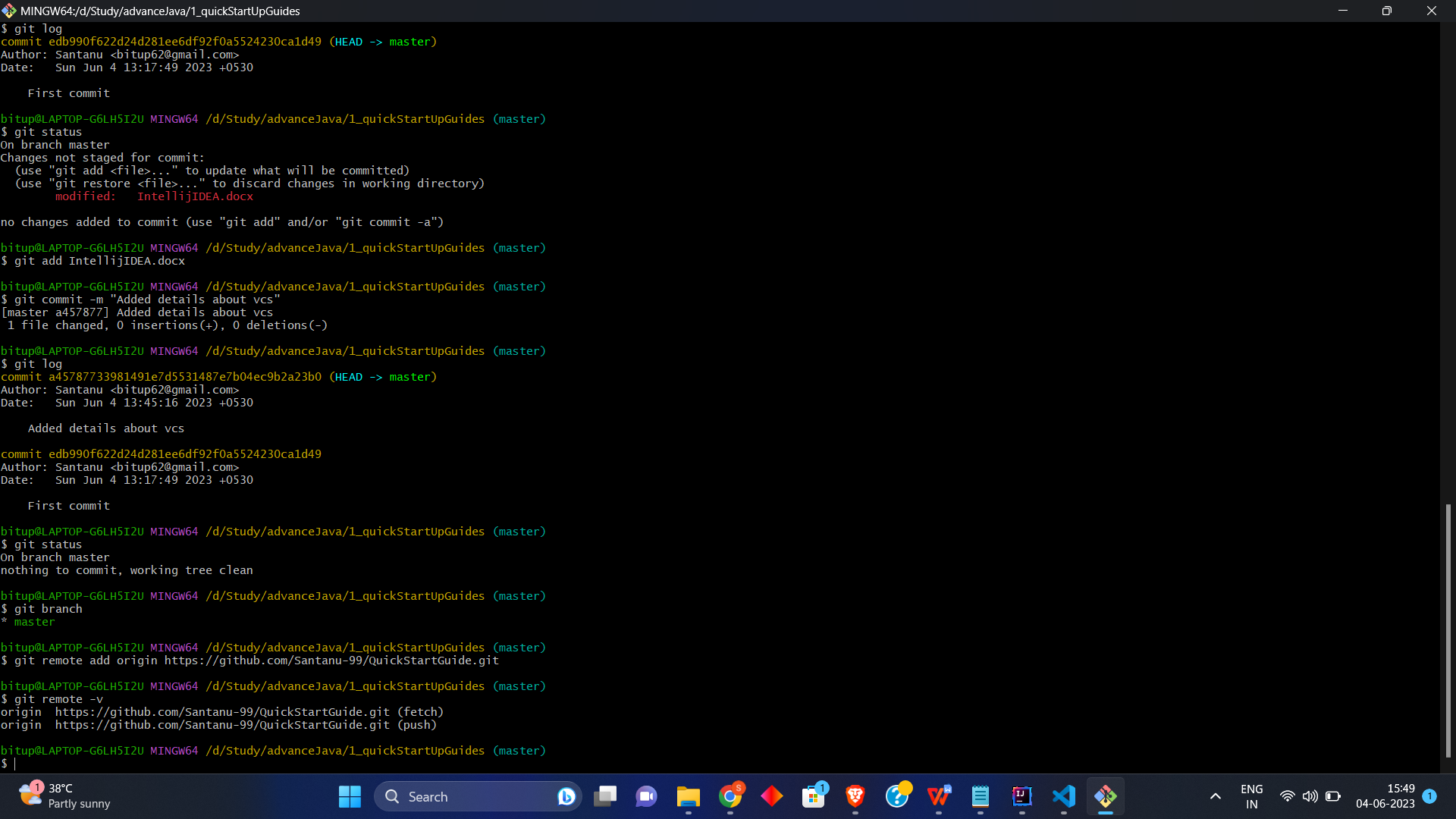
git log => To see the commit history



