Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
Microservices Practical 12

Aim : Create an image that sandboxes a small python application. The goal of this exercise is to create a Docker image which will run a python app.

Scenario : Docker, Inc. sponsors a dedicated team that is responsible for reviewing and publishing all Official Repositories content. This team works in collaboration with upstream software maintainers, security experts, and the broader Docker community. These are not prefixed by an organization or user name. In the list of images above, the python, node, alpine and nginx images are official (base) images.

User images are images created and shared by users like you. They build on base images and add additional functionality. Typically these are formatted as user/image-name. The user value in the image name is your Docker Store user or organization name. Hence,

Tasks, codes and screenshots :

1. **Create a Python app that displays some data.**

Python simple app code :

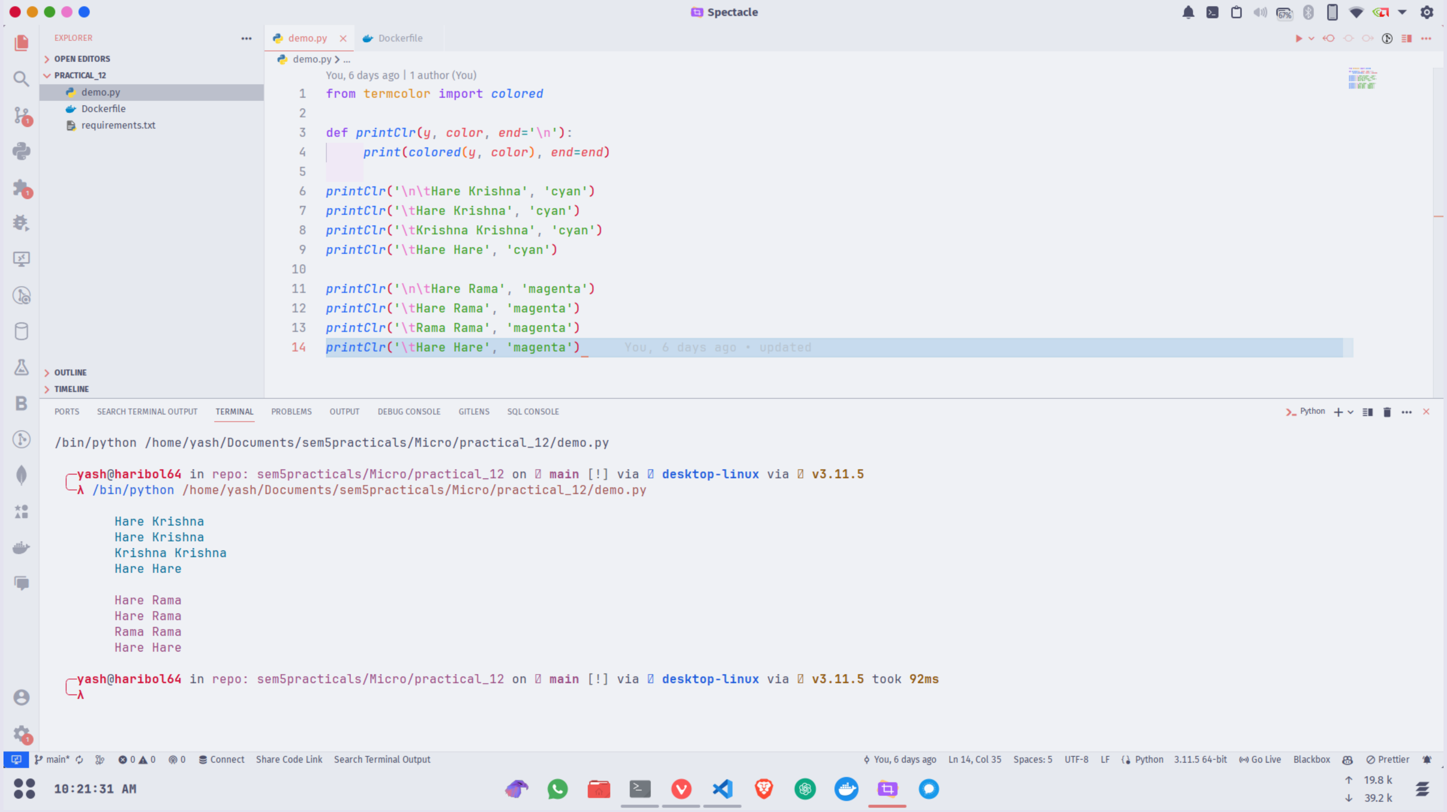
```
from termcolor import colored


def printClr(y, color, end='\n'):
print(colored(y, color), end=end)
printClr('\n\tHare Krishna', 'cyan')
printClr('\tHare Krishna', 'cyan')
printClr('\tKrishna Krishna', 'cyan')
printClr('\tHare Hare', 'cyan')


printClr('\n\tHare Rama', 'magenta')
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
Microservices Practical 12

