



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 3

Student Name: Manasvi Sharma

Branch: BE CSE

Semester: 6th

Subject Name: Full Stack Development-II

UID: 23BCS11815

Section/Group: 23BCSKRG_3A

Date of Performance: 27/01/26

Subject Code: 23CSH-309

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:

- To implement centralized state management in EcoTrack using Redux Toolkit.
- To manage asynchronous API calls using Redux async thunks.
- To handle loading, success, and error states effectively in the application.
- To structure Redux slices for scalable frontend architecture.
- To improve data flow and component communication through a global store.

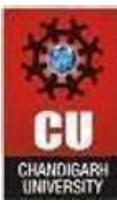
3. Implementation/Code:

logsSlice.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 },
    ];
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
const logsSlice = 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;  
            });  
    },  
});  
  
export default logsSlice.reducer;
```

store.js:

```
import { configureStore } from "@reduxjs/toolkit";  
import logsReducer from "./logSlice.js";  
  
const store = configureStore({  
    reducer: {  
        logs: logsReducer,  
    },  
});  
  
export default store;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Main.jsx:

```
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";
import "./index.css";
import App from "./App.jsx";

import { Provider } from "react-redux";
import store from "./store/store.js";

import { AuthProvider } from "./context/AuthContext.jsx";

createRoot(document.getElementById("root")).render(
<StrictMode>
  <Provider store={store}>
    <AuthProvider>
      <App />
    </AuthProvider>
  </Provider>
</StrictMode>
);
```

4. Output:

The screenshot shows a dark-themed dashboard interface. At the top, there is a navigation bar with links for Home, Overview, Reports, and Logout. Below this, the word "Dashboard" is displayed in large green text. A horizontal menu bar includes "summary | analytics".

A section titled "Total Activities" lists:

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

Below this, under "High Carbon (> 4 Kg)", it lists:

- Electricity Usage

Under "Low Carbon (<= 4 Kg)", it lists:

- Car Travel
- Cycling

At the bottom right, there is a button labeled "Refresh Logs".



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

The screenshot shows the 'Dashboard Summary' section of the EcoTrack application. At the top, there's a navigation bar with links for 'Home', 'Overview', 'Reports', and 'Logout'. Below the navigation, the title 'Dashboard' is displayed in green. A horizontal line separates this from the main content area. In the main area, there are three large rectangular boxes: the first box is labeled 'Total Carbon Footprint' and shows '320 kg CO₂', the second box is labeled 'Energy Usage' and shows '100 kWh', and the third box is partially visible at the bottom. The background of the main content area is dark gray.

Dashboard

Home Overview Reports Logout

Dashboard

summary | analytics

Dashboard Summary

Welcome to EcoTrack Dashboard Summary Page ✓

Total Carbon Footprint
320 kg CO₂

Energy Usage
100 kWh

5. Learning Outcome:

- Students will be able to configure Redux Toolkit store and slices.
- Students will understand how to use async thunks for API integration.
- Students will manage loading and error states in real-world React apps.
- Students will design clean and maintainable centralized state logic.
- Students will build production-ready data handling workflows in EcoTrack.