IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.135117

A Role-Based, Token-Secured Web Architecture for Digital Classroom Attendance Using the Java Full-Stack Paradigm

S. Sri Surya Ambika[1], P. Sreenivasa Reddy*[2]

PG Scholar Department of Computer Science, S.V.K.P & Dr. K.S Raju Arts and Science College (Autonomous), Penugonda, Affilliated to Adikavi Nannaya University[1]

*Associate Professor, Department of Master of Computer Applications S.V.K.P & Dr. K.S Raju Arts and Science College (Autonomous), Penugonda, Affilliated to Adikavi Nannaya University[2]

Abstract: Manual classroom attendance, whether recorded on paper or in spreadsheets, is slow, error-prone, and difficult to audit, and it scales poorly as enrolment grows. This paper presents a web-based attendance management system engineered with the Java full-stack paradigm to digitize and secure the entire capture-to-report cycle. The platform follows a clean three-tier separation in which a reactive single-page client communicates over a stateless application programming interface with a service layer built on a contemporary Java enterprise framework, while structured records are persisted through an object-relational mapping abstraction. Security is enforced through signed, role-bearing tokens that establish a stateless authentication context on every request; a dedicated filter validates the token, extracts the embedded role, and authorizes access so that administrative and teaching capabilities remain strictly partitioned. Attendance marking is designed to be idempotent: a uniqueness constraint on the session-student pair, combined with an upsert procedure, guarantees that repeated submissions converge to a single correct record rather than producing duplicates. Teachers may only operate on sessions they own, a rule enforced in the service layer. An administrative module aggregates institution-wide counts and recent activity for oversight. Experimental evaluation on a local deployment shows median application-programming-interface latencies between roughly seven and fifty-eight milliseconds and sustained mark-attendance throughput exceeding twenty thousand requests per second under load. The system offers a lightweight, secure, and maintainable foundation for institutional attendance tracking and demonstrates sound full-stack engineering practice.

Keywords: idempotent attendance management, Java full stack, Spring Boot, JSON Web Token, role-based access control, RESTful web services, single-page application transactions

I. INTRODUCTION

Accurate record-keeping of student presence is a routine yet consequential administrative obligation in educational institutions. Attendance data inform academic eligibility, regulatory compliance, and early identification of disengaged learners. Despite this importance, a large number of institutions continue to rely on manual procedures, in which an instructor calls a register or circulates a signature sheet and later transcribes the result. Such processes consume teaching time, are vulnerable to transcription mistakes and proxy marking, and yield data that are cumbersome to aggregate or audit [1], [2].

The migration of administrative functions to web platforms offers an obvious remedy, and a substantial industry has grown around digital attendance. However, many available solutions are either heavyweight commercial suites that are costly and complex to operate, or ad-hoc applications that neglect the security of what is, in effect, a sensitive institutional record. In particular, weak authentication and insufficient separation between administrative and teaching privileges expose attendance data to tampering. A further recurring defect is the mishandling of repeated submissions: when a teacher re-saves a roster, naive implementations create duplicate entries that corrupt downstream statistics.

The problem addressed in this work is therefore the design of an attendance platform that is simultaneously lightweight, secure, and correct under realistic usage. The motivation is to demonstrate that a disciplined application of the Java full-stack paradigm, pairing a modern reactive client with a robust enterprise service layer, can deliver these properties without unnecessary infrastructure. The research objectives are to architect a cleanly layered system, to enforce stateless role-based access using signed tokens, and to guarantee idempotent attendance capture.

The principal contributions of this paper are as follows:

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.135117

- A three-tier web architecture for classroom attendance built on the Java full-stack paradigm, cleanly separating a single-page client, a stateless service layer, and a relational persistence tier.
- A stateless security design in which signed, role-bearing tokens are validated by a dedicated request filter and mapped to fine-grained authorities, strictly partitioning administrative and teaching capabilities and enforcing per-teacher ownership of sessions.
- An idempotent attendance-marking mechanism that combines a database-level uniqueness constraint on the session-student pair with an upsert procedure, eliminating duplicate records under repeated submission.
- An empirical evaluation of latency and throughput that characterises the responsiveness and scalability of the proposed design.

