TASK

Task Given:

Please find a machine test down below.
We have created a simple FastAPI application that needs to be dockerized. The application has the following endpoints:
Repository: https://github.com/RohitPatil18/docker-fastapi-test
To find out more about running the FastAPI application refer to,
https://fastapi.tiangolo.com/tutorial/first-steps/
You can try running APIs on /docs URL
Method
Endpoint
Description
GET
/
Returns a hello message
GET
/users
Returns list of users stored in JSON file
POST
/users
Accepts and stores user data in JSON file
should be able to run an application using a docker-compose file. Please note that we are not using a database instead storing data in a users.json file in the data directory which will get automatically created if not present.
Once the application runs successfully, make sure to destroy containers and recreate another one and check if previous data is still present.
Please fork the repository add your code to your repository and share it with us.

Step:1

Fork and clone the given repo
CMD: git clone https://github.com/RohitPatil18/docker-fastapi-test

Step:2

Create a Dockerfile and docker-compose.yml file
Vi Dockerfile

```
# Use the official Python image from the Docker Hub
FROM python:3.9-slim

# Set the working directory inside the container
WORKDIR /app
```

```
# Copy the requirements.txt file into the container at /app
COPY requirements.txt /app/

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

# Copy the current directory contents into the container at /app
COPY . /app

# Make port 80 available to the world outside this container
EXPOSE 80

# Run uvicorn when the container launches
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "80"]
```

Vi docker-compose.yml

```
version: '3.8'

services:
  web:
    build: .
    ports:
      - "8080:80"
    volumes:
      - ./data:/app/data
```

Note:
Make sure files are in correct path
Use tree command

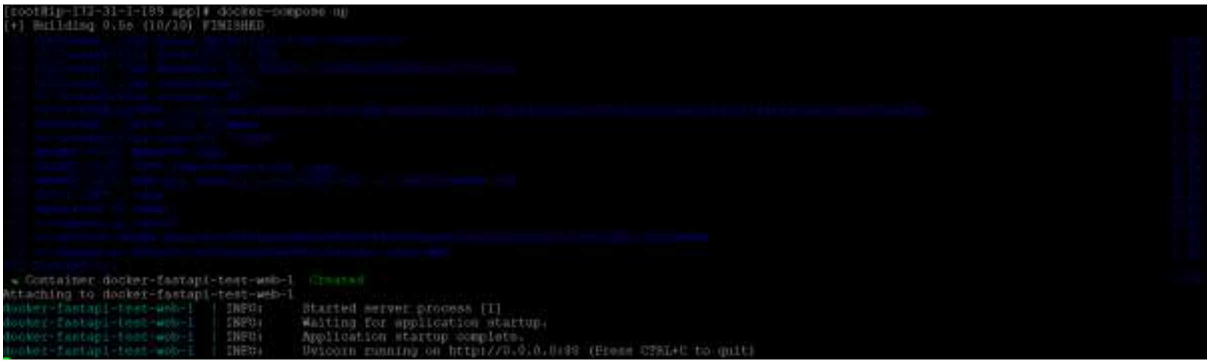
```
.


├── Dockerfile
├── README.md
├── app
│   ├── __init__.py
│   ├── __pycache__
│   │   ├── __init__.cpython-39.pyc
│   │   ├── main.cpython-39.pyc
│   │   ├── schema.cpython-39.pyc
│   │   └── services.cpython-39.pyc
│   ├── app
│   │   └── data
```

```
|   ├── data
|   |   └── users.json
|   ├── main.py
|   ├── schema.py
|   └── services.py
├── data
├── docker-compose.yml
└── requirements.txt
```

Step:3

To create a Container by using above docker files

CMD: docker-compose up

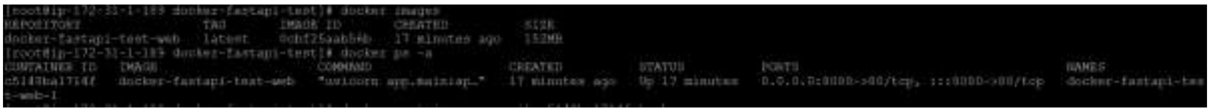


```
[root@ip-172-31-1-189 app]# docker-compose up
[+] Building 0.5s (10/10) FINISHED
[illegible]
 ✔ Container docker-fastapi-test-web-1  Created
Attaching to docker-fastapi-test-web-1
docker-fastapi-test-web-1  | INFO:     Started server process [1]
docker-fastapi-test-web-1  | INFO:     Waiting for application startup.
docker-fastapi-test-web-1  | INFO:     Application startup complete.
docker-fastapi-test-web-1  | INFO:     Uvicorn running on http://0.0.0.0:80 (Press CTRL+C to quit)
```

