📘 SAAS - Ecommerce AI Documentation

In 2025, the beauty commerce landscape was evolving rapidly. Lavishta, a pioneering e-commerce brand from Bangladesh, envisioned a SaaS platform that would revolutionize how modern brands engage with their customers. Born from the internal tools and systems that powered Lavishta’s exponential growth, this AI-powered suite was designed to bring automation, intelligence, and personalized commerce to every beauty brand in South Asia and beyond. From real-time inbox replies to UGC-powered campaigns, Loyalty systems, Smart SEO, and multilingual support, every module was handcrafted with deep industry insight. This documentation tells the story of that journey — not just of code and modules, but of a community-first, AI-native ecosystem for beauty entrepreneurs.

# 📑 Index

## Superadmin

* Superadmin - Announcement Manager
* Superadmin - Auto Billing Pdf
* Superadmin - Custom Alerts
* Superadmin - Data Usage Reports
* Superadmin - Financial Reporting
* Superadmin - Incident Logs Alerts
* Superadmin - SaaS Upgrade Suggestions
* Superadmin - System Health Monitoring
* Superadmin - White-label Manager
* Superadmin Operational Panel

## Modules/Inbox

* Inbox Auto-reply Module
* Inbox Handoff Module
* Inbox Intelligence Engine
* Inbox Ugc & Auto Reply Module

## Modules/Support

* Retail Support + Issue Tracker Module
* Retail Support Module
* SaaS Customer Support + Brand Feedback System

## Modules/Loyalty

* Points Loyalty Tier Engine
* Reward Redemption Loyalty

## Modules/Gamification

* Gamified Feedback Module

## Modules/Content

* Ai Blog Meta Optimizer
* Ai Blog Refresher
* Ai Faq Builder
* Reels Shorts Auto Suggester
* Reels Shorts Generator Module
* Smart Seo Optimizer
* Video Reel Script Generator

## Modules/Marketing

* Campaign Templates Module
* Email Sms Campaign Builder
* Email Sms Campaigns Module

## Modules/AI

* Ai Setup Wizard
* Intelligent Bundles

## Modules/Search

* Smart Search Widget

## Modules/Customer

* Customer Identity Mapping Module
* Customer Mapping Engine
* Customer Upload Mapping Engine
* Customer Upload Mapping

## Modules/Analytics

* Analytics Dashboards

## Platform

* Gdpr Compliance And Api Usage Panel
* Modular Plugin System
* Platform Oauth Integration
* SaaS Platform Build Documentation for AI-Powered Beauty Commerce Suite (Lavishta)
* Shared Oauth Infrastructure Module
* White Label Rbac Module

## Developer

* Developer Toolkit Docs
* Internal Templates Editor

## Planning

* Finalized Modules Overview
* Future - SaaS Feature Suggestions

## Mobile

* Auto Poster Module
* Lavishta\_Auto\_Poster
* Mobile Admin App Lite

# Superadmin

## Superadmin - Announcement ManagerSuperadmin - Announcement Manager

# 📣 Announcement Manager Module

## 🎯 Purpose

The Announcement Manager allows the Superadmin (and optionally, brand tenants) to publish important updates, notices, maintenance alerts, and onboarding messages directly to the SaaS dashboard UI, admin panels, or email notifications.

This system streamlines internal communications, provides clarity to all system users, and supports emergency alerts and guided onboarding announcements.

## 🧩 Core Features

### 1. 📅 Announcement Types

System-wide Broadcasts

Maintenance notifications

Feature releases

Emergency alerts

Brand-specific Announcements

Individual tenant updates

Quota warnings, plan expiry, or milestones

User-targeted Messages

Role-based messages (e.g., Admin vs Editor)

Trial reminders, onboarding checklists

Database Table: announcements

### 2. 🖥️ UI Display Logic

Dashboard Top Banner (for high-priority announcements)

Modal Popups (first login, onboarding alerts)

Sidebar Highlight (e.g., red dot on “What’s New”)

In-page Inline Banners (e.g., above feature panel)

Filter by priority/expiry on brand admin UI

### 3. 🛠️ Admin Features (Superadmin Panel)

Create, Edit, Expire Announcements

Set visibility (All Tenants, Specific Brand, Roles)

Set importance level (Low, Info, Warning, Critical)

Optional CTA buttons (e.g., “Upgrade Now”, “Read Docs”)

Preview mode

View logs (seen %, user list, dismiss tracking)

### 4. 📧 Email & Notification Integration

Toggle to send email when announcement is published

Supports bulk send via transactional API (Postmark, SendGrid, etc.)

Trigger Slack/Telegram notifications (for team use)

### 5. 👨‍💻 Developer Hooks & API

REST: GET /announcements with filters (type, tenant, unread)

POST for new announcement (admin-only)

Webhook triggers (on publish, expire)

## 🧠 Benefits

Improves internal communication across teams and tenants

Reduces support tickets during downtimes/updates

Encourages usage of new features with built-in CTA

Helps brands stay aligned with system updates

## ✅ To-Do for Dev Team

Create announcements DB schema and seed data

Build superadmin UI editor for announcements

Integrate with brand dashboard header/banner logic

Add notification service (popup/email)

Build API for tenant-level announcements

## 🔗 Connected Modules

Superadmin Operational Panel

Trial Onboarding Helper

Quota Monitor Alerts

Feature Rollout Scheduler

This module will be stored as announcement\_manager.md and linked from the Admin UX & System Tools section in the master documentation.

## Superadmin - Auto Billing PdfSuperadmin - Auto Billing Pdf

# 🧾 Superadmin Module: Auto-Billing PDF Generator

## 🎯 Objective

Automate the generation, formatting, and distribution of billing PDFs for SaaS tenant subscriptions, with branding support, downloadable invoices, and financial ledger syncing.

## 📦 Use Cases

Monthly billing statements for all brands with subscription breakdown

Auto-send invoices to brand admin emails

Downloadable PDF copies in tenant dashboards

White-labeled invoice header with brand name/logo

Admin-side access to full billing history per tenant

## 🔧 Key Features

### 1. Invoice Generation Engine

Triggered on:

Subscription creation

Recurring payment (monthly/yearly)

One-time add-on purchases

Auto-generates:

Unique Invoice ID

Payment reference / transaction ID

Subscription items + quantities

Discounts, taxes, surcharges

Net total, due date, paid status

### 2. PDF Formatter

PDF template engine using HTML to PDF library (e.g., Puppeteer, wkhtmltopdf)

Layout includes:

Invoice Header (Brand name/logo, invoice number, billing cycle)

Billing To (Company info, VAT number if applicable)

Itemized breakdown table

Tax, discount, final total

Footer with SaaS legal info + support contact

Supports Bangla/Arabic/localized languages

Option to include custom notes (e.g., “Thanks for subscribing!”)

### 3. Brand Customization Layer

Tenant branding on PDF:

Logo

Invoice color scheme

Custom footer message

Business address + VAT

### 4. Delivery System

Auto-email invoice to:

Billing email

Additional finance contacts (if added)

Attachments:

PDF invoice

CSV breakdown (optional)

Option to resend manually from dashboard

### 5. Admin Dashboard Features

Filterable invoice history by:

Brand

Invoice date

Status: Paid / Unpaid / Failed / Refunded

Export to CSV

Bulk regenerate failed invoices

Link to full ledger view

### 6. Brand Tenant View

Tab in “Billing & Subscription” menu

List of all invoices

Status labels (e.g., Paid, Due, Failed)

Download PDF

Filter by month, year

## 🔒 Permissions

Superadmin: Access all billing PDF logs + controls

Brand Admin: Only their tenant’s invoices

## 📎 Integration Points

Subscription Manager (plan logic, usage tiers)

Payment Gateway APIs (Stripe, SSLCommerz, etc.)

Financial Ledger Module

White-label Manager

## 📈 Future Enhancements

🪙 Auto-embed reward/credit redemption logs

🔗 API access for external finance sync

🧾 Multi-currency invoice support

🧠 AI summarization: “This month, your cost increased due to…”

## 🧱 Tech Stack Suggestions

Backend: Node.js, Python Flask

PDF: Puppeteer, PDFKit, wkhtmltopdf

DB: PostgreSQL + S3 (for PDF storage)

Notification: Nodemailer, SendGrid, Mailgun

This module automates financial compliance and makes tenant accounting frictionless with branded PDF billing documents and seamless delivery.

## Superadmin - Custom AlertsSuperadmin - Custom Alerts

# 🚨 Superadmin Module: Custom Alert Rules

## 🎯 Objective

Allow the SaaS superadmin (or optionally, brand admins) to configure smart alerting rules based on system behavior, thresholds, or events. These alerts serve as early warnings, automation triggers, or operational insights for better control and faster response.

## 📦 Use Cases

Notify if an API quota is nearing exhaustion

Alert if any module returns error rates above a threshold

Trigger Slack/email alert if usage spikes by X% in a day

Alert if a brand’s AI cost crosses the monthly budget

Trigger notification when a customer posts the 1000th comment in a campaign

## 🛠️ Feature Components

### 1. Alert Rule Builder UI

🔧 Visual interface for defining custom alerts

🧱 Components:

Entity selector: Brand, Module, Customer, Feature, System Resource

Condition selector: >, <, =, !=, contains, not contains, etc.

Value: Numeric or keyword-based

Trigger frequency: Real-time, hourly, daily, etc.

Action: Notify via Email, Slack, System Banner, Log Only

### 2. Predefined Alert Templates

API usage quota 90% reached

Module crash 3x within 1 hour

Usage spike +200% in 24 hours

Login failure attempts > 5

AI token usage exceeds plan cap

### 3. Alert Routing

👥 Destination types:

Superadmin Email

Brand Admin Email

Slack Webhook URL

Internal DB Log + Flag UI Badge

Push notification (future)

### 4. Alert Dashboard

💡 Visual panel showing:

🔴 Active alerts

⏱️ Historical alerts log

📊 Most triggered rules

✅ Resolved alerts (dismissal logs)

### 5. Rule Logic Engine

🧠 Built-in micro-engine for evaluating alert conditions

Periodic background jobs or event listeners evaluate:

System metrics

User behavior

API response logs

Campaign engagement

## 🔒 Permissions

Superadmins can create rules for any brand/system metric

Brand Admins (if enabled) can create alerts only for their usage/data

## 📎 Integration Points

System Metrics Engine

UGC + Engagement Analyzer

AI Cost Tracker

Webhook Dispatcher (for Slack/Email)

## ✅ Benefits

Real-time awareness

Predictive issue detection

Proactive scaling or throttling

Reduces human monitoring needs

## 🧱 Tech Stack Suggestions

Backend: Node.js / Python (Celery for job queues)

Frontend: React UI for rule builder

DB: MongoDB or PostgreSQL for rule storage

Notification: Email (SMTP), Slack Webhooks, In-app Toast

## 🚀 Future Enhancements

🧠 AI-generated rule suggestions based on past incidents

🔄 Self-healing actions (e.g., restart service if crashed X times)

📈 Alert → Auto tag → Analytics drilldown

⏳ Alert decay timer: auto-dismiss if condition resolves in Y mins

This module ensures the SaaS platform remains proactive, resilient, and secure under growth, load, or misuse.

## Superadmin - Data Usage ReportsSuperadmin - Data Usage Reports

# 📁 Data & Usage Reports

## 📊 Module Overview

The Data & Usage Reports module provides deep insights into tenant-level platform engagement, feature usage, quota consumption, and user behaviors. This is essential for: - Internal decision-making - Customer success support - Feature adoption analysis - Plan adjustment and quota recommendations

It is tightly integrated with subscription, module usage, AI generation counts, campaign activities, inbox interactions, and more.

## 🔍 Core Data Categories

### 1. 👥 Tenant Activity Overview

Number of team members, roles, and login patterns

Active vs dormant brands

Last login timestamp

Country, plan type, time zone

Tables: tenants, users, logins

### 2. ✍️ AI Feature Usage

Blogs generated

Captions/posts created

Reels/shorts scripts suggested

FAQ/SEO content generated

Per feature, per tenant, per period (daily/monthly)

Tables: ai\_logs, usage\_counters, quota\_consumption

### 3. 📈 Campaign Module Usage

Number of campaigns launched

Auto vs manual mode

Number of participants

Redemption rate on reward shop

UGC import frequency

Tables: campaigns, ugc\_stats, rewards\_redemptions

### 4. 💬 Messaging & Inbox Stats

AI replies served

Human handoffs

Missed/unread queries

Response time averages

Replied via: FB/IG/WhatsApp breakdown

Tables: inbox\_messages, ai\_responses, handoff\_logs

### 5. 📦 Product & Shop Interactions

Product search queries via Smart Search

Bundle creations

Checkout redirects via campaigns

Affiliate clicks (if enabled)

Tables: search\_logs, product\_events, affiliate\_events

### 6. 📉 Quota Burn Reports

Total usage vs limits

Overuse flagged events

AI API cost (per tenant, per engine)

Recommendations for upgrade

Tables: quota\_logs, tenant\_recommendations, costs

### 7. 💵 Billing & Financial Trends

Invoices generated (count/total)

Failed payments, retries

Manual invoicing count

Avg lifetime revenue per tenant

Tables: invoices, payment\_failures, lifetime\_value

## 📁 Report Types & Access

### A. Admin-Wide Dashboards

Multi-tenant overview

Filters: Plan, country, quota usage, recent activity

Daily/Weekly/Monthly snapshots

### B. Tenant-Level Reports

Accessible to brand admins

Export options (CSV, PDF)

Triggers for email summaries

Graphs: quota usage, AI feature trends, campaign adoption

### C. Alerts & Flags

Quota warning triggers auto-email

Monthly digest to brand: “Your Top AI Wins”

Feature usage anomaly detection (e.g. sudden drop in AI use)

## 🛠️ Backend Architecture

Scheduled cron jobs for weekly usage rollups

Materialized views for fast report fetching

Redis caching for frequent admin queries

API endpoints:

/admin/reports/usage

/tenant/report/overview

/admin/reports/financial

## 🎛️ Superadmin Panel Integration

Admin page: /admin/reports

Subtabs:

Usage Summary

Quota Alerts

Top Brands

Feature Adoption

AI API Cost Analysis

Exportable, searchable, sortable interface.

## 🔐 Access Control

Superadmin: Full access

Brand Admins: Only their brand-level views

Support Staff: Read-only by permission

## ✅ To-Do List for Dev Team

Build all reporting queries & rollup logic

Create API endpoints

Link to Superadmin dashboard

Schedule usage cron tasks

Integrate export & alert system

## 🚀 Next Module: Support & Ticketing System for Brands → SaaS Admin

## Superadmin - Financial ReportingSuperadmin - Financial Reporting

# 📊 Financial Reporting & Revenue Intelligence Module

## ✨ Module Purpose

The Financial Reporting & Revenue Intelligence Module empowers the SaaS Superadmin team with in-depth visibility into earnings, forecasts, cost centers, and module-specific financial insights. It supports real-time tracking, compliance-friendly exports, and automated analytics across all revenue streams, including:

Subscriptions (monthly, annual, custom)

One-time add-on purchases

Refunds, coupon impact, VAT/GST collection

Module-level profitability

## 🔐 Core Functionalities

### 1. 💳 Revenue Dashboard (Live View)

Total MRR / ARR (Monthly/Annual Recurring Revenue)

New revenue today / this week / this month

Compare: Previous period vs current

Revenue by:

Plan (Basic, Pro, Enterprise)

Region (Country/Zone)

Payment Provider (Stripe / Paddle / SSLCommerz)

Module type (Inbox AI / Blog Writer / Campaigns)

Visuals: - Line chart: Daily revenue trend - Pie: Module revenue contribution - Bar: Plan-wise active revenue

Tables: - revenue\_daily - module\_revenue\_share - plan\_revenue\_breakdown

### 2. 📉 Forecasting + Churn Analysis

Revenue Forecast for next:

7 days

30 days

12 months (based on renewal cycles)

Predicted churned revenue (based on canceled + paused tenants)

Growth slope (based on previous 3-month average)

CAC vs LTV metrics (if connected to marketing engine)

Tables: - tenant\_renewal\_events - churn\_prediction - forecast\_model

### 3. 📦 Refund, Discount & Tax Audit

Refund volume & refund rate

Reasons: System error / customer complaint / gesture refund

Coupon usage impact:

Top-used coupon codes

Revenue loss % per code

VAT / GST collected per region (for tax filing)

Export-ready reports:

Quarterly audit

Stripe/Paddle summaries

Tables: - refund\_logs - coupon\_effectiveness - tax\_collections

### 4. ✉️ Invoicing & Payment Logs

PDF invoice links for all tenants (automated)

Payment attempts + status

Failed billing detection

Retry logs

Manual adjustments by staff (credit/debit/waive)

Tables: - invoice\_summary - payment\_attempts - admin\_adjustments

### 5. 📊 Module Profitability Dashboard

Cost vs income per feature:

AI-generated replies (OpenAI cost vs usage)

Blog/SEO tasks

Video generation (image + frame cost)

Brand usage vs cost incurred

Profit margin per tenant

Recommendation engine: flag unprofitable accounts

Tables: - module\_costs - tenant\_module\_margins - openai\_usage

## 🔨 Dev Notes

### Backend

Sync revenue + usage daily

Set up Stripe/Paddle webhooks for updates

Scheduled job for forecast calculation

### Frontend

Page: /admin/finance

Tabs: Revenue | Forecast | Refunds | Tax | Profit

Filters: Date range, region, plan, module

Export buttons: XLS, CSV, PDF

### Alerts

Revenue drop > 20% week-on-week

Refund spike

Module loss > threshold

## 🎉 Benefits

True understanding of financial health

Module-level insights for pricing strategy

Compliance-ready tax + audit exports

Helps marketing + CAC decisions

Detects high-churn or high-risk tenants

## 🚀 To-Do

Revenue daily sync job

Build forecast model with ARIMA / ML

Connect Stripe + Paddle summaries

PDF generator for invoices

Frontend graphs and visual

Hook to subscription changes

✅ Will be stored as a separate linked file under “Admin Ops Suite” Ready to cross-link with Subscription Manager, Accounting Core, and Superadmin Panel.

## Superadmin - Incident Logs AlertsSuperadmin - Incident Logs Alerts

# ❌ Superadmin Module: Incident Logs & System Alerts

## 🎯 Objective

Create a centralized system for tracking, categorizing, and notifying all infrastructure and functional incidents across the SaaS platform. This includes outages, performance degradation, security flags, third-party failures, and internal module malfunctions.

## 🧩 Feature Scope

### 1. Real-Time Incident Detection Layer

Logs system events, internal errors, third-party failures

Integrated with:

Web servers (e.g., NGINX, Apache)

App logs (Node.js, Python)

Queue workers (Celery, Redis Queue, etc.)

AI modules and webhook dispatchers

Detects:

Downtime

500 errors / API failures

Latency spikes

Failed jobs or queue overflows

Retry loops / throttling

Memory/CPU saturation

### 2. Incident Classifier

Severity tagging: LOW, MEDIUM, HIGH, CRITICAL

Auto-categorization:

Platform infrastructure

Brand-specific module

AI API limits

Integration errors (Shopify, WooCommerce, FB/IG, etc.)

### 3. Incident Logging Engine

Schema:

timestamp

service/module

description

error log (traceback/snippet)

severity

linked tenant/brand

resolved (true/false)

auto-dismiss time (if transient)

### 4. Admin Dashboard View

Global log explorer for superadmins

Filter by:

Time range

Brand

Severity

Module/Service

UI Components:

🔴 Active incidents panel

🧠 Root cause suggestions (AI-assisted)

✅ Mark resolved + Add notes

⏱️ Auto-dismiss timer panel

### 5. Notification System

Email, Slack, In-app banner

Configurable by alert level (e.g., notify only for HIGH+)

Custom escalation matrix:

Tier 1: Engineering

Tier 2: Brand Account Manager

Tier 3: SaaS Founders/Admins

### 6. Linked Actions + Automation Hooks

Integration with custom alert rules (if rules were defined for similar events)

Trigger auto-muting, log rotation, or module disable for safety

Webhook to incident response platform (PagerDuty, OpsGenie, etc.)

## 🛠️ Backend Architecture

Node.js or Python service with queue-based collector

ELK Stack or Loki/Grafana (for log parsing/visualization)

DB: PostgreSQL for indexed structured logs, S3 for raw long-term logs

## 🔐 Permissions & Logs Access

Superadmins: Full logs and alert controls

Brand Admins: Read-only logs of their incidents only (no global issues)

Engineering: Read/write with escalated automation triggers

## 🧠 Future Enhancements

AI-generated root cause insights per pattern

Anomaly heatmaps on timelines

Auto-rollbacks if incident detected on new deploys

Inferred dependencies map (which module causes ripple failures)

This module ensures high resilience, accountability, and faster mitigation across the platform through proactive system-wide observability and incident intelligence.

## Superadmin - SaaS Upgrade SuggestionsSuperadmin - SaaS Upgrade Suggestions

# 🚀 SaaS Upgrade Suggestions

## 🎯 Objective

This document compiles high-leverage upgrade ideas for enhancing the SaaS platform’s performance, flexibility, and commercial appeal. These are targeted toward long-term scalability, customer retention, operational automation, and monetization expansion. Each item includes rationale, implementation guidelines, dependencies, and estimated impact.

