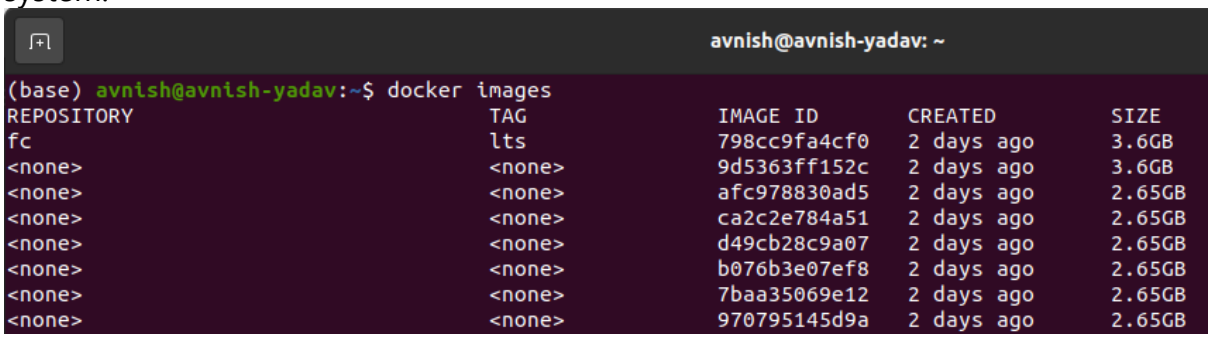
Docker & Docker Hub

Assignment 1:

Demonstrate minimum 15 basic docker command with explanation and screenshot.

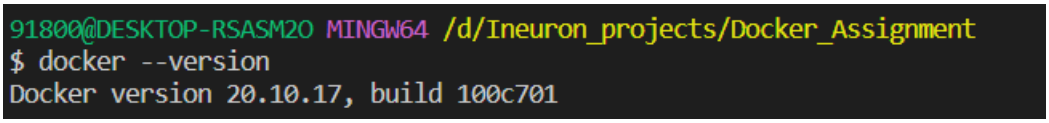
Submission Example

`docker images`
Command is used to list all the docker images available locally in your system.

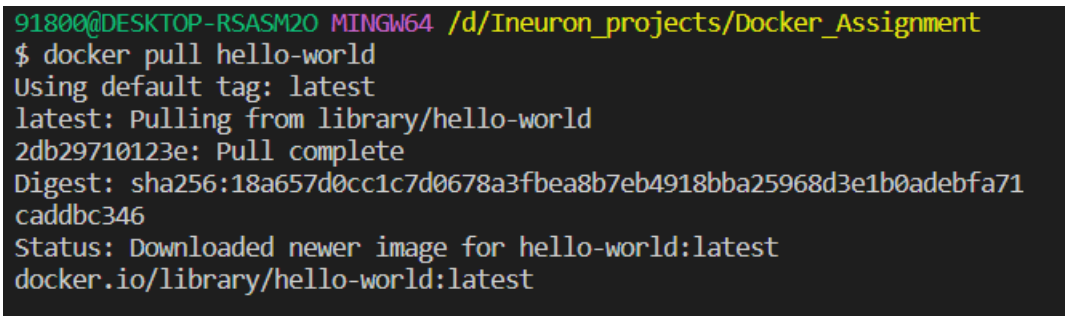


Ans. The docker commands are as follows:-

1. **docker --version ➔ shows installed docker version.**



2. **docker pull [docker_image]:[TAG] ➔ download an image from DockerHub.**



3. **docker images ➔ list all the docker images available locally in your system.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
welcome-app             latest    3a0956f1fb0e   2 days ago      1.06GB
mohitmahi/welcome-app   latest    3a0956f1fb0e   2 days ago      1.06GB
hello-world             latest    feb5d9fea6a5   12 months ago   13.3kB
```

4. **docker rmi -f [docker_image] ➔ delete docker image forcefully.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker rmi -f hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:18a657d0cc1c7d0678a3fbea8b7eb4918bba25968d3e1b0adebfa71caddb
c346
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
```

5. **docker run [docker_image] ➔ Start a new container from an image.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

