

MEDICAL MANAGEMENT

COLLEGE NAME : C.B.M COLLEGE
COLLEGE CODE : bru01

INVENTORY

TEAM ID : NM2025TMID24899

TEAM MEMBERS :

Team LeaderName: KAVUSALYA. S.R

Email: kavusisahana@gmail.com

Team Member1: SANDHIYA.R

Email: sandthiya2005@gmail.com

Team Member2: THILAGAVATHI.M

Email: thilagat585@gmail.com

Team Member3: Soundarya. V

Email: soundaeyasoundarya@gmail.com

INTRODUCTION:

PROJECT OVERVIEW:

This project is a comprehensive Salesforce application to streamline and manage various operational aspects of medical inventory. The system aims to efficiently maintain supplier details, manage purchase orders, track product details and transactions, and monitor the expiry dates of products. Maintain detailed records of suppliers, including contact information. Catalog product information, including descriptions, stock levels. Monitor and track product expiry dates to avoid using expired items. Comprehensive reports to track supplier performance, and purchase orders.

PURPOSE:

The Medical Inventory Management System is a comprehensive Salesforce application designed to streamline and manage various operational aspects of the medical inventory. It can efficiently maintain supplier details, manage purchase orders, track product details and transactions, and monitor expiry dates of products, thereby improving operational efficiency, data accuracy, and reporting capabilities.

Project Flow:

Milestone 1 : Creation of developer account

Milestone 2 : Object Creation

Milestone 3 : Tabs

Milestone 4 : The Lightning App

Milestone 5 : Fields

Milestone 6 : Updating of Page Layouts

Milestone 7 : Compact Layouts

Milestone 8 : Validation rules

Milestone 9 : Profiles

Milestone 10 : Roles

Milestone 11 : Users

Milestone 12 : Permission Sets

Milestone 13 : Flows

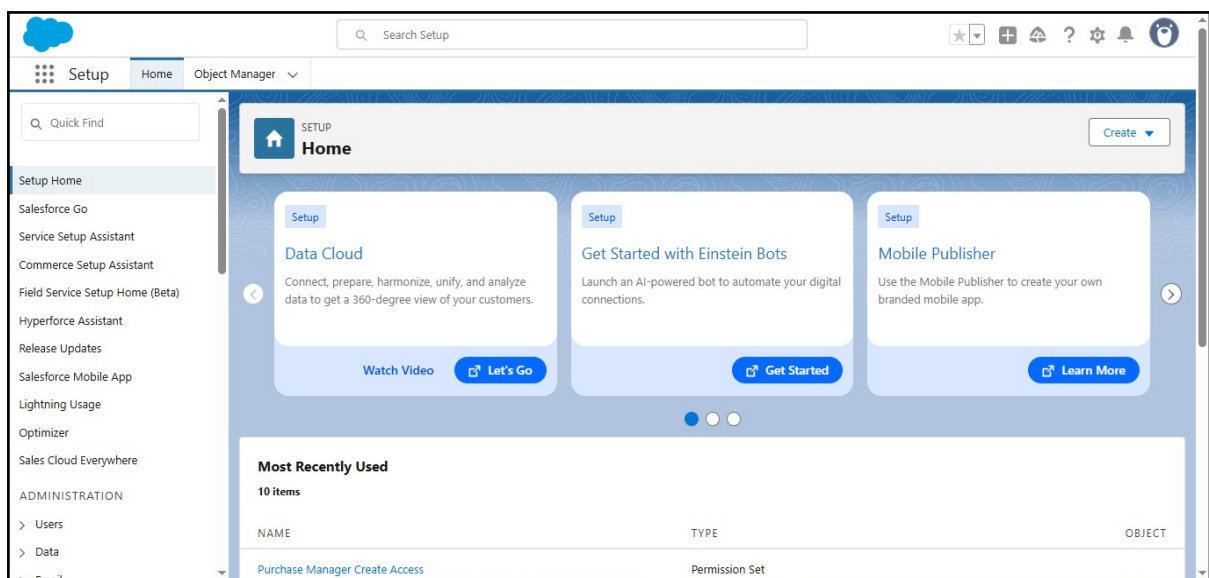
Milestone 14 : Triggers

Milestone 15 : Reports

Milestone 16 : Dashboards

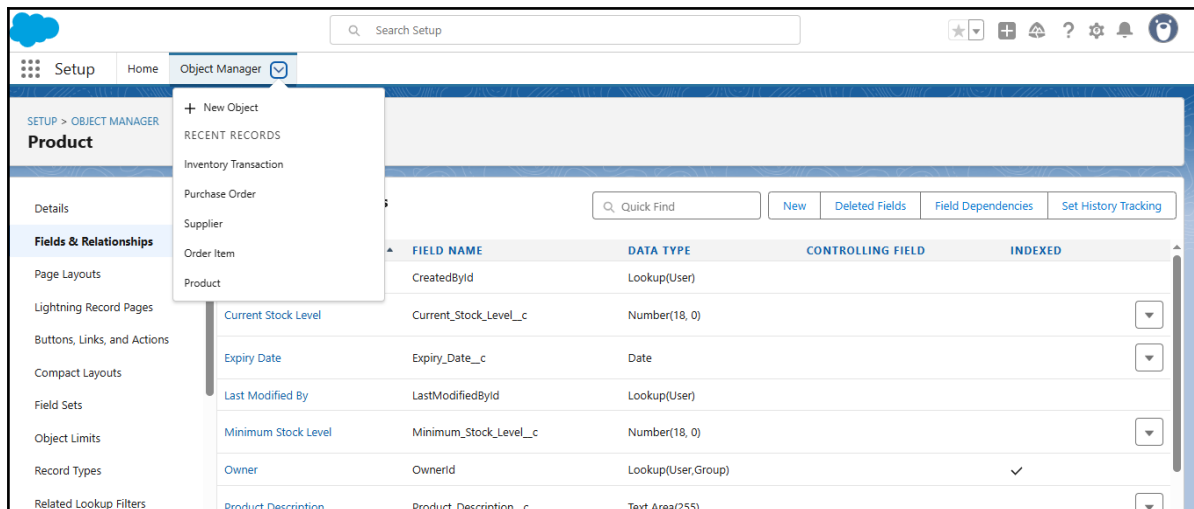
MILESTONES:

MILESTONE 1-SALESFORCE ACCOUNT:



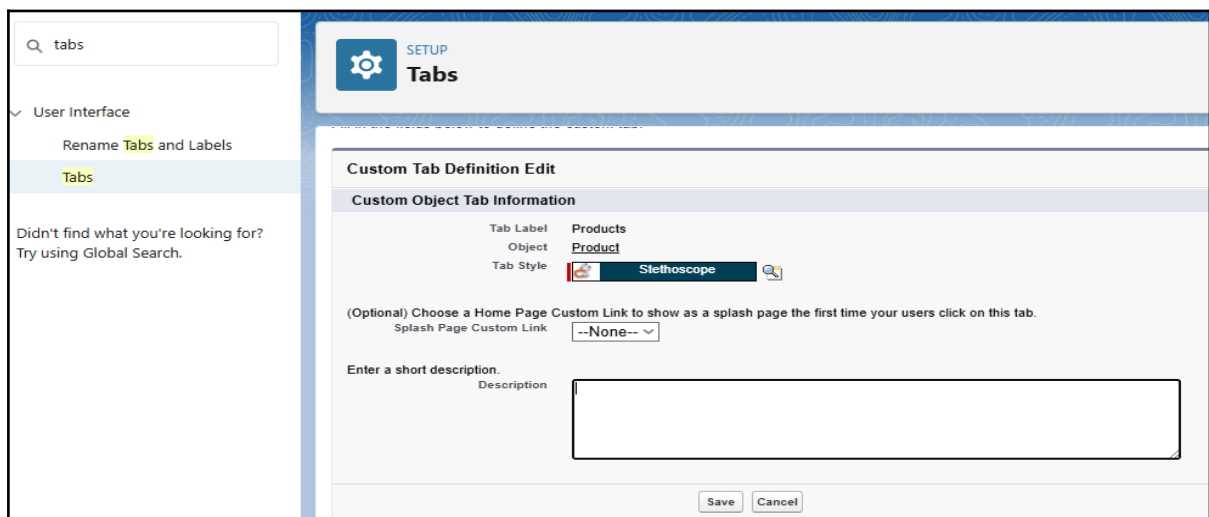
Creating a developer org in salesforce.

MILESTONE 2-OBJECTS:



Create Product, Purchase Order, Order Item, Inventory Transaction and Supplier objects.

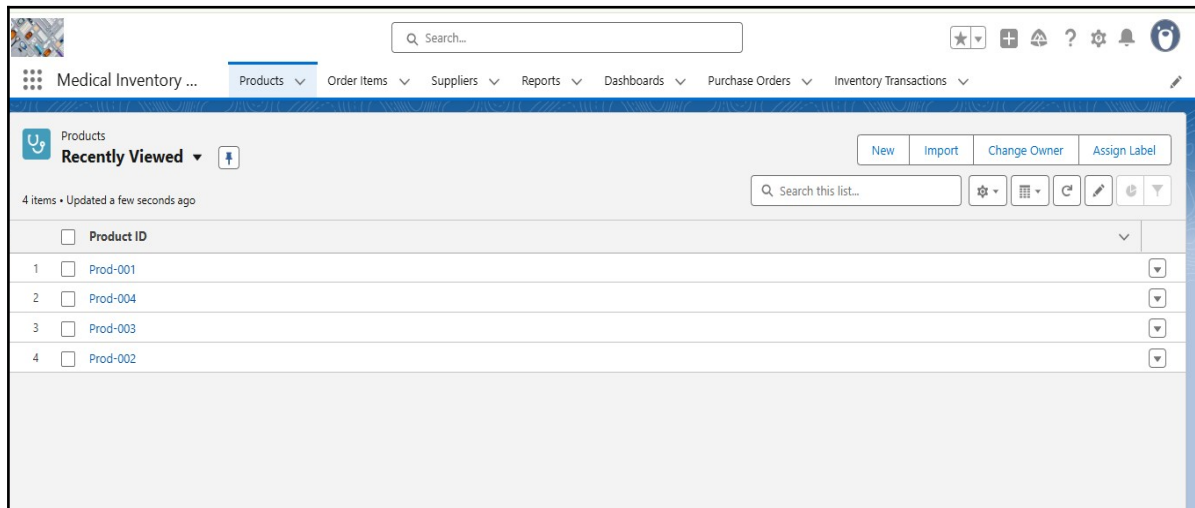
MILESTONE 3-TABS:



tabs are used to make the data stored in objects accessible to users through the user interface.

create the Tabs for the remaining Objects, they are “Product,Purchase Order, Order Item, Inventory Transaction, Supplier”.

