Srusti G C(1RV24MC106)

Program-3

Step 1: Create folder
Mkdir program3
cd program3

Step 2: create subfolders
Write a Simple Flask App (app.py)
nano app.py

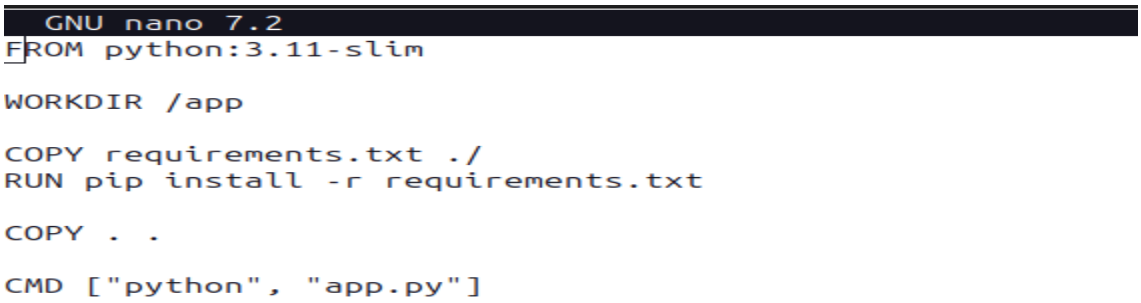
```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello from Dockerized Flask App!"

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000)
```

Step 3: create requirements.txt file
nano requirements.txt
Flask==2.3.3

```
  GNU nano 7.2
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello from  Docker!"

if __name__ == '__main__':
    # Listen on all interfaces, port 5000
    app.run(host='0.0.0.0', port=5000)
```

Step 4: create Docker file
nano Dockerfile

```
#Use official Python base image
FROM python:3.9-slim

#Set working directory inside container
WORKDIR /app

#Copy dependency file first (for better caching)
COPY requirements.txt .

#Install Python dependencies
RUN pip install --no-cache-dir -r requirements.txt

#Copy the rest of the app
COPY . .

#Expose the port Flask will run on
EXPOSE 5000

#Command to run the app
CMD ["python", "app.py"]
```



Step 5: Build the Docker Image

docker build -t flask-demo .

```
1RV24MC106_Srusti_G_C@srusti-IdeaPad-Slim-3-15IAH8:~/program-3$ docker build -t python-demo .
[+] Building 88.1s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 173B
 => [internal] load metadata for docker.io/library/python:3.11-slim
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/python:3.11-slim@sha256:fa9b525a0be0c5ae5e6f2209f4be6fdc5a15a36fed0222144d98ac0d08f876d4
 => => resolve docker.io/library/python:3.11-slim@sha256:fa9b525a0be0c5ae5e6f2209f4be6fdc5a15a36fed0222144d98ac0d08f876d4
 => => sha256:b596083aa14d47c78a652138aa9b98607585499d7c7ec343ae378f6c5770822d 1.75kB / 1.75kB
 => => sha256:870925f757415a696459255b0a20b082e914cac42efaff4424b59e21bff80d5b 5.48kB / 5.48kB
 => => sha256:d7ecded7702a5dbf6d0f79a71edc34b534d08f3051980e2c948fba72db3197fc 29.78MB / 29.78MB
 => => sha256:1ee9c106547f05aa380c4cdec2837c546439943d73d965a3fc49f228dc8be993 1.29MB / 1.29MB
 => => sha256:f002d17b63fe84a7f8a66f20cfa63aec4f6cd2a44069f05b6296b0abfcf2a8e1 14.36MB / 14.36MB
 => => sha256:fa9b525a0be0c5ae5e6f2209f4be6fdc5a15a36fed0222144d98ac0d08f876d4 10.37kB / 10.37kB
 => => sha256:65868b001a40155a1d3f5aa7f5a10ba02a7d55697301839dc047c9d549b670bc 248B / 248B
 => => extracting sha256:d7ecded7702a5dbf6d0f79a71edc34b534d08f3051980e2c948fba72db3197fc
 => => extracting sha256:1ee9c106547f05aa380c4cdec2837c546439943d73d965a3fc49f228dc8be993
 => => extracting sha256:f002d17b63fe84a7f8a66f20cfa63aec4f6cd2a44069f05b6296b0abfcf2a8e1
 => => extracting sha256:65868b001a40155a1d3f5aa7f5a10ba02a7d55697301839dc047c9d549b670bc
 => [internal] load build context
 => => transferring context: 486B
 => [2/5] WORKDIR /app
 => [3/5] COPY requirements.txt ./
 => [4/5] RUN pip install -r requirements.txt
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => writing image sha256:91780ad48cd98fa7e498bab773f58765385cf2560aa56c43cd10a8f814f307c7
 => => naming to docker.io/library/python-demo
```

Step 6: Run the Container

docker run -d -p 5000:5000 flask-demo

```
^C1RV24MC106_Srusti_G_C@srusti-IdeaPad-Slim-3-15IAH8:~/program-3$ docker run -p 5000:5000 python-demo
 * Serving Flask app 'app'
 * 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://172.18.0.2:5000
Press CTRL+C to quit
172.18.0.1 - - [05/Nov/2025 15:44:03] "GET / HTTP/1.1" 200 -
172.18.0.1 - - [05/Nov/2025 15:44:03] "GET /favicon.ico HTTP/1.1" 404 -
```

Step 7: Verify the App