Project Final Report

Introduction :

This project aims to develop a distraction-free **Learning Management System (LMS)** tailored for educational institutions, businesses, and individual instructors. The LMS will provide a comprehensive platform for **creating, managing, and delivering online courses**, ensuring a seamless learning experience. The system will be built using MySQL for database management, PHP for backend development, CSS for styling, JavaScript for interactivity, and WordPress as the content management framework.

Purpose :

The purpose of this project is to create an **efficient, scalable, and user-friendly** LMS that allows users to:

- Develop and manage courses with multimedia content.
- Track student progress and engagement.
- Facilitate communication between instructors and learners.
- Support online assessments, quizzes, and certifications.
- Provide integration with **payment gateways** for monetized courses.

This LMS will ensure an **interactive, secure, and optimized** e-learning experience for both students and instructors.

Intended Audience :

The LMS will cater to a wide range of users, including:

- **Educational Institutions** – Schools, colleges, and universities looking to manage online learning.
- **Corporate Training Providers** – Companies that require a training platform for employees.
- **Independent Instructors** – Teachers and experts who want to offer online courses.
- **Students & Learners** – Individuals looking to upskill through structured courses.

List of Stakeholders

1. **Project Owner** – The entity funding and overseeing the LMS development. 2. **Administrators** – Manage course content, user roles, and system settings. 3. **Instructors** – Educators who create and manage online

courses.

4. **Students/Learners** – Users enrolling in and engaging with courses.
5. **Developers** – The technical team responsible for building and maintaining the LMS.
6. **UI/UX Designers** – Ensure a smooth and engaging user experience.
7. **System Administrators** – Handle hosting, database management, security, and performance optimization.

Story of This Project

The idea for this project stems from the **growing demand for online education**. Many educators and institutions struggle with **complex, expensive, and rigid LMS solutions** that do not meet their specific needs.

Our goal is to **develop a LMS** that is:

- **Easy to use** for both instructors and students.
- **Customizable** to accommodate different learning needs.
- **Scalable** to support an increasing number of users and courses.
- **Secure** with robust user authentication and role management.

This LMS will serve as a **cost-effective, flexible, and high-performing** platform, allowing educators to efficiently manage and deliver online courses.

Functional and Non-functional Requirements :

Functional Requirements

1. **User Registration & Authentication** – Secure login and role-based

access. 2. **Course Management** – Instructors can create, edit, and publish courses.

3. **Lesson & Quiz Builder** – Support for multimedia lessons, quizzes, and assignments.

4. **Student Enrollment & Progress Tracking** – Students can enroll and monitor their progress.

5. **Payment & Subscription System** – Integration payment gateways.

6. **Certification & Badges** – Auto-generate certificates upon course completion.

7. **Discussion Forums & Messaging** – Enable student-instructor

communication.

8. **Multi-Instructor Support** – Allow multiple instructors per course.

9. **Automated Email & Notifications** – Alerts for enrollments, progress updates, and deadlines.

10. **Content Dripping** – Scheduled release of lessons over time.

Non-functional Requirements

1. **Scalability** – Handle large numbers of users and courses efficiently.

2. **Security** – Enforce authentication, data encryption, and secure payments.

3. **Performance Optimization** – Implement caching and database indexing for faster load times.

4. **Cross-Browser Compatibility** – Ensure compatibility with major browsers. 5.

Mobile Responsiveness – Fully optimized for mobile and tablet users. 6.

Backup & Recovery – Implement regular backups and recovery mechanisms. 7.

SEO Optimization – Improve visibility on search engines for organic reach. 8.

User-Friendly UI/UX – Intuitive navigation for students and instructors.

Software Usage Scenario (From multiple points of view) :

To understand how our Learning Management System (LMS) will function in real-world situations, we will explore its usage from multiple
perspectives, including students, instructors, administrators, and system managers.

1. Student's Perspective

● A student logs into the LMS using secure authentication.

● They browse available courses and enroll in one based on their interest.
● The student accesses course materials, including video lectures, PDFs, and quizzes.
● They complete assignments and participate in discussion forums. ● After finishing a course, the student takes a final quiz and receives a certificate.
● The student can track their learning progress and revisit completed courses.

2. Instructor's Perspective

● An instructor registers on the LMS and sets up their profile.

