

## Chatbot Prototype Documentation

### Project Name:

AI-Powered Financial Chatbot Prototype

### Objective:

To develop a simple chatbot capable of answering predefined financial queries using insights from the financial analysis of Microsoft, Tesla, and Apple (Task 1). This prototype helps demonstrate chatbot logic and basic interaction, laying the foundation for future AI-driven financial assistants.

### How It Works:

- User Interaction:**  
The chatbot interacts with users via the command line using `input()`. Users can type financial questions related to Apple, Tesla, or Microsoft.
- Query Matching:**  
The chatbot uses if-else statements to match input queries to predefined responses. The chatbot detects the company mentioned (Apple, Tesla, Microsoft) and key financial terms like "total revenue", "net income", "debt ratio", "cash flow", or "CAGR".
- Predefined Responses:**  
The chatbot responds with static data points drawn from financial analysis performed earlier. It provides information like total revenue, net income changes, debt-to-asset ratio, cash flow, and revenue CAGR.
- Fallback Logic:**  
If the user does not mention a supported company or financial metric, the chatbot provides a helpful fallback response prompting the user to specify a company and metric.

### Predefined Queries Handled:

- "What is the total revenue for [Company]?"
- "How has [Company]'s net income changed over the last year?"
- "What is [Company]'s debt-to-asset ratio?"
- "Which company has the highest operating cash flow?"
- "What is [Company]'s revenue CAGR?"

### Limitations:

- The chatbot only handles exact keywords like "total revenue" or "net income."
- It cannot understand variations or synonyms (e.g., "sales" instead of "revenue").
- It does not process free-form or follow-up questions.
- This is a prototype using **if-else logic**; no natural language processing (NLP) or machine learning models are included at this stage.

### Next Steps (optional improvements):

- Implement NLP capabilities (e.g., using spaCy or GPT API).
- Expand query flexibility and handle synonyms.
- Deploy via Flask or another web framework for a web-based interface.