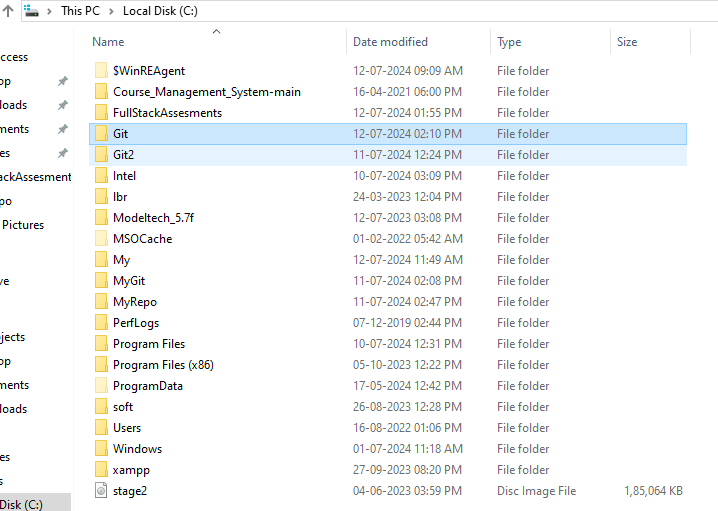
Name :karthikeyan E

Reg no :73152115036

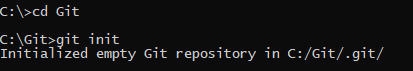
Email:ceeckarthikeyan2025@gmail.com

Day-2 Assesment

1. Create a new directory and change into it.



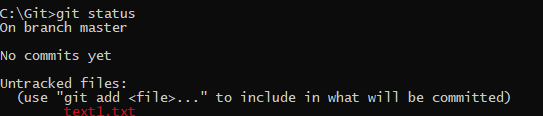
1. Use the init command to create a Git repository in that directory.



3. Observe that there is now a .git directory.

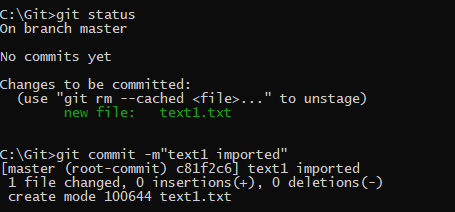
4. Create a README file.

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

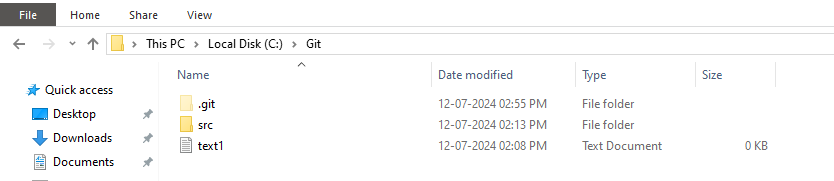


6. Use the add command to add the new file to the staging area. Again, look at the output of the status command.

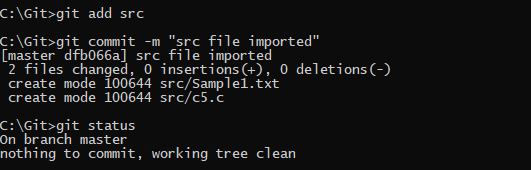
7. Now use the commit command to commit the contents of the staging area.



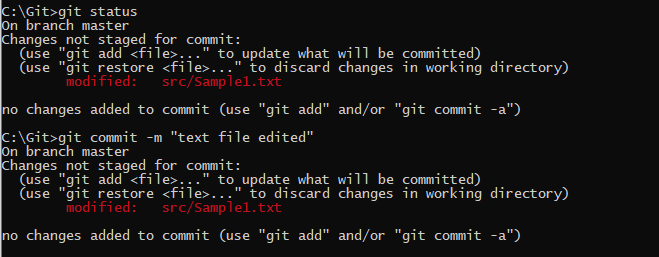
8. Create a src directory and add a couple of files to it.



