BIOMENTOR - PERSONALIZED E-LEARNING PLATFORM FOR ENGLISH MEDIUM A/L BIOLOGY SUBJECT STUDENTS IN SRI LANKA

Project ID: 24-25J-257

Project Final Report

Dharane. S	IT21068478
Sajeevan.S	IT21204302
Sujitha.S	IT21264634
Srirajan. G.A	IT21375132

B.Sc. (Hons) in Information Technology
Specializing in Software Engineering

Department of Computer Science & Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

BIOMENTOR - PERSONALIZED E-LEARNING PLATFORM FOR ENGLISH MEDIUM A/L BIOLOGY SUBJECT STUDENTS IN SRI LANKA

Project ID: 24-25J-257

Project Final Report

Dharane. S	IT21068478
Sajeevan.S	IT21204302
Sujitha.S	IT21264634
Srirajan. G.A	IT21375132

B.Sc. (Hons) in Information Technology

Specializing in Software Engineering

Department of Computer Science & Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

DECLARATION

I declare that this is my own work, and this Thesis 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 Thesis, 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).

Name	Student ID	Signature
Dharane. S	IT21068478	
Sajeevan.S	IT21204302	
Sujitha.S	IT21264634	
Srirajan. G.A	IT21375132	

The above candidate has carried out this research thesis for the Degree of Bachelor of Science (honors) Information Technology (Specializing in Software Engineering) under my supervision.

Signature of the supervisor
(Dr. Sanvitha Kasthuriarachchi)

Signature of co-supervisor
(Ms. Karthiga Rajendran)

ABSTRACT

The integration of machine learning (ML) in education has opened new avenues for personalized, adaptive, and interactive learning experiences. This research presents **BioMentor**, a web-based learning platform specifically designed for Sri Lankan A/L Biology students, which leverages ML techniques to address key learning challenges such as content overload, low retention, and lack of individualized feedback. The system consists of four main components: ML-based abstractive summarization with voice output to simplify and vocalize complex study material, answer generation and automated evaluation using transformer models to assist students in constructing and refining written responses, adaptive multiple-choice quiz generation that adjusts difficulty using Item Response Theory (IRT) and content relevance via Retrieval-Augmented Generation (RAG), and a spaced repetition module employing the SM-2 algorithm to enhance long-term vocabulary retention. A preliminary survey conducted with A/L Biology students revealed a strong demand for features such as interactive quizzes, summarized notes, flashcards, and automated feedback, all of which are incorporated into BioMentor. The system aims to bridge the gap between traditional learning methods and modern digital tools by providing a curriculum-aligned, ML-driven learning experience. The platform's modular architecture also supports scalability, making it adaptable for other subjects and languages in future iterations.

Keywords- Machine Learning (ML), A/L Biology, Abstractive Summarization, Answer Generation, Answer Evaluation, Adaptive Quiz Generation, Spaced Repetition, Transformer Models, Education Technology, Personalized Learning, Sri Lanka

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to our **supervisor, Dr. Sanvitha Kasthuriarachchi**, and **co-supervisor, Ms. Karthiga Rajendran**, for their continuous guidance, support, and encouragement throughout the development of this project. Their insightful feedback, patience, and dedication were instrumental in helping us shape and refine our ideas from inception to completion. We are also sincerely thankful to our **external supervisor, Ms. Nagalatha Thayaparan**, for her valuable input, suggestions, and support, which added great value to the progress and success of this project.

Furthermore, we extend our appreciation to the lecturers, assistant lecturers, instructors, and academic and non-academic staff of **SLIIT** for their support and assistance throughout the module. We are also grateful to our group members for their collaboration and teamwork. Lastly, we thank our families and friends for their unwavering support, motivation, and understanding, which helped us persevere through challenges and stay committed to completing this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Description
ML	Machine Learning
LLM	Large Language Model
OCR	Optical Character Recognition
RAG	Retrieval-Augmented Generation
NLP	Natural Language Processing
TTS	Text-to-Speech
MP3	MPEG Audio Layer-3
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
FAISS	Facebook AI Similarity Search
gTTS	Google Text-to-Speech
API	Application Programming Interface
IRT	Item Response Theory
SM-2	Spaced Repetition Algorithm (Version 2)
UI	User Interface
UX	User Experience
BERT	Bidirectional Encoder Representations from Transformers
LLaMA	Large Language Model Meta AI
TF-IDF	Term Frequency–Inverse Document Frequency
PDF	Portable Document Format
DOCX	Document Open XML
PPT	PowerPoint Presentation
TXT	Text File

VM	Virtual Machine
CPU	Central Processing Unit
GPU	Graphics Processing Unit

1. INTRODUCTION

1.1 Background Study and Literature Review

1.1.1 Background Study

In the rapidly evolving landscape of education, the integration of intelligent technologies has become essential to overcoming traditional learning barriers and supporting diverse learner needs. Particularly in Sri Lanka, the General Certificate of Education Advanced Level (A/L) Biology curriculum poses significant challenges due to its depth, complexity, and extensive terminology. Students enrolled in the English medium stream are often required to master vast content within limited timeframes, all while adhering to a rigid, examination-focused structure. Conventional classroom methods characterized by uniform content delivery, rote memorization, and a lack of adaptive feedback fall short of addressing individual learning differences, leading to disengagement, reduced comprehension, and inconsistent academic outcomes.

To address these systemic challenges, **BioMentor** has been developed as a personalized web-based e-learning platform specifically tailored to the A/L Biology curriculum. The platform leverages state-of-the-art Artificial Intelligence (AI) techniques to transform the conventional learning experience into a dynamic, student-centered process. Through the integration of **Retrieval-Augmented Generation (RAG)** for contextual content delivery, **transformer-based abstractive summarization** for syllabus-aligned knowledge condensation, **adaptive multiple-choice quizzes** calibrated via Item Response Theory (IRT), **automated answer evaluation**, and **spaced repetition algorithms**, BioMentor delivers a holistic and scalable educational solution.

Each component of the system is meticulously designed to enhance different stages of the learning process ranging from content understanding and engagement to self-assessment and long-term retention. Summarization tools allow students to digest complex biology topics efficiently, while adaptive quizzes tailor question difficulty to the learner's evolving skill level. The platform's answer evaluation engine provides personalized feedback, guiding students to refine their conceptual understanding.

Moreover, spaced repetition techniques reinforce vocabulary and critical concepts over time, rooted in cognitive science to optimize memory retention.

BioMentor supports multimodal learning by incorporating features such as **voice output**, **notes generation**, and **multilingual content delivery** (Sinhala and Tamil), making it inclusive and accessible to a broader demographic. Designed using modular architecture, the system ensures future scalability, allowing the integration of additional subjects and features without disrupting the core experience.

Ultimately, BioMentor bridges the gap between outdated pedagogical models and the demands of modern learners. It not only empowers students to take control of their own academic progress but also serves as a proof-of-concept for future intelligent learning platforms tailored to domain-specific curricula. By aligning cutting-edge AI with national education goals, BioMentor paves the way for more personalized, efficient, and inclusive digital learning experiences in Sri Lanka and beyond.

1.1.2 Background survey

The rapid development of machine learning (ML), particularly transformer-based models and large language models (LLMs), has significantly influenced the evolution of educational technologies. These advancements have made it possible to design systems that not only adapt to learner needs but also personalize the delivery, assessment, and reinforcement of academic content. In the context of secondary education, especially in subjects with complex content such as A/L Biology, traditional learning methods often fall short in meeting individual student needs. Learners are typically limited by static resources, rigid lesson structures, and the absence of immediate, personalized feedback. This gap calls for intelligent, curriculum-aligned solutions that leverage ML capabilities to enhance comprehension, engagement, and retention. BioMentor, a web-based learning platform tailored specifically for Sri Lankan A/L Biology students, addresses this challenge through four core components: abstractive summarization with voice output, answer generation and evaluation, adaptive quiz generation, and vocabulary memorization using spaced repetition.

To validate the platform's relevance and align its features with actual learner needs, a preliminary survey was conducted among 113 A/L Biology students. The results, illustrated in Figure 1, clearly indicated a demand for technology-enhanced learning tools. A majority of students expressed strong interest in interactive quizzes and summarized notes, followed closely by flashcards and detailed explanations. Additionally, 37% of students reported they would use a subject-specific platform on a daily basis, and 25% indicated weekly usage. This feedback highlighted the urgent need for a personalized, high-frequency digital learning environment, directly influencing the functional design of each BioMentor component.

The first component focuses on ML-based abstractive summarization with voice output. Traditional summarization methods primarily relied on extractive techniques, which select fragments of text verbatim from the source. However, recent ML approaches such as Flan-T5 and BART allow for abstractive summarization, where models generate human-like summaries that capture the core meaning of content in a condensed, paraphrased form[1]. To further enhance relevance and accuracy, the Retrieval-Augmented Generation (RAG) technique is employed. RAG improves summarization by incorporating external knowledge through semantic retrieval, ensuring that the model outputs contextually accurate and syllabus-aligned content[2][3]. The system also integrates voice output capabilities using text-to-speech (TTS), allowing auditory learners and visually impaired students to engage with content effectively[4].

The second core module involves ML-driven answer generation and automated evaluation. LLMs such as LLaMA and BERT, when fine-tuned with domain-specific data, are capable of generating detailed and context-relevant answers to structured and essay-style questions[6][7]. In BioMentor, these models are trained using past paper formats and structured biology content to mimic human-like, exam-ready responses. The platform evaluates student answers using a hybrid scoring mechanism that combines semantic similarity (via models like SciBERT), lexical similarity (TF-IDF and Jaccard), and grammar analysis to ensure fair, accurate, and insightful feedback[8][9]. This not only helps students gauge their understanding but also provides actionable suggestions for improvement—something rarely available in traditional learning environments.

The third component is the adaptive MCQ quiz generation system, which applies ML to generate and customize multiple-choice questions based on student ability. The system utilizes fine-tuned models like T5 and LLaMA to produce high-quality, curriculum-aligned MCQs automatically[10][20]. Additionally, Item Response Theory (IRT) is used to personalize quiz difficulty levels in real time by analyzing a student's past performance and response behavior[11][12]. This results in adaptive testing, where students are presented with questions that match their current level of understanding, gradually increasing in complexity to challenge them appropriately. The integration of Retrieval-Augmented Generation also ensures that questions are contextually grounded in syllabus content, improving both relevance and accuracy[13][14].

The fourth and final component of BioMentor addresses vocabulary memorization through adaptive spaced repetition. In subjects like Biology, retaining complex terms and definitions is essential. Traditional rote learning methods often lead to quick forgetting, whereas spaced repetition algorithms like SM-2, derived from cognitive psychology, optimize review intervals to improve long-term memory[15][16]. BioMentor’s vocabulary system uses SM-2 to analyze user performance and dynamically schedule reviews, emphasizing terms that the student struggles with while extending intervals for those already mastered. The system enhances this experience through gamification elements, including flashcards, progress tracking, and achievements to maintain engagement[17][18][19]. These techniques not only improve retention but also promote active learning and self-discipline.

Together, these four ML-powered components make BioMentor a comprehensive, intelligent, and adaptive learning platform designed specifically for the needs of Sri Lankan A/L Biology students. By combining advanced ML models with syllabus-based design and incorporating direct feedback from potential users through a preliminary survey (Figure 1), BioMentor addresses the gaps in traditional educational tools—providing a personalized, engaging, and effective solution that aligns with both academic standards and modern learning behaviors. Its modular and scalable architecture further allows for future expansion into additional subjects and languages, offering a robust foundation for broader educational impact.

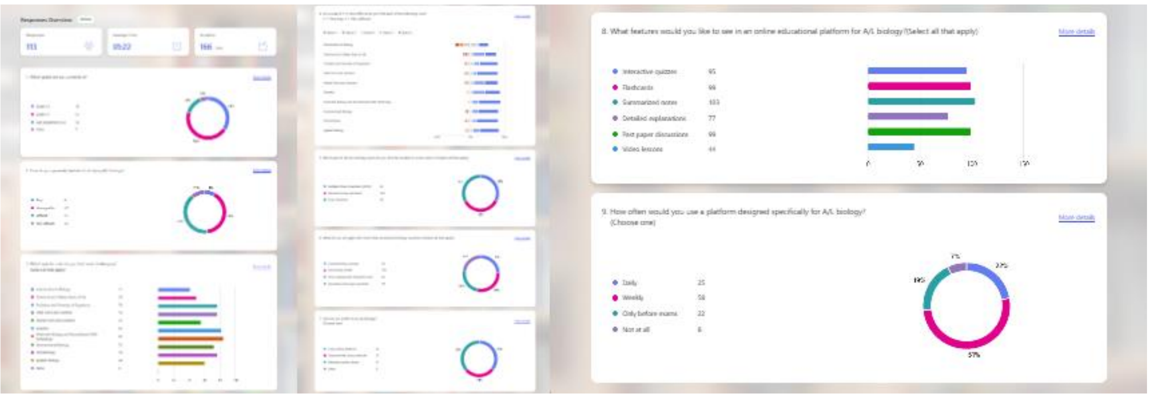

Figure 1: Survey details

1.2 Research Gap

Figures 2, 3, 4 and 5 illustrate the research gap of each of the components.

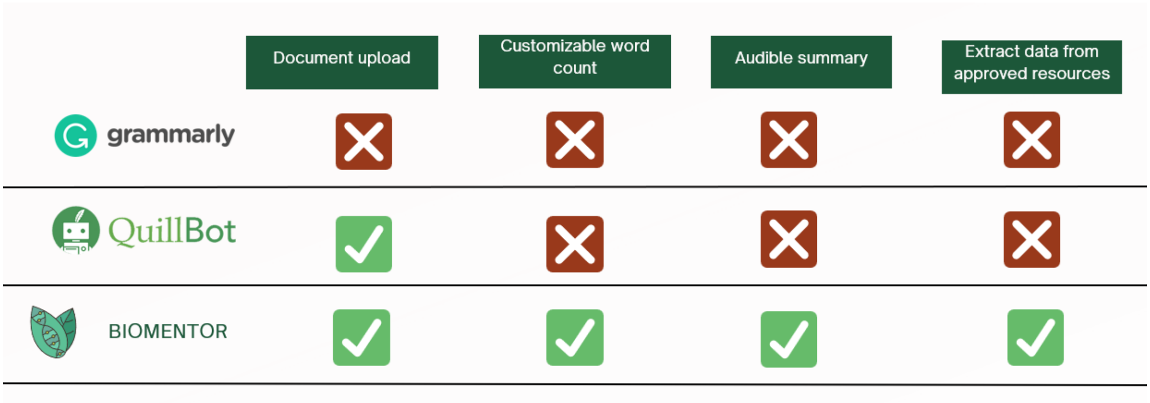

Figure 2: Research Gap for Summarization

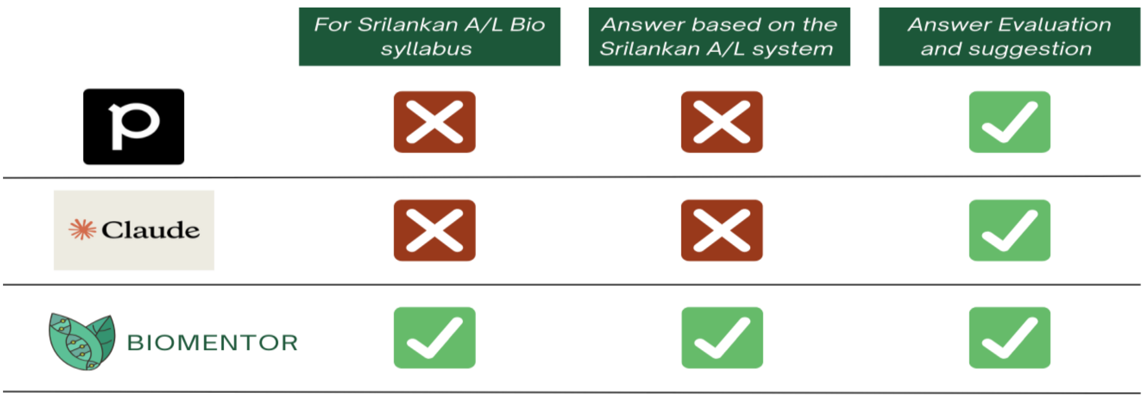

Figure 3: Research Gap for Question And Answering

Figure 4: Research Gap For Adaptive Quiz Platform

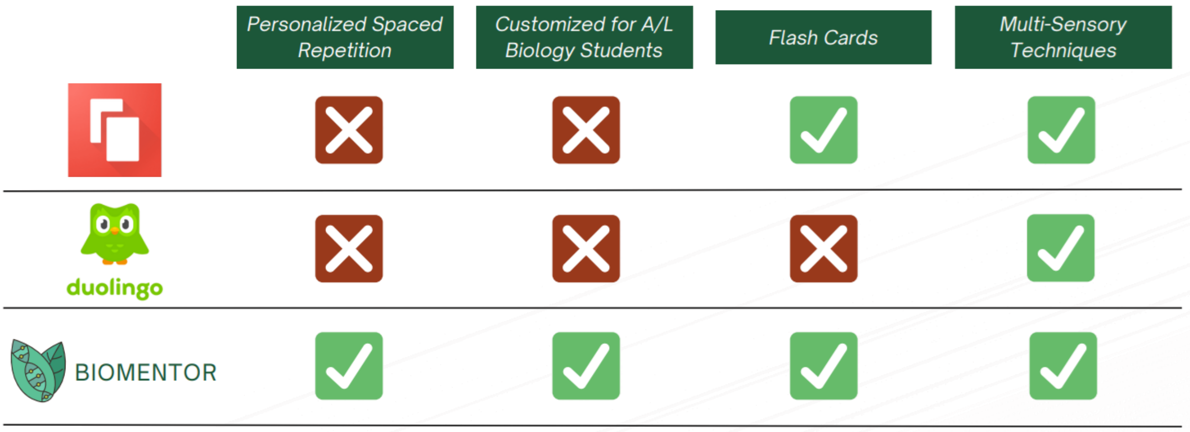

Figure 5: Research Gap for Adaptive Spaced Repetition Platform

1.3 Research Problem

In the Sri Lankan educational landscape, the General Certificate of Education Advanced Level (A/L) Biology curriculum represents one of the most rigorous and content-heavy academic challenges for students. Particularly for those studying in the English medium, this subject presents a dual-layered difficulty: mastering the scientific content itself and navigating the complexity of academic language, which includes highly technical terminology, long-form theoretical explanations, and conceptual frameworks. The traditional approach to teaching Biology at the A/L level predominantly revolves around passive content delivery, textbook-centered instruction, and examination-focused revision strategies. These methods, while effective in some contexts, often fall short of addressing the individualized needs of students who struggle with comprehension, retention, and the ability to apply knowledge critically across diverse scenarios.

In most classroom environments, the teaching process is standardized, with little room for variation based on student ability or learning style. Teachers are frequently bound by time constraints and pressured to complete the syllabus within fixed academic calendars, which leaves limited opportunity to provide personalized guidance or address specific areas where students may be falling behind. Consequently, learners are left to rely on rote memorization, intensive tuition classes, and the repetition of past examination questions. While these strategies may help in the short term, they rarely contribute to a deeper understanding of biological processes or encourage long-term knowledge retention. As a result, many students experience gaps in conceptual clarity, lack motivation due to disengagement, and face difficulties in articulating structured answers during exams.

Compounding this issue is the limited availability of digital learning platforms that are purpose-built for the Sri Lankan A/L Biology curriculum. Although e-learning adoption has grown in recent years, most available platforms offer general science resources that are not aligned with the local syllabus, and they often present content in a static, non-interactive format. These platforms typically fail to offer features that can adjust to a student's evolving knowledge level, track performance over time, or reinforce content through targeted revision. The absence of curriculum-specific design, adaptive assessments, and structured revision tools significantly reduces the educational value of such platforms, leaving students with fragmented resources that do not fully support their academic journey.

An equally critical challenge lies in how students review and retain previously learned material. Without strategic reinforcement, much of the content covered earlier in the academic year tends to be forgotten, especially in a subject like Biology where terms, diagrams, and explanations must be consistently recalled and connected. Current learning systems, whether traditional or digital, rarely incorporate cognitive learning principles that support retention, such as structured repetition or timely feedback based on performance trends. Students often revise inefficiently typically in bulk closer to exams resulting in high levels of stress and cognitive overload, and minimal improvement in actual understanding.

In this context, there is an urgent need for a specialized, well-structured, and student-centered learning platform that is tailored to the unique demands of the A/L Biology syllabus in Sri Lanka. Such a platform should be capable of delivering content in a way that supports varied learning styles, allows students to progress at their own pace, and provides timely and meaningful feedback based on their learning behaviors. It must offer tools that support continuous learning through concept summaries, topic-wise assessments, structured answer writing practice, and effective review systems based on proven learning techniques. Furthermore, the platform should promote learner independence, enabling students to take ownership of their academic progress by offering insights into their strengths, weaknesses, and performance trends over time.

This research addresses the critical gap in subject-specific, personalized learning tools for A/L Biology students in Sri Lanka. The aim is to design and implement a comprehensive, web-based educational platform that not only aligns with the national curriculum but also enhances student comprehension, engagement, and academic outcomes through structured, responsive, and user-friendly learning experiences. By focusing on the educational realities of local students and addressing the limitations of both traditional instruction and existing digital tools, this research seeks to contribute a scalable and impactful solution to the evolving needs of science education in the country.

1.4 Research Objectives

1.4.1 Main Objective

The main objective of this research is to develop **BioMentor**, personalized e-learning platform specifically designed for Sri Lankan A/L Biology students studying in English medium. The platform aims to enhance the overall learning experience by machine learning techniques such as transformer-based abstractive summarization, Retrieval-Augmented Generation (RAG), adaptive quiz generation, generation of answers for structured and essay-type questions aligned with the A/L Biology syllabus, and spaced repetition. By leveraging these technologies, BioMentor seeks to provide a customized and interactive learning environment that addresses individual student needs, improves subject comprehension, boosts knowledge retention, and ultimately enhances academic performance in the A/L Biology curriculum.

1.4.2 Specific Objectives

- **To develop an AI-based summarization system** using transformer models (e.g., Flan-T5) and Retrieval-Augmented Generation (RAG) to generate concise, syllabus-aligned summaries of A/L Biology content.
- **To implement an automated question-answering and evaluation system** that can generate structured and essay-type answers, assess student responses using hybrid evaluation metrics (semantic, lexical, grammatical), and provide adaptive feedback.
- **To implement a multi-stage moderation system** that filters student queries using rule-based and deep learning methods to ensure appropriate, high-quality educational interactions.
- **To design an adaptive multiple-choice quiz system** that dynamically adjusts question difficulty based on individual student performance using transformer models and Item Response Theory (IRT).
- **To integrate a spaced repetition module** powered by the SM-2 algorithm to improve long-term vocabulary retention through personalized review intervals and gamified learning experiences.
- **To provide multimodal learning support**, including voice output for summaries and multilingual translation (Sinhala/Tamil), to enhance accessibility and engagement for diverse learners.
- **To evaluate the effectiveness of the BioMentor platform** through performance metrics, user feedback, and expert validation, ensuring its alignment with educational goals and curriculum standards.

1.4.3 Business Objectives

- **To introduce a scalable and cost-effective e-learning solution** tailored specifically for Sri Lankan A/L Biology students, filling the market gap for localized, subject-specific digital education tools.
- **To position BioMentor as a leading ML-powered educational platform** in the Sri Lankan edtech sector by leveraging cutting-edge technologies like LLMs, RAG, and adaptive learning algorithms.
- **To generate sustainable revenue** through subscription models, institutional partnerships with schools and tutoring centers, and premium feature offerings (e.g., personalized mentorship, analytics dashboards).
- **To expand the platform's user base** by targeting English medium A/L Biology students initially, with plans to scale to other subjects, languages (Sinhala and Tamil), and education levels.
- **To enhance educational equity and accessibility** by offering affordable pricing tiers and scholarship programs for underprivileged students in rural and low-income communities.
- **To establish strategic collaborations** with educational institutions, government bodies, and edtech investors to promote national adoption and long-term growth.

2. METHODOLOGY

2.1 Introduction

This part illustrates the methodology for approach the proposed system's related functions. This is a methodological way of research. Follow the software lifecycle model to implement the system. Research has conducted more studies on the above research area. Hence, the gathered information will be used to achieve the main objectives and sub-objectives.

2.2 System overview

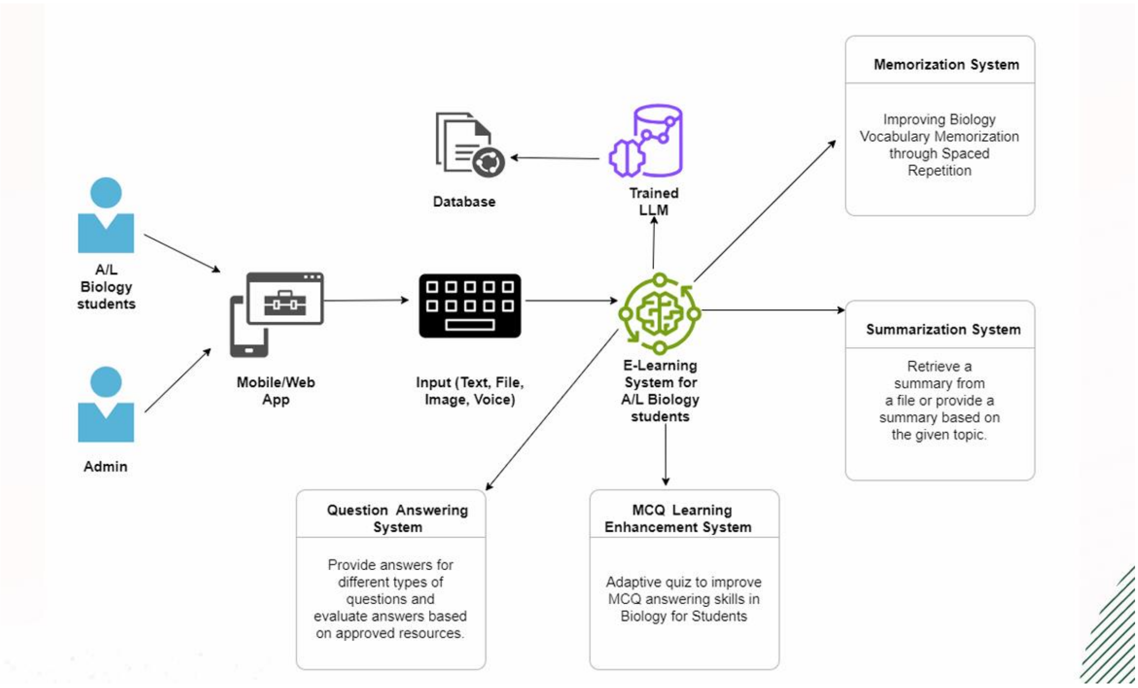

Figure 6: the system diagram of the system

2.3 Component overview

Figure 7 shows the component details of the system.

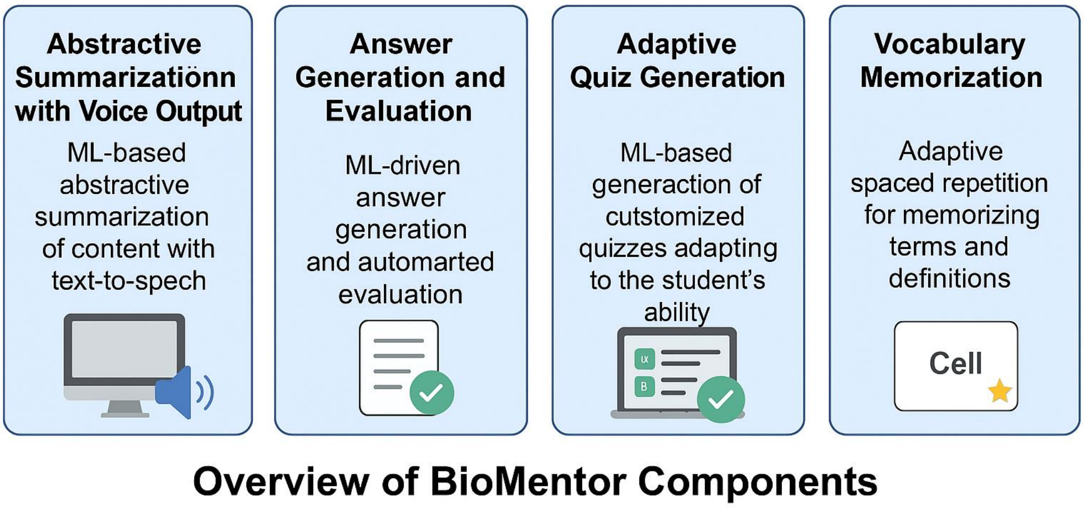

Figure 7: Component Details

Component 1: LLM-Based Abstractive Text Summarization Tool with Voice Output implemented in different software architectures

This component is designed to enhance student engagement and well-being during online learning by leveraging facial emotion analysis. It utilizes advanced computer vision and machine learning techniques to analyse the facial expressions of students during video classroom sessions. The primary goal is to detect and manage stress levels, monitor drowsiness, and gauge attention levels in real-time. Figure 8 illustrates the system overview of this component.

Methodology:

This component presents a smart summarization system specifically designed for A/L Biology, based on a fine-tuned Flan-T5 Base model. It enables abstractive text summarization, structured note creation, audio generation, and multilingual translation (limited to notes). Created to improve curriculum alignment and accessibility, it enables students to access concise and contextually relevant content via document uploads, topic-related questions, or plain text input.

The component was developed and assessed in two software architectures, monolithic and microservices to identify the most efficient deployment approach. The monolithic architecture showed enhanced performance regarding response time, deployment speed, and ease of debugging, making it a better fit for the present use case. According to these findings, the monolithic version was incorporated into the BioMentor platform. Figure 8 displays the system overview diagram of this component.

Methodological Approach:

- Summarization with Flan-T5: Executes abstractive summarization of biological material utilizing a fine-tuned extensive language model.

- RAG (Retrieval-Augmented Generation): Improves topic-focused searches by fetching pertinent biology information before summarizing.
- Note Creation: Generates organized academic notes that correspond to topics in the biology curriculum.
- Audio Output: Utilizes Google Text-to-Speech (gTTS) to create MP3 files of summaries and notes for auditory education.
- Multilingual Assistance: Converts created notes into Tamil and Sinhala, enhancing accessibility for local students.
- Architecture Analysis: Implemented in both monolithic and microservices architectures; performance indicators like response time, CPU/RAM consumption, deployment difficulty, and fault tolerance were evaluated to inform integration.

Deployment: The completed monolithic version was deployed on an Azure VM with a Nginx reverse proxy to enhance accessibility and manage system services more effectively.

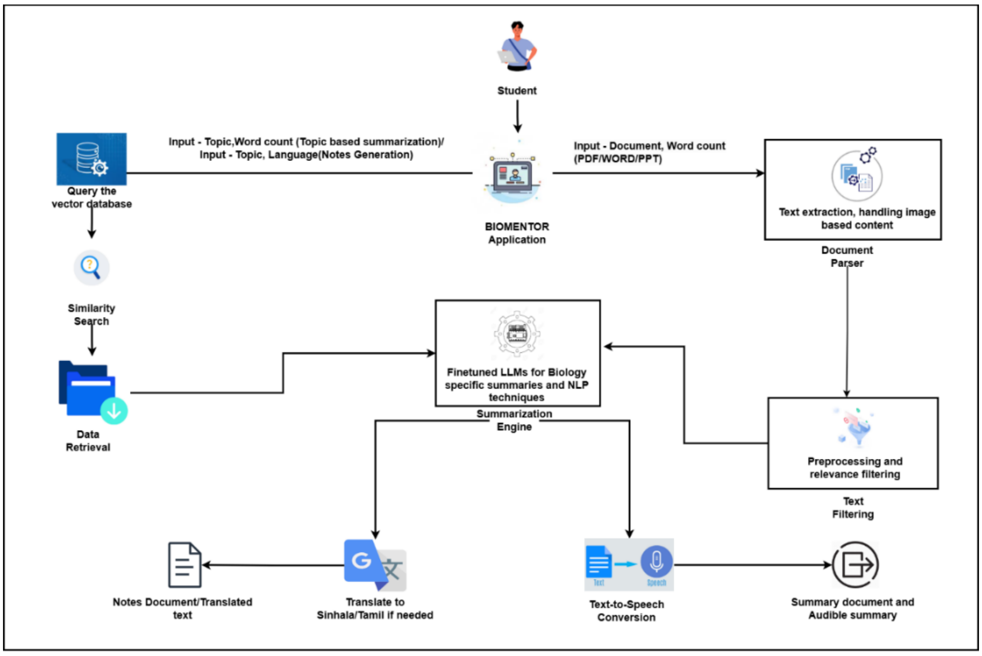

Figure 8: System diagram for Summarization

Component 2: LLM Based Provide answers for structured and essay type of questions and evaluate answers based on approved resources.

This component focuses on enhancing self-learning for Sri Lankan A/L Biology students by generating syllabus-aligned answers for both structured and essay-type questions, and evaluating student responses using advanced natural language processing techniques. It empowers students to evaluate their answers independently without mentor support, offering real-time feedback, personalized study recommendations, and performance insights through an intelligent, curriculum-specific system.

Methodology:

- Answer Generation Module: Utilized the fine-tuned LLAMA 3-Instruct model to generate structured and essay-type answers based on the A/L Biology syllabus. Post-processing with Gemini API enhanced grammar, coherence, and academic tone.
- Answer Evaluation Engine: Implemented a hybrid scoring system combining SciBERT (semantic similarity), TF-IDF, Jaccard similarity, spaCy (keyword matching), and LanguageTool (grammar analysis). Provided a comprehensive score with dimensional feedback.
- Question Moderation: Deployed a BERT-based classification model via Hugging Face Spaces to verify that submitted questions are syllabus-aligned and academically appropriate, rejecting off-topic or irrelevant entries.
- Feedback System: Generated structured feedback highlighting grammar issues, missing/extra keywords, and conceptual weaknesses. Offered students performance analytics, trend tracking, and progress insights.
- Study Recommendation System: Used SentenceTransformers and FAISS indexing to match weak areas with semantically relevant textbook content. Recommended targeted study materials to improve weak concepts and reinforce learning.

- Adaptive Learning & Dashboard: Maintained student-wise evaluation records in MongoDB, displaying personalized dashboards showing progress graphs, keyword mastery, and improvement paths.
- Backend & API Services: Developed with FastAPI, exposing modular APIs for question submission, answer generation, evaluation, analytics, and recommendations. All data stored in MongoDB.
- Frontend Interface: Built using React.js and Tailwind CSS to ensure responsiveness across devices. Integrated modal dialogs for feedback, real-time updates, and an intuitive learning experience.
- Deployment & CI/CD: Hosted on Microsoft Azure using an Ubuntu VM. Backend served via FastAPI, frontend via Nginx, and automated deployment handled using GitHub Actions.

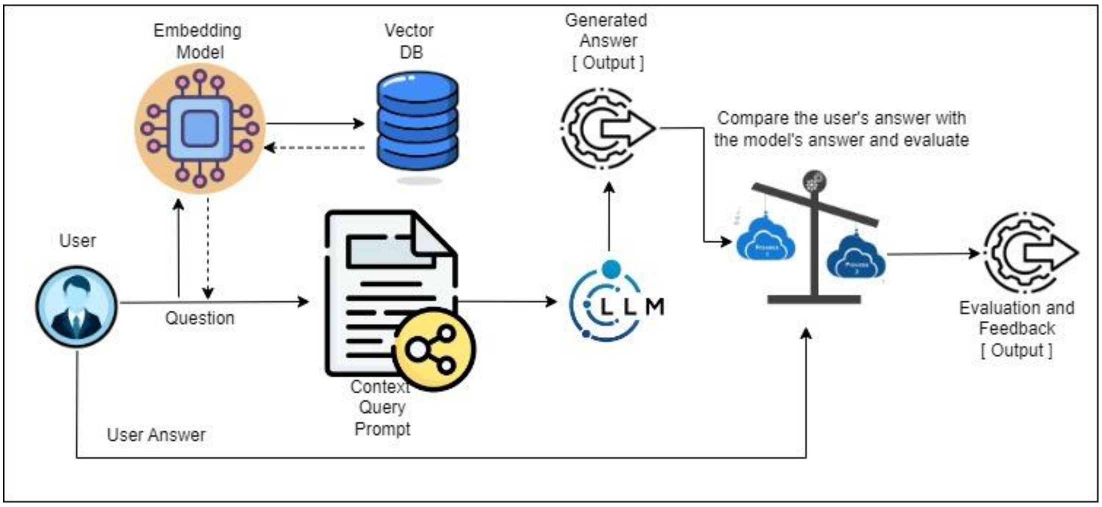

Figure 9: System Diagram For Question And Answering

Component 3: LLM-Based Adaptive Quiz Platform for A/L Biology Students with Performance-Driven MCQ Generation

This component is developed to enhance personalized learning and assessment for A/L Biology students by delivering adaptive, syllabus-aligned multiple-choice quizzes. It uses a fine-tuned large language model, llama-2-7b-chat-hf, to dynamically generate questions. To ensure each quiz attempt is unique yet relevant, the system incorporates Retrieval-Augmented Generation (RAG), which retrieves similar sample questions from a curated dataset and uses them as context for the LLM. The platform also integrates an adaptive logic framework based on Item Response Theory (IRT) to adjust question difficulty based on student performance. Real-time scoring, analytics, and performance tracking are provided through a dedicated dashboard. Figure 9 illustrates the system overview of this component.

Methodology:

- **Adaptive Quiz Engine**: Implements Item Response Theory (IRT) to adapt question difficulty in real-time based on performance metrics such as accuracy and response patterns during each quiz.
- **Dynamic LLM Based Question Generation**: Utilizes the fine-tuned llama-2-7b-chat-hf model to generate syllabus-aligned multiple-choice questions with diverse phrasing and structure.
- **Retrieval-Augmented Generation (RAG):** For each quiz generation request, the system retrieves a sample MCQ from a curated dataset using semantic similarity search (FAISS). This sample is fed into the LLM as part of the prompt to ensure that the generated questions are both relevant to the syllabus and different with each attempt, avoiding repetition.
- **Unit-wise Quiz Module**: Allows students to take unit-specific quizzes based on chapters or sections of the Biology syllabus, supporting focused revision and concept mastery.

- **Performance Tracking and Analytics**: Provides detailed dashboards showing topic-wise accuracy, score trends, and leaderboard positioning, enabling students to monitor their academic progress over time.
- **Quiz History and Retry Functionality**: Maintains a record of all completed quizzes and allows students to revisit or retry previous adaptive quizzes for further practice.

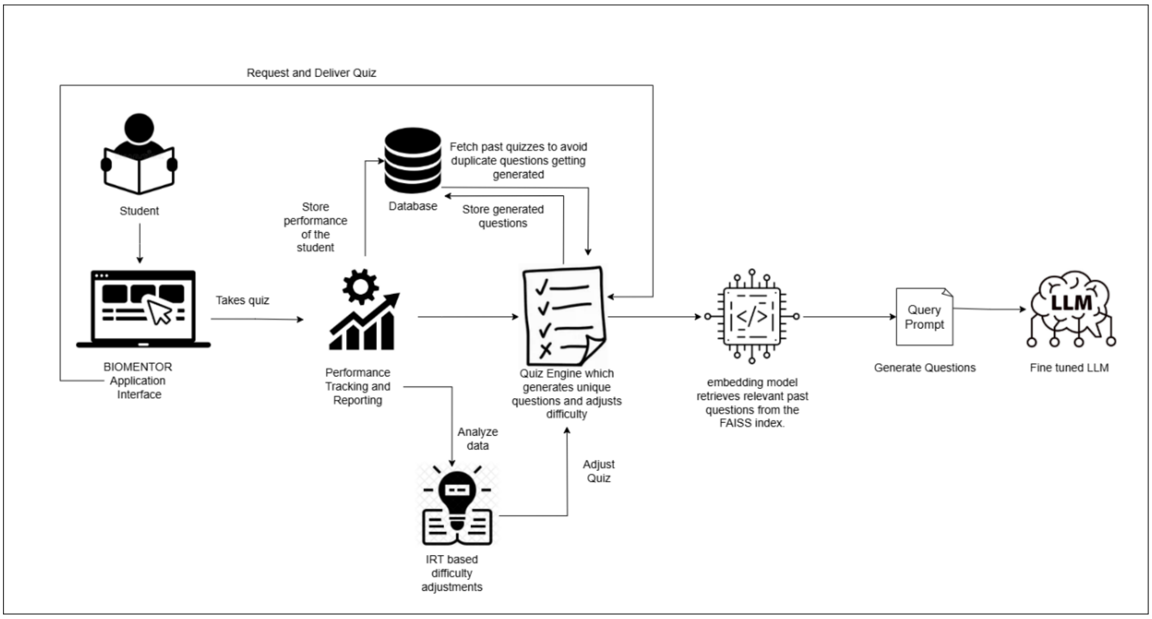

Figure 10:System Diagram for Adaptive Quiz Platform

Component 4: Enhancing Vocabulary Memorization through Adaptive Spaced Repetition

By combining cognitive science concepts with entertaining user experiences through gamification, this project uses Adaptive Spaced Repetition (ASR) to improve vocabulary memorisation. The system creates a smooth, dynamic learning environment by fusing a React frontend with a FastAPI backend. Fundamentally, the approach optimises memory retention and learning efficiency by dynamically modifying review intervals based on individual performance using the SM-2 algorithm.

Methodology:

Core Algorithm: SM-2 Adaptation

- Review schedules are modified using the SM-2 algorithm according to recall quality.
- The formula for updating the Ease Factor (EF) is

$$EF = EF + (0.1 - (5 - q) \times (0.08 + (5 - q) \times 0.02))$$

- Where q is the recall quality (rated 0–5). Review periods are longer for higher ratings and shorter for lower ratings.
- This strategy makes sure that the harder things are examined more often and the easier ones are evaluated more infrequently.

Backend (FastAPI)

- JWT-based authentication secures access and manages user sessions.
- MongoDB stores user performance metrics, review history, and spaced repetition intervals.
- Calculating real-time review schedules based on user input and learning history.
- Users can resume studies from the last point with personalized review data.

Frontend (React)

- Core learning tool featuring annotated visuals and pronunciation aids to support different learning styles.
- Leaderboards, badges, and streak trackers boost user engagement and encourage consistent practice.
- Clean, intuitive design enables users to track progress and navigate content with ease.

Real-Time Adaptation & Visualization

- Review intervals adjust in real-time based on user recall accuracy and attempt frequency.
- Offer insights into learning trends, strengths, and areas for improvement.

Gamification for Motivation

- Rewards users with badges for daily practice and milestones.
- Encourage friendly competition and peer motivation.
- Increase in difficulty to maintain an optimal learning curve.

Evaluation & Feedback

- Monitors session frequency, recall accuracy, and module engagement to assess learning outcomes.
- Collected to continuously improve usability, interface design, and algorithm effectiveness.

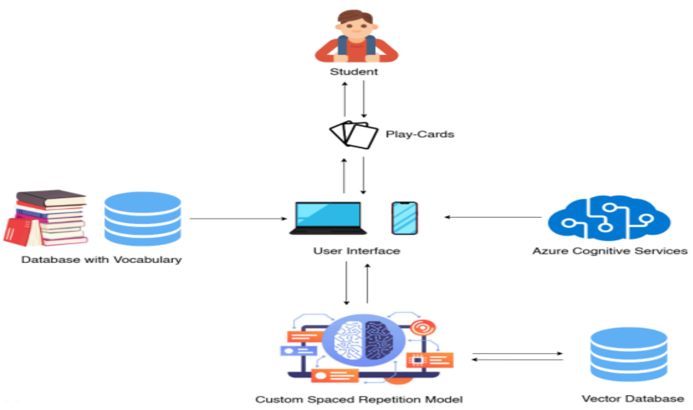

Figure 11: Adaptive Spaced Repetition System Diagram

2.3. Development Process

The development of **BioMentor**, a personalized AI-powered web-based learning platform for A/L Biology students in Sri Lanka, followed the **Agile development methodology**. Agile's iterative, feedback-driven approach was ideal for this project, where flexibility, rapid prototyping, and continuous refinement were crucial to meet evolving educational requirements and user expectations. Below is an overview of how the Agile process was applied during the development of BioMentor:

1. **Project Initiation**
    - ➢ The project began with a clear objective: to develop an intelligent learning system that enhances understanding, retention, and engagement for A/L Biology students.
    - ➢ The project began with a clear objective: to develop an intelligent learning system that enhances understanding, retention, and engagement for A/L Biology students.

2. **Product Backlog Creation**
    - ➢ A product backlog was created containing prioritized features, user stories, and technical tasks across different modules including summarization, quiz engine, vocabulary builder, and student analytics.
    - ➢ The backlog was updated regularly based on internal review, educator insights, and user feedback gathered during requirement analysis.

3. **Sprint Planning**
    - ➢ The development timeline was divided into multiple sprints, each focused on building and refining specific features or components.
    - ➢ In sprint planning meetings, tasks were selected from the backlog based on priority and feasibility. Clear sprint goals and estimated workloads were defined for each cycle.

4. **Daily Standup Meetings**
    - The team conducted short daily standups to discuss task progress, resolve blockers, and align development objectives.
    - These meetings promoted team collaboration and ensured consistent momentum throughout each sprint.

5. **Development and Testing**
    - Developers implemented core functionalities such as the summarization engine, quiz generation logic, answer generation and evaluation and the spaced repetition system.
    - Testing was conducted in parallel to development - unit, module, integration, and system testing were performed to ensure functional accuracy and stability.

6. **Collaboration and Feedback**
    - Continuous collaboration was maintained among developers, domain experts, and UI/UX designers.
    - Periodic demos were held to showcase progress to stakeholders, including A/L teachers and students, allowing the team to incorporate real-time feedback and improve the platform.

7. **Review and Adaptation**
    - At the end of each sprint, review meetings were conducted to present completed features and gather stakeholder feedback.
    - A sprint retrospective was held to assess what worked well and what could be improved in the next iteration.

8. **Continuous Integration and Deployment**
    - Features were integrated continuously to ensure code compatibility and system functionality.

- Regular builds and deployments allowed quick testing of new updates and helped maintain a stable development environment.

9. **Scaling and Release**
    - As development progressed, new features were added and refined based on stakeholder input and testing outcomes.
    - The platform was prepared for release to a broader user base, with scalability in mind to accommodate future expansions in content and user load.

10. **Ongoing Improvement**
    - Following the Agile principle of continuous improvement, the team regularly reviewed workflows, refined backlogs, and enhanced product quality.
    - This cycle of iterative development, validation, and enhancement continues to drive BioMentor's evolution as a modern, intelligent learning tool for students.

By adopting Agile methodology, the BioMentor development team ensured a flexible, user-centric process that responded effectively to both technical challenges and real-world educational needs. Figure 12 shows the Agile-based development life cycle used for BioMentor.

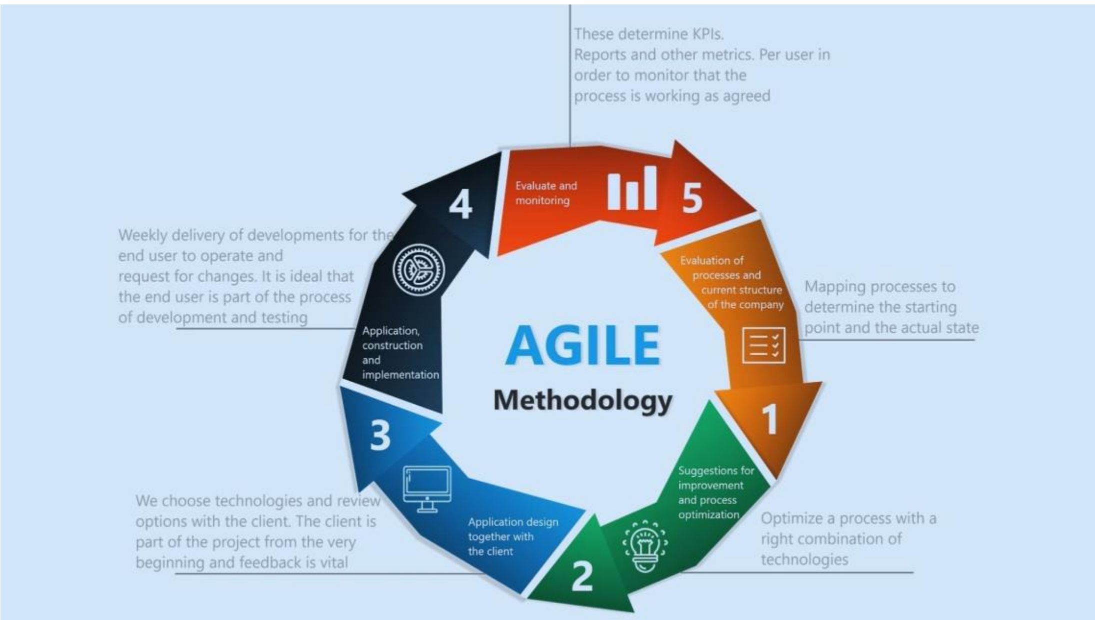

Figure 12: Agile based Development Lifecycle

Project Management

Effective project management and version control were crucial to the successful development of the BioMentor platform. The team employed **Jira** for task management and sprint planning, and **GitHub** for version control, ensuring efficient collaboration, timely deliveries, and consistent progress tracking throughout the project lifecycle.

1. **Task Planning and Organization**
   - **Jira** served as the core platform for managing all development tasks. Epics and user stories were created for major components such as summarization, adaptive quizzes, answer evaluation, and spaced repetition.
   - Tasks were organized into sprints and assigned to team members based on their roles and technical expertise.

2. **Priority Setting**

- Each task in Jira was assigned a priority label (e.g., Critical, High, Medium, Low) to help the team focus on time-sensitive and essential development areas first, especially model integration and backend APIs.

3. **Task Dependencies**
   - Jira's linking feature was used to define task dependencies. For example, model fine-tuning was marked as a prerequisite for implementing the answer generation module.
   - This structured flow helped minimize blockers and ensured seamless handovers between tasks.

4. **Deadline Tracking**
   - Sprint deadlines and individual task due dates were set in Jira's roadmap and timeline view, keeping the team aligned with overall project milestones and academic deliverables.

5. **Progress Monitoring**
   - Real-time status updates (To Do, In Progress, Done) and burndown charts allowed the team to monitor sprint progress and make adjustments as needed during weekly standups and review meetings.

6. **Collaboration and Communication**
   - Jira's commenting system enabled team members to communicate within each task, ask clarifying questions, share updates, and attach relevant documentation.
   - GitHub was integrated with Jira for seamless tracking of commits, branches, and pull requests, ensuring transparency in development and version control.

By leveraging Jira's agile project management capabilities and integrating it with GitHub, the BioMentor development process remained well-structured, transparent, and adaptive, ultimately contributing to the successful delivery of the platform.

2.3.2 Requirement Gathering

The requirement gathering phase was a critical foundation for the development of BioMentor, ensuring that the platform effectively addressed the real-world needs of Sri Lankan A/L Biology students. A combination of qualitative and quantitative methods was used to collect functional and non-functional requirements from key stakeholders, including students, teachers, and subject matter experts.

1. **Stakeholder Interviews**

- One-on-one interviews were conducted with A/L Biology teachers and students from both urban and rural areas to understand the challenges faced in traditional learning environments.
- Insights were gathered regarding preferred learning styles, difficulties with comprehension, and expectations from a digital learning tool.

2. **Surveys and Questionnaires**

- Structured surveys were distributed among English medium A/L Biology students to collect data on their study habits, technology usage, pain points in learning, and interest in AI-assisted tools.
- Feedback highlighted the need for adaptive quizzes, summarized content, and automated feedback mechanisms.

3. **Curriculum Analysis**

- The official Sri Lankan A/L Biology syllabus was thoroughly reviewed to ensure that the platform's features (e.g., summarization, MCQ generation) aligned with examination standards and learning objectives.
- This analysis informed the training datasets and feature designs for the summarization and quiz modules.

4. **Competitive Analysis**

- Existing e-learning platforms and educational tools were studied to identify gaps in personalization, subject specificity, and AI integration.

- Findings emphasized the lack of systems tailored specifically to A/L Biology in Sri Lanka, especially for the English medium demographic.

5. Technical Feasibility Study

- An initial review of AI models, data sources, and available APIs was performed to validate the feasibility of implementing components such as RAG, spaced repetition, and transformer-based summarization within the given timeline and technical constraints.

6. Requirement Documentation

- Based on the insights gathered, the team documented functional requirements (e.g., quiz creation, answer evaluation, summarization) and non-functional requirements (e.g., system performance, scalability, user interface responsiveness).
- These were categorized and managed through **Jira** for traceability and iterative refinement during development.

This comprehensive approach to requirement gathering ensured that the BioMentor platform was user-centric, syllabus-aligned, and technically viable, laying a strong groundwork for the subsequent design and development phases.

2.5.1 Development Methodology

a. LLM-Based Abstractive Text Summarization Tool with Voice Output implemented in different software architectures

This component was developed to generate educational summaries, structured notes, and audio outputs from A/L Biology content. It supports features like document and topic-based summarization, topic-based notes generation, voice synthesis, and translation of notes into Sinhala and Tamil. Built using a fine-tuned Flan-T5 model and a RAG-based semantic retrieval system, the tool enables students to access curriculum-aligned content in multiple formats. It was implemented in both monolithic and microservices architectures for performance analysis, with the monolithic version integrated into the final system due to its simplicity, faster response time, and ease of maintenance.

- Data Preparation:
  The summarization system was created using two organized datasets, one for training the model with biological texts and their summaries, and the other for facilitating semantic searches in topic-related inquiries. The data was cleaned using Python by eliminating duplicates, null values, and special characters, while also standardizing the text format. This guaranteed top-notch inputs for model fine-tuning and FAISS-driven content retrieval in line with A/L Biology material.

- Model Fine-Tuning:
  A Flan-T5 Base model was refined with the cleaned dataset in Google Colab utilizing GPU acceleration. Inputs were prepared with instructional prefixes (e.g., "summarize:") and tokenized to conform to the T5 model architecture. The training utilized Hugging Face’s Trainer API, focusing on enhancing ROUGE-based summarization quality.

- Summarization Backend:

  Built using FastAPI, the backend provides multiple endpoints to serve various summarization needs:

  - /process-document: Accepts document uploads (PDF, DOCX, PPTX, TXT), extracts and cleans text, and returns summarized output in PDF and MP3 formats.
  - /process-query: Generates summaries based on topic queries using FAISS-based semantic retrieval from a pre-indexed educational dataset.
  - /summarize-text: Takes raw user input as plain text and generates concise summaries with audio and PDF download options.
  - /generate-notes: Creates well-structured notes from a given topic, with optional translation into Tamil and Sinhala for regional accessibility.
  - /generate-notes: Creates well-structured notes from a given topic, with optional translation into Tamil and Sinhala for regional accessibility.

- Multilingual Support:

  Note generation includes optional translation to Tamil and Sinhala, using the deep_translator package. Translated notes are returned as plain text, while English notes are downloadable as PDFs.

- Voice Output:

  Summaries and notes are converted to speech using gTTS, generating MP3 audio files. Audio playback and downloads are supported by auditory learners and visually impaired students.

- Software Architectures Compared:

Two architectures were implemented:

- Monolithic: Faster response time, easier deployment and debugging.
- Microservices: More modular and fault-tolerant but with higher resource usage and complex orchestration.

Monolithic was selected for final deployment due to its responsiveness and suitability for frontend integration.

- Deployment Pipeline:

The final model was hosted on Hugging Face Hub and integrated into the backend running on a Microsoft Azure VM. The backend was containerized and made accessible via Nginx reverse proxy for production readiness. A GitHub Actions CI/CD pipeline was implemented to automate deployment on commits.

- Frontend Integration:

A fully responsive React.js frontend connects to the summarization API, offering modules for:

- Document summarization with audio and PDF output.
- Topic-based summarization with semantic search.
- Note generation with language selection. Each UI section includes modal interfaces for input, real-time alerts, audio playback, and download options.

b. LLM Based Provide answers for structured and essay type of questions and evaluate answers based on approved resources.

This component was developed to allow Sri Lankan A/L Biology students to generate answers for both **structured and essay-type questions**, evaluate their responses without mentor support, and receive real-time feedback and study recommendations. The system integrates advanced NLP techniques including **fine-tuned LLAMA 3-Instruct**, **SciBERT**, **TF-IDF**, **Jaccard similarity**, and a **BERT-based moderation model**, all aligned to the national syllabus. Key features include curriculum-aligned answer generation, grammar and keyword-based feedback, semantic performance evaluation, question filtering, and personalized study material suggestions. The backend was developed in a modular format using FastAPI, while the frontend used React for real-time interaction. The system was deployed on Azure, using GitHub Actions for CI/CD and containerized with production-level scalability.

- **Data Preparation**

Three custom datasets were developed:

- **Q&A Dataset** for model training, containing structured and essay-type A/L Biology questions and high-quality answers.
- **Study Notes Dataset** used for semantic search and study material recommendations.
- **Question Acceptability Dataset** labeled for training the BERT moderation model.

Data preprocessing included removing duplicates, stop words, and irrelevant content, as well as formatting for tokenization and model ingestion. Sentence embedding vectors were generated using SentenceTransformers for semantic indexing.

- **Model Training**

  The LLAMA 3-Instruct model was fine-tuned on a combined dataset of structured and essay-type A/L Biology questions and answers using Google Colab Pro with GPU acceleration. The training data was cleaned, tokenized, and formatted with instructional prompts to guide the model. Using Hugging Face's Transformers and Trainer API, the model was trained to generate contextually accurate, syllabus-aligned answers. Training progress was monitored through evaluation loss and sample outputs to ensure quality and relevance across both question types.

- **Evaluation Engine**

  A hybrid scoring engine was developed using:

  - **SciBERT** for semantic similarity,
  - **TF-IDF** and **Jaccard** for lexical and keyword match analysis,
  - **spaCy** for keyword extraction,
  - **LanguageTool** for grammar checks.

  Final scores were calculated by combining semantic, grammatical, and keyword level insights. Each score was broken down and visualized in the user dashboard.

- **Question Moderation System**

  A **BERT-based classification model** was trained using the Question Acceptability Dataset to identify off-topic or vague submissions. This model was deployed on Hugging Face Spaces and integrated via API into the backend to automatically filter questions before evaluation.

- **Feedback and Study Recommendation Engine**

  The feedback engine provides:

  - Grammar improvement tips,
  - Missed keyword highlights,
  - Conceptual understanding guidance.

  The study recommendation engine uses FAISS and SentenceTransformers to match weak areas in student responses with relevant textbook sections, suggesting targeted study materials.

- **Backend and API Development**

  The backend was built with **FastAPI** and included:

  - /generate-answer: Uses **LLAMA 3-Instruct** to generate structured or essay-type answers based on the given question.
  - /evaluate-answer: Evaluates student responses using the **hybrid engine** (SciBERT, TF-IDF, Jaccard, spaCy, and LanguageTool).
  - /student-analytics: Retrieves performance analytics and evaluation trends for a specific student based on past interactions.
  - /get-student-question/{student_id}: Fetches assigned **past paper-style questions** for a student by their ID.
  - /evaluate-passpaper-answer: Evaluates a student's answer by comparing it to a previously stored model answer from the past paper question set.

- **Frontend Integration**

  A fully responsive **React.js frontend** was created, offering:

  - Answer submission and evaluation forms,
  - Real-time feedback display,
  - Progress dashboards with evaluation trends and keyword mastery,

- Study recommendation results linked to topic areas.
- Pass paper submission and evaluation

Modal UIs with alerts, feedback cards, and responsive design ensured smooth user experience on both mobile and desktop devices.

- **Deployment Pipeline**

  The system was deployed on a **Microsoft Azure Ubuntu VM**. Backend services were containerized and served via **Nginx**. Deployment was automated using GitHub Actions for CI/CD:

  - Auto-builds and deployments on every push,
  - Auto-builds and deployments on every,
  - Service restart and health check integration,
  - Error logging and runtime diagnostics.

c. LLM-Based Adaptive Quiz Platform for A/L Biology Students with Performance-Driven MCQ Generation

This component was developed to provide A/L Biology students with personalized, syllabus-aligned multiple-choice quizzes that adapt in real-time to their performance level. It integrates a fine-tuned LLaMA 2 model (llama-2-7b-chat-hf) for dynamic MCQ generation, enhanced through Retrieval-Augmented Generation (RAG) to ensure contextual accuracy and uniqueness in each attempt. The adaptivity of the system is built using Item Response Theory (IRT) principles, allowing the difficulty level to adjust based on student performance. The platform also includes unit-wise quiz selection, performance analytics, and quiz history tracking. Backend services were implemented using FastAPI, and the frontend was developed in React to support responsive, real-time user interaction.

- **Data Preparation**

  A curated dataset was constructed by collecting MCQs and their corresponding answers from:

  - Past A/L examination papers
  - School term test papers
  - Model question papers

  Each question was **tagged with a difficulty level** (*easy*, *medium*, *hard*) and **categorized by unit**, with the direct support of an experienced **A/L Biology teacher (external supervisor)**. This tagging was essential to enable both adaptive scaling and unit-wise filtering. The dataset was cleaned to remove duplicates and irrelevant items and indexed using **SentenceTransformers** for semantic similarity-based retrieval. This structured dataset supports both the **RAG module** and the **unit-wise quiz generator**.

- **Model Training**

  The **llama-2-7b-chat-hf** model was fine-tuned using a curated prompt–response dataset built from collected MCQs. Fine-tuning was performed on **Google Colab** Pro using GPU acceleration. Instruction-style prompts were crafted to train the model to:

  - Generate structured, syllabus-aligned MCQs
  - Vary phrasing and cognitive complexity
  - Follow specific styles similar to A/L paper formats

The Hugging Face **Transformers + PEFT (LoRA)** approach was used to fine-tune the model efficiently, ensuring low resource consumption and high accuracy. Training was monitored using loss metrics and qualitative checks on output quality.

- **Adaptive Quiz Generation and RAG Pipeline**

  When a student starts an adaptive quiz, the system:

  - Uses semantic similarity search (FAISS) to retrieve a sample MCQ from the dataset.
  - Feeds that question as context to the fine-tuned llama-2-7b-chat-hf model using Retrieval-Augmented Generation (RAG).
  - Generates a complete quiz with diverse and syllabus-aligned MCQs that are non-repetitive and appropriate for the user's level.

  This RAG-enhanced generation ensures that each quiz attempt is unique and grounded in familiar question styles.

- **Adaptivity with Item Response Theory (IRT)**

  After quiz submission, the system applies IRT logic to evaluate performance and estimate student ability. It considers response accuracy and consistency to adapt future quiz difficulty levels. This ensures learners are continuously challenged at an appropriate level without being overwhelmed.

- **Unit-Wise Quiz Module**

  Students can attempt quizzes based on specific units or chapters. The system randomly selects questions from the manually tagged dataset corresponding to the selected unit. This supports focused revision and reinforces mastery of individual topics

- **Performance Analytics and Dashboard**

  The system includes a comprehensive performance dashboard designed to help students monitor their academic progress effectively. It displays key insights such as topic-wise accuracy, score trends over time, leaderboard rankings, and a detailed quiz attempt history. By visualizing these metrics, the dashboard enables students to identify their strengths and weak areas, track improvement, and make informed decisions about which topics to revisit. This continuous feedback loop supports personalized learning and encourages consistent engagement with the platform.

- **Backend and API Development**

  Developed in **FastAPI**, the backend provides:

  - /generate-quiz: Creates an adaptive quiz using RAG + LLaMA 2
  - /submit-quiz: Evaluates the submitted answers using IRT-based logic
  - /get-dashboard-data: Returns performance summaries and history
  - /get-unit-quiz: Retrieves random questions based on selected unit
  - /retry-quiz: Triggers a new adaptive quiz based on prior performance

- **Frontend Integration**

The **React.js** frontend delivers:

- ➢ Adaptive and unit-wise quiz interfaces
- ➢ Real-time result summaries
- ➢ Progress tracking through visual dashboards
- ➢ Retry functionality and quiz history access

UI elements are responsive and optimized for both desktop and mobile.

- **Deployment Pipeline**

Deployment is managed on a **cloud-hosted Ubuntu VM**, with services containerized and served via **Nginx**.

GitHub Actions CI/CD automates:

- ➢ Code builds and container deployment
- ➢ Backend service restarts
- ➢ Runtime diagnostics and logging

d. Vocabulary Memorization through Adaptive Spaced Repetition

The Adaptive Spaced Repetition (ASR) system was developed using the **Agile development methodology**, which supports flexibility, iterative progress, and continuous user feedback. This methodology was particularly suitable for building an educational platform that must adapt to diverse learner behaviors, cognitive responses, and engagement patterns. By dividing the development into manageable phases and integrating regular testing and feedback loops, the team ensured that each component of the system aligned with both pedagogical goals and user experience standards.

The development process unfolded in several key phases:

- **Requirement Analysis and Planning**

  During this phase, the development team identified the core objectives of the system, such as implementing the SM-2 algorithm, personalizing learning experiences, and integrating gamification. Stakeholder inputs were collected to define functional requirements (e.g., user authentication, review scheduling, performance tracking) and non-functional requirements (e.g., responsiveness, scalability, accessibility).

- **System Architecture and Design**

  The application was designed with a modular architecture to support maintainability and scalability. The backend, built with FastAPI, handled all core logic, including user management and adaptive scheduling, while the frontend, developed with React, offered an intuitive and engaging interface.

  - **Backend (FastAPI):**
    - Managed JWT-based authentication for secure access
    - Stored learning progress and scheduling data in MongoDB
    - Implemented the SM-2 algorithm for interval adjustment
    - Handled performance analytics and progress history

  - **Frontend (React):**
    - Provided an interactive flashcard interface

- Delivered real-time feedback with pronunciation and image support
- Integrated gamification features like streaks, leaderboards, and badges
- Enabled users to view dashboards tracking learning progress

- **SM-2 Algorithm Integration**

A core component of the backend was the SM-2 algorithm, which determines how review intervals should be spaced. The equation used to update the Ease Factor (EF) is:

$$EF = EF + (0.1 - (5 - q) \times (0.08 + (5 - q) \times 0.02))$$

Here, q is a quality rating from 0 to 5 based on user recall. This formula helps personalize the learning curve by decreasing intervals for poorly recalled items and increasing them for well-remembered ones. Additional enhancements were introduced to include retrieval duration and response history in the calculation.

- **User-Centric Design and Gamification**

Throughout development, user engagement remained a central goal. Visual and auditory aids were added to flashcards to support different learning styles. Gamification components were carefully crafted not just for entertainment, but to reinforce positive study habits through motivation and rewards.

- Streaks and achievement badges incentivize consistent learning
- Leaderboards foster friendly competition
- Adaptive difficulty challenges keep users in an optimal learning zone

- **Testing and Feedback**

Quality assurance was performed through unit testing, API integration testing, and user experience testing. Prototype testing with learners provided essential feedback, leading to improvements in flashcard flow, review timing, and progress visualization. Insights from performance data (e.g., accuracy rates, session length) were used to fine-tune the algorithm’s responsiveness.

- **Deployment and Monitoring**

  The application was deployed to a cloud environment with containerization support for scalability. Real-time monitoring tools were implemented to track system performance and user behavior metrics. These insights guide future development iterations and allow proactive identification of bottlenecks or user pain points.

2.5 Commercialization aspects of the product

BioMentor is a web-based AI-driven educational platform designed to revolutionize how Sri Lankan A/L Biology students engage with their syllabus. The platform's commercialization strategy focuses on accessibility, scalability, and long-term sustainability through a web-first model.

1. **Target Market**

   - The core user base includes Sri Lankan A/L Biology students (English medium), teachers, tuition centers, and independent learners.
   - Secondary markets include educational institutions looking for smart learning tools and students preparing for competitive exams. Future expansion can cover other subjects and local language versions (Sinhala/Tamil).

2. **Web-Based Delivery Model**

   - As a web application, BioMentor is easily accessible through any browser without requiring installation, ensuring cross-platform compatibility (desktop, laptop, tablet, mobile).
   - This model supports quick updates, centralized content management, and broader reach especially useful for students in areas with limited access to mobile apps or high-end devices.

3. **Revenue Streams**

   - **Subscription Plans**: Monthly/annual subscriptions offering access to advanced features such as adaptive quizzes, essay evaluation, analytics, and revision schedules.
   - **Institutional Packages**: Schools and tuition centers can purchase group licenses, with administrative dashboards for teachers.
   - **Content Packs**: One-time purchases for exam-focused materials, model papers, or exclusive revision sets.
   - **Advertisements (optional)**: Limited, non-intrusive ads for free-tier users to support platform maintenance.

4. **Marketing and Distribution**

   - Digital marketing via social media, educational forums, and influencer partnerships to reach A/L students.
   - SEO-optimized web presence, email campaigns, and demo sessions for schools and tuition providers.
   - Integration with educational blogs and YouTube channels for visibility.

5. **Scalability and Maintenance**

   - The web-based architecture allows easy scaling via cloud hosting, supporting a growing user base without major infrastructure overhaul.
   - Regular backend updates and content uploads are managed centrally, ensuring stability and fast feature rollouts.

6. **Unique Selling Proposition (USP)**

   - Domain-specific (A/L Biology), AI-powered personalization, automated evaluation, and syllabus-aligned content set BioMentor apart from generic e-learning platforms.
   - Seamless user experience across devices with real-time performance tracking and progress visualization tools.

7. **Future Expansion Opportunities**

   - Expansion into additional A/L science subjects and support for O/L students.
   - Multilingual support (Sinhala/Tamil) to reach a broader audience.
   - Potential white-labeling opportunities for tuition providers and institutions to offer custom-branded learning environments.

8. **Social Impact and Accessibility**

   - Affordable pricing tiers and scholarship-based access models promote educational equity.
   - Optimized performance for low-bandwidth environments ensures usability in rural and underserved communities.

