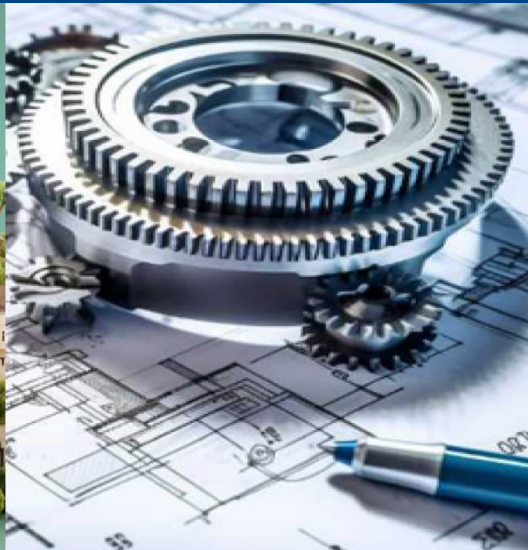
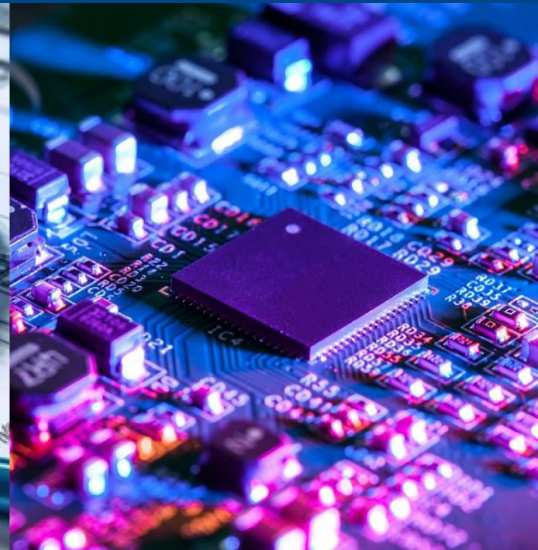
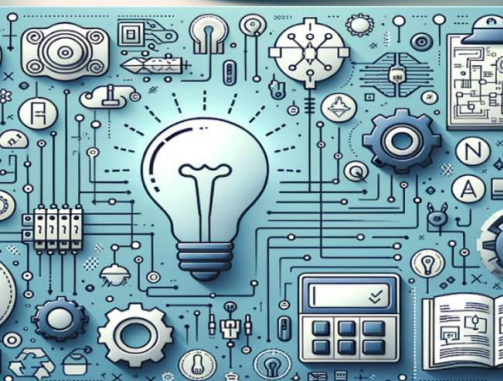




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.864

Volume 9, Issue 10, October 2026



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

College AI RAG: A Role-Aware Retrieval-Augmented Generation Assistant for College Students, Teachers and Administration

Prof. S. S. Kirte¹, Mr. Ajay Savare², Mr. Onkar Dharmale², Mr. Nikhil Bavi², Mr. Aditya Subhedar²

Project Guide, Department of Computer Science and Engineering, Brahmdevdada Mane Institute of Technology,
Solapur, Maharashtra, India¹

Department of Computer Science and Engineering, Brahmdevdada Mane Institute of Technology, Solapur,
Maharashtra, India²

ABSTRACT: College information such as notices, syllabi, question papers, placement circulars, scholarship deadlines and hostel rules is usually scattered across notice boards, messaging groups and individual staff members. General-purpose AI chatbots cannot help because they have no access to official college documents and may produce made-up answers. This paper presents College AI RAG, a role-aware Retrieval-Augmented Generation (RAG) assistant for students, teachers and administrators. Teachers and administrators upload documents once; the system extracts, chunks and embeds them, and answers natural-language questions using hybrid search (vector similarity plus full-text search) so that every answer is generated only from retrieved passages and shows its source. When no relevant passage exists, the system clearly says so instead of guessing. The same knowledge base supports lecture transcription, notes, quiz, assignment and viva generation, placement, scholarship and hostel information, grievance routing, notifications and an analytics dashboard, with voice input and answers in English, Hindi and Marathi. The system uses FastAPI, PostgreSQL with pgvector, sentence-transformers, the Anthropic Claude API, Whisper and a React frontend, and passed all 100 designed test cases.

KEYWORDS: Retrieval-Augmented Generation, Large Language Model, pgvector, Role-Based Access Control, Hybrid Search, FastAPI, React, Educational Chatbot.

I. INTRODUCTION

Large Language Models (LLMs) such as ChatGPT and Claude can understand questions written in normal language and reply like a human. However, they know nothing about a particular college: they have never seen its notices, syllabus, timetable or hostel rules, and may therefore give a wrong or invented answer to a college-specific question.

College AI RAG solves this problem. It is a web-based assistant that answers questions using only the college's own uploaded documents through Retrieval-Augmented Generation. The system first searches the documents for the most relevant passages and then asks the language model to write an answer using only those passages. Every answer shows the source document so that the user can verify it, and if the information is unavailable the system says so.

The system is role-aware: students, teachers and administrators log in with separate roles, and each role sees only the features and documents meant for it. Apart from the chatbot, the same knowledge base is reused for lecture transcription, notes, quizzes, assignments, viva preparation, placement and scholarship information, hostel information, grievance redressal, notifications and analytics.

1.1 Problem Statement

Students waste time searching for information spread over notice boards, WhatsApp groups, spreadsheets and staff members, and repeatedly ask the same questions. Teachers spend considerable time preparing quizzes, assignments and viva questions manually, and grievances are often not tracked. There is a need for a single, secure, role-based AI platform that answers college questions accurately from official documents and shows the source of every answer.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

1.2 Objectives

- Build a centralized knowledge base from college documents.
- Provide grounded answers with source citations and a clear “not found” response.
- Secure access through JWT and Role-Based Access Control (RBAC).
- Convert lecture audio to searchable text and generate notes, quizzes, assignments and viva questions.
- Provide placement, scholarship and hostel information, grievance routing and an admin analytics dashboard.
- Support voice interaction and English, Hindi and Marathi answers.

II. LITERATURE REVIEW

Lewis et al. introduced Retrieval-Augmented Generation and showed that giving a language model relevant passages retrieved from a document collection makes its answers more factual and allows knowledge to be updated by changing the documents, without retraining the model [1]. This idea is the core of our system: college documents form the collection and the model answers only from retrieved passages.

Reimers and Gurevych proposed Sentence-BERT, which produces sentence embeddings that can be compared by cosine similarity much faster than pair-wise BERT comparison [2]. We use the lightweight all-MiniLM-L6-v2 model, which gives 384-dimensional vectors and runs on a normal CPU.

pgvector adds a vector data type and similarity operators to PostgreSQL, allowing users, documents and embeddings to live in one database [6]. Combined with PostgreSQL full-text search [11], it enables hybrid search that finds results both by meaning and by exact keywords such as subject codes.

Radford et al. presented Whisper, a speech recognition model trained on 680,000 hours of multilingual audio that supports Indian English, Hindi and Marathi [3]. JSON Web Tokens [4] provide a stateless way to carry a user’s identity and role with every request, and FastAPI [5] and React [9] are widely used to build the backend and single-page interface.

Most existing college chatbots are either rule-based, with fixed question–answer pairs, or use a general model without grounding. Very few combine grounded answers, source citations, role-based document access and academic tools in one platform. Table 1 compares a general chatbot with the proposed system.

Feature	General chatbot	College AI RAG
Uses official college documents	No	Yes
Shows source of answer	No	Yes
Says "not found" instead of guessing	No	Yes
Role-based access	No	Yes
Quiz/viva from syllabus	Partly	Yes
Grievance tracking & analytics	No	Yes

Table 1. General chatbot vs. College AI RAG

III. METHODOLOGY

The RAG pipeline has two parts: an ingestion pipeline that runs when a document is uploaded, and a query pipeline that runs for every question (Fig. 1).

3.1 Ingestion Pipeline

- Extract: text is read from PDF, DOCX, PPTX, TXT, MD or CSV files (maximum 25 MB) using pypdf, python-docx and python-pptx.
- Clean: extra spaces and blank lines are removed.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Chunk:** text is split on paragraph boundaries into chunks of about 800 characters with 120 characters of overlap so that meaning is not lost at the edges.
- **Embed and store:** each chunk is converted to a 384-dimension vector with all-MiniLM-L6-v2 and stored in PostgreSQL (pgvector) with its department, subject, document type and access level.

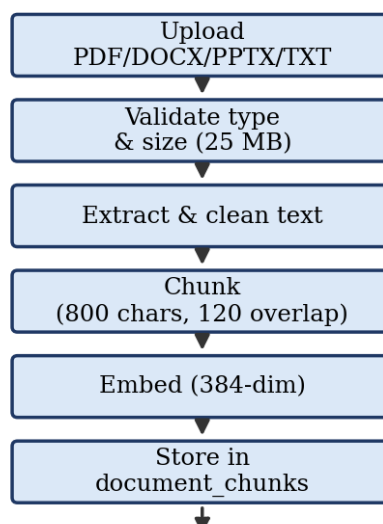
3.2 Query Pipeline

The user's question is embedded and chunks are first filtered by the access levels allowed for the user's role. Remaining chunks are ranked by a blended score:

$$\text{score} = 0.75 \times \text{vector similarity} + 0.25 \times \text{keyword rank}$$

Vector similarity (cosine) finds passages with the same meaning even when different words are used, while the PostgreSQL full-text rank helps with exact terms. The top 6 chunks and the question are sent to the Claude model with a strict system prompt that instructs it to answer only from the given context. If no chunk is retrieved, the LLM is not called at all and a fixed message, "I couldn't find this information in the college knowledge base," is returned. The answer, its sources and a grounded flag are stored in the chat history and returned to the user.

