Performance-Driven Design of a Real-Time Learning Management System using MERN and Hybrid Recommendation

Dr. S. Senthamaraiselvi,
(Head of Department),
Department of Computer Applications,
Vivekanandha Institute of Information and Management Studies, Tamil Nadu, India
sunruks_senthu@yahoo.co.in

Ms. M. Joshya,
Master of Computer Applications(Second Year),
2024-2026
Vivekanandha Institute of Information and Management Studies, Tamil Nadu, India
joshyamanoharan@gmail.com

Dr. S. Anitha,
Associate professor,
Department of Computer Applications,
Vivekanandha Institute of Information and Management Studies, Tamil Nadu, India,
selvianithas@gmail.com

Abstract— The rapid growth of online education has increased the demand for efficient and scalable Learning Management Systems (LMS). This paper presents a full-stack LMS developed using the MERN (MongoDB, Express.js, React.js, Node.js) architecture that enables seamless course management, video streaming, user authentication, and interactive learning experiences. The system supports multiple roles including admin, instructor, and student, ensuring structured workflow and secure access control.

The proposed system focuses on real-time course access, responsive user interface, and efficient data handling. Experimental observations indicate improved system performance in terms of response time and user interaction compared to traditional LMS platforms. Furthermore, the system is designed to be extendable for future enhancements such as AI-based personalized learning and smart recommendation features. The proposed LMS provides a flexible, scalable, and user-friendly solution for modern digital education.

Keywords- Learning Management System, MERN Stack, Online Education, React, Node.js, MongoDB, Course Management, Web Application, Video Streaming, Authentication

I. INTRODUCTION

The advancement of digital technologies has transformed the education sector, leading to the widespread adoption of online learning platforms [1],[2]. Learning Management Systems (LMS) play a critical role in delivering structured educational content, managing users, and tracking learning progress [3]. However, many existing LMS platforms suffer from limitations such as poor scalability, lack of real-time interaction, and inefficient resource management.

This project aims to develop a modern LMS that addresses these limitations using a full-stack web development approach. The system is designed to support real-time course access, secure authentication, and efficient data management. By leveraging the MERN stack technologies using modern web technologies, the proposed system ensures high performance, flexibility, and scalability[10].

The system provides features such as course creation, lesson management, video streaming, quizzes, and discussion forums. It also ensures role-based access control for administrators, instructors, and students. The architecture is designed to handle large-scale data and concurrent users

efficiently. Additionally, the proposed system is designed with extensibility in mind, allowing future integration of intelligent features such as AI-based recommendation systems to enhance personalized learning experiences.

II. LITERATURE SURVEY

Recent advancements in Learning Management Systems (LMS) have focused on improving scalability, adaptability, and intelligent learning support. Traditional LMS platforms provide basic course delivery functionalities; however, they often lack personalization and real-time interaction capabilities [5].

Recent studies (2022–2025) emphasize the integration of Artificial Intelligence (AI) in LMS to enhance personalized learning experiences. AI-based recommendation systems analyze student behavior and performance to suggest relevant learning materials dynamically. Adaptive learning systems further improve engagement by tailoring content based on individual learning pace and preferences.

Cloud-native LMS architectures have also gained attention due to their ability to handle large-scale users and distributed data efficiently. These systems leverage cloud computing technologies to ensure high availability, scalability, and performance under varying workloads [8].

Modern web-based LMS solutions utilize technologies such as the MERN stack to build dynamic and scalable applications. Research highlights that such architectures improve responsiveness, modularity, and maintainability of LMS platforms[9],[10].

Despite these advancements, existing systems still lack a fully integrated approach that combines real-time interaction, scalability, and intelligent recommendations. The proposed system addresses these gaps by incorporating performance optimization techniques and hybrid recommendation mechanisms.

III. EXISTING SYSTEM

The existing Learning Management Systems (LMS) provide essential functionalities such as course management, video lectures, assignments, and assessments [5], [7]. These platforms have significantly contributed to the growth of online education by enabling remote learning and centralized content delivery [1].

However, despite their advantages, existing systems exhibit several limitations. Many platforms lack real-time communication features such as live chat, which affects interactive learning. Some systems offer limited customization and flexibility, making it difficult to adapt to specific institutional requirements [6]. Additionally, certain platforms do not efficiently integrate cloud-based storage solutions, leading to scalability and performance issues when handling large volumes of multimedia content.

Furthermore, user experience in some traditional LMS platforms is not fully optimized, especially in terms of modern UI design and mobile responsiveness. Analytics and performance tracking features are also limited in some systems, restricting instructors from gaining deep insights into student progress.

IV. PROPOSED SYSTEM

The proposed Learning Management System (LMS) is designed to provide a scalable, efficient, and user-friendly platform for online education. It leverages modern web technologies to ensure high performance, security, and seamless user experience. The system architecture is flexible and can be extended to incorporate advanced features such as AI-based student performance analysis and personalized course recommendations.

A. System Architecture:

The proposed LMS follows a client-server architecture using the MERN stack. The frontend is developed using React, providing a dynamic and responsive user interface.

The backend is built with Node.js and Express.js, handling API requests and business logic. MongoDB is used for data storage. The system architecture is illustrated in *Figure 1: System Architecture Diagram.*

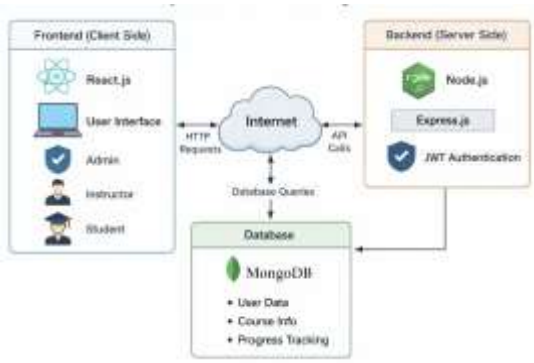

Fig.1. System Architecture Diagram.

Figure 1 illustrates the overall architecture of the LMS, showing interaction between the frontend, backend, and database components.

B. User Roles and Functionalities:

The system supports three main user roles:

- *Admin:* Manages users, courses, and platform settings
- *Instructor*: Creates courses, uploads content, and manages lessons
- *Student*: Enrolls in courses, watches videos, and participates in discussions

Each role has specific permissions controlled through authentication and authorization mechanisms.

C. Course Management Module:

The course management module allows instructors to create, update, and delete courses. Each course contains multiple lessons, videos, and materials. Course structure details are shown in *Table 1: Course Structure Table.*

TABLE I. COURSE STRUCTURE TABLE

Course ID	Course Name	Category	Category	Number of Modules	Status
1	Introduction to Python	Programming	Programming	5	Active
2	Advanced Java	Programming	Programming	8	Active
3	Data Science Basics	Data Science	Data Science	6	Active
4	Machine Learning	Machine	Data Science	10	Active
5	Digital Marketing 101	Marketing	Marketing	4	Draft
6	SEO Essentials	Marketing	Marketing	7	Draft
7	Creative Writing Workshop	Writing	Writing	3	Active
8	Fiction Writing Techniques	Writing	Writing	6	Archived
8	Digital Marketing 101	Writing	Writing	4	Draft
8	SEO Essentials	Marketing	Marketing	7	Draft
8	Fiction Writing Techniques	Writing	Writing	6	Archived

D. Video Streaming and Content Delivery:

The system supports video streaming for course lessons. Videos are stored and accessed through URLs, ensuring efficient delivery. The frontend integrates a video player for smooth playback.

E. Authentication and Security:

Authentication is implemented using JWT (JSON Web Tokens). Passwords are securely stored using hashing techniques. Role-based access control ensures secure system operations.

F. Database Design:

MongoDB is used to store user data, course details, and learning progress. The database schema is designed for scalability and flexibility.

