9530

St. MOTHER THERESA ENGINEERING COLLEGE

COMPUTER SCIENCE ENGINEERING

NM-ID: 3441E8AD7EF6DF695034F9CBA83-8503

REG NO: 953023104138

DATE:22-09-2025

Completed the project named as

Phase 2

FRONT END TECHNOLOGY

Live Wheather Dashboard

SUBMITTED BY:

P.YARWIN YAHAV

9361906857

Phase 2 – Solution Design & Architecture

1. Tech Stack Selection

The project requires lightweight, responsive, and API-driven technologies. The chosen stack: **Frontend:** React.js (or HTML, CSS, JavaScript if simpler UI is preferred) **Backend:** Node.js + Express (for handling API requests and possible caching) **API Provider:** OpenWeatherMap API (for weather & forecast data) **Database (Optional):** MongoDB or Firebase (for storing search history or user preferences) **Hosting:** GitHub Pages / Vercel (Frontend), Heroku / Render (Backend)

2. UI Structure / API Schema Design

UI Structure (Dashboard Page):
Header: Application title + search bar (city input) **Main Section:** Current Weather Card: Displays temperature, humidity, wind speed, weather condition (icon). Forecast Section: Shows weather for the next 5 days (date, temp, condition). **Footer:** Credits (e.g., “Powered by OpenWeatherMap”). **API Schema Design (Sample Response – Current Weather):** { "coord": {"lon": 80.2785, "lat": 13.0878}, "weather": [{"main": "Clouds", "description": "overcast clouds"}], "main": {"temp": 303.15, "humidity": 74, "pressure": 1005}, "wind": {"speed": 4.63}, "name": "Chennai" }

3. Data Handling Approach

- User enters city → Request sent to backend (Node.js).
- Backend calls OpenWeatherMap API.
- API returns JSON → Backend parses & forwards response to frontend.
- Frontend updates state (React hooks) and renders weather cards.
- Caching Strategy: Recently searched cities can be stored in localStorage or database.
- Error Handling: Show friendly error message if API fails or city not found.

4. Component / Module Diagram

Frontend Components: SearchBar → Input for city names. WeatherCard → Displays current weather data. ForecastList → Shows 5-day forecast cards. Dashboard → Parent component that integrates everything. **Backend Modules:** API Handler → Connects to OpenWeatherMap API. Response Formatter → Cleans and structures data for frontend. Error Handler → Manages invalid city names or failed API calls.

5. Basic Flow Diagram

- User enters city in search bar.
- Frontend sends request to Backend.
- Backend queries OpenWeatherMap API.
- API response → Extract required fields.
- Backend sends clean JSON to frontend.
- Frontend displays current weather + forecast.

6. Example Flow (Use Case: Chennai)

User types “Chennai” → Press search. Request sent: GET /weather?q=Chennai Backend fetches data from OpenWeatherMap API. API returns JSON → Extracts {temp=30°C, humidity=74%, wind=4.6 km/h, condition=Clouds}. Dashboard updates: Current weather card shows 30°C, Humidity 74%, Clouds. Forecast section shows next 5 days summary.