**🔧 System Requirement & Developer Reference Document**

**📌 Overview**

This document outlines the Learning Management System (LMS) architecture and functionalities based on the flowchart provided. It details roles, module interactions, and feature implementations to guide the development team effectively.

**👤 User Roles**

**1. Admin Login**

* Role responsible for backend operations, analytics, and course/user management.

**Modules & Features:**

1. **Manage Users**
   * Add/Edit/Delete users (students, instructors)
   * Assign roles
   * Track user activity
2. **Manage Courses**
   * Add/Edit/Delete course structure
   * Assign instructors to courses
   * Schedule content availability
3. **Analytics**
   * User engagement reports
   * Course completion rates
   * Performance dashboards
4. **System Settings**
   * Manage system configurations
   * Email/SMS notification setup
   * Role permissions & access control

**2. User Login**

Split into two primary user types:

* **Student**
* **Instructor**

**🎓 Student Panel**

1. **View Dashboard**
   * Course overview
   * Notifications
   * Calendar for upcoming tests/classes
2. **Access Courses**
   * Stream course videos
   * Download resources
   * Take quizzes & tests
3. **Track Progress**
   * Progress bar for each course
   * Test/quiz scores
   * Course completion percentage
4. **Give Feedback**
   * Submit feedback for courses/instructors
   * Rate content
5. **Community**
   * Discussion forums
   * Peer-to-peer Q&A
   * Resource sharing

**👨‍🏫 Instructor Panel**

1. **Create/Upload**
   * Upload course material (PDFs, PPTs, videos)
   * Create new modules/sections
2. **Course Content**
   * Organize modules
   * Schedule content release
   * Edit or remove lessons
3. **Assign Quizzes / Homework / Exams**
   * Create timed quizzes
   * Set homework submission deadlines
   * Auto-grading or manual evaluation setup
4. **View Students**
   * List of enrolled students
   * Participation tracking
   * Message individual students
5. **Performance Monitoring**
   * View student analytics
   * Flag struggling students
   * Generate progress reports

**🔗 Development Notes**

**🔹 Frontend Technologies:**

* React.js / Vue.js for dashboards
* REST APIs for modular communication
* Responsive design for all devices

**🔹 Backend Technologies:**

* Node.js / Django for API services
* Role-based authentication (JWT/OAuth)
* MySQL / MongoDB for user and course data

**🔹 Integrations:**

* Video streaming API (e.g., Vimeo, YouTube)
* Notification services (e.g., Firebase Cloud Messaging)
* Analytics tools (e.g., Google Analytics, custom dashboards)

**✅ Next Steps**

* Confirm tech stack
* Define database schema
* API documentation
* UI/UX wireframes for each module
* Milestone planning with deadlines