PERSONALIZED ACADEMIC INTERVENTIONS USING ADAPTIVE AND EXPLAINABLE AI: MULTI-MODAL LEARNING ANALYTICS FRAMEWORK

Dammulla Liyanage Kaveesha Nimanji

(IT22365750)

B.Sc. (Hons) Degree in Information Technology

Specializing in Information Technology

Department of information technology

Sri Lanka Institute of Information Technology

Sri Lanka

August 2025

CONVERSATIONAL AI ASSISTANT FOR ADAPTIVE ACADEMIC INTERVENTIONS

Dammulla Liyanage Kaveesha Nimanji

(IT22365750)

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Hons) in Information Technology Specializing in Information Technology

Department of information technology

Sri Lanka Institute of Information Technology

Sri Lanka

August 2025

DECLARATION PAGE OF THE CANDIDATE & SUPERVISOR

We declare that this is our 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 published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Nimanji D.L.K	IT22365750	Nimanji

The above candidates are carrying out research for the undergraduate dissertation under my supervision.

2026/04/26

Signature of the supervisor — Date

(Ms. Sanjeevi Chandrasekara)

ABSTRACT

Artificial Intelligence (AI) is being used in educational settings as a result of the growing need for intelligent academic assistance systems. Learning Management Systems (LMS) and static university websites are examples of traditional academic information access techniques that are frequently ineffective, time-consuming, and lack personalization. In order to improve student support through a clever, scalable, and user-centered method, this study suggests a Conversational AI Assistant for Adaptive Academic Interventions.

To guarantee precise and thorough results, the suggested system makes use of a hybrid architecture that combines Retrieval-Augmented Generation (RAG) and Web Search Fallback. Optical Character Recognition (OCR), text chunking, and embedding techniques are used to process academic texts, including university norms, regulations, and administrative resources, in order to create a semantic knowledge base that is stored in a vector database. When a user enters a query, the system uses a Large Language Model (LLM) to create context-aware replies and uses similarity search to find pertinent material. To guarantee constant response availability, the system dynamically changes to web-based retrieval whenever pertinent data cannot be located.

Additionally, the chatbot enhances accessibility and usability by supporting multimodal interaction, which includes text, speech, and image inputs. Student performance data is used to construct a personalization module that allows the system to provide motivational comments, proactive reminders, and tailored recommendations. By offering justification for system responses, Explainable AI (XAI) integration improves transparency. Additionally, using a local LLM through Ollama allows for offline functionality and guarantees data privacy.

Experimental results demonstrate that the proposed system improves response accuracy, user engagement, and system reliability compared to traditional chatbot systems. The system also reduces dependency on manual academic support processes and enhances the overall student experience.

Keywords: Conversational AI, RAG, Personalization, Multimodal Interaction, Explainable AI

ACKNOWLEDGEMENT

First and foremost, I would like to express my profound gratitude to my supervisor, Ms. Sanjeevi Chandrasekara, whose unwavering guidance, mentorship, and unflagging support were instrumental in every phase of my undergraduate research. Her expertise, patience, and commitment to my academic growth not only enriched the quality of this research but also significantly contributed to my personal and professional development. Additionally, I extend my heartfelt thanks to Ms. Ishara Weeratunga, the co-supervisor of this research project, whose willingness to assist and share insights proved invaluable throughout this journey. Their combined wisdom and dedication have not only made this research possible but have also instilled in me a profound appreciation for the pursuit of knowledge and the spirit of collaboration. I am profoundly grateful for their support, and I look forward to applying the invaluable lessons learned from them in my future academic and professional endeavors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Description**
AI	Artificial Intelligence
NLP	Natural Language Processing
RAG	Retrieval-Augmented Generation
LLM	Large Language Model
OCR	Optical Character Recognition
LMS	Learning Management System

1. INTRODUCTION

Higher education students typically experience challenges with accessing credible academic information and understanding institutional communications, notably on registrations, deadlines, payments, and exam procedures. Traditional Learning Management Systems (LMS) provide static interfaces of low interactivity, which compel students to sift through enormous amounts of scattered information [1]. Research on educational technology has proven that conversational interfaces can facilitate increasing accessibility, engagement, and learning through enabling students to interact in natural language [2]. Numerous research works have explored the use of chatbots in academic support. For instance, Winkler and Söllner [3] indicated the possibilities of education chatbots to support student motivation, while Tegos et al. [4] outlined how conversational agents enable collaborative learning. However, most existing academic chatbots merely reply to Frequently Asked Questions (FAQs) or provide general administrative guidance. They are less flexible, do not handle multiple input modes, and are unable to explain complex academic interventions in simple terms. To achieve this project, one needs to learn expertise in Natural Language Processing (NLP), intent recognition, and context-aware dialog management. Cutting-edge student question comprehension is achievable using sophisticated models such as transformer-based architectures (e.g., BERT, GPT) [5]. Multi-modal AI techniques also need to be applied for handling non-text inputs such as voice, images, and PDFs. The state of the art in chatbot AI takes advantage of deep learning combined with adaptive learning systems, but their use in universities remains immature, especially in academic interventions. Off-the-shelf solutions rarely provide personalized guidance on repeat -related notifications in a simplified manner. Our approach builds existing work by integrating conversational AI and adaptive academic intervention strategies so that students have access to not only facts, but also customized, simplified explanations. Unlike existing chatbots, the platform supports multi-modal inputs and delivers context-dependent interventions, making it more inclusive and effective for diverse student needs.

1.1 Background and literature survey

Academic institutions have increasingly adopted digital platforms such as Learning Management Systems (LMS) to manage educational resources, course materials, and student interactions. These platforms play a crucial role in modern education by providing centralized access to lecture notes, assignments, announcements, and communication tools. However, despite their widespread adoption, LMS platforms often require users to manually search for information through multiple pages, documents, and menus. This process can be time-consuming, inefficient, and frustrating, especially when students need quick answers to academic queries. Additionally, traditional academic support methods lack interactivity and real-time assistance, forcing students to rely on static content or delayed responses from lecturers and administrative staff. As a result, there is a growing need for more intelligent and responsive systems that can provide instant and user-friendly academic support.

Recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have introduced chatbot-based systems as a promising solution to address these challenges. Chatbots enable users to interact with systems using natural language, making information retrieval faster and more intuitive. Early chatbot systems were primarily rule-based, relying on predefined patterns and scripted responses. While these systems were useful for handling simple queries, they lacked the ability to understand complex language structures and context, resulting in limited accuracy and flexibility. With the evolution of machine learning and deep learning techniques, modern chatbots have become significantly more advanced. In particular, transformer-based models have revolutionized natural language understanding and generation, allowing chatbots to produce human-like responses and handle a wide range of queries more effectively.

One of the most significant advancements in this field is the introduction of Retrieval-Augmented Generation (RAG). RAG combines the strengths of information retrieval systems and generative language models to improve response accuracy and reliability. Instead of relying solely on the internal knowledge of a language model, RAG retrieves relevant information from external data sources, such as document databases, and uses this information to generate context-aware responses. This approach ensures that the chatbot provides more accurate and up-to-date information, particularly in domain-specific applications such as academic environments. By grounding responses in actual documents, RAG reduces the risk of generating incorrect or misleading information.

However, despite these improvements, many existing chatbot systems still have notable limitations. A significant number of systems rely solely on static knowledge bases, which restricts their ability to handle queries that fall outside the predefined domain. When users ask questions that are not covered in the internal database, these systems often fail to provide meaningful responses. Furthermore, most existing solutions lack personalization features, meaning they treat all users in the same way without considering individual needs or performance levels. This limits their effectiveness in providing targeted academic support. Additionally, many chatbots do not support proactive assistance, where the system can anticipate user needs and provide recommendations or reminders without being explicitly asked.

Another limitation is the lack of multimodal interaction capabilities. Most traditional chatbots are restricted to text-based communication, which can reduce accessibility and usability for users who prefer voice interaction or need to upload images. Finally, many modern chatbot systems rely heavily on cloud-based infrastructure, raising concerns about data privacy, security, and system reliability. These challenges highlight the need for a more advanced and adaptive chatbot system that integrates hybrid retrieval mechanisms, personalization, multimodal interaction, and privacy-preserving technologies to provide a more effective and user-centered academic support solution.

.

1.2 Research Gap

Despite the significant advancements in chatbot technologies, several critical limitations still exist in current academic support systems. Most existing chatbots rely on a single knowledge source and lack hybrid retrieval mechanisms, making them ineffective when handling both domain-specific academic queries and general knowledge questions. Additionally, these systems offer minimal personalization, as they do not utilize student performance data to provide tailored recommendations or feedback. Another major limitation is the absence of proactive academic interventions, where chatbots fail to predict and assist students before issues arise. Furthermore, many systems do not support multimodal interaction, restricting user communication to text only and reducing accessibility. There is also limited integration of Explainable AI (XAI), which results in a lack of transparency and reduced user trust in the system's responses. Moreover, the heavy dependence on cloud-based models raises concerns regarding data privacy, security, and system reliability. These gaps highlight the necessity for developing a more advanced, adaptive, and intelligent chatbot system that can address these challenges effectively.

Table 1 Comparison of the existing research methods and proposed method

Technologies and methods	**Research Paper [1]**	**Research Paper [2]**	**Research Paper [3]**	**Research Paper [4]**	**Research Paper [5]**	**Proposed research**
Multi-modal Input Handling (Text, Voice, Image, PDF)	✗	✗	✗	✓	✗	✓
Explainability of Academic Processes	✗	✓	✓	✗	✗	✓
LMS Integration for Academic Data	✗	✓	✗	✗	✓	✓
Privacy-Preserving Student Data Handling	✗	✗	✓	✗	✗	✓
Intent Recognition & Entity Extraction	✓	✓	✓	✗	✓	✓

1.3 Research problem

Most traditional academic support systems, like Learning Management Systems (LMS) and institutional portals, depend on manual search and static content delivery. Students have to go through a lot of pages, documents, and menus to find important information like academic rules, due dates, or information about their teachers. This process can take a long time, not work well, and be very frustrating, especially when users don't know where to look or how to ask their questions. Also, these systems don't allow for interaction or help in real time, which means they can't change based on what a student needs or give immediate clarification. This makes the overall student experience worse.

Although chatbot technologies have been introduced to address some of these limitations, many existing solutions are still inadequate. Traditional chatbots are often rule-based or rely on limited knowledge bases, restricting their ability to understand complex natural language queries or handle questions outside predefined domains. They also lack personalization, meaning they provide the same generic responses to all users regardless of individual needs, performance levels, or academic context. Additionally, most systems do not offer proactive support, such as reminders or recommendations, which are essential for improving student engagement and academic success. Another critical limitation is the lack of transparency in AI-generated responses, as users are often not provided with explanations on how or why a particular answer was generated. Moreover, many chatbot systems depend heavily on cloud-based services, raising concerns regarding data privacy and security, especially when handling sensitive student information.

To address these challenges, this research focuses on developing an intelligent Conversational AI Assistant that overcomes the limitations of existing systems. The proposed solution integrates advanced technologies such as Retrieval-Augmented Generation (RAG), web search fallback, and personalization mechanisms to provide accurate, context-aware, and comprehensive academic support. The system supports multimodal interaction, allowing users to communicate through text, voice, and image inputs, thereby improving accessibility and usability. It also incorporates proactive and action-based features, enabling the chatbot to provide reminders, recommendations, and direct navigation to resources. Furthermore, Explainable AI (XAI) techniques are integrated to enhance transparency and user trust, while the use of a locally deployed model ensures data privacy and system reliability. Overall, the research aims to create a more adaptive, efficient, and user-centered academic support system that significantly improves how students access and interact with academic information.

1.4 Research Objectives

1.4.1 Main Objective

To create an intelligent conversational AI assistant that improves academic support services by combining hybrid knowledge retrieval, personalization, and adaptive intervention mechanisms into a single system. The system's goal is to give students accurate, real-time, and context-aware answers to their questions by combining Retrieval-Augmented Generation (RAG) with ways to access external knowledge. This will make sure that the answers are both reliable and complete.

Also, the system wants to help students learn better and be more involved by giving them personalized academic help based on their behavior, interaction history, and performance data. It has proactive support features like notifications, reminders, and personalized suggestions to make learning continuous and guided.

The goal is also to make it possible for users to interact with the system in more than one way, such as through text, voice, and documents. This will make it easier for different types of learners to use and access the system. It is also important for the system to use AI techniques that protect privacy in order to safely handle sensitive academic data while still following data protection rules.

Overall, the goal is to create a scalable, intelligent, and user-centered academic support system that enhances the efficiency, effectiveness, and inclusiveness of modern educational environments.

1.4.2 Specific Objectives

To implement a hybrid RAG-based chatbot system

The system's goal is to create a chatbot by combining a web search fallback mechanism with a hybrid Retrieval-Augmented Generation (RAG) architecture. By using this method, the chatbot can obtain precise and pertinent information from internal academic texts, like guidelines, norms, and regulations of the university. The system dynamically changes to external web search to deliver pertinent responses when the necessary information is not found in the internal knowledge base. Compared to conventional chatbot systems that just use static data sources, this hybrid technique guarantees greater answer accuracy, enhanced coverage, and robustness.

To support multimodal interaction (text, voice, and image)

To improve accessibility and usability, the chatbot is built to accommodate a variety of input ways, such as text, voice, and image-based queries. Natural typing, speech recognition, and the uploading of photos with text-based academic content are the ways in which users can engage with the system. The system can accommodate a variety of user preferences thanks to its multimodal capacity, which also enhances the user experience in general. Furthermore, input validation procedures guarantee that only pertinent and significant inputs are handled, improving system dependability.

To develop a personalization mechanism using student performance data

Using student performance data kept in a database to add personalization is one of the system's main goals. To categorize students into various performance levels, the chatbot examines data including attendance, assignment scores, and exam results. The algorithm creates customized suggestions, reminders, and encouraging comments based on this classification. Because of this individualized approach, the chatbot is able to offer focused academic support, which helps students stay on course and perform better.

To enable proactive and action-based chatbot features

The suggested solution adds proactive and action-based features, in contrast to conventional chatbots that merely reply to user inquiries. Based on user data, such as poor performance or missed deadlines, the chatbot can automatically create alerts, notifications, and reminders. It also offers task-based functionality such using conversational instructions to download papers directly, access academic resources, and go to dashboards. As a result, the chatbot is no longer a passive source of information but rather an active aide.

To integrate Explainable AI (XAI) mechanisms

Explainable AI (XAI) approaches are integrated into the system to enhance user confidence and transparency. The chatbot provides explanations for its recommendations and responses rather than just definitive answers. For instance, the system emphasizes the particular performance indicators that drove a student's improvement suggestions. Users are better able to comprehend how decisions are made thanks to this transparency, which also boosts system trust.

To ensure data privacy and offline capability

The solution reduces reliance on external cloud-based services by using a local Large Language Model (LLM) to handle privacy concerns. This guarantees that private student information is handled safely within the system and isn't exposed to outside platforms. Furthermore, offline functioning is made possible by the use of a local model, which enables the chatbot to work even in settings with spotty or nonexistent internet connectivity. As a result, the system is more dependable and more suited for practical academic applications.

To enable continuous system improvement through user feedback

The system includes a feedback collection mechanism that allows users to provide input on chatbot responses and overall performance. This feedback is stored and analyzed to identify areas for improvement. By continuously learning from user interactions, the system can enhance response accuracy, improve user experience, and adapt to evolving user needs. This ensures that the chatbot remains effective and relevant over time.

2. METHODOLOGY

2.1 Methodology

The proposed system follows a modular and layered architecture to ensure scalability, flexibility, and efficient processing of user queries. The overall workflow is divided into several stages, starting from data collection to response generation and personalization. This structured approach allows the system to integrate multiple technologies such as OCR, vector databases, and large language models to deliver accurate and context-aware responses.

Step 1: Data Collection

In the initial stage, relevant academic data is collected from institutional sources. This includes university documents such as rules and regulations, lecturer details, and administrative information, along with student performance data. These datasets form the foundation of the chatbot's knowledge base and personalization mechanism, ensuring that the system can provide both general and user-specific responses.

Step 2: Data Processing

The collected documents, especially PDFs, are processed using Optical Character Recognition (OCR) through Tesseract to extract readable text. The extracted text is then divided into smaller segments using text chunking techniques. This improves retrieval accuracy by allowing the system to match user queries with more precise portions of the content rather than entire documents.

Step 3: Embedding Generation

After preprocessing, the text data is converted into vector representations known as embeddings using a machine learning embedding model. These embeddings capture the semantic meaning of the text, enabling the system to perform similarity-based searches rather than relying on simple keyword matching.

Step 4: Vector Storage

The generated embeddings are stored in a vector database (ChromaDB). This database is optimized for fast similarity searches, allowing the system to quickly identify the most relevant document chunks when a query is received.

Step 5: Query Processing

When a user submits a query, it is first converted into an embedding using the same embedding model. The system then performs a similarity search in the vector database to find the most relevant document segments that match the user's query.

Step 6: Decision Logic

At this stage, the system evaluates the similarity score between the query and retrieved data. If the similarity is above a predefined threshold, the system proceeds with a RAG-based response using internal data. If not, the system activates the web search fallback mechanism to retrieve relevant information from external sources, ensuring continuous response availability.

Step 7: Response Generation

The retrieved information, along with the user query, is passed to a Large Language Model (LLM). The model generates a coherent, context-aware, and human-like response, which is then delivered to the user through the chatbot interface.

Step 8: Personalization

Finally, the system analyzes student performance data stored in Firebase. Based on predefined thresholds, students are classified into performance categories, and personalized recommendations, reminders, and motivational messages are generated. This step enhances the system's ability to provide adaptive and user-specific support.

2.1.1 Materials and Methods

The development of the Conversational AI Assistant follows a structured methodology that integrates data collection, intelligent processing, and system design principles. The system is built using a hybrid architecture that combines Retrieval-Augmented Generation (RAG), web search fallback, and personalization mechanisms. Modern web and AI technologies such as React, Node.js, Firebase, and Python-based models are used to implement different layers of the system. The methodology focuses on ensuring accurate information retrieval, efficient response generation, and enhanced user interaction through multimodal capabilities. Additionally, privacy-preserving techniques and Explainable AI (XAI) mechanisms are incorporated to improve system transparency and reliability.

2.1.2 Problem Statement

Existing academic support systems, such as Learning Management Systems (LMS), require users to manually search for information, making the process time-consuming and inefficient. Traditional chatbot systems are often limited to rule-based responses or static knowledge bases, which restrict their ability to handle complex queries and out-of-domain questions. Furthermore, these systems lack personalization, proactive assistance, and multimodal interaction capabilities, reducing their effectiveness in supporting students.

Additionally, many existing solutions rely heavily on cloud-based models, raising concerns related to data privacy and security, especially when handling sensitive student information. There is also a lack of transparency in AI-generated responses, which reduces user trust. Therefore, there is a need for an intelligent, adaptive, and secure conversational system that can provide accurate, personalized, and real-time academic support while ensuring privacy and explainability.

2.1.3 Component System Architecture (Solution Design)

The proposed system is designed using a layered and modular architecture to ensure scalability, flexibility, and efficient data flow. The architecture consists of several key components:

The User Interaction Layer allows users to interact with the chatbot through a web-based interface supporting text, voice, and image inputs. The Frontend Layer, developed using React, manages user input handling and response display.

The Backend Layer, implemented using Node.js and Firebase, processes user requests, manages APIs, and handles communication between system components. It also retrieves student data and manages document storage.

The AI Processing Layer forms the core of the system, where the RAG mechanism retrieves relevant document chunks from the vector database (ChromaDB) using similarity search. These retrieved results are passed to a Large Language Model (LLM) to generate context-aware responses. If no relevant data is found, the system activates the web search fallback mechanism.

The Data Layer includes academic documents (PDFs), student performance data stored in Firebase, and vector embeddings stored in ChromaDB. Finally, the

Personalization Layer provides adaptive recommendations, reminders, and action-based features based on user data.

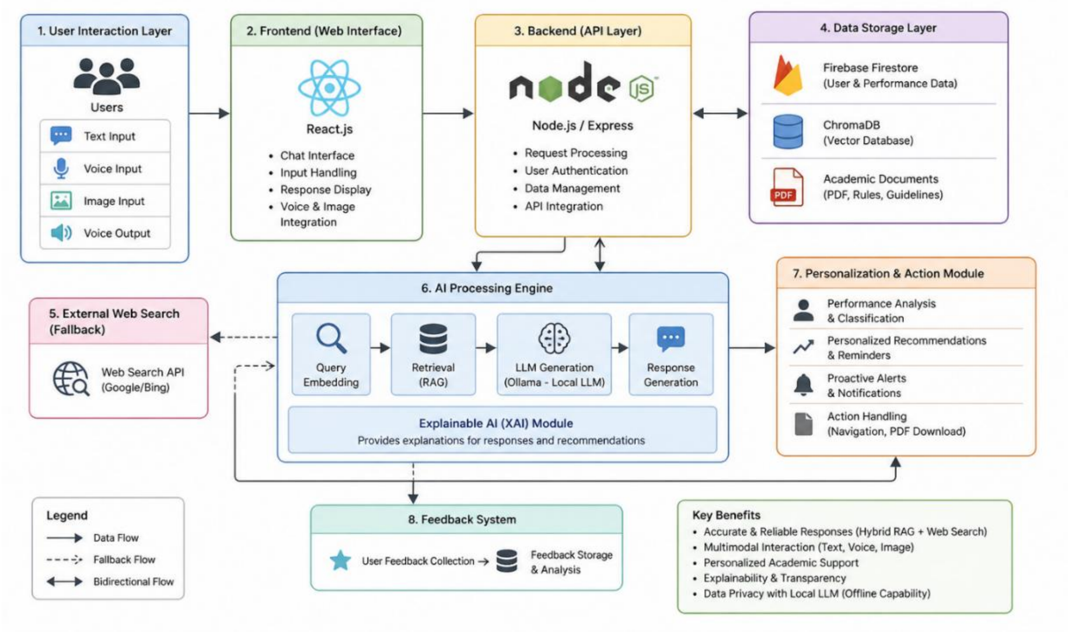

Figure 1 Component Overview Diagram

2.1.4 Data Acquisition and Processing

The system relies on multiple data sources, including academic documents, student performance data, and user interactions. Academic data such as university rules, regulations, lecturer details, and guidelines are collected from institutional sources. These documents are primarily in PDF format and require preprocessing before use.

Text extraction is performed using OCR tools such as Tesseract to convert scanned documents into machine-readable text. The extracted text is then cleaned and preprocessed to remove noise and irrelevant content. This processed data is further divided into smaller segments, known as text chunks, to improve retrieval accuracy.

Student performance data is collected from the database and includes metrics such as attendance, assignment scores, and exam results. This data is used for personalization and classification purposes. All processed data is stored securely in databases for efficient retrieval and analysis.

2.1.5 Dataset Preparation

The dataset preparation process involves transforming raw data into a structured format suitable for AI processing. Academic documents are segmented into smaller chunks and converted into vector embeddings using an embedding model. These embeddings represent the semantic meaning of the text and are stored in a vector database (ChromaDB).

For personalization, student performance data is normalized and structured into predefined attributes such as assignment scores, attendance percentages, and exam results. Threshold-based classification is applied to categorize students into performance levels. This structured dataset enables the system to generate accurate recommendations and personalized feedback.

2.1.6 Retrieval-Augmented Generation (RAG) Architecture

The proposed system utilizes a Retrieval-Augmented Generation (RAG) architecture to enhance the accuracy and relevance of chatbot responses. RAG combines two key components: information retrieval and natural language generation. Instead of relying solely on a pre-trained language model, the system first retrieves relevant information from a knowledge base and then uses a Large Language Model (LLM) to generate a context-aware response.

In this approach, academic documents such as university rules, regulations, and administrative resources are preprocessed and converted into vector embeddings using an embedding model. These embeddings are stored in a vector database (ChromaDB), enabling efficient semantic similarity search. When a user submits a query, it is also converted into an embedding, and the system performs a similarity search to identify the most relevant document chunks. This allows the system to understand the semantic meaning of the query rather than relying on keyword matching.

Once the relevant information is retrieved, it is passed along with the user query to the LLM. The model then generates a coherent and contextually accurate response based on both the retrieved data and its pre-trained knowledge. This significantly improves response accuracy and ensures that the chatbot provides reliable, document-grounded answers.

Additionally, the system incorporates a hybrid mechanism where, if the similarity score does not meet a predefined threshold, the chatbot activates a web search fallback. This ensures that the system can handle queries beyond the internal knowledge base, thereby improving overall response coverage and robustness.

Overall, the RAG architecture enables the chatbot to deliver accurate, context-aware, and reliable academic support by combining structured document retrieval with advanced language generation capabilities.

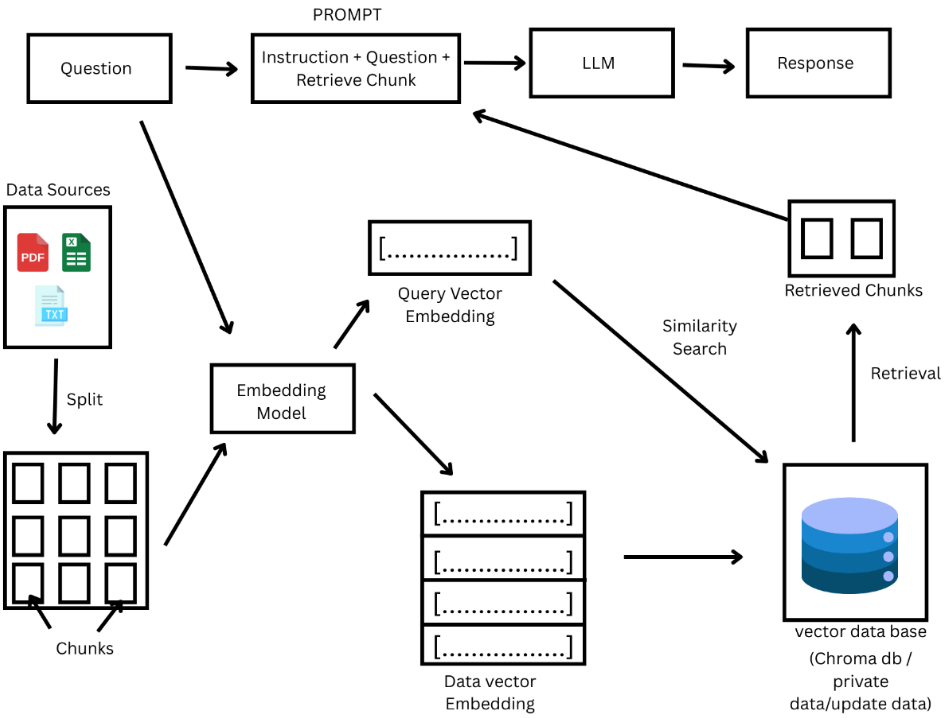

Figure 2 How to work RAG Architecture

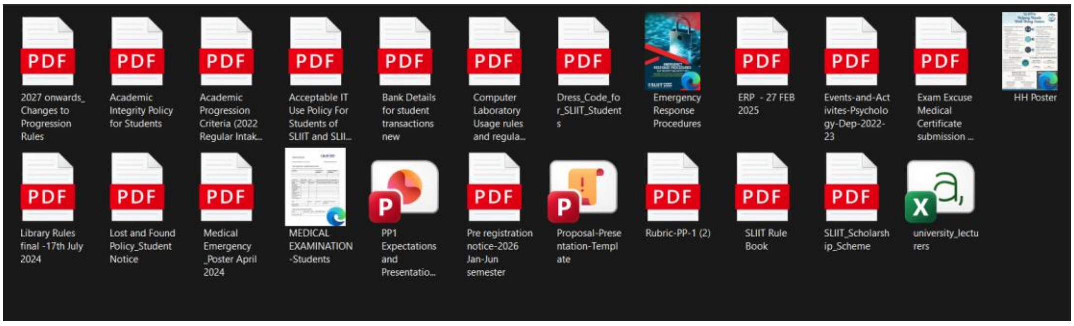

Figure 3 Screen Shot of Data Set

2.2 Project Requirements

2.2.1 Functional Requirements

The functional requirements define the essential operations that the Conversational AI Assistant must perform to effectively support users in an academic environment.

The system enables users to interact with the chatbot using multiple input modalities, including text, voice, and image-based queries. It shall process user queries and generate accurate responses by utilizing a Retrieval-Augmented Generation (RAG) mechanism, which retrieves relevant information from a structured academic document knowledge base. If the requested information is not available within the internal database, the system shall automatically activate a web search fallback mechanism to retrieve and summarize relevant external information, ensuring continuous and reliable response generation.

Furthermore, the system should provide personalized recommendations and academic reminders based on student performance data stored in the database. It shall analyze performance metrics and classify students into different categories (e.g., high, medium, and low performance), generating appropriate feedback, suggestions, and motivational messages accordingly.

The chatbot shall also support action-based functionalities, allowing users to perform tasks such as navigating to dashboards, accessing academic resources, and downloading relevant PDF documents directly through conversational commands. In addition, administrators shall be able to upload, manage, and update academic documents, including rules, regulations, and institutional guidelines, ensuring that the knowledge base remains up to date.

The system shall include a voice output feature, enabling users to listen to chatbot responses for improved accessibility. Additionally, a feedback collection mechanism shall be implemented to gather user feedback, which can be used to continuously improve system performance, accuracy, and user experience.

.

2.2.2 Non-Functional Requirements

The non-functional requirements define the quality attributes, performance standards, and operational constraints of the system.

The system should ensure high performance and responsiveness, providing real-time or near-time responses to user queries. It should maintain high accuracy in both information retrieval and response generation through the use of semantic search techniques and advanced AI models. The system must be scalable, allowing it to support multiple concurrent users without significant performance degradation.

Security and privacy are critical considerations; therefore, the system shall ensure secure data storage and processing, particularly for sensitive student performance data. The use of a local Large Language Model (LLM) shall minimize reliance on external cloud-based services, thereby enhancing data privacy and confidentiality. Additionally, the system should support offline or low-connectivity operation, ensuring availability even in environments with limited internet access.

The chatbot interface shall be user-friendly, intuitive, and accessible, supporting multimodal interaction to improve usability for a diverse range of users. The system should also ensure high reliability and fault tolerance, maintain stable operation and minimize system failures or downtime. Furthermore, it should be designed for maintainability, allowing easy updates, debugging, and future enhancements.

