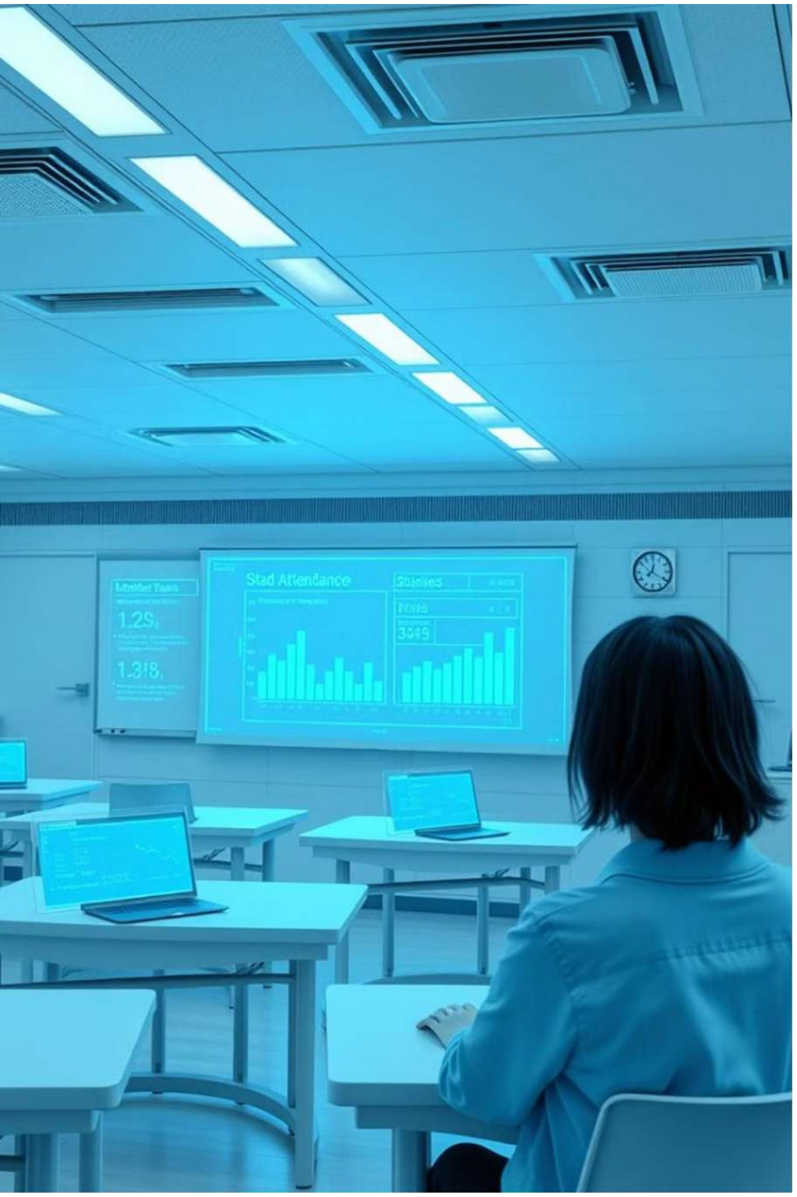


SmartTrack

SmartTrack is a smart, automated attendance management system that leverages machine learning-based facial recognition to ensure accuracy, efficiency, and minimal manual intervention.

Built using Python and Flask, the system uses an SQLite database to maintain user records, embeddings, and attendance logs. The architecture separates routing, templates, and machine learning logic for maintainability and scalability.

CSE IOTCS

Project Overview: SmartTrack at a Glance

Core Modules:

• Admin Panel: Manage all institutional data (students, teachers, subjects, classes) via forms or CSV uploads.

• Student Panel: View personal records, attendance data, and upload a video for facial registration.

• Teacher Panel: Monitor attendance logs, filter by class/subject, edit previous records, and take attendance through face recognition or CSV uploads.

