**SaaS Platform Build Documentation for AI-Powered Beauty Commerce Suite (Lavishta)**

## 🚀 Overview

A multi-tenant SaaS platform specifically built for beauty and skincare ecommerce brands in Bangladesh and South Asia. Lavishta is the first customer. The platform provides:

* AI-generated product descriptions, long-form blogs, and FAQs using OpenAI embeddings.
* Personalized and automated Facebook/Instagram comment and inbox replies with tone settings per brand.
* UGC-driven gamification engine with automated campaign triggers, auto-close, and leaderboard visibility.
* Intelligent internal linking, auto bundles, and collection creation based on use case, seasonality, and UGC.
* Blog generation based on Google Trends, customer feedback, competitor analysis, and product discovery.
* Modular setup with toggles and usage controls per module, fully managed by the SaaS superadmin.
* Full support for both WooCommerce and Shopify, integrated through shared platform-level apps.
* All modules are independent and can be used standalone with reduced options.

## 🧠 Master AI Content Brain (Embeddings + Vector DB)

### What it Looks Like

A service layer that powers product understanding across the platform. It behaves like a brain that “understands” product context in vector space. It allows: - Relevance-based search - Smart blog internal links - Matching UGC with best products - Generating product bundles

### How It Works:

1. **Extract product attributes**:
   * Fields: title, brand, category, size, color/shade, ingredients, concerns, tags, benefits, usage.
2. **Preprocess (Tokenizer & Cleaner)**:
   * Remove symbols, standardize synonyms, deduplicate terms, fix typos (e.g., “niacinamide” vs “niacinmid”).
3. **Generate Embedding**:
   * Use OpenAI text-embedding-3-small.
   * Payload: concatenated string of full product details.
4. **Store Vector**:
   * FAISS DB with metadata: { sku, brand\_id, category\_id, size, vector, source\_text, created\_at }
5. **Use Cases**:
   * Matching products from spreadsheets (fuzzy match)
   * Suggesting related products in PDPs/blogs
   * Generating smarter product tags
   * Building personalized bundles

### How to Build It:

* Stack: Python + FastAPI + FAISS + PostgreSQL + Redis
* Scheduler: daily batch sync (or webhook trigger from Woo/Shopify)
* API:
  + POST /vectorize-product → returns vector
  + GET /similar-products?sku=... → returns top 5 cosine similarity matches
* Admin Tool:
  + Monitor failed embeds, low-confidence matches

## 📦 Modules (Micro-level Feature Breakdown)

### 2. AI Product Brain

#### 🔹 Fuzzy Matching Engine

* Matches user inputs like “hair serum 50ml” to exact products.
* Used in order imports, inbox parsing, gamified responses.
* Uses cosine similarity from FAISS.

**How to Build:** - Route: POST /match-product - Input: raw\_text, context (optional category filter) - Output: product\_id, confidence, matched\_title - Include stopword list and language model fine-tuned on Bengali + English phrases.

#### 🔹 Tag Normalizer

* Unifies variations of similar concepts: e.g. anti acne, acne-prone, pimple solution → Acne Care

**How to Build:** - Controlled vocabulary set stored in tag\_synonyms.json - NLP pipeline runs tag mapping during product import/edit - Admin UI to manage vocabulary sets

#### 🔹 Internal Linking Engine

* During blog generation, vector match finds the closest 3-5 products semantically linked to topic.

**How to Build:** - Endpoint: GET /relevant-products?blog\_topic=... - Score: Embedding similarity + sales rank + stock status - Links injected directly into GPT blog output

#### 🔹 Related Products Panel

* Displayed in blog and product pages.
* Vectors matched on usage type, concern, benefit proximity.

**How to Build:** - Frontend widget consumes /related-products?sku=... - Use caching for high-traffic items

#### 🔹 Trend Linker

* Connect Google Trends keywords with best-fit products.

**How to Build:** - Run pytrends hourly, cache rising queries - Cross-check trends vs product vectors - Suggest topics like “hyaluronic acid benefits” or “pigmentation in monsoon”

### 3. Blog Engine

#### 🔹 UGC Blog Generator

* Pull UGC comments via FB/IG APIs
* Use keywords as seed: e.g. “dark circles help”
* GPT prompt includes tone + weather + skin type relevance

**How to Build:** - Scraper: Pull comments from FB group/post, store in UGC DB - Preprocess into clusters: skin concerns, product requests, advice threads - Use Codex Prompt: *“Write expert blog for Bangladeshi women struggling with dark circles. Suggest 3 Lavishta products with benefits and usage.”*

#### 🔹 Google Trends Blog Generator

* Auto-discovers blog ideas based on trending search terms

**How to Build:** - Pull data via pytrends (rising\_topics=True) - Cross-match to vector space - Blog plan: Title, SEO meta, intro, benefits, CTA, internal links

#### 🔹 Blog Refresher

* Scores existing blogs via analytics (CTR, bounce, avg time)
* Blogs below threshold are marked “stale”
* GPT regenerates parts: intro/CTA/FAQs with updated language

**How to Build:** - Cron job to analyze blog analytics via GA4 API - Queues refresh jobs via Redis - Logs before/after content + flags for review

### 4. Gamification Engine

### What It Looks Like

A full-featured campaign manager that turns user-generated engagement into discoverability and loyalty. Campaigns are designed around Facebook/Instagram posts and comments, with AI suggesting and executing gamified prompts that reward user activity.

### Core Features:

* 🧩 Campaign Builder: Manual, Assisted (AI-suggested), and Auto modes.
* 🎯 Custom Topics: Brand admins can define dynamic campaign goals (e.g., “best skincare tips”, “your fave serum”).
* 💬 AI Prompt Generator: GPT rewords a campaign into 5 engaging variants with emojis, tone-matching, etc.
* 📈 UGC Tracker: Stores and ranks comments, likes, replies.
* 🪙 Points + Tiers Engine: Auto-calculates points per action and maps to leaderboard and website profile.
* 🛑 Auto-close Campaign: Define end time or participation limit.
* 📬 Reward Dispatcher: Auto-inboxes winners or replies with redeem instructions.
* 📊 Campaign Analytics: Engagement, CTR, virality scores per post.

**How to Build:** - DB Tables: campaigns, ugc\_entries, participants, points, tiers, comment\_logs - API: - POST /campaigns — create new - GET /campaigns/live — fetch active - POST /ugc-track — log new comment - POST /points-distribute — assign points to comment ID - Integration: - Webhook from FB/IG post comments API - Campaign trigger with post ID and tracking rules - Reward delivery via Meta Inbox Message API

**Frontend Components:** - Admin Panel → Start new campaign, pick template, tweak rewards - Leaderboard → Visible on Lavishta site and sharable - Profile → Logged-in user can see past participation, unlocked badges

**Superadmin Control:** - Enable/disable module per brand - Control template types and available reward types - Set AI access limits and GPT usage per tier

### 5. Auto Poster / Post Manager Module

### What It Looks Like:

A content automation engine that generates, schedules, previews, and posts branded content to FB/IG Pages. Powered by AI, optional Canva auto-designs, and full interval/scheduling control.

### Core Features:

* 🧠 Post Idea Generator (AI prompt per brand tone)
* ✍️ Caption Generator: multiple variants with tone control, emojis
* 🖼️ Canva-Based Image Auto Designs (optional)
* 🔗 Link Inserter: Auto-insert tracking links to blogs/products
* 📆 Smart Scheduler: Interval-based or event-triggered post queue
* 📤 Auto Publisher: Posts to FB/IG via Meta Graph API
* 🔍 Preview System: See post preview before confirming
* 📝 Mode Toggle: Auto, Assisted, Manual

### How to Build:

* Stack: Node.js + Next.js + MongoDB + Meta Graph API + Redis
* Canva Integration (via API or export template)
* AI Prompts stored and regenerated per brand tone, season
* Track post\_id, platform, engagement, clicks, generated\_by

**API Examples:** - POST /generate-post-idea → returns 3 post titles + 3 captions - POST /schedule-post → queue with time, link, image, caption - GET /preview/:post\_id → renders preview - POST /publish-now → triggers real-time post

**Frontend Components:** - Admin UI → Calendar view, post preview, schedule controls - Brand Panel → Caption tone setting, interval control - Link Tracker → UTM shortener + click logs

**Superadmin Control:** - Enable Canva Designs (Add-on only) - Set image design credits/month - Toggle auto-posting per brand - Monitor logs, rate limits, failed posts

**Post Template Examples:** - 🎁 “New Arrivals This Week” - 🧖‍♀️ “Skincare Routine in 3 Steps” - 🔥 “Monsoon Must-Haves for Oily Skin” - 📣 “Last 24 Hours of Our Offer!”

This module can run 100% independently but connects deeply with: - Blog Engine → Promote recent blogs - Gamification → Promote ongoing campaigns - Product DB → Pull new arrival lists automatically

✅ All content generation respects tone, language, audience preference, and upload interval per brand.