Report

Title of the Project: Smart Attendance System (Face Recognition QR code)

Students Name:

Raghav Kumar Jha (33671024009)

Signature:

Yojana Adhikari (33671024027)

Signature:

Sonam Sharma (33671024029)

Signature:

Shruti Kumari Singh (33671024030)

Signature:

Supervisor Name: Dr Tumpa Banerjee

Signature:

Report:

Objective

The **Smart Attendance System** aims to **automate student attendance** using **facial recognition technology**.
Instead of calling names or marking manually, the system detects and recognizes faces in real time through a webcam, logs their attendance into a **MySQL database**, and provides a **Tkinter-based GUI** to manage and view records.

Overview

This Smart Attendance System is a GUI-based application designed to automate the process of recording and managing student attendance using face recognition. It leverages tkinter for the user interface, cv2 and face_recognition for image processing, and interacts with a MySQL database for managing records.

Core Modules and Workflow

1. User Authentication

- The application starts with an admin login window that requires a username and password.
- Credentials are verified against the admin table in the MySQL database.
- On successful verification, the admin is redirected to the main module; otherwise, an error message is displayed.

2. Attendance Management

- The primary user interface provides buttons for starting/stopping attendance, managing students, viewing and exporting attendance data.
- When “Start Attendance” is pressed, the system activates the webcam and collaborates with the face recognition library to identify students.
- Detected faces are compared against known encodings loaded from student images in the database. If a match is found, attendance is marked in the database with timestamp details.

3. Database Interaction

- Student details and attendance records are managed in MySQL.
- Attendance marking involves checking for existing records for the day; updates or new entries are made depending on whether the student has already checked in.

4. Student Management

- Allows for addition, update, and deletion of student records.
- Images used for face recognition are saved to a dedicated folder and their paths stored in the database.
- The interface also displays all students, enabling direct management through the GUI.

5. Attendance Export

- Records can be exported for the current day or any custom date range.
- The export creates spreadsheets using openpyxl, facilitating easy reporting and backup.
- Both simple (today's attendance) and advanced (date range) export functionalities are provided.

Face Recognition Concept Summary

The face_recognition library performs three main tasks:

1. **Encoding:** Converts faces into a 128-dimensional numeric vector.
2. **Comparison:** Compares the current frame's encodings with known encodings.
3. **Matching:** If distance < 0.6 → it's a match; otherwise, unknown.

Each student image is encoded once and stored temporarily when the app starts.

Technical Features

- Utilizes OpenCV (cv2) and the face_recognition library for real-time video streaming and face matching.
- Ensures data security by requiring admin authentication.
- Designed for scalability: new students can be added, and attendance reporting can be customized.
- Functionalities for managing accidental duplicate entries by updating time stamps if a student is marked present again the same day.
- Uses Tkinter for all GUI components, making it cross-platform compatible.

Summary

Feature	Description
Goal	Automate student attendance using face recognition
Frontend	Tkinter-based GUI
Backend	Python + OpenCV + face_recognition
Database	MySQL
Key Tables	students, attendance, admin
Recognition Logic	Records first & last attendance times
Output	Attendance logged in DB and exported CSV

Future Enhancements

- Integrate **SMS/email notification** for absentees.
- Enable **Cloud database** (for multi-device syncing).
- Add **Face dataset training** with multiple images per student.
- Create a **Django web dashboard** for analytics.