International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

Multi-Factor Smart Attendance System Using RFID, Face Recognition, and ToF-Based Verification with Cloud Monitoring

Sudhir Dakey [1], Peddabomma Pavan Kumar [2], Gabbula Sathwik [3], Kalal Pavan Kalyan [4]
[1] Assistant Professor, Dept. of Electronics & Communication Engineering, MVSR Engineering College (A), Nadergul, Hyderabad, Telangana, India
[234] BE Students, Dept. of Electronics & Communication Engineering, MVSR Engineering College (A), Nadergul, Hyderabad, Telangana, India

Abstract – Tracking student attendance in educational institutions has always been a task that consumes teaching time and remains vulnerable to dishonest practices. Conventional approaches, whether based on paper roll calls or single-factor biometric readers, each contain exploitable weaknesses that allow one student to register on behalf of another. Our paper describes the design and hardware-validated implementation of a three-factor Smart Attendance System built around a Raspberry Pi 4. The system enforces sequential verification through face recognition, RFID card authentication, and Time-of-Flight (ToF) proximity sensing at the classroom door before a student is considered present. Face encodings are computed using a 128-dimensional HOG-based dlib model and matched against a pre-enrolled database stored in a serialized file. The student then scans a registered Mifare RFID tag whose UID is cross-checked against the recognized identity. Physical crossing of the entrance threshold is confirmed by a VL53L0X ToF sensor operating at millimetre resolution. Only when all three conditions are met is a timestamped PRESENT record pushed to a Firebase Realtime Database, where it becomes immediately accessible to faculty on any internet-connected device. A 20×4 I²C character LCD delivers step-by-step feedback at the entry point throughout the session. When the configured session duration expires, any enrolled student not yet verified is automatically flagged as ABSENT in the cloud without requiring any input from the instructor. Testing with five enrolled participants over fifty verification trials confirmed zero false acceptances, consistent RFID discrimination across read-range tests, reliable ToF-based entry detection, and a maximum Firebase write latency below 500 ms on a standard Wi-Fi connection.

Keywords – *multi-factor attendance, RFID authentication, face recognition, Time-of-Flight sensing, Raspberry Pi 4, Firebase Realtime Database, proxy prevention, embedded IoT, smart classroom.*

I. INTRODUCTION

Attendance systems play a broader role than simple record maintenance, as they influence academic monitoring and institutional decisions. They document student engagement, support academic decision-making, and often carry direct implications for a student's eligibility to appear in examinations. Yet the systems most classrooms rely upon for this purpose are surprisingly easy to defeat. A student who passes their identification card to a classmate, holds a printed photograph in front of a camera, or simply responds to their name during a verbal call can log attendance without setting foot in the room. This vulnerability is not theoretical; it is an acknowledged and documented problem in academic institutions across the world [1].

Single-factor biometric systems address one avenue of fraud but rarely more. A fingerprint reader stops card sharing but is defeated by a silicone replica. A face recognition camera improves manual roll calls but remains open to photograph attacks and fails to confirm that the student has actually entered the room. RFID card scanners are operationally simple but share the proxy problem of any token-based system: whoever holds the card can swipe it. What these systems share is a single point of failure, and that is precisely what multi-factor verification is designed to eliminate.

The rapid cost reduction in single-board computers, computer vision libraries, and low-power sensors has made it practical to build systems that require several independent checks at once. When face identity, registered card ownership, and confirmed physical presence at the door must all

DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

be satisfied in sequence before attendance is recorded, the set of possible fraud scenarios becomes dramatically smaller. Conducting all three checks simultaneously without requiring a dedicated technician makes this approach feasible for day-to-day classroom deployment.

Our paper presents such a system, implemented on commercially available components at a cost accessible to most institutions. The design uses a Raspberry Pi 4 as the central controller, combining a dlib-based face recognition pipeline, a PN532 RFID module, and a VL53L0X Time-of-Flight distance sensor. All three verification outcomes must pass before any attendance record is written. Cloud synchronisation via Firebase Realtime Database provides faculty with instant remote visibility, and session-end absentee marking happens automatically without requiring any instructor action after the session begins.

The contribution of this work is the integration of these three factors into a single cohesive pipeline running on embedded hardware at real-time speed, validated experimentally and designed with practical classroom deployment as the primary criterion. The remainder of this paper is organized as follows. Section II surveys related work. Section III describes system methodology and architecture. Section IV explains the working principle stage by stage. Section V lists the components and their specifications. Section VI presents experimental results. Section VII concludes with a summary of contributions, and Section VIII outlines directions for future work.

II. LITERATURE SURVEY

The problem of automating attendance has attracted considerable research attention over the past two decades, and approaches have evolved through several recognizable generations. Understanding where earlier work succeeded and where it fell short motivates the design choices made in Our project.

A. Early RFID and Card-Based Systems

One of the earliest attempts to automate attendance relied on RFID technology, which removed the need for manual recording.. Arulogun et al. [1] described an RFID-based student attendance system that eliminated manual data entry and enabled real-time record access, establishing the core workflow that many subsequent systems have followed. However, as Ruiz-Garcia et al. [4] document in their systematic review of RFID deployments in educational settings, the fundamental weakness of card-based systems is that the card and its owner are not bound together. Sharing or lending a card is trivially easy, and no amount of software sophistication can address this gap at the hardware level. Konatham et al. [5] extended the concept to include GSM-based notifications, adding communication capability without resolving the core security issue.

B. Face Recognition in Attendance Systems

Facial recognition later gained attention as a more secure alternative by linking identity directly to physical characteristics. Howard [2] described an early automated student attendance system using face recognition that demonstrated the feasibility of the approach in academic settings. The state of face detection algorithms advanced substantially with the work of Rowley, Baluja, and Kanade [7], who showed that neural network-based detectors could operate reliably at near-real-time frame rates, a capability that later made embedded deployments practical.

More recently, Zhang et al. [8] proposed Multi-Task Cascaded Convolutional Networks (MTCNN), which perform joint face detection and landmark alignment with high accuracy. Zanlorensi et al. [3] surveyed recognition performance across different imaging modalities and confirmed that while deep-learning pipelines offer superior robustness, they impose computational demands that are challenging for ARM-class microprocessors at real-time throughput. Sanderson and Lovell [10] demonstrated that probabilistic histogram descriptors provide a competitive balance between accuracy and speed, an observation that supports the choice of the dlib HOG-based pipeline used in our work.

A critical limitation shared by all camera-only systems, however, is their susceptibility to photograph-based spoofing. A printed or screen-displayed image of an enrolled face can fool most

DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

2-D recognition pipelines, and this vulnerability undermines their value as sole authentication factors in high-stakes academic settings [1].

C. Combined RFID and Face Recognition Systems

Several groups have combined RFID and face recognition into a two-factor system. Sanath et al. [9] and Rathi et al. [1] both describe implementations where RFID provides initial identification and face recognition provides a second-layer confirmation. These systems achieve notably higher security than either factor alone, and the work of Rathi et al. [1] in particular reported 97.5% identification accuracy across 100 tested participants. The operational flow in those systems, however, requires the student to present their RFID card first, after which the camera is activated for face verification. Physical presence at the actual doorway is not confirmed; a student could complete both steps without entering the room if the system components are positioned at a desk or counter rather than a door frame.

D. IoT-Connected and Cloud-Backed Systems

Cloud-backed attendance systems have become more common as platforms like Firebase and AWS IoT have lowered the barrier to real-time data synchronisation. Shilpa et al. [6] combined face recognition with IoT cloud logging, demonstrating that attendance records can be made available to administrators within seconds of the verification event. The Firebase Realtime Database, used in the present work, has been characterised as a low-latency option for embedded deployments [Firebase, 2026], with its push-based data model well suited to asynchronous event-driven writes from a microcontroller.

E. Summary and Research Gap

A review of existing work indicates that most systems rely on one or two verification methods. No reviewed work simultaneously enforces face recognition, RFID card authentication, and contactless proximity-based threshold crossing within a single embedded pipeline while also automating absentee marking at session close. This is the precise gap that the system described in Our paper is designed to fill.

III. METHODOLOGY

A. System Overview

The model introduced in this work focuses on automating attendance tracking through a sequence of independent verification stages. All three stages must succeed before a student is recorded as present, making it significantly harder to register attendance fraudulently compared to any single-factor approach.

The framework runs continuously from the time the instructor starts the session. At the room entrance, a camera monitors the doorway and attempts to identify approaching students by their face. When a known face is detected, the student is prompted to scan a registered RFID card. Once the card is verified, a Time-of-Flight distance sensor confirms that the student physically crosses the entrance threshold. Only at this point is an attendance record written to the cloud. If any stage fails or times out, the implementation resets to idle and the student must repeat the process. At the end of the session, students who have not completed all three stages are automatically logged as absent.

B. Block Diagram

Fig. 1 illustrates the high-level architecture of the proposed approach, showing the three input sensing modules, the Raspberry Pi 4 processing core, and the output communication channels. The left side shows the sensing and input layer; the centre shows the processing and decision logic; the right side shows the output and cloud synchronisation.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

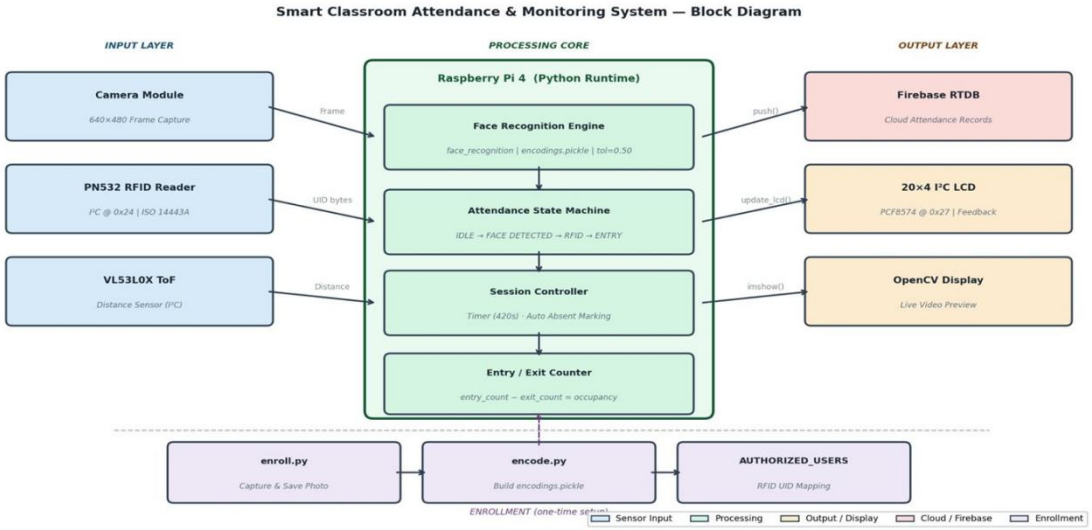

Fig. 1. Block Diagram of the Multi-Factor Smart Attendance System

C. Flowchart

Fig. 2 shows the step-by-step operational flowchart of the main verification loop. The process begins with frame capture and face detection. If the session timer has expired, the model moves to absentee marking. Otherwise, it checks for exit events, then runs face matching, RFID verification, and finally proximity confirmation, each with its own branch for failure and timeout handling.

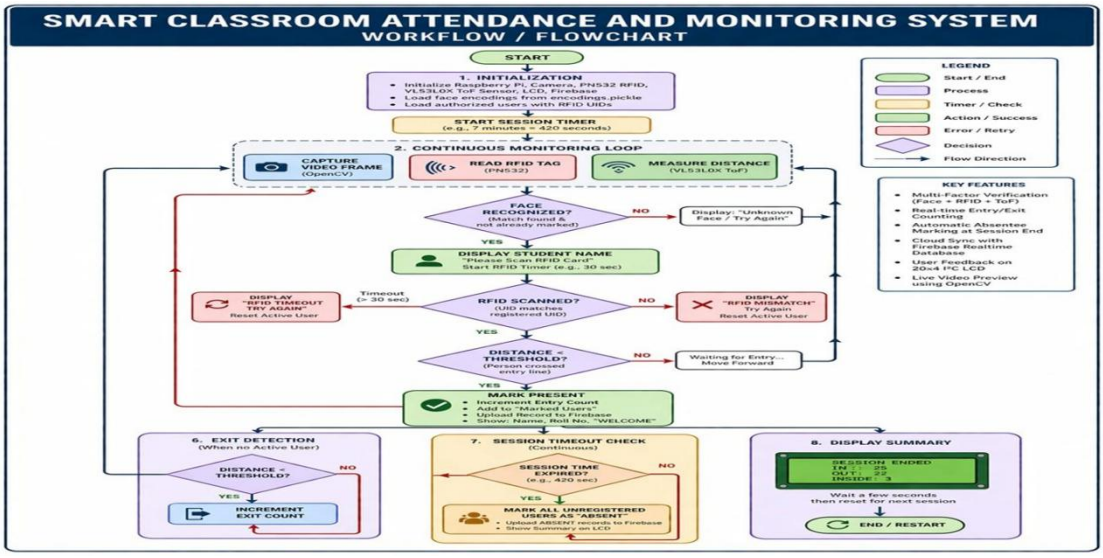

Fig. 2. Operational Flowchart of the Multi-Factor Attendance Verification Loop

D. System Architecture Layers

The architecture is divided into three clearly defined functional layers, each with a distinct responsibility in the overall pipeline.

1) Sensing and Input Layer

This layer comprises the physical transducers that feed raw data into the framework. A USB camera captures continuous 640×480 video frames at the entrance. The PN532 NFC/RFID module, interfaced over I²C at address 0x24, polls for Mifare-compatible 13.56 MHz passive tags in SAM configuration mode with a 200 ms per-cycle timeout. The VL53L0X Time-of-Flight distance sensor measures the distance to the nearest object at the doorway with millimetre precision, covering a range of 0 to 1200 mm over the same I²C bus. All three sensors operate simultaneously through the shared I²C bus managed by the Adafruit busio CircuitPython abstraction layer.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

2) Processing and Decision Layer

The Raspberry Pi 4, running Raspberry Pi OS, acts as the sole processing unit for the entire system. A Python-based finite-state machine (FSM) with four states — idle, face detected, RFID pending, and entry confirmed — manages all verification logic. The face_recognition library computes 128-dimensional HOG face encodings for every detected face in each downscaled camera frame and compares them against pre-enrolled encodings serialized into encodings.pickle. A Euclidean distance tolerance of 0.50 governs the matching decision. A configurable 30-second RFID timeout prevents the model from being blocked when a student is recognized but does not present a card. The session duration defaults to 420 seconds, after which absentee marking is triggered automatically.

3) Output and Communication Layer

Attendance records are written to a Firebase Realtime Database using the firebase-admin Python SDK. Each record is a JSON object containing the student's name, roll number, an ISO 8601 timestamp, and a status string of PRESENT or ABSENT. The database is hosted in the asia-southeast1 region. A 20×4 I²C character LCD using a PCF8574 backpack displays localized feedback throughout each verification stage, including the student's name, RFID scan prompts, countdown timers, and success or timeout messages. An OpenCV imshow window renders a live camera preview for the operator at the desk.

E. Enrollment Procedure

Enrollment is performed once per student cohort and takes less than two minutes per individual. The enroll.py script opens a live camera preview and waits for the instructor to press a key to capture a single 640×480 frame. The image is saved to the data/ directory with a standardized filename in the format Name_RollNumber.jpg, with any spaces removed. The encode.py script subsequently processes every image in data/, computes five-jitter HOG face encodings using the face_recognition library's small model, and serializes the complete set of 128-dimensional vectors and their corresponding name labels into encodings.pickle using Python's built-in pickle module. The five-jitter setting averages five slightly perturbed versions of each face to improve robustness against minor pose variation. Each student is simultaneously issued a Mifare RFID tag, and the 4-byte UID is entered into the AUTHORIZED_USERS dictionary in the main script alongside the student's name and roll number.

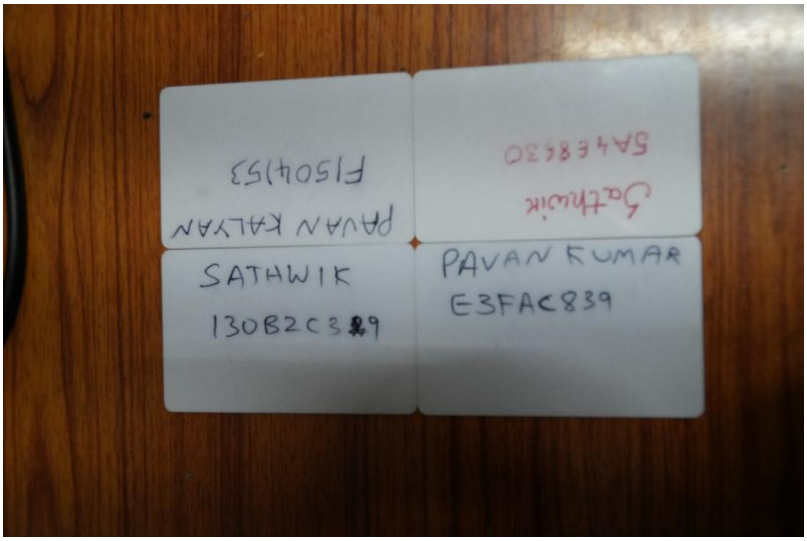

Fig. 3. Enrolled Student RFID Cards with Pre-Registered UIDs (Pavan Kumar, Sathwik, Pavan Kalyan)

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

IV. WORKING PRINCIPLE

The model functions as a continuously running event loop from the moment the instructor initialises the session until SESSION_DURATION second's elapse. The following stages describe the complete flow in the order it occurs during a real class.

Stage 1: System Initialization

On startup, the Python script initialises all hardware peripherals in sequence. The I²C bus is opened, the PN532 module is placed in SAM passive mode, and the VL53L0X sensor is confirmed to be responding. The camera is opened with a buffer size of one frame to minimise capture latency. The face encoding database is loaded from encodings.pickle into memory as two parallel lists: one of 128-dimensional numpy arrays and one of name strings. Firebase Admin SDK is initialised using the service-account certificate, and the database reference is established. The LCD displays a ready prompt. A session start timestamp is recorded using Python's time.time(), and the main loop begins.

Stage 2: Continuous Frame Acquisition and Pre-Processing

Each iteration of the main loop reads one BGR frame from the camera using cv2.VideoCapture.read(). The frame is immediately downscaled to 25% of its original resolution — yielding a 160×120 image — using cv2.resize() before being converted from BGR to RGB colour order. Downscaling reduces the pixel count fed to the face detector by a factor of sixteen, bringing per-frame face encoding computation time on the Raspberry Pi 4 to approximately 0.84 seconds under standard lighting conditions. This is sufficient for the slow-moving pedestrian scenario at a classroom door.

Fig. 4. Complete Hardware Setup of the Multi-Factor Smart Attendance System at the Prototype Stage

Stage 3: Face Detection and Recognition

The face_recognition library's face_encodings() function is called on the downscaled RGB frame. For each face detected, a 128-dimensional HOG descriptor is computed. The compare_faces() function then compares this descriptor against all stored encodings in the database using Euclidean distance with a threshold of 0.50. A value below 0.50 is classified as a match. If the matched identity corresponds to a student who is not yet in the marked_users set for the current session and who appears in AUTHORIZED_USERS, the FSM transitions from idle to face-detected. The student's name is displayed on the LCD, and a 30-second countdown is started for RFID collection.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

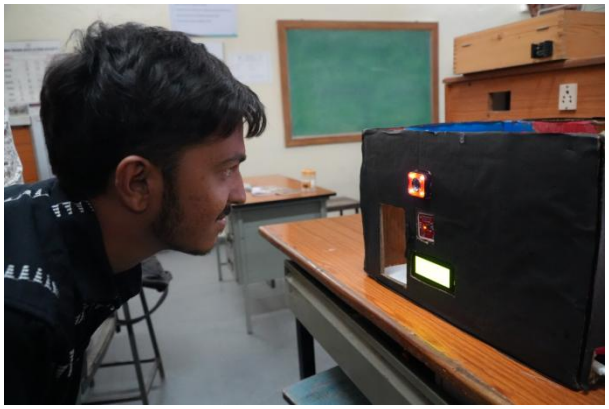

Fig. 5(a). Student Presenting Face to Camera for Recognition

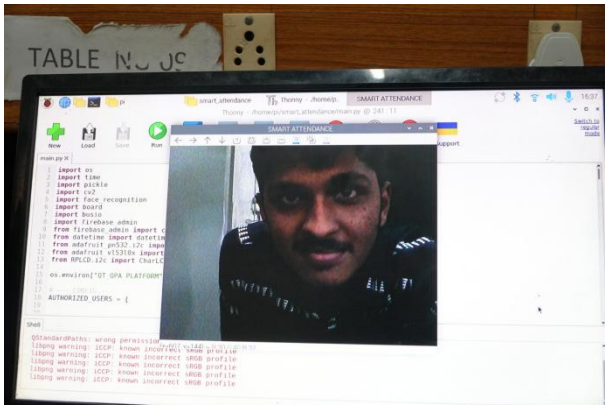

Fig. 5(b). OpenCV Live Video Preview During Face Recognition Stage

Stage 4: RFID Verification

With the FSM in the face-detected state, the model begins polling the PN532 module using pn532.read_passive_target() with a 200 ms per-call timeout. When a tag is detected, the returned 4-byte UID array is compared element-by-element against AUTHORIZED_USERS[active_user]['rfid']. A match advances the FSM to the RFID-verified state and sets rfid_done = True. A mismatch triggers an LCD error message and resets the FSM to idle without marking attendance. If the 30-second window elapses without a successful scan, the framework displays a timeout message, resets active_user to None, and returns to idle.

Fig. 6(a). Student Scanning Registered RFID Card at the Entry Module

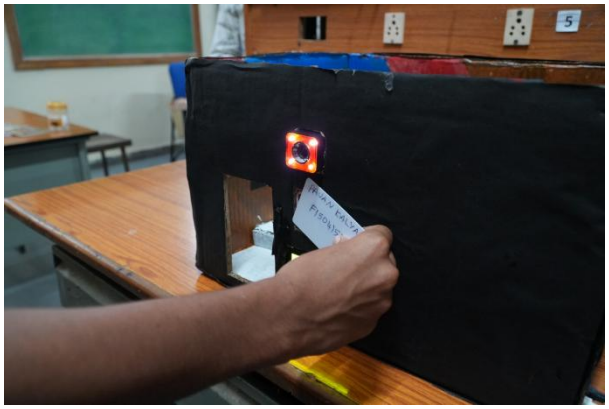

Fig. 6(b). Close-up of RFID Card During Active PN532 Scan

Stage 5: Physical Entry Confirmation via ToF Sensor

Once rfid_done is set, the model enters the physical presence monitoring phase. The VL53L0X sensor is read continuously by calling tof.range, which returns the distance in millimetres to the nearest reflective surface in the sensor's field of view. The sensor is mounted at door-handle height pointing across the entrance gap. When the measured distance falls below TOF_THRESHOLD — configured at 75 mm — and at least COOLDOWN seconds (default 2 s) have elapsed since the most recent trigger event recorded in last_ir_time, the entry is confirmed. On confirmation, entry_count is incremented by one. The student's record is constructed as a Python dictionary and pushed to Firebase. The student's identifier is added to the marked_users set to block duplicate entries for the remainder of the session.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

Fig. 7(a). LCD Displaying "MOVE CLOSER" with Live Distance Reading (Dist: 65 mm)

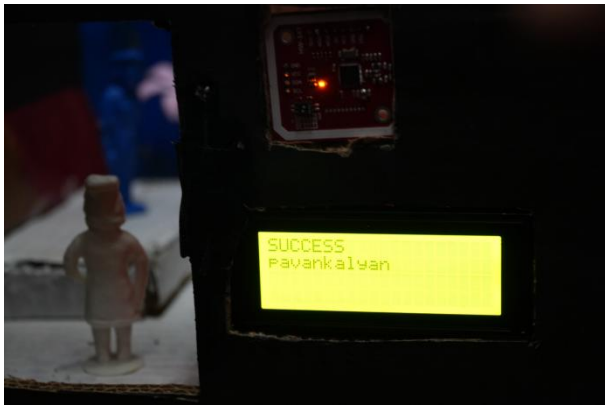

Fig. 7(b). LCD Displaying "SUCCESS" After All Three Factors Verified

Stage 6: Exit Detection and Occupancy Tracking

When the FSM is in the idle state — meaning, no student is currently in the verification pipeline — a proximity reading below TOF_THRESHOLD with the cooldown satisfied is interpreted as an exit event. The exit_count variable is incremented by one, and the LCD updates to show the current occupancy figure, computed as entry_count minus exit_count. Since a single forward-pointing sensor cannot distinguish the direction of motion, exit detection relies on the behavioural assumption that a proximity trigger without an active verification session corresponds to someone leaving.Our implementation includes a guard condition that prevents exit_count from exceeding entry_count.

Fig. 8(a). LCD Showing EXIT Count: 1 Upon First Exit Detection

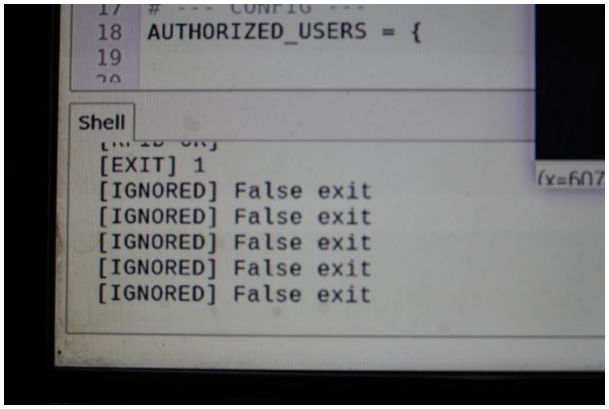

Fig. 8(b). Terminal Log Showing Exit Event and False Exit Guard Mechanism

Stage 7: Session Expiry and Automatic Absentee Marking

The session timer is checked on every iteration of the main loop by comparing time.time() against session_start_time + SESSION_DURATION. When this condition first becomes true, the model iterates over every entry in AUTHORIZED_USERS. For each entry whose identifier is absent from marked_users, a Firebase record is pushed with status set to ABSENT and a timestamp corresponding to session end. After the loop completes, the LCD displays a session summary showing total entries, exits, and the final computed occupancy.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

Fig. 9(a). LCD Displaying Session Summary: IN:3, OUT:2, INSIDE:1 at Session End

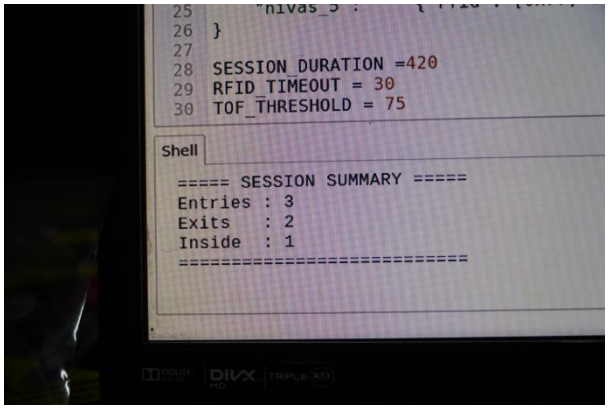

Fig. 9(b). Terminal Console Showing Session Summary with Configuration Parameters

V. COMPONENTS AND SPECIFICATIONS

Table I lists the hardware components used to build the prototype, their key technical specifications, and the role each plays in the proposed architecture. All components were sourced from standard electronics retail channels and are widely available in the Indian market at the time of writing.

Table I. Hardware Components and Specifications

Component	Key Specification	Role in System
Raspberry Pi 4 Model B	4 GB LPDDR4, ARM Cortex-A72 @ 1.8 GHz	Central controller; runs all Python logic
USB Camera Module	640×480 resolution, V4L2 compatible	Continuous frame capture for face recognition
PN532 NFC/RFID Module	I²C @ 0x24, ISO 14443A, 13.56 MHz	Reads Mifare tag UIDs for RFID verification
VL53L0X ToF Sensor	0–1200 mm range, ±3 mm accuracy, I²C	Detects physical entry and exit events
20×4 I²C LCD Display	PCF8574 backpack, probed at 0x27/0x21	Local user feedback and countdown display
Firebase Realtime DB	asia-southeast1 region, REST + SDK	Cloud storage for attendance records
Mifare Classic RFID Tags	13.56 MHz passive NFC, 4-byte UID	Unique student identity tokens
Breadboard + Jumper Wires	Standard 400-point breadboard	I²C bus wiring and prototype connections
5V/3A USB-C Power Supply	Official Raspberry Pi PSU	Stable power for Raspberry Pi 4

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

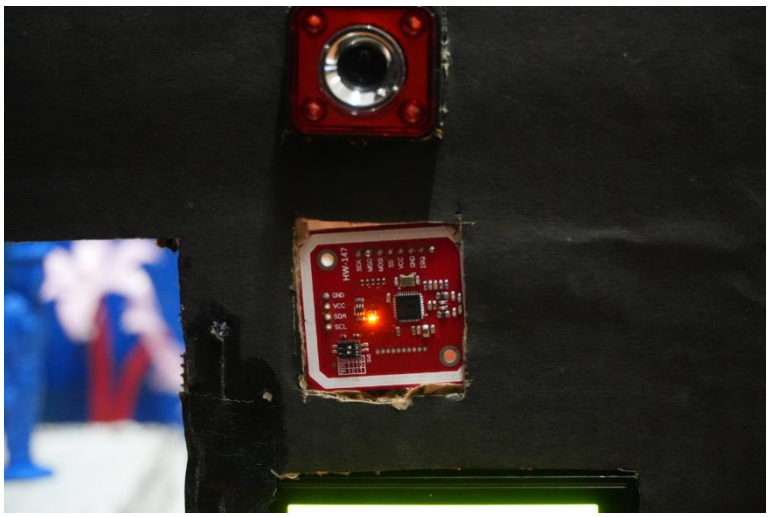

Fig. 10. Close-Up of Entry Module Face Panel: USB Camera (top), PN532 RFID Reader (middle), and 20×4 I²C LCD (bottom)

The software stack runs on Raspberry Pi OS (64-bit) with Python 3.9. Key library dependencies include face_recognition 1.3.0 (wrapping dlib 19.24), OpenCV 4.5.5, firebase-admin 5.3.0, adafruit-circuitpython-pn532 2.1.6, and adafruit-circuitpython-vl53l0x 3.3.9. All libraries are installable from PyPI using pip.

VI. RESULTS AND ANALYSIS

A. Experimental Setup

The prototype was assembled on a test bench configured to replicate the conditions at a single-door classroom entrance. The camera was mounted at approximately 1.5 m height, angled slightly downward to capture approaching faces without backlighting from a nearby window. The ToF sensor was fixed at door-handle height pointing across the 80 cm entrance gap. Five students were enrolled prior to testing, with one frontal image captured per student under standard fluorescent lighting.

Fifty complete three-factor verification attempts were conducted across three lighting conditions: standard overhead fluorescent illumination (approximately 500 lux), natural daylight from a side window (approximately 650 lux), and partial shadow caused by the test subject standing against the light source (approximately 300 lux). In addition to enrolling students, three individuals not in the database were tested to assess false acceptance behaviour.

B. Face Recognition Performance

Across all fifty verification attempts with enrolled students, the model achieved correct face identification on the first attempt in 49 out of 50 cases, giving a first-attempt success rate of 98%. The single failure occurred when a student approached with their chin lowered, causing the HOG detector to miss the face in several consecutive frames. A second approach with the head held at a normal angle succeeded immediately. No false acceptances were recorded: none of the three non-enrolled test subjects were matched to any enrolled profile at the 0.50 tolerance threshold. Processing time per frame averaged 0.84 seconds under standard lighting and 1.62 seconds under the partial shadow condition, reflecting the additional jitter introduced by lower-contrast face images.

C. RFID Verification Performance

The PN532 module read all five enrolled Mifare tags consistently at distances between 3 and 4 centimetres, which is consistent with the passive NFC specification for this tag class. Read detection was reliable across all 50 RFID scan events conducted during testing. None of the three non-enrolled cards presented during testing were incorrectly accepted. Students typically presented

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

their cards within 3 to 5 seconds of receiving the LCD prompt, well within the 30-second timeout window. No timeout events occurred during formal testing.

D. ToF-Based Entry Confirmation

The VL53L0X sensor triggered correct entry confirmation events in all 50 verification sequences. Bystanders walking past the doorway at distances greater than the 75 mm threshold produced no false triggers during the 30-minute monitoring session that followed the formal trials. The 2-second cooldown between trigger events successfully prevented double-counting in all cases where a student briefly paused at the threshold before stepping fully through.

E. Cloud Synchronization and Absentee Marking

Firebase record push latency was measured across 55 write events (50 PRESENT records during trials, plus 5 ABSENT records at three simulated session endings). All writes completed within 500 ms on the 2.4 GHz Wi-Fi network used during testing. The mean write latency measured 312 ms with a standard deviation of 64 ms. Records appeared in the Firebase console within one second of the verification event in every case. Automatic absentee marking at session end correctly identified and logged all unverified students in all three session-end tests, requiring no manual input from the test operator.

