

Cleyton Shelton Individual Project

Problem Domain

In today's world, conversational AI tools are widely available, but most of them operate as **linear chat streams** with minimal context control or conversation management. Users cannot easily organize, analyze, or branch their conversations, which limits the usefulness of AI for tasks like planning, learning, or coding assistance.

There is a need for a **structured, context-aware conversational platform** that allows users to:

- Maintain multiple independent conversations simultaneously
- Control how much past conversation context influences AI responses
- Search, branch, and export conversations for later use
- Customize the interface to suit personal workflows

Features & Requirements

Key Features (10 Total)

1. Real-Time Messaging

- Support real-time message exchange using WebSockets.
- Deliver messages in order and handle connection interruptions.

2. AI Response Generation

- Securely forward user messages to the OpenAI API via backend.
- Display AI-generated responses in the active conversation.
- Protect API credentials from exposure to the frontend.

3. Conversation Persistence

- Save conversations locally to persist across browser sessions.
- Restore conversations when the application reloads.
- Maintain message association with their respective conversations.

4. Multi-Conversation Management

- Create, rename, delete, and switch between conversations.
- Maintain independent message history for each conversation.
- Allow switching between conversations without data loss.

5. Context Depth Control

- Allow users to control how much previous conversation history is sent to the AI.
- Apply context changes immediately to new messages.
- Limit context size according to user preferences.

6. Conversation Branching

- Create a new conversation from a selected previous message.
- Preserve a reference to the original conversation.
- Ensure changes in the branched conversation do not affect the original thread.

7. Markdown and Code Rendering

- Render messages with Markdown formatting.
- Display code blocks with syntax highlighting.
- Ensure unformatted text remains readable.

8. Conversation Search and Highlighting

- Search for keywords within a conversation.
- Highlight matching terms in messages.
- Update search results dynamically as the input changes.

9. Exporting Conversations

- Export conversations as Markdown or JSON files.
- Preserve message order and timestamps in exported files.
- Allow exporting of individual conversations.

10. Theme and Interface Customization

- Switch between light and dark themes.
- Save theme preferences across sessions.
- Ensure the interface adapts to different screen sizes.

Development Roadmap

Product Development Roadmap

Sprint 1 Deliverables

1. Build the Real-Time Messaging System

- Implement sending messages from client to server via WebSockets
- Display messages in order in the chat UI
- Handle connection interruptions and automatic reconnection

2. Implement AI Response Generation

- Integrate backend proxy for OpenAI API requests
- Display AI-generated responses in the correct conversation
- Handle API errors gracefully and securely

3. Develop Conversation Persistence

- Store messages and conversations in IndexedDB
- Restore conversations on page reload
- Ensure message order is preserved per conversation

4. Implement Multi-Conversation Management & Basic Markdown/Code Rendering

- Enable creating, renaming, deleting, and switching conversations
- Render Markdown for messages
- Highlight code blocks for readability

Sprint 2 Deliverables

5. Add Context Depth Control

- Allow users to set how much previous conversation history is sent to AI
- Apply context depth immediately to new messages
- Limit context size according to user preference

6. Implement Conversation Branching

- Enable creating a new conversation from a previous message
- Maintain parent-child relationships between conversations
- Ensure original conversation remains unchanged

7. Add Conversation Search and Highlighting

- Implement keyword search within conversations
- Highlight matching terms in messages
- Update search results dynamically as input changes

8. Enable Exporting Conversations

- Allow exporting conversations as Markdown or JSON files
- Preserve message order and timestamps
- Support exporting individual conversations

9. Implement Theme and Interface Customization

- Add light and dark mode toggle
- Save theme preferences across sessions
- Ensure UI is responsive for desktop and mobile screens

10. Conduct Testing and Final Polishing

- Perform full manual testing of all features
- Fix bugs in messaging, AI responses, persistence, branching, search, export, and themes
- Prepare documentation and presentation materials

Questions?