Assignment: Real-Time Market Sentiment Analyzer Using LangChain Chains

Objective

Build a LangChain-powered pipeline (Chain) that:

1. Accepts a **company name** as input.

2. Extracts or generates its **stock code**.

3. Uses finance news search tools in LangChain to **fetch news** about the company.

4. Sends the news to an **LLM (Azure OpenAI GPT-4o or GPT-4o-mini)** to generate a structured **sentiment profile**.

5. Outputs the result as a **JSON object**.

6. Uses **mlflow** for tracing, prompt debugging, and monitoring.

Tech Stack & Tools

- **Framework**: LangChain

- **LLM**: GPT-4o or GPT-4o-mini (deployed via Azure OpenAI)

- **Data Source**: GOAT / Yahoo Finance tool / Brave Search / Exa Search in LangChain

- **Prompt Management & Observability**: mlflow

- **Environment**: Python (v3.10+ recommended)

Tasks Breakdown

Step 1: Accept Input

- Accept a company name as input (e.g., "Apple Inc").

Step 2: Get Stock Code

- Generate or extract the stock ticker/symbol using:

    o Either a static lookup table or an API/tool (like Yahoo Finance OR GOAT Symbol Suggest).

    o Integrate this as the first link in your chain.

Step 3: Fetch Company News

- Use LangChain's integration with new search tools to fetch the latest news for the extracted stock symbol.

- Extract a concise list of **recent headlines or article summaries**.

Step 4: Analyze Sentiment with GPT-4o / GPT-4o-mini

- Pass the news summaries to the GPT-4o model via LangChain with a prompt that asks the LLM to:

    o Classify sentiment.

    o Extract named entities: people, places, other companies.

    o Provide a structured JSON with the following fields:

```
{

  "company_name": "",

  "stock_code": "",

  "newsdesc": "",

  "sentiment": "Positive/Negative/Neutral",

  "people_names": [],

  "places_names": [],

  "other_companies_referred": [],

  "related_industries": [],

  "market_implications": "",

  "confidence_score": 0.0

}
```

Tip: Use StructuredOutputParser from LangChain or PydanticOutputParser for JSON formatting.

Step 5: Integrate mlflow

- Log prompts, outputs, and metadata using mlflow.

- Add tracing spans for:

    o Stock code extraction.

    o News fetching.

    o Sentiment parsing.

Deliverables

1. **Python Script / Notebook (.py or .ipynb)** with:

   o   Full implementation of the chain.

   o   LangChain Chain definition.

   o   Proper use of Azure OpenAI APIs and mlflow.

2. **README** with:

   o   Setup instructions.

   o   Azure and mlflow API configuration steps.

   o   Sample command to run the chain.

3. **Sample Output JSON** with a real example for a company like "Microsoft".

Bonus Ideas (Optional)

- Add **entity linking** for extracted people and companies (e.g., using Wikipedia or LinkedIn).

- Visualize sentiment trend if historical news is considered.

- Use LangChain's **MultiPromptChain** to classify and then process different types of news differently.

- UI using streamlit