

**FUNDAMENTAL STOCK ANALYSIS WITH LLMs AND QUALITATIVE
DATA – IMPACT OF GOVERNMENT POLICIES AND ECONOMIC
TRENDS**

By
TANG JIA HUI

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2025

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Ts. Dr Liew Sounng Yue, for his invaluable guidance and support throughout the development of this project. I am truly grateful for the opportunity to explore the intersection of Large Language Models (LLMs) and finance through this research. This project has been a significant step forward in my academic and professional journey, and I deeply appreciate your encouragement along the way.

Special thanks also go to Mr. Chng Chee Henn for his valuable technical guidance and insights throughout the project. His support played an important role in helping me address key challenges and contributed greatly to the successful development of the system. I truly appreciate the time and knowledge he shared with me.

Finally, I must express my heartfelt appreciation to my parents and my family for their love, support, and continuous encouragement throughout the course.

COPYRIGHT STATEMENT

© 2025 Tang Jia Hui. All rights reserved.

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ABSTRACT

This report presents the development of a Virtual Analyst system for fundamental stock investment, powered by GPT-4o mini and other advanced technologies. The system leverages Large Language Models (LLMs) for processing and analysing qualitative data to provide comprehensive stock analysis and investment recommendations. The system integrates web scraping techniques to extract valuable information from diverse sources such as government policies, economic reports, news articles, and financial statements. The research process involved designing a modular architecture with five core components: financial report extraction, real-time news collection, inter-company relationship mapping, qualitative analysis of government policies and economic trends, and investment insight generation. Emphasis was placed on the qualitative analysis module, which leverages Retrieval-Augmented Generation (RAG) techniques to deliver contextually relevant insights. Preliminary testing validated the system's ability to generate accurate investment recommendations in JSON format. The conclusion highlights the system's potential to democratize sophisticated financial tools and to empower retail investors with actionable insights into stock growth prospects. Planning for future work includes real-time data integration and scalability enhancements, ensuring alignment with the project's objectives of transforming financial decision-making. The proposed methods and technologies have been justified as suitable for achieving the system's objectives of delivering actionable and contextually relevant insights into stock growth prospects, thus demonstrating the potential to transform decision-making in fundamental stock analysis.

Area of Study: Application Development, Large Language Models

Keywords: Fundamental Stock Analysis, Large Language Models, Qualitative Data, Government Policies, Economic Trends

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	II
COPYRIGHT STATEMENT	III
ABSTRACT	IV
TABLE OF CONTENTS	V
LIST OF FIGURES	VIII
LIST OF TABLES	XI
LIST OF ABBREVIATIONS	XII
CHAPTER 1: INTRODUCTION.....	1
1.1 PROBLEM STATEMENT AND MOTIVATION	1
1.2 PROJECT SCOPE AND DIRECTION	2
1.3 PROJECT OBJECTIVES	3
1.4 IMPACT, SIGNIFICANCE AND CONTRIBUTION.....	4
1.5 BACKGROUND INFORMATION.....	4
1.5.1 Advancements in Fundamental Analysis.....	5
1.5.2 Impact of Government Policies on Industries.....	7
1.5.3 Influence of Economic Trends on Stock Growth	8
1.5.4 Leveraging Natural Language Processing (NLP) for Analyzing Qualitative Data in Stock Growth Prospect.....	9
1.5.5 Applications of Generative AI in Stock Analysis.....	10
1.6 REPORT ORGANIZATION.....	10
CHAPTER 2: LITERATURE REVIEW.....	13
2.1 INVESTMENT THEORIES	13
2.2 TRADITIONAL METHODS OF STOCK GROWTH EVALUATION	14
2.3 QUANTITATIVE DATA ANALYSIS AND EXISTING SYSTEMS	15
2.3.1 Artificial Neural Networks (ANNs).....	15
2.3.2 Hybrid GRU (Gated Recurrent Unit) - CNN (Convolutional Neural Network).....	17
2.3.3 Support Vector Machines (SVMs).....	18
2.3.4 Hidden Markov Models (HMMs) with Fuzzy Logic	19
2.3.5 Generative Adversarial Network (GAN)	20
2.4 QUALITATIVE DATA ANALYSIS AND EXISTING SYSTEMS.....	22
2.4.1 Textual Content Sentiment Analysis	22
2.4.2 Embedding-Based Financial Report Analysis: GPT-InvestAR.....	25
2.4.3 Domain-Specific Instruction Tuning: InvestLM	25
2.4.4 Open-Source Data-Centric Framework: FinGPT	27

2.4.5	Portfolio Optimization with LLMs	27
2.4.6	Market Trend and Scenario Analysis	28
2.4.7	Decision Support Systems: CustomGPT	28
2.5	STRENGTHS AND WEAKNESS OF PREVIOUS STUDIES	29
2.6	PROPOSED SOLUTIONS	30
CHAPTER 3: SYSTEM METHODOLOGY/APPROACH		33
3.1	SYSTEM ARCHITECTURE DIAGRAM	33
3.2	USE CASE DIAGRAM AND DESCRIPTION	37
3.2.1	Use Case Diagram	37
3.2.2	Key Use Case Description	39
3.3	ACTIVITY DIAGRAM	47
3.3.1	Register Account	48
3.3.2	User Login Page	49
3.3.3	Data Fetching and Document Processing	50
3.3.4	Send Query	51
3.3.5	Manage Watchlist	52
3.3.6	Rate Analysis Quality	53
3.3.7	Receive News Notification	54
3.3.8	View News Dashboard	55
3.4	METHODOLOGIES AND GENERAL WORK PROCEDURES	55
CHAPTER 4: SYSTEM DESIGN		57
4.1	SYSTEM BLOCK DIAGRAM	57
4.2	SYSTEM COMPONENTS SPECIFICATION	58
4.2.1	API Interface and Request Handling	58
4.2.2	Flask Application Core	59
4.2.3	Cache Management System	60
4.2.4	Database	60
4.2.5	Document Chunking	62
4.2.6	Document Embedding	63
4.2.7	Workflow Coordination and Data Fetching	64
4.2.8	Document Indexing and Retrieval	65
4.2.9	Analysis Module and Insight Generation	66
4.2.10	Response Structuring and Delivery	69
4.2.11	Response Processing Module	70
4.2.12	Feedback Loop Mechanism	70
4.2.13	Watchlist Module	71
4.2.14	Automated News Processing System	72
4.2.15	Stock Notification Module	72
CHAPTER 5: SYSTEM IMPLEMENTATION		75
5.1	HARDWARE SETUP	75

5.2	SOFTWARE SETUP	75
5.3	SETTING AND CONFIGURATION	80
5.3.1	Local Development Setup	80
5.3.2	Online Hosting	80
5.4	SYSTEM OPERATION.....	81
5.4.1	Register Page	81
5.4.2	Login Page	82
5.4.3	Chat Interface.....	85
5.4.4	Feedback Feature	90
5.4.5	Watchlist Management	91
5.4.6	News Analysis	93
5.4.7	News Alert	95
CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION		96
6.1	SYSTEM TESTING AND PERFORMANCE METRICS	96
6.2	FUNCTIONAL TESTING.....	97
6.2.1	Use Case Testing.....	97
6.2.2	Invalid Query Payloads.....	107
6.2.3	Query in Different Complexity	108
6.2.4	Cache Mechanism.....	112
6.3	NON-FUNCTIONAL TESTING.....	113
6.4	PROJECT CHALLENGES	117
6.5	OBJECTIVES EVALUATION.....	118
CHAPTER 7: CONCLUSION AND RECOMMENDATION		121
7.1	CONCLUSION	121
7.2	RECOMMENDATION	122
REFERENCES.....		124
APPENDIX.....		A-1
A.1	EXAMPLE FULL RESULTS RETURNED FROM QUERY	A-1
A.2	POSTER.....	A-7

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.5.1	Framework Approach of StockAssistant	10
Figure 2.1.1	Stock Performance Hypothesis and Theories	13
Figure 2.3.1	Architecture of the Three-Layered ANN	16
Figure 2.3.2	Flowchart of GRU-CNN model for predicting the stock market trading signals	17
Figure 2.3.3	Architecture of GAN model	21
Figure 2.2.3	Out-of-sample Analysis Results for Different Daily Sentiment Models	24
Figure 2.4.2	Expert Evaluation Results of InvestLM	26
Figure 2.4.3	Framework of FinGPT	27
Figure 2.4.4	The process of how CustomGPT addresses user questions	29
Figure 3.1.1	Architecture Diagram of the Overall System	33
Figure 3.1.2	System Architecture Diagram of the Policy and Economic Trends Analysis Module	35
Figure 3.1.3	<i>High-level AI Agent Architecture with RAG Elements</i>	36
Figure 3.2.1	Use Case Diagram of Government Policies and Economic Trends Analysis Module	37
Figure 3.3.1	Activity Diagram of Register Account Feature	48
Figure 3.3.2	Activity Diagram of User Login Page	49
Figure 3.3.3	Activity Diagram of Data Fetching and Document Processing	50
Figure 3.3.4	Activity Diagram of Data Fetching and Document Processing	51
Figure 3.3.5	Activity Diagram of Manage Watchlist	52
Figure 3.3.6	Activity Diagram of Rate Analysis Quality	53
Figure 3.3.7	Activity Diagram of Receive News Notification	54

Figure 3.3.8	Activity Diagram of View News Dashboard	55
Figure 3.4.1	Agile Development Process	55
Figure 4.1.1	Block Diagram of the Government Policies and Economic Trends Impact Analysis Module	57
Figure 4.2.1	Pinecone Dashboard	61
Figure 4.2.2	RAG Data Ingestion Pipeline	62
Figure 4.2.3	RAG Query Processing and Augmentation Flow	64
Figure 5.2.1	Scatter plot comparing MMLU benchmark scores (general reasoning quality) and pricing (USD per 1M tokens) for smaller AI models	79
Figure 5.3.1	Terminal Output of Flask Server Initialization	80
Figure 5.3.2	Cloudflare Tunnel for Online Hosting	80
Figure 5.4.1	Registration page with password validation displaying red warning message for incorrect format	81
Figure 5.4.2	Verification email sent to user	82
Figure 5.4.3	Login page featuring the “Remember Me” option for persistent sessions	82
Figure 5.4.4	Sign in with Google option in the Login Page	83
Figure 5.4.5	Google Account Selection Interface	84
Figure 5.4.6	Login Successful Message	84
Figure 5.4.7	Empty Chat Interface	85
Figure 5.4.8	Ticker Auto-filling	85
Figure 5.4.9	Analysis result returned	86
Figure 5.4.10	System response generated after the user clicks a suggested follow-up question	88
Figure 5.4.11	Feedback window allowing users to rate the analysis	89
Figure 5.4.12	Watchlist Dashboard	90
Figure 5.4.13	Stock search with autocomplete suggestions	90
Figure 5.4.14	Example User Watchlist Storage in MongoDB	91
Figure 5.4.15	Watchlist view with sector-based filtering	91
Figure 5.4.16	Watchlist view in List View	92
Figure 5.4.17	Processed News Impact	93

Figure 5.4.18	List of latest Malaysia news with clickable headlines that open analysis in a small bottom-right window	93
Figure 5.4.19	Daily news alert feature highlighting which articles impact the stocks in the user's watchlist	94
Figure 6.2.1	Input Validation Test Results of Empty and Short Query	106
Figure 6.2.2	Input Validation Test Results using Long Query	106
Figure 6.2.3	Input Validation Test Results of Absence of Query Key	106
Figure 6.2.4	Standard Query Tests	107
Figure 6.2.5	Sector-Specific Standard Query	108
Figure 6.2.6	Long-Term Impact with Sector Focus	109
Figure 6.2.7	Query with Broader Economic Implications	110
Figure 6.2.8	Baseline Query	111
Figure 6.2.9	Exact Same Query with Baseline Query	111
Figure 6.2.10	Semantically Similar Query	111
Figure 6.3.1	Baseline VS Existing System Comparison in RAGAS Metrics	113

LIST OF TABLES

Table Number	Title	Page
Table 3.2.1	UC001 Register Account Use Case Description	39
Table 3.2.2	UC002 Login Use Case Description	40
Table 3.2.3	UC003 Send Query Use Case Description	41
Table 3.2.4	UC004 Manage Watchlist Use Case Description	43
Table 3.2.5	UC005 Rate Analysis Quality Use Case Description	44
Table 3.2.6	UC006 Receive News Notification Use Case Description	45
Table 3.2.7	UC007 View News Dashboard Use Case Description	46
Table 5.1.1	Specifications of Computer	75
Table 5.2.2	Overview of the Software and Development Tools Used	75
Table 6.3.1	Testing Query and Response by Baseline VS Existing System	113

LIST OF ABBREVIATIONS

<i>ANNs</i>	Artificial Neural Networks
<i>RNNs</i>	Recurrent Neural Networks
<i>LLMs</i>	Large Language Models
<i>GPT-4o MINI</i>	Generative Pre-trained Transformers-4
<i>EPS</i>	Earnings Per Share
<i>US</i>	United States
<i>COVID-19</i>	Coronavirus Disease of 2019
<i>EU</i>	European Union
<i>GDP</i>	Gross domestic product
<i>NLP</i>	Natural Language Processing
<i>AI</i>	Artificial Intelligence
<i>EMH</i>	Efficient Market Hypothesis
<i>AMH</i>	Adaptive Market Hypothesis
<i>PER</i>	Price Earnings Ratio
<i>DER</i>	Debt to Equity Ratio
<i>NPM</i>	Net Profit Margin
<i>DPR</i>	Dividend Payout Ratio
<i>ROA</i>	Return on Assets
<i>DCF</i>	Discounted Cash Flow
<i>DDM</i>	Dividend Discount Model
<i>FCFE</i>	Free Cash Flow to Equity
<i>CNNs</i>	Convolutional Neural Networks
<i>SVMs</i>	Support Vector Machines
<i>HMMs</i>	Hidden Markov Models
<i>GANs</i>	Generative Adversarial Networks
<i>BERT</i>	Bidirectional Encoder Representations from Transformers
<i>NER</i>	Named Entity Recognition
<i>RAG</i>	Retrieval-Augmented Generation
<i>OCR</i>	Optical Character Recognition
<i>PDFs</i>	Portable Document Formats

<i>JSON</i>	JavaScript Object Notation
<i>HTTP</i>	Hypertext Transfer Protocol
<i>HTTPS</i>	Hypertext Transfer Protocol Secure
<i>RESTFUL APIs</i>	Representational State Transfer Application Programming Interface
<i>NoSQL</i>	Not only Structured Query Language
<i>APIs</i>	Application Programming Interface
<i>FYP</i>	Final Year Project
<i>TTL</i>	Time To Live
<i>RAGAS</i>	Retrieval-Augmented Generation Assessment System

CHAPTER 1: INTRODUCTION

In this chapter, the background, problem statements, objectives and motivations of the project, along with its contributions to the field of stock investment analysis and LLM-driven financial decision-making are presented.

1.1 Problem Statement and Motivation

Retail investors and traders frequently encounter difficulties in making informed investment decisions due to the overwhelming volume and complexity of qualitative data such as government policies, macroeconomic trends, and sector-specific developments that significantly influence stock performance. Most existing analytical platforms emphasize historical quantitative metrics and offer generalized insights that do not account for real-time policy changes or industry-specific dynamics [1], [2]. This inadequacy places non-institutional investors at a disadvantage compared to institutional players with access to sophisticated analytical capabilities. As a result, these tools fail to provide the contextual depth and personalization needed by non-institutional investors, thereby increasing the risk of suboptimal investment decisions. Research indicates that qualitative information, such as government policies, can enhance qualitative impact assessment in stock analysis [3], [4]. Therefore, developing a solution that can efficiently analyse qualitative data is not only relevant but essential for improving investment outcomes.

This project seeks to address these limitations by developing a Large Language Models (LLM)-powered stock analysis system that leverages GPT-4o mini, integrated with Retrieval-Augmented Generation (RAG) and domain-specific knowledge. Unlike general-purpose LLMs such as ChatGPT, Gemini, Grok which are not optimized for the financial domain, the proposed system will be enhanced with sector-aware data pipelines, real-time policy analysis, and sentiment evaluation to generate tailored, contextually relevant investment insights. The aim is to outperform existing solutions by providing a more accurate, timely, and personalized interpretation of qualitative data, thereby levelling the playing field for retail investors with decision-support tools that democratize access to advanced decision-making tools traditionally reserved for institutional investors.

1.2 Project Scope and Direction

This project delivers a modular LLM-powered financial analysis system, termed the *Virtual Analyst*, designed to assist retail investors in evaluating stock performance through qualitative data. The system comprises five core modules: financial report extraction, real-time news collection, inter-company relationship mapping, qualitative government policies and economic trends analysis, and investment insight generation. Each module contributes to a comprehensive investment framework. The first module performs structured data scraping from annual and quarterly reports to extract financial metrics such as earnings, assets, and liabilities. The second module gathers real-time financial news and commentary from verified media sources. The third module builds a dynamic graph-based representation of companies listed on Bursa Malaysia, revealing ownership links, partnerships, and sector affiliations. These three modules serve as foundational data pipelines to support downstream analysis.

The primary focus of this Final Year Project is the development of the fourth module, which conducts in-depth analysis of qualitative factors such as government policies, macroeconomic conditions, and sectoral regulations. Using Retrieval-Augmented Generation (RAG), it integrates a vector database (Pinecone) with a GPT-4o mini model to perform retrieval of relevant documents, including government publications, news transcripts, and financial commentary. These texts are parsed and embedded using LlamaIndex and OpenAI's API, enabling context-aware interpretation of regulatory impacts on industries and companies. The module assigns sectoral risk scores, summarizes economic implications, and returns structured insights via a Flask API. Results are stored in a MongoDB database and visualized through a user interface. This module synthesizes input from the previous components to provide a balanced analysis, which highlights growth opportunities and risk exposure and thus empowering retail investors to make more informed decisions. The output will include both explanatory narratives and structured analytics—such as sector-specific risk scores and sentiment-driven forecasts—designed to support users in making evidence-based investment decisions. The system will demonstrate its capabilities through curated case studies, such as evaluating a specific industry sector, such as telecommunications or energy, and analysing its projected market response to recent government policies or macroeconomic developments. The final deliverable of this project will be a functional

prototype of a “virtual stock analyst” system, with a particular focus on the development and demonstration of its qualitative analysis capabilities. Retail investors will be able to interact with the system via a user-friendly interface, querying it about a specific company or industry. In response, the system will generate intelligible, well-structured reports or direct answers that synthesize data from financial disclosures, relevant news, and the inferred impacts of government policies and macroeconomic trends. The modular architecture ensures scalability and allows this project to be fully integrated and extended for enhanced personalization and forecasting capabilities. By addressing the limitations of traditional models and integrating diverse data sources, the system offers a robust framework for evaluating stock growth and market conditions.

1.3 Project Objectives

The primary objective of this project is to design and develop an AI-driven tool using large language models (LLMs) to enhance fundamental analysis of stocks, by incorporating qualitative data such as government policies and economic trends. This system is intended to support retail investors to make more informed and strategic investment decisions by providing structured insights and assessments derived from external qualitative factors often overlooked in traditional financial models.

The main objectives are divided into three sub-objectives as follows:

- To develop a system that utilizes LLM to analyse and summarize government policies and economic trends, offering insights into their impact on various industries and helping investors stay informed about relevant external factors.
- To implement tools to assess the effects of historical government policies and economic trends on industry performance, providing contextual insights into how industries responded to past events, thereby supporting better understanding of sector behaviour.
- To enable qualitative outlook analysis for stock growth prospects by using LLMs to evaluate historical data and provide insights into the potential impacts of similar future policy changes and thus assisting investors in anticipating future market shifts.

1.4 Impact, significance and contribution

This final year project (FYP) makes a substantial impact on the field of stock market analysis by creating an advanced tool that utilizes GPT-4o mini's natural language processing capabilities. The tool overcomes the limitations of current stock market analysis systems by delivering personalized and contextually relevant insights tailored to individual investor preferences. By democratizing access to analytical tools typically reserved for institutional investors, the project empowers individual investors to make more informed and strategic decisions. This is especially beneficial for retail investors and smaller traders who often lack access to sophisticated analytical resources.

Through the integration of real-time analysis of government policies, economic trends, and sentiment analysis from financial news, this tool keeps investors ahead of market shifts, offering them a nuanced understanding of market dynamics. This approach not only enhances individual financial outcomes but also contributes to a more efficient and inclusive market. Over time, it has the potential to reshape how retail investors engage with stock markets globally, promoting financial literacy and leveling the playing field in a space traditionally dominated by large institutions. The significance of this project lies in its innovative fusion of cutting-edge AI technology with practical financial applications, addressing the pervasive issue of information asymmetry in the market. The real-time capabilities and qualitative evaluation of historical and policy data provide a scalable solution that distinguishes this tool from static, outdated methods. This project is compelling as it showcases how technology can make advanced financial tools accessible to a broader audience, aligning with the global push for equity and empowerment. Beyond its immediate impact, the project holds long-term potential to inspire further innovations, expanding into new markets and asset classes, making it an essential and valuable contribution to the future of finance.

1.5 Background Information

The stock market has historically served as a focal point for investors aiming to accumulate wealth via strategic investments. Nonetheless, despite the opportunity for wealth generation, numerous retail investors sometimes incur losses owing to an insufficient comprehension of a company's financial stability and fundamentals. Benjamin Graham asserts, “In the short run, the market is a voting machine but in the long run, it is a weighing machine,” underscoring the importance of fundamental

analysis for comprehending wealth accumulation potential, as it offers profound insights into a company's intrinsic value, transcending market fluctuations and transient volatility [5]. Developing tools that aid retail investors in performing comprehensive fundamental analysis can enable them to make more informed decisions, hence facilitating more effective wealth accumulation. The assessment of stock growth potential is an essential element of fundamental analysis, a technique employed by investors to determine the underlying value of a stock through numerous financial measurements. Historically, the assessment of stock growth has predominantly depended on quantitative data—such as financial ratios, historical performance, and statistical models—to forecast future trends and evaluate investment potential [6]. These models, such as Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs), have undergone substantial research and optimisation for forecasting short-term stock price fluctuations based on historical data [6], [7]. Nevertheless, even quantitative methodologies frequently fail to encompass the complete range of factors that affect stock growth. Qualitative data, encompassing insights on government policies, economic trends, market attitude, company policies, and other non-quantifiable elements, is essential for comprehending the stock market [8]. Sentiment analysis has been a prevalent technique for interpreting qualitative data; however, the utilisation of advanced Large Language Models (LLMs) such as Generative Pre-trained Transformers-4 (GPT-4o mini) provides a more nuanced and comprehensive examination of diverse text-based data sources, including annual reports, news articles, and financial statements [9], [10], [11].

1.5.1 Advancements in Fundamental Analysis

Fundamental analysis in investing pertains to the assessment of a company's inherent value through the examination of pertinent economic and financial elements [12]. This analytical method entails a comprehensive assessment of the macroeconomy, industry dynamics, and the company's financial health and management practices. The goal is to derive a company's intrinsic value [12].

Fundamental analysis has markedly progressed, transitioning from a solely quantitative methodology to one that integrates qualitative input. Traditionally, fundamental analysis depended significantly on quantitative information derived from financial accounts. Investors employed indicators including earnings per share (EPS), revenue

growth, and financial ratios to assess a company's financial health and valuation. This quantitative methodology offered a structured framework for assessing organizations using quantifiable financial metrics [5].

With the evolution of the investment landscape, the acknowledgment of qualitative factors—such as management quality, competitive advantage, and market sentiment—grew in significance. This has caused a change in this methodology. Qualitative analysis entails evaluating non-numeric elements that may influence a company's performance. Investors now evaluate a wider array of elements, including governmental policies, market mood, and industry trends, which can provide more profound insights into a company's prospects. A company's competitive edge, business model, corporate culture and corporate governance are essential components that quantitative data alone cannot adequately encompass [13].

The current incorporation of LLMs into basic analysis signifies a substantial progression. LLMs possess the ability to handle and evaluate extensive volumes of unstructured qualitative data from many sources, including news articles, social media, and regulatory filings. This capacity enables investors to acquire insights on market sentiment, regulatory influences, and new trends. Nonetheless, it is crucial to acknowledge that although sentiment analysis can offer valuable background, our emphasis continues to be on fundamental analysis.

Although these rules and trends can subsequently impact market sentiment—melding investor perceptions of the stock—they primarily affect the company's fundamental qualities. The principal objective of fundamental analysis is to comprehend the direct effects on a company's intrinsic value, rather than merely market sentiment [12].

The integration of LLMs into fundamental analysis represents a significant advancement. LLMs can process and analyse vast amounts of unstructured qualitative data from sources such as news articles, social media, and regulatory filings [10]. This talent improves the comprehension of how governmental policies and economic trends influence essential factors. Large Language Models can assist in deciphering intricate legislative modifications and economic analyses, which subsequently influence financial indicators and corporate valuations.

1.5.2 Impact of Government Policies on Industries

Government policies exert a significant influence on industries, shaping their operational and financial dynamics. Regulatory policies, such as environmental regulations and labour laws, impact industrial practices and costs [14].

Fiscal stimulus, aimed at boosting economic activity, stabilizing employment, and supporting growth during downturns, can be implemented through tax measures or government spending, each with different multiplier effects and economic impacts [15]. Government policies, including fiscal and monetary measures, can either bolster or undermine market confidence [16], [17], [18]. For example, a study in [16] found that unexpected changes in monetary policy have a significant impact on stock prices, with a 25-basis-point cut in the federal funds rate leading to approximately a 1% increase in stock prices. Additionally, geopolitical events such as trade wars, sanctions or supply disruption can cause abrupt market shifts by influencing stock market synchronization and connectedness [19], [20]. For instance, an analysis of the co-movements among the United States (US), Chinese, and Russian markets in [19] shows that geopolitical risks, such as heightened geopolitical tension or sanctions, can significantly disrupt the total and bilateral connectedness of these markets. Higher levels of geopolitical risk, particularly during bullish market states or higher quantiles of geopolitical uncertainty, have been found to negatively impact the synchronization between these markets, especially in the US-China and US-Russia market pairs [19]. However, the stock market relationship between Russia and China appears to be less sensitive to such geopolitical fluctuations, highlighting the varying impact geopolitical events can have on different market pairs [19]. Additionally, geopolitical events significantly affect investor behaviour and market dynamics by altering the risk premium and influencing commodity prices like oil [20]. This effect is more pronounced for oil due to the direct impact of geopolitical events on oil supply, such as attacks on oil facilities, which may not receive extensive global media coverage [20]. Moreover, the dynamic relationship between oil and stock returns, including volatility spillovers from oil to stocks, emphasizes the need to understand how geopolitical risks shape market behaviour [20]. Understanding these factors is crucial for investors aiming to navigate the complexities of the stock market.

Changes in tax policy, regulatory regulations, or economic conditions can directly influence a company's revenue, expenses, and overall financial stability [21], [22], [23].

For instance, study in [15] demonstrates how the Covid-19 pandemic led the Indonesian government to implement three fiscal stimulus packages in 2020, aimed at mitigating the economic impact across various sectors. Analysis using an input-output model revealed that these stimulus measures significantly influenced economic output and labour income, particularly in sectors such as Manufacturing, Agriculture, Forestry and Fisheries, Wholesale and Retail Trade, and Health Services [15]. These essential alterations are evident in financial statements and critical performance indicators.

Trade policies also play a crucial role in shaping industrial competitiveness. Changes in trade regulations, such as export restrictions, trade agreements, and tariffs, can either protect domestic industries from foreign competition or increase costs and provoke trade conflicts [24]. For instance, in an effort to hasten the transition to renewable energy sources and lessen reliance on fossil fuels, the European Union (EU) lowered tariffs on components and raw materials used in the green energy industry [24]. This action supported Europe's renewable energy industry but increased competition for non-EU nations exporting comparable commodities [24]. Monetary policies also affect industries through influencing economic activity and borrowing costs, such as interest rate modifications. In their investigation of the effects of monetary policy shifts on the housing market industry, the study in [25] demonstrates that the changes in interest rates can significantly affect the housing market. A decrease in interest rates typically lowers borrowing costs, encouraging more people to take out mortgages and buy homes, which boosts housing demand and prices [25]. Conversely, an increase in interest rates raises borrowing costs, potentially dampening housing demand and cooling down the market [25].

1.5.3 Influence of Economic Trends on Stock Growth

Economic developments have a significant impact on market dynamics, investing strategies, and stock growth. Macroeconomic indicators like interest rates, inflation, exchange rates and GDP growth significantly influence stock prices and overall market trends [16], [26], [27].

One important aspect influencing the success of the stock market is the rate of economic growth. A strong economy is usually indicated by stronger gross domestic product (GDP) growth, which raises stock market performance [28]. Inflation trends also significantly influence stock growth. While moderate inflation can reflect a healthy

economy, high inflation often erodes purchasing power and raises business costs, negatively impacting corporate profits and stock prices [29]. Interest rates, managed by central banks, directly affect stock market performance. Lower interest rates reduce borrowing costs for companies, promoting investment and expansion, which can drive stock prices higher [29]. Conversely, higher interest rates can increase borrowing costs and slow economic growth, leading to negative effects on stock performance [29]. The study in [30] explores how interest rate changes impact stock markets, finding that lower rates are associated with higher stock returns, while rising rates can induce market volatility.

Consumer confidence is another crucial economic trend influencing stock growth. A key economic indicator is consumer confidence, which gauges how optimistic or pessimistic consumers are about the status of the economy as a whole [31]. High consumer confidence often leads to increased consumer spending, boosting corporate earnings and driving stock prices up. According to the study, there is a significant two-way causal relationship between stock returns and consumer confidence in both the US and Europe. This suggests that high levels of consumer confidence are positively correlated with stock growth because they reflect optimism about the state of the economy and future earnings [31].

For instance, the study in [26] shows that in the coal industry, exchange rates had a significant negative impact on stock prices, emphasizing the importance of macroeconomic conditions in stock markets. Additionally, while inflation was found to have no effect due to mild inflation rates in Indonesia, the exchange rate's impact underscores the relevance of these factors in understanding market fluctuations [26].

1.5.4 Leveraging Natural Language Processing (NLP) for Analyzing Qualitative Data in Stock Growth Prospect

Unlike quantitative data, which is often structured and easily analysable, qualitative data is unstructured, making it difficult to process and interpret. This complexity arises from the inherent variability in language, context, and sentiment. Emerging technologies, particularly in natural language processing (NLP), are beginning to address these challenges. NLP approaches, including categorization, sentiment analysis, trend identification and text mining, are utilized to extract and evaluate textual information from diverse sources, such as news articles, social media, and financial

reports [32], [33]. NLP quantifies sentiment and identifies crucial events that correspond with stock market fluctuations through the preparation and analysis of textual data [32], [33]. Advanced models such as LSTM and deep learning significantly improve prediction accuracy, facilitating superior forecasting of stock price variations based on qualitative observations [32].

1.5.5 Applications of Generative AI in Stock Analysis

Advancements in Generative AI, especially with advanced models such as GPT-4o mini, are transforming the finance sector. These AI models can analyse extensive volumes of unstructured textual data—such as headlines, earnings call transcripts, and regulatory filings—and derive actionable insights by leverage their Natural Language Processing (NLP) capabilities. Case studies demonstrate that generative AI are used to forecast market trends, analyse investor sentiment, provide individualized financial advice, and automate the creation of financial reports [11]. Stock comments are essential for forecasting stock volatility and trends. However, investors frequently encounter difficulties due to the presence of noisy and unclear viewpoints.

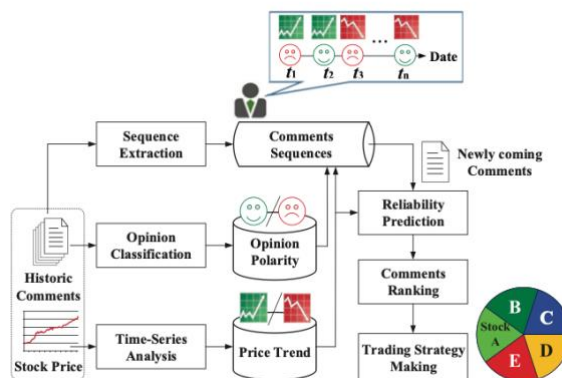


Figure 1.5.1 Framework Approach of StockAssistant. Source: [34]

To solve this problem, the study in [34] presents “StockAssistant”, a tool that evaluates comment reliability through the analysis of stock price patterns, comment content, and analyst performance utilizing ensemble learning techniques, as shown in Figure 1.5.1.

1.6 Report Organization

The report is divided into multiple chapters to describe the complete process of designing, developing, and assessing the project.

In **Chapter 1**, the project is introduced by providing the background, stating the problem, motivation, scope and goals. Also, the project's accomplishments and the organization of the report are outlined.

In **Chapter 2**, a detailed literature review is conducted. It analyses the literature on fundamental analysis, qualitative approaches to financial problem-solving, and the application of LLMs to financial analytics. This review analyses the construct and deconstruct of the sited research, and outlines how this project will fill the gaps found.

Chapter 3 is devoted to the system's methodology, which encompasses the architectural standpoint of the whole system and the necessary functional components. This chapter includes system use case and system activity diagrams that describe the system's and the user's actions.

Next, **Chapter 4** shifts the focus to system architecture and proposes the flow diagrams, construction design, and qualitative descriptions of each of the system's core components. Emphasis is placed on the integration of LLMs with RAG and financial data in the LLMs.

In **Chapter 5**, the functionality of the system is defined where the proficiency of the technical aspects of the system is polished. This includes the fulfilment of the MongoDB resting database, and the vector databases of Pinecone, as well as the yFinance, Google Search Engine APIs for the system's daily data acquisition. The chapter discusses the assessment of the system's inputs for the risk assessment of the sectors as well as the qualitative and quantitative configuration of the outputs.

In **Chapter 6**, the primary focus shifts toward the assessment of the whole system. This involves detail measurement of the system's precision as well as its overall efficacy. The functional and the non-functional system assessments are regulated in order to verify the claims which are made about the objectives which are supposed to be accomplished. This especially highlights the confidence on the assessment outputs and the holistic framework of the sector risk analysis.

Finally, the report is concluded in **Chapter 7** where the achievements of the project are presented. The chapter discusses the system's LLM moderation and rapid as well as valuable qualitative data acquisition to enrich the analysis from which the investments

CHAPTER 1

were made. The limitations are presented as well in regard to what optimizations as well as future work could be targeted on.

CHAPTER 2: LITERATURE REVIEW

2.1 Investment Theories

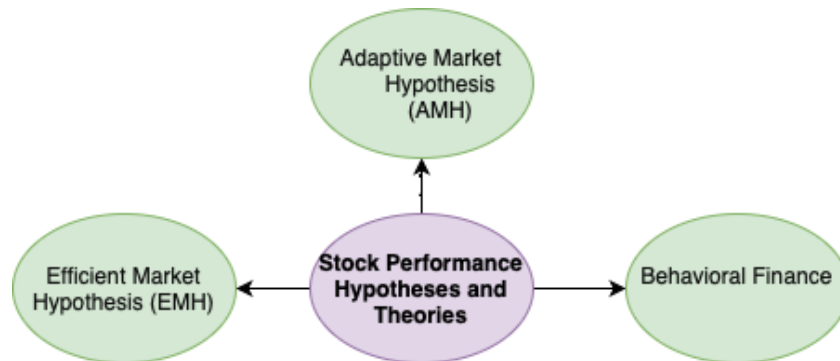


Figure 2.1.1: Stock Performance Hypothesis and Theories

Several hypotheses have been proposed by researchers to analyse stock performance. Among the most widely discussed is the **Efficient Market Hypothesis (EMH)**, which asserts that stock prices fully reflect all available information, thereby making it impossible to consistently achieve higher returns without assuming additional risk. This theory, introduced by [35], is grounded in the belief that markets are “informationally efficient,” meaning that all public information is already priced into stocks. However, EMH is often criticized for its assumption of perfect market conditions and rational investor behaviour, which may not hold true in real-world scenarios where anomalies, such as market bubbles and crashes, occur due to irrational behaviour [36].

