Zewail City of Science, Technology and Innovation

CSAI School
CSAI 498 / CSAI 499

CSAI 498 / CSAI 499

Initial System Design & Work Division

Smart Attendance System

• Team Members:

Student Name	ID	Program
Hania Khaled	202202227	DSAI
Farah Samir	202201410	DSAI
Yassmin Raafat	202201706	SW
Hager Ali	202202011	DSAI

• Supervisor:

Dr. Mohamed Ghalwash

• Semester / Year:

Fall 2025

• Team number:

32

• Submission Date:

December 13, 2025

Table of contents:

1.Project summary:

This project introduces a **Smart Attendance System** that replaces manual attendance with an automated, AI-based solution.
Instead of analyzing full video streams, the system captures a **single image every minute**, detects and recognizes students, and calculates each student's **actual presence duration** based on how many frames they appear in.

Attendance records are stored in a **centralized database** and visualized through a **web/app dashboard**, allowing instructors to easily track attendance summaries and class insights in real time.

The system reduces manual effort, eliminates errors, and provides a more accurate measurement of student presence. Future enhancements include enabling **single-image student registration** to simplify onboarding and improve scalability.

2. Progress Since Proposal:

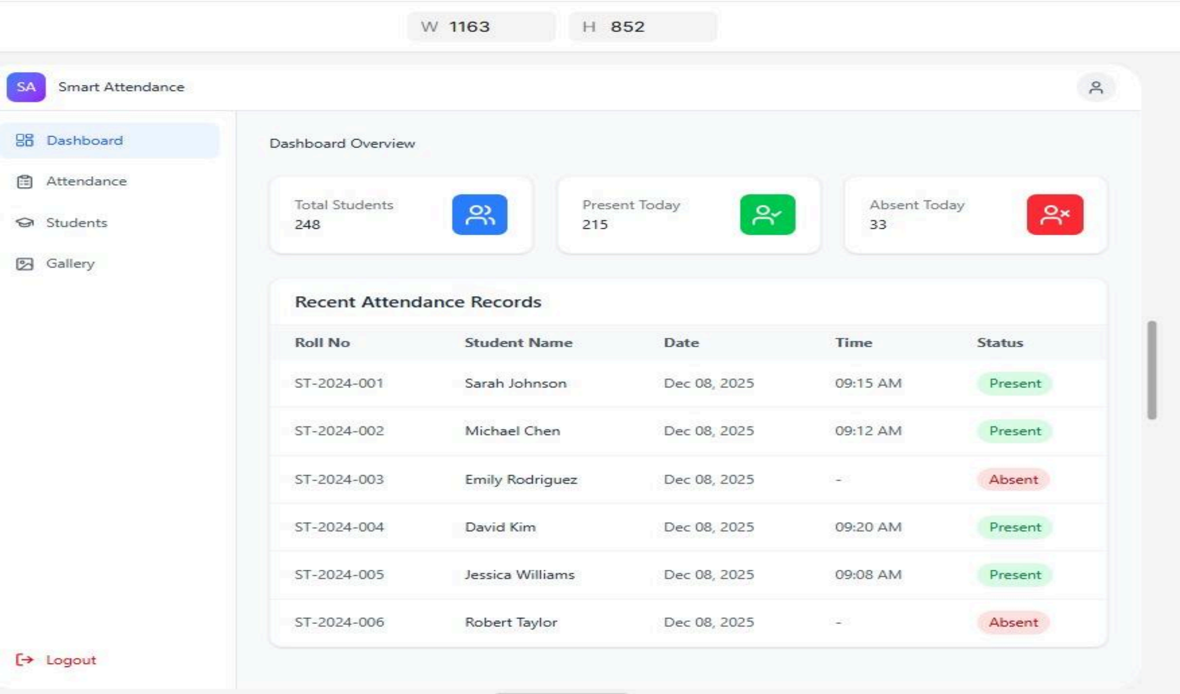
2.1 Completed Tasks (Since Week 4)

- **Data Collection:** All student images were collected and organized into structured class folders.
- **Image Preprocessing:** Cropped individual faces from the collected images using detected bounding boxes, added padding, and generated a clean dataset of face images.
- **Data Labeling:** Each cropped face was labeled with the corresponding student's **Name** and **ID** to prepare the dataset for training.
- **Model Training Started:** Initiated training of the **ResNet-based face recognition model** to learn student identities and return the correct Name/ID from an input image.
- **UI Development:**
    - Designed and implemented the **login page**.
    - Built the **dashboard** displaying attendance records.
    - Created the **student page**.
    - Developed preliminary **UI prototypes** outlining the overall system flow.
- **Repository Setup:** Initial project structure and commits were pushed to the GitHub repository.

- **Basic System Pipeline Testing:** Verified that the main pipeline works end-to-end:
  - Load image → detect face → crop → generate embedding → return identity prediction.

2.2 Research, Design, and Initial Coding

- Implemented a **two-step face detection pipeline** using **DNN SSD** as the primary detector with **Haar Cascades** as a fallback for missed detections.
- Designed the preprocessing workflow, including bounding box extraction, padding, cropping from the original image, and organizing the processed images into labeled folders.
- Started training the **ResNet-based face recognition model** to classify identities based on learned embeddings.
- Tested the full pipeline across multiple sample images to ensure detection, cropping, embedding, and prediction operate correctly.
- Developed initial UI components and **early mockups**, including the login interface, dashboard structure, and student attendance page.



Dashboard Overview

Total Students 248

Present Today 215

Absent Today 33

Recent Attendance Records

Roll No	Student Name	Date	Time	Status
ST-2024-001	Sarah Johnson	Dec 08, 2025	09:15 AM	Present
ST-2024-002	Michael Chen	Dec 08, 2025	09:12 AM	Present
ST-2024-003	Emily Rodriguez	Dec 08, 2025	-	Absent
ST-2024-004	David Kim	Dec 08, 2025	09:20 AM	Present
ST-2024-005	Jessica Williams	Dec 08, 2025	09:08 AM	Present
ST-2024-006	Robert Taylor	Dec 08, 2025	-	Absent

Figure 1: *Smart Attendance System dashboard showing attendance statistics and recent records*

3. System Architecture:

The Smart Attendance System is designed as a modular pipeline that processes periodically captured images, identifies students, and updates attendance records in a centralized database.

The system consists of five main components:

1. **Image Capture Module**
2. **Face Detection Module**
3. **Face Recognition Module**
4. **Attendance Processing & Database Module**
5. **Web/Application Dashboard**

