

# MEDICAL INVENTORY

**COLLEGE NAME: SHRI NEHRU MAHA VIDYALAYA COLLEGE OF ARTS AND SCIENCE**  
**COLLEGE CODE: BRU26**

**TEAM ID:** NM2025TMID20755

## **TEAM MEMBERS:**

Team Leader Name: **DURAI MURUGAN S**

**Email:** duraimurugan476@gmail.com

Team Member: **ARUL RAJ S**

**Email:** arulrajsarul57@gmail.com

Team Member: **AMULU V**

**Email:** ammukutty9760@gmail.com

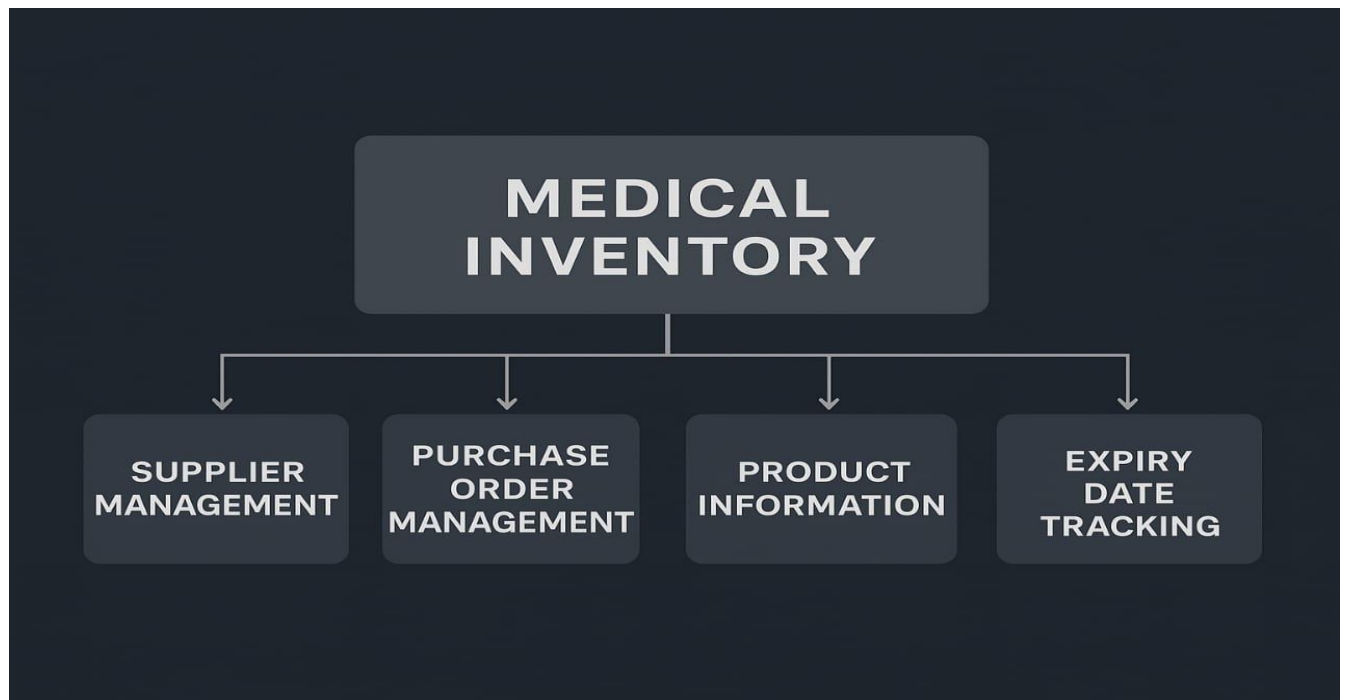
Team Member: **AKSHAYA N**

**Email:** akshayakavitha58@gmail.com

# 1.INTRODUCTION

## 1.1 Project Overview

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.



## 1.2 Purpose

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.

## DEVELOPMENT PHASE

### Creating Developer Account:

By using this URL - <https://developer.salesforce.com/signup>

Developer Edition with Agentforce

salesforce

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name Last name

Dural Murugan S

Job title Work email

student duralmurugan476@gmail.com

Company Country/Region

snmv cas India

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud Infrastructure.

☒ I agree to the Main Services Agreement - Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

### OBJECT:

Created Objects:

- ✓ **Product**
- ✓ **Purchase Order**
- ✓ **Order Item**
- ✓ **Inventory Transaction**

# ✓ Supplier

Student | Home | Salesforce | Product | Salesforce

orgfarm-b9f212a963-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL0000025zrl/Details/view

Search Setup

Setup | Home | Object Manager

SETUP > OBJECT MANAGER

### Product

**Details** [Edit] [Delete]

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

Description

API Name  
Product\_\_c

Custom  
✓

Singular Label  
Product

Plural Label  
Products

Enable Reports  
✓

Track Activities

Track Field History

Deployment Status  
Deployed

Help Settings  
Standard salesforce.com Help Window

Student | Home | Salesforce | Purchase Order | Salesforce

orgfarm-b9f212a963-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL00000260H7/Details/view

Search Setup

Setup | Home | Object Manager

SETUP > OBJECT MANAGER

### Purchase Order

**Details** [Edit] [Delete]

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

Description

API Name  
Purchase\_Order\_\_c

Custom  
✓

Singular Label  
Purchase Order

Plural Label  
Purchase Orders

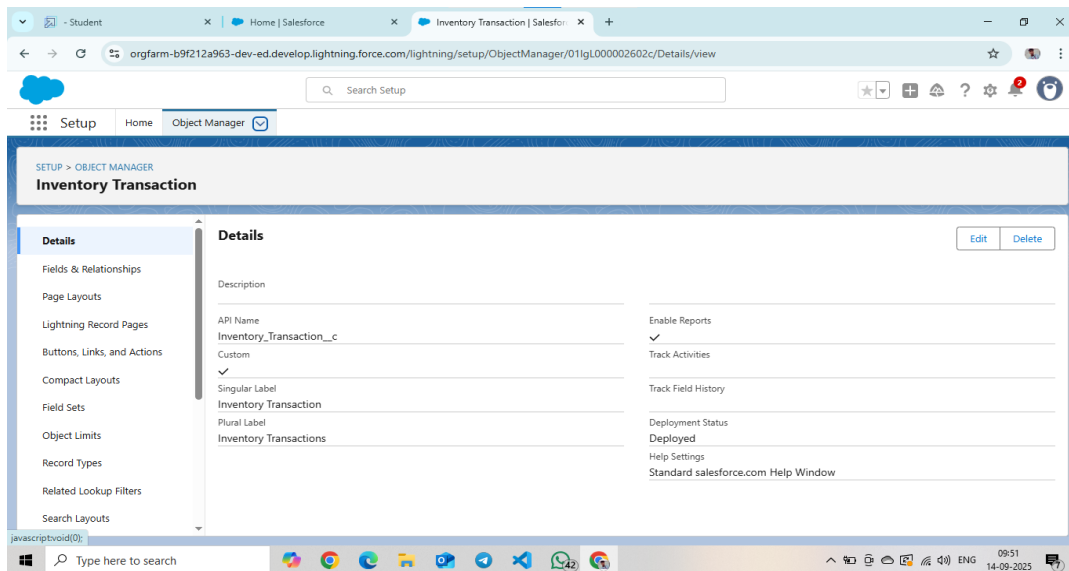
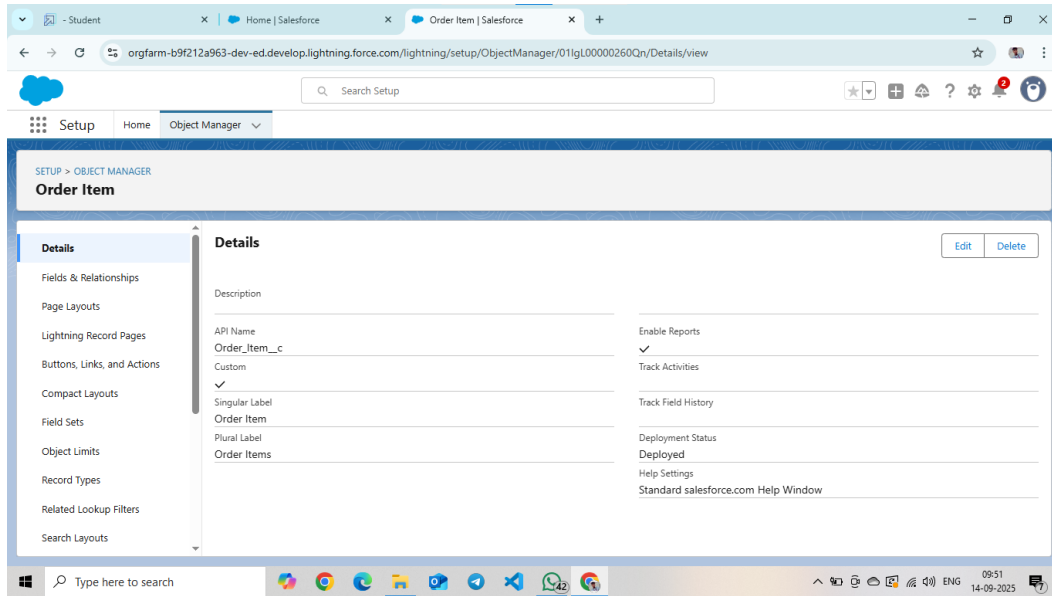
Enable Reports  
✓

