EduQuest: An AI-Assisted Academic Question Generator

R. Raja Subramanian[1], Shamraj Sheik[2]
Department of Computer Science and Engineering,
Kalasalingam Academy of Research and Education, Tamil Nadu, India
[1]rajasubramanian.r@klu.ac.in, [2]itzshamrajsheik@gmail.com

***Abstract*—EduQuest is an AI-powered platform that makes the creation of standardized question papers faster, more consistent, and pedagogically aligned for educators. EduQuest uses Google's Gemini generative AI model, along with structured prompt engineering, to generate midterm and end-semester examinations mapped onto Bloom's taxonomy and tailored to institutional requirements. It provides a system for flexible templates in disciplines such as engineering design, problem-solving, and business studies, which allows educators to customize mark distribution, difficulty levels, and learning outcome alignment. The platform will include institutional branding, PDF exports, and the ability to incorporate additional context from uploaded PDFs. Edu Quest is built on a Flask backend, with a user-friendly Bootstrap interface; it adopts a caching mechanism to reduce API calls and enhance performance. Testing done with 10 faculty members and 300 students demonstrates that the reduction in time to prepare question papers is between 60 and 70%, the variety of questions has significantly improved, and the cognitive level coverage is better. In addition, to prevent AI-generated errors, the generated content is subjected to structured prompt control through manual review. It's worth noting that Edu Quest has not so far included automatic prompting or model fine-tuning as part of improving its efficiency. Overall, Edu Quest presents a practical and scalable solution for generating quality assessment points relevant to modern-day educational requirements.**

***Keywords*— AI-assisted education; assessment automation; Bloom's taxonomy; generative AI; question paper generation**

***ICTIEE Track*—Assessment of Effective Teaching**

ICTIEE Sub-Track—Assessment for Learning: Empowering Students through Effective Assessment Practices

I. INTRODUCTION

THE art of crafting a good question paper is a basic yet challenging task of teaching, and one that has tried the patience and ingenuity of teachers for years. For decades, teachers have struggled to painstakingly design assessments that cover extensive syllabi, cater to various learning objectives, and balance difficulty levels amidst widely varying institutional guidelines. This manual process often spans hours or even days and results in inconsistencies in rigor and uneven topic coverage, besides a limited use of higher-order thinking skills as outlined in Bloom's taxonomy. These not only have adverse implications on the fairness of assessments but also prove to be a huge workload for educators, especially in high-pressure academic environments where time and support are particularly limited.With the emergence of artificial intelligence, especially generative AI, this process can now be reshaped in as-yet unimaginable ways. Such tools would go on to generate numerous, contextually relevant questions at speed-challenging rates, provided the system is fed with clear, structured guidance. However, most of the existing AI-based solutions do not deeply integrate pedagogical frameworks like Bloom's taxonomy, are institutionally uncustomized, and grant little control over cognitive levels or syllabus alignment. Their utility is, therefore, quite limited for formal educational settings.

EduQuest has been designed to respond to these challenges by combining generative AI with structured prompt engineering in the creation of pedagogically aligned midterm and end-semester question papers. The platform allows educators to customize mark distribution, difficulty levels, learning outcomes, and syllabus topics while providing support for institutional branding and optional PDF-based context extraction from syllabi, textbooks, or case studies. Built on a Flask backend with a user-friendly Bootstrap interface, EduQuest uses a caching mechanism to reduce API calls and improve performance, making it practical for institutions with limited technical resources.

Whereas EduQuest employs carefully designed manual prompts in order to maintain full control of cognitive levels and syllabus alignment, automatic prompt refinement or adaptive prompting algorithms are not currently part of the system. This is a limitation that is recognized and points towards a path for future development. The literature on fine-tuned educational models and automatic prompting systems shows that they do require substantial amounts of data and vast computational resources; EduQuest makes accessibility and interpretability key principles.

This paper describes the design, implementation, and evaluation of EduQuest through a semester-long deployment involving 10 faculty members and 300 students across the mechanical engineering, computer science, and business management programs. The platform's effectiveness in automatically generating questions is studied, along with its technical performance and the consequences on educator workload and student engagement. The results demonstrate EduQuest's potential for streamlining assessment creation while maintaining educational rigor.

