Exno: 6	**Containerizing a Simple React Application with Docker.**
12.09.2025	

Aim:

To containerize a simple React application using Docker to ensure consistent development and deployment environments. This involves creating a Docker image that packages the React app along with all its dependencies, enabling it to run seamlessly across different systems and platforms without compatibility issues.

Requirements:

1. Write a Dockerfile that:
    - Uses a Node.js base image to build the React app.
    - Uses Nginx as the production server to serve the built React app.
    - Exposes port 80 inside the container.
2. Create a .dockerignore file to exclude unnecessary files such as node_modules and build artifacts.
3. Show the steps to:
    - Build the Docker image.
    - Run the container so that the application is accessible on http://localhost:3000.

Structure:

Code:

App.js:

```
import React, { useState } from "react";

import "./App.css";


function App() {
```

```
const [leaves, setLeaves] = useState([]);

const [employee, setEmployee] = useState("");

const [days, setDays] = useState("");

const [reason, setReason] = useState("");


const handleAddLeave = () => {

  if (employee && days && reason) {

    const newLeave = {

      id: leaves.length + 1,

      employee,

      days,

      reason,

      status: "Pending"

    };

    setLeaves([...leaves, newLeave]);

    setEmployee("");

    setDays("");

    setReason("");

  } else {

    alert("Please fill all fields!");

  }

};
```

```
const handleApprove = (id) => {
  setLeaves(
    leaves.map((leave) =>
      leave.id === id ? { ...leave, status: "Approved" } : leave
    )
  );
};

const handleReject = (id) => {
  setLeaves(
    leaves.map((leave) =>
      leave.id === id ? { ...leave, status: "Rejected" } : leave
    )
  );
};

return (
  <div className="app-container">
    <h1>Employee Leave Tracker</h1>

    <div className="form-container">
      <input
        type="text"
        placeholder="Employee Name"
        value={employee}
        onChange={(e) => setEmployee(e.target.value)}
      />
      <input
        type="number"
        placeholder="No. of Days"
        value={days}
        onChange={(e) => setDays(e.target.value)}
      />
      <input
        type="text"
        placeholder="Reason"
        value={reason}
        onChange={(e) => setReason(e.target.value)}
      />
      <button onClick={handleAddLeave}>Apply Leave</button>
    </div>

    <h2>Leave Applications</h2>
    <table>
      <thead>
        <tr>
    <th>ID</th>

    <th>Employee</th>

    <th>Days</th>

    <th>Reason</th>

    <th>Status</th>

    <th>Action</th>

  </tr>

</thead>

<tbody>

  {leaves.map((leave) => (

    <tr key={leave.id}>

      <td>{leave.id}</td>

      <td>{leave.employee}</td>

      <td>{leave.days}</td>

      <td>{leave.reason}</td>

      <td>{leave.status}</td>

      <td>

        {leave.status === "Pending" && (

          <>

            <button

              className="approve"

              onClick={() => handleApprove(leave.id)}

            >
```

```
                      Approve
                    </button>
                    <button
                      className="reject"
                      onClick={() => handleReject(leave.id)}
                    >
                      Reject
                    </button>
                  </>
                )}
              </td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}

export default App;
```

Index.js:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

.dockerignore:

```
node_modules
build
.dockerignore
Dockerfile
npm-debug.log
```

