COLLEGE CODE : 9111

COLLEGE NAME: SRM MADURAI COLLEGE FOR ENGINEERING AND TECHNOLOGY

DEPARTMENT: B.E COMPUTER SCIENCE AND ENGINEERING

STUDENT NM-ID:

3DE3OEEADOADD16ABF843D795F7463AA
DE16A18153AEB4296O6F4DC6FC9A42259
31D7A43E9953565998B4B4D75D69E588
71OF5CABO746CA2B61O4D9A9EF3C4CBC

ROLLNO:911123104007
911123104027
911123104037
911123104038

DATE:

Completedtheproject named as Phase 2

TECHNOLOGYPROJECT NAME : L i v e W e a t h e r D a s h b o a r d

SUBMITTED BY,NAME:

D.Devis Akalya Pushpam
T.D.B.Kiruthikha
R.S.Priyadharshini
S.B.Priyadharshini

Live Weather Dashboard - Phase 2

Problem Statement

Weather conditions change rapidly, and people rely on accurate, real-time updates for planning travel, outdoor activities, and safety precautions. A Live Weather Dashboard is required to fetch, process, and display real-time weather data (temperature, humidity, wind, etc.) in a user-friendly and interactive interface.

The dashboard should support:

- Real-time updates via weather APIs
- Location-based search (city, pincode, or GPS-based)
- Clear visualizations (charts, icons, maps)
- Responsive UI for web and mobile

Tech Stack Selection

Frontend (UI):

- React (or Next.js for SSR)
- Tailwind CSS / Material UI (for styling)
- Chart.js / Recharts (for weather graphs & trends)

Backend:

- Node.js with Express (API gateway)
- Axios/Fetch for API integration
- WebSockets for real-time updates (optional)

Database (optional for history):

MongoDB / Firebase (to store search history, past weather data)

Weather API: OpenWeatherMap API / WeatherAPI

Hosting: Vercel/Netlify (Frontend), Render/Heroku (Backend), MongoDB Atlas / Firebase (DB)

UI Structure

- Header / Navbar – App title, location search bar, current location button
- Current Weather Card – Temperature, Condition (Sunny, Cloudy, Rainy), Icon, Date & Time

- Weather Details Section – Humidity, Wind Speed, Visibility, Pressure
- Forecast Section (Hourly/Daily) – Scrollable cards with forecast data
- Charts Section – Temperature vs Time line chart, Rain probability bar chart
- Footer – API credit, About link

API Schema Design

Endpoint: /api/weather

Request Example:

{ "location": "Chennai", "units": "metric" }

Response Example:

{ "location": "Chennai", "temperature": 30, "condition": "Cloudy", "humidity": 75, "wind_speed": 12, "forecast": [{ "time": "10:00", "temp": 29, "condition": "Cloudy" }, { "time": "11:00", "temp": 30, "condition": "Sunny" }] }

Data Handling Approach

- Fetch API – Get JSON data from OpenWeather API
- Transform – Convert raw data into UI-friendly format (°C/°F, icons)
- State Management – React Context / Redux for storing current & forecast data
- Cache – LocalStorage for recent searches
- Error Handling – API failure fallback (show last cached data)

Component / Module Diagram

[App Component] ■■■ Header (Search, Location) ■■■ CurrentWeatherCard ■■■ WeatherDetails ■■■ ForecastList ■■■ ChartsSection ■■■ Footer

Basic Flow Diagram

User → Search Location → Backend API Call → Weather API ↓ ↑ Display UI ← Process & Format Data ← Store/Cache