ISSN (Online): 3139-7689
Available online: www.ijsrmes.com
International Journal for Scientific Research in Modern Engineering and Science
Published in Conference Special Issue: Vol. 1, Special Issue 1, pp. 459–468, 2026

KUIZOTTAR: AN AI-POWERED AUTOMATED MCQ GENERATION AND LEARNING ANALYTICS PLATFORM

Gagandeep Kaur[1*], Smita Shrivastav[2], Harshit Patel[3], Barun Mukherjee[4], Advait Vithalkar[5], Ananya Wadekar[6]

[12]*Assistant professor, Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai.*
[3456]*Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai.*

Abstract

The preparation and evaluation of assessments remain time-intensive tasks in modern educational systems. Multiple-Choice Questions (MCQs) are widely used due to their scalability and objective grading; however, generating high-quality questions and plausible distractors requires substantial manual effort. This paper presents Kuizottar, an AI-powered assessment platform that automates the generation of MCQ quizzes using a pipeline of Natural Language Processing (NLP) techniques. The proposed system integrates document parsing, keyphrase extraction using MultipartiteRank, and question generation via the T5 transformer model. To enhance data accessibility, the platform supports multimodal input sources including PDFs, DOCX files, plain text documents, images processed through optical character recognition (OCR), and Wikipedia articles. Distractor options are generated using semantic relationships derived from WordNet and numeric heuristics for quantitative answers. The platform is implemented using a React-based frontend and a Flask backend with SQLAlchemy database integration. In addition to automated quiz generation, Kuizottar incorporates a performance analytics module that leverages a large language model to analyze assessment outcomes and provide personalized feedback to learners. The proposed system reduces educator workload while improving the efficiency and analytical depth of digital assessments.

Keywords: *DOCX files, SQL Alchemy, Multipartite Rank, T5 and BERT, tesseract OCR.*

1. Introduction

Assessments are fundamental to the educational process. However, conventional exam formulation and grading systems suffer from severe inefficiencies. As highlighted by recent systematic reviews [11], the manual formulation of high-quality questions and plausible distractors is a cognitively taxing and time-consuming process for educators, which often scales poorly in modern digital learning environments. Despite these challenges, Multiple-Choice Questions (MCQs) remain widely recognized as one of the most effective and scalable metrics for evaluating student comprehension [12].

Recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have paved the way for automated question generation to bridge this gap [5]. Recent studies have successfully leveraged advanced NLP models, such as T5 and BERT, to automate the generation of multiple-choice questions [1]. However, these foundational platforms typically restrict inputs to strictly digital text files and offer only basic numerical grading upon completion. Kuizottar addresses these critical gaps by introducing multimodal data ingestion via Tesseract OCR, allowing the system to process unselectable text and scanned documents. Furthermore, this platform goes beyond simple grading by integrating advanced LLM analytics to automatically map a student's cognitive strengths and weaknesses, delivering actionable feedback to enrich the overall learning experience.

Contributions

The main contributions of this work are summarized as follows:

1. **Multimodal Data Ingestion** – The platform supports multiple input sources including PDF, DOCX, text files, images processed through OCR, and Wikipedia topics.
2. **Automated Question Generation Pipeline** – A structured NLP pipeline combining keyphrase extraction and transformer-based text generation is used to produce high-quality questions.
3. **Semantic Distractor Generation** – Distractors are generated using lexical relationships from WordNet and heuristic numeric variations.
4. **Hybrid Quiz Creation Interface** – Educators can combine automated question generation with manual quiz creation.
5. **AI-Powered Performance Analytics** – A large language model analyzes assessment results to identify strengths, weaknesses, and personalized improvement strategies.

The proposed system aims to reduce educator workload while enhancing the depth and accessibility of digital assessments.

2. System Architecture

The Kuizottar platform follows a decoupled client-server architecture consisting of four primary layers:

1. Client Layer
2. Application Layer
3. AI Processing Layer
4. Data Storage Layer

The client layer is implemented using React and Tailwind CSS, providing an interactive interface for quiz creation, quiz participation, and result visualization [10]. The frontend communicates with the backend through RESTful APIs.

The application layer is implemented using the Flask framework in Python [9]. This layer manages user authentication, quiz generation requests, quiz attempts, and analytics processing.

The AI processing layer contains the Natural Language Processing pipeline responsible for text extraction, keyword identification, question generation, and distractor creation.

The data storage layer consists of a relational database (PostgreSQL or SQLite) accessed through SQLAlchemy. It stores user profiles, quiz metadata, generated questions, and assessment results.

Additionally, the system integrates with external AI services for advanced analytics processing.

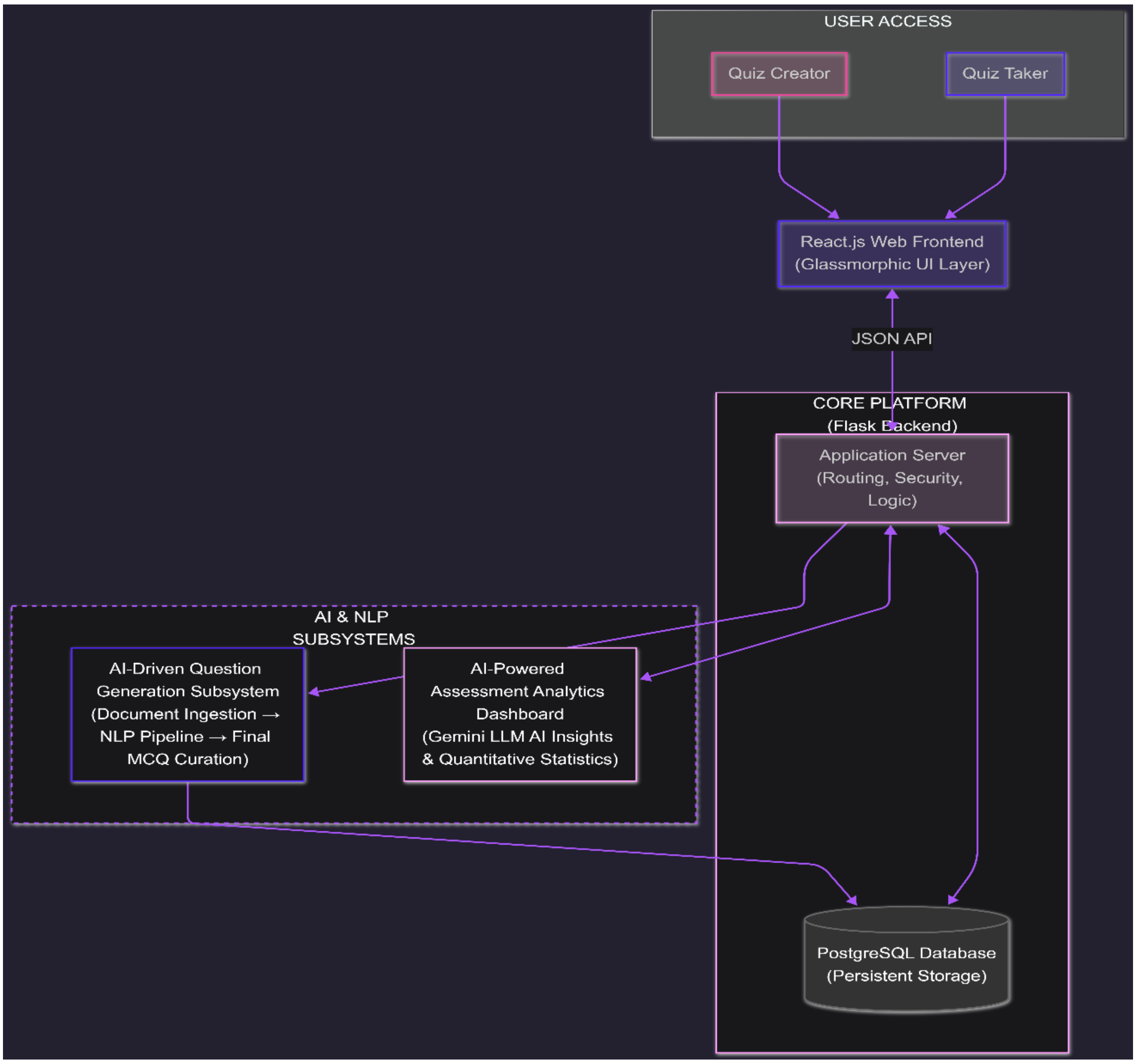

Figure 1. System Architecture

2. Methodology

The proposed system transforms unstructured textual content into structured MCQ quizzes through a multi-stage pipeline.

