DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment - 3

Student Name : Jobanjot Singh Grewal	**UID :** 23BIA50005
Branch : BE CSE (AIML)	**Section/Group :** 23AIT-KRG/G1
Semester : 6th SEM	**Date of Performance :** 28/01/26
Subject Name : Full Stack II	**Subject Code :** 23CSH-382

1. Aim :

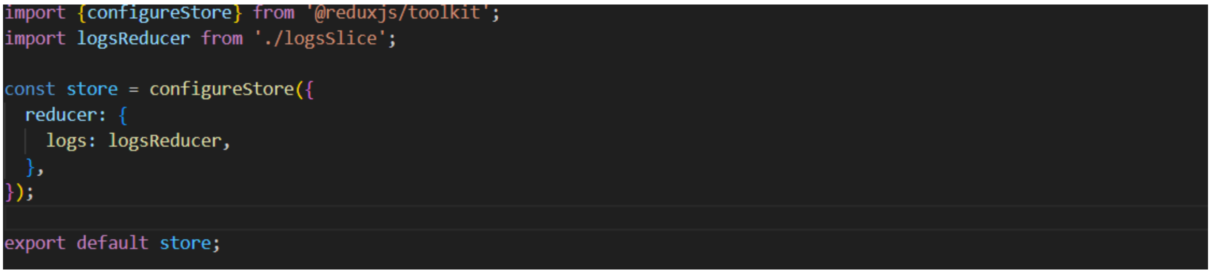
To implement centralized state management in the EcoTrack application using Redux Toolkit and to handle asynchronous data operations using Redux async thunks with proper loading and error states.

2. Objective :

- Configure a Redux store in a React application using Redux Toolkit
- Create and integrate Redux slices for managing application data
- Implement asynchronous actions using Redux async thunks
- Manage loading, success, and error states during asynchronous operations
- Connect React components to Redux state using React-Redux hooks
- Trigger asynchronous data fetching through Redux actions from UI components
- Use Redux state to derive filtered views without modifying the global store
- Enhance user experience by handling refresh actions and improving async UI feedback

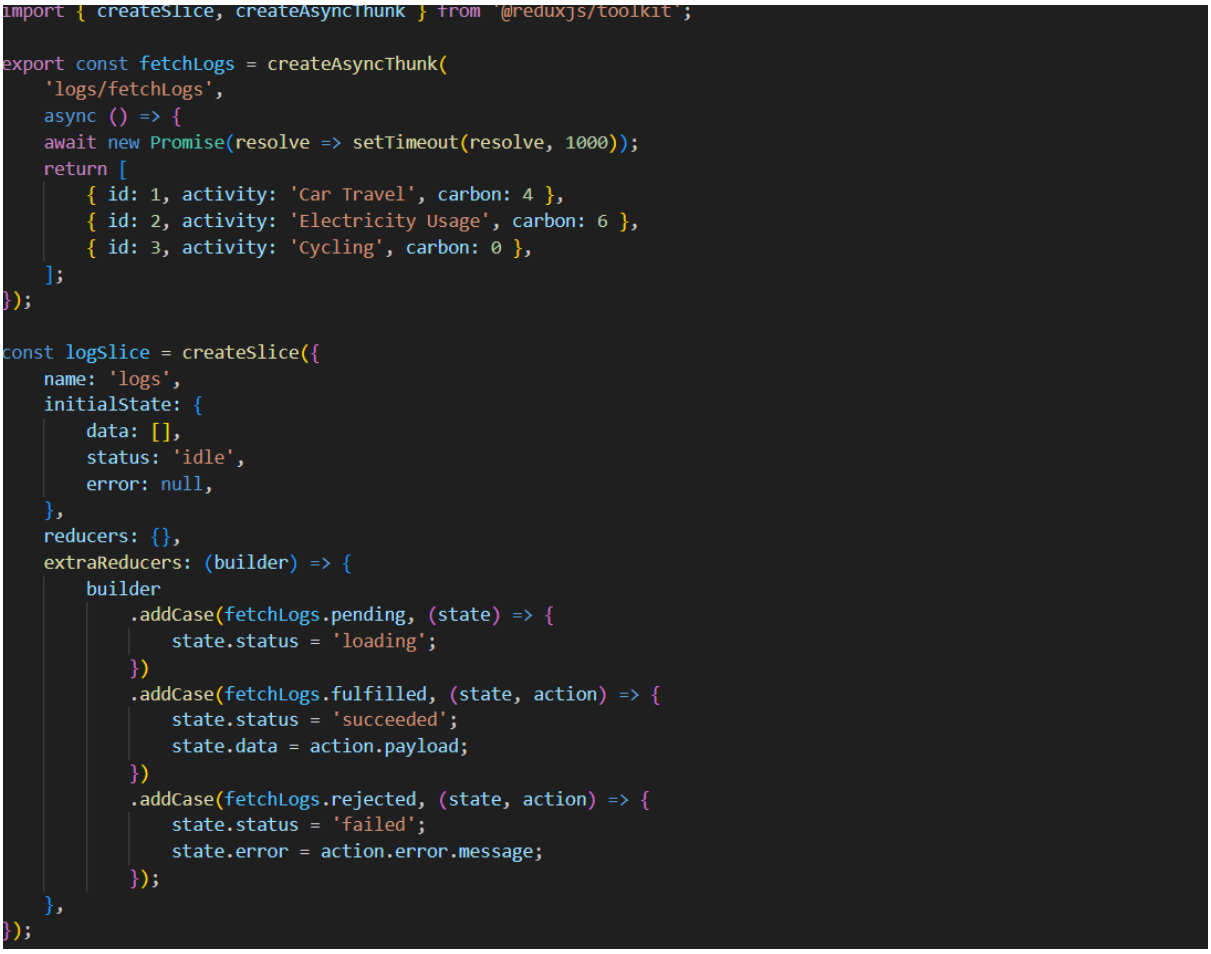
3. Implementation/Code :

