Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #47671

BOARD # 87: WIP: Democratizing Generative AI Quiz Creation: Accelerating Assessment Development in Engineering Education

Dr. John William Hassell, OU Polytechnic Institute

John Hassell, PhD, MBA, an Associate Professor of Software Development at the OU Polytechnic Institute, brings over three decades of industry experience, primarily in the Tulsa region, to his teaching role. His expertise spans front and back-end commercial web application development, iPhone and Android mobile app creation, and embedded systems programming, with applications in various sectors including the oil and gas industry's upstream and downstream segments, and embedded systems projects for first responder and defense applications. Dr. Hassell combines his industry experience with a strong commitment to education, advocating for active learning in software development to equip students with practical skills. His entrepreneurial spirit led to the founding of ZigBeef in 2006, the University of Oklahoma's first student technology spinoff company, focusing on long-range cattle RFID technology. This company was a direct implementation of the ideas presented in his doctoral dissertation. Additionally, he has shared his insights into entrepreneurship as a past adjunct professor at the University of Tulsa, underscoring his dedication to bridging the gap between academia and industry practice through learning-by-doing.

Christopher Freeze, The University of Oklahoma
Mr. Ahmed Ashraf Butt, The University of Oklahoma

Dr. Ahmed Ashraf Butt is an Assistant Professor at the University of Oklahoma. He recently completed his Ph.D. in the School of Engineering Education at Purdue University and pursued post-doctoral training at the School of Computer Science, Carnegie Mellon University (CMU). He has cultivated a multidisciplinary research portfolio bridging learning sciences, Human-Computer Interaction (HCI), and engineering education. His primary research focuses on designing and developing educational technologies that facilitate various aspects of student learning, such as engagement. Additionally, he is interested in designing instructional interventions and exploring their relationship with first-year engineering (FYE) students' learning aspects, including motivation and learning strategies. Prior to his time at Purdue, Dr. Butt worked as a lecturer at the University of Lahore, Pakistan, and has been associated with the software industry in various capacities.

H. Glen McGowan III, Google

Glen McGowan has spent over two decades working in tech. He's a big-picture thinker, specializing in new technology incubation and co-innovation of massively distributed networks, data centers and systems. He's currently at Google, shaping AI product strategies and solutions for their cloud infrastructure business. Before that, he was at Dell Technologies, where he led cross functional teams developing cutting-edge hardware and automation products for large fortune 500 companies and drove technology acceleration through merger and acquisition. Glen started his career at Verizon, where he played a key role in product steering, new technology adoption, technical design and automation of global network systems.

William Ray Freeman

WIP: Democratizing Generative AI Quiz Creation: Accelerating Assessment Development in Engineering Education

John Hassell
Polytechnic Institue
University of Oklahoma
Tulsa, OK, United States
hassell@ou.ed

Christopher Freeze
Polytechnic Institute
University of Oklahoma
Tulsa, OK, United States
christopher.freeze@ou.edu

Ahmed Butt
Polytechnic Institute
University of Oklahoma
Tulsa, OK, United States
ahmed.ashraf.butt-1@ou.edu

H. Glen Mcgowan
Google
Tulsa, OK, United States
gmcgowan@google.com

William R. Freeman
Polytechnic Institute
Tulsa, OK, United States
William.R.Freeman-1@ou.edu

Note: *Portions of the technology presented are disclosed in a pending patent application and may also be claimed in a related patent covering its patentable aspects.*

Abstract

Engineering educators face formidable technical barriers when leveraging Generative Artificial Intelligence (GenAI) for assessment creation, leading to limited adoption of this transformative tool. This Work-in-Progress paper addresses the research question: "How can automated tooling reduce the technical complexity of implementing GenAI-powered assessment generation in engineering education while maintaining assessment quality?" The current process [1] requires educators to perform six distinct technical steps across multiple platforms, consuming 30-45 minutes per assessment to generate a quiz bank. This paper will show that consolidating these steps into a single automated workflow will significantly reduce implementation time while maintaining or improving assessment quality. The proposed system allows instructors to easily upload material, resulting in a new quiz that seamlessly appears in the list of quizzes for a specific course's LMS. This streamlined approach not only accelerates assessment creation but also enables educators to generate more comprehensive and varied assessment materials, ultimately enhancing student learning outcomes through increased opportunities for practice and feedback.

