

Last Version

Giga Store Project - Comprehensive Change Log

نظرة عامة (Overview)

الوثيقة دي بتوضح كل التعديلات اللي حصلت في منصة التجارة الإلكترونية Giga Store من إصدار GitHub اللي فات لحد التطبيق الحالي. التغييرات بتشمل Architecture, وFeatures, وUI/UX, وتنظيم Code.

تغييرات في المعمارية والهيكل (Architecture & Structure Changes)

1. تنظيم جديد للملفات (New File Organization)

```
NEW FILES ADDED: (ملفات جديدة تمت إضافتها)
/assets/js —|
checkout.js # extracted checkout logic —| |
(منطق ال checkout تم استخراج)
main.js # consolidated cart & search —| |
(تجميع منطق ال cart وال search)
theme.js # Theme toggle system —| |
(نظام تبديل ال Theme)
/assets/css —|
theme.css # Theme variables —| |
(متغيرات ال Theme)
pages.css # Shared page styles —| |
(ستايلات مشتركة للصفحات)
REMOVED FILES: (ملفات تم حذفها)
assets/js/index.js # Merged into main.js —|
(تم دمج محتواه في main.js)
```

تطبيق نظام ال Theme (Theme System Implementation)

خاصية جديدة: الوضع الداكن/المضيء (New Feature: Dark/Light Mode)

الملفات المعدلة (Files Modified):

- theme.css (جديد)
- theme.js (جديد)

كل ملفات ال view (تم إضافة رابط theme.css)

تفاصيل التطبيق (Implementation Details):

1. نظام متغيرات CSS (CSS Variables System)

CSS

```
/* theme.css - Light Mode */
} theme-light.
```

```

;primary-color: #f77f48--
;background: #f8f7f5--
;card-bg: #fff--
;text: #222--
;muted: #666--
{

/* (الوضع الداكن) Dark Mode */
} theme-dark.
;primary-color: #ffa56a--
;background: #0f1720--
;card-bg: #11151a--
;text: #e6eef6--
;muted: #9aa6b2--
{

```

2. آلية تبديل ال Theme (Theme Toggle Mechanism)

- الموقع (Location): ال Header بتاع index.php
- يعتمد على أيقونات (Icon-based): شمس (☀️) للوضع المضيء، قمر (🌙) للوضع الداكن
- الاستمرارية (Persistence): بيستخدم مفتاح localStorage اسمه gigastore:theme
- تطبيق تلقائي (Auto-apply): ال JavaScript يطبق كلاس ال theme على ال <html> أول ما الصفحة تحمل

3. مكونات محدثة (Updated Components)

كل ال Components دلوقتي بتشير لمتغيرات ال theme:

- background لخلفيات الصفحات
- card-bg لل cards/containers
- text للون النص
- muted للنص الثانوي

قبل (Before):

CSS

```

;background: #ffffff
;color: #333

```

بعد (After):

CSS

```

;background: var(--card-bg)
;color: var(--text)

```

🛒 إصلاح شامل لنظام سلة التسوق (Cart System Overhaul)

مشاكل التطبيق السابق (Previous Implementation Issues):

1. تكرار الـ logic في main.js و index.js
2. عدم تطابق تنسيق مفتاح الـ Cart بين الصفحات
3. مفيش عزل للـ Cart لكل مستخدم (per-user cart isolation)

التطبيق الجديد (New Implementation):

1. تخزين Cart خاص بكل مستخدم (Per-User Cart Storage)

JavaScript

```
NEW: User-specific cart keys //  
    } function cartKey(uid)  
    "return `cart:u:${uid}`"; // Format: "cart:u:123"  
    {
```

المميزات (Benefits):

- ممكن أكثر من مستخدم يستعملوا نفس الـ browser
- فصل سلات الضيوف عن سلات المستخدمين اللي عاملين login
- الـ Cart بيفضل موجود في الجلسات المختلفة

2. تجميع منطق الـ Cart (Consolidated Cart Logic)

الملف (File): main.js

JavaScript

```
Single source of truth for cart operations //  
    (إضافة عنصر للـ cart) addProductToCart() // Add item to cart -  
    (تحديث إشارة الـ cart) updateCartCount() // Update cart badge -  
    (الذهاب لصفحة الـ checkout) goToCart() // Navigate to checkout -  
    (حماية لإجراءات الـ cart) requireLogin() // Guard for cart actions -
```

3. تجميع العناصر المكررة (Duplicate Item Aggregation)

الملف (File): checkout.js

JavaScript

```
    } function aggregateCart(cart)  
    ;{} = const aggregated  
    } <= cart.forEach(item  
;const id = item.id || 0  
    } if (!aggregated[id])  
;aggregated[id] = { ...item, quantity: 0 }  
    {  
;aggregated[id].quantity += (item.quantity || 1)  
    ;({  
;return Object.values(aggregated)  
    {
```

النتيجة (Effect):

- نفس المنتج لما يتضاف 3 مرات ← يظهر "Product × 3"
- عرض الـ checkout أنظف
- تتبع دقيق للـ quantity

خاصية البحث (Search Functionality) 🔍

الحالة السابقة (Previous State): شريط بحث مش شغال (Non-functional search bar)

التطبيق الجديد (New Implementation):

الملفات المعدلة (Files Modified):

- main.js
- index.php

1. تصفية المنتجات في الوقت الفعلي (Real-time Product Filtering)

JavaScript

```
// Debounced input search (تأخير 180ms)
} <= searchInput.addEventListener('input', (e)
    ;clearTimeout(debounceTimer)
} <= ()debounceTimer = setTimeout
;filterProducts(e.target.value)
    ;(180 ,{
    ;({
```

2. دمج زر البحث (Search Button Integration)

قبل (Before): الزرار مكنش بيعمل حاجة

HTML

```
<button class="btn btn-search">Search</button>
```

بعد (After): زرار أيقونة مع click handler

HTML

```
<"button class="btn btn-search" title="Search">
<i class="fa-solid fa-magnifying-glass"></i>
<button/>
```

3. خوارزمية المطابقة (Matching Algorithm)

JavaScript

```

    } function filterProducts(query)
    Search in product name AND description //
    ;()const title = titleEl.textContent.toLowerCase
    ;()const desc = descEl.textContent.toLowerCase
    ;const match = title.includes(query) || desc.includes(query)
    ;'p.style.display = match ? '' : 'none
    {

```

تحسينات على صفحة ال Checkout

الملف (File): checkout.php + checkout.js

1. أدوات التحكم في الكمية (Quantity Controls)

خاصية جديدة (NEW FEATURE): زرار +/- لكل منتج

JavaScript

```

    (زرار ناقص - تقليل الكمية) Minus button (decrease quantity) //
    ;minusBtn.addEventListener('click', () => updateQuantity(p.id, -1))

    (زرار زائد - زيادة الكمية) Plus button (increase quantity) //
    ;plusBtn.addEventListener('click', () => updateQuantity(p.id, 1))

```

الشكل (Visual):

Product Name [-] 2 [+] \$29.90 [Product Image]

2. صور المنتجات في ال Cart (Product Images in Cart)

قبل (Before): قائمة نصوص بس

بعد (After): صورة + اسم + أدوات تحكم + سعر

JavaScript

```

    ;const img = document.createElement('img')
    ;img.src = p.image || '../assets/images/holder.jpg
    ;img.style.width = '60px
    ;img.style.height = '60px

```

3. مسح ال Cart تلقائياً عند النجاح (Auto-Clear Cart on Success)

JavaScript

```

    } if (data && data.success)
    localStorage.removeItem(key); // Clear cart

```

```
;window.location.href = 'order_success.php?order_id=' + data.order_id
{
```

4. ملء بيانات الجلسة مسبقاً (Session Pre-fill)

PHP

```
"input type="text" name="customer_name">
<"<? value="<?= htmlspecialchars($_SESSION['user_name']) ?? ''>
"input type="email" name="customer_email">
<"<? value="<?= htmlspecialchars($_SESSION['user_email']) ?? ''>
```

إدارة الطلبات (Order Management) 📦

الملف (File): orders.php + OrderModel.php + OrderController.php

1. خاصية إلغاء الطلب (NEW) (Cancel Order Feature)

دالة قاعدة البيانات (Database Function):

PHP

```
models/OrderModel.php //
} function cancel_order($conn, $id)
('status 'Pending الى الإلغاء فقط لو الـ 'Only cancel if status is 'Pending //
;stmt = $conn->prepare("SELECT order_status FROM Orders WHERE id = ?")$
(validation logic ... //
stmt = $conn->prepare("UPDATE Orders SET order_status = 'Cancelled' WHERE id = $
;?")
{
```

نقطة نهاية الـ (Controller Endpoint): Controller

PHP

```
controllers/OrderController.php //
} if ($action === 'cancel')
Verify ownership (user or admin) //
} if (!$isAdmin && (int)$order['user_id'] !== $userId)
;echo json_encode(['success' => false, 'message' => 'Not authorized'])
;exit
{
;cancel_order($conn, $orderId)
{
```

زرار الواجهة الأمامية (Frontend Button):

PHP

```

        <? :php if ($o['order_status'] === 'Pending')?>
        <"button class="btn btn-cancel" onclick="cancelOrder(<?= (int)$o['id'] ?>)"
        i class="fa-solid fa-trash"></i> Cancel Order>
        <button/>
        <? ;php endif?>

```

معالج الـ JavaScript (JavaScript Handler)

JavaScript

```

        } function cancelOrder(orderId)
        ;if (!confirm('Are you sure?')) return

        } , 'fetch('/Giga-store/controllers/OrderController.php
        , 'method: 'POST
body: new FormData(['action', 'cancel'], ['order_id', orderId])
        ({
        then(resp => resp.json()).
        } <= then(data.
        ;())if (data.success) location.reload
        ;({
        {

```

2. التنقل في صفحة الطلبات (Order Page Navigation)

جديد (NEW): تم إضافة header بروابط

HTML

```

        <"header class="header>
        <nav>
        <a href="index.php">Home</a>
a href="dashboard.php">Dashboard</a> (if admin)>
        <a href="logout">Logout</a>
        <nav/>
        <header/>

```

3. تنسيق حالة الطلب (Styled Order Status)

PHP

```

        <" ;span style="font-weight:600;color:var(--primary-color)>
        <? htmlspecialchars($o['order_status']) =?>
        <span/>

```

إدارة المستخدمين (User Management)

الملفات (Files): userController.php (تم الدمج - MERGED)

تغيير الدمج (Consolidation Change):

Class يعتمد على الـ userController.php

PHP

```
        } class UserController
    { /* ... */ } ()public function handleUpdate
    { /* ... */ } ()public function handleDelete
    {

        Router at bottom //
        ;'' ?? action = $_REQUEST['action']$
        } if ($action === 'update')
        ;()controller = new UserController$
        ;()controller->handleUpdate$
    {
```

المراجع المحدثة (Updated References):

- edit_user.php → ../controllers/userController.php?action=update
- dashboard.php → ../controllers/userController.php?action=delete

إعادة هيكلة (CSS Refactoring) 🎨

هيكل الملفات (File Structure):

OLD: Inline styles in PHP files (ستايلات داخل ملفات PHP)
NEW: Centralized CSS files (ملفات CSS مركزية)

1. متغيرات بتراعي الـ Theme (Theme-Aware Variables)

الملف (File): store-style.css

CSS

```
    } root:
;primary-color: var(--primary-color, #f77f48)--
;background-color: var(--background, #f8f7f5)--
;card-background: var(--card-bg, #ffffff)--
;text-dark: var(--text, #222)--
{
```

نظام Fallback:

- لو theme.css متحملش ← هستخدم القيم الافتراضية
- بيضمن التوافق مع الإصدارات الأقدم (backward compatibility)

2. ستايلات الـ Component (Component Styles)

CSS

```
/* كروت سجل الطلبات (order-card) */ /* Order history cards.
/* (checkout ال فورم غلاف) checkout-card */ /* Checkout form wrapper.
/* (زرار الإلغاء الأصفر) btn-cancel */ /* Yellow cancel button.
/* (فورم تعديل المستخدم) edit-card */ /* Edit user form.
/* (كونتينر صفحة النجاح) order-success */ /* Success page container.
```

3. إزالة ال (Removed Inline Styles)

قبل (Before) (orders.php):

PHP

```
<"div style="background:#fff;padding:20px;border-radius:8px>
```

بعد (After):

PHP

```
<"div class="order-card>
```

تحسينات الاستجابة وتجربة المستخدم (Responsive & UX Improvements)

1. تحسين أيقونة البحث (Search Icon Enhancement)

CSS

```
    } btn-search i.
    ;font-size: 16px
/* clicks منع الأيقونة من منع ال */ ;pointer-events: none
{
```

2. تحديد موقع إشارة ال (Cart Badge Positioning)

CSS

```
    } cart-count#
;position: absolute
;top: -8px
;right: -8px
;background: var(--primary-color)
```

```
;border-radius: 50%  
{
```

3. حالات ال Hover للأزرار (Button Hover States)

CSS

```
btn-cancel:hover { background: #ffb300; }.  
theme-toggle:active { transform: scale(.98); }.
```

4. تطبيق ال Theming على مدخلات ال Form (Form Input Theming)

CSS

```
        } form-row input.  
/* Theme بيراعي ال */ ;background: var(--background)  
                        ;color: var(--text)  
;border: 1px solid rgba(0,0,0,0.06)  
{
```

تحسينات الأمان (Security Enhancements) 🔒

1. التحقق من الجلسة (Session Validation)

كل الصفحات المحمية بتعمل Check دلوقتي على:

PHP

```
        ;()session_start  
} if (!isset($_SESSION['user_id']))  
;header('Location: login.php')  
;exit  
{
```

2. التحقق من ملكية الطلب (Order Ownership Verification)

PHP

```
OrderController.php - Cancel order //  
;stmt = $conn->prepare("SELECT user_id FROM Orders WHERE id = ?")$  
} if (!$isAdmin && (int)$order['user_id'] !== $userId)  
;echo json_encode(['success' => false, 'message' => 'Not authorized'])  
;exit  
{
```

3. منع هجمات XSS (XSS Prevention)

كل بيانات المستخدم دلوقتي بيتعملها escape:

```
<? htmlspecialchars($userName) =?>
<? htmlspecialchars($o['order_status']) =?>
```

تغييرات تفاعل قاعدة البيانات (Database Interaction Changes) 📊

1. معالجة الأخطاء في OrderModel (Error Handling in OrderModel)

```

    } (...)function create_order
    {
        try
        {
            ;()conn->begin_transaction$
            (منطق إنشاء الطلب) order creation logic ... //
            ;()conn->commit$
        } catch (Throwable $e) {
            ;()conn->rollback$
            throw $e; // Re-throw for caller to handle
        }
    }

```

2. Prepared Statements

كل ال Queries دلوقتي بتستخدم Prepared Statements:

```

$stmt = $conn->prepare("SELECT * FROM Orders WHERE user_id = ?")$
$stmt->bind_param('i', $userId)$

```

أنماط JavaScript جديدة (New JavaScript Patterns) 🧪

1. IIFE لعزل النطاق (IIFE for Scope Isolation)

```

// checkout.js
(function () {
    const CURRENT_USER_ID = window.CURRENT_USER_ID || 0
    // logic here doesn't pollute global scope ...
    // هنا مش بيلوث النطاق العام
})();

```

2. Async/Await لمكالمات API (Async/Await for API Calls)

```
        } <= checkoutForm.addEventListener('submit', async (e)
                                ;()e.preventDefault
    } , 'const resp = await fetch('/Giga-store/controllers/OrderController.php
                                , 'method: 'POST
                                body: formData
                                ;({
                                ;()const data = await resp.json
                                ;({
```

3. Debounced Search

```
                                ;let debounceTimer = null
    } <= searchInput.addEventListener('input', (e)
                                ;clearTimeout(debounceTimer)
    ;debounceTimer = setTimeout(() => filterProducts(e.target.value), 180)
                                ;({
```

جدول ملخص (Summary Table) 📋

Category	Changes (التغييرات)	Files Affected (الملفات المتأثرة)
Theme System	Added dark/light mode with persistence (إضافة وضع داكن/امضيء مع استمرارية)	theme.css, theme.js, all views
Cart Logic	Per-user carts, aggregation, quantity controls (لكل مستخدم، تجميع، تحكم في الكمية Cart)	main.js, checkout.js
Search	Real-time filtering, icon button (تصفية في الوقت الفعلي، زرار أيقونة)	main.js, index.php
Orders	Cancel functionality, navigation header (خاصية للإلغاء، Header للتنقل)	orders.php, OrderModel.php, OrderController.php
Checkout	+/- quantity, images, session pre-fill (+/- الكمية، صور، ملء بيانات الجلسة)	checkout.php, checkout.js
CSS	Removed inline styles, theme variables (إزالة Inline styles، متغيرات Theme)	store-style.css, pages.css
Controllers	added the logic of userController (إضافة منطق userController)	userController.php
Security	Session checks, ownership validation (تحقق من الملكية Session)	All controllers & views

للتحديث من الإصدار السابق (To Update from Previous Version):

1. جلب الملفات الجديدة (Pull new files):

```
bash git pull origin main
```

2. تحديث قاعدة البيانات (Update database) (لو ال schema اتغير):

```
bash # No schema changes in this version (في الإصدار ده schema مفيش تغييرات في ال)
```

3. مسح ذاكرة التخزين المؤقت للمتصفح (Clear browser cache):

- Hard refresh (Ctrl+Shift+R)

- مسح localStorage لبيانات ال cart

4. تحديث المراجع (Update references):

- إزالة مراجع index.js

5. قائمة التحقق للاختبار (Test checklist):

- [] Theme toggle شغال

- [] Search يصفى المنتجات

- [] ال Cart يجمع التكرارات

- [] Checkout بيعرض الصور

- [] أزرار +/- بتحدث ال quantity

- [] إلغاء الطلب شغال (Pending orders)

- [] الوضع الداكن بيفضل شغال بعد ال refresh

تغييرات مؤثرة (Breaking Changes)

1. تنسيق تخزين ال Cart (Cart Storage Format):

- قديم: cart (عام - global)

- جديد: cart:u:{userId} (لكل مستخدم - per-user)

- الترحيل (Migration): ال Carts القديمة هتضيع مع التحديث

2. الاعتماد على CSS (CSS Dependencies):

- كل الصفحات لازم يكون فيها رابط theme.css

- لو الرابط مش موجود = الوضع الداكن مش هيشغل

إصلاحات الأخطاء (Bug Fixes)

1. ال Cart فاضي بعد إضافة العناصر:

- تم إصلاح عدم تطابق المفتاح بين index.php و checkout.php

- دلوقتي بيستخدم دالة cartKey () الثابتة

2. شاشة بيضاء في الوضع الداكن:

- تم استبدال ffff المكتوب بشكل صريح بـ var(--card-bg)

- تم تحديث كل مراجع الخلفية/اللون

3. عناصر مكررة في ال checkout:

- تم إضافة دالة aggregateCart ()

- بتجمع العناصر حسب Product ID

4. زر البحث مش شغال:

- تم إضافة click event listener

- الأيقونة مش بتمنع ضغط الزرار

تحسينات مستقبلية (Future Improvements) (لم يتم تطبيقها بعد) 🍌

1. تصفية الفئات + البحث (Category filtering + Search):

- دمج اختيار الفئة مع استعلام البحث

2. استمرارية ال Cart في قاعدة البيانات (Cart persistence to database):

- حفظ عناصر ال Cart في ال DB للمستخدمين اللي عاملين login

3. تتبع الطلب (Order tracking):

- تحديثات الحالة في الوقت الفعلي

4. اكتشاف ال Theme تلقائياً (Theme auto-detect):

- استخدام prefers-color-scheme media query

5. تطبيق ال Theming على Admin dashboard:

- حالياً بس index/checkout/orders اللي فيهم الوضع الداكن