DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Experiment 4

Student Name: Alisha Ameer UID: 23BCC70017

Branch: AIT-CSE Section: 23AIT_KRG-2 A

Semester: 6 Date of Performance: 06/02/2026

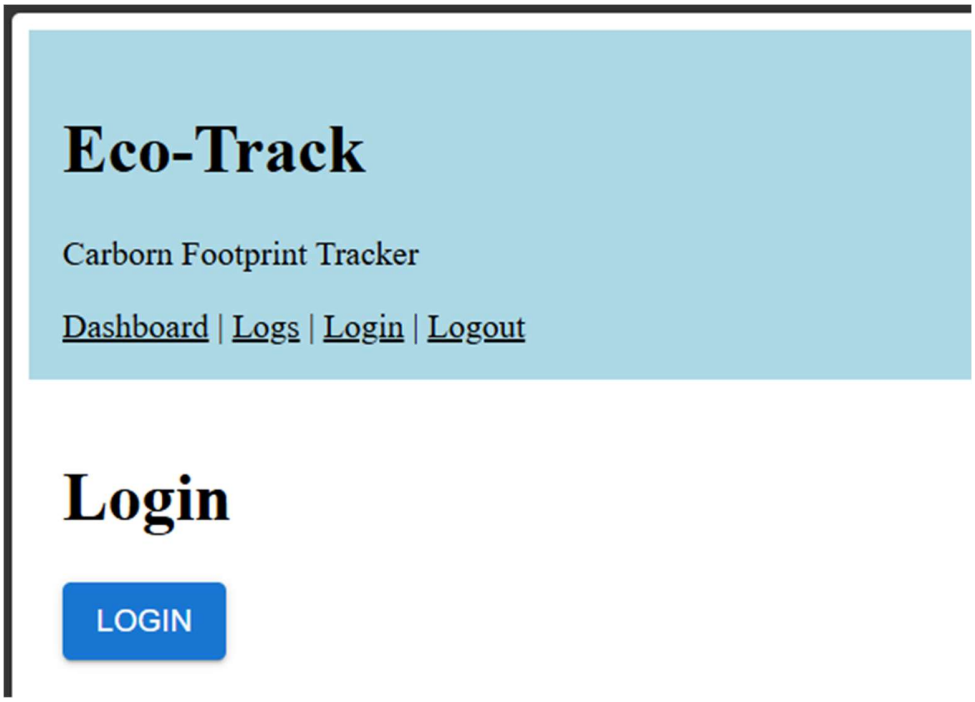
Subject Name: Full Stack II Subject Code: 23CSH-382

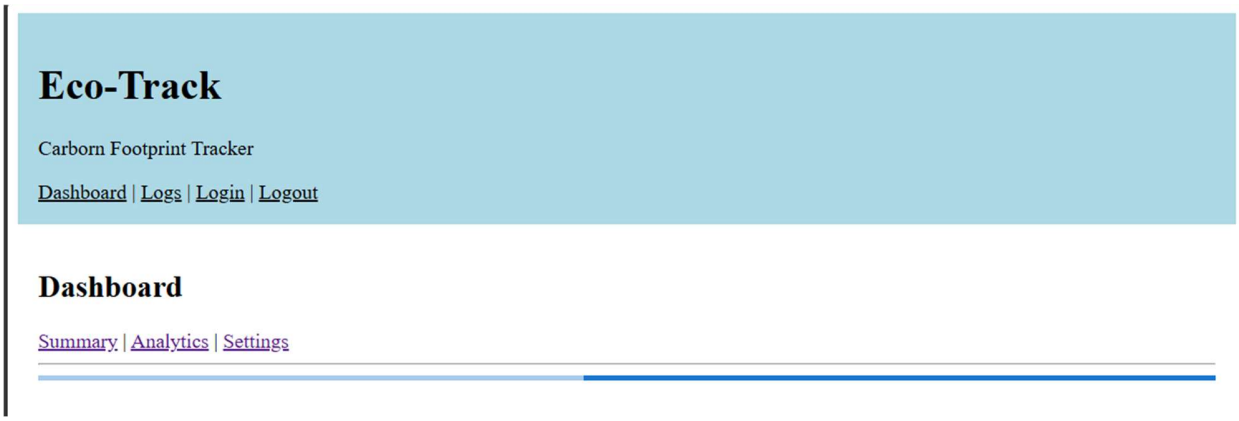
- **Aim:**
  The aim of this experiment is to optimize the performance and user experience of the "Eco-Track" application by implementing memoization techniques, routebased code splitting, and a professional design system using Material UI components.
- **Objectives:**
  The main objectives of this experiment are as follows:
  1. Understand and reduce unnecessary component re-renders for smoother UI performance.
  2. Use React.memo to prevent re-rendering when props remain unchanged.
  3. Apply useMemo to cache expensive filtering and data transformation operations.
  4. Implement useCallback to maintain stable function references for event handlers.
  5. Implement code splitting using React.lazy and Suspense to improve loading performance..

- **Implementation:**
  The following key steps were followed to implement Optimization Techniques, code splitting and designing using MUI in the EcoTrack application:
  1. Installed and integrated the Material UI core library along with the Emotion styling engine.
  2. Refactored major page components (DashboardSummary, Analytics, Logs) using React.lazy() to reduce the initial bundle size.
  3. Implemented a <Suspense> boundary within the DashboardLayout, using location.pathname as a key to trigger instant visual feedback during route transitions.
  4. Used the Material UI LinearProgress component to create visually appealing loading fallbacks for asynchronous chunks.

5. Optimized the Logs.jsx component by wrapping high-carbon filtering logic in useMemo, ensuring recalculation only when Redux store data changes.
6. Wrapped the Redux dispatch trigger for fetching logs in useCallback to maintain stable function references across re-renders.
7. Refactored or replaced the custom Button component with MUI Button components using props such as variant="contained" and color="primary" to align with the Eco-Track theme.

- **Output:**





- **Results:**
  The Eco-Track application was successfully enhanced into a high-performance, enterprise-ready web application. Code splitting significantly reduced the initial load time, while memoization techniques eliminated redundant computations, resulting in a more responsive interface even with increasing data volume. The integration of Material UI transformed the design from a basic prototype into a professional-grade dashboard, showcasing effective use of React performance optimization techniques and modern design systems.

- **Learning Outcomes:**
  After completing this experiment, I have learnt to:
  1. Learned to detect and fix unnecessary re-renders.
  2. Used useMemo to optimize expensive computations.
  3. Applied useCallback to maintain stable function references.
  4. Implemented code splitting to improve Time to Interactive.
  5. Built consistent and professional UI using Material UI.