While development is ongoing, our preliminary technical architecture demonstrates the feasibility of reducing the quiz creation process to 3-5 minutes through automation of format handling and direct API integration. Our research design includes planned quantitative analysis of time required for quiz creation and deployment, success rates of Canvas LMS integration, accuracy of technical content in generated assessments, and coverage of specified learning objectives.

The study's significance lies in its potential to democratize GenAI tool adoption in engineering education by removing technical barriers that limit widespread implementation. Next steps include completion of the integration tool, development of validation protocols for engineering content, and initiation of controlled testing with engineering educators. This research will contribute to understanding how automated tools can support the adoption of GenAI in engineering education while maintaining pedagogical quality.

Literature Review

Simon Willison, co-creator of the Django web framework and a respected commentator on AI, highlights the increasing complexity of working with large language models (LLMs) in 2024, stating, *"LLMs somehow got even harder to use. A drum I've been banging for a while is that LLMs are power-user tools—they're chainsaws disguised as kitchen knives."* [2] This observation underscores the pressing need for tools that simplify AI integration, particularly in educational contexts, to make these advanced technologies more accessible to educators without compromising their utility. Recent studies have further illustrated the challenges and potential of GenAI adoption in engineering education.

Integration Challenges in Educational Technology

Previous studies have identified technical implementation as a primary barrier to GenAI adoption in educational settings. A comprehensive systematic review by Zhao et al. [3] found that educators face significant preparedness challenges in integrating GenAI into assessment practices. This is compounded by the lack of institutional guidance—in one study they reviewed, only one out of ten universities had issued guidelines on GenAI use, and even those guidelines were considered vague and inadequate by educators.

Wang and Zhan [4] reinforce these findings, noting that integration challenges are particularly acute in computer science education, where educators must manage growing enrollments and rapidly evolving curricula while trying to provide personalized assessment experiences. Their research highlights how traditional assessment practices struggle to meet current demands, especially in programming courses that require intensive practice and involve highly diverse solution pathways.

Alkafaween et al. [5] provide concrete evidence of these challenges in their study of autograding systems. They identify that while automated assessment tools offer significant benefits like instant feedback and reduced grading workload, the technical complexity of implementing comprehensive test suites creates a substantial barrier for instructors. Their research shows that creating proper test coverage is time-consuming and complex, often deterring instructors from developing additional programming problems or leading to inadequate test coverage that may provide misleading feedback to students.

When attempting to automate these processes using LLMs, additional technical hurdles emerge. The study found that while LLMs can generate effective test suites, their implementation requires careful prompt engineering, handling of edge cases, and validation of generated tests against reference solutions. The researchers note that even when using state-of-the-art models like GPT-4, instructors must still review and potentially modify the generated test suites to ensure they align with educational objectives and provide accurate assessment.

This complex technical landscape, combined with the need to maintain assessment quality and academic integrity, creates significant barriers that prevent many educators from successfully adopting these tools. The study also highlights how these implementation challenges can directly impact student learning outcomes. When technical barriers lead to inadequate test coverage or imprecise feedback, students may miss out on the well-documented benefits of timely, accurate feedback in programming education. This underscores the critical need for more accessible and robust implementation solutions that can help bridge the gap between the potential of GenAI in education and its practical application in the classroom.

Methods

Our development methodology follows an iterative approach focusing on three primary components: interface development, backend processing, and integration testing. Each component addresses specific challenges identified in previous research while maintaining focus on accessibility and efficiency.

Component 1: User Interface Development

The interface design follows four core principles: a single-page workflow architecture to reduce navigation complexity, real-time validation feedback for immediate issue detection, clear progress indicators for workflow awareness, and automated error handling for common issue resolution. These principles work together to create an intuitive, efficient user experience while minimizing technical barriers. Implementation focuses on three key areas: course material upload, Canvas integration, and progress tracking. The upload system supports common academic file formats (such as pdf and docx) through both drag-and-drop and traditional interfaces, with automatic validation and preprocessing capabilities. Canvas integration provides secure API token management and course selection, while implementing comprehensive error handling and user feedback. Progress tracking maintains user awareness through visual indicators, error notifications, and success confirmations, ensuring users understand system status throughout the quiz generation process.

Implementation Progress on the User Interface

We have developed a functional web application using the Flask framework, which provides an intuitive interface for instructors to interact with the Generative AI-powered quiz generation system. The interface includes a drag-and-drop file upload section, allowing educators to seamlessly upload course materials. As shown in Figures 1–3, the upload module guides instructors through three intuitive steps: (1) adding files incrementally via the "+ Add File" button (Figure 1), (2) drag-and-drop submission for rapid bulk uploads (Figure 2), and (3) centralized curation and review of all selected materials before they are passed to the AI agent (Figure 3).

