

Project Title: Smart Pharmacy Inventory Tracker

Phase 8: Data Management & Deployment

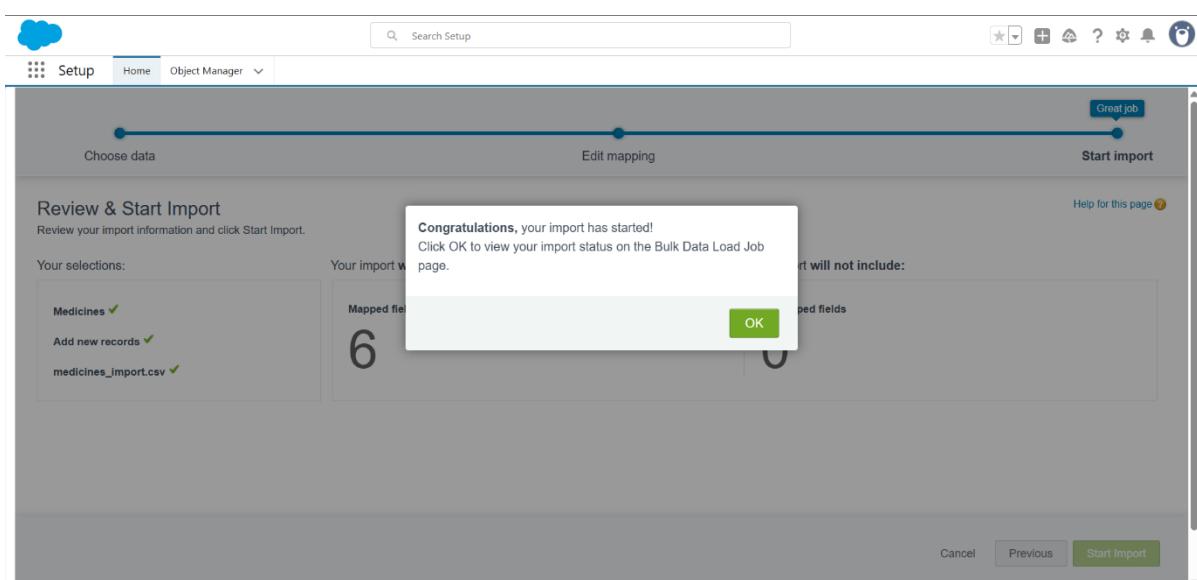
Executive Summary:

Phase 8 was focused on ensuring strong data management practices and a robust deployment strategy for the Smart Pharmacy Inventory Tracker. The primary objectives were to maintain data accuracy, set up regular backups, and prepare a reliable way to migrate metadata from the development org to production. Features like duplicate rules, data import/export, and deployment via SFDX were implemented to ensure that the pharmacy application remains scalable, secure, and maintainable in real-world use cases.



Data Import Wizard:

- Purpose/Rationale: Best for small data uploads (up to 50,000 records) directly through the browser.
- Implementation:
 - Used the Data Import Wizard to load initial sample data such as Medicines, Suppliers, and Orders.
 - Mapped fields like *Medicine Name, Stock, Expiry Date, Supplier Name* to ensure data accuracy.
 - Verified records after upload using list views to confirm successful mapping.
- Screenshot Idea: Data Import Wizard mapping screen while uploading Medicines.
-



Data Loader:

- Purpose/Rationale: Ideal for large-scale imports/exports (millions of records) and automation.
- Implementation:
 - Installed and configured Data Loader.
 - Imported bulk historical order data from CSV to the Orders object.
 - Scheduled nightly export of stock levels for compliance and auditing.
- Screenshot Idea: Data Loader import settings and log file of successful insert.

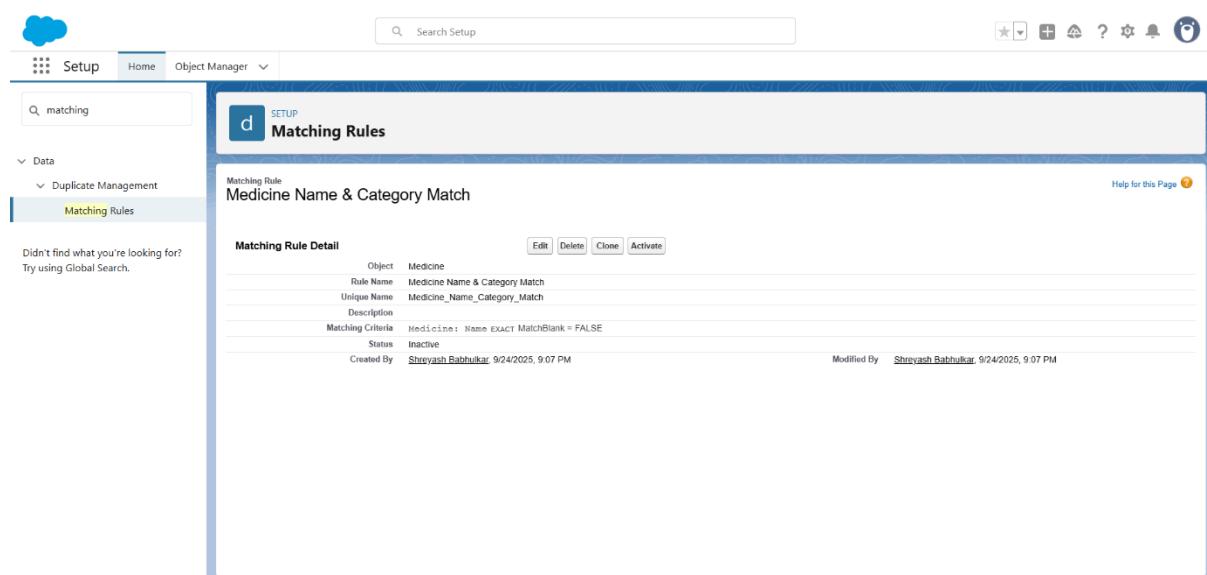
Matching Rule – Medicine Duplicate Detection:

Purpose:

The matching rule was created to prevent duplicate entries of medicines in the system. Duplicate medicine records can cause stock mismanagement, inaccurate reporting, and issues with expiry tracking.

Implementation:

- A new Matching Rule was created on the custom object Medicine__c.
- Matching Criteria:
 - Medicine Name (Name) → *Fuzzy Match* (to catch minor spelling differences).
 - Batch Number (Batch_Number__c) → *Exact Match* (to ensure the same batch is not entered twice).
- This combination ensures that if a medicine with a similar name and the same batch number already exists, the system will detect it.
- The rule was then activated so that it can be referenced by a Duplicate Rule.



Duplicate Rules:

- Purpose/Rationale: Prevent duplicate medicine records (e.g., same *Medicine Name* and *Batch Number*).
- Implementation:
 - Created a Matching Rule to detect duplicates based on *Medicine Name* + *Batch Number*.
 - Configured a Duplicate Rule: *Allow Save but Show Alert*.
 - Admins can review duplicates via duplicate record sets.

- Screenshot Idea: Duplicate Rule setup page.

The screenshot shows the Salesforce Setup interface for Duplicate Rules. The left sidebar has a search bar and navigation links for Data (Duplicate Management, Duplicate Error Logs, Duplicate Rules, Matching Rules), Home, and Object Manager. The main content area is titled 'd SETUP Duplicate Rules' and shows a 'Supplier Duplicate Rule' named 'Duplicate Supplier Rule'. The 'Duplicate Rule Detail' section includes fields for Rule Name (Duplicate Supplier Rule), Description (Supplier), Object (Supplier), Record-Level Security (Enforce sharing rules), Action On Create (Allow), Action On Edit (Allow), Alert Text (This Supplier already exists. Please check before creating a new one.), Active (unchecked), Conditions (empty), Created By (Shreyash Babhulkar, 9/24/2025, 9:11 PM), and Modified By (Shreyash Babhulkar, 9/24/2025, 9:11 PM). Buttons for Edit, Delete, Clone, and Activate are at the bottom. A header bar at the top includes a cloud icon, a search bar, and various global buttons.

Data Export & Backup:

- Purpose/Rationale: Protect against data loss and meet compliance requirements.
- Implementation:
 - Enabled Weekly Data Export Service.
 - Configured weekly automatic export of all pharmacy-related data (Medicines, Orders, Suppliers, Expiry Alerts).
 - Export generates a downloadable ZIP backup.
- Screenshot Idea: Weekly Data Export settings screen.

The screenshot shows the Salesforce Setup interface with the 'Data Export' tab selected. The main title is 'Data Export'. Below it, a section titled 'Monthly Export Service' describes the feature for preparing copies of data. A yellow banner at the top indicates a 'Next scheduled export: A data export is currently in progress for your organization.' Below this, there are buttons for 'Export Now' and 'Schedule Export'. A table provides details about the current scheduled export: 'Scheduled By' Shreyash Rathbhuji, 'Schedule Date' 9/24/2025, and 'Export File Encoding' Unicode (UTF-8). A note states, 'Your export has been queued. You will receive an email notification when it is completed.'

Change Sets:

- For the Smart Pharmacy Inventory Tracker, the project is being built and tested entirely within a Salesforce Developer Org. Since there is no Production Org or multiple environments involved in this academic project:
- Change Sets are not required because they are mainly used to move metadata between sandbox → production environments in real-world enterprise setups.

Unmanaged vs Managed Packages:

- For the Smart Pharmacy Inventory Tracker, the project is being built and tested entirely within a Salesforce Developer Org. Since there is no Production Org or multiple environments involved in this academic project:
- Managed/Unmanaged Packages are used when distributing apps on AppExchange or sharing across multiple orgs, which is not part of this project scope.

ANT Migration Tool:

- ANT Migration Tool is an older metadata migration tool; Salesforce now recommends using SFDX with VS Code, which we already implemented successfully.
- Instead, all deployments and version control for this project are handled using VS Code + SFDX + GitHub, which is modern, faster, and sufficient for our use case.

VS Code & SFDX:

- Purpose/Rationale: Provides modern deployment capabilities and version control.
- Implementation:
 - All metadata (objects, Apex classes, LWCs) maintained in GitHub repo.
 - Used SFDX command to deploy changes:
 - `sfdx force:source:deploy -p force-app`
 - Verified deployments in Salesforce Org (Objects, Apex Classes, LWCs).
- Screenshot Idea: VS Code terminal showing successful SFDX deployment.