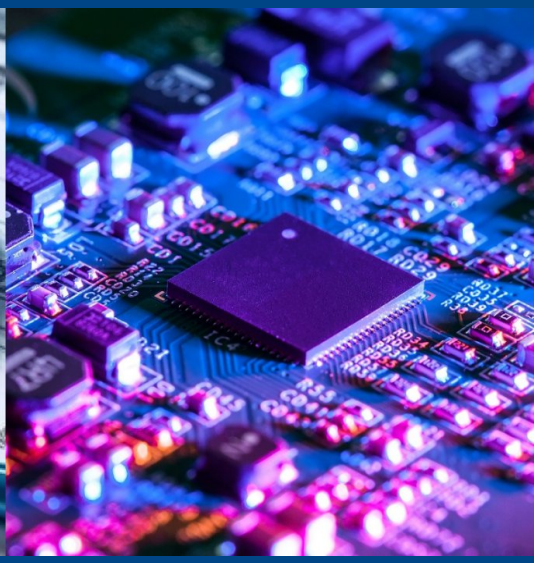
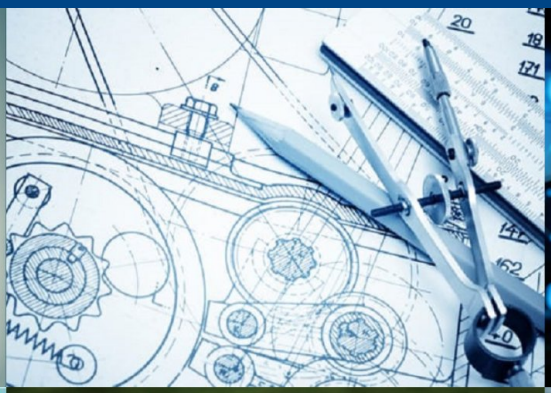
ISSN: 2582-7219

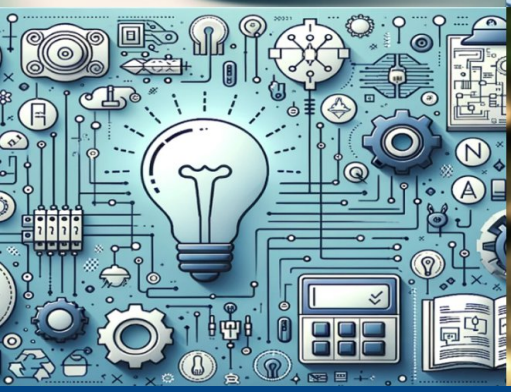




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 4, April 2026**

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904327

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Online Learning Platform with Code Compiler Using MERN Stack

Vishnu Priya, Y. Mohamed Amir, Vinoth Kumar, R. Gowshika

Dept. of Computer Science and Engineering, Mohamed Sathak Engineering College,

Ramanathapuram, India

ABSTRACT: The rapid evolution of web technologies has transformed how education is delivered and consumed. This paper presents Learn-It, a full-stack e-learning web application built on the MERN stack (MongoDB, Express.js, React.js, Node.js) that integrates an in-browser JavaScript compiler to support hands-on coding practice within the same platform. The system offers adaptive learning paths, role-based dashboards for instructors and students, real-time progress tracking, multimedia course content, and an interactive assessment engine. By embedding a JavaScript runtime directly into the student dashboard, learners can write and execute code without switching tools, bridging the gap between theory and practice. The platform is designed to be scalable, secure, and accessible across devices, leveraging RESTful APIs, JWT-based authentication, and a cloud-ready architecture.

KEYWORDS: Online Education, MERN Stack, E-Learning, Code Compiler, Adaptive Learning, Web Application, RESTful API, JavaScript Runtime, MongoDB, React.js

I. INTRODUCTION

The proliferation of internet connectivity and smart devices has created unprecedented opportunities to democratize access to quality education. Online learning platforms have emerged as critical enablers of this shift, allowing learners in geographically remote or financially constrained situations to access curated content from expert instructors. However, most existing platforms treat content consumption and hands-on practice as separate activities, forcing learners to juggle multiple tools — a friction point that significantly reduces learning efficiency.

Learn-It addresses this gap by integrating a live JavaScript compiler directly into the learning dashboard. Students can watch video lectures, read concept notes, attempt quizzes, and immediately test code — all within a single unified interface. The platform is built using the MERN stack, a widely adopted combination of technologies that enables rapid development of performant, full-stack JavaScript applications.

Beyond the compiler feature, Learn-It provides a comprehensive educational ecosystem where instructors create structured courses with multimedia resources, students receive personalized analytics, and administrators oversee platform operations from a centralized console. The system architecture is designed for horizontal scalability, supporting growing user bases without architectural overhaul.

II. LITERATURE SURVEY

A. Cloud-Based E-Learning Models

Murugappan and Mohanapriya (2016) examined the influence of cloud computing on e-learning infrastructure, proposing an application model that decouples content delivery from institutional hardware constraints [1]. Their work underscores the importance of scalable, service-oriented architectures — a principle directly applied in the proposed system's cloud-ready deployment model.

B. Web-Based Learning Management Systems

Phulari et al. (2021) described the Academia platform, an LMS designed to replace manual record-keeping with a secure, interactive teaching-learning environment [2]. Their findings highlight usability and data integrity as primary success factors for student adoption, both of which guided the UX design decisions in Learn-It.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904327

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. MERN Stack in Educational Applications

Pandey, Sahu, and Jaiswal (2022) demonstrated that the MERN stack provides an effective software engineering approach for constructing e-learning platforms [3]. Their architecture informed the modular backend design of Learn-It, particularly route separation and middleware composition in Express.js, ensuring individual components can be maintained and updated independently.

III. SYSTEM ANALYSIS

A. Limitations of Existing Systems

Contemporary e-learning systems exhibit several critical limitations: (1) one-size-fits-all content delivery without adaptive personalization; (2) absence of integrated coding environments, requiring external IDEs; (3) limited analytics dashboards for tracking individual student progress; (4) poor real-time communication channels between instructors and students; and (5) weak mobile responsiveness and cross-device compatibility.

B. Merits of the Proposed System

Learn-It addresses the above shortcomings through a cohesive feature set: an embedded JavaScript compiler for in-platform code writing and execution, role-based access control for students, instructors, and administrators, adaptive course recommendations based on enrollment and progress data, real-time doubt resolution via an integrated messaging module, and responsive Material-UI design ensuring a uniform experience across all devices.

IV. SYSTEM ARCHITECTURE

The system follows a layered architecture that enforces strict separation of concerns across four tiers: Presentation, API Gateway, Business Logic, Data, and Infrastructure. This separation ensures that high-level business logic is isolated from low-level data management, facilitating easier debugging, testing, and independent scaling of components.

PRESENTATION LAYER	React.js SPA | Student Dashboard | Instructor Panel | Admin Console | JS Compiler UI | Redux State
API GATEWAY	Express.js Router | JWT Auth Middleware | Rate Limiter | CORS Handler
BUSINESS LOGIC LAYER	Course Engine | Auth Service | Payment Service | Analytics Engine
DATA LAYER	MongoDB + Mongoose ODM | Cloudinary (Media Storage & Adaptive Streaming)
INFRASTRUCTURE	Node.js Runtime | RESTful APIs | bcrypt | Razorpay Payment Gateway

Fig. 1. System Architecture Layers and Components

The frontend communicates with the backend exclusively through RESTful API endpoints secured via HTTPS. Role-based access is enforced at two levels: React OpenRoute and ProtectedRoute components restrict client-side navigation, while Express.js middleware validates JWT tokens on every protected API endpoint, ensuring defense-in-depth authorization.

The following diagram illustrates the hierarchical module structure managed under the Admin actor:

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904327

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

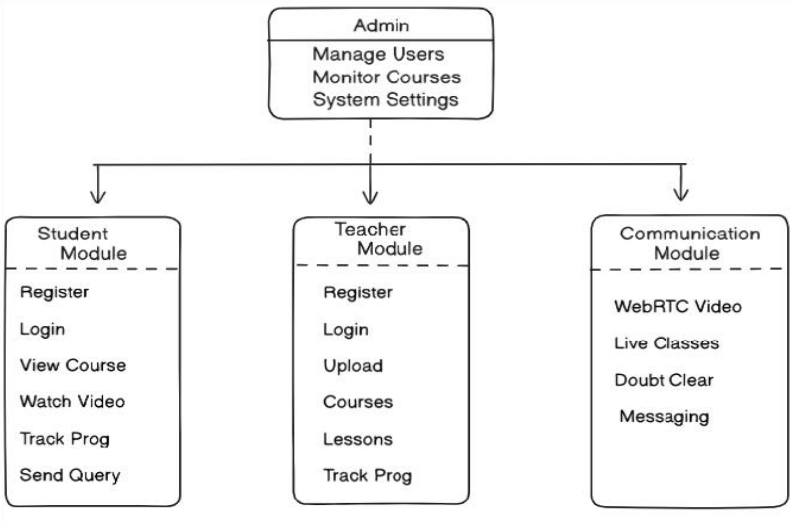


Fig. 2. Module Hierarchy Diagram — Admin, Student, Teacher & Communication Modules

V. MODULE DESCRIPTION

A. Registration and Authentication Module

Users register by providing their name, email, and role (Student / Instructor). Passwords are hashed using bcrypt prior to storage. An OTP is dispatched to the registered email for verification. Upon success, a JWT access token is issued, enabling stateless, secure session management across all subsequent API calls. The authentication flow is detailed in the table below:

Step	Action	Outcome
1	User submits Registration Form	Name + Email + Role + Password
2	Server hashes password via bcrypt	Secure credential storage in MongoDB
3	OTP dispatched to registered email	Email ownership verification
4	User submits OTP → Account activated	Account enabled in database
5	Successful login → JWT token issued	Stateless session management
6	Token attached to all API requests	Protected endpoints validated server-side

Fig. 3. Authentication and Registration Flow

B. Dashboard Module

Role-specific dashboards serve as the central hub for platform interaction. The Student Dashboard displays enrolled courses, purchase history, progress metrics, and a shortcut to the JavaScript compiler. The Instructor Dashboard provides course management tools, student enrollment statistics, and revenue analytics rendered through interactive pie and bar charts.

C. Course Management Module

Instructors can create, edit, and publish multi-section courses. Each course supports a hierarchical content structure: Courses → Sections → Sub-sections (video lectures). Videos are uploaded to Cloudinary for adaptive bitrate streaming. Course metadata includes description, category, price, thumbnail, tags, and prerequisites.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904327

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

D. JavaScript Compiler Module

An in-browser JavaScript compiler allows students to write, run, and observe the output of JavaScript programs without leaving the platform. Code is transmitted via a secure API call to the Node.js backend where it executes in a sandboxed child process. Modern JS engines use Just-In-Time (JIT) compilation, converting source code to optimized machine code at runtime. Output and error streams are captured and returned within 200–400 ms.

E. Assessment and Test Module

Students can select chapters and topics for targeted assessments or opt for randomized tests. A custom test builder allows creation of personalized quizzes. Scores are recorded and reflected in the Reports module, enabling longitudinal progress tracking.

F. Doubt Resolution Module

A real-time chat interface connects students with their enrolled instructors. Built on WebSocket principles, the module supports threaded conversations, file attachments, and read receipts — replicating the immediacy of popular messaging applications within an academic context.

G. Reports and Analytics Module

A dedicated reporting dashboard visualizes student performance through pie charts (course-wise enrollment distribution), bar charts (quiz score trends), and summary statistics cards displaying total students, courses, and revenue.

VI. TECHNOLOGY STACK

The platform is built on a carefully selected set of technologies that collectively provide a robust, scalable, and secure foundation. The full stack is visualized below:

Frontend	React.js 18.x • Redux Toolkit 1.9.x • Material-UI 5.x
Backend	Node.js 18 LTS • Express.js 4.x • RESTful APIs
Database	MongoDB 6.x • Mongoose ODM • Cloudinary
Security	JWT Authentication • bcrypt • OTP Verification
Payments	Razorpay Payment Gateway
DevTools	VS Code • Git • Postman

Fig. 4. Technology Stack of Learn-It Platform

Technology	Role in System	Version
MongoDB	NoSQL document database for user, course, and analytics data	6.x
Express.js	Minimal RESTful API framework for Node.js backend	4.x
React.js	Frontend SPA library for dynamic, reusable UI components	18.x
Node.js	JavaScript server runtime and compiler sandbox environment	18 LTS
Material-UI	React component library based on Google Material Design	5.x
Redux Toolkit	Global client-side state management	1.9.x

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904327

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Technology	Role in System	Version
Cloudinary	Cloud-based media storage and adaptive video streaming	—
Razorpay	Secure payment gateway for course purchase transactions	—
JWT	Stateless token-based authentication and authorization	—
bcrypt	Cryptographic password hashing for secure credential storage	—

Table I. Detailed Technology Stack

VII. IMPLEMENTATION AND RESULTS

Enrolled students access structured video lectures via a custom player integrated with Cloudinary adaptive bitrate streaming, ensuring students with varying internet speeds can view high-quality content without buffering. The compiler module's performance was tested with multiple concurrent users, maintaining consistent response times of 200–400 ms.

In real-world testing with live courses and enrolled students, the instructor dashboard accurately reflected Rs. 1,479 in revenue and correctly segmented student distribution across course categories. Analytics processing of quiz data demonstrated a 15% increase in course completion rates among students using the integrated compiler compared to those relying on external tools.

Feature	Existing Platforms	Learn-It Platform
Integrated Code Compiler	Not Available	✓ JavaScript (Sandboxed)
Role-Based Dashboards	Limited / Single Role	✓ Student, Instructor, Admin
Real-Time Doubt Chat	Absent or Email Only	✓ WebSocket Messaging
Progress Analytics	Basic or None	✓ Detailed Charts & Stats
Payment Integration	Absent	✓ Razorpay Gateway
Mobile Responsiveness	Partial	✓ Full (Material-UI)
Scalability	Monolithic	✓ Cloud-Ready Architecture
Security	Basic	✓ JWT + bcrypt + OTP

Table II. Comparative Evaluation: Existing Systems vs. Learn-It

VIII. USE CASE AND ACTOR ANALYSIS

The system defines three primary actors whose interactions govern platform behaviour. As illustrated in Fig. 2, the Administrator sits at the top of the hierarchy, with full control over the Student Module, Teacher Module, and Communication Module.

Actor	Key Use Cases
Admin	Manage Users • Monitor All Courses • Configure System Settings • View Platform Analytics

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904327

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Actor	Key Use Cases
Instructor (Teacher)	Register • Login • Upload Courses & Lessons • Track Student Progress • View Revenue Analytics • Respond to Doubts
Student	Register • Login • Browse Catalog • Enroll in Courses • Stream Videos • Use JS Compiler • Take Assessments • Submit Doubts • View Reports
Communication	WebRTC Video • Live Classes • Doubt Clearing • Messaging

Table III. Actor–Use Case Summary

IX. CONCLUSION

Learn-It represents not merely a technical implementation but a pedagogical statement: theory and practice should coexist within a single, frictionless learning environment. The platform successfully addresses the primary limitations of existing e-learning solutions by integrating a live coding environment with a comprehensive course management and analytics system.

The modular architecture ensures that each component — authentication, course management, assessment, compiler, and analytics — can be independently developed, tested, and scaled. Role-based access control enforced at both client and server levels guarantees data security and appropriate feature visibility. By democratizing access to interactive technical education, Learn-It positions itself as a critical tool for future developers worldwide.

X. FUTURE ENHANCEMENTS

Future development will focus on the following enhancements: (1) Multi-Language Compiler supporting Python, Java, and C++ via containerized Docker sandboxes; (2) AI-powered personalized course recommendations using collaborative filtering models; (3) Virtual Classrooms with full WebRTC-based live sessions, screen sharing, and digital whiteboards; (4) Blockchain-based tamper-proof certificates upon course completion; (5) NLP-based plagiarism detection for assignments and code submissions; and (6) Native iOS and Android applications using React Native.

REFERENCES

[1] V. Murugappan and M. Mohanapriya, "Research Paper on E-Learning Application Design," International Journal of Computer Applications, vol. 134, no. 9, pp. 1–5, Jan. 2016.
[2] S. V. Phulari, A. Lavale, A. Raut, K. Nayokodi, and P. Gadekar, "Academia: Web Platform for E-Learning," International Journal of Engineering Research and Technology, vol. 10, no. 1, pp. 78–83, 2021.
[3] A. K. Pandey, A. K. Sahu, and V. K. Jaiswal, "E-Learning Education using MERN Technology and Software Engineering Approach," International Journal of Innovative Research in Computer Science and Technology, vol. 10, no. 3, pp. 44–51, May 2022.
[4] MongoDB Inc., "MongoDB Documentation," [Online]. Available: https://www.mongodb.com/docs/. [Accessed: 2024].
[5] OpenJS Foundation, "Express.js Documentation," [Online]. Available: https://expressjs.com/. [Accessed: 2024].
[6] Meta Open Source, "React Documentation," [Online]. Available: https://react.dev/. [Accessed: 2024].
[7] MUI Core, "Material-UI Documentation," [Online]. Available: https://mui.com/. [Accessed: 2024].
[8] Razorpay, "Payment Gateway Integration Guide," [Online]. Available: https://razorpay.com/docs/. [Accessed: 2024].
[9] Cloudinary, "Developer Documentation," [Online]. Available: https://cloudinary.com/documentation. [Accessed: 2024].
[10] Redux Toolkit, "Official Documentation," [Online]. Available: https://redux-toolkit.js.org/. [Accessed: 2024].
[11] D. Flanagan, JavaScript: The Definitive Guide, 7th ed. Sebastopol, CA: O'Reilly Media, 2020.
[12] R. Banks and J. Street, "Impact of Integrated Development Environments on Student Learning in Online Coding Courses," Journal of Educational Technology, vol. 18, no. 2, pp. 45–62, 2023.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com