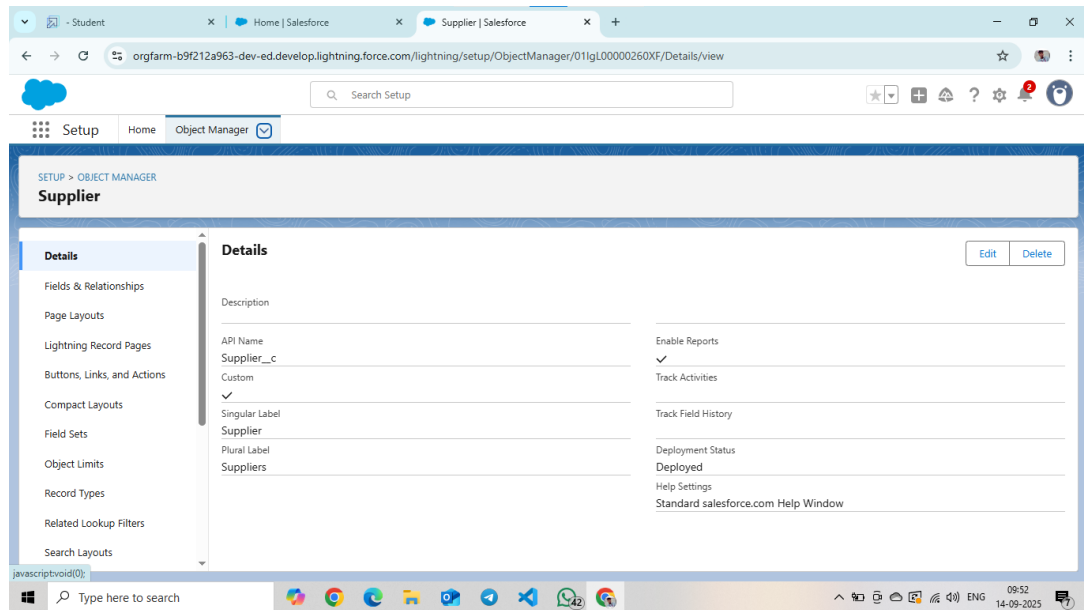
Track Activities

Track Field History

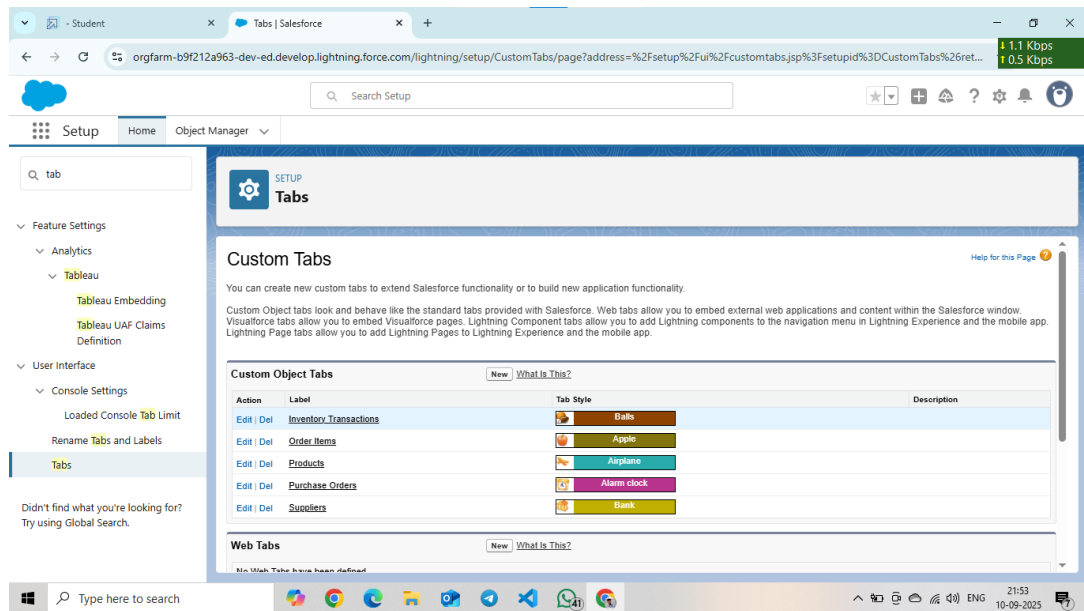
Deployment Status  
Deployed

Help Settings  
Standard salesforce.com Help Window

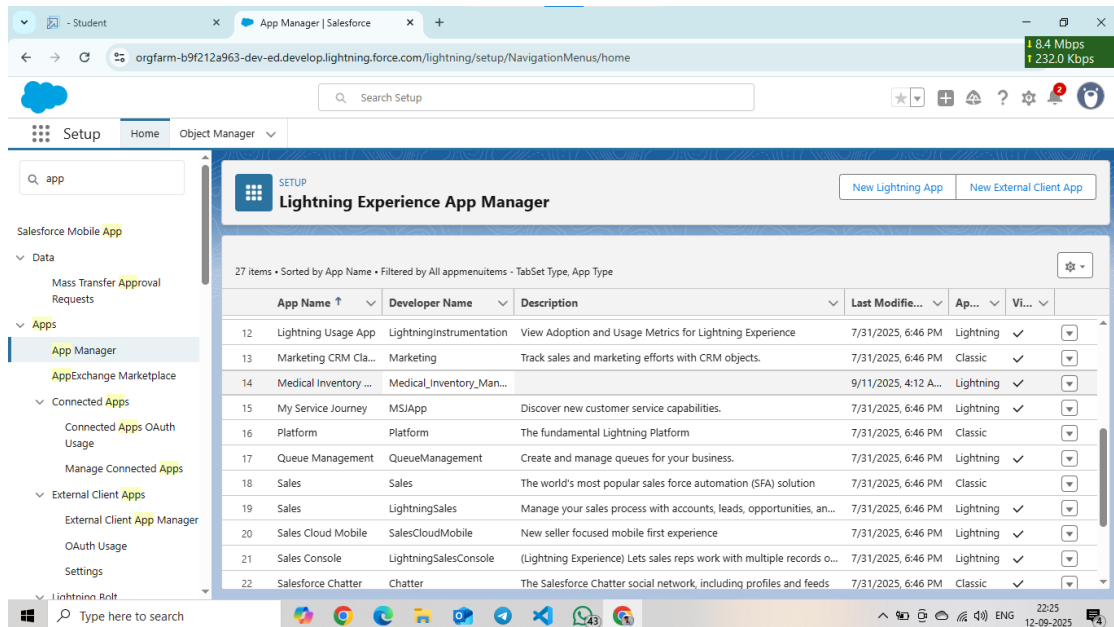




## Created Tabs

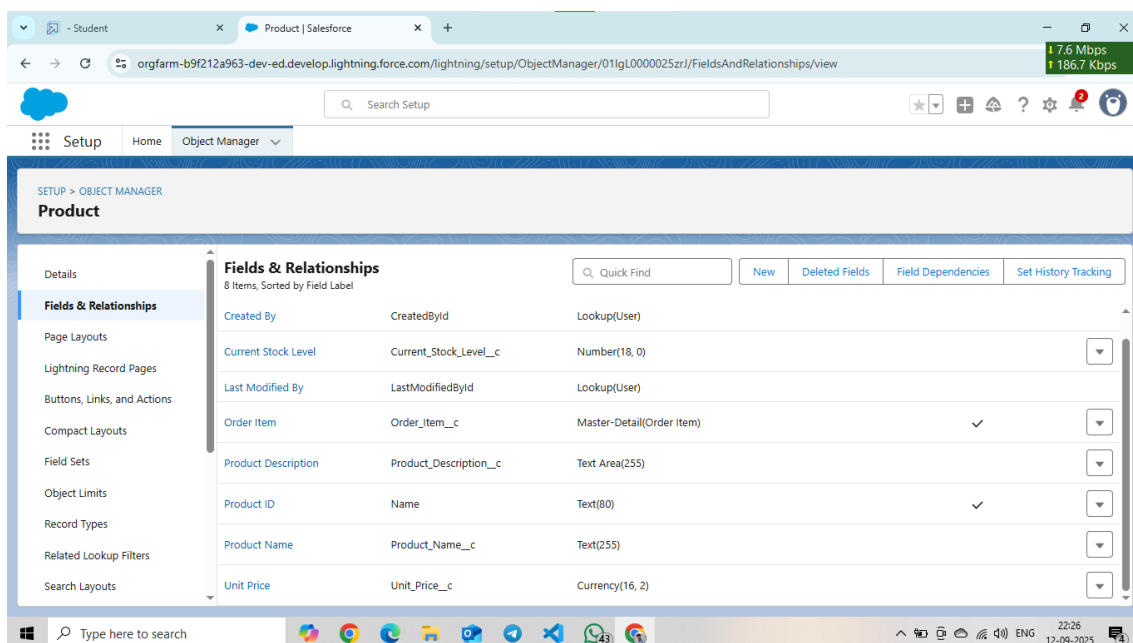


## Created a Lightning App

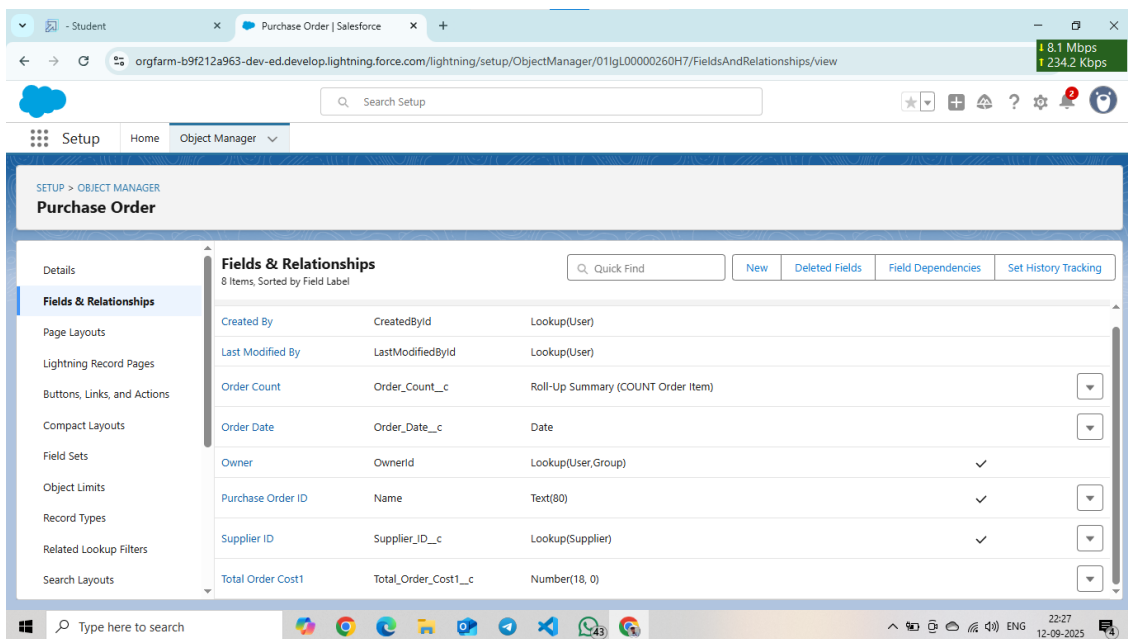


## FIELDS:

Creation of fields for the Product object:



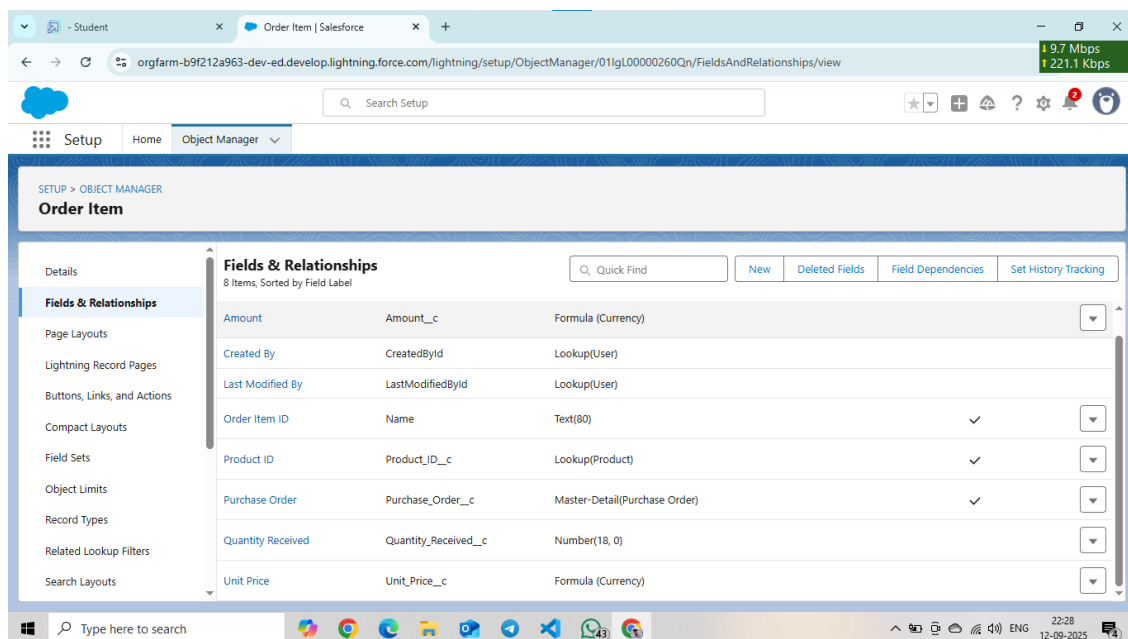
# Creation of fields for the Purchase Order object



The screenshot shows the Salesforce Setup interface for the Purchase Order object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main content area is titled 'Purchase Order' and shows the 'Fields & Relationships' section with 8 items. The fields are listed in a table with columns for Field Label, Field Name, and Field Type. The fields are: Created By (CreatedById, Lookup(User)), Last Modified By (LastModifiedById, Lookup(User)), Order Count (Order\_Count\_\_c, Roll-Up Summary (COUNT Order Item)), Order Date (Order\_Date\_\_c, Date), Owner (OwnerId, Lookup(User,Group)), Purchase Order ID (Name, Text(80)), Supplier ID (Supplier\_ID\_\_c, Lookup(Supplier)), and Total Order Cost1 (Total\_Order\_Cost1\_\_c, Number(18, 0)).

Field Label	Field Name	Field Type
Created By	CreatedById	Lookup(User)
Last Modified By	LastModifiedById	Lookup(User)
Order Count	Order_Count__c	Roll-Up Summary (COUNT Order Item)
Order Date	Order_Date__c	Date
Owner	OwnerId	Lookup(User,Group)
Purchase Order ID	Name	Text(80)
Supplier ID	Supplier_ID__c	Lookup(Supplier)
Total Order Cost1	Total_Order_Cost1__c	Number(18, 0)

# Creation of fields for the Order Item object

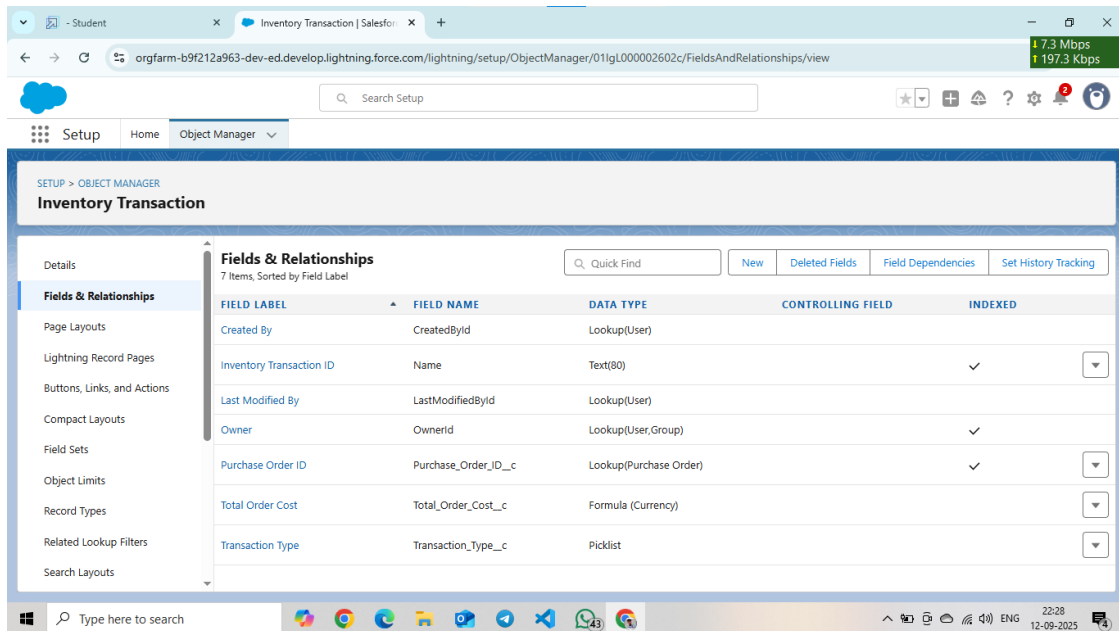


The screenshot shows the Salesforce Setup interface for the Order Item object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main content area is titled 'Order Item' and shows the 'Fields & Relationships' section with 8 items. The fields are listed in a table with columns for Field Label, Field Name, and Field Type. The fields are: Amount (Amount\_\_c, Formula (Currency)), Created By (CreatedById, Lookup(User)), Last Modified By (LastModifiedById, Lookup(User)), Order Item ID (Name, Text(80)), Product ID (Product\_ID\_\_c, Lookup(Product)), Purchase Order (Purchase\_Order\_\_c, Master-Detail(Purchase Order)), Quantity Received (Quantity\_Received\_\_c, Number(18, 0)), and Unit Price (Unit\_Price\_\_c, Formula (Currency)).

Field Label	Field Name	Field Type
Amount	Amount__c	Formula (Currency)
Created By	CreatedById	Lookup(User)
Last Modified By	LastModifiedById	Lookup(User)
Order Item ID	Name	Text(80)
Product ID	Product_ID__c	Lookup(Product)
Purchase Order	Purchase_Order__c	Master-Detail(Purchase Order)
Quantity Received	Quantity_Received__c	Number(18, 0)
Unit Price	Unit_Price__c	Formula (Currency)



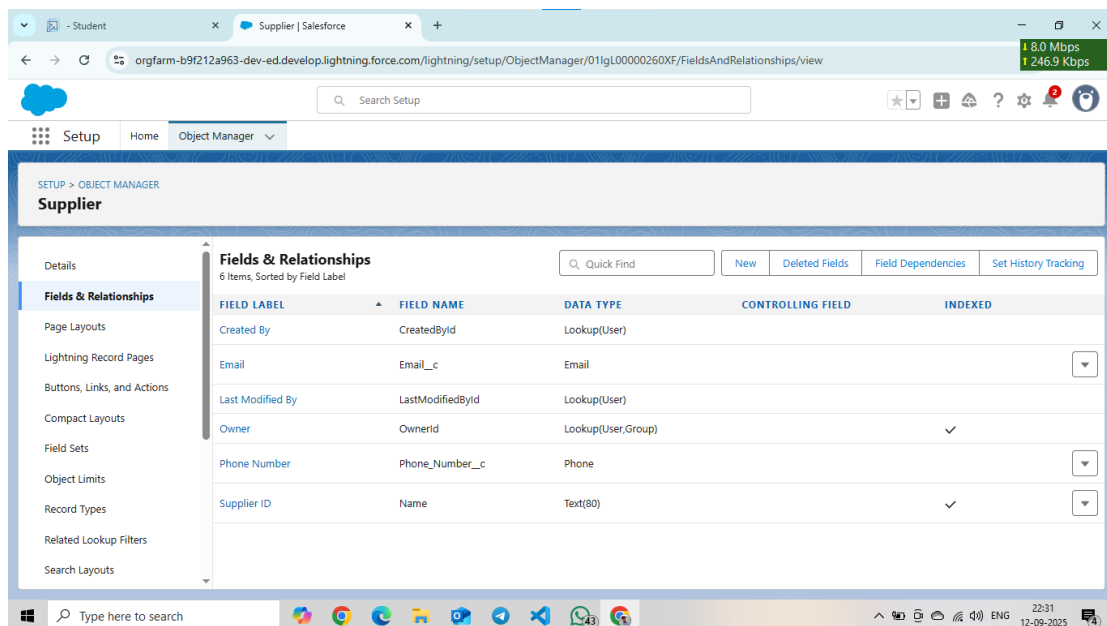
# Creation of fields for the Inventory Transaction Object



The screenshot shows the Salesforce Object Manager interface for the 'Inventory Transaction' object. The 'Fields & Relationships' section is active, displaying a table of 7 fields. The table has columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields are: Created By (Lookup(User)), Inventory Transaction ID (Text(80)), Last Modified By (Lookup(User)), Owner (Lookup(User,Group)), Purchase Order ID (Lookup(Purchase Order)), Total Order Cost (Formula (Currency)), and Transaction Type (Picklist).

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Inventory Transaction ID	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Purchase Order ID	Purchase_Order_ID__c	Lookup(Purchase Order)		✓
Total Order Cost	Total_Order_Cost__c	Formula (Currency)		
Transaction Type	Transaction_Type__c	Picklist		

# Creation of fields for the Supplier object



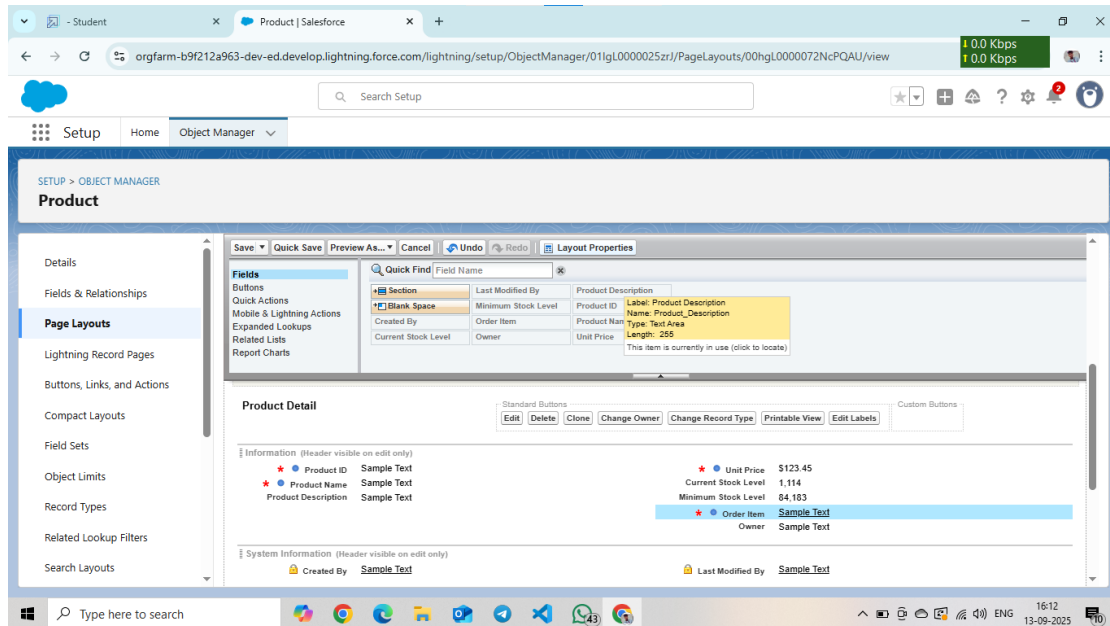
The screenshot shows the Salesforce Object Manager interface for the 'Supplier' object. The 'Fields & Relationships' section is active, displaying a table of 6 fields. The table has columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed. The fields are: Created By (Lookup(User)), Email (Email), Last Modified By (Lookup(User)), Owner (Lookup(User,Group)), Phone Number (Phone), and Supplier ID (Text(80)).

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone Number	Phone_Number__c	Phone		
Supplier ID	Name	Text(80)		✓

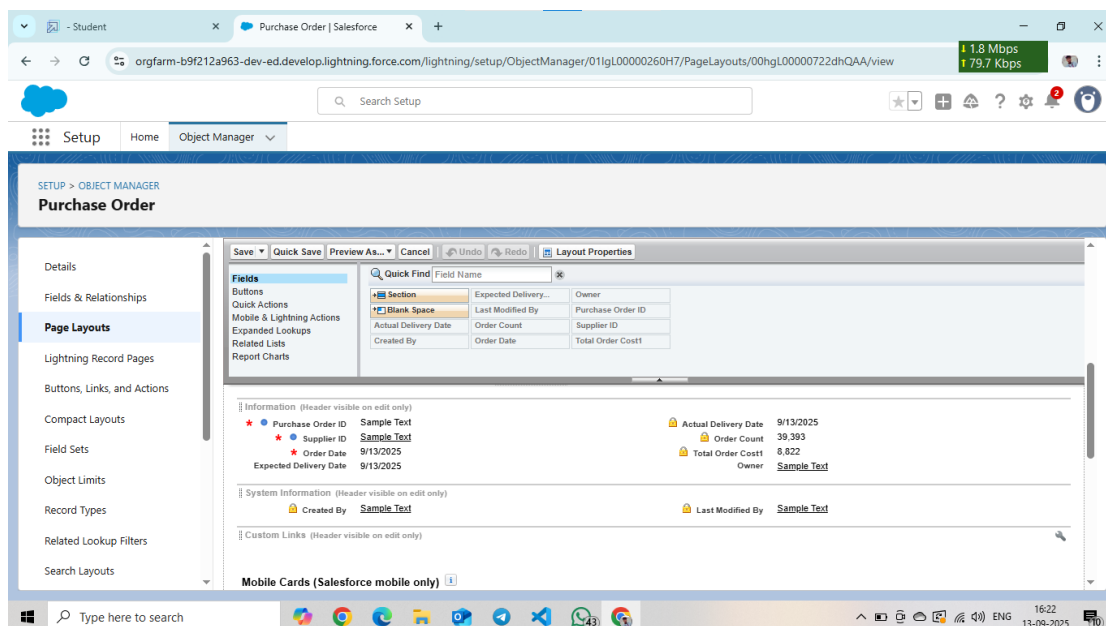
# Page Layouts;

