**Plan: Implementing the Languages Feature (Revised)**

This document outlines the three-phase plan to develop and integrate a comprehensive languages section into the NPC Engineer application, incorporating user feedback for a modal-based management system.

**General Changes**

* **Metadata:** The application version in npc engineer.html will be updated to 0.05.00 and the date will be set to the current date (2025-10-14).

**Phase 1: Data Model & Core Logic**

This initial phase focuses on updating the backend script (script.js) to support the new data structures and logic, without yet building the user interface.

1. **NPC Object Enhancement:** The core NPC object structure will be expanded. The defaultNPC object will be updated to include four new properties:
   * selectedLanguages (Array): This will hold a list of strings, where each string is a chosen language. *Default: [] (an empty array).*
   * specialLanguageOption (Number): An integer to represent the chosen special language option for future features. *Default: 0.*
   * hasTelepathy (Boolean): A flag to determine if the NPC has telepathy. *Default: false.*
   * telepathyRange (Number): A value for the telepathic range in feet. *Default: 0.*
2. **Language Data Structure:**
   * New constant arrays will be defined at the top of script.js to hold the predefined languages:
     + standardLanguages: "Common", "Dwarvish", "Elvish", "Giant", "Gnomish", "Goblin", "Halfling", "Orc", "Thieves' cant"
     + exoticLanguages: "Abyssal", "Celestial", "Draconic", "Deep speech", "Infernal", "Primordial", "Sylvan", "Druidic", "Undercommon"
     + monstrousLanguages1: "Aarakocra", "Aquan", "Auran", "Bullywug", "Gith", "Gnoll", "Grell", "Grung", "Hook horror", "Ice toad", "Ignan", "Ixitxachitl"
     + monstrousLanguages2: "Modron", "Otyugh", "Sahuagin", "Slaad", "Sphinx", "Terran", "Thri-kreen", "Tlincalli", "Troglodyte", "Umber hulk", "Vegepygmy", "Yeti"
   * A new property, userDefinedLanguages, will be added to the bestiary.metadata object. *Default: [].*
3. **Core Function Updates:**
   * **healBestiary():** This function will be updated to initialize the four new language-related properties on any legacy NPC objects when a bestiary is loaded. It will also initialize bestiary.metadata.userDefinedLanguages if it doesn't exist.
   * **updateActiveNPCFromForm():** Logic will be added to read selected languages from the five listboxes and populate the activeNPC.selectedLanguages array.
   * **updateFormFromActiveNPC():** Logic will be added to populate the listboxes and correctly highlight the languages stored in activeNPC.selectedLanguages when an NPC is loaded.
   * **updateViewport():** This function will call a new helper function, calculateLanguagesString(), to generate the final "Languages" string for the statblock. This new function will:
     + Return an empty string if no languages are selected and telepathy is false.
     + Alphabetically sort the activeNPC.selectedLanguages array.
     + Join the array into a comma-separated string.
     + Check if hasTelepathy is true and specialLanguageOption is 0. If so, it will append ", telepathy [range] ft." to the string.
     + Include a switch statement to handle 6 placeholder options for specialLanguageOption, which will do nothing for now.

**Phase 2: User Interface Implementation**

This phase involves building the visual components within the HTML and styling them with CSS.

1. **HTML Structure (npc engineer.html):**
   * The 'Languages' card header will be updated to include a "Manage Languages" button on the far right, using the same SVG icon as the "Manage Groups" button.
   * The card body will contain a flex container arranging five <divs> horizontally. Each div will contain a <label> (e.g., "Standard") and a <select multiple> listbox. All five listboxes will appear identical.
   * A new modal, manage-languages-modal, will be added. It will be a close copy of the manage-groups-modal, containing an input field, an "Add" button, and a list area for displaying and deleting custom languages.
2. **CSS Styling (style.css):**
   * A new class, .language-listbox, will be created, inheriting styles from .info-input for a consistent look. Its height will be set to show 12 items.
   * A CSS rule for .language-listbox option:checked will be added to style selected items with a desaturated purple background (#8A8AA3) and white text (#FFFFFF).

**Phase 3: Interactivity & Finalization**

The final phase connects the UI from Phase 2 with the logic from Phase 1, making the feature fully interactive.

1. **Event Handling (script.js):**
   * input event listeners will be attached to all five <select> listboxes to trigger updateActiveNPCFromForm() on change.
   * A click event listener on the new "Manage Languages" button will show the manage-languages-modal.
   * Event listeners for the "Add" and "Delete" buttons within the modal will be created.
2. **User Language Management (Modal Logic):**
   * A new showManageLanguagesModal() function will populate the modal with the current list of user-defined languages from activeBestiary.metadata.userDefinedLanguages.
   * A new addNewLanguage() function will be triggered by the modal's "Add" button. It will:
     1. Retrieve the text from the modal's input field.
     2. **Validation:** It will perform a case-insensitive check to ensure the new language does not already exist in any of the four predefined language arrays OR the userDefinedLanguages array.
     3. If valid, it adds the language to activeBestiary.metadata.userDefinedLanguages, saves the bestiary, and refreshes both the modal list and the main "User" listbox on the form.
   * A new deleteLanguage() function will remove the specified language from the metadata, save, and refresh the UI. It will also de-select the language for any NPCs that were using it.
3. **Final Review:**
   * A full review of the feature will be conducted to ensure all parts are working as expected.