Generative AI-Based Chatbot for Employee and Customer Support Automation in LOLC Company

Group ID: R25-036

Project Proposal Report

B.Sc. (Hons) Degree in Information Technology Specialization in Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

2025- January

Supervisor: Prof. Nuwan Kodagoda

Co-supervisor: Dr. Lakmini Abeywardhana

Student: Gamage U.R - IT21807480

Generative AI-Based Chatbot for Employee and Customer Support Automation in LOLC Company

Group ID: R25-036

Project Proposal Report

B.Sc. (Hons) Degree in Information Technology Specialization in Software Engineering

Sri Lanka Institute of Information Technology

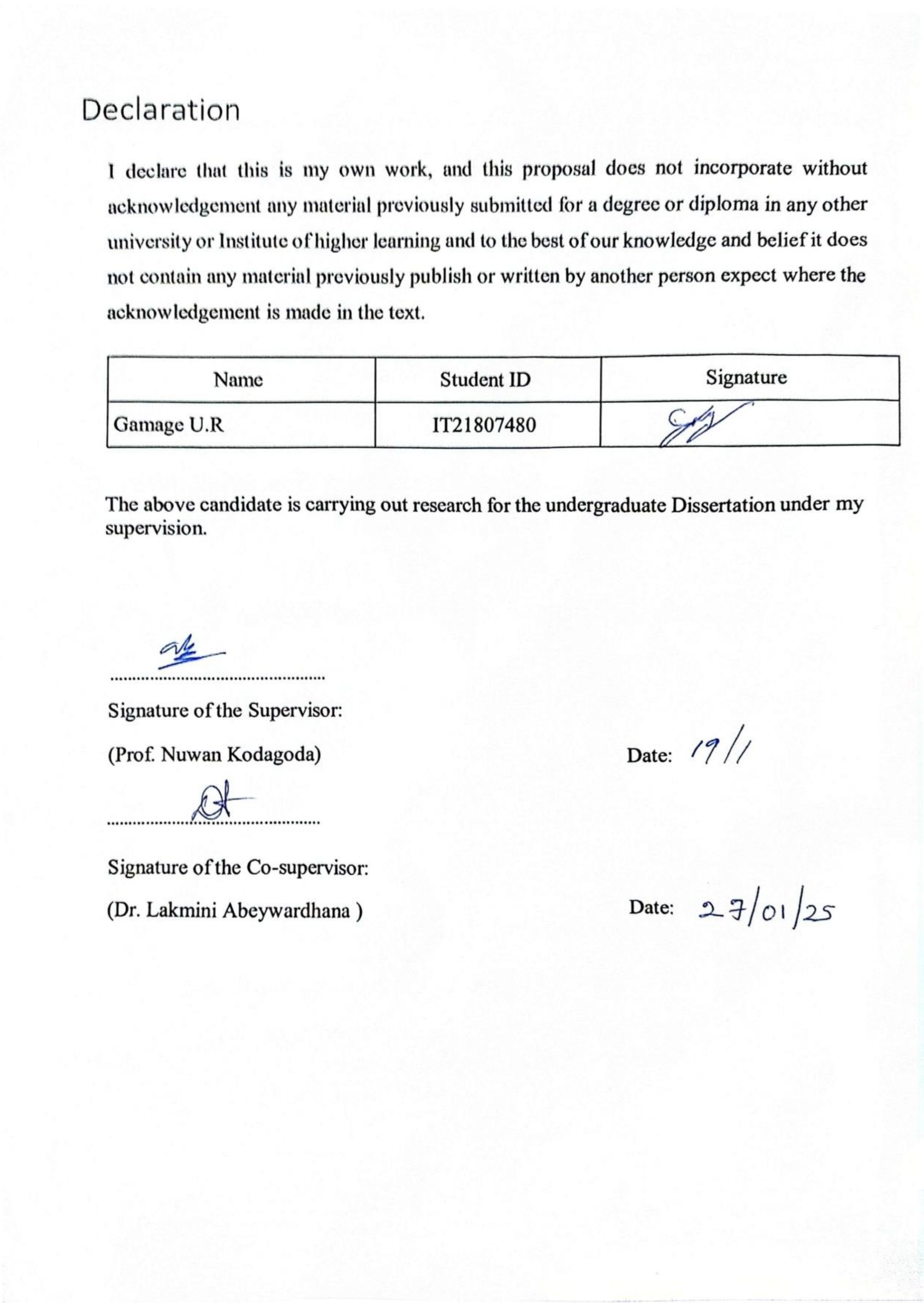
Sri Lanka

2025- January

Supervisor: Prof. Nuwan Kodagoda

Co-supervisor: Dr. Lakmini Abeywardhana

Student: Gamage U.R - IT21807480



Declaration

I declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously publish or written by another person expect where the acknowledgement is made in the text.

Name	Student ID	Signature
Gamage U.R	IT21807480	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

....

Signature of the Supervisor:

(Prof. Nuwan Kodagoda) Date: 19/1

....

Signature of the Co-supervisor:

(Dr. Lakmini Abeywardhana) Date: 27/01/25

Abstract

This research project focuses on developing a generative AI - based chatbot for LOLC Company, designed to streamline and enhance their IT support systems. The chatbot integrates a range of features that enable employees to efficiently resolve IT-related queries, gain insights into HR policies, and access business-related information. By leveraging an advanced knowledge base, the chatbot provides precise and context-specific responses to user prompts, significantly improving operational efficiency and decision-making within the organization.

A central component of the system is its intelligent knowledge base, which allows dynamic updates through a user-friendly interface. This feature empowers staff to upload new documents, update existing policies, and refine content, ensuring that responses remain accurate and relevant over time. The knowledge base also enables personalized answers by learning from user interactions and retaining episodic memory of previous queries. This ensures a tailored experience for each user while continuously improving response accuracy based on feedback.

In addition to text-based support, the chatbot features an immersive 3D avatar equipped with voice-to-text and text-to-voice capabilities, enabling seamless interaction with employees. The system addresses knowledge gaps by auto-generating answers and providing guidance for completing forms, thus reducing manual effort and improving user efficiency. An agent-based automation system further enhances productivity by executing repetitive tasks based on instructions derived from the knowledge base.

This innovative solution is built to adapt dynamically to changing organizational needs while maintaining scalability and reliability. By combining personalized support, streamlined processes, and dynamic knowledge management, the chatbot sets a new benchmark for intelligent enterprise support systems, driving efficiency and improving the overall employee experience at LOLC.

Table of Contents

Table of Figures

List of Abbreviations

RAG - Retrieval-Augmented Generation

LLM - Large Language Model

Gen AI - Generative Artificial Intelligence

BERT - Bidirectional Encoder Representations from Transformers

NLP - Natural Language Processing

TTS - Text-to-Speech

API - Application Programming Interface

LOLC - Lanka ORIX Leasing Company

AZ - Azure (Microsoft cloud platform)

UI - User Interface

CLI - Command-Line Interface

HTTPS - Hypertext Transfer Protocol Secure

TLS - Transport Layer Security

1.Introduction

1.1 Background

In today's rapidly evolving technological landscape, the rise of generative AI technologies is transforming the way businesses operate. Tasks that were traditionally performed manually by employees are increasingly being automated through AI-driven solutions. As a result, enterprises, particularly large multinational companies, must modernize their existing systems to stay aligned with the latest technological trends.

In this context, we have partnered with LOLC companies to enhance their existing IT services. Specifically, we aim to upgrade their current IT ticketing system by incorporating state-of-the-art generative AI features. Our objective is to improve employee productivity and efficiency by integrating a chatbot solution powered by an internal knowledge base.

The primary goal of this project is to design a chatbot capable of answering employee queries related to HR policies, IT support, and other business processes. This solution will streamline internal communication, reduce the time spent on routine inquiries, and ultimately enhance employee productivity. By leveraging generative AI, we aim to create a personalized, fast, and efficient ticketing system that minimizes human involvement in resolving tickets and eliminates time-consuming processes.

The current manual ticketing systems often require significant human effort, are prone to delays, and lack a personalized approach to problem-solving. Employees frequently face challenges in obtaining accurate answers quickly, which impacts their productivity. By implementing a knowledge base tailored to the company's internal data, employees will have instant access to the information they need. Additionally, the system will empower non-technical staff to update and modify data effortlessly, ensuring that the platform remains relevant and up-to-date.

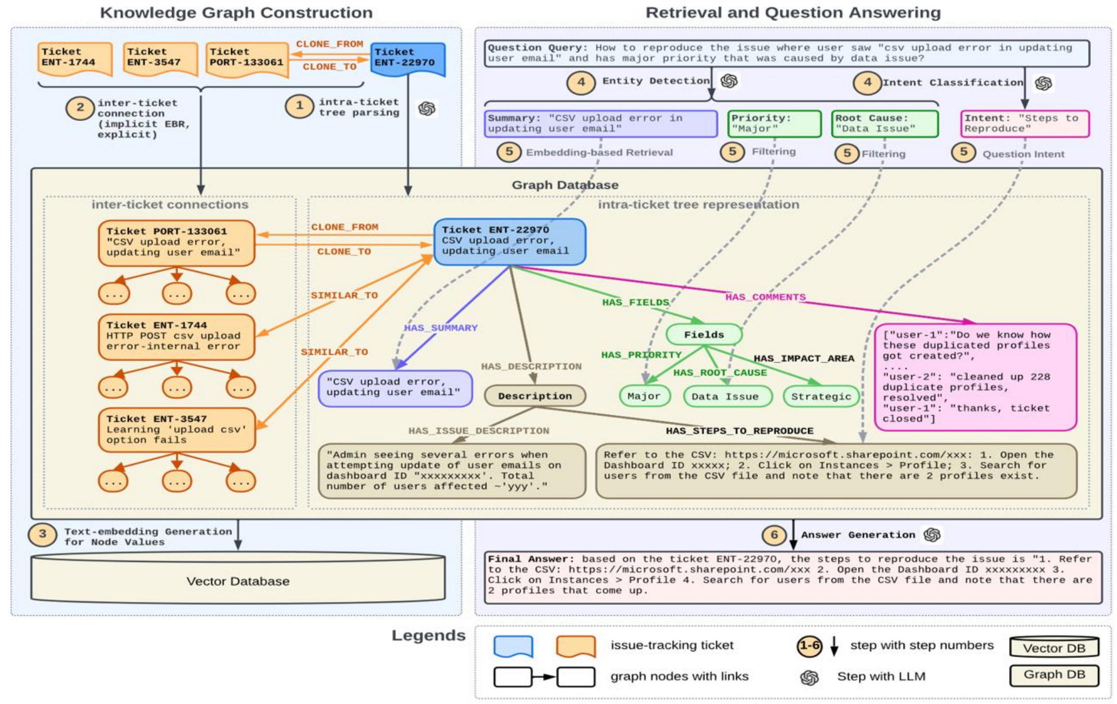
This research project seeks to develop a cutting-edge AI-powered solution that not only addresses these inefficiencies but also helps financial enterprises like LOLC integrate AI into their existing systems. This integration will enable companies to improve operational efficiency, boost employee satisfaction, and remain competitive in an AI-driven future.

