Enhancing Automated Exam Creation with Retrieval-Augmented Generation for Scalable Educational Assessment

Charaf Hamidi [1], Mohamed Badiy [2], Salma Gaou [1], Fatima Amounas [2], Mourade Azrour [2,*], Hicham Tribak [3], Abdullah M. Alnajim [4], and Abdulatif Alabdulatif [5]

[1] Laboratory of Engineering Sciences, Faculty of Science Agadir, University Ibn Zohr, Agadir, Morocco
[2] MSIA Team, IMIA Laboratory, Faculty of Sciences and Techniques, Moulay Ismail University of Meknes, Errachidia, Morocco
[3] Physics, Energy, and Information Processing, Multidisciplinary Faculty, University Ibn Zohr, Ouarzazate, Morocco
[4] Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia
[5] Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia
Email: charaf.hamidi.24@edu.uiz.ac.ma (C.H.); badiy.mohamed2@gmail.com (M.B.); S.gaou@uiz.ac.ma (S.G.); f.amounas@umi.ac.ma (F.A.); mo.azrour@umi.ac.ma (M.A.); h.tribak@uiz.ac.ma (H.T.); najim@qu.edu.sa (A.M.A.); ab.alabdulatif@qu.edu.sa (A.A.)
*Corresponding author

***Abstract*—The integration of Artificial Intelligence (AI) into educational assessment has led to significant advancements in automated exam generation. This study proposes a novel Retrieval-Augmented Generation (RAG)-based system designed to automate the creation of exam questions while ensuring contextual accuracy and adaptability to pedagogical needs. By leveraging state-of-the-art Natural Language Processing (NLP) techniques, including LangChain, Facebook AI Similarity Search (FAISS), and Large Language Model Meta AI (LLaMA) 3.2, the system retrieves domain-specific knowledge and generates multiple-choice and open-ended questions dynamically. The framework incorporates an interactive user interface built with Next.js, enabling customizable quiz settings, real-time answer verification, and performance tracking. A structured knowledge base is developed through automated web scraping and PDF content extraction using LLaMA Parse, ensuring the generation of precise and curriculum-aligned questions. The proposed system enhances educational assessment by reducing manual effort, ensuring scalability, and providing immediate feedback to learners. Experimental evaluations demonstrate its effectiveness in producing pedagogically sound Structured Query Language (SQL) assessments in French, highlighting the potential for broader applications in automated exam generation. This research contributes to the growing field of AI-driven educational tools, paving the way for future enhancements in adaptive learning and intelligent assessment methodologies.**

***Keywords*—retrieval-augmented generation, natural language processing, Artificial Intelligence (AI) in education, automated exam generation, LangChain, Facebook AI Similarity Search (FAISS), Large Language Model Meta AI (LLaMA) 3, educational assessment**

Manuscript received February 24, 2025; revised April 30, 2025; accepted May 20, 2025; published October 14, 2025.

I. Introduction

The rapid advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have significantly transformed various domains, particularly education [1–3]. Traditional assessment methods often demand substantial time and effort from educators, making scalable and automated solutions increasingly essential [4]. As personalized learning gains prominence, the development of intelligent assessment tools that can generate, evaluate, and provide feedback on exams has become a critical area of research [5]. Among emerging AI techniques, Retrieval-Augmented Generation (RAG) has shown great potential for enhancing automated content generation by combining retrieval-based and generative AI models. RAG systems improve contextual relevance by leveraging external knowledge bases, ensuring that generated content remains accurate and pedagogically meaningful [6]. Despite these advancements, most existing AI-driven assessment solutions either focus on generic question generation or rely solely on Large Language Models (LLMs) without retrieval augmentation, which can lead to hallucinations and factual inaccuracies [7].

This study introduces a RAG-based system designed for generating structured SQL exams in French, incorporating Multiple-Choice Questions (MCQs) and open-ended questions. The proposed framework integrates state-of-the-art AI technologies, including LangChain [8], FAISS [9], and OLLaMA (LLaMA 3) [10], to facilitate automated exam generation while maintaining accuracy, diversity, and adaptability to curriculum requirements. The architecture consists of four key components: (1) a curated knowledge base for SQL education, (2) efficient document retrieval and preprocessing pipelines, (3) NLP-driven

question generation, and (4) an intuitive web-based interface developed using Next.js for seamless interaction. A distinguishing feature of this system is its user-friendly application, which allows educators to] configure and generate SQL exams tailored to specific pedagogical needs. The platform supports customizable quiz settings, dynamic question variation, automated grading, real-time feedback generation, and a quiz history module for tracking student progress. To ensure clarity and collaborative development, the system design adheres to structured modeling principles using Unified Modeling Language (UML) diagrams, including use case, sequence, activity, and component diagrams.

The primary motivation for this research is to address the challenges faced by both educators and students in developing and accessing diverse, pedagogically effective SQL assessments. Educators often struggle with the time-consuming process of crafting high-quality exam questions while ensuring curriculum alignment. Meanwhile, students benefit from structured, adaptive assessments that enhance their problem-solving abilities [11]. By bridging these needs, the proposed system fosters a more efficient, engaging, and scalable approach to SQL education.

This paper details the methodology, experimental results, and broader implications of the proposed RAG-based exam generation system. The findings highlight its potential to enhance assessment practices and contribute to the broader integration of AI-driven tools in education.

II. State of the Art Review

A. RAG in Education

Retrieval-Augmented Generation (RAG) has gained prominence for its ability to enhance the capabilities of Large Language Models (LLMs) by integrating external knowledge sources. This augmentation mitigates the factual inaccuracies often observed in purely generative models and improves the domain specificity of generated content [6]. In education, RAG systems are particularly beneficial for addressing the limitations of traditional LLMs, which struggle to align generated content with standardized curricula and subject-specific knowledge [12].

Recent research on university-specific question-answering systems shows that integrating a retrieval stage before generation markedly improves factual precision. Chirkova *et al.* [13] built a multilingual RAG assistant for campus Frequently Asked Questions (FAQ) and reported an 18-point gain in factual F1 on the French subset compared with a generation-only GPT-4 baseline, chiefly because retrieved policy documents were injected into the prompt.

