DEVOPS PROGRAM 1:

Build a Docker container from a Custom Dockerfile.

PROJECT STRUCTURE:
program1/
—Dockerfile(without any extension)
—requirement.txt
—-python file app.py

Step 1: Create Project Directory

mkdir program1

Creates a new directory named program1

Step 2: Navigate to the Project Directory

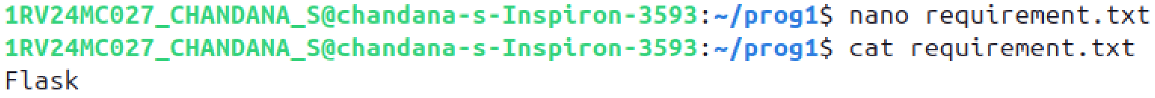
cd program1

Changes the working directory to program1, where all files will be created.

Step 3: Create and Edit requirement.txt

nano requirement.txt

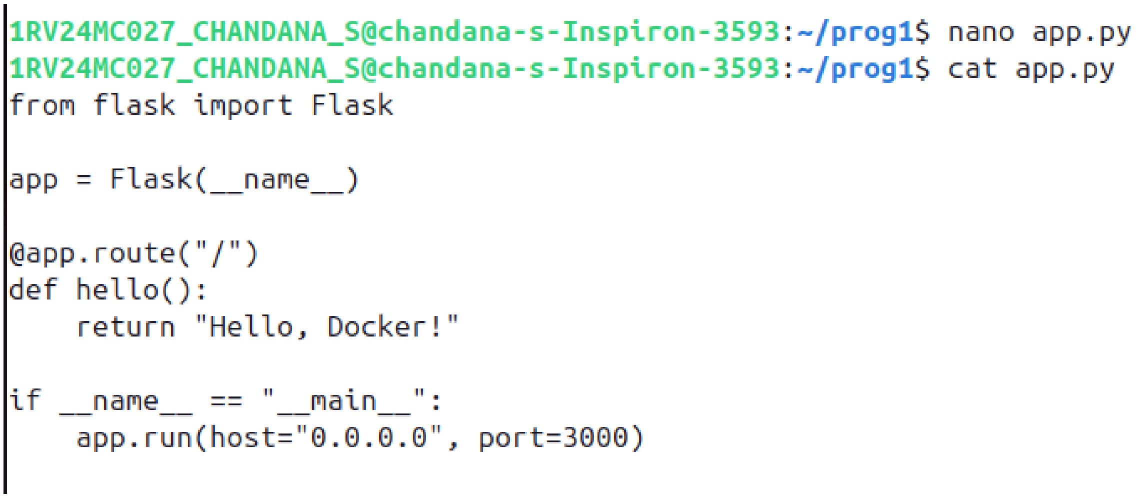
