**APCO Event Planning Assistant Documentation**

**Project Overview**

The APCO Event Planning Assistant is a web application designed to help users organize and manage various events. This application leverages OpenAI's language model through LangChain to provide AI-generated suggestions and answers to event planning queries. Built with Streamlit, it serves as a Minimum Viable Product (MVP) that showcases the core functionalities necessary for effective event planning.

**Technology Stack**

The MVP was developed using the following technologies:

* **Streamlit**: A fast way to create web applications in Python, allowing for quick prototyping.
* **LangChain**: A framework for developing applications powered by language models.
* **OpenAI**: For generating responses and suggestions related to event planning.
* **Matplotlib and WordCloud**: For visualizing user prompts as word clouds.
* **CSS**: Custom styles for enhancing the user interface.

**Project Structure**

The project is structured to facilitate easy maintenance and scalability. Here’s a breakdown of the key components:

/apco\_event\_planning\_assistant

|-- app.py # Main application file

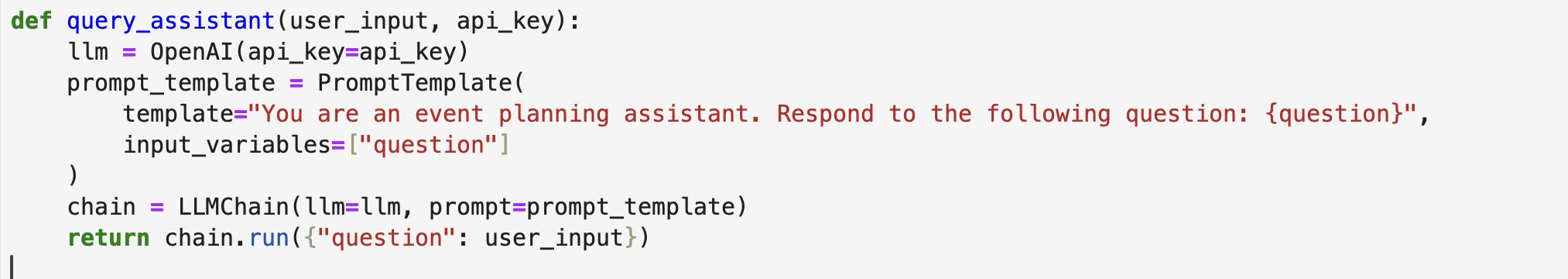
|-- requirements.txt # List of dependencies

|-- README.md # Project documentation

**Main Application File (app.py)**

This is the main file where the Streamlit application is defined. It includes the following core functionalities:

1. **User Input and AI Response**:
   * Users can ask questions related to event planning, and the application fetches responses using the OpenAI model.
   * Each query and its response are logged for future reference.



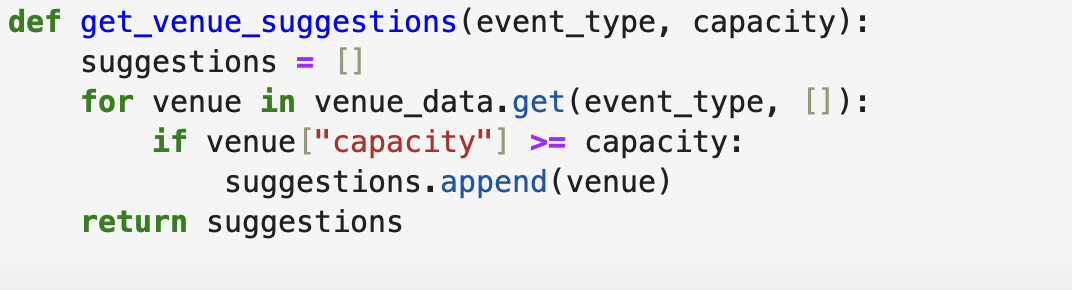
**Explanation**: The query\_assistant function initializes the OpenAI model with the provided API key, constructs a prompt template, and runs the LLM chain to get the assistant's response.

1. **Word Cloud Generation**:
   * The application visualizes the most common words from user prompts in a word cloud.

**A close-up of a computer code

Description automatically generatedExplanation**: The show\_word\_cloud function concatenates user prompts into a single string and generates a word cloud image using the WordCloud library, which is then displayed in the app.

1. **Venue Suggestions**:
   * Users can get venue suggestions based on event type and capacity requirements.

**Explanation**: The get\_venue\_suggestions function filters available venues based on user-specified criteria such as event type and required capacity.

**User Interface**

The UI is designed to be user-friendly and visually appealing, utilizing custom CSS for styling. Key features include:

* A sidebar for user authentication and application description.
* Tabs for different functionalities: querying the assistant, viewing chat history, generating word clouds, and getting venue suggestions.

**Example Usage**

1. **Asking the Assistant**: Users input questions about event planning, such as:
   * "Can you suggest venues for a wedding?"

The assistant provides a list of venue options that fit the criteria.

1. **Generating Word Clouds**: After several interactions, users can visualize the most common words used in their queries to identify trending topics.
2. **Venue Suggestions**: Users select the type of event and guest capacity, and the application responds with appropriate venue options.

**Future Development**

While this MVP serves its purpose well, transitioning to the full-stack application required in the job description involves the following enhancements:

1. **Next.js and TypeScript**: The application will be restructured using Next.js for a more robust frontend framework, leveraging TypeScript for type safety.
2. **Database Integration**: Supabase will be used to handle user data and chat history, ensuring persistent storage of interactions.
3. **Streaming Responses**: Implementing a streaming chat interface will enhance user experience by providing real-time feedback.
4. **Word Cloud Visualization**: Transitioning to amCharts for word cloud visualization as required in the project specifications.

**Conclusion**

The APCO Event Planning Assistant is a functional MVP that demonstrates the potential for an advanced event planning application. By utilizing powerful tools like LangChain and OpenAI, it offers a glimpse into how AI can streamline the event planning process. The planned enhancements will align the project with the full-stack requirements, providing a comprehensive solution for users.