IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

Intelligent Teaching Assistant using RAG and AIML

Prof. S. Y. Mandlik, Atharva Ramesh Kharmale, Prasad Pramod Mule, Argade Ganesh Dattatray
Jaihind College of Engineering, Kuran, Maharashtra
Corresponding mail address : atharvakharmale@gmail.com

Abstract : *Engineering students accumulate large collections of lecture notes, presentations, lab manuals and previous-year question papers, yet locating an accurate answer inside them remains a slow, manual task. General-purpose Large Language Models (LLMs) answer fluently but depend on static training data, may hallucinate facts, and do not use institution-specific material. Retrieval-Augmented Gener-ation (RAG) addresses these limitations by grounding generation in an external, updatable knowledge base. This review examines three recent IEEE studies, namely a systematic literature review of AI-driven Intelligent Tutoring Systems (ITS) in engineering education, a survey of RAG for LLMs, and a practi-cal AI-RAG architecture for real-time retrieval, and combines their findings with the project synopsis and presentation to propose an integrated framework for an Intelligent Teaching Assistant. The frame-work connects role-based document ingestion, chunking and embedding, query rewriting, hybrid (BM25 plus vector) retrieval, semantic re-ranking, LLM answer generation, Self-RAG verification and citation-based answers with AI quiz generation, notes generation and learning analytics for both students and teachers. The review identifies major research gaps: ITS studies rarely use RAG, RAG studies rarely target education, question-answering tools are often not personalized, and instructor-facing analytics are scarce. It also discusses challenges involving retriever-generator alignment, knowledge conflicts, la-tency, knowledge-base quality, privacy and evaluation, and outlines an evaluation plan based on the RAG evaluation triad. The resulting workflow, Ingest, Chunk, Embed, Retrieve, Re-rank, Generate, Verify, Cite and Analyze, provides a unified perspective for building grounded, transparent and personalized AI teaching assistance..*

Keywords: Retrieval-Augmented Generation; Large Language Models; Intelligent Tutoring Systems; Teaching Assistant; Semantic Search; Vector Database; Hybrid Retrieval; Self-RAG; Educational Tech-nology; Learning Analytics.

I. INTRODUCTION

Education has become increasingly digital. A typical engineering student now works with lecture notes, PDFs, slide decks, lab manuals and years of question papers for every subject. Finding the specific expla-nation or definition needed for a doubt means opening many files and scanning them by hand. Keyword search inside individual files offers little help when the student does not know the exact terminology used by the instructor, and repeated doubts continue to consume faculty time.

Large Language Models have demonstrated strong abilities in understanding and generating human language, but their real-world use is limited by fundamental weaknesses. Their knowledge is confined to their training data, they cannot access updated or institution-specific information, and they are prone to hallucination, that is, generating factually incorrect content [2, 3]. These weaknesses matter most where accuracy and source transparency are essential, and education is one such domain. A student who receives a confident but wrong explanation, without any reference to check it against, may be worse off than one who received no answer.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

Retrieval-Augmented Generation was formally proposed by Lewis et al. [4] as a hybrid architecture that combines the parametric knowledge stored in an LLM with a non-parametric external knowledge source. Before generating an answer, the system retrieves passages relevant to the query and supplies them to the model as context. This makes answers verifiable, allows the knowledge base to be updated without retraining the model, and reduces hallucination [2, 3]. In parallel, research on AI-driven In-telligent Tutoring Systems has shown how personalization, real-time feedback and learner monitoring can improve learning in technically demanding subjects [1]. The two lines of research, however, have developed largely apart.

This review brings them together. It studies three recent IEEE publications, one on ITS in engineer-ing education [1], one on the RAG paradigm [2] and one on a real-time RAG system [3], and uses the project synopsis and presentation of the Intelligent Teaching Assistant using RAG and AIML to position the proposed system against them. The paper is organized as follows. Section 2 reviews the literature. Section 3 analyzes research gaps. Section 4 presents the proposed architecture and methodology. Sec-tion 5 compares the approach with existing alternatives. Sections 6 and 7 discuss the integrated approach and its challenges. Sections 8 and 9 give future scope and conclusions.

1.1 Problem Statement

Students face difficulty finding accurate, relevant information within large volumes of study material such as PDFs, notes, PPTs and question papers, resulting in time-consuming manual searches. Generic AI chatbots do not use institution-specific resources and often return inaccurate or unsupported answers, while existing ITS and RAG research addresses only parts of the problem. Therefore, an integrated framework is required that retrieves relevant content from uploaded educational documents, generates accurate and context-aware answers with source references, verifies those answers, and supports both students and teachers.

1.2 Need of the Project

The need for the proposed system arises from the following observations made in the project synopsis:

- Easy access to educational resources, and less time spent searching through notes.
- Better concept understanding through AI-based, personalized learning and support for self-learning.
- Reliable answers that carry source references, so that students can verify what they read.
- Reduced dependency on faculty for repeated doubts, which increases productivity on both sides.
- Digital transformation of education through a modern, AI-powered learning platform.

1.3 Objectives

- To design a Retrieval-Augmented Generation based system that retrieves relevant content from institution-specific study materials such as PDFs, PPTs and notes.
- To integrate an LLM with the retrieval pipeline to generate accurate, context-aware answers along with source references.
- To reduce the time students spend searching for information and minimize dependency on faculty for repetitive doubts.
- To provide additional AI-based features such as automatic summary, notes and quiz generation from uploaded content.
- To build a role-based web application for Admin, Teacher and Student with secure access and easy content management.

1.4 Expected Outcomes

- Automated searching of study material, so that students reach accurate academic information quickly.
- Improved quality of education through personalized, AI-assisted learning based on institution-specific documents.
- Less manual searching for students and less repetitive doubt-solving for faculty.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

• More reliable learning: answers with source references reduce misinformation and encourage self-directed study.
• A step towards a smarter, more accessible, technology-driven learning environment.

II. LITERATURE REVIEW

2.1 Limitations of Standalone LLMs in Education

LLMs such as GPT and BERT-based models learn from large datasets collected at a fixed point in time. After training, their knowledge is static and they cannot retrieve real-time or revised information [3]. Consequently they may produce outdated answers or fabricate details when a question is unusual or ambiguous. They also do not cite the sources of their statements, which makes their claims hard to validate [3]. In an educational setting, these limitations are compounded by the fact that a general model has never seen the specific notes, syllabus or question papers used in a given course.

2.2 Foundations of Retrieval-Augmented Generation

The core of a RAG system is its hybrid knowledge structure, which fuses parametric knowledge, im-plicitly encoded in the LLM weights, with non-parametric knowledge held in an external, updatable knowledge base [2]. A RAG system consists of two components. The Retriever receives the user query and fetches relevant passages from the knowledge base, and the Generator receives the query together with the retrieved passages and produces the final answer [2]. The retrieved passages are combined with the original query to form an augmented prompt, so that generation is anchored to specific, verifiable information rather than to model memory alone. The survey by Ma et al. [2] organizes the evolution of the field into three stages: Naive RAG, a straightforward retrieve-then-generate pipeline; Advanced RAG, which adds optimization modules before and after retrieval; and Modular RAG, which treats the system as a set of plug-and-play components.

2.3 Knowledge Base Preparation: Chunking and Indexing

Building a knowledge base begins by cleaning raw documents and converting them into plain text. The text is then segmented into chunks. The chunking strategy is crucial: chunks that are too small may lead to incomplete information, while chunks that are too large introduce noise and worsen the “lost in the middle” problem, in which an LLM struggles to use key information buried in long inputs [2]. Each chunk is converted into a vector embedding by an encoder and stored in a vector database that supports Approximate Nearest Neighbor search, allowing rapid retrieval of the most similar chunks at query time [2]. Kambattan and Chowdary [3] follow a similar path, cleaning and tokenizing collected data, generating transformer-based embeddings and indexing the knowledge base for fast retrieval.

2.4 Retrieval Methods: Sparse, Dense and Hybrid

Retrieval methods are broadly categorized as sparse or dense [2]. Sparse retrieval, represented by TF-IDF and BM25, relies on keyword matching. It is computationally efficient but does not capture semantic similarity. Dense retrieval maps text into a low-dimensional vector space using deep neural networks, enabling semantic matching and addressing the vocabulary mismatch problem. Dense Passage Retrieval (DPR) by Karpukhin et al. [5] showed that pure dense retrieval can significantly outperform BM25 on open-domain question answering benchmarks. The mainstream architecture is the dual-encoder, with separate query and passage encoders whose dot product gives the relevance score, so passage vectors can be pre-computed offline. Such encoders are trained by contrastive learning, and DPR’s in-batch negatives strategy has proved effective [2, 5]. Hybrid search combines keyword precision with semantic understanding for more robust results [2]. The AI-RAG system [3] likewise supports dense, sparse and hybrid retrieval and ranks candidates with BM25 or transformer-based similarity scores.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

2.5 Generation, Augmentation and Knowledge Conflicts

Augmentation is the process of combining retrieved documents with the query. The most common method is concatenation, in which chunks are placed before or after the query, although fusion can also occur in intermediate model layers or at the output probability level [2]. The generator must produce an answer that is fluent and also factually consistent with the supplied context, and the model can be explicitly instructed to adhere strictly to the provided documents to reduce hallucination [2]. A known challenge is knowledge conflict, where the model's internal parametric knowledge contradicts the re-trieved context. Its causes include outdated or incorrect external knowledge and the model's own bias towards memorized beliefs, and unresolved conflict leads to unfaithful outputs [2]. For a teaching assis-tant, this reinforces the need for a verification step after generation.

2.6 Advanced RAG: Query Transformation, Re-ranking and Iteration

Advanced RAG improves the pipeline before and after retrieval [2]. On the query side, query trans-formation reformulates the user question into a retriever-friendly form. Query rewriting clarifies intent, as in the Rewrite-Retrieve-Read framework [11]. Query expansion adds related terms or hypothetical answers (for example query2doc and HyDE) to bridge vocabulary gaps. Query decomposition splits complex multi-hop questions into simpler sub-questions. After retrieval, a re-ranking module reorders documents so the most relevant appear first, using a lightweight cross-encoder or the LLM itself, and context compression shortens the context while preserving key information [2]. Advanced RAG also introduces iterative and adaptive retrieval, in which the model identifies knowledge gaps during gener-ation and issues new queries [2]. Wang et al. [10] provide an empirical search for best practices across these components.

2.7 Modular RAG and Self-RAG

As components multiply, RAG architectures are becoming modular, letting developers mix retrievers, re-rankers and generators according to task requirements [2]. Self-RAG, proposed by Asai et al. [6], trains a single LLM to generate special reflection tokens that control the process: the model decides when to retrieve, assesses whether the retrieved documents are relevant, and critiques whether its own output is supported by the evidence. The survey reports that this on-demand retrieval and self-critique improves factuality and that smaller Self-RAG models can outperform larger models and conventional RAG methods [2]. This is the mechanism the proposed system adopts as its verification stage, checking whether an answer is actually supported by the retrieved documents before showing it to the student.

2.8 Evaluation of RAG Systems

RAG evaluation is more complex than standard LLM evaluation because both the intermediate retrieval and the final generation must be assessed. Ma et al. [2] describe an evaluation triad: context relevance (are the retrieved documents related to the query), answer faithfulness (is the answer grounded in the context and free of hallucination) and answer relevance (does the answer address the user's intent). Frameworks include RAGAS [7], a reference-free approach in which an LLM scores core dimensions without ground-truth answers; ARES [8], which generates synthetic data and fine-tunes a lightweight judge classifier; and the RGB benchmark [9], which stress-tests noise robustness, negative rejection, information integration and counterfactual robustness. Kambattan and Chowdary [3] instead report classification-style metrics (precision, recall, accuracy and F1-score) together with BLEU, ROUGE and human evaluation.

2.9 AI-Driven Intelligent Tutoring Systems in Engineering Education

Rodrigues et al. [1] conducted a PRISMA-guided systematic literature review of 46 peer-reviewed stud-ies (2010 to 2024) on AI-powered ITS in engineering education. They group techniques into five macro categories: Decision-Making and Personalization Strategies (20 studies), Learning Algorithms and Adaptation (19), Pedagogical Foundations for AI Integration (12), Language and Interaction Tech-nologies (9) and Learner Modeling and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, November 2026 Impact Factor: 8.2

Knowledge Representation (7). Among functional features, Adaptive Content Selection (38 occurrences), Real-Time Feedback (34), Student Knowledge Represen-tation (29) and Personalized Learning Paths (28) dominate, whereas Question-and-Answer Systems and Step-by-Step Guidance each appear 12 times, and Affective Recognition and Teachers' Access to Stu-dents' Progress only 3 times each. The authors conclude that current systems excel at reactive individual personalization but underdeliver on dual-user support, that is, joint support for learners and instructors.

The review also notes the rise of NLP and LLM-based tutors. Šarc˘evic´ et al. [13] built an open-source generative AI chatbot based on RAG for an introductory programming course, and Allen et al. [12] proposed Q-Module-Bot, a GPT-based question-answering system for module content. The authors observe, however, that such systems raise challenges of factual accuracy, bias and curriculum alignment, and they recommend constraining LLM responses with concept-bounded content and reflection prompts. Many question-answering systems that offer real-time feedback, such as ParichartBOT [15] and Q-Module-Bot [12], return generic responses regardless of the learner's background, and only one reviewed system, STUART [14], adapts its responses to learner profiles and combines this with instructor-facing recommendations. Teacher-oriented features such as content personalization by teachers and access to student progress remain under-implemented [1].

2.10 RAG in Practice: The AI-RAG System

Kambattan and Chowdary [3] present a practical AI-RAG architecture that augments language mod-els with real-time information retrieval. Their working procedure has six stages: data ingestion and preprocessing, knowledge retrieval and integration, language model generation, validation and evalu-ation, explainability and transparency, and deployment and integration. The architecture is organized into an Ingestion Layer, Retrieval Layer, Generation Layer and Deployment Layer. Usability features include dashboards showing the top-k retrieved documents with confidence scores, human-in-the-loop interaction, explainability modules with sentence-level attribution, and security and access controls. The authors report a 45% reduction in factual hallucination, a 60% increase in answer contextuality, more than 70% improvement in user satisfaction, and precision, recall, accuracy and F1-score of 92%, 89%, 94% and 90.5% respectively. These figures are self-reported, the datasets are not described in detail, and the system is general-purpose rather than educational, so they should be read as indicative rather than as benchmarks for a teaching assistant.

2.11 Summary of Reviewed Studies

Table 1. Summary of the reviewed studies and their relevance to the proposed system

Study	Method	Key contribution	Limitation with respect to this project
Rodrigues et al., IEEE Access, 2025 [1]	ITS, AI, NLP, adaptive learning	Systematic review of 46 studies on personalization, real-time feedback, learner monitoring and teacher analytics in engineering education.	Reviews ITS solutions but does not integrate RAG, vector databases or citation-based question answering.
Ma et al., ICTBAI, 2025 [2]	RAG, LLMs, hybrid retrieval	Explains the full RAG pipeline: chunking, embeddings, vector databases, retrieval, re-ranking, Advanced and Modular RAG, evaluation.	Survey only; does not implement a domain-specific educational teaching assistant.
Kambattan and Chowdary, ICCDS, 2025 [3]	AI-RAG, dense and sparse retrieval, LLM	Practical RAG architecture with real-time retrieval, explainable and context-aware responses, reduced hallucination.	General-purpose applications; not focused on personalized education or student analytics.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, November 2026 Impact Factor: 8.2

Proposed system (project presentation)	RAG, AIML, LLM, ITS concepts	Retrieves from educational documents, gives citation-based answers, verifies with Self-RAG, generates quizzes and notes, provides teacher and student analytics.	Needs a well-maintained knowledge base; performance may fall with very large collections or limited computing resources.

III. RESEARCH GAP ANALYSIS

The three studies address different parts of the problem. ITS research supplies pedagogical structure, RAG research supplies the technical means of grounding, and the AI-RAG system shows how these pieces can be assembled in practice. Few works connect them. The following matrix contrasts existing approaches with the proposed direction across the stages of an educational question-answering workflow.

Table 2. Gap analysis matrix: existing approaches versus the proposed integrated approach

Stage	Existing approach	Limitation	Proposed approach
Knowledge access	Manual browsing of files and keyword search in notes	Time-consuming; misses semantically related content	Semantic vector search over all uploaded course material
Answer generation	Generic LLM chatbots [2, 3]	Static knowledge, hallucination, no institution-specific content	RAG grounded in uploaded institutional documents
Query handling	Raw query passed directly to retriever	Ambiguous or vocabulary-mismatched queries retrieve poorly [2]	Query rewriting and expansion before retrieval
Retrieval quality	Single-mode retrieval, no re-ranking	Sparse misses semantics; dense may miss exact terms; noisy context [2]	Hybrid BM25 plus vector search with semantic re-ranking
Verification and traceability	Answers without sources or checks	Claims cannot be validated; knowledge conflicts unresolved [2, 3]	Self-RAG verification and citations to document, page and chapter
Personalization	Q&A bots give generic answers [1]	Only one reviewed system adapts Q&A to learner profiles [1]	Learning history, quiz scores and weak-topic tracking; adaptive paths as future scope
Assessment and revision	Separate quiz and note tools	Not linked to the institution's own material	AI quiz and notes generation from retrieved content
Instructor support	Student-facing chatbots only [1]	Teacher analytics rare; dual-user support weak [1]	Teacher dashboard: searched topics, FAQs, weak chapters, class trends
Content management	Static knowledge or manual re-indexing	Outdated answers as syllabus changes	Role-based upload, update and delete with automatic knowledge-base refresh

Five major gaps emerge from this analysis.

• Gap 1: Weak grounding in institutional content. Generic chatbots and search tools do not draw on the material actually taught in the course, so answers may be inaccurate or irrelevant to the syllabus.

• Gap 2: Separation of ITS and RAG research. ITS reviews [1] do not integrate RAG, vector databases or citation-based answering, while RAG surveys [2, 3] are largely generic and not educa-tional.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

• Gap 3: Personalization-poor question answering. Q&A modules often respond identically to every learner and rarely use learner state [1].
• Gap 4: Limited dual-user support. Instructor-facing analytics and content-management tools are under-represented compared with student-facing features [1].
• Gap 5: Insufficient verification and explainability. Retrieved and generated content is rarely checked for support, and users seldom see sources, which limits trust [1–3].

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed architecture follows the RAG paradigm [2,4] and extends it with quality controls and edu-cational services. As designed in the project presentation, it consists of fifteen numbered blocks, together with user authentication, a knowledge base, and teacher and student dashboards (Fig. 1). Blocks 1 to 4 form an offline pipeline that prepares the knowledge base, blocks 6 to 12 form the online pipeline that answers a student's question, and blocks 13 to 15 produce enhanced outputs that feed the dashboards.

4.1 Module 1: User Authentication and Role Management

Access is controlled through the User Authentication block: teachers and students log in and are authen-ticated with JSON Web Tokens (JWT). Three roles are supported, as summarized in Table 3. Role-based access protects the knowledge base and student data and separates content management from content consumption.