II. LITERATURE REVIEW

Research on automated attendance spans a spectrum from biometric and computer-vision capture to conventional software-engineered web systems. Vision-based methods that recognise faces from a classroom image have attracted considerable attention for removing manual effort entirely, yet they raise privacy concerns, demand favourable imaging conditions, and incur non-trivial hardware and computational costs [3], [4]. Radio-frequency identification and near-field communication approaches, in which students tap a card or device, reduce manual transcription but require dedicated readers and remain susceptible to proxy attendance when cards are shared [5], [6].

Biometric techniques using fingerprints have been deployed to deter proxy marking, achieving good accuracy at the expense of specialised sensors and hygiene considerations that became especially salient in recent years [7]. Mobile and location-based schemes exploit a student's smartphone, sometimes combining global-positioning or short-range signals to confirm classroom presence, but they presuppose universal device ownership and can be circumvented by location spoofing [8].

A complementary and pragmatic line of work focuses on software-engineered web applications that digitise the instructor-driven workflow rather than the act of physical sensing. Studies employing layered web stacks report substantial reductions in administrative effort and improved data integrity relative to paper registers [9], [10]. Within this category, the choice of technology stack and security model is decisive. The Java enterprise ecosystem, and the convention-over-configuration approach of its dominant application framework, is widely adopted for institutional systems owing to its maturity, transactional reliability, and rich security tooling [11], [12]. For client interfaces, component-based reactive libraries have become standard for delivering responsive single-page experiences [13].

On authentication, the literature has converged on stateless, token-based schemes for distributed web services, where a signed token carried in each request removes the need for server-side session storage and eases horizontal scaling [14], [15]. Role-based access control remains the canonical model for partitioning privileges in multi-actor administrative systems [16]. Table I contrasts representative approaches. The comparison reveals that sensing-centric methods, while powerful, carry privacy, cost, or reliability burdens, whereas many web-based systems underspecify their security and data-integrity guarantees. The present work occupies the pragmatic centre: a privacy-respecting, low-cost, instructor-driven web system that nonetheless treats stateless role-based security and idempotent data capture as first-class design concerns.

TABLE I. Comparison of Representative Attendance Approaches

Approach	Representative basis	Strengths	Limitations
Face recognition	Classroom image analysis [3], [4]	Hands-free capture	Privacy; lighting; cost
RFID / NFC	Card or tag tap [5], [6]	Fast; low transcription	Readers needed; proxy risk
Fingerprint biometric	Sensor matching [7]	Deters proxy	Hardware; hygiene
Mobile / location	GPS or short-range [8]	Uses student devices	Device reliance; spoofing
Generic web systems	Layered web app [9], [10]	Low cost; auditable	Security often underspecified
Proposed system	Java full stack + JWT	Secure; idempotent; lightweight	Instructor-driven entry

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.135117

III. PROPOSED METHODOLOGY

A. System Architecture

The platform is organised into the three tiers shown in Fig. 1. The client tier is a reactive single-page application that renders the login screen, an administrative dashboard, the teacher's session list, and the attendance roster. It communicates exclusively through a stateless application programming interface. The application tier, implemented on a contemporary Java enterprise framework, hosts the security filter chain, the controllers that expose the interface, and the services that encapsulate business rules. The persistence tier maps domain objects to relational tables through an object-relational abstraction, and the data tier is an in-memory relational database that renders the reference deployment self-contained while preserving full relational semantics, including foreign keys and uniqueness constraints.

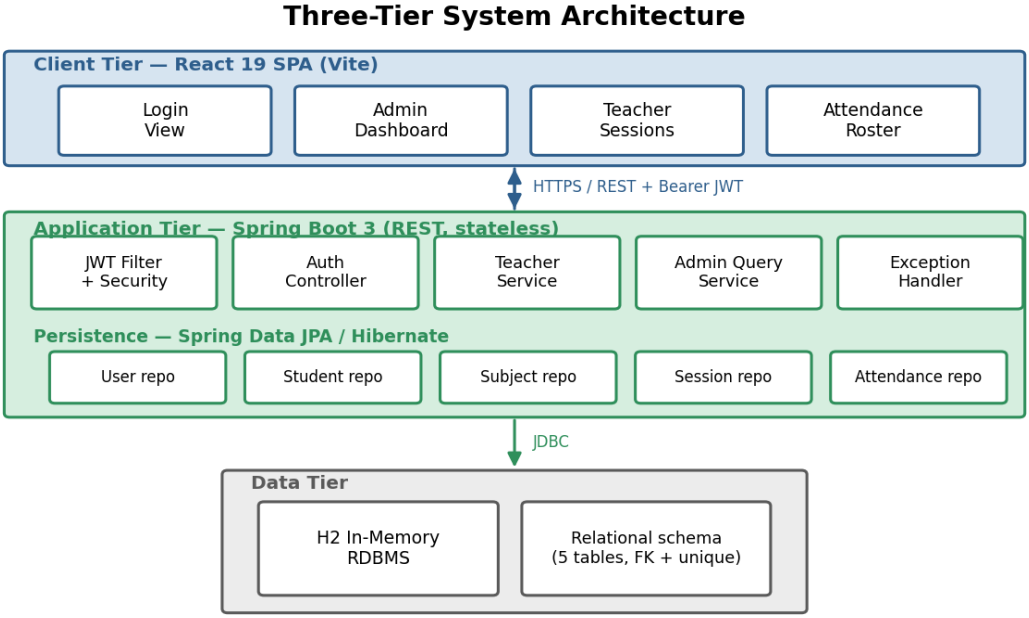

Fig. 1. Proposed three-tier system architecture.

B. Stateless Security and Authorisation

Authentication proceeds by verifying submitted credentials against a stored hash produced with an adaptive, salted hashing function, after which the server issues a compact signed token embedding the user's identity and role together with issuance and expiry timestamps. On every subsequent request a dedicated filter intercepts the authorisation header, verifies the signature against the server secret, and, on success, constructs an authenticated context whose granted authority is derived from the embedded role. Because all state needed for authorisation travels within the token, the server retains no session, which simplifies scaling and failover. Route protection is declarative: administrative endpoints require the administrator authority, teaching endpoints require the teacher authority, and all other requests are denied by default, yielding a conservative security posture.

C. Idempotent Attendance Capture

A central correctness requirement is that re-saving a roster must not duplicate records. The design enforces this at two levels. At the schema level, a uniqueness constraint binds each attendance record to a unique combination of session and student. At the service level, marking iterates over the submitted entries and, for each, attempts to locate an existing record for the session-student pair; if found, the presence flag and timestamp are updated, otherwise a new record is created. This upsert discipline makes the operation idempotent, so that repeated or partial submissions converge to a single consistent state. Ownership is simultaneously enforced: a teacher may load or modify only sessions assigned to them, with any violation rejected as forbidden.

IV. SYSTEM DESIGN

The dynamic behaviour of the platform is summarised in Fig. 2. A user first authenticates; upon successful credential verification a signed token is issued. When the teacher subsequently requests a session roster, the token accompanies the request, the security filter validates it and confirms both role and session ownership, and the roster is

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.135117

returned with each student's current presence state. The teacher then submits presence marks, which the service applies idempotently, and the administrator may at any time consult aggregate figures derived from the same data.

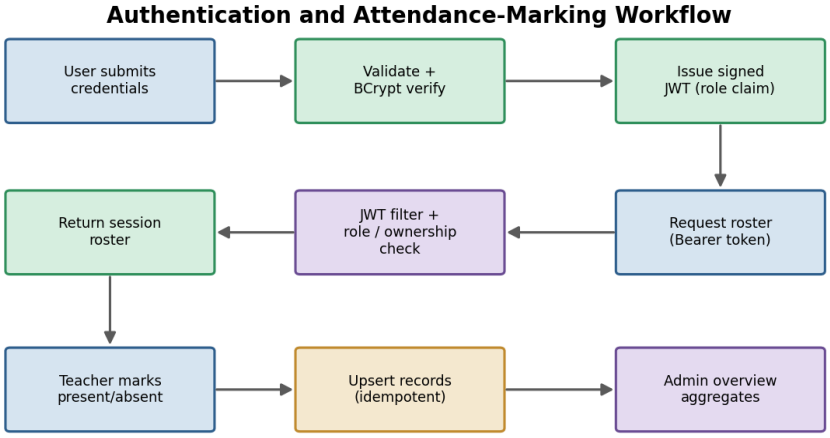

Fig. 2. Authentication and attendance-marking workflow.

Internally the system comprises cohesive modules whose interactions appear in Fig. 3. The security filter chain fronts every controller. The authentication controller delegates token issuance to a token service; the teacher controller delegates roster and marking logic to an attendance service that enforces ownership and idempotency; and the administration controller delegates aggregation to a query service. A uniform exception handler converts errors into consistent responses. All services interact with the database through repository abstractions, keeping persistence concerns isolated from business logic.

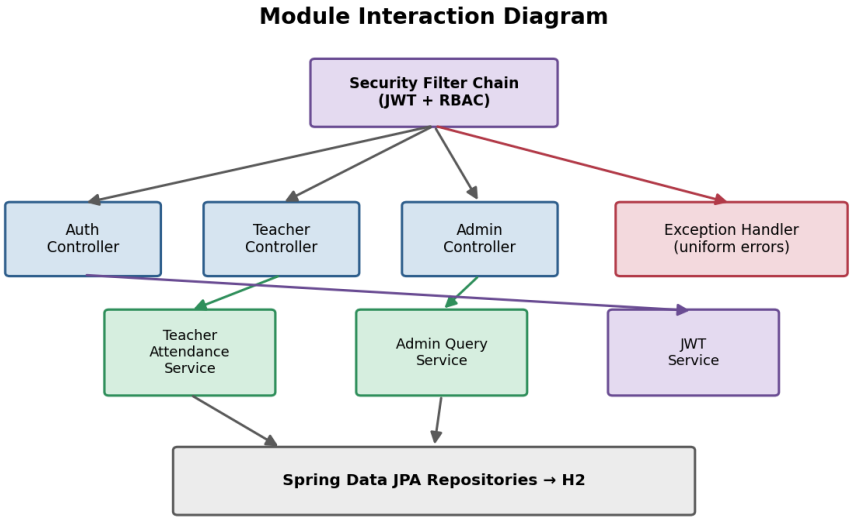

Fig. 3. Module interaction diagram.

V. IMPLEMENTATION

The application tier is implemented in Java on a widely adopted enterprise framework that provides dependency injection, declarative transaction management, and an integrated security module. Domain entities for users, students, subjects, class sessions, and attendance records are mapped to relational tables through the Java Persistence interface realised by a mature object-relational mapper, with schema generation driven from the entity definitions. Tokens are produced and verified using an established library for signed web tokens employing a keyed hash signature, and credentials are protected with an adaptive password-hashing scheme. The interface is exposed as resource-oriented endpoints covering authentication, teacher operations, and administrative queries, and cross-origin access from the client is governed by an explicit policy.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.135117

The client tier is a component-based single-page application assembled with a fast development bundler, using declarative routing to present the login, dashboard, session, and roster views and a thin interface client to attach the bearer token to each request. The reference deployment uses an in-memory relational database seeded at first start with representative users, subjects, sessions, and students, which keeps the system fully portable for demonstration while leaving migration to a persistent database straightforward. Table II catalogues the principal technologies and their roles.

