

وثيقة تصميم النظام

System Design Specification

اسم المشروع: نظام المتجر المؤسسي المتكامل

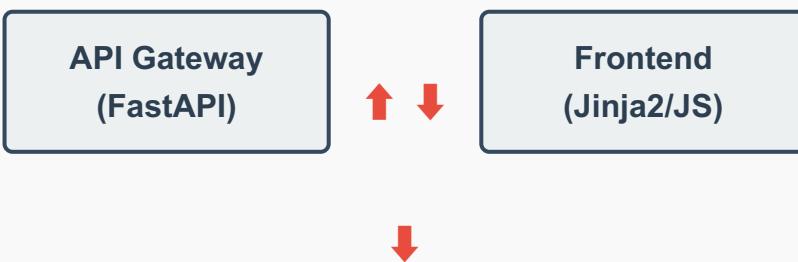
الإصدار: 1.0

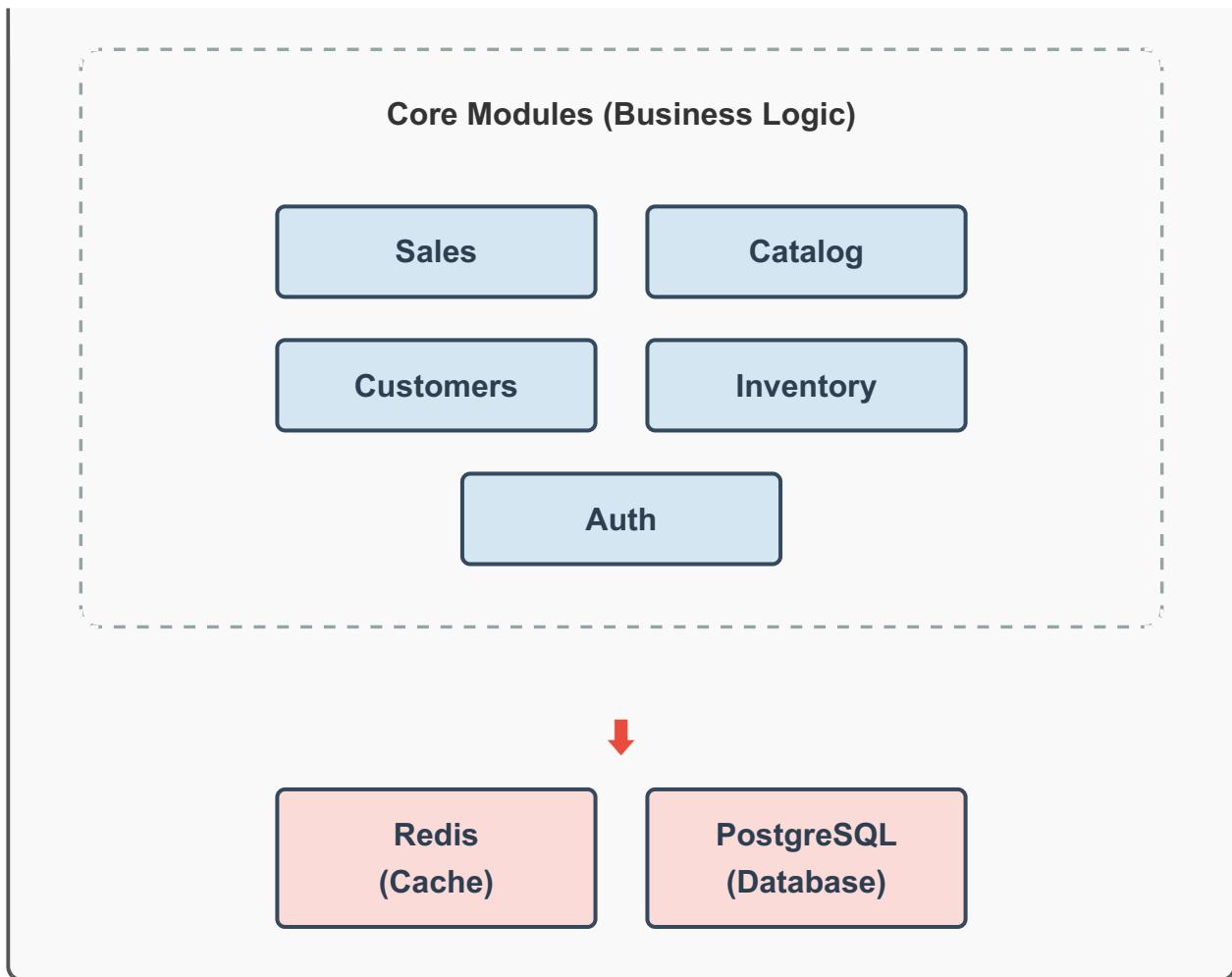
التاريخ: 13 يناير 2026

1. هندسة النظام (System Architecture)

