

Week 6

6. ReactJS-HOL

Implement a Simple Navigation Menu

Add Basic Routes (install, configure)

Use Routes in React Applications

App.js

The screenshot shows a VS Code editor window with a React application. The Explorer sidebar on the left displays the file structure, with 'App.js' selected under the 'src' directory. The Editor window shows the code for 'App.js', which imports React and Router, and defines a Router component with routes for Home, Trainers, and a Trainer detail. The Terminal at the bottom shows the command 'PS D:\Icognizant_week_practice\Week - 2\Week 6\trainerapp\trainerapp>'.

```

1 | import React from 'react';
2 |
3 | function App() {
4 |   return (
5 |     <BrowserRouter>
6 |       <div>
7 |         <h1>My Academy Trainers App</h1>
8 |         <nav>
9 |           <Link to="/">Home</Link> | <Link to="/trainers">Trainers</Link>
10 |        </nav>
11 |        <hr />
12 |        <Routes>
13 |          <Route path="/" element={<Home /> } />
14 |          <Route path="/trainers" element={<TrainersList data={trainers} /> } />
15 |          <Route path="/trainer/:id" element={<TrainerDetail /> } />
16 |        </Routes>
17 |      </BrowserRouter>
18 |    );
19 |  }
20 |
21 | export default App;
22 |

```

Home.js

The screenshot displays the Visual Studio Code interface for a React project named 'trainerapp'. The Explorer sidebar on the left shows the project structure, with the 'JS Homejs' file selected. The main editor window shows the content of 'Home.js', which defines a simple React component. The Terminal panel at the bottom shows the output of the 'npm run build' command, confirming a successful build and providing the URL to view the application in the browser.

```

File Edit Selection View Go Run Terminal Help
trainerapp
EXPLORER
  TRAINERAPP
    node_modules
    public
    src
      App.js
      Homejs
      index.css
      index.js
      trainer.js
      TrainerDetail.js
      TrainersList.js
      TrainersMock.js
    .gitignore
    package-lock.json
    package.json
    README.md

src > JS Homejs > ...
1 import React from 'react';
2
3 function Home() {
4   return (
5     <div>
6       <h2>Welcome to My Academy Trainers Pages/h2>
7     </div>
8   );
9 }
10
11 export default Home;
12

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS GITLENS
node + - - - - -

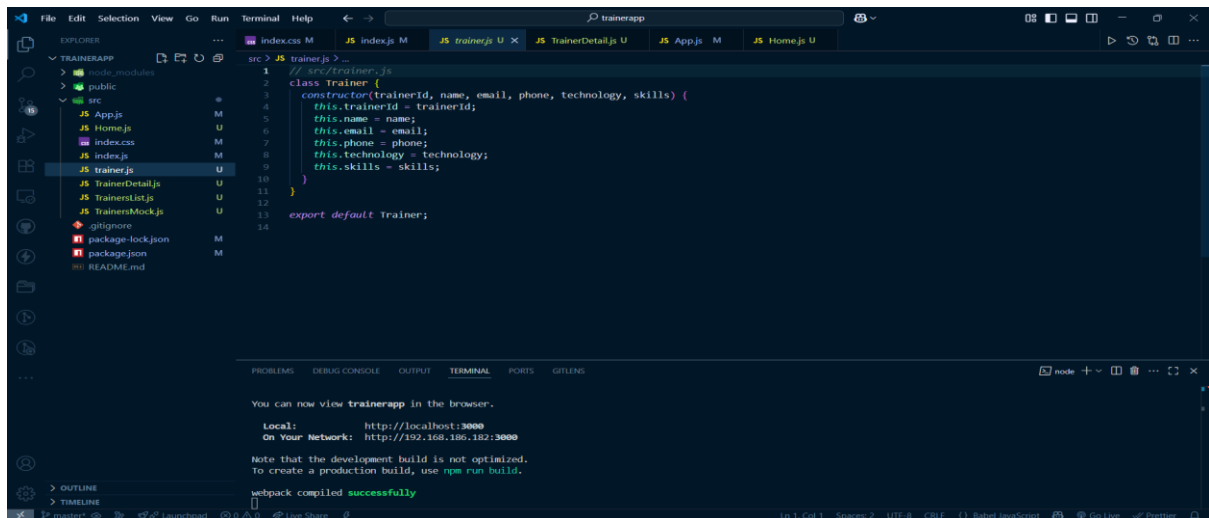
You can now view trainerapp in the browser.

Local:      http://localhost:3000
On Your Network: http://192.168.186.182:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
  
```

Trainer.js



```
src > JS trainer.js > ...
1 // src/trainer.js
2 class Trainer {
3   constructor(trainerId, name, email, phone, technology, skills) {
4     this.trainerId = trainerId;
5     this.name = name;
6     this.email = email;
7     this.phone = phone;
8     this.technology = technology;
9     this.skills = skills;
10  }
11
12  export default Trainer;
```

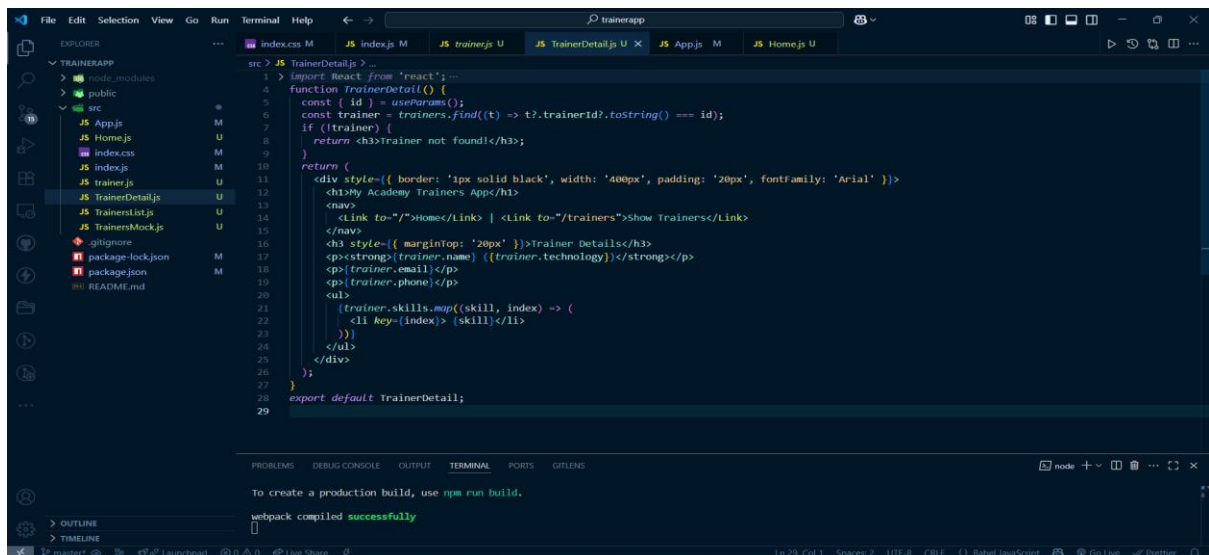
You can now view **trainerapp** in the browser.

Local: <http://localhost:3000>
On Your Network: <http://192.168.186.182:3000>

Note that the development build is not optimized.
To create a production build, use **npm run build**.

webpack compiled **successfully**

TrainerDetails.js

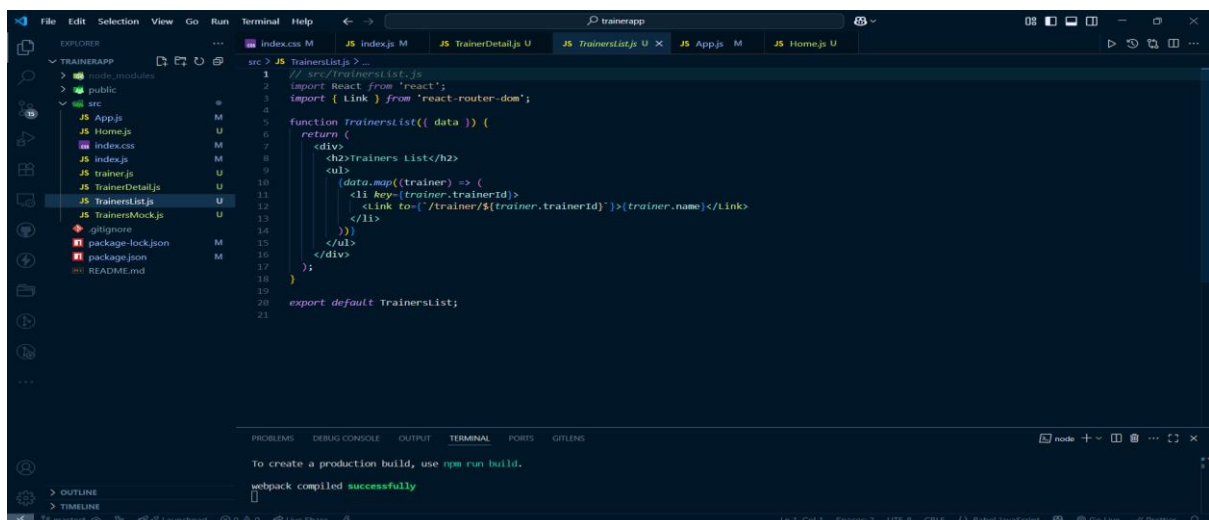


```
src > JS TrainerDetails.js > ...
1 > import React from 'react';
2
3 function TrainerDetail() {
4   const { id } = useParams();
5   const trainer = trainers.find(t => t.trainerId.toString() === id);
6   if (!trainer) {
7     return <h3>Trainer not found!</h3>;
8   }
9
10  return (
11    <div style={{ border: '1px solid black', width: '400px', padding: '20px', fontFamily: 'Arial' }}>
12      <h3>My Academy Trainers App</h3>
13      <nav>
14        <Link to="/">Home</Link> | <Link to="/trainers">Show Trainers</Link>
15      </nav>
16      <h3 style={{ marginTop: '20px' }}>Trainer Details</h3>
17      <p><strong>{trainer.name}</strong> {trainer.technology}</strong></p>
18      <p>{trainer.email}</p>
19      <p>{trainer.phone}</p>
20      <ul>
21        {trainer.skills.map((skill, index) => (
22          <li key={index}> {skill}</li>
23        ))}
24      </ul>
25    </div>
26  );
27 }
28
29 export default TrainerDetail;
```

To create a production build, use **npm run build**.

webpack compiled **successfully**

TrainersList.js

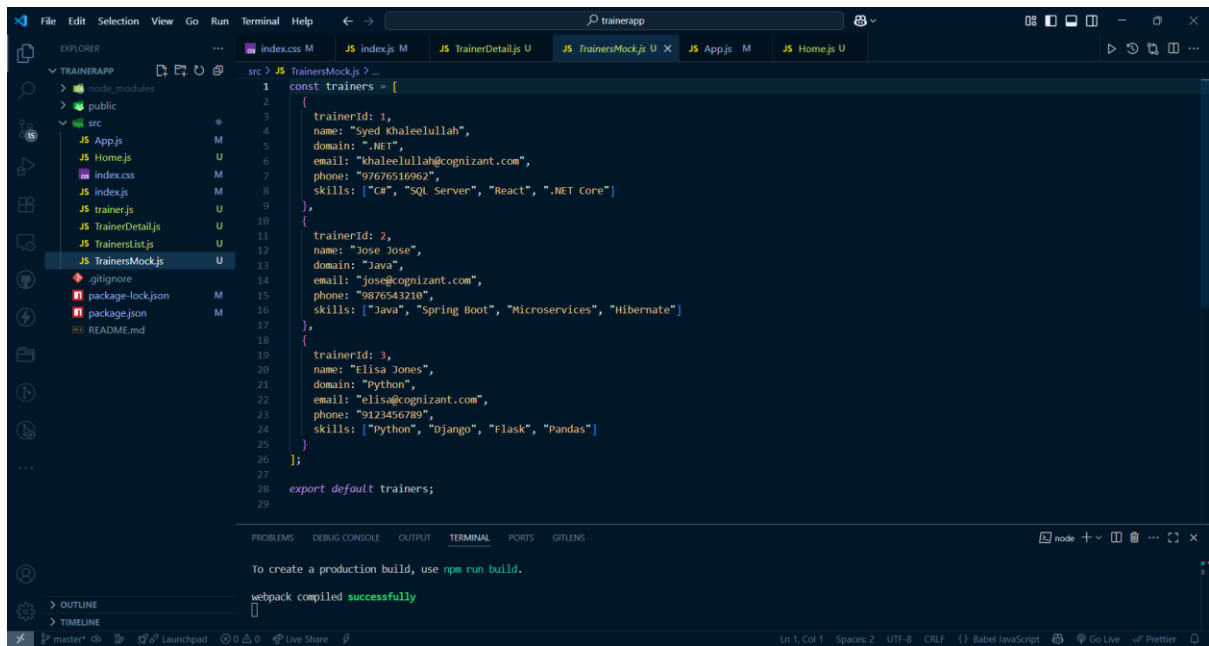


```
src > JS TrainersList.js > ...
1 // src/trainersList.js
2 import React from 'react';
3 import { Link } from 'react-router-dom';
4
5 function TrainersList({ data }) {
6   return (
7     <div>
8       <h2>Trainers List</h2>
9       <ul>
10        {data.map(trainer => (
11          <li key={trainer.trainerId}>
12            <Link to={`/trainer/${trainer.trainerId}`}>{trainer.name}</Link>
13          </li>
14        ))}
15      </ul>
16    </div>
17  );
18 }
19
20 export default TrainersList;
```