## 1. 🧠 Fine-tuned LLM for Cost & Performance

### ✅ Description

Replace GPT-4 with a fine-tuned smaller model (e.g., Mistral 7B or OpenChat) for repetitive tasks like tag generation, blog refresh, or inbox replies.

### 🔧 Implementation

Collect anonymized prompt-response logs

Curate training data

Fine-tune open-source base model

Host with API access via inference endpoint

### 💡 Benefits

Reduce OpenAI usage by 60–80% for common tasks

Improve response time

Add custom brand voice

### 📦 Dependencies

Prompt Logger & Data Cleaner

Model Training Pipeline

## 2. 📣 UGC → Blog → Bundle Automation Loop

### ✅ Description

Turn highly engaging user comments (UGC) into automated blog posts, then match relevant products into bundles and publish with one-click.

### 🔧 Implementation

Extract high-performing UGC with engagement scores

Summarize as insights

Feed to blog writer module with call-to-actions

AI suggests bundles (tag/ingredient-based)

Publish as landing page (blog + bundle cart)

### 💡 Benefits

Non-stop blog ideas

Organic, SEO-rich content

Drive bundle conversions

### 📦 Dependencies

UGC Analytics Module

Blog Generator + Bundle Generator

## 3. 🧱 Intelligent Collection Builder

### ✅ Description

Enable brands to auto-build SEO-friendly collections with filters, titles, meta, and internal linking.

### 🔧 Implementation

UI for collection rules (e.g., “anti-aging + Korean”)

Auto-fetch matching products via embedding search

AI generates title, description, SEO meta

Create WooCommerce or Shopify Collection

### 💡 Benefits

Increase traffic via long-tail keywords

Improves site UX for discovery

## 4. 🧠 AI Blog Refresher (Pro Version)

### ✅ Description

Enhance existing blog posts with updated stats, keywords, product CTAs, and optimized readability.

### 🔧 Implementation

Cron job to re-check Google Trends

Match blog topic + new data

Rewrite CTA/product links

Update blog content via Woo API

### 💡 Benefits

Keep SEO fresh

Avoid duplicate or decayed content

## 5. 🌍 Multilingual & Regional Adaptation

### ✅ Description

Allow each brand to auto-generate region-specific campaigns, captions, and product descriptions in Bangla, Hindi, Tamil, Urdu, etc.

### 🔧 Implementation

Enable multilingual toggle per campaign/blog

Use OpenAI or Gemini Translate API

Store user language setting

Optionally personalize posts by region

### 💡 Benefits

Unlock new markets

Cultural fit = higher engagement

## 6. 🪙 Gamification AI Auto Pilot

### ✅ Description

Allow brands to enable fully auto-generated campaigns (weekly/monthly) with performance tracking and reward scaling.

### 🔧 Implementation

Preset campaign types (comment, tag, quiz)

Schedule interval (weekly, monthly)

Auto-generate, post, score

Adjust points for underperforming campaigns

### 💡 Benefits

Engage passive brands

Continuous community buzz

## 7. ⚙️ Smart Sync: CRM + Meta + Shopify/Woo

### ✅ Description

Unify customer records from inbox, order sheets, email lists, Meta DMs, etc. into one customer identity.

### 🔧 Implementation

Fingerprint match via email/phone/DM

Merge into one unified profile

Track comment > UGC > order > support flow

### 💡 Benefits

Supercharged CRM

Enables loyalty, follow-ups, and behavior-based triggers

## 8. 🧾 Micro-Billing & Usage-Based Plans

### ✅ Description

Support usage-based pricing: charge by AI token usage, email volume, storage, etc.

### 🔧 Implementation

Track feature usage (credits, MB, counts)

Show real-time usage on dashboard

Stripe metered billing integration

### 💡 Benefits

Freemium upsell

High flexibility for agencies & growing brands

## ✅ Next Steps

Prioritize based on roadmap phase

Estimate dev sprints per feature

Bundle low-hanging upgrades into Q1-Q2 2026

This document will be linked as Upgrade Suggestions Book in the Superadmin Planning folder.

## Superadmin - System Health MonitoringSuperadmin - System Health Monitoring

# 🔄 System Health & Monitoring Panel

## ✨ Module Overview

The System Health & Monitoring Panel offers real-time visibility into platform-wide operational status, third-party API connections, scheduled task runners, and tenant-specific service health. This module is essential for proactive troubleshooting, uptime assurance, and performance diagnostics for both the superadmin and developer teams.

It integrates deeply with each module’s job queues, token proxies, webhook listeners, and sync processors.

## 🔧 Core Monitoring Sections

### 1. 📅 Cron Jobs & Queue Runners

Scheduled Tasks: Blog generator, Sync engines, Weekly summaries

Queues: Email jobs, Webhook processors, AI generation

Track:

Job name

Last run

Next due

Fail count

Retry queue length

Tables: - cron\_jobs - job\_queue - job\_failures

Alerts: - If retry queue > 10, flag in red - If job not run in >1hr, show warning

### 2. 🔧 API & Token Health

FB/IG Token Expiry Check (per tenant)

WooCommerce Token validity

WhatsApp Webhook test (ping)

GPT Token/API Quota Status

Tables: - token\_validations - token\_expiry\_logs - token\_alerts

Integration: - Auto-run every 12 hours - Add failure summary to Superadmin Inbox

### 3. 🚀 Uptime & Latency

Ping all tenant subdomains (e.g., brand1.myplatform.com)

Response time histogram

DNS/TLS check failures

Webhook delay monitor

Tools: - Built-in HTTP pinger - Store ping logs (last 7 days) - Alert on: - 2+ fails in 10min - >500ms average latency

### 4. ⚠️ Error Logs (Realtime)

Collect error logs from all services:

AI Gen errors

Post failures

Stripe failures

App crashes

Group by severity: [Info, Warning, Error, Critical]

Include tenant & feature flags

Stack: - Store in centralized error\_logs table - Push critical logs to Slack/Webhook - Allow per-feature filter in UI

### 5. 📊 Resource Usage Metrics

CPU, Memory (server-level)

API calls used per GPT provider

Image generation load

Peak hour charts (per 24h)

Sources: - VPS Prometheus or Light dashboard - AI Token usage logs - Background queues utilization

### 6. 🔹 Tenant Sync Status

See each tenant’s:

Last blog sync

Last product update

UGC import success/fail

Campaign launch history

Highlight stuck/inactive tenants

Tables: - tenant\_syncs - sync\_failures

## 📅 UI & Display Design

Page: /admin/system-health - Tabs: - Jobs - Tokens - Uptime - Logs - Metrics - Syncs

Default Widgets: - [!] Critical Errors Today (count) - [🚫] Expiring Tokens - [🔹] Queued Jobs Pending - [📊] API Usage Stats - [🚨] Tenants with Missed Syncs

## 🔈 Notification Integrations

Slack channel (admin): new critical error, failed sync, token expired

Email every 24h: health summary

WhatsApp optional alert (superadmin toggle)

## 🔌 Admin Actions

Manually retry stuck job

Ping token/connection again

Mark incident as resolved (logs)

Add note to health log for incident

## ✅ Benefits

Proactive alerting before customers report

Reduce downtime across AI features, syncs, inbox

Helps dev team debug system bugs faster

Tracks trends across all tenants/platform usage

## 📆 Dev To-Do

Setup all log tables with rotation

Implement centralized job queue logger

Build UI widgets using reusable Admin Template

Connect with Token Proxy and Cron Modules

Add Slack alerting via webhook

## 🌎 Future Upgrades

Add historical performance charts (last 30 days)

Auto-restart failed jobs (with max retry)

Tenant-level health score (display on Tenant Card)

✅ This file is finalized and ready to be linked under Admin Ops + Infra Monitoring group.

## Superadmin - White-label ManagerSuperadmin - White-label Manager

# 🏢 Superadmin Module: White-Label Manager

## 🎯 Objective

Enable the SaaS platform to operate in fully white-labeled mode per brand/tenant — allowing each customer to offer the solution as their own branded platform. This increases perceived ownership, premium appeal, and market flexibility.

## 🧩 Core Capabilities

### 1. 🔖 Brand Identity Manager

Allow each brand to upload and manage:

Logo (horizontal and square)

Favicon

Primary/Secondary color palette

Typography (Google Fonts)

App/Meta Title

Custom Footer Text or Copyright

### 2. 🌐 Custom Domain Mapping

Each tenant can map their own subdomain (e.g., portal.brandname.com)

SSL certificate auto-provisioned (via Let’s Encrypt)

CNAME setup guide UI + verification check

### 3. 🧠 AI Voice Customizer

Inject each brand’s tone/voice into AI outputs across:

Blog content

Inbox replies

Campaign captions

Product suggestions

Support brand instructions for LLM prompt injection (stored per tenant)

### 4. 🛠️ UI Skin Engine

Each white-labeled portal adapts to brand’s colors/logo

Login screen branding + welcome copy

Navbar + dashboard elements auto-match the tenant

### 5. 📦 Optional Feature Toggling

SaaS owner (superadmin) can enable/disable:

Modules per tenant (e.g., Gamification, Blog)

Advanced/Pro features (e.g., SEO auto refresh)

Usage limits and quotas per plan tier

### 6. 🔐 Access Control Labels

White-labeled tenants see only their own users, data, and branding

No OpenAI or platform branding unless explicitly opted in

### 7. 📬 Custom Email Sender + Templates

Brands can:

Set their own SMTP or verified sending email

Customize email headers, footers, and disclaimers

Override system-wide templates with branded ones

### 8. 📱 App Store + SDK Branding (Optional Phase 2)

Support for:

White-labeled mobile apps (iOS/Android)

App icons, splash screen, login theme

Embed widgets with per-brand tracking ID

## 👨‍🔧 Superadmin Controls

🔁 Force Reset Branding

🧪 Preview Mode: See what each tenant UI looks like

🧰 Diagnostics: Detect missing favicon, misconfigured SMTP, broken CNAME

🚨 Rebrand Reset Warnings if switching tenant visuals

## 🧱 Tech Architecture

Frontend: Tailwind + React with dynamic theme injection

Storage: S3 / Firebase for brand assets

Domains: Cloudflare API for DNS checks (optional)

Backend: Tenant context injection for theme overrides

DB: tenant\_settings, branding\_assets, custom\_prompts, email\_templates tables

## 🔄 Integration Points

AI Prompt System → Inject brand voice into replies

Email System → Brand template overrides

Admin Panel → Module toggles + subscription plans

Mobile App Builder (optional)

## ✅ Benefits

Higher brand affinity

Reseller or agency appeal

Trust boost for end users

Unlocks white-label business model

## 📈 Future Extensions

🔄 Multi-brand toggler (agencies managing >1 brand)

🧩 Per-feature micro branding (e.g., watermark per post)

🛒 Marketplace listing as white-label solution

🖼️ Brand asset CDN delivery for speed

This module enables powerful white-label SaaS deployments while preserving system stability, control, and upgrade consistency.

## Superadmin Operational PanelSuperadmin Operational Panel

# 🧑‍💼 Superadmin Operational Panel

## ✨ Module Overview

The Superadmin Operational Panel is the centralized control room of the entire SaaS platform. It empowers the SaaS owner (you) and your core team to:

Oversee all customer subscriptions and accounts

Approve/cancel/pause billing or module access

Monitor revenue, usage, and technical performance

Adjust limits, pricing tiers, API quotas per brand

View logs, alerts, and system errors

Manage white-label settings, app tokens, and permissions

This is not visible to any SaaS customer and is strictly used by the platform operator.

## 🔐 Core Functional Areas

### 1. 📆 Subscriptions & Billing Manager

View all active, trial, paused, and canceled brands

Search brands by name, email, plan

See plan: (Basic, Pro, Enterprise, Custom)

View modules subscribed per brand

Modify plan manually (with reason note)

Pause/cancel/reactivate

Add credit, discount, refund (with log)

Integrations: Stripe, Paddle, SSLCommerz, Manual Billing API

Logs: - subscriptions - subscription\_actions - subscription\_modules

### 2. 📊 Accounting & Financial Dashboard

Total monthly recurring revenue (MRR)

Revenue by plan/module

Refunds, failed payments

CSV export by month or quarter

Tax breakdown per country

Daily / weekly growth chart

Visuals: Charts for: - Net MRR, Gross Revenue, Refunds - Plan-wise subscriber counts - New signups, churns

Tables: - payment\_logs - invoice\_summary

### 3. 🛰️ System Health + Background Jobs

Status: Facebook API, Instagram, OpenAI, WooCommerce, Shopify

Show downtime/errors with timestamp

Last API ping + response time

Background task queue:

Embeddings

Blog generation

Syncs

Alerts: - Delayed syncs >5min - Failed blog jobs - FB token errors

Tables: - job\_queue - api\_ping\_logs - system\_failures

### 4. 📊 Usage Analytics Panel (Per Brand)

API token usage

Image/Video generation count

Blog/FAQ created

Inbox replies auto vs manual

Storage used

Current plan vs usage limits

Auto-flag overuse

View: - Table: Top 10 brands by usage - Export: Per-brand usage sheet

Tables: - usage\_metrics - plan\_limits

### 5. 🏛️ Module Control Center

See which brand has access to which modules

Toggle enable/disable instantly

Set module price overrides

Set feature flag (beta, legacy, hidden)

View module changelog

Global module setting (on/off globally)

### 6. 🏢 White-Label Manager

Brand name, logo

Custom domain (CNAME status)

Support email and contact info

Toggle visibility of “Powered by SaaS”

Change UI color scheme per brand

Tables: - brand\_settings - white\_label\_domains

### 7. ❌ Incident Logs & System Alerts

Logs of system errors per module

Repeated token errors or quota breaches

Integration failures

Retry queue logs (job retried 3 times, etc.)

Mark resolved + internal comment

### 8. 🔓 Permission Control & Logs

Superadmin staff access:

Owner

Developer

Billing Manager

Customer Success

Invite/remove internal staff

Grant temporary access to dev team (with expiry)

View access log by IP/time/device

Tables: - admin\_users - admin\_logs - admin\_roles

## 🔗 Integrations

Stripe / Paddle API

Facebook / Meta App Token Status

OpenAI usage API

Webhook retry services

Manual billing/credit adjustment API

## 💡 Suggested UX Design

Left-side menu: Dashboard, Subscriptions, System, Logs, Alerts

Topbar: Quick search, notifications, dark mode

Usage Graphs: Cards + toggle (day/week/month)

Toggle switches for modules

Tabs for plan switch, refund, limit change

## 🛠️ Optional Extensions

Auto-billing PDF generator

Audit log viewer by time range

Mobile admin app (Lite)

Developer CLI access + GraphQL explorer

## 🎉 Outcome

This panel becomes the heart of operations: - Makes scaling efficient - Provides real-time visibility into revenue + risk - Simplifies debugging brand issues - Controls modular access in one place - Enables internal team collaboration

✅ Now documented as a separate file. Ready to link in master doc + cross-referenced with Customer Support, Identity, and Billing modules.

# Modules/Inbox

## Inbox Auto-reply ModuleInbox Auto-reply Module

Module: Inbox Auto-Reply System (Unified for Facebook, Instagram, WhatsApp)

## 🧩 Purpose

To intelligently respond to customer inquiries in FB Page inbox, Instagram DMs, and WhatsApp using brand-specific tone and context-aware AI. The system aims to:

Reduce manual load on customer support.

Speed up query handling with GPT-generated replies.

Understand product names, concerns, order-related queries.

Support multilingual and tone-personalized replies.

Enable fallback to human (Bot-to-Human Handoff).

## 🎯 Key Features

### 🔹 Unified Inbox Connector

Centralized webhook listener for Meta Messenger, Instagram DMs, and WhatsApp Cloud API.

Detects which platform and Page/Brand message originated from.

### 🔹 GPT-Powered Smart Reply Engine

Triggers reply suggestion using:

Customer message

Brand tone/preset

Product catalog embeddings

Order sheet (if connected)

Example Prompt: > “Customer says: ‘Do you have anything for acne?’ | Reply as Lavishta in expert, helpful tone. Include 1-2 matching products, usage tips, and ask a follow-up.”

### 🔹 Auto Mode + Assist Mode

Auto: Replies sent instantly unless low confidence or fallback.

Assist: Suggest replies in panel for human agent approval.

Confidence threshold adjustable per brand.

### 🔹 Language & Tone Settings

Language Detection (Bengali, English, mixed)

Tone per brand: expert, friendly, premium, casual

Multilingual GPT fine-tuning per tone/language combo

### 🔹 Order-Specific Logic

If order number or tracking request detected → trigger Woo/Shopify order lookup

If product name detected → trigger vector match and include product info

### 🔹 Bot-to-Human Escalation

If GPT returns low confidence OR customer replies “talk to human”

Assigns to human team (visible in panel)

Marks conversation with ⚠️ flag

### 🔹 Analytics & Logs

Average response time

AI vs Human reply ratio

Top product queries

Top concern topics (for blog engine/SEO)

## 🔧 How to Build

### 💬 Messaging Listener

Meta Webhooks for: messages, message\_reactions, story\_reply, message\_reads

WhatsApp Cloud API listener for /v1/messages

Store each incoming message in inbox\_messages DB with:

{ customer\_id, channel, message\_text, intent, vector, response\_type, assigned\_to, created\_at }

### 🧠 AI Reply Generator

Use GPT prompt template per brand tone

Include previous messages as context window

Include top 3 vector-matched products

Match for FAQs (“Do you deliver in Sylhet?”, “Is this original?”)

Return: response\_text, confidence, matched\_product\_ids, response\_mode

### 🧑‍💼 Agent Panel

Inbox view with auto/assist/manual toggle

Approve/Reject/Edit suggested reply

Assign to human, mark as resolved

Emoji badge for AI-assisted reply

### 🛠️ Superadmin Settings

Enable/disable channel (Messenger, IG, WhatsApp) per brand

Set reply limits per day/week (for free vs paid tiers)

View per-brand GPT token usage

Adjust fallback threshold per brand

See all flagged conversations across brands

## 📎 Integration Points

🧠 Product Brain: For vector match of product name in inbox

🔗 Blog Engine: Suggest blog link in response if matched topic

🛍️ Order DB: Lookup past orders if message includes phone/invoice ID

🎮 Gamification: Inbox campaign prompts (e.g., comment to win)

📬 Email/SMS Module: Trigger follow-up if inbox → order confirmed

## 🧪 Example Flow

Customer sends: “What’s good for oily skin and under 1000?”

System:

Detects intent: product recommendation

Uses embeddings: finds 2 matching products under budget

GPT prompt crafted with tone “Expert”

Auto reply sent: “Here are two great picks for oily skin under ৳1000…”

Logged and marked as resolved

## 🧩 Future Add-Ons

Voice-based inbox (with Whisper transcription)

CTA buttons in replies (Open website, View product)

Smart appointment scheduling via Messenger/WA

Smart CRM tagging via inbox behavior

## Inbox Handoff ModuleInbox Handoff Module

# 🤖 Bot-to-Human Inbox Handoff Logic

## ✅ Module Overview

The Inbox Handoff module is a logic-based escalation layer built on top of the AI Smart Inbox (Messenger, Instagram DM, WhatsApp). It detects when AI responses are insufficient, sensitive, or repeatedly ignored, and automatically escalates to a human staff member. This ensures smooth transitions without losing conversation context or history.

## 🌟 Goals

Prevent customer frustration due to robotic replies

Route high-intent or unresolved queries to humans

Log every escalation with audit trail and timestamps

Let staff reply inside the same shared inbox view (or via assigned apps)

Enable hybrid bot-human conversation loops

## 🧪 Core Features

### 1. ✨ Smart Escalation Triggers

Rule-based triggers:

User repeats a question 2+ times

Keywords like “need help”, “speak to agent”, “wrong order”

Any negative sentiment detected via NLP

No AI reply sent within 15s

Confidence-score trigger:

If the AI reply confidence score is below a set threshold (e.g., 0.7)

Manual trigger:

Agent or admin can tag conversation to “Force Handoff”

### 2. 📢 Handoff Notification System

Staff Notification Options:

Email

WhatsApp internal group

Platform dashboard bell icon

Payload includes:

Customer name, message history (last 5)

Channel (Messenger / IG / WhatsApp)

Trigger reason

Time since message

### 3. 🛋️ Staff Assignment Engine

Modes:

Round-robin: among available agents

Priority routing: VIP customers go to senior agents

Manual override: Admin can assign specific inbox to someone

Staff View:

Web inbox module with:

Message view + reply

Customer profile (from mapping system)

AI reply suggestion (optional)

Mark as resolved / escalate further / tag

### 4. 🔒 Audit Logging

Logs every:

Trigger event

Escalation assignment

Staff replies and handback

Resolution timestamp

Stored in inbox\_handoff\_logs with:

message\_id, trigger\_type, assigned\_to, response\_time, channel, customer\_id

### 5. ➡️ Rejoin Bot Flow

After human resolves:

Agent marks thread as “Resolved”

Bot auto-responds with follow-up: > “Glad our team could help! Let me know if you have more questions.”

