



---

# Website Redevelopment Proposal

PREPARED BY



DATE

2 June 2026

VERSION

1.0

# Introduction

---

This proposal outlines two options for Avon to redesign their marketing website and merge it with the shop website. After analysing the requirements against the strict go-live date, Intuji has divided the project scope into two distinct categories: the core functionalities mandatory for the initial launch, and the advanced add-on modules that can be integrated post-launch.

To achieve the premium design Avon requires, standard Shopify templates are insufficient. Both options separate the visual interface from the backend, using your existing Shopify setup strictly to process transactions. Because the new frontend and the advanced add-on features are built independently, they will require dedicated external server hosting.

The two approaches detailed below allow Avon to choose the best long-term management strategy: adapting its current Shopify backend ecosystem (Option 1) or transitioning to a centralised, enterprise-grade NestJS administration platform (Option 2).

---

**2**

TECHNICAL OPTIONS

**3**

PROJECT PHASES

**3**

ADD-ON MODULES

## Scope of Work

---

The scope has been categorised into two groups to facilitate a phased, risk-managed rollout.

### CORE REQUIREMENTS

#### **Non-Negotiables**

The foundational works required to redesign the frontend, merge both websites, and deliver a fully operational global platform.

Mandatory for the initial launch.

### ADD-ON REQUIREMENTS

#### **Phased Modules**

High-value, specialised modules developed and integrated independently. Not blockers for go-live; can be selected, sequenced, or phased based on operational readiness.

## Project Phases

---

The project follows a structured three-phase methodology. Each phase has defined deliverables for both core launch requirements and optional add-on modules.



### Discovery

Deep-dive into requirements, user journeys, third-party integrations, and content architecture to define the complete technical specification.



### Design

Create the premium visual identity, responsive page templates, and complete UX flows for all B2B, B2C, and content experiences.



### Development

Build production-ready frontend code, CMS components, ecommerce logic, API integrations, and all backend systems.

04

## Core Activities

---

The following activities make up the Discovery phase. They provide the essential foundation needed to align our teams and ensure a successful website redevelopment.

### 01 Frontend Consolidation Scope

Analyse and map structural requirements for merging the marketing and shop websites into a single cohesive global frontend.

### 03 B2C User Journey Definition

Map out e-commerce cart, checkout, user account, and payment gateway requirements for the B2C flow.

### 05 Knowledge Hub Taxonomy

Define categories, tags, search filtering, and author attribution for the thought-leadership repository.

### 02 B2B User Journey Definition

Document form fields, routing rules, and notifications for the B2B lead generation workflow.

### 04 CMS Architecture & Template Constraints

Map content editing requirements and audit Shopify template limits for custom landing pages.

### 06 Third-Party Data Workflows

Map data inputs, outputs, and sync frequencies between the frontend/Shopify and Klaviyo, Docebo, and Zendesk APIs.

04

## Add-On Activities

---

Optional Discovery activities that can be selected and sequenced independently. These are not blockers for launch but add significant value when included.

**01 Interactive Map Backend & Plotting Engine**

Define technical specs for a custom backend location engine and secure coordinate transmission to the frontend.

**02 Configurator Compatibility Matrix**

Define logic and admin UI for a backend rule engine mapping strict product compatibility dependencies.

**03 Configurator Frontend Pricing**

Document the user interface requirements for step-by-step part selection, dynamic pricing aggregation, and bundle checkout.

**04 Partner Portal Role-Based Access**

Map user management architecture, role definitions, access permissions, and secure document gating to replace Showpad.

**05 Content Migration Specifications**

Define data structure, metadata fields, and import schemas for bulk migration of existing Showpad assets.

## Core Activities

---

The core design activities that translate Discovery findings into high-fidelity, responsive visual designs ready for development.

### 01 Global UI/UX Design System

Establish a comprehensive, premium digital style guide (typography, colour palette, button states, spacing rules) that elevates the Avon brand to a global standard.

### 03 Core Page Templates (High-Fidelity)

Design responsive frontend layouts for Homepage, Category/Listing Pages, Standard Text/About Pages, and Contact Pages across Desktop, Tablet, and Mobile breakpoints.

### 05 B2C Ecommerce Flow

Design the complete responsive Shopify integration journey: Product Detail Pages, Cart, secure Checkout interface, and Customer Account dashboard.

### 02 Unified Navigation & Information Architecture

Design responsive global header and footer navigation menus that intuitively direct B2B and B2C traffic from a single unified entry point.

### 04 B2B Lead Generation Flow

Design the complete UX journey for corporate and government clients, including inquiry forms, gated content access points, and confirmation states.

### 06 Knowledge Hub Interface

Design frontend layouts for the thought-leadership blog: hub index, category filtering, individual article layouts, and author profiles.

## Add-On Activities

---

Optional Design activities for specialised modules. These can be sequenced independently based on priority and budget.

### 01 Interactive Map Interface

Design the responsive frontend map component, including location pin clustering, a searchable location list/sidebar for sales reps and distributors, and individual location detail cards.

### 02 Product Configurator Frontend

Design a step-by-step, responsive visual interface that guides users through selecting base products and filtering compatible add-ons, with bundled pricing and checkout handoff.

### 03 Protected Partner Portal

Design the secure login screen, password recovery flow, and gated frontend dashboard for authenticated users to search, preview, and download secure assets.

## Core Activities

---

The core development work that builds production-ready code from approved designs. These are mandatory for launch.

### 01 Global Frontend Build

Translate high-fidelity responsive designs into production-ready frontend code for all core templates across all breakpoints.

### 02 CMS Component Development

Build backend administrative interfaces and modular page-builder components so the marketing team can create and manage landing pages without developer intervention.

### 03 B2C Ecommerce Logic (Shopify)

Implement the B2C transactional engine: dynamic cart slide-out, secure checkout flow, user account dashboards, and payment gateway connections.

### 04 B2B Lead Generation Engine

Program secure form capture, data validation logic, database storage, and automated routing rules for B2B corporate and government inquiries.

### 05 Knowledge Hub Development

Develop the dynamic thought-leadership repository, implementing taxonomy architecture, dynamic search/filtering logic, and automated author attribution.

### 06 Third-Party API Integrations

Program secure, automated data handoffs and API connections between the website and Klaviyo (marketing), Docebo (LMS), and Zendesk (support).

06

## Add-On Activities

---

Optional Development modules that can be built independently. Not blockers for go-live; sequenced based on operational readiness.

### 01 **Interactive Map**

Develop the backend location database and CRUD admin interface. Build responsive frontend map with mapping API, dynamic pins, clustering, and searchable location sidebar.

### 02 **Product Configurator**

Develop the administrative compatibility engine and dynamic frontend application that enforces rules in real time, calculates bundle pricing, and passes grouped items to checkout.

### 03 **Partner Portal**

Develop backend admin for uploading datasheets/media and managing partner accounts. Build secure login gateway, role-based access logic, and gated frontend dashboard.

## Solution Options

Two technical approaches for the platform's development. The structural differences primarily impact how the platform is built during the Development phase.

### SOLUTION

## 01

### Headless Shopify Ecosystem



Utilise the existing Shopify engine with a decoupled ReactJS frontend connected via API. Faster time-to-market with some CMS and configurator limitations.

REACTJS · SHOPIFY COMMERCE ENGINE · SHOPIFY CMS  
· SHOPIFY PRIVATE APPS

### SOLUTION

## 02

### Enterprise NestJS Architecture



Fully custom, scalable architecture. A single NestJS backend powers all content, features, and admin — Shopify handles transactions only.

REACTJS · NESTJS · PUCKJS CMS · SHOPIFY COMMERCE ENGINE

SOLUTION 1

# Headless Shopify Ecosystem

We utilise the existing Shopify engine, but decouple the frontend and develop it using ReactJS, connecting with Shopify via API.



FRONTEND  
**ReactJS**

A component-based JavaScript library to deliver a premium, bespoke storefront fully decoupled from Shopify's theme layer.



CONTENT MANAGEMENT  
**Shopify Metaobjects**

Shopify's native structured content system, allowing the marketing team to manage pages directly from the admin.



COMMERCE  
**Shopify Commerce Engine**

A GraphQL interface that powers all product data, cart, checkout, and payment processing.



CUSTOM MODULES  
**Shopify App with Node.js**

Server-side JavaScript runtime powering the Dealer Map, Product Configurator, and Partner Portal as private Shopify apps.

## SOLUTION 1

# Considerations

Key limitations and trade-offs to consider with the Headless Shopify Ecosystem approach.