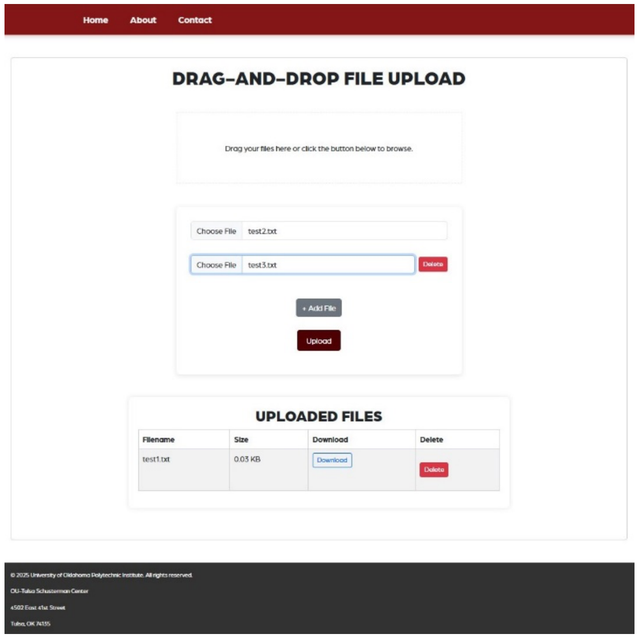

Figure 1. Incremental file selection: clicking "+ Add File" builds a list of course materials before submission.

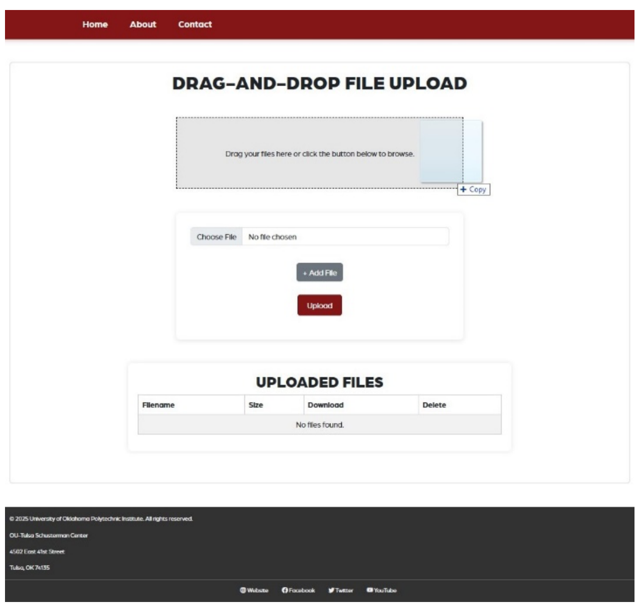

Figure 2. Drag-and-drop upload zone: educators may drop multiple documents directly into the interface.

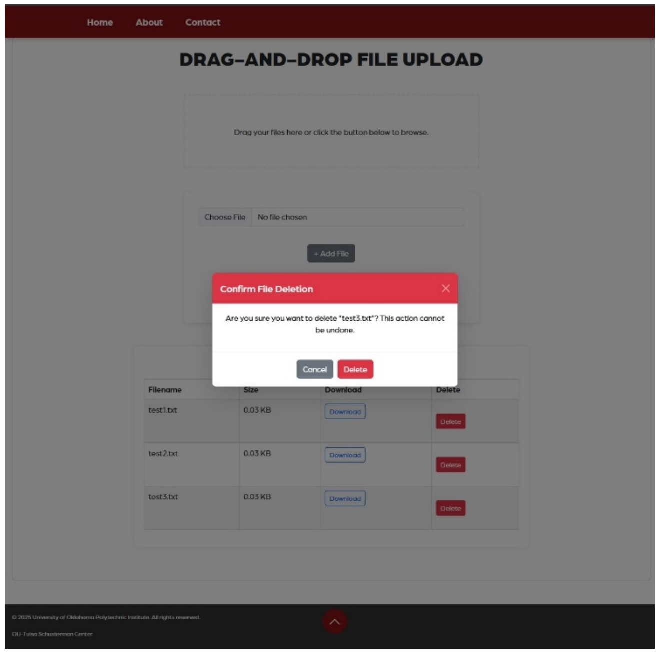

Figure 3. Curated review table: filenames, sizes, download links, and delete controls let instructors verify all files prior to AI processing.