A. Ingestion pipeline (Teacher / Admin)



B. Query pipeline (any logged-in user)

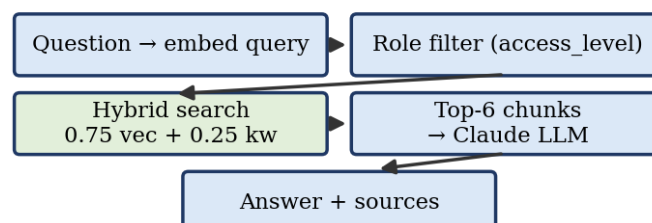


Fig. 1. Ingestion and query pipelines of the RAG engine

3.3 Role-Based Access Control

Documents carry one of four access levels: public, student, teacher or admin. Students can read public and student documents; teachers additionally read teacher documents; admins read all. Access is enforced centrally in the retriever and through a reusable `require_role()` dependency, so students cannot reach admin data. Passwords are stored as bcrypt hashes and tokens are signed JWTs (HS256, valid for 24 hours).



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. SYSTEM ARCHITECTURE

The system follows a three-layer architecture (Fig. 2). The presentation layer is a React, TypeScript, Vite and Tailwind CSS single-page application. The application layer is a FastAPI backend containing 14 REST routers under /api/v1, JWT authentication with RBAC, a service layer and the RAG engine. The data layer is PostgreSQL 15 with pgvector (15 tables, UUID primary keys) plus a folder for uploaded files. External AI services are used only for answer generation (Anthropic Claude) and speech (Whisper and text-to-speech); embeddings are computed locally.

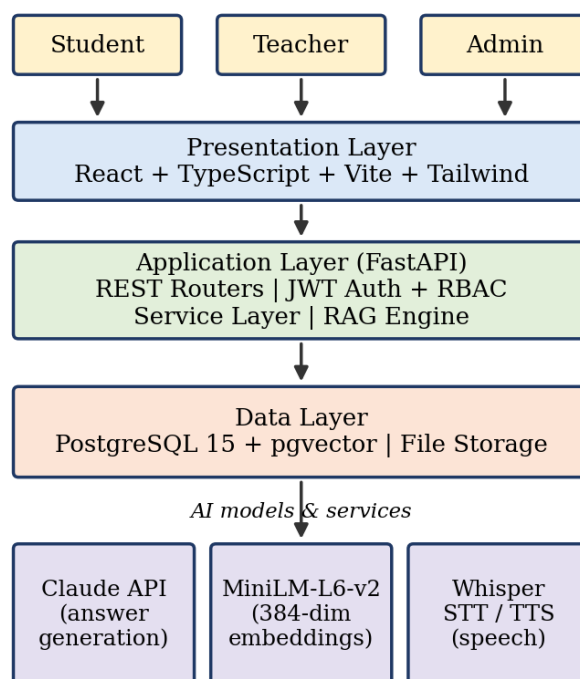


Fig. 2. Three-layer architecture of College AI RAG

The frontend and backend run in Docker containers, and the AI services are reached over HTTPS. The code is divided into routers, schemas, services, models and RAG modules so that each part can be changed independently.

V. MODULES AND FEATURES

- Document management: teachers and admins upload, list and delete documents; access level is chosen at upload time.
- RAG chat and voice assistant: grounded answers with citations; voice questions via Whisper; answers in English, Hindi or Marathi.
- Lecture assistant: audio (MP3, WAV, M4A, MP4, WEBM) is transcribed, indexed like a document, and turned into short, detailed, exam-oriented or easy notes.
- Quiz, assignment and viva generators: the LLM is given retrieved course material and returns structured JSON; quiz generation is refused when no material exists, and student attempts are scored on the server.
- Placements, internships, scholarships and hostel information: posted by staff and viewable by all, and also indexed for the assistant.
- Grievance redressal: complaints receive a reference code and are auto-routed by category (e.g., Hostel → Warden, Exam → Exam Department, Fees → Accounts Office) and tracked through Assigned, Under Review and Resolved.
- Notifications and analytics: in-app alerts with read state; an admin dashboard shows total and unanswered queries, top topics, active users and most-used documents.



