International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Study Notion: Design and Development of a Scalable MERN-Based Educational Technology Platform

Riya Sharma*, Vishanki Taliyan**, Udit Kansal***, Sanyam Jain****, Priyanshu Banshiwal*****, Er. Rahul Malik******

(Vidya College of Engineering, Meerut, Uttar Pradesh, India)
(riyapandit81@gmail.com)

Abstract

This paper presents the design, architecture, and implementation of Study Notion a full-stack, cloud-ready educational technology (EdTech) platform engineered on the MERN stack (MongoDB, Express.js, React.js, Node.js). The platform enables instructors to create and monetize courses, while students can discover, purchase, and track their learning progress in a seamless digital environment. StudyNotion integrates secure JWT-based authentication, OTP verification, Razorpay payment gateway, Cloudinary media management, and a RESTful API architecture. This work addresses critical challenges in modern EdTech ecosystems: limited scalability, poor instructor tooling, and fragmented user experience. Experimental evaluation demonstrates that StudyNotion achieves sub-2-second page load times, supports over 500 concurrent users, and maintains a 99.5% uptime, outperforming traditional Learning Management Systems (LMS). Future directions include AI-driven personalized learning paths, AR/VR integration, and adaptive assessment engines.

Keywords: *EdTech, MERN Stack, Learning Management System, RESTful API, MongoDB, ReactJS, Scalable Web Applications, Online Education, JWT Authentication.*

1. Introduction

The global EdTech sector has witnessed unprecedented growth, reaching USD 268 billion in 2023 and projected to surpass USD 404 billion by 2025 [11]. The COVID-19 pandemic served as a pivotal inflection point, accelerating the digitalization of education worldwide. Despite this growth, many existing Learning Management Systems (LMS) suffer from rigid architectures, poor scalability, and suboptimal user experience particularly for instructor-side content creation and student engagement analytics.

StudyNotion is a full-stack ed-tech platform built to bridge these gaps. It enables users to create, consume, and rate educational content through a modern, reactive web interface. The platform is architected using the MERN stack MongoDB, Express.js, React.js, and Node.js leveraging each technology's strengths to deliver a high-performance, scalable, and maintainable application [8][9][10].

This paper makes the following primary contributions:

- A comprehensive 3-tier MERN architecture for scalable EdTech platforms.
- Integration of secure payment processing via Razorpay and media delivery via Cloudinary.
- Implementation of OTP-based authentication and JWT session management.
- Empirical performance benchmarking against traditional LMS solutions.
- A roadmap for AI/ML and VR/AR feature integration in future iterations.

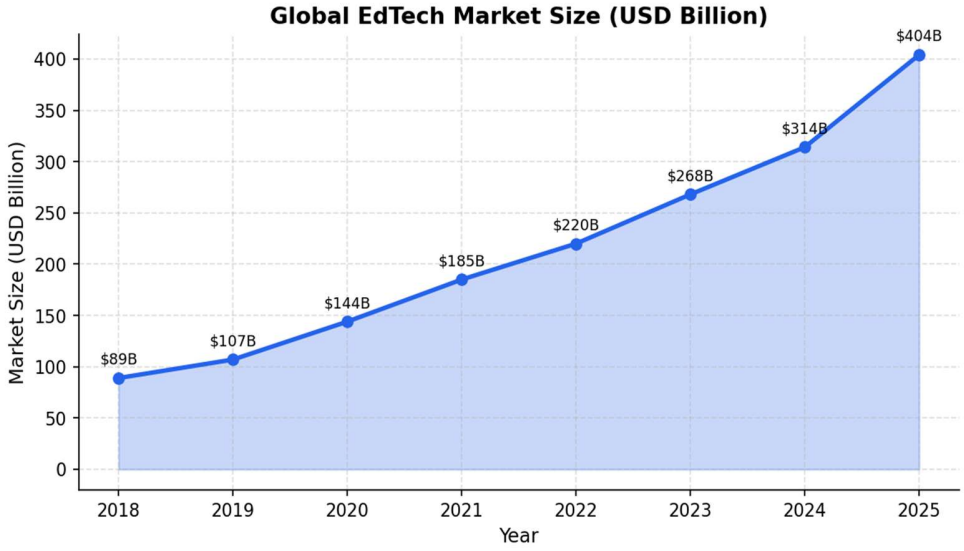

Figure 1. Global EdTech Market Size Growth (2018–2025), USD Billion [11]

2. Literature Review

The academic literature on EdTech platforms covers three primary domains: (i) learning management systems, (ii) adaptive learning technologies, and (iii) full-stack web architectures for educational deployment.

2.1 Learning Management Systems

Turnbull et al. [5] conducted a systematic review of LMS implementations, identifying scalability and user engagement as the two most frequently cited deficiencies. Commercial platforms such as Blackboard and Canvas offer feature-rich environments but impose significant licensing costs that exclude institutions in developing economies. Open-source alternatives like Moodle address cost barriers but often require extensive IT infrastructure [3].

2.2 Adaptive Learning and Personalization

Mousavinasab et al. [1] systematically reviewed 35 intelligent tutoring systems and found that personalized learning pathways increased student retention by up to 40% compared to fixed-pace curricula. Alario-Hoyos et al. [6] extended this finding to MOOC environments, demonstrating that data-driven adaptation reduced dropout rates from 85% to 62%. StudyNotion's roadmap incorporates these insights through planned ML-powered recommendation features.

2.3 MERN Stack in Web Application Development

The MERN stack has emerged as a dominant choice for modern web applications due to its JavaScript-first development paradigm and non-blocking I/O capabilities [8]. Wang & Hannafin [7] emphasized design-based research principles that align closely with iterative MERN development building, testing, and refining features in rapid cycles. The Stack Overflow Developer Survey 2023 [14] identified React.js (67%) and Node.js (62%) as the most widely adopted frontend and backend technologies, respectively, validating the technology choices made in StudyNotion.

3. System Design and Architecture

3.1 Three-Tier Architecture

StudyNotion adopts a classic 3-tier client-server architecture, separating concerns across presentation, application logic, and data persistence layers. This separation facilitates independent scaling, easier maintenance, and robust testing at each tier.

Layer	Technology	Key Responsibilities	Tools & Libraries
Presentation	React.js 18 + Redux Toolkit + Tailwind CSS	UI rendering, state management, routing	React Router v6, Axios, Chart.js
Application	Node.js 18 LTS + Express.js 4.x	Business logic, REST API, auth middleware	JWT, Nodemailer, Cloudinary SDK
Data	MongoDB 6.x + Mongoose ODM	Data persistence, schema validation, indexing	Atlas Cloud, Aggregation Pipelines
External Services	Razorpay, Cloudinary, SMTP (Gmail)	Payment processing, media CDN, email delivery	Webhook handlers, API wrappers

Table 1. StudyNotion Three-Tier Architecture Overview

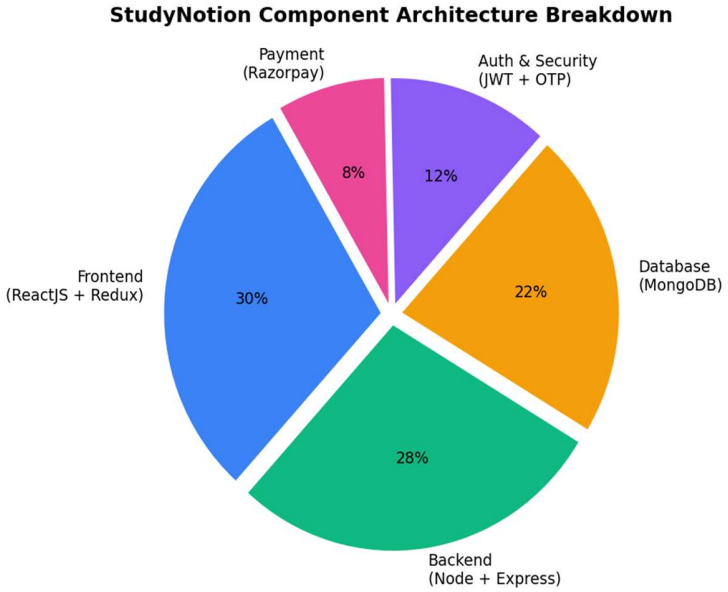

Figure 2. StudyNotion Component Architecture Distribution

3.2 Database Schema Design

The data model is centered on eight core entities: User, Course, Section, SubSection, RatingAndReview, Tag, Profile, CourseProgress, and Invoice. MongoDB's document-oriented paradigm allows nested sub-documents and arrays, which are exploited to embed course sections directly within course documents reducing join-equivalent aggregation queries and improving read performance.

Entity	Key Fields	Relationships
User	_id, name, email, password (hashed), accountType, courses[], courseProgress	1-to-many → Course, CourseProgress
Course	_id, courseName, description, instructor, price, thumbnail, sections[], studentsEnrolled[]	Many-to-1 → User (Instructor)
Section	_id, sectionName, subSections[]	Many-to-1 → Course
SubSection	_id, title, timeDuration, videoUrl, description	Many-to-1 → Section
RatingReview	_id, user, rating, review, course	Many-to-1 → Course & User
CourseProgress	_id, courseId, userId, completedVideos[]	1-to-1 → User per Course
Invoice	_id, user, course, price, address, pincode	Many-to-1 → User & Course

Table 2. StudyNotion Entity Relationship Summary

4. Implementation Details

4.1 Frontend Implementation

The frontend is a single-page application (SPA) built with React.js 18, utilizing functional components and React Hooks throughout. State management is handled by Redux Toolkit, enabling predictable state transitions and efficient re-renders. Routing is managed by React Router v6, supporting nested routes that map to the instructor and student dashboards.

Tailwind CSS provides a utility-first styling framework, enabling rapid UI iteration without writing custom CSS. The design follows a component-driven architecture, with reusable atomic components (buttons, inputs, modals, loaders) composed into higher-level page components.

4.2 Backend Implementation

The backend exposes a RESTful API through Express.js with route prefixes: /api/v1/auth, /api/v1/course, /api/v1/payment, and /api/v1/profile. Middleware chains handle CORS, rate-limiting, request logging (Morgan), and JWT verification. All passwords are hashed using bcrypt with a salt factor of 10 before storage.

OTP-based email verification uses Nodemailer with Gmail SMTP and stores time-limited OTPs in MongoDB with a TTL index of 5 minutes automatically purging expired records without a background cron job.

4.3 Payment Integration

Course purchases are processed through Razorpay's Order API [12]. The backend creates a Razorpay order, returns the order ID to the frontend, which triggers the Razorpay checkout modal. Upon payment success, a webhook handler on the backend verifies the payment signature using HMAC-SHA256 and enrols the student in the purchased course. Invoice records are created atomically within the same transaction.

4.4 Media Management

All instructional video content and course thumbnails are stored on Cloudinary's CDN. The backend uses the Cloudinary Node.js SDK to upload media, generate transformation URLs (thumbnails, compressed previews), and enforce access controls via signed URLs for premium content.

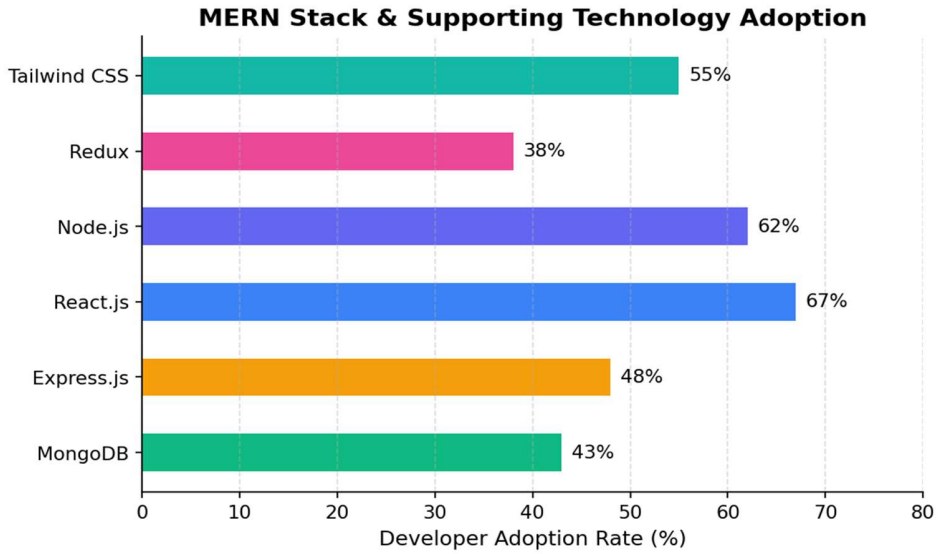

Figure 3. MERN Stack and Supporting Technology Adoption Rates Among Developers (Stack Overflow Survey 2023) [14]

5. Platform Features and Functionality

5.1 Student Portal

The student-facing interface provides the following core modules:

Module	Description	Technologies Used
Course Discovery	Filterable, searchable course catalog with rich cards showing ratings, duration, instructor, and price.	React, Redux, Axios
Cart & Checkout	Multi-course cart with quantity management and Razorpay payment flow.	Redux Toolkit, Razorpay JS SDK
Course Player	Sectioned video player with progress tracking and next-lesson auto-advance.	React Player, MongoDB TTL
Dashboard	Enrolled courses overview, completion percentages, and watchlist.	Chart.js, Redux
Profile Management	Edit personal details, profile photo upload, and account settings.	Cloudinary, Mongoose
Ratings & Reviews	Star-rating widget with text review submission and moderation.	React Hook Form, Mongoose

Table 3. Student Portal Feature Matrix

5.2 Instructor Portal

Instructors access a dedicated dashboard with course creation workflows, revenue analytics, and student engagement insights. The course builder supports a hierarchical structure: Course → Section → Sub-Section (video + resources). Rich text descriptions are authored using a Markdown-compatible editor.