1.2 Literature survey

Researchers have been conducting extensive studies over the past decades on large language models (LLMs) and chatbot-related technologies. These efforts have culminated in the development of transformer-based LLMs that have revolutionized natural language understanding and generation. In domains like IT support and HR policies, significant research contributions have focused on improving the quality and reliability of automated responses.

The integration of Retrieval-Augmented Generation (RAG) [1] with LLMs has garnered significant attention for its ability to combine generative capabilities with real-time retrieval [2], enhancing accuracy and contextual relevance. Han et al. (2024) underscore its role in automating systematic literature reviews (SLRs) [6], showcasing how RAG simplifies processes like literature search, screening, data extraction, and synthesis. The grounded approach of RAG mitigates issues such as hallucinations and ensures accurate outputs in fields requiring precise information, like medicine and law.

In IT support, [8] highlight the effectiveness of RAG systems in resolving incident tickets. Their solution combines retrieval and generative components, addressing challenges like domain-specificity and the constraints of smaller models. This approach enhances support by grounding responses in verified documentation, improving accuracy and user trust. Additionally, [13] demonstrate a novel integration of RAG with knowledge graphs for customer service, preserving structural and relational data to enhance retrieval accuracy and reduce resolution time by 28.6%.



Question Query: How to reproduce the issue where user saw "csv upload error in updating user email" and has major priority that was caused by data issue?

Summary: "CSV upload error in updating user email"

Priority: "Major"

Root Cause: "Data Issue"

Intent: "Steps to Reproduce"

["user-1":"Do we know how these duplicated profiles got created?", ... "user-2": "cleaned up 228 duplicate profiles, resolved", "user-1": "thanks, ticket closed"]

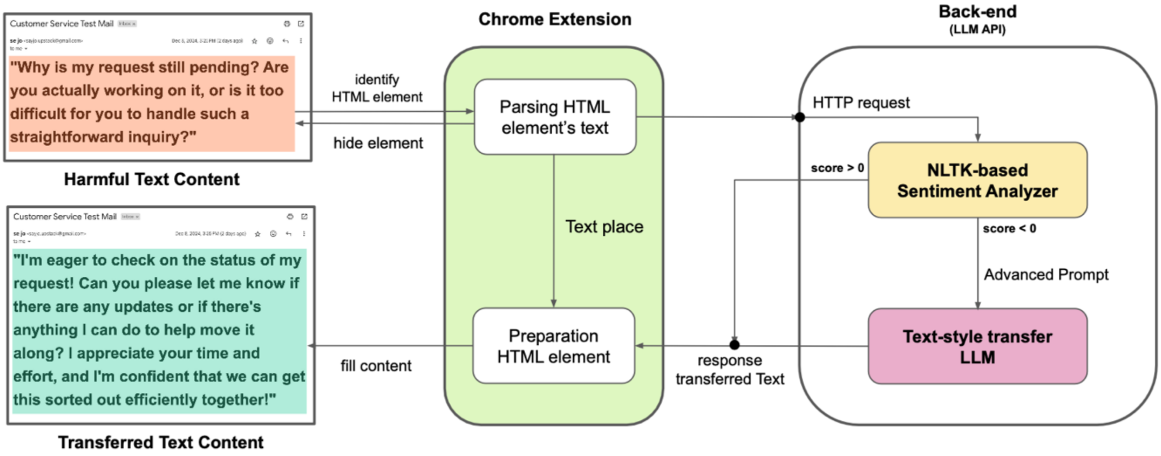
"Admin seeing several errors when attempting update of user emails on dashboard ID "xxxxxxxxx'. Total number of users affected ~'yyy'."

Refer to the CSV: https://microsoft.sharepoint.com/xxx: 1. Open the Dashboard ID xxxxx; 2. Click on Instances > Profile; 3. Search for users from the CSV file and note that there are 2 profiles exist.

Final Answer: based on the ticket ENT-22970, the steps to reproduce the issue is "1. Refer to the CSV: https://microsoft.sharepoint.com/xxx 2. Open the Dashboard ID xxxxxxxxx 3. Click on Instances > Profile 4. Search for users from the CSV file and note that there are 2 profiles that come up.

Figure 1 - Architecture of the system

Jo and Seo (2024) focus on human-centered applications, presenting ProxyLLM, which transforms harmful customer messages into neutral tones using text-style transfer. This innovation addresses the emotional strain on agents and improves operational efficiency in customer service environments. [7] explores HR chatbots, emphasizing advancements in vector search and the transition to powerful LLMs like GPT-4. These improvements align chatbot outputs with organizational policies, demonstrating the adaptability of RAG systems in enterprise applications.



"Why is my request still pending? Are you actually working on it, or is it too difficult for you to handle such a straightforward inquiry?"

"I'm eager to check on the status of my request! Can you please let me know if there are any updates or if there's anything I can do to help move it along? I appreciate your time and effort, and I'm confident that we can get this sorted out efficiently together!"

