

تقرير الفحص التقني الشامل للنظام المحاسبي

التاريخ: 13 فبراير 2026

النظام: نظام محاسبي متعدد الكيانات (Frappe Books Migration)

المستودع: alabasi2025/222222222222

المراجع: فحص تكامل تقني آلي شامل

ملخص تنفيذي.

تم إجراء فحص شامل لجميع مكونات النظام المحاسبي المهاجر من نظام Onyx ERP إلى PostgreSQL 18. يغطي هذا التقرير فحص قاعدة البيانات، وواجهة برمجة التطبيقات (API)، والواجهة الأمامية، والأمان، والأداء، وجودة الكود. تم تحديد 42 ملاحظة مصنفة حسب الخطورة إلى: درجة (8)، عالية (12)، متوسطة (14)، منخفضة (8).

نظرة عامة على البنية التقنية.

المكون	التقنية	الإصدار
واجهة الأمامية	React + TypeScript + Vite	React 19.2, Vite 7.1
الخادم الخلفي	Node.js + Express	Express 4.21
ORM	Drizzle ORM	0.45.1
قاعدة البيانات	PostgreSQL	18
الاتصال بالقاعدة	Cloudflare Tunnel	localhost:5432
UI مكتبة	Radix UI + TailwindCSS	أحدث إصدار

المنافذ المستخدمة:

الخدمة	المنفذ
الخادم الخلفي (API)	10001
واجهة الأمامية	10002
خادم MCP	(عن بعد) 3100
PostgreSQL	(عبر النفق) 5432

3. فحص قاعدة البيانات.

3.1 هيكل الجداول

تم إنشاء 16 جدولًا في قاعدة البيانات 22222 وفقاً للمخطط المعرّف في Drizzle ORM:

الجدول	عدد السجلات	الحالة
entities (الكيانات)	8	مُهاجر <input checked="" type="checkbox"/>
accounts (الحسابات)	249	مُهاجر <input checked="" type="checkbox"/>
cash_boxes (الصناديق)	33	مُهاجر <input checked="" type="checkbox"/>
banks_wallets (البنوك/المحفظ) (البنوك/المحفظ)	23	مُهاجر <input checked="" type="checkbox"/>
warehouses (المستودعات)	12	مُهاجر <input checked="" type="checkbox"/>
journal_entries (قيود اليومية)	1,273	مُهاجر <input checked="" type="checkbox"/>
journal_entry_lines (تفاصيل القيود)	1,286	مشكلة توازن 
payment_vouchers (سندات الصرف/القبض)	1,272	مُهاجر <input checked="" type="checkbox"/>
payment_voucher_operations (عمليات السندات)	0	فارغ 
items (الأصناف)	0	لم يُهاجر <input type="checkbox"/>
item_stock (أرصدة المخزون)	0	لم يُهاجر <input type="checkbox"/>
stock_movements (حركات المخزون)	0	لم يُهاجر <input type="checkbox"/>
inter_unit_transfers (التحويلات)	0	لم يُهاجر <input type="checkbox"/>
inter_unit_accounts (حسابات الجاري)	0	لم يُهاجر <input type="checkbox"/>
units (وحدات القياس)	0	لم يُهاجر <input type="checkbox"/>
item_categories (فئات الأصناف)	0	لم يُهاجر <input type="checkbox"/>

إجمالي السجلات المُهاجرة: 4,156 سجل عبر 8 جداول

3.2 المفاتيح الأجنبية (Foreign Keys)

تم تعريف 43 مفتاحاً أجنبياً عبر جميع الجداول. العلاقات المرجعية سليمة ولا توجد سجلات يتيمة:

الفحص	النتيجة
journal_entries سجلات يتيمة في	0 ✓
journal_entry_lines سجلات يتيمة في	0 ✓
payment_vouchers سجلات يتيمة في	0 ✓

3.3 الفهارس (Indexes)

تم إنشاء **24 فهرساً** (17 فهرس مفتاح أصافي + 7 فهارس متخصصة). ومع ذلك، تم اكتشاف **38** عمود مفتاح أجنبى بدون فهرس، مما يؤثر سلباً على أداء الاستعلامات المرتبطة (JOIN).

3.4 مشكلة درجة: القيود غير المتوازنة

نطورة: درجة

تم اكتشاف أن **1,253** قيد يومية من أصل **1,273 (98.4%)** غير متوازنة (المدين لا يساوي الدائن). السبب الجذري هو أن **1,228** قيد يحتوى على سطر واحد فقط بدلاً من سطرين على الأقل (مدين حيث تم ترحيل طرف واحد فقط من Oracle ودائن). هذا يشير إلى مشكلة في عملية الترحيل من كل قيد.

4. فحص واجهة برمجة التطبيقات (API)

4.1 نقاط النهاية المتوفرة

يتوفر **15** ملف route تم تعريف النهاية التالية:

المسار	العمليات	الحالة
/api/entities	GET, GET/:id, POST, PUT, DELETE	✓ يعمل
/api/accounts	GET, GET/:id, POST, PUT, DELETE	✓ يعمل
/api/cash-boxes	GET, GET/:id, POST, PUT, DELETE	✓ يعمل
/api/banks-wallets	GET, GET/:id, POST, PUT, DELETE	✓ يعمل
/api/warehouses	GET, GET/:id, POST, PUT, DELETE	✓ يعمل
/api/journal-entries	GET, GET/:id, POST	✓ يعمل (بطيء)
/api/payments	GET, GET/:id, POST, PUT, DELETE	✓ يعمل (بطيء)
/api/inventory	GET, POST	✓ يعمل
/api/stock-movements	GET, POST	✓ يعمل
/api/item-categories	GET, POST	✓ يعمل
/api/inter-unit-transfers	GET, POST	✓ يعمل
/api/dashboard	GET /stats, GET /charts/revenue	✓ يعمل
/api/model-switch	GET, POST	✓ يعمل
/api/gemini-credits	GET	✓ يعمل
/api/antigravity-credits	GET	✓ يعمل

أوقات الاستجابة 4.2

نقطة النهاية	وقت الاستجابة	التقييم
entities	3.6ms	✓ ممتاز
accounts	4.6ms	✓ ممتاز
cash-boxes	3.2ms	✓ ممتاز
banks-wallets	3.0ms	✓ ممتاز
warehouses	2.8ms	✓ ممتاز
journal-entries	1,024ms	🔴 بطيء جداً
payments	551ms	<span🟡 span="" بطيء<=""></span🟡>

4.3) مشكلة درجة: توليد المفاتيح الأساسية (ID Generation)

نطورة: درجة

ولا UUID لا) بدون قيمة افتراضية `text('id').primaryKey()` جميع الجداول تستخدم يفشل الطلب بخطأ. هذا يعني أن العميل، `id` بدون حقل POST عدد إرسال طلب nanoid. مسؤول عن توليد المعلومات، وهو نمط غير آمن وغير موثوق (**Frontend**).

الاختبار:

- POST بدون ID → خطأ 500 (فشل)
 - POST مع ID → نجاح

لكن هذا النمط غير , () يدوياً ID يولد journalEntries.ts ملاحظة: ملف routes. موحد عبر جميع

4.4 مشكلة N+1 Query في قيود اليومية

نحوه: عالیة

ملف journalEntries.ts يستخدم نمط N+1 Query الكارثي:

- يجلب جميع القيود
 - لكل قيد، يجلب التفاصيل (سطور القيد).

لكل سطر، يجلب بيانات الحساب 3.

لكل قيد، يجلب بيانات الكيان 4.

واحد. هذا يفسر بـ GET مع 1,273 قيد و 1,286 سطر، هذا يعني ~3,800 استعلام لكل طلب (1,024ms).

5. فحص الأمان.

5.1 ملخص الأمان

البند	الحالة	الخطورة
CORS	غير مُفعّل	درجة
Helmet (HTTP Headers)	غير مُفعّل	درجة
Rate Limiting	غير مُفعّل	عالية
المصادقة (Authentication)	غير موجودة	درجة
التفويض (Authorization)	غير موجود	درجة
التحقق من المدخلات	غير موجود	درجة
حماية SQL Injection	محمي (Drizzle ORM)	-
XSS حماية	غير محمي (قبل) HTML/Script	عالية
HTTPS	عبر Cloudflare Tunnel	-

5.2 تفاصيل الثغرات الأمنية

غياب المصادقة والتفويض (درجة)

مفتولة بالكامل لـ أي شخص API لا يوجد أي نظام مصادقة أو تفويض في النظام. جميع نقاط يمكنه الوصول إلى الخادم. يمكن لأي مستخدم قراءة أو تعديل أو حذف أي بيانات مدارسية بدون أي تحقق من الهوية.

