Financial Chatbot Documentation

Overview

The Financial Chatbot is a Python-based application developed using Flask, designed to provide financial information about specific companies. It operates through a web interface, allowing users to submit queries regarding various financial metrics.

How It Works

- Server Setup: The application sets up a Flask server which can handle web requests.
- Endpoints:
  - Root (`/`): Returns a welcoming message to the user.
  - Chatbot (`/chatbot`): This endpoint accepts POST requests and processes the user's input to return financial data in response to specific queries.
- Data Processing: Financial data for companies like Apple, Microsoft, and Tesla is pre-loaded into the server's memory. This data includes metrics such as total revenue, net income, total assets, total liabilities, and net cash from operations over the past three years.
- Request Handling: Users send requests with JSON payload specifying a company and a query. The chatbot parses this request, matches the company name and the query type, and retrieves the relevant data from its dataset.

Supported Queries

The chatbot can respond to the following queries:

- Total Revenue: Retrieves total revenue figures for the past three years.
- Net Income: Provides net income data for the past three years.
- Total Assets: Reports total assets over the past three years.
- Total Liabilities: Shows total liabilities data for the past three years.
- Net Cash from Operations: Returns the net cash generated from operations for the past three years.
- Net Income Increase: Calculates the percentage increase in net income from one year to the next.
- Total Assets Change: Computes the percentage change in total assets over a specific period.
- Total Revenue Change Percentage: Calculates the percentage change in total revenue over a specified period.

Limitations

1. Static Data: The chatbot uses a static dataset, meaning it cannot provide information beyond the predefined companies and time frames.
2. Limited Query Flexibility: The chatbot only understands specific predefined queries. Any deviation or incorrectly phrased query results in an error.
3. No Real-Time Data: The chatbot does not connect to external databases or APIs to fetch real-time financial data, limiting its usefulness for up-to-date financial analysis.

Conclusion

This Financial Chatbot serves as a basic tool for accessing historical financial data for a set of predefined companies. While effective within its scope, its functionality is limited by static datasets and a fixed set of queries it can respond to. Future enhancements could include real-time data integration and broader query handling capabilities to enhance its utility and user experience.