AI-BASED PERSONALIZED QUIZ GENERATION

G. M. Nethmi Monali

(IT21360428)

B.Sc. (Hons) Degree in Information Technology Specialized in

Software Engineering

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

DECLARATION

I declare that this is my own work and this dissertation1 does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:		Date:	11.04.2025

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Signature of the supervisor:		Date:	11.04.2025

ABSTRACT

Advancements in artificial intelligence and natural language processing have introduced innovative methods for delivering personalized education. This research focuses on the development of an adaptive quiz generation system that dynamically tailors question difficulty and feedback based on individual learner performance. Leveraging OpenAI's language models and Bloom's Taxonomy, the system generates both Multiple Choice Questions (MCQs) and Short Answer Questions from academic lecture materials in PDF format. A hybrid storage model is utilized, MongoDB for MCQs and Weaviate for semantically evaluated short answers, ensuring both structured and context-aware assessments.

The adaptive mechanism begins with a default difficulty level and adjusts based on real-time performance data: increasing complexity for high-performing users and reinforcing fundamentals for those who struggle. An AI-driven feedback module provides personalized insights, highlighting strengths, areas for improvement, and actionable learning strategies. The system was tested using a qualitative approach, confirming its effectiveness in generating relevant, cognitively diverse questions and delivering meaningful feedback. The overarching goal of this research is to enhance personalized learning experiences through intelligent automation, fostering continuous learner growth across cognitive levels.

Keywords – Adaptive Learning, Bloom's Taxonomy, OpenAI, MongoDB, Weaviate, Semantic Search, Personalized Quiz Generation

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to my supervisor, Dr. Kalpani Manathunga, for her continuous guidance, encouragement, and valuable insights throughout the course of this research. Her support has been instrumental in the successful completion of my work.

I also wish to sincerely thank Ms. Thilini Jayalath, our co-supervisor, for her assistance and constructive feedback, which greatly contributed to the development of this project.

Furthermore, I am deeply thankful to my family for their unwavering support, both emotionally and financially, during this journey. I also appreciate the contributions and encouragement of my friends and project team members, whose support played a significant role in the completion of this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
MCQ	Multiple Choice Question
SAQ	Short Answer Question
PDF	Portable Document Format
LLM	Large Language Model
UI	User Interface
UX	User Experience
DB	Database
JSON	JavaScript Object Notation
ADA	OpenAI Embedding Model (text-embedding-ada-002)
GPT	Generative Pre-trained Transformer

1. INTRODUCTION

In recent years, the integration of Artificial Intelligence (AI) into educational technologies has reshaped the way students engage with learning materials. Intelligent tutoring systems, personalized learning platforms, and automated content generation tools are increasingly being adopted to improve learner outcomes and enhance instructional efficiency. Among these advancements, adaptive quiz generation systems have emerged as powerful tools to provide customized assessments tailored to individual learning needs and progress.

Traditional quiz platforms often present static sets of questions that lack relevance to the learner's current understanding or educational context. Such systems do not adapt to a student's evolving performance or offer detailed feedback, which can hinder deeper learning and critical thinking. Moreover, in educational settings where learning materials are primarily delivered through non-interactive formats such as lecture slides or PDF documents, there is a notable gap in engagement and comprehension tracking.

This research addresses these limitations by proposing an AI-driven, adaptive quiz generation system capable of dynamically producing multiple-choice and short-answer questions based on academic lecture materials. The system leverages large language models, semantic vector search, and Bloom's Taxonomy to generate questions that vary in difficulty and cognitive complexity. It not only evaluates responses using both direct matching and semantic similarity but also adapts future quizzes according to the user's performance, thereby creating a personalized learning pathway.

Furthermore, the system provides AI-generated feedback that highlights user strengths, identifies weak areas, and offers actionable suggestions for improvement. This feedback loop enhances the overall learning experience, enabling learners to better understand their progress and target their revision efforts effectively.

The goal of this research is to design, develop, and evaluate a context-aware, performance-sensitive quiz system that supports self-paced learning while promoting higher-order cognitive skills. By bridging the gap between static learning content and interactive assessment, the proposed system aims to contribute significantly to the evolution of intelligent learning environments..

1.1. Background Literature

The integration of artificial intelligence (AI) into educational technologies has revolutionized personalized learning, particularly in the domain of automated quiz generation. AI-driven systems now offer adaptive learning pathways tailored to each learner's capabilities and knowledge levels. One of the most prominent advancements is the use of large language models (LLMs) such as OpenAI's GPT-series for dynamically creating educational content.

Dijkstra et al. (2022) investigated the potential of OpenAI's GPT-3 in generating multiple-choice questions (MCQs) from educational content. While demonstrating promising results, the study also highlighted persistent challenges in fine-tuning LLMs for quality question generation, particularly in constructing effective distractors and maintaining topic relevance [1]. In a related development, Zeinalipour et al. (2024) built a multilingual quiz generation system using Turkish-Quiz-Instruct, showcasing the feasibility of generating MCQs and short-answer questions in non-English languages [2].

The pedagogical foundation for personalized quiz systems is often grounded in Bloom's Taxonomy, a hierarchical model of cognitive skills introduced by Bloom et al. (1956), which categorizes learning objectives into six levels: Remember, Understand, Apply, Analyze, Evaluate, and Create [3]. Building on this, Srivastava and Goodman (2021) used Bloom's framework to condition question generation according to learner proficiency, facilitating adaptive learning experiences [4]. Similarly, Krouska, Troussas, and Virvou (2018) explored computerized adaptive assessments that align quiz difficulty with student progress using revised Bloom's Taxonomy, underscoring the importance of cognitively aware learning systems [5].

The synergy between AI and Bloom's cognitive framework is further emphasized by Lubbe, Marais, and Kruger (2025), who discuss how AI-enhanced assessments can foster critical thinking. Their work highlights how combining AI, Bloom's Taxonomy, and pedagogy promotes independent learning and critical reasoning in digital education environments [6].

Beyond question creation, effective feedback mechanisms are essential for supporting personalized learning. Stealth assessment, a concept elaborated by Shute and Spector (2008), embeds evaluations within the learning experience, allowing for real-time feedback without disrupting the learning flow [7]. AI-powered feedback systems enhance this by analyzing learner inputs and offering targeted responses based on semantic similarity. For example, systems using semantic search platforms like Weaviate allow AI models to assess user responses based on meaning, even if phrased differently, thereby delivering more accurate and dynamic feedback [8].

Ledi (2022) developed a natural language processing (NLP)-based quiz generator that transforms raw text into MCQs and short-answer questions. The system illustrated both the potential and limitations of NLP in capturing contextual and conceptual nuances required for effective question formulation [9]. Similarly, Kang (2024) introduced a personalized quiz generation feature in the Alby LMS that adapts question difficulty in real-time based on user interaction, emphasizing the importance of responsive learning systems [10].

On a broader scale, Learning Explorer (2023) explains the science behind the ScootPad Mastery-Based Adaptive Learning Platform, which uses AI to deliver adaptive content aligned with mastery-based goals. The platform illustrates how adaptive sequencing and real-time feedback can enhance engagement and comprehension [11].

Despite these advancements, the implementation of AI-based educational systems faces notable challenges. Alhazmi et al. (2021) identified key success and failure factors in Learning Management System (LMS) adoption, such as system usability, technical infrastructure, and content relevance [12]. These challenges are critical when deploying AI-powered quiz generation systems, which demand high computational resources and tight integration with educational standards.

Future research must address these challenges by focusing on improving the scalability, semantic understanding, and adaptive precision of such systems. Moreover, enhancing alignment with pedagogical goals and diversifying linguistic support will be crucial to making personalized education accessible across broader educational contexts.

1.2. Research Gap

Recent advancements in educational technology have led to various approaches to quiz generation and assessment personalization. However, significant gaps remain in integrating established cognitive frameworks, like Bloom’s Taxonomy, with dynamic, AI-driven systems that adjust based on learner performance and understanding. A comparative review of prior research and existing platforms reveals that most systems either rely on pre-designed question banks or lack the capability to adapt questions in real-time according to the learner's mastery level. As shown in Table 1, the features and limitations of representative works in this domain are highlighted.

Table 1 Research Gap

Research	Uses Bloom's Taxonomy	Adaptive Difficulty	Real-time Quiz Generation	Personalized Feedback	LLM/AI-Based	Learner Progress Dashboard
Krouska et al. (2018) [5]	☑	☑	✗	✗	✗	✗
ScootPad (2023) [11]	✗	☑	✗	☑	✗	☑
Lubbe et al. (2025) [6]	☑	☑	✗	☑	☑	✗
Proposed System	☑	☑	☑	☑	☑	☑

Krouska et al. (2018) [5], introduced a computerized adaptive testing approach utilizing Bloom's Taxonomy. While their system successfully aligned questions with cognitive levels and adjusted difficulty based on learner performance, it relied on a static database of pre-authored questions. It lacked real-time content generation, thus limiting scalability and personalization.

ScootPad (2023) [11], an intelligent learning platform, adopts mastery-based progression and provides feedback based on performance. However, it does not categorize questions based on Bloom's levels and relies heavily on repetitive practice rather than cognitive scaffolding. It also lacks dynamic content creation, which restricts its ability to respond to varying learner needs on the fly.

Lubbe et al. (2025) [6], proposed an AI-based question generation system that integrates Bloom's cognitive framework with large language models. Their system demonstrates notable advancements in combining educational theory with AI, offering both Bloom-level alignment and personalized feedback. However, the platform still lacks real-time dynamic quiz generation and focuses more on authoring assistance for educators rather than learner-side adaptation.

In contrast, the proposed system introduces a novel approach by combining Bloom’s Taxonomy with the capabilities of large language models (LLMs) to generate quiz questions dynamically. It adapts to each learner's performance, adjusts cognitive difficulty based on Bloom’s hierarchy (e.g., from "Remember" to "Analyze"), and provides feedback to guide learner improvement. The use of LLMs enables real-time question generation from various learning materials, eliminating dependency on fixed databases. Moreover, learner analytics are incorporated to inform adaptive scaling and progress tracking, which is visualized in a learner dashboard. This allows students to track their own progress on each material, giving them a clear view of their strengths and areas for improvement. This personalized approach enhances learner engagement and cognitive development through structured, responsive, and intelligent quiz delivery.

By addressing the key limitations in previous systems, such as static question pools, lack of real-time adaptation, absence of Bloom-level differentiation, and lack of detailed progress tracking, the proposed system aims to bridge the gap between theory-based assessment frameworks and modern AI-driven personalization techniques. It not only supports self-paced learning but also empowers learners to actively monitor their development through real-time feedback and progress visualization.

1.3. Research Problem

In the modern digital learning environment, students often face challenges in receiving personalized and adaptive academic support, particularly when learning through static, non-interactive content such as lecture slides or PDFs. Traditional quiz systems lack contextual relevance, adaptive difficulty, and semantic feedback mechanisms, resulting in limited student engagement and ineffective learning reinforcement.

Current educational platforms typically rely on manually curated question banks or generic quizzes that do not align with individual performance levels or the specific learning material being studied. This approach creates gaps in knowledge assessment, especially for students who require customized guidance or practice aligned with their unique strengths and weaknesses.

Moreover, conventional quiz systems often evaluate only rote memorization through fixed answer matching. They fail to accommodate nuanced or semantically equivalent responses, especially in open-ended formats, leading to inaccurate assessments of learner understanding. The inability to interpret conceptual explanations or paraphrased answers weakens the feedback loop and diminishes the system's role in facilitating deeper learning.

Another limitation is the static nature of difficulty levels in most quiz systems. Learners are exposed to questions that are either too easy or too difficult, without any dynamic adjustment based on their ongoing performance. This leads to disengagement and prevents learners from progressing through increasingly complex cognitive tasks, as prescribed in Bloom’s Taxonomy.

These issues highlight the pressing need for an intelligent, adaptive quiz generation system that utilizes AI and semantic technologies to deliver personalized learning experiences. Such a system must be capable of understanding content contextually, adjusting difficulty dynamically, and providing meaningful feedback to foster continuous learning and performance improvement.

2. RESEARCH OBJECTIVES

2.1. Main Objective

The primary goal of this research is to develop an AI-driven adaptive quiz system that dynamically generates personalized quizzes and detailed feedback based on the user's learning progress and performance. By leveraging machine learning, user analytics, and content intelligence, the system aims to enhance student engagement, identify knowledge gaps, and provide targeted learning interventions to improve academic performance. This intelligent system will serve as a responsive learning assistant capable of supporting learners with personalized assessments and growth-oriented feedback in real time.

2.2. Sub Objectives

1. Develop a Personalized Adaptive Quiz Generation Mechanism

The objective is to design an instruction based prompt that generates quizzes dynamically based on a learner's performance history, learning patterns, and difficulty preferences. By analyzing past responses and performance metrics, the system will tailor quiz questions to match the user's current knowledge level, gradually increasing complexity to ensure progressive learning. This will involve intelligent question selection from categorized question banks, learning content, and external sources.

2. Integrate Performance Analytics to Track Learner Progress

This task focuses on collecting, processing, and visualizing learner performance data over time. The analytics component will monitor accuracy, question difficulty, completion time, and topic mastery to generate meaningful insights. These analytics help determine the learner's strong and weak areas, contributing to the continuous optimization of quiz content and personalized learning strategies.

3. Incorporate Detailed Feedback Generation Based on Quiz Responses

The system will generate detailed, AI-powered feedback after each quiz attempt. This includes a breakdown of correct and incorrect answers, explanations for each question, and personalized improvement suggestions. Using LLM the feedback module will contextualize errors and provide guidance for conceptual clarity and deeper understanding.

4. Develop a Richly Tagged Question Bank for Adaptive Use

The system will include a structured and diversified repository of questions tagged by subject, difficulty level, learning outcomes, and topic. This database will be continuously updated and refined to support adaptive logic and ensure broad coverage of the curriculum. Tagging facilitates granular control over quiz generation and supports better alignment with educational standards.

5. Implement a Dynamic Feedback Loop Using Quiz Outcomes

A crucial step is to create a feedback loop where quiz results directly inform future question selection and content adaptation. Machine learning algorithms such as reinforcement learning or Bayesian modeling will be explored to continuously improve personalization accuracy. This loop ensures that learners receive content suited to their evolving knowledge profile.

6. Develop an Intuitive Web-Based Application for Quiz Delivery

An interactive web application will be created as the primary platform for users to take quizzes, view feedback, and track progress. The interface will be designed for usability, responsiveness, and accessibility across devices, ensuring learners can engage with the system effectively in various environments. Accessibility features will also be implemented for inclusive learning.

7. Train and Evaluate the Adaptive Algorithm Using Real-World Data

Using a diverse and representative dataset collected from actual learners, the adaptive algorithm will be trained and validated. Evaluation metrics such as prediction accuracy, adaptability, quiz difficulty balance, and user satisfaction will be used to assess system performance. This evaluation ensures the algorithm remains effective across different user profiles and learning conditions.

