Feature Context: Multi-Select Checkbox Filters

Version: 1.0

1. Feature Description

The goal is to implement multi-select checkbox filters for the families and tags data fields. This feature will allow the user to combine multiple criteria to perform powerful, granular searches of the plugin vault.

2. User Story

As a user, I want to be able to select multiple categories (like "compressor," "opto," and "emulation") simultaneously, so I can narrow down the list to find the exact type of tool I need for a specific production task.

3. Core Requirements

- **Dynamic Filter Generation:**
  - The application must automatically scan the entire vaultData on load.
  - It must create two unique, alphabetized lists: one for all available families and one for all available tags.
  - These unique lists must be used to dynamically generate two corresponding groups of checkboxes in the sidebar.
- **Filtering Logic:**
  - A state management system (e.g., a JavaScript object or Set) must be implemented to track which checkboxes are currently selected.
  - The filtering logic will be a pipeline:
    1. The existing live text search is applied first.
    2. The multi-select checkbox filters are then applied to the results of the text search.
  - The checkbox filtering must use **"AND" logic**. A plugin will

    only be displayed if it contains **all** of the selected families and **all** of the selected tags.

- **UI Interaction:**
    - Checking or unchecking any box must instantly trigger a re-filtering of the plugin grid.
    - The UI should provide clear visual feedback for the active filters.

4. Vibe & Aesthetic

- **Vibe:** "Tactile & Immediate". The interaction should feel like pressing buttons on a hardware controller. Each click should result in an instant and satisfying update to the main grid.
- **Analogy:** The filtering experience should be as intuitive and powerful as using the tag-based filtering system in a professional sample library manager (like XO or Atlas).
- **Styling:** The checkboxes and their labels will be styled to fit the existing "Classic Pearl" / "Subscotia Dark" themes, with clear focus states and a clean, organized layout.

5. Known Constraints

- This feature will complete the core filtering functionality for the application.