Flask



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ nano requirement.txt
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ cat requirement.txt
Flask
```

Step 4: Create and Edit app.py

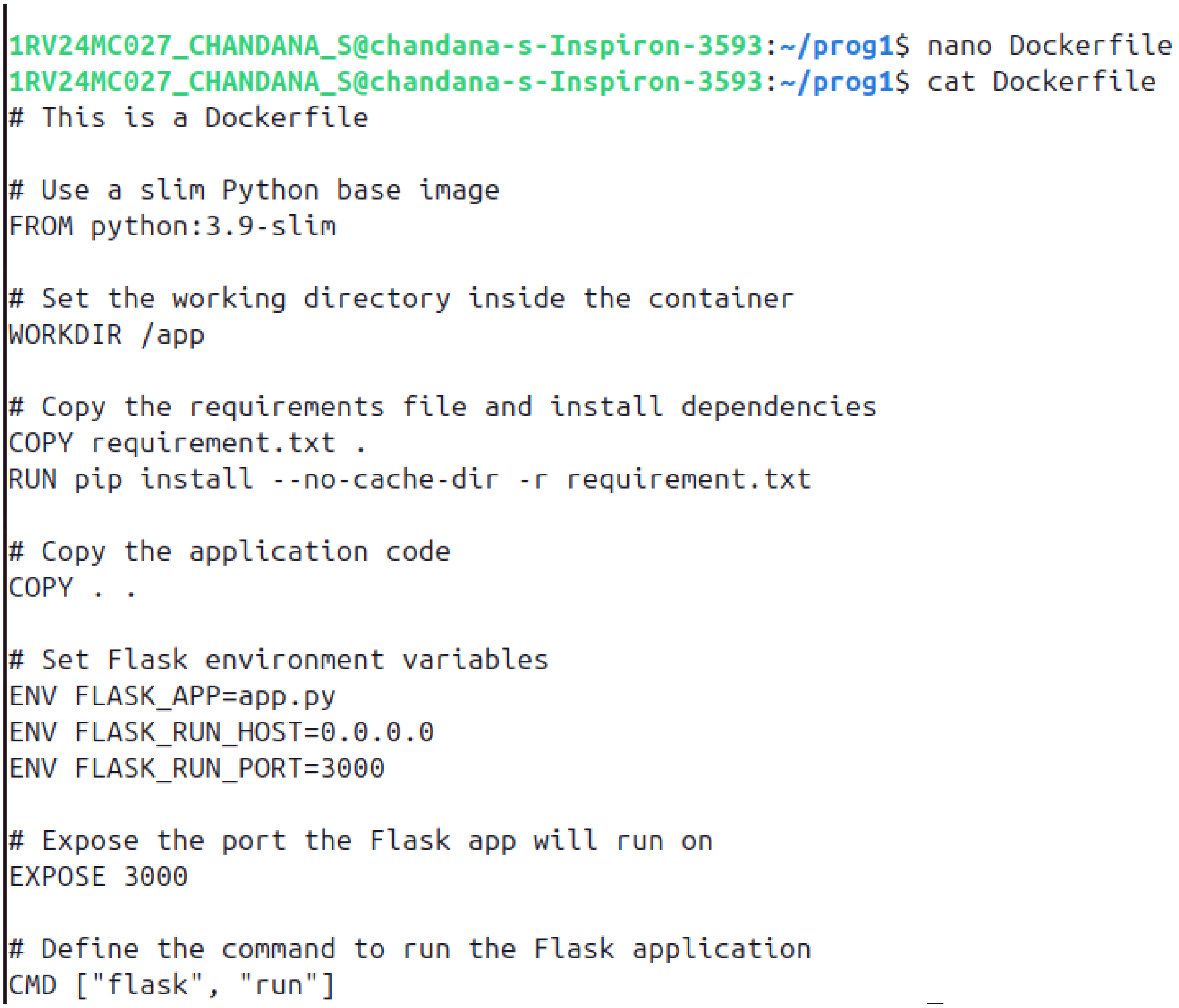
nano app.py



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ nano app.py
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ cat app.py
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, Docker!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=3000)
```

