iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.76860

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Optimizing Cross-Lingual Information Retrieval in Campus Administration using Retrieval-Augmented Generation and Semantic Chunking

Harsh Singh[1], Garvaansh Gupta[2]
Department of CSE (AI & ML), Oriental Institute of Science & Technology, Bhopal, India

Abstract: ***Digital transformation in Indian higher-education institutions is constrained not by the absence of information, but by the difficulty of accessing it across linguistic and structural boundaries. Administrative data such as admission rules, fee structures, examination schedules, and scholarship policies are published primarily in English and distributed across heterogeneous document formats, while students interact using Hindi, regional languages, and mixed Romanized scripts such as Hinglish. This paper presents an optimized Retrieval-Augmented Generation (RAG) architecture designed as a campus-scale natural language information system rather than a simple chatbot. The proposed framework integrates multilingual semantic embeddings, vector-based document retrieval, conversational state management, and grounded response generation into a unified, auditable architecture. A hybrid two-tier backend separates high-frequency user interaction from computationally intensive retrieval and inference, enabling scalable deployment across multiple institutions. Experimental evaluation demonstrates that the architectural design achieves high retrieval accuracy and low latency while preserving factual reliability, making it suitable for real-world administrative decision support in multilingual academic environments.***

Keywords: ***Retrieval-Augmented Generation, Multilingual Information Retrieval, Semantic Embeddings, Campus Information Systems, Vector Databases, LangChain, ChromaDB, Conversational AI, Knowledge Grounding***

I. INTRODUCTION

A. Motivation

Indian universities operate within a highly diverse linguistic ecosystem. Although official administrative documents such as circulars, admission brochures, scholarship guidelines, and fee notifications are issued in English, the student population commonly communicates in Hindi, regional languages, or Romanized mixed-script forms such as Hinglish. This mismatch creates a significant access barrier, where information technically exists but is functionally inaccessible to many students.

As a result, even routine queries such as fee deadlines, eligibility rules, or examination schedules often require students to physically visit administrative offices or rely on informal peer networks. This leads to congestion at help desks, delays in information delivery, and increased workload for administrative staff. While universities have adopted digital portals, these platforms largely consist of static PDF repositories or keyword-based search systems, which are poorly suited for natural language queries and conversational clarification.

Recent advances in large language models and Retrieval-Augmented Generation offer the potential to convert institutional documents into interactive knowledge systems. However, most existing deployments treat these technologies as chatbot layers rather than as part of a structured information architecture. For high-stakes academic data, where incorrect information can affect admissions, finances, or academic progression, architectural rigor and grounding are essential.

TABLE I. EXAMPLES OF MIXED-SCRIPT STUDENT QUERIES AND THEIR CORRESPONDING ADMINISTRATIVE INTENT

STUDENT QUERY (MIXED SCRIPT / INFORMAL)	INTENDED ADMINISTRATIVE MEANING
"Mujhe post matric scholarship ka last date batao"	Deadline for Post-Matric Scholarship application
"Fee kabbharnihai is semester ki?"	Semester tuition fee payment schedule

©IJRASET: All Rights are Reserved

“Isme eligibility kyahai?”	Eligibility criteria for a previously mentioned scheme
“Mera exam form kabtak fill karsaktehain?”	Examination form submission deadline
“Hostel allotment ka process kyahai?”	Procedure for hostel allocation

B. Problem Statement

Conventional campus information systems are designed around keyword search and form-based navigation. These systems assume that users know the exact terminology used in official documents and can navigate multiple PDFs to locate relevant clauses. In practice, students express queries in informal language, use transliteration, and refer to previously mentioned topics through pronouns or abbreviations, making keyword-based systems ineffective.

While large language models can generate fluent responses, they are not inherently grounded in institutional documents. Without controlled retrieval and verification, such models may hallucinate incorrect dates, rules, or eligibility conditions. In academic environments, even small factual errors can have serious consequences. The core challenge is therefore not simply generating answers, but designing an architecture that can reliably map multilingual natural-language queries to authoritative institutional documents and return contextually accurate, verifiable responses.

C. The Administrative Bottleneck

In the Indian higher education ecosystem, administrative efficiency is often inversely proportional to institutional size. A typical Tier-3 college with 2,000+ students often relies on a centralized administrative cell handled by fewer than five staff members. During peak periods—such as exam form submission or scholarship verification—the ratio of student queries to available staff creates a "Denial of Service" effect in the physical world.

Students forced to wait in queues for minor clarifications (e.g., "Is the OBC scholarship deadline extended?") lose valuable academic time. Furthermore, the reliance on notice boards and static PDFs means that information dissemination is passive. There is no active mechanism to "push" personalized answers to students, leading to a reliance on informal, often inaccurate, peer-to-peer information networks (e.g., unofficial WhatsApp groups).

D. Limitations of Existing Chatbots

Most deployed campus chatbots fall into two categories. Rule-based bots rely on predefined flows and FAQs, making them brittle and unable to handle variations in phrasing or multi-turn clarification. On the other hand, LLM-based chatbots provide flexible language interaction but lack reliable grounding in official data.

These systems also fail to handle cross-lingual and mixed-script queries, as they typically depend on translation pipelines or keyword matching. Translation introduces semantic drift, while keyword search ignores intent. Additionally, existing bots do not preserve conversational state effectively, causing follow-up questions such as “What is the last date for it?” to be misinterpreted when context is not retained. Most importantly, current chatbot designs do not provide auditable source attribution.

E. Contribution

This paper details the optimization of a RAG-based framework, "CampusMitra," specifically for the Indian academic context. The core contributions include:

1) Script-Agnostic Intent Mapping: We implemented a direct semantic alignment between regional scripts and English context to bridge the linguistic divide.
2) Contextual Persistence: By implementing conversational state, the system reduces retrieval entropy and handles multi-turn administrative dialogues effectively.
3) Two-Tier Hybrid Architecture: A distributed backend (Node.js and FastAPI) designed to handle high-frequency omnichannel traffic while maintaining CPU-intensive vector operations.
4) Auditable Grounding: The system enforces source-attributed responses to ensure 100% factual reliability in high-stakes administrative data.

II. RELATED WORK

A. Literature Review

Retrieval-Augmented Generation was introduced to overcome the static knowledge limitation of large language models by combining neural text generation with external document retrieval. Systems such as those proposed by Lewis et al. [1] and subsequent embedding-based retrievers [2] demonstrated that grounding language models in vector databases improves factual accuracy in knowledge-intensive tasks.

Multilingual sentence embedding models, including transformer-based architectures trained on cross-lingual corpora, enable semantic alignment across different scripts and languages [3]. Prior research has shown that embedding-based retrieval is more robust to paraphrasing and transliteration than keyword search, making it suitable for heterogeneous linguistic environments.

In the domain of institutional information systems, previous work has focused largely on portal design or FAQ automation. Few systems integrate semantic retrieval, conversational memory, and grounding into a unified architecture capable of supporting real-time administrative decision support.

B. Proposed Architectural Direction

This work frames RAG not as a chatbot add-on but as a campus-scale knowledge infrastructure. The architectural direction emphasizes four core layers: document ingestion and semantic chunking, multilingual embedding generation, vector-based retrieval with metadata isolation, and grounded language model synthesis.

A hybrid two-tier backend is proposed, where a high-throughput I/O layer handles user sessions and communication channels, while a dedicated inference layer performs embedding computation, similarity search, and controlled response generation. This separation ensures scalability, reliability, and institutional data isolation.