Contrasting with EMH is the **Adaptive Market Hypothesis (AMH)**, proposed by [37], which offers a more flexible and evolutionary perspective on market behaviour. AMH suggests that market efficiency is not static but evolves as market participants adapt to changes in the environment, investor psychology, and new information. According to this hypothesis, financial markets are driven by a combination of rational and irrational behaviours that vary over time, allowing for the existence of inefficiencies that can be exploited through both **fundamental** and **technical analysis**. This view aligns with the increasing evidence that investor behaviour and sentiment significantly impact market prices, thereby allowing opportunities for active management to outperform passive strategies under certain conditions [38].

Furthermore, **Behavioural Finance** has emerged as a critical theory that challenges the assumptions of traditional finance theories like EMH. It integrates insights from

psychology to explain how cognitive biases, such as overconfidence, herd behaviour, and loss aversion, can lead to predictable market outcomes that deviate from the predictions of classical models [39]. Research shows that these biases can result in anomalies such as the momentum effect, where stocks that have performed well in the past continue to do well in the future, and the value effect, where undervalued stocks outperform overvalued ones [40].

2.2 Traditional Methods of Stock Growth Evaluation

Stock growth evaluation is a fundamental component of investment analysis, aiding investors in making informed decisions about potential investments. Traditional methods of stock growth evaluation primarily involve analysing financial statements and applying quantitative metrics such as financial ratios and valuation models. **Ratio analysis** is one of the most widely used methods for evaluating a company's financial health and growth prospects. Key financial ratios influencing stock prices include Earnings per Share (EPS), Price Earnings Ratio (PER), Debt to Equity Ratio (DER), Current Ratio, Net Profit Margin (NPM), Dividend Payout Ratio (DPR), and Return on Assets (ROA) [41], provide insights into a company's profitability, valuation, efficiency, and leverage. For example, a low P/E ratio may indicate that a stock is undervalued relative to its earnings, while a high ROE suggests efficient use of equity capital to generate profits.

Financial modelling, which entails constructing mathematical representations of a company's financial performance to predict future growth, is another conventional technique for evaluating stock growth [42]. These models vary in complexity, from straightforward single-variable models to intricate multi-variable models that consider several financial indicators, including operating costs, capital expenditures, profit margins, and revenue growth [42]. The **Discounted Cash Flow (DCF)** model is a widely used method that calculates the intrinsic value of a company by predicting its future cash flows and discounted them to their present value [2]. When evaluating a company's prospects for long-term growth while taking the time value of money into account, DCF is very important. Other models, such as the **Dividend Discount Model (DDM)** and the **Free Cash Flow to Equity (FCFE)** model, focus on predicting stock prices based on expected future dividends or cash flows available to shareholders, respectively [43].

Overall, traditional methods such as ratio analysis and financial modelling remain essential tools in stock growth evaluation due to their ability to distil complex financial information into actionable insights. However, these methods must be supplemented with a broader analysis that includes qualitative factors and external economic conditions to provide a more comprehensive view of a company's growth potential. As financial markets become increasingly complex, there is a growing need for more advanced techniques, such as machine learning and natural language processing, to augment traditional methods and address their inherent limitations [44].

2.3 Quantitative Data Analysis and Existing Systems

In the field of stock market analysis, to analyse past price movements and spot possible trends, deep learning algorithms are commonly used. Several models, including Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Support Vector Machines (SVMs), have been developed and improved [45], [46], [47]. ANNs are effective in identifying non-linear relationships among stock variables [45], while RNNs and CNNs have been employed to capture sequential dependencies and spatial patterns in time-series data, respectively [46]. These techniques are robust in handling big datasets and increasing prediction accuracy, but they have drawbacks. In turbulent market situations, overfitting and limited generalizability can result from most models' primary focus on technical indicators and historical performance. Furthermore, extra adjustments are needed for some methods, including fuzzy logic models and Hidden Markov Models (HMMs), to take non-stationarity and the complexity of financial markets into consideration [48]. Recent advancements have introduced more innovative techniques, like Generative Adversarial Networks (GANs), which can generate synthetic financial data that mimics real market conditions, enhancing model robustness [49].

2.3.1 Artificial Neural Networks (ANNs)

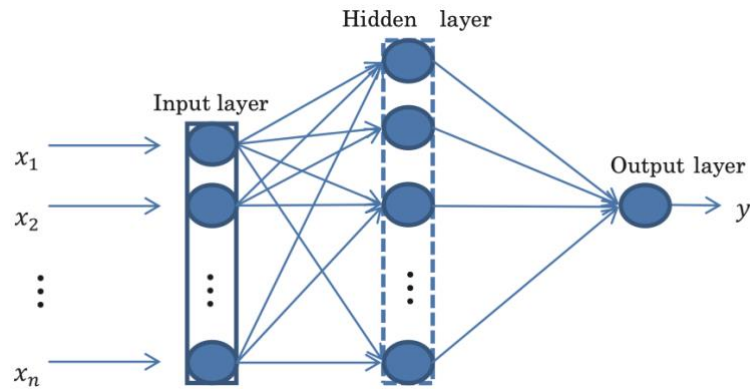


Figure 2.3.1: Architecture of the Three-Layered ANN. Source: [45]

The Japanese Nikkei 225 index prediction model in [45] employs a three-layer feedforward ANN with input, hidden, and output layers, as shown in Figure 2.2.1 to capture the nonlinear dynamics of stock market movements. By processing historical trading data and economic indicators, the model learns patterns that influence market trends. Training is carried out using the backpropagation (BP) algorithm, which reduces prediction error by iteratively adjusting model parameters. However, BP alone often converges slowly and can become trapped in local minima, limiting forecast accuracy. To address this, the study integrates metaheuristic approaches such as genetic algorithms (GA) and simulated annealing (SA). GA enhances predictive capability by searching globally for better weight–bias combinations, while SA helps the model escape poor local solutions. The hybrid GA-BP and SA-BP approaches improve market forecasting accuracy, with GA-BP achieving the lowest mean squared error (MSE) though at higher computational cost. Despite these enhancements, the model relies on a **shallow ANN architecture**, restricting its ability to capture complex sequential patterns in financial time series. Furthermore, like other structured-data-based models, it does not incorporate **unstructured textual or real-time data** such as news and policy announcements, which are crucial for modern stock market analysis. Overall, this model demonstrates how combining classical ANN structures with optimization strategies can strengthen stock index prediction, though its reliance on shallow architectures may restrict its adaptability compared to LLM-based system for stock analysis. The model continues to mainly depend on historical data and are less successful in integrating external qualitative factors—like government policies, and the calibre of the company's management as that are essential for a thorough assessment of stocks. Unlike quantitative data that primarily reflects a company's historical financial

performance, qualitative data provides insights into factors that are not easily quantifiable but significantly influence future performance.

2.3.2 Hybrid GRU (Gated Recurrent Unit) - CNN (Convolutional Neural Network)

The proposed GRU-CNN model in the study [46] is designed for financial time series prediction, specifically targeting stock market indices such as HSI, DAX, and S&P 500. The combination of a convolutional neural network (CNN) with a gated recurrent unit (GRU) allows the model to capture both local temporal patterns and long-term dependencies in stock prices, which is crucial for markets characterized by high volatility and complex correlations.

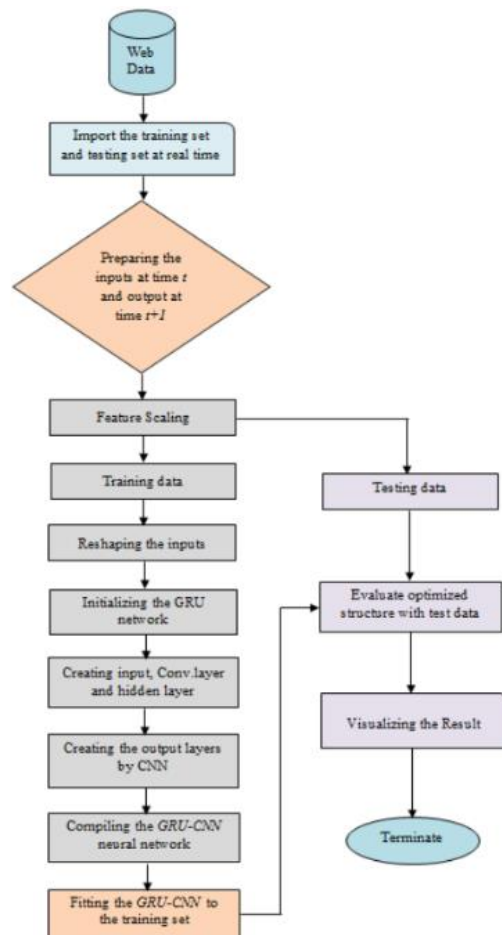


Figure 2.3.2: Flowchart of GRU-CNN model for predicting the stock market trading signals.

Source: [46]

As shown in Figure 2.3.2, the workflow begins with real-time importation of stock data from external web source. The data was then partitioned into training and testing sets and structured into input sequences at time t to predict outputs at $t+1$. After that, the feature scaling normalizes the data, reducing magnitude effects. The architecture consists of an input layer for sequential stock features. The second layer is a convolutional layer that extracts high-dimensional patterns over a context window. The third layer is a GRU layer with gating mechanisms to address vanishing gradient issues. Next, a pooling layer to reduce dimensionality, and finally a fully connected layer for predicting trading signals. While the GRU effectively models both short-term fluctuations and longer-term trends using update and reset gates. By combining CNNs with GRUs, the architecture can learn both short-term fluctuations, such as intraday or daily market noise, and longer-term trends influenced by macroeconomic indicators, which are crucial for real-world financial forecasting. One potential limitation is the sensitivity of predictions to hyperparameter settings, such as the number of hidden neurons or dropout rate, which could affect robustness across different market conditions.

Based on experimental results, the GRU-CNN achieved better accuracy, daily returns, and cumulative returns than the standard GRU, CNN, and DNN models, supporting the hypothesis that recurrent sequential modelling convolutionally feature extraction fusion is indeed effective. One important point, however, is that despite the improvements, the model's gains over GRU alone were modest, signifying that the convolutional layer adds complexity to the model without substantial predictive value for datasets dominated by temporal dynamics, thus excess computational burden. The model is also subject to severe overfitting to hyperparameter values such as the number of hidden neurons, convolutional kernel size, and the dropout rate, thereby reducing its applicability to cross different market conditions.

2.3.3 Support Vector Machines (SVMs)

Recent study in [45], [46], [47] have explored **Support Vector Machines (SVMs)** for stock market prediction due to their effectiveness in handling **high-dimensional and nonlinear financial data**. The way SVMs operate is that they try to determine the best hyperplane which divides the input features (i.e. the opening price, closing price, the

volume traded, and other technical indicators) into distinct classes that later correspond to different price movements.

A known limitation of traditional SVMs is their **sensitivity to irrelevant or redundant features**, which can introduce noise and reduce predictive reliability in financial time series. To overcome this, **Genetic Algorithms (GAs)** have been integrated with SVMs to perform **simultaneous feature selection and hyperparameter optimization**. By evolving candidate solutions through selection, crossover, and mutation guided by a fitness function, GA-SVM frameworks can significantly improve prediction accuracy, stability, and metrics such as RMSE over standalone models like Decision Trees or Neural Networks. Ensemble techniques, such as majority voting across multiple GA-optimized SVMs, further enhance robustness by reducing model variance.

However, these hybrid approaches are **computationally intensive**, with training times that can be orders of magnitude higher than conventional models. Moreover, while GA-SVMs excel at capturing technical patterns (e.g., SMA, MACD, RSI), they are **limited in handling unstructured textual financial information**, such as news or policy updates, which are increasingly critical in modern stock market analysis. This limitation is particularly relevant to **RAG-based systems**, which leverage retrieval of diverse textual data to provide richer, context-aware predictions. Despite these constraints, the GA-SVM framework remains a **technically rigorous benchmark**, demonstrating the value of hybrid optimization and ensemble learning in structured financial forecasting.

2.3.4 Hidden Markov Models (HMMs) with Fuzzy Logic

Fuzzy logic addresses uncertainty and imprecision by allowing truth values to range continuously between 0 and 1, rather than being strictly “true” or “false” (1 or 0). In the context of the stock market, it is particularly useful for modelling volatility, as it can capture the gradual and unpredictable fluctuations of stock prices more effectively than binary approaches.

Recent research [48] has explored hybrid models combining **Hidden Markov Models (HMMs)** with **fuzzy logic** to forecast stock market movements. The HMM component captures sequential dependencies in financial time series, modelling the probability of a current market state based on past observations, such as daily opening, high, low, and

closing prices. By generating **log-likelihood values** for each data vector, the HMM effectively partitions the input space into clusters of similar patterns, which then inform the creation of fuzzy rules. The **fuzzy inference system**, using Gaussian membership functions: $M_i(x_k) = e^{(-\frac{1}{2})(x_k - \mu_{fuzzy_{ik}})/(\sigma_{fuzzy_{ik}})^2}$ and TS-model consequents, translates these patterns into predictions of stock prices, while **gradient descent optimization** tunes both the linear and nonlinear parameters of the fuzzy rules to minimize prediction error. Experimentally, the HMM–fuzzy model outperformed conventional approaches such as ARIMA and shallow ANNs in terms of forecast accuracy (MAPE) and trading profit.

However, the model primarily relies on **historical price data** and technical patterns, without incorporating **microeconomic factors** (e.g., company earnings, debt, sector performance) or **macroeconomic indicators** (e.g., GDP growth, policy changes), which can significantly influence stock movements. Additionally, it does not process **unstructured textual data**, such as financial news, analyst reports, or social media sentiment, which are crucial for risk assessment and investment decisions. The computational cost is also high due to the iterative HMM and gradient descent optimization, limiting scalability for large datasets or frequent real-time updates. In contrast, a RAG-based system can integrate structured and unstructured multi-source data, providing a more versatile and context-aware approach to stock market forecasting.

2.3.5 Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs) are a framework that involves training two models in a competitive, zero-sum setup [49]. During this process, the generator attempts to produce synthetic data that closely resembles real stock market data, while the discriminator evaluates and distinguishes between the real and generated data. Through this adversarial training, the generator gradually learns to replicate the underlying distribution of the historical data so accurately that the discriminator can no longer tell them apart. This capability makes GANs particularly useful in stock market analysis, as they can simulate realistic price movements and capture complex patterns in financial time series, supporting forecasting and risk modelling.

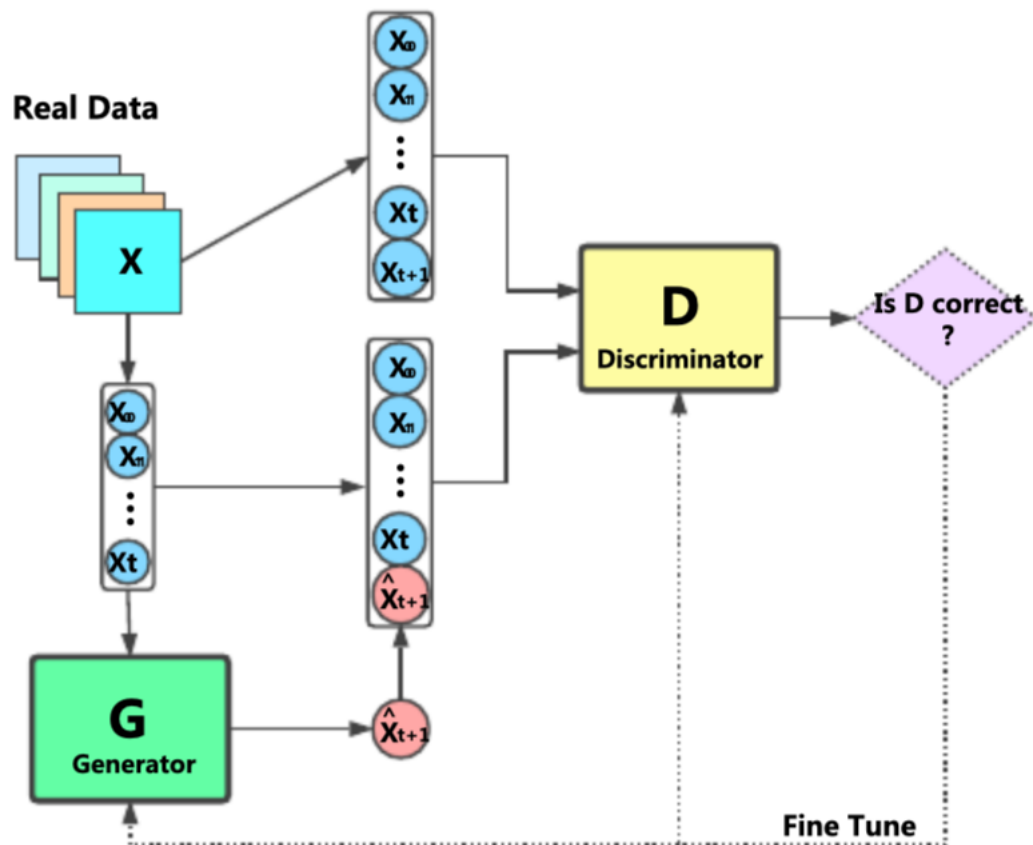


Figure 2.3.3: Architecture of GAN model. Source: [49]

The study in [49] introduced a hybrid GAN architecture for stock market prediction (see Figure 2.3.5), using a Long Short-Term Memory (LSTM) as the generator and a Multi-Layer Perceptron (MLP) as the discriminator. The model predicts daily stock closing prices by generating synthetic data for seven key features, including Open, High, Low, Close, Volume, Turnover Rate, and 5-day moving average. By adversarial learning the underlying distribution of historical stock data, practically, it allows the simulation of realistic market scenarios, which can support trading strategy testing and risk assessment.

As much as the model works well, the approach that was used in the study has a number of drawbacks. To begin with, it only uses past numbered datasets without considering the any outside qualitative parameters that external news, economy trackers, or policies could apply that impact the market. Secondly, the GAN zeroes in the predicting the daily closing figures without any mention of the intraday fluctuations or the high-

frequency trading data thereby restricting its real-world trading usefulness. Moreover, though the generator can replicate historical distributions, severe market phenomena or abrupt changes in the market could still be ignored by the model as it is bound by the parameters of the training data. Finally, this particular design, for instance, balancing the MSE with the adversarial losses, is tailored for predicting specific features which, in this case, the hyper-parameter tuning has to be adjusted. Mode collapse or loss of convergence during the adversarial training stage, which is often volatile, can in a number of instances happen.

2.4 Qualitative Data Analysis and Existing Systems

Qualitative data analysis has become increasingly vital in assessing a company's performance and investment potential, especially in stock market evaluations. Recent advancements underscore the potential of using LLMs for qualitative data analysis, particularly when assessing a company's future growth potential. By incorporating diverse qualitative data sources such as annual and quarterly reports, company news, and government policies, this approach aims to strengthen fundamental analysis and deliver more comprehensive investment advice. This strategy not only overcomes the limitations associated with a purely quantitative approach but also offers valuable insights into medium and long-term investment prospects [50].

2.4.1 Textual Content Sentiment Analysis

Traditionally, generative AI tools like Open AI GPT models have been used primarily for sentiment analysis, which is a core part of qualitative data evaluation [10]. This type of analysis involves interpreting and measuring sentiments found in textual content, such as news stories and posts from financial forums.

For example, in [9], a sentiment analysis model based on ChatGPT was combined with linear regression to predict short-term stock prices using daily returns data from the Centre for Research in Security Prices (CRSP) and news headlines. In their research, the authors asked ChatGPT if a given headline could be marked good, bad or irrelevant to a firm. Their results indicated that ChatGPT's sentiment scores correlated with subsequent stock performance, which was more predictive compared to traditional sentiment analysis. In their model, investors are classified as attentive, inattentive, or as driven by LLMs, with LLMs predictive as a function of model size and news

complexity. The authors propose a threshold model size which LLMs can predict returns profitably, and argue that with increased adoption, asset prices more closely reflect fundamentals, which LLMs predictively argue reduces predictability and enhances efficiency of the market. The primary contribution of the paper has to do with the theory; it attempts to move beyond the empirical approach of text mining and instead define in what instances and why LLMs are beneficial for the market. The sentiment scores generated by LLMs proved able to predict subsequent day returns better than the traditional sentiment analysis approach. This was particularly true for small stocks and was generated in the aftermath of negative news, supporting the idea that LLMs provide insights that the retail market underreacts to.

Beyond that, it is based on simplifying assumptions such as homogeneous risk aversion, CARA utility, and the use of textual signals only, ignoring real-world issues of LLM bias, prompt sensitivity, and hallucination. In contrast, this project seeks to address these issues by concentrating on the real-world use of LLMs through Retrieval-Augmented Generation (RAG) and LLMs linkage to policy documents, economic data, and financial news.

Similarly, another study in [10] explored the use of GPT-4 and BERT models for sentiment analysis on Stocktwits posts to predict the daily stock price movements of Apple and Tesla in 2017. To ensure the quality of textual inputs, preprocessing steps were applied differently for GPT-4 and BERT. For GPT-4, messages were cleaned by removing URLs, duplicates, and converting text to lowercase. BERT required additional cleaning, such as removing hashtags, cashtags, mentions, numbers, and emoticons, due to its limited contextual understanding. A key innovation in the system was the use of **prompt engineering**. The system achieved this by assigning GPT-4 a specific role (e.g., *financial analyst*), Instead of simply classifying messages as positive or negative, defining both the subject and analytical goal (e.g., evaluating advantages or disadvantages for Apple/AAPL), and incorporating “catch-all” elements to handle sarcasm, unrelated content, and competitor effects. Instead of rigid classifications, GPT-4 was instructed to output **probabilities**, reflecting the uncertainty of financial signals. This approach highlighted GPT-4’s ability to interpret nuanced financial contexts—for example, recognizing that “the president losing a fight with Apple” was advantageous to Apple, even if the surface sentiment appeared negative. While

structured formats (e.g., Python lists) reduced token use and improved comparability. Prompts also included fallback mechanisms to avoid verbose explanations and relied primarily on zero-shot prompting, balancing efficiency with clarity. This allowed the model not only to classify sentiment but also to capture whether the message implied an **advantage or disadvantage** for the company in question, and whether it was directly related to the firm or competitors.

The results indicated that GPT-4 was more effective than BERT at grasping subtle sentiments and complex financial language. However, the high deployment cost of GPT-4 is a significant drawback for practical applications [10].

U.S. Daily Stock News Summaries 1-1-2000 to 31-7-2020										
Classifier	R^2_{OOS} (%)		CER gain (%)			CER (%)		Sharpe ratios		
	<i>Sent.-Cont.</i>	<i>Sent.-Hist.</i>	<i>Sent.-Hist.</i>	<i>Sent.-Cont.</i>	<i>Cont.-Hist.</i>	<i>Sent.</i>	<i>Cont.</i>	<i>Sent.</i>	<i>Cont.</i>	<i>Hist.</i>
LM	0.02*	0.78***	5.786	1.36		7.349		0.706		
VA	0.11**	0.86***	4.683	0.257		6.246		0.621		
FB	-0.06	0.70***	6.496	2.07	4.427	8.059	5.989	0.77	0.597	0.247
TR	0.09*	0.85***	7.188	2.761		8.75		0.824		
GP	0.22**	0.97***	6.042	1.615		7.604		0.728		

Figure 2.4.1: Out-of-sample Analysis Results for Different Daily Sentiment Models. Source: [51]

Another study in [51] investigates the use of ChatGPT to categorize sentiment from summaries of daily U.S. market news and utilize that sentiment to forecast returns on the S&P 500 Index in the future. The authors demonstrate how ChatGPT can evaluate financial texts more skilfully than a financial analyst and how its sentiment classifications, which are consistent with behavioural finance theories, may forecast future market reversals. Additionally, they discover that ChatGPT-derived sentiments outperform other models, such as BARD and conventional sentiment research techniques in terms of economic performance, as shown in Figure 2.2.1. However, the system's reliance on sentiment classification introduces certain limitations, such as oversimplifying complex economic narratives into positive or negative signals and focusing narrowly on short-term return prediction. Prospective users should be aware that they will probably be totally dependent on developer transparency of the specifics of the model as well as any potential restrictions and biases, including those brought about by moderation [51].

LLMs perform sentiment analysis on textual data, enabling experts to gauge market sentiment and public opinion, which are crucial for evaluating stock performance. Nevertheless, these studies are often limited to short-term forecasts and rely on single data sources, without integrating a deeper fundamental analysis.

2.4.2 Embedding-Based Financial Report Analysis: GPT-InvestAR

Additionally, [52] introduced **GPT-InvestAR**, a tool designed to leverage ChatGPT for analysing annual reports. This specific tool extracts fundamental financial figures and metrics alongside the overall sentiments reflected in the company's performance from the reports and provides them in structured datasets. Answering the financial questions and assigning confidence scores as features was left to a separate downstream machine learning model to which GPT-3.5-Turbo was attached. All *mpnet-base-v2 embeddings* in conjunction with cosine similarity allowed the system to retrieve specific pieces of the reports as context for the LLM. These pieces were finely retouched segments. The qualitative text approach to financial forecasting was proven effective and the competitive edge was gained by the model that was developed with the set of features from the machine learned pipelines reinforced with the features boosted by qualitative alphas that over the S&P 500 index return predictions. The gap which was exposed is the use of single data validation in the form of annual reports is not sufficient and a broader qualitative cross data source multilevel text analysis is needed. It is also expensive, and API calls are costly as they work with data that is oversimplified and scores that are reduced with a more complex narrative to the market which are the bounded predictions of the regression model.

2.4.3 Domain-Specific Instruction Tuning: InvestLM

The framework proposed in the study [53], InvestLM, demonstrates the potential of domain-specific instruction tuning to significantly enhance the performance of large language models in financial applications. A major strength lies in the carefully curated instruction dataset, which draws from authoritative sources such as CFA exams, financial textbooks, SEC filings, and academic journals. This ensures that the model is trained on high-quality, domain-relevant data rather than generic corpora, leading to outputs that are better aligned with expert reasoning and grounded in financial logic. Furthermore, the use of LoRa for efficient fine-tuning and Linear Rope Scaling to

extend context length strengthens the model’s capacity to handle lengthy financial texts, which is critical for tasks like analyzing earnings reports or disclosures.

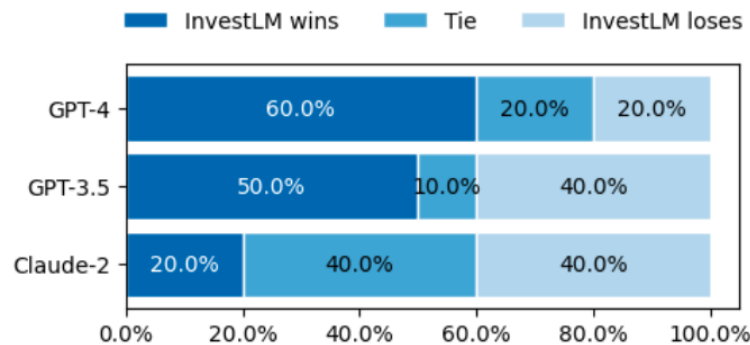


Figure 2.4.2: Expert Evaluation Results of InvestLM. Source: [53]

Based on Figure 2.4.2, the domain expert assessment confirms the approach. InvestLM frequently equals and even exceeds the capabilities of both GPT-3.5 and GPT-4, particularly praising the effectiveness of domain focus. Still, the approach described suffers some deficits. The amount of data collected is still considerably low (approximately 1,300 data points), which puts the scalability and coverage about the large domain of finance in question. Also, the manual curation, which does enhance the quality of the data set, also gives rise to the potential of biases and lack of responsiveness to ever-changing financial situations. In summary, the approach captures a pivotal element which brings the step in the direction of developing tailor-made financial LLMs, but the focus for the subsequent work should be on datasets of greater volume, hybrid assimilation of general instruction LLMs, and continual revision in relation to the ever-changing market.

2.4.4 Open-Source Data-Centric Framework: FinGPT

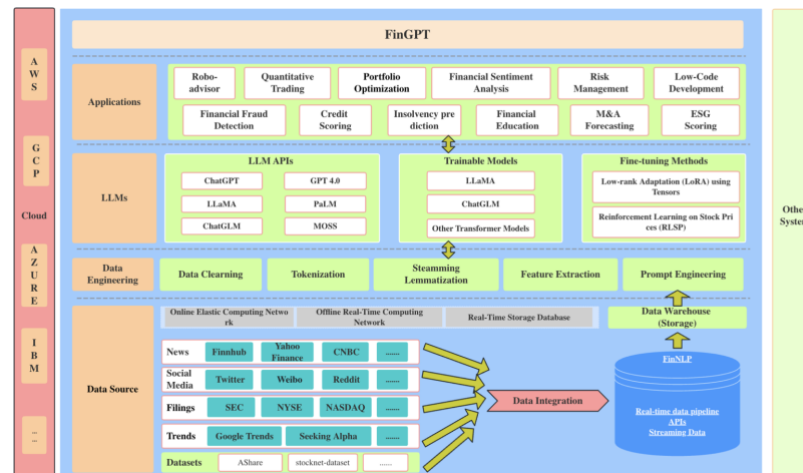


Figure 2.4.3: Framework of FinGPT. Source: [54]

FinGPT in the paper [54] is an open-source framework developed to apply large language models (LLMs) within the financial sector. The solution involves the integration and comprehension of dynamic data sources such as the data engineering real time system for financial news. The pipelines clean and tokenize data as well as sentiment extraction from the data. The suggestion of efficient finetuning: LoRA and Reinforcement Learning of Stock Prices suggests that the system is roughly as efficient and flexible as large proprietary models e.g. BloombergGPT. It's unique simultaneously sentiment and reliable filing archives are complemented with real time data on financial instruments arches for almost any time. The system has limitations with social data. The sheer volume of data from social sources is too imprecise and unstable leading an extremely low level of assurance. The problem with proxy signals for training is the oversimplified training complexities of financial markets. It's reasonable to expect that the level of FinGPT performance is affordable having said that is a direct challenge to models such as BloombergGPT. Coupled with the estimated requirements for systematic supportive updates and maintenance in relation to rapidly shifting financial ecosystems remains a substantial challenge.

2.4.5 Portfolio Optimization with LLMs

A noteworthy development is the incorporation of LLMs into portfolio optimization [11]. The study demonstrated that large volumes of unstructured data, such as financial accounts, news stories, and market reports, can be processed and analysed by LLMs

with ease. Within textual data, LLMs can identify emerging trends and nuanced market attitudes, providing a more comprehensive understanding of potential risks and possibilities. Investors can optimize their portfolios more thoroughly by merging quantitative data with LLMs' qualitative opinions. In addition to strengthening conventional models, this integration gives investors a more knowledgeable and adaptable approach, improving their capacity to handle market volatility.

2.4.6 Market Trend and Scenario Analysis

The important advancement in the use of artificial intelligence in financial forecasting has been made with the integration of GPT-4o mini's capabilities into market trend analysis. NLP is being used extensively in market forecasting as a supplement to conventional quantitative analysis techniques [9]. Making accurate and timely predictions regarding market movements depends heavily on NLP's ability to quickly evaluate massive volumes of text and extract pertinent information.

One of the strengths of LLMs is their ability to generate hypothetical scenarios based on existing data and trends, aiding experts in visualizing potential futures and considering different outcomes. [11] highlighted how LLMs can produce and analyse various scenarios, helping investors anticipate market movements. Their natural language processing (NLP) capabilities facilitate the summarization of qualitative data, making it easier to identify key patterns and trends [11].

2.4.7 Decision Support Systems: CustomGPT

LLM-driven stock analysis tools help mitigate cognitive biases in expert judgment by providing data-driven insights that challenge assumptions and highlight overlooked factors. The study in [55] compares a vanilla ChatGPT with a domain-conditioned **CustomGPT** as decision-support tools for forecasting competitions and judgmental adjustments.

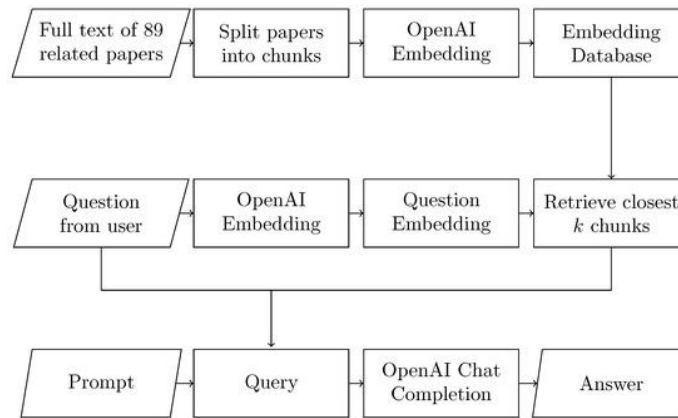


Figure 2.4.4: The process of how CustomGPT addresses user questions. Source: [55]

With reference to the RAG pipeline as shown in Figure 2.4.4, document chunking, embeddings, nearest-neighbour retrieval, and context enriched prompts, the authors [55] assessed accuracy against the ground truth. Findings indicate that CustomGPT as compared to the base model achieved a better score with fewer hallucinations, precise fact retrieval, and more narrowed directional procedures. This confirms the RAG approach as well as domain specific corpora for decision support usage. Some of the more apparent strengths of the model include substantial improvement in reliability of facts offered, more actionable process suggestions, and thoroughness of evaluation. Limitations of the study include the residual hallucinations, bounded corpus more easily accessed, maintenance costs as well as modified edge uncertainties and gaps in provenance. Most importantly, the paper devoted the most time addressing losses of factual data, rather than economical usefulness or performance of the model with hostile or out of corpus supplies. Concerning the financial uses of LLM for fundamental analysis, the study confirms embedding retrieval over curated out of corpus sample such as policy documents, central bank releases and 10-Ks, to enhance precision and interpretability. This simultaneously emphasizes the need for provenance and calibration, frequent structured index updates, and assure LLM responses as features of explainable complexity rather than primary trading signals.

2.5 Strengths and Weakness of Previous Studies

Conventional financial models, which analyse historical financial data and provide measures of profitability, valuation, and performance, such as discounted cash flow (DCF) models and ratio analysis, provide strong quantitative insights in assessing stock growth [1], [2]. These models are popular and well-established; they provide forward-

looking valuation insights and facilitate comparison with industry benchmarks. Studies in highlight their limitations, including their reliance on past data that might not accurately reflect future market conditions and their high sensitivity to assumptions that could result in valuation differences.

Quantitative data analysis, leveraging machine learning algorithms like artificial neural networks (ANNs) and recurrent neural networks (RNNs), enhances pattern recognition and short-term predictions based on historical data [45], [46]. Recent advancements, such as Generative Adversarial Networks (GANs), address some limitations by generating synthetic data, thus improving model robustness [49]. Despite their strengths, these methods can suffer a lack of transparency in their outputs and face issues with slow convergence and getting trapped in local optima [56].

Large Language Models (LLMs) bring significant advantages in qualitative data analysis by providing detailed sentiment analysis and generating scenarios based on textual data [45], [46]. They enhance expert judgment forecasting by integrating diverse qualitative inputs and facilitating scenario planning and decision support [11], [57]. However, their effectiveness depends on the quality of input data, and challenges remain in interpreting AI-generated predictions. While LLMs are effective in identifying market sentiments, they are not suited for direct computational tasks but rather serve as auxiliary tools that enhance existing quantitative models [11]. Additionally, there is a risk of over-relying on AI insights without sufficient human expertise, which may undermine the robustness of qualitative forecasts.

2.6 Proposed Solutions

To overcome the limitations stated above, this project proposes an LLM-driven stock analysis system integrating LLM, GPT-4o mini, with an AI Agent Retrieval-Augmented Generation RAG, and relevant domain knowledge. Unlike traditional models which focus on backward-looking metrics, or machine learning models which fail to take into consideration other relevant factors, this approach uses qualitative data. It evaluates a wide range of materials, such as news articles and government policy statements, to offer an evolving and holistic perspective on a company's growth potential, risk factors, and competitive market position. Employing state-of-the-art Natural Language Processing (NLP) technologies, the system classifies structured and

unstructured data to capture subtle yet critical market information which other models fail to identify.

The proposed system advances current practices in three key areas. First, it overcomes the rigidity of traditional financial analysis. While models such as DCF and ratio-based assessments offer forward-looking insights grounded in historical data, they fall short in accounting for sudden market fluctuations or qualitative developments [41], [42]. The integration of real-time qualitative data into this system provides a more agile and holistic assessment. For example, even if a DCF model undervalues a stock due to conservative forecasts, this tool can incorporate recent positive regulatory changes or executive sentiment from financial disclosures, refining its evaluation.

Second, it addresses the limitations of machine learning-based quantitative analysis. Although models like ANNs and RNNs are proficient in identifying statistical patterns [45], [46], they lack the capacity to interpret exogenous variables such as shifts in government policy or investor sentiment. In contrast, the proposed system augments these models by incorporating textual analysis via GPT-4o mini, enabling the inclusion of qualitative insights like regulatory updates or media sentiment. This approach, grounded in real-world context rather than synthetic data generation as seen in GANs [49], enhances prediction accuracy and reliability.

Third, the system significantly enhances qualitative analysis through the innovative use of LLMs. Existing applications, such as sentiment analysis on platforms like Stocktwits or corporate annual reports [10], [52], often rely on a single data source, making them vulnerable to bias and lack of contextual relevance [51], [57]. This project mitigates these issues by utilizing RAG to integrate multiple data sources—financial disclosures, macroeconomic indicators, and regulatory news—ensuring more accurate, timely, and context-sensitive insights. RAG also improves upon traditional LLM applications by cross-referencing historical data with current events, reducing the likelihood of outdated or irrelevant predictions.

The uniqueness of this system lies in its tailored integration of diverse data pipelines optimized for stock analysis, setting it apart from general-purpose LLM applications such as ChatGPT or Gemini. It combines real-time policy monitoring, sector-specific analysis, and sentiment evaluation to offer a multi-dimensional understanding of stock

valuation drivers. This framework not only improves forecasting precision but also democratizes access to sophisticated financial tools, enabling retail investors to benefit from insights typically reserved for institutional actors.

In addition, to foster trust and transparency, the system will implement explainability features to show how key predictions, such as sentiment scores or market outlooks, are derived. For instance, if a particular stock is predicted to increase in value due to positive policy changes, the system will provide related news articles or policy changes influencing the prediction, thereby enhancing user confidence. Furthermore, to ensure that the system provides up-to-date information, it will automatically pull new financial data, news, and government policy changes at regular intervals. This will be achieved through cron jobs, refreshing data from sources like yFinance, and news APIs every day, depending on the need for freshness in the data.

In a complex and rapidly evolving financial landscape, where traditional models and siloed AI tools often fall short, the proposed system represents a significant advancement. By addressing the inherent limitations of existing methodologies which is their dependence on static historical data and exclusion of qualitative market signals, it contributes to more strategic and informed investment decision-making. Additionally, by making institutional-grade analytical tools more accessible, it fosters a more inclusive and equitable financial ecosystem, ultimately advancing the field of stock market analysis.

CHAPTER 3: SYSTEM METHODOLOGY/APPROACH

This section presents an overview of the proposed solution, including system architecture diagrams, user requirements, use case diagrams and descriptions, as well as activity diagrams, to provide a comprehensive understanding of the project's approach. It also outlines the development methodology and the timeline for completing the project.

3.1 System Architecture Diagram

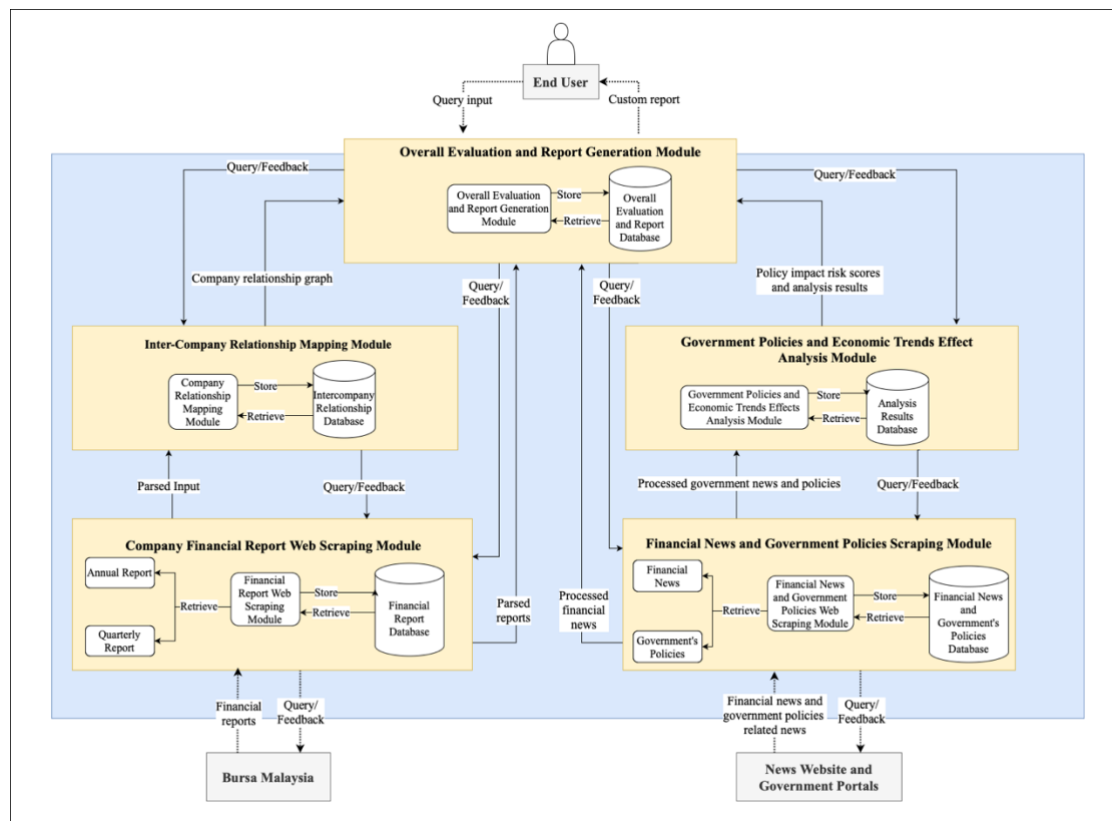


Figure 3.1.1 System Architecture Diagram of the Overall System

This system in Figure 3.1.1 is designed as a collaborative, modular framework where each component enhances the performance and accuracy of the overall stock analysis. reflect not only numerical trends but also external macro-level influences. The system aggregates data from four key sources: financial report, intercompany relationship, financial news and government policy documents. These diverse data types provide a strong foundation for evaluating a company's growth potential and market position.

The policy and economic trends analysis module enhances the final recommendation module in this project by the addition of qualitative analysis of government policies and

economic occurrences to make certain that the predictions are properly grounded. It will depend on the information that will be captured by other modules that are concerned with the web scraping of the modules of the government policies by setting a query on the previously mentioned modules. In doing so, it improves the comprehension of how economical and regulatory changes affect different industries. For example, when the final module inputs specific queries such as a “company name” the policy and economic trends analysis module is tasked with identifying relevant policies and economic trends that could affect that company. It then analyses these factors and provides the final module with a comprehensive report on how these external elements might influence the company’s performance. Similarly, if the input is a “policy name” or an inquiry about a particular “economic trend” the module will deliver an assessment of the potential impacts on the market or industry in question.

Similarly, other supporting modules like inter-company relationship mapping and data scraping for financial and policy content contribute essential context that strengthens the quality of analysis across the system. As each module feeds into the next, improvements in one area, such as more accurate entity mapping or expanded policy datasets cascade into more reliable investment insights.

A key advantage of this architecture is its **scalability** and **built-in feedback loops**. The system is intentionally built with flexibility to accommodate new modules, data sources, or analytics tools without disrupting its core functionality. Whether by incorporating new financial news APIs, enhancing analysis, or expanding to international policy feeds, future upgrades can be seamlessly integrated. Other than that, it allows feedback loops where the accuracy of insights generated by the final module can be evaluated and used to adjust earlier stages of processing. For example, if the investment recommendations consistently underperform or misinterpret trends, this feedback can inform refinements to data preprocessing and policy impact interpretation. This interconnected and forward-compatible design ensures the platform can grow with evolving user needs and technological advancements, creating a durable, intelligent tool that continues to improve over time. This makes it an asset for retail investors who are seeking for holistic, data-driven guidance.

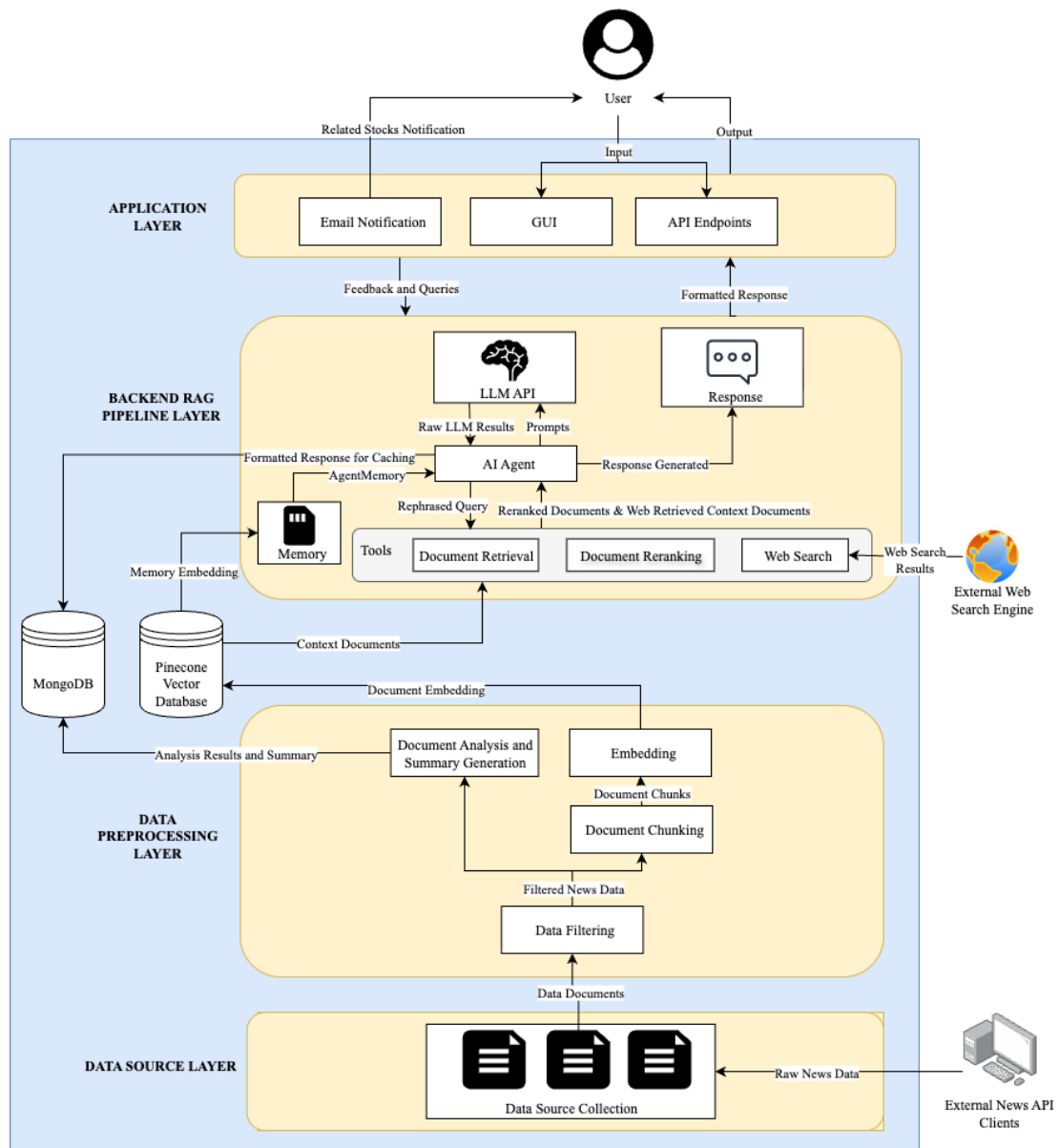


Figure 3.1.2: System Architecture Diagram of the Policy and Economic Trends Analysis Module

The system architecture is structured into four layers, each designed to handle specific stages of data acquisition, processing, analysis, and user interaction to enable intelligent financial news and stock analysis. The **Data Source Layer** serves as the entry point for raw information, collecting data from data sources which is the financial news API (News Data.io). This layer ensures that incoming data is standardized into structured formats, allowing downstream components to efficiently process it. Next, the **Data Preprocessing Layer** applies a series of advanced data engineering techniques to prepare the information for LLM-driven analysis. This includes filtering and cleaning

to remove irrelevant or low-quality data, semantic chunking of large documents into smaller, contextually coherent segments, and generation of vector embeddings using pre-trained language models (text-embedding-small-3) to facilitate high-dimensional similarity search. In addition, automatic summarization, keyword extraction, and entity recognition are applied to enhance the semantic representation of documents. The processed outputs, including embeddings, summaries, and analytical metadata, are stored in **MongoDB** for structured access and **Pinecone** for vector-based similarity retrieval, enabling efficient and scalable semantic search.

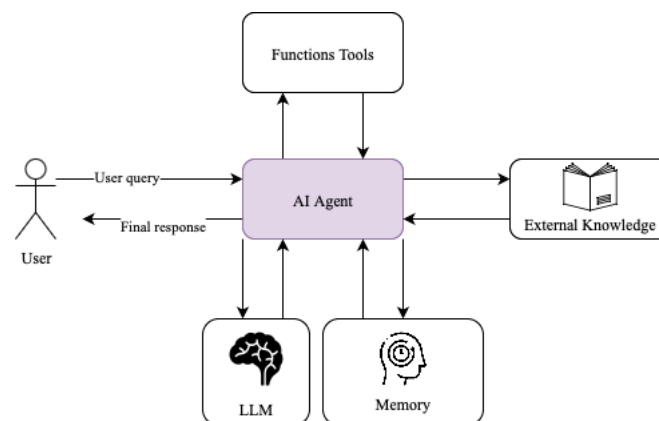


Figure 3.1.3: High-level AI Agent Architecture with RAG Elements

Fundamentally, the Backend Retrieval-Augmented Generation (RAG) Pipeline Layer has an AI agent which focuses on the user question disambiguation for the purpose of retrieval optimization, manages the orchestration of integrated tool use alongside the memory modules, and manages the retrieval of documents along with relevance retrieval and reranking (cross-encoder) of MongoDB and Pinecone results. This layer also uses the external web search API (Google Customised Search Engine) for the purpose of augmentation of the retrieved knowledge with the knowledge which is more recent. The LLM API is utilized for the purpose of constructing contextualized prompts and responses by integrating LLM retrieved embeddings and summaries along with web documents, while the caching mechanisms ensure efficient handling of repeated query submittals. Specialized techniques such as context window management and semantic reranking are utilized to maximize relevance and accuracy of the insights generated. The Application Layer has the user interface which HTML interfaces, RESTful API endpoints, and email users to collect submissions. This layer also provides users with formatted responses, stock insights, and notifications which are

captured for later use as feedback to make iterative improvements on the AI agent and retrieval system. These layers constitute a complete architecture which processes the user input and provides sophisticated reasoning on the retrieved documents utilizing LLMs for financial evaluation and guidance.

3.2 Use Case Diagram and Description

3.2.1 Use Case Diagram

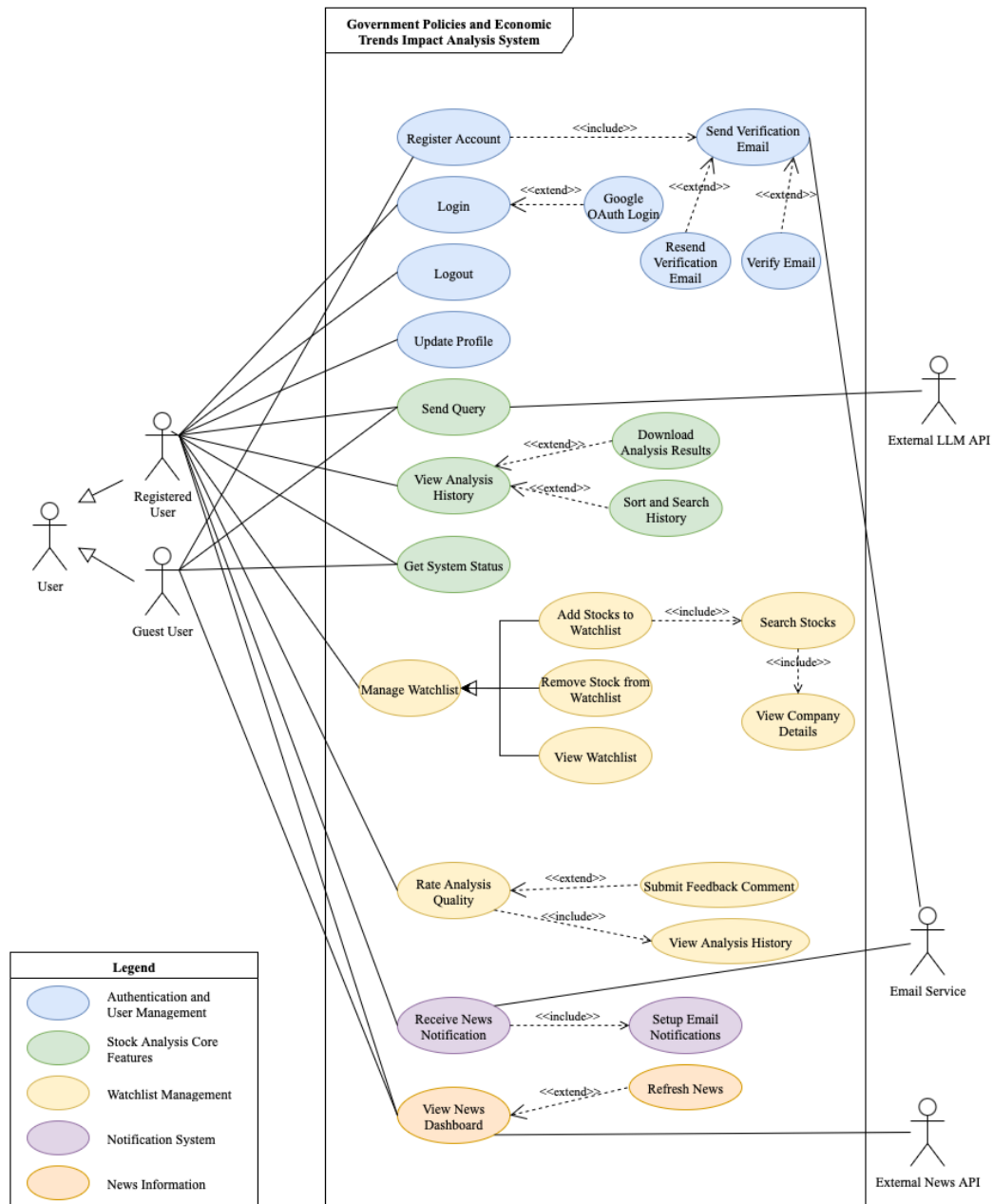


Figure 3.2.1: Use Case Diagram of Government Policies and Economic Trends Analysis Module

Figure above (Figure 3.2.1) presents the use case diagram which models the interaction between the primary actor (user and external APIs) and the system.

The system supports three main actors: Registered Users, Guest Users, and External APIs. Registered Users can access the full set of features, including advanced stock analysis, personalized watchlists, and feedback submission. Guest Users have limited access, which allows them to try the system's core functions before registering. External APIs act as supporting actors who supplies stock market data, financial news, and email notification services.

- **Authentication and User Management**

Next, authentication and user management form the basis of the system. Users can register and log in through email/password, Google OAuth, or email verification. These use cases ensure secure access while offering flexibility in authentication methods.

- **Stock Analysis Core Features**

The central feature is stock analysis, where users can assess risk, search for securities, and view historical analysis records. The “Send Query for Analysis” use case incorporates Retrieval-Augmented Generation (RAG) to generate context-aware analysis by combining multiple data sources such as financial data, news, and sentiment signals. To support ongoing improvement, a feedback feature enables users to rate analysis results, report issues, and share suggestions. This feedback loop helps refine the system and improve user experience over time.

- **Watchlist Management**

Watchlist management allows users to add or remove stocks, monitor real-time data, and set notification preferences. This supports investors in keeping track of multiple securities and receiving timely updates.

- **News and Notification System**

The system also provides daily news updates and customizable email alerts, matched to user interests and watchlist holdings. This ensures users stay informed about market movements and relevant developments.

Also, the use case relationships highlight the internal dependencies. For example, register account including email verification, while watchlist management depends on searching and selecting stocks. Extend relationships represent optional actions, such as giving detailed feedback after viewing analysis results.

3.2.2 Key Use Case Description

Table 3.2.1 UC001 Register Account Use Case Description

Use Case ID	UC001	Version	1.0
Use Case	Register Account		
Purpose	To allow a new user to create an account and validate their email before accessing the system.		
Actor	Guest User (any visitor who wants to register)		
Trigger	A user selects the “Register” link from the welcome page.		
Precondition	The system is online, and the registration form is accessible.		
Scenario Name	Step	Action	
Main Flow	1	User clicks the “Register” link on the welcome page.	
	2	System displays the registration form with fields such as username, email, and password.	
	3	User fills in the registration form and submits it.	
	4	System validates the input fields (e.g., email format, password strength).	
	5	System creates a user account in a pending verification state.	
	6	System sends verification email.	

Alternate Flow: Invalid Input	4.1.1	If input validation fails, the system highlights the invalid fields and prompts the user to correct them.
Alternate Flow: Email Already Registered	4.2.1	If the email is already registered, the system displays an error message prompting the user to login or use a different email.
Post Conditions	User account is created in pending verification state. Verification email is sent.	

Table 3.2.2 UC002 Login Use Case Description

Use Case ID	UC002	Version	1.0
Use Case	Login		
Purpose	To allow registered user to login to the system.		
Actor	Registered User		
Trigger	A user selects the “Login” link from the welcome page.		
Precondition	The system is online, and the login form is accessible.		
Scenario Name	Step	Action	
Main Flow	1	User navigates to the login page.	
	2	System displays the registration form with fields such as username, password with a “Remember Me” checkbox.	
	3	User fills in the registration form and submits it.	
	4	System validates the input fields (e.g., empty field).	
	5	System validates if “Remember Me” checkbox is ticked.	
	6	User clicks on “Login” button.	
	7	System validates the credentials against the stored authentication database.	
	8	System authenticates user and grant access to the system.	

Alternate Flow: Empty Field	4.1	If there is any empty field, system highlights it and prompts the user to enter them.
Alternate Flow: “Remember Me” checkbox ticked	5.1	If the “Remember Me” checkbox is ticked, system stores credentials in the browser for autofill in future sessions.
Alternate Flow: Invalid Credentials	7.1	If the username or password entered is incorrect, system display error message.
Post Conditions	User is successfully logged into the system	

Table 3.2.3 UC003 Send Query Use Case Description

Use Case ID	UC003	Version	1.0
Use Case	Send Query		
Purpose	To allow a user to submit a query to the chatbot for financial and policy analysis and receive a relevant, context-aware response.		
Actor	User (Registered User, Guest User)		
Trigger	A user enters a query in the chatbot interface and clicks the “Submit” or “Send” button.		
Precondition	The system is online, and the chatbot interface is accessible.		
Scenario Name	Step	Action	
Main Flow	1	User types an analysis query into the chatbot input field.	
	2	User clicks the “Send” button or presses Enter to submit the query.	
	3	The system validates the input and ticker format (e.g., ensures it is not empty and within character limits).	
	4	System checks for cache hit.	
	5	The AI agent receives the query and reformulates it for better retrieval accuracy.	

	6	AI Agent check agent memory for context from previous interactions.
	7	The system retrieves relevant documents from MongoDB and Pinecone using embeddings and vector search.
	8	AI agent optionally performs external web search to supplement retrieved documents.
	9	Retrieved documents are ranked based on relevance to the query.
	10	The LLM API generates a response using the ranked documents, external search results (if applicable), and user context from memory.
	11	The system formats and caches the AI response for display in the chatbot interface.
	12	The system delivers the results to the user via the chatbot.
Alternate Flow: Invalid Query	3.1	If the input query is empty or exceeds character limits, the system displays an error message prompting the user to modify the query.
Alternate Flow: Cache Hit	4.1	System returns cached results.
Alternate Flow: No Relevant Documents Found	6.1	If no relevant documents are retrieved from the database or external search, the system informs the user that no matching analysis is available and may provide general guidance or suggestions.
Post Conditions	<ol style="list-style-type: none"> 1. The user receives a formatted, contextually relevant response from the AI chatbot. 2. The query and response are logged in the system for reference, analytics, or memory updates. 	

Table 3.2.1 UC004 Manage Watchlist Use Case Description

Use Case ID	UC004	Version	1.0
--------------------	-------	----------------	-----

Use Case	Manage Watchlist	
Purpose	To allow users to manage their stock watchlist, including adding or removing stocks, viewing the watchlist, and updating watchlist settings.	
Actor	Registered User	
Trigger	User selects the “Watchlist” option from the application menu or dashboard.	
Sub-functions	<ul style="list-style-type: none"> • Add stocks to watchlist • Remove stocks to watchlist • View watchlist 	
Precondition	The user is logged in, and the watchlist interface is accessible.	
Scenario Name	Step	Action
Main Flow	1	User selects the Watchlist option from the application interface.
	2	System displays the watchlist interface with existing stocks.
Alternate Flow: Add Stocks to Watchlist	2.1.1	User searches for a stock using the search function.
	2.1.2	System displays search results with detailed stock information (price, market cap, pe/ratio, etc.)
	2.1.3	User clicks “Add” button to add the stock to the watchlist.
	2.1.4	System adds the stock to the watchlist database and update the display.
Alternate Flow: Remove Stocks from Watchlist	2.2.1	User selects a stock from the watchlist to remove.
	2.2.2	System prompts for confirmation.
	2.2.3	User confirms removal.

	2.2.4	System removes the stock from the watchlist and update the display.
Alternate Flow: Invalid Stock Search	2.1.1.1	If no results are found, system displays a message prompting user to refine the search.
Alternate Flow: Stock Already in Watchlist	2.1.3.1	If the stock already exists in the watchlist, the system alerts the user and prevents duplication.
Post Conditions	<ul style="list-style-type: none"> • Watchlist is updated according to user actions. • Stocks added, removed, or updated are reflected in the system and user interface. • User preferences and settings are saved. 	

Table 3.2.5 UC005 Rate Analysis Quality Use Case Description

Use Case ID	UC005	Version	1.0
Use Case	Rate Analysis Quality		
Purpose	To allow users to rate the quality of AI-generated analysis, submit feedback or comments and view the history of previous analyses.		
Actor	Registered User		
Trigger	User receives an analysis response from the chatbot and chooses to provide a rating or feedback.		
Precondition	The user is logged in, the generated analysis is accessible, and the rating/feedback interface is available.		
Scenario Name	Step	Action	
Main Flow	1	User receives an AI-generated analysis in the chatbot interface.	
	2	User clicks the Rate Analysis button or option.	
	3	System displays rating options (e.g., stars, numerical score) and optional comment field.	
	4	User selects a rating and optionally enters feedback or comments.	

	5	System validates the feedback fields and rating range.
	6	System saves the rating and comment, associating it with the specific analysis.
Alternate Flow: Invalid Feedback Data	5.1	If rating is not 1–5 or feedback_type is invalid, system returns an error and prompts the user to correct input.
Post Conditions	<ul style="list-style-type: none"> • User rating and feedback are saved in the system and linked to the corresponding analysis. • Analysis history is updated to reflect new ratings and comments. • System can use feedback to improve system responses. 	

Table 3.2.6 UC006 Receive News Notification Use Case Description

Use Case ID	UC006	Version	1.0
Use Case	Receive News Notification		
Purpose	To allow users to receive timely notifications about news, financial updates, and relevant stock alerts via email.		
Actor	Registered User		
Trigger	User subscribes to notifications or the system detects relevant news updates for stocks in the user's watchlist.		
Precondition	User is logged in, has a valid email address, and the notification service is active.		
Scenario Name	Step	Action	
Main Flow	1	User navigates to the Notification Settings page.	
	2	User enables email notifications .	
	3	User confirms settings and submits changes.	
	4	System saves notification preferences in the database (MongoDB).	

	5	System perform analysis on recent news.
Alternate Flow: Relevant News Detected	5.1	System generates a notification message summarizing the news or update.
	5.2	System sends the notification to the user's email.
Post Conditions	<ul style="list-style-type: none"> • User receives email notifications according to preferences. • Notification settings are stored and used for future news alerts. 	

Table 3.2.7 UC007 View News Dashboard Use Case Description

Use Case ID	UC007	Version	1.0
Use Case	View News Dashboard		
Purpose	To allow users to view a consolidated dashboard of news articles, financial updates, and stock-related information, with the ability to refresh for the latest news.		
Actor	Users (Registered User and Guest User)		
Trigger	User navigates to the News Dashboard or requests a refresh of news content.		
Precondition	User is logged in and the system is connected to external news sources.		
Scenario Name	Step	Action	
Main Flow	1	User navigates to the News Dashboard via the application interface.	
	2	System retrieves and displays the latest news articles from internal databases.	
	3	User clicks the Refresh button.	
	4	System requests latest news from External News API.	

	5	System processes and stores the retrieved articles in the database.
	6	Dashboard is updated with new news, highlighting recent updates.
Alternate Flow: No New News Available	2.1	If no new articles are retrieved, the system maintains the current dashboard content and may display a “No new updates” message.
Post Conditions		News dashboard displays the latest available news and updates.

3.3 Activity Diagram

This section presents eight activity diagrams representing flow of the key use cases of the system.

3.3.1 Register Account

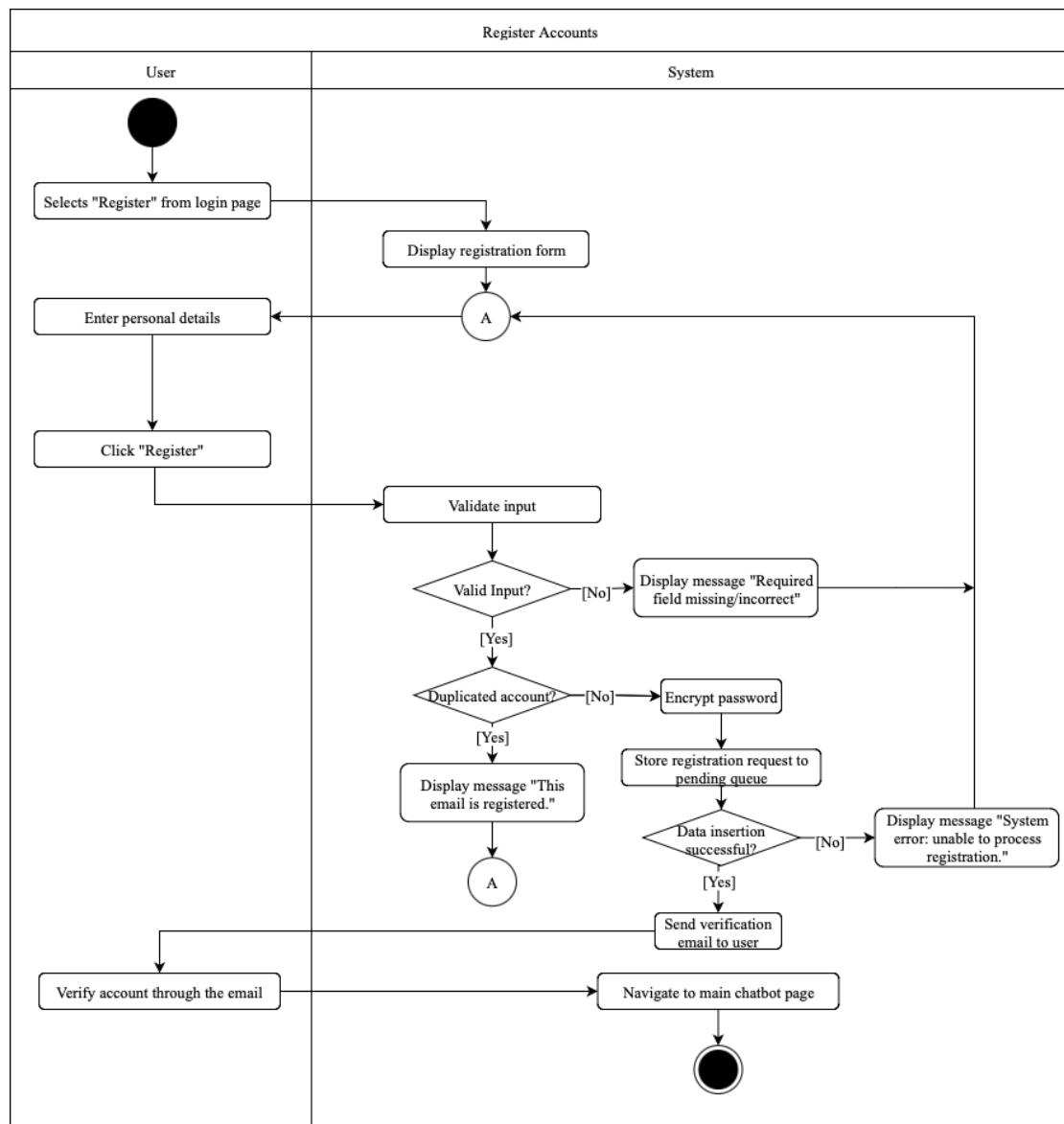


Figure 3.3.1 Activity Diagram of Register Account Feature

The registration process allows new users to sign up with either email/password or Google OAuth. Input validation checks email format, password strength, and duplicates. On successful registration, the system generates a token and sends a verification email with a 24-hour expiry.

3.3.2 User Login Page

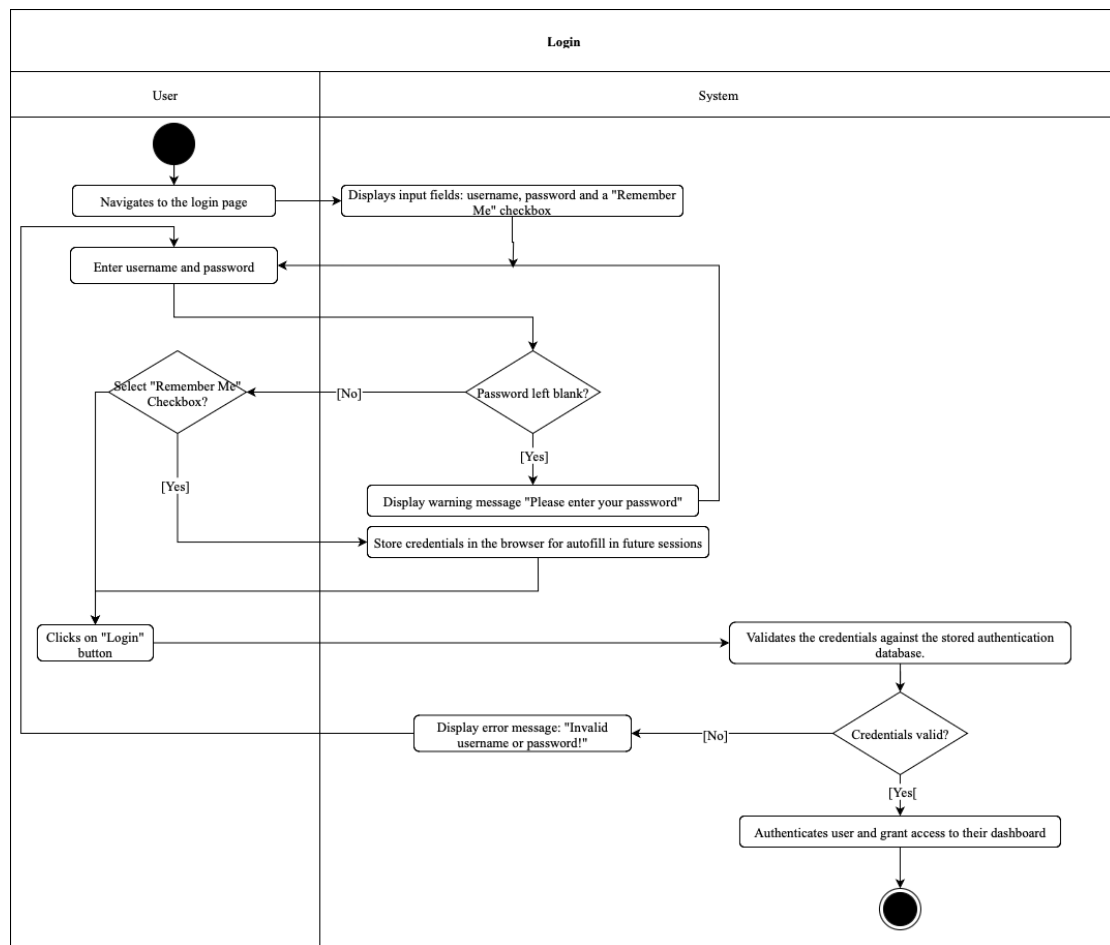


Figure 3.3.2 Activity Diagram of User Login Page

The login page offers a clean, responsive interface built with React and Tailwind CSS. Users can log in with email/password or Google OAuth. The page includes standard features such as password visibility toggle, “Remember Me” option, and real-time error messages for invalid credentials or locked accounts. Security measures like CSRF protection and input sanitization are also in place. Links are provided for account recovery, and users can switch between login and registration easily.

3.3.3 Data Fetching and Document Processing

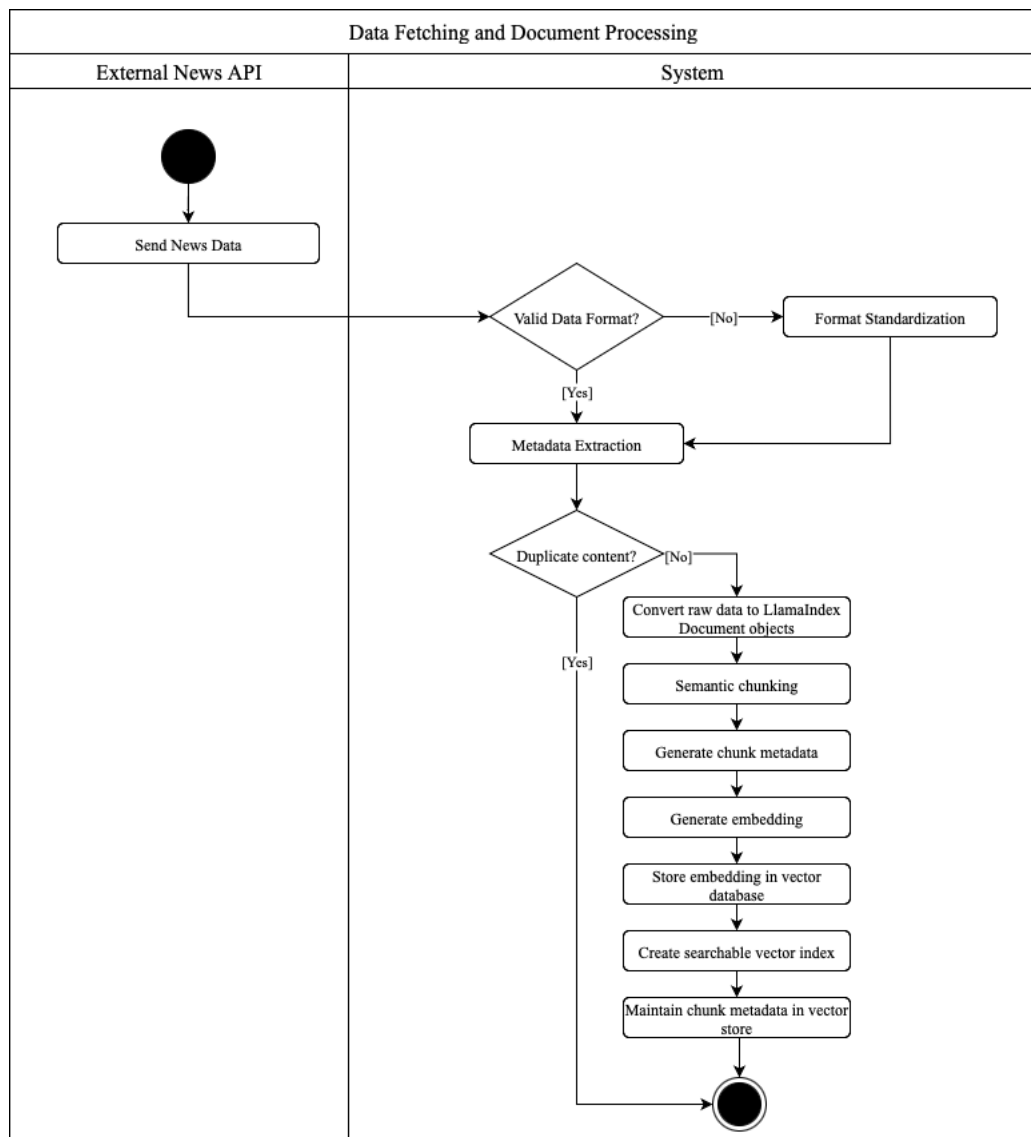


Figure 3.3.3 Activity Diagram of Data Fetching and Document Processing

The system collects data from several external APIs, including NewsData.io for news and yfinance for financial data. Data is validated before processing. Large documents are split into smaller chunks using *SpacyTextSplitter* (512 characters with overlap) for efficient embedding and vector search. Embeddings are generated with OpenAI's *text-embedding-3-small* and stored in Pinecone, along with metadata such as source, date, and relevance.

3.3.4 Send Query

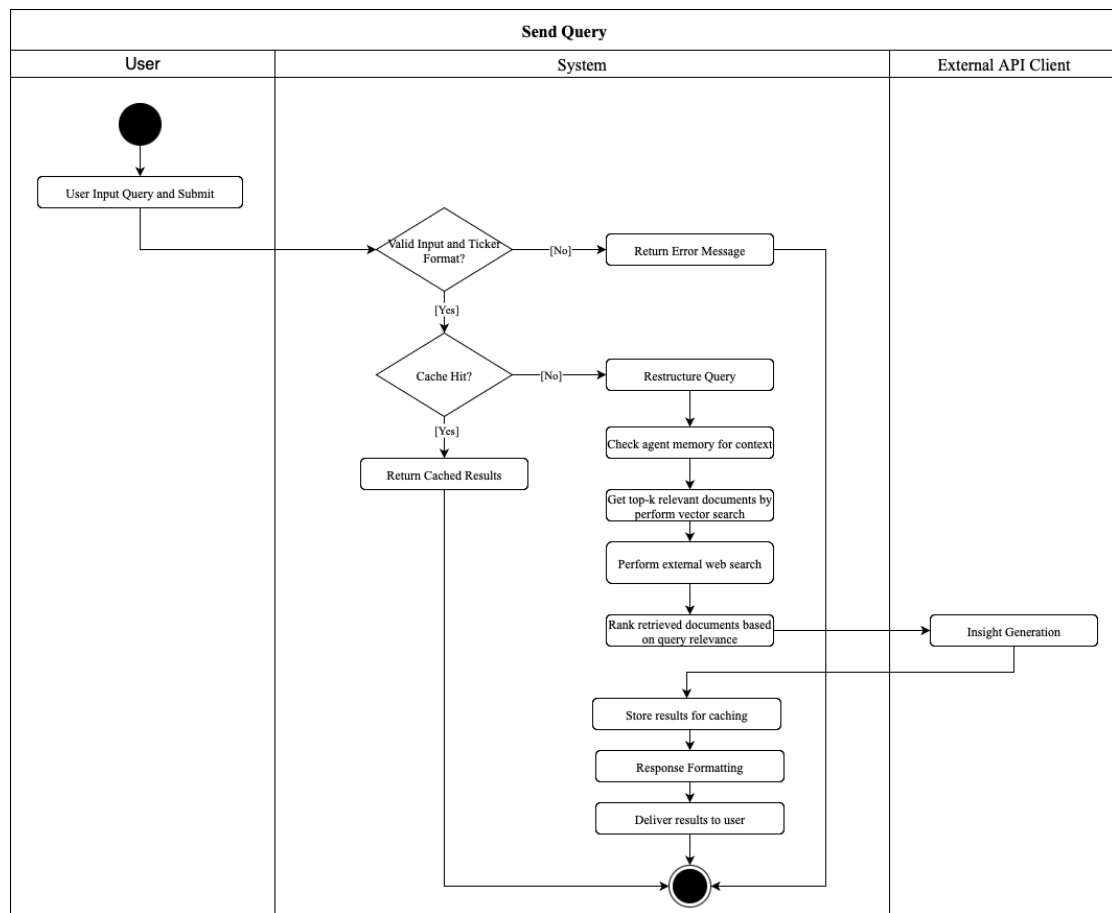


Figure 3.3.4 Activity Diagram of Send Query

The analysis flow powers the RAG pipeline. It begins by validating user queries, checking format and ticker symbols. If no cached analysis exists, the system performs a similarity search in Pinecone, retrieves relevant chunks, and supplements them with MongoDB content. The data is then passed to GPT-4o-mini, which produces a stock risk analysis including scores, summaries, and insights. Results are returned in JSON format, cached for reuse, and displayed to users with source references and confidence indicators.

3.3.5 Manage Watchlist

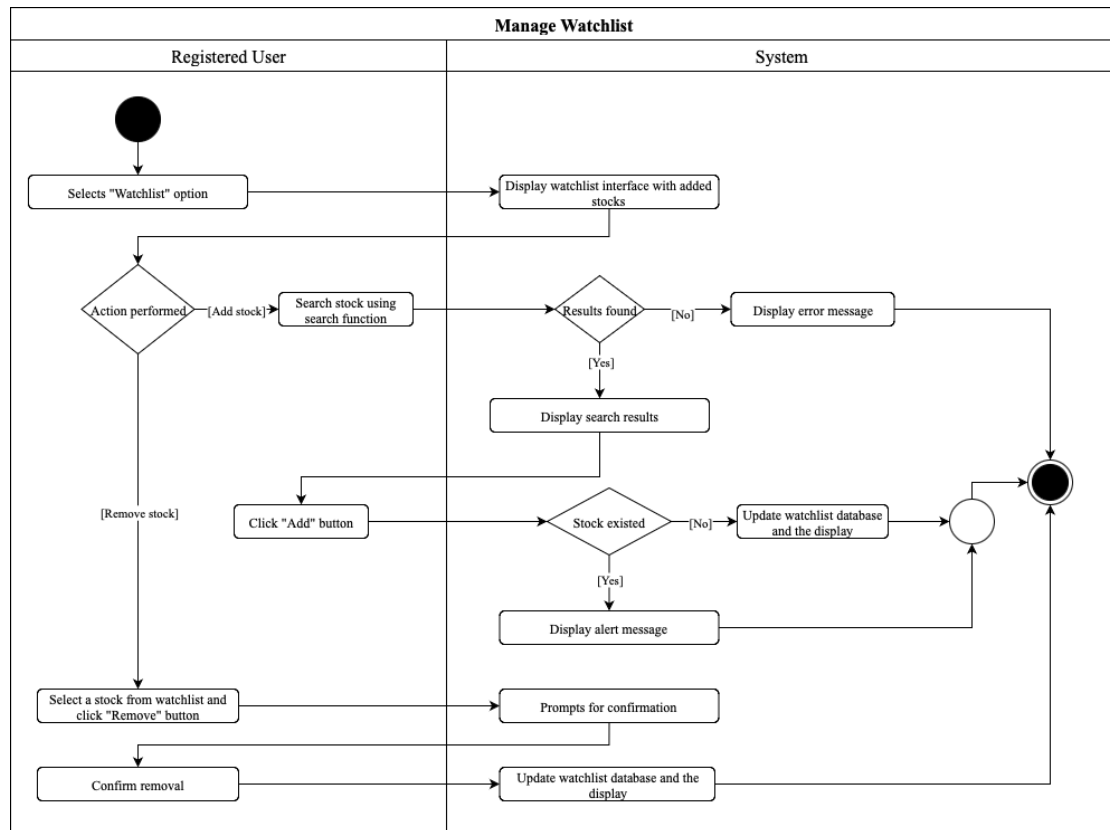


Figure 3.3.5 Activity Diagram of Manage Watchlist

In the “**Manage Watchlist**” activity diagram (see Figure 3.3.5), user interactions such as adding, removing, or viewing stocks are mapped alongside internal processes like searching for stocks, displaying company details, and updating watchlist preferences. Decision points, such as handling duplicate stock entries or invalid inputs, ensure clarity in workflow and error handling.

3.3.6 Rate Analysis Quality

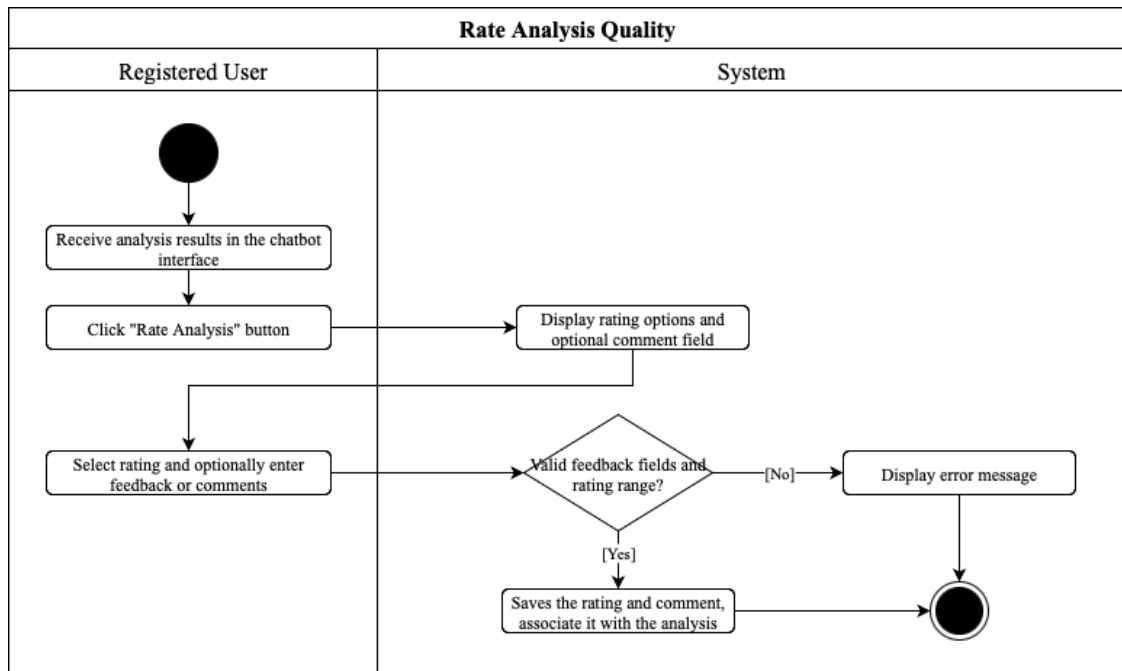


Figure 3.3.6 Activity Diagram of Rate Analysis Quality

The “**Rate Analysis Quality**” activity diagram emphasizes the feedback loop, showing users providing ratings and comments, system validation, storage of feedback in MongoDB. Alternate flows handle invalid feedback received from user side and display error message to prompt user to rectify the input format.

3.3.7 Receive News Notification

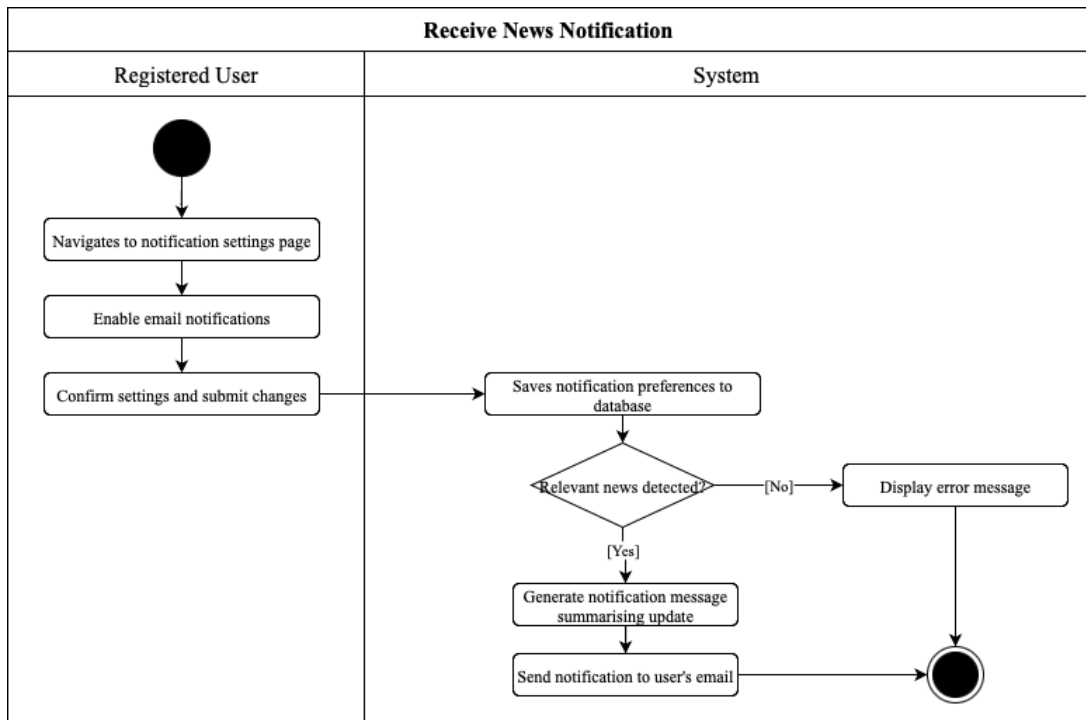


Figure 3.3.7 Activity Diagram of Receive News Notification

For the “**Receive News Notification**” use case, the activity diagram illustrates the process from setting up email notifications and preferences, validating email addresses, to detecting relevant news and delivering alerts. It also highlights alternate flows for invalid emails or absence of new news.

3.3.8 View News Dashboard

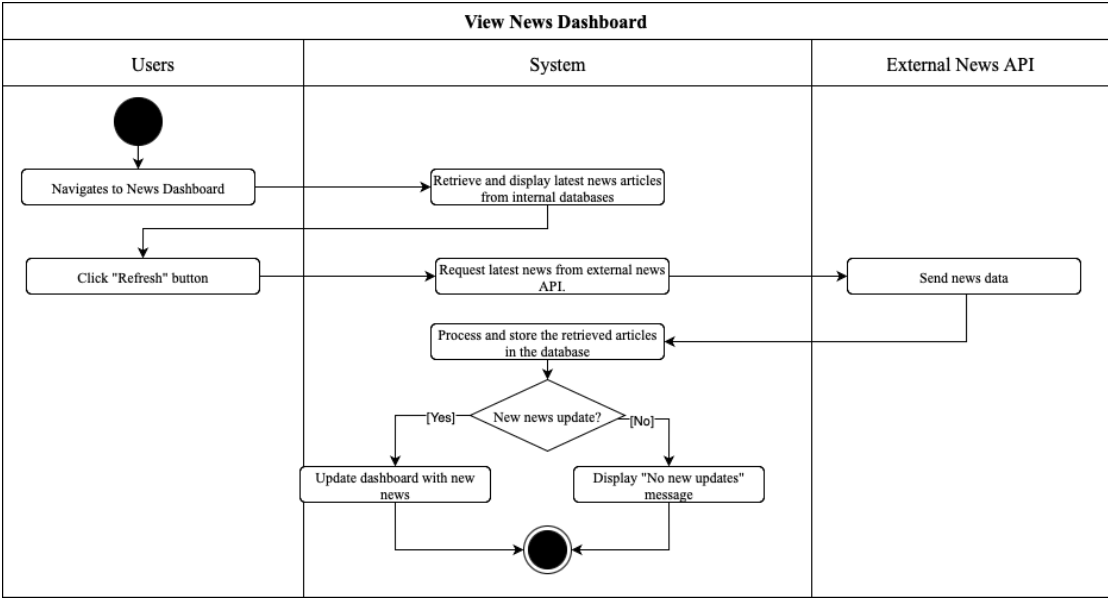


Figure 3.3.8 Activity Diagram of View News Dashboard

Finally, the “**View News Dashboard**” activity diagram captures the user’s interaction with the dashboard, including retrieving stored news, refreshing content via external news APIs, handling API failures, and presenting the latest articles. The diagram emphasizes the dynamic connection between internal storage and external sources for timely updates.

3.4 Methodologies and General Work Procedures

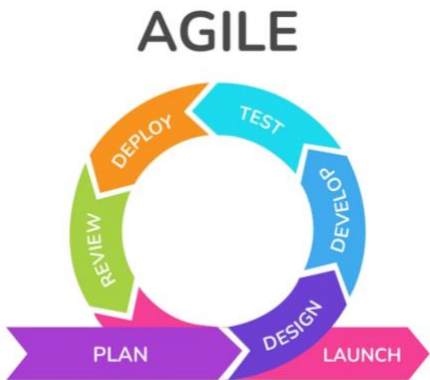


Figure 3.2.1 Agile Development Process. Source: [58]

As presented in Figure 3.2.1 [58], this project adopts the **agile methodology** to guide the development of the system. Agile methodology is particularly suitable for this project due to the fast-paced evolution of large language models (LLMs), APIs, and

financial data technologies. New tools, capabilities, or insights may emerge at any point, and Agile offers the flexibility to revise the project scope, integrate emerging technologies, or reprioritize features without disrupting the overall development progress. The project is developed in iterative sprints, allowing continuous integration of improvements such as better insight generation, faster response times, and refined trend analysis. Each sprint allows the functionality to be evaluated in real-time by incorporating feedback from the users and address technical challenges early. This ensures the system to remain up to date, scalable, and aligned with its objectives of delivering timely, accurate, and relevant insights for retail investors.

The project is structured into six main phases: **Planning**, **Design**, **Development**, **Testing**, and **Deployment Simulation and Review**, with each phase contributing directly to the completion and validation of specific system components. In the **Planning** phase, key requirements are defined. This includes risk scoring logic, sentiment analysis strategy, and integration with MongoDB and Pinecone for data storage and vector indexing. The **Design** phase then shapes the system's architecture, detailing how components interact and data flows through the system. Next, the **Development** phase involved implementing core modules such as the JSON-based webhook input handler, LLM-driven insight generator and financial news summarizer, all integrated through Flask. During the **Testing** phase, unit and integration tests were conducted to verify system reliability, including how accurately the LLM summarizes input data and generate insights. Testing is integrated into every sprint to evaluate the system's performance and functionality to allow early detection and correction of issues. In the **Deployment Simulation** phase, the system was tested under simulated API queries to validate response time, insight relevance, and system scalability to ensure that the outputs such as risk scores, summaries, and investment suggestions are accurate and timely. Finally, the Review phase gathers feedback to refine the system, completing the cycle. The iterative loop (**Planning** → **Design** → **Development** → **Testing** → **Deployment Simulation** → **Review**) repeats until the system meets the defined performance metrics, ensuring the delivery of a reliable, relevant, and scalable tool for retail investors.

CHAPTER 4: SYSTEM DESIGN

This chapter outlines the technical approach used to transform policy and company documents into a structured and searchable knowledge base for stock analysis from a top-down perspective. It detailed the system architecture and modular pipeline that enable efficient data processing and insight generation.

4.1 System Block Diagram

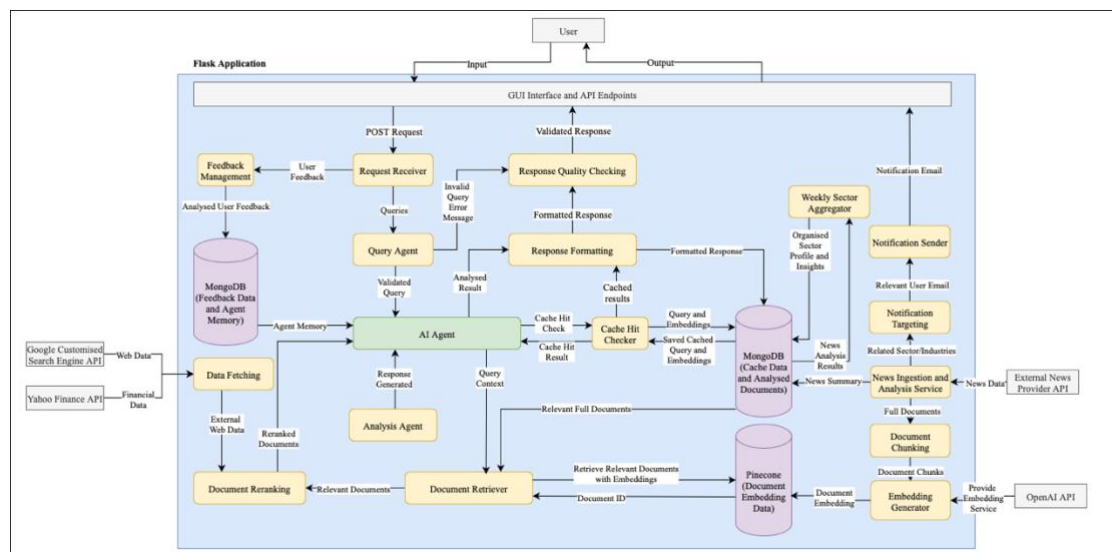


Figure 4.1.1: Block Diagram of the Government Policies and Economic Trends Impact Analysis Module

According to Figure 4.1.1, the workflow begins with the **User**, who interacts with the **API Interface** or **GUI Interface** (Flask Application) via POST requests. This request is then validated for correctness and format before being passed to the workflow engine. Depending on whether the query has been processed recently, the system either retrieves cached results from MongoDB or fetches fresh data from external APIs and document stores. Retrieved documents are then passed through a RAG pipeline and reranked, where the AI Agent synthesizes relevant information and generates insights. Retrieved documents are then passed through a **RAG pipeline and reranked**, where the **AI Agent**—comprising a memory module, a knowledge base, and an LLM as its core “brain”—synthesizes relevant information and generates insights. The memory module allows the agent to retain contextual information from past interactions, the knowledge base provides structured domain-specific data for reference, and the

LLM performs reasoning, summarization, and analysis to produce comprehensive results.

After the analysis results are generated, the system creates **document embeddings** for the processed data. These embeddings are stored in **Pinecone** (or another vector database) to facilitate efficient retrieval for future queries, enabling semantic search and improving the accuracy of subsequent RAG-based responses.

The system also performs a **daily fetch of news data** from external APIs. Newly retrieved articles are preprocessed, filtered, semantically chunked and then **converted into embeddings**, which are stored in **Pinecone** (or another vector database). These embeddings allow the RAG pipeline to efficiently retrieve relevant documents for user queries, ensuring that the AI analysis incorporates the most recent information.

The system further includes a Feedback Management module, where users can evaluate the quality of analysis by rating it, providing comments, and offering more comprehensive feedback. Feedback is validated and cross-checked before being stored in **MongoDB** and is subsequently examined to find trends in user satisfaction. Insights from feedback with a high rating and low rating are used to further enhance the performance of the LLM, improve document ranking, and the relevance and clarity of subsequent analyses. This sets up a feedback loop in the system for continuous learning, where the system learns from user feedback and improves itself.

Lastly, user preferences and watchlists are monitored by the Notification Management module. Matched with users' watchlist holdings, pertinent updates such as news, analysis summaries, and alerts for **stocks** are ascertained. Matched notifications are formatted and sent via email and other user-set options to ensure delivery of relevant and personalized information in a timely manner.

4.2 System Components Specification

4.2.1 API Interface and Request Handling

The User Interface Module serves as the primary interface for interacting with the system. Users can submit queries through a **web-based GUI** or via **REST API endpoints**. The GUI is implemented with HTML5, CSS3, and JavaScript. It exposes

several RESTful API endpoints, such as */status* for system health checks, */history* for retrieving past results, and */analyze* or */webhook* for initiating analysis tasks. The **Flask application** acts as the core web server, handling HTTP requests, routing them to the appropriate endpoints. Upon receiving a request, the **Request Receiver** component parses the JSON payload and validates the input. Specifically, the **Query Validator** checks the query string for acceptable length and structure and applies regular expression-based validation to confirm correct ticker formatting (e.g., stock symbols or Bursa Malaysia codes like 1234.KL).

Requests failing to meet the criteria are rejected with informative error messages, while valid queries proceed to the next phase of execution. This step ensures robustness and protects downstream processes from malformed or ambiguous inputs.

4.2.2 Flask Application Core

The **Flask Application Core** acts as the main web server, managing the entire request-response cycle for the system. Its main tasks involve the handling of HTTP requests, routing, managing sessions, authenticating users, and enforcing security. In the initial application, the security parameters are set such as the session signing secret key, safe cookie parameters, and the CORS policy that grants access to the domain.

The **session management** functionalities of Flask-Login allows user authentication management is coupled with MongoDB for the storage of user credentials and session metadata. Persistent user sessions along with secure login/logout and role based access control are achievable. Cross Site Request Forgery is also used to defend against unauthorized form submission, along with sanitized and validated user input to defend against injection attacks or malicious payloads.

Blueprints in Flask support structural organization for **Route Handling**, allowing the separation of application components such as query processing, feedback processing, notification handling, and API management. Each Blueprint contains its set of routes, HTTP verbs, and malfunction management, providing layer based routing and distributed exception processing. Common HTTP malfunctions such as 400, 404, 500 etc are captured by the failure management layer and structured JSON or HTML malfunctions are returned in response.

4.2.3 Cache Management System

The **Cache Management System** is designed to optimize the performance and responsiveness of the AI-powered analysis platform by minimizing redundant computations and external API calls. It implements a **multi-level caching architecture**, storing frequently accessed data at different granularities, including user queries, financial metrics, and individual stock tickers. Each cache type is configured with specific **time-to-live (TTL) policies**: the query cache retains processed user queries for seven days, the financial cache stores time-sensitive data such as stock prices for one day, and the ticker cache maintains sector profiles or embeddings for up to thirty days. The cache employs a **least recently used (LRU) eviction mechanism** augmented with popularity scoring, allowing frequently accessed or high-demand items to remain in memory even if they are not the most recently used, thereby improving cache hit rates. To support semantic search and approximate query matching, the system performs **cosine similarity checks** on vectorized queries and embeddings, considering a cache hit if the similarity exceeds a threshold of 0.9. By serving cached results for repeated or semantically similar queries, the system significantly reduces external API calls and improves response times, ensuring faster analysis delivery.

4.2.4 Database

The system implements a multi-database architecture to manage structured and persistent data efficiently. **MongoDB (NoSQL)** serves as the primary data store for raw inputs, processed results, and metadata such as risk scores and query similarity scores. Its document-based schema allows flexible and rapid insertion of diverse data structures without the need for schema migration. It maintains collections such as *malaysia_news* for news articles with metadata, *query_cache* for storing analysis results with a **7-day TTL**, users for authentication and preference data, and *user_watchlists* collection for personalized stock monitoring. Connection management is optimized through **connection pooling** with parameters *maxPoolSize=50* and *minPoolSize=10*, ensuring high-performance access even under heavy query loads. TTL indices are used for automatic cleanup of outdated cache and news entries, maintaining database efficiency. The module supports fast retrieval of structured data for query handling, feedback

storage, and notification delivery, forming the backbone of the system's persistent storage layer.

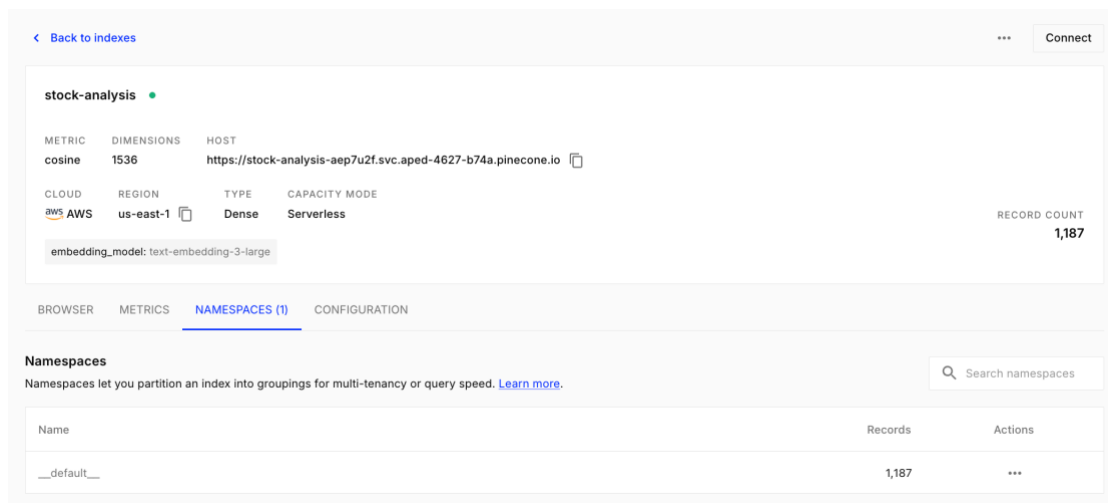


Figure 4.2.1: Pinecone dashboard

On the other hand, **Pinecone** is used for vector similarity search. Each document is encoded into a vector representation and indexed in Pinecone to support fast, approximate nearest-neighbour retrieval. The embedding generator ensures that both documents and queries are transformed into the same vector space using a consistent embedding model.

4.2.5 Document Chunking

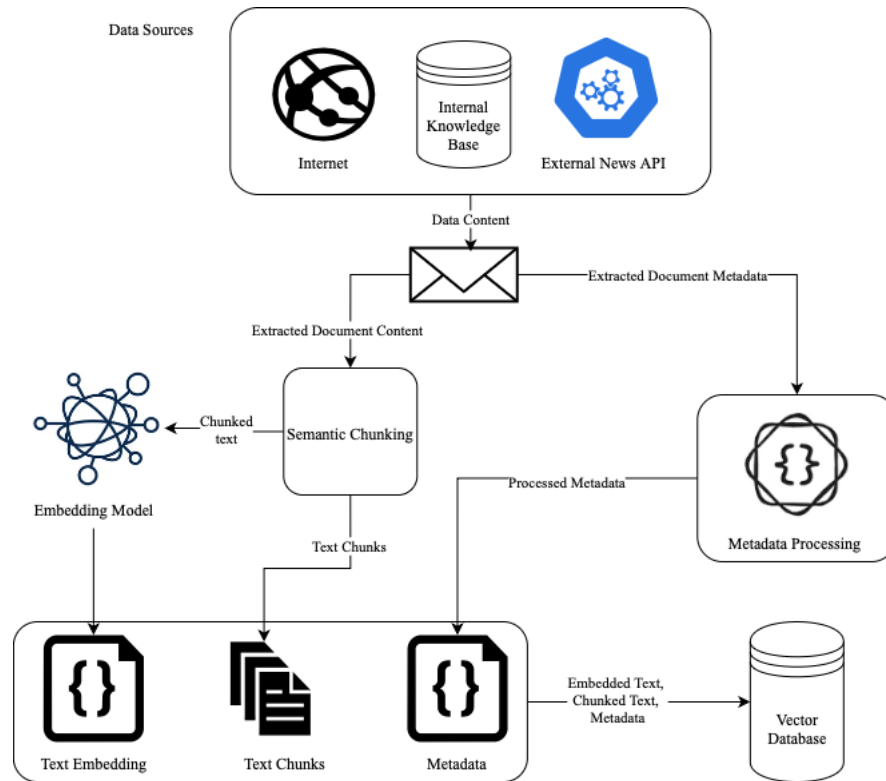


Figure 4.2.2: RAG Data Ingestion Pipeline

The **Chunking Module** is a critical component of the RAG AI Agent Stock Analysis System, responsible for breaking large documents into smaller, semantically meaningful chunks optimized for vector-based retrieval and AI analysis. **Semantic chunking is employed because it preserves the contextual integrity of information**, ensuring that each chunk contains coherent, self-contained units of meaning. This is crucial for AI analysis, as language models perform better when reasoning over contextually complete segments rather than arbitrary text slices that may disrupt sentence flow or logical structure. The module employs a **multi-strategy chunking approach** with intelligent fallback mechanisms. The primary strategy, *SpacyTextSplitter*, leverages spaCy's NLP pipeline (*en_core_web_sm*) for semantic chunking that preserves sentence boundaries and contextual meaning. If semantic splitting fails, the *RecursiveCharacterTextSplitter* performs hierarchical character-based chunking using configurable separators, and the tertiary strategy, *SentenceSplitter* from LlamaIndex, ensures sentence-aware chunking compatible with the RAG pipeline.

The module is driven by a **centralized chunking configuration system**, which defines parameters such as `chunk_size` (512 characters), `chunk_overlap` (50 characters), hierarchical separators (`["\n\n", "\n", ". ", " ", ""]`), and a maximum of 10 chunks per document to prevent excessive fragmentation. Each chunk is enhanced with **metadata** including unique IDs, original document reference, position, chunking method, overlap metrics, and processing timestamps, preserving document lineage and enabling precise retrieval.

The **document processing pipeline** coordinates activities like validation, classification (news, financial report, transcript), chunking, metadata enrichment, and error handling. The robust mechanisms for error handling and fallback ensure processing continuity by logging failures and fallback chunk generation as per need. The module modifies chunk sizes and overlaps, tracks document processing time, memory use, error metrics, chunk quality, and overlaps more, dynamically tailored to document types and content. It also supports batch processing and persistent caching to avoid repeated computation for performance.

The module's final step couples seamlessly with the RAG pipeline and vector database, generating chunk-level embeddings and storing them in Pinecone. Chunks become compatible with query and retrieval engines via the LlamaIndex, which converts them to *Document* objects for semantic searching and AI agent evaluation. This design guarantees that retrieval is contextually relevant, semantically coherent, and chunked to preserve quality for vector-sophisticated retrieval, providing the backbone for reliable, swift, and incisive document analysis.

4.2.6 Document Embedding

The certain crucial aspect of RAG data pipeline is handling the semantic representation of the documents while retaining usefulness of RAG data pipeline. New news articles or documents is split accordingly to balance granularity distortion and fidelity of the vector in chunks of 512 tokens. Each of the chunks is then passed to the OpenAI Embed API (text-embedding-small-3) to create vector embeddings. These embeddings are kept in Pinecone that is a vector database that specializes in high-dimensional semantic search under indexes like 'news-embeddings'. For user prompts the system synthesizes embeddings and locates the semantic cache in MongoDB. If a cache is present, the system fetches the preserved data while a lack of cache prompts the system to perform

a full RAG- based analysis. This module provides the system ability to scan contexts of documents quickly and makes the AI agent to use analysis based on historic and recent data while minimizing the redundant API calls and computations.

4.2.7 Workflow Coordination and Data Fetching

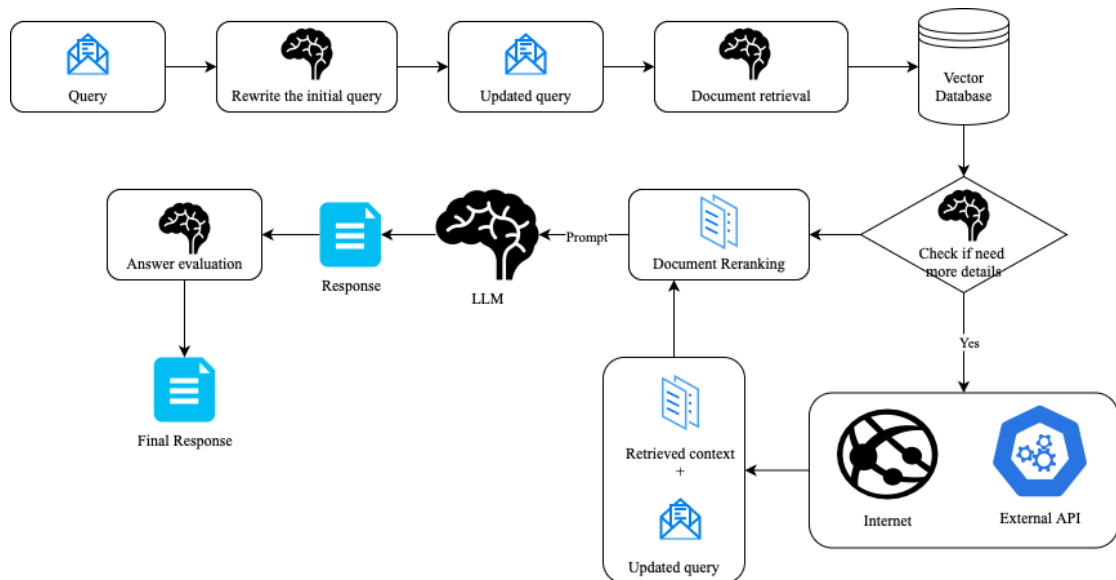


Figure 4.2.3: RAG Query Processing and Augmentation Flow

Once a valid request is received, the **Workflow Coordination and Data Fetching System** orchestrates the end-to-end request processing pipeline, ensuring reliable and efficient data retrieval from multiple heterogeneous sources. When a user submits a query, the system first performs **request validation**, including JSON parsing, input sanitization, follow-up detection, and ticker format verification. Next, a **multi-level cache** is checked, using both enhanced similarity-based embeddings and legacy caching, to quickly return previously processed results and minimize redundant computation. If a cache miss occurs, the system proceeds with **parallel data fetching** from multiple sources, including Pinecone vector database for knowledge base retrieval, MongoDB for structured document data and web search (Google) results. The system incorporates **timeout management** to prevent long-running queries, and **error handling with retry mechanisms** ensures robustness in the face of API failures or malformed data. Retrieved data is aggregated, processed into standardized document objects, and optionally reranked via the LLM's RAG pipeline. **Performance monitoring** tracks CPU, memory, response times, cache hit rates, and error counts to maintain system reliability. Finally, the processed response is returned to the user and

stored in cache for future queries. This architecture ensures that the system is resilient, scalable, and capable of intelligently coordinating multiple data sources while providing high-quality, contextually relevant results.

4.2.8 Document Indexing and Retrieval

The **Document Indexing and Retrieval System** is designed to efficiently store, retrieve, and rank large volumes of textual data for financial analysis. All document embeddings are stored in the Pinecone vector database. The original documents and associated metadata are maintained in MongoDB for structured retrieval. During query processing, relevant chunks are retrieved from Pinecone based on vector similarity and cross-referenced in MongoDB. To improve retrieval accuracy, a multi-factor reranking system is applied, which combines cross-encoder semantic scoring, feedback-based learning, recency boosting, and company/entity relevance, each weighted to optimize relevance.

- **Cross-encoder semantic scoring** evaluates query-document interactions at a deeper contextual level, enabling the retrieval of content that is semantically aligned with the query even if exact keywords are absent.
- **Recency boosting** prioritizes the latest information, which is critical in dynamic financial markets.
- **Company/entity relevance** emphasizes mentions of specific tickers or entities, ensuring investor-focused insights are highlighted.
- **Feedback-based learning** allows the system to adapt based on user interactions and historical relevance performance, further refining ranking over time.

This hybrid approach enables both scalability and contextual richness, as the system can retrieve semantically aligned content even if exact keyword matches are absent. The system produces structured, ranked results ready for RAG processing and downstream analysis. This architecture allows for precise, context-aware document retrieval, enabling the LLM to generate high-quality, actionable insights for investors.

4.2.9 Analysis Module and Insight Generation

The RAG (Retrieval-Augmented Generation) pipeline is designed to enable an AI agent to autonomously retrieve, analyze, and summarize information from multiple knowledge sources while maintaining contextual memory for follow-up interactions. At its core, the system uses a large language model (LLM), specifically GPT-4o-mini, integrated with LlamaIndex to form an AI agent capable of both structured analysis and conversational responses. When a user submits a query, the agent first generates a vector embedding of the query using the text-embedding-3-small model and retrieves the top ten most relevant document chunks from a Pinecone vector database (*similarity_top_k=10*). Each retrieved chunk includes metadata such as chunk ID, article title, source, category, affected sectors, similarity score, and MongoDB ID. The MongoDB database is then queried to fetch the original documents corresponding to these chunks. The retrieved documents are processed and reranked using context-sensitive criteria, including similarity scores, ticker relevance, and recent conversation topics, before being passed to the LLM for analysis.

The AI agent uses two modules. For the first queries, the system carries out structured synthesis checks by merging the retrieved document information, financial data, adaptive prompts, and adaptive prompts and follows a rigid output format: *Risk Score (1–5), Risk Explanation, Summary, and Actionable Insights*, including citations for all information used. In the memory-aware conversational mode, for follow-up queries, the agent uses the system prompt to autobiographically merge the previous discussion topics and discussed tickers with the latest exchanges, steering the LLM to develop non-quantitative and narrative, free-flowing constructions which may bypass the use of formalized text structures, including absence of metrics and formalized text structures. This type of memory is classified as a semantic memory, where the conversation is structured in a context with session-specific information such as queries and responses, sources, risk scores, and timestamps, and the system is organized in a way where insights are stored with confidence and importance parameters.

Prompt Engineering of LLM

1. Role Assignment and Context Setting

The system establishes the LLM's role through crafted system prompts that define the model as "a stock market analyst specializing in analyzing the effects of government policies, macroeconomic factors, market trends on industries." The system also implements dynamic prompt generation through the *generate_adaptive_prompt()* function, which creates contextually appropriate prompts based on:

- **Query Type Detection:** Identifies whether the query is about specific stocks, sectors, or general market analysis
- **Document Context:** Incorporates information from retrieved documents to enhance prompt relevance
- **Financial Data Integration:** Includes real-time financial metrics when available
- **Conversation Context:** Leverages previous conversation history for follow-up queries

2. Prompt Design

In the RAG AI Agent Stock Analysis System, prompts define the LLM as “a stock market analyst analyzing government policies, macroeconomic trends, and market conditions for retail investors.” The LLM is instructed to use available tools to extract risk-related information from chunked documents, evaluate combined impacts, assign a **Risk Gap Score (1–5)** with explanations, and produce summaries with quantitative hooks and actionable investment tips. Prompts enforce strict **source citations** for referenced data while prohibiting citations for the model’s own conclusions. They also account for multiple chunks from the same source and require a **structured output format**—Risk Score, Risk Explanation, Summary, and Actionable Insights—ensuring consistent, traceable, and actionable outputs for downstream processing.

3. Schema Enforcement

To keep outputs structured, the system employs regex-based extraction patterns for Risk Score, Summary, and Actionable Insights. Enforcement of schema preserves uniformity across responses generated by the LLM, assists in the

RAG components' subsequent processing, and guarantees correct alignment of the outputs to the tracking and visualization database. Such a method minimizes uncertainty and allows LLM outputs to be programmatically understood without the need for manual edits.

4. Parameter Tuning

Parameter tuning optimizes LLM performance and reliability. Temperature is set to **0.1** for deterministic, focused responses in main analysis, company detection, complexity assessment, and sector analysis; **0.2** is used for main RAG analysis and sector classification; and **0.3** is applied for news summarization, daily news processing, and historical news processing. The maximum token length is set to *2000* to allow comprehensive analysis, while only the top documents are included in prompts to balance context depth with token efficiency.

5. Cost and Performance Monitoring

The system monitors interaction on a per LLM basis recording prompt, completion and total tokens and estimating model costs, accounts for usage and model pricing per interaction and monitors tokens spent per interaction. Lastly, model pricing and costs are calculated, which allows for monitoring expenses while tracking set context and model performance, for these purposes if context were to be minimized performance drop, costs are controlled especially for financial tasks, which may be harder for larger context tasks, is controlled.

6. Error Handling

Robust error handling is implemented through retry mechanisms that automatically re-attempt LLM requests up to three times with exponential backoff, along with fallback strategies that allow the system to produce basic outputs if enhanced prompts or external data sources fail. These measures ensure resilience and reliability in a production environment where downtime or incomplete responses could negatively impact investor decisions.

7. Token Estimation

Token estimation is integrated to predict the size of prompts and completions, enabling efficient prompt construction within API limits, preventing truncation or incomplete outputs, and supporting cost-aware scaling of document retrieval and prompt assembly. This ensures that the system remains both performant and cost-effective while maintaining output integrity.

By estimating tokens before sending requests, the system balances **context richness with computational efficiency**, ensuring that prompts remain manageable for the LLM while retaining essential information.

Throughout this process, the agent autonomously leverages its tools and knowledge base. The *document_retrieval_tool* encapsulates the Pinecone and MongoDB retrieval, enabling the agent to fetch and rerank relevant policy, market, and news information. The LLM performs the analysis by synthesizing data across multiple chunks, applying the adaptive prompt logic, and integrating real-time financial context when available. The overall system flow consists of embedding the user query, retrieving and processing documents, selecting the appropriate agent mode, generating adaptive prompts, executing LLM analysis, extracting structured insights or producing conversational output, storing results in memory, and updating the conversation context. This architecture allows the AI agent to operate autonomously, continuously integrate new knowledge, and provide both structured and narrative insights with a high degree of contextual awareness.

The system incorporates **robust response extraction** via regex-based parsing, **token usage and cost tracking**, and mechanisms for **follow-up question generation** to maintain conversational relevance. By integrating the RAG framework with structured retrieval, adaptive prompts, and memory-aware conversational handling, the system ensures that investors receive accurate, evidence-backed insights while maintaining a professional, user-friendly interface.

4.2.10 Response Structuring and Delivery

The system's goal throughout the report generating phase is to produce informative and well-structured response. The **Response Formatter Module** ensures that raw LLM outputs are converted into structured, user-friendly, and consistent formats for both

initial and follow-up queries. The module first extracts key analysis components, including risk scores, explanations, summaries, and actionable insights, using regex-based parsing and pattern matching. Responses are then processed into **markdown and HTML formats**, correcting formatting issues and enabling rich frontend rendering. A dedicated **citation processing system** generates a source list, tracks citation usage, and updates citation numbers dynamically to maintain accuracy. Standardized error handling captures exceptions, logs severity, and delivers structured error responses for frontend processing. The module integrates tightly with the Flask backend, producing structured JSON responses that include timing metrics, conversation context, and source references. On the frontend, JavaScript routines further process first-message and follow-up responses, combine raw and formatted content, and ensure correct display of citations and related information. The response is then returned to the user by the **Response Sender** with an appropriate HTTP status code and results in JSON format.

Concurrently, the results are archived into MongoDB (*risk_scores* collection), while the input query and its corresponding embeddings are stored for future cache matching. This dual storage ensures data traceability and operational efficiency. Overall, the Response Formatter Module guarantees **clarity, reliability, and consistency** in all AI-generated outputs, enhancing interpretability and user experience while maintaining robust error handling and citation integrity.

4.2.11 Response Processing Module

Once the backend returns a result, the **Response Processing Module** evaluates the quality of the response using heuristics, such as text length and an LLM-generated relevance score from 1 to 10. Responses that do not meet quality thresholds are regenerated. The module then formats the output, adding appropriate markup for readability, and generates the **final response**, delivered either as JSON for API consumers or rendered HTML for GUI users.

4.2.12 Feedback Loop Mechanism

The component known as the Feedback Loop is essential to maintaining the system's relevance and performance over time. Users' opinions on the reports' correctness and

usefulness are gathered via the **Feedback Manager**. After then, the feedback is examined to identify potential areas for system improvement. These insights are used to iteratively improve the system, enhancing the quality of subsequent studies and better meeting user needs through process refinement. The module operates through a structured workflow: user interactions are captured and stored in **episodic, semantic, and conversation memories**, each tagged, timestamped, and embedded for retrieval. Relevant context is dynamically retrieved based on similarity and importance scores to guide context-aware response generation, ensuring continuity and relevance in conversations. The system collects implicit and session-based feedback, tracking user engagement, response quality, and interaction patterns, which are then used to adapt memory importance, clean outdated data, and refine system parameters. Semantic knowledge extraction and pattern recognition enable the system to generalize insights across multiple interactions, forming a growing knowledge base. A continuous learning pipeline processes interaction batches to identify recurring themes, update semantic memories, and adjust AI behavior, while performance metrics monitor response quality, memory utilization, and learning effectiveness.

Additionally, it logs any errors or issues encountered during the data transfer, facilitating troubleshooting and maintaining the integrity of the data exchange process. By using an iterative process, the system is guaranteed to continue providing accurate and insightful insights while also being receptive to user comments.

4.2.13 Watchlist Module

The Watchlist Module serves as a comprehensive stock tracking and notification system within the broader AI-driven stock analysis platform, seamlessly integrated with the RAG framework for autonomous insights. Users can create personalized watchlists of Bursa Malaysia stocks, with each stock containing detailed metadata including symbol, ticker, company name, sector, and industry. The module interacts with real-time financial sources via yFinance, caching sector information and stock prices in MongoDB to optimize performance. Users can search for stocks through enhanced endpoints that combine local search with fallback to yFinance, add or remove stocks, and receive email notifications. The system performs sector-based analysis of user portfolios and mapping stocks to sectors and industries for targeted news matching.

This enables the AI agent to identify relevant users for sector-specific news notifications and alerts, integrating with the daily news system to deliver personalized insights.

4.2.14 Automated News Processing System

The automated news processing system implemented is designed as a modular, event-driven pipeline that integrates LLM-powered summarization, impact classification, and vector-based retrieval on LLM-driven stock analysis and policy impact. The pipeline applies for both historical data and fresh news data. As soon as a news article is ingested, the system immediately normalizes the content, validates it, and checks for duplicates to ensure clean data. Each article then undergoes LLM-based summarization using GPT-3.5-turbo, producing concise two-sentence summaries that highlight financial implications, affected sectors, and companies. If the LLM fails or confidence is below a threshold (0.7), a **keyword-based fallback** analyses the article with parameters such as a sector-to-keyword mapping dictionary, `relevance_score` increment per keyword match, and thresholds for classification: *HIGH* ≥ 8 , *MEDIUM* ≥ 4 , *LOW* < 4 . This ensures that even under high throughput or partial AI failure, the system provides an immediate preliminary assessment.

Articles are then split into semantic chunks and embedded using OpenAI embeddings, with the vectors stored in Pinecone for high-dimensional similarity searches, enabling fast and precise semantic retrieval. MongoDB serves as the persistent document store, with optimized indexes on article ID, publication timestamp, impact classification, and sectors to ensure efficient query performance, while Pinecone vectors are linked to MongoDB IDs for hybrid metadata and semantic searches. The system incorporates a circuit breaker pattern for resilience against API failures. Finally, the enriched news articles feed directly into the stock analysis chatbot, allowing the LLM agent to correlate news sentiment, policy changes, and macroeconomic trends with industry-specific financial risks.

4.2.15 Stock Notification Module

The **Stock Notification Module** is an automated pipeline designed to fetch Malaysia-specific news, perform AI-driven sector analysis, match user preferences, and deliver

personalized email notifications daily at 8:00 AM Malaysia time (UTC+8). The scheduling engine utilizes Python's schedule library with persistent threading, time zone handling, and real-time job status monitoring to trigger daily execution. The news acquisition system integrates with the **NewsData.io API**, performing content validation, category filtering, geographic targeting, and API rate limiting to ensure high-quality news data.

The **LLM-powered sector analysis engine** uses OpenAI's GPT-3.5-turbo to classify articles into affected stock market sectors and industries, determine impact levels (high/medium/low), and provide reasoning with confidence scoring. A keyword-based fallback mechanism ensures analysis continuity, cross-referencing LLM outputs with predefined sector-industry mappings. The **user preference matching system** interfaces with MongoDB collections to retrieve users' watchlists and notification settings, mapping affected sectors to user-owned stocks and identifying recipients for personalized alerts. If no interested users are found, the workflow ends with no emails sent. Next, articles are **matched to users' watchlists**, and users without relevant articles receive no notifications. Only when all preceding conditions are satisfied does the system proceed to **generate and send personalized email notifications**, ensuring that only pertinent, contextually validated content reaches end users.

The **email generation and delivery system** employs SMTP configuration with both HTML and plain-text email support, leveraging responsive templates, color-coded sector visualizations, dynamic greetings, news content, impact indicators, stock relevance, and actionable links to dashboards and watchlists. The **news HTML update system** maintains a local HTML dashboard, automatically injecting update timestamps, article counts, and refreshed metadata. Integration with the **unified news processor** orchestrates batch processing, error handling, fallback support, and ensures compatibility with existing systems while validating processed article completeness.

The entire system is under surveillance by a module advanced enough to log processing time and articles requested, processed, and notifications generated, and system interregnum monitoring on database connectivities and health check configurations. Dependencies are schedule, newsdataapi, openai, pymongo, and python-dotenv and API keys and credentials environment variables for the module are set. The system is

set for automatic start with constant background processing and the ability to invoke test runs for the complete system to ensure immediate end-to-end functionality. This configuration offers dependable news collection and AI processing, tailored notifications with intelligent NLP processing, daily fetching with systematic error resolution, outstanding daily performance visibility, tailored notifications with resolution intelligent NLP processing, daily fetching for unintended interruption, and robust performance logs on daily error cache processing. All automatic and handcrafted daily news are routed to the users.

CHAPTER 5: SYSTEM IMPLEMENTATION

5.1 Hardware Setup

The hardware involved in this project is a computer, which plays a critical role in supporting local development. This machine is used to host the Flask-based application, manage API connections (e.g., to OpenAI, Pinecone, and MongoDB), and perform data processing pipelines. The detailed specifications of the computer are presented in Table 5.1.1.

Table 5.1.1 Specifications of Computer

Description	Specifications
Model	MacBook Air (M1, 2020)
Processor	Apple M1 chip
Operating System	macOS Sequoia Version 15.1
Graphic	-
Memory	8GB unified memory
Storage	256GB SSD

5.2 Software Setup

To ensure the successful development and deployment of the system, a combination of software tools, programming libraries, and APIs were utilized. These technologies were selected based on their compatibility with the project objectives, efficiency, scalability, and ease of integration with LLMs and RAG workflows. Table 5.2.2 summarizes the key software and development tools used throughout the development lifecycle.

Table 5.2.2 Overview of the Software and Development Tools Used

Category	Component	Tools and Technologies	Purpose
Programming Languages and Frameworks	Core Programming Language	Python	Main language for prototype development
	Backend Framework	Flask	Core web framework for REST API endpoints and routing

	Front-End Framework	React	Interactive user interface for interacting with backend
	LLM Application framework	Langchain	Text splitting for chunking
Core Technologies and Libraries	LLM Responses	OpenAI	Language model access
	Text embeddings	OpenAI text-embedding-3-small model	Generate embeddings for documents
	Data Indexing and Retrieval	LlamaIndex	RAG implementation for document indexing and retrieval
	Natural Language Processing	spaCy	Advanced NLP for text processing and semantic chunking
Data Management	Structured Storage	MongoDB	Storing processed and raw data
	Vector database	Pinecone	Storing and retrieving semantic vectors
	Data Transfer	JSON	Lightweight format for API communication and storage
Communication	Protocol	HTTP/HTTPS, RESTful APIs	API communication and integration
	Email Sending	Smtplib	SMTP email sending
Version Control	Platform	GitHub	Source code management
Scheduling and Background Tasks	Cron job scheduler	Schedule	Job scheduling for daily news fetching
	Multi-threading support	Threading	Multi-threading support for concurrent operations
Logging	Logging Structurer	Structlog	Structured logging for better analysis
API and External Services	Financial Data	yFinance	Retrieving stock and market data
	LLM Access	OpenAI API	Accessing GPT-4o mini model for insight generation and embedding generation

	News Data	NewsDataApiClient	News data provider for Malaysia-specific news
	Vector Management	Pinecone API	Managing vector indices and embeddings
	Web Search	Google Custom Search Engine API	Search for additional information to supplement analysis with web data
User Authentication and Management	User Session Management	Flask-Login	User session management and authentication state
	Password Hashing	Bcrypt	Password hashing for secure storage
	Data Verification	itsdangerous	Generate token for email verification links
	Email validation	Email-validator	Email address validation
Development Tools	IDE	Microsoft Visual Studio Code	Code writing and debugging
	API Testing	Postman	Testing RESTful API endpoints
	Version Control	Git + GitHub	Source code management

The choice of **Python** as the core programming language reflects its versatility and strong support for AI, NLP, and web development. Its mature libraries, such as NumPy, Pandas, PyTorch, and spaCy, accelerate prototype development while enabling seamless integration with LLM APIs. **Flask** was selected as the backend framework due to its lightweight, modular design, allowing precise control over REST API endpoints, routing, and middleware, making it ideal for a medium-scale AI system. For the frontend, **React** provides an interactive, component-based interface that supports real-time data visualization and smooth integration with backend APIs. **LangChain** is employed to structure the text processing pipeline, enabling semantic chunking, document retrieval, and RAG operations, which are central to the system's functionality.

Core AI and NLP components were chosen for their technical suitability. **OpenAI GPT-4o mini** and the **text-embedding-3-small model** provide high-quality

embeddings and reasoning capabilities with manageable computational cost, essential for semantic retrieval and qualitative analysis. **LlamaIndex** is used to efficiently index documents and implement retrieval-augmented generation, connecting multiple heterogeneous data sources such as policy documents, financial data, and news articles. **spaCy** enables advanced NLP preprocessing, including tokenization, named entity recognition, and semantic chunking, ensuring accurate extraction of meaningful information from large text datasets.

For data management, **MongoDB** offers flexible, schema-less storage suitable for semi-structured documents, processed LLM outputs, and metadata. Combined with **Pinecone** as a vector database, the system achieves fast and scalable semantic search for embeddings, supporting accurate top-k retrieval and reranking operations. **JSON** is employed as the standard data transfer format, allowing lightweight and interoperable communication across system components.

The system's architecture also integrates essential development, scheduling, and logging tools. **Git and GitHub** facilitate version control and collaboration, while **Visual Studio Code** supports efficient coding and debugging. **Postman** ensures systematic testing of REST APIs. **Schedule** and **Threading** enable concurrent processing and background task execution, crucial for daily news ingestion and large-scale retrieval operations. **Structlog** provides structured logging for easier debugging, monitoring, and analysis of complex AI workflows.

External APIs are selected for their technical fit and data reliability. **yFinance** delivers comprehensive stock and market data, while **NewsDataApiClient** and **Google Custom Search Engine API** provide domain-specific news and supplementary web data. **Pinecone API** supports efficient vector management for embeddings, ensuring scalable retrieval-augmented generation workflows. For security and authentication, **Flask-Login**, **Bcrypt**, **itsdangerous**, and **Email-validator** enforce secure session management, hashed password storage, token-based verification, and validated user input, adhering to standard security practices in web applications.

Overall, these tools and technologies were chosen not merely for their stated purposes but because they **technically suit the system's requirements**: enabling high-performance LLM-based analysis, scalable semantic retrieval, real-time processing,

structured output generation, and secure web service deployment. This careful selection ensures the system is **robust, maintainable, and capable of producing reliable insights** into the impact of government policies and economic trends on industries.

MMLU vs. Price, Smaller models

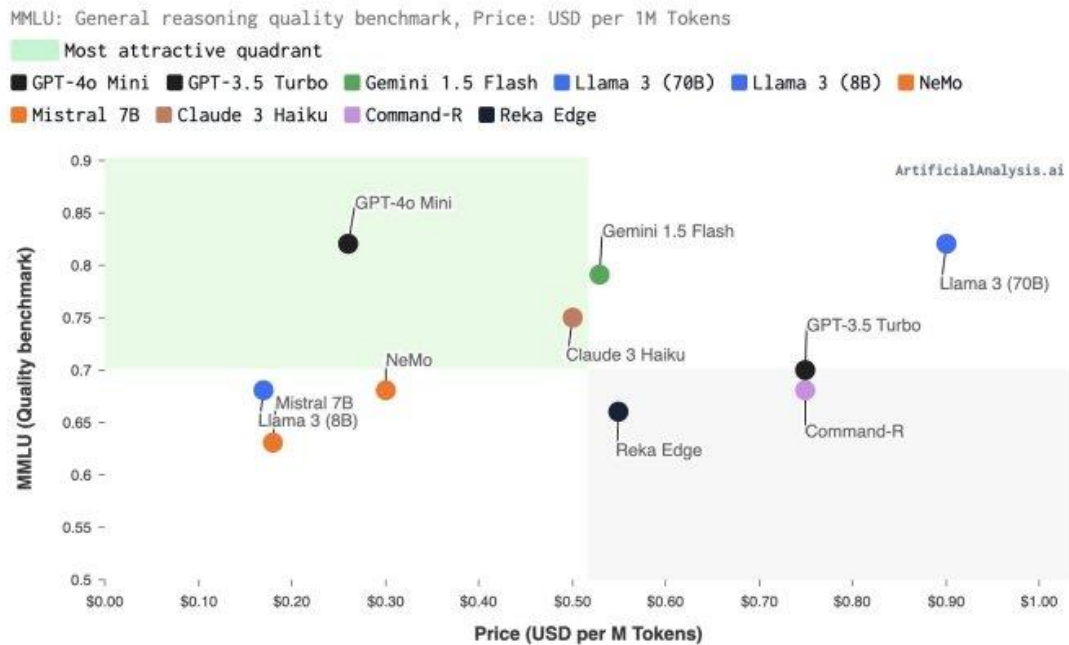


Figure 5.2.1: Scatter plot comparing MMLU benchmark scores (general reasoning quality) and pricing (USD per 1M tokens) for smaller AI models. Source: [59]

GPT-4o mini was chosen for its highly cost-effective and high performance, thus making it suitable for an application aimed at retail investors. With reference to Figure 5.2.1 [59], with a total cost of just \$0.25 per million input and output tokens, it is approximately 4.7 times cheaper than GPT-3.5 Turbo (\$0.75). This affordability ensures that the system remains scalable and sustainable, especially for solutions intended to be widely accessible. Despite its lower cost, GPT-4o mini achieves an MMLU score of 82%, which is significantly higher than GPT-3.5 Turbo's 70%. This demonstrates its strong reasoning capabilities and suitability for interpreting complex qualitative data such as policy documents and economic reports, which are central to effective stock analysis.

5.3 Setting and Configuration

5.3.1 Local Development Setup

```
(base) tangjiahui@Tangs-MacBook ProjectI_Prototype % export FLASK_APP=analysis.py
(base) tangjiahui@Tangs-MacBook ProjectI_Prototype % flask run --host=0.0.0.0 --port=5000

✓ MongoDB connection established for user authentication
* Serving Flask app 'analysis.py'
* Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.27:5000
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug:127.0.0.1 - - [19/Sep/2025 07:16:45] "GET / HTTP/1.1" 302 -
INFO:werkzeug:127.0.0.1 - - [19/Sep/2025 07:16:45] "GET /auth/login?next=/ HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [19/Sep/2025 07:16:46] "GET /favicon.ico HTTP/1.1" 204 -
INFO:werkzeug:127.0.0.1 - - [19/Sep/2025 07:23:47] "GET / HTTP/1.1" 302 -
INFO:werkzeug:127.0.0.1 - - [19/Sep/2025 07:23:47] "GET /auth/login?next=/ HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [19/Sep/2025 07:23:48] "GET /favicon.ico HTTP/1.1" 204 -
```

Figure 5.3.1 Terminal Output of Flask Server Initialization

The system was primarily developed and tested in a local environment as shown in the figure above (Figure 5.3.1). The steps were as follows:

1. Under the project directory, set the flask application entry point:
2. Run the Flask server with public host binding.
3. MongoDB was connected successfully for authentication and data storage.
4. The system became accessible locally via:
 - `http://127.0.0.1:5000`
 - `http://192.168.0.27:5000`

This setup allowed local testing and debugging of all system features, including authentication, query validation, data retrieval, and analysis generation.

5.3.2 Online Hosting

```
(base) tangjiahui@Tangs-MacBook ~ % cloudflared tunnel --url http://localhost:5000

2025-09-18T23:16:17Z INF Thank you for trying Cloudflare Tunnel. Doing so, without a Cloudflare account, is a quick way to experiment and try it out. However, be aware that these account-less Tunnels have no uptime guarantee, are subject to the Cloudflare Online Services Terms of Use (https://www.cloudflare.com/website-terms/), and Cloudflare reserves the right to investigate your use of Tunnels for violations of such terms. If you intend to use Tunnels in production you should use a pre-created named tunnel by following: https://developers.cloudflare.com/cloudflare-one/connections/connect-apps
2025-09-18T23:16:17Z INF Requesting new quick Tunnel on trycloudflare.com...
2025-09-18T23:16:22Z INF Your quick Tunnel has been created! Visit it at (it may take some time to be reachable): https://except-makes-accompanying-scenic.trycloudflare.com
2025-09-18T23:16:22Z INF Cannot determine default configuration path. No file [config.yml config.yaml] in [~/cloudflared ~/.cloudflared-warp ~/.cloudflared-warp /etc/cloudflared /usr/local/etc/cloudflared]
2025-09-18T23:16:22Z INF Version 2025.8.0 (Checksum 387e5cb88e0f2db5f30bfd7639206bad63a49ba530661f3b3e98c60569938ce3)
2025-09-18T23:16:22Z INF GOOS: darwin, GOVersion: go1.24.6, GoArch: arm64
2025-09-18T23:16:22Z INF Settings: map[ha-connections:1 protocol:quic url:http://localhost:5000]
2025-09-18T23:16:22Z INF cloudflared will not automatically update if installed by a package manager.
2025-09-18T23:16:22Z INF Generated Connector ID: 8e51afdb-7694-4d07-86fd-1b06a998a692
2025-09-18T23:16:22Z INF Initial protocol quic
2025-09-18T23:16:22Z INF ICMP proxy will use 192.168.0.27 as source for IPv4
2025-09-18T23:16:22Z INF ICMP proxy will use fe80::1830:3d50:7021:8648 in zone en0 as source for IPv6
2025-09-18T23:16:22Z INF Created ICMP proxy listening on 192.168.0.27:0
2025-09-18T23:16:22Z ERR Cannot determine default origin certificate path. No file cert.pem in [~/cloudflared ~/.cloudflared-warp ~/.cloudflared-warp /etc/cloudflared /usr/local/etc/cloudflared]. You need to specify the origin certificate path by specifying the origincert option in the configuration file, or set TUNNEL_ORIGIN_CERT environment variable originCertPath=
2025-09-18T23:16:22Z INF ICMP proxy will use 192.168.0.27 as source for IPv4
2025-09-18T23:16:22Z INF ICMP proxy will use fe80::1830:3d50:7021:8648 in zone en0 as source for IPv6
2025-09-18T23:16:22Z INF Starting metrics server on 127.0.0.1:20241/metrics
2025-09-18T23:16:22Z INF Tunnel connection curve preferences: [X25519MLKEM768 CurveP256] connIndex=0 event=0 ip=198.41.192.27
2025-09-18T23:16:23Z INF Registered tunnel connection connIndex=0 connection=88ce6a3f-22a9-421f-8d7c-a1bb19546d5a event=0 ip=198.41.192.27 location=kul01 pr
otocol=quic
```

Figure 5.3.2 Cloudflare Tunnel for Online Hosting

To allow remote access and online testing, **Cloudflare Tunnel** was used. This provided a secure HTTPS link that exposed the local Flask server to the internet as shown in Figure 5.3.2.

- Cloudflare automatically generated a public URL (e.g., <https://except-makes-accompanying-scenic.trycloudflare.com>).
- This allowed external users, testers, or supervisors to access the system without requiring a static IP or port forwarding.

5.4 System Operation

5.4.1 Register Page

Figure 5.4.1 Registration page with password validation displaying red warning message for incorrect format

The figure above (Figure 5.4.1) illustrates the registration page. When a new user attempts to register, the registration page includes built-in password validation to enforce secure account creation. If the entered password does not meet the required format, the system immediately provides feedback by displaying a red warning message beneath the input field, guiding the user to correct the error before proceeding.