Step 5: Create and Edit Dockerfile

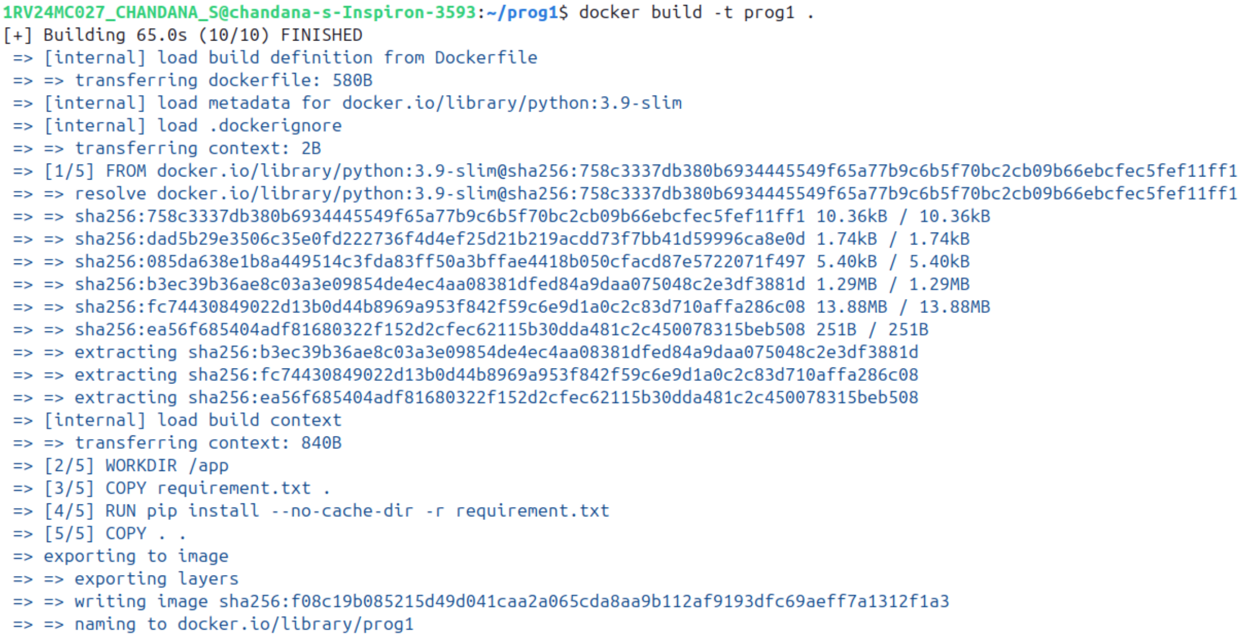
nano Dockerfile



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ nano Dockerfile
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ cat Dockerfile
# This is a Dockerfile

# Use a slim Python base image
FROM python:3.9-slim

# Set the working directory inside the container
WORKDIR /app

# Copy the requirements file and install dependencies
COPY requirement.txt .
RUN pip install --no-cache-dir -r requirement.txt

# Copy the application code
COPY . .

# Set Flask environment variables
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=3000

# Expose the port the Flask app will run on
EXPOSE 3000

# Define the command to run the Flask application
CMD ["flask", "run"]
```

Step 6: Build the Docker Image

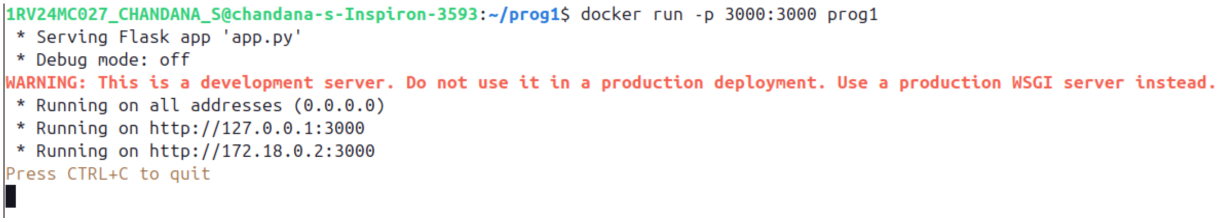
docker build -t prog1 .



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker build -t prog1 .
[+] Building 65.0s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 580B
 => [internal] load metadata for docker.io/library/python:3.9-slim
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/python:3.9-slim@sha256:758c3337db380b6934445549f65a77b9c6b5f70bc2cb09b66ebcfec5fef11ff1
 => => resolve docker.io/library/python:3.9-slim@sha256:758c3337db380b6934445549f65a77b9c6b5f70bc2cb09b66ebcfec5fef11ff1
 => => sha256:758c3337db380b6934445549f65a77b9c6b5f70bc2cb09b66ebcfec5fef11ff1 10.36kB / 10.36kB
 => => sha256:dad5b29e3506c35e0fd222736f4d4ef25d21b219acdd73f7bb41d59996ca8e0d 1.74kB / 1.74kB
 => => sha256:085da638e1b8a449514c3fda83ff50a3bffae4418b050cfacd87e5722071f497 5.40kB / 5.40kB
 => => sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d 1.29MB / 1.29MB
 => => sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08 13.88MB / 13.88MB
 => => sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508 251B / 251B
 => => extracting sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d
 => => extracting sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08
 => => extracting sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508
 => [internal] load build context
 => => transferring context: 840B
 => [2/5] WORKDIR /app
 => [3/5] COPY requirement.txt .
 => [4/5] RUN pip install --no-cache-dir -r requirement.txt
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => writing image sha256:f08c19b085215d49d041caa2a065cda8aa9b112af9193dfc69aeff7a1312f1a3
 => => naming to docker.io/library/prog1
```

