Experiment No. 9

PART I: Containerize an Application Using Docker CLI Commands

Objective:

To create an **Nginx web server container** that hosts a static web application using Docker CLI commands.

STEP 1: Download Nginx Official Image and Containerize the Web Application

docker images

docker pull nginx

STEP 2: Run the Container from the Nginx Image

root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run --name webserver1 5ef

In another terminal:

docker ps -a

Exit from the container:

Ctrl + C

Check containers again:

docker ps -a

Remove the container if required.

STEP 3: Run a New Container with Port Mapping

root@labvm:/home/devasc/Desktop/DOCKER_LAB# docker run -it -p 3031:80 --name server1 nginx:latest bash

In another terminal:

docker ps -a

STEP 4: Create a Static Website Inside the Container

Go to the HTML directory:

cd /usr/share/nginx/html/

ls

Rename the default index file:

mv index.html index.html_backup

Open a new file:

nano index.html

If nano is not installed:

apt install nano

Add this HTML content:

```
<!DOCTYPE html>
<html>
<head>
  <title>Student Login Form</title>
</head>
<body>
  <h2>Student Login Form</h2>
  <form>
    Username: <input type="text" name="username"><br><br>
    Password: <input type="password" name="password"><br><br>
    <input type="checkbox" name="remember"> Remember me<br><br>
    <input type="submit" value="Login">
    <input type="reset" value="Cancel">
    <a href="#">Forgot password?</a>
  </form>
</body>
</html>
```

Save and exit nano:
Ctrl + O, Enter, Ctrl + X

STEP 5: Check Nginx Service Status

docker exec 595 service nginx start

STEP 6: Verify in Browser

Open your browser and visit:

localhost:3031

If the container is stopped, start it again.

Pause and unpause:

docker pause 595

docker unpause 595

Remove the container:

docker rm 595

Verify removal:

docker ps -a

PART II: Creating a Docker Image Using Dockerfile

Objective:

To build a custom Docker image for the web application using a Dockerfile.

STEP 1: Create a Dockerfile

Create a file named Dockerfile and add instructions to set up your application environment (for example, based on Nginx or Apache).

STEP 2: Build the Docker Image

docker build -t sujatadocker2024/websitetest .

STEP 3: Run the Container

docker run -it -p 3032:80 sujatadocker2024/websitetest

STEP 4: Verify in Browser

Open:

localhost:3032

Your custom Docker image should now host the web application.

STEP 5: Push the Image to Docker Hub

Login to Docker Hub:

docker login

Push your image:

docker push sujatadocker2024/websitetest

Go to Docker Hub and verify that your image is uploaded successfully.