

## Experiment 9

### Devops Lab

Chaudhary Tuba

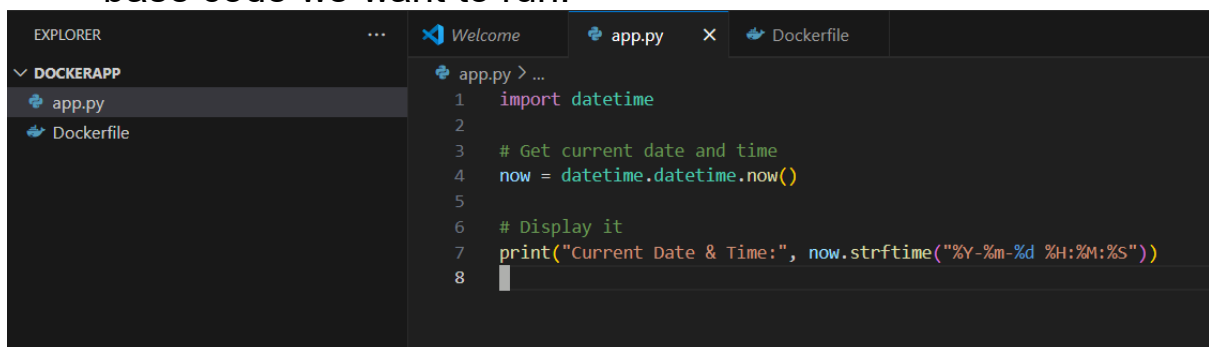
231410

TE IT

Aim: Learn Docker File Instructions, build an image for a sample application using Docker File. Learn Docker File Instructions, build an image for a sample application using Docker File.

Steps:

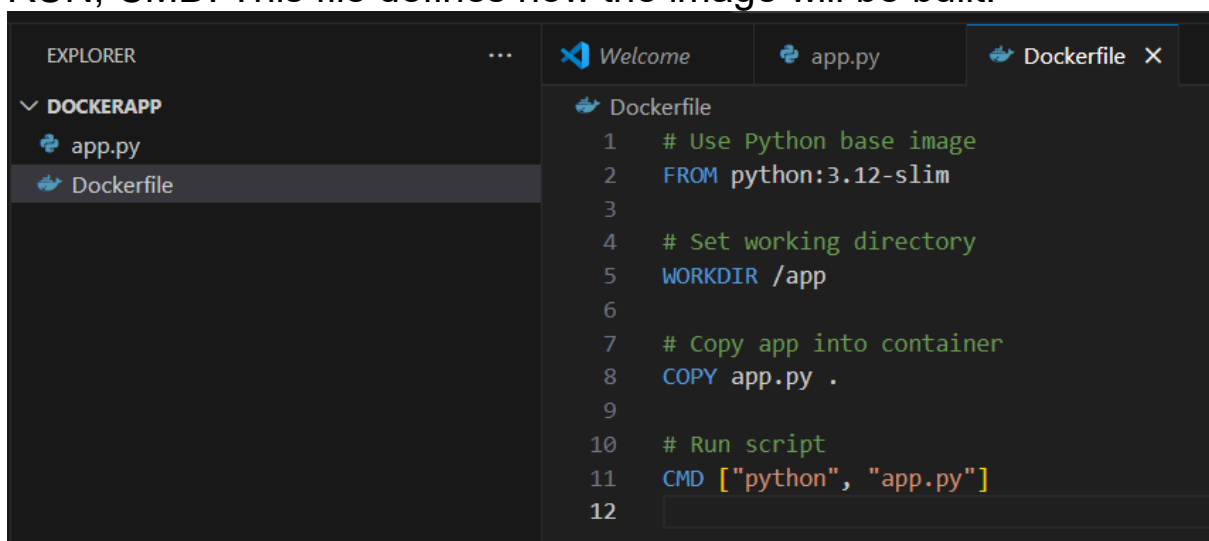
1. Create a simple application file (for example app.py or index.html) that will be deployed inside Docker. This is the base code we want to run.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project named 'DOCKERAPP' containing two files: 'app.py' and 'Dockerfile'. The 'app.py' file is selected and its content is displayed in the main editor. The code in 'app.py' is as follows:

```
1 import datetime
2
3 # Get current date and time
4 now = datetime.datetime.now()
5
6 # Display it
7 print("Current Date & Time:", now.strftime("%Y-%m-%d %H:%M:%S"))
8
```

2. Write a Docker file with necessary instructions like FROM, COPY, RUN, CMD. This file defines how the image will be built.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows the same project 'DOCKERAPP' with 'app.py' and 'Dockerfile'. The 'Dockerfile' file is now selected and its content is displayed in the main editor. The code in 'Dockerfile' is as follows:

```
1 # Use Python base image
2 FROM python:3.12-slim
3
4 # Set working directory
5 WORKDIR /app
6
7 # Copy app into container
8 COPY app.py .
9
10 # Run script
11 CMD ["python", "app.py"]
12
```




3. Open terminal and Run the command **docker build -t myapp** to build the image.

```
PS C:\Users\chaud\OneDrive\Documents\dockerapp> docker build -t my-docker-app .
>>
[+] Building 14.0s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 214B                             0.0s
=> [internal] load metadata for docker.io/library/python:3.12-slim 6.1s
=> [auth] library/python:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                   0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 196B                                 0.0s
=> [1/3] FROM docker.io/library/python:3.12-slim@sha256:abc799c7ee22b0d66f46c367643088a35e048bbab 7.2s
=> => resolve docker.io/library/python:3.12-slim@sha256:abc799c7ee22b0d66f46c367643088a35e048bbab 0.0s
=> => sha256:debe34bebc01f52b9f6e7fc26f1f74708de668b6088d586352067e5d881087d8 249B / 249B 0.7s
=> => sha256:48b120ca37b5b627d81d3cb2e63d808078495be97454db590bba47194cb017ee 12.11MB / 12.11MB 2.2s
=> => sha256:ce1261c6d567efa8e3b457673eeeb474a0a8066df6bb95ca9a6a94a31e219dd3 29.77MB / 29.77MB 4.5s
=> => transferring dockerfile: 214B                             0.0s
=> [internal] load metadata for docker.io/library/python:3.12-slim 6.1s
=> [auth] library/python:pull token for registry-1.docker.io    0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                   0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 196B                                 0.0s
```

4. Run a container from the image using **docker run my-docker-app**. This will start the application inside Docker.

```
=> => unpacking to docker.io/library/my-docker-app:latest
PS C:\Users\chaud\OneDrive\Documents\dockerapp> docker run my-docker-app
>>
Current Date & Time: 2025-09-26 16:07:37
PS C:\Users\chaud\OneDrive\Documents\dockerapp> 
```

5. verify the image is created successfully by running docker images. You should see your image in the list

<input type="checkbox"/>		python	latest	2deb0891ec3f	1 month ago	1.6 GB
<input type="checkbox"/>		python	3.12	1cb6108b64a4	2 months ago	1.59 GB
<input type="checkbox"/>		my-docker-app	latest	95e69a20d052	6 minutes ago	176.91 MB

6. you can also run the file inside the docker by going to images → my-docker-app and click on run → you can see the output.