6. **docker build -t [image_name] . ➔ build an image from Dockerfile.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker build -t  welcome_app .
[+] Building 42.3s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                0.1s
 => => transferring dockerfile: 150B                                                0.0s
 => [internal] load .dockerignore                                                   0.1s
 => => transferring context: 2B                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.7                       3.9s
 => [auth] library/python:pull token for registry-1.docker.io                       0.0s
 => [internal] load build context                                                   0.0s
 => => transferring context: 497B                                                   0.0s
 => CACHED [1/4] FROM docker.io/library/python:3.7@sha256:51dcbb98ba807f3631366d5a  0.0s
 => [2/4] COPY . /app                                                               0.1s
 => [3/4] WORKDIR /app                                                              0.1s
 => [4/4] RUN pip install -r requirements.txt                                      36.4s
 => exporting to image                                                              1.5s
 => => exporting layers                                                             1.4s
 => => writing image sha256:b0b8de4dbf99f6ad7360ae712d00aaaf7309e1175311670d311259  0.0s
 => => naming to docker.io/library/welcome_app                                      0.0s
```

7. **docker run -d -p HostPort:ContainerPort image_name ➔ start a new container in background or detached mode and map a port.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker run -d -p 8000:8000 welcome_app
39e709b8e7c4c0657e7cabf1b211d0c97e7ba5721f4dde14218b643c505b7212
```

8. **docker ps ➔ Shows a list of running containers.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker ps
CONTAINER ID   IMAGE         COMMAND           CREATED          STATUS          PORTS                    NAMES
39e709b8e7c4   welcome_app   "python app.py"   16 seconds ago   Up 14 seconds   0.0.0.0:8000->8000/tcp   musing_driscoll
```

9. **docker stop [Container_ID] ➔ stops a running container.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker stop 39e709b8e7c4
39e709b8e7c4

91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

10. **docker start [Container_ID] ➔ Start a stopped container.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker start be043bb7f8a9
be043bb7f8a9
```

11. **docker rm -f [Container_ID] ➔ removes the running container.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker rm -f 16b889733ce1
16b889733ce1

91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

12. **docker rm [Container_ID] ➔ removes a container.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker stop be043bb7f8a9
be043bb7f8a9

91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker rm be043bb7f8a9
be043bb7f8a9
```

13. **docker rename OLD_NAME NEW_NAME ➔ Rename a container.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker ps
CONTAINER ID   IMAGE         COMMAND           CREATED         STATUS         PORTS                    NAMES
0405321aab0f   welcome_app   "python app.py"   4 minutes ago   Up 3 minutes   0.0.0.0:8000->8000/tcp   strange_mendel

91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker rename strange_mendel mohit-Container

91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker ps
CONTAINER ID   IMAGE         COMMAND           CREATED         STATUS         PORTS                    NAMES
0405321aab0f   welcome_app   "python app.py"   8 minutes ago   Up 8 minutes   0.0.0.0:8000->8000/tcp   mohit-Container
```

14. **docker push [Image_name:TAG] ➔ Upload a docker image from DockerHub.**

```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_with_FastAPI
$ docker push mohitmahi/hello_fastapi:latest
The push refers to repository [docker.io/mohitmahi/hello_fastapi]
cfb9d9838ed3: Pushed
5fe4c93ed190: Pushed
5f70bf18a086: Pushed
dc2c0b6f5058: Pushed
4cfc86a86dc9: Pushed
634bb2b2ca8e: Pushed
80bc379fe03f: Pushed
0c7daf9a72c8: Pushed
75ba02937496: Pushed
288cf3a46e32: Pushed
186da837555d: Pushed
955c9335e041: Pushed
8e079fee2186: Pushed
latest: digest: sha256:1a04aeabfd36d012506da159c8b2b4659714f60186785c5
736a0d3edf82a3e92 size: 3053
```

