AI Developer Multi-Agent Task – Real Estate Asset Management

This home task is designed to evaluate the problem-solving skills and technical

abilities of an AI developer candidate specializing in multi-agent system using LangGraph within the context of real estate asset management. This task should take approximately 6-8 hours to complete and is intended to be challenging and insightful, allowing you to demonstrate your thought process and technical expertise.

Task Overview

Develop a prototype multi-agent system using LangGraph that can assist with basic real estate asset management tasks based on natural language instructions. The agent should demonstrate logical reasoning, sequential processing, and the ability to handle varied and unexpected inputs.

Detailed Instructions

1. **Scenario:**

The agent will act as a virtual real estate asset manager assistant. The user will provide a request related to asset management, such as comparing property prices, calculating profit and loss (P&L), or retrieving asset details.

2. **Input:**

The input will be natural-language instructions, for example:

- "What is the price of my asset at 123 Main St compared to the one at 456 Oak Ave?"

- "What is the total P&L for all my properties this year?"

All types of question need to be handled

3. Processing:

Using LangGraph, the multi-agent system must:

- Detect the type of request (price comparison, P&L, asset details,, general knowledge etc.)
- Extract relevant details such as property addresses,ledgers, timeframes, and
- financial data from the input.
- Retrieve information from a simple dataset or API (state which you use)
- Perform calculations or assemble the requested information
- Generate a clear, step-by-step confirmation or response with relevant data.
- Remain robust to vague, incomplete, or unexpected inputs

4. Output:

The final output should be a clear and concise response. For example:

- "The asset at 123 Main St is priced at $500,000, while the asset at 456

Oak Ave is priced at $450,000.

- "The total P&L for all your properties this year is $1,200,000."
- "Details for the property at 789 Pine Ln: Address - 789 Pine Ln, Value -

$300,000, Last Appraisal Date - 2024-01-15."

5. Error Handling:

The agent should handle situations where:

- The property address does not exist in the dataset.
- The requested financial data is not available
- Ambiguous, incomplete or unsupported user instructions
- The input is not in the expected format

The multi-agent design should include fallback logic and clarification steps whe needed.

6. **Technology:**

Use **Python** and any LLM model, but **LangGraph must orchestrate the multi-agent workflow**. Document your implementation choices.A simple user interface (e.g., Streamlit or similar) should be included to allow basic interaction with the system.

7. **Nice to have:**

Documenting their choices and the reasoning behind them is a nice-to-have addition. It helps showcase their thought process, increases transparency, and provides useful context for understanding the design decisions.

Evaluation Criteria:

- Functionality: Does the agent perform the task as described?
- Code Quality: Is the code well-written, organized, and documented?
- Problem Solving: How effectively does the candidate handle unexpected
- inputs and edge cases?
- Technical Knowledge: Does the candidate demonstrate a good understanding of LLM models, orchestration, extraction and relevant programming concepts?
- Efficiency: Is the solution reasonably efficient in terms of processing time and
- resource usage?

Dataset:

A CSV file is provided separately.

Submission:

Candidates should submit:

- Complete Python code on github
- Short demo video (screen recording) of the agent in action
- README file with:
    - Setup instructions
    - Description of your solution and architecture
    - Multi-agent workflow with Langgraph
    - Challenges you faced and how you solved them

Expected Time Commitment

This task is estimated to take 6-8 hours. Candidates should manage their time

accordingly and provide a complete, working solution within this time frame.