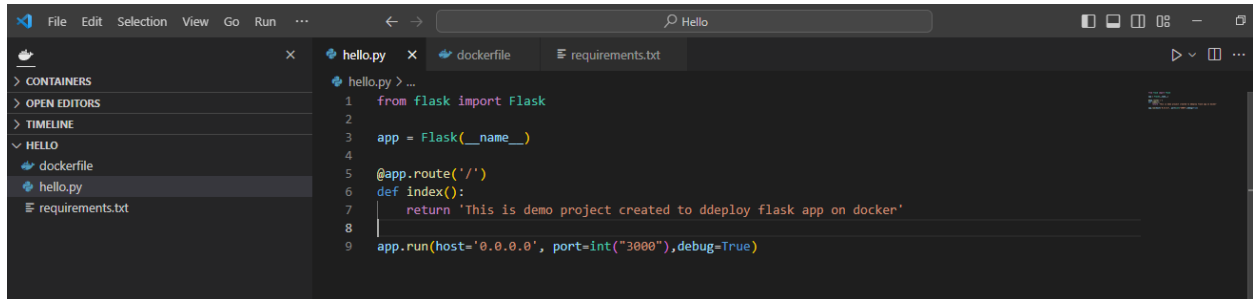


DOCKER ASSIGNMENT

1. Hello world using flask and deployment using Docker

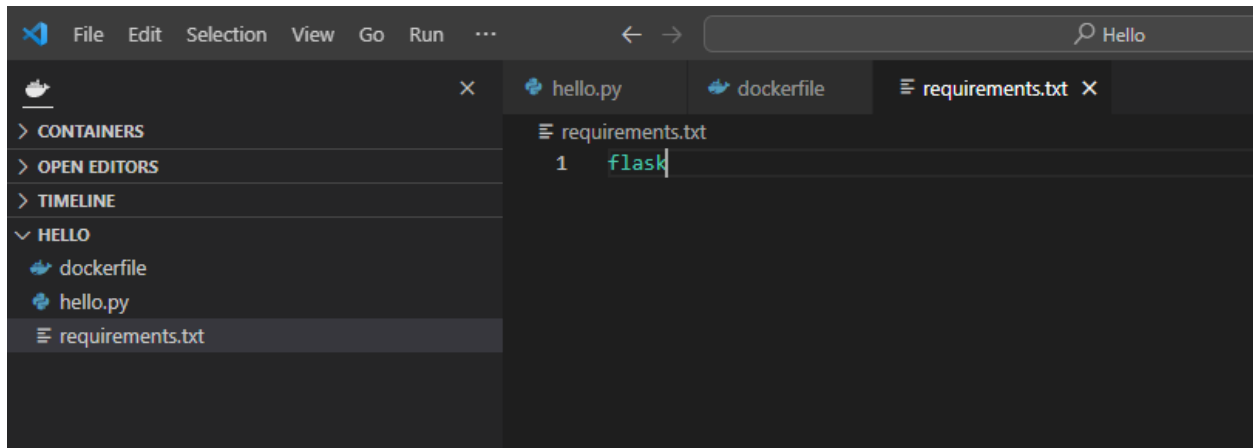
STEP 1: CREATED APP



A screenshot of the Visual Studio Code editor interface. The left sidebar shows the 'HELLO' folder with files 'dockerfile', 'hello.py', and 'requirements.txt'. The main editor window displays the 'hello.py' file with the following Python code:

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route('/')
6 def index():
7     return 'This is demo project created to ddeploy flask app on docker'
8
9 app.run(host='0.0.0.0', port=int("3000"), debug=True)
```

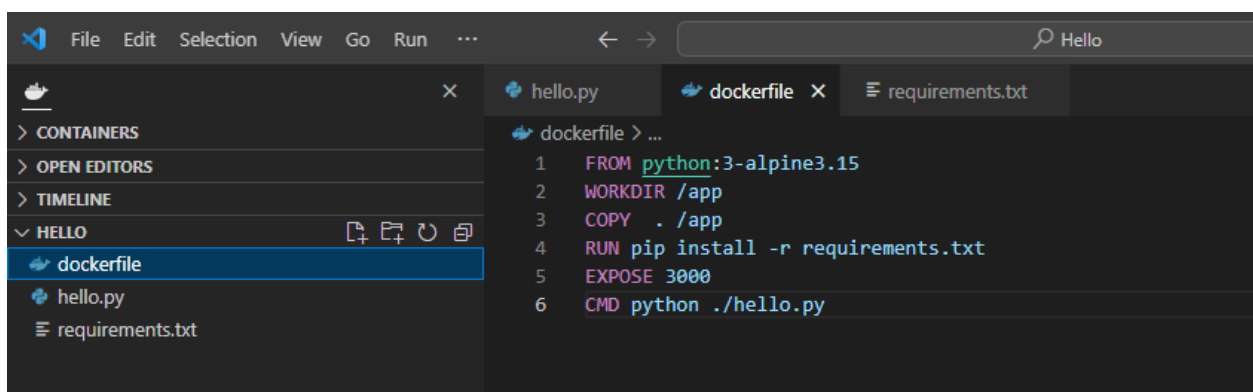
STEP 2: CREATED REQUIREMENT FILE



A screenshot of the Visual Studio Code editor interface. The left sidebar shows the 'HELLO' folder with files 'dockerfile', 'hello.py', and 'requirements.txt'. The main editor window displays the 'requirements.txt' file with the following content:

```
1 flask
```

STEP 3: CREATED DOCKER FILE



A screenshot of the Visual Studio Code editor interface. The left sidebar shows the 'HELLO' folder with files 'dockerfile', 'hello.py', and 'requirements.txt'. The main editor window displays the 'dockerfile' file with the following Dockerfile instructions:

```
1 FROM python:3-alpine3.15
2 WORKDIR /app
3 COPY . /app
4 RUN pip install -r requirements.txt
5 EXPOSE 3000
6 CMD python ./hello.py
```

STEP 4: CREATNG DOCKER IMAGE

