

UNIVERSITY OF ECONOMICS AND LAW
FACULTY OF INFORMATION SYSTEMS

GRADUATION INTERNSHIP

MAJOR IN DIGITAL BUSINESS AND ARTIFICIAL
INTELLIGENCE

**BUILDING A CHATBOT ABOUT SECURITIES
FROM STOCK TRADERS SYSTEM DATA
USING CHAT GPT**

Company:

StockTraders Investment Research Joint Stock Company

GVHD: TS. Le Hoanh Su

Student:

MSSV: K204161996

Full name: Nguyen Hoang Minh Nhat

Class: **K20416C**

Ho Chi Minh City, January 2024

ACKNOWLEDGMENTS

During her time studying at the University of Economics and Law, she has experienced a meaningful learning journey, receiving special support and attention from the teachers and the school management. I would like to express my sincere gratitude for the internship and internship opportunities that they have facilitated me to participate in, helping me to apply theoretical knowledge in practice before entering professional life.

In particular, I would like to express my deep gratitude to Dr. Le Hoanh Su, who was dedicated to guiding and accompanying me in the process of completing this report. At the same time, I did not forget to express my gratitude to StockTraders Investment Research Joint Stock Company, where I had the opportunity to intern. Mr. Nguyen Ngoc Chuc shared his knowledge wholeheartedly, bringing him valuable experiences and stimulating his confidence in the working environment.

I am well aware that this report is inevitable with limitations and shortcomings. It is the result of synthesis and reflection of the internship experience, and I hope to receive constructive feedback from teachers so that the report can be more complete in future work.

Finally, I would like to wish the teachers good health and success. At the same time, I hope that StockTraders Investment Research Joint Stock Company and all employees will continue to achieve many achievements and develop strongly in the future.

INTERNSHIP EVALUATION FORM

PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP

Họ tên sinh viên: Nguyễn Hoàng Minh Nhật

Tên Cơ quan thực tập: Công Ty Cổ Phần Nghiên Cứu Đầu Tư StockTraders

Địa chỉ Cơ quan: Lầu 5, IMC Tower số 62 Trần Quang Khải, P. Tân Định, Quận 1, TPHCM

Điện thoại Cơ quan0903.939986.....

Họ tên người đại diện cơ quan: Nguyễn Ngọc Chức.....

Chức vụ: CEO..... Điện thoại: 0908.939986.....

Thời gian thực tập tại Cơ quan: Từ ngày 25/11/2023..... Đến ngày: 13/02/2024.....

* Đề nghị Quý Cơ quan đánh giá bằng cách đánh dấu X vào cột xếp loại các nội dung đánh giá trong bảng sau.

Ghi chú :

Loại A: Tốt; Loại B: Khá

Loại C: Trung bình; Loại D: Kém

Nội dung đánh giá	Xếp loại			
	A	B	C	D
1. Tinh thần kỷ luật, thái độ				
1.1 Thực hiện nội quy của cơ quan	X			
1.2 Chấp hành giờ giấc làm việc	X			
1.3 Thái độ giao tiếp với đồng nghiệp, công sự	X			
1.4 Ý thức bảo vệ tài sản	X			
1.5 Tích cực trong công việc	X			
2. Kỹ năng chuyên môn, nghiệp vụ				
2.1 Đáp ứng yêu cầu công việc	X			
2.2 Tinh thần học hỏi, nâng cao trình độ chuyên môn	X			
2.3 Có sáng kiến, năng động trong công việc	X			
3. Kết quả thực tập				
3.1 Có sản phẩm ứng dụng thực tế đem lại lợi ích cho Cơ quan	X			
3.2 Mức độ hoàn thành nhiệm vụ thực tập	X			

INTERNSHIP AGENCY COMMENT FORM

PHIẾU NHẬN XÉT CƠ QUAN THỰC TẬP

Sinh viên Nguyễn Hàoáng Minh Nhứt - MSSV K204161996
đã hoàn thành实习 tập Công ty Cổ phần Nghiên cứu Đầu tư Stock Traders từ ngày 25/11/2023
đến ngày 13/02/2024.
Đánh giá quan điểm thực tập : XUẤT SẮC.

TP.HCM, ngày 13 tháng 02 năm 2024

XÁC NHẬN CỦA CƠ QUAN



Nguyễn Ngọc Chuẩn

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
INTERNSHIP EVALUATION FORM.....	iii
INTERNSHIP AGENCY COMMENT FORM.....	iv
TABLE OF CONTENTS.....	v
PHOTO CATEGORIES	vii
STATISTICS TABLE	viii
INTERNSHIP DIARY.....	x
BEGIN	1
1. Reason for the internship	1
2. Objectives real practice.....	1
3. Scope of Internship	2
4. Work Approach.....	2
5. Composition of exercise papers	2
CHAPTER 1: INTRODUCTION OF COMPANIES AND TOPICS	3
1.1. About StockTraders Investment Research Joint Stock Company.....	3
1.2. Products & Services	3
1.3. StockTraders Investment Philosophy.....	3
1.4. Business Requirements	4
CHAPTER 2: THEORETICAL BASIS	5
2.1. Stock market	5
2.1.1. Basic concept of the stock market	5
2.1.2. Types of goods on the stock market	5
2.1.3 Stock market	5
2.1.4 Bond market.....	6
2.1.5 Derivatives market.....	6
2.2 StockTraders AI System.....	6
2.2.1 System Overview	6
2.2.2 Important terms in the system.....	7
2.3 Open AI.....	11
2.3.1 Chat GPT	12
2.4 Swagger.....	13
2.4.1 Basic structure of Swagger files	14
2.4.2 Request body.....	15
2.4.3 Responses.....	15
CHAPTER 3: BUILDING A CHATBOT	18
3.1. Analyze stock data from the AI Stocktraders system using Python	18
3.1.1. API Connection.....	18

3.1.2. Data analysis and visualization	18
3.2. Create a Chatbot about stocks in Chat GPT	22
3.2.1. Build API connection statements.....	22
3.2.2. Configuration settings for Chatbots in ChatGPT	26
CHAPTER 4: ANALYSIS AND EVALUATION OF RESULTS	30
4.1. Chatbot Analytics and Reviews	30
CHAPTER 5: DEVELOPMENT DIRECTION	31
REFERENCES	32

PHOTO CATEGORIES

Figure 1. 1: StockTraders Investment Research Joint Stock Company.....	3
Figure 2. 1: Stock codes that make up the VN30 index in 2023[2].....	6
Figure 2. 2: The term "Waiting to buy"	8
Figure 2. 3: The term "Buy" in the AI Stocktraders system	9
Figure 2. 4: The term "Pending Sale" in the AI Stocktraders system.....	10
Figure 2. 5: The term "Sell" in the AI Stocktraders system.....	11
Figure 2. 6: Swagger Tool[2]	13
Figure 2. 7: Complete structure in swagger	15
Figure 2. 8: Example of a complete Request body in Swagger	15
Figure 2. 9: Examples of Respone in Swagger	16
Figure 2. 10: API response data with other formats	16
Figure 2. 11: Types of Status Codes and their corresponding descriptions.....	16
Figure 3. 1: API Connection in Python.....	18
Figure 3. 2: Collect and extract data into Dataframes	19
Figure 3. 3: Statistics describing the dataset	19
Figure 3. 4: Set the parameters for the line charts	20
Figure 3. 5: Visualize "buy", "sell", "waitbuy", "waitsell" data by date.....	20
Figure 3. 6: Pairplot Chart.....	21
Hình 3. 7: Correlation matrix	22
Figure 3. 8: Connecting the API in Swagger	23
Hình 3. 9: Schemas request.....	23
Hình 3. 10: Schemas response	24
Figure 3. 11: Enforce API Connection in Swagger	25
Figure 3. 12: Data returned after executing a connection in the API	26
Figure 3. 13: The main screen creates a chatbot in ChatGPT.....	27
Figure 3. 14: Fine-tune the configuration in the chatbot	27
Figure 3. 15: Add files and create options for chatbots	28
Figure 3. 16: Create an option by pasting the API connection statement in Swagger.....	28
Figure 3. 17: Testing the Post Method	29
Figure 4. 1: Chatbot Response	30

STATISTICS TABLE

Terminology	Describe
API	An API (Application Programming Interface) is a set of rules and protocols that allow other software to interact with each other.
REST API	A REST API (Representational State Transfer API) is a type of API designed to allow applications to communicate with each other through the HTTP protocol. REST APIs use HTTP methods such as GET, POST, PUT, DELETE to perform operations on resources such as data, images, or other information on the server.
Schema	In Swagger, "schema" is the part of the API description that is used to define the data structure of API requests and responses. It defines data fields, data types, and their constraints.
Media types	In Swagger, "media types" are part of the API description that defines the data format that the API can send or receive. This can be formats such as JSON, XML, or YAML.
content object	In Swagger, a "content object" is a part of the API description that is used to specify the data format that a request or response can contain.
Markdown-formatted	In Swagger, "Markdown-formatted" refers to the use of Markdown text formatting to write and format API descriptions. Markdown is an easy-to-read and easy-to-write markup language that allows you to

create titles, text formats, links, and lists
with ease

INTERNSHIP DIARY

Time	Content	Detailed work
25/11 – 03/12	Stock Market Overview	<ul style="list-style-type: none"> - History - Identify and analyze commodities on the stock market
04/12 – 17/12	Learn about technical analysis	<ul style="list-style-type: none"> - Apply Python to draw charts and technical analysis indicators - Research and learn three popular trading strategies based on technical analysis including: Trend-following, Mean-reversion strategies, Momentum strategy.
18/12 – 24/12	Learn about chat gpt	<ul style="list-style-type: none"> - Chat gpt overview - How to create a chatbot in chat gpt - Create an api connection action in chat gpt
25/12 – 07/01	Learn about swagger	<ul style="list-style-type: none"> - Swagger Overview - Data Models (Schemas) trong swagger - How to Write a Request Body - How to Write a Response

08/01 – 21/01	Make an api connection in chatgpt	<ul style="list-style-type: none"> - Build code, test, evaluate api connection from swagger - Implement code in gpt chat
22/01 – 13/02	Retrieve the API built in chat gpt	<ul style="list-style-type: none"> - Check that the model has responded correctly to the data from the connection api. - Write instructions for the chatbot to properly execute query requests to the API

BEGIN

1. Reason for the internship

The learning process at the school is not only the first step for students to get acquainted with their chosen major, but also an opportunity for them to access the necessary theoretical knowledge. However, this is not enough. In today's era, with the continuous development of society, it is extremely important to combine theoretical knowledge with practice. Only when students have the opportunity to contact and apply the knowledge they learn in the lecture hall in practice, can they really develop and contribute to society effectively.

The internship process is an important step to help students not only get acquainted with the real environment but also determine the direction of their future career. At the same time, it is also an opportunity to build self-confidence and overcome the fear of facing a real work environment. Through this process, students have the opportunity to approach large businesses, opening up bright future prospects. That's why I chose Stock Traders Investment Research Joint Stock Company to carry out my internship.

Stock Traders Investment Research Joint Stock Company is not only an ideal destination for many people but also an environment that promotes listening and input from all employees but is part of the collective. That's why I chose this company to carry out my internship. During that time, I participated in the project "Building a Chatbot about securities from Stock Traders system data using Chat GPT".

2. Internship objectives

- Have the opportunity to directly approach and observe activities in their profession. This helps to better understand the requirements and challenges of the job, thereby developing the skills and insights necessary to succeed in their careers.
- Be able to evaluate yourself, recognize your strengths as well as weaknesses. From there, you can build a plan for your own development, take advantage of your strengths and improve your weaknesses before entering the actual working environment.
- Create an overview of a real work environment, be able to record important information and experiences as a resource for each individual.

3. Scope of Internship

Operating and working at Stock Traders Investment Research Joint Stock Company.

4. Work Approach

The methods applied to implement the project include: statistical research methods, analysis methods, comparison methods, and synthesis methods. In addition, I also applied programming theories to carry out the project .

5. Composition of exercise papers

The report layout consists of 5 parts:

Chapter 1: Introduction of companies and topics

Chapter 2: Theoretical Basis

Chapter 3: Building a Chatbot

Chapter 4: Analysis and Evaluation of Results

Chapter 5: Development Direction

References

Reference link

CHAPTER 1: INTRODUCTION OF COMPANIES AND TOPICS

1.1. About StockTraders Investment Research Joint Stock Company



Figure 1. 1: StockTraders Investment Research Joint Stock Company

StockTraders, with a history since 2013, is known as one of the leading units in the field of providing stock investment strategy systems in Vietnam. With the advanced application of 4.0 technology and the use of proprietary algorithms in investment analysis, the company is committed to providing accurate and reliable recommendations to help customers in investing in stocks, minimizing risks and increasing profit opportunities.

1.2. Products & Services

Providing trading signals for buying and selling Vietnamese stocks with three main objectives:

- Assist investors in making timely trading decisions in the market.
- Providing advice on strategies for selecting potential stocks.
- Adjust risks by dividing investment capital appropriately.

In addition, the product also supports multiple platforms to provide recommendation updates quickly.

1.3. StockTraders Investment Philosophy

Market waves

The law of market volatility consists of three main stages: Uptrend, Downtrend and Sideways. Statistical data from the system shows that if trading is carried out in all these stages, the highest probability of success is only 50%, and the remaining 50% will be in two stages: Downtrend and Sideways. "Therefore, focusing on trading during bull markets will increase the chances of success and minimize the need to cut losses."

In the Uptrend period, more than 80% of stocks rose in price, and vice versa in the Downtrend period. However, determining when the market starts an uptrend and when a downtrend starts will be warned by the StockTraders system a few sessions in advance.

Select Stocks

When the market appears in waves, investors often have difficulty choosing stock codes to invest in, because there are so many options on the three main exchanges, HSX, HNX, and UPCOM, with more than 1,500 stock codes. If you choose the wrong stock code, it can lead to large losses or only achieve limited profits, even if the market is rising sharply. "StockTraders possesses an algorithm that searches for potential stock symbols and provides push notifications to investors instantly."

Disbursement strategy

Mastering the above two factors, investors can confidently trade the entire account when the market is in an uptrend phase to optimize profits, and act cautiously when the market is weak. In the event that the extent of the market wave cannot be clearly determined, the limited disbursement can lead to low profits while the market rises sharply. The system also provides push notifications about the proportion of stock purchases at each time to optimize the cost price. The buying strategy is divided into three times with different weights: 30%, 50% and 20%. The allocation of portfolio weights is a difficult problem for the departments that make investment strategies in the market, but StockTraders has developed an automated algorithm to alert investors through an app.

1.4. Business Requirements

Get a brief overview of the stock market, common commodities on the stock market, and get an overview of what the nature of the stock market is.

Learn how to create an API connection code using Swagger.

Learn about the functions included in Chat GPT plus, thereby creating a separate Chatbot with self-adjusting configurations according to the company's needs.

The company's API link with the Chatbot has developed, carried out the process of testing, evaluating, and optimizing the Chatbot.

CHAPTER 2: THEORETICAL BASIS

2.1. Stock market

2.1.1. Basic concept of the stock market

The stock market is where financial assets such as stocks, bonds, investment funds, and other securities are traded. These transactions usually take place through stock exchanges, where buyers and sellers can meet to make buying and selling transactions.

The stock market plays an important role in providing capital for businesses to expand their business scale, invest in technology, and research and develop products. At the same time, it is an important channel for investors to participate and seek profits from trading on the stock market.

The stock market is usually organized and operated by stock exchanges on a global scale. These exchanges serve as a place to buy and sell securities through brokers and trading agents. In Vietnam, the stock market is managed by the State Securities Commission under the Ministry of Finance. [1]

2.1.2. Types of goods on the stock market

Based on commodities on the stock market, we classify the stock market into three types, including: stock market, bond market, and derivatives market.

2.1.3 Stock market

The stock market is a place where investors can buy and sell shares of companies as well as equivalent assets on a stock exchange. Shares denote a portion of ownership in a company and allow investors to participate in the governance and growth of that company. The stock market often reflects the volatility of the economy, the performance of companies, as well as other factors such as market sentiment and the influence of events and news.



Figure 2. 1: Stock codes that make up the VN30 index in 2023[2]

2.1.4 Bond market

The bond market is where companies, governments, or other credit units issue bonds to raise capital from investors. Bonds are a form of financial instrument in which the buyer becomes a lender of money to the issuer, receiving interest back at a fixed or variable rate. Investors can buy and sell bonds in a specific market known as the bond market. It is an important part of the financial market, providing an opportunity for investors to invest in asset classes with fixed or variable profit margins.

2.1.5 Derivatives market

The derivatives market is where derivative financial products such as options, futures, and other derivatives are traded. These products are created from the value of an underlying asset such as stocks, indices, commodities, or exchange rates. Investors can buy or sell these derivatives contracts with the goal of hedging risk or speculating in the stock market.

Derivatives allow investors to control risk, take advantage of speculative opportunities, or even create complex strategies to achieve their investment goals. The derivatives market often operates on specialized exchanges and requires in-depth knowledge of finance and risk from investors.

2.2 StockTraders AI System

2.2.1 System Overview

The AI StockTraders system is a breakthrough platform in the field of stock investment in Vietnam, born in 2013. With a mission to make stock investing simpler,

StockTraders applies the philosophy of using 4.0 technology and proprietary algorithms to investment analysis. This system helps investors capture and profit from market waves through recommendations based on technical analysis data and Big Data. Developed by StockTraders with the goal of increasing the trading success rate by over 80%, AI StockTraders has provided services to more than 10,000 clients, bringing a new, effective investment method, based on extensive and timely data analysis. at the same time, it reshapes investment strategies for investors by providing online stock trading signals and analyzing market waves.

2.2.2 Important terms in the system

Waiting to buy

"Waiting to buy" in the StockTraders AI system is a proprietary term, indicating that the market has the potential to make a bottom and the possibility of price growth in the near future. It is a warning signal that helps investors identify when and when to buy stocks. The system uses AI technology and proprietary algorithms to determine the number of "Pending Buy" shares, providing easy-to-understand information through the number of shares and visual charts.

As the number of "Pending Buy" stocks increases, especially when it reaches the 60-symbol mark, it marks the transition from a weak market to a market with growth prospects. The higher the number of "Pending Buy" stocks, the greater the possibility of the market bottoming out and going up. This ensures that users have the most up-to-date and accurate information on market conditions, helping them take advantage of investment opportunities and maximize profits.

"Waiting to buy" is not only data on overbought data on the electronic price list, but also important information about the possibility of reversal of the stock market, provided by the AI StockTraders system to identify investment opportunities.



Figure 2. 2: The term "Waiting to buy"

Buy

In the AI StockTraders system, "buying" is the act of investing in a stock when the investor believes that the market will rise in price or has reached the bottom and is ready for a new bullish period. This action is based on analyzing and positioning the market, as well as identifying signals from the system, such as the number of "Buy" shares. When the "Buy" data shows an upward trend, it may be a favorable time to buy stocks with a low weight, waiting for additional buy signals in the future.

In some cases, despite the spread of bad news in the market, if the number of "Buy" shares increases, this can be interpreted as some investors seeing the upside potential of the stock and deciding to buy. This shows a different investment mentality and strategy, as they can see the opportunity in the decline and believe in the subsequent recovery of the market.



Figure 2. 3: The term "Buy" in the AI Stocktraders system

Pending Sale

In the AI StockTraders system, a pending sell is a status or signal that the system detects based on market data analysis, in order to indicate that a certain number of stocks are being considered by investors for sale in the near future. This usually involves anticipating or identifying when and under market conditions at which the sale of shares may benefit or mitigate risk to the investor.

Waiting for a sell-off to rise can be a sign that the market is approaching or has reached a peak, where investors have expectations that the stock price will fall, so they are prepared to sell to take profits or cut losses. In the StockTraders AI system, a "sell" signal is used to alert investors to potential changes in market trends, helping them make smarter investment decisions based on analytical and forecast data.



Figure 2. 4: The term "Pending Sale" in the AI Stocktraders system

Sell

In the AI StockTraders system, "sell" is mentioned as part of the investment decision process, especially in the context of market wave analysis and alerts from the system. Selling is an action taken by an investor to exit a position on one or more stocks they own, based on market analysis and forecasts, or to take profits or stop losses. In the AI StockTraders system, selling is not only the final decision of the trading process, but also a warning signal emitted by the system based on data and analysis algorithms to identify the optimal time for exiting a stock position.

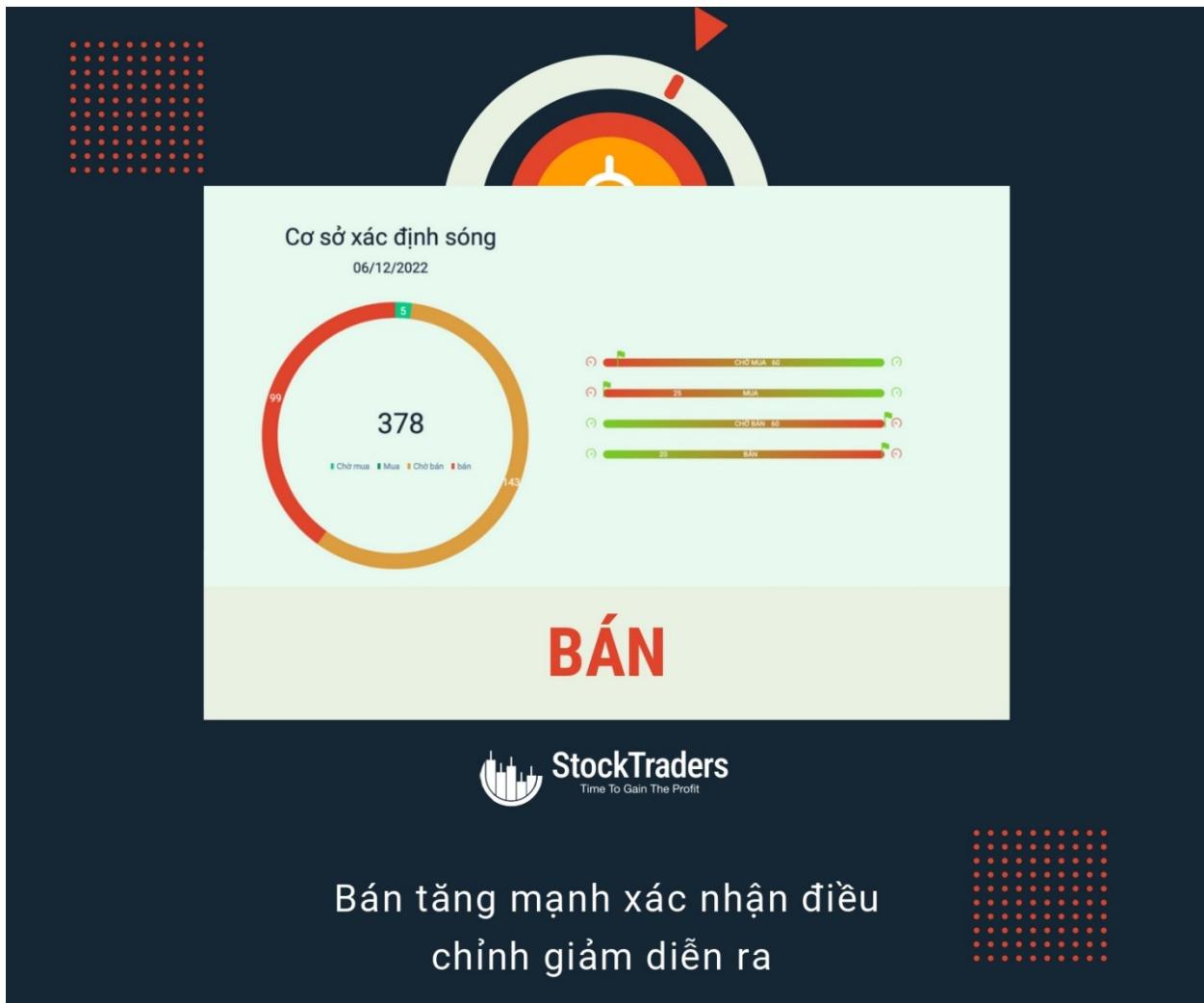


Figure 2. 5: The term "Sell" in the AI Stocktraders system

The terms used are intended to support the analysis of data from the Stocktraders AI system, in order to generate assessments and forecasts tailored to the specific context of the system.

2.3 Open AI

OpenAI is an artificial intelligence research lab, consisting of a team of researchers, engineers, and executives, focused on developing and conducting research on artificial intelligence in a safe and beneficial way. The organization was founded in December 2015 by some of the top leaders in the tech industry, including Elon Musk, Sam Altman, Greg Brockman, Ilya Sutskever, John Schulman, and Wojciech Zaremba.

OpenAI focuses on developing cutting-edge artificial intelligence technologies and algorithms and assisting the research community and businesses in accessing these tools. They work on a variety of AI projects, including natural language processing, computer vision, and robotics. The organization has also created a number of advanced AI models,

such as the GPT (Generative Pre-trained Transformer) language model, which is capable of generating human-like text.

2.3.1 Chat GPT

ChatGPT is a chatbot application built on the architecture of GPT-3.5 and GPT-4, designed to create a human-like conversational experience by processing input from natural language and generating meaningful responses. The platform is accessible through any web browser and at the same time it also has a mobile app platform.

Here are some of the popular features that Chat GPT offers:

Answer the question

ChatGPT is capable of answering questions just like other chatbots can. However, this can convey complex ideas through a variety of styles and ways of communication. In addition, it can also respond to questions that users ask, such as from search engines or virtual assistants.

Text Generation

The texts generated by ChatGPT can feel like they are generated by humans. Writing, creating content, and using other apps can all take advantage of this.

Language Translation

ChatGPT can be applied to improve performance in language translation, a feature that can be particularly beneficial for applications such as customer service or chatbots.

Shorten text content

Adapting text summarization tasks with ChatGPT can be a useful solution for creating articles, long documents, and other types of text.

Alternatives to Google Search

ChatGPT has the ability to replace Google and compete with other chatbots, because it is capable of intelligently responding to almost any question. The only drawback of Chatbot is the need for reference data sources.

Complete missing documents

Features such as predictive text input and other applications can take advantage of ChatGPT's fine-tuning feature in text completion tasks. However, it is important to

remember that the level of refinement, the quality of the data used to tune the model, and the specific context of use all affect the quality of the output.

In this internship project, we will apply Chat GPT Plus version to develop a Chatbot that inherits a large language from OpenAI, along with providing the Chatbot with data from the AI Stock Traders system to learn.

2.4 Swagger

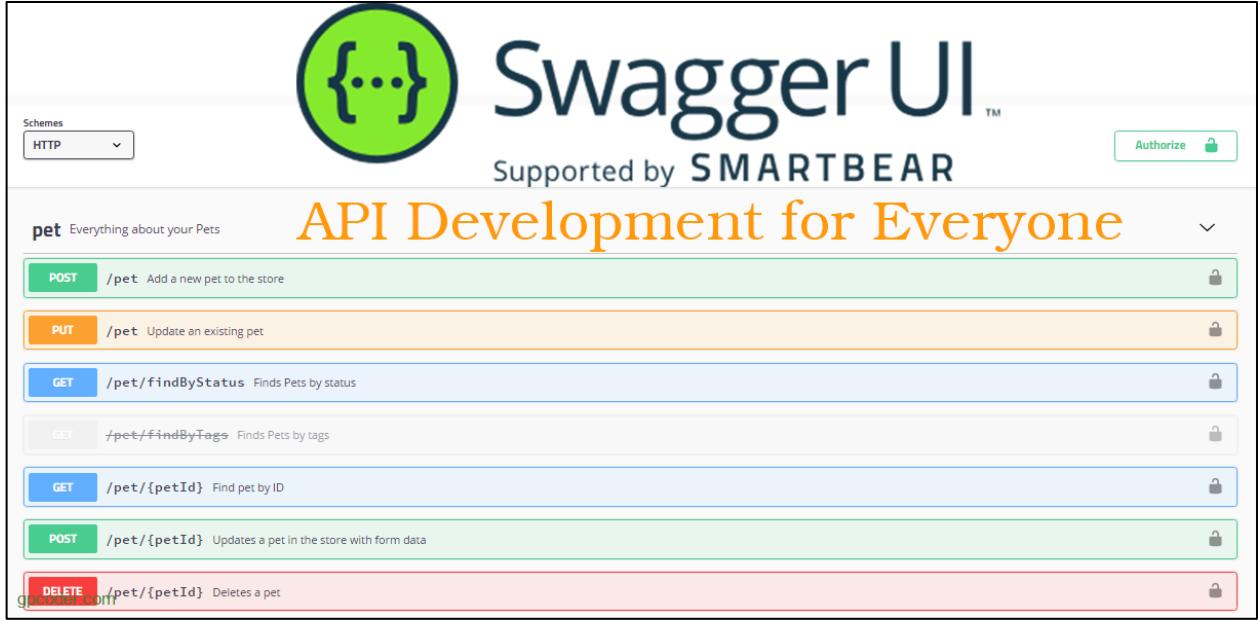


Figure 2. 6: Swagger Tool[2]

Swagger is a suite of open-source tools used to create open API descriptions that make it easier to design, document, and use REST APIs. It offers three main tools for developers:

Swagger-Editor: used to create and edit OpenAPI API description files. It provides a user-friendly graphical interface for manipulating API document formats and previewing how the API will work. Swagger-Editor makes it easy for developers to create, modify, and test API descriptions efficiently.

Swagger-Codegen: used to automatically generate source code for APIs based on their descriptions in the OpenAPI standard. It allows developers to generate source code for clients and servers automatically, reducing the time and effort required to create system components. Swagger-Codegen supports a variety of programming languages and frameworks, making API integration flexible and convenient.

Swagger-UI: used to display and interact with APIs described in the OpenAPI standard. It provides a user-friendly web user interface, allowing users to preview API documentation, test endpoints, and interact directly with requests and feedback from the API. Swagger-UI helps improve the user experience when working with APIs, providing an intuitive and convenient way to explore and use web services.

2.4.1 Basic structure of Swagger files

A Swagger file can be written in JSON or YAM, it consists of the following 3 parts

- **Metadata:** API description information such as API name, version, description, contact information, and many other API-related information. Metadata helps users understand the API and how to use it.
- **Base Url:** The base URL of the API, i.e., the address on which all API endpoints will be built. Base URLs provide a starting point for calling API endpoints easily and consistently.
- **Consumes, Produce:** Consumes defines the data formats that the endpoint can accept from the request, such as JSON, XML, or form data. Meanwhile, Produces defines the data formats that an endpoint can generate from a response, such as JSON, XML, or HTML. Both of these attributes help define the types of data that the API endpoint supports, making interacting with the API more flexible and easier.

A basic complete structure in Swagger that includes all of the above can be exemplified in the following image.

```
1. openapi: 3.0.0
2. info:
3.   title: Sample API
4.   description: Optional multiline or single-line description in [CommonMark](http://commonmark.org/help/) or HTML.
5.   version: 0.1.9
6.
7. servers:
8.   - url: http://api.example.com/v1
9.     description: Optional server description, e.g. Main (production) server
10.    - url: http://staging-api.example.com
11.      description: Optional server description, e.g. Internal staging server for testing
12.
13. paths:
14.   /users:
15.     get:
16.       summary: Returns a list of users.
17.       description: Optional extended description in CommonMark or HTML.
18.       responses:
19.         '200': # status code
20.           description: A JSON array of user names
21.           content:
22.             application/json:
23.               schema:
24.                 type: array
25.                 items:
26.                   type: string
```

Figure 2. 7: Complete structure in swagger

2.4.2 Request body

The request body is commonly used when performing "create" and "update" actions (POST, PUT, PATCH) on resources in the API.

The request body allows for the use of a variety of data types such as JSON, XML, form data, plain text, and more, along with applying different **schemas** to each **media type**. The request body consists of a **content object**, an optional **Markdown-formatted** description, and a mandatory selectable flag (false by default). It lists the media types it supports (e.g., application/json) and assigns a corresponding schema to each media type. By default, the content of the Request body is optional. To specify that the Request body is required, you can use the "required" attribute with a value of true.

```
1. paths:
2.   /pets:
3.     post:
4.       summary: Add a new pet
5.
6.       requestBody:
7.         description: Optional description in *Markdown*
8.         required: true
9.         content:
10.           application/json:
11.             schema:
12.               $ref: '#/components/schemas/Pet'
13.           application/xml:
14.             schema:
15.               $ref: '#/components/schemas/Pet'
16.           application/x-www-form-urlencoded:
17.             schema:
18.               $ref: '#/components/schemas/PetForm'
19.           text/plain:
20.             schema:
21.               type: string
22.
23.       responses:
24.         '201':
25.           description: Created
```

Figure 2. 8: Example of a complete Request body in Swagger

2.4.3 Responses

An API with specifications that specify **responses** for all API **operations**. Each operation must have at least one response defined, usually a successful response. A response is identified by an **HTTP** status code and the data is returned in the **response body** and/or **headers**. The following image is a basic example of a Response in Swagger.

```
1. paths:
2.   /ping:
3.     get:
4.       responses:
5.         '200':
6.           description: OK
7.           content:
8.             text/plain:
9.               schema:
10.                 type: string
11.                 example: pong
```

Figure 2. 9: Examples of Response in Swagger

Response Media Types

APIs can return data in a variety of formats. While JSON is the most popular format, it's not unique. To determine the type of data format returned, we use the keyword "Content" at the API activity level.

```
1. paths:
2.   /users:
3.     get:
4.       summary: Get all users
5.       responses:
6.         '200':
7.           description: A list of users
8.           content:
9.             application/json:
10.               schema:
11.                 $ref: '#/components/schemas/ArrayOfUsers'
12.             application/xml:
13.               schema:
14.                 $ref: '#/components/schemas/ArrayOfUsers'
15.             text/plain:
16.               schema:
17.                 type: string
18.
19. # This operation returns image
20. /logo:
21.   get:
22.     summary: Get the logo image
23.     responses:
24.       '200':
25.         description: Logo image in PNG format
26.         content:
27.           image/png:
28.             schema:
29.               type: string
30.             format: binary
```

Figure 2. 10: API response data with other formats

HTTP Status Codes

In the Response section of the API, each response definition starts with a **Status Code** such as 200 or 404. An operation will typically return one **successful Status Codes** and one or more **error Status Codes**. To define the scope of the response codes, you can use range definitions such as 1XX, 2XX, 3XX, 4XX, and 5XX. If a response scope is defined by specific code, the definition for that code takes precedence over the general scope definition. Each **Status Codes** Response requires a description, such as a description of the condition that caused the error. Markdown (CommonMark) can be used to format text in a flexible and varied way.

```
1.   responses:
2.     '200':
3.       description: OK
4.     '400':
5.       description: Bad request. User ID must be an integer and larger than 0.
6.     '401':
7.       description: Authorization information is missing or invalid.
8.     '404':
9.       description: A user with the specified ID was not found.
10.    '5XX':
11.      description: Unexpected error.
```

Figure 2. 11: Types of Status Codes and their corresponding descriptions

CHAPTER 3: BUILDING A CHATBOT

3.1. Analyze stock data from the AI Stocktraders system using Python

3.1.1. API Connection

First, I connect the API from the AI Stocktraders system in python language, through the request function and the post method, then we check the response status and print out the returned data.

```
url = "https://stocktraders.vn/service/data/getStockWave"
data = {"StockWaveRequest": {"account": "StockTraders"}}
response = requests.post(url, json = data)
print(response.status_code)
data_received = response.json()
data_received

201

{'StockWaveRequest': {'codeReply': {'codeID': 'S0000', 'codeName': 'SUCCESS'},
 'stockWaves': {'name': 'ALL',
 'waveDatas': [{'buy': 2,
 'date': '2020-02-03',
 'sell': 22,
 'waitbuy': 68,
 'wailsell': 18},
 {'buy': 15,
 'date': '2020-02-04',
 'sell': 13,
 'waitbuy': 94,
 'wailsell': 15},
 {'buy': 21,
 'date': '2020-02-05',
 'sell': 9,
 'waitbuy': 108,
 'wailsell': 11},
 {'buy': 60,
 'date': '2020-02-06',
 'sell': 1,}
```

Figure 3. 1: API Connection in Python

3.1.2. Data analysis and visualization

The data we receive from the API is in json form, in order to proceed with data analysis and visualization, we need to extract the data returned from the API to get the necessary columns including: buy, date, sell, waitbuy, waitsell and combine the columns into a dataframe.

```
# Extract the waveDatas from the response
```

```
wave_datas = data_received['StockWaveRequest']['stockWaves']['waveDatas']
```

```
# Convert the waveDatas to a pandas DataFrame
```

```
df = pd.DataFrame(wave_datas)
```

	buy	date	sell	waitbuy	waitsell
0	2	2020-02-03	22	68	18
1	15	2020-02-04	13	94	15
2	21	2020-02-05	9	108	11
3	60	2020-02-06	1	65	11
4	17	2020-02-07	2	52	30
...
1002	2	2024-01-29	7	17	40
1003	2	2024-01-30	8	26	43
1004	4	2024-01-31	16	22	41
1005	3	2024-02-01	4	38	41
1006	5	2024-02-02	3	32	42

1007 rows × 5 columns

Figure 3. 2: Collect and extract data into Dataframes

Below is a statistical table describing the variables in the dataset, except for the date variable under each other variable, I add a box plot table to show the dispersion of the data, to the right of the box plot is the number of outliers.

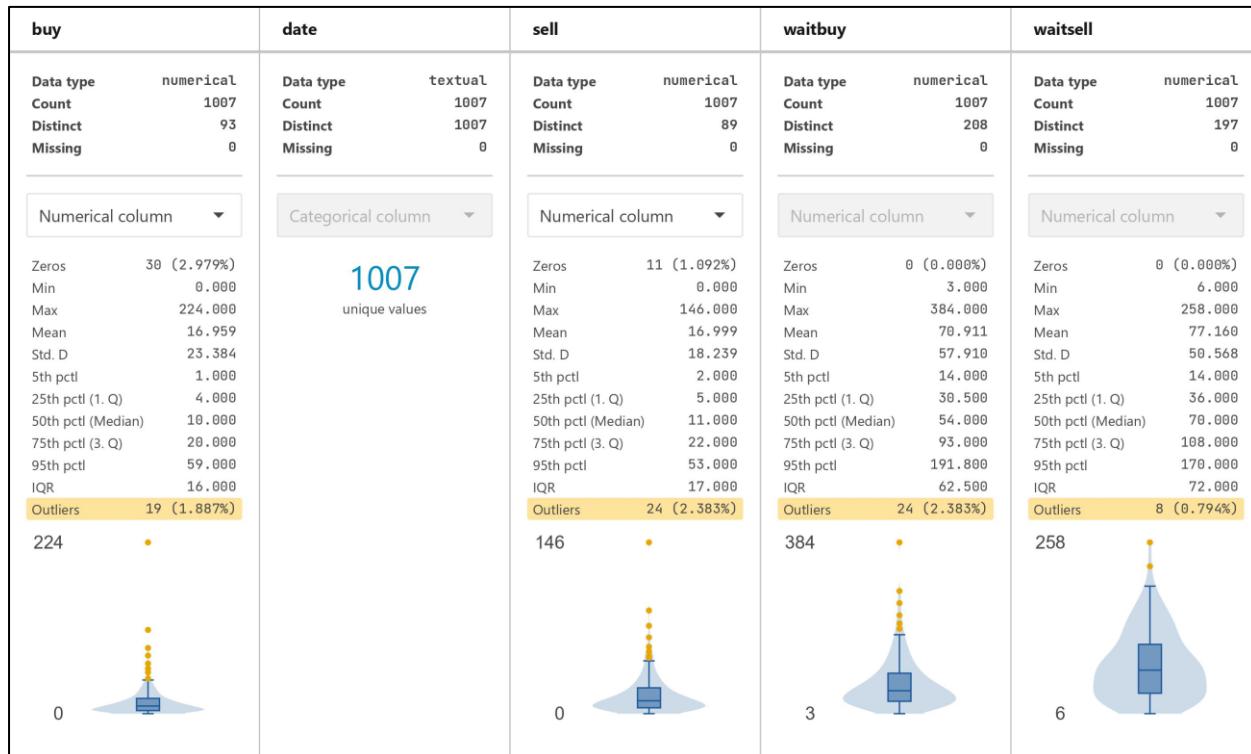


Figure 3. 3: Statistics describing the dataset

Because the data is collected over time, you will create line charts to better understand trends and patterns that move over time. Not only does this chart provide a visual and seamless view of the data, but it also allows us to capture important fluctuations and explore the connections between events over time.

```

# Visualize
fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(16, 10))
sns.set_style('whitegrid')
plt.style.use('seaborn-v0_8')

# set parameters for variable histogram graph "close"
buy = df['buy']
sell = df['sell']
waitbuy = df['waitbuy']
waitsell = df['waitsell']
dates = df.index

ax[0,0].plot(dates, buy)
ax[0,1].plot(dates, sell)
ax[1,0].plot(dates, waitbuy)
ax[1,1].plot(dates, waitsell)
# ax[0,0].plot(dates, waitsell)

ax[0,0].set_title("Dữ liệu buy theo ngày")
ax[0,0].set_ylabel('buy')

ax[0,1].set_title("Dữ liệu sell theo ngày")
ax[0,1].set_ylabel('sell')

ax[1,0].set_title("Dữ liệu waitbuy theo ngày")
ax[1,0].set_ylabel('waitbuy')
ax[1,0].set_xlabel('date')

ax[1,1].set_title("Dữ liệu waitsell theo ngày")
ax[1,1].set_ylabel('waitsell')
ax[1,1].set_xlabel('date')

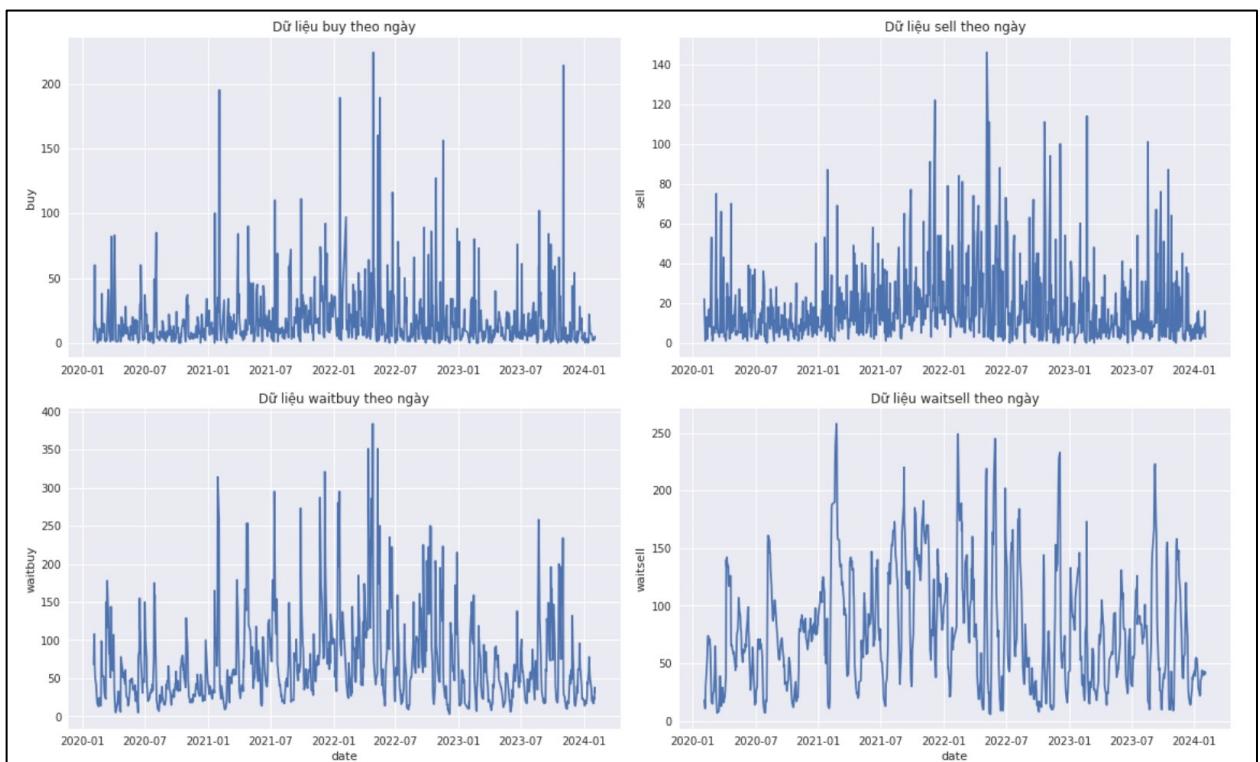
plt.tight_layout()
plt.show()

```

Figure 3. 4: Set the parameters for the line charts

Figure 3. 5: Visualize "buy", "sell", "waitbuy", "waitsell" data by date

Next, we use the Pairlot chart to find the relationships between data variables, where the pairlot chart resembles us recognizes variables that can be continuous or categorical.



```

sns.set()
sns.pairplot(df, height = 2.5)
plt.show()

