# **MediTrack Full-Stack Setup Guide**

## **📋 Project Overview**

MediTrack is a comprehensive electronic pharmacy management system built with modern web technologies. This guide will help you set up the complete development environment on your local machine.

https://www.loom.com/share/9d6dea0a9ed44ae7b089de19e5395666?sid=a934c570-06f0-4103-98f6-5afe20e30ae1

### **Tech Stack Summary**

|  |  |  |
| --- | --- | --- |
| **Component** | **Technology** | **Purpose** |
| **Backend API** | Laravel 12 (PHP 8.4) | RESTful API, authentication, business logic |
| **Database** | MariaDB/MySQL | Data persistence, user management, prescriptions |
| **Frontend** | Next.js 14 (React 18) | User interface, Arabic RTL support |
| **Styling** | Tailwind CSS | Responsive design, Arabic typography |
| **Authentication** | Laravel Sanctum | Token-based API authentication |
| **Language** | Arabic (RTL) | Primary interface language |

### **Project Structure**

meditrack-project/

├── meditrack-backend/ # Laravel API (PHP)

├── meditrack-frontend/ # Next.js app (TypeScript)

└── docs/ # Documentation

## **Prerequisites & System Requirements**

### **Required Software Versions**

|  |  |  |
| --- | --- | --- |
| **Tool** | **Minimum Version** | **Recommended** |
| **PHP** | 8.1+ | 8.4+ |
| **Composer** | 2.5+ | Latest |
| **Node.js** | 18.0+ | 20.0+ |
| **npm** | 9.0+ | Latest |
| **MariaDB/MySQL** | 10.4+ | Latest |
| **Git** | 2.30+ | Latest |

## **Step 1: Development Environment Setup**

### **For Linux (Ubuntu/Debian)**

#### **1.1 Update System**

sudo apt update && sudo apt upgrade -y

#### **1.2 Install PHP 8.4 and Extensions**

# Add PHP repository

sudo add-apt-repository ppa:ondrej/php -y

sudo apt update

# Install PHP and required extensions

sudo apt install -y php8.4 php8.4-cli php8.4-fpm php8.4-mysql php8.4-curl \

php8.4-gd php8.4-mbstring php8.4-xml php8.4-zip php8.4-bcmath \

php8.4-tokenizer php8.4-dom php8.4-intl

# Verify installation

php --version

#### **1.3 Install Composer**

# Download and install Composer

curl -sS https://getcomposer.org/installer | php

sudo mv composer.phar /usr/local/bin/composer

sudo chmod +x /usr/local/bin/composer

# Verify installation

composer --version

#### **1.4 Install MariaDB**

# Install MariaDB

sudo apt install -y mariadb-server mariadb-client

# Start and enable MariaDB

sudo systemctl start mariadb

sudo systemctl enable mariadb

# Secure MariaDB installation

sudo mysql\_secure\_installation

#### **1.5 Install Node.js and npm**

# Install Node.js 20.x

curl -fsSL https://deb.nodesource.com/setup\_20.x | sudo -E bash -

sudo apt install -y nodejs

# Verify installation

node --version

npm --version

### **For Windows**

#### **1.1 Install PHP**

1. Download PHP 8.4+ from [php.net](https://windows.php.net/download/)
2. Extract to C:\php
3. Add C:\php to your PATH environment variable
4. Copy php.ini-development to php.ini
5. Enable required extensions in php.ini: extension=curlextension=gdextension=mbstringextension=mysqliextension=pdo\_mysqlextension=zipextension=bcmathextension=intl

#### **1.2 Install Composer**

1. Download and run [Composer-Setup.exe](https://getcomposer.org/Composer-Setup.exe)
2. Follow the installation wizard
3. Verify in Command Prompt: composer --version

#### **1.3 Install MariaDB/MySQL**

1. Download [MariaDB](https://mariadb.org/download/) or [MySQL](https://dev.mysql.com/downloads/mysql/)
2. Run the installer and follow setup wizard
3. Remember your root password
4. Start MySQL/MariaDB service

#### **1.4 Install Node.js**

1. Download Node.js 20.x+ from [nodejs.org](https://nodejs.org/)
2. Run the installer
3. Verify in Command Prompt: node --version and npm --version

#### **1.5 Install Git**

1. Download [Git for Windows](https://gitforwindows.org/)
2. Install with default settings
3. Use **Git Bash** for running bash commands

## **Step 2: Clone the Repository**

### **Linux/macOS**

# Clone the repository

git clone <repository-url> meditrack-project

cd meditrack-project

# Verify project structure

ls -la

# Should see: meditrack-backend/ and meditrack-frontend/

### **Windows (Git Bash or PowerShell)**

# Clone the repository

git clone <repository-url> meditrack-project

cd meditrack-project

# Verify project structure

dir

# Should see: meditrack-backend/ and meditrack-frontend/

## **Step 3: Database Setup**

### **3.1 Create Database and User**

#### **Linux/macOS**

# Connect to MariaDB as root

sudo mysql -u root -p

# In MariaDB shell, run:

#### **Windows**

# Connect to MySQL/MariaDB

mysql -u root -p

# In MySQL shell, run:

#### **Database Commands (All Platforms)**

-- Create database

CREATE DATABASE meditrack CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

-- Create user

CREATE USER 'meditrack\_user'@'localhost' IDENTIFIED BY 'meditrack\_password123';

-- Grant privileges

GRANT ALL PRIVILEGES ON meditrack.\* TO 'meditrack\_user'@'localhost';

FLUSH PRIVILEGES;

-- Verify database creation

SHOW DATABASES;

-- Exit

EXIT;

### **3.2 Test Database Connection**

# Test connection with new user

mysql -u meditrack\_user -p meditrack

# Enter password: meditrack\_password123

# Type: EXIT; to leave

## **Step 4: Backend Setup (Laravel)**

### **4.1 Navigate to Backend Directory**

cd meditrack-backend

### **4.2 Install PHP Dependencies**

# Install Composer dependencies

composer install

# If you get memory limit errors (Linux):

php -d memory\_limit=512M /usr/local/bin/composer install

### **4.3 Environment Configuration**

# Copy environment file

cp .env.example .env

# Generate application key

php artisan key:generate

### **4.4 Configure .env File**

Edit .env file with your database credentials:

# Linux/macOS

nano .env

# Windows

notepad .env

**Update these values:**

# App Configuration

APP\_NAME="MediTrack"

APP\_ENV=local

APP\_DEBUG=true

APP\_URL=http://localhost:8000

# Database Configuration

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=meditrack

DB\_USERNAME=meditrack\_user

DB\_PASSWORD=meditrack\_password123

# Session Configuration (Important for API)

SESSION\_DRIVER=array

SESSION\_LIFETIME=120

# Sanctum Configuration

SANCTUM\_EXPIRATION=1440

SANCTUM\_STATEFUL\_DOMAINS=localhost:3000,127.0.0.1:3000

### **4.5 Database Migration and Seeding**

# Test database connection

php artisan tinker

# In tinker: DB::connection()->getPdo();

# Should show PDO object. Type: exit

# Run migrations

php artisan migrate

# Seed database with sample data

php artisan db:seed

# Verify seeded data

php artisan tinker

# In tinker: App\Models\User::count();

# Should show user count. Type: exit

### **4.6 Install and Configure Sanctum**

# Install Sanctum (if not already installed)

composer require laravel/sanctum

# Publish Sanctum configuration

php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"

## **Step 5: Frontend Setup (Next.js)**

### **5.1 Navigate to Frontend Directory**

cd ../meditrack-frontend

### **5.2 Install Node.js Dependencies**

# Install npm packages

npm install

# If you encounter permission errors on Linux:

sudo npm install

# Alternative: Use yarn

# npm install -g yarn

# yarn install

### **5.3 Environment Configuration**

# Create environment file

touch .env.local # Linux/macOS

# OR

echo. > .env.local # Windows

**Add environment variables:**

# Linux/macOS

nano .env.local

# Windows

notepad .env.local

**Content for .env.local:**

# API Configuration

NEXT\_PUBLIC\_API\_URL=http://localhost:8000/api

NEXT\_PUBLIC\_API\_BASE\_URL=http://localhost:8000

# App Configuration

NEXT\_PUBLIC\_APP\_NAME=MediTrack

NEXT\_PUBLIC\_APP\_VERSION=1.0.0

## **Step 6: Running the Application**

### **6.1 Start Backend Server**

**Terminal 1 (Backend):**

cd meditrack-backend

# Start Laravel development server

php artisan serve

# You should see:

# Laravel development server started: http://127.0.0.1:8000

### **6.2 Start Frontend Server**

**Terminal 2 (Frontend):**

cd meditrack-frontend

# Start Next.js development server

npm run dev

# You should see:

# Ready - started server on 0.0.0.0:3000

### **6.3 Access the Application**

* **Frontend**: http://localhost:3000
* **Backend API**: http://localhost:8000/api
* **Backend Health Check**: http://localhost:8000/up

## **Step 7: Verification & Testing**

### **7.1 Test API Connection**

# Test API endpoints

curl http://localhost:8000/api/medicines

# Should return JSON response

# Test login endpoint

curl -X POST http://localhost:8000/api/login \

-H "Content-Type: application/json" \

-d '{"email":"ahmed.hassan@meditrack.com","password":"password123"}'

# Should return successful login with token

### **7.2 Test Frontend**

1. **Open browser**: http://localhost:3000
2. **Check homepage**: Should show Arabic RTL layout
3. **Test login**: Go to "تسجيل دخول" (Login)
4. **Use test credentials**:
   * Email: ahmed.hassan@meditrack.com
   * Password: password123
   * Role: Doctor

### **7.3 Full Integration Test**

1. **Login as doctor** → Should redirect to doctor dashboard
2. **Login as patient** → Should redirect to patient dashboard
3. **Test API calls** → Check browser Network tab for successful requests