حزام الأمان مثبتة لكن غير مستخدمة (درجة)

لكونها غير مستوردة ولا مُفعّلة في package.json الحزم التالية مثبتة في server/index.ts :

- `cors` (v2.8.5) حماية الطلبات عبر النطاقات -
- `helmet` (v8.1.0) HTTP ترويسات أمان -
- `express-rate-limit` (v8.2.1) تحديد معدل الطلبات -
- `morgan` (v1.10.1) تسجيل الطلبات -

5.2.3 ثغرة XSS (عالية)

في حقول النص بدون أي تنقية HTML/JavaScript النظام يقبل ويخزن محتوى:

```
POST /api/entities {"name": "<script>alert(1)</script>"}
```

5.2.4 غياب التحقق من المدخلات (عالية)

في الاعتمادات، لا يتم استخدامها في أي من ملفات الـ `zod` (v4.1.12). رغم وجود مكتبة `routes`. مباشرةً إلى قاعدة البيانات بدون أي تتحقق `req.body` يتم تمرير.

6. فحص الواجهة الأمامية

6.1 هيكل الصفحات

تم تعريف 30 صفحة في مجلد `client/src/pages/`:

الفئة	الصفحات
المحاسبة	ChartOfAccounts, JournalEntries, Payments, Ledger, AccountTypes, AccountSubtypes, AccountGroups
الخزينة	CashBoxes, BanksWallets, BankReconciliation
المخزون	Inventory, Warehouses, StockMovements
الإدارة	EntitySelection, UnitSelection, CompanySelection, OrganizationStructure
التقارير	Dashboard, Reports, Budget, CostCenters
النظام	Settings, Users, Login, Home, NotFound
الأخرى	Customers, Suppliers, Contacts, Invoices, Purchases, FixedAssets, TaxManagement, Currencies, InterUnitTransfers, InterUnitAccounts

6.2 أخطاء TypeScript

عند التحويل البرمجي TypeScript تم اكتشاف 109 خطأ:

الملف	عدد الأخطاء	النوع
OrganizationStructure.tsx	57	أخطاء أنواع وخصائص
seed-all.ts	14	أخطاء أنواع ضمنية any[]
Users.tsx	8	أخطاء أنواع
Payments.tsx	8	أخطاء أنواع
Suppliers.tsx	7	أخطاء أنواع
ملفات أخرى (12 ملف)	15	أخطاء متنوعة

ثناء TypeScript يتجاوز أخطاء **ملاحظة**: رغم وجود هذه الأخطاء، النظام يعمل لأن التحويل.

6.3 ملفات النسخ الاحتياطي

يحتوي على 4 ملفات نسخ احتياطي قديمة يجب تنظيفها / backup / pages / داخل مجلد:

- BanksWallets_backup.tsx
 - CashBoxes_backup.tsx
 - ChartOfAccounts_old.tsx
 - Payments_old_backup.tsx
-

7. فحص جودة الكود.

7.1 نمط الكود

البند	الحالة	الملاحظة
TypeScript Strict Mode	✗ غير مفعّل	أخطاء 109
ESLint	✓ مُعدّ	لم يُفحص
Prettier	✓ مُعدّ	-
Husky (Git Hooks)	✓ مُعدّ	-
اختبارات وحدة	✗ غير موجودة	Vitest مُعدّ لكن بدون اختبارات
تسجيل الأحداث (Logging)	⚠ console.log فقط	Winston مثبت لكن غير مستخدم

7.2 أنماط مشكلة في الكود

7.2.1 AgentExecutionBridge يستخدم مسارات Windows

نطورة: متوسطة

ملف `agentExecutionBridge.ts` يستخدم مسارات Windows في الحالية. يحاول إنشاء مجلدات Linux وهو غير متوافق مع بيئة (D:\AntigravityAgent\...). على مسار غير موجود.

7.2.2 Dashboard يستخدم بيانات وهمية

نطورة: متوسطة

بدلاً من تجميع (hardcoded) تُرجع بيانات وهمية /api/dashboard/charts/revenue. البيانات الفعلية من قيود اليومية.

7.2.3 غياب Drizzle Relations

نطورة: متوسطة

ما يمنع استخدام `relations` في مختلف Drizzle ORM, لم يتم تعريف `db.query` مع `with` في بعض `N+1 Query` للعلاقات. هذا يفسر استخدام نمط `routes`.

7.2.4 عدم تحديث أرصدة الحسابات

نطورة: عالية

عند إنشاء قيد يومية جديد، لا يتم تديث أرصدة الحسابات المرتبطة. يوجد تعليق في الكود يشير لمنطق التحديث الفعلي "placeholder" إلى أن هذا

8. فحص ترحيل البيانات.

