



Department of Statistics & Computer Science

University of Kelaniya

ACADEMIC YEAR -2023/2024 (Semester ii)

Full-Stack Software Development

Lab Sheet 07

Questions:

In this lab, you will learn to create React applications using four different methods and practice building small components in each project.

Task 01

Project 1: Create a React App Using NPX + JSX.

- Open VS Code.
- Open a folder where you want to create your React project (for example, D:\ReactProjects).
- Open the terminal in VS Code:
- Now run these commands one by one inside the VS Code terminal:
 - **npx create-react-app npx-demo**
 - **cd npx-demo**
 - **npm start**
- Inside src/App.js, create a new component called Welcome.jsx.

A screenshot of the Visual Studio Code interface. The title bar shows "File Go Run Terminal Help". The left sidebar has "project" selected. In the center, there are two tabs: "JS App.js M X" and "Welcome.jsx U X". Below the tabs, the code editor shows the following content:

```
npx-de... Close (Ctrl+F4) Welcome.jsx > ...
1 import React from "react";
2
3 function Welcome() {
4   return <h2>Hello from NPX React App!</h2>;
5 }
6
7 export default Welcome;
8
```

The code is written in JSX, defining a functional component named "Welcome" that returns a heading "Hello from NPX React App!".

```
import './App.css';
import Welcome from './Welcome';

function App() {
  return (
    <div className="App">
      <h1>Project 1: NPX Demo</h1>
      <Welcome />
    </div>
  );
}

export default App;
```

- Save and check the browser output.

Project 2: Create a React App Using Vite + JSX.

- Run the following commands:
 - Npm create vite@latest vite-demo
 - cd vite-demo
 - npm install
 - npm run dev
- Select React + Javascript during project creation.



Vite + React

count is 0

Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

- Open src/app.jsx and add a new component called Box.jsx.

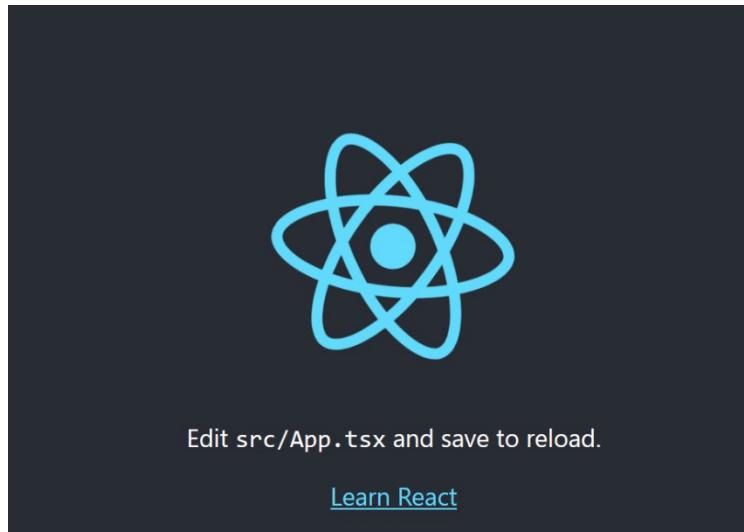
```
vite-demo > src > Box.jsx > ...
1 import React from "react";
2
3 const Box = () => {
4   return (
5     <div style={{ background: "lightblue", padding: "10px" }}>
6       Hello from Vite + JSX!
7     </div>
8   );
9 }
10
11 export default Box;
12 |
```

```
vite-demo > src > App.jsx > ...
1 import React from "react";
2 import Box from "./Box";
3
4 function App() {
5   return (
6     <div>
7       <h1>Project 4: Vite JSX Demo</h1>
8       <Box />
9     </div>
10  );
11 }
12
13 export default App;
14 |
```

- Save and check the browser output.

Project 3: Create a React App using NPX + TSX

- Use the TypeScript template with NPX:
 - npx create-react-app npx-ts-demo --template typescript
 - cd npx-ts-demo
 - npm start



- Open src/App.tsx.
- Create a new component called Message.tsx.

```
npx-ts-demo > src > Message.tsx > ...
1 import React from "react";
2
3 const Message: React.FC = () => {
4   return <h2>Hello from TypeScript + React!</h2>;
5 };
6
7 export default Message;
8 |
```

```
npx-ts-demo > src > App.tsx > ...
1 import React from "react";
2 import Message from "./Message";
3
4 function App() {
5   return (
6     <div>
7       <h1>Project 3: NPX TS Demo</h1>
8       <Message />
9     </div>
10  );
11}
12
13 export default App;
14
```

- Save and check the browser output.

