

حزمة التنصيب الشاملة - منصة سَهول

نظرة عامة

هذا الدليل يحتوي على جميع الأكواد والسكربتات المطلوبة لتنصيب منصة سَهول على سيرفر محلي (HCI).

المحتويات

1. [سكربت التنصيب التلقائي](#)

2. [ملفات الإعداد](#)

3. [سكربتات الصيانة](#)

4. [دليل التشغيل](#)

سكربت التنصيب التلقائي

1. سكربت التنصيب الرئيسي

احفظ هذا الملف باسم `install.sh` :

```
#!/bin/bash
```

```
#=====
# منصة سهول - سكريبت التنصيب الشامل
# الإصدار: 2.2.0
# التاريخ: 03-11-2025
#=====
```

```
set -e # إيقاف عند أي خطأ
```

```
# الألوان للطباعة
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
BLUE='\033[0;34m'
NC='\033[0m' # No Color
```

```
# دالة الطباعة الملونة
print_info() {
    echo -e "${BLUE}[INFO]${NC} $1"
}
```

```
print_success() {
    echo -e "${GREEN}[SUCCESS]${NC} $1"
}
```

```
print_warning() {
    echo -e "${YELLOW}[WARNING]${NC} $1"
}
```

```
print_error() {
    echo -e "${RED}[ERROR]${NC} $1"
}
```

```
# التحقق من صلاحيات المستخدم
check_root() {
    if [[ $EUID -ne 0 ]]; then
        print_error "root يجب تشغيل هذا السكريبت بصلاحيات"
        print_info "استخدم: sudo bash install.sh"
        exit 1
    fi
}
```

```
# رسالة الترحيب
welcome_message() {
```

```
clear
echo "=====
echo "||
echo "||           منصة سهول - سكربت التنصيب التلقائي           ||"
echo "||                               الإصدار 2.2.0                               ||"
echo "||=====
echo ""
print_info "سيتم تنصيب المكونات التالية:"
echo " - Node.js 22.x"
echo " - MySQL 8.x"
echo " - Nginx"
echo " - Redis"
echo " - PM2"
echo " - منصة سهول"
echo ""
read -p "هل تريد المتابعة؟ (y/n): " -n 1 -r
echo
if [[ ! $REPLY =~ ^[Yy]$ ]]; then
    print_warning "تم إلغاء التنصيب"
    exit 0
fi
}

# تحديث النظام
update_system() {
    print_info "تحديث النظام..."
    apt update && apt upgrade -y
    apt install -y curl wget git build-essential ufw
    print_success "تم تحديث النظام بنجاح"
}

# تثبيت Node.js
install_nodejs() {
    print_info "تثبيت Node.js 22.x..."

    if command -v node &> /dev/null; then
        NODE_VERSION=$(node --version)
        print_warning "Node.js موجود بالفعل $NODE_VERSION"
        read -p "هل تريد إعادة التثبيت؟ (y/n): " -n 1 -r
        echo
        if [[ ! $REPLY =~ ^[Yy]$ ]]; then
            return
        fi
    fi
}
```

```

curl -fsSL https://deb.nodesource.com/setup_22.x | bash -
apt install -y nodejs

# تثبيت pnpm
npm install -g pnpm

print_success "تم تثبيت Node.js $(node --version)"
print_success "تم تثبيت pnpm $(pnpm --version)"
}

# تثبيت MySQL
install_mysql() {
    print_info "تثبيت MySQL 8.x..."

    if command -v mysql &> /dev/null; then
        print_warning "MySQL موجود بالفعل"
        return
    fi

    apt install -y mysql-server
    systemctl start mysql
    systemctl enable mysql

    print_success "بنجاح MySQL تم تثبيت"

    # إعداد قاعدة البيانات
    print_info "...إعداد قاعدة البيانات"

    read -p "أتركها فارغة للتخطي) MySQL root أدخل كلمة مرور": MYSQL_ROOT_PASS

    if [ -n "$MYSQL_ROOT_PASS" ]; then
        mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY '$MYSQL_ROOT_PASS';"
    fi

    read -p "sahool: أدخل كلمة مرور قاعدة بيانات" SAHOOO_DB_PASS

    mysql -e "CREATE DATABASE IF NOT EXISTS sahoool CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;"
    mysql -e "CREATE USER IF NOT EXISTS 'sahool_user'@'localhost'
IDENTIFIED BY '$SAHOOO_DB_PASS';"
    mysql -e "GRANT ALL PRIVILEGES ON sahoool.* TO
'sahool_user'@'localhost';"
    mysql -e "FLUSH PRIVILEGES;"

    # حفظ كلمة المرور للاستخدام لاحقاً

```

```
echo "$SAHOOOL_DB_PASS" > /tmp/sahool_db_pass

print_success "تم إعداد قاعدة البيانات بنجاح"
}

# تثبيت Redis
install_redis() {
    print_info "تثبيت Redis..."

    apt install -y redis-server
    systemctl start redis-server
    systemctl enable redis-server

    print_success "بنجاح Redis تم تثبيت"
}

# تثبيت Nginx
install_nginx() {
    print_info "تثبيت Nginx..."

    apt install -y nginx
    systemctl start nginx
    systemctl enable nginx

    print_success "بنجاح Nginx تم تثبيت"
}

# تثبيت PM2
install_pm2() {
    print_info "تثبيت PM2..."

    npm install -g pm2

    print_success "بنجاح PM2 تم تثبيت"
}

# إعداد Firewall
setup_firewall() {
    print_info "إعداد Firewall..."

    ufw allow 22/tcp    # SSH
    ufw allow 80/tcp    # HTTP
    ufw allow 443/tcp   # HTTPS
    ufw allow 3000/tcp  # Application

    echo "y" | ufw enable
```

```

    print_success "بنجاح Firewall تم إعداد"
}

# تحميل المشروع
download_project() {
    print_info "تحميل منصة سهول..."

    APP_DIR="/opt/sahool-platform"

    if [ -d "$APP_DIR" ]; then
        print_warning "$APP_DIR المشروع موجود بالفعل في"
        read -p "هل تريد حذفه وإعادة التحميل؟ (y/n): " -n 1 -r
        echo
        if [[ $REPLY =~ ^[Yy]$ ]]; then
            rm -rf "$APP_DIR"
        else
            return
        fi
    fi

    mkdir -p "$APP_DIR"
    cd "$APP_DIR"

    git clone https://github.com/kafaat/sahool-platform.git .

    print_success "تم تحميل المشروع بنجاح"
}

# إعداد ملف البيئة
setup_env() {
    print_info "إعداد ملف البيئة..."

    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    # قراءة كلمة مرور قاعدة البيانات
    if [ -f /tmp/sahool_db_pass ]; then
        DB_PASS=$(cat /tmp/sahool_db_pass)
        rm /tmp/sahool_db_pass
    else
        read -p "أدخل كلمة مرور قاعدة البيانات: " DB_PASS
    fi

    # عشوائي JWT Secret توليد
    JWT_SECRET=$(openssl rand -base64 32)

```

```
# إنشاء ملف .env
cat > .env << EOF

# Database
DATABASE_URL=mysql://sahool_user:${DB_PASS}@localhost:3306/sahool

# Security
JWT_SECRET=${JWT_SECRET}

# Application
NODE_ENV=production
PORT=3000

# Owner Info
OWNER_OPEN_ID=admin
OWNER_NAME=مدير النظام

# App Settings
VITE_APP_TITLE=منصة سهول
VITE_APP_LOGO=/logo.png

# Redis
REDIS_URL=redis://localhost:6379

# APIs (يمكن إضافتها لاحقاً)
OPENWEATHER_API_KEY=
IQAIR_API_KEY=
NREL_API_KEY=
SENTINEL_HUB_CLIENT_ID=
SENTINEL_HUB_CLIENT_SECRET=
EOF

print_success "بنجاح .env تم إنشاء ملف"
}

# تثبيت حزم المشروع
install_dependencies() {
    print_info "تثبيت حزم المشروع... (قد يستغرق 5-10 دقائق)"

    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    pnpm install --frozen-lockfile

    print_success "تم تثبيت الحزم بنجاح"
}
```

```
# بناء المشروع
build_project() {
    print_info "بناء المشروع..."

    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    pnpm run build

    print_success "تم بناء المشروع بنجاح"
}

# إعداد قاعدة البيانات
setup_database() {
    print_info "إعداد جداول قاعدة البيانات..."

    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    pnpm db:push

    print_success "تم إعداد قاعدة البيانات بنجاح"
}

# إعداد PM2
setup_pm2() {
    print_info "إعداد PM2..."

    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    # إيقاف التطبيق إذا كان يعمل
    pm2 delete sahoool-platform 2>/dev/null || true

    # تشغيل التطبيق
    pm2 start dist/index.js --name sahoool-platform
    pm2 save
    pm2 startup systemd -u root --hp /root

    print_success "بنجاح PM2 تم إعداد"
}

# إعداد Nginx
setup_nginx() {
    print_info "إعداد Nginx..."
}
```



```

# السيرفر IP الحصول على
SERVER_IP=$(hostname -I | awk '{print $1}')

read -p "أدخل اسم النطاق (أو اضغط Enter للاستخدام IP: $SERVER_IP): " DOMAIN

if [ -z "$DOMAIN" ]; then
    DOMAIN=$SERVER_IP
fi

# إنشاء ملف إعداد Nginx
cat > /etc/nginx/sites-available/sahool << EOF
upstream sahool_backend {
    server 127.0.0.1:3000;
    keepalive 64;
}

server {
    listen 80;
    server_name ${DOMAIN};

    # Security Headers
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;

    # Logging
    access_log /var/log/nginx/sahool-access.log;
    error_log /var/log/nginx/sahool-error.log;

    # Client Body Size
    client_max_body_size 50M;

    # Proxy Settings
    location / {
        proxy_pass http://sahool_backend;
        proxy_http_version 1.1;

        proxy_set_header Upgrade \$http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host \$host;
        proxy_set_header X-Real-IP \$remote_addr;
        proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto \$scheme;

        proxy_cache_bypass \$http_upgrade;

```

```

        proxy_buffering off;
    }

    # Static Files
    location /assets {
        alias /opt/sahool-platform/dist/public/assets;
        expires 1y;
        add_header Cache-Control "public, immutable";
    }
}
EOF

# تفعيل الإعداد
ln -sf /etc/nginx/sites-available/sahool /etc/nginx/sites-enabled/
rm -f /etc/nginx/sites-enabled/default

# اختبار وإعادة تشغيل Nginx
nginx -t
systemctl restart nginx

print_success "بنجاح Nginx تم إعداد"

# حفظ النطاق للاستخدام لاحقاً
echo "$DOMAIN" > /tmp/sahool_domain
}

# إعداد SSL (اختياري)
setup_ssl() {
    print_info "إعداد SSL/TLS..."

    read -p "هل تريد تثبيت شهادة؟ (y/n): " -n 1 -r
    echo

    if [[ ! $REPLY =~ ^[Yy]$ ]]; then
        return
    fi

    DOMAIN=$(cat /tmp/sahool_domain 2>/dev/null || echo "")

    if [ -z "$DOMAIN" ]; then
        read -p "أدخل اسم النطاق: " DOMAIN
    fi

    # التحقق من نوع الشهادة
    echo "اختر نوع الشهادة:"
    echo "1) Let's Encrypt (للنطاقات العامة)"

```

```

echo "2) شهادة ذاتية التوقيع (للشبكة المحلية)"
read -p "(1/2) الاختيار: " SSL_CHOICE

if [ "$SSL_CHOICE" == "1" ]; then
    # Let's Encrypt
    apt install -y certbot python3-certbot-nginx
    certbot --nginx -d "$DOMAIN" --non-interactive --agree-tos --
register-unsafely-without-email
else
    # شهادة ذاتية
    mkdir -p /etc/nginx/ssl
    openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
        -keyout /etc/nginx/ssl/sahool.key \
        -out /etc/nginx/ssl/sahool.crt \
        -subj "/C=SA/ST=Riyadh/L=Riyadh/O=Sahool/CN=$DOMAIN"

    # تحديث إعداد Nginx
    cat > /etc/nginx/sites-available/sahool << EOF
upstream sahoool_backend {
    server 127.0.0.1:3000;
    keepalive 64;
}

server {
    listen 443 ssl http2;
    server_name ${DOMAIN};

    ssl_certificate /etc/nginx/ssl/sahool.crt;
    ssl_certificate_key /etc/nginx/ssl/sahool.key;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    # Security Headers
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;

    # Logging
    access_log /var/log/nginx/sahool-access.log;
    error_log /var/log/nginx/sahool-error.log;

    # Client Body Size
    client_max_body_size 50M;

    # Proxy Settings

