

هيكلية نظام إدارة مكتب محاماة مركزي

(قاعدة البيانات) MongoDB + (الباك) NestJS + (الفرونت) Next.js: التقنيات

1. نظرة عامة

يدير المنطق البرمجي، المصادقة NestJS تعرض لوحة تحكم للعملاء والمحامين. الباك Next.js واجهة أمامية مبنية بـ التحكم في الصلاحيات (مدير المكتب / محامي / عميل / موظف)، وربط القضايا بالمحامين (JWT + Refresh Tokens)، تخزين جميع تفاصيل القضايا، الجلسات، المستندات، وتتبع حالة القضية MongoDB والعملاء. قاعدة البيانات WebSocket الإشعارات اللحظية عند تحديث حالة القضية أو إضافة مستند تتم عبر

2. المكونات

• الفرونت (Next.js)

• الصفحات: `/login`, `/dashboard`, `/cases`, `/lawyers`, `/clients`, `/hearings`, `/documents`, `/settings`

- التصميم: Tailwind CSS + shadcn/ui
- جلب البيانات: React Query أو SWR
- المصادقة: httpOnly Cookie للتوكنات
- التحديث اللحظي: Socket.io client

• الباك (NestJS)

- الموديولات: AuthModule, UsersModule, CasesModule, HearingsModule, DocumentsModule, NotificationsModule, RolesModule
- المصادقة: Passport + JWT
- التحديث اللحظي: @nestjs/websockets أو Socket.io
- الاتصال بقاعدة البيانات: Mongoose

• قاعدة البيانات (MongoDB)

- المجموعات: users, cases, hearings, documents, notifications, roles
- أو محلي MongoDB Atlas للاستضافة

3. تدفق البيانات

1. العميل أو المحامي يسجل الدخول.
2. Access Token و Refresh Token يُعيد NestJS ويحقق.
3. المستخدم يعرض القضايا الموكلة إليه أو القضايا الخاصة به.
4. يمكن تحديث حالة القضية (قيد المتابعة، جلسة قادمة، حكم صادر... إلخ).

رفع المستندات وربطها بالقضية 5.

بث الإشعارات اللحظية للطرفين 6.

4. أمثلة سكيما MongoDB

users:

```
{
  "_id": "ObjectId",
  "email": "string",
  "passwordHash": "string",
  "role": "admin|lawyer|client|staff",
  "specialization": ["criminal", "civil", "corporate"],
  "profile": {"firstName": "string", "lastName": "string", "phone": "string"},
  "createdAt": "Date",
  "updatedAt": "Date"
}
```

cases:

```
{
  "_id": "ObjectId",
  "title": "string",
  "description": "string",
  "clientId": "ObjectId",
  "lawyerIds": ["ObjectId"],
  "status": "open|in_progress|closed",
  "court": "string",
  "documents": ["ObjectId"],
  "hearings": ["ObjectId"],
  "createdAt": "Date",
  "updatedAt": "Date"
}
```

hearings:

```
{
  "_id": "ObjectId",
  "caseId": "ObjectId",
  "date": "Date",
  "location": "string",
  "notes": "string",
  "result": "string"
}
```

5. API Endpoints أهم ال

- POST /api/auth/login
- POST /api/auth/refresh
- GET /api/users
- POST /api/cases
- GET /api/cases/:id
- PATCH /api/cases/:id/status
- POST /api/hearings
- GET /api/documents/:id
- POST /api/documents/upload

6. التحديث اللحظي (WebSocket)

- أحداث: case\:updated, hearing\:added, document\:uploaded
- JWT handshake التحقق عبر

7. رفع الملفات

- GridFS أو S3 رفع إلى + multer

8. النشر

- الفرونت: Vercel
- الباك: Docker على AWS أو Render
- قاعدة البيانات: MongoDB Atlas

9. الأمان

- httpOnly Cookie تخزين التوكن في
- HTTPS إلزامي
- Rate limiting
- التحقق من المدخلات (DTO + Zod/Joi)
- RBAC Guards