2.6 TESTING & IMPLEMENTATION

2.6.1 Testing

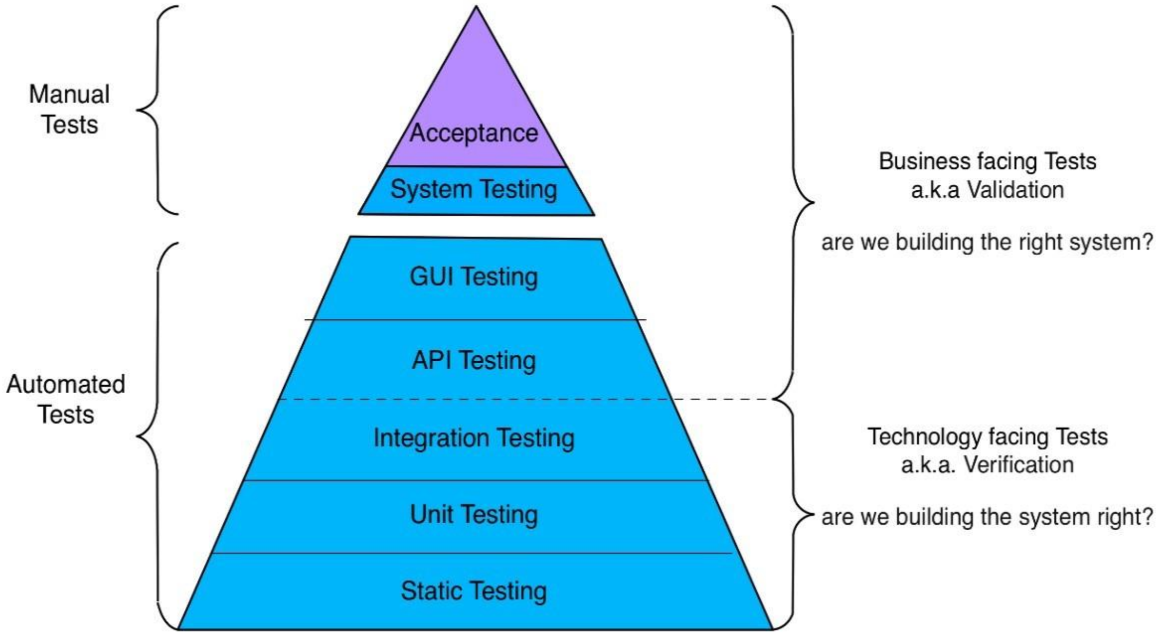

Figure 13: Test Triangle

Figure 13 illustrates the testing pyramid strategy applied during the development of the **BioMentor** web application. The testing process was conducted across all levels of the software development life cycle to ensure the delivery of a stable, high-quality, and user-friendly learning platform. Each phase of testing focused on identifying and resolving issues early, ensuring that every component of the system performed according to expectations.

1. Unit Testing

In this phase, each individual function and component of the BioMentor system such as the summarization logic, quiz generation algorithm, spaced repetition module, and answer evaluation mechanism was tested independently. The purpose was to ensure that each unit performed accurately and reliably before integration. Units that passed this stage were confirmed to be free of basic logic errors and were considered ready for modular integration.

2. Module Testing

Following unit testing, each functional module was tested in isolation to ensure internal consistency and correctness. For example, the summarization module was tested to confirm that uploaded documents were correctly processed, summarized, and displayed. The adaptive quiz module was also validated for proper flow from question selection to scoring. This phase helped ensure that grouped functionalities worked well together within each module.

3. Integration Testing

Integration testing focused on evaluating the communication between different modules. This involved checking whether components like the frontend interface, backend services, and AI logic could interact smoothly. A key example was the integration of the quiz results with the feedback system, ensuring that user performance influenced subsequent content and difficulty levels as intended.

4. System Testing

System testing was conducted after all components were fully integrated. The complete BioMentor platform was tested as a whole, simulating real user interactions such as student registration, content summarization, quiz participation, answer submission, and spaced repetition sessions. This phase ensured that the system fulfilled all specified requirements, both functionally and in terms of performance and stability.

5. User Acceptance Testing (UAT)

In the final stage, the platform was tested by actual end-users A/L Biology students and teachers. Their feedback focused on usability, content relevance, clarity of feedback, and overall learning experience. Based on this feedback, necessary refinements were made to improve user interface design, content accuracy, and adaptability of the learning path. This stage validated the platform's readiness for deployment in a real educational environment.

Maintenance

After deployment, the BioMentor platform undergoes regular maintenance to ensure stability, accuracy, and performance. This includes fixing bugs, updating content to match curriculum changes, and continuously monitoring and fine-tuning AI models for improved accuracy. User feedback is reviewed frequently to guide usability enhancements and feature upgrades. System performance is monitored to address any slowdowns or resource issues, while security updates are applied regularly to protect user data. As the user base grows, infrastructure is scaled accordingly to maintain a smooth and responsive experience.

3 RESULTS & DISCUSSION

a. LLM-Based Abstractive Text Summarization Tool with Voice Output

The BioMentor summarization tool was created to assist A/L Biology students by producing brief summaries, organized notes, and audio outputs. It was assessed based on architecture, quality of output, and educational significance.

1. Architectural Assessment

Two architectures, monolithic and microservices were deployed via Docker for comparison. The monolithic architecture excelled in speed (85% quicker response time), reduced resource consumption (~34% CPU), and simpler debugging, enhancing its suitability for direct integration. The microservices architecture provided improved scalability but brought about complexity and increased resource requirements.

2. Quality of Summarization (ROUGE Assessment)

Leveraging the fine-tuned Flan-T5 Base model, the system accomplished:

- ROUGE-1: 0.74 (effective keyword retention)
- ROUGE-2: 0.40 (strong phrase coherence)
- ROUGE-L: 0.54 (logical organization)

These scores indicate high-quality summaries aligned with the curriculum, fitting for academic purposes.

3. Educational Influence

The system effectively handled biology queries, documents, and notes instantly, delivering results in PDF and MP3 formats. Translating into Tamil and Sinhala broadened access for a larger student population. Monolithic architecture was selected for integration because of its straightforwardness and effectiveness.

In conclusion, BioMentor's summarization tool demonstrated robust technical capabilities and significant educational value, providing tailored and inclusive assistance for A/L Biology students. Figure 14 shows the postman output of /process-query API call.

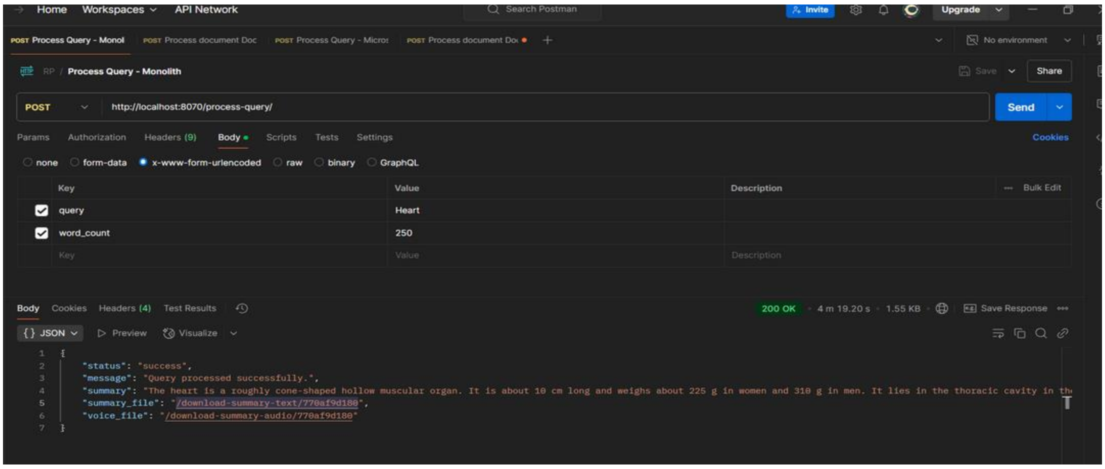

Figure 14: Response of process-query

b. LLM Based Provide answers for structured and essay type of questions and evaluate answers based on approved resources.

The developed system successfully fulfilled its goal of enabling Sri Lankan A/L Biology students to independently generate and evaluate answers for both structured and essay-type questions. The integration of a fine-tuned **LLAMA 3-Instruct** model allowed the system to generate syllabus-aligned, coherent, and contextually accurate answers for student queries. Both question types were handled within the same model pipeline using prompt-based conditioning, ensuring consistent behavior and output quality.

The **hybrid evaluation engine**, combining **SciBERT** for semantic similarity, **TF-IDF** and **Jaccard** for lexical and keyword matching, and **spaCy** and **LanguageTool** for grammatical analysis, proved effective in assessing student answers. It provided detailed scoring across multiple dimensions, including conceptual relevance, keyword usage, and language quality. Students received clear, actionable feedback, which promoted self-reflection and revision without teacher involvement.

The **question moderation module**, powered by a BERT-based classifier, reliably filtered out irrelevant or vague questions. This feature preserved the academic integrity of the system and ensured model usage remained aligned with the A/L Biology syllabus. Similarly, the **study recommendation engine**, using FAISS and SentenceTransformers, was successful in suggesting personalized textbook content based on gaps identified during answer evaluation. This feature addressed the learning needs of individual students and enhanced targeted revision.

API endpoints such as /generate-answer, /evaluate-answer, and /evaluate-passpaper-answer performed with low latency and stable results. The /get-student-question and /student-analytics routes enabled simulated exam practice and performance tracking, creating an engaging and exam-relevant learning experience.

From a system performance perspective, the backend architecture using **FastAPI** and **MongoDB**, coupled with a **React.js frontend**, ensured responsiveness and scalability. Deployment on **Microsoft Azure** with GitHub Actions-based CI/CD streamlined development and update cycles. Student testing and feedback validated the usability and educational relevance of the platform.

Overall, the results demonstrate that the system effectively supports independent student learning, automates answer evaluation with explainability, and provides curriculum-aligned academic support. The modular design allows easy future extension to other subjects and languages.

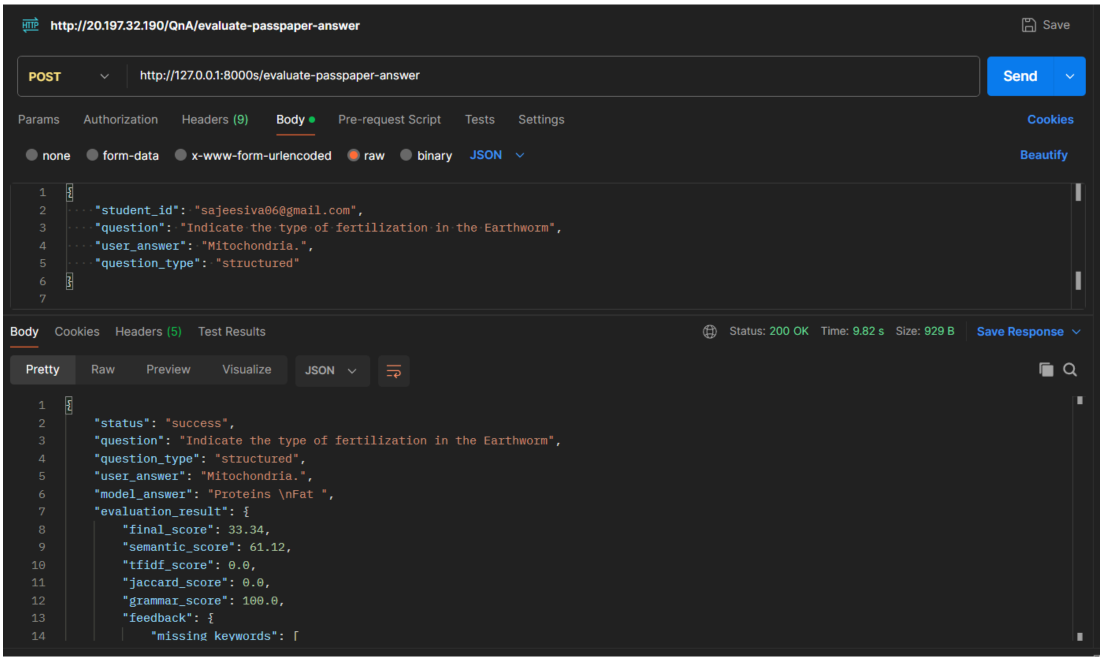

Figure 15: Sample API Testing Using Postman for Question And Answering

c. LLM-Based Adaptive Quiz Platform for A/L Biology Students with Performance-Driven MCQ Generation

The developed platform effectively supports A/L Biology students by delivering personalized, syllabus-aligned multiple-choice quizzes. Using a fine-tuned **LLaMA 2 model (llama-2-7b-chat-hf)**, the system integrates **Retrieval-Augmented Generation (RAG)** to produce diverse, contextually accurate questions. Adaptive quiz functionality is powered by **Item Response Theory (IRT)**, which adjusts difficulty levels in response to individual performance, enhancing learning engagement and efficiency.

1. Quiz Generation Performance

The system successfully generates full quizzes in real time using prompt-engineered LLM outputs enhanced by RAG. Sample MCQs are retrieved from a curated dataset to guide the model's generation, resulting in unique question sets each time. Evaluation showed that question duplication across multiple attempts remained minimal, and quiz generation time consistently stayed below 4 seconds, ensuring seamless user interaction.

