Product Requirements Prompt (PRP)

Feature: Full Responsiveness

Version: 1.0

Objective

To modify the index.html file to make the application layout fully responsive. This involves collapsing the two-column layout into a single column on mobile, adding a hamburger menu to toggle a slide-in sidebar, and ensuring the plugin grid adapts correctly to all screen sizes.

Execution Plan

Phase 1: HTML Structure Modification (index.html)

1. **Add Hamburger Button:**
    - Locate the <header> element.
    - Inside the header, before the <h1>, add a new <button> element with id="hamburger-button".
    - This button will contain a simple "hamburger" SVG icon.
    - Crucially, this button will be hidden on medium screens and up using Tailwind's responsive classes (md:hidden).
2. **Modify Main Layout Containers:**
    - The main content wrapper ( ) will remain largely the same, as Flexbox is inherently responsive.
    - The <aside> element for the sidebar will be modified to be hidden by default on mobile and appear as a slide-in overlay. This will be controlled by adding classes for positioning (fixed, inset-y-0, left-0), a transform for the slide-in effect, and a transition for the animation. Its visibility will be controlled by a new JavaScript-toggled class.
    - The <main> content area will be modified to be full-width on mobile (w-full) and revert to its fractional width on medium

screens and up (md:w-3/4).

Phase 2: CSS Styling (index.html via Tailwind Classes)

1. **Apply Responsive Prefixes:**
    - Go through the existing layout elements (aside, main).
    - Use Tailwind's responsive prefixes (e.g., md:, lg:) to apply different styles at different breakpoints.
    - **Sidebar (<aside>):**
        - Will be hidden by default (-translate-x-full).
        - When an is-open class is applied by JavaScript, it will become visible (translate-x-0).
        - On medium screens and up, it will lose its fixed positioning and transform classes and revert to being part of the flex layout (md:relative, md:translate-x-0).
    - **Main Content (<main>):**
        - Will be full-width by default (w-full).
        - On medium screens and up, it will become three-quarters width (md:w-3/4).
    - **Hamburger Button:**
        - Will be visible by default (block).
        - Will be hidden on medium screens and up (md:hidden).

Phase 3: JavaScript Logic (index.html)

A new script block will be added to manage the mobile sidebar's state.

1. **Define Elements:** Get references to the hamburger-button and the <aside> sidebar element.
2. **Event Listener:**
    - Attach a click event listener to the hamburger-button.
    - **On Click:**
        - Toggle a specific class (e.g., is-open) on the <aside>

element. This class will trigger the CSS transition to make the sidebar slide into view.
    - (Optional but recommended) Add logic to close the sidebar if the user clicks outside of it on the main content area.

Final Review

- Confirm the layout collapses to a single column on screen widths below 768px.
- Verify the hamburger button appears only on mobile and correctly toggles the sidebar's visibility.
- Test the sidebar's slide-in/slide-out animation.
- Confirm the layout reverts to the two-column view on screens wider than 768px.
- Ensure the plugin grid columns adjust correctly at all breakpoints as already configured.

Approval Request: The Product Director is requested to review this PRP. Upon approval, the Code Engine will proceed with the execution phase.