COLLEGE CODE : 9623

COLLEGE NAME : Amrita College of Engineering And Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID :FOF3EBA66EADA4782A846E244BC964C3

ROLL NO : 23CS013

DATE : 12-09-2025

Completed the project named as

Phase 1 problem understanding & requirements

PROJECT NAME : Live Weather Dashboard

SUBMITTED BY,

NAME : AJAY C

MOBILE NO :95973 54838

Phase 1 – Problem Understanding & Requirements (Live Weather Dashboard).

Problem Statement

People need quick, accurate, and real-time weather information to plan their daily activities. Current weather apps often have too many features, ads, or cluttered UI. A Live Weather Dashboard should provide simple, reliable, and real-time weather updates with a focus on usability.

Users & Stakeholders

Primary Users

Commuters who need daily weather updates before leaving home.

Outdoor workers (delivery personnel, construction workers, etc.) who rely on weather conditions.

Travelers/tourists who need forecasts for new locations.

General users who want a clean, minimal dashboard for quick checks.

Stakeholders

Product Owner / Client (defines vision).

Development Team (frontend, backend, QA).

Weather Data Provider (e.g., OpenWeatherMap API).

UX/UI Designer.

User Stories

1. As a user, I want to search weather by city name, so that I can quickly get conditions anywhere.

2. As a user, I want to see current temperature, humidity, and wind speed, so that I know the conditions right now.

3. As a user, I want to see a 5-day forecast, so I can plan my week.

4. As a user, I want to view weather in my current location automatically, so I don’t have to type it.

5. As a user, I want a clean, ad-free interface, so I can focus only on weather updates.

MVP Features

 Location-based weather (auto-detect + manual search).

 Current weather data (temperature, humidity, wind speed, weather condition).

 5-day forecast (temperature + condition).

Real-time updates via weather API.

 Responsive UI (mobile + desktop).

Wireframes / API Endpoint List

Wireframes (Basic Layout)

1. Header – App name + search bar.

2. Current Weather Card – City, temp, condition icon, humidity, wind.

3. Forecast Section – Horizontal scroll cards (Day + temp + icon).

4. Footer – Data source credit (e.g., “Powered by OpenWeather”).

(You can sketch or use Figma later)

API Endpoint List (Example: OpenWeatherMap API)

GET /weather?q={city name}&appid={API key} → Current weather.

GET /weather?lat={lat}&lon={lon}&appid={API key} → Current weather by coordinates.

GET /forecast?q={city name}&appid={API key} → 5-day forecast.

Acceptance Criteria

✅User can search weather by city and see results within 3 seconds.

✅Dashboard auto-loads weather for current location if permission granted.

✅Current weather card shows: temperature (°C/°F), condition (icon + text), humidity, wind speed.

✅Forecast shows at least 5 upcoming days with min/max temp + condition.

✓Mobile and desktop layouts are responsive.

✓No ads, simple UI with focus on weather data only.

UI/UX

✓Dashboard works seamlessly on desktop, tablet, and mobile (responsive design).

✓Weather icons are consistent, intuitive, and match the condition (e.g., sun, rain, cloud).

✓A default city (e.g., "London" or "New York") is shown if no location or input is available.

✓Error messages are shown for invalid city names or failed API calls (e.g., “City not found” / “Unable to fetch weather”).

✓Clear indication of last updated time (e.g., "Updated 2 mins ago").

Performance

✓API response errors (like 404, 500) are gracefully handled with fallback UI.

✓Search results must appear within 2–3 seconds for good UX.

✓The app must load in under 5 seconds on a 4G connection.

Data Accuracy

✓Temperature is shown in both Celsius and Fahrenheit (toggle option).

✓Wind speed is shown in km/h or mph depending on region/user setting.

✓Data is always consistent with the weather API provider’s latest update.

Security & Reliability

✓API key is stored securely and not exposed in client-side code.

✓Dashboard continues to show cached last known weather if offline.

✓App gracefully handles cases where the weather API rate limit is exceeded.