8.1 ملخص الترحيل

البند	القيمة
المصدر	Oracle 12c (Onyx ERP)
الهدف	PostgreSQL 18
إجمالي السجلات	4,156
الجدائل المهاجرة	8 من 16
ترميز النص العربي	☑ سليم
السجلات اليتيمة	0 <input checked="" type="checkbox"/>

8.2 مشكلة درجة: قيود اليومية ناقصة

نطورة: درجة

من أصل 1,273 قيد يومية مُهاجر، 1,253 قيد (98.4%) غير متوازن. السبب هو أن كل قيد يحتوي على سطر واحد فقط بدلاً من سطرين على الأقل. هذا يعني أن عملية الترحيل لم تنقل الطرف المقابل (المدين أو الدائن) لكل قيد.

التأثير: البيانات المحاسبية المهاجرة غير صالحة للاستخدام المحاسبي في وضعها الحالي وتحتاج إلى إعادة ترحيل أو إصلاح.

جداول لم تهاجر 8.3

الجداول التالية فارغة ولم يتم ترحيل بيانات إليها:

- items, item_stock, stock_movements (المخزون)
- inter_unit_transfers, inter_unit_accounts (التحويلات بين الوحدات)
- units, item_categories (البيانات المرجعية)
- payment_voucher_operations (عمليات السندات)

جدول الملاحظات حسب الخطورة 9.

حمراء (8 ملاحظات)

#	الملاحظة	المكون	التأثير
C1	غياب المصادقة والتتفويض	API	وصول غير مصرح به لجميع البيانات
C2	غير مفعّل CORS	API	طلبات من أي نطاق
C3	غير مفعّل Helmet	API	ترويسات أمان مفقودة
C4	غير موحد ID توليد	API/DB	ID فشل إنشاء سجلات بدون
C5	من القيود غير متوازنة 98.4%	بيانات	بيانات محاسبية غير صالحة
C6	غياب التحقق من المدخلات	API	بيانات غير صالحة في القاعدة
C7	ثغرة XSS (تخزين HTML)	API	مخزنة XSS هجمات
C8	كلمة مرور القاعدة في الكود	DB	تسريب بيانات الاعتماد

عالیة (12 ملاحظة)

#	الملاحظة	المكون	التأثير
H1	غير مفْعَل Rate Limiting	API	هجمات DDoS/Brute Force
H2	في قيود اليومية N+1 Query	API	بطء شديد (1 ثانية)
H3	في المدفوعات N+1 Query	API	بطء (0.5 ثانية)
H4	عدم تحديث أرصدة الحسابات	منطق الأعمال	أرصدة غير دقيقة
H5	بدون فهرس FK عمود 38	DB	أداء استعلامات ضعيف
H6	109 خطأ TypeScript	كود	مخاطر وقت التشغيل
H7	غياب اختبارات الوحدة	كود	لا ضمان جودة
H8	Winston مثبت لكن غير مستخدم	كود	لا تسجيل منظم
H9	Morgan مثبت لكن غير مستخدم	كود	لا تسجيل طلبات
H10	Farrell payment_voucher_operations	بيانات	سنادات بدون تفاصيل عمليات
H11	Drizzle Relations غياب	DB/ORM	استعلامات غير فعالة
H12	Pagination في API غياب	API	تحميل جميع السجلات دفعة واحدة

متوسطة (14 ملاحظة)

#	الملاحظة	المكون	التأثير
M1	AgentBridge يستخدم مسارات Windows	كود	خطأ في بيئة Linux
M2	بيانات وهمية Dashboard	واجهة	معلومات غير دقيقة
M3	4 ملفات backup في مجلد pages	كود	تلوث الكود
M4	جداول فارغة لم تهاجر 8	بيانات	وظائف معطلة
M5	OrganizationStructure: 57 خطأ TS	كود	صفحة غير مستقرة
M6	nanoid مثبت لكن غير مستخدم	كود	اعتمادات زائدة
M7	uuid مثبت لكن غير مستخدم	كود	اعتمادات زائدة
M8	Cache بسيط accounts فقط	API	أداء غير متسلق
M9	FKs غياب ON DELETE CASCADE في معظم	DB	مشاكل حذف متتالي
M10	الحفظ قبل توازن القيد لا يوجد تحقق من	منطق	قيود غير متوازنة
M11	geminiCredits يستخدم مسارات Windows	كود	خطأ في بيئة Linux
M12	compression middleware لا يوجد	API	حجم استجابة كبير
M13	error handling middleware عم لا يوجد	API	رسائل خطأ غير موحدة
M14	عدم استخدام transactions في بعض العمليات	API	عدم اتساق البيانات

منصة 8 ملاحظات

#	الملاحظة	العکون	التأثير
L1	23 عمود FK يقبل NULL	DB	مرونة زائدة
L2	للسداول الفارغة seed data لا يوجد	DB	لا بيانات تجريبية
L3	API documentation (Swagger) غياب	API	صعوبة التكامل
L4	health check endpoint غياب	API	صعوبة المراقبة
L5	graceful shutdown لا يوجد	خادم	فقدان بيانات محتمل
L6	مُعدّ connection pooling لا يوجد	DB	أداء تحت الحمل
L7	timestamp بدون تاريخ القيود يخزن كـ	DB	مشاكل المناطق الزمنية
L8	(سجل تدقيق) audit trail لا يوجد	نظام	لا تتبع للتغييرات

النوصيات ذات الأولوية.

المراحل الأولى: اصلاحات حادة (فهارس)

1. في (`server/index.ts`) إضافة `cors()` , `helmet()` , `g rateLimit()` (C2, C3, H1): تفعيل حزم الأمان
 2. إضافة JWT أو session-based authentication (C1): تنفيذ (C1)
 3. في `id` لجميع حقول (`$.defaultFn(() => nanoid())`) توحيد توليد المفاتيح (C4): إضافة `ID`
 4. لجميع نقاط API لتعريف `schemas` استخدام `zod`: إضافة التحقق من المدخلات (C6, C7): استخدم `zod` (C6, C7)
 5. بدلاً من كتابتها في (`DATABASE_URL`) استخدام متغيرات البيئة: نقل بيانات الاعتماد (C8) إلى الكود

المرحلة الثانية: إصلاحات عالية (خلال أسبوع)

- أو تعريف JOIN استخدام Drizzle Relations N+1 Query (H2, H3): إصلاح
 - إنشاء فهارس للأعمدة المرجعية (H5) إضافة

3. إصلاح أخطاء TypeScript (H6): خاصية OrganizationStructure.tsx
4. تفعيل منظم للأحداث والطلبات (H8, H9): Winston g Morgan
5. تحديد عدد السجلات في كل صفحة (H12): Pagination إضافة

المرحلة الثالثة: إصلاح البيانات (خلال أسبوعين)

1. التأكد من نقل جميع أسطر كل قيد (C5): إعادة ترحيل قيود اليومية
2. الأصناف، المخزون، التحويلات (M4): ترحيل الجداول المتبقية
3. تحديث أرصدة الحسابات عند إنشاء/تعديل القيود (H4): إضافة منطق تحديث الأرصدة
4. رفض القيود غير المتوازنة (M10): إضافة تحقق من توازن القيد

11. مصفوفة الامتثال

المعيار	الحالة	النسبة
سلامة البيانات (Data Integrity)	جزئي ⚠	40%
الأمان (Security)	ضعيف ✖	15%
الأداء (Performance)	مقبول ⚠	60%
جودة الكود (Code Quality)	مقبول ⚠	50%
التوثيق (Documentation)	ضعيف ✖	20%
الاختبارات (Testing)	غير موجود ✖	0%
تجربة المستخدم (UX)	جيد ✓	70%
البنية التحتية (Infrastructure)	جيد ✓	75%

التقييم العام: 41% - يحتاج تحسينات جوهرية

12. الخلاصة

النظام المحاسبي يعمل من الناحية الوظيفية الأساسية ويوفر واجهة مستخدم جديدة مع دعم سلية ومناسبة. البنية التحتية (React + Express + PostgreSQL + Drizzle) كاملة للغة العربية.

ومع ذلك، يعاني النظام من ثغرات أمنية حرجية (غياب المصادقة والتتفويض)، ومشاكل في سلامة يوصى بشدة N+1 Query). **البيانات المهاجرة** (98.4٪ من القيود غير متوازنة)، ومشاكل أداء بمعالجة الملاحظات الحرجية والعالية قبل استخدام النظام في بيئه إنتاجية.

نـم إـعـدـاد هـذـا التـقـرـير بـتـارـيـخ 13 فـبـرـاـيـر 2026 كـجـزـء مـن عـمـلـيـة الفـدـصـ التـقـنـي الشـامـلـ لـلـنـظـامـ .
المـحـاسـبـيـ