Figure 2 - ProxyLLM Background process

Despite these advancements, several gaps remain. Current systems often struggle with domain-specific nuances [12], multimodal data processing, and the efficient handling of complex queries[4]. The dependency on static datasets for retrievers can limit adaptability, especially in rapidly changing environments[3]. Future research could explore integrating adaptive retrievers with multimodal inputs, leveraging fine-tuned domain-specific LLMs, and enhancing real-time learning capabilities. Furthermore, innovative prompt engineering and reinforcement learning could refine the interaction between retrieval and generation, offering more robust and context-aware outputs.

Overall, the integration of RAG and LLMs has advanced automation in literature review, IT support, HR, and customer service. These systems demonstrate the potential to revolutionize knowledge-driven industries by delivering accurate, reliable, and efficient solutions.

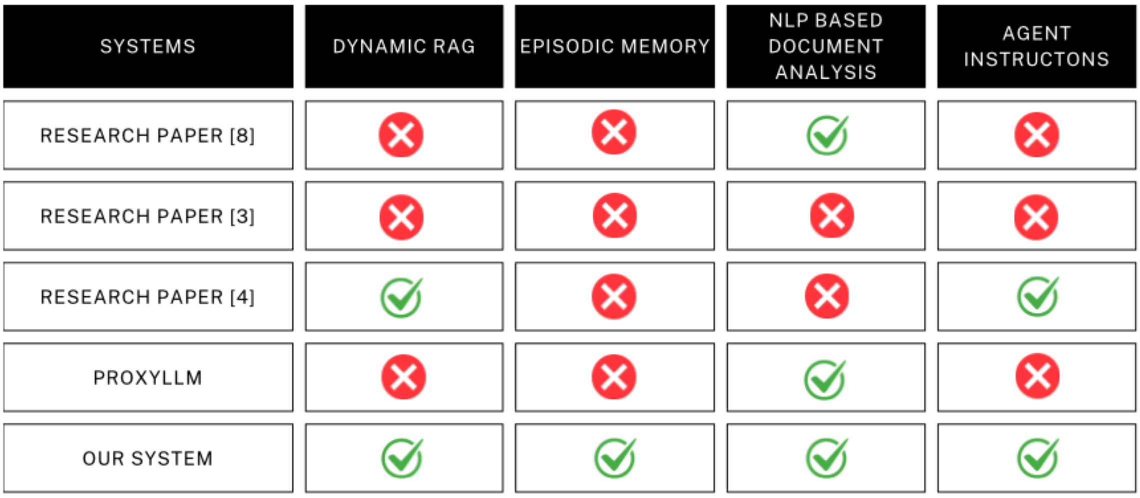
1.3 Research gap

According to the literature survey, I have identified key future implementations and research gaps related to knowledge bases. Many studies reveal common challenges in areas such as memory management, token efficiency, and effective retrieval mechanisms. One recurring issue is that most Retrieval-Augmented Generation (RAG) systems rely on static document uploads, requiring manual intervention to update content. This highlights the need for an automated system to dynamically update vector databases with new contextual information while identifying and updating or replacing similar existing content. Such a system would optimize data management and access, ensuring efficiency through advanced technologies.

Another significant challenge is the tendency of large language models (LLMs) to generate hallucinated responses to certain user queries. These responses are often unrelated to the questions posed and fail to meet user expectations, leading to dissatisfaction. Previous research on AI chatbots has primarily focused on their use in customer service, with limited exploration of their application for internal employee support in financial enterprises.

Furthermore, existing chatbot systems often rely on directly integrating OpenAI APIs and employing prompt engineering to generate responses. While this approach provides basic answers, it frequently falls short of delivering accurate and contextually relevant information. Fine-tuning LLMs for specific use cases, though critical, is often inefficient due to the high resource requirements and complexity involved. These challenges emphasize the importance of developing more effective and efficient solutions for integrating LLMs into enterprise environments to better address employee needs.

Current systems utilizing large language models (LLMs) often include the entire conversation history in prompts when generating responses, resulting in significant inefficiencies such as unnecessary token usage and increased operational costs. Despite advancements in natural language processing, there remains a critical gap in the development of efficient memory management strategies tailored for LLMs. Specifically, existing systems lack mechanisms to dynamically optimize context selection, reducing redundancy while maintaining response accuracy. Furthermore, there is a need for approaches that incorporate user feedback to evaluate satisfaction, identify successful and unsuccessful interactions, and personalize responses accordingly. Another unaddressed area involves the recognition and handling of recurring queries to minimize redundant searches and improve query retrieval efficiency. Addressing these gaps is essential for enhancing the cost-effectiveness and contextual relevance of LLM-based systems.



SYSTEMS	DYNAMIC RAG	EPISODIC MEMORY	NLP BASED DOCUMENT ANALYSIS	AGENT INSTRUCTONS
RESEARCH PAPER [8]	✗	✗	✓	✗
RESEARCH PAPER [3]	✗	✗	✗	✗
RESEARCH PAPER [4]	✓	✗	✗	✓
PROXYLLM	✗	✗	✓	✗
OUR SYSTEM	✓	✓	✓	✓

Figure 3 -Research Gap

1.4 Research Problem

Based on the identified research gaps, the research problem focuses on addressing the limitations of current Retrieval-Augmented Generation (RAG) systems and large language models (LLMs) in enterprise environments. Existing systems rely on static document uploads, requiring manual intervention to update vector databases, which hampers efficiency and scalability. Furthermore, LLMs often generate hallucinated responses that undermine user trust and satisfaction, particularly in critical domains like internal employee support for financial enterprises. Inefficiencies in memory management—such as redundant token usage and high operational costs—exacerbate these challenges.

