Capstone Project Phase A

24-2-D-26

****

Software Engineering Department  
Braude College

**Supervisor:**

Alexander Keselman

**Authors:**

Chay Fadida 318302783

Tomer Lazarovitch 316377100

****

Table of Contents

[1. Abstract 4](#_Toc177723511)

[2. Introduction 4](#_Toc177723512)

[3. Background: 5](#_Toc177723513)

[3.1 Stock Market Data API 5](#_Toc177723514)

[3.2 LLM Model (Large Language Model) 5](#_Toc177723515)

[3.3 Crew AI 6](#_Toc177723516)

[3.4 FastAPI 6](#_Toc177723517)

[3.5 React 7](#_Toc177723518)

[3.6 MongoDB 8](#_Toc177723519)

[3.7 CSS 8](#_Toc177723520)

[4. Expected Achievements: 8](#_Toc177723521)

[4.1 Achievements 8](#_Toc177723522)

[4.1.1 AI-Powered Stock Investment Recommendations 9](#_Toc177723523)

[4.1.2 Interactive AI Chatbot 9](#_Toc177723524)

[4.1.3 Integration of Fundamental and Technical Analysis 9](#_Toc177723525)

[4.1.4 Real-Time Financial Data Integration 9](#_Toc177723526)

[4.1.5 Customizable Investment Strategies 9](#_Toc177723527)

[4.1.6 Portfolio Management Tools 9](#_Toc177723528)

[4.1.7 Scalable and Secure Backend 9](#_Toc177723529)

[4.2 Criteria for Success 10](#_Toc177723530)

[4.2.1 Functionality 10](#_Toc177723531)

[4.2.2 Performance 10](#_Toc177723532)

[4.2.3 User Interaction 10](#_Toc177723533)

[4.2.4 Scalability 10](#_Toc177723534)

[4.2.5 User Satisfaction 10](#_Toc177723535)

[5. Engineering Process 10](#_Toc177723536)

[5.1 Process 11](#_Toc177723537)

[5.1.1 The Research 11](#_Toc177723538)

[5.1.2 User Personas and Their Needs 11](#_Toc177723539)

[5.1.3 Conclusions from Research 14](#_Toc177723540)

[5.1.4 Methodology and Development Process 14](#_Toc177723541)

[5.2 Product 15](#_Toc177723542)

[5.2.1 Requirements 16](#_Toc177723543)

[5.2.3 Diagrams 17](#_Toc177723544)

[5.2.4 Deployment 20](#_Toc177723545)

[5.2.5 Future Development and Feature Expansion 20](#_Toc177723546)

[6. Evaluation and Verification 21](#_Toc177723547)

[6.1 Verification 21](#_Toc177723548)

[22](#_Toc177723549)

[6.2 Evaluation 22](#_Toc177723550)

[7. AI prompts 23](#_Toc177723551)

[8. References 23](#_Toc177723552)

# 1. Abstract

In today's financial landscape, the stock market remains a lucrative but complex arena for investment, particularly for individuals without financial expertise.

The primary challenge for these individuals is the high cost and often inaccessible nature of professional financial advice, coupled with the overwhelming array of investment choices.

Our project seeks to address these barriers by developing an innovative web application that combines artificial intelligence with advanced analytics to offer personalized stock investment advice.

The proposed AI financial advisor will leverage two distinct analytical approaches:

the evaluation of quarterly financial reports and the historical performance analysis of individual stocks.

By integrating these methodologies through a sophisticated AI framework, our system aims to democratize stock market investments, making them more accessible to non-experts and reducing the reliance on costly financial consultants.

This tool will not only simplify the decision-making process but also enhance users' confidence in managing their investments independently.

# 2. Introduction

Today’s market primarily includes bank-affiliated financial advisors, various independent financial websites, investment courses, and resources for self-learning.

Each of these solutions has its limitations: bank advisors often prioritize the bank's interests, independent websites vary significantly in reliability, investment courses can be costly and time-consuming, and self-learning demands a high level of discipline and prior knowledge.

In contrast, our project introduces several innovative features that significantly enhance user experience and decision-making efficiency in stock investments.

Firstly, our AI-driven approach allows for a high degree of customization, tuning advice[[1]](#footnote-1) to incorporate a blend of different analytical schools such as technical analysis and fundamental analysis based on user preferences. This creates a tailored investment strategy for each user, recognizing the unique aspects of their financial goals and risk tolerance.

Our platform will also feature a highly interactive AI chatbot that not only provides advice but engages users in a meaningful dialogue about their investments, offering real-time, data-driven recommendations on whether to buy, hold, or sell stocks[[2]](#footnote-2). This chatbot will utilize advanced predictive analytics, combining historical data trends and real-time market sentiment to forecast stock performances more accurately than traditional models.

# 3. Background:

This section outlines the key technologies and concepts utilized in the development of our web application, highlighting how they contribute to the functionality, scalability, and user experience of the platform.

## 3.1 Stock Market Data API

Integrating with leading stock market data APIs provides our users with real-time information on stock prices, financial reports, and market trends. We partner with reputable financial data providers to ensure that our users receive accurate and up-to-date information, crucial for making informed investment decisions. This seamless integration allows users to access comprehensive financial data directly through our platform, enabling them to react swiftly to market changes and optimize their investment strategies.

## 3.2 LLM Model (Large Language Model)

Our platform harnesses the power of advanced Large Language Models (LLMs)[[3]](#footnote-3) to enhance the financial advisory services we offer. LLMs are a type of artificial intelligence designed to understand and generate human-like text based on the input they receive. This capability allows our AI-driven financial advisor to process complex financial data, interpret user queries, and deliver insightful, conversational responses.

The integration of LLMs into our platform enables several key functionalities:

1. **Natural Language Understanding**: Our LLMs can understand queries posed in everyday language, allowing users to interact with our system as they would with a human advisor. This makes financial advice accessible to users without financial jargon expertise, simplifying the investment process.

2. **Personalized Advice Generation**: By analyzing users’ input and financial data, the LLM provides customized advice that aligns with individual investment goals and risk tolerances. This personalized interaction helps users make better-informed decisions tailored to their specific circumstances.

3. **Continuous Learning**: LLMs improve over time through machine learning techniques, adapting to new financial trends, user feedback, and changes in market conditions. This continuous learning ensures that our financial advice remains relevant and accurate, enhancing user trust and engagement.

4. **Document Analysis and Integration**: LLMs possess the ability to receive, read, and analyze financial documents such as spreadsheets, PDFs, and other formats. This functionality allows users to upload their financial records directly to the platform, where the LLM

extracts and integrates relevant data into their existing portfolios.

This enhances the user experience by providing a comprehensive view of investments, updated in real-time and further enriches the personalized advice offered.

## 3.3 Crew AI

A framework for collaborative AI agents[[4]](#footnote-4), can be a powerful asset in our stock analysis investment advisor project. It allows us to build a "crew" of specialized agents, each tackling a specific aspect of stock analysis. One agent could gather financial data and news, another could analyze trends and identify patterns, while a third could translate this information into clear, actionable insights.

CrewAI's collaborative nature ensures these agents work together seamlessly, providing a more comprehensive picture compared to a singular AI model. This can empower our investment advisor to deliver well-rounded analysis and suggest investment strategies that consider various financial factors.

## 3.4 FastAPI

FastAPI[[5]](#footnote-5) is a modern, high-performance web framework for Python, specifically designed for building APIs quickly and efficiently. It is known for its ease of use, scalability, and the ability to handle asynchronous requests, making it ideal for developing data-intensive applications. FastAPI also provides automatic interactive API documentation with OpenAPI and JSON Schema, which greatly improves the development experience.

In our project, FastAPI’s efficiency and asynchronous capabilities are particularly advantageous. It enables the rapid creation of RESTful APIs, which are critical for seamless communication between the frontend and backend of our platform. The framework’s support for asynchronous code is crucial for managing large volumes of data, user queries, and the real-time integration of machine learning models for financial analysis. Moreover, FastAPI’s performance surpasses that of many other frameworks, ensuring that our application remains responsive and scalable as user demand grows.

By choosing FastAPI, we ensure that our platform remains robust, scalable, and capable of handling real-time interactions, which is essential for delivering accurate and timely financial advice to users. Its developer-friendly features also streamline the integration of machine learning components and data flow management, optimizing both the user experience and backend performance.

## 3.5 React

React[[6]](#footnote-6) is a popular JavaScript library used for building user interfaces.  
It allows developers to create reusable UI components, ensuring efficient rendering and a dynamic user experience. React operates on a component-based architecture, which makes the development process more organized and easier to maintain as the application scales. Its virtual DOM feature ensures that the interface updates efficiently without reloading the entire page, providing a seamless user experience.

In our project, React plays a crucial role in building the frontend of the AI-driven financial advisor platform. Its flexibility and efficiency allow us to create interactive and responsive user interfaces.

By leveraging React, we ensure that our platform provides an intuitive and user-friendly experience, where users can interact with the system fluidly and access key financial insights in real-time. The library’s scalability also allows us to easily expand the application’s features as user needs evolve.

## 3.6 MongoDB

MongoDB [[7]](#footnote-7) is a NoSQL database known for its flexibility and scalability. It stores data in a JSON-like format (key-value), making it highly adaptable to handle complex and unstructured data. MongoDB’s schema-less nature allows for rapid development and easy updates as project requirements evolve.

In our project, MongoDB is essential for storing and managing user data, financial information, and investment records. Its ability to handle large datasets efficiently ensures that the platform remains responsive, even as data grows.

## 3.7 CSS

CSS is used for styling and layout in web applications, ensuring a visually appealing and

responsive user interface. By separating design from content, CSS enables developers to control

the look and feel of the application with ease.

In our project, CSS is crucial for creating a clean, intuitive, and responsive interface. It ensures

that our financial dashboard, forms, and data visualizations are displayed consistently across

different devices, providing users with a seamless experience.

# 4. Expected Achievements:

In our project, we aim to develop an AI-driven web application that provides personalized financial advice, particularly focusing on stock investments.

## 4.1 Achievements

The main achievements of the project will include:

### 4.1.1 AI-Powered Stock Investment Recommendations

Develop a machine learning algorithm that analyzes users’ financial data and provides

personalized investment recommendations based on individual risk tolerance and financial

goals.

### 4.1.2 Interactive AI Chatbot

Implement a chatbot that offers real-time, data-driven advice and engages users in

personalized financial consultations, answering queries about stock performance and

investment strategies.

### 4.1.3 Integration of Fundamental and Technical Analysis

Combine quarterly financial reports (fundamental analysis) with historical stock performance

(technical analysis) to offer a holistic investment advisory system.

### 4.1.4 Real-Time Financial Data Integration

Integrate external financial data APIs to provide users with real-time stock prices, market

trends, and news, ensuring up-to-date investment advice.

### 4.1.5 Customizable Investment Strategies

Allow users to customize their investment approach by adjusting the weighting between fundamental and technical analysis, tailoring strategies to personal preferences.

### 4.1.6 Portfolio Management Tools

Develop features for tracking and managing investment portfolios, including data visualizations,

performance metrics, and rebalancing suggestions.

### 4.1.7 Scalable and Secure Backend

Implement a secure backend infrastructure using FastAPI, ensuring scalability to accommodate a growing user base and real-time data processing.

## 4.2 Criteria for Success

### 4.2.1 Functionality

* The system must successfully generate personalized financial advice for users based on their input data and selected preferences (e.g., risk tolerance).
* The AI model should provide accurate stock recommendations, with a focus on the timely delivery of buy, hold, and sell advice.

### 4.2.2 Performance

• The system should process and analyze user data in real-time, ensuring recommendations are generated without significant delay.

• The backend system should be able to handle a significant number of user requests simultaneously, without performance degradation.

### 4.2.3 User Interaction

• The chatbot should engage users in a natural and interactive manner, providing clear and actionable advice.

• The platform’s UI/UX must be intuitive and easy to navigate, ensuring a positive user experience.

### 4.2.4 Scalability

• The system must be capable of scaling up as more users interact with it, maintaining performance and responsiveness even as the user base grows.

### 4.2.5 User Satisfaction

* Positive feedback from test users regarding the usability of the platform and the accuracy of the financial advice provided will be a key success metric.

# 5. Engineering Process

This section outlines the core engineering efforts behind the development of our AI-driven financial advisory web application. It details the steps taken throughout the project, from initial system design and algorithm development to frontend and backend integration and deployment.

## 5.1 Process

### 5.1.1 The Research

The foundation of our AI-driven financial advisory platform was built on extensive research to ensure that the solution meets market demands while addressing user needs. We began by thoroughly exploring the financial advisory ecosystem, focusing on traditional services, robo-advisors, and emerging AI-powered platforms. Our research included an in-depth analysis of current solutions, examining their structure, features, and user experiences. By analyzing market leaders and niche players, we identified key trends and gaps, particularly in the areas of user engagement, real-time data utilization, and investment personalization.

Understanding user preferences was critical, and to gather feedback, we conducted interviews and [surveys](https://forms.gle/tzNmxqgGuDULdpPh8) with people from different financial backgrounds, from beginners to experienced investors. Many users expressed a desire for easy-to-understand, personalized financial advice that allowed them to adjust their investment strategies. The feedback indicated frustrations with the lack of affordable, tailored financial guidance, leading us to focus on building a user-friendly platform with flexible investment options. Real-time, accurate data also emerged as a crucial factor for gaining users’ trust.

### 5.1.2 User Personas and Their Needs

Persona 1: Gili, The Young early-career individual

• Age: 26

• Occupation: Marketing Manager

• Income: $75,000 per year

• Investment Experience: Beginner (Less than 1 year)

• Financial Goals: Building wealth and saving for a home

• Risk Tolerance: Moderate

Background:

Gili is an early-career individual who is just starting her investment journey. She is highly motivated to save for long-term goals like purchasing a home and building financial security. However, she has limited experience with stocks and investing and is intimidated by complex financial terms. She prefers to use technology to help her make decisions but is cautious about taking too much risk.

Needs:

• Simple, easy-to-understand investment advice tailored to her risk tolerance.

• A user-friendly platform that can help her learn the basics of investing.

• Regular updates on her portfolio without requiring constant monitoring.

• Recommendations for long-term investments that align with her savings goals.

Interaction with the Platform:

Gili will primarily interact with the AI chatbot to ask questions about stock performance and get advice on which investments are safe but offer growth potential. She values the educational resources that help her understand her portfolio better and would appreciate features that allow her to customize her goals without overwhelming her with data.

Persona 2: Israel, The Seasoned Investor

• Age: 45

• Occupation: IT Consultant

• Income: $150,000 per year

• Investment Experience: Advanced (10+ years)

• Financial Goals: Maximize return on investment (ROI), diversify portfolio

• Risk Tolerance: High

Background:

Israel has been investing in the stock market for over a decade and is very comfortable with market trends, technical analysis, and portfolio management. He actively monitors his investments and is always on the lookout for new opportunities to maximize his ROI. He is well-versed in investment strategies and uses various tools for research but is interested in leveraging AI for data-driven insights.

Needs:

• Advanced tools for real-time data analysis and stock performance tracking.

• Customizable investment strategies that allow him to tweak AI recommendations.

• Insights into market trends, sector performance, and predictive analytics.

• Seamless integration of his current portfolio with the platform’s analytics.

Interaction with the Platform:

Israel will use the platform’s real-time data tools to analyze stocks, using AI recommendations as a supplement to his own research. He will appreciate the ability to customize AI-driven advice to match his aggressive investment strategy. He’s interested in predictive models and detailed insights that can give him a competitive edge in making quick, high-return trades.

Persona 3: Sara, The Retiree

• Age: 62

• Occupation: Retired School Teacher

• Income: $50,000 per year (from pension and savings)

• Investment Experience: Intermediate (3-5 years)

• Financial Goals: Maintain financial security in retirement, preserve capital

• Risk Tolerance: Low

Background:

Sara is a retired teacher who is primarily focused on preserving her savings while ensuring steady growth to maintain her lifestyle during retirement. She has moderate investment knowledge and prefers low-risk investments. Emily doesn’t actively monitor the stock market and seeks a platform that provides clear, reliable advice without complex financial jargon.

Needs:

• Low-risk, conservative investment recommendations that ensure capital preservation.

• Simple, clear explanations of how her investments are performing.

• Minimal interaction, preferring automatic portfolio management and periodic updates.

• Secure platform with an emphasis on protecting her financial data.

Interaction with the Platform:

Sara will rely heavily on the platform’s automated portfolio management feature, opting for safe investments with low volatility. She will engage with the AI chatbot occasionally to check on her portfolio’s performance but will prefer to receive regular, easy-to-understand updates via email or notifications. She values security and the platform’s ability to ensure her investments are safe.

### 5.1.3 Conclusions from Research

Based on our research findings, we identified key features and system requirements that shaped the platform. Users want personalized investment strategies, real-time financial data, and a simple, intuitive interface. These insights directly influenced the modular design of the system, ensuring scalability and adaptability for future enhancements.

A significant takeaway was the importance of differentiating the platform from competitors. Our focus shifted to combining deep data analysis with AI-powered stock insights and providing an interactive chatbot for engaging user experiences.

We also realized the necessity of integrating robust AI models and tools like CrewAI to handle multi-agent collaboration efficiently.

### 5.1.4 Methodology and Development Process

For the development of our AI-driven financial advisory platform, we adopted an iterative development approach, which allowed us to build and refine the system incrementally, incorporating feedback at every step. This method ensured flexibility and adaptability as we worked toward delivering a functional and user-centric product.

We began by developing a simple prototype of the chatbot, which served as the core interaction point between the user and the platform. This initial prototype focused on the chatbot’s ability to understand and respond to natural language queries related to stock investments. Early user testing was conducted to gather feedback on the chatbot’s functionality and interface. The insights gained from this process helped us make crucial adjustments to enhance its usability, ensuring that it provided clear, personalized financial advice while being intuitive to interact with.

Our next step involved constructing the system’s architecture with scalability and flexibility as key objectives. We selected ReactJS for the front-end, leveraging its ability to create a dynamic and responsive user interface. FastAPI was chosen for the back end to efficiently handle real-time requests and facilitate seamless integration with external financial APIs.

For data storage, we opted for MongoDB, which supports the management of unstructured financial data and can easily scale as the platform grows.

To enhance the analytical capabilities of the platform, we integrated CrewAI and large language models (LLMs). These technologies allow the system to process complex financial data and interact with users in natural language. By adopting an agent-based architecture, we ensured that different AI agents could handle various tasks such as analyzing market trends, recommending investments, and providing portfolio management tips.

The development process followed a series of well-defined stages:

* Chatbot Prototype Development and Testing:

Creating the chatbot interface and refining its responses through user feedback.

* Front-End and Back-End Integration:

Building the architecture using ReactJS and FastAPI, ensuring real-time data flow.

* Data Management:

Implementing MongoDB to store and manage user and financial data securely.

* AI Integration:

Incorporating CrewAI and LLMs to enhance financial analysis and improve user interactions.

* Iterative Testing and Refinement:

Continuously testing and refining the system based on user interactions and performance metrics.

Each stage was followed by a thorough evaluation to identify any issues and apply the necessary adjustments before moving on to the next phase. This approach enabled us to maintain a modular and scalable design, allowing for future expansion and feature integration without disrupting the platform’s core functionality.

## 5.2 Product

In today’s complex financial environment, making sound investment decisions can be challenging, particularly for individuals without extensive financial knowledge. Our project seeks to simplify this process by developing an AI-driven financial advisory platform that delivers personalized investment advice, helping users make informed financial decisions based on their unique goals and preferences.

To support this AI-driven system[[8]](#footnote-8), we rely on FastAPI for the backend, ensuring efficient processing of real-time data and seamless communication between users and the platform. The real-time data, sourced from external financial APIs, is processed and delivered in a way that ensures users always receive the latest information. For data management, we utilize MongoDB, a robust, scalable NoSQL database that allows us to store and manage large volumes of financial data, user profiles, and historical transactions. MongoDB’s flexibility in handling unstructured data is critical for managing the diverse and evolving needs of our users.

The front-end of the platform is built using ReactJS, a powerful JavaScript library that enables the creation of a responsive, user-friendly interface. The user experience (UX) design is centered around simplicity and accessibility, ensuring that both novice and experienced investors can easily navigate the platform. Users will have access to a personalized dashboard where they can view their investment portfolios, receive real-time updates, and interact with the AI-driven chatbot for advice. The interface is designed to be intuitive, with clean navigation and real-time notifications, making financial management as straightforward as possible.

In terms of key features, the platform will offer users the ability to:

• Customize their investment strategies by adjusting risk levels and goals.

• Receive real-time stock recommendations based on comprehensive AI analysis.

• Track and manage their portfolios through a dynamic dashboard with data visualizations.

• Engage with an AI-powered chatbot that provides financial advice in a conversational, easy-to-understand manner.

### 5.2.1 Requirements

#### Functional

* The system will allow users to create accounts and securely log in.
* The system will enable users to create and update their personal and financial profiles.
* The system will integrate with external financial APIs.
* The system will analyze user financial data and provide tailored investment advice.
* The system will provide an AI-driven.
* The system will allow users to customize their investment strategies.
* The system will present financial data through visual elements.

#### Non-Functional

* The system will comply with data protection.
* The system will store user data.
* The system will use CrewAI to integrate multiple AI agents.
* The system frontend will be based on ReactJS framework.
* The system backend will be based on FastAPI framework.
* The system data base will be based on Mongo.
* The system will be optimized for cross-platform compatibility, providing a consistent user experience on both desktop and mobile devices.
* The system will include detailed logging.