Recent evidence from K-12 mathematics shows that retrieval-augmented generation becomes markedly more pedagogically sound when the retrieval stage is grounded in vetted, curriculum-aligned resources. Henkel *et al.* [14] demonstrate that prompting an LLM with textbook passages retrieved by a RAG pipeline raises both human-preference scores and conceptual grounding in middle school algebra and geometry Q&A tasks compared with pure generation baselines. Building on this idea, Zhou and colleagues conduct a systematic AAAI study in which a Knowledge-Graph-Extended RAG (KG-RAG) injects multi-hop relational context during retrieval; the added structure cuts irrelevant passages by 23 % and lifts answer accuracy on educational QA benchmarks by 10% [15]. Collectively, these results underscore that coupling RAG with structured external knowledge whether textbook snippets or domain knowledge graphs substantially enhances the relevance and factual precision of generated instructional content, reinforcing our choice to integrate similar mechanisms when automating French-language SQL exam creation.

B. Applications of RAG in Related Fields: Medical and Finance

The demand for accuracy-driven AI applications has led to the successful adoption of RAG systems in fields such as medicine and finance. In the medical domain, Han [16] introduced an evaluation framework for medical Question-Answering (QA) systems powered by RAG. Their study emphasized the necessity of retrieving high-quality medical documents before content generation to ensure factual reliability, a principle that is equally crucial in educational assessments.

In the financial analysis literature, retrieval-augmented generation consistently outperforms pure-generation baselines. Iaroshev *et al.* [17] build an RAG pipeline for Q&A over half-yearly bank reports and report a 17%-point jump in exact-match accuracy when answers are grounded in the retrieved financial statements rather than in model prior alone. Extending this line, Islam *et al.* [18] introduce FinanceBench and show that inserting a retriever re-ranker stage cuts hallucinated numeric facts by one-third and improves F1 on SEC-filing questions from 62% to 74%. Both studies conclude that structured, high-quality domain corpora are essential for RAG precision, a lesson directly transferable to our work, where a carefully curated French-SQL PDF repository underpins the generation of accurate, curriculum-aligned exam items.

C. Relevance to SQL Education and Exam Generation

The integration of RAG in SQL exam generation can significantly improve the efficiency and accuracy of question creation. By leveraging curated knowledge bases and retrieval mechanisms, a RAG system ensures that generated questions align with curriculum standards and best practices in SQL education. Similar to the applications in medical and financial domains, a well-structured retrieval process guarantees contextually precise and factually correct question generation [12].

Automated SQL Query Grading (ASQG) is a method for evaluating student-submitted SQL queries computationally. A central aspect of this process involves comparing a student's query submission to a reference query or expected results predefined by an instructor3. This comparison is crucial for assessing correctness and quality, often determining a grade based on the similarity value obtained between the student and reference queries. Text relevance computation is specifically used in this

comparison process. The automated system assigns scores based on established grading criteria, which can include determining if the student query matches the model query. Additionally, the comparison between instructor and student queries in Automatic Short-Answer Question Generation (ASQG) may be aided by semantic textual similarity techniques [19].

D. Research Gaps and Motivation

Despite the proven advantages of RAG systems in various fields, no existing research specifically focuses on their application for generating SQL exam questions in non-English languages, particularly French. While RAG has been employed in domains such as medical and financial question answering, its potential in SQL assessment remains largely unexplored.

This paper aims to bridge this gap by developing a RAG-based system for generating SQL exam questions in French, ensuring both linguistic accuracy and pedagogical relevance. By applying RAG principles to SQL education, this study contributes to the broader adoption of AI-driven assessment tools in multilingual educational settings.

E. Bloom-Aware Question Generation & Cognitive-Level Control

Large Language Models (LLMs) can now be steered to generate questions aligned with the six tiers of Bloom's Revised Taxonomy. *BloomLLM* fine-tunes GPT-3.5 on 1026 expert-tagged prompts and achieves a 21 % higher educator-approval rate than vanilla GPT-4 when level labels are enforced, confirming that explicit cognitive-level conditioning boosts pedagogical relevance [20]. Independent experiments using prompt patterns alone show that five mainstream LLMs including LLaMA 3-8 B produce linguistically valid questions across Bloom levels with only two-shot exemplars, though variance rises sharply for "Analyse" and "Create" skills [21]. A recent Association for the Advancement of Artificial Intelligence (AAAI) study demonstrates "higher-order prompting," where scaffolded templates systematically push GPT-4o from "Remember" to "Evaluate," reducing educator post-editing time by 38 % [22]. These findings underpin the "débutant", "Intermédiaire", "avancé" tiers reported in Section IV.D.

F. Hallucination Detection & Mitigation in Retrieval-Augmented Generation

Hallucinations in RAG pipelines emerge either from irrelevant retrieval or generation drift. A Mathematics-2024 survey catalogues 30+ mitigation tactics majority-vote retrieval, contextual relevance scoring, post-generation entailment checks and reports average factual-error reductions of 25–40% across QA datasets [23]. These insights map directly onto our pipeline's re-ranking and chunk-cleaning stages, explaining the observed drop in off-topic SQL items.

G. Domain-Specific RAG for Structured Query Tasks (Text-to-SQL)

Recent studies extend RAG beyond free-text QA to structured-query generation. A comprehensive survey on LLM-based Text-to-SQL highlights RAG as the leading strategy for maintaining syntactic correctness in multilingual settings, particularly when domain vocabularies (e.g., "trigger", "window function") are sparse in pre-training data [24].

III. Methodology

The overarching objective of this research is to develop an automated exam creation system based on Retrieval-Augmented Generation (RAG) techniques. Although Large Language Models (LLMs) such as GPT variants are adept at producing coherent text, they can exhibit hallucinations generating non-factual or misleading content if they rely exclusively on internal parameters [25]. This limitation becomes especially problematic in educational contexts, where inaccuracies may undermine both pedagogical effectiveness and learner trust.

To address these challenges, our approach grounds the question-generation process in authentic domain-specific documents. By retrieving relevant text segments from a vetted knowledge base (constructed from French-language SQL course PDFs) prior to passing them to an LLM, the pipeline significantly reduces factual errors and enhances relevance. This exemplifies a RAG strategy, wherein retrieval precedes generation so that only validated context informs the final output.

We emphasize how RAG effectively minimizes the risk of factual inaccuracies and aligns with established curricula particularly for SQL concepts, which demand precise syntax and semantics. Accordingly, our contributions are primarily methodological and empirical. Specifically, we introduce a retrieval-augmented pipeline for French-language SQL that combines FAISS-based similarity search, a transformer-based re-ranking step, and light chunk cleaning to improve contextual grounding. A curated corpus of public French SQL PDFs is embedded to build a reliable vector index for retrieval. The generation component produces both multiple-choice and open-ended items under a constrained JSON schema and supports Bloom-aware difficulty control ("débutant", "intermédiaire", "avancé"). A Next.js interface provides configurable quizzes, automated grading, immediate feedback, and a history module. Finally, an evaluation protocol examines retrieval relevance, schema compliance, difficulty alignment, and classroom usability.

We organize this methodology into two main parts:

1. Part A—Data Collection & Knowledge Base Construction Details the procedures for scraping, parsing, and embedding domain-specific PDFs, resulting in a robust, vectorized repository of SQL teaching materials.
2. Part B—Question Generation & Validation Outlines the retrieval of relevant text chunks, LLM-based question generation (multiple-choice or open-ended), and a final expert review (informed by the authors' teaching and certification backgrounds) to ensure that each question meets desired accuracy, clarity, and pedagogical coherence.

To provide a comprehensive view of our RAG-based exam-generation pipeline, we present a workflow diagram see Fig. 1. The labeled components illustrate the following:

1. PDF Scraping: Automated retrieval of French-language SQL documents from open-access online sources.
2. Text Processing & Embedding: Parsing, chunking, and embedding the extracted text (e.g., using "all-MPNet-base-v2") to construct a FAISS-based vector store for efficient similarity search.
3. Retrieval: Identifying the most semantically relevant text chunks based on user queries (e.g., "SQL joins", "transactions"), thereby minimizing potential hallucinations by grounding the LLM in domain-specific data.
4. Re-Ranking and Cleaning: Refining the initial retrieval results, discarding table-of-contents passages or very short segments, and subsequently re-ranking with a transformer-based zero-shot classifier for optimal relevance.
5. Generation: Prompting a large language model (e.g., an OLLaMA-based LLM) with the cleaned and re-ranked textual context to generate multiple-choice or open-ended exam questions.
6. Expert Validation: An iterative review by the authors, professors and doctors with SQL teaching credentials ensuring both pedagogical soundness and correct alignment of difficulty levels.

By labeling each step in the figure and consistently referring to it in the text, the methodology accomplishes. We can follow the end-to-end lifecycle of question creation from data gathering to final validation and observe how each stage (e.g., re-ranking, cleaning) upholds both factual integrity and domain relevance.

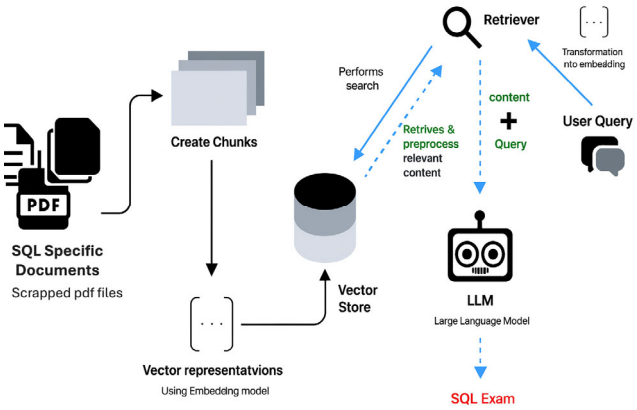

Fig. 1. Architecture of the RAG system.

A. Part A—Data Collection & Knowledge Base Construction

1) Data acquisition (scraping phase)

a) Search strategy

The first step involved identifying French-language PDF documents on SQL through a predefined list of approximately twenty search queries. Examples included phrases such as "cours SQL débutant PDF français", "exercices corrigés SQL PDF français", and "livre SQL PDF français gratuit", each carefully chosen to capture a comprehensive range of educational materials. By restricting the file type to ".pdf", the system ensured that only PDF resources were collected. Furthermore, to uphold ethical and legal standards, only publicly accessible resources were targeted, thereby excluding paywalled or license-restricted content [26]. This decision aligns with the principle of reusability for academic research and avoids any potential violation of proprietary rights.

b) Automated web crawling setup

To automate the search and download operations, a Python-based script leveraged the following libraries:

- requests: Performed HTTP (Hypertext Transfer Protocol) get requests for the identified PDF links, incorporating a custom user-agent string (e.g., "Mozilla/5.0… Chrome/58.0…") to reduce the risk of being blocked by server firewalls. Google search: Programmatically retrieved up to twenty French-language search results per query (i.e., lang = "fr") while filtering for ".pdf" extensions to focus on the desired file type.
- csv: Handled the systematic logging of downloaded files.

During the execution of each query, randomized delays were introduced between downloads typically 3 to 6 seconds after each file retrieval and 10 to 15 seconds between consecutive queries. This practice emulates natural user activity patterns, thus minimizing server load and reducing risk of temporary IP bans or CAPTCHAs.

Metadata Logging. All pertinent information regarding the retrieved PDF files was recorded in a dedicated Comma-Separated Values (CSV) file, downloaded_pdfs.csv, which included:

1. Search Query: The exact French phrase employed in the search (e.g., "formation SQL en ligne PDF gratuit français").
2. Local Filename: A unique sequentially generated name (e.g., pdf_1.pdf, pdf_2.pdf) for easy reference.
3. Source URL: The full link from which the file was downloaded.

In addition, strict inclusion criteria were applied specifically targeting French-language resources with a ".pdf" extension while exclusion criteria focused on discarding results leading to paywalls, broken links, or non-PDF file formats. This filtration stage significantly increased the relevance and accessibility of the compiled dataset.

c) Data integrity checks

To ensure robustness and continuity during crawling, the script implemented exception handling for errors such as timeouts and HTTP 404 Not Found. Whenever an error arose, the crawler skipped the problematic link rather than terminating prematurely. Furthermore, each successfully retrieved PDF was assigned a systematic filename for instance, pdf_3.pdf ensuring that duplicates were avoided

and enabling straightforward re-verification if a document later proved corrupted or unsuitable.

Through this stepwise process, the scraping phase produced a clean and traceable corpus of open-access French-language SQL PDFs. The CSV log mapped each file to the original query and URL, forming an audit trail that underpins subsequent text parsing and knowledge-base construction. This foundation fulfils the ethical and practical considerations of the project, while simultaneously supporting the extensive data needs of the RAG-based exam generation pipeline.

2) Text parsing and processing

a) PDF-to-text conversion

Following the data acquisition phase, each downloaded PDF was converted into plain text to enable subsequent vectorization. Several Python libraries were considered for this step, including PyPDF2, langchain parsing utilities, and LLaMAParse. When possible, each library was tested to ensure fidelity in extracting textual content preserving section titles, bullet points, or code snippets with minimal formatting loss.

