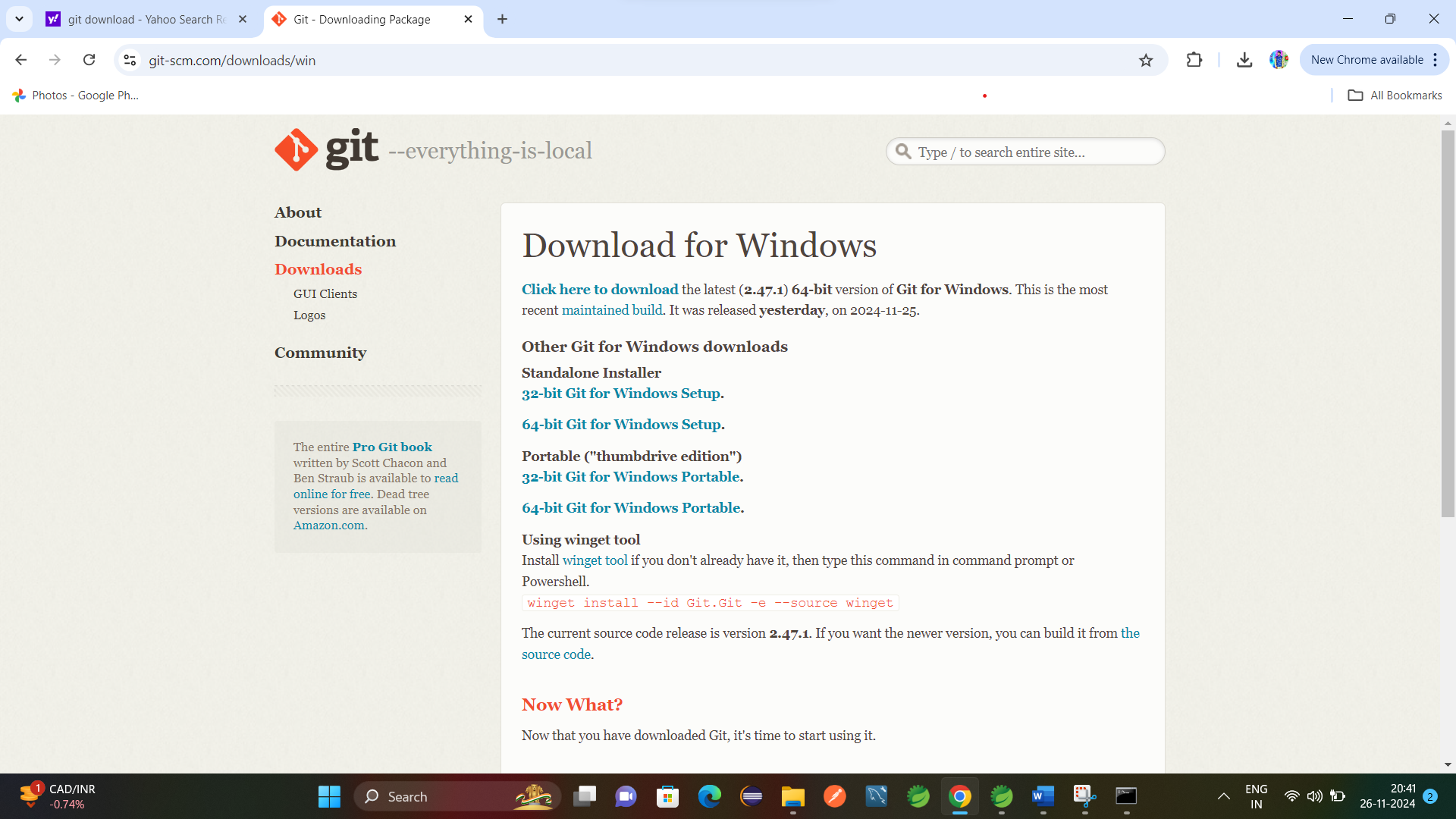
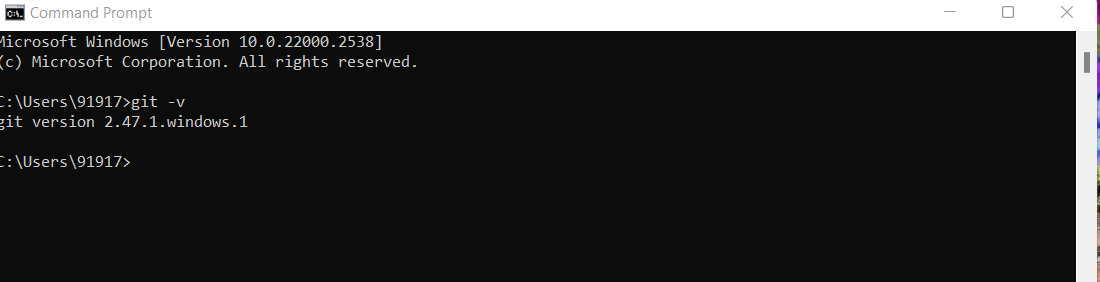
**Downloading Git Into Local System:-**

Install Git from browser and check weather its installed are not in Command Prompt by using **git -v**





**Create Git Hub Account:-**

Open browser and create Git Hub Account.

User:- chintalauday

Password:- Uday@16091996

**Configure Git User Name:-**

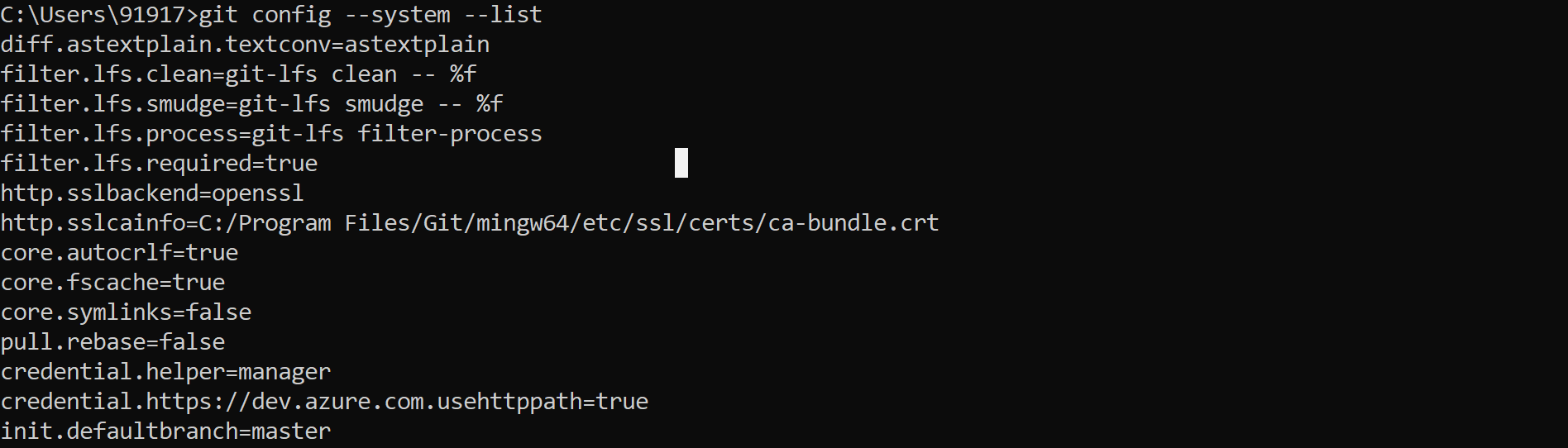
To use git we need to set user name and email.

User cmd

1.To check System Level Configuration.

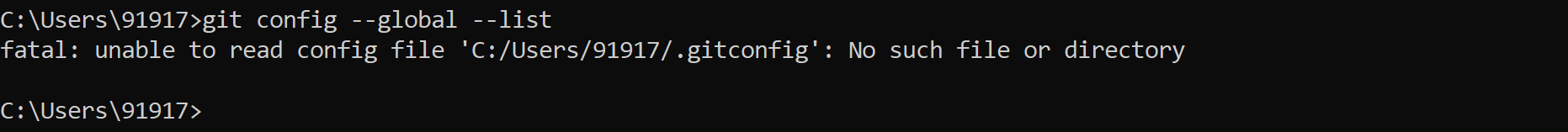
These are default values

**git config --system –list**

****

2.Global Level Config

**git config --global --list**

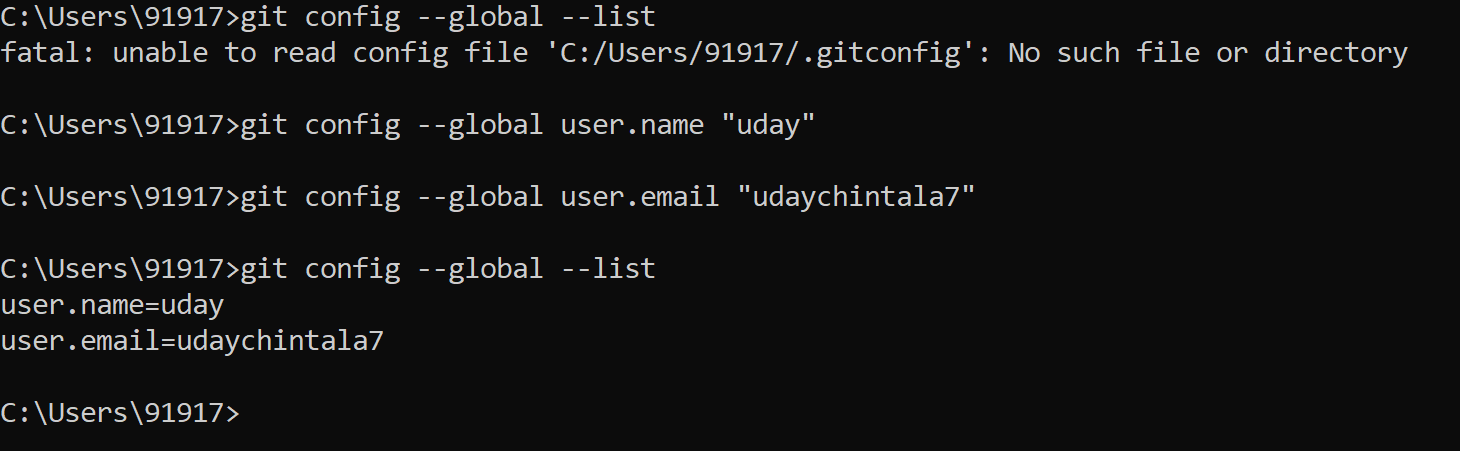


Here we are not getting anything bcz we are not setting any user and mail to git to do this follows as

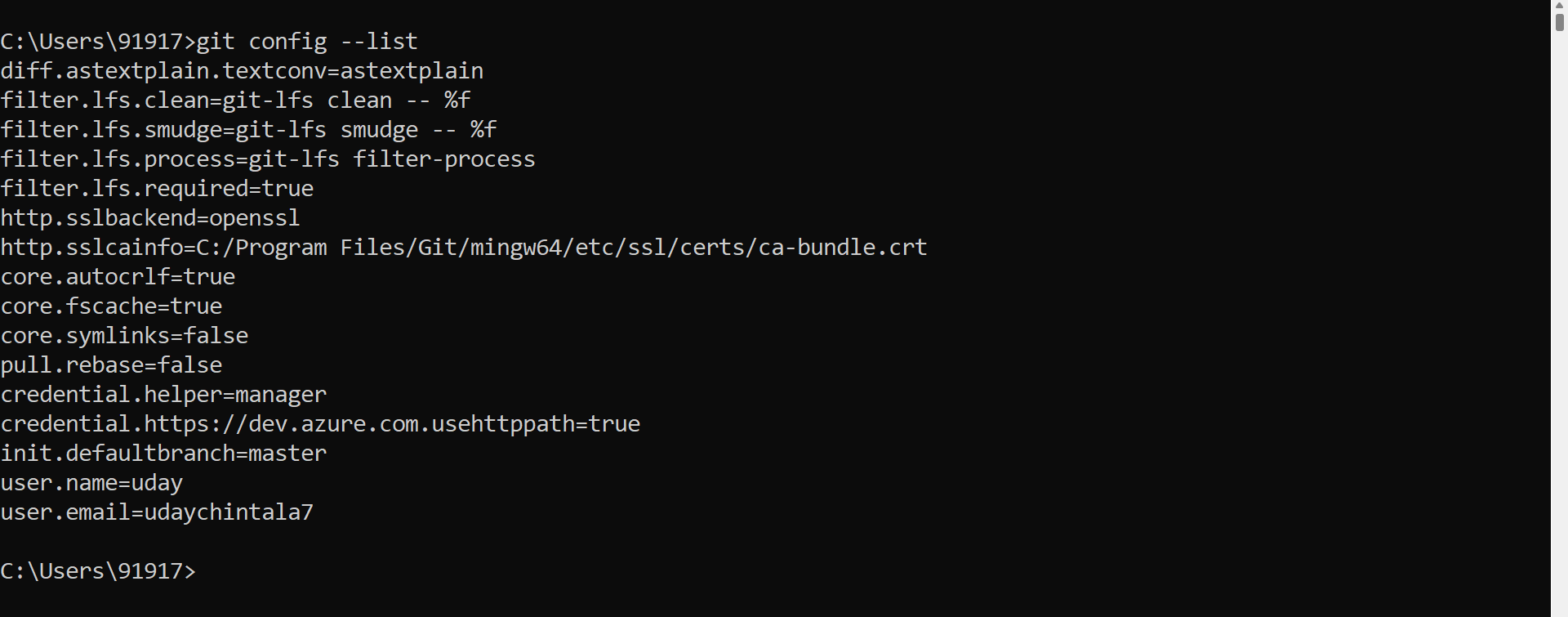
**git config --global user.name "uday"**

**git config --global user.email "udaychintala7"**

Now check global list



**git config –list** will give both System and Global Configurations.

****

**Git Commands: -**

**Git Work Flow**

**Working Directory 🡪 Staging Area 🡪 Local Repository 🡪 Cloud repository**

**git config --system –list = To check System Level Configuration.**

**git config --global --list = To check Global Level Configuration.**

**git config --global user.name "uday" = used to set global user name.**

**git config --global user.email "udaychintala7" = used to set global email id**

**git config –list = will give both System and Global Configurations.**

**git status = will tells you what changes are made in code or project.**

**git init = this is used to initialize git.**

**git add = used to add file from working directory to staging directory. (git add filename)**

**git add . = used to add all files from working directory to staging directory. (git add filename)**

**git commit = used to commit the code changes with some comments.(git commit -m “Commit message”)**

**git show = will show commit details.**

**git show commitid = will show commit details of particular Id.**

**git log = will give you the all log details.**

**git log --oneline = will give you the all log details in single line.**

**git branch = Giving branch name.**

**git branch -M main = used to change branch name to main.**

**git push = used to push our code to Git Repository. (git push origin master) here origin represents Git Repository and master represents local repository**