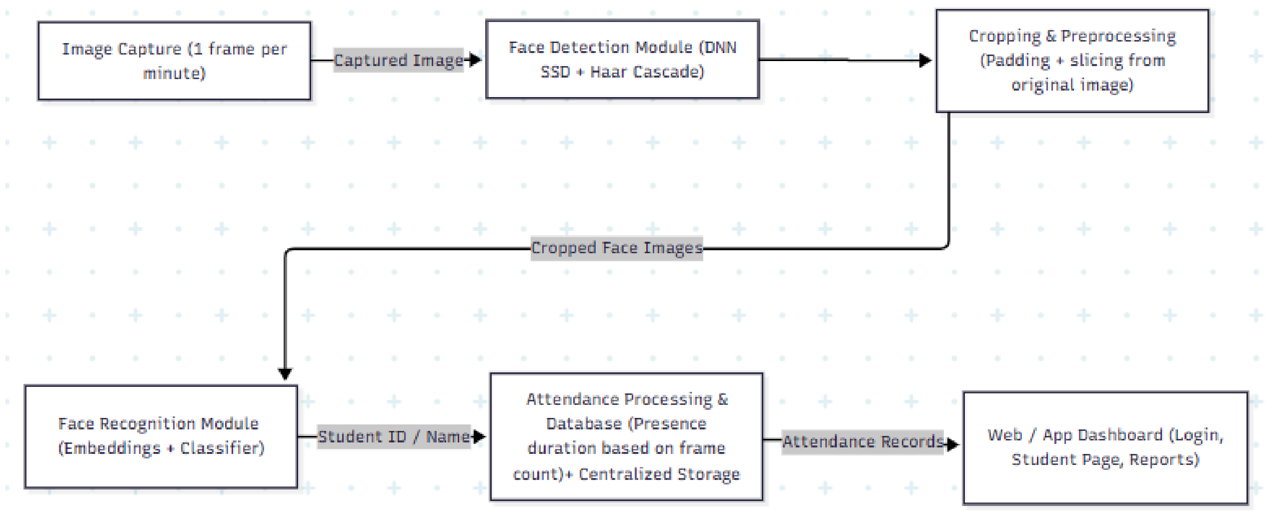


Image Capture (1 frame per minute)
Captured Image
Face Detection Module (DNN SSD + Haar Cascade)
Cropping & Preprocessing (Padding + slicing from original image)
Cropped Face Images
Face Recognition Module (Embeddings + Classifier)
Student ID / Name
Attendance Processing & Database (Presence duration based on frame count)+ Centralized Storage
Attendance Records
Web / App Dashboard (Login, Student Page, Reports)

Figure 2: *System Architecture Diagram*

4.Component Breakdown:

Component	Function	Input	Output	Main Functionality	Technologies Used
1. Image Capture Module	Periodically capture classroom images	Camera snapshot (every 1 min)	Raw image frame	Triggers timed image capture and forwards it to detection module	Python, OpenCV
2. Face Detection Module	Detect faces in each captured image	Raw image	Bounding boxes of detected faces	Uses DNN SSD to detect faces; Haar Cascades as fallback when SSD fails	DNN SSD (OpenCV DNN), Haar Cascades
3. Preprocessing & Cropping Module	Prepare clean face crops	Bounding boxes + original image	Cropped face images	Apply padding, crop faces, resize, organize images for recognition	NumPy, OpenCV
4. Face Recognition Module	Identify student identity	Cropped face image	Student Name + ID	Generate embeddings using ResNet model and classify identity	ResNet, PyTorch / TensorFlow
5. Attendance Processing Module	Calculate presence duration & store records	Student ID per frame	Attendance entries	Count frames where each student appears, compute attendance duration	Python, Custom Logic
6. Centralized Database	Store student data & attendance logs	Attendance records	Structured data	Maintain secure, real-time attendance storage & retrieval	Firebase / MySQL
7. Web/App Dashboard	Visualize attendance and manage system	Database data	Front-end pages	Show login, attendance summaries, student pages, and class reports	HTML/CSS, Flask / Bootstrap
8. User Authentication Module	Secure login to dashboard	Credentials	Authenticated user session	Validate users and restrict access	Flask Auth / JWT

5. Data Flow:

Image Capture → Face Detection

- Every minute, the Image Capture Module takes a classroom image.
- The raw image is sent to the Face Detection Module.

Face Detection → Preprocessing

- DNN SSD detects faces and returns bounding boxes.
- If SSD fails, Haar Cascades provide fallback detections.
- Bounding boxes and the original image are passed to the Preprocessing Module.

Preprocessing → Face Recognition

- The Preprocessing Module crops each face with padding and resizes it.
- Clean, labeled face crops are forwarded to the Face Recognition Module.

Face Recognition → Attendance Processing

- The ResNet-based model generates an embedding for each face and identifies the student (Name + ID).
- Identified students are passed to the Attendance Processing Module.

Attendance Processing → Database

- The system counts how many frames each student appears in.
- Presence duration is computed and stored in the centralized database as attendance records.

Database → Dashboard

- The dashboard retrieves attendance data from the database.
- Displays summaries, individual student records, and class statistics to instructors.

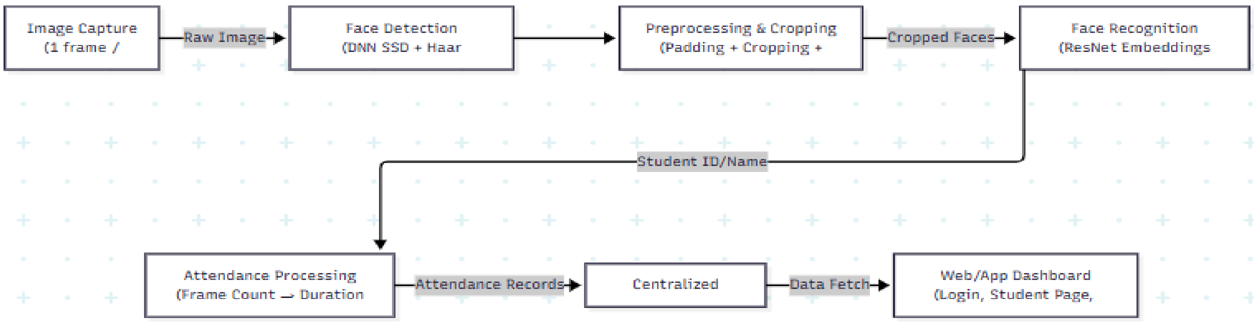


Figure 3: *System Data Flow Diagram*

6. Work Breakdown Structure (WBS):

Task	Responsible Team Member	Start Date	End Date	Status
Data Collection (Student Images)	All Members	Week 4 fall	Until now	In Progress
Face Detection Pipeline (SSD + Haar)	Hania	Week 10	Week 3 spring	In Progress
Image Preprocessing (Cropping + Padding)	Farah	Week 10	Week 14	In Progress
Dataset Labeling (Names + IDs)	Hager	Week 10	Week 14	In Progress
Face Recognition Model Setup	Farah, Hager, Hania	Week 11	Until now	In Progress
Model Training & Evaluation	Farah, Hager, Hania	Week 11	Until now	In Progress
UI Development	Yassmin	Week 11	Until now	In Progress
Backend Integration (Recognition → Attendance Logic)	All Members	Week 3 spring	End of spring	Planned
Attendance Processing Module	Farah, Hager, Hania	Week 1 spring	Week 6 spring	Planned
Database Setup (Students + Attendance)	Yassmin	Week 1 spring	Week 6 spring	Planned
Dashboard Data Visualization	Yassmin	Week 1 spring	Week 6 spring	Planned

Final Testing (End-to-End Pipeline)	ALL Members	Week 6 spring	Week 12 spring	Planned
Deployment Setup (Local/Web)	Yassmin	Week 13 fall	Week 9 spring	Planned
Documentation & Report Writing	ALL Members	Week 4 Fall	Week 13 spring	In Progress

7.Risk Analysis:

Risk Type	Description	Impact	Mitigation Strategy
Technical	Low accuracy in face recognition due to poor image quality or occlusions	High	Use high-quality image dataset, implement padding, data augmentation, and ResNet embeddings.
Technical	Face detection failures in certain lighting conditions	Medium	Two-stage detection: DNN SSD as primary, Haar Cascade as fallback; preprocess images to normalize lighting
Timeline	Delays in UI or model integration	High	Parallelize module development, schedule weekly progress checks, prioritize critical components first
Resource	Team members unavailable due to personal reasons	Medium	Reassign tasks, maintain clear documentation, implement backup responsibilities
Data	Incomplete or inconsistent dataset	High	Validate data collection, standardize preprocessing, and double-check labeling accuracy

8.References:

1] Khaled, K. (n.d.). *Computer vision playlist* [Video playlist]. YouTube. https://youtube.com/playlist?list=PL5JZLxl_tFCeB5bOLYGXNdjnmgRIU6Ov0

2] OpenCV Team. (n.d.). *OpenCV library*. https://opencv.org/