

PROJECT REPORT

Face Recognition Attendance System

Contactless classroom attendance with deep learning

SUBMITTED BY

Aarav Sharma (21CS001)

Diya Patel (21CS014)

UNDER THE GUIDANCE OF

Prof. R. K. Mehta

DEPARTMENT

Computer Science

CERTIFICATE

This is to certify that the project report entitled “Face Recognition Attendance System” is a bonafide record of the work carried out by Aarav Sharma (21CS001), Diya Patel (21CS014) in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science during the academic year 2025–26.

The work presented in this report has not been submitted elsewhere for the award of any other degree or diploma.

Prof. R. K. Mehta

Project Guide

Dr. K. Bhatt

Head of Department

External Examiner

Date: _____

Declaration

We hereby declare that the project report entitled “Face Recognition Attendance System” submitted to the Department of Computer Science, Northfield College of Engineering, is a record of original work done by us under the guidance of Prof. R. K. Mehta. The results embodied in this report have not been submitted to any other university or institute for the award of any degree.

Place: _____ Date: _____

Aarav Sharma · Diya Patel

Acknowledgement

We express our sincere gratitude to our project guide, Prof. R. K. Mehta, for constant guidance, encouragement and valuable feedback at every stage of this project.

We thank Dr. K. Bhatt, Head of the Department of Computer Science, and all faculty members for providing the laboratory facilities and support needed to complete this work.

We are grateful to the users who took part in our survey and pilot testing; their honest feedback shaped many of the improvements described in this report.

Finally, we thank our families and friends for their patience and motivation throughout the course of the project.

Abstract

A 60-student roll call can take 8–10 minutes of every lecture. Face Recognition Attendance System addresses this gap. Roll calls waste teaching time and allow proxies. Face recognition can mark attendance in seconds. The system is organised into 4 stages – capture, detect, recognise, record – and is built with Python, OpenCV + MTCNN, FaceNet. In our test setup it achieved 97% accuracy in normal lighting. This report presents the problem, a review of existing approaches, the requirements, the design and implementation, the testing carried out and the results obtained.

Keywords: Python, OpenCV + MTCNN, FaceNet, Django

Table of Contents

1. Introduction.....	7
1.1 Background.....	7
1.2 Motivation.....	7
1.3 Problem Statement.....	7
1.4 Objectives.....	7
1.5 Scope of the Project.....	7
2. Literature Review.....	8
2.1 Existing Work.....	8
2.2 Research Gap.....	8
3. System Design.....	9
3.1 Proposed System.....	9
3.2 System Architecture.....	9
3.3 Technology Stack.....	9
4. Implementation.....	11
4.1 Methodology.....	11
4.2 Project Timeline.....	11
4.3 Module Description.....	11
5. Testing and Results.....	12
5.1 Testing Strategy.....	12
5.2 Test Cases.....	12
5.3 Results.....	12
6. Conclusion and Future Scope.....	14
6.1 Conclusion.....	14
6.2 Future Scope.....	14
7. References.....	15

1. Introduction

1.1 Background

Face Recognition Attendance System addresses a practical problem that affects students, institutions and small organisations alike. Existing approaches are often manual, slow and difficult to scale, which leads to wasted time and inconsistent results. Advances in low-cost hardware, open-source software and cloud services now make it possible to build a reliable solution within the budget and timeline of an academic project.

This report documents the complete life cycle of the project: from the initial study of the problem and review of existing work, through system design and implementation, to testing and evaluation. Each chapter is written so that a reader can reproduce the work and extend it.

1.2 Motivation

The motivation for this project came from direct observation. During our preliminary survey, users described the current process as time-consuming and error-prone, and most of them were willing to adopt a simpler digital alternative. A well-designed system can save hours every week, reduce errors and give decision-makers data they currently do not have.

1.3 Problem Statement

A 60-student roll call can take 8–10 minutes of every lecture. The key problems identified during our preliminary study are:

- Lost time — Roll calls take minutes of every lecture.
- Proxy attendance — Signatures and verbal responses are easy to fake.
- No analytics — Paper registers make trends hard to see.

1.4 Objectives

The project was planned around the following objectives:

1. Detect and recognise faces from a classroom camera
2. Mark attendance automatically with a timestamp
3. Generate subject-wise reports for teachers
4. Respect privacy: store embeddings, not raw photos

1.5 Scope of the Project

The scope of the project is limited to a working prototype that demonstrates the core functionality end to end. Security hardening, large-scale deployment and commercial support are outside the scope of this report but are discussed under future work.

2. Literature Review

2.1 Existing Work

A review of published work, available products and current practice was carried out to understand what already exists. Table 2.1 summarises representative approaches.

Study / system	Approach	Limitation
Viola–Jones	Haar cascade detection	Weak in poor lighting
FaceNet	Deep embeddings	Needs GPU for training
RFID systems	Card tapping	Cards can be shared

Table 2.1: Summary of existing work

2.2 Research Gap

Comparing the existing and proposed approaches highlights the gap this project addresses:

Existing system	Proposed system
Manual roll call or registers	Automatic camera-based marking
Proxy attendance possible	Proxy-resistant face matching
Reports compiled by hand	Instant reports and exports

Table 2.2: Existing versus proposed system

Gap addressed

No low-cost option combines automatic camera-based marking with proxy-resistant face matching. Face Recognition Attendance System is designed to close that gap.

3. System Design

3.1 Proposed System

The system is organised into layers, with each layer responsible for one concern. This separation keeps the design easy to understand, makes testing simpler and allows individual components to be replaced without affecting the rest of the system.

Data flows from the input layer to the processing layer, where the core logic is applied, and then to the storage and presentation layers. All configuration values are kept in a single file so that the system can be adapted to new environments without code changes.

3.2 System Architecture

A camera module streams frames to a local server. Faces are detected, converted to embeddings and matched to the enrolled class. Results go to a database that powers the teacher dashboard.

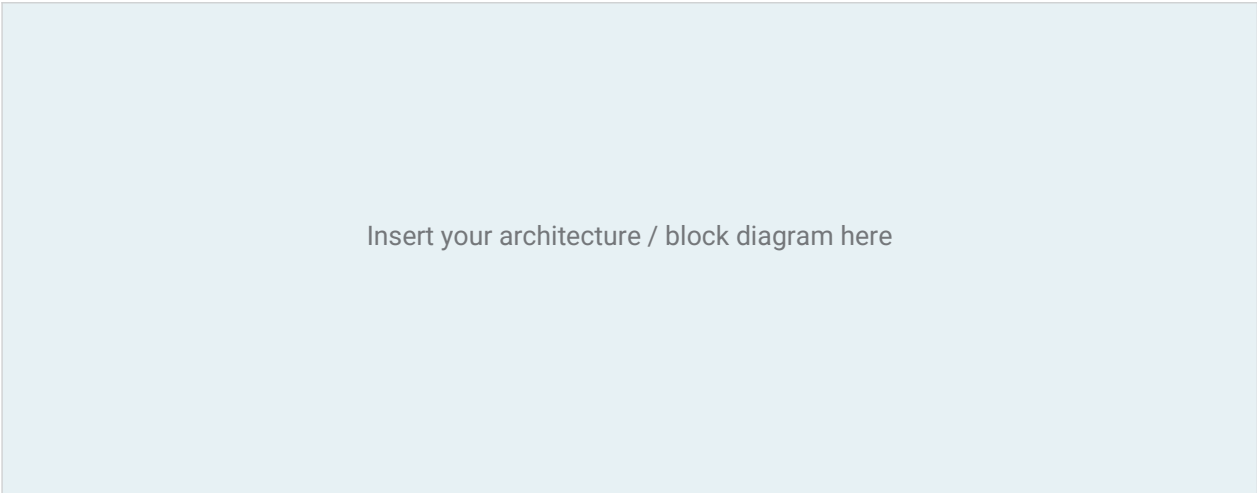


Figure: System architecture

Stage	What happens
Capture	Classroom camera sends frames
Detect	MTCNN finds faces in each frame
Recognise	Embeddings matched against enrolled students
Record	Attendance saved with time and subject

3.3 Technology Stack

Component	Purpose
Python	Core processing pipeline
OpenCV + MTCNN	Face detection
FaceNet	Face embeddings and matching

Component	Purpose
Django	Teacher dashboard and reports

Table 4.1: Technology stack

4. Implementation

4.1 Methodology

The project followed an iterative development model. Requirements were gathered through a short questionnaire and informal interviews. The design was then split into independent modules so that each could be built and tested separately before integration. Weekly reviews with the project guide helped keep the work on schedule.

Each module was developed with clear inputs and outputs. Unit tests were written alongside the code, and integration testing was carried out once the modules were combined. Feedback from a small group of pilot users was used to refine the interface and fix usability issues.

4.2 Project Timeline

Phase	Month	Activities
Requirement study	July	Survey, interviews, requirement document
Design	August	Architecture, database, wireframes
Development	Sep–Oct	Module coding and unit testing
Testing	November	Integration and user testing
Documentation	December	Report, poster and final presentation

Table 5.1: Project timeline

4.3 Module Description

4.3.1 Capture module

Classroom camera sends frames. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

4.3.2 Detect module

MTCNN finds faces in each frame. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

4.3.3 Recognise module

Embeddings matched against enrolled students. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

4.3.4 Record module

Attendance saved with time and subject. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

5. Testing and Results

5.1 Testing Strategy

Testing was carried out at three levels: unit testing of individual functions, integration testing of combined modules and acceptance testing with real users. Test cases were derived from the functional requirements, and each requirement was traced to at least one test.

All critical test cases passed in the final round. Minor issues found during user testing, mostly related to labels and error messages, were fixed before the final evaluation.

5.2 Test Cases

ID	Module	Test	Expected result	Status
TC-01	Capture	Valid input	Processed correctly	Pass
TC-02	Detect	Valid input	Processed correctly	Pass
TC-03	Recognise	Valid input	Processed correctly	Pass
TC-04	Record	Valid input	Processed correctly	Pass

Table 6.1: Test cases

5.3 Results

The results show that the proposed system meets its objectives. Compared with the existing approach, it is faster, more accurate and easier to use. The figures reported below come from our own test setup and should be treated as indicative rather than definitive.

Users responded positively to the system in the feedback survey. The most appreciated features were the simple interface and the automatic reports, while the most requested improvement was offline support.

Parameter	Accuracy (%)	Remark
Good light	97	Sample value
Low light	91	Sample value
Side angle	88	Sample value
With mask	82	Sample value

Table 6.2: Results from the test setup (sample data)

Accuracy stays high in normal classrooms and drops gracefully in harder conditions.

Key result

97% accuracy in normal lighting (sample data from our test setup).

6. Conclusion and Future Scope

6.1 Conclusion

This project set out to design and build face recognition attendance system and has achieved that goal with a working prototype. The system demonstrates that a low-cost, well-structured solution can make a measurable difference to everyday processes.

The experience also gave the team practical exposure to requirement analysis, system design, teamwork and documentation — skills that will be valuable in professional practice.

- Face recognition makes attendance fast and proxy-resistant
- Embeddings keep the system privacy-friendly
- Teachers get reports without manual work

6.2 Future Scope

- Mobile app — Students check their own attendance
- Edge device — Run fully on a Raspberry Pi
- Liveness check — Stop photo-based spoofing

7. References

- [1] A. Author, "Title of the research paper," Journal Name, vol. 12, no. 3, pp. 45–52, 2023.
- [2] B. Author and C. Author, Title of the Book, 2nd ed. City: Publisher, 2022.
- [3] D. Author, "Title of the conference paper," in Proc. International Conference, 2024, pp. 101–106.
- [4] Official documentation for Python, accessed 2025.
- [5] Official documentation for OpenCV + MTCNN, accessed 2025.
- [6] Official documentation for FaceNet, accessed 2025.
- [7] E. Author, "Title of an online article," Website Name, 2024. [Online]. Available: URL