Store.js:



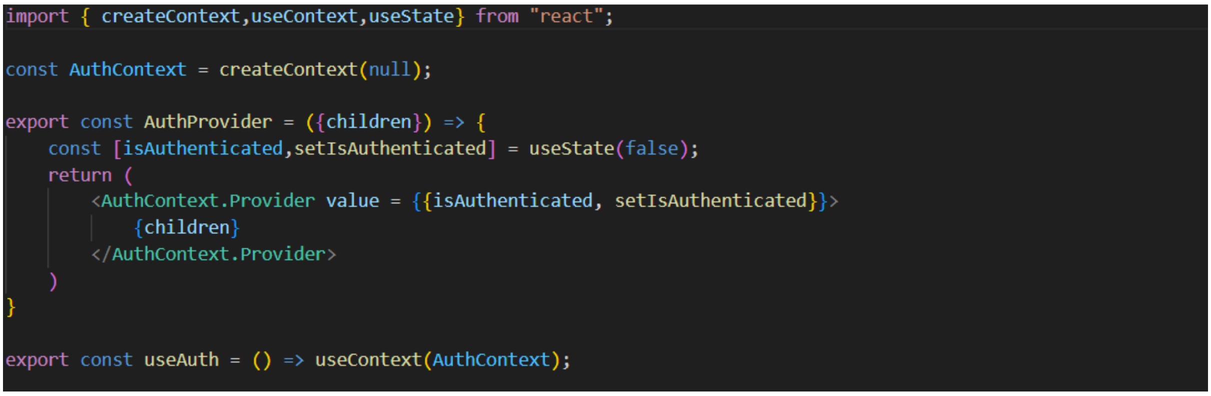
```
import {configureStore} from '@reduxjs/toolkit';
import logsReducer from './logsSlice';

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

export default store;
```

logSlice.js:



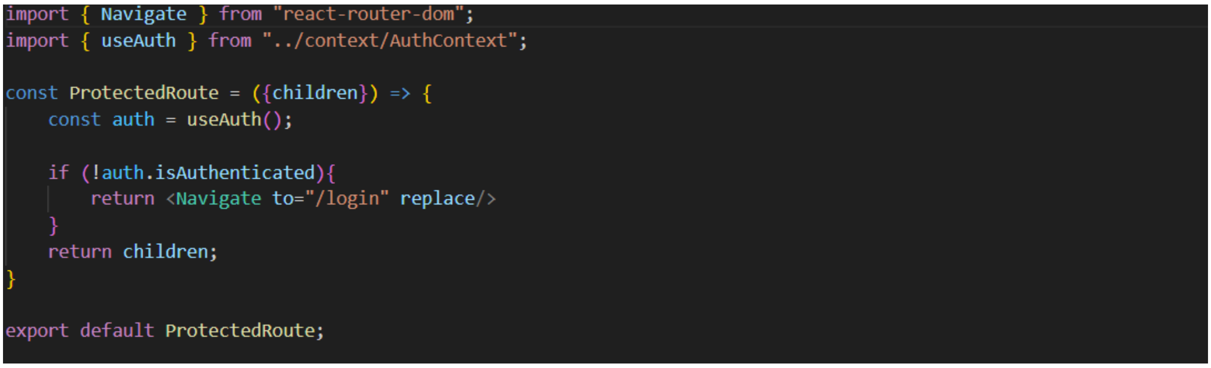
```
import { createSlice, createAsyncThunk } from '@reduxjs/toolkit';

export const fetchLogs = createAsyncThunk(
    'logs/fetchLogs',
    async () => {
    await new Promise(resolve => setTimeout(resolve, 1000));
    return [
        { id: 1, activity: 'Car Travel', carbon: 4 },
        { id: 2, activity: 'Electricity Usage', carbon: 6 },
        { id: 3, activity: 'Cycling', carbon: 0 },
    ];
});

const logSlice = createSlice({
    name: 'logs',
    initialState: {
        data: [],
        status: 'idle',
        error: null,
    },
    reducers: {},
    extraReducers: (builder) => {
        builder
            .addCase(fetchLogs.pending, (state) => {
                state.status = 'loading';
            })
            .addCase(fetchLogs.fulfilled, (state, action) => {
                state.status = 'succeeded';
                state.data = action.payload;
            })
            .addCase(fetchLogs.rejected, (state, action) => {
                state.status = 'failed';
                state.error = action.error.message;
            });
    },
});
```

AuthContext.jsx :



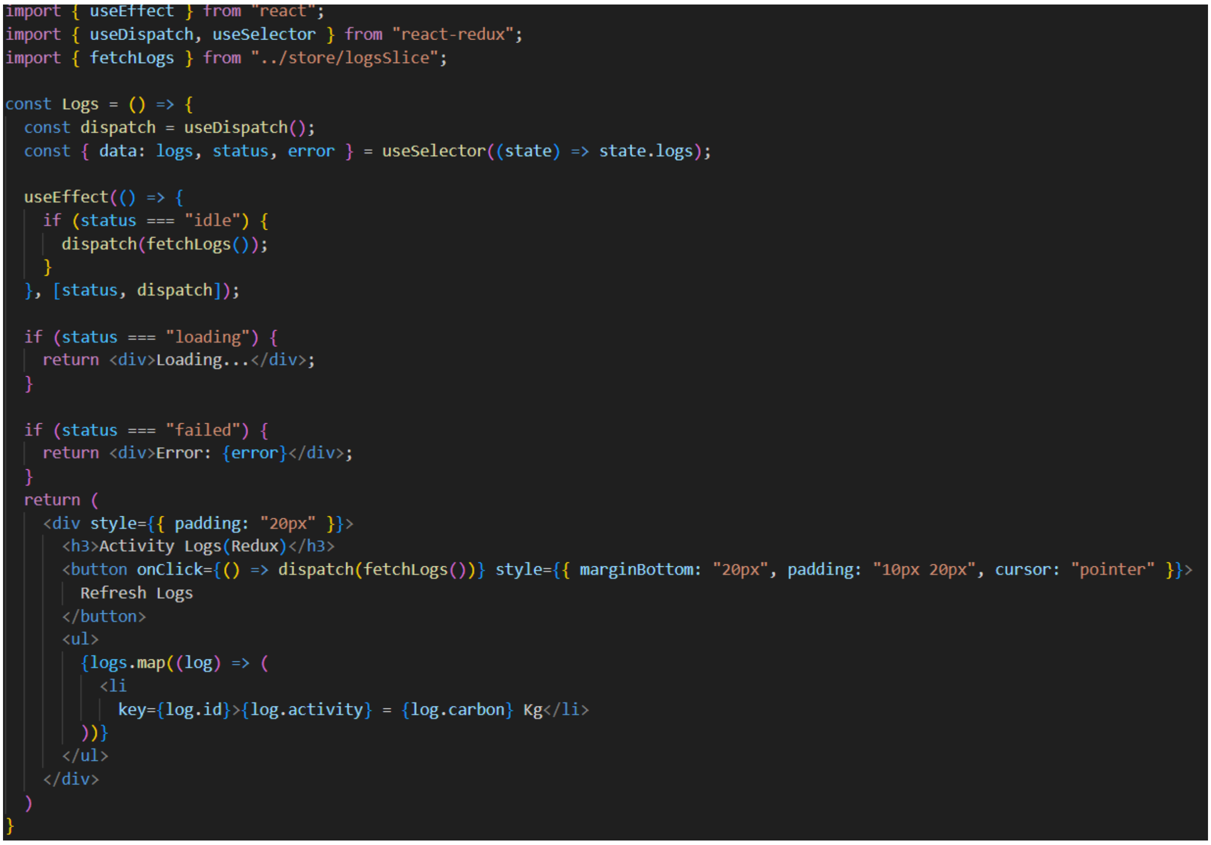
```
import { createContext,useContext,useState} from "react";

const AuthContext = createContext(null);

export const AuthProvider = ({children}) => {
    const [isAuthenticated,setIsAuthenticated] = useState(false);
    return (
        <AuthContext.Provider value = {{isAuthenticated, setIsAuthenticated}}>
            {children}
        </AuthContext.Provider>
    )
}

export const useAuth = () => useContext(AuthContext);
```

