© May 2026| IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

An Intelligent Cloud-Native Framework for Smart Classroom Management Using Agentic AI and LLMs

Prof. Kavita Babalad[1], Kavana Basavaraju[2], Sukruth MS[3], Spoorthi SV[4], Tanushree SK [5]
[1,2,3,4,5]*School of C&IT REVA University Bengaluru, India*

Abstract—The modern education system faces challenges in hybrid and digital learning, where teachers are overloaded with ad-ministrative tasks and students lack personalized, real-time support. This project proposes "An Intelligent Cloud-Native Framework for Smart Classroom Management", an AI-powered platform designed to enhance learning and reduce educator workload. It features a Streamlit frontend and FastAPI backend, powered by a multi-agent system using LangChain, Groq API, and Llama 3.3-70B. Five AI agents handle notes, doubts, quizzes, summaries, and engagement. To ensure accuracy, a Retrieval-Augmented Generation (RAG) pipeline with FAISS is used, along with SQLite for conversation memory. The system is containerized using Docker for scalable deployment, delivering a reliable and efficient smart classroom solution.

***Index Terms*—Agentic AI, Large Language Models (LLMs), Smart Classroom, Multi-agent Systems, Retrieval-Augmented Generation (RAG), Cloud-native Architecture.**

I.INTRODUCTION

The chalkboard-and-podium era of education isn't just fad-ing; it has been fundamentally rewritten. Over the last decade, we have witnessed a tectonic shift in how knowledge is shared and consumed. What began as a slow and cautious lean toward digital tools has accelerated into a complete transformation of the educational landscape. Classrooms are no longer con-fined to physical spaces; instead, learning now occurs across distributed environments enabled by cloud platforms, online resources, and real-time communication technologies. This shift has effectively dissolved geographical barriers, allowing students from remote regions to access the same high-quality content that was once limited to elite institutions. The transition to hybrid and fully online learning models has redefined accessibility and reach, making education more inclusive than ever before.

However, this transformation is not without its shortcomings. While access to information has improved dramatically, the quality of interaction and engagement has not evolved at the same pace. The digital ecosystem that supports modern education often prioritizes content delivery over meaningful learning experiences. As a result, students are exposed to vast amounts of information but are left without adequate support mechanisms to process, understand, and apply that knowledge effectively. This imbalance highlights a deeper issue within the current educational infrastructure—one that becomes increasingly evident as reliance on digital systems continues to grow.

The primary issue lies in the tools we rely on. Most modern academic institutions operate through Learning Management Systems (LMS) that, while functional, are essentially digital filing cabinets. These systems serve as centralized repositories where lecture slides, reading materials, recorded sessions, and assignment links are stored. While they improve organizational efficiency and accessibility, they do not actively contribute to the learning process itself. They lack the ability to adapt to individual learners, provide contextual explanations, or re-spond dynamically to student needs. In essence, they replicate the administrative aspects of education without enhancing its intellectual or interactive dimensions.

In a traditional classroom setting, learning is inherently interactive. Students can raise questions, seek clarification instantly, and engage in discussions that deepen their understanding. This immediacy fosters a feedback loop that is critical for effective learning. In contrast, digital environments often introduce delays and friction into this process. A student encountering difficulty must rely on asynchronous communication channels such as emails or discussion forums, where responses may take hours or even days.

During this time, confusion can compound, leading to disengagement and reduced learning outcomes. This phenomenon, often referred to as the "engagement gap," represents one of the most significant challenges in modern digital education.

Parallel to these developments, we are witnessing another major transformation driven by the rapid advancement of Artificial Intelligence. The emergence of Large Language Models (LLMs) has fundamentally changed how users interact with information. Unlike traditional systems that rely on keyword-based search, LLMs enable conversational interaction, allowing users to pose questions in natural language and receive detailed, contextually relevant responses. This shift from retrieval-based systems to generative, conversational systems introduces new possibilities for education.

Imagine a learning environment where a student is not limited to static content but has access to an intelligent assistant capable of explaining complex concepts, simplifying difficult topics, and adapting explanations based on the learner's level of understanding. Such a system could provide continuous support, bridging the gap between classroom instruction and independent study. It could function as a personalized tutor, available at any time, capable of reinforcing learning through interaction and feedback.

Despite these promising capabilities, the integration of general-purpose AI into educational systems presents significant challenges. One of the most critical concerns is the issue of hallucination, where AI models generate responses that appear plausible but are factually incorrect. In an academic setting, where precision and accuracy are essential, such errors can have serious consequences. A single incorrect explanation in a technical subject or a misrepresented historical fact can undermine the learning process and erode trust in the system. In addition to accuracy concerns, generic AI models lack contextual alignment with specific curricula. They are trained on broad and diverse datasets and therefore do not inherently understand the structure, objectives, or constraints of a particular course. They are unaware of syllabus boundaries, institutional guidelines, or instructor-specific teaching methods. This lack of contextual awareness limits their effectiveness as educational tools, as they may provide responses that are either too generalized or not aligned with the intended learning outcomes.