● They create a new course by adding lessons, multimedia content, quizzes, and assignments.
● The instructor monitors student progress using analytics and engagement reports.
● They interact with students through messaging and discussion forums.
● The instructor updates course content as needed and schedules automated notifications.
● After course completion, the instructor reviews student performances and issues certifications.

3. ***Administrator's Perspective***

● The administrator manages user roles, granting permissions to students and instructors.
● They oversee course approvals, ensuring quality content is delivered. ● The administrator monitors system performance, checking for any technical issues.
● They handle subscription plans and integrate payment gateways for monetized courses.
● The administrator manages backups, security settings, and user complaints.

4. ***System Manager's Perspective***

● The system manager ensures the platform runs smoothly by optimizing databases and server performance.
● They handle security protocols, including data encryption and user authentication.
● The system manager conducts routine maintenance, ensuring the LMS remains up to date.
● They troubleshoot bugs and deploy patches to improve functionality. ● The system manager ensures the system remains scalable to handle a growing number of users.

Software Requirement Elicitation (Quality Function Deployment)

To ensure that our Learning Management System (LMS) meets user expectations and technical requirements effectively, we will implement Software Requirement Elicitation using the Quality Function Deployment (QFD) approach.

Why We Need QFD

We need to apply Quality Function Deployment (QFD) to systematically gather, analyze, and prioritize user needs. This will help us bridge the gap between what users want and the technical features we implement. Since our LMS is designed for educational institutions, corporate trainers, and independent instructors, their requirements must be clearly understood and translated into system functionalities.

How We Will Implement QFD

1. Identify User Requirements – We will collect input from instructors, students, and administrators to understand their expectations, such as ease of course creation, student progress tracking, and secure payments.
2. Develop the House of Quality (HoQ) – We will create a structured matrix to map user needs with technical solutions. This will help us prioritize features such as interactive course management, automated grading, and mobile responsiveness.

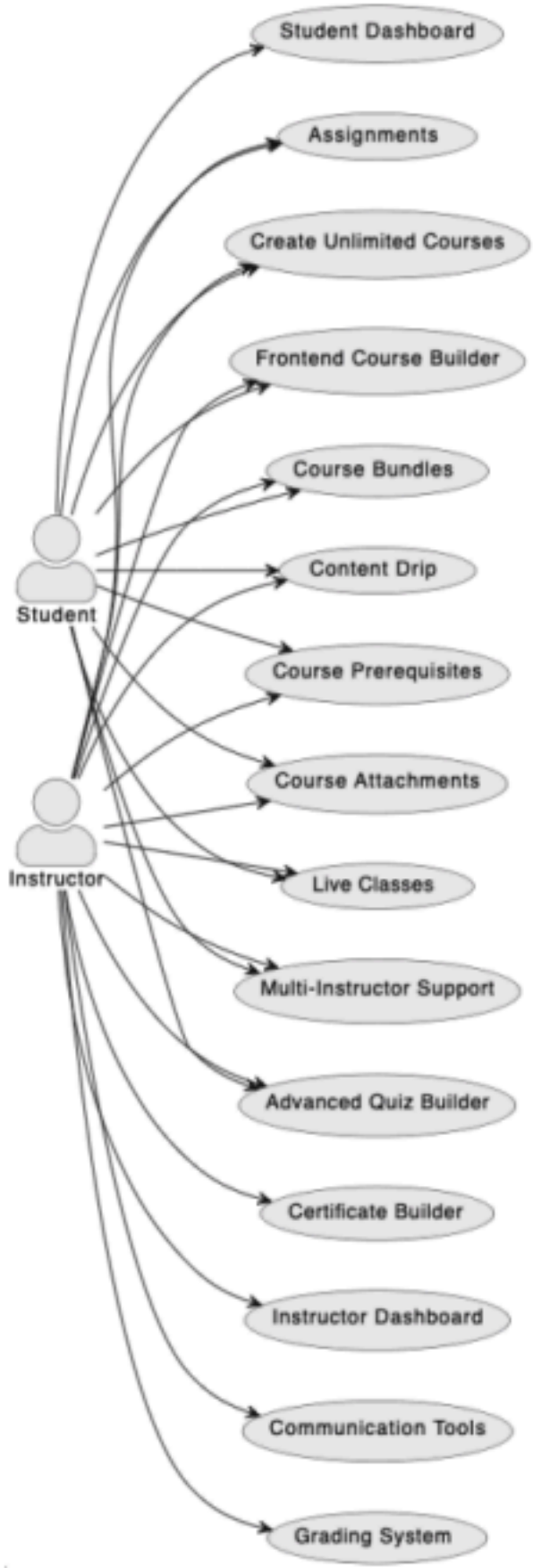
3. Prioritize Key Features – Based on user importance and feasibility, we will focus on essential functionalities like secure login, multimedia course support, certification issuance, and performance tracking.

