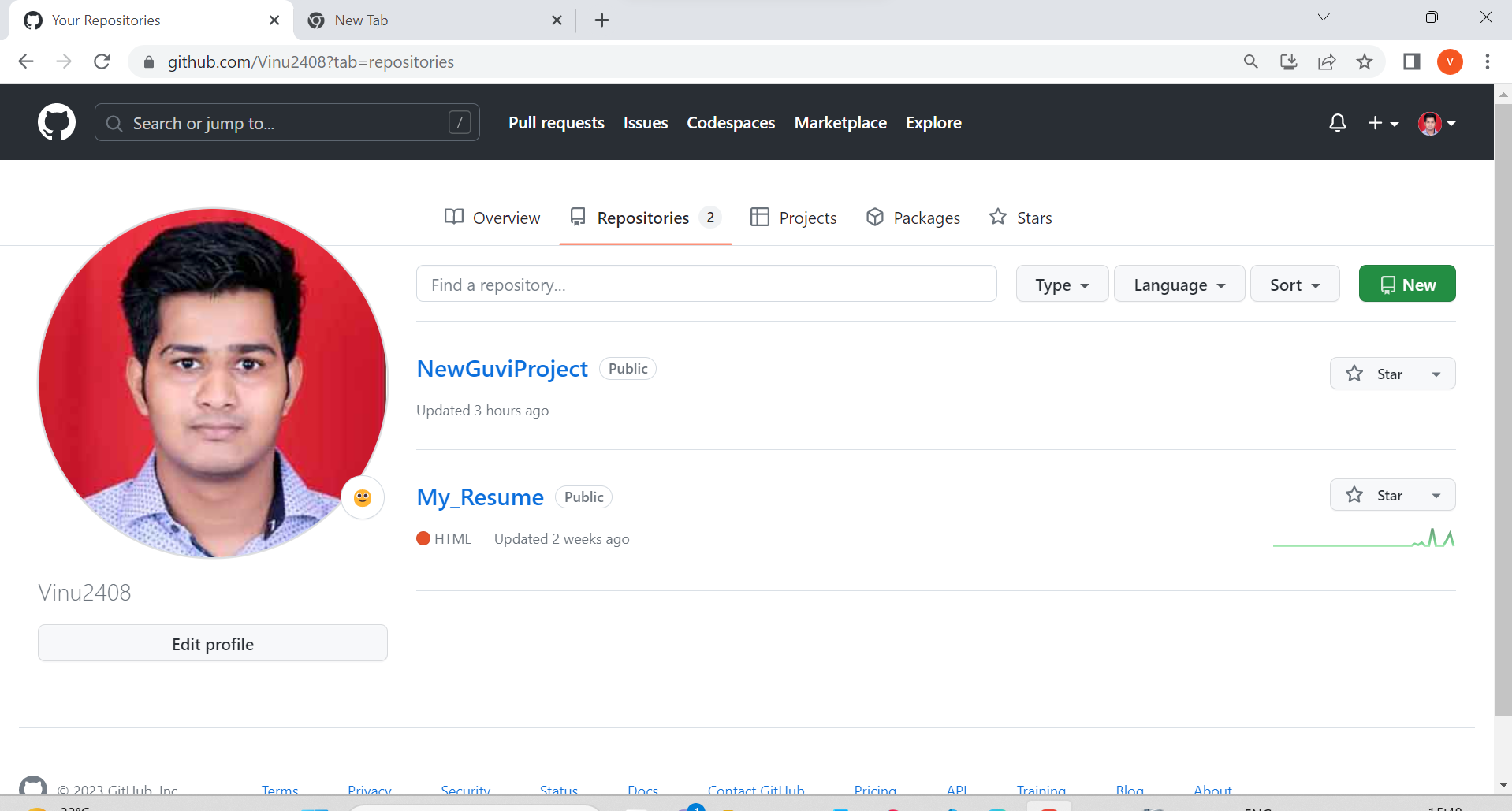
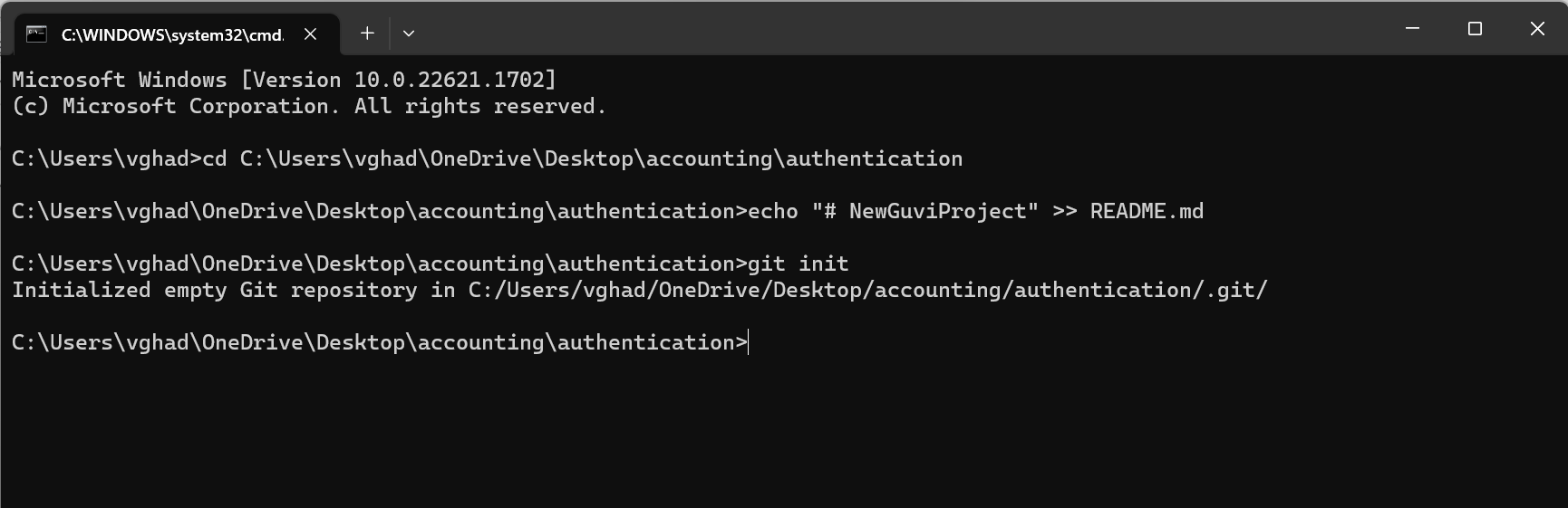
1. Create New Repository on GitHub

Here we created new repository named NewGuviProject. Screenshot attached



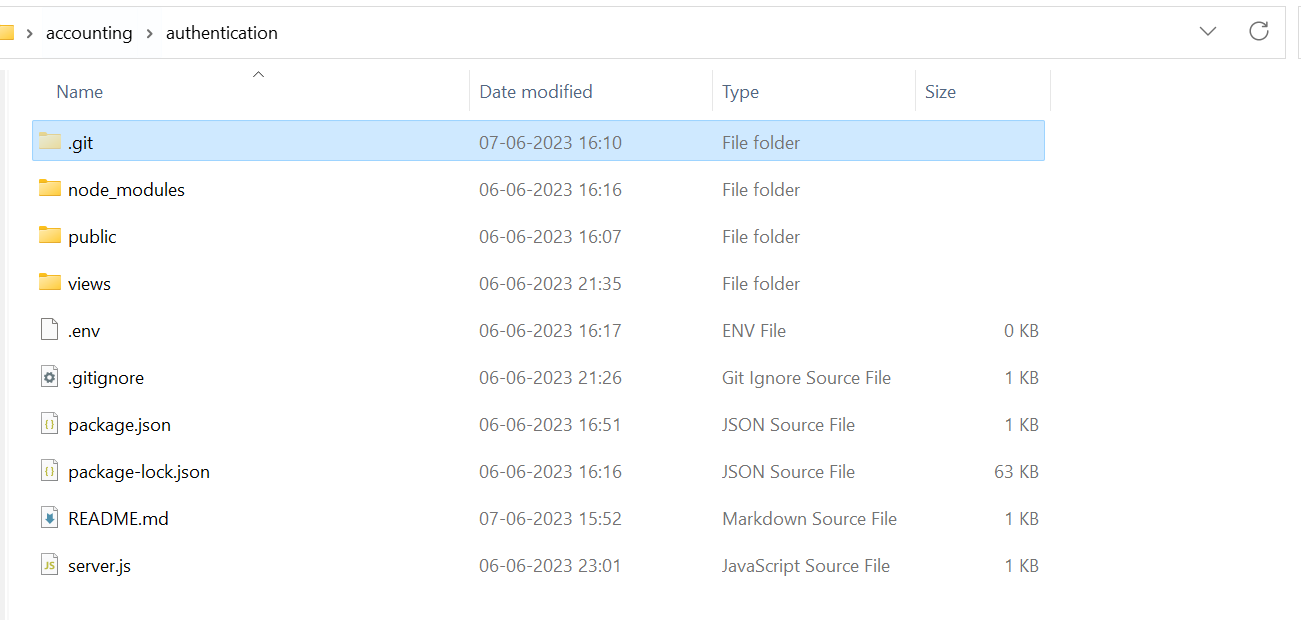
2.Use the init command to create a Git repository in that directory.