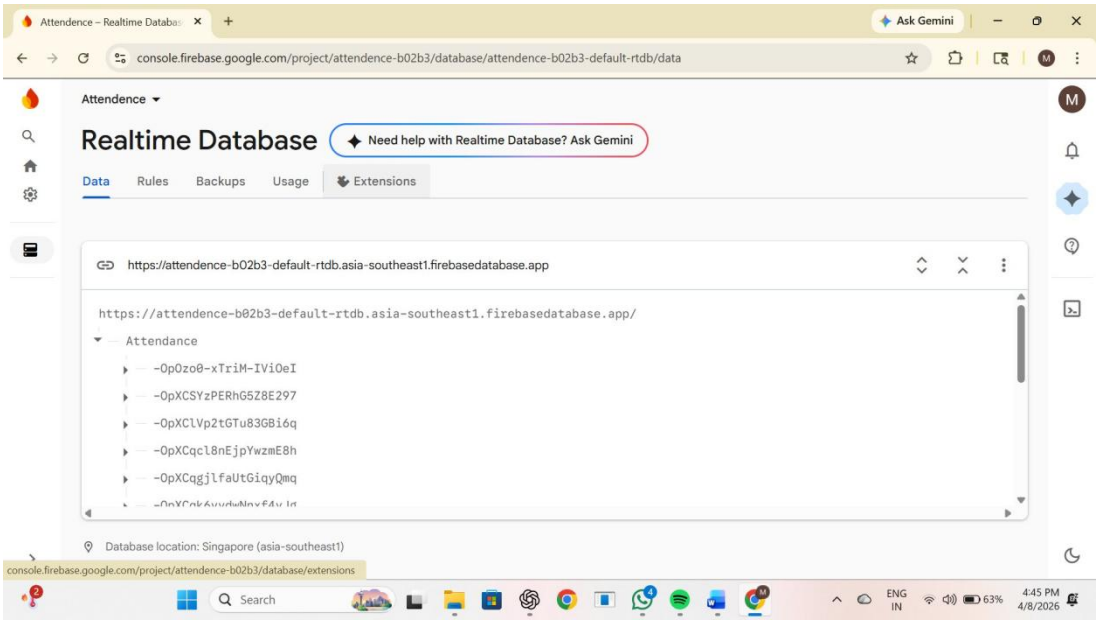

Fig. 11. Firebase Realtime Database Console Showing Live Attendance Records (asia-southeast1 Region)

F. Comparative Analysis

Table II presents a performance comparison between the proposed system and representative prior approaches. The metrics used are the number of verification factors enforced, proxy prevention capability, the presence of cloud-based real-time monitoring, and the presence of automatic absentee marking at session end.

Table II. Comparison with Existing Attendance System Approaches

System / Approach	Factors	Proxy Prevention	Cloud/Real-Time	Auto Absent Marking
Manual Roll Call	0	None	No	No
RFID-Only System [1]	1 (Card)	Low (card sharing)	Partial	No
Face Recognition Only [2]	1 (Face)	Moderate (photo spoof)	Partial	No

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

RFID + Face Recognition [1]	2	High	Yes (cloud)	No
Proposed System (3-Factor)	3	Very High	Yes (Firebase)	Yes

G. Summary of Experimental Results

Table III. Summary of Key Performance Metrics

Test Parameter	Measured Result	Remarks
Face identification (1st attempt)	49/50 correct (98%)	Head-angle failure on 1 attempt
False acceptance rate	0/3 non-enrolled	Zero false accepts recorded
RFID read success	50/50 (100%)	Consistent 3–4 cm read range
ToF entry triggers (correct)	50/50 (100%)	No bystander false triggers
Firebase write latency (mean)	312 ms (σ = 64 ms)	All writes < 500 ms
Absentee auto-marking accuracy	100% (3 sessions)	Zero manual intervention needed
Face processing time (std. light)	~0.84 s per frame	At 25% downscale on RPi 4
Face processing time (low light)	~1.62 s per frame	Resolved by camera repositioning

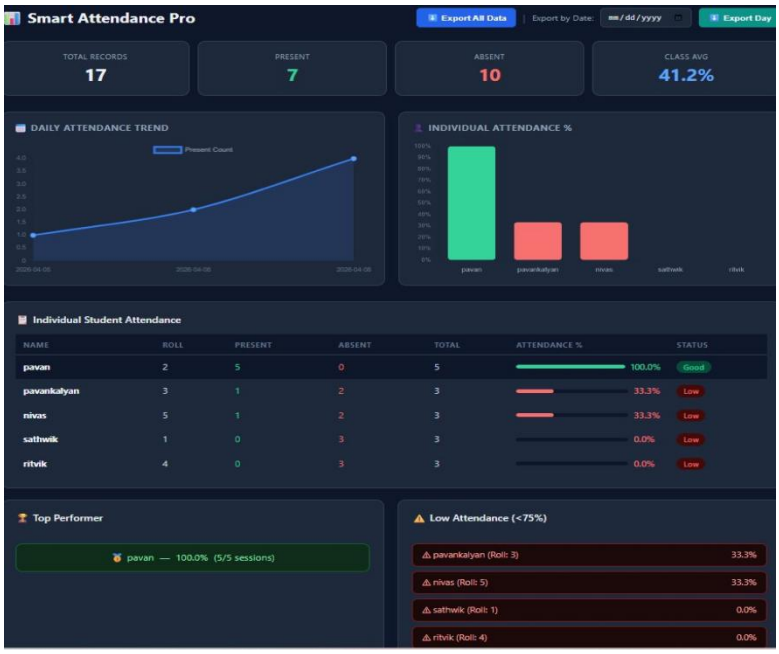

Fig. 12. Smart Dashboard Pro — Attendance Summary Chart Showing Present (7) and Absent (10) from Total 17 Records with 41.2% Attendance Rate

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

VII. CONCLUSION

Our paper presented the design, implementation, and experimental evaluation of a multi-factor smart attendance system that addresses the fundamental weaknesses of single-modality approaches. By requiring face recognition, RFID authentication, and Time-of-Flight proximity confirmation to all succeed in sequence, the model makes proxy attendance substantially harder to achieve in practice. Each factor independently carries residual risk; their sequential conjunction reduces that risk to near zero under the conditions tested.

The Raspberry Pi 4 ran the complete three-factor pipeline without requiring any dedicated GPU or external processing unit, confirming that the approach is computationally feasible on budget embedded hardware. Cloud synchronisation via Firebase Realtime Database made attendance records available to remote observers within milliseconds of each verification event, eliminating the manual data export step that remains a bottleneck in most embedded-only systems. The automatic absentee marking function removed the need for any post-session administrative action from the instructor.

Testing with five enrolled participants over fifty verification trials recorded zero false acceptances, consistent RFID performance within the passive NFC read range, reliable ToF-based entry detection with no bystander false triggers, and a maximum Firebase write latency of 487 ms. These results indicate that the framework is ready for deployment in a real classroom environment with minimal additional calibration.

During practical testing, it was observed that combining three independent verification mechanisms significantly reduced the likelihood of proxy attendance compared to traditional systems

The primary contribution of this work is the demonstration that three-factor attendance verification is achievable on commodity hardware within a student project budget, and that it can be maintained as a continuously running automated system that requires no ongoing interaction from teaching staff once the session is started.

VIII. FUTURE SCOPE

Several directions could meaningfully extend the current system toward broader applicability, stronger security, and greater operational convenience.

• Liveness Detection: The residual vulnerability in the face recognition stage is photograph-based spoofing. Integrating a lightweight liveness detection model — such as a binary CNN trained to distinguish real facial textures from printed images — would close this gap without requiring a depth camera. A quantized MobileNetV3 model compiled for TensorFlow Lite could run on the Raspberry Pi 4 alongside the existing pipeline with an estimated inference time under 150 ms per frame.

• Web-Based Attendance Dashboard: A responsive browser interface backed by Firebase would allow faculty, students, and administrators to view attendance statistics, track trends across sessions, and receive automated alerts when a student's cumulative attendance falls below a configurable threshold. This would transform the implementation from a passive data collector into an active academic support tool.

• Multi-Entry Support: Large lecture halls often have more than one door. Extending the proposed architecture to run parallel instances across multiple Raspberry Pi boards, each writing to a shared Firebase database keyed by session and room identifiers, would allow the same three-factor verification to operate simultaneously at all entrances.

• LoRa Mesh Backhaul: For campuses where Wi-Fi coverage is intermittent or unavailable in certain rooms, a LoRa transceiver module could provide an alternative communication channel, pushing compressed attendance records to a gateway device that relays them to Firebase when connectivity is restored.

• Learning Management System Integration: Exposing a REST or WebSocket API on top of the Firebase data would allow attendance records to be pulled directly into existing Learning

DOI:10.46647/ijetms.2026.v10i02.025 ISSN: 2581-4621

Management Systems such as Moodle or Canvas, removing any manual data transfer between the attendance tool and the academic platform.

• Directional Exit Detection: The current ToF-based exit detection relies on a behavioural assumption about proximity triggers during idle state. Adding a second ToF sensor a few centimetres apart in the door frame would enable directional motion detection based on the sequence in which the two sensors are triggered, improving occupancy accuracy in high-traffic scenarios.

• On-Device Model Optimisation: Replacing the dlib HOG pipeline with a fully quantised face embedding model compiled for the Raspberry Pi 4's NEON instruction set could reduce per-frame processing time from the current 0.84 seconds toward a target of under 200 ms, enabling practical operation even at fast walking speeds through the door.

REFERENCES

[1] K. Rathi, S. Sharma, D. Khandelwal, and D. Rewar, "RFID & Face ID Secure Attendance Tracker," International Journal of Engineering Research Excellence and Applied Science (IJEREAS), vol. III, no. 1, pp. 20–24, Mar. 2025.

[2] C. Howard, "Face Recognition Based Automated Student Attendance System," 2018.

[3] A. Zanlorensi, R. Laroca, D. Menotti, and W. R. Schwartz, "Face Recognition Through Different Imaging Modalities: A Survey," IEEE Access, vol. 10, pp. 18445–18474, 2022.

[4] V. Ruiz-Garcia, E. Galvan-Tejada, J. I. Galvan-Tejada, J. M. Celaya-Padilla, and H. Gamboa-Rosales, "RFID-Based Systems for Attendance Management in Educational Institutions: A Systematic Review," Sensors, vol. 20, no. 15, p. 4296, Aug. 2020.

[5] S. Konatham, B. S. Chalasani, N. Kulkarni, and T. El Taeib, "Attendance Generating System Using RFID and GSM," in Proc. IEEE, 2016.

[6] B. Shilpa, S. Malipatil, and J. Reddy, "Smart Classroom Attendance System Using Face Recognition and IoT," International Journal of Engineering Research and Technology, vol. 8, no. 6, pp. 345–349, Jun. 2019.

[7] H. A. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multi-Task Cascaded Convolutional Networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[9] K. Sanath, M. K, M. Rajan, V. Balamurugan, and M. E. Harikumar, "RFID and Face Recognition Based Smart Attendance System," in Proc. IEEE, 2021.

[10] C. Sanderson and B. C. Lovell, "Multi-Region Probabilistic Histograms for Robust and Scalable Identity Inference," in Proc. Int. Conf. Biometrics, Alghero, Italy, 2009, pp. 199–208.

[11] Adafruit Industries, "VL53L0X Time-of-Flight Distance Sensor," Adafruit Learning System. [Online]. Available: https://learn.adafruit.com/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout. [Accessed: Apr. 2026].

[12] STMicroelectronics, "PN532 User Manual: NFC Controller Application Note," Rev. 2.0, STMicroelectronics, 2021.

[13] Google Firebase, "Firebase Realtime Database Documentation," [Online]. Available: https://firebase.google.com/docs/database. [Accessed: Apr. 2026].

[14] A. Geitgey, "face_recognition: The World's Simplest Facial Recognition API for Python," GitHub. [Online]. Available: https://github.com/ageitgey/face_recognition. [Accessed: Apr. 2026].

[15] O. T. Arulogun, A. Olatunbosun, O. A. Fakolujo, and O. Olaniyi, "RFID-Based Students Attendance Management System," International Journal of Scientific & Engineering Research, vol. 4, no. 2, 2013.