### 5.2.3 Diagrams

#### 5.2.3.1 Project architecture

This diagram illustrates the architecture of the AI-driven financial advisory platform, where CrewAI and Large Language Models (LLMs) provide advanced financial analysis and natural language processing. The platform uses FastAPI for real-time backend processing and ReactJS for a responsive front-end, which displays stock data via an integrated Stock API.

MongoDB is used to store user data and financial records, with all components interacting seamlessly through a web browser interface to deliver personalized investment advice.

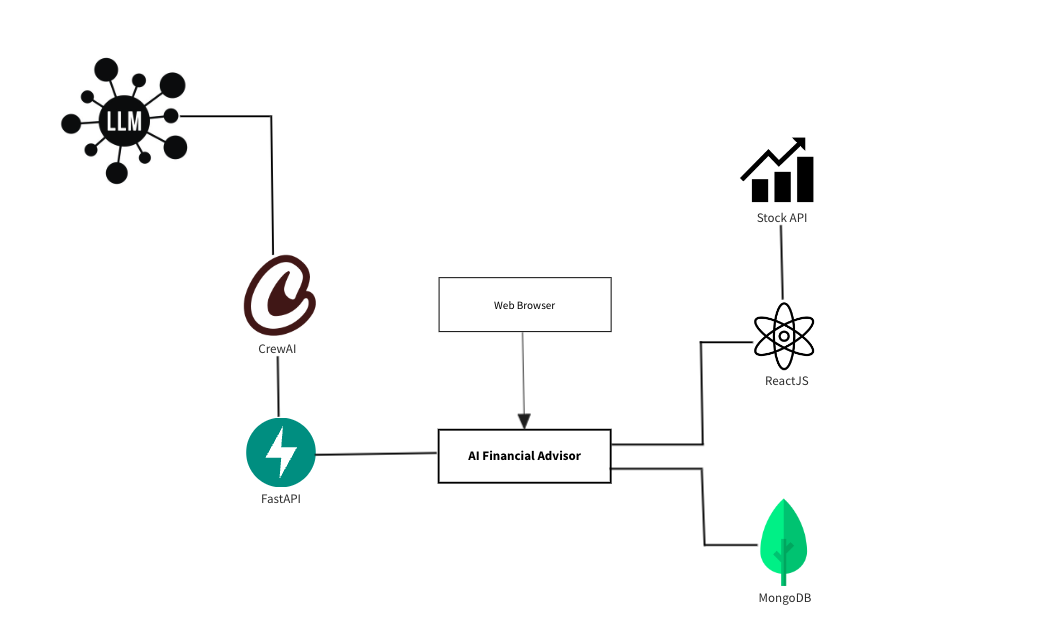


Figure 1 Project architecture

#### 5.2.3.2 Use case diagram

This use case diagram illustrates how users interact with the AI Financial Advisor platform. The main user actions include registering and logging in to access the system. Once logged in, users

can perform various tasks such as editing their profile, interacting with the AI chatbot, viewing their stock portfolio, receiving stock advice, and getting top-trending stock suggestions.

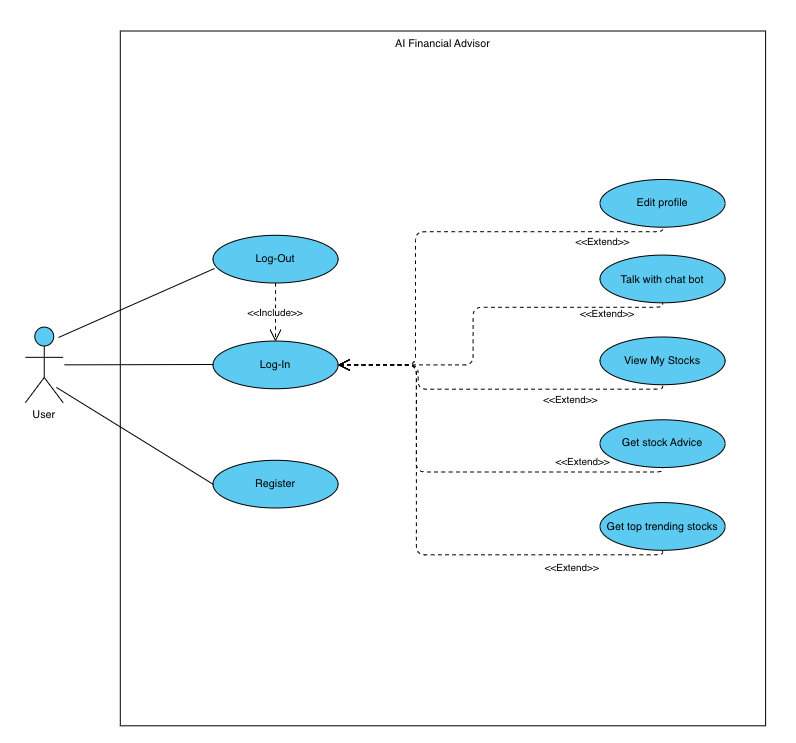


Figure 2 Use case diagram

#### 5.2.3.3 Activity Diagram

This activity diagram outlines the process for obtaining stock advice on the AI Financial Advisor platform. The user begins by either logging in or registering if they are new. After logging in, the user presses the “Get stock advice” button, which leads to a sequence of steps. The user selects their preferred stocks, defines their financial goal, and chooses their preferred investment method. Once these preferences are set, the system generates stock recommendations. The user can then review the recommendations and decide whether they are satisfied. If satisfied, the process concludes; if not, the user can adjust their selections and repeat the process to receive new recommendations.