```

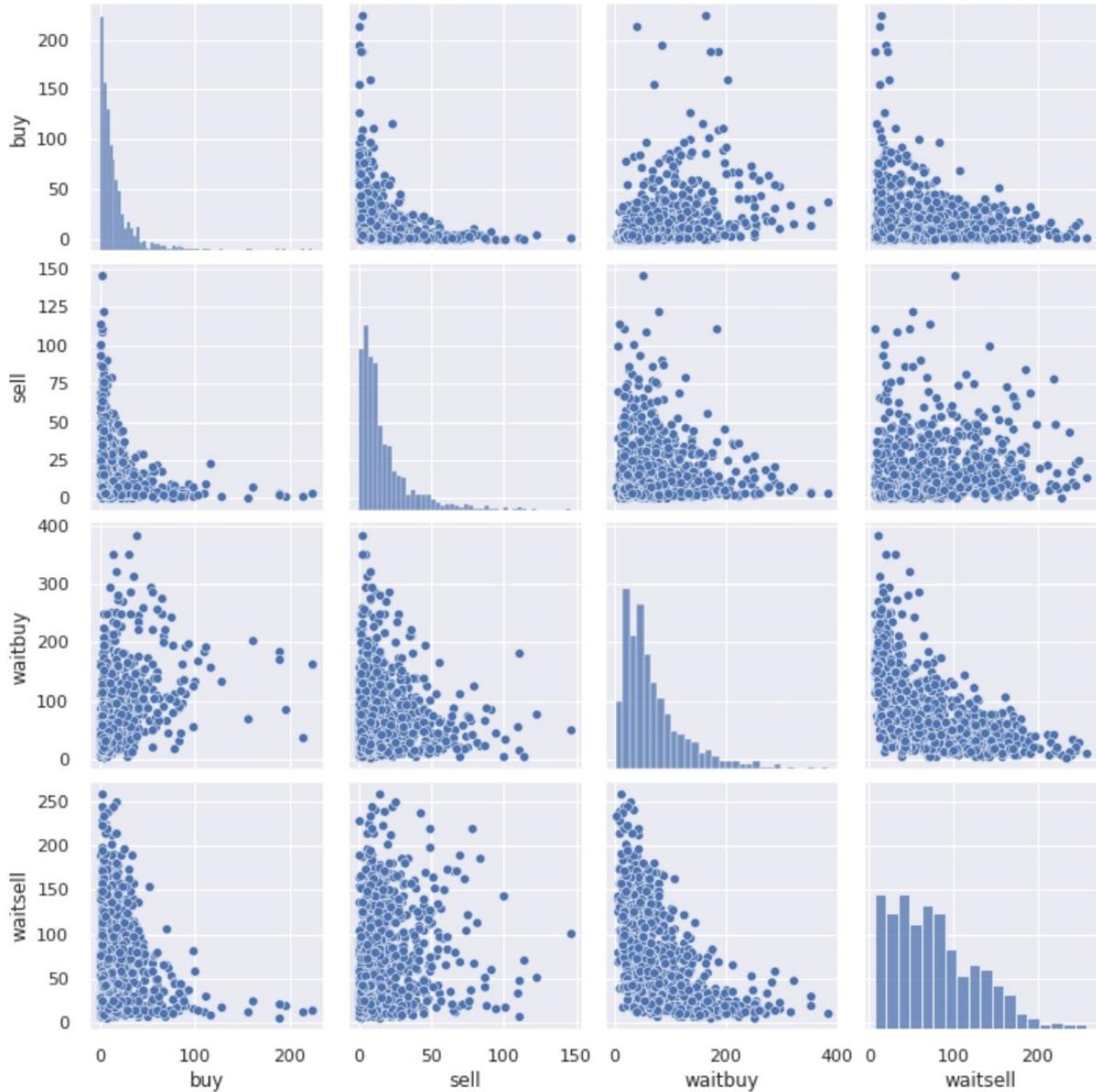


Figure 3. 6: Pairplot Chart

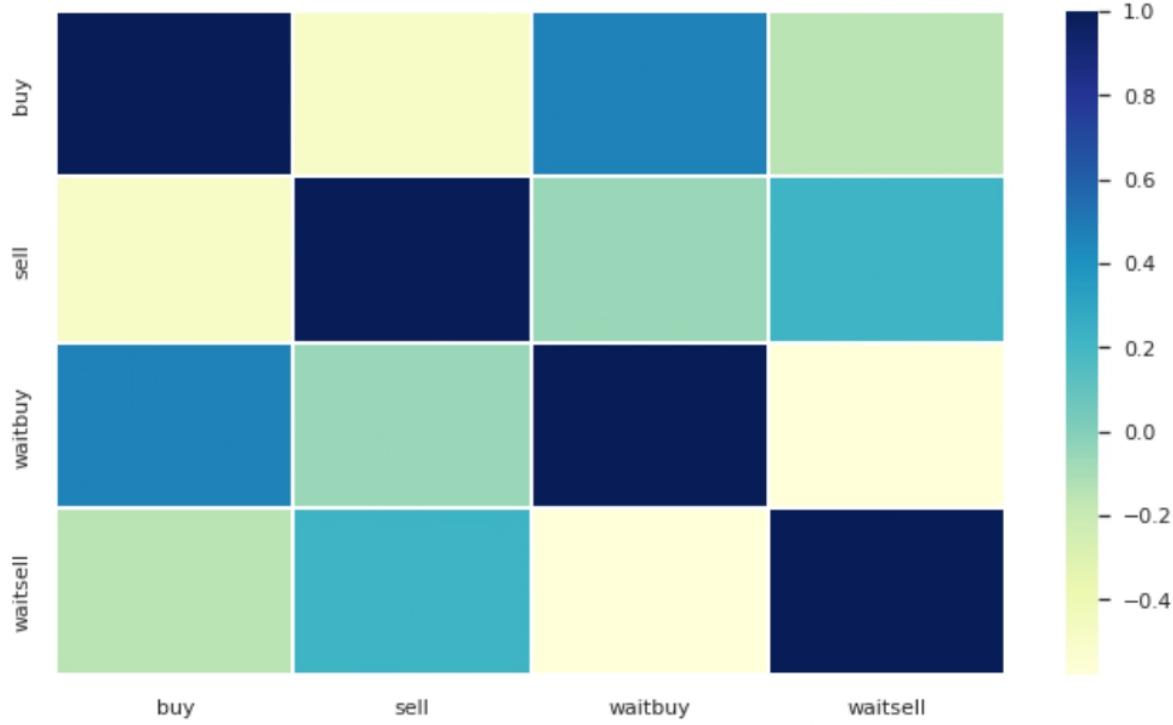
Finally, we use the correlation matrix to determine the correlation between the variables. Looking at the chart below, it can be seen that the variables "buy" and "waitbuy", have a close correlation of 1, which means that when the waitbuy data increases, the probability of the buy data will also increase. There are 2 pairs of variables that are closely correlated to -1, namely "buy" and "sell", "waitbuy" and "watisell" which means that when the buy and watibuy data increases, the "sell" and "waitsell" data decreases and vice versa.

```

corrrmat = df.corr(method='spearman')
f, ax = plt.subplots(figsize=(10, 6))
sns.heatmap(corrrmat, ax=ax, cmap="YlGnBu", linewidths=0.1)

```

<Axes: >



Hình 3. 7: Correlation matrix

3.2. Create a Chatbot about stocks in Chat GPT

3.2.1. Build API connection statements

To connect to the Chat GPT API provided by OpenAI, the first thing we need to do is prepare a set of commands that are in accordance with the regulations set out in OpenAI's Swagger document, because this document describes the standard structure and application programming interfaces (APIs) that OpenAI supports. Swagger is a powerful API design tool that helps define standard API specifications, as well as create documentation and framework code to interact with APIs easily and efficiently.

Based on the concepts introduced in the Swagger documentation, we need to build a well-structured API code, including components such as metadata, base URLs, and input (Consumes) and output (Produces) data formats. To make the source code easier to read and manage, we use \$ref referencing mechanisms to reuse definitions. Specifically, \$ref help us reference the previously defined JSON snippets: "StockWaveRequest" for the request to the API, and "StockWaveResponse" for the response from the API. Using \$ref

not only reduces code repetition, but also makes the source code structure neater and easier to maintain.

```

1  openapi: 3.0.0
2  info:
3    title: StockTraders API
4    description: API for fetching stock wave data from StockTraders.vn.
5    version: 1.0.0
6  servers:
7    # Added by API Auto Mocking Plugin
8    - description: SwaggerHub API Auto Mocking
9    url: https://virtserver.swaggerhub.com/NHATNHMD/nhatnhat/1.0.0
10   - url: https://stocktraders.vn/service/data
11     description: StockTraders API Server
12 paths:
13   /getStockWave:
14     post:
15       operationId: postStockWave
16       summary: Fetches stock wave data.
17       requestBody:
18         required: true
19         content:
20           application/json:
21             schema:
22               $ref: "#/components/schemas/StockWaveRequest"
23       responses:
24         "201":
25           description: Successful response with stock wave data
26         content:
27           application/json:
28             schema:
29               $ref: "#/components/schemas/StockWaveResponse"
30         "400":
31           description: Bad request
32         "500":
33           description: Server error
34

```

Figure 3. 8: Connecting the API in Swagger

From the Json snippet sent to the structured API: {"StockWaveRequest": {"account": "StockTraders"} } we will rewrite in Swagger as follows

```

36  components:
37    schemas:
38      StockWaveRequest:
39        type: object
40        properties:
41          StockWaveRequest:
42            type: object
43            required:
44              - account
45            properties:
46              account:
47                type: string
48                enum:
49                  - StockTraders

```

Hình 3. 9: Schemas request

Similarly, the API return Json snippet I collected from python (not all) has the structure:

```
{"StockSignalReply": {"codeReply": {"codeID": "S0000", "codeName": "SUCCESS"}, "stockSignals": [{"signalDatas": [{"date": "2021-10-14", "percent": 20, "price": 23.7, "trade": 1}]}]}
```

we'll rewrite in Swagger as follows:

```

51:   StockWaveResponse:
52:     type: object
53:     required:
54:       - StockWaveRequest
55:     properties:
56:       StockWaveRequest:
57:         type: object
58:         required:
59:           - codeReply
60:           - stockWaves
61:         properties:
62:           codeReply:
63:             type: object
64:             required:
65:               - codeID
66:               - codeName
67:             properties:
68:               codeID:
69:                 type: string
70:                 enum:
71:                   - S0000
72:               codeName:
73:                 type: string
74:                 enum:
75:                   - SUCCESS
76:             stockWaves:
77:               type: object
78:               required:
79:                 - name
80:                 - waveDatas
81:               properties:
82:                 name:
83:                   type: string
84:                   enum:
85:                     - ALL
86:                 waveDatas:
87:                   type: array
88:                   items:
89:                     type: object
90:                     properties:
91:                       buy:
92:                         type: integer
93:                         date:
94:                           type: string
95:                           format: date
96:                         sell:
97:                           type: integer
98:                         waitbuy:
99:                           type: integer
100:                        waitsell:
101:                          type: integer
102:

```

Hình 3. 10: *Schemas response*

After building the code to connect the API in Swagger, we will proceed to execute it to check whether the correct data is received or not

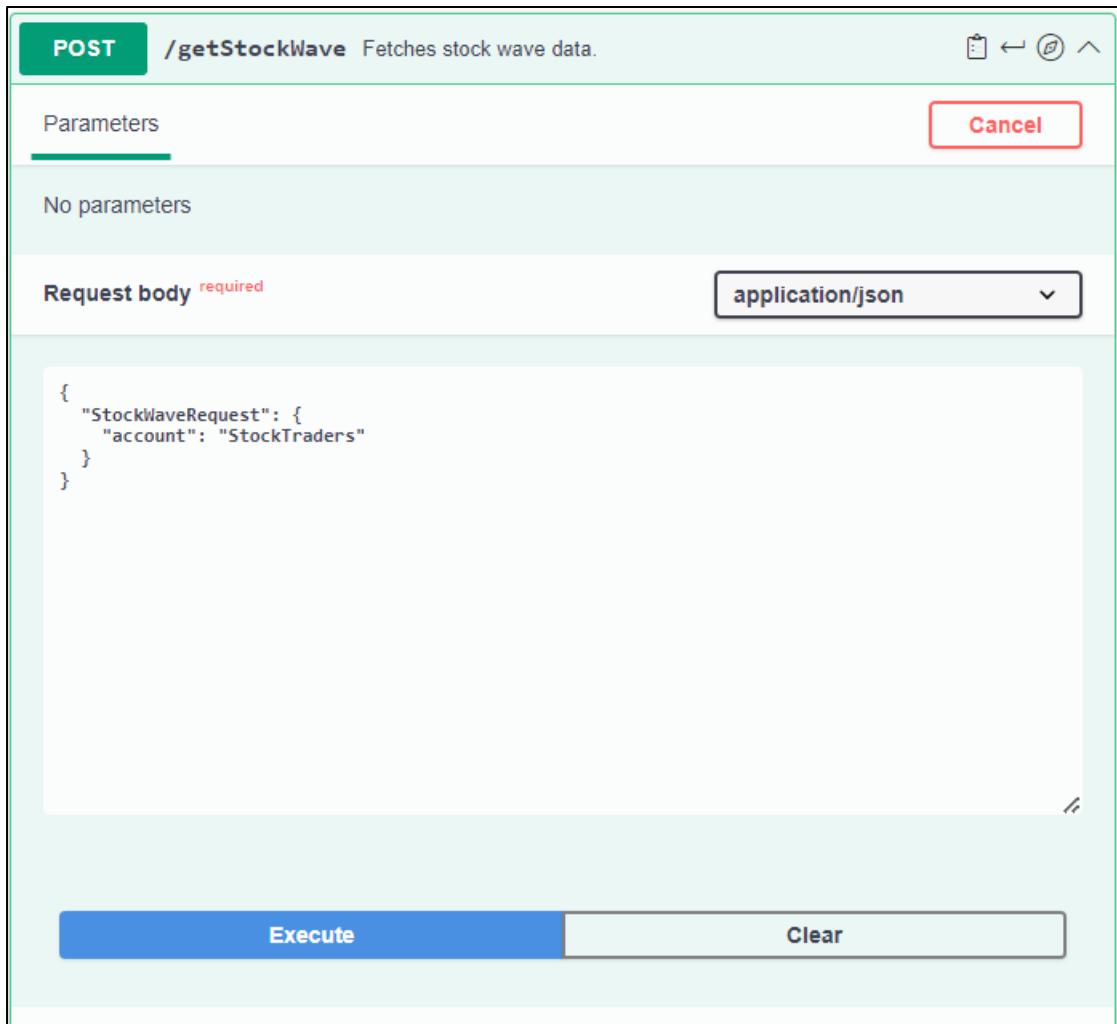


Figure 3. 11: Enforce API Connection in Swagger

The result of the received data matches the data we retrieved in python, along with the format of the returned data also matches the response schema we built earlier. The data we received matches the data that was retrieved in Python. In addition, the format of the returned data also conforms to the response schema that we have established beforehand.

