COMSATS University Islamabad

Abbottabad, Pakistan

AI-Assisted Learning Management System

By

Hanzla Nouman CIIT/FA21-BSE-015/ATD

Laiba Binta Tahir CIIT/FA21-BSE-019/ATD

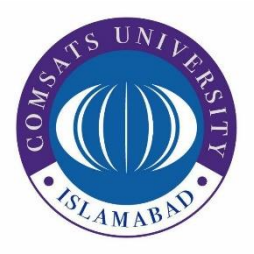
Arfah Ali CIIT/FA21-BSE-080/ATD

Supervisor

Mr. Mukhtiar Zamin

Bachelor of Science in Computer Science (20xx-20xx)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.



COMSATS University, Islamabad Pakistan

AI-Assisted Learning Management System

A project presented to

COMSATS Institute of Information Technology, Islamabad

In partial fulfillment

of the requirement for the degree of

Bachelor of Science in Computer Science (20xx-20xx)

By

Hanzla Nouman CIIT/FA21-BSE-015/ATD

Laiba Binta Tahir CIIT/FA21-BSE-019/ATD

Arfah Ali CIIT/FA21-BSE-080/ATD

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Hanzla Nouman	Laiba Binta Tahir	Arfah Ali
---	---	---

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) "AI Assisted LMS " was developed by **HANZLA NOUMAN(CIIT/FA21-BSE-015)**, **LAIBA BINTA TAHIR (CIIT/FA21-BSE-019)** and **ARFAH ALI (CIIT/FA21-BSE-080)** under the supervision of "MR. MUKHTIAR ZAMIN" and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

Supervisor

External Examiner

Head of Department
(Department of Computer Science)

EXECUTIVE SUMMARY

In today's evolving educational landscape, traditional Learning Management Systems (LMS) often fail to meet the diverse needs of students and teachers. These platforms typically lack intelligent adaptability, automated content generation, and data-driven personalization. Teachers face challenges such as repetitive content creation, manual grading, and limited tools for monitoring student performance, while students often experience one-size-fits-all content with minimal personalized learning support.

To address these challenges, our proposed AI-Assisted Learning Management System (LMS) aims to revolutionize the digital learning experience by integrating cutting-edge Artificial Intelligence technologies. This system goes beyond the conventional LMS by providing dynamic modules like the Content Generation and Recommendation Engine (CGRE), AI-Assisted Evaluation, and Performance Analytics. These modules automate essential educational tasks while tailoring learning materials and assessments to individual student profiles.

The platform features AI-generated lectures and study materials to reduce the preparation load on instructors. It also supports automated grading with intelligent feedback suggestions to save time and improve evaluation accuracy. Personalized content recommendations are generated based on user performance, preferences, and learning behavior, while insightful analytics help instructors adapt teaching strategies and support student growth more effectively.

Developed using the Component-Based Software Engineering (CBSE) methodology, the system ensures high modularity, scalability, and seamless integration across website. This AI-powered LMS reduces the manual workload for educators and transforms the student learning journey by making it more personalized, adaptive, and data-informed, paving the way for a smarter and more inclusive educational experience.

ACKNOWLEDGEMENT

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mukhtiar Zamin”. Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Hanzla Nouman Laiba Binta Tahir Arfah Ali

--- --- ---

ABBREVIATIONS

LMS	Learning Management System
CGRE	Content Generation And Recommendation Engine
OOP	Object oriented programming
UC	Use Case
SRS	Software Requirement specification document

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

This chapter provides an overview of the project titled AI-Assisted Learning Management System (LMS). It summarizes the purpose, methodology, and tools used in the development process, along with the relevance of the project to the courses studied during the Bachelor of Computer Science (BCS) program. The chapter also discusses the background and motivation behind this innovative educational system, aiming to bridge the gap between traditional and modern, AI-driven learning.

1.1. Brief

This project focuses on designing and developing an AI-Assisted Learning Management System (LMS) that transforms the educational experience for students and educators by integrating artificial intelligence. The system includes features such as AI-based content generation, personalized recommendations, automated evaluation, and performance analytics. The expected outcome of this project is a working prototype of a smart LMS that adapts to individual learners' needs and reduces the repetitive workload of teachers.

The tools and technologies used include:

- Languages: Python, JavaScript
- Frameworks: Django (for backend), NestJS(for frontend)
- AI/ML: Natural Language Processing (NLP) – Applied for content generation, keyword extraction, and personalized recommendations, enabling intelligent and contextual content delivery and Machine Learning Algorithms – Used for behavioral analysis, knowledge gap detection, clustering, and performance forecasting.
- Databases: PostgreSQL, MongoDB
- Others: TensorFlow, Scikit-learn for AI functionalities

Throughout this thesis, each chapter discusses key aspects such as system design, literature review, implementation details, evaluation, and future scope.

1.2. Relevance to Course Modules

This project directly relates to several core modules of the BCS curriculum, including:

- Artificial Intelligence – Used for building intelligent modules like content recommendation and automated grading.
- Software Engineering – Applied during system requirement analysis, design, development, and testing phases.
- Database Systems – Used to manage user data, content, and performance metrics.
- Web Technologies – Used to design and implement a user-friendly interface and interactive frontend.

- Human-Computer Interaction – Ensured the LMS is designed for a smooth, adaptive, and user-centered learning experience.
- Data Science– Enabled the visualization and understanding of students' learning trends and performance.

The knowledge gained from these subjects was instrumental in the planning and development of this AI-driven educational platform.

1.3. Project Background

Traditional Learning Management Systems (LMS) are widely used in educational institutions but often lack adaptability and personalized learning support. Most current platforms do not utilize student data effectively to enhance the learning process. The idea behind this project originated from the realization that many students struggle due to a one-size-fits-all approach, and teachers are overwhelmed by repetitive tasks such as manual grading and content preparation.

An **AI-Assisted LMS** addresses these challenges by integrating intelligent automation. It uses AI to:

- Generate lectures and content tailored to course outcomes,
- Provide recommendations to students based on their learning behavior,
- Automatically evaluate assessments and flag anomalies for manual review,
- Offer performance analytics to educators for better decision-making.

This system aims to empower both students and teachers by creating a more efficient, intelligent, and engaging learning environment.

1.4. Literature Review

This section describes current trends/ research/ products etc. related to your project.

1. Canvas LMS: Canvas is the most popular learning management system aimed at course management and student tracking. However, AI driven content generation and adaptive features are not included in this LMS. Its feedback functionality is quite basic and often requires manual input from instructors for personalized feedback.
2. Blackboard Learn: Blackboard Learn offers comprehensive course management and analytics but in AI content personalization its limited. Also, its grading and recommendation systems are not highly adaptive.
3. Google Classroom: Google classroom LMS was aimed at assignment management and collaboration. It lacks automated grading and personalized content recommendation, making it less robust for data driven insights and personalized learning experience.

Table 1.4.1: Related System Analysis with Proposed Project Solution

Application Name	**Weakness**	**Proposed Project Solution**

Canvas LMS	Limited adaptive learning & AI-driven content creation features.	The proposed system's Content Generation and Recommendation Engine (CGRe) helps teachers create interactive content, while AI-powered recommendations provide students with tailored study materials.
Blackboard Learn	Limited personalization in grading and content recommendation. Analytics are focused on risk identification rather than adaptive learning.	The AI-Assisted Evaluation module allows automated grading with customizable feedback, while analytics provide adaptive learning insights, supporting both personalized feedback and content adjustments.
Google Classroom	Lacks in-depth analytics, automated grading, and personalized content suggestions.	The proposed system offers AI-driven grading, performance analytics, and adaptive content recommendations, making the learning experience more tailored and data-informed.

1.5. Analysis from Literature Review (in the context of your project)

The literature review clearly indicates a significant gap in current LMS platforms regarding intelligent automation, adaptive learning, and personalized content delivery. While existing systems like Canvas, Blackboard, and Google Classroom offer standard LMS functionalities such as course management, basic analytics, and assignment handling, they lack AI integration to provide a dynamic, data-driven, and personalized learning experience. The AI-Assisted LMS proposed in this project addresses these limitations through:

- Automated Content Generation: Unlike Canvas and Blackboard, our system empowers educators to generate content using AI, saving time and enabling consistency in learning materials.
- Adaptive Recommendations: Based on student performance and preferences, the system offers personalized learning resources, a feature absent in traditional LMS platforms.
- Smart Evaluation: Unlike Google Classroom's manual grading system, our platform introduces an AI-based grading module that automates assessment and provides feedback instantly, with flagged exceptions for manual review.
- Insightful Analytics: Rather than focusing only on risk detection, as in Blackboard, the proposed system uses performance data to drive both content recommendation and teaching adjustments, fostering a more responsive and supportive educational environment.

This comparative analysis shows that the proposed AI-Assisted LMS is not only innovative but also addresses core shortcomings in modern educational tools by aligning with emerging trends in AI-driven education and personalized learning ecosystems.

1.6. Methodology and Software Lifecycle for This Project

To ensure an efficient, scalable, and user-centered development process, this project adopts both the Object-Oriented Programming (OOP) paradigm and the Agile Software Development Model. These approaches together support modular design, continuous improvement, and seamless integration of AI features based on real-world feedback.

1.6.1. Rationale behind Selected Methodology

The Object-Oriented Programming (OOP) approach has been chosen for the design of the AI-Assisted LMS. This methodology ensures modularity, reusability, and scalability by representing the system as a collection of interacting objects, each encapsulating its own data and behavior. System components such as user management, content generation, and analytics align naturally with core OOP principles like inheritance, encapsulation, and polymorphism.

OOP facilitates independent implementation of features, which enhances team collaboration and simplifies debugging. The use of OOP also aligns with industry-standard practices, promoting maintainability and ease of future enhancements.

1.6.2. Rationale behind Selected Methodology

The Agile Software Development Model has been selected for this project. Agile emphasizes iterative development, modular delivery, and continuous stakeholder feedback, making it highly suitable for a dynamic project like an AI-assisted LMS.

This approach ensures evolving user needs—from students, teachers, and administrators—are met effectively. Agile's flexibility is particularly important for integrating AI-driven features, which often require adjustments based on usability testing and feedback. Each sprint in the Agile cycle targets a specific module, allowing focused development, validation, and incremental progress.

2. Problem Definition

This chapter discusses the specific problem addressed by the project, along with the intended outcomes and expectations of the proposed solution.

2.1. Problem Statement

Traditional Learning Management Systems (LMS) lack the flexibility to cater to diverse learning needs and heavily rely on manual efforts by instructors for content creation, grading, and progress tracking. This results in a generic and often ineffective learning experience for students, as well as increased workload for teachers. Most existing LMS platforms do not incorporate AI to offer personalized learning experiences, automated evaluation, or performance insights.

To address these gaps, our proposed system—an AI-Assisted LMS—aims to automate repetitive tasks, offer adaptive learning paths, and use data-driven insights to enhance both teaching and learning. It will empower educators to focus more on mentoring rather than routine work, and support students with a more tailored and engaging learning environment.

2.2. Deliverable and Development Requirements

Deliverables:

- A fully functional AI-assisted web-based Learning Management System.
- A cross-platform mobile application using Flutter for students.
- A Content Generation and Recommendation Engine (CGRe) for AI-based content creation.
- An AI-Assisted Evaluation Module for automated grading and feedback.
- A Performance Analytics Dashboard for monitoring student progress and engagement.

Development Requirements:

- Frontend: TypeScript with React or similar framework.
- Backend: Node.js or Deno using TypeScript.
- AI Models: Implemented using Python libraries (e.g., TensorFlow, scikit-learn, Transformers).
- Database: PostgreSQL or MongoDB for structured and flexible data storage.
- Cloud Services: For hosting, scalability, and potential integration of third-party APIs.

3. Requirement Analysis

This chapter outlines the system's expected behavior by analyzing user needs and defining both functional and non-functional requirements for the proposed solution.

3.1. Use Cases Diagram(s)

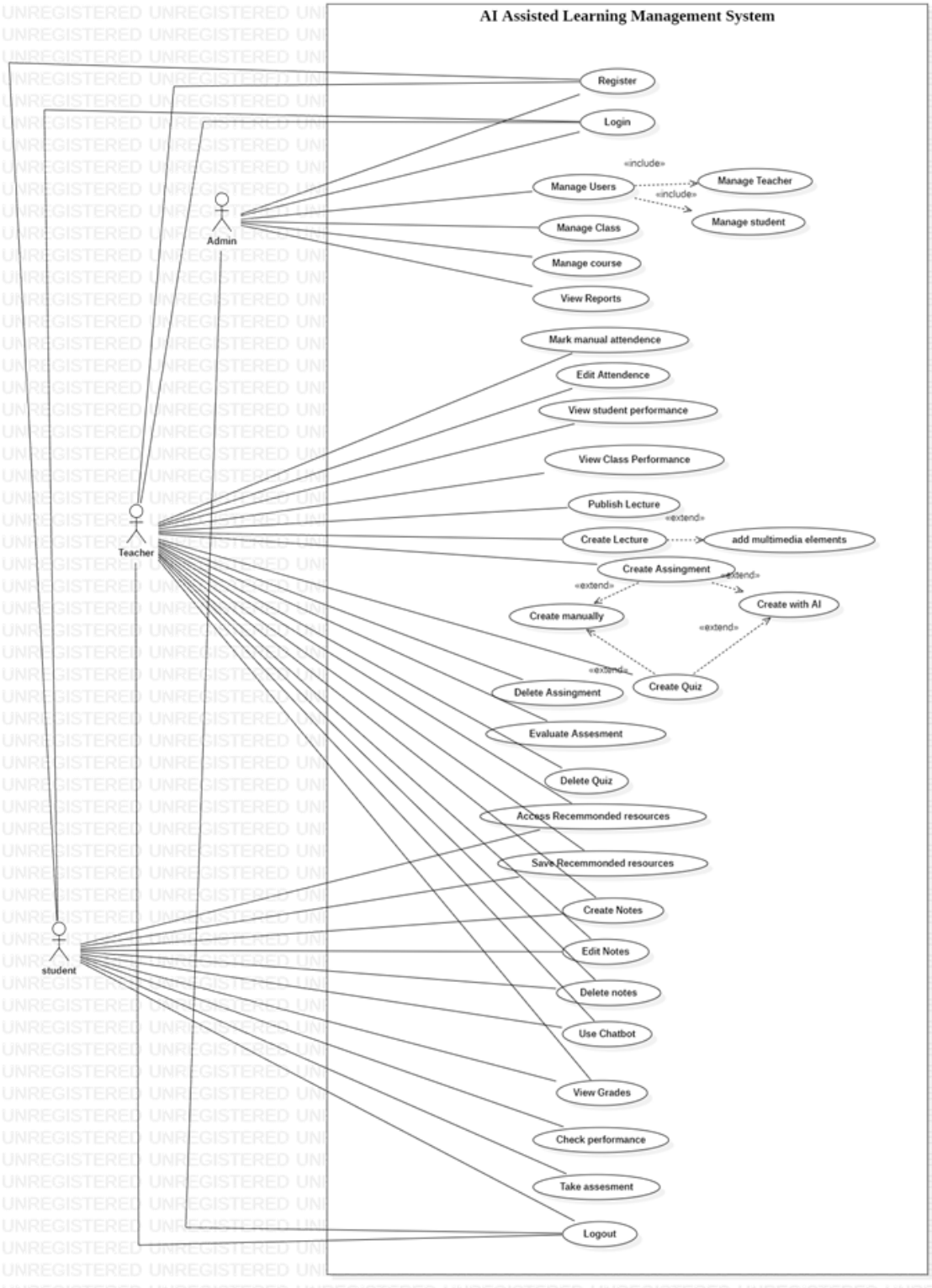


Figure 3.1 Use case Diagram

3.2. Detailed Use Case

The table below provides a detailed example of a use case for an AI-Assisted Learning Management System (LMS). This structured template helps document and present user

interactions and system functionalities in a clear and organized manner. The use case describes how specific system features are designed to meet user needs and outlines the steps, conditions, and outcomes associated with those interactions.

Use Case Description: Access Recommended Resources

Table 3.2.1 UC-1 Access Recommended Resources

Use Case ID	**UC-1**
Use Case Name	Access Recommended Resources
Actors	**Primary Actor**: Student **Secondary Actor**: Content Recommendation Module (System)
Description	The student accesses the system to view personalized recommendations for study materials based on their individual performance, preferences, and progress in the course.
Trigger	The student logs into the system and navigates to the "Recommended Resources" section.
Preconditions	1. The student must be logged into the system. 2. The student must have completed some assessments or have progress tracked in the system.
Postconditions	1. The student can view a list of personalized study materials. 2. The student can access the recommended resources.
Normal Flow	**1.0 Access Recommended Resources** 1. The student logs into the system. 2. The student navigates to the "Recommended Resources" section, either from the dashboard or the course-specific page. 3. The system analyzes the student's performance, preferences, and progress to generate a list of personalized recommendations. 4. The system presents the recommended resources (e.g., books, videos, quizzes, assignments). 5. The student reviews the recommendations, which may include explanations, additional reading materials, or practice exercises. 6. The student selects a resource to access. 7. The system directs the student to the selected resource or content.
Alternative Flows	**1.1 No Recommendations Available** 1. If the system cannot generate recommendations due to insufficient data, it displays: "No recommendations available at this time. Please complete more assessments." **1.2 No Internet Connection** 1. If the student experiences an internet issue, the system prompts:

	"Unable to access resources due to connection issues. Please try again later."
Exceptions	**1.0.E1 System Error While Accessing Resources** 1. The system displays an error message: "Unable to access the requested resource. Please try again." 2. The student may retry accessing the resource.
Business Rules	**BR-1**: Recommended resources must align with the student's current course syllabus and performance data. **BR-2**: The system may only recommend resources that are part of the officially approved content library.
Assumptions	1. The system continuously tracks and updates the student's performance data. 2. Recommended resources will always be available to students as long as they have internet access.

Use Case Description: Save Recommended Resources

Table 3.2.2 UC-2 Save Recommended Resources

Use Case ID	**UC-2**
Use Case Name	Save Recommended Resources
Actors	**Primary Actor**: Student **Secondary Actor**: Content Recommendation Module (System)
Description	The student saves recommended study resources for later review, allowing them to bookmark content they find helpful or wish to return to.
Trigger	The student clicks on the "Save for Later" button next to a recommended resource.
Preconditions	1. The student must be logged into the system. 2. The student must have accessed recommended resources.
Postconditions	1. The resource is saved in the student's "Saved Resources" section. 2. The student can access the saved content at any time.
Normal Flow	**1.0 Save Recommended Resources** 1. The student logs into the system and accesses the recommended resources section. 2. The student reviews the available recommendations and finds a resource they wish to save. 3. The student clicks on the "Save for Later" button next to the resource.

	4. The system saves the resource to the student's personalized "Saved Resources" section. 5. The system displays a confirmation message: "Resource saved successfully." 6. The student can later navigate to their "Saved Resources" section to view and access all saved content.
Alternative Flows	**2.1 Resource Already Saved** 1. If the student tries to save a resource already saved, the system displays: "This resource is already saved in your list." **2.2 Insufficient Data to Save** 1. If the system fails to save the resource, it displays: "Unable to save resource at this time. Please try again later."
Exceptions	**2.0.E1 System Error During Save Operation** 1. If an error occurs while saving the resource, an error message is displayed: "Error occurred while saving. Please try again later."
Business Rules	**BR-1**: Resources can only be saved if they are part of the official recommended list. **BR-2**: Saved resources are accessible until the student marks them as "Completed" or removes them manually.
Assumptions	1. The student may save multiple resources for later review. 2. Saved resources will not expire unless the course or content is deleted by the administrator.

Use Case Description: Create lecture

Table 3.2.3 UC-3 Create Lecture

Field	**Details**
Use Case ID	UC-3
Use Case Name	Create Lecture
Actors	**Primary Actor:** Teacher **Secondary Actor:** Content Generation System
Description	The teacher creates lecture content using the AI-assisted content generation feature. Optionally, the teacher can manually edit AI-generated content.
Trigger	The teacher clicks the **"Create Lecture"** button in the course module and selects "Create with AI."
Preconditions	1. The teacher must be logged into the LMS. 2. The teacher must have access to the specific course within the LMS.

Postconditions	The lecture is saved as a draft under the "Saved" tab. The lecture is published under the "Published" tab for student access.
Normal Flow	1. The teacher navigates to the "Courses" tab in the dashboard. 2. The teacher selects a course and clicks the "Create Lecture" button. 3. The teacher selects the "Create with AI" option. 4. The system prompts the teacher to enter inputs such as Title, Description, duration, Chapter, Academic Level, Learning Outcomes, Preferred Formats, Attachments. Objective, and Reference. 5. The AI generates lecture content based on the inputs provided and displays it for review. 6. The teacher optionally edits the AI-generated content using the built-in editor. 7. The teacher clicks "Save" to save as a draft or "Save and Publish" to make it available to students.
Alternative Flows	**AF-1:** Teacher selects "Create Manually": 1. Instead of "Create with AI," the teacher selects "Create Manually." 2. The system displays a form.
	3. The teacher manually enters all lecture details and content. 4. The teacher clicks "Save" or "Save and Publish" to complete the process.
Exceptions	**EX-1:** AI Generation Error: 1. The AI fails to generate content due to a technical issue. 2. The system suggests retrying or switching to manual.
	EX-2: Save/Publish Error: 1. The system encounters an error while saving or publishing the lecture. 2. The system advises retrying.
Business Rules	1. AI-generated content must align with inputs and academic standards. 2. Edited content must pass validation before saving or publishing.
Assumptions	1. Teachers have reliable internet access during the creation process. 2. The AI system is functional and accessible.

Use Case Description: Create Assignment

Table 3.2.4 UC-4 create assignment

Field	**Description**
Use Case ID	UC-4

Use Case Name	Create Assignment
Actors	Primary Actor: Teacher Secondary Actor: System
Description	The teacher creates and schedules an assignment by entering details, uploading resources, and applying rubrics.
Trigger	The teacher clicks the "Create Assignment" button.
Preconditions	1. The teacher must be logged into the system. 2. The teacher must have appropriate permissions to create assignments.
Postconditions	1. The assignment is saved in the system. 2. The assignment is scheduled for a specified date and time. 3. A confirmation message is displayed.
Normal Flow	1. The teacher logs into the system and navigates to the "Assignments" tab. 2. The teacher clicks "Create Assignment." 3. The system displays the "Create Assignment" form. 4. The teacher fills out the form with details such as Assignment Name, Rubrics, References, Due Date, etc. 5. The teacher uploads additional resources or enters assignment content. 6. The teacher customizes or applies a predefined grading rubric. 7. The teacher reviews the details and selects "Save and Publish" or "Save."
Alternative Flows	**AF-1: Missing Mandatory Fields** 1. The system highlights incomplete fields and prompts the teacher to fill them in. 2. The teacher enters the missing information and continues. **AF-2: Invalid Due Date** 1. The system shows an error if the due date is invalid. 2. The teacher corrects the date and proceeds.
Exceptions	**EX-1: System Error** 1. The system fails to save the assignment. 2. The system displays an error message and prompts a retry.
Business Rules	BR-1: Assignments must include either instructions or attached resources. BR-2: Assignments must have a title and a valid due date.

Assumptions	1. Teachers will create assignments within course timelines. 2. Stable internet connectivity during assignment creation.

Use Case Description: Create Quiz

Table 3.2.5 UC-5 Create quiz

Field	**Description**
Use Case ID	UC-5
Use Case Name	Create Quiz
Actors	Primary Actor: Teacher Secondary Actor: System
Description	The teacher creates and schedules a quiz by entering questions, defining correct answers, and assigning marks.
Trigger	The teacher clicks the "Create Quiz" button.
Preconditions	1. The teacher must be logged into the system. 2. The teacher must have appropriate permissions to create quizzes.
Postconditions	1. The quiz is saved in the system. 2. The quiz is scheduled for a specified date and time. 3. A confirmation message is displayed.
Normal Flow	1. The teacher logs into the system and navigates to the "Quizzes" tab. 2. The teacher clicks "Create Quiz." 3. The system displays the "Create Quiz" form. 4. The teacher fills out details such as Quiz Title, Rubrics, References, Due Date, etc. 5. The teacher adds questions (e.g., MCQs, Short Questions, Fill-in-the-Blanks) using the question editor. 6. The teacher defines correct answers and assigns marks per question. 7. The teacher reviews the quiz structure and selects "Save and Publish" or "Save."
Alternative Flows	**AF-1: Missing Mandatory Fields** 1. The system highlights incomplete fields and prompts the teacher to fill them in. 2. The teacher enters the missing information and continues. **AF-2: Invalid Date or Time** 1. The system shows an error if the scheduled date is invalid. 2. The teacher corrects the date and proceeds.

Exceptions	**EX-1: System Error** 1. The system fails to save the quiz. 2. The system displays an error message and prompts a retry.
Business Rules	BR-1: Quizzes must include at least one question. BR-2: Quizzes must have a valid scheduled date and time.
Assumptions	1. Teachers will create quizzes within course timelines. 2. Stable internet connectivity during quiz creation.

Use Case Description: Evaluate assessment

Table 3.2.6 UC-6 Evaluate assignment

Use Case ID	**UC-6**
Use Case Name	Evaluate assessment
Actors	Primary Actor: Teacher Secondary Actor: Standard Evaluation Module (System)
Description	The teacher evaluates student submissions for assignments or quizzes. The system supports AI-assisted grading and allows manual adjustments to scores and feedback before finalizing and publishing results.
Trigger	The teacher accesses submitted assessments for evaluation.
Preconditions	1. Students have submitted their assignments/quizzes. 2. The teacher is logged into the system and has access to the evaluation module.
Postconditions	1. Scores and feedback are recorded and saved. 2. The results are available for publishing to students.

Normal Flow	1. The teacher logs into the system and navigates to the "Pending Evaluations" section. 2. The system displays a list of submitted assignments/quizzes. 3. The teacher selects a specific assessment to evaluate. 4. The system performs AI-assisted grading (if enabled) based on predefined rubrics and displays the scores. 5. The teacher reviews the AI-assigned scores and feedback. 6. The teacher performs any of the following actions: Adjust scores manually, Edit or add detailed feedback for each student. 7. The teacher clicks the "Save Evaluation" button. 8. The system saves updated scores and feedback. 9. The teacher clicks "Publish Results" to make scores and feedback visible to students. 10. The system displays a confirmation message: "Results published successfully."
Alternative Flow	2.1 No Submissions Available 1. If no submissions are available, the system displays a message: "No submissions found for evaluation." 2.2 Incomplete AI Grading 1. If AI grading fails for any question or response, the system notifies the teacher: "AI grading incomplete. Manual evaluation required." 2. The teacher proceeds to manually grade the affected questions. 2.3 Partial Evaluation 1. If the teacher saves the evaluation without publishing, the system marks the submissions as "Partially Evaluated.
Exceptions	2.0.E1 System Error During Grading 1. If an error occurs during AI-assisted grading, the system displays: "Error occurred during grading. Please grade manually."
Business Rules	BR-1: Scores must be saved before publishing results.

	BR-2: Teachers can override AI-assigned scores and feedback. BR-3: Final scores and feedback cannot be edited after publishing.
Assumptions	Teachers will cross-check AI-assigned scores for accuracy. Students expect timely results after submission deadlines.

1.1.Functional Requirement

Feature 1: Personalized Learning

Table 3.3.1: Functional Requirement 1.1

Field	Details
Identifier	FR-PL-001
Title	Personalized Content Recommendations
Requirement	The system shall provide personalized learning material recommendations to each student based on their learning history, preferences, and performance metrics.
Source	Design Team
Rationale	To enhance student engagement and learning outcomes by providing relevant content.
Business Rule	Recommendations must be aligned with the curriculum.
Dependencies	FR-PL-002, FR-PA-001
Priority	High

Table 3.3.2: Functional Requirement 1.2

Field	Details
Identifier	FR-PL-002
Title	Interactive Material Generation
Requirement	The system should allow teachers to generate interactive content such as quizzes and flashcards tailored to students' needs.
Source	Teacher Feedback
Rationale	To facilitate active learning through engaging materials.
Business Rule	Materials must adhere to institutional guidelines.
Dependencies	FR-CG-001
Priority	Medium

Feature 2: Automated Evaluations

Table 3.3.3: Functional Requirement 2.1

Field	Details
Identifier	FR-AE-001
Title	Instant Feedback Mechanism
Requirement	The system shall provide instant feedback to students after quiz submissions, highlighting areas of improvement.
Source	Assessment Tools Research
Rationale	To support continuous learning through real-time feedback.
Business Rule	Feedback must include references to relevant study materials.
Dependencies	FR-PL-001, FR-PA-002
Priority	High

Table3.3.4: Functional Requirement 2.2

Field	Details
Identifier	FR-AE-002
Title	Self-Evaluation Tools
Requirement	The system shall enable students to perform self-assessments using AI-driven evaluations of written and multiple-choice responses.
Source	Student Feedback Surveys
Rationale	To empower students with self-directed learning capabilities.
Business Rule	Self-assessments must not count toward final grades unless explicitly permitted.
Dependencies	None
Priority	Medium

Feature 3: Performance Analytics

Table 3.3.5: Functional Requirement 3.1

Field	Details
Identifier	FR-PA-001
Title	Analytics Dashboard for Teachers
Requirement	The system shall provide a dashboard with visualized analytics on student performance trends, attendance, and engagement.
Source	Requirements Analysis Document
Rationale	To assist teachers in making data-driven decisions.
Business Rule	Data must be anonymized when aggregated.

Dependencies	FR-AM-001
Priority	High

Table 3.3.6: Functional Requirement 3.2

Field	**Details**
Identifier	FR-PA-002
Title	Predictive Performance Insights
Requirement	The system shall use AI to predict students' future performance based on their learning patterns and attendance records.
Source	Predictive Modeling Research
Rationale	To enable proactive intervention strategies.
Business Rule	Predictions must be flagged as estimates and not definitive results.
Dependencies	FR-PA-001
Priority	High

Feature 4: Content Generation

Table3.3.7: Functional Requirement 4.1

Field	**Description**
Identifier	CG-01
Title	AI-Assisted Lecture Creation
Requirement	The system must allow teachers to generate lecture content automatically by providing inputs such as learning objectives, topics, and references.
Source	Design Team
Rationale	Automating lecture content creation helps reduce the workload and improves efficiency for teachers.
Business Rule	The generated content must align with provided inputs and adhere to academic standards.
Dependencies	Requires a stable connection to the AI model and teacher access to the LMS.
Priority	High

Table 3.3.8: Functional Requirement 4.2

Field	**Description**
Identifier	CG-02
Title	Manual Editing of AI-Generated Content
Requirement	The system must allow teachers to edit and modify the AI-generated lecture content before saving or publishing.
Source	Teacher Feedback

Rationale	Teachers may want to customize AI-generated lectures to better fit their teaching style and student needs.
Business Rule	Edited content should be validated for required fields before saving or publishing.
Dependencies	Requires successful generation of lecture content using the AI feature.
Priority	High

Table 3.3.9: Functional Requirement 4.3

Field	**Description**
Identifier	CG-03
Title	Save and Publish Options
Requirement	The system must provide options to save the lecture as a draft or publish it directly for student access.
Source	Teacher Feedback
Rationale	Teachers may want flexibility to refine content progressively or make it immediately available to students.
Business Rule	Drafts are saved under the "Saved" tab and visible only to the teacher; published lectures appear under the "Published" tab for students.
Dependencies	Edited or AI-generated content must be validated before saving or publishing.
Priority	Medium

1.2. Non-Functional Requirements

This section outlines the quality requirements that ensure the system meets performance, usability, and reliability standards.

Usability

- The system allow users to access the Content Recommendation module and receive recommendations within 2 clicks of the main dashboard.
- The Standard Evaluation module shall provide clear, step-by-step instructions for completing evaluations to minimize user confusion.
- The system interface shall support accessibility features compliant with WCAG 2.1 Level AA standards to accommodate users with disabilities.
- The mobile version of the LMS shall provide consistent and intuitive user experience compared to the web version.
- Self-Evaluation modules shall include tooltips to guide users on their functionality.

Reliability

- The system's uptime shall be 99% over the 24/7 operational period.
- The Database Server shall recover automatically from failures within 30 seconds.
- The system shall support data backup every 12 hours to prevent data loss.

Security

- The system requires multi-factor authentication (MFA) for users accessing sensitive modules like Analytics and Attendance.
- Data transferred between Client Devices and servers shall be encrypted using HTTPS/TLS 1.3 protocols.
- All user data stored in the Database Server shall be encrypted using AES-256 encryption standards.
- The system shall logs all user activities, including logins, access, and modifications, for auditing purposes.

Performance

- 95% of webpages generated by the LMS shall load within 3 seconds over a 20 Mbps or faster internet connection.
- The Content Generation module shall process and generate learning content within 10 seconds for standard text inputs.
- The system shall support a minimum of 500 concurrent users accessing modules like Analytics and Content Recommendation without performance degradation.

Scalability

- The system shall be able to scale horizontally to support up to 10,000 concurrent users by adding additional Application Servers.
- The system shall ensure seamless performance even as the database size grows up to 1 million records.

Maintainability

- The system allow new AI components to be integrated into the Application Server with minimal changes to existing modules.
- Code shall follow industry-standard design patterns and include detailed documentation to ensure maintainability.
- System logs and error messages shall be stored to facilitate troubleshooting and future system updates.

4. Design and Architecture

This chapter provides an overview of the system's high-level architecture, data design, process flow, and various design models used during development.

4.1. System Architecture

The system architecture of the AI-Assisted Learning Management System (LMS) is based on a **Modular Monolithic Architecture**, designed to ensure scalability, maintainability, and smooth integration of AI components. The system is divided into logical modules, each responsible for a distinct functionality, yet deployed as a single cohesive unit.

The architecture consists of the following core layers:

- **Presentation Layer (Frontend):**
  Built using **TypeScript** and modern frameworks, this layer serves as the interface for students, teachers, and administrators. It communicates with the backend via APIs and presents dynamically generated content based on AI insights and user interaction.
- **Business Logic Layer (Backend):**
  Developed using **Node.js/Deno**, it handles all core functionalities such as user management, content delivery, evaluation processing, and communication with the AI models. This layer enforces business rules and controls the overall workflow.
- **AI Services Layer:**
  Implemented using **Python-based AI models**, this component handles tasks such as:
  - Personalized content recommendation
  - Automated answer evaluation
  - Quiz and material generation These services are accessed via internal APIs and are designed to be modular for easier updates and training.

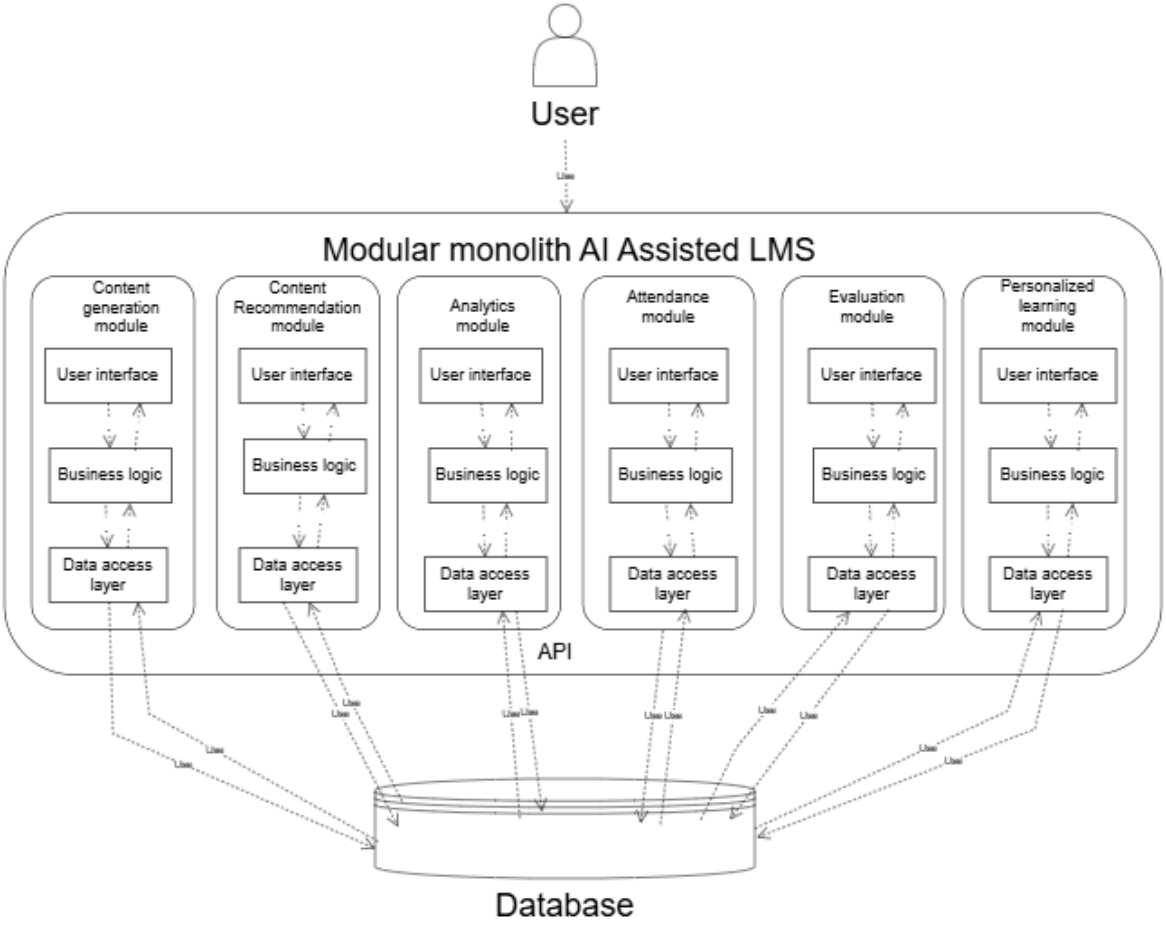


Figure 4.1 System Architecture

- **Database Layer:**
  Stores and manages persistent data including user information, course materials, results, attendance records, and system logs. It ensures data consistency and efficient retrieval for analytics and reporting.
- **Admin Control Layer:**
  Provides system-level management features such as:
  - User and role administration
  - Report generation
  - Monitoring analytics and AI performance

This architecture ensures that each module can be developed and tested independently, allowing a faster development cycle and easy integration of new features in future iterations.

4.2. Data Representation [Diagram + Description]

Class diagram

This diagram captures the relationships between classes using associations (e.g., courses enrolling users), aggregations (e.g., quizzes containing questions), and inheritance (e.g., lectures and quizzes extending content). The design reflects a modular, scalable structure capable of supporting advanced LMS functionalities.

Figure 4.2 Class diagram

4.3. Process Flow/Representation

We have created activity diagrams for major processes of the system to represent the flow of actions effectively. Below are the descriptions of each activity:

Activity Diagram: Access recommended resources

The student can access recommended resources generated by AI Content Recommendation feature.

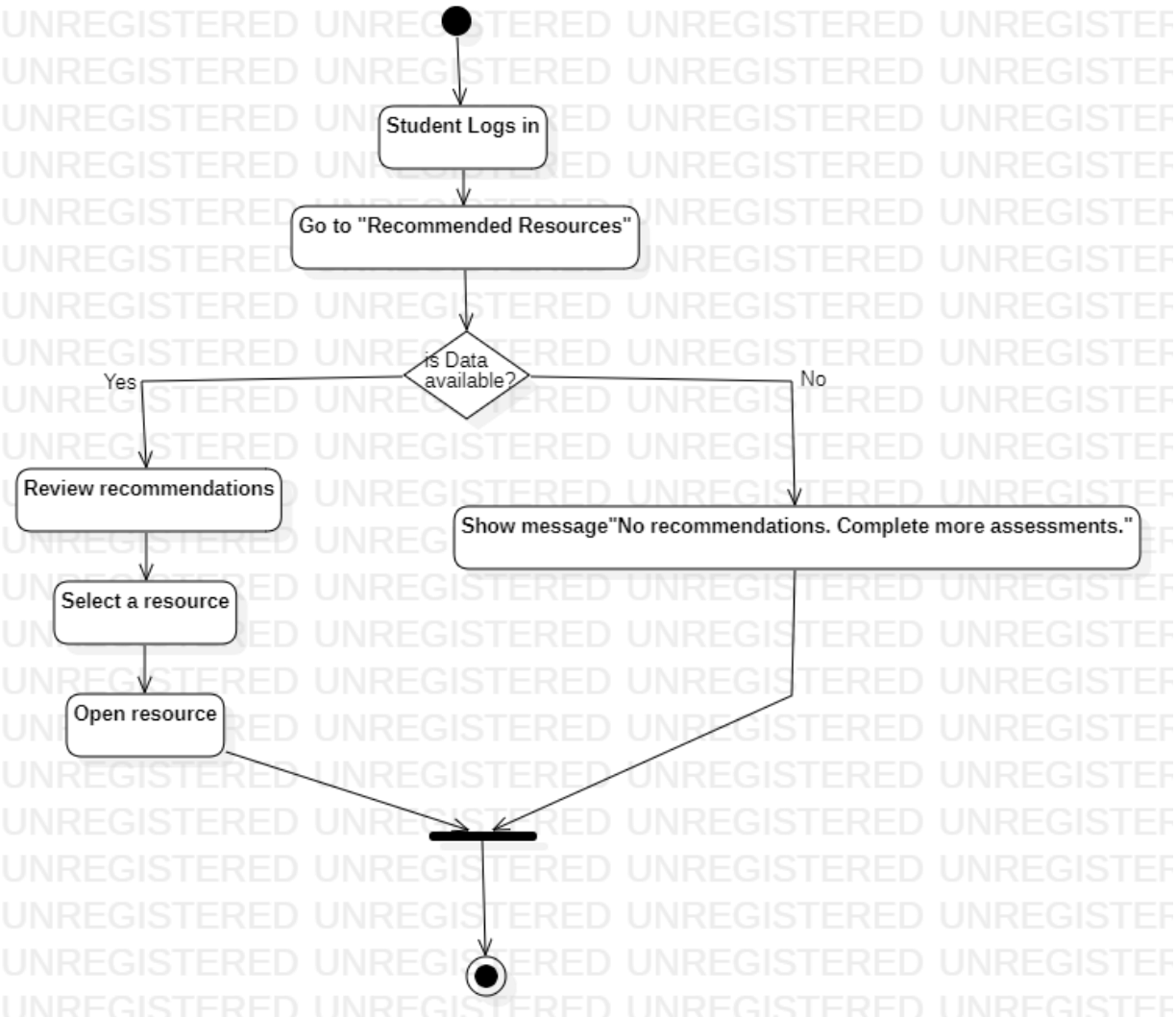


Figure 4.3 Activity diagram: Access recommended resources

Activity Diagram: Create lecture

The teacher creates lecture content using the AI-assisted content generation feature. Optionally, the teacher can manually edit AI-generated content.

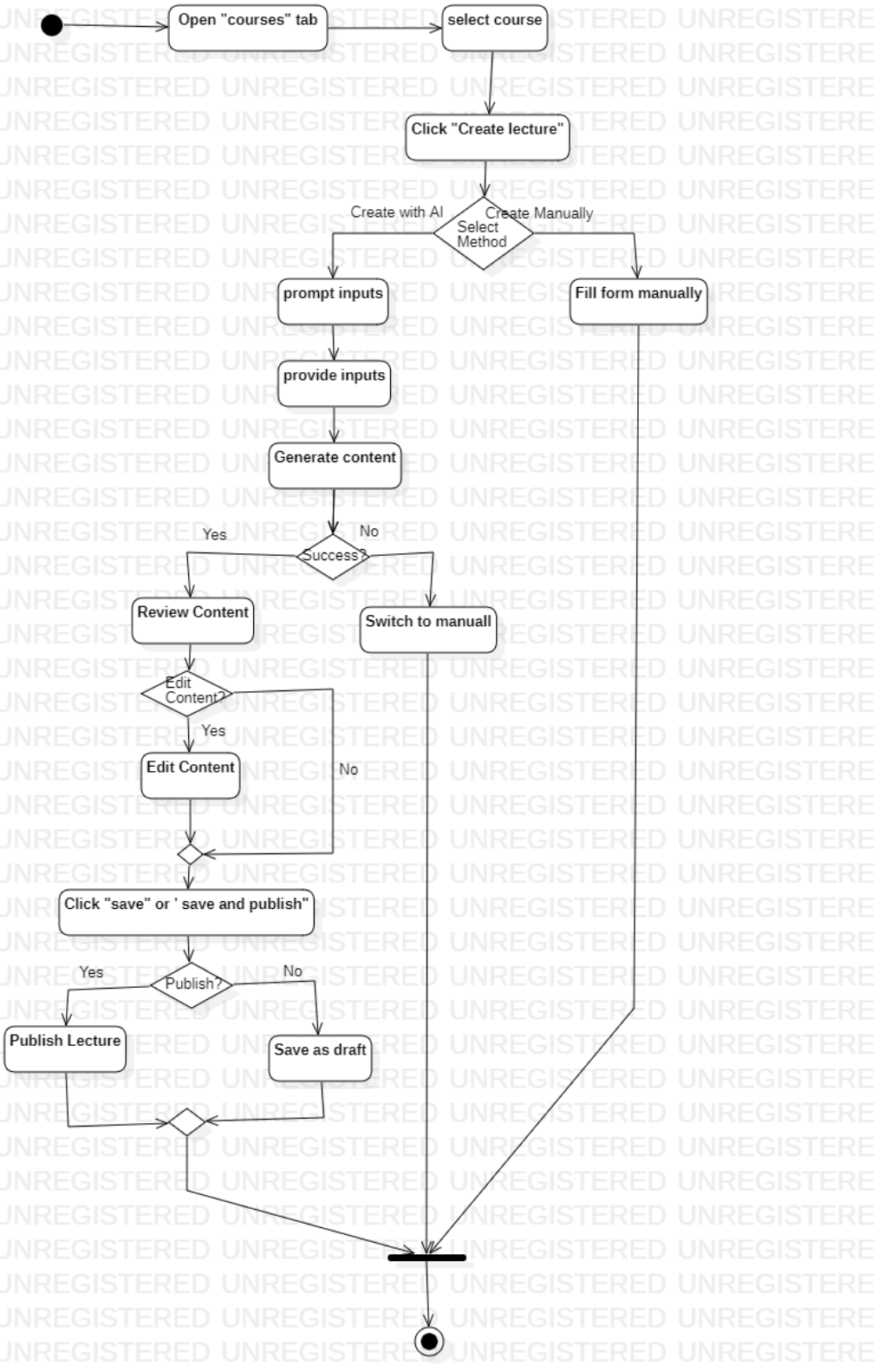


Figure 4.4 Activity Diagram: Create lecture

Activity Diagram: Create Quiz

The teacher creates and schedules a quiz by entering questions, defining correct answers, and assigning marks.

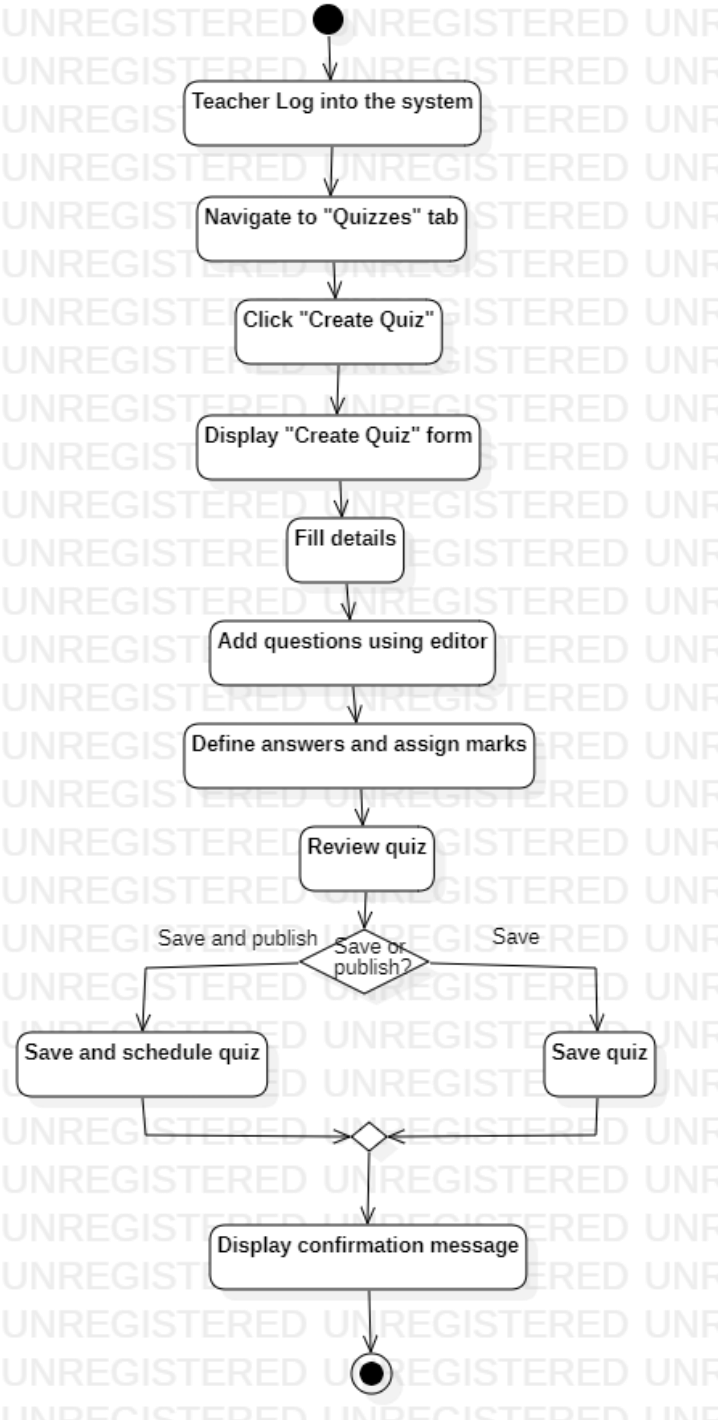


Figure 4.5 Activity Diagram: Create quiz

4.4. Design Models [along with descriptions]

For our project, we are utilizing an object-oriented approach, and therefore, the following design models are included: The applicable models may include:

Sequence diagram: Content Generation

This sequence diagram illustrates the process of lecture creation, review, publishing, and access in a Learning Management System (LMS), showing interactions between Teachers, Students, the LMS Frontend, Backend, Database, and Content Generation module (LLM).