ProtectedRoute.jsx :



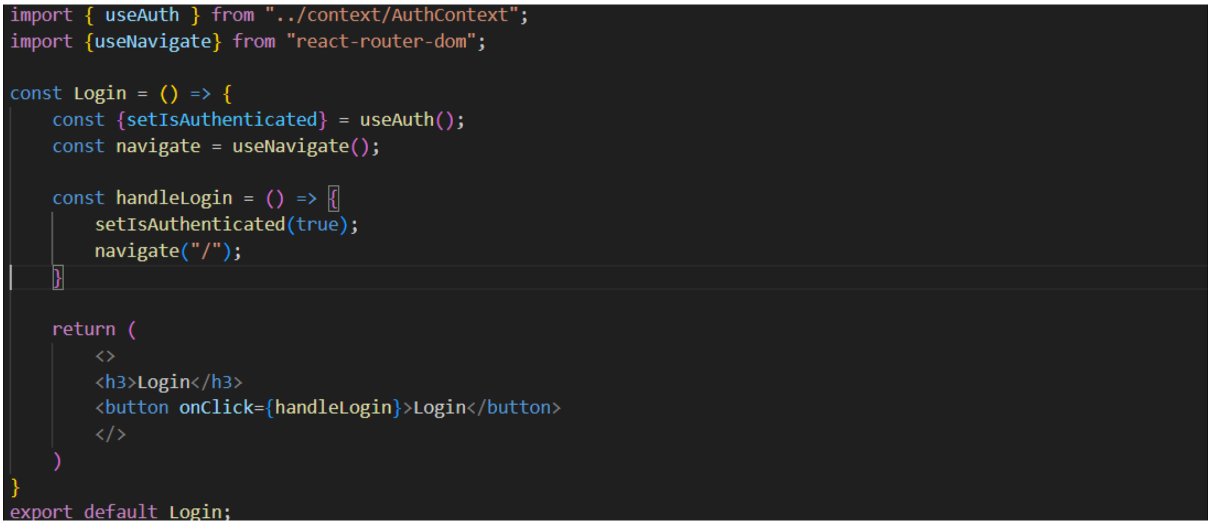
```
import { Navigate } from "react-router-dom";
import { useAuth } from "../context/AuthContext";

const ProtectedRoute = ({children}) => {
    const auth = useAuth();

    if (!auth.isAuthenticated){
        return <Navigate to="/login" replace/>
    }
    return children;
}

export default ProtectedRoute;
```

Logs.jsx :



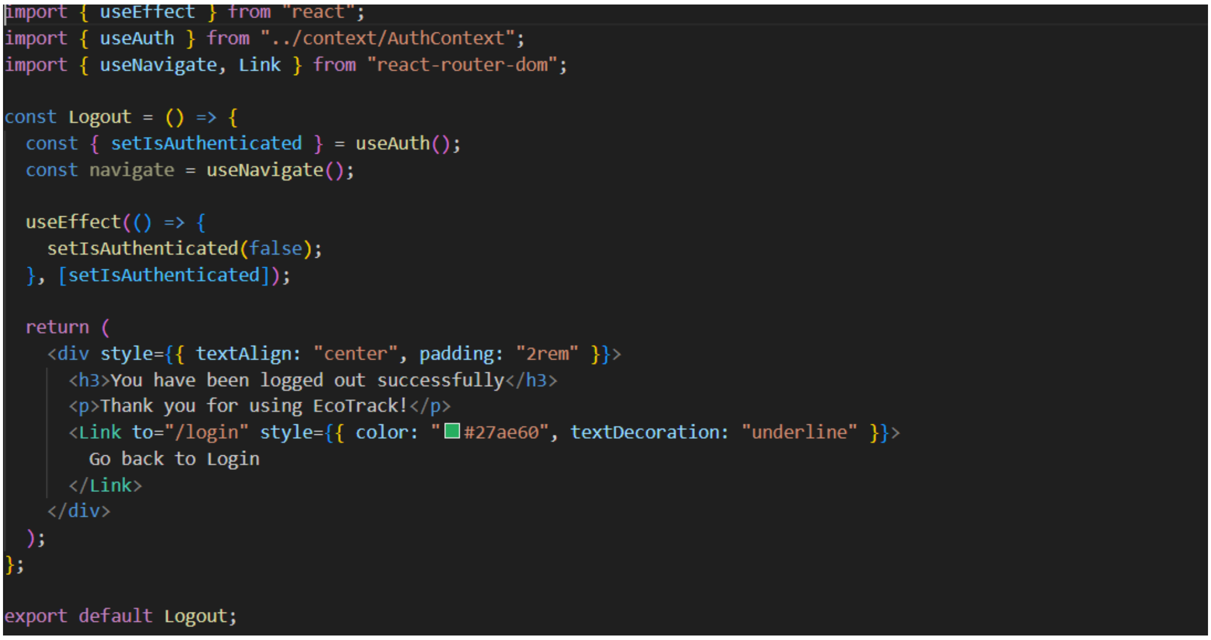
```
import { useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import { fetchLogs } from "../store/logsSlice";

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

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

  if (status === "loading") {
    return <div>Loading...</div>;
  }

  if (status === "failed") {
    return <div>Error: {error}</div>;
  }
  return (
    <div style={{ padding: "20px" }}>
      <h3>Activity Logs(Redux)</h3>
      <button onClick={() => dispatch(fetchLogs())} style={{ marginBottom: "20px", padding: "10px 20px", cursor: "pointer" }}>
        Refresh Logs
      </button>
      <ul>
        {logs.map((log) => (
          <li
            key={log.id}>{log.activity} = {log.carbon} Kg</li>
        ))}
      </ul>
    </div>
  )
}
```

Login.jsx :



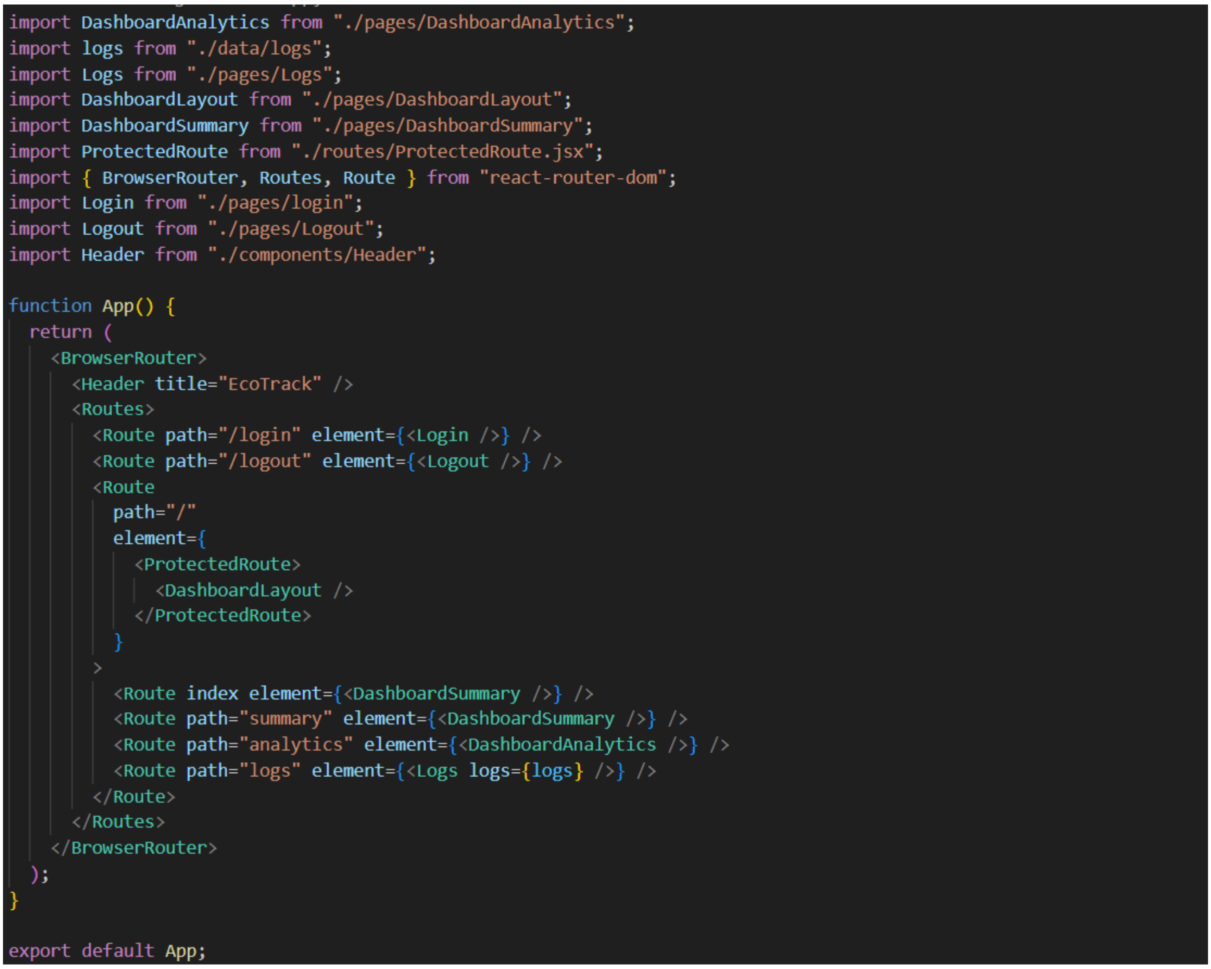
```
import { useAuth } from "../context/AuthContext";
import {useNavigate} from "react-router-dom";

const Login = () => {
    const {setIsAuthenticated} = useAuth();
    const navigate = useNavigate();

    const handleLogin = () => {
        setIsAuthenticated(true);
        navigate("/");
    }

    return (
        <>
        <h3>Login</h3>
        <button onClick={handleLogin}>Login</button>
        </>
    )
}
export default Login;
```

Logout.jsx :



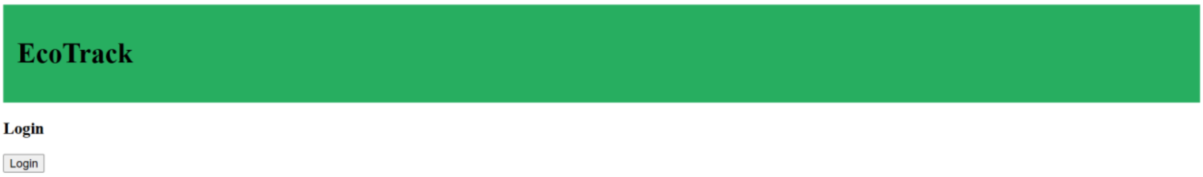
```
import { useEffect } from "react";
import { useAuth } from "../context/AuthContext";
import { useNavigate, Link } from "react-router-dom";

const Logout = () => {
  const { setIsAuthenticated } = useAuth();
  const navigate = useNavigate();

  useEffect(() => {
    setIsAuthenticated(false);
  }, [setIsAuthenticated]);

  return (
    <div style={{ textAlign: "center", padding: "2rem" }}>
      <h3>You have been logged out successfully</h3>
      <p>Thank you for using EcoTrack!</p>
      <Link to="/login" style={{ color: "🟩#27ae60", textDecoration: "underline" }}>
        Go back to Login
      </Link>
    </div>
  );
};

export default Logout;
```

