Assgn 1: q.2

sol: Docker application:

steps:

1. Open docker desktop
2. Come to command prompt..
   create a directory – mkdir my-web-app

   Cd my-web-app

   Pwd:

C:\Users\jyoth\my-web-app>

```
 Directory of C:\Users\jyoth\my-web-app

16-09-2023  12:50    <DIR>          .
16-09-2023  12:09    <DIR>          ..
16-09-2023  12:52               211 app.py
16-09-2023  12:10               186 Dockerfile
16-09-2023  12:22                15 requirements.txt
16-09-2023  13:31    <DIR>          templates
               3 File(s)            412 bytes
               3 Dir(s)  668,575,043,584 bytes free

C:\Users\jyoth\my-web-app>
```

```
C:\Users\jyoth\my-web-app>cd templates

C:\Users\jyoth\my-web-app\templates>dir
 Volume in drive C is Windows-SSD
 Volume Serial Number is 6815-26D4

 Directory of C:\Users\jyoth\my-web-app\templates

16-09-2023  13:31    <DIR>          .
16-09-2023  12:50    <DIR>          ..
16-09-2023  13:30             8,907 girl.jpg
16-09-2023  12:43             9,893 image.jpg
16-09-2023  13:32               535 index.html
               3 File(s)         19,335 bytes
               2 Dir(s)  668,572,614,656 bytes free

C:\Users\jyoth\my-web-app\templates>
```

Create all files:

1. App.py

```
C:\Users\jyoth\my-web-app\templates>cd ..

C:\Users\jyoth\my-web-app>type app.py
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

if __name__ == '__main__':
    app.run(host='0.0.0.0')

C:\Users\jyoth\my-web-app>
```

2. requirements.txt

```
C:\Users\jyoth\my-web-app>type requirements.txt
Flask==2.1.1
```

3. Dockerfile

```
C:\Users\jyoth\my-web-app>type Dockerfile
FROM python:3.9-slim
WORKDIR /app
COPY app.py .
COPY requirements.txt .
COPY templates templates
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["python", "app.py"]
```

4. index.html

```
C:\Users\jyoth\my-web-app>cd templates

C:\Users\jyoth\my-web-app\templates>type index.html
<!DOCTYPE html>
<html>
<body>

<h2>Assignment-1</h2>

<p>Virtualization and Cloud Computing</p>

<p style="background-image: url('templates/girl.jpg');">
Hi, I am Ajjarapu Jyothi<br>
Roll number : M22CS007<br>
Department : Computer Science<br>
This is a Docker Application.<br>
Made from scratch..no in build images were used.<br>
My image name is my-web-app.<br>
The details of how I executed this will be shared in a read-me file<br>
Thank you<br>
Bye Bye!!!!<br>

</p>

</body>
</html>
```

To build the images and container:

run this dockerfile.. it will create an image… run the image.. then the container is created

```
C:\Users\jyoth\my-web-app>docker build -t my-web-app .
[+] Building 2.7s (11/11) FINISHED
 => [internal] load build definition from Dockerfile                                                0.0s
 => => transferring dockerfile: 32B                                                                 0.0s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim                                  2.6s
 => [1/6] FROM docker.io/library/python:3.9-slim@sha256:44b7f161ed03f85e96d423b9916cdc8cb0509fb970fd643bdbc9896d49e1cad0  0.0s
 => [internal] load build context                                                                   0.0s
 => => transferring context: 209B                                                                   0.0s
 => CACHED [2/6] WORKDIR /app                                                                       0.0s
 => CACHED [3/6] COPY app.py                                                                        0.0s
 => CACHED [4/6] COPY requirements.txt                                                              0.0s
 => CACHED [5/6] COPY templates templates                                                           0.0s
 => CACHED [6/6] RUN pip install -r requirements.txt                                                0.0s
 => exporting to image                                                                              0.0s
 => => exporting layers                                                                             0.0s
 => => writing image sha256:b46e555460cd184aafb9d8438b115acc8a6cd92d6248283795ed068266948836        0.0s
 => => naming to docker.io/library/my-web-app                                                       0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

Running the docker image:

```
C:\Users\jyoth\my-web-app>docker run -d -p 8080:5000 my-web-app
e36ff903506e63f36f8761b04452ec9ce5cb6f47f7489b3834c9d15bec9f3d83

C:\Users\jyoth\my-web-app>
```

To check if there are any errors in the application:

```
C:\Users\jyoth\my-web-app>docker run -d -p 8080:5000 my-web-app
e36ff903506e63f36f8761b04452ec9ce5cb6f47f7489b3834c9d15bec9f3d83

C:\Users\jyoth\my-web-app>docker ps
CONTAINER ID   IMAGE        COMMAND           CREATED         STATUS         PORTS                    NAMES
e36ff903506e   my-web-app   "python app.py"   2 minutes ago   Up 2 minutes   0.0.0.0:8080->5000/tcp   wizardly_ramanujan

C:\Users\jyoth\my-web-app>docker logs e36ff903506e
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
```

In the web browser: