AbeSphere: LLM-Powered Intelligent Assistant for Academic Institution Using GenAI

Reshu Tyagi
Department of Information Technology
ABES Institute of Technology, Dr. APJ Abdul Kalam University Lucknow, India

Shitanshu Goswami
Department of Computer Science & Engineering
ABES Institute of Technology, Dr. APJ Abdul Kalam Technical University Lucknow, India

Saurabh Tiwari
Department of Computer Science & Engineering
ABES Institute of Technology, Dr. APJ Abdul Kalam University Lucknow, India

Sumit Yadav
Department of Computer Science & Engineering
ABES Institute of Technology, Dr. APJ Abdul Kalam Technical University Lucknow, India

Vishal Yadav
Department of Computer Science & Engineering
ABES Institute of Technology, Dr. APJ Abdul Kalam Technical University Lucknow, India

***Abstract*—Information seeking is a very important step for all the students of schools and colleges because it helps them in effective learning and better decision-making. In today's digital world, students are depends on various information sources, which often creates challenges such as lack of information literacy skills, difficulty in identifying credible sources, information overload or false content .Getting of accurate, relevant, and timely information is still a challenging task. We have developed a model which can sort all these problems efficiently and timely. Our model works on Python with FastAPI and LangGraph for intelligent reasoning . The system uses Retrieval-Augmented Generation (RAG) with MongoDB which is used for saving chat history and context. We have added features i.e. multilingual support, personalized guidance and multiple source data integration that gives 92% response accuratly and also 85% query resolution rate which is available for 24/7. It can reduce 60% of the official workload by using role-based access control. Also , it can solve the unique challenges faced by higher education institutions in developing countries such as India. It also includes the multilingual requirements, diverse data sources, and limited technical infrastructures.**

Keywords — Large Language Models, Generative AI, Educational Chatbot, RAG Architecture, LangGraph, NLP, Academic Support, Intelligent Assistant.

I.INTRODUCTION

With the rapid growth of modern technologies in today's education system, now it's time to reform the traditional methods used by students and faculty to handle and access a huge amount of data, which creates so much problem in maintaining the efficiency i.e. : how to provide important information quickly. Students often face trouble to find the new admission rule, where is the syllabus, and how to get good advice about placement. Traditional systems like FAQ pages which is static and strict website can only give small and general answers and are not good for real-world questions which are complex [1]. Also, the traditional chatbots are limited with less capabilities. They are not much efficient for multi-step thinking and for answering the complex questions [10]. Problems at the places like India where many language is spoken, is that they don't have good multilingual support system therefore cannot give a truly personal experience [14]. These system usually work in separate boxes, and fail to join all the big college data (from database, file, and chat logs) into one knowledge base [16]. To fix these bad gaps, we propose and make AbeSphere, an intelligent helper which is powered by GenAI and LLMs [2]. Our new approach is focused on joining the RAG (Retrieval-Augmented Generation) framework [15] to make sure facts are correct by grounding answers in real-time college data, and LangGraph [3] to help with advanced, multi-step smart thinking. This mix system allow the system to go to past simple keyword matching to do complex question solving [19]. The AbeSphere system structure is designed to process student questions through a complex LangGraph-based thinking process [3]. This process get the most relevant context from college data which is stored in a fast MongoDB database [4], which also manage user context and chat history for smooth continuous session. The RAG pipeline's main work[15] is to take out and mix information from college resources, which is then given using a smooth and responsive interface make with React JS [18]. The first goal of this project is to make a strong AI The successful making of AbeSphere is aimed to get a big cut in the average time it take to solve questions, decreasing it from a normal 15–30 minutes for human help to a quick 1–2 minutes, while at the same time automating and solving about 85% of routine college questions.

II. LITERATURE REVIEW

Our project offers a modern solution for students and faculty by facilitating quick problem-solving and efficient source of information for various mentioned problems [20]. These AI-powered system is designed to provide instantaneous, accurate, and personalized information. Our project aims to address the limitations of existing, often script-based or fragmented systems [10] by incorporating advanced Large Language Model (LLM) features [2] and the Retrieval-Augmented Generation (RAG) architecture [15] to ensure factual and complex query resolution. The following mentioned are the recent researches and efforts that highlight the requirements and effectiveness of intelligent assistants in higher education:

Enhanced Chatbot for University Admission and Educational Guidance 2025. [5] This study uses a Machine Learning (ML) and Natural Language Processing (NLP)-based chatbot, which was deployed on the LINE app , effectively. A hybrid system which was tested on 100 students - demonstrated high performance, achieving 97% accuracy in query resolution and 92% user satisfaction.