```
printClr('\tHare Rama', 'magenta')

printClr('\tRama Rama', 'magenta')

printClr('\tHare Hare', 'magenta')
```

To install requirements, only required here is **termcolor** module, so manually requirements. txt can be made to reduce data and space usage of unnecessary modules (or using command pip freeze > requirements.txt, for all modules of the current system) :

```
termcolor==2.3.0
```



```
from termcolor import colored

def printClr(y, color, end='\n'):
     print(colored(y, color), end=end)

printClr('\n\tHare Krishna', 'cyan')
printClr('\tHare Krishna', 'cyan')
printClr('\tKrishna Krishna', 'cyan')
printClr('\tHare Hare', 'cyan')

printClr('\n\tHare Rama', 'magenta')
printClr('\tHare Rama', 'magenta')
printClr('\tRama Rama', 'magenta')
printClr('\tHare Hare', 'magenta')
```

```
/bin/python /home/yash/Documents/sem5practicals/Micro/practical_12/demo.py

 ┌yash@haribol64 in repo: sem5practicals/Micro/practical_12 on main [!] via desktop-linux via v3.11.5
 └λ /bin/python /home/yash/Documents/sem5practicals/Micro/practical_12/demo.py

        Hare Krishna
        Hare Krishna
        Krishna Krishna
        Hare Hare

        Hare Rama
        Hare Rama
        Rama Rama
        Hare Hare

 ┌yash@haribol64 in repo: sem5practicals/Micro/practical_12 on main [!] via desktop-linux via v3.11.5 took 92ms
 └λ
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
Microservices Practical 12

2. Write a Dockerfile.

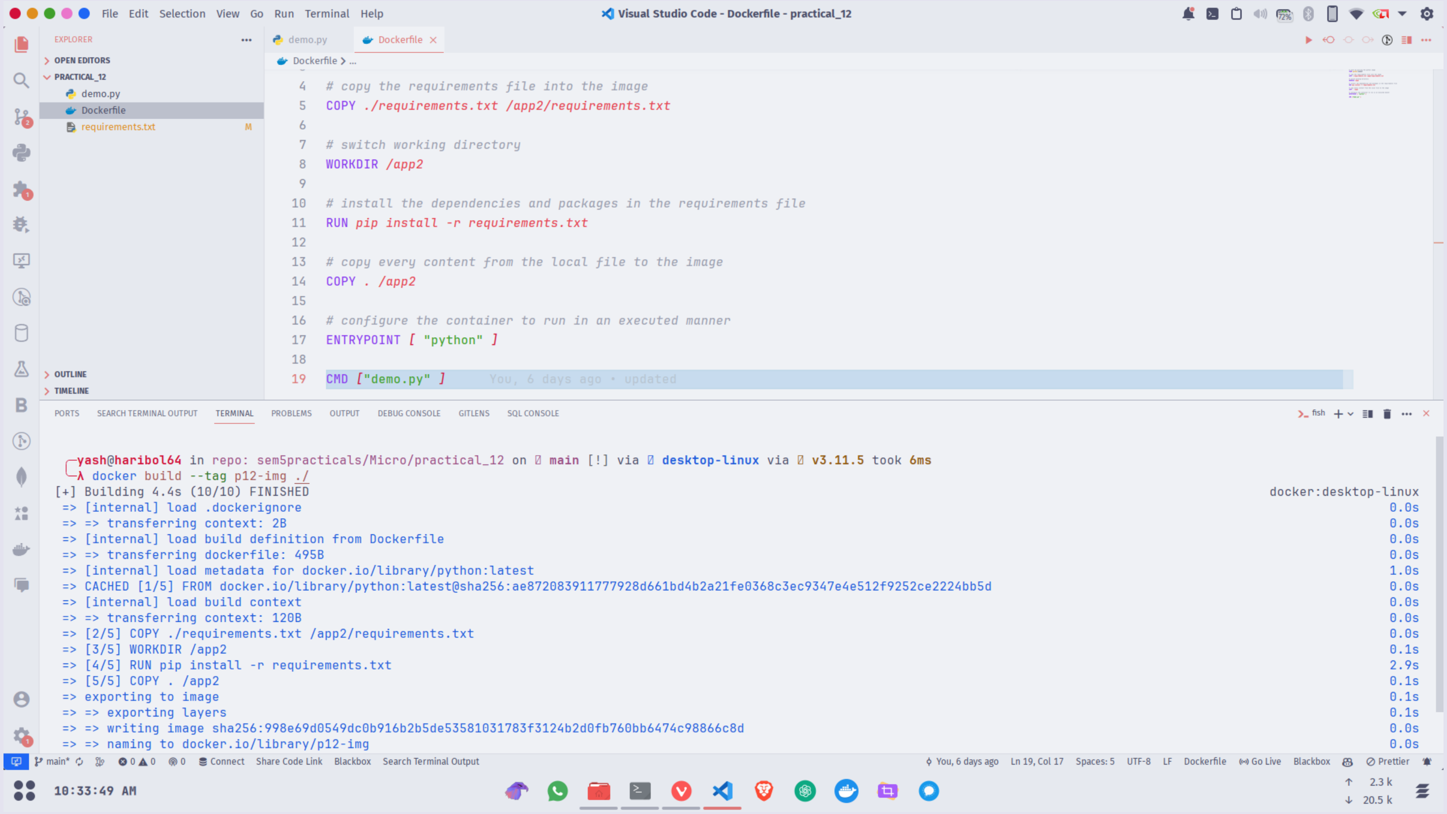
Dockerfile :

```
# start by pulling the python image

FROM python:latest


# copy the requirements file into the image

COPY ./requirements.txt /app2/requirements.txt


# switch working directory

WORKDIR /app2


# install the dependencies and packages in the requirements file

RUN pip install -r requirements.txt


# copy every content from the local file to the image

COPY . /app2


# configure the container to run in an executed manner

ENTRYPOINT [ "python" ]


CMD ["demo.py" ]
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
Microservices Practical 12

3. Build the image.

Command : docker build –tag <name> <directory>



File Edit Selection View Go Run Terminal Help
Visual Studio Code - Dockerfile - practical_12
EXPLORER
OPEN EDITORS
PRACTICAL_12
demo.py
Dockerfile
requirements.txt
OUTLINE
TIMELINE
PORTS SEARCH TERMINAL OUTPUT TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SQL CONSOLE
main* Connect Share Code Link Blackbox Search Terminal Output
You, 6 days ago Ln 19, Col 17 Spaces: 5 UTF-8 LF Dockerfile Go Live Blackbox Prettier
10:33:49 AM

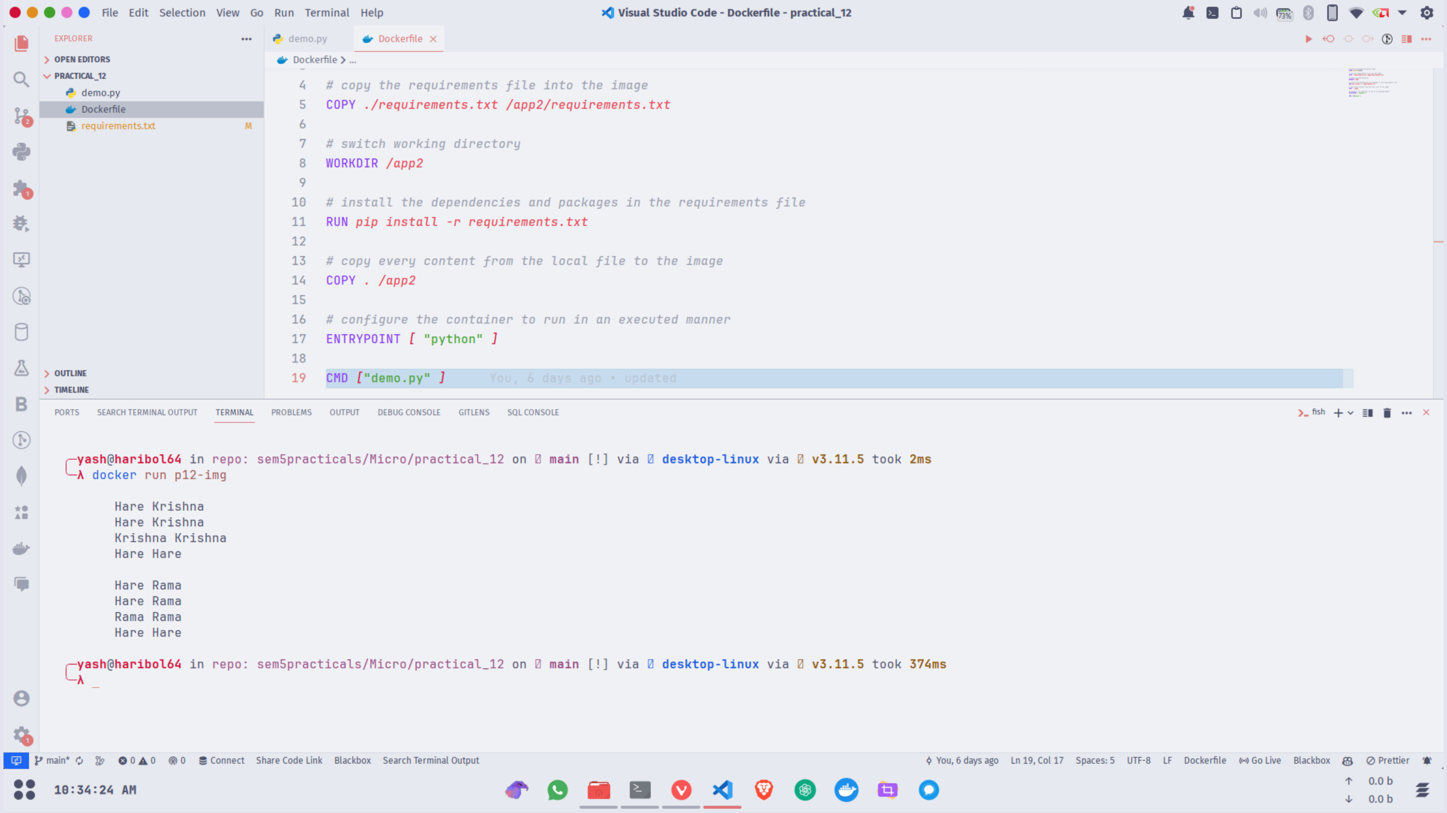
```
# copy the requirements file into the image
COPY ./requirements.txt /app2/requirements.txt

# switch working directory
WORKDIR /app2

# install the dependencies and packages in the requirements file
RUN pip install -r requirements.txt

# copy every content from the local file to the image
COPY . /app2

# configure the container to run in an executed manner
ENTRYPOINT [ "python" ]

CMD ["demo.py" ]        You, 6 days ago • updated
```

```
┌yash@haribol64 in repo: sem5practicals/Micro/practical_12 on  main [!] via  desktop-linux via  v3.11.5 took 6ms
└λ docker build --tag p12-img ./
[+] Building 4.4s (10/10) FINISHED                                                                docker:desktop-linux
 => [internal] load .dockerignore                                                                                0.0s
 => => transferring context: 2B                                                                                  0.0s
 => [internal] load build definition from Dockerfile                                                             0.0s
 => => transferring dockerfile: 495B                                                                             0.0s
 => [internal] load metadata for docker.io/library/python:latest                                                 1.0s
 => CACHED [1/5] FROM docker.io/library/python:latest@sha256:ae872083911777928d661bd4b2a21fe0368c3ec9347e4e512f9252ce2224bb5d  0.0s
 => [internal] load build context                                                                                0.0s
 => => transferring context: 120B                                                                                0.0s
 => [2/5] COPY ./requirements.txt /app2/requirements.txt                                                         0.0s
 => [3/5] WORKDIR /app2                                                                                          0.1s
 => [4/5] RUN pip install -r requirements.txt                                                                    2.9s
 => [5/5] COPY . /app2                                                                                           0.1s
 => exporting to image                                                                                           0.1s
 => => exporting layers                                                                                          0.1s
 => => writing image sha256:998e69d0549dc0b916b2b5de53581031783f3124b2d0fb760bb6474c98866c8d                     0.0s
 => => naming to docker.io/library/p12-img                                                                       0.0s
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 51
Microservices Practical 12

4. Run your image.

Command : docker run <image-name>



```dockerfile
# copy the requirements file into the image
COPY ./requirements.txt /app2/requirements.txt

# switch working directory
WORKDIR /app2

# install the dependencies and packages in the requirements file
RUN pip install -r requirements.txt

# copy every content from the local file to the image
COPY . /app2

# configure the container to run in an executed manner
ENTRYPOINT [ "python" ]

CMD ["demo.py" ]
```