Responses

Snippets ^

Request URL

```
https://stocktraders.vn/service/data/getStockWave
```

Server response

Code	Details
201	Response body
	<pre>{ "StockWaveRequest": { "codeReply": { "codeID": "50000", "codeName": "SUCCESS" }, "stockWaves": { "name": "ALL", "waveDatas": [{ "buy": 2, "date": "2020-02-03", "sell": 22, "waitbuy": 68, "waitsell": 18 }, { "buy": 15, "date": "2020-02-04", "sell": 13, "waitbuy": 94, "wailsell": 15 }] } } }</pre> <p style="text-align: right;"> </p>

Figure 3. 12: Data returned after executing a connection in the API

3.2.2. Configuration settings for Chatbots in ChatGPT

After completing the API connection in Swagger, we will proceed to create a separate chatbot, first we enter "Create".

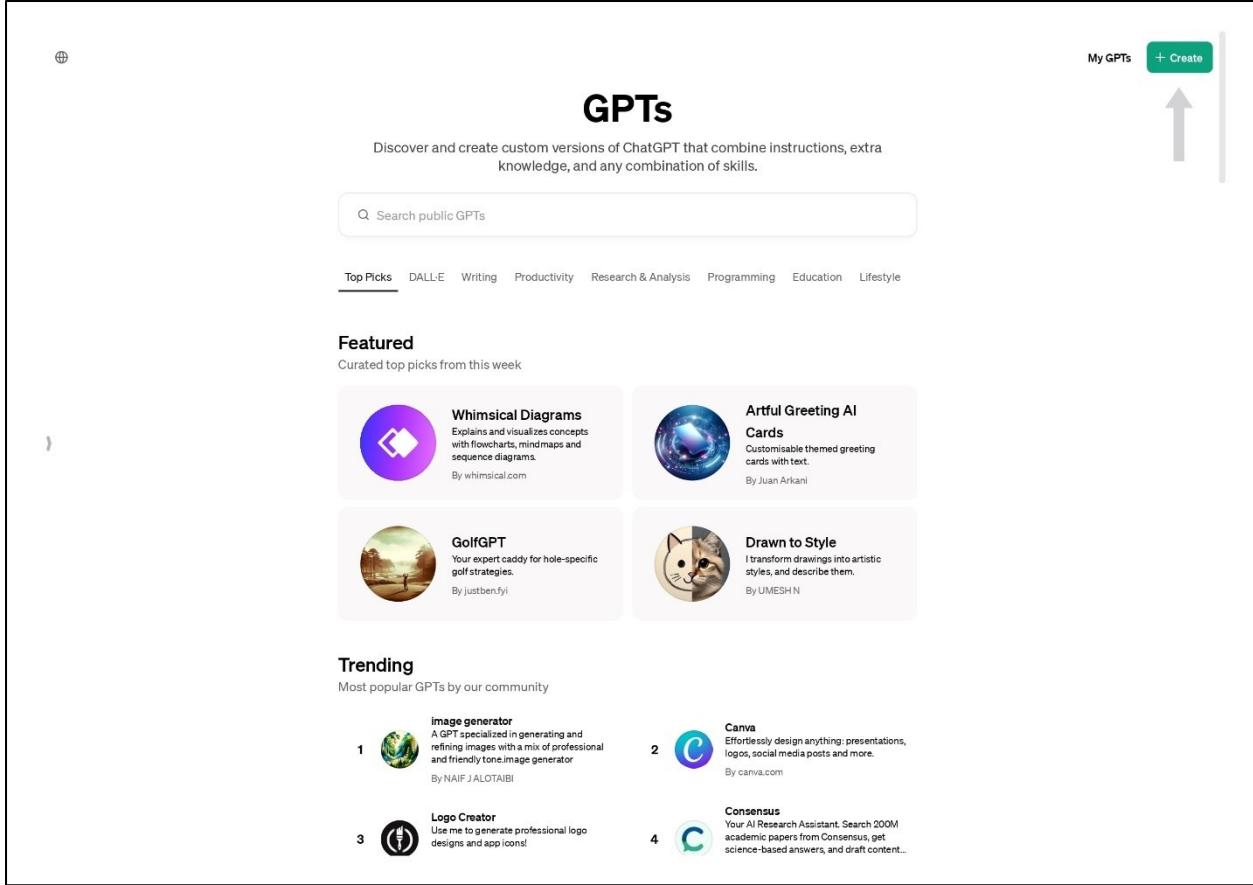


Figure 3. 13: The main screen creates a chatbot in ChatGPT

Next, we proceed to fine-tune the configurations to suit the Chatbot that we have built. In this topic, I implemented a Chat bot about stocks, so I will refine the appropriate configurations for securities

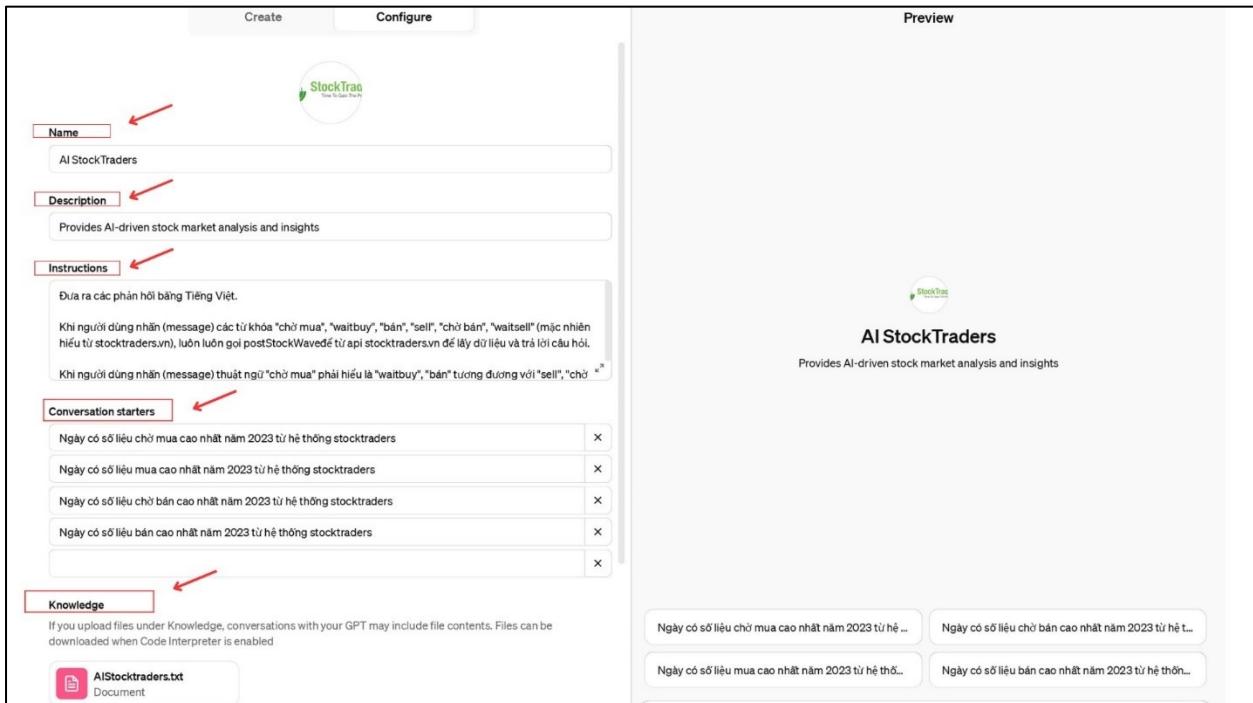


Figure 3. 14: Fine-tune the configuration in the chatbot

Next, we upload the documents of the AI StockTraders system to provide context for the chatbot to learn. At the same time, we proceed to create a new Action, Action to help the Chatbot get data from the API provided by the AI StockTraders system.

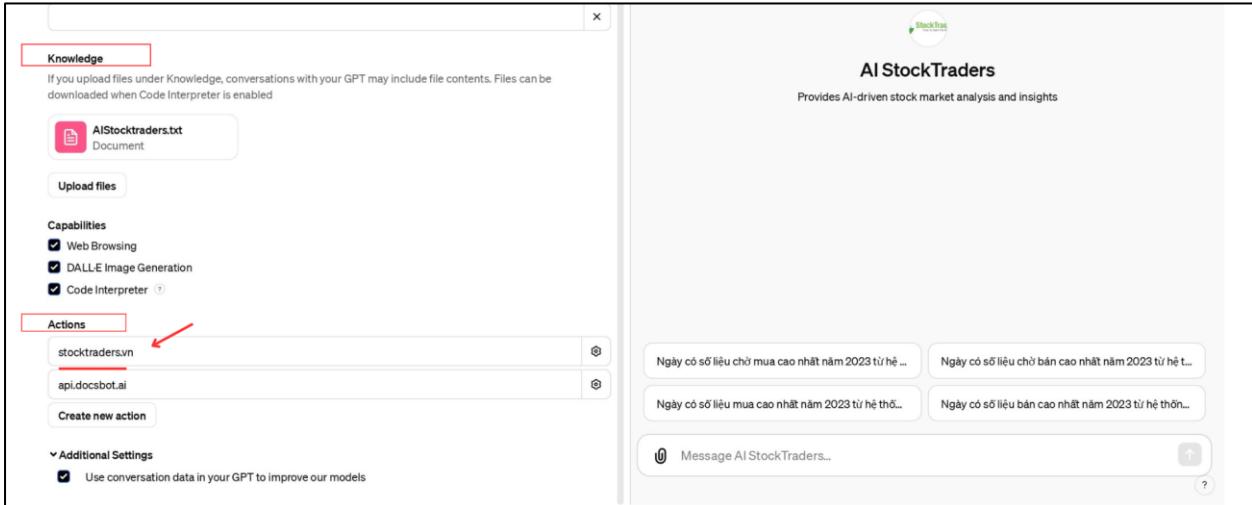


Figure 3. 15: Add files and create options for chatbots

Finally, we'll insert the code to connect to the API we've created in Swagger into the Schema section. Once this is done, the methods that we have developed will be displayed in the Available Actions section, allowing us to conduct testing.

