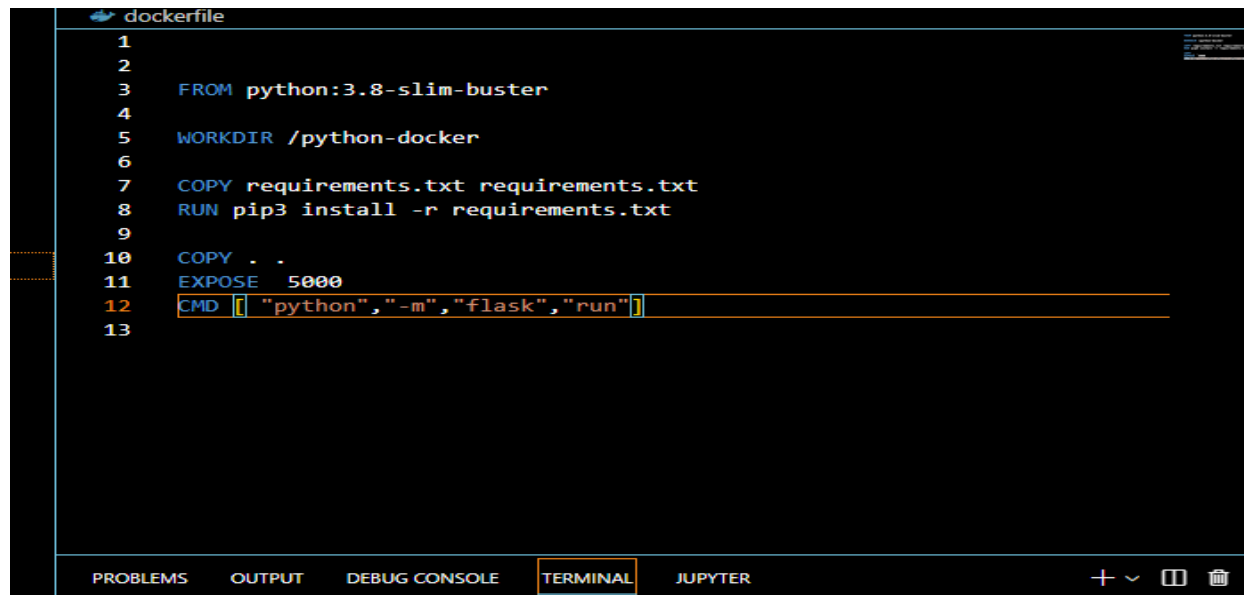


ASSIGNMENT – 4



The image shows a code editor window titled "dockerfile" with a dark theme. The code is a Dockerfile for a Python Flask application. It starts with a base image of "python:3.8-slim-buster", sets the working directory to "/python-docker", copies a "requirements.txt" file, and installs the dependencies using "pip3". It then copies the current directory's contents into the container, exposes port 5000, and sets the command to run "python -m flask run". The editor has a sidebar on the left with a file explorer showing "requirements.txt" and "Dockerfile". At the bottom, there is a panel with tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL" (which is active), and "JUPYTER". The "TERMINAL" tab shows a command prompt with the command "python -m flask run" entered.

```
1
2
3 FROM python:3.8-slim-buster
4
5 WORKDIR /python-docker
6
7 COPY requirements.txt requirements.txt
8 RUN pip3 install -r requirements.txt
9
10 COPY . .
11 EXPOSE 5000
12 CMD ["python", "-m", "flask", "run"]
13
```

operable program or batch file.

```
C:\Users\ELCOT\Desktop\flask app>docker build -t helloworld .
```

```
[+] Building 33.7s (10/10) FINISHED
```

```
=> [internal] load build definition from Dockerfile
```

```
=> => transferring dockerfile: 240B
```

```
=> [internal] load .dockerignore
```

```
=> => transferring context: 2B
```

```
=> [internal] load metadata for docker.io/library/python:3.8-slim-buster
```

```
=> [1/5] FROM docker.io/library/python:3.8-slim-buster@sha256:2faab08db0d11bb549be5b7b626ad23fcd0fe7998ad02a708381f1800a3fd5
```

```
=> [internal] load build context
```

```
=> => transferring context: 666B
```

```
=> CACHED [2/5] WORKDIR /python-docker
```

```
=> CACHED [3/5] COPY requirements.txt requirements.txt
```

```
=> CACHED [4/5] RUN pip3 install -r requirements.txt
```

```
=> [5/5] COPY . .
```

```
=> exporting to image
```

```
=> => exporting layers
```

```
=> => writing image sha256:04030a529ad4e5c30981e2c928d5c431939e9cd62a0dc4222fd73963da878f3c
```

```
=> => naming to docker.io/library/helloworld
```

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

Docker Desktop

Update to latest

Sign in

Containers

Images

Volumes

Dev Environments BETA

Extensions BETA

Add Extensions

Images on disk

Last refresh: about 1 hour ago

5 images

174 MB total size

0 Bytes / 174 MB in use

Clean up

Images

Give feedback

LOCAL

REMOTE REPOSITORIES

Search

☐ In use only

NAME ↑		TAG	IMAGE ID	CREATED	SIZE	
<none>	IN USE	<none>	441fac0f8e41	9 days ago	128.21 MB	
alpine	IN USE	latest	9c6f07244728	3 months ago	5.54 MB	
docker/getting-started	IN USE	latest	cb90f98fd791	7 months ago	28.78 MB	
helloworld		latest	04030a529ad4	1 minute ago	128.23 MB	<div> RUN</div>
sampleflaskapp	IN USE	latest	33cf23943a8c	9 days ago	128.23 MB	

Connect to Remote Content

Not connected

✓ Store and backup your images remotely

✓ Collaborate with your team

✓ Unlock vulnerability scanning for greater security

✓ Connect for free

Sign in

RAM 1.51GB CPU 0.67% Not connected to Hub

v4.13.0