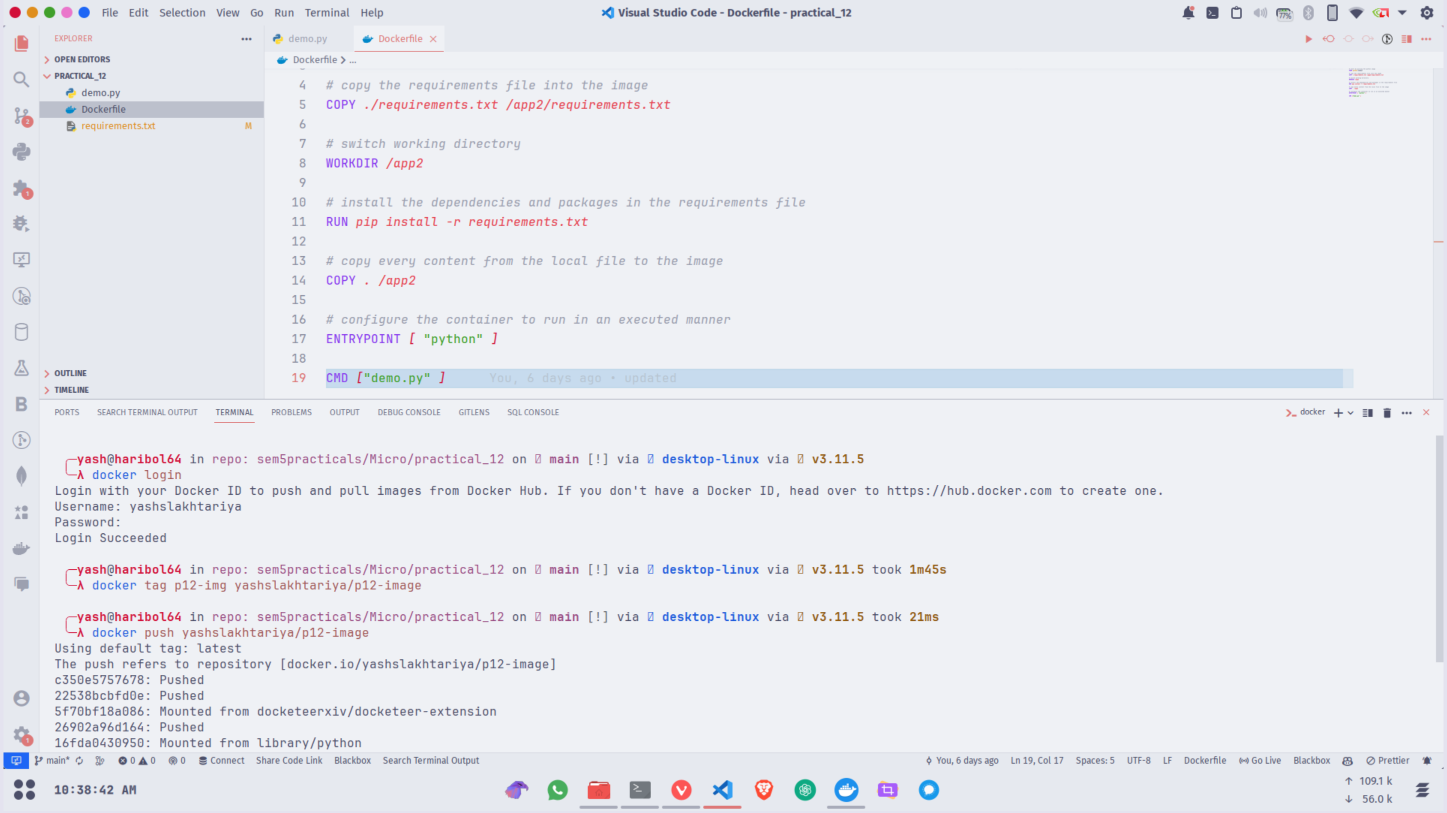
```
╭─yash@haribol64 in repo: sem5practicals/Micro/practical_12 on  main [!] via  desktop-linux via  v3.11.5 took 2ms
╰─λ docker run p12-img

    Hare Krishna
    Hare Krishna
    Krishna Krishna
    Hare Hare

    Hare Rama
    Hare Rama
    Rama Rama
    Hare Hare

╭─yash@haribol64 in repo: sem5practicals/Micro/practical_12 on  main [!] via  desktop-linux via  v3.11.5 took 374ms
╰─λ _
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 12

5. Push your image.

Commands :

```
docker login

docker tag p12-img yashslakhtariya/p12-image

docker push yashslakhtariya/p12-image
```



Visual Studio Code - Dockerfile - practical_12

```
# copy the requirements file into the image
COPY ./requirements.txt /app2/requirements.txt

# switch working directory
WORKDIR /app2

# install the dependencies and packages in the requirements file
RUN pip install -r requirements.txt

# copy every content from the local file to the image
COPY . /app2

# configure the container to run in an executed manner
ENTRYPOINT [ "python" ]

CMD ["demo.py" ]     You, 6 days ago • updated
```

```
┌yash@haribol64 in repo: sem5practicals/Micro/practical_12 on ▯ main [!] via ▯ desktop-linux via ▯ v3.11.5
└λ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: yashslakhtariya
Password:
Login Succeeded

┌yash@haribol64 in repo: sem5practicals/Micro/practical_12 on ▯ main [!] via ▯ desktop-linux via ▯ v3.11.5 took 1m45s
└λ docker tag p12-img yashslakhtariya/p12-image

┌yash@haribol64 in repo: sem5practicals/Micro/practical_12 on ▯ main [!] via ▯ desktop-linux via ▯ v3.11.5 took 21ms
└λ docker push yashslakhtariya/p12-image
Using default tag: latest
The push refers to repository [docker.io/yashslakhtariya/p12-image]
c350e5757678: Pushed
22538bcbfd0e: Pushed
5f70bf18a086: Mounted from docketeerxiv/docketeer-extension
26902a96d164: Pushed
16fda0430950: Mounted from library/python
```