# **Documentation & Report: Financial Chatbot Prototype**

## **1. Introduction**

This project focuses on building a **prototype AI chatbot** that can answer a few predefined financial queries using company financial data (Apple, Microsoft, and Tesla’s 10-K filings).

Instead of implementing advanced natural language processing or machine learning, this simplified version uses a **rule-based approach** (if-else conditions). It introduces the fundamentals of chatbot design, data integration, and query-response mapping.

## **2. Data Preparation**

* **Input Data**: Three Excel files (appl\_10k.xlsx, msft\_10k.xlsx, tesla\_10k.xlsx) containing financial statements.
* **Key Metrics Extracted**:
  + Total Revenue
  + Net Income
  + Total Assets
  + Multi-year trends for revenue and net income

Each dataset was loaded into Python using **pandas** for easy access and manipulation.

## **3. Chatbot Design**

The chatbot is designed around **predefined queries** that users can ask.

### **Supported Queries**

1. **Total Revenue** → "What is the total revenue of [company]?"
2. **Net Income (single year)** → "What is the net income of [company]?"
3. **Net Income Change** → "How has [company]'s net income changed over the last year?"
4. **Total Assets** → "What are the total assets of [company]?"
5. **Revenue Trend** → "Show me the revenue trend of [company]."

The chatbot scans the input string for **keywords** (e.g., “total revenue”, “net income”, “Apple”) and maps them to the corresponding response from the data.

## **4. Implementation (Prototype Logic)**

The chatbot is implemented in Python with the following flow:

1. **Load data** with pandas.
2. **Normalize queries** to lowercase.
3. **Match query to rules** using if-else conditions.
4. **Return a predefined response** using values extracted from the dataset.

Example function structure:

def simple\_chatbot(user\_query):

if "total revenue" in user\_query:

return "Apple's total revenue in 2024 was $XYZ."

elif "net income" in user\_query and "change" not in user\_query:

return "Tesla's net income in 2024 was $XYZ."

elif "net income" in user\_query and "change" in user\_query:

return "Microsoft's net income increased by $XYZ compared to 2023."

else:

return "Sorry, I can only answer predefined queries."

## **5. Demonstration (Sample Queries & Responses)**

**User:** What is the total revenue of Apple?  
**Bot:** Apple's total revenue in 2024 was $XYZ.

**User:** How has Tesla's net income changed over the last year?  
**Bot:** Tesla's net income increased by $XYZ from 2023 to 2024.

**User:** What are the total assets of Microsoft?  
**Bot:** Microsoft's total assets in 2024 were $XYZ.

## **6. Documentation of Limitations**

* **Predefined Queries Only**: The chatbot can only respond to the queries explicitly programmed. Any variation in phrasing (e.g., “Tell me about Apple’s earnings”) may not be recognized.
* **Static Responses**: Answers are derived from fixed values in the dataset. If new data is added, the script needs updating.
* **No Advanced NLP**: The chatbot does not interpret context, synonyms, or conversational history.
* **Single Dataset Dependency**: Limited to the three provided company datasets.

## **7. Future Improvements**

* Add **natural language processing** to understand query variations.
* Connect to an **API or live database** for real-time financial data.
* Expand supported queries to include more metrics (EPS, operating expenses, liabilities, etc.).
* Create a **Flask web app** or **Telegram bot** for interactive use.
* Integrate **visualizations** (revenue trends, income growth charts).

## **8. Conclusion**

This prototype demonstrates how financial data can be integrated into a simple chatbot using Python. While limited in scope, it highlights the **foundational steps** in chatbot development:

* Preparing data,
* Defining queries,
* Mapping logic with rules, and
* Delivering structured responses.

It serves as an **educational stepping stone** towards building more sophisticated AI-driven financial chatbots capable of real-time analysis and interactive conversations.