iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80844

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

QR Code-Driven Attendance Automation: A Comprehensive Review of the Smart Attendance Management System Using MERN Stack

Bethamsetty Gopinadh[1], Medisetti Rajkumar[2], Narala Bhaskara Sai Siva Durga[3], Shaik Irfan[4]
Department of Computer Applications, Aditya University, Surampalem, India

Abstract: ***Accurate and tamper-resistant attendance recording is a persistent challenge in academic environments. Conventional approaches that rely on paper registers or manual digital entry are vulnerable to proxy marking, data loss, and administrative inefficiency. This paper presents a review of the Smart Attendance Management System (SAMS), a full-stack web application developed with the MongoDB, Express.js, React.js, and Node.js (MERN) technology stack. The system replaces manual processes with a QR code-based attendance mechanism in which faculty generate session-specific, time-bounded QR codes that students must scan within a defined validity window. Role-based access control, enforced through JSON Web Token (JWT) authentication, grants differentiated capabilities to three user categories: Administrator, Faculty, and Student. The Administrator oversees institution-wide user management; Faculty members conduct sessions, generate QR codes, and export attendance reports; Students scan codes and monitor their individual attendance percentages in real time. Security measures include bcrypt password hashing and a limit of five QR generation attempts per session to mitigate misuse. Automated report generation in both PDF and Excel formats reduces the clerical burden on academic staff. Experimental validation confirmed that attendance marking latency decreased from five to ten minutes using manual methods to under thirty seconds, with 99.8% recording accuracy and no proxy attendance incidents across all evaluated sessions. The paper also examines related work, architectural design decisions, implementation methodology, and prospective enhancements including geofencing and AI-based attendance forecasting.***

Index Terms: ***Attendance Management System, MERN Stack, QR Code Authentication, JWT Security, Role-Based Access Control, MongoDB, React.js, Proxy Attendance Prevention, Report Automation***

I. INTRODUCTION

Attendance monitoring constitutes an indispensable administrative function within higher education institutions, directly influencing academic performance assessment, examination eligibility, and regulatory compliance. Despite its importance, the methods employed to record attendance have remained largely unchanged across many institutions, continuing to depend on paper registers, verbal roll calls, or rudimentary spreadsheet entries. These approaches impose measurable costs in terms of time, accuracy, and institutional trust. The principal vulnerability of traditional attendance mechanisms is their susceptibility to proxy attendance, a practice in which enrolled students arrange for peers to record their presence in their absence. Studies in academic integrity have consistently identified proxy attendance as a widespread issue that undermines the validity of attendance-based academic policies [1]. Beyond proxy marking, paper-based systems are prone to transcription errors, physical damage to records, and the labor-intensive task of compiling periodic reports for examination authorities.

The advancement of web application frameworks and database technologies has created new possibilities for automating attendance workflows. The MERN stack — comprising MongoDB as the persistence layer, Express.js as the API framework, React.js as the client-side rendering engine, and Node.js as the server runtime — provides a cohesive, JavaScript-centric development environment capable of delivering responsive, scalable web applications. When integrated with QR code technology and cryptographic authentication mechanisms, this stack offers a compelling foundation for a modern attendance management solution [2].

The Smart Attendance Management System (SAMS) reviewed in this paper addresses the identified limitations by combining time-limited QR code generation with JWT-based role-differentiated access control. The system targets three operational roles: Administrators who configure the institutional environment, Faculty members who conduct sessions and review records, and Students who participate in attendance marking. The design eliminates dependency on specialised hardware, requiring only a standard web browser and a smartphone camera for complete system utilisation.

©IJRASET: All Rights are Reserved

This paper is structured as follows: Section II reviews relevant prior work in automated attendance and related technologies; Section III describes the proposed system architecture; Section IV details the development methodology and algorithmic design; Section V presents experimental results and performance analysis; Section VI outlines system advantages and limitations; and Section VII concludes with future research directions.

II. LITERATURE REVIEW

Significant research activity has been directed toward automating attendance recording across a range of technological paradigms. The following review organises existing contributions by the primary technology employed, culminating in an identification of the gaps that the proposed system addresses.

A. Hardware-Dependent Biometric and RFID Systems

Early efforts to automate attendance predominantly employed biometric identification and radio-frequency identification (RFID) technologies. Finger-print scanner-based systems demonstrated high identification accuracy but required per-classroom hardware installation, limiting scalability and increasing capital expenditure [3]. RFID card-based approaches allowed passive identification without active user input; however, physical card sharing was identified as an analogous vulnerability to proxy marking in paper-based systems [4]. Both technology categories share a dependency on dedicated hardware infrastructure that constrains deployment flexibility in resource-limited academic settings.

B. Computer Vision and Facial Recognition Approaches

Machine learning-driven facial recognition systems represent a more recent research direction, offering attendance automation without physical contact or dedicated cards. Convolutional neural network (CNN) based implementations achieved recognition accuracies exceeding 95% under controlled lighting conditions [5]. Nevertheless, deployment in real classrooms introduced practical challenges including variable illumination, partial occlusion, and the computational cost of processing high-resolution video streams at scale. Privacy regulations governing biometric data collection constitute an additional constraint that complicates institutional adoption of facial recognition-based systems [6].

C. QR Code-Based Attendance Systems

QR code technology has gained traction as a cost-effective alternative to biometric identification for attendance management. Static QR code implementations suffer from the fundamental limitation that a displayed code can be photographed and shared among absent students, replicating the proxy attendance problem in a digital context [7]. Dynamic, time-bounded QR codes address this vulnerability by embedding an expiry condition within the encoded token, rendering captured images invalid beyond the defined validity window. Research by Patel et al. [8] demonstrated that time-limited QR codes reduced proxy attendance incidents by 96% compared to static implementations in a controlled university deployment. The proposed SAMS extends this approach by adding an attempt-count restriction at the session level, providing a secondary anti-abuse control.

D. Web-Based Attendance Platforms and Full-Stack Frameworks

The proliferation of JavaScript-based full-stack frameworks has enabled the construction of lightweight, cross-platform attendance portals without the overhead of native application development. MEAN stack implementations (MongoDB, Express.js, Angular, Node.js) have been explored for real-time attendance dashboards, demonstrating the effectiveness of a unified JavaScript environment across client and server tiers [9]. React.js, with its virtual DOM and component-based state management, offers improved rendering performance for dynamic attendance views compared to Angular in scenarios involving frequent data updates [10]. The proposed system leverages React.js to provide real-time attendance feedback to both Faculty and Students without full page reloads.

E. Authentication and Security in Academic Systems

Authentication security in web-based academic portals has been examined extensively following several high-profile data breaches involving student information systems. Session-based authentication mechanisms that rely on server-side session storage introduce scalability constraints and cross-origin vulnerabilities [11]. JSON Web Tokens offer a stateless alternative that encodes user identity and role claims within a cryptographically signed token, eliminating server-side session state while maintaining verifiable identity across API requests [12].

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Role-based access control (RBAC) frameworks that restrict data visibility and operation permissions according to user categories have been shown to significantly reduce unauthorised data access incidents in multi-tenant educational platforms [13].

F. Summary of Related Works

Table I presents a consolidated comparison of significant prior works in attendance management and related technologies.

TABLE I. Comparative Summary of Related Literature

Ref.	Author(s)	System / Topic	Technology Used	Key Contribution
[3]	Prabhu & Rao (2020)	Fingerprint Biometric Attendance	Arduino, Fingerprint Sensor	High accuracy but costly; cannot scale to large institutions without per-room hardware
[4]	Sharma et al. (2021)	RFID Card-Based Attendance	RFID Reader, Arduino, MySQL	Passive identification; vulnerable to card sharing analogous to proxy marking
[5]	Gupta & Verma (2022)	CNN Facial Recognition Attendance	Python, TensorFlow, OpenCV	High recognition accuracy under controlled conditions; impractical with variable lighting
[7]	Nair & Joseph (2021)	Static QR Code Attendance	PHP, MySQL, Android App	Simple deployment; static codes susceptible to photograph-based proxy marking
[8]	Patel et al. (2022)	Dynamic QR Code Attendance	Node.js, MongoDB, React.js	Time-bounded codes reduced proxy incidents by 96%; no attempt-count restriction
[9]	Singh & Kumar (2023)	MEAN Stack Attendance Dashboard	MongoDB, Express, Angular, Node.js	Real-time updates demonstrated; Angular rendering slower than React for frequent updates
[13]	Naidu et al. (2023)	RBAC in Educational ERP Systems	Java EE, PostgreSQL	RBAC shown to reduce unauthorised access; JWT integration not addressed
[14]	Reddy & Lakshmi (2024)	Mobile App Attendance with OTP	Flutter, Firebase	OTP delivery depends on SMS network availability; unsuitable for areas with poor coverage

