

# AI Boyfriend Documentation

## **Front End:**

Vatche Balian (PM)

Brandon Lequang, Sean Solomon, Jeremy Luu

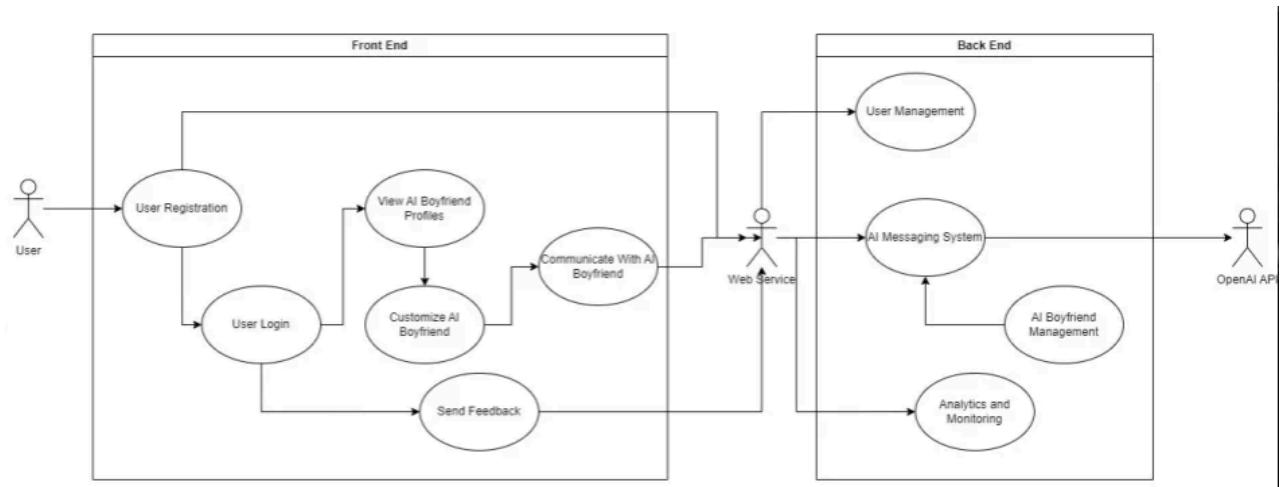
## **Back End:**

Brandon Leho (PM)

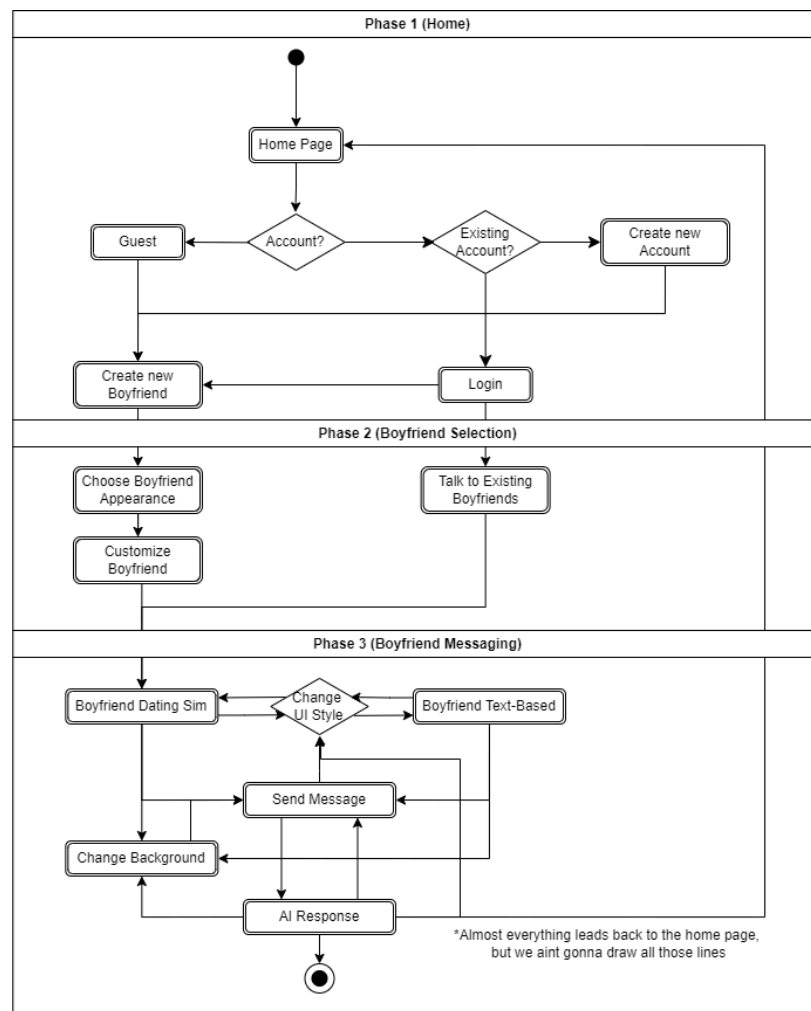
Miguel Sierra, Nicholas Ramirez-Ornelas, Juan Salas

