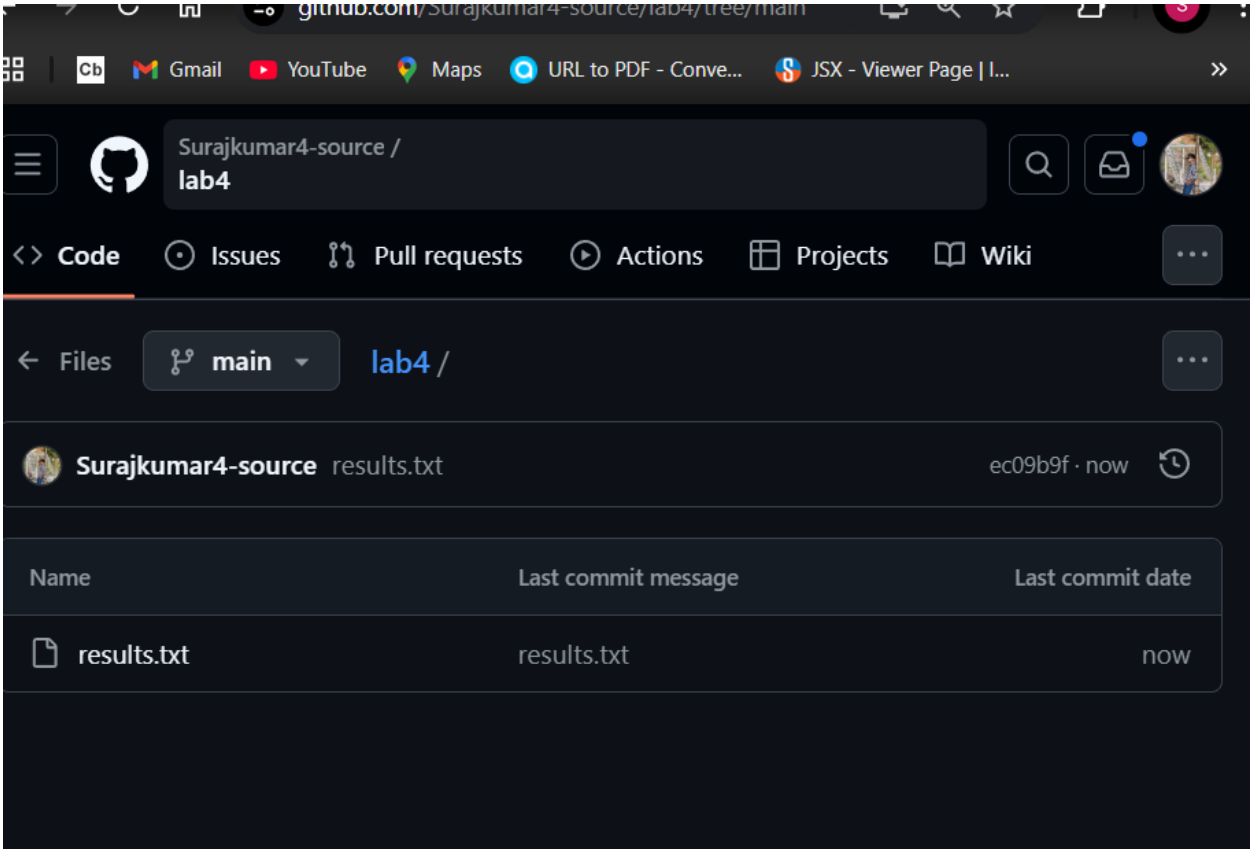
Q1. Create a repository by name lab4 on the Github. Create a file by name results.txt in it. Type some thing into it. Then clone this repository on the local machine. Modify the results.txt file locally. Then push the modifications to the remote repository.

Next modify the results.txt file on Github. Modify the results.txt file locally also. Make sure the contents are different. Then try git push. Perform git pull. Resolve if there is a conflict. Keep remote and local changes. Then push the local contents.

