

# **Intel Unnati Project Report**

## **Problem Statement: AI-Powered Interactive Learning Assistant for Classrooms**

**Project Title:** Smart Classroom Monitor

GITAM University, Bengaluru

**Team Members:** 1. Abhinay Reddy M  
2. Edimudi Gowtham

## **Table of Contents:**

1. Project Description
2. Objectives
3. Technologies Used
4. Module-wise Explanation
  - o Attendance System
  - o Emotion Detection
  - o Hand Raise Detection
  - o Face Detection
5. Streamlit User Interface
6. Challenges Faced
7. Conclusion
8. References
10. Screenshots

## **1. Project Description**

The Smart Class Monitor is an AI-powered system designed to automate classroom monitoring and analysis. It captures attendance, detects emotions, identifies hand raises, and summarizes the engagement level of each class session. The solution leverages facial recognition and emotion detection models, supported by a user-friendly Streamlit interface.

## **2. Objectives**

- Automate the process of marking student attendance using facial recognition.
- Detect students' emotional states to gauge class engagement.
- Identify hand-raising gestures to capture interactive participation.
- Summarize session data in a structured format for educators.

### **3. Technologies Used**

<b>Component</b>	<b>- Technology</b>
Programming Language	- Python 3
Interface	- Streamlit
Face Recognition	- face_recognition, OpenCV
Emotion Detection	- Pretrained FER model (fer.json, .h5)
Gesture Recognition	- Mediapipe
Data Storage	- CSV (attendance), JSON (logs)
Visualization	- Streamlit components + matplotlib

## 4. Module-wise Explanation

---

### A. Attendance System (`mark_attendance.py`)

- Loads known faces from the `known_faces/` directory.
  - Uses the `face_recognition` library to compare live webcam frames.
  - Records attendance once per person per day in `attendance.csv`.
  - Displays recognized user's name in the webcam frame.
  - Skips duplicate attendance entries on the same date.
- 

### B. Emotion Detection (`emotion_hand_combined.py`)

- Captures frames continuously through the webcam. Detects face and emotion for each student.
  - Detects faces and emotions from live camera feed. Logs only non-neutral emotions to `emotion_log.json`.
  - Matches detected faces with known encodings for identification.
  - Logs only non-neutral emotions (like Happy, Sad, Angry, etc.) to `emotion_log.json` every 10 minutes.
  - Model Used: Pretrained FER (Facial Emotion Recognition) model using Keras.
- 

### C. Hand Raise Detection (`hand_raise_detect.py`)

- Uses Mediapipe's holistic model for landmark detection.
  - Detects if right hand is raised based on wrist/shoulder comparison.
  - Records name, timestamp, and hand raise count into `hand_raise_log.json`.
  - Real-time detection using webcam.
- 

### D. Face Detection (`face_detect_test.py`)

- Standalone face recognition module.
  - Displays recognized person's name within the webcam feed.
  - Updates `current_person.txt` for real-time UI feedback.
- 

## 5. Streamlit UI Overview

- ⚙️ Start/Stop Attendance
  - ⌚️ Start/Stop Engagement Session (Emotion + Hand Raise)
-

-  Live Face Detection
-  Generate & View Daily Summary
-  View Past Summaries
-  End Class (Kills All Python Processes)

The UI has intuitive styling with color-coded buttons, toggle displays, and error handling.

---

## 6. Challenges Faced

- Integrating OpenCV, face recognition, and emotion models into one pipeline.
- Handling incompatible image formats like .tiff for emotion detection.
- Fixing cv2.imshow() related GUI issues during Streamlit integration.
- Resolving dependency conflicts (e.g., TensorFlow, Keras, protobuf, jax).
- Ensuring modules can be stopped and restarted seamlessly via the UI.

---

## 7. Conclusion

The project successfully integrates multiple AI modules into a unified system that improves classroom monitoring. It reduces manual tasks, enables deeper insight into student engagement, and maintains accuracy and usability. The modular structure allows for future expansion (e.g., voice command, quiz scoring).

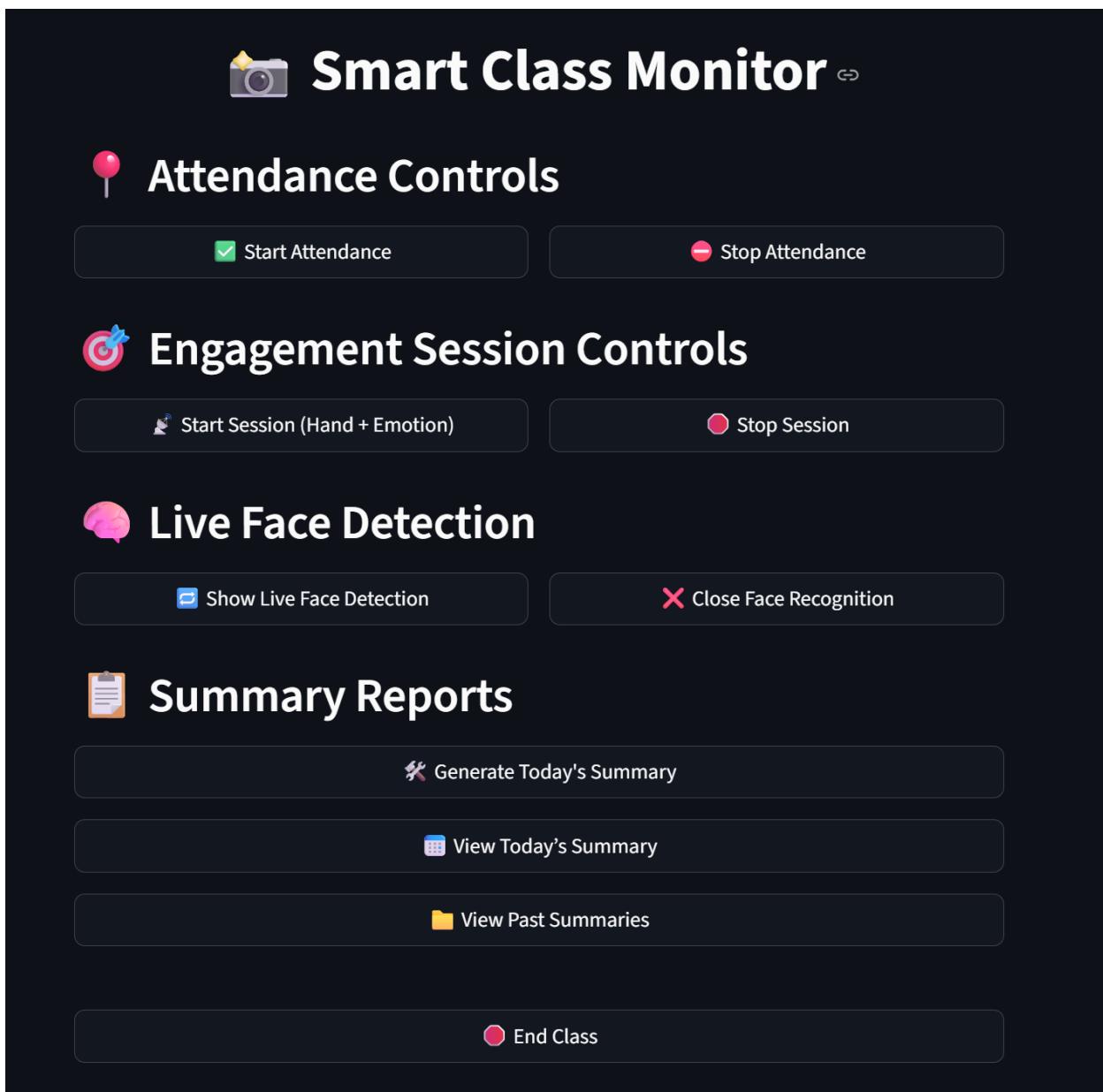
---

## 8. References

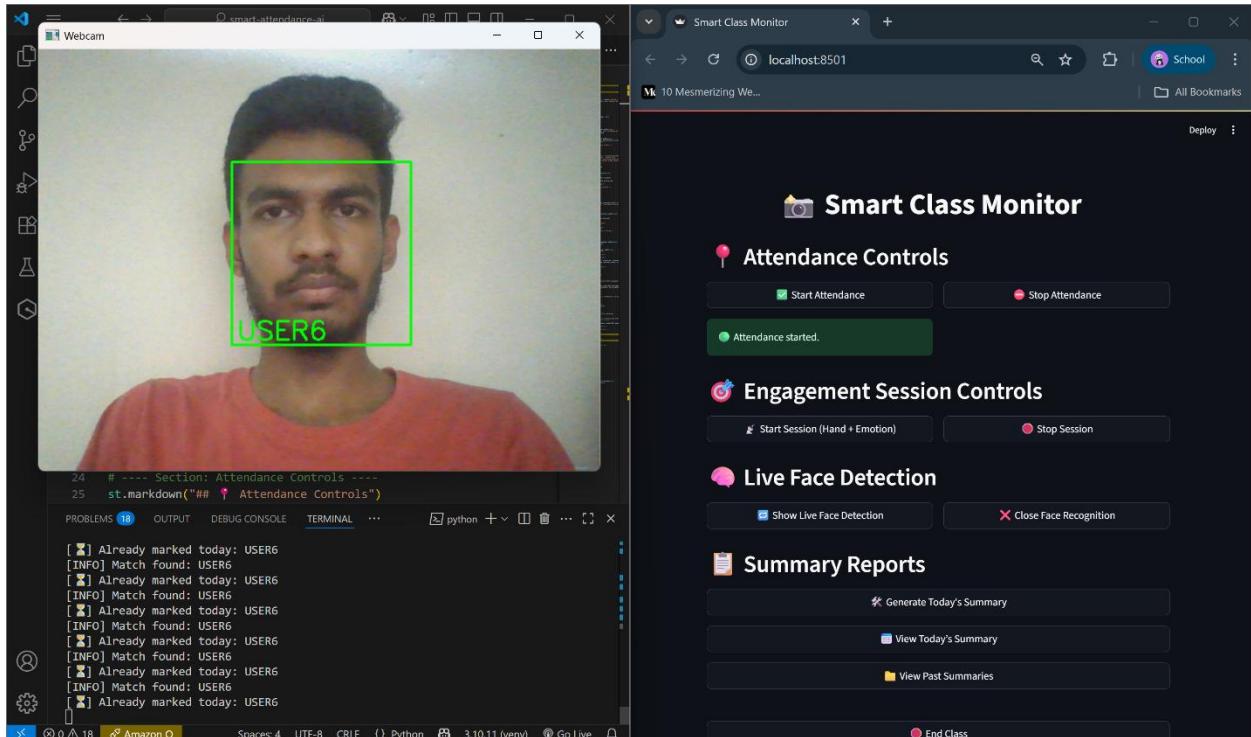
- [face\\_recognition GitHub](#)
  - [FER Model](#)
  - Mediapipe Documentation
  - Streamlit Docs
  - OpenCV Docs
-

## 9. Screenshots

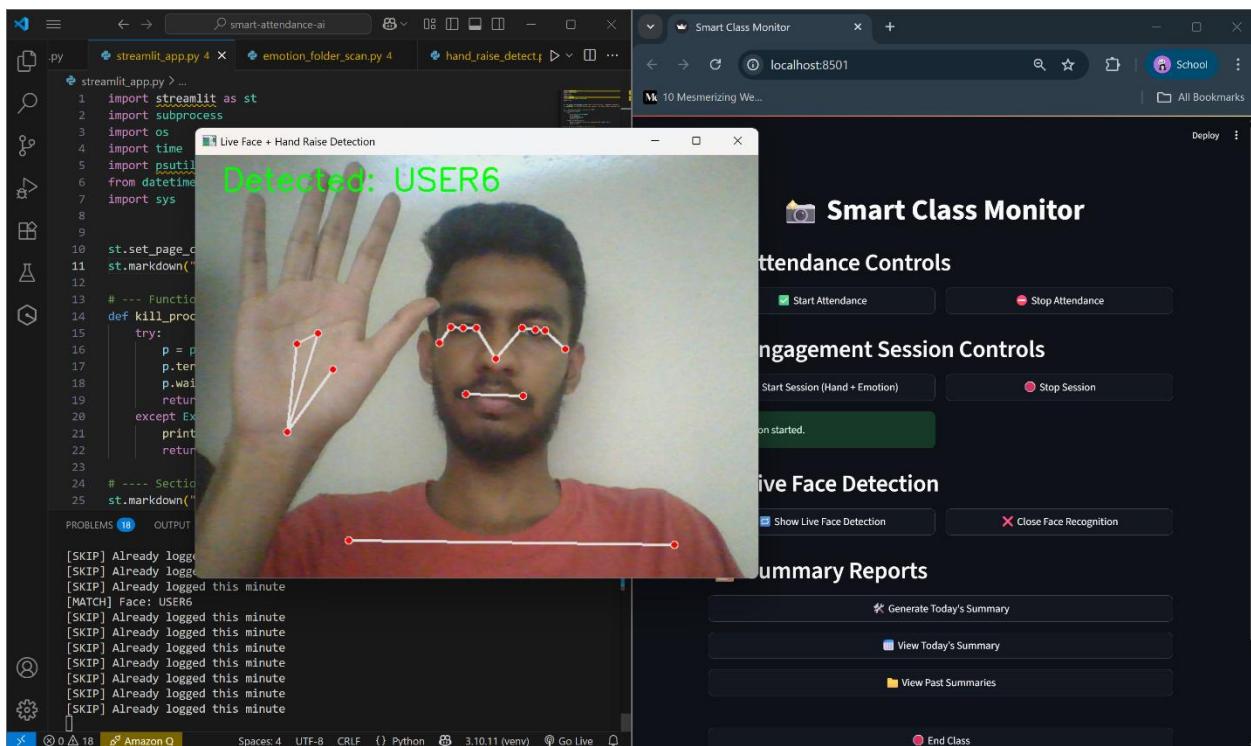
Streamlit UI:



## Face Detection:



## Hand Raise Detection:



## Summary Reports:



# Summary Reports

Generate Today's Summary

View Today's Summary

Today's Summary

\*\*Class Session Summary\*\*

Date: 2025-07-11 22:58:52

Attendance:

- USER6

Emotions Detected:

- USER7: Disgust (3)

- USER10: Happy (3)

- USER9: Fear (3)

Engagement (Hand Raises):

- USER6: 3 time(s)

Close Today's Summary

View Past Summaries

End Class