### Infrastructure

Core e-commerce and CMS hosted on Shopify's cloud. The ReactJS frontend and custom Shopify apps require independent external server hosting.



### Configurator Constraints

Building complex compatibility rules is difficult due to Shopify's system limitations, leading to potential functional compromises.



### CMS Constraints

Page designs are locked to Shopify's strict theme structure. The marketing team will have limited flexibility to build new pages or custom layouts.



### Authentication

The Partner Portal must rely on Shopify's native customer authentication rather than a true, secure role-based access management system.

## Timeline & Investment

| PHASE / MODULE                           | IMPLEMENTATION METHODOLOGY   | CAPABILITIES & LIMITATIONS   | ESTIMATED TIMELINE                 | INVESTMENT       |
|--|--|--|------------------------------------|------------------|
| <b>Phase 1: Discovery</b>                | Stakeholder workshops, technical feasibility analysis, and system architecture mapping.  | <p><b>Capabilities:</b> Delivers comprehensive Business Requirements Documents (BRD) and Functional Requirements Documents (FRD), establishing a locked-in technical blueprint and roadmap prior to design.</p> <p><b>Limitations:</b> N/A</p>   | 15 Jun – 26 Jun (10 Business Days) | Fixed: \$[X,XXX] |
| <b>Phase 2: UX/UI Design</b>             | Component-based UI design focused on premium user experiences, unrestricted by Shopify theme constraints.                                      | <p><b>Capabilities:</b> Provides fully approved, interactive visual prototypes optimised for development production.</p> <p><b>Limitations:</b> N/A</p>  | 29 Jun – 17 Jul (15 Business Days) | Fixed: \$[X,XXX] |
| <b>Phase 3: Core Feature Development</b> | Custom headless frontend developed in ReactJS. Connected to Shopify's Storefront API for e-commerce transactions and Metaobjects for CMS data. | <p><b>Capabilities:</b> Achieves a premium, unrestricted frontend UI facilitating standard B2B/B2C user journeys.</p> <p><b>Limitations:</b> Managing complex, dynamic page layouts using Shopify's native backend fields (Metaobjects) is not as intuitive for marketing teams as a dedicated visual CMS.</p> | Fixed: [X] Weeks                   | Fixed: \$[X,XXX] |

| PHASE / MODULE                      | IMPLEMENTATION METHODOLOGY  | CAPABILITIES & LIMITATIONS  | ESTIMATED TIMELINE | INVESTMENT                                 |
|-------------------------------------|---|---|--------------------|--|
| <b>Add-On: Interactive Map</b>      | <p>Backend: Custom private Shopify app (Node.js) for location data management.</p> <p>Frontend: Custom ReactJS component on the headless storefront.</p>                    | <p><b>Capabilities:</b> Provides a standard, searchable distributor map displaying location pins.</p> <p><b>Limitations:</b> Lacks advanced rendering capabilities such as visual location clustering or complex custom data filtering.</p>   | [X-X] Weeks        | <p>[\$X,XXX]</p> <p>–</p> <p>[\$X,XXX]</p> |
| <b>Add-On: Product Configurator</b> | <p>Backend: Custom private Shopify app (Node.js) to process fundamental compatibility logic.</p> <p>Frontend: Custom ReactJS interface connected to the Storefront API.</p> | <p><b>Capabilities:</b> Executes a fundamental compatibility rules engine to ensure correct product combinations at checkout.</p> <p><b>Limitations:</b> Lacks a visual administrative dashboard; compatibility rules must be hard-coded and will require manual developer updates.</p> | [X-X] Weeks        | <p>[\$X,XXX]</p> <p>–</p> <p>[\$X,XXX]</p> |

| PHASE /<br>MODULE                               | IMPLEMENTATION<br>METHODOLOGY  | CAPABILITIES & LIMITATIONS  | ESTIMATED<br>TIMELINE    | INVESTMENT                       |
|---|--|---|--------------------------|----------------------------------|
| <b>Add-On:<br/>Gated<br/>Partner<br/>Portal</b> | <p>Backend:<br/>Custom logic<br/>verifying<br/>Shopify<br/>Customer API<br/>authentication<br/>status.</p> <p>Frontend:<br/>ReactJS routes<br/>protected by<br/>authentication<br/>checks.</p> | <p><b>Capabilities:</b> Restricts<br/>specified pages and assets to<br/>authenticated partner<br/>accounts.</p> <p><b>Limitations:</b> Lacks granular,<br/>enterprise-grade role<br/>management and secure,<br/>partitioned document hosting.</p> | [X-X]<br>Weeks           | \$[X,XXX]<br>–<br>\$[X,XXX]      |
| <b>Total<br/>Estimated</b>                      |  |   | <b>[Total]<br/>Weeks</b> | <b>\$(Total) –<br/>\$(Total)</b> |

SOLUTION 2

# Enterprise NestJS Architecture

Transition to a highly scalable, future-proof architecture. A single custom NestJS backend powers all content, features, and admin — Shopify handles transactions only.



FRONTEND  
**ReactJS**

A component-based JavaScript library to deliver a premium, bespoke storefront with complete design freedom and no platform constraints.



BACKEND  
**NestJS**

A progressive Node.js framework for building efficient, scalable server-side applications. Serves as the single backend for all content, user management, and business logic.



CONTENT MANAGEMENT  
**PuckJS**

An open-source visual page editor that lets the marketing team build and edit pages with drag-and-drop, with zero vendor lock-in.



COMMERCE  
**Shopify API**

Used strictly for checkout and payment processing, keeping the storefront and admin entirely independent.

SOLUTION 2

# Advantages

Key advantages of the Enterprise NestJS Architecture approach.



## Centralised Management

The Avon team operates entirely from a single custom NestJS dashboard, streamlining all content, user, and operational workflows.



## Scalability

Completely removes Shopify's technical constraints, allowing the Product Configurator and Partner Portal to operate as true enterprise-grade software.



## CMS Autonomy

PuckJS provides complete, visual drag-and-drop design freedom without relying on developer intervention for page creation.



## Infrastructure Control

Full ownership of hosting, security policies, scaling, and deployment pipelines across both the ReactJS frontend and NestJS backend.

## Timeline & Investment

| PHASE / MODULE                           | IMPLEMENTATION METHODOLOGY   | CAPABILITIES & LIMITATIONS   | TIMELINE                           | INVESTMENT       |
|--|--|--|------------------------------------|------------------|
| <b>Phase 1: Discover</b>                 | Stakeholder workshops, technical feasibility analysis, and system architecture mapping.  | <p><b>Capabilities:</b> Delivers comprehensive Business Requirements Documents (BRD) and Functional Requirements Documents (FRD), establishing a locked-in technical blueprint and roadmap prior to design.</p> <p><b>Limitations:</b> N/A</p> | 15 Jun – 26 Jun (10 Business Days) | Fixed: \$[X,XXX] |
| <b>Phase 2: UX/UI Design</b>             | Component-based UI design focused on premium user experiences, unrestricted by Shopify constraints.  | <p><b>Capabilities:</b> Provides fully approved, interactive visual prototypes optimised for development production.</p> <p><b>Limitations:</b> N/A</p>  | 29 Jun – 17 Jul (15 Business Days) | Fixed: \$[X,XXX] |
| <b>Phase 3: Core Feature Development</b> | Custom headless frontend in ReactJS. Backend powered by NestJS (with PuckJS for CMS). Connected to Shopify's Storefront API strictly for transactions. | <p><b>Capabilities:</b> Achieves a premium frontend UI. The marketing team gains full autonomy via a drag-and-drop visual CMS, managed entirely from a centralised NestJS dashboard.</p> <p><b>Limitations:</b> N/A</p>                        | Fixed: [X] Weeks                   | Fixed: \$[X,XXX] |

| PHASE / MODULE                      | IMPLEMENTATION METHODOLOGY  | CAPABILITIES & LIMITATIONS   | TIMELINE       | INVESTMENT                           |
|-------------------------------------|---|--|----------------|--------------------------------------|
| <b>Add-On: Interactive Map</b>      | <p>Backend: Custom NestJS database and administration interface.</p> <p>Frontend: ReactJS component interacting with a dynamic mapping API.</p>   | <p><b>Capabilities:</b> Delivers an optimised mapping interface featuring custom data filtering, visual location clustering, and an easy-to-use NestJS admin dashboard to update locations.</p> <p><b>Limitations:</b> N/A</p>                                 | [X-X]<br>Weeks | <p>[\$X,XXX]<br/>–<br/>[\$X,XXX]</p> |
| <b>Add-On: Product Configurator</b> | <p>Backend: Dedicated NestJS compatibility rules engine with a visual admin dashboard.</p> <p>Frontend: Custom ReactJS interface passing automated bundles to the Shopify Cart API.</p> | <p><b>Capabilities:</b> Provides a highly sophisticated visual configuration tool for users. The Avon team receives a visual admin dashboard to easily manage and update complex compatibility rules without touching code.</p> <p><b>Limitations:</b> N/A</p> | [X-X]<br>Weeks | <p>[\$X,XXX]<br/>–<br/>[\$X,XXX]</p> |