Here in command line we have used the init command.



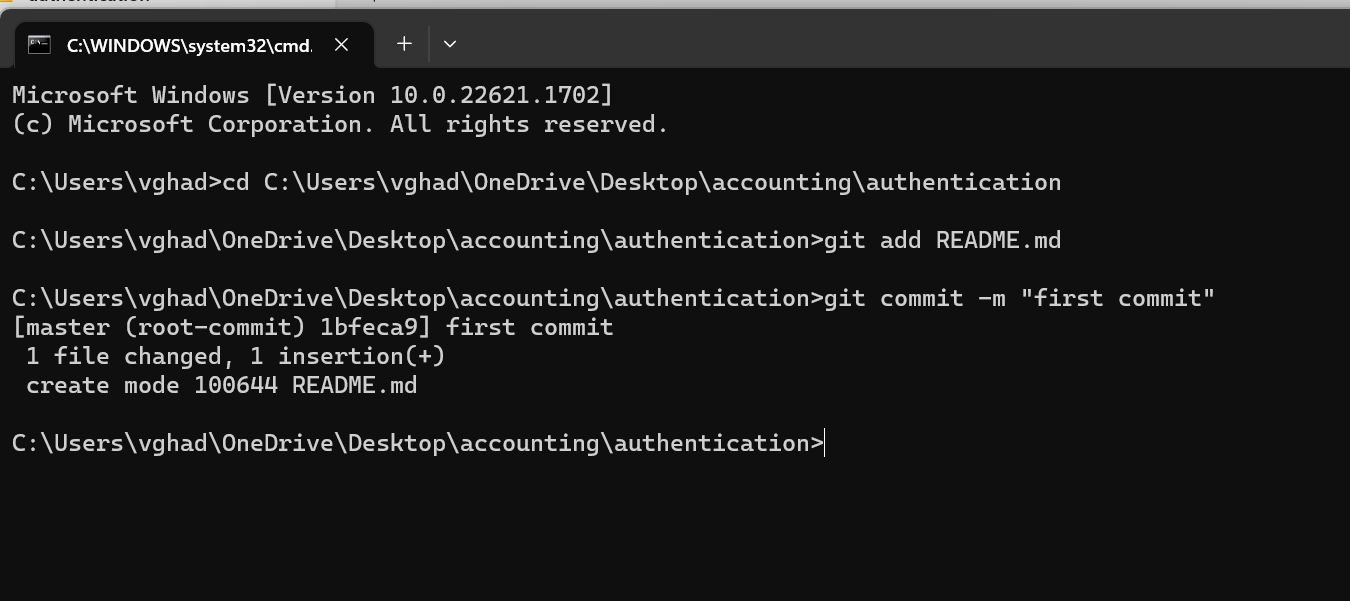
3.Observe that there is now a .git directory