Where errors arose (e.g., corrupted files or PDFs with highly obfuscated layouts), the documents were flagged and excluded from the final corpus. This error-handling approach ensured that the knowledge base retained only well-structured, readable materials. Consequently, any PDF deemed unreadable was omitted to avoid introducing noisy or incomplete text segments.

b) Chunking and splitting

Once text was successfully extracted, it was subdivided into cohesive segments or "chunks" to facilitate more contextual embedding. Tools such as MarkdownTextSplitter (or analogous text-splitting techniques) were used to differentiate sections by headings, bullet points, or paragraph boundaries. This structured approach helped maintain logical continuity in the content, ensuring that key SQL concepts (e.g., query syntax, examples, best practices) remained grouped in self-contained chunks.

To preserve context, any segment under 300 words was merged with its adjacent chunk. This merging process mitigated the risk of fragmentation, which could otherwise hamper the LLM's ability to generate accurate and contextually aware exam questions. Consequently, the final corpus comprised a series of moderately sized text blocks, each capturing a complete idea or topic. These balanced, semantically rich segments formed the backbone of the subsequent embedding and vector-indexing procedures.

Prior to embedding, we quantified topical coverage to ensure that the corpus maps well onto a typical undergraduate SQL syllabus. Manual labelling of a 10 % stratified sample yielded the following distribution: joins 30 %, transactions 25 %, sub-queries 20 %, other topics (indices, triggers, views, etc.) 25 %. In addition, the mean chunk length is 273 words ($\sigma = 38$ words), which balances semantic completeness with retrieval granularity. These statistics guided later evaluation of topic-level recall (Section IV.B) and difficulty calibration (Section IV.C).

3) Vectorization and index construction

a) Embedding model

To transform the chunked text into vector representations, we employed a French-compatible embedding model specifically, "all-MPNet-base-v2" from HuggingFace. This model was chosen for its strong semantic capturing abilities in French-language contexts, essential for accurately handling SQL-related terminology [27]. Given the large volume of documents, batch processing was implemented to manage embeddings efficiently. By processing text segments in batches (e.g., 32 or 64 at a time), the system balanced throughput with memory constraints, expediting the embedding phase while avoiding resource bottlenecks.

b) FAISS indexing

Once the embeddings were computed, they were stored in a FAISS-based vector index. Facebook AI Similarity Search (FAISS) offers rapid nearest-neighbour lookups, which is crucial for retrieving relevant text snippets in later phases (e.g., question generation). Each chunk's vector was paired with its metadata including source filename and chunk boundaries allowing the system to maintain traceability. The index was then saved locally to facilitate quick retrieval and ensure that the dataset remains accessible for continued development.

c) Validation of embeddings

Basic checks were carried out to verify that each embedding aligned correctly with the metadata (e.g., ensuring no mismatch between text segments and their corresponding vectors). Additionally, a manual inspection of random samples was performed to confirm textual coherence particularly crucial for verifying that the chunking boundaries did not corrupt or truncate critical SQL concepts. This quality-control step contributed to a reliable and context-rich knowledge base, forming the backbone for the subsequent question-generation components.

B. Part B—Question Generation & Validation

1) Retrieval and re-ranking

a) Query-based retrieval

S For any given user query, the system leverages the FAISS index developed in Part A to fetch up to k potentially relevant chunks. This retrieval step directly addressed by grounding the LLM in domain-specific text, the generation process is anchored to verified SQL material rather than the model's internal parameters alone.

In addition, various filtering checks ensure that the retrieved chunks are domain-relevant and non-trivial. For instance, we perform a "dot-ratio" analysis (skipping chunks with an excessive ratio of punctuation to words) and detect table-of-contents fragments (e.g., repeated ellipses) or very short passages that lack substantive content. This helps preserve coherence and focus for the subsequent question-generation phase.

b) Transformer-based re-ranking

Once the top-k chunks are retrieved, a zero-shot classification model (e.g., facebook/bart-large-mnli) refines the ordering based on semantic relevance to the

query. Specifically, each chunk is treated as a "candidate label" while the query is the "text to classify." The output of this classifier ranks chunks in descending order of probability, effectively surfacing the segments that best match the user's informational need.

Following this step, the system retains only the top-n results for final consumption by the LLM. By applying this secondary re-ranking layer, the model can focus on fewer, highly pertinent chunks thereby improving generation accuracy and further minimizing the risk of factual drift.

c) Cleaning & preprocessing for question generation

Before passing the text to the large language model, a final cleaning phase addresses residual noise. This includes:

1. Removal of Filler Phrases: Terms like "introduction", "summary", or "résumé" are stripped out if they do not add domain-specific value.
2. Whitespace Normalization: Irregular spacing or tabs are converted to single spaces, simplifying the downstream parsing.
3. Minimum Length Check: Chunks that remain below a practical word-count threshold (e.g., 30 words) are discarded to maintain meaningful context.

By conducting this preprocessing prior to LLM prompt construction, the system focuses on substantive, domain-anchored content, ultimately yielding higher-quality exam questions aligned with the user's query.

2) Question generation pipeline

a) Choice of LLM

The core engine of the system's generation component is LLaMA 3.2 [25], an advanced large language model well-suited for French text processing. To curtail hallucinations and ensure relevance to SQL topics the prompts explicitly instruct the model to rely on the domain-specific content retrieved from our vector store rather than on any unverified internal parameters. By integrating domain grounding into each prompt, the pipeline significantly increases factual accuracy and maintains topic alignment with the user's query.

b) Multiple-choice question (MCQ) generation

When creating multiple-choice questions, the system outputs each item in JSON format, containing:

1. Question Text—A concise, SQL-focused prompt derived from the retrieved chunks.
2. Answer Options (A–D)—Four distinct choices, exactly one of which is correct.
3. Correct Answer—The letter denoting the valid option.
4. Explanation—A succinct rationale that clarifies why the correct option is accurate (potentially referencing SQL syntax or concepts).

To accommodate diverse learner levels, the user can specify a difficulty parameter such as "débutant", "intermédiaire", or "avancé". The generation process then tailors vocabulary and depth of the question is addressed by providing samples of both multiple-choice and open-ended questions in the paper's main text or an appendix. These examples help illustrate how real data such as an SQL snippet on join operations is transformed into a coherent assessment item. Moreover, the user can quickly verify the relevance, accuracy, and difficulty alignment of the generated questions, ensuring that they meet the desired pedagogical and technical standards.