15. **docker log [Container_ID] ➔ show logs of the container.**

```
(base)
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_with_FastAPI
$ docker logs 539620f2c9f6
INFO:     Will watch for changes in these directories: ['/app']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to qu
it)
INFO:     Started reloader process [1] using StatReload
INFO:     Started server process [8]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

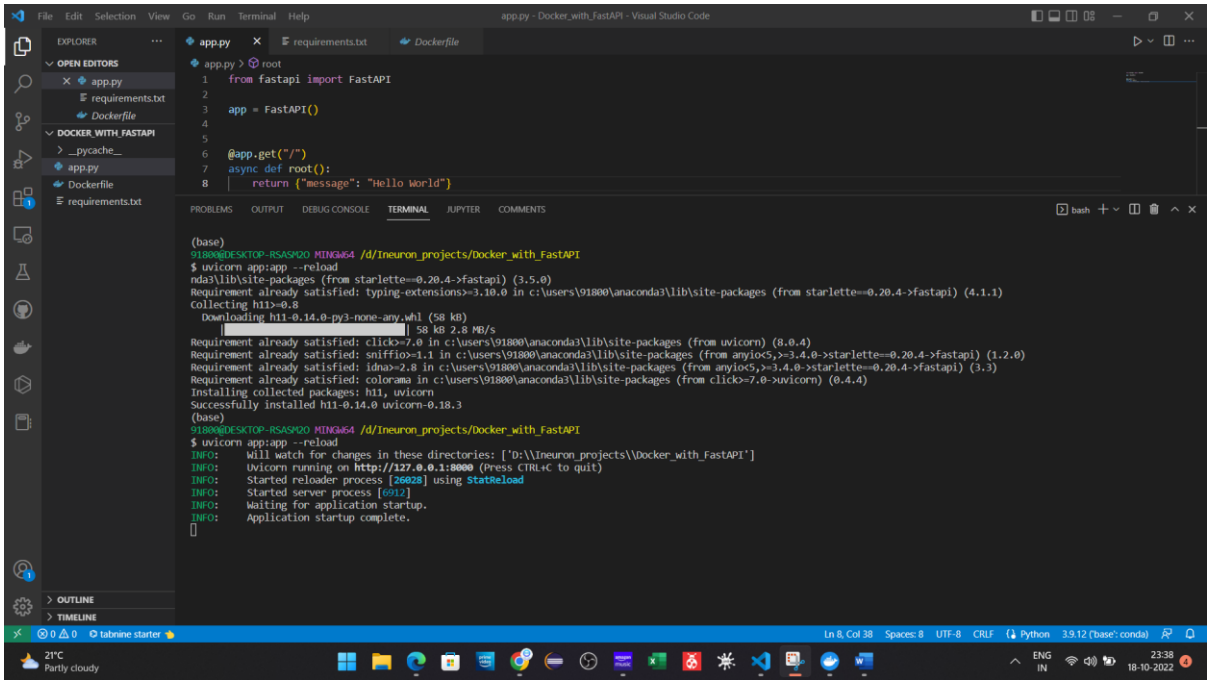
Assignment 2:

<u>Hello World Docker Image</u> Run Hello World Docker Image Locally.

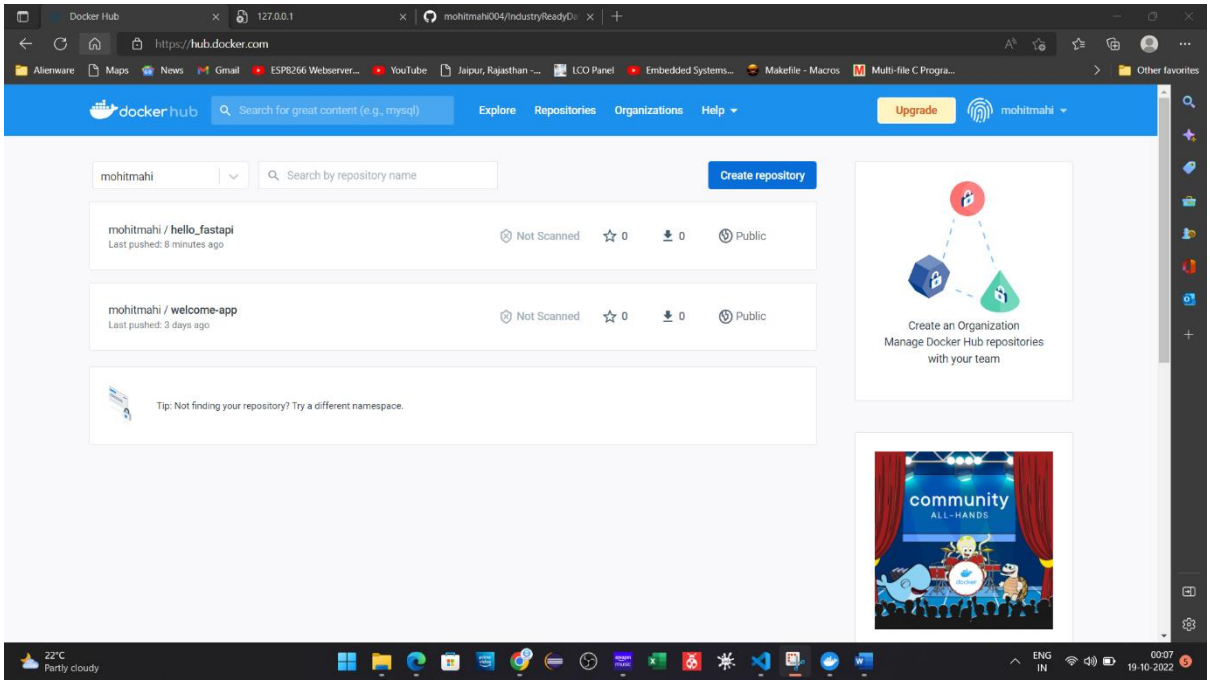
```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:18a657d0cc1c7d0678a3fbea8b7eb4918bba25968d3e1b0adebfa71
caddbc346
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
welcome-app             latest    3a0956f1fb0e   2 days ago      1.06GB
mohitmahi/welcome-app   latest    3a0956f1fb0e   2 days ago      1.06GB
hello-world             latest    feb5d9fea6a5   12 months ago   13.3kB

