Experiment 1

Name : Jetavya Singh Naruka
UID : 23BCS11171
Branch: BE-CSE
Section/Group: KRG-3B
Semester: 6th
Date of Performance: 13-1-2026
Subject Name: Full Stack-II
Subject Code: 23CSH-309

1. Aim:

To design and implement the foundational frontend architecture of the EcoTrack application using modern React practices, Vite tooling, and ES6+ JavaScript features.

2. Objective:-

- To set up a React project using Vite with proper project structure
- To understand component-based architecture in React
- To apply ES6 array methods (map, filter, reduce) for data-driven UI rendering
- To separate concerns using components, pages, and data modules

3. Implementation/Code:

- **Logs.Js :-**

```
export const logs = [
  { id: 1, activity: "Car Travel", carbon: 4 },
  { id: 2, activity: "Electricity Usage", carbon: 6 },
  { id: 3, activity: "Cycling", carbon: 0 },
  { id: 4, activity: "Public Transport", carbon: 2 },
  { id: 5, activity: "Air Travel", carbon: 10 },
];
```

- **Dashboard.Jsx :-**

```
   import {logs} from "../data/logs";


const Dashboard = () => {
    const  track =  logs.reduce((acc, log) => acc + log.carbon, 0);

return (
    <div>
        <h2>Dashboard</h2>
        <p>Total Carbon Footprint: {track} kg</p>

        <ul>
            {logs.map((log) => (
                <li key={log.id}>
                    {log.activity} = {log.carbon} kg
                </li>
```

```
            ))}
        </ul>
    </div>
)

} // this will make the rerendering easy when data changes
// dashboard is a functional componen

export default Dashboard;
```

- **Logs.Jsx :-**

```
import {logs} from "../data/logs";


const Highcarbon =() =>{
const more = logs.filter(log => log.carbon >= 4);
const less = logs.filter(log => log.carbon <= 3 && log.carbon !== 0);

return (
    <div>

        <header style = {{ padding: "0.5rem", backgroundColor: "#ff0000",color:
"white",textAlign: "center", borderRadius: "10px"}}>
        <h2>High Carbon Emission</h2>
        </header>
        <ul>
            {more.map((log) => (
                <li key={log.id}>
                    {log.activity} = {log.carbon} Kg
                </li>
            ))}
        </ul>



        <header style = {{ padding: "0.5rem", backgroundColor: "#62ff00",color:
"white",textAlign: "center", borderRadius: "10px"}}>
        <h2>Low Carbon Emission</h2>
        </header>
        <ul>
            {less.map((log) => (
                <li key={log.id}>
                    {log.activity} = {log.carbon} Kg
                </li>
            ))}
```

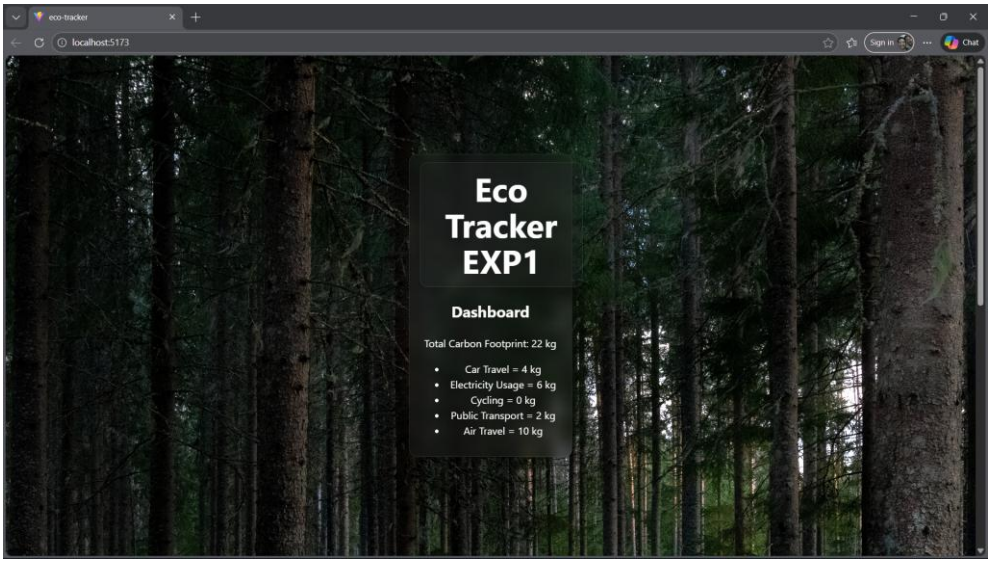
```
        </ul>

    </div>

)
}
  // this component will filter and display logs with carbon emission greater
than or equal to 4 kg


// this component will filter and display logs with carbon emission less than or
equal to 3 kg
export default Highcarbon;
```

