Sprint 3: Frontend Integration and User Interaction

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.

3.1 - Integrate Chat UI with Gemini – Manu Redd

- FR 3.1.1: Connect Gemini API output to the chat input/output window.
- FR 3.1.2: Display model responses as chat bubbles in the UI.
- FR 3.1.3: Allow the user to type questions and receive summaries in real time.

Progress Summary:

3.2 - Display Key Weather Information on Website –

- FR 3.2.1: Fetch and show current temperature, humidity, and wind speed from the weather API.
- FR 3.2.3: Ensure data updates automatically when a new location is searched.

Progress Summary:

3.3 - Add Weather Icons for Conditions (sun, rain, clouds) –

- FR 3.3.1: Display matching weather icons for conditions (e.g., sun, rain, clouds).
- FR 3.3.2: Ensure data updates automatically when a new location is searched.

Progress Summary:

3.4 - Implement Location Search – Evans Chigweshe

- FR 3.4.1: Add a search bar for city name input.
- FR 3.4.2: Add a “Use My Location” button that detects GPS coordinates.

- FR 3.4.3: Fetch and display weather data for the selected location.

Progress Summary:

3.5 - Add Unit Switching (Celsius <-> Fahrenheit) – Riley England

- FR 3.5.1: Add a toggle button to switch between C and F.
- FR 3.5.2: Convert temperature values correctly when toggled.
- FR 3.5.3: Update the UI instantly to reflect the selected unit.

Progress Summary:

3.6 - Create Loading and Error States with UI – Evans Chigweshe

- FR 3.6.1: Show a loading animation while fetching data or waiting for Gemini.

Progress Summary:

3.7 - Display Error Messages and Retry Option – Evans Chigweshe

- FR 3.7.1: Display clear and precise error messages if data fails to load or input is invalid.
- FR 3.7.2: Provide a “Try Again” button after an error.

Progress Summary:

3.8 - Verify Correctness of Location Weather\Unit Conversion with Test Cases – Riley England

- FR 3.8.1: Test search results for multiple cities and GPS locations.
- FR 3.8.2: Test “Use My Location” Button.
- FR 3.8.3: Verify temperature conversion between C and F for accuracy.
- FR 3.8.4: Check that UI updates correctly after each new query or toggle.

Progress Summary:

3.9 - Design and Integrate Chat UI Component – Manu Redd

- FR 3.9.1: Design a simple and clean chat window with user and bot bubbles.
- FR 3.9.2: Match the color scheme and style of the current website.

- FR 3.9.3: Ensure the chat section fits smoothly into the main layout and resizes for mobile devices.

Progress Summary:

3.10 - Create and Complete Sprint 3 Artifacts - Riley England

- FR 3.10.1: Compile Requirements for Sprint 3 into formatted documentation.
- FR 3.10.2: Write functional requirements for each requirement, leaving space for progress summaries.
- FR 3.10.3: Verify artifact alignment with course rubric and project objectives.
- FR 3.10.4: Upload all Sprint 3 documents to github.

Progress Summary: Compiled all Sprint 3 Requirements into the specified format of the Artifacts document. Broke each requirement down into its Functional Requirements, and gave a description for each. Finally, compiled all Sprint 3 deliverables and uploaded them to the github, ensuring the final version was pushed.