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 9, May 2026
Impact Factor: 8.2

Network-Driven Automated Attendance and Departmental Access Control for University Computer Laboratories Using DHCP Log Parsing, VLAN Segmentation, and Cloud Deployment

Purab Sachdeva[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
purabsachdeva7@gmail.com, neeharikasengar83@gmail.com
rajendra.singh@rafflesuniversity.edu.in

Abstract: *University computer laboratories continue to rely on paper-based attendance registers, a practice that introduces proxy fraud, administrative overhead, and an absence of real-time visibility. That paper present the design, implementations, and evaluations of a Smart Attendance and Lab Access System that eliminates these deficiencies through passive network monitoring. When a student's registered device joins the laboratory Wi-Fi, the DHCP server generates a lease record containing the device MAC address. A continuously executing Python agent intercepts these records, matches them against a pre-enrolled MAC registry stored in a MySQL database hosted on an Amazon Web Services EC2 instance, and writes a time-stamped session entry. Departmental access control is enforced at the network layer through VLAN segmentation and Access Control Lists configured on the laboratory router; cross-departmental connection attempts are denied, logged, and reported to the Head of Department by automated email. A time-bound session code portal built on Flask provides an equivalent check-in path for students using personal mobile hotspots. A browser-accessible live dashboard presents attendance metrics, occupancy statistics, and a chronological violation log, refreshing automatically every ten seconds. Functional testing across eight scenarios confirms complete automation of the attendance workflow, effective detection and blocking of unauthorized access, and timely alerting with zero missed events for enrolled devices. The system operates on existing Wi-Fi infrastructure without any additional hardware investment. Performance benchmarking yields a median event-processing latency of 0.31 seconds and a dashboard response time of 0.74 seconds on a free-tier cloud instance.*

Keywords: Attendance Automation, DHCP Log Parsing, MAC Address Recognition, VLAN, ACL, Flask, AWS EC2, Network Security, Python, Proxy Prevention.

I. INTRODUCTION

The obligation to maintain an accurate record of student attendance is among the oldest and most persistent administrative duties in higher education. Despite the transformation of nearly every other dimension of academic management through digital technology, the dominant mechanism for recording laboratory attendance in Indian universities remains a paper register circulated at the start of each session. This arrangement, while administratively simple, generates four compounding problems that collectively motivate the present work.

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 9, May 2026 Impact Factor: 8.2

The first problem is the consumption of instructional time. In a sixty-to-ninety-minute laboratory session, the circulation of a register and the calling of names may occupy ten to fifteen minutes. Aggregated across an academic semester, this represents a non-trivial loss of learning time. The second problem is proxy attendance—the fraudulent signing of a register on behalf of an absent peer. Paper records offer no detection mechanism, and the practice persists even where university regulations explicitly prohibit it. The third problem is the latency of reporting: manually compiled registers do not yield actionable attendance data until days or weeks after the session, by which time early academic intervention for at-risk students is no longer possible. The fourth problem, specific to shared laboratory computing environments, is the absence of any network-layer mechanism to restrict access to students of the authorized department.

Contemporary university computer laboratories are invariably equipped with managed Wi-Fi infrastructure. Each time a device connects to the network, the DHCP server generates a lease record containing the device hardware address, the assigned IP, the network segment, and a timestamp. These records are produced continuously and automatically, yet are universally ignored from an attendance perspective. This paper argues that DHCP lease events constitute a continuously generated, high-fidelity stream of device presence data that can be exploited to automate attendance recording entirely—and simultaneously enforce network-layer access control without deploying any additional hardware.

The system described here was conceived and implemented as a final-year B.Tech project at Raffles University, Neemrana. It integrates Python-based DHCP log parsing, MySQL for persistent storage, Cisco VLAN and ACL configuration for access control, Flask for the web layer, and AWS EC2 for cloud hosting. The content of this paper is organised as follows: Section II reviews related work; Section III describes the proposed architecture; Section IV presents implementation details; Section V reports evaluation results; Section VI concludes and identifies future directions.

II. RELATED WORK

Research on automated attendance spans approximately two decades and has progressed through several distinct technological phases. Early deployments assigned each student an RFID-enabled card; readers installed at room entrances logged the card identifier together with a timestamp. Bhatt and Patel demonstrated latencies below two seconds per student and confirmed the elimination of register-circulation overhead [1]. Kumar et al. extended this approach with a centralised web-accessible backend and automated daily reports [2]. However, RFID infrastructure requires a reader at every entry point, cards are susceptible to loss and damage, and the system verifies card presence rather than student presence, leaving card-sharing as a residual proxy vector. Patil and Bhosale explored ultra-high-frequency RFID to enable multi-card simultaneous reads but found that indoor multipath interference significantly degraded reliability [3].

Biometric systems address the proxy problem by binding identification to a physiological attribute. Jain et al. reviewed twenty-three fingerprint deployments in Indian institutions and reported false acceptance rates below 0.001 percent under controlled conditions [4]. However, their analysis also documented substantial capital requirements, maintenance overhead, hygiene objections heightened by post-pandemic sensitivities, and enrolment failure rates of one to three percent due to worn fingerprints. Camera-based facial recognition achieves recognition accuracy exceeding ninety-six percent under controlled illumination, as demonstrated by Zhang and Shen [5], but degrades markedly under variable lighting and demands GPU-equipped server hardware.

Mobile application approaches exploit the near-universal smartphone ownership among university students. GPS-based geofencing systems require no dedicated infrastructure but suffer from positioning inaccuracy of five to twenty metres in dense urban settings and signal loss inside reinforced-concrete structures [6]. Bluetooth Low Energy beacons achieve two-metre indoor positioning accuracy in ninety-two percent of cases but require hardware installation and periodic battery maintenance [7]. QR code systems present a session-specific code for student scanning; rotating codes that

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 9, May 2026 Impact Factor: 8.2

change every thirty seconds partially mitigate remote proxy scanning but cannot fully prevent a student outside the room from receiving a photograph of the displayed code [8].

Network-layer approaches most directly inform the present system. Dhingra et al. parsed DHCP lease files on a Linux network server to infer student presence, achieving accuracy exceeding ninety-eight percent for correctly registered devices over a two-week evaluation [9]. Kaur and Sharma demonstrated passive Wi-Fi probe request monitoring with detection rates above ninety percent, but acknowledged regulatory concerns about passive device surveillance and noted that modern iOS and Android devices increasingly randomise MAC addresses in probe frames to protect user privacy [10].

The present system extends the DHCP log parsing methodology of Dhingra et al. in four significant respects: the addition of a VLAN-based access control layer that simultaneously records and enforces authorization; the integration of a session code portal for students not using the institutional network; cloud deployment for persistent and globally accessible operation; and a live dashboard in place of batch-processed reporting. No prior work known to the authors has combined all four capabilities in a single deployed system.

III. SYSTEM ARCHITECTURE AND DESIGN

A. Overall Architecture

The system is structured as four tiers with clearly defined responsibilities and minimal interfaces between them. The Network Layer captures device presence events from the laboratory Wi-Fi infrastructure. The Data Processing Layer, hosted on an AWS EC2 t2.micro instance running Ubuntu 22.04 LTS, executes Python monitoring agents and the Flask web application. The Storage Layer is a MySQL 8.0 relational database holding all persistent records. The Presentation Layer delivers the HOD dashboard and student portal through any standard web browser. Event flow is unidirectional: the network layer generates DHCP lease log entries; the processing layer parses these entries and performs database writes; the presentation layer issues read-only queries.

B. Network Layer Design

Three VLANs segment the laboratory network by location and department. VLAN 10 serves CSE Lab A with subnet 192.168.10.0/24; VLAN 20 serves CSE Lab B with subnet 192.168.20.0/24; VLAN 30 serves the Library with subnet 192.168.30.0/24. Each VLAN corresponds to a dedicated DHCP scope on the router that generates a lease record upon each successful address assignment. Access Control Lists are applied at the Layer 3 inter-VLAN routing boundary. For VLAN 10, the ACL explicitly permits DHCP requests only from devices whose MAC addresses appear in the MAC registry with a branch value of CSE. Requests from unregistered devices or from registered devices belonging to other departments are denied, with the denial event written to the router syslog and subsequently processed by the monitoring agent to populate the access violations table. This network-layer enforcement is a capability absent from all alternative approaches reviewed in Section II.

C. Database Design

The storage layer employs a five-table normalised schema in MySQL 8.0. The students table serves as the master identity registry with columns for roll number, name, branch, semester, and institutional email. The mac_registry table maps student identifiers to device MAC addresses, supports multiple devices per student, and includes an is_active flag to deactivate lost devices. The labs table stores VLAN identifiers, capacity values, and a JSON array of permitted branch codes per laboratory. The attendance_sessions table records each session with a check-in timestamp, check-out timestamp, computed duration, assigned IP address, and a source column distinguishing DHCP-detected sessions from portal check-ins. The access_violations table logs each denied connection attempt with the offending MAC address, the VLAN on which access was attempted, the timestamp, and a violation reason code.

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 9, May 2026
Impact Factor: 8.2

D. DHCP Log Parser

The parser employs the Python watchdog library's FileSystemEventHandler to react to file modification events on the DHCP lease log, making the approach event-driven rather than polling-based. Upon detecting a modification, the handler reads all lines appended since the last recorded byte offset, applies a compiled regular expression to extract the MAC address and VLAN identifier, and invokes the decision function. This function queries the MAC registry joined with the students and labs tables.

If no matching registration is found, a violation record is written with reason 'unregistered_device' and an alert email is dispatched. If a match is found but the student's branch is not in the lab's allowed_branches array, a violation record is written with reason 'wrong_department'. If the match is both found and authorized, an attendance session record is inserted. The byte offset is stored between invocations to guarantee exactly-once processing under normal operation.

E. Session Code Portal

Students using personal mobile hotspots instead of the institutional network cannot generate DHCP lease events. To record their attendance without sacrificing proxy resistance, the system provides a Flask-based session code portal. The current code is computed as a four-digit value derived from the SHA-256 hash of a string concatenating the calendar date, the ten-minute window index, and a server-side secret key. The code is displayed on the laboratory projector via a large-font, auto-refreshing HTML page accessible only within the room. A student submits their roll number and the displayed code through the portal; the Flask handler validates both, inserts an attendance record with source set to 'portal', and confirms the check-in. Because the code is visible only on the projector, remote proxy attempts are infeasible.

F. HOD Live Dashboard and Notification Engine

The Flask dashboard presents four summary metric cards—currently present count, total sessions today, violations today, and average session duration—alongside a per-laboratory occupancy bar, a colour-coded live session log, and a reverse-chronological violations list. A JavaScript setInterval call fetches the /api/stats JSON endpoint every ten seconds and updates the relevant DOM elements without a full page reload, ensuring the displayed data is never more than ten seconds stale.

The notification engine dispatches violation alert emails synchronously upon each violation event using Python's smtplib with Gmail SMTP on port 587 and TLS encryption. A scheduled daily summary email is sent at 18:00 each working day, listing present students, absent students, per-laboratory session counts, and a violation category breakdown.

IV. IMPLEMENTATION

The complete system is implemented in Python 3.11 with Flask 3.0 for the web layer, PyMySQL 1.1 for database connectivity, Watchdog 4.0 for file system monitoring, and the Schedule library for timed tasks. All components run on an AWS EC2 t2.micro instance. The project comprises eight source files:

- config.py centralises all environment-specific parameters;
- db_setup.sql defines the schema and inserts sample data;
- simulator.py generates realistic ISC DHCP format log entries for development and testing;
- parser.py implements the event-driven attendance and violation logic;
- alerts.py provides the email notification functions;
- import_students.py bulk-imports student records from CSV; and
- app.py defines all Flask routes.

The network topology is designed and simulated in Cisco Packet Tracer 8.2 using two 2960-series Layer 2 switches connected to a 2911 router configured for inter-VLAN routing via router-on-a-stick. Each VLAN sub-interface hosts 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 9, May 2026 Impact Factor: 8.2

dedicated DHCP pool and an inbound ACL. In a production deployment, these configurations translate directly to physical managed switches without modification to the software components.

Device registration follows a one-time self-service workflow. Students navigate to the /register Flask route, submit their institutional roll number and device MAC address, and receive confirmation. The system supports multiple device registrations per student and allows devices to be deactivated through the is_active flag without deleting historical session records. An Elastic IP address is allocated to the EC2 instance to ensure that the dashboard URL remains stable across instance lifecycle events.

V. EVALUATION AND RESULTS

A. Functional Testing

A structured suite of eight test scenarios was defined and executed to validate system behavior across principal operational cases. Table I summarises the scenarios, inputs, and results.

TABLE I: FUNCTIONAL TEST SCENARIOS AND RESULTS

ID	Description	Input Condition	Expected Outcome	Result
TC-01	Authorized CSE student, correct lab	MAC registered, branch=CSE, VLAN=10	Session recorded; dashboard updated	PASS
TC-02	ECE student attempts CSE lab	MAC registered, branch=ECE, VLAN=10	Violation logged; HOD alert sent	PASS
TC-03	Unregistered device	MAC absent from registry	Violation logged; HOD alert sent	PASS
TC-04	Portal check-in, correct code	Valid roll number + current code	Session recorded, source=portal	PASS
TC-05	Portal check-in, wrong code	Valid roll number + stale code	Error returned; no session record	PASS
TC-06	Portal check-in, unknown student	Unknown roll number submitted	Error returned; no session record	PASS
TC-07	Session duration tracking	Device disconnects after 47 min	Check out and duration_mins updated	PASS
TC-08	Daily summary email	Scheduled trigger at 18:00	Email dispatched with correct counts	PASS

All eight scenarios passed without exception. For TC-01, the attendance row appeared in the database within one second of the simulated DHCP event, and the dashboard reflected the new session within the subsequent ten-second refresh cycle. For TC-02 and TC-03, violation records were inserted and HOD alert emails were received within fifteen seconds of the triggering event. Session duration tracking (TC-07) produced accurate check-out timestamps and duration values in both the DHCP-release-triggered and the cron-timeout-triggered close paths.

B. Performance Benchmarking

Performance was assessed across three dimensions using one hundred simulated DHCP events and fifty dashboard request measurements.

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 9, May 2026
Impact Factor: 8.2

TABLE II: PERFORMANCE BENCHMARK RESULTS

Metric	Median	95th Pct.
Event processing latency (DHCP to DB commit)	0.31 s	0.58 s
Dashboard /api/stats response time	0.74 s	1.21 s
Average CPU utilisation over 72 hours	3.2%	18.0% (peak)
Stable memory consumption	320 MB	—

Event processing latency was dominated by MySQL query execution, which averaged 0.22 seconds. The initial dashboard response time of 1.45 seconds was reduced to 0.74 seconds by adding composite indexes on (student_id, check_in) and (lab_id, check_in). The system operated continuously for seventy-two hours under simulated load without memory leaks or resource exhaustion, confirming suitability for sustained deployment on free-tier cloud infrastructure.

C. Comparative Analysis

Table III positions the proposed approach against representative alternatives across six criteria pertinent to the university laboratory context.

TABLE III: COMPARATIVE EVALUATION OF ATTENDANCE APPROACHES

Approach	Hw Cost	Proxy Prevention	Real-time	No Extra Device	Scalable	Access Control
Manual Register	Nil	No	No	Yes	No	No
RFID/Smart Card	High	Partial	Yes	No	Moderate	No
Fingerprint	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 / QR Code	Medium / Low	Yes / Partial	Yes / Yes	No / No	Moderate / High	No / No
Proposed (DHCP+MAC)	Nil	Yes	Yes	Yes	High	Yes

The proposed system achieves the most favorable profile across all six criteria. Its decisive advantages relative to all reviewed alternatives are the complete absence of additional hardware investment, real-time data availability, and the unique capacity to enforce departmental access control at the network layer—a capability not present in any other approach examined.

VI. CONCLUSION AND FUTURE WORK

This paper has presented a Smart Attendance and Lab Access System that resolves the five principal deficiencies of manual laboratory attendance management through passive network monitoring, VLAN-based access control, cloud deployment, and a time-bound portal fallback. Functional testing confirms that all eight defined scenarios are handled correctly. The system achieves sub-second event processing, delivers dashboard responses under one second after indexing optimisation, and operates within the resource envelope of a free-tier EC2 instance. Crucially, it requires no hardware beyond the Wi-Fi infrastructure already present in any modern university laboratory.

Several extensions are identified for future development:

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 9, May 2026 Impact Factor: 8.2

1. Integrating a lightweight face recognition module—such as MobileFaceNet running on a Raspberry Pi 4 placed at the laboratory entrance—would correlate the physical presence of the device owner with the MAC-based session record, eliminating any residual risk from MAC address spoofing.
2. A native mobile application would give students direct access to their own attendance history and push notifications approaching the minimum attendance threshold.
3. Replacing raw MAC matching with IEEE 802.1X port-based authentication backed by a RADIUS server would address the growing prevalence of MAC address randomisation in iOS and Android devices.
4. A machine learning model trained on historical attendance patterns could identify at-risk students before they breach the minimum attendance threshold, enabling earlier academic intervention.
5. Integration with the university Student Information System via REST API would eliminate the one-time CSV import step and keep enrolment data synchronised automatically.

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] D. Bhatt and R. Patel, "RFID-based attendance management system for educational institutions," International Journal of Computer Applications, vol. 68, no. 12, pp. 14-18, 2013.

[2] A. Kumar, P. Singh, and R. Verma, "Smart attendance system using RFID technology with centralised database," Procedia Computer Science, vol. 45, pp. 308-315, 2015.

[3] S. Patil and M. Bhosale, "UHF RFID for multi-student simultaneous attendance in classroom environments," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 6, no. 3, pp. 1421-1427, 2017.

[4] A. K. Jain, A. A. Ross, and K. Nandakumar, Introduction to Biometrics, 2nd ed. Springer International Publishing, 2016.

[5] L. Zhang and Z. Shen, "Automated face recognition attendance system for educational environments," IEEE Access, vol. 6, pp. 44870-44879, 2018.

[6] M. Hasan, M. Islam, and I. Zarif, "Smartphone-based student attendance system using geofencing and GPS," Journal of Network and Computer Applications, vol. 87, pp. 66-74, 2017.

[7] J. Shi, Z. Lu, and Y. Li, "Indoor attendance marking using BLE beacons in university settings," Mobile Networks and Applications, vol. 23, no. 4, pp. 1012-1021, 2018.

[8] R. Gupta, N. Sharma, and P. Mehta, "Rotating QR code attendance system with device registration for proxy prevention," in Proc. ISDEA, Springer, 2019, pp. 1071-1079.

[9] S. Dhingra, R. Goyal, and A. Patel, "Network-based automated attendance recording using DHCP logs," in Proc. ICCCS, Springer, 2018, pp. 421-428.

[10] G. Kaur and N. Sharma, "Wi-Fi probe request-based presence detection for attendance automation," Wireless Personal Communications, vol. 107, no. 3, pp. 1345-1361, 2019.

[11] S. Patel and M. Gupta, "Cloud-based real-time educational dashboards using AWS EC2 for Indian university contexts," International Journal of Cloud Computing and Services Science, vol. 10, no. 1, pp. 55-63, 2021.

[12] F. Alonso, G. Lopez, D. Manrique, and J. M. Vines, "Cloud-based learning management system deployment: Benefits, challenges, and institutional experience," British Journal of Educational Technology, vol. 51, no. 2, pp. 617-635, 2020.

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 9, May 2026
Impact Factor: 8.2

[13] Python Software Foundation. Python 3.11 Documentation. [Online]. Available: https://docs.python.org/3/
[14] Flask Project. Flask 3.0 User Guide. [Online]. Available: https://flask.palletsprojects.com/
[15] Oracle Corporation. MySQL 8.0 Reference Manual. [Online]. Available: https://dev.mysql.com/doc/
[16] Amazon Web Services. Amazon EC2 Documentation. [Online]. Available: https://docs.aws.amazon.com/ec2/