Current solutions also lack mechanisms to dynamically optimize context selection, efficiently handle recurring queries, and incorporate user feedback to enhance personalization and system improvement. Employees often face repetitive issues and submit similar queries, leading to unnecessary delays in manual ticketing systems and requiring frequent intervention from support staff. A support chatbot leveraging an updated knowledge base can reduce this inefficiency by automatically retrieving relevant context and providing steps or suggestions to resolve user queries, thereby saving significant time.

To further improve the system, implementing episodic memory can enable the chatbot to remember unique user interactions and deliver personalized responses, addressing one of the major shortcomings of existing RAG-based systems. Additionally, some user queries may require multimodal context—for example, combining IT support and HR-related information to provide a comprehensive answer. The proposed system aims to address this by combining relevant chunks from different documents and generating a single, accurate response.

Finally, the knowledge base component can assist support agents by offering actionable instructions and context-specific data to streamline their tasks. This research aims to develop an innovative framework that enables dynamic knowledge base updates, minimizes hallucinations, optimizes token efficiency, supports multimodal context, and delivers accurate, personalized, and context-aware responses tailored to enterprise needs.

2. Objectives

2.1 main objective

The primary objective is to develop a dynamic Retrieval-Augmented Generation (RAG) system integrated with OpenAI's API to streamline IT and HR support for LOLC company. The system will leverage internal data to provide precise, context-specific guidance and step-by-step solutions for IT support issues. By optimizing token and memory efficiency through episodic memory and implementing a cost-effective knowledge base, the system will deliver personalized responses tailored to employees' past interactions. Additionally, it will utilize multiple documents to generate combined answers, ensuring precise and accurate responses to users' queries. The system will also employ multiple cloud technologies to support its deployment, ensuring scalability, reliability, and accessibility. This approach aims to enhance accuracy, reduce response times, and improve overall employee satisfaction by simplifying problem-solving and minimizing manual effort.

2.2 specific objectives

Knowledge Base Development

- Develop the knowledge base server using a microservice architecture, enabling it to function as a standalone component accessible via APIs.
- Utilize prompt engineering mechanisms to provide precise and contextually relevant answers for employees' queries.
- Generate clear, step-by-step instructions for agents to efficiently execute automation tasks, specifying the order of operations to ensure smooth task completion and optimal performance.

Developing a RAG System

- Design and implement a Retrieval-Augmented Generation (RAG) system to answer user queries based on available context in the vector database.
- Use search mechanisms to retrieve answers from multiple documents and sources, specializing the knowledge base for IT support and HR policies.

Dynamic Context Update in Vector DB

- Develop a dynamic system that allows the addition of new documents or text content via the UI, automatically embedding and saving them to the database.
- The system will automatically detect similar existing contexts in the vector database and update them with relevant, newly added context.

Implementing Episodic Memory for the RAG System

- Integrate episodic memory to leverage previous chat interactions and provide more contextually accurate answers.
- Combine previous responses with new context, improving precision and memory efficiency to reduce unnecessary token usage and enhance system performance.
- Building a pipeline based on user feedback for the response system involved identifying what worked well and what didn’t. The successful responses were logged into the database to improve the system's accuracy in providing contextual answers.
- NLP based content analysis

Utilizing Cloud Technologies for System Deployment

- Incorporate Azure cloud technologies to develop a scalable, reliable, and accessible system.
- Ensure high availability and distributed performance for the knowledge base, allowing it to meet the demands of a growing enterprise.

3. Methodology

3.1 Requirement gathering

Requirement gathering was conducted through an extensive analysis of past research carried out in recent years. The primary data was collected from LOLC's internal knowledge base and through close collaboration with the company's technical team to understand customer requirements for the application. Additional information was obtained from reliable websites related to technical support. The process also involved engaging with the supervisor, co-supervisors, and the internal research team of SLIT to gather ideas while maintaining consistent communication with the LOLC internal team.

3.2 System overview

This is the system architecture diagram for our Gen AI-based chatbot, showcasing how the various components work together to develop the complete system. The architecture includes four key components, each contributing to the overall functionality.

Key features include a **3D Avatar-based Chat** and **Form Automation** function, powered by a multi-agent system designed to streamline processes and automate system features. This ensures an enhanced and seamless customer experience for our users. Additionally, the system incorporates a **Knowledge-Based Function**, leveraging advanced technologies to provide intelligent responses and support. Together, these components form a robust, efficient, and user-friendly AI-driven chatbot architecture.

