ISSN: 2473-4901
v11 n6430

AI-Powered Study Assistant for Exams: QuizAI

Thi Hong Anh Nguyen
nguyenthihonganh@cityuniversity.edu
MS in Computer Science (MSCS)

Sam Chung
chungsam@cityu.edu

School of Technology & Computing (STC)
City University of Seattle (CityU)

Abstract

The advancement of Artificial Intelligence (AI) has created numerous opportunities to enhance the self-learning experience. Automated quiz generation has attracted attention as a way to support self-learning, particularly in digital education environments. However, existing systems often face key limitations: they frequently lack references, fail to provide meaningful feedback, and do not adapt content to individual needs. These shortcomings hinder the systems' effectiveness in promoting deep understanding and engagement. This paper introduces QuizAI, a novel AI-powered quiz generation system designed to address these issues. QuizAI can generate multiple-choice questions (MCQs) and open-ended questions based on user-provided sources, including PDF files and web pages. By utilizing Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs), the system ensures that questions are grounded in their source material and accompanied by supportive feedback. While QuizAI demonstrates comparable latency in generating dynamic content, it has produced a 50% duplication rate in five test documents, particularly when processing a larger number of questions per request. Despite these limitations, QuizAI achieves moderate to strong performance in terms of response time and question diversity, effectively complementing existing systems in the field. The proposed system enhances learning outcomes, supports recollection, and helps reduce learners' anxiety in self-paced educational settings.

Keywords: Quiz Generator, AI-Quiz Maker, RAG, Personalized Practice, LLM-Powered Learning, Knowledge Evaluation

GitHub: https://github.com/hongeinh/quizai.git

ISSN: 2473-4901
v11 n6430

AI-Powered Study Assistant for Exams: QuizAI

Thi Hong Anh Nguyen and Sam Chung

1. INTRODUCTION

Continuous learning is essential for success, especially in today's rapidly changing digital landscape (Hennekam, 2015). London and Smither (1999) emphasize that empowered self-development and continuous learning are crucial for individuals to adapt to new environments, enhance their skills, and achieve personal growth. With the rise of technology and the availability of vast amounts of data, numerous learning platforms and applications have emerged to help learners acquire new skills and specific knowledge, often without the need for formal education. This allows learners to study at their own pace while following a structured pathway.

However, despite the abundance of educational tools, many are primarily focused on subjects such as linguistics and mathematics. In contrast, the increasing demand for tech-related skills, particularly in fields like computer science, has exposed a gap in the availability of learning resources for technology-focused areas. As the industry continues to evolve, a growing number of learners are seeking to build or reinforce their technical knowledge to remain competitive in the job market.

Laguna et al. (2021) indicate that there is growing interest in skill enhancement within the tech industry. This increasing demand is driven by individuals who are looking to improve their employability or address skill gaps through industry certifications and post-secondary credentials.

Supporting this trend, Ehlinger and Stephany (2023) conducted a large-scale analysis of over eleven million job postings across the UK from 2018 to mid-2024. Their findings reveal that employers are placing greater emphasis on specific, demonstrable skills rather than traditional academic degrees, especially in the field of Artificial Intelligence (AI).

Many professionals are motivated to pursue certifications, but they often face significant challenges when preparing for exams. One major issue is the sheer volume of content, which includes textbooks, online articles, and technical documentation. Learners must sift through complex materials to identify key concepts within tight time constraints, which can be overwhelming. This challenge is particularly tough for individuals with weaker academic backgrounds, who may struggle to succeed in virtual learning environments due to a lack of interaction and engagement. As noted by Baum and McPherson (2019), students who are less academically prepared are especially vulnerable in fully online courses.

While resources such as practice tests and study groups are available, they often do not provide the personalized and adaptive revision that many learners require. Most current tools lack intelligent, on-demand question generation and do not offer actionable feedback. According to Patterson et al. (2024), most digital assessment platforms provide limited feedback and insufficient support for open-ended questions, which are crucial for evaluating deep understanding.

Additionally, popular platforms like Duolingo, Socratic, and ALEKS are effective in their respective fields, but they do not meet the specific needs of learners preparing for technical certifications. These tools primarily focus on general education and do not cover the foundational knowledge required in areas such as software development, data science, or cybersecurity.

To foster better engagement, Hirulkar and Athawale (2024) introduced a gamified multiple-choice quiz generator, promoting a competitive and collaborative learning environment. While effective in boosting participation, the system lacks essential features for personalized revision and alternative question types beyond multiple-choice questions (MCQs).

To overcome these limitations, this paper introduces an AI-powered quiz generator that dynamically creates personalized quizzes based on the learner's study materials, particularly PDFs and websites. By extracting key concepts and topics from the content, the system generates relevant multiple-choice questions that adapt in real time based on the learner's performance.

As users engage with the quizzes, the system adapts the difficulty of the questions and targets specific areas based on the frequency of correct and incorrect answers. This feature allows for focused revision by emphasizing the concepts that the learner struggles with the most. Additionally, the tool provides immediate and detailed feedback, including contextual explanations and direct references to the source material. This approach not only encourages active and targeted learning but also empowers users to track their progress and effectively address any knowledge gaps. By transforming static content into interactive, personalized assessments, the system aims to enhance understanding and retention.

2. BACKGROUND

Assessment is a vital component of a student's learning journey, helping to evaluate knowledge and reinforce learning through active recall and feedback. Among the various assessment methods, quizzes prove to be an effective tool for enhancing memory retention and engagement. They create a low-pressure, interactive environment that encourages students to reflect on their understanding of the material. However, effective learning relies not only on the format of the assessment but also on the quality of the feedback provided. Supportive feedback has been shown to positively influence students' mental preparation and self-assessment (Patterson et al., 2024). Incorporating mechanisms that simulate human interaction can enhance students' learning experiences and progress.

Recent advancements in Artificial Intelligence (AI), particularly in Large Language Models (LLMs), have facilitated the generation of questions. Several studies have developed systems capable of creating multiple-choice and open-ended questions from text inputs (Kurdi et al., 2019; Mulla & Gharpure, 2023; Das et al., 2021; Hirulkar & Athawale, 2024). Despite these advancements, existing systems often lack effective feedback mechanisms and tend to generate generic questions due to the limited availability of datasets.

3. RELATED WORK

The necessity for upskilling has become increasingly critical, particularly in the tech and AI industries. Laguna et al. (2021) emphasize that hiring managers prioritize technical skills when evaluating candidates, making certifications an essential complement to project experience and work history. Similarly, Ehlinger and Stephany (2023) note a shift towards skill-based hiring in the AI and green technology sectors, where digital certifications serve as important proof of an individual’s technical abilities. Together, these studies highlight the growing demand for skill development through certifications.

Although online courses are accessible, Baum and McPherson (2019) argue that human interaction and engagement are essential for achieving positive learning outcomes. To improve the learning experience, they recommend incorporating feedback, competition, and collaboration as key features of these platforms. Supporting this idea, Patterson et al. (2024) demonstrate that providing real-time, supportive feedback during tests can help reduce anxiety and enhance students' self-assessment. These studies suggest that learning platforms should integrate encouraging feedback to maintain students' motivation and improve learning efficiency.

One effective way to enhance student engagement is by using quizzes. Yang et al. (2021) found that testing, including quizzes, significantly boosts academic achievement in various educational settings. Additionally, El-Hashash (2022) demonstrated that weekly quizzes help to improve both attendance and engagement among students. These findings indicate that incorporating quizzes can be a successful strategy for promoting active learning.

Recent advancements in AI have facilitated the automatic generation of questions. Comprehensive reviews by Kurdi et al. (2019) and Mulla and Gharpure (2023) highlight the latest methodologies in question generation, focusing on the integration of feedback, control of question difficulty, and the use of neural architecture such as sequence-to-sequence (seq2seq) models. Das et al. (2021) introduced a system that generates and evaluates subjective questions, promoting deeper understanding rather than mere recall from students. Additionally, Hirulkar and Athawale (2024) developed an AI-based quiz generator that creates multiple-choice questions based on selected topics and difficulty levels. However, the broad selection of topics may limit the effectiveness of targeted learning.

Recent studies are exploring the adjustment of question difficulty in educational settings. Alkhuzaey et al. (2023) conduct a systematic review of methods for predicting the difficulty of generated questions and advocate for

ISSN: 2473-4901
v11 n6430

personalized quiz adaptations that are based on students' performance. Tomikawa et al. (2024) utilize transformer models along with Item Response Theory (IRT) to dynamically manage question difficulty. Additionally, Fu et al. (2025) introduce the ConQuer framework, which uses external knowledge to ensure that question generation is grounded, though the system is limited by predefined concepts.

Effective feedback mechanisms are crucial for online educational tools. Tobler (2023) discusses a generative AI-based system for automated grading, addressing challenges such as LLM hallucinations and the need for verified references. Additionally, Quizzio, Junior (n.d.) offers personalized feedback and tracks user progress, although it is limited by the length of user-submitted text. Together, these works emphasize that feedback should be immediate, relevant, and accurate to enhance learning outcomes.

Numerous educational resources exist in unstructured formats, such as PDFs, which necessitate effective methods for content extraction. Siegler (2025) introduces LlamaParse, a tool designed to parse complex documents into usable formats suitable for Retrieval-Augmented Generation (RAG) systems. Additionally, Google Cloud (n.d.) outlines strategies for semantic search and content extraction within its RAG pipeline for PDFs.

Despite significant advancements, current AI-powered educational tools still face notable limitations. These systems often rely on predefined content, lack interactivity, and fail to offer personalized experiences. Furthermore, the evaluation of answers is not yet perfect, as pointed out by Tomikawa et al. (2024), due to limitations in the datasets used. Future systems must accurately analyze diverse content sources and generate meaningful assessments while providing real-time, positive feedback to enhance learning engagement.

4. APPROACH

Figure 1 outlines the main use cases available to users of QuizAI. Users will be authenticated and authorized to upload documents in either PDF or URL format. They can generate quizzes based on the uploaded documents and access previously created documents and quizzes.

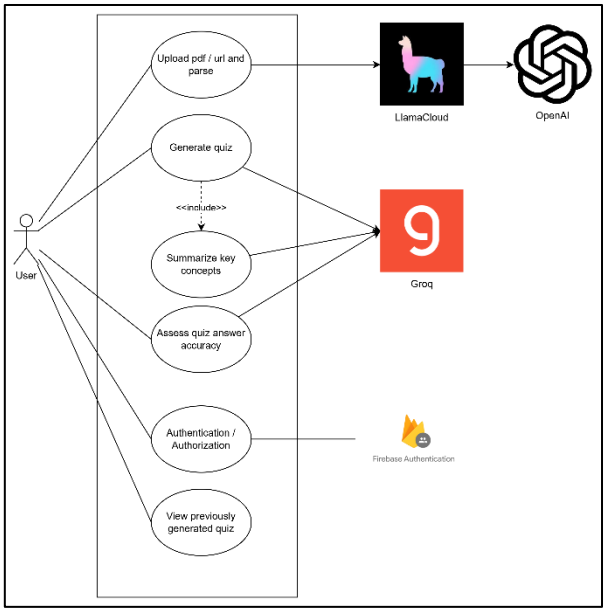

Figure 1: QuizAI Use Case Diagram

4.1. User Requirements

QuizAI requires users to authenticate before accessing its core functionalities. This authentication process protects users' privacy and digital footprints by ensuring that only authorized individuals can access their study materials and activity history. Additionally, it enhances data management and system integrity by preventing the misuse of resources.

Authentication is managed through Firebase Authentication, a service that provides identity management and access control for web and mobile applications. Firebase is a Backend-as-a-Service (BaaS) platform developed by Google, offering tools and services for building web and mobile applications. In QuizAI, users can create a new account using their email and password, or sign in using supported federated identity providers.

4.2. Design

The activity flow is outlined in Figure 2 in the Appendix. Users can either log in or register to access the core features. Once authenticated, users are able to upload a document in either PDF or URL format.

This input is then processed using LlamaParse (via LlamaCloud), a parsing service created to extract structured text and metadata from unstructured sources. Each page or section is extracted and saved to Firebase.

The parsed content is organized into smaller, semantically meaningful chunks using

ISSN: 2473-4901
v11 n6430

LlamaIndex (via LlamaCloud). This framework links these chunks to their corresponding embeddings. The embeddings are then stored in a vector database based on FAISS (Facebook AI Similarity Search). FAISS is an open-source library developed by Meta that enables efficient similarity searches for embeddings of multimedia documents. By creating searchable indices for these embeddings, FAISS allows the system to quickly retrieve semantically relevant chunks.

Once embeddings are stored, the system uses Retrieval-Augmented Generation (RAG), which integrates document retrieval with language generation to improve the process. RAG facilitates accurate and grounded question generation from the uploaded and parsed documents.

After the user has chosen the question type and number of questions, the system will retrieve relevant document chunks and generate questions. For open-ended questions, user answers are assessed to evaluate correctness and quality. The user receives feedback with direct citation to the original page or section.

Figure 3, the Class Diagram located in the Appendix, outlines the main entities stored in the database. The User collection holds the user's metadata, while authentication information is securely managed by Firebase authentication and is therefore not included in the User collection. The Document collection contains metadata for documents and ensures that only the file owners can access the content. Lastly, the Question collection stores the questions generated by QuizAI, allowing users to revisit previously generated questions if they wish to attempt them again.

4.3. Implementation

The core REST API endpoints are described below:

- POST /pdf/upload - Accepts PDF input and stores the structured result.
- POST /html/upload – Accepts a webpage URL and stores the structured result.
- POST /process - Processes parsed content and indexes using LlamaIndex.
- GET /chunks/retrieve - Retrieves the most relevant chunk.
- GET /pdf/page - Returns the specific page or section from the original document.
- POST /quiz/generate - Generates multiple-choice or open-ended quizzes based on the document.
- POST /answer/evaluate - Evaluates user-provided answers for open-ended questions.

These APIs enable clear separation of concerns and make the backend extensible for any future additions.

4.4. Technologies Used

Table 1 lists the technologies used for implementation.

Frontend	ReactJS, TailwindCSS
Backend	FastAPI (Python)
Authentication & database	Firebase authentication & Firestore
AI/ML	LlamaCloud
LLM	Groq, model llama-3.1-8b-instant

Table 1: Technologies Used

4.5. Deployment architecture

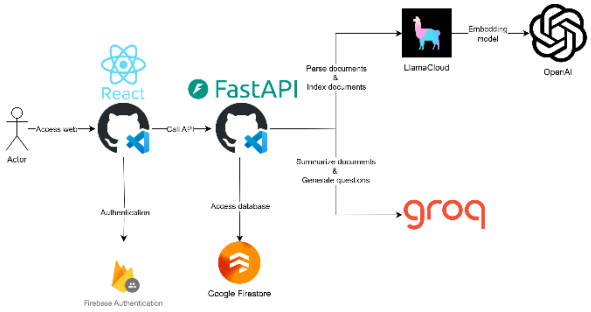

Figure 4: Deployment Diagram

The frontend and backend of the project can be deployed using GitHub Codespace, a cloud-based environment that provides access to the project's code and ready-to-use coding environments, as illustrated in Figure 4. This setup incorporates several tools and technologies.

For user authentication, Firebase Authentication is employed, supporting both the Email/Password method and third-party Google accounts. User data is stored using Firestore, a service from Google Cloud, which handles internal information, including user profiles, documents, and quizzes.

Quizzes are generated in collaboration with LlamaCloud for RAG, OpenAI for embeddings, and Groq for Large Language Model support. All external tools can be accessed via the API provided and secret keys.

5. DATA COLLECTION

Figure 5 in the Appendix illustrates the Input-Process-Output framework necessary for QuizAI.

Input:
Users are prompted to upload PDFs or provide URL links from which they want to generate quizzes.

Process:
Both PDF files and webpage links must be processed before use. Uploading a PDF will trigger LlamaParse to extract text, while URLs will require BeautifulSoup, a Python library used for parsing HTML and XML documents into plain text.

The extracted text is then cleaned by removing unnecessary content, such as headers, footers, references, and copyright information. SpaCy, a library for tokenization and named entity recognition, is employed to extract key concepts from the document, aiding in the cohesive generation of quizzes.

Output:
The cleaned text and the associated concept metadata are then input into LlamaIndex to create index objects, which are essential for future retrieval steps.

6. DATA ANALYSIS

To assess the performance and reliability of QuizAI, we use both qualitative and quantitative analyses, each focusing on different aspects of system reliability, efficiency, and learning effectiveness. Data is gathered by testing FastAPI endpoints through both automated and manual methods.

Quantitative Analysis:
The quantitative analysis aims to evaluate the efficiency of the system using the following metrics:
API Response Time: The response times for key endpoints—specifically, document processing, quiz generation using a large language model (LLM), and answer evaluation using an LLM—are recorded to calculate the average response time. Figure 6 in the Appendix illustrates the methodology used to calculate the API endpoint response time, while Figure 7 in the Appendix presents the actual response times for each API endpoint.

Question uniqueness refers to the ratio of duplicated to unique quiz questions within a document, which helps evaluate the diversity and relevance of the generated quizzes (see Figure 8 in the Appendix). On average, the repetition rate exceeds 50% for each generation. This issue is illustrated in Figure 9 of the Appendix, where the output of the LLM may contain several duplicated questions. There are two main reasons for this duplication:

1. The prompt given to the LLM does not require strictly unique questions and answer generation.
2. The summarization of the document limited the context available for the LLM, resulting in fewer unique questions being produced.

The quantitative analysis offers insights into system responsiveness and content diversity, which are essential for keeping users engaged.

Qualitative Analysis:
The qualitative analysis focuses on evaluating the accuracy and consistency of outputs generated by the LLM based on the following metrics:

Document Reference Accuracy: The generated questions are compared to the original document to assess their correctness. The Section ID provided by the endpoints (/quiz/generate) can be verified using both manual and automated methods, as illustrated in Figure 10 of the Appendix.

Scoring Consistency: Figure 11 in the Appendix shows how the LLM evaluates and scores a user's response to an open-ended question. Semantically similar user responses to the same question can be used to determine the reliability of the answer evaluation mechanism. Consistency is measured using the standard deviation of scores, as demonstrated in Figure 12 of the Appendix. Through testing, the system consistently assesses answers with an average score of 60 and delivers the same feedback for the same question and answer.
The quantitative analysis offers insights into the factual alignment, coherence, and reliability of the system's outputs.

7. FINDINGS

QuizAI demonstrates efficient processing across its core endpoints. The average response times for each endpoint are shown in Figure 7 in the Appendix. The results reveal that the system experiences moderate latency, with slightly longer processing times when handling large documents. However, it remains effective in maintaining user engagement, especially when compared to QuizMasterAI by Hirulkar and Athawale (2024), which has longer processed periods. In comparison to Quizzio (Junior), QuizAI maintains similar latency while generating dynamic content.

The quality of question generation is assessed based on the frequency of duplicate questions created in response to user requests. In testing five documents, the system produced a duplication rate of 50%, particularly when a higher number of questions were requested. This limitation arises primarily from two factors: (1) the current prompt does not ensure strict uniqueness, and (2) document summarization is truncated, limiting the context available for question generation. With a shorter summarization length, the process lacks enough information to generate a diverse set of questions.

In comparison, ConQuer by Fu et al. (2025) faces challenges with redundancy control and is constrained by predefined concepts. On the other hand, V-Doc by Ding et al. (2022) does not effectively address question diversity or system performance.

The current system is limited to PDFs and web URLs, excluding other commonly used formats such as Word documents, PowerPoint presentations, and various multimedia sources. Additionally, while answer evaluation is consistently scored for responses that are semantically similar, the current mechanism may not accurately capture subtle nuances.

Despite these limitations, QuizAI demonstrates moderate to strong performance in both response time and question diversity, making it a valuable complement to existing systems in the field. As a result, QuizAI serves as an effective tool for personalized learning and assessment.

8. CONCLUSION

This paper introduces QuizAI, an AI-powered quiz generation system aimed at helping students and learners master content through personalized assessments. By utilizing RAG and LLMs, QuizAI converts PDFs and webpage URLs into interactive quizzes for personal assessment.

QuizAI addresses several key challenges in self-study environments, such as content overload and the lack of direct engagement and feedback. Through both quantitative and qualitative analyses, the system demonstrates its ability to generate useful quizzes for recall and effectively assess users' answers. However, one challenge remains: question duplication, which is influenced by current prompt engineering and context truncation. Additionally, the research scope—specifically, focusing on an AI-powered study assistant for exams—prevented us from tackling pedagogical and ethical considerations.

Future efforts will concentrate on improving question duplication, refining answer evaluation scoring, and supporting a wider variety of content formats. Broader testing across diverse domains and user groups is necessary. We are also considering a refined user interface and user experience (UI/UX) along with an interactive game mode that allows users to compete against each other for enhanced engagement. Furthermore, we need to explore the pedagogical implications of QuizAI: How will it influence learners' critical thinking, engagement, and exam performance? What ethical considerations, such as AI bias and fairness in scoring, must we address?

9. REFERENCES

AlKhuzaey, S., Grasso, F., Payne, T. R., & Tamma, V. (2023). Text-based question difficulty Prediction: A Systematic review of automatic approaches. International Journal of Artificial Intelligence in Education, 34(3), 862–914. https://doi.org/10.1007/s40593-023-00362-1

Baum, S., & McPherson, M. (2019). The Human Factor: The Promise & Limits of Online Education. Daedalus, 148(4), 235–254. https://doi.org/10.1162/daed_a_01769

Das, B., Majumder, M., Sekh, A. A., & Phadikar, S. (2021). Automatic question generation and answer assessment for subjective examination. Cognitive Systems Research, 72, 14–22. https://doi.org/10.1016/j.cogsys.2021.11.002

Ding, Y., Huang, Z., Wang, R., Zhang, Y., Chen, X., Ma, Y., Chung, H., & Han, S. C. (2022). V-Doc : Visual questions answers with Documents. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 21460–21466. https://doi.org/10.1109/cvpr52688.2022.02083

Ehlinger, E. G., & Stephany, F. (2023). Skills or degree? The rise of Skill-Based hiring for AI and green jobs. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2312.11942

El-Hashash, A. (2022). Weekly Quizzes Reinforce Student Learning Outcomes and Performance in Biomedical Sciences in-course Assessments. Open Journal of Educational Research, 2(4), 168–178. Retrieved from

ISSN: 2473-4901
v11 n6430

https://www.scipublications.com/journal/index.php/ojer/article/view/273

Firebase Authentication. (n.d.). Firebase. https://firebase.google.com/docs/auth

Fu, Y., Wang, Z., Yang, L., Huo, M., & Dai, Z. (2025). ConQuer: A Framework for Concept-Based Quiz Generation. arXiv preprint arXiv:2503.14662.

Google Cloud. (n.d.). Parse PDFs in a retrieval-augmented generation pipeline. https://cloud.google.com/bigquery/docs/rag-pipeline-pdf

Hirulkar, S. R., & Athawale, P. S. V. (2024). Quiz Master AI: An Interactive Machine Learning-Based Quiz Generator. International Journal of Ingenious Research, Invention and Development (IJIRID), 3(5), 474–482. https://doi.org/10.5281/zenodo.14208814

Hennekam, S. (2015). Career success of older workers: the influence of social skills and continuous learning ability. Journal of Management Development, 34(9), 1113–1133. https://doi.org/10.1108/jmd-05-2014-0047

Júnior, S. (n.d.). Quizzio: AI Quiz Generator. Quizzio: AI Quiz Generator. https://www.quizzio.app/

Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2019). A systematic review of automatic question generation for educational purposes. International Journal of Artificial Intelligence in Education, 30(1), 121–204. https://doi.org/10.1007/s40593-019-00186-y

Laguna-Muggenburg, E., Bhole, M., & Meaney, M. (2021). Understanding Factors that Influence Upskilling. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2103.12193

London, M., & Smither, J. W. (1999). Empowered self-development and continuous learning. Human Resource Management: Published in Cooperation with the School of Business Administration, The University of Michigan and in alliance with the Society of Human Resources Management, 38(1), 3-15.

Mulla, N., & Gharpure, P. (2023). Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. Progress in Artificial Intelligence, 12(1), 1–32. https://doi.org/10.1007/s13748-023-00295-9

Parse PDFs in a retrieval-augmented generation pipeline. (n.d.). Google Cloud. https://cloud.google.com/bigquery/docs/rag-pipeline-pdf

Patterson, Tara & Romero, Margarida. (2024). Digital tests may be missing an opportunity to support student success: Exploring the effect of feedback during digital testing on student self-assessment and test anxiety management. 10.31237/osf.io/x4qr9

Siegler, R. (2025, January 30). RAG + LlamaPaRse: Advanced PDF Parsing for retrieval. Medium. https://medium.com/kx-systems/rag-llamaparse-advanced-pdf-parsing-for-retrieval-c393ab29891b

Tobler, S. (2023). Smart grading: A generative AI-based tool for knowledge-grounded answer evaluation in educational assessments. MethodsX, 12, 102531. https://doi.org/10.1016/j.mex.2023.102531

Tomikawa, Y., Suzuki, A., & Uto, M. (2024). Adaptive Question–Answer Generation with Difficulty Control Using Item Response Theory and Pre-trained Transformer Models. IEEE Transactions on Learning Technologies, 1–13. https://doi.org/10.1109/tlt.2024.3491801

Yang, C., Luo, L., Vadillo, M. A., Yu, R., & Shanks, D. R. (2021). Testing (quizzing) boosts classroom learning: A systematic and meta-analytic review. Psychological Bulletin, 147(4), 399–435. 1 https://doi.org/10.1037/bul0000309

Welcome to Faiss Documentation — Faiss documentation. (n.d.). https://faiss.ai/index.html