Conversation returns to AI mode unless manually locked

## 🔄 Integration Flow

✉ Messenger/IG/WhatsApp webhook receives message

🤖 AI replies if confidence score ≥ 0.7

⚠️ Otherwise, checks escalation rules

⚖️ If triggered, logs event, sends handoff alert, assigns agent

📢 Agent notified, joins inbox, sees full thread

💬 Agent resolves, hands back to bot

## 🔧 Technical Requirements

### Tables:

inbox\_conversations

inbox\_handoff\_logs

staff\_profiles

customer\_identity\_map

### APIs:

NLP Sentiment API

Confidence Engine

Staff Assignment API

Messenger / IG / WhatsApp webhook processors

## 📊 Metrics & Insights

% of conversations escalated

Avg. time to human response

Top 5 escalation triggers

Staff resolution time per agent

Bot re-engagement success rate

## 🚀 Benefits

Zero dead ends for customers

Preserves AI value with fallback safety net

Human + bot hybrid model for scalable support

Full audit trail for quality assurance

✅ Ready to export or connect with Inbox Auto-Reply module. Let me know your next step.

## Inbox Intelligence EngineInbox Intelligence Engine

Inbox Intelligence Engine – Technical Module Documentation

## 🧠 Overview

The Inbox Intelligence Engine is a core module designed to intelligently parse, identify, and map incoming customer conversations (in Messenger, Instagram DMs, and WhatsApp) to known customer identities, previous orders, and website accounts.

This engine supports personalization, order syncing, and real-time contextual replies — all by understanding who the user is based on their past history.

## 🎯 Core Goals

Identify users based on message content, name, phone, email, or order references.

Track past interaction history across platforms.

Link social media profiles to WooCommerce/Shopify customers.

Extract actionable data from conversations (e.g., complaints, interests, feedback).

Trigger workflows based on message content.

## 🔧 Technical Architecture

### 1. Message Collector Layer

Meta API Webhooks to collect:

Messenger thread messages

Instagram DM messages

Story replies, mentions, etc.

WhatsApp Business API (via provider like Twilio/360Dialog)

Standardized schema:

{  
 "platform": "facebook\_messenger",  
 "message\_id": "abc123",  
 "sender\_name": "Rafi",  
 "sender\_id": "fb:1000123",  
 "text": "Hi, I placed an order #4523 but didn’t get a call",  
 "timestamp": "2025-07-10T06:15:00Z"  
}

### 2. Entity Extraction Engine

#### NLP Pipeline

Name detection (match to user database)

Phone/email regex match

Order reference pattern (e.g., #1234, Order ID 9911)

Sentiment & intent classifier: complaint, feedback, inquiry, product\_request

Customer classifier confidence score

result = {  
 "matched\_customer\_id": 4215,  
 "confidence": 0.88,  
 "intent": "complaint",  
 "order\_id": 9911,  
 "matched\_fields": ["phone", "name"]  
}

### 3. Customer Mapping Graph

Build a Customer Identity Graph:

FB sender ID ↔ website customer ID

WhatsApp number ↔ previous order

Email from inbox ↔ user profile

Cross-links from CRM uploads

Data store: Neo4j or graph-based PostgreSQL table

Visual in Admin Panel for merged profiles

### 4. Inbox Enhancer UI

#### Sidebar Component

Shown inside unified Inbox UI:

Customer Name + Tier (Gold/Silver)

Location (from order history)

Past purchases

Tags (e.g., VIP, Regular, Refund Complainer)

Message insights (intent, sentiment)

#### Smart Reply Suggestions

GPT-generated replies based on past orders

Action button to:

Mark as complaint

Trigger re-delivery workflow

Send discount code

## 🔁 Workflow Example

Rafi sends DM: “My order 9911 didn’t arrive. Can I reorder or get help?”

Message ingested via webhook

NLP detects order ID, sentiment = complaint

Matched to WooCommerce Customer rafi.hossain@gmail.com

Sidebar shows past 3 orders, last refund history

GPT recommends reply: “Hi Rafi! I’m so sorry about your experience. Let me look into order 9911 and make this right. Can I offer you a free replacement or refund?”

Agent selects reply + triggers delivery retry

## 🧰 Tech Stack

Backend: Python (FastAPI)

NLP: spaCy + OpenAI + regex pipeline

Vector Store: FAISS for matching message context

Customer Store: PostgreSQL + Neo4j for mapping

Frontend: React (inbox plugin), connects to shared CRM

## 🔒 Privacy & Compliance

All messages encrypted in storage

GDPR-compliant consent + data usage log

Opt-out handling and masking for sensitive queries

## 🔑 Superadmin Controls

Enable/Disable this module per brand

Define platform access (FB only, or WhatsApp too)

Monitor mapping accuracy rates

Set confidence thresholds for automatic actions

## 🧪 Add-On Ideas

Conversation summary timeline

Auto ticket creation for unresolved DMs

Voice note transcription (via Whisper/OpenAI)

UGC sentiment graph (weekly mood tracker)

✅ Ready for integration with:

Gamification Module (for identity mapping)

Product Recommender (based on messages)

Loyalty Program (auto reward for helpful DMs)

Complaint Tracker

## Inbox Ugc & Auto Reply ModuleInbox Ugc & Auto Reply Module

📨 Facebook Page Inbox Auto-Reply & UGC Inbox Mining Module

## Overview

This module is the foundation for conversational AI inside the Facebook Page inbox. It allows brands like Lavishta to:

Respond to incoming DMs with smart replies using GPT.

Understand user intent (e.g., order status, product query, skincare help).

Log useful UGC content (e.g., “I love this toner!”, “Any solution for acne scars?”)

Tie inbox users to known customers (via previous orders, uploaded customer base, or CRM).

All this operates in one of 3 modes per brand: - ✅ Manual – staff reply manually, but system logs UGC and customer match. - 🧠 Assisted – AI drafts reply, human approves. - ⚙️ Auto – GPT replies automatically, logs everything.

## How It Works

### 1. Webhook Connection to Page Inbox

Subscribe to messages, message\_deliveries, message\_reads, messaging\_postbacks from Meta.

Store inbox messages in inbox\_messages table with sender\_id, message, timestamp, matched\_customer\_id, etc.

### 2. Intent Detection Pipeline

Use OpenAI GPT-4 Turbo with system prompt: > “You are a skincare advisor for a Bangladeshi beauty brand. Classify the intent of this message.”

Output: intent (product\_query, order\_status, advice, complaint, general)

Save with each message.

### 3. UGC Mining Engine

Filter messages with positive tone or product experience.

Store in ugc\_quotes table with fields:

message

product\_match (via vector match)

sentiment

used\_in\_blog: boolean

### 4. GPT Reply Generator

Pre-fill reply based on intent + customer history:

“Hi [Name], thanks for loving our Niacinamide Serum! It works even better when paired with [Product]. Want the full routine?”

GPT uses:

Brand tone settings

Product vector database

UGC context

Replies returned with confidence score + editable field.

### 5. CRM Matching Logic

Match user by:

Phone/email collected from message

Prior comment records

Uploaded customer base

Update inbox\_customer\_map table with sender\_id, customer\_id, source, match\_score

### 6. Personalization Rules per Brand

Brands can define via panel:

Reply tone: friendly, expert, short, detailed

Allowed GPT intents (e.g., auto-reply only for product\_query)

Enable/disable logging of sensitive UGC

### 7. Admin Dashboard

Shows:

Inbox threads with filters (intent, tone, match)

GPT reply suggestions (approve/send/edit)

Stats: avg response time, UGC mined, automation %, match rate

## Technical Stack

Webhook Server: Node.js or Python FastAPI

GPT Calls: OpenAI SDK (Turbo)

Storage: PostgreSQL + Redis queue

CRM Match: Fuzzy match using email, phone, or full name vector

Admin Panel: Vue/React + Tailwind

## Permissions / Facebook App Setup

Platform-level Meta App (not brand-owned)

Require pages\_messaging, pages\_manage\_metadata, pages\_read\_engagement

Brand admins approve connection in 1 click from settings panel

## Superadmin Controls

Enable inbox module per brand

Set GPT usage limits

Logs all GPT replies + UGC matches

Force manual mode for certain tiers

## Use Cases

Boost UGC collection without extra campaigns

Faster customer replies (esp. for basic queries)

Automatically turn inbox messages into blog ideas or testimonials

Grow understanding of customer needs

Next Module to Document: Messenger + Instagram + WhatsApp Replies Combined Inbox Feature?

# Modules/Support

## Retail Support + Issue Tracker ModuleRetail Support + Issue Tracker Module

# 🛠️ Retail Support + Issue Tracker Module

## ✅ Module Overview

This module manages support tickets and issue tracking for retail customers (end-users of SaaS brand clients) via social channels like Facebook and Instagram, as well as web/app support interfaces.

The system allows both AI and human agents to: - Create issues based on inbox complaints or comments - Track progress of issues (open, pending, resolved, etc.) - Automate follow-ups and customer updates - Assign tickets to agents - Generate insights on issue types, resolution times, and agent performance

This works alongside brand-managed customer service teams, while optionally integrating with their CRM or inbox tools.

## 🎯 Key Goals

Provide structured customer service to Facebook/Instagram users

Automate reply + resolution steps using AI

Enable human agents to intervene where needed

Maintain brand loyalty through clear and prompt responses

Allow customers to track issue progress via web profile or inbox messages

## 🧠 Feature Set

### 1. Issue Detection & Creation

Auto-detect issue tone from FB/IG comments/messages

Detect keywords like “problem,” “refund,” “wrong product”

Allow AI to create issues with:

Summary

Customer details (matched from mapping engine)

Platform + timestamp

Manual issue creation via agent panel

### 2. AI Response & Follow-Up Engine

Auto-reply: “We’ve created a support ticket. Here’s your issue ID.”

Auto-assign urgency tags: High / Medium / Low

Schedule follow-up messages (e.g., “We’re still checking on your issue…”)

Use ChatGPT-style tone templates per brand

Provide step-wise updates (“Shipped again”, “Refund issued”)

### 3. Agent Panel

List of open/closed issues with filters

Chat view of original message + replies

Notes & internal comments

Assign/reassign agent

Mark resolved with action summary

### 4. End-User Visibility (Web Profile)

Logged-in retail users can:

See list of their support issues

Track status (pending, shipped, resolved, etc.)

Add comments/attachments

Optional: send email/SMS/inbox updates automatically when status changes

### 5. Analytics & Reports

Avg. time to resolve per category

Peak complaint sources (platform, agent, product, etc.)

Monthly summary for brand owners

Agent scorecard (speed, quality)

## 🧩 Integrations

Facebook/Instagram inbox parser

WooCommerce / Shopify (to validate order ID, refund status)

CRM or Customer Mapping Engine

ChatGPT for AI response templates

## ⚙️ Technical Stack & Tables

### Tables:

support\_issues

support\_comments

support\_agent\_assignments

support\_issue\_templates

support\_resolution\_log

### APIs / Services:

AI Tone Generator (GPT)

Social Inbox Ingestion API

Customer Identity Mapper

Webhook triggers for status updates

## 🔐 Permissions

Retail Users: View their own issues, comment

Brand Support Staff: Create, edit, resolve tickets

Superadmin: See global logs, manage agent permissions

## 💡 Optional Extensions

SLA timer with escalation

CSAT (Customer Satisfaction) score capture after resolution

WhatsApp issue reporting support

✅ Module documented and ready for export/integration with master suite. Let me know which module you want next.

## Retail Support ModuleRetail Support Module

# 🛠️ Retail Customer Support & Issue Tracker (for SaaS Brands’ Customers)

## ✅ Module Overview

This module enables SaaS brands to manage support issues raised by their end customers (retail buyers) via social platforms (e.g., Facebook, Instagram) with AI-powered ticket creation, assignment, automated updates, and reporting. It helps streamline support workflows inside Facebook/Instagram inboxes without the need for external support tools.

## 🎯 Goals

Track and resolve customer issues raised in FB/IG messages or comments

Allow staff to assign, resolve, and follow up using internal dashboards

Empower AI to detect, log, and respond to basic issues

Build support history and performance metrics per brand

## 🔄 Workflow Summary

### 1. Issue Raised

AI detects negative sentiment or complaint from:

FB comment (e.g., “Didn’t get my order”)

Messenger/Instagram DM (e.g., “Package wrong!”)

AI replies instantly: “So sorry to hear that! I’ve flagged this to our team. We’ll update you soon.”

### 2. Support Ticket Creation

System creates a ticket:

Customer Name, Platform, Message

Order history (if matched)

Type: Delivery Issue / Product Defect / Refund Request, etc.

Auto-tags: Priority, Repeat Complaint, Escalated

### 3. Assignment & Tracking

Admin/staff assigns issue to relevant person

Internal notes added (not visible to customer)

Priority & status tracked: Open > In Progress > Resolved

### 4. Customer Follow-up

AI sends follow-up if no staff reply in X hours

Staff can send manual updates (via Messenger reply)

AI optionally checks back: “Everything okay now? 😊”

### 5. Resolution & Feedback

Ticket marked closed by staff/AI

Option to collect satisfaction feedback via Messenger emoji or message

## 🤖 AI Capabilities

Complaint detection via NLP + sentiment engine

Intent tagging (issue classification)

Auto-reply suggestions

Delay-based follow-up automation

Complaint pattern analysis

## 🧩 Integrations

Inbox AI: For real-time complaint detection

CRM & Order Mapper: Attach past purchase/order info

UGC Tracker: Check previous interactions

## 📊 Reporting Metrics

Daily ticket volume

Resolution time averages

Unresolved ticket alerts (age > 24h)

Escalated issues

Staff performance dashboard

## 🖥️ UI Components

### Staff Dashboard

Ticket Inbox: Filters by Status, Staff, Type

Issue detail pane: Timeline + Internal notes

Quick action buttons: Reply / Close / Escalate / Assign

### Admin Panel

View ticket backlog by brand

Set auto follow-up durations

Customize tags and categories

### Superadmin View

Monitor all brands’ complaint stats

No customer PII access

Exportable brand-wise metrics

## 🔐 Permissions

Agent: Create/update/resolve assigned tickets

Admin: Assign, tag, escalate

Superadmin: Metrics only, not data access

## 📦 Storage Tables

support\_tickets

ticket\_notes

ticket\_tags

customer\_issue\_map

ticket\_resolution\_logs

## 🔗 Future Extensions

AI-generated FAQ reply suggestions

Auto-merge repeated issues

Customer support satisfaction tracking per brand

✅ Ready for API + code-level integration planning.

## SaaS Customer Support + Brand Feedback SystemSaaS Customer Support + Brand Feedback System

# 🛠️ Retail Support + Issue Tracker Module

## ✅ Module Overview

This module manages support tickets and issue tracking for retail customers (end-users of SaaS brand clients) via social channels like Facebook and Instagram, as well as web/app support interfaces.

The system allows both AI and human agents to: - Create issues based on inbox complaints or comments - Track progress of issues (open, pending, resolved, etc.) - Automate follow-ups and customer updates - Assign tickets to agents - Generate insights on issue types, resolution times, and agent performance

This works alongside brand-managed customer service teams, while optionally integrating with their CRM or inbox tools.

## 🎯 Key Goals

Provide structured customer service to Facebook/Instagram users

Automate reply + resolution steps using AI

Enable human agents to intervene where needed

Maintain brand loyalty through clear and prompt responses

Allow customers to track issue progress via web profile or inbox messages

## 🧠 Feature Set

### 1. Issue Detection & Creation

Auto-detect issue tone from FB/IG comments/messages

Detect keywords like “problem,” “refund,” “wrong product”

Allow AI to create issues with:

Summary

Customer details (matched from mapping engine)

Platform + timestamp

Manual issue creation via agent panel

### 2. AI Response & Follow-Up Engine

Auto-reply: “We’ve created a support ticket. Here’s your issue ID.”

Auto-assign urgency tags: High / Medium / Low

Schedule follow-up messages (e.g., “We’re still checking on your issue…”)

Use ChatGPT-style tone templates per brand

Provide step-wise updates (“Shipped again”, “Refund issued”)

### 3. Agent Panel

List of open/closed issues with filters

Chat view of original message + replies

Notes & internal comments

Assign/reassign agent

Mark resolved with action summary

### 4. End-User Visibility (Web Profile)

Logged-in retail users can:

See list of their support issues

Track status (pending, shipped, resolved, etc.)

Add comments/attachments

Optional: send email/SMS/inbox updates automatically when status changes

### 5. Analytics & Reports

Avg. time to resolve per category

Peak complaint sources (platform, agent, product, etc.)

Monthly summary for brand owners

Agent scorecard (speed, quality)

## 🧩 Integrations

Facebook/Instagram inbox parser

WooCommerce / Shopify (to validate order ID, refund status)

CRM or Customer Mapping Engine

ChatGPT for AI response templates

## ⚙️ Technical Stack & Tables

### Tables:

support\_issues

support\_comments

support\_agent\_assignments

support\_issue\_templates

support\_resolution\_log

### APIs / Services:

AI Tone Generator (GPT)

Social Inbox Ingestion API

Customer Identity Mapper

Webhook triggers for status updates

## 🔐 Permissions

Retail Users: View their own issues, comment

Brand Support Staff: Create, edit, resolve tickets

Superadmin: See global logs, manage agent permissions

## 💡 Optional Extensions

SLA timer with escalation

CSAT (Customer Satisfaction) score capture after resolution

WhatsApp issue reporting support

# 🧰 SaaS Customer Support + Brand Feedback System (Internal)

## ✅ Module Overview

This module is designed for brands (SaaS clients) to: - Submit bug reports, feature requests, or operational complaints to the SaaS platform team - Get support with onboarding, integrations, billing, and account issues - Receive updates and resolutions from the SaaS team

This allows the SaaS admin (you) to: - Track, triage, and respond to support tickets from brand users - Route issues to relevant technical or business teams - Collect actionable product feedback for future versions

## 🧠 Core Features

### 1. Brand-Side Issue Dashboard

Submit ticket: select type (bug, feature, billing, help)

Attach screenshots, links, logs

Assign to department (tech, ops, success, billing)

Track status: Open → Under Review → Resolved

### 2. Superadmin Admin Panel

View all issues sorted by brand, type, urgency

Assign support agents or engineers

Add internal comments, resolution logs

Trigger email/SaaS dashboard alerts on updates

Export issues for reports / audits

### 3. AI Smart Triage

Auto-categorize based on ticket description

Prioritize based on keywords or affected module

Recommend resolutions based on historical tickets

Suggest FAQ links to brand automatically

### 4. SLA & Escalation Controls

Define SLAs per subscription tier (Basic, Pro, Enterprise)

SLA timers visible in admin UI

Auto-escalate if overdue

### 5. Feedback Loop

Mark tickets as “converted to feature request”

Feature request board with voting & tagging

Notify brand when released or scheduled

## 🔔 Notifications & Updates

In-app + email notifications on status change

Follow-up reminders for long-pending issues

Digest summary per brand per week/month

## 📊 Analytics

Ticket volume by brand, type, priority

Avg. resolution time per category

Recurring issues across clients

Most requested features (converted from support)

## 🔐 Access & Roles

Brand Admins: View + create issues, vote on features

SaaS Support Staff: Manage issues, add notes, resolve

SaaS Superadmin: Full control, analytics, SLA, export

## 📁 Tables:

brand\_support\_issues

brand\_support\_comments

brand\_support\_assignments

brand\_support\_resolutions

brand\_feature\_requests

brand\_support\_votes

## 🔌 Optional Integrations

Slack or Email ticket piping

HubSpot / Intercom CRM integration

Auto-sync issue types with roadmap tool (e.g., Trello, Notion)

✅ Now documented. Ready to link from unified SaaS admin panel.

# Modules/Loyalty

## Points Loyalty Tier EnginePoints Loyalty Tier Engine

🪙 Points + Loyalty Tier Engine

## ✅ Module Overview

The Points + Loyalty Tier Engine is a gamified rewards system designed to boost user engagement, retention, and community involvement for brands running on the SaaS platform. It works across UGC campaigns, purchases, referrals, and social actions (like commenting, sharing) and supports both dynamic point systems and tiered memberships (Bronze, Silver, Gold, etc.).

This module is fully independent and optional, with per-brand customization, and full control by the superadmin for point multipliers, feature access, and tier rules.

## 🎯 Goals

Reward users for engaging across FB, IG, website, and campaigns

Allow brands to define tier thresholds and reward structures

Build long-term user retention through milestone-based gamification

Enable point redemption for discount codes or exclusive rewards

Allow brand control over reward topics, points, tiers, and store linkage

## 🧩 Core Features

### 1. Point Rules Engine

Points can be earned from:

FB/IG post comment participation

UGC campaign entry

Purchase on WooCommerce/Shopify store

Referrals (tracked via links/codes)

Reacting to content (like, share, save)

Rules:

Dynamic points by action type and frequency (e.g., “5 pts per UGC entry”, “2 pts for comment”)

Time-limited bonuses (e.g., “2x points this week”)

Brand-defined blacklist or whitelist actions

### 2. Tiered Loyalty System

Default Tiers:

Bronze (0–299 pts), Silver (300–999 pts), Gold (1000+ pts)

Custom tier names and thresholds per brand

Each tier can unlock:

Exclusive rewards or bundles

Early access to campaigns

Higher reward limits or point multipliers

Shiny badges shown on profiles or UGC entries

### 3. Rewards Redemption Shop

Admins can define:

Reward items (discounts, vouchers, exclusive bundles)

Point costs per item

Limits (e.g., 1 redemption per user per month)

End-users:

See available rewards

Click to redeem using their current point balance

See redemption history in their profile

### 4. Progress Tracker & Profile Integration

For logged-in website users:

Show current tier and points

Show campaign actions completed

List of active campaigns they can join

Gamified UI with progress bar or badges

### 5. Admin Panel for Brands

Control point values for each action

Define and edit tiers + reward unlocks

View engagement stats (points distributed, redemptions)

Filter top users, campaign-wise performance

Set auto-expiry or monthly point resets (optional)

## 🛠️ Backend Requirements

### Tables:

user\_points: user\_id, brand\_id, points, last\_updated

user\_tiers: user\_id, brand\_id, tier, achieved\_at

points\_log: action\_type, user\_id, brand\_id, points, source, campaign\_id, timestamp

rewards\_catalog: brand\_id, reward\_id, title, points\_required, qty\_available, valid\_till

redemption\_log: user\_id, reward\_id, brand\_id, timestamp

### Services:

Points Tracker Service

Tier Evaluator Service (runs when points update)

Reward Redemption API

Campaign Action Logger

## 🤖 AI-Powered Enhancements (Optional)

AI suggestions for point rule improvements

Auto-upgrade tier names and reward ideas using seasonal trends

UGC template matching: bonus points for “high-performing” entries

## 🔐 Roles & Controls

Superadmin:

Enable/disable module per brand

Control quota/limits on redemptions

Pre-define common reward templates

Brand Admin:

Full control of points, tiers, and rewards

Access to redemption logs and user tiers

## 📊 KPIs to Track

Total users in each tier

Redemption conversion rate

Points earned vs used

Campaign-specific points boost

Top-performing UGC contributors

✅ Ready for export and implementation.

