

Campus Eye: An RFID And BLE Based Smart Campus System for Real-Time Attendance And Indoor Navigation

Dr. Trupti Meshram¹, Urvashi Khandale², Vedika Thaware³, Chaitanya Mamidwar⁴, Shreyash Mohabey⁵,
Ayush Fendar⁶, Vaishnav Vairagade⁷

¹Assistant Professor, Department of Computer Science & Engineering, G H Raison University, Amravati, Maharashtra

^{2,3,4,5,6,7}Department of Computer Science & Engineering, G H Raison University, Amravati, Maharashtra
doi.org/10.64643/IJIRT209744

Abstract—Attendance in engineering colleges is still recorded by roll-call or signed sheets that waste teaching time, are easy to proxy and reach students weeks later. This paper presents Campus Eye, a low-cost smart campus system that pairs an ESP32 microcontroller, an MFRC522 13.56 MHz RFID reader and a Bluetooth Low Energy (BLE) beacon at the classroom door with a React Native (Expo SDK 57) mobile application. Every card tap is posted over HTTPS to a Supabase (PostgreSQL) backend, where a timetable-aware resolver marks the student present, or late after a 10-minute grace window, and Supabase Realtime pushes the mark to every signed-in device within seconds. A single sign-in serves students, teachers and the Dean, and the same node's BLE beacon gives room-level presence for a campus map, a 60-room floor plan, 26 linked 360° panoramas and shortest-path indoor directions. A minute-by-minute evaluation over one teaching week showed that the reader resolved 825 of 1,210 minutes correctly and exposed a fallback rule behind the remaining 31.8% misattribution, which a room-strict rule eliminates. The results show that hardware costing under Rs. 1,000 per classroom can deliver accurate, auditable attendance together with campus wayfinding.

Index Terms—RFID, ESP32, Bluetooth Low Energy, smart campus, IoT attendance, React Native, Supabase, real-time database, indoor navigation, Dijkstra's algorithm.

I. INTRODUCTION

A smart campus applies sensing, networking and software to the daily work of an educational institution so that routine tasks become faster, more accurate and visible to the people they affect [1]. Attendance is the

most frequent of these tasks: a student of a typical section attends up to six periods a day, five days a week, and every period needs a record. Universities commonly require 75% attendance to appear in end-semester examinations, so the record has real academic consequences, yet it is kept in the least reliable way.

Manual roll-call costs several minutes of each 55-minute period, a classmate can answer for an absent student, and registers are compiled into spreadsheets days later, so students learn about a shortfall only near the end of term. Faculty presence in a scheduled class is not captured at all. In addition, new students and visitors struggle to locate rooms inside multi-floor blocks, where GPS and public maps do not help.

At the same time the building blocks of an alternative have become inexpensive. An ESP32 Wi-Fi and Bluetooth microcontroller costs a few hundred rupees, a 13.56 MHz RFID module even less, managed backends such as Supabase provide a database, REST API and real-time change stream without a dedicated server, and React Native with Expo lets a small team ship one application to Android, iOS and the web. Campus Eye combines these components into a single integrated system for automatic attendance, live visibility and campus navigation. The rest of this paper describes the architecture, the modules, the evaluation protocol and the measured results.

The key objectives of this research are:

- To build a classroom node from an ESP32 and an MFRC522 reader that posts every card tap to a cloud database.

- To resolve each tap automatically to the correct timetabled session and mark it present or late.
- To provide one application with a single sign-in and role-specific views for students, teachers and the Dean.
- To push attendance changes to all connected devices in real time, with a complete audit trail.
- To detect room-level presence using a BLE beacon on the same node, without storing biometric data.
- To provide a campus map, an interactive floor plan, 360° street view and shortest-path indoor directions.
- To quantitatively evaluate how accurately taps are attributed to sessions.

II. RELATED WORK

Automated attendance has been approached through several technologies. Kassim et al. [7] and Arulogun et al. [15] built RFID attendance systems connected to a web portal or a local PC database; these removed paper registers but kept data on one machine and offered no live feedback to students. Biometric systems based on fingerprints [5] and iris recognition [4] provide strong identity but serve one student at a time, creating queues at the door, and raise hygiene and cost concerns. Face-recognition systems [3] need no token but depend on lighting and a trained model

and store biometric data, which raises consent and privacy issues. QR-code systems [6] are very cheap, but a code can be photographed and forwarded for remote proxy marking.

Zanella et al. [1] describe the Internet of Things as layered sensing, network and service tiers, a model that maps naturally onto a campus with bounded space, known users and a fixed timetable. Want [2] and Finkenzeller [8] describe passive RFID, where 13.56 MHz ISO/IEC 14443 cards [9] are read in under a second at a few centimetres. For indoor positioning, Zafari et al. [11] survey Wi-Fi, BLE and UWB methods, and Faragher and Harle [10] show that BLE fingerprinting reaches metre-level accuracy but needs dense beacons and calibration. Image-based navigation popularised by Google Street View [14] offers visual guidance without positioning hardware.

A key limitation across existing work is fragmentation: systems address attendance or navigation, students or faculty, and rarely connect a hardware tap automatically to the official timetable while pushing the result live to users with different permissions.

Table 1 summarises the studies discussed above, with the method each used, its main finding and the gap it leaves for this work. Table 2 compares the main attendance approaches with Campus Eye.

TABLE 1 LITERATURE REVIEW

Ref.	Year	Methodology	Key finding	Limitation
[1]	2014	IoT layered architecture for smart cities	Sensing, network and service tiers map onto bounded spaces	Generic; no attendance workflow
[2]	2006	Introduction to RFID technology	Passive tags read quickly at short range	Survey; no system built
[3]	2013	Face-recognition attendance	No token needed per student	Lighting-dependent; stores biometrics
[4]	2010	Wireless iris-recognition attendance	Very strong identity assurance	One student at a time; high cost
[5]	2012	Fingerprint biometric attendance	Resists proxy marking	Queues at door; hygiene concerns
[6]	2014	QR-code attendance	Very low cost	Code can be forwarded for proxy
[7]	2012	Web-based RFID attendance	Removes paper registers	No live feedback to students
[8]	2010	RFID handbook	Reference for 13.56 MHz operation	Reference text only
[9]	n.d.	ISO/IEC 14443 proximity cards	Standard contactless card protocol	Defines cards, not applications

Ref.	Year	Methodology	Key finding	Limitation
[10]	2015	BLE beacon fingerprinting	Metre-level indoor accuracy	Needs dense beacons and calibration
[11]	2019	Survey of indoor localization	Compares Wi-Fi, BLE and UWB	Survey; no attendance link
[12]	n.d.	Managed PostgreSQL with Realtime	DB, REST API and change stream	Vendor-managed service
[13]	n.d.	Expo SDK cross-platform framework	One codebase for Android, iOS, web	Native module limits
[14]	2010	Street-level imagery capture	Visual guidance without hardware	No positioning or routing
[15]	2013	RFID attendance with local PC DB	Automated card-based marking	Data kept on one machine

TABLE 2 COMPARISON OF ATTENDANCE APPROACHES

Approach	Cost	Speed	Proxy resistance	Privacy impact
Manual roll-call	None	Slow	Low	Low
QR code	Very low	Medium	Low	Low
Fingerprint / iris	Med.–High	Medium	High	High
Face recognition	High	Fast	High	High
RFID card (Campus Eye)	Low	< 1 s	Medium	Low

III. SYSTEM ARCHITECTURE

Campus Eye follows a modular three-tier architecture designed for low cost, scalability and real-time responsiveness. The reader reports only “card X was seen at reader Y”; all timetable logic lives in one place in the application. At a high level it consists of three layers (Fig. 1):

1. Edge Tier (Classroom Node):

An ESP32 Dev Module with an MFRC522 reader at the door of room C017 reads card UIDs over SPI, debounces repeated taps for 3 seconds, posts scans over HTTPS, advertises a BLE beacon with the room code using NimBLE, and sends a heartbeat every 60 seconds so that an idle reader can be distinguished from an offline one.

2. Cloud Tier (Supabase):

PostgreSQL tables (readers, cards, scans, attendance, faculty_attendance, live_class and presence), an auto-generated REST API and a Realtime service that streams row changes over WebSockets [12].

3. Client Tier (Expo Application):

A React Native application [13] that keeps global state in a React context, subscribes to the realtime channels,

runs the session resolver on each new scan and renders role-specific screens for students, teachers and the Dean.

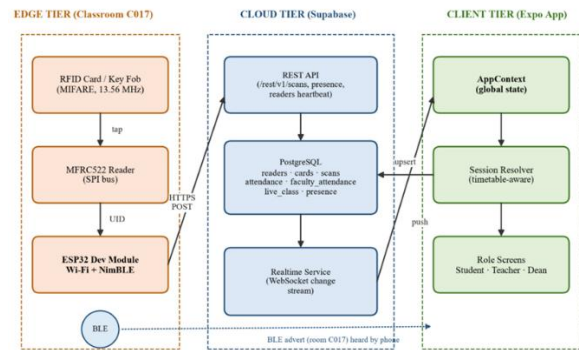


Fig. 1 Three-tier system architecture of Campus Eye

Fig. 2 shows the workflow for the system’s central feature. A card is tapped at the ESP32 reader, which posts {device_id, card_uid} to the scan’s endpoint; Realtime pushes the new scan to the app; the app checks that the card is bound to a person, uses a pinned live class if one exists, otherwise finds the timetabled class running in the reader’s room, and marks the student late or present before upserting the attendance row.

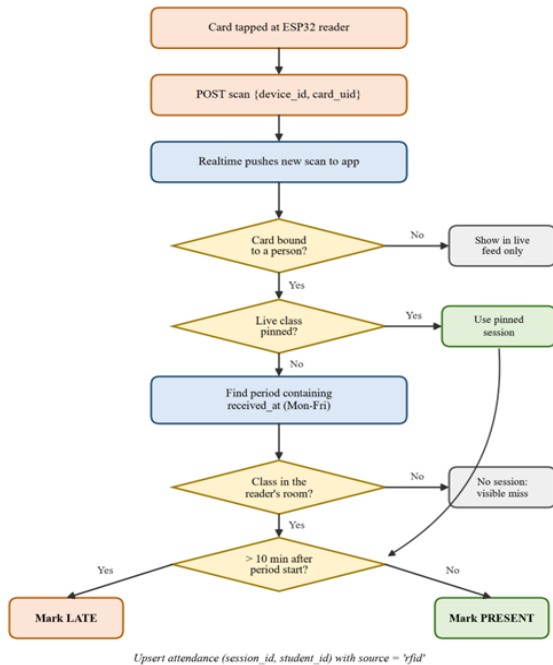


Fig. 2 RFID tap-to-attendance flowchart

The functional behaviour of the system is captured by the use case diagram in Fig. 3. Three actors use the system. Students sign in by password or RFID tap, view maps, the floor plan and street view, find a class, view their own attendance and their RFID card and tap log. Teachers start and stop live classes, take or correct attendance and view subject rosters. The Dean approves registrations, enrolls RFID cards, browses the directory with live presence and monitors faculty attendance.

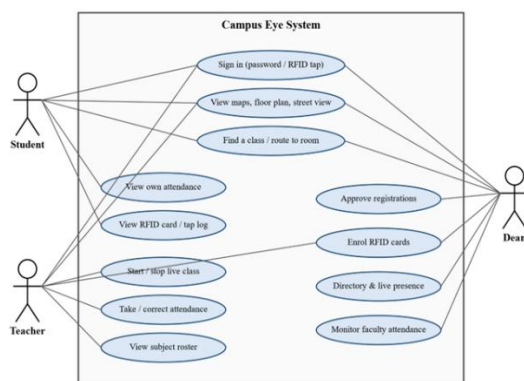


Fig. 3 Use-case diagram for the three roles

IV. SYSTEM MODULES

The proposed system is composed of the following modules, which together provide attendance, presence and navigation.

4.1 RFID Reader Node

The MFRC522 connects to the ESP32's VSPI bus: SDA/SS to GPIO 5, SCK to GPIO 18, MOSI to GPIO 23, MISO to GPIO 19 and RST to GPIO 27. It is powered from the 3.3 V pin because the module is not 5 V tolerant, and a 100 μ F capacitor steadies unreliable reads. On boot the firmware connects to Wi-Fi, synchronises time by NTP and starts BLE advertising. In the main loop it polls for a card, formats the UID as colon-separated hexadecimal, ignores the same UID within 3,000 ms, posts the JSON body {device_id, card_uid, scanned_at} to the scan's endpoint over TLS, writes a presence row for the card owner and blinks the onboard LED to signal success or error. Table 3 lists the hardware required per classroom.

TABLE 3 BILL OF MATERIALS PER CLASSROOM

Item	Qty	Approx. Rs.
ESP32 Dev Module	1	350–450
MFRC522 + card + key fob	1	120–180
100 μ F capacitor	1	5
Wires, USB cable, 5 V adapter	1 set	150–250
Total		650–900

4.2 Session Resolution Module

When a new scan arrives through Realtime, the application looks up the card owner in the cards table and calls the resolver with the reader's room and the server-side receive time. The server time is used rather than the ESP32 clock because the device clock depends on NTP and may drift; the device timestamp is retained only for audit.

The resolver applies the following rules in order:

1. Pinned live class: if a teacher has pinned a live class, attribute the tap to it (a started class auto-stop after 45 minutes).
2. Weekend: return no session on Saturday and Sunday.
3. Running period: collect timetable entries of kind "class" whose period contains the time.
4. Room match: select the entry whose room matches the reader's room.
5. Grace window: mark "late" if the tap is more than 10 minutes after the period start, otherwise "present".

The result is upserted into the attendance table keyed on (session_id, student_id), so a repeated tap updates the existing mark rather than creating a duplicate, and several devices resolving the same scan still produce exactly one row. Faculty taps are written to faculty_attendance in the same way.

4.3 Data and Real-Time Synchronisation Module

The scans table is an append-only source of truth; every attendance row references the scan that created it, providing a complete audit trail. Indexes on received_at and card_uid support the live feed and card lookups. Historical attendance for the term (264 sessions) is generated by a seeded pseudo-random generator for demonstration, and real RFID marks are applied as an overlay: any session with real rows replaces the generated marks for that session. Because every percentage on every screen flow through one function, the overlay reaches all screens without further changes. Table 4 summarises the purpose of each table.

TABLE 4 DATABASE TABLES

Table	Key	Purpose
readers	device_id	Reader, room, heartbeat
cards	card_uid	Card bound to a person
scans	id	Raw log of every tap
attendance	(Session, student)	One mark per student
faculty_attendance	(Session, faculty)	One mark per teacher
live_class	id	Currently armed session
presence	(Type, person)	Last detected room

4.4 BLE Presence Module

The ESP32 advertises a custom 128-bit service UUID whose service data carries the room code and the calibrated RSSI at one metre. The phone scans in the foreground, averages the last five RSSI readings and applies a hysteresis rule (Fig. 4): it enters the room when the average reaches -72 dBm and leaves when it falls below -82 dBm or nothing is heard for 20 seconds. The 10 dB gap prevents flicker near the door.

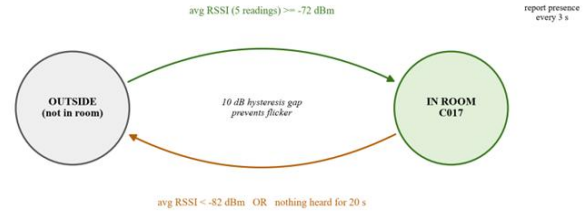


Fig. 4 BLE room-presence state machine with hysteresis

While inside, the phone reports its room every 3 seconds with an estimated distance from the log-distance path-loss model:

$$d = 10^{(P_{1m} - \text{RSSI}) / (10 \cdot n)}$$

where $P_{1m} = -59$ dBm and the indoor exponent $n = 2.5$. Bluetooth is used only for location; attendance is marked only by an RFID tap.

Presence and attendance are both tied to the card's holder through the credential record stored when the Dean enrolls a card. Table 5 shows the credential fields kept for each enrolled card, as displayed in the student's digital RFID credential.

TABLE 5 RFID CREDENTIAL RECORD

Field	Value (sample enrolled card)
Holder	Shreyash Mohabey
UID	04:C2:75:BF: 15:80:00
Serial	SC-H-AI-25
Protocol	ISO/IEC 14443-A · 13.56 MHz
Chip	MIFARE DESFire EV3 · 8K
Issued	17 Jun 2026
Valid thru	30 Jun 2027
Status	Active

4.5 Role-Based Application Module

The login screen has a single identifier field that is matched against student names, emails and roll numbers, then faculty names (titles stripped), emails and codes, and finally the Dean's record; the matched record determines the role. Users can also sign in by tapping an enrolled RFID card. New users submit a registration request that the Dean must approve before sign-in is allowed. Students see overall and subject-wise attendance, a digital RFID credential and their timetable. Teachers start and stop live classes, take or correct attendance and view subject rosters. The Dean

views a searchable directory with live presence and attendance, approves registrations, monitors faculty attendance and enrolls cards by pressing “Start listening” and binding the next tapped UID.

