College Code : 9623

College Name : Amrita College of Engineering and Technology

Department : Artificial Intelligence and Data Science

Student NM-ID : autad0001

Roll number : 962323243001

Completed the project named as : Phase 2 : Live Weather Dashboard

Submitted by : Aarthibala .R

Date : 22-09-2025

Mobile Number : 7598480329

Phase 2 — Solution Design & Architecture

1. ## Tech Stack Selection

   ### Frontend:
   - React.js (component-based UI)
   - Tailwind CSS (styling)
   - Axios / Fetch API (API calls)
   - Chart.js / Recharts (graphs)

   ### Backend:
   - Node.js + Express.js
   - OpenWeatherMap API
   - dotenv (API key management)

   ### Database (Optional):
   - MongoDB / SQLite (favorites, history)

   ### Other Tools:
   - Postman (API testing)
   - GitHub (version control)
   - Vercel/Netlify (Frontend) + Render/Heroku (Backend) (deployment)

2. ## UI Structure / API Schema Design

   ### UI Structure:
   - Homepage (search bar, current weather card, icon)
   - Forecast Section (hourly & 5-day forecast)
   - Favorites & Theme toggle

API Schema:

- GET /api/weather/:city $\rightarrow$ Current weather

- GET /api/weather/:city/forecast $\rightarrow$ 5-day forecast

- POST /api/favorites $\rightarrow$ Save favorite city

- GET /api/favorites $\rightarrow$ Get saved favorites

- GET /api/history $\rightarrow$ Recent searches

Data Handling Approach

Frontend:
- React useState & useEffect
- Axios for API calls
- Conditional rendering

Backend:
- Fetch data from OpenWeather API
- Simplify response
- Optional caching

Error Handling:

- Invalid city $\rightarrow$ 404 City not found
- API issues $\rightarrow$ Fallback messages

3. Component / Module Diagram

Frontend Components:

- App.js (main)
- SearchBar.js
- WeatherCard.js
- ForecastCard.js
- FavoritesList.js
- ThemeToggle.js

Backend Modules:

- server.js
- routes/weather.js
- controllers/weatherController.js
- models/favorite.js

4. Basic Flow Diagram

```
[User] → (enters city name)
↓
[Frontend: React + Axios]
↓ (API call)
[Backend: Node.js + Express]
↓ (Fetch data)
[OpenWeather API]
↓ (returns JSON)
[Backend formats response]
↓
[Frontend displays dashboard]
```