## Reward Redemption LoyaltyReward Redemption Loyalty

# 🎁 Reward Redemption Shop + 🤝 Loyalty Profiles

## ✅ Module Overview

This module enables brands to gamify user engagement by offering point-based rewards through a virtual Reward Shop and trackable Loyalty Profiles for each customer. It connects with the Points + Tier Engine, allowing users to redeem rewards based on their accumulated points, loyalty status, or campaign participation.

## 💰 Key Features

### 1. Reward Redemption Shop (Frontend Widget + Backend Panel)

Frontend Interface:

Shop-like UI with product cards (images, title, points required)

Filters: Category, Point Range, Loyalty Tier Required

Dynamic point balance shown in header

“Redeem” button triggers confirmation + delivery option (email/coupon/auto-order)

Responsive and embeddable on client websites or campaign pages

Backend Admin Panel:

Add/edit/delete reward items

Fields: Title, Description, Image, Points Required, Tier Required, Reward Type (Coupon, Product, Access Pass, etc.)

Inventory controls + reward limits (daily/weekly/user limit)

Auto-expiry and activation toggle

Redemption Types:

Auto-generated WooCommerce/Shopify coupon code

Manual fulfillment request sent to brand admin

Auto-apply on next order (if eCommerce linked)

### 2. Loyalty Profile (Customer View + Admin Insights)

Customer Profile Panel:

Current Point Balance

Loyalty Tier Status (Bronze, Silver, Gold, etc.)

Rewards Redeemed History

Campaigns Participated (with engagement types: comment, tag, share)

Referral Activity (if Referral module enabled)

Level Progress Bar: e.g. “1200/1500 points to Gold”

Admin Dashboard:

Leaderboards by point total, engagement score, tier

Redemption analytics: top claimed rewards, most engaged users

Filterable view by campaign, time period, user segment

CSV export for CRM usage

## 🔧 Technical Structure

### Database Tables:

rewards\_catalog

id, title, description, image\_url, points\_required, tier\_required, reward\_type, stock, daily\_limit, expiry\_date, active

reward\_redemptions

redemption\_id, user\_id, reward\_id, date, status, delivery\_type, fulfilled\_by

loyalty\_profiles

user\_id, total\_points, current\_tier, last\_redeemed\_at, campaigns\_participated

### Backend Services:

Rewards Engine

Coupon Generation API

Tier Evaluator

CRM Syncer

Reward Limit Validator

### APIs:

/api/rewards/catalog

/api/rewards/redeem

/api/loyalty/profile

/api/rewards/admin/\*

## 🚀 Use Cases

Gamified UGC campaign participation

Limited-time point redemption sales (“Flash Reward Friday”)

Exclusive rewards for top-tier members

Early access unlock via reward redemption

## 📆 Future Extensions

Gift reward to friend (social reward gifting)

Group campaigns with collective point goals

Dynamic rewards based on trending campaigns

🔹 Next Step: Connect this module with the Campaign Engine and Points Engine modules for unified loyalty experiences.

Let me know if you want a UI flow diagram or export this module to file!

# Modules/Gamification

## Gamified Feedback ModuleGamified Feedback Module

# 🧠 Gamified Feedback Collection Module

## ✅ Module Purpose

The Gamified Feedback Collection module is designed to turn product, service, or campaign feedback into a fun, engaging, and rewarding experience for users. Brands can incentivize users to share suggestions, complaints, or insights via Facebook/Instagram comments, inbox, or web forms—and then convert those insights into actionable ideas or even new UGC campaigns.

## 🎯 Goals

Collect valuable feedback from users in a non-boring way

Turn every feedback into a chance for reward or recognition

Generate data that can guide product improvements or blog campaigns

Enable participation even from users not interested in standard UGC formats

## 🧩 Features

### 1. Campaign Types

Suggest a Product: “What product should we launch next?”

Vote Your Favorite: Choose between two or more upcoming ideas

Rate & Win: Give feedback on your recent order or experience

Open Mic: General feedback collection with gamified prompts

### 2. Entry Methods

Facebook/Instagram Comments

Facebook/Instagram DMs (via Inbox GPT)

Website feedback form

AI-generated Instagram stories with poll stickers (future integration)

### 3. Gamified Mechanics

Points for each submission (configurable per brand)

Bonus points for early entries, long responses, or voted-up ideas

AI classifies suggestions by theme and tags them automatically

Voting system to highlight top community-backed ideas

### 4. Reward System Integration

Points directly added to user profile (if identity is mapped)

Eligible for leaderboard inclusion

Auto-DM to user: “Thanks for your suggestion! You’ve earned 20 pts!”

Winning entries get highlighted in stories/posts and rewarded

### 5. Admin Panel Tools

View all responses in a table with classification tags (AI-enhanced)

Mark winning feedback manually or by votes

Export CSV or turn into Blog Draft

Trigger reward assignment from panel

## 🔗 Integration Points

Connects to: Points + Loyalty Module, Blog Generator, Inbox GPT, UGC Analytics

Webhook triggers to notify when a new high-quality feedback is collected

Unified experience via Auto Poster module (posts, stories)

## 📊 Analytics

Entries per campaign

Reward cost vs feedback volume

AI tag cloud of common ideas

Voting trends over time

## ⚙️ Technical Requirements

feedback\_campaigns table

feedback\_entries table

user\_feedback\_points (linked to loyalty engine)

AI classification model for feedback tagging (can be OpenAI or local model)

Voting service (Redis or Firebase-based real-time voting counter)

## 🧠 AI-Powered Enhancements

Auto-cluster feedback themes: “30% of users suggested smaller packaging”

Suggest new product tags, campaign ideas, or bundles based on feedback

Flag low-quality/spam feedback automatically

## 🔐 Permissions

Brand Admins: Create campaigns, review entries, reward users

Superadmin: Moderate all feedback, manage abuse filters, train AI tags

## 💡 Example Use Case

Brand posts: “What product should we launch next? Comment below and win points!”

Users reply: “Travel-size toner!”, “Glow serum in a pump!”

System collects comments, tags them with product categories

Admin sees clustered suggestions and picks top one

Winner gets 100 points and is featured in story

Feedback becomes a new blog post: “Why You Asked for Travel Toners & We Listened”

✅ Module ready for export, UI generation, or system linking.

# Modules/Content

## Ai Blog Meta OptimizerAi Blog Meta Optimizer

# 📚 Module 8: AI Blog Refresher + Meta Title Optimizer

## 🎯 Overview

This module automates blog performance optimization by analyzing traffic, freshness, and SEO gaps. It detects outdated or underperforming blogs, refreshes them using GPT, and suggests or updates meta titles/descriptions for improved rankings.

It also integrates with Google Trends to keep content aligned with rising searches and auto-suggests SEO-optimized meta tags for all product/blog pages.

## ✨ Core Features

### 1. Blog Analyzer (Stale Content Detector)

Connects with GA4 API to fetch:

Bounce Rate

Avg. Time on Page

CTR from source (e.g., Google search)

Blogs are scored based on performance thresholds.

Low-performing ones are flagged as stale and queued for refresh.

### 2. Blog Refresher Engine

Uses blog’s existing content + GPT prompt to rewrite:

Headline

Intro paragraph

CTA section

FAQs (optional)

Ensures:

Tone remains brand-specific

Content remains factually aligned

Internal product links updated

Prompt Example:

Rewrite the following blog intro to be more engaging for Bangladeshi readers. Retain key points. Match tone = expert yet friendly.  
[existing blog intro...]

### 3. Google Trends Matcher

Pulls regional trend data via pytrends

Matches topics with blog/product categories via vector similarity

Suggests trending keywords like:

“niacinamide during monsoon”

“safe actives for teens”

Adds them into:

Blog titles

Meta descriptions

Headings (if enabled)

### 4. Meta Title & Description Optimizer

For all blogs and product pages

GPT suggests optimized meta fields based on:

Google Trends

Page topic

Keyword gaps (via Google Search Console data)

Optional tone/style toggles:

SEO focused

Conversational

Bangla + English mix

Prompt Example:

Generate a 60-character meta title for a blog about "retinol night routine". Add power words, emotion. Mention Bangladesh.

### 5. AI Meta Field Generator for Products

Runs nightly job to:

Detect missing/empty meta fields

Regenerate meta fields for low CTR products

Based on:

Product title, concern, benefit

Customer UGC (if enabled)

Trends

## 🛠️ How to Build

### Stack:

Language: Python + Node.js

Vector Search: FAISS

Embeddings: OpenAI

Data: GA4, Google Search Console, pytrends

Queue: Redis

### DB Tables:

blogs: { id, url, title, ctr, time\_on\_page, score, stale }

blog\_versions: historical versions pre/post refresh

meta\_fields: { type: ‘product’ | ‘blog’, id, meta\_title, meta\_desc, gpt\_generated\_at }

trending\_keywords: [ keyword, volume, matched\_topic ]

### API Endpoints:

GET /blogs/stale → list blogs to refresh

POST /blogs/:id/refresh → triggers rewrite

GET /meta-suggest?type=blog&id=... → returns meta suggestions

POST /apply-meta/:id → applies meta fields

### Scheduled Tasks:

Run pytrends every 6 hrs

Nightly job to scan blogs + products needing refresh

## 🧠 GPT Prompt Templates (for Codex)

“Rewrite this blog CTA to sound like a personal beauty expert in Bangladesh”

“Generate 3 SEO meta descriptions for a skincare blog on oily skin in monsoon”

“Suggest 5 trending Bangla-English blog titles for a blog about vitamin C serums”

## 🔗 Integrations

Blog Engine

Product Catalog / PDP

Google Search Console + GA4 APIs

AI Product Brain (for internal linking and product tag enrichment)

## 📈 Outcomes

Higher blog engagement

Better search rankings

Auto-updated SEO content

Discoverability aligned with local trends

✅ This module runs fully automated in “Auto” mode but supports Manual and Assisted workflows from brand panel.

Superadmin can:

Enable/disable blog refresh & meta optimizations per brand

Control trend sync frequency

Review GPT logs for each change

## Ai Blog RefresherAi Blog Refresher

AI Blog Refresher System: Full Technical Documentation

### Overview

The AI Blog Refresher System is an intelligent module that automatically revisits previously published blog posts, detects outdated content or low-performing articles, and refreshes them with newer insights, data, SEO-optimized copy, and internal product links. This module is a critical part of our long-term SEO automation and content lifecycle engine.

### Key Goals

Maintain SEO freshness of blog content.

Automatically insert trending keywords from Google Trends.

Revise meta titles/descriptions based on performance data.

Auto-link newer products or collections.

Suggest richer visuals and embedded videos.

### Functional Components

#### 1. Blog Crawler & Extractor

Parses and stores full content of existing blog posts.

Detects publish/update timestamps, H1-H6 tags, paragraph structure.

Sends extracted content to the refresh pipeline.

#### 2. Content Age & Performance Evaluator

Flags blogs older than X days (configurable).

Scores blog based on:

Organic traffic (from GA4 integration)

Keyword rankings (via API with SEMrush / SerpApi)

Click-through rate on meta titles

Engagement metrics (e.g., time on page)

#### 3. Refresh Trigger Engine

Scheduled to run daily/weekly.

Supports manual trigger from admin UI.

Can be filtered by brand, category, or campaign.

#### 4. AI Content Updater (Codex/ChatGPT)

Uses embedded blog data + trend signals + previous performance.

Rewrites paragraphs with:

Updated statistics.

Latest keywords (auto-inserted with Yoast-style density).

Enhanced formatting (bullet lists, bold phrases).

Keeps tone and style consistent with brand voice settings.

#### 5. Internal Link Recommender

Parses new product feeds or bestsellers.

Suggests 3–5 anchor text links inside the blog body to related products.

Can auto-insert using natural phrases.

#### 6. Meta Title & Description Rewriter

Optimizes for CTR by analyzing previous meta performance.

Adds power words, urgency, freshness.

Keeps within length guidelines.

#### 7. Media Refresh Assistant

Suggests adding newer featured images (from Canva Generator or Upload).

Recommends embedding latest YouTube or Reels content.

Ensures alt-text and schema markup are SEO-compliant.

#### 8. Version History & Approval Panel

Tracks previous vs refreshed version.

Admin can approve, reject, or further edit content.

Push live with 1-click.

### Technical Stack & Flow

Blog Fetch Layer: WordPress REST API / Shopify Storefront API to fetch existing content.

Content Parser: Node.js microservice using html-to-text, jsdom, etc.

Trigger Layer: Cron job + webhook listener for manual triggers.

AI Engine: OpenAI GPT-4 via fine-tuned model for rewrites, powered by blog embeddings.

Google Trends Integration: Real-time keyword injections.

Meta Optimizer: Shared with Smart SEO module.

Storage: MongoDB or PostgreSQL to track refresh logs and content diffs.

Admin Panel: Integrated into SaaS UI (React + Tailwind).

### Use Cases

Reviving low-traffic blogs with new data.

Keeping seasonal guides (e.g., winter skincare) relevant each year.

Aligning blogs with new product launches.

Creating an automated content refresh cycle.

### Prompts for Codex / GPT

"You are an expert SEO content editor. Refresh this blog content using the latest stats, power words, better formatting, and at least 3 internal links to products. Maintain the original tone."

"Analyze this blog post's meta title and description. Improve its click-through rate by rewriting them using SEO best practices and emotional triggers."

### Admin Controls

Enable/disable auto-refresh per brand.

Set minimum traffic threshold to skip.

View refresh logs, content changes, and engagement impact.

Manual override: select blogs and force refresh.

### Future Enhancements

Auto A/B test older vs refreshed version for CTR.

Add support for multilingual blogs.

Suggest refresh frequency by category.

Tie refreshed content to new email campaigns.

## Ai Faq BuilderAi Faq Builder

AI FAQ Builder Module

### ✅ Overview

The AI FAQ Builder module is a smart engine that analyzes:

Blog content

UGC (user-generated comments, campaign replies)

Past inbox conversations (FB/IG/WhatsApp)

And automatically generates:

High-quality, natural-sounding FAQ questions and answers

FAQ sections for product pages, campaign pages, or help centers

This dramatically reduces content ops workload and increases user trust and SEO performance by keeping helpful FAQs updated and personalized.

### 🧠 Core Features

FAQ Extraction Engine

Uses GPT with fine-tuned prompts to extract potential questions from blog, product descriptions, and user interactions.

Classifies content into:

Product FAQs (specific to items)

Campaign FAQs (e.g. gamification, offers)

Brand FAQs (delivery, refund, authenticity, etc)

Multi-source Input Mapping

Sources include:

Blogs (via internal content index)

Customer inboxes (stored via Inbox Intelligence Engine)

FB/IG comments (via UGC trackers)

Product reviews (integrated via WooCommerce/Shopify APIs)

Deduplication + Answer Optimizer

Avoids repeated questions

Chooses the most helpful answer snippet using:

Semantic similarity

User upvotes (if available)

Conversion data (future enhancement)

Auto Format for Output

Markdown or HTML FAQ lists

Structured data with schema.org/FAQPage format (for SEO)

Panel Interface for Approval & Publishing

FAQ drafts shown per product/post/campaign

Toggle between:

Auto-publish mode

Human review mode

Approve, edit, reorder questions

Smart FAQ Widgets

Embed on:

Product page

Blog post

Campaign landing page

Expand/collapse UX

Optional upvote/downvote or helpful toggle

Multilingual Support

Auto-translate and generate localized FAQs (Bangla, Hindi, Urdu, English)

Syncs with Localization Module

Scheduled FAQ Updates

Auto-refresh FAQs based on new incoming user data or new blog versions

Settings to control frequency (daily, weekly, monthly)

### ⚙️ How It Integrates

With Inbox Intelligence Engine: Extract past questions or repeated inquiries as seed FAQ topics.

With Blog Engine: Every time a blog is published or updated, the system parses it to generate contextual FAQs.

With WooCommerce/Shopify: Fetches product data and links FAQs to individual products via metadata.

With Admin Panel: Shows draft FAQs per item for approval/editing.

With Website Frontend: FAQs embedded via shortcode or injected by JS SDK.

### 🔐 Roles & Controls

Superadmin:

Can define global templates/prompts

Enable/disable auto-posting or review flows

Brand Admin:

Approves/edit FAQs per product/campaign

Tracks FAQ helpfulness

Editors:

Suggest new FAQs

Review flagged content

### 🛠️ How to Build

FAQ Extraction Pipeline

Input: Blog/product/campaign text

GPT Prompt:

Extract 5-10 FAQ style questions with accurate answers from the following content. Avoid repeating the same answer. Format as:  
- Q: ...  
- A: ...

Backend Service

Python or Node.js microservice with:

Content ingestion API

OpenAI prompt manager

Output storage in faq\_entries table with source\_type, source\_id, question, answer, language, approved\_by, published\_at

Admin Panel UI

Table view of:

Pending FAQs (auto-generated)

Approved/Published FAQs

Controls: Approve / Reject / Edit / Translate

Frontend Widget

JS SDK to display FAQs on any client site

Filter by product\_id or campaign\_id

Optional schema.org FAQ markup for SEO

Scheduler + Trigger Engine

New FAQ generation triggered by:

Blog publish

New campaign post

Weekly inbox/UGC scan

Multi-language Support

After original FAQ is generated, auto-translate using Google Translate API or OpenAI GPT-4 multilingual prompt

### 🧩 Optional Add-ons

User voting system (helpful / not helpful)

FAQ heatmap or click tracking

“Did this answer your question?” pop-up follow-ups

### 📊 Metrics to Track

# of FAQs generated per product/post

# of FAQs approved vs discarded

Click-through or engagement rate on FAQs

Avg upvote/downvote ratio

### 🎯 Benefits

Reduces support ticket volume

Boosts SEO via structured content

Improves customer trust & transparency

Helps with voice search readiness

✅ Document complete. Let me know if you’d like to export it or add visuals/mockups next.

## Reels Shorts Auto SuggesterReels Shorts Auto Suggester

# 🎥 Module: Reels / Shorts Auto-Suggester

## 🎯 Purpose

Enable brands to generate engaging short-form video content (Instagram Reels, YouTube Shorts, TikTok) automatically from their: - Product catalog - Blog posts - UGC campaigns - Trending beauty themes

The goal is to boost organic reach, engagement, and brand recall without the need for manual video editing.

## 📦 Core Features

### 1. Smart Topic Detection

Sources:

Recent blogs

Popular products

Viral UGC comments

Google Trends + seasonal beauty search terms

Classifies into:

Tutorial ideas

Product hacks

Before/after transformations

Ingredient spotlights

### 2. Storyboard Generator (AI)

Uses blog/product content to generate a 5–10 step storyboard:

Hook line (text + voiceover)

Visual suggestion (image or video type)

On-screen text / subtitle

CTA suggestion

All framed under: 15–60 seconds formats

### 3. Canva-Based Auto Visuals (Default Mode ✅)

Automatically selects template for each step of the storyboard

Pulls product image, blog banner, UGC photo, or stock placeholder

Generates animated text and transitions

Adds background music (from a royalty-free pool)

Exports final MP4 for manual or auto-publishing

### 4. Manual Override + Editing UI (Optional for Premium Brands)

Drag and drop storyboard blocks

Change visuals

Change on-screen text or caption

Upload voiceover or narration

### 5. Auto Caption Generator

From blog summary or GPT

Uses Reels-optimized, scroll-stopping copy

Includes CTA, hashtags, and emojis

### 6. Auto-Post + Scheduler (Connected with Poster Module)

Schedule reel uploads to IG/FB/TikTok (via API or reminder)

Modes:

Manual upload

Assisted (download + email alert)

Auto (for FB/IG via API + preset time slot)

## 🧠 AI Capabilities

GPT-4 Turbo: for summarizing, rewording, scripting

Vision AI: (optional) for scanning UGC/photos and suggesting angles

Trends Analyzer: maps with real-time queries via Google Trends API

## 🛠️ How to Build It

### Stack

Backend: Node.js, Python, FFmpeg, Canva API, OpenAI

Frontend: Next.js + Tailwind with rich video preview UI

Storage: AWS S3 or GDrive export

### Tools/APIs

Canva API: for video/image rendering

GPT-4: storyboard + script generation

Google Trends API: topic ranking

FFmpeg: video composition + audio merging

### DB Tables

reel\_templates

reel\_requests

reel\_outputs

scheduled\_reels

### API Endpoints

POST /reel/suggest → Suggest new reels from blogs/products

POST /reel/build → Generate visuals based on storyboard

GET /reel/history → List generated shorts per brand

POST /reel/schedule → Connect to auto-posting queue

## 🔒 Admin + Superadmin Controls

Set monthly reel generation quota

Enable/disable video editing interface

Logs of auto vs manual edits

Review usage + analytics dashboard

## 🔗 Connected Modules

Blog Engine → generates input for reels

Product Brain → pulls metadata, images

Auto Poster → schedules output

Gamification → rewards best-performing videos

## 📈 Benefits

Boost reach on IG/FB/TikTok

Repurpose blog/product content as videos

Save editing/design effort

Increase discoverability through reels + hashtags

✅ This module will be included in the base platform but image/video rendering engine (Canva API) will be treated as an add-on.