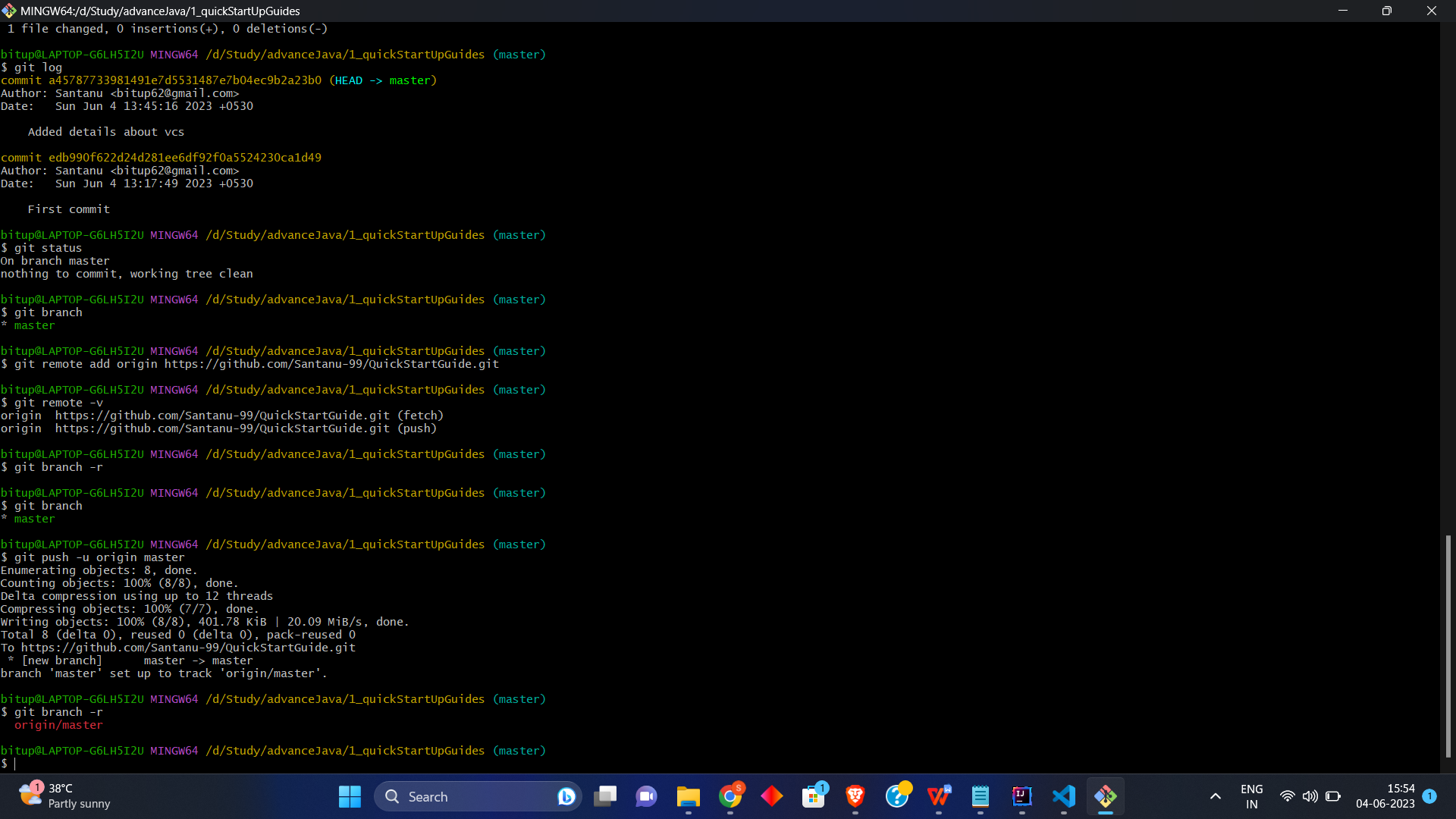
git remote -v =>To see all the remote repo

git remote add <remote repo alias> <remote repo URL> => To add a remote repo

git branch => To see all the branches in local repo

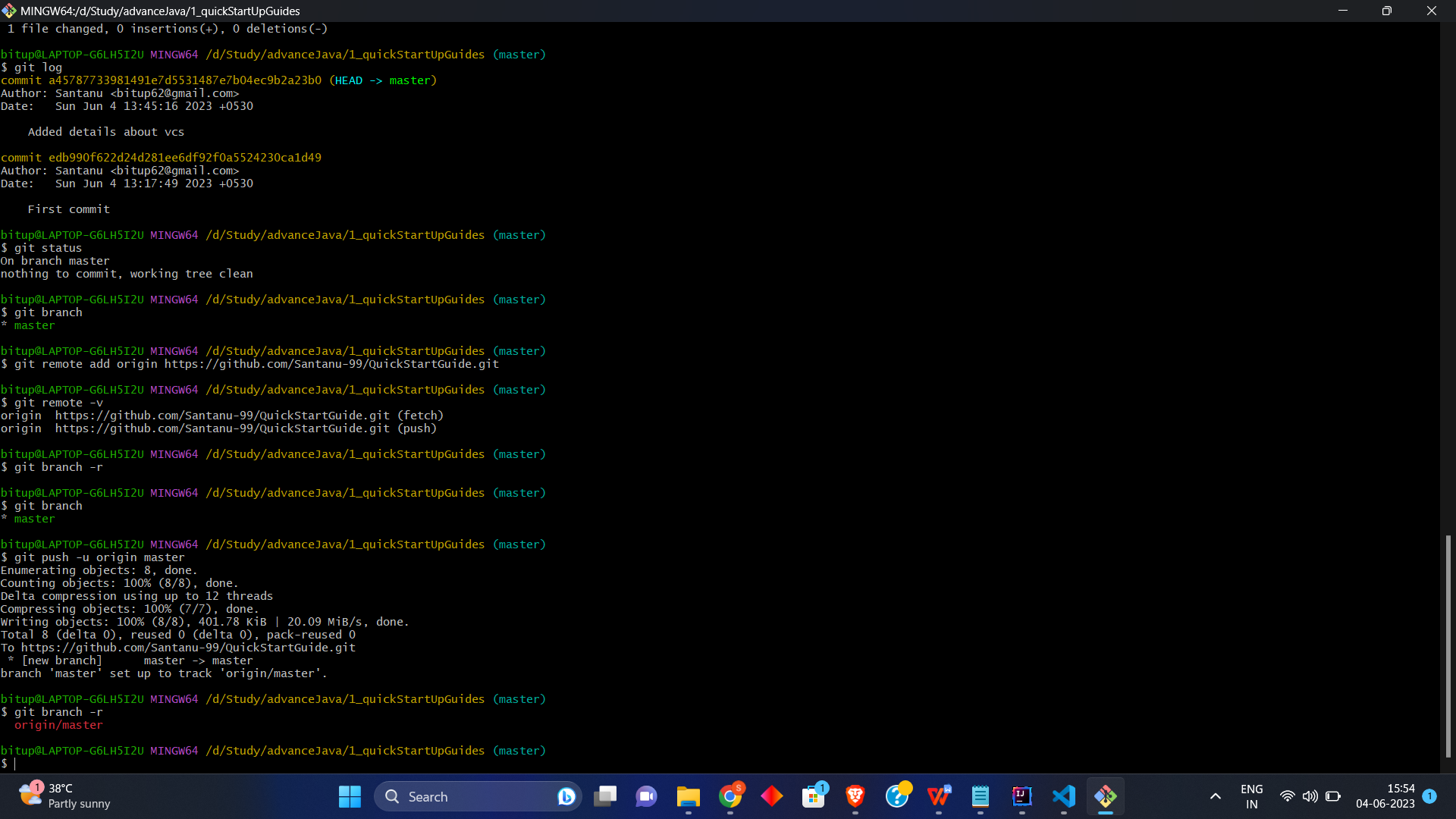


git branch -r => To sell all the branches in remote repo



1. Pushing commit to remote repo

git push -u <remote repo alias> <remote repo branch>



1. To delete a local git repository

rm -rf .git => removes the .git folder from the directory (essentially removing the repository , but other content of the folder does not get affected)