2.2.3 Other Requirements

Other requirements include system constraints, implementation considerations, and additional factors that support the overall effectiveness of the chatbot.

The system shall be developed using modern and scalable technologies, including React for the frontend, Node.js and Firebase for backend services, and Python-based AI models for intelligent processing. It shall integrate with a vector database (ChromaDB) to enable efficient semantic search and utilize Optical Character Recognition (OCR) tools such as Tesseract to extract text from PDF documents and images.

The system must adhere to ethical and regulatory standards, ensuring that sensitive personal and academic data is not exposed, misused, or improperly shared. Proper data-handling practices and access controls should be implemented to maintain user trust and compliance.

Additionally, the system shall support extensibility and future scalability, allowing integration with external systems such as Learning Management Systems (LMS), advanced AI models, or additional data sources. This ensures that the chatbot can evolve over time and adapt to changing academic and technological requirements.

2.3 Commercialization Aspects of the Product

The proposed Conversational AI Assistant demonstrates significant potential for commercialization within the rapidly growing Education Technology (EdTech) sector, which is experiencing strong global demand due to the increasing adoption of digital learning environments, hybrid education models, and AI-driven student support systems. Educational institutions are actively seeking scalable and cost-efficient solutions to improve student engagement and reduce administrative burdens, making this system highly relevant in the current market landscape.

The system can be deployed across universities, colleges, and other educational institutions as an intelligent academic support assistant that operates continuously, providing 24/7 assistance to students. This always-available nature adds significant commercial value, as it ensures uninterrupted access to academic information regardless of time zones, staff availability, or institutional working hours. In large institutions with high student populations, this capability directly addresses the scalability challenges faced by traditional support services.

By leveraging a Software-as-a-Service (SaaS) model, institutions can subscribe to the platform rather than invest in costly on-premise infrastructure or custom-built systems. This subscription-based approach allows flexible pricing strategies, such as tiered plans based on the number of users, feature access, or analytics capabilities. It also enables predictable recurring revenue for the service provider while offering institutions lower upfront costs and reduced maintenance responsibilities. Additionally, SaaS deployment ensures centralized system management, continuous feature upgrades, security enhancements, and AI model improvements without requiring manual updates at each institution.

Furthermore, the chatbot contributes to significant operational efficiency improvements by automating responses to frequently asked academic queries such as course details, examination schedules, assignment deadlines, registration procedures, and institutional policies. By handling a large volume of repetitive inquiries, the system reduces dependency on administrative staff and academic coordinators. This not only minimizes operational workload but also improves response time and service consistency. As a result, human staff can focus on higher-value tasks such as student counseling, academic planning, and decision-making processes.

The integration of personalization features further enhances its commercial value by enabling the system to deliver adaptive and context-aware responses tailored to individual student needs. By analyzing user interactions, academic performance patterns, and engagement history, the system can provide personalized recommendations, study guidance, and learning resource suggestions. This level of customization improves student satisfaction, retention, and academic performance, which are key performance indicators for educational institutions and directly increase the perceived value of the product.

Additionally, the system's ability to support multimodal interaction (such as text-based chat, voice input, and document-based queries) improves accessibility and usability for diverse user groups, including students with different learning preferences or accessibility needs. The inclusion of proactive assistance—such as notifying students about upcoming deadlines, missing submissions, or recommended learning materials—further enhances engagement by shifting the system from a reactive tool to an intelligent academic companion.

From a commercial perspective, these advanced capabilities make the solution highly competitive in the EdTech market, where institutions increasingly prefer AI-driven platforms that offer automation, intelligence, and personalization in a single integrated system. The system's modular and scalable architecture also allows easy integration with existing university systems such as Learning Management Systems (LMS), Student Information Systems (SIS), and digital libraries, further increasing its adoption potential.

With its scalability, adaptability, and user-centered design, the system has strong potential to be extended beyond the education sector. In corporate environments, it can be adapted for employee onboarding, internal knowledge management, and training assistance. In online learning platforms, it can function as an AI tutor supporting millions of learners simultaneously. It can also be utilized in professional certification programs, government training initiatives, and digital skill development platforms. This cross-domain applicability significantly expands its market reach and long-term revenue potential, making it a viable commercial AI product with sustainable growth opportunities.

2.4 Testing & Implementation

The system was implemented as a fully integrated web-based application that combines frontend, backend, and AI components into a unified architecture. The frontend interface, developed using modern web technologies, provides an interactive chatbot environment for users, while the backend manages data processing, API communication, and integration with AI services. The AI layer incorporates RAG-based retrieval, web search fallback, and personalization mechanisms to deliver intelligent responses.

To ensure system reliability, accuracy, and performance, multiple testing strategies were applied throughout the development process. Functional testing was conducted to verify that the chatbot correctly processes user queries and generates appropriate responses across different scenarios. Performance testing evaluated the system's responsiveness and ability to handle multiple users simultaneously without delays or failures. Personalization validation was carried out to ensure that student performance data is accurately analyzed and that the system generates correct recommendations and reminders based on predefined criteria.

Additionally, multimodal testing was performed to verify the proper handling of different input types, including text, voice, and image-based queries, ensuring that the system maintains accuracy and usability across all interaction modes. Integration testing played a critical role in validating seamless communication between system components, including the frontend interface, backend APIs, vector database, and AI services. This ensured that data flows efficiently across the system and that all modules function cohesively. Overall, the implementation and testing processes confirm that the system is robust, reliable, and capable of delivering high-quality academic support in real-world environments.

2.4.1 Implementation

2.4.1.1 System Overview and Setup

The Conversational AI Assistant was implemented as a full-stack intelligent system integrating frontend, backend, artificial intelligence, and data management components. The system follows a modular and layered architecture to ensure scalability, flexibility, and maintainability. The frontend was developed using React, providing a responsive and interactive user interface for chatbot interaction. The backend was implemented using Node.js and Firebase, which handles API requests, authentication, and data storage.

The AI layer consists of a hybrid architecture combining Retrieval-Augmented Generation (RAG), web search fallback, and personalization mechanisms. A local Large Language Model (LLM) was deployed using Ollama to ensure privacy and offline capability. ChromaDB was integrated as the vector database for semantic search operations.

The development environment included tools such as Visual Studio Code, Git for version control, and Postman for API testing. Proper environment configurations were established to manage dependencies and ensure smooth system execution. This setup enables seamless communication between all components, ensuring efficient data flow and high system performance.

2.4.1.2 Data Collection and Document Processing

Academic data was collected from institutional sources, including university rules, regulations, lecturer details, and administrative documents. These documents were primarily in PDF format and required preprocessing before use.

Optical Character Recognition (OCR) using Tesseract was applied to extract text from scanned documents. The extracted text was cleaned and normalized to remove noise, irrelevant symbols, and formatting issues. This preprocessing ensures high-quality input data for further processing.

The processed data forms the foundation of the chatbot's knowledge base, enabling accurate retrieval and response generation.

2.4.1.3 Text Chunking and Embedding Generation

The extracted text was divided into smaller segments known as chunks to improve retrieval precision. Chunking ensures that the system retrieves relevant portions of documents rather than entire files.

Each chunk was converted into vector embeddings using an embedding model. These embeddings capture semantic meaning, allowing the system to understand context and relationships between words. This significantly improves retrieval accuracy compared to traditional keyword-based methods.

2.4.1.4 Vector Database (ChromaDB) Integration

The generated embeddings were stored in ChromaDB, a vector database optimized for similarity search. ChromaDB allows efficient retrieval of relevant document chunks using vector distance metrics.

The database supports fast indexing and querying, ensuring quick response times. Integration with the backend enables seamless retrieval of relevant data during query processing.

2.4.1.5 Multimodal Input Implementation (Text, Voice, Image)

The system supports multiple input methods, including text, voice, and image inputs. Voice input is handled using speech recognition, while image input allows users to upload text-based images.

A validation mechanism ensures that only relevant images containing text are processed, preventing invalid inputs such as animals or scenery.

2.4.1.6 Advanced Chatbot Functionalities and Intelligent Response Mechanisms

The proposed Conversational AI Assistant integrates multiple advanced functionalities to enhance response accuracy, user experience, and system adaptability. The system primarily utilizes a Retrieval-Augmented Generation (RAG) mechanism to retrieve relevant document chunks from the vector database, which are then combined with the user query and processed by the Large Language Model (LLM) to generate accurate and context-aware responses. In situations where the internal knowledge base does not contain sufficient information, a web search fallback mechanism is activated, enabling the system to retrieve and summarize relevant content from external sources, thereby ensuring continuous response availability. The response generation process is handled by a locally deployed LLM via Ollama, which enhances data privacy, reduces reliance on external services, and supports offline functionality while producing human-like responses. Furthermore, the system incorporates a personalization module that analyzes student performance data using predefined thresholds to classify users into different performance categories and generate tailored recommendations, reminders, and feedback. In addition to informational support, the chatbot provides action-based features such as navigating system dashboards, accessing academic resources, and downloading PDF documents directly through conversational commands, improving overall efficiency. To enhance accessibility, a text-to-speech module is integrated, allowing users to listen to responses. The system also includes a feedback collection mechanism that captures user input to continuously improve performance and accuracy. Moreover, customizable chatbot settings enable users to control interaction modes, voice responses, and chat management features such as resetting conversations, thereby ensuring a flexible and user-centered experience.

2.4.1.7 AI Model and Retrieval Implementation

The AI model and retrieval component of the proposed system is designed to ensure accurate, context-aware, and reliable response generation by integrating multiple intelligent mechanisms. Initially, an embedding model is used to convert both user queries and document content into vector representations, capturing the semantic meaning of the text. These vectors are then compared using similarity search techniques based on distance metrics to identify the most relevant information from the vector database. A similarity threshold (e.g., distance ≤ 1.2) is applied as a decision criterion to determine whether the retrieved content is sufficiently relevant. If the similarity score meets this threshold, the system proceeds with Retrieval-Augmented Generation (RAG); otherwise, it activates the web search fallback mechanism to obtain external information. This results in a hybrid architecture that combines internal document-based retrieval with real-time web search, significantly improving response coverage and system robustness. Additionally, prompt engineering techniques are employed to construct structured inputs for the Large Language Model (LLM), where the user query is combined with retrieved contextual information to guide the model in generating accurate and meaningful responses. Furthermore, a rule-based personalization algorithm is implemented to analyze student performance data using predefined metrics, enabling the system to classify students into performance categories and generate tailored recommendations, reminders, and feedback. This integrated approach enhances both the intelligence and adaptability of the chatbot system.

2.4.2 Testing

The testing phase of the Conversational AI Assistant was conducted using a comprehensive and systematic approach to ensure the system's reliability, accuracy, and overall performance across all functionalities. Initially, a well-defined test plan and strategy were established to guide the testing process, covering all system components including frontend, backend, AI modules, and database interactions. Functional testing was performed to verify that the chatbot correctly processes user queries and generates accurate, relevant responses based on both internal document retrieval and external web search fallback. Performance testing was conducted to evaluate system responsiveness and scalability, ensuring that the chatbot can handle multiple concurrent users while maintaining an optimal response time of approximately 2–3 seconds. Multimodal input testing was carried out to validate the system's ability to accurately process and interpret different input types, including text, voice, and image-based queries, while also ensuring proper validation of unsupported inputs. Additionally, personalization and recommendation testing was implemented to confirm that the system correctly classifies student performance levels and generates appropriate recommendations and reminders. Integration testing ensured seamless communication and data flow between all system components, including the frontend interface, backend APIs, vector database, and AI services. User Acceptance Testing (UAT) was conducted with real users to evaluate usability, effectiveness, and overall user satisfaction. Finally, error handling and validation testing was performed to ensure that the system can gracefully manage invalid inputs, system errors, and unexpected scenarios, thereby maintaining robustness and reliability.

2.4.2.1 Test Case Design

The test case design for the Conversational AI Assistant was developed to systematically evaluate the accuracy, reliability, and functionality of all major system components. A structured set of test scenarios was created to cover different aspects of chatbot behavior and system performance. Chatbot query handling test cases were designed to assess the system's ability to understand user inputs and generate accurate, contextually relevant responses based on both document knowledge and conversational intent. RAG retrieval accuracy test cases focused on verifying whether the system correctly retrieves the most relevant document chunks from the vector database using semantic similarity search. Web search fallback test cases were implemented to validate the system's ability to activate external information retrieval when internal data is insufficient, ensuring continuous response availability. Multimodal input validation test cases were designed to evaluate how effectively the system processes and validates different input types, including text, voice, and image inputs, while rejecting invalid or irrelevant data such as non-text images. Additionally, personalization feature test cases were conducted to ensure that the system accurately classifies student performance levels and generates appropriate recommendations, reminders, and feedback. Overall, the test case design ensures comprehensive coverage of system functionalities, enabling thorough validation of both core features and advanced capabilities of the chatbot.

2.4.2.2 Expected test cases

- Data Integrity Test

Objective: Verify chat input (text, voice, image, PDF) is transmitted without tampering.
Expected Result: Manipulated or truncated data is detected and discarded.

- Login Authentication Test

Objective: Verify that only authorized students can log in to access the chatbot.
Expected Result: Invalid login attempts, or unauthorized users are disallowed.

- Multi-modal Input Test

Objective: Verify chatbot can successfully process different input types (text, voice, image, PDF).
Expected Result: All types of input are detected, translated, and processed as expected.

- Intent Recognition Accuracy Test

Objective: Test student questions (e.g., deadlines, module codes) are correctly identified by chatbot.
Expected Result: at least 90% accurate intent identification.

- Entity Extraction Test

Objective: Test that entities (module code, lecturer name, payment details) are correctly extracted by chatbot from student input.
Expected Result: Extracted entities are identical to LMS database values.

- Response Accuracy Test

Objective: Ensure that chatbot provides correct and relevant answers according to LMS data.
Expected Result: Answers align with actual deadlines, module data, or academic regulations.

- Email Explanation Test

Objective: Confirm whether chatbot can translate confusing system-generated emails to students.
Expected Result: Chatbot explains emails in an easy-to-understand language for students.

- Performance Test (Response Time)

Objective: Test response time of chatbot for regular and peak loads.
Expected Result: Answers within 2–3 seconds are received.

- Security & Privacy Test

Objective: Safeguard sensitive student information when stored and transmitted.
Expected Result: No data leakage; encrypted information is unreadable if intercepted.

- Scalability Test

Objective: Verify chatbot functionality were hundreds of students converse concurrently.
Expected Result: At least 500+ simultaneous requests are handled without system breakdown.

3. RESULTS & DISCUSSION

3.1 Results

The implemented Conversational AI Assistant demonstrated strong performance across all key functional components. The system achieved high accuracy in responding to academic queries through the use of the Retrieval-Augmented Generation (RAG) mechanism, where responses were generated based on relevant document chunks retrieved from the knowledge base. This ensured that answers were contextually accurate and aligned with institutional information such as rules, regulations, and guidelines.

In addition, the web search fallback mechanism proved to be highly effective in handling queries that were outside the scope of the internal document database. By dynamically retrieving and summarizing external information, the system maintained continuous response availability, thereby improving robustness and user satisfaction. The personalization module also performed effectively by analyzing student performance data and generating adaptive recommendations, reminders, and motivational feedback tailored to individual users.