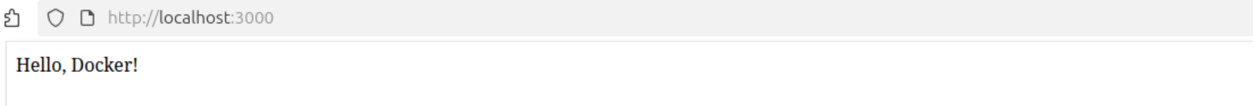
Step 7: Run the Docker Container

docker run -p 3000:3000 prog1



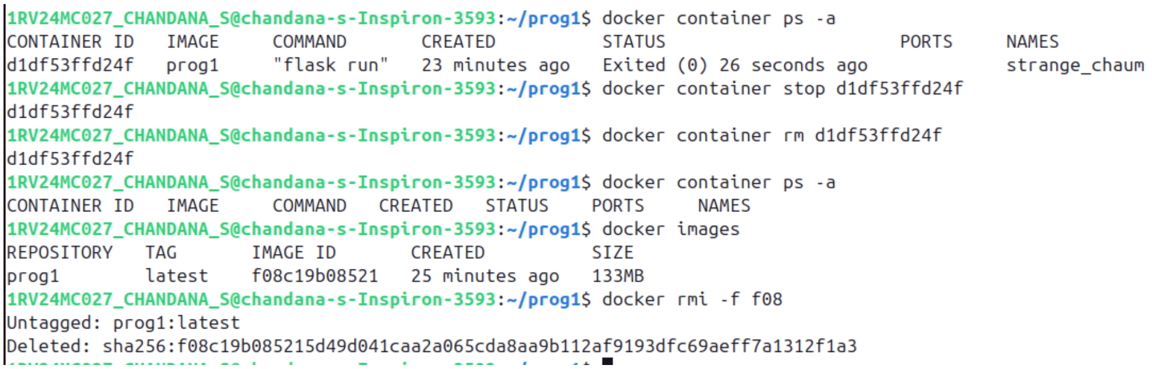
```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker run -p 3000:3000 prog1
 * Serving Flask app 'app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:3000
 * Running on http://172.18.0.2:3000
Press CTRL+C to quit
```

Step 8:Check the output



Hello, Docker!

Step 9: Verify Running Container,stop the container,remove the container



```
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker container ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS                      PORTS     NAMES
d1df53ffd24f   prog1     "flask run"   23 minutes ago   Exited (0) 26 seconds ago             strange_chaum
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker container stop d1df53ffd24f
d1df53ffd24f
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker container rm d1df53ffd24f
d1df53ffd24f
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker container ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
prog1        latest    f08c19b08521   25 minutes ago   133MB
1RV24MC027_CHANDANA_S@chandana-s-Inspiron-3593:~/prog1$ docker rmi -f f08
Untagged: prog1:latest
Deleted: sha256:f08c19b085215d49d041caa2a065cda8aa9b112af9193dfc69aeff7a1312f1a3
```