Biometric Attendance System – Full Technical Documentation

# 1. Overview

This project is a Biometric Attendance Management API built using Laravel 12, integrated with ZKTeco biometric devices (e.g., U270) via ADMS communication.   
It supports attendance data collection, device management, and remote control commands such as rebooting, clearing logs, or syncing users.   
The system runs under Nginx with PHP-FPM and HTTPS, using SQL Server as the backend database.

# 2. Core Components

|  |  |
| --- | --- |
| Component | Purpose |
| Laravel API | Handles device communication, admin APIs, and data storage. |
| SQL Server | Stores attendance data, device records, and user commands. |
| ZKTeco Devices | Push attendance logs and receive commands. |
| Nginx + PHP-FPM | Web server for serving Laravel app securely via HTTPS. |
| Sanctum Authentication | Protects admin endpoints for command management. |

# 3. Communication Flows

There are two main communication flows between devices and the server:

• Device → Server (Push attendance data)  
• Server → Device (Send commands)

# 4. Command Types

Supported commands include: REBOOT, CLEAR\_ATTLOG, CLEAR\_ALL\_USERS, DELETE\_USER, SET\_TIME, ENABLE, DISABLE, and SYNC\_USER.

SYNC\_USER allows syncing user profile and biometric templates (fingerprint or face) to devices.

# 5. SYNC\_USER Workflow

1. Admin sends SYNC\_USER command with user data.  
2. Server stores it in the command queue.  
3. Device polls /api/adms/commands and retrieves the command.  
4. Device executes by enrolling/updating the user locally.  
5. Device acknowledges completion back to /api/adms/ack.  
6. Server marks the command as acknowledged.

# 6. Legacy ADMS Support

Older devices use /iclock/getrequest and /iclock/devicecmd for plain text communication.  
Commands like REBOOT are sent as text: 'C:1:REBOOT', and user data updates are sent via 'DATA UPDATE' lines.

# 7. Security

- Devices authenticate using SN + push\_token from the database.  
- Admin APIs use Laravel Sanctum bearer tokens.  
- HTTPS is enforced with Let’s Encrypt SSL certificates.

# 8. Database Summary

|  |  |
| --- | --- |
| Table | Purpose |
| devices | Stores registered biometric devices and credentials. |
| punches | Stores attendance punch records from devices. |
| device\_commands | Tracks command queue and statuses. |
| people | Contains employee profiles linked to device enrollments. |
| biometric\_templates | Stores Base64 biometric templates for each user. |

# 9. Deployment and Server Setup

• Application Path: /var/www/html/SomApp/biometric-attendance  
• PHP-FPM Socket: /run/php/php8.2-fpm.sock  
• Nginx Config: /etc/nginx/sites-enabled/biometric-attendance  
• HTTPS: Managed by Certbot using Let’s Encrypt  
• Public Endpoint: https://nidaam.somict.so

# 10. Typical Operation Flow

1. Admin issues command (e.g., REBOOT or SYNC\_USER).  
2. Server queues it as pending.  
3. Device polls /api/adms/commands → receives the command.  
4. Device executes → sends acknowledgment to /api/adms/ack.  
5. Server updates command status (ack/failed).

# 11. Health Check

To verify server health:

curl https://nidaam.somict.so/api/health → {"ok":true}

# 12. Summary Diagram

Admin Panel ←→ Laravel API Server ←→ SQL Server  
 ↑  
 │  
Biometric Device (ZKTeco) ↔ API (ADMS Push/Poll)

This architecture ensures full control and monitoring of biometric devices and attendance data from a centralized Laravel backend.