Task

Hosting 3-tier studentapp via docker image

Name – Shashank Sharma

1. Create EC2 instance with ubuntu image and connect it.
2. Install docker in the instance terminal using root user (sudo -i) and after installing docker start the docker using
   systemctl start docker

   Link - https://docs.docker.com/engine/install/ubuntu/
3. Firstly create data base in MySQL for this use command.
   docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=1234 mysql:latest
   docker ps

```
root@ip-172-31-18-203:~# docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=1234 mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
9a5c778f631f: Pull complete
9e77c3a95bf2: Pull complete
8b279a2086e0: Pull complete
c8bfbcde7882: Pull complete
d35b074b68ec: Pull complete
beea5014e6af: Pull complete
dc3791a61558: Pull complete
52f9323b9f0e: Pull complete
7f7391eab49b: Pull complete
8d2f04b287ee: Pull complete
Digest: sha256:9d1c923e5f66a89607285ee2641f8a53430a1ccd5e4a62b35eb8a48b74b9ff48
Status: Downloaded newer image for mysql:latest
ba77ad53be96aca5ebb3b9fe9f3c628291653d7524e409a2f7c81c206b8281f0
root@ip-172-31-18-203:~# docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS         PORTS                                                  NAMES
ba77ad53be96   mysql:latest  "docker-entrypoint.s…"   9 seconds ago   Up 7 seconds   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   hardcore_snyder
```

