Annual Public Report Analysis ChatBot Using LLM

Mrs. Sujata Khedkar[#1], Purtee Santosh Mahajan[#2], Tasmiya Sarfaraz Khan[#3],

Srushti Satish Sambare[#4], Ketaki Dhananjay Nalawade[#5]

[#]*Computer Engineering Department, Vivekanand Education Society's Institute of Technology, Chembur, Mumbai - 400071, India*

[1]sujata.khedkar@ves.ac.in

[2]2021.purtee.mahajan@ves.ac.in

[3]2021.tasmiya.khan@ves.ac.in

[4]2021.srushti.sambare@ves.ac.in

[5]2021.ketaki.nalawade@ves.ac.in

***Abstract*— In today's data-driven world, interpretation of public annual reports might become a very time-consuming process for the stakeholders. This paper introduces the concept called an Annual Public Report Analysis Chatbot using LLM, through which automatically going through reports and generating intuitive answers through AI will be possible. The chatbot is designed to help stakeholders extract key insights from annual public reports, making the information easier to understand and more actionable.Utilizing AI capabilities, this system does not merely provide correct responses to questions it receives from the user but also provides comparative analysis and data visualization options to further help the decision-making process.The intent of this project is to change the way that stakeholders interact with complex reports making the process associated with analyzing those reports streamlined and empowering such decision-makers to go data-driven. The chatbot processes public reports by extracting and organizing the information into a structured knowledge base, enabling efficient retrieval of relevant insights. It helps users uncover key information from annual reports, making the content clearer and more actionable. By leveraging AI, the system provides accurate answers to user queries and offers comparative analysis and data visualization, supporting informed decision-making. One of its standout features is text-to-speech functionality, which delivers responses audibly, enhancing accessibility and inclusivity.**

Keywords— Large Language Models (LLMs), Annual Public Report, AI Chatbot, Data Visualization, Text-to-Speech, Corporate Data Analysis, Semantic Search, Stakeholder Engagement.

I. Introduction

The annual report is the major source of information to investors. Investors will read the annual report to analyze the performance of the company.The annual report is composed of the director's report, auditors report, financial statements, and notes to the accounts. Financial statements are in quantitative form while the director's report, notes to the accounts and auditors report are in qualitative form. Investors, in general, will read the annual report to make the investments. Investors will invest in the company if the company is performing well [10], then there will be positive narration in the annual report else there will be negative narration [1].With the rise of the Internet and the information age, an increasing amount of data is being uncovered and consumed by individuals. As a key vehicle for conveying corporate information, financial statements have grown progressively longer. This expansion is driven by the need to meet the diverse demands of information users, while also adhering to the standards of accounting information quality. Longer reports mean more information. While the increased information has both benefits and problems for the users. [2] . Also predicting bankruptcy is important before investing because bankruptcy status is a clear indicator of fiscal health [6].

However, manually analyzing these extensive documents can be time-consuming and prone to human error.Due to the sheer volume and complexity of these documents, they are highly labor-intensive and susceptible to human error. Recent advances in Natural Language Processing (NLP), particularly in Large Language Models (LLMs), have significantly enhanced the performance of text analysis systems by machines. Our chatbot leverages the capabilities of such models in efficiently and intelligently extracting insights from annual reports. The rapid advancement of Large Language Models (LLMs) has revolutionized the extraction and interpretation of financial data, particularly in quantifying market sentiment derived from sources such as corporate disclosures, financial news, and social media. These insights play a crucial role in influencing market movements and guiding investment decisions. LLMs have also demonstrated significant potential in the domain of Financial Time Series Analysis [8], where they contribute to forecasting trends, detecting anomalies, and classifying financial data[9]. Despite ongoing debate regarding their efficacy in this area, LLMs excel in capturing complex temporal dependencies within financial datasets through their advanced deep learning architectures. One of their most notable capabilities is

reasoning, allowing them to not only analyze data but also simulate cognitive processes similar to human decision-making.[7] This reasoning extends into Financial Planning, where LLMs assist in generating investment strategies and supporting decisions by synthesizing vast amounts of data. Furthermore, their application in Agent-based Modeling enhances the simulation of financial ecosystems, enabling the study of market behaviors and economic activities through dynamic interactions between agents and their environments[3].

This paper aims to design an annual public report chatbot that uses large language models to analyze unstructured corporate data. Users can upload or search for reports in this system, which are processed in real time to produce answers, comparative analyses, and visual data insights. This will really aid the decision-making process by ensuring a very efficient and user-friendly understanding of corporate reports.

II. Related work

The researchers discovered that the language used in annual reports is a crucial indicator of a company's financial well-being and performance [1]. They found that a positive tone is associated with profitable companies and has an impact on investor behavior. Investors are more likely to invest in companies with a positive tone and avoid those with a negative tone. The tone was measured using the Loughran-McDonald sentiment word lists, providing a structured method for evaluating qualitative financial disclosures. The study also identified a connection between tone and stock price movements, indicating that the sentiment expressed in these reports influences market perceptions and investment decisions, particularly in initial public offerings. While tone is important, the study was focused on Indian companies and suggests that incorporating quantitative data and broader contexts could improve the analysis.

A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges [3]: This paper delves into the application of Large Language Models (LLMs) in examining annual reports, with a focus on various important aspects. LLMs like BioFinBERT are utilized for gauging sentiment to forecast stock price movements based on the emotions conveyed in regulatory filings and legal documents. They also shine in extracting information, enabling stakeholders to swiftly condense crucial data for well-informed decision-making. Additionally, LLMs assist in flagging anomalies by pinpointing inconsistencies within financial statements, ensuring precision in reporting. Their capabilities extend to numerical analysis, allowing for the comprehension of intricate financial data through multi-step calculations. However, the analysis encounters challenges such as high computational expenses, the potential for misinterpreting financial terminology, susceptibility to adversarial attacks, and the presence of excessive information that might obscure crucial insights.

Automatic Analysis of Annual Financial Reports: A Case Study [4]: The analysis of annual reports involved the use of various methodologies and focused on examining 10-K reports from Ford Motor Company, General Motors Company, Google (Alphabet Inc.), and Yahoo! Inc. in the automotive and IT industries over a decade (2005-2014). Researchers specifically extracted non-financial sections from the reports, particularly Part I and Items 7 and 7A from Part II, to capture management opinions on past and future performance. They conducted linguistic analysis by considering document length, sentiment using a sentiment dictionary, the prevalence of trust and doubt keywords, and various discursive features such as personal versus impersonal pronoun ratios. Differential content analysis involved using TF-IDF weighting to identify characteristic terms for each year, while correlation analysis examined connections between linguistic characteristics and financial performance. Nevertheless, the research encountered constraints, such as a limited sample size that might not accurately reflect wider market patterns, a restricted time period that might fail to capture notable changes in reporting methods, difficulties in extracting data due to inconsistent formats, and the subjective nature of linguistic analysis. Furthermore, the conclusions might not be applicable to other types of reports or regulatory landscapes beyond the U.S., indicating the necessity for additional research using larger and more diverse samples to affirm and build upon these findings.

Narrative analysis of annual reports: A study of communication efficiency [5] : The researchers analyzed the content of annual reports, specifically focusing on the Management Discussion and Analysis (MD&A) sections. They used the Flesch Reading Ease formula to assess the readability of the texts. This method allowed them to measure the language complexity in the reports and track changes in readability over the years under review, especially during the global recession from 2008 to 2012. The objective was to evaluate how external factors, such as economic downturns, affected the clarity and openness of corporate communications.

InvestLM [12] is especially adept at analyzing annual reports because it was trained on texts that are primarily financial in nature and can come up with contextually relevant answers. Given its training on a finely curated financial dataset, InvestLM understands complex financial language and concepts found within annual reports; thus, it may obtain essential information from financial statements, management discussions, or risk factors. Its performance has been rated just as excellent in expert evaluations as with models like GPT-3.5 and GPT-4, making it truly useful in investment decision-making. InvestLM really shines in setting up risks and opportunities in the context of a company's financial health and management expectations. Automation of large volumes of text saves much time for financial professionals to do more by lending itself well beyond annual reports to tasks such as sentiment analysis and document classification.

III. Input Sources

The system deals with two main input sources, explained as follows:

A. *Direct Report Upload*

Our chatbot facilitates users directly to upload the annual public reports in PDF format. The system supports the upload of multiple files, enabling detailed comparative analysis across different reports. The reports include detailed financial statements, directors' reports, the auditors' reports, and other detailed information on a firm's performance during the fiscal year. Stakeholders can quickly analyze specific reports for insights without having to search through them manually through the direct uploads.

B. *Company Name and Year Search*

In addition to direct uploads, users can search for annual public reports by entering the company's name and the desired year. The system fetches all the reports available on the internet using API ,allowing users to download them and then upload the report PDF to the chatbot. The feature aims at making data collection easy, thus providing ease in accessing historical reports and comparing across various periods.

IV. Proposed solution

The system would have detailed and user-friendly analysis of public annual reports with the possibility of diverse data entry means. The user could upload PDF files with annual public reports or apply the search functionality of the chatbot to find [11] and download relevant reports regarding the specified company and year. Once the reports are uploaded, the text content will be extracted and broken down into more manageable pieces and interpreted in vector form-through embeddings-and stored efficiently inside a vector database, for example, Faiss. As soon as a user poses a question, the chatbot processes the query by converting it into embeddings; the app then conducts a semantic search of the stored data to decide what's the most relevant piece of text regarding the question asked. The chatbot then ranks the information from the selected source according to relevance and contextual importance.

Using LLM, the chatbot produces context-sensitive precise answers and directly answers questions of users regarding fact type, number, or type of reasoning questions. To provide better understanding in complex data, the chatbot provides data visualization in graphical trends and comparisons for those questions that contain numerical information. Furthermore, it also provides for comparative analysis when there are multiple reports uploaded, showing the comparison of differences and similarities that will give the user an overall view of the company's performance. There is also a FAQ available to its chatbot categorized into strategic goals, financial performance, operational highlights, risk mitigation, and initiatives toward sustainability; from there, one can easily find an answer to the most commonly asked question.

The inclusion of a text-to-speech function further enhances user accessibility of the chatbot, and the responses generated can now be received in listening format. All this puts into action sophisticated AI techniques that will perform multi-dimensional analysis on annual public reports, thus empowering stakeholders with a tool that does more than facilitate easy interpretation of data but also makes and executes decisions based on data-driven inputs with clarity and precision.

V. Methodology

Step 1: Uploading Files

The users can directly upload annual public report PDFs (single or multiple) into the chatbot.Additionally the user can search for a specific company's annual report by entering the company's name and desired year. The chatbot fetches relevant reports from the internet, which the user can download and then upload to the chatbot for analysis. The user uploads PDF files using Streamlit's file upload component. After selecting the file, the user clicks on the 'Process PDF' button to initiate processing.

Step 2: Processing the Files

To initiate the handling of a PDF file, the process_pdf() function is called. It makes use of the PdfReader class of the PyPDF2 library for the purpose of text extraction on each page of the PDF. The pages' extracted text is joined together to create one complete string that summarizes the contents of the document. The get_text_chunks() function is called here, which takes the created string and transforms it into smaller parts using LangChain's RecursiveCharacterTextSplitter. It uses RecusiveCharacterTextSplitter and splits the text in parts of 5000 characters with a 1000 character overlap between parts. Then, the get_vector_store() function is called, in which these text snippets are encoded as vector embeddings using Google Generative AI embeddings. The produced embeddings are stored in a vector database, built with the help of Weaviate Library for fast semantic searches.

Step 3: Building the Conversational Chain

To establish a conversational chain, the method get_conversational_chain() is invoked. This method employs the LangChain framework for building the conversational structure. We load the Gemini-pro model from the library which is called ChatGoogleGenerativeAI and assign it as the language model (LLM) intended for the generation of responses. There is a memory component in the form of ConversationBufferMemory which stores the conversation history that enables the chatbot to remember the flow of the dialogue. The dialog chain

is set in such a way that the already created vector store is used as a retriever, making it possible to search for any relevant text chunk in accordance to the user's queries. The LLM is connected to the retriever and the prompt template by the ConversationalRetrievalChain.from_llm() method, thus forming a chain that makes use of retrieved information to generate a response.

Step 4: Handling User Queries

After processing and indexing the PDF document, the user can type his or her inquiry into the input area. The user's request is responded to by a conversational chain that is maintained in Streamlit's session state and is provided with the question as input. The request is then vectorized and the resulting vector is compared to all the existing vectors in the vectorized database in order to locate the most relevant text chunks. The system retrieves the most analogous vectors on the basis of their meanings and then invokes the LLM to assemble an answer. The history of dialogue is freshed with the user's query and the system's reply and is shown to the user.

Step 5: Generating FAQs

In order to create FAQs, certain Questions and categories are premade. The geography of the responses to the FAQ section is created in a manner similar to that of the user queries. The conversational outlay fetches and answers the questions using information stored in the vector database

Step 6: Displaying the Response

The application inspects the conversation logs that are maintained in Streamlit's session state and iteratively scans through the messages one by one. The formatting positions the user messages and AI messages alternating one below the other for the clear presentation of the dialogue. Further, these tabular results are known because of certain characters being present along with the AI's response (for instance, | and ---). This tabulated information is extracted, sanitized, and presented in a neat form. If there is no tabulated data, then the result is given to the end user as normal text. Additionally, a text-to-speech feature is available for users who prefer listening to the responses, ensuring a user-friendly experience.

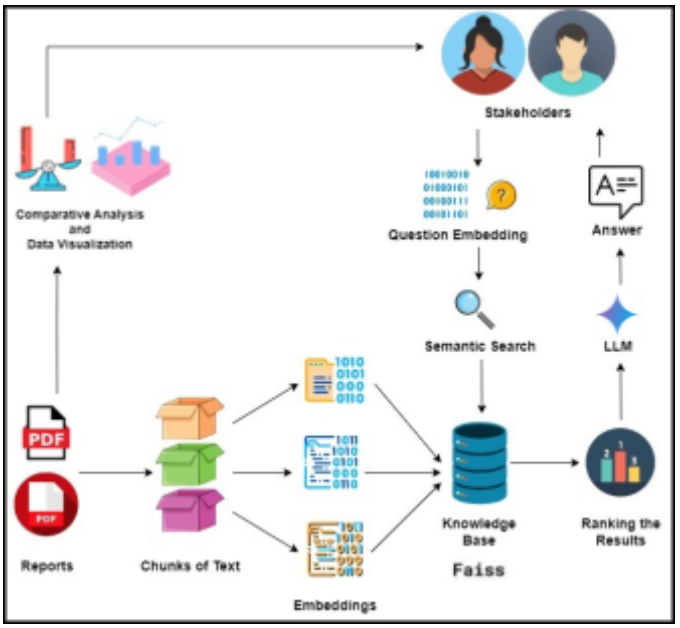


Fig. 8 System Architecture based

VI. Experiments and Results

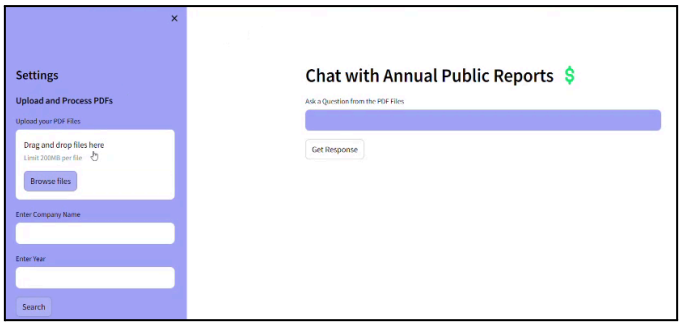


Fig 1. ChatBot Home Page

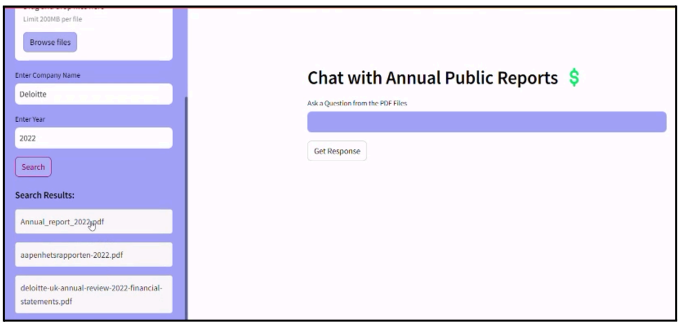


Fig 2. Results after entering the name of any company and particular year (Deloitte and 2022 in this case)

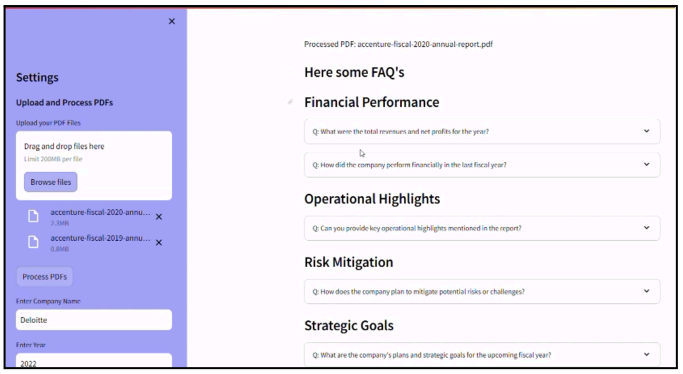


Fig 3. FAQs generated category wise with answers taken from the 2 uploaded PDFs

Fig. 4 Few answers retrieved from the two input documents in FAQ section

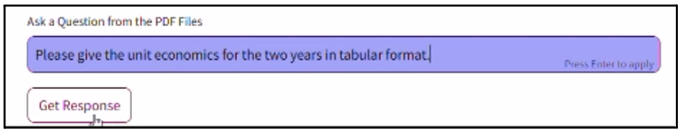
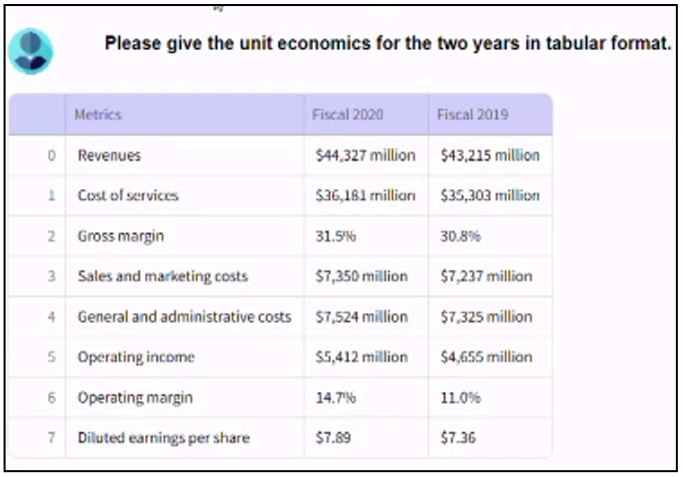


Fig. 5. Question entered by the user



	Metrics	Fiscal 2020	Fiscal 2019
0	Revenues	$44,327 million	$43,215 million
1	Cost of services	$36,181 million	$35,303 million
2	Gross margin	31.5%	30.8%
3	Sales and marketing costs	$7,350 million	$7,237 million
4	General and administrative costs	$7,524 million	$7,325 million
5	Operating income	$5,412 million	$4,655 million
6	Operating margin	14.7%	11.0%
7	Diluted earnings per share	$7.89	$7.36

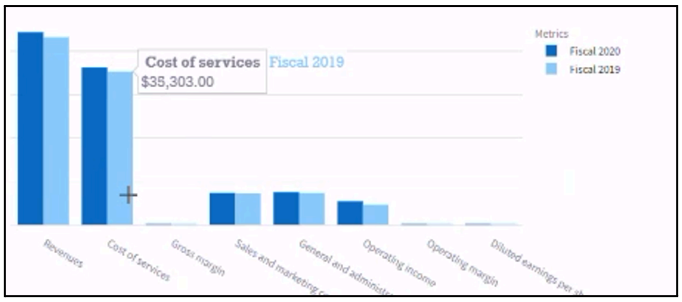


Fig 6. Answer to the question given by the user

VII. Technologies Used

A. *Google Gemini API*

Gemini is a family of highly capable multimodal models developed at Google [13]. The system is powered by the Gemini-1.5-flash model.

B. *FAISS (Facebook AI Similarity Search)*

Faiss is a library for ANNS. The core library is a collection of source files written in standard C++ without dependencies. Faiss is used in many configurations. Hundreds of vector search applications rely on it, both within Meta [14] and externally. The proposed system deals with a large amount of textual data. Hence, FAISS is always preferred for scalability as it avoids a drop in performance. The purpose of this utilization is to perform similarity search on the vectorized representations of the documents and for a quick and easy retrieval.

C. *Streamlit*

Streamlit is an open source Python framework that allows data scientists and AI/ML engineers to write dynamic data applications in a few lines of code [15]. In this framework, you can build interactive visualization plots, models, and dashboards without worrying about the underlying web framework or the deployment infrastructure used in the backend [16]. Though it is quite easy to build and deploy a Streamlit app, it is not scalable. So for large scale applications, frameworks like Flask, which is a Python-based framework, would be preferred.

D. *Langchain*

Langchain is an in-house built Large Language Model for any organization [17]. LLMs are being used very rapidly as they can effectively execute a vast range of tasks, some of which include essay composition, code writing, explanation, and debugging [18]. They take in a text string (prompt) and return a text string [18].

