



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 4

Student Name: Divyansh Manhas

UID: 23BAI70239

Branch: BE CSE AIML

Section/Group: 23AIT-KRG G1

Semester: 6th

Date of Performance: 4 February 2026

Subject Name: Full Stack

Subject Code: 23CSH-382

1. Title:

Redux Toolkit & Asynchronous State Management in EcoTrack

2. Aim:

To optimize the performance of the EcoTrack React application using memoization techniques and code splitting, and to enhance the user interface using enterprise-grade Material UI components.

3. Objective:

After completing this experiment and its follow-up tasks, the student will be able to:

1. Understand the causes of unnecessary re-renders in React applications
2. Optimize React components using `React.memo` to prevent avoidable re-renders
3. Apply `useMemo` to efficiently compute derived data and avoid redundant calculations
4. Use `useCallback` to memoize event handler functions and improve component performance
5. Implement lazy loading of components and routes using `React.lazy` and `Suspense`
6. Reduce initial bundle size and improve application load performance through code splitting

7. Enhance the visual appearance and usability of the EcoTrack application using Material UI components
8. Design a clean, consistent, and responsive user interface using Material UI layouts and typography

4. Implementation/Code:

- logSlice.jsx:

```

5. import { createSlice, createAsyncThunk } from
"@reduxjs/toolkit";
6.
7. export const fetchLogs = createAsyncThunk(
8.   "logs/fetchLogs",
9.   async () => {
10.     await new Promise((resolve) => setTimeout(resolve,
1000));
11.
12.     return [
13.       { id: 1, activity: "Car Travel", carbon: 4 },
14.       { id: 2, activity: "Electricity Usage", carbon:
6 },
15.       { id: 3, activity: "Cycling", carbon: 0 },
16.     ]
17.   }
18. )
19.
20. const logsSlice = createSlice({
21.   name : "logs",
22.   initialState : {
23.     data : [],
24.     status : "idle",
25.     error : null,
26.   },
27.   reducers : {},
28.   extraReducers : (builder) => {
29.     builder
30.       .addCase(fetchLogs.pending, (state, action) =>{
31.         state.status = "loading";
32.       })

```

```
•
33.          .addCase(fetchLogs.fulfilled, (state, action) =>{
34.              state.status = "success";
35.              state.data = action.payload;
36.          })
37.          .addCase(fetchLogs.rejected, (state, action) =>{
38.              state.status = "failed";
39.              state.error = action.error.message;
40.          })
41.      }
42.  })
43.
44. export default logsSlice.reducer;
```

Logs.jsx:

```
import { useEffect, useMemo, useCallback, memo } from "react";
import { useDispatch, useSelector } from "react-redux";
import { fetchLogs } from "../store/logSlice";

const Logs = () => {
  const dispatch = useDispatch();
  const { data, status, error } = useSelector((state) => state.logs);

  const handleRefresh = useCallback(() => {
    dispatch(fetchLogs());
  }, [dispatch]);

  useEffect(() => {
    if (status === "idle") {
      dispatch(fetchLogs());
    }
  }, [status, dispatch]);

  const totalCarbon = useMemo(() => {
    return data.reduce((acc, log) => acc + log.carbon, 0);
  }, [data]);

  if (status === "loading") {
    return (
      <div className="container">
        <div className="logs-container">
```

```
        <p style={{ textAlign: 'center', color: '#667eea' }}>Loading Logs...</p>
    </div>
</div>
);
}

if (status === "failed") {
    return (
        <div className="container">
            <div className="logs-container">
                <p style={{ textAlign: 'center', color: '#e74c3c' }}>Error: {error}</p>
            </div>
        </div>
    );
}

return (
    <div className="container">
        <div className="logs-container">
            <h1 className="logs-title">Activity Logs</h1>

            <div className="total-carbon">
                Total Carbon Footprint: {totalCarbon} kg CO2
            </div>

            <ul className="logs-list">
                {data.map((log) => (
                    <li key={log.id} className="log-item">
                        <span className="log-activity">{log.activity}</span>
                        {" - "}
                        <span className="log-carbon">{log.carbon} kg CO2</span>
                    </li>
                )));
            </ul>

            <button className="refresh-btn" onClick={handleRefresh}>
                Refresh Logs
            </button>
        </div>
    </div>
);
```

```
        •
    </button>
  </div>
</div>
);
};

export default memo(Logs);
```

