DSA4265: Sense-making Case Analysis (Economics and Finance)

Application Track: Introduction to BuffettAI

Name	Student Number
Chew Yu Cai	A0234471J
Ethan Cheung	A0239399H
Merson Cheong	A0238598H
Nicholas Lee	A0240089E
She Chee Yee	A0240383L
Wilfred Woo	A0238902A

Table of Contents

Abstract

This report introduces BuffettAI, a system designed to simulate Warren Buffett's investment advice using Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG). The project addresses the challenge of tracking vast stock market data by leveraging AI to process and summarize financial information in Buffett's distinctive style. The architecture involves fine-tuning a Llama 3.2 1B model with a curated dataset of Buffett's Q&A transcripts, quotes, and synthetic queries to capture his tone and reasoning. A hybrid RAG framework enhances response accuracy by combining internal data (e.g., shareholder letters, Berkshire trades) with dynamic web searches.

Evaluation metrics reveal that the fine-tuned model excels in "Buffett-Likeness" but trades off some general clarity and relevance compared to baseline models. The RAG component demonstrates robust performance in routing queries and retrieving high-quality information, achieving high accuracy in responses. Key improvements suggested include expanding the training dataset and refining document chunking techniques. BuffettAI offers investors an intuitive interface to access Buffett's wisdom, with potential future applications as a personalized financial assistant.

Keywords: Warren Buffett, Large Language Models, Retrieval-Augmented Generation, Financial AI, Investment Advice.

1 Introduction

Given the large size of stock markets today, with each country housing hundreds or even thousands of individual stocks, it can be challenging to track all of them effectively. Major multinational companies like Apple, NVIDIA, and Tesla dominate the market (Tyagi, 2025), and news articles about these stocks are published daily in various languages (BBC, n.d.; Prisa, 2017). The rise of Artificial Intelligence (AI) and GPT models has made it easier for investors by processing large volumes of documents to provide concise summaries of stock performances. By fine-tuning large language models to deliver insights and offer advice on a range of stocks, investors can now gather multiple perspectives without needing to consult other investors. In light of Warren Buffett's dominance in the financial and investment world, our goal is to leverage Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) models to create a system that simulates Warren Buffett's responses to investment-related queries—an intelligent advisor we call *BuffettAI*.

In this report, we introduce the architecture of BuffettAI, while explaining the various datasets used in the fine-tuning of the LLM and to supplement the RAG model to enable the operationality of BuffettAI. All relevant files needed to run the model can be found in **Appendix** alongside their descriptions.

2 Current Measures and Literature

Multiple variations of language models have been considered to provide advice, particularly in the field of Causal Language Models (CLM) which have the Transformer architecture as the foundation (Vaswani et al., 2017). Multiple types of CLMs such as GPT-2 and GPT-3 have been introduced as well to facilitate text generation and summarisation. Other methods also include Masked Language Modelling such as BERT which offers bidirectional context to make predictions (Sinha et al., 2021). In our project, we decided to use CLM models to finetune due to their ability for autoregressive text generation, hence making them suitable as a chatbot compared to MLM models which are not designed to generate text autoregressively, but rather are generating text based on context (Radford et al., 2018; Yasar, 2024). Furthermore, CLMs have demonstrated strong performance when fine-tuned with domain-specific data, making them highly effective for specialized applications such as investment advice (Radford et al., 2019). In our project, we chose CLMs for our fine-tuning phase, which will be described in greater depth in the next section.

Next, to improve the quality of natural language generation tasks by integrating external knowledge through retrieval mechanisms, the RAG architecture demonstrates that combining retrieval and generation could significantly enhance the performance of generative models on knowledge-intensive tasks (Lewis et al., 2021). Many dense retrieval methods have been introduced to detect semantically similar content even when explicit keywords are absent, such as Sentence Transformers (Reimers & Gurevych, 2019) and OpenAI embeddings (Neelakantan et al., 2022). Apart from that, to ensure further precision in keyword-matching scenarios, lexical retrieval methods such as BM25 have been introduced to assure high-recall retrieval for domain-specific inputs (Robertson & Zaragoza, 2009). Based on these various means introduced today, we decided to use these methods to ensure that the RAG model is consistently informed in the system to produce responses of high quality.

3 Description of Dataset

This section describes the various datasets utilised in the dataframe, and the relevant transformations applied to generate the datasets in the format relevant to the task at hand.

Fine-Tuning LLM. To fine-tune the LLM to closely emulate Warren Buffett's distinctive tone and word choice, we initially selected a set of authentic responses from a Question-and-Answer compilation featuring Buffett. This dataset was meticulously cleaned by removing responses from individuals other than Buffett and restructuring it into a two-column dataframe: one for questions posed and the other for Buffett's corresponding answers. However, this resulted in only 439 rows of question-answer pairs, insufficient for effective fine-tuning. To address this, we augmented the dataset by incorporating an additional 90 question-answer pairs derived from Warren Buffett quotes available on a public website, utilizing an LLM to generate appropriate questions for each quote. Furthermore, we supplemented the dataset with 100 synthetically generated question-answer pairs created by prompting an LLM to respond to generic queries, such as greetings, explicitly mimicking Warren Buffett’s conversational style. This comprehensive approach significantly expanded the dataset, enhancing the LLM's ability to produce responses reflective of Buffett's unique linguistic patterns and perspectives.

RAG Model. In this section, shareholder letters from different stocks such as Berkshire, MS, JPM and more were included for better understanding of the stocks. Trades from Berkshire were also extracted so as to ensure that the chatbot has a niche specific to Berkshire, which is similar to what Warren Buffett would be.

Table 1 describes all the data used in the project, its use, and its respective sources.

Table 1: Data used for Project

S/N	Use	Name of Dataset	Source of Dataset
1	Fine-Tuning	Q&A Transcript from Warren Buffett	Buffett FAQ
2		Quotes from Buffett and generated questions	Sarwa Blog Page
3		Synthetic Generic Queries and Responses	OpenAI ChatGPT 4o
4	RAG Model	Shareholder Letters (JPM)	Home | JPMorganChase
5		Shareholder Letters (AMZ)	Amazon.com, Inc. - Annual reports, proxies and shareholder letters
6		Shareholder Letters (MS)	Investor Relations | Morgan Stanley
7		Shareholder Letters (BRK)	Shareholder Letters
8		Shareholder Letters (GOOGL)	Investor Updates - Alphabet Investor Relations
9		Shareholder Letters (GS)	Investor Relations | Goldman Sachs
10		Shareholder Letters (MSFT)	Home page
11		BRKA Trades	LSEG

4 Overview of Model Architecture

This section describes the architecture and structure of the BuffettAI Model.

4.1 Phase 1: Fine-Tuned LLM

Throughout the development process, multiple iterations were undertaken to balance tone fidelity, response quality, and model size. Below is a detailed account of two significant iterations that had been trialled, the second of which was eventually selected as our final LLM model:

4.1.1 Other Methods Explored: Financial Domain Base Model

The objective was to leverage a lightweight, pre-trained language model specialized in financial data as the foundation for fine-tuning. The chosen model for this iteration was **FinChat-XS**, developed by Oopere. FinChat-XS is a compact language model designed to address queries related to finance, markets, investments, and economics in a conversational style.

Rationale for Model Choice. Having been pre-trained on financial data, this makes FinChat-XS a natural starting point for tasks within the financial domain. Apart from that, with approximately 300 million parameters, FinChat-XS offered a balance between computational efficiency and performance, facilitating quicker training and inference times.

Stop Token Issues. During deployment, it was observed that the model occasionally generated outputs that included unintended stop tokens, leading to abrupt or incomplete responses. This behavior was attributed to inconsistencies in the handling of stop tokens within the model's architecture.

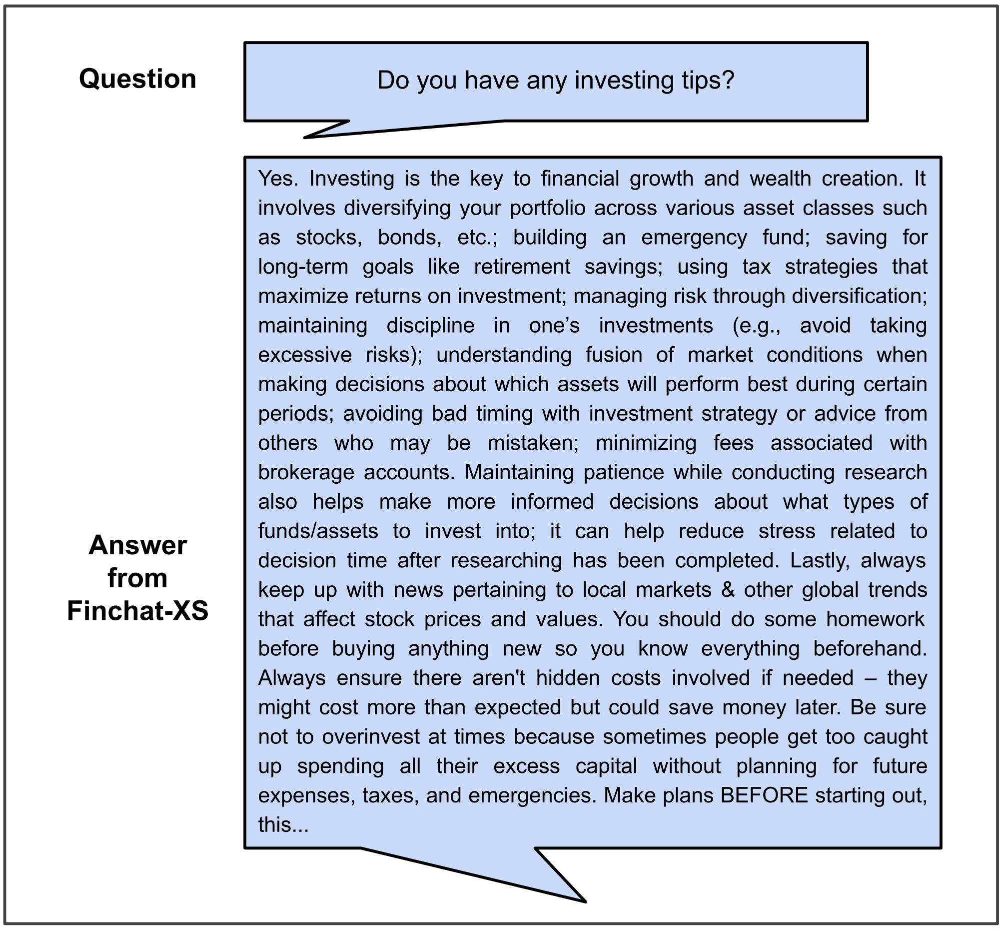

Figure 1: Stop token issue sample response

As shown by **Figure 1**, while the response demonstrates a broad understanding of investment principles, it is verbose and lacks the succinctness and clarity desired for user engagement. Additionally, the presence of stop token issues necessitated the exploration of alternative models.

In view of this issue, this led to exploration for alternative methods in other iterations.

4.1.2 Selected Model: Llama 3.2 1B Model with Enhanced Dataset

Our objective was to utilize a more robust language model capable of delivering high-quality responses with improved clarity and coherence, which would be a more suitable start point for fine tuning, given the complexity of this task and the limitations of our fine tuning dataset. The selected model for this iteration was Llama 3.2 1B, developed by Meta. This model is part of the Llama 3.2 series, known for its advanced capabilities in language understanding and generation (Meta, 2024), which are described below:

Enhanced Performance. Llama 3.2 1B has demonstrated strong performance in various benchmarks, indicating its capability to handle complex language tasks effectively.

Multilingual Support. The model offers improved multilingual support, accommodating a diverse user base and facilitating broader applicability.

The Llama 3.2 1B model was fine-tuned using the enhanced dataset comprising authentic, augmented, and synthetic question-answer pairs to tailor its responses to emulate Warren Buffett's distinctive style. The adoption of Llama 3.2 1B, combined with the enriched dataset, resulted in responses that were more concise, contextually relevant, and closely aligned with Buffett's tone and word choice. The model is evaluated in more detail below:

4.1.3 Fine Tuning of Model

The primary objective of this fine-tuning effort was to develop a compact, efficient, and expressive large language model (LLM) capable of emulating the tone, reasoning, and conversational style of Warren Buffett.

The choice fine tuning framework is Unsloth, built on top of Hugging Face Transformers, the framework provides optimisation in both speed and memory, allowing us to efficiently tune the model using the limited GPU capacity. We also opted to use the parameter-efficient fine tuning model.

Ultimately, we configured the training steps to the following parameters as shown in **Table 2**:

Table 2: Parameters used for model

Parameter	Value
LoRA Rank	64
LoRA Alpha	64
LoRA dropout	0
Trainable parameters	~11 million (1.1% of the model)

The model was then trained on 450 steps resulting in a final training loss of 0.1, from starting at 2.9. It was then saved as a 8-bit quantised model and subsequently built and served on Ollama.

4.1.4 Assessment of Model Effectiveness

In order to assess the effectiveness of our large language model (LLM) deployments, we employed a set of metrics that provide both quantitative and qualitative insights into model performance. Specifically, the following criteria were used:

1. <u>Relevance</u>. Measures how effectively the generated response addresses the user's query. A higher relevance score indicates that the content is closely aligned with the specific question or prompt.
2. <u>Accuracy</u>. Evaluates the factual correctness of the response. This includes factual claims, numerical data, and references to external events or information. A higher accuracy score suggests fewer factual errors or unsupported statements in the response.
3. <u>Clarity</u>. Reflects the level of coherence, fluency, and legibility of the text. A response with high clarity should be easily understandable and well-structured, requiring minimal effort from the reader to interpret the meaning.
4. <u>Buffett-Likeness</u>. A custom metric devised to gauge how closely the language and perspective of the response align with Warren Buffett's characteristic style, wisdom, and personality. A higher Buffett-likeness score indicates that the model mimics Buffett's tone or philosophy more convincingly.

We used the Groq API to generate these metrics in an automated fashion. Specifically, after each response was produced—whether from the fine-tuned local (Ollama) model or a baseline model—we sent the text, along with an evaluation prompt, to the Groq endpoint. The Groq service then returned a JSON object containing numeric scores for relevance, accuracy, clarity, and Buffett-likeness, as well as an overall average. This approach allowed us to maintain a consistent, standardized evaluation across all queries while leveraging a specialized LLM-based evaluator.

4.2 Phase 2: RAG Model

This section details the Retrieval Augmented Generation (RAG) architecture employed to enhance the Large Language Model's (LLM) ability to provide accurate and contextually relevant responses to user queries.

The architecture is designed to incorporate an agent to decide the following:

1. Information retrieval from both structured data stores or external web sources, ensuring comprehensive and up-to-date answers.
2. LLM model to use, ensuring a better user experience.

4.2.1 Overall Rag Architecture

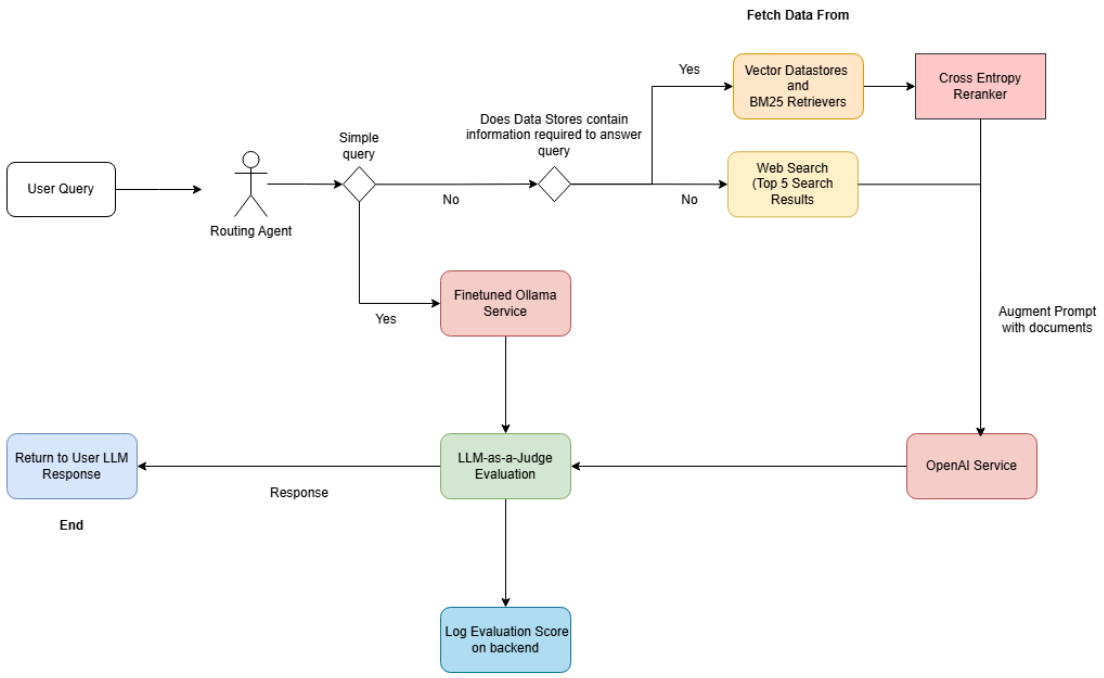

Figure 2: Diagram detailing the overall flow of RAG model

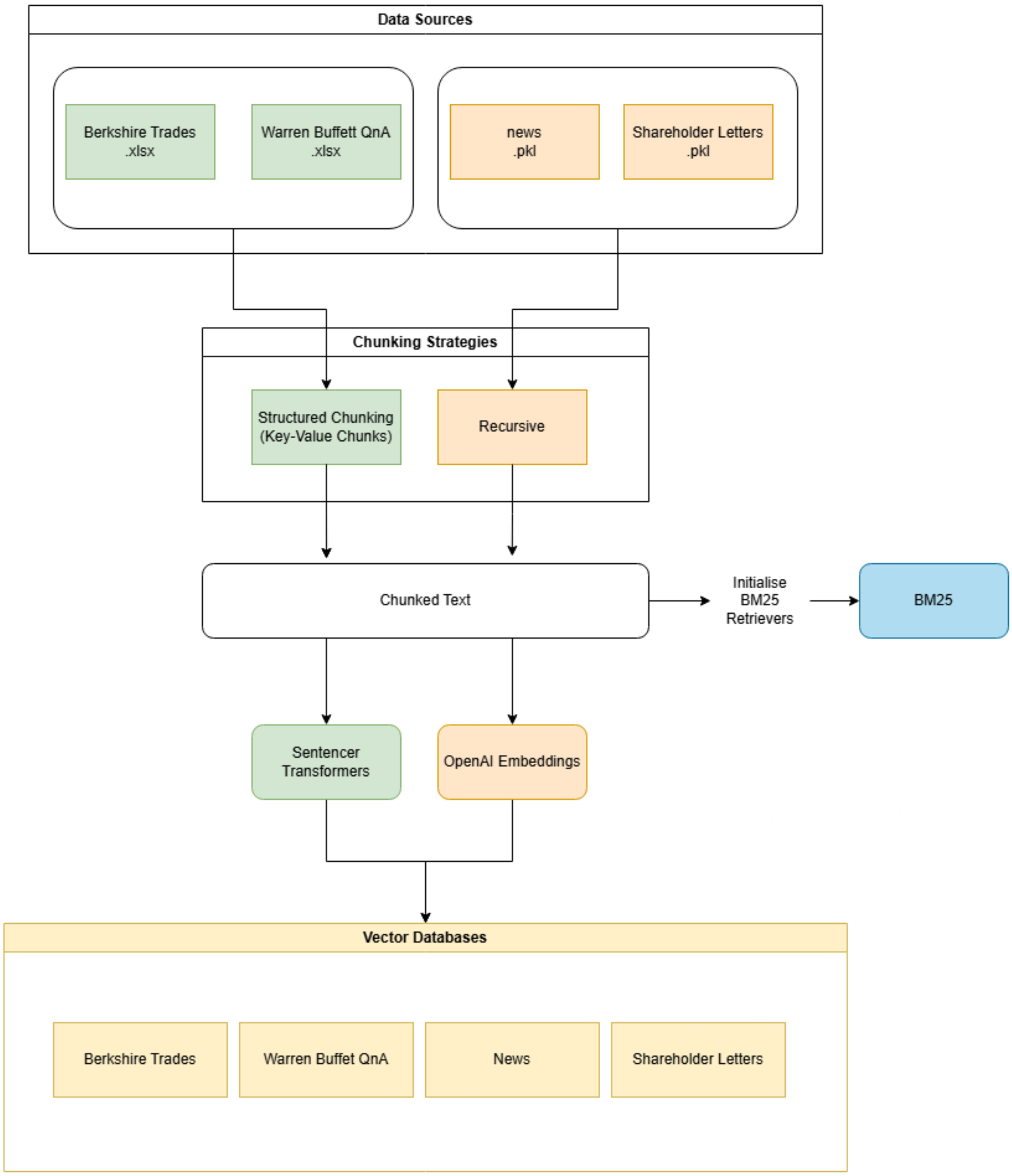

Figure 3: RAG Model Data Handling Diagram

4.2.2 Rag Pipeline

Upon startup, the backend initializes the retrieval databases by checking and loading (or building) the FAISS and BM25 indexes. Once a query is sent through the system, the following steps occur:

1. Document Retrieval (Optional)
2. Reranking (Optional)
3. Prompt Engineering
4. Response Generation
5. Response Evaluation
6. Response Delivery

4.2.3 Data Ingestion and Retrieval Architecture

The system employs a multi-faceted approach to ingest and retrieve information from diverse data sources. As illustrated in the diagram, data originates from four distinct sources: "Berkshire Trades" and "Warren Buffett Q&A" (both Excel files), "News" and "Shareholder Letters" (both in `.pkl` format).

Data Preparation. To prepare this data for efficient retrieval, two primary chunking strategies are utilized: Document-based-structured Chunking (Key-Value Chunks" and Recursive chunking. The resulting chunks are then processed in parallel using "Sentence Transformers" and "OpenAI Embeddings" respectively to generate vector representations. These embeddings are stored in separate vector databases, each corresponding to the original data source ("Berkshire Trades", "Warren Buffett Q&A", "News", "Shareholder Letters").

Use of Retrievers. Finally, a BM25 retriever is initialized to facilitate retrieval based on the chunks directly, offering a combination retrieval mechanism alongside the vector database approach. This hybrid architecture enables the system to leverage both semantic similarity (via embeddings) and term frequency-based relevance (via BM25) for comprehensive information retrieval.

4.2.4 Query Routing and Data Source Selection

Query Routing. Upon receiving a user query, a Routing Agent evaluates whether the query can be adequately answered using information present within the system's internal Vector Datastores and BM25 Retrievers. This decision is based on the query's semantic content and the scope of the indexed data.

Data Source Selection. If the Routing Agent determines that the internal data stores are sufficient, the system proceeds to retrieve information directly from these sources. Conversely, if the query requires information beyond the scope of the internal data, the system initiates a Web Search to fetch the top 5 Relevant search results.

4.2.5 Information Retrieval and Augmentation

In the process of information retrieval, there are two types of queries which require different types of information: one that can be answered using internal data stores, and another which requires external information. These two types of queries will be dealt with in depth below:

Routings to internal data stores. For queries routed to the internal data stores, a combination of vector search and BM25 retrieval is utilized. Vector search leverages semantic similarity to identify relevant documents, while BM25 provides term-based relevance scoring, ensuring a comprehensive retrieval strategy.

External information requirement. For queries requiring external information, a web search is performed, and the top 5 search results are extracted. This step ensures that the LLM has access to up-to-date information beyond the system's internal knowledge base.

The retrieved information, whether from internal data stores or web search results, is then used to augment the prompt sent to the fine-tuned LLM Service. This augmentation provides the LLM with the necessary context to generate accurate and relevant responses.

4.2.6 Response Generation and Evaluation

The augmented prompt is processed by a fine-tuned LLM, which generates a response based on the retrieved information. This finetuning ensures that the LLM is optimized for the specific domain and task at hand.

Evaluation of Response: LLM as a Judge. The generated response is then evaluated using an LLM-as-a-Judge approach. This involves using another LLM to assess the quality, relevance, and accuracy of the response. This evaluation provides a quantitative measure of the system's performance.

Generation of final output. The final output consists of the LLM's response and an evaluation score. The LLM response is delivered to the user via the Frontend (Described in **Section 7 Application Interface**), facilitated by a FastAPI request. The evaluation score is logged on the Backend for analysis and system improvement purposes.

5 Model Performance and Assessment Metrics

5.1 Results of Fine-Tuned LLM

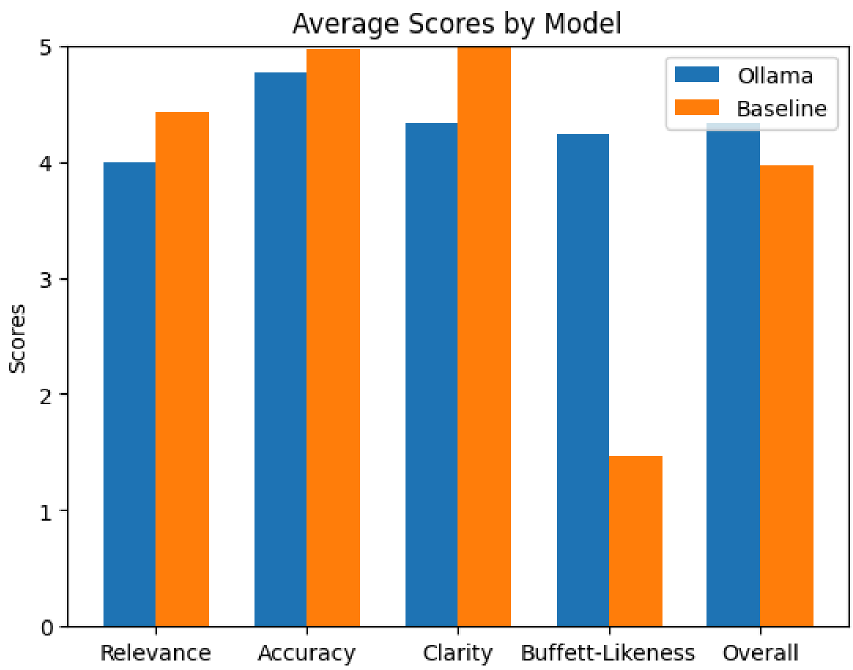

Figure 4: Evaluation of average scores of fine-tuned model

Figure 4 above visually summarizes the performance comparison between the fine-tuned model and the baseline model across several key metrics: Relevance, Accuracy, Clarity, Buffett-Likeness, and Overall score.

Relevance, Accuracy and Clarity. Conversely, the baseline model slightly outperforms the fine-tuned model in Relevance, Accuracy, and Clarity. This trend suggests that fine-tuning on a relatively small dataset, primarily to replicate Buffett's unique style, may have inadvertently led to mild overfitting. As a result, the fine-tuned model tends to prioritize stylistic imitation, possibly sacrificing some general linguistic and factual understanding.

Buffett-Likeness. The fine-tuned model significantly surpasses the baseline model in this category. This indicates that the fine-tuning process successfully instilled Warren Buffett's distinctive persona, effectively replicating his wit and characteristic communication style.

Overall Performance. Despite the fine-tuned model's superior Buffett-Likeness, the baseline model maintains slightly higher overall performance. This suggests a trade-off occurred in the fine-tuning process—achieving authenticity in personality at the expense of broader clarity and general knowledge comprehension.

To illustrate the practical implications of these observations, the following qualitative comparison of responses to the query of *"What are your thoughts on investing in gold?"*.

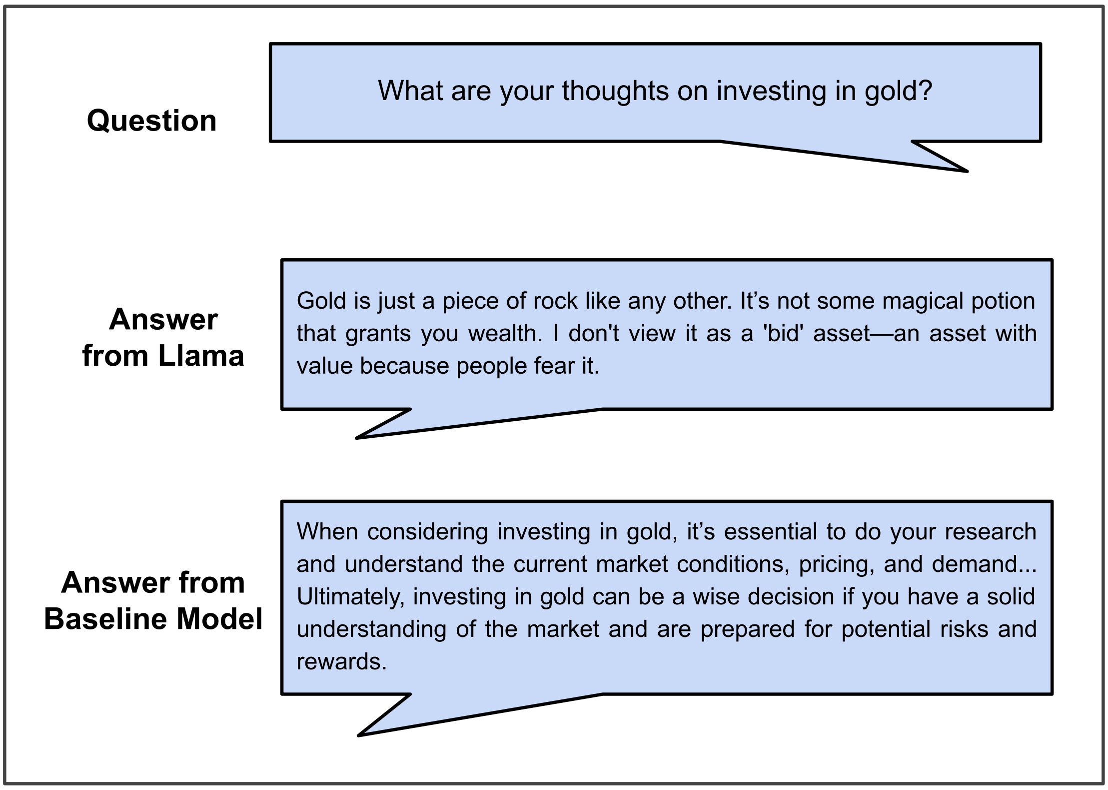

Figure 5: Responses contrasting Ollama and Baseline response

Ollama response. This response in **Figure 5** demonstrates distinct Buffett-like wit and skepticism. It provides a succinct, opinionated, and stylistically authentic portrayal of Buffett's historical stance on gold investments. However, the brief, dismissive tone may limit explanatory clarity and completeness of detail.

Baseline response. This answer, although factually accurate and clear, lacks the distinctive Buffett viewpoint. Instead, it offers balanced, general investment advice typical of standard financial communication. It achieves better clarity and comprehensiveness but at the expense of the unique Buffett personality and stylistic authenticity.

5.2 Performance of RAG Model

Apart from answering from the Buffett viewpoint, we also check for the accuracy of information retrieved

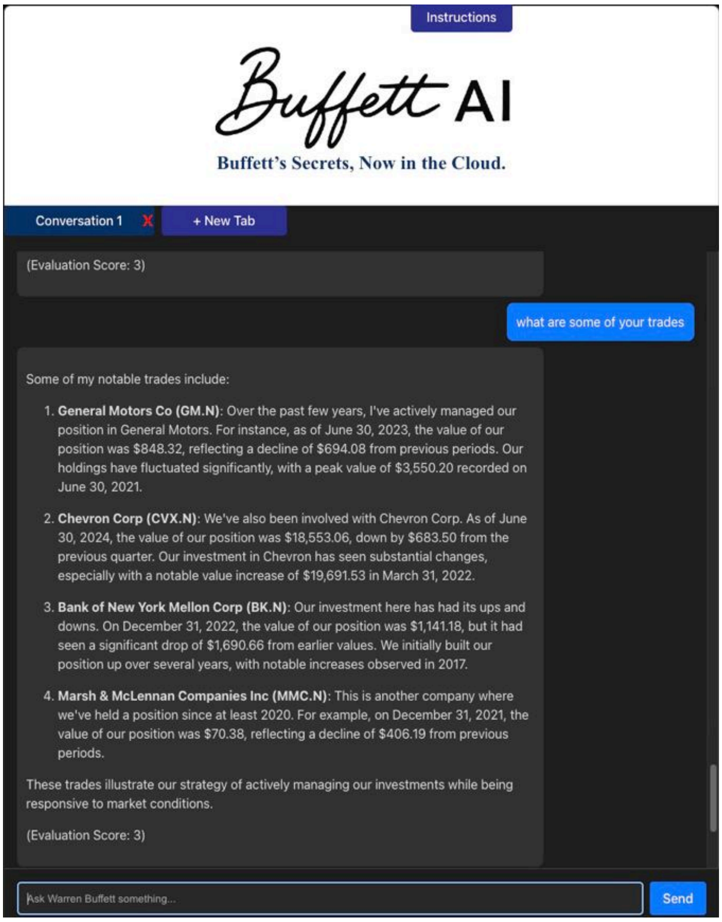

Figure 6: Routing to Internal Database

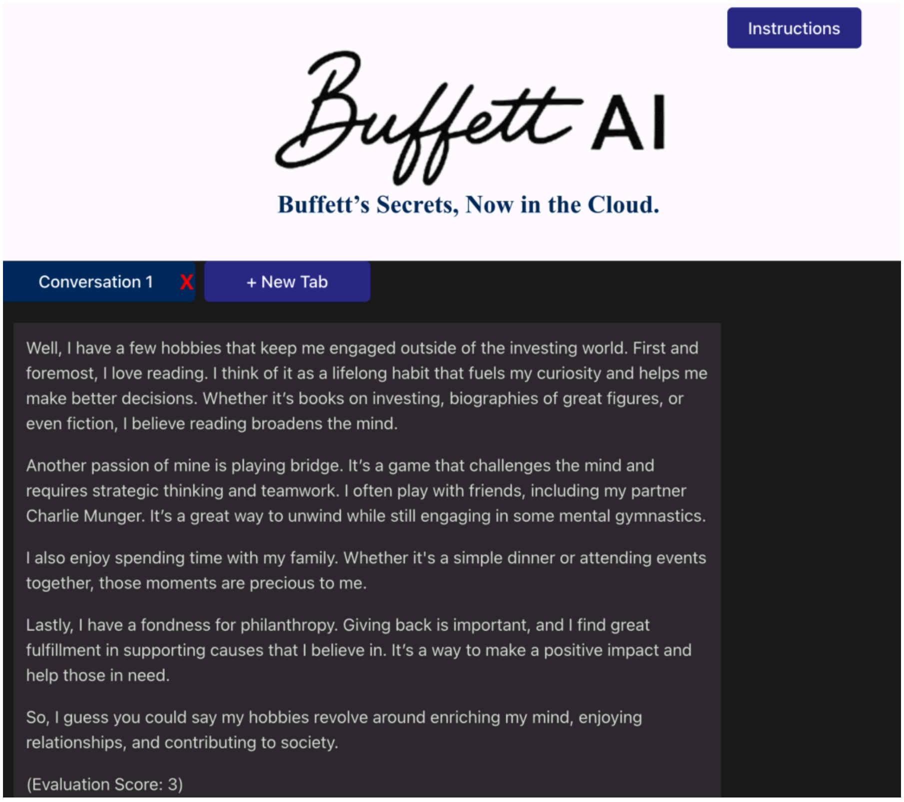

Figure 7: External Data Requirement to question "What are some of your hobbies?"

Routing to Internal Database. As mentioned in prior sections, queries routed to the internal database will have a combination of vector-bound and BM25 retrieval tools to extract the data, and evaluation quality has been rated out of 3. Referencing to **Figure 6**, BuffettAI has been able to extract data from the internal database easily and produce a concise answer which had been rated 3 out of 3.

Requiring external information. With regards to queries requiring external information, BuffettAI has proved to easily access online resources to obtain the answers to produce it concisely. This is seen in **Figure 7** where BuffettAI was able to answer the question about hobbies when the answer is not found in the internal database, and the answer has been rated to be 3 out of 3.

6 Discussion

6.1 Fine-Tuned Model

The observed performance differences may be intuitively explained as follows:

Overfitting due to Limited Training Data. With a relatively small training dataset consisting primarily of Warren Buffett's Q&A interactions and commentary, the llama model may have learned specific stylistic patterns too rigidly, thereby compromising its ability to generalize to broader topics or maintain comprehensive factual coverage. In order to improve the results, more training data should be incorporated.

Persona-Style vs General Knowledge Trade-off. Capturing Warren Buffett's persona necessitates the use of short, pithy, and occasionally indirect language. Such style naturally tends to sacrifice some explanatory clarity and detailed accuracy. By contrast, the baseline model, not constrained to a particular persona, defaults to generic language patterns that emphasize clarity, accuracy, and comprehensive coverage. This hence results in the differing results observed between the two.

6.2 RAG Model Performance

High evaluative ability. The routing agent has proven to easily route the queries into different themes, and produces responses of high quality. This could be attributed to the use of hybrid retrievals which have been proven to work well (Abraham et al., 2024). Apart from that, the routing agent aided in the decision of the best search strategy, thereby enabling computational efficiency.

Incorporation of web-search feature. Through the addition of this feature, this mitigates hallucinations of the LLM and ensures proper coverage of the answer rather than full reliance on the data which may not always be the most accurate. Therefore, the system is able to dynamically supplement internal knowledge with up-to-date, real-world information, especially for time-sensitive or less-covered topics. This heightens response reliability, which allows responses to be consistently highly rated.

7 Application Interface

This section describes the application interface for BuffettAI. Refer to **Figure 8** for a sample interface for BuffettAI.

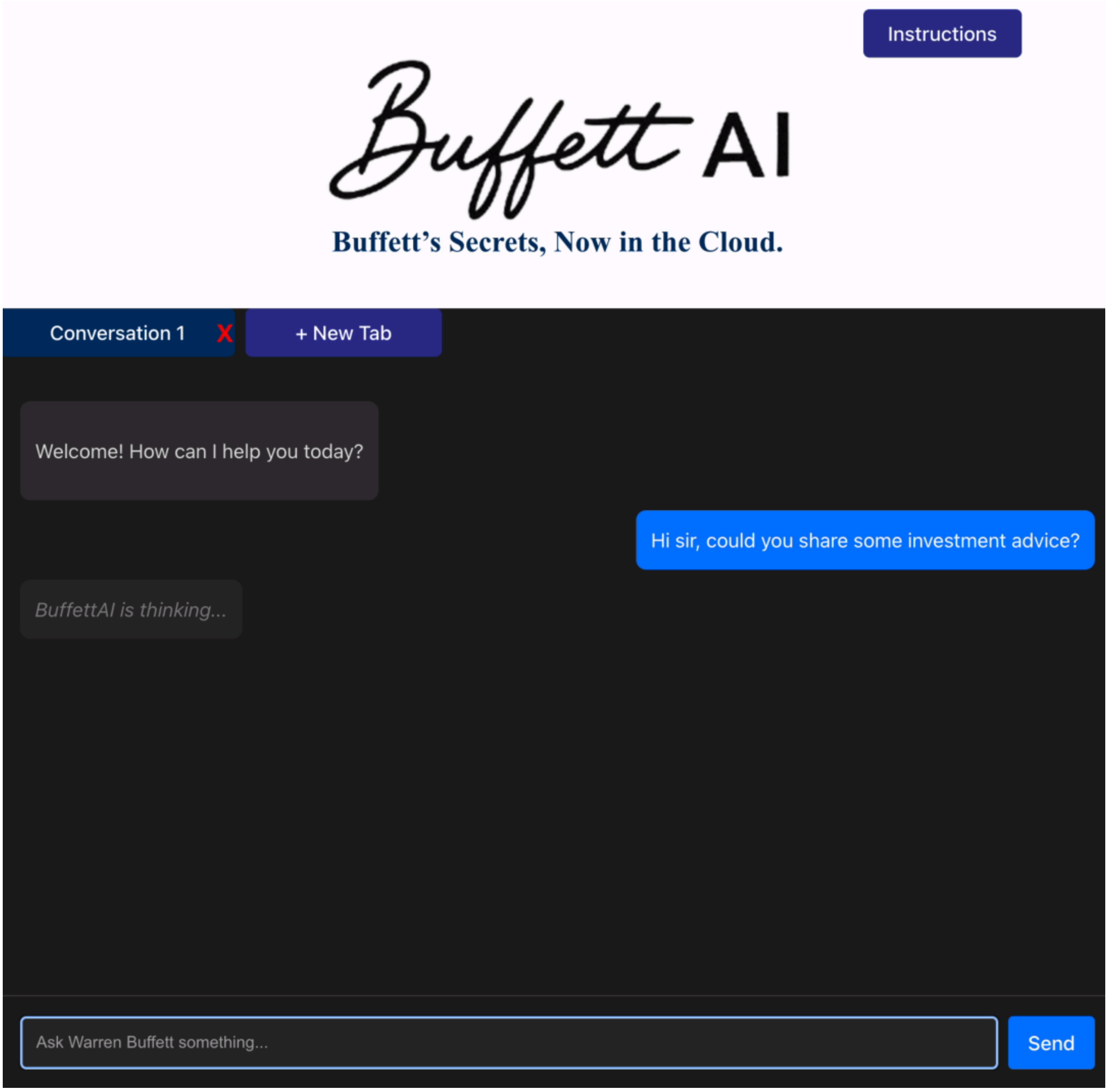

Figure 8: Sample interface for BuffettAI

The Buffett AI application employs a minimalist yet functional interface designed to prioritize clarity and ease of interaction. The primary workspace follows a card-based or segmented layout, where each conversation thread (e.g., "*Conversation 1*") is treated as an independent unit. The input area is situated at the bottom of the interface, adhering to conventional chat-based UI design principles (e.g., messaging applications), which enhances intuitive usability.

The following describes several features of the interface:

Addition of new tabs. A "+ *New Tab*" feature enables asynchronous multi-threaded interactions, allowing users to initiate parallel discussions while retaining prior context. This is a feature crucial for comparative analysis of investment strategies.

Incorporation of instructions tab. An instructions tab is positioned at the top of the interface, providing explicit guidance on how to interact with the application.

Scrollable element. The chat space functions as a scrollable element, enabling users to review previous messages. To maintain consistency, the header and input sections remain static throughout the application.

Processing Icon. Upon submitting a query, the system displays the message "BuffettAI is thinking..." while processing the response. This serves as both a confirmation of receipt and an indicator of ongoing computation. To reinforce this, the message employs alpha cycling, a pulsing fade-in-and-out animation, signaling active processing. The absence of this visual cue may indicate a system failure, suggesting the need for an application restart.

Therefore, the BuffettAI interface exemplifies an effective chat-based paradigm, prioritizing simplicity while embedding sophisticated features to support actionable investment analysis. Its streamlined design minimizes cognitive load, enabling users to focus on extracting insights rather than navigating complex controls.

8 Conclusion

Room for Improvements. There are several improvements that can be done for BuffettAI that could potentially make it more functional for users which are described as follows: (1) With the reflection of Warren Buffett's unique personality traits, this potentially came at the cost of the broader linguistic ability. Therefore, increasing the size of the training set allows future versions of the model to better balance a persona replication with general knowledge and clear, comprehensive responses; (2) Document chunking enhancements such as adaptive and overlapping chunking could be incorporated to dynamically adjust the chunk size while also preserving the contextual meaning; (3) Improvements for the interface could be included to further enhance the user experience while using BuffettAI such as voice-to-text input, Buffett's AI-generated voice output, and dynamic feedback mechanism (i.e., thumbs up/down for the generated output).

Conclusion. All in all, BuffettAI excels in reflecting Warren Buffett's unique personality and communication style, achieving high Buffett-Likeness scores. With the help of the RAG model which enabled easy routing of queries to the internal database or facilitating the web-search to find the answer to the query, BuffettAI is able to produce responses resembling the insights from Warren Buffett in real time. Beyond its current capabilities, future improvements such as expanding the training dataset and enhancing the user interface which can significantly boost both performance and user engagement. Ultimately, BuffettAI offers investors and enthusiasts an intuitive way to interact with the distilled wisdom of one of the greatest investors of all time. With further development, it could serve not just as a conversational tool, but as an investment coach or even a personalized financial assistant - thereby embodying the principles of Warren Buffett in a modern and accessible format.

9 Appendix

Table A-1 describes the various coding files used to obtain the model. More details about the files and data can also be found in the Github link here.