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. TECHNOLOGIES USED

Backend: Python 3.11, FastAPI, SQLAlchemy 2.0, Pydantic and Uvicorn. Database: PostgreSQL 15 with the pgvector extension. AI: sentence-transformers (all-MiniLM-L6-v2) [12], Anthropic Claude [8] for generation and OpenAI Whisper for speech. Security: python-jose and bcrypt. Frontend: React 18, TypeScript, Vite, Tailwind CSS, TanStack Query [10], Axios and Recharts. Testing and tools: pytest, httpx, Postman and Docker.

VII. RESULTS AND TESTING

The system was tested at unit, integration, functional, role-based, security, negative/boundary and failure-recovery levels using pytest, the FastAPI TestClient, Swagger UI and Postman with one student, one teacher and one admin account and a set of sample college documents (syllabus, lecture notes, lab manual, notice, hostel rules and meeting minutes). In total 100 test cases were designed across twelve module groups (Table 2, Fig. 3), and every functional requirement is covered by at least one test case.

Module	Test cases	Passed
Registration & Login	10	10
Roles & Access Control	11	11
Document Upload & Mgmt	9	9
Document Processing	10	10
Search & Retrieval	8	8
RAG Question Answering	12	12
Lecture Audio	6	6
Notes, Quiz, Assignment, Viva	8	8
Campus Information	4	4
Grievance & Notifications	5	5
Voice & Multilingual	5	5
Security, Reliability & UI	12	12
Total	100	100

Table 2. Test case summary by module

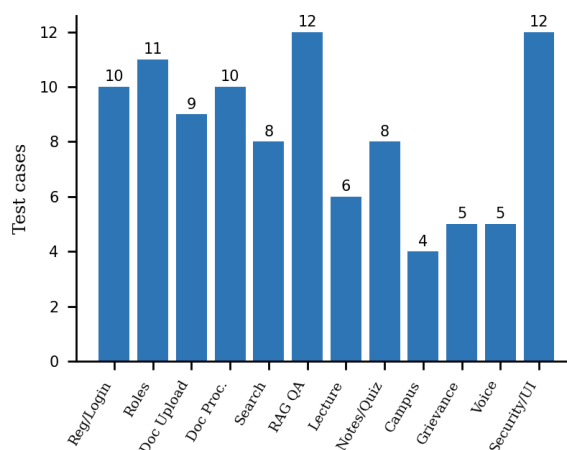


Fig. 3. Number of test cases per module



International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

All 100 test cases produced the expected output (pass rate 100%). The main observations were:

- Questions about uploaded documents were answered correctly with the source shown.
- Questions outside the knowledge base returned the fixed not-found message instead of a guess.
- A student could not open admin analytics, and each role saw only its permitted pages and documents.
- Quizzes, assignments and viva sets were generated within seconds from the uploaded material.
- Database, AI API and document-processing failures were handled without crashing the application.

7.1 Limitations

The tests used a small set of sample documents, so a very large knowledge base may behave differently. Because the language model and speech services are external, the wording of generated answers can vary slightly between runs, and load testing with many simultaneous users was not part of this work. Scanned (image-only) documents are currently not supported.

VIII. CONCLUSION AND FUTURE SCOPE

College AI RAG provides one centralized, role-aware AI platform for students, teachers and administrators. By retrieving relevant college documents before generation, it gives reliable answers with visible sources and honestly reports when information is missing. Document management, RBAC, chat and voice assistance, lecture transcription, quiz, assignment and viva generation, campus information, grievance redressal, notifications and analytics are combined in a single web application that passed all 100 test cases.

Future work includes OCR for scanned documents, a native mobile application, email, SMS and push notifications, timetable and attendance modules, deeper semester-wise analytics, more Indian languages, ERP integration and an optional self-hosted open-source LLM to reduce API cost and dependency.

REFERENCES

- [1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS, 2020. <https://arxiv.org/abs/2005.11401>
- [2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," EMNLP, 2019. <https://arxiv.org/abs/1908.10084>
- [3] A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision" (Whisper), 2022. <https://arxiv.org/abs/2212.04356>
- [4] M. Jones, J. Bradley and N. Sakimura, "RFC 7519: JSON Web Token (JWT)," IETF, 2015. <https://www.rfc-editor.org/rfc/rfc7519>
- [5] FastAPI Documentation. <https://fastapi.tiangolo.com/>
- [6] pgvector: Open-source vector similarity search for PostgreSQL. <https://github.com/pgvector/pgvector>
- [7] SQLAlchemy Documentation. <https://docs.sqlalchemy.org/>
- [8] Anthropic, Claude API Documentation. <https://docs.anthropic.com>
- [9] React Documentation. <https://react.dev/>
- [10] TanStack Query Documentation. <https://tanstack.com/query/>
- [11] PostgreSQL Full-Text Search Documentation. <https://www.postgresql.org/docs/current/textsearch.html>
- [12] Sentence-Transformers Documentation. <https://www.sbert.net/>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com