Project Plan Addendum: Vibe Coding Development Framework

Date: 20 July 2025

This document outlines the practical application of the "Vibe Coding" methodology for the Subscotia Vault project. It serves as a guide for how the Product Director (Drew Campbell) and the Code Engine (Gemini 2.5 Pro) will collaborate to develop the skills essential for this modern, AI-assisted workflow.

1. Core Technical Skills Development

- **Programming Fundamentals:** While the Code Engine will generate the bulk of the code, it will also serve as a mentor. For any generated code block, the Product Director is encouraged to ask, "Why was it done this way?" The Code Engine will then provide detailed explanations of the underlying logic, data structures, and best practices, reinforcing fundamental knowledge.
- **Version Control (Git):** We will continue our established practice of frequent, logical commits. The use of feature branches will be encouraged for any experimental "vibe" changes, allowing for fearless iteration with the safety of being able to revert to a stable state.
- **Debugging and Troubleshooting:** The Bolt DIY playground will be our primary debugging environment. When a bug is identified, we will work together: the Product Director will describe the unexpected behavior, and the Code Engine will suggest specific code sections to inspect and provide targeted fixes.
- **Prompt Engineering:** This is the cornerstone of our S&I phase. We will move beyond simple instructions to craft rich, descriptive prompts.
    - **Practice:** We will use analogies from music production (e.g.,

"the card should fade in with the slow attack of a cello note") to translate abstract "vibes" into concrete parameters the Code Engine can interpret.

2. Effective Use of Tools and AI

- **AI Tooling Mastery (Bolt DIY):** Bolt DIY is our central workbench. Our goal is to master its features, including its ability to manage multi-file projects, its live preview capabilities, and its one-click export function for deployment.
- **Context Handling:** Before starting a new task in Bolt DIY, our S&I session will explicitly define the context. We will identify which existing files are relevant and what the core goal is, ensuring the prompts given to the Code Engine are focused and effective.

3. Collaboration and Soft Skills

- **Clear Communication:** Our S&I sessions are the forum for this. The goal is to develop a shared vocabulary that effectively communicates aesthetic and functional goals (e.g., "brutalist but clean," "snappy and responsive," "ethereal and soft").
- **Feedback and Iteration:** This is the core loop of vibe coding. The process will be: **Prompt -> Generate -> Test -> Feel -> Refine**. We will embrace this rapid cycle, understanding that the first version is a starting point for a conversation, not the final product.

4. Creativity and System Design

- **UI/UX Sensibility:** This is where the Product Director's creative intuition is paramount. The Code Engine will present functional components, and the Product Director will guide the "vibe" by providing feedback on spacing, typography, color, and animation until it *feels* right.

- **Big Picture Thinking:** The Product Director will maintain focus on the overall user story and the application's architectural integrity. The Code Engine will support this by asking clarifying questions like, "How does this new component fit into the overall layout?" or "Will this filtering logic scale if we add more data?"

By consciously applying this framework, we will not only build a high-quality application but also systematically develop the skills required for a successful, modern, and highly creative development partnership.