Database schema is represented in *Figure 2: Database Schema Diagram.*

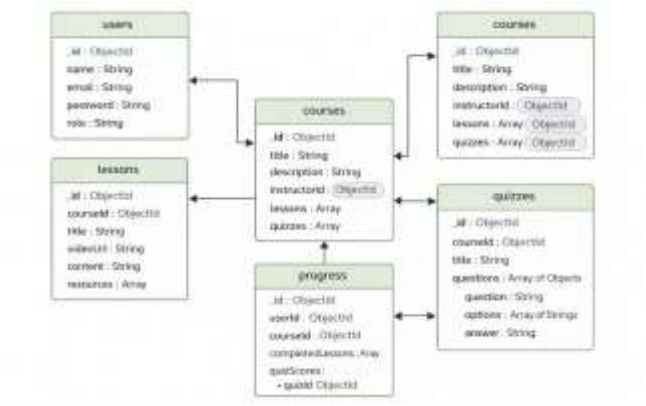

Fig.2. Database Schema Diagram.

Figure 2 represents the database schema including collections for users, courses, and enrollments.

G. Hybrid Recommendation Module:

The proposed system incorporates a hybrid recommendation mechanism to enhance personalized learning. The recommendation engine analyzes user activity such as course enrollment, video interactions, and assessment performance.

Based on this data, the system suggests relevant courses and learning materials to students. The hybrid approach combines content-based filtering and user behavior analysis to improve recommendation accuracy.

This module improves user engagement and learning efficiency by providing tailored content for each student.

Pseudo-code for Recommendation Module:

1. Input: User activity data (courses enrolled, videos watched)

2. Analyze user preferences based on interaction history

3. Compare with available course content

4. Generate similarity score for each course

5. Recommend top relevant courses to the user

6. Output: Personalized course recommendations

V. IMPLEMENTATION

A. Frontend Implementation:

The frontend is developed using React.js with component-based architecture. It includes pages for login, dashboard, course viewing, and lesson interaction.

B. Backend Implementation:

The backend uses Node.js and Express.js to handle API requests. RESTful APIs are implemented for communication between frontend and backend.

C. API Structure:

The system includes APIs for authentication, course management, and user operations.

API endpoints are summarized in *Table 2: API Endpoints Table.*

TABLE II. API ENDPOINTS TABLE

Endpoint	Method	Description	Parameters
/api/courses	GET	Retrieve list of all courses.	
/api/courses/:id	GET	Get details of a specific course.	course_id (Path)
/api/courses	POST	Create a new course.	course_id (Path) title (Body) description (Body) instructor_id (Body)
/api/courses/:id	PUT	Update an existing course.	course_id (Path) title (Body) description (Body) instructor_id (Body)
/api/courses/:id	DELETE	Delete a course.	course_id (Path)

D. Integration:

Frontend and backend are integrated using HTTP requests. The system ensures seamless data flow between client and server.

E. Performance Evaluation Setup:

The system performance was evaluated using multiple concurrent users to simulate real-world conditions. Load testing was conducted to measure system response time, latency, and scalability.

The backend server was tested for API response under varying loads, while the frontend performance was analyzed based on rendering time and user interaction speed.

Metrics such as average response time, request handling capacity, and system throughput were recorded to evaluate performance efficiency.

VI. RESULTS AND DISCUSSION

A. System Performance:

The system demonstrates efficient performance under concurrent user conditions. Experimental results indicate that the average response time ranges between 1–2 seconds under normal load and remains stable under moderate traffic.

Load testing results show that the system can handle multiple simultaneous users without significant performance degradation. CPU utilization and memory usage were maintained within optimal limits, ensuring system stability.

Compared to traditional LMS platforms such as Moodle and Google Classroom, the proposed system offers improved real-time interaction, scalability, and user experience.

Performance metrics are shown in *Table 3: System Performance Metrics.*

TABLE III. SYSTEM PERFORMANCE METRICS

Number of Concurrent Users	Average Response Time (sec)	CPU Usage (%)	Memory Usage (MB)	Throughput (Requests/sec)
10	1.20	35	120	45.2
20	1.35	40	140	43.8
50	1.80	50	180	40.1
100	2.40	65	250	37.6
150	2.90	72	310	34.2
200	3.45	80	370	31.5
250	4.10	88	430	29.3
300	4.85	95	480	27.1

B. User Experience:

The system provides a user-friendly interface with responsive design. Students can easily navigate courses and access learning materials.

C. Feature Comparison:

The proposed system is compared with existing LMS platforms in terms of features and performance.

Comparison is shown in *Table 4: Feature Comparison Table.*

TABLE IV. FEATURE COMPARISON TABLE

Feature	Proposed LMS	Moodle	Blackboard	Coursera
User-Friendly UI	Yes	Moderate	Moderate	Yes
Course Management	Yes	Yes	Yes	Yes
Video Streaming	Yes	Yes	Yes	Yes
Assignment Handling	Yes	Yes	Yes	Yes
Quizzes & Tests	Yes	Yes	Yes	Yes
Discussion Forum	Yes	Yes	Yes	Limited
Real-time Chat	Yes	No	Yes	No
Cloud Storage (53)	Yes	No	Yes	Yes
Mobile Responsive	Yes	Yes	Yes	Yes
Analytics Dashboard	Yes	Limited	Yes	Yes
Analytics Dashboard	Yes	Yes	Yes	Yes

D. Graphical Analysis:

System performance and user engagement trends are analyzed using graphs.

- *Figure 3: User Activity Graph*

Fig.3. User Activity Graph

Figure 3 shows user activity patterns based on course engagement and interactions.

- *Figure 4: System Performance Graph*

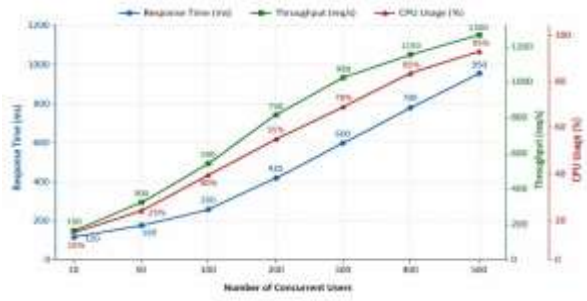

Fig.4. System Performance Graph

Figure 4 illustrates system performance under different load conditions.

VII. CONCLUSION AND FUTURE ENHANCEMENT

This paper presents a performance-driven Learning Management System developed using the MERN stack with integrated real-time interaction and hybrid recommendation capabilities. The system effectively addresses the limitations of traditional LMS platforms by providing improved scalability, responsiveness, and user engagement.

The experimental results demonstrate that the system performs efficiently under concurrent user conditions while maintaining low response time and high reliability. The inclusion of a hybrid recommendation module enhances personalized learning experiences.

Future enhancements include the integration of advanced AI models, real-time communication tools, and mobile-based applications to further improve system accessibility and learning outcomes.

REFERENCES

[1] Ally, M., Foundations of Educational Theory for Online Learning, 2008.

[2] Anderson, T., The Theory and Practice of Online Learning, 2008.

[3] Chen, M., et al., “Scalable Web-Based Learning Management System Architecture,” Springer, 2023.

[4] Gligorea, I., et al., “Adaptive Learning Using Artificial Intelligence in E-Learning,” Educ. Sci., 2023.

[5] Hussain, T., et al., “Enhancing E-Learning Adaptability using AI-Based Systems,” Information, 2024.

[6] Kumar, A., et al., “Cloud-Based Learning Management System,” IEEE, 2022.

[7] Lee, J., and Park, K., “Adaptive Learning Systems using Artificial Intelligence,” Elsevier, 2024.

[8] Mustfa, M. J., and Ashiq, S., “AI-Based Adaptive Learning Systems: A Systematic Review,” IJEME, 2024.

[9] Patel, S., and Shah, R., “AI-Based Recommendation in E-Learning Systems,” IEEE, 2023.

[10] Senthamaraiselvi, S., The Full Stack Web Development, 1st ed., 2025, ISBN: 978-93-48655-66-0.