

## Subscotia Vault: Technology Stack Overview

This document provides an overview of the key technologies we will use to build the Vault web application. Each component has been chosen because it is powerful, modern, free, and well-suited to our goal of creating a professional, local-first studio tool.

1. Backend: Python 3 What It Is: The core programming language we will use to build the server-side logic of our application. We are already using it for all our data-cleaning scripts.

Its Role in Our Project: Python will be the "brain" of the operation. It will handle reading the vault.json file, processing data, and eventually, saving new plugin entries that you add via the web interface. It does all the heavy lifting in the background on your Dell server.

Why It Was Chosen: We are already proficient with it. It has a vast ecosystem of libraries and frameworks (like Flask) that make building web applications straightforward. It is the perfect foundation for everything we plan to do.

2. Web Framework: Flask What It Is: Flask is a "micro-framework" for Python. Think of it as a toolkit that provides the essential components needed to build a web server without forcing a lot of boilerplate or unnecessary features on us.

Its Role in Our Project: Flask is the engine that will power our local server. It will listen for requests from your web browser and execute Python code in response. When your browser asks for /api/plugins, Flask will run the Python function that reads your JSON file and sends the data back. It is the bridge between your browser (the frontend) and your Python code (the backend).

Why It Was Chosen: It is extremely lightweight, flexible, and easy to learn. For a local application like ours, which only needs a few specific API endpoints, Flask is perfect. It gives us all the power we need without the complexity of larger frameworks.

3. Frontend: HTML, CSS, and JavaScript What They Are: These are the three fundamental languages of the web.

HTML (HyperText Markup Language): Defines the structure and content of the web page (e.g., this is a heading, this is a paragraph, this is a button).

CSS (Cascading Style Sheets): Defines the presentation and styling of the web page (e.g., the heading is dark grey and uses the "Inter" font; the button has rounded corners and a blue background).

JavaScript: Defines the interactivity and behavior of the web page (e.g., when the user types in the search box, filter the list of plugins; when the user clicks a card, open a detail view).

Their Role in Our Project: They are the building blocks of what you will actually see and interact with in your browser. Our Python server will send the data, and this trio will be responsible for rendering it beautifully and making it interactive.

4. CSS Framework: Tailwind CSS What It Is: Tailwind is a modern "utility-first" CSS framework. Instead of giving you pre-built components like "card" or "button", it gives you a huge set of tiny, single-purpose "utility classes" that you can use to build completely custom designs directly in your HTML.

**Its Role in Our Project:** This is our primary tool for making the Vault look "tasty". We will use Tailwind's classes to control layout, colors, spacing, typography, and responsiveness. It will allow us to rapidly build the professional, custom interface we envisioned in the project plan, including the card-based layout and dark/light themes, without writing hundreds of lines of custom CSS code.

**Why It Was Chosen:** It offers maximum design flexibility while speeding up development time immensely. It encourages consistency and makes creating responsive designs that work on any screen size trivial.

5. Icons: Lucide Icons **What It Is:** A free, open-source library of over 800 simply designed and highly consistent icons.

**Its Role in Our Project:** We will use Lucide to add small, clear icons to buttons, filters, and other parts of the interface. For example, a magnifying glass icon for the search bar or a "plus" icon for the "Add New Plugin" button. These small details significantly improve usability and contribute to a professional look and feel.

**Why It Was Chosen:** The icons are lightweight, look great, and are very easy to implement. They are designed to be clear and readable, which is perfect for a data-focused application like ours.