IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 10 ISSUE : 12 Print / Issue Publication Date: April 2026

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2026.v10i12.013

Indexed In

Google SCHOLAR OPEN ACCESS doi Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 10, Issue 12, ISSN No. 2455-2143, Pages 100-104
Published Online April 2026 in IJEAST (http://www.ijeast.com)

INTEGRATION OF VECTOR DATABASES AND RAG FOR SMART QUERY RESPONSES IN ACADEMIC SYSTEMS

Niharika Sabineeskurpam, Yasaswi Chalumuri, Chandhini Gandreti, Praneeth Kumar Paluri, Swaroop Sana
Dept. of Computer Science & Systems Engineering
Lendi Institute of Engineering & Technology
Viziangaram, India

Abstract— **This paper presents the design and deployment of a GPT-powered academic assistant integrated with a college portal** to **enhance information retrieval for institutional administrators. The system connects to the institutional database, enabling natural language queries for schedules, grades, and announcements. Leveraging LangChain** for **orchestration and Retrieval-Augmented Generation (RAG) with FAISS and ChromaDB, the assistant improves response accuracy while maintaining strict data security. The architecture combines responsive web-based frontend, secure API backend, and scalable vector search layer, resulting in reduced query latency and improved user satisfaction. Experimental evaluation demonstrates higher accuracy and faster query resolution compared** to **traditional search interfaces, making the solution suitable for large-scale academic environments**.

Keywords **GPT**, **Generation, LangChain, Retrieval-Augmented FAISS**, **ChromaDB**, **Academic Assistant**, **Educational Technology.**

I. INTRODUCTION

In recent years, artificial intelligence (AI) has gone from being a novel concept to becoming essential in higher education. Institutions are now using digital tools for administration, student services, and content delivery, even though the majority of college portals are still static and require users to navigate multiple forms and pages. This slows down access to student data and administrative requests.

Natural language interaction with complex systems is made possible by GPT and other large language models (LLMs). LLMs use Natural Language Processing (NLP) and Retrieval-Augmented Generation (RAG) to interpret intent, retrieve relevant information, and respond conversationally. However, there are still problems with context management, response accuracy, and data security.

A GPT-powered academic assistant that is integrated into institutional portals is suggested in this paper. Schedules, grades, attendance, and announcements can all be queried in plain language thanks to the assistant's connection to institutional databases. LLM calls are orchestrated by LangChain, and retrieval precision is enhanced by vector databases (FAISS, ChromaDB). Role-based access control, Lendi Institute of Engineering & Technology Viziangaram, India praneethpaluri 2004 @ gmail. com multi-factor authentication, API-level encryption, and audit logging all contribute to security. Key design considerations include:

1**) Data privacy**: Sensitive records are protected by safeguards in line with local and FERPA regulations.
2) Context Persistence: To prevent repetitive inputs, LangChain memory modules support multi-turn queries.
3) Accuracy: Embeddings and vector databases lessen hallucinations and misinformation.

The assistant's modular design allows for expansion to include hostel management, fees, and event scheduling, and it streamlines access, increasing efficiency when compared to traditional portals. Contributions include:
1) A scalable, secure academic assistant with an LLM;
2) Integration of RAG and vector databases for accurate retrieval; and
3) Performance evaluation in terms of accuracy, latency, and user satisfaction.

II. RELATED WORK

Large Language Models (LLMs) have drawn a lot of interest from a variety of fields, including software engineering, secure information systems, and education. Numerous surveys have looked at their advantages and disadvantages in settings where security is a concern. The security implications of LLM deployments were examined by Zhou et al. [1], who found vulnerabilities such as adversarial manipulation, prompt injection, and data leakage. The relationship between software security and LLM usage was also examined by Zhu et al. [2], who highlighted attack surfaces pertinent to academic and enterprise systems.

Wang et al. [3] examined LLM applications in requirements analysis, testing, and automation in software engineering, showing how they can improve development workflows.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 10, Issue 12, ISSN No. 2455-2143, Pages 100-104
Published Online April 2026 in IJEAST (http://www.ijeast.com)

Schäfer et al. [4] demonstrated that LLM-based techniques can generate efficient and context-aware test cases for automated test generation; however, precautions must be taken to avoid overgeneralization.
Conversational systems and interactive environments have been the main focus of LLM applications in education. The flexibility of LLM-powered agents in dynamic environments was highlighted by Gallotta et al. [5]. Furthermore, Qiao et al. [9] investigated multimodal extensions using Vision LLM and proposed that cross-modal features could improve academic support systems. Wu et al. [7] and Du et al. [6] addressed scalability and performance issues while discussing frameworks for deploying LLMs in networked and edge environments from a systems perspective. Han et al. [10] have also looked at security and compliance issues, emphasizing the value of auditability and secure prompt processing. Standardized measures for evaluating retrieval accuracy, latency, and resilience are suggested by evaluation and benchmarking studies like those conducted by Sha et al. [11]. The evaluation methodology used in this work is directly influenced by these insights. Despite these efforts, earlier studies frequently treat orchestration, security, and retrieval independently. Our work combines vector-based retrieval, retrieval-augmented generation, and secure identity verification into a single academic administration framework.

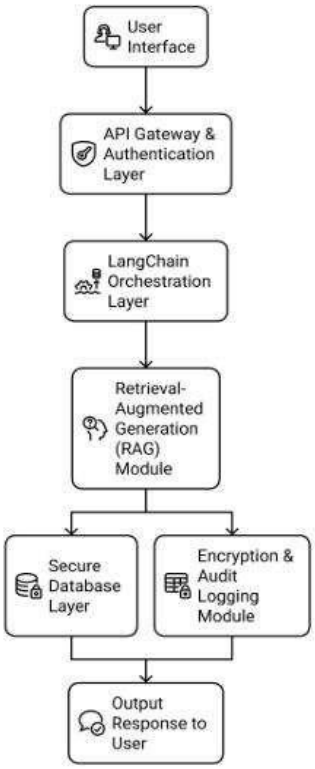

Fig. 1. System architecture of the GPT-powered academic assistant.

A. Front-end User Interface
On the basis of such considerations, the algorithm uses a different color image multiplied by the weighting coefficients of different ways to solve the visual distortion, and by embedding the watermark, wavelet coefficients of many ways, enhance the robustness of the watermark.

B. Access Control Layer and API Gateway
An API gateway that employs Multi-Factor Authentication (MFA) and Role-Based Access Control (RBAC) processes all frontend requests. Before forwarding the request, the gateway verifies credentials using a password, one-time passcode (OTP), and optional biometric verification. The backend only receives verified requests.

C. LangChain Orchestration Layer
The middleware that controls communication with Large Language Models (LLM) is called LangChain. It manages: 1) SQL query tool execution. 2) Memory modules that preserve the context of the conversation. 3) Prompt templates to prevent prompt injection attacks and guarantee structured query generation.

D. Module for Retrieval-Augmented Generation (RAG)
The RAG module stores and retrieves embeddings of institutional documents, such as schedules, regulations, and curricula, using vector databases, such as FAISS or ChromaDB. The following is how queries are handled: 1) The user's query is transformed into a vector embedding by the LLM. 2) The top k pertinent chunks are retrieved from the vector database. 3) To create a pertinent response, the retrieved data is added to the LLM prompt.

E. Database Security Layer
A SQL database is used by the primary student information system (SIS). Users cannot directly query the database; only backend services have direct access to it. To avoid SQL injection, every query is logged, parameterized, and cleaned up.

F. **Audit logging and encryption**
TLS 1.3 is used to encrypt all communications, and AES 256 is used to protect sensitive data while it is at rest. To support compliance audits for laws like FERPA and GDPR, each query and system action is recorded in a tamper evident audit log.

