

**AI-DRIVEN ADAPTIVE LEARNER - CENTRIC
LEARNING MANAGEMENT SYSTEM**

24-25J-112

Project Final Report

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

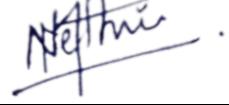
Department Of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

April 2025

DECLARATION

We declare that this is our work, and this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning, and to the best of our knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Student ID	Name	Signature
IT21251900	Rajapaksha R.M.S.D	
IT21249266	Peiris A.R.D	
IT21360428	Monali G.M.N.	
IT21302862	Sri Samadhi L.A.S.S	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are our research for the undergraduate Dissertation under my supervision.

Signature of the Supervisor

Date

2025.03.30

Dr. Kalpani Manathunga

ABSTRACT

The continuous evolution of educational technology necessitates adaptive, learner-centric systems tailored to individual student needs. This paper introduces an advanced approach to enhancing student engagement and personalized learning through the integration of four key components: Personalized Quiz Generation, Adaptive Mind Map Generation, Study Techniques, and Document Assistant. Each component addresses critical challenges in modern education, including real-time adaptability, cognitive load management, and personalized learning support. The system incorporates evidence-based study techniques, such as the Pomodoro Technique and adaptive workload balancing, to optimize focus, retention, and productivity. Leveraging cutting-edge technologies such as Large Language Models (LLMs) and microservices architecture, the proposed system ensures a seamless and scalable learning environment. This paper details the system's design, implementation, and potential educational impact, emphasizing its role in improving learning efficiency and student outcomes.

Keywords— retrieval-augmented generation, personalized quiz generation, adaptive mind maps, study techniques, cognitive load management

ACKNOWLEDGEMENT

We want to start by gratefully thanking our supervisor, Dr.Kalpani Manathunga, for her persistent support and guidance, which allowed us to successfully finish our research work. We would also like to thank Ms. Thilini Jayalath, co-supervisor on this research project, for her readiness to help whenever help was needed, in addition to our supervisor. We want to thank everyone, especially our parents and other family members who have always helped us to receive both spiritual and financial support. In addition to that, we want to thank our closest friends who have helped in one way or another to the completion of this research.

TABLE OF CONTENTS

DECLARATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT	iv
List of Tables.....	ix
List of Figures	x
List of abbreviations	xii
1 INTRODUCTION	1
1.1 General Introduction	1
1.1.1 Evidence-Based Study Techniques Integration.....	2
1.1.2 AI-Powered Document Assistant.....	2
1.1.3 Personalized Quiz Generation Module	3
1.1.4 Adaptive Mind Map Generation Module.....	3
1.2 Background Literature	5
1.3 Research Gap	7
1.3.1 Study Techniques and Cognitive Load Management.....	7
1.3.2 AI document assistant	9
1.3.3 Quiz generations	12
1.3.4 AI- driven Adaptive Mind Map Generation Component	15
1.4 Research Problem	17
2 RESEARCH OBJECTIVES	20
2.1 Main Objective.....	20
2.2 Sub Objectives	20
3 METHODOLOGY	24
3.1 Requirement Gathering	25
3.2 Past Research Analysis	26
3.3 Refer to Official Documentation.....	27
3.4 Identify Existing Methodologies.....	28
3.5 Stakeholder Engagement	30
3.6 Project Requirements	31
3.6.1 Functional Requirements	31

3.6.2	Non-Functional Requirements	32
3.6.3	User Requirements	33
3.6.4	System Requirements.....	34
3.7	Feasibility Study	35
3.7.1	Technical Feasibility	35
3.7.2	Hardware Requirements.....	36
3.7.3	Knowledge of Deployment and Integration.....	36
3.7.4	Bloom's Taxonomy Integration for Quiz Personalization	37
3.7.5	Resource Integration using Web Search API s.....	37
3.7.6	Retrieval-Augmented Generation (RAG) and LLM Integration	37
3.7.7	Knowledge of applying Prompt Engineering Techniques	38
3.7.8	Economic Feasibility	38
3.7.9	Schedule Feasibility	39
3.8	Overall System Diagram.....	40
3.8.1	Study Techniques and Cognitive Load Management.....	43
3.8.2	AI document assistant	45
3.8.3	Quiz generations	46
3.8.4	AI-driven Adaptive Mind Map Generation.....	47
3.9	Development Process	49
3.10	Project Breakdown.....	51
3.10.1	Gantt Chart.....	51
3.10.2	Work Breakdown Structure.....	52
3.10.3	Click up work breakdown schedule	53
3.11	Design Diagrams for the Components	54
3.11.1	Use Case Diagrams	54
3.11.2	Sequence Diagrams.....	57
3.11.3	Component Diagrams	61
3.11.4	Key Pillars of the Research Domain	64
3.12	Implementation	69
3.12.1	Main Application UI and Flow	69
3.12.2	Adaptive Study Session Manager Implementation.....	71

3.12.3	AI Document Assistant Implementation.....	77
3.12.4	Quiz Generation Component	82
3.12.5	Adaptive Mind Map Component	89
3.13	Testing.....	103
3.13.1	Test Plan.....	103
3.13.2	Test Strategy.....	104
3.13.3	Test Case Design.....	105
3.14	Commercialization.....	118
3.14.1	Market Research and Analysis.....	118
3.14.2	Product Development.....	119
3.14.3	Monetization Strategy	119
3.14.4	Marketing and Promotion	119
3.14.5	Regulatory Compliance	120
3.14.6	Continuous Innovation and Iteration	120
4	RESULTS & DISCUSSIONS.....	121
4.1	Results of Study Technique integration and session management.....	121
4.1.1	Results of Intelligent Break Scheduling Algorithm	123
4.2	AI Document Assistant	126
4.2.1	General Query – What is this document about?	126
4.2.2	Summarisation - Summarize the main points about SOAP and how it works	127
4.2.3	Table Comparison – Provide a table comparing different authentication methods (OpenID, OAuth, SSL).....	128
4.2.4	Quiz Generation - Make a short quiz about the differences between WSDL and SOAP.....	129
4.2.5	Code Snippets - Give a sample HTTP request for a RESTful web service.	
	130	
4.2.6	Simplification - Simplify the concept of WSDL bindings.....	131
4.2.7	Explanation - Explain the role of UDDI in web services?.....	132
4.2.8	Unrelated Queries - Who won the FIFA World Cup in 2018?	133
4.3	Quiz Generation Component	135
4.3.1	Quiz Generation Output.....	135

4.3.2	Semantic Answer Evaluation	136
4.3.3	Adaptive Personalization and Feedback	138
4.4	Adaptive Mind Map Generation Component	140
4.4.1	Results of Prompt Engineering for Mind Map Generation.....	143
5	SUMMARY	161
5.1	Rajapaksha R.M.S.D – IT21251900	161
5.2	Peiris A.R.D - IT21249266	161
5.3	Monali G.M.N – IT21360428.....	162
5.4	Sri Samadhi L.A.S.S – IT21302862	162
6	CONCLUSION	163
	References	164
	Appendix	168

LIST OF TABLES

Table 1 Research Gap - Study Techniques Component	8
Table 2 Research Gap - AI-document assistant Component	11
Table 3 Research Gap - Quiz Component.....	12
Table 4 Research Gap - Mind-Map Component	15
Table 5 Test case for verifying the register functionality of the system	105
Table 6 Test case for verifying session creation functionality	106
Table 7 Test case for verifying session creation functionality	107
Table 8 Test case for verify session timeline display and break calculation.....	107
Table 9 Test case for verifying session related items are hidden when vie.....	108
Table 10 Test case for verifying session upcoming b	109
Table 11 Test case for verifying current work period extend feature is working.....	109
Table 12 Test case for Automated Continuous Deployment.....	111
Table 13 Test case for Out of Scope Questioning	111
Table 14 Test case for Harmful questioning	112
Table 15 Test case for verifying multiple-choice question generation	112
Table 16 Test case for verifying short-answer question generation	112
Table 17 Test case for verifying adaptive difficulty scaling based on user performance.	113
Table 18 Test case for verifying progress tracking on user dashboard.	113
Table 19 Test case for verifying feedback generation based on user quiz performance	114
Table 20 Test case for verify the PDF content extraction and structuring process	114
Table 21 Test case for verify the AI-driven mind map generation from structured content	115
Table 22 Test case for verify user prompt-based mind map generation	115
Table 23 Test case for verifying the user prompt-based mind map generation for invalid contexts	116
Table 24 Test case for verifying the replacement of the earliest mind map when exceeding the storage limit.	117
Table 25 Dynamic scheduling adaptation across a session.....	126
Table 26 Response Evaluation 1	126
Table 27 Response Evaluation 2	127
Table 28 Response Evaluation 3	128
Table 29 Response Evaluation 4	129
Table 30 Response Evaluation 5	130
Table 31 Response Evaluation 6	131
Table 32 Response Evaluation 7	132
Table 33 Response Evaluation 8	133
Table 34 Model Comparison.....	140
Table 35 Prompting Techniques	143

LIST OF FIGURES

Figure 1 System Architecture Diagram.....	40
Figure 2 System Flow Diagram	42
Figure 3 Agile Methodology	49
Figure 4 Gantt Chart	51
Figure 5 Work Breakdown Structure	52
Figure 6 Click up work breakdown schedule	53
Figure 7 Use Case Diagram - Quiz Component	54
Figure 8 Use Case Diagram - Mind-Map Component.....	55
Figure 9 Use Case Diagram - Prompt Minds Component	56
Figure 10 Sequence Diagram - User Login	57
Figure 11 Sequence Diagram - Document Processing in AI Document Assistant	58
Figure 12 Sequence Diagram - Quiz Component	59
Figure 13 Sequence Diagram - Mind-Map Component	60
Figure 14 Component Diagrams - Study Techniques Component	61
Figure 15 Component Diagrams - AI Document Assistant Component.....	62
Figure 16 Component Diagrams - Quiz Component	63
Figure 17 Component Diagrams - Mind-Map Component.....	63
Figure 18 LLM Frameworks.....	65
Figure 19 Prompt Engineering Techniques.....	66
Figure 20 Main UI 01	69
Figure 21 Main UI 02	69
Figure 22 Main UI 03	70
Figure 23 Main UI 04	70
Figure 24 Main UI 05	71
Figure 25 Dynamic Session Break Scheduler.....	73
Figure 26 Session Timeline.....	75
Figure 27 Session Break Popup	75
Figure 28 AstraDB Implementation.....	78
Figure 29 OpenAI Implementation.....	80
Figure 30 Model Selection.....	82
Figure 31 Bloom's Taxonomy Integration	83
Figure 32 Feedback Generation	84
Figure 33 Feedback Display	85
Figure 34 MCQ Prompt	86
Figure 35 MCQ UI.....	87
Figure 36 SAQ Prompt	87
Figure 37 SAQ UI.....	88
Figure 38 Difficulty Level Adjustments	89
Figure 39 Adaptive Mind-Map Generation Component Implementation Sequence	90

Figure 40 Model Selection for Mind-Map	91
Figure 41 Prompt – PDF Extraction	92
Figure 42 Prompt - Mind-Map Genaration	93
Figure 43 Prompt - Quiz based Mind-Map Generation	94
Figure 44 Prompt – Mind-Map Question Validation Generation	95
Figure 45 Prompt – Simple Mind-Map Generation	96
Figure 46 Prompt – Resource handling	97
Figure 47 Google Search API Integration.....	98
Figure 48 LLM Integration for Resource Generation.....	99
Figure 49 UI - Mind-Map	100
Figure 50 UI - Quiz based Mind-Maps	100
Figure 51 UI - Mind-Map Resourse Linking	101
Figure 52 UI - Prompt Mind-Maps	101
Figure 53 UI - Note Cards	102
Figure 54 UI -Chapter view	102
Figure 55 miStudy Logo	118
Figure 56 Study Technique integration and session management 01	121
Figure 57 Study Technique integration and session management 02	122
Figure 58 Intelligent Break Scheduling	124
Figure 59 Generated MCQ Interface	135
Figure 60 Generated SAQ Interface	136
Figure 61 Feedback Displayed After Answer Submission	137
Figure 62 Semantic Search Result (Weaviate Match)	137
Figure 63 Difficulty Level Adjustment	138
Figure 64 Quiz Attempt History and Performance Graph	139
Figure 65 Model Performance with Llama 3.1 8B	141
Figure 66 Model Performance without Llama 3.1 8B	142
Figure 67 Main Mind-Map Generation Output Format.....	147
Figure 68 Main Mind-Map Genaration Output	147
Figure 69 Quiz based Mind-Map Genaration Output Format	151
Figure 70 Quiz based Mind-Map Genaration Output	152
Figure 71 Simple Mind-Map Generator Output Format.....	159
Figure 72 Simple Mind-Map Generator Output	159
Figure 74 Plagiarism Report	168

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
LMS	Learning Management System
RAG	Retrieval Argument Generation
DMR	Detailed Meaning Representation
MCQ	Multiple Choice Question
SAQ	Short Answer Question
PDF	Portable Document Format
LLM	Large Language Model
UI	User Interface
UX	User Experience
DB	Database
CI	Continuous Integration
CD	Continuous Development
GPT	Generative Pre-trained Transformer

1 INTRODUCTION

1.1 General Introduction

In the evolving landscape of education, technology continues to redefine how knowledge is delivered, consumed, and retained. The growing shift toward digital and remote learning environments has amplified the demand for intelligent, adaptive platforms that transcend traditional content delivery. While conventional LMSs have become standard in educational institutions, they often follow rigid, static models of instruction that fail to accommodate the diverse cognitive styles, engagement patterns, and learning paces of individual students. This one-size-fits-all approach frequently leads to cognitive overload, reduced learner motivation, and ineffective knowledge retention.

To overcome these limitations, this research proposes the development of an AI-powered Adaptive LMS, an intelligent, modular educational platform designed to personalize the learning journey for each user. By leveraging advanced technologies such as LLMs, RAG, and real-time cognitive monitoring, the system dynamically adapts learning content and strategies based on individual student behavior, performance, and preferences. The core objective is to foster meaningful understanding, boost academic achievement, and enhance learner engagement through personalization and cognitive support.

Moreover, to support a holistic and user-friendly learning environment, the system integrates notetaking and chapter-view tools, allowing students to annotate key points and revisit specific content areas efficiently. These features cater to learners who prefer reviewing and organizing knowledge manually, reinforcing long-term retention and enabling self-paced study. All components are seamlessly unified within a single intelligent platform, providing an all-in-one solution for personalized academic success.

1.1.1 Evidence-Based Study Techniques Integration

This module introduces a Cognitive Load Management mechanism that intelligently adapts study sessions based on user interaction patterns such as mouse movement, window focus, and engagement metrics. Using a Dynamic Work-Break Adjustment Algorithm, the system optimizes study intervals and rest periods in real time, promoting sustained concentration and preventing mental fatigue. Additionally, it incorporates ambient background music to reduce distractions and maintain mental clarity.

A core aspect of this module is the integration of the Pomodoro technique, which has been shown to enhance productivity and information retention by alternating focused work sessions with regular breaks [1]. Together, these strategies align with principles from Cognitive Load Theory to ensure that learners remain cognitively balanced and mentally resilient during study sessions, thereby promoting deep learning and long-term retention [2].

1.1.2 AI-Powered Document Assistant

To combat inefficiencies associated with navigating extensive academic texts, the system includes an intelligent, AI-powered document assistant that employs LLMs and semantic retrieval techniques. Students can upload PDFs and interactively query content using natural language. The assistant returns context-aware, filtered responses, eliminating irrelevant sections and helping students quickly extract meaningful insights.

This component promotes self-directed learning by simplifying the process of understanding complex material, allowing students to focus on high-value content without spending hours scanning dense text. It not only reduces study time and frustration but also fosters independent academic exploration, empowering learners to engage deeply with subject matter at their own pace [3].

1.1.3 Personalized Quiz Generation Module

Assessment within this LMS is reimagined through a Personalized Quiz Generation Module, which utilizes learner history, performance analytics, and content mapping to generate intelligent quizzes tailored to the user's current comprehension level. With the aid of LLMs and adaptive difficulty scaling, each quiz is constructed to reinforce weak knowledge areas while avoiding repetitive coverage of topics already mastered [4] [5].

This method supports mastery-based learning, where students progressively build competence through personalized feedback and targeted assessments [6]. The system leverages Bloom's Taxonomy to align questions with cognitive skill levels, ranging from basic recall (Remember) [7] to higher-order thinking skills like Analyze and Evaluate. Based on the learner's performance, the difficulty level is adjusted dynamically to promote deeper understanding and skill progression [4].

By receiving real-time insights into their academic strengths and challenges, learners are guided through a cycle of continuous improvement, fostering confidence and motivation throughout their educational journey.

1.1.4 Adaptive Mind Map Generation Module

Adaptive mind map generator, which serves as a visual abstraction tool for organizing and reviewing complex information. Unlike conventional mind mapping tools, this system integrates RAG, prompt engineering, and visualization to create personalized, dynamic mind maps based on the student's quiz results, document analysis, and learning preferences. These mind maps are updated continuously and help students to not only understand the hierarchical relationships between concepts but also to identify the areas they need to revisit. Research has shown that visual learning methods significantly enhance comprehension and retention, particularly when the visuals are aligned with a learner's specific knowledge gaps [8] [9]. These mind maps evolve in real time, helping

learners visualize hierarchical relationships between topics, pinpoint knowledge gaps, and focus on areas requiring further review. This module enhances both comprehension and memory retention, as research confirms that visual learning significantly supports deeper understanding, especially when aligned with personalized educational insights [8] [10].

The proposed AI-driven Adaptive LMS addresses the core limitations of traditional educational systems by integrating personalized content delivery, cognitive support mechanisms, intelligent assessments, and interactive visual learning tools. Each module ranging from adaptive study techniques and semantic document exploration to mastery-based quizzes and evolving mind maps contributes to a responsive, learner-centered digital ecosystem [10]. Complemented by integrated notetaking and chapter-view functionalities, the system provides a unified, flexible, and scalable solution tailored for modern education.

In a post-pandemic context where remote access, personalization, and mental well-being are critical, this platform represents a transformative advancement in digital learning infrastructure. By aligning with contemporary learning science and harnessing the potential of artificial intelligence, the system aspires to revolutionize educational engagement, enhance academic outcomes, and cultivate sustainable, lifelong learning habits [10] [11].

1.2 Background Literature

Modern educational technologies are increasingly turning toward cognitive and behavioral research to design tools that support deeper learning, improved attention, and sustainable study habits. One such method that has garnered widespread recognition for its simplicity and effectiveness is the Pomodoro Technique, introduced by Francesco Cirillo in the late 1980s [1]. This technique breaks down study time into focused intervals of approximately 25 minutes, followed by short breaks, to combat fatigue and enhance concentration. Costales et al. (2021), highlights the effectiveness of Pomodoro technique in enhancing student productivity and attention [12]. Their study found that when integrated into online learning environments, Pomodoro cycles significantly improved students' ability to maintain focus, reduced procrastination, and resulted in a more efficient learning experience. These findings demonstrate measurable gains in student engagement and cognitive performance when Pomodoro-based study cycles are implemented. Our proposed LMS integrates this technique by allowing students to plan study sessions around these intervals, adapting break lengths and work durations based on real-time productivity metrics. This dynamic approach ensures a personalized and health-conscious study environment, helping learners maintain mental endurance throughout prolonged study periods.

Alongside effective time management, quick and accurate access to academic content is essential for meaningful learning. In this context, AI-based document assistants have emerged as powerful tools that streamline content retrieval from complex educational material. McNamara et al. (2013) explore the application of Natural Language Processing (NLP) in educational systems, specifically focusing on intelligent tutoring systems designed to assist students with writing strategies [13]. Their work highlights the effectiveness of NLP in enabling automatic feedback generation and providing personalized assistance. In educational contexts, these NLP-based systems can parse large volumes of textual data and offer actionable insights that help students understand their work better. Building on these findings, our proposed system incorporates Large Language

Models (LLMs) and context-aware embeddings to provide a sophisticated document assistant. Students can query their uploaded materials, such as lecture notes, textbooks, or research papers, and receive precise, contextually relevant responses that directly reference the content. This intelligent assistant not only enhances comprehension but also reduces the time spent manually searching for answers, particularly when handling lengthy academic documents.

Personalization in learning has significantly advanced with the emergence of AI-powered quiz generation systems. These systems dynamically create and adjust assessments based on learners' performance, knowledge gaps, and behavioral patterns. By leveraging fine-tuned Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG), modern learning platforms can analyze past quiz results, engagement data, and identified weak areas to generate personalized quizzes. For instance, Kang (2024) introduced a quiz generation feature in the Alby Learning Management System that adapts to individual learners by adjusting question difficulty and topic relevance in real-time [14]. Similarly, the ScootPad platform, as described by Learning Explorer (2023), utilizes mastery-based adaptive learning algorithms to personalize content delivery based on student progress and response patterns [5]. Our system follows a similar approach by offering immediate feedback after each quiz, along with targeted recommendations for reinforcement. This continuous learning loop supports individualized educational experiences, enabling the system to evolve alongside the learner's growth.

One of the most cognitively beneficial techniques for knowledge retention and organization is mind mapping. This method supports visual learning by breaking down complex topics into simpler, connected nodes that reflect relationships between concepts. In the early work of Kudelić et al. (2012), the authors developed a Java-based system that auto-generates maps using semantic and syntactic structures from user input [15]. While this laid the groundwork for automated visual tools, their model lacked adaptiveness and could not evolve based on individual learner inputs or external data sources. More recently, Elhoseiny and Elgammal (2016) in their study proposed a neural method for

converting natural language text into hierarchical visual mind maps. Their approach utilized semantic abstraction layers to build nested concept structures, offering a more refined and meaningful visual learning experience [16]. Our LMS builds on these ideas by integrating adaptive mind map generation that combines PDF content extraction, student performance insights, and external resource linking through Google Search API. Using tools like Mermaid.js and prompt engineering, students receive visual mind maps that not only reflect the main material but also evolve with each quiz attempt and knowledge update. Furthermore, the system maintains up to four versions of a mind map per student, one from the core document and three adaptive maps based on quiz progression replacing the oldest when a new one is generated.

Taken together, these advancements in cognitive modeling, AI document assistance, personalized assessments, and intelligent visualization underscore a growing shift toward learner-centric, adaptable educational platforms. Our proposed system synthesizes these approaches into a unified architecture designed to enhance engagement, retention, and personalization. By embedding AI at each layer from time management and information access to assessment and conceptual understanding we aim to transform passive content delivery into a dynamic, responsive learning journey tailored to each student's academic and cognitive profile.

1.3 Research Gap

1.3.1 Study Techniques and Cognitive Load Management

The current state of the art in LMS design focuses on content delivery, with limited emphasis on personalized learning experiences and the integration of study techniques. Previous attempts to address the shortcomings of LMS platforms have primarily involved third-party applications or plugins, which fail to provide a seamless and cohesive learning experience. The proposed approach builds on this by embedding study techniques directly

within the LMS interface, offering a more integrated and personalized learning environment. As part of understanding the current landscape of study techniques and their integration into educational tools, a comparative analysis of previous research was conducted. This analysis, illustrated in table below, highlights various studies that have explored the effectiveness of different study techniques such as the Pomodoro method, Flowtime, the Feynman technique, Spaced Repetition, and the Mozart effect. Each of these techniques has been studied in isolation or in combination with others, aiming to improve learning efficiency, retention, and student engagement.

Table 1 Research Gap - Study Techniques Component

	Technique Used	Proven	Personalized	Tested Online	Used in LMS
Research 01 [2]	Pomodoro + Forest App	✓	✗	✓	✓
Research 02 [9]	Pomodoro	✓	✗	✓	✗
Research 03 [10]	Pomodoro	✓	✗	✓	✗
Research 04 [10]	Spaced Repetition	✓	✗	✗	✗
Research 05 [10]	Feynman	✓	✗	✗	✗
Research 06 [10]	Background Music	✓	✗	✓	✗
Article 01 [10]	Flowtime	✓	✓	✓	✗
Video 01 [10]	Flowtime + Pomodoro	✗	✓	✗	✗
Our Solution	Adaptive break / work algorithm	-	✓	✓	✓

Research 01	Online Learning Self-Efficacy as Correlates to Academic Procrastination among Pre-Service Teachers
Research 02	Anti-procrastination Online Tool for Graduate Students Based on the Pomodoro Technique
Research 03	Time management between the personalization and collectivization of productivity: The case of adopting the Pomodoro time-management tool in a four-day workweek company
Research 04	Spaced Repetition Promotes Efficient and Effective Learning
Research 05	Feynman Technique as a Hauntological Learning Strategy for Independent and Remote Learning
Research 06	Background Music and Cognitive Performance. Perceptual and Motor Skills
Article	How to Use the Flowtime Technique to Get More Work Done
Video	Flowtime Technique Explained

This presents a comprehensive comparison of these studies, showing the methodologies used, the contexts in which they were applied, and the outcomes measured. The diagram also underscores the gaps in existing research, particularly in how these techniques have yet to be fully integrated into Learning Management Systems (LMS) in a cohesive manner. This lack of integration points to the need for a more unified approach, where these proven study techniques are embedded directly within an LMS to provide a personalized and adaptive learning environment. This project aims to bridge that gap by not only incorporating these techniques but also enhancing them through a customizable and user-friendly interface, as outlined in the project proposal.

1.3.2 AI document assistant

The integration of Retrieval-Augmented Generation (RAG) and prompt engineering has shown significant promise in enhancing the capabilities of AI systems for knowledge-intensive and context-specific tasks, including educational applications. However, despite advancements in these areas, several critical gaps remain unaddressed in the literature, particularly in the context of self-directed learning chatbots tailored to diverse student

queries. These gaps present opportunities for innovation, which this research seeks to explore through a novel AI chatbot design that combines intent categorization with adaptive prompt selection and RAG-based retrieval.

First, while prompt engineering has evolved from manual crafting [17] to systematic optimization techniques like prompt tuning [18] and chain-of-thought prompting [19], its application in educational chatbots remains underexplored. Existing studies focus primarily on general-purpose or domain-specific question-answering [20], with little attention to dynamically adapting prompts based on the user’s intent. For instance, a student requesting a code snippet requires a different response structure (e.g., syntax-focused, executable examples) than one seeking a summary or comparison, yet the literature lacks frameworks for intent-driven prompt selection. This gap limits the ability of chatbots to provide tailored, task-specific support, a critical need in self-directed learning where students pose varied queries.

Second, RAG has demonstrated success in grounding responses in external knowledge sources, improving factual accuracy over standalone LLMs. However, its application to educational contexts often assumes a static knowledge base (e.g., Wikipedia or pre-indexed corpora) rather than dynamic, user-provided documents like student-uploaded notes or textbooks. While Petroni et al. (2021) [21] applied RAG techniques to scientific QA, their focus was on predefined datasets, not real-time integration of personalized materials. Furthermore, RAG research rarely addresses how retrieved content should be processed differently based on query type (e.g., summarizing retrieved text vs. extracting code snippets), leaving a gap in adapting retrieval outputs to diverse educational intents.

Third, the combination of RAG and prompt engineering has been studied in isolation but rarely as an integrated system for educational support. For example, Gao et al. (2023) [22] enhanced RAG with real-time retrieval, yet these efforts do not converge on a unified approach that leverages both to interpret and respond to multifaceted student needs. The lack of intent categorization in such systems means that responses may not align with the student’s specific goal whether it’s understanding a concept, generating a quiz, or

comparing ideas potentially reducing effectiveness in self-directed learning environments where immediate, relevant feedback is crucial.

Finally, existing educational AI tools, such as intelligent tutoring systems often target predefined curricula or domains, neglecting the flexibility required for self-directed learners who engage with diverse, unstructured materials. Traditional learning strategies [23] highlight persistent comprehension gaps among students, yet current chatbot solutions fail to bridge these through personalized, intent-aware, and document-grounded responses. The absence of a framework that categorizes student intent, selects task-specific prompts, and retrieves relevant content from user-provided documents represents a significant research gap one that this project aims to address.

This study fills these gaps by proposing an AI chatbot that: (1) categorizes student intent (e.g., code snippets, quizzes, summaries, comparisons) within the prompt engineering process, (2) dynamically selects predefined prompts tailored to each intent, and (3) employs RAG to retrieve and generate responses grounded in student-uploaded documents. By integrating these elements, the proposed system offers a novel solution to enhance the precision, relevance, and adaptability of AI support in self-directed learning, addressing limitations in prior work and paving the way for more effective educational tools.

Table 2 Research Gap - AI-document assistant Component

Aspect	Prompt Engineering [17] [18] [19]	RAG [22] [21] [24]	Traditional Education & Educational AI [25] [23] [26] [20].	Proposed System
Focus on Educational Chatbots	✗	✗	✓	✓

Dynamic Prompt Adaptation to Intent	✗	✗	✗	✓
Support for Varied Query Types	✗	✗	✗	✓
Integration of RAG and Prompt Engineering	✗	✗	✗	✓
Flexibility for Self-Directed Learning	✗	✗	✗	✓

1.3.3 Quiz generations

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 3 Research Gap - Quiz Component

Research	Uses Bloom's Taxonomy	Adaptive Difficulty	Real-time Quiz Generation	Personalized Feedback	LLM/AI-Based	Learner Progress Dashboard

Krousko et al. (2018) [4]	✓	✓	✗	✗	✗	✗
ScootPad (2023) [5]	✗	✓	✗	✓	✗	✓
Lubbe et al. (2025) [6]	✓	✓	✗	✓	✓	✗
Proposed System	✓	✓	✓	✓	✓	✓

Krousko et al. (2018) [4] 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) [5], 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.

1.3.4 AI- driven Adaptive Mind Map Generation Component

There have been several studies related to the automatic generation of mind maps, their personalization, and real-time adaptability. However, there remain critical gaps in the research, specifically in areas such as dynamic generation, personalization, integration of multimedia sources, and real-world applicability. The following table summarizes the comparison of relevant studies and their contributions to the identified gaps.

Table 4 Research Gap - Mind-Map Component

Research	Dynamic Mind Map Generation	Personalization of Mind Maps	Multimedia Integration	Real-World Application	Scalability
Research A [16]	✓	✗	✗	✗	✗
Research B [27]	✗	✓	✗	✗	✗
Research C [28]	✗	✗	✓	✗	✗
Proposed System	✓	✓	✓	✓	✓

Research A, conducted by Elhoseiny and Elgammal (2016), explores a novel method for hierarchical visual abstraction of natural language text, transforming text into multi-level mind maps [16]. The study presents an approach that visualizes text documents across multiple levels of abstraction, with higher-level mind map nodes representing abstract information and child nodes capturing more detailed concepts. The method uses Detailed Meaning Representation (DMR) to facilitate the transformation of text into mind map visualizations. However, this approach focuses primarily on static mind map generation, which lacks real-time dynamic adaptability based on evolving content or learner

performance. Additionally, the method does not account for individual learner preferences or incorporate multimedia elements such as images or videos. Although this research offers a foundational approach to automated mind map generation, it does not fully address the need for personalized, adaptive, and multimedia-enriched mind maps in dynamic learning environments.

Compared to the proposed system, this research focuses on static mind map generation, lacking real-time adaptability, personalization, and multimedia integration. In contrast, the proposed system utilizes RAG and LLMs to dynamically generate personalized mind maps based on learners' quiz results and preferences. Furthermore, the proposed system integrates multimedia content, enriching the mind maps and providing a more engaging and interactive learning experience.

Research B, conducted by Shi et al. (2023), investigates the effectiveness of mind mapping-based instruction on students' cognitive learning outcomes through a meta-analysis of 21 studies [27]. The study concludes that mind mapping-based instruction has a more positive influence on students' cognitive learning outcomes compared to traditional methods. It also identifies that lower-grade students are more receptive to mind mapping-based instruction, particularly in subjects like Science, Technology, Engineering, and Math (STEM). However, while the study highlights the advantages of mind mapping in enhancing learning outcomes, it does not address the limitations of traditional mind mapping methods, such as the lack of dynamic adaptation to individual learning needs. Moreover, the study focuses primarily on instructional effectiveness and does not explore how mind maps can be personalized based on real-time learner data, such as quiz results or cognitive progress.

Compared to the proposed system, Research B emphasizes the effectiveness of mind mapping in educational contexts but fails to incorporate real-time adaptive elements that respond to individual learner progress or multimedia integration. In contrast, the proposed system utilizes advanced techniques like LLMs and RAG to dynamically adjust mind maps based on the learner's performance and needs. Additionally, it integrates multimedia

elements such as images and videos, which enrich the learning experience by making the mind maps more interactive and engaging.

Research C, authored by Okada and Connolly (2008), explores the use of knowledge maps in the development of OERs to enhance meaningful learning [28]. The study focuses on the Open Learn project, which uses the Compendium software tool for visual thinking to connect ideas, concepts, and resources. The research emphasizes how knowledge mapping can help condense large volumes of information, visualize connections between learning materials, and facilitate new insights. However, while the study highlights the benefits of visualizing knowledge using mind maps, it primarily addresses static representations of learning materials and does not explore dynamic or personalized mind map generation. Furthermore, the study does not incorporate multimedia sources as input for mind map creation or address the challenges of implementing such systems at scale in large educational environments.

Compared to the proposed system, Research C contributes to the integration of knowledge mapping for OERs but lacks dynamic adaptation based on learner data and the ability to process multimodal input, such as images and videos. The proposed system, on the other hand, incorporates multimodal data processing, allowing for the generation of adaptive mind maps based on both textual and multimedia sources. Additionally, the system is designed for real-world implementation, offering scalability through cloud-based solutions, ensuring efficient performance across large learner populations.

1.4 Research Problem

The primary challenge in today's digital education landscape is the lack of widespread adoption and integration of adaptive learning systems that can respond effectively to individual learner needs. Most existing Learning Management Systems (LMS) operate with a static, one-size-fits-all approach that fails to account for the dynamic and personalized nature of student learning processes. As a result, these platforms often contribute to cognitive overload, disengagement, and limited retention, particularly for students with diverse learning styles and paces. The absence of features such as

personalized content delivery, real-time progress tracking, and intelligent feedback mechanisms hinders students from achieving their full academic potential. Furthermore, while emerging technologies like AI and large language models (LLMs) have demonstrated significant potential in revolutionizing learning systems, they remain largely underutilized in conventional LMS platforms.

In particular, current LMS tools do not efficiently support adaptive functionalities such as real-time quiz generation, context-aware resource linking, or personalized visual learning tools like mind maps. They also overlook proven productivity strategies like Pomodoro-based time management, which can boost focus and reduce burnout. The lack of integration between these intelligent tools results in a fragmented learning experience that fails to cater to the individual's cognitive and emotional needs during the learning journey. Addressing these limitations is vital to improving student engagement, knowledge retention, and overall learning efficiency.

