دليل هندسي · الإصدار ٢,٠

التوثيق التقني والمعماري

المعمارية ونموذج البيانات وقواعد العمل وواجهات API ونقاط التوسعة والتشغيل وإجراءات التغيير الآمن.

منظومة إخلاء طرف الموظفين — إيجاس

Node.js · Express · PostgreSQL · Sequelize · React · Vite

١٦ أغسطس ٢٠٢٦

الجمهور والنطاق

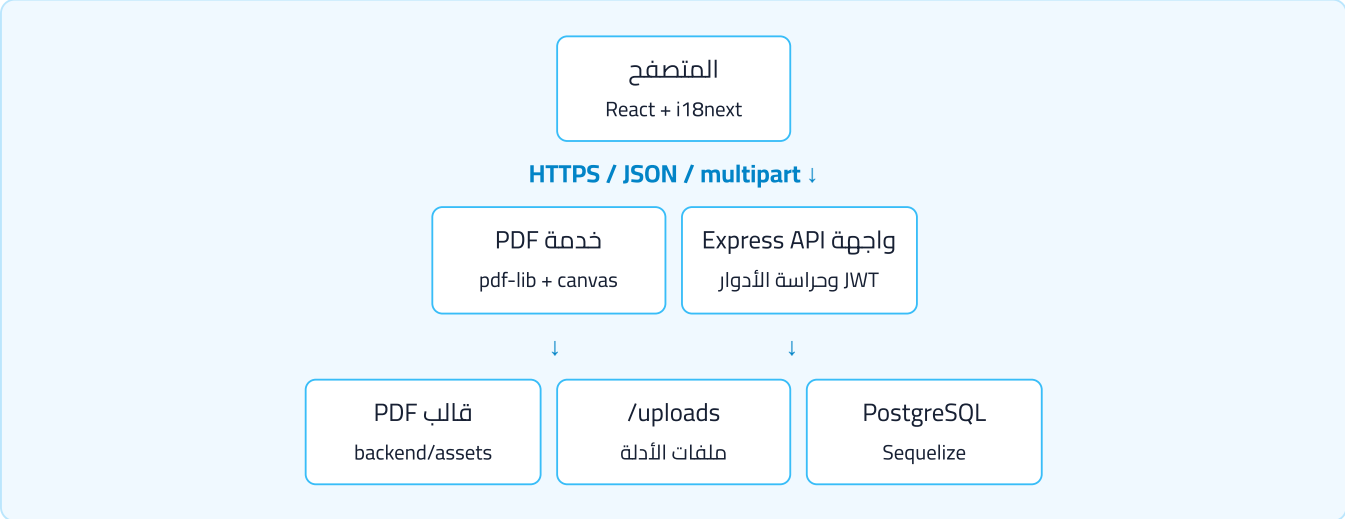
هذا الدليل للمطورين ومسئولي قواعد البيانات والتشغيل الذين يحتاجون إلى فهم التطبيق أو صيانته أو توسيعه أو استكشاف أعطاله. أُعدّ من المستودع الفعلي والنسخة المحلية العاملة.

١. نظرة عامة ومعمارية النظام	التعريف
٢. خريطة المستودع وبيئة التشغيل	المطورون
٣. نموذج البيانات العلائقي	الخلفية / DBA
٤. قواعد العمل والصلاحيات	جميع القائمين بالصيانة
٥. واجهات API ودورة الطلب	الخلفية / التكامل
٦. معمارية الواجهة الأمامية	الواجهة الأمامية
٧. الأدلة وملفات PDF	الخلفية / التشغيل
٨. وصفات التغيير الآمن	تطوير الخصائص
٩. النشر والاختبار والاستعادة	التشغيل
١٠. المخاطر وقائمة الصيانة	مالكو النظام

> **المصادر المرجعية:** المخطط في `backend/src/models/index.js`، والإدارات في `backend/src/seed/departments.data.js`، والصلاحيات في ملفات المسارات ووسيط المصادقة، والسلوك الحالي في الاختبارات والتطبيق العامل.

١. نظرة عامة ومعمارية النظام

تقوم المنظومة برقمنة إخلاء طرف موظفي إيجاس. يتصل تطبيق React أحادي الصفحة بواجهة Express JSON. تتحقق Express من رموز JWT والصلاحيات، وتحفظ البيانات في PostgreSQL عبر Sequelize، وتحفظ أدلة التوقيع على القرص، وتولد نموذج PDF مركبًا من القالب الرسمي.



الطبقة	التقنية	المسئولية
الواجهة الأمامية	React 18، Vite، React Router، Axios، Recharts	لوحات الأدوار والنماذج والتحليلات والمعاينة.
الترجمة	react-i18next	واجهة عربية/إنجليزية ودعم RTL.
الخلفية	Node.js، Express 4	REST والتحقق وقواعد العمل وتقديم الملفات.
المصادقة	JWT، bcryptjs	الدخول بالبريد وكلمة المرور وإعادة التحقق.
قاعدة البيانات	PostgreSQL، Sequelize	الحسابات والقوالب ولقطات الطلب وحالة التدقيق.
الأدلة PDFg	multer، pdf-lib، fontkit، canvas	رفع الملفات ومعاينتها ودمجها في القالب.

شكل التشغيل: تستخدم بيئة التطوير Vite على المنفذ 5173 وAPIg على 4000. عند وجود `frontend/dist` تقدم Express الواجهة المبنية وAPIg من عملية Node واحدة.

٢. خريطة المستودع وبيئة التشغيل

```
Egas-clearance-standalone/
├── backend/
│   ├── assets/                 PDF template, Cairo fonts, demo evidence
│   ├── scripts/                smoke-test.js, stress-test.js
│   ├── src/
│   │   ├── config/db.js        connection + sequelize.sync({alter:true})
│   │   ├── middleware/         JWT and role/department guards
│   │   ├── models/index.js     models and associations
│   │   ├── routes/             auth, departments, requests
│   │   ├── seed/               templates and demo data
│   │   ├── services/           request assembly + clearance PDF
│   │   └── server.js           application entry point
│   └── uploads/                runtime evidence; gitignored
├── frontend/
│   └── src/
│       ├── api/client.js       Axios + JWT interceptor
│       ├── components/         shared UI
│       ├── context/            auth and theme
│       ├── locales/            en.json and ar.json
│       ├── pages/              role screens
│       └── App.jsx             routes and guards
├── README.md
├── CLAUDE.md
└── PROJECT_STATUS.md
```

أوامر التشغيل المحلي

```
npm install
npm install --prefix backend
npm install --prefix frontend

# configure backend/.env
npm run seed:dev --prefix backend
npm run dev

# production-like departments only
npm run seed:final --prefix backend

npm run build --prefix frontend
```

متغيرات البيئة

```
PORT=4000
DATABASE_URL=postgres://USER:PASSWORD@HOST:5432/egas_clearance
JWT_SECRET=replace_with_a_long_random_secret
JWT_EXPIRES_IN=8h
```

يلزم Node.js 18 أو أحدث وPostgreSQL متاح. لا ترفع `backend/.env` أو أسرار الإنتاج إلى Git.

٣. نموذج البيانات العلائقي

انتقل التطبيق من MongoDB إلى PostgreSQL مع الحفاظ على شكل الطلب المتداخل القديم بواسطة `requestAssembly.js`. بريد المستخدم هو المفتاح `users.userID`، ومفتاح الإدارة هو `departments.key`، وتستخدم الطلبات UUID.

الجدول	الغرض والحقول المهمة
departments	قالب الإدارات الـ١٣: الاسمان والترتيب والمرحلة والإشراف ونمط التوقيع.
department_checklist_items	قالب البنود الخمسة لتكنولوجيا المعلومات.
users	البريد كمفتاح، hash، الدور، الإدارة، بند IT، الهاتف، وإجبار تغيير كلمة المرور.
clearance_requests	بيانات الموظف والمغادرة والمنشئ والحالة والاعتماد وسجل إزالة الصلاحيات.
request_departments	لقطة كل إدارة مع التوقيع وبيانات الدليل وهوية الموقع والوقت والملاحظات.
request_items	لقطة بنود IT وتوقيع ودليل كل بند.

قاعدة اللقطة التاريخية

عند إنشاء الطلب، تنسخ الأسماء والترتيب والمرحلة ونمط التوقيع والبنود إلى صفوف تخص الطلب. تؤثر تغييرات القالب اللاحقة على الطلبات الجديدة فقط. لا تربط الطلبات التاريخية بالقالب الحي بدل اللقطة.

```
indexes: [{ unique: true,
  fields: ["requestId", "departmentKey"] }]

indexes: [{ unique: true,
  fields: ["requestDepartmentId", "key"] }]
```

إدارة المخطط: يستدعي بدء التشغيل `sequelize.sync({ alter: true })` ولا توجد migrations منفصلة. اختبر كل تعديل على نسخة شبيهة بالإنتاج وخذ نسخة احتياطية قبل التشغيل.

٤. قواعد العمل والصلاحيات

إنشاء in_progress ‹ اكتمال المرحلة ١ ‹ توقيع ١٣ إدارة ‹ اعتماد الوثائق ‹ إزالة الصلاحيات completed

- الإدارات ١–١١ تعمل بالتوازي.
- لا تكتمل IT إلا باكتمال بنود الطلب الخمسة.
- تُفتح الأجور والمالية بعد اكتمال المرحلة الأولى كلها.
- إعادة الفتح تمسح التوقيع/الدليل وتلغي اعتماد إدارة الوثائق السابق.
- توقيع الإدارات كلها لا يكفي؛ `accessRevoked` هو الشرط النهائي.

الدور	حدود المسارات والبيانات
file_management	جميع الملخصات والأدلة؛ إنشاء وإعادة فتح واعتماد وحذف الطلبات المكتملة وPDF، مع إخفاء هوية الموقع.
reviewer	الإدارة الخاصة فقط، عدا إشراف الأجور/المالية. IT تحصل على مؤشرات جاهزية آمنة.
admin	إدارة file_management وreviewer فقط، دون وصول للطلبات.
super_admin	إدارة admin فقط، وحساب وحيد ينشأ من إعداد قاعدة فارغة.

```
const MANAGEABLE_ROLES_BY_ACTOR = {
  admin: ["file_management", "reviewer"],
  super_admin: ["admin"],
};
```

تفرض الخلفية الصلاحيات. حراس React لتحسين التجربة وليسوا حدًا أمنيًا. يحمل JWT هوية المستخدم والدور والإدارة وبند IT وحالة الإشراف وتغيير كلمة المرور الإجباري.

> **لا تضعف حجب البيانات.** يجب أن تطبق المسارات الجديدة حدود الدور والإدارة والبند والدليل وهوية الموقع نفسها الموجودة في مسارات الطلبات.

٥. واجهات API ودورة الطلب

تبدأ المسارات بـ `/api` . يضيف Axios الرمز من localStorage في ترويسة Bearer.