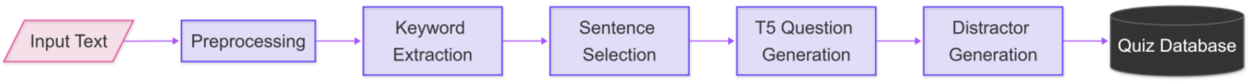

Figure 2. Workflow diagram

3.1 Data Ingestion and OCR

The system accepts diverse input formats, including PDF, DOCX, and TXT files, utilizing PyPDF2 and python-docx for reliable text extraction. For images or scanned documents, the Tesseract OCR engine is used to convert image-based text into machine-readable content [4]. This enables the extraction of knowledge directly from image formats (PNG, JPG, JPEG), a crucial capability for comprehensive document analysis that is frequently missing in standard automated quiz generators [14].

3.2 Text Pre-processing

The extracted text is processed using the Natural Language Toolkit (NLTK) [6]. Pre-processing involves several steps:

1. Sentence tokenization
2. Stopword removal
3. Text normalization
4. Candidate phrase identification

These operations ensure that the input text is clean and suitable for further NLP analysis.

3.3 Keyphrase Extraction

Keyphrase extraction is performed using the MultipartiteRank algorithm implemented in the Python Keyphrase Extraction (PKE) library.

MultipartiteRank is a graph-based ranking algorithm that constructs a multipartite graph of candidate phrases extracted from the document [7]. The importance of each phrase is determined using a ranking mechanism based on graph connectivity. High-ranking phrases represent the most important concepts within the document and are selected as potential answers for MCQ questions.

3.4 Question Formulation via T5

Instead of simple regex-based masking, Kuizottar passes the source sentence and the extracted keyword to a fine-tuned T5 (Text-to-Text Transfer Transformer) model [2]. The T5 model comprehends the context and generates a linguistically natural question targeting the specific keyword.

3.5 Intelligent Distractor Generation

Generating plausible incorrect options (distractors) is widely considered one of the most challenging aspects of automated question generation, yet it is critical for ensuring the pedagogical validity of the assessment [13]. Rather than relying on random text masking, Kuizottar employs a two-stage distractor generation strategy:

- **Linguistic Distractors:** For standard text keywords, the system queries WordNet, a large lexical database, to fetch hypernyms and hyponyms, generating semantically related but incorrect options [8].

- **Numeric Distractors:** If the correct answer is numeric, semantic relationships are not applicable. In this case, the system programmatically generates plausible mathematical variations (e.g., +/- 1, 2, 10) to ensure the options remain logically challenging.

4. Core Modules

4.1 Hybrid Quiz Generation Module

Educators are not restricted to purely automated outputs. Kuizottar features a "Hybrid Mode" consisting of:

1. **Automated AI Mode:** Educators can upload documents or specify a topic, and the system automatically generates questions using the NLP pipeline.
2. **Manual Builder Mode:** A dynamic interface allowing educators to manually draft questions, define custom

options, and select correct answers, providing ultimate editorial control.

4.2 Interactive Assessment and Anti-Cheat Mechanism

Students can participate in quizzes using a unique alphanumeric quiz code. The platform provides a timed assessment environment with configurable quiz settings.

To maintain academic integrity, the system incorporates a **forfeiture mechanism**. If a student chooses to reveal correct answers after an attempt, the system prevents further attempts to avoid unfair advantage.