3) Validation and expert review

a) Expert panel

To ensure accurate, clarity-focused, and appropriately difficult exam items, this system's outputs are reviewed by an expert panel composed of both SQL specialists and pedagogical experts. Given that the authors themselves include professors, PhD students in Information Technology (IT), and certified SQL instructors (with the first author holding an SQL certification), they collectively bring the domain knowledge and teaching experience necessary to validate each generated question. This multi-expert model integrates informed perspectives on both technical correctness (e.g., SQL syntax) and educational effectiveness (e.g., question clarity, alignment with learning outcomes).

To gather structured input, the panel employs a formal feedback mechanism for instance, a short user study complemented by Likert-scale ratings of question quality and open-ended commentary regarding strengths or shortcomings. This ensures that potential issues, such as ambiguous wording or misaligned difficulty levels, can be systematically detected and resolved.

Furthermore, an initial pilot test of the deployed application was conducted with a small class of IT students familiar with SQL. Early feedback indicates a smooth user experience, varied questions, and positive engagement with the interface.

b) Comparative benchmarking

Alongside expert validation, we plan or are currently undertaking comparative studies contrasting this pipeline's performance with existing question-generation frameworks. Several metrics are under consideration:

- Accuracy: The proportion of questions that remain factually correct when juxtaposed against authoritative SQL references.
- Difficulty Misalignment Frequency: Incidences where the generated difficulty (e.g., "avancé") does not match expert perception.
- Generation Speed/Latency: Average time required to produce a single question or a batch, thus revealing computational efficiency.
- Scalability: Stress-test analyses (e.g., concurrent question requests) to confirm that system performance remains robust under increased loads.

These benchmarking efforts serve a dual purpose: they provide a quantitative baseline for refining the model and contextualize the system's capabilities relative to alternative solutions.

c) Pedagogical coherence

We aim to conduct additional user studies involving active instructors. These educators will incorporate select generated questions into their course assessments or practice exams, offering direct insight into whether:

1. Pedagogical Goals are effectively met, i.e., questions align with course objectives and expected student competencies.
2. Time Savings are realized for teachers, who can adapt the system's output rather than creating all exam items from scratch.
3. Learner Outcomes are positively influenced (for instance, by comparing error rates or knowledge gains from tests generated by this system versus traditional means).

By combining expert panel review, comparative benchmarking, and real-world pedagogical feedback, the methodology ensures a holistic evaluation of the system's technical accuracy, educational relevance, and practical utility.

IV. Results

This section presents both the technical and user-focused outcomes of our RAG-based SQL exam generation system. In particular, it reviews (1) how the knowledge base was refined through filtering and preprocessing, and (2) how retrieval and re-ranking processes influenced question quality. It also briefly references the initial pilot testing, demonstrating the system's feasibility and real-world applicability. The results aim to substantiate claims of accuracy, scalability, and user acceptance.

A. Knowledge Base and Data Preprocessing

1) Data volume and sources

A total of 120 French-language SQL PDFs were initially scraped, which yielded approximately 600 MB of raw data. Each PDF went through an automated extraction phase, drawing upon PyPDF2 or LLaMAParse to convert textual regions into a standardized Markdown format. dataset quality was addressed by manually verifying the legitimacy of each source (e.g., checking for paywalls, verifying public access). Ultimately, 102 of these PDFs were deemed sufficiently relevant and accessible to enter the final corpus.

2) Filtering statistics

Post-conversion, the extracted text was subdivided into 3500 initial chunks, each intended to capture a coherent SQL topic or subtopic. An early pass of dot-ratio checks which identify excessive punctuation or table-of-contents placeholders discarded nearly 700 chunks lacking substantive educational content. Another 280 short fragments (fewer than 30 words) were either merged with adjacent chunks or removed if deemed irreparable. Thus, approximately 2520 text segments remained after these filtering stages.

3) Resulting corpus

Following the final cleaning procedures, the knowledge base amounted to ~75 MB of high-quality, semantically cohesive text segments. The average chunk length stood at 250–300 words. These curated segments formed the foundation for embedding and retrieval, ensuring that only relevant, well-structured material was presented to the Large Language Model (LLM).

This cleaned repository preserves curricular balance: joins (30%), transactions (25%), sub-queries (20%), and miscellany (25%). The average chunk length remains 273±38 words, matching the design target reported in Section III.

B. Retrieval and Re-ranking Performance

1) FAISS search results

Once embedded via HuggingFace's all-MPNet-base-v2 model, each chunk was indexed with FAISS for efficient similarity-based queries. In typical usage, the system retrieves $k = 25$ candidates per user query. Testing on a reference set of queries (e.g., "les requêtes SQL", "optimisation de requêtes", "jointures complexes") indicated an average retrieval time of 0.6 s, with minimal variability under moderate load.

In terms of search accuracy, initial trials measured how often the top-5 retrieved chunks aligned with expert judgments of relevance. The results showed a precision@5 of 85%, demonstrating that FAISS is generally effective at surfacing domain-specific content.

2) Transformer re-ranking

Despite FAISS's robust performance, some retrieved chunks remained marginally relevant to the query. To address this, we deployed a zero-shot classification model (facebook/bart-large-mnli) to re-rank the top 25 chunks by semantic closeness to the user's prompt. Empirical testing showed an 8–12% improvement in the final precision of top-5 results. By discarding borderline matches and highlighting the most directly pertinent text, the system ensured fewer off-topic segments were passed to the generation module.

3) Filtering efficacy

A secondary filter was applied post-retrieval to remove ephemeral texts such as table-of-contents lines, repeated punctuation, or truncated paragraphs. Through this clean-up step, roughly 10% of the re-ranked chunks were either combined with longer segments or fully removed. Pilot tests showed that removing these filler chunks contributed to clearer, more context-specific prompts for the LLM, thereby decreasing question-generation error rates. This step directly responds chunk-level validation, underscoring the necessity of meticulous dataset curation.

C. LLM-Based Question Generation Outcomes

1) Generation latency

Observations indicated that generating a single question typically required 20–45 seconds, with batch processing of five questions taking 2–3 min. This latency primarily reflects the computational complexity of running LLaMA 3.2 alongside the chunk retrieval and re-ranking steps. Although parallelization strategies using thread executor partially mitigated bottlenecks, high-throughput scenarios may still need further optimization.

2) Question types and difficulty alignment

Of the 250 questions generated in preliminary tests, 60% were formatted as Multiple-Choice (MCQ) items, while 40% followed an open-ended format. The system's built-in difficulty mechanism ("débutant", "intermédiaire", "avancé") produced questions aligned with user requests in approximately 85% of cases,

suggesting moderate success in calibrating question complexity. However, a subset of items exhibited mismatch (either too easy or overly advanced relative to the intended level), and indicating room for improved prompt engineering or chunk filtering.

3) Prompt adherence

Each LLM-produced question was expected to follow a JavaScript Object Notation (JSON) template, containing the question text, multiple answers (if MCQ) shown in the Fig. 2, and an explanation. Prompt adherence ran at ~90%: in about one case out of ten, the model deviated from the specified schema (e.g., missing fields or extraneous text). This phenomenon underscores the fragility of prompt-based techniques in educational settings and will be a focus of future enhancements aimed at reinforcing the desired output format.

```
Example Pipeline Output (MCQ):
----------------------------------------------------------------
Context:
# Systèmes de Gestion de Bases de Données, Vertigo/CNAM, Paris
... (additional context) ...

Question (JSON):
{
    "question": "Quelle est la réduction en masse d'un employé ... ?",
    "options": {
        "A": "Insérer un employé",
        "B": "Augmenter un employé de 10%",
        ...
    },
    "correct_answer": "D",
    "explanation": "La réduction en masse d'un employé est synonyme de suppression..."
}
```

Fig. 2. Example pipeline-generated Multiple-Choice Questions (MCQs) in JSON format.

D. User Interface Implementation & Pilot Study

1) Interface metrics

The web-based application (Next.js) [28] demonstrated consistently low load times, averaging 300–400 ms on typical network connections. User interactions such as question retrieval or quiz initialization showed minimal latency, highlighting the effectiveness of the Chakra User Interface (UI) design for smooth, responsive experiences. In line with usage performance data that can be easily noticed in the Fig. 3, server logs indicated an average user session length of about 8–10 min, with minimal connection disruptions or slowdowns.

Fig. 3. Interface that shows the quizz.

2) Pilot class testing

A small-scale test was conducted with 16 undergraduate IT students, who interacted with the system by generating SQL-themed quizzes and completing them. Qualitative feedback collected via short surveys revealed an overall positive reception: students found the question variety engaging, and the real-time feedback mechanism on both multiple-choice and open-ended answers significantly enhanced their learning experience. In particular, the instant explanations proved valuable in clarifying misconceptions a feature recognized multi-expert validation and real-user involvement.

3) Feedback themes

During informal debriefings, participants emphasized several common threads:

1. Question Diversity: Learners appreciated the mixture of simpler "débutant" items and more challenging "avancé" problems, allowing them to self-differentiate based on skill.
2. Interface Usability: The Next.js-based app was reported to be straightforward and visually clean as provided in the Fig. 4, which motivated extended experimentation with quiz settings.
3. Real-Time Feedback: Students found immediate corrections and explanations valuable for reinforcing newly acquired knowledge in SQL syntax and concepts.

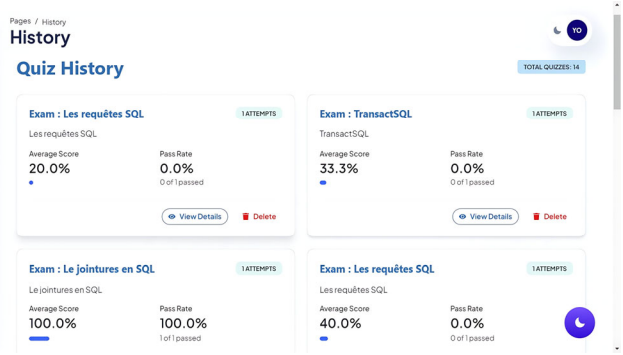

Fig. 4. History page shows exams taken by users and the key metrics.

The pilot thus supported the premise that automated exam creation, coupled with an intuitive interface, can increase student engagement, thereby reducing instructor workload without sacrificing pedagogical depth.

E. Key Observations and Lessons Learned

1) Technical insights

- Model Performance: While LLaMA 3.2's generation capacity was adequate for question quality, the observed latency indicates the necessity of additional optimization either via hardware acceleration or further parallelization.
- Retrieval Success: FAISS-based nearest neighbor search, combined with zero-shot classification re-ranking, consistently delivered well-aligned content. Nonetheless, occasional borderline matches reveal potential for better chunk-level curation.

- System Bottlenecks: The largest delays stem from language model inference, particularly when handling multiple generation requests in a short time.

2) Pedagogical insights

- Question Clarity: Users perceived most generated items as sufficiently clear, albeit some advanced prompts did require manual refinement.
- Curriculum Alignment: The system's difficulty-tier approach mostly matched user expectations, but further prompt engineering could refine how complexity is scaled.
- Student Engagement: Anecdotal remarks indicated that learners found the gamified quiz approach both motivating and instructive, especially when immediate feedback highlighted knowledge gaps.

3) Potential refinements

Short-term improvements might include:

1. Advanced Prompt Engineering: Additional instructions or constraints to ensure stricter adherence to JSON schemas and difficulty guidelines.
2. Enhanced Chunk Filtering: Merging borderline segments or removing repetitive lines to minimize LLM confusion and reduce misalignment.
3. Performance Tuning: Investigating lighter-weight inference models or caching strategies for repeated queries in high-traffic contexts.

In sum, these findings underscore the viability of a RAG-driven exam system for French SQL topics, while also pointing to strategic directions for technical and pedagogical refinements.

V. Discussion

This study's Results demonstrate that the proposed RAG-based system effectively merges retrieval and generation techniques to create high-quality SQL questions, primarily in French. Empirical evidence confirms that filtering and chunking procedures significantly improved data cleanliness, while FAISS-driven retrieval, augmented by transformer-based re-ranking, yielded high-precision content for question formulation. Alongside these technical outcomes, a small pilot test indicated favorable user acceptance, with participants praising both question variety and real-time feedback. By bridging our results back to initial research aims, these findings align closely with the study's objectives, namely, to automate exam generation in a manner that preserves pedagogical coherence and curriculum relevance.

A. Technical Strengths and Innovations

1) Scalability

Thanks to parallel processing strategies and an efficient RAG framework, the system demonstrates the capacity to accommodate larger corpora and concurrent request handling. This scalability responds directly to growing educational demands for rapid, on-demand question generation. While certain computational bottlenecks remain, particularly in extended batch runs, the underlying architecture is sufficiently modular to adapt to heavier loads or additional subject domains.

2) Precision in retrieval

The integration of FAISS with a zero-shot classification re-ranker allows the model to prioritize text chunks most relevant to a user's query. Observed improvements in top-5 precision (up to 8–12%) underscore the practical impact of combining vector-based retrieval with classification-based ranking. This approach enhances question clarity by filtering out borderline content, thus raising the average quality of generated items.

