# #DEPLOYING 3TIER STUDENT APP WITH DOCKER

1. **FIRST WE CREATE DATABASE CONTAINER - DATABASE**
2. **SECOND CONFIGUR DOCKERFILE FOR APPLICATION - BACKEND**
3. **THIRD CONFIGUR DOCKERFILE FOR FRONT PAGE - FRONTEND**

- **DATABASE**

1) **Create instance (amazon linux)**

2) **Connect instance terminal**

3) **Install docker in instance**
   yum install docker -y

4) **Start docker service**
   systemctl start docker

5) **Create docker volume**
   docker volume create <student-db>

   docker volume create database
   (docker volume ls → to check volume)

6) **Create database container in detached mode with "database" volume**
   docker run -d -v <volume name>:/var/lib/mysql -e
   MYSQL_ROOT_PASSWORD=<password> mysql

   docker run -d -v database:/var/lib/mysql -e
   MYSQL_ROOT_PASSWORD=12345678 mysql
   (docker ps → for check container)

7) **Now go inside the container**
   docker exec -it <containerID> bash
   docker exec -it 0939 bash

8) **Enter in mysql service and configure database**
   mysql -u root -p12345678
   create database studentapp; → create database
   show database; → to check database
   use database studentapp; → to use database named studentapp
   #Paste a schema → next page

```
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)
);
```

9) **exit** → exit from mysql
10) **exit** → exit from container

This is manually configure database either we can aslo configure through dockerfie

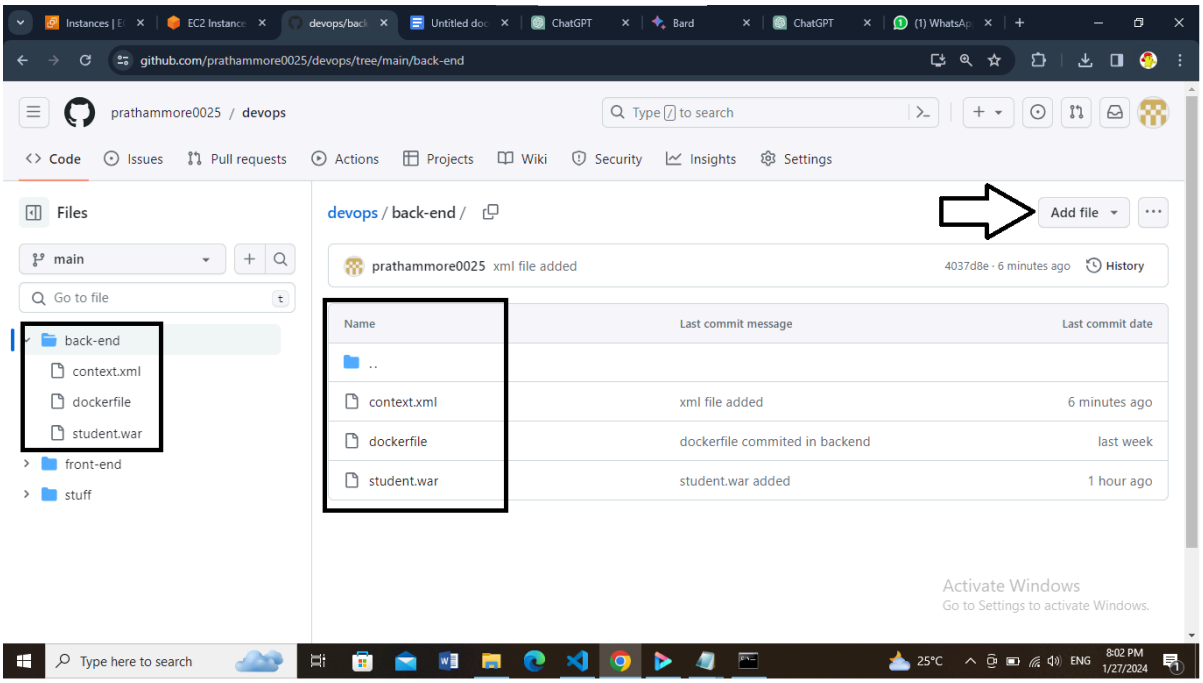
—----------*DATABASE DONE*—----------

- **BACKEND**
  For backend used docker file
  1) **Go to github website create repo**
  2) **Open that repo in visual code**
  3) **Create 2 folder named “backend” and “frontend”**
  4) **In backend folder create file name “dockerfile” and add scoure code file - student.war**
  5) **Write dockerfile for backend open dockerfile in visual code**
  6) **Write docker file :-**

```
FROM ubuntu
RUN apt-get update && \
    apt-get install -y openjdk-11-jdk
ADD
https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.98/bin/a
pache-tomcat-8.5.98.tar.gz ./
RUN tar -xzf apache-tomcat-8.5.98.tar.gz -C /opt/ && \
    rm -rf ./apache-tomcat-8.5.98.tar.gz
WORKDIR /opt/apache-tomcat-8.5.98
COPY  context.xml conf/context.xml
ADD
https://s3-us-west-2.amazonaws.com/studentapi-cit/mysq
l-connector.jar lib/mysql-connector.jar
COPY student.war webapps/student.war
EXPOSE 8080
CMD ["./bin/catalina.sh", "run"]
```