R. Raja Subramanian
Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education
rajasubramanian.r@klu.ac.in

II. LITERATURE REVIEW

Artificial intelligence is increasingly influencing the educational landscape, offering tools that automate grading, personalize learning, and generate instructional content with remarkable speed. Generative AI, in particular, has shown significant promise in producing quizzes, exercises, and learning materials. However, its application in formal assessments—such as mid-term and end-semester question papers—remains underexplored, which motivated the creation of EduQuest.

Earlier studies have examined AI-driven question generation from various perspectives. Gao et al. (2019) demonstrated the effectiveness of neural network models in generating context-aware questions from reading passages, highlighting the potential of AI for content-specific assessment. Kurdi et al. (2020) provided a comprehensive review of automatic question generation techniques, noting that while NLP-based systems are capable of producing grammatically correct and semantically meaningful questions, much of the existing work focuses on informal or formative assessments rather than structured academic examinations. Importantly, these systems generally do not align questions with pedagogical frameworks such as Bloom's taxonomy, which remains a foundational model for designing assessments that span cognitive levels from remembering to creating.

While Bloom's taxonomy has been central to assessment design for decades, only a few recent works attempt to integrate AI with cognitive-level mapping. Even those efforts often lack the flexibility to support institutional customization, learning outcome mapping, or discipline-specific requirements. Commercial AI-assisted platforms such as Quizlet, ClassMarker, and various GPT-4–based educational tools offer automated question generation, but they typically operate without deep integration of syllabus coverage, examination patterns, or institutional formatting constraints required in higher education. Consequently, they remain insufficient for generating standardized question papers that adhere to academic guidelines.

Recent advancements in large language models have introduced techniques such as automatic prompt refinement, fine-tuned domain-specific models, and reinforcement learning–based question generation. Although these approaches improve accuracy and adaptability, they often require large datasets, extensive training pipelines, and significant computational resources. For many educational institutions—particularly those with limited infrastructure—such methods are not practical. EduQuest intentionally avoids heavy fine-tuning and instead employs structured, interpretable prompt engineering to maintain educator control, transparency, and ease of deployment.

Technical literature further highlights the importance of system performance in AI applications. Caching mechanisms, as explored by Zhang and VanLehn (2016), have been shown to reduce computational load and response times, enabling scalable and cost-effective implementations—an approach integrated into EduQuest. Cloud-based educational systems research also emphasizes the need for reliable, responsive, and user-friendly platforms, reinforcing the design choices behind EduQuest's architecture.

Overall, the existing literature reveals gaps in AI-assisted question generation related to cognitive-level alignment, outcome mapping, institutional formatting, and content contextualization. EduQuest is designed to address these gaps by combining generative AI with structured prompts, optional PDF-based context extraction, and customizable templates to support disciplined, pedagogically grounded assessment practices in engineering and related domains.

III. METHODOLOGY

EduQuest was designed to combine the efficiency of AI with the structured rigor required in standardized academic assessments. The methodology involves designing system architecture, integrating AI, specifying workflows, optimizing performance, and evaluation procedures that ensure technical effectiveness and pedagogical soundness.

A. System Architecture

EduQuest architecture is organized into three central components: the User Experience Layer, the Processing Layer, and Storage Components.

The User Experience Layer provides an interface where the educator sets the examination parameters in terms of mark distribution, difficulty levels, Bloom's taxonomy levels, learning outcomes, and syllabus coverage. Additionally, the system allows the user to upload supplementary PDFs at this level, such as syllabi, textbooks, and case studies that will provide a guide towards question generation. A preview panel enables users to view the generated question paper before export.

The Processing Layer manages system operations via a Flask-based backend. Uploaded PDFs are parsed for the extraction of relevant text that serves as contextual input to question generation. The integration of Google's Gemini model is done by structured prompts that maintain alignment with Bloom's taxonomy, institutional formatting, and user-defined parameters. The Template Engine formats generated questions into a standardized exam layout, including institutional branding.

The storage components include temporary file storage for uploaded PDFs and a caching mechanism for frequently used prompts and AI responses. This reduces redundant API calls, further enhancing the response time while lowering the cost of operation.

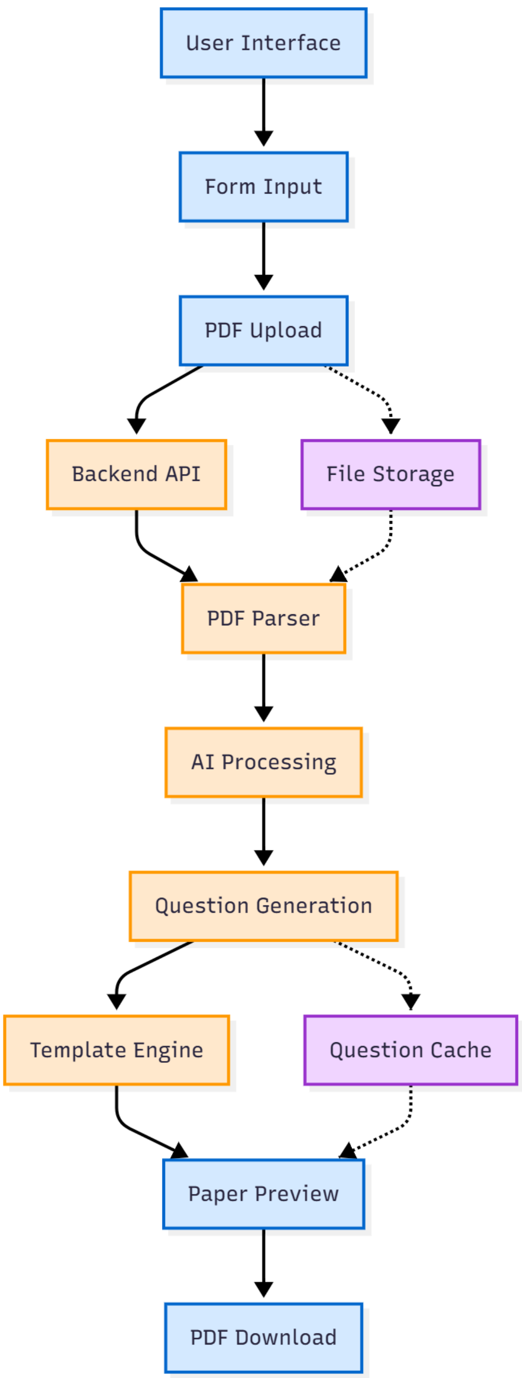


Fig. 1. Neural Architecture of EduQuest.

B. *AI Integration and Prompt Engineering*

EduQuest relies on structured prompt engineering in order to guide AI-generated content. Prompts are dynamically constructed based on exam type, cognitive level, learning outcomes, syllabus focus, and PDF-derived context. Structured prompts ensure consistency, pedagogical alignment, and relevance to discipline-specific requirements.

While the system realizes reliable question generation by using manual prompt templates, it currently does not employ automatic prompt refinement or adaptive prompting algorithms. These techniques investigated in recent AI literature usually involve fine-tuning large datasets and significant computational resources beyond the scope of operation for EduQuest. The platform focuses instead on educator-controlled parameters and transparent prompt logic.

Limitations of current AI models include variability in handling long context and occasional sensitivity to ambiguous prompts; these have been put into consideration in designing the multi-step validation workflow that produces questions with accuracy and consistency.

C. *Workflow*

The EduQuest workflow comprises five major phases:

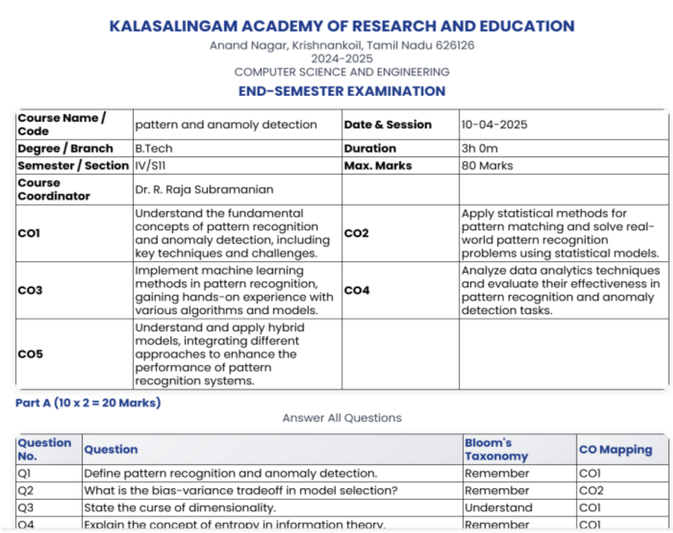
Setup: Instructors can define exam settings through a simple form-based interface in which they specify course type, difficulty levels, Bloom's taxonomy mapping, and learning outcomes.

Context Integration: Users can upload PDFs that contain relevant instructional content. While this enhances contextual integration, the system does not mandate these files in order to prepare a valid question paper. Hence, it gives more flexibility for institutions that do not provide extensive documentation.

Prompt Generation: After selecting the parameters and any optional contextual data, the system generates structured prompts. The prompt explicitly encodes Bloom's levels, marks distribution, and syllabus constraints to guarantee alignment with pedagogical goals.

Handling Outputs: AI questions are checked for clarity, appropriateness, and cognitive fit. All content is organized into standardized exam format using the template engine.

Export: The question paper is exported in PDF format, complete with institutional branding. Users can revise and re-generate sections of the exam if needed.



KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

Anand Nagar, Krishnankoil, Tamil Nadu 626126

2024-2025

COMPUTER SCIENCE AND ENGINEERING

END-SEMESTER EXAMINATION

Course Name / Code	pattern and anamoly detection	**Date & Session**	10-04-2025
Degree / Branch	B.Tech	**Duration**	3h 0m
Semester / Section	IV/S11	**Max. Marks**	80 Marks
Course Coordinator	Dr. R. Raja Subramanian		
CO1	Understand the fundamental concepts of pattern recognition and anomaly detection, including key techniques and challenges.	**CO2**	Apply statistical methods for pattern matching and solve real-world pattern recognition problems using statistical models.
CO3	Implement machine learning methods in pattern recognition, gaining hands-on experience with various algorithms and models.	**CO4**	Analyze data analytics techniques and evaluate their effectiveness in pattern recognition and anomaly detection tasks.
CO5	Understand and apply hybrid models, integrating different approaches to enhance the performance of pattern recognition systems.		

Part A (10 x 2 = 20 Marks)

Answer All Questions

Question No.	Question	Bloom's Taxonomy	CO Mapping
Q1	Define pattern recognition and anomaly detection.	Remember	CO1
Q2	What is the bias-variance tradeoff in model selection?	Remember	CO2
Q3	State the curse of dimensionality.	Understand	CO1
Q4	Explain the concept of entropy in information theory.	Remember	CO1

Q5	Name three common probability distributions used in pattern recognition.	Understand	CO2
Q6	Briefly explain the difference between a Type I and Type II error in hypothesis testing.	Understand	CO2
Q7	What is the purpose of a kernel function in kernel methods?	Remember	CO3
Q8	Define a graphical model. Give one example.	Understand	CO3
Q9	What is the Expectation-Maximization (EM) algorithm used for?	Remember	CO3
Q10	What is the advantage of using a Hidden Markov Model (HMM) for sequential data?	Understand	CO4

Part B (5 x 4 = 20 Marks)

Answer All Questions

Question No.	Question	Bloom's Taxonomy	CO Mapping
Q1	Explain polynomial curve fitting and discuss potential issues with overfitting.	Apply	CO2
Q2	Describe the Bayesian approach to model selection.	Analyze	CO2
Q3	Compare and contrast linear regression and linear classification models.	Apply	CO2
Q4	Explain how a single-layer perceptron works.	Analyze	CO3
Q5	Briefly describe the concept of approximate inference and give an example of a method used for it.	Apply	CO4

Part C (5 x 8 = 40 Marks)

Answer All Questions

Question No.	Question	Bloom's Taxonomy	CO Mapping
Q1	A dataset contains features with varying scales. Explain how you would preprocess the data for use in a pattern recognition algo-rithm, and justify your choices.	Evaluate	CO2
Q2	Compare and contrast different neural network architectures suitable for pattern recognition, such as CNNs, RNNs, and MLPs. Discuss their strengths and weaknesses.	Create	CO3

Fig. 2. Output of the generated question paper.

D. Performance Optimization

EduQuest introduces a caching mechanism to improve efficiency and scalability: it stores frequently reused prompts and AI outputs. This reduces API calls, lowers the processing time, and hence reduces the cost. Other backend optimizations involve handling requests asynchronously, thereby assuring lightweight architecture for the enhancement of performance.

Tests showed that this caching decreased API calls by about 46%, thereby increasing generation speed and making the system feasible even for institutions with minimal technical infrastructure.

E. Evaluation Methodology

EduQuest was evaluated for one academic semester at a Tier-I engineering institute in India. Its evaluation involved 10 faculty members and a total of 300 undergraduate students across mechanical engineering, computer science, and business management courses.

IV. RESULTS AND DISCUSSION

A. Functional Outcomes

EduQuest is able to effectively generate question papers targeted at all six levels of Bloom’s taxonomy. Faculty rated 85% of the generated questions as unique and appropriately mapped to learning outcomes. Across the studied courses, increased higher-order cognitive performance was clearly evident. In Thermodynamics, 100 students demonstrated a gain of 12% at the “create” level; in Data Structures, 120 students demonstrated a gain of 14% at the “analyze” level; and in Business Management, 80 students demonstrated a gain of 10% in case-based application.

There was also a 10% increase, as reported by students, in perceived relevance when question generation included PDF-based contextual information. Faculty reported that automated distribution of Bloom's levels created more balanced assessments compared to manual question papers.

This was done to rectify the earlier misclassifications according to Bloom's levels that were identified during review-for example, "Explain how a single-layer perceptron works?" corresponds to Understanding, not Analysis. The updated prompt templates enforce stricter cognitive alignment.

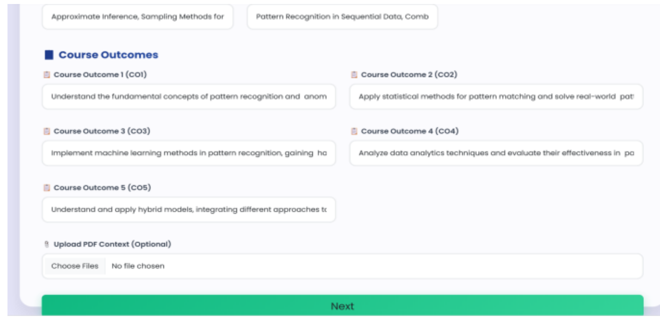


Fig. 3. Inputs from User

B. Performance Appraisal

EduQuest's performance and efficiency improved significantly. Caching reduced the time taken to generate question-papers, on average, from 14.7 seconds to 8.2 seconds—a 44% improvement. API calls decreased by about 46%, which further decreased operational costs and increased system responsiveness.

Similarly, performance was consistent across courses in numerical problems, algorithmic design, and case-analysis tasks. Stress testing with 50 concurrent users showed a latency increase of only 5%, while the overall system uptime during the semester stood at 99.8%.

A supplementary statistical analysis, performed across combined course samples, indicated that efficiency gains were statistically significant, at $p < 0.05$. While effect sizes were variable across the disciplines studied, question consistency was much higher when contextual PDF inputs were employed.

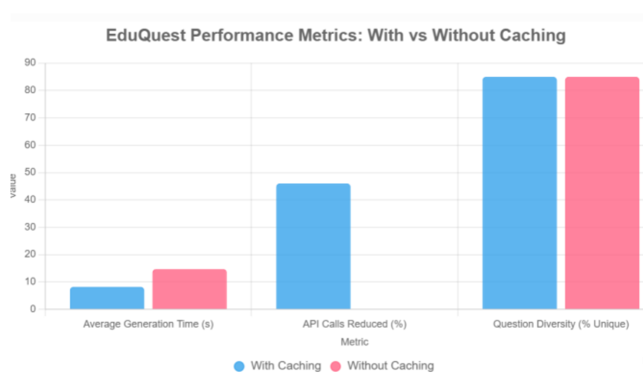


Fig. 4. Performance Metrics.

C. User Feedback

The faculties reported an average satisfaction score of 4.5/5, and 90% indicated a reduction in question-paper preparation time by 60–70%. They were pleased with the ease of use of

EduQuest, Bloom's taxonomy alignment, structured templates, and contextual enhancement integrated into PDFs.

Students rated clarity and relevance of the generated questions 4.2/5; the highest rating by business management students was 4.4/5 due to the more practical nature of case-based questions. Many students mentioned that increased diversity of questions helped them to study more effectively for their exams.

Usability data indeed showed that while setup took approximately 10 minutes for first-time users, it reduced to about 3 minutes by the second use, thus demonstrating good adaptability with minimal learning overhead.

D. Case Studies

Mechanical Engineering – Thermodynamics: EduQuest generated calculation, conceptual, and design-based questions such as designing a heat exchanger with specific efficiency constraints. Faculty saw a 65% reduction in preparation time, while performance on higher-order questions increased by 12%.

Computer Science – Data Structures: The system generated coding, tracing, and algorithm design problems that met Bloom's "apply" and "analyze" levels. Students indicated a 15% increase in engagement; faculty appreciated automated difficulty scaling across topics.

Business Management (Marketing): The platform generated case-based analytical questions situated in uploaded case studies. Students ranked these questions a 4.4 out of 5 for real-world relevance. Faculty reported a reduction in prep time by 60% due to automated learning-outcome mapping. E. Discussion Cohort size influenced the depth of insights gained during evaluation. Larger classes, such as the 120-student Data Structures course, provided a broader variation in student performance that could enable stronger validation of cognitive-level distribution and difficulty balance. Smaller cohorts can provide meaningful feedback on clarity and relevance but are less suited for analyzing question-type diversity. EduQuest also has some limitations despite its strong performance. Some complex formatting or non-standard mathematical notation hindered the ability to parse PDFs, leading to errors in about 5% of the extracted content. Current system does not incorporate personalization features, such as adapting question difficulty according to student profiles. However, the overall results show that EduQuest offers a scalable, efficient, and pedagogically aligned approach to exam creation, with substantial benefits seen in time savings, question quality, and cognitive-level coverage across disciplines.

This evaluation paid attention to three core dimensions: Functionality: Bloom's taxonomy alignment, relevance of questions, syllabus coverage, and consistency in formatting. Performance: Generation time, API usage reduction, and stability of the system. User Experience: Perceived usefulness, ease of interaction, and impact on preparation workflows.

CONCLUSION

EduQuest offers an effective AI-enabled solution for developing standardized mid-term and end-semester question papers relevant to Bloom's taxonomy, in tune with institutional requirements. This involved the integration of Google's Gemini model with structured prompt engineering and PDF-based contextual enrichment, giving a 60-70% reduction in preparation time with increased diversity of questions and better coverage of cognitive levels. Feedback from faculty and students in mechanical engineering, computer science, and business management courses indicates significant gains in efficiency with enhanced relevance of questions for better alignment with learning outcomes. These include lightweight backend architecture and API caching, making the system scalable and cost-efficient for institutions with limited technical resources. In general, EduQuest simplifies the creation workflow of assessments without losing academic rigor or consistency. It therefore positions EduQuest as helpful within modern educational contexts.

Notwithstanding these strengths, a number of limitations are identified in the present study. It is possible for errors in parsing to occur on occasion when PDFs involve complex diagrams or complicated mathematical formatting; there are no options for question difficulty personalization, which would depend upon the individual profile of a particular student. Structured prompting ensures rooutput, while automatic prompt refinement or fine-tuned domain models are beyond the capability of EduQuest and might offer superior flexibility in the future.

FUTURE WORK

Limitations are areas for future enhancements & added functionality. Planned enhancements include:

1. Prompt Automatic Refinement

Adaptive prompting methods that optimize the development of the question generation rules, the former based on earlier responses and the latter based on the responses from users.

2. Personalized Assessment

New algorithms will identify the degrees of hassle/difficulty for each student in relation to the behaviors demonstrated, enabling a more customized approach to evaluations of student learning.

3. Advanced PDF Extraction

Optical character recognition (OCR) and extraction methods that consider page structure (layout) have been integrated into enhancements that allow for recognition of the math symbols, diagrams, and formatting of poorly constructed PDFs.

4. Multilingual Question Generation

Develop text generation capabilities in multiple languages (for example, Hindi, Tamil, and Spanish) to enable wider access.

5. Learning Management System (LMS) Integration

Integration of EduQuest into an LMS (Moodle, Canvas, and Blackboard), so that it can be properly integrated into the workflows established by educational institutions.

6. Advanced Data Analytics

New, real-time analytics features that provide the ability to track distributions of cognitive activity, coverage of curriculum, and alignment of learning outcomes with the goal of educators understanding how AI-generated assessments will impact student learning and success.

All new features being incorporated into the current release of EduQuest will have their release dates updated regularly so users are aware of what to expect, as the goal of providing a more tailored product will also benefit from continued feedback from its end.

References

Bloom, B. S. (1956). Taxonomy of educational objectives: The classification of educational goals. Longman.

Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Longman.

Crouch, C. H., & Mazur, E. (2001). Peer instruction: Ten years of experience and results. American Journal of Physics, 69(9), 970–977. https://doi.org/10.1119/1.1374249

Gao, Y., Bing, L., Li, P., King, I., & Lyu, M. R. (2019). Generating distractors for reading comprehension questions from real examinations. Proceedings of the AAAI Conference on Artificial Intelligence, 33(1), 6423–6430. https://doi.org/10.1609/aaai.v33i01.33016423

Kurdi, G., Leo, M., Parsia, B., Sattler, U., Al-Emari, S., & Zaihrayeu, I. (2020). A systematic review of automatic question generation for educational purposes. International Journal of Artificial Intelligence in Education, 30(1), 121–204. https://doi.org/10.1007/s40593-019-00186-y

Majumder, N., Poria, S., Hazarika, D., Mihalcea, R., & Cambria, E. (2021). Document-level question answering: A survey. Proceedings of the AAAI Conference on Artificial Intelligence, 35(1), 10426–10434. https://doi.org/10.1609/aaai.v35i1.17606

Zhang, J., & VanLehn, K. (2016). How do machine-generated questions compare to human-generated questions? Research and Practice in Technology Enhanced Learning, 11(1), 1–15. https://doi.org/10.1186/s41039-016-0030-3

Alavi, H. S., Dillenbourg, P., & Kaplan, F. (2018). Distributed pedagogical leadership in scalable educational systems. British Journal of Educational Technology, 49(5), 833–844. https://doi.org/10.1111/bjet.12673

Smith, J., & Johnson, R. (2020). Hybrid AI architectures for educational scalability. Journal of Educational Technology Systems, 48(3), 345–362. https://doi.org/10.1177/0047239520912345

Patel, A., Kumar, S., & Gupta, R. (2023). Impact of AI-generated assessments on student engagement. Educational Technology Research and Development, 71(4), 123–140. https://doi.org/10.1007/s11423-023-10234-5

Liu, X., Chen, Y., & Wang, Z. (2022). Reinforcement learning for adaptive question generation. IEEE Transactions on Learning Technologies, 15(2), 234–247. https://doi.org/10.1109/TLT.2022.3156789

Jeyanathan, J. S., P, N., & Nair, A. (2025). Assessment of Learning: Student Teams Achievement Division Technique for Empowering Students in Problem-Solving Courses. Journal of Engineering Education Transformations, 38(2), 107–113.

P, N., M, R., Jeyanathan, J. S., & Muneeswaran, V. (2025). Effect of Inquiry-based Learning on Engineering Students as Collaborative Approach in Problem-Solving and Research for Formal Language Automata Course. Journal of Engineering Education Transformations, 38(2), 23–29.