8. Deploy the Adaptive Quiz System on Scalable Cloud Infrastructure

The final objective includes deploying the system on a robust cloud platform such as Azure or AWS to ensure scalability, reliability, and security. The deployment phase will involve API integration, cloud database configuration, and performance optimization. This will allow for real-time access, continuous monitoring, and seamless integration with other educational platforms or LMS solutions.

3. METHODOLOGY

3.1 Requirement Gathering

The requirement gathering phase of this research was focused on obtaining a comprehensive understanding of the needs, challenges, and expectations of users who interact with Learning Management Systems (LMSs). This phase involved multiple methods to ensure that the system would effectively address the gaps in existing solutions. The process was guided by both theoretical research and practical data obtained from users and experts.

3.1.1 Past Research Analysis

A thorough analysis of past research was conducted to understand the current state of personalized quiz generation systems, their limitations, and the key components necessary for effective system design. This involved reviewing studies related to quiz generation, Bloom's Taxonomy, adaptive learning technologies, and AI-based systems. Key findings from past research provided insights into common issues such as the use of static question banks, lack of real-time adaptation, and the absence of feedback mechanisms for learners. By analyzing these studies, the project was able to identify the gaps that needed to be addressed and define the features necessary for an improved system.

3.1.2 Refer to Official Documentation

Official documentation from various educational technology platforms and LMS providers was reviewed to gain an understanding of the existing tools, frameworks, and methodologies employed in quiz generation and adaptive learning. These documents helped in outlining the technical specifications and operational requirements that the system would need to integrate into existing platforms. For example, guidelines from Moodle, Blackboard, and Canvas helped ensure compatibility with commonly used LMS frameworks and standards for learning analytics, data storage, and system integration [13] [14] [15].

3.1.3 Identifying Existing Methodologies

The identification and analysis of methodologies used in existing quiz generation systems explored various adaptive learning models, including those based on Bloom's Taxonomy, mastery-based progression, and AI-driven personalization. The study examined key methodologies such as static versus dynamic question generation, feedback delivery models, and performance tracking techniques. Understanding these existing methodologies provided valuable insights, helping to define the direction for developing a more dynamic and adaptive system.

3.1.4 Data Gathering

To understand the real-world needs and challenges of LMS users, data was gathered through multiple channels. A survey was conducted with 100 university and college students who actively use LMS platforms. The survey aimed to capture their experiences with existing quiz generation systems, their preferences for adaptive learning features, and the challenges they face in personalized learning environments. Additionally, interviews were conducted with lecturers to gather insights into the pedagogical needs and challenges associated with implementing personalized assessments in LMS platforms. This data provided the foundational understanding needed to design the user-centered features of the proposed system.

3.2 Feasibility Study

The feasibility assessment for the adaptive quiz generation system evaluates its viability from technical, economic, and schedule standpoints. This system aims to personalize learner evaluation by leveraging AI-driven models, cognitive techniques, and dynamic content delivery. The technical feasibility ensures that the platform can handle complex natural language understanding tasks and real-time adaptive quiz generation. The economic feasibility focuses on cost savings through automation and enhanced learning outcomes. Finally, the schedule feasibility assesses whether development can occur within a defined project timeline, ensuring prompt integration and deployment into the LMS.

3.2.1 Technical Feasibility

The adaptive quiz generation module is technically feasible due to the advanced development of large language models (LLMs), cognitive simulation strategies, and robust deployment pipelines. A technically capable architecture that leverages modern LLMs enables real-time quiz generation, personalized question difficulty, and topic coverage analysis based on user performance. The system design also supports scalability across diverse learner profiles. Key components of technical feasibility include the following:

3.2.1.1. Knowledge of Large Language Models (LLMs)

Large Language Models like OpenAI's GPT series or Google's PaLM have become foundational for intelligent educational applications. The adaptive quiz system requires a thorough understanding of LLMs' tokenization, context window limitations, temperature tuning, and few-shot prompting techniques. These models were used for analyzing student learning patterns, transforming educational content into question formats, and interpreting natural language input from users. The depth of knowledge in using pre-trained LLMs or fine-tuning them for educational objectives is crucial to ensure accurate and relevant quiz generation tailored to specific academic content and user performance trends.

3.2.1.2. Identifying the Most Accurate Models for Quiz Generation

Accuracy and appropriateness of the generated questions play a vital role in personalized learning. Therefore, the system involved a rigorous evaluation process of available LLMs, benchmarking them on coherence, factual accuracy, Bloom's taxonomy levels, and relevance to specific academic domains. This allowed selection of the most effective models that not only generate grammatically correct questions but also match the desired difficulty level and cognitive skill. A feedback loop was established to score the questions based on difficulty, user engagement, and answer validation, which further optimized model choice and refinement.

3.2.1.3. Knowledge of Brainstorming-Level Concepts

Quiz generation at higher cognitive levels, such as critical thinking and problem-solving, required embedding brainstorming principles into the generation pipeline. Familiarity with brainstorming methodologies such as divergent thinking, idea clustering, and problem reframing helped in structuring prompts that push LLMs beyond surface-level recall questions. These principles ensured that questions stimulate curiosity, deepen understanding, and explore the subject from multiple perspectives, enhancing learner engagement and intellectual challenge.

3.2.1.4. Knowledge of Applying Brainstorming to LLM Calls

To harness LLMs for high-level question crafting, prompt engineering was applied to simulate brainstorming sessions. Multi-step prompt chaining and structured templates allowed the LLMs to first break down the topic into sub-concepts and then formulate questions addressing each sub-concept at varying complexity. This adaptive technique empowered the system to create quiz sets ranging from basic knowledge checks to open-ended analytical challenges. Embedding brainstorming logic into API calls using controlled prompt sequencing ensured cognitive variation, learner adaptability, and minimized redundancy in question patterns.

3.2.2. Economic Feasibility

Economically, the adaptive quiz generation module reduces manual effort in question creation, thereby lowering operational costs for educational institutions and content providers. By automating formative assessment and personalizing learning pathways, the system improves student retention and engagement ultimately offering better educational outcomes with reduced instructional overhead. While initial investments include cloud model access fees, LLM API costs, and compute resources for inference, the long-term ROI is positive due to improved scalability, reduced human involvement, and better learning outcomes. The financial model accounted for both infrastructure costs and the expected improvement in academic performance, which translates to institutional growth and learner satisfaction.

3.2.3. Schedule Feasibility

The development timeline of the adaptive quiz generation system is segmented into modular stages to ensure successful and timely delivery. These include (1) requirement gathering and topic classification, (2) LLM prompt engineering and evaluation, (3) system integration with the LMS, (4) user testing and feedback loop implementation, and (5) final deployment. Agile methodologies and continuous integration pipelines were adopted to ensure iterative development and regular updates. Potential schedule risks, such as API rate limits, LLM response delays, or model inaccuracies, were mitigated with fallback mechanisms and caching strategies. Given the availability of tools, pre-trained models, and experienced developers, the system is deemed schedulable within a six to eight-week timeframe for MVP delivery.

3.3. System Architecture

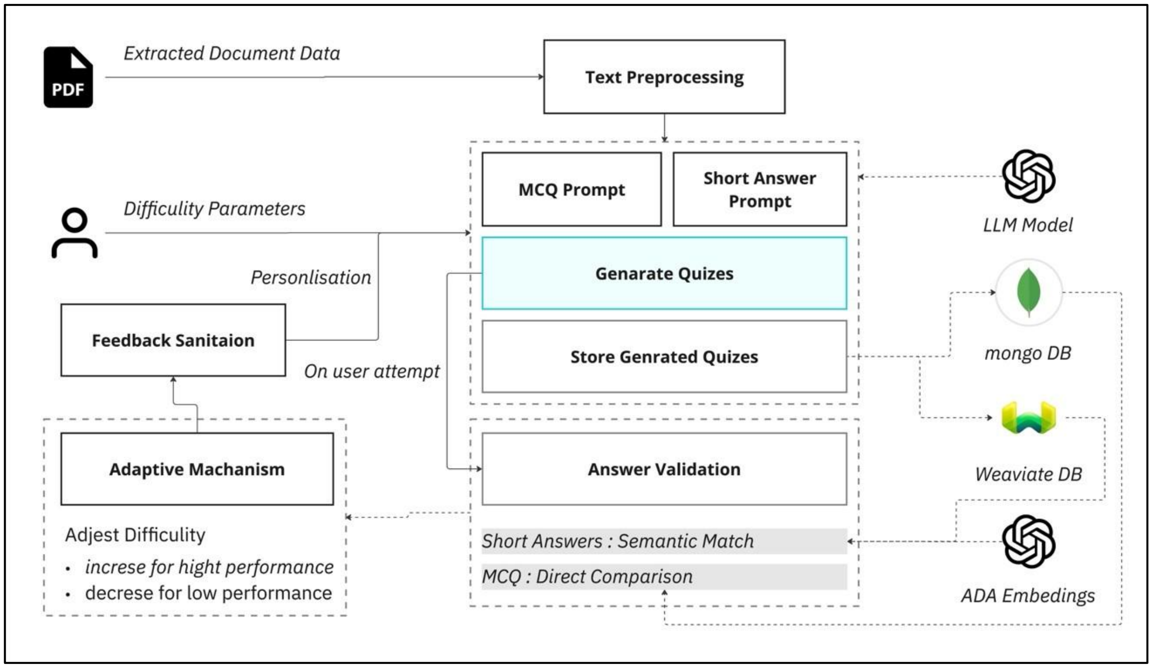


Figure 1 System Architecture

The adaptive quiz generation module is a vital component of the proposed learning management system, designed to intelligently generate personalized quizzes aligned with Bloom's Taxonomy and the learner's evolving knowledge state. This architecture integrates large language models (LLMs), structured data analysis, and user performance tracking to enable a dynamic, personalized assessment experience.

At the heart of this system lies GPT-4, which is responsible for generating both multiple-choice and short-answer questions. The quiz generation process begins with users uploading their learning content typically in PDF format through the Angular-based LMS interface. This content is extracted and processed by the backend, built using Node.js, which handles all communication between the frontend, the database, and the GPT-4 model.

Once the textual content is parsed, the system constructs specialized prompts using advanced prompt engineering techniques, including few-shot and zero-shot prompting. These prompts are tailored to generate questions across various Bloom's Taxonomy levels ranging from remembering and understanding to analyzing and creating. This ensures that the quizzes align with cognitive learning goals and provide a comprehensive evaluation of the learner's understanding.

User performance data, including quiz scores, time taken, and accuracy per Bloom's level, is persistently stored in MongoDB for multiple-choice questions and in a separate Wiveate database for short-answer questions. This data is used to assess the learner's strengths and weaknesses, allowing the system to dynamically adjust quiz difficulty levels in subsequent attempts. The backend analyzes this performance data to detect patterns and determine the most appropriate Bloom level for future questions, thereby creating a personalized scaffolding model for each learner.

Furthermore, the system supports continuous adaptation by updating the user's quiz history and integrating it with mind map generation and resource recommendation modules. Incorrect answers are analyzed, and GPT-4 provides feedback, explanations, and follow-up hints.

All components are deployed in containerized Docker environments and hosted on AWS to ensure scalability, high availability, and secure resource management. Continuous integration and deployment pipelines are managed through GitHub Actions, enabling automated testing, builds, and updates for the entire adaptive quiz generation module.

3.3.1. Backend Integration

An integrated backend system played a pivotal role in the effective execution of personlized quiz generation within the proposed learning management system. This integration was essential for ensuring the accurate assessment of learners' understanding, the preservation of quiz records, and the personalization of future assessments based on user-specific learning patterns.

The backend pipeline began with the initial generation of quizzes using GPT-4, where raw educational content typically extracted from PDF documents was sent to the backend, built using Node.js. This backend system was responsible for orchestrating prompt creation, communicating with the GPT-4 model, and saving the generated quiz content to a persistent database.

All quiz questions and their corresponding answers generated during this phase were saved in MongoDB, forming the initial version of each quiz record. Each record was associated with metadata including the source content, timestamp, and user identification.

Upon quiz completion by the user, the backend processed the submitted responses and updated the original quiz record with key performance indicators. These indicators included the Bloom's Taxonomy level assigned to each question, the user's score for each quiz item, and the difficulty level encountered by the learner. This structured update enabled the system to not only assess current performance but also build a personalized performance history for each user.

This enriched performance history was then used in subsequent quiz generation. Specifically, three parameters Bloom's level, score, and difficulty were extracted from the learner's past interactions and passed to the prompt engineering module to tailor the next set of quiz questions. This adaptive feedback loop allowed the system to gradually scaffold learning by dynamically adjusting the complexity and focus of new quizzes.

3.3.2. REST API

The REST API served as the central communication conduit between the Angular-based frontend interface and the Node.js-powered backend server in the adaptive quiz generation module of the LMS. It played a crucial role in enabling smooth data exchange, ensuring that user interactions on the frontend were seamlessly recorded, processed, and utilized by the backend services to maintain the personalized learning experience.

In this system, all large language model (LLM) interactions, including quiz generation, Bloom's taxonomy interpretation, and difficulty-level handling, were orchestrated entirely on the frontend. Once the user triggered a quiz generation request by selecting a topic or completing a learning module the frontend first passed Bloom's level specifications, the number of questions per level, and difficulty parameters to the LLM. Predefined Bloom's level descriptions guided the LLM to produce questions that aligned with cognitive learning stages such as Remembering, Understanding, Applying, and Evaluating.

Following this, the generated quiz questions, along with their metadata (e.g., associated Bloom's level, difficulty, question count), were transmitted to the backend via REST API calls. These POST requests allowed the backend to store newly generated quizzes in MongoDB, associating them with a unique user identifier and timestamp for versioning and retrieval.

Upon quiz completion, the REST API again facilitated the secure transmission of user responses and evaluation metrics back to the backend. While the validation of answers performed using a hybrid technique combining keyword matching and semantic similarity was handled entirely on the frontend, the backend remained responsible for storing final results and updating performance data. Each quiz record was updated with the user's score, question-level difficulty encountered, and inferred Bloom's level of mastery.

This iterative update mechanism enabled the backend to serve as a learning profile repository, while REST APIs ensured that these evolving datasets were accessible on-demand by the frontend. In the subsequent quiz generation cycle, the backend served the updated performance metrics back to the frontend, which then passed them as new parameters to the LLM for tailored quiz creation.

By offloading all LLM processing and validation logic to the frontend while relying on the REST API to manage data flow, storage, and retrieval, the system maintained a clean separation of concerns. This architecture not only ensured scalability but also optimized latency, as real-time interactions were processed locally, while state persistence and learning history were managed centrally.

The REST API thus functioned as the backbone of communication, enabling dynamic generation, real-time feedback, and continuous personalization all essential components of an intelligent, adaptive learning environment.

