

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 Features, Architecture, UI/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 —|
    (استخراجه)
  main.js # consolidated cart & search —|
    (search)
  theme.js # Theme toggle system —|
    /assets/css —|
    theme.css # Theme variables —|
    pages.css # Shared page styles —|
    (متغيرات الـ theme.css)
    (ستيلات مشتركة للصفحات)

REMOVED FILES : (ملفات تم حذفها)
/assets/js/index.js # Merged into main.js —|
  (تم دمج محتواه في main.js)
```

تطبيق نظام الـ 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 الـ view (تم إضافة رابط theme.css)

تفاصيل التطبيق (Implementation Details)

1. نظام متغيرات CSS (CSS Variables System)

CSS

```
/* (الوضع المضيء) */
theme.css - Light Mode */
} theme-light.
```

```

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

{
    /* (الوضع الداكن) Dark Mode */
}

;primary-color: #ffa56a--
;background: #0f1720--
;card-bg: #11151a--
;text: #e6eef6--
;muted: #9aa6b2--

{

```

2. آلية تبديل الـ (Theme Toggle Mechanism)

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

3. مكونات محدثة (Updated Components)

كل الـ Components دلوقتي بتشير لمتغيرات الـ :theme

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

:Before (قبل)

CSS

```

;background: #ffffff
;color: #333

```

:After (بعد)

CSS

```

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

```

إصلاح شامل لنظام سلة التسوق (Cart System Overhaul)

: (Previous Implementation Issues) مشاكل التطبيق السابق

1. تكرار الـ `index.js` في `logic` و `main.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)

main.js : (File)

JavaScript

```
Single source of truth for cart operations //  
(cart) addProductToCart() // Add item to cart -  
(cart) updateCartCount() // Update cart badge -  
(checkout) goToCart() // Navigate to checkout -  
(cart) requireLogin() // Guard for cart actions -
```

3. تجميع العناصر المكررة (Duplicate Item Aggregation)

checkout.js : (File)

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 يتضمن 3 مرات ← يظهر "3 × ببطة"
- عرض الـ **checkout** أنظر
- تتبع دقيق للـ **quantity**

(Search Functionality) خاصية البحث

الحالة السابقة (Previous State): شريط بحث مش شغال (bar)

التطبيق الجديد (New Implementation):

الملفات المعدلة (Files Modified)

- main.js
- index.php

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

JavaScript

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

2. دمج زر البحث (Search Button Integration)

قبل (Before): الزرار مكتنش 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 الـ 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))
```

: (Visual)

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

2. صور المنتجات في الـ 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 الـ Cart تلقائياً عند النجاح (Auto-Clear Cart on Success)

JavaScript

```
} if (data && data.success)
(cart) مسح الـ 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)

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

1. خاصية إلغاء الطلب (Cancel Order Feature)

: (Database Function)

PHP

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

: Controller (Controller Endpoint)

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)
}
```

: Frontend Button (Zrar Al-Wajha Al-Amamia)

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/>

```

ادارة المستخدمين (User Management)

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

تغبير الدمج (Consolidation Change)

userController.php يعتمد على الـ Class

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 (CSS Refactoring)

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

1. متغيرات بتراعي الـ (Theme-Aware Variables)

store-style.css : (File)

CSS

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

:Fallback نظام

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

2. ستايلات الـ (Component Styles)

كلاسات جديدة (NEW Classes)

CSS

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

3. إزالة الـ Inline Styles (Removed Inline Styles)

: (orders.php) (Before)

PHP

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

: (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 الـ ;pointer-events: none  
{
```

2. تحديد موقع إشارة الـ Cart (Cart Badge Positioning)

CSS

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

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

3. حالات الـ Hover (Button Hover States)

CSS

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

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

CSS

```
} form-row input.  
/* Theme */ /* يبراعي الـ */  
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. التتحقق 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

PHP

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

تغييرات تفاعل قاعدة البيانات (Database Interaction Changes)

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

PHP

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

2. Prepared Statements

كل الـ **Queries** دلوقتي بتستخدم **Prepared Statements**

PHP

```
;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)

JavaScript

```
checkout.js //(
    } () function)
;const CURRENT_USER_ID = window.CURRENT_USER_ID || 0
    ;logic here doesn't pollute global scope ... //(
        ;()
    )()
```

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

JavaScript

```
        } <= 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

JavaScript

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

جدول ملخص (Summary Table) 

Category	Changes (التحفيزات)	Files Affected (الملفات المتأثرة)
Theme System	Added dark/light mode with persistence (إضافة وضع داكن/مضيء مع استمرارية)	theme.css, theme.js, all views
Cart Logic	Per-user carts, aggregation, quantity controls (لكل مستخدم، تجميع، تحكم في الكمية)	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 إلغاء للتنقل، إلغاء Header)	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 إيزاله Inline styles ، متغيرات Theme)	store-style.css, pages.css
Controllers	added the logic of userController (إضافة منطق userController) لـ userController	userController.php
Security	Session checks, ownership validation (تحقق من الملكية لـ 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 الـ schema انغير):

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

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

- Hard refresh (Ctrl+Shift+R)

- مسح localStorage لبيانات الـ cart

4. تحديث المراجع : (Update references)

- إزالة مراجع index.js

5. قائمة التحقق لاختبار (Test checklist):

- [] Theme toggle شغال

- [] Search بيفصفي المنتجات

- [] الـ Cart بيعمل التكرارات

- [] Checkout بيعرض الصور

- [] أزرار +/- بتحدث الـ quantity

- [] إلغاء الطلب شغال (Pending orders)

- [] الوضع الداكن بيفضل شغال بعد الـ refresh

تغييرات مؤثرة (Breaking Changes)

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

- قديم: cart (عام -

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

- الترحيل (Migration): الـ Carts القديمة هتنصب مع التحديث

2. الاعتماد على CSS (CSS Dependencies):

- كل الصفحات لازم يكون فيها رابط theme.css

- لو الرابط مش موجود = الوضع الداكن مش هيستغل

إصلاحات الأخطاء (Bug Fixes)

1. الـ Cart فاضي بعد إضافة العناصر:

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

- دلوقتي بيستخدم دالة cartKey() الثابتة

. شاشة بيضاء في الوضع الداكن:

- تم استبدال `fff#` المكتوب بشكل صريح بـ `var(--card-bg)`

- تم تحديث كل مراجع الخلفية/اللون

. عناصر مكررة في الـ `:checkout`

- تم إضافة دالة `()aggregateCart`

- يتحمّع العناصر حسب Product ID

. زر البحث مش شغال:

- تم إضافة click event listener

- الأيقونة مش يتمتعن 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 الـ Cart في قاعدة البيانات : (Cart persistence to database)

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

3. تتبع الطلب : (Order tracking)

- تحديات الحالة في الوقت الفعلي

4. اكتشاف الـ Theme تلقائيًّا : (Theme auto-detect)

- استخدام prefers-color-scheme media query

5. تطبيق الـ Admin dashboard على Theming

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