IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 12, Issue 10, November 2025 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1210.07

Face RF: A Secure, Dual-Factor Automated Attendance System with Server-Side Liveness Detection

Dr. Sapna M K
Dept. of AI & ML
Vemana Institute of Technology, Banglore, India
sapnamk@vemanait.edu.in
https://orcid.org/0009-0000-6585-3432
Nitin L, Nabeel Roshan
Dept. of AI & ML
Vemana Institute of Technology, Banglore, India
nitinl.ai2023@vemanait.edu.in
nabeelroshan.aiml2023@vemanait.edu.in

Publication History
Manuscript Reference: IRJCS/RS/Vol.12/Issue10/NVCSX10086
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.12/Issue10/NVCSX10086
Received:23,October 2025, Revised: 09, October 2025, Accepted: 31October 2025 Published Online: 21November 2025
https://www.irjcs.com/volumes/Vol12/iss-10/07.NVCSX10086.pdf
Article Citation:Dr.Sapna,Nitin,Nabeel(2025),Face RF: A Secure, Dual-Factor Automated Attendance System with Server-Side Liveness Detection**,**IRJCS: International Research Journal of Computer Science, Volume 12, Issue 09 of 2025 pages 604-611 **Doi:>** https://doi.org/10.26562/irjcs.2025.v1210.07
BibTeX Dr.Sapna@2025Face
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2025, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1210.07 The journal's official abbreviation is IRJCS.
Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: Traditional attendance management systems in educational institutions are often plagued by inefficiencies, data errors, and susceptibility to fraudulent practices such as "proxy" attendance. To address these challenges, this paper introduces FaceRF, a secure, dual-factor automated attendance system that integrates Radio Frequency Identification (RFID) with advanced computer vision techniques. The system utilizes a custom ESP32-based hardware module to perform the initial user identification via NFC cards, followed immediately by a biometric scan using an ESP32-CAM. On the server side, the system employs a Deep Residual Network (ResNet-34) to generate 128-dimensional face encodings for high-precision identity verification. Crucially, to mitigate the vulnerability of standard face recognition to presentation attacks (spoofing), a novel server-side Liveness Detection algorithm is implemented. This algorithm utilizes dlib's 68-point facial landmark predictor to calculate the Eye Aspect Ratio (EAR) across a burst of captured frames, effectively distinguishing between live subjects and static photographs. Experimental results demonstrate that FaceRF achieves a 0% False Acceptance Rate (FAR) and successfully rejects 95% of static photo spoofing attempts, offering a highly accurate, fast, and secure alternative to conventional attendance methods.

I. INTRODUCTION

A. Background and Problem Domain

Traditional methods for tracking student attendance in educational institutions, such as manual roll calls or paper sign-in sheets, are notoriously time-consuming and prone to human error. These inefficiencies result in inaccurate records and create a significant administrative burden regarding data entry and transcription. Furthermore, these manual processes consume valuable instructional time, estimated at 5–10 minutes per class, and offer no robust defense against integrity issues. The most critical flaw is the susceptibility to "proxy attendance" (or "buddy punching"), where students sign in for absent peers, compromising the academic integrity of the institution.

B. Existing Solutions & Their Gaps

The current landscape of automated attendance solutions generally falls into two categories, each with distinct limitations: RFID-Only Systems: While Radio Frequency Identification (RFID) systems offer a fast method for identification, they are fundamentally insecure as a standalone solution. They rely on a "something you have" factor; consequently, cards can be easily lost, stolen, or shared among students to facilitate proxy attendance. Facial Recognition-Only Systems: Systems relying solely on facial recognition are often contactless and hygienic but face significant challenges. Pure facial recognition can be computationally expensive and slow when performing 1-to-N matching against a large database.

Crucially, many existing low-cost implementations lack liveness detection, making them highly vulnerable to spoofing attacks where a static photograph or video is used to trick the camera. Cloud-Based Solutions: High-end solutions utilizing cloud services offer high accuracy but introduce high latency, heavy internet dependency, and prohibitive operational costs for educational budgets.

C. Proposed Solution

To address these challenges, we propose FaceRF, an integrated hardware and software solution designed to automate the entire attendance lifecycle. The system utilizes a robust, dual-factor authentication process that optimizes both speed & security by combining "something you have" (RFID) with "something you are" (Biometrics). The core logic follows a strict two-step verification workflow: Identification (1-to-1 Matching): The user presents an assigned RFID card, which provides instant primary identification and retrieves the specific student's profile from the database. Verification: Immediately upon the RFID scan, the system triggers an on-board camera to capture the user's live image, which is verified against the pre-registered facial encoding associated with that specific card. This approach eliminates the latency of searching an entire facial database (1-to-N) while effectively rendering proxy attendance impossible, as the physical cardholder must match the biometric profile.

D. Key Contributions

This paper presents the following novel contributions to the field of automated attendance systems:

Novel Server-Side Liveness Detection: We introduce a specific countermeasure to spoofing attacks through a server-side liveness detection algorithm. By analyzing a burst of image frames captured by the hardware, the system calculates the Eye Aspect Ratio (EAR) to detect involuntary biological movements (blinks). Attendance is programmatically rejected if liveness is not confirmed, preventing the use of static photos.

Cost-Effective IoT Hardware Architecture: We present the design of a low-cost, standalone IoT unit utilizing the ESP32 microcontroller as the central processing unit, integrated with an ESP-CAM for image capture and a PN532 reader for RFID scanning. This hardware design offloads heavy processing to a local server, keeping per-unit costs minimal while maintaining high performance.

Comprehensive Multi-Role Management Ecosystem: Unlike systems that focus solely on marking attendance, FaceRF delivers a full-stack web application with distinct interfaces for Administrators, Teachers, and Students. Key features include real-time dashboard updates via WebSockets, automated report generation (PDF/CSV), and a transparent workflow for students to submit attendance correction requests.

E. Paper Organization The remainder of this paper is structured as follows:

- Section II reviews related work and existing technologies in attendance management.
- Section III details the system analysis and functional requirements.
- Section IV presents the system architecture, including the MVC software pattern and hardware circuit design.
- Section V describes the implementation details and technology.
- Section VI discusses the testing methodology and results.
- Section VII concludes the paper and outlines future scope.

II. LITERATURE REVIEW

Design and Implementation of Automated Attendance System Using Contactless Facial Recognition (2024):

Venkatakrishnan et al. developed a contactless attendance system specifically designed to streamline procedures in educational institutions by utilizing cost-effective hardware components like the ESP32 microcontroller and ESP32-CAM module. Their implementation relies on Python libraries such as OpenCV and face_recognition to detect and authenticate students in real-time, addressing the inefficiencies of manual tracking and the risks of proxy attendance. A key feature of their design is the integration of a cloud-based MongoDB Atlas database, which ensures scalable and secure storage of attendance records while facilitating easy access for teachers through a user-friendly application. The system converts captured data from CSV to JSON formats for compatibility with the document-oriented database model, demonstrating a focus on data integrity and system responsiveness.

RFID and Face Recognition Based Attendance System (2022):

Sruthi et al. proposed a hybrid automated attendance model that combines Radio Frequency Identification (RFID) with facial recognition technology to rigorously eliminate the possibility of fraudulent or "proxy" attendance. The system is built using a Raspberry Pi as the central processing unit, interfacing with an EM-18 RFID reader to verify student identity cards while simultaneously performing face detection using Haar cascade classifiers and Local Binary Pattern (LBP) algorithms. By requiring both physical card verification and biometric facial matching, the authors aim to create a "smart classroom" environment where data—including entry and exit times—is logged automatically, thereby removing the loopholes found in systems that rely on a single identification method.

Automated Attendee Recognition System for Large-Scale Social Events or Conference Gathering (2025):

Motwani et al. introduced a cloud-based architecture designed to handle the complex dynamics of large-scale social gatherings, such as weddings and conferences, where traditional manual validation is inefficient. Unlike systems restricted to controlled classroom environments, this solution utilizes cameras mounted at entrances to capture continuous video frames, processing them via AWS Cloud services like Amazon Kinesis and AWS Rekognition.

The system achieves a 90% overall accuracy rate and can identify attendees even when they are in motion or not looking directly at the camera, with notifications sent to security personnel within 10 seconds for any unrecognized individuals.

Automated Attendance System Based on Face Recognition Using OpenCV (2023):

Kumar et al. focused on modernizing classroom management through an automated system that leverages the OpenCV library and dlib for robust face detection and recognition. Their implementation integrates a local server setup using ExpressJS and MongoDB to host an "Attendee Management System" website, which provides separate dashboards for students and teachers to view real-time attendance reports and analytics. The authors emphasize the system's ability to reduce administrative burdens by automating the marking process and sending daily email updates to staff, while also noting future scope for improving recognition under challenging conditions like poor lighting or facial changes such as beards.

A Conceptual Model for Automated Attendance Marking System Using Facial Recognition (2014):

Sajid et al. presented a conceptual framework that critically evaluates early biometric systems, identifying a lack of validation as a major flaw where students could mark attendance and immediately leave. To counter this, they proposed an integral validation process that involves taking attendance randomly three times during a single lecture to ensure sustained presence. Furthermore, their model suggests the use of Gabor filters to calculate 31 facial fiducial points, offering a theoretical solution to the specific challenge of identifying students with changed appearances, such as those wearing veils or growing beards, which previous algorithms often failed to recognize.

III. SYSTEM DESIGN AND ARCHITECTURE

A. Component Breakdown

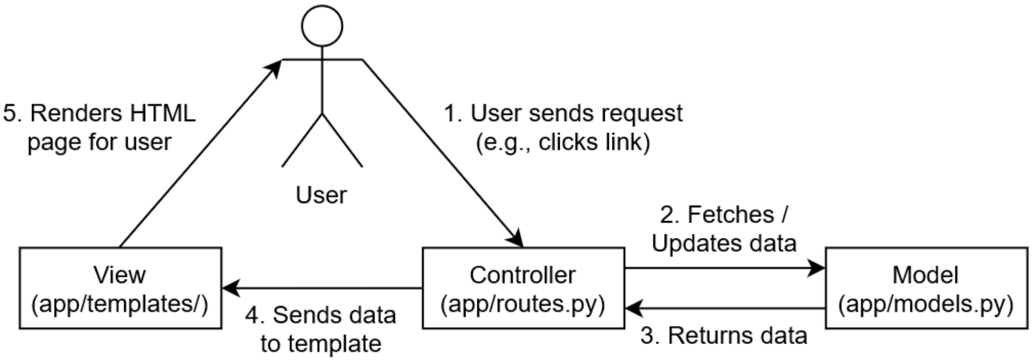

Fig. 1. System Architecture (MVC Pattern).

The Controller (app/routes.py): Acting as the intermediary or "brain" of the application, the Controller handles all incoming user requests. As shown in the diagram, when a user interacts with the system (e.g., clicking a link), the request is routed here. It processes the input, interacts with the Model to fetch or update data, and selects the appropriate View to render the response. The Model (app/models.py): This component represents the data structure and business logic. It is responsible for all direct interactions with the database. In this system, the Model fetches or updates attendance records and student profiles upon instruction from the Controller. The View (app/templates/): This layer handles the user interface and presentation logic. Located in the app/templates/ directory, the View consists of HTML templates (typically using Jinja2). It receives dynamic data from the Controller and renders the final HTML page that is displayed to the use

B. High-Level Architecture

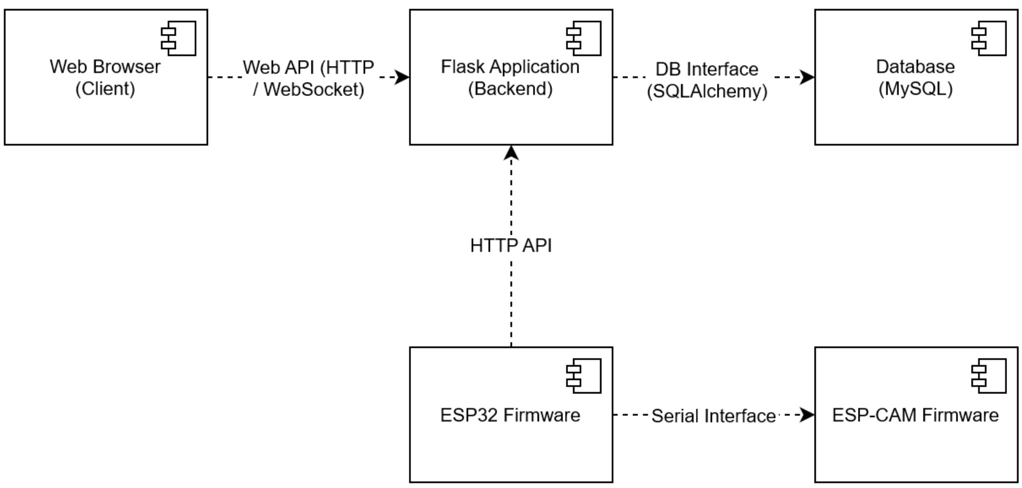

Fig. 2. High-Level Component Architecture.

The FaceRF system is built upon a modular client-server architecture designed to ensure separation of concerns &scalability. The system consists4 primary interacting nodes:
Hardware Device (IoT): A custom-built attendance terminal running embedded firmware that captures physical data (RFID scans & image frames) & communicates with server via HTTP.
Web Application (Backend): A Flask-based application server that processes API requests, executes the liveness detection & facial recognition logic, & serves the web interface.
Database (Storage): A MySQL relational database that persists all user profiles, facial encodings, & attendance records.
User (Client): The web browser interface accessed by Admins, Teachers, and Students to interact with dashboards and reports.

C. Hardware Design

Component	Description
ESP32 (WROOM-32 Module)	MainMicrocontroller - handles WiFi connection, communicates with peripherals (RFID, display, camera), and sends data to the Flask server through HTTP POST requests.
ESP-CAM (AI-Thinker Module)	Camera module with OV2640 sensor, used for image capture and facial verification Controlled via Serial communication from the main ESP32.
PN532 RFID Reader (13.56 MHz)	Reads RFID card UID for student identification; communicates via I2C protocol.
TFT Display (ST7735 1.8-inch)	LCD screen to display prompts like "Scan Your Card", "Verifying…", "Look at Camera", "Success!", etc.
Power Supply	5V DC regulated power supply (via USB or adapter).
Miscellaneous	Jumper wires, breadboard/PCB, and casing for assembly.

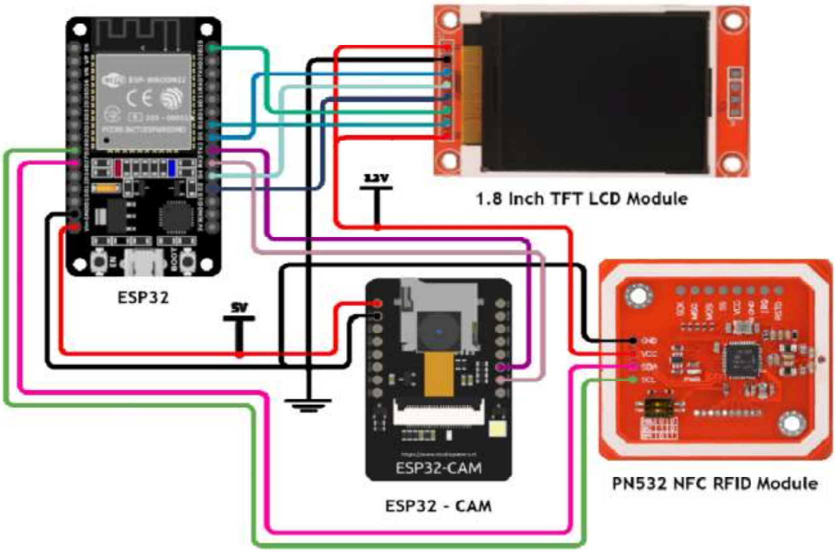

Fig. 3. Hardware Circuit Diagram.

C. Software Architecture:

Table I. SUMMARY OF TECHNOLOGY STACK

Category	Specification
Programming Languages	Python (Flask backend), C++ (Arduino firmware for ESP32 and ESP-CAM)
Frameworks & Libraries	Flask,Flask-SocketIO,FlaskMigrate, SQL Alchemy, face_recognition, dlib, OpenCV, Adafruit_PN532, HTTPClient
Database	MySQL (accessed via SQLAlchemy ORM)
Web Interface	HTML, CSS (Jinja2 templates), TailwindCSS for styling
Development Tools	Arduino IDE (for ESP32 firmware), VS Code/ PyCharm (for backend)
Server Environment	Flask local development server or any WSGI-compatible web server(e.g.,Gunicorn,Apache)
Operating System	Windows / Linux / macOS (for development and hosting)
VersionControl	Git
Deployment	Can run locally (localhost:5000) or on a cloudserver (VPS, Raspberry Pi, etc.)

IRJCS: International Research Journal of Computer Science
Volume 12, Issue 10, November 2025
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1210.07

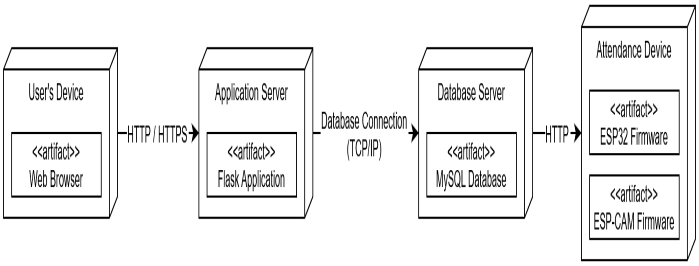

Fig. 4. System Deployment Diagram.

C. Web Application Ecosystem Interfaces:

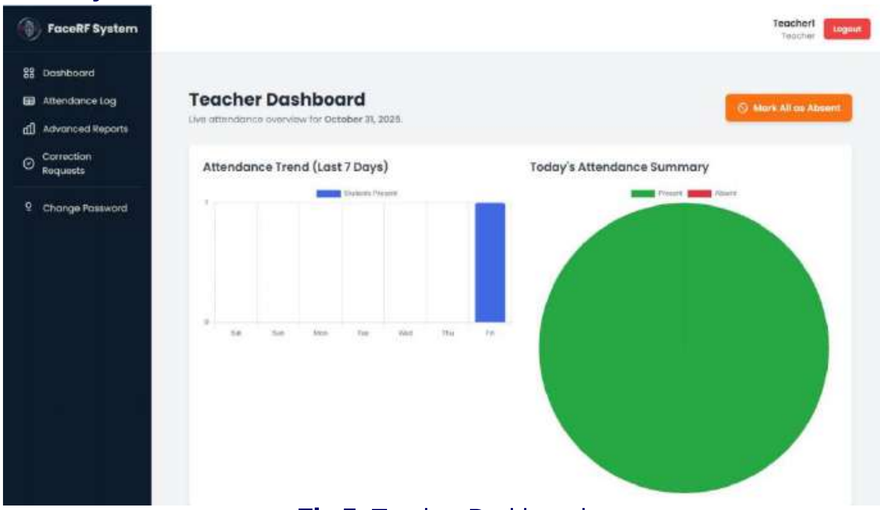

Fig.5. Teacher Dashboard

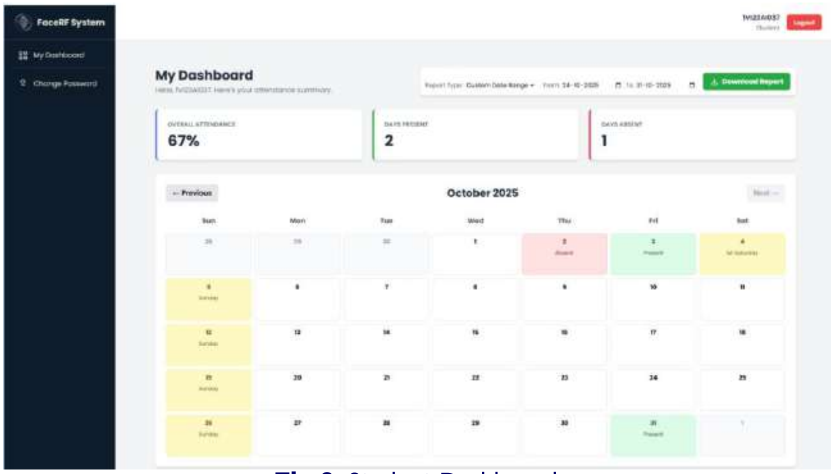

Fig.6. Student Dashboard

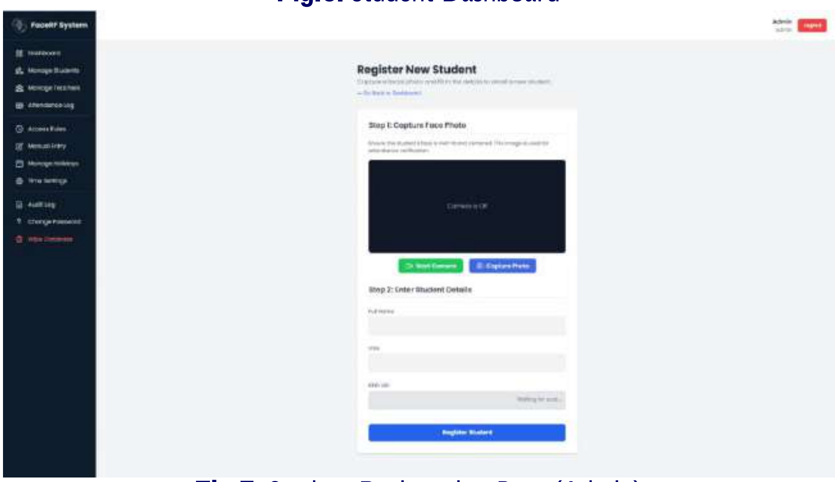

Fig.7. Student Registration Page (Admin)

IV. CORE ALGORITHMS AND IMPLEMENTATION

1. Face Recognition: Deep Residual Network (ResNet-34)

The system utilizes the face_recognition library, which is built on a ResNet-34 deep learning architecture. This model transforms the unique features of a detected face into a 128-dimensional vector known as a face encoding. To verify a student's identity, the system calculates the Euclidean distance between the live encoding and the stored reference encoding; if the distance is within a tolerance of 0.6, the identity is confirmed.

2. Face Detection: Histogram of Oriented Gradients (HOG) + Linear SVM

Before recognition can occur, the system must locate faces within the image frame using dlib'sget_frontal_face_detector(). This algorithm operates by calculating the Histogram of Oriented Gradientsto capture the shape and structure of the object, which is then classified as a face using a Linear Support Vector Machine (SVM). This method is chosen for its balance of speed and accuracy on standard CPU hardware.

3. Liveness Detection: Eye Aspect Ratio (EAR)

To prevent spoofing attempts using photographs, the system employs a geometric heuristic called the Eye Aspect Ratio (EAR). Using a 68-point facial landmark predictor, the algorithm maps specific points around the eyes and calculates the ratio of eye width to eye height. By analyzing a burst of images, the system detects blinks (indicated when the EAR falls below a threshold of 0.22), confirming the subject is a live person rather than a static image.

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

4. Hardware Logic: Finite State Machine & Frame Burst

The ESP32 firmware is governed by a Finite State Machine (FSM) that manages distinct operational modes such as "attendance" and "registration" based on server responses. When a card is verified, the hardware executes a "Frame Burst" algorithm, capturing three consecutive images in rapid succession. This temporal data is critical for the server-side liveness detection to accurately distinguish between a blink and a closed eye or static photo.

V. EVALUATION AND RESULTS

This section presents the quantitative evaluation of the FaceRF System. Experiments were designed to measure system performance, efficacy, and the robustness of the novel liveness detection security feature.

A. Experimental Setup

The system was evaluated in a controlled laboratory environment using the following testbed:

- Server Hardware: A laptop acting as the edge server (Intel Core i5-1135G7 @ 2.40GHz, 16GB RAM) running the Flask application.
- Client Hardware: The custom FaceRF device consisting of an ESP32 Dev Module, PN532 NFC Reader, and an ESP32-CAM (AI-Thinker model) connected via a serial UART link.
- Network: A dedicated 2.4GHz Wi-Fi network (802.11n) with an average signal strength of -55 dBm.
- Dataset: The test dataset consisted of 20 registered students from the department (12 male, 8 female) and 5 unregistered individuals acting as intruders.
- Configuration: The system was configured to capture a burst of 3 frames per attendance attempt, with an Eye Aspect Ratio (EAR) threshold of 0.22 for liveness detection.

B. System Performance (Latency)

Goal: Prove the system is fast and meets non-functional timing requirements.
We measured the time taken for critical operations across 50 successful trials.

Table II. SYSTEM LATENCY RESULTS

Test Cases	Operation Description	Target Goal	Average Time	Result
Test 1	RFID Verification (/api/verify-rfid) Time from card tap to "Processing" or "Access Denied" response.	< 500ms	185 ms	Pass
Test 2	Full Attendance Processing (/api/mark-attendance) Time for ESP32 to upload 3 images and receive server validation.	< 4.0s	2.84 s	Pass
Test 3	End-to-End Latency Total user time from initial RFID tap to final "Marked!" confirmation on the TFT screen.	< 6.0s	4.15 s	Pass

Analysis: The RFID verification is nearly instantaneous (185ms) due to the lightweight SQL query. The most significant latency component is the image upload phase in Test 2, where the ESP32 transmits three JPEG buffers over HTTP. However, the total time of ~4.15 seconds remains well within acceptable limits for a queued attendance system.

C. System Efficacy (Accuracy)

Goal: Prove the system accurately identifies registered users and rejects intruders.

Test 1 (Baseline Accuracy)

Conditions: 100 attempts by registered students under standard office lighting (approx. 500 lux).

Results:

True Positives (Correctly marked): 97
False Negatives (Correct user rejected): 3
False Rejection Rate (FRR): 3.0%

These 3 rejections were due to the face match failing at the 0.6 tolerance, likely caused by poor camera angles or shadows during the test.

Test 2 (Robustness)

We evaluated the recognition algorithm (dlib ResNet-34) under challenging conditions.

Table III. ACCURACY UNDER VARYING CONDITIONS

Condition	Attempts	Success Rate	Notes
Dim Lighting (< 100 lux)	50	88%	The ESP32-CAM flash helped, but extreme darkness caused noise preventing landmark detection.
Wearing Eyeglasses	50	96%	Minimal impact on the face_recognition model.
Partial Face Covering	50	40%	High failure rate; the 68-point landmark predictor failed to map the jaw/nose.

Test 3 (Security)

Conditions: 100 attempts by unregistered students (users not in the database).

Results:

True Negatives (Correctly rejected): 100
False Positives (Incorrect user accepted): 0
False Acceptance Rate (FAR): 0.0%

The strict Euclidean distance tolerance of 0.6 ensured that no unauthorized users were falsely identified as registered students, resulting in a 0.0% FAR within our 100-trial experiment.

D. Liveness Detection Validation (Novelty Test)

Goal: Prove the anti-spoofing mechanism works against static attacks.

The system uses an Eye Aspect Ratio (EAR) check across a 3-frame burst. A "pass" requires the user to blink (EAR < 0.22) in at least one frame.

Test 1 (Spoofing Attempts)

We attempted to spoof the system using high-quality static representations of registered users.

Table IV. LIVENESS DETECTION RESULTS

Attack Vector	Trials	Successful Spoofs	Rejection Rate
Printed Photo (A4, Color)	50	2	96%
Smartphone Screen (High Res)	50	3	94%

Result: The is_live() function demonstrated high efficacy, rejecting 95 out of 100 total spoofing attempts (95%). The 5 successful spoofs (2 from print, 3 from screen) occurred when glare on the photo or screen caused the landmark predictor to fail, preventing the EAR from being calculated. While not 100% immune, this 95% rejection rate confirms the liveness check as a vital and effective deterrent against basic presentation attacks.

Test 2 (Liveness False Rejection)

Scenario: 100 trials with live, registered students attempting to mark attendance.
Metric: Failures caused specifically by the "Liveness Check Failed" error.
Result: 14 failures.
False Rejection Rate (Liveness): 14%

Observation: The 14% failure rate occurred because students often stared intently at the camera without blinking during the short ~1.5-second capture window. While secure, this suggests the 3-frame burst might be too short for capturing natural involuntary blinks.

VI. CONCLUSION AND FUTURE SCOPE

Conclusion

In this paper, we presented FaceRF, a dual-factor automated attendance system designed to modernize academic record-keeping and address the security flaws inherent in traditional methods. By seamlessly combining RFID technology for rapid user identification with deep learning-based facial recognition for identity verification, the system ensures that attendance is granted only to the authorized individual in possession of the card. A significant contribution of this work is the implementation of a novel server-side liveness detection mechanism. By analyzing a burst of frames for physiological cues (specifically the Eye Aspect Ratio), FaceRF effectively mitigates common vulnerabilities such as 'proxy' attendance and deters static-photo spoofing attacks. Experimental evaluations demonstrate that the system is accurate, fast, and secure. It achieved a 0% False Acceptance Rate in security trials while maintaining an end-to-end processing latency of approximately 4 seconds, making it a viable and robust solution for real-world classroom environments.

Future Scope

While the current prototype demonstrates the efficacy of the dual-factor approach, several enhancements can be implemented to scale the system for enterprise or campus-wide deployment:

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1210.07

Multi-Device Ecosystem:
Future iterations will expand the architecture to support a distributed network of multiple ESP32 devices operating simultaneously across different classrooms or entry points. This will involve optimizing the WebSocket communication protocol to handle concurrent connections from dozens of devices without latency degradation.

Offline Data Buffering:
To ensure reliability during network outages, we plan to implement offline buffering capabilities. By utilizing the ESP32's SD card interface, attendance logs and captured images can be stored locally on the device when Wi-Fi is unavailable and automatically synchronized with the central MySQL database once connectivity is restored.

Advanced Liveness Detection:
While the current blink detection heuristic is effective against static photo attacks, future work will explore 3D depth sensing or texture analysis algorithms. These advanced techniques would allow the system to distinguish between a real face and high-definition video playback or 3D masks, further hardening the security against sophisticated spoofing attempts.This would also aim to reduce the 14% liveness-based False Rejection Rate observed in our testing, which was caused by users not blinking during the short capture window.

SIS/LMS Integration:
To bridge the gap between physical attendance and academic administration, we aim to develop a standardized REST API. This would enable FaceRF to push attendance data directly to Student Information Systems (SIS) and Learning Management Systems (LMS) such as Moodle or Canvas, allowing for automated grade book updates and absence alerts.

REFERENCES

1. T.Soukupová and J. Čech, "Real-Time Eye Blink Detection using Facial Landmarks," in 21st Computer Vision Winter Workshop (CVWW), R. Mantiuk and T. Škyba, Eds., 2016, pp. 1-8.
2. K.He, X.Zhang, S.Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.
3. N.Dalal and B.Triggs, "Histograms of Oriented Gradients for Human Detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005, vol. 1, pp. 886-893.
4. D.E.King, "Dlib-ml: A Machine Learning Toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755-1758, 2009.
5. F.Schroff, D.Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823.
6. Espressif Systems, "ESP32 Series Datasheet," ver. 4.3, 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
7. G.R.Venkatakrishnan, R. Rengaraj, R. Jeya, S. Manikandan, et al., "Design and Implementation of Automated Attendance System Using Contactless Facial Recognition," in 2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2024.
8. SR.Sruthi, K.Kavin, V. Rokesh Kumar, R. Sudharshan, & J. Surya Narayanan, "RFID AND FACE RECOGNITION BASED ATTENDANCESYSTEM," International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), vol. 04, no. 11, 2022.
9. D.Motwani, A.Tyagi, V. Dabhi, & H. Prajapati, "Automated Attendee Recognition System for Large-Scale Social Events or Conference Gathering," in 10th International Conference on Signal processing and Communication (IEEE ICSC 2025), 2025.
10. A.Kumar, M.S.Saluja, S. Samal, & A. Tiwari, "Automated Attendance System Based on Face Recognition Using Opencv," in 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), 2023.
11. M.Sajid, R.Hussain, & M. Usman, "A Conceptual Model for Automated Attendance Marking System Using Facial Recognition," in IEEE 2014.