IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, May 2026
Impact Factor: 8.2

Study Material Management System: A Web-Based Solution for Centralized Academic Resource Organization

Mr. Gaurav Ravindra Chaudhari and Asst. Prof. Rammohan Ashok Agrawal
M.C.A. Second Year Student, Department of Master of Computer Application
Assistant Professor, Department of Master of Computer Application
Hindi Seva Mandal's Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal, Maharashtra, India
gauravchaudhari@example.com and agrawalrammohan@gmail.com

Abstract: *The contemporary academic ecosystem has increasingly demanded a reliable, centralized digital platform for the systematic management of study materials, reference resources, and examination preparation content. Students pursuing higher technical education such as the Master of Computer Application (MCA) program frequently encounter the challenge of managing a diverse array of lecture notes, textbooks, research papers, practice sets, and video resources spread across multiple disconnected storage mediums, resulting in inefficient retrieval, duplication of effort, and suboptimal examination preparation strategies. In response to this pervasive academic challenge, the Study Material Management System (SMMS) has been developed as a comprehensive, full-stack web application designed to centralize, organize, and intelligently manage all forms of study content within a single authenticated digital environment. The platform is developed using React.js for the responsive frontend interface and Node.js with Express for the backend RESTful API layer, with MongoDB providing flexible, document-oriented data persistence.*

SMMS enables students to upload, categorize, tag, and track progress across all their study materials, while an integrated AI-powered study assistant driven by Google Gemini provides contextually relevant guidance, generates subject-wise practice questions, and assists in constructing personalized weekly study schedules. Role-based access control allows guides and faculty to monitor student progress, suggest resources, and provide feedback through a dedicated dashboard. Automated reminders and progress analytics ensure students remain on track with their preparation goals, while a self-testing module enables continuous formative assessment. The system fundamentally transforms fragmented, manual academic resource management into an intelligent, data-driven digital workflow, empowering students to achieve better academic outcomes through structured, technology-assisted learning.

Keywords: Study Material Management, Academic Resource System, React.js, Node.js, MongoDB, Google Gemini AI, JWT Authentication, Role-Based Access Control, Progress Tracking, Self-Testing Module, MCA Education Technology

I. INTRODUCTION

The rapid expansion of academic curricula in technical education programs, particularly at the postgraduate level, has introduced a significant challenge in the effective management and organization of study materials. Students enrolled in programs such as the Master of Computer Application are routinely required to engage with an extensive and heterogeneous collection of learning resources that spans textbooks, faculty-prepared notes, research papers, video tutorials, and practice examination sets. The absence of a unified platform for organizing these resources forces students to rely on fragmented storage solutions such as local file systems, cloud drives, and disparate web bookmarks,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 11, May 2026 Impact Factor: 8.2

creating a disjointed learning environment that imposes unnecessary cognitive overhead and reduces overall study efficiency.

This fragmentation is further compounded by the absence of structured progress tracking mechanisms, which leaves students without a reliable means of identifying knowledge gaps, monitoring coverage of examination syllabi, or benchmarking their preparation against defined academic milestones. Without an integrated platform that consolidates resource management, progress monitoring, and intelligent study guidance, students are routinely compelled to invest disproportionate time in logistical preparation tasks rather than substantive academic engagement. These inefficiencies collectively diminish the quality of examination preparation and contribute to suboptimal academic performance outcomes across institutions.

To address these persistent challenges, the Study Material Management System has been designed as an intelligent, role-aware, full-stack web application. The system enables students to upload and categorize study content by subject, resource type, and examination relevance, while tracking completion progress at a granular level. By integrating an AI-powered study assistant, automated scheduling tools, and a self-testing module, the platform transforms the passive activity of resource storage into an active, guided, and measurable learning process tailored to the individual academic needs of each student.

II. OVERVIEW

The evolution of academic resource management has progressed through several distinct technological phases. Early approaches relied entirely on physical organization methods such as printed notes, binders, and library catalogues, which offered limited searchability and were inherently susceptible to loss and duplication. The subsequent adoption of personal computing introduced basic digital file management through folder hierarchies and desktop applications, providing improved storage capacity but lacking collaborative features, intelligent organization, or real-time progress tracking. The proliferation of cloud storage platforms such as Google Drive and Dropbox represented a meaningful improvement in accessibility and synchronization but failed to address the fundamental need for subject-aware categorization, examination-aligned tagging, and integrated study guidance.

Contemporary implementations in educational technology have begun leveraging Artificial Intelligence and large language models to enhance the personalization of academic support. AI-powered tutoring systems provide adaptive learning pathways based on student performance data, while intelligent recommendation engines surface contextually relevant resources based on query semantics and learning history. Despite these substantial technological advances, a significant gap remains in developing systems that effectively combine centralized resource management with AI-assisted study planning, progress analytics, and formative self- assessment within a single cohesive platform accessible to students in resource-constrained academic environments.

The Study Material Management System directly addresses these limitations by delivering a comprehensive, integrated web platform that consolidates resource upload and categorization, subject-wise progress tracking, AI-powered study assistance, automated weekly scheduling, and self-testing into a unified digital environment. By maintaining all academic resources, progress records, and study analytics within a single authenticated application, the system eliminates the fragmentation and cognitive overhead associated with traditional multi-platform study workflows.

III. ARCHITECTURE

The Study Material Management System is built upon a modern three-tier web application architecture that enforces a clean separation between the presentation layer, the business logic layer, and the data persistence layer. At the data tier, MongoDB provides a flexible, document-oriented storage engine capable of efficiently representing the heterogeneous nature of academic resources, which vary significantly in structure depending on whether they represent uploaded files, embedded video links, self- generated notes, or externally referenced research papers. Mongoose ODM manages all database interactions, enforcing schema validation and relational constraints between user profiles, material records, progress entries, quiz responses, and scheduli ng documents.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, May 2026
Impact Factor: 8.2

The business logic tier is implemented using Node.js with the Express framework, exposing a comprehensive RESTful API that processes all client requests for material management, progress updates, AI query routing, and schedule generation. JWT-based stateless authentication protects every API endpoint, with BCrypt providing password hashing at a cost factor of ten rounds to ensure credential security. The AI query processing subsystem routes student queries to the Google Gemini API, with a deterministic rule- based fallback system ensuring uninterrupted service availability in the event of external API unavailability.

The presentation tier is constructed using React.js, providing a responsive single-page application that communicates with the Node.js backend exclusively via HTTP requests carrying JWT bearer tokens managed by an Axios interceptor. The frontend is organized into clearly defined feature modules: the Dashboard module providing a centralized overview of materials, progress statistics, and upcoming study schedule items; the Materials module enabling upload, categorization, tagging, and progress tracking for all academic resources; the Self-Test module delivering randomized subject-wise practice questions with instant feedback; the Schedule module presenting an AI-generated weekly study plan; and the AI Assistant module providing real-time study guidance and question-generation capabilities powered by Google Gemini.

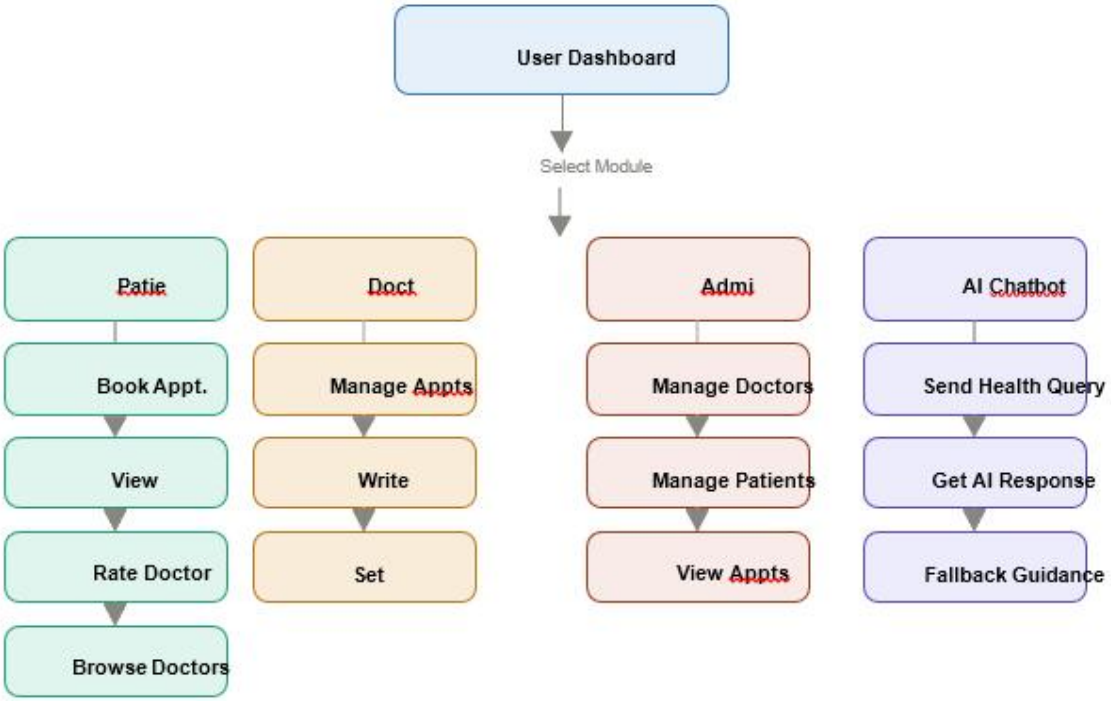

Fig. 1. User Dashboard – Module Selection Flow

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, May 2026
Impact Factor: 8.2

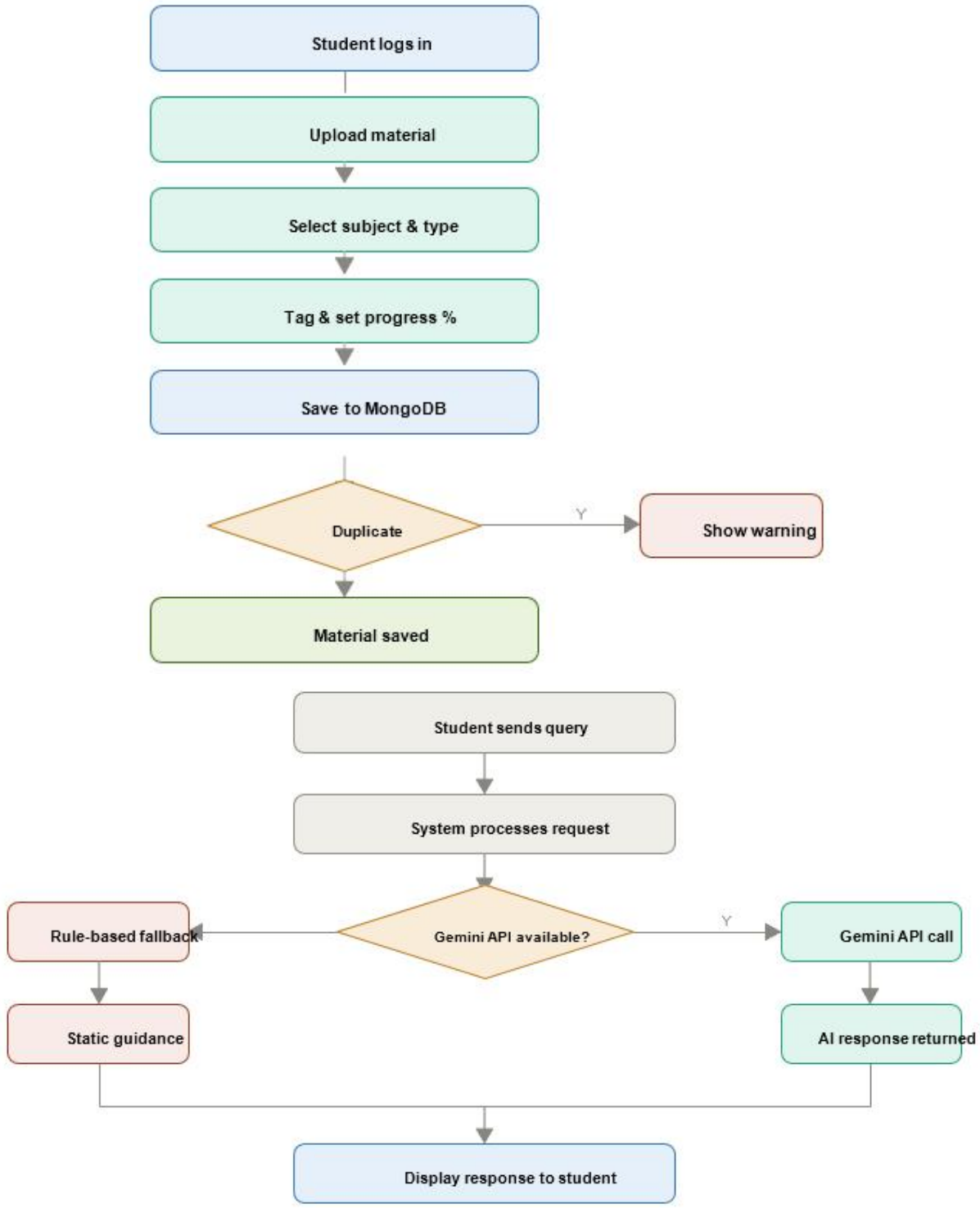

Fig. 3. AI Study Assistant & Health Guidance Module Flow.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 11, May 2026
Impact Factor: 8.2

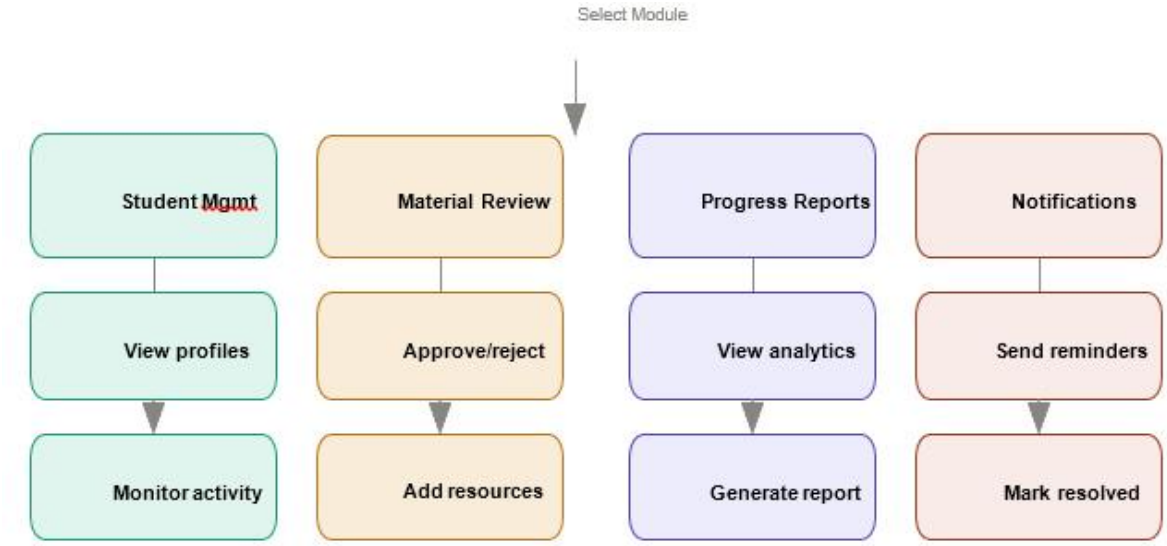

Fig. 5. Admin / Guide Dashboard – Platform Management Flow

IV. METHODOLOGY

1 . Methodology :

The technical architecture of the Study Material Management System is engineered to ensure reliable performance on standard consumer-grade hardware typically available to postgraduate students. The minimum hardware configuration for running the application locally requires a dual-core processor at 1.8 GHz or higher, with quad-core configurations strongly recommended to support the concurrent execution of the React development server, the Node.js backend service, and the MongoDB database daemon. A minimum of 4 GB of RAM is required, with 8 GB recommended to support comfortable multitasking alongside a browser and code editor. Storage requirements are modest at 20 GB of free disk space, sufficient to accommodate the application source code, Node.js modules, MongoDB data files, and user-uploaded study materials maintained in a local file store.

The software technology stack has been selected for its accessibility, open-source availability, and extensive community support. The backend is built on Node.js 18.x with the Express 4.x framework, exposing a REST API on port 5000. Authentication is implemented using JSON Web Tokens with HMAC-SHA256 signing, providing stateless, database-free session management across all protected endpoints. Passwords are hashed using BCrypt at a cost factor of ten rounds before storage in MongoDB. The frontend is built using React.js 18.x with the Vite build tool, served on port 3000 during development. All API communication is handled by Axios with request interceptors that automatically attach the JWT bearer token to outgoing requests. Cross-Origin Resource Sharing is configured on the Express backend to permit requests exclusively from the React development server, ensuring secure full-stack integration throughout the development and testing lifecycle.

Implementation:

The core security architecture of the platform is grounded in a multi-layered credential management framework. User registration triggers BCrypt hashing of the submitted password at a cost factor of ten rounds, producing a 60-character irreversible hash stored in MongoDB. On each subsequent login, the submitted credential is verified through a constant-time comparison against the stored hash. Upon successful authentication, the system generates a signed JWT containing the user identifier, assigned role, and a 24- hour expiration timestamp, signed with a 256-bit HMAC-SHA256 secret key. The React frontend stores this token in memory and attaches it to all outgoing API requests via

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 11, May 2026 Impact Factor: 8.2

Axios interceptor. Every Express route protected by the authentication middlewar e re- validates the incoming token by re-signing the header and payload with the same secret and comparing the resulting signature, enabling stateless, performance-efficient access control at every endpoint.

The AI study assistant module integrates the Google Gemini API to provide contextually appropriate academic guidance tailored to each student's query. When a student submits a query through the assistant interface, the backend constructs a structured prompt embedding the student's active subjects, recently accessed materials, and the specific query text before dispatching it to the Gemini API. The API response is parsed and returned to the frontend for display. A deterministic rule-based fallback system intercepts common query patterns and returns static guidance content in the event of API unavailability, ensuring uninterrupted service. The quiz generation sub-module similarly leverages Gemini to dynamically construct subject-specific multiple-choice questions at configurable difficulty levels, enabling continuous formative self-assessment aligned with examination requirements.

V. CONCLUSION

The development of the Study Material Management System represents a meaningful advancement in the application of modern web technologies to the practical challenges of academic resource organization and examination preparation management. By consolidating the functions of material upload and categorization, subject-wise progress tracking, AI-powered study assistance, automated schedule generation, and formative self-testing into a single cohesive platform, the system addresses the fragmentation that has historically constrained the effectiveness of student-directed study workflows. Built using React.js for the frontend and Node.js with MongoDB for the backend, the application demonstrates how contemporary full-stack engineering practices can be applied to produce genuinely valuable academic support tools that are both technically robust and practically accessible to students in standard hardware environments.

At its technical core, the system delivers stateless JWT-based authentication, role-aware access control, and an AI-powered study assistant that transitions the platform beyond simple resource storage into an intelligent, context-aware learning environment. The integration of Google Gemini enables dynamic question generation and personalized study guidance that adapts to each student's current material coverage and self-assessment performance, ensuring that the platform delivers progressively greater value as a student's engagement with the system deepens over time. The self-testing module, combined with automated weekly scheduling, transforms the preparation experience from a passive resource retrieval exercise into a structured, measurable, and continuously improving academic workflow.