1. To Clone a project from github

GUI

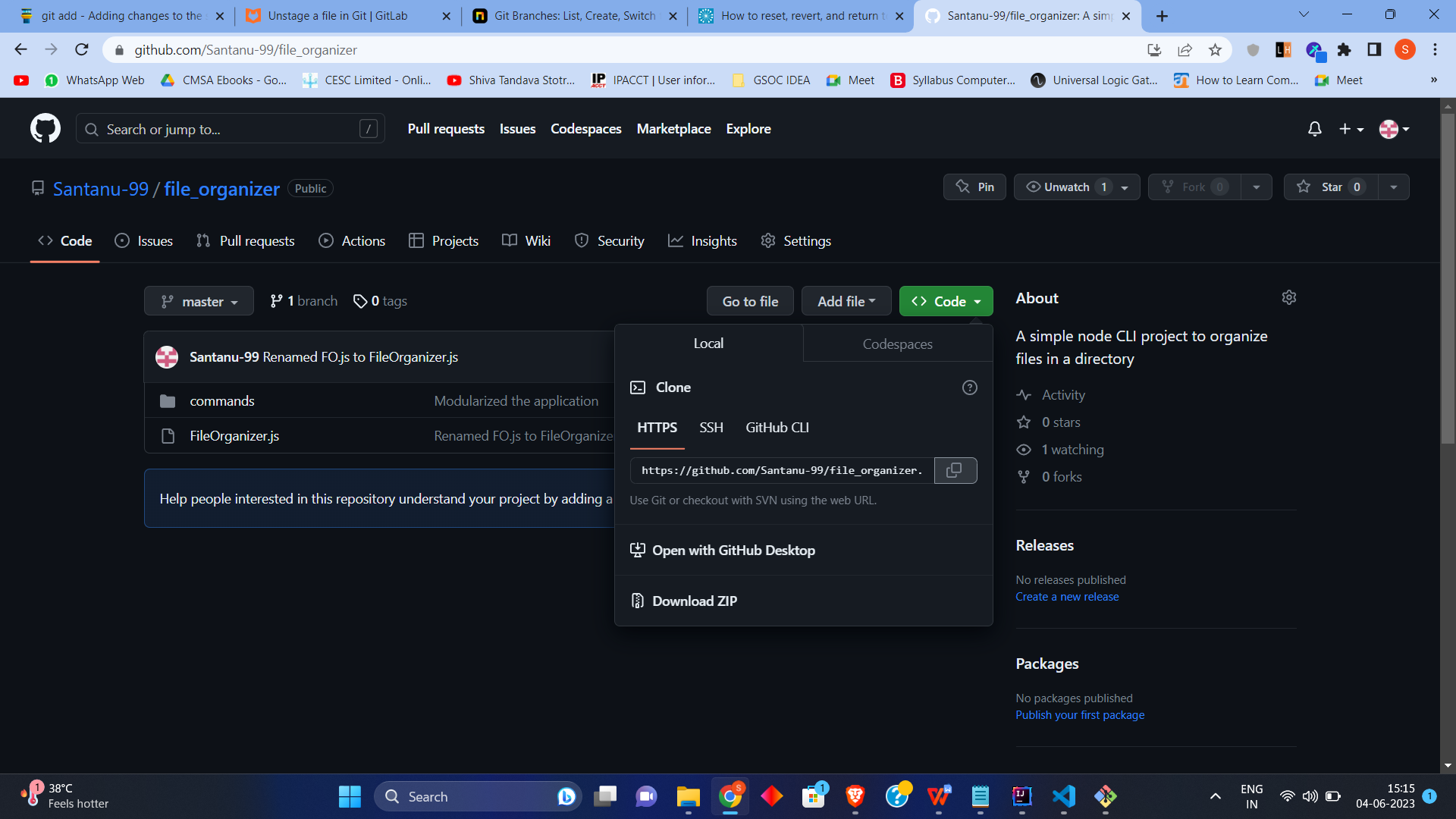
Open with VCS -> select git

-> set the directory where you want to save the repo clone AND set the remote repo URL

