GEMINI CHATBOT

Team

VctrlC

Members:

Sahana Athota

Introduction

I developed this project with the goal of creating a functional chatting website with Gemini using the Gemini API. The platform supports seamless communication between user and Gemini, and features such as document summarization and question-answering using AI. This report is for outlining the tasks completed during the development of this project.

Team Roles

Initially, I had 2 team members who were supposed to join me in this hackathon. However, they withdrew from the project. So, I took on the project solo, which turned out to be a fun adventure of learning and growing.

Task 1: Creating a Functional Chatting Website

Initial Approach

Since the primary objective of this task was to build a chatting platform where the user can exchange messages smoothly with Gemini, I focused on integrating the Gemini API and building a simple and intuitive user interface for chatting.

Execution

1. **Frontend:**
    - A simple and intuitive user interface (UI) was developed for chat functionality. The interface includes a message input field and a message display area.
2. **Backend:**
    - On integrating Gemini API**,** messages sent by the user were processed through the backend, which communicates with the Gemini API.

Technical Challenges

- Efficiently relaying messages between users and the Gemini API for AI-based responses.

Task 4: File Upload Support

Initial Approach

The task was to allow the user to interact with the uploaded documents using the Gemini API. The platform processes the documents in real-time to generate summaries and provide answers to questions related to the document. I aimed at integrating the document processing capabilities of Gemini.

Execution

1. **Frontend:**
    - The user can upload PDF files by entering the file path in the chat interface.
    - After uploading a file, the user can either request a summary of the document or ask questions about its contents.
2. **Backend:**
    - The Gemini API processes the content from the file and generates a concise summary.
    - The user can ask questions related to the file, and the Gemini API analyses the content to provide relevant answers.

Technical Challenges:

- Difficulty in temporarily storing the files uploaded to cloud storage so currently this feature only works locally, but it is on the list of future perspectives of this project.

- Another challenge during file upload implementation was retrieving the full file path to pass to the Gemini API for document processing. Currently, users must manually enter the file path, which is a limitation but despite this, the system successfully processes the documents, providing summarization and question-answering through the Gemini API.

Task 6: File Upload Support

Initial Approach

To ensure secure access to the chat platform, user authentication was implemented. This guarantees that only authorized users can interact with Gemini and upload files.

Execution

1. **Frontend:**
    - A simple UI was designed for login and registration page to take in the username and password.
2. **Backend:**
    - A simple authentication flow was implemented where users can log in using a username and password.
    - The authentication system allows users to register by providing a username and password. Both of which are stored in a MongoDB database.
    - When users attempt to log in, the backend retrieves the corresponding user record from MongoDB based on the provided username.

Technical Challenges

- Implementing password storage and authentication flow.

Tasks Incomplete

I was unable to finish the following tasks due to my current stage of learning web development. Given the time constraints of the hackathon, I chose not to explore unfamiliar areas that could risk the stability of the project. As I'm still gaining experience, I am focusing on the functionality for now and not experimenting too much.

- Task 2: UI Improvements for Code Snippets
- Task 3: Result Data Streaming
- Task 5: Voice Chat Integration
- Task 7: Chat History Preservation
- Task 8: Deployment and Project Completion

Technologies used

1. **Frontend:** HTML, CSS, JavaScript
2. **Backend:** Node.js with Express.js
3. **API Integration:** Gemini API
4. **Database:** MongoDB

Conclusion

This project successfully delivered a chatting website with advanced features such as file uploads, summarization, and question-answering using the Gemini API. The combination of file processing with the Gemini API, and secure user authentication ensures an interactive platform.

Each task came with its own challenges, some of which I wasn't able to fully overcome due to my current stage in web development. However, these challenges remain on the list of future improvements for the project. Despite not being able to complete all the tasks, this hackathon has been an incredibly fun and valuable learning experience for me as a fresher. I’m excited to keep improving and work on the stuff I couldn’t do this time.