Feature Context: Backend API for New Plugins

Version: 1.0

1. Feature Description

The goal is to create a new API endpoint in the Flask backend (app.py). This endpoint's sole purpose will be to receive new plugin data, submitted as a JSON payload from the "Add New Plugin" form on the frontend.

2. User Story

As a user, when I click the "Save" button on the "Add New Plugin" form, I want the data I have entered to be securely sent to the server for processing, so that it can be validated and prepared for saving into the main vault.

3. Core Requirements

- **New Route:** A new route, /api/plugin, must be created in app.py.
- **HTTP Method:** This route must be configured to specifically accept POST requests, as it is receiving data to create a new resource.
- **Data Handling:**
  - The endpoint must be able to receive a JSON payload in the request body.
  - It must parse the incoming JSON data into a Python dictionary.
- **Verification Step:** For this task, the endpoint's only action upon receiving data will be to print it to the Flask console. This allows us to verify that the data is being sent and received correctly before we implement the file-writing logic.
- **Response:** The endpoint must return a JSON response to the frontend to acknowledge receipt of the data (e.g., {"status": "success", "message": "Data received successfully"}).

4. Vibe & Aesthetic

- **Vibe:** "Robust & Secure". While not a visual component, the code for this endpoint should be clean, well-commented, and include basic error handling to ensure it can gracefully manage incoming requests.

5. Known Constraints

- This task **does not** involve writing the received data to the vault_master.json file. That critical step will be handled in the next task (**Task 4.3**).
- This task is focused **exclusively on the backend**. The frontend JavaScript will be updated to send the data to this new endpoint *after* the PRP for this task is approved and executed.