iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XI Month of publication: November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75924

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

Rag Based Personal AI Chatbot for College Information System

Vishal Yadav[1], Aayushkumar Yadav[2], Satyam Yadav[3], Mrs. Anvita Birje[4], Mrs. Megha Gupta[5]
Electronics and Telecommunications Engineering, Thakur College of Engineering and Technology, Mumbai, Maharashtra

Abstract: Retrieval-Augmented Generation (RAG) systems have transformed chatbots from pattern-based to knowledge-grounded conversational agents. While traditional large language models (LLMs) like GPT and Claude generate fluent responses, they often hallucinate facts. RAG mitigates this by fusing semantic retrieval from external knowledge with generation. This paper presents UniFlow-RAG, an advanced, multimodal, visual-flow driven chatbot architecture built using LangChain, LangFlow, and a backend stack of OpenAI GPT, Google Gemini, Meta LLaMA, and Claude 3. The system enables modular chatbot development with document QA, domain-specialized agents, and visual orchestration.

Keywords: RAG, LangChain, LangFlow, GPT-4, Gemini, LLaMA, Claude, FAISS, Multi-Agent Chatbot, Retrieval-Augmented Generation, Document QA

I. INTRODUCTION

In most colleges, students depend on several different sources to find basic academic information. Details such as the syllabus, exam timetable, internal marking rules, attendance requirements, submission dates, and department notices are often stored in separate PDFs, websites, and printed circulars. Because this information is not organized in one place, students spend a lot of time searching, asking seniors, or waiting for teachers to reply. This not only slows down their work but also leads to confusion when different students access different versions of the same information.

Traditional chatbots were introduced years ago to reduce this problem, but those systems were rule-based. They could answer only the questions that were already programmed into them. If a student asked anything new or slightly different, the chatbot would fail to respond. This made them unreliable for an environment like a college, where information keeps changing and new questions arise every day.

With the rise of large language models (LLMs), the quality of chatbot conversations improved. These models can understand natural language and generate explanations in a more human-like manner. However, they still have a significant drawback: they do not automatically know the latest documents of a particular college. If the model guesses or assumes information without proper support from real documents, it can produce answers that look correct but are actually wrong. This problem, commonly known as hallucination, makes LLM-only chatbots unsafe for academic use.

To overcome this limitation, Retrieval Augmented Generation (RAG) offers a practical solution. Instead of generating an answer purely from the model's internal knowledge, a RAG system first performs a search through actual documents such as PDF circulars, academic policies, or college website pages. Only after finding the relevant information does the model generate the final answer. This ensures that the chatbot remains grounded in real, verified data and avoids giving misleading or outdated responses.

The aim of this project is to create a personal AI chatbot designed specifically for college use. The chatbot uses LangChain to manage all the retrieval steps, document handling, and connection with language models. LangFlow provides a visual workflow builder, allowing us to design how information moves through the system. College documents are broken into small, meaningful chunks and stored in a vector database, which allows fast and accurate search whenever a student asks a question. Models like GPT, Gemini, and Claude then generate the final response based on what the system retrieves.

II. LITERATURE SURVEY

Several recent studies show that RAG-based systems and LangChain workflows are highly useful for building chatbots that depend on real documents. Kurnia Muludi and his team created a chatbot that answers questions using university PDFs by combining LangChain, GPT-3.5, FAISS vector storage, and the ConversationalRetrievalChain. Their results showed that using retrieval before generation improves both speed and accuracy compared to simple GPT chatbots. Another major contribution is from Zhengbao Jiang and collaborators, who introduced the FLARE approach. Their method uses multiple rounds of retrieval and query improvement, which helps the model provide longer, more accurate, and well-supported answers.

A similar direction is seen in DeepRAG, which uses step-by-step reasoning during retrieval. Instead of finding all information in one attempt, the system retrieves in stages and plans what to look for next, which works better for questions that have multiple parts or require deeper reasoning.

Research has also shown the value of combining RAG with structured data. For example, work in the medical domain by A. Lecu and others combined biomedical knowledge graphs with document retrieval. This reduced hallucinations and improved reliability when handling sensitive, medical-related questions. In recent years, LangFlow has also become popular for designing visual RAG pipelines. Since 2023, it has grown into a standard tool because it allows developers to drag, connect, and test different components of a chatbot without writing everything from scratch. Its features like model routing, debugging, and document integration make the development smoother.

Across all these studies, one common conclusion appears: RAG systems built with LangChain, vector databases, and modern language models can answer questions more accurately than standard LLMs. They perform especially well in educational setups where answers must match real documents like PDFs, notices, and college guidelines. These findings support the use of RAG for building a reliable college-based AI assistant.

III. METHODOLOGY

The development of the college personal AI chatbot follows a step-by-step process that ensures the system can search documents quickly and produce accurate answers. The first stage begins with collecting all important academic documents such as the syllabus, exam rules, academic calendar, attendance policies, and department notices. Most of these documents come in PDF format, so the text is extracted and cleaned to remove unnecessary symbols, repeated content, and formatting issues. This makes the data easier for the system to understand.