4.3 AI Performance Analytics and Grading

Grading is automated instantly upon submission. However, Kuizottar extends this by routing the student's correct and incorrect question logs to the Gemini API [3]. The LLM analyzes the conceptual themes of the specific questions and returns a structured JSON payload containing:

1. Identified strong topics.
2. Identified weak topics requiring improvement.
3. A personalized, actionable 2-sentence feedback strategy.

These insights, alongside a Recharts-powered bar graph comparing the student's score against the class average and highest score, are presented on the Result Viewer dashboard.

3. Results and Analysis

5.1 Results

The proposed system was implemented and tested as a full-stack web application. The interface includes modules for authentication, quiz generation, assessment participation, and performance analytics.

The system successfully generated quizzes from multiple document formats and image-based sources using OCR. Generated questions were evaluated for grammatical correctness, contextual relevance, and distractor quality.

The results demonstrate that the proposed system can efficiently transform unstructured textual content into structured MCQ assessments while providing meaningful feedback for learners.

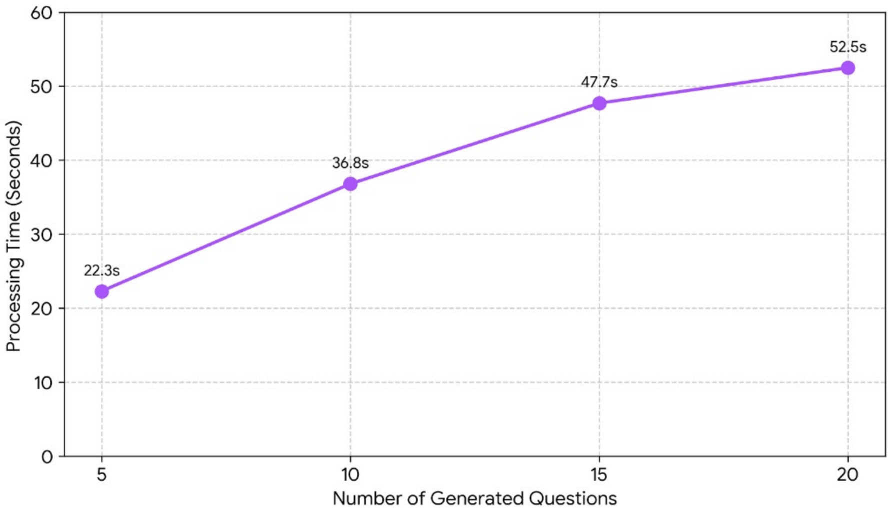

Figure 3. Kuizottar System Performance: Question Generation Latency

To evaluate the computational efficiency of the Kuizottar pipeline, a performance analysis was conducted measuring the total processing latency against the volume of generated questions. As illustrated in Figure 3, the system demonstrated a generation time of 22.3 seconds for a standard 5-question assessment. The latency scales efficiently, processing a comprehensive 20-question exam in just 52.5 seconds. This sub-minute performance highlights the pipeline's capability to deliver rapid, scalable automated assessments without significant computational bottlenecks.

Figures 4–10 illustrate various components of the implemented platform, including the login interface, dashboard, quiz generation interface, assessment interface, and analytics dashboard.