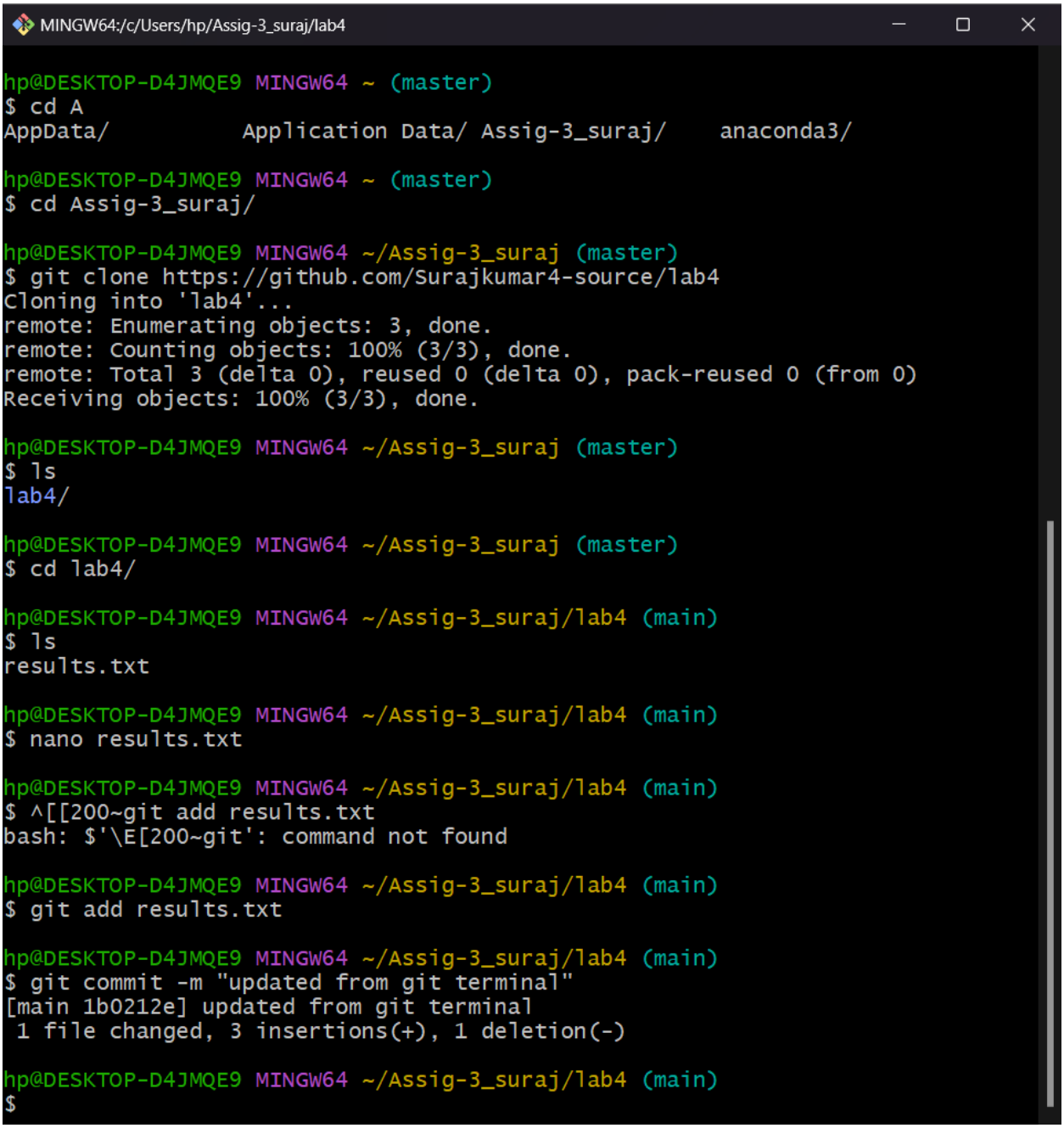
1. Create a Repository on GitHub.