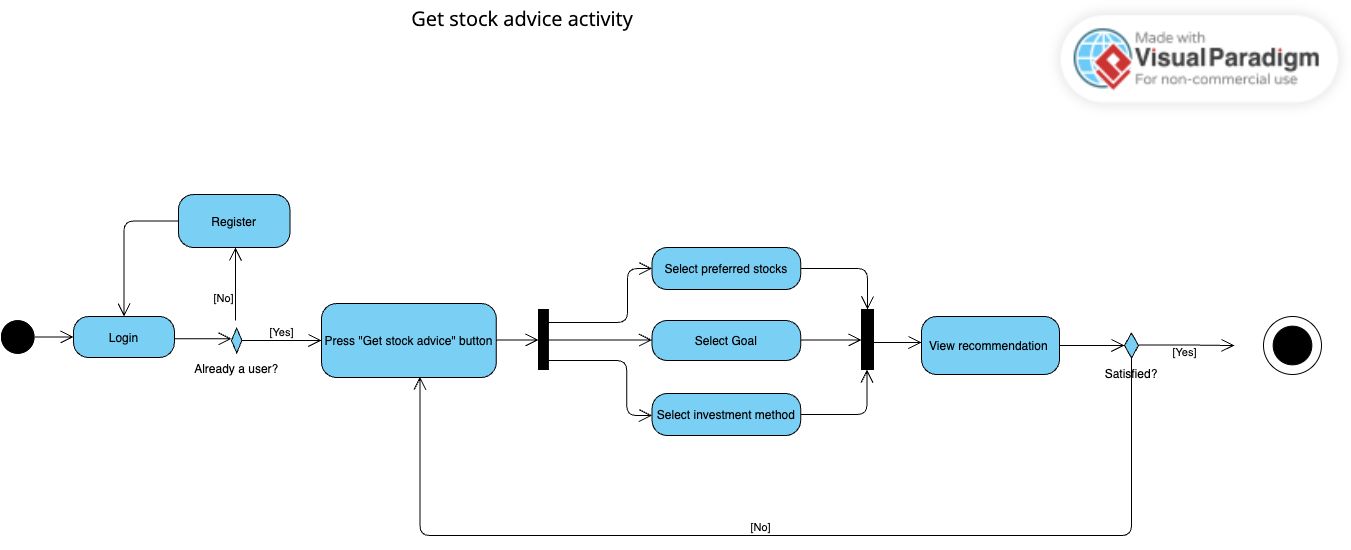


Figure 3 Activity Diagram

### 5.2.4 Deployment

Follow these instructions to activate the chatbot.

1. Install Ollama from <https://ollama.com/download>.
2. On the terminal run “ollama pull openhermes” .
3. Download the project from GitHub <https://github.com/ChayFadida/AIFinancialAdv.git> (using git clone).
4. cd AIFinancialAdv
5. cd fastApi
6. pip install -r requirements.txt
7. python app.py
8. npm install
9. npm run start

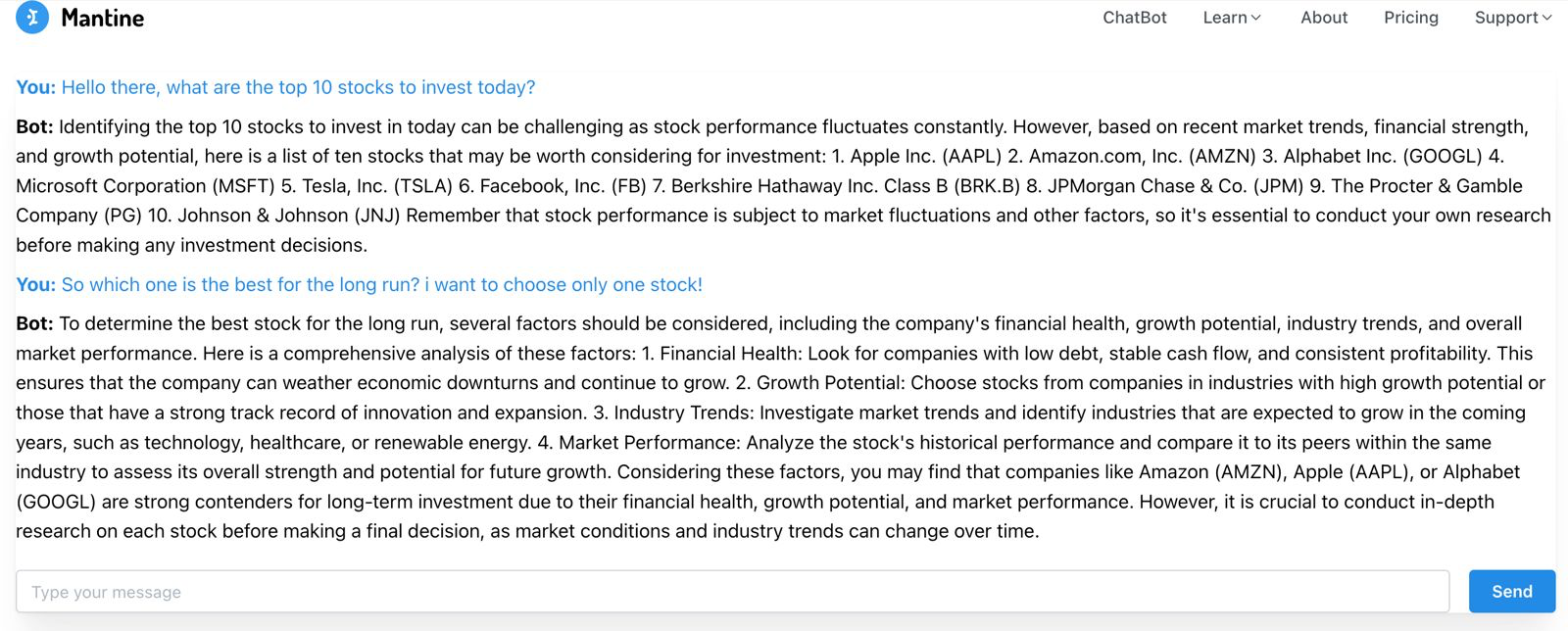


Figure 4 Chatbot example

### 5.2.5 Future Development and Feature Expansion

Looking ahead, the next phase of development will focus on expanding the platform’s core features. We plan to integrate more advanced tools for financial analysis, such as predictive models for stock performance and a portfolio management dashboard. These features will offer users deeper insights into their investments, including risk assessment, portfolio diversification, and long-term financial planning.

Additionally, we aim to improve the customization options for users. This will allow users to set personalized goals and adjust the chatbot’s recommendations based on their risk tolerance or investment preferences. Further enhancements to the chatbot will include broader access to financial data sources, allowing it to provide even more tailored advice.

One of the primary constraints in future development is ensuring smooth integration with various external financial systems, such as real-time stock and market data sources. Maintaining compatibility with multiple APIs while ensuring the system remains responsive and reliable will be a key challenge. Additionally, as user data grows more detailed, securing this sensitive information becomes increasingly important, requiring robust security protocols and compliance with data protection regulations.

Another constraint is the need to maintain the accuracy and relevance of the financial advice provided by the platform. Continuous updates and improvements to the AI models are essential to keep up with the rapidly changing financial markets.

# 6. Evaluation and Verification

## 6.1 Verification

For Back-End Testing (Python/FastAPI), we will use Pytest for unit testing, functional testing, and API endpoint testing. Pytest will be used to test individual components such as functions and classes to ensure they behave as expected. Additionally, we will use FastAPI’s TestClient and HTTPX to simulate HTTP requests and verify that all API endpoints function correctly. Integration testing will be conducted to ensure that various components, such as the database and external API integrations, work together seamlessly. Finally, performance testing will be performed to ensure the back end can handle multiple requests efficiently without any delays, maintaining optimal performance under load.

For Front-End Testing (React), we will use Jest to perform unit testing on individual React components, ensuring they render and behave as intended. React Testing Library will be employed to simulate user interactions, such as clicking buttons and verify that the components respond as expected.

Snapshot testing will ensure that the UI remains consistent when updates are made to the code.

Finally, we will integrate Cypress for end-to-end testing, which will verify the complete application flow and ensure that the front-end interacts seamlessly with the back end.

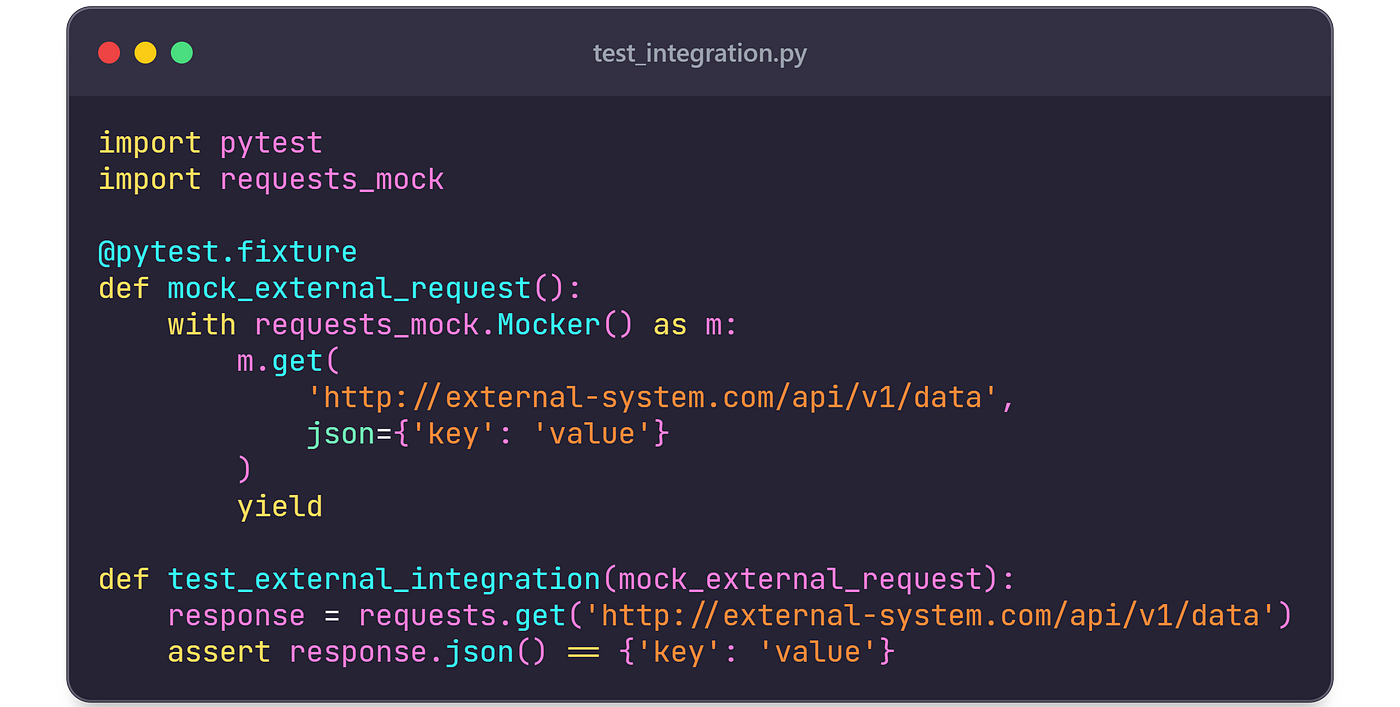


Figure 5 Pytest example

## 6.2 Evaluation

We will evaluate our product based on its ability to provide accurate and personalized financial advice, particularly focusing on stock investments. The primary metric for this evaluation will be the accuracy of the system’s recommendations and the relevance of these recommendations to users’ financial goals and risk tolerance.

This will be achieved by gathering user financial data and applying AI-driven analysis to categorize investment opportunities. Our system, powered by CrewAI, will analyze both fundamental and technical factors to offer tailored investment strategies that meet the unique needs of each user.

Our goal is to deliver a fast, reliable, and user-friendly financial advisory tool that helps users make informed investment decisions. The system’s success will be measured by how well it predicts stock performance and aligns its advice with real-time market conditions.

If we find a strong correlation between the system’s advice and improved portfolio performance or better alignment with user-defined financial goals, this will validate the accuracy of our models. These findings can then be further explored through additional analysis and feedback from users, allowing us to refine and optimize the system for even more precise recommendations.

# 7. AI prompts

As part of our research process, we utilized ChatGPT and Gemini as AI tools to gather insights and explore the effectiveness and potential challenges of AI-driven financial advisory platforms.   
Below are the key AI prompts that shaped our analysis.

* “Analyze the effectiveness of AI-driven financial advisory platforms in comparison to traditional financial advisors. Specifically, investigate the role of Large Language Models (LLMs) in providing personalized investment advice, focusing on factors such as accuracy, user satisfaction, and scalability. Gather data from existing platforms using LLMs and compare it with user feedback from traditional advisory services to assess the potential benefits and limitations of AI integration.”
* “Research the challenges and opportunities of integrating real-time financial data into AI-driven platforms. Explore the technical and ethical considerations of using APIs for stock market data, including data security, latency, and the accuracy of AI-generated investment recommendations. Analyze how these factors influence user trust and decision-making in the financial advisory domain.”

# 8. References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., & Neelakantan, A.

(2020). Language Models are Few-Shot Learners. OpenAI. Retrieved from

<https://arxiv.org/abs/2005.14165>

Chandak, Y., Choi, C., & Thomas, P. (2023). Q-function Induced Action Transformations for Safe Reinforcement Learning. arXiv. <https://arxiv.org/abs/2409.06289>

Erl, T., Cope, R., & Naserpour, A. (2017). *Web Service Contract Design and Versioning for*

*SOA*. Prentice Hall.

MongoDB, Inc. (n.d.). *MongoDB: A flexible, scalable database solution*. MongoDB

Documentation. Retrieved from <https://www.mongodb.com/what-is-mongodb>

Pandey, S., Vemula, A., & Cohen, W. W. (2023). Efficient and Accurate Hierarchical

Document Ranking Using Structured Representations. arXiv. <https://arxiv.org/abs/2408.11740>

Ramírez, S. (2018). FastAPI: A Modern Web Framework for Building APIs with Python 3.6+ Based on Standard Python Type Hints. Retrieved from <https://fastapi.tiangolo.com>

ReactJS Foundation. (n.d.). *React – A JavaScript library for building user interfaces*. ReactJS

Documentation. Retrieved from https://reactjs.org/

Vemula, A., Kumar, R., Liu, Y., & Cohen, W. W. (2023). Learning Personalized Retrievers for Search-Based Question Answering. arXiv. <https://arxiv.org/abs/2408.07720>

1. Chandak, Y., Choi, C., & Thomas, P. (2023). Q-function Induced Action Transformations for Safe Reinforcement Learning. [↑](#footnote-ref-1)
2. Pandey, S., Vemula, A., & Cohen, W. W. (2023). Efficient and Accurate Hierarchical

   Document Ranking Using Structured Representations [↑](#footnote-ref-2)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., & Neelakantan, A.

   (2020). Language Models are Few-Shot Learners. [↑](#footnote-ref-3)
4. Vemula, A., Kumar, R., Liu, Y., & Cohen, W. W. (2023). Learning Personalized Retrievers for Search-Based Question Answering. [↑](#footnote-ref-4)
5. Ramírez, S. (2018). FastAPI: A Modern Web Framework for Building APIs with Python 3.6+ Based on Standard Python Type Hints. [↑](#footnote-ref-5)
6. ReactJS Foundation. (n.d.). *React – A JavaScript library for building user interfaces*. [↑](#footnote-ref-6)
7. *MongoDB: A flexible, scalable database solution*. MongoDB Documentation. [↑](#footnote-ref-7)
8. Erl, T., Cope, R., & Naserpour, A. (2017). *Web Service Contract Design and Versioning for SOA*. Prentice Hall. [↑](#footnote-ref-8)