3.4. Implementation

Overall technologies that are going to be used for the persoanlized quiz component.

- Angular – For developing the frontend user interface of the adaptive quiz web application.
- Node.js – For handling backend logic, including adaptive algorithms and API management.
- Wiveate – Utilized for short answer generation and conducting hybrid and keyword-based searches to validate student responses.
- MongoDB Compass – For managing and visualizing the quiz-related data stored in the MongoDB database.
- Docker – To containerize services and ensure consistency across different deployment environments.
- GitHub – For version control and project collaboration.

- GitHub Copilot – To assist with AI-powered code completion and suggestion.
- Postman – For testing and validating RESTful APIs during development.

3.4.1 Model Selection

For the generation of adaptive quizzes, the GPT-4 model was selected due to its superior natural language understanding and generation capabilities. GPT-4 enables the creation of high-quality, context-aware questions that align with Bloom's Taxonomy, ensuring cognitive depth across knowledge, comprehension, and application levels.

```
try {
    const response = await openai.chat.completions.create({
        model: 'gpt-4',
        messages: [{ role: 'user', content: mcqPrompt }],
        max_tokens: 3000,
        temperature: 0.1,
    });
```

Figure 2 Model Selection

Advanced reasoning abilities support the dynamic generation of multiple question types, including MCQs and short answers. Furthermore, GPT-4 is integrated with a difficulty-level control mechanism, allowing the system to adjust the complexity of questions based on learner performance, making it ideal for personalized learning environments.

3.4.2. Bloom's Taxonomy Integration

Each level targets specific cognitive skills, enabling the system to tailor questions based on student proficiency and learning objectives. The prompt is structured using **f**ew-shot prompting, where the AI is given multiple labeled examples within the prompt itself to better understand the pattern and context of the desired output.

```
export const bloomTaxonomyPrompt = `
    ### 🌟 **Understanding Bloom's Taxonomy** 🌟

    Bloom's Taxonomy categorizes cognitive skills into six levels:

    1️⃣ **Remembering (Basic Recall)**
       - Recognizing and recalling facts, definitions, and concepts.
       - Example: *"What is the chemical symbol for oxygen?"*

    2️⃣ **Understanding (Explain Concepts)**
       - Interpreting and summarizing information.
       - Example: *"Explain how photosynthesis works."*

    3️⃣ **Applying (Use Knowledge in a New Situation)**
       - Solving problems using learned concepts.
       - Example: *"Use Newton's laws to calculate force."*

    4️⃣ **Analyzing (Break Down Concepts & Compare)**
       - Identifying relationships, patterns, and comparisons.
       - Example: *"What are the key differences between prokaryotic and eukaryotic cells?"*

    5️⃣ **Evaluating (Make Judgments & Justify Opinions)**
       - Assessing theories, arguments, or solutions.
       - Example: *"Which economic policy is better for inflation control?"*

    6️⃣ **Creating (Generate New Ideas & Solutions)**
       - Designing new approaches, models, or original work.
       - Example: *"Develop a marketing strategy for a new product."*

    ---
    **💡 The AI must use this taxonomy while generating questions.**
`;
```

Figure 3 Bloom's Taxonomy Integration

3.4.3. Feedback Generation

This function is designed to generate AI-powered feedback for quizzes by analyzing the user's answers. It takes the total number of questions, the number of correct answers, and a list of incorrectly answered questions (including the user's answer, the correct answer, and other options).

The output includes:

- A quiz summary showing the total, correct, and incorrect answers.
- A detailed breakdown of each incorrect question with the user's answer, correct answer, and other possible options.
- Feedback and suggestions that highlight strengths, identify mistakes, and offer tips for improvement, such as focusing on specific topics or practicing similar questions.

This helps users understand where they went wrong and how they can do better in future quizzes.

```
    return `
You are an AI that provides detailed feedback on quizzes based on user answers.

**Quiz Summary:**
- **Total Questions:** ${totalQuestions}
- **Correctly Answered:** ${correctAnswers}
- **Incorrectly Answered:** ${totalQuestions - correctAnswers}

**Incorrect Answers Breakdown:**
${incorrectQuestions.map(q => `
   **Question:** ${q.question}
   - **User Answer:** ${q.userAnswer}
   - **Correct Answer:** ${q.correctAnswer}
   - **Other Possible Answers:** ${q.otherAnswers.join(', ')}
`).join('\n')}

**Feedback & Improvement Suggestions:**
- Analyze the user's performance, highlighting strengths and weaknesses.
- Identify common mistakes or patterns in incorrect answers.
- Suggest strategies for improvement, such as focusing on specific topics or practicing similar questions.
- Provide actionable tips to help the user perform better in future quizzes.
`
};
```

Figure 4 Feedback Generation

The output includes:

- A quiz summary showing the total, correct, and incorrect answers.
- A detailed breakdown of each incorrect question with the user's answer, correct answer, and other possible options.
- Feedback and suggestions that highlight strengths, identify mistakes, and offer tips for improvement, such as focusing on specific topics or practicing similar questions.

This helps users understand where they went wrong and how they can do better in future quizzes.

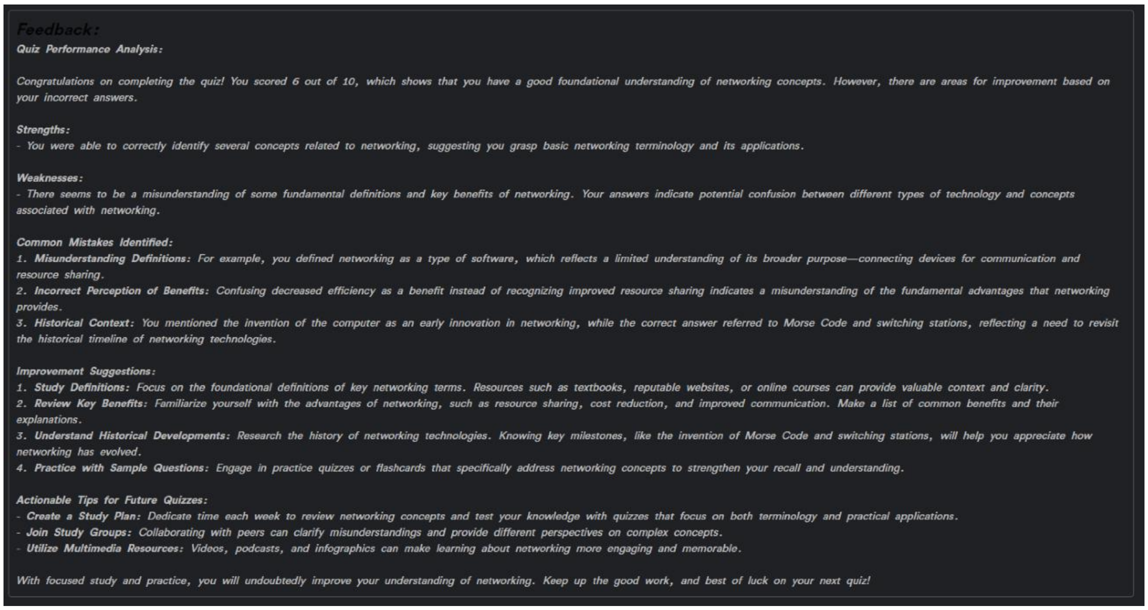
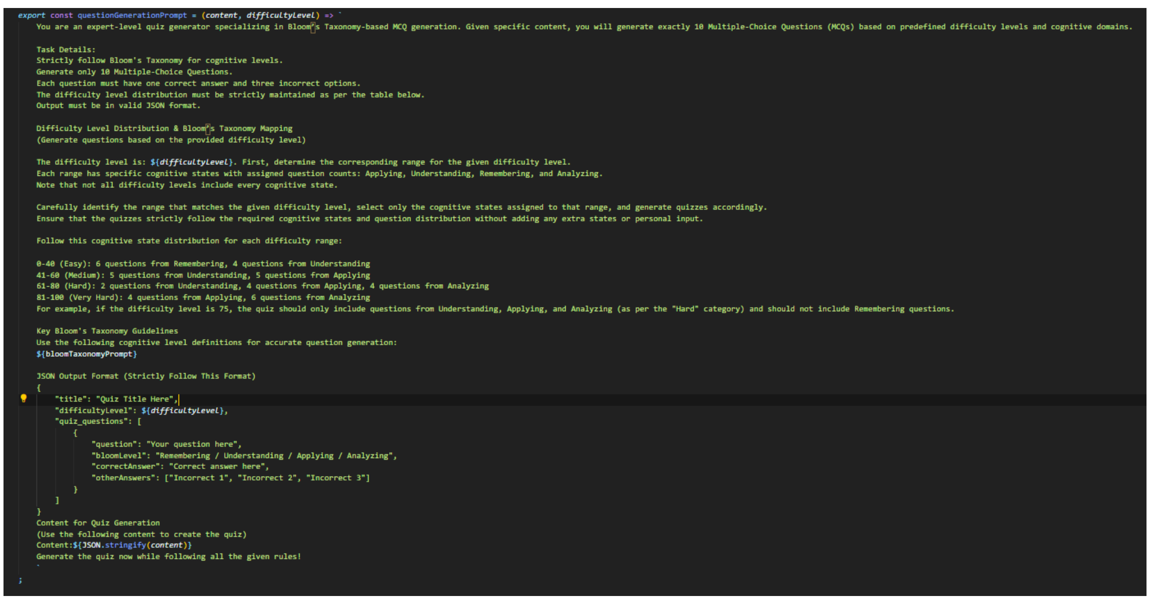


Figure 5 Feedback Display

3.4.4. Quiz Generation

To generate MCQs effectively, we integrated Bloom's Taxonomy level descriptions along with custom difficulty levels into the prompt. Each difficulty level is mapped to a specific combination of Bloom's cognitive states and a defined number of questions per level. For example, "Easy" level includes more Remembering and Understanding questions, while "Very Hard" focuses on Applying, Analyzing, and Evaluating. This ensures that the AI generates quizzes that are both cognitively balanced and aligned with the learner's level. The prompt uses few-shot prompting and structured instructions to produce consistent and high-quality MCQs in JSON format.

Similarly, for Short Answer Questions (SAQs), Bloom’s Taxonomy is used to guide the depth and complexity of the questions. SAQs are designed to encourage learners to articulate their understanding more openly, especially at higher cognitive levels such as Analyze, Evaluate, and Create. The generation prompt includes structured examples and task-specific instructions to ensure clarity, relevance, and alignment with the learner’s progress. SAQs are also output in a JSON format with corresponding reference answers, enabling precise feedback and semantic comparison with learner responses.



```
export const questionGenerationPrompt = (content, difficultyLevel) => `
    You are an expert-level quiz generator specializing in Bloom's Taxonomy-based MCQ generation. Given specific content, you will generate exactly 10 Multiple-Choice Questions (MCQs) based on predefined difficulty levels and cognitive domains.

    Task Details:
    Strictly follow Bloom's Taxonomy for cognitive levels.
    Generate only 10 Multiple-Choice Questions.
    Each question must have one correct answer and three incorrect options.
    The difficulty level distribution must be strictly maintained as per the table below.
    Output must be in valid JSON format.

    Difficulty Level Distribution & Bloom's Taxonomy Mapping
    (Generate questions based on the provided difficulty level)

    The difficulty level is: ${difficultyLevel}. First, determine the corresponding range for the given difficulty level.
    Each range has specific cognitive states with assigned question counts: Applying, Understanding, Remembering, and Analyzing.
    Note that not all difficulty levels include every cognitive state.

    Carefully identify the range that matches the given difficulty level, select only the cognitive states assigned to that range, and generate quizzes accordingly.
    Ensure that the quizzes strictly follow the required cognitive states and question distribution without adding any extra states or personal input.

    Follow this cognitive state distribution for each difficulty range:

    0-40 (Easy): 6 questions from Remembering, 4 questions from Understanding
    41-60 (Medium): 5 questions from Understanding, 5 questions from Applying
    61-80 (Hard): 2 questions from Understanding, 4 questions from Applying, 4 questions from Analyzing
    81-100 (Very Hard): 4 questions from Applying, 6 questions from Analyzing
    For example, if the difficulty level is 75, the quiz should only include questions from Understanding, Applying, and Analyzing (as per the "Hard" category) and should not include Remembering questions.

    Key Bloom's Taxonomy Guidelines
    Use the following cognitive level definitions for accurate question generation:
    ${bloomTaxonomyPrompt}

    JSON Output Format (Strictly Follow This Format)
    {
        "title": "Quiz Title Here",
        "difficultyLevel": ${difficultyLevel},
        "quiz_questions": [
            {
                "question": "Your question here",
                "bloomLevel": "Remembering / Understanding / Applying / Analyzing",
                "correctAnswer": "Correct answer here",
                "otherAnswers": ["Incorrect 1", "Incorrect 2", "Incorrect 3"]
            }
        ]
    }
    Content for Quiz Generation
    (Use the following content to create the quiz)
    Content:${JSON.stringify(content)}
    Generate the quiz now while following all the given rules!
    `
;
```