## Functional/Requirement Spec (BA)

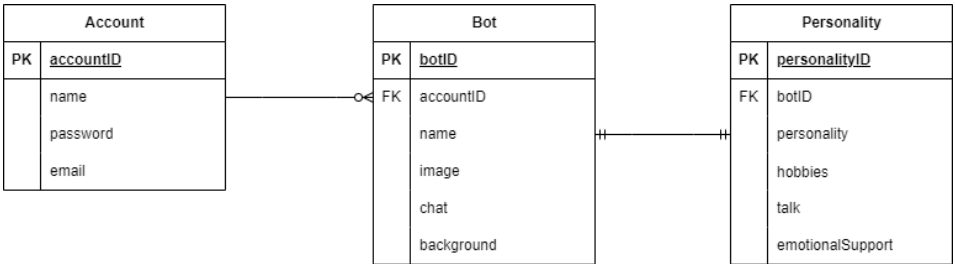
### Use Case Diagram:



### Activity Diagram:

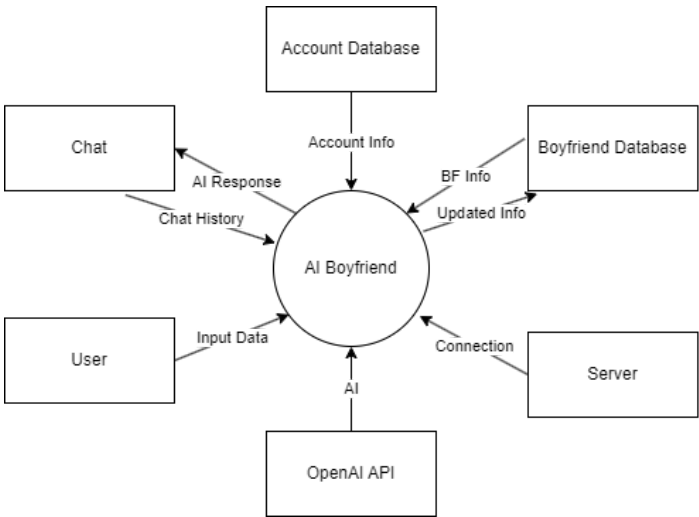


Domain Object Model:

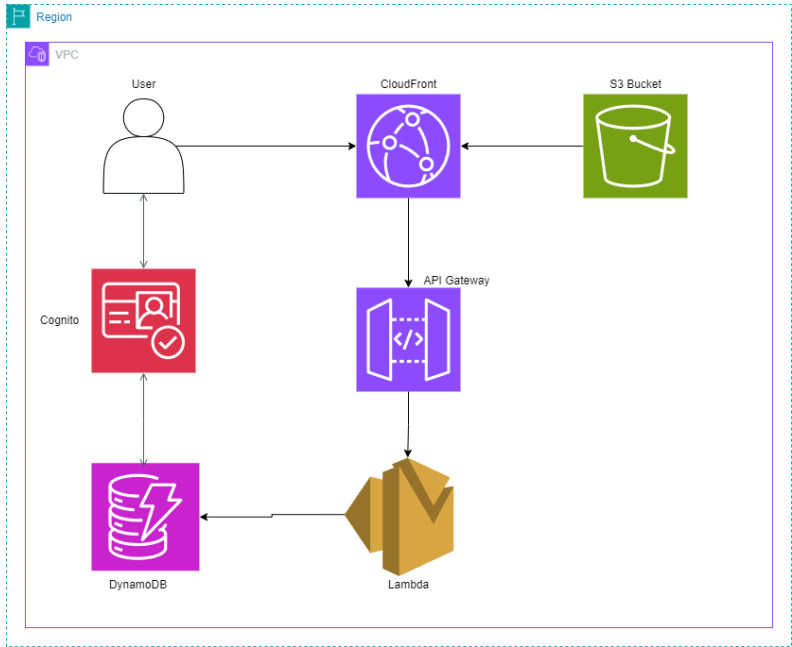


Technical/Design Spec (DEV)

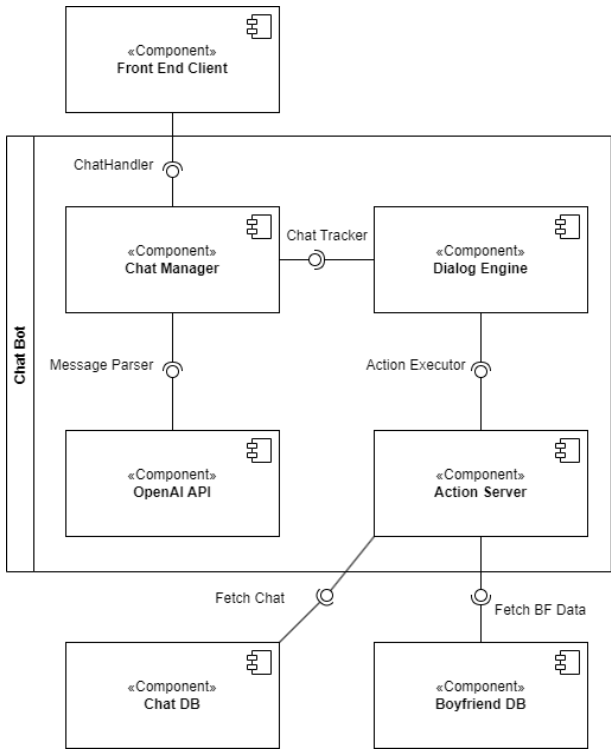
Context (Deployment) Diagram



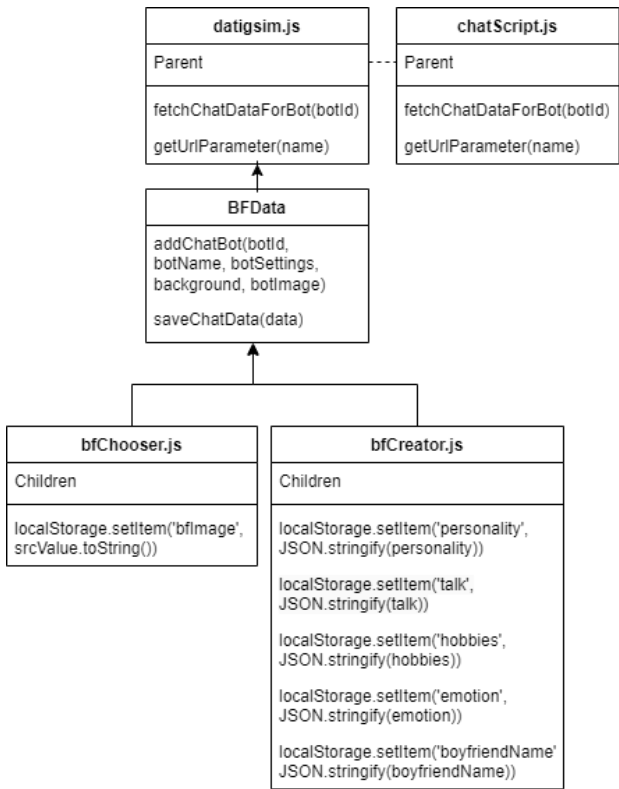
Architecture Layout:



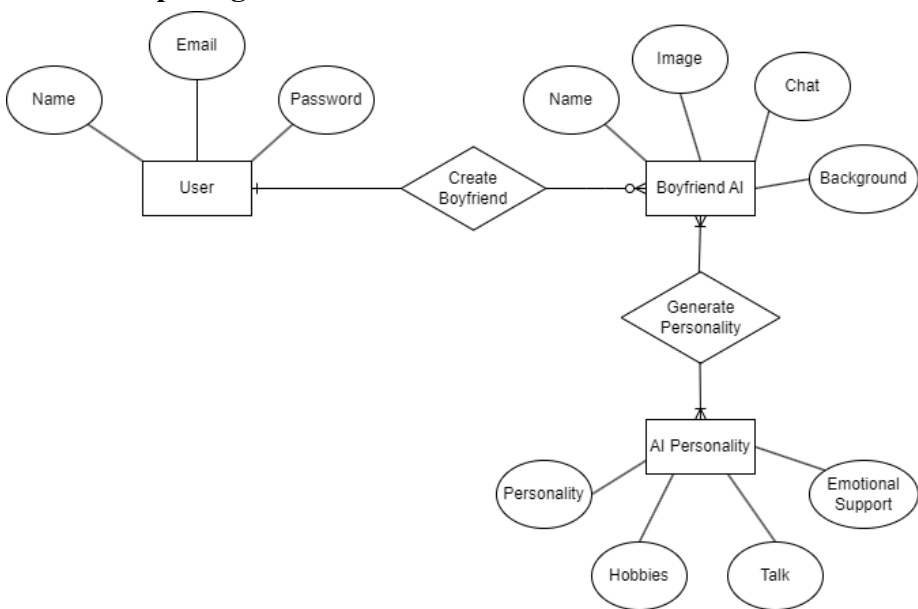
Component Diagram



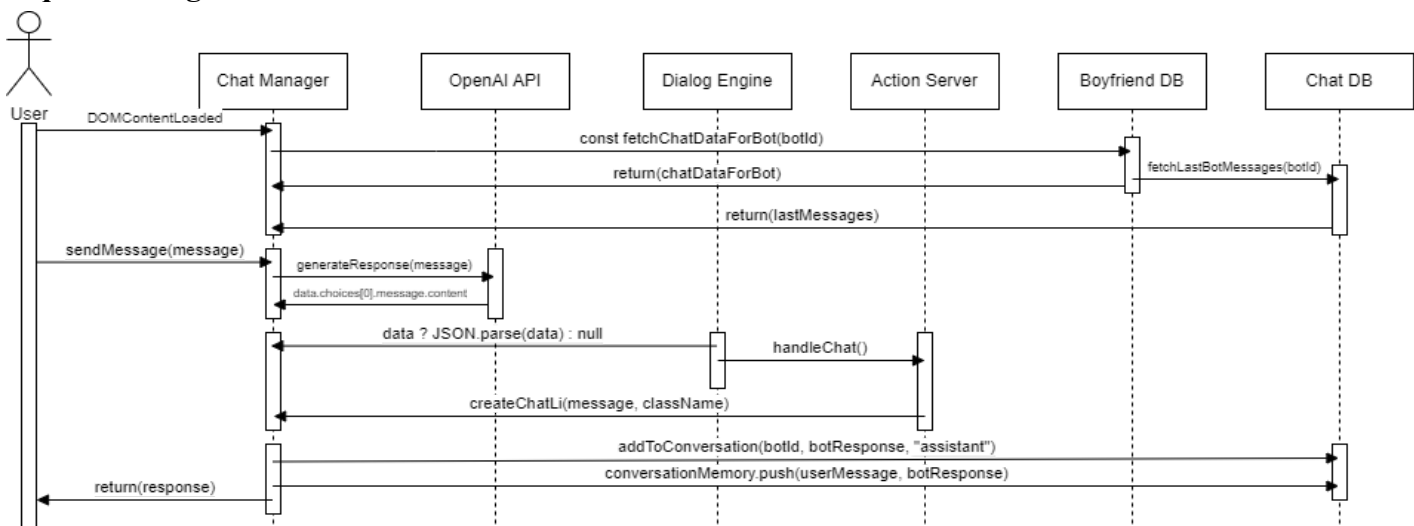
Class Hierarchy:



Relationship Diagram:



Sequence Diagram



Test Case Spec (QA)

Test Strategy/Approach

Feature Testing: Test each feature of the AI Boyfriend, such as chatting, responding to user inputs, providing advice, etc.

Compatibility Testing: Ensure the app works well across different web browsers (Chrome, Firefox, Safari, etc.) and devices (desktop, mobile, tablet).

Integration Testing: Verify that the AI seamlessly integrates with any third-party services or APIs used within the app.

Performance Testing: Assess the app's performance under various loads to ensure it responds quickly and efficiently to user interactions.

User Acceptance Testing: Involve real users (Other professors and students) in the testing process to gather feedback on the app's overall performance, including its ability to simulate a fulfilling relationship experience.