Furthermore, the integration of multimodal interaction significantly enhanced system usability and accessibility. Users were able to interact with the chatbot using text, voice, and image inputs, which improved flexibility and inclusiveness. The voice response feature further enhanced user experience by allowing responses to be delivered audibly. Overall, the system demonstrated reliable performance, efficient response generation, and improved user interaction compared to traditional academic support systems.

3.2 Research Findings

The experimental results show a number of significant conclusions about the performance and efficacy of the suggested Conversational AI Assistant system in an academic assistance setting. These conclusions come from feature evaluations, user interaction patterns, and system testing under various query contexts.

Firstly, the hybrid retrieval approach, which combines Retrieval-Augmented Generation (RAG) with a web search fallback mechanism, significantly improves system reliability, response accuracy, and coverage. The RAG component ensures that responses generated from the internal knowledge base are contextually relevant and grounded in verified academic data. However, in cases where the query falls outside the scope of the internal dataset, the web search fallback mechanism retrieves up-to-date external information. This dual-layer architecture reduces the likelihood of unanswered queries and minimizes hallucinated responses, thereby ensuring that users consistently receive meaningful and accurate answers even for unseen or out-of-domain questions.

Second, user engagement, learning efficacy, and answer contextual relevance are all improved by using personalization methods based on student performance data. The system is able to produce adaptive feedback that is customized to each student's unique learning needs by examining student interaction history, academic progress, and performance patterns. For instance, specific explanations, extra learning materials, and revision advice might be given to students who commonly struggle with particular subjects. By filling in specific knowledge gaps rather than offering general answers, this adaptive behavior promotes a more customized learning process and enhances academic results.

Thirdly, the implementation of multimodal interaction significantly improves system accessibility and usability. By supporting multiple input methods such as text-based chat, voice input, and potentially document or file-based queries, the system accommodates diverse user preferences and accessibility needs. This is particularly beneficial in educational environments where students may have varying levels of digital literacy or may prefer different interaction styles depending on context. The multimodal capability also increases system flexibility, making it more practical for real-world deployment across different devices such as mobile phones, laptops, and tablets.

Additionally, the inclusion of Explainable AI (XAI) mechanisms enhances user trust, transparency, and interpretability of system outputs. Instead of providing only final answers, the system can present explanations of how responses were generated, including references to retrieved documents, reasoning paths, or data sources. This transparency allows users to understand the logic behind recommendations or answers, reducing uncertainty and improving confidence in the system’s reliability. In an academic context, this is particularly important as students and educators require justification for information accuracy and source credibility.

Furthermore, the combination of these AI techniques—hybrid retrieval, personalization, multimodal interaction, and explainability—creates a synergistic effect that improves the overall intelligence and usability of the system. The results indicate that the system is not only capable of answering academic queries effectively but also adapts to user needs, provides transparent reasoning, and maintains high reliability across different usage scenarios. This demonstrates that integrating multiple advanced AI components leads to a more robust, user-centered, and scalable academic support solution suitable for real-world educational deployment.

3.3 Discussion

By combining several cutting-edge AI technologies into a single, intelligent architecture, the suggested Conversational AI Assistant is a major improvement above conventional chatbot systems. The suggested system uses dynamic knowledge retrieval and natural language generation techniques to generate context-aware, adaptable, and flexible responses, in contrast to traditional chatbots that mostly rely on static rule-based responses or predefined decision trees. This makes it possible for the system to respond to a wider variety of user inquiries with more precision and contextual awareness, particularly in challenging academic situations where strict rule-based systems frequently fall short.

The hybrid Retrieval-Augmented Generation (RAG) architecture's integration is a significant enhancement. This method benefits from generative AI capabilities for natural and cohesive answer formulation while guaranteeing that responses are based on validated internal knowledge sources. The hallucination problems that are frequently seen in standalone generative models are greatly reduced by the RAG method. Additionally, by allowing the chatbot to respond to requests that are out-of-domain or unseen, the addition of a web search fallback method further improves system reliability. The system is more dependable in real-world academic settings where queries can be extremely varied and unpredictable thanks to its dual-layer retrieval method, which guarantees ongoing service availability and enhances overall answer coverage.

The system's customisation features set it apart from other chatbot options. This system uses student-specific information, such as academic performance, interaction history, and learning behavior patterns, to produce customized responses, in contrast to generic assistants that offer consistent responses to all users. This makes it possible for the chatbot to offer proactive academic assistance, such as deadline reminders, tailored study suggestions, and encouraging comments. As a result, the system becomes an intelligent academic partner that actively promotes student learning and engagement rather than a reactive information source. User satisfaction and learning outcomes are greatly enhanced by this adjustable flexibility.

Additionally, the support for multimodal interaction enhances both usability and accessibility. By allowing users to interact through multiple input modalities such as text, voice, and potentially document-based queries, the system accommodates different learning preferences and accessibility requirements. This is particularly beneficial in diverse educational environments where students may have varying levels of digital literacy or may require alternative interaction methods. The multimodal design also improves system flexibility across devices, making it suitable for deployment on mobile applications, web platforms, and integrated learning management systems.

However, certain limitations must be acknowledged in the current implementation. The performance and reliability of the system are highly dependent on the quality, completeness, and relevance of the academic documents stored in the internal knowledge base. If the dataset is outdated, incomplete, or poorly structured, the accuracy of generated responses may be negatively affected. Additionally, while the web search fallback mechanism improves coverage, its effectiveness is influenced by the reliability, credibility, and ranking of external information sources, which may sometimes introduce variability in response quality.

Despite these limitations, the proposed system demonstrates strong potential for further enhancement and real-world application. With improvements in dataset curation, continuous knowledge base updates, and more advanced filtering of external sources, the system can achieve higher accuracy and reliability. Overall, the integration of RAG, personalization, multimodal interaction, and explainable AI concepts positions the system as a next-generation intelligent academic support assistant with significant applicability in modern educational environments.

3.4 Summary

In summary, the proposed Conversational AI Assistant successfully achieves its research objectives by integrating multiple advanced AI components into a unified and cohesive framework. The system combines hybrid knowledge retrieval (Retrieval-Augmented Generation with web search fallback), personalization mechanisms, multimodal interaction capabilities, and Explainable AI (XAI) features to deliver a comprehensive and intelligent academic support solution.

The experimental results and system evaluation indicate that the chatbot is capable of generating accurate, reliable, and context-aware responses across a wide range of academic queries. The hybrid retrieval approach significantly improves response completeness and reduces the limitations typically associated with standalone generative models, while also ensuring robustness when handling unseen or out-of-domain queries.

Furthermore, the integration of personalization enhances the system's ability to adapt to individual student needs by providing tailored recommendations, feedback, and proactive academic assistance. This contributes to increased user engagement, improved learning experience, and better academic performance outcomes. The inclusion of multimodal interaction further strengthens usability by enabling flexible communication methods, thereby making the system more accessible to a diverse user base.

Furthermore, multimodal interaction assistance improves accessibility and usability. The system supports various learning styles and accessibility needs by enabling users to interact through a variety of input modalities, including text, speech, and possibly document-based inquiries. This is especially helpful in various learning situations where students might need different ways to interact or have different levels of digital literacy. Additionally, the multimodal architecture increases system adaptability across devices, which makes it appropriate for implementation on online platforms, mobile applications, and integrated learning management systems.

As a result, the suggested approach clearly shows worth as a clever teaching tool that raises the effectiveness and efficiency of academic support services. Additionally, it exhibits a great deal of promise for practical implementation and additional improvement in subsequent rounds, especially through enhanced data integration, model optimization, and increased domain applicability.

