1) Introduction to the Proposed System

1. Problem Definition:

Educational institutions, particularly small coaching classes, and colleges face challenges in maintaining accurate and cheat-proof attendance records. Traditional methods like manual roll calls are easily manipulated, allowing students to mark attendance for others or skip classes undetected. This results in unreliable attendance data and hinders effective monitoring by administrators.

The "NoProxy" project addresses these issues using computer vision and facial recognition technology to create an automated, cheat-proof attendance system. This system ensures accurate recording of student presence, preventing re-marking attendance within 24 hours and enhancing overall accountability and class participation.

2. System Overview:

The "NoProxy" system consists of two main components: a Small Desktop Software and a Web Interface. The software records student attendance, while the web interface provides administrative functionalities and allows students to view their attendance records.

3. Project Functionalities with Module Specification:

1. Attendance Recording

Module: Small Desktop Software

Functionality: Captures student images via webcam and marks attendance using facial recognition.

Libraries: numpy, cv2, face_recognition, cvzone, os, pickle, firebase_admin, datetime

2. Student Management

Module: Web Interface (Admin)

Functionality: Admin can add/delete students, with data stored in Firebase.

Image Processing: Encoder.py for encoding, imgDim.py for resizing

3. Admin Authentication

Module: Web Interface (Admin)

Functionality: Admin login/logout

Libraries: Flask, MySQL

4. Attendance Viewing

Module: Web Interface (Student)

Functionality: Students can view their attendance.

Libraries: flask, json

5. Attendance Restriction

Module: Small Desktop Software

Functionality: Prevents re-marking attendance within 24 hours.

Libraries: datetime

---

4. Operating Environment (H/W & S/W Requirement Specification)

Hardware Requirements:

- Webcam: Possibly HD to Capture students' faces.
- Desktop/Laptop:
- Memory: Minimum 4 GB RAM (8 GB Recommend)
- Storage: 256 GB SSD Recommended
- Graphics: GPU Optional for faster image processing
- Monitor: HD Resolution (1920x1080)
- Internet: High-speed Internet Connection for real-time data access & storage on Firebase

Software Requirements:

- Operating System: Windows 10/11/MacOS/Linux
- Programming Language: Python 3.x
- Libraries/Modules: numpy, cv2, face_recognition, cvzone, os, pickle, firebase_admin, datetime
- Web Interface: Flask, Python 3.x
- Database: Firebase Realtime Database, MySQL
- Browser: Google Chrome, Firefox, Safari, or any Chromium-based.

2) Overview of the Proposed System

1. Proposed System

The "NoProxy" project offers a cheat-proof attendance system using computer vision and facial recognition technologies. It ensures accurate and reliable attendance tracking for small coaching classes and colleges.

2. Objectives of the System

- Easy Accurate Attendance
- Enhance Security
- Simplify Attendance Management
- Promote Transparency
- Prevent Manipulation
- Improve Accountability & Scalability

3. Feasibility Study

Technical Feasibility:

- Uses well-supported technologies (OpenCV, Face Recognition, Flask)
- Integrates with Firebase and MySQL easily
- Requires accessible and affordable hardware (computers, webcams)

Economical Feasibility:

- Minimal hardware investment
- Open-source software components reduce costs
- Cost-effective cloud storage and database solutions (Firebase)

Operational Feasibility:

- User-friendly interface for admins and students
- Quick, non-intrusive facial recognition process
- Seamless integration into existing workflows

4. User Requirement Specification

Functional Requirements:

Admin features:

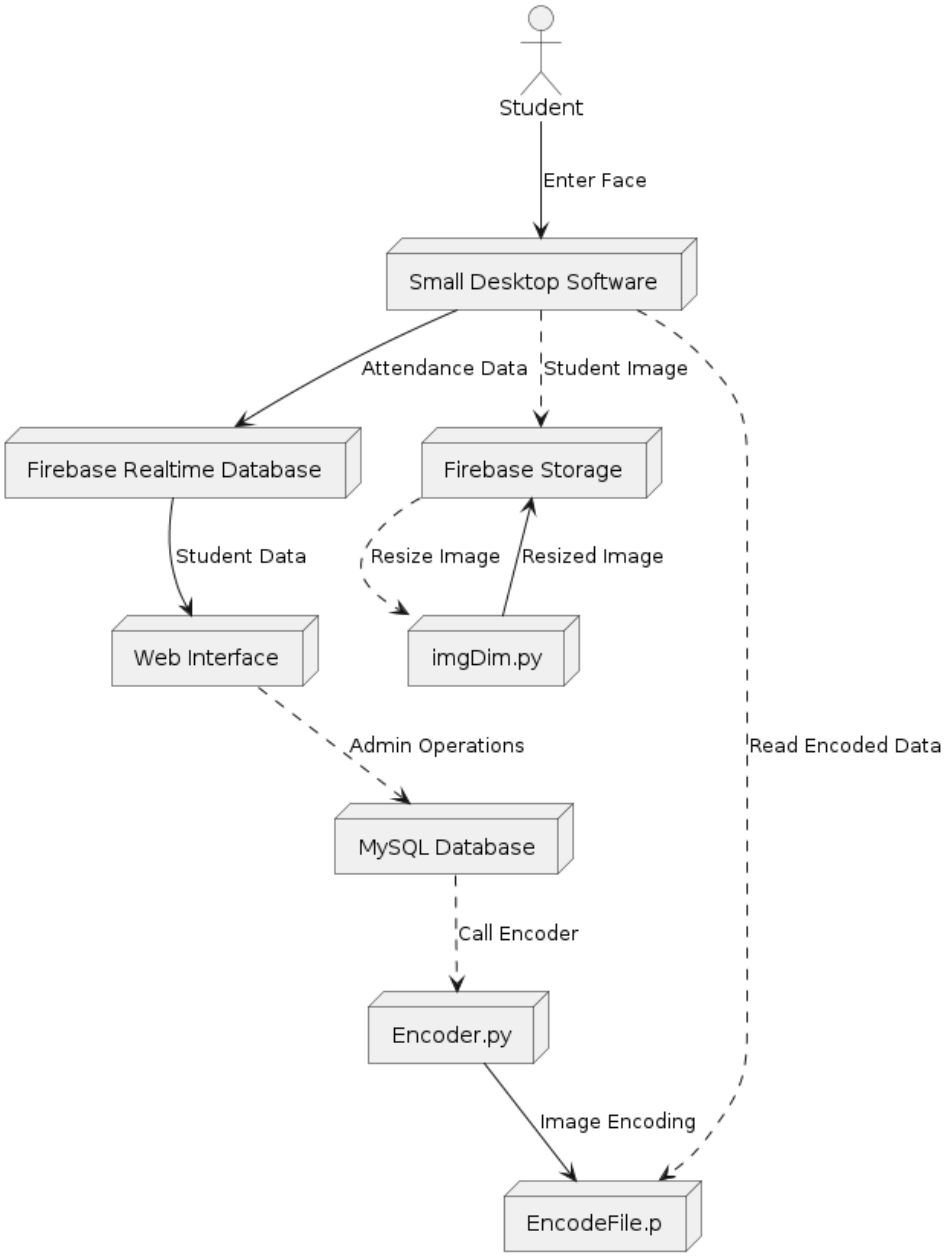
- Add, Delete Student Records
- Secure login, and logout functionality with efficient exception handling

Student features:

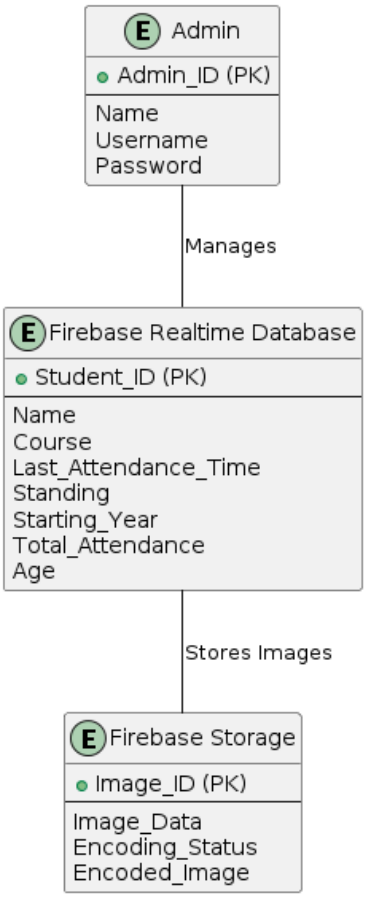
- Mark Attendance
- Access to view attendance records online

3) System Analysis & Design

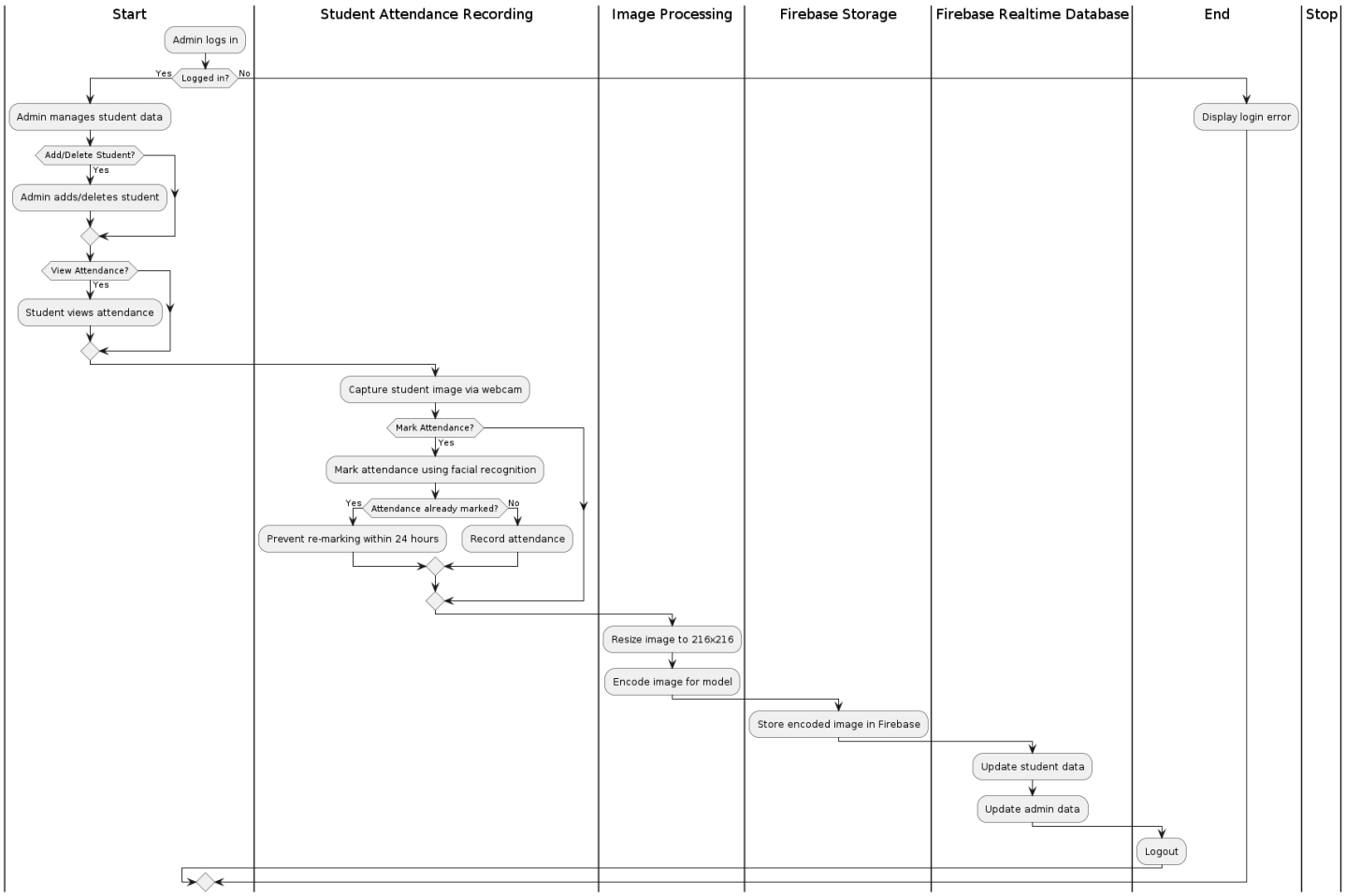
1. Data Flow Diagram



2. ER Diagram



3. Activity Diagram



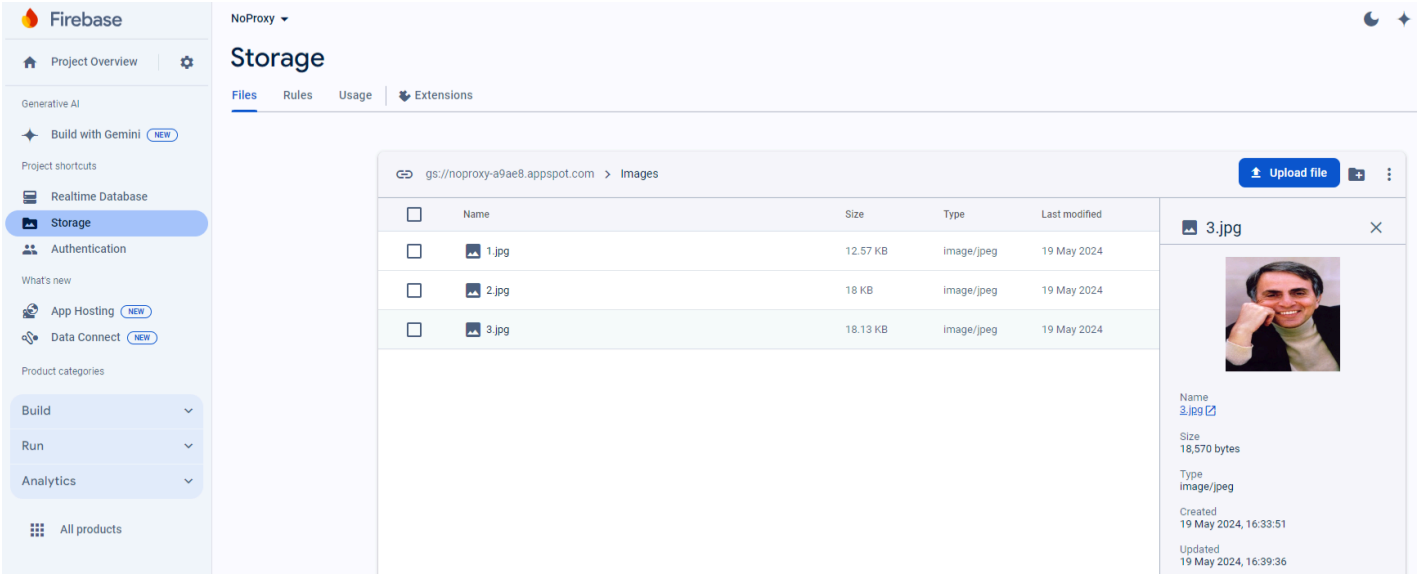
4. Data Dictionary with Table Specification

Admin Table

	id	username	password
Edit Copy Delete	3	nisoojadhav	$2b$12$o6.98v24FuaoZU5Qrt5UhePcxTQaeT8ueAjen.q/kyc...

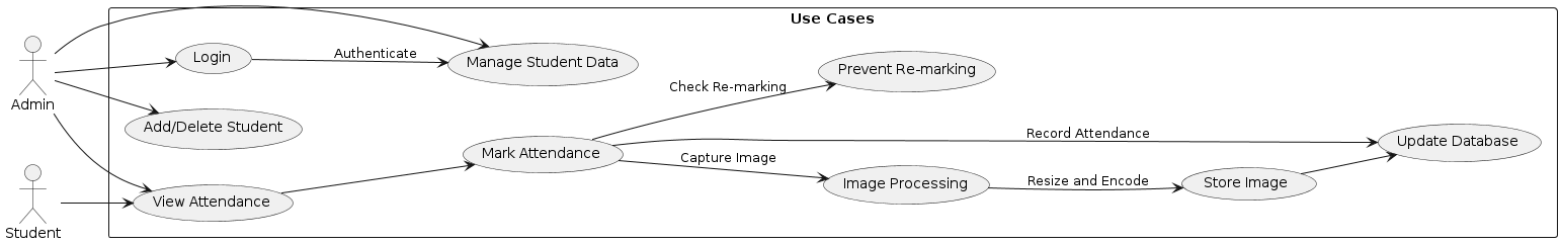
Firebase Real Time Database

Firebase Storage



Student	id	Unique Identifier
	course	Standard
	last_attendance_time	Last marked time
	name	Student name
	standing	Behaviour's Grade
	starting_year	Starting Year at tuition/college
	total_attendance	Total Attendance
	year	Age / Years old

5. Use Case Diagram



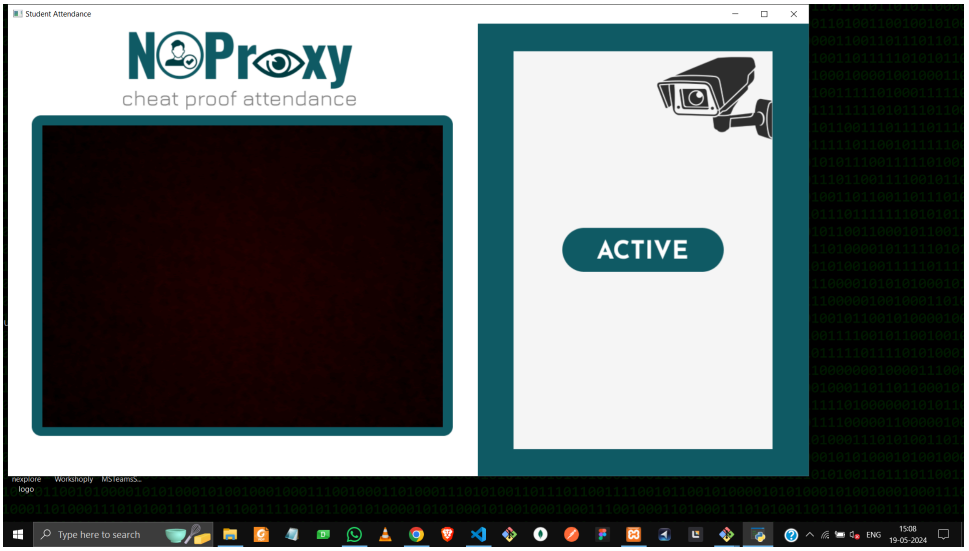
4) User Manual

1. Operational Instructions

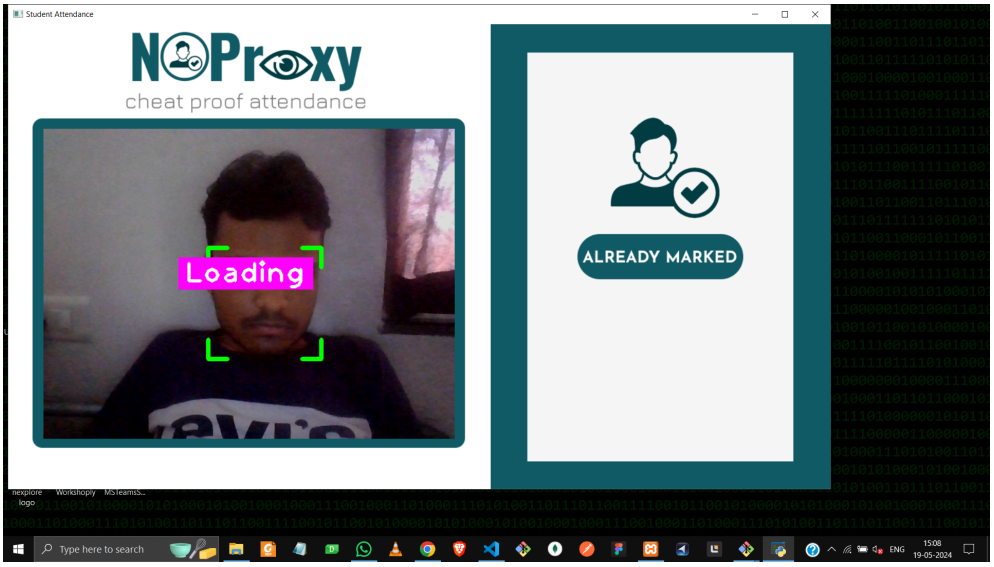
- Login: Using Credentials to access the system.
- Manage Student Data: Add/Update Delete records.
- View Attendance: Check attendance records.
- Mark Attendance: Students mark attendance with facial recognition.
- Image Processing: The system resizes & encodes images.
- Store Images: Encoded images stored in Firebase Storage.
- Update Database: The database is updated with attendance & student data.
- Prevent Re-marking: Attendance can only be marked once daily.
- Logout: Securely ending the session.

2. Input/Output Screens

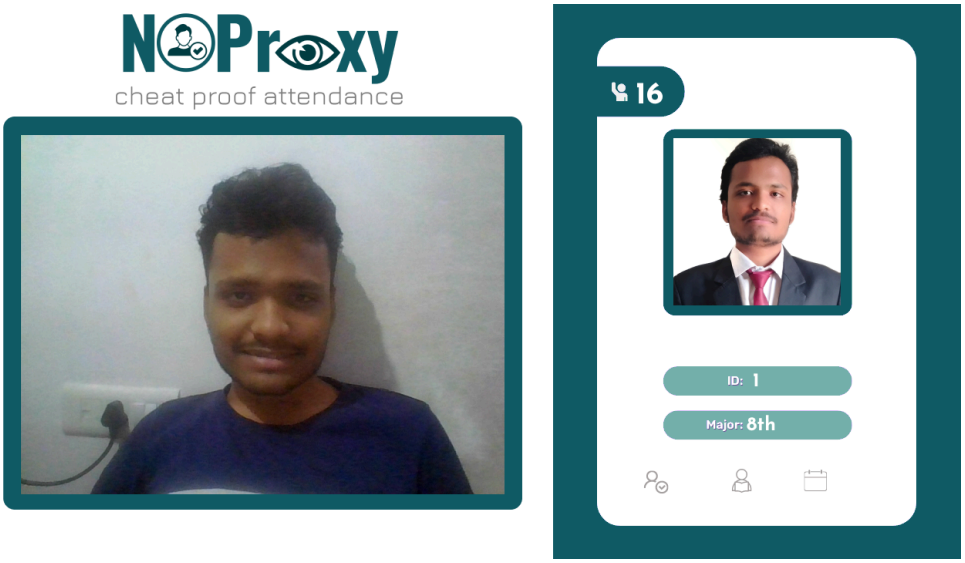
Active



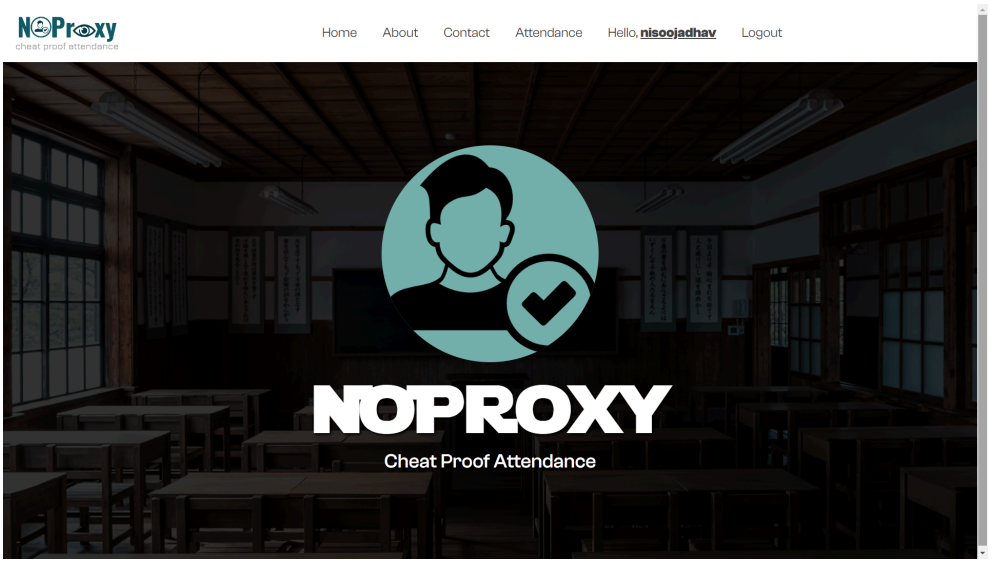
Marked



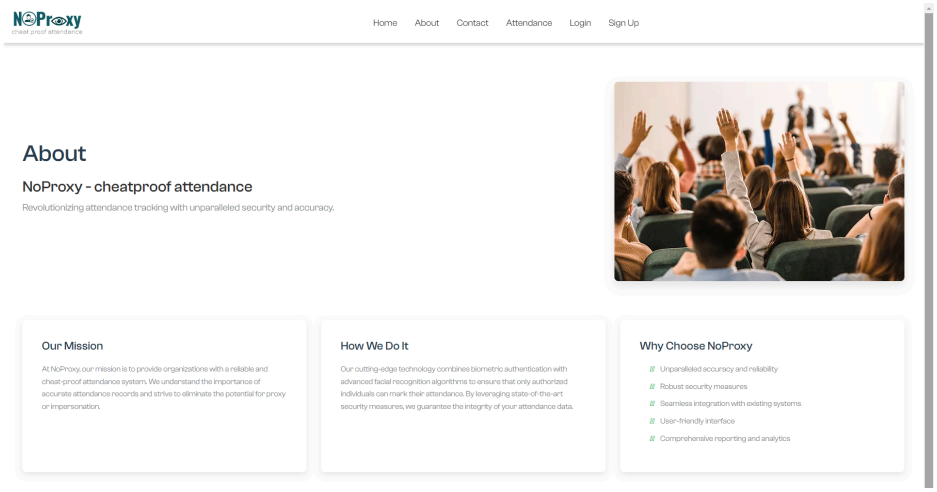
Stats



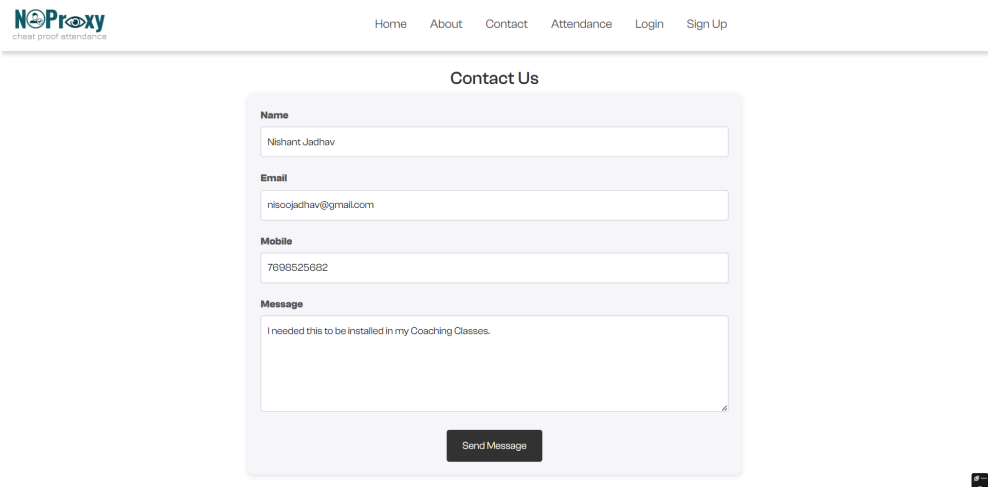
Home



About

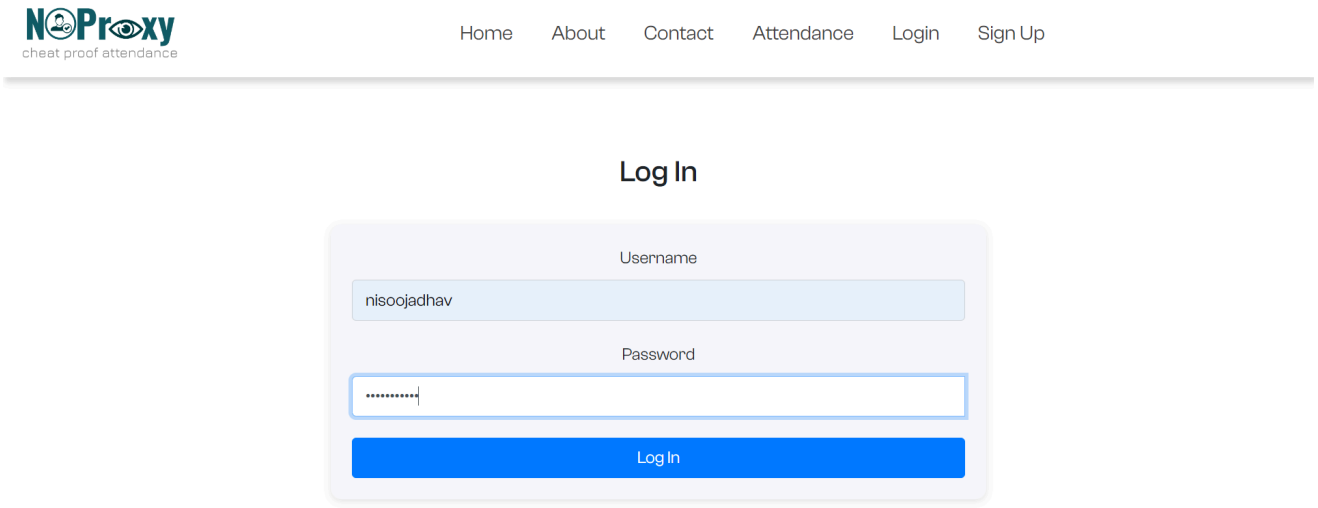


Contact

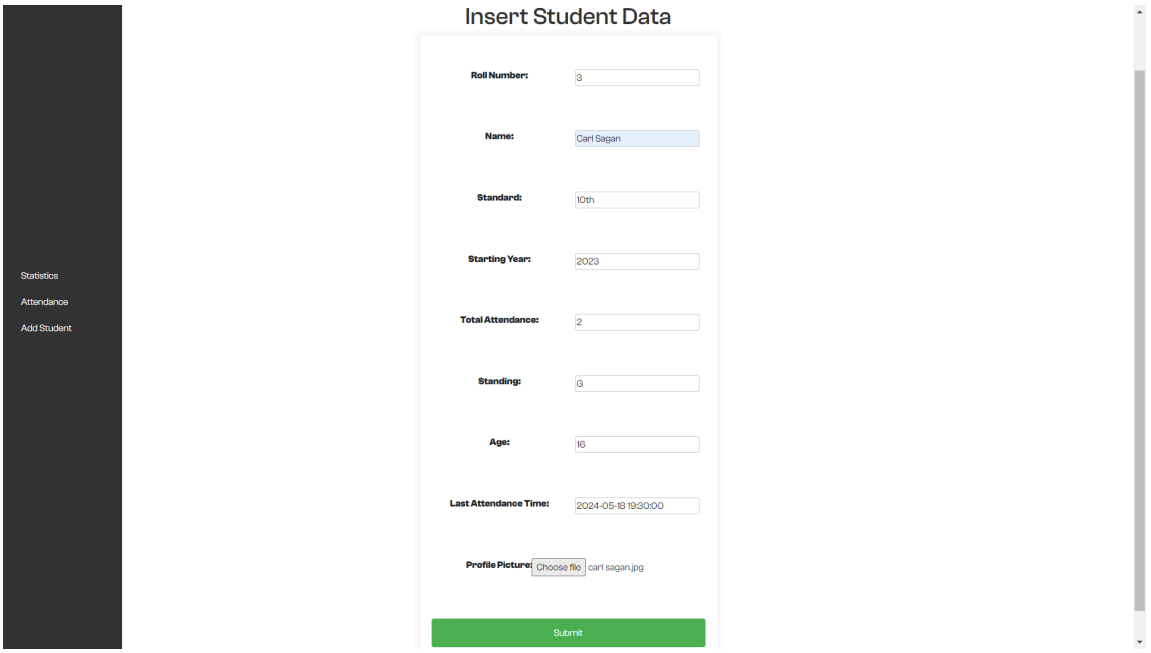



```
{
    result: "success",
    data: "{"name":["Nishant Jadhav"],"mobile":["7698525682"],"message":["I needed this to be installed in my Coaching Classes."],"email":["nisoojadhav@gmail.com"]}"
}
```

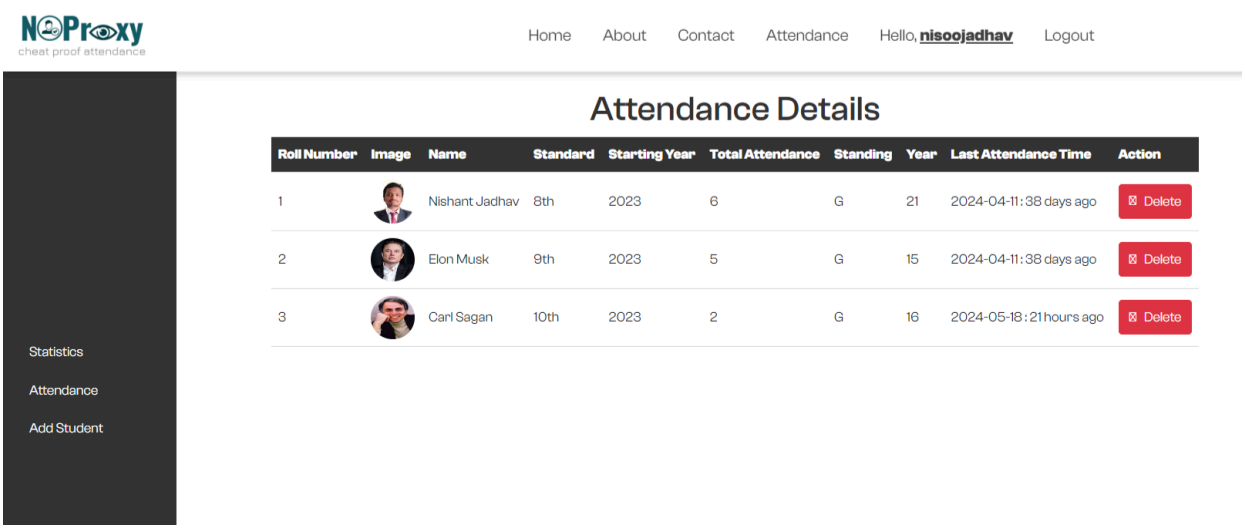
Admin Login



Insert Student Data



Attendance Details



3. **Reports**

The "NoProxy" system provides detailed reports on student attendance, including summaries, trends over time, class-wise breakdowns, and individual student histories. These reports enable administrators to track attendance patterns, identify students at risk of falling behind, and make informed decisions to improve overall attendance and student engagement. The system's reporting capabilities offer valuable insights that help enhance student performance and ensure accountability across educational institutions.

5) System Limitation

1. **Selective Attendance Marking:** The system marks attendance even if a student attends only one lecture and skips the rest of the day, which could lead to inaccuracies.

2. **Restriction on Re-marking:** Only allows re-marking of attendance after 24 hours, which may limit the flexibility of correcting attendance errors promptly.

3. **Mandatory Internet Connection:** Requires stable Internet connection for real-time data access & storage, limitation in area with poor connectivity.

4. **Dependence on Hardware:** It relies on hardware components like webcams, which can be subject to malfunctions or compatibility issues.

5. **Other limitations:**

    - **Cost**
    - **Scalability**
    - **User Training**
    - **Privacy Concerns**
    - **Integration Complexity**
    - **Facial Recognition Accuracy problem.**

6) Future Enhancement & Conclusion

- Mobile Application
- Enhanced Reporting & Statistics
- Integration with Biometrics
- Use of Machine Learning for Attendance Prediction
- Enhanced Data Analysis
- Automated Reporting

Conclusion:

The "NoProxy" attendance system provides an effective solution for tracking student attendance, despite limitations such as hardware dependencies and facial recognition accuracy. Future enhancements could include improved facial recognition, a mobile application, and automated reporting, which would further enhance its functionality and usability, making it a promising tool for improving attendance monitoring and student engagement in educational settings.

7) Bibliography & Glossary

● Definitions:

- **Admin**: Administrator, a user with privileged access to manage the system.
- **Firebase**: Google's mobile platform that helps you quickly develop high-quality apps.
- **Facial Recognition**: Technology used to identify or verify a person from a digital image or a video frame.
- **MySQL**: An open-source relational database management system.
- **Firebase Realtime Database**: A cloud-hosted NoSQL database that stores data in JSON format, which syncs in real time.
- **Firebase Storage**: A cloud service for storing user-generated content, such as photos and videos.
- **UML**: Unified Modeling Language, a standardized modeling language for software and systems development.
- **ER Diagram** Entity-Relationship Diagram, a diagram that shows the relationships between entity sets stored in a database.
- **Use Case Diagram**: A diagram that shows the interactions between actors and a system to achieve a specific goal.

- **Acronyms & Abbreviations:**

- **CV:** Computer Vision
- **FL:** Firebase Realtime Database
- **FS:** Firebase Storage/File System
- **ML:** Machine Learning
- **API:** Application Programming Interface
- **DB:** Database
- **UI:** User Interface
- **UX:** User Experience
- **CRUD:** Create, Read, Update, Delete
- **HTTP:** Hypertext Transfer Protocol
- **JSON:** JavaScript Object Notation
- **SDK:** Software Development Kit
- **URL:** Uniform Resource Locator
- **WSGI:** Web Server Gateway Interface

References:

Face_Recognition: https://github.com/ageitgey/face_recognition

Flask Documentation: https://flask.palletsprojects.com/en/3.0.x/

Firebase Documentation: https://firebase.google.com/docs

OpenCV Documentation: https://docs.opencv.org/4.x/

MySQL Documentation: https://dev.mysql.com/doc/

Python Packages: https://pypi.org/

Debugging: https://stackoverflow.com/ & https://claude.ai

The Complete Code can be found at my GitHub Repo:

https://github.com/nisoojadhav/NoProxy