Majumdar, R. et al. (Eds.) (2025). Proceedings of the 1st International Conference on Learning Evidence and Analytics. Asia-Pacific Society for Computers in Education

# An Adaptive Quiz Generation System for Moodle using moosh and LLM

**Toshihiro KITA[a*], Rwitajit MAJUMDAR[a], Izumi HORIKOSHI[b] & Hiroaki OGATA[c]**
[a]*Kumamoto University, Japan*
[b]*Uchidayoko Institute for Education Research, Japan*
[c]*Kyoto University, Japan*
*kita@rcis.kumamoto-u.ac.jp

**Abstract:** This paper describes a system for automating adaptive quiz creation within Moodle. Leveraging `moosh` and the OpenAI API, it provides personalized learning. An initialization script sets up the course, while a periodically run script analyzes learner performance on existing quizzes. This analysis drives the generation of new, tailored quizzes that focus on individual weaknesses, offering remedial or supplementary exercises to enhance the learning experience. Helper functions are modularized for reusability.

**Keywords:** LMS, Question generation, Personalized learning, Generative AI

## 1. Introduction

Manually creating personalized Moodle follow-up quizzes for adaptive learning is impractical, especially as existing automated question generation attempts (Kurdi, et al., 2020) are not integrated into general LMS. This paper introduces a Python system automating personalized quiz creation. It utilizes `moosh` (Muras, n.d.) for Moodle interaction and an LLM (such as GPT-4o-mini) for generating new questions. The system operates in two phases: **Initialization**, setting up initial Moodle quizzes; and an **Adaptive Update Cycle**, periodically analyzing quiz results to generate targeted supplementary quizzes for individual learner weaknesses.

## 2. System Architecture and Usage

The Moodle adaptive quiz system (Kita, n.d.) consists of three Python scripts and one XML data file.
The `adaptive_quiz_moosh1.py` script handles **Initialization**, running once to set up a new Moodle course (Figure 1 left). It creates personalized sections, adds an initial quiz to each section, and populates these quizzes with questions from `moodle-quest1.xml`.
The `adaptive_quiz_moosh2.py` script executes the **Adaptive Update Cycle** periodically. It identifies original quizzes, fetches participant scores, and interprets these scores as bit flags to determine which specific questions were answered incorrectly (e.g., "Question No.1, Question No.3 are incorrect"). This feedback then guides the generation of new quiz content: the script uses the OpenAI API with specific prompts to create new Moodle XML questions that target the concepts from the incorrectly answered questions. A second prompt ensures strict Moodle XML formatting. Finally, these newly generated "supplementary" quizzes are added to the corresponding learner sections in Moodle (Figure 1 right). `adaptive_quiz_moosh_mod1.py` provides shared helper functions for the other scripts.

To set up the adaptive quiz system, configure `MOODLE_DIR` and the OpenAI API key. Prepare an initial Moodle quiz XML. Run `adaptive_quiz_moosh1.py` once to create the Moodle course and initial quizzes. Periodically run `adaptive_quiz_moosh2.py` (e.g., via cron) to analyze results and generate supplementary quizzes, appearing in learner sections to focus on needed practice areas.

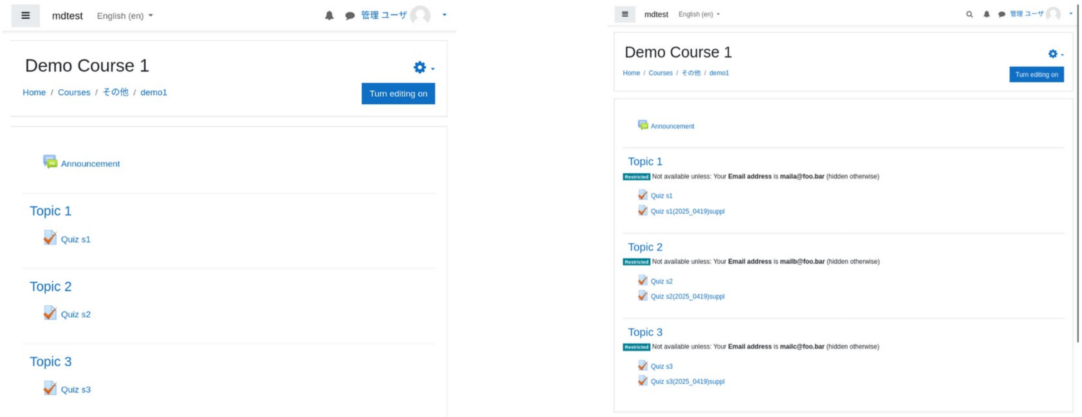


*Figure 1.* A newly created Moodle course and the quizzes by the script (left) and supplementary quizzes automatically added to each section (right)

## 3. Discussion

Although question generation functions such as the Generative AI Question Bank (Grevisse, n.d.) already exist, a mechanism lilke this framework for dynamically updating content based on the user's performance has not yet been implemented.

This automated framework provides basic adaptive quizzing in Moodle by identifying incorrectly answered questions (via bitwise score interpretation) and prompting an LLM to generate related, personalized follow-up questions. Its strengths include significant automation, tailoring content to individual weaknesses, and leveraging existing Moodle infrastructure with AI-powered question generation.

However, the system has limitations: its adaptation logic is fragile, relying on a specific bitwise score interpretation from `defaultgrade` setup, which may not accurately reflect understanding or progress. LLM output quality can vary, leading to inaccurate or poorly formatted questions, and the number of generated questions is fixed. Future work should focus on more sophisticated adaptive algorithms, robust score analysis, and improved error checking for AI-generated content. Although dependency on external LLM services can raise operational concerns such as usage fees, running LLM on an on-premise server using Ollama is an effective way to avoid these concerns.

## 4. Conclusion

The described system presents a practical approach to implementing adaptive quizzes in Moodle by integrating the `moosh` command-line tool with the generative capabilities of the OpenAI API. While the current adaptation logic based on bitwise score interpretation is relatively simple, the framework successfully automates the process of analyzing learner performance and generating targeted supplementary quiz content. This demonstrates the potential of combining existing LMS tools with AI to create more personalized and potentially more effective learning experiences.

## References

Grevisse, C. (n.d.). *Generative AI Question Bank*. https://moodle.org/plugins/qbank_genai
Kita, T. (n.d.). *moodle-quiz-autoupdate.* https://github.com/kita-toshihiro/moodle-quiz-autoupdate
Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). *A systematic review of automatic question generation for educational purposes.* International Journal of Artificial Intelligence in Education, 30, 121-204. doi: https://doi.org/10.1007/s40593-019-00186-y
Muras, T. (n.d.). *moosh*. https://moosh-online.com/