Figure 6 MCQ Prompt

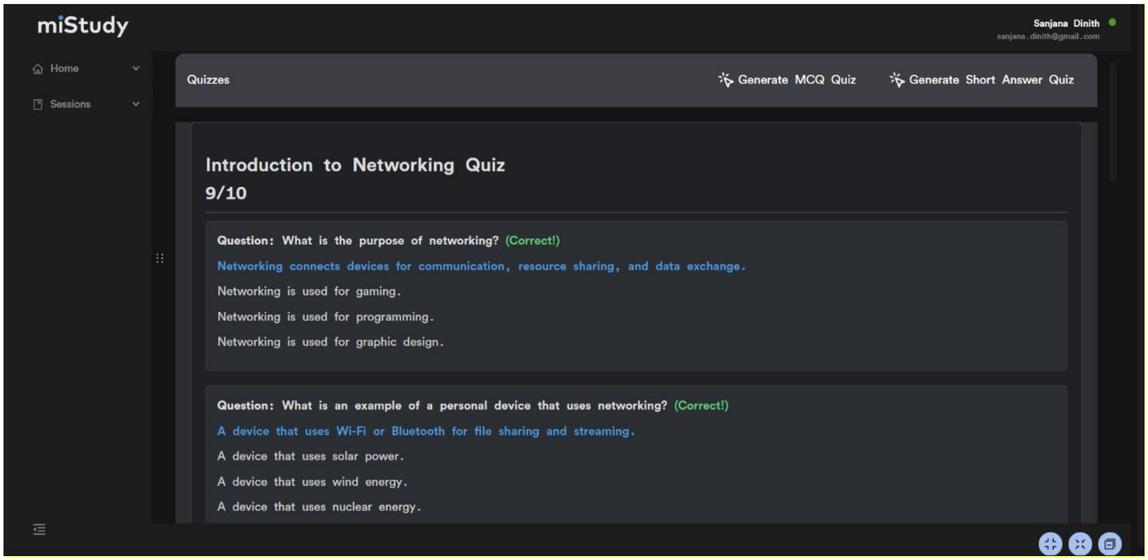
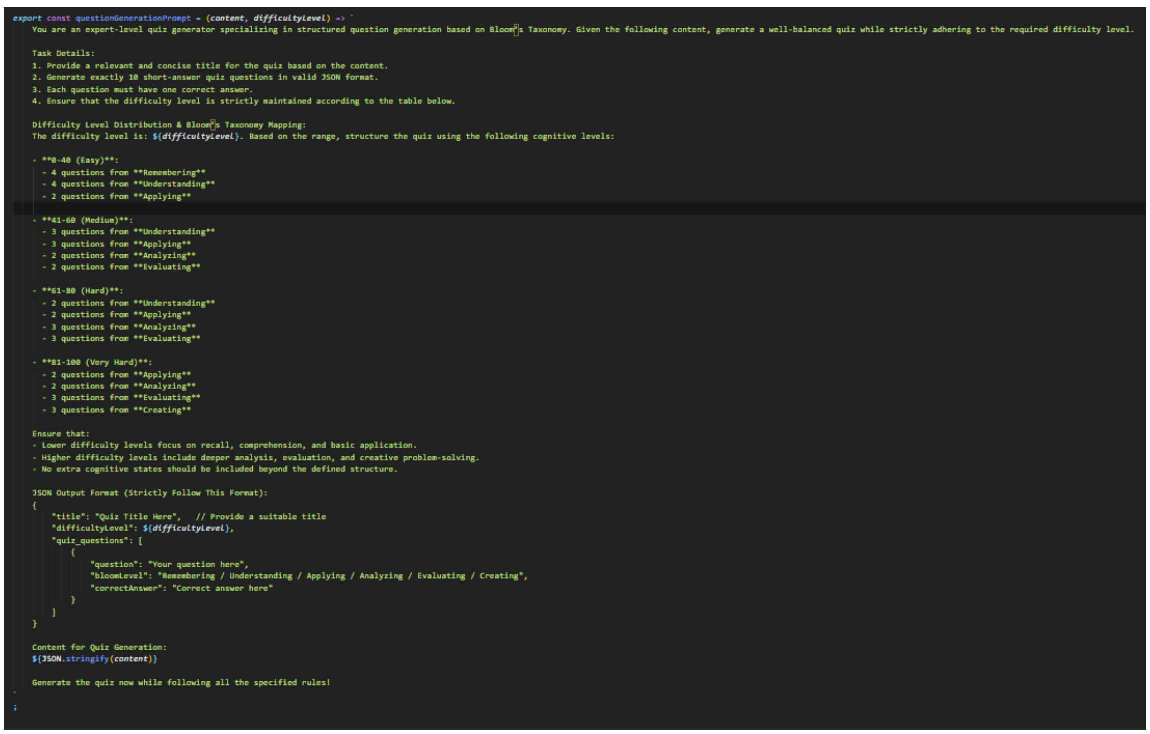


Figure 7 MCQ UI



```
export const questionGenerationPrompt = (content, difficultyLevel) => `
    You are an expert-level quiz generator specializing in structured question generation based on Bloom's Taxonomy. Given the following content, generate a well-balanced quiz while strictly adhering to the required difficulty level.

    Task Details:
    1. Provide a relevant and concise title for the quiz based on the content.
    2. Generate exactly 10 short-answer quiz questions in valid JSON format.
    3. Each question must have one correct answer.
    4. Ensure that the difficulty level is strictly maintained according to the table below.

    Difficulty Level Distribution & Bloom's Taxonomy Mapping:
    The difficulty level is: ${difficultyLevel}. Based on the range, structure the quiz using the following cognitive levels:

    - **0-40 (Easy)**:
      - 4 questions from **Remembering**
      - 4 questions from **Understanding**
      - 2 questions from **Applying**

    - **41-60 (Medium)**:
      - 3 questions from **Understanding**
      - 3 questions from **Applying**
      - 2 questions from **Analyzing**
      - 2 questions from **Evaluating**

    - **61-80 (Hard)**:
      - 2 questions from **Understanding**
      - 2 questions from **Applying**
      - 3 questions from **Analyzing**
      - 3 questions from **Evaluating**

    - **81-100 (Very Hard)**:
      - 2 questions from **Applying**
      - 2 questions from **Analyzing**
      - 3 questions from **Evaluating**
      - 3 questions from **Creating**

    Ensure that:
    - Lower difficulty levels focus on recall, comprehension, and basic application.
    - Higher difficulty levels include deeper analysis, evaluation, and creative problem-solving.
    - No extra cognitive states should be included beyond the defined structure.

    JSON Output Format (Strictly Follow This Format):
    {
        "title": "Quiz Title Here",   // Provide a suitable title
        "difficultyLevel": ${difficultyLevel},
        "quiz_questions": [
            {
                "question": "Your question here",
                "bloomLevel": "Remembering / Understanding / Applying / Analyzing / Evaluating / Creating",
                "correctAnswer": "Correct answer here"
            }
        ]
    }

    Content for Quiz Generation:
    ${JSON.stringify(content)}

    Generate the quiz now while following all the specified rules!
`
;
```

Figure 8 SAQ Prompt

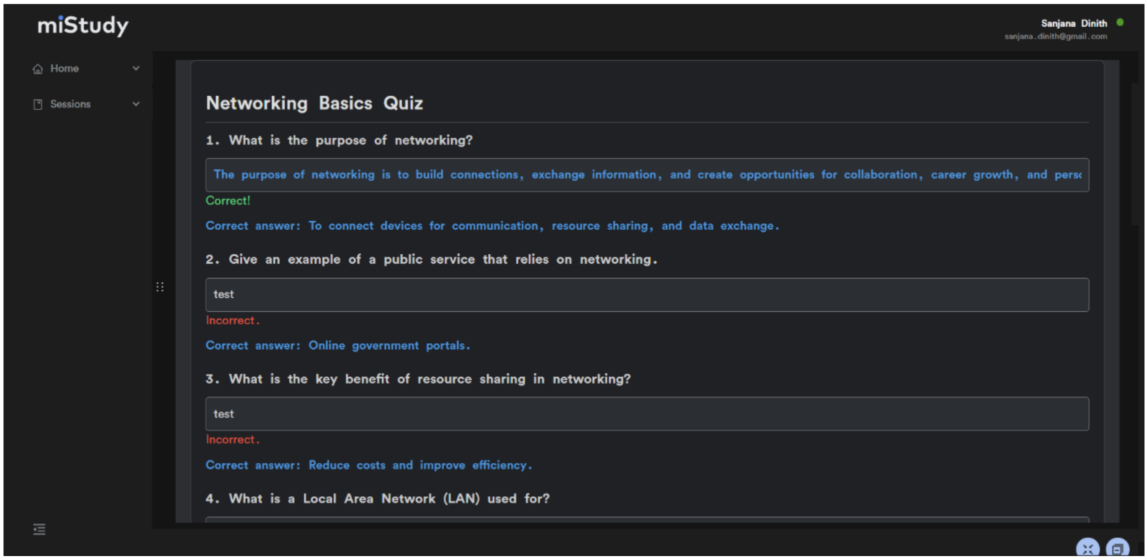


Figure 9 SAQ UI

3.4.6. Difficulty level and score storage with database level

After a user completes a quiz, both the difficulty level of the attempted quiz and the achieved score are stored in the MongoDB database. This data is essential for tracking individual progress and is used in adaptive features of the system such as personalized mind map generation and dynamic difficulty adjustments for future quizzes. Each quiz attempt is logged with associated metadata including the timestamp, selected difficulty level, and calculated score, ensuring a reliable data structure for analytics and feedback loops.

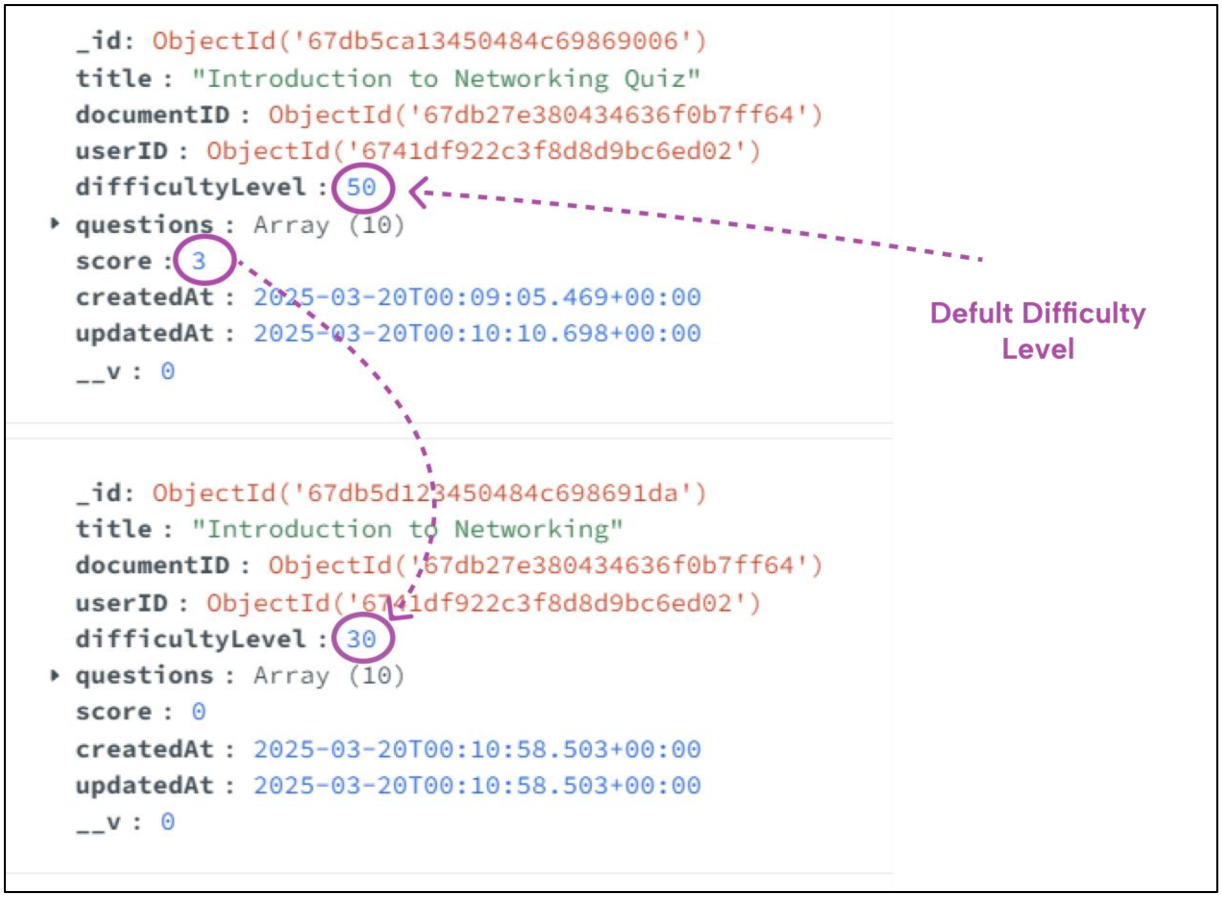


Figure 10 Difficulty Level Adjustments

3.5. Project Requirements

3.5.1. Functional Requirements

1. The system should allow users to upload PDF study materials, which are automatically parsed and indexed for quiz generation.

2. The system must get the extracted textual content from the uploaded PDFs data base content and should be used in generating quiz questions.

3. Users should be able to generate an unlimited number of quizzes based on the uploaded content and personal learning preferences.

4. The quiz generation engine should utilize Bloom's Taxonomy to create questions across different cognitive levels (e.g., Remember, Understand, Apply, Analyze, Evaluate, Create).

5. The system must analyze the user's performance history and generate the next quiz based on the user's individual difficulty level and brainstorming stage.

6. Generated quizzes must consist of two types:
   - Multiple Choice Questions (MCQs): Managed and stored using MongoDB.
   - Short Answer Questions: Handled using the Wiveate database.

7. Users must be able to view the difficulty level of each question and track their performance per quiz.

8. A performance feedback mechanism must be in place, providing accuracy scores and improvement suggestions after quiz completion.
9. Users should be allowed to reattempt quizzes or request new ones, adapting to their learning curve in real-time.

10. Quizzes should be generated dynamically using a combination of extracted PDF content, pre-stored templates, and Bloom's-aligned logic.

3.5.2. Non-Functional Requirements

Performance

- The system must be able to generate quizzes within 3 seconds of user request, even under concurrent load.
- Quiz generation should be optimized to minimize redundant processing using caching or pre-processing strategies.

Scalability

- The system should scale horizontally to handle increasing user requests and content upload volumes.
- MongoDB and Wiveate databases should be capable of handling growing datasets without performance degradation.

Reliability

- The quiz generation process should ensure question quality, avoiding duplication and maintaining relevance to the PDF topic.
- In case of any failure (e.g., database disconnection), the system should fail gracefully and offer retry options.

Security

- All uploaded PDFs and user-generated quizzes must be securely stored with role-based access.
- User performance data and content interactions should be encrypted and protected from unauthorized access.

Usability

- The quiz interface should be user-friendly, allowing users to take quizzes, review answers, and see explanations seamlessly.
- Users should receive immediate feedback with clear score summaries, difficulty insights, and next-step suggestions.

Maintainability

- The quiz generation logic should be modular, allowing developers to add new question types or cognitive levels easily.
- The system must include clear documentation on the quiz generation algorithm, content extraction logic, and database schemas.
-

Interoperability

- The system should be designed to integrate with other LMS components such as the mind map generator and personalized dashboard.
- REST APIs must be exposed for quiz access, performance tracking, and external analytics integrations.

3.5.3. User Requirements

- Users should be able to generate quizzes based on their uploaded PDF files without limitations on quantity.
- Users should be presented with quizzes adapted to their learning stage, increasing in complexity over time.
- Users must receive detailed results after each quiz, including correct answers, explanations, and scores.
- Users should have the option to retake or regenerate quizzes dynamically based on new study sessions or performance drops.
- The system should personalize future quizzes using historical performance metrics, such as scores, time taken, and attempted difficulty levels.

3.5.4. System Requirements

- The backend is built using Node.js, interfacing with MongoDB for MCQs and Wiveate for short answer question data management.
- Quiz content is extracted from PDFs using integrated PDF parsing tools and natural language processing techniques.
- The system utilizes Bloom's Taxonomy mapping logic to classify and generate appropriate questions per cognitive level.
- Adaptive quiz logic is implemented through intelligent algorithms that track difficulty scaling and user performance over time.
- The frontend is developed using Angular, offering users an intuitive interface for uploading PDFs, taking quizzes, and viewing results.
- The system should maintain audit logs of quiz attempts, user interactions, and feedback to review usage patterns.
- Integration with external APIs (e.g., AI-based summarization, AI-based question generation models) should be supported for enhanced quiz quality.

