

# ***LMS Project Detailed SRS***

---

## ***1. Introduction***

### **1.1 Purpose**

This document provides the **Software Requirements Specification (SRS)** for the LMS platform, which aims to automate learning, content management, and assessments for students, instructors, and administrators.

### **1.2 Scope of the Product**

- **Platform: Web-based Learning Management System (LMS)**
- **Users: Students, Instructors, Administrators**
- **Features:**
  - User registration & authentication
  - Course management (create, upload, enroll)
  - Assessments & grading
  - Reporting & progress tracking
  - Notifications & messaging system

### **1.3 Intended Audience**

- **Project Supervisor: Mam Sara Fareed**
  - **Developers: Frontend (ReactJS), Backend (NodeJS/MongoDB)**
  - **QA/Testers**
  - **Stakeholders (School/College Administrators)**
-

## 2. Overall Description

### 2.1 Product Perspective

- The LMS will be a standalone web application integrated with **MongoDB** for data storage and **NodeJS/Express** backend APIs.
- The web interface will be built with **ReactJS**, fully responsive for desktop and mobile devices.

### 2.2 User Classes

| User Type  | Description                    | Key Functionalities  |
|------------|--------------------------------|--|
| Student    | Learner who enrolls in courses | Enroll, view content, submit quizzes, track progress         |
| Instructor | Educator who manages courses   | Create course, upload content, grade students                |
| Admin      | System manager                 | Manage users, assign roles, monitor system, generate reports |

### 2.3 Operating Environment

- OS: Windows
- Browser: Chrome
- Internet: Required

---

## 3. Use Case Diagrams

### 3.1 Student Use Case Diagram

Actors: Student

Actions: Login, Search Course, Enroll, View Content, Take Quiz, View Grades

*Diagram Suggestion:* LMS system in the center with Student actor linked to all actions.

### 3.2 Instructor/Admin Use Case Diagram

Actors: Instructor, Admin

Instructor Actions: Create Course, Upload Content, Grade Submissions

Admin Actions: Manage Users, Monitor System, Generate Reports

*Diagram Suggestion:* System in center with Instructor & Admin actors linked to respective actions.

---

## 4. Use Case Descriptions

### UC-001: User Login (Student/Admin)

| Field            | Description  |
|------------------|--|
| Use Case ID      | UC-001   |
| Actors           | User (Student/Admin)   |
| Description      | Authenticate user credentials and provide access to the dashboard  |
| Trigger          | User clicks "Login"  |
| Preconditions    | User must be registered  |
| Postconditions   | User session is created, dashboard displayed                       |
| Normal Flow      | User enters credentials → System validates → Redirect to dashboard |
| Alternative Flow | Forgot Password, Invalid Credentials, Account Locked               |
| Business Rules   | Max 5 failed login attempts → Account locked                       |

### UC-002: Course Enrollment

| Field          | Description   |
|----------------|---|
| Use Case ID    | UC-002  |
| Actors         | Student   |
| Description    | Enroll in a course and update the database  |
| Trigger        | Student clicks "Enroll" on selected course  |
| Normal Flow    | Student selects course → System checks availability → Enrollment recorded → Confirmation sent |
| Postconditions | Student added to course, progress tracking activated  |
| Exceptions     | Course full → Enrollment denied   |

### UC-003: Submit Assignment/Quiz

| Field       | Description  |
|-------------|--|
| Use Case ID | UC-003   |
| Actors      | Student  |
| Description | Submit assessment and system automatically grades multiple-choice questions      |
| Normal Flow | Student uploads submission → System validates → Score calculated → Feedback sent |

## 5. Functional Requirements

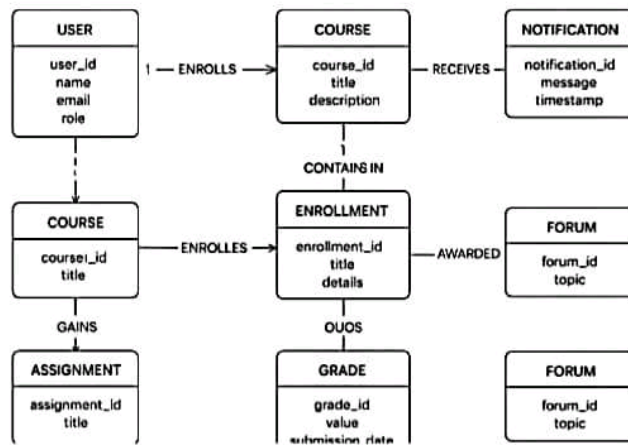
| ID     | Requirement         | Rationale                              | Priority |
|--------|---------------------|--|----------|
| FR-001 | User Authentication | Secure login for students/admins       | High     |
| FR-002 | Course Management   | Instructor can create & upload courses | High     |
| FR-003 | Enrollment Tracking | Record student enrollment              | High     |

|        |                     |                                       |        |
|--------|---------------------|---------------------------------------|--------|
| FR-004 | Content Delivery    | Allow students to view course content | High   |
| FR-005 | Assessment Handling | Submit & auto-grade quizzes           | High   |
| FR-006 | Reporting           | Generate performance & course reports | Medium |
| FR-007 | Notifications       | Email/push notifications for updates  | Medium |

## 6. Non-Functional Requirements

| Category        | Requirement     | Description  |
|-----------------|-----------------|--|
| Performance     | Response Time   | System should respond within 2 seconds for content loading |
| Security        | Data Encryption | Sensitive data stored with AES-256 encryption              |
| Usability       | Accessibility   | WCAG 2.1 compliance, responsive UI                         |
| Maintainability | Modular Code    | Code structured in components for easy updates             |
| Reliability     | Backup          | Daily database backups, 99.5% uptime                       |
| Portability     | Cross-Browser   | Supports latest versions of Chrome, Firefox, Edge          |

## Diagrams



## 1. Definitions, Acronyms, and Abbreviations

| Term | Meaning                              |
|------|--------------------------------------|
| LMS  | Learning Management System           |
| UI   | User Interface                       |
| API  | Application Programming Interface    |
| WCAG | Web Content Accessibility Guidelines |



## 2. System Features

Expand each feature into sub-features:

- **User Registration & Authentication**
    - Password recovery, role-based access.
  - **Course Management**
    - Categorization of courses, prerequisites, versioning of content.
  - **Assessments & Grading**
    - Multiple types of quizzes, assignment submission, grade calculation.
  - **Progress Tracking**
    - Dashboard for students, instructor analytics.
  - **Notifications & Messaging**
    - Alerts for assignment deadlines, system messages.
- 

## 3. External Interface Requirements

- **User Interfaces**
    - Login page, Dashboard, Course page, Quiz page.
  - **Hardware Interfaces**
    - Optional: if mobile app integration.
  - **Software Interfaces**
    - Database (MongoDB), Backend APIs (NodeJS/Express), Frontend (ReactJS).
  - **Communication Interfaces**
    - Email, SMS, Push notifications.
- 

## 4. Data Flow Diagrams (DFD)

|   |
|---|
| • Level 0: System overview.   |
| • Level 1: Student actions (login → enroll → submit assignments → view grades).                   |
| • Level 1: Instructor/Admin actions (create course → upload content → grade → report generation). |

5. Entity-Relationship Diagram (ERD)

- Entities: Student, Instructor, Admin, Course, Assignment, Quiz, Enrollment, Grade.
- Relationships: Student enrolls in Course, Instructor manages Course, Course has Assignments & Quizzes, etc.

6. Requirements Traceability Matrix (RTM)

| Requirement ID | Description         | Source      | Status      |
|----------------|---------------------|-------------|-------------|
| FR-001         | User Authentication | Stakeholder | Implemented |
| FR-002         | Course Management   | Stakeholder | Pending     |

7. Assumptions and Dependencies

- Users have internet access.
- System compatible with latest browsers.
- Database connection will remain stable.

8. Constraints

- System must comply with GDPR or local data protection laws.



- Mobile responsiveness mandatory.
  - Maximum concurrent users: 1000.
-