Financial Chatbot Documentation

Overview: The Financial Chatbot is a simple web-based application developed using Flask, a lightweight Python web framework. The chatbot leverages a predefined set of queries to provide users with financial insights based on a dataset of company financial data.

How It Works:

1. **User Input:** Users interact with the chatbot by entering company names, years, and predefined queries into the input fields provided by the web interface.
2. **Query Processing:** The Flask backend receives the user input and processes it using a Python function called **financial_chatbot**.
3. **Data Retrieval:** The **financial_chatbot** function accesses a dataset containing financial data for various companies and years.
4. **Response Generation:** Based on the user's input query, the **financial_chatbot** function retrieves the relevant financial data from the dataset and generates a response.
5. **Output:** The response is then returned to the frontend and displayed to the user in the web interface.

Predefined Queries: The Financial Chatbot can respond to the following predefined queries:

1. **Total Revenue:** “What is the total revenue?”
2. **Net Income Change:** "How has the net income changed?"
3. **Total Assets and Liabilities:** "What are the total assets and liabilities?"

Limitations:

1. **Predefined Queries:** The chatbot can only respond to queries that match the predefined formats. Queries outside these formats may not be recognized or processed correctly.
2. **Dataset Coverage:** The chatbot's responses are limited to the data available in the dataset. If the requested company or year is not present in the dataset, the chatbot will return a "Data not available" response.
3. **Natural Language Understanding:** The chatbot does not perform advanced natural language processing (NLP) to understand variations in user input. It relies on exact matches to predefined query formats.

4. **Error Handling:** Limited error handling is implemented in the current version of the chatbot. Invalid inputs or unexpected errors may result in generic error messages.

Future Enhancements:

1. **NLP Integration:** Improve the chatbot's ability to understand variations in user input through NLP techniques.
2. **Expanded Query Support:** Add support for additional financial queries and more flexible query formats.
3. **Data Visualization:** Integrate data visualization tools to provide graphical representations of financial data.
4. **User Authentication:** Implement user authentication to personalize responses or access restricted data.
5. **Error Handling:** Enhance error handling to provide more informative and context-specific error messages.