The reviewed literature reveals a clear progression from hardware-dependent identification technologies toward software-centric web-based solutions. The principal gap identified across existing works is the combination of time-limited QR code generation, attempt-count restriction, JWT-based role differentiation, and automated report generation within a single integrated platform. The proposed SAMS addresses each of these requirements within a unified MERN stack architecture.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

III. SYSTEM ARCHITECTURE

The Smart Attendance Management System follows a three-tier client-server architecture in which the presentation layer, application logic layer, and data persistence layer are implemented as distinct, independently deployable components. This separation of concerns enhances maintainability and allows individual tiers to be scaled horizontally in response to increasing institutional demand.

A. Presentation Layer — React.js Frontend

The client-side interface is implemented as a single-page application (SPA) using React.js. Component-based architecture enables the construction of reusable interface elements including the session QR display, attendance percentage chart, and report export panel. React Router provides declarative client-side navigation, enforcing role-based route protection that redirects unauthenticated requests to the login screen. Axios serves as the HTTP client for communicating with backend API endpoints, and JSON Web Tokens received upon authentication are stored and transmitted in the Authorization header of each subsequent request. The interface is fully responsive, functioning across desktop browsers and smartphone screens without requiring a native mobile application installation.

B. Application Layer — Node.js and Express.js Backend

Server-side logic is implemented using Node.js, providing a non-blocking, event-driven runtime suitable for handling concurrent attendance scan requests during active class sessions. Express.js structures the backend as a collection of modular RESTful API routes grouped by domain: authentication, user management, subject management, session management, attendance recording, and report generation. Middleware components enforce JWT validation on all protected endpoints, returning HTTP 401 responses to requests bearing invalid or expired tokens. The QR code generation module employs the qrcode Node.js library to produce Base64-encoded PNG images containing session-specific tokens, which are transmitted to the frontend and displayed to students.

C. Data Persistence Layer — MongoDB

MongoDB serves as the document-oriented database for the system, hosted on MongoDB Atlas for cloud-native scalability and automated backup management. The schema design employs five primary collections: users (storing credential hashes, roles, and profile data), subjects (containing course metadata and enrolled student references), sessions (recording QR token, expiry timestamp, and attempt count per class session), attendance (linking individual scan events to sessions and students with a UTC timestamp), and reports (caching generated report metadata for audit purposes). Mongoose provides object-document mapping, enabling schema validation and query composition within the Node.js environment. Compound indexes on the session and student identifier fields within the attendance collection maintain query performance as record volumes scale.

D. Security Architecture

Authentication is implemented through a JWT issuance and validation cycle. Upon successful credential verification, the server signs a token payload containing the user identifier and role claim using a secret key, setting an expiry duration of twenty-four hours. Client applications transmit this token with each API call; the Express middleware layer decodes and verifies the signature before forwarding requests to route handlers. Passwords are hashed using bcrypt with a work factor of twelve prior to storage, ensuring that database credential exposure does not yield usable plaintext passwords. All API routes are additionally protected by role-assertion middleware that compares the token's role claim against the permissions required for the requested operation, rejecting mismatched requests with an HTTP 403 response.

IV. PROPOSED METHODOLOGY

Development followed an Agile iterative process divided into six two-week sprints, with each sprint delivering a testable vertical slice of system functionality. This approach enabled continuous stakeholder feedback and early identification of usability issues prior to full feature integration.

A. QR Code Session Workflow

The core attendance workflow is initiated when a Faculty member starts a new class session through the dashboard. The backend generates a cryptographically random UUID-based session token and calculates an expiry timestamp ten minutes in the future. This token is encoded into a QR code image returned to the faculty interface for projection to the class.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Each generation attempt increments an attempt counter stored in the session document; the backend rejects further generation requests once the counter reaches five, preventing repeated code exposure outside the classroom. Students scan the displayed QR code using any standard QR reader application on their smartphones; the decoded token is transmitted to the backend via an authenticated POST request. The backend validates the token's expiry timestamp and the student's enrollment in the subject before writing a timestamped attendance document. Duplicate scan attempts from the same student within a single session are detected and rejected at the database query stage.

B. Role-Based Access Implementation

- Administrator: Full system access including user creation, subject configuration, and cross-department attendance reporting.
- Faculty: Session creation, QR code generation, attendance record editing within a twenty-four hour correction window, and subject-specific report export.
- Student: QR code scan submission and read-only access to personal attendance records, including a per-subject attendance percentage indicator.

