DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment 1

Student Name: Premdeep Singh UID: 23BCS11890
Branch: BE CSE Section/Group: KRG 3-A
Semester: 6th Date of Performance: 12/01/26
Subject Name: Full Stack Development Subject Code: 23CSH-309

Aim:

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

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

Implementation/Code:

logs.js:

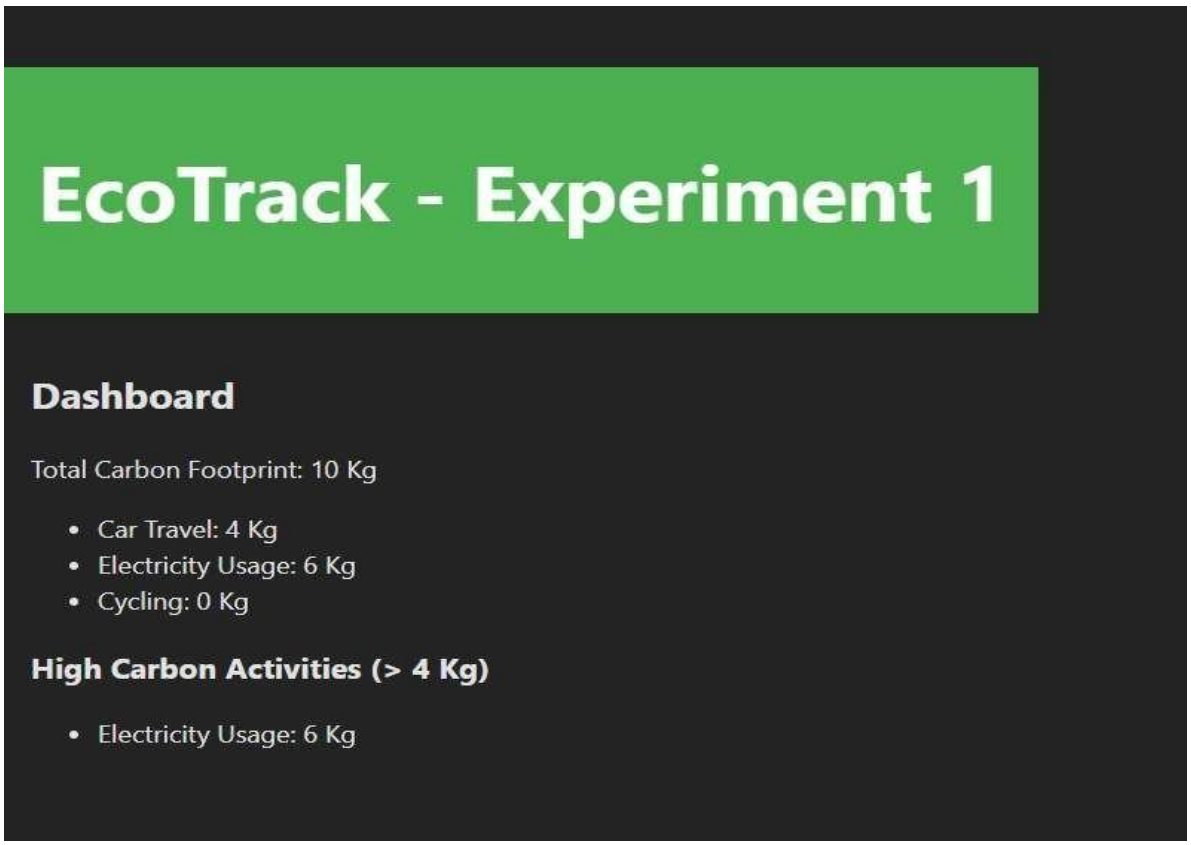
```
export
const logs = [
  { id: 1, activity: "Car Travel", carbon: 4 },
  { id: 2, activity: "Electricity Usage", carbon: 6 },
  { id: 3, activity: "Cycling", carbon: 0 },
];
```

Dashboard.jsx:

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

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

  const highCarbonActivities = logs.filter(
    log => log.carbon > 4
  );
  return (
    <div>
      <h2>Dashboard</h2>
```

```
      <p>Total Carbon Footprint: {calculateTotalCarbon} Kg</p>

      <ul>
        {logs.map((log) => (
          <li key={log.id}>
            {log.activity}: {log.carbon} Kg
          </li>
        ))}
      </ul>
      <h3>High Carbon Activities (&gt; 4 Kg)</h3>
      <ul>
        {highCarbonActivities.map(log => (
          <li key={log.id}>
            {log.activity}: {log.carbon} Kg
          </li>
        ))}
      </ul>
    </div>
  );
};
export default Dashboard;
```

Output:



Learning Outcome:

- Set up a modern React application using Vite and understand the advantages of fast build tools and optimized project scaffolding.

- Design applications using component-based architecture, enabling better code reusability, readability, and separation of concerns.

- Apply ES6 JavaScript features such as map(), filter(), and reduce() to process data and dynamically render UI components.

- Organize project structure effectively by separating data, components, and pages for scalable frontend development.

- Analyze and visualize data in a React application, demonstrating how state-independent data modules can drive UI logic and presentation.