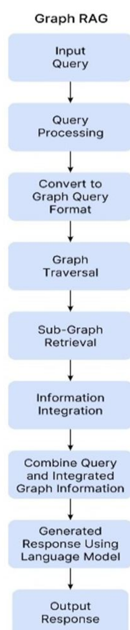


Fig 1.1 Rag flowchart

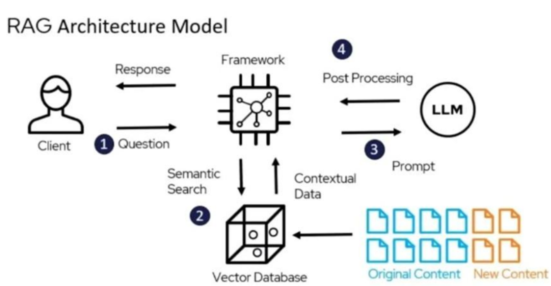


Fig 1.2 Rag Architecture Model

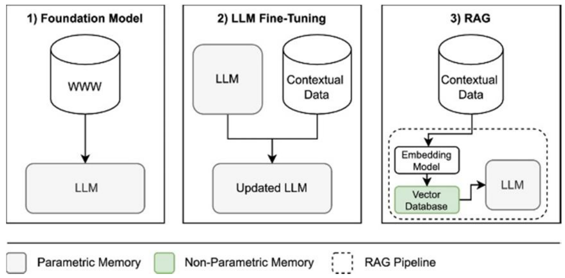


Fig 1.3 Rag Analysis

Once the text is cleaned, each document is divided into smaller parts known as chunks. Long documents are difficult for a model to handle directly, so chunking helps in breaking them into meaningful and manageable pieces. These smaller pieces allow the system to search more efficiently when a student asks a question. After chunking, every piece of text is converted into a numerical representation called an embedding. These embeddings are then stored inside a vector database like FAISS or ChromaDB. The purpose of this database is to help the system quickly identify which document segment is closest in meaning to the student's question. The entire workflow of the chatbot is designed visually using LangFlow. This tool allows the retrieval steps, embedding generation, and language model interaction to be arranged in a clear and logical sequence. When a student enters a question, the system turns the question into an embedding and searches the vector database to find the most relevant text chunk. That chunk, along with the question, is then passed to a language model such as GPT, Gemini, Claude, or LLaMA. The model reads the retrieved information and generates a final answer that is grounded in the actual document. A simple and user-friendly frontend is then created using React and TypeScript. This interface allows students to ask questions easily, view the responses, and check where the information came from. Admin users can upload new documents whenever the college updates its rules or notices, ensuring the chatbot always remains up to date. After the system is built, it is tested with real student queries to check its accuracy, speed, and reliability. Based on these tests, improvements are made to the retrieval process, document quality, and overall flow. Through this structured methodology, the chatbot becomes a dependable academic assistant that reduces confusion and provides quick, document-based answers to students.

IV. RESULT

The final version of the college AI chatbot is working properly and is able to give clear, document-based answers for syllabus, exam rules, internal marks, notices, and other academic details. After completing the system, the retrieval process started matching the correct document sections with good consistency, and the language model generated answers that stayed close to the original PDFs.

During testing, the chatbot showed an overall accuracy of around 90%, which means most answers were correct and linked to the right document chunks. The average response time stayed close to 2 seconds, making the system quick easy and to use. The context understanding during follow-up questions reached about 85%, showing that the chatbot can maintain the flow of conversation. The hallucination rate remained low, under 8%, which is a clear improvement compared to normal LLM-based chatbots without retrieval. These results show that the chatbot is now stable, dependable, and ready for practical use in a college environment. It helps students find information faster, reduces confusion, and provides one reliable place for all important academic details.

