

هيكلية نظام إدارة مكتب محاماة مركزي

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

نظرة عامة .1

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

المكونات .2

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 لل توكنات
- 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
- الاستضافة أو مطلي: MongoDB Atlas

تدفق البيانات .3

- العميل أو المحامي يسجل الدخول.
- NestJS يتحقق ويعيد Access Token و Refresh Token.
- المستخدم يعرض القضايا الموكلة إليه أو القضايا الخاصة به.
- يمكن تحديث حالة القضية (قيد المتابعة، جلسة قادمة، حكم صادر... إلخ).

رفع المستندات وربطها بالقافية.

بث الإشعارات اللحظية للطرفين.

4. أمثلة سكريبت 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. أهم API Endpoints

- 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
- التحقق عبر JWT handshake

7. رفع الملفات

- multer + رفع إلى S3 أو GridFS

8. النشر

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

9. الأمان

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