# 🔒 Role-Based Access Control (RBAC) + 🛑 Abuse Prevention System

## ✅ Module Overview

This module implements a secure, scalable, and flexible access control mechanism and abuse moderation system. It ensures that: - Teams within each brand have appropriate access scopes - Sensitive features are guarded by role rules - Public user-generated content (UGC) is moderated for abuse, spam, or policy violations

## 🎯 Goals

* Protect brand and customer data
* Provide granular user permissions per module
* Prevent misuse of public campaigns
* Ensure platform compliance with moderation policies

## 🔐 Role-Based Access Control (RBAC)

### Supported Roles (per Brand):

1. **Superadmin** (SaaS Owner Only)
   * Full access to all tenants, modules, and internal logs
   * Can impersonate brand admins
2. **Brand Owner / Admin**
   * Full control over their brand’s modules, users, and usage
3. **Campaign Manager**
   * Can create/edit posts, campaigns, and gamification flows
4. **Customer Support**
   * Can manage inbox replies, customer identity mapping
5. **Editor / Analyst**
   * Can view reports, suggest prompt edits, view UGC data
6. **Read-Only**
   * Only allowed to access reports or monitor dashboards

### Role Matrix Example:

| Feature | Superadmin | Admin | Campaign Manager | Support | Analyst | Read-Only |
| --- | --- | --- | --- | --- | --- | --- |
| Access all modules | ✅ | ✅ | 🚫 | 🚫 | 🚫 | 🚫 |
| Edit Prompt Templates | ✅ | ✅ | 🚫 | 🚫 | ✅ | 🚫 |
| Manage Inbox Replies | ✅ | ✅ | 🚫 | ✅ | 🚫 | 🚫 |
| Run Setup Wizard | ✅ | ✅ | 🚫 | 🚫 | 🚫 | 🚫 |
| View Leaderboards | ✅ | ✅ | ✅ | ✅ | ✅ | ✅ |
| Approve Campaigns | ✅ | ✅ | ✅ | 🚫 | 🚫 | 🚫 |

### UI:

* Role editor in brand panel
* Assign users + permissions
* Add/edit/delete users (by Admin)

## 🛑 Abuse Prevention System

### Purpose:

To detect, flag, and moderate inappropriate UGC and comment-based campaign submissions.

### 🧠 Features

* **Keyword Filtering Engine**
  + Maintain global + brand-specific blacklists (e.g., offensive, spammy terms)
  + Real-time check during comment ingestion
* **Language Detection + Toxicity Scoring**
  + Use NLP-based toxicity detection (e.g., perspective API or in-house LLM)
  + Score entries from 0–1 based on toxicity
* **Auto-Moderation Triggers**
  + Delete or hide posts above certain threshold
  + Notify brand admins when multiple abusive comments detected from a user
* **Manual Review Queue**
  + UGCs flagged go to review panel
  + Roles with moderation permission (Admin/Support) can approve/reject
* **Rate Limiting**
  + Prevent spam by applying comment rate limits
  + Example: Max 5 comments/minute per user ID/IP

### 🚨 Alerts & Logs

* UGC Flagged: Logged with timestamp, user info, content snapshot
* Admin Dashboard: Heatmap of toxic comments by campaign
* Exportable CSV reports

### Moderation Settings (Brand Panel):

* ✅ Enable/Disable Auto-Moderation
* Set thresholds for Toxicity Score (e.g., auto-hide if > 0.7)
* Blacklist Management UI (add terms/phrases)
* Enable Language Restrictions (e.g., only Bangla/English allowed)

## 🔧 Backend Requirements

### Tables:

* user\_roles → user\_id, brand\_id, role, assigned\_by
* ugc\_flags → ugc\_id, flag\_type, score, timestamp, resolved\_by
* blacklist\_terms → term, brand\_id, source

### Services:

* Role Management Service (token-based)
* UGC Ingestion Engine (with toxicity filter)
* Moderation Dashboard API
* Admin Alerts Service

## 📊 Metrics to Track

* UGC flagged per campaign
* Role assignments per brand
* Approval turnaround time for flagged UGC
* Total comments hidden/auto-deleted

## 🌐 Integrations

* Inbox module: Comment spam flagged in real time
* Leaderboard: Exclude flagged comments from scoring
* Notification Center: Alerts to moderators

## ✅ Outcomes

* Brands feel safer running open campaigns
* Admins gain visibility + control over user actions
* Trust score per brand for public UGC moderation
* Reduced support load by automatic filtering

✅ Let me know if you’d like to visualize this as a dashboard next or proceed to the next pending module.