4. Translate Requirements into Technical Specifications – The gathered requirements will guide our database structure, backend development, and UI/UX design, ensuring a smooth learning experience.

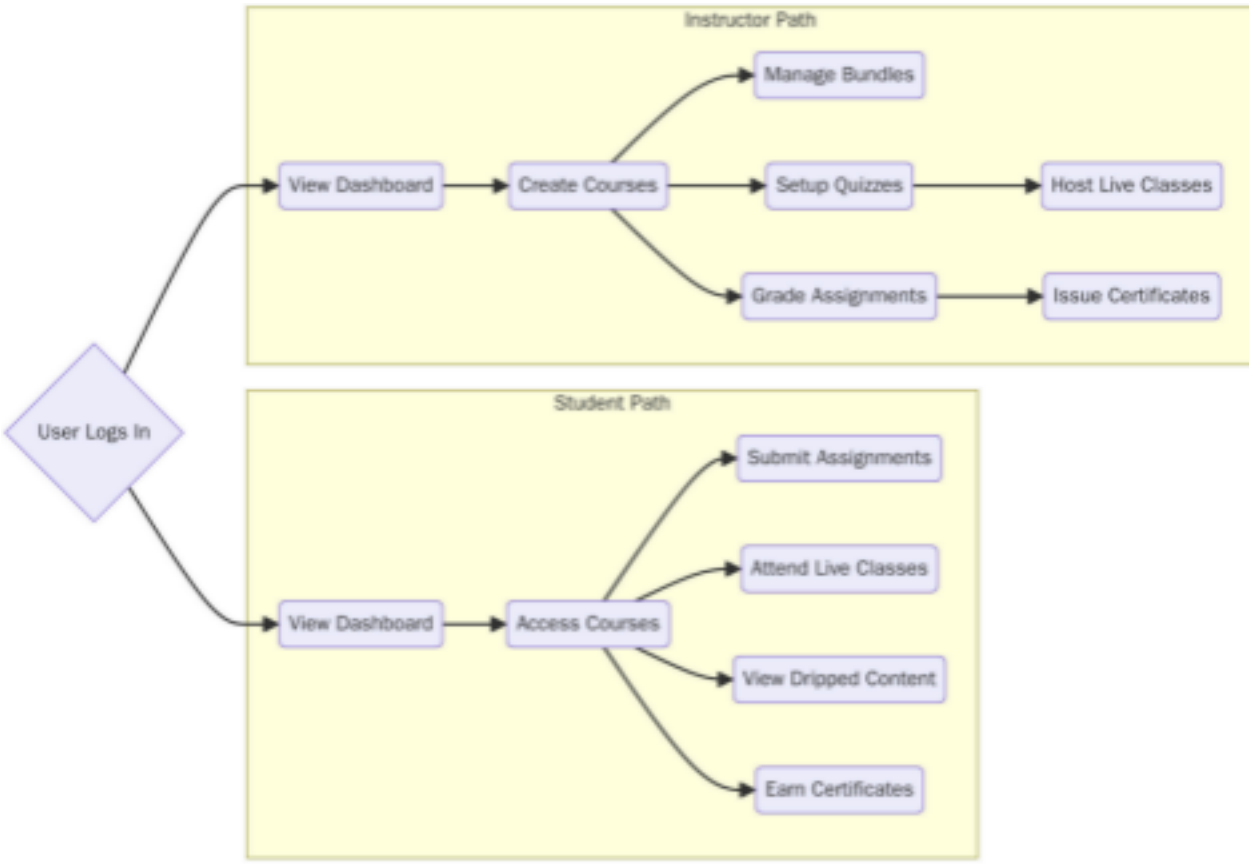
5. Continuous Evaluation – We will refine requirements iteratively based on user feedback and testing, ensuring the LMS evolves to meet real-world needs.

Use Case Diagram :

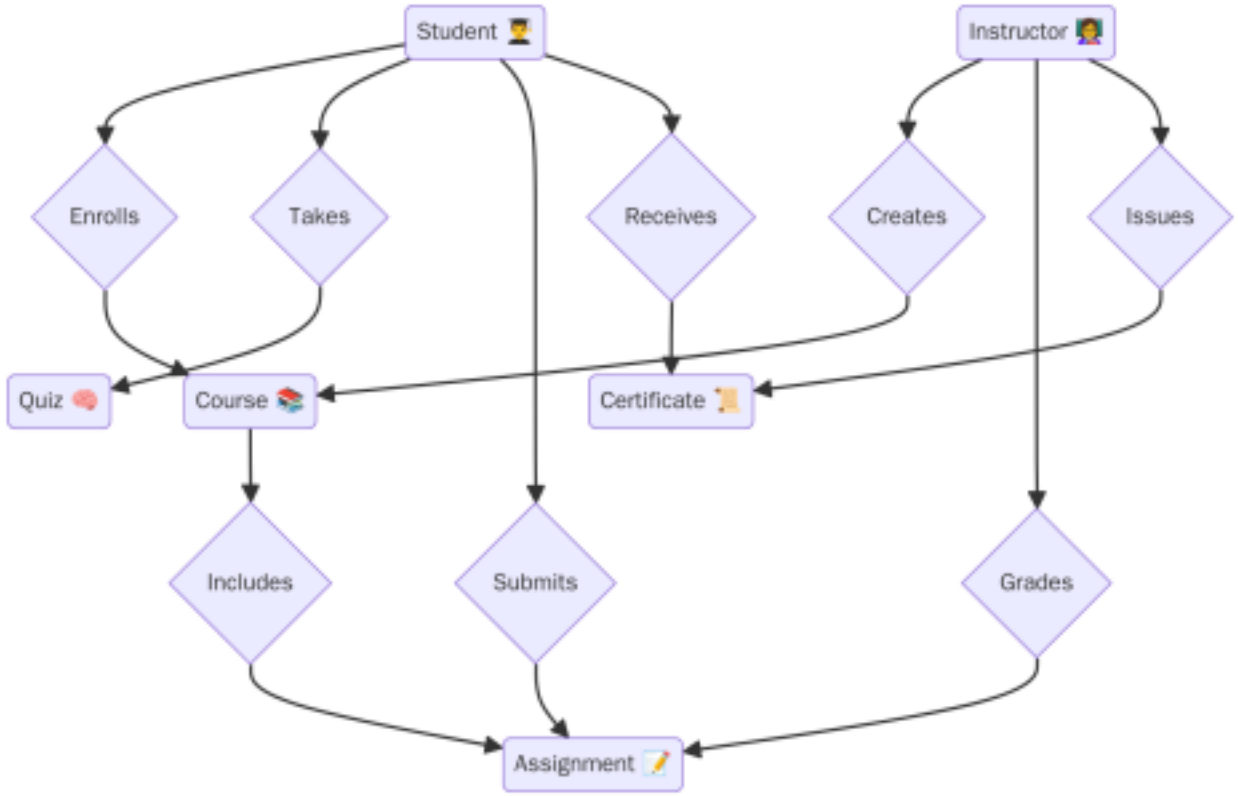

Native eCommerce Integration
Subscription Models
Coupon Management
Tax & Order Management
Detailed Reports
Earnings & Commission Tracking
Exportable Analytics
Two-Factor Authentication
Fraud Protection
Active Session Management
Page Builder Compatibility
Multilingual Support
Third-Party Integrations
Admin


Student Dashboard
Assignments
Create Unlimited Courses
Frontend Course Builder
Course Bundles
Content Drip
Course Prerequisites
Course Attachments
Live Classes
Multi-Instructor Support
Advanced Quiz Builder
Certificate Builder
Instructor Dashboard
Communication Tools
Grading System
Student
Instructor

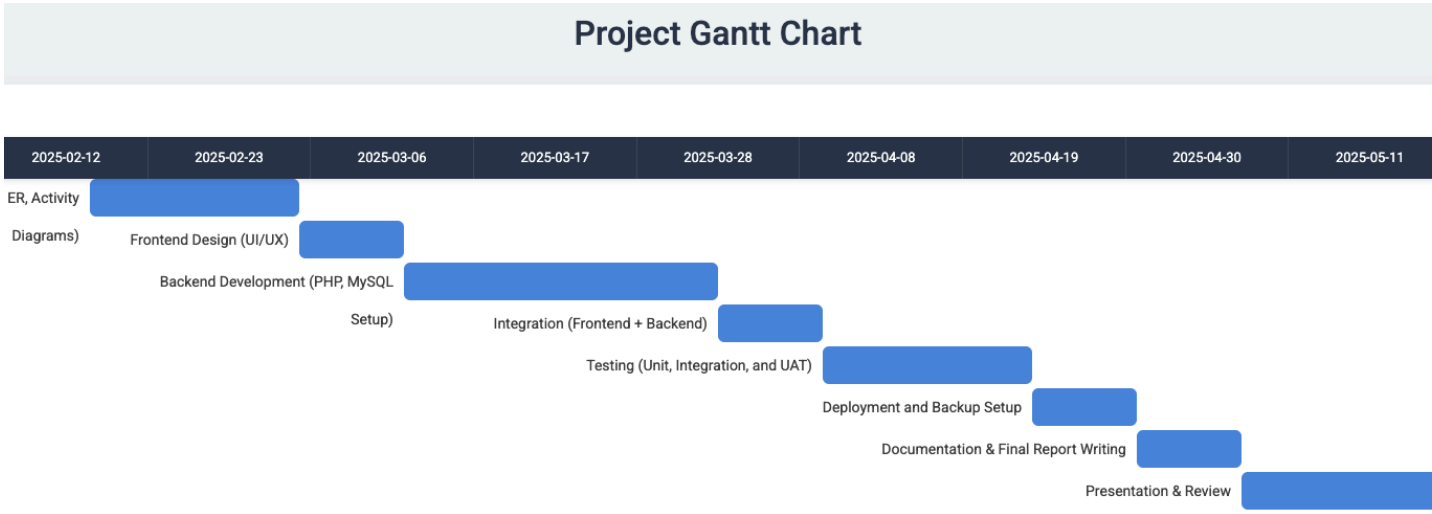
Activity Diagram :