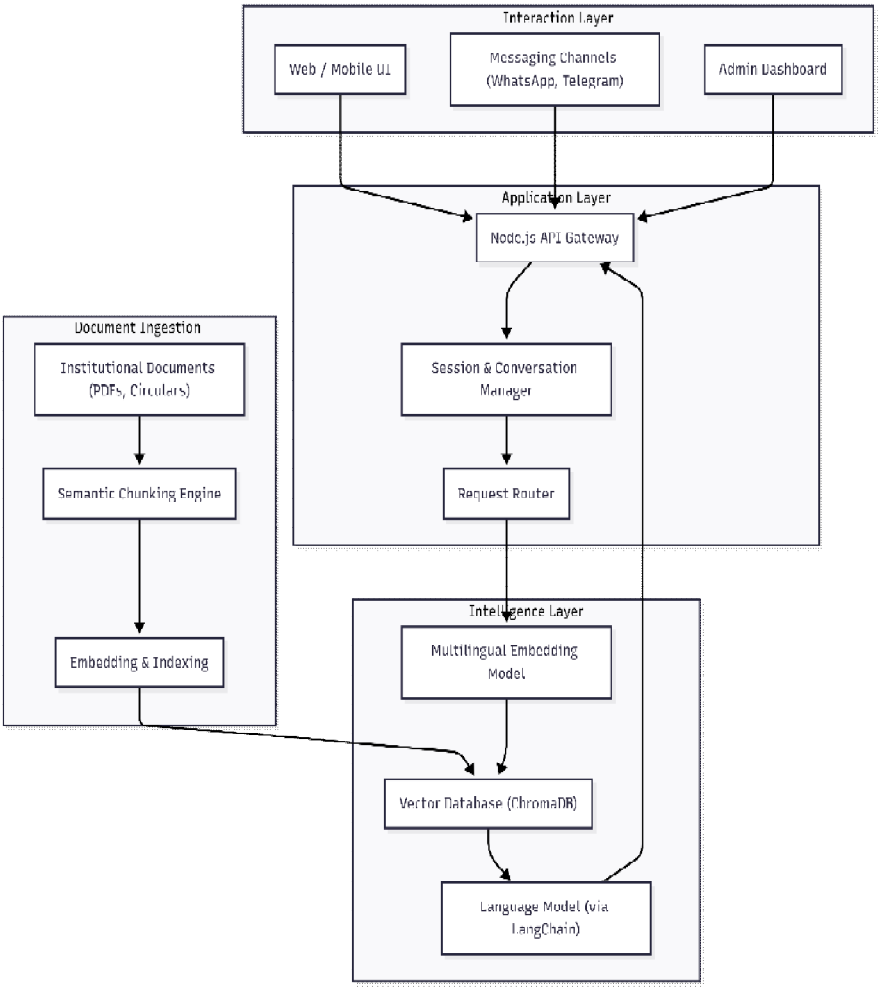


Fig. 1. Layered Logical Architecture of the Proposed RAG-BasedCampus Information System

C. Gap Analysis

Existing RAG implementations focus primarily on improving answer quality but do not address deployment constraints such as multi-tenancy, auditability, and safety in high-stakes domains. In the context of campus administration, errors cannot be tolerated, and responses must be traceable to authoritative sources. Furthermore, most current systems treat multilinguality as a translation problem rather than an embedding alignment problem, leading to loss of semantic precision. There is a clear gap between general-purpose conversational AI systems and the requirements of institutional information infrastructure.

III. PROPOSED METHODOLOGY

A. Semantic Chunking Strategy

To maintain the integrity of administrative context, we implemented a Recursive Character Splitting strategy. Unlike fixed-size chunking, this method respects logical boundaries like paragraphs and sentences. We utilized a 1000-character chunk size with a 200-character (20%) overlap.

This strategy was adopted because administrative documents often contain dense lists and eligibility clauses. The 20% overlap ensures that if a critical fact (e.g., a scholarship deadline) spans a split point, it remains retrievable in both adjacent vectors, maintaining contextual continuity.

B. Linguistic Agnosticism and Intent Mapping

The system is Intent-Aware rather than language-aware. By utilizing the high-density Stella-v5 multilingual embedding model, we map queries from any Indian script into a unified 768-dimensional vector space. This optimization eliminates the need for a noisy transliteration layer, as the model aligns the "Meaning" of a Hinglish query directly with the "Meaning" of an English document chunk.

The similarity between a query embedding q and a document embedding d is computed as

$$cos(q, d) = \frac{q \cdot d}{\| q \| \| d \|}$$

C. Contextual Persistence (Session Memory)

To handle the conversational nature of campus queries, our framework conditions the current retrieval query on the dialogue history. By injecting the last N messages into the query-refinement phase [4], we reduce retrieval entropy. This allows the system to resolve ambiguous references (e.g., "When is the last date for it?") based on the previously identified subject (e.g., "Post-Matric Scholarship").This conditioning step transforms under-specified follow-up queries into fully grounded retrieval queries.

D. Uncertainty Handling and Decision Gates

In campus administration, a wrong answer regarding fees or exams is a liability. We implemented a Confidence-Aware Retrieval gate. If the Cosine Similarity score (τ) of the top-retrieved chunk falls below a predefined threshold (e.g.,$\tau < 0.7$), the system triggers a Smart Escalation Path to a human administrator rather than generating a guess.