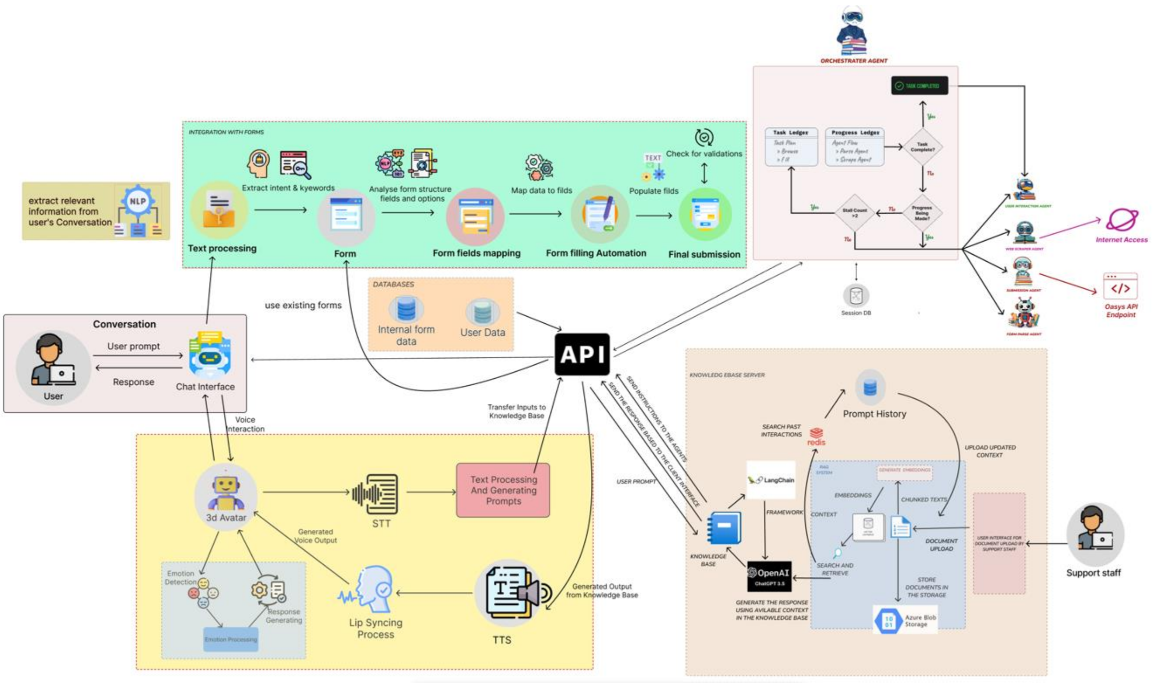


Figure 4 - Overall System Diagram

In this Gen AI-based chatbot, the most crucial aspect is the development of its knowledge base, as it plays a key role in generating precise answers for users. Many features are incorporated into this knowledge base, as illustrated in the component diagram below.

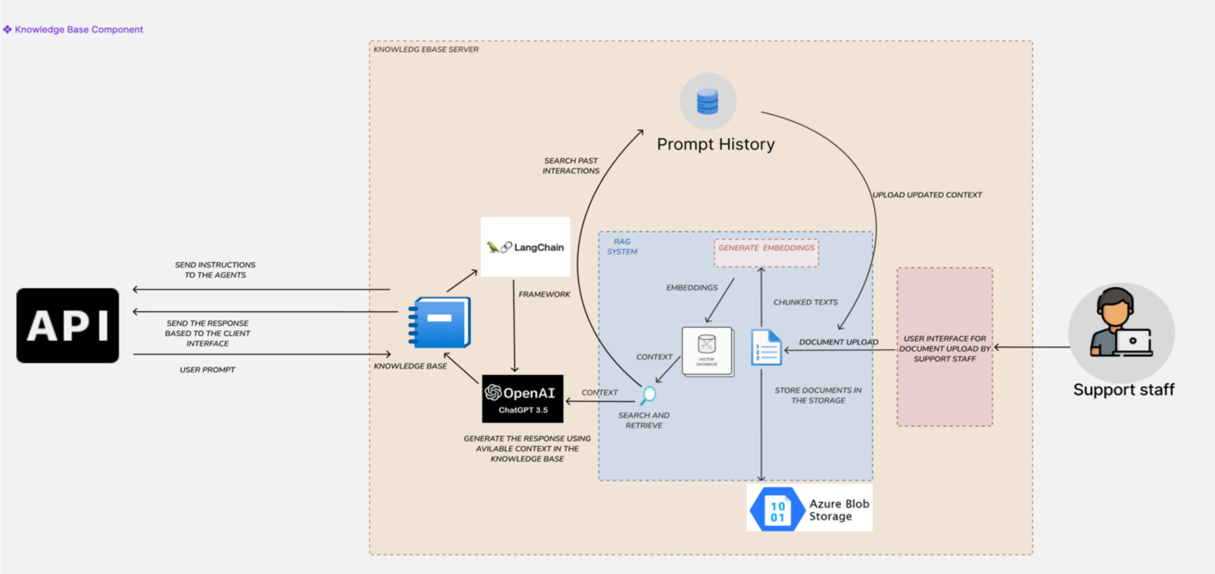


Figure 5 - Knowledgebase Architecture

The initial step in developing this knowledge base is to gather the required internal data from LOLC and clarify the system requirements. The knowledge base component will be developed using a microservice architecture due to its standalone nature and scalability features. Collected documents and data will need to be updated within a vector database, leveraging embedding text models to implement a robust search and retrieval mechanism. This will allow the system to fetch relevant data from the database and provide more accurate answers to user prompts by enhancing the context with the help of large language models (LLMs).

To ensure the knowledge base remains accurate and up-to-date, a system will be implemented to dynamically update content. This includes adding new documents or policies, as well as replacing outdated content with the latest information. A user interface (UI) will be developed for staff to manage these updates, allowing them to oversee and modify the knowledge base efficiently. The system will also identify and replace outdated information with new content, ensuring the knowledge base remains reliable and precise.

Additionally, a pipeline will be developed to save prompt history and build an episodic memory system. This system will store and identify past interactions, including keywords, timestamps, prompts, and user feedback. This historical data will enable the generation of personalized answers based on users' previous interactions. User feedback will also help identify which answers were effective and which were not. Successful prompts and answers will be updated in the database, ensuring the system can provide accurate and refined answers for similar queries in the future.

