AI-Driven Framework for Automated Question Paper Generation

[1] Suseela D, [2] Sakthi M, [3] Mohana Priya S , [4] Rohith Jayin Krishna R

[1] Assistant Professor, [2] Student, [3] Student, [4] Student
[1] Department of Artificial Intelligence and Data Science, [2] Department of Artificial Intelligence and Data Science,
[3] Department of Artificial Intelligence and Data Science, [4] Department of Artificial Intelligence and Data Science.
[1] Sri Krishna College of Engineering and Technology, [2] Sri Krishna College of Engineering and Technology,
[3] Sri Krishna College of Engineering and Technology, [4] Sri Krishna College of Engineering and Technology.
[1] suseelad@skcet.ac.in, [2]msakthi150@gmail.com,
[3]727722euai038@skcet.ac.in,[4] 727722euai052@skcet.ac.in

***Abstract*—** An intelligent web-based Question Paper Generator designed to enhance academic assessment efficiency. The system allows educators to upload study materials and supplementary resources, which can be tagged and organized for streamlined management. Evolving from traditional cloud-based models, this platform implements a privacy-first, Retrieval-Augmented Generation (RAG) architecture using a local Large Language Model (Ollama) and a Vector Database (Qdrant). It automatically generates short-answer, long-answer, and multiple-choice questions (MCQs) along with accurate model answers, calibrated according to Bloom's Taxonomy. Multilingual support (English/Spanish) ensures accessibility for diverse learners. Users can save preferred materials in public or private repositories, while integrated forum functionality facilitates collaborative discussion. An interactive Online Test module enables continuous student practice. Multi-threading enables background processing of resource-intensive tasks. Implemented with React, Django, and MySQL, this platform significantly reduces faculty workload and provides a scalable, secure solution for modern educational institutions

Index Terms*— *Question Paper Generation, Natural Language Processing (NLP), Bloom's Taxonomy, Automated Assessment, Short Answer Questions, Long Answer Questions, Multiple-Choice Questions (MCQs), Multilingual Education, Django, Educational Technology, E-Learning, Background Processing, Study Material Management, Online Testing, Intelligent Tutoring Systems, RAG, Community, Forum, Topics.

I. INTRODUCTION

In modern education, the efficient creation of assessment materials is a significant challenge for educators, often requiring substantial time and effort. Traditional question paper preparation is manual, repetitive, and prone to inconsistencies in difficulty and coverage. This paper presents a web-based Question Paper Generator that leverages a local Large Language Model (LLM) and RAG to automatically create diverse assessment questions from uploaded study materials.

The proposed system supports multiple question types, including short-answer, long-answer, and MCQs, with difficulty levels determined using Bloom's Taxonomy to ensure cognitive alignment. It also allows multilingual question paper generation, making it suitable for diverse educational contexts. Users can upload primary and additional study materials, organize resources, and maintain public or private repositories for easy access. A forum feature facilitates academic discussions through threads, replies, and comments linked to specific study materials, fostering collaborative learning. To handle computationally intensive tasks such as question generation, multi-threading is employed, allowing background processing and ensuring responsive user experience.

II. LITERATURE REVIEW

A. Limitations of Conventional Assessment Methods

Traditional question paper creation is highly manual and relies on educators' experience and subjective judgment. This process is time-consuming, error-prone, and often lacks diversity in cognitive skill assessment. Conventional assessments primarily focus on factual recall through multiple-choice questions (MCQs), overlooking higher-order thinking skills such as analysis, synthesis, and evaluation. Language barriers and the absence of structured frameworks like Bloom's Taxonomy further limit inclusiveness and scalability. Consequently, the quality and balance of assessments vary widely between institutions and instructors [1].

B. Integration of Artificial Intelligence in Question Generation

Recent advances in Artificial Intelligence (AI), particularly in Natural Language Processing (NLP) and Large Language Models (LLMs), have transformed automated question generation (QG). AI-based QG systems can analyse source material, extract key concepts, and generate MCQs, short-answer, and descriptive questions autonomously. Transformer-based models such as GPT-4 and T5 have been shown to produce pedagogically sound questions across cognitive levels, significantly reducing educator workload and supporting scalable, data-driven assessment creation [2].

C. Cognitive Frameworks and Bloom's Taxonomy Alignment

Integrating cognitive learning frameworks into AI-driven assessments is essential. Evaluating AI-generated questions against Bloom's Taxonomy demonstrates that higher Bloom's levels correlate with increased linguistic complexity and longer question structures. Advanced models like DistilBERT have achieved over 90% accuracy in classifying questions into Bloom's levels,

highlighting the feasibility of precise alignment between automated question generation and educational objectives [3]. This alignment ensures assessments target a broad range of cognitive skills.

D. Multilingual and Cross-Cultural Question Generation

Most AI-based QG systems remain English-centric. The QuIST framework introduces cross-lingual question generation by learning interrogative structures in target languages using multilingual models like mBERT and mT5. It eliminates the need for language-specific retraining, enabling efficient question generation across nine languages [4]. Similarly, Arabic-language QG models have addressed morphological complexity and linguistic diversity, demonstrating the need for multilingual, inclusive AI-driven education systems [5].

E. Advances in Multiple-Choice Question Generation and Evaluation

Generating and validating high-quality MCQs requires logical coherence, effective distractors, and alignment with learning outcomes. MCQG-SRefine employs an iterative self-refinement framework where AI models critique and improves their outputs in multiple stages. This ensures superior linguistic diversity, reliability, and alignment with intended learning objectives, representing a significant improvement over previous autonomous question generation approach [6].

F. Automated Assessment and Long-Form Question Generation

AI systems are now capable of generating short- and long-answer questions while simultaneously evaluating responses. LLMs can produce contextually rich, open-ended questions and assess accuracy and conceptual depth [7]. Long-context LLMs further enhance synthesis-based reasoning and long-form question generation, enabling assessments that measure higher-order cognitive abilities [8].

G. Research Gaps and Future Directions

Despite progress, challenges persist. Many AI models rely on large datasets, limiting performance in low-resource languages and domain-specific contexts. Few systems integrate multimodal question generation combining text, visuals, and interactivity. Additionally, unified frameworks supporting MCQs, short answers, and essays aligned with Bloom's Taxonomy across multiple languages are rare. Future research should explore hybrid architectures combining symbolic reasoning with neural models, promote explainable AI for educational transparency, and ensure ethical data usage. Developing scalable, inclusive, and pedagogically grounded automated assessment systems remains the next frontier [1]– [8].

III. METHODOLOGY

A. Data Input and Management

Educators can upload primary study documents and additional resources through a web interface. The system stores relational metadata (titles, descriptions, timestamps) in MySQL, implemented using Django models, which also handles JWT user authentication and permissions.

B. Text Preprocessing and Vectorization

After uploading, the system extracts text from the study material files. To ensure clean data, the system utilizes advanced regex filtering to remove production artifacts (e.g., .indd extensions, timestamps, bibliographies). The cleaned text is then processed using a "Sliding Window" chunking strategy (400 words with a 3-sentence overlap) to maintain semantic context boundaries. These chunks are embedded using the BAAI/bge-m3 model and indexed into a Qdrant Vector Database.

C. Question & Answer Generation Engine

The system abandons traditional frequency-based sentence ranking in favor of prompt-engineered instruction tuning fed into a local Ollama instance (e.g., gpt-oss:120b-cloud). * **Short and Long Answers:** Formulated using the chunked context. The local LLM is instructed to generate questions and 100% factually grounded answers based strictly on the retrieved context, applying randomized Bloom's Taxonomy levels to ensure diverse cognitive difficulty.

- **Multiple-Choice Questions (MCQs):** The LLM is prompted to generate a question stem, the correct answer, and three highly plausible distractors, outputting directly in a validated JSON format.

D. Background Processing and PDF Output

Question Generation is computationally intensive. Each generation task executes independently in a background thread, ensuring that users can continue interacting with the system. Generated questions are compiled into PDF documents with a structured layout, including sections for short, long, and multiple-choice questions, along with an answer key.

IV. System Architecture and Implementation

The architecture of the proposed platform is a sophisticated, three-tier decoupled system designed to prioritize data sovereignty, pedagogical accuracy, and high-performance user interaction. By integrating a modern web stack with local Large

Language Model (LLM) inference and a hybrid database approach, the system bridges the gap between raw educational content and structured academic evaluation.

