# 🔍 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?