Project 4: Create a React App using Vite + TSX

- Npm create vite@latest vite-ts-demo
 - cd vite-ts-demo
 - npm install
 - npm run dev
- Choose React + Typescript during project setup

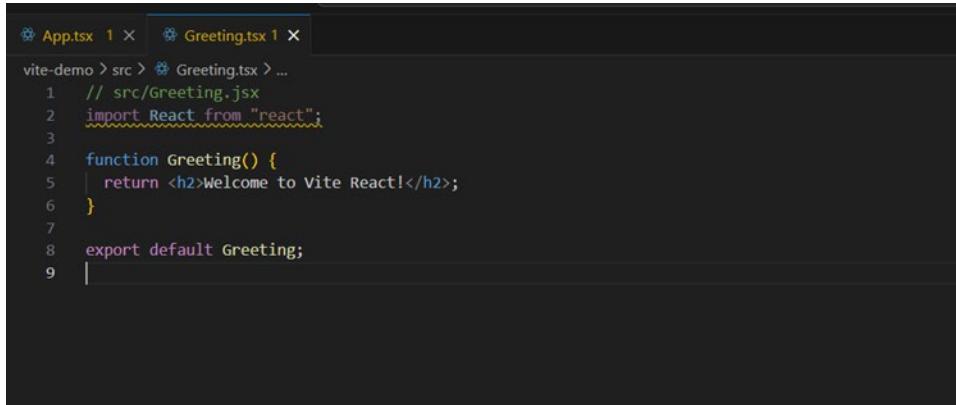


Vite + React

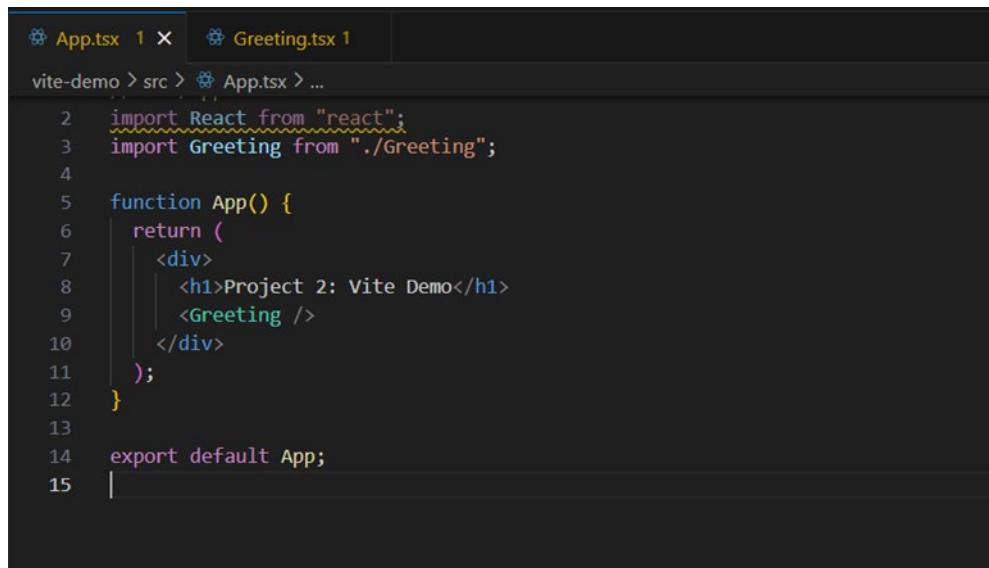
count is 0

Edit src/App.tsx and save to test HMR

- Open src/App.tsx and create a new component called Greeting.tsx.



```
vite-demo > src > Greeting.tsx ...
1 // src/Greeting.jsx
2 import React from "react";
3
4 function Greeting() {
5   return <h2>Welcome to Vite React!</h2>;
6 }
7
8 export default Greeting;
9
```



```
vite-demo > src > App.tsx ...
2 import React from "react";
3 import Greeting from "./Greeting";
4
5 function App() {
6   return (
7     <div>
8       <h1>Project 2: Vite Demo</h1>
9       <Greeting />
10    </div>
11  );
12}
13
14 export default App;
15
```

- Save and check the browser output.

Task 02

1. List two different ways to create a React app.
2. Explain the difference between .jsx and .tsx files.
3. What is the purpose of the node_modules/ folder in a React project?
4. List two important files found inside the public/ folder and explain their roles.
5. What is the function of the index.html file in a React app?
6. What is the purpose of the src/ folder?
7. Take the screenshots of your updated app running in the browser.

Submission Guidelines:

- Prepare a Word document named "**PS/XXXX/XXX_Tutorial07**" or "**EC/XXXX/XXX_Tutorial07**".
- **Include the following elements:**
 - Header with your student number, Tutorial Number and course code.
 - Screenshots of your work.
 - Footer with page number.