**git clone = cloning repository into local system.(git clone https://github.com/chintalauday/Employee-Crud.git )**

**git pull = getting changes from Git Repository to local.**

**.gitignore = this is file name here we can mention the file names ti be ignored**

**git checkout committed = going to that committed id**

**git revert commitid = revert code back to previous**

**git revert HEAD = revert code back to previous one commit**

**git revert HEAD-2 = revert code back to previous 2 commit**

**git reset = it will modify commit history (git reset commit id)**

**git reset HEAD-number = will modify commit history up to that number**

**git reset –soft commit id = will delete commit history and not code and code moved to staging area**

**git reset –hard commit id = will delete commit history and code also**

**git branch = to see branch name.**

**git branch <branch name> = to create new branch. (git branch dev)**

**git checkout <branch name> = if new branch created, we want t do changes in new branch, then we have to move to that branch for that this is used**

**git checkout dev git checkout -b dev = it will create new branch dev and moving to dev branch from current branch**

**git merge dev = merging dev branch to current branch.**

**git rebase**

**git fork**

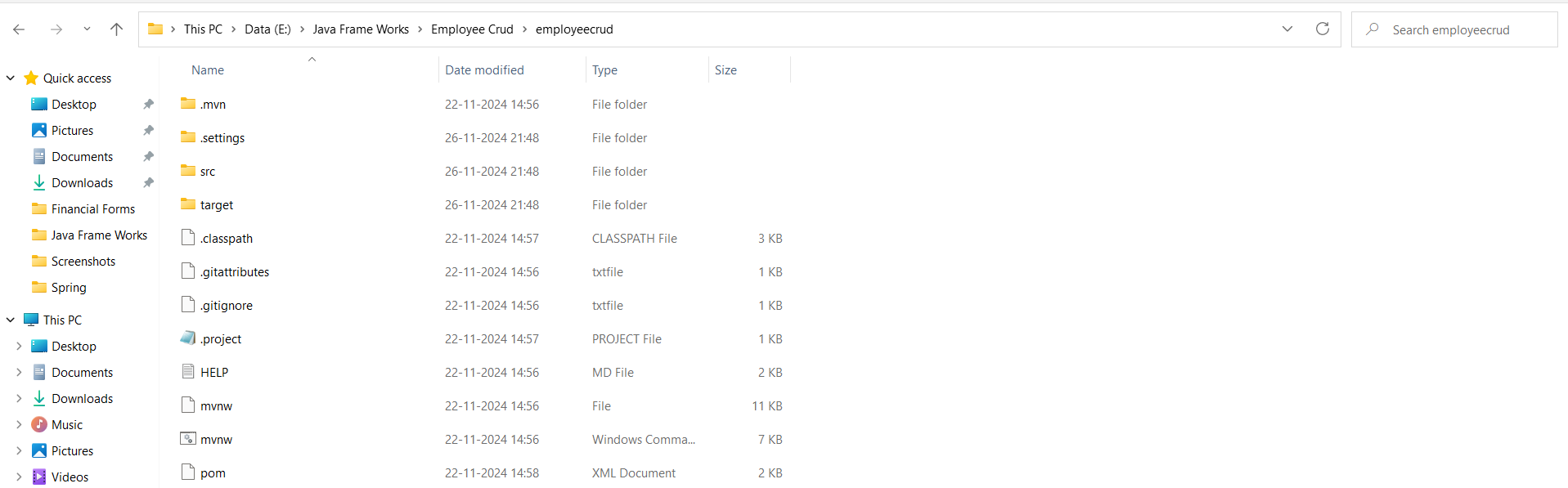
**git stash**

**git cherry pick**

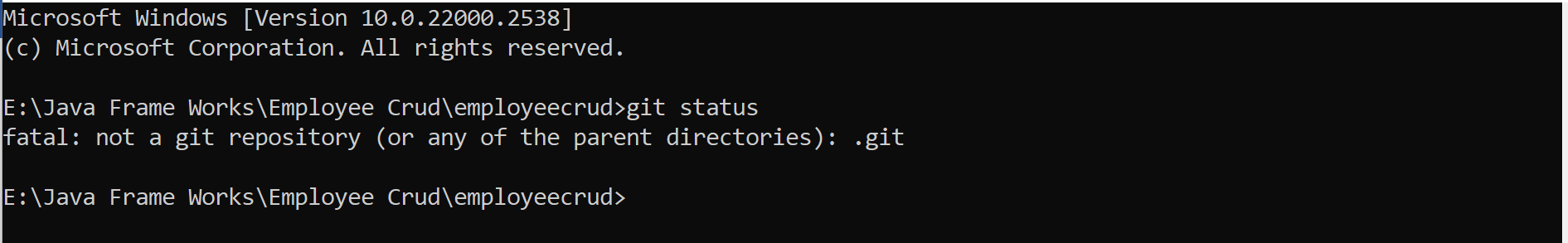
**git status** = will tells you what changes are made in code or project.

1.go to folder you want to add it to Git.

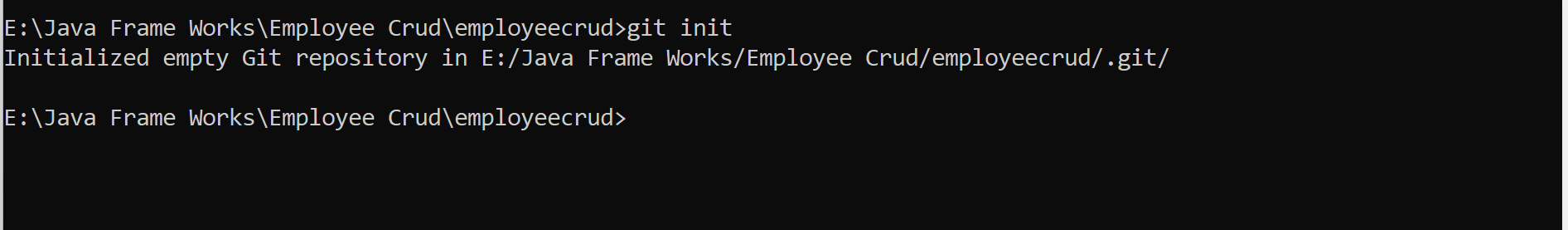
Here I wants to move this Spring boot folder into Git.



Open cmd in this folder location and type **git status** it is telling that git not initiated in this folder so do as follows

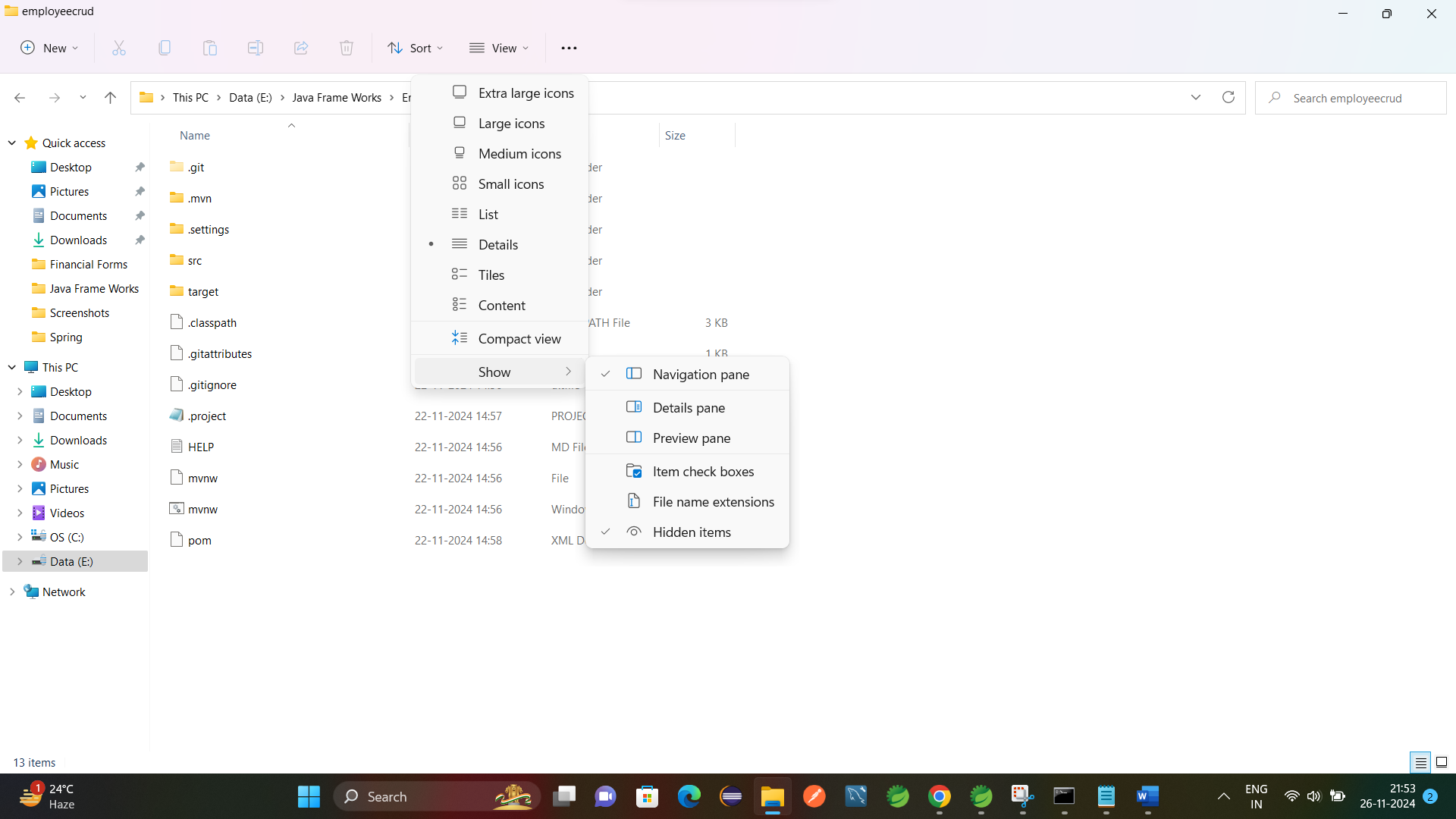


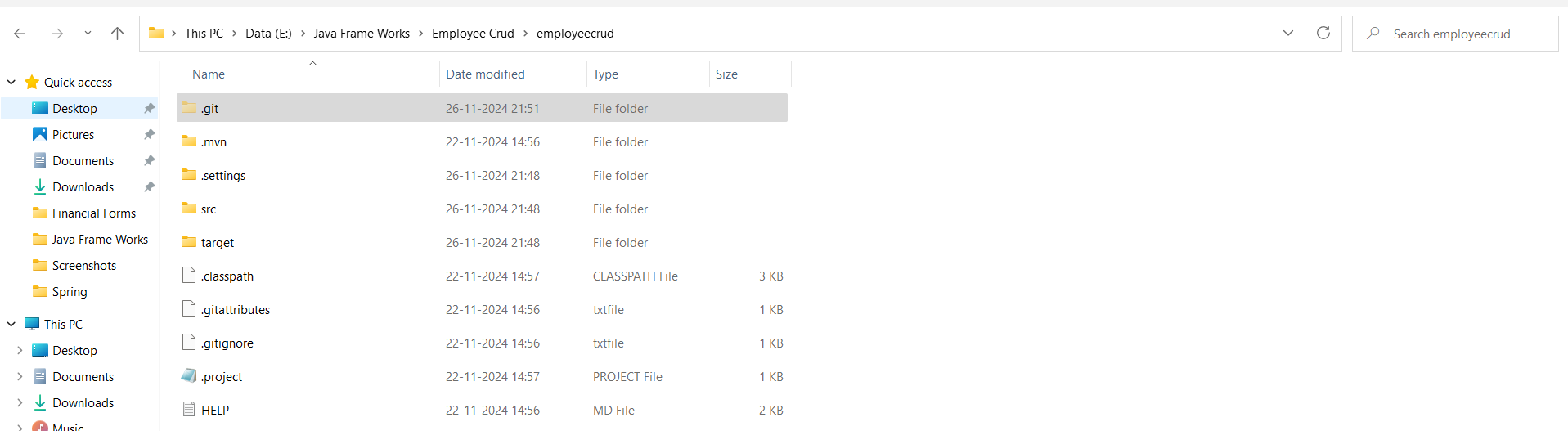
**2.git init** = this is used to initialize git



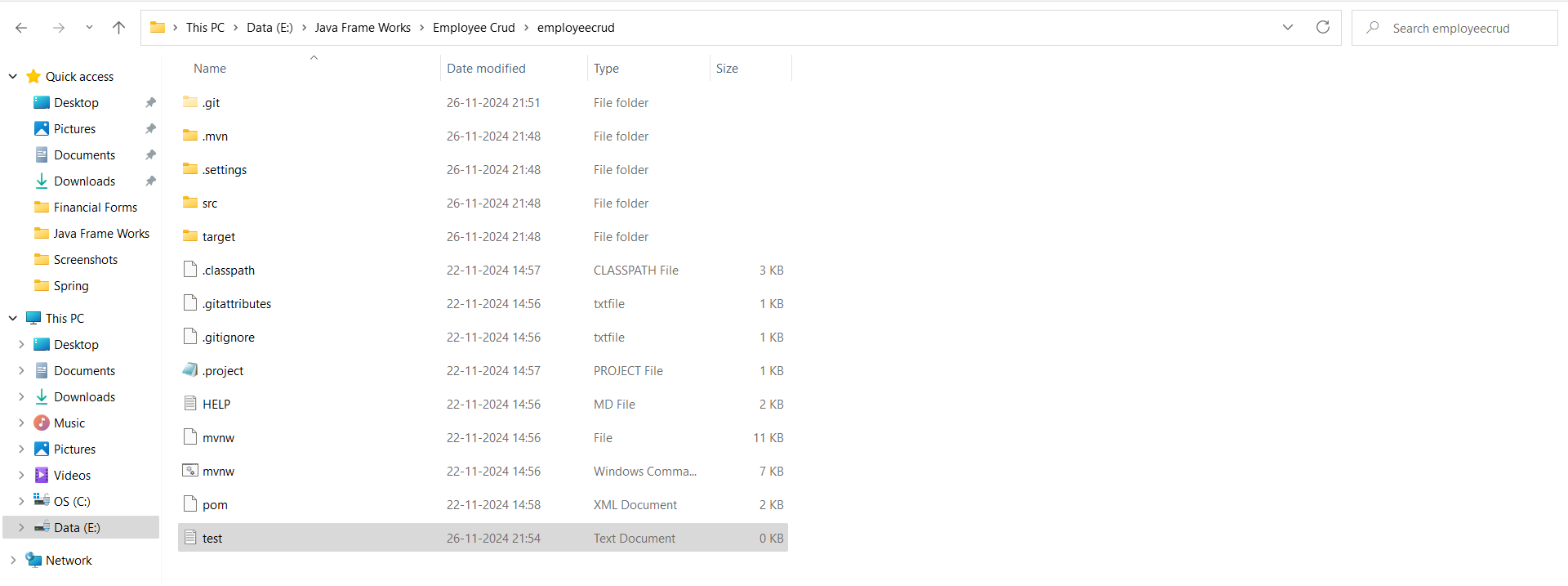
After it will generate .git folder by default its disabled to view so we need to enable that.