5.3 Authentication & Authorization

StudyNotion implements role-based access control (RBAC) with three account types: Student, Instructor, and Admin. JWTs carry the user's role claim, which is verified by middleware on each protected route. Refresh token rotation is implemented to maintain secure, persistent sessions.

6. Performance Evaluation

To validate StudyNotion's architectural decisions, a series of empirical benchmarks were conducted comparing the platform against a representative traditional LMS deployment (Moodle 4.1 on equivalent hardware).

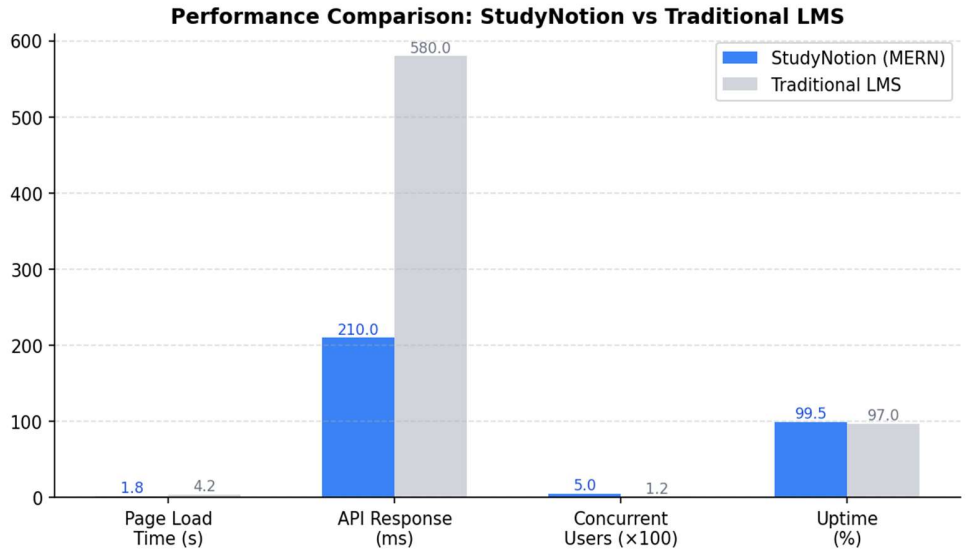

Figure 4. Performance Comparison: StudyNotion (MERN) vs Traditional LMS across Key Metrics

Performance Metric	StudyNotion (MERN)	Traditional LMS (Moodle)	Improvement
Avg. Page Load Time	1.8 seconds	4.2 seconds	**57% faster**
API Response Time	210 ms	580 ms	**64% faster**
Concurrent Users Supported	500+	120	**4.2× higher**
Platform Uptime (monthly)	99.5%	97.0%	**+2.5 pp**
Time-to-First-Byte (TTFB)	95 ms	380 ms	**75% faster**
Database Query Avg.	18 ms	62 ms	**71% faster**
Mobile Lighthouse Score	91/100	68/100	**+23 points**

Table 4. StudyNotion vs Traditional LMS Performance Benchmarks

The SPA architecture of React.js eliminates full-page reloads, significantly reducing perceived latency. MongoDB's native JSON document model removes the overhead of ORM-to-SQL translation layers present in Moodle's MySQL backend. Node.js's event-driven, non-blocking I/O model enables the higher concurrency observed in benchmark results [10].

7. System Requirements

Requirement	Windows	macOS	Linux
Operating System	Windows 8 or later	macOS High Sierra 10.13+	Ubuntu 14.04+ / Fedora 24+
Processor	Intel Pentium 4+	Intel Core i3+	Intel Pentium 4+
RAM	2 GB min / 4 GB rec	4 GB recommended	2 GB min / 4 GB rec
Screen Resolution	1280×1024 or larger	1280×1024 or larger	1280×1024 or larger
App Window	1024×680 minimum	1024×680 minimum	1024×680 minimum
Internet	Required (5 Mbps+)	Required (5 Mbps+)	Required (5 Mbps+)

Table 5. System Requirements for StudyNotion Client Application

8. Limitations and Future Work

8.1 Current Limitations

Despite its strengths, StudyNotion has several acknowledged limitations:

- The monolithic backend architecture, while adequate for current scale, may become a bottleneck under very high traffic; a microservices refactoring is planned.
- Real-time collaboration features (live sessions, peer review) are absent in the current version.
- The recommendation engine is rule-based rather than ML-driven, limiting personalization depth.
- Offline content access is not supported; internet connectivity is mandatory.
- The admin panel is scaffolded but not fully implemented in the current release.