   docker exec -it <container-id> mysql -u root -p1234

4. After entering this command you enter into the mysql use commands to create database.

```
create database studentapp;
use studentapp;
CREATE TABLE if not exists students(student_id INT NOT NULL
AUTO_INCREMENT,
        student_name VARCHAR(100) NOT NULL,
   student_addr VARCHAR(100) NOT NULL,
        student_age VARCHAR(3) NOT NULL,
        student_qual VARCHAR(20) NOT NULL,
        student_percent VARCHAR(10) NOT NULL,
        student_year_passed VARCHAR(10) NOT NULL,
        PRIMARY KEY (student_id)
);
desc students;
exit
```

```
mysql> create database studentapp;
Query OK, 1 row affected (0.00 sec)

mysql> use studentapp;
Database changed
mysql> CREATE TABLE if not exists students(student_id INT NOT NULL AUTO_INCREMENT,
    -> student_name VARCHAR(100) NOT NULL,
    ->     student_addr VARCHAR(100) NOT NULL,
    -> student_age VARCHAR(3) NOT NULL,
    -> student_qual VARCHAR(20) NOT NULL,
    -> student_percent VARCHAR(10) NOT NULL,
    -> student_year_passed VARCHAR(10) NOT NULL,
    -> PRIMARY KEY (student_id)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

5. To see image ip. (mysql ip)
   docker inspect <container-id> |grep “IP”

6. Copy mysql ip for the further use.

