Document Version: 2.1

Date: 20 July 2025

Part 1: Project Governance & Work Plan

1.1. Core Philosophy & Roles

This project will be executed using a professional, AI-assisted development model. To ensure clarity and efficiency, project roles and responsibilities are defined as follows:

- **Product Director (Drew Campbell):** Defines the high-level goals, features, and desired outcomes for the application. The Product Director holds final authority on the project's direction and scope.
- **Project Manager & IDE (Bolt DIY):** A locally-run, open-source tool responsible for managing the development workflow. Its duties include orchestrating AI model interaction, scaffolding the multi-file project structure, and providing a local development environment for immediate testing and iteration.
- **Code Engine (Gemini 2.5 Pro):** The primary AI model responsible for all code generation. The Code Engine will produce high-quality, complete codebases based on structured requests and will assist with advanced refactoring, debugging, and strategic technical advice when prompted.

1.2. The Development Workflow

Each new feature or component will be developed using a structured, three-step process:

1. **Strategy & Ideation (S&I):** A collaborative session between the Product Director and the Code Engine to discuss feature requirements, brainstorm creative approaches, explore technical possibilities, and establish a definitive implementation plan.
2. **Execution:** The Product Director provides the high-level goal

from the S&I session to the Project Manager (Bolt DIY). The Project Manager then handles the prompt engineering and directs the Code Engine to generate the necessary code and file structure.

3. **Testing & Refinement:** The Product Director tests the generated feature within the Project Manager's local playground environment. Based on the results, the team can initiate a new S&I cycle to discuss and implement any necessary refinements or improvements.

1.3. Project Phases & Status

- **Phase 1: Backend Foundation & API**
  - **Status:** ✅ Complete.
- **Phase 2: Frontend UI/UX**
  - **Status:** 🟡 In Progress.
  - **Tasks:**
    - 2.1: Integrate Tailwind CSS & Inter Font. (✅ Complete)
    - 2.2: Build Main Application Layout. (Next Step, via Bolt DIY)
    - 2.3: Implement Plugin "Card" Layout.
    - 2.4: Implement Dark/Light Theme.
    - 2.5: Ensure Full Responsiveness.
- **Phase 3: Advanced Functionality & Interactivity**
  - **Status:** ⚪ Not Started.
- **Phase 4: Data Management Interface**
  - **Status:** ⚪ Not Started.
- **Phase 5: Production Deployment**
  - **Status:** ⚪ Not Started.

Part 2: Onboarding Guide: Bolt DIY

This section outlines the one-time setup required on the primary

development workstation (subwin).

2.1. Installation Procedure

Bolt DIY is a free and open-source application that runs locally. The primary dependencies are Node.js and the pnpm package manager.

1. **Install Node.js:** Download and install the latest LTS (Long-Term Support) version of Node.js from the official website: https://nodejs.org/.
2. **Install pnpm:** Node.js includes the npm package manager. Open a new PowerShell or Terminal window and execute the following command to install pnpm globally:
   npm install -g pnpm
3. **Install Bolt DIY:** Follow the official installation instructions provided on the Bolt DIY GitHub repository. This will typically involve cloning the repository and using a pnpm command to install its dependencies.

2.2. API Key Acquisition & Configuration

To enable Bolt DIY to utilize the Code Engine (Gemini 2.5 Pro), a Google AI API key is required.

- **Note on Service Costs:** It is important to note that developer API usage is provisioned on a pay-as-you-go model and is **not connected to any consumer-level subscriptions** (e.g., Google One). Google provides a substantial free tier for its developer API, which is generally sufficient for development-level usage. Costs will only be incurred if usage exceeds the limits of this free tier.
- **API Key Generation Steps:**
  1. Navigate to **Google AI Studio**: https://aistudio.google.com/
  2. Sign in with the appropriate Google account.

3. From the left-hand menu, select the **"Get API key"** option.
4. Select the option to **"Create API key in new project"**.
5. Copy the newly generated API key string to a secure location. This key must be treated as a confidential credential and should not be committed to version control or shared publicly.

2.3. Initial Application Configuration

Once installed, Bolt DIY must be configured to use the generated API key. Within the application's settings interface, select the appropriate AI model (Gemini 2.5 Pro) and paste the API key into the designated field to authenticate all subsequent requests to the Code Engine.