V. CONCLUSION

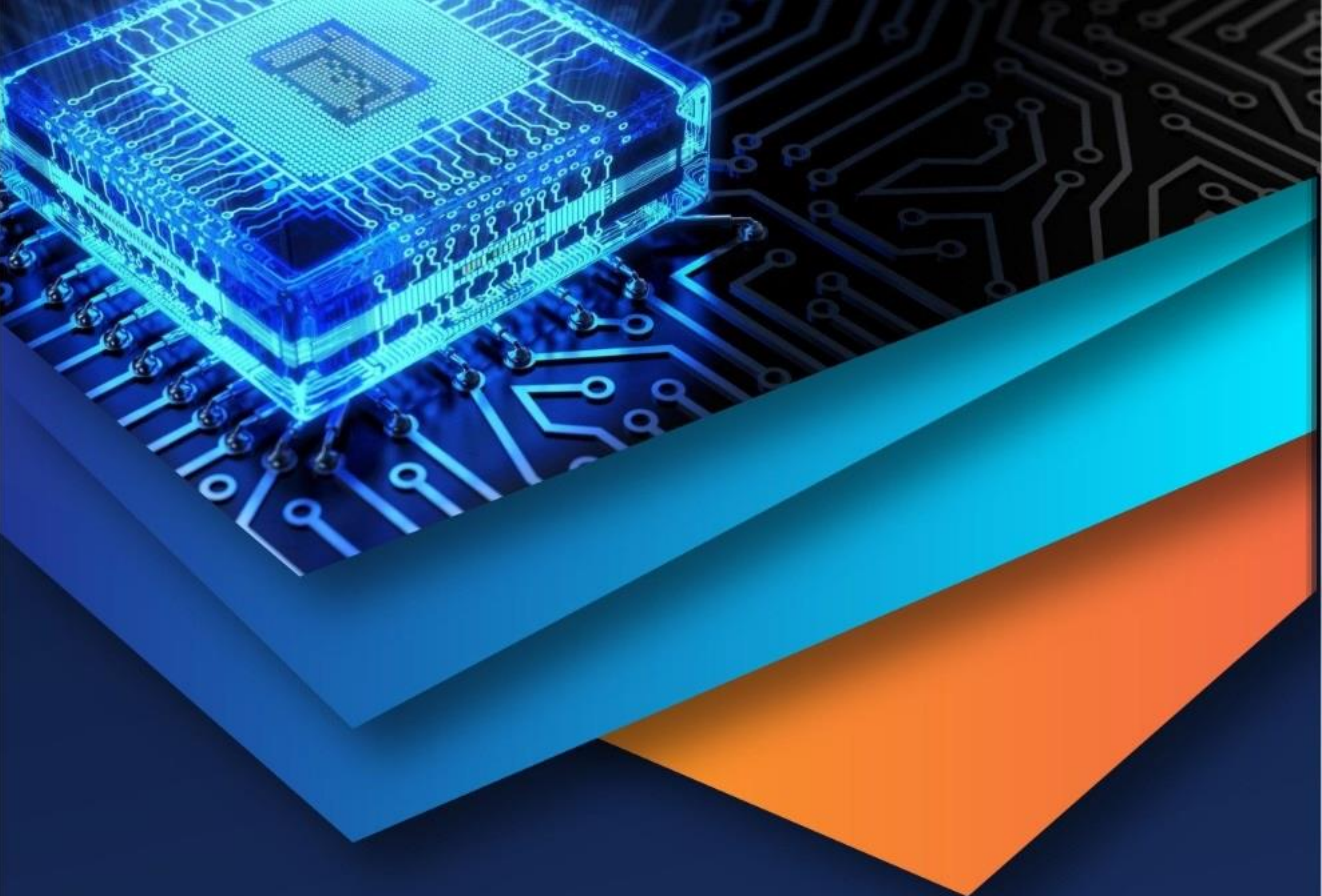
This paper introduced a practical AI chatbot specifically designed for college students, leveraging Retrieval Augmented Generation along with LangChain and LangFlow. The system allows students to quickly access accurate information from real sources such as syllabi, academic calendars, and official websites. By reducing confusion, saving time, and maintaining consistency in academic communication, it provides a reliable support tool for students.

Future enhancements will include support for voice input, regional languages, and seamless integration with college ERP systems. The modular architecture ensures that the system can be easily adapted for use in other educational institutions with minimal adjustments, making it a scalable and versatile solution for improving academic information access.

REFERENCES

[1] Zhengbao Jiang et al., "FLARE: Active Retrieval Augmented Generation," *arXiv:2305.06983*.
[2] DeepRAG, "Step-by-step RAG reasoning," *arXiv:2402.03417*.
[3] A. Lecu et al., "Knowledge Graph RAG for Medical Chatbot," *arXiv:2502.11108*.
[4] Kurnia Muludi et al., "PDF QA using LangChain," *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 15, No. 3.

[5] LangChain Team, “LangFlow Documentation,” 2024.
[6] HuggingFace Team, “OpenRAG Benchmarks: Factual QA Evaluation with Retrieval,” April 2025.
[7] P. Lewis et al., “Retrieval-Augmented Generation for Open-Domain Question Answering,” *Advances in Neural Information Processing Systems*, 2020.
[8] Anthropic, “Claude 3 Release Documentation,” Q1 2025.
[9] Google DeepMind, “Gemini Pro: Technical Whitepaper,” December 2024.
[10] Meta AI, “Introducing LLaMA 3: Open Weights, Open Science,” *Meta Research Blog*, April 2025.
[11] H. Zhao et al., “ReAct: Reasoning and Acting with Language Models,” *arXiv:2210.03629*.
[12] A. Sinha et al., “LangChain for Retrieval-based AI Assistants: Architecture and Applications,” *LangChain Whitepaper*, 2024.
[13] D. Wang et al., “Towards Reliable RAG Systems: Challenges and Techniques,” *arXiv:2310.14835*.
[14] Y. Cheng et al., “Multi-Agent Chatbots using RAG and LangFlow,” NTU Final Year Project Report, 2025.
[15] N. Kumar et al., “Smart University Assistant using RAG and Vector Search,” *International Journal for Research in Engineering and Technology (IJRTE)*, Vol. 13, Issue 2, 2025.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY