**Project Roadmap: Stock Swap Tracker**

**🗺️ Phase 1: Core Foundation (V1)**

* **Core Feature: Database & App Setup**
  + Initialized Flask application.
  + Established SQL Server connection.
  + Created core database models: SwapRequest and InventoryLog.

**🧠 Phase 2: Logic & Data Integrity (V2)**

* **Key Feature: "Static Bin" Logic**
  + Fixed the initial flaw where parts were not returned to their original bins.
  + cancel\_request, reopen\_for\_claim, and edit\_dispatch routes were all modified to log stock back into the *specific, original* stock\_bin the part came from.
* **Key Feature: Acronym & SKU Association**
  + Stopped incorrectly logging the *requested* acronym on dispatch.
  + log\_inventory\_change was updated to auto-lookup the correct acronym for a part\_sku during adjustments (like cancellations), ensuring the log history remains accurate.

**🎨 Phase 3: UI/UX & Error Proofing (V3)**

* **Key Feature: Filtered Stock Dropdowns**
  + The **Dispatch** and **Edit Dispatch** pages are now filtered.
  + Users can only select parts from stock that have the same part\_acronym as the one being requested, preventing incorrect dispatches (e.g., dispatching an "LCDBC" for a "BC" request).
* **Key Feature: "Suggestion" Datalists**
  + The **Receive** and **Inventory Management** pages were changed from simple text boxes to <input> fields with <datalist> suggestions.
  + This helps maintain consistency for part\_sku and bin entries.
* **Key Feature: Dynamic Acronym Creation**
  + The **Inventory Management** page's "Part Acronym" field was changed from a <select> dropdown to an <input> with a <datalist>, allowing you to add new acronyms to the system simply by typing them.

**📦 Phase 4: Batch Processing Workflow (V4)**

* **Key Feature: Multi-Part Requests**
  + request.html was rebuilt to be dynamic.
  + You can now "Add Another Part" to log multiple part requests under a single, shared **WO/Ticket**, **Serial Number**, and **OEM Claim #**.
* **Key Feature: Batch Dispatch & Receive**
  + dispatch.html and receive.html were rebuilt to handle batches.
  + Clicking "Dispatch" or "Receive" on any *one* part now loads a page showing *all* related parts for that same WO/Serial/Claim, allowing you to process them simultaneously.

**🔒 Phase 5: Auditing & Security (V5)**

* **Key Feature: User Authentication**
  + Integrated Flask-Login and flask-bcrypt into the application.
  + Added a User model to the database.
  + Created /login, /logout, and /register routes.
* **Key Feature: Auditing**
  + All app routes (Dashboard, Log Request, Dispatch, etc.) are now protected by @login\_required.
  + The log\_inventory\_change function now **prepends the current\_user.username** to every "Notes" entry in the InventoryLog for a complete audit trail.
  + Flash messages for key actions now include the username.

**🚀 Future (On the Roadmap)**

* **Key Feature: Searchable Audit Log**
  + A new dedicated page (/audit\_log) will be created.
  + It will feature a search form allowing you to filter the *entire* InventoryLog history by:
    - **Serial Number** (by looking up related\_request\_ids)
    - **Part SKU**
    - **Bin Location**
    - **Part Acronym**
    - **Notes / User** (e.g., search for "gmartinez" or "PPID")