Prepared By: Sufyan Khan & Muhammad Amir **Data: 16/01/2025**

In collaboration with: Muhammad Shehroz , Junaid Khan , Suleman Sehar & Abdul Ahad

Submission Title: Thriftology.com

Technical Documentation

1. Overview:
The website is an online platform designed to sell premium, branded second-hand clothing, home textiles, and accessories at affordable prices. The platform serves as an eco-friendly marketplace that allows users to buy high-quality, second-hand goods while promoting sustainability. Key features of the platform include an easy-to-navigate user interface, secure payment options, condition grading for products, and an EcoPoints program to incentivize sustainable purchases.

2. System Architecture:
The system is designed as a modern e-commerce platform with several interconnected components:

Frontend (Client-Side):
Built with Next.js, React Library and Tailwind CSS to provide a responsive, user-friendly interface.
Customers can browse products, filter by categories (Men, Women, Kids, Home Textiles), search for items by size, brand, or condition, and view product details.
Secure login system for customer accounts and a checkout process that integrates with various payment gateways (Credit Card, All banking platfoms).

Backend (Server-Side):
Sanity CMS will act as the central hub for managing the data for your e-commerce platform, including inventory management, order processing, shiopment , payment and integration with the EcoPoints system.

Inventory Management through Sanity:

Product Data Modeling: In Sanity, you can define a schema for products where each product will have attributes like name, description, price, category, available stock, and images.

Dynamic Content: You can create a flexible, real-time inventory management system. As the inventory (product stock) updates in Sanity, the frontend can pull this data in real-time to ensure customers always see accurate product availability.

Order Processing with Sanity:

Order Schema: Create an order schema to capture essential details such as the customer's information, product(s) ordered, shipping address, order status (processing, shipped, completed), and payment status.
Tracking Orders: Sanity can store historical order data, which can be accessed by admins or used by the API to provide order status updates to customers.

Order Workflow: You can integrate Sanity with external services (e.g., payment gateways or shipping providers) via webhooks or API calls to automate order processing. This means when a customer places an order, it updates the status in Sanity, and the backend processes the next steps (payment, inventory reduction, etc.).

EcoPoints System Integration:

User Points Schema: Store EcoPoints balances and transaction histories in Sanity by creating a schema for user EcoPoints. Each time a user earns or spends points, this information can be stored and updated in Sanity.
EcoPoints Logic: You can use Sanity's flexible content model to store and update EcoPoints transaction records, such as when points are awarded for product purchases or used for discounts.
Real-Time Updates: Sanity allows your application to reflect EcoPoints changes immediately as they are updated in the database. The backend can calculate discounts or rewards based on the user's points balance, and frontend data can be fetched to reflect the points in real-time.

API Layer:
RESTful API endpoints for:
Product management (CRUD operations)
User management (authentication and profile)
Order management (creating, processing, and tracking orders)
EcoPoints management (crediting points, applying discounts)
Authentication and authorization using JWT (JSON Web Tokens) for secure user login and access control.

Payment System:
Integrated with third-party payment gateways like Stripe or all banking platforms to process payments securely.