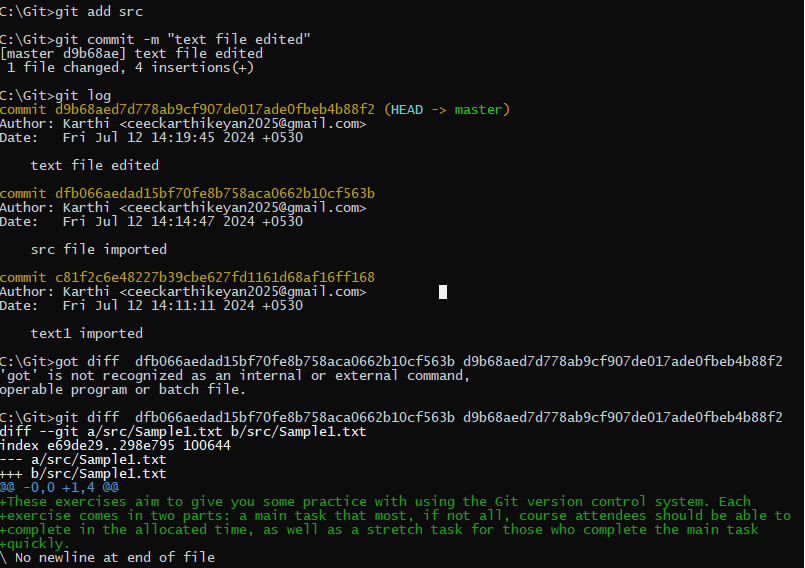
9. Use the add command, but name the directory, not the individual files. Use the status command. See how both files have been staged. Commit them.



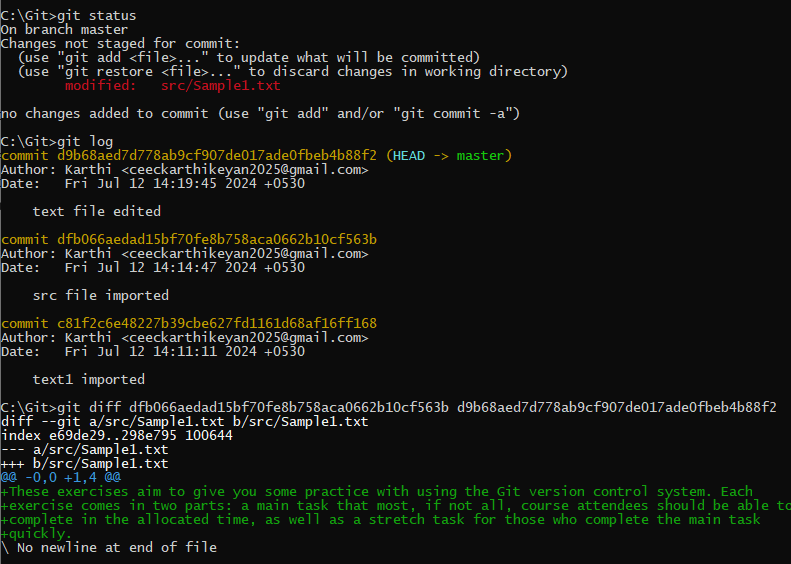
10. Make a change to one of the files. Use the diff command to view the details of the change.



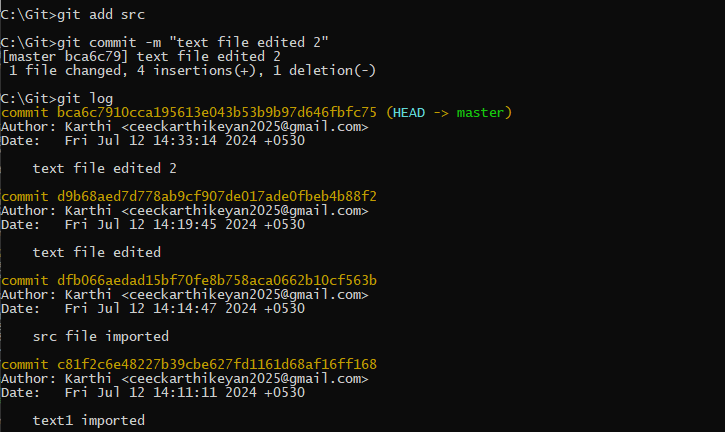
11. Next, add the changed file, and notice how it moves to the staging area in the status output. Also observe that the diff command you did before using add now gives no output. Why not? What do you have to do to see a diff of the things in the staging area? (Hint: review the slides if you can’t remember.)



12. Now – without committing – 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’re 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.



13. Use the log command in order to see all of the commits you made so far.



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