91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_Assignment
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Assignment 3:

Create a hello world fastapi application. Create a Dockerfile for your fastapi hello world application. Build Docker image using Docker file. Run docker image build in previous step. Push your Docker image to Docker Hub.




```
91800@DESKTOP-RSASM2O MINGW64 /d/Ineuron_projects/Docker_with_FastAPI
$ docker push mohitmahi/hello_fastapi:latest
The push refers to repository [docker.io/mohitmahi/hello_fastapi]
cfb9d9838ed3: Pushed
5fe4c93ed190: Pushed
5f70bf18a086: Pushed
dc2c0b6f5058: Pushed
4cfc86a86dc9: Pushed
634bb2b2ca8e: Pushed
80bc379fe03f: Pushed
0c7daf9a72c8: Pushed
75ba02937496: Pushed
288cf3a46e32: Pushed
186da837555d: Pushed
955c9335e041: Pushed
8e079fee2186: Pushed
latest: digest: sha256:1a04aeabfd36d012506da159c8b2b4659714f60186785c5
736a0d3edf82a3e92 size: 3053
```

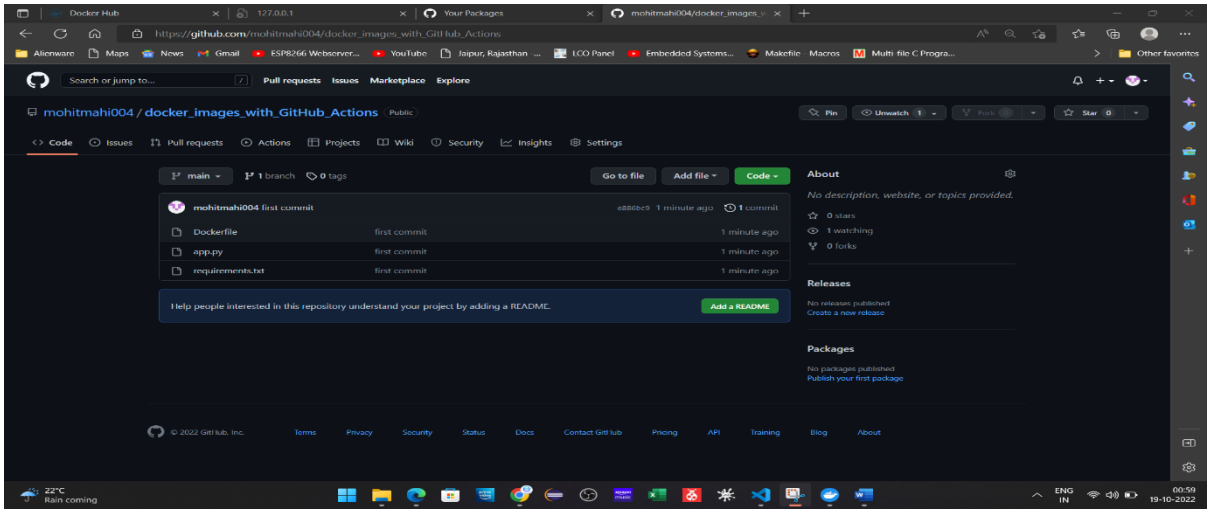


Assignment 4:

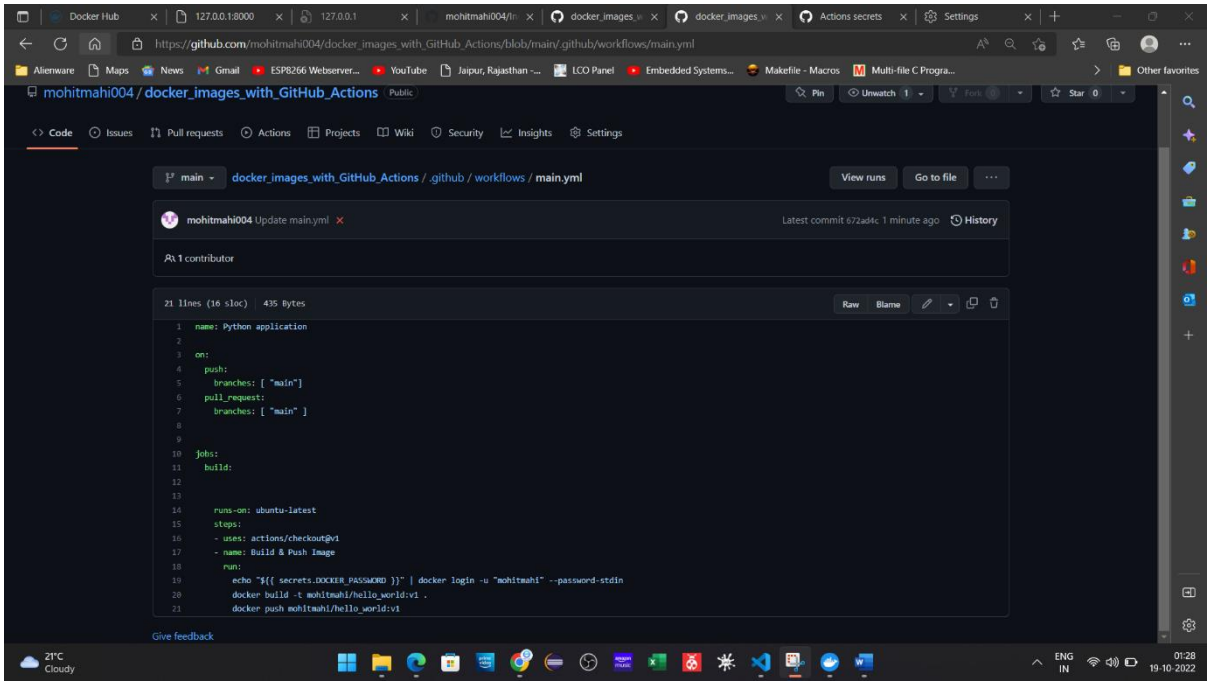
Automate Assignment below task using github action.