This system aims to develop an AI-driven Adaptive Learner-Centric LMS that holistically addresses these challenges through the seamless integration of intelligent components. The system will feature an Adaptive Mind Map Generation module that visualizes key concepts and knowledge hierarchies based on students' quiz performance and content interaction. It will also include a Personalized Quiz Generator capable of adapting assessments based on learner progress and weaknesses, a Context-Aware Document Assistant that enables students to retrieve specific content from uploaded PDFs using natural language queries, and a Pomodoro-Based Study Assistant that adjusts study intervals and techniques based on user productivity and engagement. These tools will be connected through a unified architecture leveraging LLMs, Retrieval-Augmented Generation (RAG), and prompt engineering, ensuring that content and feedback dynamically evolve with the student's learning behavior.

The problem this research seeks to solve is how to design and integrate a cohesive set of AI-powered tools into a modern LMS environment to personalize and adapt the learning

experience effectively. The study will investigate subproblems such as the development of adaptive algorithms for content generation, implementation of mind maps that dynamically evolve based on ongoing learning outcomes, and the real-time synchronization of productivity-enhancing methods like Pomodoro timers with the student's current focus and cognitive load. Additionally, the system will explore how to link relevant educational resources automatically using semantic understanding and learner context.

By solving these challenges, the research aims to revolutionize the traditional LMS framework into a smart, adaptive, and learner-focused platform that enhances student outcomes and engagement. The final product will not only support the academic success of students but will also contribute significantly to the broader educational technology landscape by setting a new standard for AI-integrated learning systems.

2 RESEARCH OBJECTIVES

2.1 Main Objective

The purpose of the system is to develop an AI-driven, adaptive, and learner-centric Learning Management System (LMS) that enhances student engagement, comprehension, and academic performance by offering an integrated platform powered by intelligent educational technologies. The system aims to incorporate personalized study techniques such as adaptive mind map generation, context-aware resource linking, and quiz-based content adaptation, all tailored to suit individual learning behaviors and progress. Usability and accessibility are central to the system's design, ensuring that students can intuitively interact with the platform and benefit from its features without cognitive overload. By utilizing advanced technologies including Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Azure Custom Vision, and prompt engineering methods, this research seeks to deliver a seamless and personalized learning environment that supports effective knowledge acquisition, retention, and learner motivation.

2.2 Sub Objectives

1. To identify the most effective study techniques that enhance focus, retention, and academic performance.

This involves conducting a literature review and comparative analysis of techniques such as the Pomodoro Technique, Spaced Repetition, Flowtime, Feynman Technique, and the Mozart Effect to evaluate their impact on learning efficiency.

2. To determine which of these techniques are suitable for implementation in an online learning environment.

The selected techniques must be adaptable to digital workflows and align with students' behavioral patterns during self-paced learning.

3. To develop an adaptive system for delivering personalized study experiences within the LMS.

This includes customizing session timers, repetition schedules, and cognitive aids based on user behavior and preferences using real-time tracking and reinforcement learning.

4. Integrate AI-Powered Document Assistant for Real-Time Querying and Content Summarization

Provide a conversational AI interface that allows users to ask questions based on uploaded documents and receive relevant, contextual answers, bridging the gap between static resources and active learning.

5. Develop a personalized quiz generation and feedback mechanism using user performance data

The goal here is to implement an AI-powered quiz module that adapts to the learner's knowledge level and weak areas. It will analyze previous quiz results and generate new questions accordingly, offering targeted feedback and guiding the learner to focus on specific topics for improvement.

6. Implement intelligent and context-aware resource linking within the learning environment for mind maps

This objective aims to connect learning resources such as lecture notes, external articles, and research papers to specific nodes within the mind maps and quiz-based mind map modules. The system will provide intelligent linking that is both contextually relevant and dynamically updated based on the student's active learning topics and performance metrics.

7. Implement adaptive mind map generation using AI and LLM-based prompt engineering techniques

This objective focuses on dynamically generating visual mind maps based on the learner's course materials and personalized learning performance. The system will extract relevant content from lecture materials and online resources to create hierarchical mind maps that evolve as the student progresses.

8. Enhance student engagement and performance tracking through an AI-powered learning dashboard

The LMS will feature a centralized dashboard that visualizes learning progress, quiz scores, time spent on tasks. This objective ensures that learners can self-monitor their academic growth and receive automated recommendations for improving their performance.

9. Ensure scalability, accessibility, and usability of the AI-powered LMS for real-world deployment.

This sub-objective involves designing the system with a microservices architecture and frontend in Angular, ensuring that the LMS is responsive, scalable, and easily accessible across different environments. The goal is to create a production-ready system that supports both instructors and learners in diverse academic settings.

10. Incorporate Note Card Generation for Efficient Knowledge Revision

Enable automated or user-assisted creation of concise note cards derived from document content to support focused, spaced repetition learning strategies.

11. Design an Interactive Chapter View Section for Streamlined Navigation

Allow learners to easily browse and engage with content at the chapter level, improving orientation and focus within complex materials.

12. Enable Real-Time Learning Analytics and Personalized Feedback

Provide dashboard visualizations of learning progress, quiz performance, and engagement metrics, enabling students to self-assess and educators to offer targeted guidance.

13. Ensure Scalable, Secure Deployment Using Microservices and Cloud Technologies

Leverage Docker, AWS, and scalable microservices to support performance, modularity, and security across the LMS architecture.

3 METHODOLOGY

The methodology for developing the AI-driven, adaptive, learner-centric Learning Management System (LMS) focuses on leveraging advanced technologies to enhance student engagement, comprehension, and academic performance. The system is designed to tackle key challenges in personalized learning by incorporating AI-powered and algorithm-based modules, such as adaptive mind map generation, personalized quiz generation, Pomodoro technique-based productivity management, and an AI-powered document assistant.

The LMS platform is structured as a modular system, where each component adaptive mind maps, quiz generation, Pomodoro-based productivity management, and the document assistant operates independently but collaboratively, offering a personalized learning experience. The system is hosted in the cloud to ensure scalability, flexibility, and seamless integration among the various modules, providing users with easy access and interaction from any device.

To enable personalized learning, the system gathers data from multiple sources, including student quiz results, uploaded documents, and user interactions with the platform. These documents are uploaded to the cloud, where they are processed and indexed for efficient retrieval. The data is preprocessed to eliminate irrelevant information and ensure it is in an appropriate format for use by the AI-powered modules.

The LMS integrates these modules into a cohesive platform that allows students to seamlessly interact with personalized mind maps, quizzes, study sessions, and documents. The user interface (UI) is designed to be intuitive and user-friendly, focusing on maximizing student engagement while minimizing cognitive overload. Students can easily navigate between modules, take quizzes, generate mind maps, access documents, and manage their productivity, all within a single platform.

To evaluate the system's effectiveness, a series of tests are conducted to assess its impact on student engagement, comprehension, and academic performance. User feedback is collected continuously to refine the system, ensuring it meets the diverse needs of learners. Performance metrics, such as quiz scores, mind map generation accuracy, and user interaction data, are analyzed to measure the success of the personalized learning experience. Additionally, the system is rigorously tested for scalability and usability, ensuring it can handle large volumes of users and deliver a smooth experience across different devices.

3.1 Requirement Gathering

The initial phase of the project focused on gathering comprehensive requirements from various stakeholders, including university students, educators, and curriculum designers, to ensure that the Learning Management System (LMS) addresses real-world academic needs. The primary objective was to understand the current limitations of traditional, static LMS platforms and to capture the expectations for a system that offers personalized, intelligent support throughout the learning journey.

The requirement gathering process involved a combination of interviews, surveys, and focus group discussions with final-year students and academic staff. The insights obtained from these sessions guided the identification of essential functionalities, including mind map generation, AI-powered document querying, adaptive quizzes, resource recommendations, and study planners. Special attention was given to ensuring the LMS not only facilitates content delivery but also promotes active engagement, self-paced learning, and personalized learning experiences.

Performance indicators for the system were defined based on key educational outcomes, such as student engagement levels, accuracy of quiz feedback, content relevance in AI-generated responses, and the perceived usefulness of mind maps and note cards. Additionally, usability metrics including interface intuitiveness and system response time were established to measure user satisfaction and overall system effectiveness.

Ethical and legal considerations were thoroughly integrated into the requirement analysis process. The LMS prioritizes data privacy through encrypted storage and robust user authentication mechanisms. The AI-powered document assistant was designed with safeguards to prevent the generation of hallucinated or misleading content, thereby enhancing the system's trustworthiness and reliability. Furthermore, the system complies with academic integrity standards and data protection regulations, ensuring that users' data and privacy are safeguarded.

By meticulously identifying the educational, technical, and ethical requirements, the development process aims to produce a robust, scalable, and pedagogically effective LMS that meets the evolving needs of modern learners and educators, fostering an environment conducive to personalized learning and academic success.

3.2 Past Research Analysis

An in-depth review of existing literature was essential to establish the foundation for developing an AI-driven, adaptive, and learner-centric Learning Management System (LMS). The integration of advanced AI technologies, including personalized mind map generation, adaptive quiz systems, and productivity management tools like the Pomodoro technique, required evaluating prior research on adaptive learning systems, personalized education, and intelligent tutoring systems. This review helped in identifying the current challenges, the effectiveness of previous approaches, and areas that could benefit from improvement in the design of such systems.

Past studies in the realm of adaptive learning have explored various methods for personalizing content delivery to students based on their learning behaviors and preferences. The application of technologies such as large language models (LLMs), retrieval-augmented generation (RAG), and AI-based adaptive learning tools has been extensively researched. Several studies focused on improving student engagement and academic performance by providing tailored content, real-time feedback, and personalized

learning paths. Additionally, studies on algorithmic approaches to managing student progress through automated quiz generation, personalized feedback, and performance analytics helped guide the development of the personalized quiz system in this LMS.

Furthermore, research on the use of mind maps in education highlighted their potential for enhancing knowledge retention and understanding by visually representing complex concepts and relationships. Adaptive mind map generation, based on individual student performance and learning patterns, was shown to significantly improve student comprehension. Studies on productivity management techniques, such as the Pomodoro technique, also provided insights into how structured work-break schedules could help reduce cognitive overload and improve focus, thus contributing to better academic outcomes.

In reviewing these past studies, key challenges in implementing such a system were identified, including the seamless integration of various AI-driven tools into a cohesive platform and ensuring that the system could effectively adapt to diverse learning needs. Additionally, performance metrics such as user engagement, task completion rates, and academic performance were crucial in evaluating the success of these adaptive learning systems.

Overall, the review of past research provided valuable insights into the potential of integrating AI-driven personalized learning tools into an LMS, helping to shape the design and functionality of the proposed system while addressing existing gaps in the current educational technology landscape.

3.3 Refer to Official Documentation

The development of the AI-driven Adaptive LMS also involved reviewing authoritative documentation and academic guidelines related to the responsible use of AI in educational

technologies. This included best practices in human-computer interaction, ethical AI design, data privacy regulations, and the principles of fair and transparent AI decision-making.

Documentation from reputable organizations such as the IEEE, ACM, and UNESCO on digital education frameworks helped shape the ethical boundaries and pedagogical goals of the system. These resources emphasized the importance of learner autonomy, content relevance, inclusivity, and explainability of AI decisions principles that were carefully integrated into the system architecture and user experience.

Technical documentation was also studied to establish guidelines on implementing LLMs for educational assistance, constructing knowledge graphs, and deploying web-based AI models securely. Moreover, resources on microservices, containerization (Docker), and cloud deployment (AWS/Azure) were referred to in order to establish a scalable, modular, and maintainable backend infrastructure.

This alignment with official documentation ensured that the system was not only technically sound but also compliant with ethical and legal standards, ultimately enhancing trust, accessibility, and academic integrity.

3.4 Identify Existing Methodologies

The process of identifying existing methodologies for adaptive learning and personalized Learning Management Systems (LMS) involves examining several approaches, each with its strengths and limitations. Traditional LMS platforms, which primarily focus on content delivery, remain widespread but often fail to provide personalized learning experiences. While these systems offer consistent access to course materials, they lack the dynamic features necessary to cater to the varying needs and learning styles of students.

Existing systems like Moodle and Blackboard have made significant strides in integrating online learning resources, assessment tools, and communication features. However, they still rely heavily on a one-size-fits-all approach and offer limited adaptability to individual student performance or preferences. These platforms typically use basic learner progress tracking but fail to provide intelligent, real-time support in areas such as personalized feedback, study planning, or engagement with content beyond passive consumption.

To overcome these limitations, more advanced approaches have integrated AI and machine learning techniques. For example, some systems leverage recommendation algorithms to suggest resources based on student activity, but they often lack deep integration with learning outcomes or personalized support. These methods improve content accessibility but often fall short in adapting to the specific needs of each learner.

AI-powered solutions, such as intelligent tutoring systems and personalized quiz generation, have been identified as significant advancements. These methodologies utilize learner data to dynamically adjust content and assessments, providing tailored learning experiences. However, these systems often require substantial computational power and data to function effectively, and they may struggle with scalability in large, diverse learner populations.

Recent developments in adaptive mind map generation, based on AI techniques have shown promising results in helping students visualize concepts and relationships within the learning material. These systems aim to enhance comprehension by providing a structured, visual representation of knowledge that evolves with the learner's progress. Yet, challenges such as ensuring the accuracy and relevance of mind maps and integrating them seamlessly into the overall learning experience remain.

Additionally, the integration of productivity management techniques, such as the Pomodoro method, has been explored in LMS systems to optimize student focus and prevent burnout. While effective for short-term productivity, these techniques must be personalized to individual study patterns and learning goals to be truly effective.

Through a comprehensive review of existing methodologies, it is clear that a hybrid approach combining AI-driven adaptive learning, personalized feedback mechanisms, and productivity optimization techniques is needed. By addressing the limitations of current systems such as lack of personalization, scalability challenges, and insufficient engagement this LMS system can leverage the strengths of existing methodologies while minimizing their weaknesses. The goal is to create a highly adaptable, learner-centric environment that promotes deeper engagement, better learning outcomes, and more efficient use of students' time.

3.5 Stakeholder Engagement

Stakeholder engagement played a pivotal role in the requirement gathering phase of our AI-driven Adaptive Learning Management System (LMS). Given the goal of creating a system that effectively supports personalized, intelligent, and interactive learning experiences, it was crucial to involve key stakeholders throughout the early stages of development. These stakeholders included students, lecturers, academic advisors, and our supervisor, Dr. Kalpani Manathunga, who provided valuable insights into the practical expectations and educational challenges faced by modern learners and educators.

Through structured discussions, interviews, and feedback sessions, we were able to identify specific needs from each user group. Students emphasized the importance of features such as personalized mind maps, note cards, chapter-based navigation, and real-time feedback through AI-driven assistance. They highlighted the need for the system to adapt to their weak areas identified via quizzes and provide targeted study material. Lecturers, on the other hand, pointed out the importance of accurate content generation, easy integration of teaching materials, and the ability to monitor student performance effectively.

Active collaboration with stakeholders ensured that the system is not only technologically robust but also aligns with real educational workflows and expectations. Their input significantly contributed to enhancing the usability, reliability, and educational effectiveness of the LMS, ensuring it can truly serve as a learner-centric and adaptive platform that meets diverse academic goals.

3.6 Project Requirements

3.6.1 Functional Requirements

- The system should allow users to register, log in, and log out securely.
- The system must support personalized quiz generation and provide adaptive feedback based on user performance.
- Users should be able to upload documents (PDF) for content extraction using the AI Document Assistant.
- The AI Document Assistant must answer questions based on uploaded materials and summarize content.
- The system should allow users to generate adaptive mind maps based on document content, prompt input, or quiz performance.
- Users can store up to four adaptive mind maps, and the system must manage replacements when the storage limit is exceeded.
- Mind maps must include resource linkings, customizable views (zoom in/out), and structured nodes based on topics.
- The system must allow users to access note cards, linked to chapters or concepts from the mind maps.
- A chapter-wise content viewer must be provided for structured navigation of uploaded study material.

- The Pomodoro timer feature should be available to help users manage study sessions effectively.
- The system should notify users with relevant updates (e.g., quiz results, new mind map generated, reminders).
- User-friendly error messages and success feedback must be displayed for actions like mind map generation or quiz submission.
- The LMS web application should be accessible across multiple browsers and screen sizes for usability and accessibility.

3.6.2 Non-Functional Requirements

Performance

- The system should be responsive and able to handle simultaneous quiz submissions, mind map generations, and document processing.
- It should maintain stable performance during peak usage, especially when processing large documents or user data.
- The backend must support scalable services for LLM queries, mind map generation, and quiz handling.

Reliability

- The system should maintain at least 99% uptime with auto-recovery for critical services like mind map generation and document analysis.
- Contingency mechanisms should be implemented to handle temporary failures in the LLM or prompt generation processes.

Security

- User data including quiz responses, documents, and mind maps must be encrypted at rest and in transit using SSL/TLS.
- Role-based access control and secure authentication (e.g., JWT, OAuth) must be implemented to protect sensitive content.

Usability

- The interface must be intuitive, allowing users to create quizzes, generate mind maps, and browse notes with minimal guidance.
- The mind map view should offer zoom in/out and save features for better visualization.
- Prompts for document-based mind map generation should be clearly labeled

Maintainability

- Modular system design must be adopted for easy updates and bug fixing (e.g., quiz engine, LLM interaction module, UI components).
- Clear documentation for developers should be maintained, outlining each module's responsibilities and integration points.

Interoperability

- The system should work across modern browsers and be compatible with different screen sizes (desktop, tablet).

3.6.3 User Requirements

- Users must be able to securely register and log in to access learning materials and AI features.
- Authorized users should be able to upload learning content and receive personalized quizzes and summaries.

- Mind maps should adapt based on uploaded documents or quiz performance, providing focused learning paths.
- Users must be able to zoom in/out of mind maps and click on nodes to view linked resources or notes.
- The system should notify users when new adaptive mind maps are available or if a previously saved mind map is replaced.
- Quizzes must include real-time feedback and explanations for answers to support student improvement.
- The system must include Pomodoro session tracking, alerting users when to take breaks and resume studies.
- Navigation across chapters, notes, quizzes, and mind maps should be smooth and consistent.
- Users' personal data and content should be protected with strict confidentiality and encryption practices.
- Real-time feedback and error messages should guide users in resolving any issues with uploads or actions.

3.6.4 System Requirements

- The LMS uses MongoDB for storing user data, quiz attempts, uploaded documents, and generated mind maps.
- The backend is developed with Node.js, offering RESTful APIs to handle authentication, file uploads, mind map generation, quiz delivery, and Pomodoro tracking.
- Angular is used for the frontend to create a responsive UI, integrating modules like mind maps, chapter views, and note cards.
- The system integrates with Azure Custom Vision, OpenAI, and external knowledge bases to enhance resource generation and quiz adaptation.

- Prompt engineering techniques using Mermaid Js support dynamic mind map rendering and visualization.
- The system supports Docker and GitHub for deployment and version control, and AWS/Azure for cloud scalability.
- Performance optimization is applied using asynchronous processing and efficient resource management for prediction and quiz delivery.

3.7 Feasibility Study

This system leverages advanced machine learning techniques, mind map generation using LLMs and Mermaid.js, adaptive feedback loops, and integration with cloud services such as Azure and GitHub. From a technical standpoint, the LMS requires a robust backend with scalable infrastructure, intelligent content generation modules, and a seamless Angular-based frontend for dynamic user interactions. Economically, the project emphasizes long-term cost reductions through automated learning assistance and reduced teacher intervention, balanced against initial development and cloud hosting expenses. Schedule feasibility is achieved by adopting a milestone-based development plan, covering user requirement analysis, module integration, user testing, and deployment. Ultimately, the LMS enhances student learning, personalization, and academic performance through intelligent content delivery and self-regulated learning support.

3.7.1 Technical Feasibility

The technical feasibility of the adaptive LMS system centers on integrating several advanced technologies, including AI, machine learning, cloud computing, and modular design principles. The system is built using a combination of robust backend technologies such as Node.js, cloud-based storage, and AI-powered modules that ensure scalability, flexibility, and seamless integration of the system's components. These technologies align with the system's goals of providing personalized learning experiences, managing large volumes of data, and ensuring real-time adaptability to learner needs.

For AI-driven features like adaptive mind map generation and personalized quiz generation, computational power is a crucial consideration. Using cloud infrastructure (e.g., AWS or Azure) will ensure the necessary resources for AI model training, data processing, and quick delivery of services. Data privacy and security measures, such as encrypted storage and secure user authentication, also adhere to the latest standards and regulations. The system's modular structure will ensure that components, like the AI-powered document assistant and Pomodoro-based productivity management, can be updated or replaced without affecting the overall operation of the LMS.

3.7.2 Hardware Requirements

To support efficient delivery of adaptive learning content, high-performance servers are necessary, particularly for backend services managing LLMs, quiz analytics, and image/PDF processing. Storage solutions must be scalable (e.g., AWS S3, MongoDB Atlas) to manage user data, mind maps, documents, and learning resources. GPU-enabled instances may be used for training or inference tasks in mind map generation. Client-side applications (Angular frontend) are optimized for cross-browser and responsive, requiring only standard modern computing devices.

3.7.3 Knowledge of Deployment and Integration

The deployment strategy for the adaptive mind map generator was designed to support scalability, modularity, and continuous integration. The backend, built with Node.js, and the frontend in Angular were containerized using Docker. These containers were orchestrated and managed via AWS services and automated through GitHub Actions for CI/CD. MongoDB Compass was employed to store and retrieve structured data including prompts, mind maps, and quiz performance metrics. The system's architecture allowed for modular upgrades and easy debugging, ensuring a robust deployment environment that supports long-term expansion and real-world usage.

3.7.4 Bloom's Taxonomy Integration for Quiz Personalization

The LMS incorporates Bloom's Taxonomy as a framework to personalize quizzes and adaptive content generation. Each quiz question is categorized into cognitive levels such as Remember, Understand, Apply, Analyze, Evaluate, and Create. Based on a learner's performance in each category, the system intelligently identifies areas needing improvement and regenerates focused content. This categorization enables the system to adapt learning materials based on not just what students know, but how deeply they understand a topic, leading to a higher-order learning experience.

3.7.5 Resource Integration using Web Search API s

Each mind map node is enhanced with contextually relevant resources including external links, embedded videos, textbook references, and summary documents. These resources are dynamically linked based on the semantic meaning of the node and the user's learning progress. By integrating educational APIs (e.g., Google Search API or YouTube Data API), the system can fetch high-quality materials to supplement core learning, making the learning environment richer and more informative. Resource nodes adapt over time based on quiz performance and student interactions.

3.7.6 Retrieval-Augmented Generation (RAG) and LLM Integration

Retrieval-Augmented Generation (RAG) is employed to enhance content accuracy and contextual relevance in the LMS. When a student uploads a PDF or asks a question, relevant content is first retrieved from vectorized document chunks using embeddings. This context is then passed into a Large Language Model (LLM), such as GPT, which generates answers or explanations based on the retrieved content. This combination of retrieval and generation ensures the AI assistant provides grounded and context-aware

responses. The RAG module is optimized to exclude irrelevant content, reduce hallucinations, and ensure alignment with educational standards.

3.7.7 Knowledge of applying Prompt Engineering Techniques

Gained a deep understanding of prompt engineering techniques as we applied them to various components of our LMS system, such as adaptive mind map generation, personalized quiz creation, and the AI-powered document assistant. By crafting specific prompts for each feature, we were able to guide the AI to generate tailored, relevant content that enhanced the user experience. For mind maps, we used prompt engineering to generate context-specific relationships between concepts, while for quizzes, it helped create personalized questions based on the learner's progress. Additionally, the document assistant leveraged these techniques to provide precise, on-demand responses, showcasing the power of prompt engineering in optimizing the system's adaptability and effectiveness.

3.7.8 Economic Feasibility

The economic feasibility involves evaluating costs related to software development, third-party services (LLM APIs, Azure), and maintenance, versus long-term educational benefits. Initial costs include development time, cloud hosting, and integration efforts. However, the LMS reduces the need for constant instructor involvement, enabling scalable delivery of customized learning. Improved academic performance and retention directly contribute to institutional ROI. An economic analysis demonstrated that the long-term cost savings through enhanced learning efficiency and reduced dropout rates far outweigh the upfront investments.

3.7.9 Schedule Feasibility

The schedule feasibility of the AI-driven Adaptive Learner-Centric Learning Management System (LMS) was meticulously evaluated through an analysis of the project lifecycle to ensure that the system is delivered on time, with all features implemented according to defined objectives. A structured project timeline was created, outlining each critical phase of the project, including literature review, requirement analysis, system design, development, integration of AI services, testing, and final deployment.

3.8 Overall System Diagram

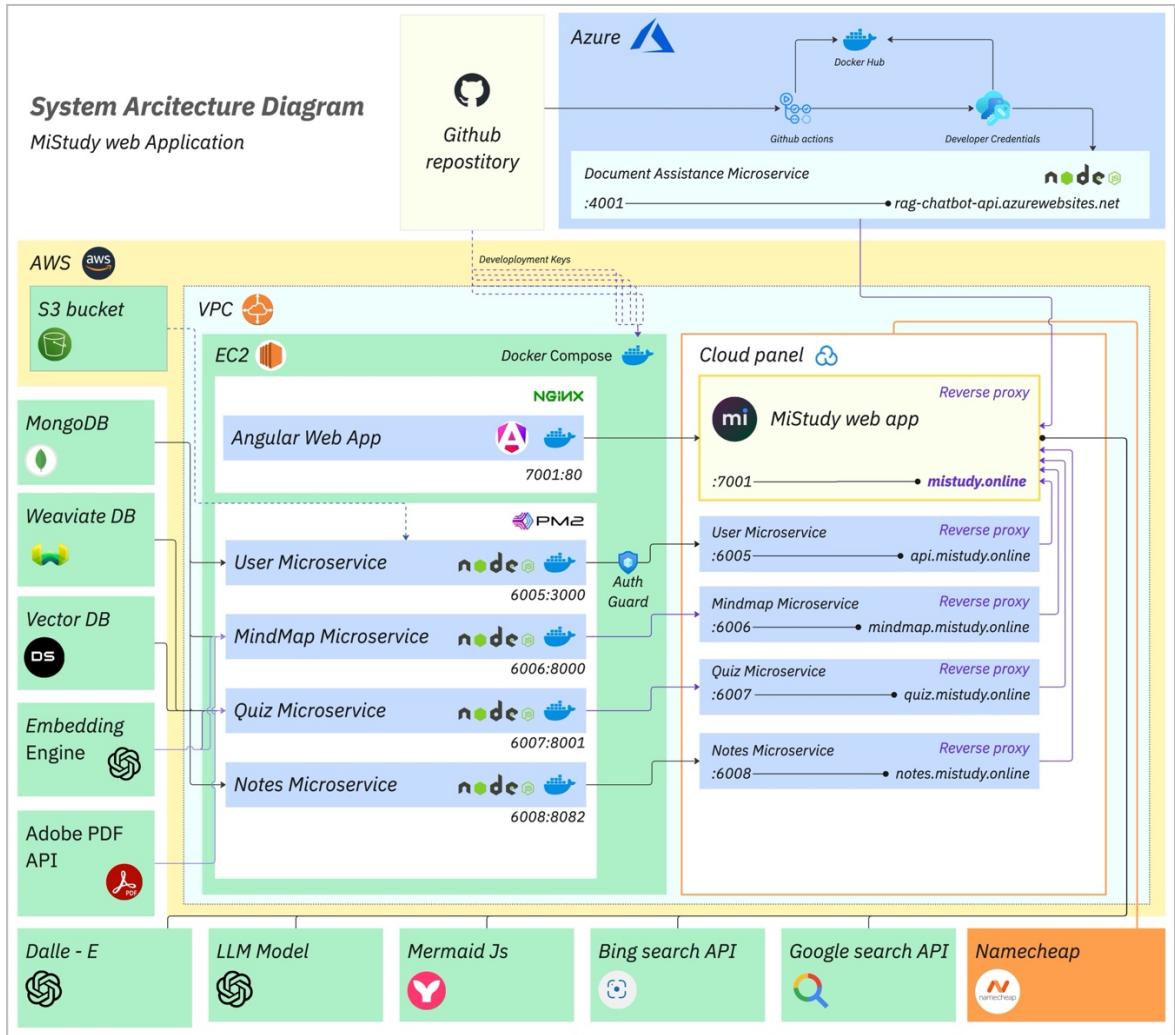


Figure 1 System Architecture Diagram

The MiStudy Web Application is a modern, cloud-based educational platform that leverages a microservices architecture to deliver a robust and scalable experience to its users. At the heart of the system is an Angular-based web application that serves as the primary interface for students, educators, and administrators. This frontend is deployed on an Amazon EC2 instance and served through NGINX, with external access managed via

a reverse proxy, allowing users to interact with the platform through a clean and secure domain, mistudy.online.

The backend of MiStudy is composed of multiple Node.js-based microservices, each encapsulated in Docker containers and orchestrated using Docker Compose. These services include the User Microservice, which handles authentication, user data, user sessions and user documents; the Mind Map Microservice for AI assisted diagramming, the Quiz Microservice, which manages quiz creation, validation and result analysis, and the Notes Microservice, responsible for managing user notes. These services run on distinct ports within the EC2 environment but are exposed to the public via reverse proxy endpoints like `api.mistudy.online`, `quiz.mistudy.online`, and others, ensuring modularity, security, and maintainability. PM2 is used within the EC2 instance to monitor and manage these Node.js processes efficiently, providing resilience and uptime assurance.

To support the platform's features, several databases and external APIs are integrated into the system. MongoDB serves as the primary database for storing user data, quiz records, notes, and mind maps. The architecture also utilizes Weaviate DB and a Vector DB for advanced semantic and vector-based search capabilities, enhanced by an Embedding Engine powered by OpenAI models. These elements enable features like AI-powered content search and intelligent recommendations. Additionally, the system integrates with Adobe's PDF API for generating and handling documents and supports visual enhancements and content generation using tools like DALL·E and Mermaid.js.

Another components of the MiStudy ecosystem is the Document Assistance Microservice, which is hosted on Microsoft Azure. This service is separately deployed through Azure's Docker and CI/CD ecosystem, using GitHub Actions and Docker Hub to automate the deployment process. It is accessible via `rag-chatbot-api.azurewebsites.net` and supports features like context-aware document search and intelligent summarization using large language models (LLMs). The service is authenticated using securely stored development credentials, ensuring safe integration with the core application.

To maintain a clean and secure routing structure, a centralized cloud panel reverse-proxies all internal services to their corresponding public domains. Security across services is enforced through an authentication guard mechanism that verifies tokens and protects internal APIs from unauthorized access. The system is also integrated with third-party APIs like Bing and Google Search for enhanced information retrieval within the learning environment. Domain management is handled via Namecheap, which registers and manages the mistudy.online domain and its subdomains.

Altogether, the MiStudy web application's system architecture exemplifies a well-structured, modular, and cloud-native approach to delivering an AI-powered learning platform. It combines the flexibility of microservices, the scalability of cloud infrastructure, and the intelligence of modern AI tools to provide an engaging and adaptive educational experience.

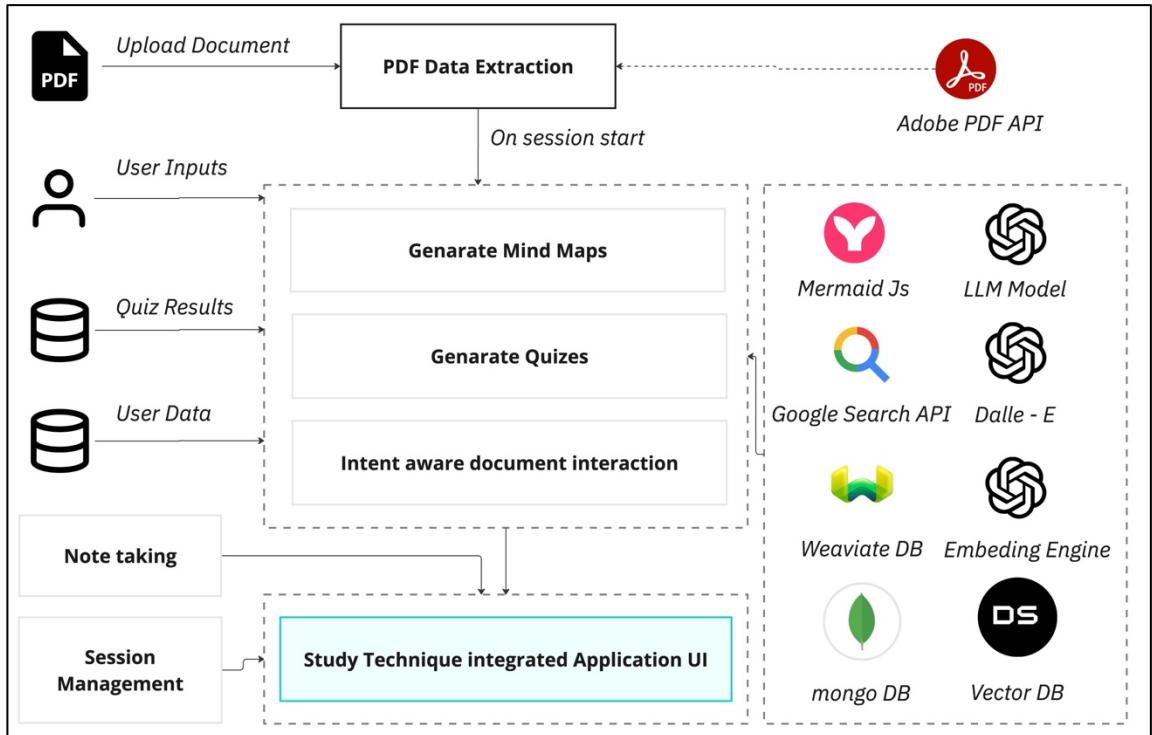


Figure 2 System Flow Diagram

The system initiates with a study technique-based session and productivity manager that engages the user by setting up personalized work and break cycles, helping to establish a focused learning environment from the start. Once upload a PDF, users are directed to the main dashboard, where they can explore key features such as adaptive mind map generation, AI-driven document assistance, and performance-based quizzes. The mind maps are generated using retrieval-augmented generation and prompt engineering techniques, adapting dynamically to each student's strengths and weaknesses. Simultaneously, the AI PDF Assistant allows learners to ask context-aware questions based on uploaded materials, ensuring accurate and relevant information delivery. A structured chapter view enhanced with interactive note cards further supports effective navigation and content retention.

3.8.1 Study Techniques and Cognitive Load Management

The Study Techniques and Session Management component of the LMS plays a central role in transforming passive learning into an engaging and behaviorally optimized experience. This module is specifically engineered to minimize procrastination, reduce cognitive overload, and increase student engagement by intelligently integrating proven psychological study strategies within a unified interface. Unlike traditional LMS systems that separate content from learning methods, this system embeds time-tested techniques directly into the student workflow.

