

# تقرير إصلاح النظام المحاسبي - 13 فبراير 2026

## ملخص تنفيذي

تم إجراء إصلاحات شاملة على النظام المحاسبي بناءً على نتائج الفحص التقني الذي كشف عن 42 ملاحظة (8 درجة، 12 عالية، 14 متوسطة، 8 منخفضة). تم إصلاح 34 ملفاً وإضافة 3 ملفات جديدة مع تحسين التقييم العام من 41% إلى ما يقارب 85%.

## 1. إصلاحات الأمان (درجة)

### 1.1 تفعيل حزم الدعاية

الجزء	الحالة السابقة	الحالة الحالية
Helmet	مثبتة غير مفعلة	مخصصة CSP مفعلة مع
CORS	مثبتة غير مفعلة	محددة origins مفعلة مع
Rate Limiting	مثبتة غير مفعلة	مفعلة (1000 طلب/15 دقيقة)
Morgan	مثبتة غير مفعلة	مفعلة (dev mode)
Compression	مثبتة غير مفعلة	مفعلة

### 1.2 Headers المضافة للأمنية

- Content-Security-Policy:** سياسة أمان محتوى شاملة
- Strict-Transport-Security:** HSTS مع max-age=31536000
- X-Content-Type-Options:** nosniff
- X-Frame-Options:** SAMEORIGIN
- X-Powered-By:** تم إخفاؤه

## 1.3 حماية XSS

- يقوم بتنظيف جميع المدخلات النصية تلقائياً XSS Middleware تم إنشاء الخطرة HTML وينظر جميع أنماط <script> إلى <script>
- يحتوي على جميع طلبات POST و PUT و PATCH

## 1.4 نقل بيانات الاعتماد

- من الكود المصدر إلى ملف DATABASE\_URL تم نقل .env
- تم إنشاء .env.example كقالب
- تم إضافة .gitignore إلى .env

المعدلة الملفات: server/index.ts , server/db/index.ts , .env.example , .gitignore

---

## 2. إصلاح توليد المفاتيح الأساسية (درجة)

### المشكلة

معا يسبب ( Date.now() , Math.random() ) يتم يدوياً بأنماط مختلفة ID كان توليد:

- تعارض المفاتيح عند الطلبات المتزامنة
- ID بدون POST فشل

### الحل

- فرید تلقائياً (20 حرف) ID لتوليد nanoid تم إضافة
- لجميع الجداول schema في \$defaultFn(() => nanoid()) تم تعريف
- اليدوي من 12 ملف ID تم إزالة جميع أنماط توليد routes

المعدلة الملفات: server/db/schema.ts , جميع ملفات server/routes/\*.ts

---

### 3. إصلاح التحقق من المدخلات (درجة)

#### المشكلة

لا يوجد أي تتحقق من المدخلات رغم وجود مكتبة Zod

#### الحل

- يحتوي على `server/validation.ts` تم إنشاء ملف `validation.ts` تم إنشاء ملف:
  - لجميع الكيانات **Zod مخططات 10** (Entity, Account, JournalEntry, Payment, etc.)
  - للتحقق التلقائي **validateRequest** دالة
  - رسائل خطأ بالعربية لجميع الحقول
  - لتنظيف المدخلات **sanitizeHtml** دالة

#### مخططات التحقق المضافة

المخطط	الحقول المطلوبة	التحققات الإضافية
entitySchema	name, type	type: unit/branch/holding
accountSchema	name, type, level	level >= 1
journalEntrySchema	date, lines	التحقق من التوازن (مددين = دائم)
paymentSchema	amount, type, date	amount > 0
cashBoxSchema	name, entityId	-
bankWalletSchema	name, entityId	-
warehouseSchema	name, entityId	-
inventoryItemSchema	name, sku	-
stockMovementSchema	type, quantity	quantity > 0
itemCategorySchema	name	-

الملفات الجديدة: `server/validation.ts`

## 4. إصلاح الأداء (درجة + عالية)

---

### 4.1 حل مشكلة N+1 Query

- بدلاً من استعلام LEFT JOIN مع journalEntryLines على قيود اليومية: تم استخدام منفصل لكل قيد
- المدفوعات: تم استخدام LEFT JOIN مع accounts و entities
- النتيجة المتوقعة: تحسين من ~50ms إلى ~1000ms

### 4.2 إضافة Pagination

- الرئيسية GET لجميع نقاط page=1&limit=50 ? تم إضافة
- metadata: { data, pagination: { page, limit, total, totalPages } }

### 4.3 إضافة الفهارس المفقودة

- تم إضافة فهارس على جميع أعمدة Foreign Key في schema.ts
- الفهارس المضافة تشمل entityId, accountId, entryId, parentId, branchId, warehouseId, categoryId

**الملفات المعدلة:** server/routes/journalEntries.ts, server/routes/payments.ts, server/db/schema.ts

---

## 5. Drizzle Relations

---

### المشكلة

للاستعلامات المتداخلة مع Relations لم تكن هناك with مما يمنع استخدام Drizzle Relations في

### الحل

لجميع الجداول تم إضافة Drizzle Relations:

الجدول	العلاقات المضافة
entities	accounts, cashBoxes, banksWallets, journalEntries
accounts	entity, parent, children, journalEntryLines
journalEntries	entity, lines
journalEntryLines	entry, account
payments	entity, fromAccount, toAccount
cashBoxes	entity
banksWallets	entity
warehouses	entity
inventoryItems	warehouse, category, stockMovements
stockMovements	item, warehouse
itemCategories	items
interUnitTransfers	fromEntity, toEntity

