Use Case: Setting up a Simple Docker and performing docker commands

Scenario:
You are tasked with setting up a simple Docker environment for testing purposes. You need to pull an image, create a container, run the container, check logs, stop it, and inspect its metadata. Additionally, you need to ensure the container is running within a specific network.

Here are the questions based on the use case scenario:

1. **How can you check if Docker is running on your system?**
2. **What command would you use to pull the hello-world image from Docker Hub?**
3. **How do you create a container named hello_test_container using the hello-world image?**
4. **What is the command to start the hello_test_container container?**
5. **How can you view the logs of the hello_test_container container to check if it is functioning correctly?**
6. **Which command will stop the hello_test_container container once testing is complete?**
7. **How can you inspect the metadata of the hello_test_container container to gather detailed information about it?**
8. **How can you check if a specific network exists or create a network called test_net for your container to use?**

a. **You are asked to perform the operation in the CLI mode**
b. **You need to open CMD using the Ctrl + R and Type CMD**
c. **The windows CMD will open you will need to perform all the commands requested in the question**
d. **After the commands you will need to run the testcase and submit the testcase to git**

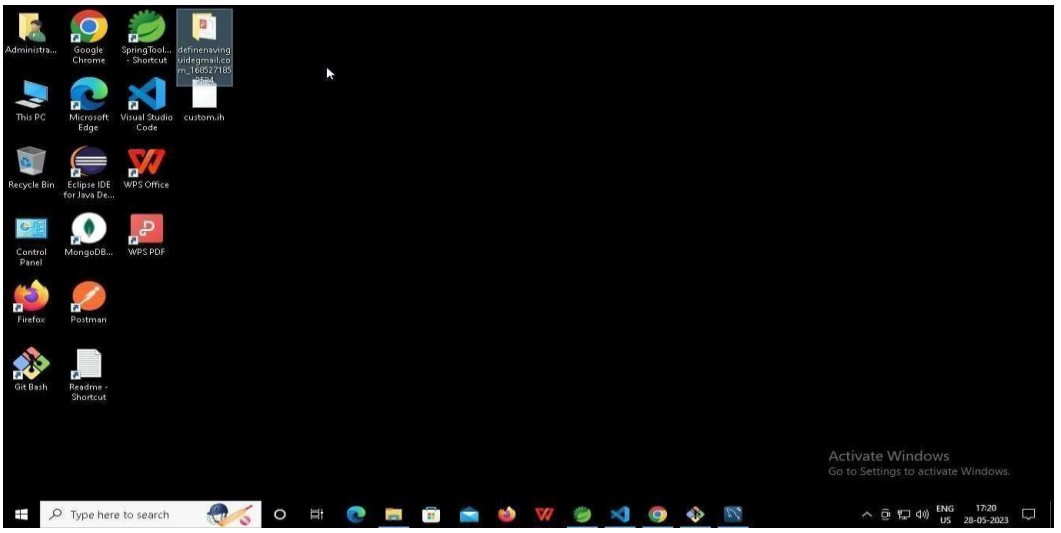
Run the code and submit the code

- **Goto the project folder and in URL open cmd then run the python testcase code**
- **To Run the testcase run the python file in the project folder using the command using the command python run_tests.py**
- After the testcase is completed you push the code to the repository

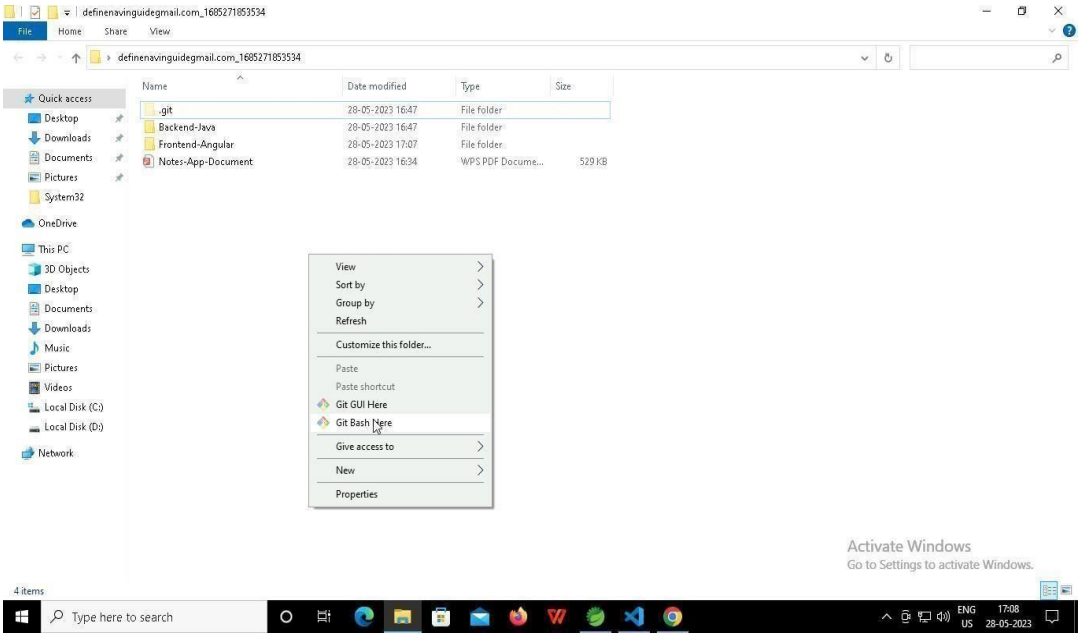
Push the code the repository

You can run test cases as many numbers of times and at any stage of Development, to check how many test cases are passed/failed and accordingly refactor your code.

1. **Make sure before final submission you commit all changes to git**. For that open the project folder available on desktop

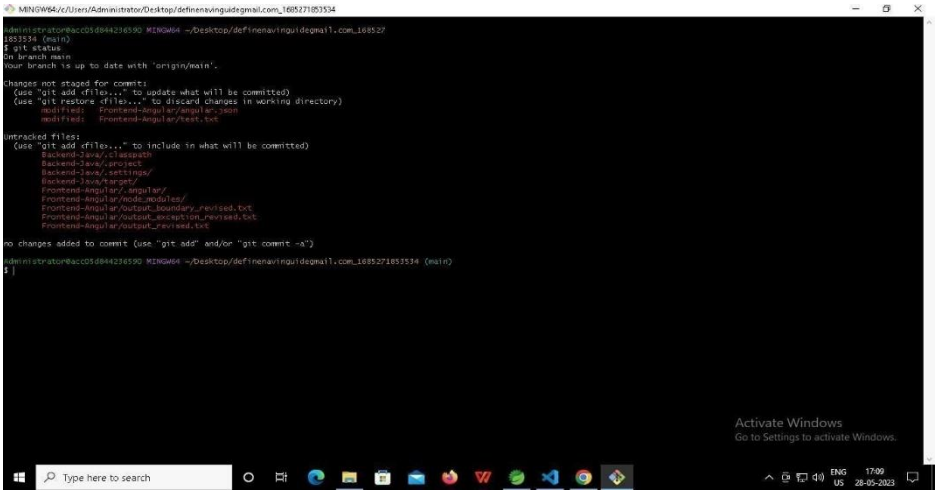


a. Right click in folder and open Git Bash

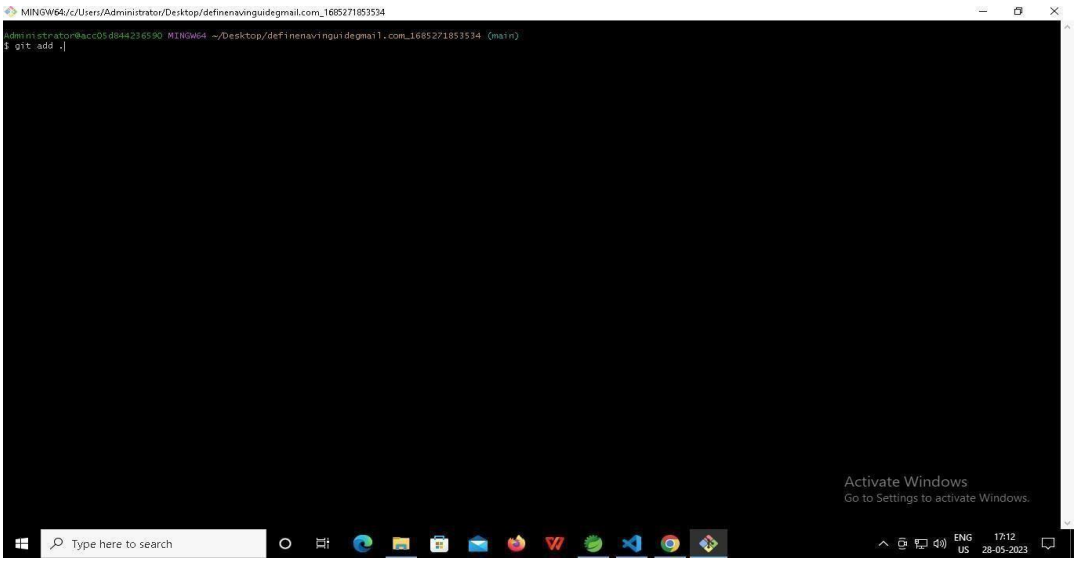


b. In Git bash terminal, run following commands

c. git status

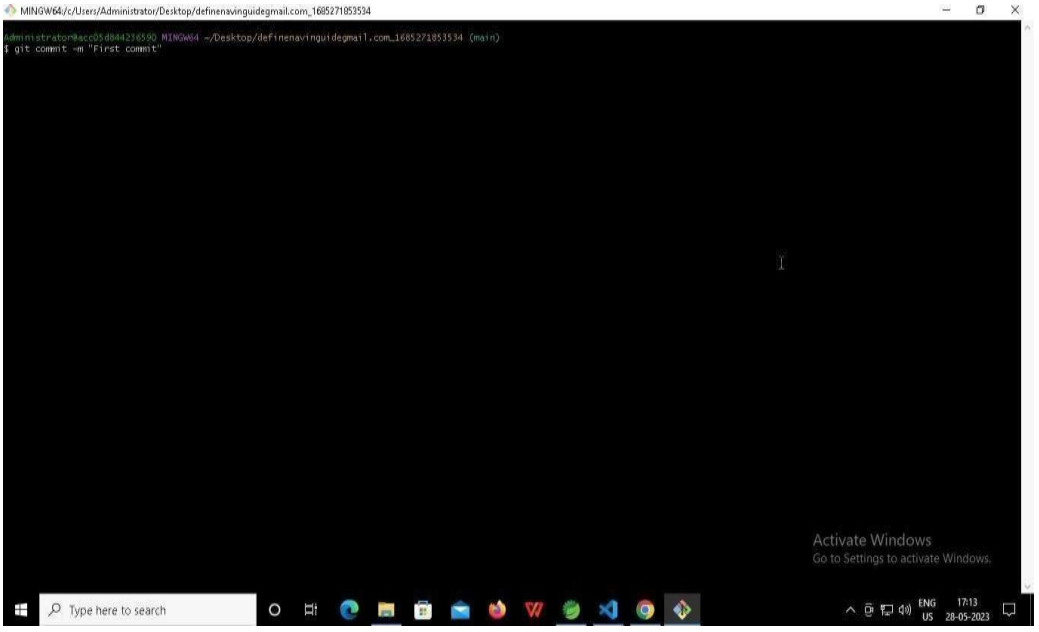


d. git add .

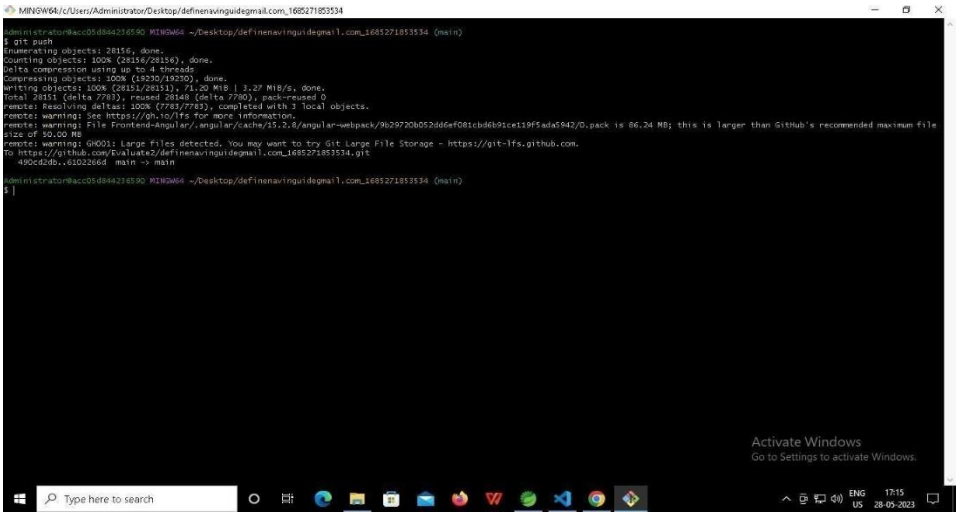


e . git commit -m “First commit”

(You can provide any message every time you commit)



F .git push


MINGW64:/c/Users/Administrator/Desktop/definenavinguidegmail.com_1685271853534
Administrator@acc05d844236590 MINGW64 ~/Desktop/definenavinguidegmail.com_1685271853534 (main)
$ git push
Enumerating objects: 28156, done.
Counting objects: 100% (28156/28156), done.
Delta compression using up to 4 threads
Compressing objects: 100% (19230/19230), done.
Writing objects: 100% (28151/28151), 71.20 MiB | 3.27 MiB/s, done.
Total 28151 (delta 7781), reused 28148 (delta 7790), pack-reused 0
remote: Resolving deltas: 100% (7781/7781), completed with 1 local objects.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File Frontend-Angular/.angular/cache/15.2.8/angular-webpack/9b2972bb052dd6ef081cbd6b91ce119f5ada5942/0.pack is 86.24 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/Evaluate2/definenavinguidegmail.com_1685271853534.git
490cd2db..61022666 main -> main
Administrator@acc05d844236590 MINGW64 ~/Desktop/definenavinguidegmail.com_1685271853534 (main)
$
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
US
17:15
28-05-2023