4.6 Campus Map and Indoor Navigation Module

Buildings, roads and grounds were traced from a satellite view into an 876×642 coordinate space and drawn as SVG polygons. The C Block plan has four floors with 15 rooms each, and tapping a room shows the class running there now. Twenty-six panoramas captured inside and around C Block are shown with the Pannellum viewer, each linked to its neighbours by arrows. For directions, every panorama with a floor position becomes a node of a walking graph and every link becomes a corridor or staircase edge weighted by length, with a cost of 150 floor-plan pixels per stair flight. Rooms join the graph at the nearest node, and Dijkstra’s algorithm computes the shortest route from the room where the user was last detected (by BLE or an RFID tap), or from the building entrance when this is unknown. The route is split into one leg per floor, drawn on the plan and converted into step-by-step text; a “Find a class” search lets users look up a room, subject or teacher.

4.7 Key Features

- Automatic, timetable-aware present and late marking from a single RFID tap
- Live attendance pushed to every signed-in device within seconds
- One sign-in by identifier or RFID card, with Dean-approved registration
- Role-specific views for students, teachers and the Dean
- Live-class start and stop, attendance correction and faculty attendance by fob tap
- BLE room-level presence without storing biometric data
- Campus map, 60-room C Block floor plan, 26 linked 360° panoramas and shortest-path indoor directions

V. METHODOLOGY

The development of Campus Eye followed a hybrid methodology that combines the Waterfall model for structured planning with iterative refinement driven by test results:

- Requirement analysis: observing roll-call, discussing pain points with faculty and transcribing the official Semester VII timetable into a data file.
- System design: the three-tier architecture, database schema, resolution rules, role model and hardware wiring.
- Hardware and data setup: assembling and flashing the node, creating the Supabase schema and realtime publication, and capturing 360° panoramas.
- Implementation: the firmware in Arduino C++ and the application in JavaScript with React Native, React Navigation, react-native-svg and react-native-ble-plx.
- Testing: functional, integration and analytical testing, including a minute-by-minute sweep of the resolver against the real timetable.
- Deployment: APK builds with EAS Build and over-the-air JavaScript updates with EAS Update, so fixes reach every phone without reinstalling.

The implemented system was exercised end to end on Android and the web. Representative functional test cases are listed in Table 6.

TABLE 6 REPRESENTATIVE TEST CASES

ID	Test case	Expected result
TC-01	Sign-in by identifier	Matched record determines the role
TC-02	Sign-in by RFID card	Enrolled card signs its holder in
TC-03	Registration request	Sign-in blocked until Dean approves
TC-04	Card enrolment	Next tapped UID bound to the person
TC-05	Live-class start and stop	Taps attributed to the pinned class
TC-06	Tap within grace window	Marked present
TC-07	Tap after 10 minutes	Marked late
TC-08	Duplicate tap	One attendance row per student per session
TC-09	Faculty fob tap	Row written to faculty_attendance
TC-10	Floor switching and routing	Route drawn per floor with step text
TC-11	Street-view navigation	Linked panoramas open by arrow

5.1 Evaluation Protocol

Every minute of the teaching week of 7 to 13 September 2026 was passed through the real resolver module, with no mocks, for each candidate reader room. A minute is *resolvable* when a class is running anywhere in the section. For each room the number of resolvable minutes attributed to the correct session and to a wrong session was counted.

VI. RESULTS AND DISCUSSION

6.1 Functional Evaluation

The functional flows covered sign-in by identifier and by card, registration with Dean approval, card enrolment, live-class start and stop, automatic present and late marking, duplicate-tap handling, faculty attendance, floor switching on the building plan, route generation and street-view navigation. A tap was reflected on subscribed devices within a few seconds, meeting the real-time requirement, and the composite primary keys guaranteed one attendance row per student per session.

6.2 Session-Resolution Accuracy

Table 7 and Fig. 5 report the results of the minute-by-minute sweep.

TABLE 7 SESSION-RESOLUTION SWEEP BY READER ROOM

Room	Resolvable min	Correct	Wrong	Wrong %
C013	1,210	825	385	31.8
C102	1,210	220	990	81.8
C114	1,210	110	1,100	90.9
C302	1,210	55	1,155	95.5

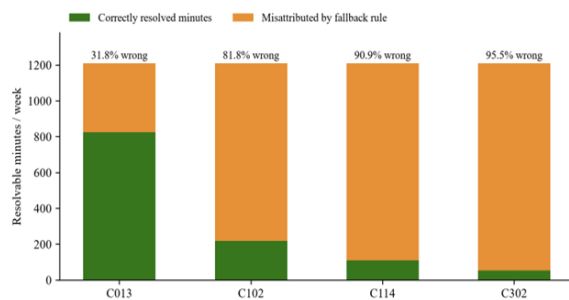


Fig. 5 Correct versus misattributed minutes per reader room

The misattribution comes from the resolver's last rule: when no class matches the reader's room, it falls back to the first class running anywhere, so a student tapping the C013 reader at 08:30 on Monday is marked present in a lecture running in C102. Returning no session when no room matches keeps all 825 correct minutes for C013 and removes all 385 wrong ones, turning a silent wrong mark into a visible miss. This finding shows that the correctness of IoT attendance depends on the resolution logic as much as on the hardware.

6.3 Performance

The attendance overlay scans every overlaid session on each call. With 199 covered sessions (1,990 overlay rows), a full roster sweep rose from 0.00082 ms to 1.736 ms. This is negligible for one section but grows with class size; indexing the overlay by student at merge time reduces each call to a single lookup.

6.4 Comparison with the Existing System

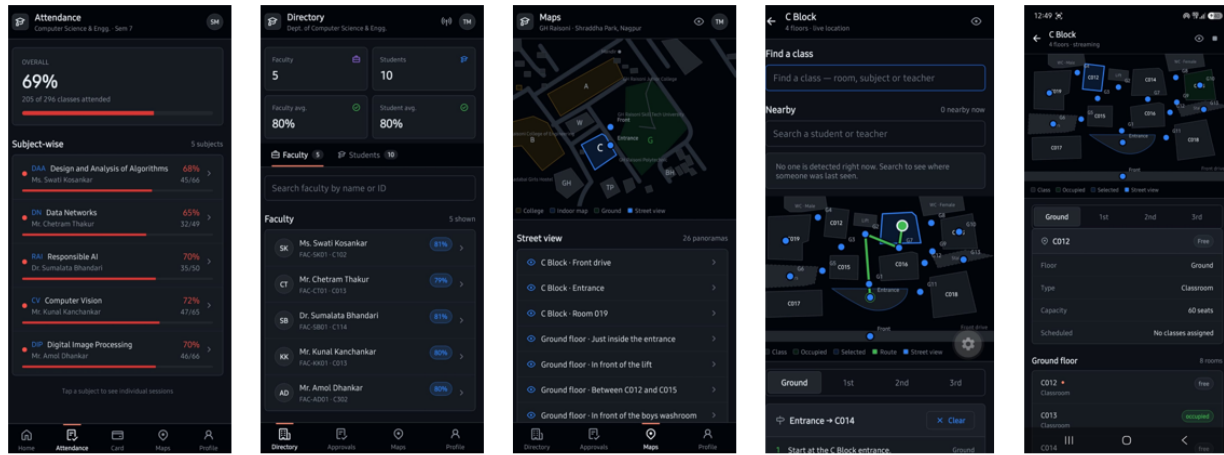
Table 8 compares the manual register with Campus Eye across the parameters that matter to students, faculty and administration.

TABLE 8 MANUAL REGISTER VERSUS CAMPUS EYE

Parameter	Manual	Campus Eye
Time per period	Minutes	< 1 s per student
Proxy	Easy	Needs physical card
Student visibility	Weeks later	Within seconds
Faculty record	None	Fob tap
Audit trail	Paper	Linked raw scan
Navigation	Ask around	Map, plan, route
Cost	Staff time	Rs. 650–900 / room

6.5 Application Screenshots

Fig. 6 shows five screens of the Android build of Campus Eye used during the evaluation: student attendance, the Dean directory, the campus map and street view, find a class with an indoor route, and the C Block floor plan.



(a) Student attendance (b) Dean directory (c) Map & street view (d) Find a class & route (e) C Block floor plan
Fig. 6 Screenshots of the Campus Eye mobile application

6.6 Discussion

A reasonable-looking fallback rule silently misattributed 31.8% of a reader's resolvable minutes; measuring every minute of a real week against the real timetable is what exposed it, and a room-strict rule removes the error entirely. The system also shows that real-time visibility and auditability can be delivered on low-cost hardware: every attendance row links back to the raw scan that created it, and each role sees only the view its permissions allow. Table 9 maps each research objective to the outcome achieved.

