Experiment 8.2 – Serve the React App with Nginx

Aim:

To build a React application and serve it using the Nginx web server inside a Docker container, ensuring optimized production deployment.

Tools / Technologies Used:

- React.js
- Node.js & npm
- Nginx Web Server
- Docker & Docker CLI
- Code Editor (VS Code recommended)
- Operating System: Windows / Linux / macOS

Theory (Short Explanation):

React applications are typically developed using a development server, but for production deployment, they must be served as static files.

Nginx is a lightweight high-performance web server used to serve static web content efficiently.

The process involves:

- Building React app into static files using: *npm run build*
- Copying the build output to the Nginx default directory
- Running Nginx to serve these files inside a Docker container

This ensures faster performance and secure deployment of React apps in production.

Procedure:

1. Create a React App (if not created already):

```
npx create-react-app my-react-app
```

2. Navigate to the project directory:

```
cd my-react-app
```

3. Build the app for production:
`npm run build`

4. Create a file named **Dockerfile** in the project root.

5. Add instructions to build React and serve via Nginx.

6. Open terminal in project folder and build Docker image:
`docker build -t react-nginx-app .`

7. Run the container:
`docker run -p 80:80 react-nginx-app`

8. Open a browser and visit: `http://localhost`

9. The React app should now be served through Nginx.

Dockerfile Code (React + Nginx):

```
# Step 1: Build the React app
FROM node:18-alpine AS build

WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

# Step 2: Serve using Nginx
FROM nginx:alpine

# Copy build output to Nginx html folder
COPY --from=build /app/build /usr/share/nginx/html

# Expose port 80
EXPOSE 80

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

Output:

React app successfully built and containerized.
App served through Nginx on port 80.

User can access the app via browser: `http://localhost`

Sample Terminal Output:

```
Successfully built c3fd21a5bc77
Successfully tagged react-nginx-app:latest
```

Learning Outcomes:

After performing this experiment, the student will be able to:

- Build a React app for production
- Understand multi-stage Docker builds
- Serve static web apps using Nginx
- Deploy React app inside a Docker container
- Understand production deployment workflow