To create a production build, use **npm run build**.

webpack compiled **successfully**

TrainersMock.js

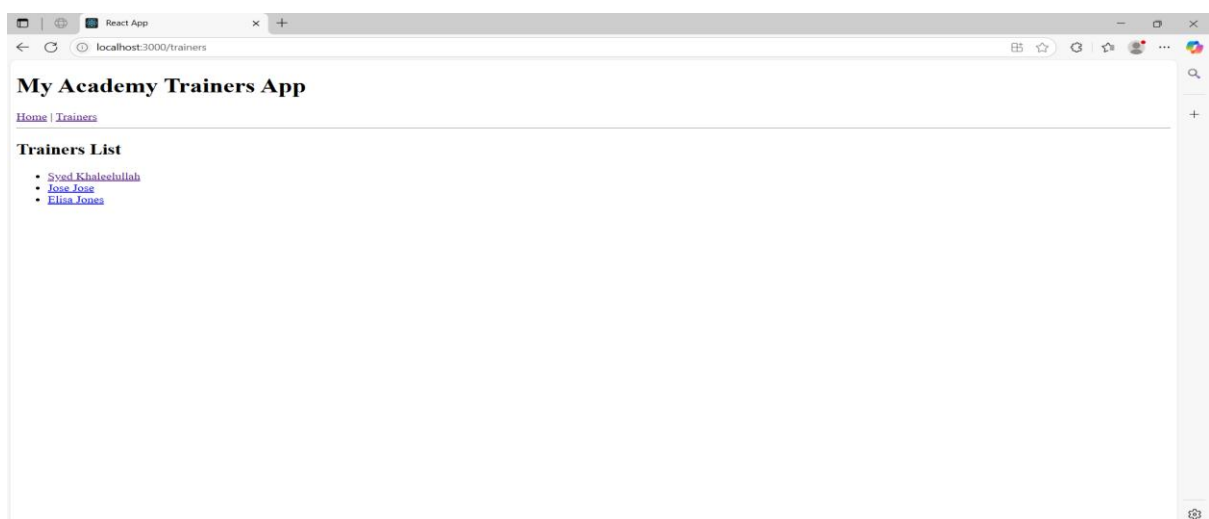
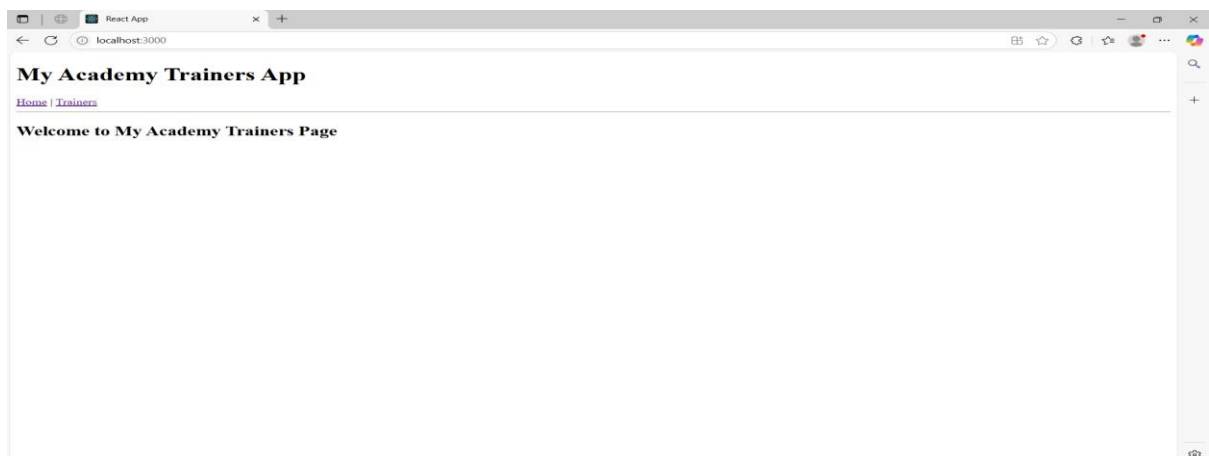


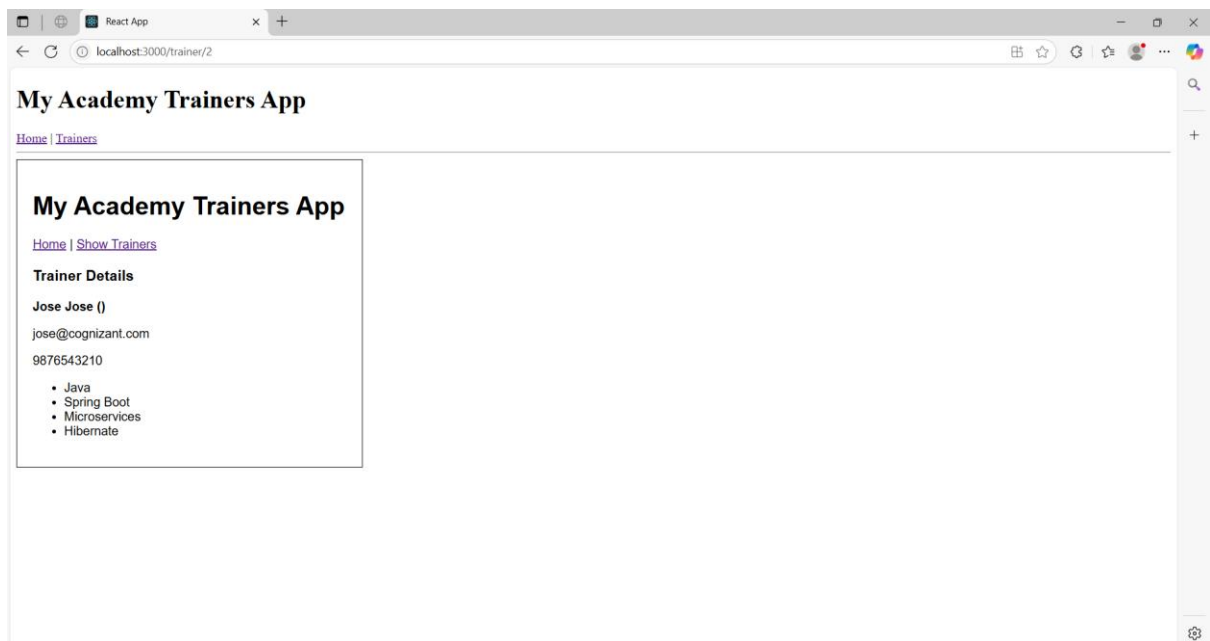
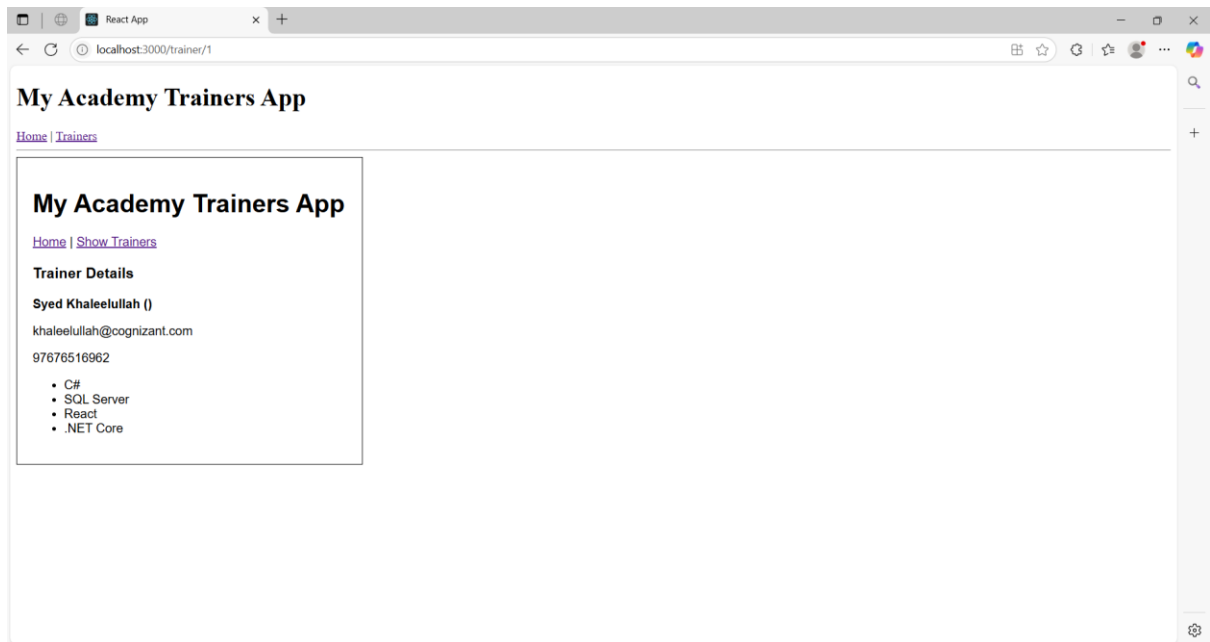
The screenshot shows a VS Code editor window with the file explorer on the left displaying the project structure. The main editor shows the `TrainersMock.js` file with the following code:

```
1 const trainers = [
2   {
3     trainerId: 1,
4     name: "Syed Khaleelullah",
5     domain: ".NET",
6     email: "khaleelullah@cognizant.com",
7     phone: "97676516962",
8     skills: ["C#", "SQL Server", "React", ".NET Core"]
9   },
10  {
11    trainerId: 2,
12    name: "Jose Jose",
13    domain: "Java",
14    email: "jose@cognizant.com",
15    phone: "9876543210",
16    skills: ["Java", "Spring Boot", "Microservices", "Hibernate"]
17  },
18  {
19    trainerId: 3,
20    name: "Elisa Jones",
21    domain: "python",
22    email: "elisa@cognizant.com",
23    phone: "9123456789",
24    skills: ["Python", "Django", "Flask", "Pandas"]
25  }
26 ];
27
28 export default trainers;
```

The terminal at the bottom shows the message: `webpack compiled successfully`.

