[redacted]

NAME: HERAMB.R.PAWAR

SUB: SEPM

PRACTICAL: 9

AIM: TO LEARN DOCKERFILE INSTRUCTION, BUILD AN IMAGE FOR SAMPLE WEB APPLICATION USING DOCKER FILE

PROBLEM STMT:
Create a DockerFile with appropriate Instruction to build an Image for sample web application and use Docker to build and run the Image.

Theory:

Docker File:

- A DockerFile is a text File that Contains a series of Commands or Instructions
- These instructions are executed in order which they are written
- Exeating of these Instructions takes place on a base Image
- On building the DockerFile the successive action form a new Image from the base Parent Image

List of Docker Commands for Creating a Dockerfile:

Docker File Consist of specific commands that guide you on how to build specific Docker image

* FROM
* RUN
* ENTRYPOINT
* ENV
* PULL
* CMD
* ADD
* MAINTAINER

1) FROM → Creates a layer from UBUNTU:18.04
2) PULL → Add Files from Docker Repository
3) RUN → Builds your Container
4) CMD → Specifies what command to run within CONTAINER
5) ENTRYPOINT → Allows specifying Command along with Parameters
6) ADD → Command helps in Copying Data in a Docker Image
7) ENV → ENV provides default values for Variables that can be accessed within the Container
8) MAINTAINER → It declares the Author field of Images

Building an Image Using DockerFile:

```
docker build -t <dockerfile_name>
```

We can Use Docker Images to take a look at Created Images. After that we can Use " docker Container run -d <image_name>

to run container using the image

Conclusions:

Thus we learned about DockerFile, created and wrote Docker for sample Flask Web app and built a Docker Image using it

Name: Heramb Pawar **Subject**:SEPM **Roll:** 67

Practical Output:

Pre-requisite: Simple Flask Application.

App.py

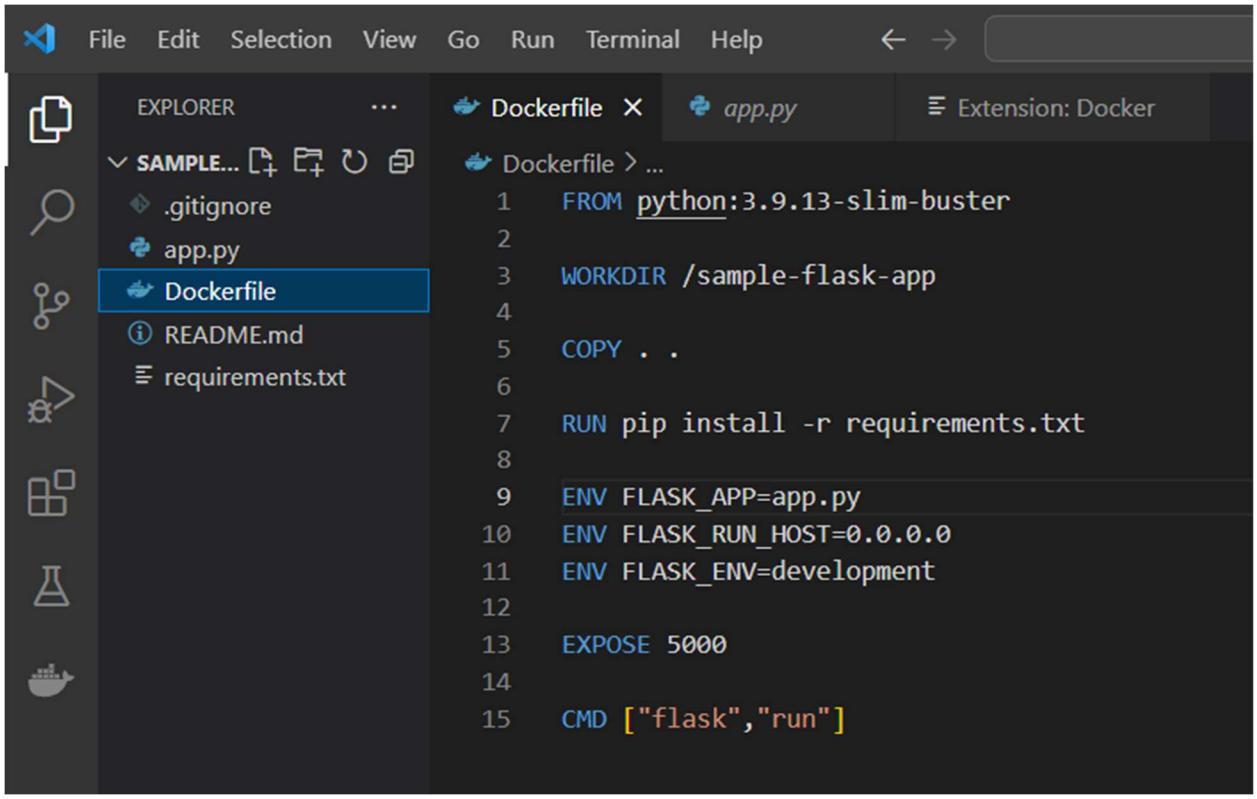
```python
from flask import Flask, jsonify

app = Flask(__name__)


@app.route('/')
def hello_world():
    return jsonify({
        "message": "Hello This Is Heramb"
    })


if __name__ == '__main__':
    app.run()
```

Creating A Docker File And Adding It In the Same Folder



Creating Image File for the Flask Application:

```
PS C:\Users\heram\Downloads\sample-flask-app-main\sample-flask-app-main> docker build  -t flask-sample .
[+] Building 5.5s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 266B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/python:3.9.13-slim-buster
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load build context
 => => transferring context: 388B
 => [1/4] FROM docker.io/library/python:3.9.13-slim-buster@sha256:e0bf67a281748c0f00c320dbe522631e92c649bef22a14f00a599c1981dac2a6
 => CACHED [2/4] WORKDIR /sample-flask-app
 => [3/4] COPY . .
```

Checking If the Image File is created or not:

```
PS C:\Users\heram\Downloads\sample-flask-app-main\sample-flask-app-main> docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
flask-sample            latest    624f8158891e   9 seconds ago    129MB
<none>                  <none>    ab01288f9f88   7 minutes ago    129MB
nginx                   latest    904b8cb13b93   2 weeks ago      142MB
redis                   latest    f9c173b0f012   2 weeks ago      117MB
docker/getting-started  latest    3e4394f6b72f   2 months ago     47MB
```

Running the container:

```
PS C:\Users\heram\Downloads\sample-flask-app-main\sample-flask-app-main> docker container run -p 5000:5000 -d flask-sample
f59de3670c996c3dd7176ccd809ce6410bcefd3175a3dc2aca7b2f1abb93b87e
```

Final Output: