



COLLEGE CODE : 9238

COLLEGE NAME: Mangayarkarasi College of Engineering

DEPARTMENT: CSE

STUDENT NM-ID:
5040D5ACBEEB3EE144AD7AACD13DB424D

ROLL NO.: 923823104023

DATE: 08-09-2025

Completed the project named as Phase 1 – Problem Understanding and
Requirements

FRONT END TECHNOLOGY

PROJECT NAME: LIVE WEATHER DASHBOARD

SUBMITTED BY,

NAME: KARTHIKA DEVI S

MOBILE NO: 7904591497



LIVE WEATHER DASHBOARD

Phase 1 – Problem Understanding & Requirements

1. Problem Statement

In today's fast-paced and climate-sensitive world, access to accurate and real-time weather data is essential for individuals, businesses, and governments. However, most existing weather platforms suffer from one or more of the following issues:

Lack of real-time updates or delays in data refresh

Overly complex interfaces that hinder usability

Limited localization and customization options

Fragmented data sources and inconsistent formats

These limitations pose challenges for users who rely on weather insights for decision-making—whether it's planning travel, managing agricultural operations, or preparing for extreme weather events.

The goal of the **Live Weather Dashboard** is to address these gaps by creating a centralized, intuitive, and responsive platform that delivers real-time weather data tailored to user preferences. The dashboard will serve as a reliable tool for monitoring current conditions, forecasting trends, and visualizing weather metrics in a clean and actionable format.

2. Users & Stakeholders

Primary Users

General Public: Individuals seeking daily weather updates for personal planning.

Travelers: Users needing weather forecasts for multiple destinations.

Agricultural Professionals: Farmers and agronomists monitoring weather for crop planning.

Event Planners: Organizers needing accurate forecasts for outdoor events.

Logistics & Transportation Managers: Professionals optimizing routes based on weather conditions.

Stakeholders

Product Owner: Oversees project vision, prioritization, and delivery.

Development Team: Engineers and designers responsible for building the dashboard.

Data Providers: External weather API services (e.g., OpenWeatherMap, Weather.net).

Business Analysts: Ensure alignment with user needs and business goals.

QA/Testers: Validate functionality, performance, and usability.

3. User Stories

User stories help define the product from the perspective of its users.

Here are key examples:

As a traveler, I want to view the weather forecast for multiple cities so that I can plan my itinerary.

As a farmer, I want to monitor rainfall and temperature trends so that I can schedule irrigation and harvesting.

As an event planner, I want to receive alerts about extreme weather so that I can make contingency plans.

As a general user, I want to see the current weather in my location so that I can dress appropriately.

As a data analyst, I want to download historical weather data so that I can perform trend analysis.

Each story will be mapped to specific features in the MVP and future releases.

4. MVP Features

The Minimum Viable Product (MVP) will focus on delivering core functionality that solves the primary user needs:

Feature	Description
Real-Time Weather Data	Display current temperature, humidity, wind speed, and conditions

Feature	Description
Location-Based Forecasts	Auto-detect user location or allow manual city selection
5-Day Forecast View	Show upcoming weather trends with icons and metrics
Weather Alerts	Notify users of severe weather conditions (e.g., storms, heatwaves)
Interactive Charts	Visualize temperature and precipitation trends
Responsive UI	Mobile-friendly and accessible design
Auto-Refresh Option	Periodic data updates without manual refresh
Search History	Save and revisit previously searched cities

5. Wireframes / API Endpoint List

Wireframes (Conceptual Overview)

Home Dashboard: Displays current weather, location, and forecast summary.

City Search Page: Allows users to input and save multiple cities.

Forecast Detail View: Interactive charts showing temperature, humidity, and wind trends.

Alert Panel: Highlights warnings and advisories.

(Note: Wireframes to be created using tools like Figma, Adobe XD, or Balsamiq.)

API Endpoint List

Endpoint	Method	Description
/weather/current?city=XYZ	GET	Fetch current weather for specified city
/weather/forecast?city=XYZ	GET	Retrieve 5-day forecast data
/weather/alerts?city=XYZ	GET	Get active weather alerts
/weather/history?city=XYZ	GET	Access historical weather data
/location/detect	GET	Auto-detect user location via IP or GPS

APIs will be integrated using RESTful architecture, with JSON responses and error handling.

6. Acceptance Criteria

To ensure the dashboard meets expectations, the following criteria will be used:

Functional Criteria

The dashboard must display real-time weather data for at least one city.

Users must be able to search and view weather for multiple cities.

Forecast data must include temperature, humidity, wind speed, and conditions.

Alerts must be shown when severe weather is detected.

Usability Criteria

The UI must be responsive across desktop and mobile devices.

Users must be able to navigate between views with minimal clicks.

Data must refresh automatically or via a manual refresh button.

Performance Criteria

API calls must respond within 2 seconds under normal load.

The dashboard must support at least 100 concurrent users without degradation.

Security Criteria

All API requests must be secured via HTTPS.

User data (if any) must be handled in compliance with privacy standards.