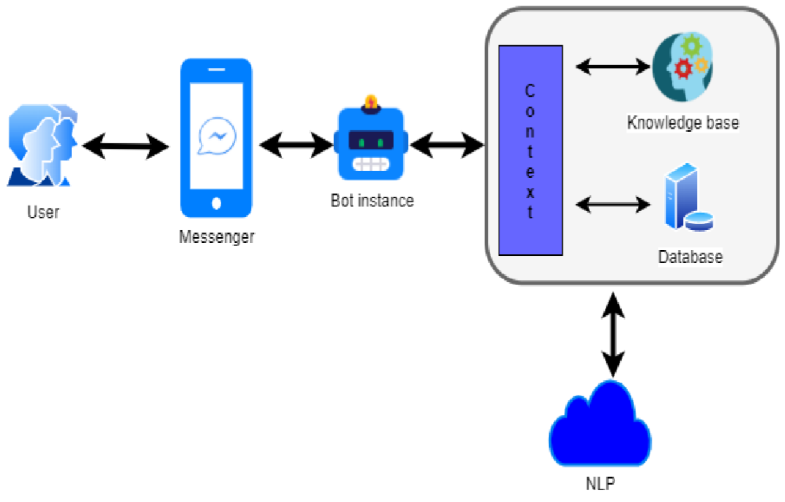
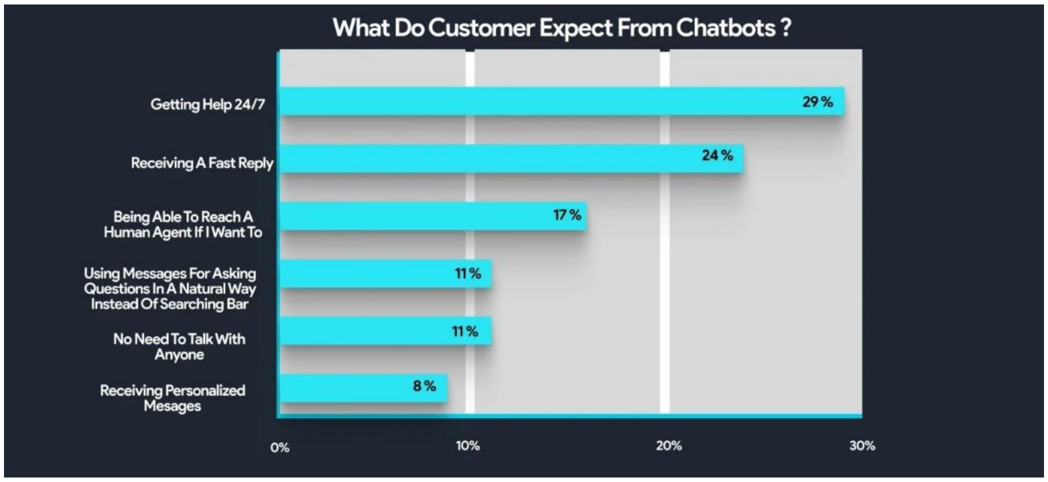
Project Title	Bank Customer ChatBot using RAG
Group No.	07
Names of Group Member	Soleman Hossain, Sihab Mahmud, Suhana Islam, Nafis Anzum.
Email addresses	soleman.hossain@northsouth.edu, sihab.mahmud@northsouth.edu, suhana.islam@northsouth.edu , nafis.anzum@northsouth.edu

Introduction

Providing practical, individualized, and 24/7 customer service is a significant problem for the banking industry. The increasing volume of inquiries frequently overwhelms traditional customer support channels like call centres and FAQs, resulting in lengthy wait times and subpar user experiences. Intelligent, automated systems that can instantly comprehend and react to consumer inquiries are desperately needed to address this. This project suggests using Retrieval-Augmented Generation (RAG). This state-of-the-art AI technique blends the advantages of natural language generation and information retrieval to create a bank customer chatbot. Enhancing customer satisfaction, cutting operational expenses, and offering a scalable solution for managing a range of banking inquiries are the driving forces behind this project.



Providing more incredible customer-focused services is the primary application or advantage of employing chatbots in the banking sector. Personal banking chatbots can be created to understand users' needs and offer tailored recommendations based on past transactions and buying patterns. For example, a chatbot can recommend a credit card with a lower interest rate to a customer who regularly has debt on their current credit card. It may also predict what products the customer wants based on consumer data or buying habits. (Prasannan, 2023)



Current technologies, such as rule-based chatbots and simple AI-driven conversational agents limit the ability to handle sophisticated, context-specific queries. While traditional AI chatbots frequently produce generic or incorrect responses because they rely on pre-trained models without real-time data integration, rule-based systems are inflexible and unable to adjust to complex client needs (Tom B. Brown, 2020). Many chatbots, for example, are unable to obtain current account information or offer tailored financial advise (Ghandour, 2021). These gaps highlight the need for a more sophisticated system that can dynamically acquire pertinent data and produce context-aware replies. The suggested RAG-based chatbot overcomes these constraints by utilizing real-time data retrieval and sophisticated language models and providing a more precise, tailored, and practical customer support experience (Miehleketo Mathebula, 2024).

To close the gap between conventional chatbots and contemporary customer expectations, we used Retrieval-Augmented Generation (RAG) to create a Bank Customer Chatbot. Our method combines a generative model to create contextually relevant replies with a retrieval mechanism to retrieve real-time financial data. This project's main achievements are the deployment of an intuitive chatbot interface, the development of a collection of banking-specific queries and answers, and the application of a strong RAG framework designed for the banking industry. Our chatbot sets a new benchmark for AI-driven customer care in the banking sector by fusing retrieval and generation capabilities, increasing response accuracy and improving the overall customer experience.

Background and Related Work

Chatbots have become a vital tool in the banking sector to enhance customer interactions, streamline services, and reduce operational costs. Traditional chatbots were mainly AI-driven, with limited flexibility in handling dynamic queries. Therefore, this couldn’t be answered beyond their programmed responses. AI-driven chatbots based on intent recognition and Natural Language Processing (NLP) have improved user experience but often suffer from poor contextual understanding and outdated information. These limitations were overcome by more advanced AI techniques like Retrieval-Augmented Generation (RAG). RAG helped to improve the overall responsiveness and contextual accuracy.

Existing Solutions:
So far, several chatbots have been implemented in the banking sectors, such as:
1. Rule-Based Chatbots: These chatbots operate on decision trees & keyword-matching. While they ensure consistency, they lack context-aware responses.
2. Machine Learning-based chatbots: AI-driven models use NLP & DL. It improved the ability to process queries, but they depend heavily on static trained data.
3. Hybrid Chatbots: Some banks integrate their chatbots with backend API, but they can’t answer broader financial queries dynamically.

Rag Based Chatbot:
The retrieval augmented generation (RAG) framework boosted the chatbot capabilities by integrating information retrieval with generative responses. Unlike the previous models, it can retrieve data from the knowledge base & real-time banking API. By leveraging a dual system of retrieval and generation, it can
· Generate responses based on user history.
· Access live financial database.
· Improved Accuracy and response.

This framework ensures accurate data while significantly minimizing misinformation. It improves customer engagement and also solves real-time banking needs.

Problem Identification

Banks always try to ensure customers have access to relevant and recent data regarding the bank services. However, traditional customer support methods, such as call centres, FAQ pages and rule-based chatbots, often fail to deliver efficient, real-time responses. Furthermore, most banks require human intervention for basic queries, which leads to inefficiency and waiting times.

To address these problems, Bank Customer Chatbots are developed to ensure accurate and instant responses for bank-related queries, providing a smooth user experience. So the main task is to implement a scalable chatbot that can handle various bank pdf-related queries and generate an instant response to the user. However, most AI-driven chatbots today rely on pre-trained models without real-time data retrieval, leading to generic, outdated and incorrect responses.

Challenges in Existing Systems:

- **i)** **Limited Context:** Most chatbots rely on small datasets and fail to retrieve specific answers using dynamic queries.
- **ii)** **Inability to handle PDF-based queries:** Many traditional chatbots cannot process PDF-based documents.
- **iii)** **Slow and Inefficient Retrieval:** Many chatbots use keyword-based searches rather than vector embeddings for similarity matching, leading to inaccurate responses.
- **iv)** **Lack of Automation Testing:** The automation testing method is preferable and efficient, whereas traditional systems that use manual testing can be very time-consuming.

Differences between Traditional Chatbots and RAG-Based Chatbots are represented in a tabular form given below:

Feature	**Traditional Chatbot**	**RAG-based Chatbot**
1.Response time	Predefined Responses	Dynamic, AI-generated responses
2. Context Awareness	Limited	High
3.PDF Document Handling	Not supported	Fully supported
4. Personalization	Limited	High
5. Accuracy	Sometimes incorrect	High
6. Learning Ability	Static	Improves over time

Progress of 1st Week:
As per the weekly progress of the 1st week, in-depth research was done on RAG(Retrieval Augmented Generation) and LLM(Large Language Model). A draft roadmap was prepared for the implementation of the chatbot, which included :
-Understanding Rag and setting up the environment
-Processing the pdf ly, loading and chunking
-embedding and storing data
-Use Streamlit to build a Chatbot interface
-Test the Chatbot with a number of test sets, preferably 1000

The pdf has been loaded, divided into chunks and embedded into a vector database. The chatbot interface has been developed using Streamlit and manual testing has been done so far with at least 100-150 queries. In the upcoming weeks, the chatbot is expected to do automated tesing.

Solution Methodology

To address the above challenges, the following methodologies will be implemented:

i) **Data Preprocessing and Embedding:** The chatbot will preprocess a 42-page PDF as a source of knowledge. The pdf is first loaded and text is extracted using PyMyPDF. Text is divided into smaller chunks and converted into vector embeddings using the vector database.

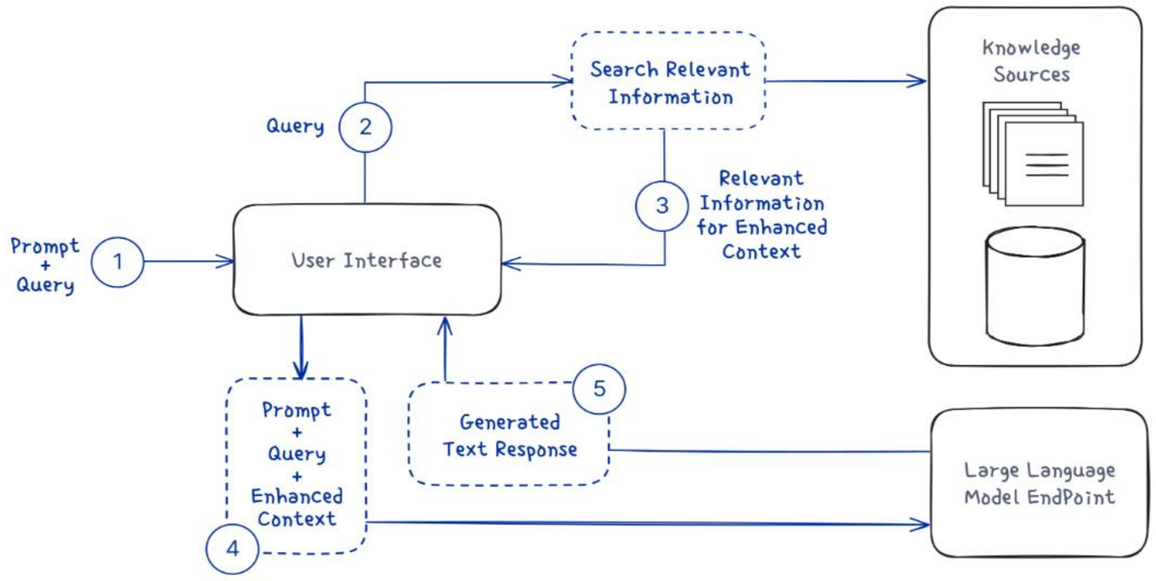

Prompt + Query
1
User Interface
Query
2
Search Relevant Information
Knowledge Sources
3
Relevant Information for Enhanced Context
Prompt + Query + Enhanced Context
4
Generated Text Response
5
Large Language Model EndPoint

Figure: RAG-based chatbot system architecture diagram

ii) **Vector-database Implementation: ChromaDB or FAISS is preferably used as** a source for storing the vector embeddings for the chatbot. The chosen database stores the embeddings which allows efficient similarity searches. After the user passes a query into the chatbot, it is converted to an embedding and matched with the most relevant chunk in ChromaDB/FAISS.

iii) **Retrieval-Augmented Generation(RAG):** The chatbot will retrieve relevant chunks of information using RAG and LLM(Ollama' Llama 2 model) using the similarity search approach as RAG typically does. This will ensure the chatbot generates accurate responses to user queries. It also displays the PDF reference section in its response.

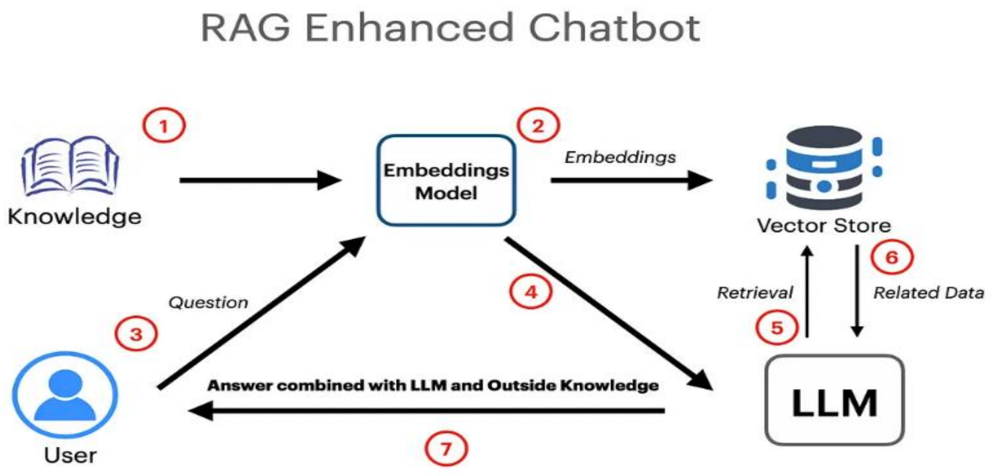
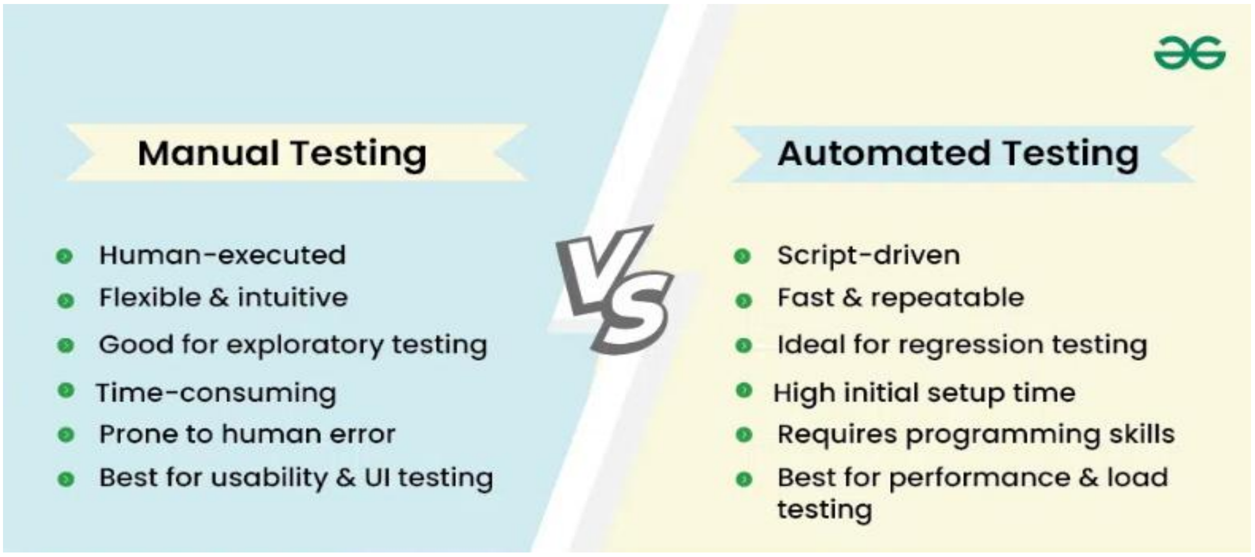


Figure: Query Processing Workflow –From User Input to Response Generation

iv) **UI Developments with Streamlit:** The chatbot interface is developed using Streamlit to provide smooth user interaction with the chatbot. A standard look of a chabot is maintained. The features include live chat interaction with saved chat history, response time displayed on the interface, and a button to clear chat history if the user wishes to do so.

v) **Testing and Evaluation:** After the development of the chatbot, automated testing is done using a test set of at least 1000 queries. Automated testing is done to reduce the errors of manual testing. The chatbot's performance can be evaluated by analysing the response time and how accurately relevant information is retrieved.

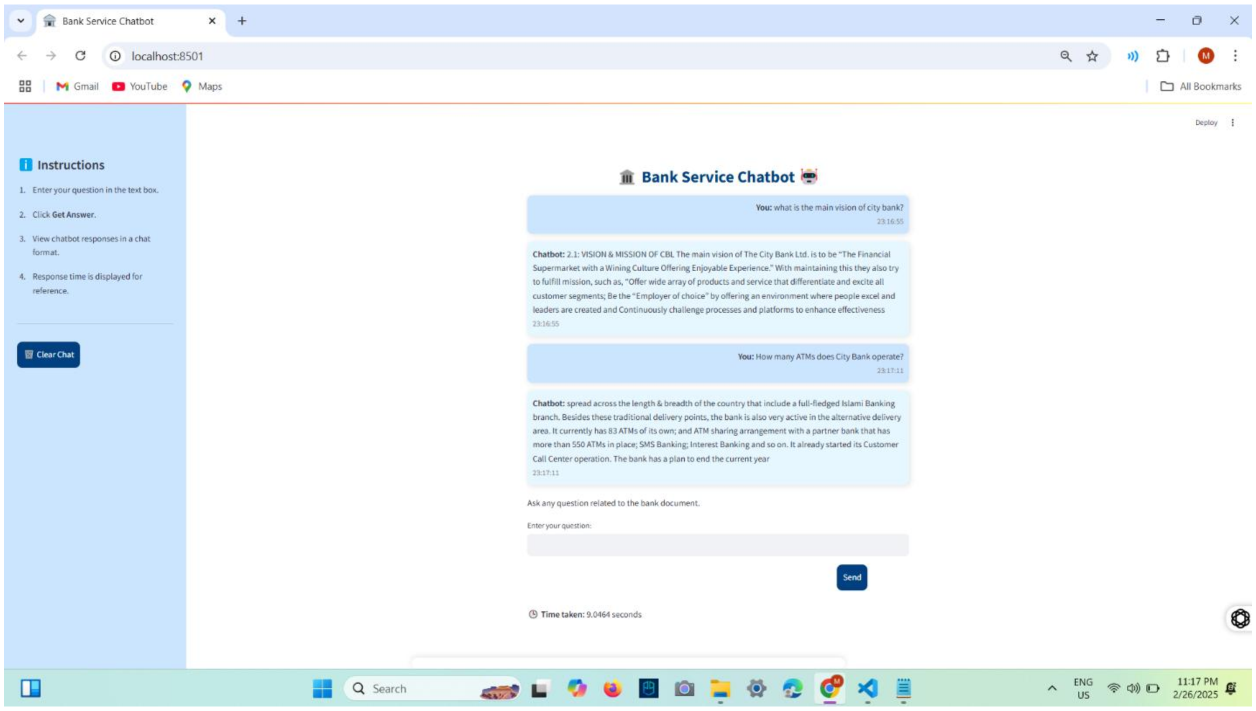


Manual Testing	Automated Testing
Human-executed	Script-driven
Flexible & intuitive	Fast & repeatable
Good for exploratory testing	Ideal for regression testing
Time-consuming	High initial setup time
Prone to human error	Requires programming skills
Best for usability & UI testing	Best for performance & load testing

Figure: Manual Testing VS Automated Testing

Progress Made So Far:
So far, the chatbot interface has been built and some manual testing has been done with various questions. The goal was to reduce the response time as much as possible for every query passed into the chatbot.

Screenshot:



Instructions

1. Enter your question in the text box.
2. Click **Get Answer**.
3. View chatbot responses in a chat format.
4. Response time is displayed for reference.

Bank Service Chatbot

You: what is the main vision of city bank?
23:16:55

Chatbot: 2.1: VISION & MISSION OF CBL The main vision of The City Bank Ltd. is to be "The Financial Supermarket with a Wining Culture Offering Enjoyable Experience." With maintaining this they also try to fulfill mission, such as, "Offer wide array of products and service that differentiate and excite all customer segments; Be the "Employer of choice" by offering an environment where people excel and leaders are created and Continuously challenge processes and platforms to enhance effectiveness
23:16:55

You: How many ATMs does City Bank operate?
23:17:11

Chatbot: spread across the length & breadth of the country that include a full-fledged Islami Banking branch. Besides these traditional delivery points, the bank is also very active in the alternative delivery area. It currently has 83 ATMs of its own; and ATM sharing arrangement with a partner bank that has more than 550 ATMs in place; SMS Banking; Interest Banking and so on. It already started its Customer Call Center operation. The bank has a plan to end the current year
23:17:11

Ask any question related to the bank document.

Enter your question:

Time taken: 9.0464 seconds

ENG US 11:17 PM 2/26/2025

Expected Timeline

The development of the RAG-based banking chatbot is divided into structured milestones over 4 months.

PHASE	DURATION	MILESTONE	DELIVERABLES
REQUIREMENT ANALYSIS	Week 1	Define system requirements, data sources, and APIs	System architecture, dataset selection
DATA COLLECTION & PREPROCESSING	Week 2-3	Gather and clean banking-related datasets	Preprocessed dataset, API integration plan
RAG MODEL DEVELOPMENT	Week 4-5	Implement retrieval & generation pipeline	Initial RAG model prototype
UI DEVELOPMENT	Week 6	Develop chatbot UI and backend connectivity	Functional chatbot interface
CHATBOT INTEGRATION	Week 7	Integrating the UI and Backend	Final Functional Chatbot
TESTING & VALIDATION	Week 8-9	Conduct unit testing, evaluate response accuracy	Enhanced chatbot with improved response accuracy
DEPLOYMENT & EVALUATION	Week 10-11	Deploy chatbot and run live tests	Fully operational chatbot, user feedback report

We can evaluate the chatbot based on its response accuracy, user engagement, and efficient integration with the banking system.

Conclusion

The banking industry's increasing use of AI-powered chatbots demonstrates how revolutionary they are for improving customer service and operational effectiveness. Chatbots have become crucial in fintech app development due to the growing popularity of smartphones and other cutting-edge technology, allowing banks to maintain smooth client interactions while lowering their dependency on human employees. According to Juniper Research, in 2023, chatbot engagements are expected to save banks 862 million hours, or $7.3 billion in worldwide cost reductions. Furthermore, in 2023, 79% of chatbot interactions on mobile banking apps were fruitful, representing a startling 3150% rise in fruitful exchanges from 2019. These figures highlight how essential chatbots are for increasing efficiency and lowering costs. (Prasannan A. , 23). In summary, the bank service chatbot created with **Retrieval-Augmented Generation** (RAG) is a major step forward for the banking industry in terms of AI-powered customer service. This chatbot tackles important issues, including slow responses, a lack of personalization, and inefficiencies in conventional customer care techniques, by fusing real-time data retrieval with sophisticated natural language production. The RAG framework enables the chatbot to respond to intricate banking questions—from account inquiries to financial advice and transaction details—in a precise, context-aware, and customized manner. In addition to improving client satisfaction, this lowers operating expenses and increases financial institutions' scalability. By using state-of-the-art AI technologies, the chatbot can provide a smooth, human-like engagement experience while also adapting to changing client needs. This project thus establishes a new standard for AI-powered banking solutions, opening the door to a future in the financial services sector that is more inventive, efficient, and focused on consumer needs.

References

Ghandour, A. (2021). Opportunities and Challenges of Artificial Intelligence in Banking: Systematic Literature Review. *Research Gate*, 1. From **https://www.researchgate.net/publication/356600100_Opportunities_and_Challenges_of_Artificial_Intelligence_in_Banking_Systematic_Literature_Review**

Miehleketo Mathebula, A. M. (2024). Fine-Tuning Retrieval-Augmented Generation with an Auto-Regressive Language Model for Sentiment Analysis in Financial Reviews. *Research Gate*, 1.
From **https://www.researchgate.net/publication/386032705_Fine-Tuning_Retrieval-Augmented_Generation_with_an_Auto-Regressive_Language_Model_for_Sentiment_Analysis_in_Financial_Reviews**

Prasannan, A. (2023). The Rise of Chatbot Banking: Understanding its Potential. *Mindster*, 0. From **https://mindster.com/mindster-blogs/the-rise-of-chatbot-banking-understanding-its-potential/**

Prasannan, A. (23). The Rise of Chatbot Banking: Understanding its Potential. *Minister*, 1. From **https://mindster.com/mindster-blogs/the-rise-of-chatbot-banking-understanding-its-potential/**

Tom B. Brown, B. M. (2020). Language Models are Few-Shot Learners. *Research Gate*, 1. From **https://www.researchgate.net/publication/341724146_Language_Models_are_Few-Shot_Learners**

GeeksforGeeks
From **https://www.geeksforgeeks.org/software-engineering-differences-between-manual-and-automation-testing/**