49. Create a web application with simple web page containing login details and create a docker image of the application.(Use Ngnix Web server)

To create a web application with a simple login page and package it into a Docker image using NGINX as the web server, follow these steps:

1. Set Up the Project Directory

Create a directory for the project:

bash

Copy code

```
mkdir simple-login-app

cd simple-login-app
```

2. Create HTML for the Login Page

Inside the project directory, create an `index.html` file for the login page:

index.html

html

Copy code

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Simple Login Page</title>
```

```html
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="login-container">
        <h2>Login</h2>
        <form action="/login" method="POST">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
            <button type="submit">Login</button>
        </form>
    </div>
</body>
</html>
```

styles.css Create a simple CSS file for basic styling:

css

Copy code

```css
body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    background-color: #f0f0f0;
}

.login-container {
    background-color: #fff;
    padding: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 300px;
    border-radius: 8px;
}

h2 {
    text-align: center;
}

label, input {
    display: block;
    width: 100%;
    margin-bottom: 10px;
```

```
}

button {
    width: 100%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}
```

3. Configure NGINX

Create an `nginx.conf` file for NGINX to serve the HTML file.

nginx.conf

nginx

Copy code

```
server {
    listen 80;
    server_name localhost;
    location / {
        root /usr/share/nginx/html;
        index index.html;
    }

    location /login {
        return 200 'Login functionality is currently a placeholder.';
        add_header Content-Type text/plain;
    }
}
```

4. Create a Dockerfile

The Dockerfile will create an image using NGINX to serve the web page.

Dockerfile

dockerfile

Copy code

```
# Use the official NGINX base image
FROM nginx:latest

# Copy HTML and CSS files to the NGINX html directory
COPY index.html /usr/share/nginx/html/
COPY styles.css /usr/share/nginx/html/
```

```
# Copy custom NGINX configuration file

COPY nginx.conf /etc/nginx/conf.d/default.conf


# Expose port 80

EXPOSE 80


# Start NGINX server

CMD ["nginx", "-g", "daemon off;"]
```

5. Build and Run the Docker Image

Build the Docker image by running the following command in your project directory:

bash

Copy code

```
docker build -t simple-login-app .
```

Once the image is built, you can run a container using this image:

bash

Copy code

```
docker run -p 8080:80 simple-login-app
```

6. Test the Application

Open your browser and navigate to `http://localhost:8080` to see the login page served by the NGINX web server running in a Docker container.

With these steps, you've created a simple login page, set up NGINX as a web server, and dockerized the application!