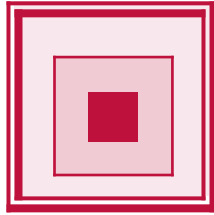


NORTHFIELD COLLEGE OF ENGINEERING, PUNE

Department of Computer Science



A Project Report on

Face Recognition Attendance System

Contactless classroom attendance with deep learning

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology

SUBMITTED BY

Ananya Iyer (21IT031)
Rohan Gupta (21EE008)
Sneha Nair (21CE017)
Arjun Rao (21CA005)

UNDER THE GUIDANCE OF

Dr. P. Srinivasan

DEPARTMENT

Computer Science

Session 2025-26

CERTIFICATE

This is to certify that the project report entitled “Face Recognition Attendance System” is a bonafide record of the work carried out by Ananya Iyer (21IT031), Rohan Gupta (21EE008), Sneha Nair (21CE017), Arjun Rao (21CA005) 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.

Dr. P. Srinivasan

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 Dr. P. Srinivasan. The results embodied in this report have not been submitted to any other university or institute for the award of any degree.

Place: _____ Date: _____

Ananya Iyer • Rohan Gupta • Sneha Nair • Arjun Rao

Acknowledgement

We express our sincere gratitude to our project guide, Dr. P. Srinivasan, 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

Introduction.....	10
1.1 Background.....	10
1.2 Motivation.....	10
1.3 Problem Statement.....	10
1.4 Objectives.....	10
1.5 Scope of the Project.....	11
1.6 Organisation of the Report.....	11
Literature Review.....	12
2.1 Existing Work.....	12
2.2 Research Gap.....	12
Requirement Analysis.....	14
3.1 Requirement Gathering.....	14
3.2 Functional Requirements.....	14
3.3 Non-functional Requirements.....	14
3.4 Hardware and Software Requirements.....	15
3.5 Feasibility Study.....	15
3.6 Use Cases.....	16
System Design.....	17
4.1 Proposed System.....	17
4.2 System Architecture.....	17
4.3 Technology Stack.....	18
4.4 Data Flow.....	18
4.5 Data Design.....	19
4.6 User Interface Design.....	19
Implementation.....	21
5.1 Methodology.....	21
5.2 Project Timeline.....	21

5.3 Module Description.....	21
5.4 Key Code Listings.....	22
5.5 Challenges Faced.....	23
Testing and Results.....	24
6.1 Testing Strategy.....	24
6.2 Test Cases.....	24
6.3 Results.....	24
6.4 Outcomes Against Objectives.....	25
Results and Discussion.....	26
7.1 Observations by Period.....	26
7.2 Comparison with the Existing System.....	26
7.3 Discussion.....	27
7.4 Limitations.....	27
Project Management.....	28
8.1 Team Roles.....	28
8.2 Schedule (Gantt Chart).....	28
8.3 Risk Register.....	28
8.4 Budget Summary (sample).....	29
Deployment, Maintenance and Impact.....	30
9.1 Deployment Plan.....	30
9.2 Maintenance Schedule.....	30
9.3 Social, Ethical and Environmental Impact.....	30
Conclusion and Future Scope.....	31
10.1 Conclusion.....	31
10.2 Future Scope.....	31
References.....	32
Appendix A: User Manual.....	33
Appendix B: Project Diary.....	35
Appendix C: Glossary.....	37
Appendix D: Detailed Test Logs.....	38

Appendix E: Survey Questionnaire.....	40
Appendix F: Survey Responses (sample data).....	41

List of Tables

Table	Title
2.1	Summary of existing work
2.2	Existing versus proposed system
3.1	Functional requirements
3.2	Non-functional requirements
3.3	Hardware and software requirements
3.4	Estimated project cost (sample)
4.1	Technology stack
4.2	Data dictionary
5.1	Project timeline
6.1	Test cases
6.2	Results from the test setup (sample data)

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.

1.6 Organisation of the Report

- Chapter 2 reviews existing work and identifies the research gap.
- Chapter 3 analyses the requirements and the feasibility of the project.
- Chapter 4 describes the system design, architecture and data design.
- Chapter 5 explains the implementation of each module.
- Chapter 6 presents the testing strategy, test cases and results.
- Chapter 7 discusses the results and limitations.
- Chapter 8 covers project management: roles, schedule, risks and budget.
- Chapter 9 describes deployment, maintenance and impact.
- Chapter 10 concludes the report and outlines future work.

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

2.1.1 Viola–Jones

Viola–Jones follows an approach based on haar cascade detection. This works well in controlled conditions and shows that the core idea is practical. However, the approach is limited because it is weak in poor lighting. For our users this limitation matters, which is why the proposed system treats it as a design requirement rather than an afterthought.

2.1.2 FaceNet

FaceNet follows an approach based on deep embeddings. This works well in controlled conditions and shows that the core idea is practical. However, the approach is limited because it is needs GPU for training. For our users this limitation matters, which is why the proposed system treats it as a design requirement rather than an afterthought.

2.1.3 RFID systems

RFID systems follows an approach based on card tapping. This works well in controlled conditions and shows that the core idea is practical. However, the approach is limited because it is cards can be shared. For our users this limitation matters, which is why the proposed system treats it as a design requirement rather than an afterthought.

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.

Requirement Analysis

3.1 Requirement Gathering

Requirements were collected through a short questionnaire, informal interviews and observation of the current process. Respondents were asked how they handle the task today, what problems they face and what they would expect from a better system. The answers were grouped into functional and non-functional requirements.

3.2 Functional Requirements

ID	Module	Requirement	Priority
FR-01	Capture	Classroom camera sends frames	High
FR-02	Detect	MTCNN finds faces in each frame	High
FR-03	Recognise	Embeddings matched against enrolled students	Medium
FR-04	Record	Attendance saved with time and subject	Medium
FR-05	System	Detect and recognise faces from a classroom camera	Medium
FR-06	System	Mark attendance automatically with a timestamp	Medium
FR-07	System	Generate subject-wise reports for teachers	Medium

Table 3.1: Functional requirements

3.3 Non-functional Requirements

Quality	Requirement
Performance	Typical responses within 0 under normal load
Reliability	Runs continuously for at least 7 days in the pilot without manual restart
Usability	A new user completes the main task within 5 minutes without training
Security	Role-based access; passwords stored as salted hashes; data backed up daily
Maintainability	Modular code with configuration kept in one file and documented interfaces
Cost	Hardware and hosting within the budget of a small institution (15 s)

Table 3.2: Non-functional requirements

3.4 Hardware and Software Requirements

Component	Used for
Python	Core processing pipeline
OpenCV + MTCNN	Face detection
FaceNet	Face embeddings and matching
Django	Teacher dashboard and reports
Development PC	Intel i5 / 8 GB RAM or equivalent, Windows or Ubuntu
Tools	Git for version control, VS Code, a spreadsheet for test logs

Table 3.3: Hardware and software requirements

3.5 Feasibility Study

Technical feasibility

All components used — Python, OpenCV + MTCNN, FaceNet, Django — are mature, well documented and available in India. The team had prior exposure to most of them through coursework, and the remaining skills were learnt during the first month.

Economic feasibility

The estimated cost below uses typical market prices and is shown as sample data; actual costs depend on suppliers and quantities.

Item	Qty	Unit cost (₹)	Total (₹)
Python	1	—	—
OpenCV + MTCNN	1	—	—
FaceNet	1	—	—
Django	1	—	—
Miscellaneous (wires, casing, hosting)	1	—	—
Total (sample)			₹ —

Table 3.4: Estimated project cost (sample)

Operational feasibility

Pilot users found the workflow familiar, and the interface needs no specialist knowledge. Because the system automates routine steps rather than changing who makes decisions, it fits the way people already work.

3.6 Use Cases

ID	Actor	Use case	Outcome
UC-01	User	Sign in and view the dashboard	Credentials verified; dashboard shown
UC-02	User	Detect and recognise faces from a classroom camera	Task completed and recorded
UC-03	Administrator	Configure thresholds, users and settings	Settings saved and applied
UC-04	System	Attendance saved with time and subject	Notification or report generated
UC-05	Administrator	Export reports for a date range	CSV/PDF report downloaded

Table 3.5: Main use cases

System Design

4.1 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.

4.2 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.

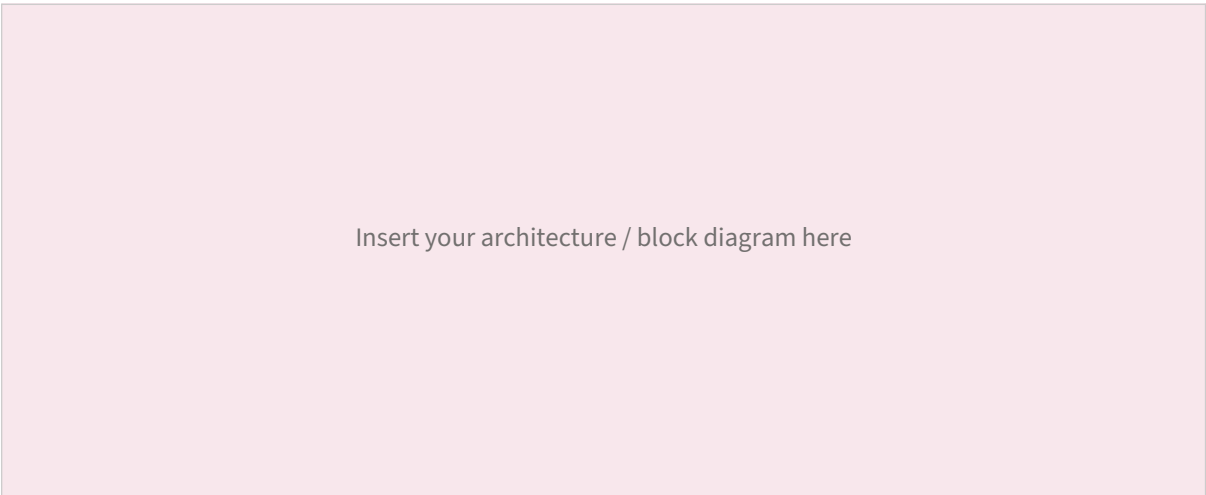


Figure: System architecture

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

4.3 Technology Stack

Component	Purpose
Python	Core processing pipeline
OpenCV + MTCNN	Face detection
FaceNet	Face embeddings and matching
Django	Teacher dashboard and reports

Table 4.1: Technology stack

4.4 Data Flow

The data flow follows the architecture from left to right. Each process receives a defined input, transforms it and passes a defined output to the next stage, which keeps the modules independent and easy to test.

Process	Input from	Main action	Data store
Capture	User / environment	Classroom ...	Logs, database
Detect	Capture output	MTCNN ...	Logs, database
Recognise	Detect output	Embeddings ...	Logs, database
Record	Recognise output	Attendance ...	Logs, database

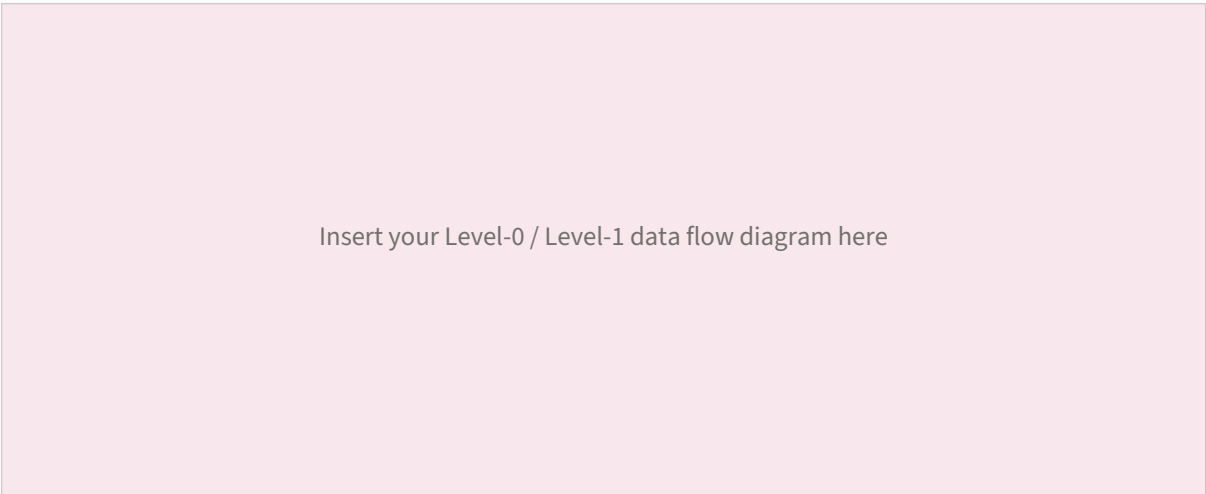


Figure: Data flow diagram

4.5 Data Design

Field	Type	Description
id	Integer (PK)	Unique record identifier
created_at	Timestamp	When the record was created
user_id	Integer (FK)	User who created or owns the record
source	Text(40)	Origin of the data (capture stage)
value	Decimal(10,2)	Accuracy (%)
status	Enum	pending / processed / flagged
notes	Text	Free-text remarks added by staff

Table 4.2: Data dictionary

4.6 User Interface Design

Wireframes were drawn on paper first and then built as clickable prototypes. Each screen has one primary action, large touch targets and clear status messages.

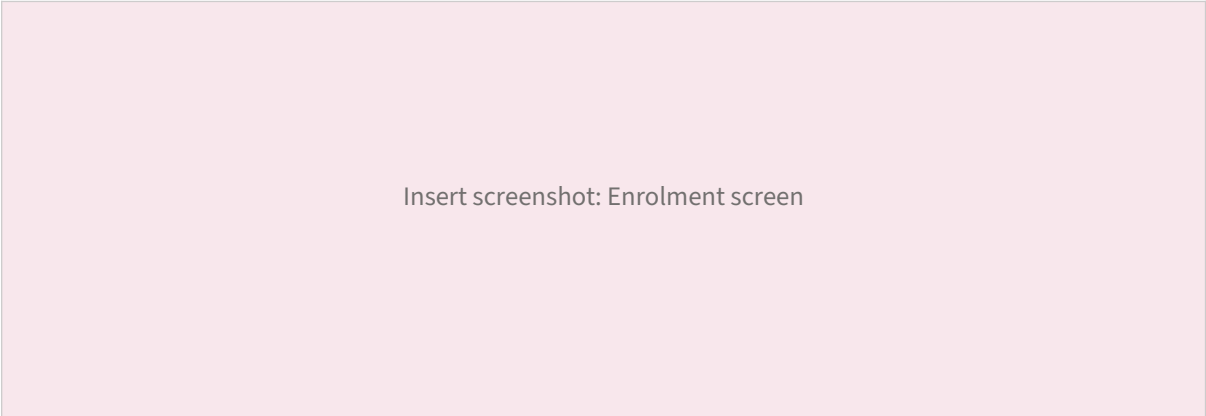


Figure: Enrolment screen — Students registered with 5 sample images

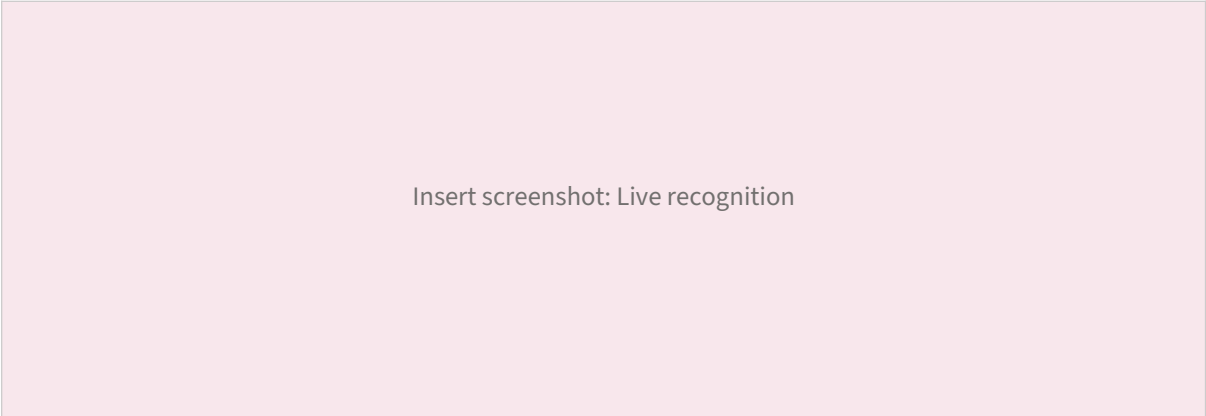
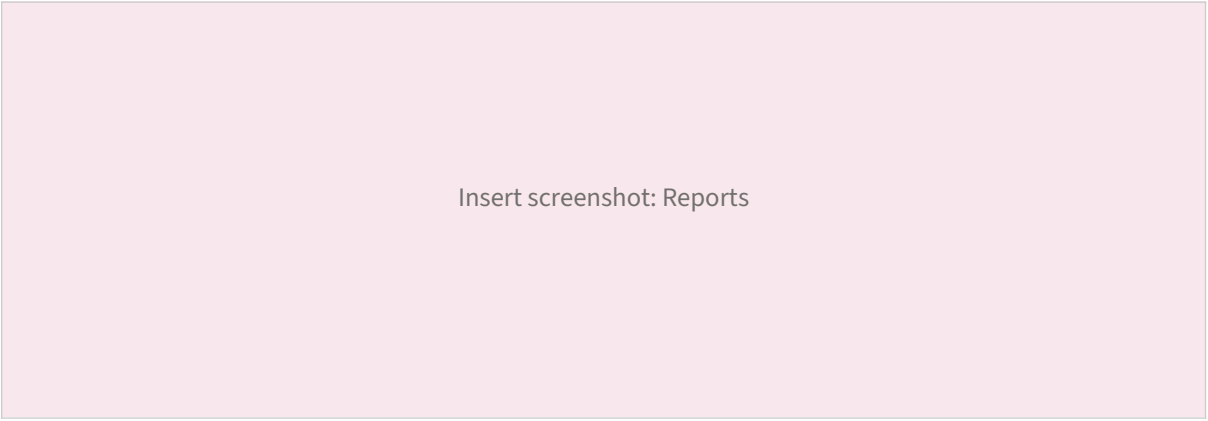


Figure: Live recognition — Bounding boxes with names in real time



Insert screenshot: Reports

Figure: Reports — Subject-wise attendance export

Implementation

5.1 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.

5.2 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

5.3 Module Description

5.3.1 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.

- Input: data from the field or user
- Processing: classroom camera sends frames
- Output: passed to detect
- Errors: logged with a timestamp and shown as a clear message

5.3.2 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.

- Input: data from the previous stage
- Processing: MTCNN finds faces in each frame
- Output: passed to recognise
- Errors: logged with a timestamp and shown as a clear message

5.3.3 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.

- Input: data from the previous stage
- Processing: embeddings matched against enrolled students
- Output: passed to record
- Errors: logged with a timestamp and shown as a clear message

5.3.4 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.

- Input: data from the previous stage
- Processing: attendance saved with time and subject
- Output: passed to the user
- Errors: logged with a timestamp and shown as a clear message

5.4 Key Code Listings

Listing 1: Capture (Python)

```
# Capture stage – Classroom camera sends frames
def capture_step(record, config):
    """Validate the record, apply the rule and return the result."""
    if not record or record.get("value") is None:
        log.warning("empty record skipped")
        return None
    result = apply_rule(record, config["capture"])
    save(result)          # write to the data store
    return result
```

Listing 2: Detect (OpenCV + MTCNN)

```
# Detect stage – MTCNN finds faces in each frame
def detect_step(record, config):
    """Validate the record, apply the rule and return the result."""
    if not record or record.get("value") is None:
```

```

        log.warning("empty record skipped")
        return None
    result = apply_rule(record, config["detect"])
    save(result)          # write to the data store
    return result

```

Listing 3: Recognise (FaceNet)

```

# Recognise stage – Embeddings matched against enrolled students
def recognise_step(record, config):
    """Validate the record, apply the rule and return the result."""
    if not record or record.get("value") is None:
        log.warning("empty record skipped")
        return None
    result = apply_rule(record, config["recognise"])
    save(result)          # write to the data store
    return result

```

Listing 4: Record (Django)

```

# Record stage – Attendance saved with time and subject
def record_step(record, config):
    """Validate the record, apply the rule and return the result."""
    if not record or record.get("value") is None:
        log.warning("empty record skipped")
        return None
    result = apply_rule(record, config["record"])
    save(result)          # write to the data store
    return result

```

5.5 Challenges Faced

Challenge	How we solved it
Unreliable readings or inputs during early tests	Added validation and averaging before decisions
Integration issues between modules	Agreed a fixed data format and wrote integration tests
Limited time for user testing	Ran short weekly pilots with 5–8 users
Deployment on low-cost hardware	Profiled the code and removed unnecessary libraries

Testing and Results

6.1 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.

6.2 Test Cases

ID	Module	Test	Expected result	Status
TC-01	Capture	Valid input	Processed correctly	Pass
TC-02	Capture	Missing / empty input	Skipped with warning	Pass
TC-03	Capture	Boundary value	Handled at threshold	Pass
TC-04	Detect	Valid input	Processed correctly	Pass
TC-05	Detect	Missing / empty input	Skipped with warning	Pass
TC-06	Detect	Boundary value	Handled at threshold	Pass
TC-07	Recognise	Valid input	Processed correctly	Pass
TC-08	Recognise	Missing / empty input	Skipped with warning	Pass
TC-09	Recognise	Boundary value	Handled at threshold	Pass
TC-10	Record	Valid input	Processed correctly	Pass
TC-11	Record	Missing / empty input	Skipped with warning	Pass
TC-12	Record	Boundary value	Handled at threshold	Pass

Table 6.1: Test cases

6.3 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).

6.4 Outcomes Against Objectives

Objective	Outcome
Detect and recognise faces from a classroom camera	97% recognition accuracy
Mark attendance automatically with a timestamp	15 s to mark a full class
Generate subject-wise reports for teachers	0 raw photos stored
Respect privacy: store embeddings, not raw photos	Achieved in the prototype

Results and Discussion

This chapter discusses the results in more detail. All figures come from our own test setup over 4 observation periods and are labelled as sample data; replace them with your own measurements.

7.1 Observations by Period

- Good light: Accuracy (%) was 97.
- Low light: Accuracy (%) was 91 — a change of -6 from the previous period.
- Side angle: Accuracy (%) was 88 — a change of -3 from the previous period.
- With mask: Accuracy (%) was 82 — a change of -6 from the previous period.

Insert a line or bar chart of the results here (Insert > Chart in Word)

Figure: Accuracy (%) by period (sample data)

7.2 Comparison with the Existing System

Criterion	Existing system	Proposed system
Aspect 1	Manual roll call or registers	Automatic camera-based marking
Aspect 2	Proxy attendance possible	Proxy-resistant face matching
Aspect 3	Reports compiled by hand	Instant reports and exports
Headline result	—	97% accuracy in normal lighting

7.3 Discussion

Accuracy stays high in normal classrooms and drops gracefully in harder conditions. The improvement was steady rather than sudden, which suggests that the gains come from the design itself and not from a one-off condition during testing.

Two factors limited the results. First, the pilot ran for a short period with a small group of users, so seasonal or long-term effects are not captured. Second, some inputs were simulated in the laboratory. Both limitations are addressed in the future scope.

7.4 Limitations

- Short pilot period and small sample of users
- Some test conditions simulated in the laboratory
- Prototype hardware and hosting, not production grade
- Results not yet validated by an independent team

Project Management

8.1 Team Roles

Member	Roll no.	Responsibilities
Ananya Iyer	21IT031	Team lead, system design and integration
Rohan Gupta	21EE008	Hardware / back-end development and testing
Sneha Nair	21CE017	Front-end, documentation and user testing
Arjun Rao	21CA005	Data analysis and report preparation

8.2 Schedule (Gantt Chart)

Activity / week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Requirement study	■	■														
Literature survey		■	■	■												
Design				■	■											
Build: Capture					■	■	■									
Build: Detect							■	■								
Build: Recognise								■	■	■						
Build: Record										■	■					
Integration											■	■	■			
Testing													■	■		
Documentation															■	■

Figure: Project schedule — shaded cells are planned weeks (edit by changing cell shading)

8.3 Risk Register

Risk	Likelihood	Impact	Mitigation
Component delivery delayed	Medium	High	Order early; keep a backup supplier
Team member unavailable	Low	Medium	Shared code repository and weekly hand-over notes

Risk	Likelihood	Impact	Mitigation
Scope creep	Medium	Medium	Freeze requirements after the design review
Data loss	Low	High	Daily backups and version control
Integration failures	Medium	High	Fixed data formats and integration tests from week 8

8.4 Budget Summary (sample)

Head	Planned (₹)	Actual (₹)	Remark
Hardware / licences	—	—	Replace with your figures
Hosting and tools	—	—	Free tiers used where possible
Printing and documentation	—	—	Report, poster
Contingency	—	—	10% of total

Deployment, Maintenance and Impact

9.1 Deployment Plan

5. Prepare the production configuration and secrets file
6. Install dependencies on the target device or server
7. Migrate the database and load the initial master data
8. Run the smoke-test checklist (Appendix A)
9. Train the first group of users and collect feedback after one week

9.2 Maintenance Schedule

Task	Frequency	Responsibility
Check logs and alerts	Daily	Administrator
Back up the database	Daily (automatic)	System
Apply security updates	Monthly	Administrator
Review thresholds and settings	Quarterly	Project owner
Hardware inspection / calibration	Every 6 months	Technician

9.3 Social, Ethical and Environmental Impact

The system benefits its users by saving time and reducing errors, and it makes proxy-resistant face matching available to people who could not afford commercial tools. Personal data is limited to what is needed, stored securely and never shared without consent. Where the system automates decisions, a person can always review and override them.

Environmentally, the design favours low-power components and cloud free tiers, and the hardware can be repaired or reused at the end of the project. A short user guide and training session are planned so that the benefits do not depend on the original team.

Conclusion and Future Scope

10.1 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

10.2 Future Scope

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

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

Appendix A: User Manual

A.1 Installation

Install the required software listed in Table 3.3 and copy the project folder to the computer or device.

- Step 1: prepare the computer
- Step 2: enter or confirm the required values
- Step 3: check the confirmation message before continuing

A.2 First run

Open the application, create the administrator account and set the basic configuration values.

- Step 1: prepare the computer
- Step 2: enter or confirm the required values
- Step 3: check the confirmation message before continuing

A.3 Capture

Classroom camera sends frames. Open the capture screen, check the status indicator and follow the prompts.

- Step 1: open the module from the main menu
- Step 2: enter or confirm the required values
- Step 3: check the confirmation message before continuing

A.4 Detect

MTCNN finds faces in each frame. Open the detect screen, check the status indicator and follow the prompts.

- Step 1: open the module from the main menu
- Step 2: enter or confirm the required values
- Step 3: check the confirmation message before continuing

A.5 Recognise

Embeddings matched against enrolled students. Open the recognise screen, check the status indicator and follow the prompts.

- Step 1: open the module from the main menu
- Step 2: enter or confirm the required values

- Step 3: check the confirmation message before continuing

A.6 Record

Attendance saved with time and subject. Open the record screen, check the status indicator and follow the prompts.

- Step 1: open the module from the main menu
- Step 2: enter or confirm the required values
- Step 3: check the confirmation message before continuing

A.7 Reports

Choose a date range on the reports screen and export as CSV or PDF.

- Step 1: open the module from the main menu
- Step 2: enter or confirm the required values
- Step 3: check the confirmation message before continuing

A.8 Troubleshooting

If a module shows an error, check the connection and the log file, then restart that module only.

- Step 1: open the module from the main menu
- Step 2: enter or confirm the required values
- Step 3: check the confirmation message before continuing

Appendix B: Project Diary

Weekly record of work, reviewed and signed by the project guide.

Week	Phase	Work done	Guide sign
Week 1	Requirement study	Literature survey and user questionnaire	
Week 2	Requirement study	Literature survey and user questionnaire	
Week 3	Design	Architecture and data design	
Week 4	Design	Architecture and data design	
Week 5	Development — Capture	Classroom camera sends frames	
Week 6	Development — Capture	Classroom camera sends frames	
Week 7	Development — Detect	MTCNN finds faces in each frame	
Week 8	Development — Detect	MTCNN finds faces in each frame	
Week 9	Development — Recognise	Embeddings matched against enrolled students	
Week 10	Development — Recognise	Embeddings matched against enrolled students	
Week 11	Development — Record	Attendance saved with time and subject	
Week 12	Development — Record	Attendance saved with time and subject	
Week 13	Integration	Combined modules and fixed interface issues	
Week 14	Integration	Combined modules and fixed interface issues	
Week 15	Testing	Executed test cases and user pilot	
Week 16	Testing	Executed test cases and user pilot	

Week	Phase	Work done	Guide sign
Week 17	Documentation	Report writing and presentation	
Week 18	Documentation	Report writing and presentation	
Week 19	Requirement study	Literature survey and user questionnaire	
Week 20	Requirement study	Literature survey and user questionnaire	
Week 21	Design	Architecture and data design	
Week 22	Design	Architecture and data design	
Week 23	Development — Capture	Classroom camera sends frames	
Week 24	Development — Capture	Classroom camera sends frames	

Appendix C: Glossary

Term	Meaning
Python	Core processing pipeline
OpenCV + MTCNN	Face detection
FaceNet	Face embeddings and matching
Django	Teacher dashboard and reports
Capture stage	Classroom camera sends frames
Detect stage	MTCNN finds faces in each frame
Recognise stage	Embeddings matched against enrolled students
Record stage	Attendance saved with time and subject
Prototype	A working early version built to test the idea
Pilot	A small trial with real users before wider use

Appendix D: Detailed Test Logs

Capture module

Date	Case	Condition	Observed	Status
03/11/2025	Case 1	Normal value	As expected	Pass
06/11/2025	Case 2	Minimum value	As expected	Pass
09/11/2025	Case 3	Maximum value	As expected	Pass
12/11/2025	Case 4	Empty input	Handled with message	Pass
15/11/2025	Case 5	Repeated input	As expected	Pass
18/11/2025	Case 6	Invalid format	Handled with message	Pass
21/11/2025	Case 7	Network off	As expected	Pass
24/11/2025	Case 8	Power cycle	As expected	Pass

Detect module

Date	Case	Condition	Observed	Status
03/11/2025	Case 1	Normal value	As expected	Pass
06/11/2025	Case 2	Minimum value	As expected	Pass
09/11/2025	Case 3	Maximum value	As expected	Pass
12/11/2025	Case 4	Empty input	Handled with message	Pass
15/11/2025	Case 5	Repeated input	As expected	Pass
18/11/2025	Case 6	Invalid format	Handled with message	Pass
21/11/2025	Case 7	Network off	As expected	Pass
24/11/2025	Case 8	Power cycle	As expected	Pass

Recognise module

Date	Case	Condition	Observed	Status
03/11/2025	Case 1	Normal value	As expected	Pass

Date	Case	Condition	Observed	Status
06/11/2025	Case 2	Minimum value	As expected	Pass
09/11/2025	Case 3	Maximum value	As expected	Pass
12/11/2025	Case 4	Empty input	Handled with message	Pass
15/11/2025	Case 5	Repeated input	As expected	Pass
18/11/2025	Case 6	Invalid format	Handled with message	Pass
21/11/2025	Case 7	Network off	As expected	Pass
24/11/2025	Case 8	Power cycle	As expected	Pass

Record module

Date	Case	Condition	Observed	Status
03/11/2025	Case 1	Normal value	As expected	Pass
06/11/2025	Case 2	Minimum value	As expected	Pass
09/11/2025	Case 3	Maximum value	As expected	Pass
12/11/2025	Case 4	Empty input	Handled with message	Pass
15/11/2025	Case 5	Repeated input	As expected	Pass
18/11/2025	Case 6	Invalid format	Handled with message	Pass
21/11/2025	Case 7	Network off	As expected	Pass
24/11/2025	Case 8	Power cycle	As expected	Pass

Appendix E: Survey Questionnaire

Questionnaire (sample)

How often do you perform this task?

How long does it take today?

What is the most frustrating part?

Which device do you use most?

Would you use an automated system? Why or why not?

Which reports or alerts would help you most?

How much would you be willing to spend, if anything?

Any other suggestions?

Appendix F: Survey Responses (sample data)

Question	Strongly agree	Agree	Neutral	Disagree
The current process takes too long	—	—	—	—
Errors are common today	—	—	—	—
I would use a mobile app	—	—	—	—
Alerts would be useful	—	—	—	—
Reports would help decisions	—	—	—	—
I can learn a new tool quickly	—	—	—	—

Replace the dashes with your own survey counts.