genAI prototyping

Wildsea Character Sheet - Requirements Document

1. Purpose & Context

A digital character sheet for the Wildsea TTRPG that supports character creation, gameplay tracking, and character advancement across sessions.

2. Core Features (Mode-Independent)

- **Character Identity**: Name, Bloodline, Origin, Post
- **Aspects**: Selectable traits/gear/companions with damage tracking
- **Edges**: 7 types of character strengths (select 3)
- **Skills & Languages**: Point-based progression system
- Resources: Tracking items in 4 groups: Charts, Salvage, Specimens, and Whispers

3. Mode Structure

The app operates in three distinct modes with different purposes:

3.1 Creation Mode

Purpose: Build a new character from scratch

Features:

- Character naming
- Random character generator
- Bloodline/Origin/Post selection (dropdowns)
- Aspect selection: Choose 4 from available options across three categories
- Edge selection: Choose 3 from 7 options
- Skills & Languages: Allocate 8 points (max rank 2 during creation)
  - Low Sour language starts at rank 3 (free)

- Establish Drives and Mires
- Add starting Resources

Validation Rules:

- Maximum 4 aspects
- Maximum 3 edges
- Maximum 8 skill/language points
- Cannot reduce Low Sour below rank 3

3.2 Play Mode

Purpose: Track character state during active gameplay

Features:

- Read-only character info display
- Aspect damage tracking: Click to cycle through states (default → marked → burned → default)
- Display selected edges, skills, languages (read-only)
- Manage changes to Drive and Mire text, and mire status
- Manage milestones
- Manage resources

Visual Emphasis:

- Compact vertical spacing for maximum on-screen content
- Clear visual states for aspect damage (red for marked, dark red for burned)

3.3 Advancement Mode

Purpose: Improve character between sessions

Features:

- Add new aspects (no limit)
- Expand aspect tracks: +/- buttons on selected aspects (max track size: 5)

- Train skills & languages: Up to rank 3 (no point limit)
- Manage milestones

Visual Indicators:

- New track boxes shown in bright green
- Original track boxes remain standard styling

4. UI/UX Requirements

Typography:

- Headings/labels: Faustina (Google Fonts)
- Body/content: System default

Color Palette:

- Primary: Black (`#000000`)
- Background: White (`#FFFFFF`)
- Borders/dividers: Gray-300
- Selected states: Black background, white text
- Aspect damage: Red-500 (marked), Red-900 (burned)
- New tracks: Lime-500

Layout Patterns:

- Three-column grid for aspect selection
- Consistent header structure across modes

Interaction Patterns:

- Dropdowns for constrained choices
- +/- buttons for numeric values
- Click-to-cycle for aspect damage states

5. Data Structure Requirements

Character Object:

- `name: string`
  `- bloodline: enum`
  `- origin: enum`
  `- post: enum`

Aspects:

- `aspectId: string (composite key: "Source-AspectName")`
  `- source: enum (Bloodline/Origin/Post)`
  `- type: enum (Trait/Gear/Companion)`
  `- name: string`
  `- baseTrack: number (default track size from data)`
  `- trackSize: number (current size: base + expansions, max 5)`
  `- damageStates: array of states per box (default/marked/burned)`
  `- selected: boolean`

Skills/Languages:

- `Map of name → rank (0-3)`
  `- Special case: Low Sour minimum rank 3`