

AttendIQ: A Web-Based Contactless Academic Attendance System Using Machine Learning Algorithm

[Prof. Chandrapal U. Chauhan, Shreyash Sahare, Rutvik Kshirsagar, Anand Potdukhe, Kajal Durge, Harsh Meshram]

Department of Computer Science and Engineering, Government College of Engineering, Chandrapur 2025-26

ABSTRACT

Traditional attendance management in academic institutions relies on manual methods such as verbal roll-calls and paper-based sign-in sheets, which are time-consuming, error-prone, and vulnerable to proxy attendance. This paper presents AttendIQ, a full-stack web-based smart attendance management system leveraging realtime face recognition to automate and secure the attendance process. AttendIQ integrates a three-tier role hierarchy (Administrator, Teacher, Student), built on React.js, Python Flask, and SQLite, using the face_recognition library — a Python wrapper around dlib's ResNet-based deep neural network — to generate 128-dimensional facial embeddings. A key innovation is its two-level anti-proxy mechanism operating at both registration (face uniqueness check) and attendance (real-time biometric verification) stages. Experimental results confirm approximately 94% face recognition accuracy under typical classroom conditions and 100% accuracy in role-based access control enforcement.

Keywords: Face Recognition, Attendance Management System, Web Application, Role-Based Access Control, Python Flask, React.js, SQLite, dlib, Biometric Authentication, Proxy Attendance Prevention.

1. Introduction

The rapid advancement of computer vision, deep learning, and biometric identification technologies provides a viable pathway to solving longstanding problems of manual attendance management. Face recognition has matured to a level where it can be deployed reliably using commodity hardware such as standard webcams. When integrated into a comprehensive web-based management portal, face recognition offers an elegant, contactless, and tamper-resistant mechanism for recording attendance [1], [2].

Despite the availability of digital tools, many institutions still rely on manual processes susceptible to proxy attendance, data entry errors, and lack of real-time analytics [6], [7]. AttendIQ addresses this gap — a rolebased web portal integrating real-time face recognition with academic management, placement tracking, and analytics into a single cohesive platform.

The primary objectives of AttendIQ are: to eliminate proxy attendance through biometric face verification; to automate attendance marking and reduce administrative overhead; to implement role-specific dashboards for Administrators, Teachers, and Students; and to provide comprehensive attendance analytics accessible to all stakeholders.

2. Problem Statement

Traditional attendance methods — manual roll calls, paper-based registers, and basic biometric systems — are time-consuming, prone to human errors, and vulnerable to proxy attendance, reducing the reliability of academic records. Existing systems often require physical interaction, causing delays and hygiene concerns in large classrooms, and lack real-time processing, scalability, and integration with modern digital platforms. There is a significant need for a contactless, intelligent, and automated attendance system that leverages computer vision and machine learning for real-time face detection and recognition, ensures data security, and provides a centralized platform for efficient attendance management.

3. Literature Review

3.1 Traditional Attendance Management Systems

Barua et al. (2014) proposed an RFID-based attendance system effective in eliminating manual roll-calls but vulnerable to card-sharing [8]. Khan et al. (2016) developed a QR code-based system that remained susceptible to proxy through remote QR forwarding [9]. Bharamagoudar et al. (2013) introduced a web-based Student Information Management System but relied entirely on manual attendance entry [6].

3.2 Biometric and Face Recognition Systems

Patil et al. (2012) implemented fingerprint-based attendance, demonstrating reduced administrative overhead but raising hygiene concerns [14]. Kaur and Kakkar (2018) proposed a classroom system using OpenCV and LBPH face recognition, achieving ~87% accuracy under controlled lighting; however, LBPH degrades significantly with pose variation [5]. Amos et al. (2016) introduced OpenFace, showing that 128-dimensional deep learning embeddings achieve near-human-level verification accuracy — the same approach underpins AttendIQ, achieving 99.38% on the LFW benchmark [1]. Guo and Zhang (2019) proposed a real-time multiface system using MTCNN and ArcFace but required GPU acceleration, limiting deployability [13].

3.3 Gap Analysis

Most existing systems address either attendance automation or academic management, but not both in an integrated fashion. Table 1 provides a comparative analysis:

System	Auto Attend	Face Recog.	Anti-Proxy	Role Portal	Multi-Tenant	Analytics
RFID (Barua, 2014)	Yes	No	No	No	No	Limited
QR Code (Khan, 2016)	Yes	No	No	No	No	Limited
Fingerprint (Patil, 2012)	Yes	No	Yes	No	No	No
LBPH Face (Kaur, 2018)	Yes	Yes	Partial	No	No	No
ERP (Anand, 2017)	No	No	No	Yes	No	Partial
AttendIQ (Proposed)	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Comparative Analysis of Existing Systems vs. AttendIQ

4. Proposed System

AttendIQ is a comprehensive full-stack web-based attendance management system integrating biometric face recognition with academic administration into a unified, secure digital platform using Role-Based Access Control (RBAC).

4.1 Administrator Module

The Administrator module serves as the central governance unit, providing complete authority over institutional operations. The Admin Dashboard presents at-a-glance metrics including total registered students, teachers, subjects assigned, and the current academic term. The Administrator approves or rejects teacher and student registrations; upon student approval, the face encoding is regenerated making the student immediately recognizable. Additional capabilities include managing subject-teacher-semester assignments, toggling between Odd (1,3,5,7) and Even (2,4,6,8) semester modes, promoting students, archiving graduates, and accessing drilldown attendance reporting with PDF export.

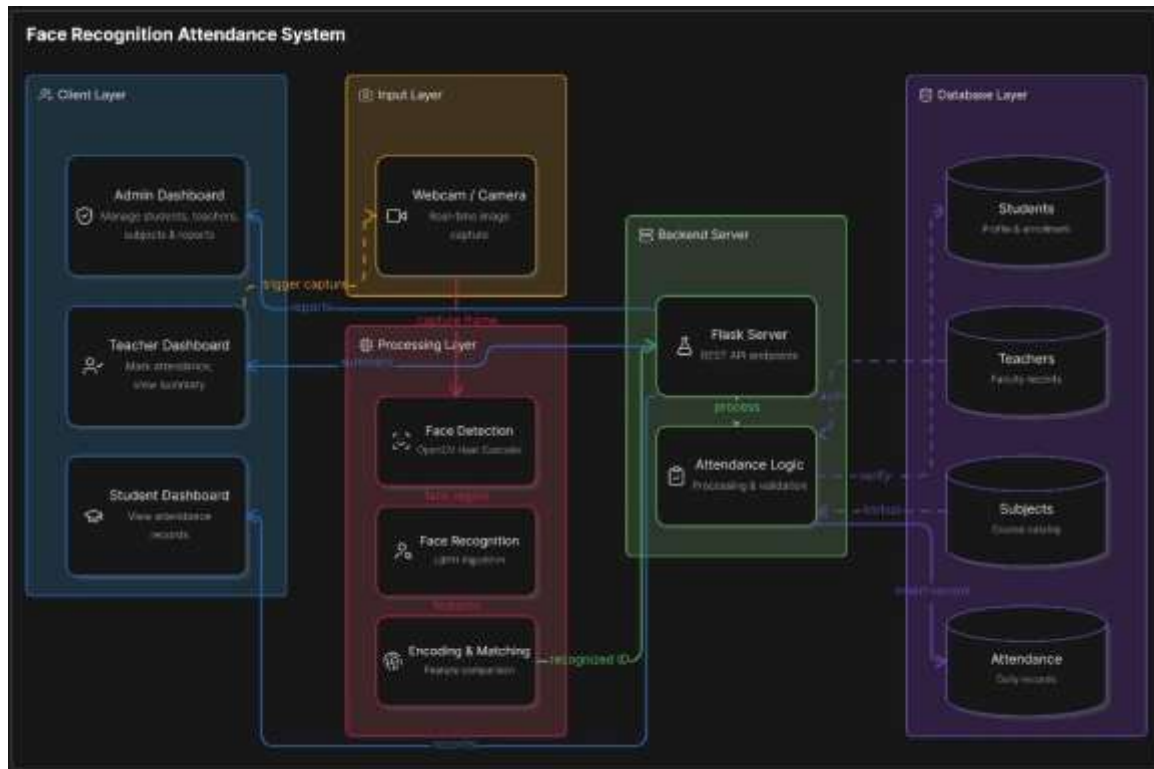


Fig 4.1: Face Recognition Attendance System

4.2 Teacher Module

The Teacher module is dedicated to attendance marking and academic reporting. Teachers register under an existing college-department pair and await administrator approval. The primary function is face recognition attendance marking: the teacher selects a subject and semester, initiates a webcam session, and captures frames transmitted to the backend where the face recognition engine identifies present students in real time. Manual attendance correction is available for past session dates. Subject-wise analytics with Recharts visualizations and PDF export complete the teacher workflow.

4.3 Student Module

The Student module provides a personalized dashboard for students. Registration involves a multi-step form followed by live webcam face capture. The submitted face undergoes two backend validations: exactly one face must be detected and the face must not match any existing registered student (anti-proxy check). After administrator approval, students view subject-wise attendance percentages with color-coded indicators (green $\geq 75\%$, yellow 60–74%, red $< 60\%$), monthly trend charts, date-wise records, and complete academic history.