3.6. Design Diagrams

3.6.1. Use Case Diagram

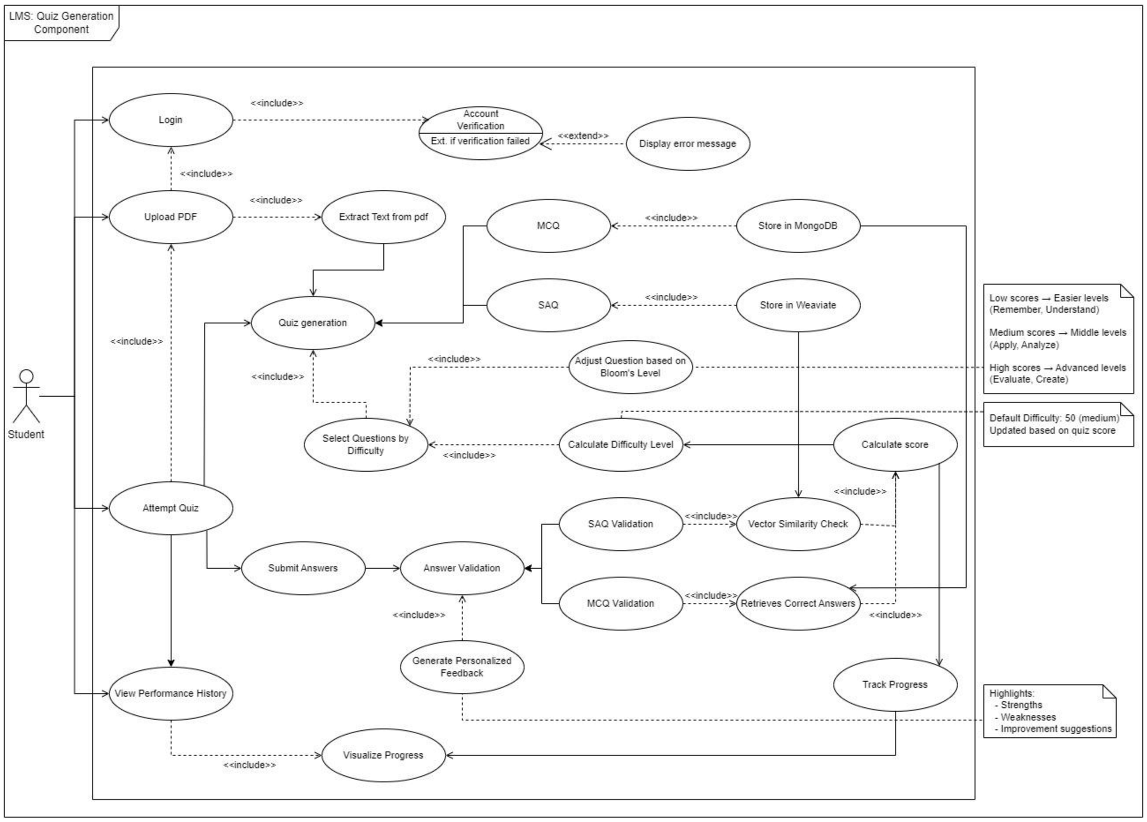


Figure 11 Use Case Diagram (see Appendix I)

3.6.2. Sequence Diagram

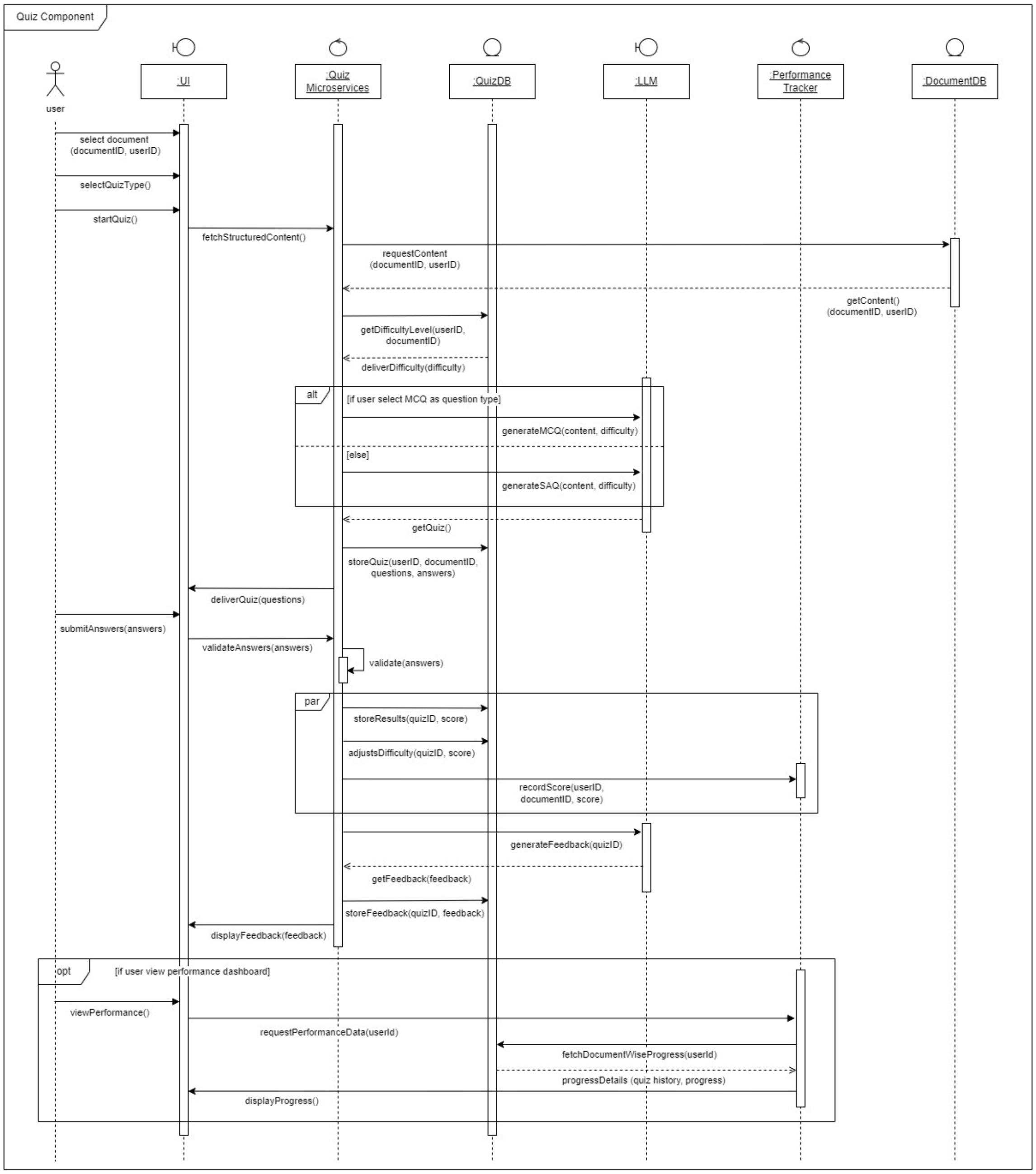


Figure 12 Sequence Diagram (see Appendix II)

3.7. Commercialization

miStudy is an AI-powered Adaptive Learning Management System (LMS) that aims to revolutionize the digital learning landscape in Sri Lanka. One of the key intelligent components of miStudy is the Personalized Adaptive Quiz Generator, designed to deliver targeted assessments that dynamically adjust based on individual learner performance, cognitive level, and Bloom's taxonomy. This intelligent assessment solution addresses the growing need for personalized education tools that enhance student engagement and academic achievement.

Figure 13 miStudy Logo

Market Research and Analysis

- A detailed analysis of the Sri Lankan and South Asian education markets was conducted to assess the demand for personalized learning tools and smart evaluation methods.
- The primary target audiences include universities, private tuition centers, e-learning platforms, and educational content providers.
- Feedback from pilot testing in university settings showed a marked improvement in learner satisfaction and knowledge retention due to the adaptive nature of the quiz experience.

Product Development

- miStudy is built as a modular web-based LMS using Angular for the frontend and Node.js with MongoDB for the backend.
- The Adaptive Quiz Generator uses advanced LLM-based prompt engineering to create quizzes that align with Bloom's taxonomy and adapt in difficulty based on the learner's previous performance.
- All LLM interactions, including quiz creation and keyword-based hybrid answer validation, are handled on the frontend. The backend is responsible for storing quiz metadata, including difficulty level, Bloom's level, user performance metrics, and historical quiz data.
- After each quiz, feedback is generated and associated scores, Bloom's level, and difficulty values are updated in the database. These values are then passed as input parameters for generating the next quiz in the cycle.

Monetization Strategy

- A tiered subscription model has been devised to cater to different market segments:
    - Individual learners and tutors: Basic plan offering limited adaptive quiz attempts and feedback features.
    - Educational institutions: Professional plan with full access to adaptive quizzes, analytics dashboards, and student progress tracking.
    - Corporate clients and universities: Enterprise plan including custom integration, API access, advanced analytics, and batch management.
- A freemium model is also incorporated to allow new users to experience the platform with limited access before opting into a subscription.

Marketing and Promotion

- A comprehensive marketing strategy is being developed to include digital outreach via social media platforms (LinkedIn, Facebook), educational content creators, and paid promotions through Google Ads.
- miStudy will be showcased at education technology expos, university exhibitions, and national academic events to attract institutional clients.
- Collaborative campaigns with academic influencers and YouTube educators will help build brand credibility and increase adoption.

Regulatory Compliance

- The miStudy platform adheres to modern data protection standards, ensuring the security and privacy of learner data.
- All personal and academic records are encrypted and stored securely, with role-based access controls and regular compliance checks aligned with educational data regulations.

Continuous Innovation and Iteration

- miStudy is committed to continuous R&D efforts to improve quiz accuracy, feedback generation, and learner profiling.
- Future plans include expanding the quiz module to support additional formats like drag-and-drop, coding questions, and comprehension-based scenarios.
- Personalized recommendation engines for content and quizzes will be developed using learner analytics and quiz performance insights.

3.8. Testing

Robust testing procedures were essential for validating the effectiveness and reliability of the Adaptive Quiz Generator component within the miStudy platform. The primary goal of testing was to ensure that the personalized quiz generation logic produced accurate, meaningful, and pedagogically aligned questions that reflected the learner's cognitive development and performance trends.

The testing phase involved a comprehensive assessment of quiz generation mechanisms, focusing on the system's ability to generate appropriate questions aligned with Bloom's Taxonomy levels, difficulty levels, and learner performance data. A controlled dataset comprising simulated learner profiles with varying scores, knowledge gaps, and progression patterns was used to evaluate the adaptability and personalization of the quizzes.

Key factors such as question relevance, difficulty alignment, Bloom's level mapping accuracy, and adaptive sequencing were systematically validated. The testing also focused on how accurately the system could update Bloom's level, score, and difficulty tags after each quiz attempt and how effectively these parameters influenced the generation of subsequent quizzes.

Hybrid keyword-based validation logic, handled entirely on the frontend, was tested against a series of sample answers to assess the robustness of the feedback and answer evaluation mechanisms. Additionally, API communication between the backend and frontend was tested for consistency, ensuring proper retrieval and updating of quiz data in MongoDB.

Performance metrics such as question quality accuracy, adaptive response rate, and feedback consistency were recorded. Furthermore, user studies were conducted where participants interacted with multiple quiz sessions, and their engagement levels and learning progress were monitored.

Table 2 Test case for verifying multiple-choice question generation

Test Case ID	01
Test Case	Verify multiple-choice question generation.
Test Case Scenario	System generates a multiple-choice question based on the structured content from the database.
Input	"What is the capital of France?" (From MongoDB)
Expected Output	The system generates a question with four options (one correct and three distractors).
Actual Result	MCQ generated successfully with the correct answer and distractors.
Status (Pass / Fail)	Pass

Table 3 Test case for verifying short-answer question generation

Test Case ID	02
Test Case	Verify short-answer question generation.
Test Case Scenario	System generates a short-answer question based on content retrieved from Weaviate.
Input	"Describe the process of photosynthesis." (From Weaviate)
Expected Output	The system generates a short-answer question with a prompt for the user to answer.
Actual Result	SAQ generated successfully with an appropriate prompt.
Status (Pass / Fail)	Pass

Table 4 Test case for verifying adaptive difficulty scaling based on user performance

Test Case ID	03
Test Case	Verify adaptive difficulty scaling based on user performance.
Test Case Scenario	User answers multiple quizzes, and the system adjusts the difficulty based on their performance.
Input	User completes a set of quizzes with varying difficulty levels.
Expected Output	System adjusts the difficulty level for the next quiz based on previous responses (e.g., increases difficulty after correct answers).
Actual Result	Difficulty level adjusted correctly after each quiz.
Status (Pass / Fail)	Pass

Table 5 Test case for verifying progress tracking on user dashboard

Test Case ID	04
Test Case	Verify progress tracking on user dashboard.
Test Case Scenario	The system tracks and displays user progress on the dashboard for each material.
Input	User completes a quiz on a specific topic.
Expected Output	User's progress is updated and displayed on the dashboard, showing completion percentage.
Actual Result	Progress is accurately tracked and displayed on the dashboard.
Status (Pass / Fail)	Pass

Table 6 Test case for verifying feedback generation based on user quiz performance

Test Case ID	05
Test Case	Verify feedback generation based on user quiz performance.
Test Case Scenario	The system provides feedback to the user after completing a quiz, offering improvement suggestions based on their performance.
Input	User completes a quiz and receives a score of 60%.
Expected Output	The system provides feedback such as "You scored 60%. Focus on understanding key concepts related to the topic. Review material X for improvement."
Actual Result	Feedback is generated and displayed according to the user's performance.
Status (Pass / Fail)	Pass

4. RESULTS & DISCUSSION

4.1. Results

This section outlines the implementation outcomes of the personalized quiz generation system, tested using academic lecture materials in PDF format. The functionality and effectiveness of the system were evaluated through qualitative testing and user observation. Each component was examined independently to ensure accurate content generation, semantic response validation, and adaptive behavior.

4.1.1. Quiz Generation Output

The system was designed to generate two types of questions: Multiple Choice Questions (MCQs) and Short Answer Questions, which were tailored to the content of the uploaded lecture PDFs. The quiz generation process utilized OpenAI's language model, incorporating Bloom's Taxonomy to guide the cognitive level of the questions.

- MCQs were generated with a focus on the first four levels of Bloom's Taxonomy: Remembering, Understanding, Applying, and Analyzing.
- Short Answer Questions covered all six levels, extending to Evaluating and Creating, fostering deeper learning and critical thinking.

Through initial testing, the generated questions were found to align well with the source material. The questions appropriately reflected key concepts, terminology, and scenarios within the academic content. The process ensured that questions progressively increased in difficulty based on cognitive skill levels, ensuring both engagement and challenge for users.

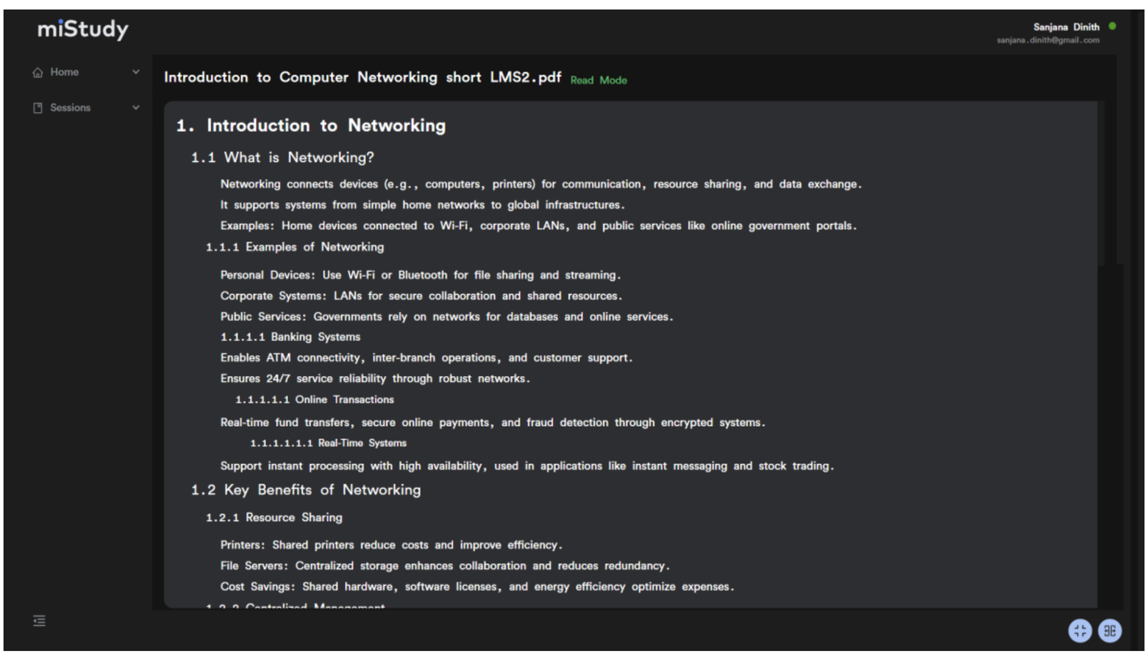


Figure 14 source material

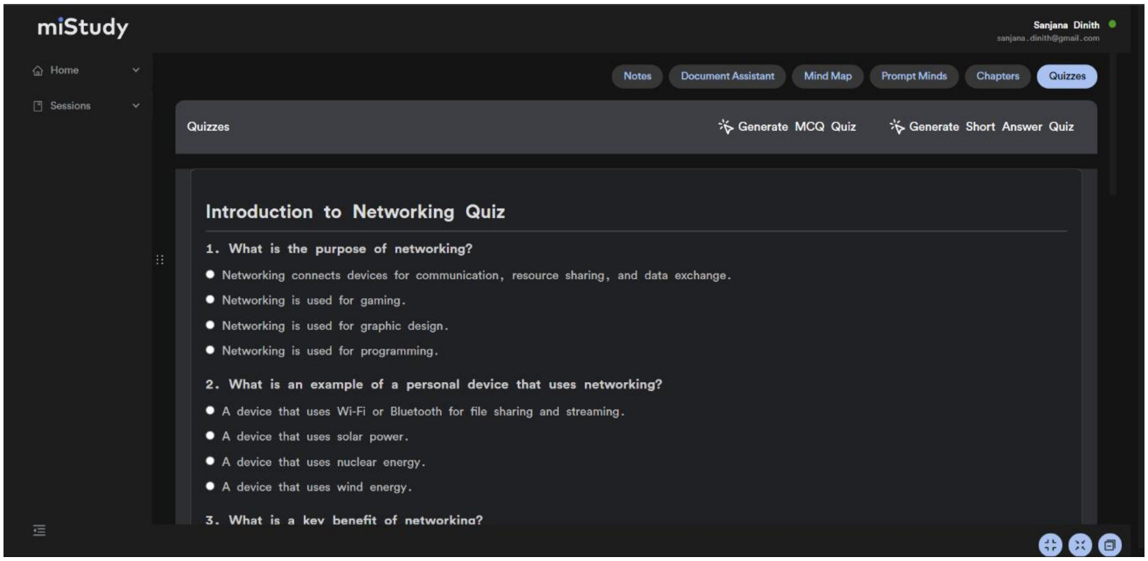


Figure 15 Generated MCQ Interface

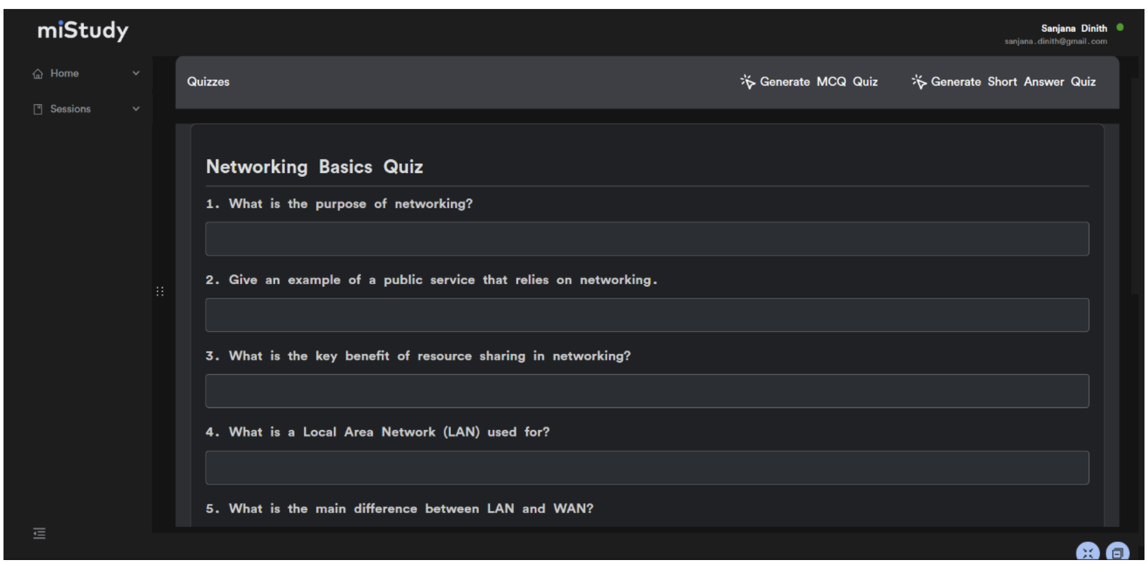


Figure 16 Generated SAQ Interface

The system takes a source paragraph as input and generates corresponding questions tailored to assess various cognitive levels. As shown in Figure 14, the paragraph serves as the basis for generating both short-answer and multiple-choice questions. Figure 15 illustrates a sample short-answer question (SAQ) derived from the paragraph, while Figure 16 presents an example of a multiple-choice question (MCQ) generated using the same source.

4.1.2. Semantic Answer Evaluation

The system used two distinct methods for answer validation: direct matching for MCQs and semantic similarity analysis for Short Answer Questions.

- MCQs were validated by direct comparison with the stored correct answer, providing immediate feedback on the user's selection.
- Short Answer Questions were evaluated using semantic vector matching through Weaviate’s hybrid search. The user’s response was converted into a vector and compared to the correct answer’s vector. When the semantic similarity reached a predefined threshold, the system marked the response as correct. Trial runs demonstrated that even paraphrased answers were accurately identified as correct, while irrelevant responses were flagged as incorrect.

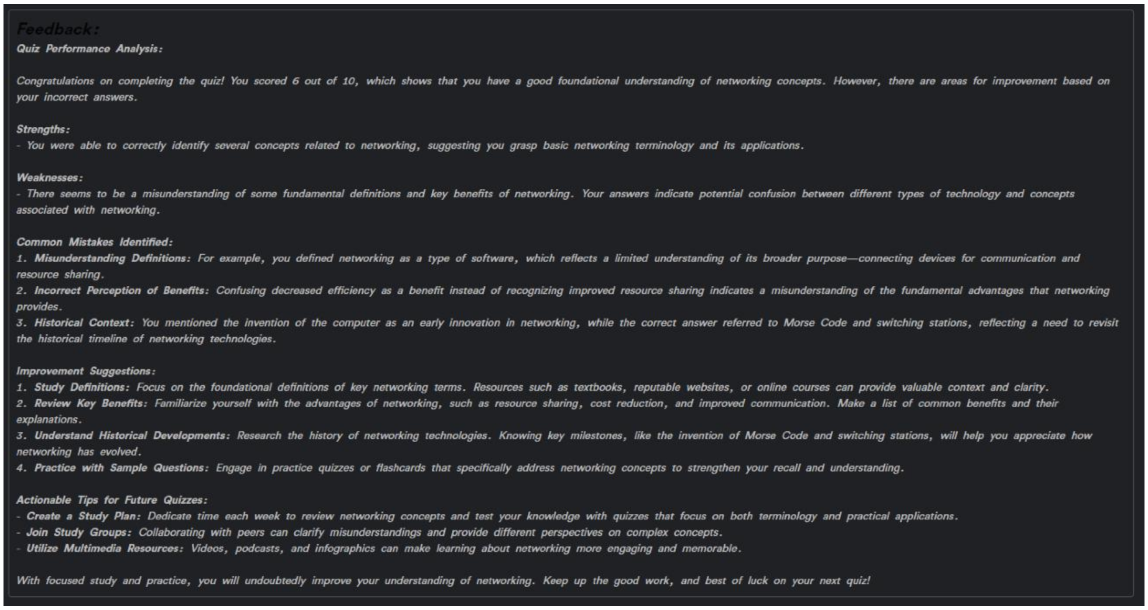


Figure 17 Feedback Displayed After Answer Submission

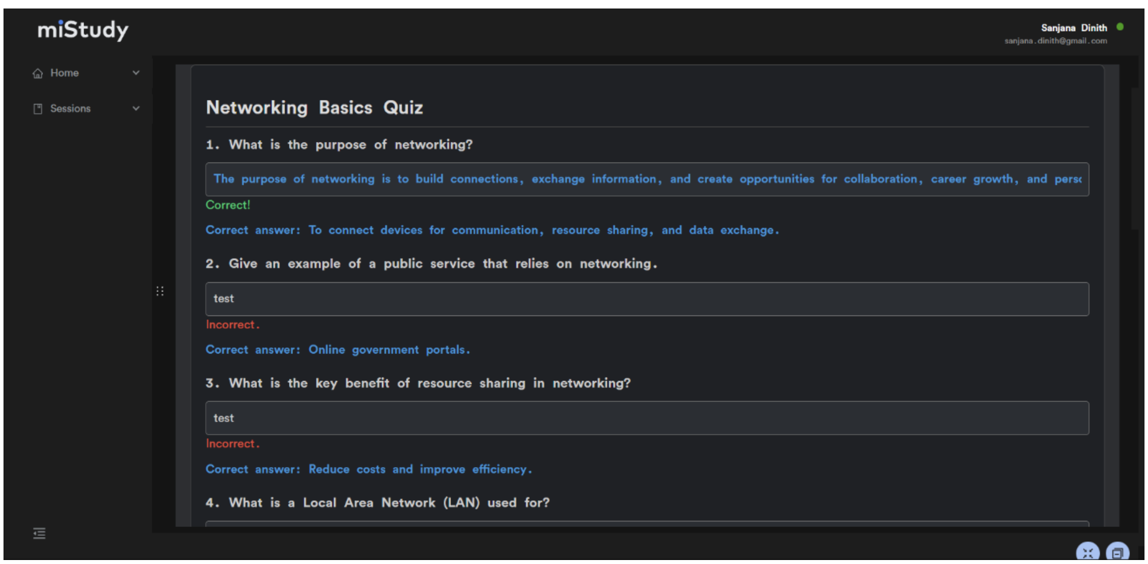


Figure 18 Semantic Search Result (Weaviate Match)

4.1.3. Adaptive Personalization and Feedback

A distinctive feature of the system was its ability to adjust quiz difficulty based on a user's past performance. This dynamic personalization was designed to ensure that users would not be overwhelmed by overly complex questions or bored with too-easy ones.

- Initial Difficulty Level: The first quiz attempt for each user started with a default difficulty level of 50 (Intermediate difficulty), with a balanced mix of lower-order and higher-order cognitive questions.