## **Test Plan**

### **1. Feature Testing:**

#### **a. Chatting Feature:**

- Test sending and receiving messages.
- Test different types of messages (text, emojis, etc.).
- Verify message delivery and receipt timestamps.

#### **b. Response to User Inputs:**

- Test various user inputs and validate the AI's responses.
- Ensure the AI understands and responds appropriately to different contexts.

#### **c. Providing Advice:**

- Test scenarios where users seek advice on different topics.
- Evaluate the quality and relevance of the advice provided by the AI.

### **2. Compatibility Testing:**

#### **a. Browser Compatibility:**

- Test the app on different web browsers (Chrome, Firefox, Safari, Edge).
- Verify consistent functionality and appearance across browsers.

### **3. Integration Testing:**

#### **a. Third-Party Services/API Integration:**

- Verify seamless integration with any third-party services or APIs.
- Test functionality dependent on external services (e.g., Text-to-Speech).

### **4. Performance Testing:**

#### **a. Load Testing:**

- Simulate varying levels of user traffic to assess app performance.
- Measure response times and server resource utilization under different load conditions.

b. Stress Testing:

- Test the app's stability and performance under extreme load conditions.
- Identify any bottlenecks or performance degradation.

5. User Acceptance Testing (UAT):

a. Real User Feedback:

- Invite professors and students to interact with the AI Boyfriend.
- Gather feedback on the app's overall performance and user experience.
- Evaluate the app's ability to simulate a fulfilling relationship experience.

6. Test Execution and Reporting:

a. Test Execution:

- Execute test cases according to the test plan.
- Record test results and any issues encountered during testing.

b. Defect Management:

- Log any defects found during testing using a defect tracking system.
- Prioritize and track defect resolution to ensure timely fixes.

c. Test Reporting:

- Prepare test reports summarizing test coverage, results, and any outstanding issues.
- Provide regular updates to stakeholders on testing progress and outcomes.

**Traceability Matrix**

	Test Case	TC1	TC2	TC3	TC4	TC5	TC6	TC6	TC7	TC8	# of Test Cases for Req
Req.											
Chatting		X	X	X				X	X	X	10+
AI Response			X	X				X	X	X	10+
Advice				X							1
Browser (Chrome)					X			X	X	X	4+
Browser (Firefox)						X					1
Browser (OperaGX)							X				2

Third-Party API								X			1
UAT (Students)									X		7
UAT (Professors)										X	3

## Test Cases

### Chatting Feature:

#### Message Sending

Description: Verify that users can send messages to the AI Boyfriend.

Steps:

- Open the chat interface.
- Type a message in the input field.
- Press Enter or click the send button.

Expected Result: The message is sent and displayed in the chat interface.

#### AI Response

Description: Verify that the AI Boyfriend receives and displays messages from the user.

Steps:

- Send a message from the user.
- Wait for the AI to respond.

Expected Result: The AI responds to the user's message appropriately.

### Response to User Inputs:

#### Emotional Response

Description: Verify that the AI Boyfriend responds with appropriate emotions to user inputs.

Steps:

- Send a message with a positive sentiment.
- Send a message with a negative sentiment.

Expected Result: The AI responds with matching emotions, e.g., joy for positive messages and empathy for negative messages.

#### Contextual Understanding

Description: Verify that the AI Boyfriend understands and responds appropriately to context.

Steps:



- Initiate a conversation on a particular topic.
- Introduce a related but different topic.

Expected Result: The AI maintains context and responds relevantly to the new topic or redirects the conversation smoothly.

### **Providing Advice:**

#### Advice Relevance

Description: Verify that the AI Boyfriend provides relevant and helpful advice to users' queries.

Steps:

- Ask the AI Boyfriend for advice on a specific topic.

Expected Result: The AI provides advice that is appropriate and useful for the given query.

### **Browser and Device Compatibility:**

#### Chrome Browser Compatibility

Description: Verify that the app works seamlessly on the Google Chrome browser.

Steps:

- Open the app using the Chrome browser.
- Interact with various features of the app.

Expected Result: All features work without any issues on the Chrome browser.

### **Third-Party Integration:**

#### TTS API Integration

Description: Verify that the AI Boyfriend can provide Text-to-Speech using a third-party API.

Steps:

- Prompt AI Boyfriend to respond to a message.
- See if TTS audio plays when response is given.

