Software Requirements Specification
For
TALIM - An AI Based Virtual Educational Assistant

Version 1.0

Abdullah Tahir Imaan Ibrar Laiba Atiq

Advisor: Dr Seemab Latif
Co-advisor: Dr Aimal Rextin

22nd October, 2024

Contents

Revision History

Name	Date	Reason For Changes	Version

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the software requirements for the development of TALIM, an AI-based virtual educational assistant. TALIM is a software developed for the Final Year Project titled "**TALIM: An AI-Based Virtual Educational Assistant.**" This document is on the first release of the product, **version 1.0**.

TALIM is designed to assist educators and students by automating several educational processes, such as generating quizzes and assignments along with their assessment rubrics, and providing interactive assistance for students' questions related to course material. The system processes various inputs, such as lecture notes and presentations in pdf format, and outputs context-aware responses, assignments, and quizzes. Applications of the system involve all the way from facilitating learning to making teaching easier for an educator.

This SRS defines the scope and core functionalities of TALIM, including input processing, quiz and assignment generation, and student-teacher interaction modules, powered by AI-driven natural language processing (NLP) techniques. The document serves as a fundamental reference for the development team to ensure that TALIM is built in alignment with the specified requirements.

1.2 Document Conventions

- **Default Font**
  - Font Style: Computer Modern
  - Font Size: 10pt (for article)
- **Section Heading Font**
  - Size: 12pt (when the base font size is 10pt)
  - Format: Bold
- **Subsection Heading Font**
  - Size: 11pt (when the base font size is 10pt)
  - Format: Bold
- **Subsubsection Heading Font**
  - Size: 10pt (when the base font size is 10pt)
  - Format: Bold and italic

1.3 Intended Audience and Reading Suggestions

Different individuals are likely to use this document for different purposes, as follows:

- **Developers (Team Members):**
  - Team Members are the primary developers responsible for the design and implementation of the TALIM system. They will use this document to understand the functional and non-functional requirements, as well as the system architecture.
- **Testers:**
  - Team Members will use this SRS to design test cases and verify the functionality, performance, and robustness of the TALIM system.
- **End Users (Teachers and Students):**
  - While the document is largely technical, educators and students can use it in understanding how the system would meet their needs, especially about features like automatic generation of quizzes, concept clarification, and assignment management.
- **Documentation Writers:**
  - They are the writers responsible for creating user guides or help documentation, relying on this SRS in developing understandable instructions about using the TALIM system and also system behavior.

For Readers

- **Developers:** Read Sections 2: Overall Description and 3: System Features to understand entire scope and functionalities.
- **Testers:** Read Sections 3: System Features and 4: Interface Requirements in order to assist with designing appropriate test cases.
- **End Users:** Read Section 3: System Features in order to understand how TALIM will be used in support of educational activities.
- **Documentation Writers:** Focus on Section 3: System Features and Section 4: External Interfaces to create detailed user documentation.

1.4 Product Scope

TALIM - AI-based educational assistant software will transform learning by integrating advanced technology into educational practices. It will automatically process lecture notes and presentations, creating a comprehensive knowledge base that students can ask questions on and receive instant clarifications. The

software would further automate the generation of quizzes and assignments, streamlining the process for educators to mark during assessment. Making such an investment will back corporate objectives of innovation and access to education, enhancing the strategies for increased engagement of the students, better learning results, and reducing administrative burdens on teachers. The organization intends to lead the way in educational technology space by moving forward the mission of quality education, making it more effective and accessible for all learners.

2 Overall Description

2.1 Product Perspective

The educational assistant is an AI-based product developed from scratch as a standalone, self-contained one for modern educational systems. In no case will it replace any existing system, but rather bring new abilities for students and teachers, from the start with capabilities based on AI-driven technologies for the support of interactive learning and automatically generated assessments. It takes lecture notes and presentations as input through a web-based portal and converts it into useful text for both students and teachers' tasking using modern deep learning techniques and leverage Large Language Models. The software will be a complete solution that captures lecture notes, presentations in an educative activity designed to prepare a knowledge base, quizzes, assignments, and feedback. Its modular architecture would allow it to be deployed in many educational institutions, making it really flexible and adaptable.

2.2 Product Functions

The AI-based educational assistant will provide the following major functions:

User Management:

- It offers role-based access, which will cater to the needs of both the teacher and the student, including their personal preferences.
- It will ensure secure login and data handling.

Lecture Content Ingestion:

It will give the ability to upload lecture notes and presentations (PDFs) into the system for content processing.

Knowledge Base Creation:

The system will automatically maintain a structured knowledge base from the uploaded content, which can be used for answering student queries.

Assessment Generation:

- It will generate quizzes in multiple versions and formats, including multiple-choice questions (MCQs), short-answer questions, and a mix of both based on teachers' requirements.
- It can generate theoretical or practical assignments based on lecture content along with grading rubrics for aiding in evaluation.

Interactive Query System:

- This will be a VTA module that will allow students to ask questions based on lecture content and retrieve relevant answers from the knowledge base.

Help and Documentation:

A link will be provided on the web portal that will allow the users to access any help they need in using the system effectively. A prompt engineering guide to efficiently prompt the system to generate the best results will also be provided.

2.3 User Classes and Characteristics

The primary users of the AI-based educational assistant can be categorized into the following classes:

Teachers/Instructors:

- **Frequency of Use:** Moderate to frequent, as it depends on the number of classes or subjects they teach.
- **Functions Used:** They will be uploading content (notes and lecture), generating automated quizzes and assignments along with rubrics for evaluation.
- **Technical Expertise:** Basic to moderate; they should be able to upload documents and use educational tools, although no advanced technical skills are required.
- **Security/Privilege Level:** High; they have administrative privileges, including the ability to manage content, generate assessments, and access student data.
- **Educational Level/Experience:** Experienced in subject matter but may have varying levels of familiarity with educational technology tools.

Students:

- **Frequency of Use:** Frequent, as they interact with the assistant regularly for queries related to lectures.
- **Functions Used:** Interactive query system.
- **Technical Expertise:** Basic; the system will be designed to be user-friendly and intuitive such that it requires minimal technical expertise.
- **Security/Privilege Level:** Moderate; students can access their own data (courses enrolled) and interact with the system for learning purposes.
- **Educational Level/Experience:** Undergraduate or graduate students, with varying levels of prior exposure to AI-based educational tools.

Administrators/System Managers:

- **Frequency of Use:** Infrequent, but it is essential for system setup and maintenance.
- **Functions Used:** System setup, user management, access control, and system troubleshooting.
- **Technical Expertise:** High; administrators are expected to have a strong technical background to manage and maintain the system.
- **Security/Privilege Level:** Very high; they have full control over the system, including user accounts, security policies, and content management.
- **Educational Level/Experience:** They must be experienced in IT and system administration with expertise in managing educational platforms.

The primary focus is on satisfying the needs of teachers and students, as they are the main users. Administrators play a crucial role in ensuring the system functions smoothly but interact with it less frequently.

2.4 Operating Environment

The software for the AI-based educational assistant system will operate in a versatile environment, designed to function across multiple platforms to accommodate diverse user needs. TALIM is designed to operate as a web application compatible with latest browsers and various operating systems, including Windows, macOS and Linux. The system will be built as a web application using Python with a MERN stack for recording user signups and data. The backend will consist of Python APIs running Deep Learning models (LLMs) on the data taken as input on the frontend. Furthermore,It is an extended goal that the software will peacefully coexist with standard applications used by educational institutions, including Learning Management Systems (LMS) Sandbox version.

2.5 Design and Implementation Constraints

The backend of the system consists of deep learning architectures that take the data provided by the teacher as input and uses it for creating assessments and providing context- aware response to students. This will be implemented using a RAG based approach while leveraging the power of SOTA LLMs available. This will be implemented using Python and its libraries. To ensure easier compatibility between the user interface and the backend server, we will be using the MERN Stack. Each component will have its own distinct functionality and purpose, hence ensuring modularity. All modules calling the deep learning algorithms will do so via well-defined APIs..

2.6 User Documentation

The following user documentation components will be delivered with the AI-based educational assistant:

- **User Manual:** A detailed guide explaining the system's features, navigation, and usage instructions for both students and teachers.
- **Prompt Engineering Guide:** A guide to help users understand how to get the best out of TALIM by providing the necessary input.

The documentation will be delivered in both PDF and HTML formats, ensuring accessibility across different platforms. All guides will be made available in English.

2.7 Assumptions and Dependencies

The following assumptions have been made regarding the operation and requirements of TALIM:

- **Internet Connectivity:** Users must have a secure and reliable internet connection to access features of the system.
- **Web-Based Interface:** TALIM will have a web-based interface, which users will access through web browsers. It is assumed that users have access to a compatible web browser.
- **Data Availability:** The system depends on the availability of well-formatted lecture notes and presentations from users. Poorly structured data may hinder the effectiveness of quiz generation and chatbot responses.

If any of these assumptions change or prove to be incorrect, the project scope and requirements might need to be revised.

3 External Interface Requirements

3.1 User Interfaces

The TALIM platform is mainly constructed with two roles in mind: Students and Teachers, and each of the user types should be able to interact with the system differently. Some key functions for each role are elaborated below:

3.1.1 Student Interface

- **Login/Signup**: A student can sign up for an account or login to their account by credentials. This ensures that access to the system as well as their signed-up courses are safe.
- **Enroll in Course**: After logging into the system, a student will be able to view the courses available and enroll in them. This makes the course available on their dashboard where they can engage with content of the course, assignments and quizzes.
- **View Content**: Once a student is enrolled in a course, they will be able to view the uploaded resources such as lecture notes, PDFs, and other kinds of documents to support the learning cycle.
- **Ask Query**: Students can ask questions related to the lecture content via an AI-powered Virtual Teaching Assistant, which aims to provide instant responses. If the system cannot resolve the query, it may be forwarded to the instructor for manual response.
- **Receive Answer**: The platform, through the Virtual Teaching Assistant (VTA), will respond to student queries, providing answers and additional resources as needed.

3.1.2 Teacher Interface

- **Login/Signup**: Teachers can sign up or log in to manage their courses, students, and teaching materials securely.
- **Create Course**: Any registered teacher can create their course on the platform by filling in the relevant course details. Students can enroll by using the enrollment code once the course is created.
- **Input Content**: he teachers can upload lecture contents and learning resources to their courses which he students can access.
- **Generate Assessment**: The platform uses a RAG based approach and LLMs to generate multiple versioned assessments according to the course contents.
- **Generate Rubrics**: Teachers are also able to use the system to generate the rubrics for the generated assessments in order to standardize the grading process.

3.2 Hardware Interfaces

TALIM AI is primarily a web-based application, and thus there is minimal direct interaction with hardware. However, some key characteristics include:

- **Server Hardware**: The platform will be hosted on cloud servers (such as AWS, Google Cloud, or a similar provider), requiring reliable and scalable hardware to support multiple concurrent users. These servers will handle the backend operations, including data storage, processing, and delivering responses to user requests.
- **User Devices**: The platform is accessible on any modern device that supports web browsers, including desktops, laptops, tablets, and smartphones. Thye users just need basic quirements such as havving internet connectivity and a browser which is compatible with modern web aspects such as HTML5, CSS3, and JavaScript for platform interaction.

Given the web-based nature of the application, no special hardware is required from users or administrators.

3.3 Software Interfaces

The TALIM AI platform connects with several software components to ensure a smooth system. These software components are as follows:

- **Operating Systems**: The platform is compatible with all modern operating systems that support standard web browsers, including:
  - Windows
  - macOS
  - Linux
  - iOS/Android (for mobile devices)
- **Web Browsers**: The platform supports modern web browsers, ensuring compatibility with:
  - Google Chrome
  - Mozilla Firefox
  - Microsoft Edge
  - Safari
  - Other modern browsers
- **Database**: The platform will utilize **MongoDB** as the backend database to store and manage data related to courses, users (teachers and students), assessments, and content uploads.

- **Backend Framework**: The server-side of the platform will be built using **Node.js** with **Express.js** as the framework, in order to have a smooth communication between the frontend and the database. Python APIs for deep learning models will also be used in the backend for connecting with the models.
- **Frontend Framework**: The frontend will be developed using **React.js**, which enables dynamic user interactions and a smooth user experience through a component-based architecture.
- **Authentication and Authorization**: The platform will integrate with **JWT (JSON Web Tokens)** for secure user authentication and authorization, in order to make sure that only registered authorized users can access any relevant content.

3.4 Communications Interfaces

The TALIM platform requires certain communication protocols to function effectively and maintain seamless interaction between different components:

- **HTTP/HTTPS**: The platform will rely on **HTTPS** for secure communication between the client (user's web browser) and the server. HTTPS ensures that all data transmitted, such as user credentials and content, is encrypted and protected against unauthorized access.
- **Email Notifications**: The platform will integrate with email services to send notifications to users (teachers and students).
- **REST APIs**: The platform will use **RESTful APIs** for communication between the frontend and backend, ensuring data exchange in JSON format. These APIs will handle all the communications between the backend and the frontend for authenticating users, answering user queries, displaying content, and other uses.
- **Network Protocols**: The platform will rely on standard **TCP/IP** protocols for reliable data transmission over the internet.

4 System Features

4.1 System Feature 1: User Authentication

4.1.1 Description and Priority

This feature allows students and teachers to create an account, log in, and authenticate using JWT. this is necessary in order to make sure that the platform and its contents are only accessed by the authorized users.
Priority: High

4.1.2 Stimulus/Response Sequences

- The user opens the login/signup page.
- The user enters their account credentials.
- The credentials are verified with the database by the system system verifies the credentials by checking the database.
- If credentials are valid, the user is authenticated and is redirected to their dashboard depending on if they a student or a teacher.
- If credentials are invalid, the system displays an error message.

4.1.3 Functional Requirements

REQ- 1: The system should allow users to register using their email address and a password.

REQ- 2: The system should authenticate users via JWT tokens and maintain session security.

REQ- 3: The system should restrict access to certain features based on the user role (student or teacher).

REQ- 4: The system should log out users and invalidate the JWT upon request.

4.2 System Feature 2: Course Creation and Enrollment

4.2.1 Description and Priority

This feature enables teachers to create courses and upload relevant materials, while students can enroll in these courses using the specific enrollment key. It is one of the main functionality of the system and all other features can only be performed if this works correctly.
Priority: Very High

4.2.2 Stimulus/Response Sequences

- A teacher logs in and goes to the course creation page.
- The teacher provides course details (name, description, materials) and creates a course.
- The system stores the course in the database and lists it in the available courses section for students, and provides the teachers with a specific enrollment key for the course.
- A student logs in, views available courses, and enrolls in a course if they have the enrollment key.
- The system confirms the enrollment and adds the course to the student's dashboard.

4.2.3 Functional Requirements

REQ- 1: The system should allow teachers to create a course by entering course details and uploading materials.

REQ- 2: The system should generate a specific enrollment key for each course that the teacher can give to their students.

REQ- 3: The system should display available courses for students to view and enroll.

REQ- 4: The system should allow students to enroll in courses and view course content.

REQ- 5: The system should ensure that only students with a valid enrollment key can enroll in a course.

REQ- 6: The system should store course data, including materials and student enrollments.

4.3 System Feature 3: VTA-based Q&A Sessions

4.3.1 Description and Priority

Students can ask questions to the Virtual Teaching Assistant (VTA), a RAG-based LLM system, which provides answers based on the course content. This feature automates the question-answering process and ensures after class student support to the student with alignment with course materials.
Priority: Very High

4.3.2 Stimulus/Response Sequences

- A student logs in and goes to the virtual teaching assistant in their enrolled course.
- The student submits a query related to the course content.
- The system processes the query and forwards it to the VTA.
- The VTA analyzes the query and provides a response based on the relevant course materials.
- The system displays the response to the student.

4.3.3 Functional Requirements

REQ- 1: The system should allow students to ask questions related to the course content.

REQ- 2: The system should have an integrated VTA to understand the student queries and give context-aware responses.

REQ- 3: The system should display the response provided by the VTA to the student.

REQ- 4: The system should maintain a log of student queries and VTA responses for reference.

4.4 System Feature 4: Automated Assessment Generation and Rubric Provision

4.4.1 Description and Priority

Teachers can use the system for automatic generation of multiple versioned assessments (quizzes, assignments) according to the course content. The system can also provide the rubrics or grading criteria to assist teachers in grading these generated assessments.
Priority: Very High

4.4.2 Stimulus/Response Sequences

- A teacher logs in and navigates to the specific course in their dashboard.
- The teacher instructs the system to generate an assessment and also tells about the number of versions they need of the assessment, along with the levels needed in grading rubrics.
- The system generates the assessment based on the course materials and the criteria defined by the teacher.
- The assessment is published for students to view.

- The system also generates and provides the teacher with a rubric according to the requirements to ensure fairer and effortless grading.

4.4.3 Functional Requirements

REQ- 1: The system should be able to generate assessments based on specific course contents and teacher requirements.

REQ- 2: The system should be able to generate multiple versions of the same assessment on similar difficulty level and same topics.

REQ- 3: The system should generate rubrics or grading criteria based on the content of the assessment to assist teachers in fairer grading.

REQ- 4: The system should make the assessments available to the students enrolled in the course as per willingness of the teacher.

4.5 System Feature 5: User Documentation

4.5.1 Description and Priority

The system will provide all the users with comprehensive user guide for the system along with a prompt engineering guide to help the users use the system effectively and take advantage of its full potential.
Priority: Normal

4.5.2 Stimulus/Response Sequences

- All users can access the user manual and prompt engineering guide from their dashboards after signing in.
- The user manual gives step-by-step instructions on using the system features.
- The prompt engineering guide guides the users to formulate queries to get the best possible outputs from the embedded LLM models.

4.5.3 Functional Requirements

REQ- 1: The system should provide a detailed user manual to all users that gives a complete map of features and explains how to navigate through the system, explaining each feature and how to access it in detail.

REQ- 2: The system should include a prompt engineering guide to help users optimize their prompts for interacting with the Virtual Teaching Assistant (VTA).

REQ- 3: Both the documents should be able to be downloaded from the dashboard.

REQ- 4: All user documentation should be available in English.

5 Non-Functional Requirements

5.1 Performance Requirements

The system must provide fast and reliable performance under different conditions:

- Response time for user queries should be under 10 seconds in normal usage scenarios.
- The system must handle the processing of large datasets such as lecture notes, presentations, and student data efficiently.
- During critical activities like quizzes and assignments, the maximum delay per request must not exceed 5 seconds to ensure smooth operation.

These requirements are crucial for maintaining a high-quality user experience and ensuring that students and faculty can interact with the system without delays.

5.2 Safety Requirements

To ensure safe operation, the system must prevent potential loss, damage, or harm:

- Unauthorized access to sensitive data must be avoided through secure user authentication mechanisms.
- In case of a system failure, no data corruption should occur, with regular backups in place to prevent loss of quiz results, lecture notes, or student submissions.
- The system should trigger security alerts and temporarily block access after multiple failed login attempts.
- External educational safety regulations must be followed, with relevant safety certifications ensuring compliance.

Safeguards include regular data backups, recovery mechanisms to restore lost data after failures, and robust access control measures.

5.3 Security Requirements

Security and privacy are key priorities for the system:

- User authentication must be implemented with multi-factor authentication for faculty members and secure single sign-on (SSO) for students to prevent unauthorized access.

- Data encryption should be enforced for all communications between users and the system, ensuring that personal data and course materials are protected during transmission.
- There should be mechanisms for logging and auditing all user activities, with special attention to anonymizing sensitive data to protect user privacy.
- The system must be designed to meet external security policies and certifications, ensuring it meets the highest standards for data integrity and protection.

5.4 Software Quality Attributes

The software should exhibit strong quality characteristics, which are essential for both developers and users:

- **Reliability**: The system should ensure 99.9% uptime, minimizing downtime and ensuring constant availability of educational resources.
- **Usability**: The user interface must be intuitive and easy to navigate, requiring minimal training for both students and instructors.
- **Maintainability**: The software should be easy to update and modify without causing disruption to ongoing classes or student work. Modular design will facilitate efficient bug fixes and upgrades.
- **Portability**: It should function consistently across various platforms, including desktop and mobile devices, without performance loss or functional discrepancies.
- **Testability**: Key features, such as user authentication, data retrieval, and content generation, should be easily testable with automated and manual tests ensuring system reliability.

5.5 Business Rules

The following operational principles will guide the system's usage:

- **Faculty roles and permissions**: Only authenticated faculty members can upload or modify course materials, create quizzes, assignments, and access aggregated student performance data.
- **Student access**: Students will only be able to access materials for the courses they are enrolled in and ask queries.

6 Other Requirements

The following business rules are to be implemented by the system:

- User privacy: Data specific to the user must be visible only to the user themselves. This is to be ensured using a robust and dependable signup and login system.
- Real-time processing: The data received must be sent for processing instantaneously. In case there is an overload on the backend server, the user is to be prompted to try again.

A Glossary

Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations.

RAG (Retrieval-Augmented Generation): A method that combines information retrieval with natural language generation, allowing the system to generate answers or responses based on retrieved content.

LLM (Large Language Model): A machine learning model trained on large amounts of text data to understand and generate human-like language. Examples include GPT-3.5, LLAMA, and Gemini.

Virtual Teaching Assistant (VTA): A digital assistant powered by AI and large language models, designed to answer student queries related to course content based on course contents.

Sequence Diagram: A type of UML (Unified Modeling Language) diagram that shows how objects in a system interact in a particular sequence to perform a function or achieve a goal.

Data Flow Diagram (DFD): A graphical representation of the flow of data within a system, showing how data moves from input to processing and storage, and finally to output.

MERN Stack: A web development stack consisting of MongoDB, Express.js, React.js, and Node.js, used for building dynamic web applications.

Prompt Engineering: The process of designing input prompts that guide AI models (like LLMs) to produce the desired output, especially when interacting with conversational AI systems.

B Workflow Diagram

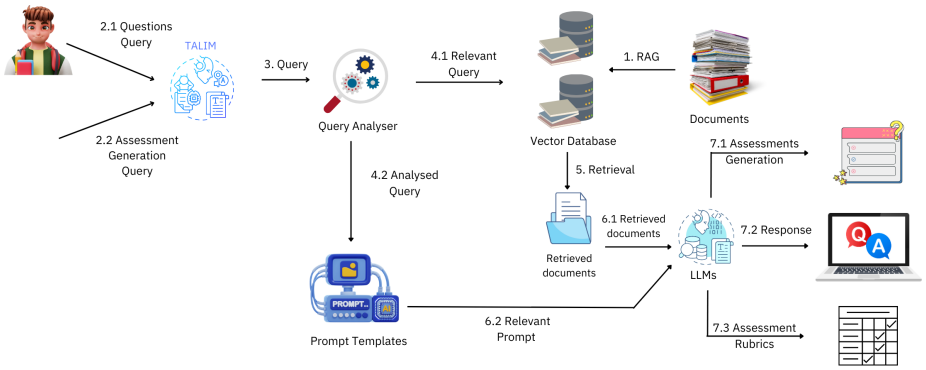

Figure 1: Work Flow Diagram of the System

C Use Case Diagram

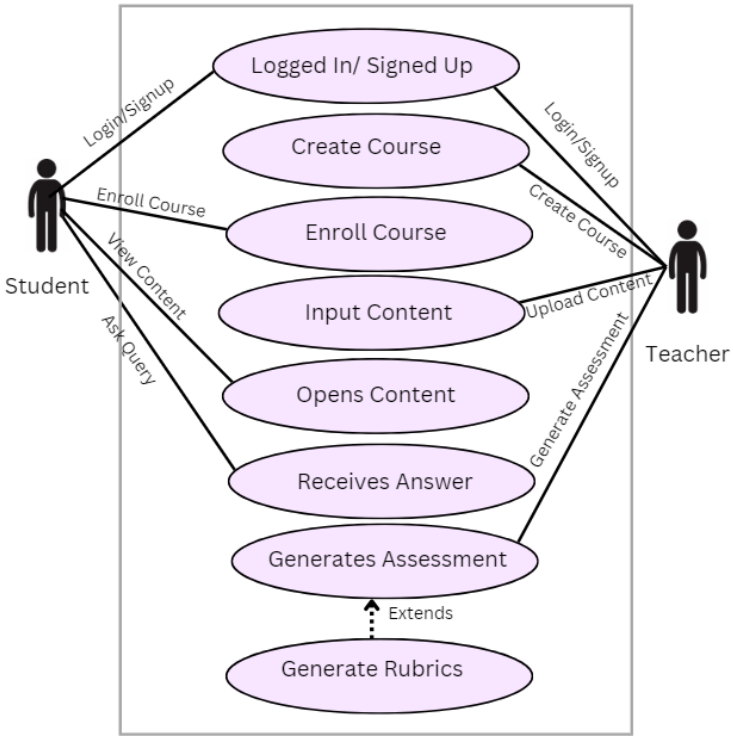

Figure 2: Use Case Diagram of the System

D Sequence Diagrams

D.1 Login Sequence Diagram

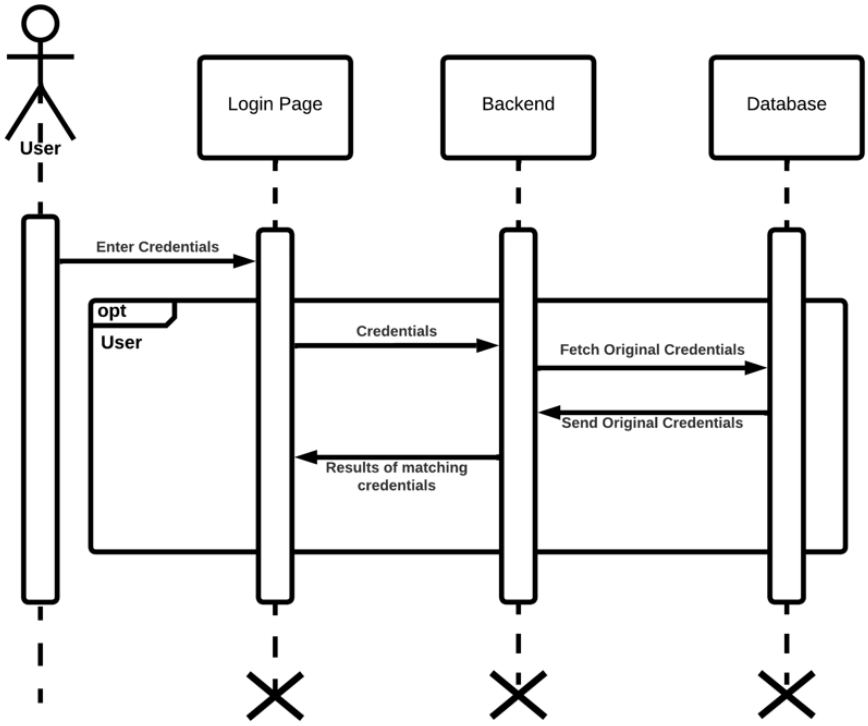

Figure 3: Sequence Diagram for User Login

D.2 Signup Sequence Diagram

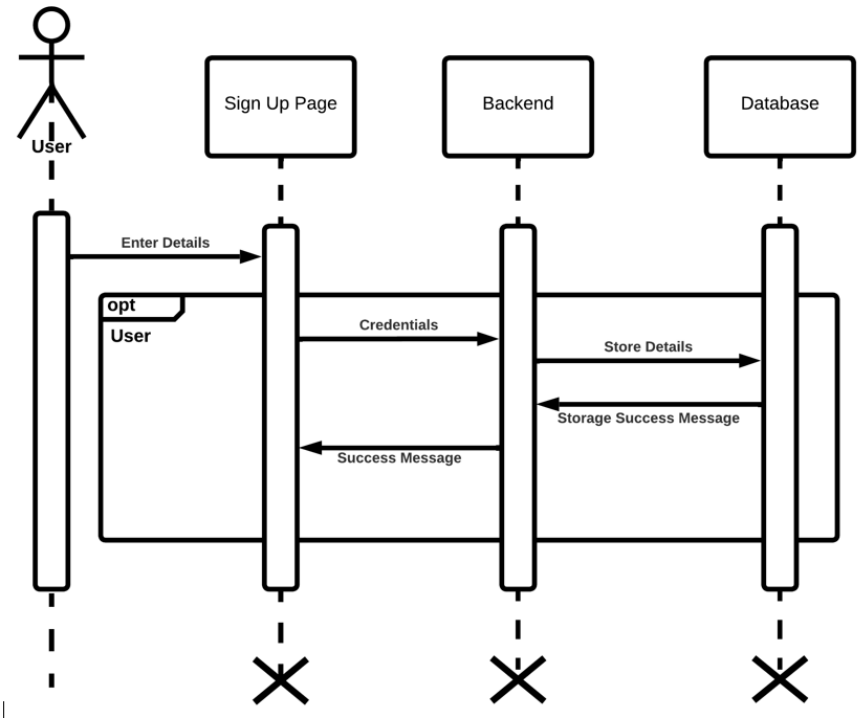

Figure 4: Sequence Diagram for User Signup

D.3 Creating a Course Sequence Diagram

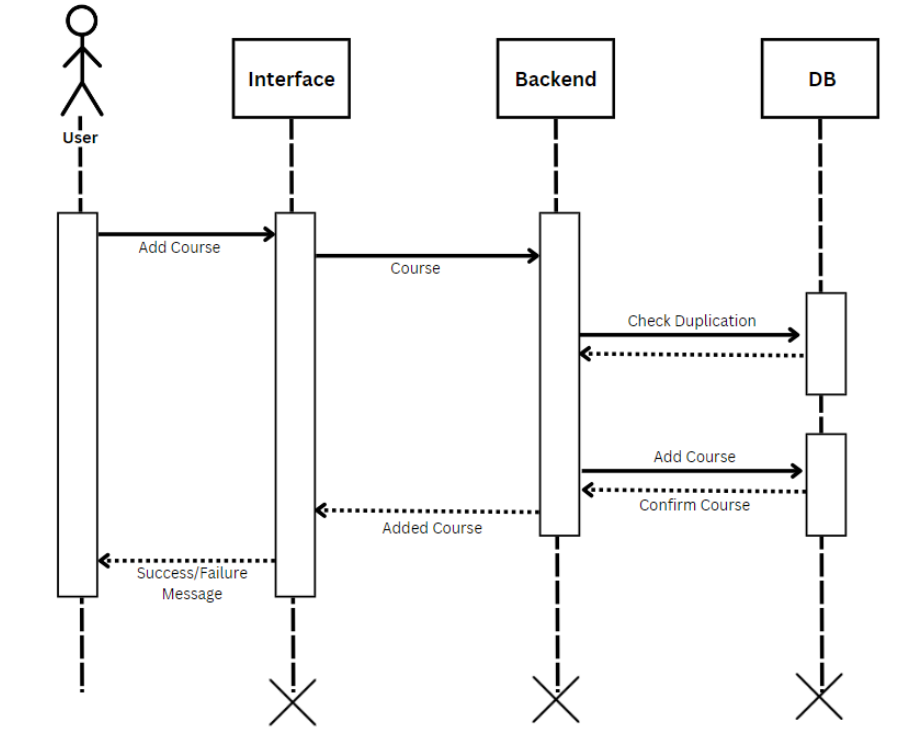

Figure 5: Sequence Diagram for Creating a Course

D.4 Enrolling in a Course Sequence Diagram

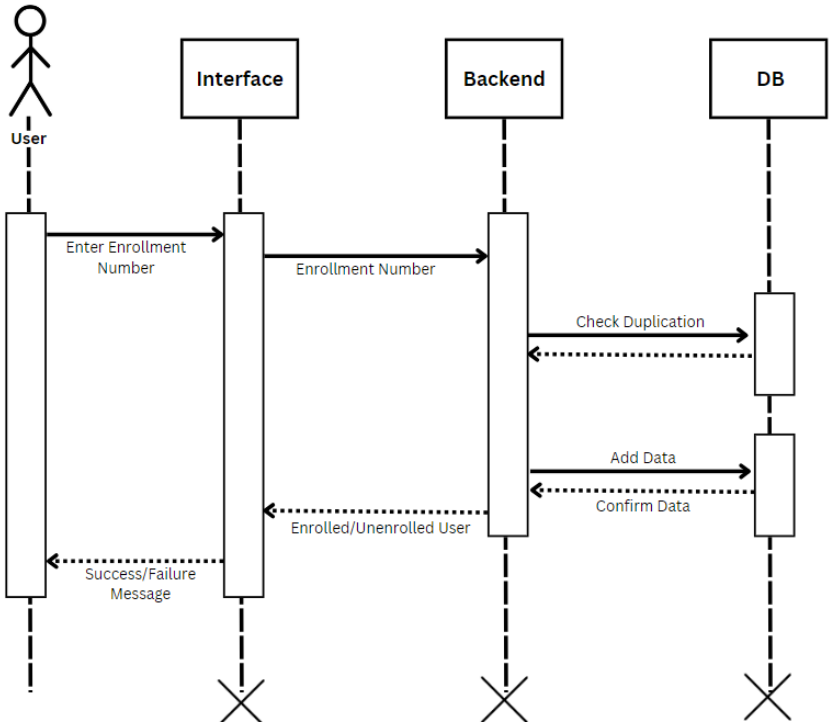

Figure 6: Sequence Diagram for Enrolling in a Course

D.5 Adding Content Sequence Diagram

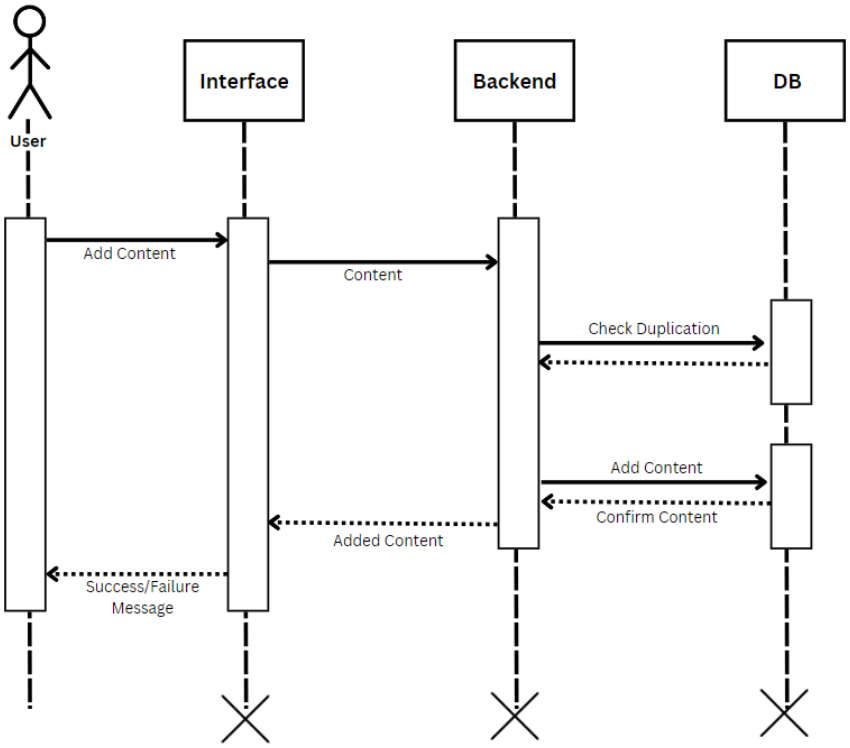

Figure 7: Sequence Diagram for Adding Content

D.6 Viewing Content Sequence Diagram

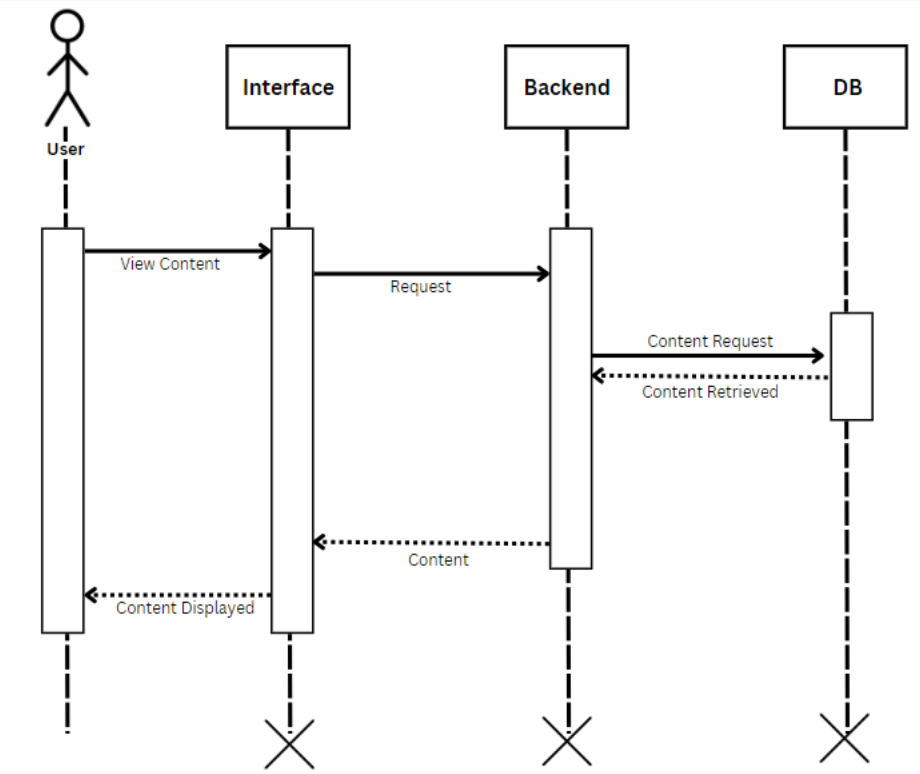

Figure 8: Sequence Diagram for Viewing Content

D.7 Assessment Generation Sequence Diagram

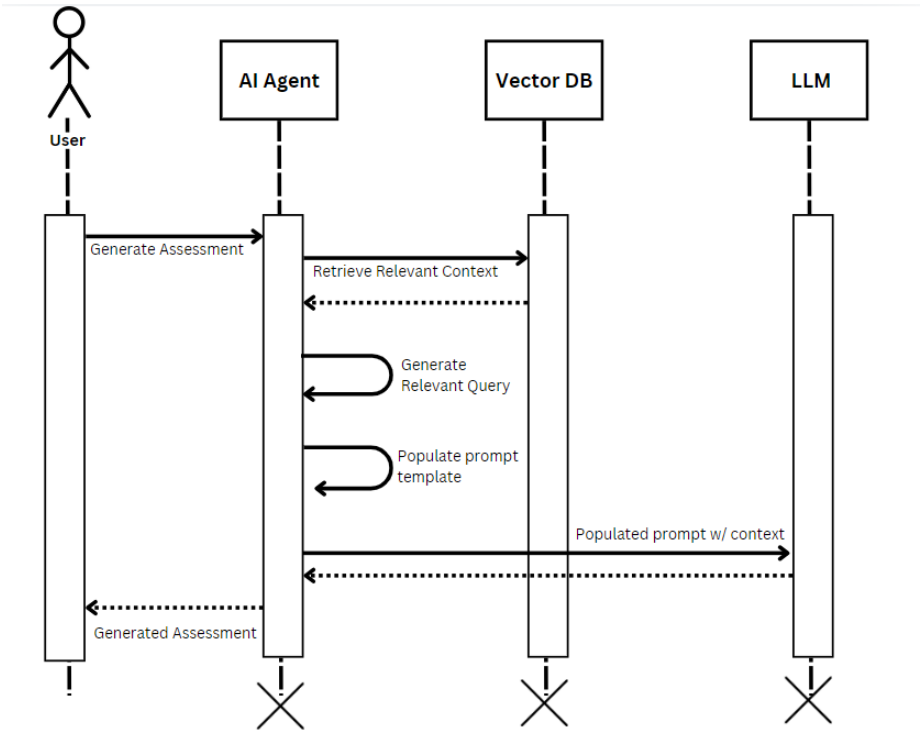

Figure 9: Sequence Diagram for Assessment Generation

D.8 Content Querying Sequence Diagram

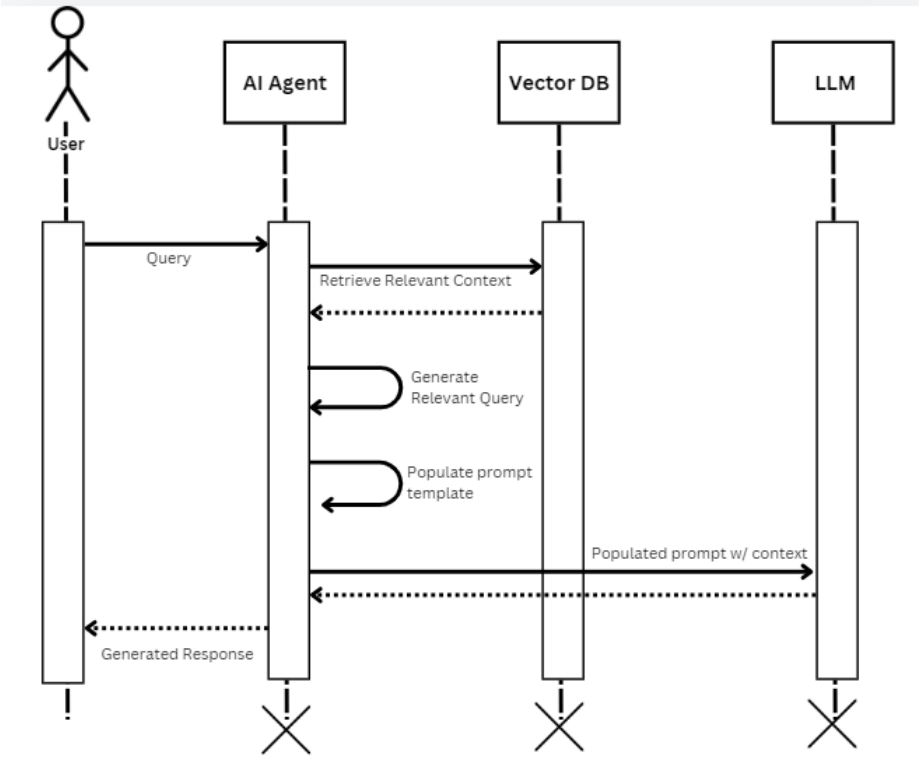

Figure 10: Sequence Diagram for Content Querying

E Data Flow Diagram

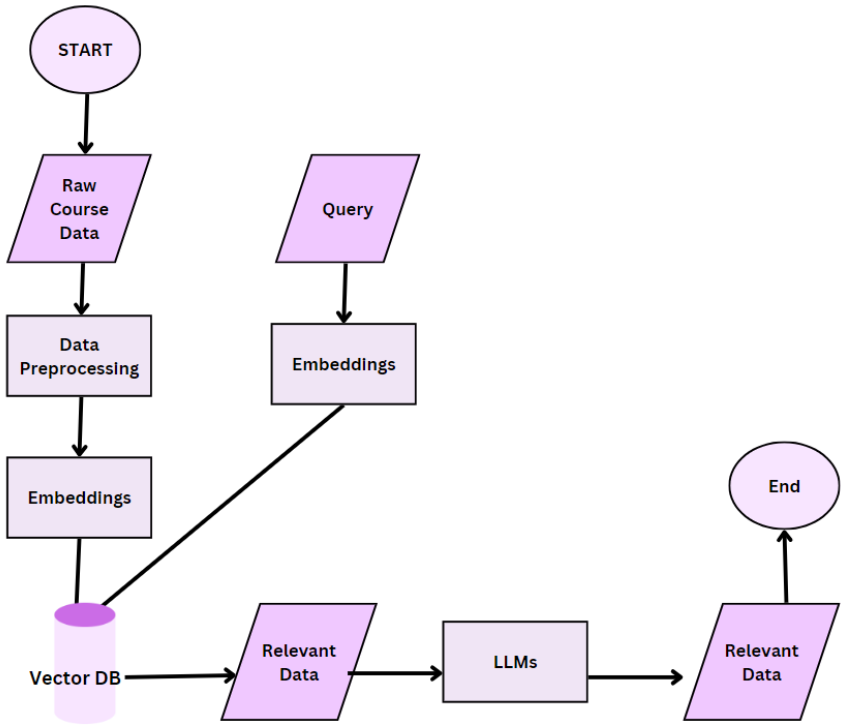

Figure 11: Data Flow Diagram of the System

References

[1] Meta. *Llama-2-7b-chat-hf*. Available at: https://huggingface.co/meta-llama/Llama-2-7b-chat-hf. Accessed: October 2024.

[2] AGI Sphere. *Retrieval-Augmented Generation with Llama2*. Available at: https://agi-sphere.com/retrieval-augmented-generation-llama2/. Accessed: October 2024.

[3] The Brazilian Ba. *The Amazing ChatGPT: Functional and Non-Functional Requirements*. Available at: https://thebrazilianba.com/2023/04/09/the-amazing-chatgpt-functional-and-non-functional-requirements/. Accessed: October 2024.

[4] Onlim. *Top Requirements for Chatbot Software*. Available at: https://onlim.com/en/top-requirements-for-chatbot-software/. Accessed: October 2024.

[5] Yellow.ai. *Chatbot Features*. Available at: https://yellow.ai/blog/chatbot-features/. Accessed: October 2024.

[6] Ivy.ai. *Example AI Chatbot Requirements*. Available at: https://go.ivy.ai/hubfs/Marketing/Handouts%20and%201%20Pagers/Example%20AI%20%20Chatbot%20Requirements.pdf. Accessed: October 2024.

[7] Pakistan Credit Rating Agency (PACRA) *Educational Statistics*. Available at: https://www.pacra.com/view/storage/app/Education%20-%20PACRA%20Research%20-%20June%2723_1688738214.pdf. Accessed: August 2024.

[8] Statista. *Projected Market Volume of EdTech Solutions* . Available at:https://www.statista.com/outlook/emo/online-education/pakistan. Accessed: September 2024.

[9] Researchgate *Leveraging chatbots to improve self-guided learning through conversational quizzes*. Available at: https://www.researchgate.net/publication/311508246_Leveraging_chatbots_to_improve_self-guided_learning_through_conversational_quizzes. Accessed: August 2024.

[10] Researchgate *TA-bot: An AI agent as a Teaching Assistant using Google's Conversational Technologies*. Available at: https://www.researchgate.net/publication/325332045_TA-bot_An_AI_agent_as_a_Teaching_Assistant_using_Google's_Conversational_Technologies. Accessed: August 2024.

[11] inria *Rexy, a configurable application for building Virtual Teaching Assistants*. Available at: https://inria.hal.science/hal-02544593/document. Accessed: August 2024.

[12] Researchgate *Evolution in Education: Chatbots*. Available at: https://www.researchgate.net/publication/342357938_Evolution_in_Education_Chatbots. Accessed: August 2024.

[13] Researchgate *Education 4.0: Artificial Intelligence Assisted Task-and Time planning system*. Available at: https://www.researchgate.net/publication/359120286_Education_40_Artificial_Intelligence_Assisted_Task-and_Time_planning_system. Accessed: August 2024.

[14] Researchgate *Kwame for Science: An AI Teaching Assistant for Science Education in West Africa*. Available at: https://www.researchgate.net/publication/361604415_Kwame_for_Science_An_AI_Teaching_Assistant_for_Science_Education_in_West_Africa. Accessed: August 2024.

[15] arxiv *Ethel: A Virtual Teaching Assistant*. Available at: https://arxiv.org/abs/2407.19452. Accessed: August 2024.

[16] ResearchGate *Virtual teaching assistant for undergraduate students using natural language processing and deep learning*. Available at: https://www.researchgate.net/publication/378084679_Virtual_teaching_assistant_for_undergraduate_students_using_natural_language_processing_deep_learning. Accessed: August 2024

[17] assets publishing *Teacher Workload Survey*. Available at: https://assets.publishing.service.gov.uk/media/5e12fcb7e5274a0f9e82e4fd/teacher_workload_survey_2019_main_report_amended.pdf. Accessed: September 2024.

[18] pewresearch *managing workload by teachers*. Available at: https://www.pewresearch.org/social-trends/2024/04/04/how-teachers-manage-their-workload/. Accessed: September 2024.

[19] egrove *Forecasted Online Education Market*. Available at: https://egrove.olemiss.edu/cgi/viewcontent.cgi?article=3167&context=etd. Accessed: September 2024.

[20] RSP Science *RSP Quiz Generation*. Available at: https://rspsciencehub.com/index.php/journal/article/view/834/688. Accessed: September 2024.