

ProGymHub وثائق مشروع

نظام إدارة النوادي الرياضية المتكامل

جدول المحتويات

- [نظرة عامة](#)
 - [متطلبات النظام](#)
 - [هيكل المستخدمين](#)
 - [الميزات الرئيسية](#)
 - [معمارية النظام](#)
 - [قاعدة البيانات](#)
 - [واجهات المستخدم](#)
 - [والخدمات API](#)
 - [الأمان والحماية](#)
 - [التثبيت والإعداد](#)
 - [دليل المطور](#)
 - [دليل المستخدم](#)
 - [استكشاف الأخطاء](#)
 - [التحديثات المستقبلية](#)
-

نظرة عامة

ProGymHub ما هو

هو نظام متكامل لإدارة النوادي الرياضية والصالات الرياضية، مصمم لربط النوادي والمدربين والأعضاء ProGymHub في منصة واحدة شاملة. يوفر النظام حلاً متقدماً لإدارة الاشتراكات، وتنظيم خطط التدريب، ومتابعة تقدم الأعضاء.

الهدف من المشروع

- تبسيط إدارة النوادي الرياضية
- تحسين التواصل بين النوادي والمدربين والأعضاء
- توفير نظام شامل لمتابعة التقدم والأداء
- أتمتة العمليات الإدارية والمالية

الفوائد الرئيسية

- **للنوادي:** إدارة مبسطة للأعضاء والاشتراكات والمدربين
- **للمدربين:** أدوات متقدمة لإنشاء خطط التدريب ومتابعة العملاء
- **للأعضاء:** تجربة مستخدم محسنة لمتابعة التقدم والاشتراكات
- **للمشرفين:** تحكم كامل في النظام ومراقبة شاملة

متطلبات النظام

متطلبات الخادم

- **الإصدار 8.0 أو أحدث: PHP**
- **الإصدار 9.0 أو أحدث: Laravel**
- **MySQL 8.0+ أو PostgreSQL 13+ قاعدة البيانات**
- **Apache 2.4+ أو Nginx 1.18+ خادم الويب**
- **موصى بها 4GB RAM كحد أدنى، الذاكرة: 2**
- **كحد أدنى 10GB مساحة التخزين: 10**

متطلبات العميل

- **المتصفحات المدعومة:** Chrome 90+، Firefox 88+، Safari 14+، Edge 90+
- **كحد أدنى 768x1024 الدقة:**
- **الاتصال:** اتصال إنترنت مستقر

التبعيات والمكتبات

```
json
{
  "php": "^8.0",
  "laravel/framework": "^9.0",
  "laravel/tinker": "^2.7",
  "laravel/ui": "^4.0",
  "intervention/image": "^2.7",
  "barryvdh/laravel-dompdf": "^2.0",
  "maatwebsite/excel": "^3.1"
}
```

هيكل المستخدمين

1. المشرف (Admin)

الصلاحيات:

- إدارة جميع النوادي في النظام
- مراقبة الإحصائيات العامة
- إدارة إعدادات النظام
- إدارة المستخدمين على جميع المستويات
- الوصول إلى التقارير الشاملة

الواجبات:

- لوحة تحكم شاملة
- إدارة النوادي
- تقارير مفصلة
- إعدادات النظام

2. النادي (Club)

الصلاحيات:

- إدارة المدربين التابعين للنادي
- إنشاء وإدارة خطط الاشتراك
- إدارة الأعضاء والعضويات
- مراقبة الإيرادات والمدفوعات
- إدارة مرافق النادي

الواجبات:

- لوحة تحكم النادي
- إدارة المدربين
- إدارة الاشتراكات
- إدارة الأعضاء
- التقارير المالية

3. المدرب (Coach)

الصلاحيات:

- إدارة قائمة العملاء المخصصين

- إنشاء وتعديل خطط التدريب
- متابعة تقدم العملاء
- تسجيل حضور الجلسات
- التواصل مع العملاء

الواجهات:

- لوحة تحكم المدرب
- إدارة العملاء
- خطط التدريب
- متابعة التقدم

4. المستخدم/العضو (User)

الصلاحيات:

- عرض معلومات الاشتراك
- متابعة خطة التدريب الشخصية
- تسجيل التمارين والتقدم
- التواصل مع المدرب
- عرض الإحصائيات الشخصية