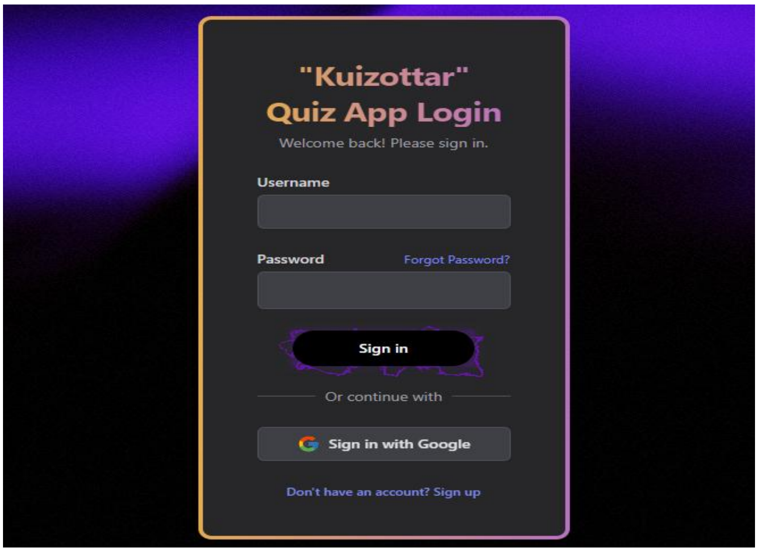

Figure 4. Login Interface

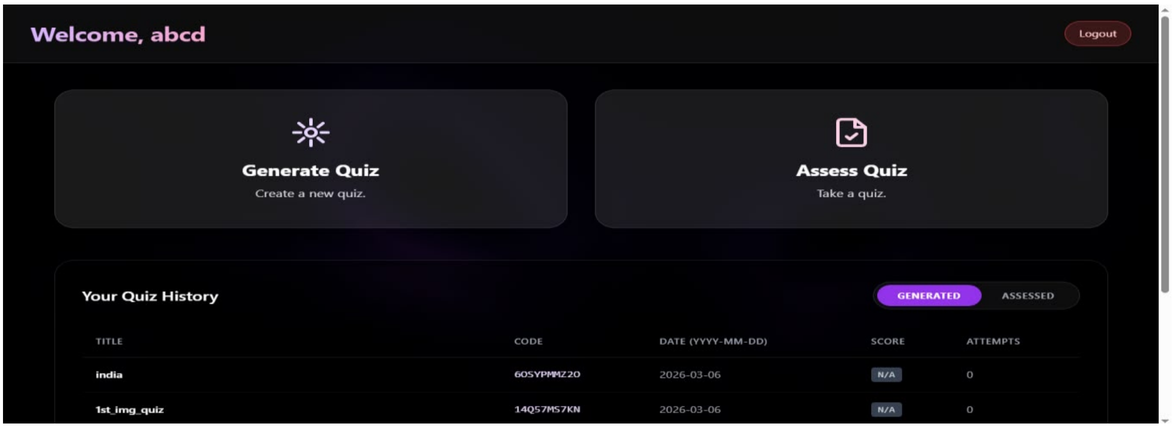

Figure 5. User Dashboard

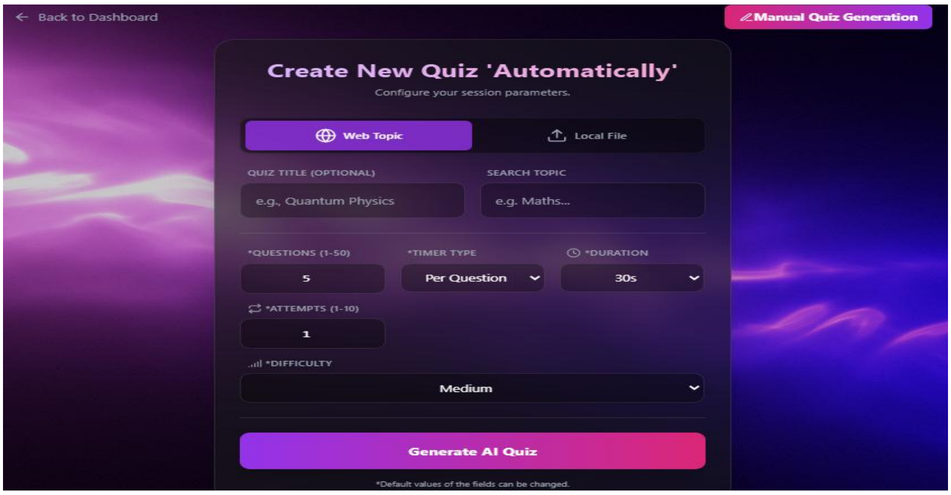

Figure 6. Automatic Quiz Generation Interface

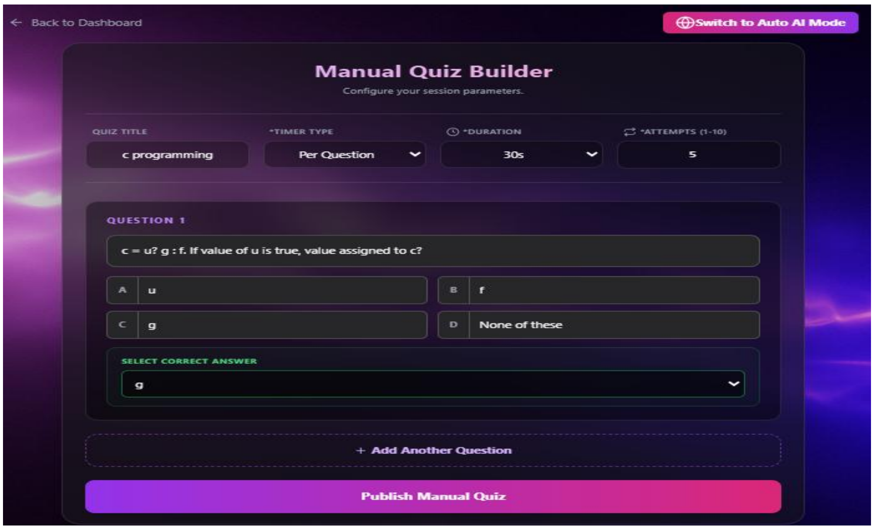

Figure 7. Manual Quiz Generation Interface

Figure 8. Assessment Interface

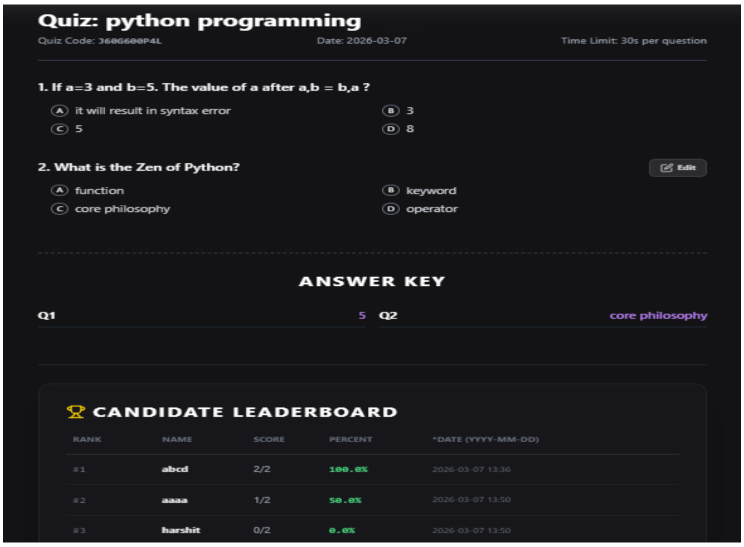

