# 🚨 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.