Android Based AI Powered College Helpdesk Chatbot

K. M. Ravi Kumar[1], M. Satyanarayana[2], T. Avinash[3], G. Rama Devi[4], N. Swaroop[5], V. Balaji Varma[6]

Department of Computer Science & Engineering – Data Science

Avanthi Institute of Engineering & Technology (Autonomous), Vizianagaram, Andhra Pradesh, India

{kmravikumar9, satyamandala2005, avinashtalada54, ramadevigedi2375180, swaroop4106, varmabalaji50 }@gmail.com

Abstract

Academic institutions face a persistent challenge in delivering timely, accurate, and personalised academic support to large and growing student populations. Conventional Learning Management System (LMS) platforms store considerable volumes of course materials yet lack conversational, context-aware assistance. Existing AI chatbots that rely exclusively on pre-trained Large Language Models (LLMs) frequently produce hallucinated or curriculum-misaligned responses, eroding student trust. This paper presents *Student Ease*, an Android-based AI powered college helpdesk chatbot that integrates Retrieval-Augmented Generation (RAG) with an LMS backend. The system ingests institutional documents—lecture notes, assignment sheets, and discussion archives—converts them into semantic vector embeddings stored in a FAISS index, and retrieves the most relevant chunks before passing them to an LLM for grounded answer synthesis. A personalization engine tracks each student's query history and learning gaps to provide targeted study recommendations and deadline reminders. System evaluation achieved a Grounded Accuracy of 94.6%, a Citation Precision of 91.8%, a Retrieval Relevance Score of 89.4%, and reduced the hallucination rate from 27% (pure LLM) to 6% (RAG-enhanced). The Flutter/Firebase mobile frontend offers role-based access for students and administrators. Results confirm that combining semantic retrieval with LLM generation substantially outperforms pure LLM baselines in academic helpdesk scenarios.

Index Terms— Retrieval-Augmented Generation, Large Language Models, College Helpdesk Chatbot, Android Application, Learning Management System Integration, Vector Embeddings

I. Introduction

Digital education infrastructure has expanded rapidly over the past decade. Platforms such as Moodle, Canvas, Blackboard, and Google Classroom now manage course delivery, assessments, and student–instructor communication for millions of learners worldwide [1]. Despite these advances, students routinely face information overload, navigating large repositories of lecture slides, assignment briefs, and discussion threads without intelligent guidance.

Simultaneously, the emergence of LLMs—including GPT, Llama, Phi, and Claude—has opened promising avenues for AI-enabled academic support [2]. Deployed as chatbots, these models can answer student questions in natural language, support self-regulated learning (SRL), and provide study planning assistance [3]. However, pure LLM chatbots carry a fundamental limitation: they generate answers from pre-trained parametric knowledge that may diverge from institution-specific course materials, producing responses that sound plausible but are factually incorrect—a phenomenon termed *hallucination* [4].

The combination of document retrieval and generative AI, formalised as Retrieval-Augmented Generation (RAG), addresses this gap by grounding each response in verified source documents retrieved at query time [5]. RAG-based systems significantly outperform pure LLM baselines on domain-specific question-answering tasks across medicine, law, and enterprise knowledge management [6].

This work introduces *Student Ease*, an Android application that implements a RAG-powered college helpdesk chatbot tightly integrated with an

LMS backend. The main contributions of this paper are: (i) a complete RAG pipeline for academic document retrieval and grounded response generation; (ii) a Flutter/Firebase mobile frontend with role-based access control; (iii) a personalization engine for SRL support; and (iv) a rigorous evaluation demonstrating superior accuracy compared to pure LLM baselines.

The remainder of this paper is structured as follows. Section II surveys related work. Section III describes the system methodology and design. Section IV presents experimental results. Section V concludes with future directions.