❖ Backend

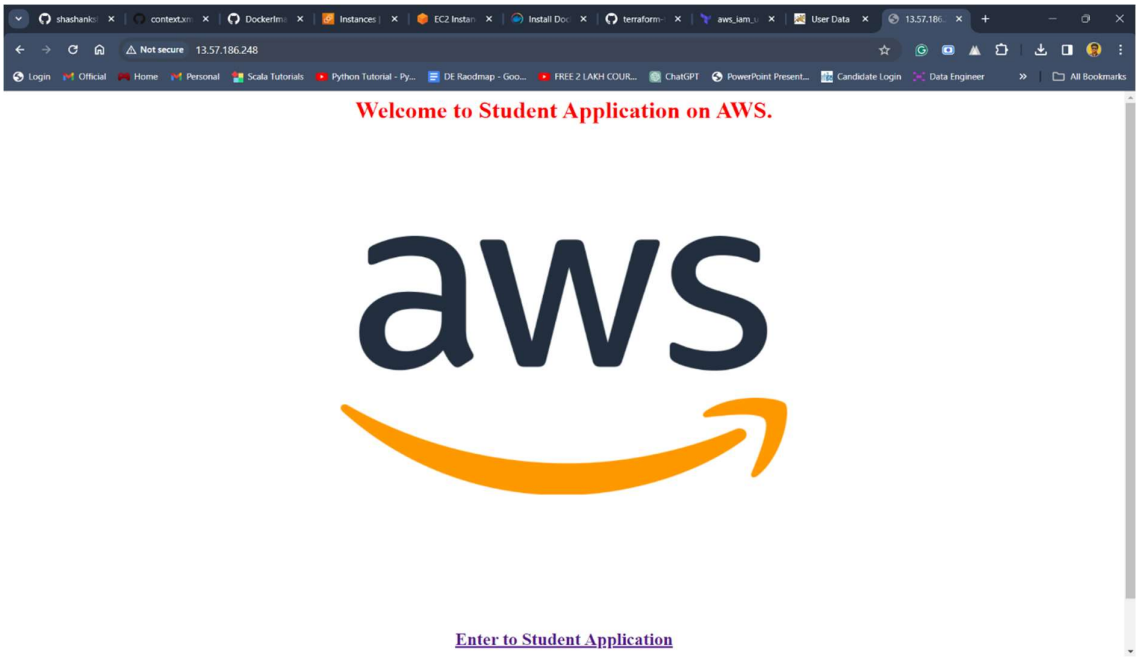
1. Create repo in git or use existing and make two folder in the repo.
   1.frontend
   2.backend
2. In backend create three files.
   1.Dockerfile -> your image
   2.context.xml -> add mysql ip
   3.studnet.war ->

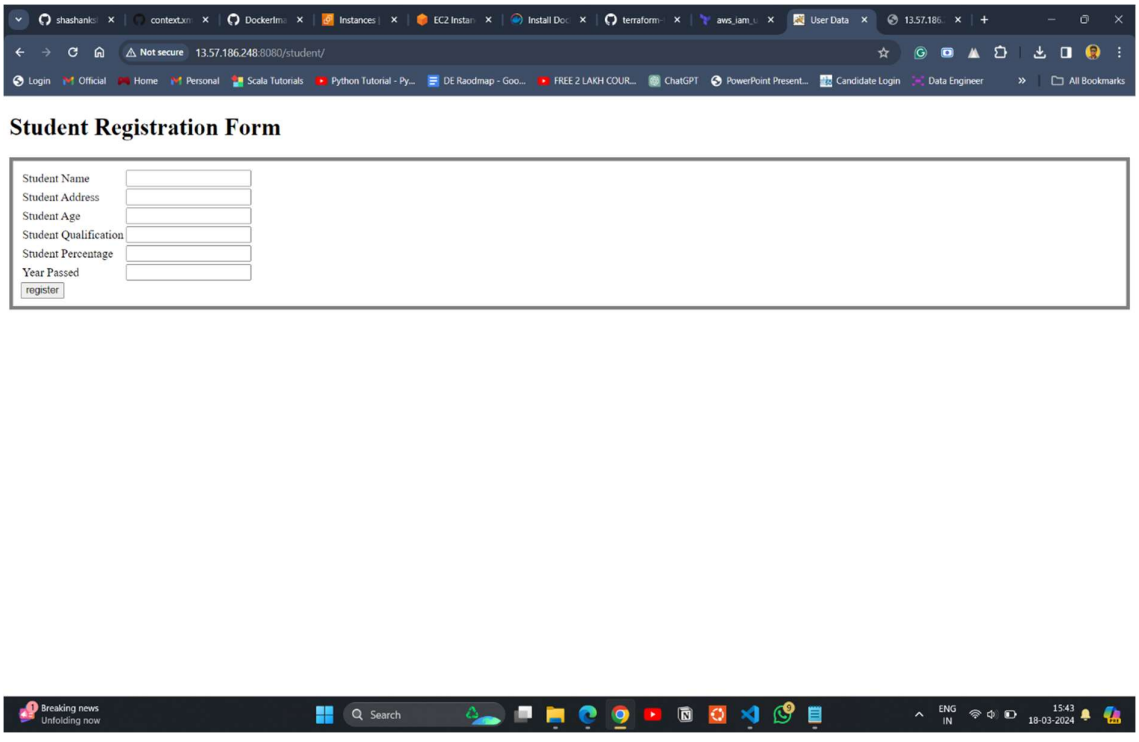
3. Make git clone and build docker image.
   docker build .
   docker images
   Assign ip to image -> docker run -d -p 8080:8080 <image-id>
   docker ps
4. Hit ip to see hosting.

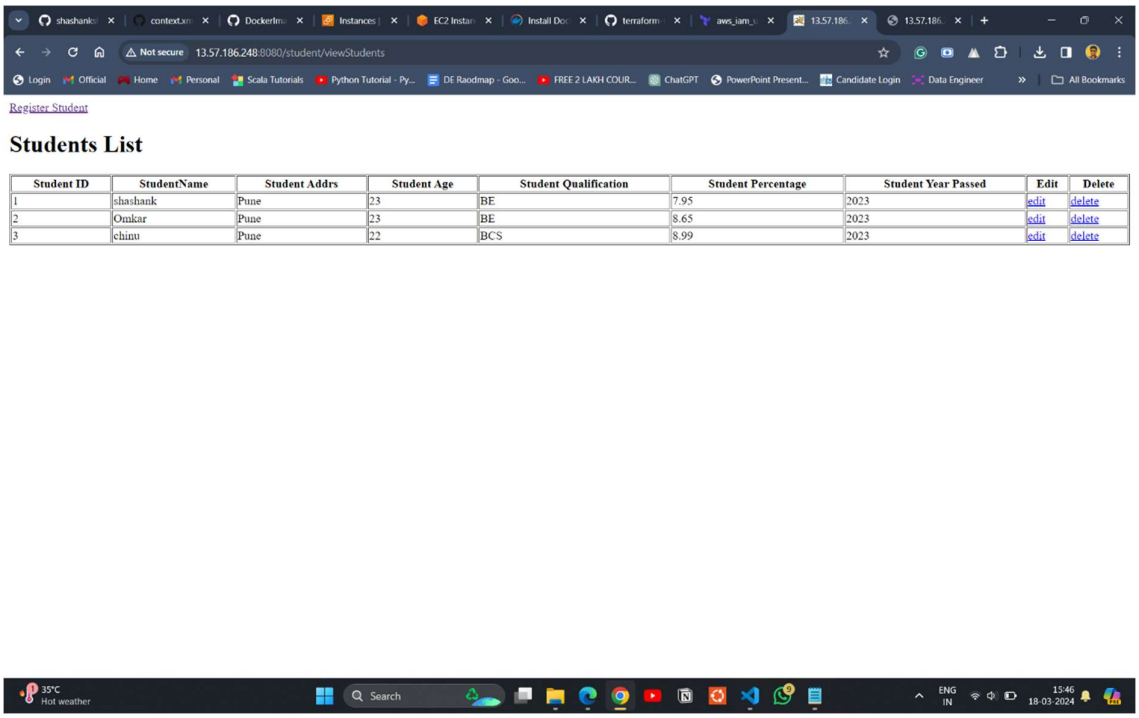
❖ Frontend

1. Create two files in frontend folder.
   1.Dockerfile -> your image
   2.index.html -> paste your EC2 instance ip

2. Git push or pull
3. Build docker image -> docker build .
4. Docker images
5. Assign ip -> docker run -d -p 80:80 <image-id>
6. docker ps


Not secure 13.57.186.248
Welcome to Student Application on AWS.
Enter to Student Application


Not secure 13.57.186.248:8080/student/
Student Registration Form
Student Name
Student Address
Student Age
Student Qualification
Student Percentage
Year Passed
register



- If you want to push files to the dockerhub provide tags to your images and then login into the dockerhub using command.
  docker login
  docker push <dockerhub-usrname>/<dockerhub-repo>:[<tag>]