Output





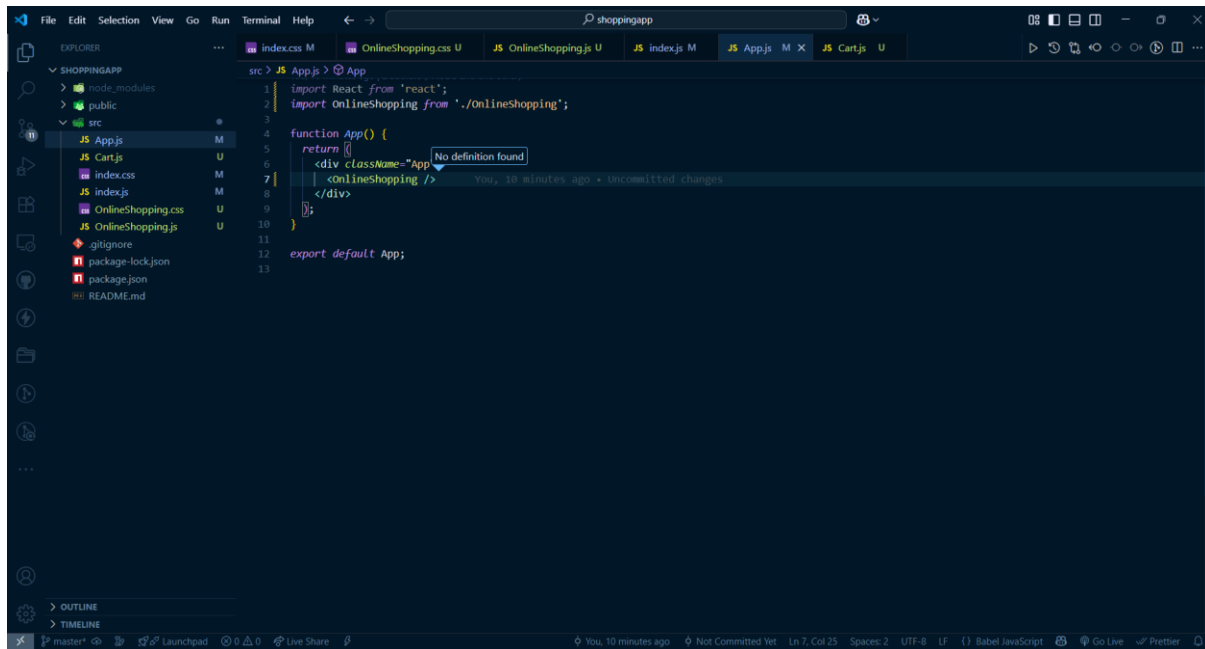
7. ReactJS-HOL

Create a React Application named “shoppingapp” with a class component named “OnlineShopping” and “Cart”.

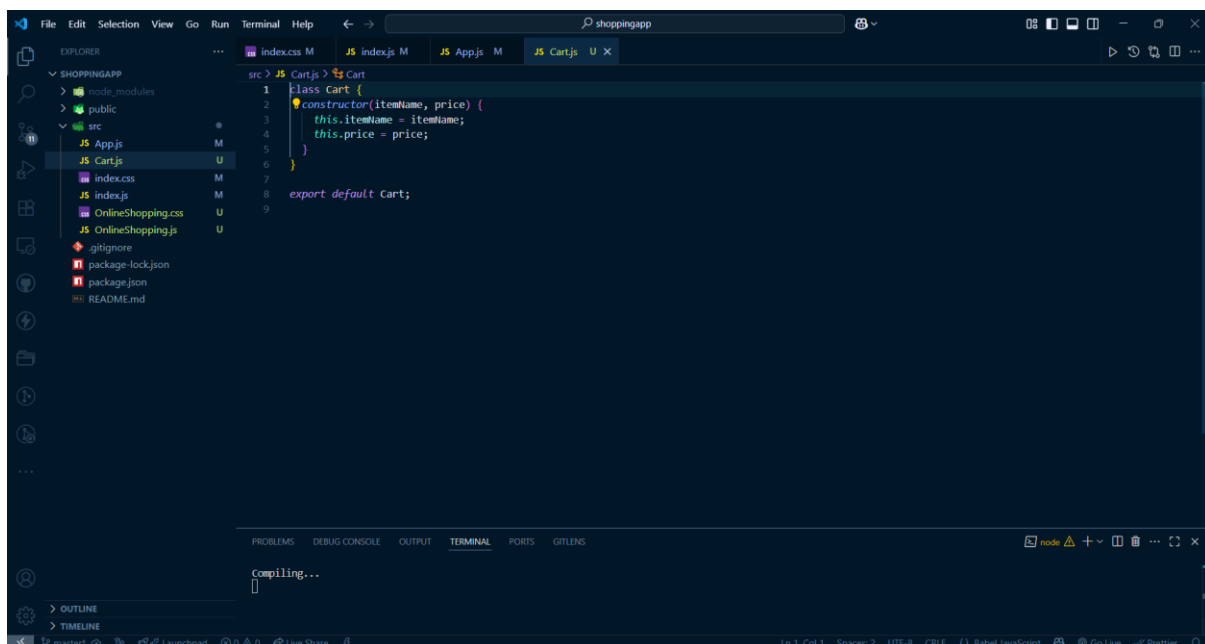
In Cart class, create 2 properties as mentioned below:

