Software Requirements Specification (SRS): LMS Platform

1. Introduction

1.1 Purpose:

This document defines the technical and functional specifications for a Learning Management System (LMS) similar to platforms like Simplilearn. The LMS will enable instructors to upload and manage course content and allow students to enroll, view, and complete courses with video lectures, quizzes, and certification.

1.2 Scope:

The LMS will include features such as user registration, course management, video delivery, authentication, and payment processing. It is intended to be scalable, secure, and performant.

2. Overall Description

2.1 Product Perspective:

This is a web-based platform developed with React.js on the frontend and Node.js + Express.js for the backend. It uses MongoDB as the primary database and integrates third-party services for video delivery, authentication, and payments.

2.2 User Classes:

- **Admin**: Full access to user and course management

- **Instructor**: Create, manage, and monetize courses

- **Student**: Enroll in and consume course content

2.3 Operating Environment:

- **Frontend**: React.js (Vite or CRA), Tailwind CSS

- **Backend**: Node.js with Express.js

- **Database**: MongoDB (Atlas)

- **Video Delivery**: AWS S3 with CDN or Mux

- **Authentication**: Clerk or OAuth (Google, GitHub)

- **Payment**: Stripe (International) or Razorpay (India)

---

3. System Features

- User registration and login via Clerk or OAuth

- Role-based access: student, instructor, admin

- Course creation, editing, publishing by instructors

- Video upload and streaming (via Mux or AWS S3 + CDN)

- Enrollments and progress tracking

- Secure payments and subscription management

- certification generation

- Email notifications for user activity

---

4. External Interfaces

- **Authentication**: Clerk API or OAuth 2.0

- **Video Storage**: Mux API or AWS S3 + Cloudflare/AWS CloudFront CDN

- **Payment Gateway**: Stripe or Razorpay APIs

- **Email**: SendGrid or smtp2go for transactional emails

---

5. Technology Stack

Component	Technology
Frontend	React.js + Tailwind CSS
Backend	Node.js + Express.js
Database	MongoDB Atlas
Auth	Clerk or OAuth
Video Delivery	AWS S3 + CDN or Mux
Payments	Stripe / Razorpay

✅ Basic User Flow – LMS Platform

🎓 Student Flow

1. **Visit LMS platform**

2. **Sign up or log in** using Google, GitHub, or email (via Clerk or OAuth)

3. **Browse available courses**

4. **View course details** (description, instructor, price, rating)

5. **Purchase course** using Stripe or Razorpay

6. **Get access to course content** instantly upon successful payment

7. **Watch videos** (securely streamed via Mux or AWS S3 + CDN)

8. **Mark lessons as completed**

9. **Complete course and download certificate**

10. Optionally **review or rate the course**

👩‍🏫 Instructor Flow

1. **Sign up or log in**

2. **Apply to become an instructor** (optional approval step)

3. **Access instructor dashboard**

4. **Create a new course** (title, description, pricing, category)

5. **Upload video content** (handled via Mux or S3)

6. **Organize content into modules and lessons**

7. **Publish the course**

8. **Track student enrollments and earnings**

9. Optionally **respond to student Q&A or comments**

💁‍♀️ Admin Flow

1. **Log in to admin panel**

2. **Monitor platform activity:**

   - Total users

   - Active courses

   - Revenue analytics

3. **Approve or review course content** (if moderation is needed)

4. **Manage users** (ban, promote to instructor, etc.)

5. **Handle instructor applications**

6. **Oversee financial reports**

7. **Ensure compliance and content quality**