Step:4

Check the container

CMD: docker ps -a

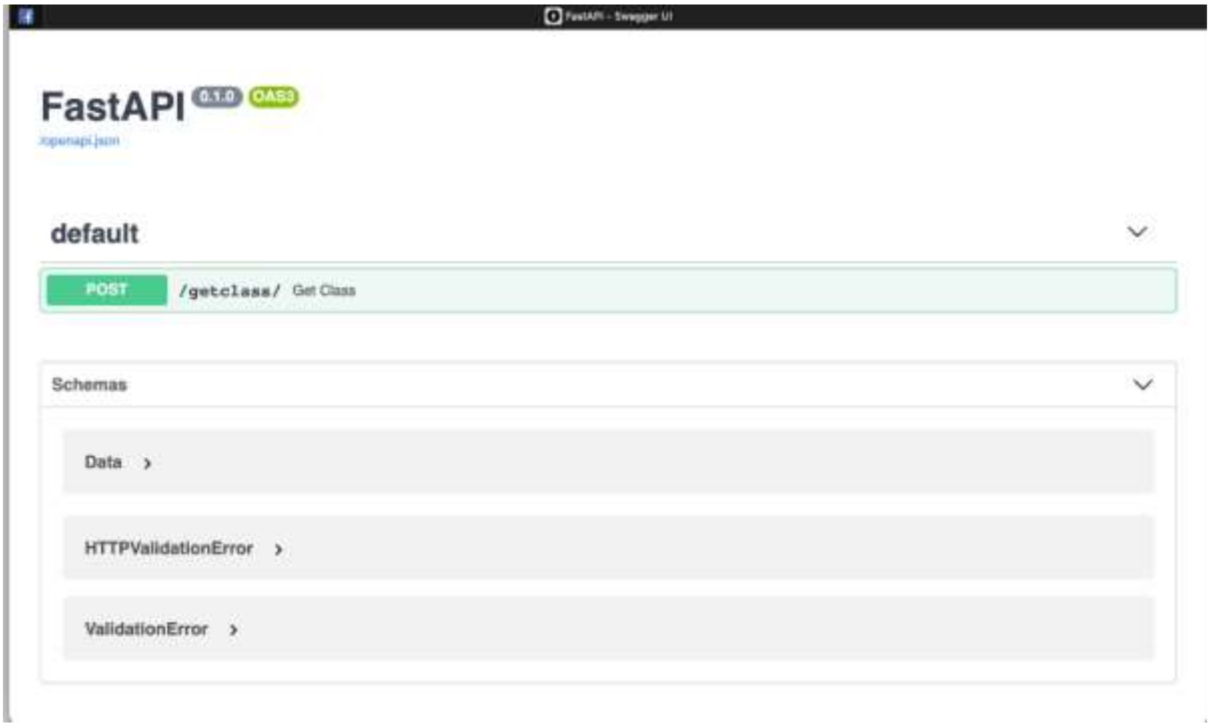


```
[root@ip-172-31-1-189 docker-fastapi-test]# docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
docker-fastapi-test-web   latest    0cbf25aab54b   17 minutes ago   152MB
[root@ip-172-31-1-189 docker-fastapi-test]# docker ps -a
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS                                   NAMES
c5143ba1714f   docker-fastapi-test-web   "uvicorn app.main:ap…"   17 minutes ago   Up 17 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   docker-fastapi-tes
t-web-1
```

Step:5

Check the application using

http:// 3.144.242.242:8080



Step:6

Test the Application

POST http:// 3.144.242.242:8080/users” -H “Content-Type:application/json” -d ‘{“id”:1, “name”:”John Doe”, “email”: “John@example.com

Step:7

Stop the Container Verify data persistence

CMD: docker-compose down

Step:8

Check the data/users.json File: Ensure that the file contains the data you previously added.

curl -X GET http://3.144.242.242:8000/users

Step:9

Commit and push your changes to your forked repository:

CMD: git add.

git commit -m “Fast API App”

git push origin main