

Development Log & Testing Guide

This document summarizes the key implementation steps for each version of the application and provides instructions on how to test them in a local environment using the Firebase Emulator Suite.

v1.0.0: Initial Setup & Basic Chat

Implemented Features

- **Basic Chat Functionality:** Established a real-time chat interface where users can send messages and receive responses from the AI.
- **Default AI Persona:** Created a default personality for the AI, focusing on being a good listener and providing a safe space for introspection.
- **Memory Deletion:** Implemented a feature allowing users to delete all their stored data (chat history, etc.) from Firestore.
- **Firebase Emulator Setup:** Configured `firebase.json` to use Auth, Functions, Firestore, and Hosting emulators for local development.

Local Testing Instructions

1. Start the Firebase emulators:

```
firebase emulators:start
```

2. Open your browser and navigate to `http://localhost:5000`.
3. Sign in using the emulated Google authentication.
4. **Test Chat:** Send messages in the chat interface and verify that you receive replies from the AI.
5. **Test Memory Deletion:** Click the "Delete Memory" button. Confirm in the Firestore Emulator UI (`http://localhost:4000`) that all data under your user ID (`/users/{uid}`) has been deleted.

v2.0.0: Frontend & Backend Separation

Implemented Features

- **Code Refactoring:** Separated the project into a `static` directory for all frontend code (HTML, CSS, JavaScript) and a `functions` directory for all backend logic (Cloud Functions for Firebase). This improves maintainability and modularity (DR8, DR9, DR10).

Local Testing Instructions

1. Follow the same testing instructions as for `v1.0.0`.
2. The primary goal is to confirm that all functionalities (chat, login, memory deletion) work exactly as they did before the refactoring.

v3.0.0: Daydream - Personality Analysis

Implemented Features

- **Daydream Function:** Implemented a background Cloud Function (`create_personality_analysis`) that is triggered every 10 messages sent by the user.
- **Personality Analysis:** This function analyzes the last 50 messages using the Gemini API and a "Big Five" personality model prompt.
- **Data Storage:** The analysis result is saved as a new document in the `users/{uid}/personalityAnalyses` collection in Firestore.

Local Testing Instructions

1. Start the Firebase emulators:

```
firebase emulators:start
```

2. Open the application at `http://localhost:5000` and sign in.
3. Open the Firestore Emulator UI in another tab: `http://localhost:4000`.
4. Send exactly **10 messages** to the AI.
5. After the 10th message, check the Cloud Functions logs for output indicating that a "daydream" is being triggered.

6. In the Firestore Emulator UI, navigate to the `users/{your-uid}/personalityAnalyses` collection and verify that a new document has been created with `type: daydream` and contains the analysis results.

v4.0.0: Dream - AI Persona Evolution

Implemented Features

- **Dream Function:** Implemented a background Cloud Function (`create_dream_analysis`) that synthesizes past personality analyses to evolve the AI's core persona.
- **Trigger Condition:** This function is triggered automatically after every 5 `daydream` analyses are completed.
- **Persona Update:** The 'Dream' function calls the Gemini API to generate a new, more refined `basePersonality` for the AI, which is then updated in the `users/{uid}/aiPersona/current` document. The execution itself is also logged as a `dream` type analysis.

Local Testing Instructions

Testing this feature requires triggering 5 `daydream` events, which would normally require 50 user messages. Use the following shortcut to test more quickly.

1. Start the Firebase emulators:

```
firebase emulators:start
```

2. Open the application (`http://localhost:5000`) and sign in to create your user entry in Firestore.
3. Open the Firestore Emulator UI (`http://localhost:4000`).
4. **Manually create test data:**
 - Navigate to the `users/{your-uid}` document.
 - Create a new collection named `personalityAnalyses` .
 - Inside `personalityAnalyses` , manually add **four** new documents. Each document should have a single field: `type` with the string value `daydream` . The content of the rest of the document doesn't matter for this test.
5. **Trigger the final Daydream and Dream:**
 - Return to the application UI at `http://localhost:5000` .
 - Send **10 more messages**.

- After the 10th message, the 5th `daydream` will be triggered, which in turn will trigger the `dream` function.

6. Verify the results:

- Check the Cloud Functions logs for messages indicating both "daydream" and "dream" were triggered.
- In the Firestore Emulator, navigate to `users/{your-uid}/aiPersona/current` and verify that the `basePersonality` text has been updated to a new, more detailed persona.
- In `users/{your-uid}/personalityAnalyses`, verify that a new document with `type: dream` has been created.

Deployment Note

The attempt to deploy using the `classic_firebase_hosting_deploy` tool failed because this project uses Cloud Functions, which requires a more advanced deployment setup.

Recommendation: Use **Firebase App Hosting** for deployment. This service is designed for modern web apps with backend logic and can be configured from the Firebase Console to connect directly to your GitHub repository for automated builds and deploys.

Please refer to the official documentation for setup: [Firebase App Hosting Docs](#)