Table of Contents

1. Diagram 2

1.1 Description 3

2. Functional Requirements 4

2.1 User Management 4

2.2 User Profile Update 4

2.3 Course Management 4

2.4 Recommended Courses for Specialization 4

2.5 Professor Recommendation System 4

2.6 Payment & Subscription Management 4

2.7 Content Management 4

2.8 Assignment 4

2.9 Assessment 4

2.10 RAG Implementation for Descriptive Questions 4

2.11 Assessment Creation 5

2.12 Grading System 5

2.13 Attendance Tracking 5

2.14 Discussion Forums & Chat 5

2.15 Progress Tracking & Reports 5

2.16 Feedback & Survey System 5

2.17 Customer Care 5

2.18 Notifying Actionable to Admin 5

3. Nonfunctional Requirements 6

3.1 Performance 6

3.2 Scalability 6

3.3 Availability 6

3.4 Security 6

3.5 Reliability 6

3.6 Usability 6

3.7 Maintainability 6

3.8 Compatibility 6

3.9 Response Time 6

4. External Interface Requirements 6

4.1 User Interfaces 7

4.2 External System Interfaces 7

4.3 System-to-System Interfaces 8

5. Development Phases 8

5.1 Phase 1: Core Functionalities (MVP Development) 8

5.2 Phase 2: Advanced Learning Features 8

5.3 Phase 3: AI-Powered Enhancements & Automation 9

6. Member contribution table 10

**1 Diagram:**A screenshot of a computer screen

AI-generated content may be incorrect.

**Parallelograms –** Inputs in the Web UI.

**Rectangles –** process

**Cylinders –** Entities in the Database

**1.1 Description:**

Students create (Register) Student accounts, while Admin credentials are directly given by the code. Instructor Accounts are registered by the Admin. After logging in, both students and Instructors can update their profiles.

The courses are allotted to the Instructor by the Admin by filling the form in his Account and he can decide the strength of students under each Instructor however the students’ courses are added only after they pay the bill of the vacant courses.

The materials are uploaded by the lecturer and can be downloaded and displayed to the students and that instructor.

The Assignments are created by the lecturer which answers will be uploaded to the student users and some of them that are descriptive are corrected by the AI with the similarity scores and this feature optional according to the lecturer interest.

Roll call attendance form will be given to the instructor and based on the Instructor’s filled form the attendance table is updated.

The customer care is an LLM application based chatbot which answers both students and Instructors’ questions and find the actionable items and send notifications to the Admin.

**2 Functional Requirements:**

**2.1 User Management**

* Allow students, instructors, and admins to create and manage accounts.
* Role-based access control for different user types.

**2.2 User Profile Update**

* Allow the user to update their profile with the user information.
* Private and Public mode for user Information.

**2.3 Course Management**

* Admin can create, edit, and delete courses.
* Students can enroll in and access course materials.

**2.4 Recommended Courses for Specialization**

* Customized Student Recommendation System

**2.5 Professor Recommendation System**

* Professor score based on Spot Evaluation.

**2.6 Payment & Subscription Management**

* Enable paid courses and subscription models.

**2.7 Content Management**

* Upload and organize multimedia content (videos, PDFs, quizzes, etc.).
* Provide downloadable course materials to Students.

**2.8 Assignment**

* Allow instructors to create assignments.
* Enable students to submit assignments online.

**2.9 Assessment**

* Jumbled quiz questions (MCQ (jumbled choices), True or False).
* Descriptive questions correction using Similarity Search.

**2.10 RAG Implementation for Descriptive Questions**

* Extracting Info from course material PDFs
* Transforming data into Vectors
* Loading into the Vector Database

**2.11 Assessment Creation**

* Form for lecturer to create quiz and Descriptive questions

**2.12 Grading System**

* Auto-grade multiple-choice quizzes.
* Allow manual grading with feedback for assignments or cosine similarity.

**2.13 Attendance Tracking**

* Track student attendance for live sessions.
* Generate attendance reports.

**2.14 Discussion Forums & Chat**

* Enable student-instructor and peer discussions.
* Provide real-time messaging for better collaboration.

**2.15 Progress Tracking & Reports**

* Generate student progress reports and analytics.
* Allow students to track their own progress.

**2.16 Feedback & Survey System**

* Collect student feedback on courses and instructors.
* Provide analytics on survey responses.

**2.17 Customer Care**

* LLM-based customer service

**2.18 Notifying Actionable to Admin**

* Identify actionable queries and report them to the admin.

**3 Non-Functional Requirements:**

**3.1 Performance**

* The system should support **at least 1,000 concurrent users** without degradation.

**3.2 Scalability**

* The LMS should be **easily scalable** to accommodate more students and courses as needed.

**3.3 Availability**

* The system should have to ensure uninterrupted access to learning materials.

**3.4 Security**

* Implement **role-based access control (RBAC)** and **data encryption** for user data protection.

**3.5 Reliability**

* The LMS should **recover automatically** from system failures and backup data every 24 hours.

**3.6 Usability**

* The interface should be **intuitive and user-friendly** for students, teachers, and admins.

