# ****Project Report: Face Recognition Attendance System Using OpenCV and Firebase****

## 1. Title:

**Face Recognition Attendance System Using Open CV and Firebase**

## 2. Objective:

To develop an automated attendance marking system using face recognition technology integrated with Firebase for real-time data management and seamless classroom attendance tracking.

## 3. Tools and Technologies Used:

**Python**

**OpenCV**

**face\_recognition** (d lib-based face detection and recognition)

**Firebase Realtime Database** (for attendance and student data storage)

**Firebase Cloud Storage** (for storing student images)

**CV zone** (for enhanced UI overlays and display)

**NumPy**

## 4. Project Overview:

This project replaces traditional manual attendance with an automated system that identifies students via webcam using face recognition. It integrates with Firebase to store student profiles and attendance data in real-time. The system retrieves student details from Firebase, verifies attendance based on timestamp restrictions, and visually displays student information alongside their face on a custom interface.

## 5. Key Features:

Real-time face recognition leveraging Open CV and face\_recognition libraries

Firebase integration for centralized, real-time student data and attendance management

Retrieval of student images from Firebase Storage for display overlays

Prevention of duplicate attendance marking within a configurable time threshold

Custom graphical interface including background images and information overlays for better UX

## 6. System Workflow:

Capture live video feed using Open CV.

Resize and convert video frames to RGB for faster face detection.

Detect and encode faces in the current frame.

Compare detected face encoding against the known database.

Fetch student profile and attendance info from Firebase.

Check last attendance timestamp to avoid duplicates.

If eligible, update attendance in Firebase.

Overlay student details and image on the video display in real-time.

## 7. Firebase Database Structure:

markdown

CopyEdit

Students/

├─ student\_id/

├─ name

├─ major

├─ year

├─ total\_attendance ├─ starting\_year

├─ last\_attendance\_time

├─ standing

## 8. Error Handling

Manages missing or corrupt image files (supports .jpg, .png, .jpeg).

Prevents attendance marking if face detected but corresponding image is missing in Firebase Storage.

Displays fallback UI for loading, duplicate attendance attempts, or errors during recognition or data fetching.

## 9. Limitations:

Recognition accuracy depends on image quality and face angle.

Performance can degrade in poor or inconsistent lighting conditions.

Optimized primarily for frontal face images; side profiles may not register well.

## 10. Future Improvements:

Implement GUI-based student registration with photo upload directly to Firebase.

Support multiple faces in a single frame for batch attendance marking.

Add email or SMS notifications to students or faculty upon attendance updates.

Maintain historical attendance logs and analytic dashboard.

## 11. Conclusion:

This project successfully implements a prototype of a face recognition attendance system that combines robust facial recognition with Firebase’s cloud capabilities. It automates attendance with real-time updates and improves accuracy and efficiency in classroom management.

## 12. Error Debugging - Installing face\_recognition and Dependencies:

To install the face\_recognition module and its dependencies, especially on Windows, follow these steps:

**Step 1:** Install C Make from https://cmake.org/download/

**Step 2:** Install **Microsoft Visual C++ Redistributable** from  
[https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170](https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170" \t "_new)

**Step 3:** Install all required C++ packages included with the redistributable.

**Step 4:** Verify Python version in command prompt:

bash

CopyEdit

python --version

**Step 5:** Download the matching .whl file for your Python version from the GitHub repo:  
[https://github.com/z-mahmud22/Dlib\_Windows\_Python3.x](https://github.com/z-mahmud22/Dlib_Windows_Python3.x" \t "_new)  
(e.g., for Python 3.10, download dlib-19.22.99-cp310-cp310-win\_amd64.whl)

**Step 6:** Open command prompt and navigate to the download folder.

**Step 7:** Install the wheel file using pip:

bash

CopyEdit

pip install dlib-19.22.99-cp310-cp310-win\_amd64.whl

**Step 8:** Once d lib is installed, run:

bash

CopyEdit

pip install face\_recognition