VisualChat: Intelligent Charts and Conversations Documentation

VisualChat is a powerful Streamlit application that allows users to generate insightful charts and visualizations from CSV files while engaging in seamless conversations with their data. This documentation will guide you through the application's features, setup instructions, and code breakdown.

Features

- CSV File Integration: VisualChat seamlessly integrates with CSV files, enabling users to upload and analyze their data effortlessly.

- Interactive Charts and Visualizations: Users can generate various types of charts and visualizations based on their queries and data, providing valuable insights into trends and patterns.

- Natural Language Query: VisualChat leverages natural language processing to allow users to ask questions or provide prompts in a conversational manner, making data exploration more intuitive.

- Multi-Model Support: The application supports multiple language models, including ChatGPT-3.5, allowing users to choose the most suitable model for their needs.

- Code Generation and Explanation: VisualChat generates Python code to create the requested visualizations and provides explanations for the generated code, enhancing user understanding and transparency.

- Downloadable Content: Users can download the generated Python code, visualizations, and explanations for further analysis or sharing.

- Responsive User Interface: The application features a modern and responsive user interface with a consistent theme, providing an excellent user experience.

Setup Instructions

To run the VisualChat application locally, follow these steps:

1. Install Python: Ensure that you have Python installed on your machine. You can download the latest version of Python from the official website: https://www.python.org/downloads/

2. Install Required Libraries: Open a terminal or command prompt and navigate to the project directory. Then, install the required libraries using the following command:

```

pip install langchain openai pandas pillow streamlit streamlit-option-menu

```

3. Download the Application Files: Download or clone the repository containing the application files (`app.py`, `styles.css`, and any other necessary files).

4. Set OpenAI API Key: Obtain an OpenAI API key by creating an account at https://openai.com. Set the API key as an environment variable or modify the code to provide the key directly.

5. Run the Application: In the terminal or command prompt, navigate to the project directory and run the following command:

```

streamlit run app.py

```

This command will start the Streamlit application, and a local server will be launched. A URL will be displayed in the terminal or command prompt, which you can copy and paste into your web browser to access the application.

Code Breakdown

The VisualChat application consists of the following main components:

1. Imports: The necessary libraries and modules are imported, including `langchain`, `openai`, `pandas`, `streamlit`, and custom utility modules.

2. Streamlit Configuration: The application's page configuration, such as the title, layout, and initial sidebar state, is set using `st.set\_page\_config`.

3. Sidebar Menu: A sidebar menu is created using the `streamlit-option-menu` library, allowing users to navigate between the "About the App" and "Charts" sections.

4. CSV File Upload: Users can upload a CSV file using the `st.file\_uploader` function, which is then loaded into a pandas DataFrame.

5. Dataset Selection: Users can select the desired dataset from the uploaded CSV files using radio buttons.

6. Model Selection: Users can choose the language model(s) they want to use for generating visualizations and insights.

7. Query Input: Users can enter their natural language query or prompt in a text area.

8. Query Execution: When the "Generate" button is clicked, the application processes the user's query, formats it, and sends it to the selected language model(s) for analysis.

9. Visualization Generation: The application generates the requested visualizations using the model's output and displays them in the application.

10. Code and Explanation Display: The generated Python code for creating the visualizations and the model's explanation for the code are displayed in separate columns.

11. Download Functionality: Users can download the generated Python code, visualizations, and explanations using the provided download buttons.

12. Dataset Tabs: The application displays the uploaded datasets in separate tabs, allowing users to easily switch between them.