Here we can see that one hidden folder is created named .git



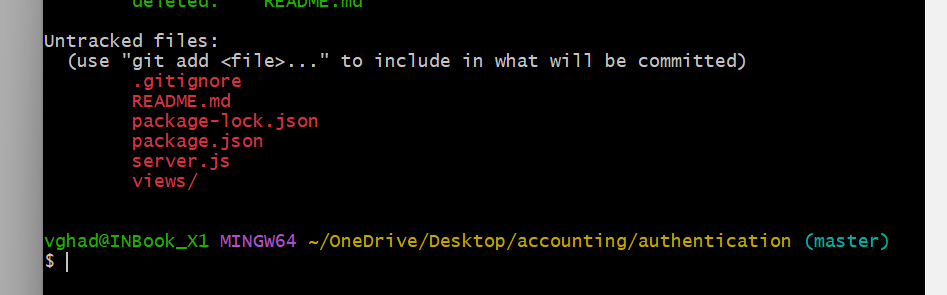
4.Create README file.

To create README file we use command git add README.md



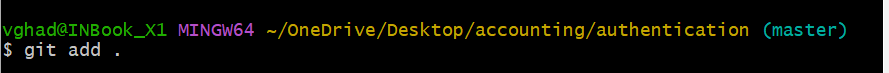
1. Look at the output of the status command, the README you created should appear as an untracked file.

As we see here in this screenshot README.md file

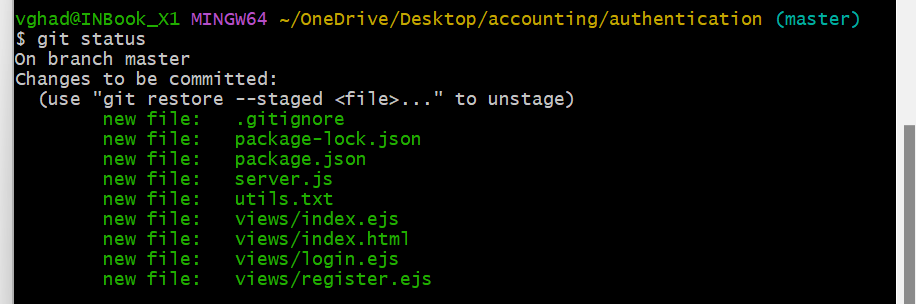


1. Use the add command to add the new file to the staging area. Again look at the output of the status command.

We have added new files also earlier file named README to the staging area here

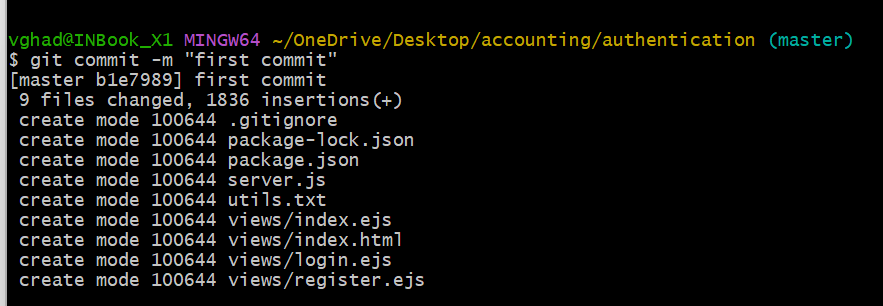


Output of status command



1. Now use the commit command to commit the changes of the staging area.

So we can commit with the help of  git commit -m "first commit"



1. Create a src directory and add couple of files to it.

Here we used mkdir src command to create src directory.

Using the touch command we have created 2 files named file1.txt and file2.txt respectively.

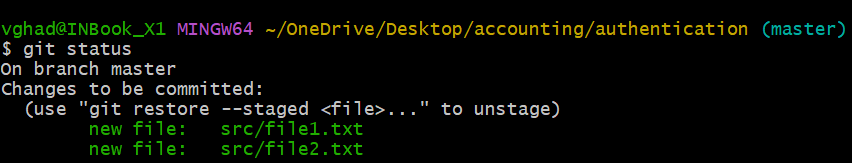


1. Use the add command, but name the directory, not individual files. Use the status command See how both files have been staged. Commit them.

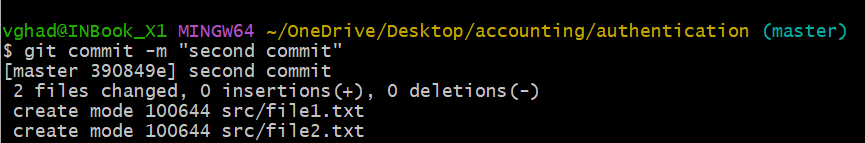
We have to write the below command or we can write git add . To add all files to staging area.



Status command output

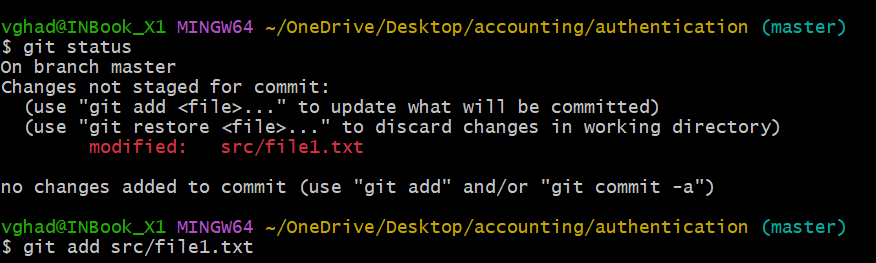


Committed using git commit -m "second commit"

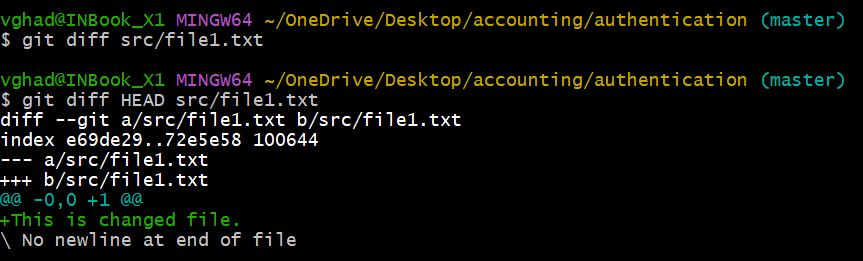


1. Make a change to one of the files . Use diff command to view the details of the change.

We made changes to the file1.txt

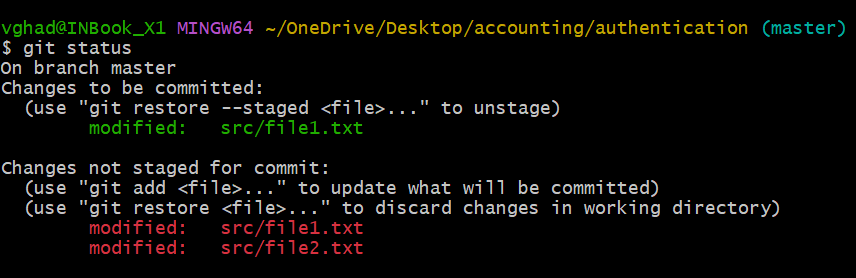


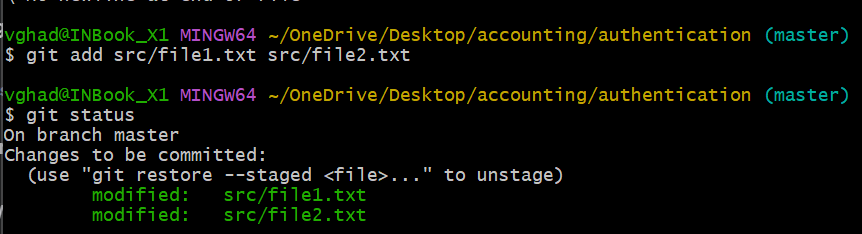
details of the change using diff command

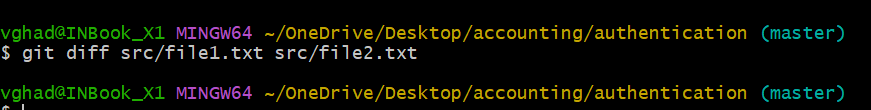


1. Next add the changed file, and notice how it moves to the staging area in the status output. Also observe the diff command you did before using add now gives no output. Why not ?

What do you have to do to see a difference in staging area?

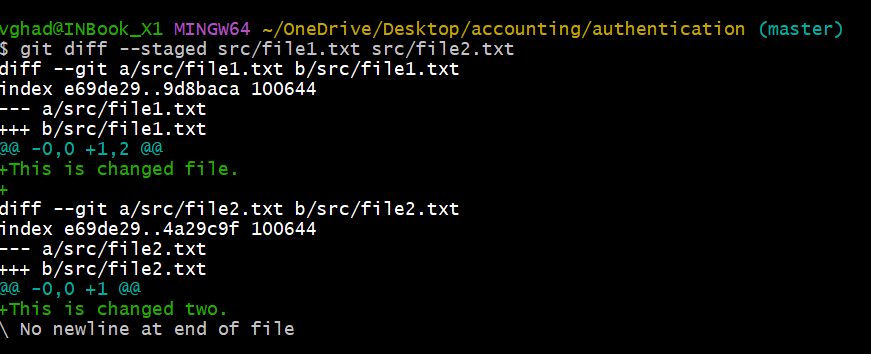






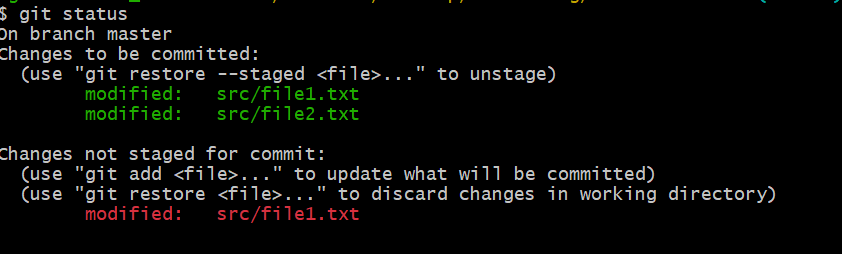
The above command will not show the difference as these files are now in staging area.

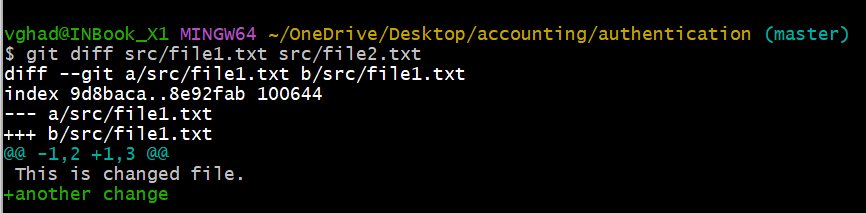
to display output using diff we have to use diff command with --staged



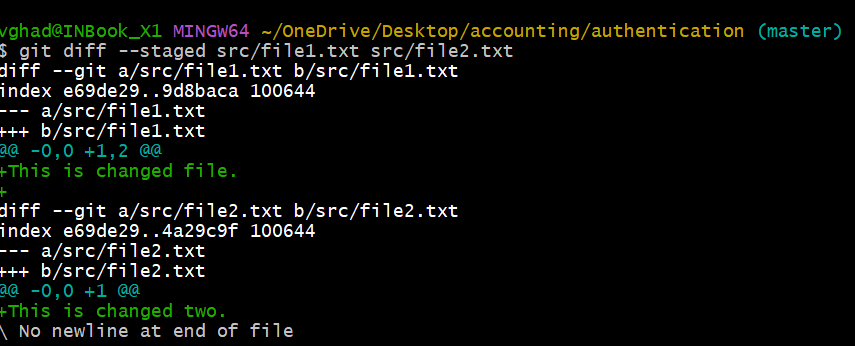
1. Now without commiting – make another change to the same file you changed in step 10. Look at the status output, and the diff output. Notice how you can have both staged and unstaged changes , even when you are talking about a single file. Observe the difference when you use the add command to stage the latest round of changes. Finally commit them. You should now have started to get a feel for the staging area.