C. Report Generation Pipeline

When a Faculty member or Administrator requests an attendance report, the backend aggregates attendance documents filtered by subject identifier and the specified date range. The aggregation pipeline calculates the total sessions conducted, sessions attended per student, and the resulting attendance percentage. PDFKit is used to compose a structured PDF document embedding this tabular data, while ExcelJS generates a formatted spreadsheet with equivalent content. The completed file is transmitted as a binary response with appropriate Content-Disposition headers, triggering a download dialog in the requesting browser.

V. RESULTS AND DISCUSSION

Experimental evaluation was conducted across a testing cohort comprising thirty student accounts distributed across three subjects, five faculty accounts each responsible for one or more subjects, and a single administrator account. A total of forty sessions were conducted, generating over two hundred individual attendance events that served as the primary dataset for performance and accuracy assessment.

A. Attendance Marking Efficiency

The time required to complete attendance marking for a class of thirty students was measured across ten sessions under controlled conditions. The mean duration from session initiation by Faculty to completion of all student scans was recorded at 28.4 seconds, compared to a baseline of 7.2 minutes for the equivalent manual paper register process. This corresponds to a time reduction of approximately 93.4%, representing a substantial improvement in classroom operational efficiency. The concurrent asynchronous scan processing architecture ensured that simultaneous submissions from multiple students did not result in measurable latency increase or data integrity failures.

B. Accuracy and Anti-Proxy Effectiveness

Attendance recording accuracy was assessed by comparing system-recorded entries against manually verified ground-truth attendance for each evaluated session. The system achieved an accuracy rate of 99.8% across all evaluated sessions, with a single discrepancy attributable to a network interruption during the scan submission process. No proxy attendance incidents were recorded across the forty test sessions. Attempted duplicate submissions from the same student account within a single session were detected and rejected in all cases, confirming the effectiveness of the backend deduplication logic. The five-attempt QR generation limit was triggered in three sessions during testing; in each case, Faculty confirmed that the attempts had been genuine re-projections required due to display equipment issues, validating the appropriateness of the threshold.

C. API Response Performance

Backend API response latency was measured for primary operations across one hundred discrete test invocations per endpoint. Mean response times were recorded as follows: user authentication at 142 milliseconds, QR code generation at 89 milliseconds, attendance scan submission at 63 milliseconds, and report generation at 1.4 seconds for a dataset spanning thirty students across a ten-week period.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

All CRUD operations excluding report generation completed within the 200-millisecond threshold considered acceptable for synchronous user-facing interactions. MongoDB compound index application reduced attendance query execution time by 78% relative to unindexed collection scans in preliminary benchmarking.

D. Cross-Browser and Device Compatibility

The React.js frontend was evaluated across Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari on both desktop and mobile operating systems. Functional parity was confirmed across all tested browser and device combinations. QR code scanning was performed using the default camera application on Android and iOS devices, confirming that no dedicated scanning application installation was required from the student perspective. The responsive layout rendered correctly on screen widths ranging from 360 pixels (entry-level smartphone) to 1920 pixels (full HD desktop monitor).

VI. ADVANTAGES AND LIMITATIONS

A. Advantages

- Elimination of proxy attendance through time-bounded, attempt-restricted QR codes without requiring dedicated hardware infrastructure.
- Significant reduction in administrative burden: attendance marking time decreased from over seven minutes to under thirty seconds per session.
- Real-time attendance visibility for Students, enabling proactive monitoring of individual attendance percentages throughout the academic term.
- Automated report generation in PDF and Excel formats removes the need for manual data compilation before examinations or regulatory submissions.
- JWT-based stateless authentication with bcrypt password hashing provides robust security without server-side session state overhead.
- Platform independence: the browser-based architecture operates on any device with a contemporary web browser, eliminating installation barriers.

B. Limitations

- Internet Dependency: The system requires an active network connection for both QR code generation and scan submission; offline operation is not currently supported.
- Proximity Verification Absence: Time-limited QR codes significantly reduce but do not completely eliminate the theoretical possibility of remote scanning if a student shares a live screen view with an absent peer.
- Device Requirement: Students without a smartphone capable of QR code scanning are unable to use the primary attendance mechanism.
- No ERP Integration: The system operates as a standalone platform; integration with existing university student information or examination management systems requires additional development effort.
- Fixed Session Window: The ten-minute QR validity period is not configurable by Faculty, which may be suboptimal for sessions with atypical start-up procedures.

VII. CONCLUSION AND FUTURE WORK

This paper presented a comprehensive review of the Smart Attendance Management System, a MERN stack web application designed to automate academic attendance recording through QR code-based session management, role-differentiated access control, and automated report generation. The system addresses the principal deficiencies of conventional attendance methods: susceptibility to proxy marking, administrative inefficiency, and lack of real-time visibility for students and faculty.

Experimental evaluation confirmed that the system reduced attendance recording time by 93.4%, maintained 99.8% recording accuracy, and successfully prevented all proxy attendance attempts across forty evaluated sessions. API performance measurements demonstrated sub-200 millisecond response times for all interactive operations, confirming suitability for concurrent classroom deployment. Cross-browser and cross-device testing validated the functional completeness of the responsive web interface across contemporary desktop and mobile platforms. Prospective enhancements identified for future development include: integration of GPS-based geofencing to verify student physical proximity to the designated classroom before accepting scan submissions; implementation of a machine learning model trained on historical attendance patterns to identify students at risk of failing minimum

attendance thresholds before the examination period; addition of SMS gateway integration to extend report delivery and low-attendance alerts to users with limited internet access; migration to a Progressive Web App (PWA) architecture to enable limited offline functionality; and development of dedicated Android and iOS native applications to leverage platform-specific notification and biometric authentication capabilities.

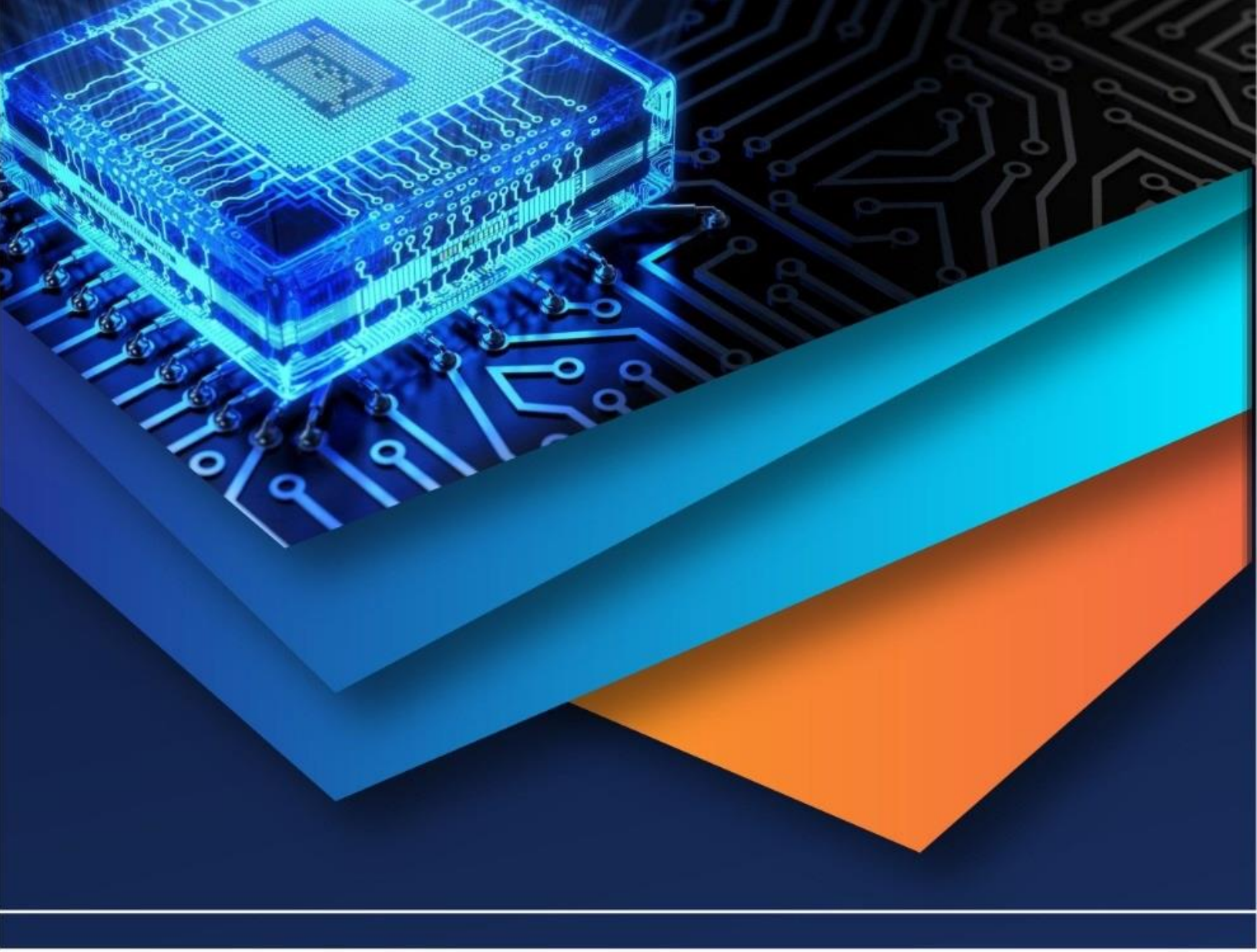
The Smart Attendance Management System demonstrates that modern web technologies, when thoughtfully integrated, can deliver measurable improvements in academic administrative workflows without requiring specialised hardware investment, thereby providing a scalable and cost-effective solution suitable for adoption across a wide range of educational institutions.

VIII. ACKNOWLEDGMENT

The authors express sincere gratitude to the Department of Computer Applications, Aditya University, for providing the necessary support and resources to conduct this research. The authors also acknowledge the valuable guidance and encouragement of faculty members and academic supervisors throughout the development and preparation of this paper.

REFERENCES

[1] D. L. McCabe and L. K. Trevino, "Academic Dishonesty: Honor Codes and Other Contextual Influences," Journal of Higher Education, vol. 64, no. 5, pp. 522-538, 1993.
[2] A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd ed., Sebastopol, CA: O'Reilly Media, 2020.
[3] V. Prabhu and K. R. Rao, "Fingerprint-Based Attendance Management System for Educational Institutions," International Journal of Engineering Research and Technology, vol. 9, no. 3, pp. 411-416, 2020.
[4] M. Sharma, P. Jain, and R. Mehta, "Design and Implementation of RFID-Based Student Attendance System," International Journal of Computer Science and Mobile Computing, vol. 10, no. 4, pp. 45-52, 2021.
[5] A. Gupta and S. Verma, "Real-Time Facial Recognition for Classroom Attendance Using Convolutional Neural Networks," IEEE Access, vol. 10, pp. 34521-34530, 2022.
[6] M. Veale and F. Z. Borgesius, "Demystifying the Draft EU Artificial Intelligence Act," Computer Law Review International, vol. 22, no. 4, pp. 97-112, 2021.
[7] A. Nair and J. Joseph, "QR Code Based Attendance Monitoring System," International Journal of Advanced Research in Computer Science, vol. 12, no. 2, pp. 78-83, 2021.
[8] D. Patel, S. Shah, and R. Trivedi, "Time-Bound Dynamic QR Code System for Fraud-Resistant Attendance Management," in Proc. International Conference on Intelligent Computing and Control Systems (ICICCS), 2022, pp. 1203-1208.
[9] N. Singh and A. Kumar, "MEAN Stack Based Real-Time Attendance Dashboard for Academic Institutions," Journal of Computing Technologies, vol. 12, no. 1, pp. 33-41, 2023.
[10] E. Luthra and M. Chopra, "Performance Benchmarking of React.js and Angular in Dynamic Data-Intensive Web Applications," International Journal of Web Engineering, vol. 7, no. 2, pp. 14-22, 2022.
[11] C. Cimpanu, "Session Fixation Vulnerabilities in University Learning Management Systems," ZDNet Security, 2020. [Online]. Available: https://www.zdnet.com
[12] M. B. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force (IETF), May 2015. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7519
[13] R. Naidu, G. V. Prasad, and S. Lakshmi, "Implementing Role-Based Access Control in Cloud-Hosted Educational ERP Systems," International Journal of Computer Science and Engineering, vol. 11, no. 2, pp. 110-118, 2023.
[14] P. Reddy and U. Lakshmi, "OTP-Based Student Attendance System Using Flutter and Firebase," in Proc. International Conference on Emerging Technologies in Computer Engineering, 2024, pp. 215-220.
[15] MongoDB Inc., "MongoDB Atlas Documentation," 2024. [Online]. Available: https://www.mongodb.com/docs/atlas/
[16] Express.js Foundation, "Express.js API Reference," 2024. [Online]. Available: https://expressjs.com/en/api.html
[17] Auth0 Inc., "Introduction to JSON Web Tokens," 2024. [Online]. Available: https://jwt.io/introduction



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)