TECHNICAL DESIGN DOCUMENT (TDD)

Project: React Weather Dashboard (Frontend API-Based Application)

Author: Rasagna, Sanjana, Hari Krishnan

Date: 2025

1. Introduction

1.1 Objectives

The purpose of this project is to build a client-side React weather dashboard that fetches and displays real-time weather information and a 3-day forecast using a public third-party API (OpenWeatherMap). The application demonstrates React components, state management, secure API handling, and responsive UI.

1.2 Background

This application is part of a frontend React internship project where interns create a functional UI that communicates directly with an external API. No backend or database is involved.

1.3 Assumptions

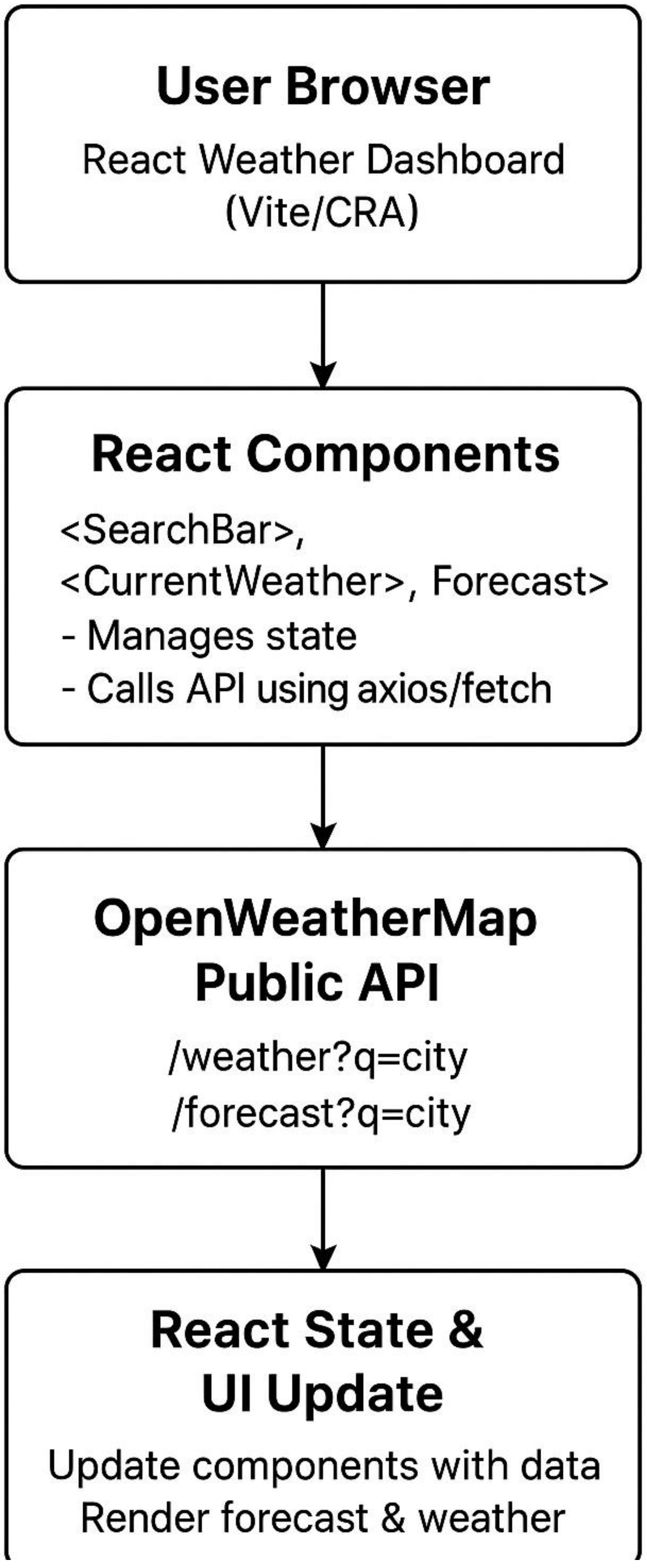
- User has internet connection.

- API is accessible.

- API key stored in .env.

- User may allow location access.

2. Solution Architecture


User Browser
React Weather Dashboard (Vite/CRA)
React Components
<SearchBar>,
<CurrentWeather>, Forecast>
- Manages state
- Calls API using axios/fetch
OpenWeatherMap Public API
/weather?q=city
/forecast?q=city
React State & UI Update
Update components with data
Render forecast & weather

3. Database Schema

Not Applicable (No backend or database).

4. API Endpoint Specification

4.1 OpenWeatherMap Current Weather API

GET
https://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}&units=metric

4.2 OpenWeatherMap Forecast API

GET
https://api.openweathermap.org/data/2.5/forecast?q={city}&appid={API_KEY}&units=metric

5. Frontend Design

Components:

- App

- SearchBar

- CurrentWeather

- Forecast

- UnitToggle

- ErrorMessage

- LoadingSpinner

State Variables:

city, weatherData, forecast, units, isLoading, error

6. Environment Variables

Stored in .env:

VITE_APP_API_KEY=your_api_key

7. Testing Strategy

Unit tests, integration tests, manual tests for responsiveness, invalid city, slow network.

8. Deployment

Deploy to Vercel/Netlify, add environment variables, test production.

9. Risks & Mitigations

- API rate limit → retry

- Network issues → loading + retry

- Key leak → secure .env

10. Conclusion

This TDD defines how the React Weather Dashboard will be implemented, covering architecture, components, API usage, state, and deployment.