# Attendance Monitoring System Through RFID and Face Detection

**[1]Pratham CS, [2]Vaibhav M, [3]Suhrid Sen, [4]Kapil N, [5]Priya Badrinath**

[1,2,3,4]Undergraduate Student, [5]Associate Professor

Department of Computer Science and Engineering, PES University

PES University, Bengaluru, India

[1]192003pratham@gmail.com, [2]vaibhavmullaguri520@gmail.com

*Abstract*—For effective class attendance management, reducing human error, and increasing transparency, an automated atten dance system is a crucial tool. In order to automate the procedure, this paper discusses a system that was created using Real-Time Face Recognition (RTFRS) and Radio Frequency Identification (RFID). The system is implemented with a Flutter-based web interface for administrative functions, MongoDB as the database for student records, and Python as the main language for handling logic. In order to ensure correct attendance tracking for just those students present for the whole class period, the methodology includes real-time monitoring of student entries and exits against preset class period times. Through a dedicated web interface, the technology also makes it easier to handle student profiles, RFID tags, and facial recognition data

*Index Terms*— RFID, RTFRS, Python, MongoDB, Flutter, Real-time monitoring

## I. INTRODUCTION

Technologies like RFID and face recognition systems are being investigated as a result of the growing demand for automated solutions in educational institutions to control attendance. Conventional manual attendance techniques are laborious, prone to mistakes, and frequently do not allow for real-time monitoring. To overcome these obstacles, this study presents an automated system that combines RFID and the Real-Time Face Recognition System (RTFRS). By recording entries and exits and comparing them with a timetable database, the suggested method guarantees precise and effective student attendance tracking. When students enter the classroom, the system records their presence and keeps track of them until the conclusion of each allotted period. Students are only recorded as present if they remain until the end of the class period. This approach improves reliability and drastically lowers the likelihood of proxy attendance.

## II. LITERATURE SURVEY

The capacity of RFID and face detection to increase accuracy and remove human error has drawn a lot of attention to the development of attendance monitoring systems. While lowering the possibility of proxy attendance, the combination of RFID and facial detection technology offers a dependable way to guarantee real attendance tracking.

A smart attendance system that tracks attendance in real-time using RFID and Face ID technologies was introduced by R. A. et al. [1]. Their research showed how effective it is to combine several identifying techniques to improve user comfort and system dependability.

In order to create a long-lasting campus-wide attendance tracking system, P. Agarwal et al. [2] suggested a framework that combines RFID, facial detection, and Ethernet networks. Their strategy highlights how crucial connectivity and smooth communication are for extensive deployments.

Using a dual-system approach that combined facial recognition with ID card detection, Kushal et al. [3] made use of TensorFlow and OpenCV. Their approach demonstrates how machine learning may be used to build reliable systems that can manage challenging real-world situations.

Z. Zhang et al. [4] created a technique for multi-angle face identification in the face recognition field that generates frontal faces. This method tackles issues with different illumination and facial orientations, which are critical for efficient attendance systems

X. Jia et al.'s work [5], which examined RFID's possibilities in the Internet of Things space, demonstrates the technology's scalability and versatility. Their results highlight how the technology might improve real-time tracking in attendance systems. In a similar vein, Ajay Joshi et al. [6] examined an RFID-Based Attendance System with an emphasis on its usefulness and affordability in educational settings.

Lastly, S. Haji and A. Varol [7] introduced a Real-Time Face Recognition System (RTFRS) for surveillance, showing how its improved security and real-time decision-making may be used for attendance monitoring.

The integration of face detection and RFID into attendance systems is made possible by these investigations. By integrating Python, FastAPI, and MongoDB with RFID and Real-Time Face Recognition, the current study expands upon this framework and produces a scalable and effective solution designed for institutional settings.

## III. SYSTEM ARCHITECTURE

The system primarily comprises four main components that entail various functionalities in attaining efficient and smooth student management and attendance tracking besides identity verification: the RFIDs, RTFRS module, Database and Back-end, and the admin interface.

### III.A RFID module

The RFID module handles students' check-ins and check-outs processes to be used in Radio-Frequency Identification technologies. Functionality: Each student is given an RFID card with a unique ID. The student's entry or exit is detected by the system when the student swipes or taps the RFID card on the reader. Real-time Processing: The module records the event in real-time, marking attendance or updating the check-out time. This ensures that there is an accurate record of each student's presence. Automation: This module automates the check-in and check-out of students. Thus, it minimizes manual effort, human error, and accelerates the recording of attendance.

### III.B RTFRS Module

The Real-Time Facial Recognition System module makes use of facial recognition technology that authenticates the identity of the students.

Purpose: For added security and the verification of the authenticity of check-ins by students. Technology Stack: Advanced algorithms for machine learning, can be built which would look at the student's images with face recognition in congruence with the previously registered images of students within the system database Process: Students use cameras to capture their faces concerning the system. This happens at entry to campus. RTFRS module will then process this captured face to align with a student record on the database Once validated, the attendance status gets updated. Proxy attendance will not be there after this. Ensures that the system is safe since it provides a less-intrusive fast identity verification process. The overall system reliability will be high while cross-verifying attendance using RFID

### III.C Database and Backend

The central nervous of this system is the Database and Backend which manages, stores, and performs actions concerning the data. The backend utilizes the code from Python, and the database used is MongoDB. Database(MongoDB): It encompasses and maintains the core records- student profiles, students attendance records, and class timings. Facilitates a flexible, scalable solution for efficiently managing large volumes of data. It supports real-time updates to ensure consistency and reliability of stored information. Backend (Python): It is the processing engine that executes operations such as attendance updates, data retrieval, and schedule management. It facilitates smooth communication between the various modules (RFID, RTFRS, and Admin Interface). This holds business logic to process the data without any inaccuracies or ineffectiveness. It comprises all the major features of it: Real time logging for attendance Fetch student records and attendance rapidly Protecting and dealing with private student data securely

### III.D Admin interface

This is the interface developed through Flutter in which Admin has control over the monitoring and administration of the system. Features and Functionality : Schedule Management: The class schedule can be created, updated, or modified by the administrator. The system is synchronized with real-time timetables. Attendance Tracking: Real-time attendance logs can be viewed. Detailed reports of student attendance can be generated for analysis. Student Information Management: Add, update, or delete student profiles. Maintain accurate and up-to-date records. Performance Analytics: The interface can be extended to provide analytics such as attendance trends, anomalies, and student participation metrics. User Experience: The Flutter design is based on the feature of a smooth, responsive interface with Flutter. The system functionalities remain easily accessible to administrators. It can be deployed multiple times on various platforms, whether it is a mobile phone, tablet, or desktop machine. The four modules are

RFID Module, RTFRS Module, Database and Backend, and Admin Interface. These four modules combine to form a system that streamlines attendance tracking, student management, and identity verification. Each module is very critical in the system being secure, accurate, and efficient.

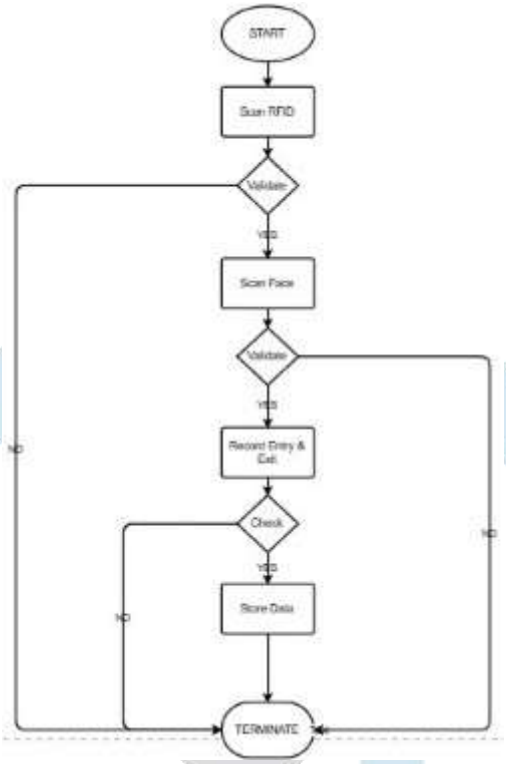


Figure 1: Flowchart of Attendance Monitoring System

## IV. METHODOLOGY

This chapter develops on the design and workflow of the automated attendance monitoring system that includes its core functionalities, RFID integration, and RTFRS, along with the application of Python libraries for a reliable and efficient solution. Methodology is divided into three primary parts: RFID-based entry and exit logging, integration of the timetable with verification of attendance, and face recognition for security.

### IV.A RFID-Based Entry and Exit Logging

The RFID-based subsystem is essential in recording the time of entry and exit to ensure that attendance is tallied correctly. The steps below detail its implementation process:

- RFID Tag Registration:Each student will be assigned a unique RFID tag, which will be associated with his/her profile stored in a database using Mongo Db-For fetching and storing student info will interface smoothly and fast will have the help of pymongo library v4. 10.1. This allows high performance coupled with scalability and reliability. dealing with large number of Student records.

- Real-Time Data Capture:In a real-time manner, records and events are captured regarding a student. The system is constantly tracking the RFID events produced by backend services developed in FastAPI version 0.111.1 with an asynchronous server framework uvicorn version 0.30.3. Every read of the RFID causes an event. Python scripts catch this event. The environment variables, including database passwords and API keys, are well handled in a secure method by python-dotenv, the version used is 1.0.1 that provides excellent configuration management.

- Data Processing: The scanning of an RFID will trigger a specific endpoint in the FastAPI application. The endpoint will log the entry or exit event and check for validity against the current class schedule. In this way, attendance can be marked only for the students when they are present during their scheduled classes and avoids errors or unauthorized claims.

### IV.B Timetable Integration and Attendance Verification

The system integrates a well-structured timetable in a way that ensures proper marking of attendance based on the presence of students for particular periods.

- Timetable Management: The timetable data is stored in MongoDB as JSON documents, and access and management are done by the pymongo library. Every entry in the timetable is specified with the start time and end time for a particular period.

- The RFID-based presence logs are cross-referenced with this schedule by the system. - Attendance is validated by checking whether the student's RFID remains active till the end of the scheduled class period.

- Entry Validation Logic: Pydantic (version 2.8.2) is used by FastAPI's routes to handle requests and validate data. This guarantees that incoming facial recognition and RFID data are processed safely and in the correct forms.

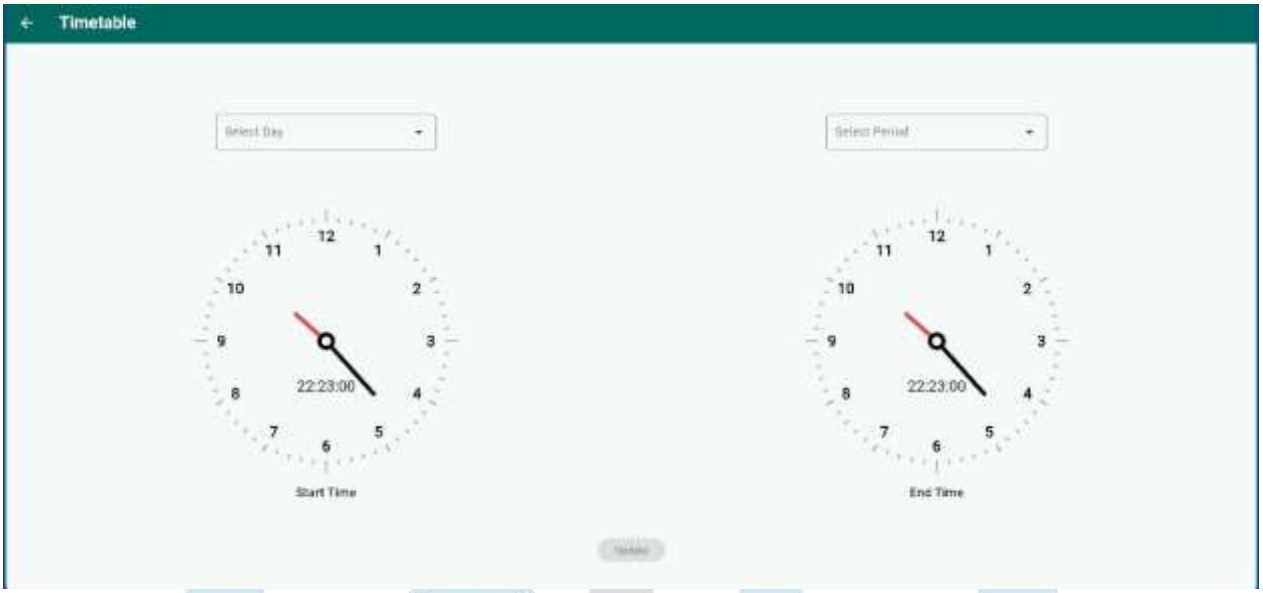


Figure 2: Web Interface of Timetable

## IV.C Face Recognition for Enhanced Security

In the prevention of unauthorized entry, security will be enhanced through the installation of real-time facial recognition in combination with RFID-based logging. This two-factor verification system gives another layer of authentication.

- Image Capture and Processing:The system captures and pre-processes real time images of faces using library opencv-contrib-python, version 4.10.0.84. - Opencv is used for facial detection and feature identification. This is to ensure that the recognition is accurate enough, even under changing lighting conditions. - The Pillow library, version 11.0.0 is used to further process the image as per requirement, such as re-sizing or enhancement of quality of the image before going for analysis.

- Facial Encoding and Comparison:When a photograph is captured, it is converted to a series of numerical data that represents facial information. These conversions are handled by numpy version 1.26.4, which facilitates working with matrices for comparison purposes when matching the captured face information against the stored records in the database. - It validates the identity of a student by comparing the new conversion with the one associated with his profile in MongoDB. - This way, only authenticated students are marked as present.

- JWT Authentication: This system uses pyjwt version 2.8.0 to secure communication between the web application and backend. The system uses JWT for: - Encrypting and decrypting session data to verify session validity. Ability to ensure access to alteration of attendance information for users who are authorized users - Block unauthorized attempts at trying to manipulate the system


Figure 3: Web Interface of Students info

## V. IMPLEMENTATION DETAILS

### V.A Libraries and Their Roles

- python-dotenv (1.0.1): Manages environment variables, keeping sensitive information like database credentials secure.
- fastapi (0.111.1): Powers the RESTful API endpoints for handling RFID and face recognition data.
- uvicorn (0.30.3): Serves as the ASGI server to run the FastAPI app for real-time data handling.
- pymongo (4.10.1): Interfaces with MongoDB, facilitating CRUD operations for managing student profiles, attendance logs, and the timetable.
- pydantic (2.8.2): Ensures robust data validation for incoming API requests, enforcing strict type-checking and validation rules.
- pyjwt (2.8.0): Manages token-based authentication, ensuring secure communication between the back-end and the Flutter web interface.
- opencv-contrib-python (4.10.0.84): Handles image capture and facial recognition processes.
- numpy (1.26.4): Supports complex numerical operations, especially in comparing and manipulating face encodings.
- pillow (11.0.0): Assists in processing images for verification tasks.

### V.B Detailed System Workflow

- Initial Setup:
  - The FastAPI-powered back-end API makes endpoints available that watch for facial recognition and RFID scan data.
  - Python-dotenv is used to load environment variables so that database credentials and configuration can be safely accessed.
- Entry and Exit Detection:
  - Pymongo is used to query the database for a student's profile when they scan their RFID tag. The student's entry is recorded with a timestamp if it is legitimate.
  - The system compares the current time with the end time of the class period for exits. The student is not granted attendance for the period if the exit timestamp comes before the conclusion of the period.

- Face Recognition Flow:
  - Capture and Encoding: OpenCV is used to capture images and numpy is used to process and encode them into numerical arrays. Prior to encoding, pillow improves image quality.
  - Comparison logic: Using numpy's matrix operations, new face data is compared to previously stored encodings. Attendance logs are triggered when matches are above a predetermined threshold, confirming identity.
- Real-Time Validation:
  - All interactions are managed by FastAPI, which validates payloads from the RFID and RTFRS modules using Pydantic. Pyjwt is used to maintain secure sessions.

## V.C Web Interface with Flutter

- Student Management: Students' profiles, including their face and RFID information, can be added or updated by administrators.
- Attendance Dashboard: Confirms student's presence for each period by displaying attendance logs.
- Timetable Interface: Permits class schedule changes, which Python updates instantly.

# VI. RESULTS AND DISCUSSION

## VI.A Performance Evaluation

Performance of the system was tested for three prominent parameters, that is accuracy, response time, and reliability.

- Accuracy: Facial recognition and RFID technology of the dual-factor authentication mechanism enhanced the overall accuracy of the system.The entire system was accurate up to a point of 95
- Response Time: The use of the Uvicorn as an asynchronous server to serve the request of the API improved real time performance. Experimental results indicated that the average response time of the system was below 100 ms for all the operations making it a smooth fast process.
- Reliability: Use of JWT-JSON Web Tokens through the library pyjwt eased more secure sessions in the system. It was an accessible system only to the authorized users; hence there were no manipulations present in attendance, and it maintained high reliability.


Figure 4: Database of Student documents


Figure 5: database of Student Attendance

## VI.B Challenges and Mitigation

There were certain problems during the development and deployment phases present in the system. Proper strategies were adopted to deal with the problems for better performance. Lighting and Facial Recognition

- Challenge: Unpredictable illumination made the accuracy of the Real-Time Facial Recognition System (RTFRS) degrade most of the time due to insufficient lighting in face detection. Mitigation: The input images have openCV preprocessing filters, thus improving face detection. Histological equalization and Gaussian smoothing of the image are made in order to get proper and reliable recognition for changing light intensity.
- Database Load: Challenge: It had extremely high loads in the database because of many reads/writes when there was extreme usage of the system. Mitigation: It used the Pymongo-managed index on key collections of MongoDB for indexing. This highly optimized the performance of queries along with the latency of fetching data. This is what made the database handle high loads of transactions smoothly.

## VI.C Summary of Results

With RFID integrated with facial recognition, and backend optimizations, the following results occurred:

- 95 percent accuracy
- Response times less than 100 ms
- Increased reliability by JWT security
- Reducing the problems presented by lighting and database performance

### Response Time

$$\text{Response Time} = \frac{\text{Total Time Taken for All Operations}}{\text{Number of Operations}} \quad (1)$$

This formula calculates the mean time per operation, helping to evaluate the system's real-time performance.

### Database Query Time

$$\text{Query Time} = \frac{\text{Total Query Time}}{\text{Number of Queries}} \quad (2)$$

This metric measures the average time taken for indexed database read/write operations, ensuring optimal database performance.

Table 1: System Performance Metrics

| Metric | Measured Value | Remarks |
|---|---|---|
| Accuracy (Overall) | 95% | Combined RFID and facial recognition. |
| Response Time | 8 s | Average processing time per operation. |
| Precision (Facial Rec.) | 94% | For facial recognition verification. |
| Recall (Facial Rec.) | 93% | System's ability to detect true faces. |
| FAR | 2.5% | False acceptance of unauthorized faces. |
| FRR | 3.0% | False rejection of authorized users. |
| Database Query Time | 55 ms | Indexed MongoDB read/write operations. |

## VII. TEST CASES

The test cases assess the Attendance Monitoring System's dependability and effectiveness using RFID and RTFRS in a range of circumstances. Under normal circumstances, the system can authenticate attendance in less than 8 seconds, even when there are minor changes like the presence of a beard or glasses. It does this when both RFID and facial recognition match. Invalid attendance is the result of mismatches, such as different faces or RFIDs, or the lack of facial data (e.g., wearing a cap or not using face recognition). Additionally, using RFID or face recognition alone produces erroneous results, underscoring the need for both elements for precise validation.

Table 2: Performance of Different Cases

| SL. NO. | CASE NAME | RFID | RTFR | VALIDITY | TIME |
|---|---|---|---|---|---|
| 1 | Normal | Yes | Yes | Valid | < 8 sec |
| 2 | Only RFID | Yes | No | Not Valid | 8 sec |
| 3 | Different Face | Yes | Yes | Not Valid | < 8 sec |
| 4 | Different RFID | Yes | Yes | Not Valid | < 8 sec |
| 5 | With Spectacles | Yes | Yes | Valid | < 8 sec |
| 6 | With Beard | Yes | Yes | Valid | < 8 sec |
| 7 | With Cap | Yes | No | Not Valid | 8 sec |
| 8 | Without Beard | Yes | Yes | Valid | < 8 sec |
| 9 | Only Face | No | No | Not Valid | Nil |

## VIII. CONCLUSION AND FUTURE WORKS

The RFID and face detection-based attendance monitoring system efficiently automates and streamlines the tracking of real-time students' attendance. The application of RFID technology in this system makes it further enhance its ability toward quick identi-

fication for security purposes while further reducing the chances of proxy while tracking in real time. Educational institutions is making use of tools and frameworks like Python, FastAPI, MongoDB with a Flutter-based web interface towards building a stable, efficient system, and scalable to enable better attendance management. Using two-factor authentication within this system improves the accuracy for reliability, thereby making an effective solution in real-life deployments.

- Mobile App Integration:Design a cross-platform mobile application for the administrator and the student. The application will be used in
  - Real-time attendance tracking
  - Sending alert messages that classes have not been attended
  - Easy management of schedules and reports
- Data Analytics: Provide data analytics feature in order to track attendance over time. These comprise
  - Attendance Trend Tracking of the student to avoid frequent absence in classes
  - Generation of reports in making more-informed decisions
- Cloud Deployment: Move the system to cloud-based architecture to avail the benefits of scalability and access from remote locations. All of the above advantages will come:
  - Accessibility from numerous devices
  - Storage management would be improved
  - The performance will rise in case of heavy traffic.
- Advanced Face Detection: It would use deep learning models like Convolutional Neural Networks for better recognition of face This would enhance robustness across various scenarios. Including lighting and partial occlusions and facial expression changes

This system is a great starting point for automatic attendance management but still has much room to do better. One possible future evolution of this feature would be mobile integration along with further analytics advancements, deploying this on the cloud, such that this solution would still remain futuristic, scalable, and adaptable to the changing needs of an educational institution.

## References

[1] R. A, S. Brindha, S. S. B and G. A, ”Smart Attendance System Using RFID and Face ID,” 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2022, pp. 1-5, doi: 10.1109/IC3IOT53935.2022.9768003.

[2] P. Agarwal, V. K. Shukla, R. Gupta and S. Jhamb, ”Attendance Monitoring System Through RFID, Face detection and Ethernet Network: A Conceptual Framework for Sustainable Campus,” 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2019, pp. 321-325, doi: 10.1109/ISCON47742.2019.9036209.

[3] Kushal, Micky V, Kushal J, Charan M, Pappa. (2020). ID Card Detection with Facial Recognition using Tensorflow and OpenCV. 742-746. 10.1109/ICIRCA48905.2020.9183342, doi:10.1109/ICIRCA48905.2020.9183342

[4] Z. Zhang, H. Zhang, H. Liu, S. Xin, N. Xiao and L. Zhang, ”Frontal Face Generation Based Multi-angle Face Identification System,” 2021 International Conference on Computer, Control and Robotics (ICCCR), Shanghai, China, 2021, pp. 329-334, doi: 10.1109/ICCCR49711.2021.9349409.

[5] X. Jia, Q. Feng, T. Fan and Q. Lei, ”RFID technology and its applications in Internet of Things (IoT),” 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, China, 2012, pp. 1282-1285, doi: 10.1109/CECNet.2012.6201508.

[6] Ajay Joshi, Aman Ahmad, Arpit Saxena and Poonam Juneja, ”RFID Based Attendance System”International Journal for Modern Trends in Science and Technology, 7(01): 40-43, 2021 Copyright © 2021 International Journal for Modern Trends in Science and Technology ISSN: 2455-3778 online DOI: https://doi.org/10.46501/IJMTST0701009

[7] S. Haji and A. Varol, ”Real time face recognition system (RTFRS),” 2016 4th International Symposium on Digital Forensic and Security (ISDFS), Little Rock, AR, USA, 2016, pp. 107-111, doi: 10.1109/ISDFS.2016.7473527.