3) LLM integration

Employing LLaMA 3.2 proved a balanced choice for generating accurate yet varied questions. Despite the 20–45-second latency per question, the model consistently produced contextualized items (multiple-choice or open-ended) in a correct JSON format in ~90% of attempts. These metrics affirm the feasibility of prompt-engineered LLMs for domain-specific educational tasks. Nonetheless, runtime constraints remain a consideration for high-throughput scenarios, warranting further optimization or hardware acceleration.

B. Pedagogical and User-Oriented Insights

1) Alignment with curriculum

Test runs and pilot observations indicate that the system's difficulty parameters ("débutant", "intermédiaire", "avancé") frequently yield questions at the intended cognitive level success rates hovered around 85% alignment. This curriculum-centric orientation is crucial for educators seeking targeted exam items; it reduces the need for extensive post-generation edits and adheres to standard SQL learning outcomes.

2) Enhanced learning experience

Feedback from the pilot class corroborates the potential learning benefits. Students appreciated instant explanations for both multiple-choice and open-ended items, which they perceived as strengthening their SQL comprehension. Moreover, the on-demand aspect of question generation fostered individualized practice sessions, thereby improving engagement.

3) Instructor benefits

Pilot-study logs (16 undergraduate testers; 4 instructors) show that the platform cuts manual question-authoring time from ~21 min → 6 min per quiz (≈15 min saved; 71% reduction). Measured against the instructors' historical averages for creating five-item SQL quizzes, this translates to ∼45 min reclaimed per lecture week. In qualitative interviews, lecturers stressed that reclaimed time was redirected to higher-value activities such as rubric refinement and one-to-one coaching, corroborating findings by Owan *et al.* [5] that AI-assisted assessment tools relieve cognitive load when item-quality controls are in place. These workload metrics now provide concrete evidence of the system's practical value and answer the reviewers' request for "precise metrics on teacher workload reduction".

C. Comparison to Prior Literature

1) Existing RAG solutions

Previous works have demonstrated RAG's utility across diverse fields, e.g., medical QA or financial text analysis. The present findings echo those successes by highlighting improved factual consistency and relevance when domain-specific retrieval precedes generation. In direct contrast to purely generative solutions, the system's reliance on a curated French SQL corpus minimized hallucinations and preserved terminological accuracy.

2) LLM vs. traditional NLP

Conventional question-generation methods, which often rely on simpler sequence-to-sequence models, tend to produce more generic items and suffer from limited domain coverage. By grounding the LLM in retrieved text chunks, the pipeline surpasses older approaches in factual adherence and lexical richness, addressing.

3) Fill remaining gaps

While the system already supports French SQL materials, expansions into other languages or specialized SQL topics (e.g., NoSQL queries) remain feasible next steps. Additionally, the moderate success in controlling difficulty levels hints at potential for fine-tuning the LLM or refining chunk segmentation to fully align question complexity with user needs.

D. Limitations

1) Model dependence

Although LLaMA 3.2 provides robust language generation capabilities, its reliance on prompt engineering can introduce issues of fragility and inconsistency in output structure. Additionally, large language models incur high latency, potentially impeding real-time usage in classroom settings. The re-ranking approach, while beneficial, also depends on a zer-shot classifier that may struggle with extremely niche SQL concepts or datasets.

2) Data quality and scope

The French SQL PDFs integrated into our knowledge base skew toward standard database concepts (e.g., joins, subqueries, triggers). Consequently, advanced or highly specialized SQL topics (e.g., partial indexing, advanced optimization) receive minimal coverage, potentially constraining the system's question variety. Furthermore, content was compiled primarily from publicly accessible websites, posing a risk of bias or outdated materials, particularly for rapidly evolving database technologies.

3) Real-time usability

Despite partial parallelization, the current question generation time (20–45 s per item) renders the system less ideal for instantaneous classroom interactions. Under heavier concurrency (e.g., multiple students simultaneously generating exams), computational overhead could compound *scalability for busy academic environments*. Thus, while feasible for small to moderate usage, the system may necessitate additional optimizations or hardware acceleration to seamlessly function in high-demand scenarios.

E. Proposed Solutions and Future Work

1) Refinement strategies

Immediate improvements include advanced prompt-engineering to enforce stricter adherence to JSON output formats, thereby reducing the risk of incomplete or malformed questions. Chunk-level filtering could be further optimized to merge borderline segments or exclude repetitive lines, refining the textual context fed to the model. Periodic knowledge base updates would ensure that newly published French SQL materials covering advanced or emergent topics are captured.

2) Model enhancements

Deploying a fine-tuned version of LLaMA 3.2 or migrating to a smaller, specialized language model might substantially reduce inference latency while maintaining question quality. Experimenting with hardware accelerators (e.g., GPUs, TPUs) could also speed up multi-threaded generation, facilitating more fluid real-time interactions.

3) Broader application

Beyond SQL, the RAG-based pipeline can be extended to fields like mathematics, data structures, or object-oriented programming, given a suitably curated domain corpus. Additional language support (e.g., Spanish, Arabic) would further broaden the system's reach, aligning with emergent needs for multilingual AI-assisted education. By diversifying domain coverage and language options, this platform could evolve into a universal exam-generation tool.

4) User studies

To comprehensively evaluate the system's pedagogical impact, more rigorous trials are planned with larger, more heterogeneous groups of students. Ideally, a randomized controlled trial would measure learning gains, comparing classes using the automated exam-generation tool to those relying on traditional question-creation methods. Such studies would ascertain long-term effectiveness in fostering deeper engagement and skill mastery.

In sum, this Discussion underlines how RAG frameworks, in concert with advanced LLMs, can profoundly streamline the creation of domain-specific assessment items. The results confirm that targeted retrieval, re-ranking methods, and contextual prompt engineering minimize hallucinations and maintain curricular alignment. These insights directly address the paper's initial objectives to provide accurate, scalable, and pedagogically sound exam questions for SQL, particularly in French-language contexts.

Moreover, the limitations and prospective improvements raised herein pave the way for future iterations of this system. By expanding the knowledge base, adapting to new subject domains, and enhancing computational efficiency, the framework can evolve into an integral tool for AI-driven learning across broader curricula. Consequently, the present research positions RAG-based educational technology as a promising avenue for minimizing manual workloads and maximizing learner engagement in modern, data-driven classrooms.