61A Bot Report: AI Assistants in CS1 Save Students Homework Time and Reduce Demands on Staff.2025. [6] This research deployed a GPT-4 based homework assistant integrated into an autograder for Computer Science 1 (CS1). After launching this initiative with more than 2000 students, it clearly showed that the work pressure on the staff decreased efficiently, and students became more efficient at finishing their assignments.

BARKPLUG V.2.2024. [9] This project uses the intelligent assistant and LLM + RAG pipeline specifically designed for campus resources. The system was rigorously evaluated using RAGAS (a framework for RAG evaluation) and the System Usability Scale (SUS). The results of this project shows the enhanced resource discovery capabilities along with high usability scores.

Intelligent Chatbot for Admission in Higher Education.2023. [11] This study focused on an admission-related chatbot developed using the Botsify platform. When tested with 42 postgraduate (PG) students, it achieved a Customer User Quality (CUQ) score of 76.6 and demonstrated 91% accuracy specifically for admission-related queries.

NEU-chatbot: Chatbot for admission of National Economics University.2021. [13] This system was developed using the RASA framework integrating the Recurrent Neural Networks (RNN) and BERT. As a retrieval-based model, it achieved a 97.1% accuracy when tested with over 50 students regarding admissions.

Virtual Assistants for Learning: A Systematic Literature Review.2021. [14] This project's literature review focuses on the use of Virtual Assistants (VAs) in higher education. The result concludes that VAs have a positive impact on learning outcomes and student productivity by reducing their workload.

Including all these paper's studies collectively, it demonstrate the effectiveness of AI chatbots and intelligent assistants in the today's educational sector [1, 20]. They also critically highlight the need for improving accuracy through advanced architectures like the RAG framework [15] and the necessity of incorporating advanced reasoning capabilities (like those provided by LangGraph [3]) to handle the complexity and needs of the real-world academics and administrative burden , our project is positioned to build upon this foundation by integrating these cutting-edge features.

II. PROPOSED METHODLOGY

A. System Architecture : The AbeSphere system consists of six components that are:

Data Ingestion Pipeline: It collect college information from websites, PDF files, databases, and event schedules through automatic collection, cleanup, and organization [16].

Backend Processing (FastAPI + LangGraph): This works like the brain of the system. It helps the AI to think, handles requests, keeps the conversation going in order, and connects all parts of the system together.[17, 3].

LLM Integration: Here the AI language model is connected, so it can give smart answers. It follows the instructions and also checks the answers a little to make them correct.[2]

RAG System:This mixes searching and answering. First it finds the important documents, then it adds that information in the AI prompt, so the answer becomes more accurate and helpful.[15]

MongoDB Storage:This stores the chats, user data, and the whole conversation history. This helps to give better experience next time[4].

User Interface (React): This gives a simple and easy chat window for users. It has things like typing animation, quick updates, image/video support, and is also helpful for users who have disabilities[18].

B. Data Flow Pipeline: The system processes queries in the following given steps:

Step 1: Query Reception - Here, user can submits their queries to React frontend [18], which is validated by user context.
Step 2: Classification - LangGraph use to classify query [3] into categories likes: Admission, Academic, Placement, Events, Facilities, and General.
Step 3: Context Retrieval - The RAG system finds the important documents by using vector search. It checks which document is more relevant and then shows the ones first.

Step 4: Response Generation -The LLM gets the user's question, the context, and the chat history. Then it thinks in many small steps and creates the final answer.
Step 5: Validation and Delivery -The answer goes through some quality checks. After that, it gets saved in MongoDB and then the response is sent to the user.
Step 6: Learning – It takes user feedback and analysis analytics data to find useful behaviour and based on this knowledge the base is regularly updated [12].

C. System Diagrams:

Use Case Diagram: This Use Case Diagram shows what the AbeSphere system can do and how it interacts with three main users: Student, Faculty, and Administrator:

Student :A student mainly looks for information and guidance. The main things a student can do are: check admission info, see syllabus, get placement guidance, view events, and ask questions about facilities.
Faculty : Faculty helps by giving content and keeping things updated. Their main use cases are: answer student queries, update academic information, manage resources, and see analytics like what type of questions students ask.
Administrator : The admin controls and manages the whole system. Their main work is: manage data, monitor usage, change settings, and create reports about system performance.

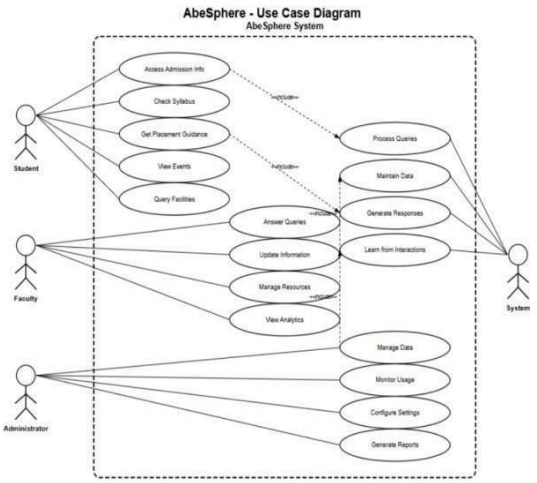


Fig1: Use Case Diagram

Flowchart Diagram: Flowchart diagram illustrates the query processing flow of the AbeSphere system, which is based on the LangGraph and RAG architectures, it also gives the information about how an end user will interact with the system and the end user may be faculty, student or a random learner.
The architecture is given as:
Input & Validation: The system receives the user query and performs its initial validation.
LangGraph Routing: The LangGraph Intent Router analyzes the query's intent and routes it through two parallel paths:
RAG Retrieval: For factual information (documents, rules), data is extracted from the RAG pipeline. MongoDB Context:For giving a more personal experience, the system takes the session history and user context from MongoDB.

Response Generation: The information taken from Retrieval and Context is sent to the LLM, and then the LLM creates the final answer.

Quality Check & Delivery: The answer goes through some checking to make sure it is fine. After that, it is sent to the user through the React UI.

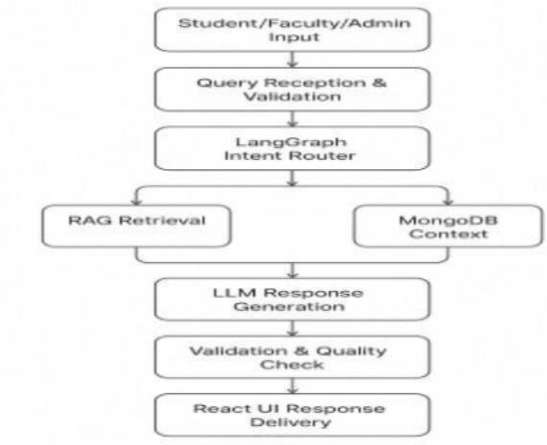


Fig2: Flow Chart Diagram

III. Technology Stack

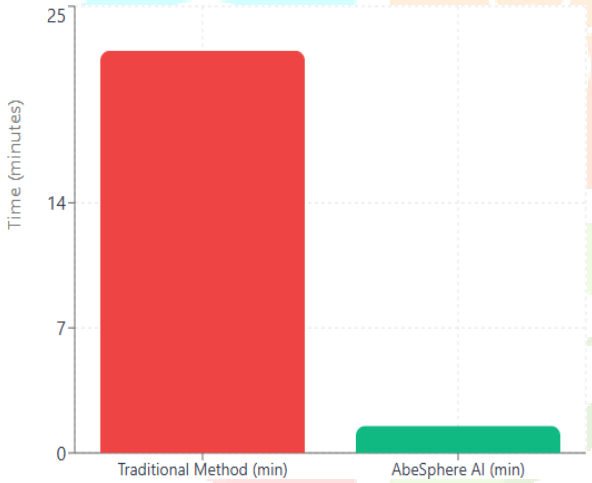
Our project is built upon a cutting-edge technology stack specifically chosen to ensure the intelligent assistant is robust, accurate, and high-performing.
This system's core reasoning engine is powered by LangGraph, which is utilized for multi-step reasoning. This feature is important because it helps the system handle hard queries. It can do complex work and make decisions by taking information from many sources at the same time.The system uses MongoDB to store its memory. Its purpose is twofold: it efficiently stores the vast amounts of institutional data, and more importantly, it maintains the session context and chat history for each user. This helps the system continue smoothly and give more personal interactions that match the user's academic journey.
For the development of quick, reliable responses and overall high performance of the application, FastAPI serves as the framework for the backend services.
To guarantee the factual accuracy and trustworthiness of the responses, we have integrated the Retrieval-Augmented Generation (RAG) Architecture. The primary benefit of RAG is that it reduces hallucination (when an LLM generates plausible but incorrect information) by grounding the model's output in verifiable institutional data.
The Frontend User Interface (UI) is developed using React. This choice ensures a smooth, modern chat experience for the user with real-time responses and a responsive design.
Finally, Python Pipeline manages the entire data ingestion process. Its function is to automate the collection and structuring of all the disparate

