Technical Challenge: Salvit Client Configuration Console

Context

Salvit Advisors is evolving its operations by transitioning from fragmented, spreadsheet-based workflows to a centralized, web-based **SKU Management Platform**. As we partner with multiple e-commerce businesses of various types, we need a robust, dynamic interface that allows us to manage product data and financial logic efficiently across different brands.

The Goal

Build a **Dynamic SKU Management Interface** that allows users to switch between different e-commerce companies and manage their product catalogs. The core of this task is to create an interface where Salvit Advisors users can edit product metadata and define "Effective COGS" (Cost of Goods Sold) using a hierarchical resolution engine.

1. Requirements & Business Logic

A. Multi-Tenant Interface

- **Company Switcher:** Implement a way to toggle between different e-commerce companies (e.g., a dropdown). Changing the company should update the entire dataset displayed in the dashboard.
- **Dynamic Table:** An interactive table displaying the list of products for the selected company.

B. SKU Integrity & Editable Fields

- **SKU as the Anchor:** The **SKU is the unique identifier and is strictly non-editable**. It serves as the base ID for all logic.
- **Editable Metadata:** Users must be able to edit the **Title** and **Category** of any given SKU.
- **Cost Management:** Users must be able to define and edit costs at three different levels:
    - **SKU Level** (Highest priority/Most specific).
    - **Title Level**.
    - **Category Level** (Lowest priority/General).

- **Date Sensitivity:** Each cost rule must include an optional `startDate` and `endDate`. A rule with dates that encompass "Today" always overrides a "General" rule (one without dates) at the same level.

C. User Experience

- **Status Badges:** For each SKU, show a status (e.g., "Resolved", "Override", or "Missing Cost").
- **Detail Drawer:** Clicking a SKU should open a drawer to edit data (Title, Category) and manage the cost rules history for that specific item.
- **Persistence:** For the purpose of this test, all changes should be **saved locally (State/LocalStorage) or simulated**. No external database connection is required.

2. Tech Stack & Tools

- **Framework:** React.js
- **State Management:** Redux (Strictly Required).
- **Styling:** Tailwind CSS (Following the provided Salvit Brand System).
- **Leverage AI:** You are **expected** to use AI tools (Cursor, Copilot, Codex, etc.) to accelerate your workflow. We are evaluating your ability to prompt, architect, and validate AI-generated code.

3. Styling Guidelines

Please use the following brand constants in your Tailwind implementation:

- **Primary Blue:** `#002D72` (`brand-blue`) — Use for headers and authority.
- **CTA Orange:** `#F47C1F` (`brand-orange`) — Use for primary actions.
- **Accent Indigo:** `#4f46e5` (`brand-indigo`) — Use for data highlights.
- **Fonts:** `Barlow` for headings, `Inter` for UI/Body.