  7) **Commit and sync changes**

8) **Add war file and xml in backend folder from github website**




**Go to add file for add war and xml file select file and commit it**
**So now have 3 files in backend 1-dockerfile, 2-student.war 3-context.xml**
**Before adding context.xml do some config within file update:-**

```
<Resource name="jdbc/TestDB" auth="Container"
type="javax.sql.DataSource"
               maxTotal="100" maxIdle="30" maxWaitMillis="10000"
               username="USERNAME" password="PASSWORD"
driverClassName="com.mysql.jdbc.Driver"
               url="jdbc:mysql://DB-ENDPOINT:3306/DATABASE"/>
```

**USERNMAME=“root” , PASSWORD=“12345678”**
**DB-ENDPOINT “172.17.0.2” (**to get this ip go to terminal hit “docker inspect <containIP>” copy ip address and paste in xml file**) database containerIP.**
**DATABASE=“studentapp”**

**~Connect to instance terminal**

9) **Install git**
   yum install git -y

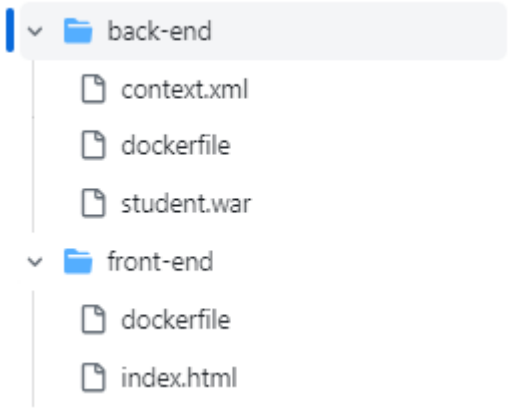
10) **Clone the repo**
   git clone <http link of repo>
   git clone https://github.com/prathammore0025/devops.git

11) **Go to backend directory**
   cd /devops/backend

12) **Now build the docker image from docker file**
    docker build . -t  back:v1
    (docker images)
13) **Run the backend container**
    docker run -d -p 8080:8080 back:v1
    (docker ps → for check)
14) **Hit ip**
    instance pubIP:8080 → tomcat (http://13.39.24.10:8080/)
    instamncepubIP:8080/student → studentapp
    (http://13.39.24.10:8080/student/)
    Save data

—----------*BACKEND COMPLET*----------

- **FRONTEND**
  * Create a folder named "**frontend**" in visual code git codebase below the backend folder
  * In that frontend folder create file **index.html** and **dockerfile**




1) **Create index.html file**
2) **Open in vs code enter code**

```
<h1 style="text-align: center;"><span style="color: #ff0000;">Welcome
to Student Application on AWS.</span></h1>
<p><img style="display: block; margin-left: auto; margin-right: auto;"
src="https://cdn-images-1.medium.com/max/2000/1*tFl-8wQUENETYLjX5mYWuA.
png" alt="" width="1200" height="630" /></p>
<p> </p>
<h2 style="text-align: center;"><a
href="http://35.181.154.243:8080/student/"><strong>Enter to Student
Application</strong></a></h2><p> </p>
<p> </p>
```

**#add backend link in index.html file in front of href="backend link"**
**#Commit and sync changes**

3) **Create dockerfile**
4) **Open in vs code write**

```
FROM ubuntu
LABEL author=prathameshmore

# Update package list and install Apache
RUN apt-get update && \
    apt-get install -y apache2

# Copy the index.html file to the web server root directory
COPY index.html /var/www/html/index.html

# Set the working directory to the web server root directory
WORKDIR /var/www/html

# Expose port 80 for HTTP traffic
EXPOSE 80

# Start Apache in the foreground as the entry point
CMD ["apache2ctl", "-D", "FOREGROUND"]
```

**#commit and sync changes**

5) **Open terminal**
6) **Go inside the repo**
   cd <repo name>

7) **Pull the changes: updated frontend dockerfile**
   git pull origin main

8) **Go inside the frontend directory**
   cd frontend

9) **Build frontend image**
   docker build . -t front:v1
   (check :- docker images)

10) **Run container from frontend image**
    docker run -d -p 80:80 front:v1
    (check :- docker ps)
    **Hit publicIP of instance on browser to check frontend**

—----------*FRONTEND COMPLET*----------

**#Git repo link=** https://github.com/prathammore0025/devops.git
**#At the end there was 3 container running**