



Custom Theme and Plugins Discovery & Documentation

Overview

This document provides a structured summary of custom code, theme modifications, and plugin functionality identified during our discovery process. It is formatted to show the level of detail we capture in our reviews while obfuscating any client-specific branding, industries, or proprietary code.

Key Findings

Theme Customizations

The child theme contains numerous template overrides, styling changes, and functional adjustments. Below is an example of the type of line-by-line documentation we provide:

- functions.php – Lines 4–12: Pulls in custom results template.
- Lines 15–21: Sets theme category row count.
- Lines 23–31: Adjusts product price formatting.
- Lines 33–40: Adds CSS classes to product titles.
- Lines 44–49: Removes default lightbox and button actions.
- Lines 51–53: Adds set-based product pricing.
- Lines 55–68: Adds custom lightbox library for product images.
- Lines 70–115: Implements additional modals for product information.
- Lines 117–122: Disables wishlist option.
- Lines 124–131: Loads integration code for cart page.
- Lines 137–147: Adds “more details” link to product boxes.
- Lines 164–171: Redirects continue shopping button to homepage.
- Lines 173–261: Adds checkout integration and styling.
- Lines 263–506: Retrieves product attributes for main shop loop.
- Lines 508–512: Adds “continue shopping” button to checkout.
- Lines 514–520: Displays product brand.

Custom Plugin Scripts

- add_to_cart.php – Adds multiple products to cart in one action.
- add_upsells_to_products.php – Creates upsells before checkout.
- advanced-search.php – Provides advanced search filtering.
- arbitrary_fees.php – Fully commented out and can be archived.
- asapshipping.php – Marks products as fast-shipping.
- attribute_report.php – Generates attribute-based sales reports.
- brand_breadcrumbs.php – Creates brand breadcrumbs.
- caps.php – Prevents deletion of critical product categories.
- coupon_ajax.php – Reloads cart when coupon added.
- customer-quotes.php – Allows product quoting instead of checkout.
- emails.php – Customizes email footer, attributes, and phone formatting.
- fitment_api.php – Connects to external API for compatibility data.
- freight-option-fields.php – Adds freight charge fields.
- inventory.php – Allows CSV import of vendor inventory.
- labels.php – Custom shipping label generation.
- multiple_status_order_page.php – Creates custom order status dashboard.
- payment-gateway-icon.php – Adds custom payment gateway icons.
- phone-authentication.php – Enables phone number login.
- product_permalinks.php – Adds SKU to product permalink.
- reporting.php – Custom reporting for sales and attributes.
- search-by-sku.php – Implements part number search shortcode.
- shipping.php – Adds vendor notifications and tracking.
- tire_search.php – Provides custom tire search functionality.
- vehicle_search.php – Core search logic for product compatibility.
- wishlist.php – Extends wishlist functionality for product bundles.

Recommendations

- Code Consolidation – Merge fragmented plugins into a single modular plugin for ease of maintenance.
- Remove Unused Code – Archive deprecated and commented-out code offsite to reduce clutter.
- Documentation – Maintain this level of detailed line-by-line documentation for future updates.
- Refactoring – Adopt coding best practices and modular structures for stability and upgrade resilience.

Estimated effort for consolidation: 20–40 hours depending on complexity.

Conclusion

This discovery demonstrates Inspry's process for documenting each line of custom code, identifying unused functionality, and providing recommendations for consolidation and maintainability. By following these steps, clients gain improved stability, reduced technical debt, and easier long-term management of their website.

Prepared by Inspry – WordPress & WooCommerce Development Experts