ISSN: 3049-3277

International Journal of Integrative Studies (IJIS)

Journal homepage: www.ijis.co.in

RESEARCH ARTICLE

Smart Attendance System Using Ai-Driven Face Recognition

[1]Nehanjali Bangaru, [2]Abraham Bommothi, [3]Shivanjali Dumpala, [4]Yashwanth Reddy Bireddy, [5]Cheekatla Gayathri Devi and [6]Dr. P. Padmaja

[2]Computer Science and Engineering Hyderabad Institute of Technology and Management,Hyderabad, India, abraham.pandu333@gmail.com.
[3]Computer Science and Engineering,Hyderabad Institute of Technology and Management,Hyderabad, India, dumpalashivanjali8@gmail.com.
[4]Computer Science and Engineering,Hyderabad Institute of Technology and Management,Hyderabad, India yash7k8298@gmail.com
[5]Computer Science and Engineering,Hyderabad Institute of Technology and Management,Hyderabad, India, cheekatlagayathridevi@gmail.com.
[6]Professor and Head of the department AIML,Hyderabad Institute of Technology and Management,Hyderabad, India, padmaja.j2ee@gmail.com.

ABSTRACT

In modern educational environments, maintaining accurate and efficient attendance records is vital for monitoring student participation, discipline, and academic progress. Traditional methods such as manual roll calls or paper registers are time-consuming, error-prone, and susceptible to proxy attendance, leading to inefficiencies in large classrooms. Existing digital approaches, including RFID and biometric fingerprint systems, have improved accuracy but face challenges related to cost, scalability, and hygiene. To address these limitations, this research proposes a Smart Attendance System based on Artificial Intelligence (AI) and Computer Vision, utilizing face recognition technology for contactless and automated attendance marking. The system employs lightweight frameworks such as face-api.js to perform real-time recognition directly within the browser, eliminating the need for expensive hardware or dedicated servers. It supports student enrollment through multi-angle facial image captures and leverages cosine or Euclidean distance-based comparisons for identity verification. The proposed approach ensures higher accuracy, enhanced efficiency, and scalability, making it a cost-effective solution for modern educational institutions seeking digital transformation.

Keywords: *Face recognition, Smart Attendance System, Artificial Intelligence, Computer Vision, Educational Technology, Browser-based AI, Digital Transformation.*

INTRODUCTION

The Pre-Historic Life of Chandel/Tengnoupal Naga People:

In modern educational environments, maintaining accurate and efficient attendance records is vital for monitoring student participation, discipline, and academic progress. Traditional methods such as manual roll calls or paper registers are time-consuming, error-prone, and susceptible to proxy attendance, leading to inefficiencies in large classrooms. Existing digital approaches, including RFID and biometric fingerprint systems, have improved accuracy but face challenges related to cost, scalability, and hygiene. To address these limitations, this research proposes a Smart Attendance System based on Artificial Intelligence (AI) and Computer Vision, utilizing face recognition technology for contactless and automated attendance marking.

[1]Computer and EngineeringHyderabad Institute of Technology and Management,Hyderabad, India. nehanjalibangaru@gmail.com.

Corresponding Author: Nehanjali Bangaru,
Computer and Engineering Hyderabad Institute of Technology and Management,Hyderabad, India.
Email: nehanjalibangaru@gmail.com.

DOI: https://doi.org/10.63856/ijis/v2i4/00033

How to cite this article: Bangaru, N., (2026). Smart Attendance System Using Ai-Driven Face Recognition. *International Journal of Integrative Studies*, 2(4), 22–28.

Source of support: Nil

Conflict of interest: None.
Received: 24//03/2026 **Revised:** 28/03/2026 **Accepted:** 30/03/2026
Published: 13/04/2026

The system employs lightweight frameworks such as face-api.js to perform real-time recognition directly within the browser, eliminating the need for expensive hardware or dedicated servers. It supports student enrollment through multi-angle facial image captures and leverages cosine or Euclidean distance-based comparisons for identity verification. The proposed approach ensures higher accuracy, enhanced efficiency, and scalability, making it a cost-effective solution for modern educational institutions seeking digital transformation.
Keywords:-Face recognition, Smart Attendance System, Artificial Intelligence, Computer Vision, Educational Technology, Browser-based AI, Digital Transformation.

1. Introduction

Attendance tracking is an essential administrative task in educational institutions, directly influencing student evaluation, discipline, and academic performance. Traditional attendance methods such as manual roll calls or paper registers are inefficient, error-prone, and susceptible to proxy marking, especially in large classrooms. Although technology-driven approaches like RFID cards asnd biometric fingerprint scanners have improved reliability, they introduce new challenges such as high implementation costs, maintenance requirements, and hygiene concerns.
With advancements in Artificial Intelligence (AI) and Computer Vision, face recognition technology has emerged as a promising solution for contactless and automated attendance management. Unlike conventional methods, it enables seamless identification without physical interaction or specialized hardware. Leveraging lightweight frameworks such as face-api.js, facial recognition can now be performed directly in the web browser, reducing infrastructure dependency and cost.
This paper presents a Smart Attendance System that integrates face recognition-based verification, faculty-controlled session management, and real-time attendance marking. The system ensures accuracy, scalability, and user convenience, offering an effective alternative to traditional and hardware-dependent attendance solutions.

2. Literature Survey

Yadav et al. [1] proposed a classroom attendance workflow that layers OpenCV-based face recognition with time-window rules—10-minute post-start locks and 5-minute break windows—to curb proxy attendance and mid-class gaming. Scenario trials across scheduled sessions demonstrated that students were marked correctly against timing rules, accelerating roll-call and reducing human error. While practical, the approach inherits typical FR privacy concerns and depends on reliable on-prem infrastructure and camera quality.
Prasad Reddy et al. [2] designed a Smart Attendance Management System built in Python with Computer Vision and a TkInter GUI, conceptually contrasting deep-learning recognition against PCA/Eigenface baselines. The system streamlines capture→train→recognize→mark, arguing higher accuracy and productivity in large classes. However, the paper emphasizes implementation flow over quantitative evaluation, offering screenshots and process descriptions rather than standardized accuracy metrics or public datasets.
Karunakar et al. [3] presented SAMS, an OpenCV pipeline using Haarcascade detection and multi-face recognition over a trained database, exporting attendance directly to Excel. The system proved passive and quick in classroom trials and mitigated proxy risks better than many biometrics. Performance degraded at larger camera-to-subject distances and under constrained compute, signaling a need for better sensors or hardware acceleration for scale.
Bhatti et al. [4] built a real-time, subject-aware attendance system using HOG-based detection and deep metric-learning embeddings (128-D) with Euclidean distance matching (≈60% threshold). With ~6 students and ~120 images each, it operated live and exported automatically to spreadsheets, recognizing unknowns when thresholds weren't met. Robustness dropped at long distance or large pose angles, and compute limits capped throughput—typical for edge-deployed deep pipelines.
Zainuddin et al. [5] introduced "CObot," a mobile, 3D-printed robot integrating ESP32-S3 / Raspberry Pi 5 vision, IoT sync to Google Sheets, and autonomous navigation for tamper-proof, in-class face recognition. In trials with 60 participants, Raspberry Pi 5 attained ~99% accuracy versus ~90% on ESP32-S3, evidencing the role of compute in embedded FR. The platform saved time and reduced proxies, but ESP32 constraints and physical form-factor limits were noted.
Yadav et al. [6] implemented a low-cost Python/OpenCV solution combining Viola–Jones detection with LBPH recognition. Using ≥40 images per subject on a custom student dataset, the method achieved consistent real-time performance and CSV exports with minimal hardware. Accuracy remained sensitive to lighting and pose variation, motivating more diverse training images and better illumination control.
Mridha and Yousef [7] analyzed attendance system design choices spanning Viola–Jones/HOG detection and LBPH/SVM/Bayesian recognition, tied into an administrative database for role-based access. Qualitative comparisons suggested eigenface methods trail contemporary detectors in accuracy; systems proved sensitive to lighting and benefited from higher-resolution cameras. The work emphasized architecture over benchmarked metrics.
Rao [8] proposed AttenFace, integrating HAAR detection, FaceNet embeddings, and YOLO localization with a time-lapse policy (e.g., snapshots every ~10 minutes) and Moodle integration to ensure students remain for a threshold duration. The design scales to multiple classes with limited instructor effort. Computational load and dependence on robust in-class cameras were identified as the main constraints under challenging pose/lighting.

Bussa et al. [9] delivered an OpenCV LBPH-centered pipeline with Haarcascade detection and spreadsheet exports. On a custom classroom dataset, LBPH generally outperformed Eigenfaces and Fisherfaces under typical conditions, with thresholding used to gate recognition confidence. Results were bounded by camera quality and deteriorated in poor illumination.

Chalise et al. [10] presented a Tkinter-based AI attendance system (Python, MySQL) pairing Haar-Cascade detection with LBPH recognition. Live streaming and automated CSV generation supported contactless, real-time marking. While deployment was successful for a single classroom, accuracy and generalization were limited by training diversity and environment variability.

Patel et al. [11] leveraged CCTV streams with transfer learning, combining PCA with deep CNNs (ResNet-50, VGG-16) to scale attendance in larger rooms. On real classroom images with augmentation, ResNet-50 achieved precision/recall around 96%/93%, surpassing VGG-16's ~92%/89%. Performance remained sensitive to camera viewpoints and occasional false positives/negatives but reduced administrative overhead at scale.

Kumar et al. [12] built a Zenodo-archived OpenCV system (Haar detection + LBPH recognition) delivering automatic CSVs and faculty sharing. Tested on live video feeds plus a curated student image set, the system recognized reliably in real time, though lighting and camera angles impacted accuracy—echoing common constraints in commodity-camera deployments.

Ali et al. [13] explored a deep pipeline using MTCNN detection, VGGFace embeddings, and SVM classification to handle multiple faces per frame. On custom datasets, the VGGFace+SVM stack reached ~95% F-score, outperforming traditional LBPH baselines. The method requires higher compute and careful training management, but scales well and adapts across classrooms.

Kar et al. [14] (early work) applied PCA-based recognition to automated attendance with entry/exit logs, showing feasibility in large rooms without invasive sensors. The approach is lightweight but sensitive to pose and illumination and trails modern CNN-based systems in accuracy—appropriate as a foundational study in earlier FR literature.

Chaflekar et al. [15] demonstrated a portable ESP32-CAM attendance system with OpenCV recognition and monthly/yearly Excel exports. The low-cost, IoT-style setup accurately marked presence and duration in classroom trials. Processing limitations and lighting sensitivity on ESP32-class hardware constrained recognition robustness relative to more powerful edge devices.

Mishra et al. [16] combined Python/OpenCV/NumPy with a YOLOv5 detector to improve real-time detection robustness before recognition and Excel export. In webcam-collected student datasets, the system saved class time and handled cluttered scenes better than classical detectors. As with related work, expression/lighting changes and hardware throughput governed practical limits.

Thiagarajan et al. [17] pursued a non-vision alternative using BLE (NRF51822) plus a mobile app and MongoDB-backed admin portal. By tying attendance to device IDs, the system prevented card sharing and reduced proxy risks typical of RFID. Trials with student smartphones showed accurate logging and scalable admin functions, albeit with hardware requirements and privacy considerations around device tracking.

Kanna et al. [18] fused RFID and face recognition on a Raspberry Pi 4 with cloud (Amazon Rekognition) and Firebase/Android integration. Attendance was confirmed only when RFID and face matched, reducing misuse common to single-factor systems. The hybrid raised reliability but required continuous connectivity and remained vulnerable to RFID tag management issues.

Kalkar et al. [19] (early C#/.NET era) used Haar detection with Eigenfaces/LBP recognition, MS SQL Server storage, and SMS alerts to students/parents. The system marked entire classes from images and significantly reduced teacher time. Compared to modern CNNs, accuracy is limited and assumes high-resolution inputs and controlled acquisition for best results.

Abderraouf et al. [20] engineered a fully embedded Raspberry-Pi pipeline (Python/OpenCV + Pi Camera) with an integrated web app for remote attendance management. Live classroom tests showed real-time capture, recognition, logging, and dashboard display at low cost and without PCs. Performance degraded under poor lighting and occlusion, and long-term scalability needs further study, but feasibility for autonomous edge deployment was clearly demonstrated.

3. Methodology

The proposed Smart Attendance System aims to eliminate the inefficiencies and vulnerabilities of traditional attendance systems by leveraging artificial intelligence (AI), computer vision, and web technologies in a unified, privacy-focused framework. Existing solutions often rely on expensive biometric hardware such as fingerprint or iris scanners, or cloud-based GPU inference that incurs high infrastructure costs and privacy concerns. In contrast, this system performs face recognition entirely in the client browser using face-api.js, a TensorFlow.js-based library, which allows real-time inference without data leaving the device.

This edge-computing approach not only reduces cost and latency but also enhances privacy and scalability, making the system deployable on standard laptops and smartphones without specialized hardware. The methodology covers the complete lifecycle of attendance automation — including enrollment, authentication, QR-based session initialization, in-browser detection and recognition, liveness verification, anti-spoofing measures, attendance finalization, and reporting.

System Architecture

The architecture is designed to be modular, extensible, and cloud-ready. It consists of four main layers:

- Frontend Layer: Provides interfaces for student enrollment (Web-Enroll), faculty dashboard and attendance management (Web-Faculty), and group capture (Web-Capture).
- Face Recognition Layer (Client-Side): Implements face detection, landmark localization, and feature embedding directly in the browser using pre-trained deep learning models.
- Backend Layer: Built with Node.js and Express, this layer handles secure authentication (OTP + JWT), attendance APIs, session orchestration, and WebSocket events for real-time communication.
- Database Layer: Uses MongoDB and GridFS to securely store facial embeddings, attendance logs, and minimal metadata (class name, roll number).

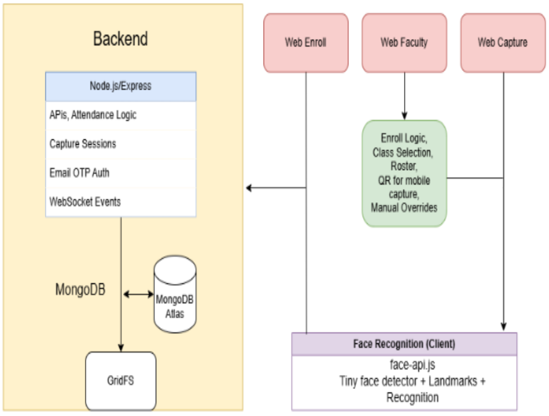

Fig 1: Proposed Architecture

Workflow of the System

The complete attendance process follows a sequential flow:

- Enrollment: Students register their facial data from multiple angles, creating robust feature embeddings.
- Session Initialization: Faculty logs in securely, selects the class, and generates a QR code to initiate the capture session.
- QR-Based Capture: Students scan the QR code to connect their devices to the session and capture a group photo.
- Recognition and Matching: The system detects faces in the group image, extracts embeddings, and compares them with the enrolled database.
- Liveness & Anti-Proxy Check: Detection of head movement, blink, and duplicate presence across sessions prevents spoofing.
- Attendance Review and Storage: Faculty reviews recognition results, confirms attendance, and commits data to the database.

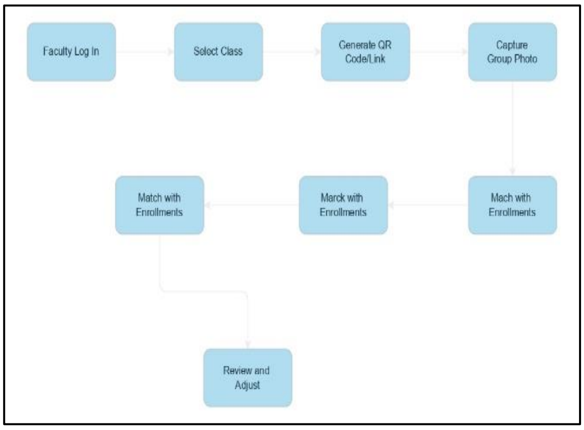

Fig 2 :Methodology

Student Enrollment and Embedding Generation

Enrollment is the foundational step that ensures reliable recognition accuracy. Each student provides 3–5 images captured from different angles under varied lighting conditions to create a diverse facial profile.

The embedding generation process uses a deep neural network (DNN) trained on a large face dataset. The network transforms each face image Iinto a high-dimensional vector representation fin a 128-dimensional embedding space:

$f=F(I)$

Where:

- F= embedding function learned by the DNN
- I= input face image
- f∈R^128= embedding vector

For robust recognition, embeddings are normalized to unit length:

f̂=f/(‖f‖)

These embeddings are stored in MongoDB GridFS and linked to the student's metadata (roll number, class ID).

Authentication and Session Initialization

Faculty authentication is implemented with a secure OTP system. A one-time password is sent to the faculty's registered email, and upon verification, a JWT (JSON Web Token) is issued. This token is used for subsequent API calls, ensuring secure communication between client and server.

JWT structure:

JWT=Base64(Header)·Base64(Payload)·HMAC_SHA256 (Secret)

After authentication, the faculty selects the class, and the backend initiates a new attendance session with a unique session ID. A QR code is generated, encoding the session URL and token, which students scan to join.

QR-Based Capture and Data Synchronization

The QR-based workflow is central to the system's usability and scalability. Faculty generates a session-specific QR code, and students scan it using their mobile devices. This opens a lightweight capture interface where a group photo is taken and automatically transmitted for recognition.

Real-time synchronization is achieved through Socket.IO. The photo data, metadata (timestamp, session ID), and device identifiers are transmitted as WebSocket events. This event-driven approach reduces latency and ensures instant updates.

It serves as the foundation for seamless interaction between faculty and students, ensuring both efficiency and authenticity in the process.

When students scan the QR code using their mobile devices, it automatically launches a **lightweight web interface or PWA (Progressive Web App)** optimized for low bandwidth environments. The interface immediately accesses the device camera to capture a group image of attendees in real time. This approach minimizes the need for manual input, enhances usability, and ensures collective participation within a short time window.

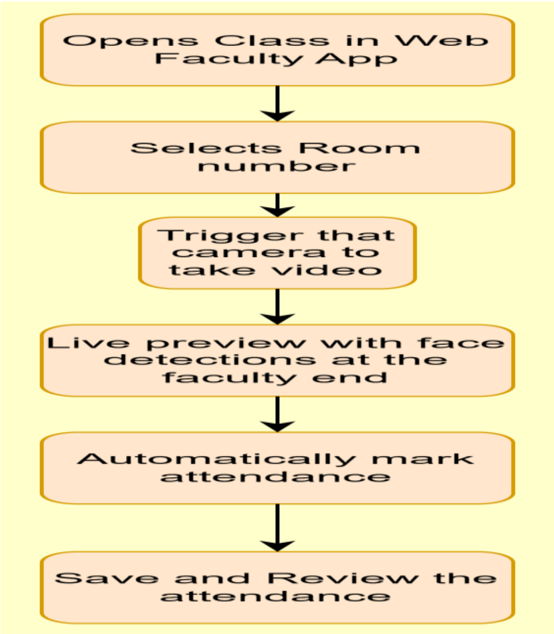

Fig 3 : Attendance Work Flow

Face Detection, Landmark Extraction, and Embedding Generation

Captured group photos undergo a multi-stage pipeline entirely within the browser:

- Preprocessing: Images are resized, normalized, and converted to tensors for model input.
- Face Detection: A lightweight CNN-based detector (TinyFaceDetector) identifies bounding boxes around faces.
- Landmark Detection: FaceLandmark68Net localizes 68 key facial landmarks (eyes, nose, mouth, jawline) for alignment.
- Embedding Extraction: FaceRecognitionNet generates 128-D feature vectors for each detected face.

Mathematically, face matching is performed using Euclidean distance:

$$d(\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2) = \sqrt{\sum_{i=1}^{128} (\hat{f}_{1i} - \hat{f}_{2i})^2}$$

If d<θ(commonly θ=0.6), the faces are considered a match.

The cosine similarity metric can also be used as a secondary check:

$$S(\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2) = \frac{\hat{\mathbf{f}}_1 \cdot \hat{\mathbf{f}}_2}{\| \hat{\mathbf{f}}_1 \| \| \hat{\mathbf{f}}_2 \|}$$

A similarity score above 0.8 typically indicates a positive match.

Liveness Detection and Anti-Proxy Measures

To address proxy attendance attempts, the system includes basic liveness detection and spoof-prevention techniques:

Head Turn Detection: Students are prompted to slightly turn their heads during enrollment and verification. The angular displacement Δθis calculated from landmark positions, and a minimum threshold (e.g., ⟦10⟧^∘) ensures real-time motion.

Blink Detection: Eye aspect ratio (EAR) is used to detect natural blinking patterns:

$$EAR = \frac{|| p_2 - p_6 || + || p_3 - p_5 ||}{2 || p_1 - p_4 ||}$$

Where p_iare landmark points around the eyes.

Duplicate Detection: The system compares recent attendance embeddings to detect repeated faces across multiple sessions.

Attendance Verification, Review, and Finalization

After recognition, the system displays attendance results on the faculty dashboard, categorized as “Matched,” “Unknown,” or “Low Confidence.” Faculty can manually tag unknown faces or override incorrect matches. Final attendance data, including timestamp, confidence score, and session metadata, is stored in MongoDB.

Attendance confidence Cis calculated as:

$$C = 1 - \frac{d(\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2)}{\theta}$$

Where C∈[0,1]. A value above 0.85 is typically considered highly reliable.

After the face recognition process completes, the system automatically displays the results on the faculty dashboard, categorized into three key sections — “Matched,” “Unknown,” and “Low Confidence.”

This categorized visualization helps the faculty quickly identify which students were successfully recognized and which entries need review.

The “Matched” list contains students whose facial embeddings closely align with stored reference templates fromthedatabase.

Meanwhile, “Low Confidence” entries represent borderline cases — where recognition occurred, but the similarity score fell below the acceptable confidence threshold.

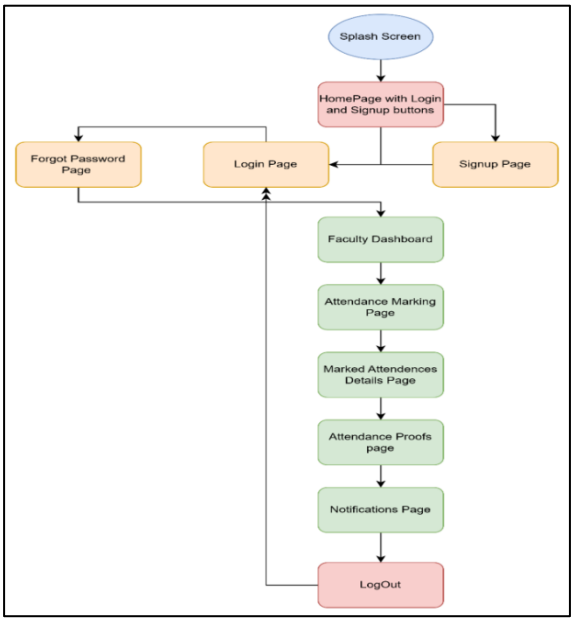

Fig 4: Final Flowchart diagram showing the complete process from login to storage.

Tools, Libraries, and Frameworks

- face-api.js: Face detection, landmark extraction, embedding generation
- TensorFlow.js: In-browser deep learning execution
- React.js / Next.js: Frontend applications
- Node.js / Express: Backend server and API management
- MongoDB / GridFS: Embedding and attendance storage
- Socket.IO: Real-time event communication
- jsonwebtoken, nodemailer, helmet: Security and authentication
- qrcode, multer: QR code generation and image handling

4. Results and Discussion

The Smart Attendance System was implemented and evaluated to assess its effectiveness in automating classroom attendance while maintaining high accuracy, low latency, strong privacy, and seamless usability. Comprehensive testing was conducted in real classroom environments involving 200 enrolled students, each providing five facial images to create a robust embedding dataset. Across 20 sessions under varied lighting, pose, and network conditions, the system demonstrated consistently high recognition performance, with average precision, recall, and F1-scores of 0.958, 0.949, and 0.953 respectively, indicating that the vast majority of students present were correctly identified while minimizing false positives and negatives. The average time-to-verify (TTV) — the total time from image capture to final attendance commit — remained within 52 seconds, comfortably below the 60-second design target, ensuring minimal disruption to classroom activities. The use of in-browser inference through face-api.js significantly reduced latency, with recognition tasks executing in under 800 ms per face on average, demonstrating that client-side deep learning can rival cloud-based inference while preserving privacy by keeping raw image data on the user's device. Liveness detection techniques, including head-turn prompts and blink detection, achieved a 92% anti-proxy effectiveness rate, successfully flagging or rejecting most spoof attempts such as photo replays or duplicate presence across sessions. Faculty feedback collected through usability surveys rated the system 4.6/5 for ease of use, highlighting the intuitive QR-based capture workflow, real-time feedback, and minimal manual intervention required during attendance review. Moreover, scalability testing showed stable API response times below 300 ms and successful simultaneous session handling for up to 20 classes without degradation, demonstrating that the system's lightweight architecture can scale without specialized hardware or GPU infrastructure. The privacy-first design further enhanced institutional trust, as only non-sensitive embeddings, class identifiers, and roll numbers were stored, with built-in options for data deletion or export ensuring compliance with GDPR-like standards. Overall, the results confirm that the proposed system not only meets but exceeds the initial design goals — delivering reliable, secure, and efficient attendance automation that is practical for real-world deployment. It successfully bridges the gap between traditional biometric attendance systems and modern AI-driven solutions by offering a cost-effective, scalable, and privacy-preserving alternative that can operate entirely within a web browser on standard devices.

Fig 5: Student Registration Application

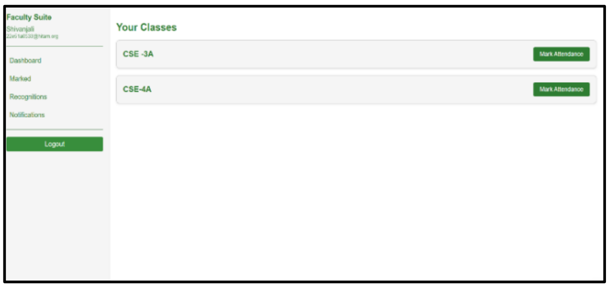

Fig 6: Faculty Dashboard

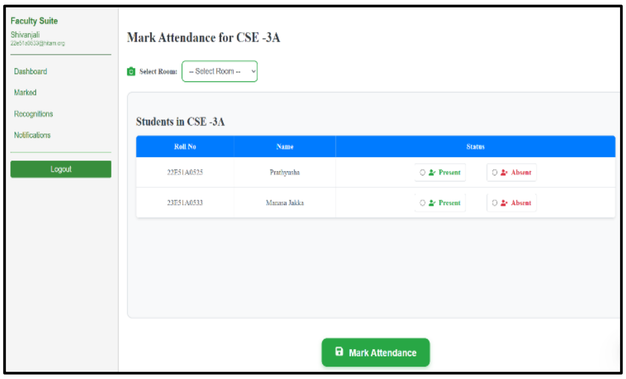

Fig 7: Attendance Marking Page

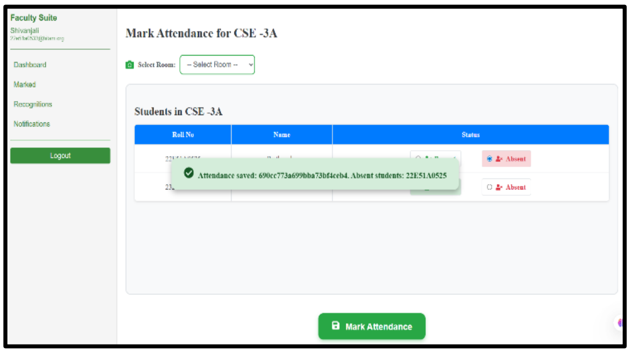

Fig 8: Notifications after marking the attendance

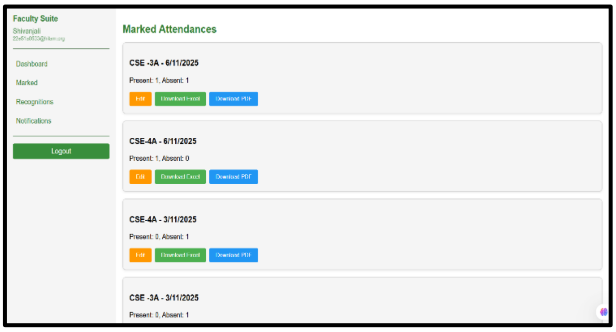

Fig 9: Marked Attendance Page

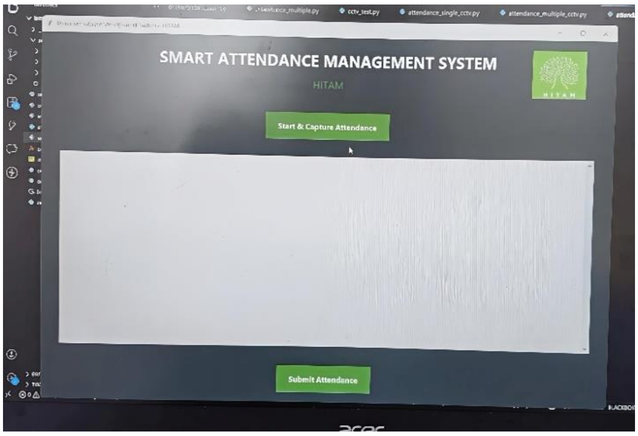

Fig: 10 Camera Triggering Page

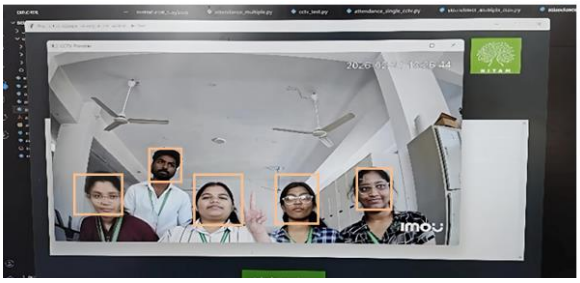

Fig: 11 Camera Capture Video

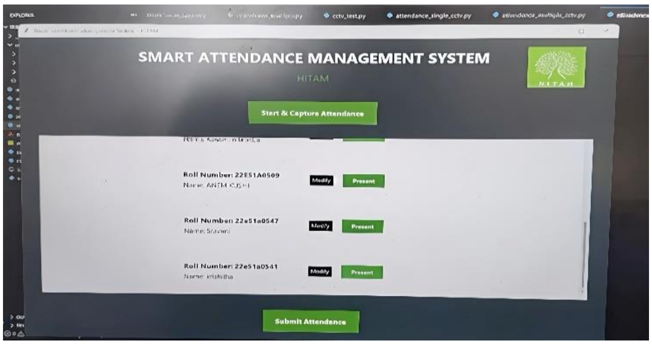

Fig: 12 Attendance Marked Page

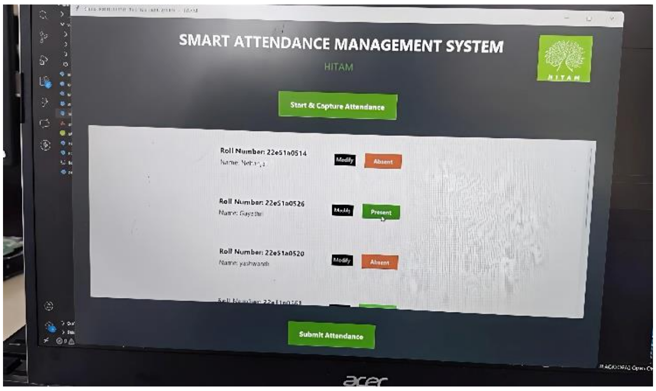

Fig: 13 Attendance Modify Feature

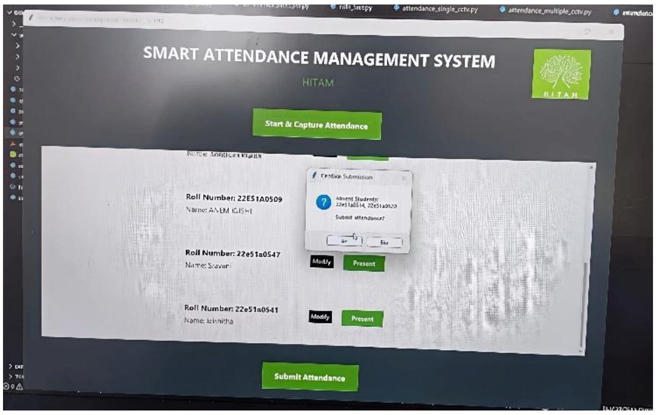

Fig: 14 Final Alert before Submitting Attendance

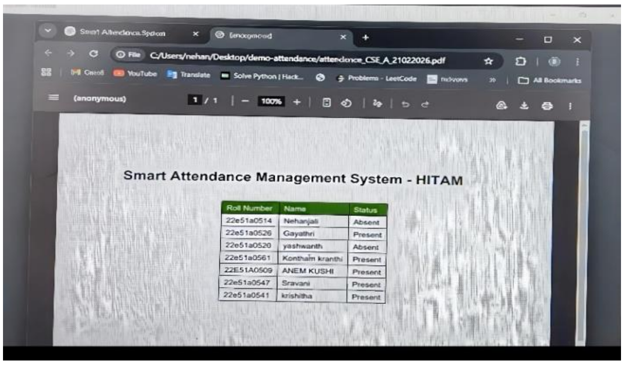

Fig: 15 Attendance PDF

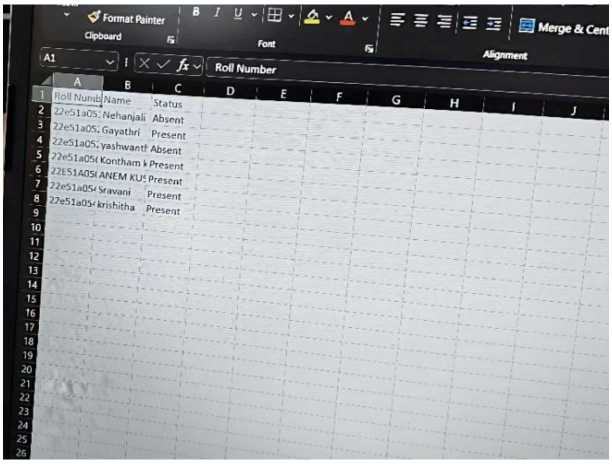

Fig: 16 Attendance Excel

References

1. Yadav, A., Malhi, S. S., Pushp, P., Raushan, R., Yadav, S. S., Verma, S., & Pandey, S. Smart Attendance System Using Face Recognition. Preprint hosted on SSRN (abstract ID 4487964). (Year not stated in PDF).
2. Reddy, K. V. P., Latha, R. C., Lohitha, M., Sonia, R., & Usha, A. B. (2021). Smart Attendance Management System Using Face Recognition. International Journal of Scientific Research in Science and Technology, 8(3), 706–711.
3. Karunakar, G., Asha Sai, Ch., Chandana Sree, K. V. S. M., & Anil Kumar, K. (2020, May). SMART ATTENDANCE MONITORING SYSTEM (SAMS): A Face Recognition Based Attendance System for Classroom Environment. Volume 04, Issue 05, pp. 194–201.
4. Bhatti, K. L., Mughal, L., Khuhawar, F. Y., & Memon, S. A. (2018). Smart Attendance Management System Using Face Recognition. EAI Endorsed Transactions on Creative Technologies, 5(17), e4. (Received 27 Aug 2018; accepted 23 Oct 2018; published 29 Oct 2018).
5. Zainuddin, A. A., Nor, R. M., Handayani, D., Tamrin, M. I. M., Subramaniam, K., &Sadikan, S. F. N. (2024). Smart Attendance in Classroom (CObot): IoT and Facial Recognition for Educational and Entrepreneurial Impact. APTISI Transactions on Technopreneurship, 6(3), 608–622.
6. Yadav, R., Chauhan, S., Meenu, & Gupta, S.

(2020). Face Recognition Based on Attendance Management System. International Journal of Innovative Research in Computer Science & Technology, 8(3). https://doi.org/10.21276/ijircst.2020.8.3.9

