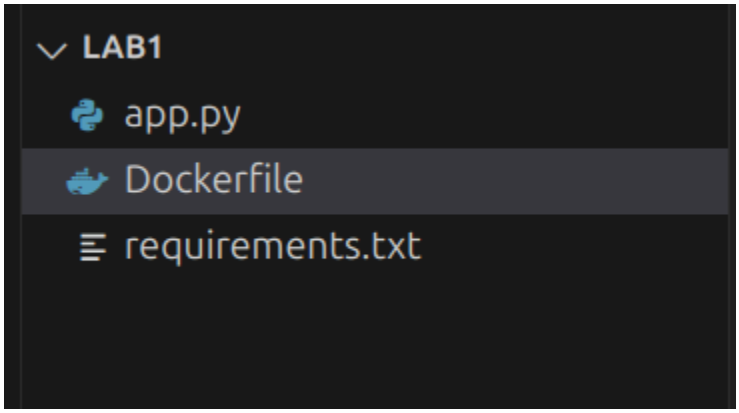
Program - 1
Build a Docker Container from a Custom Dockerfile

Project Structure:

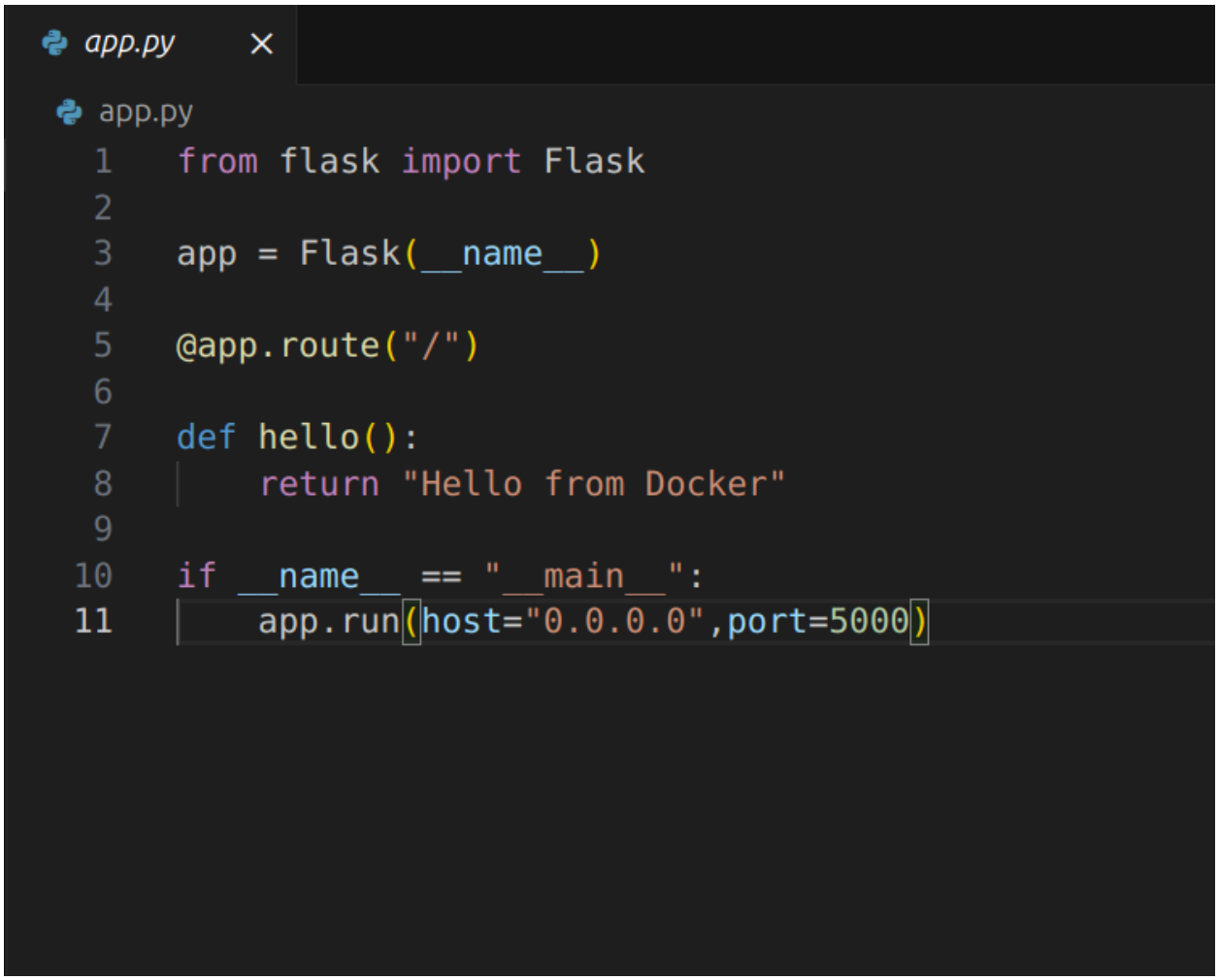


Step 1: Create the `Dockerfile`

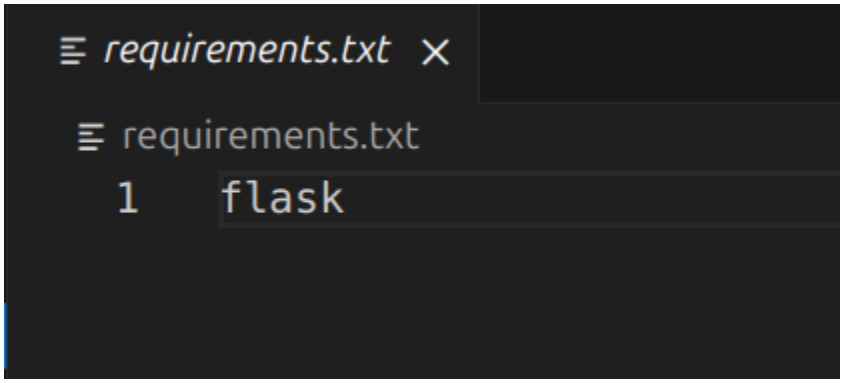
Dockerfile

```
1   FROM python:3.9-slim
2
3   WORKDIR /app
4
5   COPY requirements.txt .
6
7   RUN pip install --no-cache-dir -r requirements.txt
8
9   COPY . .
10
11  ENV FLASK_APP=app
12  ENV FLASK_RUN_HOST=0.0.0.0
13  ENV FLASK_RUN_PORT=5000
14
15  EXPOSE 5000
16
17  CMD ["flask", "run"]
18
```

Step 2: Create the Python file app.py



Step 3: Create the simple text file requirements.txt

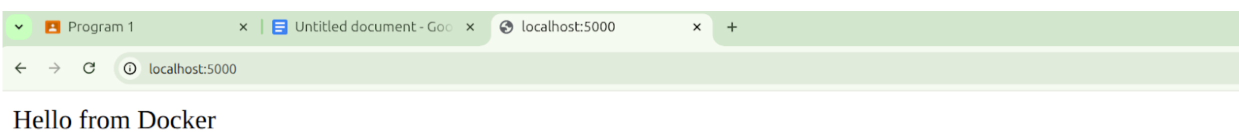


Step 4: Execute the Docker Build command

```
docker build -t lab1 .
```

Step 5: Run the Docker Container

docker run -d -p 5000:5000 --name lab1image lab1

```
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
rv24mc094_sangeethan@sangeetha-n-ThinkPad-X1-Carbon-6th:~/Desktop/Devops_lab/lab1$ docker build -t lab1 .
[+] Building 28.9s (10/10) FINISHED                                                              docker:default
 => [internal] load build definition from Dockerfile                                                       0.1s
 => => transferring dockerfile: 266B                                                                       0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim                                         5.4s
 => [internal] load .dockerignore                                                                          0.0s
 => => transferring context: 2B                                                                            0.0s
 => [internal] load build context                                                                          0.0s
 => => transferring context: 93B                                                                           0.0s
 => [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b   6.6s
 => => resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b   0.0s
 => => sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d 1.29MB / 1.29MB             1.2s
 => => sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b 10.36kB / 10.36kB           0.0s
 => => sha256:dad5b29e3506c35e0fd222736f4d4ef25d21b219acdd73f7bb41d59996ca8e0d 1.74kB / 1.74kB             0.0s
 => => sha256:085da638e1b8a449514c3fda83ff50a3bffae4418b050cfacd87e5722071f497 5.40kB / 5.40kB             0.0s
 => => sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08 13.88MB / 13.88MB           3.6s
 => => sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508 251B / 251B                 1.6s
 => => extracting sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d                  0.4s
 => => extracting sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08                  2.4s
 => => extracting sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508                  0.0s
 => [2/5] WORKDIR /app                                                                                     0.4s
 => [3/5] COPY requirements.txt .                                                                          0.2s
 => [4/5] RUN pip install --no-cache-dir -r requirements.txt                                              15.0s
 => [5/5] COPY . .                                                                                         0.2s
 => exporting to image                                                                                     0.8s
 => => exporting layers                                                                                    0.7s
 => => writing image sha256:863df0d6b8294964b5eeb7f85c54e6443ed70c89b3d95156f13270c931862ed8               0.0s
 => => naming to docker.io/library/lab1                                                                    0.0s
rv24mc094_sangeethan@sangeetha-n-ThinkPad-X1-Carbon-6th:~/Desktop/Devops_lab/lab1$ docker run -p 5000:5000 lab1
 * 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://172.17.0.2:5000
Press CTRL+C to quit
172.17.0.1 - - [05/Nov/2025 11:39:25] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [05/Nov/2025 11:39:25] "GET /favicon.ico HTTP/1.1" 404 -
```

Step 6: Test the Container