

# An Adaptable Client-Server Architecture for Generating Educational Content using Large Language Models

Raymond D. Morland, *Member, IEEE*, and Fuhua Lin, *Senior Member, IEEE*

**Abstract**—Recent advancements in generative artificial intelligence (GenAI), particularly large language models (LLMs), have transformed the landscape of AI-driven educational applications. In this paper, we report on the design and use of a general and adaptable client-server web application architecture that harnesses LLMs for automated educational content generation. This architecture seamlessly integrates modern web technologies with AI-driven content creation workflows, enabling instructors to generate instructional materials and assessment items efficiently. The system leverages retrieval-augmented generation (RAG) to incorporate relevant course materials, ensuring that generated content aligns with predefined learning objectives and pedagogical frameworks. Additionally, prompt engineering techniques are employed, leveraging structured course modeling, and human-AI interaction in optimizing the quality and usability of AI-generated content. To evaluate the effectiveness of this architecture, we discuss the outcomes of multiple research studies that implement this framework in a research setting. These studies examine various use cases, AI integration strategies, and iterative improvements in content generation, highlighting both the potential and challenges of LLM-driven educational applications. Furthermore, the application of this architecture to real-world educational settings is discussed. By providing a scalable, adaptable, and research-driven approach, this work contributes to the ongoing development of AI-enhanced learning environments, paving the way for future innovations in automated content generation, adaptive learning, and AI-assisted instruction.

**Index Terms**— Generative AI, Prompt Engineering, Large Language Models, Human-AI Interaction, Educational Content

## I. INTRODUCTION

The development of high-quality educational content is essential for effective instruction. These materials form the foundation upon which learners build their understanding. However, creating new educational content, such as assessment items and instructional materials, is both time-consuming and labor-intensive. A key challenge is generating personalized content that aligns with individual student needs while also targeting specific learning objectives (LOs), concepts, or skills within a course. Personalized learning materials are crucial for adaptive learning systems (ALS) that identify student weaknesses and provide tailored instructional support. However, the development of targeted, personalized content continues to pose a substantial challenge for educators, particularly in contexts where professional development, technological infrastructure, and instructional resources have not evolved to support the demands of personalized learning environments, making effective

implementation difficult [1].

Recent advancements in Generative Artificial Intelligence (GenAI) have opened new possibilities for applying AI in education, especially for automating the creation of educational content [2]. Large Language Models (LLMs), as well as generative image and video models, can produce customized instructional materials based on natural language prompts. Tools such as ChatGPT and Claude enable educators to generate content dynamically. However, current GenAI applications lack seamless integration with Learning Management Systems (LMS), requiring manual effort to transfer generated materials. Additionally, existing LLM tools do not have access to structured course models or existing LMS content, limiting their ability to generate contextually relevant materials. This gap presents an opportunity to develop GenAI-driven applications that integrate with LMS platforms, leveraging course models and educational data to automate content generation.

Beyond integration, purpose-built systems can incorporate advanced AI techniques such as Retrieval-Augmented Generation (RAG), fine-tuning, and specialized prompting strategies to enhance content quality and personalization. By embedding these techniques into a well-structured application architecture, researchers and developers can refine AI-driven educational content generation, allowing instructors to focus on curriculum development rather than the complexities of AI system implementation.

This paper presents a system architecture for a client-server application designed to generate educational materials using LLMs and advanced prompting techniques. The architecture has been implemented and evaluated in multiple research studies to evaluate different content generation strategies and human-AI interaction (HAI) models. We provide a detailed description of the system design, its implementation, and the outcomes of these studies.

## II. LITERATURE REVIEW

### A. Large Language Models in Education

AI has played a significant role in educational research for decades [3]. The field of AI in Education (AIED) has explored various computational approaches to enhance learning, including algorithms for knowledge tracing (KT) [4] and educational data mining [5]. More recently, the rapid advancement of LLMs has sparked a surge of interest in AIED due to their ability to generate high-quality content through natural language prompts. Researchers and educators are actively investigating the implications of LLMs across educational levels, from primary

Received May 22, 2025, Accepted June 12, 2025, Publish online June 19, 2025."

Raymond Morland is an MSc student in Information Systems and a research assistant with the Intelligent Educational Systems research group at Athabasca University.

Fuhua Lin is a Full Professor of Computing and Information Systems at Athabasca University, Canada (oscarl@athabascau.ca).

school to higher education and lifelong learning [6-11]. A prominent area of research has been the development of chatbot-based tutoring systems, which leverage conversational AI to provide interactive, adaptive learning experiences like other LLM-powered tools [11, 12]. LLMs have also been explored for automated student performance evaluation, including grading and feedback generation [13-15]. Additionally, LLMs have been applied to the automatic generation of assessment items, such as multiple-choice questions (MCQs), in multiple learning domains, enabling scalable and efficient test creation [16-18].

### B. Enhancing Generated Content

Since state-of-the-art (SOTA) LLMs are trained on vast and diverse datasets, their outputs often reflect patterns, generalizations, and statistical associations derived from the content they have been exposed to during training. As a result, improving the precision, relevance, and pedagogical quality of LLM-generated educational materials requires additional techniques. Several strategies have been explored to refine LLM outputs, including prompt engineering, RAG, and model fine-tuning.

Prompt engineering involves carefully designing the input text given to an LLM to optimize the quality and specificity of the generated response [19]. By structuring prompts with explicit instructions, constraints, or formatting guidelines, the output can be aligned more closely with educational objectives. This includes using structured templates, chaining prompts for multi-step reasoning, or leveraging techniques like zero-shot and few-shot learning, where the model is given examples to guide its response generation [14, 20].

RAG enhances LLM responses by dynamically injecting relevant external information into the prompt [21, 22]. In education, this technique involves retrieving contextual data from a structured knowledge base, such as textbooks, lecture notes, or an instructor's curated materials [23]. To achieve this, educational content is first embedded into vector representations, enabling semantic search using similarity measures to find and incorporate the most relevant information into the prompt before passing it to the LLM. By grounding the model in accurate, domain-specific content, RAG reduces hallucinations and improves factual consistency.

Fine-tuning an LLM involves retraining it on a domain-specific dataset to enhance its performance for specialized educational applications. Techniques such as supervised fine-tuning and reinforcement learning with human feedback (RLHF) [24] enable the model to adjust its weight based on high-quality educational data and expert-curated feedback. This process improves the model's ability to generate accurate, pedagogically sound content that aligns with LOs and instructional strategies. Additionally, other reinforcement learning (RL) techniques can further optimize content generation. For example, Scarlatos et al. [15] applied direct preference optimization (DPO) to improve the alignment and accuracy of generated feedback for math assessments.

Beyond general techniques, education-focused strategies can further refine LLM-generated content. For instance, integrating Bloom's Taxonomy into prompt engineering allows for structuring content by cognitive complexity levels [25]. Embedding educational frameworks into content generation pipelines enables LLMs to produce scaffolded educational materials that foster deeper learning and engagement. By leveraging these techniques, LLMs can be better adapted to the needs of educators and students, improving the effectiveness, relevance, and

personalization of AI-generated educational content.

### C. Modeling Educational Domains

A fundamental aspect of AI-driven educational content generation is how the content is modeled. To ensure that generated materials align with existing course structures and instructional pedagogies, it is essential to incorporate well-defined educational domain models into the generation process. This structured representation of educational content allows GenAI to produce materials that are cohesive, pedagogically sound, and relevant to specific curricula.

One widely used approach in educational modeling is curriculum modeling [26], which organizes course content into structured hierarchies. Courses can be represented as a layered framework consisting of overarching subject areas, thematic modules, granular knowledge components (KCs), and specific LOs. These relationships provide a structured foundation for generating targeted educational materials. By referencing specific KCs, AI systems can generate practice questions, explanations, or summaries that align with predefined LOs and student knowledge levels.

Another powerful technique for modeling educational content involves knowledge graphs and ontologies, which explicitly define relationships between concepts, skills, and prerequisites in each subject area [27]. These structured representations enable AI models to retrieve contextually relevant content, generate logically structured instructional materials, and ensure that students receive content that builds upon prerequisite knowledge. For example, in STEM education, a mathematics knowledge graph could outline connections between algebraic concepts, allowing LLMs to generate adaptive learning pathways based on student proficiency [9, 28].

A robust domain model also enables adaptive learning, where generated content is dynamically adjusted based on student performance and engagement. This approach ensures that AI-generated content is not only domain-aware but also personalized to individual learning trajectories.

## III. SYSTEM IMPLEMENTATION

While existing research systems showed that GenAI is effective for generating educational content, many existing solutions lack comprehensive integration with key educational components such as structured course models, student learning profiles, and external learning resources. Furthermore, while these systems demonstrate promising capabilities, their full architectures are often not fully described, making it difficult to assess their scalability, adaptability, and effectiveness in real-world educational settings.

Our implementation addresses these gaps by providing a fully integrated system capable of generating educational content while seamlessly incorporating course structures, student models, and external learning materials. By leveraging structured course models, our system ensures that generated content aligns with predefined LOs and instructional strategies. Additionally, the integration of student models enables personalized content generation, adapting to individual learning needs and proficiency levels. Furthermore, by utilizing RAG with external learning resources and instructional files, our approach enhances the

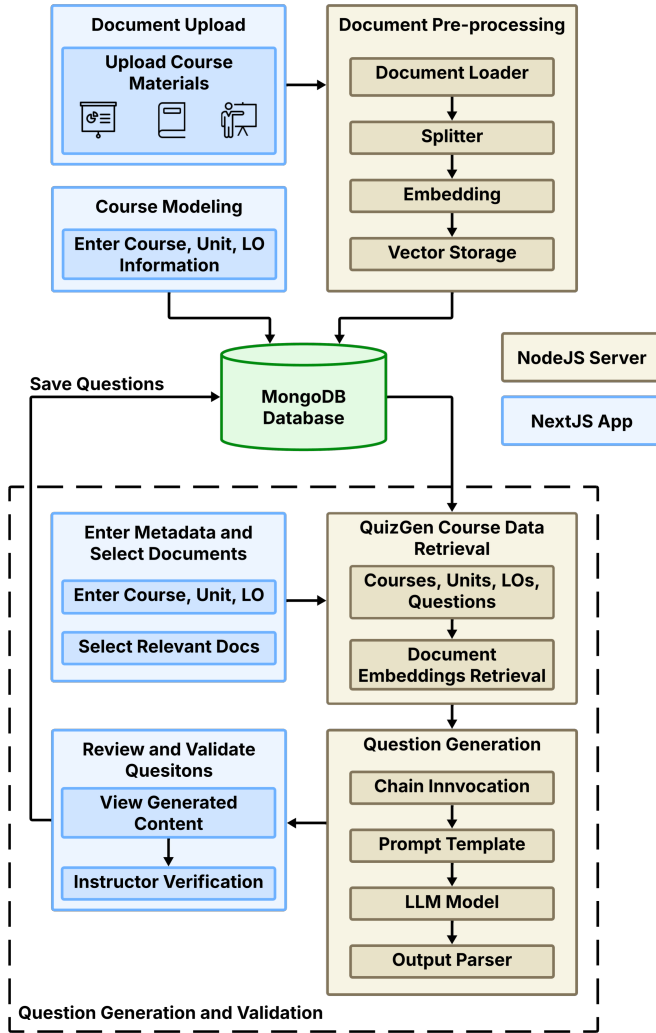


Figure 1: The architectural diagram illustrates the question generation system's client-server framework, where a Next.js front-end empowers instructors with intuitive controls for question generation. On the server side, a NodeJS application manages database interactions and executes question generation algorithms. The system utilizes a MongoDB database both as a traditional document store and as an embedded vector store, providing flexibility in content storage and retrieval

relevance and contextual accuracy of AI-generated materials.

#### A. System Architecture

Our educational content generation system follows a client-server architecture, designed for seamless integration with course models, student models, and AI-driven content generation workflows (see Figure 1). The front-end is developed using Next.js<sup>1</sup>, a React-based framework that enables a dynamic and responsive user interface, ensuring an intuitive experience for instructors and administrators. The server-side is built on Node.js<sup>2</sup>, leveraging its efficient asynchronous processing capabilities to manage API requests, content generation workflows, and database operations.

A MongoDB<sup>3</sup> database serves as the system's primary data store, organizing and managing structured course information,

including courses, units, LOs, and generated assessment questions. This structured storage facilitates the retrieval of relevant educational content for RAG, ensuring that AI-generated materials align with existing curricular frameworks. The system's architecture allows for scalable and efficient handling of educational data while supporting real-time updates and integrations with AI-driven content generation models.

#### B. LLM Integration

At the core of our educational content generation system is the integration of LLMs, which drives the creation of instructional materials, assessment questions, and personalized learning content. To enable seamless interaction with LLMs, we leverage LangChain<sup>4</sup>, a powerful framework that facilitates the orchestration of AI-driven workflows. LangChain provides a unified interface for integrating multiple LLM providers, including OpenAI, Anthropic, Mistral, Vertex AI, and AWS, allowing for easy adaptability across different models. This flexibility enables developers to experiment with and switch between LLMs based on performance, cost, or preference, making the architecture highly adaptable for various educational applications. Our implementation primarily utilizes OpenAI's GPT models. By integrating LangChain with OpenAI's LLMs, we ensure that the system produces coherent, pedagogically sound, and dynamically adaptable educational content, supporting a wide range of instructional needs.

#### C. Retrieval-Augmented Generation

To enhance content generation, we implement RAG, allowing instructors to upload learning resources such as textbooks, lecture notes, or supplementary materials directly through the instructor interface. These documents are processed, embedded, and indexed, enabling the system to retrieve relevant context and improve the quality of AI-generated content.

Upon upload, documents undergo a processing pipeline that prepares them for efficient retrieval (illustrated in Figure 2).

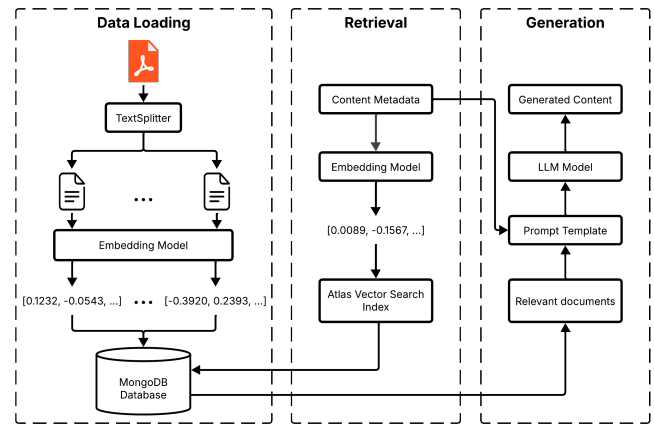


Figure 2: Retrieval augmented generation process for educational content generation.

First, the documents are segmented using LangChain TextSplitters, ensuring that the content is broken down into manageable chunks. Each segment is then embedded using the OpenAI embedding model, transforming text into high-dimensional vector representations that capture semantic meaning. These vector embeddings are stored and indexed using

<sup>1</sup><https://nextjs.org>

<sup>2</sup><https://nodejs.org>

<sup>3</sup><https://mongodb.com>

<sup>4</sup><https://langchain.com>

MongoDB Atlas Vector Search<sup>5</sup>, allowing for efficient similarity-based retrieval. The original file is securely stored in an AWS S3 bucket, ensuring scalable and reliable access to learning materials.

#### D. Prompt Engineering

Prompt engineering plays a crucial role in our educational content generation pipeline, ensuring that AI-generated materials are well-structured, pedagogically sound, and aligned with LOs. This process is implemented within the server application, where prompts are dynamically constructed and formatted using LangChain's prompt templates. These templates enable the injection and structuring of contextual information into the LLM prompt, ensuring that generated content adheres to educational best practices.

The system incorporates course-specific metadata into the prompt, including course hierarchy, Question Validation Criteria (QVC), feedback requirements, and instructional methodology. Depending on the type of content being generated, such as MCQs, open-ended assessments, or instructional explanations, the appropriate prompt template is selected to ensure consistency and clarity in the output.

To optimize prompt performance, the system supports zero-shot, one-shot, and multi-shot prompting techniques, allowing instructors and researchers to experiment with different input structures and evaluate their impact on content quality. Through systematic testing and refinement, prompts can be adjusted to improve response accuracy, increase alignment with pedagogical principles, and reduce LLM hallucinations.

A key advantage of this approach is that expertly defined prompt templates encapsulate complex learning science principles, enabling instructors to focus solely on defining LOs without needing expertise in AI-driven instructional design. By embedding educational theories, assessment frameworks, and domain-specific constraints directly into the prompts, the system ensures that generated content is instructionally effective, contextually relevant, and tailored to student needs.

#### E. Human-AI Interaction

The user interface of our educational content generation

Figure 3: Interface for selecting content metadata to generate content in the QuizGen application.

architecture is designed to provide a seamless and intuitive experience for instructors, balancing the power of AI-driven automation with human control and validation. Built with Next.js, the front-end leverages a modular component-based architecture, enabling the flexible and scalable development of interactive interfaces. The application interface enables instructors to create courses, where each course can contain multiple units, and each unit is associated with specific LOs. This structured course information is stored in MongoDB, forming the backbone of the content generation system. Once courses are set up, instructors can upload relevant documents, which are automatically processed and indexed for use in RAG.

In the content generation workflow, instructors begin by selecting structured metadata such as content type, course, unit, LO, and configuration parameters like the number of questions and answer choices (see Fig. 3). This metadata triggers the server-side generation process, which retrieves relevant course information, uses RAG to fetch document embeddings, and constructs an LLM prompt based on a predefined template. The LLM-generated content is formatted as JSON and returned to the front end, where instructors review, accept, or edit the output through a validation interface. This HAI approach ensures instructors maintain control over final content, enabling iterative refinement and improving pedagogical quality, while the intuitive interface supports ease of use, trust, and effective AI integration into educational workflows.

### IV. RESEARCH USE CASES

#### A. MCQs with QVCs: QuizGen

To evaluate the effectiveness of applying QVC to AI-generated MCQs, we developed QuizGen, an application built on our educational content generation architecture. QuizGen serves as a testbed for experimenting with different prompt engineering techniques and assessing the impact of structured question validation frameworks on AI-generated assessment items. By incorporating QVCs into the prompt engineering process, the system aims to improve the quality, clarity, and pedagogical alignment of generated MCQs.

In this implementation, we modified the base prompts to integrate expertly defined QVCs (see Fig. 4), ensuring that generated questions adhered to specific standards of accuracy, cognitive complexity, and instructional effectiveness. Additionally, we explored multiple prompting strategies to determine the most effective approach for generating high-quality assessment items. These strategies were designed to optimize question quality while maintaining efficiency in content generation.

The first prompting strategy involved a zero-shot prompt approach, where all QVCs, course metadata, and contextual information were embedded within a single query to the LLM. This method aims to streamline the generation process by providing comprehensive input to the model in one step. However, a potential limitation of this approach was the difficulty in enforcing all QVCs simultaneously, leading to inconsistencies in quality question.

To improve upon this, we implemented a two-round critic-based prompting strategy. In this approach, the first round utilized course metadata and contextual information to generate an initial question, while the second round functioned as a self-critic, instructing the LLM to evaluate and refine the question based on

<sup>5</sup><https://www.mongodb.com/products/platform/atlas-vector-search>

```

You are the instructor of a course on {course}. You are creating
a formative assessment for the unit {unit} to test your
students' understanding of the learning objective
{learningObjective}

Use the following context to aid in generating the content:
{context}

Provide {numOfQuestions} multiple-choice questions. Each
question should have {numOfOptions} options and only a single
correct answer.

The question and its options should align with the following
criteria. If they do not, adjust them.

1. Relevance and Alignment: The question aligns with the
learning outcome and its cognitive level
2. Clarity and Comprehension: The question is clear, concise,
complete, and easy to understand
3. Fairness and Bias: The question is free of race, ethnic,
and sex bias
4. Plausibility of Distractors: The options include plausible
and homogeneous distractors with parallel length
5. Readability of Options: The options are in logical,
alphabetical, chronological, or numerical order
6. Test-wise and Logic Cues: The options are similar in length,
use the same expression style, and have consistent grammar
7. Independency of Options: The options are independent and
free of overlapping responses and extreme modifiers

You have already generated the following questions:
{previous_questions}

DO NOT repeat questions that you have already created.

The output must have the following format:
{parser.get_format_instructions()}

```

Figure 4: Prompt template for generating Multiple-Choice Questions (MCQs) with Question Validation Criteria (QVC), used in the QuizGen application.

the predefined QVCs. This method introduced an additional layer of validation and enhancement, allowing the model to identify and correct flaws or ambiguities in the initial question.

Building on this iterative approach, we evaluated a multi-round enhancement strategy, where the question underwent multiple refinement iterations. In this method, the first round produced a baseline question, followed by subsequent rounds that iteratively improved the question one validation criterion at a time. Each refinement stage used a new prompt that incorporated feedback from the previous iteration, progressively enhancing the alignment with QVCs, readability, and cognitive complexity.

While preliminary results from this study show no statistically significant differences in the quality of generated questions, subject matter experts rated the overall quality as favorable. These findings suggest that structured prompt engineering, when paired with validation-driven refinement, holds promise for enhancing question quality. However, further research is needed to more precisely define the criteria required to meaningfully improve the quality of AI-generated educational assessments.

#### B. MCQs for Adaptive Formative Assessments: QuizMaster

To evaluate the impact of AI-driven content generation on instructor efficiency in an ALS, we integrated our generation architecture into QuizMaster, an adaptive formative assessment platform designed to evaluate adaptive algorithms [29]. This implementation aims to streamline question creation workflows, allowing instructors to efficiently generate and customize assessment items while ensuring pedagogical alignment with LOs.

In this version of the architecture, multiple types of MCQs were developed, including standard MCQs, fill-in-the-blank questions,

sentence completion (finish-the-sentence), and true/false questions. Each of these question formats required distinct generation instructions, which were expertly crafted and dynamically injected into the prompt whenever a specific MCQ type was selected. This structured approach ensured that the LLM-generated questions followed the appropriate format, met instructional standards, and provided students with clear and meaningful assessments.

Additionally, each MCQ also includes feedback for each answer option, which is automatically generated and aligned with predefined feedback criteria embedded in the prompt. When a student completes an assessment, QuizMaster displays targeted feedback at the end of the test, helping learners understand their mistakes and reinforcing adaptive learning principles. Because the generated questions are immediately available for adaptive assessments, the system enhances scalability and efficiency, ensuring that instructors can rapidly deploy high-quality, personalized assessments without manual question creation.

By automating question generation, formatting, and feedback integration, this implementation of QuizMaster demonstrates how AI-driven tools can significantly improve instructor productivity, reducing the time spent on assessment design while maintaining high-quality learning experiences.

#### C. Iterative Educational Material Generation: EMGen

To explore different HAI interaction techniques in educational content creation, we developed EMGen, an interface designed to evaluate an iterative content generation paradigm. This system allows instructors to engage in a dynamic, multi-step interaction with the LLM, refining and enhancing generated content through a continuous feedback loop. By integrating structured instructor inputs, pedagogical frameworks, and AI-assisted refinements, EMGen enables the systematic evaluation of iterative HAI in content generation.

The iterative process begins with the instructor selecting LO metadata, ensuring that generated content aligns with specific course goals. Next, the instructor selects a content type, which includes tutorials, concept explanations, MCQs, and exercises. Based on the chosen content type, context-aware pedagogical options are presented. For example, if generating questions, the instructor can apply QVCs to improve assessment quality. If generating tutorials or concept explanations, the system offers options based on learning theories, such as the GROW model for guided learning [30] or Feynman's Technique for deep concept understanding [31].

Once the initial content is generated, the instructor enters an iterative refinement phase, where they can review, edit, and provide feedback to the LLM. The AI processes this feedback, refining the content in multiple iterative cycles until the instructor is satisfied. This interactive workflow enables instructors to maintain control over content quality while leveraging AI for efficiency and adaptability. EMGen serves as a testbed for iterative generation paradigms, allowing researchers to evaluate different HAI techniques within the base adaptable LLM generation system architecture.

## V. FUTURE WORK

While preliminary evaluations of QuizGen, EMGen, and QuizMaster have shown promise, future research will focus on real-world testing with instructors and students to assess the effectiveness of HAI in educational content generation. This includes conducting user studies to evaluate usability,

engagement, and pedagogical impact, while refining iterative interaction techniques to support instructor autonomy, student adaptability, and personalized content. Currently driven by prompt engineering and RAG, the system will evolve through fine-tuning with domain-specific data, instructor feedback, and student performance metrics. Key future directions include integrating RLHF, expert feedback loops, and knowledge graphs to improve educational alignment, question quality, and adaptability. The goal is to develop a fully adaptive, research-driven AI system that enhances teaching, learning, and assessment through continuous refinement and empirical validation.

## VI. CONCLUSION

This paper presents a modular, adaptable architecture for AI-driven educational content generation that integrates LLMs with structured course models, RAG, and HAI. Through the implementation of QuizGen, EMGen, and QuizMaster, the study demonstrates how AI can support instructors in efficiently creating assessments, instructional materials, and adaptive learning resources. The research underscores the value of prompt engineering, iterative HAI collaboration, and pedagogical alignment techniques in enhancing content quality. By enabling instructors to iteratively refine AI-generated materials and incorporating QVCs and adaptive frameworks, the system ensures educational relevance and personalization. This architecture offers a scalable, research-driven foundation for AI-assisted education, emphasizing the importance of combining AI automation with instructor expertise to create more effective and accessible learning experiences.

## ACKNOWLEDGMENT

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), Alberta Innovates, and Athabasca University, Canada. Additionally,

## REFERENCES

- [1] A. J. Bingham, J. F. Pane, E. D. Steiner, and L. S. Hamilton, 'Ahead of the Curve: Implementation Challenges in Personalized Learning School Models', *Educational Policy*, vol. 32, no. 3, pp. 454–489, May 2018, doi: 10.1177/0895904816637688.
- [2] M. P.-C. Lin, D. Chang, S. Hall, and G. Jhaji, "Preliminary Systematic Review of Open-Source Large Language Models in Education," in *Generative Intelligence and Intelligent Tutoring Systems*, vol. 14798, A. Sifaleras and F. Lin, Eds., Cham: Springer Nature Switzerland, 2024, pp. 68–77. doi: 10.1007/978-3-031-63028-6\_6.
- [3] P. Brusilovsky, "AI in Education, Learner Control, and Human-AI Collaboration," *Int J Artif Intell Educ*, vol. 34, no. 1, pp. 122–135, Mar. 2024, doi: 10.1007/s40593-023-00356-z.
- [4] G. Abdelrahman, Q. Wang, and B. Nunes, 'Knowledge Tracing: A Survey', *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–37, Nov. 2023, doi: 10.1145/3569576.
- [5] K. R. Koedinger, S. D'Mello, E. A. McLaughlin, Z. A. Pardos, and C. P. Rosé, 'Data mining and education', *WIREs Cognitive Science*, vol. 6, no. 4, pp. 333–353, Jul. 2015, doi: 10.1002/wcs.1350.
- [6] D. Hennekeuser, D. D. Vaziri, D. Golchinfar, D. Schreiber, and G. Stevens, "Enlarged Education – Exploring the Use of Generative AI to Support Lecturing in Higher Education," *Int J Artif Intell Educ*, Aug. 2024, doi: 10.1007/s40593-024-00424-y.
- [7] P. Atchley, H. Pannell, K. Wofford, M. Hopkins, and R. A. Atchley, "Human and AI collaboration in the higher education environment: opportunities and concerns," *Cogn. Research*, vol. 9, no. 1, p. 20, Apr. 2024, doi: 10.1186/s41235-024-00547-9.
- [8] J. Jeon and S. Lee, "Large language models in education: A focus on the complementary relationship between human teachers and ChatGPT," *Educ Inf Technol*, vol. 28, no. 12, pp. 15873–15892, Dec. 2023, doi: 10.1007/s10639-023-11834-1.
- [9] R. Meissner et al., "LLM-generated competence-based e-assessment items for higher education mathematics: methodology and evaluation," *Front. Educ.*, vol. 9, p. 1427502, Oct. 2024, doi: 10.3389/educ.2024.1427502.
- [10] A. Attard and A. Dingli, "EMPOWERING EDUCATORS: LEVERAGING LARGE LANGUAGE MODELS TO STREAMLINE CONTENT CREATION IN EDUCATION," Seville, Spain, Nov. 2024, pp. 1312–1321. doi: 10.21125/iceri.2024.0400.
- [11] D. Yigci, M. Eryilmaz, A. K. Yetisen, S. Tasoglu, and A. Ozcan, "Large Language Model-Based Chatbots in Higher Education," *Advanced Intelligent Systems*, p. 2400429, Aug. 2024, doi: 10.1002/aisy.202400429.
- [12] E. Chen, J.-E. Lee, J. Lin, and K. Koedinger, "GPTutor: Great Personalized Tutor with Large Language Models for Personalized Learning Content Generation," in *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, Atlanta GA USA: ACM, Jul. 2024, pp. 539–541. doi: 10.1145/3657604.3664718.
- [13] I. Estévez-Ayres, P. Callejo, M. Á. Hombrados-Herrera, C. Alario-Hoyos, and C. Delgado Kloos, "Evaluation of LLM Tools for Feedback Generation in a Course on Concurrent Programming," *Int J Artif Intell Educ*, May 2024, doi: 10.1007/s40593-024-00406-0.
- [14] D. Kulshreshtha, M. Shayan, R. Belfer, S. Reddy, I. V. Serban, and E. Kochmar, "Few-shot Question Generation for Personalized Feedback in Intelligent Tutoring Systems," Jun. 08, 2022, arXiv: arXiv:2206.04187. doi: 10.48550/arXiv.2206.04187.
- [15] A. Scarlatos, D. Smith, S. Woodhead, and A. Lan, "Improving the Validity of Automatically Generated Feedback via Reinforcement Learning," 2024, doi: 10.48550/ARXIV.2403.01304.
- [16] M. O. Omojekunola and E. Y. Kardanova, "Automatic generation of physics items with Large Language Models (LLMs)," *REiD*, vol. 10, no. 2, pp. 168–185, Oct. 2024, doi: 10.21831/reid.v10i2.76864.
- [17] A. Olney, "Generating multiple choice questions from a textbook: LLMs match human performance on most metrics," *Workshop on Empowering Education with LLMs - the Next-Gen Interface and Content Generation at the AIED'23 Conference*, 2023.
- [18] A. R. Gilal, A. Waqas, B. A. Talpur, R. A. Abro, J. Jaafar, and Z. H. Amur, "Question Guru: An Automated Multiple-Choice Question Generation System," in *Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems*, vol. 573, M. A. Al-Sharafi, M. Al-Emran, M. N. Al-Kabi, and K. Shaalan, Eds., Cham: Springer International Publishing, 2023, pp. 501–514. doi: 10.1007/978-3-031-20429-6\_46.
- [19] U. Lee et al., "Few-shot is enough: exploring ChatGPT prompt engineering method for automatic question generation in english education," *Educ Inf Technol*, vol. 29, no. 9, pp. 11483–11515, Jun. 2024, doi: 10.1007/s10639-023-12249-8.
- [20] C. Cohn, N. Hutchins, T. Le, and G. Biswas, "A Chain-of-Thought Prompting Approach with LLMs for Evaluating Students' Formative Assessment Responses in Science," Mar. 21, 2024, arXiv: arXiv:2403.14565. doi: 10.48550/arXiv.2403.14565.
- [21] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NeurIPS '20)*, Red Hook, NY, USA: Curran Associates Inc., 2020, pp. 9459–9474.
- [22] P. Zhao et al., "Retrieval-Augmented Generation for AI-Generated Content: A Survey," Jun. 21, 2024, arXiv: arXiv:2402.19473. doi: 10.48550/arXiv.2402.19473.
- [23] S. Jacobs and S. Jaschke, "Leveraging Lecture Content for Improved Feedback: Explorations with GPT-4 and Retrieval Augmented Generation," in *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*, Würzburg, Germany: IEEE, Jul. 2024, pp. 1–5. doi: 10.1109/CSEET62301.2024.10663001.
- [24] S. Lamsiyah, A. El Mahdaouy, A. Nourbakhsh, and C. Schommer, "Fine-Tuning a Large Language Model with Reinforcement Learning for Educational Question Generation," in *Artificial Intelligence in Education*,

- vol. 14829, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds., Cham: Springer Nature Switzerland, 2024, pp. 424–438. doi: 10.1007/978-3-031-64302-6\_30.
- [25] N. Scaria, S. Dharani Chenna, and D. Subramani, “Automated Educational Question Generation at Different Bloom’s Skill Levels Using Large Language Models: Strategies and Evaluation,” in *Artificial Intelligence in Education*, vol. 14830, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds., Cham: Springer Nature Switzerland, 2024, pp. 165–179. doi: 10.1007/978-3-031-64299-9\_12.
- [26] F. Lin and R. Morland, “Curriculum Modeling for Adaptive Learning”, *Proceedings of International Conference on Human-Computer Interaction*, 2025, Springer.
- [27] G. Jhajj, X. Zhang, J. R. Gustafson, F. Lin, and M. P.-C. Lin, “Educational Knowledge Graph Creation and Augmentation via LLMs,” in *Generative Intelligence and Intelligent Tutoring Systems*, vol. 14799, A. Sifaleras and F. Lin, Eds., Cham: Springer Nature Switzerland, 2024, pp. 292–304. doi: 10.1007/978-3-031-63031-6\_25.
- [28] F. Zhang et al., “Math-LLMs: AI Cyberinfrastructure with Pre-trained Transformers for Math Education,” *Int J Artif Intell Educ*, Jul. 2024, doi: 10.1007/s40593-024-00416-y.
- [29] F. Lin, R. Morland, and H. Yan, “QuizMaster: An Adaptive Formative Assessment System,” in *Generative Intelligence and Intelligent Tutoring Systems*, vol. 14798, A. Sifaleras and F. Lin, Eds., Cham: Springer Nature Switzerland, 2024, pp. 55–67. doi: 10.1007/978-3-031-63028-6\_5.
- [30] Panchal, S., & Riddell, P. (2020). The GROWS model: Extending the GROW coaching model to support behavioural change. *The Coaching Psychologist*, 16(2), 12–25.
- [31] E. P. Reyes, et al., ‘Feynman Technique as a Heutagogical Learning Strategy for Independent and Remote Learning’, *RMRIJ*, vol. 9, no. 2, pp. 1–13, Dec. 2021, doi: 10.32871/rmrj2109.02.06.





**Raymond Morland** is an MSc student in Information Systems and a research assistant with the Intelligent Educational Systems (IES) research group at Athabasca University, Canada. He holds a Bachelor of Science in Physics from the University of Saskatchewan and a Bachelor of Science in Computing and Information Systems from Athabasca University, Canada.



**Fuhua Lin** (Senior Member, IEEE) Dr. Fuhua (Oscar) Lin is a Full Professor of Computing and Information Systems at Athabasca University, Canada. Dr. Lin obtained his PhD from Hong Kong University of Science and Technology in 1998. He has more than 160 publications. Dr Lin's research interests include AI in education, adaptive learning, intelligent systems, online learning, virtual reality, modelling and

simulation, multi-agent systems, machine learning, and computer based sequential decision making.