View 🡪 Show 🡪 Hidden items

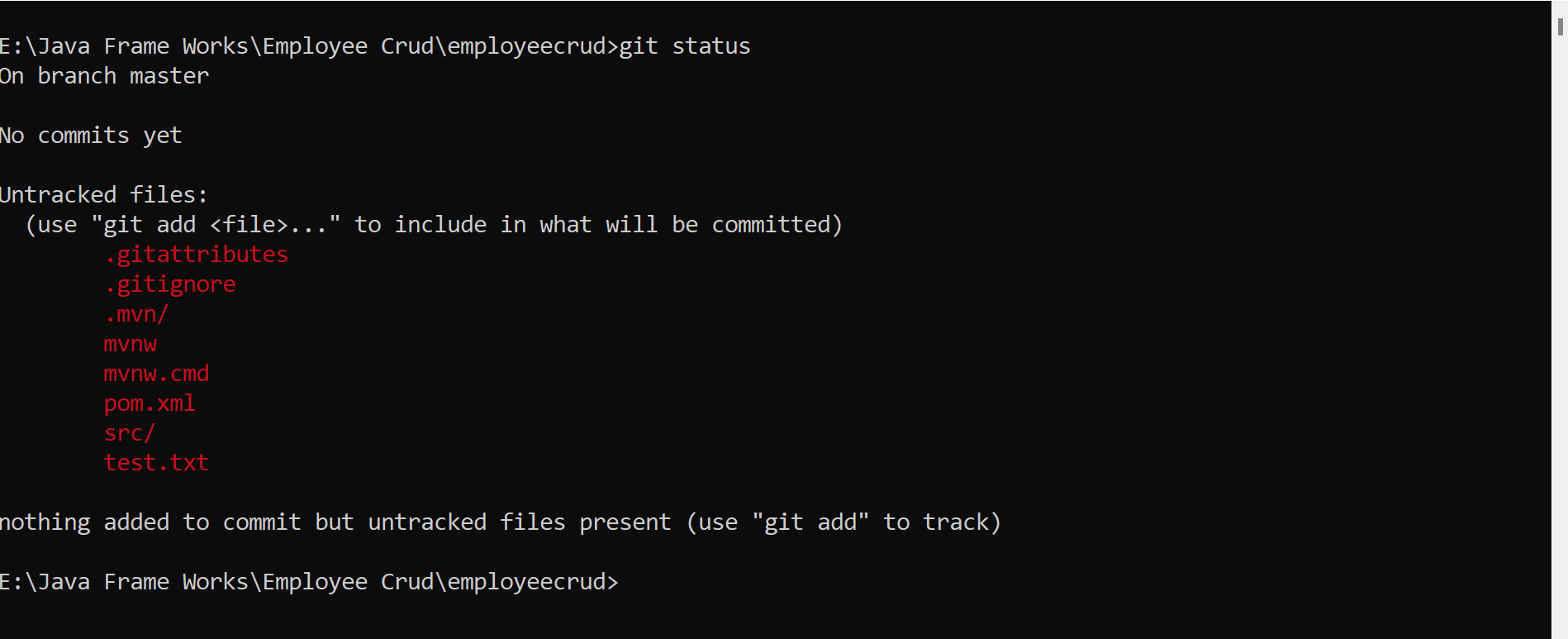




Now added one new file into that folder



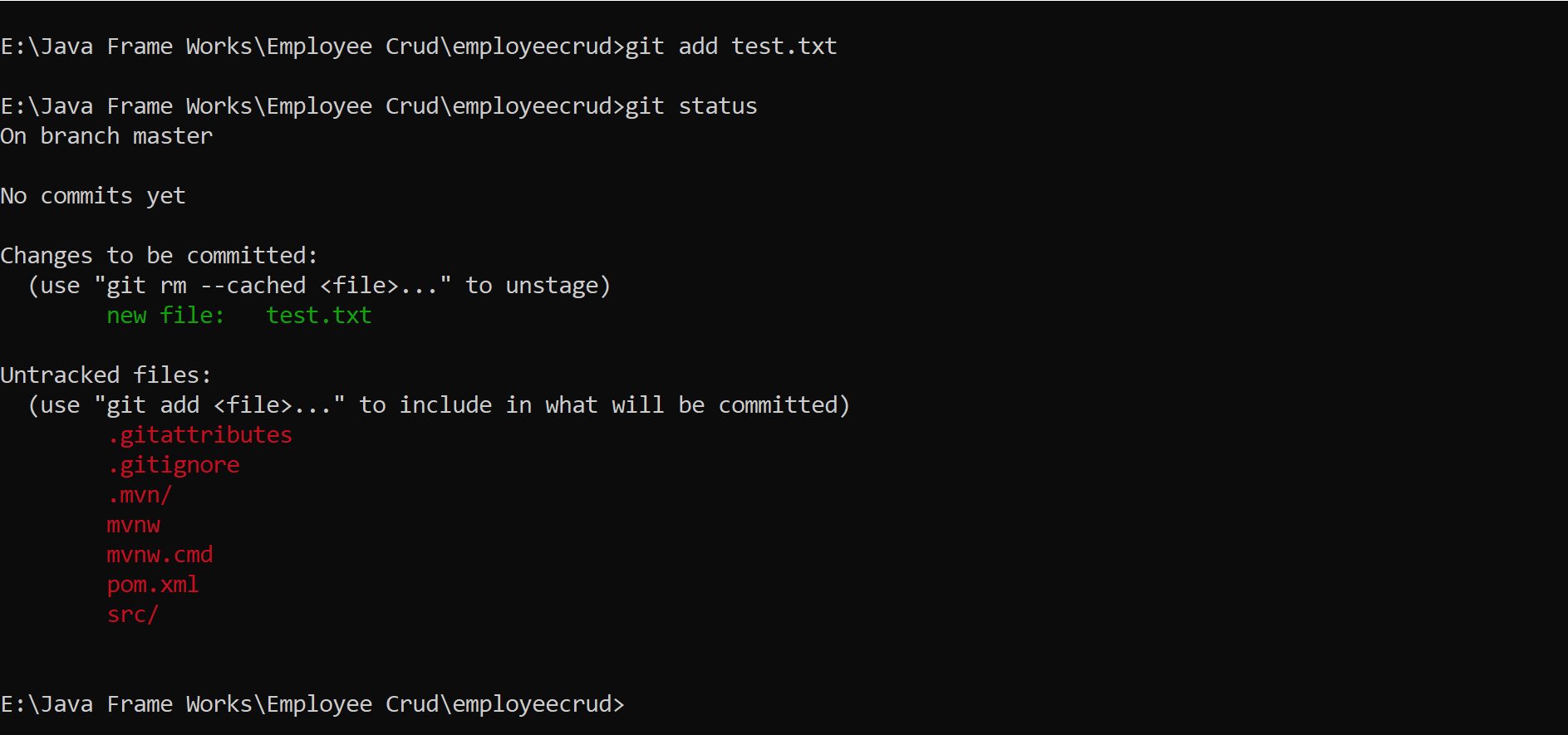
Now run **git status**



It is telling that new file created but not committed. This is because file not moved to staging area from working directory for this will use **git add** test.txt

**git add =** used to add file from working directory to staging directory.

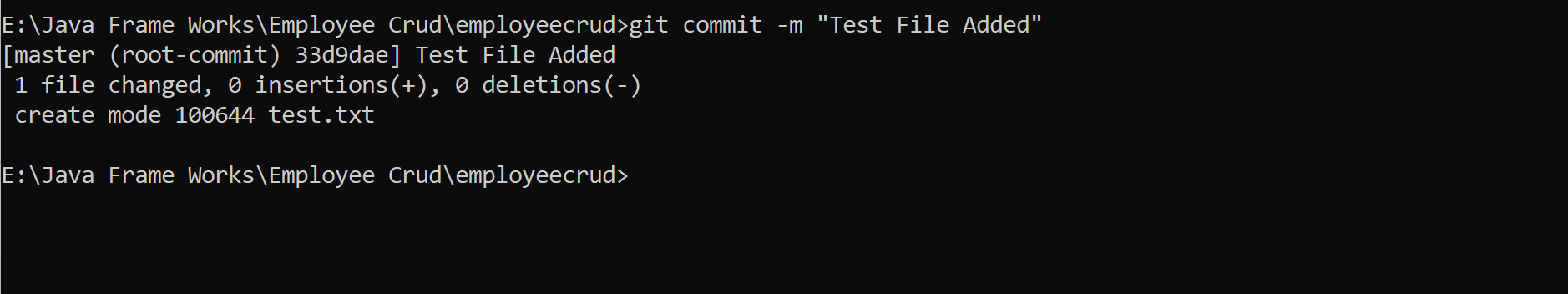
And check **git status** it is telling that one file not committed.



After file moved to staging area, we need to commit

**git commit -m “Test File Added”**

**git commit =** used to commit the code changes with some comments.

****

It will tell what changes we made.

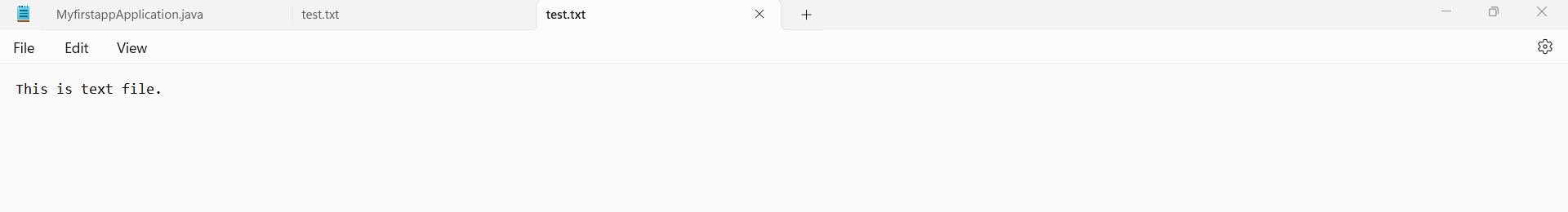
Now check **git status** if any files not added to git add them by using **git add .**

**git add filename =** adding single file from working repository to staging repository.

**git add . =** adding all files to staging repository from working repository.

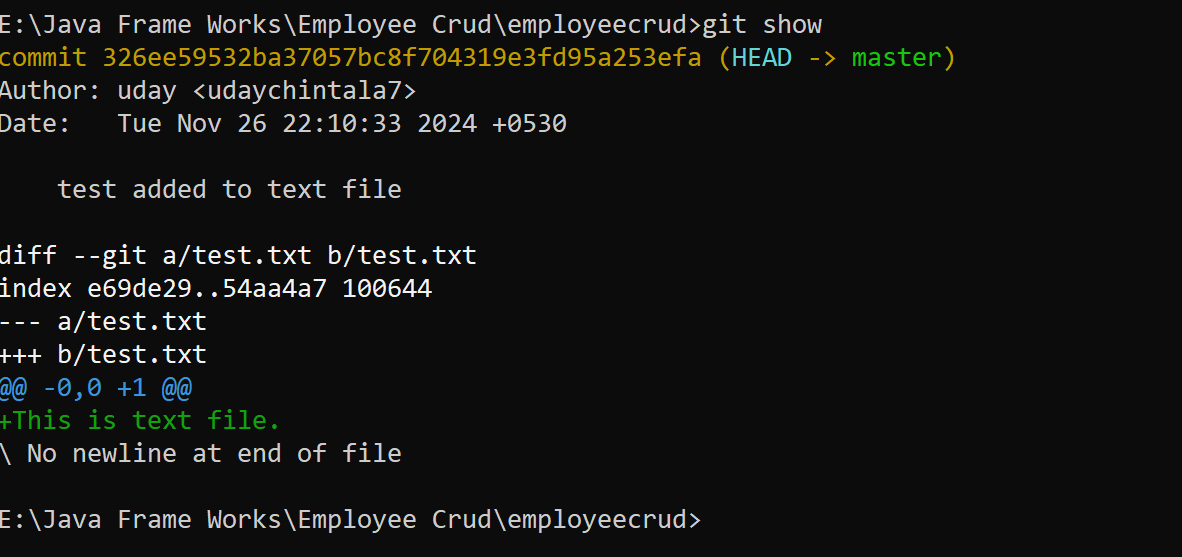
After adding all files give commit **git commit -m “message”**

Now add text in test file and do steps as follows.