## **Troubleshooting Common Issues**

### **Backend Issues**

#### **Issue: PHP Extensions Missing**

# Linux: Install missing extensions

sudo apt install php8.4-{extension-name}

# Example for missing mbstring:

sudo apt install php8.4-mbstring

# Restart web server

sudo systemctl restart apache2 # or nginx

#### **Issue: Composer Memory Limit**

# Increase PHP memory limit

php -d memory\_limit=512M /usr/local/bin/composer install

# Or update php.ini:

memory\_limit = 512M

#### **Issue: Database Connection Failed**

# Check MariaDB/MySQL is running

sudo systemctl status mariadb # Linux

# OR

net start mysql # Windows

# Test database connection

mysql -u meditrack\_user -p meditrack

# Check .env database credentials

cat .env | grep DB\_

#### **Issue: Migration Errors**

# Reset migrations if needed

php artisan migrate:fresh --seed

# Check migration status

php artisan migrate:status

#### **Issue: Port 8000 Already in Use**

# Kill process using port 8000

sudo lsof -ti:8000 | xargs kill -9 # Linux

# OR

netstat -ano | findstr :8000 # Windows (note the PID)

taskkill /PID <PID> /F # Windows

# Use different port

php artisan serve --port=8001

### **Frontend Issues**

#### **Issue: Node Modules Installation Failed**

# Clear npm cache

npm cache clean --force

# Delete node\_modules and reinstall

rm -rf node\_modules package-lock.json # Linux/macOS

# OR

rmdir /s node\_modules & del package-lock.json # Windows

npm install

#### **Issue: Port 3000 Already in Use**

# Kill process using port 3000

sudo lsof -ti:3000 | xargs kill -9 # Linux

# OR

netstat -ano | findstr :3000 # Windows

taskkill /PID <PID> /F # Windows

# Use different port

npm run dev -- --port 3001

#### **Issue: Tailwind CSS Not Working**

# Clear Next.js cache

rm -rf .next # Linux/macOS

# OR

rmdir /s .next # Windows

# Restart development server

npm run dev

#### **Issue: API Connection Failed**

1. **Check backend is running**: http://localhost:8000
2. **Verify .env.local**: Check NEXT\_PUBLIC\_API\_URL
3. **Check browser console**: Look for CORS or network errors
4. **Test API directly**: curl http://localhost:8000/api/medicines

### **CORS Issues**

#### **Issue: CORS Policy Errors**

# Check CORS middleware is registered in bootstrap/app.php

# Verify allowed origins include http://localhost:3000

# Clear Laravel cache

php artisan config:clear

php artisan cache:clear

### **Permission Issues (Linux)**

# Fix Laravel permissions

sudo chown -R $USER:www-data storage bootstrap/cache

sudo chmod -R 775 storage bootstrap/cache

# Fix npm permissions

sudo chown -R $USER ~/.npm

## **Development Workflow**

### **Daily Startup Routine**

# Terminal 1: Start backend

cd meditrack-backend && php artisan serve

# Terminal 2: Start frontend

cd meditrack-frontend && npm run dev

# Terminal 3: Database operations (when needed)

mysql -u meditrack\_user -p meditrack

### **Useful Development Commands**

#### **Backend Commands**

# View Laravel logs

tail -f storage/logs/laravel.log

# Clear all caches

php artisan optimize:clear

# Check routes

php artisan route:list --path=api

# Reset database with fresh data

php artisan migrate:fresh --seed

# Generate new application key

php artisan key:generate

#### **Frontend Commands**

# Build for production

npm run build

# Lint code

npm run lint

# Clear Next.js cache

rm -rf .next

# Install new packages

npm install <package-name>

## **Additional Resources**

### **Documentation Links**

* [Laravel 12 Documentation](https://laravel.com/docs/12.x)
* [Next.js 14 Documentation](https://nextjs.org/docs)
* [Tailwind CSS Documentation](https://tailwindcss.com/docs)
* [Laravel Sanctum Documentation](https://laravel.com/docs/12.x/sanctum)

### **Seeded Test Data**

* **Doctors**: 5 sample doctors with credentials
* **Patients**: 10 sample patients
* **Medicines**: 15 sample medicines
* **Prescriptions**: 20 sample prescriptions

**Test Credentials:**

* **Doctor**: ahmed.hassan@meditrack.com / password123
* **Patient**: mohammad.salim@patient.com / password123

### **API Endpoints Overview**

POST /api/register # User registration

POST /api/login # User login

POST /api/logout # User logout

GET /api/me # Get current user

GET /api/medicines # Get medicines list

GET /api/prescriptions # Get user prescriptions

POST /api/doctors/prescriptions # Create prescription (doctors only)

## **Success Checklist**

Your setup is complete when:

* [ ] Backend server starts without errors (php artisan serve)
* [ ] Frontend server starts without errors (npm run dev)
* [ ] Database connection successful
* [ ] Homepage loads with Arabic RTL layout
* [ ] Login works with test credentials
* [ ] API requests successful (check browser Network tab)
* [ ] No CORS errors in browser console
* [ ] Doctor/Patient dashboards accessible after login