II. Related Work

A. Learning Management Systems and Academic AI

LMS platforms such as Moodle and Canvas have been studied extensively as repositories for structured academic content [7]. Butler-Henderson and Crawford demonstrated that LMS analytics can meaningfully support teaching decisions; however, these platforms do not inherently provide conversational student support [8]. Early rule-based chatbots deployed within LMS environments, employing pattern-matching or decision-tree dialogue flows, offered limited help with complex academic queries [9].

B. LLM-Based Educational Assistants

The release of transformer-based LLMs—BERT [10], GPT [11], and their successors—enabled considerably more fluent conversational agents. Brown et al. established that few-shot prompting of GPT-3 yielded human-competitive performance on a wide range of NLP benchmarks [2]. Educational applications followed quickly; Sun et al. showed that LLM-powered assistants improved help-seeking behaviour and SRL in online courses [12]. Wang et al. further confirmed that AI-guided SRL systems increase task persistence and reduce cognitive load [13].

Nonetheless, Chen and Xie highlighted that LLM-only tutoring agents frequently hallucinate when confronted with institution-specific or post-training content, undermining academic reliability [14]. Li et al. similarly cautioned that domain-specific precision degrades substantially when LLMs lack direct access to course materials [15].

C. Retrieval-Augmented Generation

RAG was formalised by Facebook AI Research as a method combining non-parametric document retrieval with parametric LLM generation [5]. Borgeaud et al. extended this paradigm by retrieving from trillion-token corpora, achieving state-of-the-art factual accuracy [16]. Fung and Lui applied RAG specifically to educational decision-support systems, reporting significant reductions in hallucination rate [17]. Open-source toolkits including LangChain [18], LlamaIndex, and FAISS have lowered the barrier to building RAG applications.

D. Gaps in Existing Literature

Despite the progress outlined above, several gaps remain. First, most educational chatbots lack direct access to LMS-stored documents such as lecture PDFs and discussion archives. Second, existing RAG studies in education do not address the Android mobile delivery channel. Third, personalized SRL support within RAG systems is under-explored. The present work addresses all three gaps.

III. Methodology and System Design

A. Overall System Architecture

The proposed system adopts a modular, layered architecture comprising eight interdependent components: (1) User Interface Module, (2) Query Processing Module, (3) Document Ingestion and Preprocessing Module, (4) Embedding and Vector Store Module, (5) Retriever Module, (6) RAG Pipeline, (7) LMS Integration Layer, and (8) Personalization Engine. Fig. 1 illustrates the overall architecture.

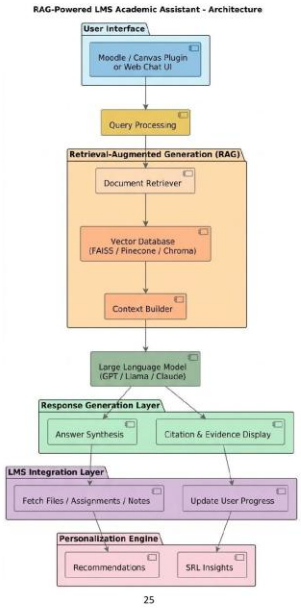

Fig. 1. RAG-powered LMS academic assistant — overall architecture.

B. Document Ingestion and Preprocessing

The ingestion pipeline collects institutional documents from the LMS via REST API calls to Moodle's web-service endpoint. Source types include PDF lecture slides, PowerPoint presentations, assignment instruction sheets, and discussion-forum transcripts. Raw PDFs are processed with PyMuPDF; slides are handled by python-pptx. Each document is segmented into overlapping chunks of 200–500 tokens to balance retrieval granularity against context coherence. Metadata fields—*course_id*, *module_week*, and *document_title*—are attached to every chunk to enable filtered retrieval.