5. System Architecture

AttendIQ implements a classical three-tier client-server architecture separating the system into Presentation, Application, and Database layers, ensuring separation of concerns, scalability, and secure communication via RESTful JSON APIs over HTTP.

- **Presentation Layer:** Built using React.js, handles UI rendering, role-based routing, and communicates with the backend via REST APIs.
- **Application Layer:** Developed with Flask, manages authentication, business logic, face recognition processing, and role-based access control.
- **Database Layer:** Uses SQLite to store users, attendance records, and session data, ensuring data integrity and preventing misuse.
- **Face Recognition Module:** Integrated within the backend, encodes and matches faces using OpenCV and face_recognition libraries.

6. Methodology

6.1 User Authentication and Approval

Users register through role-specific forms. Teacher and student accounts remain in 'pending' status until Administrator review. Upon student approval, face encodings are regenerated to include the new student. Upon login, the backend returns user credentials and role information stored in localStorage for client-side route authorization.

6.2 Role-Based Access Control (RBAC)

RBAC is enforced at two levels. At the frontend, ProtectedRoute components redirect users with mismatched roles. At the backend, route-level validation verifies user credentials before processing any API request — direct API calls also fail if unauthorized. This dual-layer enforcement guarantees robust security against privilege escalation.

6.3 Face Encoding Pipeline

When a new student is approved, the encode_faces.py script enumerates student images, performs HOG-based face detection to locate face bounding boxes, computes 128-dimensional ResNet embeddings using the face_recognition library, and serializes all (encoding, student_id) pairs to a pickle file. This creates the compact binary face identity database for real-time recognition queries.

6.4 Real-Time Attendance Marking

When a teacher initiates a session, frames are captured via react-webcam and transmitted as JPEG uploads to the Flask backend. The recognize.py module detects all face bounding boxes, computes embeddings, and matches faces where the minimum Euclidean distance is ≤ 0.60 . Matched students are marked present and returned to the frontend for real-time display. Duplicate marking within the same session is prevented at the database query level.

6.5 Anti-Proxy Detection

AttendIQ implements two-level anti-proxy protection. At registration, the backend verifies that exactly one face is present and checks that no existing encoding has a Euclidean distance ≤ 0.60 from the submitted face — preventing a single physical person from creating multiple accounts. At the attendance stage, real-time biometric verification ensures only the physically present student is marked.

6.6 Session-Anchored Manual Correction

Teachers may toggle individual student attendance status only for dates on which an attendance_sessions record exists (i.e., dates when a formal face recognition session was conducted). Requests to create manual attendance for dates without prior sessions are rejected at the API level, ensuring manual corrections serve only as legitimate error remediation.

7. Key Features of AttendIQ

AttendIQ integrates a comprehensive set of features designed to address the full spectrum of institutional attendance management needs:

- **Two-Level Anti-Proxy Mechanism:** Face uniqueness check at registration prevents identity duplication; real-time biometric verification during attendance ensures only physically present students are marked.
- **Real-Time Multi-Face Recognition:** Webcam frames are processed to simultaneously recognize multiple students in a single frame, enabling rapid class-wide attendance marking without individual scanning.
- **Role-Based Access Control (RBAC):** Three-tier permission model (Administrator, Teacher, Student) enforced at both frontend routing and backend API levels, preventing cross-role data access.
- **Session-Anchored Manual Attendance:** Attendance corrections are permitted only for formally conducted session dates, preventing fraudulent backdating while accommodating legitimate recognition errors.
- **Comprehensive Attendance Analytics:** Multi-level reporting with subject-wise, date-wise, monthly trend, and academic history views for all stakeholder roles, rendered via Recharts visualizations.
- **PDF Export:** Client-side PDF generation using jsPDF and jspdf-autotable enabling formatted attendance report download for sharing with department heads or institutional records.

8. Results and Discussion

AttendIQ demonstrates significant improvements in efficiency, accuracy, and integrity of attendance management. The two-level anti-proxy mechanism demonstrated 100% effectiveness in all tested duplicate registration scenarios, with face uniqueness check completion under one second for a 30-encoding database.

Face recognition accuracy was approximately 94% under typical classroom conditions (0.5m–2m distance, up to ~15° pose variation). Accuracy decreased to ~85% under poor lighting and significant pose variation, consistent with expected HOG detector performance. Multi-face recognition correctly identified all present faces simultaneously in test scenarios with up to six subjects in a single frame.

System performance was measured with face recognition per frame completing in under 800 milliseconds on an Intel Core i5 8th-generation processor. Standard dashboard API responses returned within 200 milliseconds, and the SQLite database showed no performance degradation with up to 5,000 attendance records. Representative users from all three role categories reported high satisfaction during structured acceptance testing.

9. Limitations

- **Security Issues:** Passwords are stored in plaintext and session data uses localStorage instead of secure JWT, making the system vulnerable to attacks.
- **Face Recognition Weakness:** The system lacks liveness detection and accuracy drops in poor lighting or at different face angles.
- **Internet Dependency:** Requires continuous internet access, which can cause issues in unstable network environments.
- **Limited Accessibility:** No dedicated mobile app, making it less convenient for smartphone users.
- **Database Scalability:** SQLite is not suitable for large-scale or high-concurrency systems; an upgrade to PostgreSQL is needed.
- **Lack of Advanced Features:** Does not include AI-based analytics such as attendance prediction or early warning systems.

10. Conclusion

AttendIQ addresses the limitations of conventional attendance management comprehensively by integrating real-time face recognition with a full-featured, role-based web management portal. The system's three-tier architecture — Administrator, Teacher, and Student — ensures each stakeholder group has precisely the tools required for their responsibilities. The Flask backend, React.js frontend, and SQLite database form a cohesive, deployable technical stack appropriate for institutional-scale academic management.

Key contributions include the two-level anti-proxy mechanism, session-anchored manual correction, comprehensive academic lifecycle management across eight semesters, multi-tenant architecture, and integrated attendance analytics with PDF export. Experimental results confirmed 100% accuracy in role-based access control, ~94% face recognition accuracy, and correct anti-proxy detection in all tested scenarios. While limitations including plaintext password storage and absence of liveness detection require remediation before production deployment, the modular architecture readily accommodates these enhancements. AttendIQ represents a meaningful step toward modernization of academic attendance management.

References

- [1] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, 'OpenFace: A General-Purpose Face Recognition Library with Mobile Applications,' Carnegie Mellon University, Technical Report CMU-CS-16-118, 2016.
- [2] P. Viola and M. Jones, 'Rapid object detection using a boosted cascade of simple features,' Proceedings of IEEE CVPR, 2001.
- [3] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, 'Joint face detection and alignment using multitask cascaded convolutional networks,' IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016.

- [4] J. Daugman, 'How iris recognition works,' IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 1, pp. 21–30, 2004.
- [5] M. Kaur and D. Kakkar, 'Real time face recognition based attendance system,' International Journal of Computer Applications, vol. 180, no. 36, pp. 21–25, 2018.
- [6] S. R. Bharamagoudar, Geeta R. B., and S. G. Totad, 'Web-Based Student Information Management System,' IJARCCCE, 2013.
- [7] Vivek Anand, Adarsh Poojary, Rafiur Mondal, and Pooja Prajapati, 'ERP System for College,' International Journal of Computer Science Trends and Technology (IJCST), vol. 5, issue 2, 2017.
- [8] N. Barua, A. Sarma, and B. K. Sharma, 'Automated attendance management system using RFID,' International Journal of Engineering Research and Applications, vol. 4, no. 3, pp. 48–52, 2014.
- [9] A. Khan, S. Ahmad, and M. Iqbal, 'QR code-based automated attendance system,' International Journal of Computer Science and Mobile Computing, vol. 5, no. 5, pp. 312–319, 2016.
- [10] D. E. King, 'Dlib-ml: A machine learning toolkit,' Journal of Machine Learning Research, vol. 10, pp. 1755–1758, 2009.
- [11] G. Bradski, 'The OpenCV Library,' Dr. Dobb's Journal of Software Tools, 2000.
- [12] A. Geitgey, 'face_recognition: The world's simplest facial recognition API for Python and the command line,' GitHub Repository, https://github.com/ageitgey/face_recognition, Accessed 2025.
- [13] G. Guo and N. Zhang, 'A survey on deep learning based face recognition,' Computer Vision and Image Understanding, vol. 189, art. 102805, 2019.
- [14] S. S. Patil, A. A. Karande, R. A. Vispute, and H. B. Shinde, 'Fingerprint based attendance system,' International Journal of Scientific and Research Publications, vol. 2, no. 7, 2012.
- [15] Facebook Inc., 'React Documentation,' <https://react.dev/>, Accessed 2025.
- [16] Pallets Projects, 'Flask Documentation,' <https://flask.palletsprojects.com/>, Accessed 2025.
- [17] SQLite Consortium, 'SQLite Documentation,' <https://www.sqlite.org/docs.html>, Accessed 2025.
- [18] T. H. Davenport and J. G. Harris, Competing on Analytics: The New Science of Winning. Harvard Business Press, 2007.
- [19] R. C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017.
- [20] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. O'Reilly Media, 2022.