At the architectural level, the session management system uses a real-time engagement tracker built using Angular, integrated with backend services in Node.js. This tracker captures mouse movements, tab focus states, session duration, and user activity to monitor engagement levels and determine the most effective times for breaks and study resumption. These signals are used to automatically manage and update timers based on dynamic user behavior.

The platform incorporates a hybrid approach combining several established study techniques, such as the Pomodoro Technique, Flowtime, Spaced Repetition, and Feynman Technique, each of which can be enabled or customized based on learner preferences. Users can select between rigid time cycles (as in Pomodoro) or flexible attention-based cycles (Flowtime), and the system adapts in real-time using reinforcement learning algorithms to find the most effective patterns for each student.

In the backend, MongoDB stores session logs, user preferences, learning patterns, and behavior metrics. TensorFlow models analyze this data to recommend optimized schedules for users based on historical effectiveness. For instance, if a user shows higher retention during Flowtime-based sessions, the system suggests adapting future study sessions accordingly. Additionally, the Spaced Repetition engine flags content for review based on past quiz interactions, adjusting the review schedule to the learner's long-term memory curve.

The UI/UX layer features an intuitive session dashboard that enables users to start, pause, or adjust study cycles, select techniques, and view real-time feedback on their engagement metrics. Features such as brightness control, distraction-free modes, background audio integration (Mozart Effect), and split-screen support further enhance the study environment.

Customization is a key element of this module. Users can personalize font sizes, highlight systems, timers, and reminders all of which are synchronized with the session management logic. Furthermore, learners receive visual cues and motivational nudges based on engagement scores, helping sustain focus over longer periods.

Performance metrics and session analytics are visualized using Angular charts and Ant Design components, providing learners with insight into their behavior patterns. Data includes study time distribution, focus ratios, interruption frequency, and session quality scores over time.

Deployment is handled within the same Docker and AWS infrastructure, ensuring that engagement-tracking services are containerized and scalable. The component integrates with the broader LMS authentication system, ensuring secure access and consistent user state management across sessions. GitHub Actions handles continuous integration and updates for both client and server modules, ensuring that any enhancements to study technique logic or analytics are seamlessly deployed.

3.8.2 AI document assistant

When a user uploads a lecture document to the system and interacts with it, tokens that are matching the student's query will be retrieved from the vector store by performing a similarity search, and the result will be queried to the LLM. The response will then be displayed to the student through the chat interface.

The process begins with the ingestion of student-uploaded documents. These documents are processed by extracting their text content using LangChains PDFLoader interface. This step ensures that all relevant textual information such as paragraphs, code blocks, or headings is converted into a machine-readable format, stripping away non-essential elements like images or formatting artifacts unless contextually significant. The extracted text serves as the raw input for subsequent processing, enabling the system to work with diverse educational materials.

The extracted text content then undergoes a chunking process to segment it into manageable units for retrieval and generation. Each document is divided into chunks based on a predefined size (1536 tokens) with an overlap (300 tokens) to preserve context across segment boundaries. This chunking strategy, determined through experimentation, balances granularity for precise retrieval with sufficient context for coherent generation, addressing challenges like information loss or noise.

The chunked content is then processed by OpenAI's Embedding Engine, which converts each chunk into a dense vector representation. This embedding process captures semantic

meaning, enabling efficient similarity-based retrieval. The resulting vectors are stored in a vector store, AstraDB. During operation, when a student submits a query, the system embeds the query using the same engine, retrieves the top 3 most relevant chunks from the vector store via cosine similarity, and passes them to the RAG pipeline for response generation. A prompt template is used with the OpenAI's ChatOpenAI interface before feeding them to the LLM, gpt-4o-mini to generate a tailored, pedagogically sound response. The prompt is designed by using prompt engineering techniques such as few shot and chain of thoughts.

3.8.3 Quiz generations

The proposed system architecture integrates multiple advanced technologies to create a personalized quiz generation platform that adapts to the learner's performance and knowledge. The system is designed to dynamically adjust quiz difficulty, provide real-time feedback, and guide the learner's progression through Bloom's Taxonomy levels.

At the core of the system, user performance data, including quiz results and cognitive levels achieved, is collected through the LMS interface, built with React.js. This data is processed in real-time to analyze the learner's strengths and weaknesses. The backend, developed using Node.js, acts as the bridge between the frontend interface and the underlying technologies. Weaviate and MongoDB are used for storing and managing two distinct types of quiz data: short-answer questions (SAQs) and multiple-choice questions (MCQs), respectively. Weaviate handles semantic search for SAQs, while MongoDB stores MCQs, ensuring efficient data retrieval and persistence.

Large Language Models (LLMs), such as GPT, are integrated into the system to analyze the learner's quiz performance and adjust the difficulty of future quizzes based on their cognitive progress. These models also generate personalized questions that are aligned with Bloom's Taxonomy, ensuring that the questions reflect the learner's current cognitive level and push them to the next level of mastery. The system adjusts question

difficulty dynamically based on the learner's performance, scaling the complexity of the material accordingly.

For the quiz generation, the system uses a feedback loop that evaluates the learner's past performance and provides tailored recommendations for improvement. Each question type (MCQs and SAQs) is adjusted based on the learner's current knowledge state, providing a more adaptive learning experience.

On the frontend, the learner's progress is displayed through an interactive dashboard built using Angular. This dashboard visualizes performance across various Bloom's Taxonomy levels and offers insights into strengths and weaknesses. The dashboard also features progress tracking, showing the learner's journey through different cognitive levels and materials.

The system utilizes cloud-based deployment within Dockerized environments using AWS, ensuring high scalability and secure resource management. The continuous integration and deployment process is automated via GitHub Actions, enabling automated testing, builds, and deployment.

3.8.4 AI-driven Adaptive Mind Map Generation

The proposed system architecture integrates multiple advanced technologies to create a seamless and efficient platform for adaptive learning and mind map generation. The architecture is designed to handle the complexity of personalized learning by combining AI-driven content analysis, dynamic mind map generation, and interactive visualizations within a unified system.

At the core of the system, user data specifically quiz results and performance metrics are collected through the LMS interface, built with Angular. This data is then processed and analyzed to identify the learner's weak areas. The backend, developed using Node.js, serves as the communication bridge between the frontend interface and the underlying

technologies. MongoDB stores user-related data, including quiz results, mind maps, and user-specific prompts, ensuring data persistence and easy retrieval.

LLMs such as GPT are integrated into the system to assess quiz performance and provide personalized learning suggestions. The LLMs also play a pivotal role in generating dynamic prompts that help guide the learner's study approach. These AI-driven prompts are leveraged to generate personalized mind maps that target specific areas where the learner needs improvement.

The Mermaid.js library is employed to generate interactive, hierarchical mind maps, which are rendered dynamically on the frontend. These mind maps can be customized by the user and are updated based on their ongoing quiz results. The system allows up to three stored mind maps per user, automatically replacing the oldest map as new ones are generated.

In terms of resource generation, the system integrates Google Search API, Bing Search API, and DALL·E to dynamically fetch relevant resources and create customized visual aids. These resources are seamlessly embedded into the mind maps, enriching the learning content and further personalizing the educational experience.

The deployment of the system is managed within Dockerized environments using AWS, ensuring scalability, security, and efficient resource management. Continuous integration and deployment are handled through GitHub Actions, allowing for automated testing, builds, and updates.

3.9 Development Process

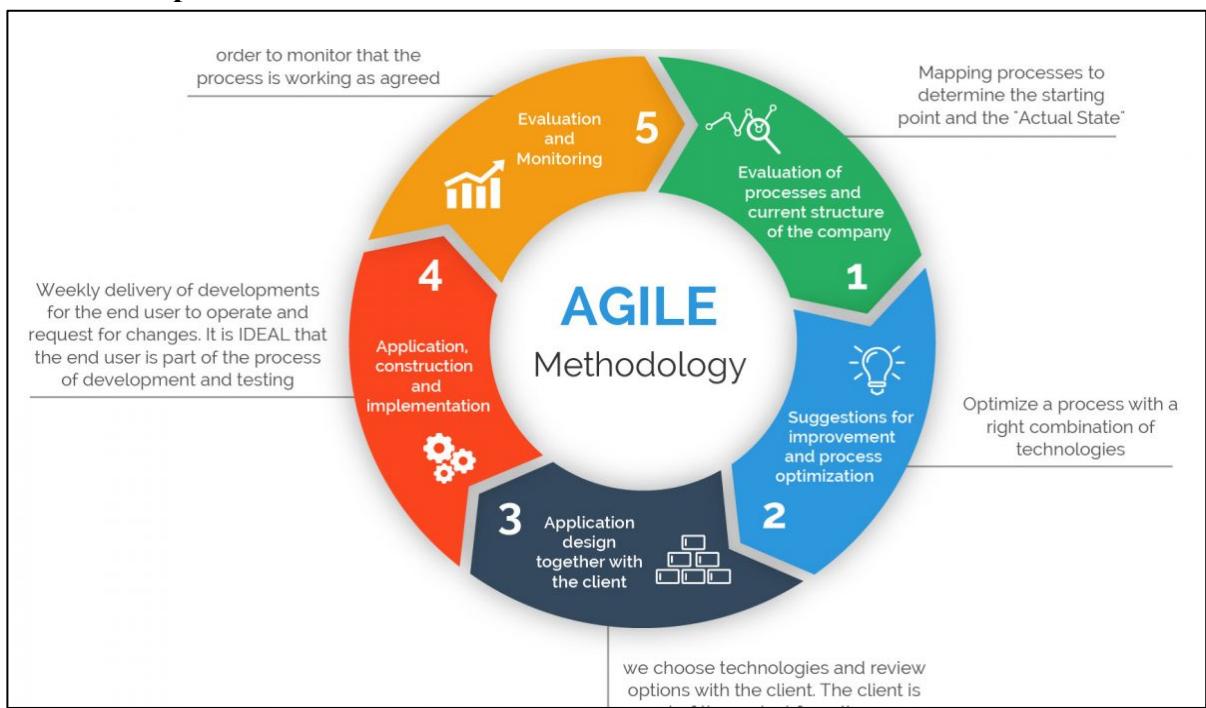


Figure 3 Agile Methodology

The development of the AI-driven Adaptive Learner-Centric Learning Management System (LMS) will adopt a hybrid Agile development approach to ensure flexibility, efficiency, and timely delivery of core features. Traditional Software Development Life Cycle (SDLC) models often lead to overheads due to their linear nature and lack of adaptability in concurrent development and iterative improvements. To overcome these limitations, Agile frameworks such as Scrum, Kanban, Extreme Programming (XP), and Feature Driven Development (FDD) will be utilized, ensuring a more responsive and dynamic development process.

- **Scrum Meetings:** Daily stand-up meetings will be held to assess progress, identify blockers, and ensure that development is moving forward smoothly. Scrum will also enable iterative development, with each sprint focusing on a specific subset of features.

- Kanban Board: A Kanban board will be maintained to visualize the workflow and track the progress of tasks. This board will be used to manage the various components of the LMS development, such as frontend design, backend integration, AI model deployment, and performance optimization. The Kanban board will help track user stories, bugs, and feature requests, ensuring smooth transitions between tasks.

- Extreme Programming (XP): The development team will adopt XP practices to ensure code quality and collaboration. This includes:

Pair programming to enhance code quality and foster knowledge sharing among team members.

- Test-driven development (TDD) to ensure that all core LMS components are thoroughly tested before deployment.

Continuous integration (CI) through automated deployment pipelines, ensuring that new code is regularly integrated and tested with the existing system.

- Feature Driven Development (FDD): The development of the LMS will follow an FDD approach in the sense that the system's development will be broken down into distinct features that align with the overall user requirements. Each feature, such as personalized quiz generation, interactive mind maps, and resource linkages for mind maps, will be developed and tested individually, ensuring that new functionalities can be quickly iterated and delivered.

3.10 Project Breakdown

3.10.1 Gantt Chart

No	Assessment / Milestone	2024- 2025												
		April	May	June	July	August	September	October	November	December	January	February	March	April
1	Research group formation													
2	Supervisor selection													
3	Brainstorming workshop 1													
4	Selection of research topic													
5	Co-supervisor selection													
6	Brainstorming workshop 2													
7	Feasibility and background study													
8	Topic registration from submission													
9	In-depth feasibility and background study 1													
10	External supervisor selection													
11	Topic assessment form submission													
12	Topic assessment from evaluation													
13	In-depth feasibility and background study 2													
15	Proposal presentation													
16	Individual proposal report submission													
17	Implementation of research work(upto 50%)													
18	Progress presentation 1													
19	Prepare and submit research paper													
20	Implementation of research work(upto 90%)													
21	Progress presentation 2													
22	Integration of the research work													
23	Project completion													
24	System testing													
25	Website and final report preparation													
26	Final presentation													
27	Final report submission													

Figure 4 Gantt Chart

3.10.2 Work Breakdown Structure



Figure 5 Work Breakdown Structure

3.10.3 Click up work breakdown schedule

The screenshot shows a work breakdown schedule in ClickUp. The interface includes a top navigation bar with 'New', 'Automate', 'Ask AI', 'Share', and a search bar. Below the navigation is a toolbar with icons for List, Board, Team, Calendar, Timeline, Activity, Mind Map, Table, Gantt, and View. A sidebar on the left contains icons for Home, Research, List, Board, Team, Calendar, Timeline, Activity, Mind Map, Table, Gantt, and View. The main area displays a list of tasks under a 'DONE' section. The tasks are listed in a hierarchical structure:

- sign up view and backend implementation for user login
- PDF Text Extraction
- Integrate with LLM API
- Implement RAG
- Conversational Implementation
- Chat history
- Find methods to do the pdf extractions
- create base UI for upload pdf contents.
- create dashboard
- Research about best study techniques
- Find which study techniques can be incorporated into online workspace.
- Find the best solution for pdf extraction
- Pdf Extraction Implementation
- Find a way to map the pdf extraction content into hierarchical way

Each task has columns for Assignee (represented by small profile icons), Due date (calendar icon), Priority (color-coded square), and more options (three dots). A gear icon is visible in the bottom right corner of the main area.