Figure 9. Quiz view and Leaderboard for the Creator

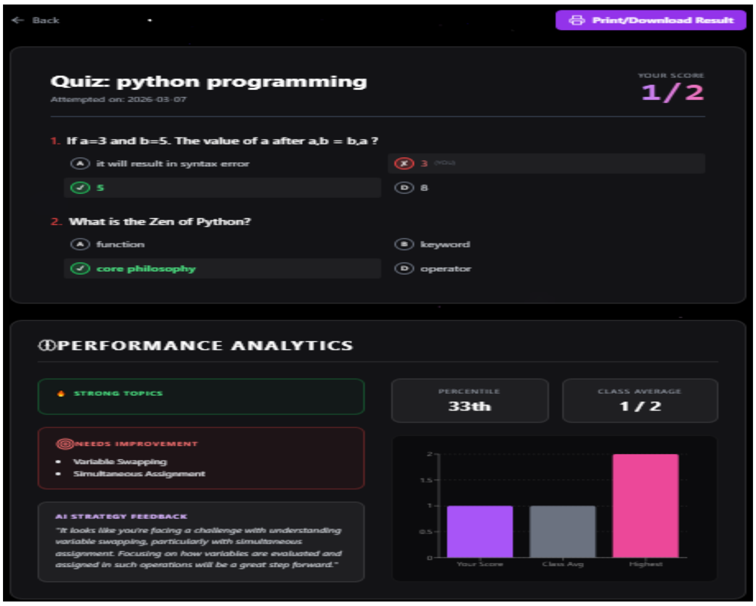

Figure 10. Quiz Result and Analytics dashboard

5.2 Analysis

To evaluate the comprehensiveness of Kuizottar, Table 1 presents a comparative analysis against existing automated assessment platforms. As demonstrated, Kuizottar uniquely integrates OCR capabilities and LLM-powered analytics, distinguishing it from prior literature [1].