2. Adaptive Difficulty and Performance Evaluation

The adaptive logic accurately adjusted quiz difficulty based on student ability using IRT scoring principles. After each attempt, the system analyzed accuracy and consistency to estimate performance and influence the difficulty distribution of future quizzes. Student testing confirmed that quiz difficulty scaled meaningfully, encouraging learners to remain challenged without being overwhelmed.

3. Educational Impact

The platform supports both **adaptive** and **unit-wise** quiz modes. Unit-wise quizzes allowed students to focus on specific chapters, while the adaptive mode offered dynamic difficulty scaling based on real-time performance. The integrated dashboard displayed key metrics such as topic-wise accuracy, leaderboard rankings, score trends, and quiz history. These features empowered students to monitor progress independently and revise purposefully based on their performance data.

4. Usability and System Feedback

Students noted that the quiz flow, difficulty variation, and performance insights contributed to a more personalized and engaging learning experience. The retry option enabled repeated attempts on adaptive quizzes, encouraging continuous improvement. User testing confirmed that the system interface was smooth and responsive, with minimal delays during quiz loading and result display.

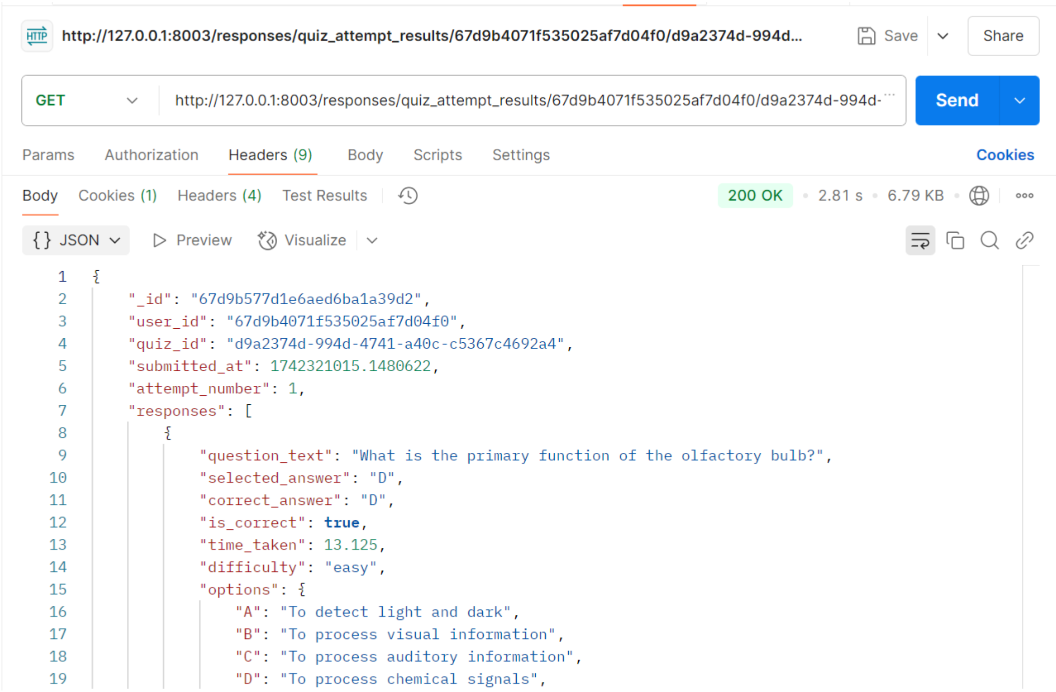

Figure 16: Sample API Testing Using Postman for Adaptive Quiz Platform

d. Enhancing Vocabulary Memorization through Adaptive Spaced Repetition

The implementation of **Adaptive Spaced Repetition (ASR)** demonstrated a significant enhancement in learning efficiency and memory retention among users. Learners who engaged with the ASR-based system showed marked **improvements in recall accuracy**, particularly for vocabulary items that were initially more difficult to remember. By leveraging the **SM-2** algorithm, the system intelligently modified review intervals based on **individual performance**, ensuring that challenging concepts were reviewed more frequently while allowing easier items to be spaced further apart. This personalization of review timing contributed to a more efficient and targeted learning process.

One of the most notable outcomes was the **improvement in long-term retention**. Users maintained high recall rates even during delayed follow-up reviews, suggesting that the adaptive spacing effectively reinforced memory consolidation. The reduction in cognitive load was also evident, as learners were not overwhelmed with unnecessary repetitions. Instead, the system's focus on optimizing effort distribution allowed them to invest their cognitive resources more effectively, which translated into better overall performance and learning outcomes.

The user experience was further enhanced by the integration of **gamification** elements. Features such as streak tracking, badges, and leaderboards helped sustain motivation and encouraged consistent study habits. Additionally, the inclusion of annotated **pronunciation** support addressed diverse learning preferences, catering to both **visual** and **auditory** learners. As a result, the platform not only improved cognitive outcomes but also created an engaging and personalized environment conducive to long-term learning.

However, the success of **ASR** was not uniform across all users and was dependent on a few key factors. The system's ability to assess item difficulty accurately relied on consistent and honest self-assessment by users, particularly in assigning quality scores after each review. In cases where user feedback was inaccurate or inconsistent, the algorithm's effectiveness diminished. Furthermore, regular interaction with the platform was crucial. Users who skipped sessions or used the system sporadically did not benefit as much as those with consistent engagement, highlighting the importance of habit formation in adaptive learning systems.

Despite these challenges, the findings underscore the potential of **ASR** as a powerful tool for personalized education. Its benefits are evident not only in improved recall but also in learner satisfaction and sustained engagement. Looking forward, the system could be enhanced by incorporating more advanced **personalization** techniques, such as machine learning-based predictions or biometric data analysis, to refine difficulty assessment and review scheduling further. Additionally, expanding the application of **ASR** to broader educational domains, including professional certification and medical training, could further validate its versatility and impact.

4 SUMMARIES OF EACH STUDENT'S CONTRIBUTION

IT Number	Name	Component	Role
IT21068478	Dharane.S	LLM-Based Abstractive Text Summarization Tool with Voice Output implemented in different software architectures	Business Analyst Developer Tester Project Manager
IT21204302	Sajeevan. S	LLM Based Provide answers for structured and essay type of questions and evaluate answers based on approved resources.	Business Analyst Developer Tester
IT21264634	Sujitha.S	LLM-Based Adaptive Quiz Platform for A/L Biology Students with Performance-Driven MCQ Generation	Business Analyst Developer Tester
IT21375132	Srirajan. G.A	Enhancing Vocabulary Memorization through Adaptive Spaced Repetition	Business Analyst Developer Tester

Table 1: Summary Of Each Student's Contribution

5 CONCLUSION & FUTURE WORK

This research successfully presented **BioMentor**, a personalized web-based e-learning platform designed to enhance the academic performance and engagement of Sri Lankan A/L Biology students. By integrating advanced artificial intelligence techniques such as transformer-based summarization, Retrieval-Augmented Generation (RAG), adaptive quiz generation, automated answer evaluation, and spaced repetition, the platform offers a comprehensive, user-centric learning experience aligned with the national curriculum.

Through modular and scalable architecture, BioMentor addresses the limitations of traditional learning methods by adapting to each student's learning pace and proficiency. Testing and user feedback demonstrated the system's effectiveness in delivering accurate, relevant, and engaging educational content. The combination of interactive features and intelligent learning algorithms positions BioMentor as a powerful tool to improve both short-term academic performance and long-term knowledge retention.

Future Work

While BioMentor has shown strong potential in its current form, several areas have been identified for future development and enhancement:

- **Subject Expansion**
  Extend the platform to support additional A/L subjects such as Chemistry, Physics, and General English, using similar AI-driven methodologies.
- **Multilingual Support**
  Implement Sinhala and Tamil language options for both content and interface to make the platform more inclusive and accessible to a wider student base.
- **Mobile Application Development**
  Develop dedicated Android and iOS mobile applications to enhance usability and accessibility for students who rely on mobile devices.
- **Real-Time Analytics and Dashboards**
  Integrate advanced analytics for teachers and students to visualize learning progress, identify weak areas, and receive personalized improvement suggestions.

- **Gamification Enhancements**

  Introduce more advanced gamification features such as quizzes with time challenges, progress-based unlockables, and competitive leaderboards to further boost engagement.
- **AI Model Optimization**

  Continuously fine-tune and retrain AI models using more localized and updated datasets to improve contextual accuracy and subject relevance.
- **Collaboration with Educational Institutions**

  Partner with schools and the Ministry of Education to officially integrate BioMentor into classrooms as a supplementary learning tool.

By pursuing these future directions, BioMentor can evolve into a holistic digital learning ecosystem that not only supports students academically but also transforms the way they learn, revise, and interact with complex subjects.

REFERENCES

[1] D. Sudharson et al., “An Abstractive Summarization and Conversation Bot Using T5 and its Variants,” ICAICCIT 2023, IEEE, pp. 431–437.

[2] K. Maurya et al., “NLP-Enhanced Long Document Summarization: A Comprehensive Approach for Information Condensation,” 2024 2nd Int. Conf. on Advancement in Computation & Computer Technologies (InCACCT), pp. 187–192.

[3] M. Ramina et al., “Topic Level Summary Generation Using BERT-Induced Abstractive Summarization Model,” Proc. ICICCS 2020, IEEE, pp. 747–752.

[4] A. Goyal et al., “TalkifyPy: The Pythonic Voice Assistant,” 2024 1st Int. Conf. on Advanced Computing and Emerging Technologies (ACET), IEEE, DOI:10.1109/ACET61898.2024.10730081.

[5] J. Christian et al., “Analyzing Microservices and Monolithic Systems: Key Factors in Architecture, Development, and Operations,” IC2IE 2023, IEEE, pp. 64–69.

[6] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *arXiv preprint arXiv:1810.04805.*

[7] Brown, T., Mann, B., Ryder, N., et al. (2020). “Language Models are Few-Shot Learners.” *NeurIPS 2020.*

[8] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv preprint arXiv:1810.04805.*

[9] Dutta, H. S., & Das, B. (2020). "Hybrid AI Models for Educational Applications." *Journal of Artificial Intelligence Research, vol. 67, pp. 231-245.*

[10] A. S. K. Shukla, D. Arora, and A. K. Sharma, "Automatic Question Answer Generation Using T5 and NLP," IEEE ICCCA, 2019.

[11] P. Kumar, N. Agarwal, and R. Nath, "Generation of Multiple-Choice Questions From Textbook Contents of School-Level Subjects," IEEE ICCCA, 2019.

[12] S. Kumar and M. Gupta, "Automatic Question Generation for Intelligent Tutoring Systems," IEEE ICCCA, 2019.

[13] A. R. Patel, P. K. Jha, and S. Roy, "MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning," IEEE ICCCA, 2019.

[14] R. Sharma and K. Singh, "Generation of Multiple-Choice Questions From Indian Educational Text," IEEE ICET, 2023.

[15] N. J. Cepeda, H. Pashler, E. Vul, J. T. Wixted, and D. Rohrer, "Distributed practice in verbal recall tasks: A review and quantitative synthesis," *Psychological Bulletin*, vol. 132, no. 3, pp. 354–380, 2006.

[16] H. L. Roediger and A. C. Butler, "The critical role of retrieval practice in long-term retention," *Trends in Cognitive Sciences*, vol. 15, no. 1, pp. 20–27, 2011.

[17] P. Pavlik and J. R. Anderson, "Using a model to compute the optimal schedule of practice," *Journal of Experimental Psychology: Applied*, vol. 14, no. 2, pp. 101–117, 2008.

[18] S. H. Kang, "Spaced repetition promotes efficient and effective learning: Policy implications for instruction," *Policy Insights from the Behavioral and Brain Sciences*, vol. 3, no. 1, pp. 12–19, 2016.

[19] J. D. Karpicke and A. Bauernschmidt, "Spaced retrieval: Absolute spacing enhances learning regardless of relative spacing," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 37, no. 5, pp. 1250–1257, 2

[20] Y. Tomikawa, A. Suzuki, and M. Uto, "Adaptive Question–Answer Generation With Difficulty Control Using Item Response Theory and Pretrained Transformer Models," IEEE Transactions on Learning Technologies