لقد قمت باعتمار معمارية **Modular Monolith** لتصميم هذا النظام. هذا الخيار هو الأنسب حالياً لأنه يجمع بين بساطة النشر (Deployment) وقوة الفصل المنطقي (Logical Separation) التي تمهد الطريق للخدمات المصغرة (Microservices) في المستقبل.

المخطط المعماري للنظام





2. تصميم قاعدة البيانات (Database Schema - ERD)

لقد قمت بتصميم المخطط العلائقى (ERD) لضمان تكامل البيانات وسرعة الاستعلام. فيما يلي أهم الجداول والعلاقات:

أ. الكاتالوج والمنتجات (Catalog Module)

المنتج هو قلب النظام. العلاقة بين المنتج والخيارات (Variants) هي 1:N.

أهم الأعمدة (Columns)	الوصف	الجدول (Table)
id, name, slug, product_type, category_id, brand_id	الجدول الرئيسي للمنتجات	products
id, product_id, sku, price, stock_qty, attributes (JSON)	المتغيرات (مثل: قميص أحمر مقاس (L))	product_variants
id, name, parent_id, lft, rgt (for tree traversal)	التصنيفات (شجري)	categories

ب. المبيعات والطلبات (Sales Module)

يرتبط الطلب بالعميل (N:1) وبالمنتجات عبر جدول وسيط (OrderItems).

الجدول (Table)	الوصف	أهم الأعمدة (Columns)
orders	رأس الطلب	id, user_id, status, total_amount, shipping_address_id
order_items	تفاصيل المنتجات في الطلب	id, order_id, variant_id, quantity, unit_price, total
cart_items	سلة الشراء المؤقتة	id, session_id, variant_id, quantity

ج. المخزون (Inventory Module)

الجدول (Table)	الوصف	أهم الأعمدة (Columns)
inventory_transactions	سجل حركات المخزون	,id, variant_id, quantity_change, type (IN/OUT) reference_id

3. خوارزميات النظام (Core Algorithms)

لضمان دقة العمليات، قمت بتصميم الخوارزميات التالية للتعامل مع العمليات الحساسة:

أ. خوارزمية خصم المخزون (Inventory Deduction)

```
FUNCTION ProcessOrder(Order order):
    START TRANSACTION

    FOR EACH item IN order.items:
        # 1. Lock Row for Update (تأمين الصف لمنع التضارب)
        current_stock = SELECT stock FROM variants
                      WHERE id = item.variant_id
                      FOR UPDATE

        # 2. Check Availability
        IF current_stock < item.quantity:
            ROLLBACK
            RETURN Error("Out of Stock: " + item.product_name)

        # 3. Deduct
        UPDATE variants
        SET stock = stock - item.quantity
```

```

WHERE id = item.variant_id

# 4. Log Transaction
INSERT INTO inventory_transactions (
    variant_id, qty, type, ref_order
) VALUES (
    item.variant_id, -item.quantity, 'SALE', order.id
)
END FOR

COMMIT TRANSACTION
RETURN Success

```

ب. خوارزمية مطابقة القيود (Constraint Matching Engine)

تستخدم هذه الخوارزمية لتحديد ما إذا كانت طريقة الدفع/الشحن متاحة للعميل الحالي.

```

FUNCTION EvaluateConstraints(Constraints list, Context ctx):
    ValidOptions = []

    FOR EACH constraint IN list:
        IsMatch = True

        FOR EACH condition IN constraint.conditions:
            # Check Condition Type
            IF condition.type == 'CART_TOTAL':
                IF ctx.cart_total < condition.min OR ctx.cart_total >
                condition.max:
                    IsMatch = False
                    BREAK

            IF condition.type == 'LOCATION':
                IF ctx.user_city NOT IN condition.allowed_cities:
                    IsMatch = False
                    BREAK

            # ... check other conditions

    END FOR

    IF IsMatch == True:
        ValidOptions.add(constraint.target_option)

RETURN ValidOptions