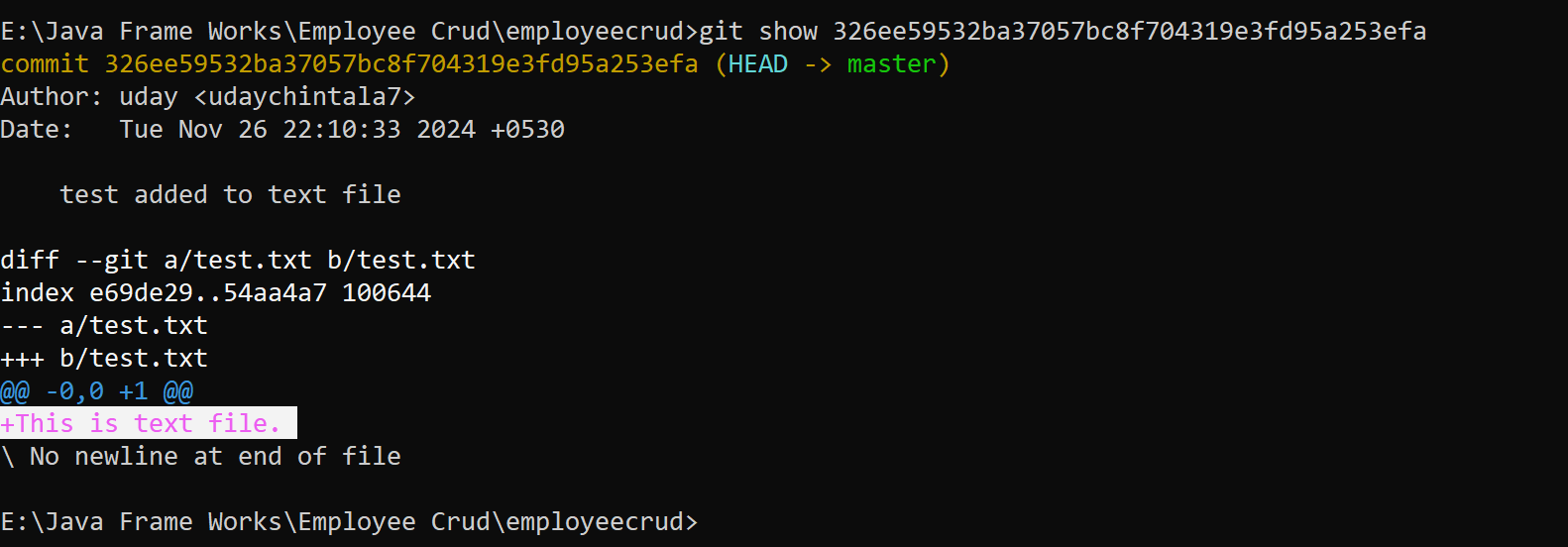


**git show =** will show commit details

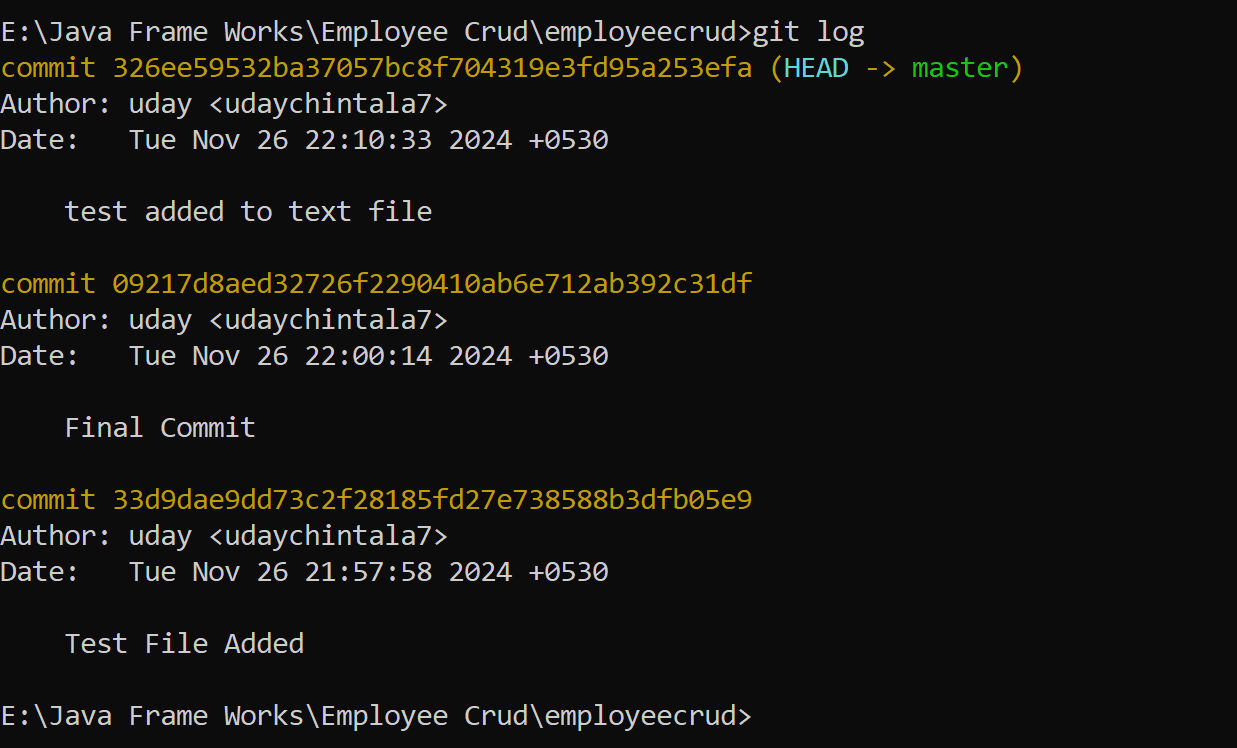


**Git show** with commit id will give you changes made in that commit.

Example **git show 326ee59532ba37057bc8f704319e3fd95a253efa**

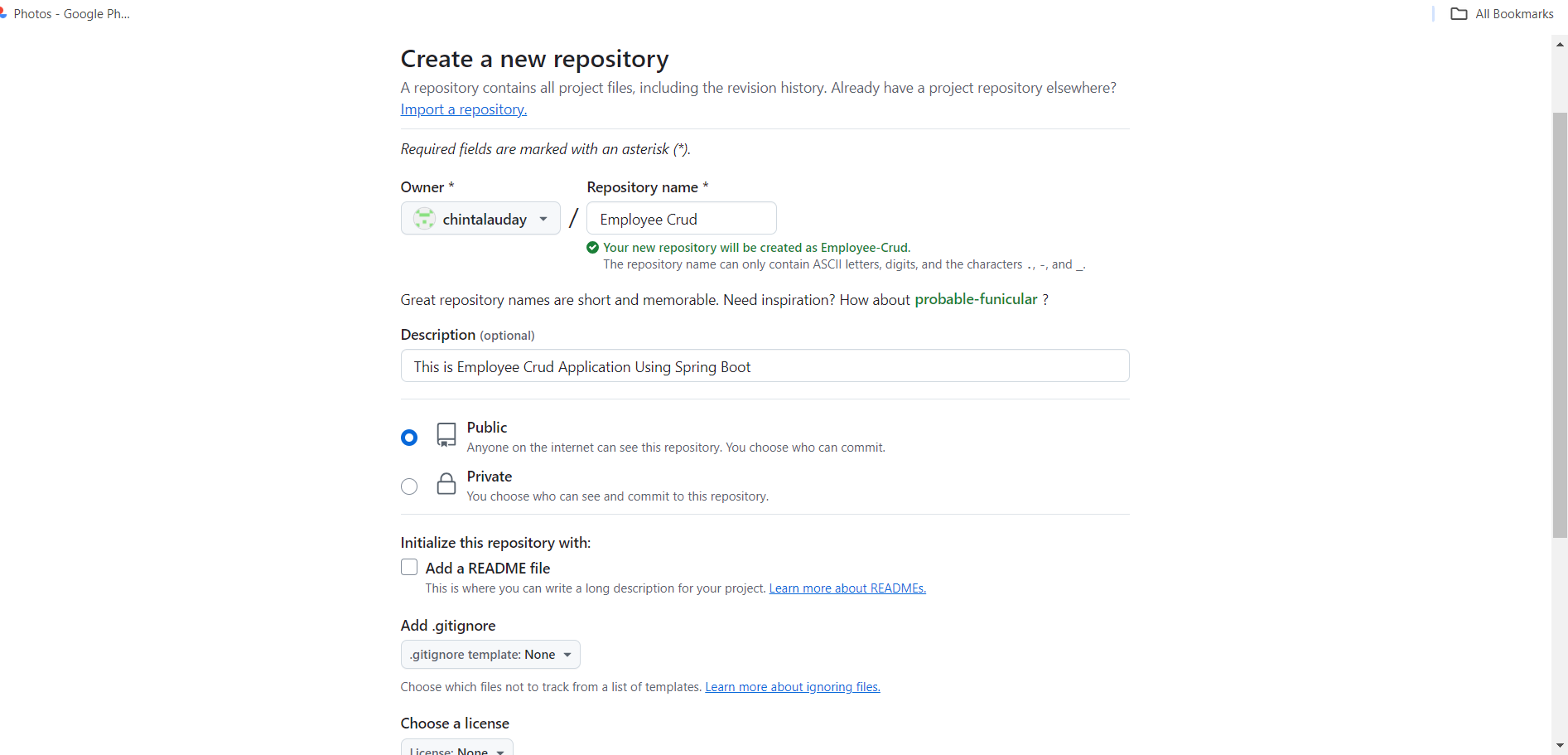


**git log** = will give you log details.

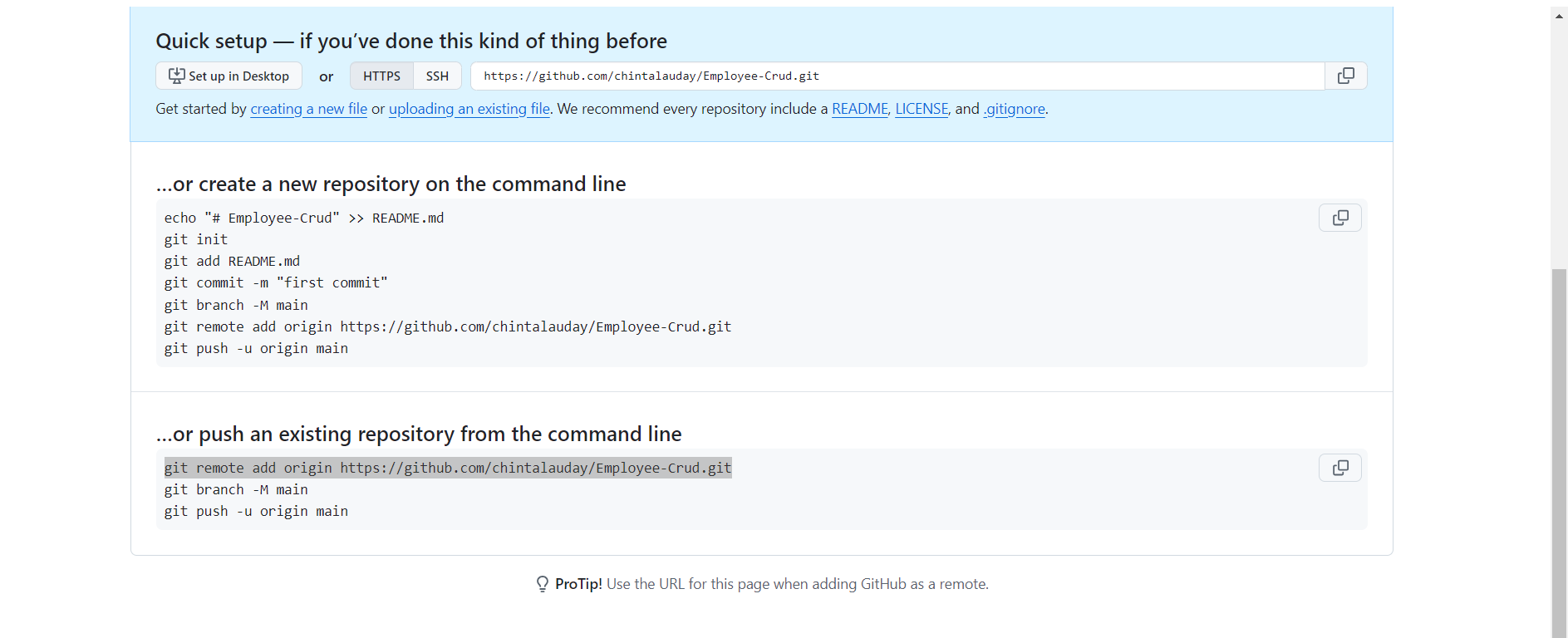


**Uploading Project into Git Hub from Local System**

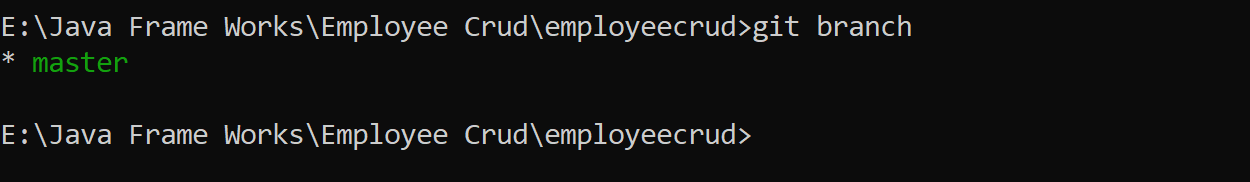
Open Git Hub and create Repository



Take highlighted commands from repository



**git branch** = Giving branch name.



Here in our local system the branch is master but in Git Hub repository branch is main

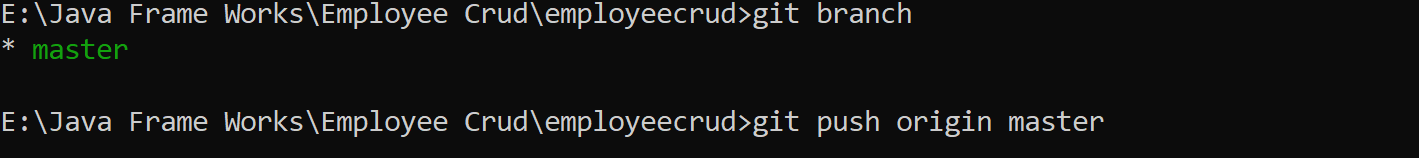
So, if we want then will rename our local branch name to main. (Sample command = **git branch -M main)**

**git branch -M main** = used to change branch name to main.

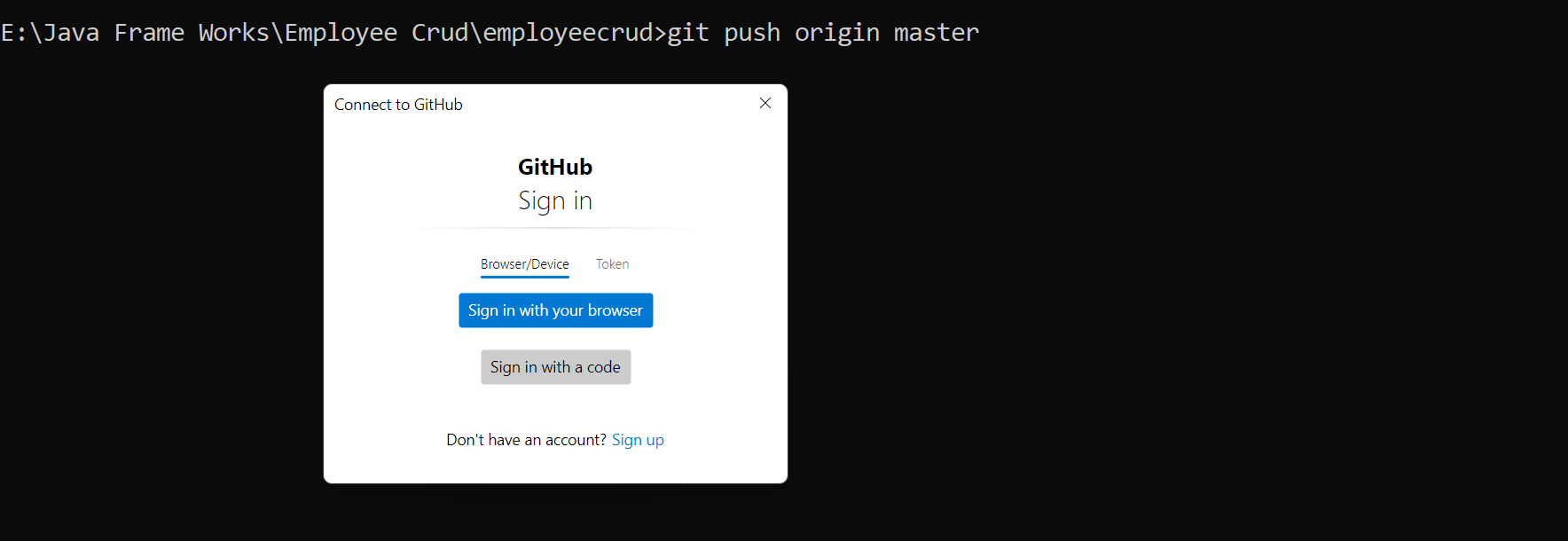
But here we are not changing our branch name in local system if we are changing local branch then we have to use **git push -u origin main** command to push code to Git Hub otherwise will use **git push -u origin master.**

**git push =** used to push our code to Git Repository. (**git push origin master**)

here origin represents Git Repository and master represents local repository.

