Requirements Specification: Svelte Chatbot Application

Version: 1.0

Date: 2023-05-19

1. Introduction

   1.1 Purpose

   This document specifies the functional and non-functional requirements for the Svelte chatbot application. The application aims to provide users with an interactive platform to communicate with an AI chatbot using various models.

   1.2 Scope

   The requirements specified in this document apply to the Svelte chatbot application, including its user interface, API communication, and overall functionality.

   1.3 Definitions, Acronyms, and Abbreviations

   - API: Application Programming Interface
   - UI: User Interface
   - NASA: National Aeronautics and Space Administration
   - IEEE: Institute of Electrical and Electronics Engineers

2. Functional Requirements

   1. User Interface

      1. The application shall display a title "Chatter" to identify the chatbot application.
      2. The application shall provide a textarea for users to enter their messages.
      3. The application shall display a list of available models for users to select from.
      4. The application shall highlight the selected model using a dynamic gradient background.
      5. The application shall display the conversation history between the user and the AI, showing both user messages and AI responses.

   2. Model Selection

      1. The application shall fetch the list of available models from the API endpoint "/api/tags".
      2. The application shall allow users to select a model by clicking on the corresponding model element.
      3. The application shall update the selected model based on user interaction.

   3. Chat Functionality

      1. The application shall send a POST request to the API endpoint "/api/chat" with the selected model and user message.
      2. The application shall receive the AI's response from the API and display it in the chat history.
      3. The application shall support keyboard shortcuts for generating a response (Ctrl + Enter) and copying the chat history (Ctrl + Click).
      4. The application shall provide a "Clear Chat" button that allows users to clear the current chat history and start a new conversation.
         - While waiting for a response from the AI, the "Clear Chat" button shall change to "Chat In Progress" with the text bouncing left-to-right and right-to-left.

   4. Markdown Support

    1. The application shall use the svelte-markdown component to render the AI's responses as Markdown.
    2. The application shall display formatted and structured AI responses based on the Markdown syntax.
3. Non-Functional Requirements
    1. Performance
        1. The application shall provide a responsive user interface with minimal latency.
        2. The application shall optimize API requests to ensure fast response times.
    2. Usability
        1. The application shall have a user-friendly and intuitive interface.
        2. The application shall provide clear instructions and feedback to guide users through the chat process.
    3. Compatibility
        1. The application shall be compatible with modern web browsers, including Chrome, Firefox, Safari, and Edge.
        2. The application shall be responsive and adapt to different screen sizes and devices.
    4. Reliability
        1. The application shall handle errors gracefully and display appropriate error messages to users.
        2. The application shall recover from network failures and API errors without crashing.
    5. Maintainability
        1. The application shall follow modular and reusable code practices to facilitate future maintenance and updates.
        2. The application shall be well-documented, including inline comments and external documentation.
4. Constraints
    1. The application shall be developed using the Svelte framework.
    2. The application shall communicate with the specified API endpoints ("/api/chat" and "/api/tags") using JSON format.
    3. The application shall be deployed on a web server compatible with Svelte applications.
5. Assumptions and Dependencies
    1. The API endpoints ("/api/chat" and "/api/tags") are assumed to be available and functional.
    2. The application depends on the svelte-markdown component for rendering Markdown responses.
6. Future Enhancements
    1. Implement error handling and user-friendly error messages.
    2. Validate and sanitize user input to prevent empty or excessively long messages.
    3. Optimize the application's layout and styling for various screen sizes.
    4. Enhance the application's accessibility with ARIA attributes, keyboard navigation, and alternative text for images.