

**THE UNIVERSITY OF HONG KONG**  
**DEPARTMENT OF COMPUTER SCIENCE 2024 - 25**



**COMP4801/FITE4801**  
**Final Year Project**  
Individual Final Report

**Group 24079**  
Lee Jong Seung (3035555547)

**Topic**  
TradeInbox: LLM-based Real-time Personalized Financial News Notification System

## **Abstract**

The project TradeInbox examines the challenges faced by retail investors in today's volatile financial market, where the overwhelming volume of news hinders timely and relevant updates on stock portfolios. This project addresses these issues by processing and filtering news for the US public equities, enhancing it with a keyword generation and utilizing the keywords to an embedding search to filter related news articles. News summary and analysis modules process the news articles to deliver real-time insights, empowering investors to make informed decisions.

## **Acknowledgement**

The project team expresses profound gratitude to Dr. Chow Ka Ho, final-year project advisor, for his exceptional guidance and unwavering support throughout the project. His insightful feedback and mentorship were pivotal in refining the research and enriching the team's learning experience. Deep appreciation is also extended to the Second Examiner, Dr. Hubert Chan, for their rigorous evaluation and valuable insights. The dedicated efforts, resilience, and collaboration of the project team members were also instrumental in achieving the project's objectives. Additionally, the team acknowledges the data vendors such as LSEG Refinitiv and NewsCatcher, which generously provided complimentary access to their APIs, enabling the successful execution of the project. Lastly, sincere thanks are due to the financial and investment analysts whose expert perspectives and constructive feedback greatly enhanced the study from the financial service industry perspectives.

# Table of Contents

<b>1. Project Introduction</b>	<b>1</b>
<b>2. Project Objectives</b>	<b>3</b>
<b>3. Project Contribution</b>	<b>4</b>
<b>4. Project Background and Challenges</b>	<b>5</b>
<b>5. Project Methodology</b>	<b>7</b>
5.1 Data Source and APIs	8
5.2 System Design Architecture	9
5.3. Engineering Choices	11
5.4. Core Functional Modules	12
5.4.1. Authentication	13
5.4.2. Real-time News Data Polling Agent	13
5.4.3. News Data Concurrency Handling	14
5.4.4. Embedding Search of Relevant News Articles	14
5.4.5. Keyword Generators for Indirect Matching	16
5.4.6. News Summary	17
5.4.7. Key Metrics	18
5.4.8. Sentiment Analysis	19
5.4.9. News Impact Analysis	21
5.4.10. Keyword Feedback for Indirect Matching	25
5.4.11. Delivery to External Applications	25
<b>6. Experiments and Results</b>	<b>26</b>

6.1. Experiments	26
6.1.1. UI/UX	26
6.1.2. Keyword Generation for Embedding Search	28
6.1.3. Calibration of Similarity Metrics	29
6.2. Results	30
6.2.1. Authentication Page	31
6.2.2. Tickers Selection Page	32
6.2.3. Dashboard Page	32
6.2.4. News Analysis Page	33
6.2.5. Delivery to External Applications	34
<b>7. Conclusion and Future Works</b>	<b>35</b>
7.1. Conclusion	35
7.2. Future Works	36
7.2.1. Technology	36
7.2.2. Usability	37
7.2.3. Testing	37
<b>Citation</b>	<b>39</b>

## List of Figures

<b>Figure 1.</b> Yahoo Finance Webpage for NVDA	<b>5</b>
<b>Figure 2.</b> Bloomberg Terminal News Page	<b>6</b>
<b>Figure 3.</b> Main Data Vendors (LSEG Refinitiv, NewsCatcher)	<b>8</b>
<b>Figure 4.</b> Data Load Function	<b>8</b>
<b>Figure 5.</b> System Architecture Design	<b>10</b>
<b>Figure 6.</b> Engineering Choices	<b>11</b>
<b>Figure 7.</b> News Polling Agent	<b>13</b>
<b>Figure 8.</b> Extraction of Embeddings from News Data	<b>15</b>
<b>Figure 9.</b> Definition of Cosine Similarity	<b>15</b>
<b>Figure 10.</b> News Summary Prompt	<b>17</b>
<b>Figure 11.</b> Key Metrics Prompt	<b>18</b>
<b>Figure 12.</b> Sentiment Analysis Prompt	<b>20</b>
<b>Figure 13.</b> News Impact Analysis Prompt	<b>23</b>
<b>Figure 14.</b> Simple Yes or No User Feedback System	<b>25</b>
<b>Figure 15.</b> Initially Developed Dashboard UI	<b>26</b>
<b>Figure 16.</b> Revised Dashboard UI	<b>27</b>
<b>Figure 17.</b> Keyword Generation Examples	<b>28</b>
<b>Figure 18.</b> Authentication Page	<b>31</b>
<b>Figure 19.</b> Ticker Selection Page	<b>32</b>
<b>Figure 20.</b> News Analysis Page	<b>33</b>
<b>Figure 21.</b> Discord Message Notification	<b>34</b>

## List of Tables

<b>Table 1.</b> Contributions of Team Members	<b>4</b>
<b>Table 2.</b> Examples of Keyword Matching Results	<b>29</b>

# 1. Project Introduction

The financial market exhibits significant volatility, with news serving as a primary catalyst for price movements. A pertinent illustration of this phenomenon occurred on January 8, 2025, when Nvidia CEO Jensen Huang commented at CES that quantum computing remains 15–30 years from commercialization, contradicting prevailing market optimism [1]. This statement precipitated a bearish market response, resulting in substantial single-day declines in quantum computing-related stocks, including Rigetti Computing (RGTI, -45%), IonQ (IONQ, -39%), D-Wave Quantum (QBTS, -36%), and Quantum Computing (QUBT, -43%).

Another instance of news-driven market volatility arose from macroeconomic developments. On January 10, 2025, the announcement of stronger-than-expected U.S. job growth for December, accompanied by a decline in the University of Michigan consumer sentiment index, adversely affected market sentiment [2]. This led to significant declines across major U.S. indices, with the Dow Jones Industrial Average falling 1.63%, the S&P 500 dropping 1.54%, and the Nasdaq Composite decreasing 1.63%. However, many retail investors face challenges in interpreting such news, often due to a limited understanding of the theoretical linkages between macroeconomic data releases and their impact on asset price movements.

Political shifts are also significant influencers of global markets, with policy decisions often triggering substantial volatility. The reciprocal tariffs imposed by U.S. President Donald Trump as of April 2, 2025, serve as a notable example of this dynamic. A recent instance involved Trump's reciprocal tariffs, which precipitated the largest two-day decline in major U.S. benchmarks since March 2020 [3], underscoring the profound impact of political



developments on financial market stability. In this context, there is a need for a smart news aggregator to swiftly interpret and respond to such market-moving political events.

Academic findings also support that news is a significant source of market volatility. A study by Atkins, Niranjana, and Gerding demonstrates that financial news outperforms other figures such as historical closing prices in predicting stock market volatility [4]. The research highlights that news events, particularly those with substantial informational content, prompt swift investor reactions, leading to amplified price fluctuations. This effect is driven by the rapid adjustment of market expectations following unexpected news, such as earnings reports or macroeconomic updates.

Despite the pivotal role of financial news analysis in investment decision-making, retail investors frequently lack the time to effectively process huge daily inflows of news. A web traffic study by Fintext [5] indicates that financial news readers typically engage with only three to four pages daily, dedicating approximately 30 seconds to one minute per page. This limited engagement, totalling just two to four minutes per day, may lead to an incomplete understanding of market dynamics, potentially resulting in suboptimal investment decisions and financial losses.

Although existing financial news platforms such as Yahoo Finance provide some level of news curation, these services often rely on rudimentary keyword-based filtering and lack the nuanced contextual analysis necessary to align news content with an individual's previous experience level in investment processes. Recent advancements in large language models (LLM) offer transformative potential for delivering real-time, personalized news. LLM, with their ability to process and interpret complex languages, can be prompt-engineered using clear instructions to identify news relevant to a user's portfolio.

By integrating LLM into news delivery systems, retail investors can stay up-to-date with market trends, make timely and informed decisions. This technological innovation presents a significant opportunity to bridge the information disparity between retail investors and financial institutions, thereby facilitating more effective and informed investment strategies from the retail side.

## 2. Project Objectives

To address the challenges faced by retail investors, this final-year project ultimately aims to narrow the information asymmetry in the investment process and educate users on fundamental investment knowledge. Leveraging prompt-engineered LLM and aggregating reliable data sources such as LSEG Refinitiv and Newscatcher API, the project seeks to deliver actionable insights and improve the decision-making process for retail investors.

To achieve the primary objective, the team proposes a personalized real-time financial news notification application that leverages LLM and Natural Language Processing (NLP) to deliver highly relevant and time-sensitive information to investors. This application aims to enable retail investors to make informed decisions quickly and effectively by delivering summarized news updates and insights. The proposed solution TradeInbox is expected to:

- 1. Resolve information asymmetry:** The TradeInbox system delivers real-time, personalized news alerts tailored to an individual's portfolio. It fetches live news data and sends notifications through third-party platforms such as Discord, ensuring retail investors receive timely and comprehensive access to market events impacting their investments.
- 2. Alleviate financial market literacy:** The project aims to provide detailed summaries and in-depth analyses of news articles, highlighting key points and potential impacts

on stock prices, while offering explanations of financial concepts and terminology through the LLM, empowering retail investors to better understand market dynamics and make informed decisions.