ISSN: 2473-4901
v11 n6430

APPENDIX

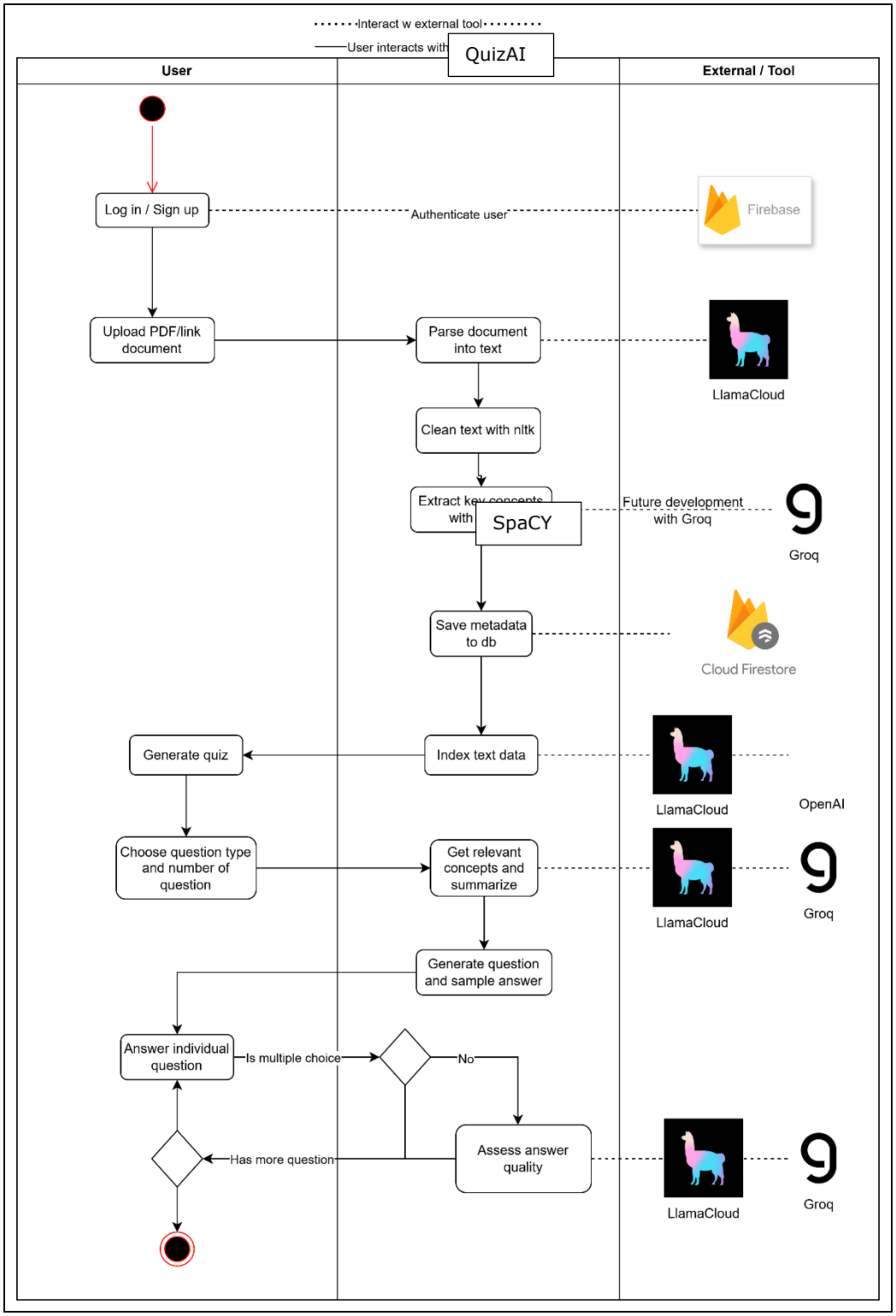

Figure 2: Activity Diagram

ISSN: 2473-4901
v11 n6430

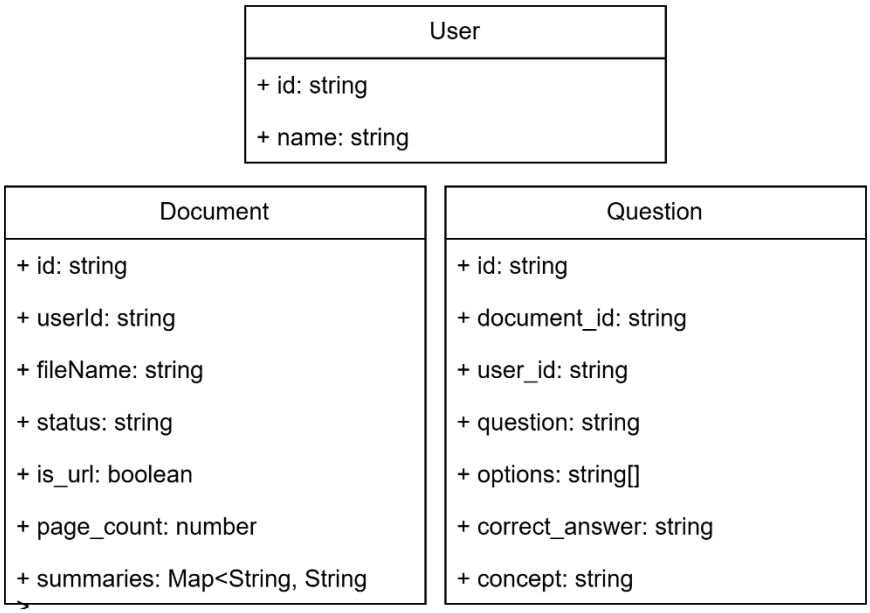

Figure 3: Class Diagram

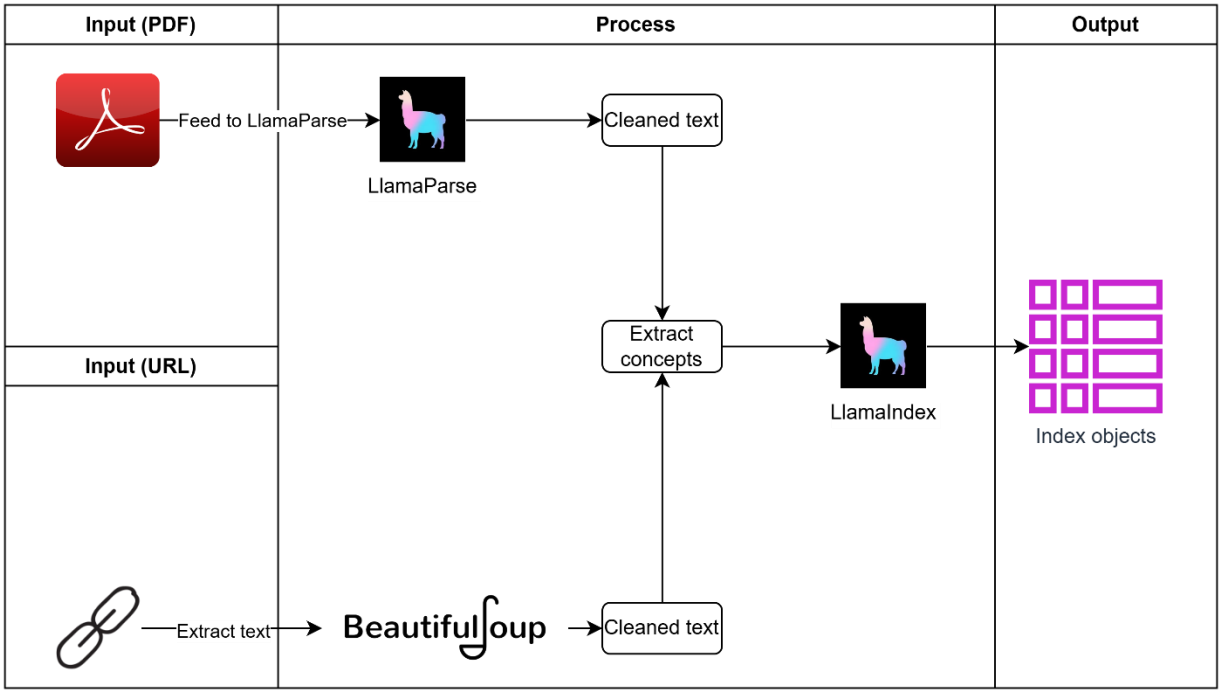

Figure 5: Input – Process - Output (IPO) Diagram

ISSN: 2473-4901
v11 n6430

```python
def measure_endpoint(endpoint: str, method: str = "GET", data: dict = None, files: dict = None, headers: dict = None):
    """Measure response time for a FastAPI endpoint."""
    start_time = time.time()
    try:
        if method == "POST":
            response = requests.post(f"{FASTAPI_URL}{endpoint}", json=data, files=files, headers=headers)
        else:
            response = requests.get(f"{FASTAPI_URL}{endpoint}", params=data, headers=headers)
        response.raise_for_status()
        duration = time.time() - start_time
        asyncio.run(log_metric({
            "endpoint": endpoint,
            "duration_seconds": duration,
            "status_code": response.status_code
        }))
        logger.info(f"Endpoint {endpoint} took {duration:.2f} seconds")
        return response.json(), duration
    except Exception as e:
        duration = time.time() - start_time
        asyncio.run(log_metric({
            "endpoint": endpoint,
            "duration_seconds": duration,
            "status_code": getattr(e.response, "status_code", 500),
            "error": str(e)
        }))
        logger.error(f"Error calling {endpoint}: {str(e)}")
        raise
```

Figure 6: Automated Script for Measuring API Response Time

Endpoint	Average time (s)
/pdf/process	2.064079999923706
/url/process	2.2693395614624023
LlamaIndex process	60
/quiz/generate (5 MCQ quizzes)	11.553621768951416
/quiz/generate (5 open-ended quizzes)	7.378465175628662
/answer/evaluate	5.7

Figure 7: API Response Time

```python
async def analyze_question_uniqueness(document_id: str, questions: list):
    """Count unique vs. duplicate questions."""
    question_texts = [q["question"] for q in questions]
    question_counts = Counter(question_texts)
    unique_questions = len([q for q, count in question_counts.items() if count == 1])
    duplicate_questions = len(question_texts) - unique_questions
    await log_metric({
        "document_id": document_id,
        "metric": "question_uniqueness",
        "unique_questions": unique_questions,
        "duplicate_questions": duplicate_questions
    })
    logger.info(f"Document {document_id}: {unique_questions} unique, {duplicate_questions} duplicate questions")
    return unique_questions, duplicate_questions
```

Figure 8: Automated Script for Measuring Uniqueness

ISSN: 2473-4901
v11 n6430

Figure 9: Repetitive Question Generation

ISSN: 2473-4901
v11 n6430

```python
async def analyze_page_references(document_id: str, questions: list, token: str):
    """Verify if question page references match document content."""
    correct_references = 0
    total_questions = len(questions)
    headers = {"Authorization": f"Bearer {token}"}
    for question in questions:
        page = question["page_reference"]
        query = f"Content from page {page} related to the question: {question['question']}"
        data = {"document_id": document_id, "query": query, "top_k": 1}
        response = requests.post(f"{FASTAPI_URL}/chunks/retrieve", json=data, headers=headers)
        response.raise_for_status()
        chunks = response.json()
        if chunks and chunks[0]["metadata"]["page"] == page:
            correct_references += 1
    accuracy = (correct_references / total_questions) * 100 if total_questions > 0 else 0
    await log_metric({
        "document_id": document_id,
        "metric": "page_reference_accuracy",
        "value": accuracy
    })
    logger.info(f"Page reference accuracy for {document_id}: {accuracy:.2f}%")
    return accuracy
```

Figure 10: Automated Script for Measuring Reference Accuracy

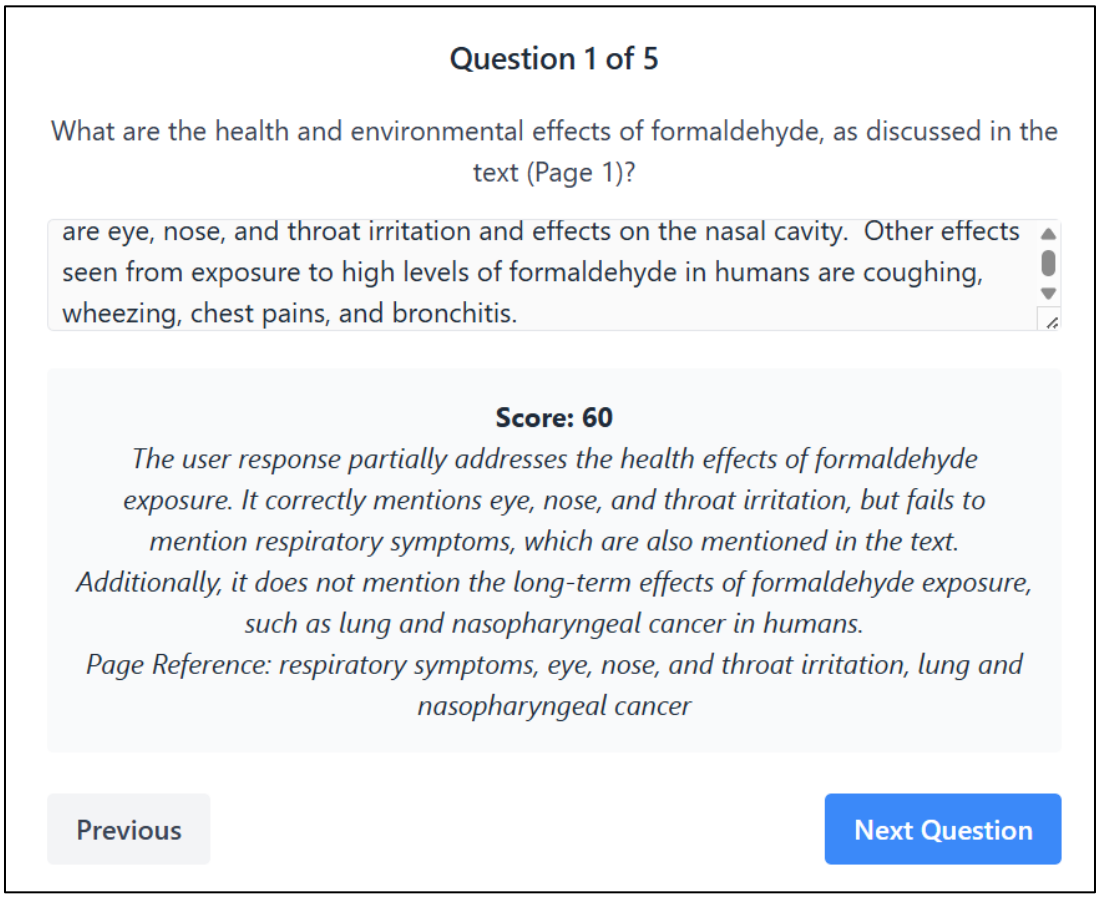

Figure 11: LLM Score for Open-Ended Question Evaluation

ISSN: 2473-4901
v11 n6430

```python
async def analyze_scoring_consistency(document_id: str, question: str, page_number: int, answers: list, token: str):
    """Test scoring consistency for similar answers."""
    scores = []
    headers = {"Authorization": f"Bearer {token}"}
    for answer in answers:
        data = {
            "document_id": document_id,
            "question": question,
            "user_answer": answer,
            "page_reference": page_number
        }
        response = requests.post(f"{FASTAPI_URL}/answer/validate", json=data, headers=headers)
        response.raise_for_status()
        validation = response.json()
        scores.append(validation["score"])
    variance = sum((s - sum(scores) / len(scores)) ** 2 for s in scores) / len(scores) if scores else 0
    await log_metric({
        "document_id": document_id,
        "metric": "scoring_variance",
        "value": variance
    })
    logger.info(f"Scoring variance for question in {document_id}: {variance:.2f}")
    return variance
```

Figure 12: Automated Script for Analyzing System's Consistency