**Ticker-Sense**

**Project Title** - Ticker-Sense

**Ticker-Sense**: An Intelligent stock market assistant.

**Ticker-Sense** is an innovative application designed to provide traders and investors with comprehensive stock information communicated in natural, human-like language. Leveraging data from Yahoo Finance and powered by OpenAI's advanced language models, StockSense offers real-time financial data, interactive charts, and a structured learning path for beginners entering the stock market.

**Project Objectives**

* **Real-Time Information**: Deliver up-to-date stock prices, financial statements, and market trends.
* **Natural Language Interaction**: Use AI to interpret user queries and provide understandable responses.
* **Visual Data Representation**: Offer charts and graphs for better data visualization.
* **Educational Support**: Provide a step-by-step learning path for beginners to understand stock market fundamentals.

**Model Building: LLM/API Integration Path**

**Optimizing API Integration and Application Testing**

**LLM Interaction:** For this project I am using yahoo finance python module – yfinance.

Though there were few limitation which I managed to resolved.

**Limitations:**

Initially Alphavantage api was used to fetch data but it has it’s limitation. After few queries it will stop responding for sometime. Due to this the proper response was getting delayed and the whole process breaking.

**Solution:**

**Yahoo Finance yfinance** python module. I searched for few of the finance apis and in that I got yfinance module which could be installed and consumed easily into the model.

**Challenge:**

Though I found the solution, the challenge was to learn yfinance library functions. I there is no proper documentation which could be found in on place. It took me two days to get to the basics and little bit of intermediate level. The app is running and it’s providing results but it require more fine tuning in yfinance response and in openai calls as well.

**Prompt Engineering**

I have to experiment a lot with prompt engineering part as the input and output are very critical for user.

**Input Prompt issue:**  
There could be multiple tickers/stock in one request with different financial requirement.

**Output Prompt issue:**

Output has to be very specific and within asked limit. It can not provide simple text every time. The response could be in various format and to choose a specific format according to the users request is a major challenge.

**Data Handling and Preprocessing**:

Now I am getting response from the function calls that openai is requesting from yfinance. But the issue as mentioned before of putting the cleaned and formatted data in well structured manner is important.

**Solution:**

I am working on formats to produce few results which are mandatory or best for stock information. This way most of the common queries could be provided in a single format. For others I am working on chat features. If time will permit then I will explore more in chat interactions.

**Functionality Testing:**

1. Yfinance
2. Openai
3. Function calling.
4. Each function in yfinance for different attributes and modules are being tested and defined separately.
5. Integration between openai and yfinance is sometime working with little glitch. On this I will talke to Mr Arshad tomorrow.
6. There are few more queries which I have to talk to him regarding full functionality.

**Integration of Model/API with Interface**

UI Selection & Setup

**Gradio** has been selected for this project.

Project is setup to respond financial queries related to stocks information. It will will provide data as well as charts.

**Backend to Frontend Communication:**

UI is setup properly using function call for various information. Right now 6 different functions has been added in the code to provide response. The same is also put required to but in two different output.

Gradio provide visual output in a easy format but when it comes to complex visuals it has it’s own challenges.

**Error handling**

Input error handling is being handled through step by step testing. Every stage it changing gradually.

Output error handling is more complicated as it breaks when unexpected calls are made and proper data is not called back.

Input being tests:

Q1. what is the status of graphic card.

Response: Its picking Nvidia stock and pushing all the information.

Q2. How can I learn trading.

Response: 8 Points in with little bit of details.