institutional data (documents, databases, logs) into a unified and ready-to-use knowledge base for the RAG system.

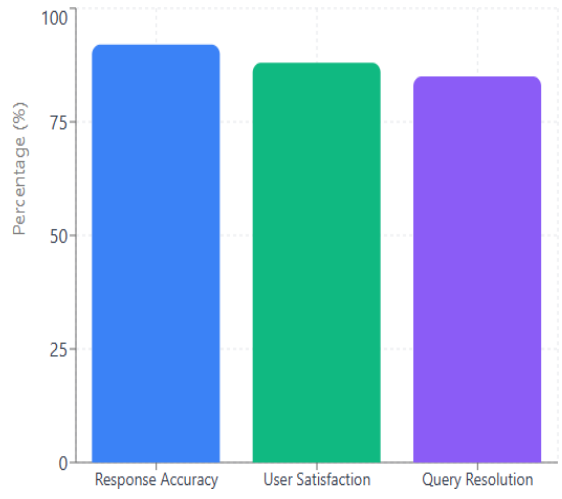
IV. RESULT ANALYSIS

The platform primarily assists students by addressing the issue of obtaining information within the college. Often students are unsure where to inquire, who to approach or they receive responses from various individuals. However this system provides clear and mostly accurate answers ensuring students don't lose time. It also operates efficiently and seamlessly making the experience straightforward, for all users.

A. Technical Performance: LangGraph take 2-3 seconds for easy questions and 5-8 seconds for hard questions that need many steps. RAG system can find right documents maximum times and give wrong information only 5% time(approx), which is much better than normal LLM that give wrong answer 20-30% times.The RAG system is also effective since it locates the documents in the majority of instances. The MongoDB database is extremely quick well. It can display chat history in than 100 milliseconds making it seem nearly instantaneous. Additionally it can handle over 500 users chatting concurrently without any decrease, in speed.



Graph1: It shows Average Query Resolution Time



Graph2: System Performance Metrics

What System Can Do: Our system help students in many things like admission process, seeing syllabus and academic rules, exam schedules, placement information, event details and campus facilities like library hostel and sports area etc.The system supports students across domains. Additionally it provides placement details announcements about events and workshops along with information regarding campus amenities such as the library, hostel, sports facilities, labs and, beyond.This transforms the system into a destination, for any college-related questions significantly reducing the time and effort required by student.

B. User Results:.Based on user feedback 92% of individuals stated that the responses provided by the system are accurate and helpful. 88% Of students mentioned a preference for using the AI assistant over conventional methods such, as consulting seniors or contacting departments.Previously numerous students spent 15–30 minutes searching for answers whereas now they obtain the information within only 1–2 minutes. Additionally 85% of inquiries are resolved without any human assistance indicating the system's dependability.

C. Problem We Solved: To ensure data accuracy the system verifies information from sources and chooses the most reliable one. For queries LangGraph divides the main question, into smaller segments and addresses them sequentially.To ensure data accuracy the system verifies information from sources and chooses the most reliable one. For queries LangGraph divides the main question, into smaller segments and addresses them sequentially.To remember past conversations, MongoDB stores the complete chat history and also creates summaries of the important parts. This helps the system give more personal answers and know what the user was talking about earlier.

V. LIMITATIONS AND FUTURE SCOPE

A. Current Limitations:

Technical: Complex queries take 5-8 seconds and requires continuous updates and dependent on external sources .

Functional: Limited visual content (no video or audio content) and some real-time data need manual updates.

Scope: It focuses on single institution only, using only english for the communication. Having web interface only and we are planning for the mobile app also.

B. Future Enhancement:

Phase 1 (3-6 months): We are planning for the full Hindi support along with regional languages too. Enhanced personalization with LMS integration, email/SMS notifications, response caching, query prediction optimization etc.

Phase 2 (6-12 months): We are planning for it's availability on native mobile apps (iOS/Android) having the facilities of push notifications, offline mode, voice input/output, image recognition, PDF table extraction, analytics dashboard, usage patterns, proactive assistance, automated reminders etc.

VI. CONCLUSION

AbeSphere demonstrates how LLM and RAG technology can truly transform the way students access information within a college. Previously students needed to explore sources consult multiple individuals or wait extensively to obtain straightforward answers. However AbeSphere. Accelerates this entire procedure. A major accomplishment of the system is that it provides about 92% answers resolves 85% of questions automatically and significantly saves time for both students and faculty.

AbeSphere demonstrates how LLM and RAG technology can truly transform the way students access information within a college. Previously students needed to explore sources consult multiple individuals or wait extensively to obtain straightforward answers. However AbeSphere. Accelerates this entire procedure. A major accomplishment of the system is that it provides about 92% answers resolves 85% of questions automatically and significantly saves time for both students and faculty.

Benefits: Students have the ability to pose questions whenever they choose—be it morning, night or during holidays—and receive assistance. They aren't reliant on teachers or office personnel, for every issue. Teachers benefit well experiencing significant relief since they no longer must answer the same repeated inquiries repeatedly.For the college administration the system cuts down an amount of additional effort. They can lower expenses organize tasks efficiently and obtain valuable insights straight from the data. This also minimizes a deal of paperwork as most processes transition to digital.

This project demonstrates that AI is capable of assisting individuals by responding to typical and everyday questions. For complex issues it still enables actual humans to intervene ensuring the assistance stays dependable.Applying RAG is crucial in education since students require responses rather than arbitrary guesses. Tailoring answers for each learner also aids their comprehension, which is a major factor, in why students appreciate using the AI platform.Looking ahead we aim to introduce features that tailor the system more closely to each individual student. We also plan to investigate the extent to which the system effectively supports students in enhancing their learning and minimizing their uncertainty.The main goal of AbeSphere is to make an AI system that works like a smart and patient teacher—someone who is always ready to help students anytime during their college life.

VII. REFERENCES

[1]. Chen, M., et al. "Evaluation of AI Chatbots in Education." *Computers & Education*, vol. 195, 2023.

[2]. OpenAI. "ChatGPT API Documentation." 2024.

[3]. LangChain AI. "LangGraph Documentation." 2024.

[4] MongoDB, Inc. "MongoDB Official Documentation." 2024.

[5] Ramírez-Montoya, M.S., et al. "An Enhanced Chatbot for University Admission." *Int. J. Educational Technology*, vol. 22, 2025.

[6] Denny, P., et al. "61A Bot Report: AI Assistants in CS1." ACM Learning @ Scale, 2025.

[7] Kumar, V., Sharma, S. "SocratiQ: AI Powered Learning Companion." *J. Educational Computing Research*, vol. 63, 2025.

[8] Tan, S.C., et al. "Battling Botpoop using GenAI." Computers & Education: AI, vol. 6, 2025.

[9] Brown, T., et al. "BARKPLUG V.2: Informed Chatbot." IEEE Trans. Learning Technologies, vol. 17, 2024.

[10]Kasneci, E., et al. "Role of AI chatbots: systematic review." npj Science of Learning, vol. 9, 2024.

[11] Ahmad, S., Khan, M. "Intelligent Chatbot for Admission." Int. J. Advanced CS Applications, vol. 14, 2023.

[12] Okonkwo, C.W., Ade-Ibijola, A. "Virtual teaching assistant impact." Int. J. Educational Technology, vol. 19, 2022.

[13] Nguyen, H.T., et al. "NEU-chatbot." J. Computer Science and Cybernetics, vol. 37, 2021.

[14] Wollny, S., et al. "Virtual Assistants: Systematic Review." Education and Information Technologies, vol. 26, 2021.

[15] Lewis, P., et al. "Retrieval-Augmented Generation." NeurIPS, vol. 33, 2020.

[16] Mitra, S., Sharma, K. "Data Collection for Educational AI." Int. J. AI in Education, vol. 32, 2022.

[17] Sebastián, R. "FastAPI Documentation." 2023.

[18] Facebook. "React Official Documentation." 2023. [19] Vaswani, A., et al. "Attention Is All You Need." NeurIPS, vol. 30, 2017.

[20] Chen, M., et al. "AI Chatbots for Higher Education." 2023.