Given the constraints of limited time for analyzing multilingual data and the inability to access undisclosed information related to private investments, the project scope is confined to addressing publicly listed equities in the United States. The team will focus exclusively on processing financial news available in English.

### 3. Project Contribution

This section details the individual team members' roles and contributions, highlighting how the group strategically allocated tasks according to each member's specialized expertise. The detailed breakdown of individual responsibilities and tasks is provided below:

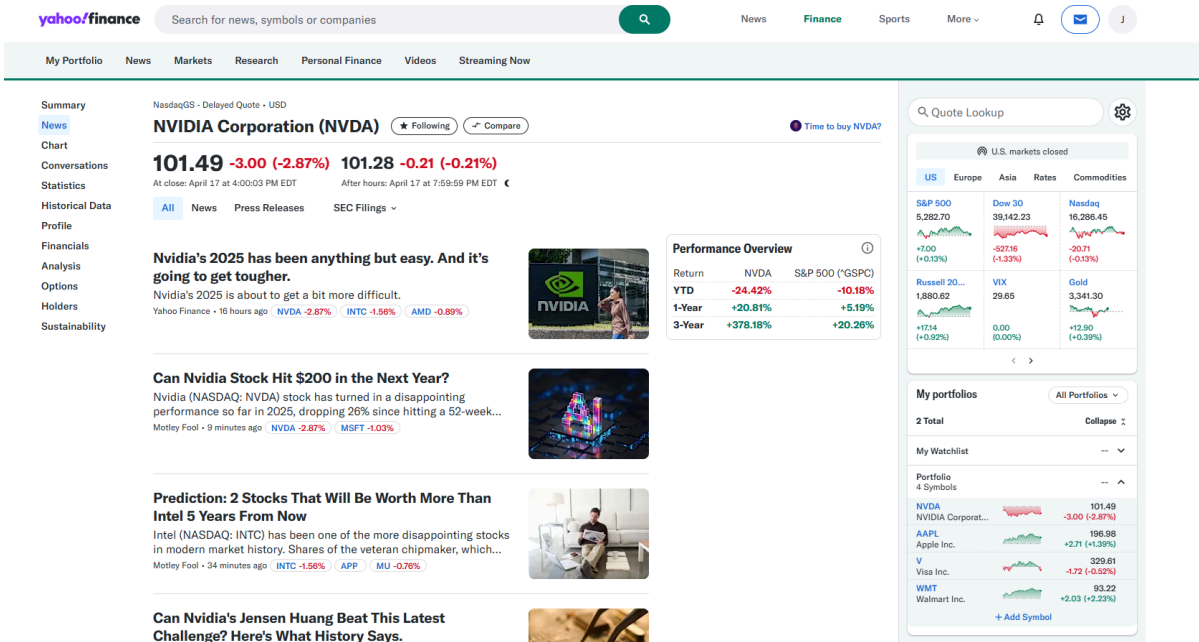
Group Member	Main Role	Detailed Contribution
Changjin Lee	Backend Developer	<ul style="list-style-type: none"> <li>- Built the JWT user authentication mechanism</li> <li>- Built and maintain a vector DB with embedding search for semantic queries</li> <li>- Integrated all LLM modules into backend systems</li> </ul>
Jong Seung Lee	Product Manager, Data Analyst	<ul style="list-style-type: none"> <li>- Aligned all workflows with financial domain expertise</li> <li>- Managed data vendor relations and created cleaned datasets</li> <li>- Co-designed optimized UI/UX workflows</li> </ul>
Taehyun Kim	Frontend Developer	<ul style="list-style-type: none"> <li>- Primarily designed UI/UX</li> </ul>

		<ul style="list-style-type: none"> <li>- Developed a React-based frontend application</li> <li>- Assisted with frontend and backend integration</li> </ul>
--	--	--

**Table 1.** *Contributions of Team Members*

## 4. Project Background and Challenges

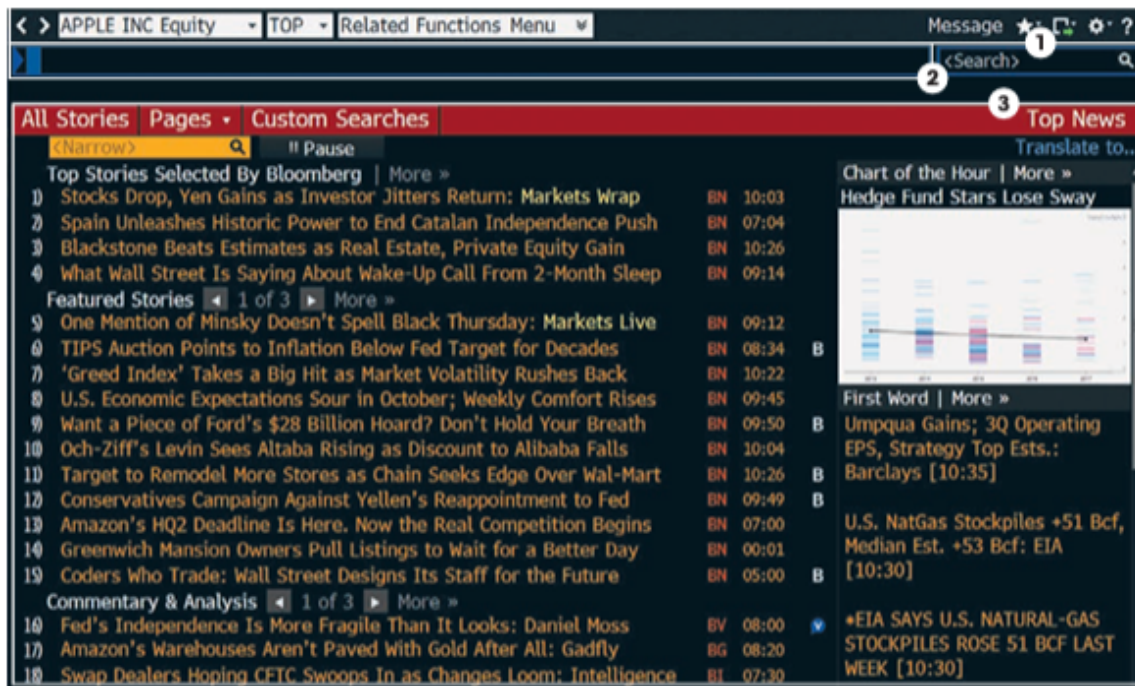
This section examines several widely used financial news aggregators, explores the additional functionalities offered by TradeInbox, and discusses some potential challenges in implementing these features. As of now, Yahoo Finance and Bloomberg Terminal are among the most common sources of financial news for retail and institutional investors, respectively.



**Figure 1.** *Yahoo Finance Webpage for NVDA*

Yahoo Finance provides retail investors with an accessible platform for financial data and news. However, its stock-ticker-specific news filtering relies on direct keyword matching, aggregating only articles that explicitly mention the company name or ticker

symbol. This approach fails to capture the broader implications of relevant news, as it overlooks indirect yet material developments, such as macroeconomic trends, regulatory changes, or sector-wide disruptions, that may significantly impact equity prices.



**Figure 2.** *Bloomberg Terminal News Page [6]*

The Bloomberg Terminal provides institutional investors with comprehensive access to real-time financial data and news. Despite its extensive capabilities, the platform presents three significant barriers to retail investors: its prohibitively expensive subscription model, the steep learning curve required to master its advanced analytical functions and tools, and a complex user interface lacking intuitive navigation for beginners. These compounding factors create substantial accessibility challenges that exclude non-professional market participants from utilizing the platform.

TradeInbox distinguishes itself from conventional financial news platforms through several innovative features. At its core, the system employs a Keyword Generator LLM coupled with an Embedding Search methodology to capture any indirect nuances. Beyond

basic retrieval, the platform dynamically tailors news analysis content according to individual user profiles, mainly incorporating investment proficiency to deliver maximally relevant insights. Collectively, these capabilities position TradeInbox as a powerful decision-support tool that democratizes access to sophisticated market intelligence for retail investors.

The development of such an advanced system presents some challenges posed by the complexity of the system design. First, the similarity search architecture must transcend conventional keyword matching to capture both explicit and implicit news impacts on securities, a task requiring sophisticated relevancy scoring algorithms capable of interpreting nuanced financial contexts. Second, the system's multi-model integration poses significant engineering complexity, demanding flawless interoperability between real-time news polling agents, heterogeneous LLM analytical pipelines, and high-dimensional embedding search infrastructure, while maintaining latency and reliability standards. Finally, the platform's financial education domain could necessitate extensive fine-tuning or prompt engineering of the existing LLM. This optimization process might require iterative validation paired with expert qualitative assessment to ensure delivery of contextually appropriate results.

## **5. Project Methodology**

This section presents the technical framework and implementation methodologies employed in the project. Specifically, it discusses the data acquisition pipeline, the comprehensive system architecture, engineering choices and the core functional components enabling personalised, real-time financial news summarisation and analysis.

## 5.1 Data Source and APIs



**Figure 3.** *Main Data Vendors (LSEG Refinitiv, NewsCatcher)*

The success of this financial news aggregation project fundamentally depends on accessing high-quality, real-time news data. To achieve this, the system integrates two complementary API services. The NewsCatcher API [7] serves as the primary news data source, providing critical functionality including real-time news updates with various filtering parameters that can be utilized for testing and customization. The API delivers data in JSON format, containing comprehensive metadata fields such as article title, author, published date, source link, cleaned URL, excerpt, full text content, website authority rank, topic classification, country of origin, and language. This full-spectrum news data facilitates robust analysis by providing both the raw content and contextual metadata necessary for accurate processing.

```
def news_load():
    datetime_now = datetime.now(pytz.timezone('UTC')) + timedelta(minutes=-10)
    datetime_str = datetime_now.strftime("%Y/%m/%d %H:%M:%S")

    all_articles = newscatcherapi.get_search(
        q = '*',
        lang = 'en',
        countries = 'US',
        from_ = datetime_str,
        published_date_precision = 'full',
        ranked_only = True,
        to_rank = 500,
        sort_by = 'date'
    )

    return all_articles['articles']
```

**Figure 4.** *Data Load Function*

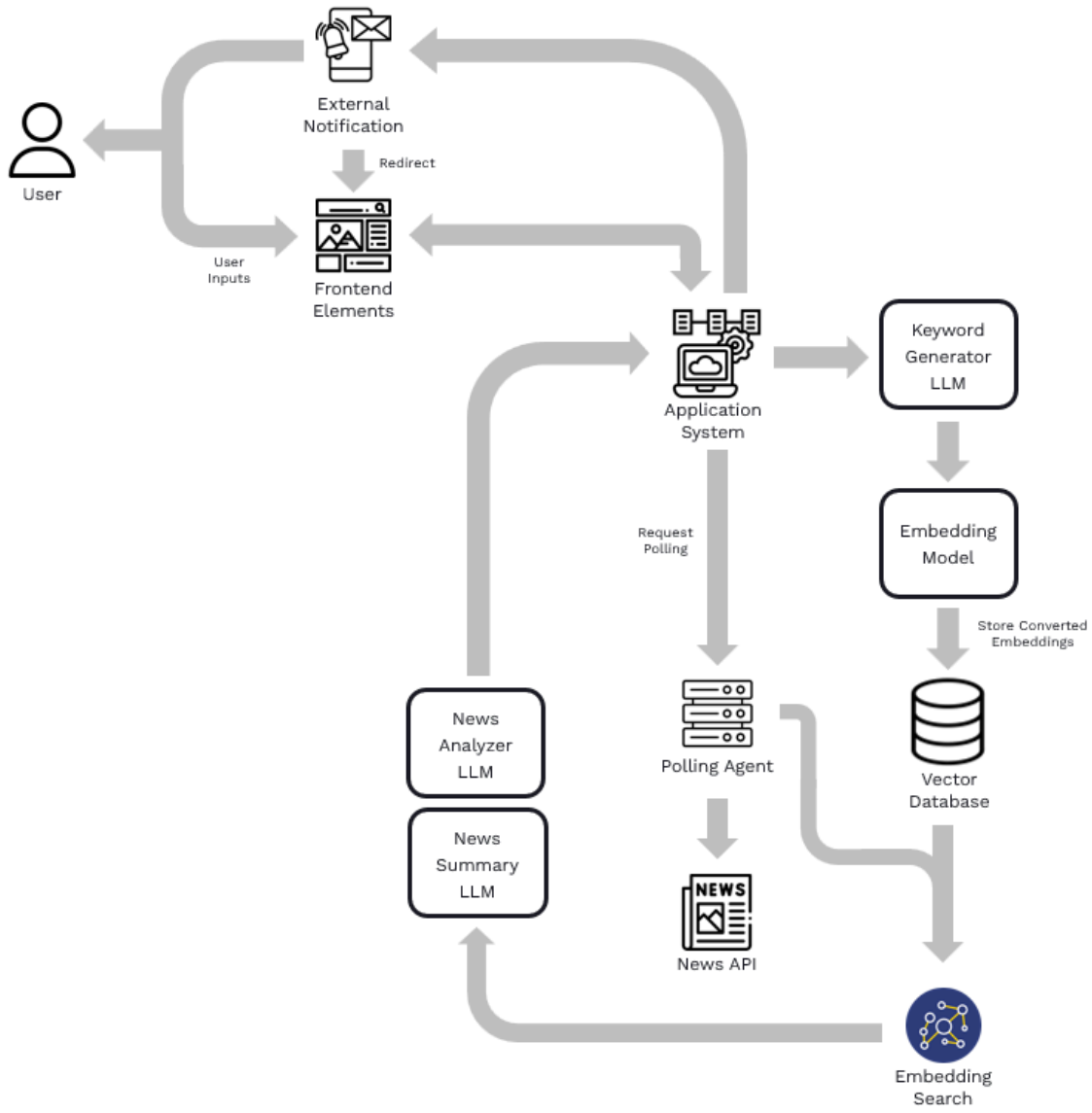
The news retrieval function above collects all articles published in the last ten minutes. To maximize data authenticity and processing efficiency, some pre-filtering with API functions has been applied. First, it restricts content to English-language articles published exclusively within the United States market. Second, it enforces strict temporal metadata requirements, accepting only entries with complete timestamp precision to ensure the timeliness of the acquired data. Finally, it incorporates a proxy for a credibility measure by drawing exclusively from the top 500 news websites ranked by traffic. These selective parameters ensure source validation as well as the elimination of unnecessary processing of incompatible or geographically irrelevant content.

While NewsCatcher API provides the essential news article data required to fulfill the project's primary objectives, the team also integrated the LSEG Refinitiv API to enhance user convenience and functionality. This secondary source supplies live market information, including stock prices and trading volumes, serving as valuable reference data that can be presented visually to support users' investment decisions.

## **5.2 System Design Architecture**

This project incorporates various modules to be integrated, thus requiring careful process design to minimize any latency.





**Figure 5.** *System Architecture Design*

The proposed system architecture mainly employs embedding generation techniques to process user-specified financial preferences, which are subsequently stored in a vector database for semantic retrieval. The architecture starts with the users providing a list of stocks to be tracked. Then the stock lists are fed to the keyword generation module, which translates individual stock input data into semantically related keyword embeddings to be stored in a vector database.

While the collection of modules is in charge of handling any user input, a dedicated polling agent queries financial news sources at ten-minute intervals to retrieve newly published articles. The articles undergo vectorization through embedding models, which enables semantic comparison. The embedding search module performs similarity assessments between article embeddings and stock keywords embeddings to identify relevance matches.

Upon identifying relevant articles, the system processes them through prompt-engineered LLMs for summary and analysis. These analytical results are immediately distributed through two channels. First, they are updated into the system's frontend dashboard news feeds. Also, they are pushed as automated alerts to external messaging platforms such as Discord. This dual-channel delivery ensures users receive timely updates regardless of their preferred access method.

The architecture prioritizes latency minimization through asynchronous data processing, ensuring near-real-time availability of analytical insights. This temporal efficiency enables users to review dynamically updated market analyses, thereby facilitating informed and timely trading decisions.

### 5.3. Engineering Choices



**Figure 6.** *Engineering Choices*

To maximize the efficiency of the design, the frontend is developed with React, selected for its component-based architecture, which promotes reusability and maintainability

when building dynamic interfaces. React’s robust ecosystem accelerated the development processes, enabling seamless integration with the backend and real-time data rendering.

The backend of the project is built using FastAPI, a modern Python framework known for its high performance, scalability, and asynchronous capabilities. FastAPI leverages ASGI (Asynchronous Server Gateway Interface), operating on a single-threaded model, which optimizes resource usage and enhances performance. This is particularly crucial for TradeInbox’s heavy reliance on external API calls such as OpenAI for LLM processing and Zilliz Cloud for vector database operations.

For vector database operations, Milvus was chosen for its industry-leading performance, delivering 2,406 queries per second (QPS) with a remarkably low computation latency of 1 ms, which outperforms the competitors [8]. This ensures efficient handling of high-dimensional data for semantic search and retrieval-augmented generation (RAG) tasks.

As a result, Python was selected as the primary language of development due to its seamless compatibility with key libraries and SDKs, including OpenAI, LangChain, and Milvus, all of which offer native Python support.

#### **5.4. Core Functional Modules**

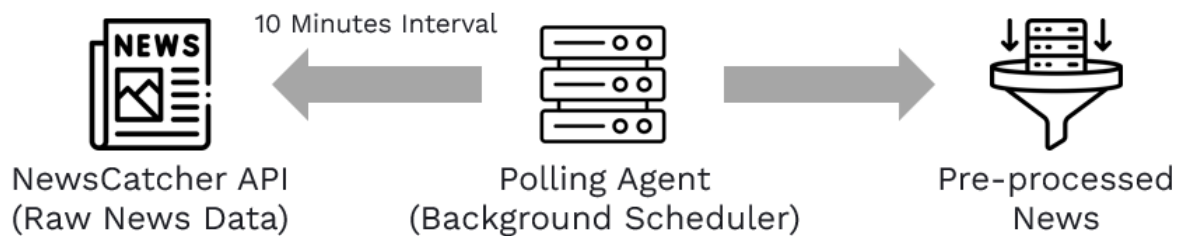
The implementation of TradeInbox focuses on integrating its various functionalities into a cohesive system, with each component carefully selected to ensure seamless interoperability. This section elaborates on how core functionalities are implemented in terms of technical perspectives.

### 5.4.1. Authentication

To ensure robust authentication, the TradeInbox system implements JSON Web Token (JWT) as its primary authentication mechanism. When a user logs in with their credentials, the server generates a JWT, comprising a header, payload, and signature and signs it with a secret key. This token is then included in the Authorization HTTP header for subsequent requests, allowing the server to verify the user's identity without maintaining a session.

The team selected JWT for its stateless design, which is ideal for distributed and scalable environments. JWT only requires a shared secret key for validation. JWT is also self-contained, so embedding essential user details like their ID directly within the token. This reduces redundant database queries, improving both performance and scalability.

### 5.4.2. Real-time News Data Polling Agent



**Figure 7.** *News Polling Agent*

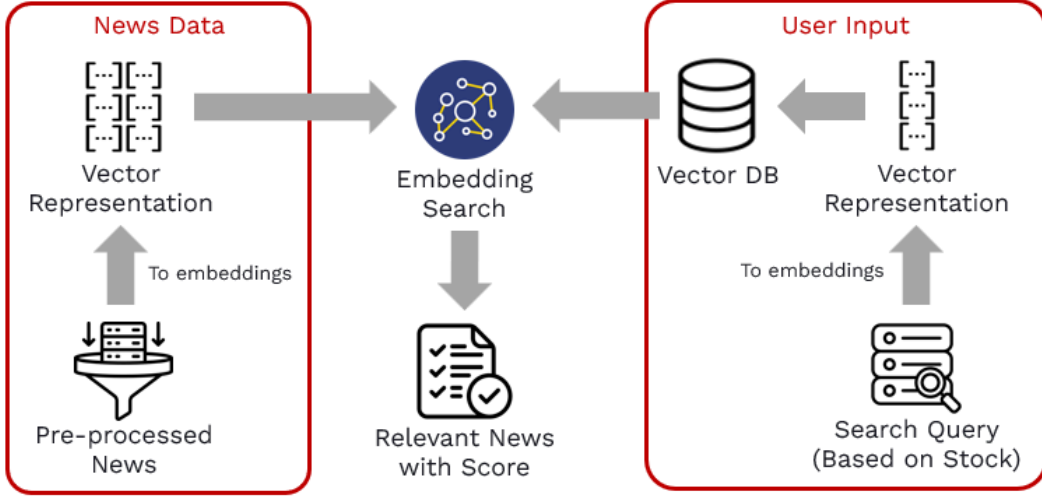
The data load function described in Section 5.1 will be scheduled to run at regular intervals, operating asynchronously to ensure efficiency. A dedicated polling agent is deployed on the backend server, continuously fetches fresh news articles from the API every 10 minutes. This agent is implemented using the Background Scheduler module from APScheduler, which enables recurring tasks to run in the background. Upon receiving a news article, the agent initiates later stages of the pipeline.

### **5.4.3. News Data Concurrency Handling**

However, the consequent modules of the agents involve the tasks that rely heavily on external LLM capabilities such as OpenAI. This introduces significant latency, which might fall into a range of 5 to 10 seconds per call. This is much slower compared to traditional APIs with 50 to 500ms latency. If all modules are executed serially, each news article could take over 30 seconds to process, making the system unscalable for real-time demands. To address this, the system leverages `asyncio` and `aiohttp` for concurrent processing of I/O-bound operations. These Python modules can schedule coroutines as non-blocking asynchronous tasks, allowing multiple operations, such as parallel API calls, to run simultaneously. This approach is further optimized by FastAPI's single-threaded ASGI model, which handles concurrent I/O operations efficiently with minimal resource overhead.

### **5.4.4. Embedding Search of Relevant News Articles**

Embedding constitutes vector-based numerical representations of textual units that encode semantic relationships within linguistic data [9]. Due to their capacity to interpret contextual nuances, these representations are commonly employed in semantic search frameworks to compute similarity metrics between textual elements at varying granularities, from phrases to full documents.



**Figure 8.** *Extraction of Embeddings from News Data*

The figure above elaborates on the overall architecture for the embedding search module. This requires two sets of input, which are the embedding vectors from user-provided stock names stored in the Milvus vector database and the news article embeddings continuously collected and converted by the polling agent.. The implemented system conducts an embedding search with cosine similarity to quantify the relevancy between the stocks and the news articles. Cosine similarity is calculated with the formula below:

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}},$$

**Figure 9.** *Definition of Cosine Similarity*

Articles are only considered matches and stored in the database if their similarity score exceeds the predetermined threshold of 0.4, which was determined through experimental validation. News articles completely unrelated to the given stocks consistently demonstrate negligible similarity scores. For instance, the article 'Spurs take on the Grizzlies

on 3-game losing streak' yields a score of 0.016 when compared against NVIDIA Corporation's embedding. The system processes entire news article bodies as single embeddings, which may lead to semantic dilution due to text length, resulting in generally lower similarity scores. In contrast, clearly relevant articles such as 'Nasdaq Sell-Off: After Losing Nearly \$800 Billion in Market Cap, Is Nvidia Stock a Buy Anymore?' consistently produce similarity scores above the 0.4 threshold.

#### **5.4.5. Keyword Generators for Indirect Matching**

The approach of directly converting stock names like 'Nvidia' into vector embeddings works effectively when news articles explicitly mention the name. However, this method proves insufficient for articles that are indirectly related to the stock without explicitly naming it. To address this limitation, the system employs a Keyword Generator LLM that expands the user's initial input into a more comprehensive set of related terms. When a user specifies an interest in a stock like Coca-Cola, the LLM generates additional contextual phrases such as 'beverage industry trends' or 'impact of sugar taxes on traditional beverage demand'. This expansion enables more thorough semantic analysis during the embedding search process.

The implementation involves the generation of 10 supplementary keywords from each user-provided stock name. These keywords are converted into embeddings and stored in the database. During news processing, the system compares each article's embedding against all stored vectors, selecting the highest cosine similarity score to determine relevance. This multi-keyword approach successfully captures both explicitly mentioned stocks (where the original stock name embedding matches) and indirectly related content (where generated keyword embeddings match).

For example, using 'Nvidia' as input, the system generated keywords including 'Graphics Processing Units', 'AI Technology Trends', and 'Data Center Demand'. While the direct comparison between 'Nvidia' and the 'article AI datacenters want to go nuclear. Too bad they needed it yesterday' scored less than the threshold of 0.4. However, the keyword 'Data Center Demand' achieved a 0.494, successfully identifying the article as relevant. In addition, the explicitly mentioned article 'Nasdaq Sell-Off: After Losing Nearly \$800 Billion in Market Cap, Is Nvidia Stock a Buy Anymore?' matched directly with the 'Nvidia' embedding at 0.52. These outcomes confirm the system's capability to identify both direct and indirect stock-related news through comprehensive keyword expansion.

#### 5.4.6. News Summary

```
[%system%]

You're a professional journalist and an expert in summarizing a news article
into a concise and high-quality summary.

Your Task: Generate a 3-4 sentences summary from the given news article
content from any topic. In addition, generate a short ONE-SENTENCE summary.

[IMPORTANT]

Note that the generated keywords and examples MUST NOT BE too general. IT
MUST BE RELATED TO the provided user input.

{
  "summary": "xxx",
  "one_sentence_summary": "yy"
}

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

News Article Content: {{$article_content}}

Your JSON Output:
```



**Figure 10.** *News Summary Prompt*

The summary prompt defines a structured summarization framework for news articles, aiming for two output formats tailored to different use cases within the platform: a three-to-four-sentence summary for the analysis page and a single-sentence summary for the dashboard, both encoded in JSON. This framework strictly emphasizes alignment of summaries with the article's content to avoid generic or irrelevant terms, ensuring enhanced accuracy in summary delivery.