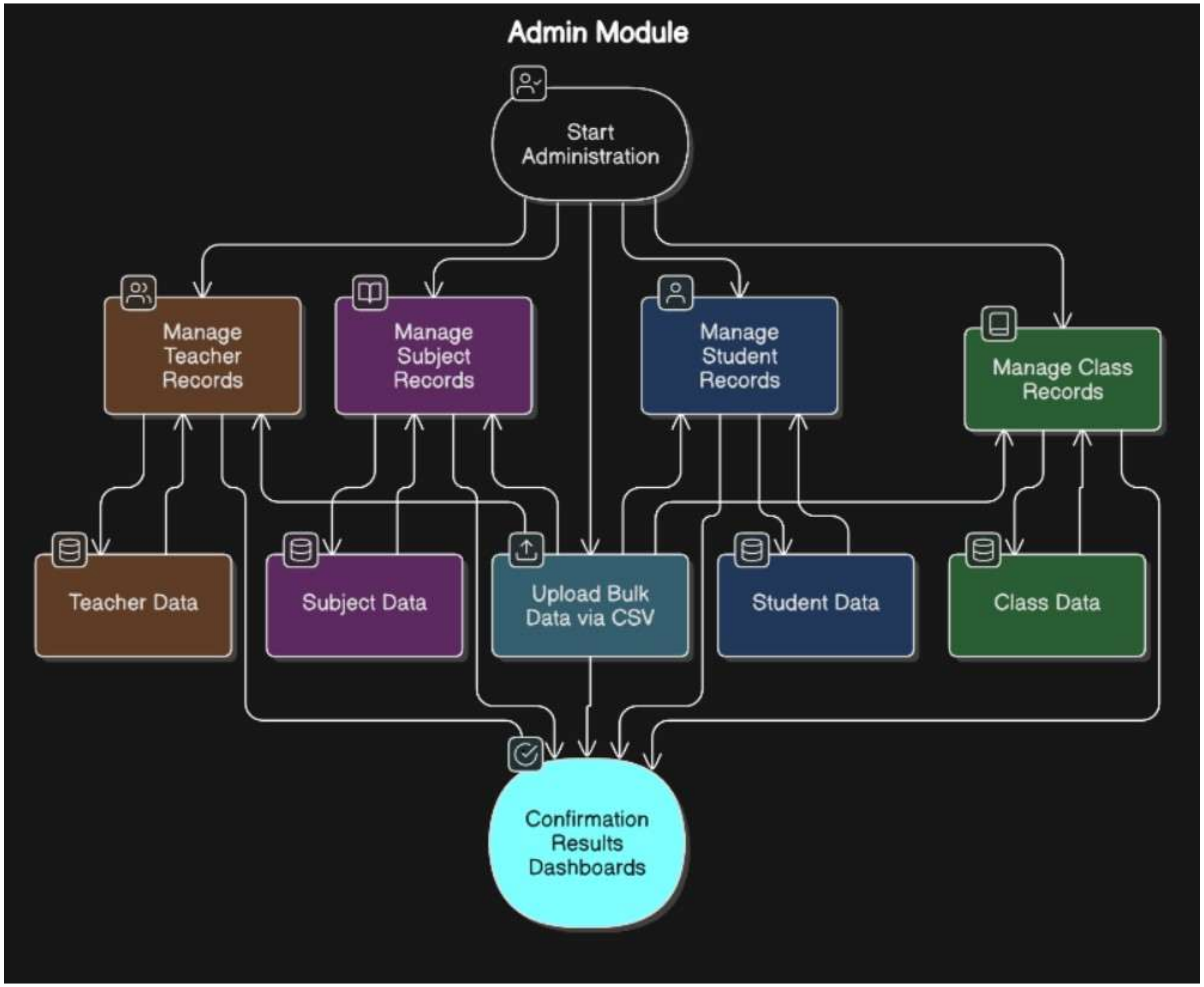
• ML Engine: Handles face registration and recognition using DeepFace with MTCNN and cosine similarity, storing face embeddings in a database for future matching. .

Why We Made the Project

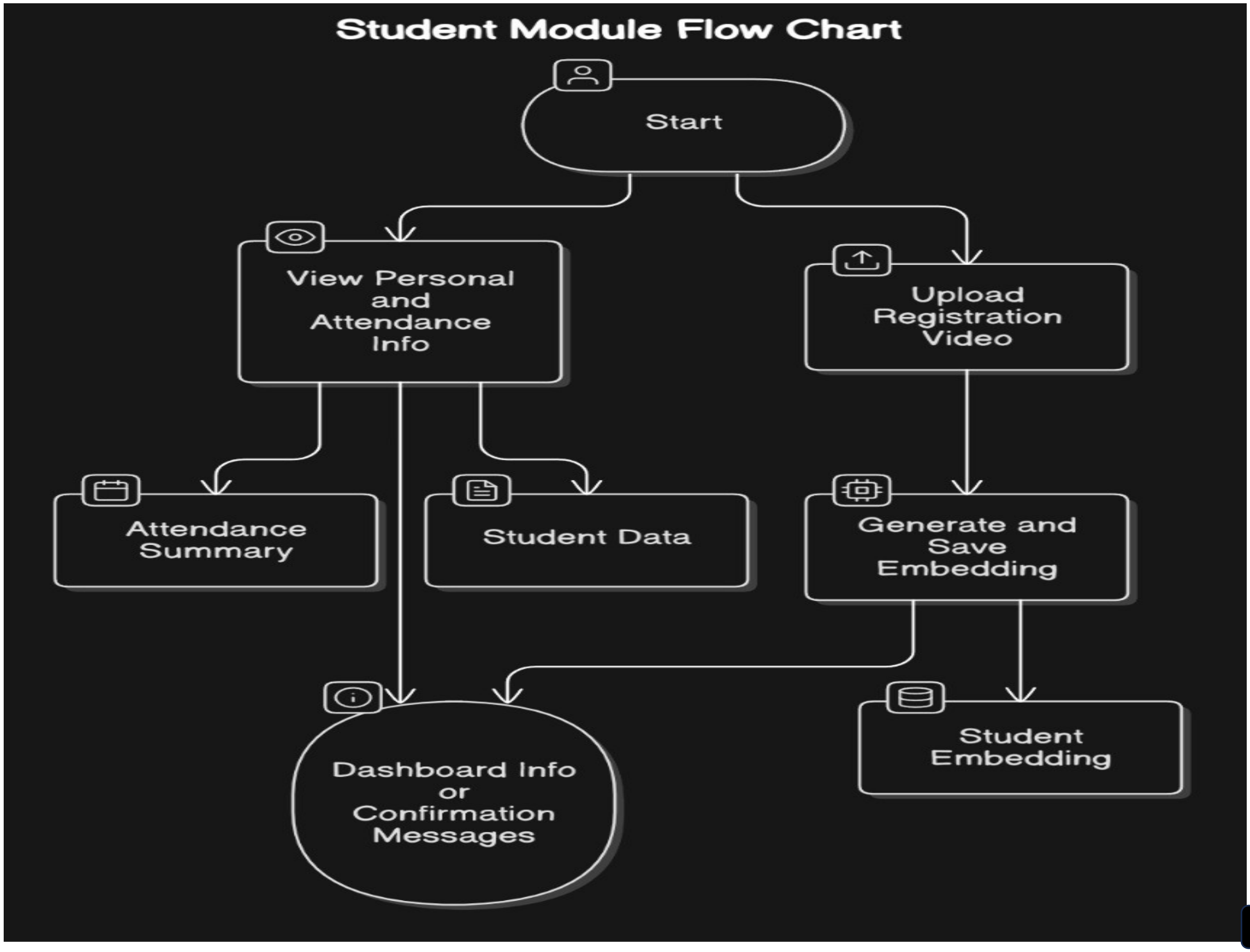
The core motivation behind SmartTrack was to innovate a traditional, error-prone process — student attendance. Manual methods are time-consuming, inconsistent, and prone to manipulation.
Our aim was:
• To create a contactless, efficient, and secure attendance system.
• To integrate facial recognition in a real-world academic application.
• To allow teachers and admins to operate attendance systems without complex training.
• To use machine learning practically and build a scalable web-based solution.

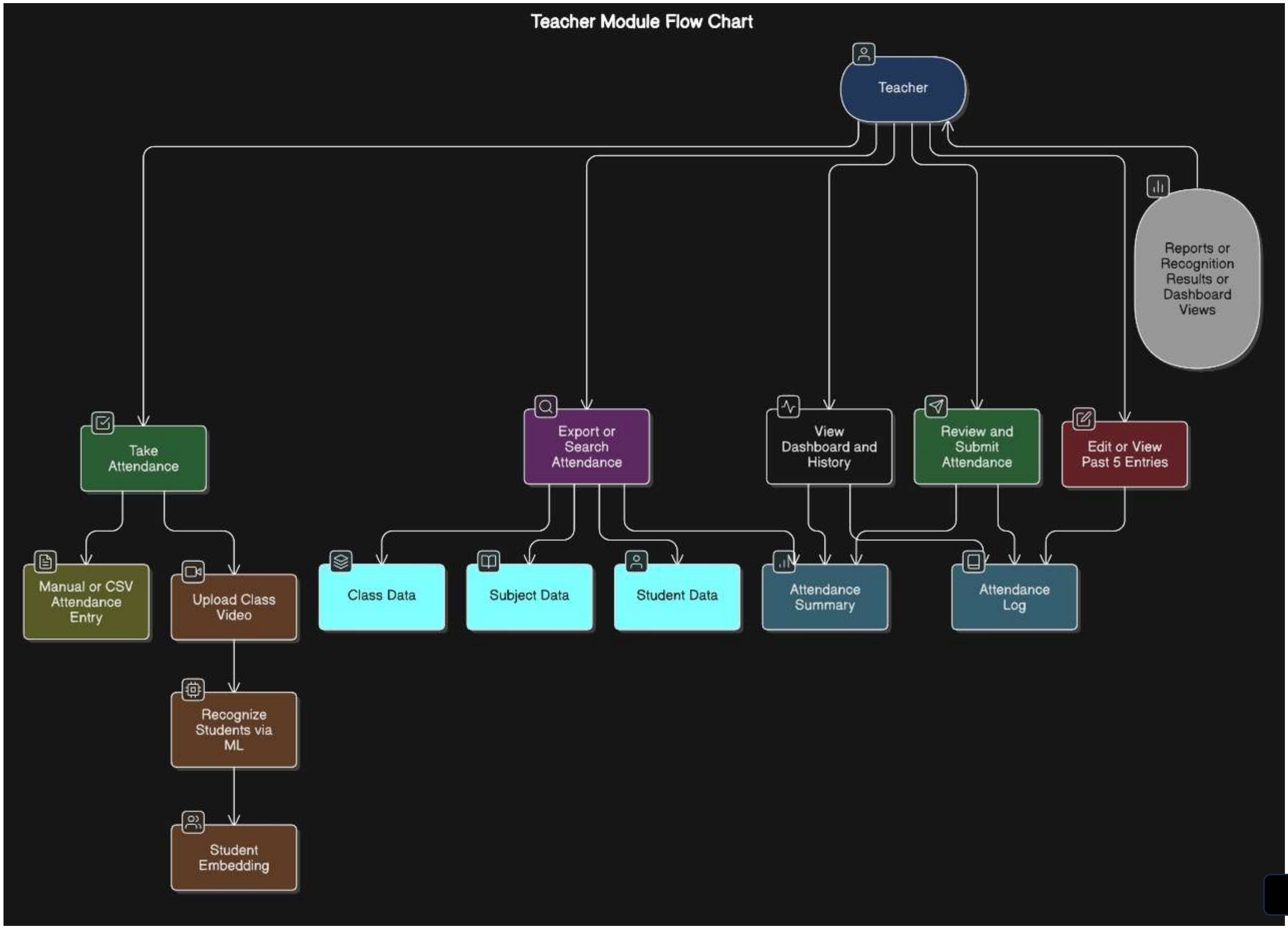
The **smart attendance system** is designed with different roles and features to make attendance tracking easier and more efficient. 🌟

1. **User Roles & Access** 🔑
   The system has three user roles: **Admin**, **Student**, and **Teacher**, each with different access levels to manage and view data.
2. **Admin Functionalities** 🛠️
   Admins have a special dashboard to **manage** everything! They can add, edit, or remove **students**, **teachers**, **classes**, and **subjects**. 📚 They can also upload bulk data through **CSV files** 📈, and everything is stored in a secure **SQLite database** for smooth management. 💾
3. **Student Functionalities** 👨‍🎓
   Students can view their personal data like name, class, and contact info along with their **attendance records**. 📝 They register with a **video upload** 📹 and use **facial recognition** 🧑‍💼 for future attendance tracking, making the process automated and secure. ✅
4. **Teacher Functionalities** 🍎
   Teachers have a dashboard where they can easily **view** the last 5 attendance records, **filter** by **class** or **subject**, and **search** by **Student ID**. 🔍 They can **export** attendance data 📊, **edit** or **update** records, and take attendance by selecting the class, subject, period, and date. 📅 Using **ML processing** 🎥, the system recognizes students from uploaded classroom videos, and teachers can **review** and submit the final attendance. ✅ If needed, teachers can **manually add** students or upload attendance via **CSV**. 🗂️

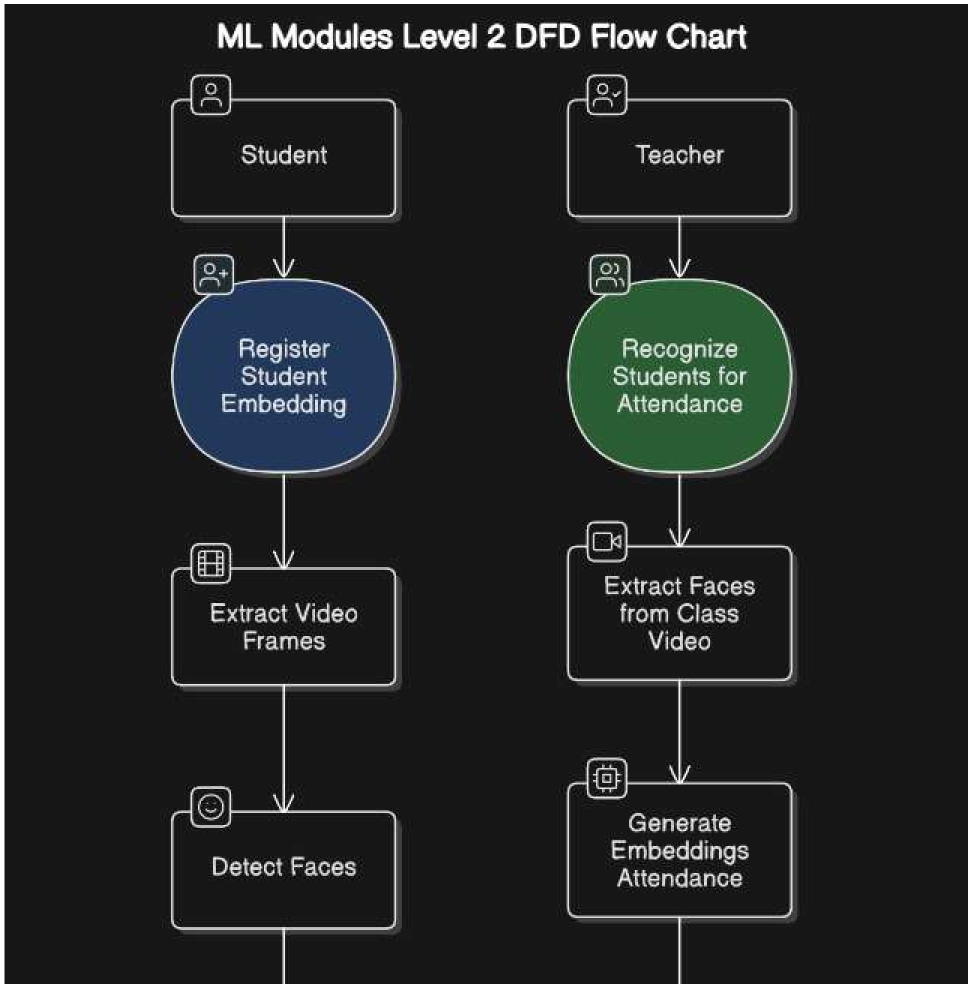

Admin Module
Start Administration
Manage Teacher Records
Manage Subject Records
Manage Student Records
Manage Class Records
Teacher Data
Subject Data
Upload Bulk Data via CSV
Student Data
Class Data
Confirmation Results Dashboards

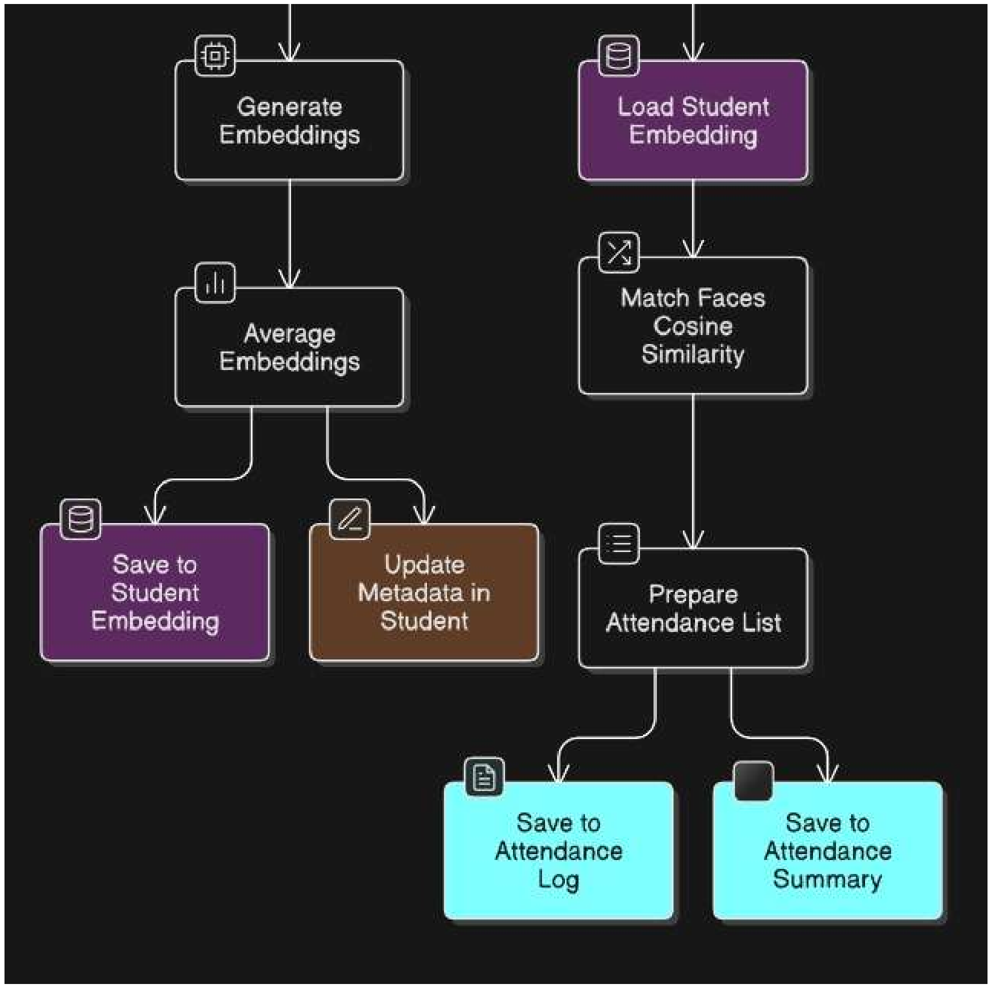
CSE IOTCS


Student Module Flow Chart
Start
View Personal and Attendance Info
Upload Registration Video
Attendance Summary
Student Data
Generate and Save Embedding
Dashboard Info or Confirmation Messages
Student Embedding


Teacher Module Flow Chart
Teacher
Reports or Recognition Results or Dashboard Views
Take Attendance
Export or Search Attendance
View Dashboard and History
Review and Submit Attendance
Edit or View Past 5 Entries
Manual or CSV Attendance Entry
Upload Class Video
Class Data
Subject Data
Student Data
Attendance Summary
Attendance Log
Recognize Students via ML
Student Embedding

CSE IOTCS


ML Modules Level 2 DFD Flow Chart
Student
Teacher
Register Student Embedding
Recognize Students for Attendance
Extract Video Frames
Extract Faces from Class Video
Detect Faces
Generate Embeddings Attendance


Generate Embeddings
Load Student Embedding
Average Embeddings
Match Faces Cosine Similarity
Save to Student Embedding
Update Metadata in Student
Prepare Attendance List
Save to Attendance Log
Save to Attendance Summary

The Machine Learning things…

Tools	Purpose & Key Functionality
OpenCV (cv2)	Video/Frame Extraction 🎥 : Extract frames from videos, process images for face detection
MTCNN	Face Detection 👀 : Detect faces in images and videos
DeepFace	Face Embedding 🧠 : Generate 128-dimensional face embeddings (using Facenet)
Facenet	Face Embedding Backend 🔬 : Convert faces into a vector representation (used within DeepFace)
Scipy	Cosine Similarity (Face Matching) 🔍 : Compute the similarity between embeddings using cosine distance
SQLAlchemy	Database Interaction 💾 : ORM for querying and saving data to the database (e.g., embeddings, student info)
Flask App Models	Data Storage and Retrieval 🗂️ : Store and retrieve user data and metadata (e.g., student profiles, attendance records)

Core Technologies Powering the System

🔧 **Software Stack:**

- **Frontend**: HTML, CSS, JavaScript (Jinja2 templating with Flask)
- **Backend**: Python (Flask microframework)
- **Database**: SQLite (via SQLAlchemy ORM)
- **Machine Learning Libraries**:
  - OpenCV (Video frame extraction)
  - MTCNN (Face detection)
  - DeepFace (Face embedding with Facenet)
  - NumPy, SciPy (Vector operations & cosine similarity)
- **Others**:
  - Werkzeug (Security, password hashing)
  - Flask-WTF (Form handling)
  - Pandas (CSV parsing)

Usefulness & Future Scope: Beyond Automation

Time Saving

Automates attendance marking, marking, saving time time for teachers.

Error Reduction Reduction

Reduces manual errors and proxy attendance.

Accessibility

Provides easy accessibility to attendance records.

Analysis

Helps in analyzing student participation participation trends. trends.

SmartTrack reduces errors, provides easy accessibility to attendance records, and helps in analyzing in analyzing student participation trends. Future enhancements include integration of marks & marks & assignments and a cloud-based database. The project offers easy access to attendance attendance records for both teachers and students.

Learning Outcomes: Skills Acquired

This project was a rich blend of **web development** and **machine learning**, allowing us to build an end-to-end real-world system. We learned:

- **Python and Flask** web development: Routing, forms, templates, session management.
- **Database modeling** using SQLAlchemy.
- **Facial recognition concepts**, including:
  - Face embedding using neural networks (Facenet via DeepFace)
  - Similarity comparison (Cosine Similarity)
  - Handling true/false positives and negatives in real data
- **Video processing** using OpenCV and handling edge cases like low lighting or frame skips.
- **Role-based access control** and multi-user interfaces.
- **Integration of ML into traditional web applications**, bridging the gap between AI models and user-facing platforms.

Thank You