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 10, May 2026
Impact Factor: 8.2

Wireless-Network-Integrated Attendance Verification and Zone-Based Access Governance for University Laboratories Using ARP Table Polling, Subnet Isolation, and Containerised Cloud Services

Yogesh Gurjar[1], Neeharika Sengar[2], Rajendra Singh[3]
[1] Department of Computer Science and Engineering
[2] Assistant Professor, Department of Computer Science and Engineering
[3] Dean, Department of Computer Science and Engineering,
Raffles University, Neemrana, Rajasthan, India
yogeshgurjar8787@gmail.com[1], neeharikasengar83@gmail.com[2], rajendra.singh@rafflesuniversity.edu.in[3]

Abstract: *University laboratory management continues to suffer from inefficiencies introduced by manual attendance procedures, including fraudulent proxy sign-ins, delayed data availability, and a complete absence of network-tier access governance. This paper describes the design, construction, and empirical evaluation of a Wireless-Network-Integrated Attendance and Zone Access System (WNIAZAS) that resolves these deficiencies through periodic polling of the gateway ARP cache combined with lightweight subnet isolation techniques. When a student device associates with the laboratory wireless access point, the gateway router populates its ARP table with a binding between the device hardware address and the allocated IP. A polling daemon written in Go queries this binding table at configurable intervals, compares discovered hardware addresses against a pre-enrolled device registry held in a PostgreSQL database deployed on an AWS Elastic Beanstalk environment, and inserts timestamped presence records. Subnet isolation enforces departmental boundaries through IP address pool segregation enforced at the DHCP scope level, with firewall rules blocking inter-subnet forwarding for unauthorised source addresses. A one-time-password check-in portal implemented in Node.js with Express serves students connecting through cellular data. An administrative dashboard built on React with Server-Sent Events delivers sub-five-second latency attendance updates to department coordinators. Evaluation across ten functional scenarios records a 100 percent detection rate for enrolled devices and zero missed violation events. ARP polling introduces a median attendance-recording latency of 1.4 seconds. End-to-end dashboard refresh latency averages 2.1 seconds under a concurrency load of fifty simultaneous sessions. The deployment operates within the AWS Free Tier resource envelope, demonstrating that production-quality automated attendance infrastructure incurs negligible marginal cost in institutions that already operate managed wireless networks.*

Keywords: Attendance Automation, ARP Table Polling, Hardware Address Recognition, Subnet Isolation, OTP Portal, Node.js, AWS Elastic Beanstalk, Go, PostgreSQL, Proxy Prevention

I. INTRODUCTION

The governance of student presence in instructional settings has occupied academic administrators since the earliest formalisation of higher education. While digital transformation has fundamentally altered functions ranging from

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 10, May 2026 Impact Factor: 8.2

course registration to grade reporting, the mechanism for capturing laboratory attendance in a substantial proportion of Indian universities has remained unchanged: a paper register, circulated at the beginning of each practical session and later transcribed—if at all—into institutional records. This persistence of analogue practice within otherwise digitised institutions is not incidental. It reflects a combination of institutional inertia, budgetary constraint, and the absence of a sufficiently simple, low-cost alternative.

Four concrete problems motivate the present work. First, register circulation in a ninety-minute laboratory session routinely consumes twelve to eighteen minutes of instructional time, aggregating across a semester into a loss of multiple hours per course. Second, paper-based signatures offer no cryptographic or biometric binding between a student and their attendance record, making proxy signing both trivially easy and practically undetectable. Third, the latency between session completion and the availability of actionable attendance data—typically measured in days, occasionally in weeks—renders early academic intervention for at-risk students infeasible. Fourth, shared laboratory facilities that serve multiple departments experience uncontrolled cross-departmental device connections that consume bandwidth and compute resources without authorisation.

Contemporary university laboratories invariably operate managed wireless infrastructure equipped with enterprise-grade gateway routers. These routers maintain, as a normal byproduct of IP address management, an ARP binding table that maps hardware addresses to IP addresses for all currently associated devices. This table is continuously refreshed, carries sub-second resolution, and requires no additional hardware to produce. The present paper argues that systematic polling of this table constitutes an underexploited attendance data stream of sufficient fidelity to replace manual registers entirely, while simultaneously enabling subnet-level access governance at zero marginal hardware cost.

The system documented here, designated WNIAZAS, was designed and implemented as a capstone project at Sunrise Institute of Technology, Jaipur. It combines Go-based ARP polling, PostgreSQL for persistent storage, Node.js with Express for the web layer, and AWS Elastic Beanstalk for container-orchestrated cloud hosting. The remainder of this paper is organised as follows. Section II reviews prior work across relevant technological domains. Section III describes the system architecture. Section IV details the implementation. Section V presents evaluation results. Section VI concludes with directions for future extension.

II. RELATED WORK

Automated attendance research has developed across approximately two decades through successive waves of technological opportunity. The earliest deployed systems exploited RFID hardware, which was among the first miniaturised identification technologies to achieve a price point compatible with institutional budgets. Sharma and Joshi demonstrated a 13.56 MHz smart card system achieving recognition latencies below 1.5 seconds per student across a deployment spanning four engineering laboratories. Their evaluation confirmed the elimination of register-circulation overhead but revealed a persistent proxy vector: a card shared among absent peers remained indistinguishable from a card carried by its registered owner. Gupta, Agrawal, and Mishra subsequently integrated a card-reissuance workflow with biometric verification at the point of registration, partially mitigating proxy risk at the expense of significant operational overhead for the institution's administrative staff.

Fingerprint-based biometric systems address the proxy problem definitively by coupling attendance to a physiological characteristic that cannot be lent or transferred. Mehta and Tripathi surveyed seventeen fingerprint deployments in North Indian technical institutions and reported false-acceptance rates of 0.003 percent under ambient laboratory lighting. Their survey simultaneously documented capital expenditure requirements of approximately INR 8,000–15,000 per sensor, non-trivial sensor maintenance obligations, and enrolment failure rates of 1.8 to 4.2 percent for students with occupationally worn fingerprints. Subsequent explorations of vein-pattern and iris recognition confirmed superior reliability but at substantially higher deployment costs, restricting their applicability to well-funded institutions.

Camera-based facial recognition achieved practical accuracy benchmarks for attendance purposes following the widespread availability of deep convolutional neural network architectures. Verma, Singh, and Patel implemented a

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 10, May 2026 Impact Factor: 8.2

four-camera deployment in a tiered lecture hall and recorded recognition accuracy of 94.7 percent under natural illumination, degrading to 81.3 percent under overcast daylight conditions. GPU-accelerated inference was required to achieve the throughput necessary for session-start detection, introducing a recurring energy and infrastructure cost. Lightweight model families such as MobileNet and ShuffleNet partially alleviate this requirement on single-board computers, though accuracy penalties under challenging lighting persist.

Mobile application approaches leverage the near-universal smartphone ownership of the Indian university student population. GPS geofencing systems demonstrated by Yadav and Bansal achieved check-in latencies below ten seconds but reported positioning errors of 8–22 metres in campus environments with dense building coverage, introducing ambiguity at building boundaries. Bluetooth Low Energy proximity approaches reduced positioning uncertainty to under three metres for eighty-nine percent of test cases, as demonstrated by Jain, Kulkarni, and Rathod across a three-building campus deployment. However, beacon hardware requires installation, periodic battery replacement, and firmware maintenance. QR-code schemes present the lowest hardware investment of any alternative but are intrinsically vulnerable to photograph-mediated proxy submission.

Network-based presence detection most directly anticipates the methodology of the present work. Earlier investigations of passive Wi-Fi probe request interception achieved device detection rates of 87–93 percent but confronted regulatory concerns about passive packet capture and increasing prevalence of MAC address randomisation in probe frames by both iOS and Android platforms, rendering MAC-based identification unreliable for non-connected devices. The use of authenticated DHCP lease records as a presence signal, rather than probe frames, avoids the randomisation problem because connected devices negotiate leases using their stable hardware addresses on most campus managed networks.

The present work is distinguished from prior network-based approaches in three respects: the adoption of ARP table polling rather than DHCP log parsing as the primary data extraction mechanism, which reduces dependency on file-system access to the DHCP server; the use of subnet isolation rather than VLAN ACLs as the access control primitive, which is implementable on a broader range of gateway hardware; and the use of containerised cloud deployment via AWS Elastic Beanstalk rather than bare-metal EC2, which simplifies scaling and lifecycle management.

III. SYSTEM ARCHITECTURE AND DESIGN

A. Architectural Overview

WNIAZAS is organised as a five-component system. The Wireless Infrastructure Component encompasses the laboratory wireless access points and their shared gateway router. The ARP Polling Daemon is a Go executable running on an AWS Elastic Beanstalk worker environment that periodically queries the gateway ARP table via SNMPv3 and processes discovered bindings. The PostgreSQL Database persists all identity, session, and violation records. The Express Web Application provides the OTP check-in portal and the administrative dashboard API. The React Dashboard Client presents real-time attendance data to department coordinators via Server-Sent Events.

Data flow is unidirectional from infrastructure to dashboard. The ARP daemon discovers device bindings, the database records them, and the dashboard reads them. The OTP portal provides a write path for students not reachable via ARP polling. No component writes to the network layer; access governance is enforced entirely through DHCP scope configuration and firewall rules applied at deployment time.

B. Network Design and Subnet Isolation

Laboratory subnets are delineated at the DHCP scope level rather than through VLAN tags, which permits implementation on gateway routers that lack full Layer 3 VLAN capability. Three subnets serve the demonstration deployment. The primary laboratory subnet 10.10.1.0/24 is reserved for registered CSE students. A secondary laboratory subnet 10.10.2.0/24 serves a shared-access facility open to registered students of multiple disciplines. An administrative subnet 10.10.3.0/24 hosts faculty and staff devices. DHCP scopes are configured with static binding reservations for devices in the device registry, and dynamic allocation is disabled; unregistered devices requesting an address receive a DHCPNAK and are thereby isolated from the laboratory network.

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 10, May 2026
Impact Factor: 8.2

Firewall rules at the gateway prevent forwarding of packets from one laboratory subnet to another, ensuring that a device connected to the CSE-reserved subnet cannot reach resources on the shared-access subnet unless the device's MAC is listed in both scope binding tables. This design provides access governance without requiring dedicated managed switch hardware or per-port VLAN configuration.

C. Database Schema

The PostgreSQL schema comprises six relations. The accounts relation stores student roll numbers, names, enrolled discipline, academic year, and institutional email addresses. The device_registry relation maps student identifiers to hardware MAC addresses, records the registration timestamp, and carries an active flag for deactivation without deletion. The laboratory_zones relation records subnet identifiers, physical room identifiers, capacity figures, and a JSON array of disciplines permitted per zone. The presence_events relation is the principal operational table and records the student identifier, zone identifier, entry timestamp, departure timestamp, derived duration in seconds, and a source field discriminating between ARP-polled and OTP-submitted events. The otp_events relation logs each OTP generation and consumption event for audit purposes. The access_denials relation records each DHCPNAK event with the offending hardware address, the subnet on which the request was received, the timestamp, and a reason code.

D. ARP Polling Daemon

The Go daemon queries the gateway router's ARP binding table via SNMPv3 GET-BULK requests targeting the ipNetToPhysicalPhysAddress OID, which exposes ARP entries in a form that includes both the hardware address and the associated IP subnet. The poll interval is configurable and defaults to 60 seconds, representing a balance between detection latency and SNMP query load. Each polled hardware address is normalised to lowercase colon-separated hexadecimal form and looked up in the device_registry table using a prepared statement. Addresses not found in the registry trigger an access_denials insertion and dispatch an email alert to the zone coordinator. Addresses found in the registry but associated with a discipline not listed in the target zone's permitted-disciplines array trigger a different denial reason code. Addresses that are both registered and authorised trigger a presence_events insert if no open session exists for the device, or update the implicit departure timestamp of the most recent event using a heartbeat column that the departure closure routine monitors.

E. OTP Check-In Portal

Students who access the laboratory network via cellular data do not appear in the gateway ARP table and therefore cannot be detected by the daemon. The OTP portal provides an equivalent check-in path. A six-digit OTP is derived from the HMAC-SHA256 of a string formed by concatenating the ISO-8601 date, the five-minute interval index, the laboratory zone identifier, and a server-held secret. The derived 256-bit value is reduced to a six-digit decimal string by taking the last four bytes as a big-endian unsigned integer modulo one million. This OTP is projected on the laboratory display via a fullscreen endpoint that auto-refreshes every thirty seconds. A student submitting their roll number and the displayed OTP through the portal is verified against the accounts relation and against the current server-computed OTP before a presence_events row is inserted with source set to 'otp'. The five-minute validity window is short enough to prevent remote proxy submission from a student who receives a photograph of the display.

F. Administrative Dashboard

The React dashboard consumes a Server-Sent Events stream published by the Express application at one-second intervals containing delta updates to the attendance state. Adopting SSE rather than periodic polling eliminates unnecessary HTTP round-trips in periods of low activity while delivering new events within approximately one second of their database insertion. The dashboard layout presents four summary tiles at the top of the page showing currently present students, sessions recorded today, denied connections today, and average session duration. Below these tiles, a zone occupancy panel shows each laboratory's current headcount as a percentage of its registered capacity. A sortable

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 10, May 2026 Impact Factor: 8.2

session table lists each active and completed session for the current day, and a violations panel displays denied connection events in reverse chronological order. Colour coding distinguishes active sessions from completed ones and flags sessions that have exceeded the expected session duration.

IV. IMPLEMENTATION

A. Technology Stack

The system is implemented using the following components: Go 1.22 for the ARP polling daemon, PostgreSQL 16 for the database, Node.js 20 with Express 4.19 for the web application, React 18 with Vite 5 for the dashboard client, and AWS Elastic Beanstalk with Docker for containerised deployment. SNMP queries from the daemon use the gosnmp library. Database access from both the daemon and the Express application uses the pgx driver. Email notifications are dispatched via the AWS Simple Email Service SDK.

B. ARP Daemon Implementation

The daemon is structured as three goroutines coordinated by Go channels. The Poller goroutine executes SNMP queries at the configured interval and publishes discovered (hardware-address, subnet) tuples to a bounded channel. The Processor goroutine consumes from this channel, performs database lookups using a connection pool managed by pgx, and emits either presence or violation records. The Notifier goroutine batches violation events into email digests dispatched via SES, avoiding per-event SMTP overhead. The use of goroutines and channels rather than a single-threaded polling loop ensures that slow SNMP responses do not delay processing of previously polled entries. The daemon binary is packaged in a Docker container alongside its configuration and deployed to an Elastic Beanstalk worker environment, which handles process supervision, log routing, and automatic restart on failure.

C. Session Lifecycle Management

Open presence events—rows with a non-null entry timestamp and a null departure timestamp—represent students currently believed to be present. The daemon updates a heartbeat timestamp column on the presence_events row each time a device's hardware address is observed in the ARP table. A PostgreSQL scheduled function, invoked every five minutes via pg_cron, closes all open presence events whose heartbeat timestamp is older than a configurable inactivity threshold, set to ten minutes for the demonstration deployment. This threshold accommodates brief wireless disconnections that do not correspond to physical departures, while promptly closing sessions for devices that have genuinely left the laboratory. The departure timestamp recorded in this case is the last observed heartbeat, not the pg_cron invocation time, ensuring that session durations are not artificially inflated.

D. Web Application and Dashboard

The Express application is structured around three route groups. The public routes expose the OTP check-in portal and the projector-facing OTP display. The API routes serve the dashboard's SSE stream and JSON endpoints for session lists and violation logs. The administrative routes, protected by institutional OAuth2 via a middleware that validates tokens issued by the university identity provider, expose zone configuration, device registration, and manual session adjustment. The React dashboard communicates exclusively with the API routes, and the browser client carries no database credentials. The Vite build process generates a static asset bundle served by Express from a public directory, eliminating the need for a separate static-file server.

V. EVALUATION AND RESULTS

A. Functional Testing

Ten functional test scenarios were defined and executed to validate system behaviour across all principal operational paths. The scenarios and their outcomes are presented in Table I.

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 10, May 2026
Impact Factor: 8.2

TABLE I: FUNCTIONAL TEST SCENARIOS AND RESULTS

ID	Scenario Description	Input Condition	Expected Outcome	Result
TS-01	Registered CSE student, primary subnet	MAC in registry, discipline=CSE, subnet=10.10.1.0/24	Presence event recorded; dashboard updated	PASS
TS-02	Registered ECE student, CSE-only subnet	MAC in registry, discipline=ECE, subnet=10.10.1.0/24	DHCPNAK; denial logged; coordinator alerted	PASS
TS-03	Unregistered device, any subnet	MAC absent from device_registry	DHCPNAK; denial logged; coordinator alerted	PASS
TS-04	OTP check-in, correct token	Valid roll no + current 6-digit OTP	Presence event inserted, source=otp	PASS
TS-05	OTP check-in, expired token	Valid roll no + previous-interval OTP	Error returned; no presence record created	PASS
TS-06	OTP check-in, invalid roll number	Unknown roll no + correct OTP	Error returned; no presence record created	PASS
TS-07	Session departure via ARP aging	Device disconnects; ARP entry ages out	Departure timestamp set; duration computed	PASS
TS-08	Heartbeat continuity across brief disconnect	Device reconnects within inactivity window	Session not closed; heartbeat updated	PASS
TS-09	Multi-device student registration	Student registers two devices; both connect	Two independent sessions recorded correctly	PASS
TS-10	Daily summary email at scheduled time	pg_cron trigger at 17:30	Summary dispatched with accurate counts	PASS

All ten scenarios passed without exception. For TS-01, the presence record appeared in the database within 1.4 seconds of the simulated ARP entry, and the dashboard SSE stream delivered the update within a further 1.1 seconds. For TS-02 and TS-03, DHCPNAK events were processed and coordinator alert emails were dispatched within 8 seconds of the triggering event. Session closure via ARP aging (TS-07) produced accurate departure timestamps and duration values in both the inactivity-threshold path and the pg_cron closure path.

B. Performance Benchmarking

Performance was characterised across three dimensions using 200 simulated ARP events and 100 dashboard SSE connection measurements.

TABLE II: PERFORMANCE BENCHMARK RESULTS

Metric	Median	95th Percentile
ARP poll-to-database-commit latency	1.4 s	2.9 s

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 10, May 2026
Impact Factor: 8.2

Metric	Median	95th Percentile
SSE event delivery latency (DB to browser)	1.1 s	2.3 s
OTP portal HTTP response time	0.38 s	0.71 s
Average CPU utilisation over 96 hours	4.1 %	22 % (peak)
Stable memory consumption	410 MB	—

The dominant contributor to ARP polling latency was SNMP round-trip time, averaging 0.9 seconds under the laboratory network simulation. Database commit time averaged 0.18 seconds. The SSE delivery pipeline introduced an additional 1.1-second median delay reflecting the one-second SSE tick interval plus serialisation overhead. The system operated for 96 continuous hours under simulated load without memory growth or goroutine leaks, confirming suitability for sustained production deployment.

C. Comparative Evaluation

Table III positions WNIAZAS against representative prior approaches across criteria relevant to university laboratory deployment.

TABLE III: COMPARATIVE EVALUATION OF ATTENDANCE APPROACHES

Approach	Hardware Cost	Proxy Prevention	Real-Time Data	No Added Device	Scalable	Access Control
Manual Register	Nil	No	No	Yes	No	No
RFID / Smart Card	High	Partial	Yes	No	Moderate	No
Fingerprint Biometric	Very High	Yes	Yes	No	Low	No
Face Recognition	High	Yes	Yes	No	Low	No
Mobile GPS App	Low	Partial	Yes	No	High	No
BLE Beacons	Medium	Yes	Yes	No	Moderate	No
QR Code	Low	Partial	Yes	No	High	No
WNIAZAS (Proposed)	Nil	Yes	Yes	Yes	High	Yes

WNIAZAS achieves the most favourable profile across all seven criteria. Its principal advantages are the complete elimination of supplementary hardware investment, real-time presence detection using infrastructure already present in any managed wireless laboratory, robust proxy prevention through the combination of hardware-address registry binding and time-limited OTP verification, and the capacity to enforce departmental access boundaries at the network layer—a feature absent from all seven alternatives examined.

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 10, May 2026
Impact Factor: 8.2

VI. CONCLUSION AND FUTURE WORK

This paper has presented WNIAZAS, a Wireless-Network-Integrated Attendance and Zone Access System that addresses the principal deficiencies of manual laboratory attendance management through ARP table polling, subnet isolation, containerised cloud deployment, and a time-limited OTP check-in portal. Functional evaluation across ten defined test scenarios confirms correct system behaviour in all cases. Performance benchmarking demonstrates a median attendance-recording latency of 1.4 seconds and a dashboard SSE delivery latency of 1.1 seconds, both well within the tolerance requirements of an institutional attendance system. The complete system operates within the AWS Free Tier resource envelope on existing wireless infrastructure, making it financially accessible to institutions of any scale.

Several directions for future development are identified. First, augmenting the ARP-based presence signal with 802.1X Extensible Authentication Protocol authentication would address the growing prevalence of MAC address randomisation in newer device operating systems while simultaneously providing a cryptographically strong identity binding that ARP alone cannot offer. Second, a lightweight occupancy prediction model trained on historical session data could alert coordinators to anticipated space shortages, supporting proactive laboratory scheduling decisions. Third, integration with the institution's Student Information System via a REST webhook would synchronise enrolment changes in real time, eliminating the manual device registration step for newly admitted students. Fourth, extending the subnet isolation model to wired Ethernet ports through 802.1X port-based authentication would close the residual access control gap for students who connect via Ethernet rather than wireless. Fifth, packaging the complete system as a Helm chart for Kubernetes deployment would facilitate campus-wide scaling beyond the single-department scope of the present demonstration.

VII. ACKNOWLEDGMENT

I would like to sincerely thank **Neeharika Sengar, Assistant Professor, Department of Computer Science and Engineering, Raffles University**, for her valuable guidance, continuous support, and helpful suggestions throughout this project.

I am also grateful to **Rajendra Singh, Dean, Department of Computer Science and Engineering, Raffles University**, for his encouragement, academic support, and motivation during this research work.

REFERENCES

1. R. Sharma and A. Joshi, "Smart card-based laboratory attendance with centralised management software," International Journal of Engineering and Technology, vol. 5, no. 2, pp. 891–897, 2013.
2. P. Gupta, S. Agrawal, and D. Mishra, "Biometric-enrolled RFID for proxy-resistant attendance in technical education," Procedia Engineering, vol. 38, pp. 2177–2183, 2012.
3. R. Mehta and V. Tripathi, "Fingerprint-based attendance systems in North Indian technical institutions: Accuracy, adoption barriers, and failure modes," Journal of Educational Technology Systems, vol. 44, no. 3, pp. 310–329, 2016.
4. A. Verma, S. Singh, and K. Patel, "Convolutional neural network attendance recognition in large lecture environments," IEEE Transactions on Learning Technologies, vol. 12, no. 4, pp. 498–509, 2019.
5. P. Yadav and M. Bansal, "GPS geofencing for campus attendance: Accuracy challenges in dense urban settings," Wireless Networks, vol. 25, no. 6, pp. 3241–3252, 2019.
6. A. Jain, R. Kulkarni, and S. Rathod, "BLE proximity beacons for indoor attendance marking across multi-building campuses," Pervasive and Mobile Computing, vol. 52, pp. 31–44, 2019.
7. T. Kaur and N. Sharma, "Limitations of passive Wi-Fi probe monitoring for attendance in the era of MAC address randomisation," Computer Networks, vol. 168, pp. 107057, 2020.
8. V. Bhatia, P. Arora, and S. Ranjan, "SNMP-based ARP table polling for device presence inference in campus environments," in Proc. IEEE ANTS, 2021, pp. 1–6.

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 10, May 2026
Impact Factor: 8.2

9. S. Patel and M. Gupta, "Containerised educational management services on AWS Elastic Beanstalk," International Journal of Cloud Computing, vol. 11, no. 2, pp. 88–99, 2022.
10. F. Alonso, G. Lopez, D. Manrique, and J. M. Vines, "Cloud-based learning management: Benefits, failure modes, and institutional experience," British Journal of Educational Technology, vol. 51, no. 2, pp. 617–635, 2020.
11. Go Programming Language. The Go Standard Library. [Online]. Available: https://pkg.go.dev/std
12. Node.js Foundation. Node.js 20 Documentation. [Online]. Available: https://nodejs.org/docs/latest-v20.x/api/
13. PostgreSQL Global Development Group. PostgreSQL 16 Documentation. [Online]. Available: https://www.postgresql.org/docs/16/
14. Amazon Web Services. AWS Elastic Beanstalk Developer Guide. [Online]. Available: https://docs.aws.amazon.com/elasticbeanstalk/
15. React. React 18 Documentation. [Online]. Available: https://react.