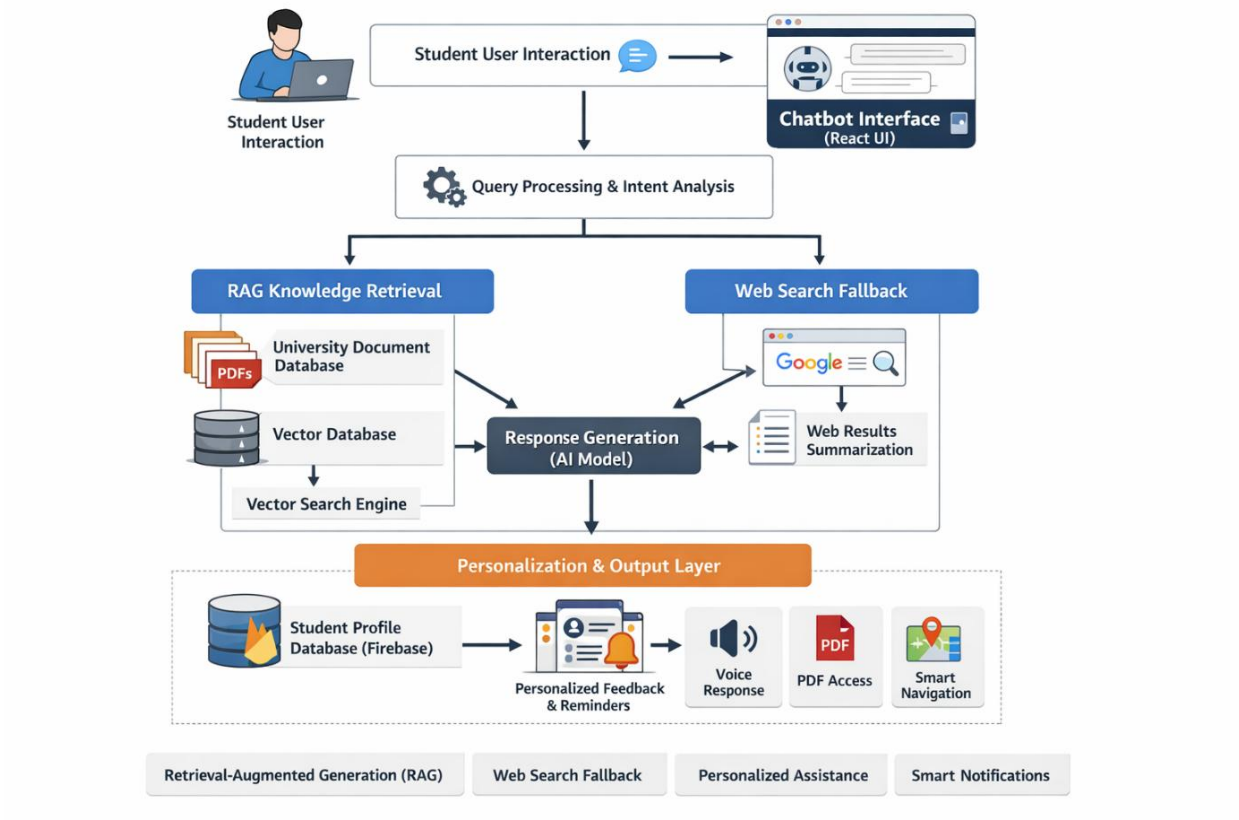

Figure 4 System Architecture

4. DESCRIPTION OF PERSONAL AND FACILITIES

Table 2 Description of personnel and facilities

Registration No	Name	Task Description

IT22365750	Nimanji D.L. K	• Collect academic data from the LMS such as module data, deadlines, lecturer information, and registration records. • Prepare and develop datasets consisting of student queries, emails, and documents (text, voice, images, PDFs) to train the chatbot. • Train machine learning models for intent recognition and entity extraction for effective understanding of student queries. • Provide multi-modal support so that the chatbot can process text, voice, images, and PDFs. • Connect the chatbot with the LMS to fetch real-time academic data and reply with customized solutions. • Add explainability features to simplify complex academic emails or system alerts in a step-by-step manner for students. • Create a secure web-based chat window using React.js for easy student interaction. • Use Firebase Firestore for encrypted storage of conversation history and user interactions. • Ensure strong security with JWT authentication and data communication encryption. • Test the system with students and lecturers for accuracy, performance, and usability.

5. CONCLUSION

This study successfully created an intelligent Conversational AI Assistant that integrates cutting-edge AI technology to improve academic help. The system offers a complete and flexible solution for student support by integrating Retrieval-Augmented

Generation (RAG), online search fallback, personalization options, and multimodal interaction. When compared to conventional academic help techniques, the chatbot exhibits enhanced accuracy, usability, and response.

A local Large Language Model improves data protection and lessens reliance on outside services, while Explainable AI guarantees openness in system replies. These characteristics support the system's dependability, security, and user confidence. Additionally, by adding proactive and action-based features, the chatbot becomes an intelligent assistant that can help students with more than just answering simple questions.

Overall, by resolving important issues with current chatbot solutions, the suggested approach makes a substantial addition to the field of instructional AI. With great potential for practical implementation in educational institutions, it offers a scalable, adaptable, and user-centered approach to academic support. In order to further increase accessibility and efficacy, future enhancements might include increased multilingual capabilities, improved customisation using sophisticated machine learning models, and deeper integration with learning management systems.

REFERENCES

[1] L. Labadze, T. Pirveli, and K. Buadze, “Role of AI chatbots in education: a systematic literature review,” *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, pp. 1–27, 2023.

[2] N. F. Davar, “AI chatbots in education: Challenges and opportunities,” *Information*, vol. 16, no. 3, p. 235, 2025.

[3] M. A. Kuhail, M. Alturki, and M. Al-Samarraie, “Interacting with educational chatbots: A systematic review,” *Education and Information Technologies*, vol. 28, pp. 1417–1444, 2023.

[4] C. W. Okonkwo and A. Ade-Ibijola, “Chatbots applications in education: A systematic review,” *Smart Learning Environments*, vol. 8, no. 1, pp. 1–25, 2021.

[5] J. Chan and Y. Li, “Enhancing higher education with generative AI: A multimodal approach for personalised learning,” *arXiv preprint* arXiv:2502.07401, 2025.

[6] S. Latif, M. Usama, and A. Qadir, “PhysicsAssistant: An LLM-powered interactive learning robot for physics lab investigations,” *arXiv preprint* arXiv:2403.18721, 2024.

[7] S. Winkler and M. Söllner, “Unleashing the potential of chatbots in education: A state-of-the-art analysis,” *International Journal of Educational Technology in Higher Education*, vol. 18, no. 1, pp. 1–20, 2021.

[8] A. Wollny, K. Schneider, and J. L. Seeber, “AI-based conversational agents in higher education: A systematic review,” *Computers and Education: Artificial Intelligence*, vol. 3, p. 100034, 2022.

[9] P. K. Sahu and R. R. Padhy, “Chatbot for education system,” in *2021 2nd International Conference on Advances in Education and Information Technology (AEIT)*, 2021, pp. 90–95.

[10] A. Smutny and P. Schreiberova, “Chatbots for learning: A review of educational chatbots for the Facebook Messenger,” *Computers & Education*, vol. 151, p. 103862, 2020.

[11] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of NAACL-HLT*, 2019.

[13] T. Brown *et al.*, “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[14] OpenAI, “GPT Models Documentation,” 2024. [Online]. Available: https://platform.openai.com/docs

[15] F. Chollet, *Deep Learning with Python*. Manning Publications, 2017.

[16] Tesseract OCR, “Tesseract Open Source OCR Engine,” 2023. [Online]. Available: https://github.com/tesseract-ocr/tesseract

[17] ChromaDB, "Chroma: Open-source embedding database," 2024. [Online]. Available: https://www.trychroma.com

[18] Firebase, "Cloud Firestore Documentation," Google, 2024. [Online]. Available: https://firebase.google.com/docs/firestore

[19] A. Vaswani *et al.*, "Attention is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[20] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Pearson, 2023.

APPENDIX