VIII. FUTURE WORK

The current implementation of the risk assessment system has considerable utility at accelerating evaluation of the financial documents and risk reports' compilation. However, there are multiple areas where the system can be extended and improved to better meet the needs of organizations and enhance its capabilities:

- Customizable Report Formats: Another important feature which should be anticipated in the future is the possibility of an implementation to define the layout and structure of the risk reports produced. It has become a norm for different companies to have preferred style of reporting, including sections preferred in the report, preferred kind of diagrams or tables, or industry-specific language. It was proposed that, through a reporting feature that can be customized, the user would be able to prepare the report according to internal rules or other requirements.
- Linking of documents to more than one document type: The present system of analyzing and interpreting the PDF-file format of financial reports may be expanded in subsequent versions for other formats, such as Excel tables, Word documents or even access to database information. This would enhance flexibility of the system in handling accounting information from various data feeds from different sources, and would also ensure compatibility to various documentation processes by companies.

IX. CONCLUSION

This research outlines a simple way of automating financial risk assessment through Gemini 1.5 Flash coupled with enhanced user interface. The system is extremely effective to scan financial documents, flag risks and offer actionable intelligence. Automating risk analysis therefore enables organizations to take quicker and better decisions. Additional additions such as report features, or industry-specific models will enhance the system's flexibility making the system relevant for companies as they grapple with sophisticated risk levels.

REFERENCES

[1] P. K. Aithal, D. A. U. and G. M., "Analyzing Tone of the Annual Report - An Indian Context," 2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), THIRUVANANTHAPURAM, India, 2022, pp. 536-542, doi: 10.1109/SPICES52834.2022.9774247.
keywords: {Instruments;Signal processing algorithms;Companies;Signal processing;SPICE;Stock markets;Information technology;Portfolio Management;Tone;Initial Public Offerings;Parallel Algorithms;Message Passing Interface},

[2] Yuyan, Guo. "An analysis of the growing length of the annual reports." The Frontiers of Society, Science and Technology 5, no. 2 (2023).

[3] Nie, Yuqi, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen, and Stefan Zohren. "A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges." arXiv preprint arXiv:2406.11903 (2024).

[4] Smailović, Jasmina & Žnidaršič, Martin & Valentinčič, Aljoša & Loncarski, Igor & Pahor, Marko & Martins, Pedro & Pollak, Senja. (2018). Automatic Analysis of Annual Financial Reports: A Case Study. Computación y Sistemas. 21. 10.13053/cys-21-4-2863.

[5] Srinivasan, Padmini, and Ana Cristina Marques. "Narrative analysis of annual reports: A study of communication efficiency." (2017).

[6] Mai, Feng, Shaonan Tian, Chihoon Lee, and Ling Ma. "Deep learning models for bankruptcy prediction using textual disclosures." European journal of operational research 274, no. 2 (2019): 743-758.

[7] Hadi, Muhammad Usman, Qasem Al Tashi, Abbas Shah, Rizwan Qureshi, Amgad Muneer, Muhammad Irfan, Anas Zafar et al. "Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects." Authorea Preprints (2024).

[8] Yu, Xinli, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. "Temporal Data Meets LLM--Explainable Financial Time Series Forecasting." arXiv preprint arXiv:2306.11025 (2023).

[9] Lee, Jean, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. "A survey of large language models in finance (finllms)." arXiv preprint arXiv:2402.02315 (2024).

[10] Olayinka, Aminu Abdulrahim. "Financial statement analysis as a tool for investment decisions and assessment of companies' performance." International Journal of Financial, Accounting, and Management 4, no. 1 (2022): 49-66.

[11] https://newsapi.org/docs

[12] Yang, Yi, Yixuan Tang, and Kar Yan Tam. "Investlm: A large language model for investment using financial domain instruction tuning." arXiv preprint arXiv:2309.13064 (2023).

[13] G. Team *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv.org*, Dec. 19, 2023. https://arxiv.org/abs/2312.11805

[14] M. Douze et al., "The Faiss library," arXiv.org, Jan. 16, 2024. https://arxiv.org/abs/2401.08281

[15] "Streamlit Docs." https://docs.streamlit.io/

[16] Sreeram a, Adith & Sai, Jithendra. (2023). An Effective Query System Using LLMs and LangChain. International Journal of Engineering and Technical Research. 12.

[17] K. Pandya and M. Holia, "Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations," *arXiv.org*, Oct. 09, 2023. https://arxiv.org/abs/2310.05421

[18] Topsakal, Oguzhan & Akinci, T. Cetin. (2023). Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. International Conference on Applied Engineering and Natural Sciences. 1. 1050-1056. 10.59287/icaens.1127.