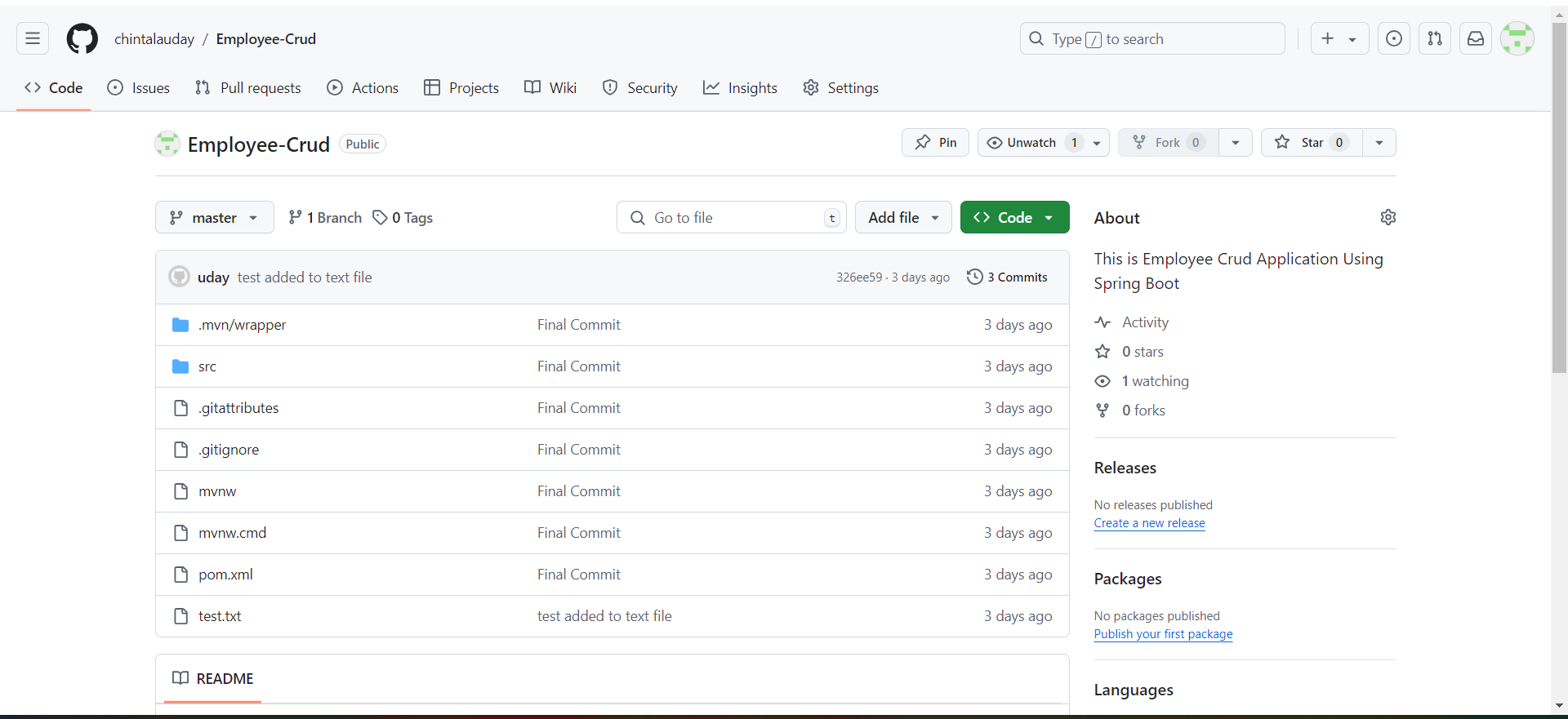


This will opens Authentication page



Here sign in to Git Hub account.

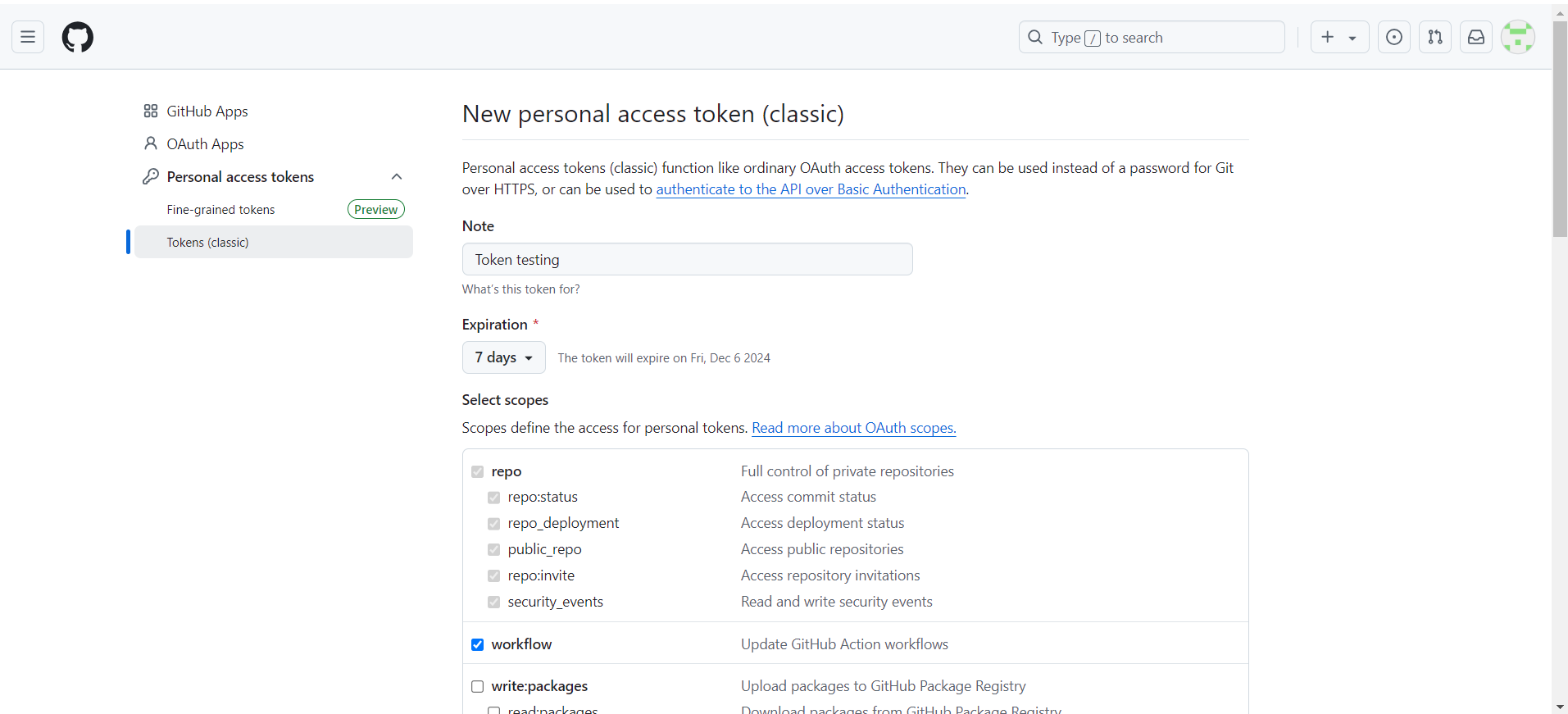
Now refresh Git Hub there your project is added.



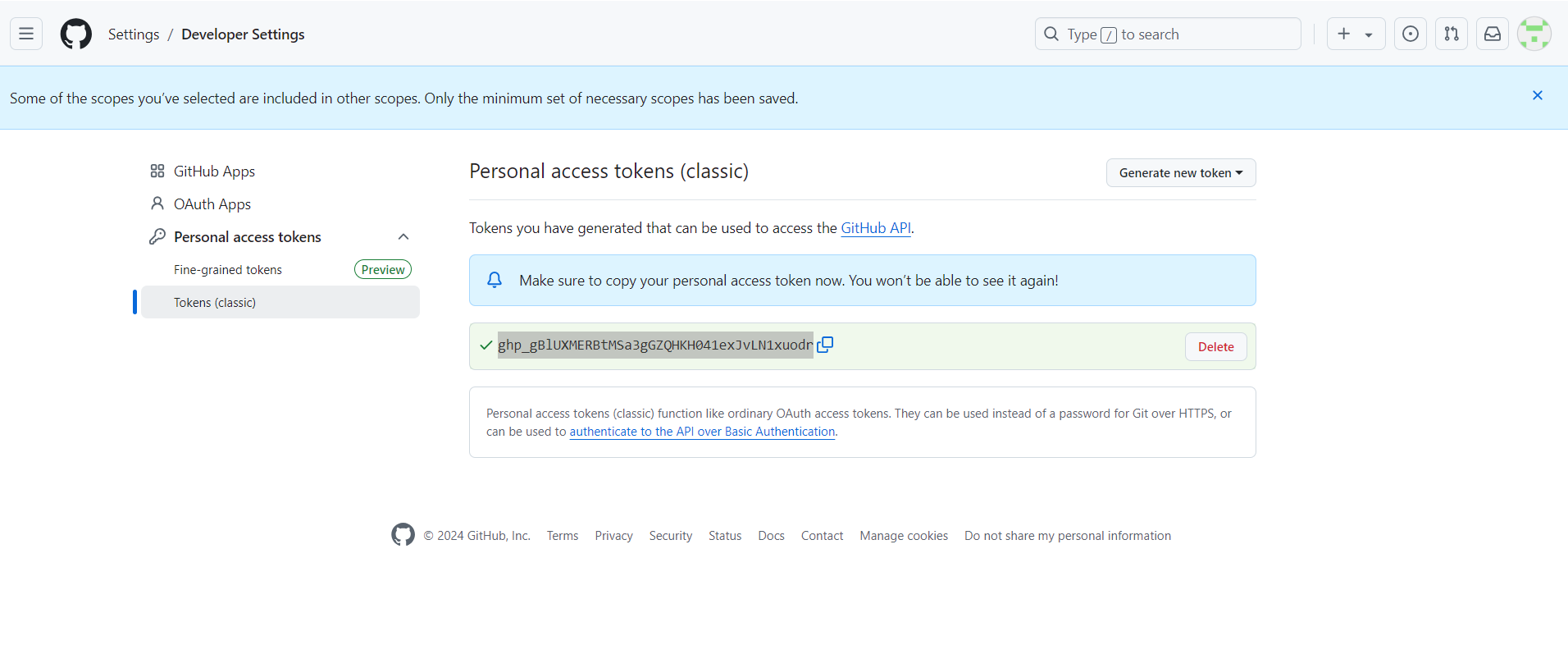
**Creating Access Token**

This is for Authenticate purpose to give access to modify Git Hub Repository.

Open Git Hub Account and go as follows **profile 🡪settings 🡪developer settings 🡪personal access tokens 🡪Tokens(classic) 🡪Generate Token** then following screen will display.

****

After submitting token will generate.

****

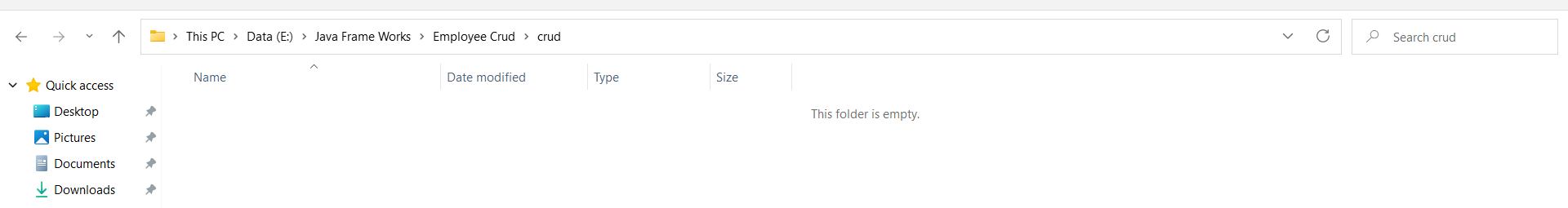
Will share this token to person to whom you want to give access to push code into Git Hub Repository.

**Cloning Repository**

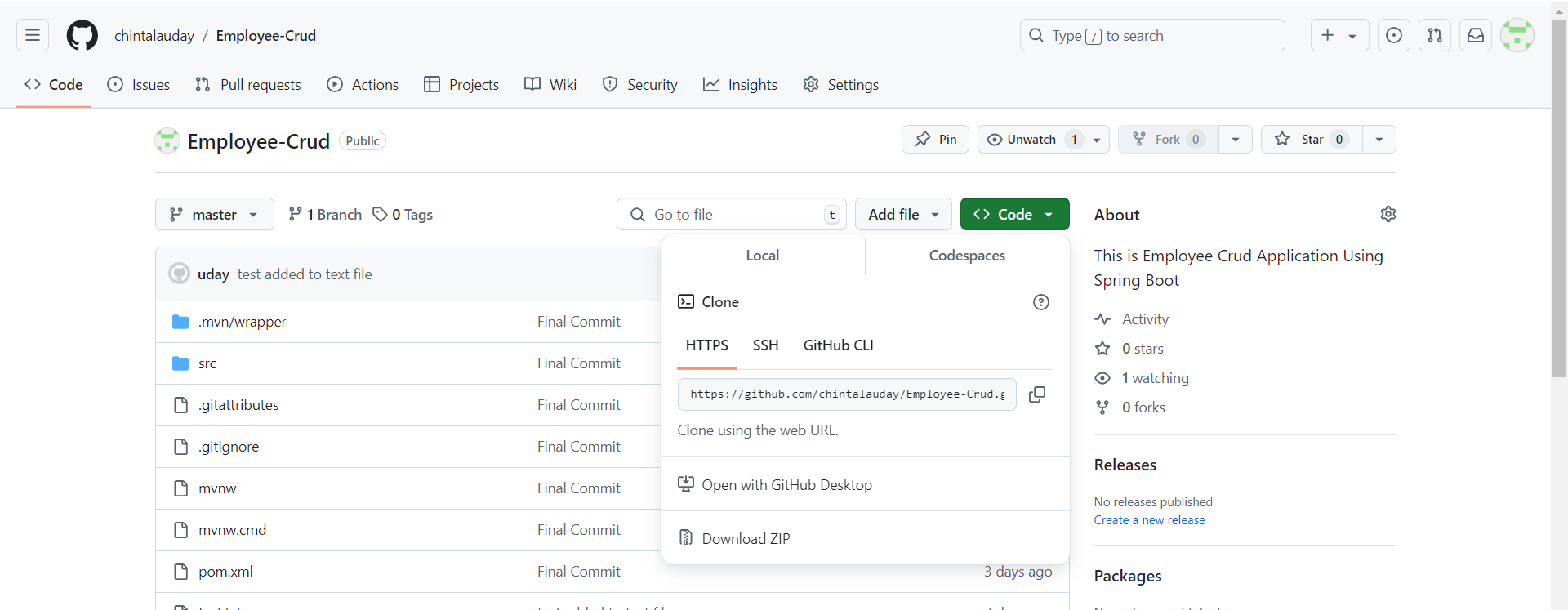
**git clone =** cloning repository into local system.

First create one folder in local system to clone repository into local system.

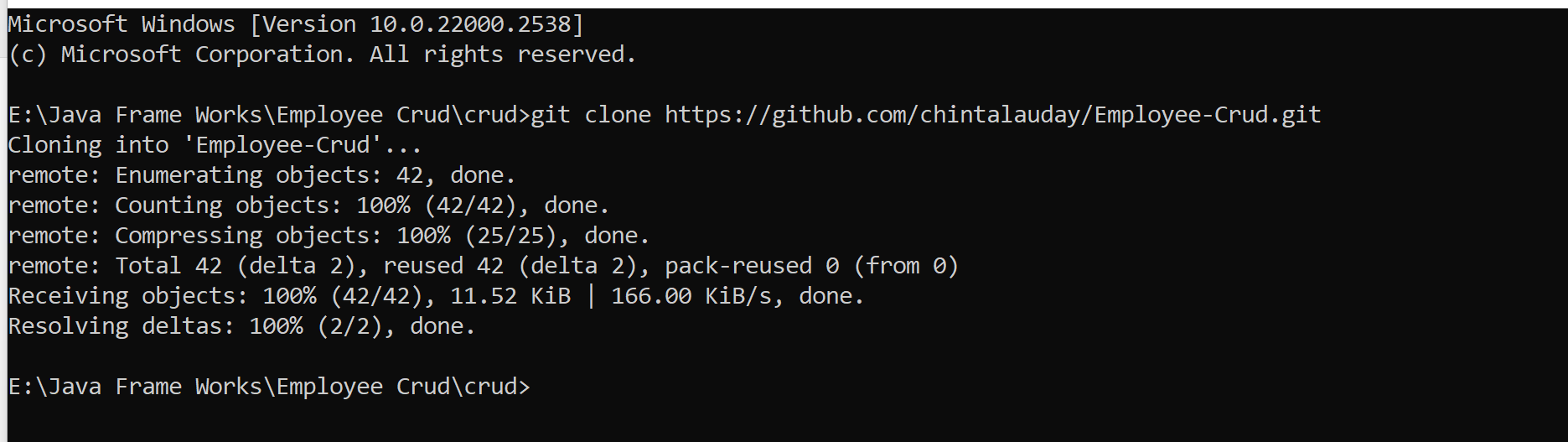
Here I created one folder name crud

****

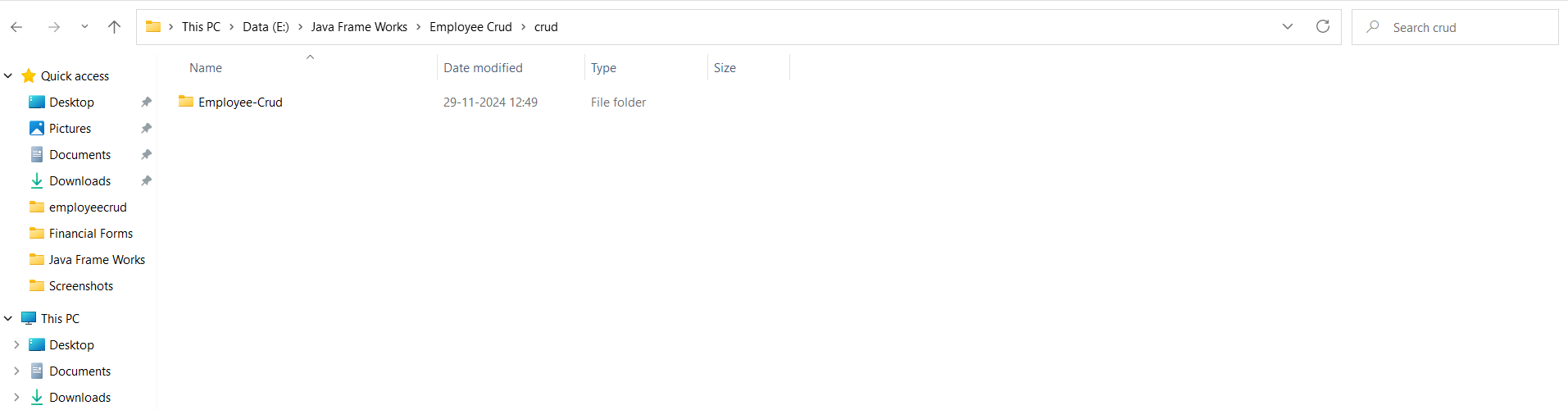
First open Git Hub Repository and go to **Code** and copy that **URL**

****

Run command like **git clone** [**https://github.com/chintalauday/Employee-Crud.git**](https://github.com/chintalauday/Employee-Crud.git)in that folder path



Now check code is cloned into local folder.



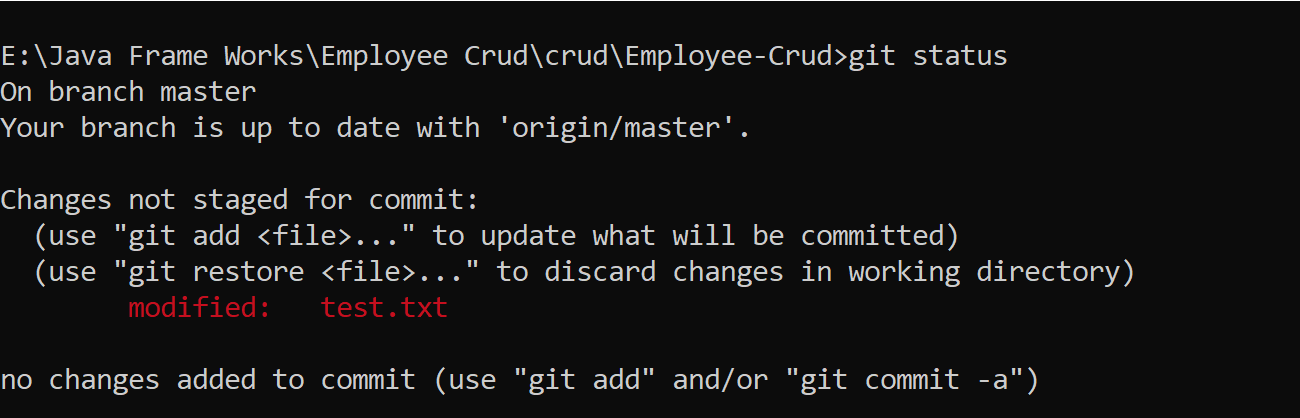
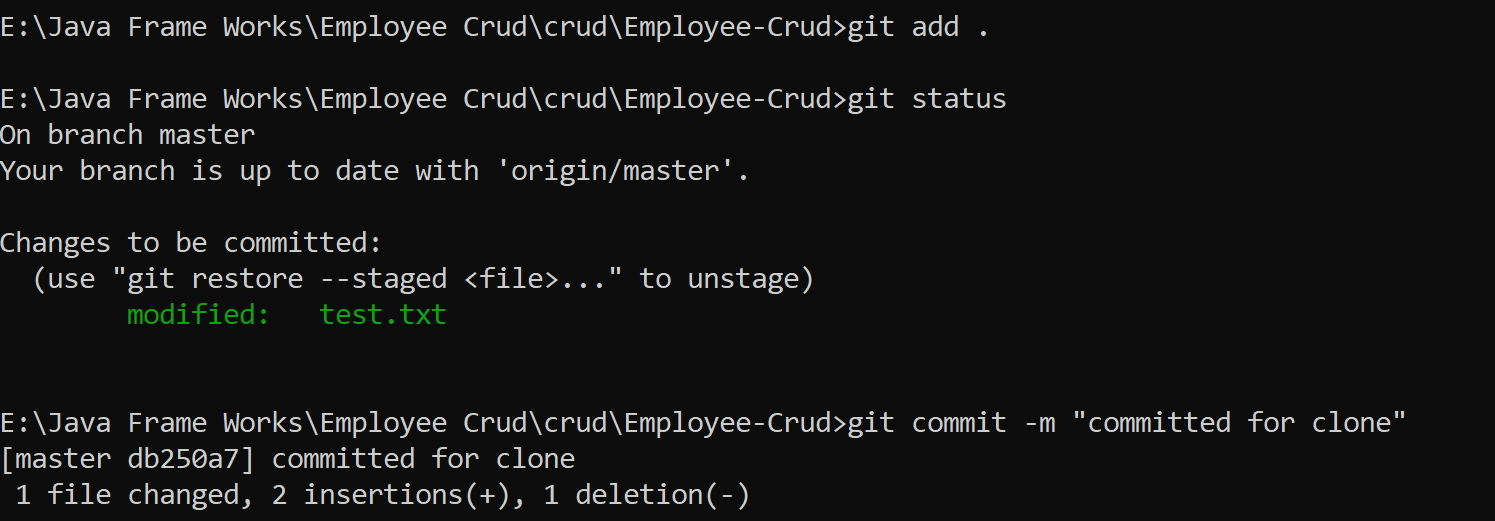
Now to project folder in command prompt and run **git log** it will show you the log.



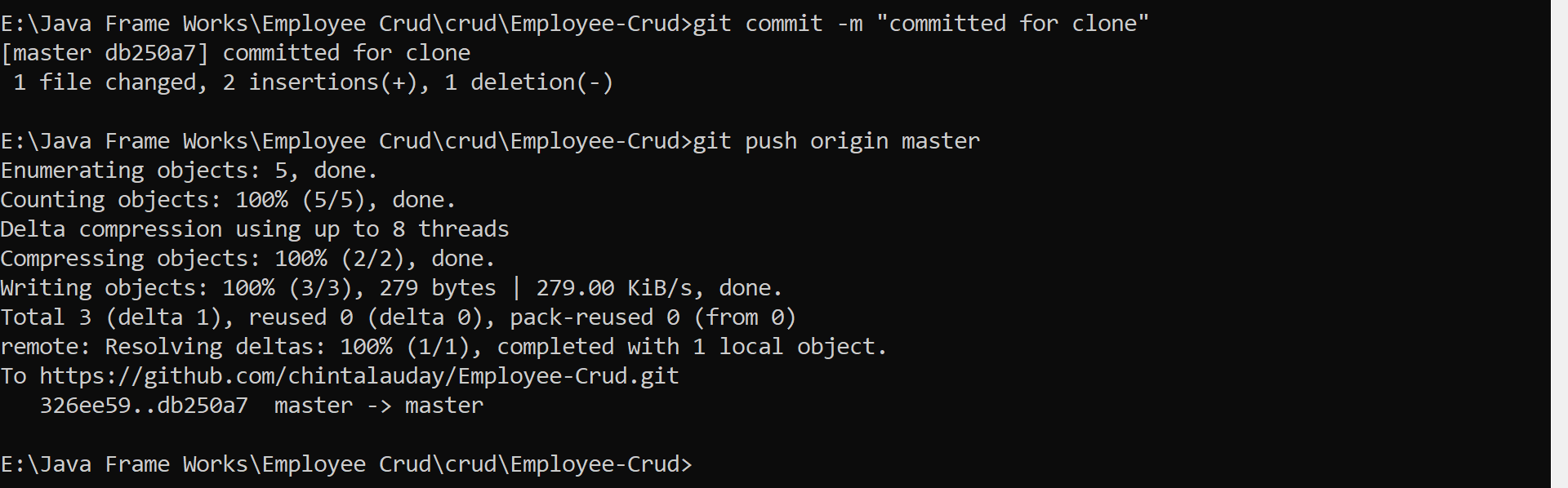
Now try to do modifications in local project and again push that code in to Git Hub.

For that in folder one test.txt file is there I am using that file

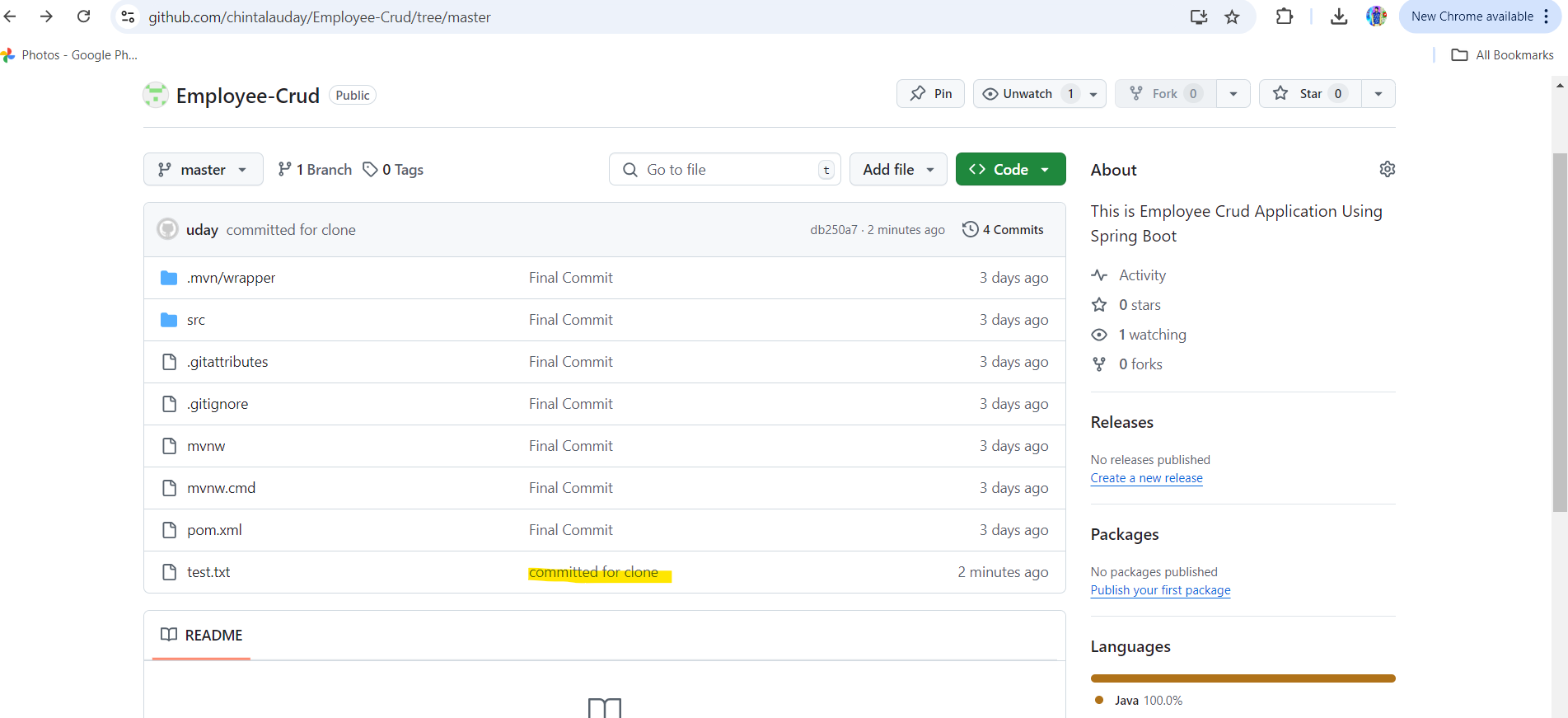
I am making changes in that file and pushing that in to git repo.

Now push code to Git Repository by running **git push origin master**.

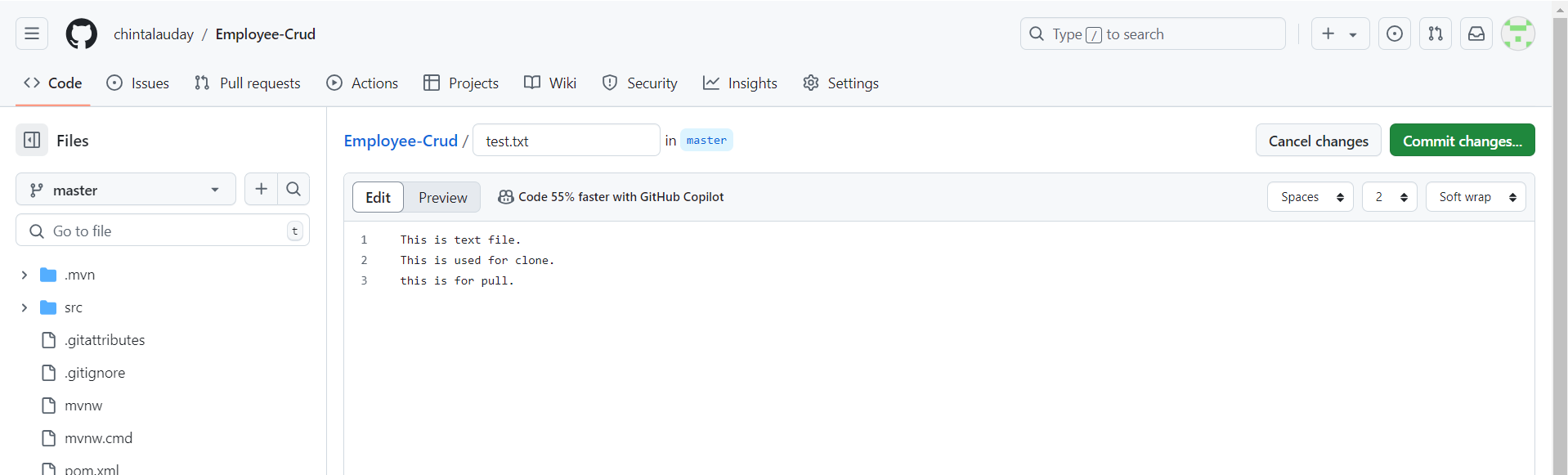


Now our code is pushed to Git Hub.



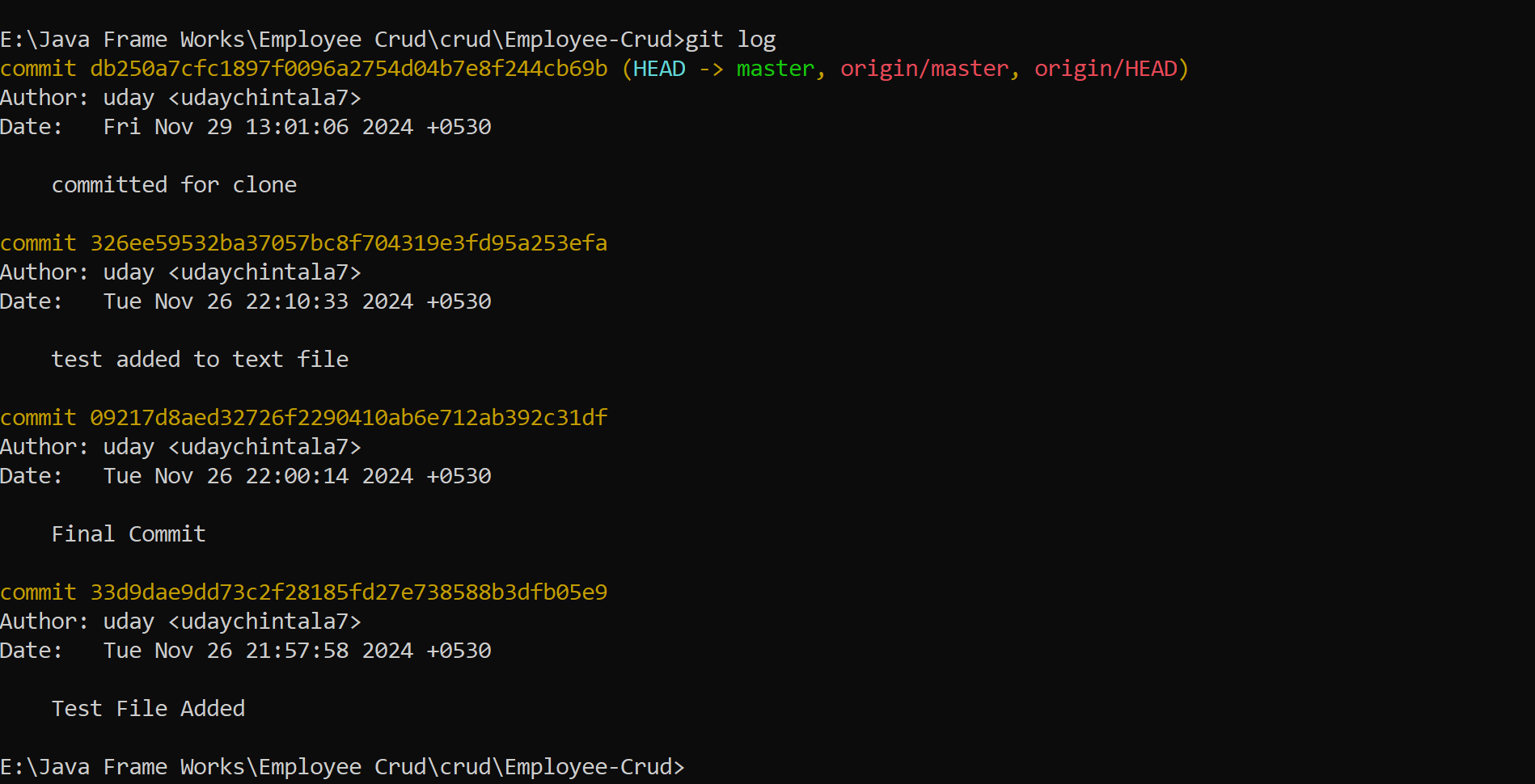
**git pull** = getting changes from Git Repository to local.

For this I made changes in test.txt in Git Repository as follows.

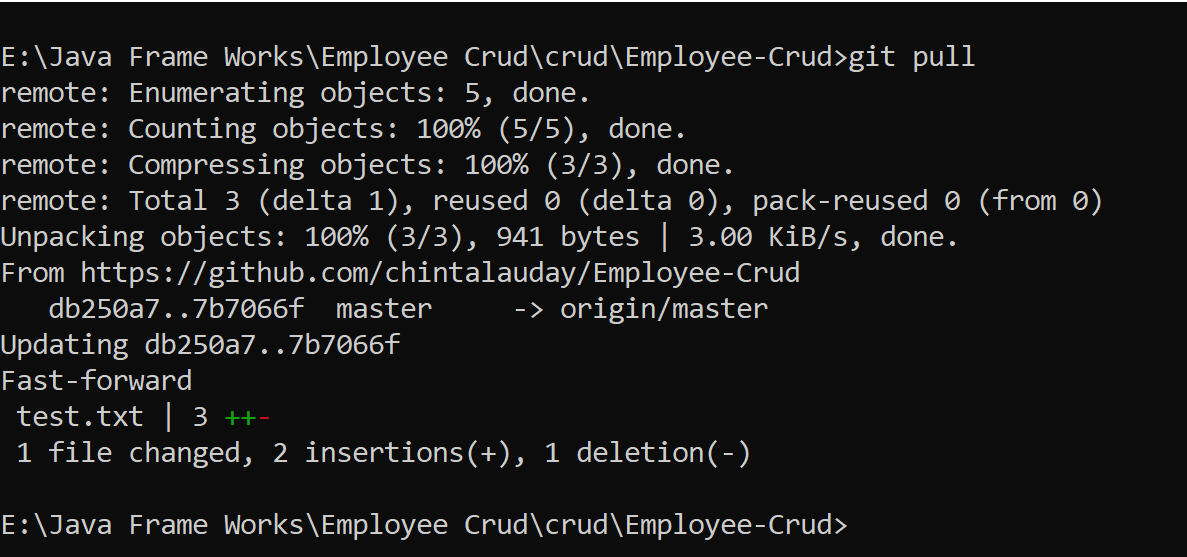


Then click on commit changes with commit message.

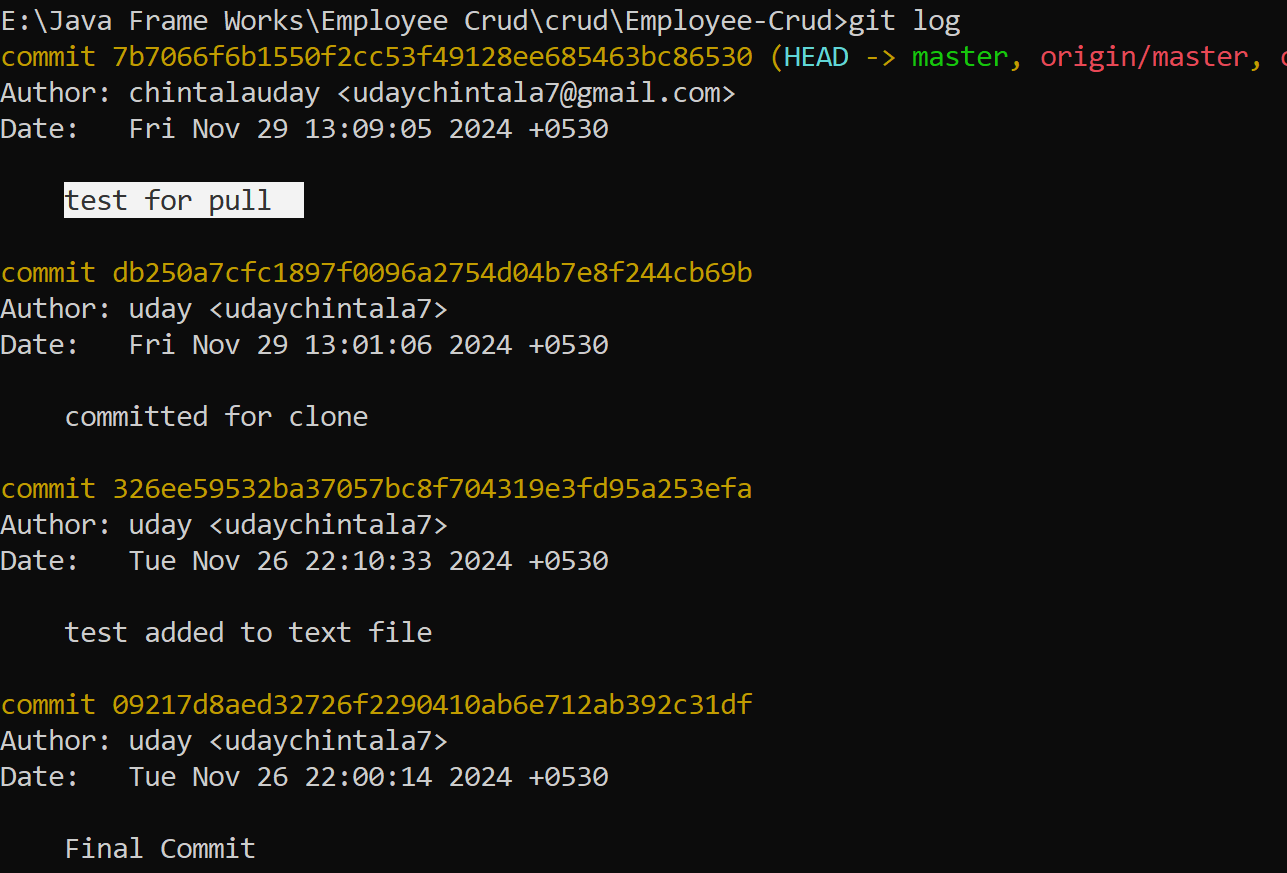
Now run **git log** in local you are not find the code change in Git Repository.



Now we have to use **git pull**



Now check log it will gives you all changes.

