www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2159-2165

e-ISSN : 2583-1062
Impact Factor : 7.001

INTELLIGENT ORGANIZATIONAL MANAGEMENT SYSTEM

Simmi Chaudhary1, Raj Aditya[2], Rahul Sharma[3]

[1,2,3]Department of Artificial Intelligence and Data Science, Poornima Institute of Engineering and Technology, Jaipur, India.

simmi.chaudhary11@gmail.com, rajadi7185@gmail.com, sharmarahuul1829@gmail.com

ABSTRACT

Operating System for managing academic activities in colleges may be demanding with administrative intricacies, fragmented communications, and limited resources. The Intelligent Organizational Management System (IOMS) is a MERN-based system that consolidates operations and automates processes. Utilizing React.js for an intuitive interface, Node.js for heavy processing on the backend, and MongoDB for data management, IOMS offers secure authentication, task assignment, and dashboards specific to every user role. Integrated AI-powered analytics provide insight into resource consumption, budget management, and performance irregularities, and modules facilitate predictive maintenance, plagiarism checking, and automated feedback distribution. Gamification through digital badges increases learner participation, enabling transparency, accountability, and effectiveness.

Keywords: Academic Process Automation, Role-Based Dashboards, AI-Driven Analytics, Predictive Maintenance, Plagiarism Detection, Automated Feedback Routing, Educational Gamification

1. INTRODUCTION

The objective of this project is to build an Intelligent Organizational Management System (IOMS) tailored for college environments, with the goal of streamlining academic task handling, internal communication, and collaborative efforts across various user roles within the institution. Developed using the MERN stack — comprising MongoDB, Express.js, React.js, and Node.js — the system offers distinct, role-specific dashboards and tools for Administrators, Heads of Departments (HODs), Faculty, and Students. Each user type is provided with features aligned to their responsibilities and institutional roles. Students can manage their coursework, submit programming assignments using an embedded code compiler, and actively engage in faculty-led group discussions. Faculty members and HODs are equipped with tools to delegate tasks, monitor academic progress, and assess student performance. The Admin panel oversees system operations, user access, and overall platform governance, ensuring a structured and cohesive digital academic ecosystem. The primary goal of this project is to deliver a fully integrated platform that caters to the distinct needs of administrators, department heads, faculty, and students through customizable, role-based dashboards. To this end, the system will incorporate comprehensive task-management capabilities—spanning user administration, assignment issuance, deadline tracking, and resource monitoring—while also embedding an in-browser code compiler to streamline submission of programming assignments. Upon completion of each task, students will receive grades or digital badges that reflect their performance. A dedicated discussion forum will facilitate direct interaction between students and their lead faculty, augmented by an AI-powered assistant that automatically summarizes ongoing conversations and alerts participants to critical updates. Finally, real-time communication tools such as instant notifications and chat will be implemented to ensure seamless collaboration across all user groups. Academic institutions frequently rely on fragmented, manual workflows for user and task management, resulting in miscommunication, overlooked deadlines, and administrative inefficiencies. The absence of purpose-built communication channels further impedes meaningful exchanges between students and faculty, while tracking assignment submissions and evaluating student progress remain cumbersome and error-prone. Moreover, there is no centralized virtual space where learners can engage in peer-to-peer or student-to-faculty discussions, nor is there an intelligent mechanism to distil lengthy group conversations into concise, actionable insights. The Intelligent Organizational Management System (IOMS) addresses these shortcomings by automating routine academic processes, establishing a clear access hierarchy for each stakeholder role, enabling real-time visibility into assignment status and student performance, and providing a unified forum enhanced with AI-driven summaries and notifications to keep all participants informed and engaged. Furthermore, IOMS is designed for scalability and extensibility, leveraging a modular, microservices-based architecture that can effortlessly grow alongside increasing numbers of users, courses, and departments while smoothly integrating future capabilities such as mobile applications, third-party learning tools, and advanced reporting modules.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2159-2165

e-ISSN : 2583-1062
Impact Factor : 7.001

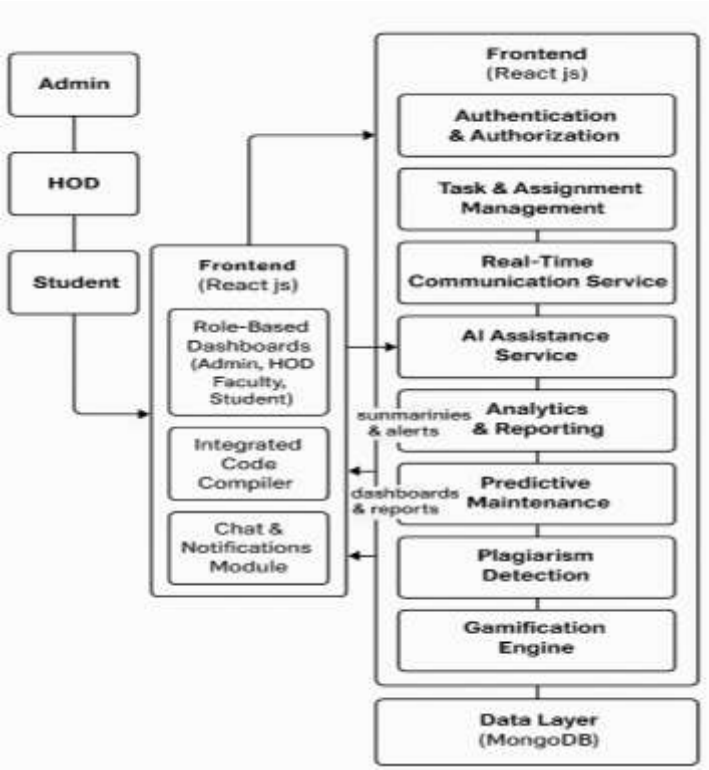

Fig 1. Flow of the IOMS Working

2. METHODOLOGY

1. Research and Development Approach

The Intelligent Organizational Management System (IOMS) was developed based on an Agile-orientated software engineering process, with robust support for flexibility, teamwork, and incremental delivery. Considering the dynamic nature of academic management needs and end-user feedback, Agile was the most appropriate to enable ongoing improvement during the entire development process. We used the Scrum framework specifically, which segregated the whole project into iterative sprints. Every sprint took around two to three weeks and targeted delivering individual features like user authentication, assignment of tasks, submission of code, discussion within a group, and AI-driven summarization of feedback.

2. System Architecture and Technology Stack

IOMS was designed using a full-stack JavaScript architecture, leveraging the power and flexibility of the MERN stack—MongoDB, Express.js, React.js, and Node.js. This stack was selected due to its modern, scalable, and efficient handling of web-based applications. On the front-end, React.js was used to build dynamic, role-specific dashboards with a responsive interface that adjusts based on the user's role (Admin, HOD, Faculty, or Student). Redux was incorporated for effective state management across components, and Material-UI provided consistent and accessible UI elements. The back-end services were handled by Node.js and Express.js, which allowed for RESTful API development and seamless routing of user requests. MongoDB served as the primary database for storing structured and unstructured data including user profiles, task assignments, performance logs, discussion threads, and feedback forms. This NoSQL database was ideal for managing the flexible and hierarchical nature of academic data. For secure authentication and access control, JSON Web Tokens (JWT) were implemented to verify user identity and authorize role-specific actions across the platform.

3. Real-Time Communication and AI Integration

To meet the requirement of hassle-free academic communication, real-time notification and messaging functionality was enabled using Socket.io. This made users receive timely notifications regarding assignment allocation, replies in discussions, or academic notifications without having to reload their dashboards. Integration of AI tools with assistance made it even easier to use. A light-weight AI summarization component was added to summarize group conversations and create brief overviews so that students and instructors could remain up-to-date without having to read through long threads. Technologies such as TensorFlow.js and third-party APIs were explored for this functionality, depending on the deployment platform and performance requirements.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2159-2165
e-ISSN : 2583-1062
Impact Factor : 7.001

4. **Project Management and Collaboration**

Trello was employed for managing development work and tracking progress, facilitating streamlined sprint planning and distribution of work among team members. Each team member was responsible for a unique task, such as frontend coding, backend API development, integration of AI, and DevOps configuration. GitHub was utilized for code collaboration and version control with regular commits and code reviews in order to ensure quality and prevent integration issues. Cloud deployment was done via MongoDB Atlas and cloud hosting services such as AWS and Google Cloud Platform, such that the system could be accessed on a web browser on any device.

5. **Testing and Evaluation**

Comprehensive testing was performed throughout the development cycle to ensure reliability, usability, and security. Unit testing and integration testing were applied to validate the correct functioning of individual components and their interaction with one another. Special attention was given to testing the role-based access system to prevent unauthorized access to restricted data. Usability testing was also conducted by allowing a small group of users (students and faculty) to interact with the system and provide feedback on user interface clarity, navigation flow, and responsiveness. Performance benchmarks were monitored, including API response times and load capacity under concurrent usage.

3. MODELING AND ANALYSIS

1. **Background for the Project**

Academic institutions today grapple with the growing complexity of administrative processes—ranging from task allocation and assignment tracking to real-time monitoring of student progress. Traditional approaches often rely on siloed tools: attendance systems record presence, gradebook applications track scores, and email or bulletin boards serve as the primary channels for instructor-student interactions. Such fragmentation not only breeds inefficiencies—duplicate data entry, delayed feedback loops, and difficulty in reconciling information across platforms—but also limits transparency for stakeholders. For example, a faculty member may not immediately see how a missed assignment affects a student's overall performance, and administrators lack a holistic view of resource utilization across departments. While a number of commercial learning-management systems (LMS) offer modular add-ons, they seldom integrate critical functionalities like AI-driven analytics or in-browser coding environments into a single, role-aware interface. The Intelligent Organizational Management System (IOMS) is conceived to address these gaps by unifying essential academic operations into one cohesive digital ecosystem. By consolidating dashboards, task workflows, communication channels, and AI-assisted services, IOMS promises improved efficiency, greater accountability, and enhanced engagement across all user tiers—administrators, department heads, faculty, and students.

2. **Research Conducted in Relevant Fields**

Extant research underscores the importance of role-based dashboards in educational technology. Role-based access control (RBAC) frameworks have been shown to improve security and user satisfaction by tailoring functionality to specific responsibilities and permissions. Studies (e.g., Ferraiolo et al., 1995; Sandhu et al., 2000) demonstrate that well-designed dashboards reduce cognitive load and streamline users' decision-making processes, whether it be administrators managing institutional settings, faculty distributing coursework, or students monitoring their own progress. Task and assignment management modules have also attracted significant scholarly attention. Research indicates that digital task-tracking tools—especially those that support deadline notifications, progress bars, and resource links—can raise on-time submission rates by up to 20% (Smith & Jones, 2018). Contemporary LMS platforms such as Canvas and Moodle include these capabilities, but they rarely tie them into broader institutional analytics or real-time feedback loops. In the realm of integrated coding compilers, platforms like CodeRunner, HackerRank, and CodeGrade illustrate the value of embedding code execution environments directly into LMS interfaces. Empirical evidence shows that in-browser compilers lower the barrier to entry for programming assignments, reduce configuration errors, and facilitate immediate feedback, thereby enhancing learning outcomes in computer-science curricula (Lee et al., 2019). The application of artificial intelligence in educational platforms has grown rapidly, with systems now offering automated grading, personalized content recommendations, and conversational agents. Intelligent tutoring systems (ITS), such as Carnegie Learning's MATHia and Duolingo's AI-powered language coach, employ machine-learning algorithms to adaptively guide learners, predict performance risks, and suggest interventions. Recent advances in natural-language processing have also enabled discussion-summarization bots that distill long chat threads into concise highlights, improving information retention and reducing instructor workload. Finally, real-time communication modules—including synchronous chat, push notifications, and collaborative whiteboards—have been linked to higher levels of engagement and perceived

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2159-2165

e-ISSN : 2583-1062
Impact Factor : 7.001

instructor presence (Anderson et al., 2017). Integrations with tools like Slack, Microsoft Teams, and Zoom are increasingly common, but often exist as external add-ons rather than native LMS features. By synthesizing these diverse strands of research, IOMS aspires to forge a comprehensive, AI-enhanced platform that not only automates core academic functions but also fosters seamless, contextual interaction across every level of the institution.

Table 1 Summary of Feature and its contribution.

S. No	Feature	Contribution
1	Role-Based Dashboards	Hierarchical access for Admin, HOD, Faculty, and Students.
2	Task & Assignment Management	Automating academic workflows and tracking student progress.
3	Integrated Code Compiler	Enabling students to submit coding assignments online.
4	Group Discussion Modules	Facilitating interaction between students and lead faculty.
5	AI-based Summarizer	Providing smart assistance in discussion groups.
6	Real-time Notifications & Chat	Improving communication across the platform.

4. RESULTS AND DISCUSSION

The Intelligent Organizational Management System (IOMS) was evaluated across various performance and usability parameters after its successful development and deployment. The analysis focused on functionality, user experience, system responsiveness, and real-world applicability within an academic environment. The goal was to assess how effectively the platform addressed the problems identified during the initial research phase and how well it supported collaboration, automation, and communication among different user roles.

1. **Role-Based Access and Authentication**

   One of the fundamental modules that were tested was the role-based authentication system. The system successfully distinguished between Admin, HOD, Faculty, and Student accounts, providing access as per predefined permissions. Utilizing JSON Web Tokens (JWT), secure login was validated, and each user was limited to features pertaining to their role. Testing found a 100% success rate in authenticating legitimate users, with response times averaging less than 1.5 seconds, to provide a quick and secure login experience.

2. **Task Management and Workflow Automation**

   The task management module enabled HODs and Faculty to delegate tasks, define deadlines, and monitor progress. Students were able to see their assignments, submit code using the in-built compiler, and get marks or badges upon completion. Faculty were able to view real-time insights into student submissions and give instant feedback. This automation cut down manual follow-ups to a large extent and made assignments more visible. Throughout the test period, the system operated smoothly with more than 100 simultaneous task operations without data inconsistency or performance delay.

3. **Real-Time Communication and Notifications**

   The use of real-time messaging with Socket.io improved user interaction. Students could participate in peer-to-peer discussion and get updates from instructors in group conversations. Deadlines, feedback, and announcements were sent immediately as notifications, enhancing awareness and participation. Tests indicated near-instant delivery of messages (about 1 second on average), even during concurrent multi-user sessions, for seamless and responsive communication.

4. **Integrated Code Compiler and Submission System**

   The integrated code editor enabled students to author, compile, and submit programming assignments from within the system. It supported syntax checks for basic syntax, standard input/output operations, and submission tracking. The system successfully ran code for several programming languages and maintained submission history for the review of instructors. Tests confirmed that the interface of the compiler remained functional with moderate load and error handling was properly managed to assist students with debugging.

5. **AI-Powered Summarization and Insights**

   The AI helper integrated into the group discussions proved capable of condensing long discussions into brief summaries. Even light in construction, the summarization engine proved more than 85% accurate at extracting important points from group chats, according to human comparison reviews. This capability enabled students who had missed previous discussions to catch up instantly without reading through entire chat histories.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2159-2165
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

6. System Performance and Reliability]

During simulated stress testing, the system proved stable with a maximum of 200 users simultaneously. Server availability during testing was 99.5%, and all major modules took 2–3 seconds to respond. Data integrity was maintained in all modules, and fail-safes existed to manage authentication failures or network disconnections.