Store.jsx:

```
import { configureStore } from "@reduxjs/toolkit";
import logsReducer from "./logSlice"

const store = configureStore({
  reducer : {
    logs : logsReducer,
  },
});

export default store;
```

- AuthContext.jsx:

```
45. import { createContext, useContext, useState } from "react";
46.
47. const AuthContext = createContext(null);
48.
49. export const AuthProvider = ({children}) => {
50.   const [isAuthenticated, setIsAuthenticated] = useState(false);
51.
52.   return (
53.     <AuthContext.Provider value = {{isAuthenticated,
54.     setIsAuthenticated}}>
55.       {children}
56.     </AuthContext.Provider>
57.   )
58.
59. export const useAuth = () => useContext(AuthContext);
```

- DashboardLayout.jsx:

```
60. import { Link, Outlet } from "react-router-dom";
61. import { memo } from "react";
62.
63. const DashboardLayout = () => {
64.   return (
65.     <div className="container">
```

```

66.      <div className="dashboard-container">
67.        <h1 className="dashboard-title">Dashboard</h1>
68.
69.        <div className="nav-buttons">
70.          <Link to="summary" className="nav-btn">
71.            Summary
72.          </Link>
73.
74.          <Link to="analytics" className="nav-btn">
75.            Analytics
76.          </Link>
77.
78.          <Link to="settings" className="nav-btn">
79.            Settings
80.          </Link>
81.        </div>
82.
83.        <Outlet />
84.      </div>
85.    </div>
86.  );
87. };
88.
89. export default memo(DashboardLayout);
90.
```

Header.jsx:

```

import { memo } from "react";
import { Link } from "react-router-dom";
import { useAuth } from "../context/AuthContext";

const Header = () => {
  const { isAuthenticated } = useAuth();

  return (
    <header style={headerStyle}>
      <div style={containerStyle}>
        <h1 style={logoStyle}>🌐 EcoTrack</h1>

```

```

<nav style={navStyle}>
  <Link to="/" style={linkStyle}>
    Dashboard
  </Link>

  <Link to="/logs" style={linkStyle}>
    Logs
  </Link>

  {isAuthenticated ? (
    <Link to="/logout" style={linkStyle}>
      Logout
    </Link>
  ) : (
    <Link to="/login" style={linkStyle}>
      Login
    </Link>
  )}
</nav>
</div>
</header>
);
};

}

