



Experiment 1.1

Student Name: Ayush Tiwari

Branch: BE CSE

Semester: 6th

Subject Name: Full Stack - II

UID: 23BCS12329

Section/Group: Krg-3B

Date of Performance: 08/01/26

Subject Code: 23CSH-309

1. Aim: To develop a web-based **Carbon Footprint Monitoring Dashboard** that tracks daily activities, calculates total carbon emissions, and categorizes data to help users identify high-emission behaviors and promote environmental awareness.

2. Objective:

Data Aggregation: Consolidate activity logs (e.g., travel, electricity usage, cycling) into a centralized data structure.

Real-time Calculation: Use JavaScript array methods to calculate the total carbon sum ($\sum \text{carbon}$) across all recorded logs.

Data Filtering & Insights: Identify "High Carbon" activities

Identify "Low Carbon" activities (emissions < 3kg).

Extract the "Top 2" highest-emitting activities for immediate user focus.

Visual Representation: Provide a clear UI with color-coded feedback (Red for high emission, Green for low/zero emission) to enhance readability.

3. Implementation:

The project is implemented using React.js and structured into modular components and data files

log.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: "Electricity Usage", carbon: 8 },
  { id: 5, activity: "Car Travel", carbon: 2 },
  { id: 6, activity: "Electricity Usage", carbon: 5 },
  { id: 7, activity: "Cycling", carbon: 0 },
  { id: 8, activity: "Electricity Usage", carbon: 7 },
  { id: 9, activity: "Car Travel", carbon: 3 },
  { id: 10, activity: "Electricity Usage", carbon: 9 },
  { id: 11, activity: "Cycling", carbon: 0 },
  { id: 12, activity: "Electricity Usage", carbon: 6 },
  { id: 13, activity: "Car Travel", carbon: 1 },
]

```

Dashboard.jsx

```

import { logs } from '../data/log';

const Dashboard = () => {
  const totalCarbon = logs.reduce((sum, log) => {
    return sum + log.carbon;
  }, 0);
  const getNonZero=(logs)=>{
    return logs.filter(log=>log.carbon!==0)
  }
  const getCarbonColor = (carbon)=>{
    return carbon >= 4 ? "text-red-600" : "text-green-600";
  }

  const top2Logs = logs
    .slice()
    .sort((a, b) => b.carbon - a.carbon)
    .slice(0, 2);

  return (
    <div>
      <h1>Logs</h1>

      <div>Sum: {totalCarbon}</div>

      <h2>All Logs</h2>

      <ul>
        {getNonZero(logs).map((log) => [
          <li key={log.id}>
            <span className={`font-semibold ${getCarbonColor(log.carbon)} `}>{log.activity}</span>
            <span className={`font-semibold ${getCarbonColor(log.carbon)} `}>
              {log.carbon} kg
            </span>
          </li>
        ])}
      </ul>
    </div>
  )
}

```

Component Architecture

Header.jsx:

```

import React from 'react'

const Header = ({title}) => {
  return (
    <div>
      <h1 className='p-5 bg-orange-300'>{title}</h1>
    </div>
  )
}

export default Header

```

(i) App.jsx

```

import React from 'react'
import { useState } from 'react'
import Dashboard from './pages/dashboard'
import { Logs , LowCarbon} from './pages/logs'
import Header from './components/Header'

function App() {
  const [count, setCount] = useState(0)

  return (
    <div>
      <Header title="data"/>
      <Dashboard />
      <Logs />
      <LowCarbon />
    </div>
  )
}

export default App

```

4. Output

data

Logs

Sum: 51

All Logs

Car Travel4 kg

Electricity Usage6 kg

Electricity Usage8 kg

Car Travel2 kg

Electricity Usage5 kg

Electricity Usage7 kg

Car Travel3 kg

Electricity Usage9 kg

Electricity Usage6 kg

Car Travel1 kg

Top 2 Carbon Emissions

Electricity Usage - 9 kg

Electricity Usage - 8 kg

High Carbon Activities more than 4

Car Travel - 4 kg

Electricity Usage - 6 kg

Electricity Usage - 8 kg

Electricity Usage - 5 kg

Electricity Usage - 7 kg

Electricity Usage - 9 kg

Electricity Usage - 6 kg

Low Carbon Activities less than 3

Car Travel - 4 kg

Electricity Usage - 6 kg

5. Learning Outcome

Proficiency in React Component Architecture

- **Modularization:** Learned how to break down a user interface into reusable functional components like Header, Dashboard, and Logs.
- **Props Management:** Mastered passing data from a parent component (App.jsx) to a child component (Header.jsx) using props to create dynamic titles.
- **State Management:** Implemented the useState hook to initialize and manage application data flow.

2. Advanced JavaScript Data Handling

- **Array Manipulation:** Gained hands-on experience using ES6+ methods to transform raw data:
 - `.reduce()`: Used for calculating the cumulative sum of carbon emissions.
 - `.filter()`: Applied to create specific datasets, such as isolating activities with high carbon output (≥ 4 kg).
 - `.sort()` and `.slice()`: Utilized to rank data and extract the "Top 2" highest values for prioritized reporting.
- **Immutability:** Practiced the use of `.slice()` before sorting to ensure the original data array remains unchanged (data integrity).

3. Logical Conditionals and Styling

- **Dynamic UI Rendering:** Developed logic to change the visual state of the application based on data values (e.g., the `getCarbonColor` function).

- Conditional Formatting: Learned how to integrate ternary operators within JSX to apply CSS classes (Tailwind) conditionally, providing immediate visual feedback to the user regarding their environmental impact.
- #### 4. Data Mapping and Key Management
- Dynamic Lists: Implemented the `.map()` function to iterate over data arrays and render lists of components or elements.