```
File Edit Selection View Go Run ... Hello
hello.py dockerfile requirements.txt
CONTAINERS
OPEN EDITORS
TIMELINE
HELLO
dockerfile
hello.py
requirements.txt
1 FROM python:3-alpine3.15
2 WORKDIR /app
3 COPY . /app
4 RUN pip install -r requirements.txt
5 EXPOSE 3000
6 CMD python ./hello.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\admin\Desktop\Hello> docker build -t helloapp:latest .
[+] Building 25.1s (10/10) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 169B 0.3s
-> [internal] load .dockerignore
-> => transferring context: 2B 0.1s
-> [internal] load metadata for docker.io/library/python:3-alpine3.15 4.2s
[auth] library/python:pull token for registry-1.docker.io 0.0s
[1/4] FROM docker.io/library/python:3-alpine3.15 10.8s
-> resolve docker.io/library/python:3-alpine3.15 0.1s
-> sha256:1567c44f47c8136752ee4301be2e94130a5f8e1fb7a0e737ccac28fb8f66fd0f 1.65kB / 1.65kB 0.0s
-> sha256:a28968201bb92c5950a8382aa3603e273855a8cde6d73f402fb559566b1fe244 1.37kB / 1.37kB 0.0s
-> sha256:c692f3b79c36c5465dbcc4b8ed4348a03cd4285fc9419e048bce324aafe1aebb 7.03kB / 7.03kB 0.0s
-> sha256:c202fc059ff2085bf2bc9f7aeb00c093fd3db159fcc1623a0898cec97ba21d5c 13.07MB / 13.07MB 8.2s
-> sha256:0621f1afde84853b2f96ff34fc7f7460712247c01cbab483c5fa7132cf782ca 2.82MB / 2.82MB 2.4s
-> sha256:029c086462bcc047737664d4f322a7e83d923bc049e40f799bc3629c623613c4 230B / 230B 2.7s
-> sha256:28ebalc748bb36c3d442bba4d3953cfcdf798b476fac3f7c67988cc5d61a6ad 4.8s
-> extracting sha256:7dcb358f5cf3d2ed07d13dd90b2d780245d81b3017875ed3478dac69ae7fc70 0.7s
-> extracting sha256:c202fc059ff2085bf2bc9f7aeb00c093fd3db159fcc1623a0898cec97ba21d5c 1.1s
-> extracting sha256:029c086462bcc047737664d4f322a7e83d923bc049e40f799bc3629c623613c4 0.0s
-> extracting sha256:28ebalc748bb36c3d442bba4d3953cfcdf798b476fac3f7c67988cc5d61a6ad 0.5s
-> [internal] load build context 0.8s
-> => transferring context: 469B 0.7s
[2/4] WORKDIR /app 0.4s
[3/4] COPY . /app 0.2s
```

STEP 5: CHECKING DOCKER IMAGES

Docker Desktop Upgrade plan Search for local and remote images, containers, and more... Ctrl+K rohan9...

ContainersImagesVolumesDev Environments (BETA)Learning CenterExtensionsDisk usageAdd Extensions

Images Give feedback

An image is a read-only template with instructions for creating a Docker container. Learn more

LocalHubArtifactory (EARLY ACCESS)

67.41 MB / 126.45 MB in use 2 Images Last refresh: about 2 hours ago

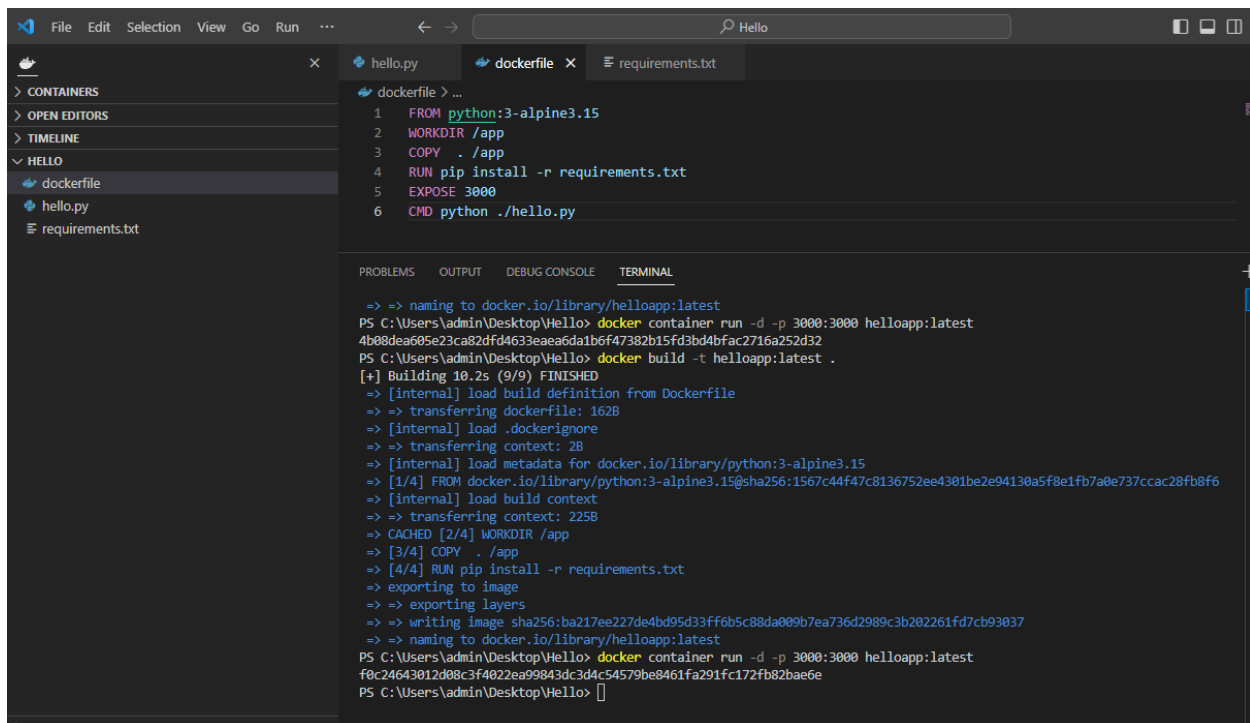
Search

Name	Tag	Status	Created	Size	Actions
helloapp ba217ee227de	latest	In use	less than a minute a	67.41 MB	
<none> 8fab6fe80fb5	<none>	Unused (dangl	3 minutes ago	67.41 MB	

```
C:\Users\admin\Desktop\Hello>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
helloapp latest ba217ee227de 11 minutes ago 67.4MB
<none> <none> 8fab6fe80fb5 13 minutes ago 67.4MB

C:\Users\admin\Desktop\Hello>
```

STEP 6: CREATING CONTAINER



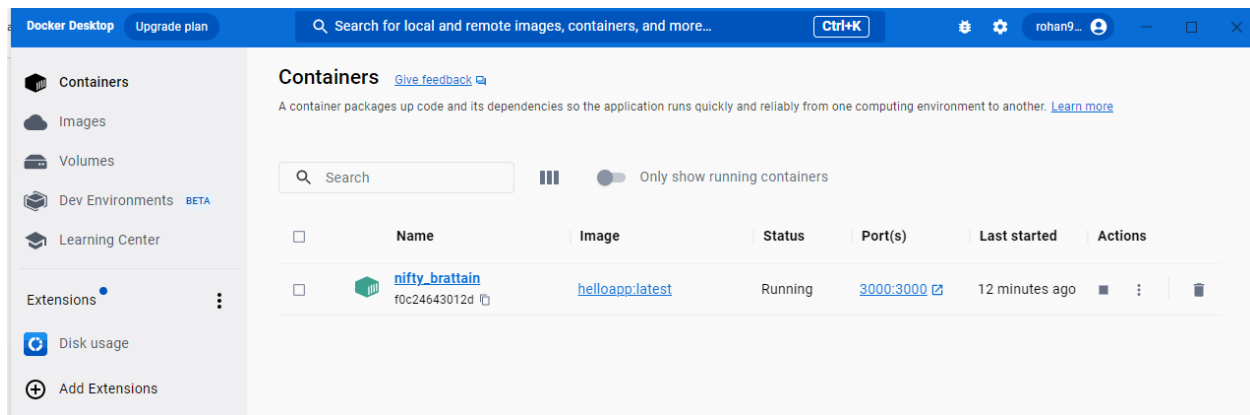
The screenshot shows the Visual Studio Code editor with a Dockerfile open. The Dockerfile contains the following instructions:

```
1 FROM python:3-alpine3.15
2 WORKDIR /app
3 COPY . /app
4 RUN pip install -r requirements.txt
5 EXPOSE 3000
6 CMD python ./hello.py
```

The terminal output shows the following commands and their results:

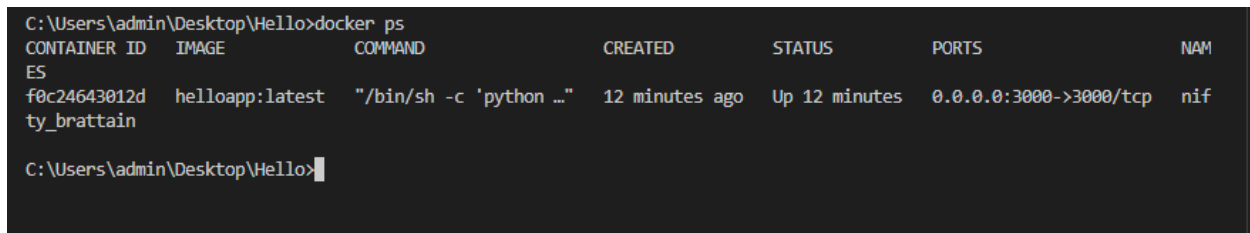
```
=> => naming to docker.io/library/helloapp:latest
PS C:\Users\admin\Desktop\Hello> docker container run -d -p 3000:3000 helloapp:latest
4b08dea605e23ca82df4633eaa6da1b6f47382b15fd3bd4bfac2716a252d32
PS C:\Users\admin\Desktop\Hello> docker build -t helloapp:latest .
[+] Building 10.2s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 162B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3-alpine3.15
=> [1/4] FROM docker.io/library/python:3-alpine3.15@sha256:1567c44f47c8136752ee4301be2e94130a5f8e1fb7a0e737ccac28fb8f6
=> [internal] load build context
=> => transferring context: 225B
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:ba217ee227de4bd95d33ff6b5c88da009b7ea736d2989c3b202261fd7cb93037
=> => naming to docker.io/library/helloapp:latest
PS C:\Users\admin\Desktop\Hello> docker container run -d -p 3000:3000 helloapp:latest
f0c24643012d08c3f4022ea99843dc3d4c54579be8461fa291fc172fb82bae6e
PS C:\Users\admin\Desktop\Hello>
```

STEP 7: CHECKING CONTAINER



The screenshot shows the Docker Desktop interface. The 'Containers' tab is selected, displaying a list of running containers. The table below shows the details of the running container:

Name	Image	Status	Port(s)	Last started	Actions
nifty_brattain f0c24643012d	helloapp:latest	Running	3000:3000	12 minutes ago	[Stop] [Restart] [Delete]



```
C:\Users\admin\Desktop\Hello>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
f0c24643012d   helloapp:latest "/bin/sh -c 'python ..." 12 minutes ago Up 12 minutes  0.0.0.0:3000->3000/tcp            nifty_brattain

C:\Users\admin\Desktop\Hello>
```

STEP 8: OPENING OR RUNNING CONTAINER

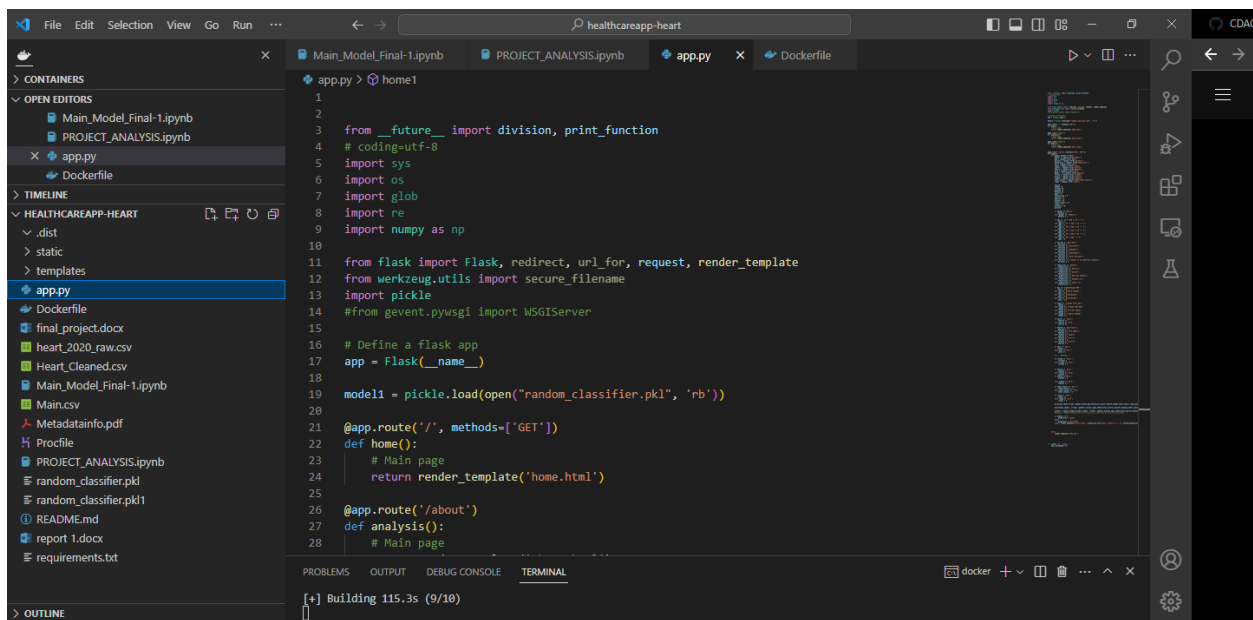


This is demo project created to ddeploy flask app on docker

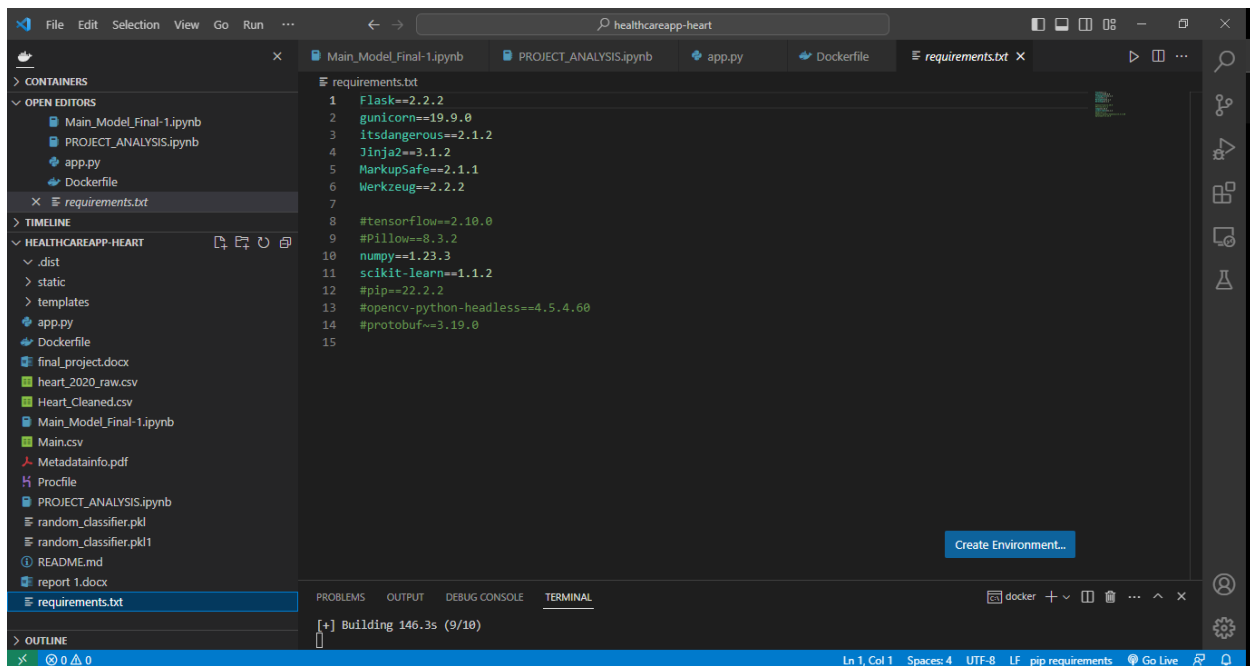
STEP 1: CRETED APP

- 2. Create a python project with UI and Backend with database, create Docker file and deploy it.

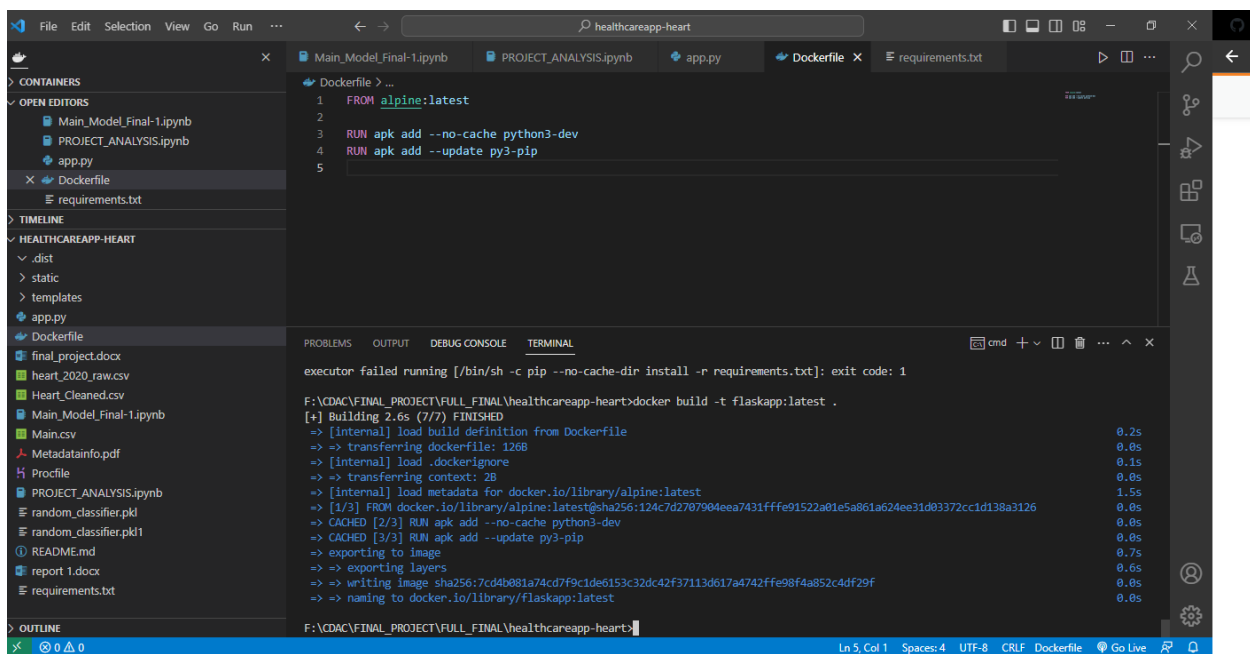
STEP 1: Creating python project using flask.



STEP 2: Adding requirement.txt i.e environment setup requirement



STEP 3: Creating Docker file to install all dependencies



STEP 4: Running build command to create image file

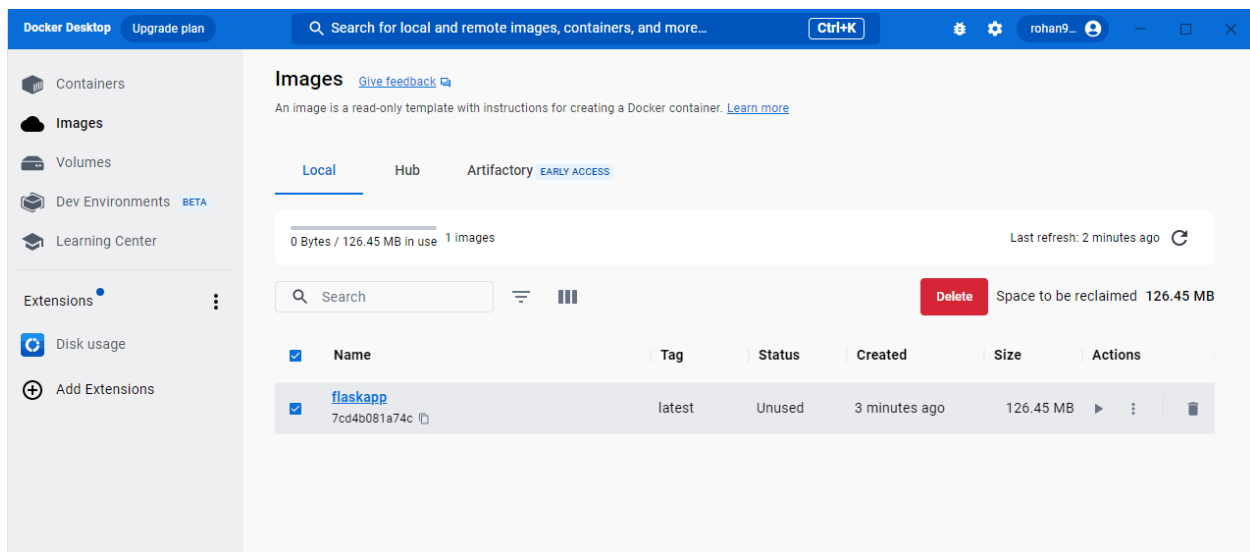
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
executor failed running [/bin/sh -c pip --no-cache-dir install -r requirements.txt]: exit code: 1

F:\CDAC\FINAL_PROJECT\FULL_FINAL\healthcareapp-heart>docker build -t flaskapp:latest .
[+] Building 2.6s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                0.2s
=> => transferring dockerfile: 126B                                              0.0s
=> [internal] load .dockerignore                                                 0.1s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/alpine:latest                 1.5s
=> [1/3] FROM docker.io/library/alpine:latest@sha256:124c7d2707904eea7431fffe91522a01e5a861a624ee31d03372cc1d138a3126 0.0s
=> CACHED [2/3] RUN apk add --no-cache python3-dev                             0.0s
=> CACHED [3/3] RUN apk add --update py3-pip                                    0.0s
=> exporting to image                                                           0.7s
=> => exporting layers                                                         0.6s
=> => writing image sha256:7cd4b081a74cd7f9c1de6153c32dc42f37113d617a4742ffe98f4a852c4df29f 0.0s
=> => naming to docker.io/library/flaskapp:latest                             0.0s

F:\CDAC\FINAL_PROJECT\FULL_FINAL\healthcareapp-heart>
```

