International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)
ISSN: 2456-236X
Vol. 11 Special Issue 02 | 2026

Self-Learn Learning Management System

Khan Arfaat Firoz[1] ,Shaikh Sameer Shaikh Yunus[2] ,Neha Narayan Khandare[3], Sanchi Ratanlal Tayde[4] , Prof. A. D. Bhople[5]
[1,2,3,4]*Student, Department of Computer Science and Engineering, Padmashri Dr. V. B Kolte College of Engineering, Maharashtra, India*
[5]*Professor, Department of Computer Science and Engineering, Padmashri Dr. V. B Kolte College of Engineering, Maharashtra, India*

DOI:10.5281/zenodo.20454145

ABSTRACT

The integration of artificial intelligence (AI) into learning management systems (LMS) represents a significant advancement in educational technology. This report presents SelfLearn, an AI-integrated LMS developed using the MERN stack, which facilitates interactive learning, real-time course management, and personalized education pathways. The system incorporates an AI chatbot for instant query resolution and a dynamic roadmap generator for tailored learning plans. Key features include secure authentication via JSON Web Tokens (JWT), role-based access control, and modular architecture for scalability. Through detailed analysis of system design, implementation, and evaluation, this work demonstrates how AI enhances user engagement and learning efficiency. The proposed system addresses limitations in traditional LMS by providing adaptive, user-centric functionalities, with implications for higher education and professional training.
Keywords:-Learning Management System; Artificial Intelligence; MERN Stack; Personalized Learning; AI Chatbot; Roadmap Generator; Secure Authentication; Educational Technology

1. INTRODUCTION

Learning management systems have evolved from static repositories of educational content to dynamic platforms that support interactive and personalized learning experiences. The advent of AI technologies has further transformed these systems, enabling adaptive responses to individual learner needs. SelfLearn is designed as a comprehensive LMS that leverages the MERN stack—comprising MongoDB, Express.js, React.js, and Node.js—to deliver a robust, scalable solution for educational institutions and independent learners.
The platform addresses the growing demand for flexible, technology-driven education in an era where remote and self-paced learning are prevalent. By integrating AI components, SelfLearn enhances user interaction through real-time assistance and customized learning paths. This report outlines the system's architecture, implementation details, and potential impacts, providing a blueprint for similar developments in educational software engineering.

Fig-1: Learning Management Systems(LMS)

1.1 Motivation And Objectives

The motivation for developing Self-Learn stems from the limitations observed in conventional LMS platforms, such as insufficient personalization and lack of real-time support, which often lead to reduced learner engagement and suboptimal outcomes. With the proliferation of online education, there is a need for systems that adapt to diverse learning styles and provide immediate feedback.
The primary objectives of the project are as follows:
To design and implement a full-stack LMS using modern web technologies for efficient course management. To integrate AI functionalities that offer personalized learning experiences and interactive assistance.To ensure system security through advanced authentication mechanisms and role-based controls.To evaluate the system's performance in terms of scalability, usability, and educational effectiveness.To contribute to the body of knowledge in AI-enhanced educational tools, facilitating future research and applications.

1.2 Problem Statement

Traditional LMS platforms typically offer standardized content delivery, which fails to accommodate individual learner preferences, paces, and knowledge gaps. This results in challenges such as low retention rates, inadequate progress tracking, and limited administrative oversight. Furthermore, the absence of intelligent features hinders the provision of real-time guidance, making it difficult for users to navigate complex subjects effectively.

2. LITERATURE REVIEW

Existing research on AI-integrated LMS highlights the potential for enhanced personalization and efficiency in education. A review of adaptive learning platforms emphasizes the role of AI in tailoring content based on learner performance, noting improvements in engagement and outcomes. Bibliometric analyses reveal a growing trend in AI-LMS integration, focusing on educational perspectives and implementation challenges. Studies on the future of teaching and learning underscore the need for AI systems that minimize bias and promote fairness. Systematic reviews of AI in higher education LMS applications indicate positive impacts on student management and performance monitoring. Hybrid AI models for academic intervention have been proposed, integrating machine learning techniques to predict and enhance learner success. Additionally, investigations into AI-LMS integration in higher education reveal enhancements in institutional performance and student success.

3. PROPOSED SYSTEM

The proposed system, SelfLearn, is an end-to-end LMS that combines frontend interactivity with backend robustness and AI intelligence. It enables users to enroll in courses, track progress, and receive AI-assisted guidance. Administrators manage content and users through a dedicated dashboard. The system employs a client-server architecture, with React.js handling the user interface, Node.js and Express.js managing server logic, and MongoDB storing data.

AI components are integrated via external APIs to provide chatbot functionality and roadmap generation, ensuring modularity and ease of updates. The system prioritizes user experience with responsive design using Tailwind CSS and Shadcn UI components.

4. SYSTEM ARCHITECTURE

SelfLearn follows a layered architecture typical of MERN applications:

Presentation Layer:- Built with React.js, this layer renders dynamic user interfaces, including course listings, enrollment forms, and AI interaction panels. Tailwind CSS and Shadcn ensure responsive and accessible design.

Application Layer:- Node.js and Express.js form the backend, handling request routing, business logic, and API endpoints. This layer processes user inputs, interacts with the database, and interfaces with AI services.

Data Layer:- MongoDB serves as the NoSQL database, storing schemas for users, courses, enrollments, and progress data. It supports flexible querying and scalability.

AI Integration Layer:- External AI APIs are invoked for chatbot responses and roadmap generation.

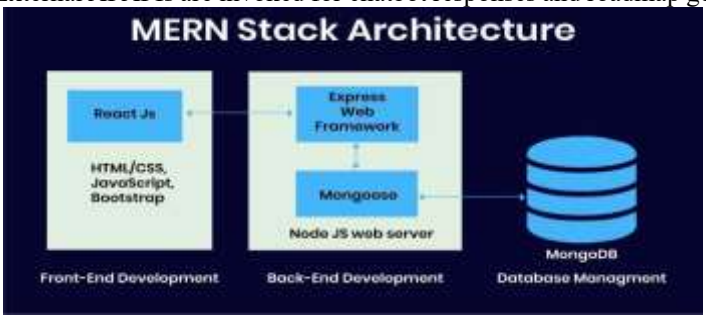

Fig-2: Mern Stack Architecture

5. DATABASE DESIGN

MongoDB is utilized for its schema flexibility, accommodating evolving data structures. Key collections include:

Users:- { _id: ObjectId, username: String, email: String, password: String (hashed), role: String ('user' or 'admin'), createdAt: Date }

Courses:- { _id: ObjectId, title: String, description: String, instructor: String, modules: Array of Objects (title, content), createdAt: Date }

Enrollments:- { _id: ObjectId, userId: ObjectId, courseId: ObjectId, progress: Number (0-100), completed Modules: Array of Strings, enrolledAt: Date }

Roadmaps:- { _id: ObjectId, userId: ObjectId, topic: String, steps: Array of Objects (step, resources: Array of Strings), generatedAt: Date }

Indexes are applied on frequently queried fields like userId and courseId for efficient retrieval. Relationships are managed via references, ensuring data integrity.

6. API DESIGN AND BACKEND WORKFLOW

The backend exposes RESTful APIs for CRUD operations:
User APIs:- POST /api/auth/register, POST /api/auth/login (returns JWT), GET /api/users/:id (authenticated).
Course APIs:- GET /api/courses, POST /api/courses (admin only), PUT /api/courses/:id, DELETE /api/courses/:id.
Enrollment APIs:- POST /api/enrollments, GET /api/enrollments/user/:userId, PUT /api/enrollments/:id/progress.
AI APIs:- POST /api/ai/chat (sends query to AI), POST /api/ai/roadmap (generates plan).
Workflow: Requests are routed through Express.js middleware for authentication. Validated inputs trigger database operations via Mongoose. Responses are JSON-formatted, with error handling for robustness.

- AI Integration and Working Mechanism

AI features are implemented using external APIs, such as those from OpenAI or similar providers, to ensure high-quality natural language processing.
AI Chatbot:- User queries are sent via POST requests to the AI endpoint. The backend proxies the request, appending context from course data. Responses are parsed and displayed in real-time, facilitating concept explanations and question answering.

7. SECURITY AND AUTHENTICATION

Security is enforced through JWT for stateless authentication. Upon login, a token is generated and stored client-side. Subsequent requests include the token in headers, verified by middleware. Role-based access control restricts admin endpoints.
Additional measures include password hashing with bcrypt, input sanitization against injections, and HTTPS for data transmission. Rate limiting and CORS policies mitigate common vulnerabilities.

8. PERFORMANCE AND SCALABILITY CONSIDERATIONS

Performance is optimized via efficient querying in MongoDB, asynchronous operations in Node.js, and frontend code splitting in React.js. Caching with Redis can be added for AI responses.
Scalability is achieved through horizontal scaling of Node.js instances and MongoDB sharding. Load testing ensures the system handles concurrent users, with metrics monitored via tools like PM2.

9. ADVANTAGES OF THE SYSTEM

The system offers personalized learning through AI, improving engagement. Its modular design facilitates maintenance and extensions. Secure architecture protects data, while real-time features enhance usability. Compared to traditional LMS, it provides cost-effective, scalable solutions for diverse educational needs.

10. LIMITATIONS AND FUTURE ENHANCEMENTS

Dependency on external AI APIs introduces potential latency and costs. Limited offline functionality restricts access in low-connectivity areas. The system assumes basic digital literacy, which may exclude some users. Scalability requires infrastructure investment for large deployments.
Future work could include mobile app integration, advanced analytics with machine learning, multilingual support, and blockchain for certificate verification. Enhancing AI with custom models trained on educational data would further personalize experiences.

11. CONCLUSION

SelfLearn represents a robust AI-integrated LMS that advances educational technology by combining MERN stack efficiency with intelligent features. Through its design and implementation, it addresses key challenges in personalized learning, offering a scalable solution for modern education. The system's contributions pave the way for further innovations in AI-driven platforms.

12. REFERENCES

[1] K. A. A. Gamage, "Artificial intelligence-enabled adaptive learning platforms: A review," *Computers and Education: Artificial Intelligence , vol. 6, 2025.
[2] M. A. Conde et al., "Artificial Intelligence (AI) and Learning Management Systems (LMS): A bibliometric analysis," ResearchGate, 2025.
[3] S. S. Saini, "A Comprehensive Survey on AI in Learning Management System," *Preprints.org*, 2025.
[4] U.S. Department of Education, "Artificial Intelligence and the Future of Teaching and Learning," 2023. S. Singh et al., "Leveraging Artificial Intelligence in Learning Management Systems: A Systematic Review of Applications and Impacts in Higher Education," *Indian Journal of Science and Technology*, 2024.
[5] Y. Li et al., "Artificial intelligence in student management systems to enhance academic performance," Scientific Reports, vol. 14, 2024. [Online]. Available: https://www.nature.com/articles/s41598-025-19159-4