Int. j. adv. multidisc. res. stud. 2026; 6(3):687-691

Received: 01-04-2026
Accepted: 10-05-2026

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

An AI-Enabled Dual-Database Chatbot for Academic Advising and Training Management in Distance Higher Education

[1] Thi Thu-Hien Phung, [2] Tuan-Anh Truong, [3] Quang-Ninh Dinh

[1] Department of Electronics, Thai Nguyen University of Technology, Thai Nguyen 24000, Vietnam

[2] Department of Electrical Engineering, Thai Nguyen University of Technology, Thai Nguyen 24000, Vietnam

[3] Education Technology and Adaptive Learning Institute, Thai Nguyen University of Technology, Thai Nguyen 24000, Vietnam

DOI: https://doi.org/10.62225/2583049X.2026.6.3.6347

Corresponding Author: **Quang-Ninh Dinh**

Abstract

This article presents the design, implementation, and initial evaluation of an artificial-intelligence-enabled chatbot for academic advising and training management in a university distance-learning context. The study addresses a practical management problem: distance learners often need timely, accurate, and personalized answers about regulations, curricula, credit registration, learning progress, assessment, and graduation conditions, while academic advisors face high volumes of repetitive inquiries during peak periods. The proposed solution combines Retrieval-Augmented Generation (RAG), a vector database for unstructured institutional documents, a relational database for structured student and training records, intent classification, prompt engineering, guardrails, and a web-based chat interface. The system transforms formal documents such as training regulations, advisor handbooks, curricula, and appendices into semantic embeddings, retrieves relevant passages at query time, and generates grounded answers with a lower risk of hallucination. For personalized advising, structured data are queried through a controlled relational layer. Initial evaluation shows approximately 85% accuracy for regulation-related questions, an average response time of about five seconds, effective storage of conversation history, and an estimated 60-70% reduction in repetitive manual advising workload. The contribution of the article is a practical dual-database architecture and implementation framework that can support digital transformation in training management, especially for open and distance education institutions in contexts where policy documents and student records must be handled together.

Keywords: Academic Advising, Distance Education, Chatbot, Retrieval-Augmented Generation, Training Management, Vector Database, Relational Database, Learning Support

1. Introduction

Digital transformation in higher education is no longer limited to digitizing documents or moving courses to a learning management system. It increasingly requires institutions to transform administrative and academic-support processes into data-driven services that are accurate, timely, scalable, and learner-centered. This requirement is particularly urgent in distance education, where students are geographically dispersed, rely heavily on online communication, and may not have immediate access to academic advisors during working hours.

In many distance-learning programs, students repeatedly ask questions about registration procedures, credit accumulation, grade-point average, exemption or recognition of prior learning, examination regulations, academic warning, graduation requirements, and administrative forms. These questions are usually answered through advisor messages, phone calls, student handbooks, or long PDF/Word documents. Such channels are useful but not optimal: advisors cannot provide continuous support, learners need to manually search lengthy documents, and policy updates may not be reflected quickly in printed or static materials.

Generative artificial intelligence and large language models (LLMs) create new opportunities for intelligent academic-support systems. However, a generic LLM is not suitable for institutional advising unless its answers are grounded in verified documents and constrained by governance rules. In academic management, a wrong answer about credits, graduation conditions, or examination procedures can affect a student's learning plan. Therefore, an advising chatbot must combine natural language interaction with traceable knowledge retrieval, structured database access, privacy protection, and administrative oversight.

This study develops and analyzes an AI-enabled chatbot for academic advising and training management in a distance higher education setting. The article does not discuss the administrative title or internal code of the underlying institutional research work. Instead, it extracts the transferable research problem, solution architecture, implementation method, evaluation results, and implications for technology application in educational management.

2. Literature Background and Conceptual Basis

2.1 Large language models, hallucination, and the need for grounding

Modern LLMs are based on the Transformer architecture, which uses self-attention mechanisms to model relations among tokens in a sequence and enables efficient parallel computation (Vaswani *et al*., 2017) [8]. These models can generate fluent answers, summarize documents, and support conversational interaction. Nevertheless, they may produce unsupported or inaccurate statements when the required knowledge is absent, outdated, or overly domain-specific.

In educational administration, hallucination is a critical risk. A chatbot that invents a rule, misquotes a regulation, or fabricates a deadline can undermine trust and create operational consequences. UNESCO's guidance on generative AI emphasizes the need for human-centered, regulated, and capacity-building approaches when applying GenAI in education and research (UNESCO, 2023) [7]. Thus, technical design must be accompanied by guardrails, source control, data governance, and human review.

2.2 Retrieval-Augmented Generation for institutional knowledge

Retrieval-Augmented Generation addresses the limitation of static model knowledge by retrieving relevant external documents before generating an answer. Lewis *et al*. (2020) [2] introduced RAG as a method that combines parametric memory in a pre-trained sequence-to-sequence model with non-parametric memory accessed through a dense vector index. In institutional settings, this approach allows a chatbot to answer questions based on official regulations, manuals, and curricula without retraining the whole language model after every document update.

RAG is especially suitable for training-management documents because such documents are often long, hierarchical, and frequently updated. Instead of relying on exact keyword matching, semantic search can retrieve passages that are close in meaning to the student's question. For example, a learner may ask, 'What should I do if I fail a subject?', while the official document may use the term 'retake a course' or 'academic result below the passing threshold.' Embedding-based search can bridge this vocabulary gap.

2.3 Conversational AI, vector databases, and structured records

A complete advising chatbot needs more than document retrieval. It must classify user intent, extract entities such as student ID or course name, preserve dialogue context, call business logic, and format responses safely. Rasa provides open-source conversational AI components for natural language understanding, dialogue management, and integration with third-party systems (Rasa, 2025) [6]. Vector databases such as Chroma are designed to store embeddings, metadata, and searchable representations of documents for retrieval workflows (Chroma, 2026) [1].

However, many advising questions require structured data, not only text retrieval. For example, a student may ask how many credits remain, whether they have met a graduation condition, or which failed modules affect their progress. Such queries require reliable access to student records, grades, curricula, and credit structures. Therefore, the proposed model adopts a dual-database architecture: a vector database for unstructured knowledge and a relational database for structured academic records.

3. Research Methodology

The study follows a design science research orientation. The artifact is an operational chatbot architecture and prototype for distance-learning academic advising. The research process includes: (i) analysis of advising problems and institutional data sources; (ii) specification of functional and non-functional requirements; (iii) design of a multi-layer and dual-database architecture; (iv) implementation of core modules; (v) functional testing and preliminary performance evaluation; and (vi) formulation of implementation lessons and future development directions.

The institutional data sources include formal distance-training regulations, an advisor handbook with frequently asked questions, detailed curricula for several undergraduate programs, appendices on admission and priority rules, and structured training records. These sources represent two knowledge types. The first is unstructured or semi-structured text, which must be chunked, embedded, and searched semantically. The second is structured information, which must be stored in relational tables and queried through secure, validated logic.

Table 1: Design science research components and operationalization

Research component	Operationalization in the study	Expected contribution
Problem analysis	Identify repeated advising questions, document-search difficulty, and need for 24/7 learner support	Define the management problem and user requirements
Knowledge engineering	Convert official documents into semantically searchable chunks with metadata	Improve retrieval accuracy and document traceability
System design	Combine RAG, intent routing, SQL access, prompt rules, and admin dashboard	Create an integrated architecture for both general and personalized advising
Evaluation	Assess answer accuracy, response time, workload reduction, history storage, and technical stability	Provide evidence for practical adoption and further refinement

4. System Architecture and Design

4.1 Multi-layer architecture

The system is designed as a multi-layer architecture. The presentation layer provides a chat widget for the institutional website, supports markdown rendering, audio feedback, suggested questions, and streaming responses. The dialogue-orchestration layer uses intent classification and slot memory to determine whether a question is a general regulation inquiry, a procedure question, a personal academic-progress request, or an out-of-scope message. The AI engine layer performs semantic retrieval, prompt construction, generation, and post-processing. The data layer combines ChromaDB for document embeddings and SQL Server for structured records.

The most important design principle is separation of responsibilities. The LLM is not allowed to directly access raw student records or invent institutional regulations. It receives controlled context from either the vector-retrieval pipeline or the structured-query pipeline. Backend guardrails validate inputs, sanitize prompts, translate internal status codes into user-friendly language, and reject outputs that expose raw JSON, database keys, or internal variable names.

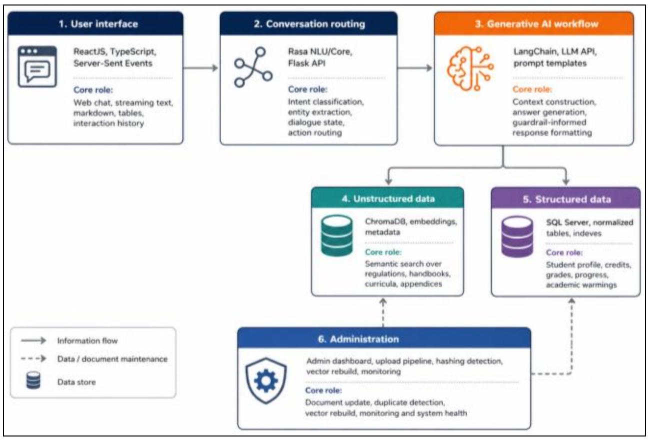

Fig 1: Proposed architecture for an AI-enabled advisory chatbot in distance education

4.2 Dual-database model

The dual-database model addresses the mixed nature of academic advising knowledge. Regulations, handbooks, curriculum descriptions, and policy appendices are text-heavy and may contain hierarchical structures such as chapters, articles, clauses, and points. These documents are processed through chunking, metadata assignment, embedding, and vector storage. In contrast, GPA, credit accumulation, subject status, registration information, and program completion are structured records that require SQL queries, indexes, and referential integrity.

This design avoids two common weaknesses. First, it prevents the chatbot from using a generative model as a substitute for an authoritative database. Second, it prevents a relational database from being misused for open-ended semantic search over policy documents. Each data technology is used for the task it handles best.

Table 2: Main technical modules of the proposed chatbot system

Layer / module	Main technology	Core role
User interface	ReactJS, TypeScript, Server-Sent Events	Web chat, streaming text, markdown tables, interaction history
Conversation routing	Rasa NLU/Core, Flask API	Intent classification, entity extraction, dialogue state, action routing
Generative AI workflow	LangChain, LLM API, prompt templates	Context construction, answer generation, guardrail-informed response formatting
Unstructured data	ChromaDB, embeddings, metadata	Semantic search over regulations, handbooks, curricula, appendices
Structured data	SQL Server, normalized tables, indexes	Student profile, credits, grades, progress, academic warnings
Administration	Admin dashboard, upload pipeline, hashing	Document update, duplicate detection, vector rebuild, monitoring

4.3 Retrieval-Augmented Generation pipeline

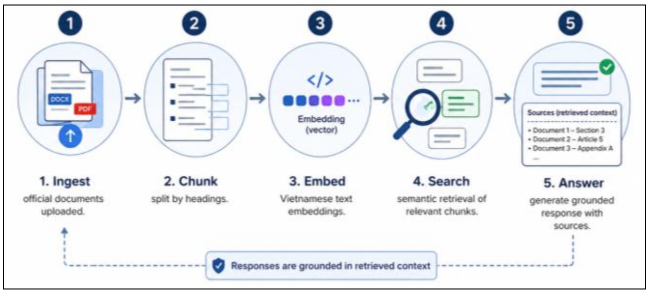

Fig 2: Retrieval-Augmented generation workflow used to ground responses in institutional documents

The RAG pipeline begins when an administrator uploads official documents. The system computes a hash to prevent duplicate ingestion, parses the document, splits it according to logical headings, creates embeddings, and stores each chunk with metadata. When a learner asks a question, the system expands abbreviations and synonyms, retrieves the most relevant chunks, and inserts them into a prompt. The LLM then generates a concise answer based only on retrieved context and institutional response rules.

Semantic mapping is important in this context because students often use informal expressions, abbreviations, or non-standard terminology. Mapping terms such as 'GPA,' 'credits,' 'retake,' 'graduation,' and 'failed subject' to institutional equivalents increases recall while preserving the original query.

4.4 Personalized academic-progress pipeline

Personalized advising is handled separately. If the user provides a valid student identifier and asks about individual progress, the dialogue layer routes the request to a structured-data action. The backend validates the identifier, retrieves relevant records, computes academic indicators, and prepares a normalized context for the LLM. The final answer may include accumulated credits, unmet requirements, failed or retake subjects, and recommended next steps. The model is not given unrestricted database access; it only receives formatted and policy-compliant outputs from controlled queries.

SQL indexing is used to reduce query time for common lookups, such as student ID, subject code, curriculum code, semester, grade status, and completion status. Microsoft documentation emphasizes the role of well-designed indexes in improving query performance and reducing unnecessary I/O for frequently used queries (Microsoft, 2025) [5].

5. Implementation Techniques

5.1 Data processing and document updating

The system applies heading-based chunking rather than random fixed-length splitting. This maintains semantic integrity because each chunk corresponds more closely to a meaningful regulation, article, procedure, or curriculum element. Chunk size is controlled to avoid both context loss

and excessive noise. Moderate overlap between adjacent chunks reduces the risk of losing information at boundaries.
Document updates are managed through an administrator dashboard. After a new document is uploaded, the system calculates a hash value, skips duplicate files, parses content, rebuilds embeddings, and stores metadata. This design allows regulatory changes to be reflected quickly without retraining an LLM. It also creates an operational workflow that non-technical academic managers can supervise.

5.2 Prompt engineering and guardrails

The system prompt defines the chatbot as an academic advisor for distance learners and establishes behavioral rules: use clear language, answer only from official context, do not expose raw system data, avoid unsupported claims, and refer students to human staff when information is insufficient. Post-processing guardrails check for raw data leakage, internal variable names, JSON-like structures, or responses that fail to follow language and formatting requirements.
Guardrails also support accountability. In a management setting, the chatbot should not replace formal decision-making. It should guide learners, explain regulations, and summarize academic indicators, but final administrative decisions still belong to authorized staff and official systems.

5.3 Streaming interaction and user experience

The web interface uses streaming responses so that the first part of the answer appears quickly instead of making the user wait for the entire answer. Server-Sent Events enable a server to push messages to a web page over an HTTP connection, making them appropriate for one-way text streaming from AI systems to browser interfaces (MDN, 2026) [4]. The interface also supports conversation history, suggested questions, and markdown formatting for tables such as grade summaries or progress reports.
Good user experience is not a secondary issue in distance education. If learners perceive the system as slow, confusing, or unreliable, they will return to informal messaging channels. Therefore, response speed, friendly wording, source-grounded answers, and clear escalation to human advisors are part of the educational value of the system.

6. Evaluation Results

Fig 3: Evaluation summary: Accuracy, response speed, workload reduction and database optimization

Preliminary evaluation indicates that the system meets its main functional objectives. For regulation-related questions, it achieved about 85% answer accuracy in the tested scope and did not exhibit uncontrolled hallucination when relevant context was retrieved. The average response time was approximately five seconds, which is suitable for a web-based academic-advising service. The conversation history module reduced repeated questions, while SQL indexes and caching improved response stability.
The expected managerial impact is also significant. The system is estimated to reduce 60-70% of repetitive manual advising workload, especially for common questions during registration, assessment, and graduation periods. This does not eliminate the need for advisors; rather, it allows them to focus on complex cases, exceptions, and learner support requiring human judgment.

Table 3: Preliminary evaluation summary

Evaluation criterion	Target / expectation	Observed or estimated result	Interpretation
Regulation question accuracy	Approximately 90%	Approximately 85%	Promising result; additional test data and document refinement are needed
Average response time	Under 10 seconds	About 5 seconds	Acceptable for online academic support
Hallucination control	No unsupported regulation answers	No uncontrolled hallucination observed in grounded cases	Guardrails and RAG reduced risk
Manual advising workload	Meaningful reduction	Estimated 60-70% reduction in repetitive questions	High practical value during peak periods
Maintainability	Fast document update	Admin upload and vector rebuild workflow available	Supports evolving regulations

