Financial Intelligence Chatbot Assignment

Objective

Develop a robust, scalable financial chatbot capable of processing both structured and unstructured financial documents. The system should extract key insights, summarize content, and answer user queries based on uploaded data. The solution must automatically determine which processing action or tool to invoke, ensuring smooth integration between user intent and backend processing.

Functional Requirements

1. Data Handling & Analysis

- **File Types:**
  - Accept document formats including CSV, Excel, Links, PDF, and DOCX.
- **Data Extraction:**
  - Extract financial data such as trends, totals, statistical insights, and other performance metrics.
  - Interpret both tabular and textual financial information.
- **User Query Support:**
  - Respond to queries regarding specific financial metrics, comparisons, or trends.
  - Provide summaries and in-depth analyses upon request.

2. Autonomous Tool Invocation

- **Intent Recognition:**
    - Automatically detect user intent to decide if a specialized tool is required.
- **Tool Selection:**
    - Integrate with tools for summarization, CSV/Excel analysis, table querying, visualizations, etc.
    - Define clear capabilities, input types, and output formats for each tool.
- **Dynamic Switching:**
    - Transition between LLM-generated answers and tool-based outputs when necessary.

3. Tool System & Execution Logic

- **Modular Design:**
    - Develop modular tools (e.g., CSV analysis, text summarization, table querying) that can be invoked independently.
- **Registration & Description:**
    - Register and document each tool's capabilities for automatic detection and execution.
- **Loose Coupling:**
    - Ensure tools are loosely coupled and reusable across various financial data use-cases.
- **Expandability:**
    - Design the system to easily incorporate new tools as additional financial scenarios emerge.

4. User Interface

- **Frontend:**
  - Develop a user-friendly chat interface using Streamlit (with a Node.js backend recommended for API integration and file processing).
- **Features:**
  - Support real-time chat, file uploads, query input, and language selection.
  - Display outputs from both the LLM and any invoked tools in a clear and organized layout.

5. Multilingual Support

- **Language Detection:**
  - Automatically detect user language and adjust processing accordingly.
- **Manual Override:**
  - Provide users the option to manually switch conversation languages.
- **Financial Terminology:**
  - Ensure accurate translation and understanding of financial terms across different languages.
- **Extendability:**
  - Allow straightforward addition of new languages through configuration.

6. Document and Web Content Integration

- **Content Extraction:**
  - Extract key sections, tables, charts, and points from uploaded documents or online sources.
- **Context Matching:**

    - Accurately match user queries with relevant document content.
- **Real-time Summarization:**
    - Support dynamic summarization and lookups from document and web-based content.

7. History Management

- **Session-Based or Persistent Storage:**
    - Maintain chat history with timestamped interactions.
- **Document-Specific History:**
    - Track and store processing history for each uploaded document.
- **Export Features:**
    - Enable users to export or review their session history for auditing and tracking.

Examples of Queries

To illustrate the range of financial queries the chatbot should be able to handle, consider the following examples:

- **Financial Trend Analysis:**
    - "Show me the revenue trends for Q1 and Q2 from the attached Excel file."
    - *The chatbot should extract relevant data, plot the trends, and provide insights on the quarter-to-quarter performance.*
- **Comparative Analysis:**
    - "Compare the total expenses between the 2023 and 2022 financial reports in the PDF documents."

    - *The system should locate the relevant sections in both documents, calculate the differences, and generate a comparative summary.*

- **Statistical Summaries:**
    - "What is the average monthly profit according to the CSV data provided?"
    - *The chatbot must compute averages and other statistical measures from the tabular data.*
- **Detailed Document Summaries:**
    - "Summarize the key points from the DOCX document on market outlook."
    - *The LLM should produce a concise summary capturing the main insights from the uploaded report.*
- **Data Extraction from Tables:**
    - "Extract the top five products by sales from the embedded table in the Excel file."
    - *The tool should parse the table, sort the data, and return the top five entries.*
- **Multilingual Query Handling:**
    - "¿Cuáles son las tendencias de ingresos del último trimestre según el informe adjunto?"
    - *The chatbot must accurately detect the language (Spanish in this case), process the query, and return the analysis in Spanish.*
- **Real-Time Summarization and Lookup:**
    - "What are the latest market trends mentioned in the linked online report?"
    - *The system should fetch relevant content from online sources, summarize it, and provide actionable insights.*

Technical Considerations

Architecture

- **Modularity:**
    - Design the system as a collection of independent modules (data ingestion, processing, tool execution, response generation) that communicate seamlessly.
- **Scalability:**
    - Ensure the architecture can scale to handle a growing number of documents and concurrent users.
- **Error Handling:**
    - Implement robust error handling and fallback mechanisms for unexpected inputs or tool failures.

Integration & Tool Calling

- **Dynamic Invocation:**
    - The chatbot should autonomously decide when to call specific tools based on user queries.
- **API Integration:**
    - Use Node.js for backend API integration to handle file processing, document ingestion, and other computational tasks.
- **Seamless Transitions:**
    - Design the system to transition between natural language responses and specialized tool outputs smoothly.

Data Security & Privacy

- **Confidentiality:**

    - Ensure all financial data is processed securely and kept confidential.

- **Compliance:**
    - Adhere to relevant data protection regulations and financial data handling standards.

    -

Evaluation Criteria

Technical Competence

- **Code Quality:**
    - Clean, well-documented, and modular code.
- **Tool Integration:**
    - Effectiveness in tool invocation and seamless integration with the core LLM.
- **Scalability:**
    - Ability to handle large volumes of data and multiple concurrent users.

Innovation & Problem Solving

- **End-to-End Flow:**
    - Design an effective document processing pipeline.
- **Tool Selection:**
    - Choose appropriate tools to dynamically address user queries.
- **User Experience:**
    - Develop an intuitive and engaging user interface.

Multilingual & Cross-Platform Support

- **Language Accuracy:**
    - Ensure high accuracy in handling multiple languages, especially with domain-specific financial terminology.
- **Interface Adaptability:**

    - Maintain a consistent experience across different devices and platforms.

Documentation & Maintainability

- **Comprehensive Documentation:**
    - Provide detailed documentation of the system architecture, tool APIs, and codebase.
- **Extensibility:**
    - Ensure the system is easy to extend and modify with new features.

Submission Guidelines

- **Code Repository:**
  - Provide access to a Git repository containing the full source code, documentation, and setup instructions.
- **Demo:**
  - Include a working demo showcasing key functionalities.
- **Documentation:**
  - Submit comprehensive documentation detailing your design decisions, tool integration, and instructions for running and testing the system.
- **Evaluation Report:**
  - Provide a report explaining how your solution meets the assignment requirements and any challenges encountered during development.