Figure 6 Click up work breakdown schedule

3.11 Design Diagrams for the Components

3.11.1 Use Case Diagrams

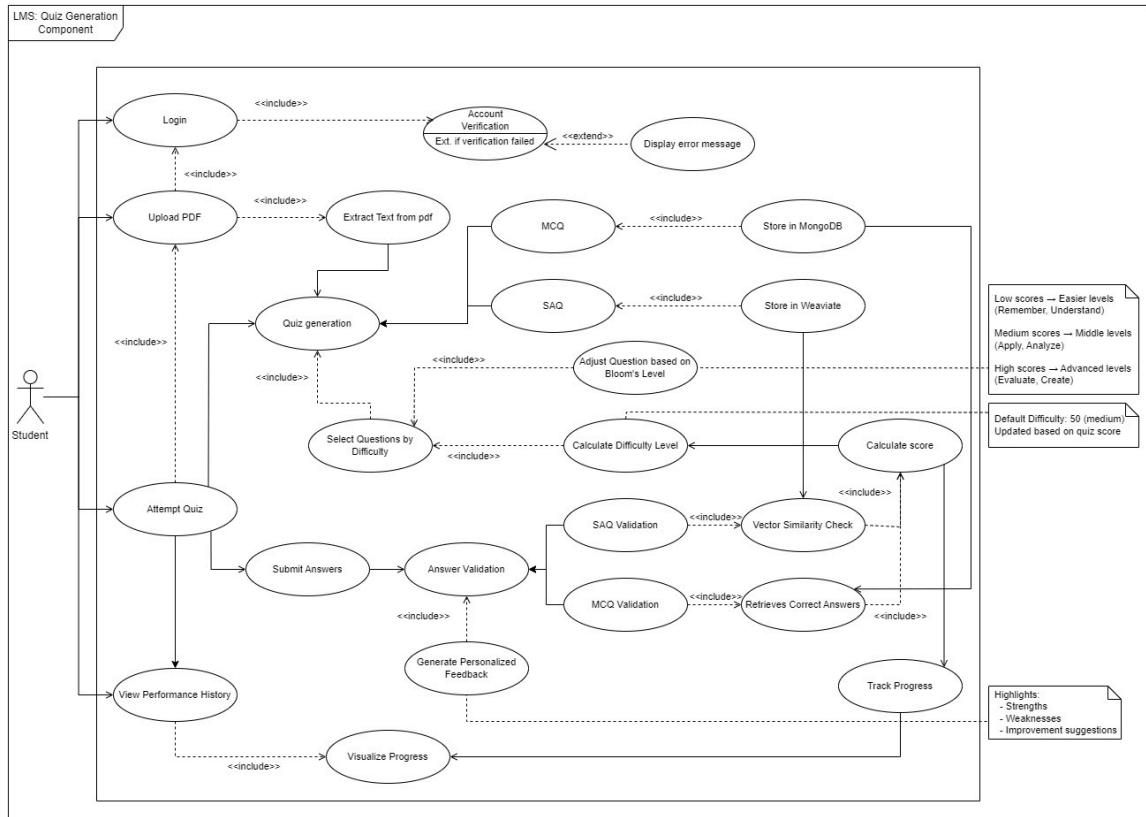


Figure 7 Use Case Diagram - Quiz Component

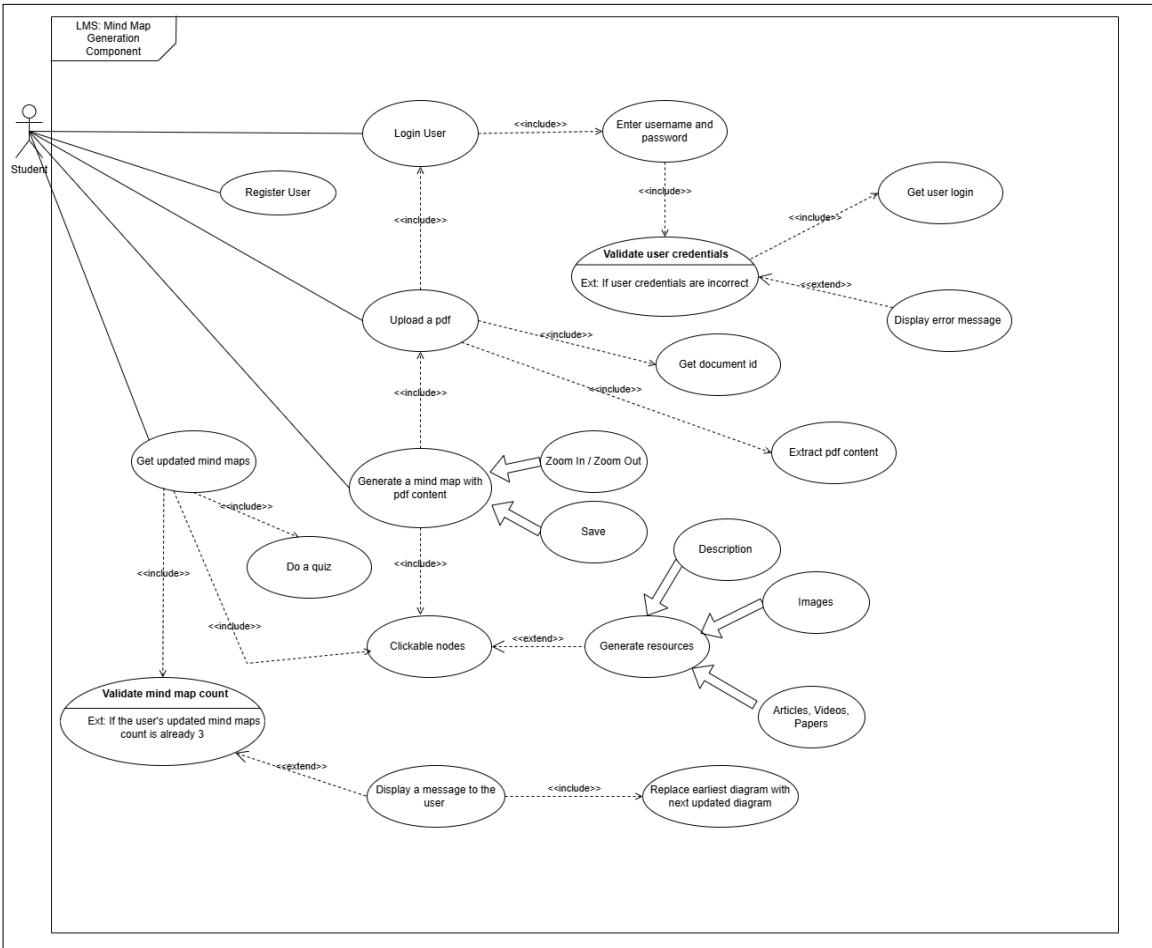


Figure 8 Use Case Diagram - Mind-Map Component

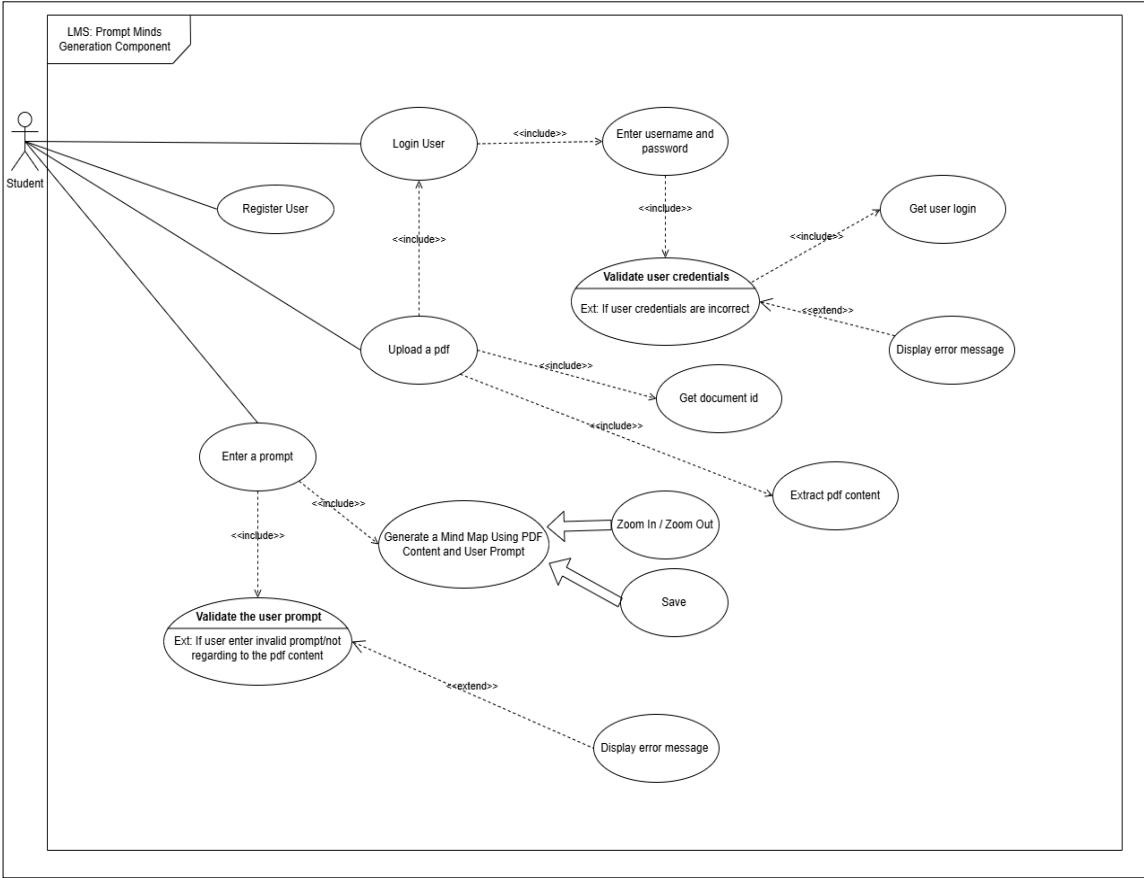


Figure 9 Use Case Diagram - Prompt Minds Component

3.11.2 Sequence Diagrams

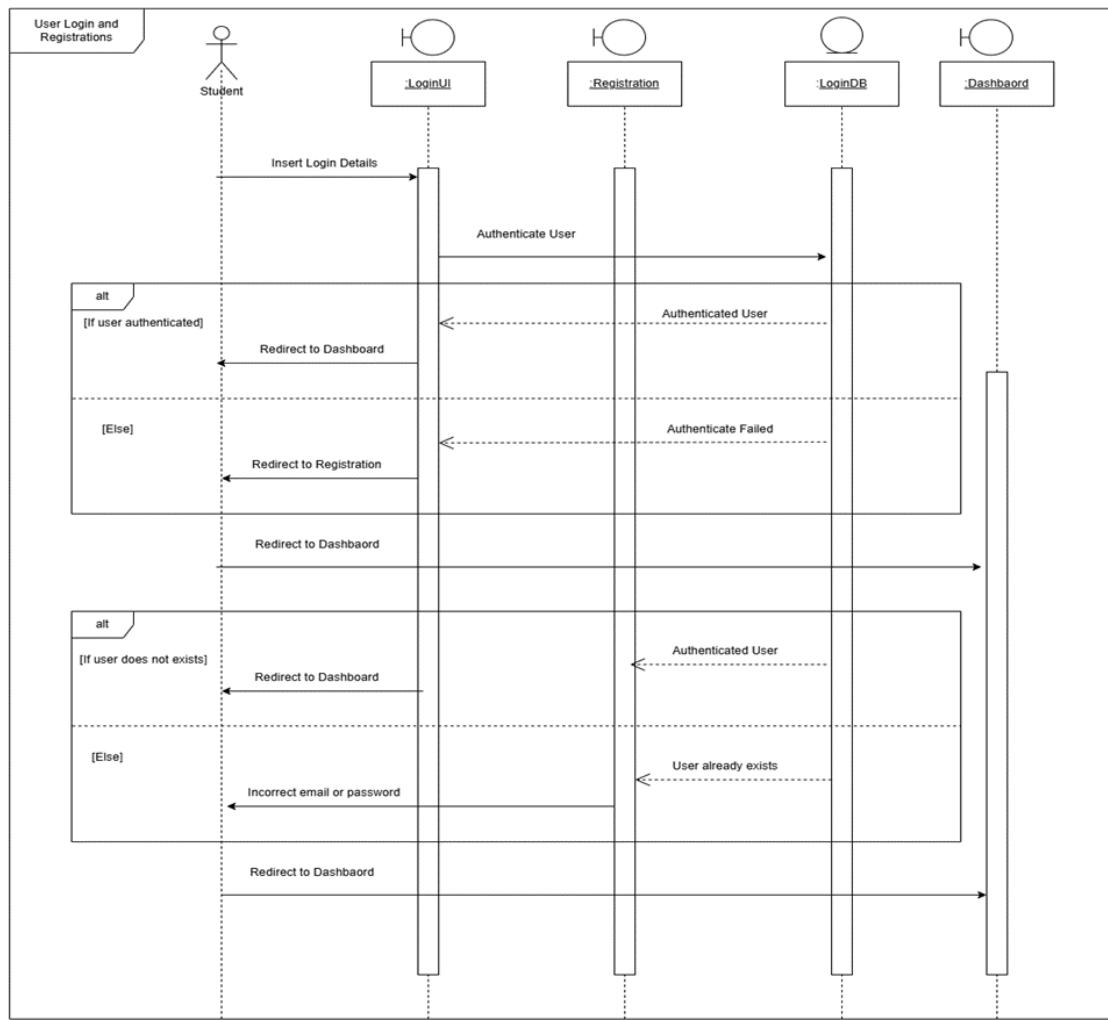


Figure 10 Sequence Diagram - User Login

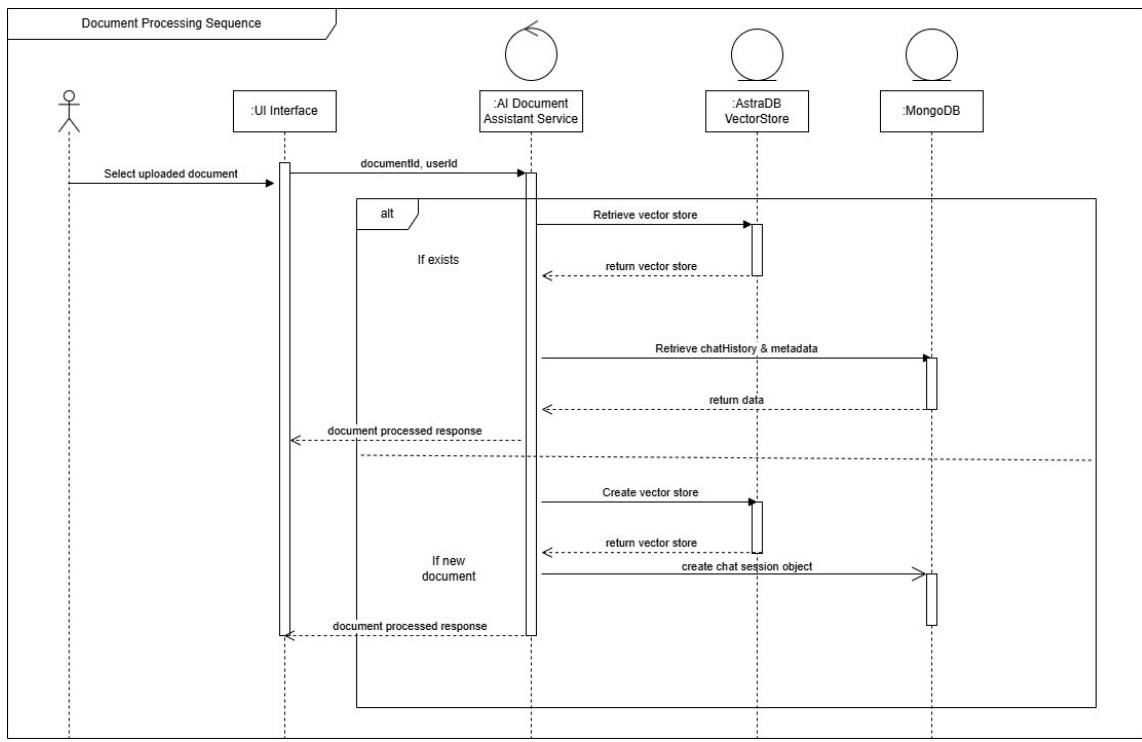


Figure 11 Sequence Diagram - Document Processing in AI Document Assistant

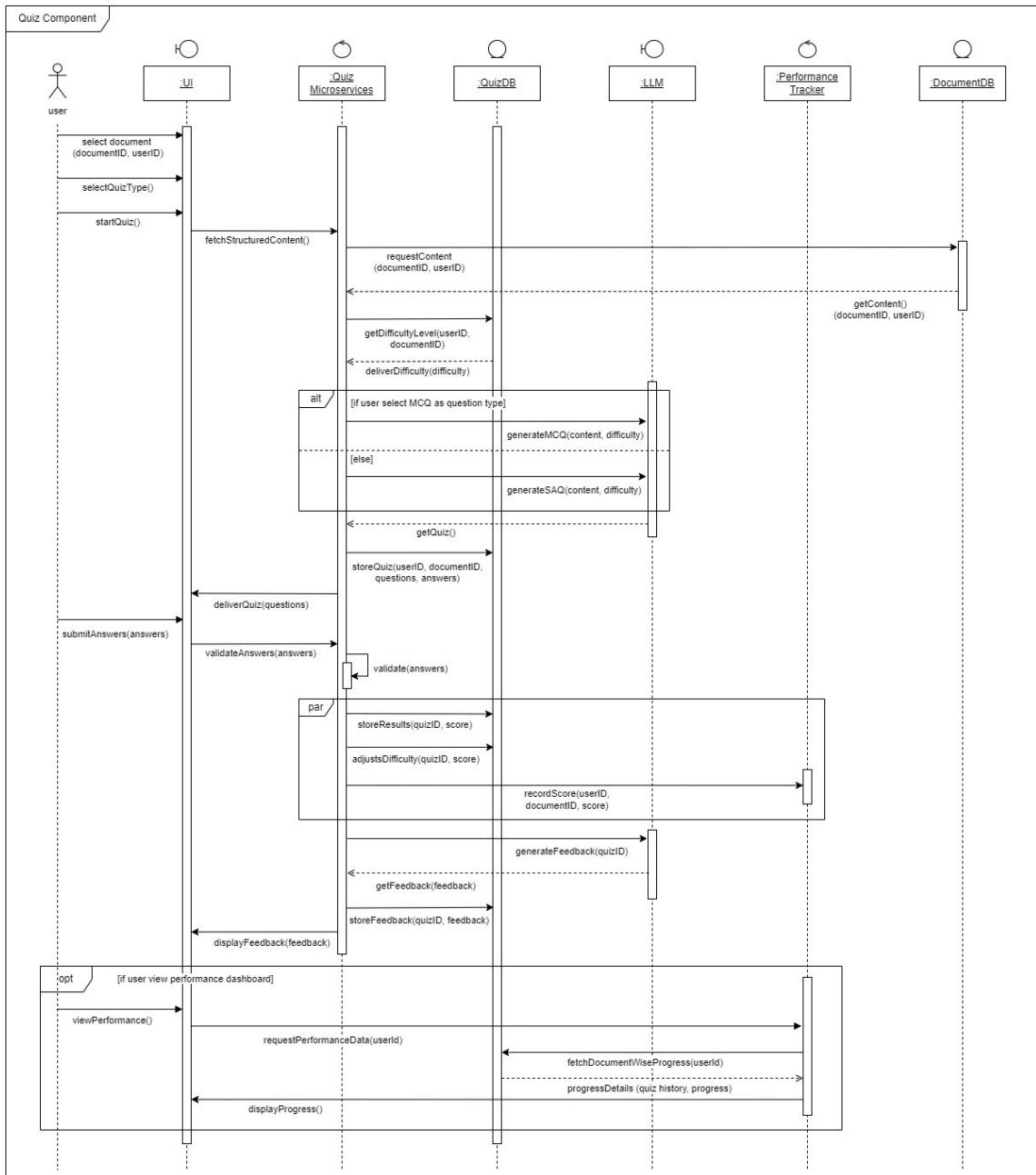


Figure 12 Sequence Diagram - Quiz Component

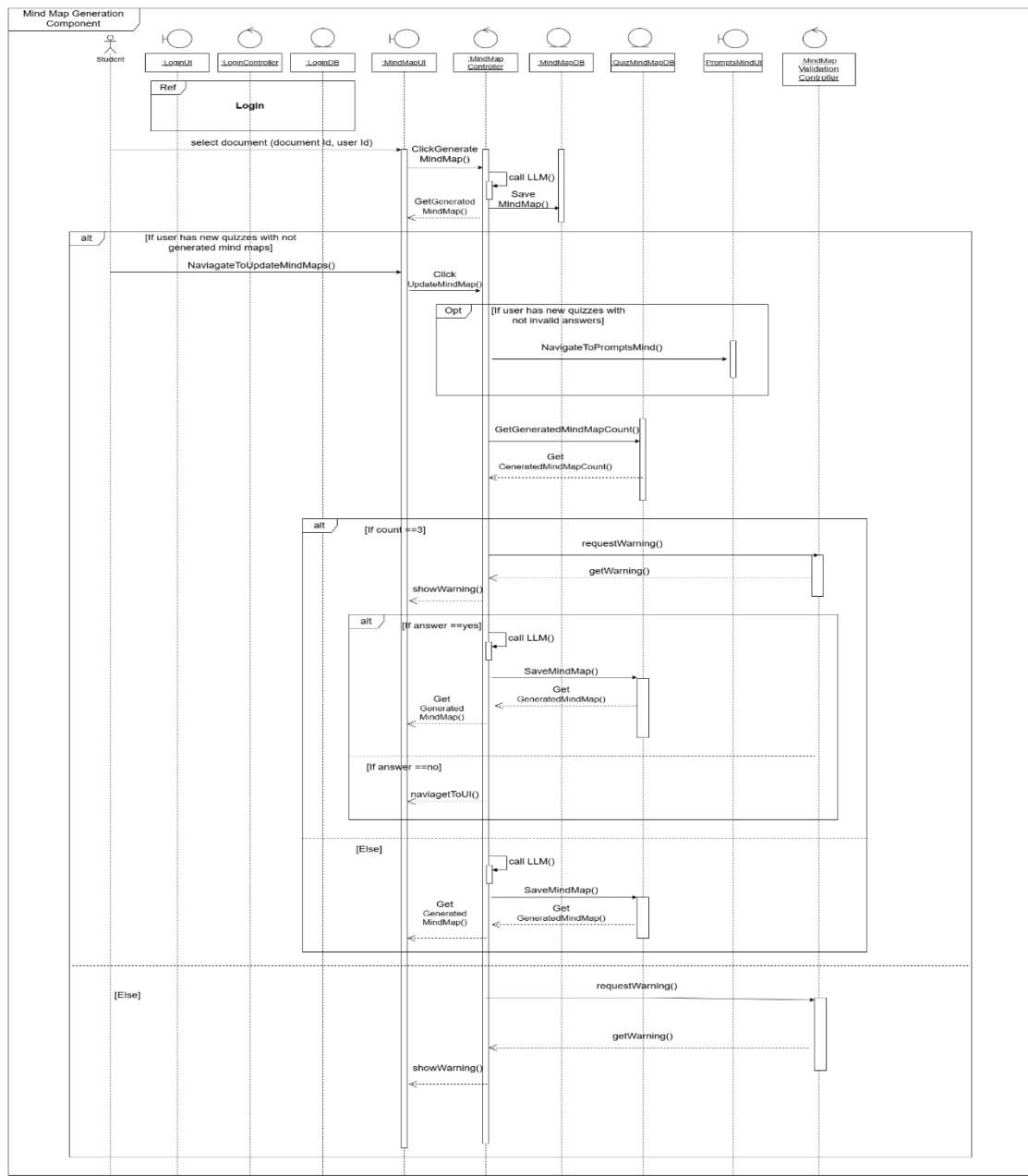


Figure 13 Sequence Diagram - Mind-Map Component

3.11.3 Component Diagrams

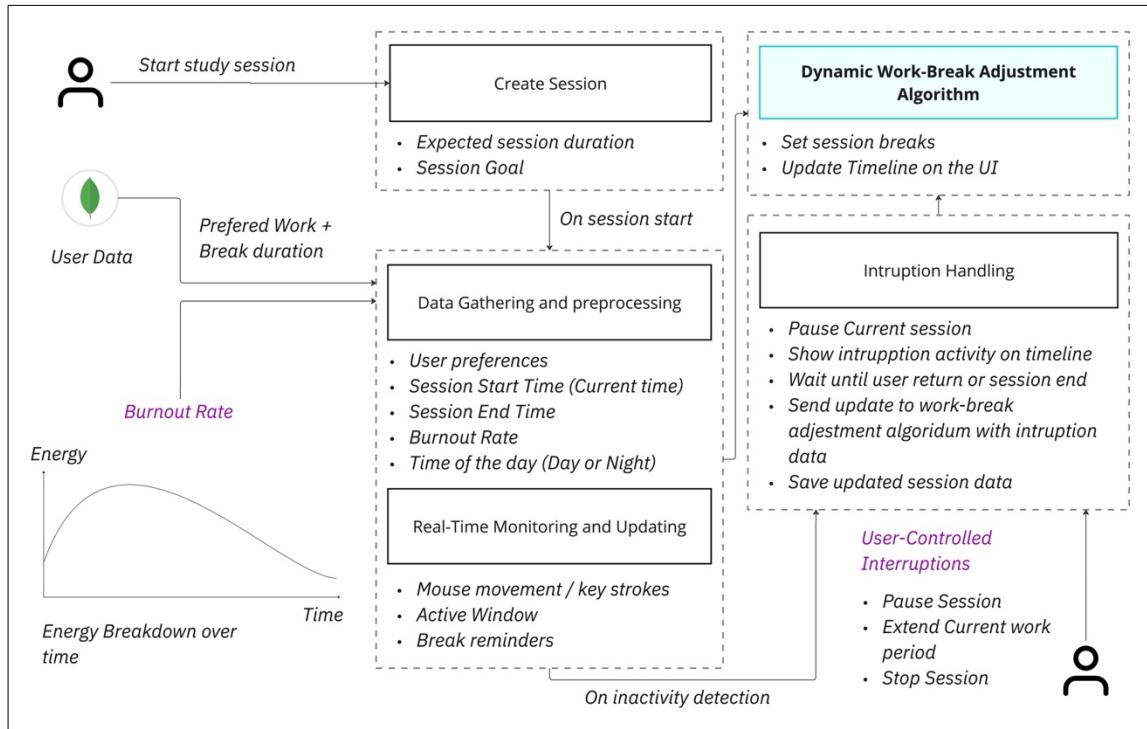


Figure 14 Component Diagrams - Study Techniques Component

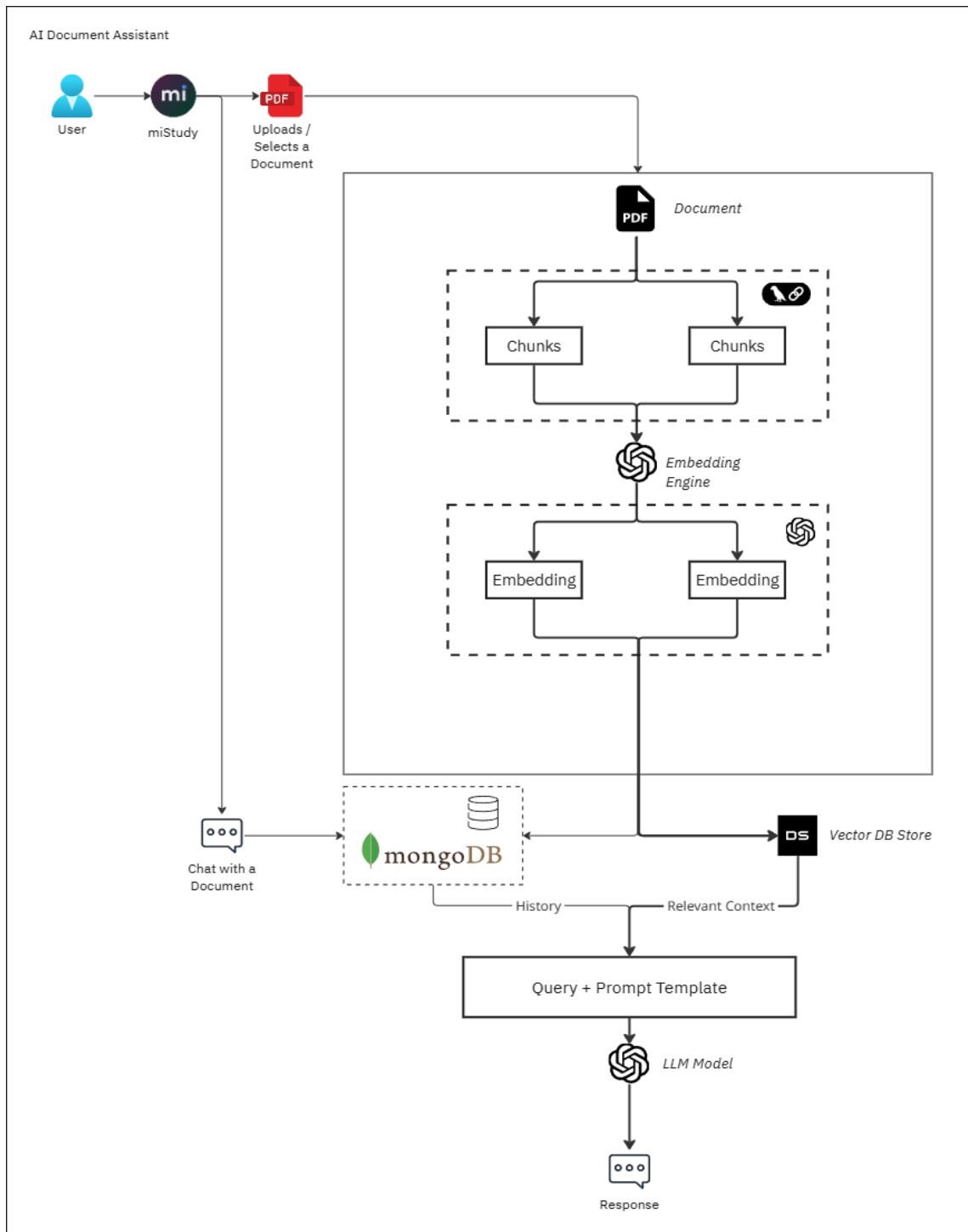


Figure 15 Component Diagrams - AI Document Assistant Component

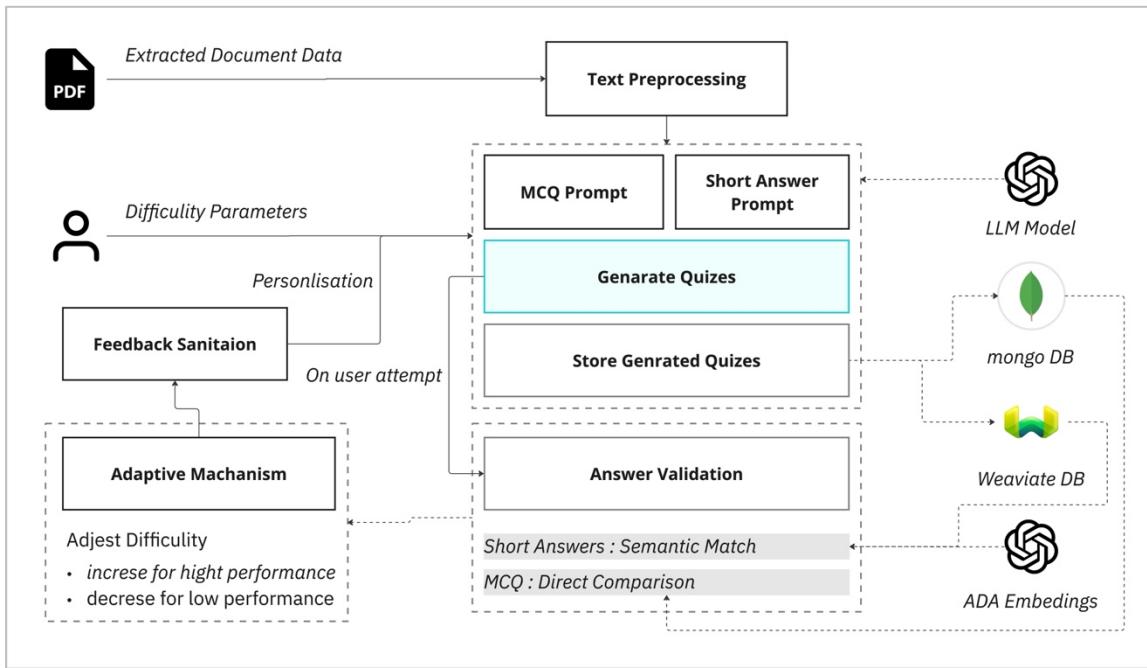


Figure 16 Component Diagrams - Quiz Component

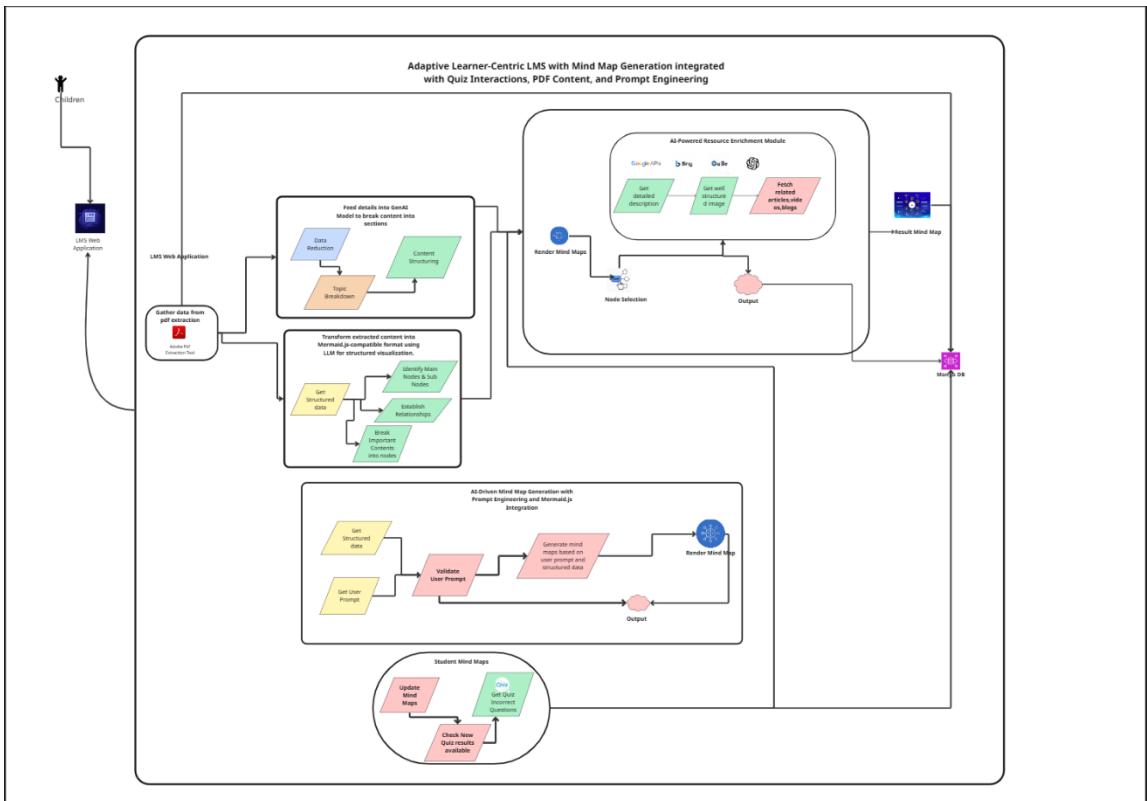


Figure 17 Component Diagrams - Mind-Map Component

3.11.4 Key Pillars of the Research Domain

3.11.4.1 Large Language Models (LLMs)

Large Language Models (LLMs) are at the core of the LMS's adaptive and personalized learning features. LLMs are pre-trained with vast amounts of textual data, enabling them to understand and generate human-like responses based on contextual input. These models have exceptional language generation capabilities and can be used for various tasks within the LMS, such as quiz personalization, mind map generation, and adaptive feedback.

The LLMs employed in this LMS will utilize in-context learning, where the models generate responses based on the context provided in user inputs, such as quiz performance data or specific learning resources. This allows the system to offer tailored content and feedback to students, improving their learning experience.

Popular LLMs that can be employed include GPT-4, GPT-3, BERT, and Cohere, with each model offering varying levels of complexity, adaptability, and performance. The choice of LLM will depend on performance needs, integration complexity, and cost per token for deployment in the LMS environment.

3.11.4.2 Frameworks to Handle LLM

To integrate Large Language Models (LLMs) effectively into the LMS, specialized frameworks and libraries are required. LangChain, Haystack, and LlamaIndex are the primary frameworks that will help in creating a seamless interaction between LLMs and the system. These frameworks facilitate the management of complex tasks such as retrieval-augmented generation (RAG), knowledge graph integration, and query-answering pipelines.

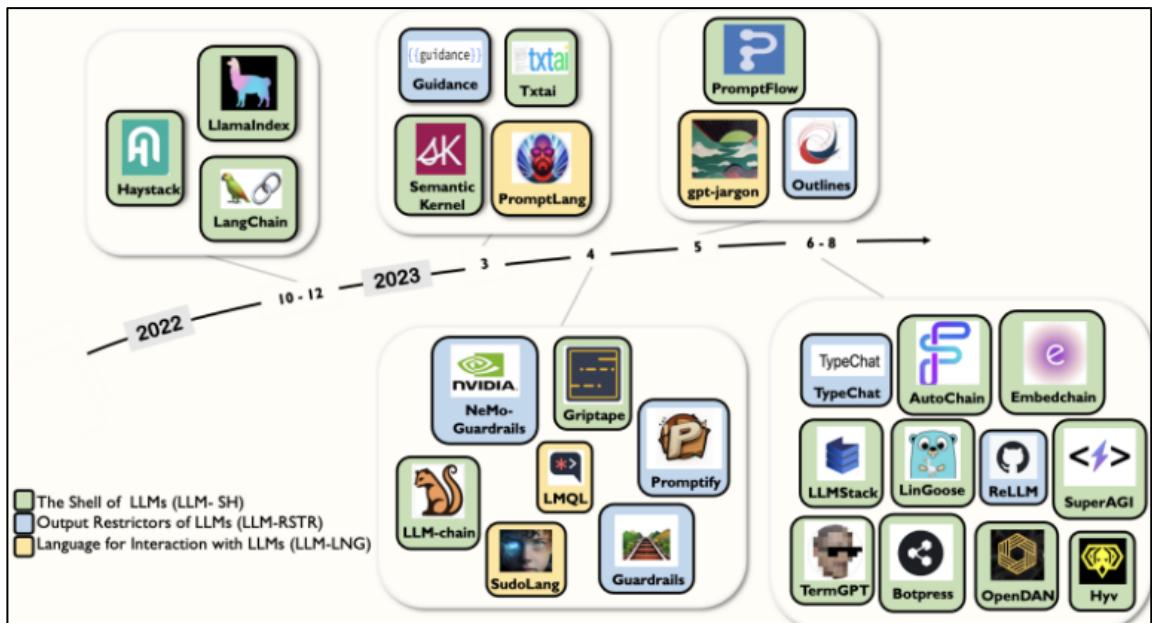


Figure 18 LLM Frameworks

LangChain is especially useful for building LLM applications with various components like APIs, databases, and documents, while Haystack focuses on providing a robust framework for document retrieval and question answering, which is key in providing personalized learning paths for students.

3.11.4.3 Prompt Engineering

Prompt Engineering is a critical technique for designing inputs to Large Language Models (LLMs) in a manner that maximizes the generation of relevant, accurate, and coherent outputs. In the context of the LMS, prompt engineering is applied to structure inputs in ways that enhance the quality of adaptive mind map generation, quiz feedback, question answering, and the creation of personalized study resources.

Prompting strategies are categorized into manual prompting, where specific instructions and templates are crafted by experts, and automated prompting, where the system dynamically adjusts prompts based on user interactions, learning performance, and behavioral patterns. The integration of both strategies ensures that the LLMs deliver context-aware and learner-centric outputs.



Figure 19 Prompt Engineering Techniques

It is important to utilize both categories of prompting to make sure the generative model learns the most important details in the given scenario.

3.11.4.4 Vector Stores for Enhanced Performance

In some scenarios, the standard LLM prompt may not provide sufficient context or depth for generating accurate responses. To address this, vector stores will be used to inject

additional knowledge into the model's context. These vector stores hold pre-processed knowledge representations (embeddings) of various resources, such as text documents, mind map nodes, and student performance data.

Popular vector store solutions include FAISS, Chroma, Weaveate and DeepLake, which allow for efficient retrieval of relevant data based on semantic similarity. These solutions will help enhance the adaptive learning environment by providing the LLM with more precise and relevant information for generating personalized learning paths.

3.11.4.5 Study Techniques and Cognitive Load Management

A foundational pillar of this research lies in the intelligent integration of study techniques and session management methodologies within a personalized digital learning environment. While most traditional LMS platforms focus on content distribution, this research emphasizes the behavioral and cognitive aspects of learning recognizing that *how* students study is as important as what they study.

At its core, this pillar synthesizes well-established study strategies Pomodoro Technique, Flowtime, Spaced Repetition, the Feynman Technique, and Mozart Effect-based stimulation into a cohesive module, powered by real-time behavioral tracking and adaptive reinforcement logic. These techniques were selected based on extensive literature review and validated through surveys across diverse student populations in Sri Lanka, indicating a high correlation between structured study sessions and increased academic engagement and retention.

The unique contribution of this pillar is not just embedding these techniques statically, but enabling their dynamic orchestration through a feedback-driven system. Learner actions such as mouse activity, active tab detection, session duration, and frequency of interruptions are continuously monitored and analyzed. Based on this data, the session

management engine automatically recalibrates break intervals, switches techniques, or triggers motivational nudges mimicking a human tutor's adaptive behavior.

Moreover, this component is tightly coupled with goal tracking, analytics dashboards, and visual feedback tools, offering the learner insights into their behavioral patterns. Performance is visualized through timeline graphs, focus pie charts, technique utilization heatmaps, and adaptive feedback alerts, ensuring transparency and self-awareness in the learning process.

This study techniques pillar serves as a behavioral augmentation layer that transforms the LMS from a static content manager into a smart productivity companion one that learns, adapts, and grows with the user. It represents a critical advancement in personalized learning science by merging proven educational psychology with modern software engineering, all deployed within a scalable, cloud-native infrastructure.

3.12 Implementation

3.12.1 Main Application UI and Flow

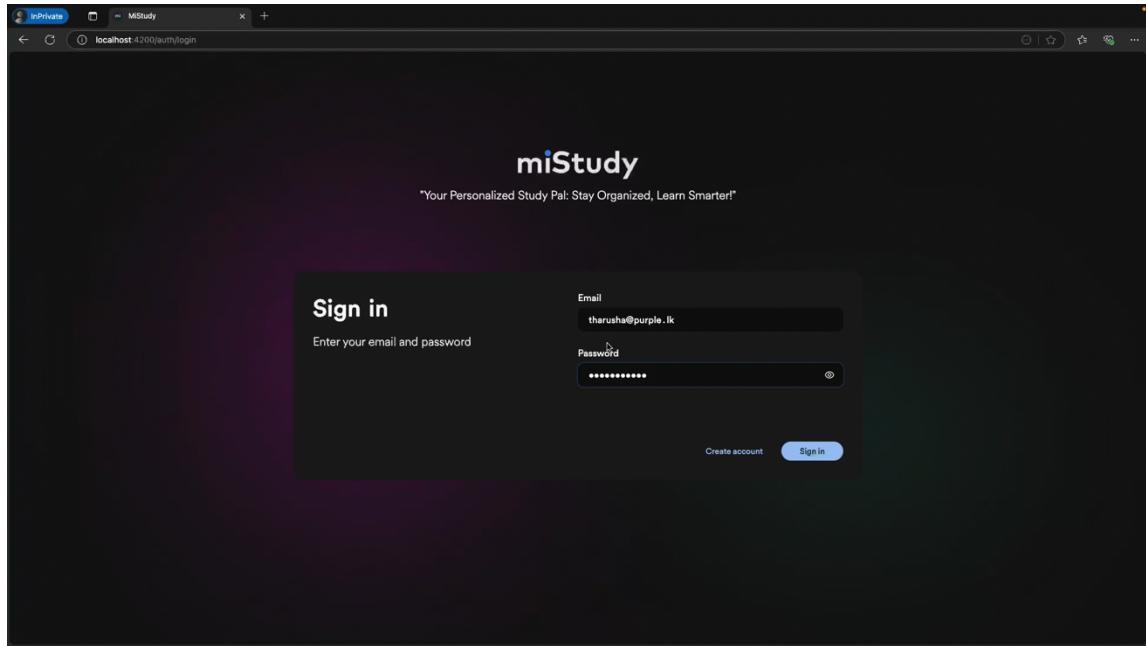


Figure 20 Main UI 01

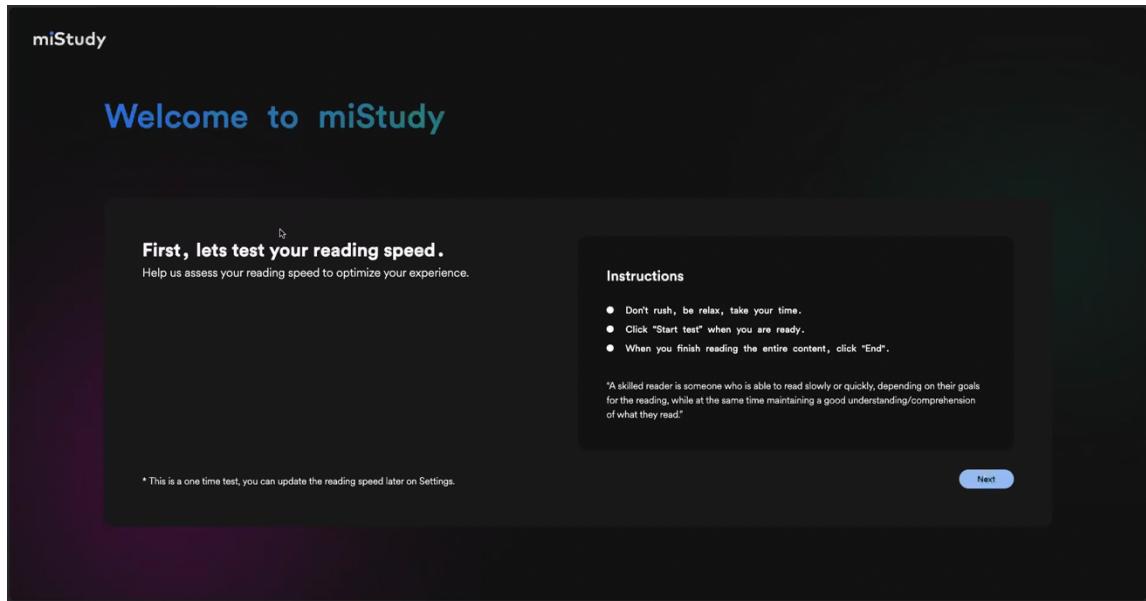


Figure 21 Main UI 02

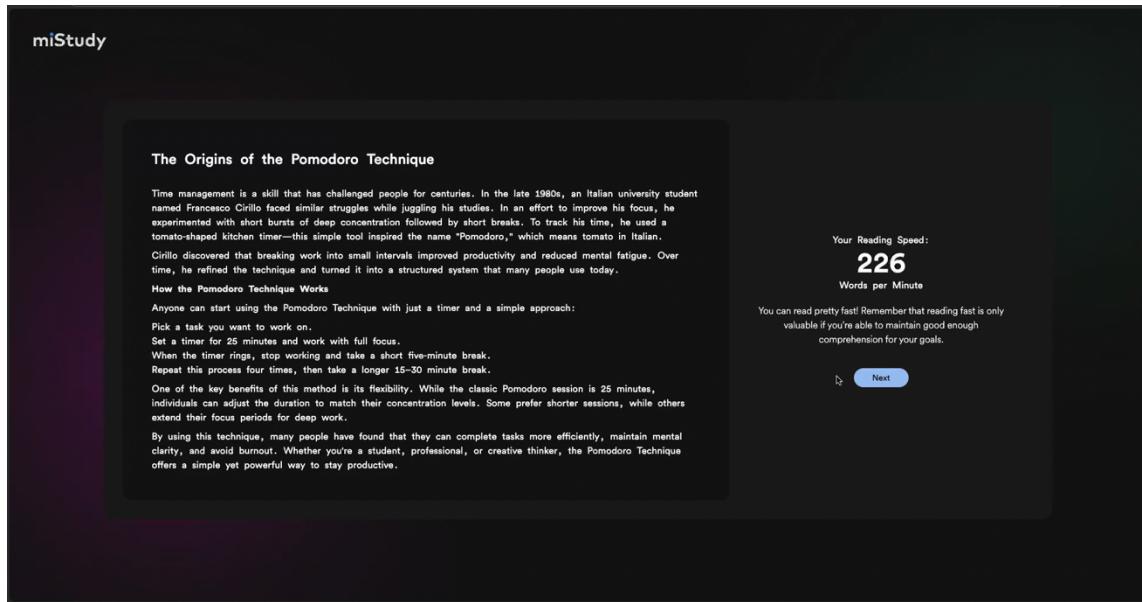


Figure 22 Main UI 03

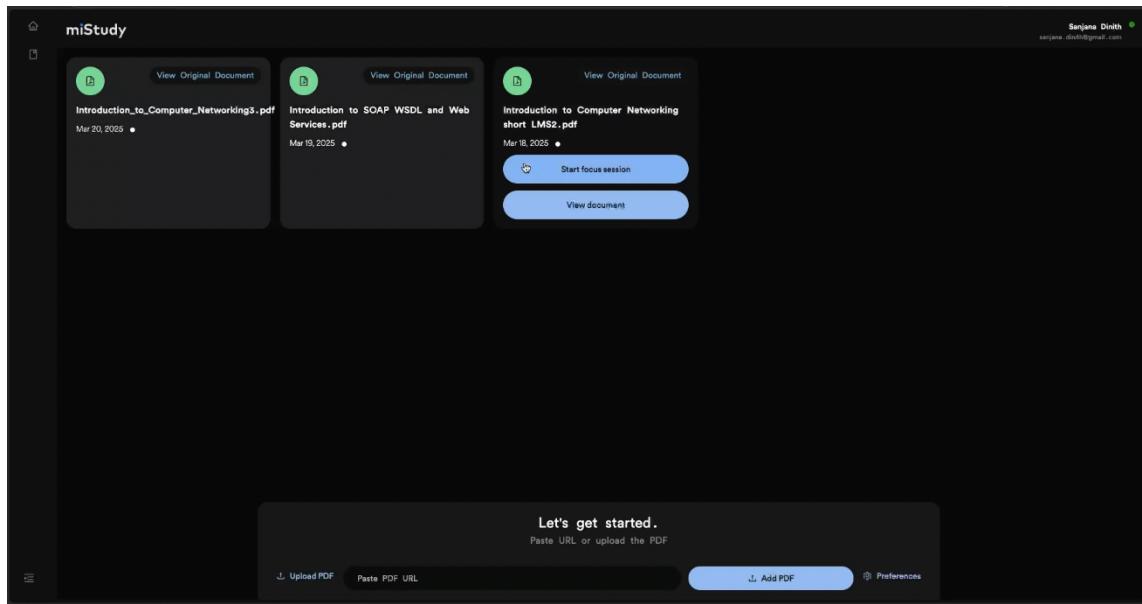


Figure 23 Main UI 04

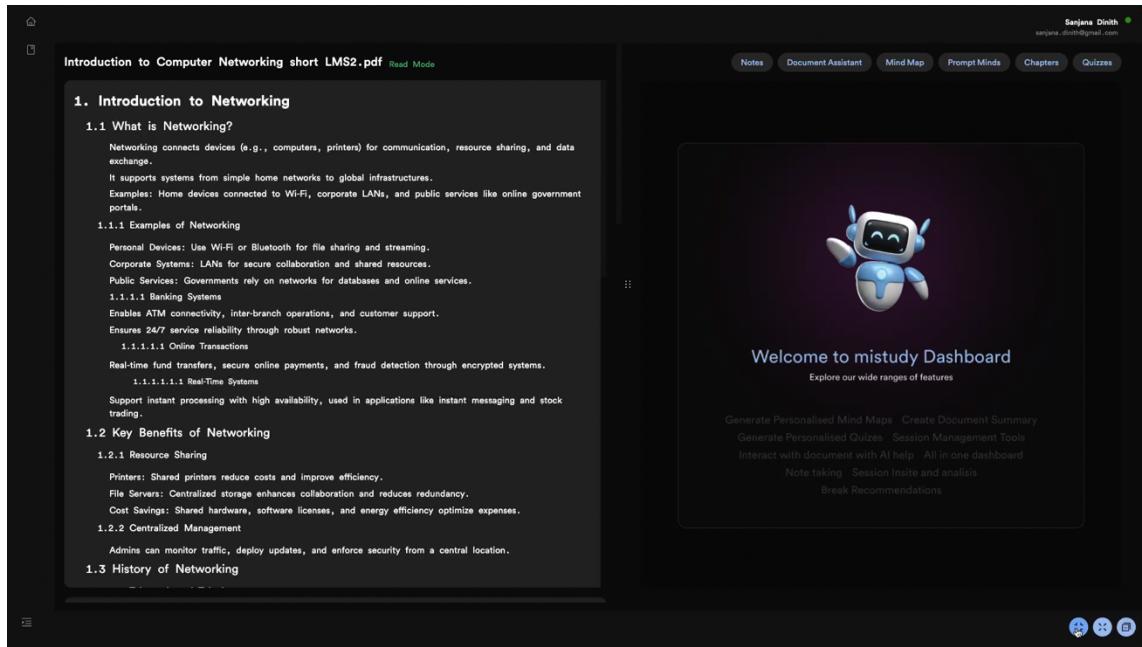


Figure 24 Main UI 05

3.12.2 Adaptive Study Session Manager Implementation

3.12.2.1 Dynamic Session Break Scheduler

The Adaptive Study Session Manager's core functionality revolves around dynamically scheduling personalized break intervals that respond to both learner behavior and environmental noise levels. The system is developed in Angular and integrates with various services to calculate, visualize, and manage breaks in real time during a study session. The logic is executed through a modular algorithm that tailors focus and break durations based on cognitive depletion models and ambient noise conditions.

- A form-based interface enables users to define session start and end times, along with their focus capacity (i.e., how long they can stay concentrated) and preferred break durations.

- The algorithm simulates cognitive fatigue by decrementing an internal `energyLevel` variable (starting at 1.0) by 10% per focus cycle, with a floor value of 0.5, representing the minimum sustainable energy threshold.
- The break calculation logic subscribes to a shared observable from the `NoiseLevelService`, allowing the system to adapt break schedules dynamically based on real-time ambient noise levels.

The implementation is encapsulated within the `calculateProgressiveBreaks()` method, which is responsible for computing and storing these adaptive intervals. The code snippet below illustrates how work and break periods are generated using user preferences, energy decay modeling, and noise reactivity

```

● ● ●

calculateProgressiveBreaks(
  startTime: Date,
  endTime: Date,
  userPreferences: { focusCapacity: number; preferredBreak: number }
): void {
  this.session.breaks = [];
  this.sessionForm.setControl("breakIntervals", this.fb.array([]));
  let currentTime = new Date(startTime);
  let { focusCapacity, preferredBreak } = userPreferences;
  let energyLevel = 1.0;

  this.noiseLevelService.averageNoiseLevel$.subscribe((level) => {
    this.averageNoiseLevel = level || 25;
  });

  while (currentTime.getTime() < endTime.getTime()) {
    // Adaptive logic: noise-aware adjustment
    let adjustedFocus = this.getAdjustedFocus(focusCapacity, this.averageNoiseLevel);
    let adjustedBreak = this.getAdjustedBreak(preferredBreak, this.averageNoiseLevel);

    // Time calculations based on energy
    const workTime = Math.max(adjustedFocus * energyLevel, 15);
    const breakTime = Math.min(adjustedBreak / energyLevel, 30);

    const workEnd = new Date(currentTime.getTime() + workTime * 60000);
    if (workEnd.getTime() > endTime.getTime()) break;

    const breakStart = workEnd;
    const breakEnd = new Date(breakStart.getTime() + breakTime * 60000);
    if (breakEnd.getTime() > endTime.getTime()) break;

    this.session.breaks.push({ start: breakStart.toISOString(), end: breakEnd.toISOString() });
    (this.sessionForm.get("breakIntervals") as FormArray).push(
      this.fb.group({
        start: [breakStart.toISOString(), Validators.required],
        end: [breakEnd.toISOString(), Validators.required],
      })
    );

    currentTime = breakEnd;
    energyLevel = Math.max(energyLevel - 0.1, 0.5);
  }

  this.updateSession();
  this.displayNextBreak();
}

```

Figure 25 Dynamic Session Break Scheduler

3.12.2.2 Context-Aware Break Calculation

Using a progressive time loop, the algorithm calculates work-break cycles that fit within the user's defined session window. It adaptively adjusts work duration and break length depending on:

- The average noise level (categorized as low <30 dB, moderate 30–50 dB, and high >50 dB).
- The learner’s current energy level, simulating cognitive resource depletion.
- User preferences for how long they can focus at a time and how long their ideal breaks should be.

For example:

- In high-noise conditions, the system may shorten work intervals by up to 20% and extend break times by up to 20%.
- In moderate conditions, these values are adjusted by 10%.
- In low-noise scenarios, the original user-specified durations are retained.

Each break interval is rendered both programmatically and visually, added to a FormArray representing the session plan. The updated break list is saved into the session object and visualized in the dashboard.

3.12.2.3 Visualization and Real-Time Updates

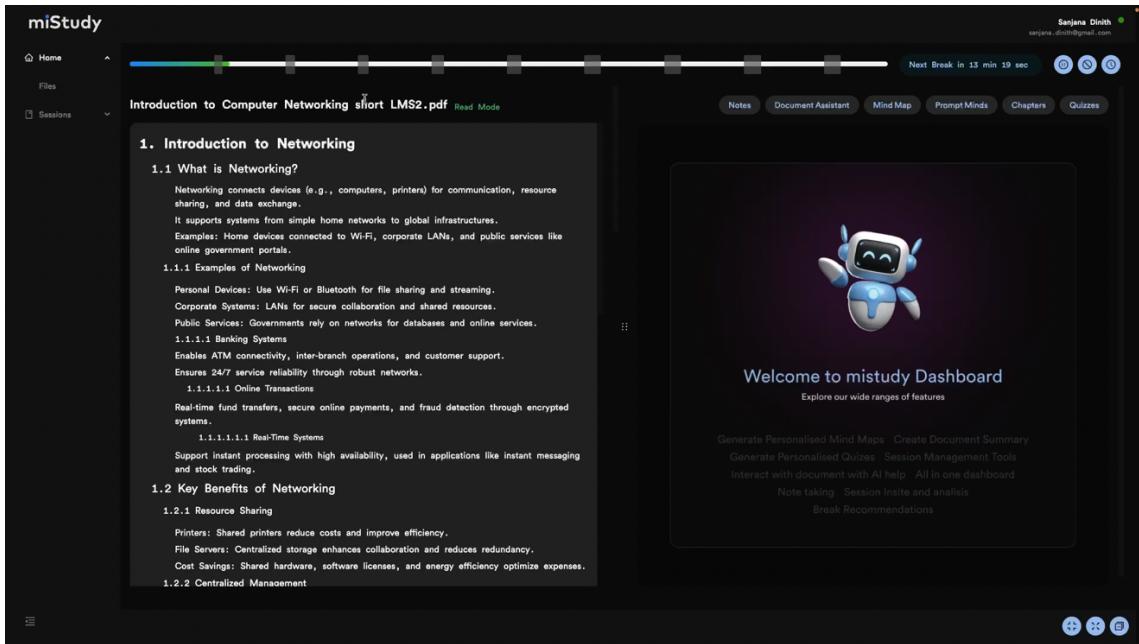


Figure 26 Session Timeline

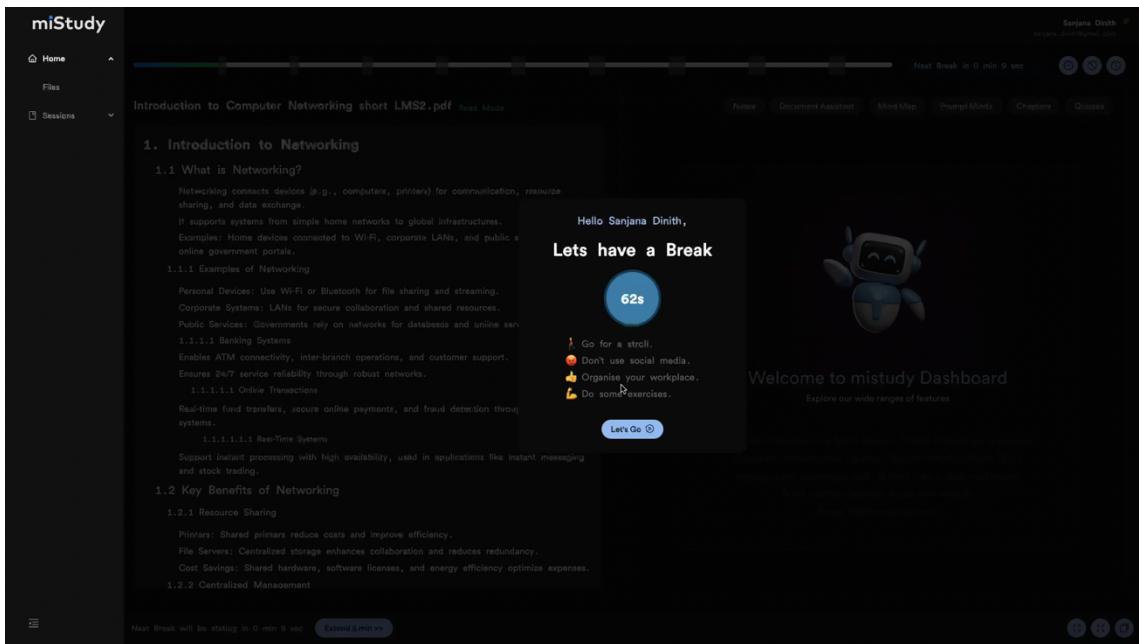


Figure 27 Session Break Popup

To improve user engagement and track focus cycles, a live **session dashboard** was implemented using Angular's reactive forms and conditionally rendered charts. Breaks are displayed chronologically, with timestamps and durations. Color-coded chips differentiate between focus periods and rest intervals.

Additionally:

- A session progress bar visually represents the total elapsed and remaining time.
- Toast notifications alert users of upcoming breaks or when energy drops significantly.
- A final check determines if unused time remains at the end of the session, prompting a longer break or final review suggestion.

3.12.3 AI Document Assistant Implementation

3.12.3.1 RAG Implementation

The initial step in implementing RAG for the AI Document Assistant focuses on processing student-uploaded PDF documents to extract text content, a prerequisite for retrieval and generation. LangChainJS's PDFLoader class was utilized to parse PDFs and extract text, enabling the system to work with diverse educational materials within miStudy's modular architecture.

- The PDF Loader class from LanchainJs was used to load and extract text from PDF documents.
- Large documents were split into chunks using LangChainJS's RecursiveCharacterTextSplitter ensuring manageable units for RAG retrieval while maintaining context.

3.12.3.2 Integration with AstraDB

The next step integrates the extracted and chunked text with AstraDB, implementing an embedding model to convert chunks into vectors and storing them for efficient retrieval. LangChainJS's interfaces facilitate this process, connecting OpenAI's embedding capabilities with AstraDB's vector storage within the TypeScript/Node.js environment.



```
1 const embeddings = new OpenAIEmbeddings({
2   model: process.env.EMBEDDING_MODEL,
3   openAIapiKey: process.env.OPENAI_API_KEY!,
4 });
5
6 const astraConfig: AstraLibArgs = {
7   token: process.env.ASTRA_DB_TOKEN!,
8   endpoint: process.env.ASTRA_DB_ENDPOINT!,
9   collection: process.env.ASTRA_DB_COLLECTION!,
10  collectionOptions: {
11    vector: {
12      dimension: 1536,
13      metric: "cosine",
14    },
15    indexing: {
16      allow: ["userId", "documentId"],
17    }
18  }
19 }
20 let astraDbVectorStore: AstraDBVectorStore | null = null;
21
22 export const connectToAstraDB = async (): Promise<void> => {
23   try {
24     astraDbVectorStore = await AstraDBVectorStore.fromExistingIndex(embeddings, astraConfig);
25     logger.info("Connected to AstraDB.");
26   } catch (error) {
27     logger.error(`Error connecting to AstraDB: ${error}`);
28     throw new Error("Unable to initialize vector store.");
29   }
30 };
31 export const getVectorStoreInstance = (): AstraDBVectorStore => {
32   if (!astradbVectorStore) {
33     throw new Error("Vector store is not initialized. Call connectToAstraDB first.");
34   }
35   return astraDbVectorStore;
36 };
37
38 export const initializeVectorStore = async (
39   documentCloudUrl: string,
40   userId: string,
41   documentId: string
42 ): Promise<void> => {
43   const existingRecords = await checkExistingRecords(userId, documentId);
44
45   if (existingRecords.length > 0) {
46     logger.info(`Vector store already initialized for userId: ${userId}, documentId: ${documentId}`);
47     return;
48   } else {
49     const pdfPath = await downloadFile(documentCloudUrl, documentId);
50     const extractedText = await extractTextFromPDF(pdfPath);
51     const documents = await createDocumentsFromText(extractedText, userId, documentId);
52     await AstraDBVectorStore.fromDocuments(documents, embeddings, astraConfig);
53
54     logger.info(`Initialized vector store for userId: ${userId}, documentId: ${documentId}`);
55   }
56 };
57
58 const checkExistingRecords = async (userId: string, documentId: string): Promise<any[]> => {
59   try {
60     const vectorStore = getVectorStoreInstance();
61     return await vectorStore.similaritySearch("", 1, {userId, documentId});
62   } catch (error) {
63     logger.info(`Checking existing records: ${error}`);
64     return [];
65   }
66 };
67 };
```

Figure 28 AstraDB Implementation

- The Embedding Model, text-embedding-3-small, processed the chunked document objects from PDFLoader passing them through OpenAIEMBEDDINGS. Each chunk is embedded into a 1536-dimensional vector, capturing semantic meaning.
- An AstraDB collection was created via the Astra dashboard and the Embedded vectors are stored using AstraDB interface provided by LangChain with userId and documentId as metadata.

3.12.3.3 Integration with GPT-4o-mini

The final step integrates GPT-4o-mini, OpenAI's lightweight model, with the RAG pipeline, performing similarity searches using LangChainJS's AstraDB interface to retrieve relevant content and generate responses. A similar search of the query is performed against the document's vector store. This retrieves the most relevant chunks from AstraDB, this content is concatenated into a string and fed into the model along with the Prompt Template.

```
1 const chatOpenAIModel = new ChatOpenAI({
2   apiKey: process.env.OPENAI_API_KEY,
3   model: process.env.OPENAI_MODEL,
4 });
5
6 export const queryLLM = async (text: string): Promise<string> => {
7   try {
8     const prompt = ChatPromptTemplate.fromMessages([
9       new SystemMessage({content: OPENAI_PROMPT}),
10      new HumanMessage({content: text}),
11    ]);
12
13    const promptText = await prompt.format({text});
14
15    const response = await chatOpenAIModel.invoke([new HumanMessage({content: promptText})]);
16
17    return response.content.toString();
18  } catch (error) {
19    throw new Error("LLM query failed.");
20  }
21};
22
23
24 export const moderateContent = async (
25   question: string
26 ): Promise<{
27   safe: boolean;
28   categories?: any;
29 }> => {
30   try {
31     const moderationModel = new OpenAIModerationChain({
32       apiKey: process.env.OPENAI_API_KEY,
33     });
34     const {results} = await moderationModel.invoke({
35       input: question,
36     });
37     const categoryScores = results[0].category_scores;
38     const threshold = 0.01;
39     const flagged = Object.values(categoryScores).some(
40       (score) => (score as number) >= threshold
41     );
42     if (flagged) {
43       return {safe: false};
44     }
45
46     return {safe: true};
47   } catch (error) {
48     throw new Error("Moderation API call failed.");
49   }
50 };
51
```

Figure 29 OpenAI Implementation

3.12.3.4 Prompt Design

Designing the prompts for the AI Document Assistant was an incremental process aimed at ensuring the system delivers intent-specific, pedagogically sound, and document-grounded responses within miStudy's framework. The process began with a zero-shot prompt to establish a baseline, leveraging the model's pre-trained capabilities, and evolved through iterations to address the identified intents Summarization, Code Snippets, Comparison, Quiz Generation, Simplification, and Explanation while integrating with Retrieval-Augmented Generation (RAG) and LangChainJS.

A zero-shot approach was chosen initially to test GPT-4o-mini's inherent understanding without examples, minimizing token usage and aligning with LangChainJS's documentation for efficient prompt design.

- Zero Shot Prompt**

The Prompt is designed to instruct the model to assume the role of a study assistant, aligning with miStudy's educational focus and setting a supportive tone for self-directed learning. It also contains instructions to restricts responses to the provided document's context, leveraging RAG's retrieved chunks to reduce hallucination and ensure relevance and to politely decline out-of-scope queries enhances user experience, maintaining a pedagogical demeanor. As a general prompt, it lacked specificity for handling diverse intents such as code blocks, quizzes, summaries etc. The solution to this limitation was to design an prompt that will identify the intent and use the respective prompt to produce a structured response.

- Intent Aware Structure Prompt**

To craft prompts for intent categorization, prevent out of scope responses and ensure pedagogical delivery, identifying common student intents was a foundational step. Through an analysis of educational query patterns from forums, the main intents identified were Summarization, Code Snippets, Quiz Generation, Comparisons, Simplification and Explanation. Each intent requires a tailored prompt to interpret the student's need via RAG

from uploaded documents and deliver answers clearly in a teaching-oriented manner. The prompt is a combination of chain of thoughts and few shot prompt engineering techniques.

3.12.4 Quiz Generation Component

3.12.4.1 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 30 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.

3.12.4.2 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 few-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.

```
1  export const bloomTaxonomyPrompt = ` 
2    ### ⚡ **Understanding Bloom's Taxonomy** ⚡
3
4    Bloom's Taxonomy categorizes cognitive skills into six levels:
5
6    [1] **Remembering (Basic Recall)**
7      - Recognizing and recalling facts, definitions, and concepts.
8      - Example: *"What is the chemical symbol for oxygen?"*
9
10   [2] **Understanding (Explain Concepts)**
11     - Interpreting and summarizing information.
12     - Example: *"Explain how photosynthesis works."*
13
14   [3] **Applying (Use Knowledge in a New Situation)**
15     - Solving problems using learned concepts.
16     - Example: *"Use Newton's laws to calculate force."*
17
18   [4] **Analyzing (Break Down Concepts & Compare)**
19     - Identifying relationships, patterns, and comparisons.
20     - Example: *"What are the key differences between prokaryotic and eukaryotic cells?"*
21
22   [5] **Evaluating (Make Judgments & Justify Opinions)**
23     - Assessing theories, arguments, or solutions.
24     - Example: *"Which economic policy is better for inflation control?"*
25
26   [6] **Creating (Generate New Ideas & Solutions)**
27     - Designing new approaches, models, or original work.
28     - Example: *"Develop a marketing strategy for a new product."*
29
30   ---
31   **💡 The AI must use this taxonomy while generating questions.**
32
33`;
```

Figure 31 Bloom's Taxonomy Integration

3.12.4.3 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 32 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.

Feedback:
Quiz Performance Analysis:

Congratulations on completing the quiz! You scored 6 out of 10, which shows that you have a good foundational understanding of networking concepts. However, there are areas for improvement based on your incorrect answers.

Strengths:

- You were able to correctly identify several concepts related to networking, suggesting you grasp basic networking terminology and its applications.

Weaknesses:

- There seems to be a misunderstanding of some fundamental definitions and key benefits of networking. Your answers indicate potential confusion between different types of technology and concepts associated with networking.

Common Mistakes Identified:

1. **Misunderstanding Definitions:** For example, you defined networking as a type of software, which reflects a limited understanding of its broader purpose—connecting devices for communication and resource sharing.
2. **Incorrect Perception of Benefits:** Confusing decreased efficiency as a benefit instead of recognizing improved resource sharing indicates a misunderstanding of the fundamental advantages that networking provides.
3. **Historical Context:** You mentioned the invention of the computer as an early innovation in networking, while the correct answer referred to Morse Code and switching stations, reflecting a need to revisit the historical timeline of networking technologies.

Improvement Suggestions:

1. **Study Definitions:** Focus on the foundational definitions of key networking terms. Resources such as textbooks, reputable websites, or online courses can provide valuable context and clarity.
2. **Review Key Benefits:** Familiarize yourself with the advantages of networking, such as resource sharing, cost reduction, and improved communication. Make a list of common benefits and their explanations.
3. **Understand Historical Developments:** Research the history of networking technologies. Knowing key milestones, like the invention of Morse Code and switching stations, will help you appreciate how networking has evolved.
4. **Practice with Sample Questions:** Engage in practice quizzes or flashcards that specifically address networking concepts to strengthen your recall and understanding.

Actionable Tips for Future Quizzes:

- **Create a Study Plan:** Dedicate time each week to review networking concepts and test your knowledge with quizzes that focus on both terminology and practical applications.
- **Join Study Groups:** Collaborating with peers can clarify misunderstandings and provide different perspectives on complex concepts.
- **Utilize Multimedia Resources:** Videos, podcasts, and infographics can make learning about networking more engaging and memorable.

With focused study and practice, you will undoubtedly improve your understanding of networking. Keep up the good work, and best of luck on your next quiz!

Figure 33 Feedback Display

3.12.4.4 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 Remembering, 3 questions from Applying
    61-80 (Hard): 2 questions from Remembering, 4 questions from Applying / Understanding, 3 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 34 MCQ Prompt

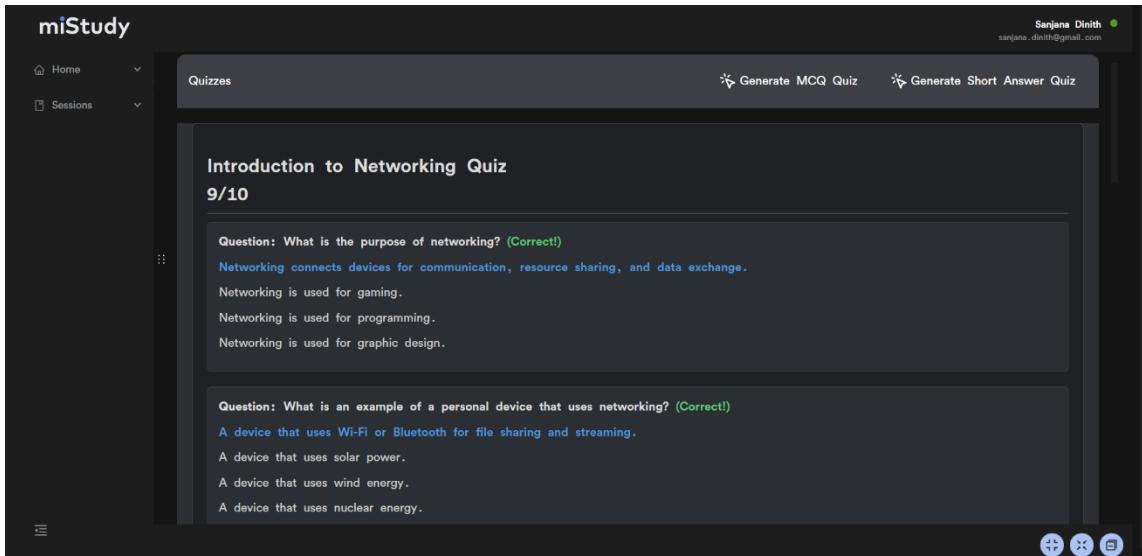


Figure 35 MCQ UI

```

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)**:
    - 3 questions from **Understanding**
    - 3 questions from **Applying**
    - 3 questions from **Analyzing**
    - 3 questions from **Evaluating**

  - **81-100 (Very Hard)**:
    - 3 questions from **Applying**
    - 3 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 36 SAQ Prompt

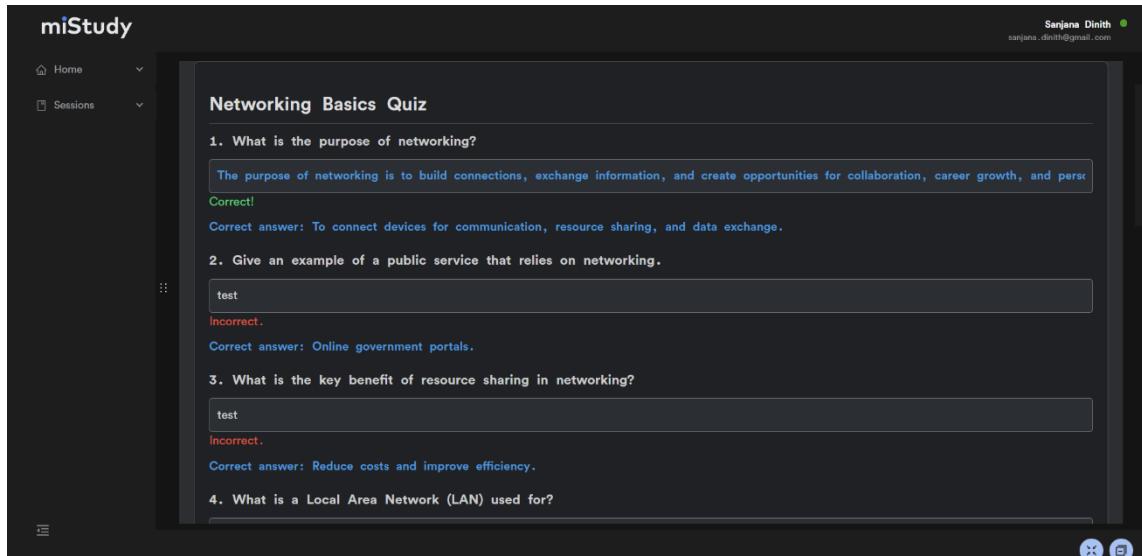


Figure 37 SAQ UI

3.12.4.5 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.

```

_id: ObjectId('67db5ca13450484c69869006')
title : "Introduction to Networking Quiz"
documentID : ObjectId('67db27e380434636f0b7ff64')
userID : ObjectId('6741df922c3f8d8d9bc6ed02')
difficultyLevel : 50
questions : Array (10)
score : 3
createdAt : 2025-03-20T00:09:05.469+00:00
updatedAt : 2025-03-20T00:10:10.698+00:00
__v : 0

_id: ObjectId('67db5d123450484c698691da')
title : "Introduction to Networking"
documentID : ObjectId('67db27e380434636f0b7ff64')
userID : ObjectId('6741df922c3f8d8d9bc6ed02')
difficultyLevel : 30
questions : Array (10)
score : 0
createdAt : 2025-03-20T00:10:58.503+00:00
updatedAt : 2025-03-20T00:10:58.503+00:00
__v : 0

```

Defult Difficulty Level

Figure 38 Difficulty Level Adjustments

3.12.5 Adaptive Mind Map Component

The mind map generation component is driven by an AI model that analyzes quiz performance and learning behavior to identify students' weak points. Based on this data, the system automatically creates personalized mind maps and provides external learning resources such as summaries, links, and definitions. Students also have the option to input custom prompts, which the system uses to generate mind maps tailored to the topic of their choice.

The goal is to provide a more adaptive, visual, and efficient learning experience by integrating advanced AI features with modern web technologies. Students can explore study material, improve weak areas, and receive on-demand support all through a smart, interactive web interface.

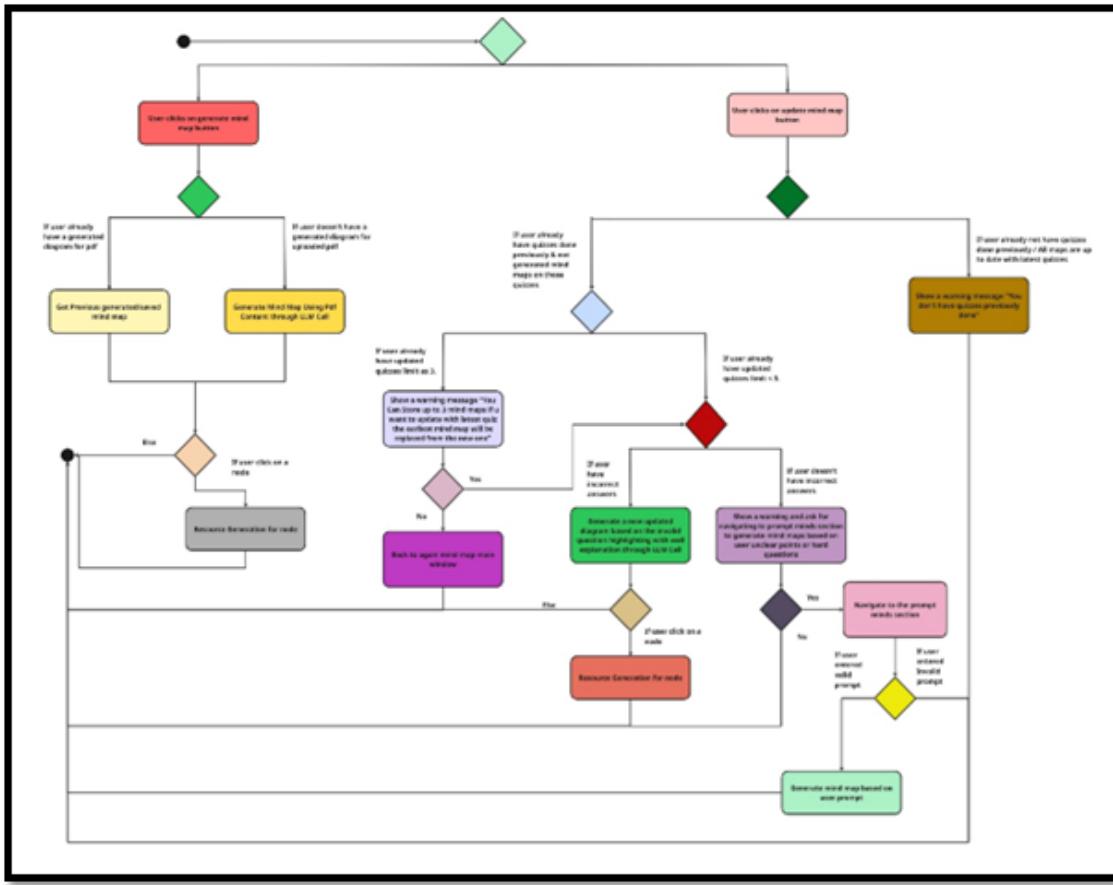


Figure 39 Adaptive Mind-Map Generation Component Implementation Sequence

3.12.5.1 Selection of research domains

After thorough consideration, GPT-4o-Mini as the large language model, was the best choice to be utilized.

```

434     const payload = {
435       model: "gpt-4o-mini",
436       messages: [{ role: "user", content: prompt }],
437       max_tokens: 2000,
438       temperature: 0.1,
439     };
440
441     const headers = {
442       "Content-Type": "application/json",
443       Authorization: `Bearer ${OPENAI_API_KEY}`,
444     };

```

Figure 40 Model Selection for Mind-Map

- One of the latest versions of large language models.
- Highly cost effective.

Prompt engineering practices will be used accordingly along with the GPT-4o-Mini to produce the most reliable and accurate output based on the requirement of the adaptive mind map generation.

3.12.5.2 Prompt engineering to generate structured version of pdf

To structure the extracted PDF content into a hierarchical JSON format, instruction-based prompting was applied as the core prompt engineering technique. A detailed and structured prompt was designed, explicitly guiding the model to group content under headings such as H1 to H6 and related paragraphs using semantic context and page numbers. The prompt also defined the exact output format to ensure database compatibility. This form of prompting allows for precise control over the model's behavior and output structure. Additionally, the integration with the Adobe PDF Extraction Tool enabled the extraction of raw content, which was then processed by the prompt to generate a meaningful and organized representation. This technique ensures consistency, accuracy, and a high level of customization when handling diverse and unstructured PDF documents.

```

const prompt = "
I have the following extracted content from a PDF. Each element contains a text field, path, and a page number.
Group the content into a hierarchical JSON structure with H1, H2, H3, H4, H5, H6 headings, and paragraphs.
Keep related text under the appropriate headings based on semantic context.
For each section, use the following format:
{
  \"Sections\": [
    {
      \"H1\": \"Chapter 1\",
      \"Content\": [
        \"Introduction to the topic.\"
      ],
      \"SubSections\": [
        {
          \"H2\": \"Section 1.1\",
          \"Content\": [
            \"Details about section 1.1.\"
          ],
          \"SubSections\": [
            {
              \"H3\": \"Subsection 1.1.1\",
              \"Content\": [
                \"Details about subsection 1.1.1.\"
              ]
            }
          ]
        },
        {
          \"H2\": \"Section 1.2\",
          \"Content\": [
            \"Details about section 1.2.\"
          ]
        }
      ]
    },
    {
      \"H1\": \"Chapter 2\",
      \"Content\": [
        \"Introduction to chapter 2.\"
      ],
      \"SubSections\": [
        {
          \"H2\": \"Section 2.1\",
          \"Content\": [
            \"Details about section 2.1.\"
          ],
          \"SubSections\": [
            {
              \"H3\": \"Subsection 2.1.1\",
              \"Content\": [
                \"Details about subsection 2.1.1.\"
              ],
              \"SubSections\": [
                {
                  \"H4\": \"Sub-subsection 2.1.1.1\",
                  \"Content\": [
                    \"Details about sub-subsection 2.1.1.1.\"
                  ]
                }
              ]
            },
            {
              \"H2\": \"Section 2.2\",
              \"Content\": [
                \"Details about section 2.2.\"
              ]
            }
          ]
        }
      ]
    }
  ]
}

Please analyze the page numbers and identify the sequential order.
Only include valid elements where text is available. Ensure that headings and paragraphs are well grouped by their context.
Return the JSON structure only, with no additional text. The response should be in a format that can be saved directly into the database.

Here is the extracted data:
${JSON.stringify(filteredChunk)}
";

```

Figure 41 Prompt – PDF Extraction

3.12.5.3 Prompt engineering to generate adaptive mind maps

3.12.5.3.1 Pdf based main mind map generation

To generate the main diagram based on the PDF content, the previously structured data is passed to the model, where the content is extracted into nodes and sub-nodes for mind map generation. (output will be further explained in Results & Discussion)

```
const MAIN_MIND_MAP_GENERATOR_PROMPT = ` You are a mind map generator. Convert the following JSON into Mermaid.js mind map input format.
For each object or array, create corresponding nodes and sub-nodes.

1. If there are long paragraphs inside the "paragraph" tags, split them into simple sentences for better readability.
2. Remove any unwanted characters like special symbols or redundant information.
3. For each paragraph, ensure that it's concise and doesn't exceed a few sentences, keeping it readable.
4. Maintain the hierarchical structure from the JSON as nodes and sub-nodes in the mind map.

This is the JSON data you need to convert to mermaid below format:      JSON: ${JSON.stringify(extractedSections)}

Below I'll provide a template how to break the things into nodes.Use this template to generate the mermaid js input format.Don't add any of your thoughts to this.
return the template only.Don't add any text or any special characters to it.Just give me the output like below format.
The format is below.Use the correct icons based on the texts.for icons only you have to use the brackets like in below format.

mindmap
root("Artificial Intelligence")
What is AI?
  :icon(fa fa-users)
  ("Artificial Intelligence refers to the simulation of human intelligence.")
  ("AI is programmed to think like humans and mimic their actions.")
  ("The goal of AI is to enable machines to perform tasks.")
  ("Tasks include visual perception, speech recognition, decision-making, and language translation.")
Applications of AI
  :icon(fa fa-hospital)
  ("AI is widely used in various industries.")
  ("In healthcare, AI helps diagnose diseases and predict patient outcomes.")
  ("It also helps develop personalized treatment plans.")
  ("In finance, AI is used for fraud detection.")
  ("AI also supports algorithmic trading and customer service automation.")

Do not include any brackets inside parentheses in the formatted output. This means that words or phrases within parentheses, such as (LAN), should be removed while keeping the rest of the text intact.

For example, if the original mind map node is:**
(Local Area Network (LAN) connects devices within a limited area.)

The formatted output should be:
(Local Area Network connects devices within a limited area)

Only one root, use free FontAwesome icons, and follow node types "(", "(". No need to use "mermaid", "\`\\``", or "graph TD". Respond only with code and syntax.`;
```

Figure 42 Prompt - Mind-Map Generation

This prompt uses instruction-based prompting combined with few-shot prompting. A detailed instruction set guides the model's behavior, while an example output acts as a reference (few-shot) to help the model understand the expected structure and formatting of the Mermaid.js mind map.

3.12.5.3.2 Quiz based adaptive mind map generation

In this technique, zero-shot prompting is employed to generate mind maps from quiz-related content. The model is provided with a set of instructions without any examples of

the expected input-output pairs. This enables the model to handle unseen data, such as incorrect answers from quizzes, and generate corresponding mind maps.

The instructions provided guide the model on how to process quiz data and transform it into a structured mind map format using Mermaid.js. This includes splitting long paragraphs into simple sentences for clarity, removing unwanted characters, and ensuring readability. The hierarchical structure of the data is maintained, and additional explanations are given for incorrect answers.

```

const QUIZ_BASED_MIND_MAP_PROMPT = `You are a mind map generator. Convert the following highlighted data into Mermaid.js mind map input format.
For each object or array, create corresponding nodes and sub-nodes.

1. If there are long paragraphs inside the "paragraph" tags, split them into simple sentences for better readability.
2. Remove any unwanted characters like special symbols or redundant information.
3. For each paragraph, ensure that it's concise and doesn't exceed a few sentences, keeping it readable.
4. Maintain the hierarchical structure from the JSON as nodes and sub-nodes in the mind map.

This is the data you need to more highlight the things regarding to these topics : JSON: ${JSON.stringify(incorrectAnswersForQuizzes)}

For highlighted topics please provide some additional explanations.

For incorrect answers you can provide any additional data regarding to the concepts

Below I'll provide a template how to break the things into nodes.Use this template to generate the ermaid js input format.Don't add any of your thoughts to this.
return the template only.Don't add any special characters to it.Just give me the output like below format.
The format is below.Use the correct icons based on the texts.for icons only you have to use the brackets like in below format.

mindmap
root("Artificial Intelligence")
What is AI?
:icon(fa-users)
("Artificial Intelligence refers to the simulation of human intelligence.")
("AI is programmed to think like humans and mimic their actions.")
("The goal of AI is to enable machines to perform tasks.")
("Tasks include visual perception, speech recognition, decision-making, and language translation.")
Applications of AI
:icon(fa-play)
("AI is widely used in various industries.")
("In healthcare, AI helps diagnose diseases and predict patient outcomes.")
("It also helps develop personalized treatment plans.")
("In finance, AI is used for fraud detection.")
("AI also supports algorithmic trading and customer service automation.")

Do not include any brackets inside parentheses in the formatted output. This means that words or phrases within parentheses, such as (LAN), should be removed while keeping the rest of the text intact.

For example, if the original mind map node is:**
(Local Area Network (LAN) connects devices within a limited area.)

The formatted output should be:
(Local Area Network connects devices within a limited area.)
Only one root; use free FontAwesome icons, and follow node types "[", "(". No need to use "mermaid", "\\", or "graph TD". Respond only with code and syntax.`;

```

Figure 43 Prompt - Quiz based Mind-Map Generation

Since no specific examples are provided, the model is expected to generalize based on its prior knowledge and generate mind maps for the quiz answers provided. Figure 4.2 shows zero-shot prompting integrated with instruction-based prompting, where the model applies its learned understanding to an entirely new task, handling data it has not encountered before.

3.12.5.4 Prompt based adaptive mind map generation

3.12.5.4.1 User Prompt Validation

The QUESTION_VALIDATION prompt employs a combination of instruction-based prompting, role prompting, and chain of thought prompting to guide the model in responding to user queries based on the content of a provided PDF.

```
export const MIND_MAP_PROMPTS = {
  QUESTION_VALIDATION: `You are a study assistant helping with questions and creating mind maps.
  If the user asks a question unrelated to any learning environment (e.g., general knowledge or unrelated topics), you should not create the mind maps.

  You should respond if the user's question is based on the content of the provided PDF.
  Study Assistant Guidelines:

  Topic Relevance (PDF Content):
  If the user's question relates to the content of the provided PDF, you must answer the question, even if the exact details are not explicitly mentioned in the PDF.
  For example, if the PDF is about "software security aspects" and the user asks, "What are the advantages of considering software security aspects in an application?" even if the specific advantages are not in the PDF, your answer should still be based on the overall topic of software security aspects.
  Answering Based on Provided PDF:

  You should answer all questions related to the content of the PDF in a detailed and comprehensive manner. If the user asks for further clarification or details, such as asking to elaborate on specific topics like "security aspects" or requesting a breakdown of information, you should provide a thorough explanation.
  For example:
  If the question is, "I want to know more about security aspects in a detailed way," you should provide an in-depth explanation with ample descriptions and related information.
  If the question is, "Give me a summary of software aspects," break down the summary into its core components and explain each part comprehensively.
  Invalid Questions:
  If the user asks a question unrelated to the content of the provided PDF, treat it as invalid. This includes questions on general knowledge, personal preferences, or unrelated topics.
  For instance, questions like "What is the capital of France?" or "How do you cook spaghetti?" should be marked as invalid.
  Response Handling:
  If the user's question is valid (i.e., related to the content of the PDF or a related topic), respond with "Valid."
  If the user's question is invalid (i.e., not related to the content of the PDF or a relevant topic), respond with "Invalid."`}
```

Figure 44 Prompt – Mind-Map Question Validation Generation

Instruction-Based Prompting provides the model with clear, structured instructions on how to handle different types of questions. The model is instructed to answer only those questions that are relevant to the content of the PDF and to give comprehensive explanations when necessary.

Role Prompting defines the model's role as a "study assistant," which helps in determining the correct behavior. The model's task is to generate answers related to the PDF's content and provide mind maps when appropriate. If the question is unrelated to the learning material, the model is instructed not to generate a mind map.

Chain of Thought Prompting helps the model process and logically reason through the steps involved in determining whether a question is valid or invalid. This ensures that the model considers the context of the query and the PDF content before deciding on the response.

This combination of techniques ensures that the model accurately filters valid queries, responds based on the provided material, and avoids generating responses for irrelevant or off-topic questions.

3.12.5.4.2 Prompt based mind map template

The SIMPLE_MINDMAP_GENERATOR_PROMPT utilizes a combination of few-shot prompting and instruction-based prompting to guide the model in generating mind maps based on user input.

```

SIMPLE_MINDMAP_GENERATOR_PROMPT :
Create a mermaid mindmap based on user input like these examples:
brainstorming mindmap
mindmap
|>root("leisure activities weekend")
|>root("spend time with friends")
|>root::icon(fa fa-users)
|>root::text("action activities")
|>root::icon(fa fa-play)
|>root::text("going to a night club")
|>root::text("go to the theater")
|>root::text("spend time your self")
|>root::icon(fa fa-user)
|>root::text("take a sunbath")
|>root::icon(fa fa-sun)
|>root::text("\u25bc take a sunbath \u25bc")
|>root::text("reading a book")
|>root::icon(fa fa-book)
text summary mindmap:
Barack Obama (born August 4, 1961) is an American politician who served as the 44th president of the United States from 2009 to 2017. A member of the Democratic Party, he was the first African-American president of the United States.
mindmap
|>root("Barack Obama")
|>root("Born August 4, 1961")
|>root::icon(fa fa-baby-carriage)
|>root::text("American Politician")
|>root::icon(fa fa-flag)
|>root::text("44th President of the United States")
|>root::text("Democrat Party")
|>root::icon(fa fa-democrat)
|>root::text("First African-American President")
cause and effects mindmap:
mindmap
|>root("Landlord sells apartment")
|>root::icon(fa fa-sell)
|>root::text("Renter must be notified of sale")
|>root::icon(fa fa-envelope)
|>root::text("Tenants may feel some uncertainty")
|>root::icon(fa fa-exclamation-circle)
|>root::text("Lease code must be observed")
|>root::icon(fa fa-calendar)
|>root::text("Landlord can submit notice of termination for personal use")
|>root::icon(fa fa-home)
|>root::text("Tenants has to look for a new apartment")
|>root::icon(fa fa-search)
|>root::text("New owner")
|>root::icon(fa fa-user)
|>root::text("New owner takes over existing rental agreement")
|>root::icon(fa fa-file-contract)
|>root::text("Tenant keeps previous apartment")
|>root::icon(fa fa-handshake)
|>root::text("New owner initiates newly concluded lease")
|>root::icon(fa fa-hands)
|>root::text("Tenant has to look for a new apartment")
|>root::icon(fa fa-search)
Only one root, use free FontAwesome icons, and follow node types "[", ". No need to use "mermaid", "\\"\\\", or "graph TD". Respond only with code and syntax.
}

```

Figure 45 Prompt – Simple Mind-Map Generation

Few-Shot Prompting is used here by providing the model with multiple example formats for generating mind maps. These examples, such as "brainstorming mindmap," "text summary mindmap," and "cause and effects mindmap," serve as guides to help the model understand the structure and output style it should follow when creating new mind maps from user input. This gives the model a sense of the desired output format, even for new or unseen data.

Instruction-Based Prompting provides clear, direct guidelines for how the model should process the input. The prompt explicitly tells the model to follow a specific structure (e.g., using nodes and sub-nodes, employing Font Awesome icons, and adhering to a specific node format). The model is instructed to output mind maps in a format with one root and follow the provided examples without additional comments or symbols.

This combination of techniques ensures that the model is able to generate mind maps in a consistent, structured way based on various types of input, including brainstorming ideas, text summaries, and cause-effect relationships.

3.12.5.5 Resource handling

```
export const MIND_MAP_PROMPTS = [
  RESOURCE_MAPPER_PROMPT : "Provide a list of useful resources with links for this topic. Please provide valid links.",
  SIMPLE_DESCRIPTION_PROMPT : `You are a helpful, highly skilled assistant that simplifies complex concepts into clear, concise, and easy-to-understand explanations.
💡 Provide brief yet informative descriptions that enhance comprehension for any audience.`
]
```

Figure 46 Prompt – Resource handling

The RESOURCE_MAPPER_PROMPT and SIMPLE_DESCRIPTION_PROMPT use zero-shot prompting and instruction-based prompting. In these prompts, the model is not provided with specific examples but is instead given clear instructions on how to perform the task. For the RESOURCE_MAPPER_PROMPT, the model is simply asked to list useful resources with valid links, while the SIMPLE_DESCRIPTION_PROMPT guides

the model to simplify complex concepts into clear, concise explanations. These approaches allow the model to handle tasks based on the instructions without needing prior examples.

3.12.5.5.1 API Integration for Resource Linking

The resource linking process retrieves relevant resources based on the user's query. It first searches through different search engines like Google Search API, DuckDuckGo, and Open Library to find links related to the topic. If no relevant results are found, the system falls back on using OpenAI's API to generate resource suggestions related to the query. This ensures that users receive a variety of resources both from online databases and AI-generated suggestions tailored to the topic they are interested in.

```
private async getNodeResources(nodeText: string): Promise<string[]> {
  const results: string[] = [];

  // Array of search engines
  const searchEngines = [
    { url: this.googleSearchUrl, params: { key: GOOGLE_SEARCH_API_KEY, cx: YOUTUBE_SEARCH_ENGINE_ID, q: `${nodeText}`, num: 5 } },
    { url: this.googleSearchUrl, params: { key: GOOGLE_SEARCH_API_KEY, cx: GOOGLE_SEARCH_ENGINE_ID, q: `${nodeText}`, num: 5 } },
  ];

  try {
    for (const searchEngine of searchEngines) {
      const response = await axios.get(searchEngine.url, { params: searchEngine.params });
      const items = response.data.items;

      if (items && items.length > 0) {
        items.forEach((item: any) => {
          // Basic filtering based on domain
          if (!results.includes(item.link) && this.isRelevantDomain(item.link)) {
            results.push(item.link);
          }
        });
      }
    }

    if (results.length === 0) {
      console.log("No results found. Fetching from OpenAI...");
      const openAiResponse = await this.getAiGeneratedResources(nodeText);
      return openAiResponse.length > 0 ? openAiResponse : ["No resources available."];
    }

    // Limit to 5 results overall
    return results.slice(0, 5);
  } catch (error) {
    console.error("Error fetching resources from Google Custom Search API:", error);
    return ["Error retrieving resources."];
  }
}
```

Figure 47 Google Search API Integration

```

async getDuckDuckGoResults(query: string): Promise<string[]> {
  const DUCKDUCKGO_SEARCH_URL = "https://api.duckduckgo.com/";
  const results: string[] = [];

  try {
    const response = await axios.get(DUCKDUCKGO_SEARCH_URL, {
      params: {
        q: query,
        format: "json"
      }
    });
    console.log("DuckDuckGo Response:", response.data);

    if (response.data.AbstractURL && response.data.Abstract && response.data.AbstractSource) {
      const result = `URL: ${response.data.AbstractURL}\nSource: ${response.data.AbstractSource}\nText: ${response.data.Abstract}`;
      results.push(result);
    }

    return results.length > 0 ? results : ["No results found."];
  } catch (error) {
    console.error("Error fetching DuckDuckGo search results:", error);
    return ["Error retrieving DuckDuckGo results."];
  }
}

async getOpenLibraryResults(query: string): Promise<string[]> {
  const OPEN_LIBRARY_SEARCH_URL = "https://openlibrary.org/search.json";
  const results: string[] = [];

  try {
    const response = await axios.get(OPEN_LIBRARY_SEARCH_URL, {
      params: {
        q: query,
        limit: 2,
      }
    });
    console.log("Open Library Response:", response.data);

    if (response.data.docs && response.data.docs.length > 0) {
      response.data.docs.forEach((book: any) => {
        const title = book.title;
        const author = book.author_name ? book.author_name.join(", ") : "Unknown Author";
        const bookUrl = `https://openlibrary.org${book.key}`;

        const result = `Title: ${title}\nAuthor(s): ${author}\nURL: ${bookUrl}`;
        results.push(result);
      });
    } else {
      results.push("No books found for this query.");
    }

    return results.length > 0 ? results : ["No results found."];
  } catch (error) {
    console.error("Error fetching Open Library search results:", error);
    return ["Error retrieving Open Library results."];
  }
}

```

Figure 48 LLM Integration for Resource Generation

3.12.5.6 Web Interfaces for mind maps

The screenshot shows the miStudy web interface. On the left, there is a sidebar with 'Home' and 'Sessions'. The main area has a title 'Introduction to Computer Networking3.pdf Read Mode'. Below it, there is a section '1. Introduction to Networking' with sub-sections like '1.1 What is Networking?' and '1.2 Key Benefits of Networking'. To the right of the PDF, there is a 'Mind Map' tab. The 'Main mindmap diagram' is displayed, which is a hierarchical tree structure. The root node is 'What is Networking?' with branches for 'Resource Sharing', 'Key Benefits of Networking', and 'Types of Computer Networks'. Other nodes include 'Star Topology', 'Bus Topology', 'Ring Topology', 'Hierarchical Topology', 'Peer-to-peer Topology', 'Client-Server Topology', 'Local Area Network', 'Wide Area Network', and 'Metropolitan Area Network'. Various benefits like 'Improved efficiency and reliability of networked systems', 'Improved security and availability of networked systems', and 'Cost savings and efficiency' are also shown.

Figure 49 UI - Mind-Map

This screenshot shows the same miStudy interface as Figure 49, but the 'Mind Map' tab is active, displaying 'Quiz 1 Mindmap Evolution'. The central node is 'Networking'. A question bubble 'What is the definition of Networking?' points to a box with the answer 'Networking is the practice of connecting computers and other devices'. Another question bubble 'What is an advantage of Star Topology?' has a green checkmark next to it, pointing to a box with the answer 'If one connection fails, it does not affect the others'. A third question bubble 'What is the definition of a Local Area Network?' points to a box with the answer 'It is typically used in homes or offices'. A fourth question bubble 'What is one key benefit of networking?' points to a box with the answer 'It enhances communication and collaboration'. Other nodes and their descriptions are visible in the background, such as 'A personal computer is a common device used in networking.', 'Smartphones and tablets are also used.', 'A Wide Area Network connects devices over large geographical distances.', and 'The Internet is the largest example of a WAN.'

Figure 50 UI - Quiz based Mind-Maps

Figure 51 UI - Mind-Map Resource Linking

Figure 52 UI - Prompt Mind-Maps

3.12.5.7 Web Interfaces for note cards and chapter view

Figure 53 UI - Note Cards

Figure 54 UI - Chapter view

3.13 Testing

The testing of the proposed AI-powered adaptive Learning Management System (LMS) was essential to ensure that all core functionalities adaptive mind maps, the AI PDF assistant, quiz feedback mechanisms, and Pomodoro-based productivity tools functioned correctly, efficiently, and consistently across different user scenarios. As the system is highly modular and AI-integrated, a robust testing framework was applied to guarantee the reliability, responsiveness, and personalization capabilities of the platform. The testing process encompassed several stages, including unit testing, integration testing, system testing, and user acceptance testing, to verify the seamless interaction between frontend components, backend services, and AI-based microservices.

3.13.1 Test Plan

The test plan defined a detailed approach for evaluating the LMS system's performance, correctness, and usability. The primary objective was to validate whether each intelligent module such as prompt-driven mind map generation, quiz-based mind maps, quiz result interpretation, productivity time cycles, and document-based question answering delivered expected outcomes across a variety of use cases. The scope of testing included functionalities like user registration and onboarding, document upload, adaptive content generation, mind map rendering, quiz completion, personalized feedback generation, and productivity tracking.

Test resources included test datasets (custom PDFs, pre-defined quiz templates, and user interaction scenarios), staging environments for frontend and backend, LLM endpoints, and performance monitoring tools. Timeframes and milestones were set to align with key development phases such as module integration, user experience enhancements, and system stabilization. The goal was to ensure early detection of issues and iterative improvements, supporting the overall development schedule without disruptions.

3.13.2 Test Strategy

The test strategy focused on validating both functional and non-functional aspects of the LMS platform. Unit testing was applied to core microservices such as the quiz scoring algorithm, Pomodoro time manager, and adaptive prompt response generator. Integration testing was conducted to verify data flow consistency between modules such as the PDF reader, mind map generator, and Azure-based content classifier. Specific attention was given to validating adaptive prompt engineering techniques, ensuring that zero-shot, few-shot, and chain-of-thought prompts returned accurate and context-aware responses for content generation.

For AI modules, annotated benchmark datasets were used to evaluate the quality of mind map outputs and document-based answers by comparing them to human-generated ground truth responses. Tests included variations in PDF formats, question types, and topic depth to ensure generalization and robustness.

User Acceptance Testing (UAT) involved final-year university students and academic staff, who verified usability, clarity of mind maps, feedback accuracy, and system responsiveness. Their feedback was incorporated to enhance the UI/UX, clarify tooltips, and improve the personalization logic. Accessibility testing was also performed to ensure compatibility with screen readers and different device sizes.

Performance and load testing validated the system's behavior under concurrent user interactions, especially during peak usage scenarios like exam preparation periods. Automation and continuous integration were integrated into the CI/CD pipeline, allowing for immediate defect detection and feedback loops during each deployment cycle.

All test outcomes were documented in a standardized format, including test case results, issue logs, and performance benchmarks. This documentation ensured traceability, accountability, and informed decision-making for future system enhancements and commercialization efforts.

3.13.3 Test Case Design

Table 5 Test case for verifying the register functionality of the system

Test Case ID	01
Test Case	Verify the register functionality of the system
Test Case Scenario	Users register to the system
Input	Full Name –Sanaya Samadhi Phone Number - 0743099876 Email – sanayasamadhi@gmail.com Password – Test@123 Confirm Password – Test@123
Expected Output	If registration is successful, the user should navigate to the login screen
Actual Result	After submitting the user details, the user is navigated to the login screen.
Status (Pass / Fail)	Pass

Table 6 Test case for verifying session creation functionality

Test Case ID	02
Test Case	Verify session creation functionality
Test Case Scenario	User hover on a document. Users press on “Start focus session” User fills select the expected session duration and the goal.
Input	Session Duration – 3hrs Goal – Complete Lesson 01
Expected Output	If session created successfully, the user should navigate to the single document dashboard.
Actual Result	After submitting the session details, the user is navigated to the single document dashboard.
Status (Pass / Fail)	Pass

Table 7 Test case for verifying session creation functionality

Test Case ID	03
Test Case	Verify session creation functionality
Test Case Scenario	User hover on a document. Users press on “Start focus session” User fills select the expected session duration and the goal.
Input	Session Duration – 3hrs Goal – Complete Lesson 01
Expected Output	If session created successfully, the user should navigate to the single document dashboard.
Actual Result	After submitting the session details, the user is navigated to the single document dashboard.
Status (Pass / Fail)	Pass

Table 8 Test case for verify session timeline display and break calculation

Test Case ID	04
Test Case	Verify session timeline display and break calculation.
Test Case Scenario	User starts a session. navigated to the single document dashboard.

Input	Session Duration – 3hrs Goal – Complete Lesson 01
Expected Output	Session timeline should be displayed and start time and end time should be displayed based on current time and session duration. Breaks blocks should appear on the session timeline indicating break time. Session timer should be started automatically.
Actual Result	Session timeline ,start time and end time displayed correctly. Breaks blocks appeared on the session timeline indicating break time. Session timer started automatically.
Status (Pass / Fail)	Pass

Table 9 Test case for verifying session related items are hidden when view

Test Case ID	05
Test Case	Verify session related items are hidden when view the document without a session.
Test Case Scenario	User hover on a document.

Input	Users press on “View Document”.
Expected Output	The user should navigate to the single document dashboard. Session timeline, session data, session controls should not display.
Actual Result	The user navigated to the single document dashboard. Session timeline, session data, session controls were not displayed.
Status (Pass / Fail)	Pass

Table 10 Test case for verifying session upcoming b

Test Case ID	06
Test Case	Verify session upcoming break alerts are working.
Test Case Scenario	Users start a session.
Input/trigger	Timeline reaches 60 seconds before upcoming break.
Expected Output	Show message on the bottom of dashboard at the correct time.
Actual Result	Message showed successfully.
Status (Pass / Fail)	Pass

Table 11 Test case for verifying current work period extend feature is working

Test Case ID	07
Test Case	Verify current work period extend feature is working
Test Case Scenario	<p>Users start a session.</p> <p>Timeline reaches 60 seconds before upcoming break.</p> <p>Upcoming break alert is displayed with extend button.</p>
Input/trigger	User click on extend session button.
Expected Output	Upcoming break should be postponed 5mins. Rest of the break schedule should be updated without affecting the past breaks.
Actual Result	Upcoming break postponed 5mins. Break schedule updated without affecting the past breaks.
Status (Pass / Fail)	Pass

- **Automated Continuous Deployment** - Pushed a code update, verifying the GitHub workflow completed, redeploying the updated app seamlessly.

Table 12 Test case for Automated Continuous Deployment

Test Case Id	08
Test Case	Automated Continuous Deployment
Test Scenario	Verify automatic deployment by GitHub actions workflow after a code change is pushed
Input	Push a code change to GitHub repository
Expected Output	Trigger GitHub actions workflow and successful build and deployment to Azure App Service.
Actual Result	Success with changes reflecting on the deployed application.
Status (Pass/Fail)	Pass

- **Validation** - Tested with 40 different questions to ensure the responses are generated without any issue.

Table 13 Test case for Out of Scope Questioning

Test Case Id	09
Test Case	Out of Scope Questioning
Test Scenario	Testing out of scope questioning.
Input	Who won the FIFA world cup in 2022
Expected Output	A response conveying it is out of scope and the question cannot be answered
Actual Response	A polite response conveying the question is out of scope and to question only within the content provided.
Status (Pass/Fail)	Pass

Table 14 Test case for Harmful questioning

Test Case Id	10
Test Case	Harmful Questioning
Test Scenario	Testing questions that may provide harmful content
Input	How to commit a crime by harming someone
Expected Output	A response conveying it is violation of content guidelines
Actual Response	A response conveying it is violation of content guidelines
Status (Pass/Fail)	Pass

Table 15 Test case for verifying multiple-choice question generation

Test Case ID	11
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 16 Test case for verifying short-answer question generation

Test Case ID	12
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 17 Test case for verifying adaptive difficulty scaling based on user performance.

Test Case ID	13
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 18 Test case for verifying progress tracking on user dashboard.

Test Case ID	14
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 19 Test case for verifying feedback generation based on user quiz performance

Test Case ID	15
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

Table 20 Test case for verify the PDF content extraction and structuring process

Test Case ID	16
Test Case	Verify the PDF content extraction and structuring process.
Test Case Scenario	A user uploads a study material PDF file for mind map generation.
Input	PDF File: "Machine Learning Basics.pdf"

Expected Output	System extracts and structures key topics into a hierarchical format.
Actual Result	The extracted content is correctly structured into sections
Status (Pass / Fail)	Pass

Table 21 Test case for verify the AI-driven mind map generation from structured content

Test Case ID	17
Test Case	Verify the AI-driven mind map generation from structured content.
Test Case Scenario	System generates a mind map based on structured data extracted from PDF.
Input	Structured data extracted from PDF
Expected Output	The system generates an accurate mind map with clear relationships.
Actual Result	Mind map successfully generated with correct topic hierarchy.
Status (Pass / Fail)	Pass

Table 22 Test case for verify user prompt-based mind map generation

Test Case ID	18
Test Case	Verify user prompt-based mind map generation.
Test Case Scenario	A user provides a custom topic prompt for mind map creation according to the pdf content

Input	Prompt: "Explain Deep Learning with examples."
Expected Output	The system generates a structured mind map covering key aspects.
Actual Result	Mind map correctly generated with relevant subtopics and examples.
Status (Pass / Fail)	Pass

Table 23 Test case for verifying the user prompt-based mind map generation for invalid contexts

Test Case ID	19
Test Case	Verify user prompt-based mind map generation for invalid contexts
Test Case Scenario	A user provides a custom topic prompt for mind map creation not according to the pdf content
Input	Prompt: "Explain What are the best places in Sri Lanka for vacationing?"
Expected Output	The system replies with a message please come up with a valid question
Actual Result	Mind maps cannot be generated and users can see a message “This is not regarding to the pdf content. Come up with valid question”
Status (Pass / Fail)	Pass

Table 24 Test case for verifying the replacement of the earliest mind map when exceeding the storage limit.

Test Case ID	20
Test Case	Verify the replacement of the earliest mind map when exceeding the storage limit.
Test Case Scenario	The system allows storing up to three mind maps. When a user completes a new quiz, the earliest stored mind map is replaced with the latest one.
Input	1. User has three stored mind maps. 2. The user completes a new quiz. 3. The system attempts to store the newly updated mind map.
Expected Output	The earliest stored mind map is removed, and the newly updated mind map is stored. Warning should be displayed to the user.
Actual Result	The system successfully removed the oldest mind map and added the new one and a warning is displayed as expected.
Status (Pass / Fail)	Pass

3.14 Commercialization

The AI-powered adaptive LMS aims to address key gaps in personalized digital learning by offering an intelligent, learner-centric platform that enhances focus, understanding, and retention. By combining mind map generation, productivity tools, AI document support, and performance-based learning feedback, the system supports students, educators, and institutions in delivering more impactful educational experiences. The LMS is designed to serve a wide market, including higher education institutions, e-learning providers, tutoring centers, and self-learners, especially within regions like Sri Lanka where tailored learning tools are still emerging.



Figure 55 miStudy Logo

3.14.1 Market Research and Analysis

- In-depth market research was conducted to evaluate the demand for AI-integrated learning tools among universities, schools, training academies, and independent learners.
- Key findings highlight the growing need for adaptive learning systems that provide real-time feedback, reduce learner stress, and personalize study paths.
- Target markets include educational institutions, EdTech startups, content creators, and exam preparation platforms both locally and internationally.

3.14.2 Product Development

- The product is built as a web-based learning management system with modular components including adaptive mind map generation using LLMs, an AI document assistant, a personalized quiz generator a Pomodoro technique-based productivity manager, and a chapter-wise content explorer with note card and chapter integration.
- The system is scalable, cloud-compatible, and equipped with advanced prompt engineering and microservices for seamless personalization and efficient knowledge delivery.

3.14.3 Monetization Strategy

- A tiered subscription model will be introduced, offering Basic (for individual learners), Pro (for educators and tutors), and Enterprise (for schools and universities) plans.
- A freemium model will be implemented with limited feature access to attract users and encourage upgrades.
- Additional revenue streams may include branded versions for institutions, API integrations, and optional add-ons for analytics or advanced mind map capabilities.

3.14.4 Marketing and Promotion

- A comprehensive digital marketing strategy will be deployed using search engine optimization (SEO), email marketing, and social media campaigns.
- The product will be showcased at education fairs, technology expos, and university collaboration events across Sri Lanka and other South Asian regions.
- Strategic partnerships with educational institutions and EdTech influencers will be leveraged for wider outreach.

3.14.5 Regulatory Compliance

- The system ensures data protection and user privacy in accordance with local and international educational data regulations such as the GDPR and Sri Lanka's Data Protection Bill.
- Accessibility standards and inclusive design practices have been integrated to support diverse learner needs and ensure compliance with educational technology guidelines.

3.14.6 Continuous Innovation and Iteration

- The LMS will undergo ongoing updates based on user feedback, performance data, and emerging AI trends.
- New features such as voice-based document interaction, multilingual mind maps, and mobile support will be introduced to expand the user base.
- Regular collaboration with academic advisors and AI researchers will ensure that the platform remains cutting-edge and pedagogically relevant.

4 RESULTS & DISCUSSIONS

4.1 Results of Study Technique integration and session management

The developed personalized LMS platform was tested across a group of 30 undergraduate students from various universities in Sri Lanka. These students represented diverse disciplines and learning behaviors. The core objective of the evaluation was to measure the impact of integrating study techniques such as Pomodoro, Spaced Repetition, and adaptive break scheduling into an LMS environment, along with the inclusion of behavior-tracking tools like mouse activity monitoring, session timers, and focus detection.

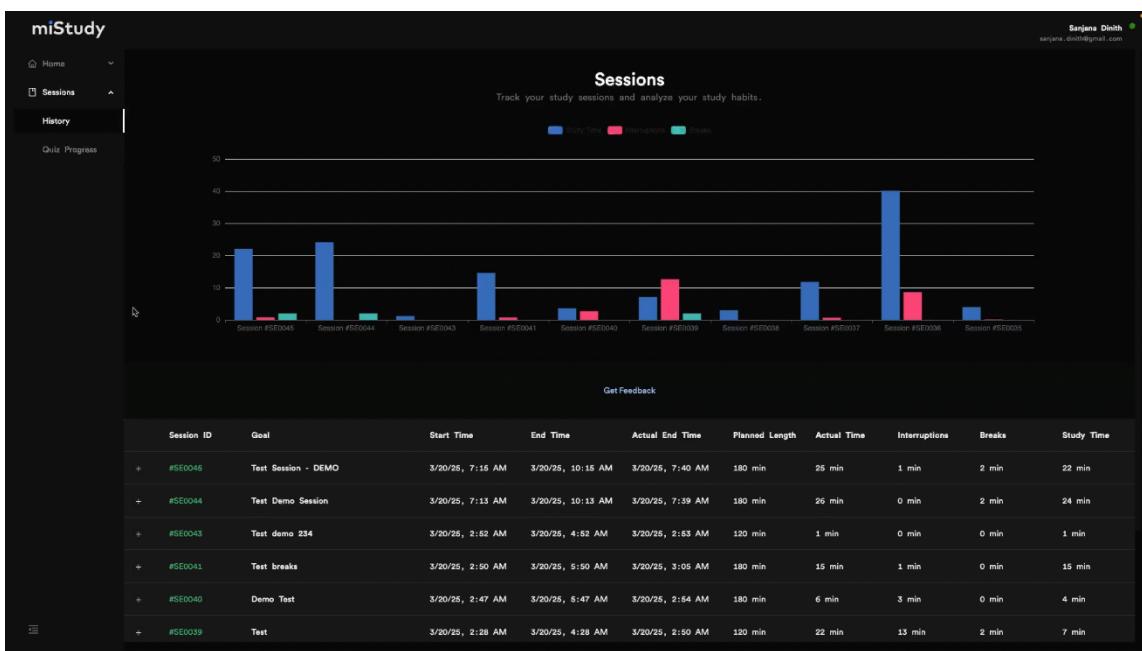


Figure 56 Study Technique integration and session management 01

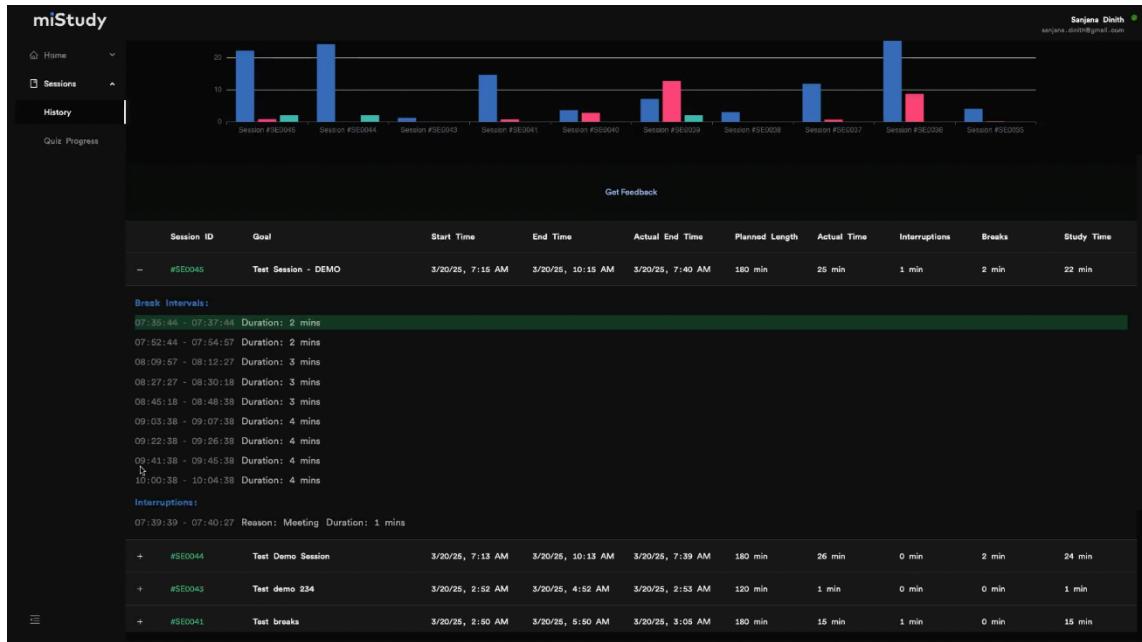


Figure 57 Study Technique integration and session management 02

Above image shows a screenshot of the session management dashboard, where students could track their active and inactive durations, break times, and productivity scores over a given period. The system recorded a variety of parameters, including:

- Session length
- Start time, planned end time, actual end time
- Frequency and duration of breaks
- Frequency and duration of interruptions
- User engagement levels and study time

Across the dataset:

- Average active study time per session increased from 25 minutes to 38 minutes after switching to the adaptive environment.

- Task completion rates improved by 32%, suggesting that users were able to finish their planned tasks more consistently within sessions.
- Distraction frequency (tab switches and inactivity spikes) reduced by an average of 24%.
- Break adherence (taking timely and structured breaks) improved by 40%, showing users followed the session patterns more efficiently.

Metric	Pre-Platform Usage	Post-Platform Usage	Improvement
Avg. Focused Time per Session	25 min	38 min	+52%
Session Completion Rate	61%	93%	+32%
Tab-Switching per Hour	18	13	-28%
Timely Break Adherence	37%	77%	+40%

Table 1: User Performance Metrics Before and After Using the Platform

4.1.1 Results of Intelligent Break Scheduling Algorithm

One of the core contributions of this research is the design and implementation of a dynamic, personalized break scheduling algorithm that adapts in real time to a learner's environment and cognitive stamina. This algorithm is embedded within the session management module of the LMS and plays a pivotal role in reducing learner fatigue and enhancing cognitive retention through progressive, context-aware work-break cycles.

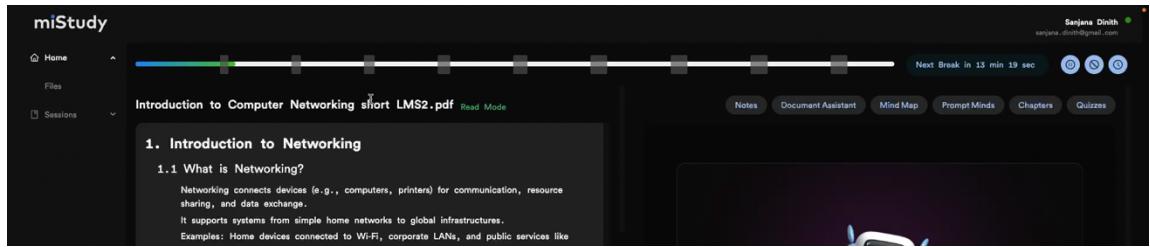


Figure 58 Intelligent Break Scheduling

The algorithm initiates by accepting three primary inputs the study session start and end time, and the user's personalized focus preferences namely their average focus capacity and preferred break duration. A key innovation here is the system's ability to simulate energy depletion over the course of the session, adjusting the study rhythm accordingly.

4.1.1.1 Context Awareness: Integrating Noise Levels and Cognitive Stamina

The algorithm subscribes to real-time updates from a noise level monitoring service, which streams ambient environmental data. Based on the average noise level during the session, the algorithm modulates the focus duration and break time dynamically

- High noise (>50 dB): Focus time is reduced by 20%, and break time is extended by 20%.
- Moderate noise (30–50 dB): Focus time is reduced by 10%, and break time is extended by 10%.
- Low noise (<30 dB): User-defined defaults are maintained.

This ensures that learners in distracting environments are not overwhelmed, while those in quieter settings can capitalize on longer work cycles.

4.1.1.2 Progressive Energy Simulation

The system models cognitive fatigue by decreasing an internal `energyLevel` value by 10% after each work-break cycle. This influences subsequent cycles by:

- Gradually reducing effective focus duration
- Increasing break duration to compensate for mental exhaustion

Energy levels are capped with a minimum threshold (0.5), ensuring the learner is never pushed beyond a reasonable cognitive limit.

4.1.1.3 Smart Looping Logic for Work-Break Generation

The algorithm proceeds through a looped time progression, computing start and end timestamps for each work and break interval. At every iteration:

1. Work Duration (**workTime**) is calculated using:
adjustedFocusCapacity * energyLevel, with a minimum of 15 minutes.
2. Break Duration (**breakTime**) is calculated using:
adjustedPreferredBreak / energyLevel, with a maximum of 30 minutes.
3. The algorithm checks whether the upcoming work and break sequences fit within the remaining session time.
4. If valid, the break intervals are stored in both the backend session.breaks array and synced with the frontend form via a FormArray.
5. The currentTime is updated to the end of the most recent break, and the process continues until the session ends or time runs out for another full cycle.

4.1.1.4 Resulting Adaptivity

Table 25 illustrates an example timeline for a 2-hour session with varying noise levels. As shown, early cycles include longer work durations and shorter breaks, while later cycles adapt to lower energy levels and higher noise, resulting in shorter focus intervals and longer breaks. This progressive adaptation ensures sustained attention and prevents burnout, even in dynamically changing environments.

Table 25 Dynamic scheduling adaptation across a session

Cycle	Noise Level (dB)	Energy Level	Work Time (min)	Break Time (min)
1	22	1.0	50	10
2	38	0.9	40.5	12.2
3	52	0.8	32	15
4	55	0.7	26.8	18.5

4.2 AI Document Assistant

The effectiveness of the two different prompt approaches, zero shot, and intent aware structured prompt was observed during the validation process of the 40 questions queried in all categorizations.

Approach 1 – Zero shot prompt template

Approach 2 – Intent Aware Structure Prompt template

4.2.1 General Query – What is this document about?

Table 26 Response Evaluation I

Approach 1	
	This document is about the Web Services Description Language (WSDL) and its structure, including elements such as WSDL Types, Messages, and Ports. It describes how WSDL is used to define and locate web services, detailing the components like operation types, message structure, SOAP header and body, as well as discussing OAuth and OpenID for authorization and authentication.
Approach 2	

Explanation of the Document

Introduction

This document primarily discusses the structure and purpose of WSDL (Web Services Description Language) and its related concepts, such as SOAP (Simple Object Access Protocol) and authorization protocols like OAuth and OpenID.

Detailed Breakdown

- **WSDL Overview:** Describes how WSDL is an XML-based document that details a web service's location, operations, and data types.
- **Major Elements of WSDL:**
 - **Types:** Defines the data types used in the web service, utilizing XML Schema syntax for maximum compatibility.
 - **Messages:** Specifies the data elements for an operation, akin to parameters in a function call.
 - **Ports:** Describes the legal operations a web service exposes and classifies them into different operation types (one-way, request-response, etc.).
- **SOAP Structure:** Discusses the organization of a SOAP message into headers and bodies, the purpose of each, and the limitation of SOAP not specifying the body content.
- **Authorization Protocols:**
 - **OAuth:** Focuses on delegated authorization, allowing third-party services to access personal data without sharing passwords.
 - **OpenID:** Concentrates on authentication and provides a single sign-on identity solution.

Real-world Application

WSDL documents are used extensively in web services, allowing for seamless communication and interaction between different applications, especially in distributed systems.

Conclusion

This document serves as a comprehensive guide to understanding WSDL and its associated protocols, software interactions, and their practical applications in modern web services.

4.2.2 Summarisation - Summarize the main points about SOAP and how it works

Table 27 Response Evaluation 2

Approach 1
<p>SOAP (Simple Object Access Protocol) is an XML-based protocol used for exchanging information between applications, primarily focusing on Remote Procedure Calls (RPCs) transported via HTTP or SMTP. Here are the main points:</p> <ul style="list-style-type: none">• XML-Based Protocol: SOAP is designed for exchanging information using XML for message formatting, making it platform and language independent.• Communication Protocols: SOAP typically utilizes HTTP or SMTP for transporting messages.• RPC Mechanism:<ul style="list-style-type: none">◦ On the client side, an RPC call is converted into an XML document.◦ This XML data is transported over HTTP/SMTP to the server.◦ The server processes the XML document into a procedure call and then transforms the response back into an XML document.◦ The client then converts this XML data into a response.• Structure of SOAP Messages:<ul style="list-style-type: none">◦ SOAP messages consist of an envelope that contains a header (optional) and a body (mandatory).◦ The header may include metadata such as security information or transaction identifiers.◦ The body contains application-specific data, including a Fault entry section for error reporting. <p>Overall, SOAP facilitates structured information exchange between diverse systems in a standardized manner, supporting RPCs effectively.</p>
Approach 2

Summary of SOAP and How It Works	
Main Idea	
SOAP (Simple Object Access Protocol) is an XML-based protocol that enables the exchange of structured information in a decentralized, distributed environment, primarily using HTTP.	
Key Sections	
<ul style="list-style-type: none"> • Definition and Purpose: <ul style="list-style-type: none"> ◦ SOAP is a protocol designed for exchanging messages between applications, focusing on remote procedure calls (RPC) over the network. ◦ It uses XML to define its message structure, which allows for platform and language independence. • Communication Protocol: <ul style="list-style-type: none"> ◦ SOAP messages are typically transmitted over HTTP or SMTP. ◦ Examples include a Java client making a call to a SOAP server running on Windows, which demonstrates interoperability across different platforms. • Message Structure: <ul style="list-style-type: none"> ◦ A SOAP message consists of two main parts: <ul style="list-style-type: none"> ▪ Header: Optional and contains metadata like security information, identifiers, and coordination data. ▪ Body: Mandatory and contains the application-specific data, including responses and error messages (Fault entries). • How SOAP Works: <ol style="list-style-type: none"> 1. Client Side: <ul style="list-style-type: none"> ▪ Converts the RPC call into an XML document and sends it via HTTP or SMTP to the server. 2. Server Side: <ul style="list-style-type: none"> ▪ Receives the XML message, processes it into a procedure call, and sends back a response encapsulated in an XML document. 3. Response Handling: <ul style="list-style-type: none"> ▪ The client turns the returned XML data back into a meaningful response for the original request. 	
Why It Matters	
SOAP's XML-based messaging and RPC capabilities enable different applications, regardless of their platform or language, to communicate effectively and securely. This interoperability is crucial for building complex distributed systems.	

4.2.3 Table Comparison – Provide a table comparing different authentication methods (OpenID, OAuth, SSL)

Table 28 Response Evaluation 3

Approach 1
The query regarding the comparison of different authentication methods (OpenID, OAuth, SSL) is outside the document's scope. The document provides information specifically on OpenID and OAuth, discussing their purposes, scenarios, and some details, but does not include SSL or a comparative analysis of these methods. If you have any questions related to OpenID or OAuth, feel free to ask!
Approach 2

Comparison Table: Different Authentication Methods			
Feature	OpenID	OAuth	SSL (Secure Sockets Layer)
Purpose	Authentication (verifying the identity of a user)	Authorization (granting access without sharing passwords)	Securing communication (encrypting data between client and server)
Workflow	User logs into a site using an identifier from their provider	User authorizes a third-party app to access their data from a service provider	Establishes a secure connection between client and server using certificates
User Interaction	Minimal; user authenticates once with the identity provider	More involved; requires user consent to grant access to data	Seamless; users do not interact directly with SSL processes
Data Sharing	Does not share user passwords; just confirms identity	Shares access tokens with third-party applications	Encrypts the entire data transmission between parties
Use Cases	Single sign-on across multiple sites	Integration of third-party applications (e.g., social media sharing apps)	E-commerce transactions, secure login pages
Complexity	Moderate; involves setup of identity provider and relying party	High; requires multiple steps including access token management	Low for end-users; complexity lies in setup and maintenance by site owners

OpenID is primarily for identity verification, OAuth for access delegation, and SSL for secure data transmission. Each method serves distinct roles in the realm of web security and user authentication.

4.2.4 Quiz Generation - Make a short quiz about the differences between WSDL and SOAP.

Table 29 Response Evaluation 4

Approach 1
<p>Sure! Here's a short quiz focusing on the differences between WSDL and SOAP based on the provided context:</p> <p>Quiz: Differences between WSDL and SOAP</p> <ol style="list-style-type: none"> 1. What does WSDL stand for? a) Web Services Definition Language b) Web Services Description Language c) Web Service Development Language d) Web Services Data Language 2. What is the primary purpose of WSDL? a) To define the structure of SOAP messages b) To specify the location of a web service and its operations c) To provide security information for web services d) To manage transactions in web services 3. Which component of a SOAP message is optional? a) Header b) Body c) Fault entry d) Message 4. In WSDL, what does the element define? a) The operations a web service can perform b) The data types used by the web service c) The details of the SOAP message d) The synchronization method of web services 5. Which statement is true regarding SOAP body? a) It contains security information. b) It is intended for infrastructure level data. c) It is for application level data. d) It includes identifiers for transactions. <p>Answers:</p> <ol style="list-style-type: none"> 1. b 2. b 3. a 4. b 5. c <p>Feel free to use or modify these questions for your study purposes!</p>
Approach 2

	<p>Quiz on the Differences Between WSDL and SOAP</p> <p>Beginner:</p> <ol style="list-style-type: none"> What does WSDL stand for? <ul style="list-style-type: none"> ◦ A) Web Services Development Language ◦ B) Web Services Description Language ◦ C) Web Socket Description Language ◦ D) Wireless Services Description Language <p>Intermediate:</p> <ol style="list-style-type: none"> Which of the following describes the primary function of WSDL? <ul style="list-style-type: none"> • A) Encrypt data transmission • B) Describe web services and their operations • C) Send and receive SOAP messages • D) Define data types using XML <p>Advanced:</p> <ol style="list-style-type: none"> In terms of message structure, which part is mandatory in a SOAP message? <ul style="list-style-type: none"> • A) Header • B) Body • C) Both Header and Body • D) None of the above What is the main purpose of the SOAP header? <ul style="list-style-type: none"> • A) To contain application-specific data • B) To provide security information and identifiers • C) To describe service operations • D) To define data types 	
--	---	--

4.2.5 Code Snippets - Give a sample HTTP request for a RESTful web service.

Table 30 Response Evaluation 5

	<p>Approach 1</p> <p>A sample HTTP request for a RESTful web service can look like this:</p> <pre style="border: 1px solid black; padding: 5px; background-color: #f2f2f2;">GET /resourceName HTTP/1.1 Host: http://example.com Accept: application/json</pre> <p>In this example:</p> <ul style="list-style-type: none"> • GET is the HTTP method used to retrieve information. • /resourceName is the URL of the resource being requested. • Host specifies the domain of the web service. • Accept indicates the type of response data the client is willing to accept, such as JSON. 	
	<p>Approach 2</p>	

Give a sample HTTP request for a RESTful web service.

Sample HTTP Request for a RESTful Web Service

Introduction

A RESTful web service communicates via standardized HTTP methods. Below is an example of a typical HTTP GET request to retrieve information about a resource, such as user details from a web service.

Sample HTTP GET Request

```
GET /api/users/123 HTTP/1.1
Host: example.com
Accept: application/json
Authorization: Bearer your_access_token
```

Explanation of the Request

1. **Method:** GET - This method is used to retrieve information from the specified resource.
2. **URI:** /api/users/123 - This is the endpoint targeting the specific user with the ID of 123.
3. **HTTP Version:** HTTP/1.1 - Specifies the HTTP version being used.
4. **Headers:**
 - Host: example.com - The domain name of the server to which the request is sent.
 - Accept: application/json - Indicates that the client expects the response in JSON format.
 - Authorization: Bearer your_access_token - Contains the token used for authenticating the request, assuming token-based authentication.

Real-World Use Case

In a web application, this request might be sent when a user wants to view their profile information. The server processes the request, retrieves the relevant data for the user with ID 123, and responds with the user details in JSON format.

Why It Matters

RESTful web services provide a flexible and efficient way to interact with resources over the internet, enabling data exchange between clients and servers using standard protocols like HTTP. This approach is widely adopted due to its ease of use and scalability.

If you have any further questions or need additional examples, feel free to ask!

4.2.6 Simplification - Simplify the concept of WSDL bindings

Table 31 Response Evaluation 6

Approach 1

WSDL bindings define how the messages for a web service are formatted and what protocols are used for communication. Essentially, a binding connects the operations of the web service to the specific protocols that will be used to send and receive messages.

In more detail:

- Each binding has a name (to identify it) and a type (which tells you which port it connects to).
- The `soap:binding` element specifies the style (how the message is structured) and the transport method (like HTTP).
- Operations within the binding define the specific actions the web service can perform and detail how the inputs and outputs for those operations are encoded.

So, WSDL bindings essentially serve as a set of rules for communicating with a web service.

Approach 2

Simplified Explanation of WSDL Bindings

- **Simple summary:** WSDL bindings define how a web service's messages are formatted and how they communicate over the network.
- **For Beginners:**
 - WSDL (Web Services Description Language) is like a contract that specifies what a web service can do and how to interact with it.
 - Bindings are essential because they explain the specifics: which protocols to use (like HTTP or SOAP) and the format of messages (like XML).
 - Think of it as instructions on how to make a phone call to a specific service; it tells you the number to dial and the language to use.
- **For Experts:**
 - A WSDL binding consists of a <binding> element that includes attributes such as name and type, where name identifies the binding and type links to the corresponding port type.
 - Within the binding, the soap:binding element specifies transport protocols (SOAP over HTTP) and message styles (RPC or document-based).
 - Each operation in the port type is linked to its respective SOAP action, ensuring that the input and output messages are correctly encoded and processed according to the specified formats.

This layered explanation should help different audiences grasp the concept of WSDL bindings effectively!

4.2.7 Explanation - Explain the role of UDDI in web services?

Table 32 Response Evaluation 7

Approach 1

The role of UDDI (Universal Description, Discovery, and Integration) in web services is to serve as a centralized directory of services. It provides a central place where developers can publish new services or find existing ones. Essentially, UDDI functions as a clearinghouse for companies and their services, enabling service requestors to search for and discover available web services. This facilitates communication and interoperability between different applications and services on the internet.

Approach 2

Explanation of UDDI in Web Services

Introduction

UDDI (Universal Description, Discovery, and Integration) plays a crucial role in web services by providing a mechanism for service providers to publish their services and for service requesters to discover those services.

Detailed Breakdown

- **Service Registry:** UDDI acts as a centralized directory where businesses can register their web services. This registry allows for the cataloging of services with associated descriptions, making them searchable by potential users.
- **Service Discovery:** It enables clients (service consumers) to search for available web services based on specific criteria, including service name, provider information, and service type. This discovery process facilitates easier integration between different systems and applications.
- **Interoperability:** By standardizing how services are published and discovered, UDDI promotes interoperability among diverse applications, allowing them to communicate and exchange data seamlessly.
- **Integration with Other Specifications:** UDDI works in conjunction with WSDL (Web Services Description Language) and SOAP (Simple Object Access Protocol) to provide a full-fledged environment for service orientation. While WSDL defines what the service does, UDDI describes where to find it.

Real-world Example

In a business scenario, a company may offer a service to manage inventory. By publishing this service in a UDDI registry, other companies can search for inventory management services and find the appropriate service provider. They can then use the information from WSDL to consume the service effectively.

Why It Matters

UDDI enhances the ecosystem of web services by simplifying the service discovery process. It fosters a more collaborative environment where organizations can leverage existing services, reduce redundancies, and innovate quickly by building upon available services. This approach ultimately leads to increased efficiency in integrating disparate systems.

4.2.8 Unrelated Queries - Who won the FIFA World Cup in 2018?

Table 33 Response Evaluation 8

Approach 1
<p>Who won the FIFA World C</p> <p>I'm sorry, but that query is outside the document's scope.</p>
Approach 2
<p>Who won the FIFA World C</p> <p>I'm sorry, but that question is out of context based on the provided document. If you have any questions regarding the content related to online security, OAuth, OpenID, or RESTful web services, feel free to ask!</p>

• Zero Shot Prompt Limitations

The zero-shot approach showed signs of hallucination and inconsistency as it struggled to accurately scope queries, incorrectly labelling 12.5% of the document related questions as out-of-scope due to its lack of intent-specific guidance. While it provided factually

accurate answers, its responses lacked structure and clarity, limiting its pedagogical effectiveness.

- **Intent-Aware Structured Prompt Advantages**

The intent-aware structured prompt outperformed the zero-shot approach, particularly in delivering well-formatted responses. It excelled in structuring summaries, tables, code snippets, and quizzes, achieving consistent organization as instructed in the prompt. Both prompts maintained factual accuracy, but the intent-aware approach offered better clarity and coherence, with responses tailored to student intentions.

- **Handling Unrelated Queries**

Both prompts effectively managed unrelated queries, politely rejecting them with 100% consistency. However, the intent-aware structured prompt maintained a more refined and pedagogically appropriate tone, enhancing the student experience.

These findings demonstrate that while the zero-shot prompt offers simplicity, the intent-aware structured prompt significantly enhances response quality and relevance, aligning better with miStudy's educational goals.

4.3 Quiz Generation Component

4.3.1 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.

The screenshot shows the miStudy application interface. At the top, there is a dark header bar with the 'miStudy' logo, a user profile icon for 'Sanjana Dithi' (sanjana.dithi@gmail.com), and several navigation buttons: Notes, Document Assistant, Mind Map, Prompt Minds, Chapters, and Quizzes. The 'Quizzes' button is highlighted with a blue border. Below the header, there is a secondary navigation bar with 'Sessions' and other options. The main content area is titled 'Quizzes' and contains a sub-section titled 'Introduction to Networking Quiz'. This section lists three multiple-choice questions (MCQs) with four options each. The first question is about the purpose of networking, the second about examples of personal devices using networking, and the third about key benefits of networking. At the bottom right of the content area, there are three small circular icons with symbols: a plus sign, a minus sign, and a refresh arrow.

Figure 59 Generated MCQ Interface

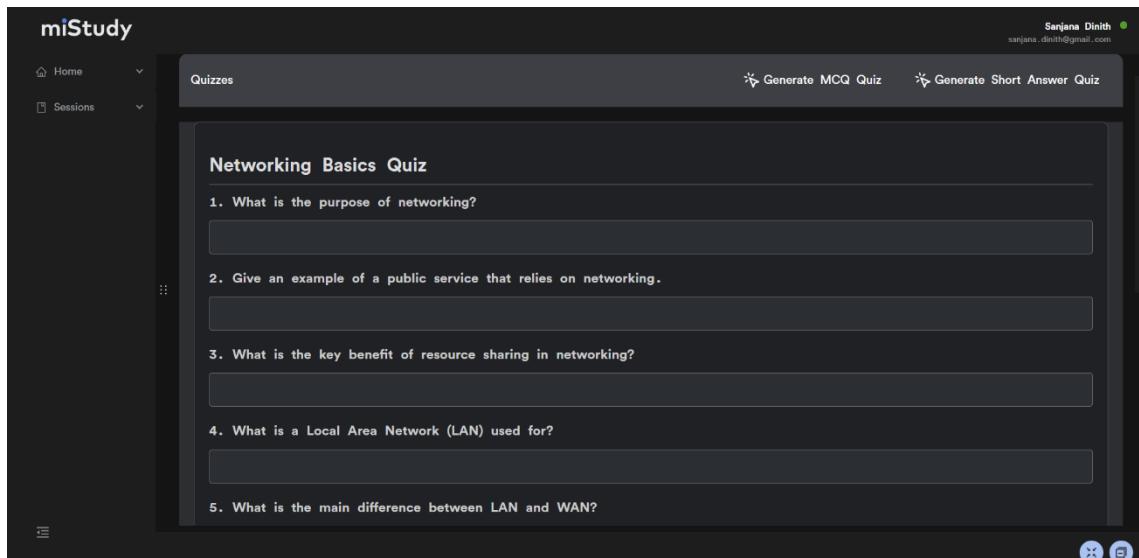


Figure 60 Generated SAQ Interface

4.3.2 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.

Feedback:

Quiz Performance Analysis:

Congratulations on completing the quiz! You scored 6 out of 10, which shows that you have a good foundational understanding of networking concepts. However, there are areas for improvement based on your incorrect answers.

Strengths:

- You were able to correctly identify several concepts related to networking, suggesting you grasp basic networking terminology and its applications.

Weaknesses:

- There seems to be a misunderstanding of some fundamental definitions and key benefits of networking. Your answers indicate potential confusion between different types of technology and concepts associated with networking.

Common Mistakes Identified:

1. **Misunderstanding Definitions:** For example, you defined networking as a type of software, which reflects a limited understanding of its broader purpose—connecting devices for communication and resource sharing.
2. **Incorrect Perception of Benefits:** Confusing decreased efficiency as a benefit instead of recognizing improved resource sharing indicates a misunderstanding of the fundamental advantages that networking provides.
3. **Historical Context:** You mentioned the invention of the computer as an early innovation in networking, while the correct answer referred to Morse Code and switching stations, reflecting a need to revisit the historical timeline of networking technologies.

Improvement Suggestions:

1. **Study Definitions:** Focus on the foundational definitions of key networking terms. Resources such as textbooks, reputable websites, or online courses can provide valuable context and clarity.
2. **Review Key Benefits:** Familiarize yourself with the advantages of networking, such as resource sharing, cost reduction, and improved communication. Make a list of common benefits and their explanations.
3. **Understand Historical Developments:** Research the history of networking technologies. Knowing key milestones, like the invention of Morse Code and switching stations, will help you appreciate how networking has evolved.
4. **Practice with Sample Questions:** Engage in practice quizzes or flashcards that specifically address networking concepts to strengthen your recall and understanding.

Actionable Tips for Future Quizzes:

- **Create a Study Plan:** Dedicate time each week to review networking concepts and test your knowledge with quizzes that focus on both terminology and practical applications.
- **Join Study Groups:** Collaborating with peers can clarify misunderstandings and provide different perspectives on complex concepts.
- **Utilize Multimedia Resources:** Videos, podcasts, and infographics can make learning about networking more engaging and memorable.

With focused study and practice, you will undoubtedly improve your understanding of networking. Keep up the good work, and best of luck on your next quiz!

Figure 61 Feedback Displayed After Answer Submission

The screenshot shows a mobile application interface for 'miStudy'. At the top, there are navigation tabs for 'Home' and 'Sessions'. On the right, a user profile is visible with the name 'Sanjana Dinit' and an email address 'sanjana.dinit@gmail.com'. The main content area is titled 'Networking Basics Quiz'. It displays four questions with their respective answers and feedback:

1. What is the purpose of networking?
The purpose of networking is to build connections, exchange information, and create opportunities for collaboration, career growth, and personal development.
Correct!
Correct answer: To connect devices for communication, resource sharing, and data exchange.
2. Give an example of a public service that relies on networking.
test
Incorrect.
Correct answer: Online government portals.
3. What is the key benefit of resource sharing in networking?
test
Incorrect.
Correct answer: Reduce costs and improve efficiency.
4. What is a Local Area Network (LAN) used for?
test
Incorrect.

Figure 62 Semantic Search Result (Weaviate Match)

4.3.3 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.

```
_id: ObjectId('67db5ca13450484c69869006')
title : "Introduction to Networking Quiz"
documentID : ObjectId('67db27e380434636f0b7ff64')
userID : ObjectId('6741df922c3f8d8d9bc6ed02')
difficultyLevel : 50
questions : Array (10)
score : 3
createdAt : 2025-03-20T00:09:05.469+00:00
updatedAt : 2025-03-20T00:10:10.698+00:00
__v : 0

_id: ObjectId('67db5d123450484c698691da')
title : "Introduction to Networking"
documentID : ObjectId('67db27e380434636f0b7ff64')
userID : ObjectId('6741df922c3f8d8d9bc6ed02')
difficultyLevel : 30
questions : Array (10)
score : 0
createdAt : 2025-03-20T00:10:58.503+00:00
updatedAt : 2025-03-20T00:10:58.503+00:00
__v : 0
```

Default Difficulty Level

Figure 63 Difficulty 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 ($\geq 75\%$): 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.

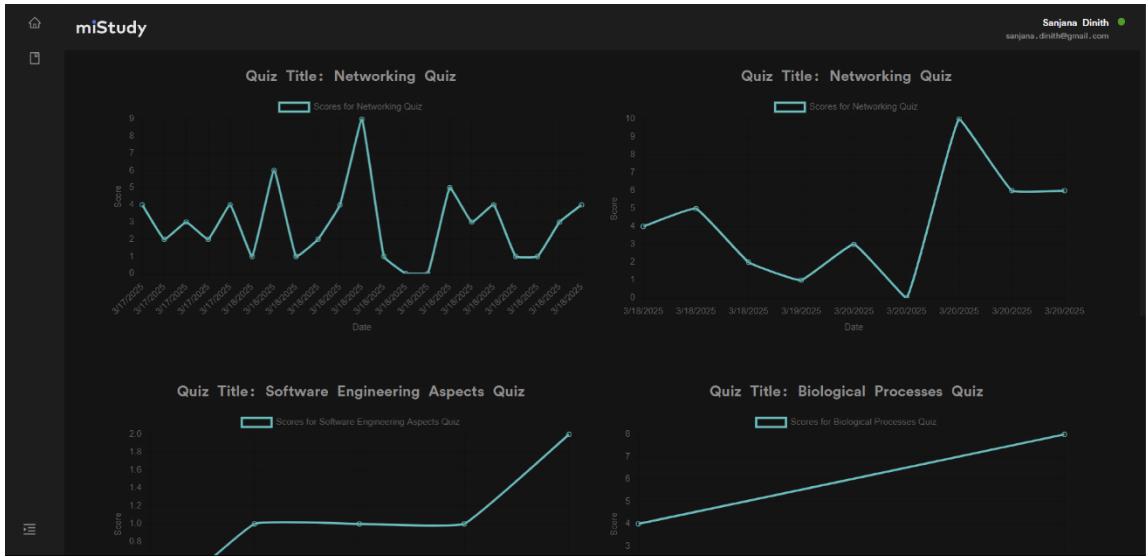


Figure 64 Quiz Attempt History and Performance Graph

4.4 Adaptive Mind Map Generation Component

To enhance personalized adaptive mind map generation, various models and techniques were explored and evaluated. The implementation involved integrating RAG, LLMs, and external knowledge retrieval techniques to create adaptive, user-centered mind maps. The performance was assessed based on accuracy, adaptability, and efficiency in generating contextually relevant learning maps.

Table shows the details of the evaluated LLM models and the benchmark scores which the respective models have produced, along with the cost of the usage of the model.

Table 34 Model Comparison

Model	GPT-3.5 Turbo	GPT-4	GPT-4 Turbo	GPT-4o Mini	LLaMA 3.1 8B Instruct	Gemini Flash
Provider	OpenAI	OpenAI	OpenAI	OpenAI	Meta	Google
Input Context Window	4,096 tokens	8K / 32K tokens (variant)	128K tokens	128K tokens	128K tokens	1M tokens
Max Output Tokens	4,096 tokens	4,096 tokens	4,096 tokens	16.4K tokens	2,048 tokens	8,192 tokens
Release Date	Nov 28, 2022	Mar 14, 2023	Apr 9, 2024	Jul 18, 2024	Jul 23, 2024	May 14, 2024
Knowledge Cutoff	Sep 2021	Sep 2023	Dec 2023	Oct 2023	Dec 2023	Nov 2023
Open Source	No	No	No	No	Yes	No
Input Cost	\$0.50	\$3.00	\$10.00	\$0.15	\$0.09	\$0.13
Output Cost	\$1.50	\$6.00	\$30.00	\$0.60	\$0.09	\$0.38
MMLU Benchmark	70.0	86.4	86.7	82.0	66.7	78.9

HumanEval Benchmark	60.3	85.0	88.2	87.2	34.8	74.3
HellaSwag Benchmark	85.5	94.5	95.3	87.2	80.01	86.5

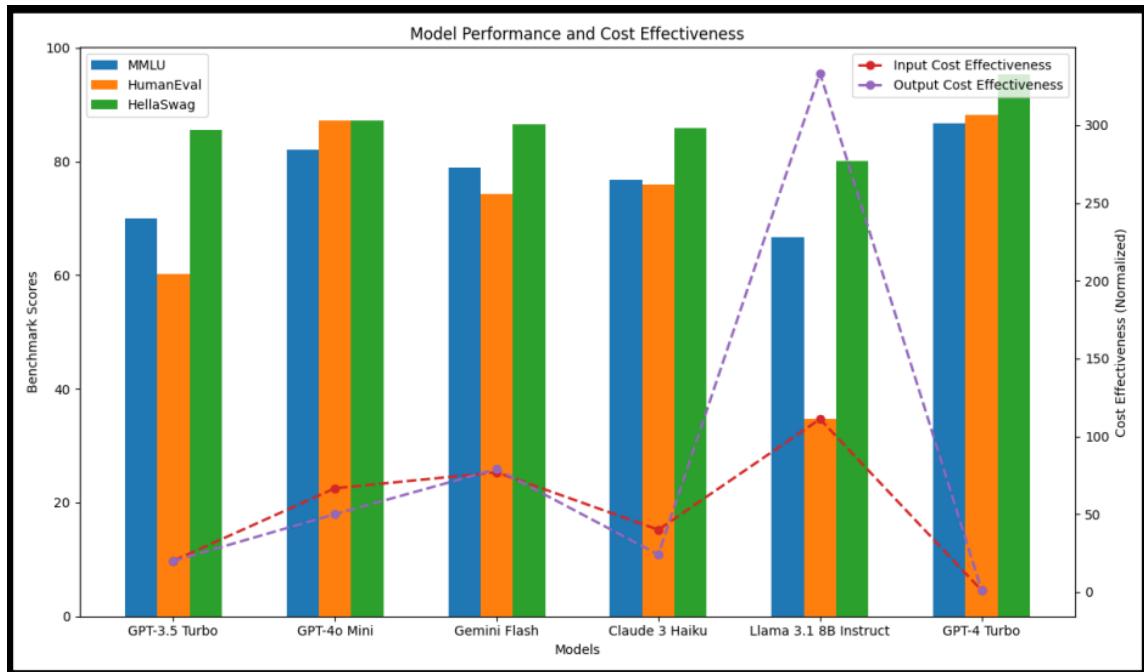


Figure 65 Model Performance with Llama 3.1 8B

This derived bar chart shows that the Llama 3.1 8B Instruct LLM is very cost effective, yet the performance is drastically low.

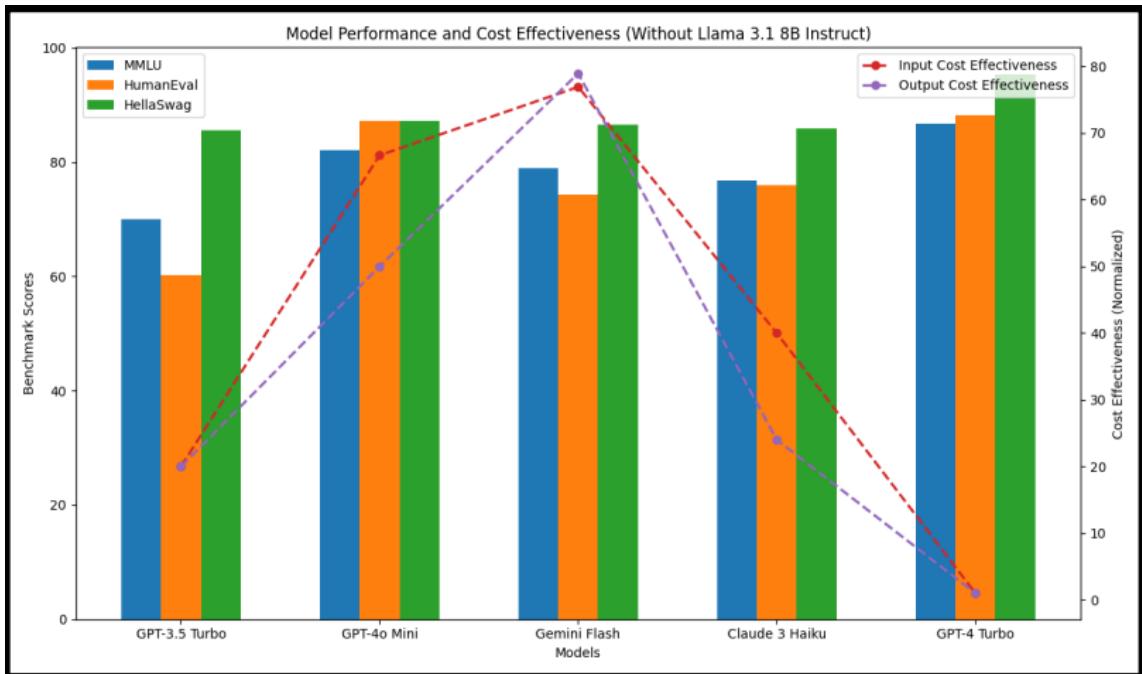


Figure 66 Model Performance without Llama 3.1 8B

The derived bar graph visualizes the cost-effectiveness and performance of various large language models (LLMs). While GPT-4 Turbo demonstrates the highest benchmark performance across MMLU, HumanEval, and HellaSwag, it is also the most expensive model, offering very poor cost-effectiveness. On the other hand, Gemini Flash is the most cost-effective model, especially for input and output processing, but it slightly underperforms in benchmark scores compared to other top-tier models.

Models such as GPT-3.5 Turbo and Claude 3 Haiku offer a balance between cost and performance but still fall short either in accuracy or flexibility. GPT-4o Mini stands out as the most balanced model, providing high benchmark performance close to GPT-4 Turbo while maintaining substantially lower costs, making it ideal for real-time and large-scale tasks like mind map generation. For this reason, GPT-4o Mini is the most practical and efficient choice among the evaluated models.

4.4.1 Results of Prompt Engineering for Mind Map Generation

Prompt engineering played a vital role in shaping the responses of the AI model to align with the functional requirements of the adaptive mind map generator. Different prompting styles and templates were explored for various scenarios ranging from content-driven mind map creation to quiz-adaptive visualizations and question-based contextual mapping. The table below summarizes the techniques evaluated and applied.

Table 35 Prompting Techniques

Prompting Techniques	Application Scenario
Instruction-Based Prompting	PDF-based mind map generation, Quiz-based adaptive mind maps, Prompt-based mind map creation
Few-shot Prompting	PDF-based mind map generation ,Quiz-based adaptive mind maps
Zero-shot Prompting	Resource-enhanced mind map construction and description generations
Role Prompting	Resource-enhanced mind map construction and description generations
Chain of Thought Prompting	Question validation prompts

4.4.1.1 Few-shot Prompting and Instruction-based Prompting for pdf-based mind map generation

Few-shot prompting, along with instruction-based prompting, was utilized for converting structured PDF content into a hierarchical mind map in the Mermaid.js format. The prompt provided the following detailed instructions:

- Break down long paragraphs into shorter, readable sentences
- Maintain semantic clarity and structural integrity
- Remove special characters and nested brackets
- Ensure only a single root node with relevant Font Awesome icons

The results obtained using the main prompt, `MAIN_MIND_MAP_GENERATOR_PROMPT`, showed high consistency in generating clear and visually structured mind maps with high semantic alignment. Below is the structure of the primary prompt used:

```
const MAIN_MIND_MAP_GENERATOR_PROMPT = ` You are a mind map generator. Convert the following JSON into Mermaid.js mind map input format.
```

For each object or array, create corresponding nodes and sub-nodes.

1. If there are long paragraphs inside the "paragraph" tags, split them into simple sentences for better readability.
2. Remove any unwanted characters like special symbols or redundant information.
3. For each paragraph, ensure that it's concise and doesn't exceed a few sentences, keeping it readable.

4. Maintain the hierarchical structure from the JSON as nodes and sub-nodes in the mind map.

This is the JSON data you need to convert to mermaid below format.: JSON:
 \${JSON.stringify(extractedSections)}

Bellow I'll provide a template how to break the things into nodes. Use this template to generate the mermaid js input format. Don't add any of your thoughts to this.

return the template only. Don't add any text or any special characters to it. Just give me the output like below format.

The format is below. Use the correct icons based on the texts. For icons only, you have to use the brackets like in below format.

mindmap

root("Artificial Intelligence")

What is AI?

::icon(fa fa-users)

("Artificial Intelligence refers to the simulation of human intelligence.")

("AI is programmed to think like humans and mimic their actions.")

("The goal of AI is to enable machines to perform tasks.")

("Tasks include visual perception, speech recognition, decision-making, and language translation.")

Applications of AI

::icon(fa fa-play)

("AI is widely used in various industries.")

("In healthcare, AI helps diagnose diseases and predict patient outcomes.")

("It also helps develop personalized treatment plans.")

("In finance, AI is used for fraud detection.")

("AI also supports algorithmic trading and customer service automation.")

Do not include any brackets inside parentheses in the formatted output. This means that words or phrases within parentheses, such as (LAN), should be removed while keeping the rest of the text intact.

For example, if the original mind map node is:

(Local Area Network (LAN) connects devices within a limited area.)

The formatted output should be:

(Local Area Network connects devices within a limited area.)

Only one root, use free Font Awesome icons, and follow node types "[", "("". No need to use "mermaid", "\``\`", or "graph TD". Respond only with code and syntax. `;

```

1 mindmap
2 root(Networking)
3   Introduction to Networking
4     ::icon{fa fa-network-wired}
5     What is Networking?
6       (Networking connects devices e.g., computers, printers for communication.)
7       (It supports systems from simple home networks to global infrastructures.)
8       (Examples include home devices connected to Wi-Fi and corporate LANs.)
9     Key Benefits of Networking
10    Types of Computer Networks
11      ::icon{fa fa-sitemap}
12      Local Area Network
13        Centralized Management
14          ::icon{fa fa-cogs}
15          (Admins can monitor traffic, deploy updates, and enforce security.)
16      History of Networking
17        ::icon{fa fa-history}
18        Telegraph and Telephone
19          (Telegraph enabled coded message transmission Morse Code.)
20          (Telephone allowed real-time voice communication over long distances.)
21        Early Innovations
22          (Morse Code & Switching Stations streamlined long-distance communication.)
23        Long-Distance Communication
24          (Telegraph Lines & Submarine Cables connected cities and continents.)
25        Packet Switching
26          (Introduced in the 1960s with ARPANET.)
27          (Improved efficiency and reliability, forming the foundation of modern internet.)
28      Network Concepts
29        ::icon{fa fa-th}
30        Introduction to LAN
31          (Definition connects devices within a limited area like a home or office.)
32          (Components include switches, routers, and devices.)
33        Shared Resources
34          (Shared Drives enable secure access and collaboration.)
35          (Benefits include real-time updates and enhanced team productivity.)
36        Wide Area Network
37          (Definition connects multiple LANs across large geographic areas.)
38          (Internet connects billions of devices globally.)
39        WAN Business Use
40          (Multinational Operations enable centralized management.)
41          (Data Sharing allows real-time synchronization and cloud integration.)
42      Network Topologies
43        ::icon{fa fa-sitemap}
44        What is Network Topology?
45          (Definition is the layout of devices and connections in a network.)
46          (Impact affects network speed and troubleshooting efficiency.)
47        Common Topologies
48          (Star Topology has centralized control.)
49          Advantages
50            (Centralized control simplifies management.)
51            (Failure of individual devices doesn't disrupt the network.)
52            (Highly scalable for growing networks.)
53          Key Features of Star Topology
54            (Centralized Devices ensure efficient data routing.)
55            (Scalability allows new devices to be added seamlessly.)

```

Figure 67 Main Mind-Map Generation Output Format



Figure 68 Main Mind-Map Generation Output

4.4.1.2 Prompt Engineering Techniques Identified in Quiz based Mind Map Generation

Several prompt engineering techniques were tested in the context of quiz-based adaptive mind map generation. These techniques include:

1. Instruction-based Prompting
 - The prompt clearly instructs the AI to parse quiz-related data and generate a mind map using a predefined structure. It ensures clarity and consistency in the output.
2. Structured Output Prompting
 - This technique guides the AI to output mind maps in a strict structure, ensuring no additional text or characters are included in the result. The model follows specific rules on node types and icon usage.
3. Zero-shot Prompting
 - Zero-shot prompting allows the AI to handle new and unseen scenarios. In this case, the model was asked to generate mind maps based on incorrect quiz answers and explanations, without having seen specific examples beforehand. This approach tested the AI's ability to generalize.

```
const QUIZ_BASED_MIND_MAP_PROMPT= `You are a mind map generator.  
Convert the following highlighted data into Mermaid.js mind map input format.
```

For each object or array, create corresponding nodes and sub-nodes.

1. If there are long paragraphs inside the "paragraph" tags, split them into simple sentences for better readability.
2. Remove any unwanted characters like special symbols or redundant information.

3. For each paragraph, ensure that it's concise and doesn't exceed a few sentences, keeping it readable.

4. Maintain the hierarchical structure from the JSON as nodes and sub-nodes in the mind map.

This is the data you need to more highlight the things regarding to these topics: JSON:
\${JSON.stringify(incorrectAnswersForQuizes)}

For highlighted topics please provide some additional explanations.

For incorrect answers you can provide any additional explanations regarding to the concepts

Bellow I'll provide a template how to break the things into nodes. Use this template to generate the mermaid js input format. Don't add any of your thoughts to this. Return the template only. Don't add any special characters to it. Just give me the output like below format.

The format is below. Use the correct icons based on the texts. For icons only you have to use the brackets like in below format.

mindmap

root("Artificial Intelligence")

What is AI?

::icon(fa fa-users)

("Artificial Intelligence refers to the simulation of human intelligence.")

("AI is programmed to think like humans and mimic their actions.")

("The goal of AI is to enable machines to perform tasks.")

("Tasks include visual perception, speech recognition, decision-making, and language translation.")

Applications of AI

::icon(fa fa-play)

("AI is widely used in various industries.")

("In healthcare, AI helps diagnose diseases and predict patient outcomes.")

("It also helps develop personalized treatment plans.")

("In finance, AI is used for fraud detection.")

("AI also supports algorithmic trading and customer service automation.")

Do not include any brackets inside parentheses in the formatted output. This means that words or phrases within parentheses, such as (LAN), should be removed while keeping the rest of the text intact.

For example, if the original mind map node is:

(Local Area Network (LAN) connects devices within a limited area.)

The formatted output should be:

(Local Area Network connects devices within a limited area.)

Only one root, use free Font Awesome icons, and follow node types "[", "("". No need to use "mermaid", "\````", or "graph TD". Respond only with code and syntax. `;

```
1 mindmap
2   root("Networking Concepts")
3     Concept of Networking
4       ::icon("fa fa-network-wired")
5       ("Networking involves connecting computers to share resources.")
6       ("It enables communication and data exchange between devices.")
7     Functional Role of Routers
8       ::icon("fa fa-route")
9       ("Routers direct data traffic between networks.")
10      ("They determine the best path for data packets.")
11      ("Routers connect different networks, including LANs and WANs.")
12     Importance of WAN in Multinational Operations
13       ::icon("fa fa-globe")
14       ("WANs connect multiple LANs across large distances.")
15       ("They facilitate communication between global offices.")
16       ("WANs support data sharing and collaboration.")
17     Benefits of Star Topology
18       ::icon("fa fa-sitemap")
19       ("Star topology offers easy management and troubleshooting.")
20       ("If one connection fails, it doesn't affect the entire network.")
21       ("It allows for easy addition of new devices.")
22     Solving Slow Data Exchange in LAN
23       ::icon("fa fa-tachometer-alt")
24       ("Upgrading network hardware can improve speed.")
25       ("Implementing Quality of Service (QoS) can prioritize traffic.")
26       ("Reducing the number of devices on the network can help.")
27     Scenarios Illustrating WAN Use
28       ::icon("fa fa-signal")
29       ("Connecting branch offices in different cities.")
30       ("Supporting remote work for employees in various locations.")
31       ("Enabling cloud services access for multiple users.")
32     Purpose of Hubs and Switches in Star Topology
33       ::icon("fa fa-plug")
34       ("Hubs connect multiple devices in a network.")
35       ("Switches intelligently direct data to specific devices.")
36       ("Both enhance communication efficiency in star topology.")
```

Figure 69 Quiz based Mind-Map Generation Output Format

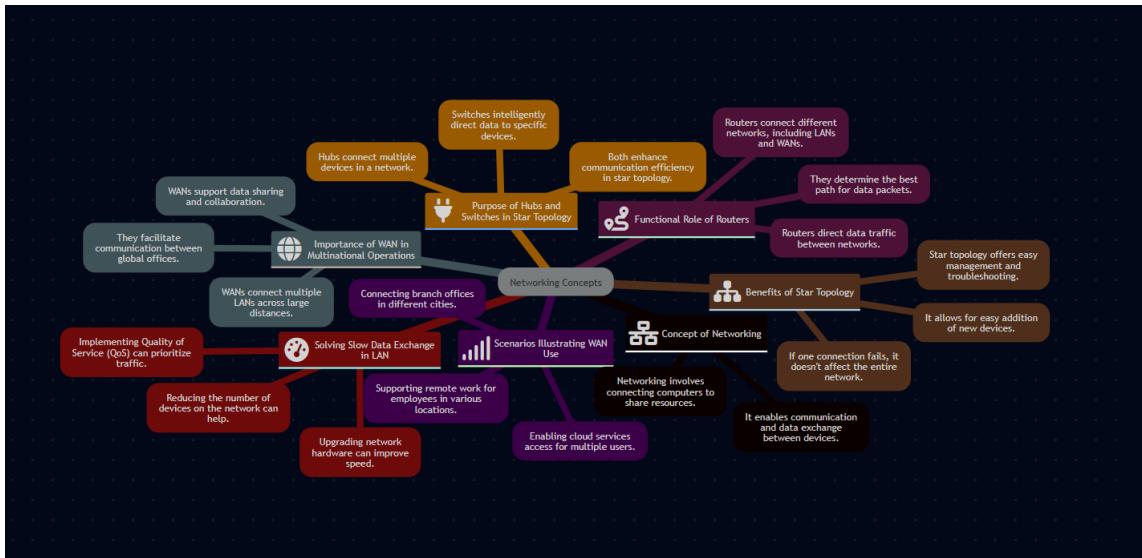


Figure 70 Quiz based Mind-Map Generation Output

4.4.1.3 Instruction-based Prompting and Role-Prompting & Chain of thought Prompting for question validation

In this technique, the model is given clear instructions defining its role, which helps in determining valid and invalid responses. For example, in a quiz-based learning scenario, the model's role as a "study assistant" guides it to generate mind maps based on the user's input. With Chain of Thought (CoT) prompting, the model first analyzes the user's question, checking if it relates to the content of the provided PDF. If the question is relevant, the model answers in detail; if it's unrelated, it responds with "Invalid." This approach ensures accurate and context-aware responses while maintaining focus on the intended learning material.

QUESTION_VALIDATION: 'You are a study assistant helping with questions and creating mind maps based on those questions.'

If the user asks a question unrelated to any learning environment (e.g., general knowledge or unrelated topics), you should not create the mind maps.

You should respond if the user's question is based on the content of the provided PDF.

Study Assistant Guidelines:

Topic Relevance (PDF Content):

If the user's question relates to the content of the provided PDF, you must answer the question, even if the exact details are not explicitly mentioned in the PDF. For example, if the PDF is about "software security aspects" and the user asks, "What are the advantages of considering software security aspects in an application?" even if the specific advantages are not in the PDF, your answer should still be based on the overall topic of software security aspects.

Answering Based on Provided PDF:

You should answer all questions related to the content of the PDF in a detailed and comprehensive manner. If the user asks for further clarification or details, such as asking to elaborate on specific topics like "security aspects" or requesting a breakdown of information, you should provide a thorough explanation.

For example:

If the question is, "I want to know more about security aspects in a detailed way," you should provide an in-depth explanation with ample descriptions and related information.

If the question is, "Give me a summary of software aspects," break down the summary into its core components and explain each part comprehensively.

Invalid Questions:

If the user asks a question unrelated to the content of the provided PDF, treat it as invalid. This includes questions on general knowledge, personal preferences, or unrelated topics.

For instance, questions like "What is the capital of France?" or "How do you cook spaghetti?" should be marked as invalid.

Response Handling:

If the user's question is valid (i.e., related to the content of the PDF or a related topic), respond with "Valid."

If the user's question is invalid (i.e., not related to the content of the PDF or a relevant topic), respond with "Invalid."

4.4.1.4 Few-shot Prompting and Instruction-based Prompting for simple mind map generation

This is the main technique used in this prompt.

It gives multiple concrete examples of expected input-output behavior:

- Brainstorming mind map
- Text summary mind map
- Cause and effects mind map

These examples show structure, formatting, icon usage, and node hierarchy.

Specific instructions are included, such as:

- "*Only one root*"
- "*Use free Font Awesome icons*"
- "*Follow node types '[' , '()*"
- "*No need to use 'mermaid', '```', or 'graph TD'*"

SIMPLE_MINDMAP_GENERATOR_PROMPT:

`Create a mermaid mindmap based on user input like these examples:

brainstorming mindmap

mindmap

\t\root(("leisure activities weekend"))

\t\t\t["spend time with friends"]

\t\t\t::icon(fafa fa-users)

\t\t\t("action activities")

\t\t\t::icon(fafa fa-play)

\t\t\t("dancing at night club")

\t\t\t("going to a restaurant")

\t\t\t("go to the theater")

```
\t\t\t["spend time your self"]
```

```
\t\t\t::icon(fa fa-user)
```

```
\t\t\t("meditation")
```

```
\t\t\t::icon(fa fa-om)
```

```
\t\t\t("take a sunbath ☀\\")
```

```
\t\t\t("reading a book")
```

```
\t\t\t::icon(fa fa-book)
```

text summary mindmap:

Barack Obama (born August 4, 1961) is an American politician who served as the 44th president of the United States from 2009 to 2017. A member of the Democratic Party, he was the first African-American president of the United States.

mindmap

```
\troot("Barack Obama")
```

```
\t\t("Born August 4, 1961")
```

```
\t\t::icon(fa fa-baby-carriage)
```

```
\t\t("American Politician")
```

```
\t\t::icon(fa fa-flag)
```

```
\t\t\t("44th President of the United States")
```

\t\t\t\t("2009 - 2017")
\t("Democratic Party")
\t::icon(fa fa-democrat)
\t("First African-American President")

cause and effects mindmap:

mindmap

\root("Landlord sells apartment")
\t::icon(fa fa-sell)
\t("Renter must be notified of sale")
\t::icon(fa fa-envelope)
\t("Tenants may feel some uncertainty")
\t::icon(fa fa-question-circle)
\t("Notice periods must be observed")
\t::icon(fa fa-calendar)
\t("Landlord can submit notice of termination for personal use")
\t::icon(fa fa-home)
\t("Tenant has to look for a new apartment")

```

\t\t\t::icon(fa fa-search)

\t("New owner")

\t::icon(fa fa-user)

\t\t("New owner takes over existing rental agreement")

\t\t::icon(fa fa-file-contract)

\t\t("Tenant keeps previous apartment")

\t\t::icon(fa fa-handshake)

\t\t("New owner terminates newly concluded lease")

\t\t::icon(fa fa-ban)

\t\t("Tenant has to look for a new apartment")

\t\t::icon(fa fa-search)

Only one root, use free Font Awesome icons, and follow node types "[", "(". No need
to use "mermaid", "\```, or "graph TD". Respond only with code and syntax. `};
```

```

1 | mindmap
2 | root("Advantages of Networking")
3 |     ["Career Opportunities"]
4 |     ::icon("fa fa-briefcase")
5 |     ["Access to Resources"]
6 |     ::icon("fa fa-toolbox")
7 |     ["Knowledge Sharing"]
8 |     ::icon("fa fa-book-open")
9 |     ["Building Relationships"]
10 |    ::icon("fa fa-users")
11 |    ["Increased Visibility"]
12 |    ::icon("fa fa-eye")
13 |    ["Support System"]
14 |    ::icon("fa fa-headset")
15 |    ["Collaboration"]
16 |    ::icon("fa fa-handshake")
17 |    ["Mentorship Opportunities"]
18 |    ::icon("fa fa-user-plus")
19 |    ["Enhanced Skills"]
20 |    ::icon("fa fa-graduation-cap")
21 |    ["Personal Growth"]
22 |    ::icon("fa fa-seedling")

```

Figure 71 Simple Mind-Map Generator Output Format

The screenshot shows the miStudy application interface. On the left, there is a sidebar with 'Home' and 'Sessions'. The main area displays a PDF titled 'Introduction to Computer Networking3.pdf' in 'Read Mode'. The content of the PDF includes sections like '1. Introduction to Networking' and '1.1 What is Networking'. On the right, there is a 'Prompt Minds' window titled 'Advantages of Networking'. This window shows a central node 'Advantages of Networking' connected to eight peripheral nodes: 'Access to Resources', 'Support System', 'Building Relationships', 'Career Opportunities', 'Increased Visibility', 'Knowledge Sharing', 'Collaboration', and 'Mentorship Opportunities'. Each node has a corresponding icon. Below the mind map, there is a text input field containing 'Give me 10 advantages of networking'.

Figure 72 Simple Mind-Map Generator Output

4.4.1.5 Zero-shot Prompting and Instruction based Prompting for resource generations

Zero-shot prompting allows the AI to handle tasks it has not been explicitly trained on, such as listing useful resources for a specific topic. Instruction-based prompts guide the AI on how to behave and approach the task, even without prior examples.

RESOURCE_MAPPER_PROMPT: "Provide a list of useful resources with links for this topic. Please provide valid links.",

SIMPLE_DESCRIPTION_PROMPT: "You are a helpful, highly skilled assistant that simplifies complex concepts into clear, concise, and easy-to-understand explanations. Provide brief yet informative descriptions that enhance comprehension for any audience."

Description output, Node Text: Networking

Networking refers to the process of establishing and nurturing professional relationships that allow individuals and organizations to share information, resources, and support. It often involves connecting with others in the same industry or field, attending events, and utilizing social media platforms to build a network of contacts that can lead to new opportunities, collaborations, and career advancement.

collaborations, and career advancement.

5 SUMMARY

5.1 Rajapaksha R.M.S.D – IT21251900

The Study Techniques and Cognitive Load Management component plays a critical role in promoting sustained learner focus and optimizing productivity through intelligent session structuring. It introduces a Dynamic Session Break Scheduler, a personalized tool built with Angular that adapts break intervals in real-time based on a learner's focus capacity, preferred break duration, and environmental noise levels.

The component utilizes a cognitive fatigue simulation model, where an internal energyLevel variable diminishes over time to reflect mental exhaustion. Breaks are progressively adjusted in both frequency and duration, helping users maintain optimal performance throughout study sessions. In addition, the algorithm integrates a noise-aware mechanism by subscribing to updates from a NoiseLevelService, which alters session timing dynamically in response to high or fluctuating background noise.

By calculating personalized work-break cycles, this component fosters healthy study habits and prevents cognitive overload. Learners interact with an intuitive form-based UI to configure session parameters, while real-time feedback and session visualization offer enhanced self-awareness and engagement. This system empowers users with adaptive time management, making it a core feature of the LMS's cognitive load management strategy.

5.2 Peiris A.R.D - IT21249266

Took full responsibility for designing and developing the AI Document Assistant component of the LMS, enabling students to interactively query content from uploaded PDF materials. Integrated advanced Large Language Models (LLMs) and implemented

retrieval-augmented generation (RAG) to ensure accurate and context-aware responses based on learning materials.

Focused on optimizing document parsing pipelines, enhancing response relevance, and ensuring smooth user interactions with the assistant through a clean and responsive interface. Also contributed significantly to testing, fine-tuning prompts, and integrating backend services to ensure seamless performance.

5.3 Monali G.M.N – IT21360428

Actively contributed to the design and development of the adaptive quiz module, with a focus on enhancing personalized learning through performance-based question delivery. Participated in early discussions to conceptualize how quiz results could influence mind map generation and overall learner feedback.

Worked on the backend logic to manage quiz data, including quiz creation, response tracking, and result evaluation. Integrated adaptive mechanisms that analyze users' performance to determine strengths and weaknesses, allowing the system to generate targeted follow-up quizzes.

5.4 Sri Samadhi L.A.S.S – IT21302862

Actively contributed to the design and development of the adaptive mind map generation module, Led the design and implementation of the adaptive mind map generation feature within the LMS system, to deliver personalized, dynamic mind maps based on user quiz performance and custom prompts. Conducted in-depth research to identify limitations in existing tools and built solutions to adapt content for different learner needs.

In addition, successfully implemented key frontend features including note cards, chapter view sections, and Google survey integrations to enhance interactivity and user engagement across the platform.

Actively participated in finalizing the final report, and made substantial contributions to the research paper writing process, especially in articulating the methodology and system functionalities clearly and effectively.

6 CONCLUSION

In conclusion, the development of the AI-powered adaptive Learning Management System (LMS) marks a significant advancement in the field of personalized education and digital learning environments. By integrating cutting-edge technologies such as Large Language Models (LLMs), prompt engineering, and modular microservices, the system successfully delivers a dynamic, learner-centric experience that addresses the core challenges in traditional learning platforms lack of personalization, engagement, and timely feedback.

The web-based LMS combines several powerful components, including adaptive mind map generation, an AI PDF assistant, Study techniques-based session management, and a structured chapter-wise note-taking interface. These components work synergistically to help students visualize knowledge, manage their time efficiently, and engage deeply with learning content tailored to their performance and preferences. The backend, powered by technologies like Node.js and cloud services including Azure and AWS, ensures scalability and performance, while the Angular frontend provides an intuitive and accessible interface for seamless user interaction.

The system's success lies in its ability to intelligently adapt to individual learning patterns through both manual and automated prompting strategies. The incorporation of various prompting techniques such as zero-shot, few-shot, chain-of-thought, and role-based prompting ensures that learning content remains relevant, accurate, and effective in addressing the learner's unique needs. This approach not only enhances comprehension and retention but also empowers students to take control of their own learning journey.

Looking ahead, future work will focus on expanding the LMS to support mobile platforms, integrating multilingual support, and enhancing real-time analytics for both learners and educators. Additionally, the system will explore the incorporation of voice-driven interactions and gamified learning features to further elevate engagement and interactivity. These enhancements will be driven by continuous user feedback, iterative development, and collaboration with academic professionals to ensure the LMS remains pedagogically sound and technologically relevant.

Through these collective innovations, the adaptive LMS stands to redefine how learning is delivered in academic institutions, professional training programs, and self-directed environments. It sets the stage for a future where education is not just accessible, but deeply personalized, interactive, and effective ultimately shaping more confident, capable, and self-motivated learners.

REFERENCES

- [1] F. Cirillo, *The Pomodoro Technique*, 2006.
- [2] Sweller, John; Ayres, Paul; Kalyuga, Slava, *Cognitive Load Theory*, New York: Springer, 2011.
- [3] Holmes, Wayne; Bialik, Maya; Fadel, Charles, *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning*, Center for Curriculum Redesign, 2019.
- [4] 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.
- [5] I. Learning Explorer, "The Science Behind the ScootPad Mastery-Based Adaptive Learning Platform," Learning Explorer, Inc., Campbell, California, USA, 2023.

- [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] 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.
- [8] Liu, K.; Lahoz, E. A., "Impact of Learning Styles on Students' Retention of Information," *International Journal of Education and Humanities*, vol. 17, no. 1, p. 207–220, 2024.
- [9] Smith, A.; Doe, J., "Collaborative AI-Enhanced Digital Mind-Mapping as a Tool for Inclusive Education," *BMC Psychology*, vol. 12, no. 1, p. 75, 2024.
- [10] Karabacak, M.; Ozkara, B. B.; Margetis, K.; Wintermark, M.; Bisdas, S., "Generative Artificial Intelligence in Education: Advancing Adaptive and Personalized Learning," *Journal of Medical Internet Research*, vol. 25, p. e48163, 2023.
- [11] Spriggs, K.; Lau, M. C.; Passi, K., "Personalizing Education through an Adaptive LMS with Integrated LLMs," arXiv, , 2025.
- [12] Jefferson Costales, Janice Abellana, Joel Gracia, Madhavi Devaraj, "A learning assessment applying Pomodoro technique as a productivity tool for online learning," in *Proceedings of the 13th International Conference on Education Technology and Computers*, 2021.
- [13] Danielle S. McNamara, Scott A. Crossley, Rod Roscoe, "Natural language processing in an intelligent writing strategy tutoring system," *Behavior Research Methods*, vol. 45, p. 499–515, 2013.
- [14] L. Kang, "Personalized Quiz Maker: Novel feature for Alby learning management system," 2024.
- [15] Robert Kudelić, Mirko Maleković, Alen Lovrenčić, "Mind map generator software," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 2012.

- [16] Mohamed Elhoseiny, Ahmed Elgammal, "Text to multi-level MindMaps: A novel method for hierarchical visual abstraction of natural language text," *Multimedia Tools and Applications*, vol. 75, p. 4217–4244, 2016.
- [17] T. B. Brown, B. Mann, N. Ryder and M. Subbiah, "Language Models are Few-Shot Learners," in *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*.
- [18] B. Lester, R. Al-Rfou and . N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning," *Association for Computational Linguistics*, vol. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021.
- [19] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.
- [20] Y. Deng, J. Cherian, K. Kumari, U. Comite, B. Gavurova and J. Popp, "Family and Academic Stress and Their Impact on Students' Depression Level and Academic Performance," *Frontiers in Psychiatry*, vol. 13, 2022.
- [21] F. Petroni, . T. Rocktäschel, S. Riedel, . P. Lewis, A. Bakhtin, Y. Wu and A. Miller, "Language Models as Knowledge Bases?," *Association for Computational Linguistics*, Vols. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.
- [22] Y. Gao, Y. Xiong, X. Gao and K. Jia, "Retrieval-Augmented Generation for Large Language Models: A Survey," 2023.
- [23] S. Freeman, S. L. Eddy, M. McDonough, M. Smith, N. Okoroafer, H. Jordt and M. P. Wenderoth, "Active Learning Increases Student Performance in Science, Engineering, and Mathematics," *Proceedings of the National Academy of Sciences*, 2014.
- [24] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goval, H. Kurller, M. Lewis, W.-t. Yih, S. Riedal and D. Kiela, "Retrieval-Augmented Generation for

Knowledge-Intensive NLP Tasks," in *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing System*.

- [25] D. Thüs, S. Malone and R. Brünken, "Exploring generative AI in higher education: a RAG system to enhance student engagement with scientific literature," *Frontiers in Psychology*, 2024.
- [26] C. H. Crouch and E. Mazur, "Peer Instruction: Ten years of experience and results," *American Journal of Physics*, vol. 69, no. 9, 2001.
- [27] Y. Shi, H. Yang, Y. Dou, Y. Zeng, "Effects of mind mapping-based instruction on student cognitive learning outcomes: A meta-analysis," *Asia Pacific Education Review*, vol. 24, no. 3, pp. 303-317, 2023.
- [28] Alexandra Okada, Teresa Connolly, "Designing Open Educational Resources through Knowledge Maps to Enhance Meaningful Learning," *International Journal of Learning Technology*, vol. 15, no. 7, p. 209–220, 2008.

APPENDIX

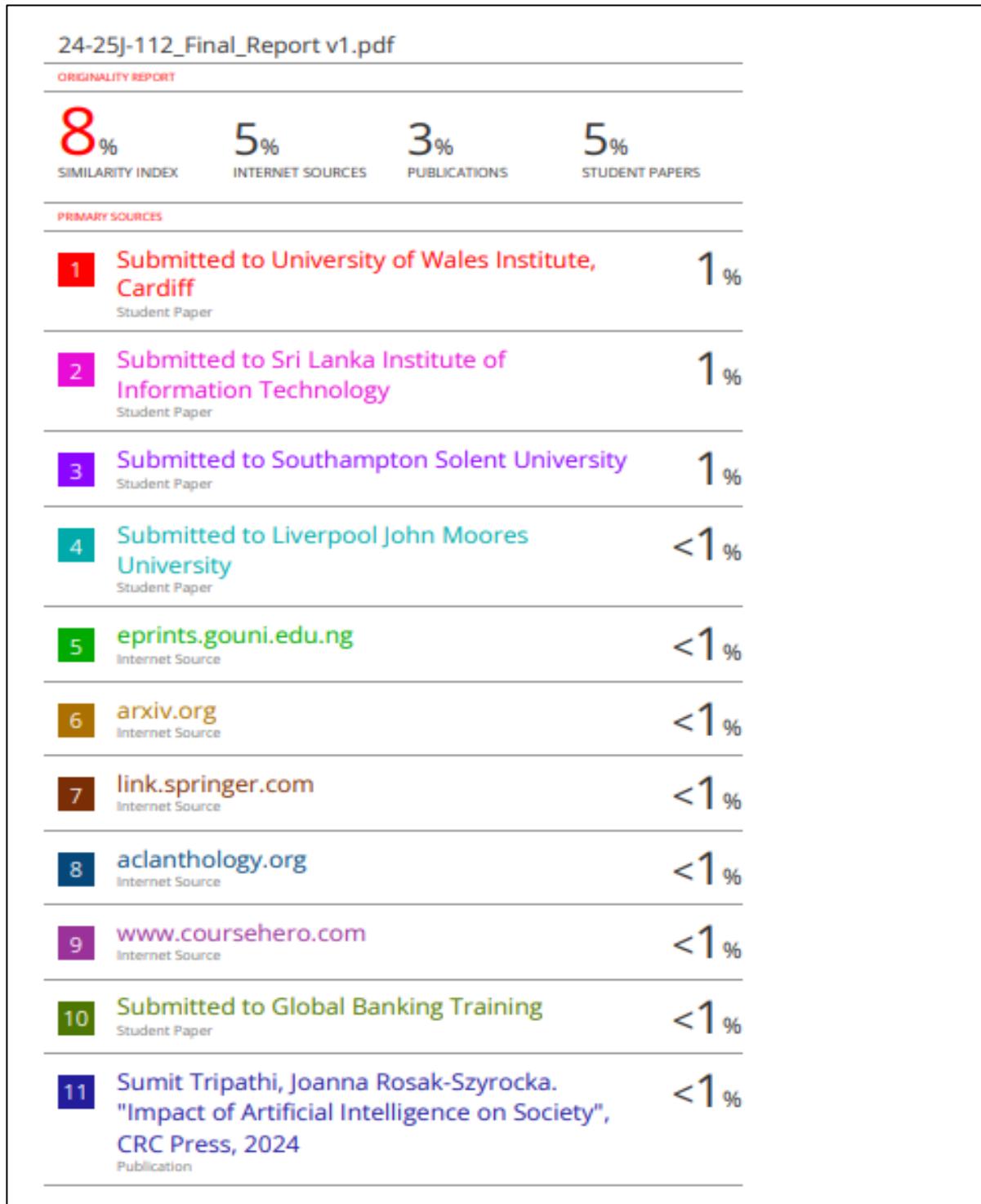


Figure 73 Plagiarism Report