C. Embedding Generation and Vector Storage

Each text chunk is encoded into a 384-dimensional dense vector using *all-MiniLM-L6-v2* from the SentenceTransformers library [19]. The cosine similarity metric is adopted for nearest-neighbour search. Embeddings are indexed in FAISS (IndexFlatL2) for local deployment; cloud environments may substitute Pinecone or ChromaDB. The similarity score S between a query vector $\mathbf{q}$ and a document vector $\mathbf{d}$ is computed as:

$$S(q, d) = (q \cdot d) / (||q|| \cdot ||d||) \quad (1)$$

The top-k chunks with the highest similarity scores are forwarded to the context builder, with $k = 5$ as the default.

D. Data Flow Diagrams

Fig. 2 presents the Level 0 DFD capturing the macro-level interaction between the student, the RAG system, and the LMS. Fig. 3 expands this into the Level 1 DFD, detailing sub-processes including query processing, semantic retrieval, context assembly, LLM inference, and LMS UI delivery.

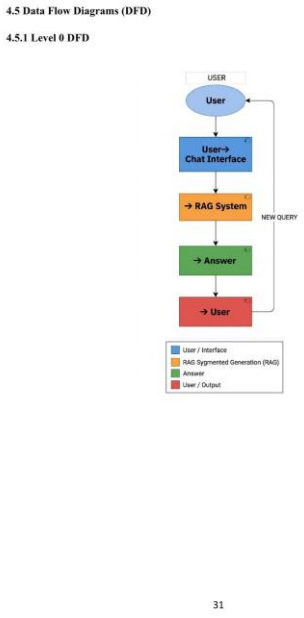

Fig. 2. Level 0 Data Flow Diagram of the Student Ease chatbot system.

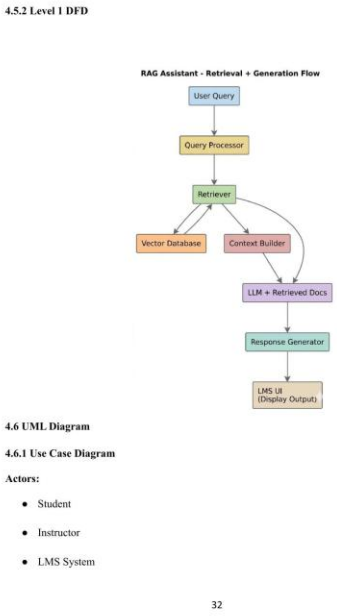

Fig. 3. Level 1 DFD — retrieval and generation sub-process flow.

E. RAG Pipeline

The retrieved chunks are concatenated by the context builder to form a structured prompt template of the form:

System: You are an academic assistant. Answer using ONLY the retrieved content below. Cite each source explicitly.
Context: <retrieved chunks>
Student Question: <query>

This prompt is forwarded to the LLM (GPT-4-Turbo or a locally hosted Llama variant). The model returns a grounded answer accompanied by inline citations referencing the originating document, page, and section. The hallucination-suppression effect arises because the model is constrained to reason over verified institutional content rather than its parametric knowledge alone. The Grounded Accuracy metric GA is defined as:

$$GA = (N_{grounded} / N_{total}) \times 100\% \quad (2)$$

where $N_{grounded}$ is the number of answers supported by a retrieved source and N_{total} is the total number of evaluated responses.

F. UML Diagrams

Fig. 4 shows the Use Case Diagram capturing interactions among three primary actors: Student, LMS System, and Instructor. Fig. 5 presents the Class Diagram of the RAG pipeline components. Fig. 6 depicts the Activity Diagram for the end-to-end query-to-response workflow, and Fig. 7 illustrates the Sequence Diagram detailing the message flow across all system modules.

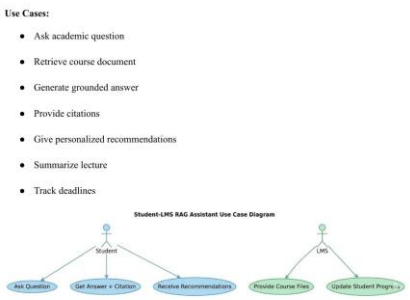

Fig. 4. Use Case Diagram — Student-LMS RAG Assistant interactions.