**3.7 Maintainability**

* The system should allow **easy updates and bug fixes** without downtime.

**3.8 Compatibility**

* The LMS should work on **all major browsers (Chrome, Firefox, Safari, Edge)** and mobile devices.

**3.9 Response Time**

* Page loads should take **less than 3 seconds**, and course videos should buffer smoothly.

**4 Interfaces of a Student Learning Management System (SLMS)**

Interfaces define how different users and external systems interact with the SLMS. These interfaces ensure smooth communication between users, system components, and external applications.

**4.1 User Interfaces (UI)**

These interfaces allow users (students, instructors, and administrators) to interact with the system.

**A. Student Interface**

* **Dashboard**: Displays enrolled courses, notifications, upcoming deadlines, and recent activities.
* **Course Page**: Allows students to view course materials, participate in discussions, and track progress.
* **Assignment Submission**: Enables students to upload assignments, take quizzes, and receive grades.
* **Profile Management**: Allows students to update their personal information and view academic records.
* **Messaging:** Provides communication with instructors.
* **Discussion Forums**: Supports peer collaboration and instructor-led discussions.

**B. Instructor Interface**

* **Course Management**: Enables instructors to create, modify, and delete courses.
* **Content Upload**: Allows instructors to upload lecture materials (videos, PDFs, slides, assignments).
* **Assignment and Quiz Management**: Facilitates creation, grading, and feedback of assignments and quizzes.
* **Student Performance Monitoring**: Provides analytics and reports on student progress.
* **Communication Tools**: Includes messaging, announcements, and discussion forums.
* **Gradebook**: Allows entry and modification of student grades.

**C. Administrator Interface**

* **User Management**: Allows administrators to add, remove, and manage students and instructors.
* **Course Management**: Enables creation, modification, and deletion of courses.
* **Reporting and Analytics**: Generates reports on student engagement, system usage, and performance metrics.

**4.2 External System Interfaces (APIs & Integrations)**

These interfaces allow the SLMS to interact with third-party applications and services.

**A. Student Information System (SIS) Integration**

* **Data Sync**: Imports and syncs student records, enrolment details, and academic history.
* **Authentication Integration**: Institutional login credentials.

**B. Cloud Storage and File Sharing**

* **Google Drive/OneDrive Integration**: Supports uploading and accessing course materials from cloud storage.
* **File Synchronization**: Ensures real-time document updates.

**4.3 System-to-System Interfaces**

These interfaces facilitate internal interactions between different system components.

**A. Database Interface**

* **Database (NoSQL)**: Stores course content, student records, grades, and logs.
* **Data Query API**: Enables secure retrieval and storage of student performance data.

**5 Development Phases:**

**5.1 Phase 1: Core Functionalities (MVP Development)**

**Objective:** Establish a functional platform with essential features for user management, course management, and basic assessments.

1. **User Management**
   * Student, Instructor, and Admin account creation & management
   * Role-based access control
2. **User Profile Update**
   * Profile editing options
   * Private and Public mode for user information
3. **Course Management**
   * Admin: Create, edit, and delete courses
   * Students: Enroll in and access course materials
4. **Assignment System**
   * Instructors create assignments
   * Students submit assignments online
5. **Assessment**
   * Basic quiz functionality (MCQs & True/False)
   * Auto-grading for MCQs
6. **Grading System**
   * Auto-grade MCQs
   * Manual grading for descriptive answers with feedback
7. **Progress Tracking & Reports**
   * Basic student progress tracking
8. **Customer Care**
   * LLM-based customer service for common queries

**5.2 Phase 2: Advanced Learning Features**

**Objective:** Improve learning experience through personalized recommendations, better assessment mechanisms, and enhanced interaction.

1. **Recommended Courses for Specialization**
   * Implement a student recommendation system based on enrolled courses
2. **Professor Recommendation System**
   * Implement a scoring system based on student spot evaluations
3. **Payment & Subscription Management**
   * Enable paid courses and subscription models
4. **Content Management**
   * Upload and organize multimedia content (videos, PDFs, quizzes)
   * Provide downloadable course materials
5. **Assessment Enhancements**
   * Jumbled quiz questions for MCQs
   * Descriptive question evaluation using similarity search
6. **Attendance Tracking**
   * Track attendance for live sessions
   * Generate attendance reports
7. **Feedback & Survey System**
   * Collect student feedback on courses and instructors

**5.3 Phase 3: AI-Powered Enhancements & Automation**

**Objective:** Implement AI-driven automation for assessments, grading, and administrative tasks.

1. **RAG Implementation for Descriptive Questions**
   * Extract information from course material PDFs
   * Transform data into vectors
   * Load into a vector database for similarity-based evaluation
2. **Grading System Enhancement**
   * AI-based auto-grading using cosine similarity for descriptive questions
3. **Advanced Progress Tracking & Analytics**
   * Generate detailed reports for students and instructors

**6. Member contribution table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Member name** | **Contribution description** | **Overall Contribution (%)** | **Note (if applicable)** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |