Experiment 1

Student Name: Ayush Rana **UID: 23BCS10602**
Branch: BE-CSE **Section/Group: KRG-3A**
Semester:06 **Date of Performance: 12/01/26**
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:**

App.jsx:

```
import Dashboard from "./pages/Dashboard";

function App() {
  return (
    <div className="app">
      <div className="header">
        <h1> EcoTrack</h1>
        <p>Building a sustainable future with smart tracking</p>
      </div>
      <Dashboard />
    </div>
  );
export default App;
```

ecoData.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 ecoData from "../data/ecoData";
import EcoCard from "../components/EcoCard";

function Dashboard() {

 const totalPoints = ecoData.reduce(
  (sum, item) => sum + item.points, 0
 );

 const topActivity = ecoData.reduce(
  (max, item) => item.points > max.points ? item : max
 );

 return (
  <>
   <div className="stats">
    <div>Total Eco Points: {totalPoints}</div>
    <div>Activities: {ecoData.length}</div>
   </div>

   <div className="card-grid">
    {ecoData.map(item => (
     <EcoCard
      key={item.id}
      activity={item.activity}
      points={item.points}
     />
    ))}
   </div>

   <div className="highlight">
    <h2>       Top Eco Activity</h2>
    <p>{topActivity.activity}</p>
```

```
      <p>{topActivity.points} Points</p>
    </div>
   </>
  );
}

export default Dashboard;
```

index.css :

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: "Segoe UI", sans-serif;
  background: linear-gradient(120deg, #e8f5e9, #ffffff);
  color: #2e7d32;
}

.app {
  max-width: 1100px;
  margin: auto;
  padding: 30px;
}

/* Header */
.header {
  text-align: center;
  margin-bottom: 40px;
}

.header h1 {
  font-size: 2.8rem;
}

.header p {
  color: #4caf50;
  font-size: 1.1rem;
}
```

```
/* Stats */
.stats {
  display: flex;
  justify-content: space-between;
  background: #ffffff;
  padding: 20px;
  border-radius: 14px;
  box-shadow: 0 10px 25px rgba(0,0,0,0.08);
  margin-bottom: 30px;
}

.stats div {
  font-size: 1.2rem;
  font-weight: bold;
}

/* Cards */
.card-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
  gap: 20px;
}

.card {
  background: #ffffff;
  border-radius: 14px;
  padding: 20px;
  box-shadow: 0 12px 25px rgba(0,0,0,0.1);
  transition: all 0.3s ease;
}

.card:hover {
  transform: translateY(-8px);
}

.card h3 {
  margin-bottom: 8px;
}

.card p {
  font-size: 1.2rem;
  font-weight: bold;
```

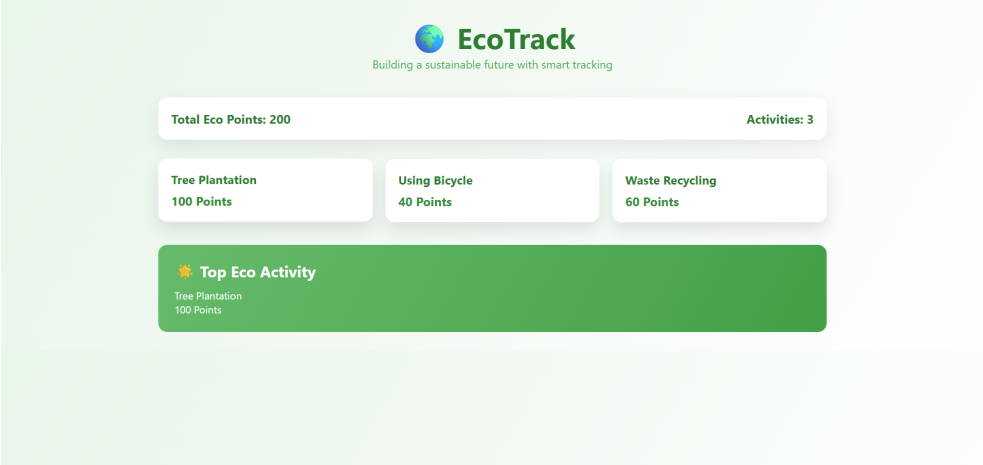
```
  color: #388e3c;
}

/* Highlight */
.highlight {
  margin-top: 35px;
  padding: 25px;
  border-radius: 14px;
  background: linear-gradient(120deg, #66bb6a, #43a047);
  color: white;
}

.highlight h2 {
  margin-bottom: 10px;
}

)
```

4. Output



5. Learning Outcome

1. Ability to set up and configure a React application using Vite, understanding modern frontend tooling and project structure.
2. Understanding of component-based architecture in React, enabling modular, reusable, and maintainable UI development.
3. Practical use of ES6 JavaScript array methods (map, filter, reduce) for implementing data-driven user interface rendering.
4. Application of separation of concerns principles by organizing code into components, pages, and data modules.
5. Capability to design a basic scalable frontend architecture, suitable for future