Software Requirement Specification (SRS)

Project Title: Mini Learning Management System (LMS)

1. Introduction

1.1 Purpose

The purpose of this document is to specify the requirements for a Learning Management System (LMS) to be developed as a 10-day intern project. This system will provide a platform for students and instructors to interact, manage courses, enrollments, and payments, with simple but practical functions.

1.2 Scope

The LMS will allow instructors to create and manage courses, while students can browse, enroll, and make payments (using a Stripe-like payment integration). The system will also support user authentication and basic reporting.

The application will be developed using **MERN stack (MongoDB, Express, React, Node.js)** with clear modularity for future expansion.

2. Overall Description

2.1 User Classes and Characteristics

- **Admin:** Manages the platform, approves/rejects courses, views reports.
- **Instructor:** Creates courses, uploads content, manages enrolled students.
- **Student:** Browses and enrolls in courses, makes payments, views progress.

2.2 System Environment

- Web-based application (React frontend + Node/Express backend).
- MongoDB as database.
- Payment service integrated using Stripe or mock service.

3. Functional Requirements

The system will support the following **core 6 functions**:

1. **User Authentication & Roles**
    - Sign up / login with role (Admin, Instructor, Student).
    - Role-based dashboard.

2. **Course Management**
    - Instructors can create, update, and delete courses.
    - Courses include title, description, category, and price.
3. **Enrollment System**
    - Students can enroll in available courses.
    - Track enrolled students per course.
4. **Payment Integration**
    - Students can make payments for paid courses.
    - Mock Stripe integration for learning purpose.
5. **Content Delivery**
    - Upload/view course materials (PDFs, videos, links).
    - Students can access after enrollment.
6. **Basic Reporting & Dashboard**
    - Admin can view number of courses, total enrollments, and payments summary.
    - Students can view progress (completed vs pending modules).

4. Non-Functional Requirements

- **Performance:** Support at least 10 concurrent users.
- **Security:** Password encryption, secure API.
- **Usability:** Simple and intuitive UI.
- **Maintainability:** Clear documentation and modular code.

5. Deliverables

- Source Code (Frontend + Backend).
- Database schema + ER diagram.
- API Documentation.
- SRS document.
- Demo presentation (day 10).

7. Timeline (10 Days Plan)

- **Day 1–2:** Setup project, database design, ER diagram.
- **Day 3–4:** User authentication + role management.
- **Day 5–6:** Course management + enrollment.
- **Day 7:** Payment integration (mock Stripe).
- **Day 8:** Content delivery feature.
- **Day 9:** Reporting dashboards.
- **Day 10:** Testing, documentation, demo.

8. Conclusion

This SRS outlines the requirements for a mini LMS system that interns can complete within 10 days. It provides hands-on practice with authentication, CRUD operations, payment integration, and reporting in a real-world scenario.