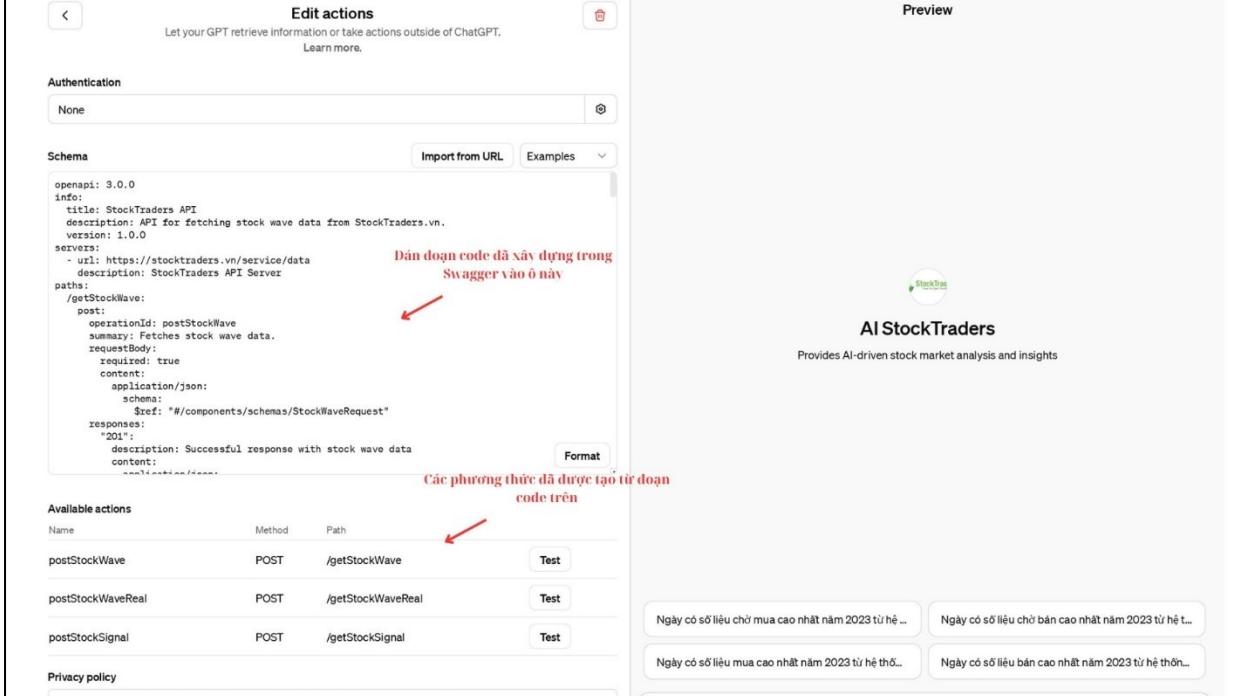


Figure 3. 16: Create an option by pasting the API connection statement in Swagger

Finally, we'll test whether the Chatbot is capable of pulling data from the API. The test results show that the Chatbot has successfully connected to the API and the correct data has been returned from the API.

Unpublished changes ... Update ▾

Preview

You
Call the stocktraders.vn API with the postStockWave operation

AI StockTraders

```
> [debug] Calling HTTP endpoint
> [debug] Calling HTTP endpoint
▽ [debug] Response received
{
  "response_data": {
    "StockWaveRequest": {
      "codeReply": {
        "codeID": "S0000",
        "codeName": "SUCESS"
      },
      "stockWaves": {
        "name": "ALL",
        "waveDatas": [
          {
            "buy": 2,
            "date": "2020-02-03",
            "sell": 22,
            "waitbuy": 68,
            "wailsell": 18
          },
          {
            "buy": 15,
            "date": "2020-02-04",
            "sell": 13,
            "waitbuy": 94,
            "wailsell": 15
          },
          {
            "buy": 21,
            "date": "2020-02-05",
            "sell": 9,
            "waitbuy": 108,
            "wailsell": 11
          },
          {
            "buy": 60,
          }
        ]
      }
    }
  }
}
```

Message AI StockTraders... ?

Figure 3. 17: Testing the Post Method

CHAPTER 4: ANALYSIS AND EVALUATION OF RESULTS

4.1. Chatbot Analytics and Reviews

We'll test the Chatbot by asking questions and seeing its responses. The answer from the Chatbot shows that it has succeeded in connecting to the API through "Talked to stocktraders.vn" and that the data received is accurate compared to the data from the API.

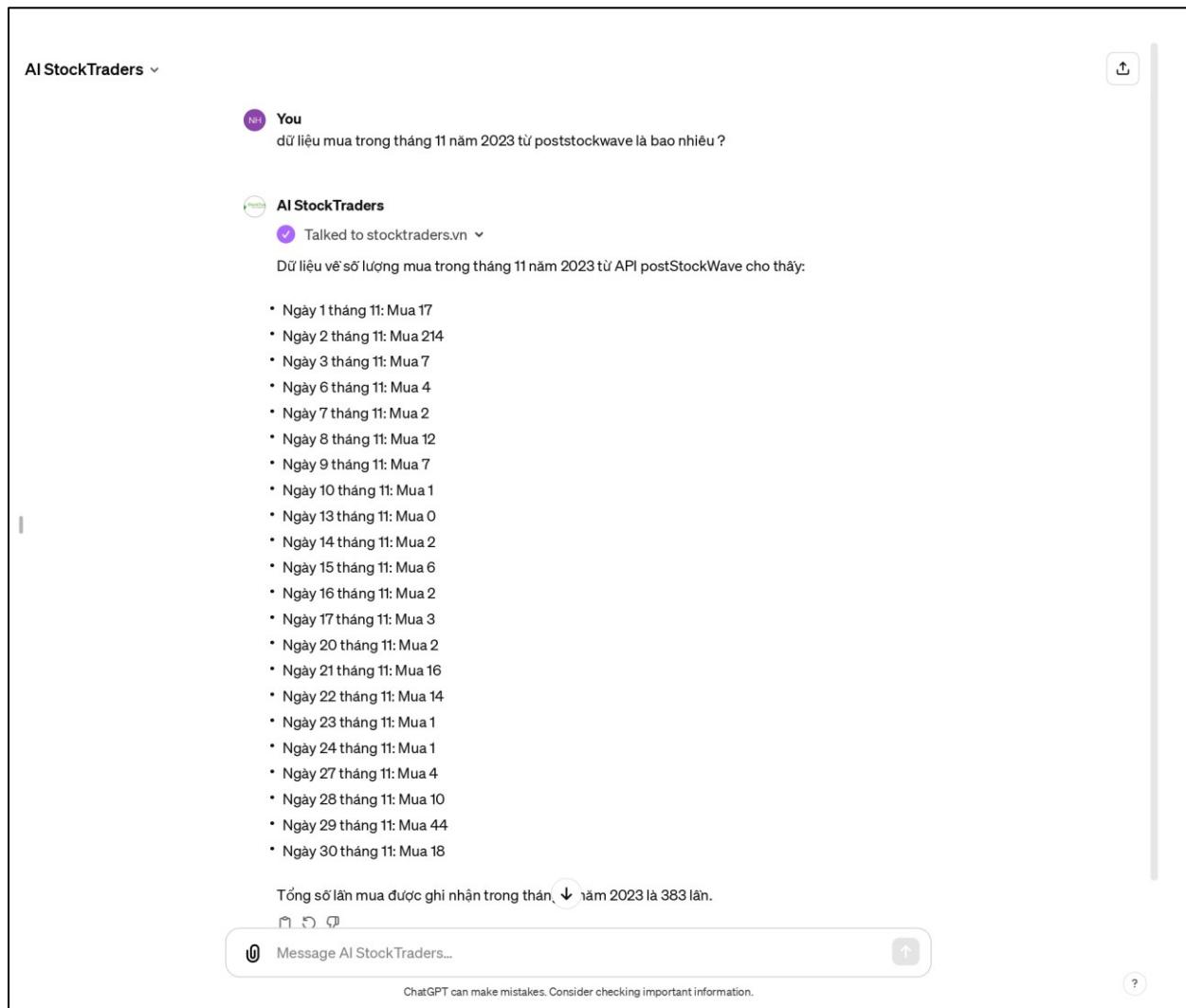


Figure 4. 1: Chatbot Response

CHAPTER 5: DEVELOPMENT DIRECTION

In this project, I have created a chatbot about stocks based on data from the StockTraders system by connecting the API from the system to the chatbot. In addition, chatbots are fine-tuned parameters to suit business needs, such as chatbots learning in-depth documents from businesses along with instructions for chatbots to retrieve in accordance with the system's intent.

In the future, chatbots can be developed and further completed with the following functions:

- Adding actions for users can ask the chatbot to refer to the AI StockTraders system to trade stocks right on the chatbot interface.
- Adding materials on market psychology and trading psychology helps the chatbot make more relevant recommendations.

REFERENCES

- [1] "Introduction - State Securities Commission." Accessed: Feb. 14, 2024. [Online]. Available: https://ssc.gov.vn/webcenter/portal/ubck/pages_r/m/giithiu
- [2] "Basic Structure." Accessed: Feb. 17, 2024. [Online]. Available: <https://swagger.io/docs/specification/basic-structure/>
- [3] "Describing Request Body." Accessed: Feb. 18, 2024. [Online]. Available: <https://swagger.io/docs/specification/describing-request-body/>
- [4] "Describing Responses." Accessed: Feb. 18, 2024. [Online]. Available: <https://swagger.io/docs/specification/describing-responses/>