**🚀 DỰ ÁN FACE DETECTION WITH NAME RECOGNITION**

**📋 TỔNG QUAN DỰ ÁN**

Đây là một hệ thống quản lý điểm danh thông minh sử dụng AI Face Recognition, bao gồm 3 phần chính:

* **Frontend Web (React)** - Quản lý hệ thống qua web
* **Mobile App (Flutter)** - Ứng dụng điểm danh trên mobile
* **Backend (Node.js)** - API server và xử lý AI

**🌐 FRONTEND WEB - REACT (my-app/)**

**Cấu trúc dự án:**

* **Framework:** React 18.2.0 với Create React App
* **Routing:** React Router DOM v7.8.0
* **Styling:** CSS-in-JS với custom styles
* **State Management:** React Hooks (useState, useEffect)
* **HTTP Client:** Fetch API với custom service layer

**Các trang chính và chức năng:**

**1. Login.jsx - Trang đăng nhập**

**Chức năng:**

* Form đăng nhập với username/password
* Validation form với error handling
* Redirect dựa trên role (admin, teacher, student)
* Tích hợp với auth service
* Loading states và error messages
* Auto-redirect nếu đã đăng nhập

**2. AdminDashboard.jsx - Dashboard quản trị**

**Chức năng:**

* Hiển thị thống kê tổng quan hệ thống
* Cards thống kê với hover effects
* Quick actions cho admin
* Management cards cho các chức năng chính
* Real-time statistics updates
* Navigation đến các trang quản lý

**3. ClassManagement.jsx - Quản lý lớp học**

**Chức năng:**

* CRUD operations cho lớp học
* Quản lý sinh viên trong lớp
* Giao diện responsive
* Search và filter classes
* Add/remove students from classes
* Class schedule management

**4. UserManagement.jsx - Quản lý người dùng**

**Chức năng:**

* Quản lý thông tin user (admin, teacher, student)
* Phân quyền và role management
* Tích hợp với face recognition
* User activation/deactivation
* Bulk user operations
* User profile editing

**5. FaceRegistration.jsx - Đăng ký khuôn mặt**

**Chức năng:**

* Upload video để tạo dataset
* Training model nhận diện
* Preview và validation
* Face detection feedback
* Training progress monitoring
* Dataset quality assessment

**6. AttendanceSessionManagement.jsx - Quản lý buổi điểm danh**

**Chức năng:**

* Tạo và quản lý session điểm danh
* Theo dõi attendance real-time
* Reports và analytics
* Session scheduling
* Attendance statistics
* Export attendance data

**7. Subject\_ScheduleManagement.jsx - Quản lý môn học và lịch**

**Chức năng:**

* Quản lý danh sách môn học
* Tạo lịch học cho từng lớp
* Time slot management
* Teacher assignment
* Schedule conflicts detection

**Services:**

* **api-service.js:** HTTP client với authentication, error handling
* **auth-service.js:** Quản lý authentication và authorization
* **camera-service.js:** Xử lý camera và face capture

**Components:**

* **Header, Sidebar, Notification, LoadingOverlay**
* **Custom styling system** với CSS-in-JS
* **Responsive design** cho mobile và desktop

**📱 MOBILE APP - FLUTTER (lib/)**

**Cấu trúc dự án:**

* **Framework:** Flutter với Dart
* **Target Platform:** Android (với hỗ trợ iOS)
* **Architecture:** Service-oriented với StatefulWidget
* **State Management:** Flutter StatefulWidget + Services
* **Camera Integration:** camera package với OpenCV

**Các màn hình chính và chức năng:**

**1. Login Screen (login\_screen.dart)**

**Chức năng:**

* Email/password authentication
* Form validation với error handling
* Loading states và progress indicators
* Error handling với SnackBar
* Auto-redirect dựa trên user role
* Integration với AuthService
* Offline login support

**2. Student Dashboard (student\_dashboard.dart)**

**Chức năng:**

* **Tab 1: Lịch học (Schedules)**
* Hiển thị thời khóa biểu theo ngày
* Subject và teacher information
* Time slot details
* **Tab 2: Phiên điểm danh (Active Sessions)**
* Danh sách session đang hoạt động
* Join session functionality
* Real-time session updates
* **Tab 3: Lịch sử điểm danh (Attendance History)**
* Theo dõi attendance records
* Attendance status tracking
* Historical data view

**3. Face Capture Screen (face\_capture\_screen.dart)**

**Chức năng:**

* **Camera Integration:**
* Front/back camera switching
* Camera initialization và control
* Permission handling
* **Face Detection:**
* Real-time face detection với OpenCV
* Face alignment và positioning
* Quality assessment
* **Image Capture:**
* High-quality image capture
* Flash control (auto/manual/off)
* Image preview và validation
* **Upload Integration:**
* Gửi ảnh lên server để attendance
* Progress tracking
* Error handling

**4. Teacher Dashboard (teacher\_dashboard.dart)**

**Chức năng:**

* **Tab 1: Lịch học (Schedule)**
* Xem lịch giảng dạy
* Class information
* Time management
* **Tab 2: Phiên điểm danh (Attendance Sessions)**
* Tạo session điểm danh mới
* Quản lý session đang hoạt động
* Đóng session
* Real-time attendance monitoring

**5. Admin Dashboard (admin\_dashboard.dart)**

**Chức năng:**

* System statistics overview
* Quick access management cards
* User management shortcuts
* System health monitoring
* Configuration access

**6. Session List Screen (session\_list\_screen.dart)**

**Chức năng:**

* Hiển thị danh sách session
* Filter và search sessions
* Session details view
* Join session functionality

**7. Attendance Result Screen (attendance\_result\_screen.dart)**

**Chức năng:**

* Hiển thị kết quả điểm danh
* Attendance confirmation
* Success/error feedback
* Return to dashboard

**Services:**

* **api-service.dart:** HTTP client với authentication, RESTful API integration
* **auth-service.dart:** JWT token management, user session handling
* **camera-service.dart:** Camera initialization, permission handling, image capture

**Models:**

* **User:** Role-based user model (student, teacher, admin)
* **AttendanceSession:** Session management
* **Attendance:** Attendance records
* **Schedule:** Class scheduling
* **Subject:** Subject information

**Widgets:**

* **CameraPreviewWidget:** Reusable camera component
* **AttendanceCard:** Attendance information display
* **CustomButton, CustomTextField:** Styled UI components
* **LoadingDialog:** Loading indicators

**�� BACKEND - NODE.JS (backendjs/)**

**Cấu trúc dự án:**

* **Framework:** Express.js 4.18.2
* **Database:** MySQL với connection pool
* **Authentication:** JWT + bcrypt
* **File Upload:** Multer với validation
* **Validation:** Joi schema validation
* **Documentation:** Swagger API docs
* **Security:** Helmet.js, CORS, Rate limiting

**Các routes chính và chức năng:**

**1. Auth Routes (/auth)**

**Chức năng:**

* **POST /auth/login:** Đăng nhập với username/password
* **POST /auth/register:** Đăng ký user mới
* **GET /auth/profile:** Lấy thông tin profile
* **POST /auth/logout:** Đăng xuất
* **Password hashing** với bcrypt
* **JWT token generation** và validation

**2. Face Routes (/face)**

**Chức năng:**

* **POST /face/upload-video:** Upload video để tạo dataset
* **POST /face/train:** Training face recognition model
* **POST /face/recognize:** Nhận diện khuôn mặt
* **GET /face/dataset-stats:** Thống kê dataset
* **Face detection** với Haar Cascade
* **Model training** với LBPH algorithm

**3. Class Routes (/classes)**

**Chức năng:**

* **GET /classes:** Lấy danh sách lớp học
* **POST /classes:** Tạo lớp mới
* **PUT /classes/:id:** Cập nhật thông tin lớp
* **DELETE /classes/:id:** Xóa lớp
* **Student management** trong lớp
* **Class scheduling** và time slots

**4. Subject Routes (/subjects)**

**Chức năng:**

* **Subject CRUD operations**
* **Teacher assignment** cho môn học
* **Schedule management** cho subjects
* **Prerequisite handling**

**5. Attendance Routes (/attendance)**

**Chức năng:**

* **POST /attendance/session:** Tạo session điểm danh
* **POST /attendance/check:** Điểm danh với face recognition
* **GET /attendance/reports:** Báo cáo điểm danh
* **Real-time attendance tracking**
* **Session management**
* **Attendance statistics**

**6. Admin Routes (/admin)**

**Chức năng:**

* **GET /admin/dashboard:** Dashboard admin
* **GET /admin/users:** Quản lý users
* **GET /admin/statistics:** Thống kê hệ thống
* **System monitoring**
* **User role management**
* **System configuration**

**Services:**

**FaceRecognitionService:**

* **Tích hợp Python scripts** qua child\_process
* **01\_face\_dataset.py:** Tạo dataset từ video với OpenCV
* **02\_face\_training.py:** Training model LBPH
* **03\_face\_recognition.py:** Nhận diện khuôn mặt
* **Dataset management** và statistics
* **Model validation** và quality assessment

**Database Service:**

* **MySQL connection pool** với connection management
* **Prepared statements** cho security
* **Transaction support** cho data integrity
* **Connection error handling**

**Security Features:**

* **Helmet.js** cho security headers
* **Rate limiting** (100 requests/15min)
* **CORS configuration** với origin validation
* **JWT authentication** với token validation
* **Password hashing** với bcrypt
* **Input validation** với Joi schemas
* **SQL injection prevention**

**Python Integration:**

* **OpenCV** cho face detection và processing
* **Haar Cascade classifier** cho face detection
* **LBPH (Local Binary Pattern Histogram)** cho face recognition
* **Video processing** để tạo dataset
* **Image preprocessing** và enhancement
* **Model training** và validation

**🔄 KẾT NỐI VÀ DATA FLOW**

**API Endpoints:**

* **Base URL:** http://localhost:8000 (web), http://10.0.2.2:8000 (Flutter)
* **Authentication:** Bearer token trong headers
* **File upload:** Multipart form data với validation
* **Response format:** JSON với success/error handling

**Data Flow:**

1. **Frontend (React/Flutter)** gửi request qua service layer
2. **Backend (Node.js)** xử lý với Express routes
3. **Python scripts** xử lý face recognition AI
4. **MySQL Database** lưu trữ thông tin
5. **Response** trả về frontend với proper error handling

**Cross-platform Integration:**

* **Shared API endpoints** cho cả web và mobile
* **Consistent data models** và validation
* **Unified authentication** system
* **Real-time synchronization** giữa platforms

**🎯 KEY FEATURES TỔNG THỂ**

**1. Face Recognition System:**

* **Real-time face detection** với OpenCV
* **Multi-face support** trong một frame
* **Quality assessment** cho captured images
* **Model training** và continuous improvement
* **Dataset management** và validation

**2. Multi-role Authentication:**

* **Student:** Điểm danh bằng khuôn mặt
* **Teacher:** Quản lý session và attendance
* **Admin:** Quản lý toàn bộ hệ thống
* **Role-based access control**
* **Session management**

**3. Real-time Features:**

* **Live attendance tracking**
* **Session status updates**
* **Real-time notifications** (có thể implement)
* **Live camera preview**
* **Instant feedback**

**4. Comprehensive Management:**

* **User management** với role assignment
* **Class management** và scheduling
* **Subject management** và teacher assignment
* **Attendance tracking** và reporting
* **System monitoring** và statistics

**5. Mobile-First Design:**

* **Responsive web interface**
* **Native mobile experience** với Flutter
* **Offline support** và data caching
* **Cross-platform compatibility**
* **Touch-optimized UI**

**6. Security & Performance:**

* **JWT-based authentication**
* **Rate limiting** và DDoS protection
* **Input validation** và sanitization
* **Database optimization** với connection pooling
* **File upload security** với validation

Đây là một hệ thống hoàn chỉnh và chuyên nghiệp, tích hợp AI face recognition để quản lý điểm danh thông minh, hỗ trợ cả web và mobile platforms với kiến trúc scalable và secure.

đưa tôi các api cho phần backend nodejs

Dựa trên việc phân tích code backend Node.js, tôi sẽ cung cấp cho bạn toàn bộ các API endpoints:

**🚀 BACKEND NODE.JS API ENDPOINTS**

**📍 BASE URL: http://localhost:8000**

**�� AUTHENTICATION ROUTES (/auth)**

**1. Đăng nhập**

**http**

**Apply to auth.js**

POST /auth/login

Content-Type: application/json

{

  "username": "string",

  "password": "string"

}

**Response:**

json

Apply to auth.js

{

  "message": "Login successful",

  "token": "jwt\_token\_here",

  "user": {

    "id": 1,

    "username": "user123",

    "full\_name": "Full Name",

    "email": "user@example.com",

    "role": "student",

    "student\_id": "SV001",

    "class\_name": "Class A",

    "is\_active": true,

    "face\_trained": true,

    "created\_at": "2024-01-01T00:00:00Z"

  }

}

**2. Đăng ký**

**http**

**Apply to auth.js**

POST /auth/register

Content-Type: application/json

{

  "username": "string",

  "password": "string",

  "full\_name": "string",

  "email": "string",

  "role": "student|teacher|admin",

  "student\_id": "string", // required if role is student

  "class\_name": "string"  // required if role is student

}

**3. Lấy thông tin profile**

**http**

**Apply to auth.js**

GET /auth/profile

Authorization: Bearer {token}

**4. Đăng xuất**

**http**

**Apply to auth.js**

POST /auth/logout

Authorization: Bearer {token}

**�� FACE RECOGNITION ROUTES (/face)**

**1. Upload video để tạo dataset**

**http**

**Apply to auth.js**

POST /face/upload-video

Authorization: Bearer {token}

Content-Type: multipart/form-data

{

  "video": "video\_file",

  "userId": "number"

}

**Response:**

json

Apply to auth.js

{

  "message": "Video uploaded, dataset created and model trained successfully",

  "dataset\_result": "dataset\_creation\_result",

  "train\_result": "training\_result"

}

**2. Training face recognition model**

**http**

**Apply to auth.js**

POST /face/train

Authorization: Bearer {token}

**3. Nhận diện khuôn mặt**

**http**

**Apply to auth.js**

POST /face/recognize

Authorization: Bearer {token}

Content-Type: multipart/form-data

{

  "image": "image\_file"

}

**4. Thống kê dataset**

**http**

**Apply to auth.js**

GET /face/dataset-stats

Authorization: Bearer {token}

**Response:**

json

Apply to auth.js

{

  "User.1": 50,

  "User.2": 45,

  "User.3": 52

}

**�� CLASS MANAGEMENT ROUTES (/classes)**

**1. Lấy danh sách lớp học**

**http**

**Apply to auth.js**

GET /classes

Authorization: Bearer {token}

**Query Parameters:**

* page: number (pagination)
* limit: number (items per page)
* search: string (search by class name)

**2. Tạo lớp mới**

**http**

**Apply to auth.js**

POST /classes

Authorization: Bearer {token}

Content-Type: application/json

{

  "name": "string",

  "description": "string",

  "capacity": "number",

  "teacher\_id": "number"

}

**3. Cập nhật thông tin lớp**

**http**

**Apply to auth.js**

PUT /classes/:id

Authorization: Bearer {token}

Content-Type: application/json

{

  "name": "string",

  "description": "string",

  "capacity": "number",

  "teacher\_id": "number"

}

**4. Xóa lớp**

**http**

**Apply to auth.js**

DELETE /classes/:id

Authorization: Bearer {token}

**5. Lấy thông tin chi tiết lớp**

**http**

**Apply to auth.js**

GET /classes/:id

Authorization: Bearer {token}

**6. Thêm sinh viên vào lớp**

**http**

**Apply to auth.js**

POST /classes/:id/students

Authorization: Bearer {token}

Content-Type: application/json

{

  "student\_ids": ["number"]

}

**7. Xóa sinh viên khỏi lớp**

**http**

**Apply to auth.js**

DELETE /classes/:id/students/:student\_id

Authorization: Bearer {token}

**📚 SUBJECT MANAGEMENT ROUTES (/subjects)**

**1. Lấy danh sách môn học**

**http**

**Apply to auth.js**

GET /subjects

Authorization: Bearer {token}

**2. Tạo môn học mới**

**http**

**Apply to auth.js**

POST /subjects

Authorization: Bearer {token}

Content-Type: application/json

{

  "name": "string",

  "code": "string",

  "description": "string",

  "credits": "number"

}

**3. Cập nhật môn học**

**http**

**Apply to auth.js**

PUT /subjects/:id

Authorization: Bearer {token}

Content-Type: application/json

{

  "name": "string",

  "code": "string",

  "description": "string",

  "credits": "number"

}

**4. Xóa môn học**

**http**

**Apply to auth.js**

DELETE /subjects/:id

Authorization: Bearer {token}

**5. Lấy thông tin chi tiết môn học**

**http**

**Apply to auth.js**

GET /subjects/:id

Authorization: Bearer {token}

**📅 ATTENDANCE ROUTES (/attendance)**

**1. Tạo session điểm danh**

**http**

**Apply to auth.js**

POST /attendance/session

Authorization: Bearer {token}

Content-Type: application/json

{

  "name": "string",

  "class\_id": "number",

  "subject\_id": "number",

  "start\_time": "datetime",

  "end\_time": "datetime",

  "location": "string"

}

**2. Điểm danh với face recognition**

**http**

**Apply to auth.js**

POST /attendance/check

Authorization: Bearer {token}

Content-Type: multipart/form-data

{

  "image": "image\_file",

  "session\_id": "number",

  "user\_id": "number"

}

**3. Lấy báo cáo điểm danh**

**http**

**Apply to auth.js**

GET /attendance/reports

Authorization: Bearer {token}

**Query Parameters:**

* session\_id: number
* class\_id: number
* date\_from: string (YYYY-MM-DD)
* date\_to: string (YYYY-MM-DD)
* user\_id: number

**4. Lấy danh sách session**

**http**

**Apply to auth.js**

GET /attendance/sessions

Authorization: Bearer {token}

**5. Lấy chi tiết session**

**http**

**Apply to auth.js**

GET /attendance/sessions/:id

Authorization: Bearer {token}

**6. Đóng session điểm danh**

**http**

**Apply to auth.js**

PUT /attendance/sessions/:id/close

Authorization: Bearer {token}

**7. Lấy lịch sử điểm danh**

**http**

**Apply to auth.js**

GET /attendance/history

Authorization: Bearer {token}

**Query Parameters:**

* user\_id: number
* date\_from: string
* date\_to: string
* status: string (present|absent|late)

**👨‍�� ADMIN ROUTES (/admin)**

**1. Dashboard admin**

**http**

**Apply to auth.js**

GET /admin/dashboard

Authorization: Bearer {token}

**Response:**

json

Apply to auth.js

{

  "total\_users": 150,

  "total\_students": 120,

  "total\_teachers": 25,

  "total\_sessions": 500,

  "total\_attendances": 15000

}

**2. Quản lý users**

**http**

**Apply to auth.js**

GET /admin/users

Authorization: Bearer {token}

**Query Parameters:**

* role: string (student|teacher|admin)
* status: string (active|inactive)
* page: number
* limit: number

**3. Thống kê hệ thống**

**http**

**Apply to auth.js**

GET /admin/statistics

Authorization: Bearer {token}

**4. Retrain face recognition model**

**http**

**Apply to auth.js**

POST /admin/retrain-model

Authorization: Bearer {token}

**5. Quản lý user roles**

**http**

**Apply to auth.js**

PUT /admin/users/:id/role

Authorization: Bearer {token}

Content-Type: application/json

{

  "role": "student|teacher|admin"

}

**6. Kích hoạt/vô hiệu hóa user**

**http**

**Apply to auth.js**

PUT /admin/users/:id/status

Authorization: Bearer {token}

Content-Type: application/json

{

  "is\_active": "boolean"

}

**👨‍🏫 TEACHER ROUTES (/teacher)**

**1. Tạo session điểm danh**

**http**

**Apply to auth.js**

POST /teacher/create-session

Authorization: Bearer {token}

Content-Type: application/json

{

  "name": "string",

  "class\_id": "number",

  "subject\_id": "number",

  "start\_time": "datetime",

  "end\_time": "datetime"

}

**2. Lấy danh sách session của giáo viên**

**http**

**Apply to auth.js**

GET /teacher/sessions

Authorization: Bearer {token}

**3. Đóng session**

**http**

**Apply to auth.js**

PUT /teacher/session/:id/close

Authorization: Bearer {token}

**4. Lấy attendance của session**

**http**

**Apply to auth.js**

GET /teacher/attendance/:session\_id

Authorization: Bearer {token}

**👨‍🎓 STUDENT ROUTES (/student)**

**1. Đăng ký khuôn mặt**

**http**

**Apply to auth.js**

POST /student/register-face

Authorization: Bearer {token}

Content-Type: multipart/form-data

{

  "video": "video\_file"

}

**2. Điểm danh**

**http**

**Apply to auth.js**

POST /student/attendance

Authorization: Bearer {token}

Content-Type: multipart/form-data

{

  "image": "image\_file",

  "session\_id": "number"

}

**3. Lấy lịch sử điểm danh**

**http**

**Apply to auth.js**

GET /student/attendance-history

Authorization: Bearer {token}

**4. Lấy session đang hoạt động**

**http**

**Apply to auth.js**

GET /student/active-sessions

Authorization: Bearer {token}

**🔧 UTILITY ROUTES**

**1. Kiểm tra kết nối**

**http**

**Apply to auth.js**

GET /

**Response:**

json

Apply to auth.js

{

  "message": "Face Attendance API Server",

  "version": "1.0.0",

  "status": "running"

}

**2. Swagger Documentation**

**http**

**Apply to auth.js**

GET /docs

**📁 FILE UPLOAD ENDPOINTS**

**1. Upload ảnh**

**http**

**Apply to auth.js**

POST /uploads/images

Authorization: Bearer {token}

Content-Type: multipart/form-data

{

  "image": "image\_file"

}

**2. Upload video**

**http**

**Apply to auth.js**

POST /uploads/videos

Authorization: Bearer {token}

Content-Type: multipart/form-data

{

  "video": "video\_file"

}

**⚠️ ERROR RESPONSES**

**Validation Error (400)**

**json**

**Apply to auth.js**

{

  "error": "Validation error message",

  "details": [

    {

      "field": "username",

      "message": "Username is required"

    }

  ]

}

**Authentication Error (401)**

**json**

**Apply to auth.js**

{

  "error": "Invalid credentials"

}

**Authorization Error (403)**

**json**

**Apply to auth.js**

{

  "error": "Access denied"

}

**Not Found Error (404)**

**json**

**Apply to auth.js**

{

  "error": "Route not found"

}

**Server Error (500)**

**json**

**Apply to auth.js**

{

  "error": "Something went wrong!",

  "message": "Internal server error"

}

**🔒 AUTHENTICATION HEADERS**

Tất cả các API endpoints (trừ /auth/login, /auth/register, /) đều yêu cầu:

http

Apply to auth.js

Authorization: Bearer {jwt\_token}

Content-Type: application/json

**�� QUERY PARAMETERS CHUNG**

**Pagination:**

* page: number (default: 1)
* limit: number (default: 10, max: 100)

**Search & Filter:**

* search: string (search by name/email)
* status: string (filter by status)
* role: string (filter by role)
* date\_from: string (YYYY-MM-DD)
* date\_to: string (YYYY-MM-DD)

**Sorting:**

* sort\_by: string (field name)
* sort\_order: string (asc|desc)

**🚀 RATE LIMITING**

* **Limit:** 100 requests per 15 minutes per IP
* **Headers:**
* X-RateLimit-Limit
* X-RateLimit-Remaining
* X-RateLimit-Reset

Đây là toàn bộ API endpoints của backend Node.js, hỗ trợ đầy đủ các chức năng cho hệ thống Face Recognition Attendance Management.