

تقرير الفحص التقني الشامل للنظام المحاسبي

التاريخ: 13 فبراير 2026

(Frappe Books Migration) النظام: نظام محاسبي متعدد الكيانات

المستودع: alabasi2025/222222222222

المراجع: فحص تكامل تقني آلي شامل

1. ملخص تنفيذي

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

2. نظرة عامة على البنية التقنية

المكون	التقنية	الإصدار
الواجهة 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	عبر localhost:5432
UI مكتبة	Radix UI + TailwindCSS	أحدث إصدار

المنافذ المستخدمة:

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

فحص قاعدة البيانات 3.

هيكل الجداول 3.1

Drizzle ORM تم إنشاء **16 جدولاً** في قاعدة البيانات 22222 وفقاً للمخطط المُعرّف في

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

إجمالي السجلات المُهَاجَرَة: 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

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

3.3 الفهارس (Indexes)

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

3.4 مشكلة حرجية: القيود غير المتوازنة

● **خطورة: حرجية**

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 يحتوي 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

يوفر نقاط النهاية التالية **route** تم تعريف **15 ملف**

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	بطيء 🟡

4.3 مشكلة حرجة: توليد المفاتيح الأساسية (ID Generation)

خطورة: حرجة 🚫

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

الاختبار:

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

لكن هذا النمط غير ، (`id-${Date.now()}`) يدوياً ID يولد `journalEntries.ts` ملاحظة: ملف `routes` موحد عبر جميع الـ.

4.4 في قيود اليومية N+1 Query: مشكلة

خطورة: عالية 🟡

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

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

لكل سطر، يجلب بيانات الحساب 3.

لكل قيد، يجلب بيانات الكيان 4.

واحد. هذا يفسر ببطء GET مع 1,273 قيد و1,286 سطر، هذا يعني ~3,800 استعلام لكل طلب (1,024ms) الاستجابة.

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) ✅	-
حماية XSS	غير محمي (يقبل HTML/Script) ❌	عالية
HTTPS	عبر Cloudflare Tunnel ✅	-

5.2 تفاصيل الثغرات الأمنية

5.2.1 غياب المصادقة والتفويض (درجة)

مفتوحة بالكامل لأي 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 محاسبية بدون أي تحقق من الهوية.

5.2.2 حزم الأمان مثبتة لكن غير مستخدمة (درجة)

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

- `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 XSS (عالية)

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

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

5.2.4 غياب التحقق من المدخلات (عالية)

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

6. فحص الواجهة الأمامية

6.1 هيكل الصفحات

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

الفئة	الصفحات
المحاسبة	ChartOfAccounts, JournalEntries, Payments, Ledger, AccountTypes, AccountSubtypes, AccountGroups
الخبزينة	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 TypeScript أخطاء

عند التحويل البرمجي TypeScript تم اكتشاف 109 خطأ

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

أثناء TypeScript يتجاوز أخطاء Vite ملاحظة: رغم وجود هذه الأخطاء، النظام يعمل لأن التطوير.

6.3 ملفات النسخ الاحتياطي

يحتوي على 4 ملفات نسخ احتياطي قديمة يجب تنظيفها pages/ داخل backup/ يوجد 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 مُفعَّل ❌	أخطاء 109
ESLint	مُعدَّ ✅	لم يُفحص
Prettier	مُعدَّ ✅	-
Husky (Git Hooks)	مُعدَّ ✅	-
اختبارات وحدة	غير موجودة ❌	مُعدَّ لكن بدون اختبارات Vitest
تسجيل الأحداث (Logging)	فقط console.log ⚠️	مُثبت لكن غير مستخدم Winston

7.2 أنماط مشكلة في الكود

7.2.1 AgentExecutionBridge يستخدم مسارات Windows

🟡 **خطورة: متوسطة**

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

7.2.2 Dashboard يستخدم بيانات وهمية

🟡 **خطورة: متوسطة**

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

7.2.3 غياب Drizzle Relations

● **خطورة: متوسطة**

مع `db.query` مما يمنع استخدام Drizzle ORM في مخطط `relations` لم يتم تعريف `routes` في بعض الـ `N+1 Query` للعلاقات. هذا يفسر استخدام نمط

7.2.4 عدم تحديث أرصدة الحسابات

● **خطورة: عالية**

عند إنشاء قيد يومية جديد، لا يتم تحديث 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
الجدول المُهاجرة	من 16 8
ترميز النص العربي	سليم ✓
السجلات اليتيمة	0 ✓

8.2 مشكلة حرجية: قيود اليومية ناقصة

● **خطورة: حرجية**

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 المُهاجرة غير صالحة للاستخدام المحاسبي في وضعها الحالي وتحتاج إلى إعادة ترحيل أو إصلاح.

8.3 جداول لم تُهاجر

الجدول 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)

9. جدول الملاحظات حسب الخطورة

● حرجة (8 ملاحظات)

#	الملاحظة	المكوّن	التأثير
C1	غياب المصادقة والتفويض	API	وصول غير مصرح به لجميع البيانات
C2	غير مُفعّل CORS	API	طلبات من أي نطاق
C3	غير مُفعّل 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	(HTML تخزين) XSS ثغرة	API	مخزنة XSS هجمات
C8	كلمة مرور القاعدة في الكود	DB	تسريب بيانات الاعتماد

● عالية (12 ملاحظة)

#	الملاحظة	المكوّن	التأثير
H1	Rate Limiting غير مُفعّل	API	هجمات DDoS/Brute Force
H2	N+1 Query في قيود اليومية	API	بطء شديد (1 ثانية)
H3	N+1 Query في المدفوعات	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	TypeScript أخطاء 109	كود	مخاطر وقت التشغيل
H7	غياب اختبارات الوحدة	كود	لا ضمان جودة
H8	مثبت لكن غير مستخدم Winston	كود	لا تسجيل منظم
H9	مثبت لكن غير مستخدم Morgan	كود	لا تسجيل طلبات
H10	فارغ payment_voucher_operations	بيانات	سندات بدون تفاصيل عمليات
H11	Drizzle Relations غياب	DB/ORM	استعلامات غير فعالة
H12	API في Pagination غياب	API	تحميل جميع السجلات دفعة واحدة

متوسطة (14 ملاحظة) ●

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

● منخفضة (8 ملاحظات)

#	الملاحظة	المكوّن	التأثير
L1	NULL يقبل FK عمود 23	DB	مرونة زائدة
L2	للجداول 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	connection pooling مُعدّ لا يوجد	DB	أداء تحت الحمل
L7	timestamp بدون timezone تاريخ القيود يُخزن كـ	DB	مشاكل المناطق الزمنية
L8	audit trail (سجل تدقيق) لا يوجد	نظام	لا تتبع للتغييرات

10. التوصيات ذات الأولوية

المرحلة الأولى: إصلاحات حرجية (فورية)

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

المرحلة الثانية: إصلاحات عالية (خلال أسبوع)

1. **Drizzle Relations** أو تعريف JOIN استخدام (H2, H3) **N+1 Query** إصلاح
2. **إضافة فهارس للأعمدة المرجعية** (H5) إنشاء فهارس لـ 38 عمود

3. **TypeScript** (H6) إصلاح أخطاء
4. **Winston g Morgan** (H8, H9) تسجيل منظم للأحداث والطلبات
5. **Pagination** (H12) تحديد عدد السجلات في كل صفحة

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

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

التقييم العام: 41% - يحتاج تحسينات جوهرية

12. الخلاصة

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

ومع ذلك، يعاني النظام من **ثغرات أمنية حرجية** (غياب المصادقة والتفويض)، و**مشاكل في سلامة** يُوصى بشدة (N+1 Query) **البيانات المُهاجرة** (98.4% من القيود غير متوازنة)، و**مشاكل أداء** بمعالجة الملاحظات الحرجة والعالية قبل استخدام النظام في بيئة إنتاجية.

نم إعداد هذا التقرير بتاريخ 13 فبراير 2026 كجزء من عملية الفحص التقني الشامل للنظام المداسبي.