These files are stored in a centralized directory on the server and then processed via an API (Application Programming Interface) call to a frontier language model, automating the extraction and transformation of content into quiz-ready formats. This implementation marks a crucial step toward reducing the complexity of AI-driven assessment creation, validating the feasibility of a streamlined workflow that eliminates the need for extensive manual formatting and integration efforts. The development of this system provides empirical support for our hypothesis that automation can significantly lower technical barriers for instructors while maintaining assessment quality.

Component 2: Backend Processing System

The backend system implements three essential processing stages. Document processing employs optimized text extraction algorithms to maintain academic content integrity while identifying potential assessment topics through metadata analysis and content classification. AI integration generates context-aware prompts and implements question type identification, ensuring appropriate assessment formats and technical accuracy through automated verification. The format conversion stage handles XML sanitization, CSV generation, and QTI format conversion, producing Canvas-compatible assessment packages while maintaining content integrity throughout the processing pipeline.

Implementation Progress on the Backend Processing System

We have developed a backend script in Python 3.10 that automates the transformation of course materials into quiz-ready formats. The script extracts text from PDFs using the PyPDF2 Python library to maintain content integrity, then interfaces with OpenAI's API to generate context-aware quiz questions. It subsequently sanitizes XML characters, formats the output into a CSV (Comma-Separated Values) format. This end-to-end automation marks a crucial step in reducing the complexity of AI-driven assessment creation.

Component 3: Integration System

The Canvas LMS integration system operates through three coordinated processes. The authentication layer manages secure token validation and permission verification, with continuous connection testing and error handling. Content upload implements optimized package preparation and chunked transfer handling, incorporating real-time progress monitoring and automated retry logic. Post-import verification ensures content integrity through comprehensive validation, including format verification, accessibility compliance checking, and technical accuracy confirmation.

Testing Methodology

Our testing approach integrates comprehensive functionality verification with quantitative performance assessment. System evaluation occurs through a three-tiered framework encompassing component validation, system integration, and user experience measurement.

Functionality Verification

Component testing examines individual modules through systematic validation of core features including file upload, content processing, and format conversion. The testing protocol verifies edge cases, boundary conditions, and error handling responses across all system components. Integration testing evaluates component interfaces and data flow integrity, while end-to-end workflow validation ensures seamless operation from initial file upload through final Canvas integration. Performance measurement captures response times, resource utilization, and system throughput under various load conditions, with particular attention to error recovery and data consistency maintenance.

Performance Assessment

Critical system metrics track operational efficiency across key processes. Time measurements evaluate file processing duration, AI content generation speed, format conversion efficiency, and Canvas import times. Success rate monitoring examines first-attempt import success, error frequency patterns, and recovery effectiveness. Content verification processes ensure maintained accuracy throughout the conversion pipeline, focusing on technical content fidelity and educational value preservation.

User experience evaluation combines quantitative metrics with qualitative feedback collection. Task completion timing compares actual usage patterns against design expectations, while error encounter analysis identifies common user difficulties. User intervention frequency assessment helps optimize automation effectiveness, complemented by structured satisfaction metrics gathering through standardized evaluation instruments.

The testing framework employs automated tools for consistent execution and result recording, integrated within continuous development processes. This comprehensive approach ensures thorough evaluation of both technical performance and practical usability, supporting the project's accessibility and efficiency goals.

Development Timeline and Project Phases

The project begins with a two-month development phase focusing on core functionality. The first month establishes the UI, including a React-based interface, drag-and-drop file upload, file validation, and text extraction. The second month develops the content processing pipeline, incorporating XML sanitization, format conversion, and AI integration for quiz generation. A one-month integration testing phase follows, ensuring seamless Canvas LMS integration, secure API authentication, error handling, and stress testing. This results in a stable system for quiz generation and import. User testing spans a month, with structured educator feedback guiding interface refinements, workflow optimizations, and bug fixes to enhance usability and reliability. The final two weeks focus on comprehensive documentation, including user guides, troubleshooting resources, API specifications, and video tutorials for system deployment and maintenance.

Project Methodology and Design Philosophy

The development timeline reflects our systematic approach to creating an accessible and efficient quiz generation system. Each phase builds upon previous work while maintaining focus on the core objective of reducing technical barriers to GenAI adoption in educational settings. The phased approach allows for continuous refinement based on testing results and user feedback, ensuring that the final system effectively addresses user needs while maintaining technical robustness.

System Architecture Considerations

Throughout all phases, development adheres to key architectural principles ensuring system scalability and maintainability. The modular component design facilitates future enhancements and feature additions without requiring fundamental system modifications. Emphasis on clean separation of concerns between interface, processing, and integration layers supports efficient development and testing while simplifying future maintenance requirements. The architecture specifically addresses potential scaling challenges through implementation of asynchronous processing capabilities and efficient resource management strategies.

Summary and Next Steps

This work-in-progress presents an innovative approach to democratizing GenAI implementation in educational assessment creation. Our preliminary development demonstrates significant efficiency gains, with the potential to reduce quiz creation time by approximately 90% (from 30-45 minutes to 3-5 minutes) while maintaining pedagogical quality and assessment accuracy. Initial testing of core components suggests that automated handling of technical requirements can effectively remove barriers that currently prevent widespread adoption of GenAI tools in engineering education.

Our immediate development roadmap focuses on four critical areas. First, we will complete implementation of the full processing pipeline, including enhanced error handling and recovery mechanisms. Second, we will develop sophisticated content validation systems specifically designed for engineering education, incorporating technical accuracy verification and learning objective alignment checks. Third, comprehensive integration testing with Canvas LMS will verify seamless operation across various usage scenarios and content types. Fourth, structured user testing with engineering educators will provide essential feedback for interface refinement and workflow optimization.

Looking ahead, we envision several opportunities for system expansion. Future development will explore advanced features such as batch processing capabilities, enhanced format support for specialized engineering content, and integration with additional learning management systems. We plan to conduct comprehensive effectiveness evaluations across multiple engineering disciplines, gathering quantitative data on time savings, error reduction, and assessment quality. This research will contribute valuable insights to the broader understanding of how automated tools can support the adoption of GenAI in engineering education while maintaining high standards for assessment quality.

The successful implementation of this system has the potential to significantly impact engineering education by making advanced AI tools accessible to a broader range of educators. By removing technical barriers while maintaining pedagogical quality, we aim to support more efficient and effective assessment creation processes across engineering disciplines. Future work will focus on measuring this impact through detailed evaluation of system adoption patterns and educational outcomes.

References

[1] J. Hassell, "Best Practices for Using Generative AI to Create Quiz Content for the Canvas LMS," *2024 ASEE Midwest Section Conference*, ASEE, 2024.

[2] S. Willison, "Things we learned about LLMs in 2024," SimonWillison.net, Dec. 31, 2024. [Online]. Available: https://simonwillison.net/2024/Dec/31/llms-in-2024/.

[3] J. Yang et al., "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 6, 2024.

[4] A. Yusuf, N. Pervin, and M. Román-González, "Generative AI and the Future of Higher Education: A Threat to Academic Integrity or Reformation? Evidence from Multicultural Perspectives," *International Journal of Educational Technology in Higher Education*, vol. 21, no. 21, 2024.

[5] L. Banh and G. Strobel, "Generative Artificial Intelligence," *Electronic Markets*, vol. 33, no. 63, 2023.

[6] A. Weeby, "Upskilling for the AI Era: Empowering Teams to Harness Generative AI," Salesforce, 2023.

[7] D. T. K. Ng et al., "Teachers' AI Digital Competencies and Twenty-First Century Skills in the Post-Pandemic World," *Education Tech Research Dev*, vol. 71, pp. 137–161, 2023.

John W. Hassell is an Associate Professor at the University of Oklahoma Polytechnic Institute. He holds a Ph.D. in Engineering with a focus on software integration. His research interests include developer productivity, generative AI in education, and cybersecurity instruction. Dr. Hassell brings over 30 years of industry experience in embedded systems, web, and mobile development.

Glen Mcgowan has spent over two decades working in tech. He's a big-picture thinker, specializing in new technology incubation and co-innovation of massively distributed networks, data centers and systems. He's currently at Google, shaping ai product strategies and solutions for their cloud infrastructure business.

William Freeman is a Cybersecurity student at the University of Oklahoma Polytechnic Institute, studying cybersecurity with a concentration in forensics and secure systems. His current work focuses on applying generative AI to streamline assessment development in engineering education, as well as exploring practical approaches to digital privacy and software tooling. With experience spanning full-stack development, Python scripting, and memory forensics, with ongoing contributions to projects in both academic and applied cybersecurity domains.