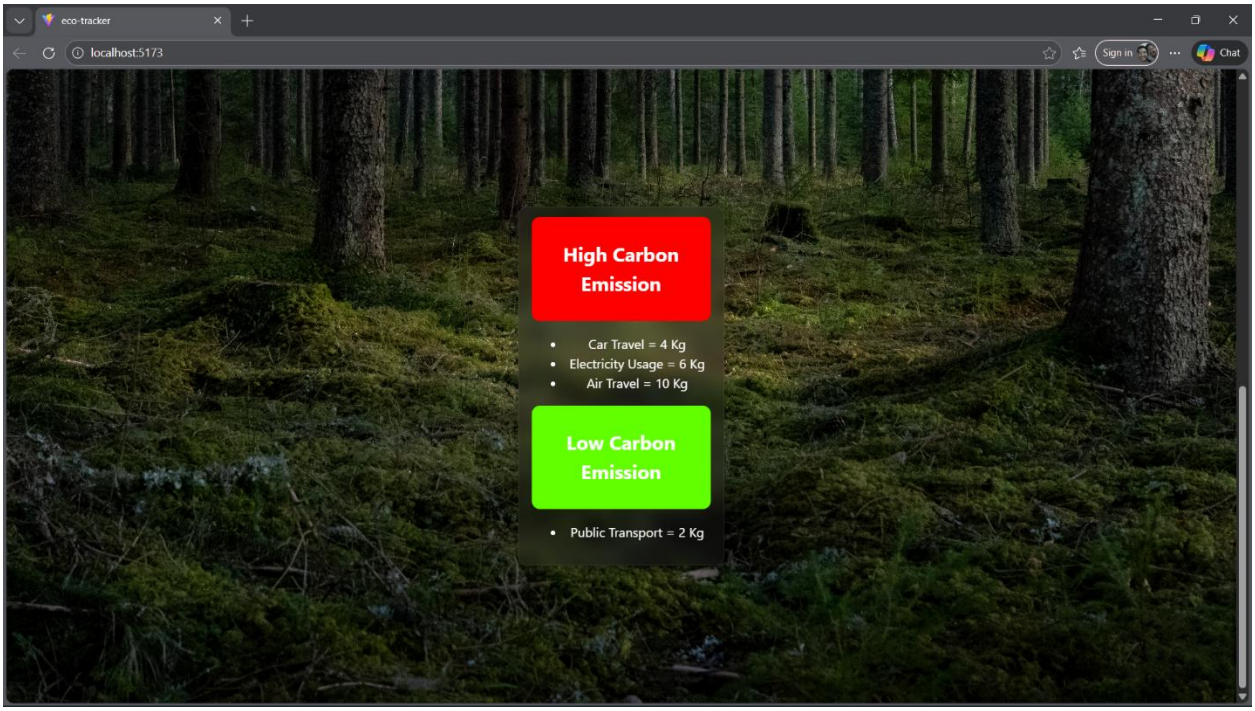
- **App.Jsx:-**

```
import Header from './components/Header.jsx';

import Dashboard from './pages/dashboard.jsx';
import Logs from './pages/Logs.jsx';
import bg from './assets/bg.jpg';
import './App.css';

const App = () =>{
  return (
    <>
     <div style = {{
      backgroundImage: `url(${bg})`,
      backgroundSize: "cover",
      backgroundPosition: "center",
    }}>
      <div className='center-page'>
      <div className='glass-tile'>
      <Header title = "Eco Tracker EXP1" />
    <main Style = {{ pading: "0rem", textAlign: "auto"}}>
        <Dashboard/>

      </main>
      </div>
    </div>
    <div>
     <div className='center-page'>
      <div className='glass-tile'>
      <Logs/>
      </div>
      </div>
```

```
    </div>



    </div>
    </>
  )
}

export default App;
```

4. Output





5. Learning Outcome :-

- Developed an **Eco Tracker application** using React to analyze carbon emissions.
- Implemented a **Dashboard component** to display emission data from JavaScript logs.
- Used **map(), filter(), and reduce()** to classify low and high carbon emission elements.
- Applied **component-based architecture** for better code organization.
- Achieved **dynamic data rendering** based on emission levels.
- Enhanced understanding of **data processing and state-driven UI in React.**