**Operating System project:**

**Dockerizing DBMS application**

|  |  |
| --- | --- |
| Title of the Project: **Dockerizing of the DBMS Project** | |
| USN: **1MS21CS400** | NAME: BHAVANI M R |
| USN: **1MS21CS403** | NAME: KEERTHI K |
| USN: **1MS21CS406** | NAME: RAVISHA J H |
| USN: **1MS21CS409** | NAME: SNEHA C D |

**Description of our DBMS Project**

During situations like COVID, going out and buying pharmaceutical products is challenging as there is greater probability of ourselves getting exposed to virus. And also, old age people cannot go out and buy due to their age-related issues. So, our project focuses on overcoming these scenarios by creating pharmaceutical database to order medicines online.

A **Pharmaceutical** **Database Application** is our DBMS project.

A place where a user can add his orders for medicine to be delivered safely.

He is able to easily order medicines through our user-friendly interface.

Anyone can add their orders provided they have account(registered) with this app.

Like ordering Dolo 650, Garnier facewash, etc.

It is possible for an adult to add his orders he is wishing to be delivered and delete the order before delivery if not required.

Can be used by anybody as a virtual medical store.

**Dockerizing our DBMS project**

1. **Creating the Docker files and Docker CLI**

|  |
| --- |
| FROM php:7.4-apache |
|  | RUN apt-get update && apt upgrade -y |
|  | RUN docker-php-ext-install mysqli pdo pdo\_mysql && docker-php-ext-enable mysqli |
|  | ADD ./app /var/www/html |
|  | COPY ./app/my-site.conf /etc/apache2/sites-available/my-site.conf |
|  | RUN echo 'SetEnv MYSQL\_DB\_CONNECTION ${MYSQL\_DB\_CONNECTION}' >> /etc/apache2/conf-enabled/environment.conf |
|  | RUN echo 'SetEnv MYSQL\_DB\_NAME ${MYSQL\_DB\_NAME}' >> /etc/apache2/conf-enabled/environment.conf |
|  | RUN echo 'SetEnv MYSQL\_USER ${MYSQL\_USER}' >> /etc/apache2/conf-enabled/environment.conf |
|  | RUN echo 'SetEnv MYSQL\_PASSWORD ${MYSQL\_PASSWORD}' >> /etc/apache2/conf-enabled/environment.conf |
|  | RUN echo 'SetEnv SITE\_URL ${SITE\_URL}' >> /etc/apache2/conf-enabled/environment.conf |
|  | RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf &&\ |
|  | a2enmod rewrite &&\ |
|  | a2enmod headers &&\ |
|  | a2enmod rewrite &&\ |
|  | a2dissite 000-default &&\ |
|  | a2ensite my-site &&\ |
|  | service apache2 restart |
|  | EXPOSE 80 |
|  | EXPOSE 443 |

**Docker-compose.yml**

|  |
| --- |
| version: "2" |
|  | services: |
|  | webserver: |
|  | image: mydemophpimage |
|  | ports: |
|  | - "80:80" |
|  | - "443:443" |
|  | volumes: |
|  | - ./app:/var/www/html |
|  | environment: |
|  | MYSQL\_DB\_CONNECTION: test |
|  | MYSQL\_DB\_NAME: test |
|  | MYSQL\_USER: test |
|  | MYSQL\_PASSWORD: test |
|  | SITE\_URL: http://localhost |
|  | networks: |
|  | - test |
|  |  |
|  | networks: |
|  | test: |
|  | external: true |

1. **Running the app in Docker**

To run the app, you need to perform two steps:

1. Build the image: This is done using the build command, which uses the Dockerfile you just created. To build the image, run the command below: docker build . -t dbms-project . This command should be executed in the directory where the Docker file lives. The -t flag tags the image so that it can be referenced when you want to run the container.
2. Run the image: This is done using the docker run command. This will convert the built image into a running container. To run the app, execute the below command:

docker run -it -p 8000:8000 dbms-project

You can proceed to view your app in the browser at localhost:8000.

1. **To check the containers which are in running state**

docker ps -a

Here a refers to all images which are in running state