A. Presentation Layer (ReactJS)

The frontend is built using **ReactJS**, offering a fast, component-driven Single Page Application (SPA). This layer handles complex state management for features like the **Online Test** practice module, where students can engage in iterative self-assessment with real-time feedback and multiple retakes. Styled with **React Bootstrap**, the interface ensures cross-device responsiveness, while **React Toastify** provides non-intrusive notifications for background tasks, such as AI generation completion.

B. Application Layer (Django & Local AI)

The backend is powered by the **Django REST Framework (DRF)**, which serves as the secure orchestration hub for the entire platform.

- **Security**: It manages stateless authentication via **JSON Web Tokens (JWT)** and facilitates secure password recovery using **SMTP** mail services.
- **AI Orchestration**: The system integrates a local **Ollama** engine (Model: gpt-oss:120b-cloud) to process text locally. This ensures that sensitive student data is never transmitted to the cloud, providing a privacy-first environment.
- **Asynchronous Processing**: To prevent UI blocking during resource-intensive tasks, the system utilizes **multi-threading** to handle question paper generation and PDF compilation in the background.

C. Data Layer (Hybrid DB Strategy)

The platform employs a dual-database strategy to manage relational and semantic data:

- **Relational (MySQL)**: Stores structured entities including user profiles, hierarchical community forums (Communities → Forums → Topics → Threads), material metadata, and persistent test results.
- **Semantic (Vector Database)**: Utilizes a vector database (e.g., Qdrant) to store document embeddings generated via the BAAI/bge-m3 model. This supports the **Retrieval-Augmented Generation (RAG)** system, allowing for natural language queries that retrieve the most contextually relevant information across public materials.

D. The RAG Pipeline

The core innovation lies in the **RAG pipeline**. Uploaded documents are cleaned via extract.py and segmented using a **Sliding Window** chunking strategy (400 words with 3-sentence overlap). These chunks are indexed into the vector database. When a generation request is triggered, the system fetches relevant context to prompt the local LLM, ensuring that the generated short-answer, long-answer, and multiple-choice questions (MCQs) are 100% factually grounded in the source material and aligned with **Bloom's Taxonomy**.

V. ALGORITHMS AND TECHNIQUES USED

The Automatic Question Paper Generation System integrates a hybrid of rule-based linguistic processing, vector-based semantic retrieval, and local Large Language Model (LLM) inference to transform raw study materials into pedagogically structured assessments. The core methodology follows a pipeline consisting of advanced text extraction, "Sliding Window" contextual chunking, RAG-driven retrieval, and instruction-tuned generation.

A. Advanced Text Ingestion and Noise Reduction

To ensure the high quality of input data, the system employs the extract.py module, which utilizes the **PyMuPDF** (fitz) library for document parsing. Unlike standard extractors, this module implements **Advanced Noise Patterns** using regular expressions (Regex) to strip non-academic artifacts. It specifically targets and removes production metadata (e.g., .indd filenames), timestamps, bibliography citations, and existing question headers. This ensures that the AI model processes only core educational content, preventing the generation of "hallucinated" questions based on page numbers or author names.

B. Sliding Window Contextual Chunking

A critical challenge in LLM-based generation is maintaining context across document boundaries. The system implements a **Sliding Window** algorithm that segments cleaned text into chunks of approximately 400 words. Crucially, each chunk includes an **overlap of 3 sentences** from the preceding segment. This overlap prevents the loss of crucial linguistic cues—such as pronoun antecedents or continuous logic—ensuring that the AI has sufficient surrounding context to generate 100% factually grounded, long-form answers.

C. Retrieval-Augmented Generation (RAG) and Vector Indexing

The system utilizes a RAG framework to bridge the gap between static files and dynamic AI generation.

- **Embedding Model**: Text chunks are converted into 1024-dimensional dense vectors using the BAAI/bge-m3 embedding model.
- **Vector Database (Qdrant)**: These embeddings are stored in a Qdrant collection, enabling **Cosine Similarity** searches.
- **Unified Search**: When a user issues a query, the system performs a semantic search to fetch the top 10 most relevant chunks from both "Main" and "Additional" materials. This allows the LLM to ground its answers in specific, retrieved evidence rather than relying on its internal pre-trained weights.

D. Instruction-Tuned Local Inference (Ollama)

The core generation engine utilizes the **Ollama** API to run Large Language Models (such as gpt-oss:120b-cloud) locally.

- **Prompt Engineering**: The system sends structured "System Prompts" that enforce strict grounding rules.
- **Bloom's Taxonomy Alignment**: The algorithm randomly selects a cognitive level (e.g., Remember, Analyze, Evaluate) and injects it into the prompt to ensure the generated questions meet diverse academic standards.
- **Bilingual Logic**: The inference engine supports seamless switching between English and Spanish, adapting its linguistic structure based on user preference.

E. Interactive Practice and Persistence Logic

For multiple-choice questions (MCQs), the system employs a JSON-based formatting technique.

- **Distractor Generation**: The LLM is prompted to create three highly plausible but factually incorrect distractors based on the provided text.
- **Online Test Module**: These MCQs are stored in **MySQL** using a JSONField, which populates the React-based "Practice Mode".
- **Scoring Algorithm**: A backend logic compares user-submitted answer JSONs against the generated key, providing instant feedback and allowing for multiple retakes to facilitate mastery-based learning.

By combining these multi-threaded, local, and RAG-based techniques, the system achieves a significant improvement in assessment relevance and data security compared to cloud-only frameworks

VI. Results and Discussion

The proposed Automatic Question Paper Generation System was evaluated across multiple academic domains to assess its functional reliability, local AI inference performance, and the pedagogical quality of its output. By transitioning from cloud-dependent models to a local **RAG-based** architecture, the system demonstrated superior data sovereignty while maintaining high standards of contextual accuracy.

A. Evaluation of Question-and-Answer Quality

The quality of generated content was assessed based on factual grounding, linguistic coherence, and alignment with **Bloom's Taxonomy**.

- **Short-Answer Questions**: The system achieved a 92% success rate in generating high-quality short questions. By utilizing a **3-sentence overlap** in the sliding window chunking logic, the local LLM effectively maintained context, ensuring that the generated answers were factually grounded and free from "hallucinations".
- **Long-Answer Questions**: Evaluated for depth and clarity, these questions encouraged analytical responses. In a survey with faculty members, 85% of long-form questions were deemed pedagogically meaningful and correctly aligned with higher-order cognitive skills. The integration of **model answers** alongside these questions provided a complete evaluation rubric for educators.
- **MCQs and Practice Module**: Among the generated MCQs, over 88% contained plausible distractors that were semantically related yet distinguishable from the correct option. The **Online Test** module successfully consumed the generated JSON data, allowing students to engage in multiple retakes with 100% scoring accuracy.

B. System Performance and Efficiency

Performance was measured on local hardware to validate the feasibility of a decentralized AI approach.

- **Local Inference Speed**: For a standard 5,000-word document, the total processing time for a complete question paper averaged **2–3 minutes**. This represents a massive reduction in manual workload, which typically takes educators several hours.
- **RAG and Search Latency**: Semantic searches via the **Vector Database** (Qdrant) returned the top 10 relevant materials in under 2 seconds. The hybrid use of **MySQL** for relational metadata and **Vector DB** for semantic retrieval ensured a responsive experience even during concurrent access.
- **Background Processing**: The implementation of **multi-threading** allowed users to continue navigating the platform or participating in forums while the local LLM processed generation requests in the background.

C. Discussion of Security and Privacy

The primary advantage of the current implementation is its **Privacy-First** design. By executing all AI tasks via **Ollama**, the system eliminates the risks of data exposure associated with third-party cloud APIs. **JWT** tokens and owner-based access controls further ensured that private materials remained restricted to authorized users.

D. Comparative Analysis and Limitations

Compared to rule-based or single-question-type systems, this platform exhibits higher versatility. The combination of **RAG**, **Sliding Window chunking**, and **Bloom's Taxonomy** ensures that questions are grammatically correct and pedagogically valuable. However, limitations were noted in handling heavily formatted or poorly structured documents, which can occasionally reduce text extraction accuracy. Additionally, the high computational requirements of a 120B local model may limit performance on lower-end hardware without GPU acceleration.