Itemname Price

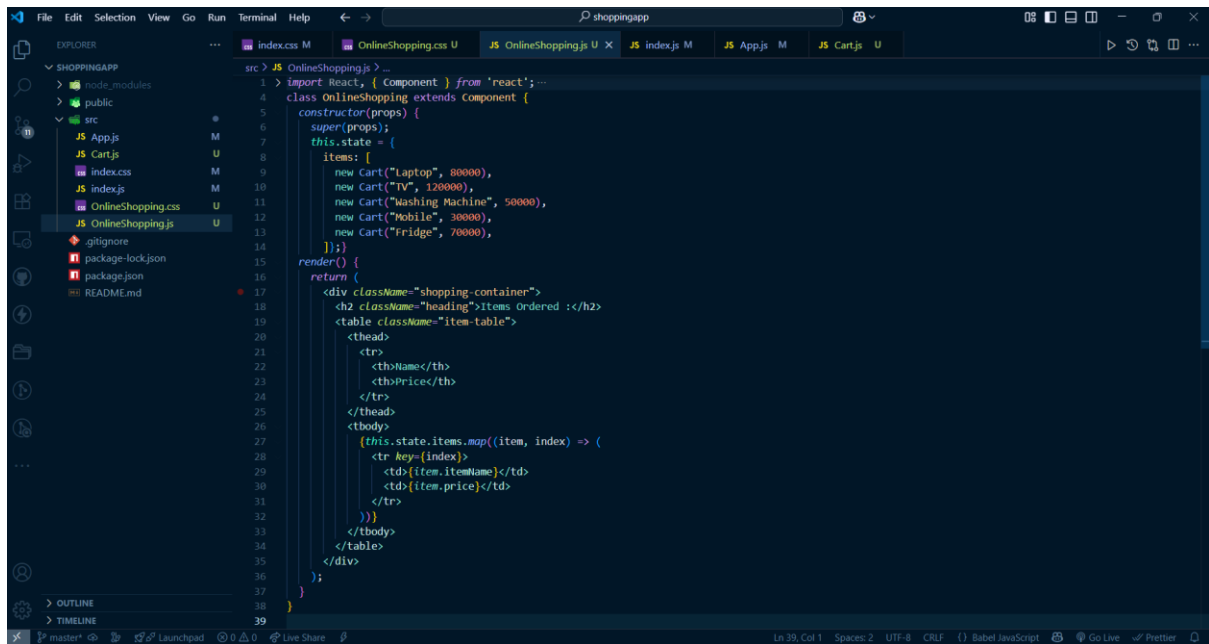
App.js



Cart.js

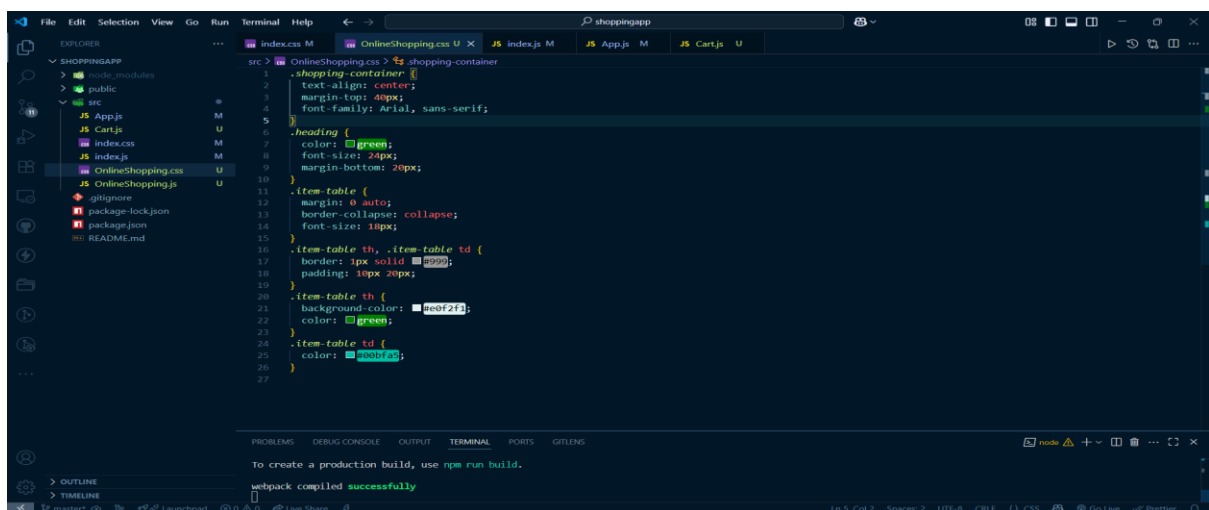


OnlineShopping.js



```
src > JS OnlineShopping.js >
1 > import React, { Component } from 'react';
2 class OnlineShopping extends Component {
3   constructor(props) {
4     super(props);
5     this.state = {
6       items: [
7         new Cart("Laptop", 80000),
8         new Cart("TV", 120000),
9         new Cart("Washing Machine", 50000),
10        new Cart("Mobile", 30000),
11        new Cart("Fridge", 70000),
12      ]
13    };
14  }
15  render() {
16    return (
17      <div className="shopping-container">
18        <h2 className="heading">Items Ordered :</h2>
19        <table className="item-table">
20          <thead>
21            <tr>
22              <th>Name</th>
23              <th>Price</th>
24            </tr>
25          </thead>
26          <tbody>
27            {this.state.items.map((item, index) => (
28              <tr key={index}>
29                <td>{item.itemName}</td>
30                <td>{item.price}</td>
31              </tr>
32            ))}
33          </tbody>
34        </table>
35      </div>
36    );
37  }
38 }
39
```

OnlineShopping.css



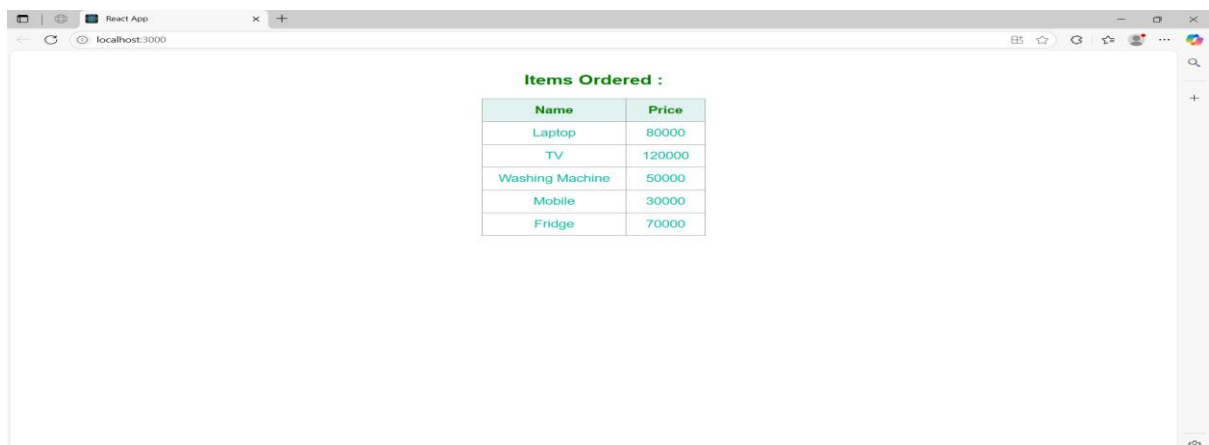
```
src > OnlineShopping.css > shopping-container
1 .shopping-container {
2   text-align: center;
3   margin-top: 40px;
4   font-family: Arial, sans-serif;
5 }
6 .heading {
7   color: green;
8   font-size: 24px;
9   margin-bottom: 20px;
10 }
11 .item-table {
12   margin: 0 auto;
13   border-collapse: collapse;
14   font-size: 18px;
15 }
16 .item-table th, .item-table td {
17   border: 1px solid #999;
18   padding: 10px 20px;
19 }
20 .item-table th {
21   background-color: #e0f2f1;
22   color: green;
23 }
24 .item-table td {
25   color: #000000;
26 }
27
```

Problems DEBUG CONSOLE OUTPUT TERMINAL PORTS GIT LENS

To create a production build, use `npm run build`.

webpack compiled successfully

Output



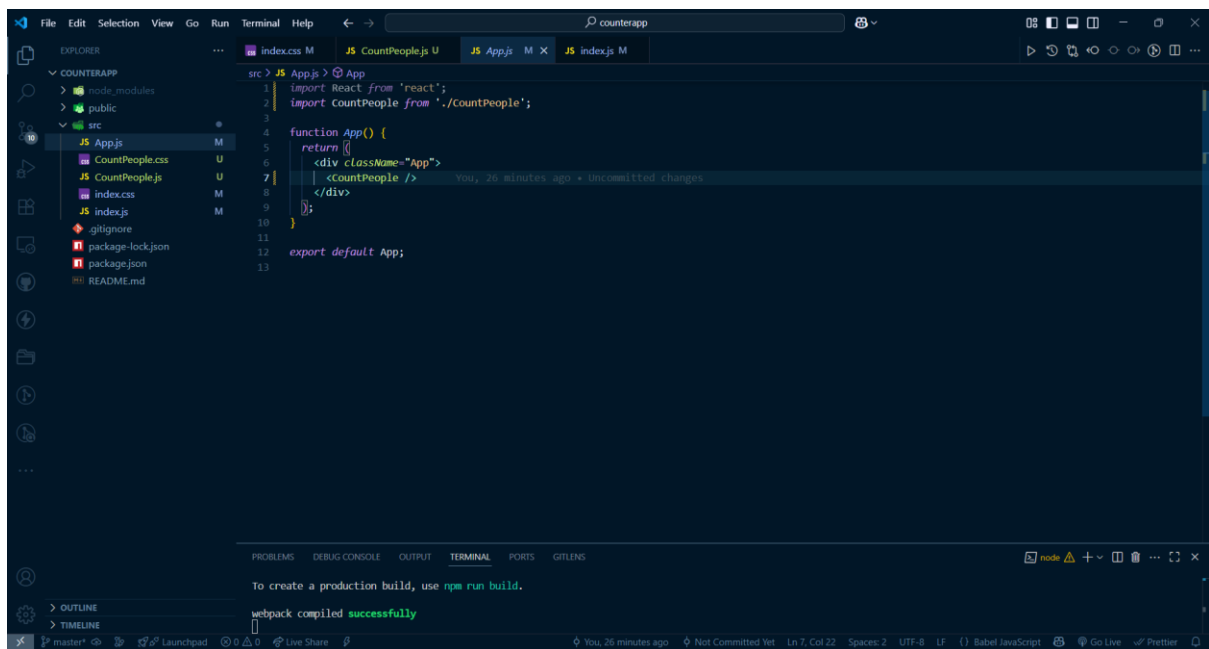
8. ReactJS-HOL

Create a React App “counterapp” which will have a component named “CountPeople” which will have 2 methods.

UpdateEntry() → which will display the number of people who entered the mall.

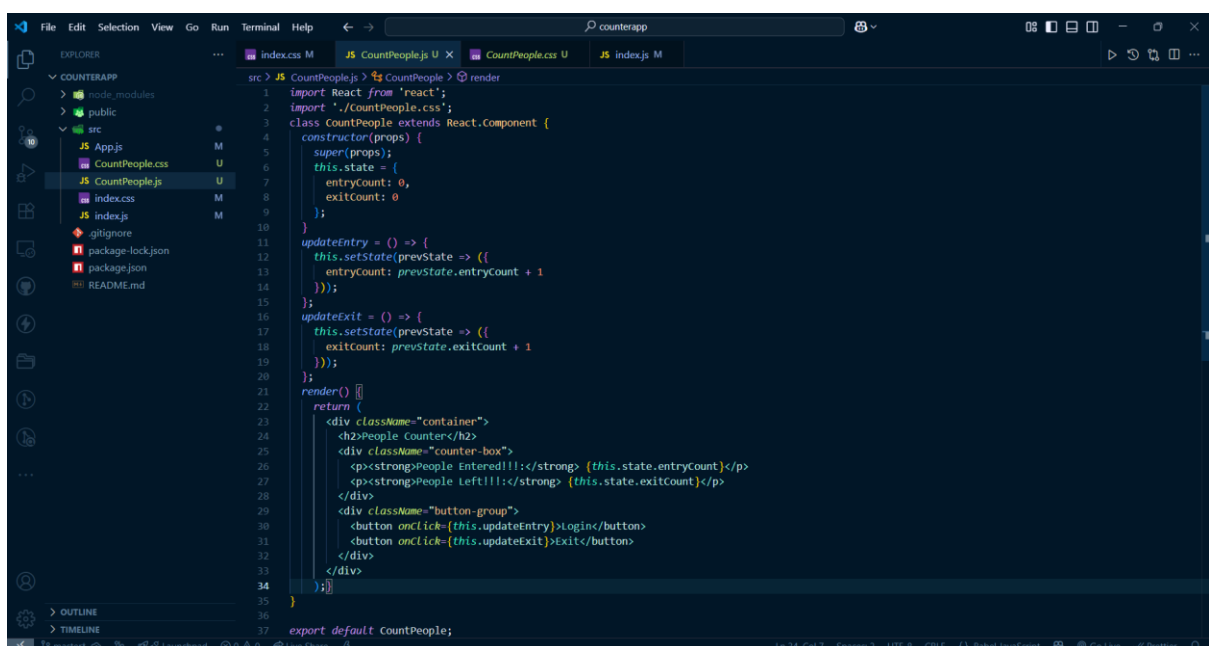
UpdateExit() → which will display the number of people who exited the mall.