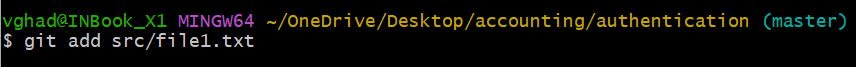
Status output

 diff output

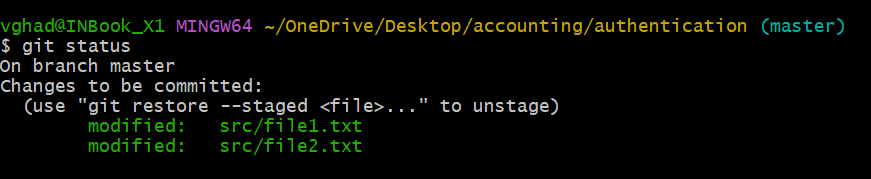


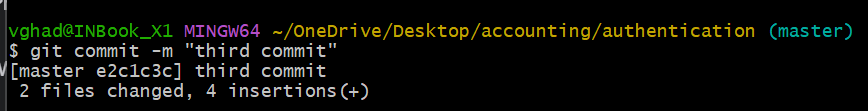
diff output staged

 added latest changed file

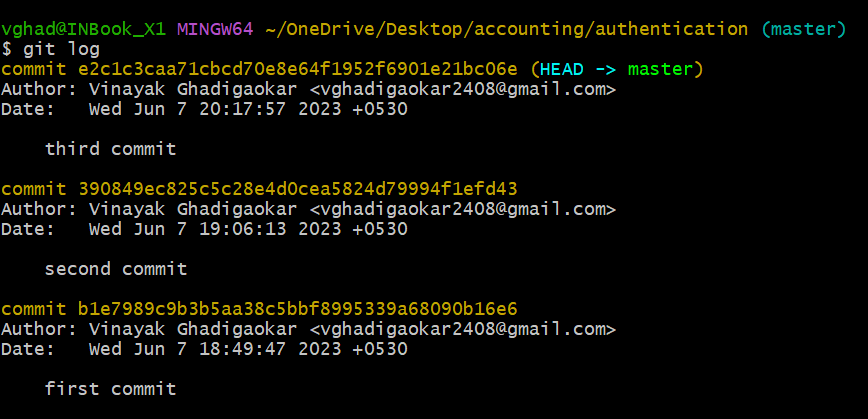


status



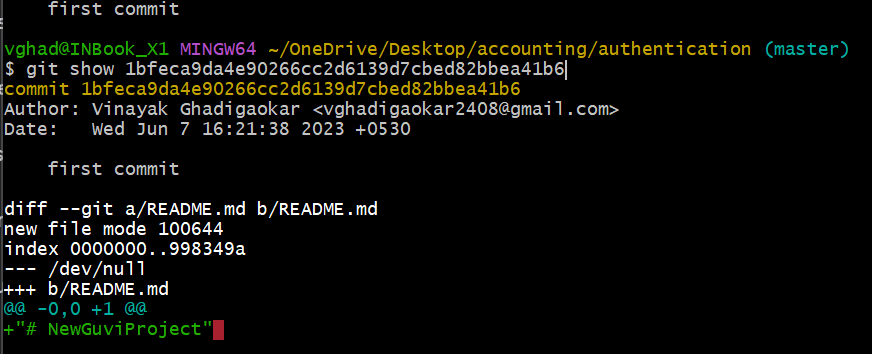
commited changes here finallyd

1. Use the log command to see the commits you have made so far.



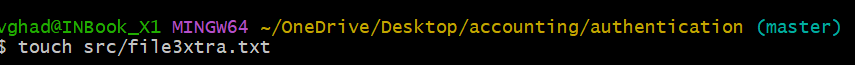
1. Use the show command to look at the individual commits. How many characters of the commit identifier can you get away with typing at a minimum?

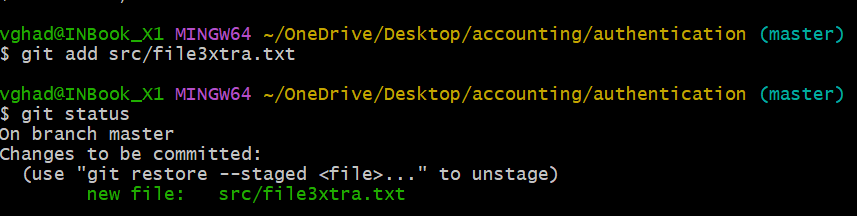
--you can get away with 7 character of the commit identifier or just right click and copy whole as done below.



1. Make couple of more commits, atleast one of it should add an extra file.

Extra file created

 Added to staging

 Commited

