cisdiJournal

Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 17 No. 3, September 2026 - www.cisdijournal.net

Multimodal Biometric Authentication: Fusing Face and Fingerprint Recognition for Intelligent Attendance Systems

[1]Abiola, O.A., [2]Babalola, A.I., [3]Adeyemi, F.A., [4]Joel, J.
& [5]Odufuwa, J.E.
Department of Computer Science, University of Ibadan
Ibadan, Oyo State, Nigeria
E-mails: [1]oladimejiarowolo@yahoo.co.uk, [2] ibabalola1@gmail.com, [3] adeyemifavour844@gmail.com ,
[4] joeljoshua250@gmail.com
jesufunminire2832000@gmail.com

ABSTRACT

This paper documents the design and implementation of an intelligent multimodal attendance and authentication system that combines facial recognition and fingerprint verification using IoT devices, aimed at the well-known weaknesses of manual roll calls, paper registers, and RFID cards in academic settings. The hardware consists of an ESP32-CAM for facial capture and a separate ESP32 board paired with an AS608 optical fingerprint sensor. Both report to a Laravel 13 web application responsible for student records, course enrolment, attendance sessions, and device management. Development was iterative: facial recognition was first tested against the system webcam before the ESP32-CAM was integrated, with the fingerprint module added afterward. The system was thoroughly tested to confirm that the core attendance objectives were met. Results confirmed that the combined facial and fingerprint approach addresses the proxy attendance problem and removes the manual overhead that paper-based systems impose.

Keywords: Biometric Authentication, Facial Recognition, Fingerprint Verification, manual Overhead, Attendance Management System, ESP32

CISDI Journal Reference Format
Abiola Oladimeji Abiola, Babalola Ayomide Isaiah, Adeyemi Favour Adesoji , Joel Joshua & Odufuwa Jesufunminire Emmanuel (2026): Multimodal Biometric Authentication: Fusing Face and Fingerprint Recognition for Intelligent Attendance Systems. Computing, Information Systems, Development Informatics & Allied Research Journal.
Vol 17 No 3, Pp 23-40. Available online at www.isteams.net/cisdijournal. dx.doi.org/10.22624/AIMS/CISDI/V17N3P2

1. INTRODUCTION

Attendance management in modern educational institutions serves as a foundational administrative requirement for verifying student participation and academic compliance. Despite its importance, traditional tracking mechanisms remain highly vulnerable to security gaps and operational inefficiencies. Manual paper registers and verbal roll calls consume significant instructional time and are inherently susceptible to clerical errors and proxy attendance. While token-based replacements, such as Radio Frequency Identification (RFID) cards, reduce administrative overhead, they are possession-based credentials that can be easily shared or misplaced, failing to establish absolute physical identity. To overcome the vulnerabilities of token and knowledge-based credentials, academic institutions have increasingly deployed biometric systems. However, standard single-factor (unimodal) biometric systems create severe operational bottlenecks due to a lack of technical redundancy.

In a unimodal fingerprint authentication loop, physical factors such as dry skin, grease, or surface abrasions on an optical sensor plate can cause immediate false rejections. Similarly, unimodal facial recognition systems frequently degrade under variable classroom lighting or off-angle poses, leading to system timeouts. As a result of unimodal frameworks lacking a structural fallback track, any single sensor failure or environmental constraint results in user lockout and requires manual database overrides.

Recent advancements in computer vision and artificial intelligence have enabled real-time, high-fidelity deep learning inference to execute smoothly across distributed network layers. Deep Convolutional Neural Networks (CNNs) can extract abstract facial geometries directly from raw pixel grids, compressing them into high-dimensional, compact feature vector embeddings for rapid similarity cross-referencing. Concurrently, fingerprint processing algorithms reliably map localized friction ridge patterns and minutiae coordinates such as ridge endings and bifurcations into distinct serialized strings. By leveraging these complementary biometric traits, an institutional network can migrate away from cloud-dependent configurations and deploy lean edge hardware architectures to handle immediate capture sequences.

This study directly addresses the structural limitations of unimodal systems by developing an Intelligent Multimodal Attendance and Authentication System. By fusing contact-based optical fingerprint scanning with server-side computer vision facial recognition pipelines, the proposed system establishes a disjunctive, multi-layered defense perimeter. The primary objective is to engineer a fault-tolerant architecture where the two independent physiological modalities compensate for each other's environmental and mechanical limitations. This multi-tier framework delivers real-time synchronization and rapid verification loops under two seconds without requiring expensive infrastructure upgrades, establishing a secure, seamless, and automated attendance logging workflow.

1.1 Research Objectives

This study presents the design and implementation of an Intelligent Multimodal Attendance and Authentication System that integrates ESP32-CAM-based facial recognition with AS608 fingerprint authentication, coordinated by a Laravel web application and a dedicated Python biometric service. The aim is to design and implement a secure, IoT-enabled multimodal biometric attendance system that achieves accurate, real-time student verification while allowing either biometric modality, or a manual fallback, to confirm a student's presence. The specific objectives are to: review existing biometric attendance literature to identify gaps; design and develop a functional facial recognition module using deep learning; implement flexible authentication that supports facial recognition, fingerprint, or both, with fallback for suboptimal conditions; build an IoT-enabled architecture that captures, verifies, and logs attendance in real time; provide a web portal for administrators and lecturers to manage and view attendance records; and test the system's performance, accuracy, and reliability under realistic conditions.

2. RELATED WORK

Several studies have explored facial recognition for classroom attendance. Demir (2023) implemented a CNN-based facial recognition pipeline for educational settings, achieving improved robustness over rule-based methods but reporting reduced accuracy under extreme lighting and occlusion. Kamath et al. (2023) applied ArcFace-based embeddings to improve class separability among visually similar students, though the approach required substantial high-quality training images per individual. Patel et al. (2025) used CCTV feeds for passive, contactless attendance recognition, but found that camera resolution, viewing angle, and occlusion from student movement affected accuracy.

Adeyemi (2024) integrated facial recognition into an electronic gate system for combined access control and attendance, while Touzene et al. (2024) demonstrated lightweight MobileNet-based models suitable for on-device inference at the edge. Abiola et al. (2024a) applied YOLO to real-time face detection in an automatic door system, achieving reliable recognition under live operating conditions; the work stops short of attendance logging or any secondary verification method. Across these studies, a consistent gap is the absence of an alternative verification pathway when facial recognition fails, leaving systems vulnerable to a single point of failure.

Fingerprint-based systems show a parallel pattern. Obiora et al. (2025) deployed an embedded fingerprint system at a Nigerian university with register and record modes, reporting high accuracy but noting that environmental factors such as the harmattan season affected dry-finger readings. Moses et al. (2024) documented a complete hardware and software implementation framework using the R307 sensor, highlighting enrolment time and sensor maintenance as ongoing concerns. Abiola et al. (2024b) demonstrated a related application of embedded AI, building a gesture-controlled autonomous dustbin that responds to body movements in real time; a design that shares the same edge-inference challenges seen in biometric IoT deployments, namely constrained hardware and the need for low-latency response. Other studies on IoT-connected fingerprint modules (Kumar et al., 2024) and edge-based fingerprint matching (Tiwari et al., 2025) reported network dependency and limited computational capacity as recurring constraints. As with facial-only systems, fingerprint-only implementations generally lack a fallback when the sensor cannot read a print, forcing either manual entry or denial of attendance.

Taken together, the literature points to two persistent gaps: single-modality systems create a single point of failure, and most academic prototypes are evaluated on sample datasets rather than on live hardware in realistic conditions. This study addresses both gaps directly by combining facial recognition and fingerprint verification under an OR-based fallback policy, and by carrying the implementation through to working ESP32 hardware tested with a live webcam and camera feed rather than stopping at an algorithmic demonstration.

Table I: Related Works Comparison

System (Modality)	Ref.	Reported Accuracy	Key Limitations	Gap Addressed by Proposal
Multi-Biometric Authentication for Attendance Fraud Mitigation (Multimodal – Fingerprint + Facial)	(Abdulkareem et al., 2024)	Reported 98% for the combined fingerprint+facial system	Both fingerprint and face are required together, so failure of either sensor blocks a legitimate user	This study replaces the AND-based requirement with an OR-based fallback (either modality, or manual override, is sufficient)
YOLO-Based Face Detection for Automatic Door Access (Facial)	(Abiola et al., 2024a)	Reliable recognition under live operating conditions	No attendance logging or secondary verification method	Extends live face-detection reliability into a full attendance system with OR-based fallback
IoT-Based Biometric Attendance Management System (Biometric/IoT)	(Adedoyin et al., 2025)	99.3%, with fingerprint capture-and-verify time of ~4s per student (vs. ~13s for manual roll call)	Network failures disrupting functionality (uses NodeMCU + fingerprint sensor + LCD)	Adds OR-based dual-modality fallback and a real-time WebSocket-driven session view
Electronic Gate System for Attendance Monitoring (Facial)	(Adeyemi, 2024)	High accuracy reported across datasets	Performance depended on camera position and lighting at entry points	Improved attendance accuracy and enhanced security
Class Attendance System with Deep Learning (Facial)	(Demir, 2023)	Competitive recognition accuracy	Computationally intensive; limited suitability for low-power devices	Optimizes deep learning models for deployment on embedded edge devices
ArcFace-Based Attendance Automation (Facial)	(Kamath et al., 2023)	High (>95% under controlled conditions)	Performance degrades under occlusion, pose variation, and poor lighting	Introduces multimodal fallback to maintain reliability under adverse conditions
Smart Attendance System using Sensor & IoT (Fingerprint)	(Kumar et al., 2024)	High accuracy with real-time logging	Network dependency and latency	Shifts core biometric processing to the edge for offline resilience
Biometric Fingerprint Attendance System	(Moses et al., 2024)	Moderate to high accuracy	Susceptibility to environmental factors and large enrolment time	Implementation framework combining hardware and software

System (Modality)	Ref.	Reported Accuracy	Key Limitations	Gap Addressed by Proposal
(Fingerprint)				
Development of Fingerprint-Based Attendance Monitoring System (Fingerprint)	(Obiora et al., 2025)	High recognition accuracy	Performance affected by dirty/worn fingerprints; harmattan-season dry-finger readings	Combines fingerprint with facial recognition to reduce failure cases
CCTV-Based Classroom Attendance System (Facial)	(Patel et al., 2025)	High accuracy in static classroom settings	Requires fixed camera angles and stable lighting; privacy concerns	Uses edge-based verification and optional fingerprint authentication to enhance flexibility and privacy
Facial Recognition Attendance Monitoring System using Deep Learning (Facial)	(Thalor & Gaikwad, 2023)	Not reported in title-level record	Despite the "deep learning" title, the described methodology uses classical techniques (Haarcascade, Eigenfaces, SVM, Fisherface) rather than CNN-based embeddings	Replaces classical recognition with CNN-based (dlib/InsightFace) embeddings, stores records in a relational database with real-time WebSocket updates, and adds a fingerprint fallback
Intelligent Edge-Based Fingerprint Attendance System (Fingerprint)	(Tiwari et al., 2025)	High accuracy under controlled usage	Limited computational power for complex matching	Uses optimized models and optional facial authentication for robustness
Embedded Intelligent Attendance System (Facial)	(Touzene et al., 2024)	High accuracy with lightweight CNNs	Limited robustness to severe occlusion and crowd density	Integrates secondary biometric modality for redundancy

3. METHODOLOGY

3.1 System Architecture

The system follows a hybrid edge-cloud architecture. Two classes of edge devices handle biometric capture: an AI-Thinker ESP32-CAM module mounted on an ESP32-CAM MB carrier board captures facial images, and a separate ESP32 development board interfaced with an AS608 optical fingerprint sensor captures fingerprint scans.

Both devices authenticate to the backend using device-specific API tokens and communicate over HTTP, sending periodic heartbeats that update their online status and IP address. The backend is a Laravel 13 application that handles routing, the database, background job processing, and a REST API split into two authentication groups: Sanctum-based tokens for human users (administrators and lecturers), and device-specific tokens for the ESP32 hardware. The frontend is rendered server-side using Laravel Blade templates with Livewire components for reactive behaviour, styled with Tailwind CSS, and bundled with Vite. A standalone Python service built on FastAPI performs all biometric computation. It exposes endpoints for embedding generation, matching, liveness detection, and health checks, and uses InsightFace for face detection and embedding extraction. Laravel communicates with this service over HTTP rather than performing biometric computation itself, keeping the concerns of web application logic and machine learning inference separate.

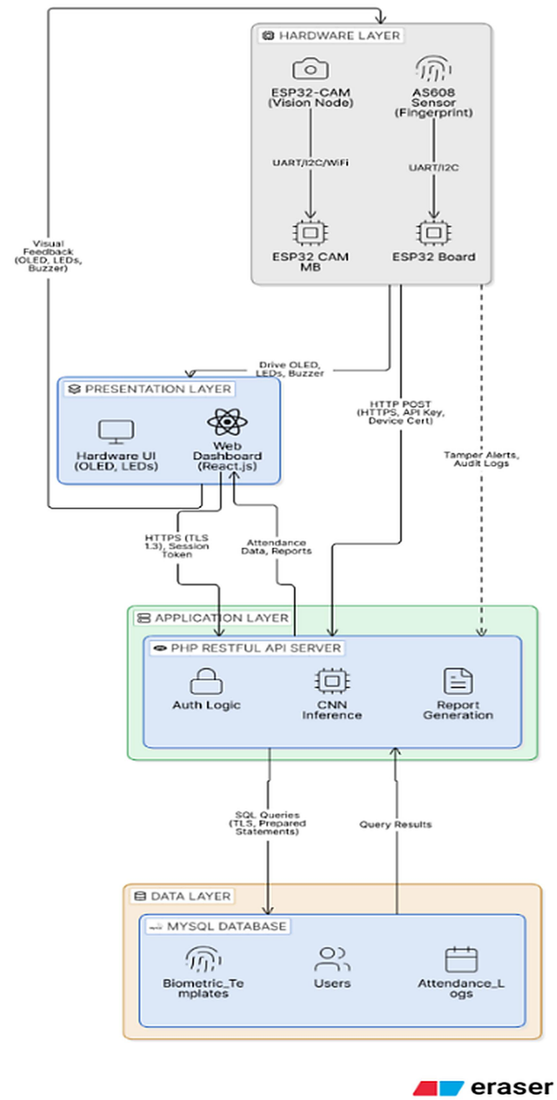

Fig. 1. System Architecture

Real-time updates are delivered through Laravel Reverb, a WebSocket server. When an attendance record is created, whether from the camera, the fingerprint device, or a manual override, an event is broadcast on private channels for the relevant session and lecturer, allowing the live session view to update without a page refresh. Computationally heavier or non-blocking tasks, such as generating end-of-session summaries and deleting temporary attendance images, are dispatched to a background queue worker so that they do not delay the HTTP response to the requesting device.

3.2. Database Design

The relational schema, implemented in MySQL through Laravel migrations, centres on a students table that tracks each student's biometric enrolment status as one of pending, face_only, fingerprint_only, or complete.

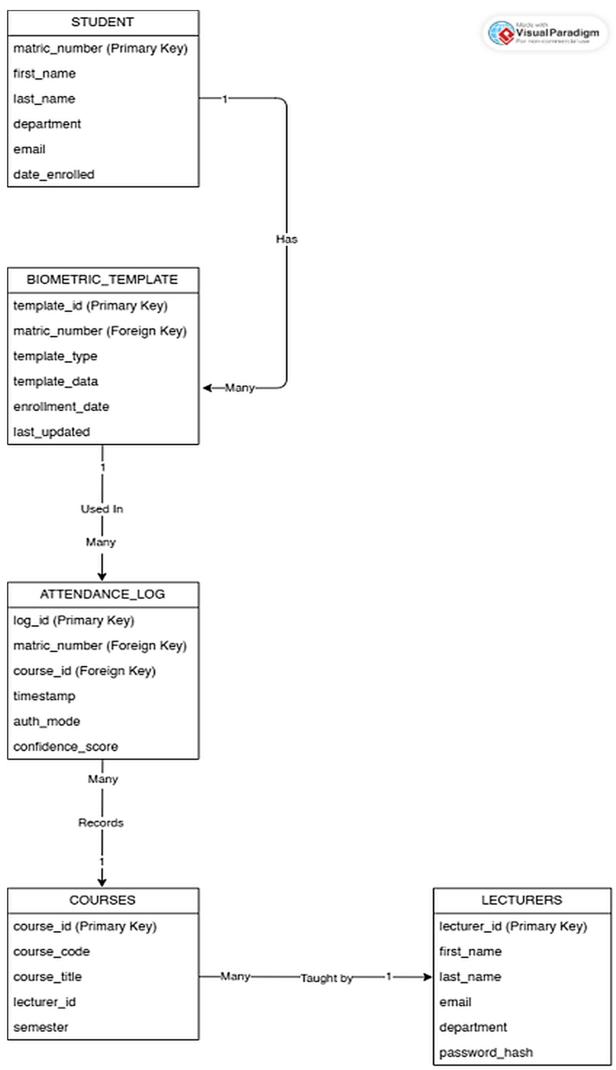

Fig. 2. Database Design

A face_embeddings table stores the 128-dimensional embedding vectors returned by the Python service as JSON, with each student retaining only the most recent active embedding. A fingerprint_enrollments table links a student to a sensor template identifier on a specific device, so that when the fingerprint scanner reports a template match, the corresponding student can be looked up. Course and enrolment tables define which students belong to which course, and the attendance session roster is built from these enrolments rather than the full student list. The attendance_records table stores one record per student per session, recording the verification method (face, fingerprint, face_confirmed, or manual), a confidence score for biometric matches, and an idempotency key. A unique database constraint on the combination of student and session prevents duplicate marking, even under concurrent requests.

3.3 Authentication and Verification Workflow

Verification follows a flexible, adaptive workflow. During enrolment, a captured face image is sent to the Python service, which detects the face, extracts a 128-dimensional embedding, scores image quality (sharpness and face size), and returns the embedding for storage if it passes quality checks and is not too similar to an existing enrolment. Fingerprint enrolment is performed on the device itself: the lecturer initiates an enrolment command, the device prompts the student to place a finger on the AS608 sensor, and the resulting template identifier is reported back to Laravel.

During an active attendance session, the ESP32-CAM periodically captures a still image, which Laravel proxies to the Python matching endpoint along with the embeddings of every student enrolled in the course roster. The matching pipeline computes a query embedding for the captured frame and compares it against each candidate using cosine distance, returning the closest match and its confidence score if it falls below a configurable rejection threshold. Independently, the fingerprint device scans a finger, identifies the corresponding template, and reports the result with a session identifier, timestamp, and idempotency key. Either method, on its own, is sufficient to mark a student present, or a lecturer can additionally mark a student present manually from the live roster if both biometric methods fail. This OR-based fallback logic is the core mechanism by which the system avoids the single-point-of-failure weakness identified in the reviewed literature.

The fingerprint module also maintains an offline queue: if network connectivity is lost, scans are stored locally and replayed once connectivity is restored, with server-side idempotency checks preventing duplicate or stale records from being committed.

3.4. Implementation Tools

The system was implemented using the following tools and technologies:

- Firmware: C++ on the Arduino framework for ESP32, built and flashed using PlatformIO; ArduinoJson for payload serialisation; the Adafruit Fingerprint Sensor library for AS608 communication; LittleFS for persisting device credentials and offline attendance records on the fingerprint module, and NVS for the camera module.
- Backend: Laravel 13 with Blade, Livewire for reactive UI components, Laravel Reverb for WebSocket broadcasting, Laravel Sanctum for API token authentication, Tailwind CSS for styling, and Vite for asset bundling.
- Biometric service: Python with FastAPI and Uvicorn, InsightFace for face detection and embedding, and Pillow/NumPy for image processing.
- Local development environment: Laravel Herd for the PHP/Nginx/Node runtime, DBngin for a local MySQL instance, and TablePlus for database inspection.

3.5 Development Approach

Development proceeded iteratively. The facial recognition pipeline was first validated end-to-end using the development machine's webcam before the ESP32-CAM hardware was introduced, allowing the detection, embedding, and matching stages to be confirmed in isolation from hardware integration concerns. The ESP32-CAM was then assembled on its carrier board, flashed with firmware defining a custom four-partition layout to support credential persistence and future over-the-air updates, and provisioned with Wi-Fi credentials, a server URL, and an API token via a serial provisioning interface. Once the camera module was confirmed to be working end-to-end with live hardware, the fingerprint module was integrated and tested in the same manner, including a deliberate network disconnection to validate the offline queue and synchronization logic.

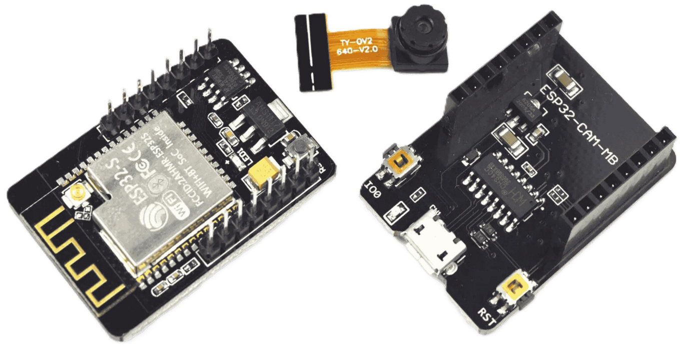

Fig. 3. ESP32-CAM Individual Components

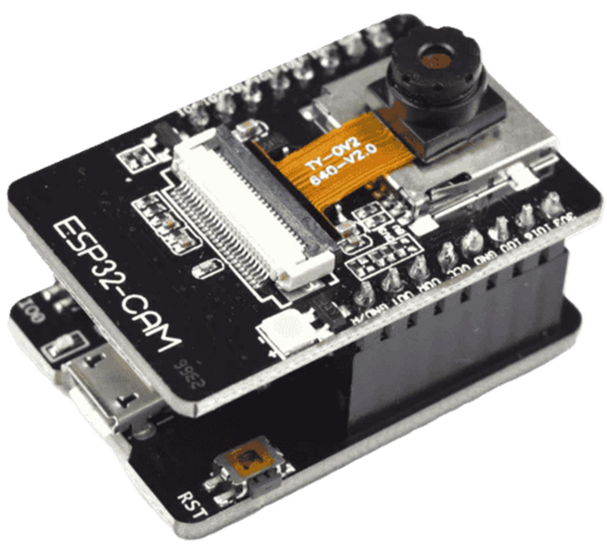

Fig. 4. ESP32-CAM Coupled Components

Fig. 5. AS608 Fingerprint Scanner

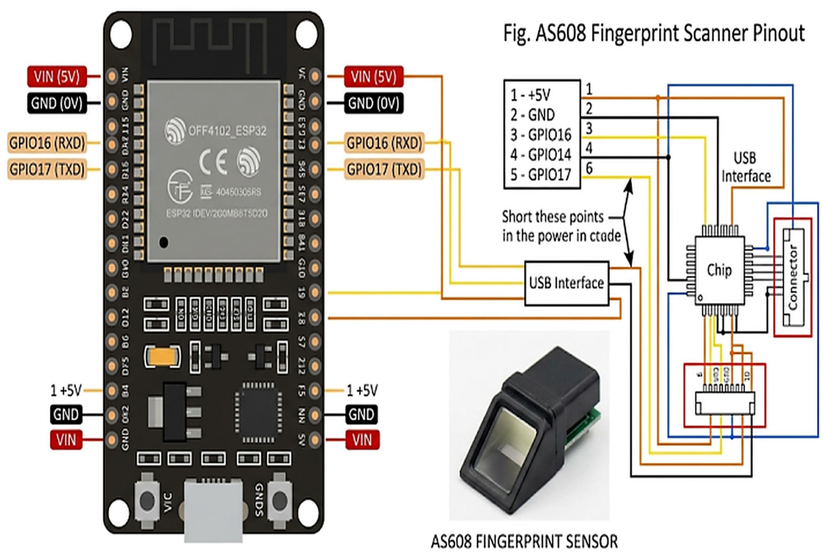

Fig. 6. AS608 Wiring Diagram

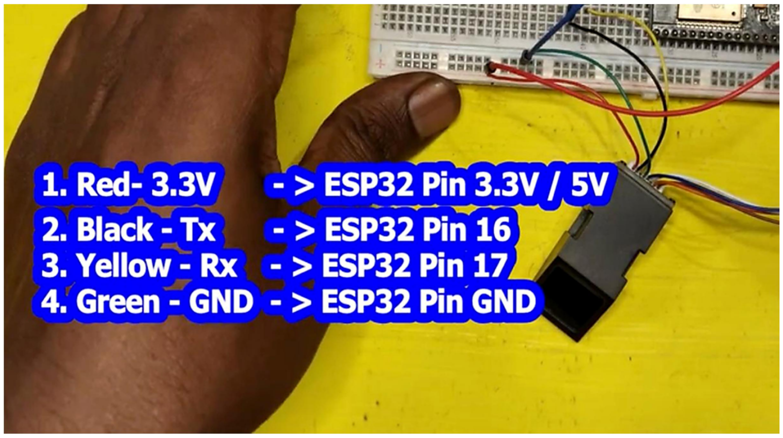

Fig. 7. AS608 Pin Connection

4. RESULTS AND DISCUSSION

The completed system runs as a set of coordinated processes: the Laravel application serving the web interface and REST API, a queue worker processing background jobs, the Reverb WebSocket server pushing live updates, the Python biometric service handling face computation, and the ESP32 devices operating independently on the local network. During testing, both the ESP32-CAM and the AS608 fingerprint module were assembled, flashed, provisioned, and registered through the Laravel admin panel, after which their heartbeat requests updated their status from offline to online within seconds.

The facial recognition pipeline was first validated against the system webcam, confirming that the embedding and matching stages worked correctly before hardware was introduced. With the ESP32-CAM integrated, a successful match against an enrolled student produced an attendance record that appeared on the lecturer's live session view within seconds of capture, broadcast via the Reverb WebSocket channel.

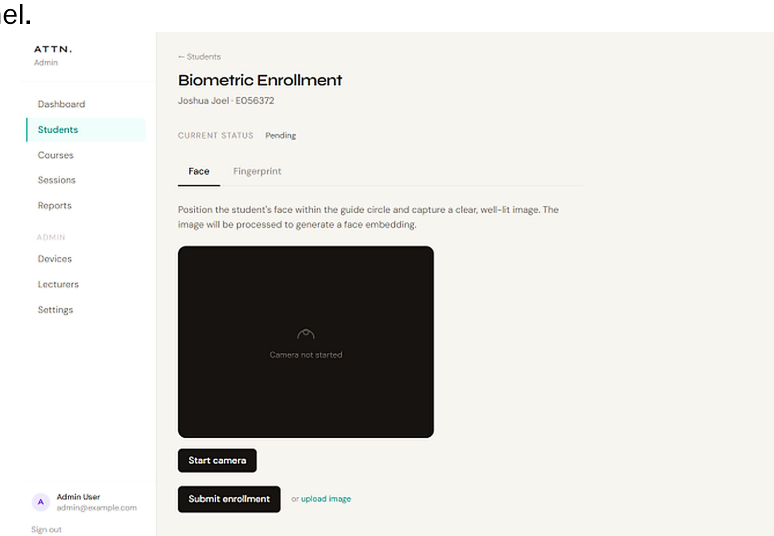

Fig. 8. Biometric Enrolment

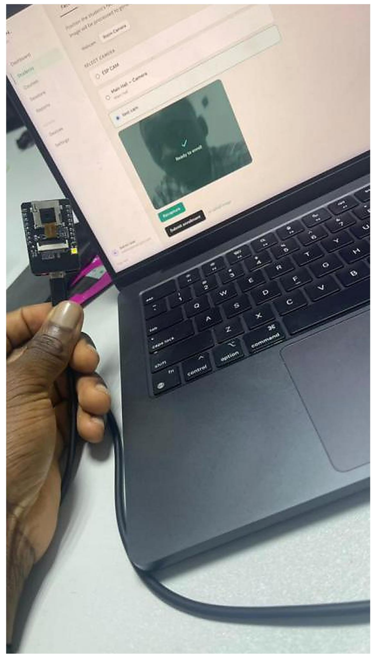

Fig. 9. Facial Enrolment

The fingerprint module was similarly validated, including a test in which the device's network connection was deliberately interrupted; queued scans were replayed and committed correctly once connectivity was restored, without producing duplicate records.

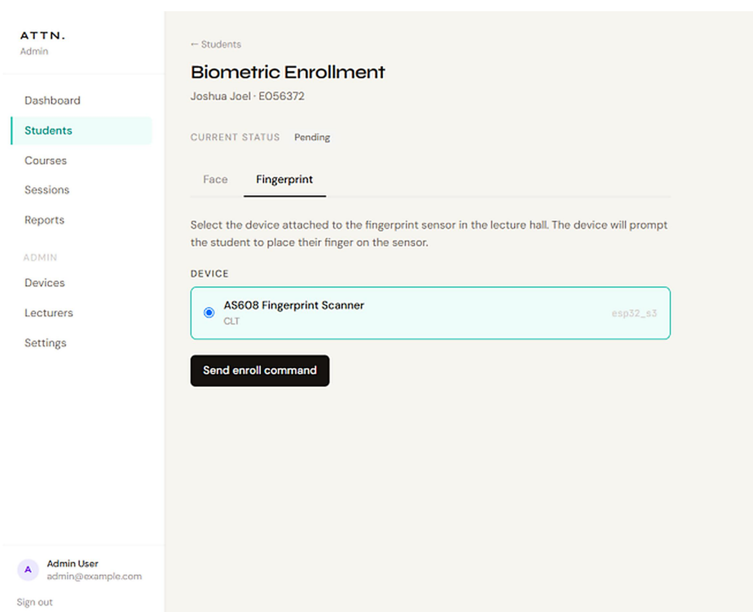

Fig. 10. Fingerprint Enrolment

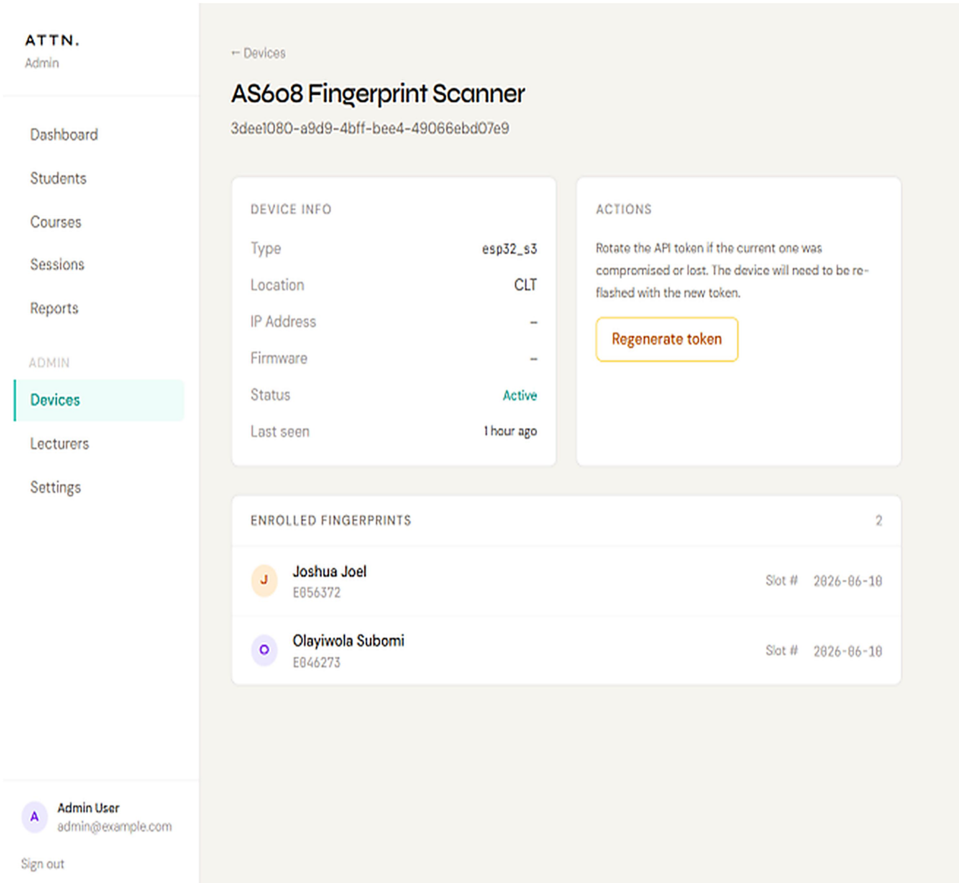

Fig. 11. AS608 Details

Two implementation issues were encountered and resolved during development. First, the originally selected face recognition library produced inconsistent results once real webcam frames were used as input; this was resolved by switching to InsightFace, which handled live camera input reliably. Second, a misconfigured service address initially caused Laravel to report the Python biometric service as unreachable even though it was running and responding to direct requests; correcting the configured address resolved the issue. Both findings reinforce the value of testing against the actual input source and deployment configuration early in development, rather than relying on convenient stand-ins such as sample images or default addresses.

Functional testing covered the Python service's quality scorer and matcher, Laravel's API endpoints for device communication, and full system tests with real hardware. User acceptance testing with lecturers acting as testers did not surface functional defects; feedback led to minor interface adjustments, such as making the manual override control easier to locate and adding an explanatory note when a face match falls into an uncertain confidence range. The flexible enrolment status model, with four states reflecting whether a student has facial, fingerprint, both, or neither biometric registered, together with the manual override option, ensures that a student without one functioning biometric modality is not excluded from attendance marking.

Overall, the implementation meets its stated objectives for a single-machine prototype: dual-biometric verification with OR-based fallback is functional, attendance is logged with method and confidence data in real time, and the web portal supports the core administrative and lecturer workflows. The principal limitation is that the system has been validated on a single development machine with one student authenticating at a time; concurrent authentication across multiple devices and sessions, and behaviour under the load of a 200-plus-student class, were outside the scope of the testing performed and remain open questions for a production deployment.

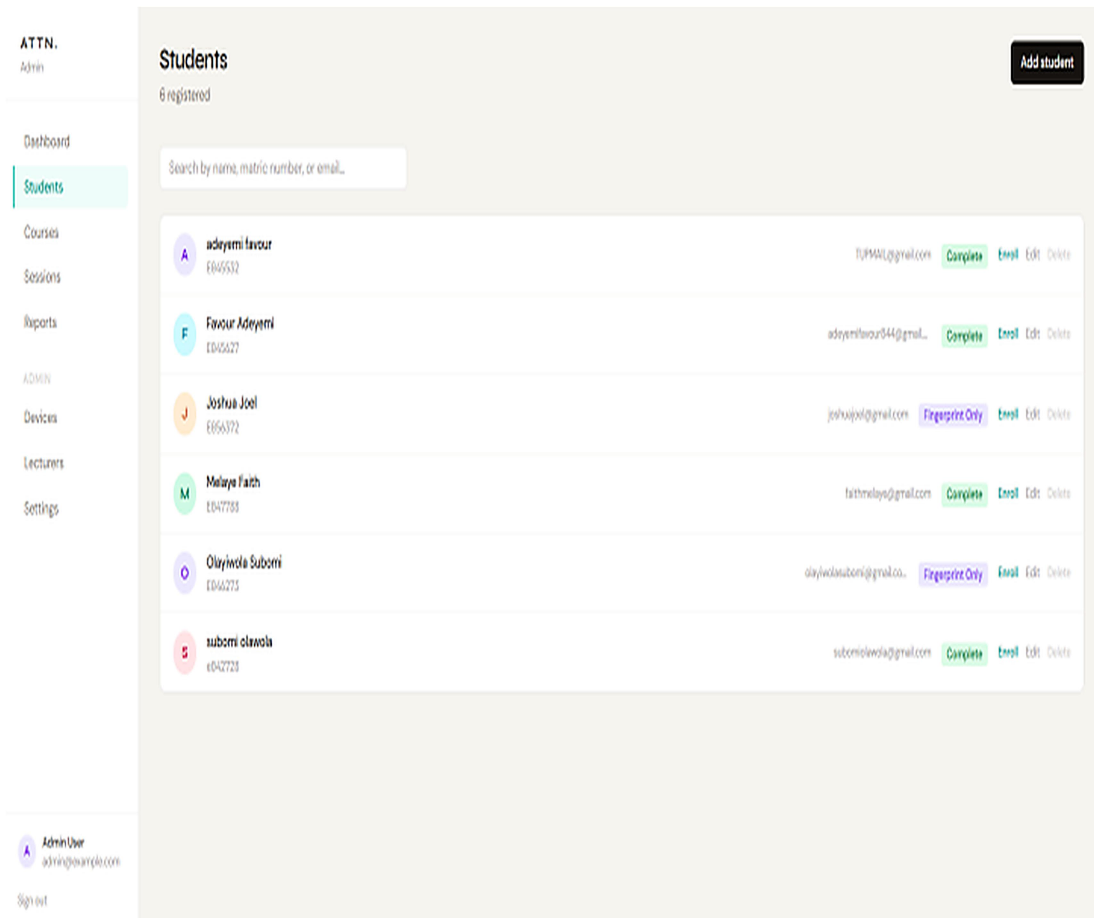

Fig. 12. Enrolled Students

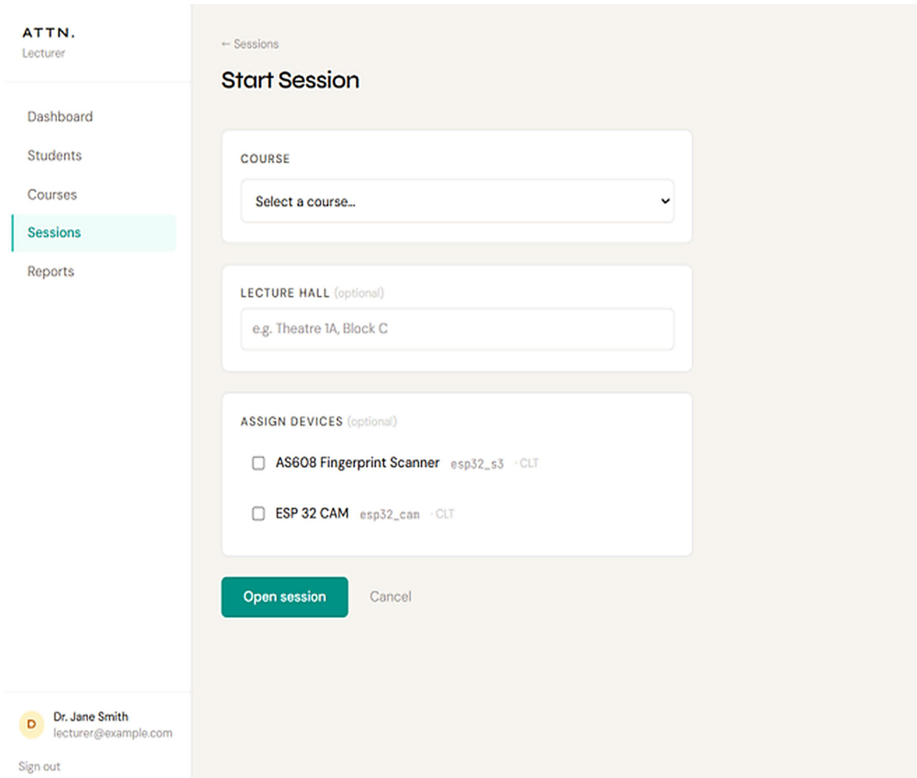

Fig. 13. Start of a Live Class Session

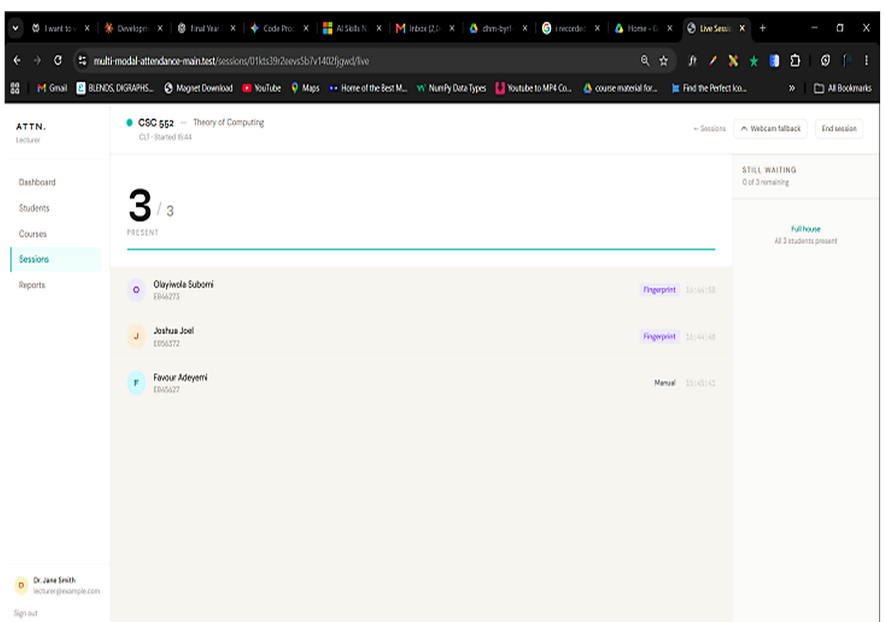

Fig. 14. Recorded Attendance

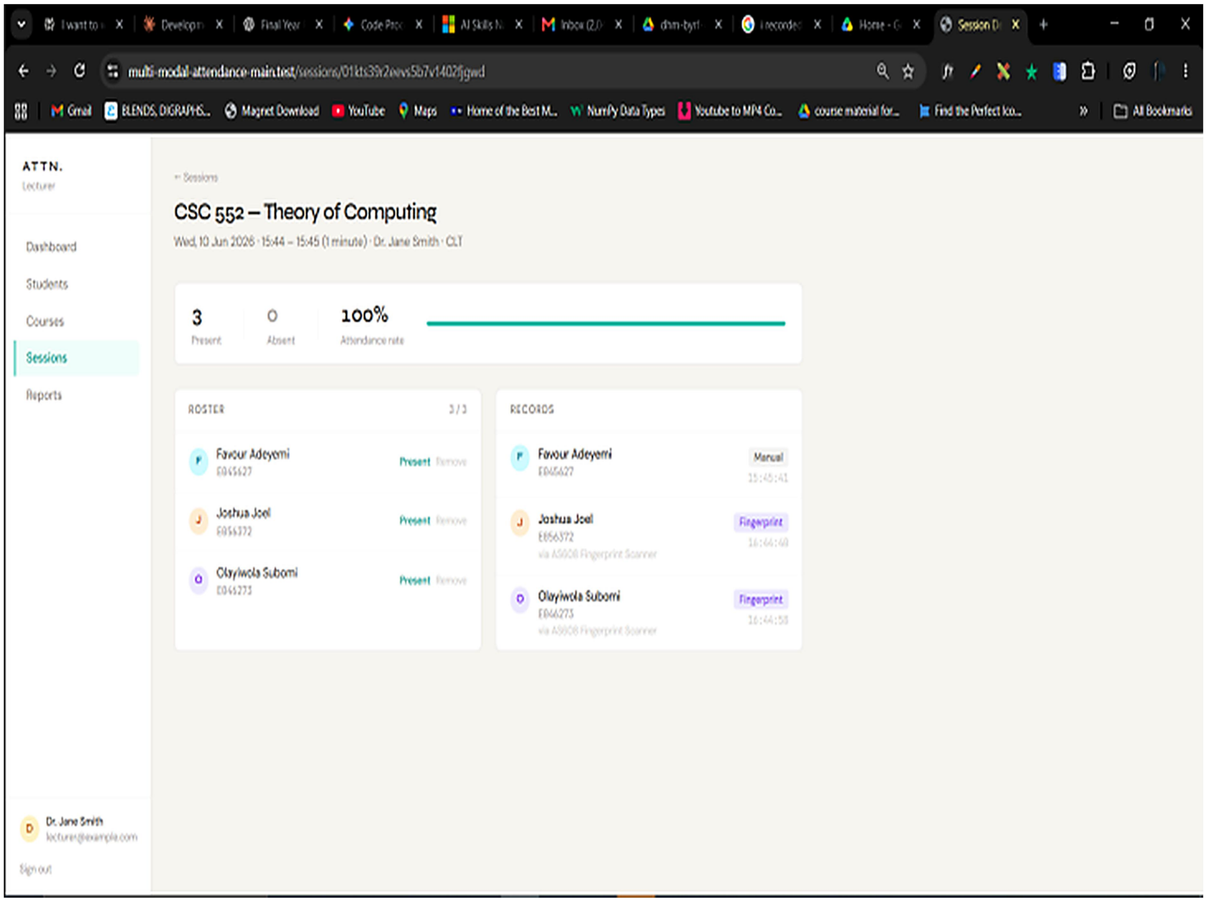

Fig. 15. Attendance Summary

5. SUMMARY, CONCLUSION, AND RECOMMENDATION

This study designed, implemented, and tested an intelligent multimodal attendance and authentication system that combines ESP32-CAM-based facial recognition with AS608 fingerprint verification, coordinated through a Laravel web application and a FastAPI-based biometric service using InsightFace. The system addresses the single-modality vulnerability identified in prior work by allowing either biometric method, or a manual override, to confirm attendance, and by logging each record with its verification method, confidence score, and timestamp for later review. Real hardware testing, including a deliberate network outage for the fingerprint module, confirmed that the offline queue and synchronization mechanism prevents duplicate or stale records, and that attendance updates reach the lecturer's screen within seconds via WebSocket broadcasting. The project met its stated aim and objectives within the scope of a single-machine prototype tested in a single classroom-like setting. The two most time-consuming implementation issues, replacing the initial face recognition library and correcting a service address misconfiguration, illustrate the importance of testing against real input sources and deployment configurations from early in development rather than relying on convenient substitutes.

The following recommendations are proposed for further development, deployment, and research:

- Conduct load testing of the Python matching endpoint with rosters at or beyond the 200-student class size to establish concrete performance bounds.
- Add automated tests for the ESP32 firmware logic that does not depend on physical hardware, particularly the offline queue's batching and idempotency handling.
- Before any institutional pilot, resolve data protection, consent, and biometric re-enrolment policies in line with the Nigeria Data Protection Act and, where applicable, GDPR, and move the Laravel application, Python service, and database off a single development machine to support multiple concurrent devices and sessions.
- Conduct a direct comparison of dlib-based and InsightFace-based embedding pipelines under live camera conditions, with measured accuracy and latency, to generalize the finding from this implementation.

REFERENCES

Abdulkareem, Q. B., Adedotun, K. J., Musa, O. S., & Raji, A. K. (2024). Mitigating attendance fraud through multi-biometric authentication: A comparative study of fingerprint and facial recognition technologies. Journal of Advanced Science and Optimization Research, 6(9).

Abiola, O. A., Okechukwu, R. I., & Akinola, S. O. (2024a). YOLO real-time face mask detection in an automatic door system. Journal of Advances in Mathematics and Computer Science, 12(1), 87–99.

Abiola, O. A., Adeleke, O., & Ajayi, J. O. (2024b). Development of an artificial intelligence body gesture detection dustbin with an autonomous system. Journal of Digital Innovations and Contemporary Research in Science, Engineering and Technology, 12(4), 1–8.

Adedoyin, M. A., Shoewu, O. O., Yussuff, A., Adenowo, A., Shitta, A., & Okedokun, O. (2025). Development of an IoT-based biometric attendance management system. FUOYE Journal of Engineering and Technology, 6(1).

Adeyemi, A. O. (2024). Revolutionizing entry points: Developing an electronic gate system with facial recognition for improved attendance monitoring and access control. International Journal of Renewable Energy and Environmental Sustainability, 9(1), 8–15.

Demir, H. (2023). Class attendance system in education with deep learning and face recognition [Preprint]. arXiv. https://arxiv.org/abs/2309.13317

Kamath, J., Naik, P., M. S., V., Reddy, V. S., & Sadasivam, G. K. (2023). ArcFace: Attendance automation using deep learning. International Journal of Engineering Research & Technology.

Kumar, A., Singh, R., & Verma, S. (2024). Smart attendance system using fingerprint sensor and IoT technology.

Moses, G. E., Kelvin, A. B., & Kenneth, A. P. (2024). Implementation of biometric fingerprint attendance system using ESP8266 and fingerprint scanner. Journal of Scientific and Engineering Research, 11(9), 22–30.

Obiora, I. C., Okoye, M. O., & Adeyanju, O. M. (2025). Development of a fingerprint-based attendance monitoring system. African Journal of Engineering Research, 13(1), 45–58.

Patel, J., Gandhi, S., Katheriya, V., Pataliya, P., & Majumdar, A. (2025). Enhancing classroom attendance systems with face recognition through CCTV using deep learning.

Thalor, M. A., & Gaikwad, O. S. (2023). Facial recognition attendance monitoring system using deep learning techniques. International Journal of Innovative Science and Technology.

Tiwari, P., Mishra, S., & Yadav, A. (2025). An intelligent fingerprint-based attendance system for educational institutions.

Touzene, A., Abed, A. J. W., & Larabi, S. (2024). An embedded intelligent system for attendance monitoring [Preprint]. arXiv. https://arxiv.org/abs/2406.13694