| PHASE / MODULE                      | IMPLEMENTATION METHODOLOGY   | CAPABILITIES & LIMITATIONS  | TIMELINE      | INVESTMENT                   |
|-------------------------------------|--|---|---------------|------------------------------|
| <b>Add-On: Gated Partner Portal</b> | <p>Backend: NestJS-powered user authentication and role-based access management.</p> <p>Frontend: Secure ReactJS routing and asset delivery.</p> | <p><b>Capabilities:</b> Functions as an enterprise-grade standalone software platform, featuring strict role-based asset permissions and secure document management, entirely controlled via NestJS.</p> <p><b>Limitations:</b> N/A</p> | [X-X] Weeks   | <p>[\$X,XXX] – [\$X,XXX]</p> |
| <b>Total Estimated</b>              |  |   | [Total] Weeks | <p>[\$Total] – [\$Total]</p> |

## Investment & Timeline Summary

Discovery and Design timelines and investments remain identical regardless of the selected technical architecture.

| PHASE / MODULE                           | OPTION 1<br>Headless Shopify                         | OPTION 2<br>Enterprise NestJS                        |
|--|--|--|
| <b>Phase 1: Discovery</b>                | 15 Jun – 26 Jun (10 days)<br>Fixed: \$[X,XXX]        | 15 Jun – 26 Jun (10 days)<br>Fixed: \$[X,XXX]        |
| <b>Phase 2: UX/UI Design</b>             | 29 Jun – 17 Jul (15 days)<br>Fixed: \$[X,XXX]        | 29 Jun – 17 Jul (15 days)<br>Fixed: \$[X,XXX]        |
| <b>Phase 3: Core Feature Development</b> | [X] Weeks<br>Fixed: \$[X,XXX]                        | [X] Weeks<br>Fixed: \$[X,XXX]                        |
| <b>Add-On: Interactive Map</b>           | [X-X] Weeks<br>\$[X,XXX] – \$[X,XXX]                 | [X-X] Weeks<br>\$[X,XXX] – \$[X,XXX]                 |
| <b>Add-On: Product Configurator</b>      | [X-X] Weeks<br>\$[X,XXX] – \$[X,XXX]                 | [X-X] Weeks<br>\$[X,XXX] – \$[X,XXX]                 |
| <b>Add-On: Gated Partner Portal</b>      | [X-X] Weeks<br>\$[X,XXX] – \$[X,XXX]                 | [X-X] Weeks<br>\$[X,XXX] – \$[X,XXX]                 |
| <b>Total Estimated</b>                   | <b>[Total] Weeks</b><br><b>\$(Total) – \$(Total)</b> | <b>[Total] Weeks</b><br><b>\$(Total) – \$(Total)</b> |

## Key Considerations

---

### 01 **Fixed Timeline Urgency**

To stay on track for launch, we'll need to lock in the architecture choice and contract as soon as possible.

### 02 **Timeline Adherence**

The project must progress per the proposed timeline to meet the deadline. If any delays occur, we will reassess and adjust the plan to ensure a **September 21** go-live.

### 03 **Prerequisite for Prioritisation**

During Discovery, we'll work together to agree on what gets built first, so development starts with a clear, shared roadmap.

### 04 **MVP Scope Management**

We'll keep the initial build focused on what's essential for launch. Advanced add-on features can be layered in afterwards.

### 05 **Estimation Refinement**

These estimates are high-level for now. We'll lock in exact timelines and costs once Discovery and Design wrap up. The **September 04** launch target stays firm.

### 06 **Our Recommendation**

Both solutions are priced similarly. We recommend **Solution 2** because it makes future upgrades easy. For example, the Configurator can grow with your needs without rebuilding from scratch.

## Next Steps

---