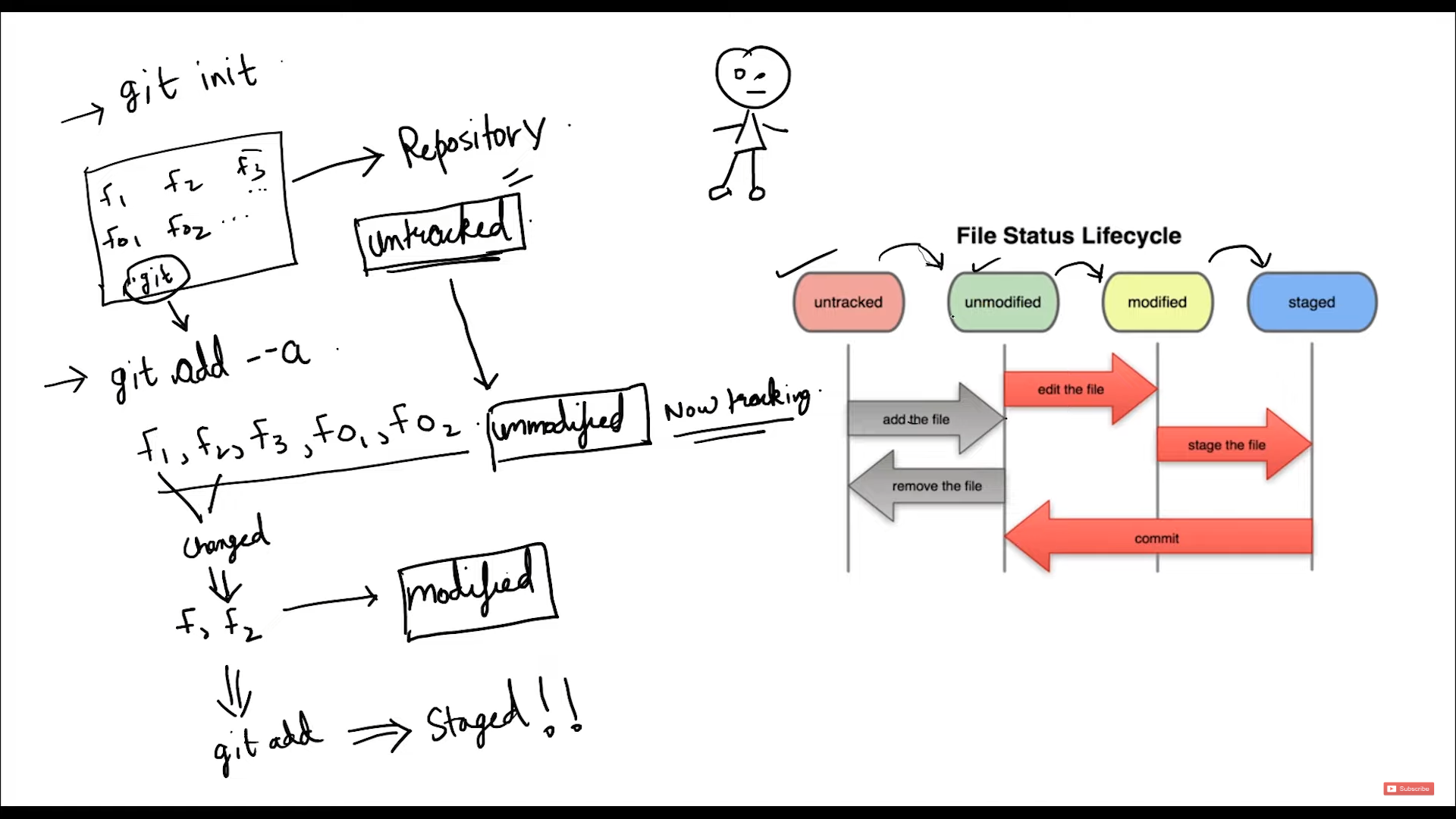
OR

git clone <Repo URL>

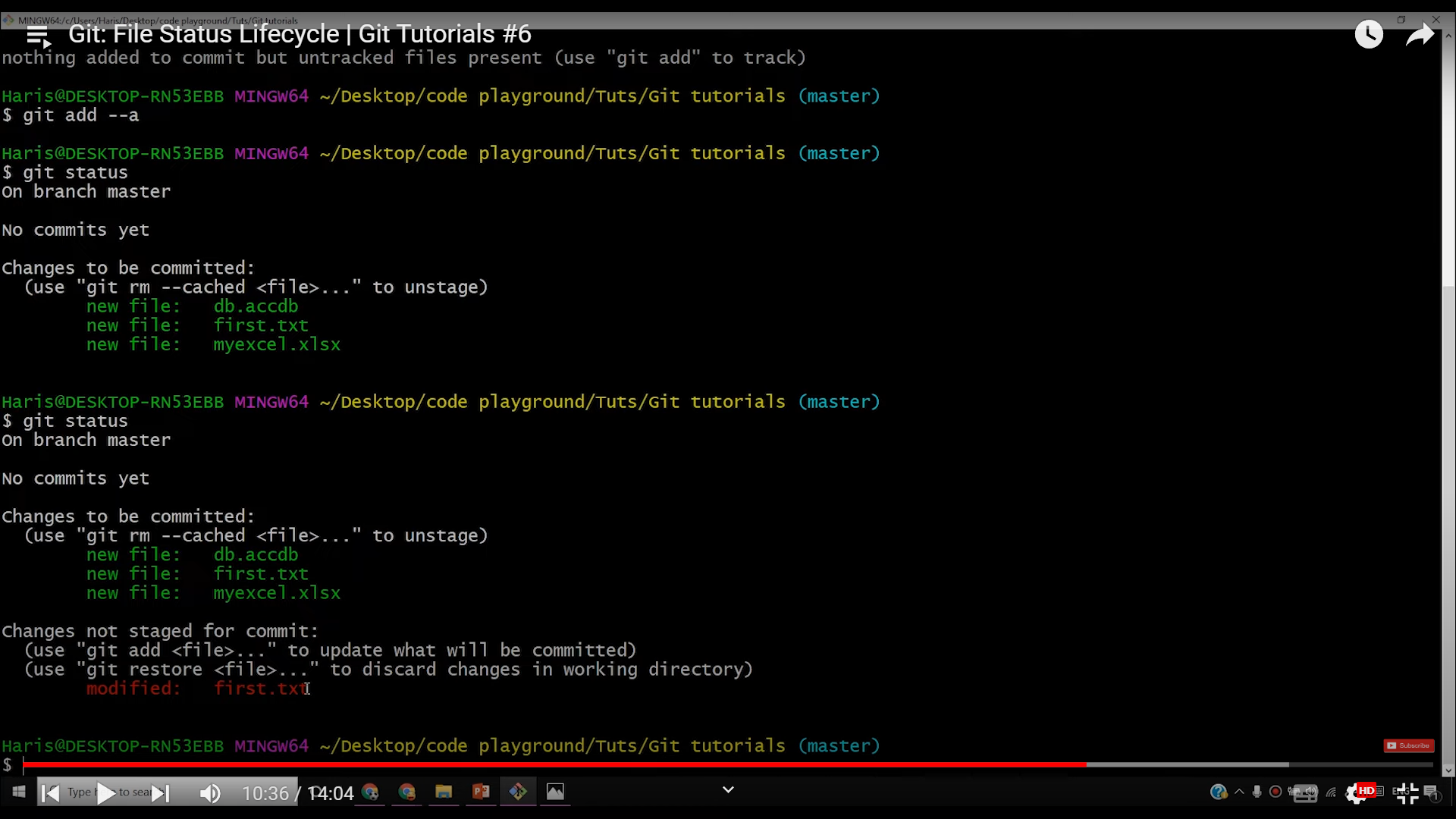
Repository URL From GitHub



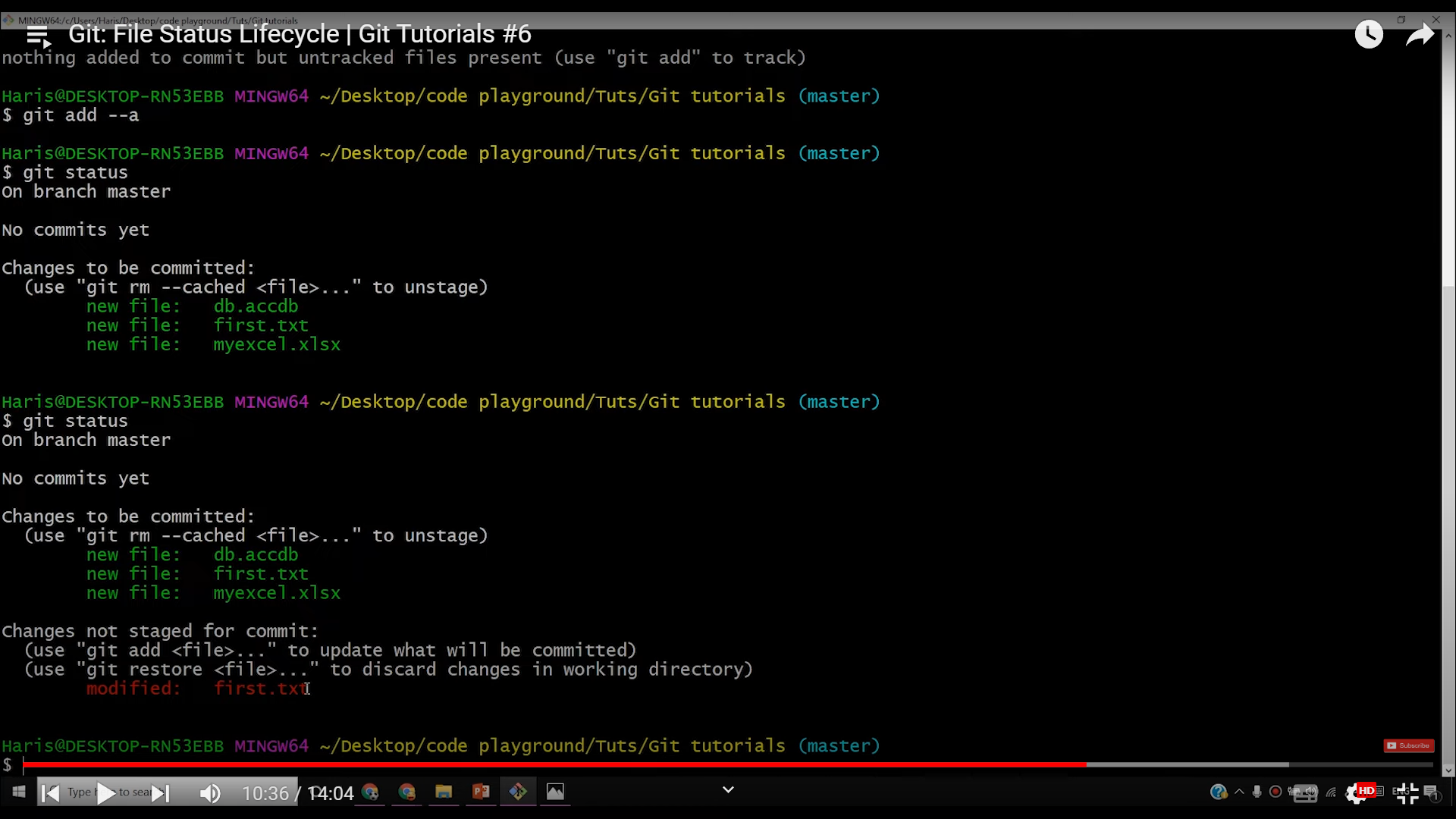
1. File Status Life Cycle