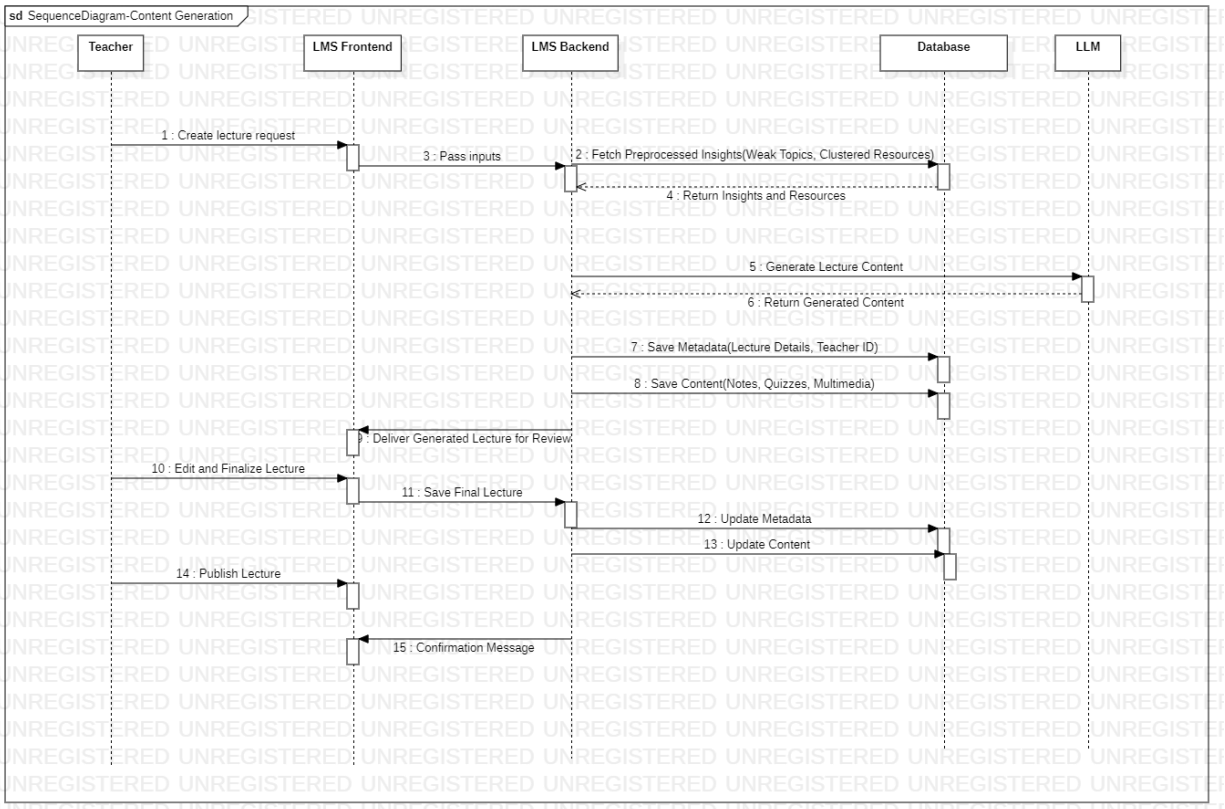


Figure 4.6 Sequence diagram for Content Generation

System Sequence Diagram: Create Lecture

This diagram illustrates the process of lecture creation access in a Learning Management System (LMS), showing interactions between Teachers, the LMS system.

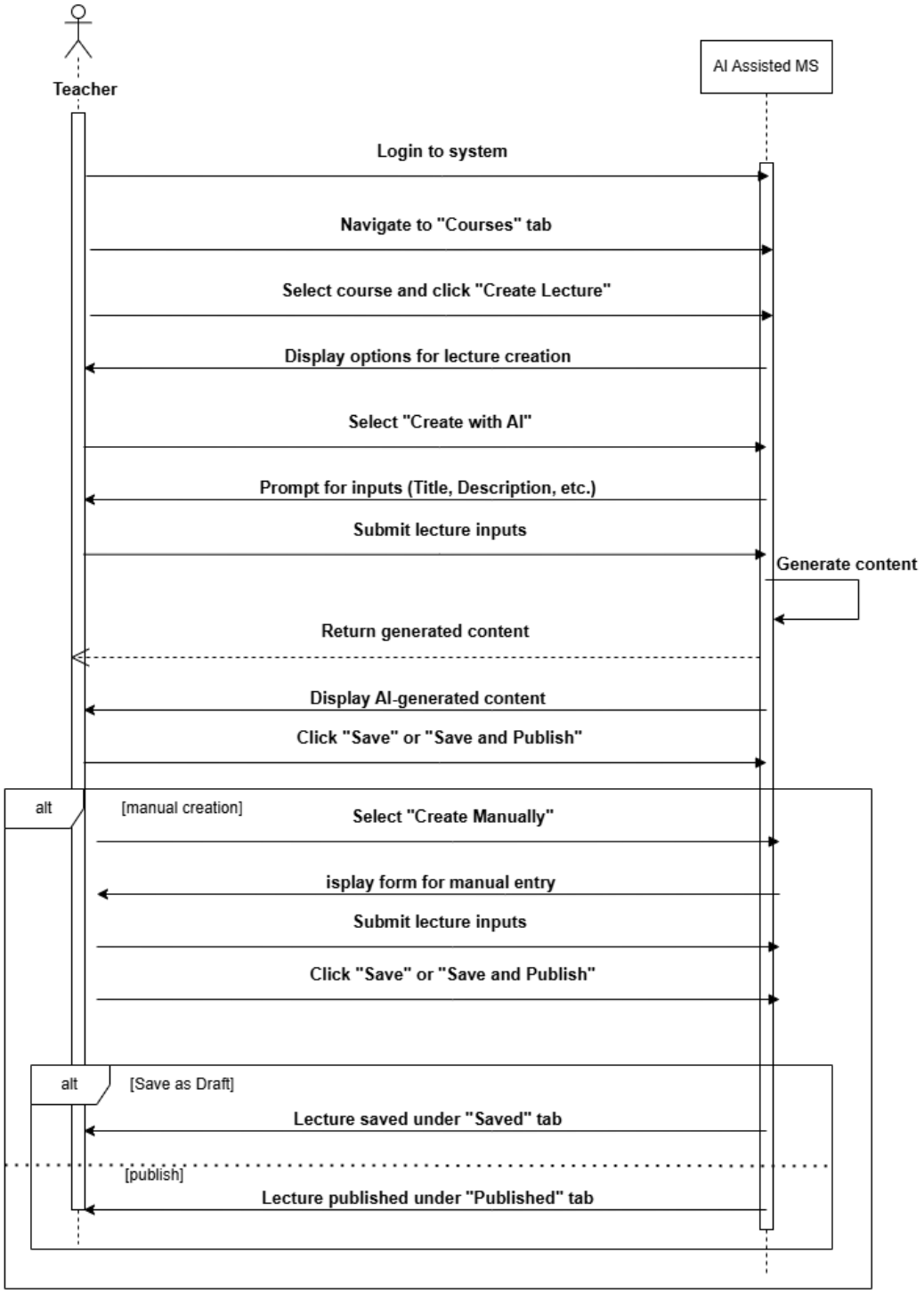


Figure 4.7 System Sequence Diagram: Create Lecture

Human interface design

The system ensures a user-centered interface, focusing on intuitive navigation, accessibility, and responsive design to enhance user experience. Following are the screen images of AI assisted LMS:

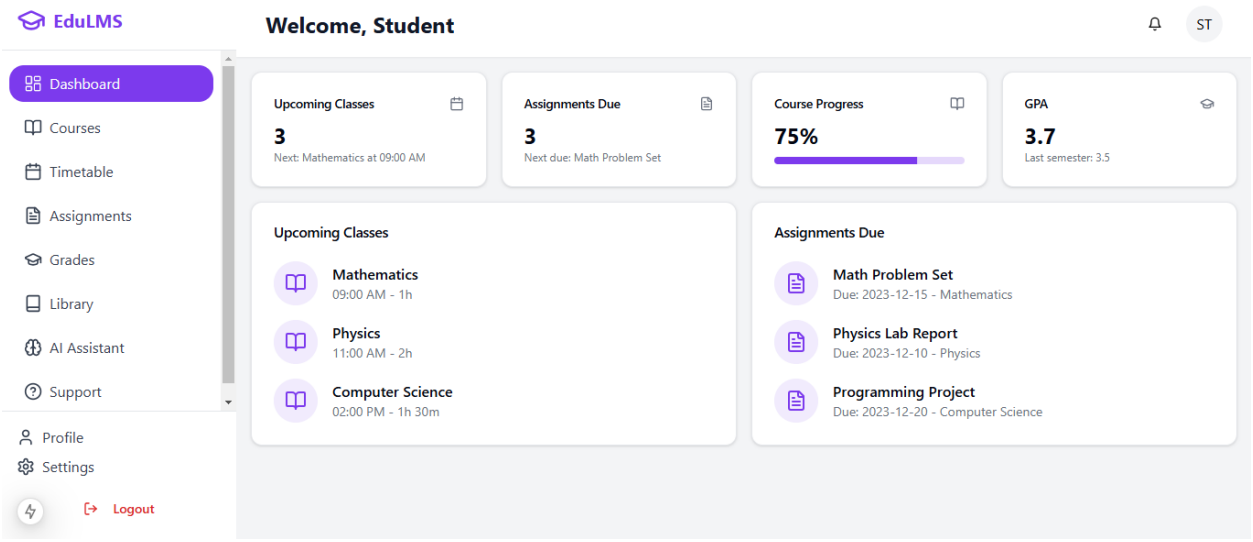


Figure 4.8 LMS Dashboard

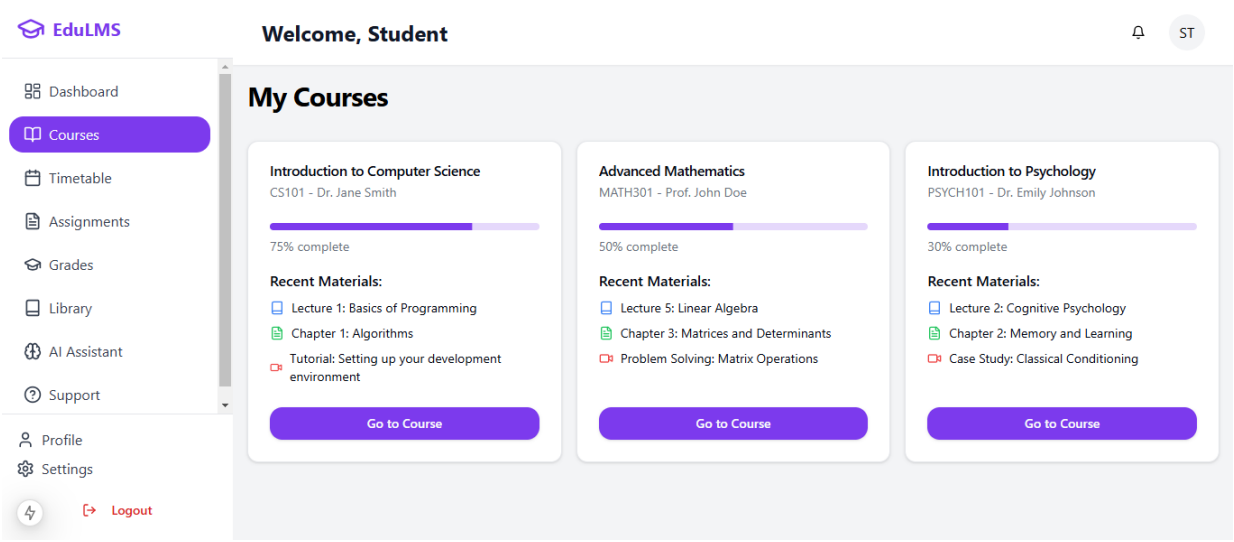


Figure 4.9 Enrolled Courses

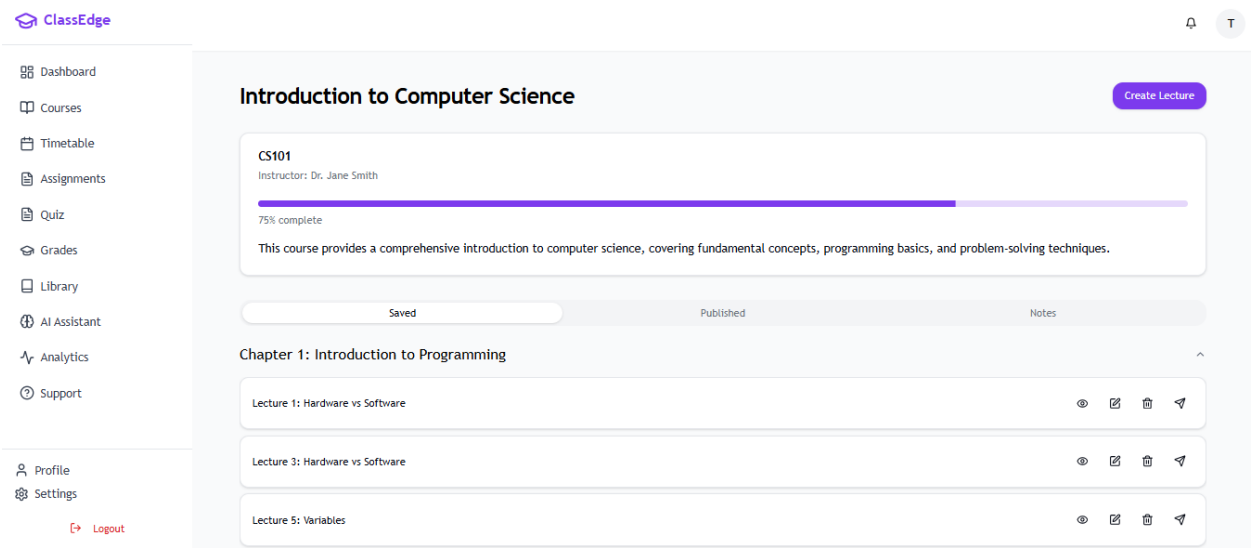


Figure 4.10 Specific course screen

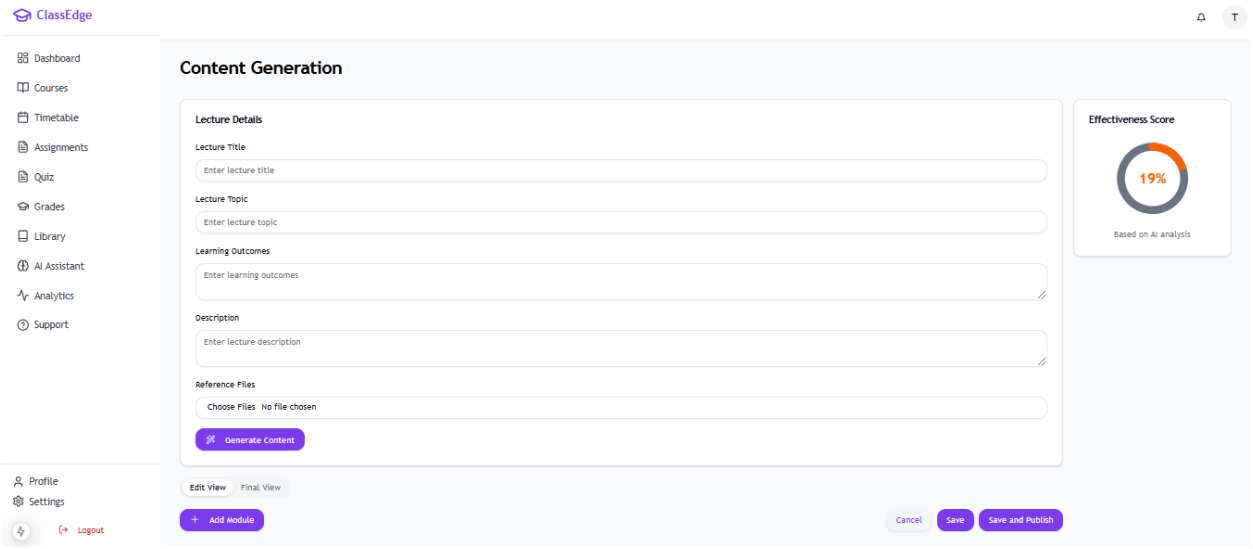


Figure 4.11 Content generation screen

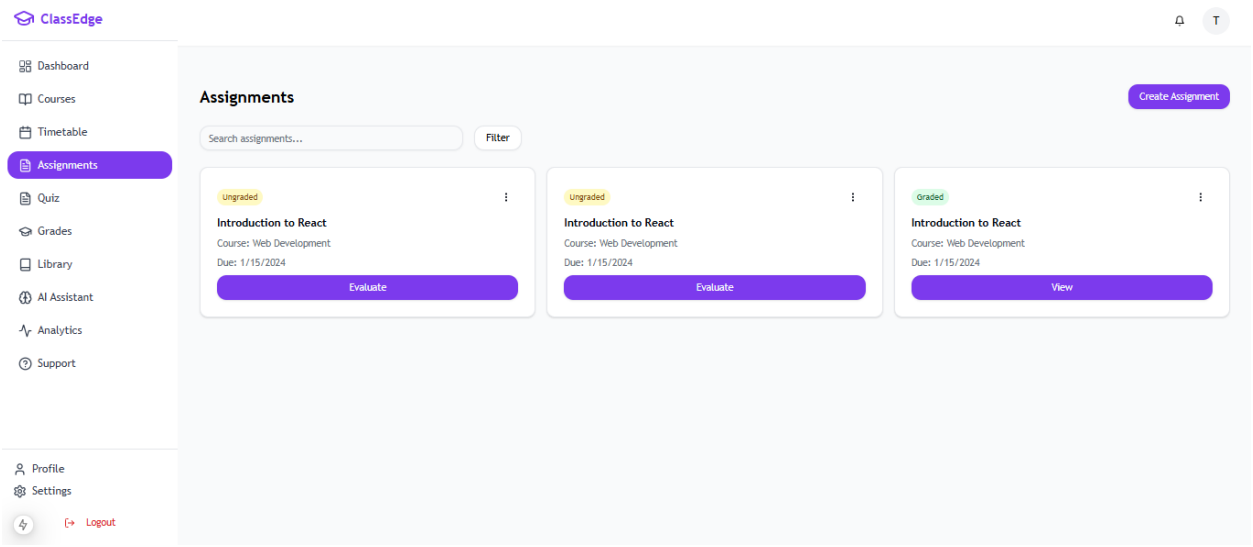


Figure 4.12 Assignment screen

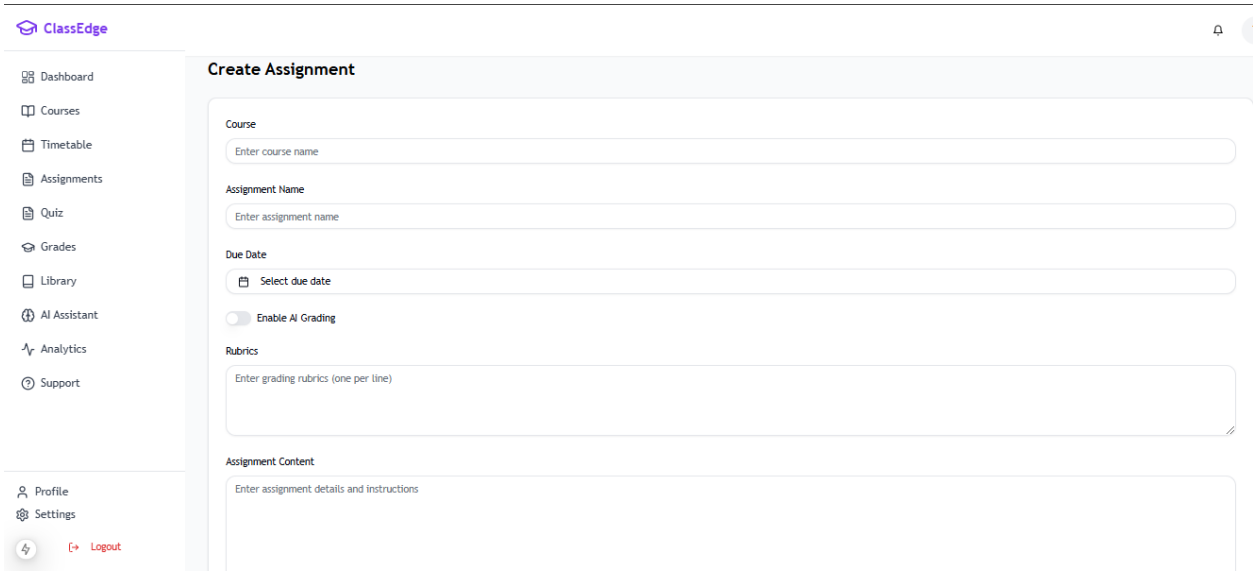


Figure 4.13 create Assignment screen

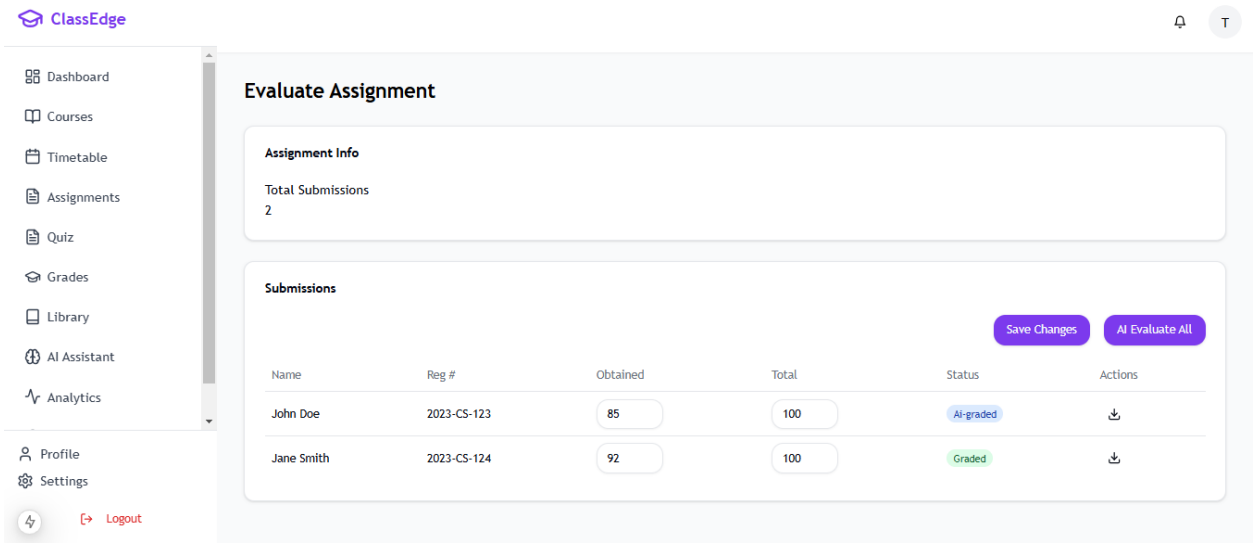


Figure 4.14 Evaluate assignment screen

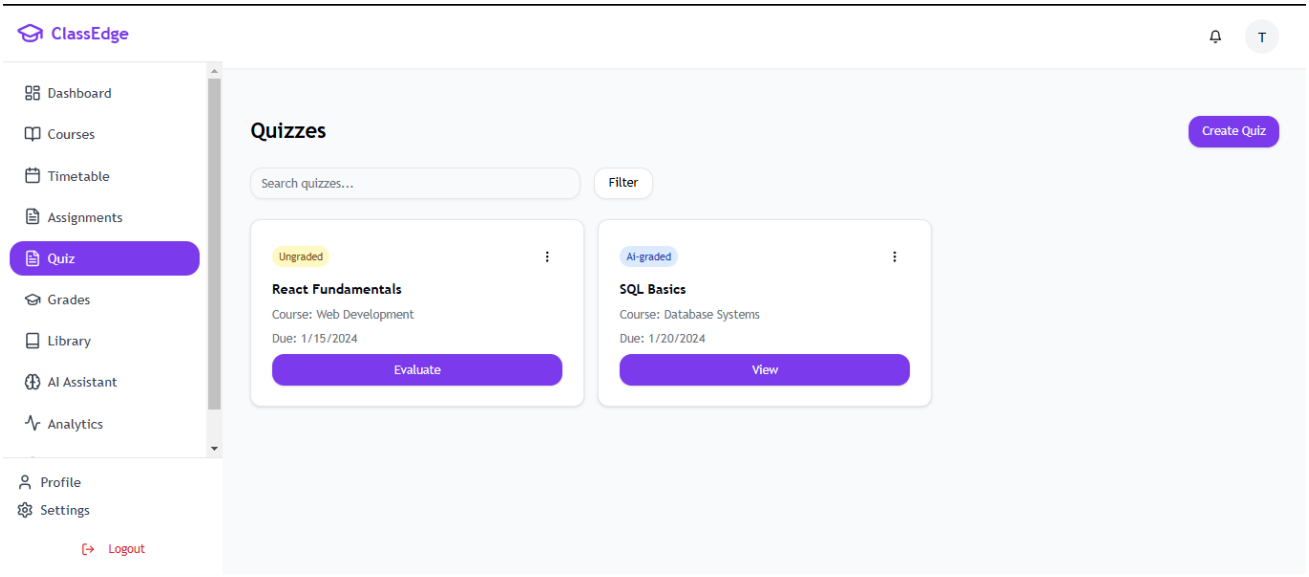


Figure 4.15 Quizzes screen

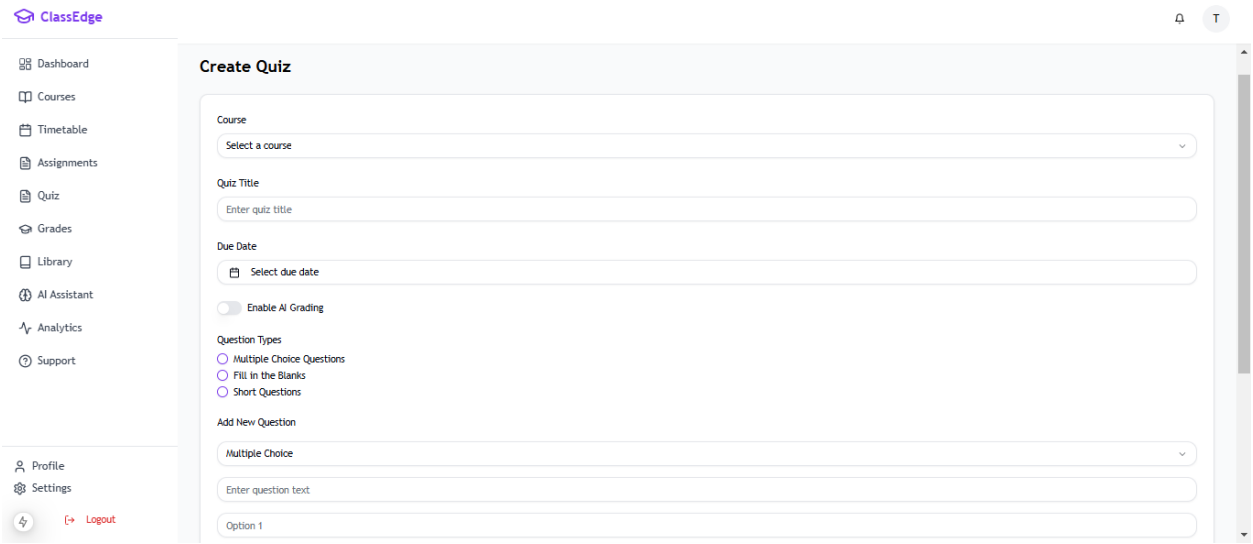


Figure 4.16 Create quiz screen

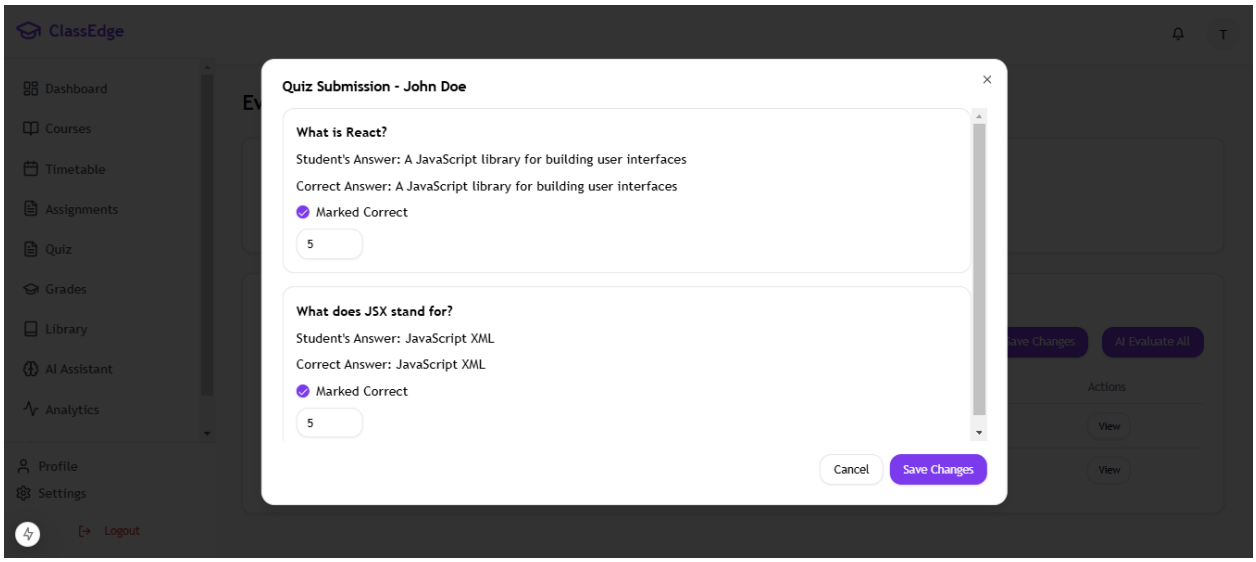


Figure 4.17 Quiz submission screen

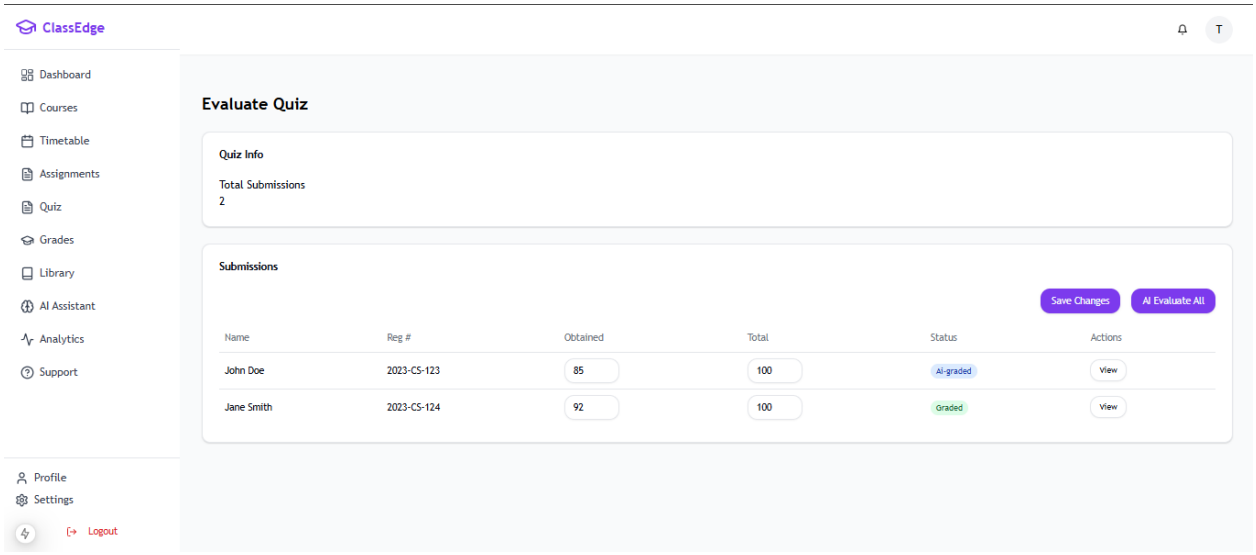


Figure 4.18 Evaluate quiz screen