IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511208

AntiProxy: Smart Attendance System Using Face Recognition and GPS-Based Geofencing

Prof. S. N. Benkikeri[1], Shreenivas Goundi[2], Sujal Ganiger[3], Vishwanath S. Kaladagi[4], Shrinivas Das[5]

Assistant Professor, Department of Computer Science and Engineering, B.V.V. Sangha's Basaveshwar Engineering College, Bagalkot[1]

Students, Department of Computer Science and Engineering, B.V.V. Sangha's Basaveshwar Engineering College, Bagalkot[2,3,4,5]

Abstract: AntiProxy is a multi-factor digital attendance management system designed to reduce proxy attendance in educational institutions. It combines QR-code-based session validation, facial recognition, and GPS-based geofencing to to verify the student's identity and check whether the mobile device is within the configured classroom geofence before attendance is recorded. The system is built with a Flutter mobile application, a Python Flask backend, and a MySQL database, and provides separate interfaces for administrators, faculty members, and students. Faculty members open a time-bound attendance session with a dynamic QR code and a classroom boundary; a student scans the code and is verified through facial embeddings generated by the InsightFace Buffalo_L model and compared with enrolled data using cosine similarity, followed by a geofence check of the device location. Attendance is stored only when all checks succeed, and duplicate entries are prevented. The system also provides attendance history, PDF reports, and low-attendance notifications. A comparative evaluation of proxy-attempt prevention, attendance completion time, and verification factors is presented using illustrative estimates, and measured performance testing is identified as future work.

Keywords: AntiProxy, Smart Attendance System, Face Recognition, GPS-Based Geofencing, QR Code.

I. INTRODUCTION

Attendance records influence academic assessment, examination eligibility, and institutional reporting, and the spread of mobile and artificial-intelligence technologies has made their automation practical. Manual registers and simple digital check-in tools, however, remain open to proxy attendance, unauthorized marking, transcription errors, and slow report preparation. A dependable attendance system therefore has to establish both who is marking attendance and where that person is.

Three mechanisms are commonly used for this purpose, and each leaves a gap when used alone. A QR code identifies an active attendance session, but it cannot tell who is scanning it, and a code forwarded to an absent student can be used without any identity check. Facial recognition verifies identity [7], [8], yet it does not show that the person is inside the classroom. GPS-based geofencing checks the location of a device [10], [12], but not who is holding it. Beyond verification, institutions also need centralized records with role-based access so that administrators, faculty members, and students can manage and monitor attendance without manual reconciliation.

This paper presents AntiProxy, a smart attendance system that integrates QR-code validation, facial recognition using the InsightFace Buffalo_L model, and GPS-based classroom geofencing, and verifies student identity and classroom presence before attendance is recorded. The main contributions are as follows:

- A layered architecture of a Flutter mobile application, a Python Flask REST API, a verification layer, and a MySQL database, with separate role-based interfaces for administrators, faculty members, and students.
- A sequential three-stage verification workflow (QR session validation, face verification, and geofence check) in which attendance is stored, with duplicate prevention, only when all stages succeed.
- An implementation that uses InsightFace Buffalo_L embeddings with cosine similarity for identity verification and a faculty-configured rectangular boundary for location verification.
- Attendance monitoring features, including attendance history, PDF reports, and notifications for attendance below an 80% threshold.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511208

In the remainder of this paper, Section II reviews related work. Section III presents the proposed system, covering its architecture, verification methodology, and algorithm. Section IV presents a comparative evaluation and discusses limitations and ethical considerations. Section V concludes the paper and outlines future enhancements.

II. RELATED WORK

Research on automated face analysis increasingly relies on deep-learning approaches. ArcFace [7] improves facial feature discrimination through an additive angular margin loss, while AdaFace [8] adapts the margin according to image quality to improve recognition under low-quality imaging conditions. RetinaFace [1] provides face detection in unconstrained images and supports facial localization for recognition. Since photographs and replayed videos can be used to deceive face recognition systems, face anti-spoofing techniques have been developed using patch- and depth-based convolutional neural networks (CNNs) [2], auxiliary supervision [3], and studies examining the influence of model design and training data [4]. Yu et al. [9] reviewed deep-learning-based face anti-spoofing approaches, covering pixel-wise supervision, domain generalization, and multimodal sensors. These techniques are relevant to strengthening face-based attendance systems against presentation attacks involving photographs and videos.

Among attendance systems, Habila et al. [6] developed a real-time face-recognition system based on InsightFace and ArcFace features, with consent, encryption, and access-control safeguards, and recommended liveness detection as a further enhancement. QR codes provide a standardized way to encode information for machine reading [5], and recent systems build on them: Nwabuwe et al. [10] combined dynamic QR codes, geofencing, and IMEI verification to counter QR sharing and proxy attendance, Taju et al. [11] combined QR codes with geolocation for student attendance, and Gaftandzhieva and Lyavova [12] proposed a Flutter-based system that uses time-limited dynamic QR codes, device binding, and Haversine-based geofencing; they note that countermeasures against advanced location spoofing, such as mock-GPS tools, were not benchmarked.

These works address different parts of attendance verification, and each has limitations when applied on its own: QR codes validate a session but not the student, facial recognition verifies identity but not presence, and GPS checks device location but not identity. AntiProxy combines these mechanisms with centralized database management and role-based access control to strengthen attendance verification and reduce proxy attendance.

III. PROPOSED WORK

AntiProxy is a layered client–server system in which every attendance request must pass three verification stages before it is recorded. The system is developed using Flutter for mobile application development, Python Flask for backend API services, and MySQL for database management. OpenCV is used for image processing, while InsightFace Buffalo_L supports facial feature extraction and identity verification. The Flutter Geolocator package provides device location information for GPS-based geofencing. These technologies work together to support mobile application functionality, backend communication, facial verification, location checking, and centralized attendance record management [14][19].

A. System Architecture

The architecture consists of four layers, the frontend layer, application layer, verification layer, and data storage layer, as shown in Fig. 1. This separation keeps the system modular, simplifies its management, and allows several verification mechanisms to be integrated into one process.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511208

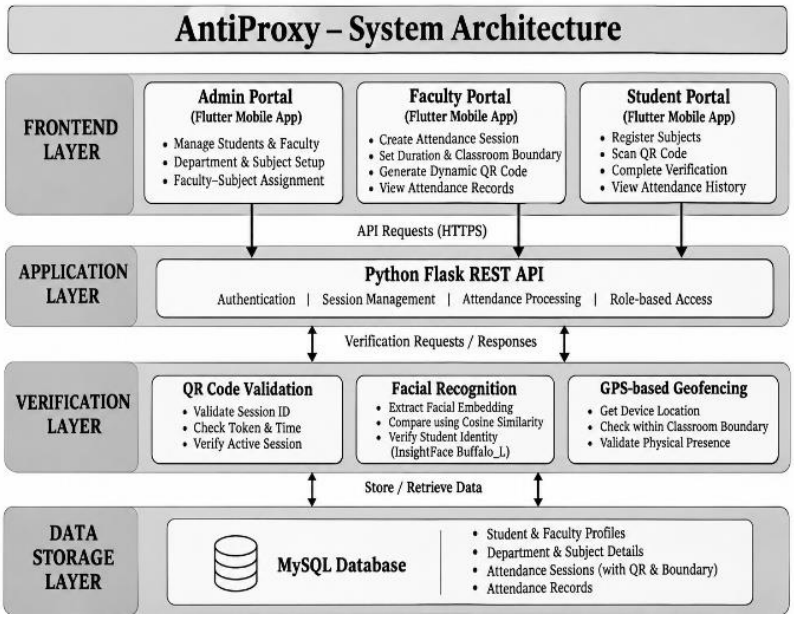


Fig. 1 System architecture of AntiProxy.

Frontend layer. The Flutter mobile application provides separate portals for administrators, faculty members, and students. Administrators manage users, departments, subjects, and faculty–subject assignments; faculty members create attendance sessions, generate the dynamic QR code, and view attendance records; and students register subjects, scan QR codes with the integrated scanner, complete verification, and view their attendance history. The application communicates with the backend through REST API requests.

Application layer. A Python Flask REST API handles authentication, role-based access, session management, and attendance processing. It receives requests from the mobile application, passes verification requests to the verification layer, and manages attendance recording.

Verification layer. This layer contains three modules: QR-code validation, facial recognition with InsightFace Buffalo_L and OpenCV-based image processing, and GPS-based geofencing against the classroom boundary configured by the faculty member.

Data storage layer. A MySQL database stores student and faculty profiles, department and subject details, faculty–subject assignments, attendance sessions (with their QR and boundary information), and attendance records.

B. Verification Methodology

AntiProxy follows a sequential multi-factor process so that attendance is recorded only for an authorized student who is present in the classroom. The process, illustrated in Fig. 2, has six steps.

1) Registration and face enrollment. Students register through the Flutter application by providing their profile details. During enrollment, the captured facial image is processed by the InsightFace Buffalo_L model, and the resulting facial embedding is stored as the reference for later verification.

2) Session creation. A faculty member selects the subject, sets the session duration, and configures the classroom location and dimensions. The system then generates a dynamic QR code containing session identification and security information.

3) QR-code validation. The student scans the QR code with the mobile application. The Flask backend validates the session ID, security token, token validity, and active session status, and rejects invalid or expired requests. The dynamic QR token is refreshed periodically and remains valid for 30 seconds, reducing the possibility of reusing an expired QR code.

4) Facial verification. After QR validation, the student's face is captured and its embedding is generated with InsightFace Buffalo_L. The embedding is compared with the enrolled data using cosine similarity, and requests that do not match are

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511208

rejected. The student interface also performs a blink-based live-face check as an additional liveness verification step. Attendance verification proceeds only when the configured liveness check succeeds.

5) Geofence verification. The application obtains the device's GPS coordinates and sends them to the backend, which compares them with the classroom reference location and the configured boundary dimensions. The mobile device's reported position is evaluated against the rectangular classroom boundary using north–south and east–west displacement, and requests from outside the permitted area are rejected.

6) Attendance recording. Only when all required checks succeed is the attendance stored in MySQL, with duplicate entries prevented, and a confirmation is shown to the student. A request that fails any stage is rejected and an error message is displayed. Students can view their attendance history, and faculty members can access session-wise records and generate PDF reports; the application also issues low-attendance notifications when a student's attendance falls below the configured 80% threshold.

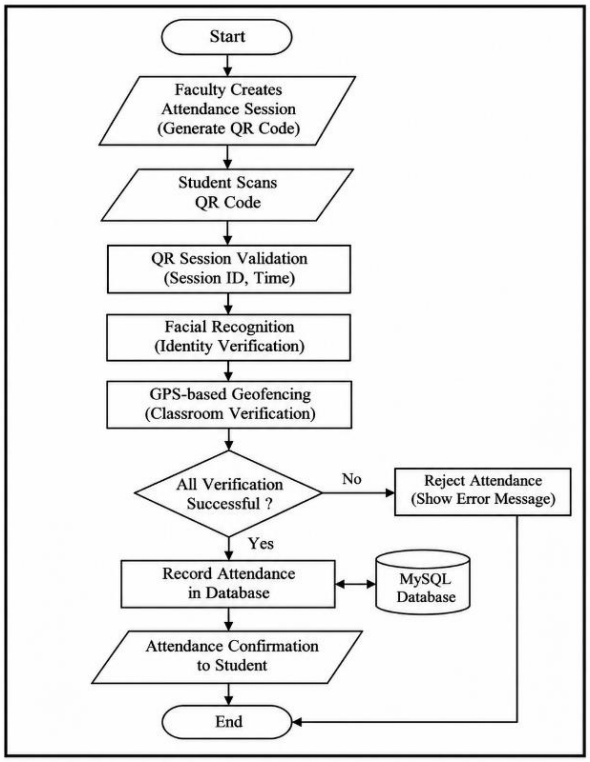


Fig. 2 Overall workflow of the proposed AntiProxy system.

C. Proposed Algorithm

The attendance verification procedure of AntiProxy is summarized as follows.

Begin

Step 1: The faculty member selects the subject, duration, and classroom location and dimensions; the system creates the session and generates a dynamic QR code.

Step 2: The student scans the QR code in the mobile application.

Step 3: The backend validates the session ID, security token, token validity, and session status; if any check fails, reject and go to Step 9.

Step 4: The student's face is captured and its embedding is generated with InsightFace Buffalo_L.

Step 5: Compute the cosine similarity between the captured and enrolled embeddings; if the identity does not match, reject and go to Step 9.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511208

Step 6: The application obtains the device coordinates through the Geolocator package and sends them to the backend.

Step 7: The backend evaluates the north–south and east–west displacement of the device against the rectangular classroom boundary; if the device is outside the boundary, reject and go to Step 9.
Step 8: If all verification stages succeed and no previous attendance record exists for the student in the current session, the backend stores the attendance as PRESENT in MySQL. If a record already exists, the duplicate request is rejected.

Step 9: Return the attendance confirmation or the error message to the student.

End

IV. COMPARATIVE ANALYSIS AND DISCUSSION

The AntiProxy system was evaluated through a comparative analysis of proxy attendance prevention, attendance completion time, and the verification mechanisms used by different attendance systems. The evaluation focuses on how the integration of QR-code validation, facial recognition, and GPS-based geofencing provides multiple layers of attendance verification. The graphs in this section compare AntiProxy with manual, QR-code-based, face-recognition-based, and combined QR and face-recognition attendance approaches.

A. Comparison of Proxy Attendance Attempts

Fig. 3 compares the number of proxy attendance attempts accepted by the different systems, considering 50 attempts for each system. AntiProxy verifies the attendance session, student identity, and classroom presence together. Relative to manual attendance, which accepted 42 of the 50 attempts in this illustrative comparison, the chart shows 28 accepted attempts for QR-code-based attendance (33.3% reduction), 12 for face-recognition-based attendance (71.4%), 5 for combined QR and face recognition (88.1%), and 1 for AntiProxy (97.6%). This multi-factor approach is designed to reduce opportunities for unauthorized attendance marking compared with systems that rely on an individual verification method.

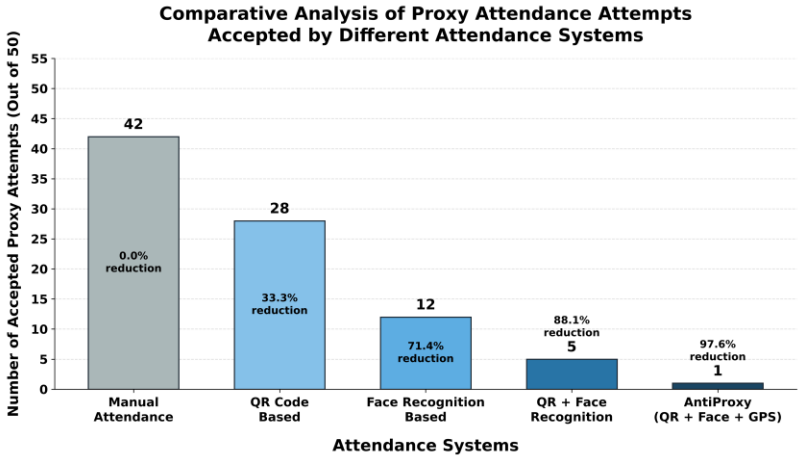


Fig. 3 Comparative analysis of proxy attendance attempts accepted by different attendance systems.

B. Comparison of Attendance Completion Time

Fig. 4 compares the estimated total attendance completion time for 60 students. Unlike sequential attendance recording, the comparison assumes that students submit attendance simultaneously through their mobile devices. AntiProxy is provisionally estimated to complete attendance marking within approximately two minutes for 60 students. The actual time depends on network connectivity, server capacity, facial recognition processing, and simultaneous API requests, so practical concurrent-load testing is required to determine it.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511208

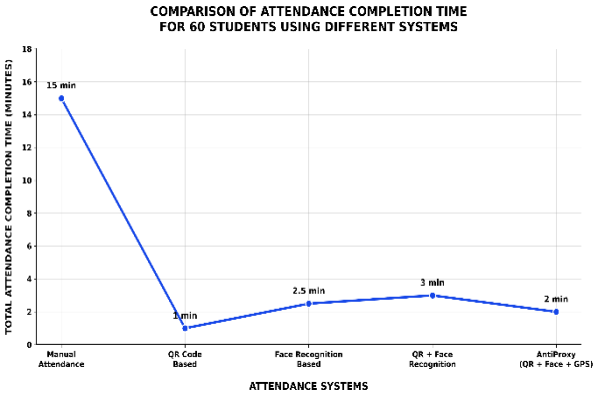


Fig. 4 Comparison of attendance completion time for 60 students using different attendance systems.

C. Comparison of Verification Factors

Fig. 5 compares the number of verification factors used by each system. Manual attendance, as represented in this comparison, does not include automated digital identity or location verification, QR-code-based and face-recognition-based systems each use a single mechanism, and systems that combine QR codes with facial recognition provide two factors. AntiProxy integrates three: QR-code validation for the attendance session, facial recognition for student identity, and GPS-based geofencing for classroom location.

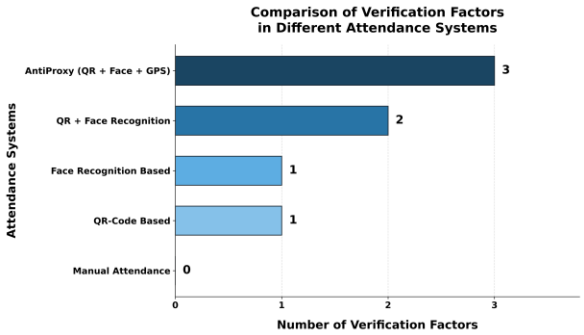


Fig. 5 Comparison of verification factors employed by different attendance systems.

D. Overall Evaluation

The comparison highlights the main advantages of AntiProxy in multi-factor verification and proxy attendance prevention. Integrating the three mechanisms provides multiple checks before attendance is recorded in the database, and the system additionally supports centralized attendance management and report generation. However, the numerical values in the proxy-attempt and completion-time graphs (Figs. 3 and 4) are illustrative estimates and must not be regarded as experimentally measured results. Further testing under identical conditions, including simultaneous attendance requests from 60 students, is necessary to establish actual system performance.

Practical operation also depends on camera availability, GPS positioning conditions, network connectivity, and environmental factors, and GPS may be less precise indoors, which can affect classroom location verification. The present evaluation does not establish facial recognition accuracy, false acceptance rate, false rejection rate, or GPS positioning error.

The verification-factor comparison represents the number of distinct verification mechanisms incorporated into each system and does not directly measure authentication accuracy or resistance to sophisticated spoofing attacks.

E. Ethical and Privacy Considerations

AntiProxy handles facial and location data, so privacy and data protection are essential. Such data should be collected with appropriate consent, used only for attendance verification, and retained only for legitimate academic purposes, and students should be informed how their data are collected and used. Role-based access control is used to restrict access to student and faculty records according to assigned permissions. Appropriate safeguards should also be applied to facial and location information to reduce unauthorized access, disclosure, and misuse, and appropriate security measures and institutional data protection policies should be followed to prevent unauthorized access and misuse. In India, the

DOI: 10.17148/IJARCCE.2026.1511208

processing of digital personal data is governed by the Digital Personal Data Protection Act, 2023 [13], so consent and purpose limitation should be built into the deployment of such a system.

V. CONCLUSION AND FUTURE ENHANCEMENTS

This paper presented AntiProxy, a multi-factor digital attendance management system that integrates QR-code validation, facial recognition, and GPS-based geofencing to reduce proxy attendance in educational institutions. Developed using Flutter, Python Flask, and MySQL, the system verifies the attendance session, student identity, and mobile device location before recording attendance. It also provides role-based access, attendance history, centralized record management, and PDF report generation. The comparative analysis illustrates the benefits of combining three verification mechanisms, although the numerical values presented are illustrative estimates rather than experimentally measured results.

Future work will include testing attendance performance with 60 simultaneous student requests, evaluating facial recognition accuracy through false acceptance and rejection rates, improving indoor location verification, and conducting comprehensive security and usability testing.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to **Prof. S. N. Benkikeri**, Project Guide, Department of Computer Science and Engineering, Basaveshwar Engineering College, Bagalkot, for his valuable guidance, continuous support, and encouragement throughout the development of the AntiProxy system. The authors also thank the Head of the Department and faculty members for providing the necessary support and facilities to complete this project successfully.

REFERENCES

[1]. J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 5203–5212, 2020.

[2]. Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based CNNs," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, pp. 319–328, 2017.

[3]. Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 389–398, 2018.

[4]. Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Face anti-spoofing: Model matters, so does data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3507–3516, 2019.

[5]. ISO/IEC 18004:2015, *Information technology—Automatic identification and data capture techniques—QR Code bar code symbology specification*. International Organization for Standardization, 2015.

[6]. M. Habila, F. N. Francisca, L. Ishaya, M. K. Ahmed, U. A. Muhammad, and H. P. Charles, "Smart real-time attendance system for Nigerian universities," *J. Inf. Organ. Sci.*, vol. 49, no. 1, pp. 121–138, 2025, doi: 10.31341/jios.49.1.8.

[7]. J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 5962–5979, Oct. 2022, doi: 10.1109/TPAMI.2021.3087709.

[8]. M. Kim, A. K. Jain, and X. Liu, "AdaFace: Quality adaptive margin for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 18729–18738, 2022, doi: 10.1109/CVPR52688.2022.01819.

[9]. Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep learning for face anti-spoofing: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5609–5631, May 2023, doi: 10.1109/TPAMI.2022.3215850.

[10]. A. Nwabuwe, B. Sanghera, T. Alade, and F. Olajide, "Fraud mitigation in attendance monitoring systems using dynamic QR code, geofencing and IMEI technologies," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 4, pp. 938–945, 2023, doi: 10.14569/IJACSA.2023.01404104.

[11]. S. W. Taju, Y. P. Mamahit, and J. A. Pongantung, "Implementing QR code and geolocation technologies for the student attendance system," *CogITo Smart J.*, vol. 10, no. 1, pp. 221–232, Jun. 2024, doi: 10.31154/cogito.v10i1.636.642-653. [11] S. W. Taju, Y. P. Mamahit, and J. A. Pongantung, "Implementing QR code and geolocation technologies for the student attendance system," *CogITo Smart J.*, vol. 10, no. 1, pp. 221–232, Jun. 2024, doi: 10.31154/cogito.v10i1.636.642-653.

[12]. S. Gaftandzhieva and G. Lyavova, "AttendEase: Fraud-resistant attendance tracking in higher education using dynamic tokens and geofencing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 17, no. 9, pp. 230–242, 2026.

[13]. Government of India, *The Digital Personal Data Protection Act, 2023* (Act No. 22 of 2023), Aug. 2023.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511208

[14]. Flutter, "Flutter Documentation," Google. [Online]. Available: https://docs.flutter.dev/. [Accessed: Nov. 22, 2026].
[15]. Pallets Projects, "Flask Documentation." [Online]. Available: https://flask.palletsprojects.com/. [Accessed: Nov. 22, 2026].
[16]. Oracle Corporation, "MySQL Reference Manual." [Online]. Available: https://dev.mysql.com/doc/. [Accessed: Nov. 22, 2026].
[17]. OpenCV, "OpenCV Documentation." [Online]. Available: https://docs.opencv.org/. [Accessed: Nov. 22, 2026].
[18]. InsightFace, "InsightFace: 2D and 3D Face Analysis Project." [Online]. Available: https://github.com/deepinsight/insightface. [Accessed: Nov. 22, 2026].
[19]. Flutter Community, "Geolocator: Flutter Plugin for Location Services." [Online]. Available: https://pub.dev/packages/geolocator. [Accessed: Nov. 22, 2026].