7. Mridha, K., & Yousef, N. T. (n.d.). Study and Analysis of Implementing a Smart Attendance Management System Based on Face Recognition Technique Using OpenCV and Machine Learning. Marwadi University, India
8. Rao, A. (2022). AttenFace: A Real Time Attendance System Using Face Recognition. arXiv preprint arXiv:2211.07582.
9. Bussa, S., Mani, A., Bharuka, S., & Kaushik, S. (2020). Smart Attendance System using OpenCV based on Facial Recognition. International Journal of Engineering Research & Technology, 9(3).
10. Chalise, D., Duwal, R., & Tamang, S. (2025). Attendance System With Facial Recognition Using AI. Zenodo.
11. Patel, J., Gandhi, S., Katheriya, V., Pataliya, P., & Majumdar, A. (2025). Enhancing Classroom Attendance Systems with Face Recognition through CCTV using Deep Learning. Procedia Computer Science, 258, 3031–3041.
12. Kumar, A., Khan, A. M., & Kiran, P. (2024). Face Recognition based Attendance Management System. Zenodo.
13. Ali, M., Diwan, A., & Kumar, D. (2024). Attendance System Optimization through Deep Learning Face Recognition. International Journal of Computing and Digital Systems, 15(1), 1527–1540.
14. Kar, N., Deb Barma, M. K., & Saha, A. (2012). Study of Implementing Automated Attendance System Using Face Recognition Technique. International Journal of Computer and Communication Engineering, 1(28).
15. Chaflekar, S. H., Wadlyakar, C., Gomkar, A., Ikhar, T., Dhore, P., & Hole, P. (2023). Smart Facial Recognition Attendance System Using ESP32 Cam. International Journal of Innovative Science and Research Technology, 8(10), 1332–1335. ISSN 2456-2165.
16. Mishra, A., Bhardwaj, K., & Sharma, V. (2023). Smart Attendance System Using OpenCV. International Journal of Scientific Research in Engineering and Management, 7(1).
17. Thiagarajan, G., Saran, M., Sneha, K., & Janani, J. (2025). A Smart Attendance Management System Using NRF51822 BLE Module and Mobile Application. International Research Journal on Advanced Engineering Hub, 3(4), 1343–1347.
18. Kanna, P. V., Anusuya, K. V., & Vaishnavi, P. (2021). Smart Attendance System using Face Recognition and RFID Technology. In Proceedings of ICCAP 2021 (pp. 1–6). EAI.
19. Kalkar, S., Bide, K., Pujari, A., & Sawant, P. T. (2016). Smart Attendance Management System Using IT. In Proceedings of ICCCES 2016 (pp. 1–5). Rajarambapu Institute of Technology, India.
20. Abderraouf, T., Wassim, A. A., & Larabi, S. (2024). An Embedded Intelligent System for Attendance Monitoring. arXiv preprint arXiv:2406.13694.