Table 1. Feature Comparison of Existing Systems vs. Kuizottar

Features	Questgen	Quizly	OpExam	EduAssessPro [1]	Kuizottar (Our system)
MCQ Generation via NLP	Yes	Yes	Yes	Yes	Yes
Exam Creation & Hosting	No	No	Yes	Yes	Yes
Tesseract OCR (Image to Text)	No	No	No	No	Yes

Hybrid Mode (Manual + AI)	No	No	No	No	Yes
Timer & Attempt Limits	No	No	No	Yes	Yes
LLM-Powered Analytics	No	No	No	No	Yes

4. Conclusion

This paper presented Kuizottar, an AI-driven platform for automated quiz generation and performance analytics. The system integrates multiple NLP techniques, including keyphrase extraction, transformer-based question generation, and semantic distractor creation, to convert unstructured textual data into meaningful MCQ assessments. By supporting multimodal inputs such as documents, images, and web content, the platform significantly expands the range of educational resources that can be transformed into quizzes. Additionally, the integration of AI-based analytics enables deeper insights into student learning patterns beyond simple numerical scores.

Future work may focus on extending the system to support subjective question generation, automated grading of descriptive answers, and adaptive quiz mechanisms that dynamically adjust difficulty based on student performance.

References

[1] A. Khairkar, S. Bulbule, S. Manchalkar, S. Phatate, and K. Soni, "Eduassesspro: An Automated Quiz

[2] Generator and Exam Creator," International Research Journal on Advanced Science Hub, vol. 06, issue 07, pp. 189-194, July 2024.

[3] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5)," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1-67, 2020.

[4] Google, "Gemini API Documentation," Google AI for Developers, 2024. [Online]. Available: https://ai.google.dev/.

[5] R. Smith, "An Overview of the Tesseract OCR Engine," Ninth International Conference on Analysis and Recognition (ICDAR 2007), Parana, Brazil, 2007, pp. 629-633.

[6] B. Das, M. Majumder, and S. Phadikar, "Automatic question generation and answer assessment: a survey," Research and Practice in Technology Enhanced Learning, vol. 16, no. 5, 2021.

[7] S. Bird, E. Loper, and E. Klein, Natural Language Processing with Python, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2009.

[8] F. Boudin, "Unsupervised Keyphrase Extraction with Multipartite Graphs," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 515-520.

[9] G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol. 38, no. 11, pp. 39-41, Nov. 1995.

[10] A. Ronacher, "Flask: A Python Microframework," 2010. [Online]. Available: HTTPs.

[11] Meta Platforms, Inc., "React – A JavaScript library for building user interfaces," 2013. [Online]. Available: https://react.dev/.

[12] A. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A Systematic Review of Automatic

[13] Question Generation for Educational Purposes," International Journal of Artificial Intelligence in Education, vol. 30, pp. 121-204, 2020.

[14] D. Nicol, "E-assessment by design: using multiple-choice tests to good effect," Journal of Further and Higher Education, vol. 31, no. 1, pp. 53-64, 2007.

[15] S. Susanti, H. Tokunaga, H. Nishikawa, and H. Obari, "Automatic Multiple-Choice Question Generation from Text: A Survey and a Case Study," Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, pp. 46-55, 2014.

[16] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," IEEE Access, vol. 8, pp. 142642 142668, 2020.

[17] "PTQ vs QAT Accuracy Trade-offs," ResearchGate.(https://www.researchgate.net/figure/Accuracy-of-Res-Net-18-for-different-sized-calibration-sets-at-various-quantization-levels_fig9_355019034)