## Reels Shorts Generator ModuleReels Shorts Generator Module

Reels/Shorts Content Generator Module

## 🎥 Overview

The Reels/Shorts Generator is a modular content engine that helps beauty and skincare brands automatically generate engaging short video concepts (15s–60s) optimized for Instagram Reels, Facebook Reels, and YouTube Shorts. Designed for brands with minimal design resources, this module leverages AI to transform product features, blogs, or campaign UGC into shareable visual scripts and Canva-based animations.

## 🧠 Core Capabilities

### 1. Content Input Sources

✅ Blogs generated from platform

✅ Product highlights (e.g. best-sellers, launch items)

✅ UGC comments or gamification responses

✅ Prebuilt brand messages (campaigns or promos)

### 2. AI-Powered Script Generator

Converts input into:

Hook (first 3 seconds)

Visual Actions (e.g. “show applying on cheek”)

Text-on-screen (storyline overlay)

CTA (e.g. “Shop now on Lavishta.com”)

Prompt Example: > “Create a 30s Reels script from blog titled ‘How to fix pigmentation in monsoon’. Add text overlays, audio suggestion, and CTA to shop two recommended products.”

### 3. Shot Layout Builder

Breaks script into scenes with:

Frame layout (e.g. close-up, flatlay, product-in-hand)

On-screen text

Emoji/emotion cues

Time per scene

### 4. Canva Auto Visual Add-On

Auto-generates static visual reels or animated text using Canva APIs.

Example scenes:

Scene 1: “Rainy skin? 🌧️” overlay on stock image

Scene 2: Product 1 fade-in

Scene 3: Text: “Fade pigmentation fast!”

Optional Features: - Custom brand colors/logo overlay - Add audio track suggestions (licensed or recommended trend sound)

### 5. Modes of Operation

🧠 Auto Mode → Generate + Schedule without human input

✍️ Assisted Mode → Generate suggestions, brand approves or edits

🖐️ Manual Mode → Create storyboard manually, then AI enhances

## 📤 Publishing & Export Options

Export as Canva link (editable)

Download as MP4

Schedule directly via Post Manager module

Share to brand’s IG/FB directly if permissions are connected

## 🛠️ How to Build It

### Stack:

Backend: Node.js + Python

AI: OpenAI Codex or GPT-4, Canva API

Storage: MongoDB for drafts + logs

### Key APIs:

POST /generate-reel-script → Accepts blog/product/ugc reference

GET /reel-preview/:id → Shows storyboard + text + CTA

POST /reel-to-canva → Creates editable Canva template

POST /reel-to-schedule → Adds to Post Manager queue

### Database:

reels\_scripts: stores scene-by-scene layout

canva\_links: stores generated design link

scheduled\_reels: status, platform, engagement

## 🎯 Use Cases

Promote blogs with visual storytelling

Turn campaigns into short-format viral clips

Create product usage demo via auto templates

Turn UGC into authentic-feel Reels

## 🔐 Superadmin Control

Enable/disable module per brand

Limit credits per month (script generations / Canva exports)

Moderate access to Auto Mode

## 🔄 Integration Touchpoints

Post Manager → Directly schedule generated Reels

Gamification → Convert top campaign comments into clips

Email/SMS Module → Link reel in campaign footer

## ✅ Summary

This module enables brands to break into short-form video content effortlessly using AI and Canva. With assisted or full-auto modes, it suits both lean teams and advanced marketers. It multiplies blog/campaign/product visibility while aligning with social trends, improving engagement and conversion.

## Smart Seo OptimizerSmart Seo Optimizer

Module 7: Smart SEO Optimizer

## 🔍 What This Module Does:

An intelligent, automated SEO refinement system that audits and enhances meta titles, descriptions, and discoverability of blog posts, products, and category pages. It uses real-time search trend data, CTR signals, and AI-generated rewrite logic to optimize for click-through rates and better rankings.

## ✨ Core Features

### 1. 🌐 Google Trends + Local Query Matching

Integrates Google Trends hourly via pytrends with localized (BD/SAARC) filters.

Maps trending keywords to product vectors or blog topics.

Example: “niacinamide serum price BD” ➔ match to best-selling niacinamide product page.

How to Build: - API: GET /trend-matches - Process: - Fetch rising queries for category terms (e.g., “serum”, “pigmentation”, “glow skin”). - Use FAISS to vector-match to relevant product/blogs. - Score by trend match + click-through rate + availability.

### 2. 📈 Meta Title + Description Refresher

Audits every product/blog/category SEO metadata weekly.

Flags underperforming CTR (via GA4 API)

Uses OpenAI to generate improved variants.

Codex Prompt Format: > Rewrite this product’s meta title and description to improve SEO. Focus on benefits, include brand and target keyword “[TERM]”. Target audience is women in Bangladesh looking for [CONCERN].

How to Build: - Cron job: weekly scan - Query CTR + bounce data from GA4 - GPT batch generator + store diff - Manual override or auto-publish based on brand settings

### 3. ⚡ Auto-Refresh Engine

Low-performance pages get refreshed in background mode.

Uses content embeddings to ensure context relevance isn’t lost.

Supports multi-lingual outputs (BN, HI, UR, EN).

API: - POST /refresh-metadata - Input: page\_id, reason, lang, current\_metadata - Output: { improved\_title, improved\_meta, confidence\_score }

### 4. 🌎 SEO Score Analyzer

Gives per-page health score: CTR trend, bounce, keyword overlap, freshness, duplicate detection

Provides smart suggestions per field

Frontend UI: - Page table with: - ✔ SEO Score - ↓ CTR drop - 📅 Last updated - ✔ “Improve” button (triggers GPT patch)

### 5. ✨ Multilingual Meta Support

Automatically generates alternate language meta info for blogs + products

Output fields: meta\_bn, meta\_hi, meta\_ur

### 6. 🎡 Brand Tone Calibration

Each brand can define tone: Informative / Expert / Casual / Aspirational

Affects GPT rewrite style

## 📅 Technical Stack

Scheduler: Redis Queue + Cron Jobs

NLP: OpenAI Codex, FAISS

CTR/Analytics: GA4 API

Frontend: React + Table Grid + Diff Preview UI

## 📌 Usage Settings

Brand-Level Control Panel - Enable/disable auto SEO - Set review before publish (Manual, Auto, Assisted) - Upload brand tone samples - Select active languages

Superadmin Controls - GPT usage quota - Max pages to optimize/month - Logs of regenerated metadata

## 🔄 Integrations

Blog Engine: Pass stale blogs for metadata update

Auto Poster: Use refreshed titles in post captions

Smart Widget: Improve product discovery terms

## ✅ Useful Outcomes

Higher click-through rate on search

Better local keyword targeting

SEO-friendly content refresh without manual editing

This module, once active, runs 90% automatically and generates massive SEO value with minimal brand input. Suitable for both aggressive and passive brands alike.

Next Module to Document: Shared OAuth + App Infrastructure

## Video Reel Script GeneratorVideo Reel Script Generator

# 🎥 Video / Reel Script Auto Generator (from Blogs, Products, and Campaigns)

## ✅ Module Overview

This module generates ready-to-use video or reel scripts from existing content sources like: - Blogs - Product descriptions and highlights - Campaign captions or prompts

The output is designed for short-form platforms like Instagram Reels, YouTube Shorts, Facebook Stories, and TikTok. The goal is to make video marketing effortless for brands, helping them turn every piece of content into highly engaging video scripts.

## 🌟 Goals

Increase content mileage through multi-format reuse

Enable brands to create high-conversion Reels without manual scriptwriting

Reduce dependency on creative teams

## 🧰 Feature Set

### 1. Input Sources

Blog to Reel: Extracts tone, CTA, and core story from blog posts

Product to Reel: Pulls ingredients, benefits, usage, and visual highlights

Campaign to Reel: Converts any caption/prompt into a punchy script

### 2. Script Generation Types

Story-Style: 3-act format with hook, transformation, and CTA

Listicle: “Top 3 Reasons to Try…”

Tutorial: Step-by-step usage explained visually

Before/After: Transformation narrative

Offer-Based: “Grab This Deal Before It Ends!”

### 3. Structure of Generated Script

Intro Hook (2-3 sec): Grab attention

Core Message (5-8 sec): Feature/product/benefit explained

Visual Instructions (optional): Suggestions for B-roll, demo footage

Text Overlay Prompts: On-screen text

Voiceover Script: Optional spoken part

End CTA (2-3 sec): Swipe, click, follow, shop, etc.

### 4. Tone & Style Customization

Brand voice selector: Formal / Fun / Expert / Trendy / Gen Z

Language: Supports multilingual output from Localization module

Platform: Adjust length and format for Instagram, TikTok, etc.

### 5. Script Preview Modes

Simple Text Preview

Storyboard-style View: Frame-by-frame description

Downloadable Script File (TXT / JSON)

### 6. Script-to-Design Integration

Push content to Auto Canva Image Generator for visuals

Send script to Reels/Shorts Auto Posting queue

Feed into AI Voiceover or animation tools (future phase)

## ⚙️ Technical Architecture

### Tables

reel\_script\_templates

reel\_script\_instances

content\_to\_reel\_map

brand\_video\_styles

reel\_usage\_stats

### Services

Prompt Engine (GPT-4 / Fine-tuned model)

Content Extractor Engine (for parsing blogs/products)

Storyboard Generator

Tone Adapter & Style Mapper

## 📊 Analytics

Track which scripts got used

Measure video views and engagement (via IG/TikTok APIs)

Popular script formats per product/campaign

## 🎨 UI Components (Brand Panel)

Script Generator Panel

Source selector (blog, product, campaign)

Script type dropdown

Brand tone & language selector

Visual idea suggestions (optional AI image prompt)

Preview & Download Panel

Show visual plan with text + voice overlay

Option to approve/edit/regen

## 🔒 Permissions

Brand Admin: Generate/edit/download

Content Editor: Request regen / mark for posting

Superadmin: Manage template types

## 📊 Benefits

Turns every blog or product into a viral marketing script

No editing or creative knowledge required

Compatible with future video tools (text-to-video, AI avatars, etc.)

🔄 Ready to connect this with posting engine or Canva-based visual flow.

# Modules/Marketing

## Campaign Templates ModuleCampaign Templates Module

# 🧩 Campaign Templates Marketplace + Persona-Based Suggestions

## ✅ Overview

The Campaign Templates Marketplace is a shared library of pre-approved, high-performing gamification campaign templates available to all brands within the platform. Combined with Persona-Based Suggestions, it allows each brand to:

Browse from proven campaign styles

Get automated recommendations tailored to their brand voice, tone, and audience

Instantly clone, edit, and deploy campaigns

This enables rapid growth, engagement, and UGC generation without needing to constantly invent from scratch.

## 🎯 Goals

Provide ready-to-use gamified campaign templates

Increase campaign diversity, creativity, and success rates

Help brands discover new campaign ideas by persona

Reduce friction in campaign launch cycles

Offer community-vetted formats with proven results

## 🧠 Feature Set

### 1. Template Library (Shared by Superadmin)

Curated list of templates, each with:

Campaign Type: Comment Prompt / Tag Prompt / Quiz / Photo UGC / Video Entry / Poll

Example Caption

Suggested Visual Style (Canva template)

Reward Suggestions

Metrics from other brands (likes/comments/CTR if shared)

Metadata fields:

template\_id, title, campaign\_type, language, persona\_fit, conversion\_score, created\_by, times\_used

### 2. Persona-Based Campaign Recommender

Based on the brand’s persona (from AI Setup Wizard), recommend:

Top 3 matching templates per campaign type

Past successful templates for similar audiences

Recommender logic uses:

Audience match (Gen Z, Moms, Teens, Men)

Tone style (Formal, Fun, Friendly)

Language preference (Bangla, English, etc)

Engagement signals from historical usage

### 3. Template Preview + Clone/Edit Flow

Preview full sample post with dummy visuals, hashtags, rules

Click “Use Template” to:

Auto-fill Campaign Builder form

Optionally modify: Title, Caption, CTA, Reward logic

Customize schedule and publishing plan

### 4. Template Contribution (Optional)

Brands can submit successful campaigns back to the marketplace

Submitted templates reviewed by Superadmin before publishing

Earn badges (e.g., Top Contributor, Most Cloned)

### 5. Smart Filters + Tags

Filter by:

Persona type

Engagement type (photo, comment, quiz)

Visual complexity (Simple vs Designed)

Reward level (No reward / Coupon / Bundle)

### 6. Template Metrics Dashboard

Track:

Templates with highest engagement

Most-used by other brands

Avg campaign lifespan

Contribution leaderboard

## 💡 Example Templates

## ⚙️ Technical Design

### Backend Tables

campaign\_templates

id, title, type, persona\_fit, language, template\_body, preview\_url, metrics\_json, created\_by, shared, times\_used, reviewed, tags[]

template\_usage

brand\_id, template\_id, date\_used, performance\_data

persona\_index

brand\_id, audience\_type, tone\_style, language

### Services

Template Recommendation Engine (uses persona index)

Template Builder Renderer

Admin Approval API

Usage Tracker + Metrics Collector

### Frontend Modules

Library UI: Cards with Preview + Clone

Filters/Tags Component

Suggestion Box (“Best for your audience”)

Edit-in-Builder flow

Usage stats page for each template

## 🔐 Roles

Superadmin:

Create/edit global templates

Review & approve community submissions

Moderate inappropriate entries

Brand Admin:

Clone/edit templates

Submit new campaign formats

## 📊 Metrics to Track

# of times each template used

Avg engagement per campaign type

Brand satisfaction rating post-usage

Template performance index (likes/comments/CTR per 1k impressions)

Reuse rate (how often same brand repeats a template type)

## 🎯 Benefits

Dramatically speeds up campaign creation

Builds consistency in performance

Encourages creative experimentation

Enables knowledge sharing across brands

Supports multilingual, persona-driven engagement

✅ Document ready. Let me know if you’d like to export it or begin the next module (e.g., Developer Toolkit, Smart Search Widget, or Fallback Engine).

## Email Sms Campaign BuilderEmail Sms Campaign Builder

Email & SMS Campaign Builder Module Documentation

## 📧 Overview

A standalone yet integrable module for automated, AI-powered Email & SMS campaign creation, delivery, and analytics tailored for ecommerce brands in the beauty/skincare space. Supports:

Prebuilt campaign templates (restock alerts, sale teaser, UGC winner announcements)

Automated generation of subject lines, email/SMS copy, and CTA links using OpenAI

Customer mapping via inbox/comments/order history uploads

Rate limiting and anti-overflow logic

Manual, Assisted (AI), and Fully Auto modes

## ⚖️ Use Cases

Automatically notify customers about:

Product restocks

Campaign outcomes

Seasonal offers

Recommended product bundles

Reach customers who commented on FB/IG with mapped identity

Tie Email/SMS to reward engine, abandoned carts, wishlist triggers

## 💡 Feature Breakdown

### 1. 🔹 Campaign Types (Prebuilt + Custom)

Prebuilt Templates:

Restock Alert

UGC Campaign Winner

Flash Sale Notification

Blog Highlight Teaser

Personalized Bundle Recommendation

Custom Campaigns:

Brands can start from scratch

Compose their own triggers + segmentation

How to Build:

Template Library stored in MongoDB

Each template includes:

{  
 "template\_id": "flash\_sale",  
 "title": "Big Savings on Your Faves!",  
 "subject\_prompt": "Create a catchy subject line for a 3-day skincare sale",  
 "body\_prompt": "Announce the sale, add 3 product suggestions, CTA to shop now",  
 "cta\_link\_type": "collection",  
 "utm\_params": {...}  
}

### 2. 🔹 AI Campaign Generator

Auto-generate:

Subject line / SMS intro

Email body with personalized tone, emojis

CTA links: to product, collection, blog

Segment suggestions (e.g. “frequent buyers”, “abandoned carts”)

Prompt Example:

Write a short email to announce a 3-day flash sale for oily skin products. Mention hot weather, discounts, and include emojis.

Endpoints:

POST /generate-campaign

Payload: template\_id, brand\_tone, sale\_items, campaign\_goal

POST /send-preview

Sends preview to brand email/number

### 3. 🔹 Customer Mapping Engine

Use customer uploads + inbox sync to enrich email/SMS list

Match FB/IG comments to users using:

Inbox replies

WhatsApp entries

Website order phone/email

GDPR compliant, user-controlled export/delete

Tables:

customers, mapped\_channels, campaign\_logs

How to Build:

Matching logic: fuzzy match on name + phone

Consent tracking: gdpr\_accepted field per customer

### 4. 🔹 Smart Scheduler + Frequency Cap

Set delivery intervals: hourly, daily, weekly

Suppression logic:

No more than 2 campaigns per user per 24h

Auto-hold if campaign type sent recently

Config Panel:

Campaign mode: Auto / Assisted / Manual

Max messages/day

Enable per platform: Email / SMS / Both

### 5. 🔹 Analytics & Reporting

Opens, Clicks, Conversions

UTM parameter builder per brand

Track CTR per CTA link

Opt-out logs and bounce tracking

Dashboards:

Top campaign performers

User journey impact (email > site > purchase)

SMS vs Email comparison

## 🛍️ Admin / Brand Controls

✅ Toggle Email/SMS module per brand

✅ Upload email/SMS credits per month

✅ GDPR: Export/Delete requests + audit

✅ Brand-specific sender name, SMS mask

## 🚀 Tech Stack

Backend: Node.js + MongoDB + Redis + Twilio + SendGrid

AI: OpenAI GPT-4 Turbo / fine-tuned LLM (optional)

Frontend: Next.js

Email Templates: MJML-based auto design engine

## 🌐 Future Enhancements

AI subject line A/B testing

Trigger-based sends (e.g. restock webhook, cart abandoned)

Personalized product carousels in email

Festival-aware copywriting (localized)

Shopify+Woo triggers to sync events

## 💪 Integration Hooks

POST /sync-orders

POST /sync-inbox

GET /user-preferences

POST /trigger-campaign

## ✅ Summary

This module creates automated but brand-personalized communication flows that drive traffic, reactivate users, and increase UGC loop. Designed to respect user preferences and superadmin-level usage caps.

All content is AI-assisted and respects tone + delivery limits. Fully independent yet natively connected to other modules like UGC, Post Manager, and Customer Brain.

Standalone? ✅  
GDPR Support? ✅  
AI Copy Gen? ✅  
Cross-Platform Reach? ✅

## Email Sms Campaigns ModuleEmail Sms Campaigns Module

Email & SMS Campaign Builder Module Documentation

## 📬 Overview

A customizable and automated system for brands to run personalized campaigns via email and SMS. Designed as an optional add-on, this module can:

Automate messaging for events like restocks, UGC rewards, new blogs, or offer announcements

Generate campaign copy using GPT with tone/style matching

Track delivery, opens, clicks, and conversions

Segment customers based on behavior, tags, past engagement, or mapped identity (from Customer Upload Module)

## 🎯 Key Features

### 1. Prebuilt Campaign Templates

Restock Alert: Notifies when specific product is back

UGC Winner Notification: Informs winners with links to claim rewards

Offer Teasers: Early access or FOMO-driven campaigns

New Blog Alerts: Drives traffic from engaged customers

Bundle Suggestions: Based on recent browsing or purchases

Admin UI:

Pick template → preview → modify content → choose channel (email/SMS) → schedule/send

### 2. Campaign Generator (AI-Powered)

Auto-fills message content based on:

Product name, promo, or blog

Tone/voice set by brand

Language preference (EN/BN/HI/UR)

Channel type (SMS = concise, Email = visual, long-form)

Prompt Sample:

"Write a friendly email to notify customers that our Niacinamide 10% Serum is back in stock, ideal for oily skin in humid weather. Mention Lavishta, keep it concise with a button to buy now."

### 3. Audience Segmentation

Filters:

All customers

Tagged users (e.g., acne, dry skin, K-beauty lover)

UGC participants

Users who commented/tagged/replied on a specific post

Based on last purchase, last engagement, campaign joins

How It Works:

Customer upload → mapped to phone/email → tagged on behavior → targeted for campaigns

### 4. Automation Engine

Trigger-Based:

Product restocked

Campaign winner announced

New blog published

UGC point tier reached

Scheduled-Based:

Monthly new arrivals

Weekly roundups

Birthday messages (optional)

Workflow UI:

Visual drag-drop builder

IF [event] THEN [send campaign to segment X] → choose timing

### 5. Delivery & Performance Tracking

Email: open rate, click rate, bounce, unsubscribe

SMS: delivery, click-through (via short link)

UTM auto-tagging for blog/product links

Conversions: sync with order sheet or Woo/Shopify order DB

## 🛠️ Technical Stack

Backend: Node.js + Redis + PostgreSQL

Email API: Mailgun (or Sendgrid support)

SMS API: Twilio or BD-based provider (SSL Wireless / Wavy / Robi Axiata)

AI: OpenAI GPT 4o with campaign prompt templates

Front-end: React (Next.js) with drag-drop campaign builder

Rate Limits: Per brand, controlled by Superadmin

## 🧑‍💻 API Endpoints

POST /campaign/create

GET /campaign/segments

POST /campaign/schedule

POST /campaign/send-now

GET /campaign/logs

## 🔐 Superadmin Controls

Enable/disable per brand

Set max SMS/email sends per month

Approve custom templates or restrict to defaults

View per-campaign cost estimate

## 📈 Use Cases

Increase repeat orders via timely reminders

Turn blog readers into buyers

Reward engaged users to increase retention

Recover inactive users with personalized incentives

## 🧠 AI Prompts Library

Prewritten prompt types per template:

“Restock notification for X”

“Campaign win notice”

“Your skin needs help – read this blog”

“You earned a new UGC badge”

“Don’t miss out on this 24hr offer”

✅ All data usage is GDPR-compliant. Consent, unsubscribe, and opt-in required before storing or sending to contacts.

# Modules/AI

## Ai Setup WizardAi Setup Wizard

# 🎓 AI-Powered Setup Wizard + 🧠 Prompt Optimizer

## ✅ Module Overview

The AI Setup Wizard is the onboarding brain for each new brand entering the Lavishta-powered SaaS ecosystem. It detects the brand’s industry, tone, audience, product catalog, and social footprint, and configures default behaviors across all modules. It works hand-in-hand with the Prompt Optimizer engine, which helps fine-tune all AI prompt behaviors brand-by-brand.

## 🎯 Goals

Eliminate onboarding friction

Auto-setup defaults per module based on brand insights

Let brands choose tone, language, persona style

Calibrate GPT prompt structures per brand

Recommend best-fit campaign types, auto-post schedules, and language settings

## 🧠 Features Breakdown

### 1. AI Brand Onboarding Engine

Inputs Collected:

Brand name, logo, website URL

Industry (e.g. skincare, fitness, food)

Target audience (e.g. Gen Z, Moms, Men, Teens)

Social links: Facebook, Instagram, WhatsApp

Language preferences

Backend Actions:

Scrapes website + FB/IG for keywords, captions, visual tone

Categorizes industry vertical

Detects common language style (formal, casual, emoji usage)

Analyzes average post formats, offer frequency

Outputs:

Brand Persona Snapshot

Pre-configured GPT Prompt Styles

Auto-defaults for:

Gamification campaign types

Inbox reply tone

Blog voice (expert vs casual)

Language translation toggles

### 2. Prompt Optimizer Engine

Stores AI prompts per module:

Inbox replies

Blog generation

Campaign post captions

Reels scripts

UGC replies

Allows:

Brand admins to edit base prompt for any module

Test run prompt versions and compare output (A/B style)

Lock default prompt or allow assistants to tweak it

Tracks:

GPT token usage per prompt

Response success scores (via thumbs-up/down or click-throughs)

Prompt performance log (accuracy, tone match, etc)

## 🧩 Workflow

### Step-by-Step Brand Setup:

User clicks “Start Setup Wizard”

Uploads logo → auto-color theme extracted

Inputs industry, audience, socials

System shows pre-filled Persona Profile (editable)

Brand chooses tone presets:

Voice: Expert / Friendly / Quirky / Minimalist

Emoji Use: High / Low / None

Language: Bangla / English / Hindi / Urdu

Suggested modules + campaign types shown with toggle

Prompt templates for each feature pre-filled and editable

Auto-save + send summary to admin panel

## 💡 Prompt Editing UI

Live prompt editor per feature (with version history)

Side-by-side comparison of prompt versions

Prompt templating engine:

Variables: {brand\_name}, {audience}, {product\_type}, etc

Conditional logic: IF campaign\_type = contest THEN use engaging tone

Superadmin templates library (can push to brands)

## 🔧 Backend Requirements

### Tables:

brand\_profiles

brand\_id, industry, audience, voice\_style, emoji\_level, language, socials, color\_palette, etc

ai\_prompts

prompt\_id, brand\_id, feature\_name, prompt\_text, version, token\_usage, updated\_by

### Services:

AI Profiler API (scrape and analyze)

Prompt Manager API

Persona Builder Service

Wizard Engine: guides user through onboarding flow

## 🤖 AI Prompts Examples

Prompt to generate brand persona summary:

Based on this brand's website, FB/IG posts, and audience type, summarize the tone, style, and ideal content type they prefer. Provide a summary like:  
- Brand Tone: ...  
- Target Audience: ...  
- Preferred Campaign Style: ...  
- Emoji Usage: ...

Prompt to auto-create inbox reply prompt:

Write a default reply for a skincare brand in Bangla targeting women aged 20-35 asking about delivery. Keep it helpful, casual, warm, and less than 80 words.

## 🔐 Roles & Permissions

Superadmin:

Define prompt templates globally

Set locked prompt zones for brands

Track prompt usage logs

Brand Admin:

Customize prompts

Edit persona setup

Run setup wizard again (if rebranding)

Editors:

Preview prompt outputs

Suggest edits (requires approval)

## 📊 Metrics to Track

Completion rate of Setup Wizard

Prompt edits per brand

GPT usage per module per brand

Response engagement by prompt version

## 🔄 Integrations

Auto-posting: pulls tone from Setup

Inbox auto-replies: use prompt from Wizard

Campaign Generator: chooses formats based on audience fit

Blog Engine: adjusts tone + keywords

Localization: aligns with Setup Wizard’s language pick

## 🧠 Benefits

Faster onboarding for clients

Less support time to customize AI replies

Scalable prompt management across 100s of brands

Consistent brand voice across all outputs

✅ Document complete. Let me know if you’d like to export or illustrate the Setup Wizard UI flows next.

## Intelligent BundlesIntelligent Bundles

# 📦 Product Bundles + Intelligent Collection Builder

## ✅ Module Overview

This module enables brands to create intelligent product bundles and SEO-optimized collections automatically based on trends, user intent, seasonal behaviors, and UGC patterns. The system allows manual, assisted, and fully automated bundle/collection generation. It ties directly with campaign performance, search queries, blog topics, and platform-wide insights.

## 🌟 Core Goals

Increase AOV (Average Order Value) through logical bundling

Automatically generate category/collection pages based on UGC, blogs, and sales

Help brands build curated sets without manual curation

## 🧠 Feature Set

### 1. Product Bundler Engine

Auto Bundle Types:

Frequently Bought Together

UGC Mentioned Sets

Blog + Product Bundles

Seasonal / Occasion-based

Smart Pairing Logic:

Uses co-purchase matrix + keyword vector similarity

Bundle size, price tier, inventory level considered

Bundle Types:

Static (one-time set)

Dynamic (auto-adjusts based on rules)

Conditional (e.g., “For Oily Skin + Under 1000 BDT”)

Display Options:

On PDP (Product Detail Page)

On Homepage Modules

As Campaign Highlights

### 2. Collection Builder Engine

Trigger Sources:

Search queries (e.g., “K-beauty anti-aging”)

Blog internal links

Top UGC clusters (e.g., “summer glow routine”)

Festival triggers

SEO Optimization:

Title + Meta from trending queries

Internal linking from blogs + social posts

Smart canonical & tag-based structure

Collection Logic:

Rule-based: “All products tagged with ‘hydration’ & in stock”

Curated: Drag/drop selection UI

AI Suggested: Based on performance + customer journey

### 3. Brand Panel Features

Create Bundle Wizard:

Auto mode: Choose goal + price range

Manual mode: Add from catalog

Assisted mode: Approve AI suggestions

Collection Dashboard:

Create SEO-ready collections in 3 clicks

Auto-refresh rules (e.g., every Monday re-check matching products)

Preview layout before publishing

Publishing Controls:

Visibility (Public / Campaign-only / Hidden)

Placement: Menu, Homepage, Blog Sidebar, Offer Pages

## ⚖️ Technical Details

### Tables

product\_bundles

collection\_groups

bundle\_rules

auto\_collection\_triggers

bundle\_performance\_metrics

### Services

Bundle Suggestion Engine (cron + on-demand)

Collection Indexer (SEO & sitemap)

Inventory Sync Hook

Smart Filter API (for conditional logic)

### Data Sources

Product tags, sales, inventory

Blog topics + linking

Search logs + vector embeddings

UGC keywords & campaign data

## 📈 Benefits

Enables passive upselling through smart bundles

Allows SEO scaling via high-intent collections

Brands can tap into trends without manual curation

Future-ready for automation-based merchandising

## 🔒 Permissions

Brand Admin:

Create/Edit bundles & collections

Toggle automation per group

Superadmin:

Adjust trigger logic

Review collection SEO compliance

## 🔍 AI Support

Suggest bundle names & descriptions

Auto-generate SEO meta + schema markup

Predict CTR based on historical layout performance

✅ Ready for flowchart, UI planning, and Codex-based build prompts. Let me know the next module.

# Modules/Search

## Smart Search WidgetSmart Search Widget

# 🔍 Smart Search Widget (Embeddable AI Search for Client Sites)

## ✅ Module Overview

The Smart Search Widget enables AI-powered, fuzzy product and content discovery across client storefronts. Built as an embeddable component, it integrates seamlessly into WooCommerce, Shopify, or custom frontend pages. Powered by real-time vector search and OpenAI-optimized embeddings, it provides natural-language understanding for beauty shoppers (e.g., “best Korean moisturizer for oily skin”).

## 🎯 Core Features

### 1. AI-Powered Product Search

Understands user queries using semantic meaning, not exact keyword match

Maps user intents to product metadata, benefits, tags, ingredients

Supports multi-language search (Bangla, English, Hindi)

### 2. Embeddable Widget

JavaScript snippet to embed on any site section (homepage, category page, mobile app)

Supports theme-based styling and layout config

Responsive + fast (sub-200ms response)

### 3. Real-Time Vector Search Engine

Uses OpenAI or custom embedding model to index product data

Search index stored in high-performance vector DB (e.g., Weaviate, Pinecone, or custom MySQL-based vector store)

Auto-updates with product changes or new blog entries

### 4. Unified Search: Products + Blogs + UGC

Returns hybrid results: product cards, helpful blogs, campaign posts

Boosts UGC/blogs that match user concern or skin type

Click-through sends user to canonical page

### 5. Configurable Behavior

Superadmin can define field weights: e.g., boost tag match > ingredient match

Clients can enable/disable blog/UGC search

Custom synonym map per client (e.g., “zits” → “acne”)

## 💡 UX & Integration

### Widget Options:

Inline search bar

Popup modal on click

Fixed floating icon on page bottom

### Data Shown:

Thumbnail + product name + benefit summary

Price, rating, and “Add to Cart”/“View More”

Blog snippet if applicable (e.g., “Why This Moisturizer Suits Oily Skin”)

### Integration APIs:

GET /ai-search?q={query}&brand\_id=123

POST /update-embeddings → Sync products

GET /config/search-behavior → Loads weights/preferences per brand

## ⚙️ Technical Stack

Frontend: Vanilla JS + CSS for embeddable version (with React variant for React apps)

Backend:

Search API (Node.js or Python FastAPI)

Vector DB layer (Weaviate, Qdrant, or Pinecone)

Embedding Generator (OpenAI, MiniLM, or custom finetuned model)

Database Tables:

product\_embeddings: product\_id, embedding, brand\_id, last\_updated

search\_logs: query, user\_ip, results\_shown, clicked\_item

## 🔐 Permissions & Access

Brand Admin:

Customize layout, weight preferences

Enable/disable blog/ugc blending

Superadmin:

Set default config

Manage indexing intervals

Audit performance + logs

## 📊 Metrics & Analytics

Query volume by brand

Click-through rate per search

Zero-result queries

High-engagement search terms

## 🔄 System Events

New product added → triggers update-embeddings

Blog generated → added to index if blending enabled

UGC campaign ends → related content added to vector index

## 🤖 Example Prompt (Embedding Context Generator)

“Generate an embedding vector from the following product name, description, benefits, tags, and concerns, suitable for AI-powered ecommerce search matching. Output: single dense vector.”

✅ Module ready. Would you like to move on to the next one?

# Modules/Customer

## Customer Identity Mapping ModuleCustomer Identity Mapping Module

# 👤 Customer Identity Mapping & UGC Matching Module

## 🧩 Purpose

This module connects the dots between UGC participants (commenters, inbox users, WhatsApp senders) and existing ecommerce customers (WooCommerce/Shopify). It unlocks:

Accurate reward distribution for gamified campaigns

Mapping historical orders with social activity

Building user profiles across FB, IG, WhatsApp, and web

Enabling remarketing, retention, and personalization

## 🧠 Core Features

### 1. Customer Profile Builder

Centralized identity graph combining multiple touchpoints

Attributes: full\_name, fb\_id, ig\_username, phone, email, location, comment\_history, inbox\_msgs, orders, campaigns\_joined

Dynamic confidence score (0-100) per identity match

How It Works: - Collect FB/IG/WhatsApp identifiers (e.g., name, username, phone) - Run fuzzy match with ecommerce DB (email, phone, name) - Boost score if inbox history or UGC matches past campaign logs

DB Schema: customer\_identity\_map - social\_id, platform, matched\_user\_id, confidence\_score, source

### 2. Customer Upload Tool (Add-On)

Allows brand to upload their past customer base

Fields: name, phone, email, city, order\_id

Stored for future matching with FB/IG/WhatsApp interactions

Settings Panel: - GDPR checkbox (“I confirm consent”) - Upload validator and field mapper

### 3. Inbox Conversation Scraper

Pulls past conversation messages from FB/IG inbox

Runs NLP pipeline to extract names, locations, products, concerns

Uses these to strengthen matching

How to Build: - Use Meta’s Inbox API to fetch message threads - Text preprocessing to extract PII and intent - Update identity graph

### 4. Comment Identity Mapper

Maps public comments to known users

Especially useful in campaign leaderboards and points tracking

Match Strategy: - fb\_name vs order\_name - comment text → if they wrote “ordered 2pcs” or gave phone - campaign id → check if same user inboxed

### 5. Cross-Platform History Panel

View customer’s journey across:

FB comments

IG story replies

Inbox

Orders

WhatsApp DMs

Visual timeline and tags for each action

Useful For: - Support agents - Campaign auditing - High-LTV customer identification

### 6. Email/SMS Fetcher (with Consent)

Attempts to locate emails or phone from inbox/chat context

If found, stores and links to ecommerce DB

Consent logic ensures opt-in-only mapping

## ⚙️ Technical Stack

Node.js (API Layer)

MongoDB (Flexible identity graph)

Redis (Match score caching)

Meta Graph API, WhatsApp Business API, Shopify/Woo REST APIs

## 🔐 Privacy & Compliance

All mappings require brand opt-in

GDPR-compliant logging, opt-out support, consent record per match

Superadmin can disable module or redact entries

## 🔧 Superadmin Controls

Module access toggle per brand

Upload quota per brand

Enable/disable auto-mapping

Review identity graph logs and conflict resolution

## 🔗 Integration Touchpoints

Gamification → fetch participant identity, reward accurately

Blog Engine → pull quotes or product tips from UGC

Loyalty Engine → show unified reward history per user

Inbox Module → show mapped profile when user DMs

## 🧪 Example Use Case

A customer named “Mimi Akter” comments on a Lavishta post: “Ordered the hyaluronic set last night from your site. Love it!”

Comment parsed → product: hyaluronic, tone: positive

Name = “Mimi Akter” matched to Woo customer with same name + phone match

Inbox history shows convo about hyaluronic

System links her FB ID → Woo customer ID → adds 10 points to campaign

✅ Module ready for production. Connects deeply with engagement, campaign, and support modules.

## Customer Mapping EngineCustomer Mapping Engine

# 🧬 Customer Mapping + UGC Matching Engine

## ✅ Module Overview

This module creates a unified identity graph that connects all customer touchpoints across platforms (Facebook comments, inbox DMs, WhatsApp replies, Instagram threads, and ecommerce activity). It ensures brands can:

Recognize returning users across channels

Associate past orders and behavior with new comments or inboxes

Power personalization, loyalty, and analytics

## 🎯 Goals

De-duplicate customer identities across platforms

Link ecommerce (WooCommerce, Shopify) users to social interactions

Auto-match UGC (comments, DMs) to customer profiles

Enable personalized replies, targeted rewards, and predictive triggers

## 🧠 Key Features

### 1. Customer Identity Graph Builder

Input Sources:

Website: Name, Email, Phone, Order History

Facebook: Comment name, Inbox name, Messenger ID

Instagram: IG handle, message threads

WhatsApp: Number, replies

Matching Algorithms:

Email exact match

Phone fuzzy match (with +880, 880, 01 variations)

Name + IP/Device patterns

Inbox scraping for delivery address/history

Graph Node Storage:

customer\_nodes table:

node\_id, platform\_id, platform\_type, name, email, phone, ig\_handle, wc\_user\_id, mapped\_to

Merging Engine:

Background job that runs daily to consolidate identity clusters

Manual override interface for Admins

### 2. UGC Matcher Engine

Purpose: Map a Facebook/Instagram comment to a real user profile

Triggers:

New comment on campaign post

New message in Messenger/Inbox

Matching Logic:

Match by Name + Existing Comment History

Fuzzy match by Delivery Address (scraped from past messages)

Phone/email extraction from UGC (if any)

Output:

Mapped customer\_node\_id

Past orders summary

Loyalty tier & rewards

UGC history summary (comment counts, engagement)

### 3. Manual Upload + Mapping Interface

Brands can bulk upload their past CRM/customer CSVs

Columns: Name, Phone, Email, Last Order, Total Orders, Notes

System tries to map uploaded entries to existing social UGC profiles

Tools Provided:

Conflict Resolver: Pick correct match if duplicate

Confidence Score Viewer: AI prediction confidence in linking

Edit/Add Notes for Customer Profiles

## 🔄 Workflow Example

A user comments on a giveaway: “Hi, I bought this 3 weeks ago!”

System sees name = “Fariha Rahman” and Messenger ID = X

Previously uploaded CRM has a “Farhia Rahman” with same phone

Fuzzy match (90%) → identity matched

Profile enriched with:

2 past orders

Bronze loyalty tier

4 past UGC comments

Auto-reply includes personalization: “Welcome back Fariha! 🎉”

## 🔐 Data Privacy & Security

GDPR-compliant mapping only with brand-collected data

Customer Upload explicitly consented

Internal logs of all mapping actions

Mapping can be disabled per brand

## 🔧 Tables & Backend Infra

customer\_nodes

ugc\_events

mapping\_jobs

manual\_uploads

identity\_conflicts

### Services:

Identity Mapper Engine

UGC Extractor + Pattern Matcher

Confidence Score Calculator (ML based)

CRM Importer Service

## 💡 Benefits

Personalized replies & auto-rewards

Accurate campaign targeting (e.g., “Only for repeat buyers”)

Clean, deduplicated CRM for remarketing

Future prediction models can work better

✅ Module draft complete. Let me know if you want visual flows or integration maps next.

## Customer Upload Mapping EngineCustomer Upload Mapping Engine

# 🧩 Module 7: Customer Upload + Mapping Engine

## 🎯 Overview

This module allows each brand to securely upload their existing customer base (name, phone, email, address, previous order data) and automatically match these users when they: - Comment on a Facebook/Instagram campaign - Message in inbox/DMs - Participate in UGC - Place orders or engage on the brand’s website/app

The goal is to centralize identity across all channels to: - Enable personalized replies - Send follow-up Email/SMS campaigns - Auto-fill user profiles in leaderboards - Improve reporting & attribution

## 🗂️ Core Features

### 1. CSV Upload Interface

Brands can upload a CSV file with fields: name, email, phone, address, order history, user ID (optional).

Real-time column mapping UI with preview

Validation for email/phone format

Upload limit settings (configurable per plan)

### 2. Identity Normalization & Indexing

Phone/email is normalized (e.g., removing +880, 0-prefix)

Addresses tokenized to allow approximate match

Hash map & inverted index to speed up lookup during future mapping

### 3. Cross-Channel Mapping

When new UGC, comment, or inbox data comes in:

Match commenter’s name with known name + fuzzy rules

Match phone/email (if exposed via Meta API) directly

Infer connection based on past inbox chats, shared address, city, zone

### 4. Matching Confidence Scoring

Score is generated based on exact match, fuzzy name match, behavioral signal overlap (like area, product interest)

Admins can define minimum score threshold to auto-map

UI to approve/reject edge cases

### 5. Profile Enrichment

Matched user data is linked to existing or new system profile

Used to:

Auto-fill leaderboard profiles

Enable reward dispatch

Add to email/SMS flows

Run user-specific reporting

### 6. GDPR + Permissions

Consent field available per user (“allow marketing communication”)

Deletion API for GDPR right to erasure

Uploaded data is siloed per brand with encryption at rest

## 🔧 How to Build It

### Stack

Backend: Node.js + PostgreSQL + Redis

Frontend: Next.js + Tailwind

File Parsing: PapaParse for frontend; csv-parser or fast-csv in backend

Identity Match: Fuse.js or custom fuzzy engine for names; phone-lib for phone normalization

### DB Tables

customer\_uploads: upload metadata per brand

customer\_profiles: master profile with UID, email, phone, last\_seen

customer\_mappings: tracks cross-platform connections

consents: GDPR, opt-in, communication permission

### API Endpoints

POST /upload-customers → Upload and validate

GET /uploads/:brand\_id → Show previous uploads

POST /match-customer → Attempt to resolve identity

POST /delete-customer/:uid → GDPR compliance

## ⚙️ Superadmin Controls

Enable/disable module for brand

Set upload quota

Set mapping confidence threshold

Logs of all customer matches (audit trail)

Force full re-matching across brands

## 🧠 AI Assist Add-ons (Optional)

AI Identity Matcher: GPT powered identity suggestion with reason

Address Normalizer: AI-based city/locality correction from partial/incomplete address

GPT-powered UGC reader: Suggest matching customer from UGC or inbox message (“Hi, I ordered a pink serum last month”)

## 🔗 Connected Modules

Email/SMS Campaign Builder → audience segments, abandoned cart flows

UGC/Gamification → map comments to user

Inbox/Messenger → recognize repeat users and personalize

Analytics → accurate cohort reporting, per-customer lifetime value

## 📈 Usage Benefits

Higher personalization & engagement

Unified customer identity across sales channels

Enhanced marketing attribution

Lower manual effort for mapping DMs to users

✅ This module will function as a standalone add-on and also empower nearly every other module when enabled.

## Customer Upload MappingCustomer Upload Mapping

# 📦 Customer Upload & Identity Mapping Module (Add-On)

## ✅ Overview

This module allows brands to upload and manage their customer base to enable deeper personalization, UGC engagement tracking, inbox-to-order mapping, and omni-channel identity resolution. It works as a foundation layer for automations, reward attribution, and remarketing personalization.

## 🎯 Key Objectives

Upload existing customer data from CSV/Excel

Match social identities (Facebook, Instagram, WhatsApp) to known customers

Track UGC engagement and link to known user profiles

Enable rewards, campaigns, inbox follow-ups based on mapped customer actions

Ensure GDPR-compliant secure handling and control

## 🔄 Core Features & Workflows

### 1. 🔼 Customer Upload & Validation

Admin uploads CSV file containing customer data with fields like:

Name, Phone, Email, Address, Order History, Tags

System validates file structure and formats

Duplicate detection by phone/email

Email/Phone normalization & deduplication

Manual override for conflicts

### 2. 🔗 Identity Linking Engine

Automatically attempts to match:

Phone numbers to WhatsApp contacts

Emails to customer database or order logs

Facebook/Instagram profile names from comment/inbox with past records

Uses fuzzy logic and partial matches to link user activities with profile

Manual override in case of ambiguity

Webhooks from Inbox module auto-trigger mapping attempts

### 3. 👤 Profile Enrichment Layer

Each customer gets a unified profile in DB:

Basic info (Name, Phone, Email, Address)

Social Links (FB Profile URL, IG Handle, WhatsApp ID)

Order History (from website sync or upload)

Campaign Participation Logs

Rewards / Tier Status

### 4. 🕸️ Unified Activity Log

Every tracked action per customer is stored chronologically:

FB comment on campaign

WhatsApp question about a product

Order placed via website or offline

Email campaign clicked

Allows precise segmentation and personalization

### 5. 🔐 GDPR Controls & Consent

Admin must confirm consent before upload

Checkbox per customer entry: “Consent to store and use data”

Deletion API: removes all traces of a customer upon request

UI setting to export all data linked to a customer (for compliance)

## ⚙️ Technical Design

### DB Tables:

customers — unique profiles

customer\_contacts — phones, emails, social IDs

customer\_activities — timeline of events

customer\_tags — admin-assigned labels like “VIP”, “From Group”

### APIs:

POST /upload-customers — takes file, returns preview of imported entries

POST /map-social-identity — links social activity to customer ID

GET /customer/:id/activity — fetch complete profile and logs

DELETE /customer/:id — delete + anonymize data (GDPR)

### Stack:

Backend: Python + FastAPI

DB: PostgreSQL

File Parser: Pandas

Fuzzy Mapping: RapidFuzz

Identity Match Engine: Async Tasks + Redis Queue

Frontend: ReactJS (for preview/merge UI)

## 🧠 Integration Points

📬 Inbox Assistant → auto-map chat contacts to CRM

🎯 Campaign Engine → assign points only to mapped profiles

📤 Email/SMS → send to matched users only

🏷️ Tag Engine → auto-tag based on interaction pattern (e.g., “Frequent Commenter”)

## 🔐 Superadmin Controls

Enable/disable this module per brand

Limit CSV upload size per plan

Toggle identity match strength thresholds

Set GDPR deletion frequency

## 📝 Future Upgrades

AI matching for profile pictures (if privacy compliant)

OTP-based customer verification to enrich profiles

Integration with Meta Custom Audiences (for ad targeting)

Smart merge tool for fuzzy duplicate records

## ✅ Why It Matters

Enables unified customer view across inbox, campaign, store

Powers personalized replies, remarketing, rewards

Builds long-term relationship map per user, increasing LTV

Unlocks advanced analytics and micro-segmentation

This module is essential to bridge offline + social + website data and create a seamless 360° customer view.

# Modules/Analytics

## Analytics DashboardsAnalytics Dashboards

# 📊 Analytics Dashboards (UGC / SEO / Engagement)

## ✅ Module Overview

The Analytics Dashboards module delivers visual, actionable insights for brand owners to track User-Generated Content (UGC) performance, SEO growth, and audience engagement across all connected platforms (Facebook, Instagram, website, blogs). It works independently for each brand while allowing the superadmin to monitor overall system metrics.

## 🎯 Goals

Provide real-time visibility into campaign and content performance

Empower brand owners to make data-backed decisions

Consolidate metrics from social, blog, product, and UGC pipelines

Drive higher ROI on campaigns by highlighting winning patterns

## 🧠 Feature Set

### 1. UGC Dashboard

Metrics Tracked:

Total UGC entries (by campaign, type, time)

Participation rate (comments, shares, tags)

Campaign CTR / virality score

Reward redemption vs earned points

Top-performing users / contributors

Widgets:

Campaign Leaderboard

Tag Cloud of most common UGC words

Virality Timeline Chart

UGC vs Sales Funnel Impact

Filters:

Time range

Campaign type

Platform (FB/IG)

Demographics (if available)

### 2. SEO Dashboard

Metrics Tracked:

Blog views, CTR, bounce rate

Meta title performance (click-throughs, impressions)

Keyword ranking trend (Google Search Console API)

Internal link engagement

Widgets:

Keyword Position Timeline

Blog Performance Table

Auto Blog vs Manual Blog Comparison

Search Intent Bubble Map (AI-generated)

Filters:

Category (e.g., skin, makeup)

Blog type (auto / manual)

Keywords

Rank buckets (1-10, 11-30, etc)

### 3. Engagement Dashboard

Metrics Tracked:

Likes, shares, saves, comments per post

Post format analysis (carousel, video, AI-generated)

Follower growth / churn

Engagement by campaign or AI template

Widgets:

Engagement Heatmap by Week/Day

Format Performance Comparison

Top Posts Table

AI Template Effectiveness Chart

Filters:

Platform: Facebook / Instagram / Blog

Format: Video, Text, Carousel, Reel

Campaign tag

## 🧰 Tech Requirements

### Tables Required

ugc\_analytics

seo\_analytics

social\_engagement

brand\_metrics

### APIs + Data Sources

Facebook Graph API (insights)

Instagram Insights API

Google Search Console API

Google Analytics (blog views, behavior)

Internal system logs (reward points, blog generator, etc)

### Services

Analytics Aggregator (scheduled job to collect and store)

Chart Generator API (for all visualizations)

Filter Engine with caching

## 🎨 UI Components

Unified dashboard homepage with tab navigation

Each widget card can be exported as PNG/CSV

KPI boxes with daily % change

Tooltip explanations for non-tech users

Toggle between brand view and superadmin view (with averages and trends)

## 🔐 Permissions

Superadmin:

View all brands

Compare brands

Push global benchmark alerts

Brand Admin:

See only their own metrics

Export data

Share links with their team (view-only)

Analysts:

Drill-down access

Historical trend view

Anomaly detection logs

## 💡 AI-Enhanced Insights

Smart suggestions: “This blog post is underperforming due to low meta title CTR. Try revising title.”

Anomaly detection: “Campaign XYZ had unusually low participation on July 15. Possible overlap with national holiday.”

UGC clustering: Auto-tag themes across UGC to recommend new blog/campaign ideas

## 🧠 Benefits

One dashboard to rule all growth metrics

Helps tie UGC and blogs back to ROI

Data-backed iteration of AI campaigns

Drives informed product decisions via community insight

✅ Ready for export or flowchart generation. Let me know your next action.

# Platform

## Gdpr Compliance And Api Usage PanelGdpr Compliance And Api Usage Panel

# 🔒 GDPR Compliance + API Usage Control Panel

## ✅ Module Overview

This module ensures that the SaaS platform strictly adheres to global privacy and data usage standards (especially GDPR), while offering each brand owner complete transparency and control over how third-party APIs are used across the system. It is crucial for building trust with clients, protecting user data, and staying legally compliant across jurisdictions.

This panel also gives Superadmins fine-grained control over API usage limits, throttling rules, audit logs, and billing alignment for each brand using third-party resources.

## 🎯 Goals

Comply with GDPR and similar global data protection laws.

Allow full audit and control over 3rd-party API access, data syncs, AI usage.

Let customers manage consent, deletion, and data portability requests.

Track per-module API usage, cost, and limits by brand.

Prevent overuse and abuse of AI tools or integrations.

## 🧩 Features Breakdown

### 1. GDPR Settings Panel (Brand-Level)

Consent Management:

Consent banners + checkboxes embedded in widgets/forms (UGC campaigns, comments, profile pages).

Records user consent timestamp and metadata in a separate audit log.

Data Rights Interface (Frontend):

“Download My Data” request button.

“Delete My Account + All Data” request.

Option to revoke previous consent.

Data Exports:

JSON/CSV export of customer data: interactions, orders, replies, UGC entries, campaigns joined, reward points.

Privacy Policy Link:

Auto-injected into all user-facing pages, UGC forms, and email footers.

### 2. Superadmin API Usage Dashboard

View all 3rd-party APIs used per brand:

GPT/OpenAI

Facebook/Instagram Graph API

WhatsApp Business Cloud API

Shopify/WooCommerce APIs

Email/SMS gateways (e.g., Twilio, SendGrid, WhatsApp)

Metrics Tracked:

API calls/day/week/month

Token usage for GPT calls

Errors, retries, fallbacks triggered

Latency and rate limits

Tools:

Throttling rules: Define daily/weekly quotas per brand/module/API

Cost mapping: View API cost breakdown per feature/brand

Disable toggle for any integration at brand or global level

Alerts for overage or abuse

### 3. Compliance Engine Services

Data Audit Logger:

Tracks every data write, sync, or API response involving PII

Logs brand, timestamp, endpoint, user ID, payload hashes

Data Access Request Handler:

When customer clicks “Delete My Data”, notifies brand admin

Auto-deletes data if no response within X days (configurable)

Purges data across:

Comment logs

UGC

Reward profile

CRM fields

Campaign history

Consent Syncer:

Pushes/revokes consent from customer to 3rd party if supported (e.g., Shopify customer deletion API, FB data deletion callback, etc)

## 🔐 Backend Tables

api\_usage\_logs:

id, brand\_id, module, api\_name, call\_count, tokens\_used, cost\_estimate, timestamp

consent\_logs:

user\_id, brand\_id, consent\_type, timestamp, source

data\_deletion\_requests:

user\_id, brand\_id, request\_time, status, completion\_time, modules\_affected

gdpr\_settings:

Per brand configuration: auto-delete time, consent enforcement toggle, export enablement

## 📊 Admin Insights

Weekly email to Superadmin:

API usage summary across brands

Consent violations or pending deletion actions

Brands nearing quota

Brand-level dashboard shows:

Their current quota & cost estimation

Last 30 days’ usage

Toggle consent & deletion workflow

## 🧠 Benefits

Prevents legal/regulatory risks at scale

API usage tracking keeps server cost optimized

Puts brands in control of customer privacy

Adds transparency and confidence to clients

✅ This document is ready. Let me know if you want GDPR request flows or API usage charts designed next.

## Modular Plugin SystemModular Plugin System

# 🔌 Modular Plugin System

## ✅ Overview

The Modular Plugin System allows the SaaS platform to scale flexibly by letting developers and partners build, install, and manage additional feature modules as plugins. Brands can selectively enable these modules via their dashboard. Superadmins maintain control over approvals, access levels, usage limits, and billing.

## 🧩 Core Goals

Let third-party developers or internal teams add new features without changing core code

Allow brands to selectively enable or disable modules based on need or plan

Provide a plug-and-play architecture for extending system capabilities

Maintain centralized governance over module integrity, permissions, and API usage

## 🏗️ Architecture & Components

### 1. Plugin Registry

Central database of all available plugins

Fields:

plugin\_id, name, version, developer\_id

category (e.g., Marketing, Analytics, Engagement)

description, required\_permissions, dependencies, status

is\_approved, is\_default\_enabled, price\_model

### 2. Plugin Loader Engine

Dynamically loads plugins into the dashboard and system

Supports:

UI injection (dashboard widgets, feature tabs)

Backend APIs (routes and logic injection)

Scheduled jobs (CRON-compatible plugins)

### 3. Plugin Sandbox (Security Layer)

Each plugin runs in an isolated sandbox environment

No direct access to global variables, core services unless allowed via API proxy

Rate limiting and logging enforced

### 4. Superadmin Plugin Panel

Approve or deny plugin submissions

Assign modules to brands or packages

Set limits: API tokens, GPT usage, UI features

Disable plugins across brands if security/bugs are detected

### 5. Developer Dashboard (External Partners)

Submit plugin via manifest + zipped code repo

See sandbox results

Test plugins using test tenant

Submit version upgrades

Documentation guidelines and plugin rules available

## ⚙️ Plugin Manifest Example

{  
 "name": "Pinterest Poster",  
 "version": "1.0.2",  
 "category": "Marketing",  
 "main": "index.js",  
 "permissions": ["post\_generation", "media\_upload"],  
 "ui\_hooks": ["campaign\_editor", "post\_scheduler"],  
 "api\_hooks": ["/create-post", "/upload-media"],  
 "scheduled\_jobs": ["weekly\_summary\_report"],  
 "dependencies": ["image\_generator"],  
 "config\_schema": {  
 "apiKey": "string",  
 "boardId": "string"  
 }  
}

## 🔐 Security & Permissions

Plugins can request only approved permissions

Plugins with AI or API usage are metered and billable

Superadmins can disable plugins immediately across tenants

All plugin logs stored per brand

## 📦 Usage Flow

### For Superadmin:

Review plugin submission

Approve and categorize plugin

Enable plugin for specific brand(s) or tiers

Monitor plugin usage (metrics + logs)

### For Brands:

Visit Plugin Marketplace in dashboard

Enable/disable available plugins

Configure per plugin settings (API keys, toggle on/off, schedules)

Access plugin features inside core modules (if UI hooks defined)

## ✅ Example Plugins

Pinterest Poster

WhatsApp Reply Handler

Review Booster

TikTok Content Syncer

UGC Heatmap Analyzer

LinkedIn Auto-Scheduler

## 💡 Benefits

Extend core SaaS features without bloat

Build a developer ecosystem

Offer exclusive features as upsells

Easily launch limited-time experiments

Support region- or industry-specific innovations

✅ Modular Plugin System documentation is now complete.

## Platform Oauth IntegrationPlatform Oauth Integration

# 🔐 Platform-Level Meta App Integration (OAuth & Token Proxy Layer)

## ✅ Module Overview

This module enables all connected brands to leverage a single platform-level Meta (Facebook/Instagram) app for authenticating and accessing social features (Pages, Messenger, IG DMs, etc.) without needing to register their own Meta Developer accounts. The system uses a secure proxy token and permission layer to route access and usage based on tenant configurations.

## 🎯 Goals

Eliminate the need for each brand to register their own Facebook app

Centralize and secure token storage, refresh, and API call routing

Handle permissions, scopes, and reauthentication logic across tenants

Enable fine-grained access control per brand

Ensure GDPR and Meta Platform policy compliance

## ⚙️ System Architecture

### Key Components:

Meta App (Platform-Level):

Single approved app with permissions for: pages\_messaging, instagram\_manage\_messages, pages\_read\_engagement, etc.

Admin-controlled by the SaaS owner

OAuth Proxy Service:

Handles Meta login redirects and auth callback

Stores and encrypts access tokens securely per brand

Maps token to tenant-specific brand ID

Token Vault:

Encrypted storage layer (e.g., using AWS KMS, HashiCorp Vault)

Handles refresh tokens automatically via scheduled background workers

Tenant Permission Router:

Validates if the tenant has enabled Facebook/IG modules

Checks user roles and active scopes before allowing access

Maintains audit logs for all token usage

Usage Throttler + Tracker:

Monitors daily API limits and alerts superadmin if nearing quota

Reports usage per brand for fair billing and throttling

## 🧩 Feature Capabilities

Connect Facebook Page → triggers OAuth flow

Admins see which Page(s) are connected

Brand admins can disconnect/re-auth when needed

Token expiry reminders + auto-refresh flow

Option to manually force re-authentication per brand

Logs all access tokens, scopes, last-used timestamp, and expiry

FB/IG API calls are made using token matched to brand’s assigned profile

## 🔐 Security Controls

Token encryption at rest using platform-wide secure keys

API calls routed through internal proxy service (not directly from frontend)

Tenant isolation by verifying brand\_id on each API access

Superadmin controls access toggles per brand:

Enable Facebook integration

Enable IG DM access

Enable Inbox automation

Logs per token usage: method, response, latency, user

## 🔄 Integration Points

Inbox Module → Meta token for FB/IG replies

Campaign Module → FB Post auto-generation

Gamification → Tracks comment-based campaigns

Blog Engine → Auto-publishing via Facebook Page

Auto Poster Scheduler → Uses Page/IG token to post at set times

## 📊 Admin Dashboard View

List of all brands using Meta App

Pages/IG profiles connected

Token health (valid, expired, expiring soon)

Last used, last refreshed, next refresh

Actions: Revoke / Reauth / Resync Scopes

## 🔧 Developer Notes

### Database Tables:

brand\_meta\_tokens: brand\_id, page\_id, access\_token, scopes, expires\_at, last\_used, token\_status

meta\_token\_logs: brand\_id, method, timestamp, response\_code, duration

### Services:

MetaOAuthService

MetaTokenRefreshWorker

FBApiProxyHandler

## 📘 Sample Prompt for Codex Integration:

“Build a secure OAuth handler for Facebook that saves access tokens for a multi-tenant app. Encrypt tokens, map them to brand\_id, store scope and expiry. Allow secure token refresh and proxy all FB API calls through a backend handler. Add admin dashboard to manage all tokens by tenant.”

✅ Module finalized. Let me know if you’d like this integrated with the Inbox module next or proceed with the next pending module.

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

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:

Extract product attributes:

Fields: title, brand, category, size, color/shade, ingredients, concerns, tags, benefits, usage.

Preprocess (Tokenizer & Cleaner):

Remove symbols, standardize synonyms, deduplicate terms, fix typos (e.g., “niacinamide” vs “niacinmid”).

Generate Embedding:

Use OpenAI text-embedding-3-small.

Payload: concatenated string of full product details.

Store Vector:

FAISS DB with metadata: { sku, brand\_id, category\_id, size, vector, source\_text, created\_at }

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.

## Shared Oauth Infrastructure ModuleShared Oauth Infrastructure Module

# Module 8: Shared OAuth + App Infrastructure

## ✅ Module Title: Shared OAuth + App Infrastructure (Multi-Tenant)

## 🎯 Purpose:

This module ensures that all Facebook, Instagram, WhatsApp, Google, and Shopify/WooCommerce integrations are powered by a central set of platform-level apps. This avoids each brand/customer having to create their own developer app and allows the SaaS owner (Lavishta) to control, monitor, and secure all integrations from a single point.

## 🧩 What It Enables:

One-time authentication and permissions via platform-managed apps.

No need for individual customers to go through Facebook App Review.

Faster onboarding via click-to-connect.

Central audit logs, token refresh system, and rate limit handling.

OAuth token management per brand, scoped to their data only.

## 🔐 Platforms Supported:

Facebook/Instagram (Meta Graph API + Webhooks)

WhatsApp Business Cloud API

Google (Gmail, Google Trends, Google Analytics, Search Console)

WooCommerce REST API

Shopify Admin API

## 🏗️ System Architecture:

Platform-level App (e.g., 1 Facebook App, 1 Google Project)

OAuth2 Flow per brand (stored in brand\_tokens table)

Access Token + Refresh Token securely encrypted

Webhooks (e.g., Meta Post/Comment Events) routed to tenant-aware handler

Internal Auth Proxy to wrap calls and tag them with brand context

## 🧰 Database Tables:

brands: holds info like brand\_id, domain, status

brand\_integrations: brand\_id, platform, client\_id, token, refresh\_token, scopes, connected\_on, expires\_at

webhook\_logs: brand ID, event type, payload, processed time

## 🧪 Example Integration Flow (Facebook):

User clicks “Connect Facebook Page” in their brand panel

Redirects to Meta OAuth URL with platform app ID

After user accepts permissions, Facebook redirects to /oauth/facebook/callback

SaaS backend exchanges code for access\_token

Store token in brand\_integrations with scope = pages\_messaging, pages\_read\_engagement

Register Webhook with callback = https://platform/api/meta/webhook and brand context

Incoming webhooks now contain Page ID → map to brand\_id

## 🔐 Security Controls:

AES256 encryption for tokens

Scoped access per brand

Tokens auto-refreshed via CRON every 12h

Superadmin can manually revoke or re-auth

All API calls to Meta/Google/Woo/etc are wrapped by a brand-aware proxy with rate limiters

## 🧑‍💼 Superadmin Controls:

Enable/Disable any integration per brand

View logs of connection attempts, token expiry, webhook failures

Re-auth token manually

Limit API usage per brand to avoid abuse

## 🚦 Error Handling:

Token expired → auto-refresh or show “Re-auth Required” warning

Invalid scope → show in brand panel with fix instruction

Connection failure → log and alert superadmin

## 🧠 AI-Powered Onboarding:

Setup Wizard (per brand): Auto guides users to connect Meta, WhatsApp, Gmail, Woo/Shopify

Tips: Explains what each permission does and how it will be used

Flags missing scopes (e.g., user skips “Instagram Access”)

## 🌐 Admin API Endpoints:

POST /oauth/:platform/start

GET /oauth/:platform/callback

GET /integrations/:brand\_id

POST /integrations/revoke

POST /integrations/refresh

## ⚙️ Tech Stack:

Backend: Node.js (Express) or Python (FastAPI)

DB: PostgreSQL

Encryption: AES256 + Vault secrets

Queue: Redis (for refresh retries)

Frontend: React-based Panel with real-time status widgets

## ✅ Result:

This module makes it extremely easy for brands to connect 3rd-party platforms securely with 1-click. The SaaS superadmin has complete control, audit visibility, and modular enable/disable capabilities.

✅ Fully isolated per brand ✅ Secure token management ✅ Platform apps managed once centrally ✅ Reduces onboarding time to <1 min

Next Module: Inbox Reply Module (FB, IG, WhatsApp unified)?

## White Label Rbac ModuleWhite Label Rbac Module

# 🎨 White-label Branding & Role-Based User System

## ✅ Module Overview

This module supports SaaS-level scalability by enabling full white-label support for agencies/resellers and enforcing fine-grained, secure access control through Role-Based Access Control (RBAC). This allows SaaS owners to let brands rebrand the entire panel, while also offering multiple user roles within each brand team (e.g., Admin, Editor, Analyst, Support).

## 🎯 Goals

Empower resellers/agencies to offer the system under their own branding

Give brand teams the ability to securely delegate tasks

Control access to each feature per user

## 🧱 Core Features

### 1. White-Label Support

Custom Branding:

Upload brand logo

Set primary/secondary color schemes

Custom favicon, app name, login page banner

Add custom domain (e.g., ai.brandname.com)

Custom Panel Content:

Replace footer, terms of use, about us

Custom email sender name + logo in system emails

Hide “Powered by” or replace with agency name

Reseller Dashboard (Superadmin only):

Manage brands onboarded under each reseller

Assign allowed modules, pricing plans

Access analytics per white-label portal

### 2. Role-Based Access Control (RBAC)

Roles:

Superadmin: Master control of SaaS

Reseller: Manages white-label clients

Brand Admin: Manages own brand, invites team

Editor: Can write/edit content but not publish

Analyst: View-only access to analytics & reports

Support: Access inbox, campaign results only

Permission Matrix:

Toggle per feature/module (e.g., Campaign Builder, Inbox AI, Blog Generator)

Control actions (Create, Edit, Delete, Publish, View)

Custom Roles:

Allow brands to define custom roles (e.g., Marketing Intern)

Audit Logging:

Track all actions by user, time, and module

## ⚙️ Technical Details

### Tables

user\_roles

role\_permissions

reseller\_brands

white\_label\_config

audit\_log

### APIs

POST /admin/assign-role

GET /roles/permissions

PATCH /white-label/settings

GET /reseller/stats

### UI Panels

White-Label Setup (Superadmin)

Domain, branding assets, module toggles

User & Role Manager (Brand Admin)

Invite users, assign roles, revoke access

Permission Matrix Viewer

Interactive table to configure feature-level access

Audit Log Viewer

Filter by date, user, module, action

## 🔐 Security & Compliance

JWT/Token-based access control per role

Enforced at frontend + backend route levels

GDPR & ISO-friendly audit trails

## 🧩 Integrations

Applies across all modules: Inbox AI, Campaigns, Analytics, Blog, etc.

Seamless integration with Reseller Panel, Customer Upload, Admin Dashboard

## 💡 Benefits

Resellers can scale sales without exposing core SaaS brand

Brands can delegate work safely to their team

Encourages agency adoption and multi-team collaboration

✅ Ready to proceed with flowcharts, component wiring, or backend code prompts.

# Developer

## Developer Toolkit DocsDeveloper Toolkit Docs

# 🛠️ Developer Toolkit Module (API, Webhooks, Docs)

## ✅ Module Overview

The Developer Toolkit provides all external developers, partners, or brand-side engineers with robust tools to integrate, extend, or automate Lavishta SaaS features. It offers programmatic access via REST APIs, event-driven Webhooks, dynamic API documentation, and testing playgrounds. This makes the platform open for modular innovation while remaining secure and controlled.

## 🎯 Goals

Enable external integrations with WooCommerce, Shopify, CRM tools, or custom dashboards

Empower agencies to build their own branded extensions

Support deeper automation via events and API triggers

Maintain security with scoped API keys and audit logs

## 📦 Toolkit Contents

### 1. REST API Gateway

Secure, versioned API base (e.g., https://api.lavishta.ai/v1/)

OAuth2 or token-based access

Supports CRUD actions on key modules:

Campaigns

Blogs

Inbox Replies

Customers

Orders (read-only)

UGC Points + Leaderboard

Product Bundles

Loyalty Rewards

#### Example Endpoints:

GET /campaigns?brand\_id=123

POST /ugc-replies

GET /customers/{id}/profile

### 2. Webhook Manager

Define and register custom Webhook endpoints per brand

Events supported:

campaign.created

ugc.submitted

order.placed

points.redeemed

customer.linked

Retry with exponential backoff

Signature verification + timestamp check

### 3. OAuth App Platform (Multitenant)

Brands or agencies can create/manage OAuth apps

Scopes:

ugc:read, campaign:write, order:read, etc.

Per-brand app keys and secrets

Admin approval required to publish

Throttling & quota control via superadmin

### 4. Interactive Docs + Playground

Swagger UI with live API tryout per brand

Auto-updated OpenAPI spec

Code snippets: Python, JS (Axios), PHP, cURL

Sample request builders

### 5. Plugin SDK + Quickstarts

NPM package for frontend widgets (e.g., leaderboard)

Starter templates:

Custom inbox bot

UGC display on brand sites

Reward points tracker embed

WordPress plugin skeleton (for Woo/Shopify hybrid use)

## 🔐 Security

Scoped tokens with TTL

Refresh tokens + revoke mechanism

IP whitelisting (optional)

Logged per API key usage

Alert on unusual patterns or abuse

## 🧠 Superadmin Controls

Set per-brand API limits (daily, hourly)

Approve/reject webhook registrations

View logs by:

IP

Brand

Endpoint

Error type

## 💡 Use Cases

Brand devs auto-publish campaign data to external dashboards

Resellers sync UGC points to CRM

Shopify brand auto-issues coupons on ugc.submitted

Affiliate tools track shares/clicks via webhook

✅ Document complete. Let me know if you’d like to export, illustrate the API Explorer UI, or generate SDK scaffolding next.

## Internal Templates EditorInternal Templates Editor

# 🛠️ Internal Templates Editor (Superadmin-Only)

## ✅ Module Overview

The Internal Templates Editor is a Superadmin-only module that allows platform administrators to define, update, and deploy prompt, content, and campaign templates across the SaaS ecosystem. It acts as a centralized control center for all reusable assets used in:

Auto-replies (Inbox, Comments, DMs)

Blog intros and CTAs

Campaign prompts (Comment prompts, post ideas)

Product description formats

Reel/Shorts script skeletons

UGC prompts & reward announcements

This module ensures quality, compliance, and tone consistency across hundreds of client brands while allowing brand-level overrides.

## 🎯 Goals

Let Superadmins manage and deploy prompt/content templates globally

Ensure brand-safe, optimized, and up-to-date templates

Allow testing and performance logging for A/B versions

Enable template targeting by industry, audience, or module

## 🧠 Features Breakdown

### 1. Template Library (Global & Scoped)

Templates organized by type:

Inbox Reply Templates

Blog Templates

Campaign Post Captions

Gamified Prompt Scripts

Offer Announcements

Loyalty Point Messages

Each template includes:

Template Name

Category (e.g., “Reels CTA”, “Delivery FAQ”)

Audience Tag (optional)

AI Prompt Body

Variables used (e.g., {brand\_name}, {discount\_value})

Preview Output

Status (Active/Draft/Deprecated)

### 2. Versioning + A/B Management

Create multiple versions of a template

Add test cases and compare outputs

Set A/B % for brands using this template

Logs engagement metrics (CTR, thumbs up/down, etc.)

### 3. Targeting Logic

Template targeting by:

Module (Inbox, Blog, Campaign, etc.)

Industry vertical (e.g., Skincare, Apparel)

Brand size (Small, Medium, Large)

Audience segment (e.g., Gen Z, Moms)

Locale/language

### 4. Lock & Push System

Superadmin can lock templates from being overridden at brand level

Or push updates with override options for brand admins

Push alerts sent to affected brand admins with changelog

### 5. Template Editor Interface

Live Prompt Editor with syntax highlighting

Variable suggestion engine

Real-time output preview

Industry best practices panel

Testing input form (simulate brand profile)

## 🔧 Backend Tables

templates

template\_id, name, type, category, audience, prompt, variables, version, status, created\_by, updated\_at

template\_versions

version\_id, template\_id, prompt\_body, test\_cases, output\_samples, engagement\_metrics, live\_percent

template\_logs

log\_id, brand\_id, template\_id, version\_used, module\_used, timestamp, result\_rating

## 🔐 Roles & Permissions

Superadmin:

Full template management

Push/update/lock templates

View usage metrics

Brand Admins:

View global templates

Customize only if not locked

Assistants/Editors:

Suggest edits (approval required)

## 🔄 Integrations

AI Blog Generator: uses blog intro/CTA templates

Inbox AI: uses predefined reply types

Campaign Engine: pulls from campaign CTA/post libraries

Auto Poster: fetches announcement styles from this repo

Rewards Engine: sends messages using template hooks

## 📊 Metrics Tracked

Template engagement performance (by version)

Usage frequency per brand/module

Override frequency

Template freshness (last updated)

## 🧠 Example Prompt Template

Name: Delivery FAQ - Bangla Casual Module: Inbox Reply Prompt:

Write a short Bangla reply to a customer's DM asking about delivery time. The tone should be friendly and warm. Limit to 70 words.

Variables: none Version: v1 Output Preview: “Apnar delivery usually 2-4 din er moddhe delivery hoy. Dhonnobad apnar order er jonno! 💕”

✅ Document complete. Would you like to export this module or move to the next one?

# Planning

## Finalized Modules OverviewFinalized Modules Overview

Finalized Module Documents – AI-Powered SaaS for Beauty Commerce

### ✅ 1. Master Overview

Document: SaaS Platform Build Documentation for AI-Powered Beauty Commerce Suite (Lavishta) - General overview - System philosophy (modular, brand-first, independent modules) - Multitenant support - Full list of modules and architecture

### ✅ 2. Master AI Product Brain

Document: AI Product Brain Module - Embeddings & vector DB - Fuzzy matching - Tag normalizer - Internal linking engine - Related products & trend linker

### ✅ 3. Blog Engine

Document: Blog Engine Module - UGC-powered blog generation - Google Trends blog generator - Blog refresher - Internal linking - Category mapping

### ✅ 4. Gamification Engine

Document: Gamification Engine Module - Campaign builder (auto, manual, assisted) - Prompt variant generator - UGC tracker - Points & Tiers - Reward dispatcher - Auto-close & leaderboard - Templates Marketplace - Persona-based suggestions - Wall of Fame - Feedback campaigns - Admin-level moderation

### ✅ 5. Auto Poster / Post Manager

Document: Auto Poster Module - Post generator (Auto/Manual) - Caption + link + AI templates - Canva auto-design integration - Scheduler - Analytics - Calendar preview - Image generation add-on (planned)

### ✅ 6. Inbox Intelligence Engine

Document: Inbox Intelligence Engine Module - Comment/inbox scraping - UGC labeling - Customer mapping - GPT-powered suggested replies - Bot-to-human handoff - Message priority & tagging - FB/IG/WhatsApp integration

Pending Next: - Customer Upload + Identity Mapping - Email + SMS Campaign Builder - AI Blog Refresher + Meta Title Optimizer - Reels / Shorts Auto-Suggester - Points + Loyalty Tier Engine - Multilingual + Localization System - Advanced Admin Dashboard - GDPR Compliance + API Usage Controls - Role-based Access & Abuse Prevention - Platform-Level App Integrations - Internal Templates Editor (Superadmin) - AI FAQ Builder - Reseller Panel + Tiered Pricing

## Future - SaaS Feature SuggestionsFuture - SaaS Feature Suggestions

# 🧠 Final Suggestions Checklist for SaaS Platform (Lavishta AI Commerce Suite)

A master checklist of future-ready suggestions and enhancements to complete the vision of a fully modular, intelligent, scalable SaaS system.

## ✅ 1. Performance & Scalability Enhancements

Edge Caching & CDN Rules per Brand

Tenant-based Rate Limiting / Throttling

Auto-scaling AI Workers and Queues

Per-region Latency Monitoring System

## ✅ 2. Compliance & Security Layers

Two-Factor Authentication (2FA) for Admin/Brand Logins

Audit Trail Logging (All User Actions / Settings Changes)

Data Retention Auto-Policies (Purge After X Days)

Customizable Terms & Privacy Template per Brand

“Export My Data” + Consent Logs (GDPR / CCPA)

## ✅ 3. Developer & Plugin Ecosystem

Plugin Compatibility Checker

Developer Sandbox Mode

Plugin Marketplace Interface + Toggle Control

Plugin Rating & Moderation System

## ✅ 4. AI Evolution Support

Model Version Switcher (e.g. GPT-4, Claude)

Brand-based Fine-tuning Settings Panel

Prompt Library Debugging & Testing Suite

AI Usage Efficiency Tracker (Token ROI)

## ✅ 5. Reseller & White-Label Enhancements

Sub-Brand Control Panel (for Agencies)

White-Labeled Onboarding Materials (PDF, PPT, Video)

Brand Setup Templates (Auto Campaigns, Blogs, Tags)

Reseller Analytics Dashboard (usage, issues, earnings)

## ✅ 6. App & API Resilience

Token Expiry Warnings (Meta, Shopify, etc.)

Auto Token Rotation Layer (per integration)

Webhook Retry Queue + Logging Panel

## ✅ 7. UX Enhancements for Brand Admins

AI Setup Wizard V2 (Industry-based config generator)

Contextual Tooltip + Video Demos inside Admin UI

Dynamic Font/Color Style Injections for Posts

## ✅ 8. Platform Intelligence Engine (Optional)

Brand Health KPI Score

Weekly AI Summary Generator for Brands (Wins, Gaps)

Predictive Stock / Campaign Insights (Trending vs Stock)

This document will be used as a planning guide to implement these enhancements as future modules. Each suggestion can be converted into its own module documentation if prioritized.

# Mobile

## Auto Poster ModuleAuto Poster Module

AUTO POSTER / POST MANAGER MODULE

## 🗓️ Overview

This module handles the automatic creation, scheduling, design, and posting of marketing content to Facebook and Instagram. It supports brand-specific modes (Auto, Manual, Assisted), AI-generated captions, and optional image/video generation via Canva or advanced render engines.

Each brand can manage their own post queue, customize tone and interval settings, and track performance via analytics. Posts can be synced with blogs, products, campaigns, and seasonal themes.

## 📘 Features

### 1. Post Creation Engine

AI-Generated Posts

Powered by OpenAI (or fine-tuned LLM).

Input: Brand voice, product, blog topic, campaign name.

Output: Caption + Hashtags + Emojis (optional tone: fun, premium, expert).

Modes:

Auto: Fully AI-generated + posted

Assisted: AI drafts, human approval required

Manual: Admin writes and schedules

Prompt Examples:

"Write a fun Instagram caption for a post featuring Lavishta's new Vitamin C serum, highlighting glow benefits, use emojis."

### 2. Canva-Based Image Generator (Optional Add-on)

Templates:

Product Highlights

Offer Cards

UGC Highlights

Seasonal Campaigns

Settings:

Brand color, logo, font from Brand Profile

Product photo auto-insert from Woo/Shopify media

Auto Mode:

System matches template + product

AI adds text overlays (title + benefit + CTA)

Output:

High-quality PNG image auto-attached to scheduled post

### 3. Scheduler

Interface:

Calendar-based view

List of upcoming posts

Filters: Brand, campaign, product, status (scheduled, published)

Config:

Days/times for each brand

Frequency control (e.g. max 1 post/day)

### 4. Post Manager Panel

Draft, Edit, Duplicate, Pause, Delete posts

Approve Assisted Posts

Track:

Published time

Post ID (Meta)

Clicks, reactions, comments (via Graph API)

### 5. Link Manager

Add blog links, product links

UTM tracking code generator

Shortlink generator (e.g. lav.link/xxx)

Click tracking via internal redirect

## 🧱 Tech Stack & APIs

Backend: Node.js or Python FastAPI

Scheduler: Cron + Redis Queue

Meta API: Facebook Graph API for Pages and Instagram Business accounts

Frontend: React + Tailwind

Image Engine: Canva API (or Puppeteer + template HTML rendering)

DB: PostgreSQL for posts, metadata, schedules

## ⚖️ Superadmin Controls

Enable/Disable Auto Poster per brand

Canva API usage tracking

AI Token usage stats

Approve template access per tier

Logs: failed posts, rejected creatives

## 🌈 Web UI - Brand Panel

Settings Tab:

Default Tone: [Fun, Premium, Expert, Soft, Gen Z, Neutral]

Frequency per week

Time slots per day

CTA types preferred (Shop Now, Learn More, DM Us)

Canva Tab:

Select templates allowed

Upload brand assets

## ✅ Example Use Case

Brand sets mode = Auto, tone = Expert

Every 2 days, system picks a product with low CTR

Writes educational + emotional post

Uses Canva template to render PNG

Posts automatically to FB Page and IG

Link clicks tracked and shown in analytics

Next: Inbox Reply Engine + UGC Inbox Mining

## Lavishta\_Auto\_PosterLavishta\_Auto\_Poster

## Mobile Admin App LiteMobile Admin App Lite

# 📱 Mobile Admin App (Lite Version)

## 🎯 Objective

To build a lightweight mobile admin application for Superadmins and Brand Owners to monitor and manage the SaaS operations on-the-go, covering critical metrics, notifications, basic user controls, and instant alerts.

This is a companion app to the full web dashboard and includes only the most necessary functions for real-time insight and urgent action.

## 🧩 Feature Scope

### 1. Authentication & Access

Secure Login via OAuth (linked to main SaaS account)

2FA (Two-Factor Authentication) via OTP/Authenticator App

Role-aware navigation (Superadmin, Brand Admin, Support Agent)

### 2. Real-Time Metrics Dashboard

Quick snapshot cards:

🔄 Total Active Brands

👥 Active Users Today

💸 Revenue This Month

⚠️ Unresolved Incidents

📦 Pending Campaigns/Posts

Tap into detailed views for each metric

### 3. Alert Center (Priority Notifications)

🔔 Live alerts for:

System incidents (from Incident Logs Module)

Subscription issues

Brand complaints

Failed integrations (e.g. FB/IG disconnected)

Toggle settings for:

Alert sound

Push notifications

Quiet hours

### 4. Tenant Quick Actions

View tenant list with filters (active/inactive/suspended)

Tap to:

Pause subscription

Extend trial

Manually retry failed syncs (campaigns/posts/orders)

View brand’s incident log snapshot

### 5. Campaigns & Inbox Summary

🔄 View most recent campaigns per tenant

📥 Active DMs/Inbox Load status (summary)

📊 UGC performance cards

Mark urgent brands for follow-up

### 6. Reports Snapshot

Last 7/30-day summary:

Revenue & active user graph

# Campaigns run

Email/SMS delivery rate (summary only)

Incident trend line

### 7. Superadmin Tools (Lite)

Trigger user-wide announcements

Send Slack/Webhook alert

View system health (OK / Warning / Down)

Manually trigger background job (sync, reindex, report)

## 🛠️ Technical Details

Built using React Native (cross-platform)

Authentication via SaaS central API + Firebase token

Push via Firebase Cloud Messaging (FCM)

Light/dark theme toggle

Offline caching of last known metrics

## 📱 Future Features

Full-screen chart views

Approval queue (campaign/posts)

In-app voice note recording for support notes

AI suggestions on next actions (e.g., which tenant needs attention)

This mobile app is designed to empower founders, superadmins, and brand stakeholders with fast decision-making tools while away from the web dashboard. All modules are read+basic action only (no editing or publishing workflows).