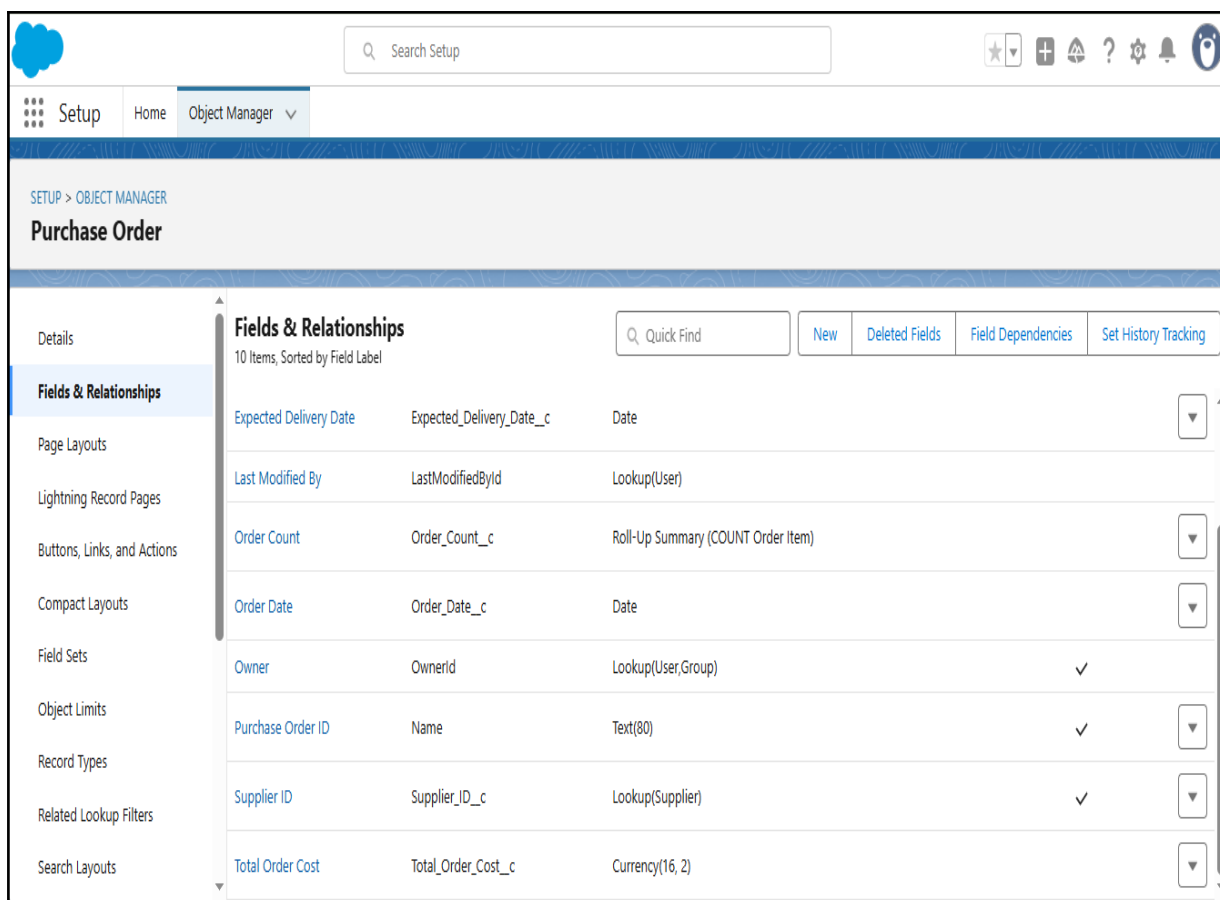
MILESTONE 4-THE LIGHTNING APP:



Built a unified Lightning App: “Medical Inventory System”

Added navigation items, utility bar, and branding

MILESTONE 5-FIELDS:

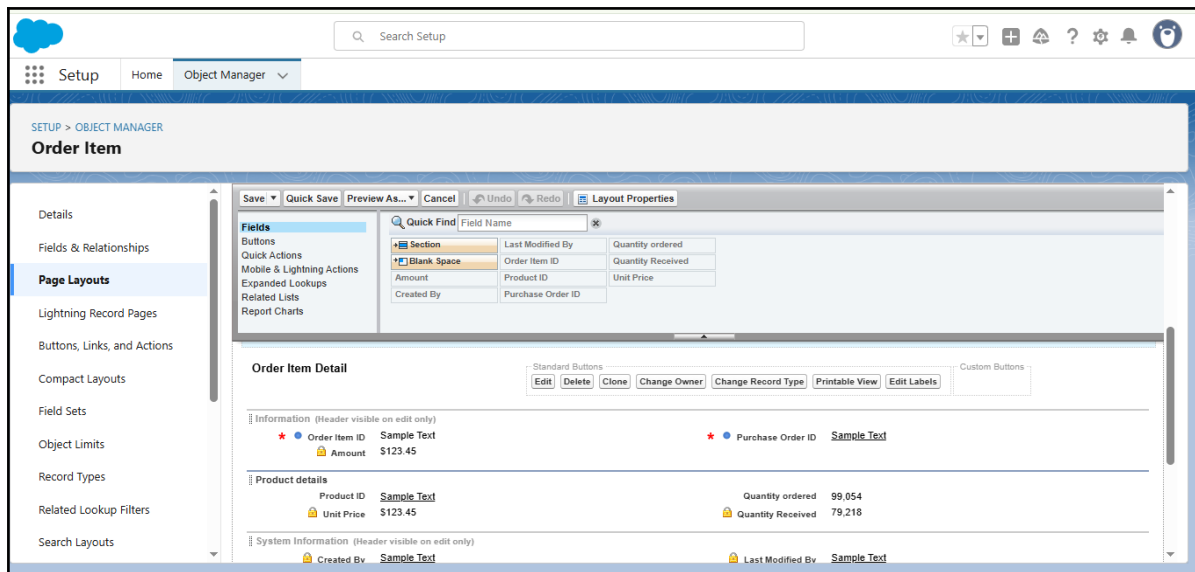


Object	Field Name	Data Type
--------	------------	-----------

Product	Product ID(Standard)	Text
	Product Name	Text
	Product Description	Text Area
	Minimum Stock Level	Number(18, 0)
	Current Stock Level	Number(18, 0)
	Unit Price	Currency(16, 2)
	Expiry Date	Date
Purchase Order	Purchase Order ID(Standard)	Text
	Supplier ID	Lookup(Supplier)
	Order Date	Date
	Expected Delivery Date	Date
	Actual Delivery Date	Date
	Order Count	Roll-Up Summary (COUNT Order Item)
	Total Order Cost	Currency(16, 2)
Order Item	Order Item ID(Standard)	Text
	Product ID	Lookup(Product)
	Purchase Order ID	Master-Detail(Purchase Order)
	Quantity Ordered	Number(18, 0)
	Quantity Received	Number(18, 0)
	Unit Price	Formula(Currency)
	Amount	Formula(Currency)
Inventory Transaction	Transaction ID(Standard)	Text
	Purchase Order ID	Lookup(Purchase Order)
	Transaction Date	Date
	Transaction Type	Picklist
	Total Order Cost	Formula(Currency)
Supplier	Supplier ID(Standard)	Text
	Supplier Name	Text
	Contact Person	Text
	Phone Number	Phone
	Email	Email
	Address	TextArea

Creating a Text Field in Product,Purchase Order,Order Item,Inventory Transaction,Supplier

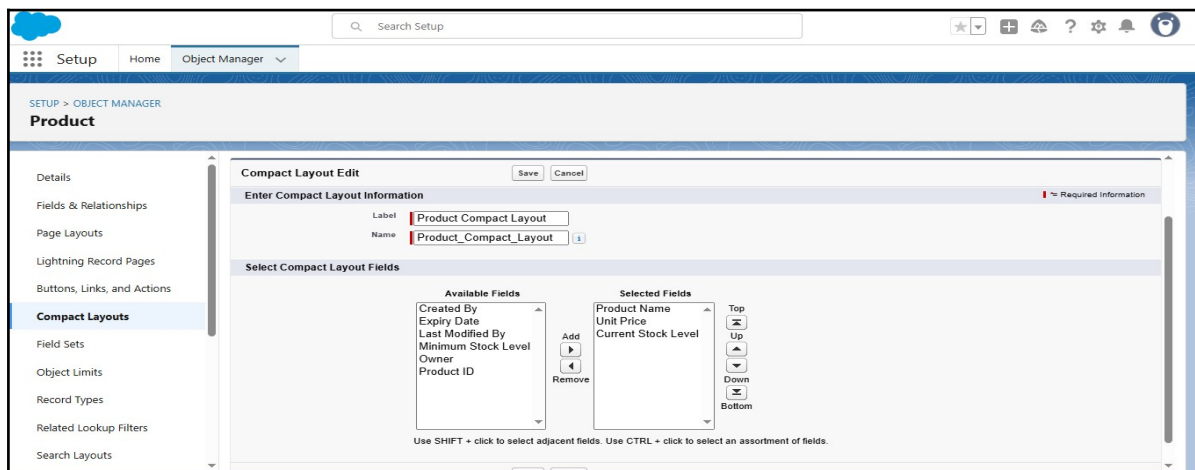
MILESTONE 6-PAGE LAYOUTS:



Page layouts in Salesforce are used to customize the organization, structure, and content of pages for viewing and editing records. They determine which fields, related lists, and custom links are visible to users, as well as the order and grouping of those elements.

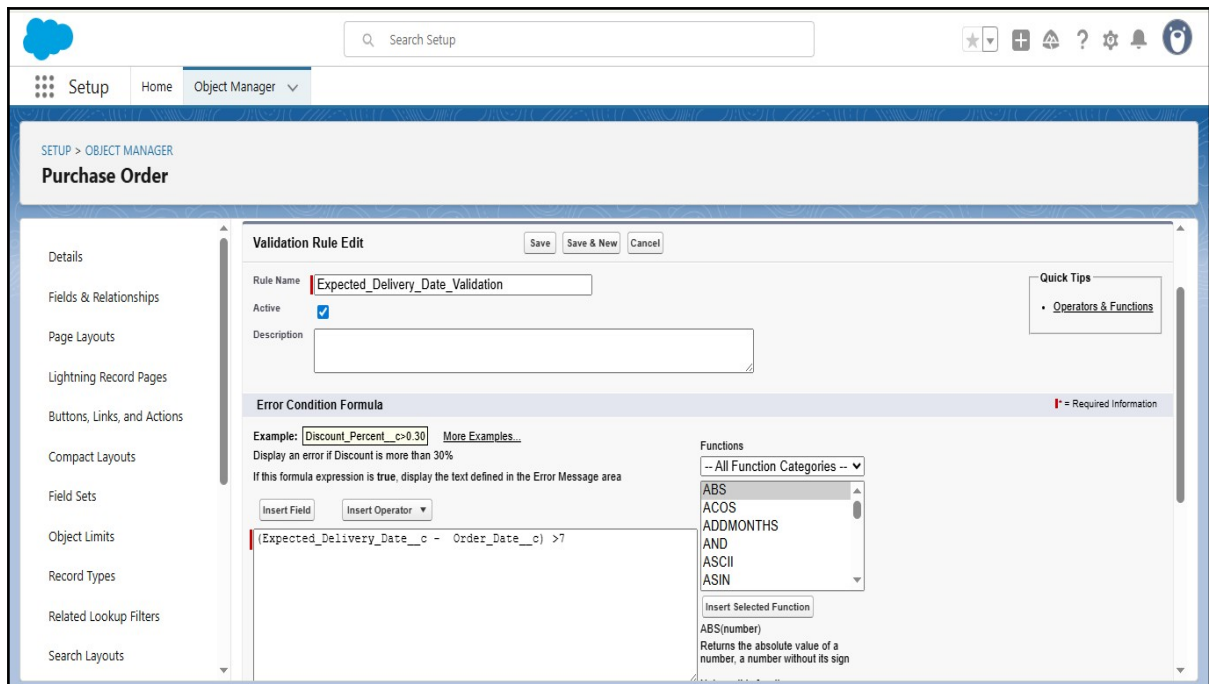
Create an page layout product,Purchase order,order item,inventory transaction,supplier .

MILESTONE 7-COMPACT LAYOUT:



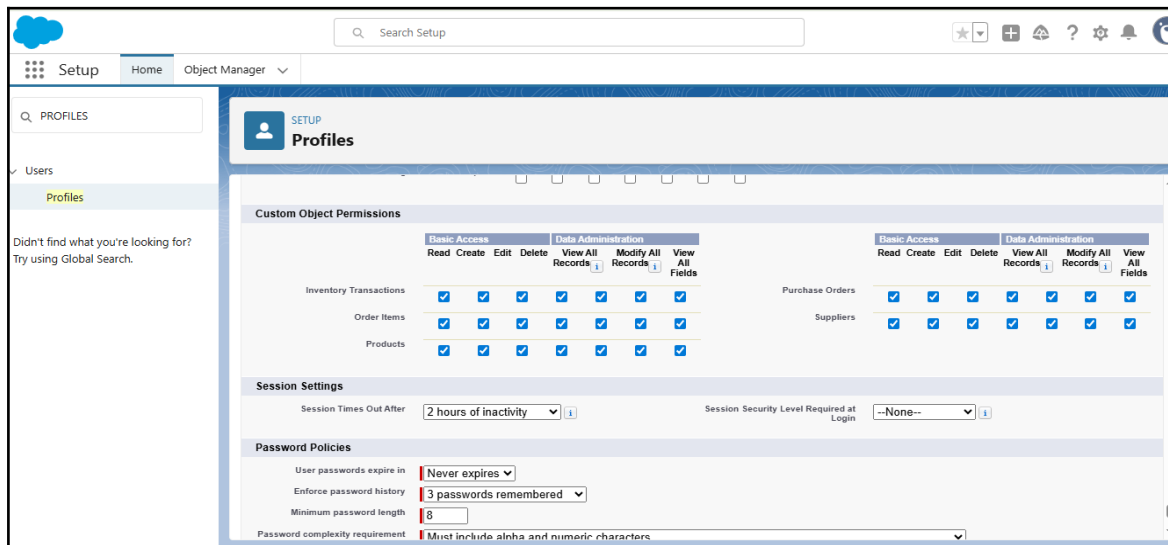
Compact layouts display a record's key fields at a glance, providing important information quickly without needing to open the record.

MILESTONE 8-VALIDATION RULES:



To create an Expected Delivery Date Validation rule to a Employee Object .

MILESTONE 9-PROFILES:

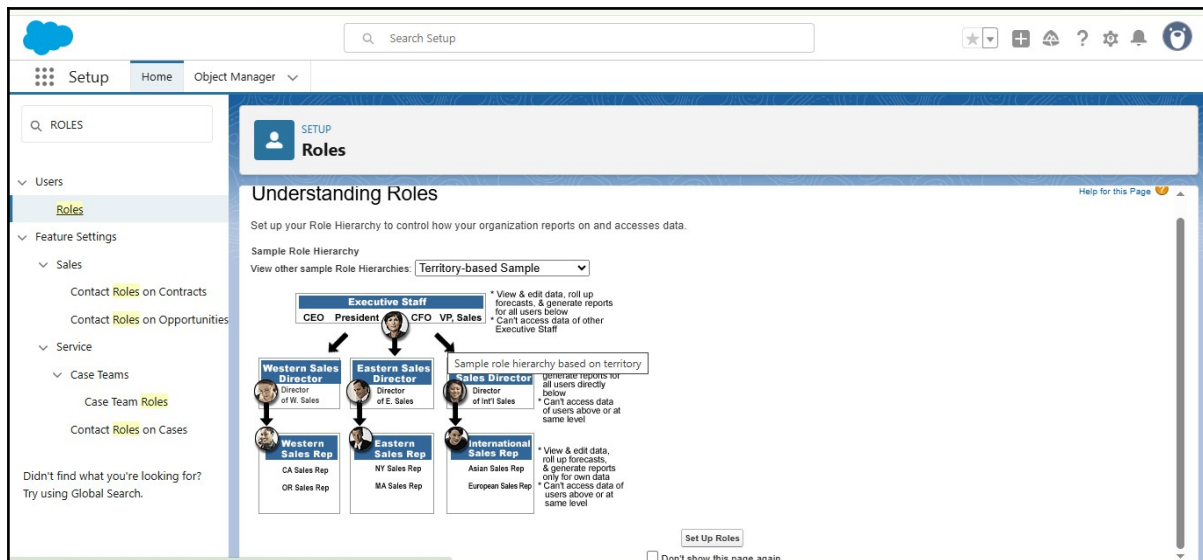


Profiles control a user's permissions to objects, fields, tabs, apps, and other settings.

To create an Inventory Manager Profile

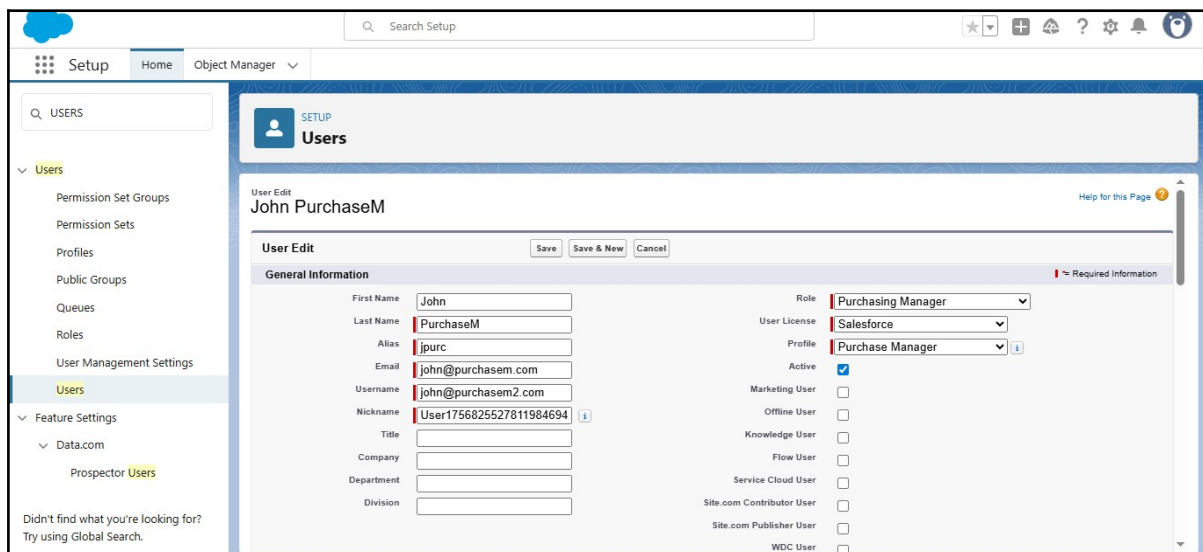
To create an Purchase Manager Profile

MILESTONE 10-ROLES:



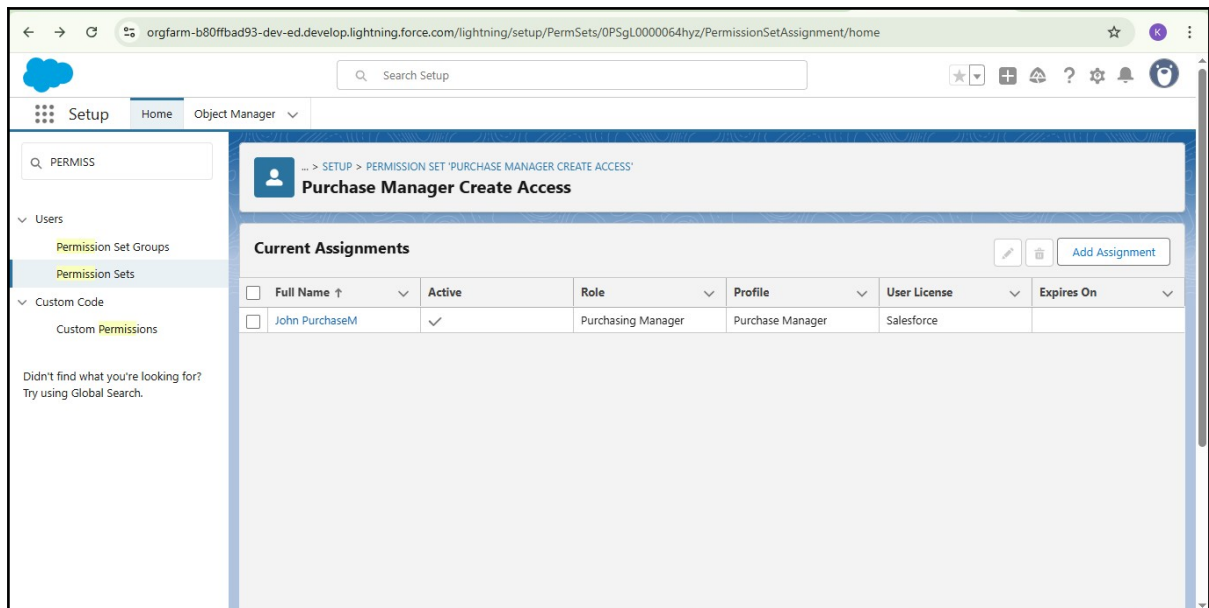
Create a Purchasing Manager Role And Inventory Manager.

MILESTONE 11-USERS:



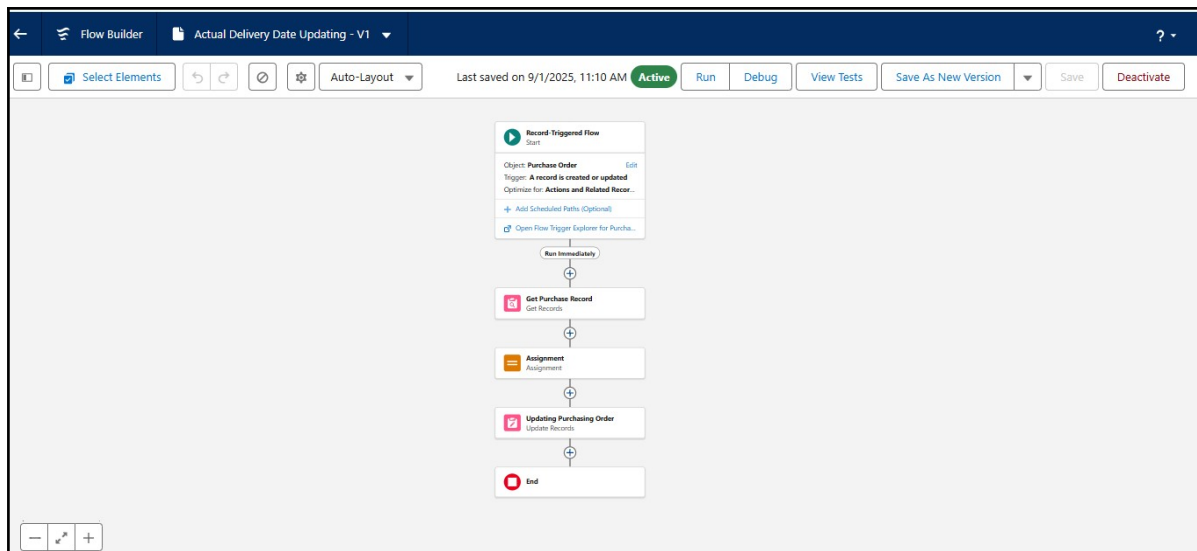
Create a users in purchasing manager.

