Software Requirement Specification (SRS)

Face Recognition Attendance System

Using OpenCV + FaceNet | Python Flask | Deploy on Render

1. Introduction

Purpose: To develop a web-based Face Recognition Attendance System that automatically marks attendance
using OpenCV and FaceNet.

Scope: Capture face images, generate embeddings, recognize users, mark attendance automatically,
and provide an admin dashboard. Deployment will be done on Render cloud platform.

2. Technologies Used

Backend: Python, Flask
Computer Vision: OpenCV
Deep Learning Model: FaceNet
Database: MongoDB / MySQL
Frontend: HTML, CSS, Bootstrap
Deployment: Render

3. Functional Requirements

• User Registration with face dataset collection
• Face embedding generation using FaceNet
• Real-time face recognition
• Automatic attendance marking
• Admin dashboard
• Attendance export (CSV/Excel)

4. Non-Functional Requirements

Performance: Recognition time < 2 seconds
Security: Admin authentication required
Usability: Simple and responsive UI
Reliability: 95% uptime

5. System Architecture

Webcam → OpenCV Face Detection → FaceNet Embedding → Cosine Similarity Matching → Database Storage → Flask Backend → Admin Dashboard

6. Database Design

Users Table:
- user_id
- name
- email
- embeddings
- registration_date

Attendance Table:
- attendance_id
- user_id
- date
- time
- status

7. Deployment on Render

1. Push project to GitHub
2. Create Web Service on Render
3. Add build command: pip install -r requirements.txt
4. Add start command: gunicorn app:app
5. Deploy

8. Future Enhancements

- Liveness detection
- Mobile app integration
- Cloud storage integration
- Real-time notifications