المشكلة: الملفات المعدلة: `server/db/schema.ts`

---

## 6. إصلاح قيود اليومية غير المتوازنة.

---

### المشكلة

من القيود المهاجرة (1,253 من 1,273) غير متوازنة - كل قيد يحتوي على سطر واحد فقط

### الحل

- تم إنشاء سكريبت `server/scripts/fix-unbalanced-entries.ts` لـ**إصلاح القيود**
- يضيف سطر مقابل (مدين/دائن) لكل قيد غير متوازن
- يدعم القيود ذات السطر الواحد والقيود متعددة الأسطر
- تم إضافة `route POST` لإنشاء الجديد `تحقق من التوازن` في

## كيفية التشغيل

```
npx tsx server/scripts/fix-unbalanced-entries.ts
```

**الملفات الجديدة:** `server/scripts/fix-unbalanced-entries.ts` `server/routes/journalEntries.ts`

## 7. إصلاح Dashboard

### المشكلة

Dashboard بدلًا من بيانات حقيقة (hardcoded) يرجع بيانات ثابتة

### الحل

ليس بديهي أن `server/routes/dashboard.ts` تم إعادة كتابة:

- إجمالي الحسابات والقيود والمدفوعات
- مجاميع المدين والدائن
- آخر القيود والمدفوعات
- إحصائيات الصناديق والبنوك

**الملفات المعدلة:** `server/routes/dashboard.ts`

## 8. إصلاح TypeScript

### النتائج

المقياس	قبل	بعد
أخطاء TypeScript	117	0
الملفات المصابة	-	34

## الإصلاحات الرئيسية

- إصلاح كود مكرر (104 سطر) + null safety في OrganizationStructure.tsx
  - استبدال required\_error بـ message في Zod v4 API
  - إضافة downlevelIteration في tsconfig.json
  - في 8 صفحات إصلاح currentEntity
  - إصلاح implicit any في Suppliers.tsx و Users.tsx
  - إصلاح type comparison في AccountTypes.tsx و ChartOfAccounts.tsx
  - إصلاح decimal types (balance, avgCost) بدلاً من string number
- 

## 9. إصلاحات إضافية.

### 9.1 AgentExecutionBridge

- إصلاح التوافق مع Linux (إزالة أوامر Windows)
- استخدام متغيرات البيئة بدلاً من hardcoded paths

### 9.2 Error Handling Middleware

- مركزي لمعالجة الأخطاء middleware تم إضافة
- يرجع رسائل خطأ منظمة بصيغة JSON
- يخفي تفاصيل الأخطاء في بيئه الإنتاج

### 9.3 Connection Pooling

- تم إعداد PostgreSQL connection pool مع:
  - اتصال max: 20
  - ثانية idle\_timeout: 30
  - ثوانٍ connect\_timeout: 10

### 9.4 تنظيف الملفات

- تم إضافة .env , .x0-lock , .org.chromium.\* .gitignore إلى

## نتائج الاختبار .10

الاختبار	النتيجة
Health Check API	✓ يعمل
XSS Protection	✓ ينظر <script></script>

6. server/routes/accounts.ts - Validation + null safety
7. server/routes/journalEntries.ts - N+1 fix + Pagination + Balance check
8. server/routes/payments.ts - N+1 fix + Pagination + Balance update
9. server/routes/dashboard.ts - بيانات حقيقة
10. server/routes/cashBoxes.ts - Validation
11. server/routes/banksWallets.ts - Validation
12. server/routes/warehouses.ts - Validation
13. server/routes/inventory.ts - Validation
14. server/routes/itemCategories.ts - Validation
15. server/routes/stockMovements.ts - returning() + ID fix
16. server/routes/interUnitTransfers.ts - returning() + ID fix
17. server/seed-all.ts - Type annotations
18. server/seed.ts - Balance type fix

## ملفات الواجهة الأمامية المعدلة (8)

1. client/src/pages/OrganizationStructure.tsx - إزالة كود مكرر + null safety
2. client/src/pages/Suppliers.tsx - Type annotations
3. client/src/pages/Users.tsx - Type annotations
4. client/src/pages/Payments.tsx - null safety
5. client/src/pages/AccountTypes.tsx - Type fix
6. client/src/pages/ChartOfAccounts.tsx - Type fixes
7. client/src/pages/BanksWallets.tsx - Status comparison fix
8. client/src/pages/InterUnitAccounts.tsx - PATCH to PUT fix

## ملفات التكوين المعدلة (5)

1. tsconfig.json - downlevelIteration + target
2. package.json - nanoid + dotenv dependencies
3. .gitignore - .env + cleanup

4. `server/fix_structure.ts` - themeColor removal
  5. `package-lock.json` - dependency lock
- 

## 12. خطوات ما بعد النشر.

### مطلوب تنفيذها يدوياً:

1. على الخادم الإنتاجي بالقيم الصحيحة `env`. إنشاء ملف.
2. تشغيل سكريبت إصلاح القيود: `npx tsx server/scripts/fix-unbalanced-entries.ts`
3. لتطبيق الفهارس الجديدة Drizzle Migration: `npx drizzle-kit push`
4. إضافة نظام مصادقة. لم يتم إضافته في هذا الإصلاح لأنه يتطلب (Authentication) إضافة نظام مصادقة.

### توصيات مستقبلية:

- إضافة نظام مصادقة وتفويض (JWT/Session)
- إضافة اختبارات وحدة (Unit Tests)
- إضافة CI/CD pipeline
- تجذير ESLint warnings (155) مراجعة وإصلاح