```

App.jsx:

```

import { Route, Routes } from "react-router-dom";
import { Suspense, lazy } from "react";
import ProtectedRoute from "./routes/ProtectedRoute";
import Header from "./components/Header";

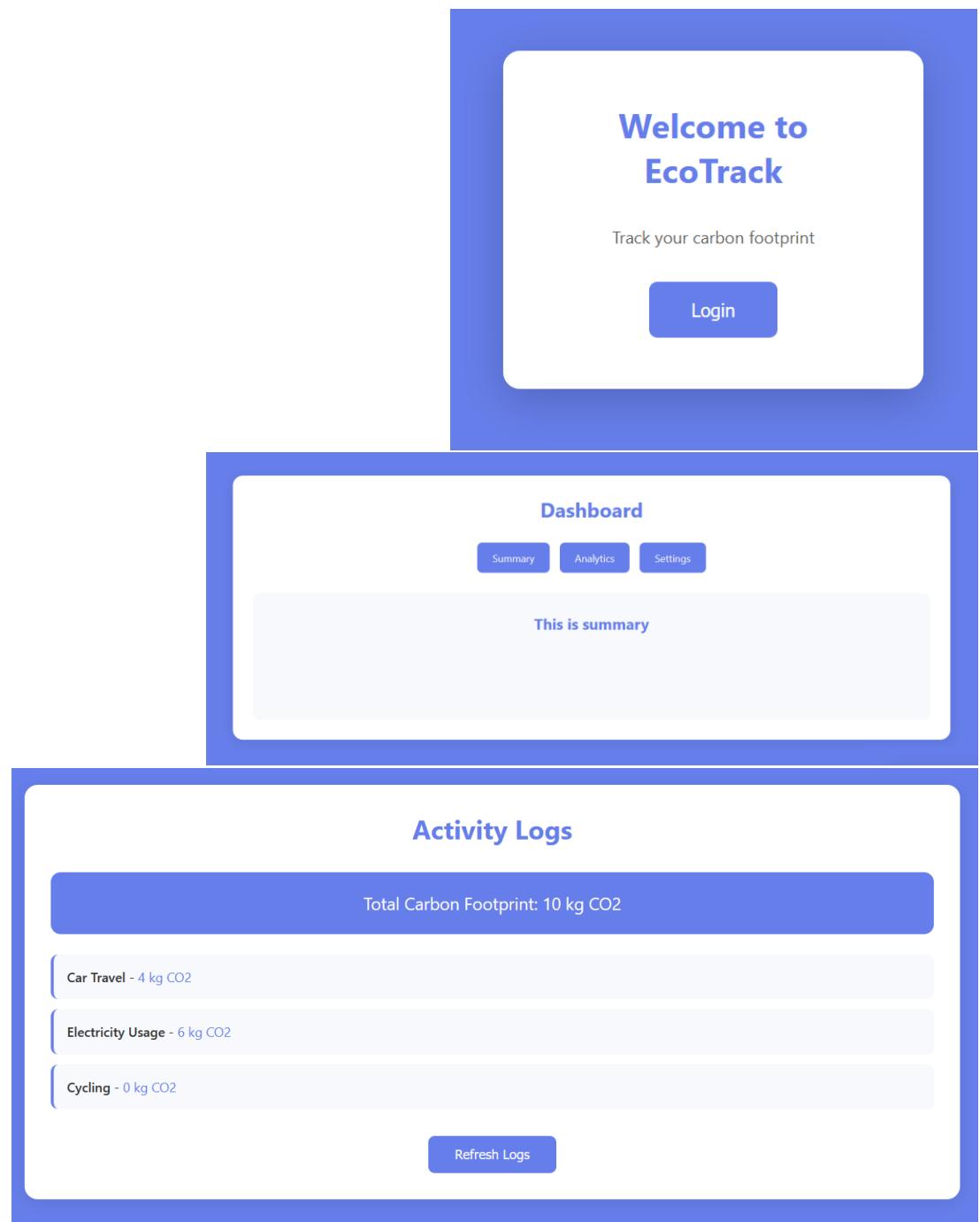
const Login = lazy(() => import("./pages/Login"));
const Logout = lazy(() => import("./pages/Logout"));
const DashboardLayout = lazy(() => import("./pages/DashboardLayout"));
const DashboardSummary = lazy(() => import("./pages/DashboardSummary"));
const DashboardSettings = lazy(() => import("./pages/DashboardSettings"));
const DashboardAnalytics = lazy(() => import("./pages/DashboardAnalytics"));
const Logs = lazy(() => import("./pages/Logs"));

```

```
function App() {
  return (
    <>
    <Header />
    <Suspense fallback={<h2>Loading...</h2>}>
      <Routes>
        <Route path="/login" element={<Login />} />
        <Route path="/logout" element={<Logout />} />
        <Route
          path="/"
          element={
            <ProtectedRoute>
              <DashboardLayout />
            </ProtectedRoute>
          }
        >
          <Route index element={<DashboardSummary />} />
          <Route path="settings" element={<DashboardSettings />} />
          <Route path="summary" element={<DashboardSummary />} />
          <Route path="analytics" element={<DashboardAnalytics />} />
        </Route>
        <Route
          path="/logs"
          element={
            <ProtectedRoute>
              <Logs />
            </ProtectedRoute>
          }
        />
      </Routes>
    </Suspense>
  </>
);
}

export default App;
```

91.Output



92.Learning Outcome

- We learnt about React Apps and how to create them.
- We learnt about redux and it's components.

- We learnt about the use of thunks and Slices.
- We learnt about Authentication and index pages • We learnt the use of MaterialUI and useSelector.
- We learnt about the flow of a React project.