# 5G Face Recognition Attendance System Setup Guide

This document provides a step-by-step walkthrough for building and upgrading a face recognition-based attendance system that fetches student data from a Google Form and logs attendance into a PostgreSQL database.

## 1. Initial Setup and Environment Preparation

### ✅ Required Libraries

Install the following Python libraries using pip:

pip install opencv-python  
pip install face\_recognition  
pip install pandas  
pip install requests  
pip install openpyxl  
pip install gdown  
pip install psycopg2-binary

## 2. Basic Face Recognition from Webcam

Initially, a script was written to: - Capture webcam feed using OpenCV - Load known faces from a local folder and encode them - Use face\_recognition to detect and match real-time faces - Display recognized names on the webcam feed

Attendance logging was not implemented in this phase.

## 3. Manual Entry-Based Attendance Logging

After face recognition, attendance was manually logged with hardcoded values: - Subject and lecture slot were defined as fixed strings - A CSV file or log file was used to record the entries

This stage helped verify the face matching and logging flow.

## 4. Using Google Form Excel for Student Details

A Google Form was created to collect: - Timestamp, Email Address, ID, Name, Branch, Batch, and Google Drive Photo Link

The responses were exported as an .xlsx file. The code was updated to: - Periodically download the updated Excel file using a direct Google Docs export URL - Save it locally to the E: drive - Use pandas to parse and match names to their respective IDs

## 5. Downloading Images from Google Drive Links

Student photos were submitted as Google Drive links. To sync them: - The image file ID was extracted from the link (via id= or /d/ pattern) - Images were downloaded using gdown or direct HTTP requests - They were renamed to a consistent format: ID\_Name.jpg and saved locally

### Issues Addressed:

* Invalid or incomplete URLs
* Non-image files
* Handling repeated downloads

## 6. Dynamic Attendance Logging with PostgreSQL

To upgrade from CSV logging, a PostgreSQL database was introduced:

### Database Schema

* Table: attendance\_logs
* Fields: id, date, time, name, branch, batch, subject, lecture\_slot

### How It Works:

* When a face is recognized, its name is matched against Excel data
* If found, corresponding details (ID, branch, batch) are fetched
* Attendance is inserted into the database if not already logged

### Tools Used:

* psycopg2 for PostgreSQL integration
* face\_recognition for encoding/matching faces
* pandas to handle Excel parsing
* datetime for timestamps

## 7. Error Handling & Debugging

During development, several issues were debugged: - UnidentifiedImageError from PIL: caused by incomplete downloads - Missing modules due to incorrect file naming - Parameter mismatch in functions (e.g., log\_attendance) - Face match failures due to incorrect image naming

## 8. Future Improvements

Suggested next steps: - Migrate from local Excel to Google Sheets API for real-time updates - Build a backend API using Flask or FastAPI - Add frontend UI for instructors/admin - Deploy the complete system using Docker to a 5G Edge Server

## 9. Folder Structure

5G Face Attendance System/  
├── edge-face-recognition/  
│ ├── liveFaceRecognition.py  
│ ├── syncKnownFaces.py  
│ ├── attendance\_logger.py  
├── known faces/ # Synced face images  
├── form\_responses.xlsx # Downloaded Excel from Google Form

This concludes the documented development history of your intelligent 5G face recognition attendance system.