## Editing of Page Layouts:

### Editing of Page Layout for the Product object



### Editing of Page Layout for the Purchase Order object



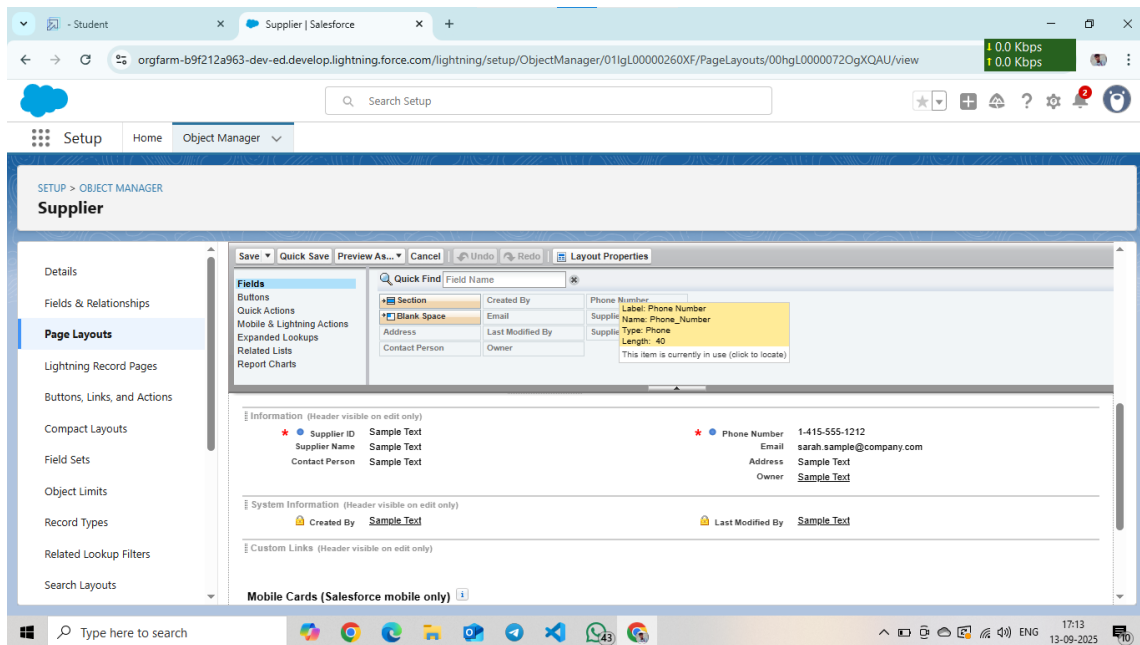
## Editing of Page Layout for the Order Item object

The screenshot shows the Salesforce Page Layout Editor for the Order Item object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts (selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main area displays the 'Order Item Detail' page layout. At the top, there are tabs for 'Save', 'Quick Save', 'Preview As...', 'Cancel', 'Undo', and 'Redo', along with a 'Layout Properties' button. Below these tabs is a 'Quick Find' search bar and a list of fields. The fields are organized into sections: 'Information (Header visible on edit only)' containing 'Order Item ID' and 'Amount'; 'Product details' containing 'Product ID' and 'Unit Price'; and 'System Information (Header visible on edit only)' containing 'Created By' and 'Last Modified By'. The right side of the editor shows the layout preview, which includes a table with columns for 'Quantity Ordered' and 'Quantity Received'. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 16:59 on 13-09-2025.

## Editing of Page Layout for the Inventory Transaction object

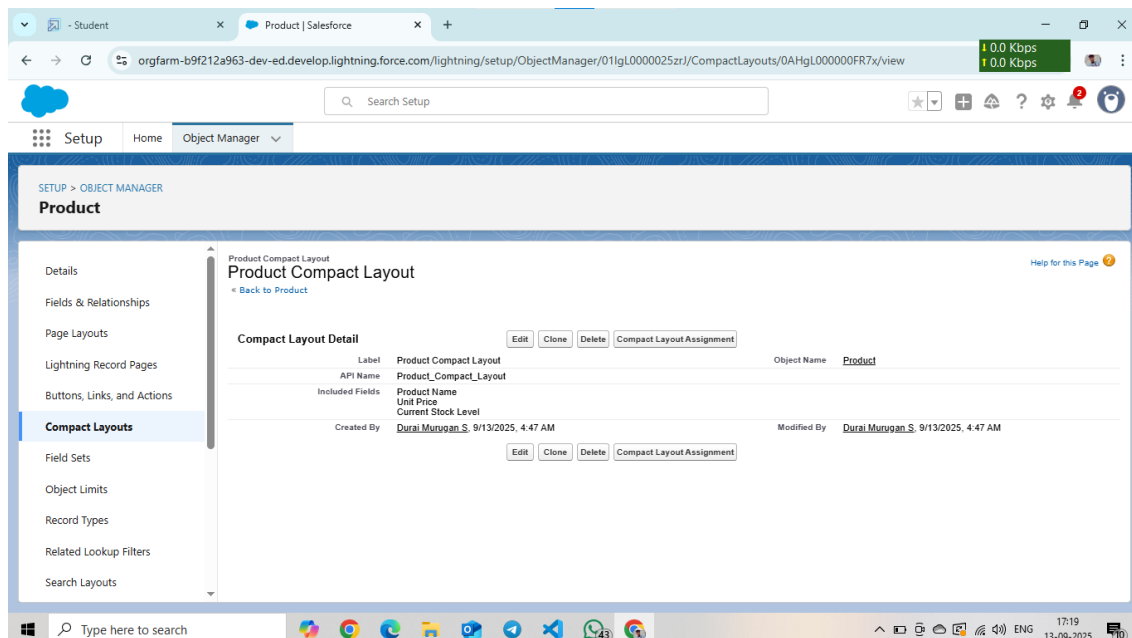
The screenshot shows the Salesforce Page Layout Editor for the Inventory Transaction object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts (selected), Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main area displays the 'Inventory Transaction' page layout. At the top, there are tabs for 'Save', 'Quick Save', 'Preview As...', 'Cancel', 'Undo', and 'Redo', along with a 'Layout Properties' button. Below these tabs is a 'Quick Find' search bar and a list of fields. The fields are organized into sections: 'Information (Header visible on edit only)' containing 'Transaction ID', 'Purchase Order ID', 'Transaction Type', and 'Total Order Cost'; 'System Information (Header visible on edit only)' containing 'Created By' and 'Last Modified By'; and 'Custom Links (Header visible on edit only)'. The right side of the editor shows the layout preview, which includes a table with columns for 'Owner', 'Transaction Date', 'Transaction ID', and 'Transaction Type'. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 17:04 on 13-09-2025.

## Editing of Page Layout for the Supplier object

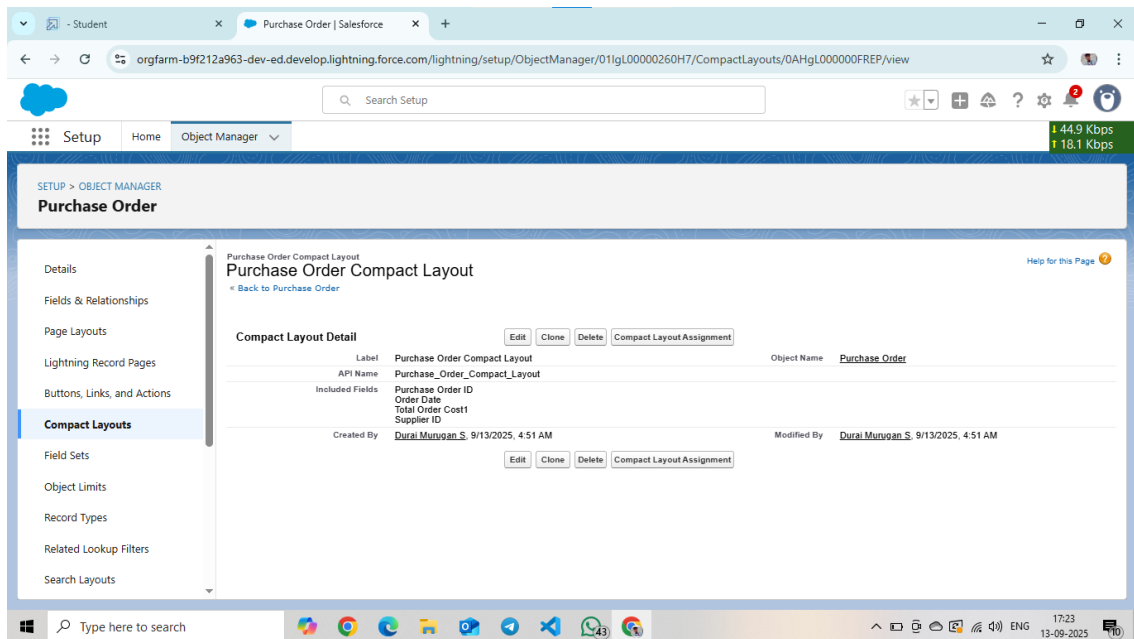


## Compact Layouts

### Created a Compact Layout for the Product Object

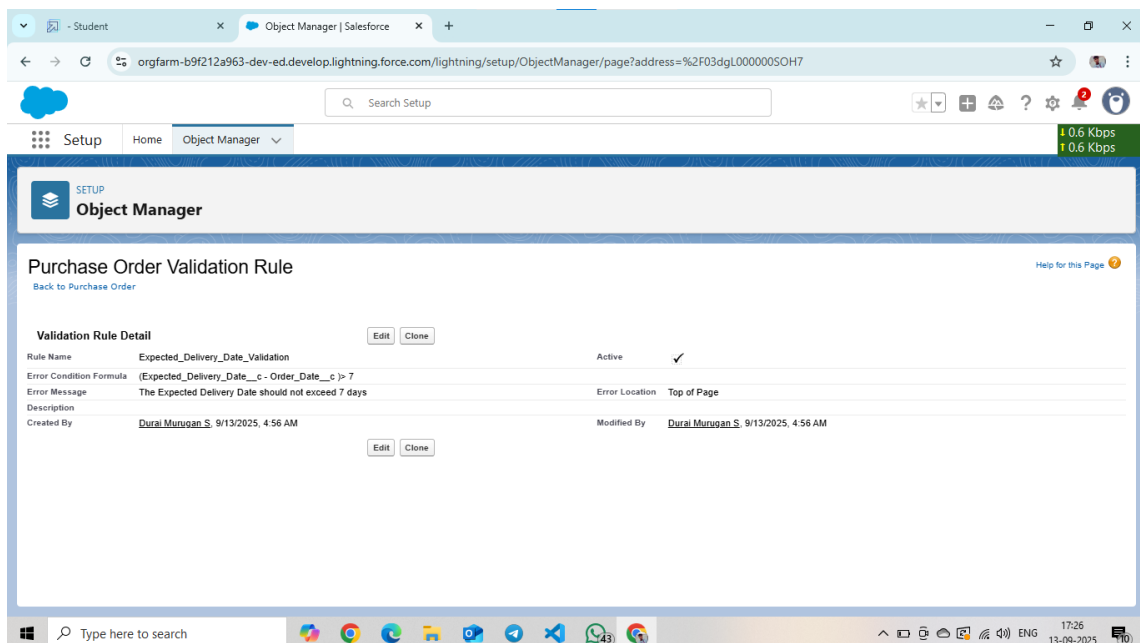


## Created a Compact Layout for the Purchase Order Object



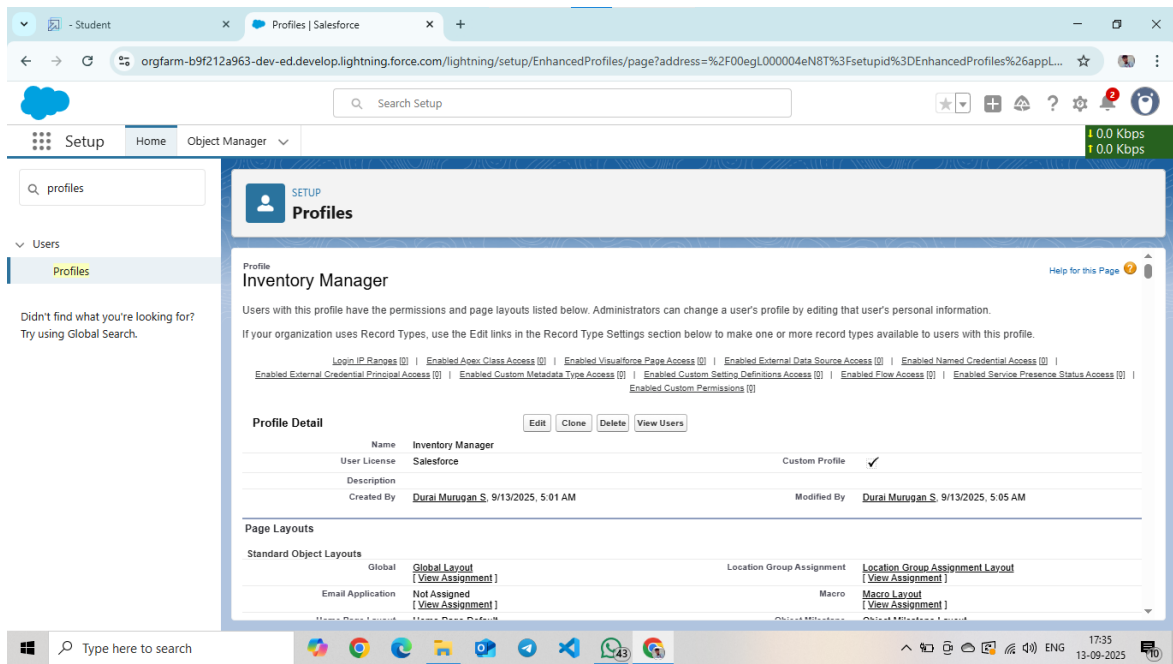
## Validation Rules

### Created an Expected Delivery Date Validation rule to an Employee Object

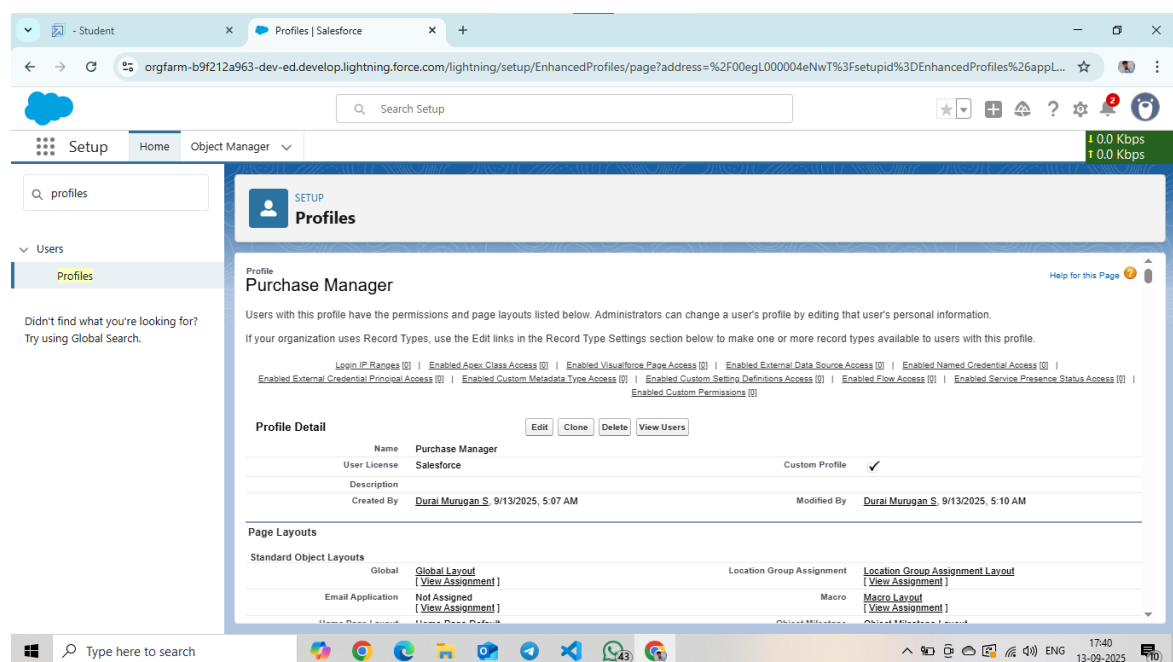


# Profiles

## Created an Inventory Manager Profile

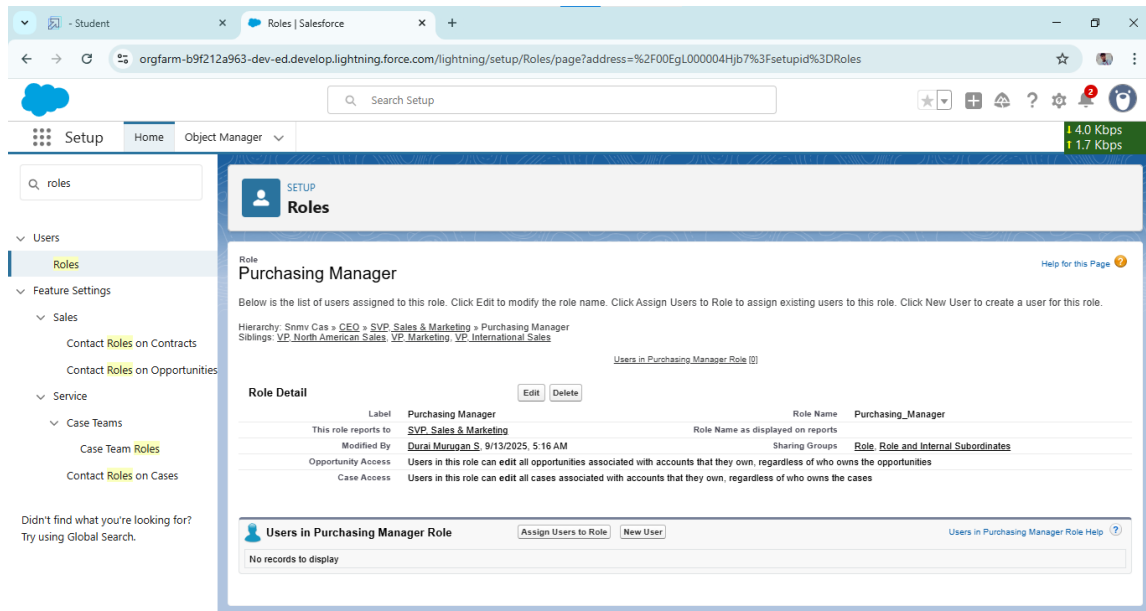


## Created a Purchase Manager Profile

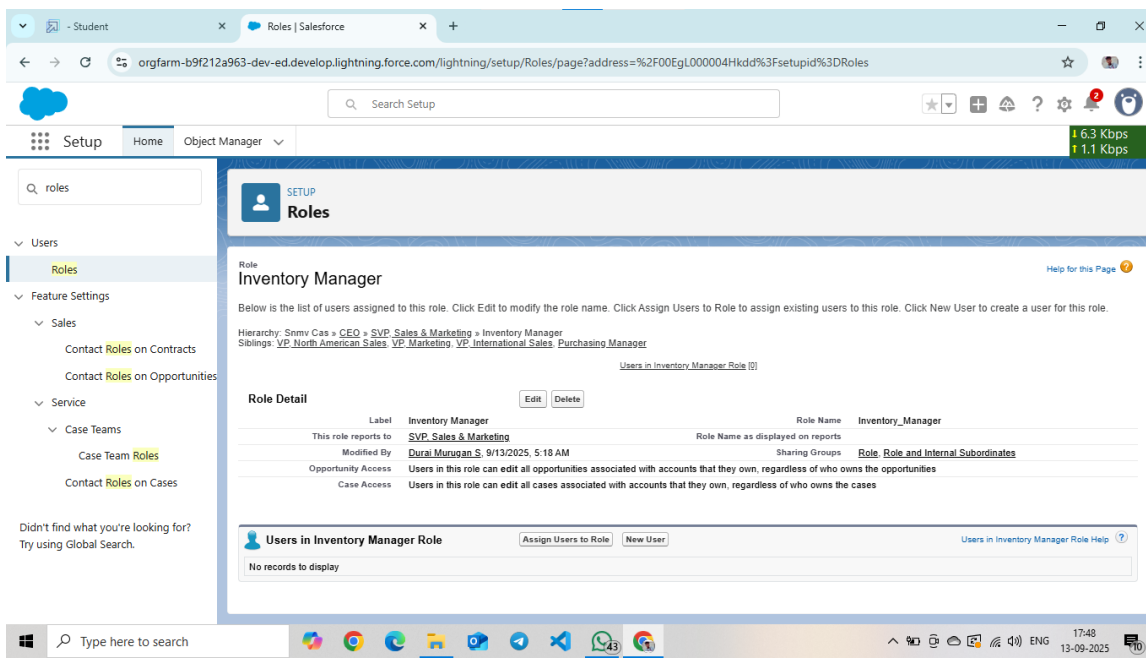


# Roles

## Created a Purchasing Manager Role (Purchasing Manager)

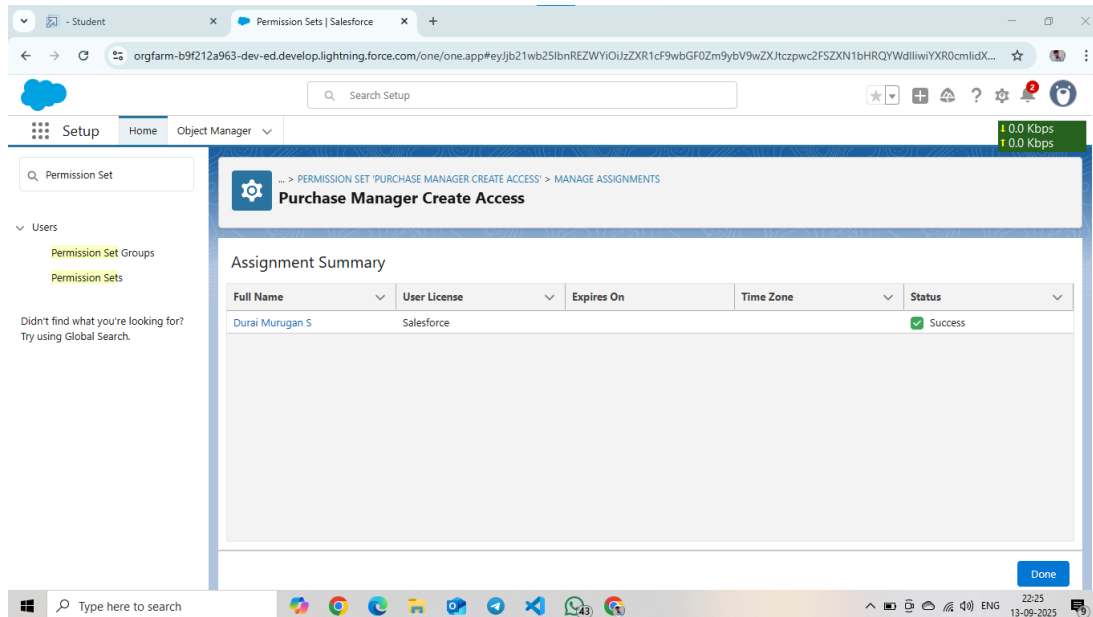


## Created a Purchasing Manager Role (Inventory Manager)



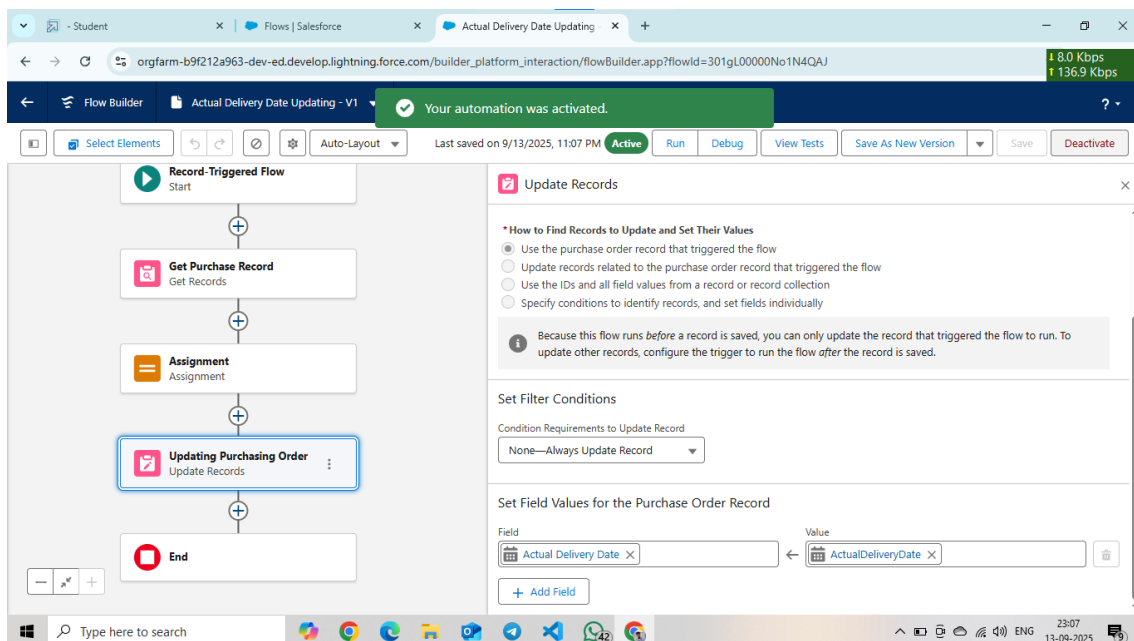
# Permission Sets

## Create a Permission Set

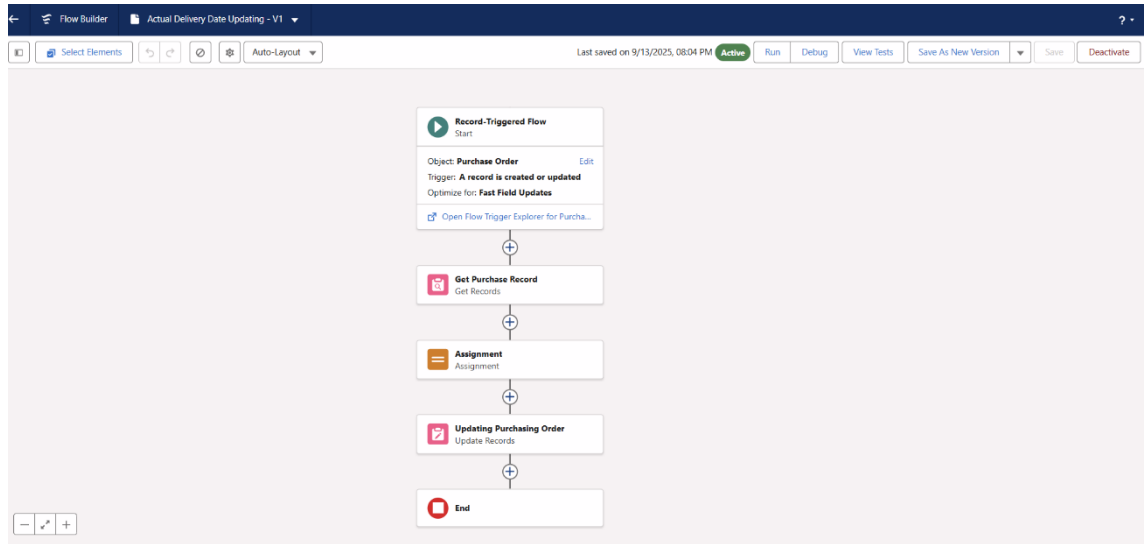


# Flows

## Created Flow to update the Actual Delivery Date







## Triggers

### Create an Apex Trigger

```
File • Edit • Debug • Test • Workspace • Help • < >
CalculateTotalAmountTrigger.apex • CalculateTotalAmountHandler.apex •
Code Coverage: None • API Version: 64 • Go To

1 trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {
2
3     // Call the handler class to handle the logic
4
5     CalculateTotalAmountHandler.calculateTotal(trigger.new, trigger.old, trigger.isInsert, trigger.isUpdate, trigger.isDelete, trigger.isUndelete);
6
7 }
8
```

## Coding

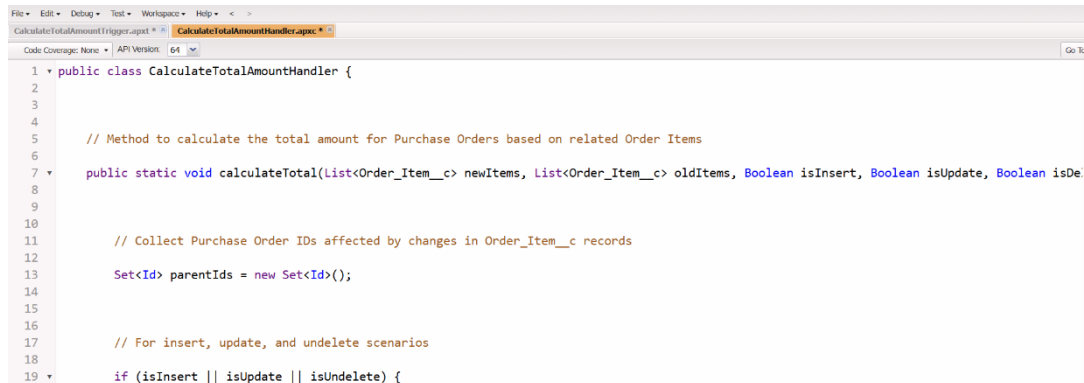
trigger CalculateTotalAmountTrigger on Order\_Item\_\_c (after insert, after update, after delete, after undelete) {

// Call the handler class to handle the logic

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

}

# Create an Apex Class



## Coding

```
public class CalculateTotalAmountHandler
```

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

```
    Set<Id> parentIds = new Set<Id>();
```

```
    if (isInsert || isUpdate || isUndelete) {
```

```
        for (Order_Item__c ordItem : newItems) {
```

```
            parentIds.add(ordItem.Purchase_Order_Id__c);
```

```
        }
```

```
    }
```

```
    if (isUpdate || isDelete) {
```

```
        for (Order_Item__c ordItem : oldItems) {
```

```
            parentIds.add(ordItem.Purchase_Order_Id__c);
```

```
        }
```

```
    }
```

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

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

```

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

];

for (AggregateResult aggr : aggrList) {

    Id purchaseOrderId = (Id)aggr.get('Purchase_Order_Id__c');

    Decimal totalAmount = (Decimal)aggr.get('totalAmount');

    purchaseToUpdateMap.put(purchaseOrderId, totalAmount);

}

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);

}

if (!purchaseToUpdate.isEmpty()) {

    update purchaseToUpdate;

}

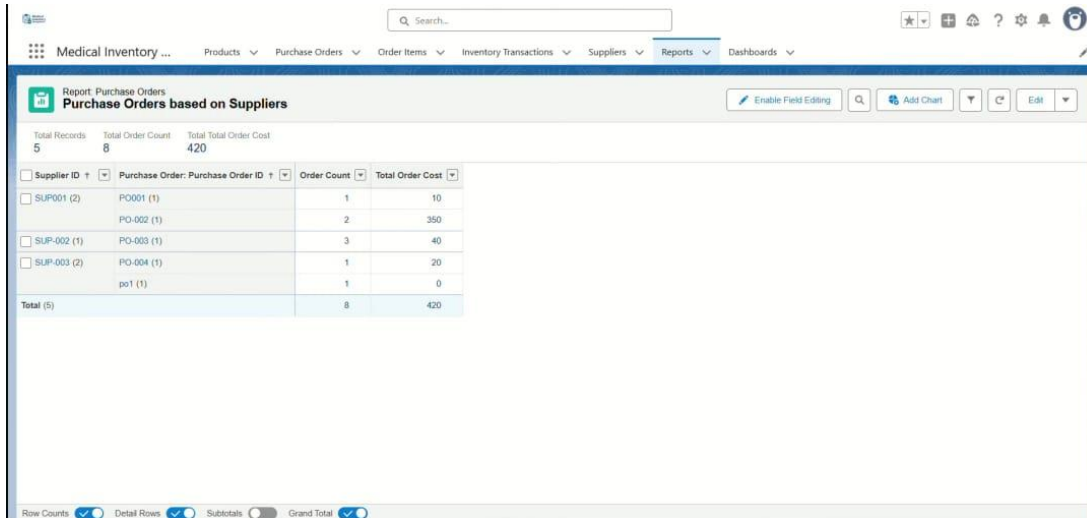
}

}

```

# Reports

## Created a Purchase Orders based on Suppliers (Summary) Report



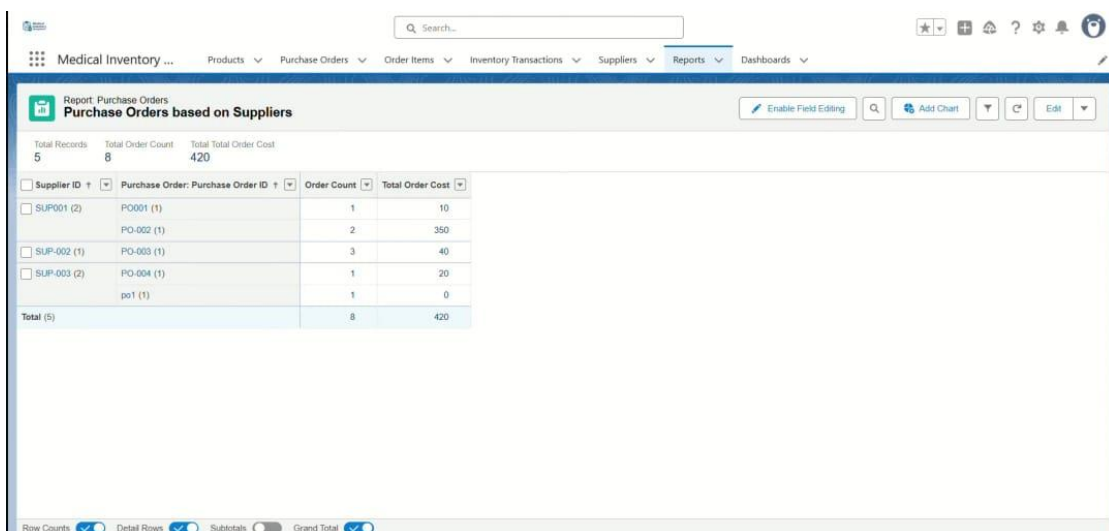
Report: Purchase Orders  
**Purchase Orders based on Suppliers**

Total Records: 5    Total Order Count: 8    Total Total Order Cost: 420

Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
SUP-001 (2)	PO-001 (1)	1	10
	PO-002 (1)	2	350
SUP-002 (1)	PO-003 (1)	3	40
SUP-003 (2)	PO-004 (1)	1	20
	po1 (1)	1	0
<b>Total (5)</b>		<b>8</b>	<b>420</b>

Row Counts: ☒    Detail Rows: ☒    Subtotals: ☐    Grand Total: ☒

## Created a Complete Purchase Details Report



Report: Purchase Orders  
**Purchase Orders based on Suppliers**

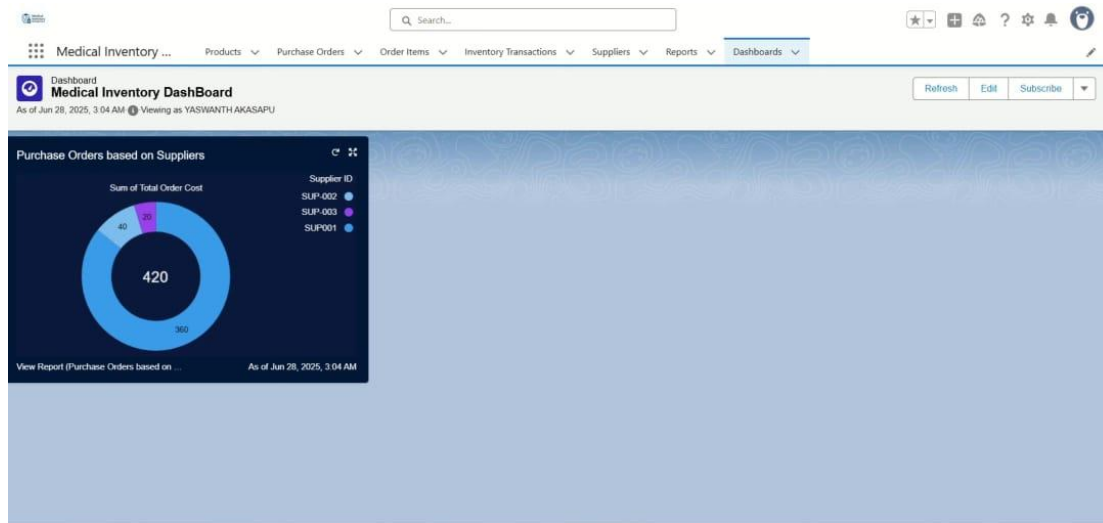
Total Records: 5    Total Order Count: 8    Total Total Order Cost: 420

Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
SUP-001 (2)	PO-001 (1)	1	10
	PO-002 (1)	2	350
SUP-002 (1)	PO-003 (1)	3	40
SUP-003 (2)	PO-004 (1)	1	20
	po1 (1)	1	0
<b>Total (5)</b>		<b>8</b>	<b>420</b>

Row Counts: ☒    Detail Rows: ☒    Subtotals: ☐    Grand Total: ☒

# Dashboards

## Creation and view of Dashboard



## Conclusion

The Salesforce-based Medical Inventory Management System successfully streamlines and automates the key operational aspects of medical inventory. By integrating supplier management, purchase order tracking, product cataloging, and expiry date monitoring, the system ensures accuracy, efficiency, and reliability in handling medical supplies. It reduces the risk of errors, prevents the use of expired products, and enhances decision-making through comprehensive reports and analytics. Overall, this project demonstrates how Salesforce can be effectively utilized to improve healthcare inventory operations, contributing to better resource management and patient safety.