The ultimate value of the Study Material Management System extends beyond technological convenience. It fundamentally repositions academic preparation as a data-driven, intelligently guided process rather than a disorganized accumulation of disconnected resources. By automating the logistical overhead of material organization and study scheduling, the system enables students to redirect their cognitive resources from administrative preparation tasks toward substantive academic engagement, ultimately contributing to improved learning outcomes and more confident, well-prepared examination performance. As digital literacy and technology adoption in higher education continue to grow, the Study Material Management System provides a scalable, extensible foundation for the future of intelligent, student-centered academic support infrastructure.

VI. FUTURE SCOPE

The developmental roadmap for the Study Material Management System encompasses several strategically significant enhancements designed to deepen the platform's intelligence, extend its accessibility, and broaden its institutional utility. The most immediate priority is the integration of a personalized adaptive learning engine that analyzes each student's self-test performance history to dynamically adjust the difficulty and subject weighting of generated practice questions, ensuring that the platform's guidance remains optimally calibrated to each student's evolving knowledge profile. Complementing this, a collaborative study group feature is planned that will enable students to share curated

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 11, May 2026 Impact Factor: 8.2

material collections, annotate shared resources, and coordinate group study sessions through an integrated scheduling interface.

On the infrastructure side, the application is planned for containerization using Docker and deployment to a cloud hosting environment such as AWS or Azure, transitioning from the current local development configuration to a publicly accessible, highly available web service. The MongoDB instance will be migrated to a managed cloud database service to ensure automated backups, horizontal scalability, and geographic redundancy. A native mobile application for Android and iOS platforms is planned using React Native, enabling students to access their study materials, review AI-generated practice questions, and track preparation progress on mobile devices. Additionally, integration with institutional Learning Management Systems such as Moodle through LTI standards is envisioned to enable seamless exchange of course materials and assessment data between the SMMS platform and existing institutional academic infrastructure.

REFERENCES

[1] Pressman, R. S., & Maxim, B. R. (2019). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.
[2] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine (PhD Dissertation).
[3] Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT). RFC 7519. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc7519
[4] Provos, N., & Mazières, D. (1999). A Future-Adaptable Password Scheme. USENIX Annual Technical Conference, FREENIX Track.
[5] Google LLC. (2024). Gemini API Documentation: Generative AI for Developers. https://ai.google.dev/docs
[6] OpenJS Foundation. (2024). Node.js Documentation. https://nodejs.org/en/docs
[7] Meta Platforms Inc. (2024). React Documentation. https://react.dev
[8] MongoDB Inc. (2024). MongoDB Manual. https://www.mongodb.com/docs/manual/
[9] Mongoose Team. (2024). Mongoose ODM Documentation. https://mongoosejs.com/docs/
[10] Auth0 Inc. (2024). JSON Web Tokens: Introduction. Okta Inc. https://jwt.io/introduction
[11] OWASP Foundation. (2023). OWASP Top Ten: Web Application Security Risks. https://owasp.org/www-project-top-ten/ [12] Sommerville, I. (2015). Software Engineering (10th ed.). Pearson Education Limited.
[13] Fowler, M. (2018). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley Professional.
[14] Newman, S. (2021). Building Microservices: Designing Fine-Grained Systems (2nd ed.). O'Reilly Media.
[15] Microsoft Corporation. (2024). TypeScript Documentation. https://www.typescriptlang.org/docs/
[16] Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. IETF. https://datatracker.ietf.org/doc/html/rfc8446
[17] World Health Organization / UNESCO. (2021). Digital Education Global Report. UNESCO Press.
[18] Katz, J., & Lindell, Y. (2020). Introduction to Modern Cryptography (3rd ed.). CRC Press.
[19] Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley Professional.
[20] National Institute of Standards and Technology. (2022). Digital Identity Guidelines: Authentication and Lifecycle Management. NIST SP 800-63 B.