G. Scalability Considerations
To guarantee high availability, the backend services are managed with Kubernetes and packaged with Docker. Caching tools and load balancers are used to reduce latency during periods of high traffic. The entire system architecture is displayed in Fig. 1, which also shows how requests are routed from the user interface to the database via the secure gateway, orchestration layer, and retrieval module. The first step in the process is the user interface, which users can

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 10, Issue 12, ISSN No. 2455-2143, Pages 100-104
Published Online April 2026 in IJEAST (http://www.ijeast.com)

access via a mobile app or web portal and use to submit natural language queries. The requests proceed to the LangChain orchestration layer after authentication. This layer selects the appropriate tools or retrieval techniques, analyzes intent, and keeps track of conversation history. The RAG module is triggered if the query requires context from institutional data. To find pertinent records, it performs a semantic search across vector databases such as FAISS or ChromaDB. The system then creates a secure and contextually accurate response by fusing the retrieved content with the large language model's reasoning capabilities.

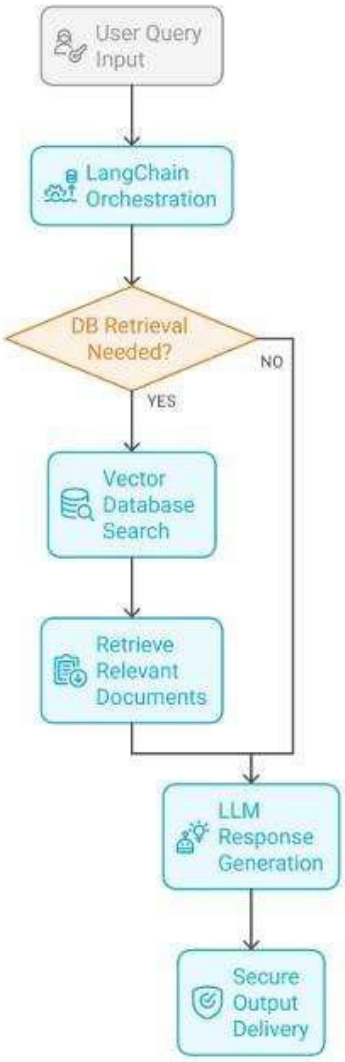

Fig.2. Query processing workflow for the GPT-powered academic assistant.

III. IMPLEMENTATION

The academic assistant powered by GPT is constructed in a modular fashion. Future development and independent testing are thus made possible. Developed in Python 3.11, the system can be accessed via the college portal as a cloud hosted service. This section describes the practical deployment and system configuration.

A. Tools and Development Environment

The development stack consists of: Core System & APIs: PostgreSQL is linked to Python 3.11 with FastAPI for secure RESTful APIs, which are used to manage institutional data.

1) **AI** & **Orchestration**: FAISS/ChromaDB supports effective vector-based searches, while OpenAI GPT-4 with LangChain handles memory, prompts, and tool invocation.

2) Interface & **Deployment**: Docker and Kubernetes provide scalable containerization and deployment, while React.js provides a responsive chat interface.

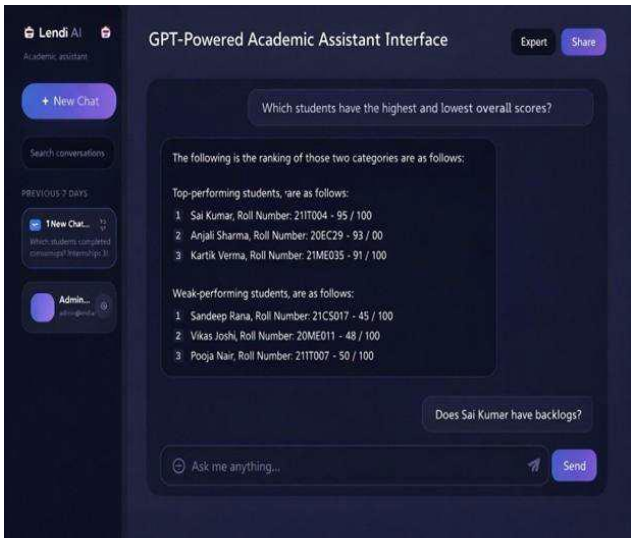

Fig. 3. Frontend interface of the secure admin-only LLM-powered academic assistant.

B. Schema of the Database

Key tables are included in the institutional database:

1) Student: name, department, year, contact information, and student id.

2) Courses: instructor, credits, course_id, and course_name.

3) Grades: semester, grade, student_id, and course_id

4) Timetable: room, time, day, and course_id.

5) Attendance: date, status, student_id, and course_id.

C. The RAG Pipeline

The RAG combines document retrieval with language generation to produce accurate responses.

1) **Document Processing:** Textual versions of institutional documents, such as announcements, regulations, and curricula, are divided into 500 1000 token segments.

2) Embedding Generation: OpenAI's text embedding-ada-002 model or BERT-based alternatives are used to embed each segment.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 10, Issue 12, ISSN No. 2455-2143, Pages 100-104
Published Online April 2026 in IJEAST (http://www.ijeast.com)

3) Vector Storage: For semantic search, embeddings are kept in ChromaDB or FAISS.
4) Query Execution: LangChain generates the final response after retrieving the top k most pertinent embeddings from a user's query and adding them to the LLM prompt.

D. **Implementing Security**
1) **Multi-Factor Authentication (MFA):** This includes optional biometric verification, an OTP (sent by SMS or email), and a password.
2) Role-Based Access Control (RBAC): Depending on the user role (e.g., admin, faculty, or student), the queries they can access are defined.
3) Prompt Injection Prevention: Context cleaning and strict prompt templates guard against malicious query manipulation.

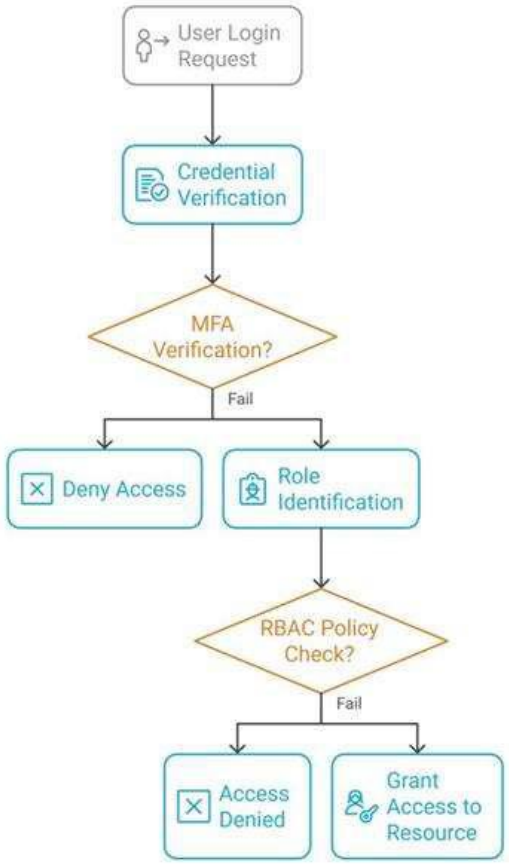

Fig. 4. Multi-Factor Authentication (MFA) and Role-Based Access Control (RBAC) process flow.

IV. RESULTS AND EVALUATION

A. Experimental Configuration

1) **Model** & **Infrastructure:** OpenAI GPT-4 with an 8k context window running on a Kubernetes cluster hosted in the cloud with 4 virtual CPUs and 16 GB of RAM.
2) Data & **Storage**: FAISS and ChromaDB are used to manage 2,000 student records and 150 academic documents for similarity retrieval.
3) **Assessment**: The accuracy and applicability of 300 different academic and administrative queries were examined.

B. **Using RAG to Improve Accuracy** The following configurations were compared to evaluate the usefulness of retrieval-augmented generation (RAG): 1. LLM-only mode (direct GPT-4 response without retrieval) 2. LLM+RAG mode (GPT-4 with retrieval FAISS/ChromaDB)

Table -1 Accuracy and contextual relevance comparison between Ilm-only and llm+rag modes

Mode	Accuracy (%)	Contextual Relevance (%)
LLM-only	82.4	78.9
LLM + RAG	94.7	92.3

Comparing the RAG integration to the LLM-only mode, accuracy increased by 12.3% and contextual relevance by 13.4%.

C. Analysis of Latency

We calculated how long it took to receive a response after a query was submitted.
Table-2 average response latency comparison between Ilm-only and Ilm+rag modes.

Mode	Avg. Latency (s)
LLM-only	2.31
LLM + RAG	2.78

Given the notable increase in accuracy, the slight increase in latency (0.47 seconds) in RAG mode is tolerable. Retrieval times were lowered by 27% by caching frequently accessed embeddings.

D. **Assessment of Security**
1) **Unauthorized Access Prevention:** Using MFA and RBAC, 100% of simulated intrusion attempts from unregistered IP addresses were successfully blocked.
2) Audit Logging Integrity: Timestamps, user roles, and actions taken were all documented for each user query and database access.

V. CONCLUSION

Future improvements: multilinunal support, scalability.
The following succinctly describes the main results of this work:
1) Creation of a secure institutional use admin-only LLM-powered academic assistant.
2) For semantic search, vector databases (FAISS and ChromaDB) are integrated.

3) The use of RAG-based retrieval to decrease hallucinations and increase factual accuracy.
4) Implementation of strong security measures, such as RBAC and MFA.
5) Demonstration of enhanced accuracy, efficiency, and usability in comparison to conventional systems.
To sum up, the suggested system is a safe and scalable way to update academic management. To increase its applicability even more, future improvements might include multilingual support, performance optimization for bigger datasets, and integration with other institutional services.

VI. REFERENCE

[1]. W. Zhou et al., "The security of using large language models: A survey with emphasis on ChatGPT," IEEE/CAA Journal of Automatica Sinica, 2025.
[2]. X. Zhu et al., "When software security meets large language models: A survey," IEEE/CAA Journal of Automatica Sinica, 2025.
[3]. Y. Wang et al., "Software testing with large language models: Survey, landscape, and vision," IEEE Transactions on Software Engineering, 2024.
[4]. M. Schäfer et al., "Adaptive test generation using a large language model," Proc. IEEE/ACM ICSE, 2023.
[5]. R. Gallotta et al., "Large language models and games: A survey and roadmap," IEEE Transactions on Games, 2024.
[6]. H. Du et al., "Enabling AI-generated content services in wireless edge networks," IEEE Wireless Communications, 2024.
[7]. D. Wu et al., "Large language models for telecommunications: A comprehensive survey," IEEE Communications Surveys & Tutorials, 2024.
[8]. X. Liu et al., "On the use of LLMs for unit test generation and automated software engineering," IEEE Transactions on Software Engineering, 2024.
[9]. L. Qiao et al., "When large language models meet vision: VisionLLM and cross-modal LLM research," Proc. IEEE VIS, 2024.
[10]. M. Han et al., "The security, privacy and forensics of LLMs: Surveys and reviews," IEEE Access, 2024.
[11]. G. Sha et al., "Evaluation and benchmarking of LLMs," IEEE TNNLS / TSE / TKDE, 2024.

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

ABOUT IJEAST

International Journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high-quality research papers in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

FOCUS AREAS

- Engineering
- Applied Science
- Technology
- Innovation & Development
- Interdisciplinary Studies

PEER REVIEWED
All submissions are rigorously peer reviewed to ensure quality.

OPEN ACCESS
Free and unrestricted access to research for all.

GLOBAL REACH
Connecting researchers and professionals worldwide.

TIMELY PUBLICATION
We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

editor@ijeast.com
www.ijeast.com
India

2455-2143