Table A-1 Description of Files

File Name	Purpose	Description of File
App.tsx	Frontend	Frontend application code
App.css		Frontend aesthetics
trades_preprocessing.ipynb	Data Extraction	Code to preprocess trading data
shareholder_letters_preprocessing.ipynb		Code to preprocess shareholder letters
data_preprocessing.ipynb		Code to clean Warren Buffett QnA
buffett_qna.ipynb		Data extraction of Warren Buffett QnA
finetuning.ipynb	LLM Fine-Tuning	Code for fine-tuning of LLM
ft_evaluation.ipynb		Code for evaluation of fine-tuned model
agent.py	RAG Model	Code for function to call agent
prompt_engineering.py		Code to generate evaluation prompt
rag_pipeline.py		Code for RAG pipeline
reranker.py		Code for reranking of documents
retrieval.py		Code for retrieval functions
text_splitter.py		Code for text splitting
main.py		Code to run RAG model
rag.py		Code to run RAG pipeline
data_utils.py		Utility functions
vectorstore.py		Code for vector stores

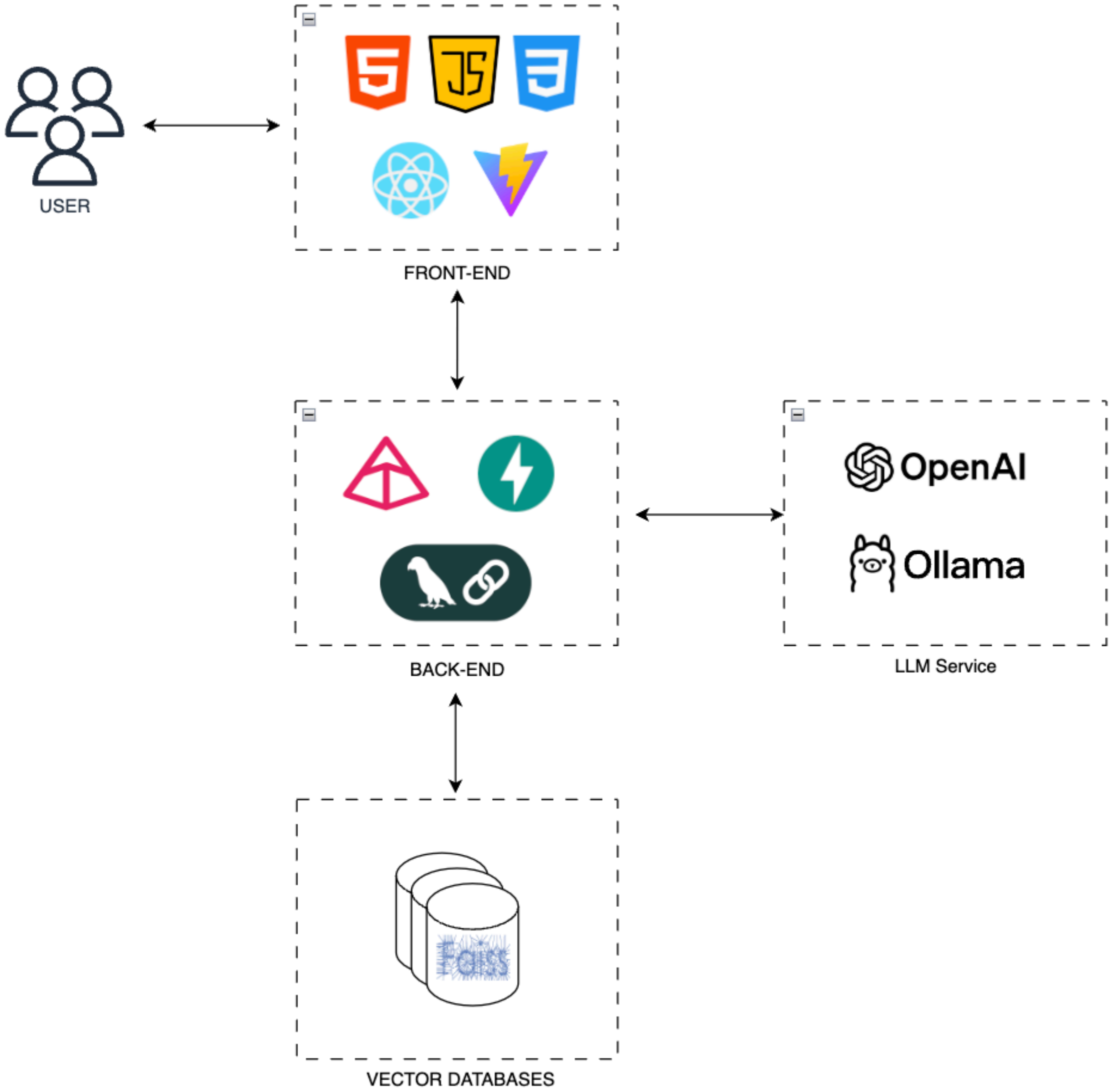

Figure A: Full-stack architecture diagram

10 References

Abraham, A., Kapronczay, M., & Turner, R. (2024, March 18). *Optimizing RAG with Hybrid Search & Reranking | VectorHub by Superlinked*. https://superlinked.com/vectorhub/articles/optimizing-rag-with-hybrid-search-reranking

BBC. (n.d.). *BBC Business | Economy, Tech, AI, Work, Personal Finance, Market news*. Retrieved March 20, 2025, from https://www.bbc.com/business

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks* (arXiv:2005.11401). arXiv. https://doi.org/10.48550/arXiv.2005.11401

Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J. M., Tworek, J., Yuan, Q., Tezak, N., Kim, J. W., Hallacy, C., Heidecke, J., Shyam, P., Power, B., Nekoul, T. E., Sastry, G., Krueger, G., Schnurr, D., Such, F. P., Hsu, K., … Weng, L. (2022). *Text and Code Embeddings by Contrastive Pre-Training* (arXiv:2201.10005). arXiv. https://doi.org/10.48550/arXiv.2201.10005

Prisa. (2017, April 3). *EL PAÍS Economía: The best of business and financial news in Spanish | Prisa*. https://www.prisa.com/en/noticias/notas-de-prensa/el-pais-economia-the-best-of-business-and-financial-news-in-spanish

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-Training*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*.

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* (arXiv:1908.10084). arXiv. https://doi.org/10.48550/arXiv.1908.10084

Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and

Beyond. *Foundations and Trends® in Information Retrieval*, *3*(4), 333–389. https://doi.org/10.1561/1500000019

Sinha, K., Jia, R., Hupkes, D., Pineau, J., Williams, A., & Kiela, D. (2021). *Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little* (arXiv:2104.06644). arXiv. https://doi.org/10.48550/arXiv.2104.06644

Tyagi, H. (2025, February 14). *How US is dominating Global Stock Market*. INDmoney. https://www.indmoney.com/blog/us-stocks/us-commands-half-of-global-stock-market-apple-nvidia-tesla-and-other-top-10-giants-hold-32-of-it

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Yasar, K. (2024, May). *What are Masked Language Models (MLMs)? | Definition from TechTarget*. Search Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/masked-language-models-MLMs