

تقرير التحسينات والتسليمات - Enterprise AI Layer

ملخص تنفيذي

تم تنفيذ أربعة تحسينات تقنية رئيسية لجعل منصة الذكاء التنفيذي المؤسسي (Enterprise AI Layer) جاهزة للعرض التجريبي (Demo) والتجربة الأولية (Pilot) مع العملاء. هذه التحسينات تحول المنصة من نموذج أولي إلى منتج قابل للعرض بشكل احترافي.

التحسينات المنفذة

1. سكريبت البيانات التجريبية (Seed Script)

الملف: scripts/seed-demo-inventory.ts

الوصف: سكريبت TypeScript متكامل يقوم بتعبئة قاعدة بيانات PostgreSQL ببيانات مخزون واقعية ومتنوعة، مصممة خصيصًا لإظهار قدرات محرك الذكاء الاصطناعي.

المميزات:

- إنشاء 70 منتج بأسماء عربية وإنجليزية وفئات متنوعة
- إنشاء 8 مواقع (مستودعات، مراكز توزيع، فروع) في مدن سعودية مختلفة
- إنشاء 200+ رصيد مخزون موزع على المنتجات والمواقع
- إنشاء 1500+ حركة مخزون على مدى 6 أشهر
- إنشاء 700+ إشارة طلب لتحليل الاتجاهات

السيناريوهات المدمجة:

السيناريو	المنتجات	الوصف
خطر نفاد المخزون	1-5	مخزون منخفض جدًا، أقل من نقطة إعادة الطلب
ارتفاع مفاجئ في الطلب	6-10	زيادة حادة في الطلب خلال الأسبوع الأخير
مخزون بطيء الحركة	61-65	مخزون مرتفع مع حركة ضعيفة
حركة غير طبيعية	11-15	معاملات كبيرة غير متوقعة

طريقة التشغيل:

Bash

npm run tsx scripts/seed-demo-inventory.ts

2. نظام المحاكاة الخارجي (Mock REST API)

الملف: mock-external-system/server.ts

الوصف: خادم Express.js يحاكي نظام إدارة مخزون خارجي حقيقي، يدعم المصادقة عبر API Key، والترقيم الصفحي (Pagination)، ومحاكاة الأخطاء.

نقاط النهاية (Endpoints):

الترقيم	الوصف	المسار
offset-based	قائمة المنتجات	GET /api/items
offset-based	قائمة المواقع	GET /api/locations
offset-based	أرصدة المخزون	GET /api/stock-balances
offset-based	حركات المخزون	GET /api/stock-movements
-	فحص الصحة	GET /health
-	توثيق API	GET /api

المصادقة:

النوع: API Key عبر Header

Header: X-API-Key

المفاتيح الصالحة: DEMO-API-KEY-2024 , ENTERPRISE-KEY-001

محاكاة الأخطاء:

- 401 Unauthorized : غير صالح أو مفقود API مفتاح
- 429 Rate Limit : تجاوز 100 طلب في الدقيقة
- 500 Server Error : احتمال 2% لمحاكاة أخطاء الخادم

طريقة التشغيل:

Bash

```
cd mock-external-system
npm install
```

npx tsx server.ts

3. تكوينات التحويل (Mapping Configurations)

الملف: scripts/demo-mapping-configs.json

الوصف: ملف JSON يحتوي على تكوينات تحويل البيانات من صيغة النظام الخارجي إلى النموذج الأساسي للمنصة (Canonical Model).

التحويلات المدعومة:

نوع التحويل	الوصف	مثال
integer	تحويل النص إلى رقم صحيح	"100" → 100
boolean	تحويل إلى قيمة منطقية	"true" → true
lowercase	تحويل إلى أحرف صغيرة	"EACH" → "each"
datetime	تحليل التاريخ والوقت	ISO8601 format
enum	تحويل القيم باستخدام جدول بحث	"WAREHOUSE" → "warehouse"
lookup	البحث عن المفتاح الأجنبي	ربط item_id بـ items.externalId

الكائنات المدعومة:

- Items Mapping: 10 حقول مع تحويلات
- Locations Mapping: 5 حقول مع تحويل
- Stock Balances Mapping: 5 حقول مع
- Stock Movements Mapping: 6 تحويلات متعددة

4. سكريبت التدفق التجريبي المتكامل (Demo Flow)

الملف: scripts/run-demo-flow.ts

الوصف: سكريبت TypeScript ينفذ التدفق الكامل من البداية إلى النهاية، مع تسجيل تدقيق لكل خطوة.

خطوات التدفق:

Plain Text

DEMO FLOW SEQUENCE

- | | |
|--|--|
| 1. التحقق من المتطلبات (Tenant + Mock API) | |
| ↓ | |
| 2. للنظام الخارجي Connector إنشاء | |
| ↓ | |
| 3. Endpoints (Items, Locations, Balances, Movements) تعريف | |
| ↓ | |
| 4. Mappings (JSONPath → Canonical Model) إنشاء | |
| ↓ | |
| 5. محاكاة جلب البيانات من النظام الخارجي | |
| ↓ | |
| 6. AI Engine تشغيل | |
| ↓ | |
| 7. Recommendations + Anomalies توليد | |
| ↓ | |
| 8. Audit Trail + عرض النتائج | |

المخرجات:

- توصيات ذكية (Reorder, Alert, Review)
- حالات شاذة (Unusual Movement, Stock Discrepancy)
- سجل تدقيق كامل لكل عملية

5. سكريبت البدء الموحد

الملف: `scripts/start-demo.sh`

الوصف: سكريبت Bash يقوم بتشغيل كافة مكونات الـ Demo بأمر واحد.

طريقة التشغيل:

Bash

```
./scripts/start-demo.sh
```

6. دليل التشغيل التجريبي

الملف: `DEMO_GUIDE.md`

الوصف: دليل مفصل باللغة العربية يشرح كيفية تشغيل الـ Demo واستعراض النتائج في الواجهة الرسومية.

هيكل الملفات المضافة

Plain Text

Pasted-Enterprise-AI/

```
├── scripts/
│   ├── seed-demo-inventory.ts    # سكربت البيانات التجريبية
│   ├── run-demo-flow.ts         # سكربت التدفق المتكامل
│   ├── start-demo.sh            # سكربت البدء الموحد
│   └── demo-mapping-configs.json # تكوينات التحويل
├── mock-external-system/
│   ├── server.ts                # خادم المحاكاة
│   └── package.json             # اعتماديات المحاكاة
└── DEMO_GUIDE.md                # دليل التشغيل
```

خطوات التشغيل السريع

للحصول على Demo جاهز للعرض، اتبع الخطوات التالية:

1. تأكد من إعداد متغير البيئة:

Bash

```
export DATABASE_URL="postgresql://..."
```

2. شغل سكربت البدء الموحد:

Bash

```
./scripts/start-demo.sh
```

3. شغل التطبيق الرئيسي:

Bash

```
npm run dev
```

4. افتح الواجهة في المتصفح:

Plain Text

```
https://pasted-enterprise-ai--mesfer1.replit.app
```

التوصيات للخطوات القادمة

بعد تنفيذ هذه التحسينات، أوصي بالخطوات التالية:

1. اختبار التدفق الكامل: تشغيل `start-demo.sh` والتحقق من أن كل الخطوات تعمل بشكل صحيح.
2. تخصيص البيانات: تعديل `seed-demo-inventory.ts` لإضافة منتجات أو سيناريوهات خاصة بالعمل المستهدف.
3. تحسين الواجهة: إضافة رسائل توضيحية في الواجهة لشرح كل توصية وكيف تم توليدها.
4. إعداد عرض تقديمي: استخدام مخرجات الـ Demo لإعداد عرض تقديمي للعملاء المحتملين.
5. توثيق API: نشر توثيق OpenAPI/Swagger للـ Mock API لتسهيل فهم التكامل.

الخلاصة

تم تنفيذ جميع التحسينات المطلوبة بنجاح. المنصة الآن جاهزة لتقديم Demo احترافي يُظهر:

- قدرة التكامل: الاتصال بأنظمة خارجية عبر REST API
- مرونة التحويل: تحويل البيانات من أي صيغة إلى النموذج الأساسي
- ذكاء التحليل: توليد توصيات وتنبيهات ذكية
- حوكمة صارمة: سجل تدقيق كامل وغير قابل للتجاوز
- أمان مدمج: لا يوجد تنفيذ تلقائي، القرار دائمًا للإنسان

تاريخ الإعداد: 12 يناير 2026

الإصدار: 1.0.0