Table 3. User roles and capabilities

Role	Capabilities
Admin	Manages users, subjects and system settings.
Teacher	Uploads, updates and deletes study material; views the teacher dashboard with most-searched topics, frequently asked questions, weak chapters, student performance and class learning trends.
Student	Asks questions, receives cited answers, views notes, attempts quizzes and tracks progress on the student dashboard (progress, quiz performance, weak topics, recommended topics, study streak and history).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

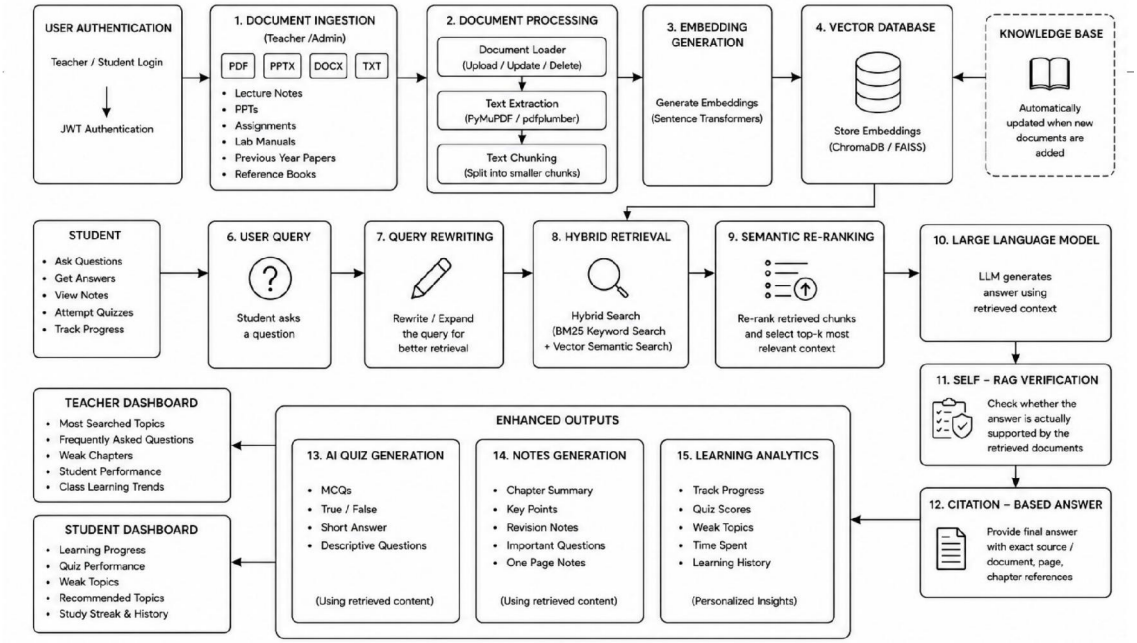


Figure 1. System architecture of the Intelligent Teaching Assistant using RAG and AIML (adapted from the project presentation).

4.2 Module 2: Document Ingestion and Knowledge-Base Management (Block 1)

Teachers and administrators upload lecture notes, PPTs, assignments, lab manuals, previous-year ques-tion papers and reference books in PDF, PPTX, DOCX and TXT formats. Documents can be uploaded, updated or deleted, and the knowledge base (shown as a dashed block in Fig. 1) is updated automatically when new documents are added. This keeps answers aligned with the current syllabus, addressing the static-knowledge limitation described in Section 2.1.

4.3 Module 3: Document Processing and Chunking (Block 2)

A document loader passes files to a text-extraction stage (PyMuPDF or pdfplumber), which converts them into clean text. The text is then split into smaller chunks and tagged by section and topic. Chunk size is a design parameter to be tuned experimentally, because too-small chunks lose context while too-large chunks add noise and aggravate the lost-in-the-middle effect [2].

4.4 Module 4: Embedding Generation and Vector Database (Blocks 3 and 4)

Each chunk is converted into a dense vector using Sentence Transformers and stored in a vector database (ChromaDB or FAISS) together with metadata such as source document, page and chapter. The metadata is what later allows the system to return exact source references. Storing vectors offline enables low-latency approximate nearest-neighbor search at query time [2, 5].

4.5 Module 5: User Query and Query Rewriting (Blocks 6 and 7)

In block 6 the student asks a question in natural language. Such a question is often short, informal or ambiguous. The query rewriting module rewrites or expands it to improve retrieval, following the query transformation family of techniques (rewriting, expansion and decomposition) described for Advanced RAG [2, 11]. For multi-part questions, decomposition into sub-questions may also be used.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

4.6 Module 6: Hybrid Retrieval and Semantic Re-ranking (Blocks 8 and 9)

The rewritten query is run through hybrid retrieval, combining BM25 keyword search with vector se-mantic search. Keyword matching preserves exact technical terms, formulas and abbreviations, while semantic search captures paraphrases and related concepts [2]. The candidate chunks are then semanti-cally re-ranked and the top-k most relevant chunks are selected as context. This reduces noise passed to the LLM and limits computational cost.

4.7 Module 7: LLM Generation, Self-RAG Verification and Citations (Blocks 10 to 12)

The LLM (an OpenAI GPT model or an open-source LLM) receives the question and the retrieved chunks and is instructed to answer only from the provided context. The draft answer then passes through Self-RAG verification, which checks whether each claim is supported by the retrieved documents [6]. Unsupported statements can be removed, regenerated or flagged. Finally, the system returns a citation-based answer that includes the exact source document, page and chapter, so the student can verify the response and read the original material. This module directly targets the hallucination and traceability gaps identified in Section 3.

4.8 Module 8: Enhanced Outputs and Dashboards (Blocks 13 to 15)

Beyond question answering, the retrieved content is reused for three services. AI quiz generation cre-ates multiple-choice, true/false, short-answer and descriptive questions. Notes generation produces chapter summaries, key points, revision notes, important questions and one-page notes. Learning ana-lytics tracks progress, quiz scores, weak topics, time spent and learning history. These feed the student and teacher dashboards described in Table 3, giving the system the dual-user character that the ITS review identifies as under-represented [1].

4.9 Algorithm

The core algorithms of the system are Retrieval-Augmented Generation and semantic vector search using embeddings, supported by BM25 keyword search in the hybrid retriever. Their end-to-end operation is summarized in Table 4.

Table 4. Algorithm 1: End-to-end RAG procedure of the proposed system

Step	Operation	Block in Fig. 1
1	Authenticate the user with JWT and identify the role (Admin, Teacher or Student).	User Authentication
2	Teacher or Admin uploads PDF, PPTX, DOCX or TXT study material.	1
3	Extract text with PyMuPDF or pdfplumber and split it into smaller chunks.	2
4	Encode chunks with Sentence Transformers and store vectors with source metadata in ChromaDB or FAISS; refresh the knowledge base when documents change.	3, 4, Knowledge Base
5	Student submits a question.	6
6	Rewrite or expand the question for better retrieval.	7
7	Retrieve candidate chunks using BM25 keyword search plus vector semantic search.	8
8	Re-rank candidates and select the top-k most relevant chunks as context.	9
9	The LLM generates an answer from the question and the retrieved context.	10
10	Self-RAG verification checks whether the answer is supported by the retrieved documents; unsupported content is revised or flagged.	11
11	Return the final answer with document, page and chapter references.	12
12	On request, generate quizzes and notes from retrieved content, and record activity for analytics.	13, 14, 15
13	Update the student and teacher dashboards.	Dashboards

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

4.10 Technology Stack

Table 5. Proposed technologies and frameworks

Layer	Technology	Purpose
Core algorithms	RAG; semantic vector search (embeddings)	Retrieve relevant content and ground the generated answer
Programming language	Python	Implementation of ingestion, retrieval and generation pipeline
Text extraction	PyMuPDF / pdfplumber	Extract text from PDF and other documents
Embeddings	Sentence Transformers	Convert chunks and queries into dense vectors
Vector database	FAISS / ChromaDB	Store embeddings and support semantic search
Retrieval	BM25 plus vector search	Hybrid retrieval with re-ranking
RAG framework	LangChain	Orchestrate retrieval and generation
LLM	OpenAI GPT / open-source LLM	Answer, quiz and notes generation
Backend	Flask / Django	Web application and APIs
Authentication	JWT	Secure, role-based access
Development environment	VS Code	Development and testing

4.11 Proposed Evaluation Plan

The system is at the design stage and the paper does not claim experimental results. The literature suggests a two-level evaluation plan. At the RAG level, the evaluation triad of context relevance, answer faithfulness and answer relevance [2] can be measured with RAGAS [7] or ARES [8], and robustness to noisy or contradictory documents can be probed with RGB-style tests [9]. Retrieval quality can additionally be reported using precision and recall, and response time can be recorded for latency. At the educational level, the plan includes time taken to find an answer compared with manual search, quiz score improvement, user satisfaction surveys for students and teachers, and a reduction in repeated doubts referred to faculty. Because the ITS review criticizes short, narrow evaluations [1], a course-embedded, multi-week pilot with real study material is preferable to a one-off demonstration.

The complete workflow of the framework is: Ingest → Chunk → Embed → Retrieve → Re-rank → Generate → Verify → Cite → Analyze.

V. COMPARATIVE ANALYSIS

Table 6 compares the proposed system with generic LLM chatbots, conventional ITS and basic (Naive) RAG. The comparison is qualitative and based on the capabilities described in the reviewed literature [1–3].