These challenges reveal a critical gap in the current state of educational technology. On one hand, digital platforms provide the infrastructure for content delivery, and on the other hand, AI provides the capability for intelligent interaction. However, there is no cohesive system that effectively integrates these two dimensions while ensuring reliability, contextual relevance, and scalability. Bridging this gap requires a new approach—one that combines structured knowledge sources with intelligent processing and real-time interaction.

This project is born out of that necessity. It aims to move be-yond the limitations of static file-sharing systems and generic AI tools by introducing a dynamic, intelligent, and curriculum-aware framework for smart classroom management. By lever-aging multi-agent architectures, the system distributes tasks across specialized components, ensuring efficiency and precision. The integration of Retrieval-Augmented Generation (RAG) ensures that responses are grounded in verified academic content, thereby mitigating the risk of hallucination and maintaining alignment with course materials.

Furthermore, the system is designed to function as an active participant in the learning process rather than a passive repository of information. It enables real-time interaction, supports multi-turn conversations, and provides context-aware assistance tailored to individual queries. At the same time, it reduces the administrative burden on educators by automating repetitive tasks such as note generation, quiz creation, and content summarization.

By grounding AI within structured academic contexts and deploying it within a scalable, cloud-native architecture, the proposed system seeks to redefine how digital learning environments operate. It transforms them from static platforms into intelligent ecosystems capable of supporting continuous, interactive, and personalized learning. In doing so, it addresses the core limitations of existing systems while unlocking the full potential of AI in education, ultimately providing students with the reliable, responsive, and engaging support they need to succeed.

II.PROPOSED METHODOLOGY

The methodology for An Intelligent Cloud-Native Frame-work for Smart Classroom Management involves creating a cohesive, containerized application where every service, from the frontend user interface to the backend core intelligence and data layer, is designed for cloud-native deployment and real-time interaction. The entire system is defined and managed as a set of docker-compose services.

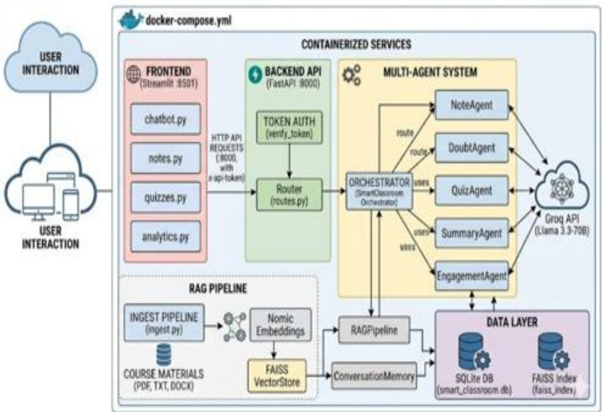

Fig. 1. Detailed Containerized System Architecture and End-to-End Data Flow Diagram.

The methodology is structured to provide accurate, verified academic assistance while reducing the administrative load on teachers. The design ensures high performance and scal-ability by leveraging modern web framework and hardware-accelerated inference.

A. System Architecture

The proposed system follows a cloud-native, multi-layered architecture designed to support real-time interaction, scalability, and efficient processing of academic tasks within a smart classroom environment. It adopts a client-server model consisting of a frontend interface, backend services, a centralized orchestration layer, and an intelligent multi-agent system. This layered approach ensures modularity, ease of maintenance, and the ability to scale seamlessly across different deployment environments.

As shown in the architecture diagram in Fig. 1, the frontend, developed using Streamlit, serves as the primary interface for users to interact with the system. It provides a chat-based interaction model where users can perform various academic tasks such as asking questions, generating summaries, creating quizzes, and accessing notes. User inputs are captured and transmitted as structured HTTP requests to the backend.

The backend, implemented using FastAPI, functions as the core processing unit of the system. It handles request validation, token-based authentication, and routing of requests to appropriate components. The asynchronous capabilities of FastAPI enable the system to support multiple concurrent users efficiently, ensuring minimal latency even under high load conditions.

At the core of the system lies the Smart Classroom Orchestrator, which acts as the central decision-making component. It analyzes user queries and determines the appropriate processing pathway. This routing is primarily driven by Large Language Model (LLM)-based intent recognition, allowing the system to understand semantic context. In cases where the intent is ambiguous, a fallback keyword-based routing mechanism ensures reliability and prevents processing failures.

The multi-agent system forms the intelligence layer of the architecture. It consists of task-specific agents such as Note, Doubt, Quiz, Summary, and Engagement agents. Each agent is responsible for executing a distinct function, enabling modular processing and improving overall system efficiency. This design reduces cognitive overload on a single model and enhances response accuracy by assigning specialized responsibilities.

For knowledge retrieval, the system integrates a Retrieval-Augmented Generation (RAG) pipeline supported by a FAISS vector database. This ensures that responses are generated us-ing verified academic content rather than relying solely on pre-trained model knowledge. Additionally, a SQLite-based memory module maintains conversation history, enabling context-aware and multi-turn interactions.

The entire system is containerized using Docker and deployed in a cloud-native environment, ensuring portability, scalability, and consistent performance across different plat-forms. This architecture enables seamless integration of user interaction, intelligent processing, and data-driven retrieval to deliver a responsive and reliable smart classroom system.

1) Multi-Agent System Design: The proposed system employs a multi-agent architecture to efficiently handle diverse academic tasks while avoiding the limitations of monolithic Large Language Model (LLM) systems. Instead of relying on a single model to perform all operations, the system distributes responsibilities across specialized agents, each designed for a specific function.

At the center of this architecture is the Smart Classroom Orchestrator, which manages the coordination between agents.

It interprets user queries using LLM-based reasoning and routes them to the most suitable agent. This dynamic routing ensures that each request is handled by a component optimized for that specific task.

The system includes five primary agents: Note Agent, Doubt Agent, Quiz Agent, Summary Agent, and Engagement Agent. Each agent operates independently with its own processing logic and prompt design. This modular approach improves scalability, enables parallel processing, and allows the system to be easily extended with additional agents in the future.

2) Implementation and Processing Pipeline: The implementation follows a structured pipeline that ensures efficient flow from user input to response generation. The process begins when a user submits a query through the Streamlit interface, which is transmitted as an HTTP request to the FastAPI backend.

The backend validates the request and forwards it to the Smart Classroom Orchestrator. The orchestrator performs in-tent analysis and routes the request to the appropriate agent. Each agent executes task-specific logic, including optional retrieval, processing, and response generation.

For factual queries, the system invokes the RAG pipeline, where the input is converted into vector representations and compared against embeddings stored in the FAISS database. Relevant content is retrieved and injected into the LLM prompt to ensure accurate and context-aware responses.

3) RAG-Based Retrieval Mechanism: The RAG pipeline plays a critical role in ensuring response accuracy and pre-venting hallucinations. During the ingestion phase, course materials such as PDFs and documents are segmented into smaller semantic chunks and converted into vector embed-dings. These embeddings are stored in the FAISS vector database for efficient retrieval.

During runtime, user queries are transformed into vector representations and matched against stored embeddings using semantic similarity search. The retrieved content is then combined with the query through context injection before being passed to the LLM. This ensures that generated responses are grounded in verified academic content.

4) Conversational Memory and Context Handling: The system incorporates a conversational memory mechanism to support context-aware and multi-turn interactions. A SQLite-based memory module stores user queries and system responses, each associated with a unique session identifier.

When processing a new query, the system retrieves relevant conversation history and incorporates it into the prompt. This allows the system to understand follow-up questions, resolve ambiguities, and maintain continuity in dialogue, enhancing the overall user experience.

5) Data Layer and Storage Design: The system uses a hybrid storage approach to manage both structured and un-structured data. SQLite is used as a relational database to store session data, conversation history, and system logs. This ensures persistent storage and efficient retrieval of structured information.

For semantic retrieval, the system utilizes FAISS to store high-dimensional vector embeddings generated from academic content. A centralized database management layer ensures efficient communication between these storage systems and other components of the architecture.

6) UI/UX Interaction Layer: The Streamlit-based frontend provides a clean and intuitive user interface designed for ease of use. The system adopts a chat-based interaction model, enabling users to communicate naturally with the system.

The interface supports structured outputs for tasks such as summaries, quizzes, and notes, ensuring that generated content is organized and easy to understand. Session management features allow users to maintain continuity across interactions, improving usability and engagement.

7) Deployment and Scalability: The system is deployed using a cloud-native approach with Docker-based container-ization, ensuring portability and consistent execution across environments. This allows the system to scale horizontally by deploying multiple instances during peak usage periods.

Heavy computational tasks, particularly LLM inference, are offloaded to the Groq API, enabling high-speed processing without requiring local high-performance hardware. The combination of asynchronous backend processing and containerized deployment ensures that the system remains scalable, efficient, and responsive under varying workloads.

III. RESULTS AND DISCUSSION

The proposed framework demonstrated significant improvements in student engagement when compared to traditional Learning Management System (LMS) platforms. Unlike conventional systems that primarily function as static repositories, the proposed system enables continuous and interactive learning through a conversational interface. Students are able to actively engage with the system by asking questions, generating summaries, and interacting with dynamically generated content such as quizzes and notes. This shift from passive content consumption to active interaction contributes to a more engaging and responsive learning experience.

A key factor contributing to this improvement is the integra-tion of the Retrieval-Augmented Generation (RAG) pipeline. By grounding responses in verified academic content retrieved from the FAISS vector database, the system ensures that outputs remain accurate and contextually relevant. This significantly reduces the occurrence of hallucinations commonly associated with generic Large Language Models, thereby im-proving the reliability and trustworthiness of the system in academic use cases.

In addition to accuracy, the system also demonstrates im-proved response efficiency. By leveraging high-speed inference through the Groq API, the framework is able to generate responses with minimal latency. This near real-time interaction capability allows students to receive immediate clarification on their queries, helping maintain learning continuity and reducing frustration caused by delays in traditional systems.

The automation of academic and administrative tasks further enhances the effectiveness of the framework. Components such as the NoteAgent and QuizAgent automate the generation of structured notes and assessment content, reducing the manual effort required from educators. This not only improves operational efficiency but also allows instructors to focus more on teaching and student interaction rather than repetitive content creation tasks.

Overall, the combined impact of interactive engagement, context-aware retrieval, low-latency response generation, and task automation positions the proposed framework as a significant improvement over existing LMS and generic AI-based educational systems.

A. Tables

TABLE I COMPARISON OF PROPOSED SYSTEM WITH EXISTING SYSTEMS

Feature	Traditional LMS	Generic AI Systems	Proposed System
Real-time interaction	No	Yes	Yes
Curriculum alignment	High	Low	High
Hallucination risk	None	High	Low
Automation capability	Low	Medium	High
Personalizatio n	Low	Medium	High

IV. CONCLUSION

The proposed framework effectively transforms traditional static learning environments into dynamic, interactive educational hubs. Unlike conventional systems that primarily focus on content storage and distribution, the framework introduces an intelligent layer that actively participates in the learning process. Through a conversational interface and real-time interaction capabilities, students are able to engage with the system continuously, enabling a more immersive and responsive learning experience.

This transformation is driven by the integration of Agentic AI with a Retrieval-Augmented Generation (RAG) pipeline. The use of a multi-agent architecture allows the system to handle diverse academic tasks through specialized components, ensuring that each function—such as doubt resolution, note generation, and quiz creation—is executed with greater precision and efficiency. This modular approach not only improves system performance but also enables scalability and extensibility for future enhancements.

The RAG pipeline further strengthens the system by ensuring that all generated responses are grounded in verified academic content. By retrieving relevant information from a FAISS-based vector database and incorporating it into the response generation process, the system maintains alignment with curriculum-specific materials. This significantly reduces the risk of hallucinations and enhances the reliability of the outputs, making the system suitable for academic use where accuracy is critical.

In addition to improving student interaction and response quality, the framework also addresses the operational challenges faced by educators. By automating repetitive tasks such as content generation, summarization, and assessment creation, the system reduces administrative workload. This allows educators to dedicate more time to core teaching activities, including mentoring, conceptual explanation, and personalized guidance.

Overall, the framework establishes a balanced ecosystem where intelligent automation supports both learners and educators. By combining interactive engagement, context-aware intelligence, and task automation, it bridges the gap between traditional LMS platforms and next-generation smart class-room systems.

REFERENCES

[1] G. Kostopoulos et al., "Agentic AI in Education: State of the Art and Future Directions," Journal of Artificial Intelligence in Education, in press, 2025.

[2] AutoFeedback Team, "Using Generative AI and Multi-Agents to Provide Automatic Feedback," arXiv preprint, Jan. 2024.

[3] Z. Li et al., "Retrieval-Augmented Generation for Edu-cational Applications," Computers & Education, vol. 205, pp. 112–128, Feb. 2025.

[4] P. Sikarwar et al., "Cloud Computing in Education: A Revolution in Learning Management Systems," Innovare Journal of Education, vol. 13, no. 1, pp. 22–29, Jan. 2025.

[5] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural In-formation Processing Systems, vol. 33, pp. 9459–9474, Dec. 2020.

[6] E. Kasneci et al., "ChatGPT for good? On opportuni-ties and challenges of large language models for education," Learning and Individual Differences, vol. 103, p. 102274, Apr. 2023.

[7] Z. Xi et al., "The Rise and Potential of Large Language Model Based Agents: A Survey," arXiv preprint arXiv:2309.07864, Sept. 2023.

[8] A. Qazi et al., "Cloud computing for higher education institutes: architecture, strategy and recommendations for effective adaptation," Computers in Human Behavior, vol. 111, p. 106431, Oct. 2020.

[9] L. Wang et al., "A Survey on Large Language Model Based Autonomous Agents," Frontiers of Computer Science, vol. 18, pp. 1–26, Jan. 2024.