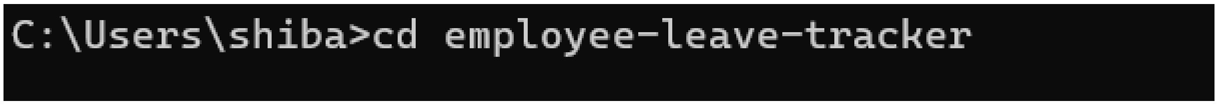
dockerfile:

```
# Stage 1: Build React app
FROM node:18 AS build


WORKDIR /app
COPY package.json package-lock.json* ./
RUN npm install


COPY . .
RUN npm run build


# Stage 2: Serve with Nginx
FROM nginx:stable-alpine


# Copy React build files to Nginx
COPY --from=build /app/build /usr/share/nginx/html


# Expose port 80
EXPOSE 80


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

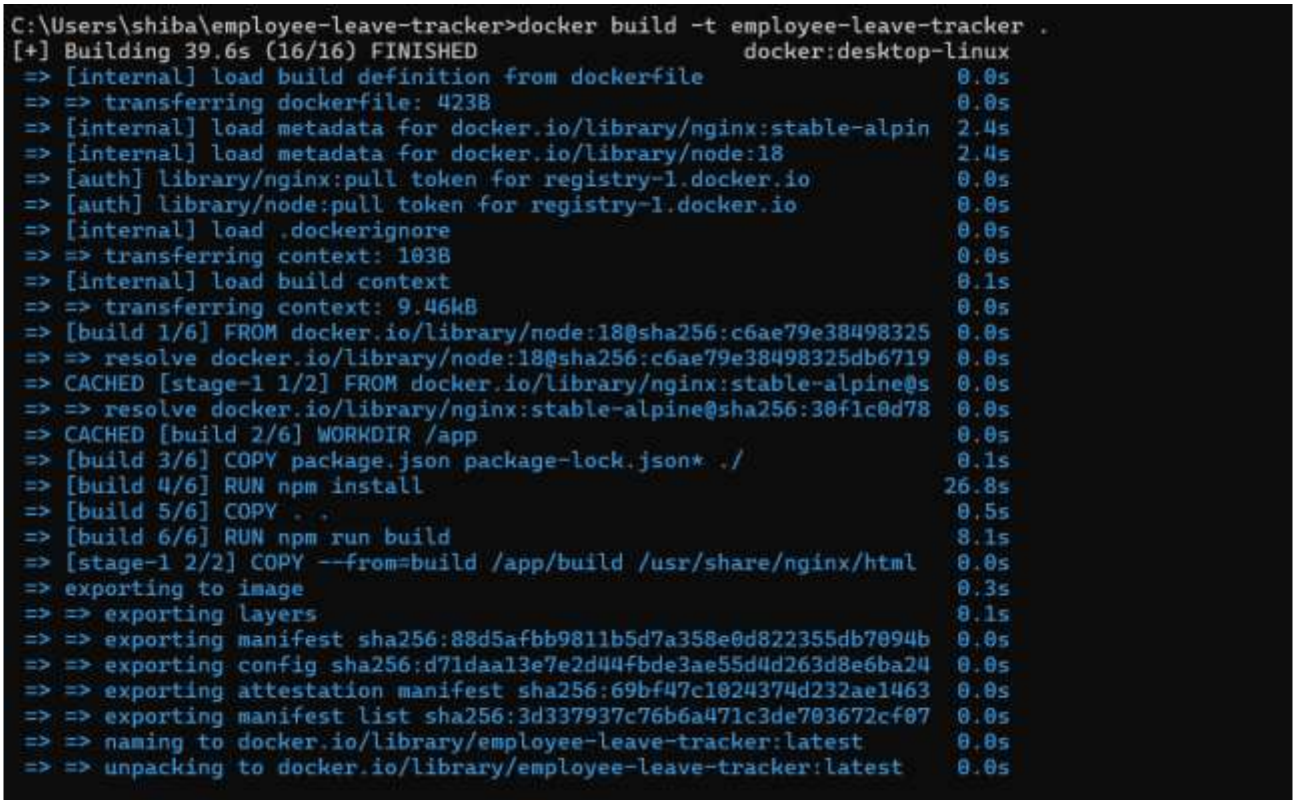
Package.json:

```
{
  "name": "employee-leave-tracker",
  "version": "0.1.0",
  "private": true,
  "homepage": "./",
  "dependencies": {
    "@testing-library/dom": "^10.4.1",
    "@testing-library/jest-dom": "^6.8.0",
    "@testing-library/react": "^16.3.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^19.1.1",
    "react-dom": "^19.1.1",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject",
    "predeploy": "npm run build",
    "deploy": "gh-pages -d build"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "gh-pages": "^6.3.0"
  }
}
```

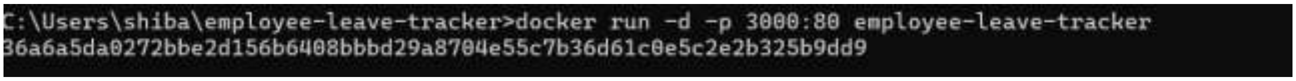
OUTPUT:



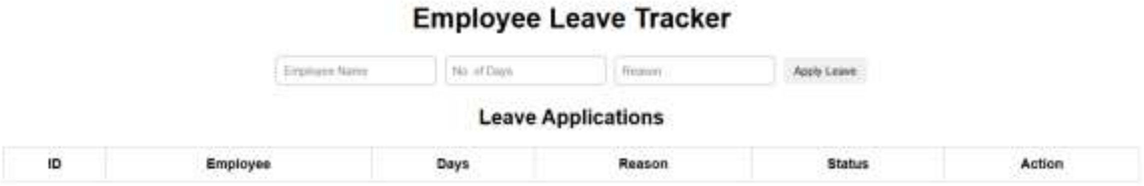
```
C:\Users\shiba>cd employee-leave-tracker
```