App.jsx :



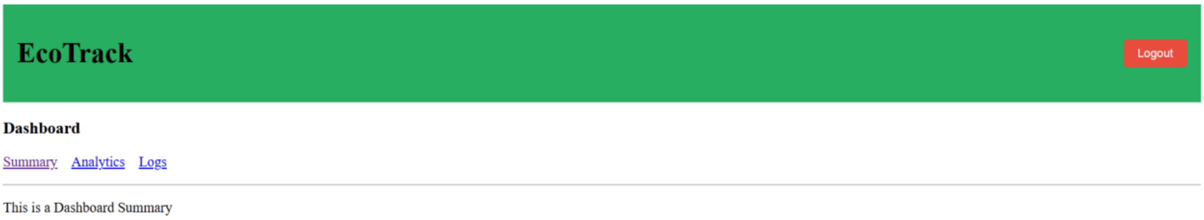
```
import DashboardAnalytics from "./pages/DashboardAnalytics";
import logs from "./data/logs";
import Logs from "./pages/Logs";
import DashboardLayout from "./pages/DashboardLayout";
import DashboardSummary from "./pages/DashboardSummary";
import ProtectedRoute from "./routes/ProtectedRoute.jsx";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Login from "./pages/login";
import Logout from "./pages/Logout";
import Header from "./components/Header";

function App() {
  return (
    <BrowserRouter>
      <Header title="EcoTrack" />
      <Routes>
        <Route path="/login" element={<Login />} />
        <Route path="/logout" element={<Logout />} />
        <Route
          path="/"
          element={
            <ProtectedRoute>
              <DashboardLayout />
            </ProtectedRoute>
          }
        >
          <Route index element={<DashboardSummary />} />
          <Route path="summary" element={<DashboardSummary />} />
          <Route path="analytics" element={<DashboardAnalytics />} />
          <Route path="logs" element={<Logs logs={logs} />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

4. Output :



EcoTrack

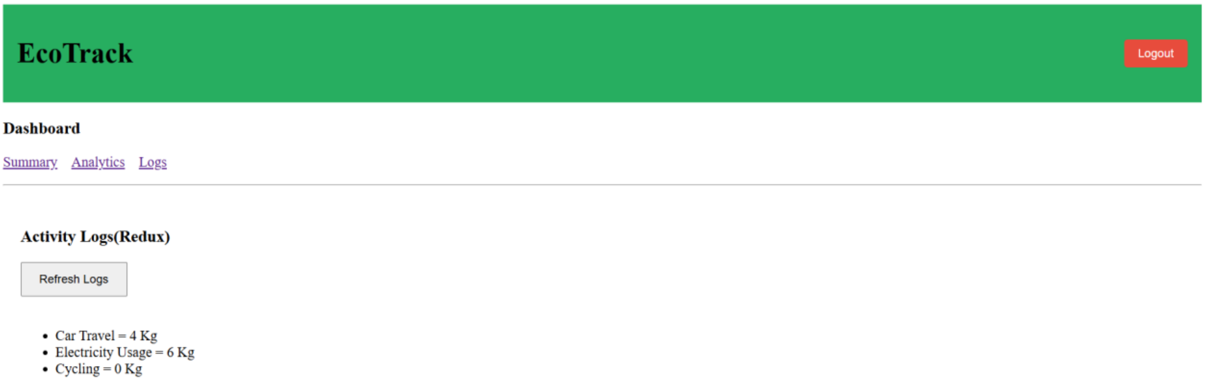
Login



EcoTrack

Dashboard

This is a Dashboard Summary



Activity Logs(Redux)

- Car Travel = 4 Kg
- Electricity Usage = 6 Kg
- Cycling = 0 Kg

5. Learning Outcomes :

- 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 andSlices.
- We learnt about Authentication and index pages
- We learnt the use of useContext and useState.
- We learnt about the flow of a React project.