1. **Build Docker Image**
2. **Push Docker Image to Docker hub.**

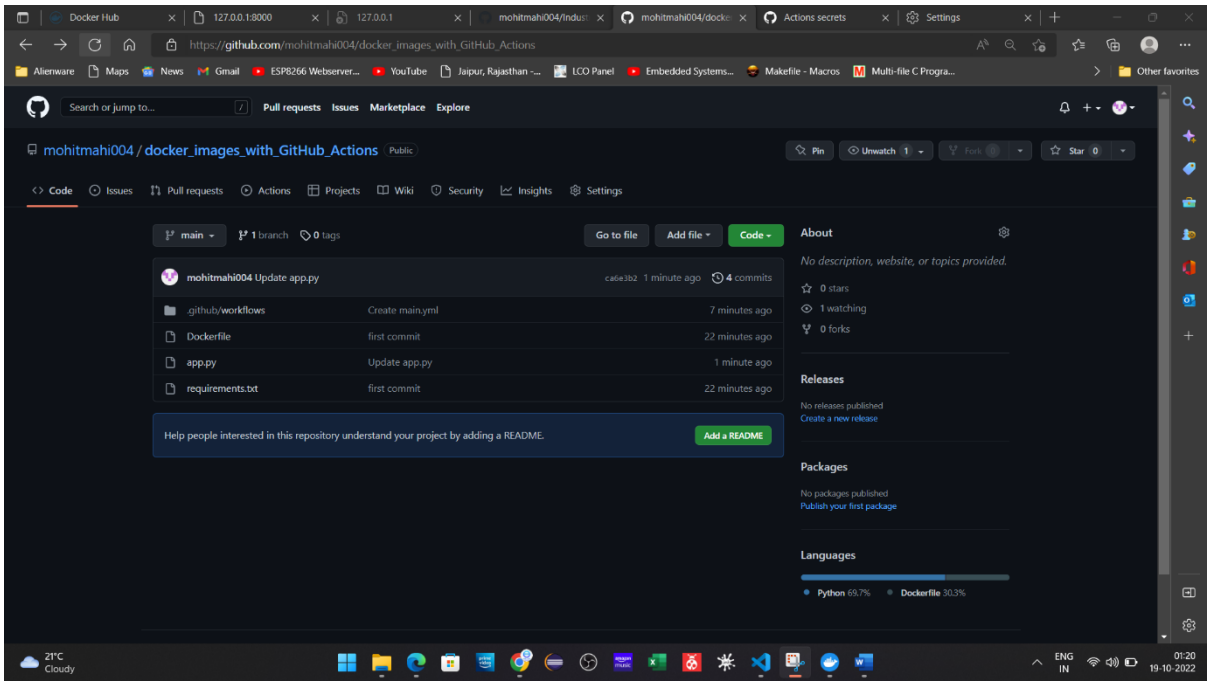
1. Adding files to github



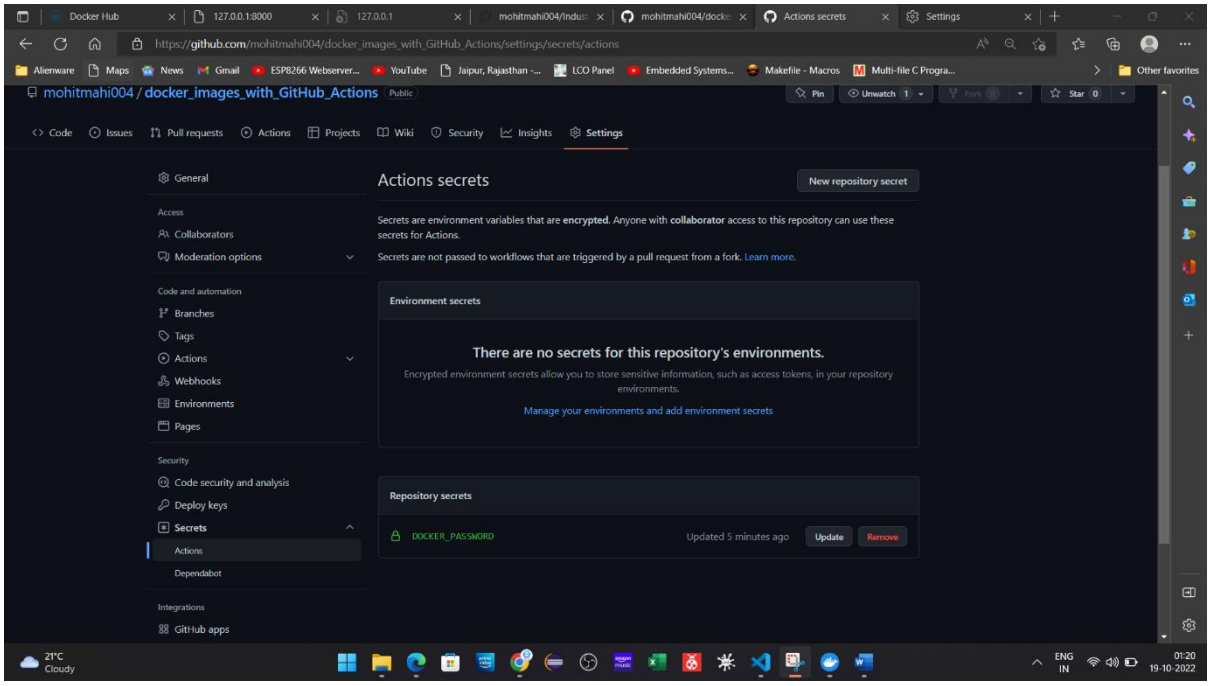
2. Adding main.yml file from GitHub Action.



3. Update the app.py file.



4. **Create a secret named as DOCKER_PASSWORD**



5. **Finally, go to Actions and see the workflow of main.yml file.**