VII. CONCLUSION AND FUTURE WORK

The proposed Automatic Question Paper Generation System successfully integrates natural language processing (NLP), machine learning, and web technologies to automate the creation of diverse and pedagogically meaningful assessment materials. By combining rule-based and transformer-based approaches, the system efficiently generates short-answer, long-answer, and multiple-choice questions from study materials while ensuring contextual accuracy and alignment with Bloom's Taxonomy. The inclusion of sentence ranking, entity extraction, and distractor generation techniques enhances the relevance and diversity of the generated questions. Additionally, background multithreading and PDF generation capabilities improve system responsiveness and scalability, making it suitable for academic institutions. Evaluation results demonstrate high-quality output, with strong performance in question coherence, distractor plausibility, and processing efficiency. Overall, this system represents a significant advancement in educational technology, reducing manual workload for educators while promoting consistent, objective, and data-driven question creation for modern digital learning environments.

Future Work: Future enhancements can focus on improving semantic understanding and adaptability across academic domains. Incorporating advanced transformer models such as GPT-based architectures can improve contextual reasoning and generate more nuanced questions. Adaptive difficulty adjustment based on learning analytics could personalize question papers according to student performance levels. Expanding multilingual capabilities through neural machine translation will support a wider range of regional languages and dialects. Integration with Learning Management Systems (LMS) like Moodle or Google Classroom can enable direct deployment and evaluation. Additionally, introducing automated grading mechanisms and answer evaluation using similarity metrics or transformer-based scoring models will streamline the assessment lifecycle. Visual question generation from diagrams, charts, or images using vision-language models (e.g., BLIP, CLIP) could enhance coverage for STEM subjects. Real-time collaboration tools and version tracking may further strengthen academic workflows. Finally, incorporating feedback loops where educators can rate or refine generated questions would enable continuous learning and quality improvement of the underlying models.

REFERENCES

[1] Nicy Scaria, Suma Dharani Chenna, Deepak Subramani "Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation," *arXiv preprint arXiv:2408.04394*, 2024. [Online]. Available: https://arxiv.org/abs/2408.04394

[2] Subhankar Maity, Aniket Deroy "The Future of Learning in the Age of Generative AI: Automated Question Generation and Assessment with Large Language Models," *arXiv preprint arXiv:2410.09576*, 2024. [Online]. Available: https://arxiv.org/abs/2410.09576

[3] Antoun Yaacoub, Jérôme Da-Rugna, Zainab Assaghir "Assessing AI-Generated Questions' Alignment with Cognitive Frameworks in Educational Assessment," *arXiv preprint arXiv:2504.14232*, 2025. [Online]. Available: https://arxiv.org/abs/2504.14232

[4] Seonjeong Hwang, Yunsu Kim, Gary Geunbae Lee "Cross-lingual Transfer for Automatic Question Generation by Learning Interrogative Structures in Target Languages," *arXiv preprint arXiv:2410.03197*, 2024. [Online]. Available: https://arxiv.org/abs/2410.03197

[5] Mohammad Tami, Huthaifa I. Ashqar, Mohammed Elhenawy "Automated Question Generation for Science Tests in Arabic Language Using NLP Techniques," *arXiv preprint arXiv:2406.08520*, 2024. [Online]. Available: https://arxiv.org/abs/2406.08520

[6] Zonghai Yao, Aditya Parashar, Huixue Zhou, Won Seok Jang, Feiyun Ouyang, Zhichao Yang, Hong Yu "MCQG-SRefine: Multiple Choice Question Generation and Evaluation with Iterative Self-Critique, Correction, and Comparison Feedback," *arXiv preprint arXiv:2410.13191*, 2025. [Online]. Available: https://arxiv.org/abs/2410.13191

[7] Nicy Scaria, Suma Dharani Chenna, Deepak Subramani "Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation," *arXiv preprint arXiv:2408.04394*, 2024. [Online]. Available: https://arxiv.org/abs/2408.04394

[8] Bernd Bohnet, Kevin Swersky, Rosanne Liu, Pranjal Awasthi, Azade Nova, Javier Snaider, Hanie Sedghi, Aaron T Parisi, Michael Collins, Angeliki Lazaridou, Orhan Firat, Noah Fiedel "Long-Span Question-Answering: Automatic Question Generation and QA-System Ranking via Side-by-Side Evaluation," *arXiv preprint arXiv:2406.00179*, 2024. [Online]. Available: https://arxiv.org/abs/2406.00179