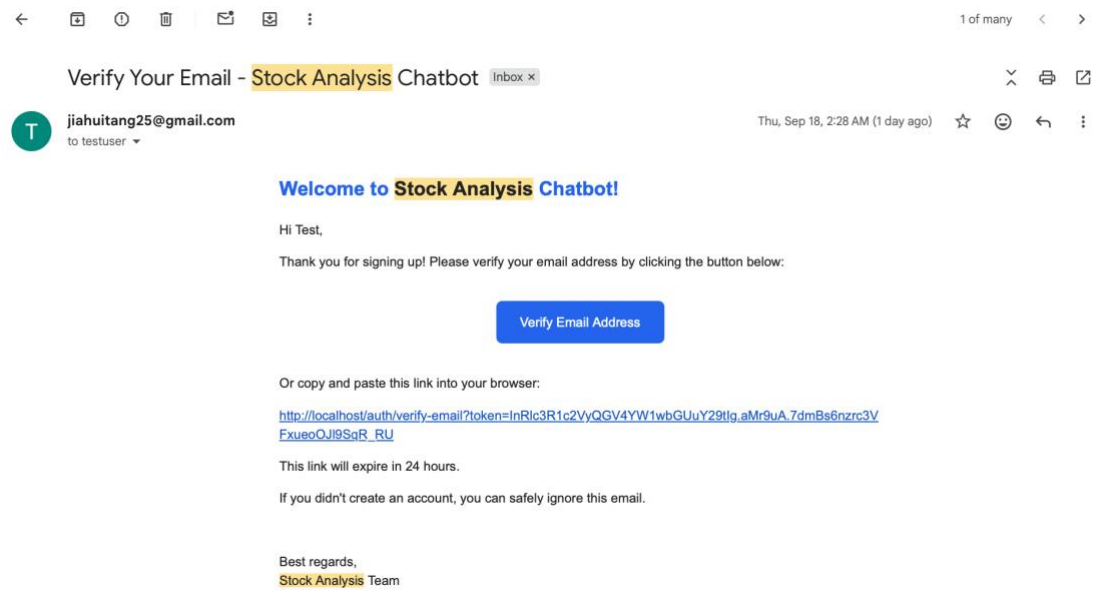


Figure 5.4.2 Verification email sent to user

Following account registration, the system enforces an additional layer of security by requiring email verification. Once the registration form is submitted, the system automatically generates a verification token and sends an email to the provided address (see Figure 5.4.2). The email contains a secure verification link which must be clicked to activate the account. This step ensures the authenticity of user accounts and prevents misuse through fake or invalid registrations.

5.4.2 Login Page

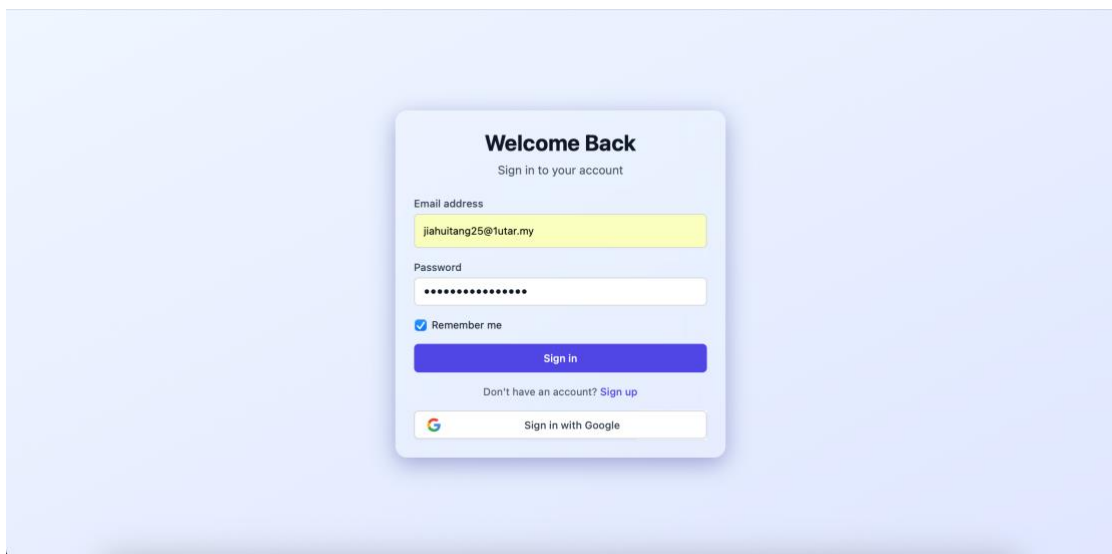


Figure 5.4.3 Login page featuring the “Remember Me” option for persistent sessions

As shown in Figure 5.4.3, when user access the system, they will be directed to the login page by default. This interface includes a standard login form with an additional **“Remember Me”** option, allowing users to maintain their session across visits without repeated logins, thus improving convenience and usability.

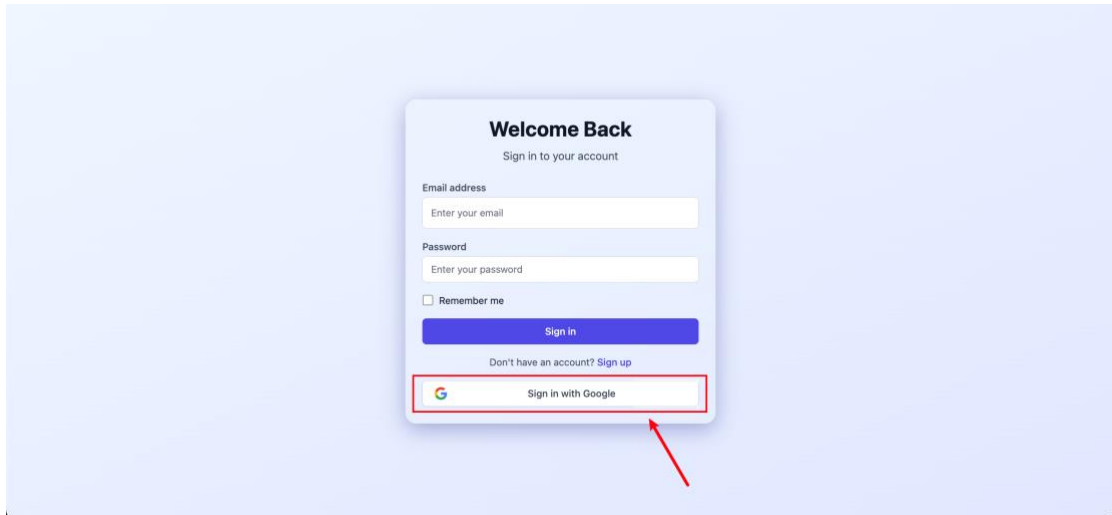


Figure 5.4.4: Sign in with Google option in the Login Page

The stock analysis chatbot system implements **Google OAuth 2.0 authentication** to securely manage user login and identity verification. When a user clicks the **“Sign in with Google”** button on the frontend, the system generates an **OAuth authorization request** directed to Google’s authentication servers. This request includes essential parameters such as the *client_id* (corresponding to the chatbot application), the *redirect_uri* (endpoint in the backend to handle the authentication response), the requested scope (profile and email), and the *response_type* set to code.

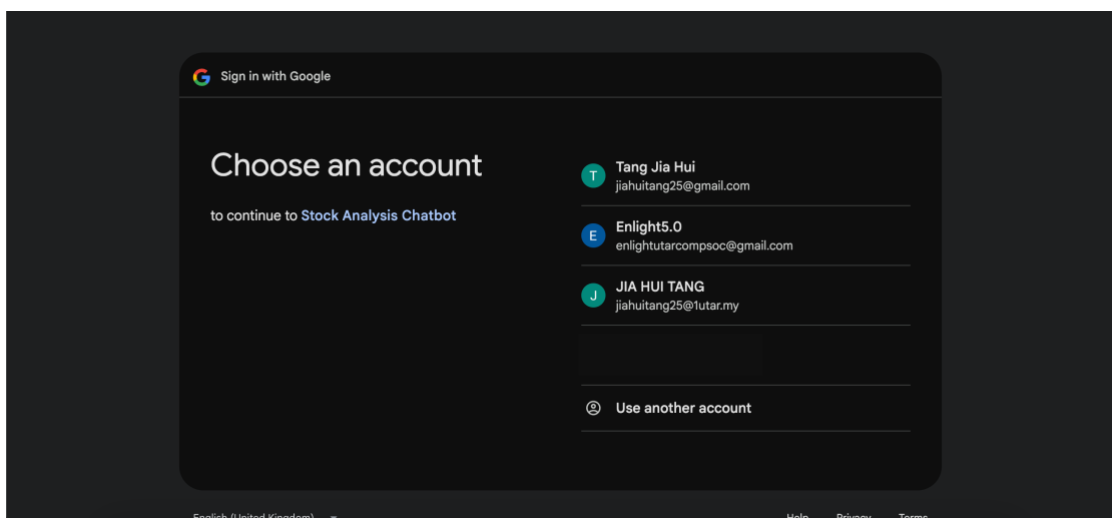
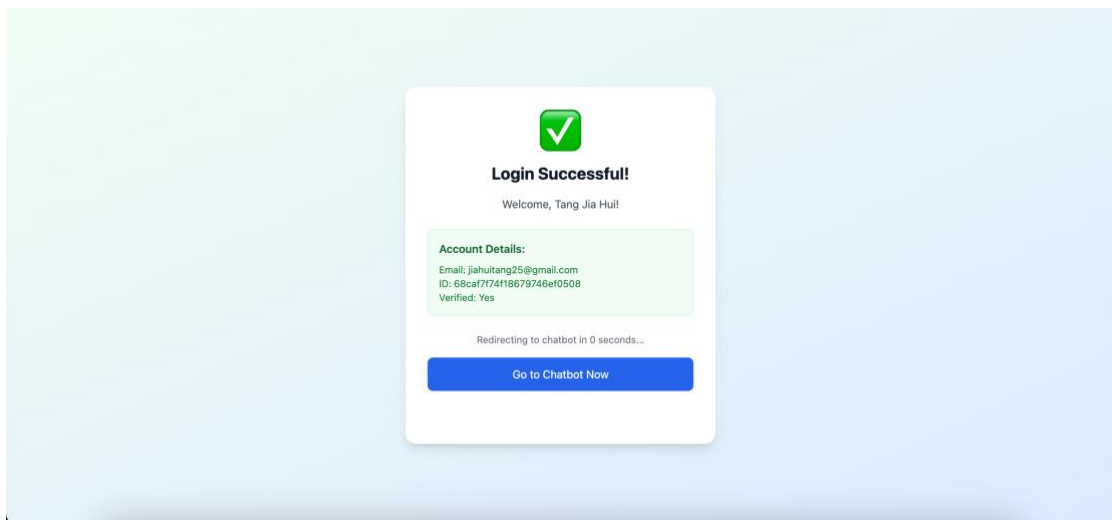


Figure 5.4.5: Google Account Selection Interface

Upon receiving the request, Google presents the user with an **account selection interface**, listing all available Gmail accounts associated with the device or session. The user selects the account to be used for authentication, and Google handles the verification of credentials, including password validation and multi-factor authentication if enabled. This ensures that sensitive user credentials are processed exclusively by Google's secure infrastructure, adhering to OAuth security best practices.

*Figure 5.4.6: Login Successful Message*

The “Login Successful” page is served to the frontend after the access token is validated. The page contains the additional feature of a short countdown timer (4 seconds) which aids the user in visually acknowledging the successful authentication before the page redirects to the chatbot interface. This page transition makes the process smoother and ensures unnecessary changes in context.

The user is then redirected to the stock analysis chatbot interface where the backend provides a session to the user tied to the authenticated identity. This enables context aware functionality and permits the user to access unused financial analysis to enable and retrieve prior conversation history all from a customized watchlist. The design framework in which the flow is built from shows a clear separation of concerns: the frontend is responsible for user interaction, authentication is done by Google servers,

and the backend is responsible for validating the tokens and managing the session which guarantees ease of access and security.

5.4.3 Chat Interface

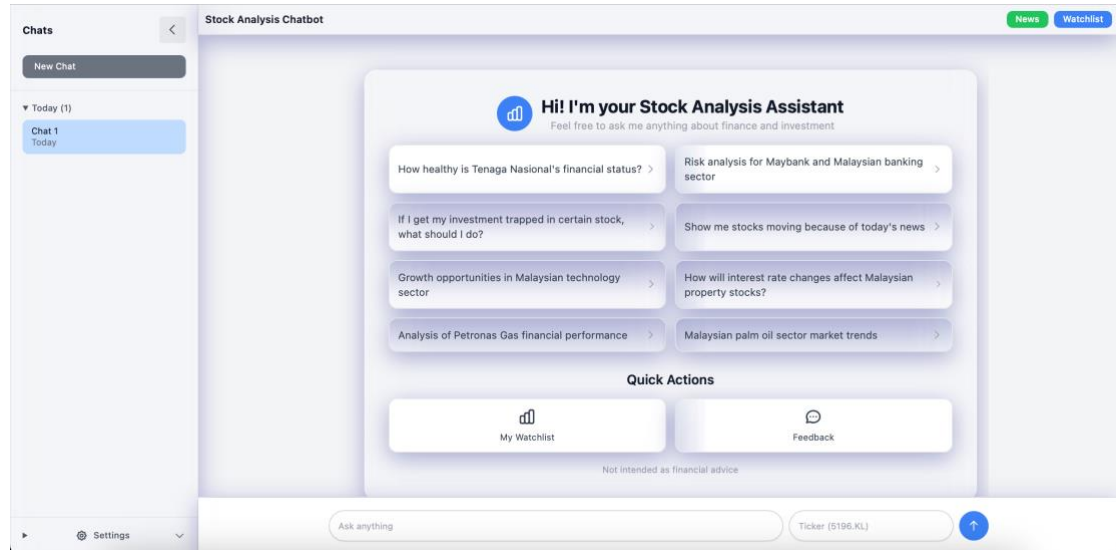


Figure 5.4.7 Empty Chat Interface

Once authenticated (see Figure 5.4.7), the user is presented with the system's main chat interface. At this stage, the chat window is initially empty but populated with a set of suggested queries. These suggestions serve as entry points to help users quickly explore the system's analytical capabilities without needing to formulate their own queries from scratch.

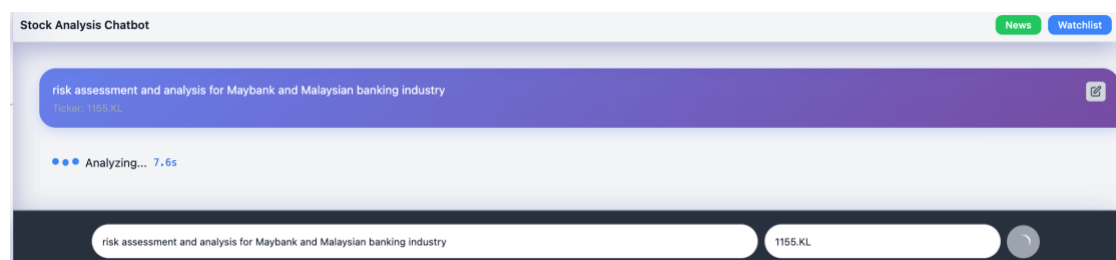


Figure 5.4.8 Ticker Auto-filling

In the chatbot, if any user query containing a company name, the system implements **automatic company detection and financial data enrichment** by first passing user queries through an LLM-based company detection function (*detect_company_with_llm*) that identifies Malaysian company names, including abbreviations and common variations. Detected names are then resolved to their respective **Bursa stock tickers**

via a **multi-tiered fuzzy matching pipeline**, which prioritizes abbreviation mapping, similarity scoring, and partial matching to ensure high confidence in name-to-ticker resolution. Once tickers are mapped, the system retrieves **real-time financial metrics**—such as current price, 52-week high/low, beta, market capitalization, and sector/industry information—via the yFinance API, supplemented by caching mechanisms for performance optimization.

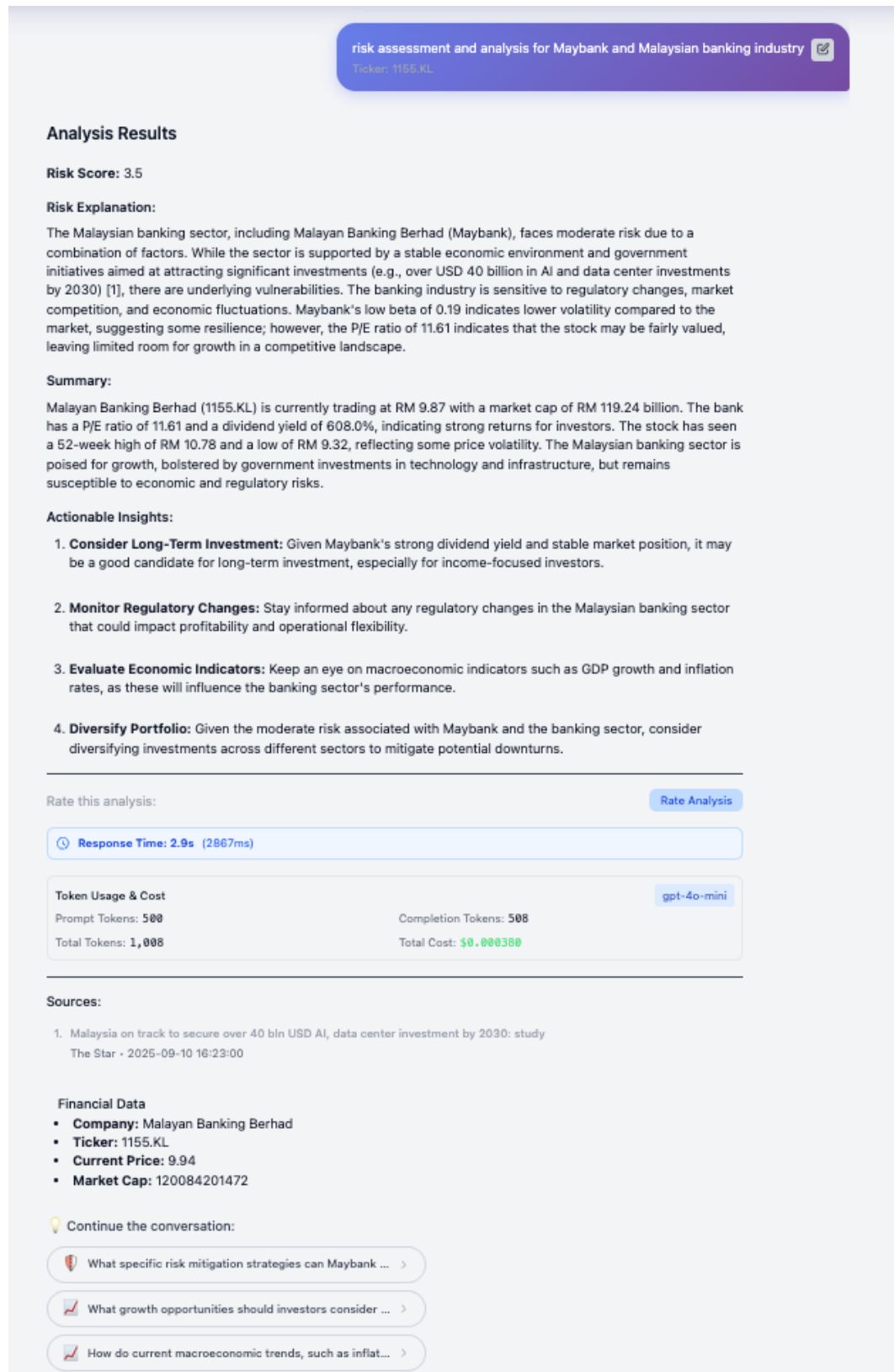


Figure 5.4.9 Analysis result responding to query

Users can input queries into the chat interface to request stock or policy analysis as illustrated in Figure 5.4.9. When a query is submitted, the system executes its Retrieval-

Augmented Generation (RAG) pipeline to produce a structured analysis report. The output includes several key components: a **risk score** rated on a numerical scale, a **risk explanation** detailing the underlying factors, a **summary** that condenses market trends, and **actionable insights** providing practical recommendations for decision-making.

In addition, the interface transparently displays **token usage and estimated cost** associated with the analysis, giving users awareness of resource consumption. The response also lists **sources and citations**, ensuring traceability and reliability of the information. Finally, the system suggests **follow-up questions**, enabling users to naturally continue their exploration and refine their analysis through interactive dialogue.

How do current macroeconomic trends, such as inflation and GDP growth, impact the overall stability and profitability of the Malaysian banking industry, including Maybank?

Index - 1155.KL

Response Follow-up Response

Impact of Macroeconomic Trends on the Malaysian Banking Industry and Maybank (1155.KL)

The stability and profitability of the Malaysian banking industry, particularly for Maybank Berhad (Maybank), are significantly influenced by macroeconomic trends such as inflation and GDP growth. This analysis will explore these factors in detail, incorporating current financial data and relevant market conditions.

Current Macroeconomic Landscape

Inflation Trends:

- Malaysia has been experiencing fluctuating inflation rates, which can impact consumer spending and borrowing. As of the latest reports, inflation remains a concern globally, affecting purchasing power and potentially leading to tighter monetary policies.
- Bank Negara Malaysia has maintained the interest rate at 3.00% to support economic recovery while managing inflationary pressures [1]. This stable interest rate environment is crucial for banks as it influences lending rates and profitability.

GDP Growth:

- The Malaysian economy is projected to recover gradually, supported by government initiatives aimed at economic revitalization. The recent unveiling of a comprehensive economic recovery plan by the Malaysian government is expected to bolster GDP growth [2].
- A growing GDP typically correlates with increased lending activity, as businesses and consumers are more likely to borrow for expansion and consumption, respectively. This is beneficial for banks like Maybank, which rely on loan growth for revenue.

Financial Performance of Maybank (1155.KL)

Current Price: RM 9.87
Market Cap: RM 119.24B
P/E Ratio: 11.61
Dividend Yield: 6.08%
Beta: 0.19

Maybank's financial metrics indicate a relatively stable investment profile. The P/E ratio of 11.61 suggests that the stock may be undervalued compared to its peers, potentially offering a buying opportunity for investors. The high dividend yield of 6.08% is particularly attractive, reflecting the bank's commitment to returning value to shareholders, which is crucial in a low-interest-rate environment.

Opportunities and Risks

Opportunities:

- **Loan Growth:** With GDP growth projected to improve, Maybank stands to benefit from increased loan demand across various sectors, including consumer, SME, and corporate lending.
- **Digital Transformation:** Maybank has been investing in digital banking solutions, which can enhance customer experience and operational efficiency, potentially leading to higher profitability.

Risks:

- **Credit Risk:** An economic downturn or prolonged inflation could lead to higher default rates on loans, impacting the bank's asset quality.
- **Regulatory Changes:** Changes in monetary policy or banking regulations could affect profitability margins, particularly if interest rates rise unexpectedly.

Conclusion and Actionable Insights

In summary, the Malaysian banking industry, and Maybank in particular, are navigating a complex macroeconomic environment characterized by stable interest rates and a recovering GDP. While there are significant opportunities for growth, particularly in lending and digital transformation, investors must remain vigilant regarding potential risks, including credit risk and regulatory changes.

Next Steps for Investors:*

- Monitor Economic Indicators:** Keep an eye on inflation rates and GDP growth forecasts, as these will directly impact Maybank's performance.
- Evaluate Loan Growth Trends:** Assess the bank's quarterly reports for insights into loan growth and asset quality.
- Consider Valuation Metrics:** Given the current P/E ratio, evaluate whether Maybank presents a compelling investment opportunity compared to its peers.

By staying informed and analyzing these macroeconomic trends, investors can make more informed decisions regarding their investments in Maybank and the broader Malaysian banking sector.

Was this response helpful? 👍 👎

⌚ **Response Time:** 27.5s (27544ms)

Sources:

1. Bank Negara Malaysia Keeps Interest Rates Unchanged at 3.00%
The Edge Malaysia
2. Malaysian Government Unveils Comprehensive Economic Recovery Plan
Malaysian Economic Review

💡 Continue the conversation:

🔍 How do you see the recent interest rate changes affecting...

🏆 Considering the competitive landscape, how might M...

📋 What strategic initiatives should Maybank prioritize t...

Figure 5.4.10 System response generated after the user clicks a suggested follow-up question

When the user clicks on one of the suggested follow-up questions, the system automatically reprocesses the query without requiring manual text input as shown in Figure 5.4.10. This interaction streamlines the analysis process, guiding users toward deeper exploration of market risks, policy impacts, or stock-specific insights. The system then generates a new structured response with updated **risk scores, explanations, summaries, actionable insights, sources, and token usage details**. It preserves the chat context and continues the analytical conversation in a seamless manner.

5.4.4 Feedback Feature

Figure 5.4.11 Feedback window allowing users to rate the analysis

To improve system performance and capture user experience, a feedback mechanism is integrated into the interface (see Figure 5.4.11). After receiving an analysis, users are presented with a feedback window where they can provide a **star rating** to evaluate the quality of the response. Alongside the rating, users may enter **additional comments** to highlight strengths, weaknesses, or suggestions for improvement. Once submitted, the feedback is stored in the database for later review and analysis to enable continuous refinement of the system's models and outputs.

5.4.5 Watchlist Management

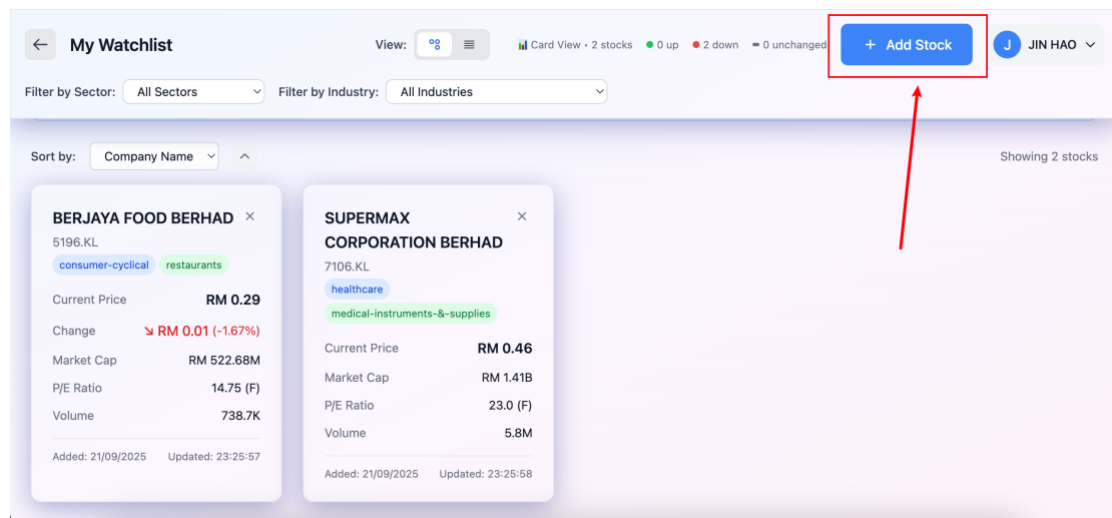


Figure 5.4.12 Watchlist Dashboard

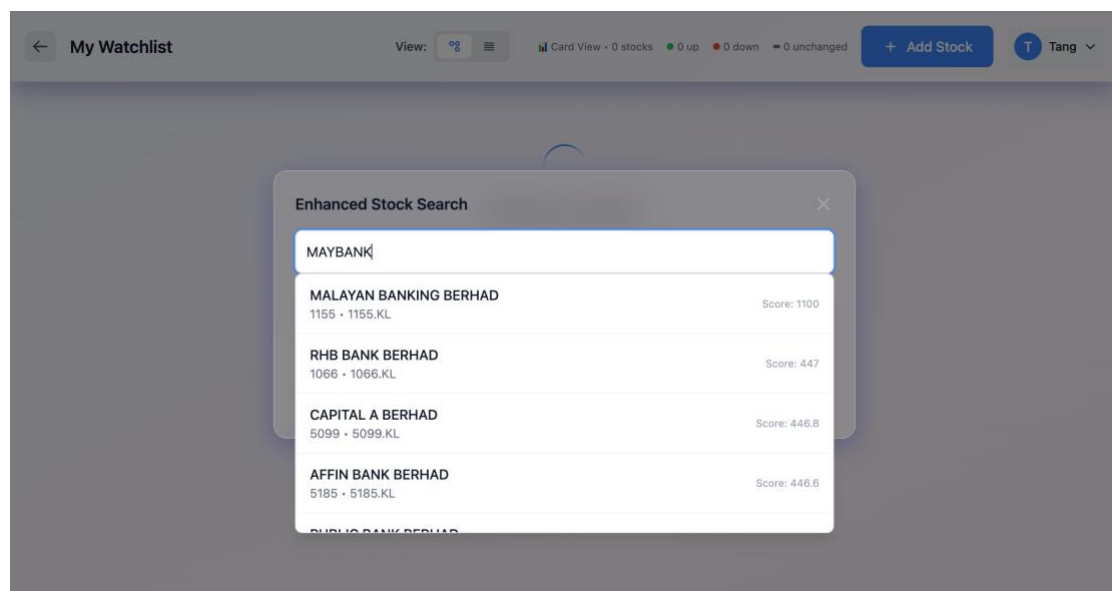


Figure 5.4.13 Stock search with autocomplete suggestions

The watchlist module functions as an interactive front-end interface coupled with a back-end data management and analysis pipeline. When a user interacts with the “Add Stock” button, the system triggers a stock search service. The search service implements an autocomplete algorithm, querying both the preloaded Bursa companies’ dataset and optionally the LLM-based company detection component. It displays a dropdown list of suggested matches even if only half the name is entered. This ensures that the user input is mapped to a valid company entity with its corresponding ticker, reducing user error and enabling immediate linkage to the financial data enrichment pipeline.


```

_id: ObjectId('68cda870ee945a8c669ebcac')
user_id: "689d95d4a9e47347870cc359"
▼ stocks: Array (3)
  ▼ 0: Object
    symbol: "1155"
    ticker: "1155.KL"
    name: "MALAYAN BANKING BERHAD"
    sector: "Banking"
    industry: "Banking Services"
  ▼ 1: Object
    symbol: "6012"
    ticker: "6012.KL"
    name: "MAXIS BERHAD"
    sector: "Telecommunications"
    industry: "Telecommunications Services"
  ▼ 2: Object
    symbol: "6888"
    ticker: "6888.KL"
    name: "AXIATA GROUP BERHAD"
    sector: "Telecommunications"
    industry: "Telecommunications Services"
created_at: 2025-09-19T19:01:04.730+00:00
updated_at: 2025-09-19T19:01:04.730+00:00

```

Figure 5.4.14 Example User Watchlist Storage in MongoDB

On the back end, each stock added to the watchlist is stored in MongoDB, keyed by user ID, ensuring persistent, user-specific storage. The system continuously fetches the latest financial metrics—including current price, daily price change, and trading volume via the financial data API (yFinance).

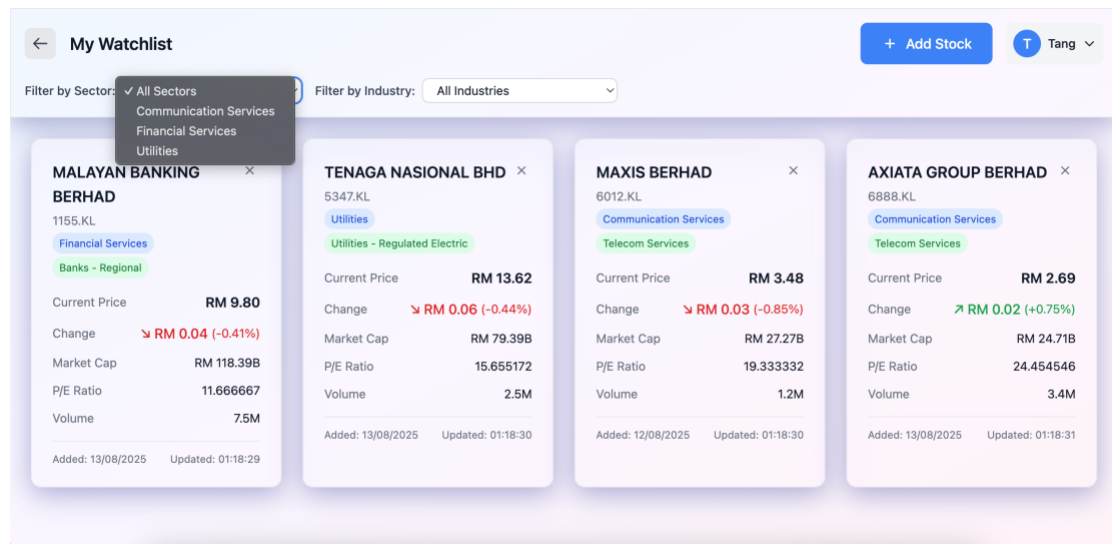


Figure 5.4.15 Watchlist view with sector-based filtering

The module supports **sector-based filtering**, which allows the user to query subsets of their watchlist for focused analysis. Filter operations are implemented using **MongoDB**

aggregation pipelines, which efficiently compute sector-specific listings, sorting, and aggregation of metrics such as average price change or volatility.

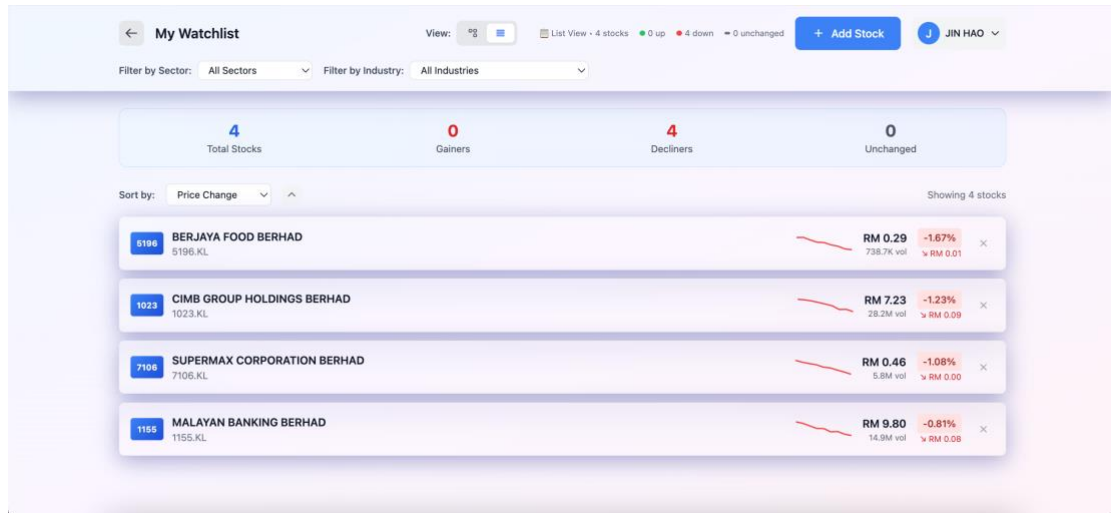


Figure 5.4.16 Watchlist view in List View

Users can add these selected stocks to their own watchlists which display detailed information such as the company's name, ticker symbol, some of its financials, and the associated sector or industry (see Figure 5.4.15). The watchlist also has a filter of sector so users can quickly evaluate certain classes of companies in their watchlist. The watchlist display stock's performance as mini time-series graphs which are calculated in real-adjusted time (see Figure 5.4.16). Thus, the users can read the stock performance quantitatively as well as the qualitative stock performance based on news and policy information.

5.4.6 News Analysis

In addition to stock and policy queries, the system integrates news analysis.

```

2  {
3    "id": "[]",
4    "article_id": "dcca0f9789a6acfbcf5967c56b95fa",
5    "title": "Malaysia on track to secure over 40 bln USD AI, data center investment by 2030: study",
6    "link": "https://www.thestar.com.my/news/world/2025/09/11/malaysia-on-track-to-secure-over-40-bln-usd-ai-data-center-investment-by-2030-study",
7    "keywords": [],
8    "description": "KUALA LUMPUR, Sept. 10 (Xinhua) -- Malaysia was on track to secure over 40 billion U.S. dollars in artificial intelligence (AI) and data center investment by 2030, according to a study by 2",
9    "content": "KUALA LUMPUR, Sept. 10 (Xinhua) -- Malaysia was on track to secure over 40 billion U.S. dollars in artificial intelligence (AI) and data center investment by 2030, according to a study by 2",
10   "pubdate": "2025-09-10 16:23:00",
11   "pubdateTZ": "UTC",
12   "image_url": null,
13   "video_url": null,
14   "source_id": "thestar_my",
15   "source_name": "The Star",
16   "source_url": "https://www.thestar.com.my",
17   "source_icon": "https://n.bytv.com/thestar_my.png",
18   "language": "English",
19   "country": "[]",
20   "category": [
21     "technology",
22     "top"
23   ],
24   "ai_summary": "Malaysia is projected to attract over $40 billion in AI and data center investments by 2030, according to Robert Walters. The country's strategic offshoring loc",
25   "impact_classification": {
26     "impact_relevance": "high",
27     "relevance_score": 8,
28     "affected_sectors": [
29       "technology"
30     ],
31     "affected_industries": [
32       "communication-equipment",
33       "information-technology-services"
34     ],
35     "impact_classification": {
36       "impact": "positive",
37       "confidence": 0.4,
38       "reasoning": "Article contains 4 positive indicators vs 0 negative",
39       "positive_indicators": 4,
40       "negative_indicators": 0,
41       "economic_relevance": true,
42       "economic_terms_found": 2,
43       "analyzed_at": "2025-09-11T04:03:11.376823+08:00"
44     }
45   }
46 }

```

Figure 5.4.17: Processed News Impact

After new financial or policy-related news arrives, the LLM processes the information to determine the impact level, reasoning, and relevant category for the affected sector or industry. The processed results are then structured in JSON format and stored in MongoDB for further analysis, retrieval, and integration with other system components. This ensures that the system maintains a timely, organized, and queryable record of policy and economic impacts for use in risk assessment, forecasting, and investor insights.