E. RAG vs. Fine-Tuning for Administrative Data

We deliberately chose a Retrieval-Augmented Generation (RAG) architecture over fine-tuning a Large Language Model (LLM). Administrative rules change frequently (e.g., semester dates, fee structures). Fine-tuning a model like Llama-3 or GPT-4 on institutional data would "bake in" this knowledge, rendering the model obsolete the moment a new circular is released.

In contrast, our RAG pipeline decouples knowledge from reasoning. The vector database *(ChromaDB)* acts as a dynamic external memory. When a new fee notice is released, we simply embed the PDF and update the vector store—a process taking seconds—without retraining the neural network. This ensures *"Day-Zero"* accuracy for new notifications.

IV. SYSTEM IMPLEMENTATION

A. Hybrid Two-Tier Backend

The system was implemented using a hybrid backend composed of two independent services. A Node.js–based interaction layer handles all user-facing traffic, including HTTP requests, session management, and multi-channel communication. This layer acts as the system's API gateway and ensures scalability under high query loads.

A Python FastAPI service forms the inference layer, responsible for embedding generation, similarity search, and language model orchestration. This separation ensures that computationally expensive vector operations and model inference do not interfere with real-time user interaction, allowing both layers to be scaled independently.

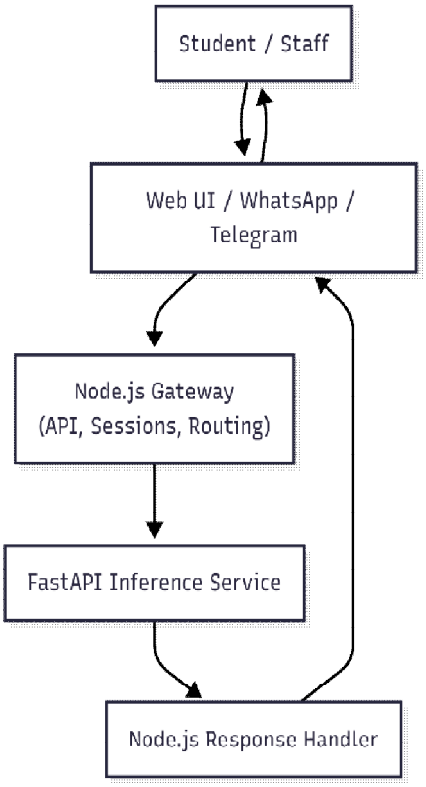


Fig. 2. Two-Tier Backend Architecture for Query Processing

B. Vector Database and Metadata Isolation

ChromaDB was used as the vector database to store document embeddings and metadata. Each document chunk is stored along with institution-specific metadata such as college ID, document category, and timestamp. During retrieval, metadata filtering ensures that only documents belonging to the requesting institution are searched, preventing cross-tenant data leakage. This design allows multiple universities to be hosted on the same system instance while preserving strict data isolation, which is a key requirement in academic governance [5].

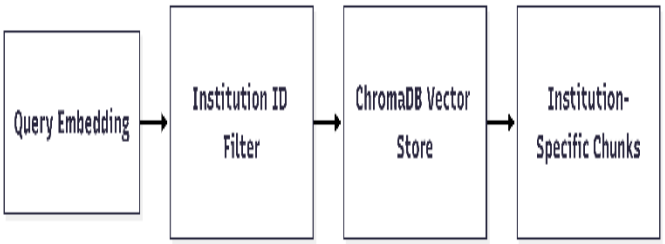


Fig. 3. Metadata-Aware Vector Retrieval in the Campus Knowledge Base

C. Langchain-Based Orchestration

LangChain was employed to manage the RAG workflow. It coordinates query refinement, conversational memory injection, document retrieval, and response synthesis. The language model is constrained to operate only on retrieved document context, which ensures that every answer is grounded in authoritative institutional data. Each generated response includes source references to the originating documents, enabling verification and improving institutional trust.

