Feature Context: Add New Plugin Form

Version: 1.0

1. Feature Description

The goal is to design and build the user interface for an "Add New Plugin" form. This form will be presented to the user in a modal window and will contain input fields for all the necessary data points required to create a new entry in the vault, consistent with the schema in VAULT_GUIDE.md.

2. User Story

As a user, I want a simple and intuitive form within the web app to add a new plugin to my vault, so I no longer need to manually edit the vault_master.json file, which is slow and prone to errors.

3. Core Requirements

- **Access Point:** A clearly visible "Add New Plugin" button, likely featuring a "plus" icon, will be added to the main header.
- **Modal Presentation:** Clicking the "Add New Plugin" button will open the form in a modal window, similar to the existing detail view modal.
- **Input Fields:** The form must include fields for the following essential data points:
  - name (text input)
  - product (text input)
  - developer (text input, with autocomplete for existing developers)
  - type (dropdown/select menu limited to the official types: "instrument", "fx", "utility", "container", "expansion")
  - families (a "tag-style" input field that allows for adding multiple, distinct values)
  - tags (a "tag-style" input field, same as for families)

    - active (a checkbox, checked by default)
    - notes (a multi-line textarea)
- **Form Actions:** The modal must contain a "Save Plugin" button and a "Cancel" button.
- **Data Validation (Future Goal):** While not in the scope of this task, the form should be designed with future client-side validation in mind (e.g., marking the name field as required).

4. Vibe & Aesthetic

- **Vibe:** "Efficient & Guided". The form should guide the user through the data entry process, making it fast and reducing the cognitive load. It should feel like a purpose-built data entry tool.
- **Analogy:** The interaction should be as smooth as filling out a well-designed form in a professional application like Reaper's project settings or a modern web app.
- **Styling:** The form elements (inputs, labels, buttons) will be styled to perfectly match the established "Classic Pearl" / "Subscotia Dark" themes.

5. Known Constraints

- This task is focused **exclusively on the frontend UI**. The "Save Plugin" button will not be functional at this stage. The backend API endpoint and the logic to handle the form submission will be built in **Task 4.2**.