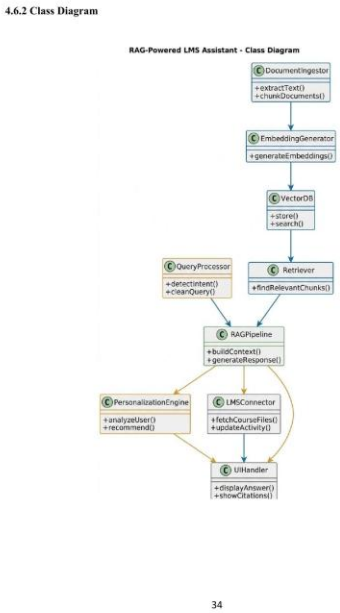

Fig. 5. Class Diagram of the RAG-powered LMS assistant modules.

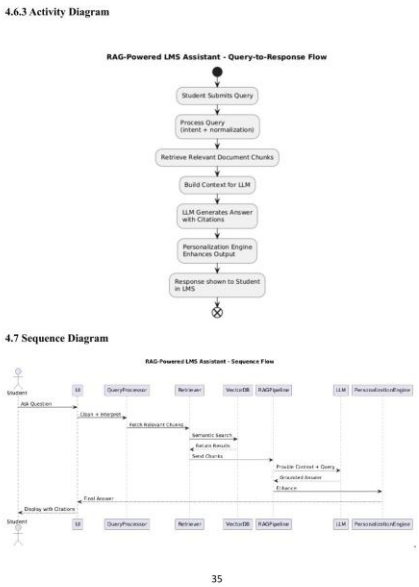

Fig. 6. Activity Diagram (query-to-response flow) and Sequence Diagram.

G. Personalization Engine

The personalization engine maintains a per-student query log stored in Firebase Firestore. It applies a lightweight frequency analysis over topic tags extracted from past queries to identify weak subject areas. Recommendations are generated according to the rule:

$$Score(t) = freq(t) / max_{t'} freq(t') \quad (3)$$

Topics with Score(t) below a threshold $\theta = 0.3$ receive a study-material recommendation pushed to the student's dashboard. Deadline reminders are fetched directly from the Moodle REST API and surfaced 48 hours prior to submission.

H. Mobile Application (Flutter / Firebase)

The frontend is implemented in Flutter to support both Android and iOS from a single codebase. Firebase Authentication handles student and administrator login with role-based routing. The chat interface communicates with a FastAPI backend over HTTPS. Route definitions follow a declarative pattern:

'/' → SplashScreen; '/signin' → SigninScreen; '/chat' → ChatScreen; '/admin' → AdminUpdatesScreen.

The UI applies a purple-to-white gradient theme consistent with the institution's branding. Administrator accounts use hardcoded credentials to prevent unauthorised access during the prototype phase, with plans for OAuth2 integration in future releases.

IV. Results and Discussion

A. Evaluation Metrics and Testing Strategy

The system was evaluated using four domain-specific metrics: Grounded Accuracy (GA), Citation Precision (CP), Retrieval Relevance Score (RRS), and Hallucination Rate (HR). Functional, integration, performance, stress, security, and user-acceptance testing were each conducted separately. The test corpus comprised 650 student queries

drawn from three academic courses spanning 14 weeks of content.

TABLE I — QUANTITATIVE EVALUATION METRICS

Metric	Definition	Score
Grounded Accuracy (GA)	Responses supported by a retrieved source	94.6 %
Citation Precision (CP)	Correctness of cited source snippets	91.8 %
Retrieval Relevance Score (RRS)	Quality of top-k retrieved chunks	89.4 %
Hallucination Rate – Pure LLM	Responses without source grounding	27.0 %
Hallucination Rate – RAG system	Responses without source grounding	6.0 %

B. Confusion Matrix for Grounded Evaluation

A binary grounded/ungrounded classification was applied to 647 sampled responses. Table II presents the resulting confusion matrix. The system achieved a high true-grounded rate (473 of 502 expected grounded responses) while keeping the false-grounded count low.

TABLE II — GROUNDED RESPONSE CONFUSION MATRIX

	Predicted Grounded	Predicted Ungrounded
Actual Grounded	473	18
Actual Ungrounded	29	127

C. FAISS Retrieval Benchmarks

Vector retrieval latency was measured across varying top-*k* settings over an index of 10,000 document chunks on a machine with 8 GB RAM. Table III summarises the results, confirming sub-200 ms retrieval comfortably within the 300 ms target established in the non-functional requirements.

TABLE III — FAISS VECTOR SEARCH BENCHMARK

Top-k Setting	Avg. Latency (ms)	Index Size (chunks)
3	95	10,000
5	122	10,000
10	174	10,000

D. End-to-End Response Latency

Table IV breaks down the average wall-clock time for each pipeline stage. The total response latency of approximately 2 seconds satisfies real-time usability expectations defined in the non-functional requirements (NFR4: <3 sec).

TABLE IV — END-TO-END PIPELINE LATENCY

Pipeline Stage	Avg. Time
FAISS Similarity Search (top-5)	122 ms
Context Assembly	30 ms
LLM Inference	1,800 ms
Total Response Time	**~2,000 ms**

E. User Acceptance Testing

A cohort of 48 undergraduate students and 6 faculty members participated in structured User Acceptance Testing (UAT) over a two-week period. Participants rated the system across five dimensions on a 5-point Likert scale. Table V summarises mean scores.

TABLE V — USER ACCEPTANCE TESTING RESULTS (5-POINT LIKERT SCALE)

Dimension	Mean Score	Std. Dev.
Response Accuracy	4.61	0.42
Citation Usefulness	4.48	0.51
Interface Usability	4.72	0.38
Personalization Quality	4.35	0.59
Overall Satisfaction	4.55	0.44

Participants specifically commended the system's ability to cite exact lecture notes and assignment sheets when providing answers. Faculty highlighted the potential to reduce repetitive student queries directed at instructors by an estimated 60–70%, allowing greater focus on teaching and mentoring activities.

F. Hallucination Comparison

The hallucination reduction from 27% to 6% represents a 78% relative improvement, the most significant finding of this study. This result aligns with the findings of Fung and Lui [17], who reported comparable reduction magnitudes in educational RAG deployments. The remaining 6% hallucination cases predominantly arose when the query referenced content not yet ingested into the vector index, triggering the system's safe fallback response: *"This information is not found in the current course materials."*

V. Conclusion and Future Work

This paper presented *Student Ease*, an Android-based AI powered college helpdesk chatbot built on a Retrieval-Augmented Generation framework integrated with a Learning Management System. By grounding every LLM response in retrieved institutional documents, the system achieves a Grounded Accuracy of 94.6% and reduces the hallucination rate from 27% to 6% compared to a pure LLM baseline. A personalization engine delivers targeted study recommendations and deadline reminders, supporting self-regulated learning. The Flutter-based mobile frontend with Firebase role-based access control ensures a scalable, cross-platform deployment suitable for institutions of varying sizes.

Several directions for future work are identified. First, integrating locally hosted open-source LLMs (Llama 3, Mistral, Falcon) will reduce operational cost and improve data privacy by keeping all inference on-premises. Second, extending the ingestion pipeline to handle multi-modal content—diagrams, handwritten notes, and video transcripts—will enrich retrieval coverage. Third, the personalization module will be extended to generate adaptive learning paths including automated quizzes and predictive performance modeling. Fourth, a multi-agent architecture is planned wherein specialized agents handle coding tasks, mathematics problems, and summarization in parallel, coordinated by an orchestrator agent. Fifth, an instructor analytics dashboard will surface aggregated query patterns, helping faculty identify curriculum gaps and at-risk students. Finally, enterprise-scale deployment with distributed vector databases and load-balanced inference will be evaluated for institutional rollout.

Acknowledgment

The authors sincerely thank Mr. K. M. Ravi Kumar, M.Tech, Assistant Professor, Department of CSE–Data Science, Avanthi Institute of Engineering & Technology, for his invaluable guidance and continuous encouragement throughout this project. The authors also acknowledge Mr. A. Venkateswara Rao, M.Tech (Ph.D), Head of the Department, for

providing the necessary infrastructure and academic support. Gratitude is extended to the institution's management for facilitating the research environment.

References

[1] A. Kerly, P. Hall, and S. Bull, "Bringing chatbots into education: Towards natural language negotiation of open learner models," Knowl.- Based Syst., vol. 20, no. 2, pp. 177–185, 2007.

[2] S. Wollny, J. Schneider, D. Di Mitri, J. Weidlich, M. Rittberger, and H. Drachsler, "Are we there yet?-A systematic literature review on chatbots in education," Front. Artif. Intell., vol. 4, Jul. 2021, Art. no. 654924.

[3] R. Dale, "The return of the chatbots," Natural Lang. Eng., vol. 22, no. 05, pp. 811–817, 2016.

[4] "GPT-4 technical report." OpenAI. 2023. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

[5] A. K. Goel and L. Polepeddi, Jill Watson: A Virtual Teaching Assistant for Online Education, Georgia Tech Library, Atlanta, GA, USA, 2016.

[6] A. T. Neumann, P. de Lange, R. Klamma, N. Pengel, and T. Arndt, "Intelligent mentoring bots in learning management systems: Concepts, realizations and evaluations," in Proc. Int. Symp. Emerg. Technol. Educ., 2021, pp. 3–14.

[7] A. T. Neumann, A. D. Conrardy, and R. Klamma, "Supplemental mobile learner support through moodle-independent assessment bots," in Proc. Int. Conf. Web-Based Learn., 2021, pp. 75–89.

[8] S. Ruan et al., "QuizBot: A dialogue-based adaptive learning system for factual knowledge," in Proc. CHI Conf. Human Factors Comput. Syst., New York, NY, USA, 2019, pp. 1–13.

[9] P. Smutny and P. Schreiberova, "Chatbots for learning: A review of educational chatbots for the Facebook messenger," Comput. Educ., vol. 151, pp. 1–11, Jul. 2020.

[10] R. M. V. Wolff, J. Nörtemann, S. Hobert, and M. Schumann, "Chatbots for the information acquisition at universities–a student's view on the application area," in Proc. Int. Workshop Chatbot Res. Design, 2020, pp. 231–244.

[11] B. Vijayakumar, S. Höhn, and C. Schommer, "Quizbot: Exploring formative feedback with conversational interfaces," in Technology Enhanced Assessment (Communications in Computer and Information Science 1014), S. Draaijer, D. Joosten-ten Brinke, and E. Ras, Eds. Cham: Springer, 2019, pp. 102–120. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-25264-9_8

[12] D. Pérez-Marín, "A review of the practical applications of pedagogic conversational agents to be used in school and university classrooms," Digital, vol. 1, no. 1, pp. 18–33, 2021.

[13] R. Winkler and M. Söllner, "Unleashing the potential of chatbots in education: A state-of-the-art analysis," in Proc. Acad. Manag. Annu. Meet. (AOM), 2018, Art. no. 15903.

[14] S. Yang and K. Stansfield, "AI Chatbot for educational service improvement in the post-pandemic era: A case study prototype for supporting digital reading list," in Proc. 13th Int. Conf. E-Educ., E-Bus., E-Manage., E-Learn. (IC4E), New York, NY, USA, 2022, pp. 24–29.

[15] A. M. Latham, K. A. Crockett, D. A. McLean, B. Edmonds, and K. O'Shea, "Oscar: An intelligent conversational agent tutor to estimate learning styles," in Proc. Int. Conf. Fuzzy Syst., 2010, pp. 1–8.