Expected Result: Audio of the text response is read out by the TTS API.

### **User Acceptance Testing:**

#### Feedback Collection

Description: Gather feedback from real users (professors and students) regarding their experience with the AI Boyfriend.

Steps:

- Provide access to the app to professors and students.
- Ask them to interact with the AI and provide feedback on their experience.

Expected Result: Collect qualitative feedback on the app's usability, effectiveness, and overall user experience.

### **Test Result Summary**

The purpose of this test summary report is to provide stakeholders with an overview of the testing conducted on the AI Boyfriend web app. The testing covered various aspects, including functionality, compatibility, integration, performance, and user acceptance.

#### Test Execution Summary:

Total Test Cases Executed: 39+

#### Key Findings:

The app showed a satisfactory performance across most test categories, with minor issues identified in compatibility and advice provision.

#### Functional Testing:

Chatting Feature: All test cases passed successfully. The AI responded appropriately to user inputs and maintained context during conversations.

Advice Provision: The AI provided relevant advice in most cases, but there were occasional instances of unrelated or generic responses.

#### Compatibility Testing:

Browser Compatibility: The app performed well on Chrome, Firefox, and OperaGX.

#### Integration Testing:

Third-Party API Integration: The integration with the weather API was successful, with accurate weather updates provided by the AI Boyfriend.

#### Performance Testing:

Load Testing: The app demonstrated robust performance under normal load conditions, with minimal latency observed during concurrent user interactions.

Stress Testing: The app maintained stability under stress conditions, although there was a slight degradation in response time with exceptionally high loads.

#### User Acceptance Testing (UAT):

Feedback from Professors: Overall, professors found the app engaging and entertaining, with positive feedback on the AI's conversational abilities.

Feedback from Students: Students appreciated the app's interface but suggested improvements in the variety and depth of advice provided by the AI.

### Conclusion:

The AI Boyfriend web app has shown overall good results in testing, with the majority of test cases passing successfully.

### **Source Code**

<https://github.com/BrandonLeho/AI-Boyfriend-Executable-Code/tree/main>

### **Project Build & Deployment Instructions**

#### Prerequisites:

- Ensure you have a modern web browser installed (e.g., Chrome, Firefox, OperaGX).
- Make sure you have a text editor or integrated development environment (IDE) installed for editing HTML, CSS, and JavaScript files.

#### Build Steps:

- Download or Clone the Project:
- Download the project files from the repository or clone the repository using Git:
- If necessary, update any configuration settings or variables in the HTML, CSS, or JavaScript files according to your preferences or requirements.
- \*replace "API\_KEY\_HERE" in datingsim.js and chatScript.js with OpenAI API key of your own\*

#### Open HTML File in Browser:

- Double-click the main HTML file ("home.html") to open it in your default web browser. Alternatively, right-click the file and select "Open with" to choose a specific web browser.

#### Interact with the App:

- Once the web page is loaded in your browser, you can create a boyfriend and then chat with the AI Boyfriend by typing messages in the chat interface and receiving responses from the AI.

#### Local Hosting:

- If you want to share the app with others on your local network, you can host it using a local server.

### **Project Release Notes**

Version: 1.0.0

Release Date: 6 May, 2024

Overview:

The AI Boyfriend web app version 1.0.0 represents the initial release of our virtual companion platform. This release introduces core features aimed at providing users with emotional support, relationship advice, and engaging conversations.

#### Key Features:

##### Chatting Feature:

Users can engage in conversations with the AI Boyfriend, receiving responses tailored to their inputs and emotional context.

##### Advice Provision:

The AI Boyfriend offers personalized advice on various topics, ranging from relationships to personal well-being, based on user queries.

##### Compatibility and Accessibility:

The app is compatible with major web browsers (Chrome, Firefox, OperaGX) and accessible

##### Improvements and Enhancements:

Enhanced natural language processing capabilities for more accurate understanding and responses.

Improved emotional intelligence algorithms for more empathetic and nuanced interactions.

Performance optimizations to ensure responsiveness and stability under varying loads.

##### Known Issues:

Occasionally, the AI's responses may not fully align with the user's input context, requiring further refinement of contextual understanding algorithms.

Lack of account setup and server side database. Right now users cannot make a personalized account and all of the data is saved locally, which can be potentially viewed as a good thing.

##### Future Roadmap:

None