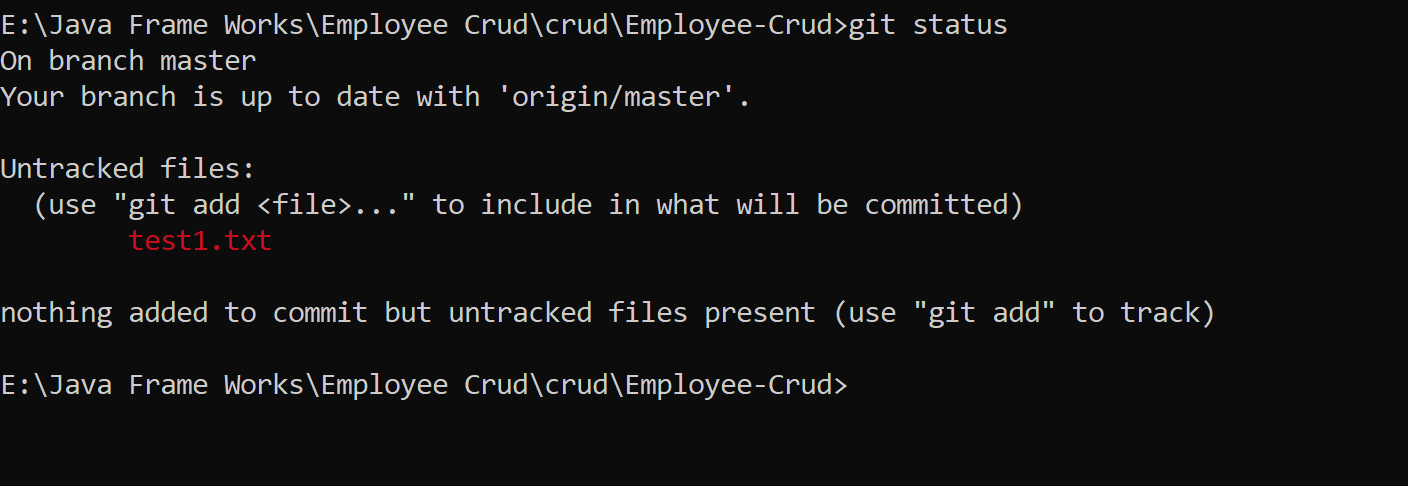


**git ignore** = ignoring files from local to Git Repo.

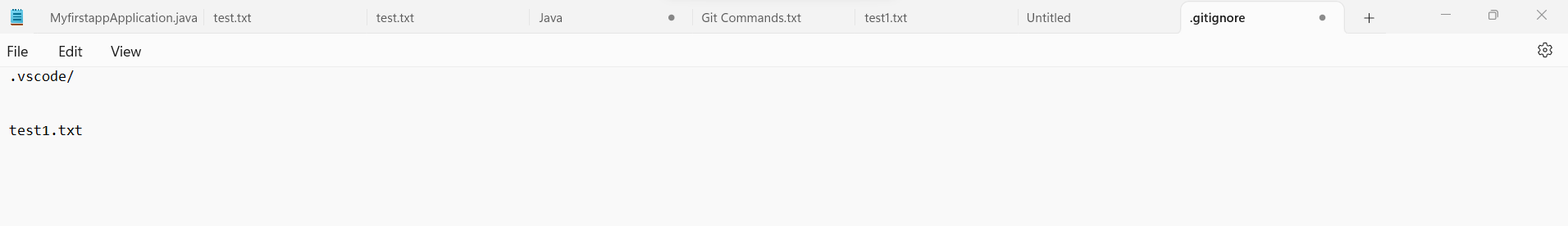
This is file name .**gitignore**

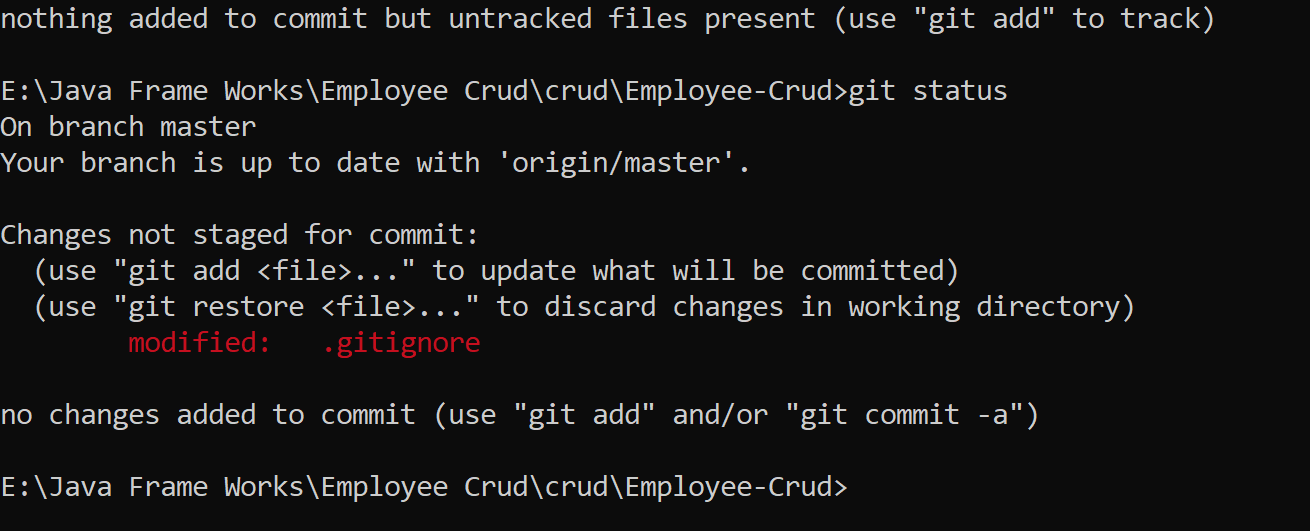
Try to create new file in folder test1.txt

Check git status



Now create one .gitignore file in our folder and add that ignored file name here and run git status



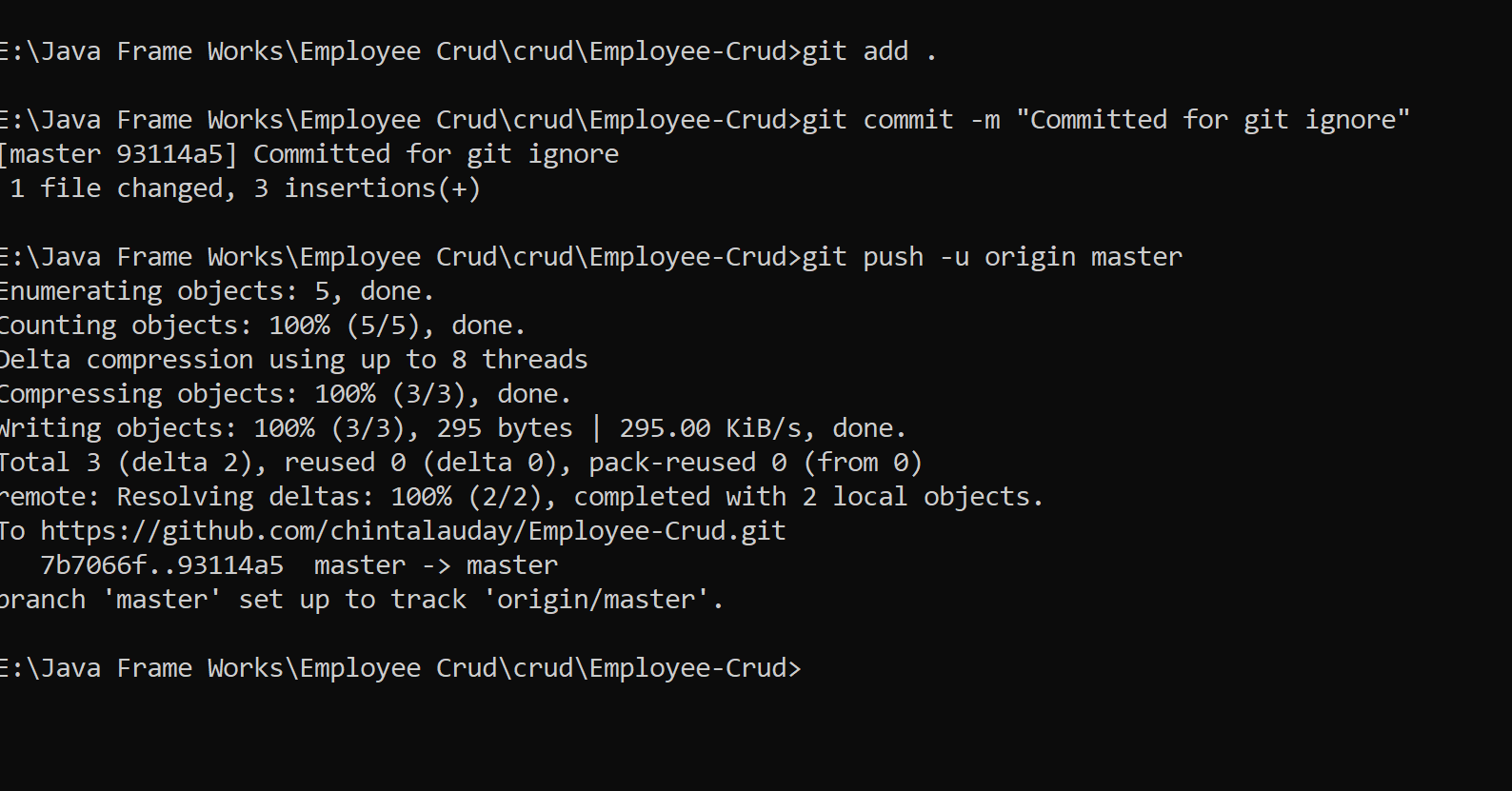


Now push code to Git Repository

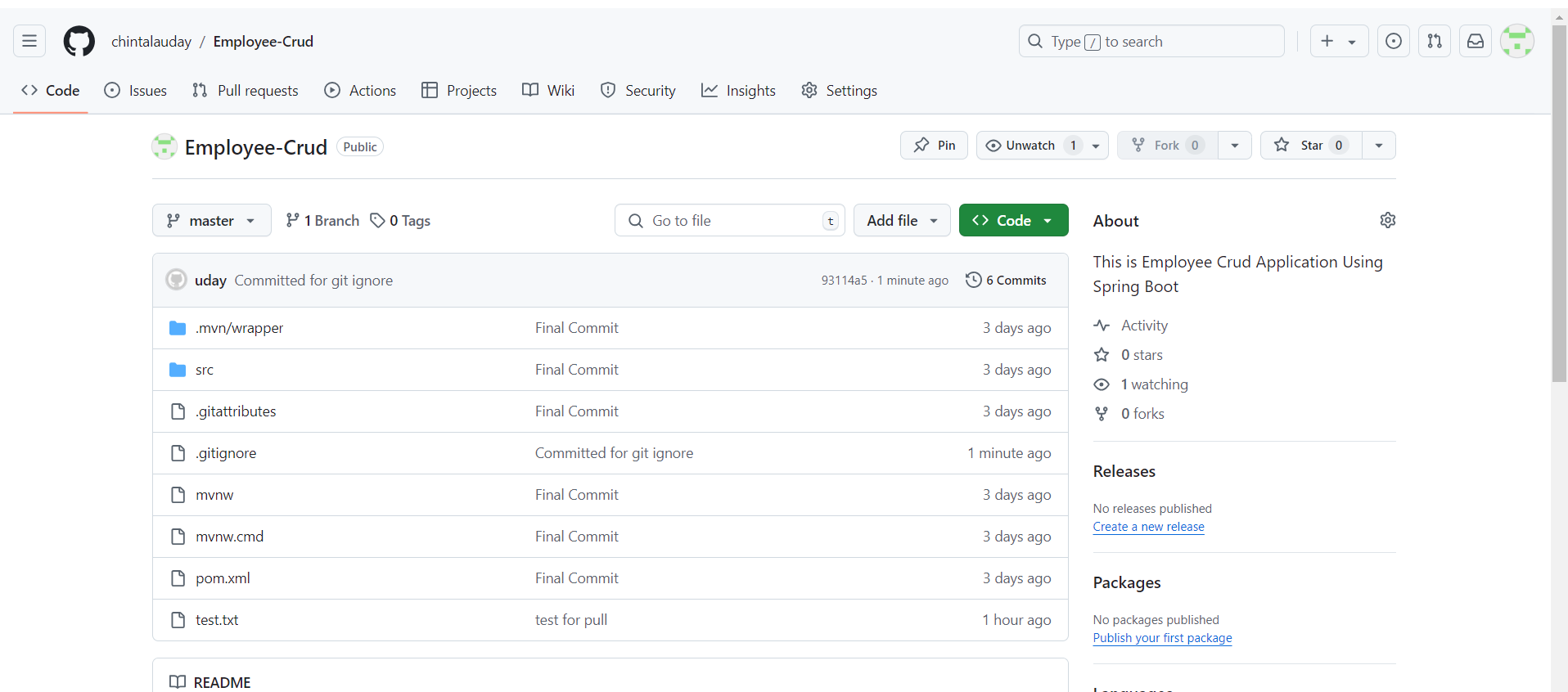
**\*.txt = ignores all text files**

**/folder name = ignores folder from master**

**folder name/ =ignores all folder with same name under master**

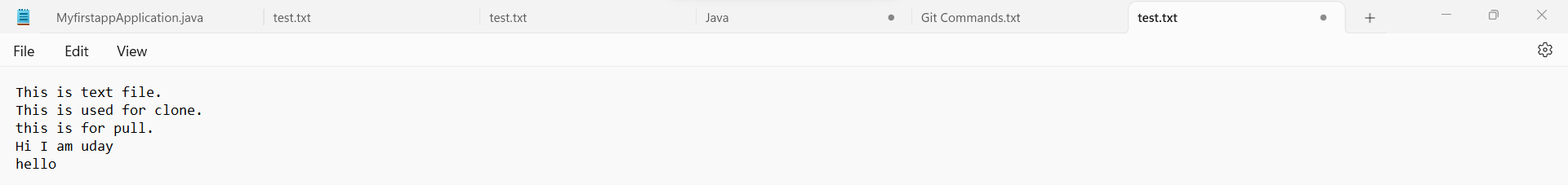
****

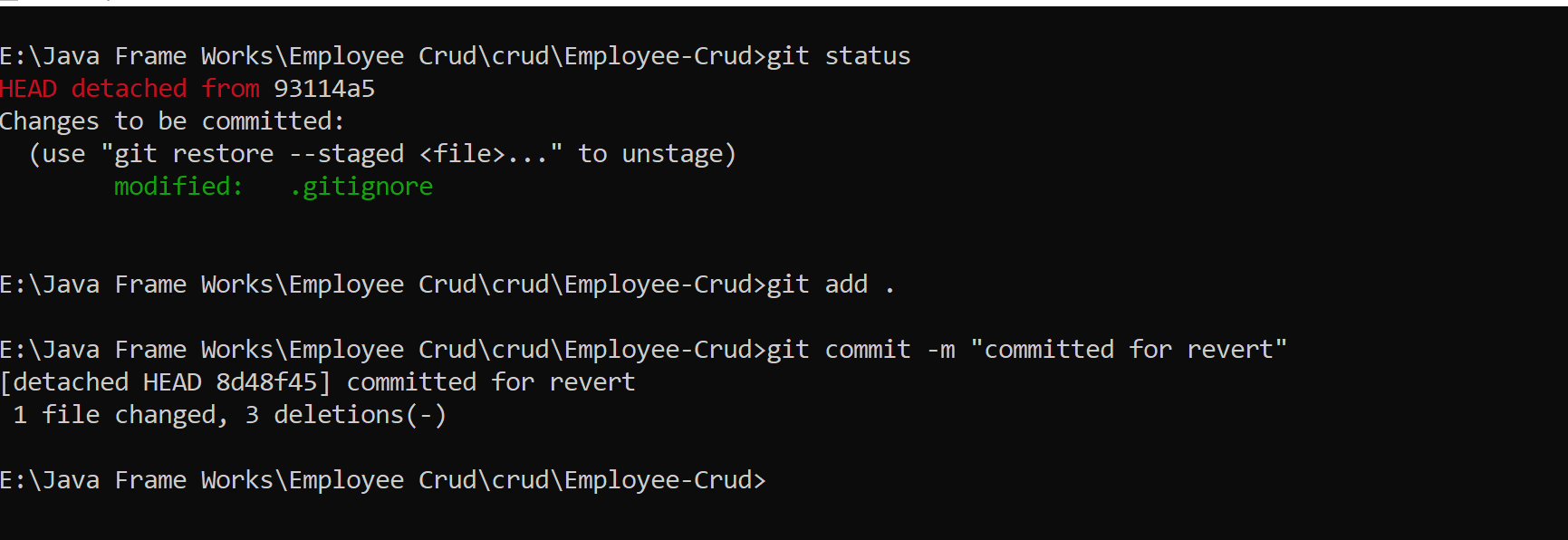
Now check test1.txt not moved to Git Repo.



**Git checkout committed** = going to that committed id

Now open test.txt and some changes and do commit on those changes.





Now I don’t want to go with my changes in test.txt then I have to use **git revert**

**git revert commitid** = revert code back to previous

**git revert HEAD** = revert code back to previous **one commit**

**git revert HEAD-2** = revert code back to previous **2 commit**

above will delete the text from test.txt

**git reset** = it will modify commit history

**git reset commit id**

**git reset HEAD-number** = will modify commit history up to that number

**git reset –soft commit id** = will delete commit history and not code and code moved to staging area

**git reset –hard commit id** = will delete commit history and code also

**Git Branch**

Every git repository having default branch known as master/main

Each branch represents a distinct line of development

**Branch Commands: -**

**git branch** = to see branch name.

**git branch <branch name>** = to create new branch. (git branch dev)

**git checkout <branch name>** = if new branch created, we want t do changes in new branch, then we have to move to that branch for that this is used

**git checkout dev git checkout -b dev** = it will create new branch dev and moving to dev branch from current branch

**git merge dev** = merging dev branch to current branch.