الواجهات:

- الملف الشخصي
- خطة التدريب
- متابعة التقدم
- الاشتراكات

الميزات الرئيسية

1. نظام الاشتراكات المتقدم

أنواع الاشتراكات

- اشتراك شهري: فترة 30 يوم
- اشتراك ربع سنوي: فترة 90 يوم
- اشتراك نصف سنوي: فترة 180 يوم

- **اشترك سنوي:** فترة 365 يوم
- **اشترك مخصص:** فترة قابلة للتخصيص

إدارة المدفوعات

- تتبع حالة الدفع (مدفوع، معلق، متأخر)
- طرق دفع متعددة (نقد، بطاقة ائتمان، تحويل بنكي)
- إيصالات آلية
- تذكير بالدفعات المستحقة

التجديد التلقائي

- إعدادات التجديد التلقائي
- إشعارات ما قبل انتهاء الاشتراك
- خيارات التجديد المرنة

2. إدارة النوادي الشاملة

معلومات النادي

php

// مثال على بنية بيانات النادي

```
[  
    'name' => 'اسم النادي',  
    'description' => 'وصف النادي',  
    'address' => 'العنوان الكامل',  
    'phone' => 'رقم الهاتف',  
    'email' => 'البريد الإلكتروني',  
    'working_hours' => [  
        'opening_time' => '06:00',  
        'closing_time' => '23:00'  
    ],  
    'facilities' => [  
        'parking' => true,  
        'wifi' => true,  
        'lockers' => true,  
        'sauna' => true,  
        'swimming_pool' => false  
    ]  
]
```

إدارة المرافق

- مواقف السيارات
- خدمة الواي فاي
- الخزائن الشخصية
- الساونا
- المسبح
- صالة الكارديو
- صالة الأوزان

3. نظام المدربين المتطور

ملف المدرب

- المعلومات الشخصية والمهنية
- الشهادات والمؤهلات
- سنوات الخبرة
- التخصصات (كمال أجسام، لياقة بدنية، يوغا، إلخ)
- التقييمات والمراجعات

إدارة العملاء

- تخصيص العملاء للمدربين
- متابعة تقدم كل عميل
- إنشاء خطط تدريب مخصصة
- تسجيل الملاحظات والتوصيات

4. تجربة المستخدم المحسنة

التتبع الشخصي

- (BMI) مؤشر كتلة الجسم
- نسبة الدهون في الجسم
- الوزن والطول
- القياسات الجسدية
- مستوى اللياقة البدنية

خطط التغذية

- نصائح غذائية
 - جداول الوجبات
 - حساب السعرات الحرارية
 - متابعة استهلاك الماء
-

معمارية النظام

نمط MVC (Model-View-Controller)

Models (النماذج)

php

// نماذج رئيسية في النظام

- User.php (المستخدمين)
- Club.php (النوادي)
- Coach.php (المدربين)
- Subscription.php (الاشتراكات)
- WorkoutPlan.php (خطط التدريب)
- Payment.php (المدفوعات)

Views (الواجهات)

resources/views/

— admin/	# واجهات المشرف
— club/	# واجهات النادي
— coach/	# واجهات المدرب
— user/	# واجهات المستخدم
— auth/	# واجهات المصادقة
— layouts/	# القوالب الأساسية

Controllers (المتحكمات)

php

app/Http/Controllers/

└─ AdminController.php

└─ ClubController.php

└─ CoachController.php

└─ UserController.php

└─ SubscriptionController.php

└─ PaymentController.php

الخدمات والمساعدات

php

app/Services/

└─ SubscriptionService.php

└─ PaymentService.php

└─ NotificationService.php

└─ EmailService.php

قاعدة البيانات

الجدول الرئيسية

جدول المستخدمين (users)

sql

```
CREATE TABLE users (  
  id BIGINT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE NOT NULL,  
  phone VARCHAR(20),  
  password VARCHAR(255) NOT NULL,  
  role ENUM('admin', 'club', 'coach', 'user') NOT NULL,  
  email_verified_at TIMESTAMP NULL,  
  profile_image VARCHAR(255),  
  date_of_birth DATE,  
  gender ENUM('male', 'female'),  
  height DECIMAL(5,2),  
  weight DECIMAL(5,2),  
  fitness_level ENUM('beginner', 'intermediate', 'advanced'),  
  medical_conditions TEXT,  
  emergency_contact VARCHAR(255),  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  deleted_at TIMESTAMP NULL  
);
```

جدول النوادي (clubs)

sql

```
CREATE TABLE clubs (  
    id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    description TEXT,  
    address TEXT NOT NULL,  
    phone VARCHAR(20),  
    email VARCHAR(255),  
    opening_time TIME,  
    closing_time TIME,  
    facilities JSON,  
    logo VARCHAR(255),  
    owner_id BIGINT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
    deleted_at TIMESTAMP NULL,  
    FOREIGN KEY (owner_id) REFERENCES users(id)  
);
```

جدول الاشتراكات (subscriptions)

sql

```
CREATE TABLE subscriptions (  
    id BIGINT PRIMARY KEY AUTO_INCREMENT,  
    user_id BIGINT NOT NULL,  
    club_id BIGINT NOT NULL,  
    plan_id BIGINT NOT NULL,  
    start_date DATE NOT NULL,  
    end_date DATE NOT NULL,  
    status ENUM('active', 'expired', 'cancelled') DEFAULT 'active',  
    amount DECIMAL(10,2) NOT NULL,  
    payment_status ENUM('paid', 'pending', 'failed') DEFAULT 'pending',  
    payment_method VARCHAR(50),  
    notes TEXT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
    FOREIGN KEY (user_id) REFERENCES users(id),  
    FOREIGN KEY (club_id) REFERENCES clubs(id),  
    FOREIGN KEY (plan_id) REFERENCES subscription_plans(id)  
);
```

العلاقات بين الجداول

- Users (1:N) Subscriptions
- Clubs (1:N) Subscriptions
- Clubs (1:N) Coaches
- Coaches (1:N) User Assignments
- Users (1:N) Workout Sessions
- Users (1:N) Progress Records

واجهات المستخدم

تصميم متجاوب

النظام مصمم ليكون متجاوباً مع جميع أحجام الشاشات

- سطح المكتب: تخطيط كامل مع شريط جانبي
- الأجهزة اللوحية: تخطيط مبسط مع قوائم قابلة للطي
- الهواتف الذكية: واجهة محسنة للمس

نظام الألوان

```
css

:root {
  --primary-color: #3498db;
  --secondary-color: #2ecc71;
  --danger-color: #e74c3c;
  --warning-color: #f39c12;
  --dark-color: #2c3e50;
  --light-color: #ecf0f1;
}
```

المكونات القابلة لإعادة الاستخدام

- بطاقات المعلومات
- جداول البيانات
- نماذج الإدخال
- أزرار الإجراءات
- الإشعارات والتنبيهات

والخدمات API

RESTful API Endpoints

مصادقة المستخدم

http

POST /api/auth/login

POST /api/auth/register

POST /api/auth/logout

POST /api/auth/refresh

GET /api/auth/user

إدارة النوادي

http

GET /api/clubs

POST /api/clubs

GET /api/clubs/{id}

PUT /api/clubs/{id}

DELETE /api/clubs/{id}

إدارة الاشتراكات

http

GET /api/subscriptions

POST /api/subscriptions

GET /api/subscriptions/{id}

PUT /api/subscriptions/{id}

DELETE /api/subscriptions/{id}

خدمات البريد الإلكتروني

php

// إعدادات البريد الإلكتروني //

MAIL_MAILER=smtp

MAIL_HOST=smtp.gmail.com

MAIL_PORT=587

MAIL_USERNAME=your-email@gmail.com

MAIL_PASSWORD=your-password

MAIL_ENCRYPTION=tls

خدمة الإشعارات

- إشعارات الويب
 - الرسائل النصية
 - البريد الإلكتروني
 - الإشعارات المضمنة في النظام
-

الأمان والحماية

تشفير البيانات

- bcrypt تشفير كلمات المرور باستخدام
- تشفير البيانات الحساسة
- حماية جلسات المستخدمين

التحقق من الهوية

- التحقق من البريد الإلكتروني
- المصادقة الثنائية (اختيارية)
- إعادة تعيين كلمة المرور الآمنة

حماية من الهجمات

- CSRF حماية من
- SQL Injection حماية من
- XSS حماية من
- (Rate Limiting) تحديد معدل الطلبات

إدارة الصلاحيات

php

// مثال على التحقق من الصلاحيات

```
public function authorize()  
{  
    return $this->user()->can('manage-subscriptions');  
}
```

التثبيت والإعداد

متطلبات ما قبل التثبيت

1. تثبيت PHP 8.0+
2. تثبيت Composer
3. تثبيت Node.js و npm
4. إعداد قاعدة البيانات

خطوات التثبيت

bash

1. استنساخ المشروع

```
git clone https://github.com/your-repo/progymhub.git  
cd progymhub
```

2. تثبيت التبعيات

```
composer install  
npm install
```

3. إعداد البيئة

```
cp .env.example .env  
php artisan key:generate
```

4. إعداد قاعدة البيانات

```
php artisan migrate  
php artisan db:seed
```

5. بناء الأصول

```
npm run build
```

6. تشغيل الخادم

```
php artisan serve
```

إعدادات البيئة (.env)

env

APP_NAME=ProGymHub

APP_ENV=production

APP_KEY=base64:your-app-key

APP_DEBUG=false

APP_URL=https://your-domain.com

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=progyhub

DB_USERNAME=your-username

DB_PASSWORD=your-password

دليل المطور

هيكل المشروع

```
progyhub/  
├─ app/  
│   ├── Http/  
│   │   ├── Controllers/  
│   │   ├── Middleware/  
│   │   └── Requests/  
│   ├── Models/  
│   ├── Services/  
│   └── Providers/  
├─ database/  
│   ├── migrations/  
│   ├── seeders/  
│   └── factories/  
├─ resources/  
│   ├── views/  
│   ├── js/  
│   └── css/  
├─ routes/  
│   ├── web.php  
│   ├── api.php  
│   └── channels.php  
└─ config/
```

قواعد التطوير

1. للمتغيرات camelCase ، للكلاسات PascalCase **تسمية الملفات**: استخدام
2. **التوثيق**: توثيق جميع الدوال والكلاسات
3. **الاختبارات**: كتابة اختبارات لجميع الميزات الجديدة
4. **Git**: واضحة ومفصلة commit استخدام رسائل

إنشاء ميزة جديدة

bash

إنشاء متحكم جديد

```
php artisan make:controller NewFeatureController
```

إنشاء نموذج جديد

```
php artisan make:model NewFeature -m
```

إنشاء طلب تحقق

```
php artisan make:request NewFeatureRequest
```

إنشاء اختبار

```
php artisan make:test NewFeatureTest
```

دليل المستخدم

للمشرفين

1. تسجيل الدخول: استخدام بيانات المشرف للوصول للوحة التحكم.
2. إدارة النوادي: إضافة وتعديل وحذف النوادي.
3. مراقبة النشاط: متابعة الإحصائيات والتقارير.
4. إدارة المستخدمين: التحكم في صلاحيات المستخدمين.

للكوادر

1. إعداد الملف التعريفي: تحديث معلومات النادي والمرافق.
2. إدارة المدربين: إضافة مدربين جدد وتحديث معلوماتهم.
3. إدارة الاشتراكات: إنشاء خطط اشتراك وإدارة المدفوعات.
4. متابعة الأعضاء: عرض قائمة الأعضاء وحالة اشتراكاتهم.

للمدربين

1. إدارة العملاء: عرض قائمة العملاء المخصصين.
2. خطط التدريب: إنشاء وتعديل خطط التدريب المخصصة.
3. متابعة التقدم: تسجيل تقدم العملاء وإضافة ملاحظات.
4. التواصل: إرسال رسائل وإشعارات للعملاء.

للأعضاء

1. الملف الشخصي: تحديث المعلومات الشخصية والصحية.

- الاشتراكات: عرض حالة الاشتراك وتاريخ التجديد.
- خطة التدريب: متابعة التمارين المخصصة.
- التقدم: تسجيل الأوزان والقياسات ومتابعة التحسن.

استكشاف الأخطاء

الأخطاء الشائعة وحلولها

خطأ قاعدة البيانات

```
bash

# التحقق من الاتصال
php artisan tinker
>>> DB::connection()->getPdo();
```

مشاكل الصلاحيات

```
bash

# إعداد صلاحيات الملفات
chmod -R 755 storage/
chmod -R 755 bootstrap/cache/
```

مشاكل البريد الإلكتروني

```
bash

# اختبار إعدادات البريد
php artisan tinker
>>> Mail::raw('Test email', function($msg) { $msg->to('test@example.com'); });
```

رسائل الخطأ الشائعة

الخطأ	السبب	الحل
500 Internal Server Error	خطأ في التكوين	فحص سجلات الأخطاء
403 Forbidden	مشكلة في الصلاحيات	إعادة تعيين صلاحيات الملفات
Connection refused	مشكلة في قاعدة البيانات	التحقق من إعدادات الاتصال

أدوات التشخيص

bash

عرض سجلات الأخطاء

`tail -f storage/logs/laravel.log`

فحص حالة النظام

`php artisan about`

مسح الذاكرة المؤقتة

`php artisan cache:clear`

`php artisan config:clear`

`php artisan view:clear`

التحديثات المستقبلية

الإصدار 2.0

- **iOS تطبيق الجوال:** تطوير تطبيقات للأندرويد و
- **التكامل مع الأجهزة:** ربط أجهزة قياس اللياقة البدنية
- **الذكاء الاصطناعي:** توصيات تدريب ذكية
- **الواقع المعزز:** دروس تدريب تفاعلية

الإصدار 2.5

- **التحليلات المتقدمة:** تقارير وإحصائيات معمقة
- **التكامل المالي:** ربط أنظمة الدفع الإلكتروني
- **المجتمع الافتراضي:** منتديات ومجموعات نقاش
- **التدريب الجماعي:** إدارة الحصص الجماعية

خارطة طريق التطوير

gantt

title خارطة طريق ProGymHub

dateFormat YYYY-MM-DD

section 1.0 الإصدار

النسخة الأساسية :done, v1, 2024-01-01, 2024-06-30

section 1.5 الإصدار

تحسينات الأداء :active, v1-5, 2024-07-01, 2024-12-31

section 2.0 الإصدار

تطبيق الجوال :v2, 2025-01-01, 2025-06-30

section 2.5 الإصدار

النكاه الاصطناعي :v2-5, 2025-07-01, 2025-12-31

الدعم والتواصل

فريق التطوير

- مطور رئيسي: [اسم المطور]
- مصمم واجهات: [اسم المصمم]
- محلل نظم: [اسم المحلل]
- مختبر جودة: [اسم المختبر]

قنوات الدعم

- البريد الإلكتروني: support@progyhub.com
- الهاتف: +962-6-1234567
- الدردشة المباشرة: متوفرة في النظام
- التذاكر: نظام تذاكر مدمج

المساهمة في المشروع

نرحب بالمساهمات من المطورين. يرجى اتباع دليل المساهمة والتأكد من

- كتابة كود نظيف ومفهوم
- إضافة اختبارات للميزات الجديدة
- توثيق التغييرات بوضوح
- اتباع معايير الترميز المعتمدة

الملاحق

ملحق أ: أكواد الاستجابة

```
php
// المستخدمة HTTP أكواد حالة
200 => 'نجاح الطلب',
201 => 'تم الإنشاء بنجاح',
400 => 'طلب غير صحيح',
401 => 'غير مسموح',
403 => 'محظور',
404 => 'غير موجود',
422 => 'بيانات غير صحيحة',
500 => 'خطأ في الخادم'
```

ملحق ب: قائمة التحقق للإطلاق

- ☐ اختبار جميع الوظائف
- ☐ تحسين الأداء
- ☐ مراجعة الأمان
- ☐ إعداد النسخ الاحتياطية
- ☐ تدريب المستخدمين
- ☐ إعداد المراقبة
- ☐ توثيق النظام
- ☐ خطة الطوارئ

ملحق ج: قاموس المصطلحات

المصطلح	التعريف
API	واجهة برمجة التطبيقات
MVC	Model-View-Controller نمط التصميم
ORM	Object Relational Mapping
CRUD	Create, Read, Update, Delete
JWT	JSON Web Token

تاريخ آخر تحديث: [التاريخ الحالي]

إصدار الوثائق: 1.0

إصدار النظام: 1.0