Attend-X Final Setup Documentation

1. Project Overview

Attend-X is a robust Face Attendance System built with Python (Backend) and HTML/JS (Frontend).
It uses Supabase (PostgreSQL) for data storage and face_recognition for biometrics.
This version implements Multi-Admin Isolation, Strict Security Rules, and Production-Ready structure.

2. Architecture

Frontend: HTML5, Vanilla JS, CSS3 (Client-side logic)
Backend: Flask API (Python) serving 4 main services:

- Auth Service: JWT Verification
- Face Service: Encoding & Matching (0.55 threshold)
- Attendance Service: Logic & Logging
- Database Service: Supabase Connection

Database: Supabase (PostgreSQL) with RLS enabled.

3. Database Schema

Tables:

- admins (id, user_id, email, college)
- students (id, admin_id FK, name, roll_number, photo_url)
- face_encodings (id, student_id FK, encoding JSON)
- attendance (id, student_id FK, date, status, verified)

Key Changes:

- Added admin_id to students/attendance for isolation.
- Moved face encodings from pickle file to 'face_encodings' table.
- Enabled RLS policies.

4. API Endpoints

POST /register_face - Register student (Admin Auth Required)
POST /verify_face - Verify face only
POST /mark_attendance - Verify & Mark Attendance (Secure)
POST /delete_face - Remove face data (Admin Auth Required)

POST /delete_student_data - Cleanup student data

5. Security Improvements

1. JWT Authentication: Admin routes protected by Supabase Auth Token.
2. Data Isolation: Admins can only see/edit their own students.
3. Backend Validation: Attendance marking moved to backend endpoint.
4. Cascade Delete: Deleting a student removes face data and attendance logs.
5. Face Confidence: Minimum 72% confidence (0.55 distance) enforced.

6. Deployment Steps

1. Database: Run 'backend/FINAL_DATABASE_SETUP.sql' in Supabase SQL Editor.
2. Environment: Create .env in backend/ with:
```
SUPABASE_URL=...
SUPABASE_KEY=...
```
3. Backend: Run 'pip install -r requirements.txt' then 'python app.py'
4. Frontend: Serve via any static host (Vercel, Netlify) or local server.

7. Folder Structure

```
backend/
  app.py (Main Application)
  auth_service.py (Security)
  face_service.py (Biometrics)
  database_service.py (Data Layer)
  attendance_service.py (Logic)
  FINAL_DATABASE_SETUP.sql (Schema)
frontend/
  index.html, login.html... (UI)
  script-fixed.js (Logic)
```

Cleaned up unused files and legacy scripts.