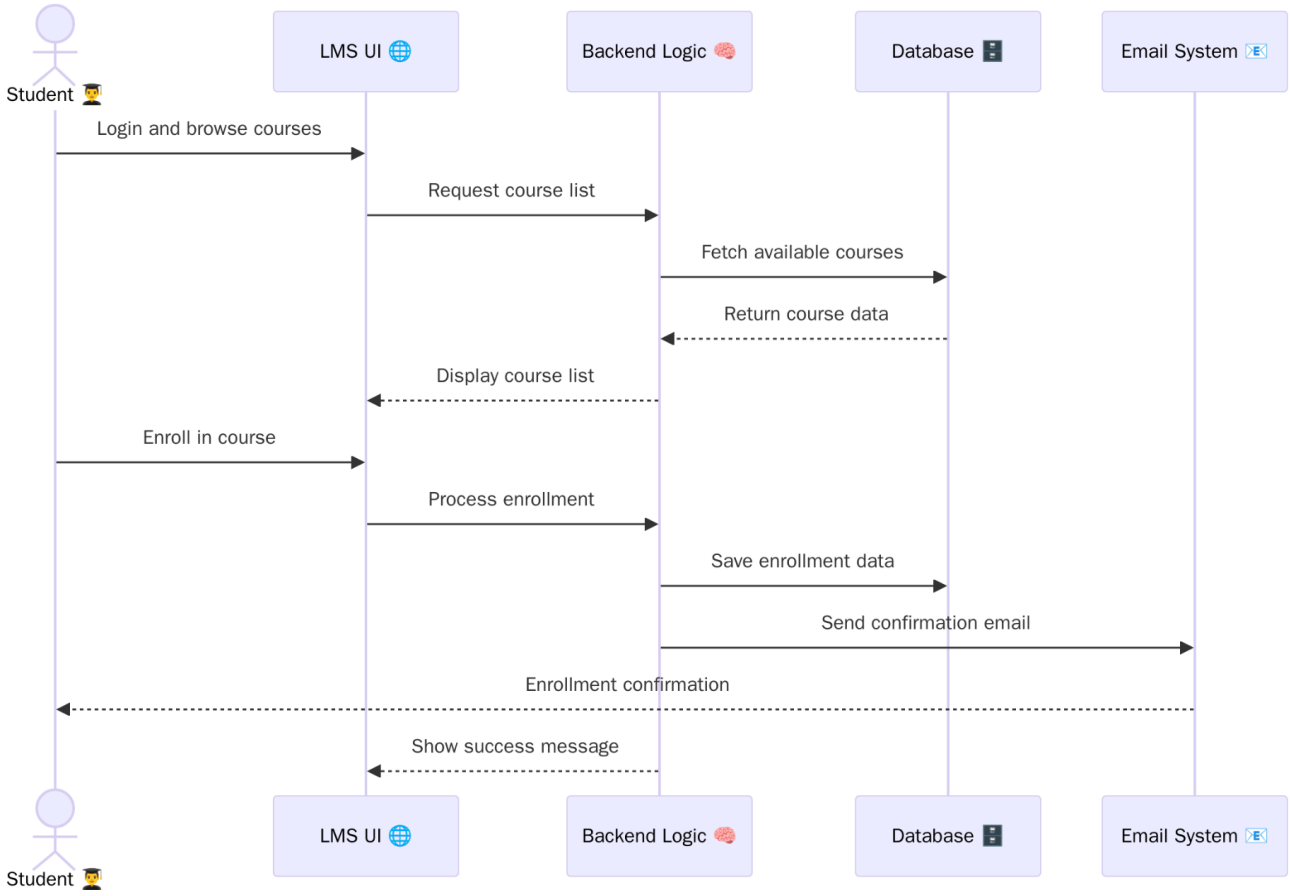
ER Diagram :



Gantt Chart :



Sequence Diagram :



Test Case :

Test Case ID	Test Scenario	Expected Result
TC001	User Registration	Account is created and user receives confirmation email
TC002	User Login	User is logged in and redirected to dashboard
TC003	Course Creation (Instructor)	Course appears in instructor dashboard and public listing
TC004	Course Enrollment (Student)	Course added to student's dashboard
TC005	Quiz Attempt	Results recorded and feedback shown
TC006	Certificate Generation	Certificate available for download
TC007	Payment Integration	Payment successful, course enrolled
TC008	Admin User Management	User status changes appropriately
TC009	Forum Discussion Posting	Comment appears under course discussion board
TC010	Responsive Design Test	All pages display correctly across screen sizes

Which Development Model to Use :

Incremental Model : (Because, It aligns best with our goals, timeline, and resource constraints.)

Why Incremental Model?

- Modular Delivery: We built the system in modules like User Auth, Course Management, Quiz System, etc.
- Early Partial Deployment: Allowed early testing of login/enrollment while other modules were still being developed.
- Flexible to Feedback: We refined features like UI design and course feedback mid-project without affecting prior progress.
- Risk Management: Smaller increments reduced overall complexity and isolated risk in smaller functional blocks.

Project Demonstration :

Technologies Used:

- Frontend: HTML, CSS, JavaScript
- Backend: PHP (Core logic), MySQL (Database)
    - Tools: VS Code, XAMPP, Git

- Deployment: live server

1. Login/Register
    - Student and instructor can sign up or sign in securely.
2. Course Browsing
    - Student can view course list, categories, and details.
3. Enrollment
    - Students click “Enroll” → Entry saved in database.
4. Course Access
    - View lessons, take quizzes.
5. Quiz Attempt
    - Grading logic shows results immediately.
6. Certificate
    - Upon completion, students can download certificate.
7. Admin Panel
    - View users, block/unblock, manage courses.

Conclusion

This project is designed to deliver a **feature-rich, scalable, and user-friendly LMS** using **WordPress, MySQL, PHP, CSS, and JavaScript**. The system will empower educators and learners by providing an interactive and accessible online learning environment.