PROJECT TITLE: MEDICAL INVENTORY MANAGEMENT

Collage name: LRG Government arts college for women

Collage Code: bru07

TEAM ID: NM2025TMID28060

TEAM MEMBERRS:

Team Leader name: Raska c

Email: rasika9705@gmail.com

Team Member 1: Raagini R

Email: ragini29122005@gmail.com

Team Member2: Raja Priya S

Email: priyagold1818@gmail.com

Team Member3: Ramya M

Email: ramyasathya379@gmail.com

1.INTRODUCTION

1.1 Project Overview

The Medical Inventory Management System is designed to streamline the management of medicines, suppliers, and purchase transactions within a healthcare organization. It allows users to efficiently manage purchase orders, order items, stock adjustments, and supplier-based reports, while minimizing errors in manual tracking**.**

1.2 Purpose

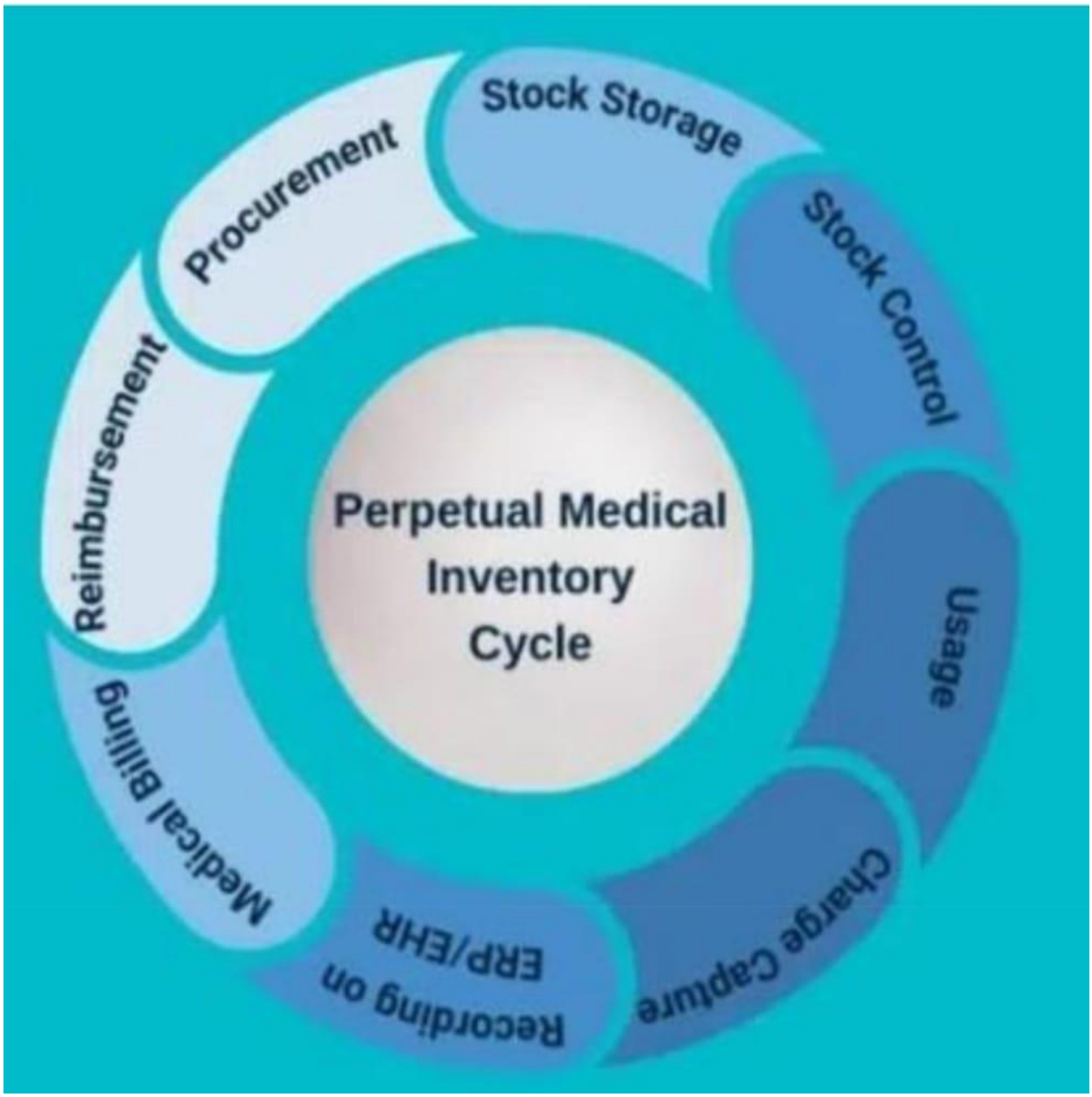
The main purpose of the project is to:

Track and manage purchase orders for medical supplies.

Maintain accurate records of inventory stock levels.

Enable quick generation of reports and dashboards for analysis.

Reduce data entry errors and ensure data integrity through validations.

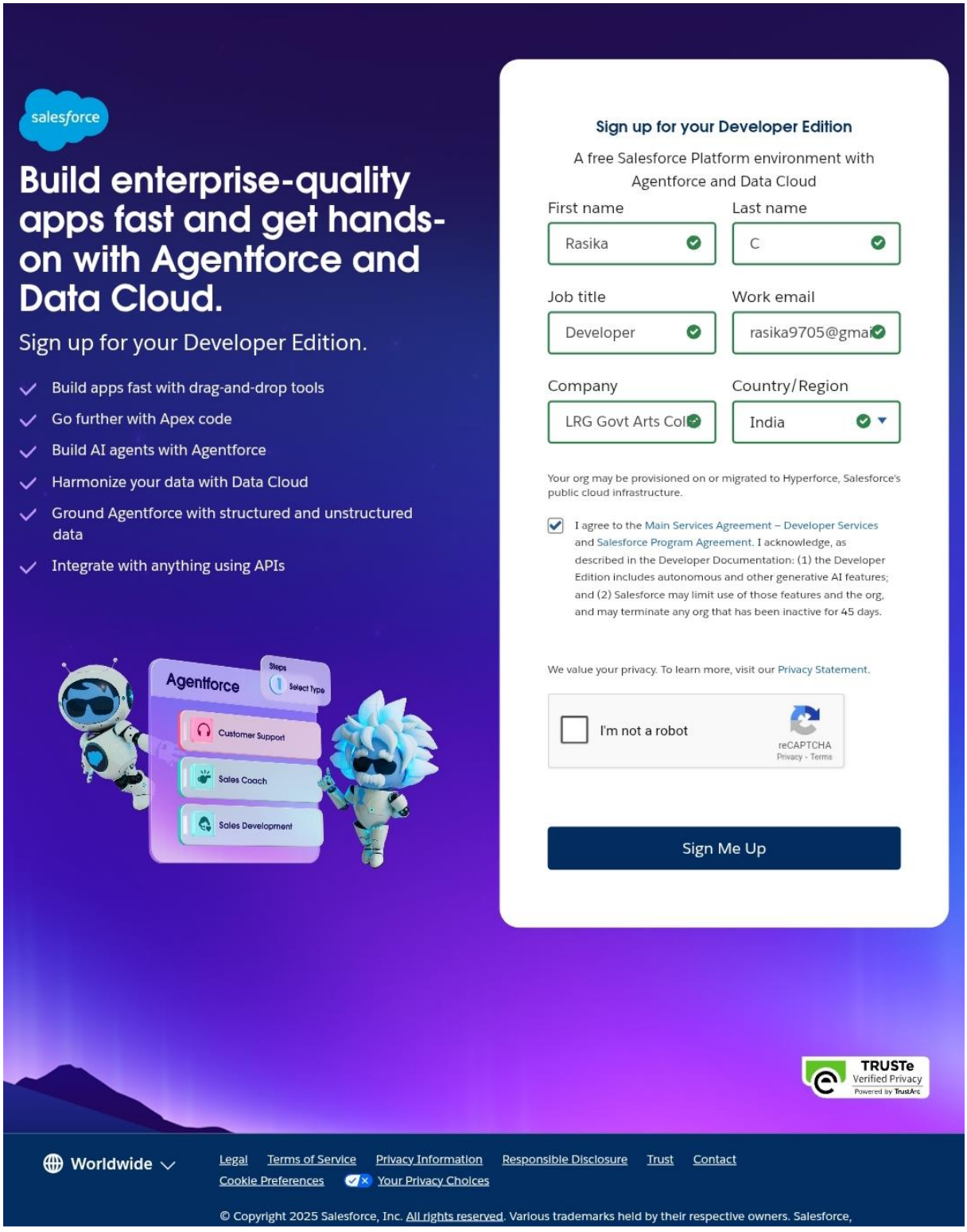


2. DEVELOPMENT PHASE

Creating Developer Account

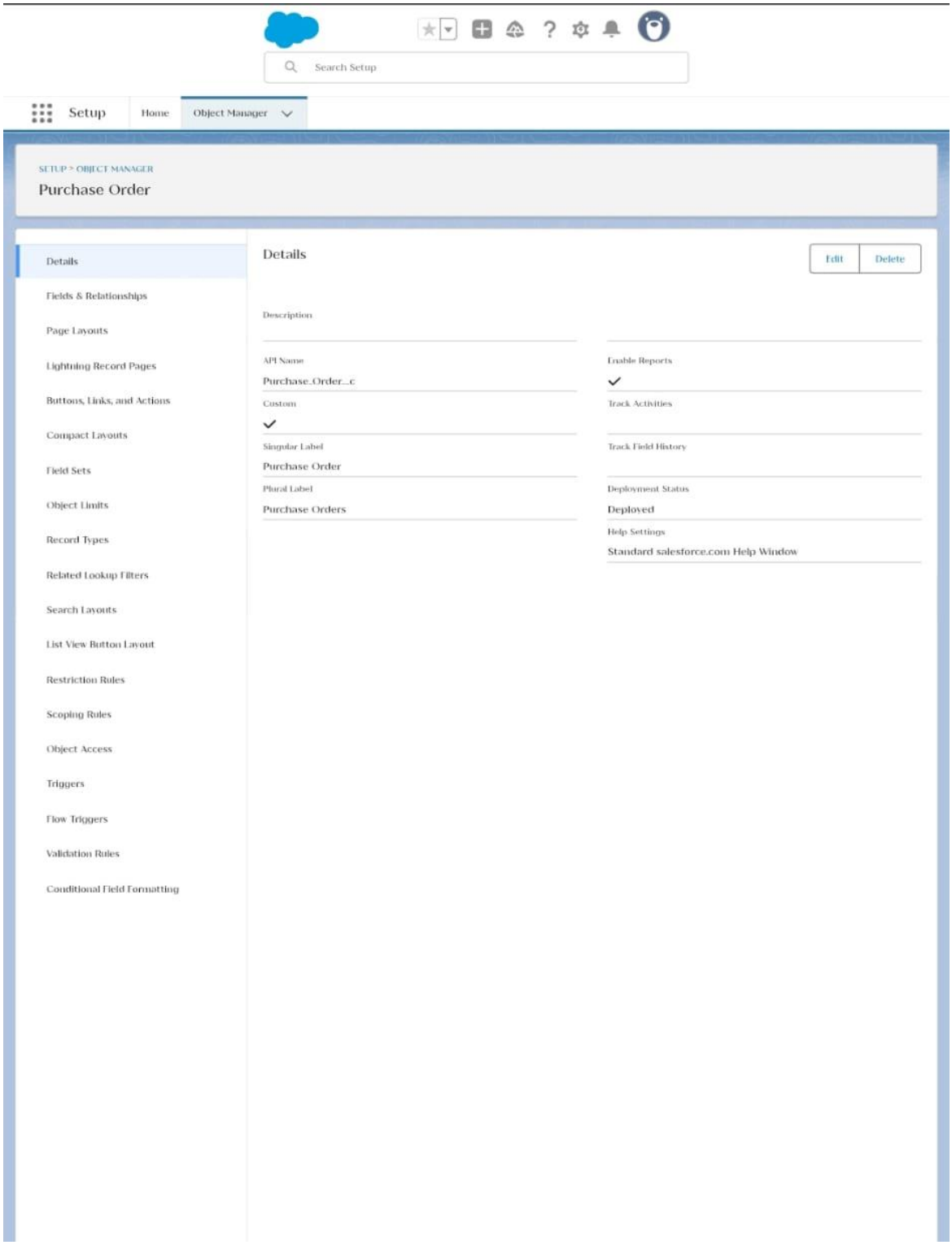
The project was developed on Salesforce Developer Org, created via:

https://developer.salesforce.com/signup

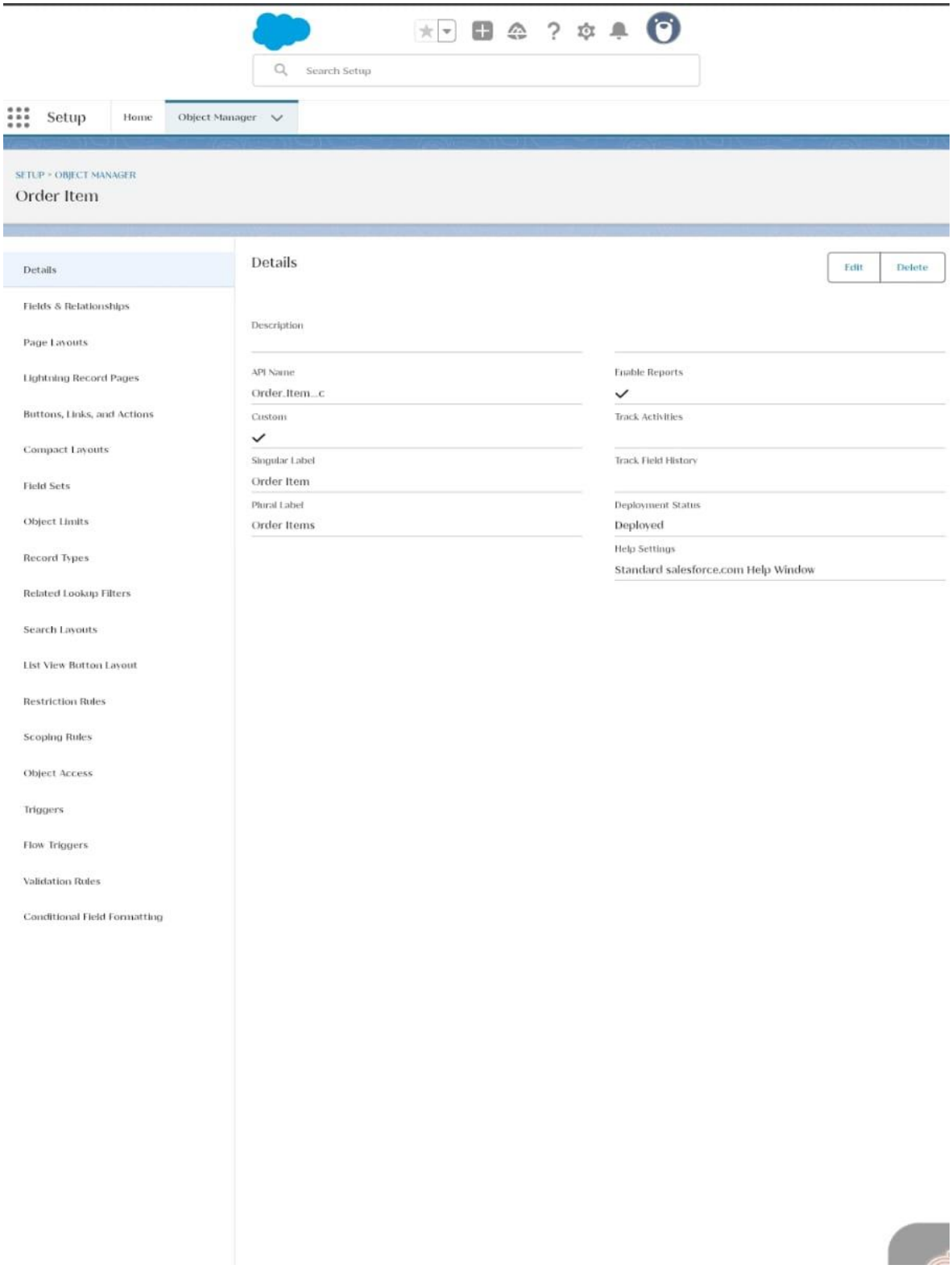


Object created

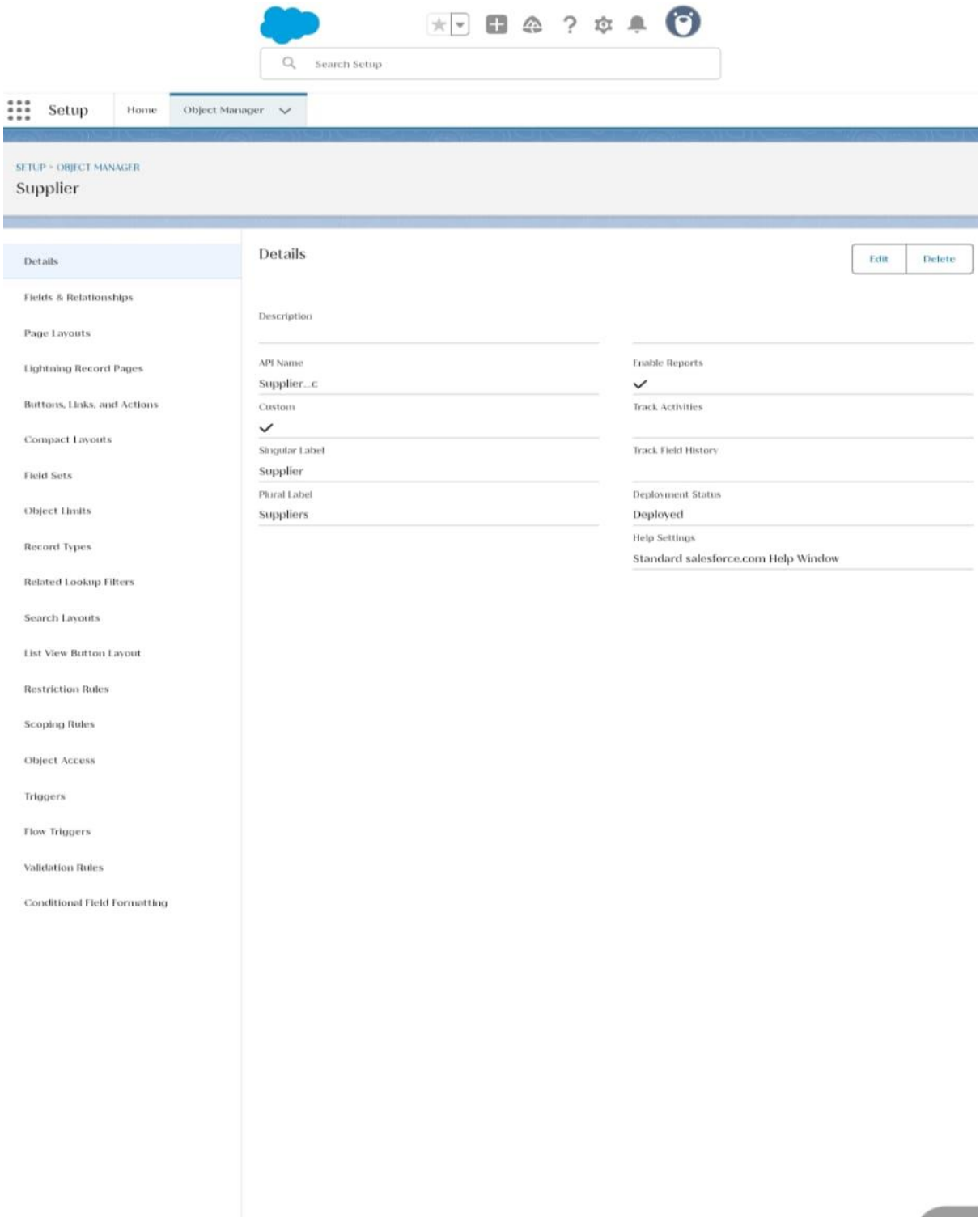
Purchase Order – To record supplier and order details.



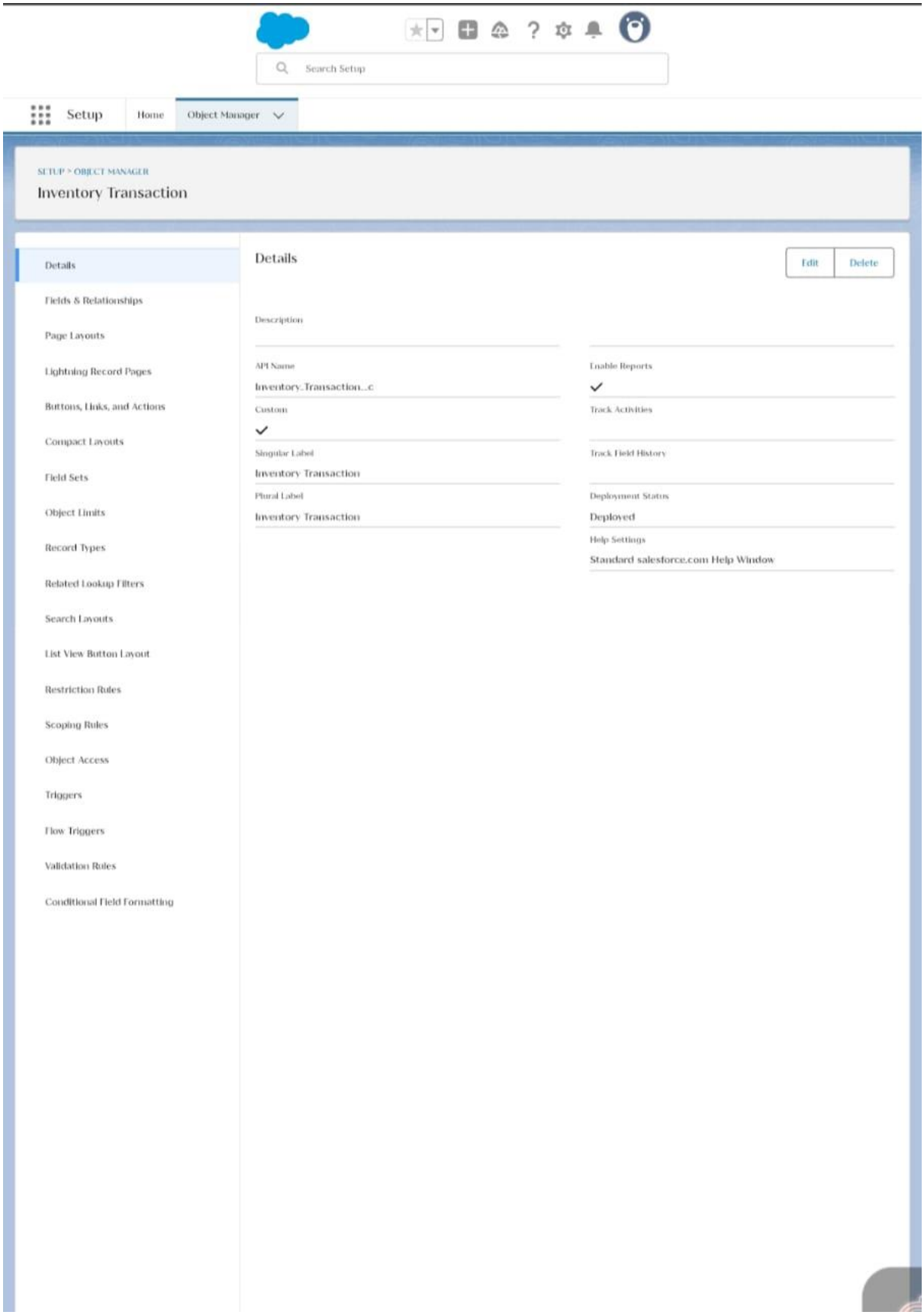
Order Item – To capture product-level details such as quantity and cost.



Supplier – To store supplier information.



Inventory – To track stock availability.


Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Inventory Transaction
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Details
Edit
Delete
Description
API Name
Inventory_Transaction__c
Custom
Singular Label
Inventory Transaction
Plural Label
Inventory Transaction
Enable Reports
Track Activities
Track Field History
Deployment Status
Deployed
Help Settings
Standard salesforce.com Help Window

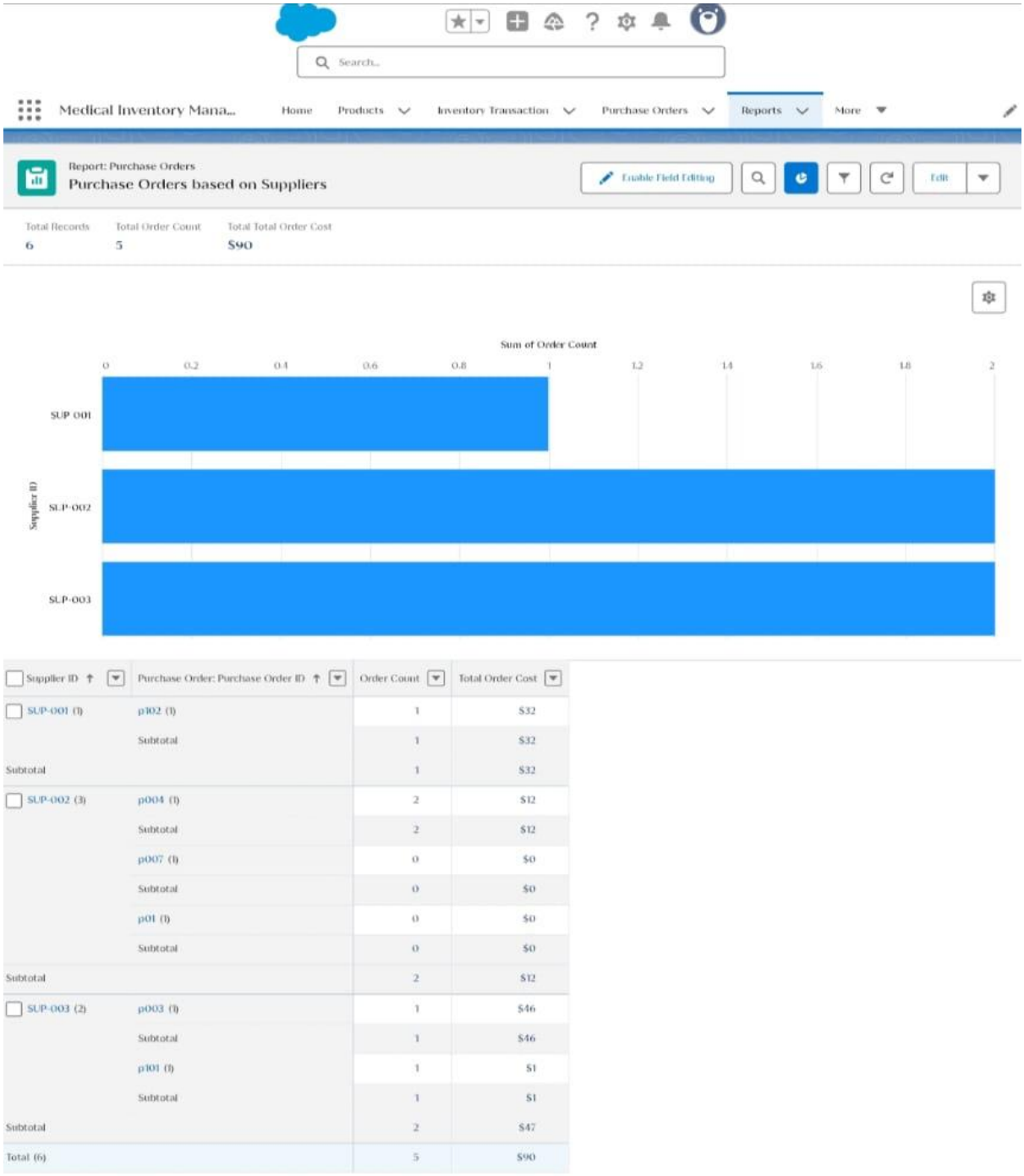
Fields & Relationships Configured

*Lookup relationship between Purchase Order and Supplier.

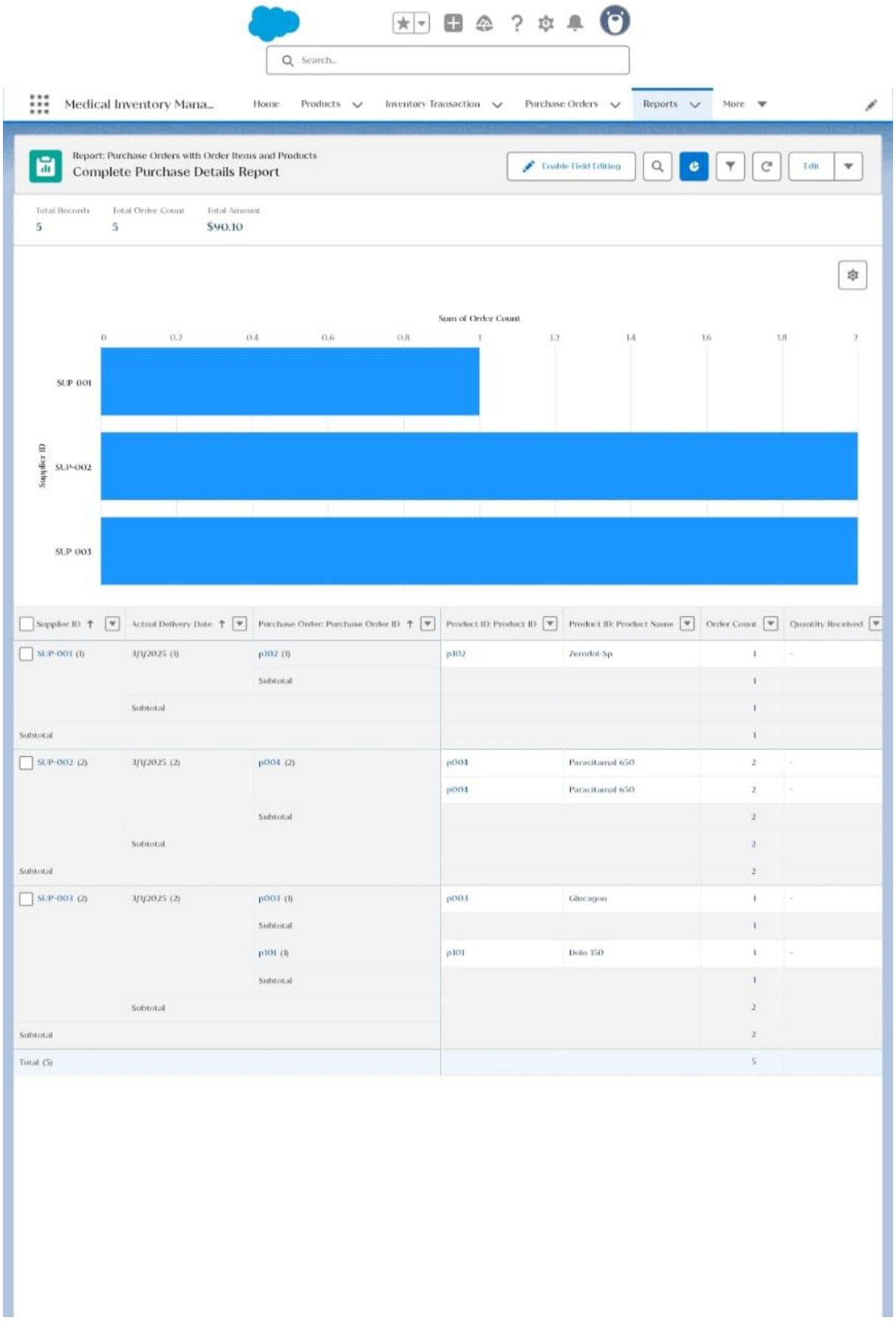
*Master-Detail relationship between *Purchase Order and Order Items.

*Formula field for calculating Total Order Cost.

Dashboard “**Medical Inventory Dashboard**” with charts for supplier analysis

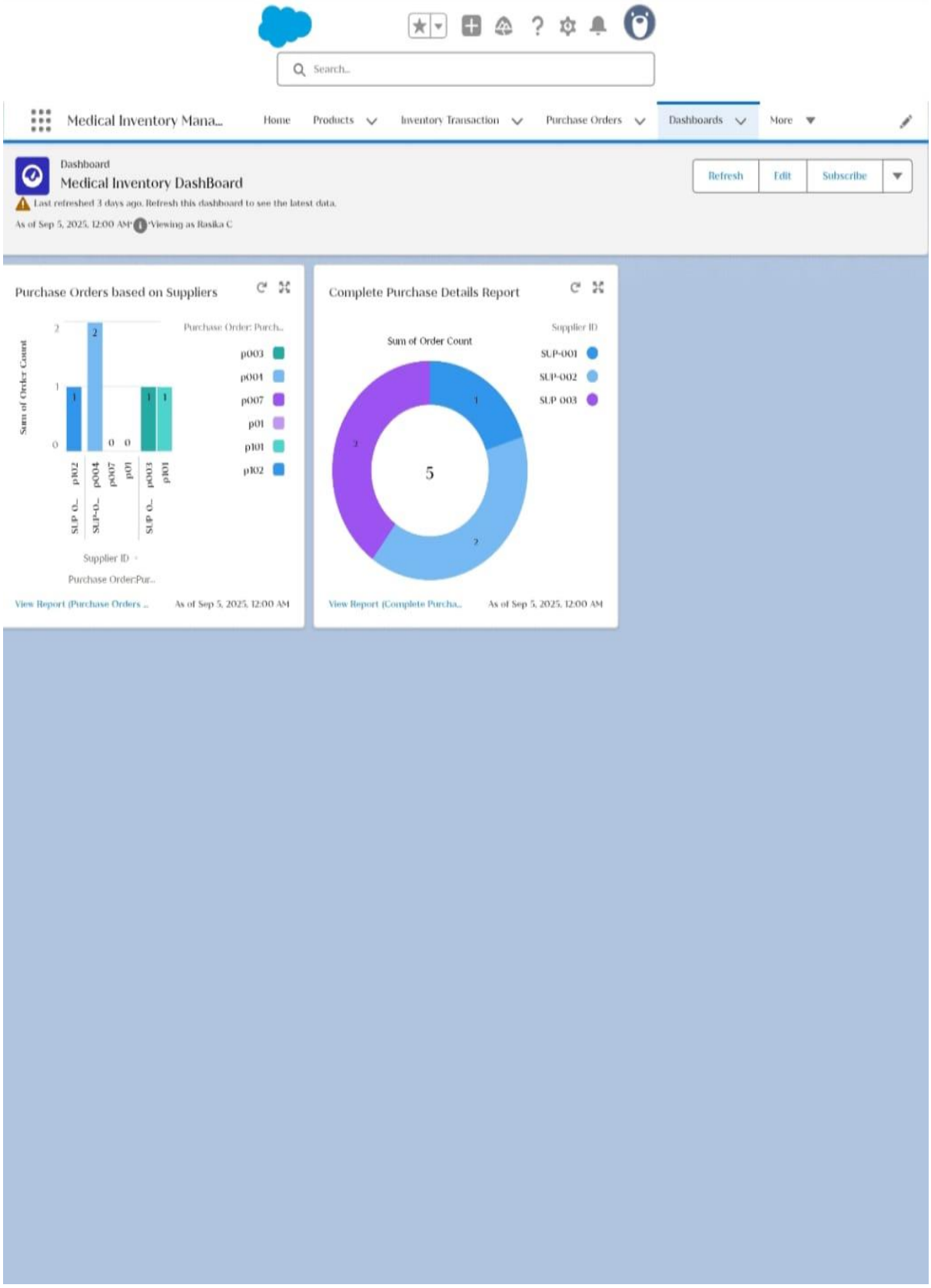


Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
SUP-001 (1)	p102 (1)	1	$32
	Subtotal	1	$32
Subtotal		1	$32
SUP-002 (3)	p004 (1)	2	$12
	Subtotal	2	$12
	p007 (1)	0	$0
	Subtotal	0	$0
	p01 (1)	0	$0
	Subtotal	0	$0
Subtotal		2	$12
SUP-003 (2)	p003 (1)	1	$46
	Subtotal	1	$46
	p101 (1)	1	$1
	Subtotal	1	$1
Subtotal		2	$47
Total (6)		5	$90


Search...
Medical Inventory Mana...
Home
Products
Inventory Transaction
Purchase Orders
Reports
More
Report: Purchase Orders with Order Items and Products
Complete Purchase Details Report
Enable Field Editing
Edit
Total Records
5
Total Order Count
5
Total Amount
$90.10
Sum of Order Count
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
SUP 001
SUP-002
SUP 003
Supplier ID
Supplier ID
Actual Delivery Date
Purchase Order: Purchase Order ID
Product ID: Product ID
Product ID: Product Name
Order Count
Quantity Received
SUP-001 (1)
3/1/2025 (1)
p102 (1)
p102
Zerodol-Sp
1
-
Subtotal
1
Subtotal
1
Subtotal
1
SUP-002 (2)
3/1/2025 (2)
p004 (2)
p004
Paracitamal 650
2
-
p004
Paracitamal 650
2
-
Subtotal
2
Subtotal
2
Subtotal
2
SUP-003 (2)
3/1/2025 (2)
p003 (1)
p003
Glucagon
1
-
Subtotal
1
p101 (1)
p101
Dolo 150
1
-
Subtotal
1
Subtotal
2
Subtotal
2
Total (5)
5

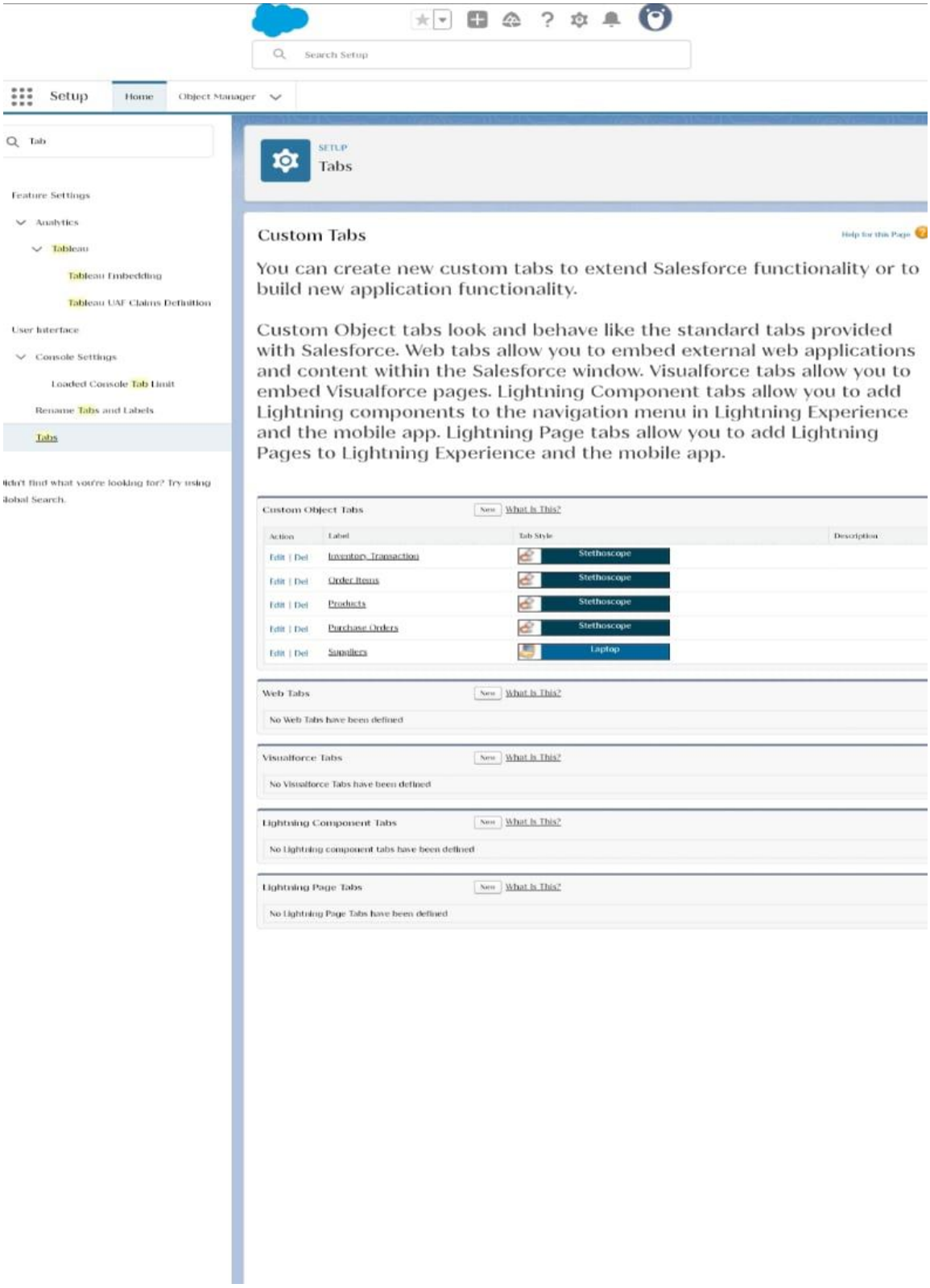
Dashboard “**Medical Inventory Dashboard**” with charts for supplier analysis

---

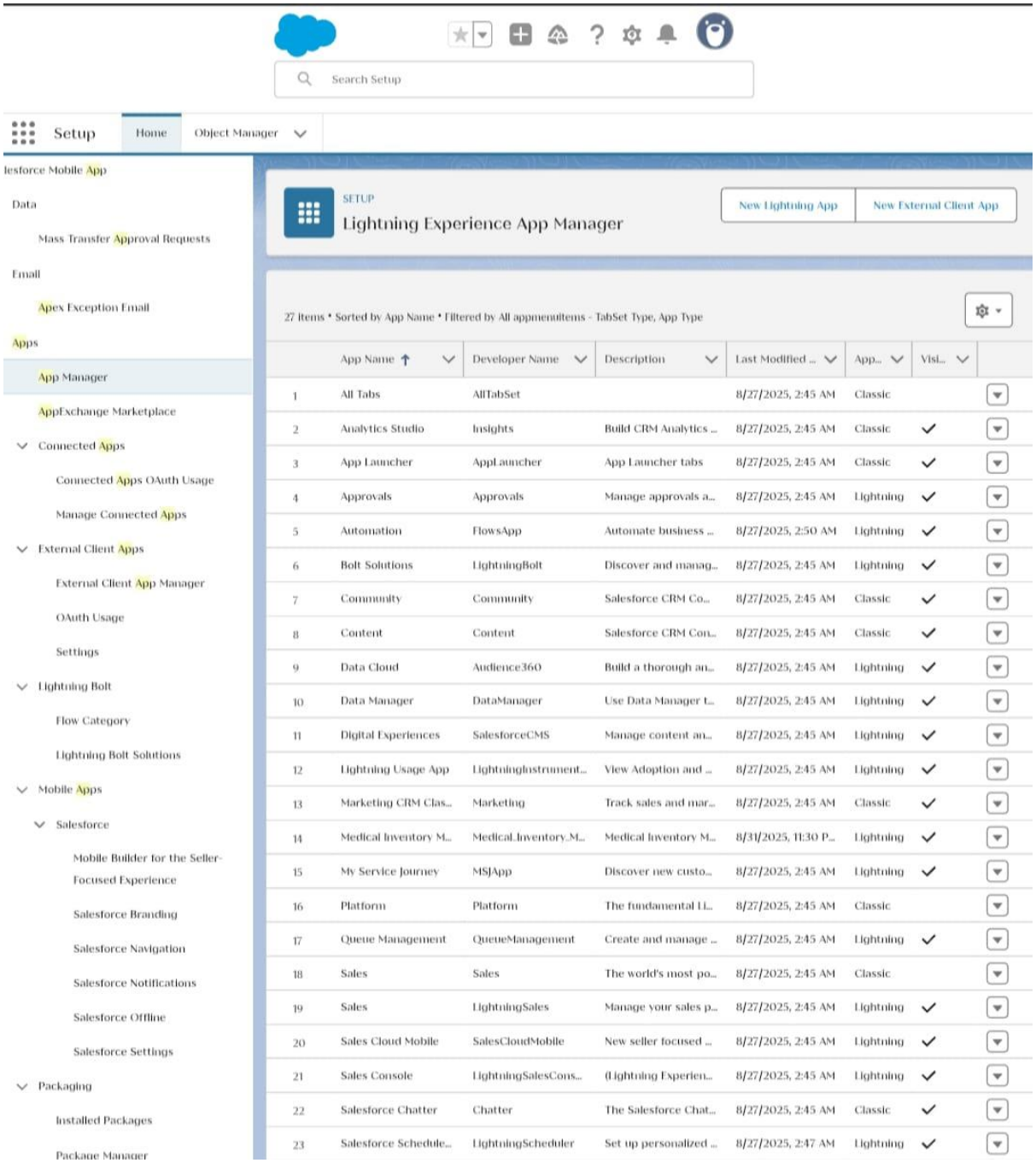


4.IMPLEMENTATION

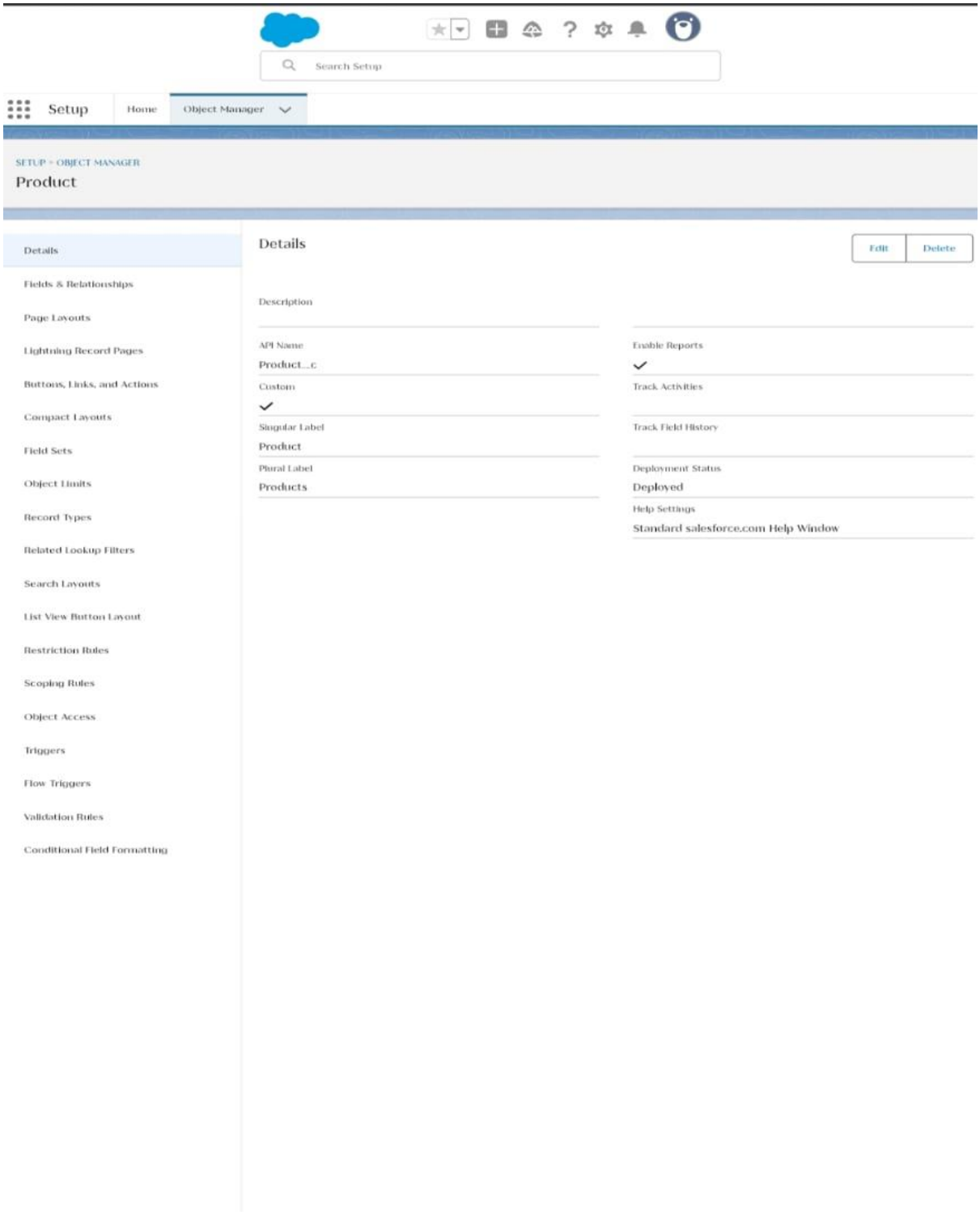
Tabs: you can create new custom tabs to extend sales for functionality or to build new application functionality.



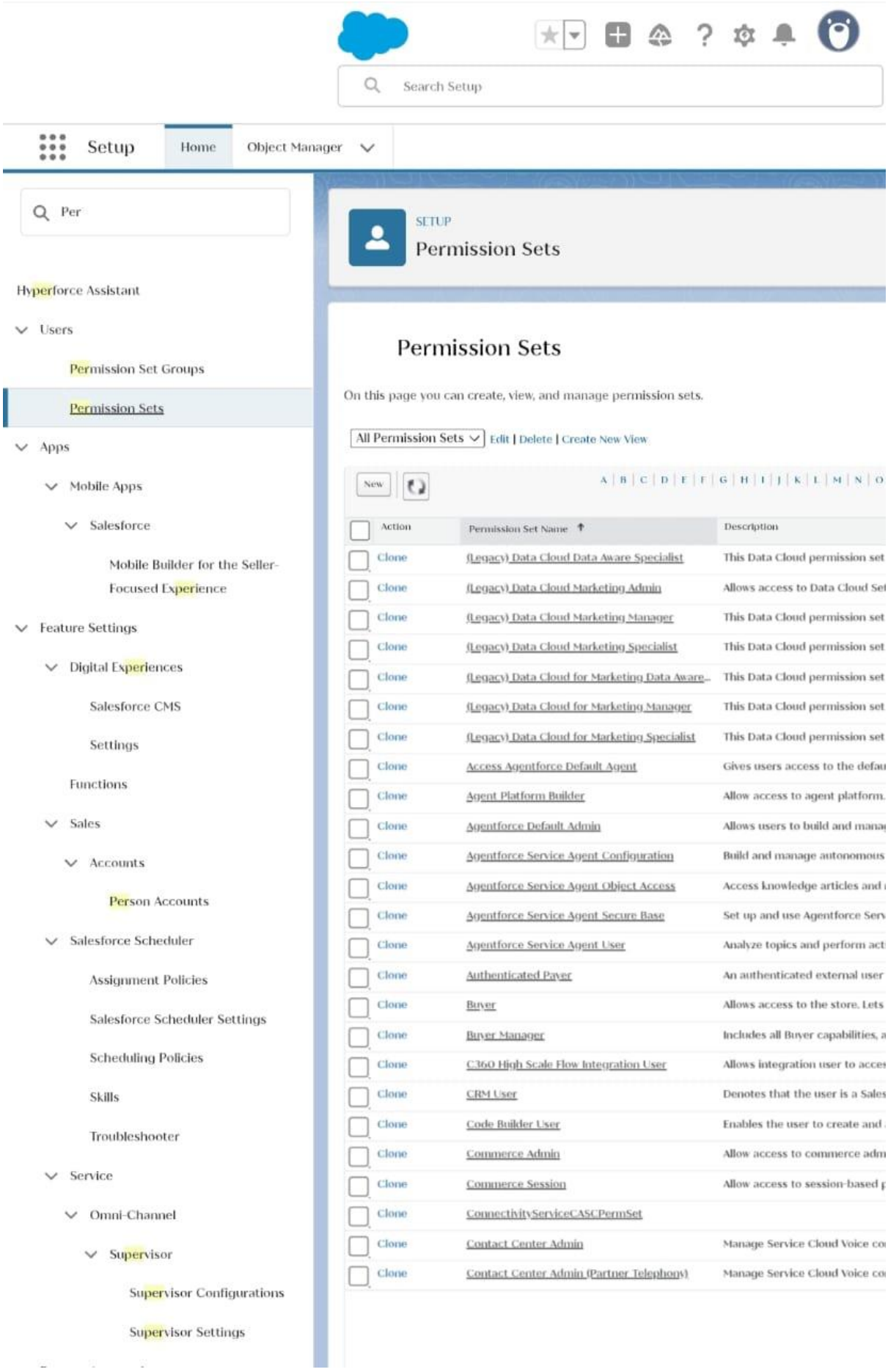
Lightings App: Is used to create a medical inventory management, which shows as a dashboard in that it allows us to provide the data, reports etc…,,



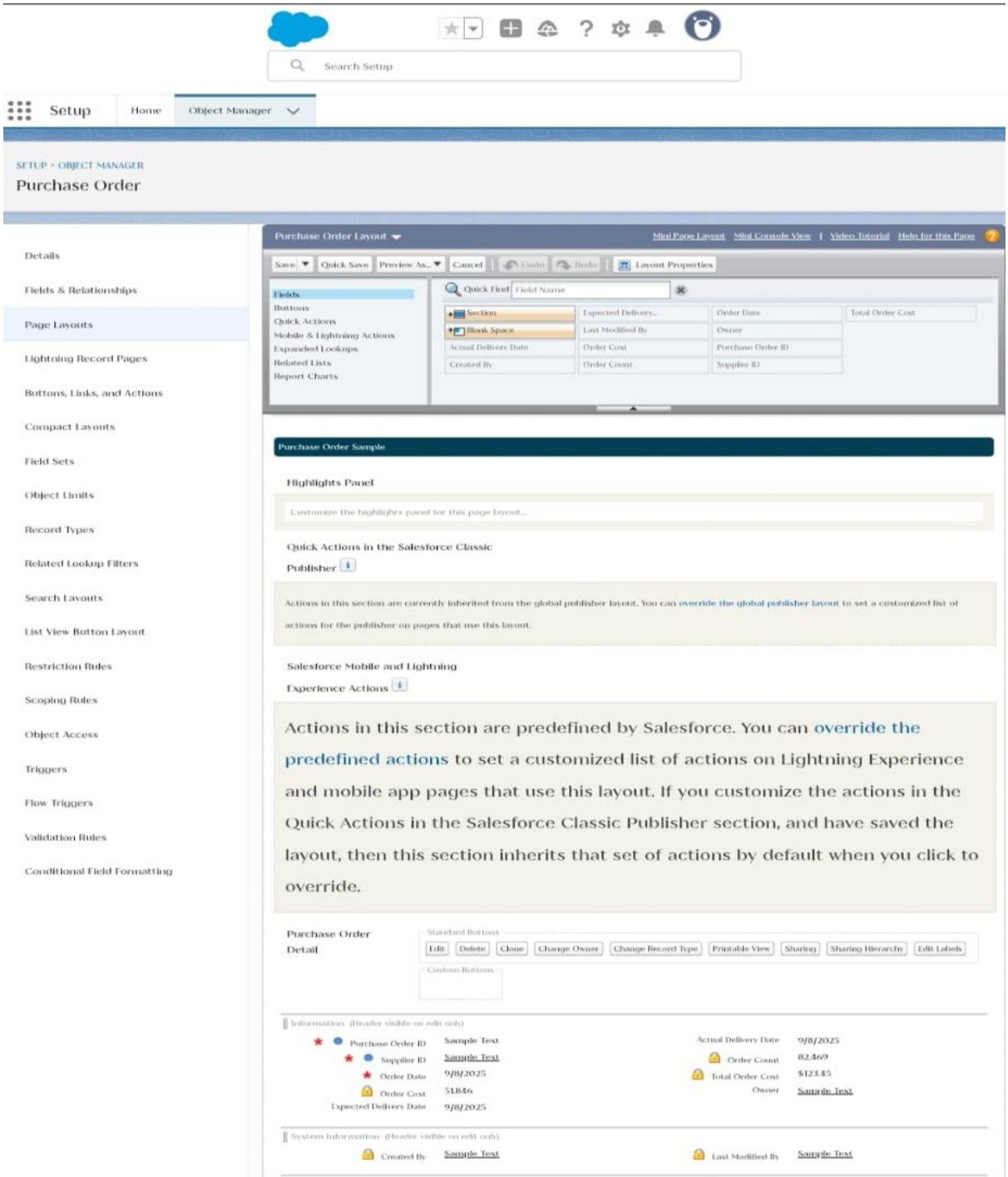
Text field: A text field is used whenever you need to capture descriptive or alphanumeric details that identify, describe, or provide context about medical items, suppliers, or transactions.

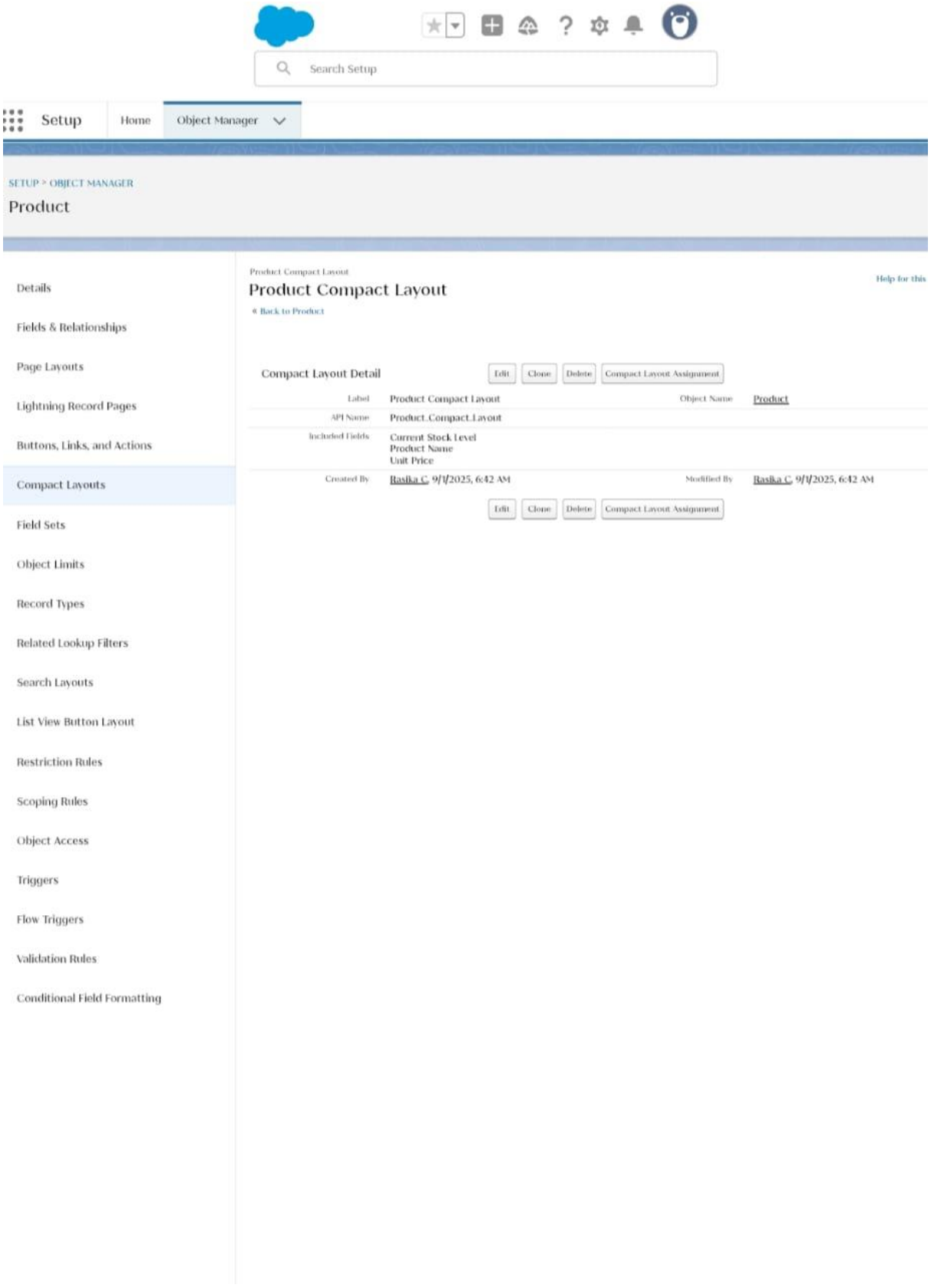


Validation Rules: Prevent negative or blank quantities.

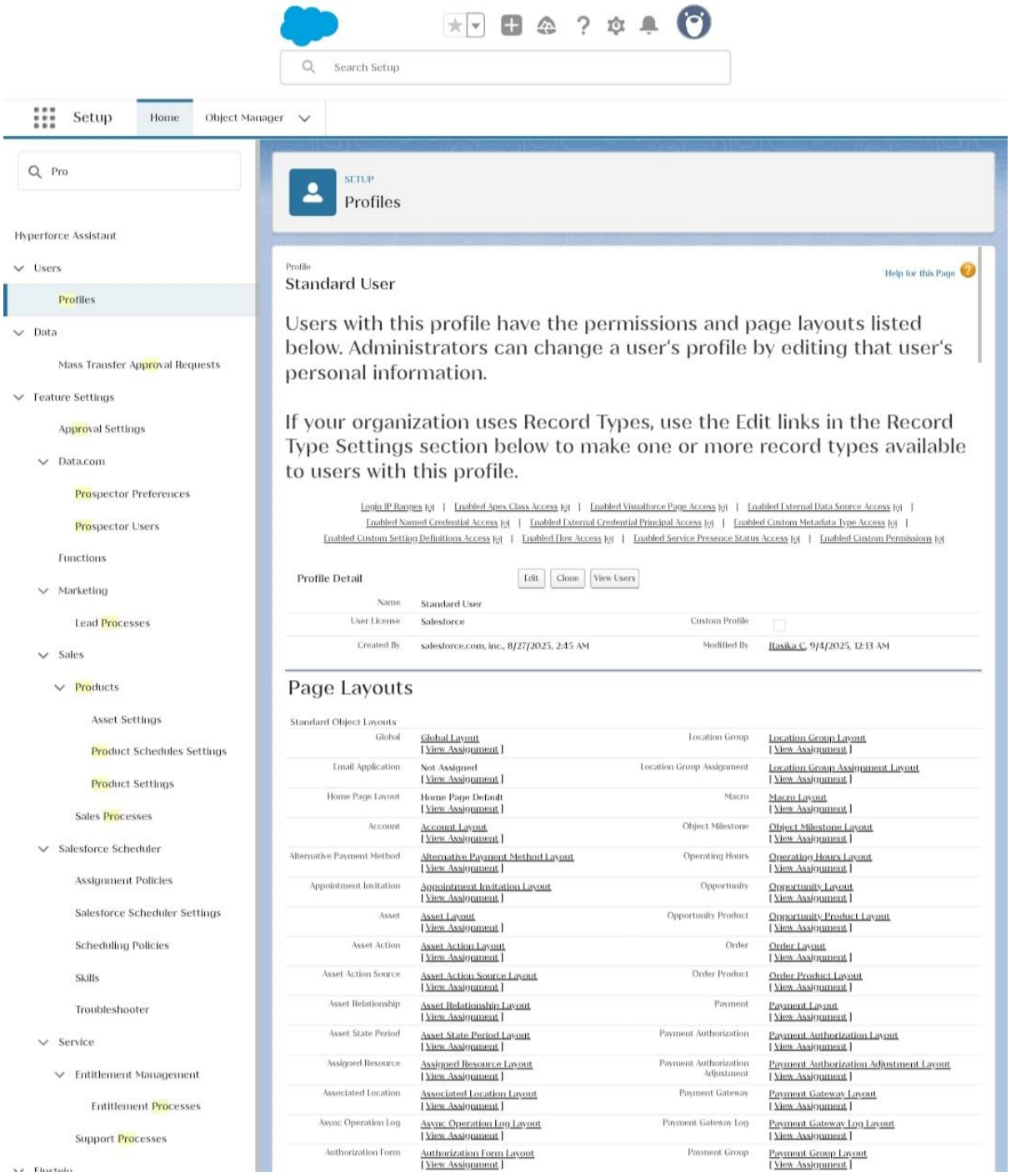

Search Setup
Setup
Home
Object Manager
Per
Hyperforce Assistant
Users
Permission Set Groups
Permission Sets
Apps
Mobile Apps
Salesforce
Mobile Builder for the Seller-Focused Experience
Feature Settings
Digital Experiences
Salesforce CMS
Settings
Functions
Sales
Accounts
Person Accounts
Salesforce Scheduler
Assignment Policies
Salesforce Scheduler Settings
Scheduling Policies
Skills
Troubleshooter
Service
Omni-Channel
Supervisor
Supervisor Configurations
Supervisor Settings
SETUP
Permission Sets
Permission Sets
On this page you can create, view, and manage permission sets.
All Permission Sets
Edit | Delete | Create New View
New
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O
Action
Permission Set Name
Description
Clone
(Legacy) Data Cloud Data Aware Specialist
This Data Cloud permission set
Clone
(Legacy) Data Cloud Marketing Admin
Allows access to Data Cloud Set
Clone
(Legacy) Data Cloud Marketing Manager
This Data Cloud permission set
Clone
(Legacy) Data Cloud Marketing Specialist
This Data Cloud permission set
Clone
(Legacy) Data Cloud for Marketing Data Aware...
This Data Cloud permission set
Clone
(Legacy) Data Cloud for Marketing Manager
This Data Cloud permission set
Clone
(Legacy) Data Cloud for Marketing Specialist
This Data Cloud permission set
Clone
Access Agentforce Default Agent
Gives users access to the defau
Clone
Agent Platform Builder
Allow access to agent platform.
Clone
Agentforce Default Admin
Allows users to build and manag
Clone
Agentforce Service Agent Configuration
Build and manage autonomous
Clone
Agentforce Service Agent Object Access
Access knowledge articles and
Clone
Agentforce Service Agent Secure Base
Set up and use Agentforce Serv
Clone
Agentforce Service Agent User
Analyze topics and perform act
Clone
Authenticated Payer
An authenticated external user
Clone
Buyer
Allows access to the store. Lets
Clone
Buyer Manager
Includes all Buyer capabilities, a
Clone
C360 High Scale Flow Integration User
Allows integration user to acces
Clone
CRM User
Denotes that the user is a Sales
Clone
Code Builder User
Enables the user to create and
Clone
Commerce Admin
Allow access to commerce adm
Clone
Commerce Session
Allow access to session-based
Clone
ConnectivityServiceCASCPermSet
Clone
Contact Center Admin
Manage Service Cloud Voice co
Clone
Contact Center Admin (Partner Telephony)
Manage Service Cloud Voice co

Page layout: The Page Layout in the Product object is used to customize how medical product information is displayed, making it easier to manage medicines, ensure compliance, and improve accuracy in inventory management.

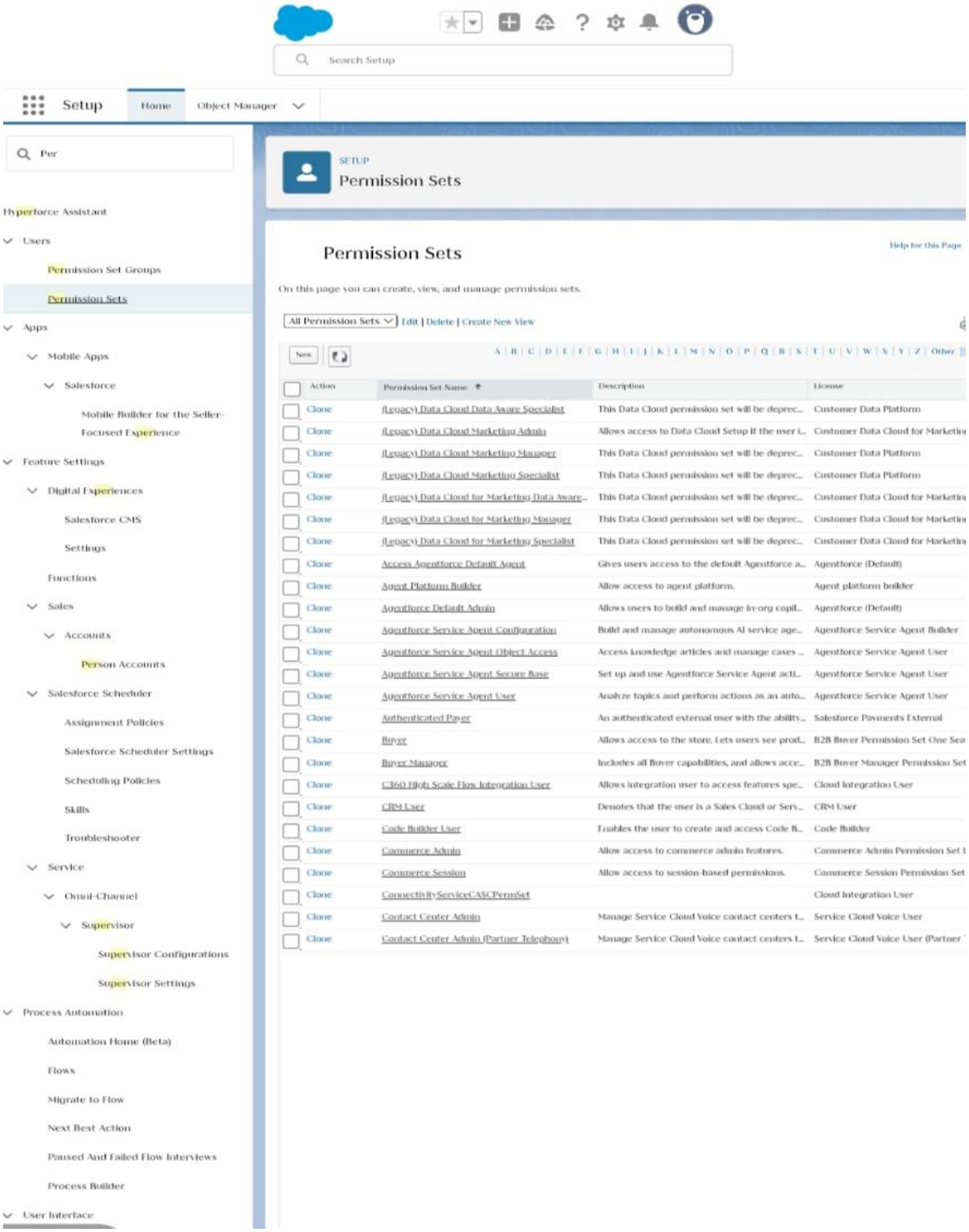


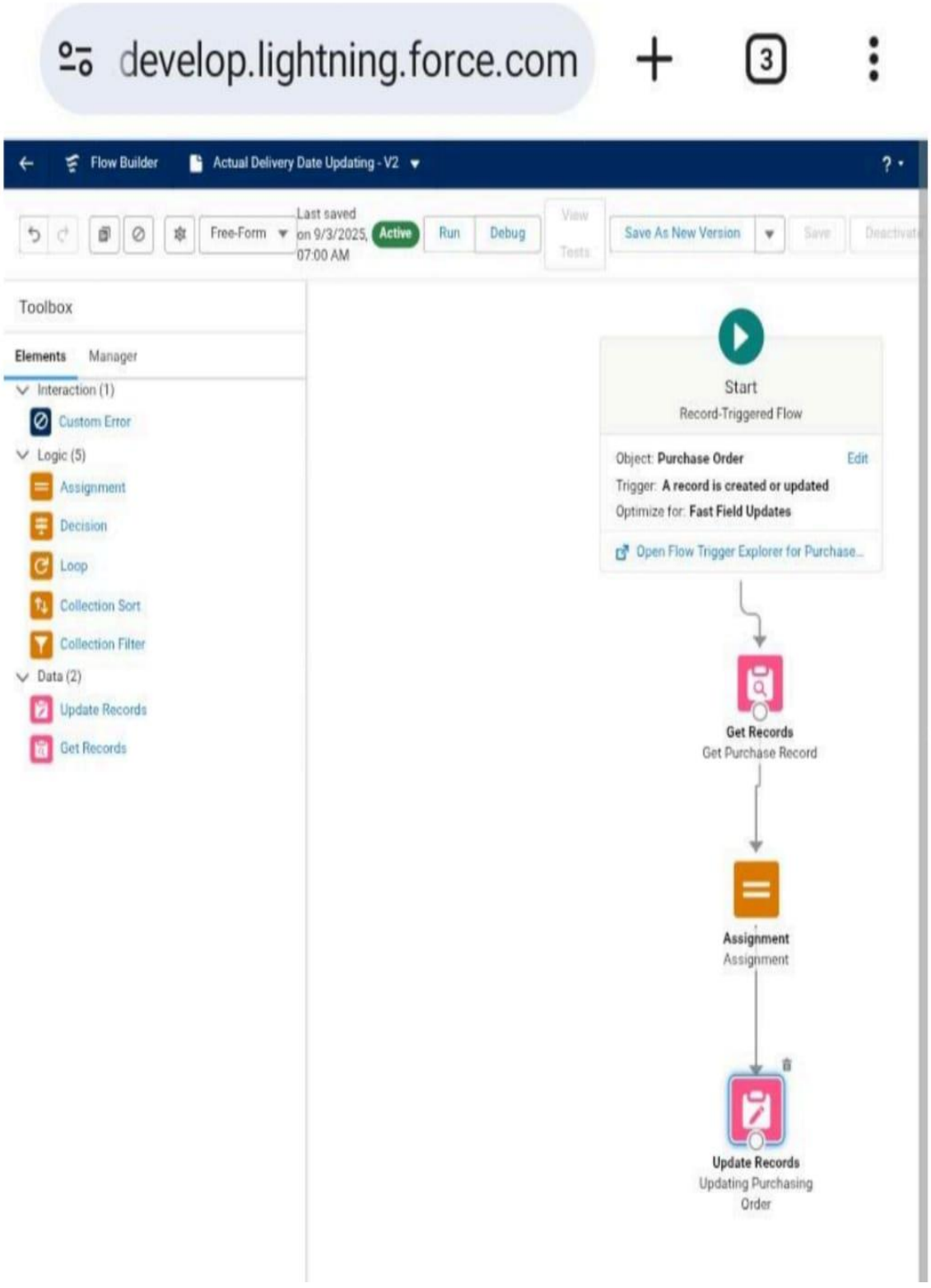

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Product
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Product Compact Layout
Product Compact Layout
Help for this
« Back to Product
Compact Layout Detail
Edit
Clone
Delete
Compact Layout Assignment
Label
Product Compact Layout
Object Name
Product
API Name
Product_Compact_Layout
Included Fields
Current Stock Level
Product Name
Unit Price
Created By
Rasika C, 9/1/2025, 6:42 AM
Modified By
Rasika C, 9/1/2025, 6:42 AM
Edit
Clone
Delete
Compact Layout Assignment

Profiles: Profiles in medical inventory management are used to control user permissions. They decide who can view, create, edit, or delete records like Products, Purchase Orders, and Transactions, ensuring data security and proper role-based access.

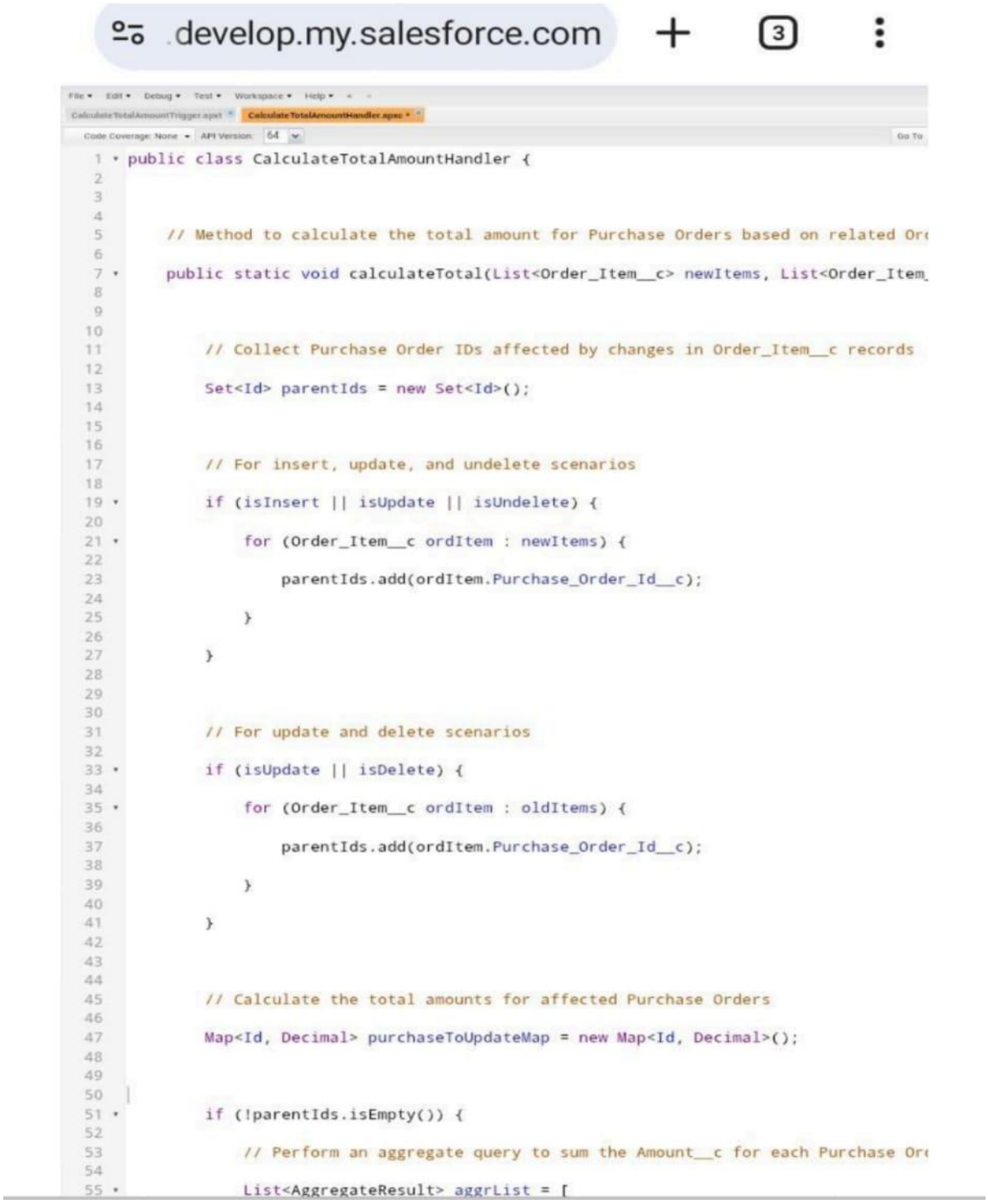


Permission Set: Permission Sets in medical inventory management are used to give additional access to users without changing their profile. They allow specific staff to perform extra tasks like updating stock, approving purchase orders, or viewing reports when needed.

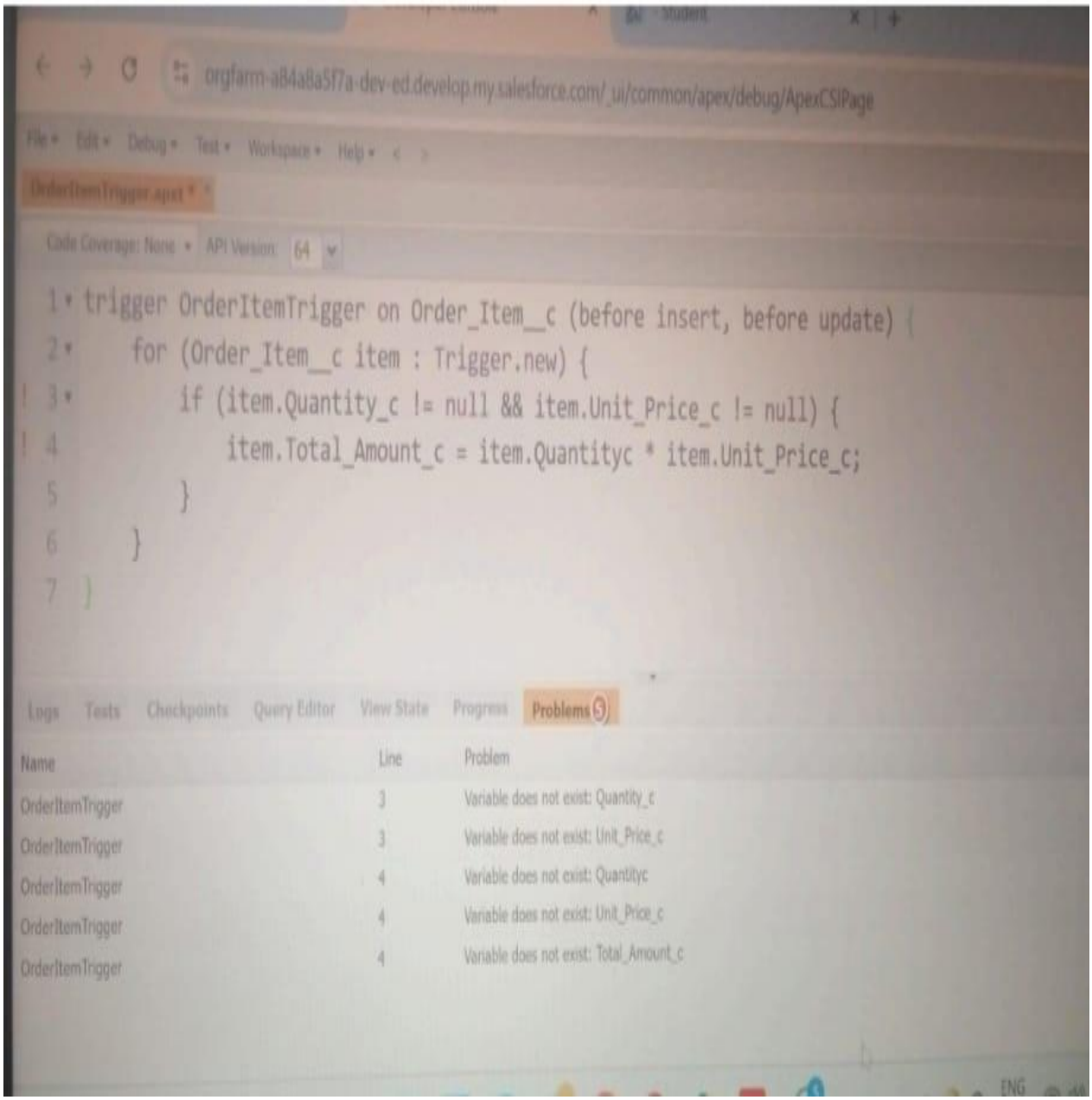




Apex Trigger: Auto-calculate total cost from Order Items.



Error Handling: Resolved issues like “field not writable” and “no records found” by modifying field accessibility and report filters.



3. FUNCTIONAL AND PERFORMANCE TESTING

Tested formula fields to ensure automatic total calculation.

Verified report filters to display correct purchase order data.

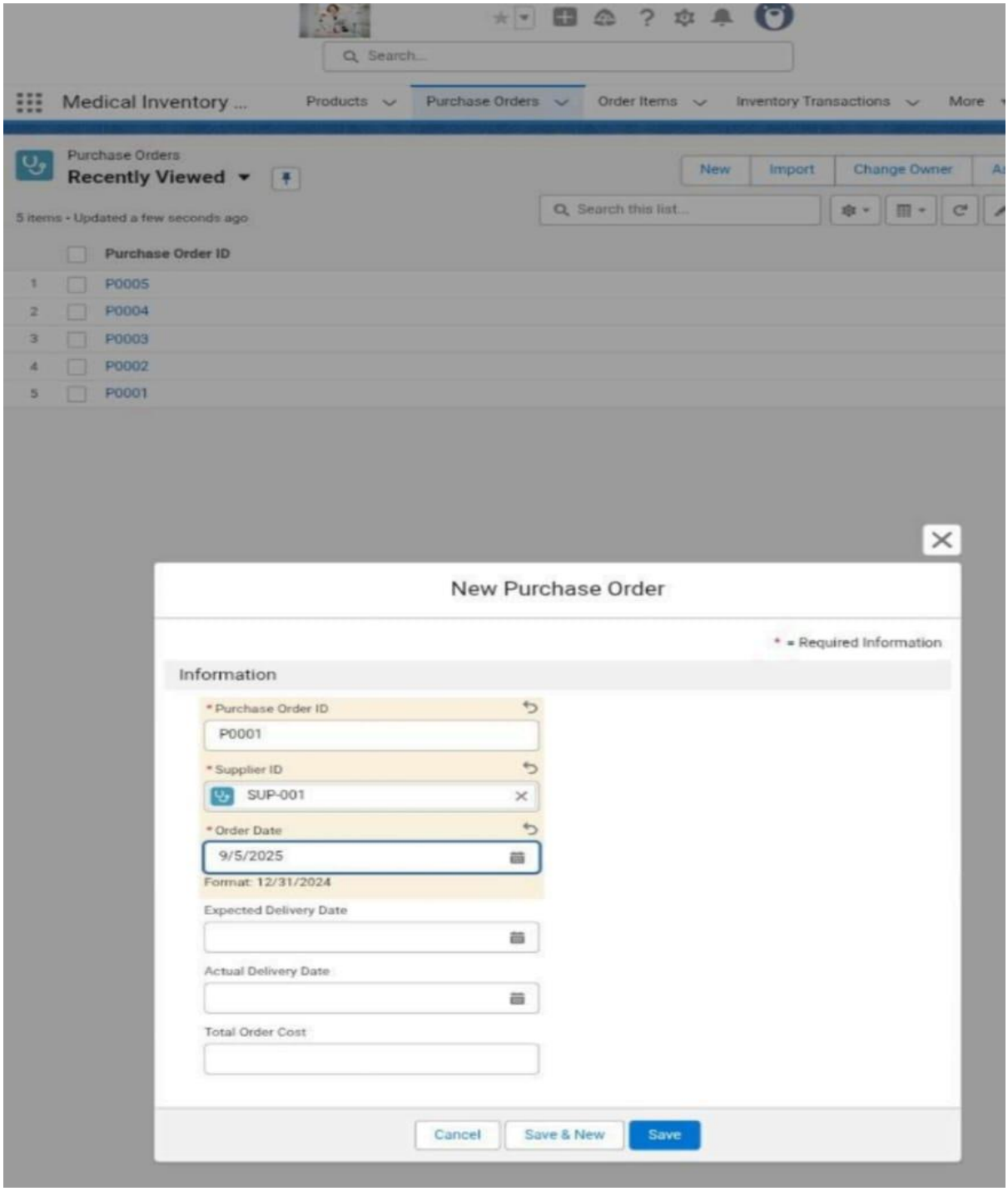
Simulated user errors such as entering invalid quantity or cost.

Checked dashboard visualizations with multiple test purchase orders.

4. RESULTS

Output screenshots (to be inserted):

Report: Purchase Orders by Supplier.



5. ADVANTAGES & DISADVANTAGES

Advantages

1) Centralized tracking of medical supplies.

2) Automated calculation of total costs.

3) Supplier-based analysis for better procurement decisions.

4) Improved accuracy through validations.

Disadvantages

1) Initial setup requires technical knowledge.

2) Dependent on Salesforce access and customization.

6. CONCLUSION

The Medical Inventory Management System successfully provides an organized solution for managing medical supplies. By automating purchase orders, inventory tracking, and report generation, it reduces manual errors, improves transparency, and enhances efficiency for healthcare organizations.

7. APPENDIX

Sample Code Snippet – Apex Trigger for Total Order Cost

```
trigger CalculateTotalCost on Order_Item__c (after insert, after update, after delete) {

Map<Id, Decimal> purchaseToUpdateMap = new Map<Id, Decimal> ();

If (Trigger.isInsert || Trigger.isUpdate) {

for (Order_Item__citem: Trigger.new) {

purchaseToUpdateMap.put(item.Purchase_Order__c,
```

```
purchaseToUpdateMap.get(item.Purchase_Order__c) == null?
item.Quantity__c * item.Price__c:
purchaseToUpdateMap.get(item.Purchase_Order__c) +
(item.Quantity__c * item.Price__c));
}
}

If (Trigger.isDelete) {
for (Order_Item__c item: Trigger.old) {
purchaseToUpdateMap.put(item.Purchase_Order__c,
purchaseToUpdateMap.get(item.Purchase_Order__c) == null?
0 - (item.Quantity__c * item.Price__c):
purchaseToUpdateMap.get(item.Purchase_Order__c) -
(item.Quantity__c * item.Price__c));
}
}
List<Purchase_Order__c> updateList = new List<Purchase_Order__c>
();
For (Id poId : purchaseToUpdateMap.keySet()) {
updateList.add(new Purchase_Order__c(Id = poId,
Total_Order_Cost__c = purchaseToUpdateMap.get(poId)));
}
update update List;
}
```