Table 6. Comparison of the proposed system with alternative approaches

Capability	Generic LLM chatbot	Conventional ITS	Basic RAG	Proposed system
Primary objective	General conversation and answers	Personalized instruction and feedback	Grounded question answering	Grounded answers, revision aids and analytics
Knowledge source	Training data (static)	Authored course model	External knowledge base	Institution-specific uploaded documents
Hallucination control	Limited	Rule-based, limited scope	Partial	Grounding plus Self-RAG verification
Source citation	Usually none	Not typical	Optional	Document, page and chapter references

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40944

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

Query optimization	No	Not applicable	Rare	Query rewriting and expansion
Retrieval	None	None	Single mode	Hybrid (BM25 plus vector) with re-ranking
Quiz and notes generation	On request, ungrounded	Some systems	Rare	Generated from retrieved course content
Teacher analytics	No	Rare [1]	No	Teacher dashboard
Student progress tracking	No	Common [1]	No	Yes, through learning analytics
Content update	Requires retraining	Manual authoring	Re-index	Automatic knowledge-base refresh

VI. DISCUSSION OF THE INTEGRATED APPROACH

The main distinction of the proposed system is not a single new algorithm. Its contribution lies in connecting capabilities that are usually implemented separately: retrieval and grounding from RAG research, learner and teacher support from ITS research, and verification from Self-RAG. Each element has support in the reviewed literature, and their combination responds directly to the gaps in Section 3.

The design also follows the recommendations of the ITS review. That review advises “LLM safety-by-design”, meaning that responses from language technologies should be constrained by bounded, curriculum-aligned content to reduce hallucination [1]. Restricting generation to retrieved institutional documents and verifying it against them is an implementation of this idea. The review further recom-mends dual-user analytics that turn learner signals into instructor actions [1], which the teacher dash-board (weak chapters, frequently asked questions, class trends) begins to address. Finally, exposing sources and, where possible, reasons for retrieval supports the explainability and trust that both the RAG survey [2] and the AI-RAG study [3] identify as important.

The architecture is compatible with graduated deployment. In a first phase it can serve only as a cited question-answering tool over one subject. Once retrieval and faithfulness are measured to be satisfactory, quiz and notes generation and dashboards can be enabled, and adaptive personalization can follow. This staged approach matches the ITS review’s caution that pedagogical integration in AI tutors is still developing and that theory should inform, not merely inspire, system design [1]. Teachers remain in control of the knowledge base, which keeps a human responsible for the material the system draws on.

VII. CHALLENGES, LIMITATIONS AND ETHICAL CONSIDERATIONS

7.1 Knowledge-Base Quality and Maintenance

The quality of answers is bounded by the quality of the knowledge base. Scanned or badly formatted PDFs, slides with images and equations, duplicated material and outdated notes can all degrade retrieval. Performance may also decrease with very large collections or limited computing resources. Regular curation by teachers and admins is required.

7.2 Chunking and Retrieval Errors

Poor chunk boundaries can split a definition from its example, and retrieval quality directly determines the upper bound of the generation stage [2]. Chunk size, overlap, the number of chunks retrieved and the balance between keyword and semantic scores must be tuned on real course material.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

7.3 Retriever-Generator Alignment and Knowledge Conflicts

The retriever is trained for relevance, while the generator may prefer documents that are clearly written and easy to synthesize, a preference gap that remains a central RAG challenge [2]. In addition, the LLM's parametric knowledge may conflict with the retrieved context, producing answers that ignore the documents [2]. Verification mitigates but does not eliminate these issues.

7.4 Hallucination, Bias and Over-Reliance

Even with grounding, LLMs can misinterpret context. The ITS review highlights factual accuracy, bias and curriculum alignment as continuing challenges for LLM-based tutors [1]. There is also a pedagogical risk that students rely on the assistant instead of engaging with the material. Citations, reflection-style prompts and quizzes can encourage students to consult the sources and test their own understanding.

7.5 Latency, Scalability and Cost

Query rewriting, hybrid retrieval, re-ranking, generation and verification each add computation. RAG complexity increases overhead and inference latency, limiting real-time use [2]. Efficient indexing, caching, model choice (hosted API versus local open-source model) and limiting the number of re-ranked chunks are practical levers.

7.6 Privacy, Security and Access Control

Learning analytics depends on storing student interactions, which raises privacy concerns. The ITS review flags privacy and surveillance concerns for personalization and analytics [1]. Role-based ac-cess, JWT authentication, minimal data retention, transparent use of analytics and controls over which documents each role may retrieve are needed. Copyright of uploaded material should also be respected.

7.7 Evaluation and Academic Integrity

Measuring educational benefit is harder than measuring answer accuracy. Short pilots and proxy tasks may not reflect real learning [1]. Automatic quiz generation can also produce poor or ambiguous ques-tions and should be reviewed by teachers before being used for graded assessment. Teachers should decide how the assistant may be used in assignments and examinations.

VIII. FUTURE SCOPE

Several directions extend the framework. The synopsis identifies four of them: support for multime-dia content such as video lectures and audio transcripts, multilingual question answering for regional-language learners, cloud deployment for scalable institutional access, and personalized learning paths with adaptive quiz difficulty based on individual performance.

The literature suggests further options. Multimodal RAG can retrieve diagrams, tables and images us-ing multimodal encoders [2], which matters for engineering subjects that rely on circuit diagrams and figures. GraphRAG, which uses knowledge graphs as the external source, supports multi-hop reasoning across related concepts [2] and connects to the concept-graph constraints recommended for educational Q&A [1]. Agentic RAG, in which an LLM plans, retrieves and reflects autonomously, is a promis-ing direction [2]. Privacy-preserving approaches such as federated learning could allow institutions to improve models without sharing raw student data [1]. Linking affective and ability signals to teacher dashboards would address another gap noted in the ITS review [1]. Finally, larger controlled studies using RAG-specific and learning-outcome measures are needed to validate the approach.

IX. CONCLUSION

Students need fast, accurate and trustworthy access to course-specific knowledge, and standalone LLMs and manual search each fall short. This review examined an ITS review in engineering education [1], a survey of RAG [2] and a

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, November 2026 Impact Factor: 8.2

practical AI-RAG architecture [3], and found that they are complementary: ITS research offers pedagogical goals and dual-user design principles, while RAG research offers the technical means to ground language models in verifiable knowledge. Existing work seldom brings the two together.

The proposed Intelligent Teaching Assistant using RAG and AIML connects them through document in-gestion, chunking, embedding, query rewriting, hybrid retrieval, semantic re-ranking, LLM generation, Self-RAG verification and citation-based answers, together with quiz generation, notes generation and analytics for students and teachers. The resulting lifecycle, Ingest → Chunk → Embed → Retrieve → Re-rank → Generate → Verify → Cite → Analyze, moves educational assistance from generic chatbot responses towards grounded and transparent support. The approach does not replace teachers; it sup-ports them by handling repetitive doubts and by surfacing where students struggle. Its effectiveness will depend on knowledge-base quality, retrieval and faithfulness performance, latency, privacy protection and teacher oversight, and these should be verified through the evaluation plan proposed in this paper.

REFERENCES

[1] B. Rodrigues, R. Pinto, and G. Gonçalves, "A systematic literature review of AI-driven intelligent tutoring systems in engineering education: Emphasizing personalization, feedback, and student monitoring," IEEE Access, vol. 13, pp. 190152–190177, 2025, doi: 10.1109/ACCESS.2025.3626473.

[2] Y. Ma, H. Nie, C. Chen, J. Zhang, J. Jiang, B. Wang, and Y. Xia, "A survey of retrieval-augmented generation (RAG) for large language models," in Proc. 2025 Int. Conf. Trustworthy Big Data Artif. Intell. (ICTBAI), 2025, pp. 7–13, doi: 10.1109/ICTBAI68361.2025.00008.

[3] K. R. Kambattan and P. S. Chowdary, "AI-RAG: Enhancing language models with real-time in-formation retrieval," in Proc. 2nd Int. Conf. Comput. Data Sci. (ICCDS), 2025, doi: 10.1109/IC-CDS64403.2025.11209598.

[4] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Adv. Neural Inf. Process. Syst., vol. 33, pp. 9459–9474, 2020.

[5] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 6769–6781.

[6] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in Proc. 12th Int. Conf. Learn. Represent. (ICLR), 2024.

[7] S. Es, J. James, L. E. Anke, and S. Schockaert, "RAGAS: Automated evaluation of retrieval augmented generation," in Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics: Syst. Demonstrations, 2024, pp. 150–158.

[8] J. Saad-Falcon, O. Khattab, C. Potts, and M. Zaharia, "ARES: An automated evaluation framework for retrieval-augmented generation systems," in Proc. NAACL, 2024, pp. 338–354.

[9] J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking large language models in retrieval-augmented genera-tion," in Proc. AAAI Conf. Artif. Intell., vol. 38, no. 16, 2024, pp. 17754–17762.

[10] X. Wang et al., "Searching for best practices in retrieval-augmented generation," in Proc. EMNLP, 2024, pp. 17716–17736.

[11] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting in retrieval-augmented large language models," in Proc. Conf. Empirical Methods Natural Lang. Process., 2023, pp. 5303–5315.

[12] M. Allen, U. Naeem, and S. S. Gill, "Q-Module-Bot: A generative AI-based question and answer bot for module teaching support," IEEE Trans. Educ., vol. 67, no. 5, pp. 793–802, Oct. 2024.

[13] A. Šarcˇevic´, I. Tomicˇic´, A. Merlin, and M. Horvat, "Enhancing programming education with open-source generative AI chatbots," in Proc. 47th MIPRO ICT Electron. Conv., May 2024, pp. 2051–2056.

[14] A. R. P. Damasceno et al., "STUART: An intelligent tutoring system for increasing scalability of distance education courses," in Proc. 19th Brazilian Symp. Hum. Factors Comput. Syst., Oct. 2020, pp. 1–10.

[15] S. Sakulwichitsintu, "ParichartBOT: A chatbot for automatic answering for postgraduate students of an open university," Int. J. Inf. Technol., vol. 15, no. 3, pp. 1387–1397, Mar. 2023.