```
C:\Users\shiba\employee-leave-tracker>docker build -t employee-leave-tracker .
[+] Building 39.6s (16/16) FINISHED                         docker:desktop-linux
 => [internal] load build definition from dockerfile                        0.0s
 => => transferring dockerfile: 423B                                        0.0s
 => [internal] load metadata for docker.io/library/nginx:stable-alpin       2.4s
 => [internal] load metadata for docker.io/library/node:18                  2.4s
 => [auth] library/nginx:pull token for registry-1.docker.io                0.0s
 => [auth] library/node:pull token for registry-1.docker.io                 0.0s
 => [internal] load .dockerignore                                           0.0s
 => => transferring context: 103B                                           0.0s
 => [internal] load build context                                           0.1s
 => => transferring context: 9.46kB                                         0.0s
 => [build 1/6] FROM docker.io/library/node:18@sha256:c6ae79e38498325       0.0s
 => => resolve docker.io/library/node:18@sha256:c6ae79e38498325db6719       0.0s
 => CACHED [stage-1 1/2] FROM docker.io/library/nginx:stable-alpine@s       0.0s
 => => resolve docker.io/library/nginx:stable-alpine@sha256:30f1c0d78       0.0s
 => CACHED [build 2/6] WORKDIR /app                                         0.0s
 => [build 3/6] COPY package.json package-lock.json* ./                     0.1s
 => [build 4/6] RUN npm install                                            26.8s
 => [build 5/6] COPY . .                                                    0.5s
 => [build 6/6] RUN npm run build                                           8.1s
 => [stage-1 2/2] COPY --from=build /app/build /usr/share/nginx/html        0.0s
 => exporting to image                                                      0.3s
 => => exporting layers                                                     0.1s
 => => exporting manifest sha256:88d5afbb9811b5d7a358e0d822355db7094b       0.0s
 => => exporting config sha256:d71daa13e7e2d44fbde3ae55d4d263d8e6ba24       0.0s
 => => exporting attestation manifest sha256:69bf47c1024374d232ae1463       0.0s
 => => exporting manifest list sha256:3d337937c76b6a471c3de703672cf07       0.0s
 => => naming to docker.io/library/employee-leave-tracker:latest            0.0s
 => => unpacking to docker.io/library/employee-leave-tracker:latest         0.0s
```



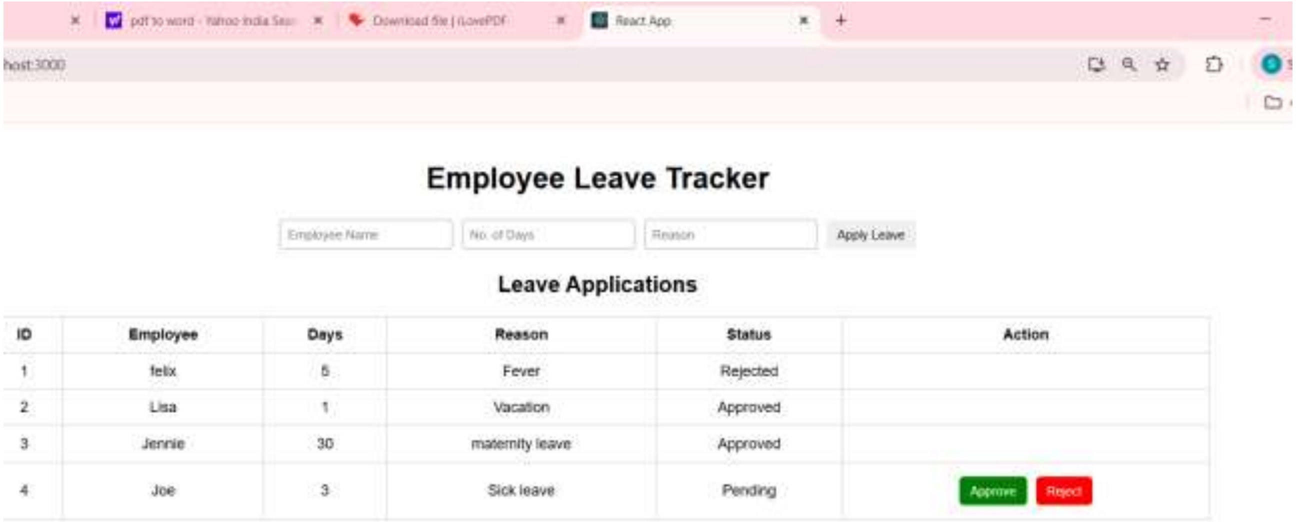
```
C:\Users\shiba\employee-leave-tracker>docker run -d -p 3000:80 employee-leave-tracker
36a6a5da0272bbe2d156b6408bbbd29a8704e55c7b36d61c0e5c2e2b325b9dd9
```



Employee Leave Tracker

Leave Applications

ID	Employee	Days	Reason	Status	Action



Employee Leave Tracker

Leave Applications

ID	Employee	Days	Reason	Status	Action
1	felix	5	Fever	Rejected	
2	Lisa	1	Vacation	Approved	
3	Jennie	30	maternity leave	Approved	
4	Joe	3	Sick leave	Pending	Approve Reject

Rubrics:

GitHub commands Syntax & Description (5)	Implementation & Execution (20)	Time management (5)	Viva (10)	Total (40)

Result:

The project successfully demonstrates core Docker operations including container creation ,image management, file transfer , and in-container verification, validating Docker’s efficiency in scalable and portable application deployment.