App.js

A screenshot of the Visual Studio Code editor. The Explorer panel on the left shows a project named 'counterapp' with a 'src' folder containing 'App.js', 'CountPeople.js', 'index.css', and 'index.js'. The main editor window displays the code for 'App.js'. The code imports React and CountPeople, defines a function App() that returns a JSX element with a div containing a CountPeople component, and exports App as the default export. The status bar at the bottom indicates 'Ln 7, Col 22' and 'Spaces: 2'.

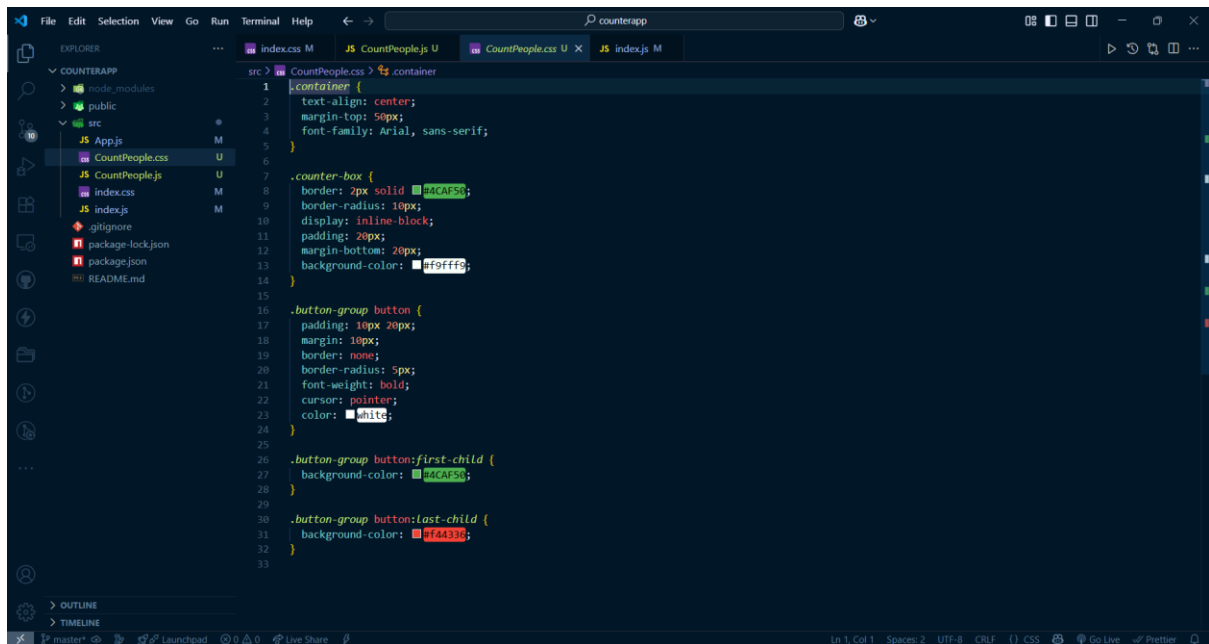
```
1 import React from 'react';
2 import CountPeople from './CountPeople';
3
4 function App() {
5   return (
6     <div className="App">
7       <CountPeople />
8     </div>
9   );
10 }
11
12 export default App;
```

CountPeople.js

A screenshot of the Visual Studio Code editor showing the code for 'CountPeople.js'. The Explorer panel on the left shows the same project structure as the previous screenshot. The main editor window displays the code for 'CountPeople.js'. The code imports React, defines a class CountPeople extending React.Component, and implements stateful methods updateEntry and updateExit, along with a render method that returns a JSX element. The status bar at the bottom indicates 'Ln 34, Col 7' and 'Spaces: 2'.

```
1 import React from 'react';
2 import './CountPeople.css';
3 class CountPeople extends React.Component {
4   constructor(props) {
5     super(props);
6     this.state = {
7       entryCount: 0,
8       exitCount: 0;
9     };
10   }
11   updateEntry = () => {
12     this.setState(prevState => ({
13       entryCount: prevState.entryCount + 1
14     }));
15   };
16   updateExit = () => {
17     this.setState(prevState => ({
18       exitCount: prevState.exitCount + 1
19     }));
20   };
21   render() {
22     return (
23       <div className="container">
24         <h2>People Counter</h2>
25         <div className="counter-box">
26           <p><strong>People Entered!!!:</strong> {this.state.entryCount}</p>
27           <p><strong>People Left!!!:</strong> {this.state.exitCount}</p>
28         </div>
29         <div className="button-group">
30           <button onClick={this.updateEntry}>Login</button>
31           <button onClick={this.updateExit}>Exit</button>
32         </div>
33       </div>
34     );
35   }
36 }
37 export default CountPeople;
```

CountPeople.css



The screenshot shows the Visual Studio Code editor with the 'CountPeople.css' file open. The file is located in the 'src' directory of a project named 'counterapp'. The CSS code defines styles for a container, a counter box, and button groups.

```
1 .container {
2   text-align: center;
3   margin-top: 50px;
4   font-family: Arial, sans-serif;
5 }
6
7 .counter-box {
8   border: 2px solid #4CAF50;
9   border-radius: 10px;
10  display: inline-block;
11  padding: 20px;
12  margin-bottom: 20px;
13  background-color: #f9fff9;
14 }
15
16 .button-group button {
17   padding: 10px 20px;
18   margin: 10px;
19   border: none;
20   border-radius: 5px;
21   font-weight: bold;
22   cursor: pointer;
23   color: white;
24 }
25
26 .button-group button:first-child {
27   background-color: #4CAF50;
28 }
29
30 .button-group button:last-child {
31   background-color: #f44336;
32 }
33 }
```

Output

