Job Simulation: Live Weather Dashboard

Project Title: Implementation of a Live Weather Dashboard

Role: Front-End Development Intern

Technology Stack: React, Material UI (MUI), useState, useEffect

Objective

To design and implement a responsive, real-time weather dashboard using React and Material UI. The app will fetch and display live weather information for selected cities from a public API. This project introduces interns to real-world data fetching, conditional rendering, and state handling using React hooks, all within a dashboard-style UI.

Task Overview

As part of your hands-on learning, you are required to build a single-page weather dashboard that allows users to:

- Select a city from a dropdown
- Automatically fetch and display the current weather data for the selected city
- Dynamically style the card based on the weather conditions (e.g., hot, cold, clear, cloudy)

- Display a loading indicator while data is being fetched

Task Requirements

1. Functionality:

- Use useState to manage selected city and fetched weather data
- Use useEffect to call a weather API (e.g., Open-Meteo) when the selected city changes
- Provide a dropdown with 3–5 predefined cities
- Display temperature, city name, weather condition, and last updated time
- Show a "Loading..." spinner while fetching data

2. User Interface (UI):

- Use only Material UI components:

  <Select>, <MenuItem>, <Card>, <Typography>, <Grid>, <CircularProgress>

- Design should include:
  - Dropdown for city selection

- ○ Card displaying weather info with temperature and condition
- ○ Color-coded background:
  - ■ Red for hot
  - ■ Blue for cold
  - ■ Grey for cloudy

- Use sx prop and MUI's grid system for layout and responsiveness

3. Code Structure:

- App.jsx – Main app layout, handles city selection
- WeatherCard.jsx – Renders the weather data in a styled card
- api.js (optional) – Utility file to handle API logic
- Use useEffect to handle side effects (data fetch on city change)
- Add code comments explaining how weather data is fetched and rendered

Bonus (Optional):

- Include live clock or "Last updated at" timestamp

- Add transition effect when weather card changes
- Display relevant weather icons using @mui/icons-material (e.g., WbSunny, Thunderstorm)

Deliverables:

Submit a project folder containing:

- App.jsx – Application structure
- WeatherCard.jsx – UI component displaying weather data
- api.js – (optional) helper for API logic
- **No custom HTML or CSS used — strictly MUI components**

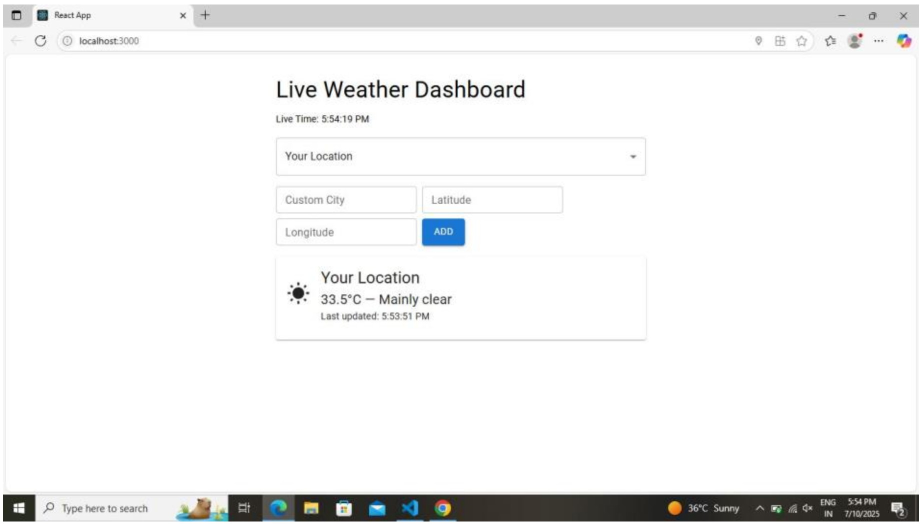
Learning Outcomes:

By completing this task, you will gain practical experience in:

- Managing and updating state using useState
- Performing side effects and data fetching with useEffect
- Building responsive UI layouts with Material UI Grid

- Using conditional rendering and dynamic styling based on real-time data

- Organizing React components for maintainable project structure



Live Weather Dashboard

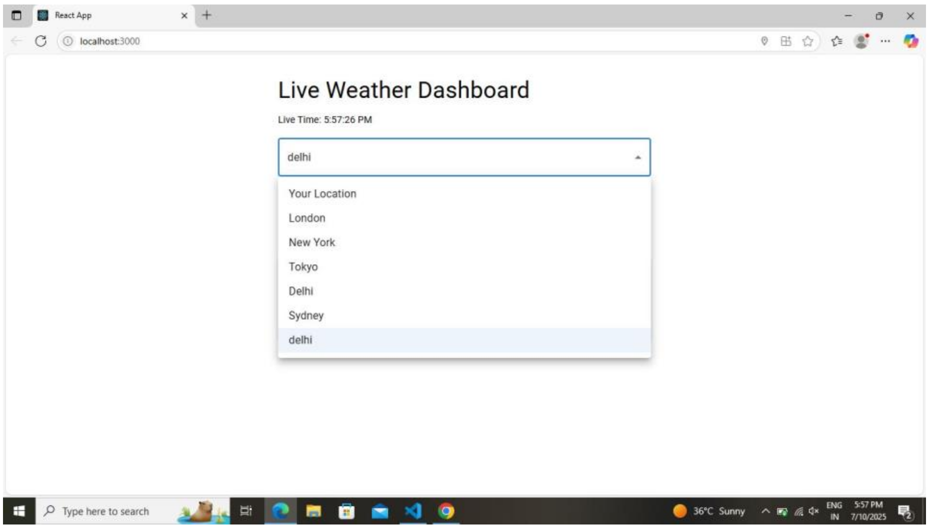
Live Time: 5:54:19 PM

Your Location

Your Location

33.5°C — Mainly clear

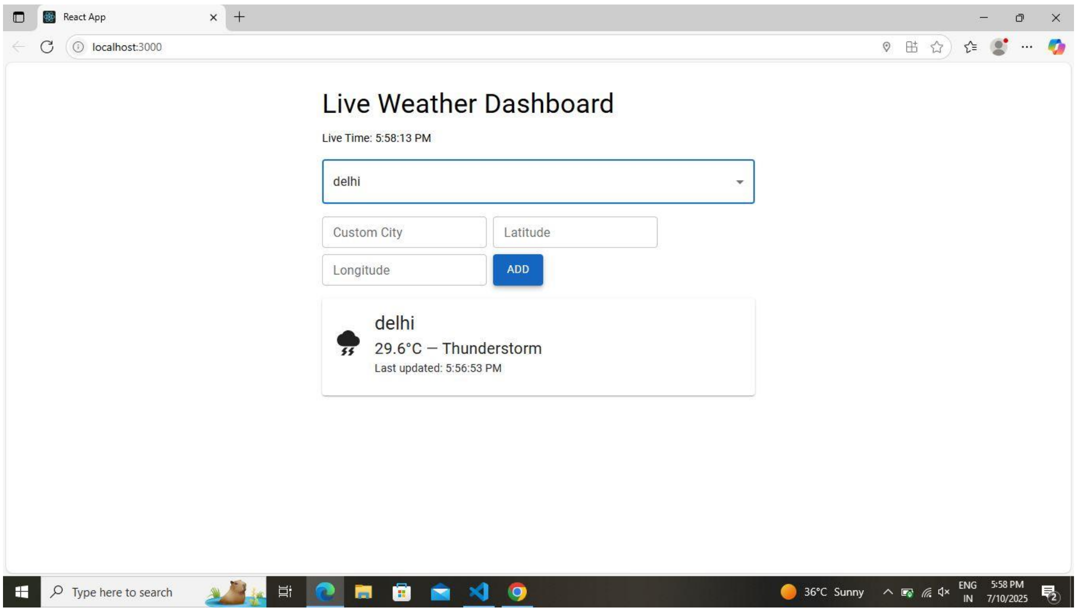
Last updated: 5:53:51 PM



Live Weather Dashboard

Live Time: 5:57:26 PM

delhi



Live Weather Dashboard

Live Time: 5:58:13 PM

delhi

Custom City | Latitude

Longitude | ADD

delhi

29.6°C — Thunderstorm

Last updated: 5:56:53 PM