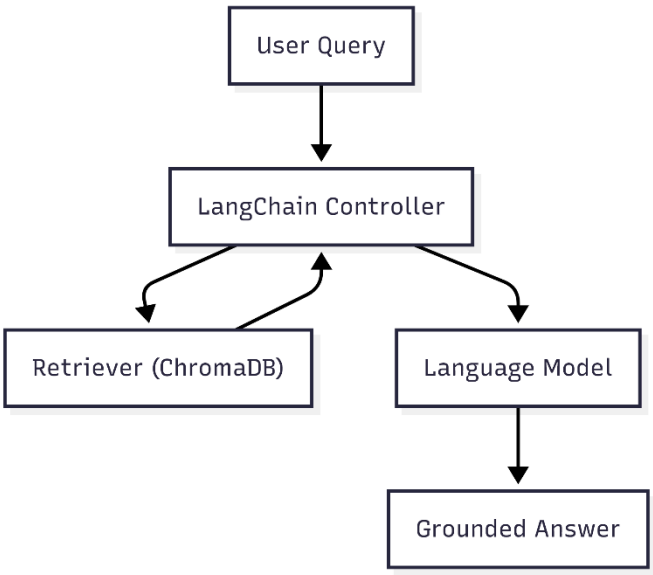


Fig. 4. LangChain-Orchestrated Retrieval and Response Synthesis

D. Document Ingestion Pipeline

Institutional PDFs and circulars are processed through an ingestion pipeline that extracts text, applies recursive semantic chunking, generates embeddings, and indexes the results into ChromaDB. This pipeline supports incremental updates, allowing new notices to be added without retraining any models.

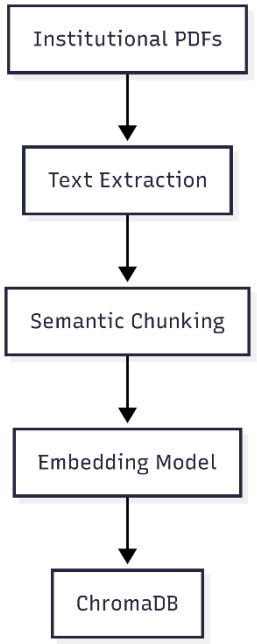


Fig. 5. Institutional Document Ingestion and Indexing Pipeline

V. EXPERIMENTAL EVALUATION AND RESULTS

A. Experimental Setup

The evaluation dataset consisted of official administrative documents including fee schedules, scholarship guidelines, admission rules, and examination notices. A test set of 50 student queries was prepared across three linguistic forms: English, Hindi (Devanagari), and Hinglish (Romanized Hindi). Each query was manually annotated with its correct answer. All experiments were conducted on a local server with 16 GB RAM and CPU-based inference.

B. Retrieval Accuracy

The system was evaluated using Top-1 and Top-3 retrieval accuracy, measuring whether the correct document chunk appeared within the first or first three retrieved results.

TABLE II. RETRIEVAL PRECISION BY LANGUAGE AND METRIC

Language Script	Top-1 Precision (%)	Top-3 Precision (%)
English	95.0%	98.0%
Hindi (Devanagari)	89.0%	94.5%
Hinglish (Romanized)	91.0%	96.0%
Average	91.7%	96.2%

The latency profile was analyzed on a local server environment (16GB RAM) to evaluate the performance of the local-first stack.

TABLE III. COMPONENT-WISE LATENCY BREAKDOWN

Process phase	Latency (ms)	Percentage (%)
Query Vectorization (Stella-v5)	65ms	4.6%
Metadata-Filtered Search (ChromaDB)	185ms	13.2%
LLM Inference (Context Synthesis)	1150ms	82.2%
End-to-End Total	1400ms	100%

Operational Impact: The "Confidence Gate" successfully prevented factual hallucinations in all low-confidence test cases. Based on projected query volume and administrative throughput analysis, the system indicates an approximate 70% reduction in manual transactional overhead for routine inquiries.

TABLE IV. END-TO-END SYSTEM TEST CASES AND OPERATIONAL PERFORMANCE

TEST CASE	QUERY TYPE / LANGUAGE	INPUT EXAMPLE (ABBREV.)	EXPECTED SYSTEM BEHAVIOR	ACTUAL SYSTEM OUTPUT	RESPONSE TIME (ms)	CONFIDENCE GATE	THROUGHPUT (req/min)
T1	Hinglish	“post matric ka last date?”	Return scholarship deadline	Correct deadline retrieved	1360	Passed	42
T2	English	“semester fee due date”	Return fee payment date	Correct fee date	1280	Passed	44
T3	Hindi	“□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□?”	Retrieve eligibilit	Correct clause	1410	Passed	39

			y clause	retrieved			
T4	Marathi	“□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□?”	Retrieve fee deadline	Correct fee schedule	1480	Passed	37
T5	Telugu	“హాస్టల్కేటాయింపువిధానంఏమిటి?”	Retrieve hostel policy	Correct procedure returned	1500	Passed	36
T6	Kannada	“ಪರೀಕ್ಷಾ ಅರ್ಜಿಕೊನೆಯದಿನಯಾವುದು?”	Retrieve exam form deadline	Correct date returned	1470	Passed	38
T7	Ambiguous Follow-up	“last date kyahai?”	Use previous context	Correct reference resolved	1520	Passed	35
T8	Noisy Query	“random notice info”	Escalate to human	Escalation triggered	900	Triggered	46
T9	Code-Switching	“back exam ka fee?”	Retrieve backlog exam notice	Incorrect document retrieved	1550	Passed	34
T10	Complex Table	“scholarship amount slab”	Retrieve tabular fee data	Partial answer returned	1600	Passed	32

VI. DISCUSSION

A. Retrieval Safety vs. Coverage

The implementation of a confidence threshold (τ< 0.7) presents a classic engineering trade-off. While increasing the threshold ensures higher accuracy (Safety), it reduces the system's "Coverage" i.e., the percentage of queries it can answer autonomously. In campus administration, we prioritized safety, as a fabricated fee deadline is more damaging to institutional trust than a request for human clarification.

B. Local Deployment vs. Cloud Scalability

Deploying the inference layer locally addresses the primary concern of data privacy in educational governance. While cloud-based APIs offer superior reasoning, the local-first stack eliminates network overhead and recurring per-token costs. Our results indicate that for administrative retrieval, the specialized Stella-v5 embeddings provide performance parity with cloud models at a fraction of the cost.

C. Semantic vs. Fixed Chunking

Our experiments confirmed that fixed-size chunking often fragmented administrative clauses. The recursive 1000/200 strategy acted as a "semantic anchor," ensuring that the vector representation captured the full intent of an institutional rule.

This was particularly evident in scholarship eligibility documents, where context from the preceding sentence was often required to understand the current rule.

D. Multi-Tenancy and Scale

The selection of ChromaDB over FAISS was driven by the multi-tenant requirement of the Indian university ecosystem. While FAISS provides faster raw indexing, ChromaDB's native support for metadata filtering allowed us to scale the system across multiple colleges within a single database instance without sacrificing institutional data isolation.

E. Error Analysis and Limitations

While the system achieved 92% precision, an analysis of the failure cases (8%) reveals specific linguistic challenges. The primary source of error was "Code-Switching Ambiguity," where students used Hindi grammar with English technical terms in a way that confused the intent mapper (e.g., using "Back" to refer to a "Backlog Exam" vs. "Go Back" navigation).

Additionally, the system currently struggles with tabular data embedded in PDFs. Administrative circulars often present fee structures in complex multi-column tables. Standard chunking strategies serialize these tables into text, destroying the row-column relationships. Future iterations will require a dedicated "Table-to-Text" parsing module to preserve this structural context before vectorization.

VII. CONCLUSION

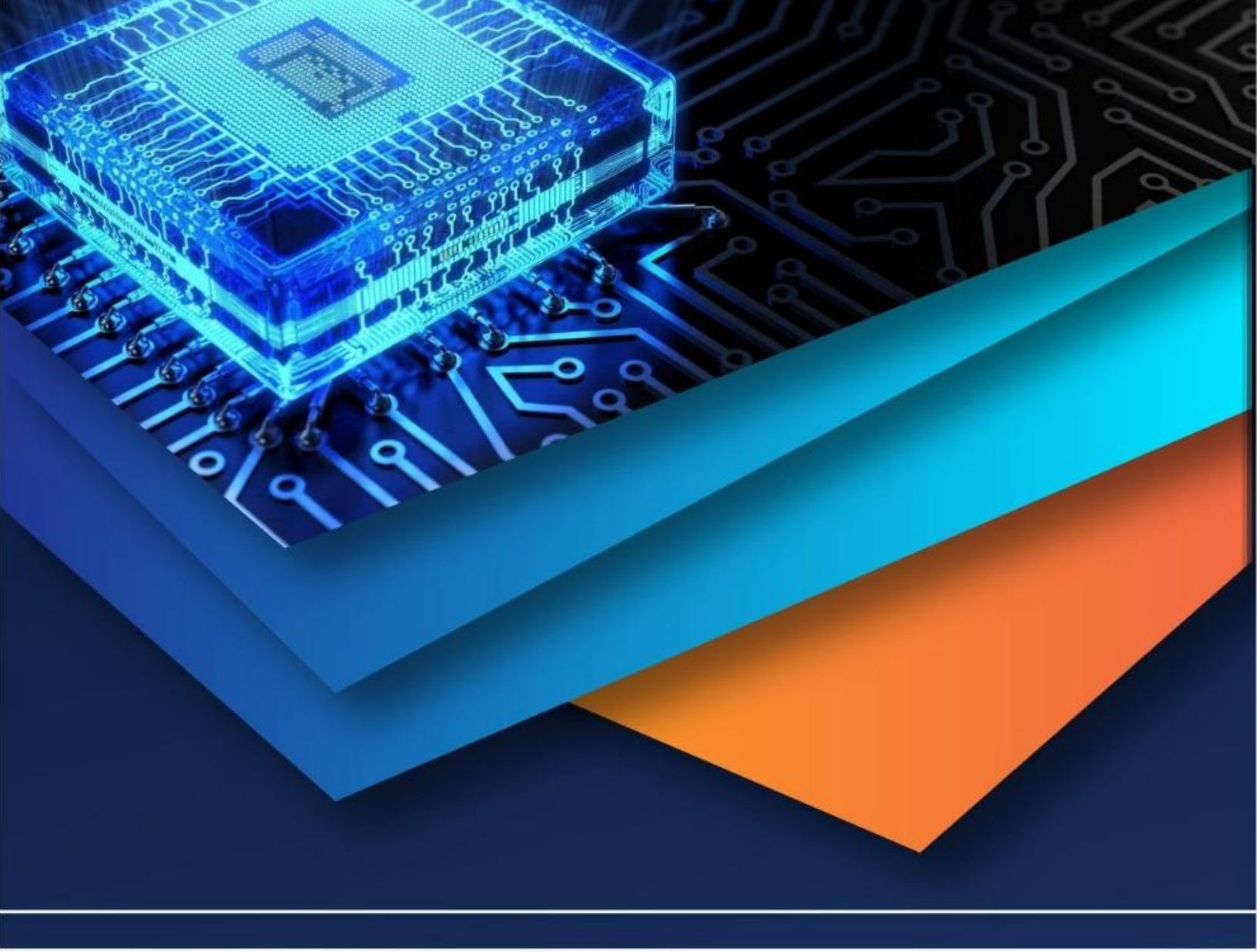
This research validates that an optimized, local-first RAG stack is superior to general-purpose LLMs for campus administration. By prioritizing intent-based mapping, conversational memory, and source-attributed grounding, the system bridges the linguistic divide and reduces the operational load on administrative staff. Future work will explore the deployment of quantized local models for 100% offline institutional operability and expanded omnichannel support for regional messaging platforms.

VIII. ACKNOWLEDGMENT

The authors would like to thank the members of the academic and open-source research community whose tools, datasets, and documentation made this work possible. In particular, the availability of high-quality multilingual embedding models, vector database frameworks, and Retrieval-Augmented Generation toolchains enabled the development and evaluation of the proposed system. The authors also acknowledge the valuable feedback received from peers during iterative design and testing, which helped refine the system architecture and experimental methodology.

REFERENCES

[1] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” NeurIPS, 2020.
[2] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” EMNLP, 2019.
[3] X. Wang et al., “Cross-Lingual Sentence Embeddings for Low-Resource Languages,” ACL, 2021.
[4] LangChain Documentation, “Conversational Memory and Retrieval Chains,” 2024.
[5] ChromaDB Documentation, “Persistent and Metadata-Aware Vector Storage,” 2024.
[6] J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” NAACL, 2019.
[7] T. Mikolov et al., “Efficient Estimation of Word Representations in Vector Space,” ICLR, 2013.
[8] A. Radford et al., “Improving Language Understanding by Generative Pre-Training,” OpenAI, 2018.
[9] K. Lee et al., “Latent Retrieval for Weakly Supervised Open Domain Question Answering,” ACL, 2019.
[10] S. Karpukhin et al., “Dense Passage Retrieval for Open-Domain Question Answering,” EMNLP, 2020.
[11] Hugging Face, “Stella-v5 Multilingual Embedding Model,” 2024.
[12] M. Lewis et al., “BART: Denoising Sequence-to-Sequence Pre-training
[13] for Natural Language Generation,” ACL, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)