2. Create a File named results.txt on GitHub.

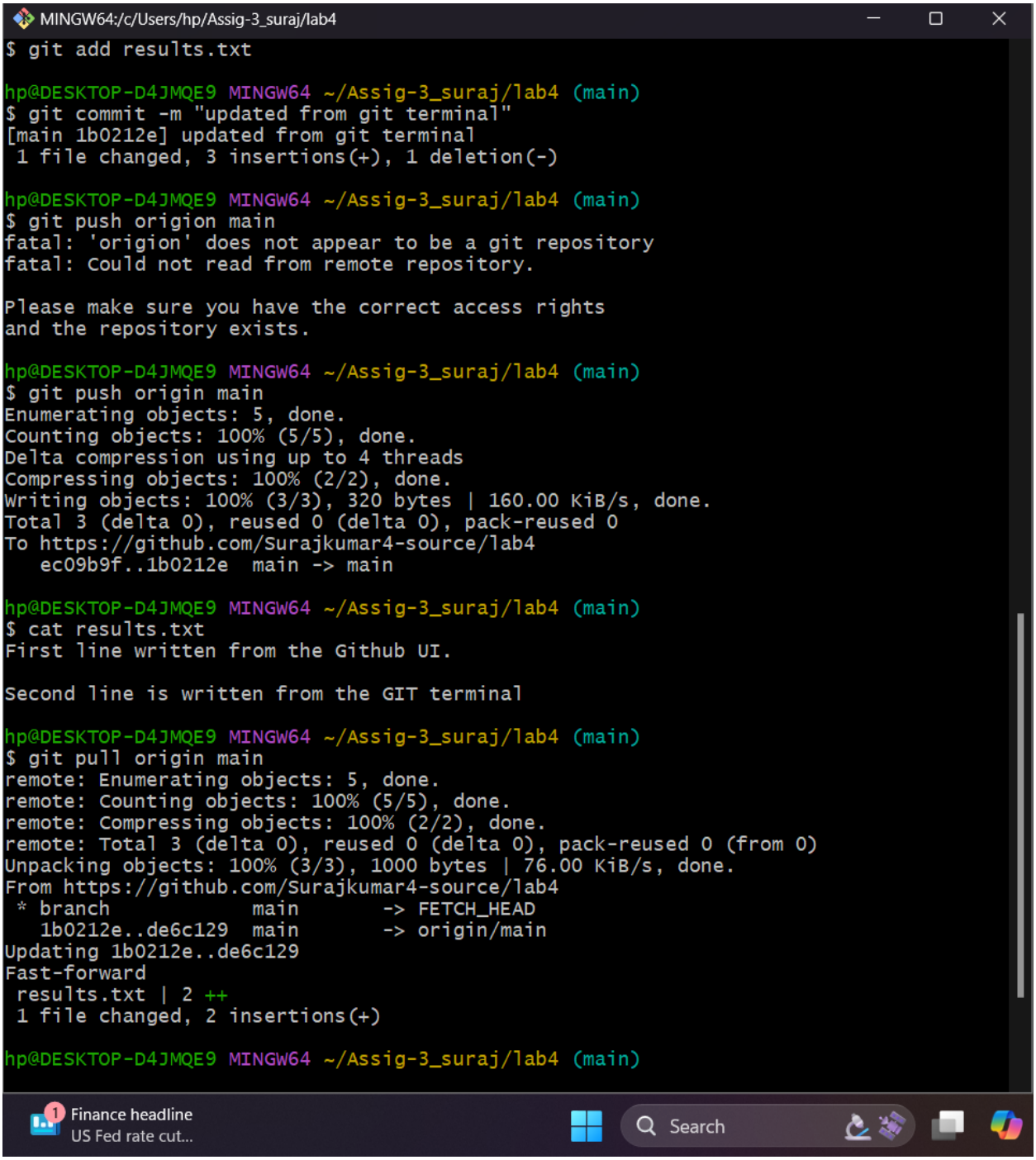


3. Clone the Repository to Your Local Machine.

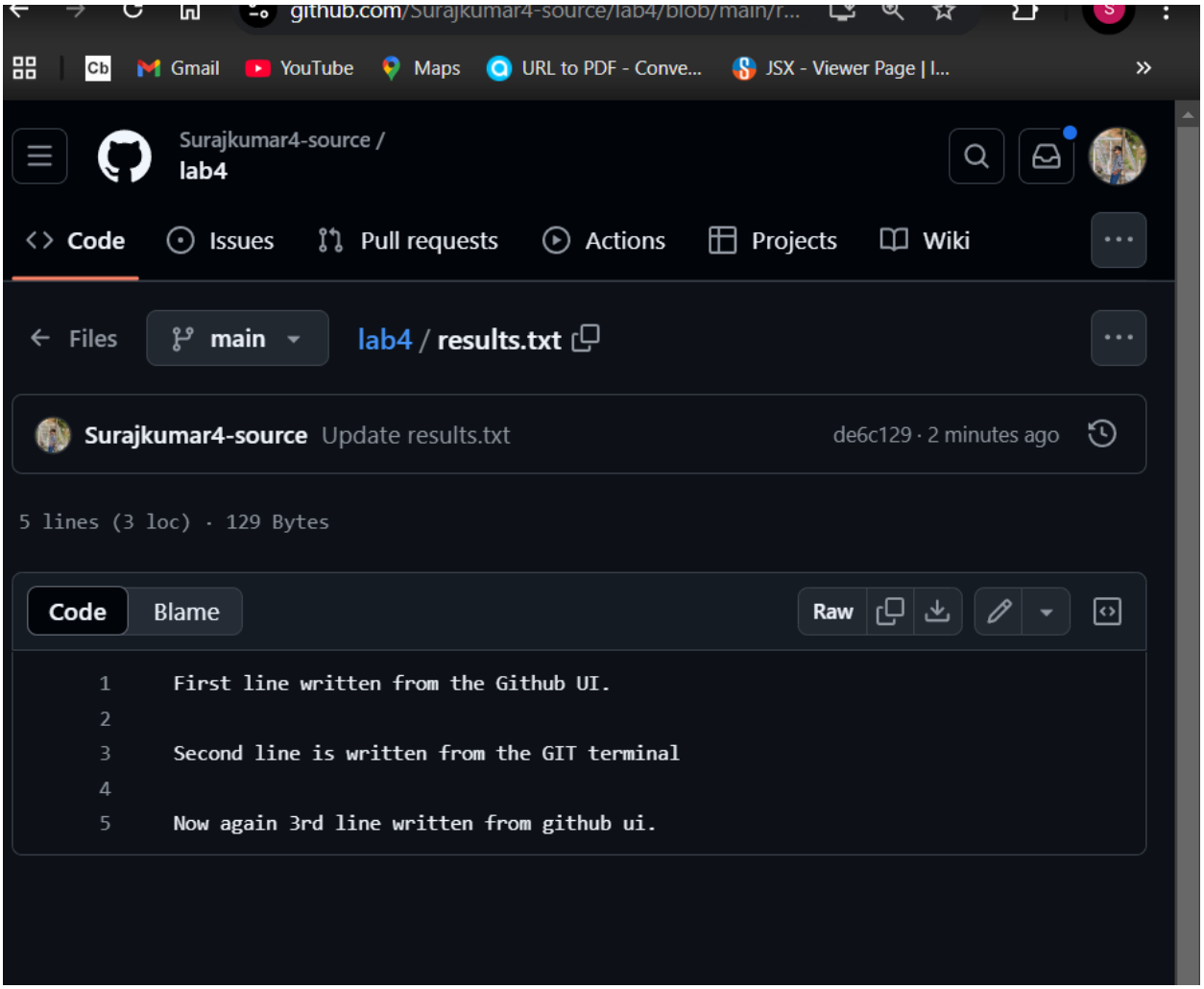
4. Modify the results.txt File Locally.



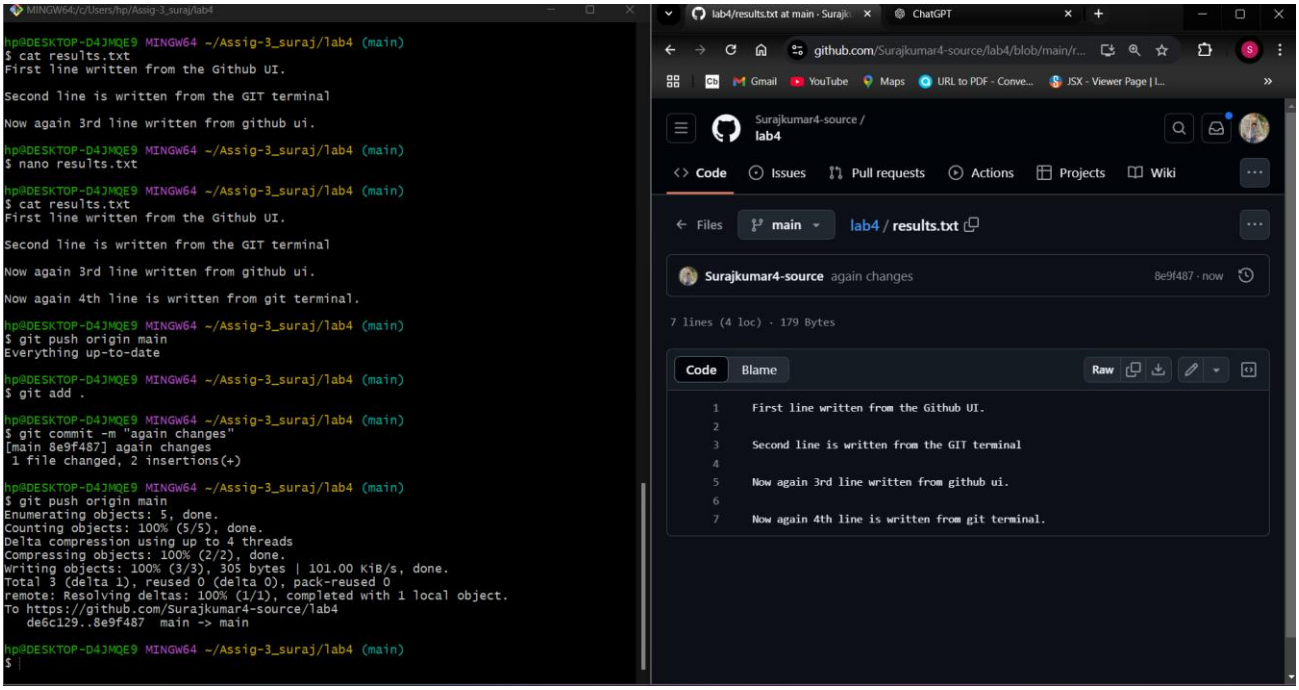
5. Stage and Commit the Changes.



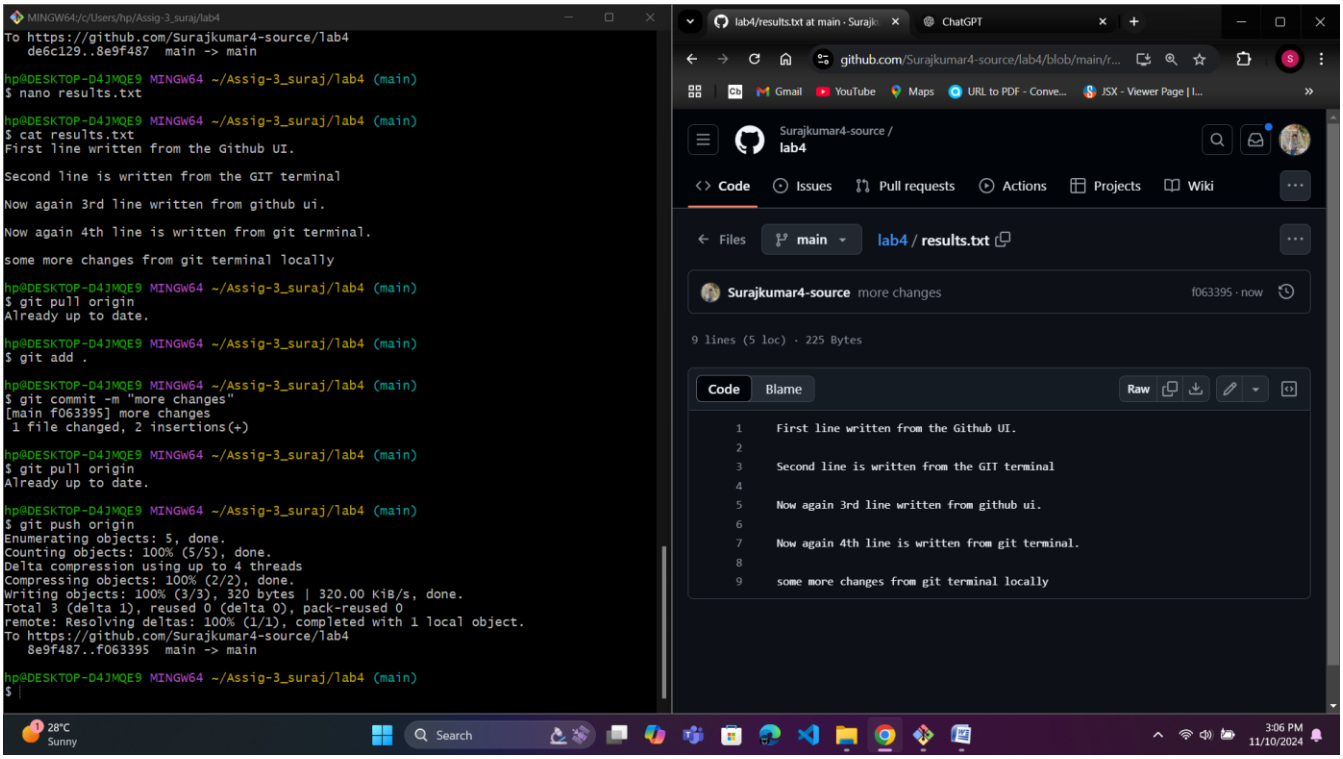
7. Push the Changes to GitHub



Modify the `results.txt` File on GitHub



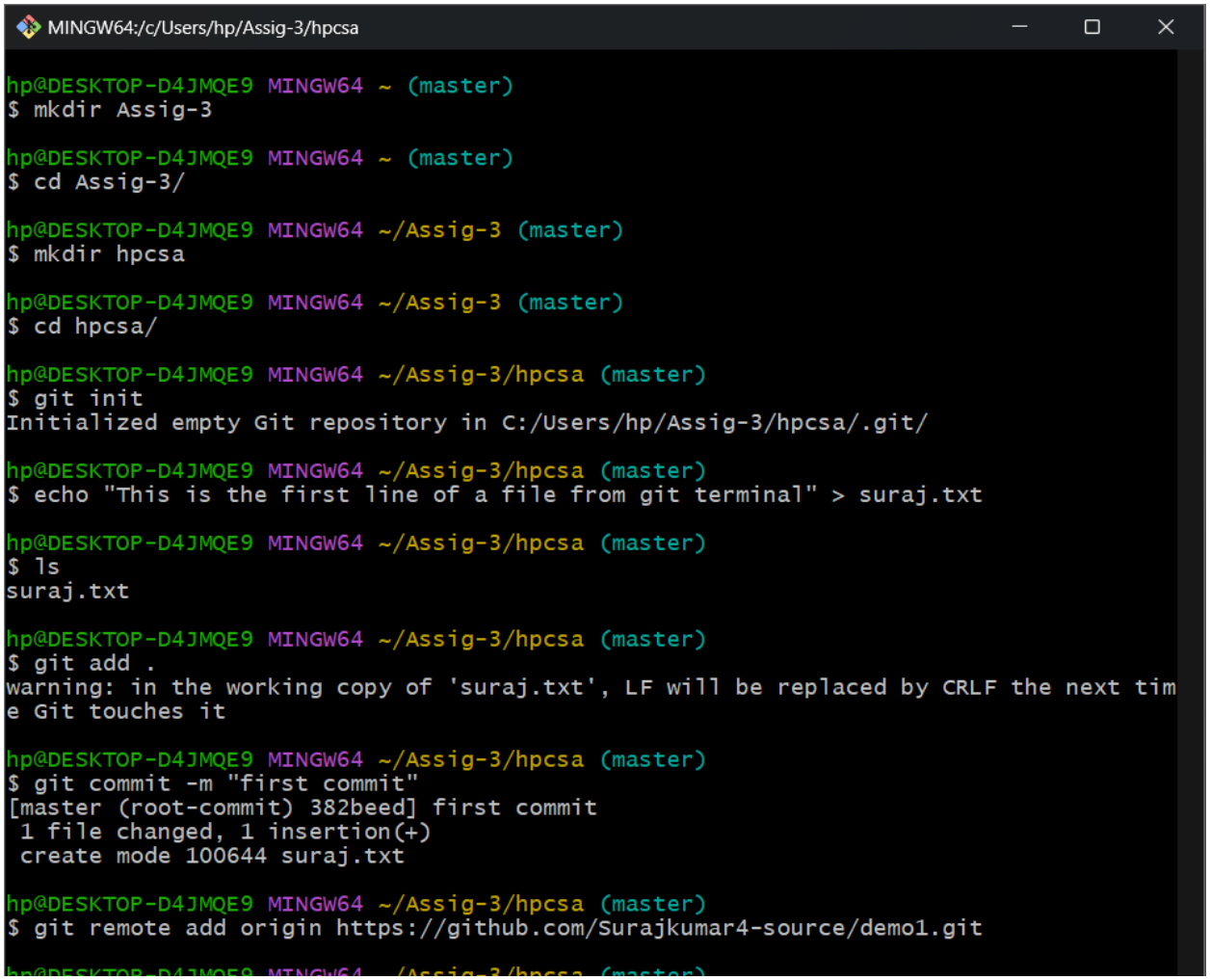
Stage and Commit the Local Changes



Q3. Create a local repository in a directory named hpcsa. Create a file and commit. Create a new repository on Github by name demo1. Link the local repository to the remote repository. Perform git pull. Create a new file by name index.html on local repository and then push it to the remote demo1 repostiroty.

Follwing are the steps to implement:-

1. Create a Local Repository in a Directory name hpcsa
2. Initialize the Local Repository.
3. Create a File and Commit.



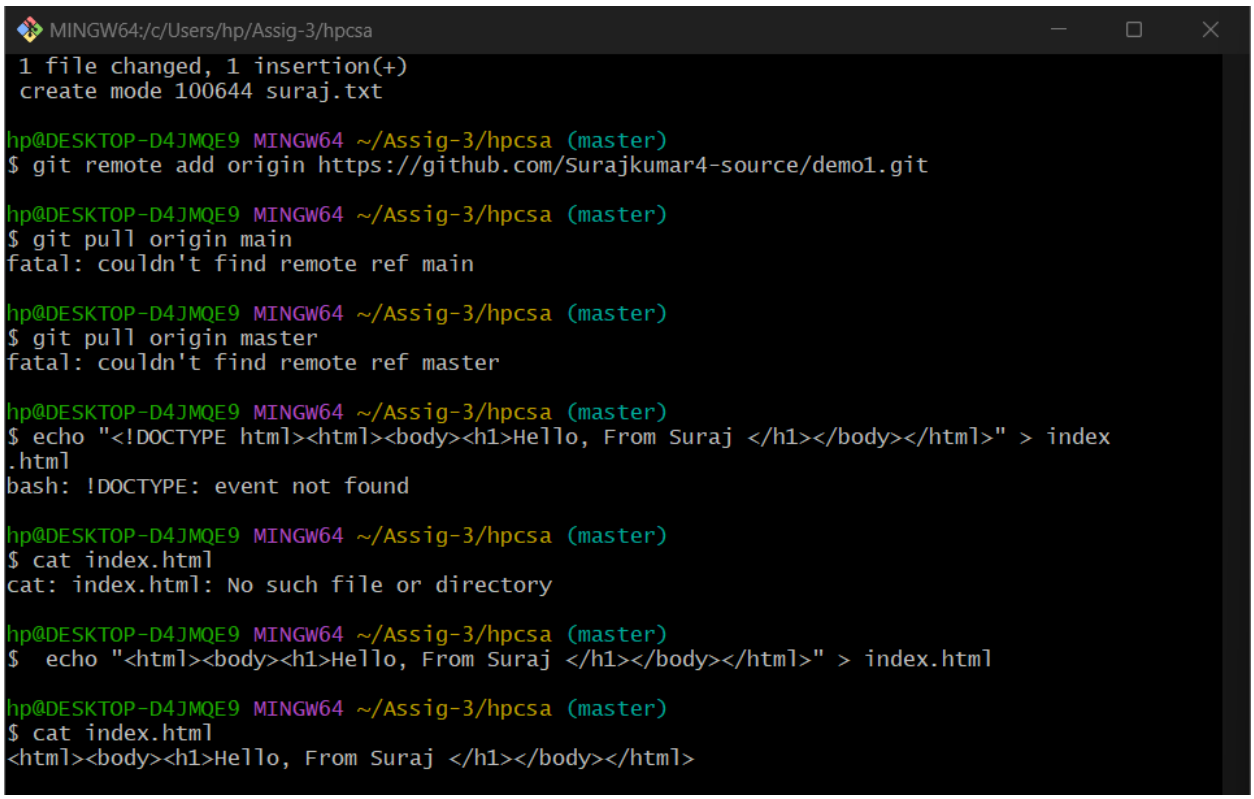
4. Create a New Repository on GitHub Named `demo1`

5. Link the Local Repository to the Remote Repository

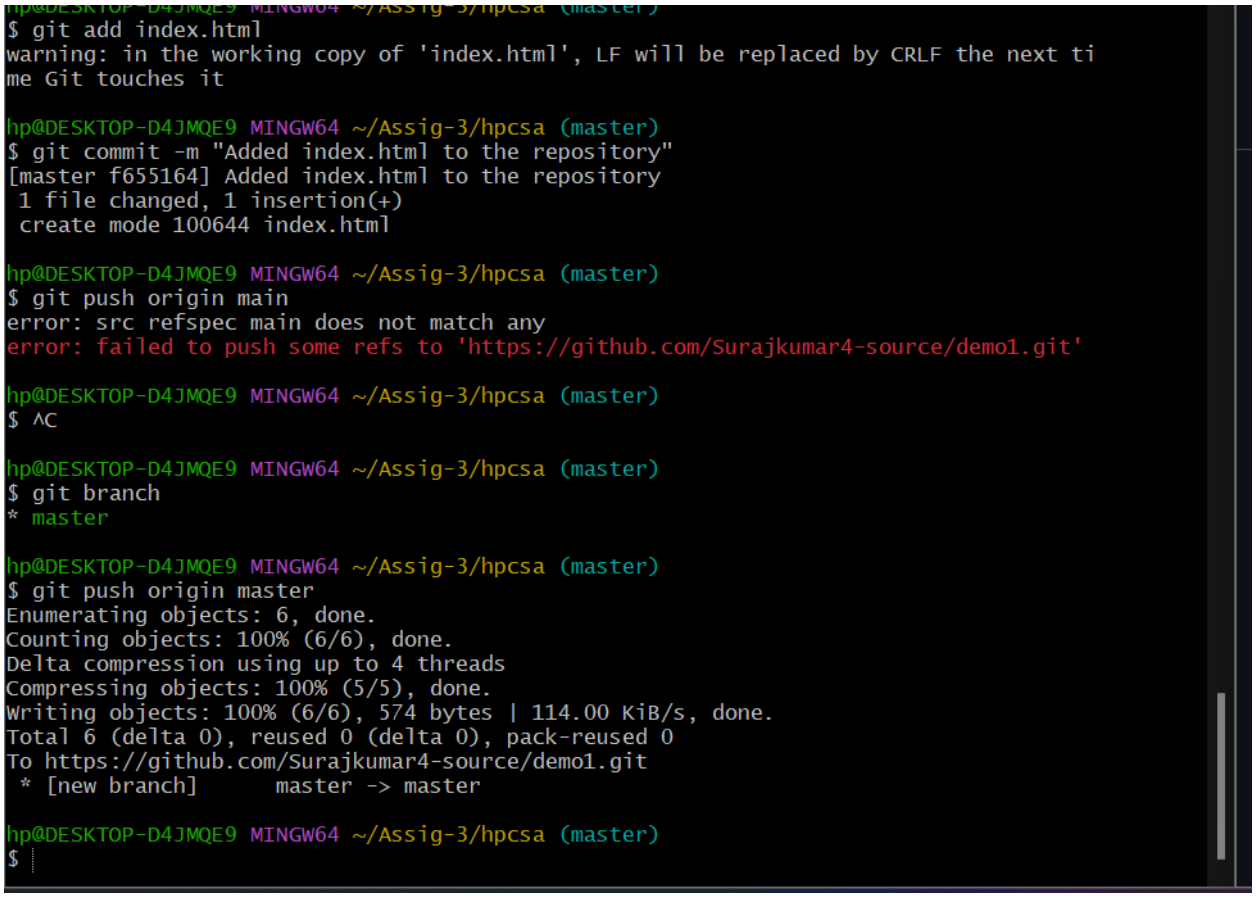
6. Perform a git pull (to Fetch Changes from GitHub)

7. Create a New File Named index.html Locally

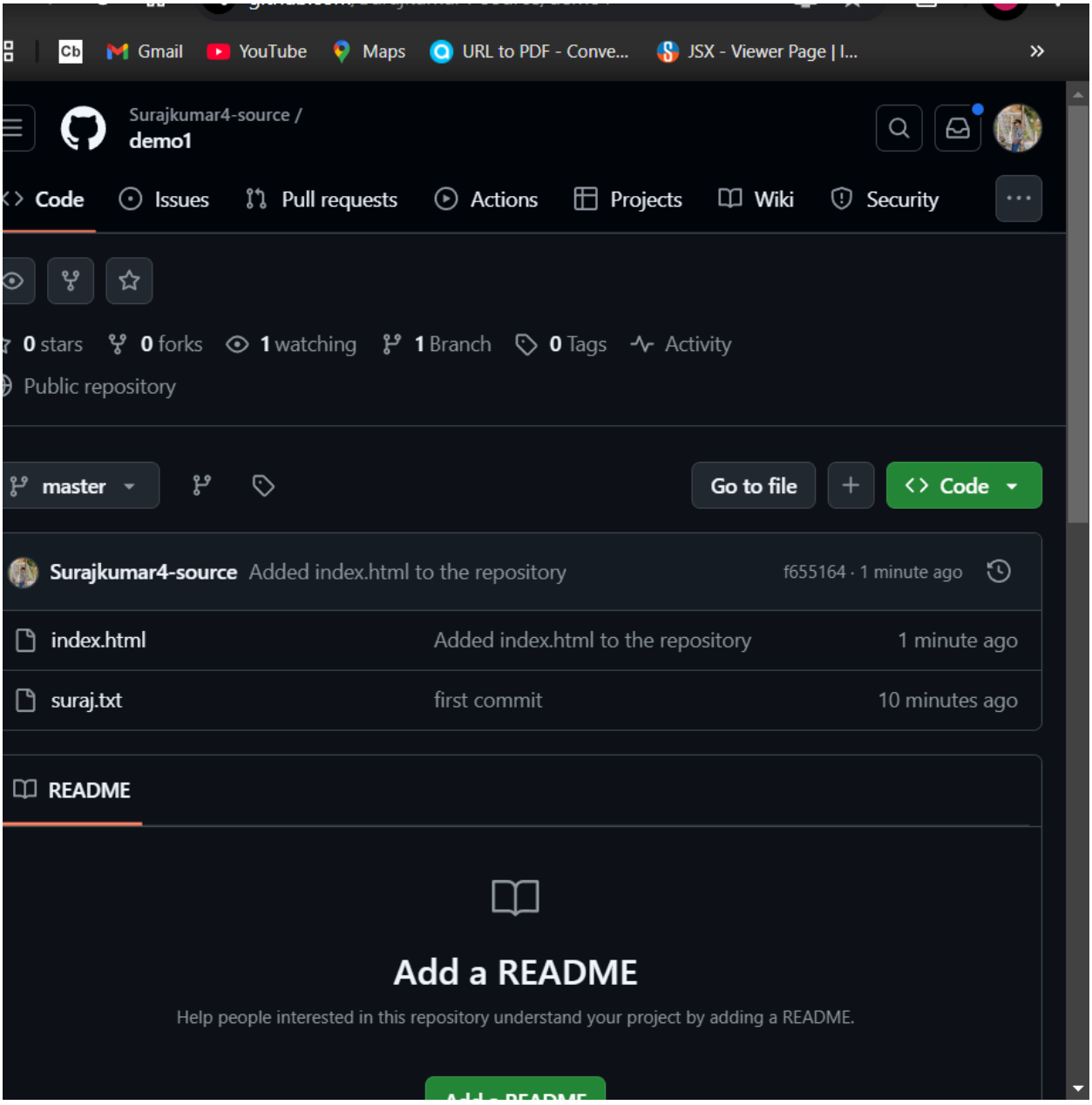
8. Stage and Commit index.html



9. Push `index.html` to the Remote `demo1` Repository on GitHub



Final Output



Q3. Create a container using ubuntu image. Install python in it. Save the container as a image. Delete earlier container. Create a new container and run python in it.

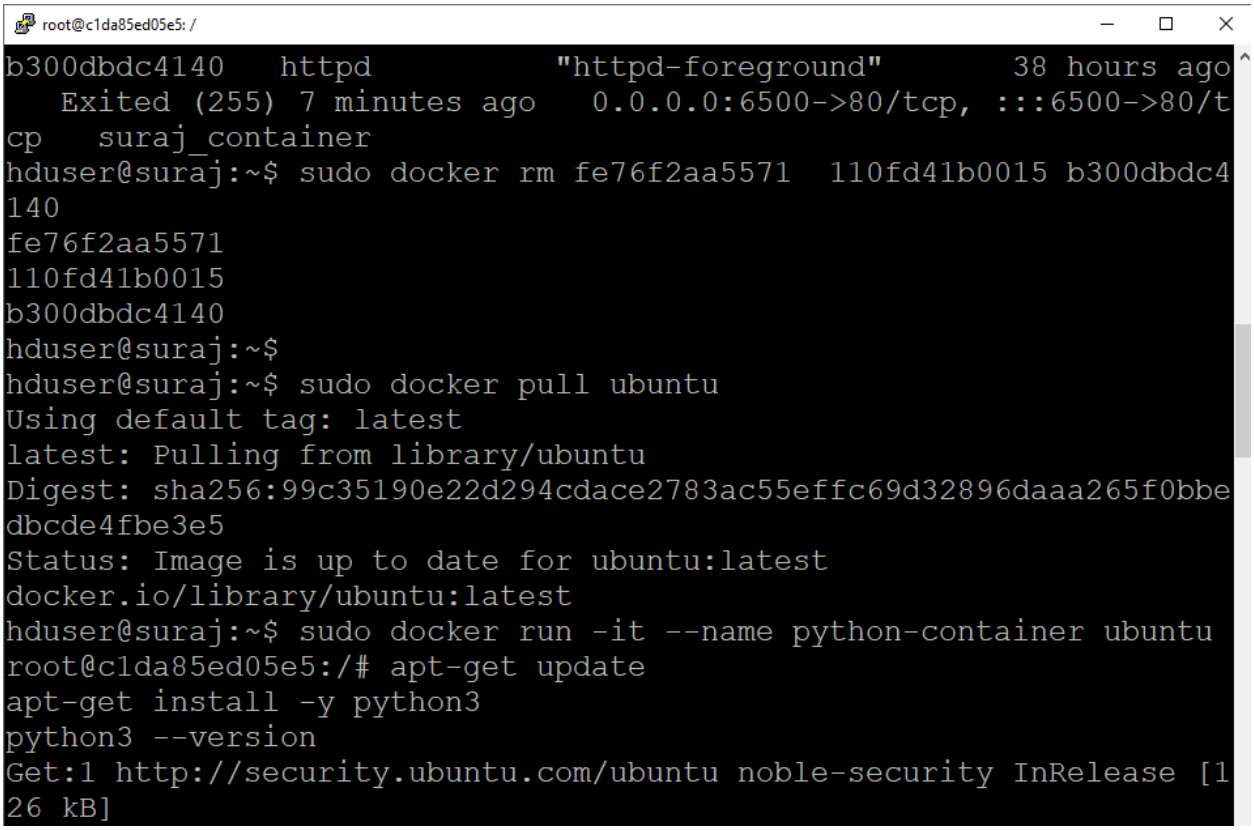
1. Create a container using the Ubuntu image
2. Install Python inside the container
3. Save the container as an image

