

Sprint 4: Backend Integration and Final Polish

1. System Overview

The WeatherSensor AI aims to provide users with real-time weather information enhanced by natural-language summaries generated through a large language model.

2. Functional Requirements

User Story: As a user, I want the application to retrieve real-time weather data, handle errors smoothly, and generate a clear, natural-language summary so I can quickly understand current weather conditions without reading raw data.

4.1 - Create and Implement Backend Chat Handler –

- FR 4.1.1: Build a backend endpoint to receive chat messages and forward them to Gemini.
- FR 4.1.2: Return Gemini responses to the front-end chat in real time.
- FR 4.1.3: Log all chat exchanges for debugging and testing.

Progress Summary:

4.2 - Add Conversational Limits and Constraints –

- FR 4.2.1: Restrict chat replies to weather-related questions only.
- FR 4.2.2: Add checks to ignore unrelated or unsafe inputs.
- FR 4.2.3: Display a short “out-of-scope” message when a user asks a non-weather topic.

Progress Summary:

4.3 - Implement Error Handling for Chat –

- FR 4.3.1: Detect failed Gemini API calls and show a user-friendly error message.
- FR 4.3.2: Retry failed requests once before stopping.
- FR 4.3.3: Log connection or response errors for review.

Progress Summary:

4.4 - Verify Topic and Conversational Flow with Test Cases –

- FR 4.4.1: Create test conversations for valid and invalid weather topics.
- FR 4.4.2: Verify Gemini replies stay within the weather domain.
- FR 4.4.3: Check that error messages appear correctly when issues occur.

Progress Summary:

4.5 - Polish UI (Ensure Website is Professional) –

- FR 4.5.1: Align fonts, colors, and spacing across all pages.
- FR 4.5.2: Improve chat and weather panel layout for readability.
- FR 4.5.3: Replace placeholder text and ensure consistent design tone.
- FR 4.5.4: Ensure spelling and grammar correctness.

Progress Summary:

4.6 - Final Responsiveness and Polish –

- FR 4.6.1: Test responsiveness on desktop, tablet, and mobile.
- FR 4.6.2: Optimize loading animations and transitions.
- FR 4.6.3: Ensure all links, buttons, and routes work before final deployment.

Progress Summary:

4.7 - Update README and Documentation –

- FR 4.7.1: Update README with setup, usage, and deployment instructions.
- FR 4.7.2: Add a section summarizing all sprints and key features.
- FR 4.7.3: Include contributor credits and project structure overview.

Progress Summary:

4.8 - Create Demo Video and Retrospective –

- FR 4.8.1: Record a short demo showing weather lookup and chat summary.
- FR 4.8.2: Present key accomplishments and challenges faced during development.
- FR 4.8.3: Compile lessons learned and improvement ideas for future versions.

Progress Summary: