Here's a clean, step-by-step user guide you can keep pinned or printed for your **Vault Manager GUI**. Think of it as your one-click vault concierge.

🧰 Subscotia Vault Manager GUI — User Guide

Purpose: Add new instruments, FX, or utilities to your master vault quickly, safely, and beautifully.

📁 Files You Need

Place these scripts in the same project folder:

- vault_gui.py ← the GUI app
- generate_catalog_ids.py
- manage_untagged.py
- backup_vault.py
- build_html_catalog.py
- Your main vault file (e.g. vault_instr_master.json)
- Your new tools file (new_tools.json ← no ids yet)

🖥 How to Launch

Open vault_gui.py in **PyCharm** or **VS Code** and hit ▶️.
Or from terminal:

```
python vault_gui.py
```

🧭 Walkthrough of Each Step

1. Master Vault JSON

🟦 This is the big one: your current full vault.
📝 It usually lives at: vault_instr_master.json
✅ Enter its path or click **Browse...**

🧠 You'll see a reminder under the field:
“(This file is your MASTER vault – change its name as needed)”

---

2. New Tools JSON

🟨 A fresh file with only the new entries (no "id"s yet).
Looks like:

```
[
 {
   "name": "Sky Reverb",
   "type": "Effect",
   "developer": "Blue Aura",
   "tags": ["reverb", "vst3", "atmospheric"]
 }
]
```

Create this manually or however you collect new additions.

✅ Select this file in the GUI.

---

3. ID’d Tools JSON

🟧 After you assign IDs, the script will create
a file like new_tools_with_ids.json.

✅ You can manually browse to this file
or let the GUI auto-fill this field after Step 4.

🧪 Step-by-Step Buttons

🔢 1) Assign IDs

• Reads your “new tools” JSON
• Injects a unique id for each entry

• Saves as new_tools_with_ids.json
• Auto-fills the ID’d Tools field
• Also creates a CSV index (e.g. new_tools_with_ids_index.csv)

🔁 2) Merge Tools

• Compares vault_instr_master.json with new_tools_with_ids.json
• Matches on name + developer
• Overwrites duplicates
• Appends any brand-new tools
• Saves new file as vault_merged.json

🧽 You decide if/when to rename it back to your live master.

```
mv vault_merged.json vault_instr_master.json
```

▯ 3) Backup Vault

• Creates a timestamped copy of your master vault
• Example: backups/vault_instr_master_20250701_153200.json

Safe to run before any big operation!

🌍 4) Rebuild HTML

• Runs your static-site generator
• Refreshes subscotia_vault.html with all your newest tools
• Instant gratification

▯ Output Files You’ll See

File	What It Is
new_tools_with_ids.json	New tools + assigned IDs
vault_merged.json	Merged vault (existing + new entries)
vault_instr_master.json	Your live vault (overwrite this manually)

File	What It Is
subscotia_vault.html	Your sexy interactive catalog
backups/vault_*.json	Daily snapshots—just in case
*_index.csv	Handy developer-friendly summary list

💡 Tips & Extras

• You can run individual scripts (like generate_catalog_ids.py) from **PyCharm Run Configurations**
• If any entry is missing "tags", use manage_untagged.py to extract/fix/merge
• Want to delete tools? I built you a pruning tool too—just ask

---

Let me know if you'd like this tutorial baked into the GUI itself with a Help menu or printable PDF. You've got a whole command center now! 😎