



Custom Shopify Theme and Apps Discovery

Overview

This document provides a structured summary of custom code, theme modifications, and app 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 theme contains a significant number of custom Liquid edits, template overrides, JavaScript modifications, and non-standard app integrations. Below is a sample of the type of line-by-line detail included in our reviews:

- theme.liquid — Lines 12–28: Replaces native Shopify scripts with custom tracking logic.
- Lines 44–67: Injects custom metafield rendering for dynamic product specs.
- Lines 88–103: Adds custom event tracking before add-to-cart actions.
- product.liquid — Lines 14–22: Overrides core product media gallery.
- Lines 33–46: Adjusts price formatting and introduces tiered pricing display.
- Lines 50–77: Injects custom buy-box layout and auxiliary CTAs.
- collection.liquid — Lines 10–29: Adds custom collection filtering logic tied to tags/metafields.
- cart.liquid — Lines 55–82: Adds support for line-item property validation.
- checkout.liquid (Shopify Plus only) — Lines 14–39: Custom checkout script for upsell prompts.
- global.js — Lines 102–158: Overrides theme's default cart drawer behavior.
- snippets/product-card.liquid — Lines 21–36: Adds brand badges and inventory indicators.

Custom Private App Scripts

Below is an example of the type of custom app functionality identified during an audit:

- bulk-add-to-cart.js – Enables adding multiple variants to the cart in a single request.
- upsell-engine.js – Injects dynamic upsells pre-checkout based on metafield rules.
- advanced-search-endpoint.js – Creates custom search endpoints using Shopify Storefront API.
- custom-fees.js – Calculates additional fees via Shopify Functions.
- fast-shipping-flag.js – Marks select SKUs as expedited shipping items.
- brand-breadcrumbs.js – Generates brand-based navigation elements.
- inventory-sync.js – Syncs vendor feeds via Shopify Admin API.
- order-status-dashboard.js – Adds extended order reporting via a private admin app.
- phone-auth.js – Implements phone-number authentication through a custom user flow.

- sku-search.js – Enables instant SKU/part-number search with predictive results.
- custom-labels.js – Generates shipping labels using a third-party logistics API.
- reporting.js – Pulls Sales, SKU, and attribute reports via Shopify's Analytics API.
- compatibility-engine.js – Integrates external product fitment/compatibility API (auto/machinery example).
- wishlist-extension.js – Extends wishlist features for bundles and variant groups.

Deprecated or fully commented-out scripts are flagged separately.

Recommendations

- App Consolidation – Several private app features should be merged into a single modular private app or custom Shopify app to simplify maintenance and streamline API usage.
- Remove Legacy Code – Outdated Liquid blocks, unused snippets, and commented-out JS should be archived offsite and removed from the theme.
- Improve Documentation – Maintain this level of line-by-line clarity to improve future code handoffs and long-term maintainability.
- Refactor for Best Practices – Implement Shopify best practices including section schema updates, reducing DOM manipulation, and offloading logic to metafields or Functions where possible.

Estimated consolidation effort: 20–40 hours, depending on historical complexity and API usage.

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 – Shopify Development Experts