# SmartChain – Feature & Technology Blueprint

Gradio UI • GPT-only intelligence • lightweight, no deployment or shared-services details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Feature | What it does | How it works (GPT‑centric) | Core tech stack |
| 1 | Predictive Stocker & Order Manager | Produces purchase‑order proposals that hit service‑level and budget targets while avoiding stock‑outs. | 1. Pull recent sales, on‑hand, and lead‑time rows from PostgreSQL. 2. Send a structured prompt to GPT‑4o (“Here’s 90 days of sales for SKU 123, current stock = 150, lead‑time = 7 days—recommend reorder qty”). 3. Parse GPT’s JSON reply; surface it in Gradio and via /predict‑restock endpoint. | • Data store: PostgreSQL (+ TimescaleDB optional) • LLM: OpenAI GPT‑4o via Python SDK • API: FastAPI |
| 2 | AI Markdown Optimizer | Suggests the right discount (and optional extra stock) to clear inventory with minimal margin loss. | 1. Use Serper API to fetch live competitor prices for a SKU. 2. Combine those prices with current stock/price in a GPT‑4o prompt (“Our price = ₹999, comps = ₹949/₹979/₹999—suggest discount % and expected sell‑through”). 3. Return GPT’s JSON (discount %, units expected, reasoning) to Gradio. | • Search: Serper API • Data: Pandas • LLM: GPT‑4o • API: FastAPI |
| 3 | Dead‑Stock Liquidator & Strategy Emailer | Detects stale SKUs and emails a straightforward action plan. | 1. Scheduler script (or manual Gradio button) queries PostgreSQL for items with days‑on‑hand > 60 or sell‑through < 20 %. 2. Feed the SKU list to GPT‑4o (“For each item, pick flash‑sale / bundle / donate and write an email body”). 3. Send GPT’s drafted email through SendGrid API. | • Scheduler: simple Python script or Celery beat • Email: SendGrid • LLM: GPT‑4o |
| 4 | SmartChain Copilot (Gradio Front‑End) | Chat interface where staff ask inventory questions and receive GPT‑generated recommendations with one‑click “Apply”. | 1. Build a Gradio Chatbot + supporting widgets (tables, plots). 2. A LangChain/LangGraph Router Agent interprets each user query, decides which internal endpoint(s) to call, or asks GPT‑4o directly. 3. Consolidate JSON outputs into plain‑language answers and visuals in Gradio. | • UI: Gradio • Orchestration: LangChain RouterAgent / LangGraph • Optional cache: Redis |

## Flow Summary

1. Internal data & competitor prices are fetched.  
2. GPT‑4o reasons over that data to generate restock, discount, or liquidation guidance.  
3. Gradio Copilot displays the recommendations and allows users to trigger the corresponding actions or emails—all in a single, lightweight stack.