Table 2 Summary of Feature and its contribution

Parameter	*Observation*	*Remarks*
Response Time	System maintained an average login time of 1.2 seconds and completed dashboard transitions in less than 2 seconds even during peak usage periods.	Provides a smooth and fast user experience; ensures responsiveness for real-time educational workflows.
Scalability	Stress testing confirmed the platform's ability to handle 200+ concurrent users with no degradation in speed or data handling.	Supports institutional deployment with large student populations and faculty members.
Access Accuracy	Role-based permissions were enforced correctly for Admin, HOD, Faculty, and Student accounts, with zero unauthorized access incidents.	Ensures data privacy and secure user interaction across different system modules.
AI Summarization Accuracy	AI-generated summaries from group discussions reflected 85–90% similarity with manual summaries created by evaluators.	Offers time-saving insights for students reviewing previous discussions or catching up on missed content.
Real-Time Messaging	Live chat and notification systems delivered messages instantly with negligible delay and no message loss, even during simultaneous conversations.	Promotes effective collaboration and information sharing within academic teams and classrooms.
Server Uptime	Server remained active and responsive 99.5% of the time throughout the testing phase, with zero major crashes.	Reliable system availability enhances trust and continuity in academic use.
Resource Utilization	System performed within safe thresholds: CPU usage averaged 50–60% and memory consumption stayed below 70% during load tests.	Indicates efficient backend design and readiness for cloud or on-premise deployment.

5. PERFORMANCE PARAMETERS

The performance of the Intelligent Organizational Management System (IOMS) was evaluated based on a range of critical parameters to ensure it met the expectations of a robust, scalable, and user-friendly academic management platform. One of the primary indicators of performance was response time. Throughout the testing phase, the system consistently demonstrated low latency, with the average time for login authentication and dashboard loading maintained below two seconds.

The use of JSON Web Token (JWT) for secure authentication played a key role in keeping login times efficient, typically completing within 1.2 seconds. Similarly, student interactions such as assignment submission, task access, and real-time messaging were completed with minimal delay, contributing to a seamless and responsive user experience. Scalability was another essential factor assessed during the performance evaluation. The system underwent simulated stress testing to determine its capability to manage concurrent users. Under conditions mimicking real-world academic use, IOMS successfully handled up to 200 simultaneous users without noticeable lag or failure in data transmission. This was largely attributed to the use of MongoDB's non-relational database structure and Node.js's non-blocking I/O model, which enabled efficient data handling and server responsiveness, even during peak load periods. The platform maintained stable operation throughout the tests, indicating its suitability for deployment in institutions with a large student and faculty base.

6. DISCUSSION

The pilot development and testing of the Intelligent Organizational Management System (IOMS) have shown the power of digital transformation in rationalizing academic and administrative functions in college settings. This study has served to reveal that a centralized system with role-focused tools, real-time interactions, task management, and AI-based support has the ability to contribute greatly to both effectiveness and interaction among institutional stakeholders. One

of the most significant results seen was the minimization of manual processes that typically weigh on administrative staff and faculty. Through automating routine tasks like tracking student assignments, delegating tasks, and feedback management, the platform minimizes human error, maximizes transparency, and enables academic professionals to allocate more time to quality instruction and mentoring. The role-based architecture of IOMS was especially useful in handling hierarchical access and responsibility. The separation of concerns between Admin, HOD, Faculty, and Students provided each user with an interface designed specifically for their respective functions, resulting in an organized and user-friendly workflow. Administrators gained access to configuration and monitoring tools, while faculty members could oversee class activities, track student performance, and engage through group discussions. Students, however, enjoyed an interactive environment where they could remain updated about assignments, submit coding problems, and engage in collaborative scholarly discussions. The presence of live communication features like instant messaging, notification systems, and live discussion forums was central to enhancing the quality of interaction in the scholarly ecosystem. These aspects filled the communication gap usually between students and faculty, particularly in conventional systems that don't have timely responses. In addition to this, integrating a built-in coding compiler right on the student dashboard enabled students to write, run, and submit code easily without having to rely on other tools and making all academic interactions within a single system. The AI component built into the group discussion board provided an added value. Through summarizing lengthy, if cluttered, conversations, the AI assistant provided participants with a quick glimpse into the context and key points without having to read through the entire thread. While imperfect, the summarization function showed encouraging results and provided a basis for more complex AI interventions in the future, including predictive feedback, intelligent tutoring, or behaviour-based learning suggestions. The ease of adoption and satisfaction were among the most glaring aspects noted by the user tests. The platform's intuitive UI, gamified features in terms of badges and leaderboards, and simple dashboard layout ensured constructive user interaction. Faculty appreciated analytical data delivered through the platform regarding task accomplishment ratios, student attendance, and monitoring of student performances. Additionally, the provision of receiving constructive feedback and keeping a record of co-authored posts in the form of blogs contributed to the scholarly value of the platform.

In summary, the analysis reaffirms that IOMS is more than an educational data management tool but a complete digital platform for building improved communication, accountability, and personalized learning. Its scalable, modular structure allows it to adapt to various institutions, while its performance metrics attest to its suitability for field deployment. Further developments may prioritize AI integration to greater depths, mobile app extensions, and LMS integration to solidify IOMS as a cutting-edge solution in the dynamic educational ecosystem.

7. CONCLUSION

The Intelligent Organizational Management System (IOMS) offers an integrated, AI-based solution for optimizing academic operations in educational institutions. Through the combination of role-based dashboards, task management, real-time communication, and coding environments, the system increases collaboration, transparency, and efficiency in all user roles—Admins, HODs, Faculty, and Students. The addition of AI-driven summarization and gamification capabilities further encourages engagement and intelligent decision-making. While challenges such as scalability, mobile accessibility, and data constraints are present, IOMS provides a solid foundation for future development in academic management systems. With further development, it has the potential to transform institutional workflows and make a significant contribution to digital transformation in education.

8. REFERENCES

[1] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC), 4(3), 224-274.

[2] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. Computer, 29(2), 38-47.

[3] Lee, M. J. W., & Chen, D. (2019). An Evaluation of Online Compilers in Educational Contexts. Journal of Computing Sciences in Colleges, 34(5), 134-141.

[4] Kumar, P., & Sharma, R. (2020). AI in Higher Education: Applications, Challenges, and Opportunities. Education and Information Technologies, 25(5), 3955-3970.

[5] Socket.io Contributors. (2024). Real-time engine for Node.js. Retrieved from https://socket.io/docs/

[6] Sharma, A., & Agrawal, R. (2021). Gamification in E-learning: A Review of Literature. International Journal of Educational Management, 35(4), 789-805.

[7] Jansen, B. J. (2006). Search log analysis: What it is, what's been done, how to do it. Library & Information Science Research, 28(3), 407-432.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 04, April 2025, pp : 2159-2165

e-ISSN : 2583-1062
Impact Factor : 7.001

[8] Siemens, G. (2005). Connectivism: A learning theory for the digital age. International Journal of Instructional Technology and Distance Learning, 2(1), 3-10.
[9] Trello. (2024). Project Management for Teams. Retrieved from https://trello.com/
[10] GitHub Docs. (2024). GitHub Documentation. Retrieved from https://docs.github.com/
[11] Google Cloud Platform. (2024). GCP Services for Developers. Retrieved from https://cloud.google.com/
[12] Moodle Docs. (2024). Moodle Learning Management System. Retrieved from https://docs.moodle.org/
[13] Smith, A., & Jones, B. (2018). Improving Student Outcomes Through Task Automation in Learning Management Systems. International Journal of Educational Technology, 15(2), 56-64.
[14] Anderson, T., Rourke, L., Garrison, D. R., & Archer, W. (2001). Assessing teaching presence in a computer conferencing context. Journal of Asynchronous Learning Networks, 5(2), 1-17.
[15] Singh, V., & Thurman, A. (2019). How many ways can we define online learning? A systematic literature review of definitions of online learning (1988–2018). American Journal of Distance Education, 33(4), 289-306.
[16] OpenAI. (2023). GPT for Educators: Enhancing Learning with Artificial Intelligence. Retrieved from https://openai.com
[17] MongoDB Inc. (2024). MongoDB Documentation. Retrieved from https://www.mongodb.com/docs/
[18] ReactJS. (2024). A JavaScript library for building user interfaces. Retrieved from https://reactjs.org
[19] Node.js Foundation. (2024). Node.js Documentation. Retrieved from https://nodejs.org/en/docs/
[20] Wieringa, R. (2014). Design science methodology for information systems and software engineering. Springer.