TABLE 9 OBJECTIVES AND ACHIEVED OUTCOMES

Objective	Outcome
ESP32 + MFRC522 classroom node	Every tap posted over HTTPS; Rs. 650–900 per room
Automatic session resolution	Timetable-aware present / late with 10-minute grace
One app, role-specific views	Single sign-in for students, teachers and the Dean
Real-time push with audit trail	Marks reach devices in seconds; linked raw scans
BLE room-level presence	Hysteresis at -72 / -82 dBm; no biometric data
Campus wayfinding	Map, 60-room plan, 26 panoramas, Dijkstra routes
Quantitative evaluation	825 / 1,210 correct; fallback error isolated and removed

VII. CONCLUSION AND FUTURE WORK

In this paper, Campus Eye has been presented as an integrated, low-cost smart campus system. An ESP32 node with an MFRC522 reader and a BLE beacon, a managed real-time backend and a single cross-platform application are sufficient to replace roll-call with automatic, timetable-aware attendance that reaches students, teachers and the Dean within seconds, each through a view limited to their role. Dean-approved registration, card enrolment and live-class control make the system usable without technical help, while the campus map, the 60-room floor plan, the panoramas and shortest-path directions extend it from record-keeping to wayfinding. The measured evaluation is itself a contribution: a reasonable-looking fallback rule silently misattributed 31.8% of a reader's resolvable minutes, and a room-strict rule removes this error entirely.

Future work includes applying the room-strict resolver and handling split-batch periods; installing one reader per room with low-attendance alerts to students and parents; hardening security with key rotation and row-level security policies; pairing card taps with BLE presence from the student's own phone to deter proxy attendance; buffering taps on the node during Wi-Fi outages; and extending the floor plan into a multi-floor 3D model with walking points on every floor.

In summary, the main contributions of this work are:

- A classroom node costing under Rs. 1,000 that combines an RFID reader and a BLE beacon on one ESP32.

- A timetable-aware session resolver with grace-window late marking and duplicate-safe upserts.
- A real-time, auditable data model in which every attendance row links to the raw scan that created it.
- A single cross-platform application with role-specific views for students, teachers and the Dean.
- Campus wayfinding through a map, floor plan, linked panoramas and shortest-path indoor directions.
- A minute-by-minute evaluation that exposed and removed a silent misattribution in the resolution logic.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science & Engineering, G H Raison University, Amravati, for its guidance and support throughout this project.

REFERENCES

- [1] Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [2] R. Want, "An Introduction to RFID Technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.
- [3] V. Chintalapati and M. V. Raghunadh, "Automated Attendance Management System Based on Face Recognition Algorithms," in *Proc. IEEE ICCIC*, 2013.
- [4] S. Kadry and M. Smaili, "Wireless Attendance Management System Based on Iris Recognition," *Scientific Research and Essays*, vol. 5, no. 12, pp. 1428–1435, 2010.
- [5] O. Shoewu and O. A. Idowu, "Development of Attendance Management System Using Biometrics," *Pacific Journal of Science and Technology*, vol. 13, no. 1, pp. 300–307, 2012.
- [6] F. Masalha and N. Hirzallah, "A Students Attendance System Using QR Code," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 5, no. 3, 2014.
- [7] M. Kassim, H. Mazlan, N. Zaini, and M. K. Salleh, "Web-Based Student Attendance System Using RFID Technology," in *Proc. IEEE ICSGRC*, 2012, pp. 213–218.
- [8] K. Finkenzer, *RFID Handbook*, 3rd ed. Wiley, 2010.
- [9] ISO/IEC 14443, *Identification Cards—Contactless Integrated Circuit Cards—Proximity Cards*.
- [10] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.
- [11] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.
- [12] Supabase, "Supabase Documentation: Database, REST API and Realtime." [Online]. Available: <https://supabase.com/docs>
- [13] Expo, "Expo SDK 57 Reference." [Online]. Available: <https://docs.expo.dev/versions/v57.0.0/>
- [14] D. Anguelov *et al.*, "Google Street View: Capturing the World at Street Level," *Computer*, vol. 43, no. 6, pp. 32–38, 2010.
- [15] O. T. Arulogun, A. Olatunbosun, O. A. Fakolujo, and O. M. Olaniyi, "RFID-Based Students Attendance Management System," *International Journal of Scientific & Engineering Research (IJSER)*, vol. 4, no. 2, 2013.