After adding the files into staging area



After performing some changes to already staged file: first.txt



In order to commit the updated first.txt , in next commit command execution

we will have to add the first.txt into staging area by

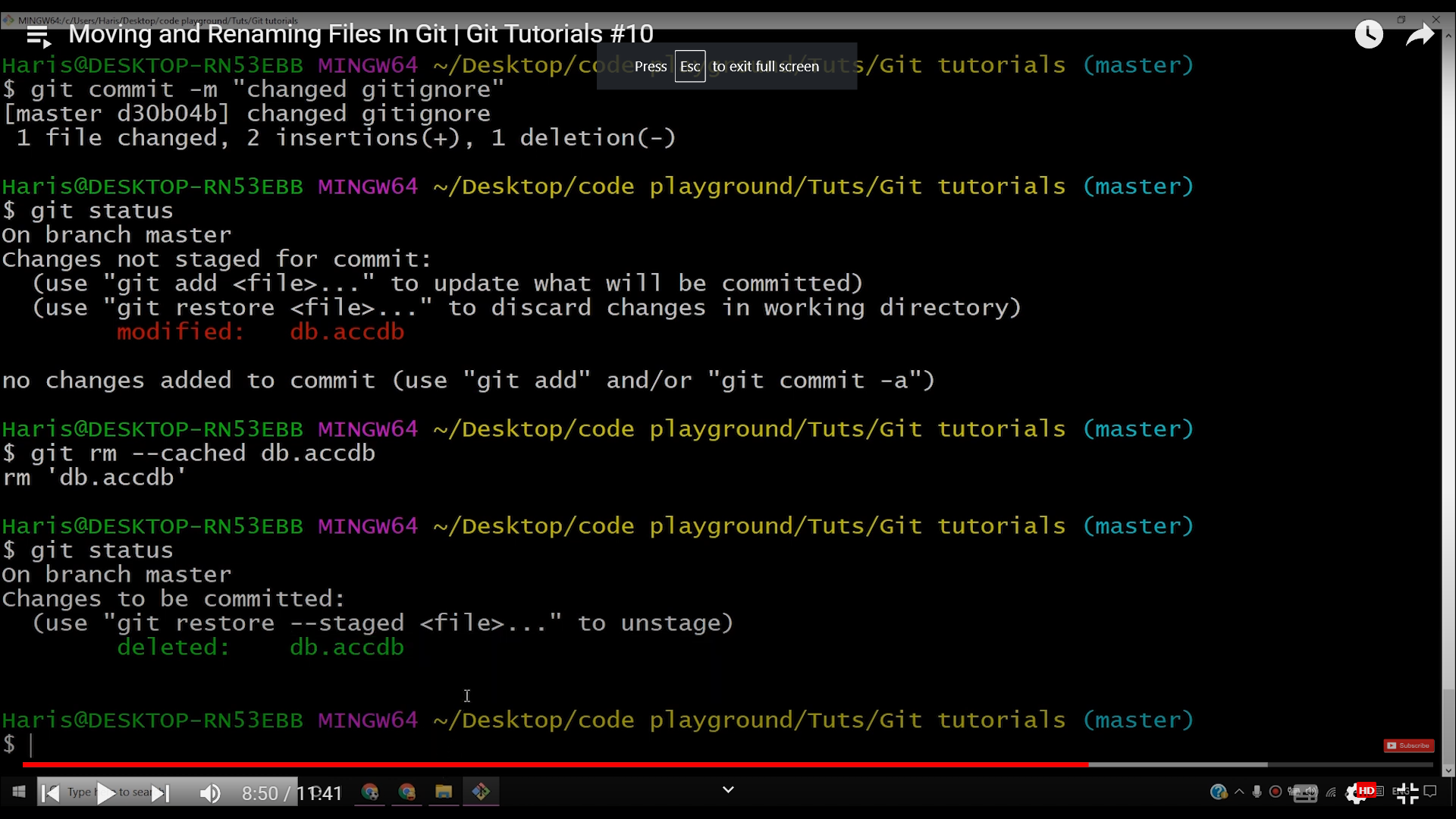
git add first.txt

Then,

git commit -m “<commit message>”

Else if we directly execute commit command without staging the modification, the modifications won’t get committed.

To move a file from Tracked to UnTracked

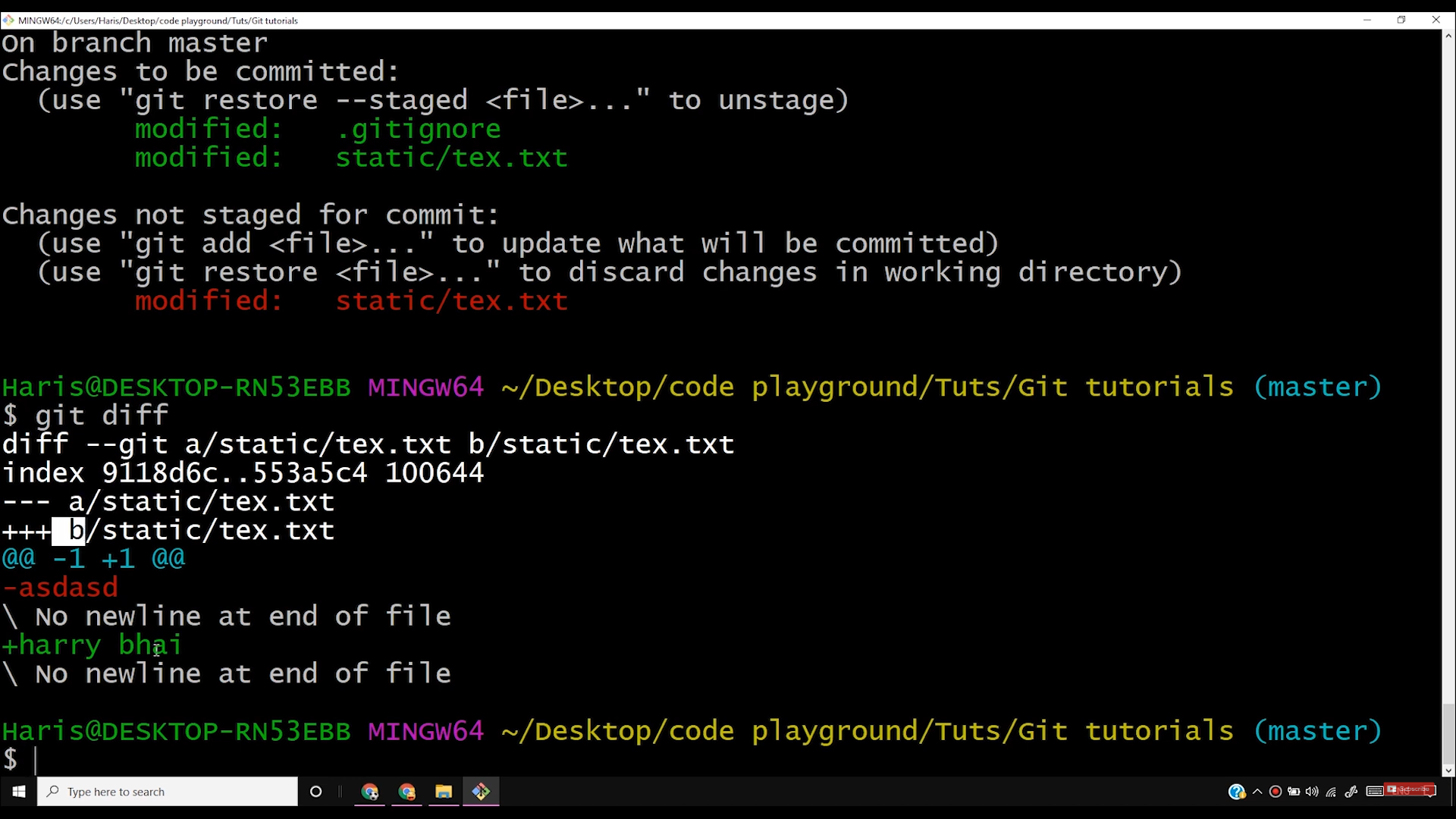


Steps:

a)git rm --cached <fileName> => To make the file untracked(note this will not delete the file)

b)Add the <fileName> record to .gitignore file => To make the file not appear in untracked files

1. Comparing the staged and modified(if exists) version of the same file



Here tex.txt is compared

Command used:

git diff

1. Comparing last commit with current staging area

git diff --staged