4. Delete the earlier container

5. Create a new container using the saved image

6. Run Python inside the new container

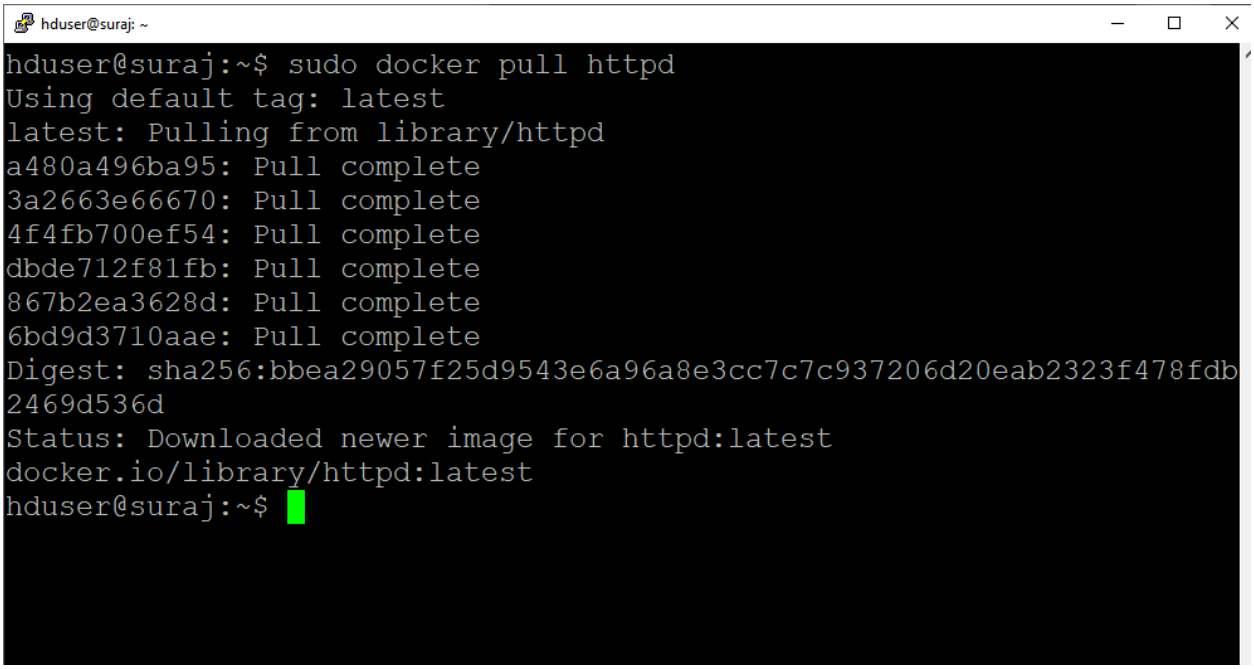
Final Output



```
Running hooks in /etc/ca-certificates/update.d...
done.
Python 3.12.3
root@c1da85ed05e5:/# sudo docker commit python-container python-wit
h-python3
bash: sudo: command not found
root@c1da85ed05e5:/# exit
exit
hduser@suraj:~$ sudo docker commit python-container python-with-pyt
hon3
sha256:5c058a339d2a43228a30b0cea155a1bc1cd7b5641b402ddb021a918be28c
15f5
hduser@suraj:~$ sudo docker rm -f python-container
python-container
hduser@suraj:~$ sudo docker run -it python-with-python3
root@ca1b726bf324:/# python3
Python 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more informati
on.
>>> print("Hello from Suraj")
Hello from Suraj
>>>
```

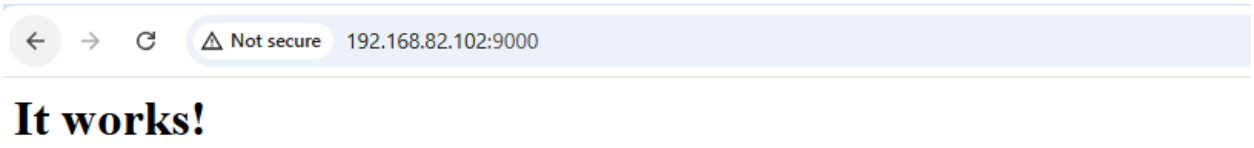
4. Create a container using httpd image. Map outside port 9000. open localhost:9000 on your docker machine. It should display It Works message.

1. **Pull the** httpd **image**
2. Run a container using the httpd image
3. **Access the web server on** localhost:9000
4. **Access the web server on** localhost:9000



```
hduser@suraj:~$ docker run --name webserver -d -p 9000:80 httpd
12431dbc0a81a844992b339051f0b8bef28f25d688d6ff72894c1d30b506a960
hduser@suraj:~$ docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED         STATUS         PORTS                                     NAMES
12431dbc0a81   httpd     "httpd-foreground"   3 seconds ago   Up 2 seconds   0.0.0.0:9000->80/tcp, :::9000->80/tcp   webserver
hduser@suraj:~$ curl http://webserver:9000
curl: (6) Could not resolve host: webserver
hduser@suraj:~$ curl http://localhost:9000
<html><body><h1>It works!</h1></body></html>
hduser@suraj:~$
```

Final output



It works!