7. Discussion

The findings suggest that AI chatbots can become practical digital assistants for training management when they are designed as governed information systems rather than simple generative interfaces. The dual-database architecture is the core contribution because academic advising involves both policy interpretation and individual academic-record analysis. A pure RAG system may answer regulation questions well but cannot reliably compute student progress. A pure SQL chatbot may retrieve records but cannot explain long policy documents. Their combination creates a more complete advising service.
The system also illustrates a feasible path for digital transformation in resource-constrained educational units. By using existing documents and records, the institution can create a knowledge base without building a full enterprise data warehouse from the beginning. The admin dashboard enables academic staff to maintain documents. The chatbot interface can be embedded in a website using an independent deployment model, reducing risk to the main institutional web system.
Nevertheless, the current implementation has limitations. It primarily supports Vietnamese-language advising, depends on internet access for LLM API calls, and requires

continued curation of documents. Accuracy may decline if regulations change but the knowledge base is not updated. Long-context questions and image-based documents remain challenging. In addition, privacy governance must be strengthened as more student-specific functions are added.

8. Recommendations for Institutional Deployment

First, institutions should define a clear knowledge governance process. Every document ingested into the chatbot should have an owner, version, effective date, and review schedule. Outdated documents should be archived or deactivated to prevent contradictory answers.

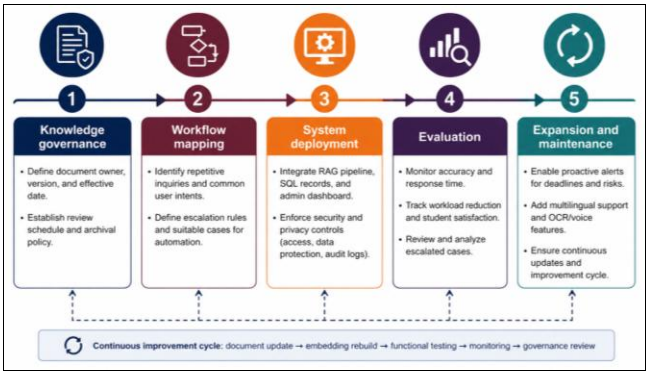

Fig 4: Practical implementation roadmap for institutional adoption and maintenance

Second, academic advising workflows should be mapped before automation. Repetitive questions, rule-based procedures, and status lookups are appropriate for automation. Ambiguous cases, complaints, appeals, and decisions involving exceptions should be escalated to human staff.

Third, evaluation should combine technical and managerial indicators: answer accuracy, retrieval relevance, response time, number of resolved inquiries, advisor workload reduction, student satisfaction, and number of escalated cases. These indicators should be reviewed each semester.

Fourth, the system should be developed toward proactive support. Instead of only responding to questions, future versions can notify students about registration deadlines, new grades, academic warnings, missing credits, and graduation readiness.

9. Conclusion

This article has presented a transferable model for applying AI technology to academic advising and training management in distance higher education. The proposed system combines RAG, a vector database, relational student records, intent routing, prompt engineering, guardrails, and a web-based chat interface. The architecture addresses the central problem of distance-learning support: students need fast, reliable, personalized information, while advisors need to reduce repetitive workload without sacrificing accuracy or accountability.

The preliminary results are encouraging: approximately 85% accuracy for regulation-related questions, average response time of about five seconds, conversation-history support, and a meaningful estimated reduction in repetitive manual advising. The main scientific and practical contribution is the dual-database architecture, which enables simultaneous handling of unstructured institutional documents and structured academic records. Future work should expand multilingual support, strengthen offline and local deployment options, integrate OCR and voice interaction, and develop predictive analytics for early academic-risk warning.

10. References

1. Chroma. Chroma documentation: Introduction, 2026. https://docs.trychroma.com/docs/overview/introduction
2. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, *et al*. Retrieval-Augmented Generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 2020. https://arxiv.org/abs/2005.11401
3. LangChain. Build a RAG agent with LangChain, 2026. https://docs.langchain.com/oss/python/langchain/rag
4. MDN Web Docs. Server-sent events, 2026. https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events
5. Microsoft. SQL Server indexes. Microsoft Learn, 2025. https://learn.microsoft.com/en-us/sql/relational-databases/indexes/indexes
6. Rasa. Introduction to Rasa Open Source and Rasa Pro, 2025. https://rasa.com/docs/rasa/
7. UNESCO. Guidance for generative AI in education and research, 2023. https://www.unesco.org/en/articles/guidance-generative-ai-education-and-research
8. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, *et al*. Attention is all you need. Advances in Neural Information Processing Systems, 2017. https://arxiv.org/abs/1706.03762