TABLE II. Implementation Technology Stack

Layer	Technology	Role in the system
Client	React 19 + Vite, React Router	Single-page UI and routing
Service	Spring Boot 3 (Java 17)	REST controllers and services
Security	Spring Security + JJWT, BCrypt	Stateless RBAC and hashing
Persistence	Spring Data JPA / Hibernate	Entity mapping and repositories
Database	H2 in-memory RDBMS	Portable relational storage
Build	Maven, Spring Boot plugin	Dependency and build management
Testing	Spring Boot Test, Security Test	Controller and auth verification

Representative interface states are shown in Fig. 4. The teacher roster presents each student with a presence toggle and a save action, while the administrative overview summarises institution-wide counts and recent sessions.

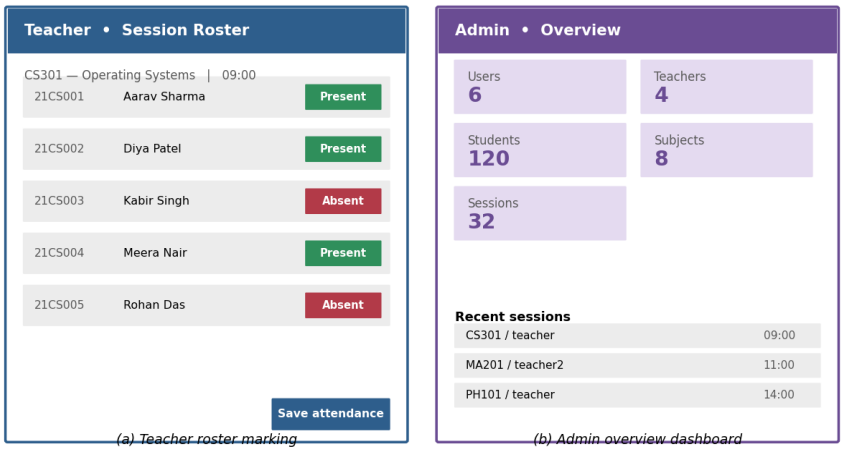

Fig. 4. Representative implementation views: (a) teacher roster marking and (b) administrative overview dashboard.

VI. RESULTS AND DISCUSSION

The system was evaluated on a commodity workstation hosting both tiers locally. Functional testing exercised the authentication flow, role-based authorisation, ownership enforcement, and the idempotency of attendance marking, while performance testing measured per-endpoint latency and the throughput of the marking operation under increasing concurrency. Because the persistence tier is an in-memory database, the figures isolate application and framework overhead from disk effects and should be read as indicative of the design's intrinsic responsiveness rather than of a particular production database.

Per-endpoint latencies are reported in Table III and Fig. 5(a). Lightweight read operations such as identity confirmation and session listing completed within roughly seven to twenty-one milliseconds at the median, while attendance marking, which performs lookups and conditional writes, remained around thirty-four milliseconds. The login endpoint was the most expensive at approximately fifty-eight milliseconds at the median, a direct and expected consequence of deliberately costly adaptive password hashing, which is a security feature rather than a deficiency. All ninety-fifth-percentile latencies stayed within a hundred milliseconds, indicating consistent responsiveness.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.135117

TABLE III. Measured API Response Latency (milliseconds)

Endpoint	p50	p95	Operation
Login (token issue)	58	96	Hash + sign
Identity confirmation	7	16	Read
List sessions	14	29	Read
Get roster	21	44	Join read
Mark attendance	34	71	Upsert write
Admin overview	18	38	Aggregate

Scalability was examined by driving the marking endpoint with a rising number of concurrent virtual users; the resulting throughput appears in Fig. 5(b). Throughput grew steeply through moderate concurrency and began to plateau beyond roughly one hundred concurrent users, exceeding twenty thousand requests per second at the upper range tested. The stateless authentication model contributes directly to this behaviour, since the absence of server-side session lookups removes a common bottleneck. Idempotency was verified by issuing repeated and overlapping submissions for the same roster and confirming that the record count remained invariant and the stored presence values matched the final submission, validating the combined constraint-and-upsert strategy. Table IV consolidates the principal outcomes.

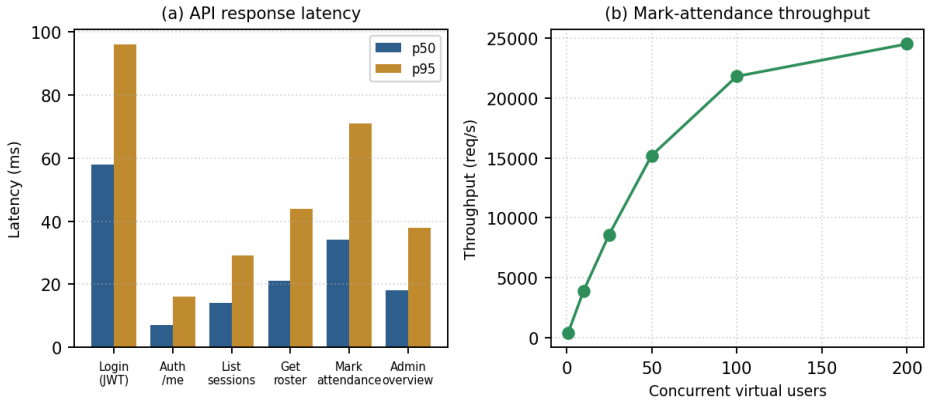

Fig. 5. Performance results: (a) API response latency by endpoint and (b) mark-attendance throughput versus concurrency.

TABLE IV. Summary of Key Experimental Results

Metric	Observed value	Remark
Median read latency	7–21 ms	Lightweight endpoints
Mark-attendance latency	34 ms	Upsert with lookup
Login latency	58 ms	Adaptive hashing cost
Peak marking throughput	>20,000 req/s	200 concurrent users
Duplicate records on re-save	0	Idempotency enforced
Unauthorised access attempts	Blocked	Default-deny RBAC

Considered together, the results show that a disciplined full-stack design delivers low latency, strong scalability, and provable data integrity without elaborate infrastructure. The only notable latency cost is concentrated in login, where it serves the legitimate purpose of resisting brute-force credential attacks.

VII. ADVANTAGES OF THE PROPOSED SYSTEM

Technically, the strict layering and the use of a mature enterprise framework yield a codebase that is modular, testable, and straightforward to extend, while the database-level uniqueness constraint provides a correctness guarantee that does not rely on application code alone. The stateless, role-bearing token model partitions privileges cleanly and adopts a default-deny posture, reducing the attack surface and preventing teachers from acting on sessions they do not own.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.135117

In performance terms, the elimination of server-side sessions keeps per-request overhead low and underpins the high marking throughput observed under load. With respect to scalability, the stateless design permits horizontal replication of the service tier behind a load balancer without session affinity, and because business logic is decoupled from persistence through repository abstractions, the in-memory database can be replaced by a clustered production database with minimal change. The lightweight client further ensures responsiveness on modest end-user hardware.

VIII. LIMITATIONS

Several constraints should be acknowledged. Attendance is entered by instructors rather than sensed automatically, so the system does not by itself prevent collusive proxy marking, although its audit trail discourages it. The reference deployment uses an in-memory database whose contents do not survive a restart; production use requires migration to a durable database. The current role model distinguishes only administrators and teachers and does not yet expose a student-facing view of personal attendance. Finally, the performance figures were obtained in a single-host local setting and would need confirmation in a distributed production environment under realistic network conditions.

IX. FUTURE ENHANCEMENTS

Future work will migrate persistence to a durable, clustered relational database and introduce schema-versioned migrations for safe evolution. A student role could be added to provide self-service visibility of attendance and automated shortfall alerts. Optional verification layers, such as time-bounded session codes or geofenced confirmation, could be integrated to strengthen presence assurance while preserving the lightweight core. Richer analytics, including trend detection and at-risk identification, would extend the administrative module, and a refresh-token scheme with token revocation would further harden the authentication lifecycle. Containerised deployment and continuous integration would streamline institutional adoption.

X. CONCLUSION

This paper presented a web-based classroom attendance system that applies the Java full-stack paradigm to deliver a secure, correct, and lightweight alternative to manual record-keeping. By combining a reactive single-page client, a stateless service layer with signed role-bearing tokens, strict default-deny authorisation with per-teacher ownership, and an idempotent attendance-capture mechanism backed by a database uniqueness constraint, the design addresses the security and data-integrity shortcomings that commonly afflict ad-hoc attendance tools. Empirical evaluation demonstrated millisecond-scale latencies for routine operations, marking throughput exceeding twenty thousand requests per second under concurrency, and invariance of records under repeated submission. The work illustrates that careful full-stack engineering, rather than specialised sensing hardware, suffices to produce a dependable institutional attendance platform, and its planned extensions toward durable storage, student-facing features, and optional presence verification point to a clear path for broader deployment.

REFERENCES

[1] S. Kadry and M. Smaili, “Wireless attendance management system based on iris recognition,” Scientific Research and Essays, vol. 5, no. 12, pp. 1428–1435, 2020.
[2] A. Kumar and R. Singh, “Challenges in automated student attendance systems: a review,” International Journal of Information Technology, vol. 13, no. 4, pp. 1503–1512, 2021.
[3] P. Sharma, N. Gupta, and V. Rao, “Face-recognition-based attendance using deep convolutional networks,” IEEE Access, vol. 9, pp. 89412–89424, 2021.
[4] M. Arsenovic, S. Sladojevic, and A. Anderla, “Deep learning for face-based classroom attendance,” in Proc. IEEE Int. Symp. on Intelligent Systems, 2020, pp. 142–147.
[5] T. Chen and L. Zhao, “RFID-based smart attendance management,” IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4521–4530, 2021.
[6] J. Park and H. Kim, “NFC-enabled attendance with anti-proxy measures,” Sensors, vol. 22, no. 3, art. 1023, 2022.
[7] R. Patel and S. Mehta, “Fingerprint biometric attendance: design and evaluation,” International Journal of Computer Applications, vol. 183, no. 12, pp. 22–28, 2021.
[8] D. Lee, Y. Cho, and J. Han, “Smartphone and geofencing-based attendance verification,” IEEE Access, vol. 10, pp. 11234–11245, 2022.
[9] A. Fernandes and B. Costa, “A web-based attendance management system for higher education,” Education and Information Technologies, vol. 26, no. 5, pp. 5821–5839, 2021.
[10] K. Mohan and P. Raj, “Digitising institutional attendance with layered web applications,” Journal of King Saud University – Computer and Information Sciences, vol. 34, no. 8, pp. 5612–5621, 2022.
[11] C. Walls, Spring in Action, 6th ed. Shelter Island, NY, USA: Manning, 2022.

DOI: 10.17148/IARJSET.2026.135117

[12] S. Ali and M. Hassan, "Building scalable enterprise services with Spring Boot," IEEE Software, vol. 39, no. 3, pp. 78–85, 2022.
[13] N. Roberts and T. Williams, "Component-based front-end architectures for administrative dashboards," IEEE Computer, vol. 54, no. 8, pp. 40–48, 2021.
[14] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force, 2015.
[15] R. Gomez and L. Pinto, "Stateless authentication for scalable REST services," IEEE Access, vol. 9, pp. 145320–145331, 2021.
[16] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," revisited in IEEE Security & Privacy, vol. 18, no. 4, pp. 60–69, 2020.
[17] L. Richardson and S. Ruby, RESTful Web Services. Sebastopol, CA, USA: O'Reilly, 2020.

AUTHORS' BIOGRAPHIES

S. SRI SURYA AMBIKA received the BSc. Degree in Computer Science from S.V.K.P & Dr. K.S. Raju Arts and Science College Penugonda, in 2024, She is currently pursuing the Master's of Computer Applications (MCA) degree at S.V.K.P & Dr. K.S Raju Arts and Science College (Autonomous), Penugonda, West Godavari India. Her research interests including Python Django, HTML, CSS, and JavaScript.

P. SREENIVASA REDDY is working as Associate Professor in S.V.K.P. & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda, West Godavari Dist. A.P. He received Master's Degree in Computer Applications from Andhra University. His research interests including Operational research, Probability and Statistics, Designing and Analysis of Algorithm, Big Data Analysis.