نظام المتجر المؤسسي المتكامل

وثيقة التخطيط والتحليل - الإصدار 1.0

م/ زكريا الماوري

Lead Software Engineer / System Architect

13 يناير 2026

مقدمة المشروع والرؤية الاستراتيجية

حجر الزاوية في SDLC

تعد مرحلة التخطيط والتحليل الأساس المتين لضمان نجاح دورة حياة تطوير النظام.

حل تقني متقدم

نهدف لبناء منصة قابلة للتوسع تلبي احتياجات السوق المتغيرة وتدعم النمو المستقبلي.

تجربة مستخدم استثنائية

تأسيس بنية تحتية رقمية قوية توفر تجربة تسوق سلسة ومتكاملة للعملاء والمديرين.

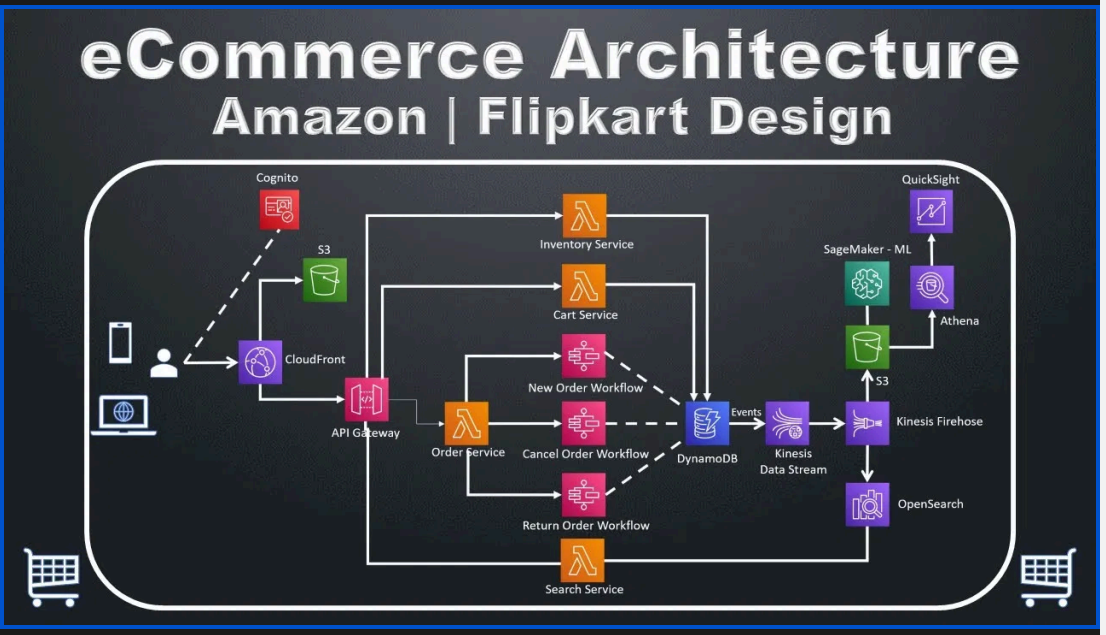
نطاق المشروع والمستهدفون

بناء منصة Single-Tenant Enterprise لخدمة الكيانات الكبرى

- **نوع النظام:** استضافة داخلية أو سحابية خاصة (Private Cloud)
- **التركيز:** الأداء العالي، مرونة القواعد، والتكامل السلس
- **النموذج:** تخصيص التجربة لمتجر واحد ضخم (Single-Tenant)

مديرو النظام | العملاء | مديرو المخزون

Architecture Overview



المتطلبات الوظيفية: الكتالوج والمبيعات

إدارة الكتالوج (Catalog)

- **المنتجات:** دعم الأنواع المادية، الرقمية، والخدمات.
- **الخيارات والسمات:** نظام ديناميكي للمتغيرات المعقدة.
- **التصنيفات:** هيكل شجري غير محدود المستويات.

المبيعات والطلبات (Sales)

- **سلة الشراء:** تحقق لحظي من المخزون والأسعار.
- **إدارة الطلبات:** دورة حياة كاملة من الطلب حتى التسليم.
- **محرك القواعد:** قيود متطورة بناءً على الموقع والقيمة.

المتطلبات الوظيفية: المخزون والتسويق

إدارة المخزون

- تتبع دقيق للكميات لحظياً
- تنبيهات آلية عند انخفاض المخزون
- سجل حركات كامل (Audit Log)

إدارة العملاء

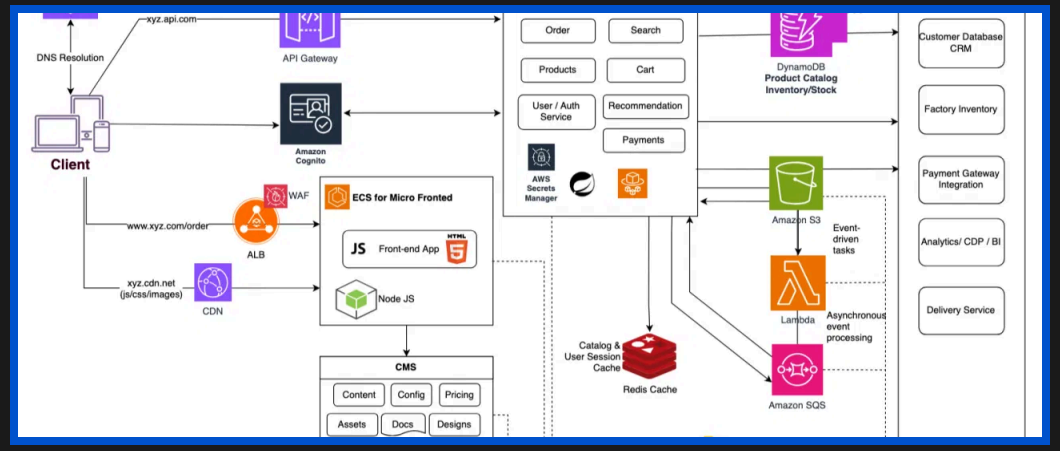
- ملفات شخصية وسجل طلبات شامل
- تصنيف المجموعات (VIP، جملة)
- تطبيق قواعد أسعار مخصصة

التسويق الذكي

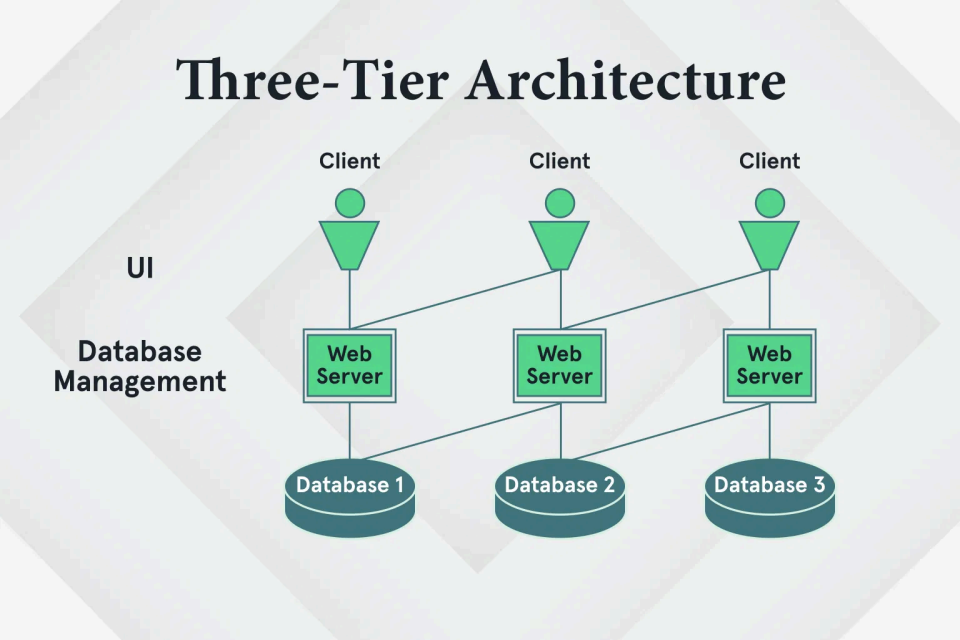
- محرك كوبونات مرن ومتعدد الشروط
- عروض ترويجية تلقائية
- خصومات بناءً على سلوك العميل

تكامل المكونات

يتم الربط بين وحدات المخزون والعملاء والتسويق لضمان تدفق البيانات بسلاسة، مما يسمح باتخاذ قرارات تجارية مبنية على بيانات دقيقة ولحظية.



المتطلبات غير الوظيفية والجودة



الأداء (Performance)

زمن استجابة API أقل من 200ms ودعم آلاف الاتصالات المتزامنة باستخدام Async I/O.

القابلية للتوسع (Scalability)

معمارية Modular Monolith تسمح بالتحول إلى Microservices مستقبلاً دون إعادة بناء النظام.

الأمان (Security)

تشفير Bcrypt/Argon2، توثيق OAuth2/JWT، وحماية ضد ثغرات SQLi و XSS و CSRF.

الصيانة (Maintainability)

الالتزام بمبادئ Clean Code و SOLID مع توثيق شامل باستخدام Swagger/OpenAPI.

الجدوى التقنية وحزمة البرمجيات

اختيار التقنيات بناءً على الأداء والموثوقية

Python 3.10+

بيئة غنية بالمكتبات، دعم ممتاز للذكاء الاصطناعي، وسهولة الصيانة والقراءة.

FastAPI

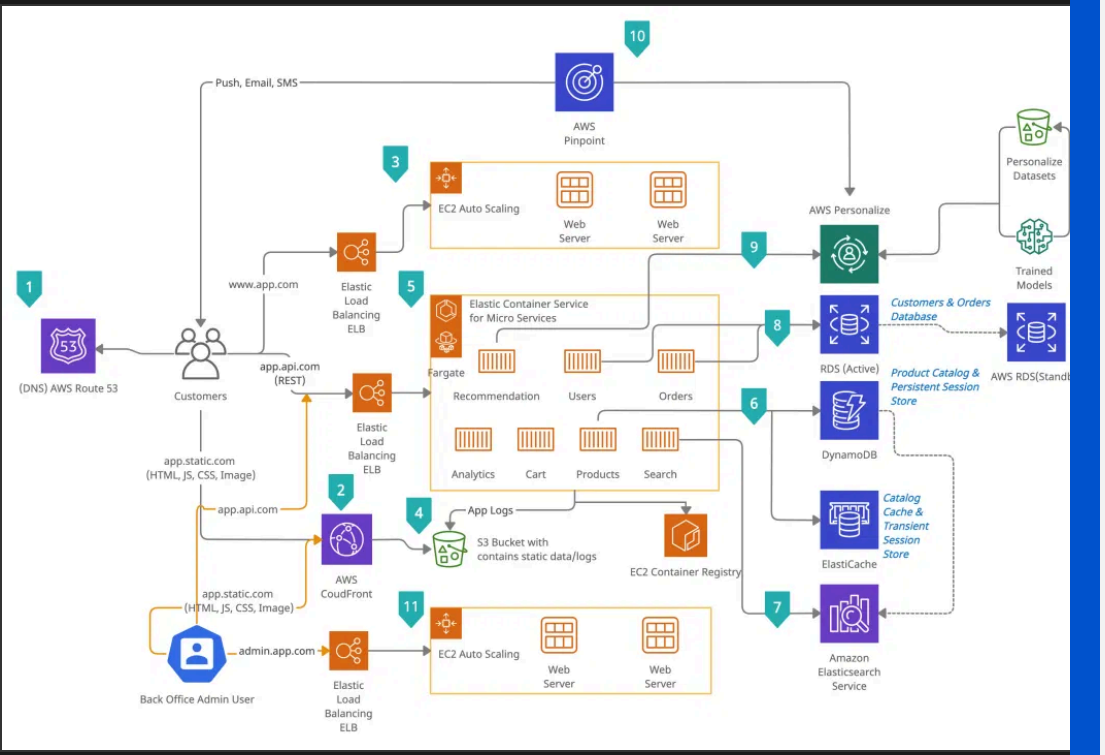
أسرع أطر عمل Python، يدعم البرمجة غير المتزامنة (Async) ويوفر توثيقاً تلقائياً.

PostgreSQL

موثوقية عالية مع SQLAlchemy Async، وأداء ممتاز في العمليات العلائقية المعقدة.

Hybrid Frontend

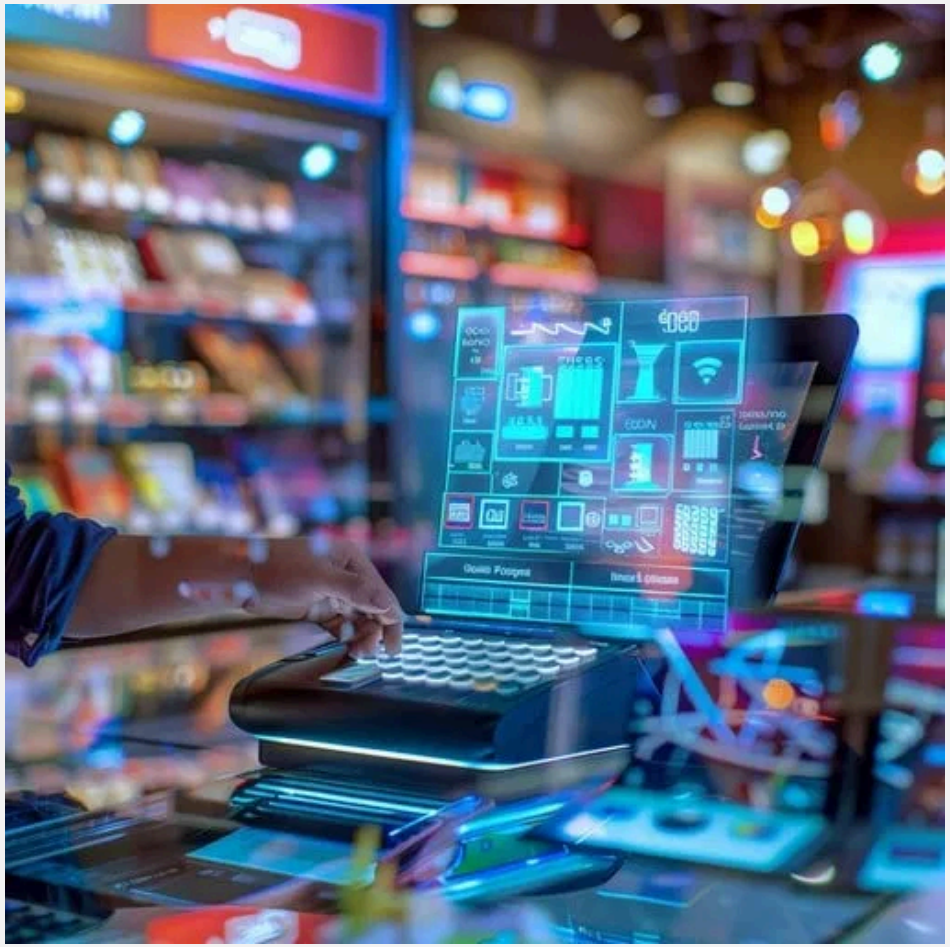
استخدام Jinja2 لسرعة التحميل و SEO، مع Vanilla JS للتفاعلية دون تعقيد البناء.



معمارية النظام المقترحة

تحليل المخاطر واستراتيجيات التخفيف

الخطر	الاحتمالية	التأثير	استراتيجية التخفيف
تعقيد قواعد العمل	مرتفع	مرتفع	استخدام محرك قواعد مرن (Condition Engine) واختبارات مكثفة لكل سيناريو لضمان دقة التنفيذ.
مشاكل الأداء مع البيانات	متوسط	مرتفع	تطبيق التصحيف (Pagination) في كل الواجهات، واستخدام الفهارس (Indexes) المتقدمة في قاعدة البيانات.
تغيير المتطلبات	مرتفع	متوسط	اعتماد منهجية Agile واستخدام التصميم المعياري (Modular Design) ليسهل التعديل دون التأثير على النظام ككل.



الخاتمة والقيمة المضافة

نقلة نوعية في الإدارة

يمثل هذا النظام حلاً متكاملاً لإدارة العمليات التجارية للمتاجر الكبرى بكفاءة عالية.

أصل رقمي استراتيجي

نحن لا نبني مجرد برمجية، بل نؤسس بنية تحتية رقمية قوية قادرة على استيعاب النمو المستقبلي.

إعداد: م/ زكريا الماوري - 2026