```

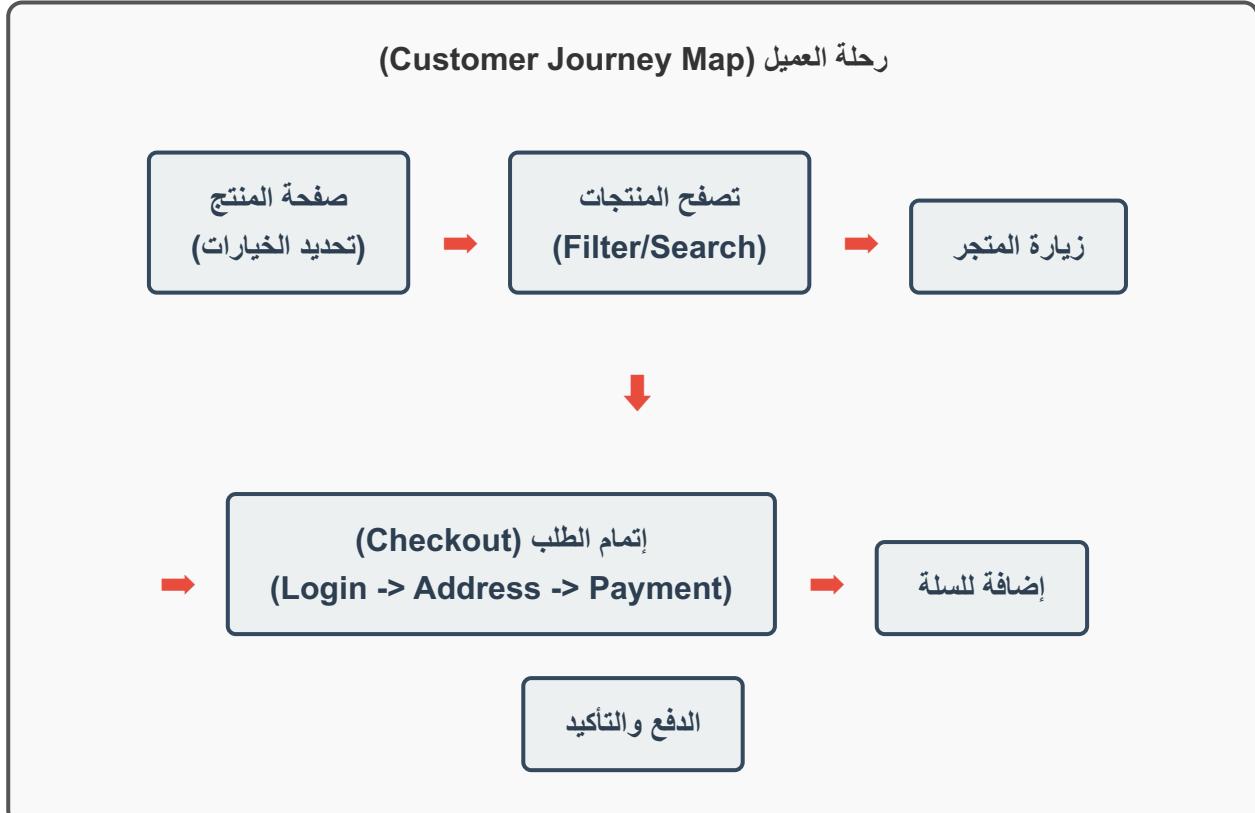
4. تصميم واجهة برمجة التطبيقات (API Design)

لقد قمت بتصميم نقاط النهاية (Endpoints) وفق معايير RESTful لضمان سهولة الاستخدام والتكامل:

الوصف	المسار (Endpoint)	الطريقة (Method)
جلب قائمة المنتجات (مع دعم التصفية والبحث)	api/products/	GET
تفاصيل منتج معين شاملة المتغيرات	api/products/{id}/	GET
إضافة منتج للسلة	api/cart/items/	POST
إنشاء طلب جديد (Checkout)	api/orders/	POST
تتبع حالة الطلب	api/orders/{id}/	GET
جلب قيود الدفع المتأصلة	api/settings/constraints/payment/	GET

5. رحلة المستخدم وتصميم الواجهة (UI/UX Journey)

رحلة العميل (Customer Journey Map)



إعداد وتصميم:

م/ زكريا الماورى
Lead System Architect