#### 5.4.7. Key Metrics

```
[%system%]

You're a professional financial data extraction assistant and an expert in
analyzing news articles and generating key financial metrics in the news
article. Focus on numerical values related to the company's financial
performance, stock movement, and market reactions.

Your Task: Extract key financial metrics from the provided financial news
article. Focus on numerical values and financial indicators relevant to the
company or market mentioned.

[IMPORTANT]

Note that the key metrics MUST NOT BE generated on your own. It must be
consolidated from the user input. Restrict your each response into one short
concise sentence. Return 1 to 5 key metrics, ensuring each is short and
concise while covering all critical financial indicators. If there's no key
metrics, return empty list in JSON format given.

// Few-shot learning examples hidden due to the length of the text.

Extraction Criteria:

1. Company Performance: Revenue, Net Income, Earnings Per Share (EPS),
Year-over-Year (YoY) or Quarter-over-Quarter (QoQ) changes, Operating Profit,
Gross Margin, etc.
```

```

2. Stock Market Metrics: Stock price changes (e.g., % increase or decrease),
pre-market or after-hours movement, analyst target price updates, trading
volume, market capitalization, etc.

3. Financial Ratios: Price-to-Earnings (P/E) ratio, Debt-to-Equity (D/E)
ratio, Dividend Yield, Free Cash Flow, etc.

4. Macroeconomic Indicators (if applicable): Interest rate impact, inflation
rate, GDP growth, unemployment rate.

[%user%]

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

Return the key metrics as concise, well-formatted short sentences.


Article: {{$content}}

Your JSON Output:

```

**Figure 11.** *Key Metrics Prompt*

The key metrics prompt conducts an extraction of any important quantitative measures from news articles to assist decision-making based on figures. This function prioritizes numerical indicators that are known to be related to the movement of stock price, such as corporate performance (e.g.) revenue growth, earnings per share), stock dynamics (e.g.) pre-market and after-market movements) and macroeconomic factors (e.g.) interest rates, unemployment rates). Outputs are formatted into zero to five most important figures encoded in JSON format.

#### **5.4.8. Sentiment Analysis**

```

[%system%]

You're a professional financial expert who specializes in sentiment analysis
of a financial news article related to a provided stock name.


The sentiment analysis includes the following items.

```

1) One of [Strong Negative, Negative, Neutral, Positive, Strong Positive].

- where strong negative means the given news article or the contents of the news article are high indicators of the given stock's price will fall down in the future.

- where strong positive means the given news article or the contents of the news article are high indicators of the given stock's price will go up in a future

- Neutral means that it's not reasonable or there's no significant indicator of the stock price from the news article.

2) Insightful, factual, reasonable, logical, detailed analysis & rationale & proofs & evidence of your claim from item (1).

Do not merely give meaningless, verbose, no-depth, abstract reasons. Give DEFINITIVE PROOFS OR EVIDENCE OR RATIONALE to your claim in a structured manner.

3) Also, provider a "SENTIMENT SCORE" ranging from [-5 ~ 5] (inclusive) to indicate how positive/negative the given news is.

- 5 means Strong positive, and -5 means Strong Negative, and 0 means Neutral.

- Give it as a floating-point number.

Make sure that your sentiment analysis output is CONCISE AND INSIGHTFUL. It should not be verbose and long. Must be around 3~4 sentences depending on the situation.

In addition, you will be provided a specific stock name that you must analyze the impact with.

Your sentiment analysis of the news article must be done in consideration of this given stock name.

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

You must return response in the following "JSON" format (only JSON)

```
{  
  
  "sentiment": "xxx", // item (1) - either Strong_Negative, Negative,  
Neutral, Positive, Strong_Positive  
  
  "sentiment_score": xx,  
  
  "analysis": "..." // item (2)
```

```
}

Given stock name: {{$stock_name}}

News Title: {{$title}}

News Content: {{$content}}

Your JSON Output:
```

**Figure 12.** *Sentiment Analysis Prompt*

The sentiment prompt quantifies the sentiment within financial news, classifying articles into five discrete categories (Strong Negative, Negative, Neutral, Positive, Strong Positive), providing a numerical sentiment score (-5 to +5) and mandating evidence-backed rationales. This module aims to provide an analysis mainly on the emotional tone or attitude expressed in texts.

#### 5.4.9. News Impact Analysis

```
[%system%]

You are a stock market news analysis agent that evaluates how news impacts
stock prices, catering to retail investors with three expertise levels (easy,
intermediate, expert).

The news could be discussing any kind of topic, but related to the stock. The
key task is to find an impact to stock price of this news, and possibly
provide a logical explanation behind the stock price prediction in English.

You will also be given a stock name. Create your stock impact analysis report
towards the given stock.

<Requirements>
```

1. Summary of the news is not compulsory, mainly discuss the implication to the stock price and logical explanation behind it.
2. If the news has negligible impact to stock price then you can just give some logical explanation of why it is not important.
3. For the analysis, please find below instructions for your reference. It would be better if you can provide industry-specific viewpoints, as well.
4. **\*\*Easy\*\***: Explain all industry-specific/technical/financial terms (e.g., P/E ratio, EBITDA) in simple language. Give full definition of the terms below the analysis. better
5. **\*\*Intermediate\*\***: Assume basic financial knowledge; skip explanations for common terms (e.g., dividends, market capitalization). Better to give some explanation of complex technicals below the analysis.
6. **\*\*Expert\*\***: Use advanced knowledge (e.g., discounted cash flow, beta volatility and many others) and financial ratios without much explanations.

<OutputFormat>

Please return a JSON format like the following:

```
{  
  "easy": "..."  
  "intermediate": "..."  
  "expert": "..."  
}
```

[%user%]

Example 1: Biotech FDA Approval

Stock: BioPharma Inc.

News Content:

BioPharma Inc. receives FDA approval for its Alzheimer's drug, projects \$1.2B in peak annual sales, and sets a 12-month price target of \$85. A short seller report warns of trial data inconsistencies.

[%assistant%]

```
{
  "easy": "BioPharma Inc. got approval from the FDA (U.S. drug regulators) for its Alzheimer's treatment, which it expects to generate $1.2 billion per year. Analysts predict the stock could reach $85 within a year, but a critical report claims some test results might be unreliable. Risks include competition from larger drugmakers and high R&D costs (money spent developing new drugs).",
  "intermediate": "BioPharma's FDA approval supports a bullish $85 PT (35x P/E), but a $300M per quarter cash burn raises dilution risks. Pipeline catalysts include a Parkinson's drug entering Phase 2 trials.",
  "expert": "BioPharma's Alzheimer's drug approval (FDA label includes broad indication) drives PT to $85 (DCF: WACC 12%, $1.2B peak sales, 55% probability-adjusted). Short seller claims on trial data heterogeneity (p=0.07 in subgroup) may limit near-term upside. With a cash runway of six quarters at the current burn rate, an equity offering (15-20% dilution) is likely. EV/sales at 5x vs. sector 7x reflects pipeline overhang."
}

[%user%]

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

Stock: {{$stock}}

News Content: {{$content}}

Your JSON Output:
```

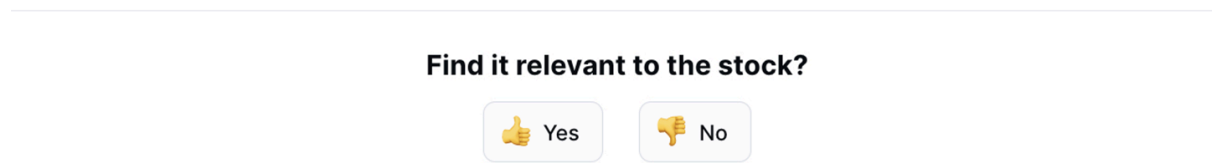
**Figure 13.** *News Impact Analysis Prompt*

The news impact analysis prompt is engineered to systematically evaluate the influence of the news articles on specific stock's price fluctuation, tailoring outputs to three distinct investor expertise levels. For easy-level analysis, technical terms and financial jargon are simplified into plain language (e.g.) explaining FDA approval as U.S. drug regulators approval) while emphasizing risks and rewards in accessible terms. Intermediate outputs assume fundamental level of financial literacy, focusing on actionable metrics like price

targets, cash burn rates, or pipeline catalysts without over-explaining basics. Expert-tier analysis employs advanced valuation models (e.g.) Discounted Cash Flow (DCF) with probability-adjusted revenue scenarios) and sector-specific terminology (e.g.) EV/sales multiples applied for technology sector) to deliver institutional level insights. The prompt prioritizes objectivity by grounding conclusions in cited evidence from the article, such as trial data inconsistencies or regulatory milestones, while aligning outputs with industry dynamics (e.g., biotech dilution risks or semiconductor demand cycles).

Few-shot learning is integrated into the prompt design to calibrate the model's response accuracy across diverse news scenarios. By providing annotated examples like the BioPharma Inc. case, the model learns to generalize patterns such as distinguishing between expertise levels—for instance, defining "cash burn" for novice investors versus discussing "dilution risks" for intermediates. These examples also train the model to prioritize critical financial metrics (e.g., peak sales projections, WACC assumptions) and contextualize them within broader market narratives, such as short seller reports or competitive threats. The structured exemplars further reduce ambiguity by demonstrating how to format rationale (e.g., linking FDA approval breadth to DCF assumptions) while maintaining conciseness. This method ensures adaptability to niche sectors, from pharmaceuticals to fintech, by reinforcing domain-specific causal relationships between news events and stock price trajectories.

#### 5.4.10. Keyword Feedback for Indirect Matching



**Figure 14.** *Simple Yes or No User Feedback System*

A critical feature of the system is its personalization mechanism, which tailors content by capturing news articles relevant to user-specified stocks. Relevance is determined through a combination of explicit mentions and inferred associations within the news data. However, the interpretations of relevance may vary among users. For instance, debates may arise over whether an article about the gaming industry is sufficiently connected to a GPU producer, given differing perspectives on the industry's reliance on specific hardware. To address this subjectivity, the system employs a dynamic feedback loop, enabling users to evaluate articles flagged as relevant by the platform.

The feedback mechanism has been designed to target specific keywords generated. When users mark an article as irrelevant, the system identifies a keyword responsible for the match, removes it from the vector database, and generates a replacement keyword. This ensures that irrelevant associations are excluded from future searches while preserving other valid keywords, thereby enhancing personalization without compromising scalability.

#### 5.4.11. Delivery to External Applications

Discord webhooks offer a simple way to automatically send messages and updates to channels without constructing a bot. By sending an HTTP POST request to a unique URL, any application can deliver real-time notifications with formatted text, images, or interactive buttons directly to Discord clients.



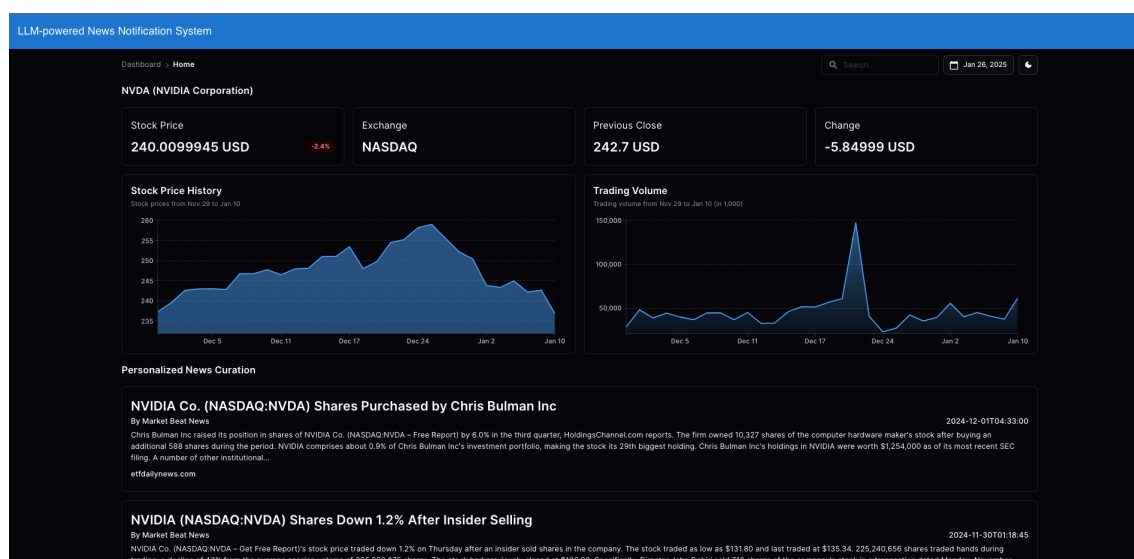
## 6. Experiments and Results

This section examines the iterative challenges encountered throughout the development lifecycle and evaluates the outcomes as a final product.

### 6.1. Experiments

There were three main experiments done during the development process and these are regarding UI/UX, keyword generations and calibration of similarity score thresholds.

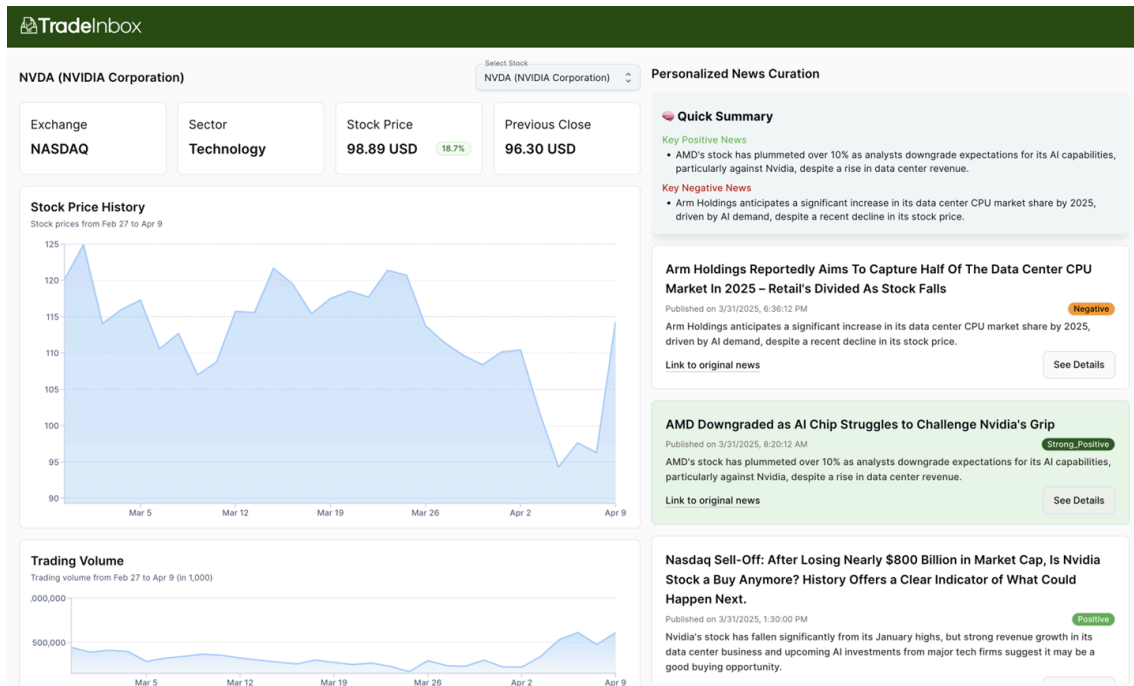
#### 6.1.1. UI/UX



**Figure 15.** *Initially Developed Dashboard UI*

The platform underwent several iterative experiments to optimize its interface design and user experience. The initial UI version, shown above, featured a dark mode theme with a basic layout structure. Stock information data and graphs positioned at the top section, followed by news curation panels below. However, early evaluations revealed shortcomings in both visual hierarchy and usability. The range of initial color scheme lacked for highlighting some articles that require extra attention from the users, while the spatial

arrangement of elements did not guide user attention to the news curation, which are the main elements of this project. These design limitations compromised intuitive navigation, prompting subsequent refinements to improve accessibility and user engagement.



**Figure 16.** *Revised Dashboard UI*

After several iterations of internal testing and gathering feedback from potential users, the team was able to establish a more user-centric UI. Taking the dashboard UI page as an example, the top-bottom layout was changed to a left-right layout to place greater focus on the news content. Additionally, components such as a quick summary were added for user convenience. Lastly, the color scheme was adjusted to light mode to make it easier to highlight the most relevant articles. For example, a news article regarding the AMD downgrade was highlighted in pale green since the LLM categorized it as having a strongly positive sentiment toward Nvidia. Similarly, any strongly negative articles were marked with pale red.

### 6.1.2. Keyword Generation for Embedding Search

```
Stock Name: NVIDIA Corporation
Example Keywords: [
  'graphics processing units',
  'AI technology trends',
  'gaming industry growth',
  'data center demand',
  'autonomous vehicle partnerships',
  'semiconductor supply chain',
  'cloud computing expansion',
  'major competitor AMD',
  'machine learning applications',
  'CEO Jensen Huang statements'
]

Stock Name: Apple Inc.
Example Keywords: [
  'iPhone sales',
  'Apple Watch market',
  'MacBook performance',
  'App Store revenue',
  'smartphone competition',
  'supply chain disruptions',
  '5G technology impact',
  'CEO Tim Cook statements',
  'consumer electronics trends',
  'global chip shortage'
]
```

**Figure 17.** *Keyword Generation Examples*

For the keyword generating module, there were multiple experiments to determine the optimal number of keywords for its embedding search functionality. Through qualitative assessments evaluating contextual nuances and user feedback, alongside quantitative benchmarking of processing latency, the team identified that there is a trade-off between precision and efficiency. While generating more keywords improved detection results, it disproportionately increased processing time due to the increasing numbers of embedding search calculations. Conversely, fewer keywords reduced latency but compromised the result accuracy. 10 keywords emerged as the optimal configuration, ensuring scalability without sacrificing analytical depth.

### 6.1.3. Calibration of Similarity Metrics

News Title	Matched Keyword	Similarity Score
Arm Holdings Reportedly Aims To Capture Half Of The Data Center CPU Market In 2025 – Retail's Divided As Stock Falls	AI Technology Trends	0.437
Nasdaq Sell-Off: After Losing Nearly \$800 Billion in Market Cap, Is Nvidia Stock a Buy Anymore? History Offers a Clear Indicator of What Could Happen Next.	NVIDIA Corporation	0.521
AI datacenters want to go nuclear. Too bad they needed it yesterday	Data Center Demand	0.494
AMD Downgraded as AI Chip Struggles to Challenge Nvidia's Grip	Major Competitor AMD	0.616

**Table 2.** *Examples of Keyword Matching Results*

Initial trials for calibrating acceptable similarity scores for the news feed and user feedback system revealed challenges in balancing precision and recall. For the news feed, an arbitrary threshold of 0.6 was initially set to filter articles. However, this excluded contextually relevant content, as demonstrated by some of the keyword matching results above. Some of the articles could have been omitted despite clear thematic alignment. Consequently, the threshold was adjusted to 0.4 based on qualitative assessments of multiple cases with various sectors of stocks and news from different published dates.

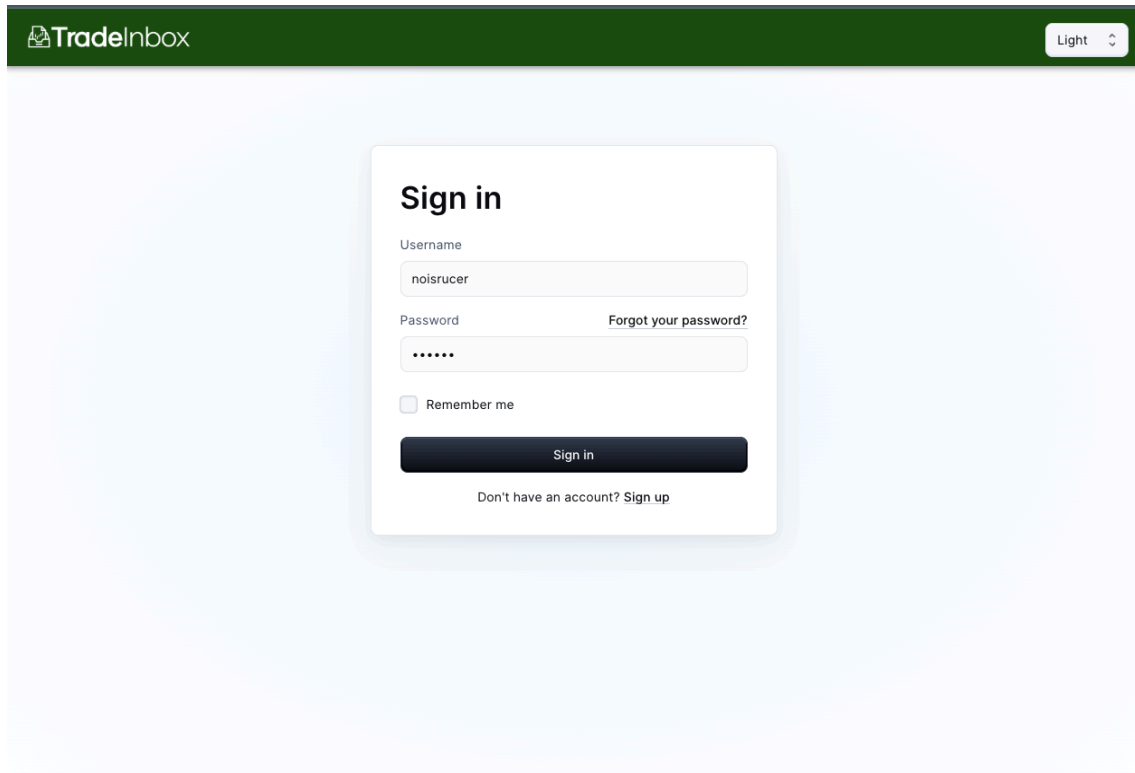
For the user feedback system, the team tried out different methodologies, and adjusting the global threshold was one of the experiments. The user feedback loop adjusted similarity score thresholds globally. If the user marked the article as irrelevant, then the

threshold is increased by +0.5, while relevant responses maintain the current value of the threshold. However, this method proved ineffective when articles matched multiple keywords with varying scores. Consider an article with similarity scores of 0.5 for the keyword "AI Technology Trend" and 0.45 for "Data Center Demand." If multiple users mark "Data Center Demand" articles as irrelevant, raising the threshold over 0.5 would successfully filter out these unwanted matches. However, this adjustment would also incorrectly exclude the relevant "AI Technology Trend" articles. This limitation prompted a transition to keyword-specific threshold adjustments, allowing precise filtering of unwanted terms while keeping relevant matches.

## **6.2. Results**

This section discusses the implementation outcome, mainly focused on the user journey within the platform and how the system supports throughout the journey.

### 6.2.1. Authentication Page

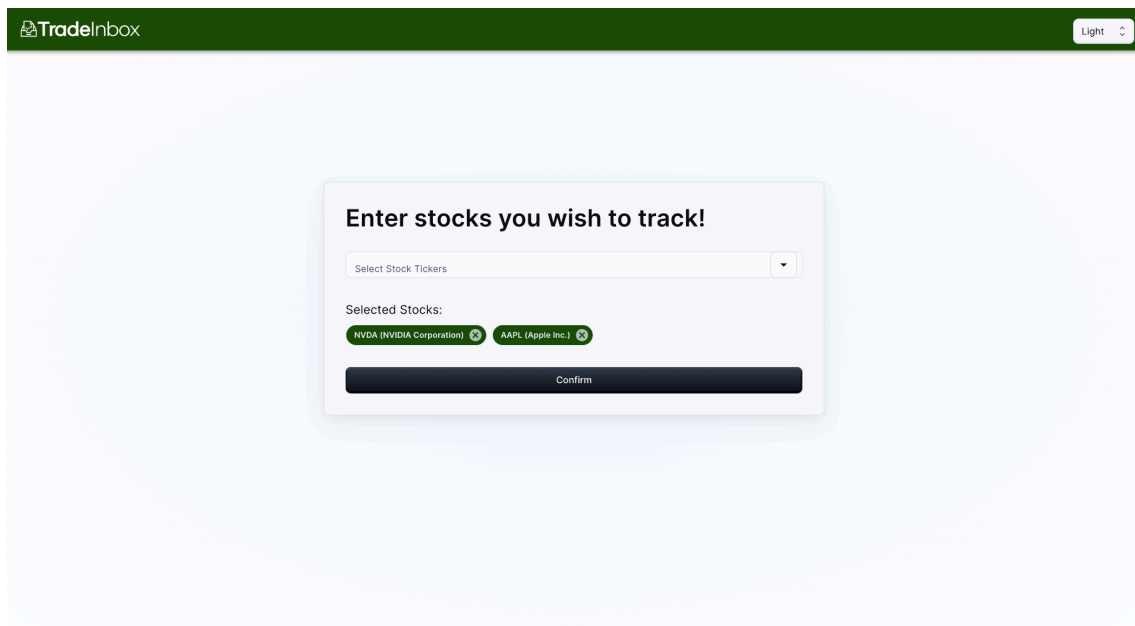


The screenshot displays the TradeInbox authentication page. At the top, a dark green header bar contains the TradeInbox logo on the left and a 'Light' theme toggle with a dropdown arrow on the right. The main content area has a light blue background. Centered on this background is a white sign-in card with a subtle shadow. The card is titled 'Sign in' in bold. It features a 'Username' label above a text input field containing 'noisrucer'. Below this is a 'Password' label above a text input field with masked characters '.....'. To the right of the password field is a link that says 'Forgot your password?'. Underneath the password field is a checkbox labeled 'Remember me'. A dark blue 'Sign in' button is positioned below the checkbox. At the bottom of the card, there is a link that reads 'Don't have an account? Sign up'.

**Figure 18.** *Authentication Page*

Upon initial access, users are directed to an authentication interface requiring username and password credentials. The authentication process, as outlined in Section 5.4.1, is implemented through JWT within the backend architecture to ensure secure and stateless session management.

### 6.2.2. Tickers Selection Page



**Figure 19.** *Ticker Selection Page*

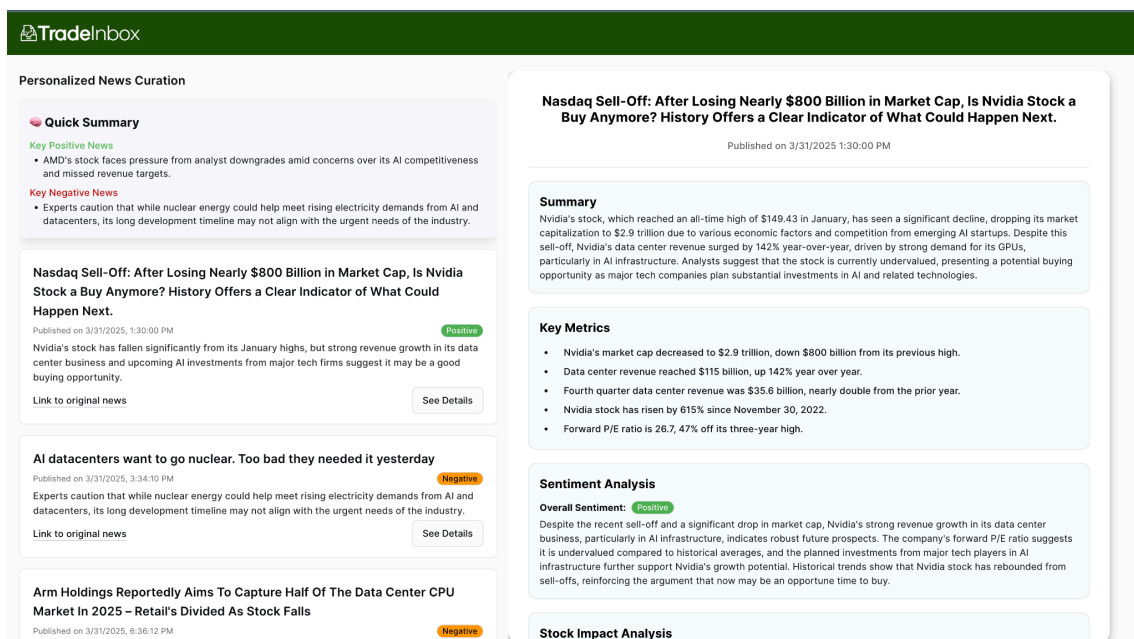
Following successful authentication, users can input a portfolio of securities for real-time monitoring. The system currently integrates a curated database of 4,793 NASDAQ-listed equities to support this functionality. To optimize usability, query operations accommodate both formal corporate designations (e.g.) NVIDIA Corporation) and abbreviated ticker symbols (e.g.) NVDA) for user convenience.

### 6.2.3. Dashboard Page

Referring back to Figure 16, the dashboard page serves as a centralized interface for financial analysis, integrating three functionalities. The stock information display is placed on the left of the screen and allows the users to select a specific stock from a dropdown menu, presenting real-time metrics such as current price, trading volume, and some basic stock information. On the right of the display, the overall summary section leverages sentiment analysis scores to surface key headlines, filtering out the most positive and negative news

articles and featuring them at the top of the feed. Complementing these features, the news feeds dynamically update a real-time stream of articles that are relevant to a particular stock, each entry including a headline, a single-line summary, a source link, and sentiment classification.

## 6.2.4. News Analysis Page

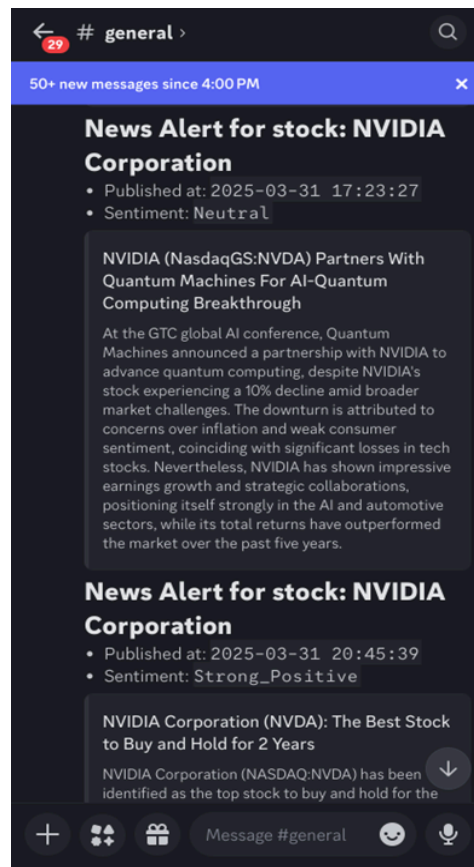


**Figure 20.** *News Analysis Page*

The news analysis page organizes AI-generated insights into a structured, two-pane layout to optimize user workflows. The right panel contains a comprehensive analysis, including summaries, key financial metrics, sentiment evaluations, news impact assessments, and user feedback mechanisms, as detailed in Section 5.4. On the left panel, an aggregated newsfeed and an overview summary section from the dashboard are reused. This enables users to navigate between different articles and select one particular news for in-depth analysis. This design pattern allows more clarity and accessibility for the users and reusability for the developers.



### 6.2.5. Delivery to External Applications



**Figure 21.** *Discord Message Notification*

The system leverages Discord's webhook functionality to send real-time mobile notifications to users, ensuring they stay updated with the latest news. Each notification includes key details such as the news headline, publication date, sentiment and a concise summary. When users click the embedded link within the message, they are seamlessly redirected to the platform where they can access a deeper analysis. This approach enhances user engagement by providing immediate, digestible updates while offering an easy way to explore further details on demand. Also, this functionality is easily extendable to other mobile chatting services supporting the webhook, such as Telegram.

## **7. Conclusion and Future Works**

### **7.1. Conclusion**

This project demonstrates how personalized news curation can transform investor engagement with market information. At its core, the development of a context-aware LLM framework generates relevant keywords, a concise summary and expert-level analysis. The integration of these outputs with embedding-based semantic search presents a scalable approach for filtering highly relevant content, ensuring curated news feeds maintain both accuracy and contextual relevance. By connecting the semantic processes with real-time news data stream, the system delivers an effective solution for real-time, personalized decision support in fast-moving markets.

The project successfully addressed its primary objectives of reducing information asymmetry and improving financial literacy among retail investors. Key achievements include the integration of real-time data collection with advanced LLM processing and embedding search for accurate financial news aggregation, user-driven relevance scoring for dynamic content personalization, a multi-level analytical system (Easy, Intermediate, Expert) accommodating varying expertise levels while enhancing financial education, and an intuitive interface design improving accessibility and usability.

Initial user testing showed strong satisfaction levels (exceeding 70%), confirming the platform's effectiveness in achieving its core objectives. The results validate the system's innovative approach to financial news delivery and its potential to reshape how retail investors interact with real-time information.

## **7.2. Future Works**

The current implementation of the system fulfills the primary objectives of this project and core functional requirements, but the team believes that some enhancements can further refine its robustness and usability.

### **7.2.1. Technology**

The system will implement several performance enhancements to improve scalability and efficiency. These include introducing caching mechanisms and optimized data structures to reduce processing times. While basic asynchronous processing is already in place, additional refinements will maximize the performance. The architecture can be upgraded with advanced asynchronous processing capabilities and robust queueing systems to better manage high-volume concurrent requests during peak periods of news publication.

### **7.2.2. Usability**

The system can establish direct API integrations with leading brokerage platforms, including Interactive Brokers (IBKR) or Alpaca, to enable automated synchronisation of user portfolio data instead of manual stock information entry. The system can maintain precise, up-to-date records of all holdings, such as share quantities. The data fetching could also facilitate enhanced portfolio visualization capabilities within the dashboard interface.

The user interface can include better news feed filtering capabilities. One of the possible improvements could be multi-dimensional sorting options, allowing users to arrange news articles by publication date or sentiment scores. Another would be a topic-based search function that enables users to evaluate articles regarding one particular topic. The system can also adopt a recommender system for content prioritization based on individual interactions.

The platform can generate actionable trading signals by detecting significant sentiment shifts, including extreme positive-to-negative reversals and other critical pattern changes if users enable these features. These alerts can be delivered through in-app pop-up notifications or via Discord webhook integrations to send time-sensitive message.

### **7.2.3. Testing**

The system would benefit from more rigorous domain-specific testing. While financial and investment analysts have reviewed the content generated by the platform partially, constructing a large labelled dataset for financial news sentiment analysis and accessing a database of equity research reports from investment banks would significantly enhance the model's ability to replicate professional-level analysis generation.

Additionally, user testing was limited to a single iteration due to time constraints. More extensive testing cycles are necessary to identify potential improvements or defects that may have been overlooked during initial development.

## Citation

[1] J. Wittenstein, "Quantum Computing Stocks Drop as Nvidia CEO Jensen Huang Sees Use Years Away," *Bloomberg.com*, Jan. 08, 2025.

<https://www.bloomberg.com/news/articles/2025-01-08/quantum-computing-stocks-drop-as-nvidia-ceo-sees-use-years-away> (Accessed Jan. 8, 2025).

[2] J. M. Cherian, S. Gupta, and C. Mandl, "Wall Street ends lower as blowout job data spooks traders," *Reuters*, Jan. 11, 2025. Available:

<https://www.reuters.com/markets/us/futures-drop-caution-ahead-key-payrolls-data-2025-01-10/>

[3] C. Valetkevitch, "S&P 500 loses \$5 trillion in two days in Trump tariff selloff," *Reuters*, Apr. 05, 2025. Available:

<https://www.reuters.com/markets/global-markets-wrapup-1-2025-04-04/>

[4] A. Atkins, M. Niranjana, and E. Gerding, "Financial news predicts stock market volatility better than close price," *The Journal of Finance and Data Science*, vol. 4, no. 2, pp. 120–131, Jun. 2018, doi: 10.1016/j.jfds.2018.02.002.

[5] "Who Reads Finance News? Traffic and User Behaviour," *FinText*, Feb. 21, 2023.

<https://www.fintext.io/case-studies/benchmarking/who-reads-financial-news-web-traffic-and-user-behaviour/> (Accessed Nov. 26, 2024).

[6] Bloomberg LP, "Getting Started Guide for Students (English)," Bloomberg LP. [Online]. Available:

<https://data.bloomberglp.com/professional/sites/10/Getting-Started-Guide-for-Students-English.pdf>. (Accessed: Apr. 10, 2025)

- [7] "News API: Search Global News Data for Insights and Analysis,  
"https://www.newscatcherapi.com. [Online]. Available:  
https://www.newscatcherapi.com/docs/v3/documentation/get-started/overview. (Accessed:  
Oct. 15, 2024).
- [8] "Picking a vector database: a comparison and guide for 2023," benchmark.vectorview.ai.  
[Online]. Available: https://benchmark.vectorview.ai/vectordbs.html. (Accessed: Oct. 15,  
2024).
- [9] J.-T. Huang et al., "Embedding-based Retrieval in Facebook Search," in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD '20)*, pp. 2553–2561, Aug. 2020. doi: 10.1145/3394486.3403305.