الطريقة والمسار	الغرض والصلاحية
GET /health	فحص سلامة العملية.
POST /auth/login	تحويل بيانات الدخول إلى رمز.
GET /auth/setup-status POST /auth/setup	الإعداد الأول؛ ينشئ المسؤول الرئيسي فقط إذا كان عدد المستخدمين صفرًا.
GET/POST /auth/accounts	للمسؤولين ومفلتر حسب الأدوار المسموح بها.
POST /auth/reset-password POST /auth/delete-account	إعادة تعيين/حذف مع حدود الدور وإعادة التحقق.
POST /auth/set-new-password	استبدال كلمة المرور المؤقتة.
GET /departments	قالب عام للنماذج.
GET /departments/coverage	فجوات التغطية في المسؤول والوثائق والمراجعين.
POST /requests	إنشاء الطلب ولقطات الإدارات.
GET /requests و id:/	قائمة/تفاصيل حسب الدور.
POST /requests/:id/sign	توقيع إدارة عادية مع دليل.
POST /requests/:id/items/:itemKey/sign	توقيع بند IT مع دليل.
POST .../reopen	مسح التوقيع والدليل للمخول.
POST /requests/:id/approve	اعتماد إدارة الوثائق.
POST /requests/:id/revoke-access	إجراء IT النهائي وحساب الاكتمال.
GET .../evidence/...	تقديم دليل بعد فحص الصلاحية.
GET /requests/:id/pdf	توليد PDF بعد فحص الصلاحية.

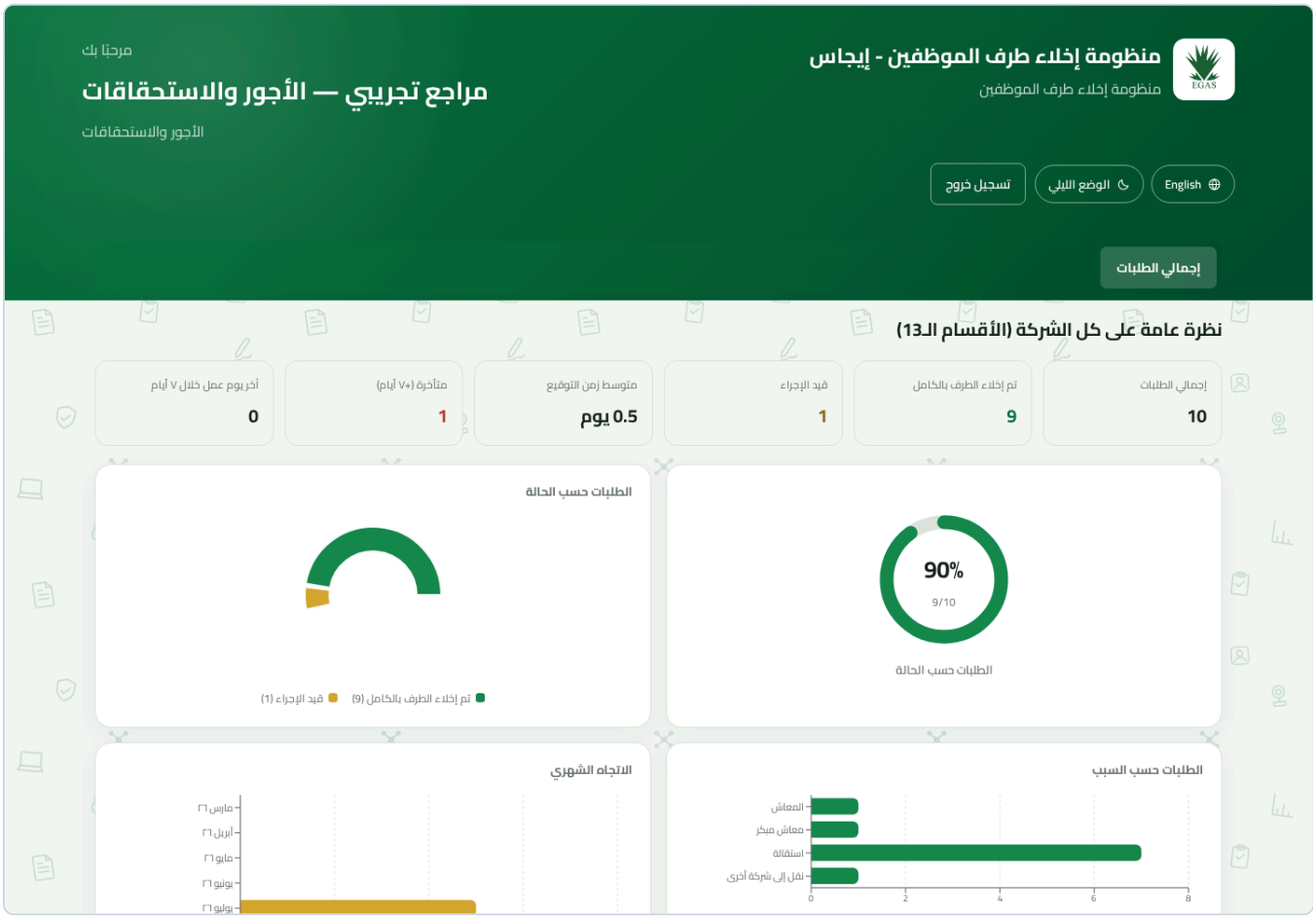
نمط مسار الكتابة

1. المصادقة والتحقق من الدور/الإدارة/البند.
2. جلب الطلب المجمع للتحقق من قواعد العمل.
3. إعادة التحقق من كلمة المرور للإجراء المؤثر.
4. فحص الملف وشروط انتقال الحالة.
5. تحديث صفوف Sequelize المحددة.
6. إعادة الجلب ومزامنة الحالة العامة.
7. إرجاع العرض المسموح للدور فقط.

٦. معمارية الواجهة الأمامية

يعرف `App.jsx` المسارات العامة وأربع لوحات محمية. يحول `RequireRole` غير المسجلين، ويفرض تغيير الكلمة المؤقتة، ويمنع عرض دور خاطئ.

```
/login              Login.jsx
/setup              FirstRunSetup.jsx
/set-new-password   SetNewPassword.jsx
/file-management    FileManagementDashboard.jsx
/reviewer           ReviewerDashboard.jsx
/admin              AdminDashboard.jsx
/super-admin        AdminDashboard.jsx
```



واجهة المراجع العربية الفعلية وتوضح الجداول والمخططات والترجمة والعرض المبني على الدور.

الحالة والمكونات المشتركة

- `AuthContext.jsx` : الدخول والرمز والمستخدم والخروج.
- `ThemeContext.jsx` : الوضع الفاتح/الداكن.
- `DepartmentDashboard.jsx` : تخطيط المراجع والتحليلات.
- `SignaturePanel.jsx` : كلمة المرور والدليل.
- `EvidencePreview.jsx` : معاينة آمنة للصورة/PDF.
- `RequestOversightGrid.jsx` : إشراف ١٣ إدارة.
- `SetupCoverageWarning.jsx` : تحذيرات نقص التغطية.

قاعدة الترجمة

يجب إضافة كل نص جديد إلى `en.json` و `ar.json` . اختبر العربية RTL والإنجليزية، بما فيها النصوص الطويلة والجداول والنوافذ والمخططات. لا تكتب نصوص الواجهة مباشرة داخل المكونات.

٧. الأدلة وملفات إخلاء الطرف

تُحفظ الأدلة تحت `/<backend/uploads/<requestId` . تحفظ قاعدة البيانات الرابط وMIME والاسم الأصلي، بينما تقرر المسارات المحمية حق المستخدم في الملف.

تقبل مسارات التوقيع JPG وPNG وWEBP وPDF. يجب أن يظل التحقق متسقًا بين إرشاد الواجهة multer والمعاينة والبيانات التجريبية ودمج PDF.

قالب PDF الرسمي + بيانات الطلب + أدلة التوقيع ‹ PDF مركب

تحمل `clearancePdf.js` القالب، وتضمن خطوط Cairo، وتعالج الصور وPDF، وتضع التوقيعات في إحداثيات الإدارات، ثم تعيد الملف. لا تستبدل النموذج قبل التحقق من المقاسات وكل الإحداثيات.

> **دورة حياة الأدلة:** إعادة الفتح أو حذف الطلب يمس قاعدة البيانات والملفات معًا. يجب نسخ قاعدة البيانات ومجلد uploads واستعادتهما كوحدة واحدة.

- لا تقدم uploads كمجلد static عام.
- احتفظ بتطبيع المسار وفحص الصلاحية.
- لا تثق في الاسم الأصلي كمسار تخزين.
- تحقق من النوع والحجم في الخادم.
- لا تسجل كلمات المرور أو JWT أو محتوى الأدلة أو رابط DB كاملًا.

٨. وصفات التغيير الآمن

إضافة إدارة أو تعديلها

1. عدّل `departments.data.js` بمفتاح ثابت واسمين وترتيب ومرحلة وإشراف ونمط توقيع.
2. حدّث قالب PDF ومواضع التوقيع إذا تغيرت الصفوف.
3. شغّل upsert الإدارات المناسب.
4. أنشئ حسابات التغطية؛ ولكل بند IT مالك واحد.
5. اختبر طلبًا جديدًا وتأكد أن الطلبات القديمة حافظت على لقطاتها.
6. حدّث الترجمات والعرض والاختبارات والوثائق.

> **لا تغير مفتاح الإدارة بسهولة.** هو مفتاح طبيعي وأجنبي ويظهر في المستخدمين واللقطات والمسارات والعرض وخرائط PDF. اعتبر تغييره migration بيانات.

إضافة سبب مغادرة

1. أضف قيمة enum ثابتة إلى `LEAVING_REASONS` .
2. أضف التسميات العربية والإنجليزية وخرائط المساعدة.
3. اختبر الإنشاء والقوائم والتحليلات وPDF والبيانات التجريبية.
4. خذ نسخة قبل أن تغير Sequelize enum الإنتاج.

إضافة حقل

1. حدد هل يخص القالب الحي أم لقطة الطلب أم كليهما.
2. أضف تحقق النموذج وAPI.
3. حدّث التجميع والإسقاط والحجب حتى لا يتسرب.
4. حدّث المسارات والنموذج والعرض والترجمتين وPDF والاختبارات.
5. اختبر alter على نسخة مماثلة للإنتاج.

إضافة دور أو صلاحية

حدّث enum JWT والوسيط والمسارات والحجب والتوجيه وتسلسل إدارة الحسابات وفحص التغطية والترجمات والاختبارات. ابدأ بصلاحيات الخادم ثم العرض.

تعديل البيانات أو حذفها بأمان

الحاجة	الطريقة المفضلة	ما يجب تجنبه
إنشاء/إعادة/حذف مستخدم	واجهة المسؤول أو API المصدق.	SQL مباشر يتجاوز hash والقواعد.
تعديل قالب الإدارات	مصدر البيانات ثم upsert المتكرر بأمان.	تغيير لقطات الطلبات التاريخية جماعيًا.
حذف طلب	إجراء إدارة الوثائق المخول على طلب مكتمل، ويتعامل مع DB والملفات.	حذف الصف فقط أو الملفات فقط.
تصحيح دليل	إعادة فتح الإدارة/البند ثم التوقيع.	استبدال الملف على القرص.
تصحيح تاريخي	migration معتمدة ومدعومة بنسخة transactiong.	SQL عشوائي على الإنتاج.

```
BEGIN;
SELECT ... FOR UPDATE;

UPDATE ... WHERE ...;

SELECT ...;  -- verify counts and invariants
COMMIT;      -- or ROLLBACK
```

استخدم نافذة صيانة للتغييرات المهمة، وسجل المفاتيح المتأثرة والعدد المتوقع وخطوات الرجوع. لا تعدل `passwordHash` يدويًا؛ استخدم مسار إعادة التعيين.

٩. النشر والاختبار والاستعادة

قائمة الإصدار

- ☐ راجع git diff واستبعد الأسرار والأدلة والملفات المؤقتة.
- ☐ انسخ PostgreSQL uploadsg معًا.
- ☐ ثبت الاعتماديات من lockfiles وبنسخة Node مدعومة.
- ☐ ابنِ الواجهة بنجاح.
- ☐ اختبر المخطط على staging أو نسخة مستعادة.
- ☐ شغل smoke test على قاعدة غير إنتاجية مزروعة.
- ☐ اختبر لوحات الأدوار بالعربية والإنجليزية.
- ☐ اختبر الدخول والتوقيع وإعادة الفتح والاعتماد والمعاينة وPDF وإزالة الصلاحيات.
- ☐ انشر وافحص `GET /api/health`.
- ☐ تأكد من تغطية المسؤول وإدارة الوثائق والإدارات وبنود IT.

```
# terminal 1
npm run dev

# terminal 2 — seeded, non-production server
node backend/scripts/smoke-test.js

npm run build --prefix frontend

# review before using; never target production casually
node backend/scripts/stress-test.js
```

تنشئ smoke test حسابات وطلبات مؤقتة وتختبر حدود الأدوار والمسار الكامل. لا توجه اختبارات الحذف أو الضغط إلى الإنتاج دون مراجعة صريحة.

وحدة النسخ والاستعادة

تشمل النسخة الكاملة قاعدة PostgreSQL ومجلد uploads ومراجعة الكود المنشورة وقالب PDF ومتغيرات البيئة في مخزن أسرار معتمد. اختبر الاستعادة في بيئة معزولة وافحص معاينة الدليل وتوليد PDF، وليس عدد الصفوف فقط.

١٠. المخاطر وقائمة الصيانة

الخطر	الأثر	التخفيف
قفل المسؤول الرئيسي	لا يستطيع دور آخر إعادة تعيينه.	حفظ آمن واعتماد إجراء استعادة تنظيمي مجرّب.
تعديل المخطط تلقائيًا	قد ينفذ sync alter تغييرات عند البدء.	نسخ staging وإصدارات مضبوطة؛ التفكير في migrations رسمية.
الأدلة على الملفات	نسخة DB وحدها ناقصة والتوسع يحتاج تخزينًا مشتركًا.	نسخ موحد ووحدة دائمة؛ object storage عند التوسع.
JWT في localStorage	XSS قد يكشف الجلسة.	CSP ونظافة المدخلات وتجنب HTML غير الآمن؛ دراسة HttpOnly cookies.
مسارات الإعداد والاتصال العامة	سطح غير مصدق مقصود.	أقل حقول ممكنة وإعادة فحص users=0 لحظيًا.
تقادم الوثائق	القواعد المتغيرة تجعل الإجراءات غير آمنة.	تحديث README وCLAUDE وPROJECT_STATUS والاختبارات والأدلة مع التغيير.

تعريف اكتمال أي تغيير

- ☐ قاعدة العمل وأثر الصلاحية واضحان.
- ☐ التحقق والحجب في الخادم مختبران.
- ☐ لقطات الطلبات الجارية ما زالت صحيحة.
- ☐ العربية RTL والإنجليزية سليمتان.
- ☐ أثر الأدلة وPDF تم التحقق منه.
- ☐ build وsmoke test ناجحان.
- ☐ خطة النشر والرجوع موجودة.
- ☐ الوثائق محدثة.

> **المبدأ المعماري الأساسي:** حافظ على لقطات الطلب التاريخية وطبّق كل صلاحية في الخادم. الواجهة والوثائق وانضباط قاعدة البيانات تدعم هاتين القاعدتين ولا تستبدلهما.

تم التحقق من هذا الدليل مقابل المستودع والتطبيق المحلي في ١٦ أغسطس ٢٠٢٦. راجع آخر كود وحالة المشروع قبل أي تغيير إنتاجي.