MILESTONE 12-PERMISSION SETS:



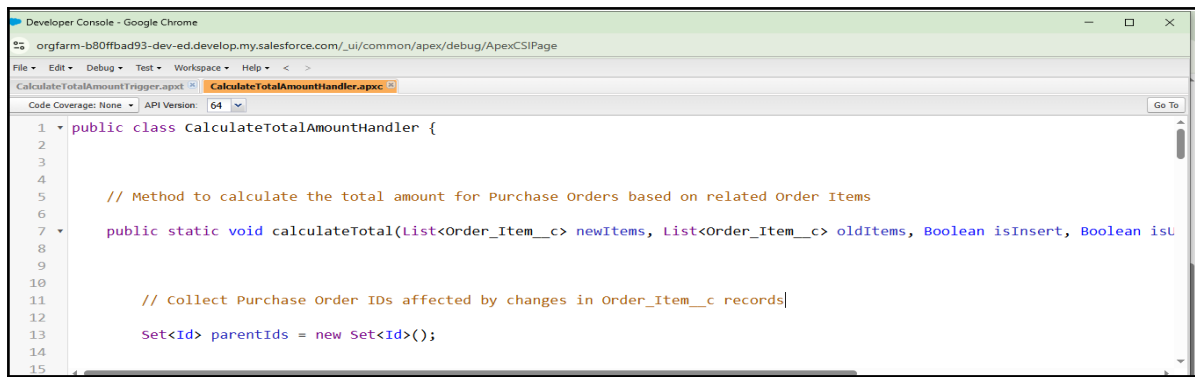
Create a Permission Set To Purchase Manager Create Access.

MILESTONE 13-FLOWS:



Create Flow to update the Actual Delivery Date.

MILESTONE 14-TRIGGERS:



Create an Apex Trigger:

trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {

// Call the handler class to handle the logic

CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);
}

Step 4:

i) In the Developer Console window, go to the top menu and click on "File".

ii) Select New: From the dropdown menu under "File", select "New".

iii) Choose Apex Class: Name it as CalculateTotalAmountHandler

public class CalculateTotalAmountHandler {

// Method to calculate the total amount for Purchase Orders based on related Order Items

public static void calculateTotal(List<Order_Item__c> newItems, List<Order_Item__c> oldItems, Boolean isInsert, Boolean isUpdate, Boolean isDelete, Boolean isUndelete) {

// Collect Purchase Order IDs affected by changes in Order_Item__c records

Set<Id> parentIdS = new Set<Id>();

// For insert, update, and undelete scenarios

if (isInsert || isUpdate || isUndelete) {

for (Order_Item__c ordItem : newItems) {

parentIdS.add(ordItem.Purchase_Order_Id__c)

```
    }  
}
```

```
// For update and delete scenarios
```

```
if (isUpdate || isDelete) {  
    for (Order_Item__c ordItem : oldItems) {  
        parentIds.add(ordItem.Purchase_Order_Id__c);  
    }  
}
```

```
// Calculate the total amounts for affected Purchase Orders
```

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

```
if (!parentIds.isEmpty()) {
```

```
    // Perform an aggregate query to sum the Amount__c for each  
    Purchase Order
```

```
    List<AggregateResult> aggrList = [  
        SELECT Purchase_Order_Id__c, SUM(Amount__c) totalAmount  
        FROM Order_Item__c  
        WHERE Purchase_Order_Id__c IN :parentIds  
        GROUP BY Purchase_Order_Id__c  
    ];
```

```
// Map the result to Purchase Order IDs
```

```
for (AggregateResult aggr : aggrList) {  
    Id purchaseOrderId = (Id)aggr.get('Purchase_Order_Id__c');  
    Decimal totalAmount = (Decimal)aggr.get('totalAmount');  
    purchaseToUpdateMap.put(purchaseOrderId, totalAmount);  
}
```

```
// Prepare Purchase Order records for update
```

```
List<Purchase_Order__c> purchaseToUpdate = new  
List<Purchase_Order__c>();
```

```

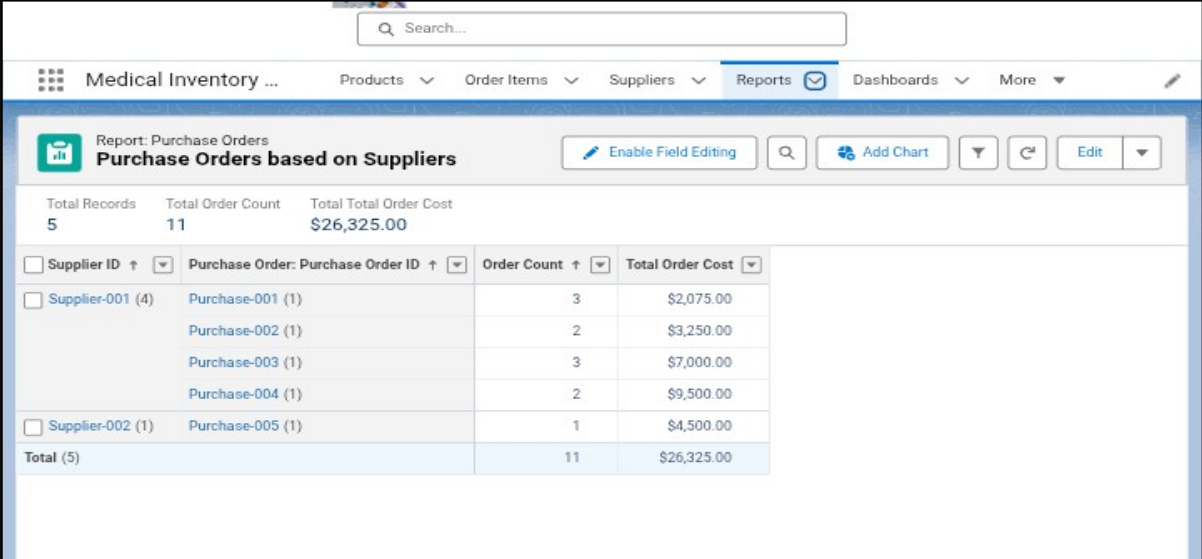
    for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {
        Purchase_Order__c purchaseOrder = new Purchase_Order__c(Id =
purchaseOrderId, Total_Order_cost__c =
purchaseToUpdateMap.get(purchaseOrderId));
        purchaseToUpdate.add(purchaseOrder);
    }

    // Update Purchase Orders if there are any changes
    if (!purchaseToUpdate.isEmpty()) {
        update purchaseToUpdate;
    }
}
}
}

```

Save it.

MILESTONE 15-REPORTS:



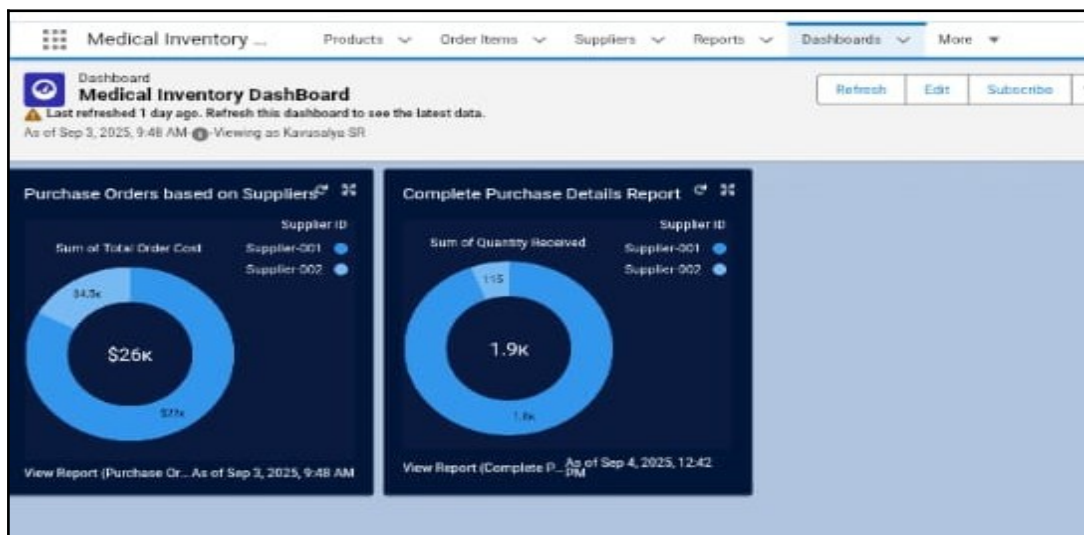
The screenshot shows a Salesforce report interface. At the top, there's a search bar and navigation tabs for 'Medical Inventory ...', 'Products', 'Order Items', 'Suppliers', 'Reports' (selected), 'Dashboards', and 'More'. Below the tabs, the report title is 'Report: Purchase Orders' and 'Purchase Orders based on Suppliers'. There are buttons for 'Enable Field Editing', 'Add Chart', and 'Edit'. Summary statistics show: Total Records: 5, Total Order Count: 11, Total Total Order Cost: \$26,325.00. The main table has columns: Supplier ID, Purchase Order: Purchase Order ID, Order Count, and Total Order Cost. It lists 5 purchase orders grouped by supplier, with a total row at the bottom.

Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
Supplier-001 (4)	Purchase-001 (1)	3	\$2,075.00
	Purchase-002 (1)	2	\$3,250.00
	Purchase-003 (1)	3	\$7,000.00
	Purchase-004 (1)	2	\$9,500.00
Supplier-002 (1)	Purchase-005 (1)	1	\$4,500.00
Total (5)		11	\$26,325.00

Report: Purchase Orders with Order Items and Product ID					
Complete Purchase Details Report					
Total Records	Total Order Count	Total Quantity Received	Total Amount		
11	11	1,940	\$109,350.00		
Supplier ID	Actual Delivery Date	Purchase Order: Purchase Order ID	Product ID: Product ID	Order Count	Product ID: Product Name
Supplier-001 (10)	9/20/2025 (3)	Purchase-001 (3)	Prod-001	3	Paracetamol 500mg
			Prod-002	3	Caught syrah 100ml
			Prod-001	3	Paracetamol 500mg
		Subtotal		3	
				3	
	9/21/2025 (2)	Purchase-002 (2)	Prod-004	2	stepstils 200
			Prod-003	2	Dolo 150
		Subtotal		2	
				2	
	9/22/2025 (3)	Purchase-003 (3)	Prod-004	3	stepstils 200
			Prod-003	3	Dolo 150
			Prod-003	3	Dolo 150
		Subtotal		3	
				3	
	9/23/2025 (2)	Purchase-004 (2)	Prod-001	2	Paracetamol 500mg
			Prod-004	2	stepstils 200
		Subtotal		2	
				2	
Subtotal				10	
Supplier-002 (1)	9/24/2025 (1)	Purchase-005 (1)	Prod-003	1	Dolo 150
		Subtotal		1	
				1	
Subtotal				1	
Total (11)				11	

Create a Purchase Orders based on Suppliers(Summary) Report.

MILESTONE 16-DASHBOARDS:



They are powerful tools for monitoring real-time data, tracking progress towards goals, and gaining actionable insights at a glance. Dashboards consist of components such as charts, tables, metrics, and gauges that display data from underlying reports.

CONCLUSION:

The Medical Inventory Management System project successfully demonstrates the design and implementation of a comprehensive, real-time Salesforce application tailored to the specific operational needs of medical inventory management. Through this project, we streamlined the tracking and control of essential inventory components such as supplier information, purchase orders, product details, transactions, and product expiry dates.