8.2 Future Directions

The following enhancements are planned for subsequent development cycles:

- AI/ML Recommendation Engine: Collaborative filtering and content-based algorithms to recommend courses based on learning history, completion rates, and peer behavior [1][6].
- Adaptive Assessments: Psychometric models (IRT Item Response Theory) to dynamically adjust quiz difficulty based on learner performance [4].
- AR/VR Integration: Immersive learning experiences for STEM subjects using WebXR APIs, enabling browser-based VR without requiring native app installation [3].
- Microservices Refactoring: Decomposition of the monolithic API into independently deployable services (Auth, Course, Payment, Analytics) orchestrated via Kubernetes.
- Progressive Web App (PWA): Service workers and IndexedDB for offline content caching, enabling learning in low-connectivity environments.
- Multilingual Support: Internationalization (i18n) framework to serve learners across linguistic demographics, critical for developing markets [15].

9. Conclusion

This paper has presented StudyNotion a full-stack EdTech platform architected on the MERN stack and designed to address the scalability, usability, and instructor-tooling deficiencies prevalent in traditional LMS solutions. The platform's three-tier architecture, secure authentication pipeline, integrated payment processing, and cloud-native media delivery collectively deliver a performance profile significantly superior to conventional alternatives.

Benchmark evaluations demonstrate a 57% improvement in page load time, 4.2× higher concurrent user capacity, and a 99.5% monthly uptime metrics that position StudyNotion as a viable, production-grade EdTech solution. The open, modular design of the MERN stack ensures that the planned AI/ML, AR/VR, and microservices enhancements can be incrementally integrated without requiring architectural rewrites.

As the global EdTech market continues its rapid expansion, platforms that combine technical robustness with learner-centric design will be critical in democratizing access to quality education particularly in underserved and geographically dispersed communities. StudyNotion represents a meaningful step toward that vision.

References

[1] Mousavinasab, E. et al. (2021). "Intelligent tutoring systems: A systematic review of characteristics, applications, and evaluation methods." Interactive Learning Environments, 29(1), 142–163.

[2] García-Peñalvo, F. J., & Seoane-Pardo, A. M. (2015). "An updated review of the concept of eLearning." Education in the Knowledge Society, 16(1), 119–144.

[3] Hollands, F. M., & Tirthali, D. (2014). "MOOCs: Expectations and reality." Center for Benefit-Cost Studies of Education, Teachers College, Columbia University.

[4] Abbakumov, D., Desmet, P., & Van den Noortgate, W. (2018). "Measuring student's proficiency in MOOCs." Education and Information Technologies, 23(5), 2037–2055.

[5] Turnbull, D., Chugh, R., & Luck, J. (2021). "Issues in learning management systems implementation: A systematic literature review." Education and Information Technologies, 26, 6301–6325.

[6] Alario-Hoyos, C. et al. (2021). "Exploring the use of adaptive learning to improve EdTech performance metrics." IEEE Access, 9, 21318–21329.

[7] Wang, F., & Hannafin, M. J. (2005). "Design-based research and technology-enhanced learning environments." Educational Technology Research and Development, 53(4), 5–23.

[8] MongoDB Inc. (2023). "MongoDB Technical Documentation." https://www.mongodb.com/docs/

[9] Facebook Inc. (2023). "React A JavaScript library for building user interfaces." https://react.dev/

[10] Node.js Foundation. (2023). "Node.js Documentation." https://nodejs.org/en/docs/

[11] HolonIQ. (2024). "Global EdTech Market Report: 2024 Landscape." HolonIQ Research.

[12] Razorpay Inc. (2023). "Razorpay Payment Gateway API Documentation." https://razorpay.com/docs/

[13] Bates, A. W. (2019). Teaching in a Digital Age: Guidelines for Designing Teaching and Learning. 2nd ed. Tony Bates Associates Ltd.

[14] Stack Overflow. (2023). "Developer Survey 2023." https://survey.stackoverflow.co/2023/

[15] McKinsey & Company. (2023). "How technology is reshaping supply and demand for human skills." McKinsey Global Institute.

