

A PROJECT REPORT

on

“RAG-Based Loan Advisory Chatbot for Top Indian Banks”

**Submitted to:-
KIIT Deemed to be University**

**In Partial Fulfillment of the Requirement for the Award of
BACHELOR’S DEGREE IN
INFORMATION TECHNOLOGY**

BY:-

Atreyee Joardar	2106020
Ayush Jana	2106021
Dipayan Goswami	2106026
Pratibh Sinha	2106043
Rion Sutradhar	2106051

UNDER THE GUIDANCE OF

Dr. Murari Mandal

**SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

BHUBANESWAR, ODISHA - 751024

November, 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024



KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

CERTIFICATE

This is certify that the project entitled

“RAG-Based Loan Advisory Chatbot for Top Indian Banks”

Submitted by-

Atreyee Joardar	2106020
Ayush Jana	2106021
Dipayan Goswami	2106026
Pratibh Sinha	2106043
Rion Sutradhar	2106051

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2023-2024, under our guidance.

Date: /10/2024

Dr. Murari Mandal
(Project Guide)

Acknowledgements

We extend our deepest gratitude to **Dr Murari Mandal** of the **School of Computer Science and Engineering**. His expert guidance, unwavering support, and continuous encouragement have been instrumental in the successful completion of this project. From its inception to its fruition, **Dr Murari Mandal** has been a beacon of light, guiding us through the complexities of our work with his profound knowledge and vast experience.

His patience and perseverance in steering this project have left an indelible impression on us. His insightful critiques and constructive feedback have not only improved the quality of our work but also spurred personal and academic growth. His dedication to fostering a nurturing and stimulating learning environment has truly been inspiring.

We would also like to acknowledge our peers and colleagues, whose camaraderie and cooperation have made this journey an enriching and enjoyable experience. Their diverse perspectives and innovative ideas have greatly contributed to the project.

Lastly, we express our gratitude to our families and loved ones for their constant support and understanding throughout this process. Their faith in our abilities and their encouragement during challenging times have been the driving force behind our efforts.

In conclusion, we are immensely thankful for having had the opportunity to work under the guidance of **Dr. Murai Mandal** and for the invaluable experiences we have gained during this project. We look forward to applying the knowledge and skills we have acquired in our future endeavors.

Atreyee Joardar	2106020
Ayush Jana	2106021
Dipayan Goswami	2106026
Pratibh Sinha	2106043
Rion Sutradhar	2106051

Abstract

This project presents a Retrieval-Augmented Generation (RAG) chatbot developed to deliver precise, unbiased information on loan schemes offered by prominent Indian banks. Built using tools and libraries such as LangChain, ChatGroq, and Streamlit, the chatbot provides users with timely and accurate responses, facilitating informed financial decisions. The core RAG architecture harnesses retrieval-based search mechanisms and document-grounded response generation, resulting in answers that are contextually accurate and reliable.

The chatbot initiates with a comprehensive document ingestion pipeline using LangChain's PyPDFDirectoryLoader, which imports and organizes loan-related documents from multiple banks into a centralized knowledge repository. To enable efficient search and quick responses, the documents are divided into smaller, manageable segments through LangChain's RecursiveCharacterTextSplitter, ensuring optimal retrieval and processing.

To manage these document chunks, the chatbot applies Google Generative AI embeddings, transforming text data into high-dimensional semantic embeddings. These embeddings capture the nuanced meaning within the text, which are then stored in the FAISS vector database. FAISS, a scalable vector storage system, facilitates rapid similarity-based retrieval of document segments, allowing the chatbot to access and retrieve information relevant to each user query effectively.

The user interface, powered by Streamlit, takes user questions as input, initiating the RAG pipeline to retrieve and synthesize answers. Within this pipeline, the retrieval module collaborates with ChatGroq's LLaMA 3 generative language model. By feeding the model with relevant document chunks through a structured chat prompt template, responses remain anchored in the document context, minimizing the risk of error or irrelevant information. The system supports multiturn conversations, retaining context across interactions, which provides users with an uninterrupted, natural conversational experience.

To further build trust, a "Document Similarity Search" feature is integrated. Displayed through a Streamlit expander, this functionality shows document segments most similar to the user's query, offering transparency into the chatbot's decision-making process. By displaying relevant document sections alongside generated answers, users gain insights into the source material, enhancing the credibility of the information provided.

This RAG-based chatbot combines sophisticated retrieval techniques with generative responses, making it a valuable tool for users navigating complex financial information. The blend of advanced document processing, vector-based retrieval, and coherent response generation empowers users with transparent, trustworthy, and data-driven answers, enabling better financial decisions.

Contents

S. No.	Topic	Page No.
1	Introduction	6
2	Literature Review	7
	2.1 Text-Based Chatbot in Financial Sector	7
	2.2 Fine-Tuning Gemma-7B for Enhanced Sentiment Analysis of Financial News Headlines	7
	2.3 Improving Retrieval for RAG based Question Answering Models on Financial Documents	7
3	Requirement Specifications	8
	3.1 Project Planning	8
	3.1.1 Introduction	8
	3.1.2 Overall description	9
	3.2 Project Analysis (SRS)	9
	3.2.1 Functional requirements	9
	3.2.2 Performance requirements	10
	3.2.3 Design Constraints	11
	3.2.4 Quality attributes	11
	3.2.5 Requirements gathering	12
	3.3 System design	14
	3.3.1 Development/Implementation	15
	3.3.2 System architecture	17
4	Implementation and Testing	19
	4.1 Methodology/Proposal	19
	4.2 Testing /Verification Plan	20
	4.3 Result Analysis	22
	4.4 Screenshots of software analysis	22
5	Standards Adopted	27
	5.1 Design Standards	27
	5.2 Coding Standards	27
	5.3 Testing Standards	27
6	Conclusion and Future Scope	28
	6.1 Conclusion	28
	6.2 Future Scope	28
	References	29
	Individual Contribution	30

Chapter 1

1. Introduction

In today's fast-paced digital world, consumers seeking loans often face an overwhelming amount of information spread across various sources. Loan products, rates, eligibility criteria, and terms vary significantly between financial institutions, making it challenging for users to gather reliable, unbiased information on loan schemes that suit their specific needs. This complexity is compounded when users need to manually compare multiple loan products across banks, a time-intensive process that requires considerable financial literacy to interpret terms accurately. The lack of accessible and easily understandable advisory services leaves many users uncertain, often relying on third-party advice that may be biased or inconsistent. Addressing this need, we propose the development of a **Retrieval-Augmented Generation (RAG)-Based Loan Advisory Chatbot** tailored for India's top banks.

Problem Statement

Despite the digitalization of banking services, there remains a significant gap in accessible, user-friendly loan advisory tools for consumers in India. Key issues identified are:

1. **Information Overload:** The variety of loan schemes, interest rates, and eligibility conditions across banks can be confusing for users, especially when each bank presents this information differently. This results in an information overload that complicates the decision-making process.
2. **Lack of Centralized Information:** Users currently need to navigate multiple websites or contact bank representatives to gather details on loan options. This scattered information often leads to delays, increased effort, and, sometimes, missing out on better offers or suitable products.
3. **Complex Financial Jargon:** Loan documents often contain complex terms and conditions that are difficult for an average user to interpret without guidance, leaving users at risk of misunderstanding loan commitments.
4. **Limited Advisory Resources:** Traditional advisory services are either expensive or prone to bias, often focusing on certain banks or loan products. Additionally, physical interactions with bank advisors may not always be feasible for users in remote areas, reducing their access to tailored financial guidance.
5. **Time-Consuming Comparisons:** Manually comparing loans from different banks is time-consuming and requires significant effort from users. Without a streamlined comparison tool, users may find it challenging to make informed choices, leading to suboptimal decisions.

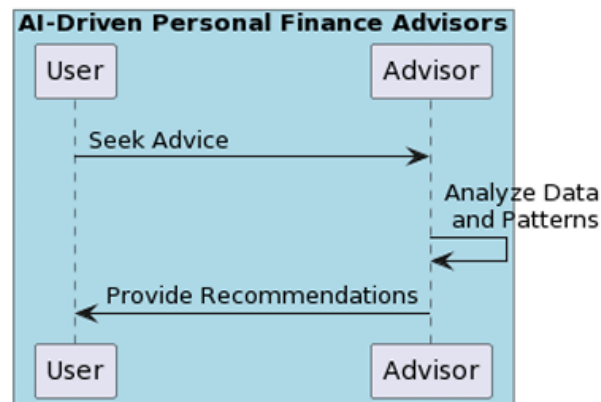


Fig. 1. Control Flow Diagram

Chapter 2 Literature Review

Text-Based Chatbot in Financial Sector: A Systematic Literature Review: Hana Demma Wube*, Sintayehu Zekarias Esubalew, Firesew Fayiso Weldesellase and Taye Girma Debelee

Text-based chatbots are implemented in the financial sector to enhance the relationship between the customer and services provided by the sector, and also to address external challenges and customer requirements. Chatbot technology in the financial sector serves to examine customers' frequently asked questions and the representation of the process using machine learning and AI models. In light of this, this study presents a comprehensive systematic literature review of articles focused on text-based chatbots in the financial sector. This systematic literature review analyzes the implementation, adoption, and user perception of text-based chatbots in the financial sector. It discusses challenges related to trust, privacy, and security, and identifies strategies to enhance the effectiveness of financial chatbots.

Disadvantages:

- The current research only includes chatbot communication via written language.

Fine-Tuning Gemma-7B for Enhanced Sentiment Analysis of Financial News Headlines: Kangtong Mo, Wenyan Liu, Xuanzhen Xu, Chang Yu, Yuelin Zou, Fangqing Xia

Exploring the application of sentiment analysis on financial news headlines to understand investor sentiment. By leveraging Natural Language Processing (NLP) and Large Language Models (LLM), we analyze sentiment from the perspective of retail investors. The FinancialPhraseBank dataset, which contains categorized sentiments of financial news headlines, serves as the basis for our analysis. This study investigates the application of sentiment analysis on financial news headlines using the Gemma-7B model. The fine-tuned model demonstrated significant improvements in accuracy, indicating its robustness in capturing financial sentiment nuances, which can aid in market insights and investment decisions.

Disadvantages:

- The study's limitations include domain specificity, short text constraints, data availability, real-time adaptability, interpretability, and potential biases, affecting the model's robustness and generalizability.

Improving Retrieval for RAG based Question Answering Models on Financial Documents Spurthi Setty, Harsh Thakkar, Alyssa Lee, Eden Chung, Natan Vidra

This paper addresses limitations in traditional RAG architectures when applied to financial documents. It proposes enhancements in document chunking and retrieval processes to improve the effectiveness of RAG-based question-answering models in knowledge-intensive and domain-specific tasks. Implementing these approaches can substantially improve the retrieval quality, thereby elevating the overall performance and reliability of LLMs in processing and responding to queries.

Disadvantages:

- primarily focuses on enhancing retrieval techniques but does not extensively address the integration of these improvements with the generation component of RAG models, potentially limiting the overall performance gains in question-answering tasks.

Chapter 3

2. Requirement Specifications

The core requirement of the project is to create a Retrieval Augmented Generation (RAG) chatbot by using the power of Gemma that will serve as a chatbot data assistant for users to find details of different types of loans offered by major banks in India. The major banks include SBI, ICICI, AXIS, HDFC and Kotak bank. The chatbot will help users who want to learn more about various loan products offered by financial institutions, such as home loans, personal loans, education loans, business loans. The system should provide accurate and timely answers regarding the details of each type of loan, including eligibility criteria, interest rates, fees, service charges, information requirements and procedures.

The RAG architecture was chosen to enable the chatbot to provide valuable insights, context and awareness of the user's questions, along with the ability to store and generate responses from previously recorded or archived data. For example, when a user asks about the personal loan interest rate of a particular bank, the chatbot needs to provide clear information regarding the bank and the type of loan. To maintain accuracy, the system needs to access new information and verify the accuracy of each bank loan.

In addition, a chatbot should have a user-friendly interface that allows people to interact with emotions without requiring deep knowledge. For example, if a user asks about eligibility and asks about documents, the chatbot should follow up on details to give a mixed answer.

The task of this project involves building a simple retrieval system to quickly decipher and process data. To be effective, chatbots need to optimize response time and allow for seamless interaction so users can easily track and retrieve specific credit information. Overall, the RAG chatbot aims to increase the accessibility of information and support users in making financial decisions.

Project Planning

Introduction Purpose

Now we will present the features of a RAG-based chatbot designed to provide users with reliable information about various types of loans offered by top banks in India. The chatbot provides responses to all the questions of the users and is designed to meet the needs of various financial institutions to access credit information. This not only helps users in making financial decisions by ensuring they get the correct credit details but also ensures transparency of banks offering information. The bot leverages the powerful ingestion and generation capabilities of the RAG architecture to provide consistent and relevant answers to various credit questions. This increases trust in financial services as users can understand the correct information about the loan and the bank can enhance customer service by providing credit information, new money and expansion through technology.

3.1.2. Overall Description

Product Perspective

In this project, we have focused on developing a RAG (Retrieval-Augmented Generation) chatbot using tools and libraries like LangChain, ChatGroq and Streamlit. We designed this product to provide users with fast, reliable and relevant information about various loan schemes that are available in Indian banks. RAG architecture allows us to leverage its power to create a retrieval-based search which provides generated answers to user questions by loading information from relevant documents. This product helps users to make better financial decisions that is devoid of any biased information and focuses more on providing precise and accurate information.

Product Functions

1. **Document Ingestion and Processing:** The chatbot uses a rich document ingestion pipeline from LangChain's PyPDFDirectoryLoader to ingest and process documents related to loan offerings by different banks. This will provide a centralized, structured repository of loan information. Once ingested, the documents are chunked using a RecursiveCharacterTextSplitter, splitting them into manageable chunks, each chunk-sized for quick retrieval and efficient response generation.
2. **Storage and Vector Embeddings:** For efficient vector-based storage of the documents, the chatbot uses Google Generative AI embeddings in order to convert the document chunk into high dimensional embeddings capturing semantic information in the text. These embeddings are stored in the FAISS database that is a scalable vector database supporting fast similarity-based retrieval. This vector store will enable the chatbot to look up chunks of information most related to each query that they pose.
3. **Question Processing and Answer Generation:** User interaction with the chatbot is through a Streamlit interface, which takes a question as input and starts the RAG pipeline to fetch and generate an answer. It integrates a retrieval module with a generative language model, ChatGroq's LLaMA 3 model, which is prompted to respond based on relevant document chunks. A chat prompt template is a structured prompt that the language model uses to generate contextually appropriate responses. This template also limits the response only up to the available document's context, thus ensuring it would be accurate and relevant. Multiturn conversations are enabled; thus, users can even go ahead and ask further follow-up questions. That means the system will carry on with the context further for smooth and coherent interaction to improve the user experience and keep it interactive and conversational.
4. **More Features:** The "Document Similarity Search" functionality is incorporated into the chatbot with a Streamlit expander, showing chunks that are most similar to a user's query, thus allowing users to get a sense of the context in which the answers are generated as well as to see parts of the documents relevant to the questions asked, further increasing transparency and trust.

3.1.3 Project Analysis (SRS)

a) Functional Requirements obtained by chatbot

1. Loan Query Processing

- **FR1:** The chatbot accepts queries in simple language, and supports text input.
- **FR2:** It understands user intents related to loan products, including home loans, personal loans, car loans, and education loans.
- **FR3:** The system categorizes queries by loan type to streamline information retrieval.

2. Retrieval-Augmented Generation (RAG) for Answer Generation

- **FR4:** The chatbot integrates with an indexed database of loan-related FAQs, terms, and conditions for each partner bank.
- **FR5:** It retrieves the most relevant documents related to the user query, using keyword and context-based matching.
- **FR6:** The system generates responses by combining retrieved document data with generative AI capabilities, ensuring the final answer is clear, accurate, and relevant.
- **FR7:** Answers are contextually aware of Indian banking norms and regulations to maintain compliance and relevance.

3. Bank-Specific Loan Information Retrieval

- **FR8:** The chatbot displays customized loan options, rates, eligibility criteria, and tenure options for the top Indian banks, including SBI, HDFC, ICICI, Axis Bank, and Punjab National Bank.
- **FR9:** For authenticated users, it provides personalized offers, interest rates, and eligibility information based on user profile and credit score data from the bank's backend.
- **FR10:** The chatbot provides comparisons across banks for similar loan types based on updated data from respective bank databases.

4. Loan Application Assistance

- **FR11:** The chatbot guides users through the loan application process by providing step-by-step instructions.
- **FR12:** It validates user inputs for mandatory fields such as loan amount, tenure, and collateral details (if applicable) and prompts users to correct any missing or invalid data.

5. Eligibility Check and Document Verification

- **FR13:** The chatbot performs a preliminary eligibility check by asking relevant questions regarding income, employment status, credit history, etc.
- **FR14:** The system validates and extracts necessary documents, including identity proofs, income proofs, and other required paperwork, either by directly uploading files or by providing document IDs linked with Aadhaar, PAN, etc.

6. Loan Query Processing

- **FR15:** It understands user intents related to loan products, including home loans, personal loans, car loans, and education loans.
- **FR16:** The system categorizes queries by loan type to streamline information retrieval.

b) Non Functional Requirements obtained by chatbot

1. Performance Requirements

- **NFR1: Response Time** - The chatbot responds to user queries within 2 seconds for standard information retrieval and within 5 seconds for more complex, multi-document RAG-based answers.
- **NFR2: Concurrent Users** - The system supports concurrent users without degradation in performance.
- **NFR3: Latency** - The chatbot maintains low latency with a goal of less than 100ms for data retrieval from the backend.
- **NFR4: Error Handling** - The chatbot must gracefully handle unexpected inputs, network issues, or backend failures, displaying appropriate error messages without affecting the user experience.

2. Scalability Requirements

- **NFR5: Horizontal Scalability** - The system supports horizontal scaling to handle increases in traffic, ensuring it can handle peak loads, especially during high-demand periods (e.g., promotional campaigns or end-of-quarter loan processing).
- **NFR6: Data Volume** - The chatbot is able to scale to accommodate large volumes of loan-related documents and FAQ databases, ensuring the RAG model can access and process all necessary information efficiently.
- **NFR7: Regional Adaptability** - The architecture should support the addition of new regions, languages, and banks without extensive reconfiguration.

3. Security Requirements

- **NFR8: Compliance with Regulations** - The system must comply with RBI regulations and data privacy laws (e.g., GDPR if applicable) and be regularly audited for adherence to these standards.
- **NFR9: User Anonymity for Non-Account Holders** - Non-account-holders interacting with the chatbot should remain anonymous, with no personal data stored without user consent.

4. Usability Requirements

- **NFR10: Intuitive User Interface** - The chatbot should have a simple, user-friendly interface accessible through web browsers, mobile applications, and third-party platforms like WhatsApp.
- **NFR11: Accessibility Compliance** - The interface should be designed in compliance with accessibility standards.

5. Portability Requirements

- **NFR12: Device Compatibility** - The chatbot should be compatible across devices (e.g., desktop, mobile, tablet) and platforms (e.g., Android, iOS, Windows).
- **NFR14: Deployment Flexibility** - The solution should support deployment on cloud-based and on-premises infrastructures, depending on the bank's preference.

6. Documentation Requirements

- **NFR13: User Documentation** - Comprehensive user manuals and FAQs should be available for end-users, providing guidance on using the chatbot for loan advisory.
- **NFR14: Technical Documentation** - Detailed technical documentation should be provided, including system architecture, API references, and integration instructions to aid developers in future maintenance and updates.
- **NFR15: Administrator Guide** - An administrator guide for people should include instructions on updating chatbot content, monitoring usage, and handling escalations.

Solution Overview

The RAG-Based Loan Advisory Chatbot is designed to address these challenges by providing a centralized, efficient, and reliable advisory platform that aggregates loan-related data from leading Indian banks and delivers precise, document-based answers to user queries. This chatbot is built using Retrieval-Augmented Generation (RAG) architecture, leveraging tools and libraries such as LangChain, ChatGroq, and Streamlit. By combining retrieval-based search with generative language capabilities, the chatbot provides accurate, contextually relevant responses to user questions based on official documents and terms from each bank.

Expected Benefits

The RAG-Based Loan Advisory Chatbot offers multiple benefits:

- **Simplified Loan Comparison:** Users can easily compare loan options across different banks without needing to visit multiple websites or engage with numerous bank representatives.
- **Improved Financial Literacy:** By presenting information in simple language and clarifying complex financial terms, the chatbot empowers users with the knowledge to make informed financial choices.
- **Increased Access to Advisory Services:** The chatbot provides 24/7 accessibility to users across India, including those in remote areas with limited access to in-person financial advisors.
- **Reduced Bias:** The chatbot delivers unbiased information directly from bank documents, ensuring users receive factual data without third-party influence.

Performance Requirements

PR1: Scalability

PR1.1: The RAG-Based Loan Advisory Chatbot must be scalable to handle an increasing number of users across different banks, offering a consistent performance even as user demand grows.

PR2: Speed

PR2.1: In the RAG-Based Loan Advisory Chatbot, swift processing of user inquiries and loan recommendations is essential. When users seek loan options or advice, the chatbot must quickly process these requests to deliver real-time, accurate responses. This includes analyzing the user's financial profile, matching it with available loan options from partner banks, and returning results immediately. Efficient response times enhance user experience, encouraging regular usage and aiding in effective loan recommendation. Therefore, ensuring responses within an acceptable time frame is crucial for the chatbot's effectiveness and user satisfaction.

Design Constraints

DC1: AI and Machine Learning Technology

DC1.1: The chatbot will utilize AI and machine learning models for more accurate loan recommendations based on users' risk profiles. Leveraging AI allows the system to analyze large datasets, evaluate multiple variables (like credit scores, income, and bank requirements), and provide tailored loan options. This technology improves the recommendation's accuracy and ensures that the chatbot remains reliable as it advises users on optimal loan products. AI-driven personalization eliminates the need for constant human intervention, reducing operational costs and increasing efficiency.

DC2: Integration with Banking Systems

DC2.1: In the RAG-Based Loan Advisory Chatbot, integration with each bank's loan management system is critical. This allows seamless data exchange, ensuring that loan options, interest rates, eligibility criteria, and user application statuses are consistently synchronized. It enables real-time tracking of loan recommendations and approval statuses, enhancing the chatbot's effectiveness. Moreover, it facilitates automatic updating of loan options and criteria based on the latest information from banks, improving accuracy. Thus, seamless integration is essential for comprehensive and accurate loan advisory services.

Quality Attributes

QA1: Accuracy

QA1.1: The chatbot's primary objective is to accurately recommend suitable loan options based on individual user profiles. By leveraging machine learning, each user is assigned a risk profile (RAG-based: Red, Amber, or Green) during the initial interaction. This profile considers key attributes, including credit score, income, and debt-to-income ratio, and is stored securely. When a user seeks advice, their profile is used to identify the most appropriate loan products. This process ensures accurate loan recommendations, providing personalized guidance to consumers and helping banks reach suitable borrowers, ultimately reducing default rates and supporting informed lending practices.

QA2: Security

QA2.1: Data security in the RAG-Based Loan Advisory Chatbot is crucial, as it handles sensitive data like user financial details and personal information. This is achieved through secure storage, data encryption, and strict access controls. Regular security audits and system updates further strengthen data protection. By ensuring high data security, the chatbot builds user trust, protects privacy, and maintains the reliability of the loan advisory process across partner banks, promoting safe and secure loan recommendation practices.

Project Initiation

The project, "RAG-Based Loan Advisory Chatbot," has the following scope and objectives:

Scope:

1. **User Interface:** Design an intuitive interface that allows users to easily interact with the RAG-Based Loan Advisory Chatbot. The interface should enable users to input their financial details, view recommended loan options, and receive advisory insights based on their unique profiles.
2. **Scalability:** Ensure that the chatbot can handle a high volume of user inquiries across various banks, providing efficient service without performance degradation as user numbers grow.
3. **Security:** Implement robust access controls and encryption protocols to safeguard sensitive user and financial data from unauthorized access, maintaining user privacy and trust.

Objectives:

4. **User Financial Empowerment:** Provide users with accurate, personalized loan recommendations, helping them make informed decisions and find the best loan products for their financial needs.

5. **Bank Efficiency:** Aid banks in reducing the risk of loan defaults by accurately assessing user profiles and recommending loans to suitable borrowers, ultimately optimizing their lending processes.
6. **Marketplace Trust:** Foster a trustworthy financial marketplace by offering transparent, AI-driven loan recommendations that enhance user confidence in banking products and promote responsible borrowing practices.

Stakeholders:

In the RAG-Based Loan Advisory Chatbot, stakeholders include:

- **Users:** They use the chatbot and provide feedback while also relying on the system for accurate results.
- **System Developers:** They design, implement, and maintain the system, ensuring its functionality and security.
- **Regulatory Bodies:** They oversee the system's compliance with data protection and consumer rights laws. These stakeholders play crucial roles in the system's operation and effectiveness.

Project Team:

The project team for the "RAG-Based Loan Advisory Chatbot" includes:

- **Project Manager:** Oversees the project, ensuring it stays on track.
- **System Analyst:** Defines system requirements and specifications.
- **LLM Specialist:** Develops and maintains the LLM for better integration with the app.
- **Database Administrator:** Manages the database.
- **Quality Assurance Engineer:** Ensures the system meets quality standards.
- **Legal Advisor:** Ensures compliance with laws.

Requirements Gathering

To understand user needs for a "RAG-Based Loan Advisory Chatbot" system, both user interviews and surveys can be conducted. Here are some examples:

User Interviews:

Bank Loan Officers: Gather insights into their challenges with loan recommendations and risk assessment processes, and understand how an AI-based advisory chatbot could streamline their workflows.

- What challenges do you face when assessing loan applications?
- How do you currently evaluate borrower risk profiles?
- What features would be beneficial in an AI-based loan advisory system?

Bank Customers: Explore their experiences with finding suitable loan products and gauge their interest in a chatbot that offers tailored loan recommendations.

- What challenges do you encounter when looking for a suitable loan?
- Would a chatbot that provides personalized loan options be useful for you?
- What features or guidance would you find helpful in such a system?

Financial Advisors: Understand their processes for assessing borrower risk and advising on loan options, and explore how a chatbot could complement their advisory work.

- How do you determine the right loan products for different customers?
- Would an AI-based chatbot that assists in risk assessment and loan recommendation be useful?
- What concerns or requirements do you have for using such a system?

Surveys:

Loan Awareness and Interest Survey: Assess user awareness of various loan products, their interest in personalized recommendations, and their comfort level with using AI-based advisory tools.

System Usability Survey: Gather feedback on the usability of the chatbot system, including the ease of providing personal details, receiving loan recommendations, and navigating the platform.

Feature Preference Survey: Understand which features users find most valuable, such as personalized loan suggestions, risk assessments, interest rate comparisons, and application guidance.

These interviews and surveys will provide valuable insights into user needs and preferences, guiding the development of the "RAG-Based Loan Advisory Chatbot" system.

Requirements

The user requirements for the “RAG-Based Loan Advisory Chatbot for Top Indian Banks” system, including the tech stack and features to be developed, are as follows:

Tech Stack:

1. Backend:

- Python: We used python as the core programming language and for integrating all the components with each other along with data processing.
- LangChain: All the various interactions between the language model is done using the LangChain framework. This includes document ingestion and text splitting of the relevant chunks and retrieval-augmented generation (RAG) operations.

2. Data Ingestion and Processing:

- PyPDF2: Library which is a part of PyPDFDirectoryLoader and is used for data ingestion by loading and parsing PDF Documents.
- RecursiveCharacterTextSplitter: Tool which is used for efficient retrieval of relevant chunks which are made by splitting documents.

3. Vector Embeddings and Storage:

- Google Generative AI Embeddings: Used for making high quality embeddings for document chunks, translating text into vector representations that capture semantic meaning.
- FAISS: We used this for retrieving and managing large collections of embeddings.

4. Language Model and Prompting:

- ChatGrok: Language model (LLaMA 3) used for generating responses based on retrieved document context. ChatGrok’s LLaMA 3 model processes structured prompts for precise, context-aware answers.
- ChatPromptTemplate: Custom prompt design tool that structures the input to the language model, specifying response context and ensuring relevance to the query.

5. User Interface:

- Streamlit: Web-based application framework for creating an interactive user interface. Streamlit enables real-time user input, response display, and additional features like "Document Similarity Search" using expandable sections.

6. Environmental Management and API Keys:

- dotenv: Library for managing environment variables and securely loading API keys, including GROQ API Key and Google API Key, from a .env file.

Hardware Environment: The hardware requirements would depend on the scale of the project. For development purposes, a standard computer with a good processor (Intel i5 or above), sufficient RAM (8GB or above), and ample storage (256GB or above) would be needed. For deploying the system, a server with higher specifications would be required, depending on the expected load and number of transactions

System Design

The chatbot is designed to answer user questions by retrieving and generating responses based on document-based loan information. The system is divided into two main stages: Loading & Indexing and Query Processing

1. Loading & Indexing Stage

This stage prepares the documents for efficient retrieval during user queries. It includes several steps

- The system connects to multiple data sources, such as documents, databases, and APIs. These connectors allow access to the raw content needed for the chatbot's knowledge base.
- The system extracts content from ingested documents about loans and creates manageable chunks of text using the library from LangChain RecursiveCharacterTextSplitter and then each document is split into smaller segments (e.g., 100 characters with 20-character overlap) to improve retrieval relevance and efficiency.
- The system then generates vector embeddings for each document chunk using Google Generative AI Embeddings which contain the semantic meaning of the content, allowing similar content to be retrieved based on user queries.
- A FAISS (Facebook AI Similarity Search) vector store is then used to store these embeddings and manage document chunks, forming a searchable index that supports fast similarity-based retrieval.

2. Query Processing Stage

This stage handles user interactions by processing queries and retrieving relevant content to generate responses.

- When a user submits a question, the system generates an embedding of the query text using the same embedding model used for document chunks. This ensures that both the query and document chunks are represented in the same vector space for accurate similarity comparison.
- The query embedding is used to perform a semantic search within the FAISS vector store. The search retrieves the most relevant document chunks (nodes) based on their vector similarity to the query, ensuring contextually appropriate responses.
- Retrieved document chunks are structured into a prompt template using ChatPromptTemplate. The prompt is passed to the ChatGroq (LLaMA 3) language model, instructing it to generate a response based solely on the context provided by the retrieved chunks. This model, enhanced by the structured prompt, generates a response that is both contextually relevant and concise.
- We used a Streamlit interface to display the generated response providing the user with an answer based on the document data. The UI includes an expander for "Document Similarity Search," allowing users to view relevant document chunks related to their query.

Workflow Summary

1. The user enters a question on the Streamlit interface.
2. The system retrieves relevant document chunks from FAISS based on query similarity.
3. Retrieved chunks are used to create a prompt for the language model.
4. ChatGroq generates a response based on the context.
5. The response, along with relevant document chunks, is displayed to the user.

Development/Implementation

This project utilized several advanced technologies to build an efficient Retrieval-Augmented Generation (RAG) chatbot for answering queries on loan schemes. Key tools and libraries included LangChain for document management, ChatGroq for response generation, Google Generative AI for embedding creation, and Streamlit for developing the user interface.

Environment Setup and API Integration

This project was developed using **Visual Studio Code** as the local IDE, leveraging **Streamlit** for the user interface, allowing for easy interaction on a web-based platform. The system environment variables were configured through **Python's os module** and the **dotenv library**, securely loading the **GROQ** and **Google API keys** from a dedicated `.env` file.

Initial Phase: Defining Core Components

The project's initial phase involved defining the chatbot's primary components, starting with **Streamlit's structure** and **title** to guide user interactions. A **GROQ language model**, "**Llama3-8b-8192**," was initialized for processing and responding to user queries. This selection was based on the model's **optimal performance** in handling **large language datasets** relevant to **financial advisory**.

Data Preparation: Embedding Financial Documents

A critical step in preparing data for the chatbot involved **embedding financial documents**. The function `vector_embedding()` created embeddings using **Google's Generative AI embeddings model**. This embedding model, "**models/embedding-001**", was applied to **financial PDFs** loaded through **PyPDFDirectoryLoader** from a specified directory on the local machine. **Recursive text splitting** was implemented using **RecursiveCharacterTextSplitter** to divide large documents into manageable text chunks with a **chunk size of 1000** and a **200-character overlap**, ensuring that **context was preserved across boundaries**. This split data was embedded using **FAISS**, a **vector database library** that enables high-speed **similarity search**, creating a **searchable vector database**.

Core Chatbot Logic and User Input Processing

For the main chatbot logic, a **dynamic prompt** was generated based on the **user's input**, with instructions guiding the model to respond strictly within the **context of the embedded documents**. The chatbot interface accepted user questions and leveraged **Streamlit's** `st.text_input()` function to capture the input. When a user triggered document embedding, the `vector_embedding()` **function** was called to load and process the PDF files, embedding them to prepare for **subsequent query processing**.

Query Processing and Answer Retrieval

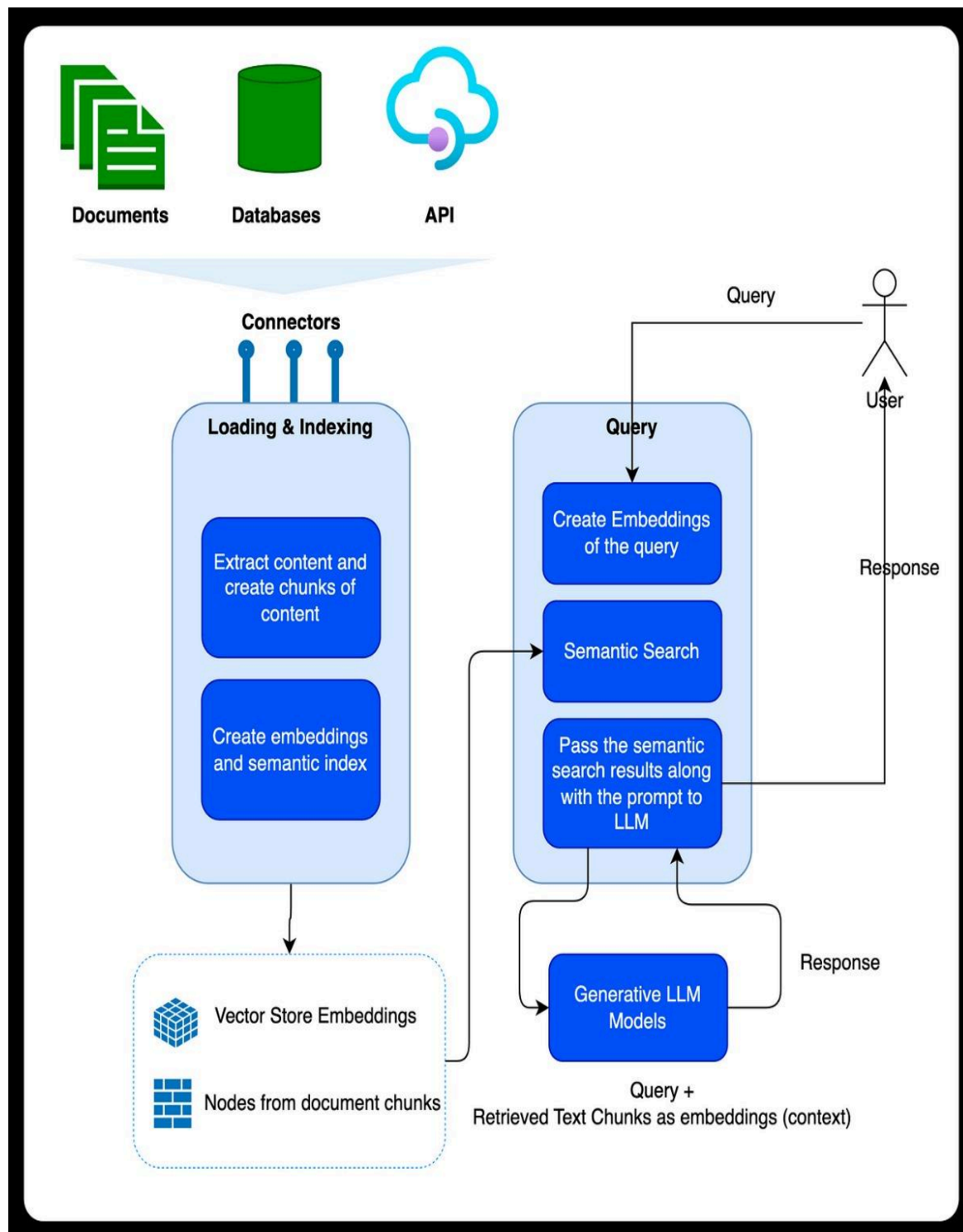
Through the **Streamlit interface**, users input queries that trigger the chatbot's **retrieval chain**. Using **LangChain's** `create_retrieval_chain()` function, the chatbot searches relevant document segments within the **FAISS vector database** and feeds these sections into the **ChatGroq LLM model (Llama 3)** for **response generation**. By combining **document similarity search** with the **GROQ model's output**, the retrieval chain ensures that **responses are contextually accurate** and grounded in the **embedded documents**. Additionally, **query processing time** was measured, providing **valuable insights** into **system performance** and areas for further optimization.

Displaying Responses and Ensuring Document Transparency

For response presentation, **Streamlit's** `st.write()` function was used to output generated answers directly in the interface. Additionally, an expandable **Streamlit** `st.expander` component provided access to relevant **document snippets** retrieved from the **FAISS database**. This feature allowed users to view **supporting context** for each response, enhancing **transparency** by displaying the specific **source segments** that informed the chatbot's answers

With this systematic approach, the RAG chatbot was successfully developed to provide accurate, transparent, and contextually relevant answers to loan-related queries, facilitating a reliable user experience for individuals seeking financial guidance.

System Architecture



Chapter 4

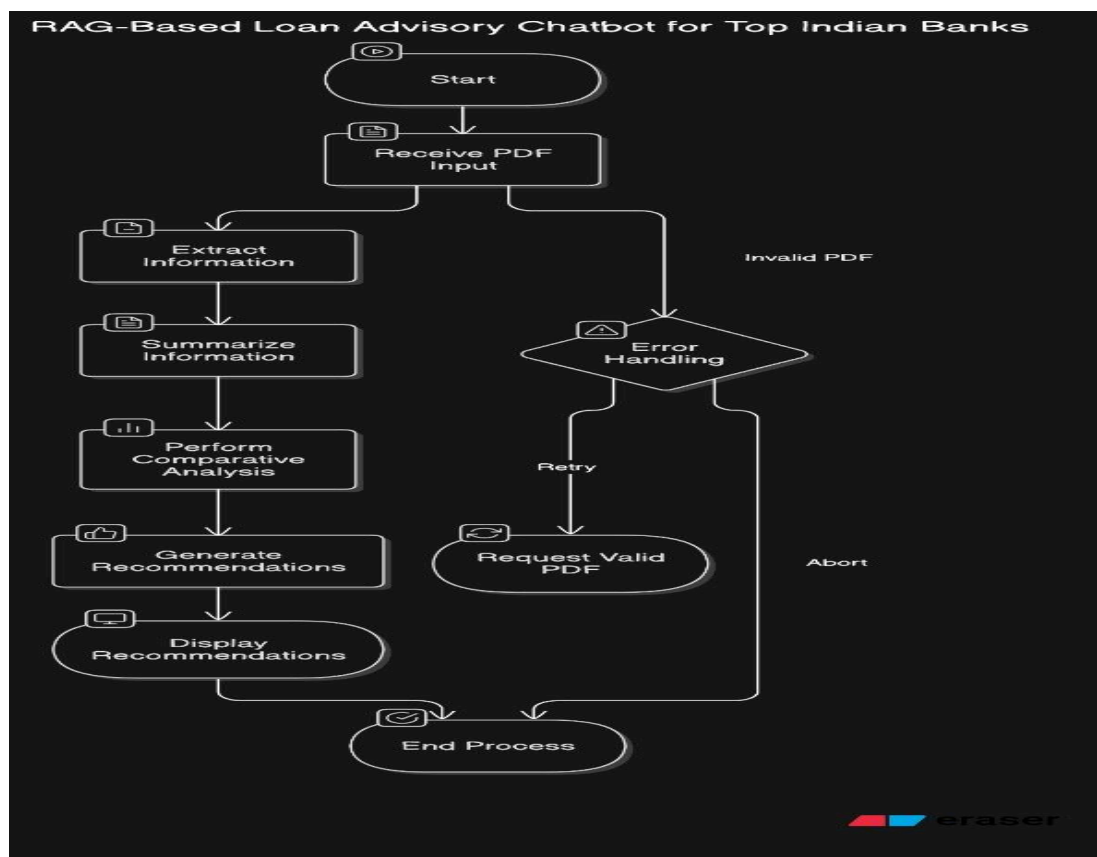
Implementation and Testing

The software implementation phase and the testing phase are studies that aim to provide stakeholders with information about the quality of the product or service being tested. Software testing can also provide an objective and independent view of the software so that the company can assess and understand the risks of implementing the software. Testing techniques involve running a program or application to find software errors (bugs or other errors). Software testing involves running a software component or system component to evaluate one or more properties of interest. In general, these characteristics indicate the extent to which a component or system has been tested.

- meets the requirements that guide its design and development, responds correctly to all possible inputs.
- performs its tasks in a reasonable amount of time, is reasonably usable,
- is installable and usable in its intended environments, and
- achieves the overall result desired by the stakeholders.

Since the number of possible tests for even simple software components is almost unlimited, all software testing uses some strategy to select tests that are feasible with available time and resources. As a result, software testing generally (but not exclusively) attempts to run a program or application with the goal of finding software bugs (bugs or other defects). Testing is an iterative process, like fixing a single bug; it may reveal other deeper bugs or even create new ones.

Methodology/Proposal



Testing/Verification Plan

The main aim of testing is to analyze the performance and to evaluate the errors that occur when the program is executed with different input sources and running in different operating environments.

The following standard testing methods are essential for evaluating a Retrieval-Augmented Generation (RAG) chatbot, such as the one developed in this project. These approaches help ensure that the chatbot's responses are accurate, relevant, and user-friendly.

1. **Relevance Testing:** Sample queries with known relevant document segments are used to verify the retrieval component's accuracy. This helps ensure the chatbot consistently retrieves appropriate information aligned with user queries.
2. **Precision and Recall Analysis:** Calculating precision and recall for retrieval performance is critical. By comparing retrieved segments against expected results, this method assesses the chatbot's ability to fetch relevant information accurately and without omissions.
3. **Groundedness and Factuality Checks:** This testing method verifies that responses are grounded in source documents, ensuring that generated answers are factually correct and based on retrieved information, minimizing unsupported content.
4. **Usability Testing:** Real-user testing is essential for assessing interface ease-of-use and overall user experience. Collecting feedback from users highlights areas for improving navigation, engagement, and interface clarity.
5. **Multiturn Conversation Evaluation:** Simulating multi-turn conversations tests the chatbot's capacity to retain context across exchanges, ensuring coherent and uninterrupted interaction throughout complex queries.
6. **Transparency Feature Validation:** Features such as Document Similarity Search are tested to ensure they are user-friendly and enhance trust. Allowing users to view document segments related to their questions improves the chatbot's credibility.
7. **Latency and Performance Testing:** This method measures response times to confirm that the chatbot delivers answers promptly. Testing under varying loads verifies that the system remains efficient and responsive even during high-traffic usage.
8. **Scalability Testing:** As the document repository expands, stress-testing the vector database, such as FAISS, is crucial to ensure sustained performance. This testing confirms that retrieval and processing remain effective with growing data volumes.

These testing methods collectively ensure that a RAG chatbot delivers accurate, contextually relevant, and user-centric responses, making it suitable for real-world applications.

The testing steps are:

- Unit Testing.
- Integration Testing.
- Validation Testing.
- User Acceptance Testing

UNIT TESTING: Unit testing, also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs.

INTEGRATION TESTING: Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together.

VALIDATION TESTING: At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected. Validation testing can be defined in many ways; here the testing validates the software function in a manner that is reasonably expected by the customer.

USER ACCEPTANCE TESTING: Performance of an acceptance test is actually the user's show. User motivation and knowledge are critical for the successful performance of the system. The above tests were conducted on the newly designed system performed to the expectations.

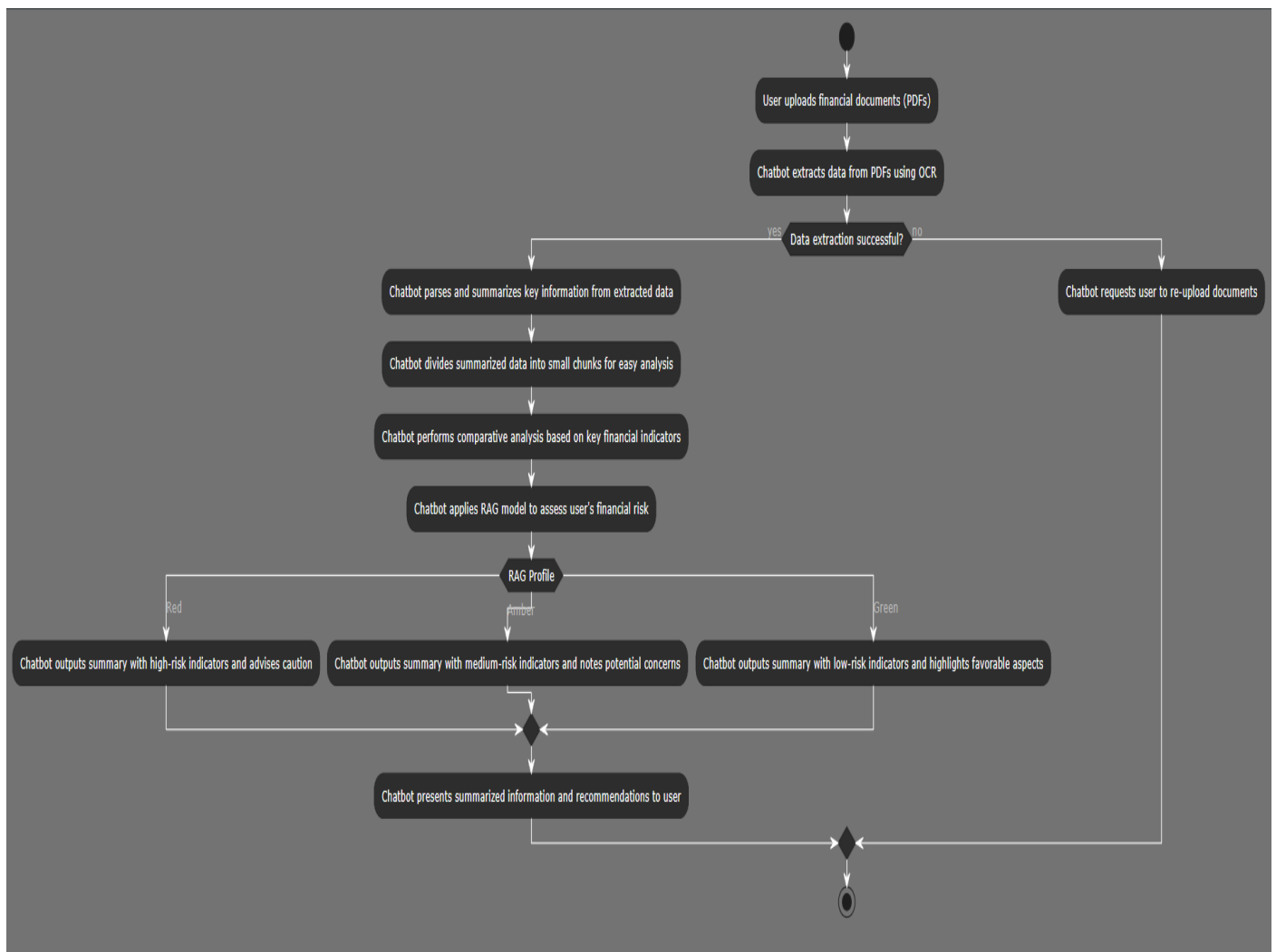
Test Table:

Test Case	1
Name of Test	Comparison of Home Loan interest
Input	Query to compare all home loan interest rates of different banks in India
System Behavior	Will generate information regarding all home loan interest of different banks
Expected Output	Response relevant to Query
Result	Successful

Test Case	2
Name of Test	Eligibility Criteria of Home Loan
Input	Query to get information about current eligibility criteria of home loan of ICICI Bank
System Behavior	Will generate information relevant to the eligibility criteria of home loan of ICICI Bank
Expected Output	Response relevant to Query
Result	Successful

Test Case	3
Name of Test	Shaurya Scheme of SBI
Input	Query to get information on how to enroll to the Shuarya Scheme of SBI
System Behavior	Will generate information relevant about how to enroll to the Shuarya Scheme of SBI
Expected Output	Response relevant to Query
Result	Successful

Result Analysis



Screenshots of Software Analysis

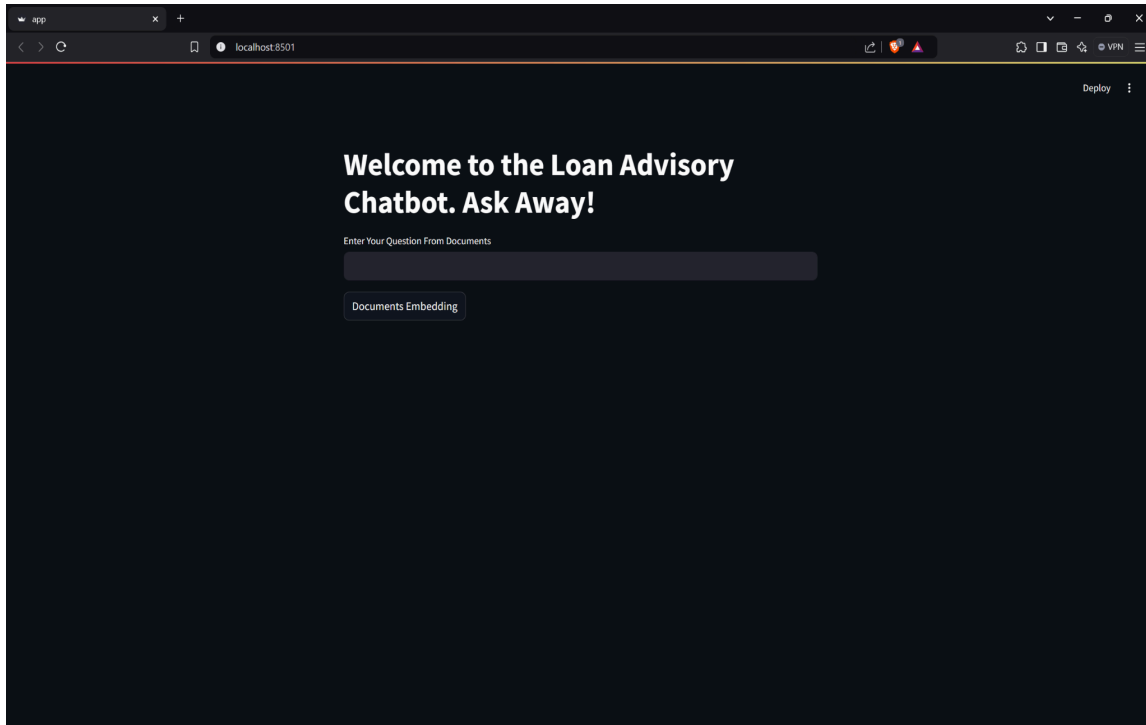


Fig. 4.1. Homescreen UI

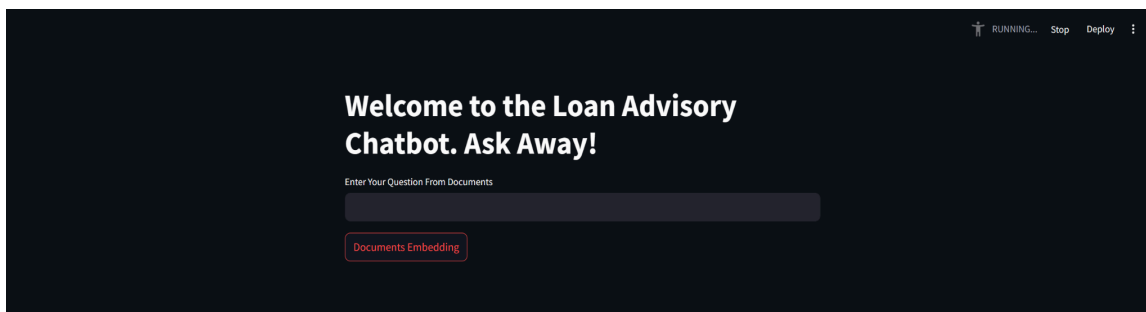


Fig. 4.2. Document Embeddings being created

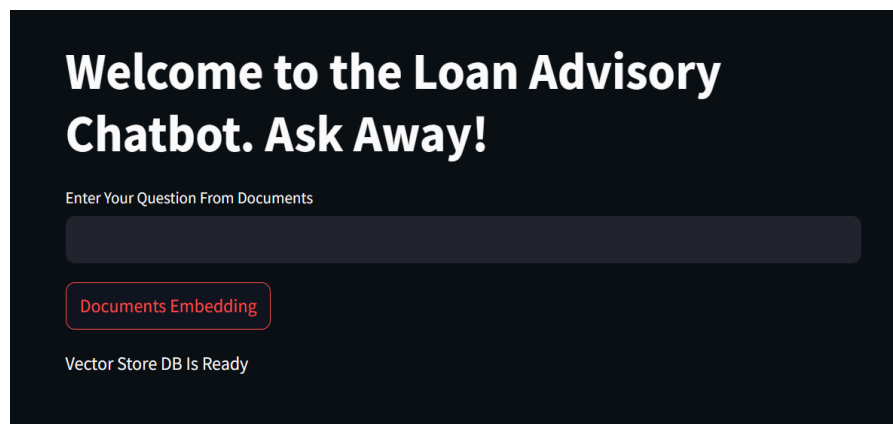


Fig. 4.3. Vector Store DB ready

Welcome to the Loan Advisory Chatbot. Ask Away!

Enter Your Question From Documents

can you tell me about student or education loan from kotak bank

Documents Embedding

Vector Store DB Is Ready

Based on the provided context, here are the key points about Kotak Bank's Education Loan:

Loan Amount:

- For education in India: Up to Rs. 10 Lakhs*
- For education abroad: Up to Rs. 20 Lakhs*

Eligibility Criteria:

- Age Eligibility: Not specified in the context
- Student Eligibility: The student should be an Indian National (including NRIs) and have secured admission to a foreign university/Institution.

Eligible Expenses:

- Fee payable to college/school/hostel
- Examination/Laboratory/Library fee
- Travel expenses/passage money for studies abroad
- Purchase of books/equipments/instruments/uniforms/computer (if required for course completion)
- Caution deposit/building fund/refundable deposit (supported by Institution bills/receipts)
- Premium of 'RiNn Raksha' insurance policy
- Contingency expenses (up to 5% of loan amount) for unforeseen requirements.

Repayment:

- Repayment Holiday/Moratorium: Course period plus 1 year
- Repayment Period: Up to 15 years after completion of course and moratorium
- Interest Servicing: Optional during course and moratorium

Top-Up Loan:

- Available for further studies, subject to certain conditions.

Please note that this information is based on the provided context and may not be exhaustive or up-to-date. For more information, I recommend visiting the Kotak Bank website or contacting their customer service directly.

Document Similarity Search



Fig. 4.4. Prompt for Kotak Bank with response

Kotak Bank Education Loan:

Education Loan

A good education paves the path to success, and we believe that everyone has the right to an education that will help them grow and prosper. That's why our Education Loans are designed to finance meritorious students who pursue higher education, both in India and abroad. With quick approval rates, minimal documentation, and high loan amounts based on income and repayment capacity, you will never be far from making your dreams come true.

- **Loan Amount for Education in India:** Up to Rs. 10 Lakhs*
- **Loan Amount for Education Abroad:** Up to Rs. 20 Lakhs*
- **Quick Approval**
- **Minimal Documentation Requirements**

Disclaimer: For loans: Conditions apply. Credit at the sole discretion of the Bank and subject to guidelines issued by RBI from time to time. Bank may engage the services of marketing agents for sourcing loan assets.

Eligibility Criteria

- **Age Eligibility**

will be recovered from the borrower:

- Penal interest @ 2% p.m. to be charged for loans above Rs. 4 lacs for the overdue amount and overdue period.
11. **Repayment Holiday/ Moratorium :** • Repayment Holiday/ Moratorium: Course period plus 1 year
 - Servicing of Interest during Course & Moratorium: Optional
 12. **Repayment Period:** • Repayment Period: Upto 15 years after completion of course & moratorium.
 - If the student is not able to complete the course within the scheduled time, extension of time for completion of course may be permitted for a maximum period of 2 years.
 - In case the interest is not serviced during course & moratorium, the accrued interest to be added to the principal and repayment to be fixed in Equated Monthly Installments (EMI).
 13. **Top Up Loan**

If a student approaches the bank for a loan for further studies, the same can be granted to him/her subject to the following conditions:

 - The combined loan amount should not exceed the

Fig. 4.5. Document Similarity Search Feature for above Question

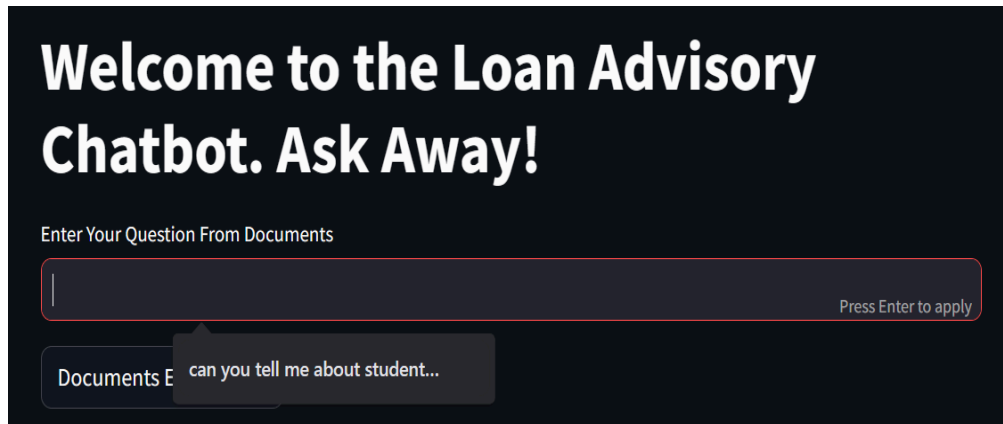


Fig. 4.6. Previously asked Question is stored in memory

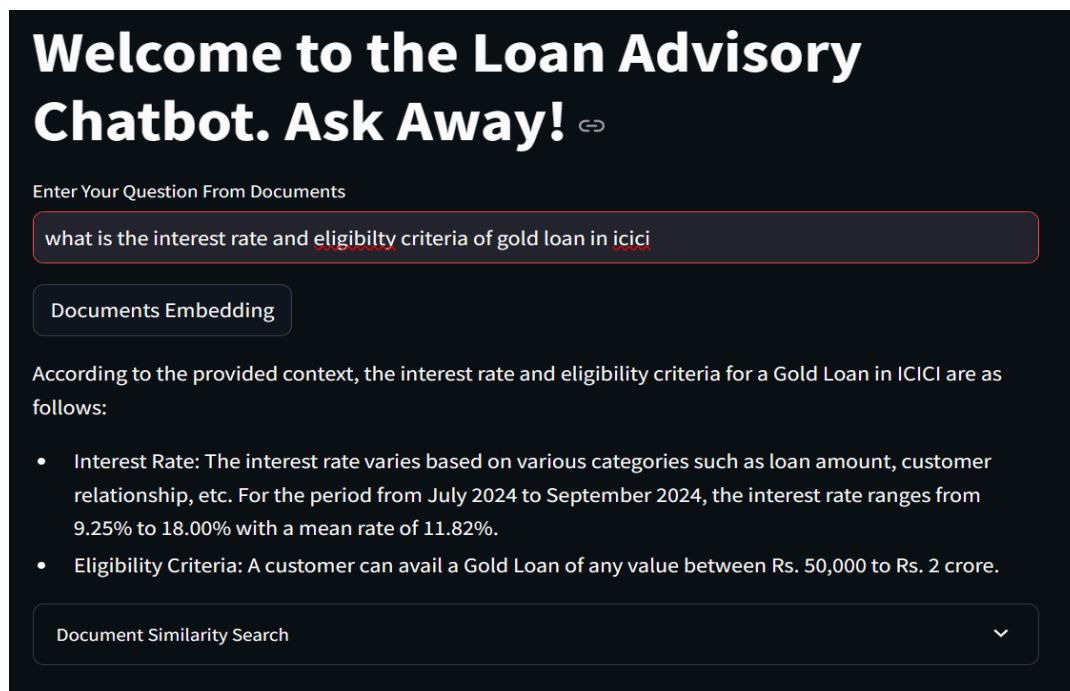


Fig. 4.7. Some more prompts and its responses

Chapter 5

Standards Adopted

Design Standards

For the design aspect of our project, we adhere to established engineering standards such as those defined by IEEE (Institute of Electrical and Electronics Engineers) and ISO (International Organization for Standardization). These standards provide a framework for designing robust and reliable systems. In software engineering, we also follow industry- standard practices such as Unified Modeling Language (UML) diagrams and database design standards to ensure clarity and coherence in our design approach.

Coding Standards

In terms of coding, we adhere to established coding standards encompassing rules, guidelines, and best practices. Some of the key principles we follow include:

- Writing concise code with as few lines as possible to enhance readability and maintainability.
- Adhering to appropriate naming conventions to facilitate understanding and maintainability of the codebase.
- Segmenting blocks of code within the same section into paragraphs for clarity and organization.
- Using indentation to demarcate the beginning and end of control structures, thereby improving code readability.
- Ensuring that functions are concise and focused, ideally performing a single task to promote code modularity and reusability.

Testing Standards

In our project, we adhere to established testing standards defined by ISO (International Organization for Standardization) and IEEE (Institute of Electrical and Electronics Engineers) to ensure the quality and reliability of our product. These standards encompass various aspects of quality assurance and testing methodologies. Specific standards followed for testing and verification of our project work include ISO/IEC 25010:2011 for software product quality and ISO/IEC/IEEE 29119 for software testing.

This section draws upon established standards and best practices in engineering, software development, and testing to ensure the robustness, reliability, and quality of the system.

Chapter 6

Conclusion and Future Scope

Conclusion

This project successfully developed a Retrieval-Augmented Generation (RAG) chatbot designed to provide accurate, unbiased information on loan schemes from prominent Indian banks. By combining LangChain, ChatGroq, and Streamlit, the chatbot effectively streamlines vast financial data, making it accessible for users. The RAG architecture merges precise retrieval methods with document-grounded response generation, ensuring contextually relevant answers to user queries.

A robust document ingestion pipeline organizes loan-related documents, with FAISS and Google Generative AI embeddings facilitating efficient, similarity-based retrieval. This enables the chatbot to respond quickly and accurately by drawing directly from pertinent document segments. To enhance transparency, a “Document Similarity Search” feature is integrated, allowing users to view relevant document sections linked to their queries, thus strengthening trust in the information provided.

This chatbot offers a valuable resource for users navigating complex financial choices, underscoring the practical potential of RAG frameworks in delivering reliable, data-driven insights. Future enhancements may broaden its scope across financial domains and incorporate evolving information, solidifying its role as a trustworthy, user-centric tool for financial decision-making.

Future Scope

The RAG chatbot developed in this project has significant potential for expansion and improvement. In future iterations, the chatbot could broaden its scope beyond loan schemes to include a wider range of financial products, such as savings accounts, investment options, credit cards, and insurance policies. This would make it a comprehensive financial advisory tool, assisting users with various aspects of personal finance.

To enhance adaptability, the system could incorporate real-time data integration, enabling it to provide users with the latest information on interest rates, policy changes, and promotional offers from different banks. Additionally, incorporating multilingual capabilities would allow the chatbot to serve a more diverse user base, making financial information accessible to non-English speakers across India.

Another promising area for development is the incorporation of advanced natural language understanding, allowing the chatbot to handle even more nuanced user queries and provide personalized responses. Enhancing the chatbot’s contextual memory for multi-turn conversations would also improve its ability to deliver seamless, human-like interactions.

Furthermore, integrating a feedback mechanism could enable the chatbot to learn from user interactions and improve over time, while also allowing financial institutions to better understand customer needs. With these advancements, the RAG chatbot could become an invaluable tool in promoting financial literacy and empowering users to make informed financial decisions.

References

- [1]. Yunfan Gaoa, Yun Xiongb, Xinyu Gaob, Kangxiang Jiab, Jinliu Panb, Yuxi Bic, Yi Daia, Jiawei Suna, Meng Wangc, and Haofen Wang “Retrieval-Augmented Generation for Large Language Models: A Surve” arXiv:2312.10997v5 [cs.CL] 27 Mar 2024, aShanghai Research Institute for Intelligent Autonomous Systems.
- [2]. Pouria Omrani, Alireza Hosseini, Kiana Hooshanfar, Zahra Ebrahimian, Ramin Toosi, Mohammad Ali Akhaee, Hybrid Retrieval-Augmented Generation Approach for LLMs Query Response Enhancement, 2024 10th International Conference on Web Research (ICWR), Faculty of Electrical Engineering, K. N. Toosi University of Technology.
- [3]. Dr. Selva Kumar S, Afifah Khan Mohammed Ajmal Khan, Imadh Ajaz Banday, Manikantha Gada, Vibha Venkatesh Shanbhag, “Overcoming LLM Challenges using RAG-Driven Precision in Coffee Leaf Disease Remediation”, 2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS).
- [4]. YEONCHAN AHN JUNHO SHIM 1,SANG-GOO LEE 2,(Member, IEEE), 3,(Senior Member, IEEE), AND JAEHUI PARK 2, “ Retrieval-Augmented Response Generation for Knowledge-Grounded Conversation in the Wild” ,Received 2 November 2022, accepted 8 December 2022, date of publication 12 December 2022, date of current version 21 December 2022.
- [5].Sonia Vakayil , Anitha. J, D. Sujitha Juliet , Sunil Vakayil, “ RAG-based LLM Chatbot using Llama-2” , 2024 7th International Conference on Devices, Circuits and Systems (ICDCS) | 979-8-3503-5047-0/24/\$31.00 ©2024 IEEE | DOI: 10.1109/ICDCS59278.2024.10561020.
- [6]. Text-Based Chatbot in Financial Sector: A Systematic Literature Review: Hana Demma Wube, Sintayehu Zekarias Esubalew, Firesew Fayiso Weldesellasi and Taye Girma Debelee.
- [7]. Fine-Tuning Gemma-7B for Enhanced Sentiment Analysis of Financial News Headlines: Kangtong Mo, Wenyan Liu, Xuanchen Xu, Chang Yu, Yuelin Zou, Fangqing Xia.
- [8]. Improving Retrieval for RAG based Question Answering Models on Financial Documents Spurthi Setty, Harsh Thakkar, Alyssa Lee, Eden Chung, Natan Vidra

INDIVIDUAL CONTRIBUTION

RAG-Based Loan Advisory Chatbot for Top Indian Banks

ATREYEE JOARDAR
2106020

Abstract: Navigating the vast array of loan options available today can be overwhelming for customers seeking the best financial solutions. To simplify this process, a Rule-based and Generative AI (RAG) Loan Advisory Chatbot offers tailored loan recommendations for leading Indian banks. This advanced system uses a hybrid model, combining rule-based logic with generative AI to provide accurate, personalized guidance that aligns with each individual's financial needs. With the help of machine learning and data analytics, the chatbot can assess user input and preferences to suggest ideal loan options, supporting customers in making well-informed choices. This technology improves the customer experience by simplifying the decision-making process and enables banks to offer a faster, more seamless loan selection journey. By enhancing personalization and efficiency, it helps build stronger relationships between customers and financial institutions.

Individual contribution and findings: Throughout this project, I concentrated on utilizing my skills in requirements analysis and data management to support our objectives. My contributions included drafting the Software Requirements Specification (SRS) document, designing the system architecture, and collecting comprehensive data on various HDFC loan types. I thoroughly cleaned and preprocessed this data before feeding it into the Large Language Model (LLM) to enable accurate loan recommendations. Working closely with the team, I aimed to ensure that our chatbot's functionality was grounded in precise requirements and reliable data, contributing meaningfully to the project's success by supporting a streamlined and effective loan advisory solution.

Full Signature of Supervisor:

.....

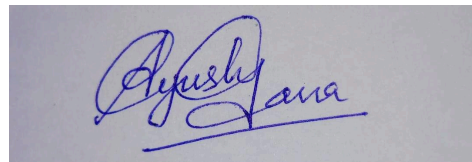
Full signature of the student:

.....

AYUSH JANA
2106021

Abstract: The rising demand for financial products has increased the complexity of loan selection, making it challenging for customers to find suitable loan options. A Rule-based and Generative (RAG)-Based Loan Advisory Chatbot offers a solution by providing users with personalized loan recommendations for top Indian banks. This system utilizes a hybrid approach, combining rule-based methods with generative AI to offer accurate and reliable advice tailored to individual financial profiles. By integrating machine learning algorithms and data analytics, the chatbot can proactively analyze user input and preferences, delivering optimal loan choices while helping users make informed decisions. This technology benefits both customers and banks, as it improves user satisfaction and streamlines the loan selection process.

Individual contribution and findings: In this project, I focused on conducting a comprehensive literature survey and drafting the Software Requirements Specification (SRS) document along with data procurement of all types of loans provided by **Axis Bank**. My literature survey involved analyzing existing RAG-based systems and assessing the applicability of such technology in the context of loan advisory for Indian banks. By developing a well-defined SRS document, I ensured clarity in the project's requirements and contributed to setting a solid foundation for the project roadmap. My efforts showcased my commitment to understanding and documenting the necessary specifications for our chatbot, ultimately aiding in the project's success.



Full Signature of Supervisor:

.....

Full signature of the student:

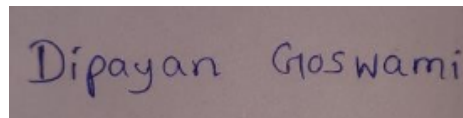
.....

DIPAYAN GOSWAMI

2106026

Abstract: The rising demand for financial products has increased the complexity of loan selection, making it challenging for customers to find suitable loan options. A Rule-based and Generative (RAG)-Based Loan Advisory Chatbot offers a solution by providing users with personalized loan recommendations for top Indian banks. This system utilizes a hybrid approach, combining rule-based methods with generative AI to offer accurate and reliable advice tailored to individual financial profiles. By integrating machine learning algorithms and data analytics, the chatbot can proactively analyze user input and preferences, delivering optimal loan choices while helping users make informed decisions. This technology benefits both customers and banks, as it improves user satisfaction and streamlines the loan selection process.

Individual contribution and findings: In this project, I focused on developing core sections of the project report, including the Development/Implementation, Implementation/Testing, Conclusion, and Future Scope. My work involved detailing the technical processes, implementation strategy, and testing methods to ensure the chatbot's functionality and reliability. I also assessed the outcomes and proposed directions for future improvements. Additionally, I procured and provided data on all types of loans from ICICI Bank, which served as a crucial dataset for training the chatbot. By contributing to both the documentation and data preparation stages, I helped establish a comprehensive understanding of the project's requirements and provided essential resources that enhanced the chatbot's effectiveness.

A rectangular box containing a handwritten signature in purple ink that reads "Dipayan Goswami".

Full Signature of Supervisor:

.....

Full signature of the student:

PRATIBH SINHA

2106043

Abstract: Navigating the vast array of loan options available today can be overwhelming for customers seeking the best financial solutions. To simplify this process, a Rule-based and Generative AI (RAG) Loan Advisory Chatbot offers tailored loan recommendations for leading Indian banks. This advanced system uses a hybrid model, combining rule-based logic with generative AI to provide accurate, personalized guidance that aligns with each individual's financial needs. With the help of machine learning and data analytics, the chatbot can assess user input and preferences to suggest ideal loan options, supporting customers in making well-informed choices. This technology improves the customer experience by simplifying the decision-making process and enables banks to offer a faster, more seamless loan selection journey. By enhancing personalization and efficiency, it helps build stronger relationships between customers and financial institutions.

Individual contribution and findings: Throughout this project, I collaborated closely with my team to contribute valuable insights into the chatbot's technical architecture and functionalities. By working on the backend, I ensured seamless data ingestion and retrieval processes, optimizing the interaction between the RAG architecture and the data sources for accurate loan-related responses. My efforts were integral in establishing a robust backend framework that maintained fast and reliable document-based response generation. I also provided critical input into enhancing user data privacy measures, integrating secure API handling, and implementing efficient database management practices.

Full Signature of Supervisor:

.....

Full signature of the student:

.....

RION SUTRADHAR

2106051

Abstract: With increasing demand in the products of finance, loan selection has become more complex and, as a result, more difficult to find the right loan option for customers. An RAG-based loan advisory hybrid Rule-based and Generative can solve this problem by generating recommendations on available top banks in India, according to each individual's financial profiles. Integration of machine learning algorithms and data analytics enables the chatbot to analyze user inputs and preferences proactively, deliver the best possible loans to customers, and guide the informed decision-making of the users. Customer and banks' benefits are notable - in that the satisfaction of the users is ensured while the process of loan selection is simplified.

Individual contribution and findings: Throughout the development phase of our project, I worked with my peers on implementing the ChatBot by using various libraries. I also leveraged my data analysis skills to find out data about existing loan schemes from different banks and also cleaned and processed them. I also helped with formulating and defining the problem statement along with making recommendations on what future scopes this project has.

Full Signature of Supervisor:

.....

Full signature of the student:

.....