The document management system will utilize a cloud environment, with the entire system deployed in Microsoft Azure using Azure technologies. OpenAI’s APIs will be integrated to enhance the chatbot’s ability to provide contextually accurate answers.

Furthermore, the system will generate step-by-step instructions for agents to take actions based on the user’s query and the identified context. When users ask questions through the UI, the knowledge base will categorize the query (e.g., whether it is related to support) and format the answer appropriately using prompt engineering techniques. This ensures the system delivers accurate responses tailored to the user’s specific needs.

3.3 Tools and technologies

Tool/Technology	Purpose/Use
OpenAI GPT-3.5/4	Used for natural language understanding and generating responses in the chatbot based on user queries.
Langchain	Framework used to build applications with LLMs (like GPT-3.5/4o) that integrate with various data sources.
ChromaDB	A vector database used for storing and retrieving context-specific embeddings, enabling efficient retrieval.
MongoDB	NoSQL database for storing and managing data such as user queries, system responses, and knowledge base documents.
Flask	Python-based web framework used for developing and hosting the backend API for the chatbot application.
Azure	Cloud platform for hosting the application, storing data, and providing computing resources for the system.
Huggingface	Provides pre-trained AI models and tools for natural language processing tasks, integrated into the chatbot system.
Redis	To store the memory in cache for faster data retrieval

Figure 6 -Technologies

4. Project requirements

4.1 Functional requirements

Answer User Queries

- Provide accurate and context-specific answers to user queries related to IT support, HR policies, and business operations.

Suggestions and Resolution Paths

- Offer actionable suggestions and detailed step-by-step guidance to resolve IT support issues effectively.

Explanations and Guidance

- Deliver clear explanations and comprehensive guidance on HR-related queries and business support tasks.

Dynamic Knowledge Base Management

- Develop an intuitive and user-friendly UI that allows staff to upload, update, and manage documents and information, ensuring the knowledge base remains up-to-date and accurate.

Agent Instruction Generation

- Automatically generate instructions and tasks for agent-based automation systems, enabling them to execute repetitive and predefined workflows efficiently.

Personalized Responses

- Adapt responses based on chat history, preferences, and previous interactions to deliver a customized and engaging support experience.

4.2 Non-Functional Requirements

Performance

- The chatbot must provide responses within 2 seconds to ensure seamless user interaction.
- The system should handle multiple user request at the same time.

Scalability

- The system must be scalable to support increased user demand and future expansion in features and functionalities.

Reliability

- Ensure 99.9% uptime for the Knowledgebase and related services to guarantee availability.
- The system should have mechanisms for error detection and recovery to minimize downtime.(Logger)

Security

- All data transmissions must be encrypted (e.g., using HTTPS and TLS) to protect user and company data.
- Implement strict access controls to ensure only authorized staff can manage and update the knowledge base.
- Sensitive information, such as HR and business data, must be securely stored in secure manner

Maintainability

- The system should be modular and built using a microservice architecture to allow easier updates, debugging, and maintenance.
- Provide comprehensive documentation for the system, including APIs, workflows, and knowledge base management processes.

Usability

- The Document upload UI must be intuitive and user-friendly for both employees and administrative staff.
- Voice and text interactions should be clear, natural, and accessible to users of varying technical expertise.

Accessibility

- The system must be accessible to all the employees that working in diffetrent branches and office premises.

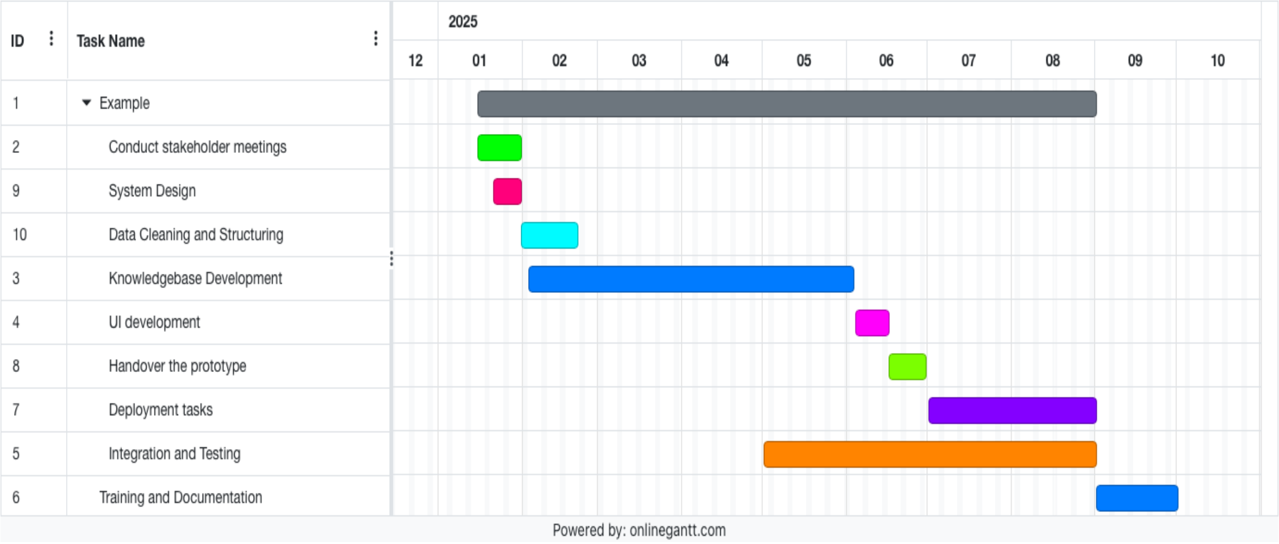
Adaptability

- The chatbot must adapt to changes in the knowledge base dynamically without requiring a restart or downtime.
- It should support integration with the LOLC's current system

Compliance

- The system must comply with all organizational and industry standards, including IT support best practices, data protection laws, and corporate policies.

5.Grantt Chart



ID	Task Name
1	▾ Example
2	Conduct stakeholder meetings
9	System Design
10	Data Cleaning and Structuring
3	Knowledgebase Development
4	UI development
8	Handover the prototype
7	Deployment tasks
5	Integration and Testing
6	Training and Documentation

Figure 7 - Gantt chart

6. Work Breakdown Chart

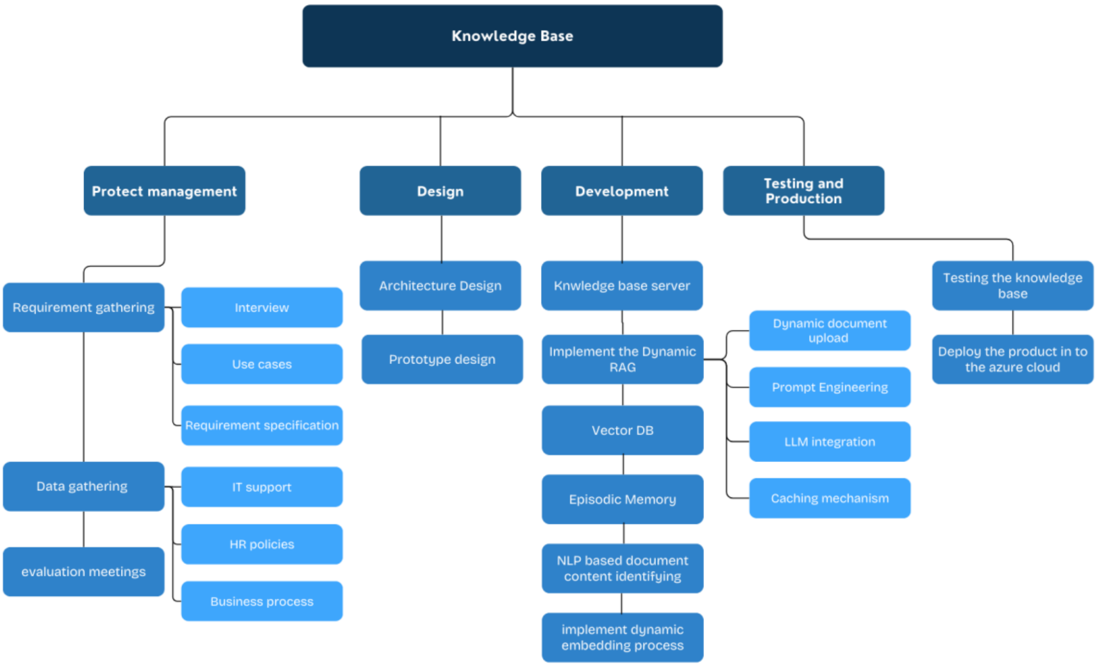


Figure 8 - work breakdown chart

Reference

[1] Cuconasu, Florin, et al. "The power of noise: Redefining retrieval for rag systems." Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024.

[2] Petroni, Fabio, et al. "IR-RAG@ SIGIR24: Information retrieval's role in RAG systems." Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024.

[3] Yuan, Ye, et al. "A Hybrid RAG System with Comprehensive Enhancement on Complex Reasoning." arXiv preprint arXiv:2408.05141 (2024).

[4] Jeong, Cheonsu. "A Study on the Implementation Method of an Agent-Based Advanced RAG System Using Graph." arXiv preprint arXiv:2407.19994 (2024).

[5] Zhao, Shengming, et al. "Towards understanding retrieval accuracy and prompt quality in RAG systems." arXiv preprint arXiv:2411.19463 (2024).

[6] Xia, Peng, et al. "Mmed-rag: Versatile multimodal rag system for medical vision language models." arXiv preprint arXiv:2410.13085 (2024).

[7] Exploring the role of large language model (LLM)-based chatbots for human resources

[8] Gong, Zhiyun, et al. "Enhancing Trust in LLM Chatbots for Workplace Support Through User Experience Design and Prompt Engineering." The Human Side of Service Engineering(2024)

[9] Oluwagbade, Elizabeth. "Conversational AI as the New Employee Liaison: LLM-Powered Chatbots in Enhancing Workplace Collaboration and Inclusion." (2024).

[10] Finsås, Mats, and Joachim Maksim. Optimizing RAG Systems for Technical Support with LLM-based Relevance Feedback and Multi-Agent Patterns. MS thesis. NTNU, 2024.

[11] Cederlund, Oscar, Sadi Alawadi, and Feras M. Awaysheh. "LLMRAG: An Optimized Digital Support Service using LLM and Retrieval-Augmented Generation." 2024 9th International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, 2024.

[12] Agrawal, Garima, Sashank Gummuluri, and Cosimo Spera. "Beyond-RAG: Question Identification and Answer Generation in Real-Time Conversations." arXiv preprint arXiv:2410.10136(2024).

[13] Xu, Zhentao, et al. "Retrieval-augmented generation with knowledge graphs for customer service question answering." Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024.

Appendices