```

```

location / {
    proxy_pass http://sahool_backend;
    proxy_http_version 1.1;

    proxy_set_header Upgrade \ $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host \ $host;
    proxy_set_header X-Real-IP \ $remote_addr;
    proxy_set_header X-Forwarded-For \ $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto \ $scheme;

    proxy_cache_bypass \ $http_upgrade;
    proxy_buffering off;
}

# Static Files
location /assets {
    alias /opt/sahool-platform/dist/public/assets;
    expires 1y;
    add_header Cache-Control "public, immutable";
}
}

server {
    listen 80;
    server_name ${DOMAIN};
    return 301 https://\ $server_name\ $request_uri;
}
EOF

    nginx -t
    systemctl restart nginx
fi

print_success "بنجاح SSL تم إعداد"
rm -f /tmp/sahool_domain
}

# إعداد النسخ الاحتياطي
setup_backup() {
    print_info "...إعداد النسخ الاحتياطي التلقائي"

    mkdir -p /backup/sahool

    # إنشاء سكريبت النسخ الاحتياطي
    cat > /opt/sahool-platform/backup.sh << 'EOF'

```

```
#!/bin/bash
BACKUP_DIR="/backup/sahool"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR

# نسخ قاعدة البيانات
DB_PASS=$(grep DATABASE_URL /opt/sahool-platform/.env | cut -d':' -f3 | cut -d'@' -f1)
mysqldump -u sahoool_user -p"$DB_PASS" sahoool > $BACKUP_DIR/db_$DATE.sql

# ضغط النسخة
gzip $BACKUP_DIR/db_$DATE.sql

# حذف النسخ القديمة (أكثر من 7 أيام)
find $BACKUP_DIR -name "*.sql.gz" -mtime +7 -delete

echo "Backup completed: $DATE"
EOF

chmod +x /opt/sahool-platform/backup.sh

# إضافة إلى cron
(crontab -l 2>/dev/null; echo "0 2 * * * /opt/sahool-platform/backup.sh
>> /var/log/sahool-backup.log 2>&1") | crontab -

print_success "تم إعداد النسخ الاحتياطي التلقائي (يوميًا الساعة 2 صباحاً)"
}

# التحقق من التنصيب
verify_installation() {
    print_info "التحقق من التنصيب..."

    # التحقق من الخدمات
    SERVICES=("mysql" "redis-server" "nginx" "pm2")

    for service in "${SERVICES[@]}; do
        if systemctl is-active --quiet "$service" 2>/dev/null || pm2 list |
grep -q "sahool-platform" 2>/dev/null; then
            print_success "$service يعمل بنجاح"
        else
            print_warning "$service قد لا يعمل بشكل صحيح"
        fi
    done

    # التحقق من التطبيق
```

```
sleep 3
if curl -s http://localhost:3000 > /dev/null; then
    print_success "التطبيق يعمل بنجاح"
else
    print_warning "التطبيق قد لا يعمل بشكل صحيح"
fi
}

# رسالة النهاية
final_message() {
    DOMAIN=$(hostname -I | awk '{print $1}')

    echo ""
    echo "===== "
    echo "|| "
    echo "||                  🎉! تم التنصيب بنجاح                  || "
    echo "|| "
    echo "|| "
    echo "===== "
    print_success "! منصة سهول جاهزة للاستخدام"
    echo ""
    echo "معلومات الوصول:"
    echo "  - HTTP:  http://$DOMAIN"
    echo "  - HTTPS: https://$DOMAIN (إذا تم تفعيل SSL)"
    echo ""
    echo "الأوامر المفيدة:"
    echo "  - عرض حالة التطبيق:  pm2 status"
    echo "  - عرض اللوجات:         pm2 logs sahoor-platform"
    echo "  - إعادة التشغيل:      pm2 restart sahoor-platform"
    echo "  - إيقاف التطبيق:      pm2 stop sahoor-platform"
    echo ""
    echo "مسارات مهمة:"
    echo "  - المشروع:            /opt/sahoor-platform"
    echo "  - النسخ الاحتياطي:    /backup/sahoor"
    echo "  - لوجات Nginx:         /var/log/nginx/sahoor-*.log"
    echo ""
    print_info "لمزيد من المعلومات: https://github.com/kafaat/sahoor-platform"
    echo ""
}

#####
# البرنامج الرئيسي
#####

main() {
    check_root
```

```
welcome_message

update_system
install_nodejs
install_mysql
install_redis
install_nginx
install_pm2
setup_firewall

download_project
setup_env
install_dependencies
build_project
setup_database

setup_pm2
setup_nginx
setup_ssl
setup_backup

verify_installation
final_message
}

# تشغيل البرنامج الرئيسي
main

exit 0
```

ملفات الإعداد الإضافية

2. ملف systemd service (بديل لـ PM2)

احفظ باسم `sahool.service` :

[Unit]

Description=Sahool Platform

Documentation=<https://github.com/kafaat/sahool-platform>

After=network.target mysql.service redis.service

[Service]

Type=simple

User=root

WorkingDirectory=/opt/sahool-platform

Environment=NODE_ENV=production

ExecStart=/usr/bin/node /opt/sahool-platform/dist/index.js

Restart=on-failure

RestartSec=10

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=sahool-platform

[Install]

WantedBy=multi-user.target

التثبيت:

```
sudo cp sahoool.service /etc/systemd/system/  
sudo systemctl daemon-reload  
sudo systemctl enable sahoool  
sudo systemctl start sahoool
```

3. سكريبت التحديث

احفظ باسم `update.sh`:


```
#!/bin/bash

#=====
# سكربت تحديث منصة سَھول
#=====

set -e

APP_DIR="/opt/sahool-platform"
BACKUP_DIR="/backup/sahool/updates"

echo "🔄 بدء عملية التحديث..."

# إنشاء نسخة احتياطية
echo "📦 إنشاء نسخة احتياطية..."
mkdir -p "$BACKUP_DIR"
DATE=$(date +%Y%m%d_%H%M%S)

# نسخ قاعدة البيانات
DB_PASS=$(grep DATABASE_URL "$APP_DIR/.env" | cut -d':' -f3 | cut -d'@' -f1)
mysqldump -u sahoool_user -p"$DB_PASS" sahoool | gzip >
"$BACKUP_DIR/db_before_update_$DATE.sql.gz"

# نسخ الملفات
tar -czf "$BACKUP_DIR/app_before_update_$DATE.tar.gz" -C /opt sahoool-
platform

echo "✅ تم إنشاء النسخة الاحتياطية"

# إيقاف التطبيق
echo "🛑 إيقاف التطبيق..."
pm2 stop sahoool-platform

# تحديث الكود
echo "⬇️ تحديث الكود من GitHub..."
cd "$APP_DIR"
git pull origin master

# تثبيت الحزم الجديدة
echo "📦 تثبيت الحزم..."
pnpm install --frozen-lockfile

# بناء المشروع
echo "🔨 بناء المشروع..."
```

```
pnpm run build

# تحديث قاعدة البيانات
echo "🔄 تحديث قاعدة البيانات..."
pnpm db:push

# إعادة تشغيل التطبيق
echo "▶ إعادة تشغيل التطبيق..."
pm2 restart sahoool-platform

# التحقق من الحالة
sleep 3
pm2 status sahoool-platform

echo "✅ تم التحديث بنجاح"
echo "📊 pm2 logs sahoool-platform"
```

4. سكريبت المراقبة

احفظ باسم `monitor.sh`:

```
#!/bin/bash
```

```
#=====
# سكربت مراقبة منصة سهول
#=====
```

```
APP_DIR="/opt/sahool-platform"
```

```
# الألوان
GREEN='\033[0;32m'
RED='\033[0;31m'
YELLOW='\033[1;33m'
NC='\033[0m'
```

```
echo "
echo "|| منصة سهول - لوحة المراقبة ||"
echo "
echo ""
```

```
# حالة الخدمات
echo "📊 حالة الخدمات:"
echo
echo "_____"
```

```
services=("mysql" "redis-server" "nginx")
for service in "${services[@]}; do
    if systemctl is-active --quiet "$service"; then
        echo -e "     $service: ${GREEN}يعمل${NC}"
    else
        echo -e "     $service: ${RED}متوقف${NC}"
    fi
done
```

```
# حالة التطبيق
if pm2 list | grep -q "sahool-platform.*online"; then
    echo -e "    ✅ sahoool-platform: ${GREEN}يعمل${NC}"
else
    echo -e "    ❌ sahoool-platform: ${RED}متوقف${NC}"
fi
```

echo ""

```
# استخدام الموارد
echo "استخدام الموارد 🖥️"
echo
```

```
"  
  
# CPU  
CPU_USAGE=$(top -bn1 | grep "Cpu(s)" | awk '{print $2}' | cut -d'%' -f1)  
echo " CPU: ${CPU_USAGE}%"  
  
# RAM  
MEM_TOTAL=$(free -m | awk 'NR==2{print $2}')  
MEM_USED=$(free -m | awk 'NR==2{print $3}')  
MEM_PERCENT=$(awk "BEGIN {printf \"%.1f\\\", ($MEM_USED/$MEM_TOTAL)*100}")  
echo " RAM: ${MEM_USED}MB / ${MEM_TOTAL}MB (${MEM_PERCENT}%)"  
  
# Disk  
DISK_USAGE=$(df -h / | awk 'NR==2{print $5}')  
echo " Disk: $DISK_USAGE مستخدم"  
  
echo ""  
  
# قاعدة البيانات  
echo " 📊 قاعدة البيانات:"  
echo  
"  
  
DB_PASS=$(grep DATABASE_URL "$APP_DIR/.env" | cut -d':' -f3 | cut -d'@' -f1)  
DB_SIZE=$(mysql -u sahoor_user -p"$DB_PASS" -e "SELECT  
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)' FROM  
information_schema.tables WHERE table_schema='sahoor';" -sN)  
echo " حجم قاعدة البيانات: ${DB_SIZE} MB"  
  
TABLE_COUNT=$(mysql -u sahoor_user -p"$DB_PASS" -e "SELECT COUNT(*) FROM  
information_schema.tables WHERE table_schema='sahoor';" -sN)  
echo " عدد الجداول: $TABLE_COUNT"  
  
echo ""  
  
# آخر 5 سطور من اللوجات  
echo " 📝 آخر اللوجات:"  
echo  
"  
  
pm2 logs sahoor-platform --lines 5 --nostream  
  
echo ""  
echo " لعرض اللوجات الكاملة: pm2 logs sahoor-platform"
```

دليل التشغيل السريع 🚀

خطوات التنصيب:

```
# 1. تحميل السكريبت التنصيب
wget https://raw.githubusercontent.com/kafaat/sahool-
platform/master/install.sh

# 2. إعطاء صلاحيات التنفيذ
chmod +x install.sh

# 3. تشغيل السكريبت
sudo bash install.sh
```

بعد التنصيب:

```
# عرض حالة التطبيق
pm2 status

# عرض اللوجات
pm2 logs sahoool-platform

# إعادة التشغيل
pm2 restart sahoool-platform

# المراقبة
bash monitor.sh

# التحديث
bash update.sh
```

📞 الدعم

للمزيد من المعلومات:

GitHub: <https://github.com/kafaat/sahool-platform> •

- التوثيق: https://github.com/kafaat/sahool-platform/blob/master/API_DOCUMENTATION.md
-

تم إنشاء هذا الملف بواسطة فريق منصة سَهل 🌾