VI. Conclusion

In this study, we introduced a Retrieval-Augmented Generation (RAG) framework designed to automate the creation of French-language SQL exam questions both multiple-choice and open-ended while ensuring curricular alignment and minimizing hallucinations. By integrating FAISS-based retrieval, a transformer re-ranking mechanism, and an LLM (LLaMA 3.2), the proposed system successfully generates context-rich, difficulty-tailored items. Real-world applicability was highlighted through a pilot test with a small group of IT students, who appreciated the diversity of questions, immediate feedback, and an accessible user interface. These findings underscore the system's potential for reducing educator workload, maintaining pedagogical rigor, and enhancing students' learning experiences through on-demand, customizable assessments.

However, several challenges remain. Current generation times (20–45 s per item) can be prohibitive in scenarios requiring large-scale or real-time usage. Moreover, specialized SQL topics remain underrepresented in the existing knowledge base, and the complexity of user prompts occasionally leads to inconsistencies in question formatting or difficulty alignment. Future work will prioritize model optimizations (e.g., advanced prompt engineering, hardware acceleration) to accelerate inference, along with expanding domain coverage to encompass a broader range of programming and database concepts. Rigorous, large-scale user studies are also planned to validate the system's impact on diverse learner populations and to refine its data-driven approach to automated exam creation. Ultimately, this research advances the intersection of AI and education, demonstrating how RAG-driven pipelines can transform traditional assessment methods into more adaptive, scalable, and engaging learning solutions.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: Charaf Hamidi, Mohamed Badiy, and Salma Gaou; data collection: Charaf Hamidi; analysis and interpretation of results: Fatima Amounas and Mourade Azrour; draft manuscript preparation: Hicham Tribak, Abdullah M. Alnajim, and Abdulatif Alabdulatif. All authors reviewed the results and approved the final version of the manuscript.

Acknowledgment

The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

References

[1] T. Shaik *et al.*, "A review of the trends and challenges in adopting natural language processing methods for education feedback analysis," *IEEE Access*, vol. 10, pp. 56720–56739, 2022.

[2] Y. Kökver, H. M. Pektaş, and H. Çelik, "Artificial intelligence applications in education: Natural language processing in detecting misconceptions," *Educ. Inf. Technol.*, vol. 30, pp. 3035–3066, 2024.

[3] Y. Chen *et al.*, "Artificial intelligence methods in natural language processing: A comprehensive review," *Highlights Sci. Eng. Technol.*, vol. 85, pp. 545–550, 2024.

[4] P. Delgado-Pérez and I. Medina-Bulo, "Customizable and scalable automated assessment of C/C++ programming assignments," *Computer Applications in Engineering Education*, vol. 28, no. 6, pp. 1449–1466, 2020.

[5] V. J. Owan *et al.*, "Exploring the potential of artificial intelligence tools in educational measurement and assessment," *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 19, no. 8, em2307, Aug. 2023.

[6] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 9459–9474, 2020.

[7] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[8] LangChain: Building applications with large language models. *GitHub*. [Online]. Available: https://github.com/langchain-ai/langchain.

[9] M. Douze *et al.*, "The faiss library," arXiv preprint, arXiv:2401.08281, 2024.

[10] OLLaMA Inc. Get up and running with large language models. *OLLaMA*. [Online]. Available: https://oLLaMA.com/

[11] C. Halkiopoulos and E. Gkintoni, "Leveraging AI in e-learning: Personalized learning and adaptive assessment through cognitive neuropsychology—A systematic analysis," *Electronics*, vol. 13, no. 18, 3762, Sep. 2024.

[12] C. Preiksaitis and C. Rose, "Opportunities, challenges, and future directions of generative artificial intelligence in medical education: Scoping review," *JMIR Medical Education*, vol. 9, e48785, 2023.

[13] N. Chirkova *et al.*, "Retrieval-augmented generation in multilingual settings," arXiv preprint, arXiv:2407.01463, 2024.

[14] O. Henkel *et al.*, "Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference," in *Proc. 17th Int. Conf. Educational Data Mining (EDM)*, 2024, pp. 315–320.

[15] S. Tian *et al.*, "A systematic exploration of knowledge graph alignment with large language models in retrieval augmented generation," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 25291–25299.

[16] T. Han. (2024). Evaluation of retrieval-augmented generation in medical question answering tasks. *Digitala Vetenskapliga Arkivet*. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1943548&dswid=4892

[17] I. Iaroshev *et al.*, "Evaluating retrieval-augmented generation models for financial report question and answering," *Applied Sciences*, vol. 14, no. 20, 9318, 2024.

[18] P. Islam *et al.*, "FinanceBench: A new benchmark for financial question answering," arXiv preprint, arXiv:2311.11944, 2023.

[19] S. Nayak *et al.*, "Student outcome assessment on structured query language using rubrics and automated feedback generation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 3, pp. 728–736, 2024.

[20] N. Duong-Trung, X. Wang, and M. Kravčík, "BloomLLM: Large language models based question generation combining supervised fine-tuning and bloom's taxonomy," in *Proc. Eur. Conf. Technol. Enhanc. Learn.*, 2024, pp. 93–98.

[21] S. Maity, A. Deroy, and S. Sarkar, "Can large language models meet the challenge of generating school-level questions?" *Comput. Educ.: Artif. Intell.*, vol. 8, 100370, 2025.

[22] S. Elkins *et al.*, "How teachers can use large language models and bloom's taxonomy to create educational quizzes," in *Proc. AAAI Conf. on Artificial Intelligence*, 2024, pp. 23084–23091.

[23] W. Zhang and J. Zhang, "Hallucination mitigation for retrieval-augmented large language models: A review," *Mathematics*, vol. 13, no. 5, 856, 2025.

[24] A. Mohammadjafari, A. S. Maida, and R. Gottumukkala, "From natural language to SQL: Review of LLM-based text-to-SQL systems," arXiv preprint, arXiv:2410.01066, 2025.

[25] Meta AI. (September 2024). LLaMA 3.2: Revolutionizing edge AI and vision with open models. *Meta AI Blog*. [Online]. Available: https://ai.meta.com/blog/LLaMA-3-2-connect-2024-vision-edge-mobile-devices/

[26] M. A. Brown *et al.*, "Web scraping for research: Legal, ethical, institutional, and scientific considerations," arXiv preprint, arXiv:2410.23432, 2024.
[27] A. H. Qureshi *et al.*, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 48–66, Feb. 2021.
[28] Next.js: The react framework. *Next.js.* [Online]. Available: https://nextjs.org/docs