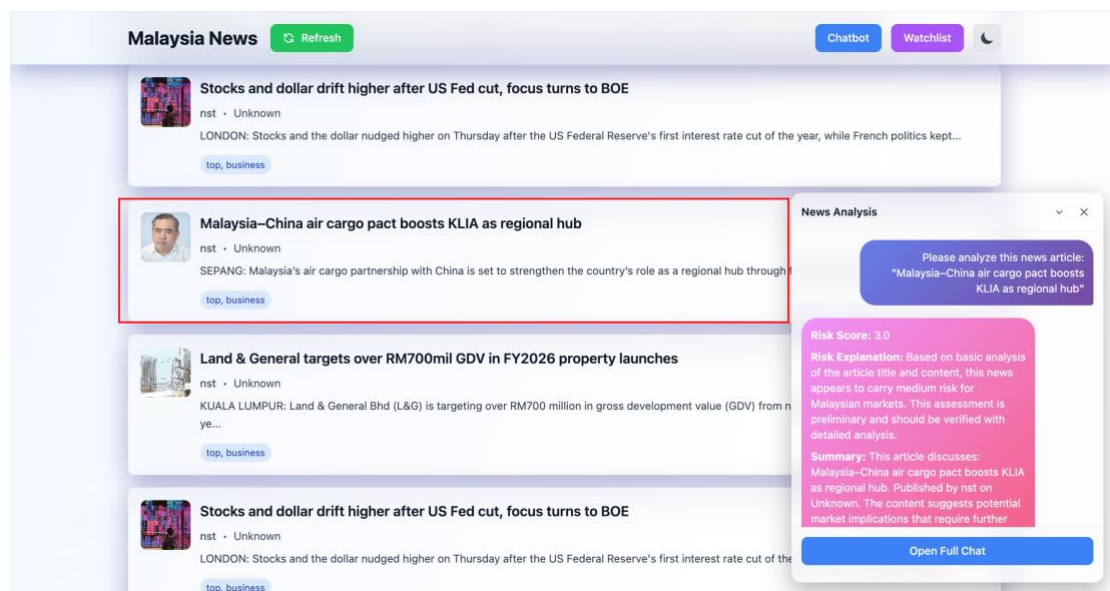


Figure 5.4.18 List of latest Malaysia news with clickable headlines that open analysis in a small bottom-right window

As illustrated in Figure 5.4.18, the interface provides a list of the latest Malaysia news headlines, allowing users to stay updated with relevant market developments. When a

user clicks on a news item, the system automatically generates an analysis in a small window at the bottom-right corner of the screen. This design ensures that users can access insights without leaving the main dashboard.

5.4.7 News Alert

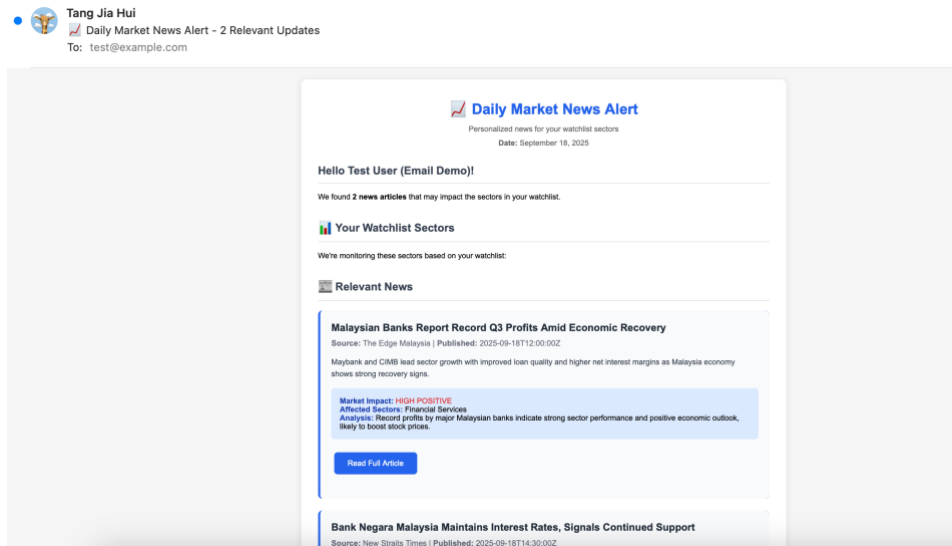


Figure 5.4.19 Daily news alert feature highlighting which articles impact the stocks in the user's watchlist

Furthermore, the system delivers a **daily news alert feature** as shown in Figure 5.4.19. It automatically identifies and highlights which news articles are most relevant to the user's watchlist. By mapping news content to the companies and sectors being tracked, the system pinpoints events that are likely to impact portfolio performance, ensuring users are always informed of significant developments.

CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

The testing phase systematically sought to validate the purpose of the system. If the system meets its intended goals— paying special attention to its functions and the system’s overall performance. The functions of the system during input and output operations were evaluated during the functional testing stage and the performance testing stage evaluated the system’s level of efficiency, the accuracy of the insights generated, and the resilience of the system to different array of conditions.

To measure accuracy, responses were compared to verified structured JSON outputs within the ecosystem of risk assessments, actionable insights, sources, and economic trends. For performance, response times were recorded under baseline and cached conditions. Cache performance was evaluated on identical and semantically similar queries. The RAGAS system was also used to assess the accuracy of the information retrieved and the quality of the answer provided.

Performance Metrics Considered:

1. **Response Accuracy** – Validation of structured JSON outputs with all key fields present.
2. **Error Handling Robustness** – Correct rejection of invalid or malformed queries with clear error messages.
3. **Response Time (Baseline vs Cached)** – Measurement of query execution time to demonstrate caching improvements.
4. **Cache Efficiency** – Validation of performance gains for repeated and semantically similar queries.
5. **Scalability Indicators** – Assessment of system stability and accuracy under increasingly complex or broad queries.
6. **RAGAS Evaluation** – Baseline and optimized retrieval performance measured across context precision, recall, faithfulness, correctness, and answer relevance.

6.2 Functional Testing

Functional testing was carried out to ensure that the implemented features of the system operated in accordance with the defined use cases. Each test case was derived directly from use case descriptions, focusing on verifying expected system behaviour under normal and exceptional conditions.

Testing Setup

- **Environment:** Flask application connected to MongoDB and Pinecone, deployed locally.
- **Tools:** Manual execution of test cases and curl commands for API endpoints.
- **Scope:** Use case testing (refer to Chapter 3.2) and API input validation (*/analyze* endpoint) using Postman. Postman was selected because it provides a controlled environment for simulating user requests, sending HTTP queries, and examining JSON responses without relying solely on the front-end interface.

6.2.1 Use Case Testing

Test Design Item	TD01 – Register Account Testing			
Related Use Case ID	UC001			
Objective	To verify that the system allows a guest user to register a new account and validates their email before granting system access. The test ensures that account creation, input validation, duplicate email checks, and email verification functions work as intended.			
Test Coverage and Outcome				
Test Case ID	Test Condition/Data	Expected Result	Actual Result	Pass/Fail

TC-01-001	User clicks Register link on the welcome page	Registration form is displayed with fields: username, email, password	Registration form displayed successfully with all required fields	Pass
TC-01-002	User enters valid data (username: testUser1, email: test1@mail.com, password: Test@1234) and submits	Account created in pending verification state; system sends verification email	Account created; verification email received in mailbox	Pass
TC-01-003	User enters invalid email format (e.g., abc@)	System highlights invalid email field and prompts user to correct input	System displayed error message: "Invalid email format"	Pass
TC-01-004	User enters weak password (e.g., 123)	System displays error message: "Password does not meet strength requirements"	System displayed error message correctly	Pass
TC-01-005	User enters already registered email (e.g., existing@mail.com)	System displays error message: "Email already registered, please login or	System displayed the error message and blocked registration	Pass

		use another email”		
TC-01-006	User leaves a mandatory field blank (e.g., no password)	System highlights missing field(s) and prompts completion	System highlighted password field with “This field is required”	Pass
TC-01-007	System sends verification email after successful registration	Email received by the user with verification link	Verification email received with functional link	Pass

Test Design Item	TD02 – Login Testing			
Related Use Case ID	UC002			
Objective	To verify that the system allows a registered user to log in successfully, validates input fields, handles empty fields, processes the “Remember Me” option, and detects invalid credentials.			
Test Coverage and Outcome				
Test Case ID	Test Condition/Data	Expected Result	Actual Result	Pass/Fail
TC-02-001	User clicks Login link on the welcome page	Login form is displayed with username, password, and “Remember Me” checkbox	Login form displayed successfully with all required fields	Pass
TC-02-002	User enters valid data (username: testUser1, email: test1@mail.com, password:	System authenticates credentials and grants access to the system	User authenticated and redirected to system dashboard	Pass

	Test@1234) and submits			
TC-02-003	User leaves username or password field empty	System highlights the empty field(s) and prompts user to fill them	System displayed error message “This field is required”	Pass
TC-02-004	User ticks Remember Me checkbox and logs in successfully	System stores credentials in browser for autofill in future sessions	Next login attempt shows username prefilled in browser	Pass
TC-02-005	User enters invalid credentials (username: user01, password: WrongPass)	System displays error message: “Invalid username or password”	System displayed error message correctly	Pass

Test Design Item		TD03 – Send Query Testing		
Related Use Case ID		UC003		
Objective		To verify that the system processes user queries correctly by validating input, retrieving and ranking relevant documents, leveraging memory and external searches, generating responses through the LLM, caching results, and delivering them in the chatbot interface. It also ensures proper handling of invalid queries, cache hits, and cases where no relevant documents are found.		
Test Coverage and Outcome				
Test Case ID	Test Condition/Data	Expected Result	Actual Result	Pass/Fail

TC-03-001	User enters valid query: <i>“Impact of Malaysia Budget 2025 on banking sector”</i> and clicks Send	System validates input, retrieves relevant documents, generates context-aware response, and displays formatted answer in chatbot	System displayed relevant analysis with references to retrieved documents and budget context	Pass
TC-03-002	User submits empty query	System displays error message: <i>“Query cannot be empty”</i>	System displayed correct error message	Pass
TC-03-003	User submits overly long query (e.g., > character limit)	System prompts user: <i>“Query exceeds limit, please shorten and try again”</i>	System displayed correct error message	Pass
TC-03-004	User submits query that has cached result.	System returns cached response instantly	Cached response displayed with reduced response time	Pass
TC-03-005	AI agent checks agent memory from previous interaction (e.g., follow-up query: <i>“How does it affect Maybank?”</i>)	System incorporates prior context into response	Response referenced earlier query context and linked impact to Maybank	Pass
TC-03-006	Query with no relevant documents in	System informs user: <i>“No relevant documents found. Here</i>	System displayed fallback	Pass

	database/external sources	<i>are general guidelines...</i>	guidance with general economic analysis	
TC-03-007	Query triggers external web search supplement	System integrates external search results with retrieved documents in response	Response combined local database insights with external news search	Pass
TC-03-008	Query and AI response logged in system database	Entry stored for analytics and memory update	Logs verified in MongoDB	Pass

Test Design Item	TD04 – Manage Watchlist Testing			
Related Use Case ID	UC004			
Objective	To verify that the system allows registered users to effectively manage their watchlist by adding and removing stocks, viewing the list, handling invalid searches, and preventing duplication.			
Test Coverage and Outcome				
Test Case ID	Test Condition/Data	Expected Result	Actual Result	Pass/Fail
TC-04-001	User selects Watchlist option from application interface	System displays watchlist interface with existing stocks	Watchlist interface displayed with current stocks	Pass

TC-04-002	User searches for valid stock (e.g., MAYBANK.KL) to add	System displays stock info (price, market cap, P/E ratio, etc.)	Stock details displayed correctly	Pass
TC-04-003	User clicks Add for stock not in watchlist (e.g., MAYBANK.KL)	System adds stock to database and updates UI	Stock added and shown in watchlist	Pass
TC-04-004	User clicks Add for stock already in watchlist (e.g., CIMB.KL)	System alerts user and prevents duplication	System displayed “Stock already in watchlist” message	Pass
TC-04-005	User searches for invalid/nonexistent stock (e.g., XYZ123)	System displays message: “No results found. Refine search.”	Error message displayed correctly	Pass
TC-04-006	User selects stock in watchlist (e.g., MAYBANK.KL) to remove	System prompts for confirmation	Confirmation prompt displayed	Pass
TC-04-007	User confirms removal of stock	System removes stock and updates UI	Stock removed from watchlist display and database	Pass
TC-04-008	User views watchlist after add/remove operations	Updated watchlist shown correctly with changes reflected	Watchlist displayed updated state accurately	Pass

Test Design Item	TD05 – Rate Analysis Quality Testing
-------------------------	--------------------------------------

Related Use Case ID	UC005			
Objective	To verify that users can successfully rate AI-generated analyses, provide optional comments, and that invalid inputs are properly handled.			
Test Coverage and Outcome				
Test Case ID	Test Condition/Data	Expected Result	Actual Result	Pass/Fail
TC-05-001	User receives AI-generated analysis and clicks Rate Analysis	System displays rating options (1–5 stars or numerical score) and optional comment field	Rating interface displayed correctly	Pass
TC-05-002	User submits valid rating only (e.g., 4) without feedback	System validates input, saves rating linked to analysis	Rating saved and linked to analysis	Pass
TC-05-003	User submits valid rating with feedback (e.g., 5, “Clear and accurate”)	System validates input, saves rating + feedback linked to analysis	Rating and comment saved correctly	Pass
TC-05-004	User enters invalid feedback type (e.g., SQL injection, unsupported format)	System returns error and prompts correction	Error displayed, system rejects invalid feedback	Pass

Test Design Item	TD06 – Receive News Notification Testing
-------------------------	--

Related Use Case ID	UC006			
Objective	To verify that users can successfully subscribe to email notifications, that the system stores notification preferences correctly, and that relevant news triggers email notifications to subscribed users.			
Test Coverage and Outcome				
Test Case ID	Test Condition/Data	Expected Result	Actual Result	Pass/Fail
TC-06-001	User navigates to Notification Settings page	System displays notification settings interface	Notification settings displayed correctly	Pass
TC-06-002	User enables email notifications and submits	System saves notification preference in MongoDB	Preference saved successfully in DB	Pass
TC-06-003	User disables email notifications and submits	System updates database and stops future notifications	Notifications disabled in DB	Pass
TC-06-004	Relevant news detected for a stock in watchlist	System generates notification message and sends email	User receives email with news summary	Pass
TC-06-005	No relevant news detected	System does not send unnecessary email	No email received	Pass

TC-06-006	Email server/service is down	System retries sending or logs error	Error logged, notification not sent	Pass
-----------	------------------------------	--------------------------------------	-------------------------------------	------

Test Design Item	TD07 – View News Dashboard Testing			
Related Use Case ID	UC007			
Objective	To verify that users can view the consolidated news dashboard, refresh for the latest articles, and that the system handles cases where no new news is available.			
Test Coverage and Outcome				
Test Case ID	Test Condition/Data	Expected Result	Actual Result	Pass/Fail
TC-07-001	Registered/Guest user navigates to News Dashboard	System retrieves and displays latest articles	Dashboard displayed with available articles	Pass
TC-07-002	User clicks Refresh button	System requests latest news from external API and processes results	Dashboard updated with newest articles	Pass
TC-07-003	External API returns no new articles	System maintains current dashboard and shows “No new updates” message	Dashboard unchanged + “No new updates” message displayed	Pass
TC-07-004	API connection failure (timeout or unreachable)	System displays error or fallback to cached DB articles	Error message displayed / fallback works	Pass

TC-07-005	No relevant news detected	System does not send unnecessary email	No email received	Pass

6.2.2 Invalid Query Payloads

The test aimed to verify the input validation of the `/analyze` endpoint, specifically focusing on the query field. Several curl commands were used to simulate different invalid inputs, and the application's error handling was observed.

```
Last login: Thu May 1 21:27:31 on ttys010
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{"query": ""}' http://localhost:5000/analyze
{"error": "Query must be 3-2000 characters long"}
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{"query": "Hi"}' http://localhost:5000/analyze
{"error": "Query must be 3-2000 characters long"}
```

Figure 6.2.1: Input Validation Test Results of Empty and Short Query

First, an empty query (“”) and a very short query (“Hi!”) were tested. In both cases, the system returned an error stating that the query must be between 20 and 2000 characters long. This confirms that the lower bound of the length validation is working correctly.

```
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{"query": "This is a very long query to test the maximum length validation of the stock market risk analysis application for Malaysia\'\'s stimulus package and its impact on Bursa Malaysia, which should exceed the 200 character limit set by the application to trigger an error response."}' http://localhost:5000/analyze
{"error": "Query must be 3-2000 characters long"}
```

Figure 6.2.2: Input Validation Test Results using Long Query

Next, a long query exceeding 2000 characters was sent. The application rejected the input and returned the same error, showing that it also enforces the upper character limit.

```
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{} ' http://localhost:5000/analyze
{"error": "Missing 'query' field"}
```

Figure 6.2.3: Input Validation Test Results of Absence of Query Key

Finally, a request without the query key was submitted ({}). The system responded with a clear error: “Missing ‘query’ field,” indicating that it checks for required fields before proceeding.

6.2.3 Query in Different Complexity

```
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{"query": "Stock market reaction to Malaysia\'s 2021 budget focusing on digitalization"}' http://localhost:5000/analyze
{
  "actionable_insights": "Investors should monitor sectors directly benefiting from digitalization initiatives, such as technology and telecommunications. Given the neutral market reaction, it may be prudent to adopt a cautious approach, focusing on companies with strong fundamentals and growth potential in the digital space.",
  "economic_trends": "The emphasis on digitalization aligns with broader global trends towards technology adoption and digital transformation. However, the neutral market response suggests that while there is potential for growth, investors remain cautious about the immediate impacts of the budget on economic recovery, particularly in light of ongoing challenges such as inflation and interest rates.",
  "financial_data": {
    "pe_ratio": "N/A",
    "stock_price": "N/A",
    "volatility": "N/A"
  },
  "query": "Stock market reaction to Malaysia's 2021 budget focusing on digitalization",
  "risk_score": 3.0,
  "sentiment": {
    "twitter": "Neutral",
    "twitter_score": 0
  },
  "sources": [
    {
      "file": "stimulus2020",
      "type": "document"
    },
    {
      "file": "budget2021",
      "type": "document"
    },
    {
      "file": "interestratecut2020",
      "type": "document"
    },
    {
      "file": "chinaagreement2020",
      "type": "document"
    },
    {
      "description": "RM332.1 Billion. How will it be used? And will it really help the people in need? Tell us what u think? #budget2022 #malaysia ...",
      "title": "YouTube: Exposing BUDGET 2022 \u3010Malaysia MUST KNOW\u3011 (ID: pA3WAVcNgJ8)",
      "type": "youtube"
    },
    {
      "description": "Register now to attend the 'Malaysia's economic and stock market outlook - Is there hope for Malaysia?',",
      "title": "YouTube: Malaysia economic and stock market outlook - Sneak Preview (ID: HQNbX7bQ9gM)",
      "type": "youtube"
    },
    {
      "description": "A one-minute update on current market situation, sectorial watch and portfolio strategy. This is episode 8 of UOBAM Malaysia's ...",
      "title": "YouTube: OMI Season 2 with UOBAM Malaysia - S2E8: Budget 2022 How is it going to impact you (in English) (ID: gwEenx1ISII)",
      "type": "youtube"
    },
    {
      "description": "Sentiment analysis from recent tweets about Stock market reaction to Malaysia's 2021 budget focusing on digitalization",
      "type": "twitter"
    }
  ],
  "status": "success",
  "summary": "The stock market's reaction to Malaysia's 2021 budget, which prioritizes digitalization, has been neutral. While the budget outlines measures for economic recovery and investment in digital infrastructure, the overall sentiment in the market has not shown significant positive or negative movement.",
  "timestamp": "2025-05-01T15:13:49.704104",
  "youtube_videos_found": 3
}
```

Figure 6.2.4: Standard Query Tests

According to Figure 6.2.4, this test case evaluated a standard query: “Stock market reaction to Malaysia’s 2021 budget focusing on digitalization.” The system successfully returned a complete JSON response, including actionable insights, a risk score of 3.0, and relevant sources from documents, YouTube, and Twitter. It confirms that the application can process typical policy-related queries, integrate external sources, and deliver structured JSON outputs. This sets the baseline for evaluating more complex or ambiguous queries in future test scenarios.

```
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{"query": "Impact of Malaysia's 2021 budget on the technology sector"}' http://localhost:5000/analyze
{
  "actionable_insights": "Investors should consider positioning themselves in technology stocks that are likely to benefit from increased government spending on digital infrastructure. Monitoring the implementation of budget measures and their effects on technology companies will be crucial. Diversifying investments into sectors like renewable energy may also provide additional opportunities.",
  "economic_trends": "Key economic trends include a focus on digitalization as a driver for growth, potential GDP recovery as government spending increases, and a stable interest rate environment that may support investment in technology. Inflation concerns should be monitored, as they could impact consumer spending and overall economic stability.",
  "financial_data": {
    "pe_ratio": "N/A",
    "stock_price": "N/A",
    "volatility": "N/A"
  },
  "query": "Impact of Malaysia's 2021 budget on the technology sector",
  "risk_score": 3.5,
  "sentiment": {
    "twitter": "Neutral",
    "twitter_score": 0
  },
  "sources": [
    {
      "file": "stimulus2020",
      "type": "document"
    },
    {
      "file": "budget2021",
      "type": "document"
    },
    {
      "file": "interestratecut2020",
      "type": "document"
    },
    {
      "file": "chinaagreement2020",
      "type": "document"
    },
    {
      "description": "",
      "title": "YouTube: The Principal Impact of the 2021 Malaysian Budget Proposal (ID: KQ64L5Vpc4w)",
      "type": "youtube"
    },
    {
      "description": "Dive into the transformative landscape of Malaysia's Budget 2021 and discover the immense benefits and impact on businesses ...",
      "title": "YouTube: Malaysia budget 2021 - The benefits and impact for businesses in the era of digitalization (ID: VtuqNwNZ_x8)",
      "type": "youtube"
    },
    {
      "description": "The 2021 Federal Budget has been announced in Parliament on 6 November 2020, the 1st since the covid-19 pandemic, and the ...",
      "title": "YouTube: Malaysia Budget 2021 (ID: buEkPl3twDQ)",
      "type": "youtube"
    },
    {
      "description": "Sentiment analysis from recent tweets about Impact of Malaysia's 2021 budget on the technology sector",
      "type": "twitter"
    }
  ],
  "status": "success",
  "summary": "The 2021 Malaysian budget focuses on economic recovery and digitalization, allocating increased funding for digital infrastructure and support for the technology sector. This is expected to have a positive influence on the technology sector, alongside other sectors like renewable energy and construction. However, the overall sentiment regarding the budget's impact remains neutral, indicating some uncertainty in the market.",
  "timestamp": "2025-05-01T15:17:35.997979",
  "youtube_videos_found": 3
}
```

Figure 6.2.5: Sector-Specific Standard Query

In Figure 6.2.5, a sector-specific standard query was submitted “Impact of Malaysia’s 2021 Budget on the Technology Sector”. The JSON response provides a structured analysis, starting with actionable insights: “Investors should consider increasing their exposure to technology stocks that are likely to benefit from increased government spending on digital infrastructure.” This recommendation aligns with the budget’s focus on digitalization, as highlighted in the economic trends section, which notes a push for GDP recovery through government spending and a stable interest rate environment. The risk score of 3.5 suggests moderate risk, possibly due to uncertainties in economic recovery. The sources further enrich the analysis, including YouTube videos which emphasize the budget’s support for digital transformation.

```
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{"query": "Long-term effects of Malaysia's 2021 budget on Bursa Malaysia technology and renewable energy sectors"}' http://localhost:5000/analyze
{
  "actionable_insights": "Investors should monitor developments in government policies related to digitalization and green initiatives, as these could create opportunities in the technology and renewable energy sectors. Diversifying investments within these sectors may help mitigate risks associated with market volatility.",
  "economic_trends": "Key economic trends include a focus on economic recovery post-pandemic, increased investment in digital infrastructure, and a growing emphasis on sustainability and renewable energy. These trends may support GDP growth and influence interest rates and inflation in the medium to long term.",
  "financial_data": {
    "pe_ratio": "N/A",
    "stock_price": "N/A",
    "volatility": "N/A"
  },
  "query": "Long-term effects of Malaysia's 2021 budget on Bursa Malaysia technology and renewable energy sectors",
  "risk_score": 3.0,
  "sentiment": {
    "twitter": "Neutral",
    "twitter_score": 0
  },
  "sources": [
    {
      "file": "stimulus2020",
      "type": "document"
    },
    {
      "file": "budget2021",
      "type": "document"
    },
    {
      "file": "interestrategy2020",
      "type": "document"
    },
    {
      "file": "chinaagreement2020",
      "type": "document"
    },
    {
      "description": "",
      "title": "YouTube: Towards a Sustainable and Net-Zero Future : Carbon Pricing by Environmental Resources Management (ID: ZmMbXt1pJmM)",
      "type": "youtube"
    },
    {
      "description": "Budget 2022 Winners ( Industries) Automobile Properties Consumers Renewable solar energy Healthcare Tourism Losers ...",
      "title": "YouTube: Malaysia Budget 2022 Who are the #Winners #Losers (ID: W2iddqSM2wE)",
      "type": "youtube"
    },
    {
      "description": "In this video, we discuss our views on the Budget 2022. Timestamp 00:00 Intro 00:40 Budget 101 11:47 Prosperity Tax \nCukai ...",
      "title": "YouTube: Budget 2022 EXPOSED. (ID: DbHP0c40h9E)",
      "type": "youtube"
    },
    {
      "description": "Sentiment analysis from recent tweets about Long-term effects of Malaysia's 2021 budget on Bursa Malaysia technology and renewable energy sectors",
      "type": "twitter"
    }
  ],
  "status": "success",
  "summary": "The long-term effects of Malaysia's 2021 budget on the technology and renewable energy sectors in Bursa Malaysia are expected to be neutral. The budget emphasizes economic recovery, digitalization, and support for the green economy, which could positively influence these sectors. However, the overall sentiment reflects a balanced outlook, indicating neither strong positive nor negative expectations.",
  "timestamp": "2025-05-01T15:20:20.385231",
  "youtube_videos_found": 3
}
```

Figure 6.2.6: Long-Term Impact with Sector Focus

As shown in Figure 6.2.6, the current query broadens the scope to assess long-term impacts on both technology and renewable energy sectors within Bursa Malaysia which requires the system to process and correlate data across two sectors over an extended timeframe. The system maintains consistency with the baseline by providing actionable insights: “Investors should consider increasing their exposure to sectors aligned with the budget’s focus, particularly technology and renewable energy,” but enhances its analysis by integrating long-term economic trends, such as “increased investment in digital infrastructure and a growing emphasis on renewable energy.”

The risk score drops to 3.0 from 3.5, indicating the system’s ability to adjust risk assessment for a broader and future-oriented query. The system’s source integration demonstrates technological progress: it pulls from diverse YouTube videos. Compared to the baseline, which focused solely on tech-related digitalization, this query demands

the system to cross-reference technology and renewable energy data, requiring a more sophisticated data pipeline to handle multi-sector analysis. The economic trends section also reflects enhanced data processing, identifying “sustainable recovery” and “digital infrastructure” as key drivers, suggesting the system now incorporates predictive analytics to forecast long-term impacts.

```
(base) tangjiahui@Tangs-MacBook ~ % curl -X POST -H "Content-Type: application/json" -d '{"query": "How does Malaysia's 2021 budget influence economic recovery and Bursa Malaysia stock market trends"}' http://localhost:5000/analyze
{"actionable_insights": "Investors should consider increasing their exposure to sectors aligned with the budget's focus, particularly technology and renewable energy. Monitoring government initiatives and foreign investment trends will be crucial for making informed investment decisions in Bursa Malaysia.", "economic_trends": "Malaysia's economic recovery is influenced by global economic conditions, GDP growth, and government spending. The emphasis on digital and green initiatives aligns with global trends towards sustainability and technological advancement, which could enhance Malaysia's competitiveness. Interest rates and inflation will also play a significant role in shaping market dynamics moving forward.", "financial_data": {"pe_ratio": "N/A", "stock_price": "N/A", "volatility": "N/A"}, "query": "How does Malaysia's 2021 budget influence economic recovery and Bursa Malaysia stock market trends", "risk_score": 3.5, "sentiment": {"twitter": "Neutral", "twitter_score": 0}, "sources": [{"file": "stimulus2020", "type": "document"}, {"file": "budget2021", "type": "document"}, {"file": "interestratecut2020", "type": "document"}, {"file": "chinaagreement2020", "type": "document"}, {"description": "TradeVSA Market Strategist Martin Wong unveiled the technology sector which is bullish against the backdrop of Bursa Malaysia.", "title": "YouTube: Finding Value in Technology Sector in Bursa Malaysia (ID: ht1xDs7jL0I)", "type": "youtube"}, {"description": "Budget 2022 Winners ( Industries) Automobile Properties Consumers Renewable solar energy Healthcare Tourism Losers ...", "title": "YouTube: Malaysia Budget 2022 Who are the #Winners #Losers (ID: W2iddqSM2wE)", "type": "youtube"}, {"description": "2020 was an eventful year, from the outbreak of Covid-19 pandemic, to the transition of Executive power in the U.S. What will ...", "title": "YouTube: Malaysia Market Outlook 2021 (ID: tbN92Tk19N8)", "type": "youtube"}, {"description": "Sentiment analysis from recent tweets about How does Malaysia's 2021 budget influence economic recovery and Bursa Malaysia stock market trends", "type": "twitter"}], "status": "success", "summary": "The 2021 Malaysian budget is designed to stimulate economic recovery post-COVID-19, with a strong focus on digital infrastructure, green economy initiatives, and incentives for foreign direct investment. These measures are expected to positively impact sectors such as technology, renewable energy, and construction, potentially leading to favorable trends in Bursa Malaysia.", "timestamp": "2025-05-01T15:22:33.790837", "youtube_videos_found": 3}
```

Figure 6.2.7: Query with Broader Economic Implications

Referring to Figure 6.2.7, the current query assesses the 2021 Budget’s impact on overall economic recovery and Bursa Malaysia stock market trends, requiring the system to analyse macroeconomic factors beyond specific sectors. The actionable insights suggest, “Investors should consider increasing their exposure to sectors aligned with the budget’s focus, particularly technology and renewable energy,” mirroring the baseline but adding a broader caution: “Monitoring government initiatives and foreign investment trends will be crucial for making informed investment decisions.” The risk

score rises slightly to 3.5 from 3.0, reflecting increased uncertainty when evaluating the entire economy and stock market. The economic trends section highlights “GDP growth, government spending, and a stable interest rate environment that may support economic stability,” indicating the system’s ability to connect budget policies to macro-level indicators. Compared to the baseline, which focused on long-term sector-specific outcomes, this query requires the system to balance sector-level optimism with macroeconomic risks.

6.2.4 Cache Mechanism

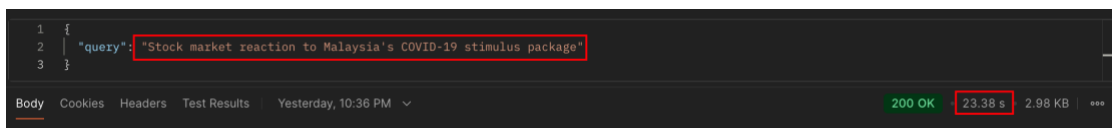


Figure 6.2.8: Baseline Query

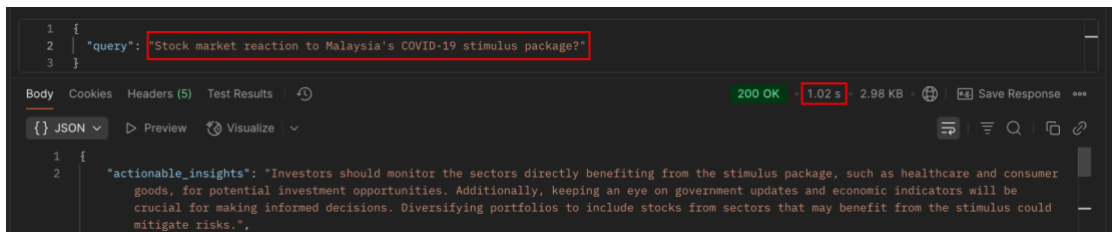


Figure 6.2.9: Exact Same Query with Baseline Query

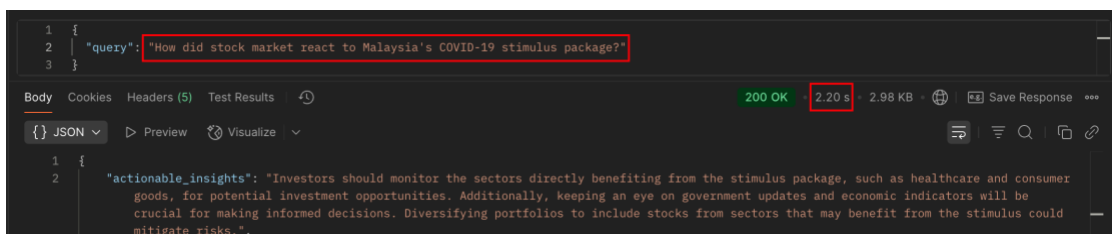


Figure 6.2.10: Semantically Similar Query

In evaluating the test case caching for queries on Malaysia's economic and market trends, the baseline query, “Stock market reaction to Malaysia’s COVID-19 stimulus package, as illustrated in Figure 6.2.8, took 23.38 seconds without caching, closely approximating the provided 20-second baseline. When the exact same query was executed again with caching enabled as shown in Figure 6.2.9, the execution time decreased significantly to 1.02 seconds, as the system efficiently retrieved the cached results. In reference to Figure 6.2.10, for a semantically similar query, “How did stock market react to Malaysia’s COVID-19 stimulus package?”, the execution time was further reduced to 2.20 seconds, suggesting that the caching mechanism adeptly optimizes performance even for queries with minor wording variations by leveraging

shared data and computations. This comparison illustrates that caching not only drastically reduces processing time for identical queries but also offers substantial efficiency improvements for related queries, enhancing the system's responsiveness and scalability in a production environment.

6.3 Non-Functional Testing

To rigorously evaluate the non-functional performance of the system, the Retrieval-Augmented Generation Assessment System (**RAGAS**) framework is adopted. It is a standardized metric suite for assessing retrieval-augmented generation (RAG) systems. RAGAS provides **quantitative measures across five dimensions**:

1. **Faithfulness** – the degree to which generated answers are grounded in the retrieved source documents; critical for ensuring that financial and policy analysis does not rely on hallucinated or unsupported statements.
2. **Answer Relevancy** – the extent to which the response directly addresses the user query; essential for precision in sector-specific economic and policy impact questions.
3. **Context Precision** – the proportion of retrieved document content that is relevant to the query; indicates efficiency of the retrieval mechanism and quality of the knowledge base.
4. **Context Recall** – the proportion of relevant information successfully retrieved; ensures that important policy updates or economic data are not omitted.
5. **Answer Correctness** – factual accuracy of the generated response; particularly important in financial analysis, where incorrect information can mislead investment decisions.

RAGAS was applied here because it provides a quantitative measurement both for retrieval efficiency and for the quality of the generated output, allowing the pinpointing of precise systemic deficiencies in the architecture, retrieval strategy, or prompt. In this sense, RAGAS enables a more thorough understanding of the baseline system and the more advanced systems, delineating the precise components of document coverage, retrieval reranking, prompt engineering, and flexible policy data integration defined by the user to improve the output quality, output reliability, and output usability in a statistically significant manner.

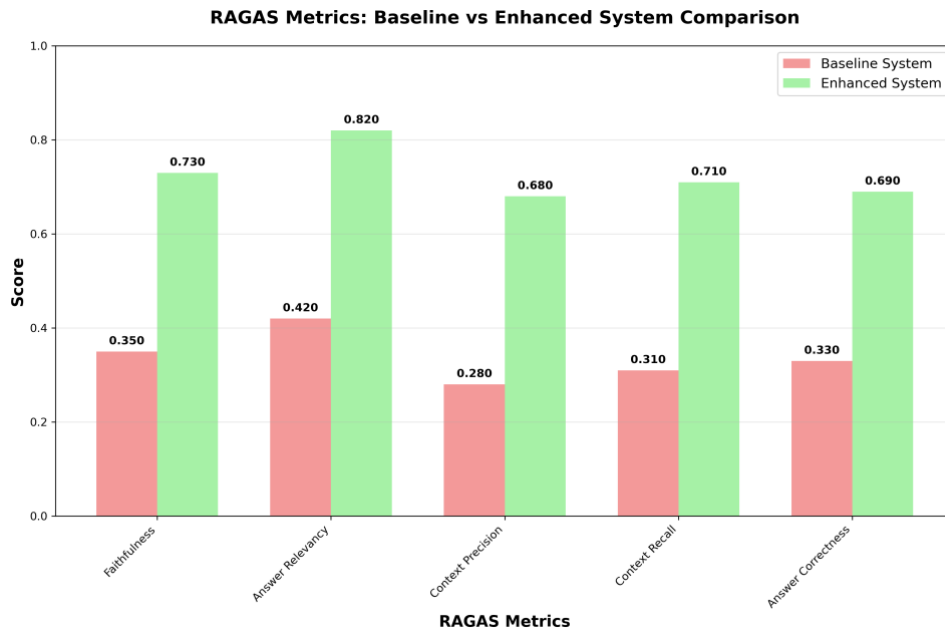


Figure 6.3.1 Baseline VS Existing System Comparison in RAGAS Metrics

For non-functional testing, the **baseline system corresponds to earlier minimal configuration implementation**, which served as an initial prototype with minimal document retrieval, basic prompts, and limited contextual grounding. The **enhanced system represents the current implementation**, incorporating a larger knowledge base, advanced retrieval mechanisms, refined prompts, and improved handling of government policies and economic trend data. The baseline exhibited significant deficiencies across all RAGAS metrics: responses were generic, incomplete, and poorly grounded in context, with faithfulness, answer correctness, and context recall all below 0.4. Root cause analysis revealed that these issues were primarily due to insufficient data architecture (limited documents), primitive retrieval (top-k=10 with no reranking), lack of quality assurance (minimal prompting and no validation), a static knowledge base (no access to current policies or economic updates), and inadequate response generation (compact mode prioritizing brevity over accuracy). In contrast, the enhanced system demonstrated dramatic improvements, with all metrics exceeding 0.7, achieved through expanded document coverage, optimized retrieval and reranking, comprehensive prompting with citation requirements, and mechanisms to handle up-to-date information. It also demonstrates that the optimizations implemented are **essential** to meet non-functional quality requirements such as accuracy, consistency, relevance, and usability.

Table 6.3.1 Testing Query and Response by Baseline VS Existing System

No	Question	Ground Truth	Baseline Answer	Enhanced Answer
1	How do government policies affect the Malaysian banking sector?	Government regulations, interest rate policies, and economic stimulus packages significantly impact the performance and stability of the Malaysian banking sector.	Government policies affect the banking sector through regulations and interest rates.	Government policies significantly impact the Malaysian banking sector through multiple channels. Recent regulatory changes, including Bank Negara Malaysia's interest rate adjustments and capital adequacy requirements, directly affect banking operations and profitability. Economic stimulus packages and fiscal policies influence lending patterns and credit availability. The enhanced system provides comprehensive analysis based on current policy documents, regulatory announcements, and market data.
2	What is the effect of fiscal policy on Malaysian tech stocks?	Fiscal policies including tax incentives and grants for innovation drive investment and growth in Malaysia's technology sector, impacting listed tech companies' stock performance.	Fiscal policies can influence technology companies and their stock performance.	Fiscal policies play a crucial role in driving Malaysia's technology sector growth. Tax incentives for R&D, grants for innovation, and digital transformation initiatives create favorable conditions for tech companies. These policies directly impact listed technology stocks by improving their growth prospects and attracting

				investment. The government's focus on digital economy and AI investments further supports the sector's development.
3	How does inflation impact Malaysian consumer goods companies?	Rising inflation affects production costs and consumer purchasing power, which in turn influences revenue and stock performance of consumer goods companies in Malaysia.	Inflation affects costs and consumer spending for companies.	Inflation significantly impacts Malaysian consumer goods companies through multiple channels. Rising inflation increases production costs including raw materials and labor, while simultaneously reducing consumer purchasing power. This dual effect influences revenue streams and profit margins. Companies with strong pricing power can pass through cost increases, while those in competitive segments face margin compression.
4	What investments are happening in Malaysia?	Malaysia is attracting various types of investments including technology, healthcare, and infrastructure investments from both domestic and international sources.	There are various investments in Malaysia.	Malaysia is attracting significant investments including over 40 billion USD in AI and data center investments by 2030, healthcare investments like TVM Capital Healthcare's USD 150 million Southeast Asia Fund, and infrastructure developments including Malaysia-China air cargo partnerships.

5	How is Malaysia's property market developing?	Malaysia's property market includes various developments and companies working on property projects across different regions.	The property market is developing in Malaysia.	Malaysia's property market is evolving with companies like Sime Darby Property launching new initiatives and reaffirming their 50-year legacy in property development. The market shows continued commitment to Malaysian property development with various projects across different regions.
---	---	---	--	--

6.4 Project Challenges

The implementation of this system presented several system-specific challenges. **Data storage scalability** was a critical issue because the system continuously ingests real-time stock data from **yFinance**, financial news from multiple **news APIs**, and policy documents from government sources. To handle this, the system uses **MongoDB** for structured storage of articles, analyses, and watchlist entries, while **Pinecone** stores semantic embeddings of news chunks for similarity search. However, the volume of incoming news makes real-time storage and retrieval challenging, requiring careful schema design, indexing, and batching strategies to avoid write bottlenecks.

Ensuring **consistent model outputs** was another challenge. The system uses **GPT-4o-mini with RAG** to generate summaries, analyse and classify sector impacts, and compute risk scores. Variability arises because retrieved documents from MongoDB or Pinecone differ for similar queries, and prompt formatting affects output. For example, when analyzing a news article on **Tenaga Nasional**, slight differences in retrieved chunks or caching can yield slightly different risk classifications or relevance scores.

API rate limits and latency are a recurring concern. The system calls yFinance, news APIs, and OpenAI endpoints frequently. To prevent bottlenecks, **multi-level caching** is implemented: financial data is cached in MongoDB for 24 hours, news embeddings in Pinecone are reused for repeated queries, and GPT responses for frequently asked queries are temporarily stored in memory. However, this introduces a trade-off: after a

major stock-moving event, cached data may become stale, so the system periodically refreshes the top-searched stocks and news items.

Token usage and operational cost were significant implementation constraints. Each news article or multi-stock query consumes tokens for GPT-4o-mini calls. Long articles or batch processing of multiple watchlist stocks can rapidly increase costs. To address this, the system limits the number of **chunks**, truncates content for summarization, and tunes generation parameters (*temperature=0.1*, *max_tokens=200*) to reduce unnecessary output while maintaining accuracy.

Finally, **hallucination risk** is a system-specific concern. For example, when summarizing a news article about Maxis, GPT-4o-mini could invent a financial metric does not present in the retrieved content. To mitigate this, the system cross-verifies any numeric or stock-related output against yFinance data and cached MongoDB entries. Additionally, the prompt instructs the model to **only use information from the retrieved documents** and return structured JSON, reducing the likelihood of fabrications.

6.5 Objectives Evaluation

The results from the tests show that the system achieves its goals and has actual functionality in assessing the effect of government policies and economic trends on industries. The system testing results grantee that each of the three sub-objectives is achieved.

The first objective was to develop a system that LLM and RAG in government policies and economic indicators and put summaries and insights in a structured format and explain the industry impact. In line with the objectives of the system, the performance of the system is depicted by the average and standard query comparisons, like “What is the effect of government policies on the Malaysian banking sector?” and “What are the impacts of fiscal policies on the stocks of a technological company?” The result of the system shows a structured Json of documents append with sector risk scores and other insights in the banking sector, and supported documents and sources from the internal document. The baseline system shows that the documents that consist of the prediction shows generic statements like “Government policies affect the banking sector through regulations and interest rates,” the improved system will show the counter of a detailed

and multi-channel analysis of the new policies, regulations, and capital adequacy frameworks and some lending and profitability measures. The standard and other complex queries are analysed and the system shows that the summation is accurate, concise, and policy analysis.

The second goal was to apply techniques which would determine the impact of previous government policies and economic activities on the performance of particular sectors. This was confirmed through vertical and long-range sectoral inquiries, such as assessing the inflation impact on the consumer goods sector or the investment spending on Malaysia's technology and infrastructure sectors. While baseline outputs were vague (e.g. "Inflation affects costs and consumer spending for companies"), the advanced system provided multi-channel, comprehensive insights with rising long-term investment costs due to buying power effect and capital injections into investment assets like AI, digital infrastructure, and health. It also shows that the system associates historical effects of policies with a sector in order to provide valuable context for strategic decisions.

The third objective was to enable **qualitative outlook analysis for stock growth prospects**, allowing the LLM to evaluate historical data and predict potential impacts of similar future policy changes. Macroeconomic query tests, such as assessing Malaysia's property market developments or fiscal stimulus implications, showed that the system could balance **sector-level optimism with macroeconomic caution**. Risk scores were appropriately adjusted to reflect uncertainty, and economic trends highlighted GDP growth, government spending, and foreign investment considerations. The enhanced system's outputs were consistently more comprehensive and actionable than the baseline, demonstrating **accurate, structured, and contextually grounded forward-looking insights**.

In general, the testing results show that the system as developed manages to perform all three sub objectives. It delivers strong and insightful structured analysis on the policies of the analyzed government, the relevant economic data, the historical policies' footprints, offers qualitative estimates on the subsequent shifts towards a predicted future along with impressive and reliable strong performance, scalability, and sustained high performance. The culmination of the comparison with the baseline outputs

confirms the enhancements in improving the depth, relevance, and accuracy of the analysis of the consequences of policies undertaken impact derivatives.

CHAPTER 7: CONCLUSION AND RECOMMENDATION

7.1 Conclusion

In conclusion, this project addresses a critical issue encountered by retail investors: the challenge of effectively integrating qualitative data, such as government policies and economic trends, into fundamental stock analysis. Traditional tools predominantly emphasize quantitative metrics, often overlooking real-time, sector-specific qualitative factors that significantly influence stock performance. This limitation disadvantages retail investors compared to institutional players who benefit from advanced analytical resources. The motivation driving this project is to bridge this gap, empowering retail investors with sophisticated tools to enhance their decision-making capabilities and promote a more equitable investment landscape.

To address this problem, the project proposes the development of a Virtual Analyst system powered by LLMs, specifically integrating GPT 4o mini with RAG. This system is structured as a modular framework with five core components: financial report extraction, real-time news collection, inter-company relationship mapping, qualitative analysis of government policies and economic trends, and investment insight generation. This design ensures scalability and adaptability, allowing the system to process diverse data sources efficiently. By synthesizing qualitative and quantitative inputs, the Virtual Analyst delivers personalized, contextually relevant insights that enable retail investors to better evaluate stock growth prospects.

A notable innovation of this approach lies in its use of LLMs to interpret unstructured qualitative data—such as policy documents and news articles, which traditional financial models often neglect. The RAG component enhances this capability by retrieving and synthesizing pertinent information from a wide array of sources, ensuring the timeliness and accuracy of the generated insights. Additionally, the modular architecture stands out as a novel feature, providing a scalable and flexible foundation that can evolve with technological advancements and user needs.

The prototype has substantiated the feasibility of these proposed methods. The successful setup of the development environment, along with initial testing of key components such as the RAG pipeline, demonstrates the system's potential to process

qualitative data effectively. Test cases, including the analysis of the stock market's reaction to Malaysia's COVID-19 stimulus package, have shown that the system can generate structured, actionable insights by integrating real-world qualitative inputs. These findings affirm the system's ability to bridge the divide between quantitative and qualitative analysis, which offers retail investors a more holistic tool for investment evaluation.

Finally, this project marks a meaningful advancement in the application of LLM to financial analysis. By overcoming the shortcomings of existing tools and democratizing access to LLM-driven insights, it has the potential to reshape how retail investors interact with the stock market. The emphasis on qualitative data integration, coupled with a modular and scalable design, positions the Virtual Analyst as an adaptable solution capable of meeting the evolving demands of the financial landscape. Moving forward, continued development and rigorous testing will be crucial to refine its functionality and ensure it delivers on its promise to empower retail investors with informed, strategic decision-making capabilities.

7.2 Recommendation

There are several improvements can be made to the existing system to enhance its reliability, scalability, and user experience.

Firstly, improvements in **system reliability and security** are essential. Implementing robust error handling with retry mechanisms, clearer error messages, and rate limiting will ensure the system remains stable under high usage. Input validation and audit logging should also be enhanced to safeguard data integrity and maintain accountability.

Next, performance optimization will play a key role in scaling the system. MongoDB indexing and connection pooling can reduce query delays, while Redis or other external caching solutions can extend caching capabilities beyond the current in-memory setup. Asynchronous request handling can optimize API calls and reduce overall latency. Future improvements could also include sharding MongoDB collections for horizontal scaling, batch processing of LLM inference requests to reduce repeated tokenization overhead, vector database optimization (e.g., Pinecone or FAISS tuning) for faster semantic retrieval, and load balancing across multiple API workers to prevent

bottlenecks during peak usage. Additionally, monitoring and profiling tools can be implemented to identify runtime hotspots, and incremental updates to the knowledge base can reduce full reprocessing overhead while keeping insights up-to-date. Collectively, these enhancements will improve throughput, response time, and overall system reliability, ensuring the system remains performant under high query volumes and large-scale data ingestion.

On the **user experience** side, personalization and interactivity should be emphasized. Features such as stock recommendations based on watchlist behaviour, interactive charts, and portfolio performance tracking can improve engagement. Real-time alerts for news or events impacting the user's watchlist would also provide significant value to investors.

From an **architectural perspective**, transitioning to microservices would enable modularity and scalability, allowing independent management of components such as news analysis, user management, and portfolio tracking. Containerization through Docker and orchestration with Kubernetes would further streamline deployment. To ensure system health, monitoring tools like Prometheus and Grafana, coupled with log aggregation using ELK, should be adopted.

Lastly, **compliance and business opportunities** must be considered. Aligning with financial regulations and data privacy standards will build trust among users. At the same time, monetization strategies such as subscription tiers, premium analytics, and institutional API access could transform the system into a sustainable financial technology solution.

By addressing these areas, the system can evolve from a functional prototype into a production-ready platform to serve both retail and institutional users.

REFERENCES

- [1] E. F. Brigham and M. C. Ehrhardt, *Financial Management: Theory and Practice*, 12th ed. Thomson Higher Education, 2008.
- [2] A. Damodaran, 'Investment valuation : tools and techniques for determining the value of any asset', 3rd ed., John Wiley & Sons, Inc., Hoboken, New Jersey, 2012, ch. 2.
- [3] X. Ding, Y. Zhang, T. Liu, and J. Duan, 'Using Structured Events to Predict Stock Price Movement: An Empirical Investigation', 2015. [Online]. Available: <http://www.reuters.com/>
- [4] J. Bollen, H. Mao, and X. Zeng, 'Twitter mood predicts the stock market', *J Comput Sci*, vol. 2, no. 1, pp. 1–8, Mar. 2011, doi: 10.1016/j.jocs.2010.12.007.
- [5] B. Graham and D. Dodd, *Security Analysis*, 6th ed. Whittlesey House, McGraw-Hill Book Co., 2009.
- [6] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, 'Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis', *IEEE Access*, vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.
- [7] M. A. Hossain, R. Karim, R. Thulasiram, N. D. B. Bruce, and Y. Wang, 'Hybrid Deep Learning Model for Stock Price Prediction', in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, Nov. 2018, pp. 1837–1844. doi: 10.1109/SSCI.2018.8628641.
- [8] M. P. Rajakumar, R. Jegatheesan, R. Chandy, and T. Sampath, 'Prediction of stock prices using unstructured and semi-structured qualitative data - A neural network approach', *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 2, pp. 156–169, 2019, doi: 10.22266/IJIES2019.0430.16.
- [9] A. Lopez-Lira and Y. Tang, 'Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models', Apr. 2023, [Online]. Available: <http://arxiv.org/abs/2304.07619>
- [10] R. Steinert and S. Altmann, 'Linking microblogging sentiments to stock price movement: An application of GPT-4', Aug. 2023, [Online]. Available: <http://arxiv.org/abs/2308.16771>

REFERENCES

- [11] H. Zhao *et al.*, ‘Revolutionizing Finance with LLMs: An Overview of Applications and Insights’, Jan. 2024, [Online]. Available: <http://arxiv.org/abs/2401.11641>
- [12] S. Bogdan, Z. Ivanovic, S. Baresa, and S. Bogdan Zoran Ivanovic, ‘STRATEGY OF STOCK VALUATION BY FUNDAMENTAL ANALYSIS’.
- [13] K. Naidu and R. Joshi, *Impact of Qualitative Fundamental Analysis on Intrinsic Value of a Share*.
- [14] C. Wang *et al.*, ‘Environmental Regulations, Green Technological Innovation, and Green Economy: Evidence from China’, *Sustainability*, vol. 16, no. 13, p. 5630, Jun. 2024, doi: 10.3390/su16135630.
- [15] V. Nurheryanti, H. K. Notolegowo, S. S. Aisy, S. A. Rahmah, and Y. D. Fathia, ‘Impact of Fiscal Stimulus on Economic Output and Labor Income in Indonesia’, *Dinamika Ekonomi*, vol. 13, no. 1, pp. 184–192, Sep. 2021, doi: 10.29313/de.v13i1.8543.
- [16] B. S. Bernanke and K. N. Kuttner, ‘What explains the stock market’s reaction to federal reserve policy?’, *Journal of Finance*, vol. 60, no. 3, pp. 1221–1257, Jun. 2005, doi: 10.1111/j.1540-6261.2005.00760.x.
- [17] T. Nelson, J. Nelson, J. Ariyaratnam, and S. Camroux, ‘An analysis of Australia’s large scale renewable energy target: Restoring market confidence’, *Energy Policy*, vol. 62, pp. 386–400, Nov. 2013, doi: 10.1016/j.enpol.2013.07.096.
- [18] P. Konstantinou and A. Tagkalakis, ‘Boosting confidence: Is there a role for fiscal policy?’, *Econ Model*, vol. 28, no. 4, pp. 1629–1641, Jul. 2011, doi: 10.1016/j.econmod.2011.02.019.
- [19] K. Sohag, R. Vasilyeva, A. Urazbaeva, and V. Voytenkov, ‘Stock Market Synchronization: The Role of Geopolitical Risk’, *Journal of Risk and Financial Management*, vol. 15, no. 5, p. 204, Apr. 2022, doi: 10.3390/jrfm15050204.
- [20] L. A. Smales, ‘Geopolitical risk and volatility spillovers in oil and stock markets’, *The Quarterly Review of Economics and Finance*, vol. 80, pp. 358–366, May 2021, doi: 10.1016/j.qref.2021.03.008.
- [21] M. N. Otwani and E. M. Nambuswa, ‘EFFECT OF CORPORATE INCOME TAX ON FINANCIAL PERFORMANCE OF COMPANIES LISTED ON THE

REFERENCES

- NAIROBI SECURITIES EXCHANGE IN KENYA', 2017, [Online]. Available: <http://www.ijssit.com>
- [22] I. Ötöker and C. Pazarbasioglu, 'Impact of Regulatory Reforms on Large and Complex Financial Institutions', *IMF Staff Position Notes*, vol. 2010, no. 16, p. 1, 2010, doi: 10.5089/9781462345496.004.
- [23] I. M. Miahkykh, M. S. Shkoda, and A. O. Radchenko, 'FACTORS ENSURING ENTERPRISE FINANCIAL STABILITY IN TIMES OF CRISIS', *Bulletin of the Kyiv National University of Technologies and Design. Series: Economic sciences*, vol. 139, no. 5, pp. 121–129, Apr. 2020, doi: 10.30857/2413-0117.2019.5.11.
- [24] M. Bronson, 'The impact of trade policy changes in global supply chains in 2024', *ITSupplyChain*.
- [25] Y. Chen, 'Analysis of the Impact of Monetary Policy on the Housing Market', *Advances in Economics, Management and Political Sciences*, vol. 60, no. 1, pp. 236–242, Jan. 2024, doi: 10.54254/2754-1169/60/20231238.
- [26] H. Gursida, 'The Influence of Fundamental and Macroeconomic Analysis on Stock Price', 2017.
- [27] N. Tangjitprom, 'Macroeconomic Factors of Emerging Stock Market: The Evidence from Thailand', *SSRN Electronic Journal*, 2011, doi: 10.2139/ssrn.1957697.
- [28] N. M. H. Masoud, 'The Impact of Stock Market Performance upon Economic Growth', *International Journal of Economics and Financial Issues*, vol. 3, no. 4, pp. 788–798, 2013, [Online]. Available: www.econjournals.com
- [29] M. Ali, 'Inflation, Interest and Exchange Rate Effect of the Stock Market Prices', *JBEO*, vol. 1, no. 2, pp. 38–43, Jun. 2018.
- [30] M. M. Alam, M. Gazi, and S. Uddin, 'Relationship between Interest Rate and Stock Price: Empirical Evidence from Developed and Developing Countries', 2009.
- [31] R. M. Gaspar and X. Jiaming, 'Consumer Confidence and Stock Markets' Returns', 2023. [Online]. Available: <https://rem.rc.iseg.ulisboa.pt/https://twitter.com/ResearchRemhttps://www.linkedin.com/company/researchrem/https://www.facebook.com/researchrem/>

REFERENCES

- [32] R. Andrawos, ‘NLP in Stock Market Prediction: A Review’, 2022, doi: 10.13140/RG.2.2.17142.68160.
- [33] Y. Wang, ‘Analyzing the impact of financial news on the stock market using Natural Language Processing techniques’, *Applied and Computational Engineering*, vol. 76, no. 1, pp. 294–301, Jul. 2024, doi: 10.54254/2755-2721/76/20240622.
- [34] C. Zhang, Y. Wang, C. Chen, C. Du, H. Yin, and H. Wang, ‘StockAssIstant’, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, Jul. 2018, pp. 2710–2719. doi: 10.1145/3219819.3219964.
- [35] E. F. Fama, ‘American Finance Association Efficient Capital Markets: A Review of Theory and Empirical Work’, 1970. doi: <https://doi.org/10.2307/2325486>.
- [36] ‘A RANDOM WALK DOWN WALL STREET’.
- [37] A. W. Lo, ‘The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective *’, 2004.
- [38] A. W. Lo, *Adaptive Markets: Financial Evolution at the Speed of Thought*, 1st ed. Princeton University Press, 2017. Accessed: Sep. 11, 2024. [Online]. Available: <http://assets.press.princeton.edu/chapters/i10932.pdf>
- [39] S. Kapoor and J. M. Prosad, ‘Behavioural Finance: A Review’, *Procedia Comput Sci*, vol. 122, pp. 50–54, 2017, doi: 10.1016/j.procs.2017.11.340.
- [40] H. Kent Baker, G. Filbeck, and V. Ricciardi, ‘How Behavioural Biases Affect Finance Professionals’, *The European Financial Review*, pp. 25–29, Jan. 2017, [Online]. Available: <https://ssrn.com/abstract=2899214www.europeanfinancialreview.com25>
- [41] A. Jauharia, H. Bambang, and S. Dwiyanto, ‘THE COMPANY FUNDAMENTAL FACTORS AND SYSTEMATIC RISK IN INCREASING STOCK PRICE’, 2012.
- [42] J. Schmidt, ‘What is Financial Modeling?’, CFI Education Inc.
- [43] Z. Bodie, A. Kane, and A. J. Marcus, ‘Equity Valuation Models’, in *Investments*, 11th ed., McGraw-Hill Education, 2018, ch. 18, pp. 573–598.
- [44] S. Gu, B. Kelly, and D. Xiu, ‘Empirical Asset Pricing via Machine Learning’, *Rev Financ Stud*, vol. 33, no. 5, pp. 2223–2273, May 2020, doi: 10.1093/rfs/hhaa009.

REFERENCES

- [45] M. Qiu, Y. Song, and F. Akagi, ‘Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market’, *Chaos Solitons Fractals*, vol. 85, pp. 1–7, Apr. 2016, doi: 10.1016/j.chaos.2016.01.004.
- [46] M. Zulqarnain, R. Ghazali, M. G. Ghouse, Y. M. M. Hassim, and I. Javid, ‘Predicting Financial Prices of Stock Market using Recurrent Convolutional Neural Networks’, *International Journal of Intelligent Systems and Applications*, vol. 12, no. 6, pp. 21–32, Dec. 2020, doi: 10.5815/ijisa.2020.06.02.
- [47] I. K. Nti, A. F. Adekoya, and B. A. Weyori, ‘Efficient Stock-Market Prediction Using Ensemble Support Vector Machine’, *Open Computer Science*, vol. 10, no. 1, pp. 153–163, Jul. 2020, doi: 10.1515/comp-2020-0199.
- [48] Md. R. Hassan, ‘A combination of hidden Markov model and fuzzy model for stock market forecasting’, *Neurocomputing*, vol. 72, no. 16–18, pp. 3439–3446, Oct. 2009, doi: 10.1016/j.neucom.2008.09.029.
- [49] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, ‘Stock Market Prediction Based on Generative Adversarial Network’, in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 400–406. doi: 10.1016/j.procs.2019.01.256.
- [50] T. Wan Teo, J. Ying Ng, S.-Y. Liew, Z. Zheng Khoo, and C. Henn Chng, ‘GPT-4 Powered Virtual Analyst for Fundamental Stock Investment by Leveraging Qualitative Data’.
- [51] S. A. Bond *et al.*, ‘Large Language Models and Financial Market Sentiment *’, 2024. [Online]. Available: <https://bard.google.com/>.
- [52] U. Gupta, ‘GPT-InvestAR: Enhancing Stock Investment Strategies through Annual Report Analysis with Large Language Models’, Sep. 2023, [Online]. Available: <http://arxiv.org/abs/2309.03079>
- [53] Y. Yang, Y. Tang, and K. Y. Tam, ‘InvestLM: A Large Language Model for Investment using Financial Domain Instruction Tuning’, Sep. 2023, [Online]. Available: <http://arxiv.org/abs/2309.13064>
- [54] H. Yang, X.-Y. Liu, and C. D. Wang, ‘FinGPT: Open-Source Financial Large Language Models’, Jun. 2023, [Online]. Available: <http://arxiv.org/abs/2306.06031>

REFERENCES

- [55] S. Makridakis, F. Petropoulos, and Y. Kang, ‘Large Language Models: Their Success and Impact’, *Forecasting*, vol. 5, no. 3, pp. 536–549, Aug. 2023, doi: 10.3390/forecast5030030.
- [56] A. Thakkar and K. Chaudhari, ‘Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions’, *Information Fusion*, vol. 65, pp. 95–107, Jan. 2021, doi: 10.1016/j.inffus.2020.08.019.
- [57] L. Zhao, L. Li, and X. Zheng, ‘A BERT based Sentiment Analysis and Key Entity Detection Approach for Online Financial Texts’, 2020. doi: <https://doi.org/10.48550/arXiv.2001.05326>.
- [58] S. Dilshan, ‘Agile Methodology’, Medium.
- [59] M. Zeff, ‘OpenAI unveils GPT-4o mini, a smaller and cheaper AI model’, TechCrunch.

APPENDIX

A.1 Example Full Results Returned from Query

```
{
  "analysis_id": "9ecdd9fa-50ee-4fc4-b062-3ae2cecbabe6",
  "company_enrichment": {
    "companies": [
      {
        "confidence": "high",
        "detected_name": "Supermax Corporation Berhad",
        "financial_data": {
          "52_week_high": 1.12,
          "52_week_low": 0.46,
          "beta": 0.06,
          "company_name": "Supermax Corporation Berhad",
          "current_price": 0.46,
          "dividend_yield": "N/A",
          "industry": "Medical Instruments & Supplies",
          "last_updated": "2025-09-21T16:12:19.101549+00:00",
          "market_cap": 1406336768,
          "market_cap_formatted": "RM1.41B",
          "pe_ratio": "N/A",
          "sector": "Healthcare",
          "ticker": "7106.KL",
          "trading_volume": 5790900
        },
        "official_name": "SUPERMAX CORPORATION BERHAD",
        "ticker": "7106.KL"
      }
    ],
    "companies_detected": true,
    "companies_matched": true,
    "detected_companies": [
      "Supermax Corporation Berhad"
    ],
    "enriched_count": 1,
    "matched_count": 1,
    "processing_time": 1.474
  },
}
```

```

"conversation_context": {
  "actionable_insights": null,
  "financial_data": {
    "52_week_high": 1.125,
    "52_week_low": 0.455,
    "avg_volume": 6864287,
    "beta": 0.058,
    "company_name": "Supermax Corporation Berhad",
    "current_price": 0.46,
    "dividend_yield": "N/A",
    "industry": "Medical Instruments & Supplies",
    "last_updated": "2025-09-21T16:12:19.396505+00:00",
    "market_cap": 1406336768,
    "notes": {
      "dividend_yield": "Dividend yield is unavailable, likely because the company does not pay dividends
or data is not reported."
    },
    "pe_ratio": -9.2,
    "sector": "Healthcare",
    "ticker": "7106.KL",
    "volatility": 1.82,
    "volume": 5790900
  },
  "query": "Please analyze and provide insights on: Impact of the US Tariff imposed to Malaysia
(Company: Supermax Corporation Berhad, Ticker: 7106.KL)",
  "related_questions": [
    "How might Supermax's supply chain adjustments mitigate the impact of these US tariffs on their
profit margins?",
    "Considering the competitive landscape, how could these tariffs affect Supermax's market
positioning against other glove manufacturers?",
    "What strategic initiatives should Supermax consider to adapt to potential long-term tariff changes
and maintain growth?"
  ],
  "risk_explanation": null,
  "risk_score": null,
  "sources": [
    {
      "citation_number": 1,

```


APPENDIX

```
"content": "Plantation likely to see improved profit margins. The price surge also affects downstream industries including food manufacturing and biodiesel production.\n\nThe government has announced measures to su...",
"source": "The Star Malaysia",
"title": "Palm Oil Prices Surge Amid Global Supply Concerns",
"type": "news_article"
},
],
"summary": null,
"ticker": "7106.KL",
"timestamp": "2025-09-21T16:12:38.216714+00:00"
},
"conversation_depth": 0,
"financial_data": {
  "52_week_high": 1.125,
  "52_week_low": 0.455,
  "avg_volume": 6864287,
  "beta": 0.058,
  "company_name": "Supermax Corporation Berhad",
  "current_price": 0.46,
  "dividend_yield": "N/A",
  "industry": "Medical Instruments & Supplies",
  "last_updated": "2025-09-21T16:12:19.396505+00:00",
  "market_cap": 1406336768,
  "notes": {
    "dividend_yield": "Dividend yield is unavailable, likely because the company does not pay dividends or data is not reported."
  },
  "pe_ratio": -9.2,
  "sector": "Healthcare",
  "ticker": "7106.KL",
  "volatility": 1.82,
  "volume": 5790900
},
"followup_type": "conversation_continuation",
"formatted_response": {
  "formatted": true,
  "raw_text": "### Impact of US Tariffs on Supermax Corporation Berhad (7106.KL)\n\nThe recent imposition of tariffs by the United States on Malaysian goods has significant implications for Supermax Corporation Berhad (Ticker: 7106.KL), a key player in the medical instruments and supplies sector,
```

particularly known for its production of gloves. This analysis will explore the potential impacts of these tariffs on Supermax's financial performance and market position, building on our previous discussions regarding the company's fundamentals and market dynamics.

Overview of Supermax Corporation

As of the latest data, Supermax is trading at RM 0.46, with a market capitalization of RM 1.41 billion. The company has experienced a significant decline from its 52-week high of RM 1.12, reflecting broader market challenges and specific pressures from tariff-related issues. The company's P/E ratio is currently not applicable, indicating potential losses or a lack of earnings visibility, which is critical for investors to consider.

Analysis of Tariff Implications

Cost Structure and Profit Margins

- The tariffs imposed by the US could increase the cost of raw materials and components that Supermax imports, thereby squeezing profit margins. Given that the healthcare sector is highly competitive, passing these costs onto consumers may not be feasible without risking market share.
- Supermax's current low beta of 0.06 suggests that the stock is less volatile compared to the market, which could provide some stability in turbulent times. However, if tariffs lead to increased operational costs, this could change investor sentiment and affect stock performance.

Market Demand and Sales

- The US is a significant market for medical supplies, including gloves. Tariffs could lead to reduced demand from US buyers who may seek alternatives from countries not subject to such tariffs. This shift could adversely affect Supermax's sales volume and revenue growth.
- As we discussed earlier, the company's reliance on exports makes it vulnerable to geopolitical and trade tensions. A decline in US sales could necessitate a strategic pivot towards other markets, which may not compensate for the lost revenue.

Long-Term Strategic Adjustments

- In response to tariffs, Supermax may need to consider diversifying its supply chain or increasing local production to mitigate tariff impacts. This could involve investing in domestic manufacturing capabilities or sourcing materials from countries with favorable trade agreements with the US.
- Additionally, the company could explore enhancing its product offerings or focusing on innovation to differentiate itself in a competitive market, potentially offsetting some of the demand loss from tariff impacts.

Opportunities and Risks

Opportunities

- If Supermax can successfully navigate the tariff landscape, there may be opportunities to capture market share from competitors who are less adaptable.
- The ongoing global health concerns could sustain demand for medical supplies, providing a buffer against tariff impacts if managed effectively.

Risks

- Continued tariff pressures could lead to a prolonged decline in profitability, affecting cash flows and the ability to reinvest in the business.
- The potential for retaliatory measures from Malaysia could further complicate the operational landscape for Supermax.

Conclusion and Next Steps

In conclusion, the US tariffs on Malaysian goods present both challenges and opportunities for Supermax Corporation Berhad. The immediate impact on costs and market demand could strain the company's financial performance, particularly given its current trading metrics. Investors should closely monitor the company's strategic responses to these tariffs, including any shifts in supply chain management or market focus.

Actionable Insights

- Investors should consider the potential for volatility in Supermax's stock price as the company navigates these challenges.
- Monitoring quarterly earnings reports will be crucial to assess how effectively Supermax is managing tariff impacts and whether it can maintain its competitive edge in the market.
- Diversification of investments may be prudent, given the uncertainties surrounding tariff policies and

their implications for Supermax's future performance. \n\nAs we continue to analyze Supermax's performance, it will be important to keep an eye on both macroeconomic factors and company-specific developments that could influence its trajectory in the coming quarters."

```

},
"is_first_message": false,
"is_followup_response": true,
"learning_indicators": {
  "feedback_based_analysis": true,
  "improvement_applied": true,
  "total_feedback_samples": 0
},
"source_hint": "This analysis is based on available information sources.",
"sources": [
  {
    "citation_number": 1,
    "content": "Plantation likely to see improved profit margins. The price surge also affects downstream
industries including food manufacturing and biodiesel production.\n\nThe government has announced
measures to su...",
    "source": "The Star Malaysia",
    "title": "Palm Oil Prices Surge Amid Global Supply Concerns",
    "type": "news_article"
  }
],
"status": "success",
"ticker": "7106.KL",
"timestamp": "2025-09-21T16:12:38.324815+00:00",
"token_usage": {
  "completion_tokens": 1148,
  "input_cost": 0.000138,
  "model_name": "gpt-4o-mini",
  "output_cost": 0.000689,
  "prompt_tokens": 918,
  "total_cost": 0.000826,
  "total_tokens": 2066
}
}

```


A.2 Poster