STEP 5: Using Docker desktop to check wheatear image is created or not

```
F:\CDAC\FINAL_PROJECT\FULL_FINAL\healthcareapp-heart>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
flaskapp      latest    7cd4b081a74c  9 minutes ago  126MB
```



STEP 5: Adding more dependencies to Docker file

```

Dockerfile > ...
1  FROM alpine:latest
2
3  RUN apk add --no-cache python3-dev
4  RUN apk add --update py3-pip
5  WORKDIR /app
6  COPY . /app
7  RUN pip install -r requirements.txt
8  EXPOSE 3000
9  CMD python ./app.py
10

```

STEP 6: Again building a images and creating container using command

```

F:\CDAC\FINAL_PROJECT\FULL_FINAL\healthcareapp-heart>docker container run -d -p 3000:3000 flaskapp:latest
991d663b7e5b559238e46e8d95e41ece47619c56e3efdc01f10e0c0318a686fc

```

STEP 7: CHECKING CONTAINER

Docker Desktop Upgrade plan Search for local and remote images, containers, and more... Ctrl+K rohan9...

Containers [Give feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Search Only show running containers ☐

	Name	Image	Status	Port(s)	Last started	Actions
<input type="checkbox"/>	xenodochial_faraday 991d663b7e5b	flaskapp:latest	Exited	3000:3000	52 seconds ago	▶ ⋮ 🗑️

```

F:\CDAC\FINAL_PROJECT\FULL_FINAL\healthcareapp-heart>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
991d663b7e5b   flaskapp:latest  "/bin/sh"               2 minutes ago  Exited (0)   2 minutes ago          xenodochial_faraday

```