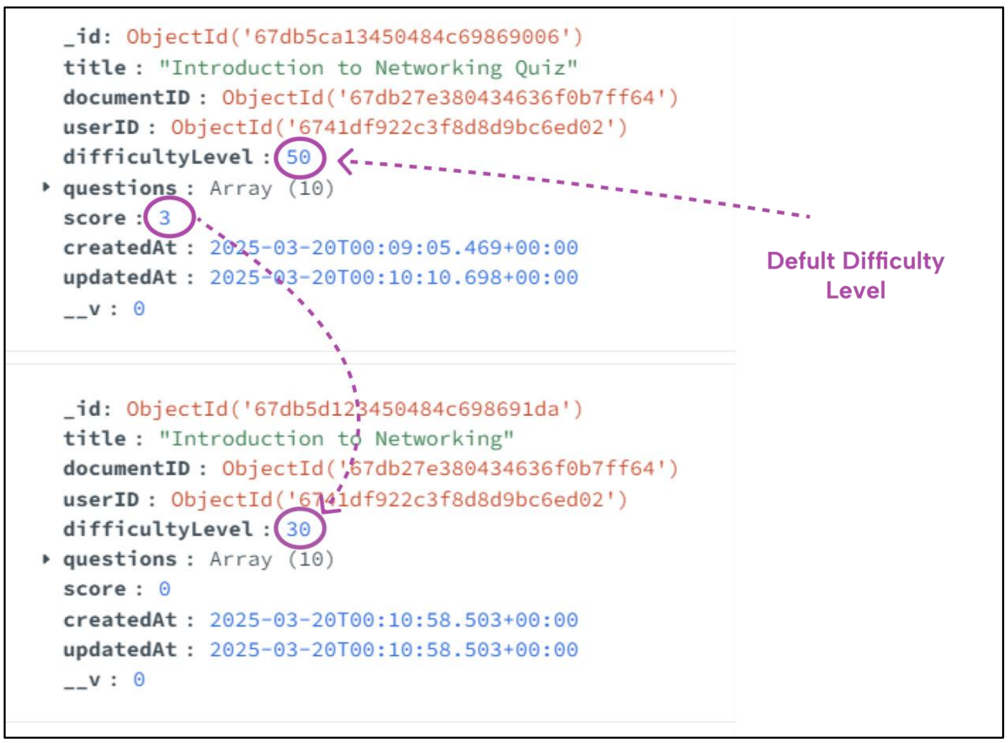
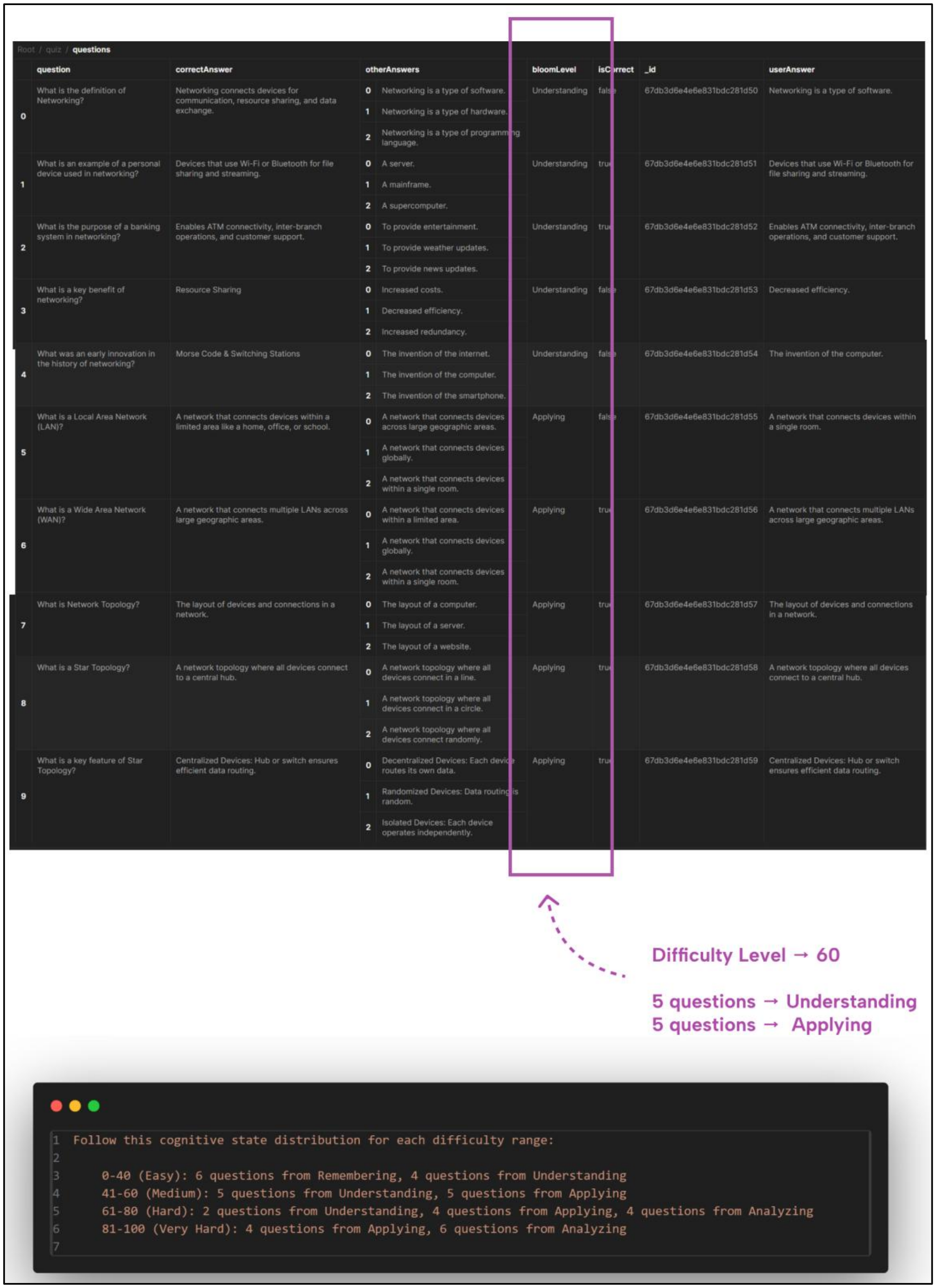


Figure 19 Difficutly Level Adjustment

- Adaptive Difficulty Adjustment: Based on the user's performance in subsequent quiz attempts, the difficulty level was adjusted to match their current competency:
    - High Performance (≥75%): The difficulty was increased, progressing towards higher-order cognitive skills such as Analyzing, Evaluating, and Creating.
    - Moderate Performance (50-74%): The difficulty remained the same, ensuring continued reinforcement of concepts.
    - Low Performance (<50%): The difficulty was decreased to focus on reinforcing basic concepts like Remembering, Understanding, and Applying.

This adaptive mechanism allowed the system to cater to the evolving needs of the user, ensuring that each quiz was neither too challenging nor too simple. The feedback generated for each attempt highlighted areas of strength, weaknesses, and provided actionable recommendations for improvement.



Root / quiz / questions

	question	correctAnswer		otherAnswers	bloomLevel	isCorrect	_id	userAnswer
0	What is the definition of Networking?	Networking connects devices for communication, resource sharing, and data exchange.	0	Networking is a type of software.	Understanding	false	67db3d6e4e6e831bdc281d50	Networking is a type of software.
			1	Networking is a type of hardware.				
			2	Networking is a type of programming language.				
1	What is an example of a personal device used in networking?	Devices that use Wi-Fi or Bluetooth for file sharing and streaming.	0	A server.	Understanding	true	67db3d6e4e6e831bdc281d51	Devices that use Wi-Fi or Bluetooth for file sharing and streaming.
			1	A mainframe.				
			2	A supercomputer.				
2	What is the purpose of a banking system in networking?	Enables ATM connectivity, inter-branch operations, and customer support.	0	To provide entertainment.	Understanding	true	67db3d6e4e6e831bdc281d52	Enables ATM connectivity, inter-branch operations, and customer support.
			1	To provide weather updates.				
			2	To provide news updates.				
3	What is a key benefit of networking?	Resource Sharing	0	Increased costs.	Understanding	false	67db3d6e4e6e831bdc281d53	Decreased efficiency.
			1	Decreased efficiency.				
			2	Increased redundancy.				
4	What was an early innovation in the history of networking?	Morse Code & Switching Stations	0	The invention of the internet.	Understanding	false	67db3d6e4e6e831bdc281d54	The invention of the computer.
			1	The invention of the computer.				
			2	The invention of the smartphone.				
5	What is a Local Area Network (LAN)?	A network that connects devices within a limited area like a home, office, or school.	0	A network that connects devices across large geographic areas.	Applying	false	67db3d6e4e6e831bdc281d55	A network that connects devices within a single room.
			1	A network that connects devices globally.				
			2	A network that connects devices within a single room.				
6	What is a Wide Area Network (WAN)?	A network that connects multiple LANs across large geographic areas.	0	A network that connects devices within a limited area.	Applying	true	67db3d6e4e6e831bdc281d56	A network that connects multiple LANs across large geographic areas.
			1	A network that connects devices globally.				
			2	A network that connects devices within a single room.				
7	What is Network Topology?	The layout of devices and connections in a network.	0	The layout of a computer.	Applying	true	67db3d6e4e6e831bdc281d57	The layout of devices and connections in a network.
			1	The layout of a server.				
			2	The layout of a website.				
8	What is a Star Topology?	A network topology where all devices connect to a central hub.	0	A network topology where all devices connect in a line.	Applying	true	67db3d6e4e6e831bdc281d58	A network topology where all devices connect to a central hub.
			1	A network topology where all devices connect in a circle.				
			2	A network topology where all devices connect randomly.				
9	What is a key feature of Star Topology?	Centralized Devices: Hub or switch ensures efficient data routing.	0	Decentralized Devices: Each device routes its own data.	Applying	true	67db3d6e4e6e831bdc281d59	Centralized Devices: Hub or switch ensures efficient data routing.
			1	Randomized Devices: Data routing is random.				
			2	Isolated Devices: Each device operates independently.				

Figure 20 Question Generation Based on Bloom's Taxonomy

Figure 20 illustrates how questions are generated based on Bloom's Taxonomy levels. In this example, the difficulty level is set to 60. As a result, the system generates a total of 10 questions, 5 targeting the *Understanding* level and 5 targeting the *Applying* level, demonstrating how the difficulty setting influences the cognitive depth of the generated questions.

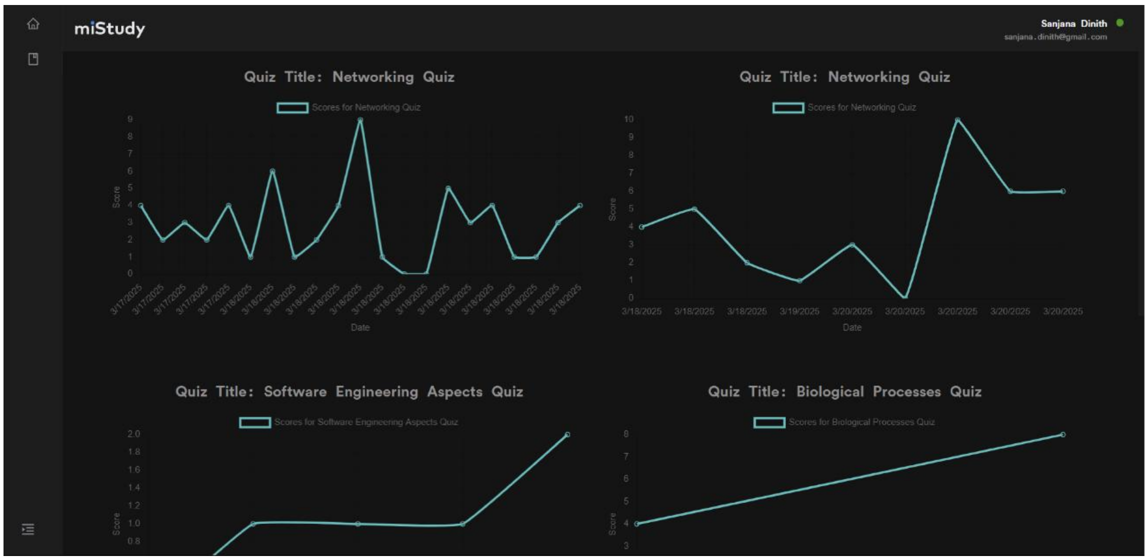


Figure 21 Quiz Attempt History and Performance Graph

Figure 21 displays the quiz attempt history and performance graph for a user. It visualizes the user's scores over multiple quiz attempts, highlighting trends in performance and improvement over time.

4.2. Research Findings

The research findings focus on the effectiveness and impact of the adaptive quiz generation system, particularly in its ability to generate contextually relevant questions and adapt to individual user performance. The personalized nature of the system bridges a significant gap in current educational tools, offering an advanced and responsive learning experience for users.

The system's use of Bloom’s Taxonomy for generating questions ensures a structured approach to learning, moving from basic knowledge recall to higher-order thinking skills. This structure aids in scaffolding users’ understanding of topics, which leads to a more effective learning progression. By dynamically adjusting the quiz difficulty based on user performance, the system effectively personalizes the learning experience, enabling users to engage with content that matches their evolving skill level.

Additionally, the implementation of semantic answer evaluation through Weaviate’s hybrid search system significantly enhances the accuracy and flexibility of answer validation. By leveraging semantic search, the system can assess responses that are contextually similar, even if they are paraphrased. This not only ensures fairness but also accommodates diverse user expressions, which is critical for real-world educational environments.

Furthermore, the feedback mechanism is a key aspect of the system’s effectiveness. The AI-driven feedback helps users identify strengths and areas for improvement, offering actionable insights that support learning enhancement. The dynamic difficulty adjustment ensures that users are continuously challenged while reinforcing foundational concepts, optimizing the learning curve.

The findings also highlight the potential of the system to be incorporated into a wider range of educational settings. With its ability to adapt quiz difficulty and provide real-time feedback, it has the potential to bridge the gap in personalized learning, supporting both students who require reinforcement and those seeking higher-level challenges.

Incorporating such an adaptive learning system into web and mobile platforms would provide users with easy access to personalized quizzes, fostering a deeper understanding of the material and facilitating continuous progress tracking. This approach has the potential to transform traditional quiz-based learning, ensuring that it is tailored to the unique needs of each user, enhancing both engagement and learning outcomes.

The research indicates that adaptive quiz generation, backed by AI-driven feedback and semantic validation, represents a significant advancement in personalized education, offering substantial benefits for users at various learning stages.

4.3. Discussion

The primary objective of this research was to develop a user-friendly, adaptive quiz generation system that enhances personalized learning by integrating artificial intelligence, cognitive learning theory, and semantic evaluation technologies. Drawing from prior literature and technological foundations, the system was carefully designed to meet the educational needs of users seeking tailored and responsive learning experiences.

Throughout the development process, a key learning curve involved aligning pedagogical frameworks, specifically Bloom's Taxonomy, with the capabilities of large language models. This integration enabled the generation of questions across various cognitive levels, ensuring not only knowledge recall but also conceptual understanding, analysis, and creative thinking. Significant effort was dedicated to refining prompt structures to maintain contextual relevance while varying difficulty levels dynamically.

The evaluation of different validation techniques led to the use of a hybrid approach combining deterministic and semantic-based answer checking. While traditional direct answer matching sufficed for MCQs, the more complex nature of Short Answer Questions demanded a more nuanced method. By leveraging Weaviate's semantic search powered by OpenAI embeddings, the system achieved flexible and accurate evaluation, capable of identifying correct answers even when phrased differently from the expected response.

In order to ensure adaptability, the system incorporated a performance-based difficulty adjustment mechanism. This dynamic adaptation encouraged learners to improve without overwhelming them, striking a balance between challenge and support. The personalization logic was integrated into the backend prompt generation layer, allowing each new quiz to be influenced by previous user performance, effectively simulating a tutor-like experience through AI.

To ensure broader accessibility and future scalability, the system's architecture was designed with modularity in mind. While the current prototype focuses on academic PDFs, the framework can be extended to other domains and integrated into mobile or web learning platforms. With the increasing popularity of online education and self-paced learning, such systems can serve as intelligent study companions that evolve alongside the learner.

Overall, the research demonstrates the effectiveness of combining AI language models, educational psychology, and semantic vector technology in building a responsive and intelligent quiz system. The successful implementation of adaptive difficulty, contextual relevance, and personalized feedback underscores the potential of such solutions in shaping the future of digital learning. As AI continues to advance, future iterations of the system can include more refined evaluation metrics, broader subject integration, and enhanced real-time analytics, further enhancing the quality and accessibility of personalized education.

5. CONCLUSION

In conclusion, this research has demonstrated the potential of adaptive learning systems powered by artificial intelligence to transform personalized education. Through the integration of OpenAI's language models, Bloom's Taxonomy, and semantic vector technologies, the system effectively generates and evaluates contextually relevant quiz content tailored to individual learner performance.

The dual-model approach utilizing MongoDB for structured MCQs and Weaviate for semantically-evaluated short-answer questions provides a balanced framework for assessing both foundational knowledge and higher-order thinking. This structure ensures comprehensive cognitive engagement across multiple levels of difficulty and learning objectives.

Moreover, the system's performance-based difficulty adaptation ensures that learners are continually challenged in alignment with their current abilities, fostering both confidence and progress. The AI-driven feedback component enhances this experience by offering personalized insights, corrective suggestions, and encouragement, mimicking the benefits of a human tutor.

The results indicate that such an adaptive quiz generation system holds significant promise for educational settings, particularly in self-paced and remote learning environments. As technology continues to advance, the architecture of this system provides a scalable foundation that can be expanded across disciplines and integrated into broader digital learning platforms.

Future developments may explore the incorporation of real-time analytics, cross-topic knowledge tracking, and voice-based interaction for accessibility. Collaborative efforts with educators and domain experts will be crucial in refining question quality and ensuring pedagogical alignment. Ultimately, this research marks a significant step toward intelligent, learner-centric educational technologies that support continuous, personalized growth.

REFERENCE

[1] Dijkstra, R., Genç, Z., Kayal, S., and Kamps, J., "Reading Comprehension Quiz Generation using Generative Pre-trained Transformers," in *in iTextbooks@ AIED*, 2022.

[2] Zeinalipour, K., Keptiğ, Y.G., Maggini, M., and Gori, M., "Automating Turkish Educational Quiz Generation Using Large Language Models," in *in International Conference on Intelligent Systems and Pattern Recognition*, 2024.

[3] Bloom, B. S.; et al., Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook I: Cognitive Domain, New York, NY, USA: Longmans, Green, 1956.

[4] Srivastava, M., and Goodman, N., "Question Generation for Adaptive Education," arXiv, 2021. [Online]. Available: https://arxiv.org/abs/2106.04262.

[5] Krouska, A.; Troussas, C.; Virvou, M., "Computerized Adaptive Assessment Using Accumulative Learning Activities Based on Revised Bloom's Taxonomy," in *Joint Conference on Knowledge-Based Software Engineering*, Cham, 2018.

[6] Lubbe, A.; Marais, E.; Kruger, D., "Cultivating Independent Thinkers: The Triad of Artificial Intelligence, Bloom's Taxonomy, and Critical Thinking in Assessment Pedagogy," *Education and Information Technologies,* 2025.

[7] V. J. S. a. D. M. Spector, "SCORM 2.0 white paper: Stealth assessment in virtual worlds.," 2008.

[8] Weaviate, "Semantic Search for Education: Enhancing Learning Systems with Hybrid Vector Databases," Weaviate, 2024. [Online]. Available: https://weaviate.io/.

[9] Ledi, Agbeyeye K., "Automatic Quiz Generation System-Using Natural Language Processing," 2022.

[10] L. Kang, "Personalized Quiz Maker: Novel feature for Alby learning management system," 2024.

[11] I. Learning Explorer, "The Science Behind the ScootPad Mastery-Based Adaptive Learning Platform," Learning Explorer, Inc., Campbell, California, USA, 2023.

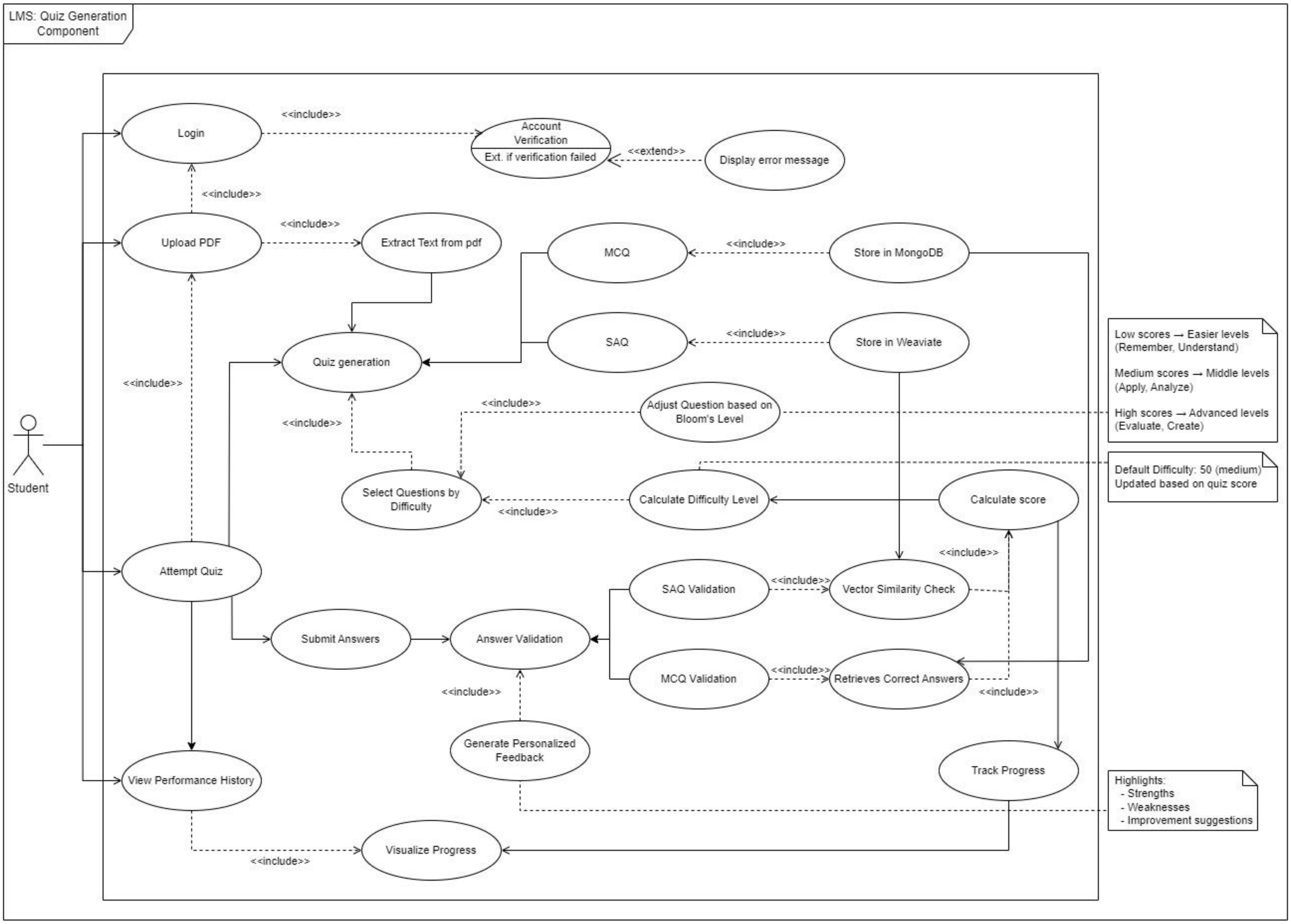
[12] A. K. Alhazmi, A. Imtiaz, F. Al-Hammadi, E. Kaed, "Success and Failure Aspects of LMS in E-Learning Systems," *International Journal of Interactive Mobile Technologies,* vol. 15, no. 11, pp. 4-17, 2021.

[13] M. HQ, "Moodle - Open-source learning platform," [Online]. Available: https://moodle.org/.

[14] B. Inc., "Blackboard Learn – Learning Management System," [Online]. Available: https://www.blackboard.com/.

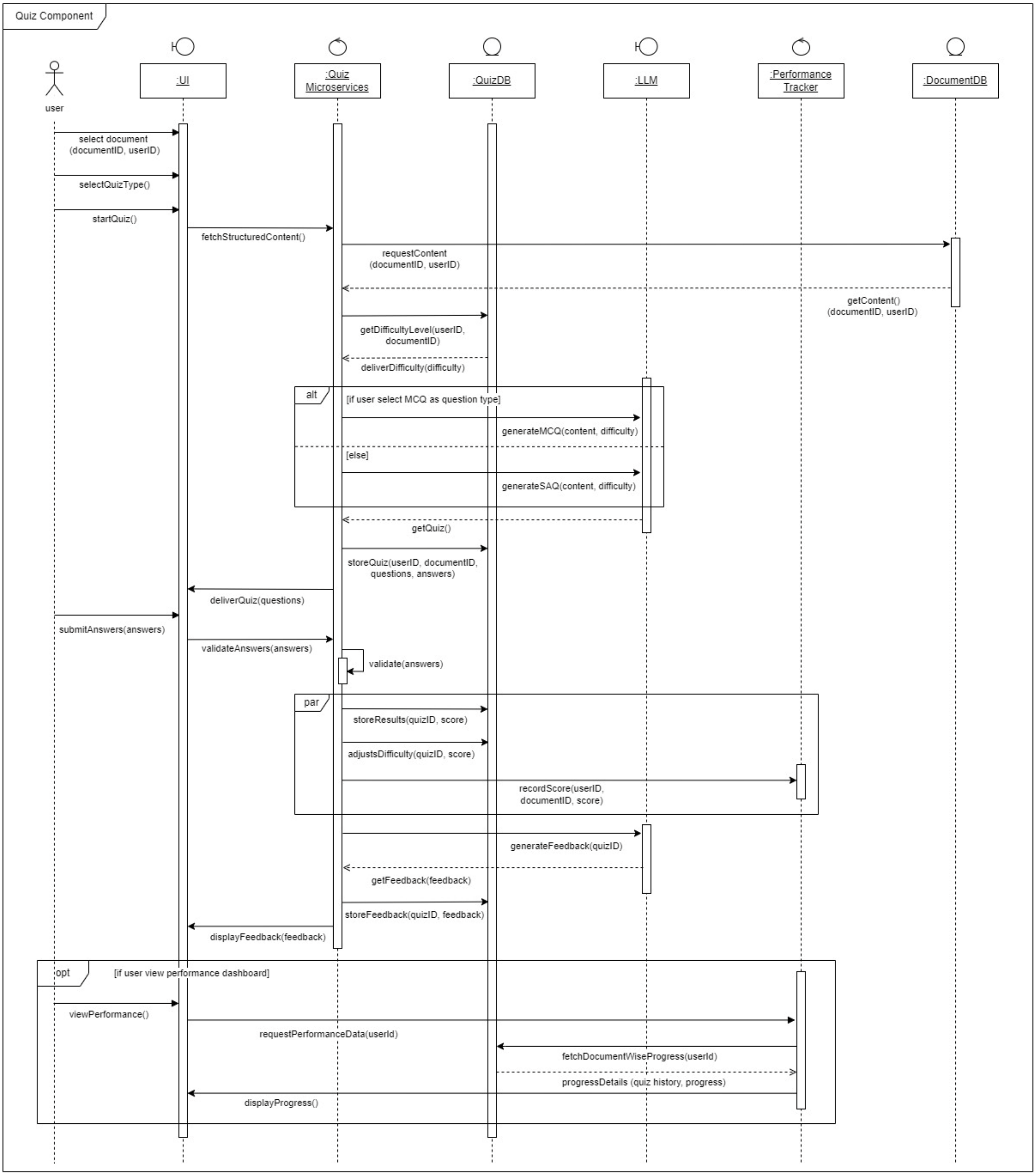
[15] I. Inc., "Canvas LMS by Instructure," [Online]. Available: https://www.instructure.com/canvas.

APPENDIX I



LMS: Quiz Generation Component
Student
Login
<<include>>
Account Verification
Ext. if verification failed
<<extend>>
Display error message
Upload PDF
Extract Text from pdf
MCQ
SAQ
Store in MongoDB
Store in Weaviate
Quiz generation
Adjust Question based on Bloom's Level
Select Questions by Difficulty
Calculate Difficulty Level
Calculate score
Attempt Quiz
SAQ Validation
Vector Similarity Check
Submit Answers
Answer Validation
MCQ Validation
Retrieves Correct Answers
Generate Personalized Feedback
Track Progress
View Performance History
Visualize Progress
Low scores → Easier levels (Remember, Understand)
Medium scores → Middle levels (Apply, Analyze)
High scores → Advanced levels (Evaluate, Create)
Default Difficulty: 50 (medium) Updated based on quiz score
Highlights:
- Strengths
- Weaknesses
- Improvement suggestions

APPENDIX II


Quiz Component
user
:UI
:Quiz Microservices
:QuizDB
:LLM
:Performance Tracker
:DocumentDB
select document (documentID, userID)
selectQuizType()
startQuiz()
fetchStructuredContent()
requestContent (documentID, userID)
getContent() (documentID, userID)
getDifficultyLevel(userID, documentID)
deliverDifficulty(difficulty)
alt
[if user select MCQ as question type]
generateMCQ(content, difficulty)
[else]
generateSAQ(content, difficulty)
getQuiz()
storeQuiz(userID, documentID, questions, answers)
deliverQuiz(questions)
submitAnswers(answers)
validateAnswers(answers)
validate(answers)
par
storeResults(quizID, score)
adjustsDifficulty(quizID, score)
recordScore(userID, documentID, score)
generateFeedback(quizID)
getFeedback(feedback)
storeFeedback(quizID, feedback)
displayFeedback(feedback)
opt
[if user view performance dashboard]
viewPerformance()
requestPerformanceData(userId)
fetchDocumentWiseProgress(userId)
progressDetails (quiz history, progress)
displayProgress()

APPENDIX III

Turnitin Originality Report

Processed on: 12-Apr-2025 00:27 IST
ID: 2608229241
Word Count: 11346
Submitted: 4

IT21360428 - Final Report.pdf By nethmi monali

Similarity Index	Similarity by Source	
9%	Internet Sources:	6%
	Publications:	3%
	Student Papers:	4%

1% match (Internet from 27-Feb-2025)
https://www.coursehero.com/file/202995681/IT17006712pdf/

< 1% match (student papers from 07-Aug-2020)
Submitted to Sri Lanka Institute of Information Technology on 2020-08-07

< 1% match (student papers from 02-Nov-2018)
Submitted to Sri Lanka Institute of Information Technology on 2018-11-02

< 1% match (Internet from 29-Jun-2024)
https://www.srjis.com/downloadPdf/KCE%20Shikaripura%20Final.pdf/7506/225

< 1% match (student papers from 30-Jan-2024)
Submitted to AlHussein Technical University on 2024-01-30

< 1% match (Internet from 03-Apr-2025)
https://consortiacademia.org/wp-content/uploads/2023/v14i01/2025_IJRSE_v14i01_FULL.pdf

< 1% match (Internet from 22-Jan-2025)
https://harbinengineeringjournal.com/index.php/journal/article/download/3711/2169/5923

< 1% match (student papers from 12-Sep-2024)
Submitted to Southern New Hampshire University - Continuing Education on 2024-09-12

< 1% match (student papers from 04-Jan-2018)
Submitted to University of Wales Institute, Cardiff on 2018-01-04

< 1% match (Internet from 29-Jan-2025)
https://www.mdpi.com/2071-1050/15/20/14668