Support Agent Chatbot for CDP "How-To" Questions

Overview

This project aims to develop a support agent chatbot that answers "how-to" questions related to four Customer Data Platforms (CDPs): Segment, mParticle, Lytics, and Zeotap. The chatbot extracts relevant information from the official documentation of these CDPs to guide users on performing tasks or achieving specific outcomes within each platform.

Data Sources

The chatbot will pull information from the following official documentation sources:

- **Segment:** https://segment.com/docs/?ref=nav
- **mParticle:** https://docs.mparticle.com/
- **Lytics:** https://docs.lytics.com/
- **Zeotap:** https://docs.zeotap.com/home/en-us/

Core Functionalities

1. Answer "How-To" Questions

The chatbot will understand and respond to user queries about performing specific tasks or using features within each CDP. Examples:

- "How do I set up a new source in Segment?"
- "How can I create a user profile in mParticle?"
- "How do I build an audience segment in Lytics?"
- "How do I integrate my data with Zeotap?"

2. Extract Information from Documentation

- The chatbot will retrieve relevant instructions or steps from the official documentation.
- It will navigate through the documentation, identify relevant sections, and extract necessary information.

3. Handle Variations in Questions

- Handle different phrasings and lengths of questions.
- Detect and ignore irrelevant queries (e.g., "Which movie is releasing next month?").

Bonus Features

1. Clear Chat History

- When the user sends a "clear chat" command, all chat history should be deleted.

2. Cross-CDP Comparisons

- The chatbot can compare features and processes between different CDPs.

- Example: "How does Segment’s audience creation process compare to Lytics?"

3. Advanced "How-To" Questions

- The chatbot will provide guidance on advanced configurations, integrations, and use cases.

Tech Stack

Frontend:

- HTML
- CSS
- JavaScript

Backend:

- JavaScript (for chatbot logic)

Deployment:

- Hosted on a web server

Development Steps

1. **Chatbot UI Design:** Create a user-friendly interface using HTML and CSS.
2. **JavaScript Logic:** Implement chatbot behavior using JavaScript to process user input and display responses.
3. **Documentation Handling:** Store pre-defined responses based on extracted documentation content.
4. **Bonus Feature Implementation:** Add support for clearing chat history, cross-CDP comparisons, and advanced queries.

Future Enhancements

- Enhance chatbot intelligence with AI-based processing.
- Integrate APIs for real-time data retrieval.
- Improve response accuracy with continuous learning.

This README provides a structured overview of the project. For contributions and improvements, feel free to suggest or fork the repository once development begins!