Medical Inventory Management

College Name: Sri Ramalinga sowdambigai college of science and commerce

College Code: Bru3y

TEAM ID: NM2025TMID25514
TEAM MEMBERS: 4

Team Leader Name: Chandru S

Email: Dhineshchandru8656@gmail.com

Team Member1: Boopathi C

Email: Bboopathixo@gmail.com

TeamMember: Pranav M

Email: Pranavpranavm10@gmail.com

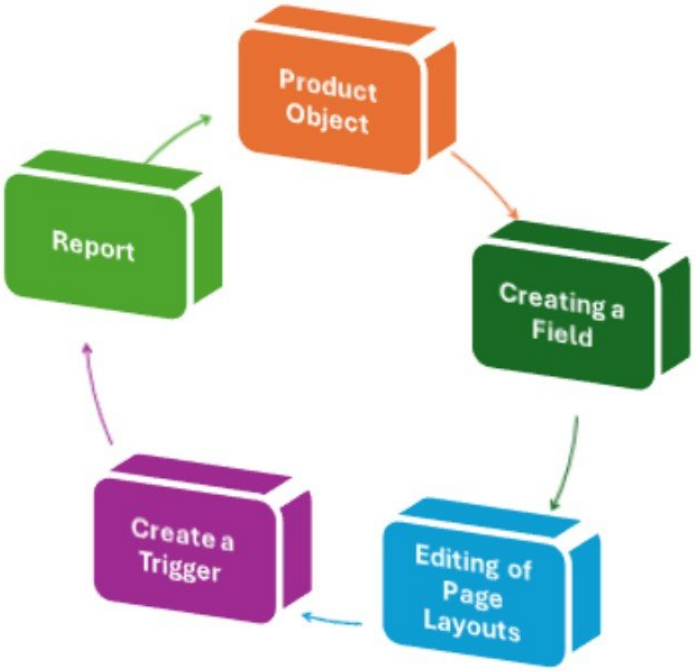
TeamMember: Hari Prasath T

Email: Mrrxveriyan950@gmail.com

1.INTRODUCTION

1.1 Project Overview

TheMedicalManagementSystem is a Salesforce-based application developed under the Naan Mudhalvan scheme to streamline hospital and clinic operations. It manages patients, doctors, appointments, prescriptions, billing, and medical records in a centralized cloud-based platform. The system ensures quick access to patient history, automated appointment reminders, and secure communication between patients and healthcare providers.



1.2 Purpose

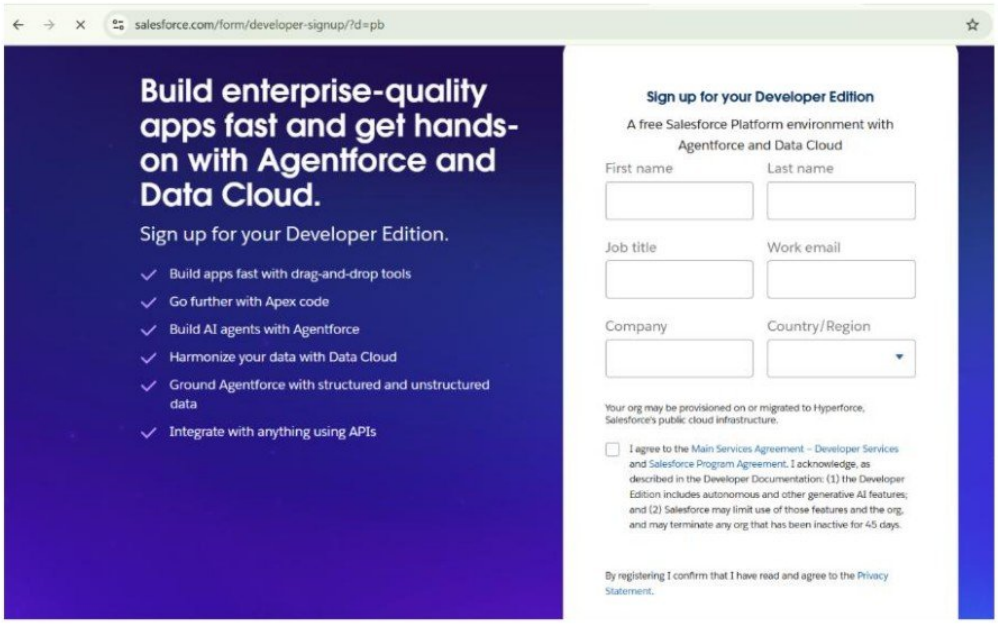
The main objective of the project is to create a centralized medical management solution that ensures efficiency, accuracy, and transparency in healthcare. The purpose is:

- To simplify patient registration and reduce duplicate records.
- To enable online and automated appointment booking with doctors.
- To send reminders and follow-up notifications to patients.
- To maintain accurate electronic health records (EHRs).
- To automate billing and approval workflows.
  To improve doctor–patient communication using Salesforce's email and notification system.

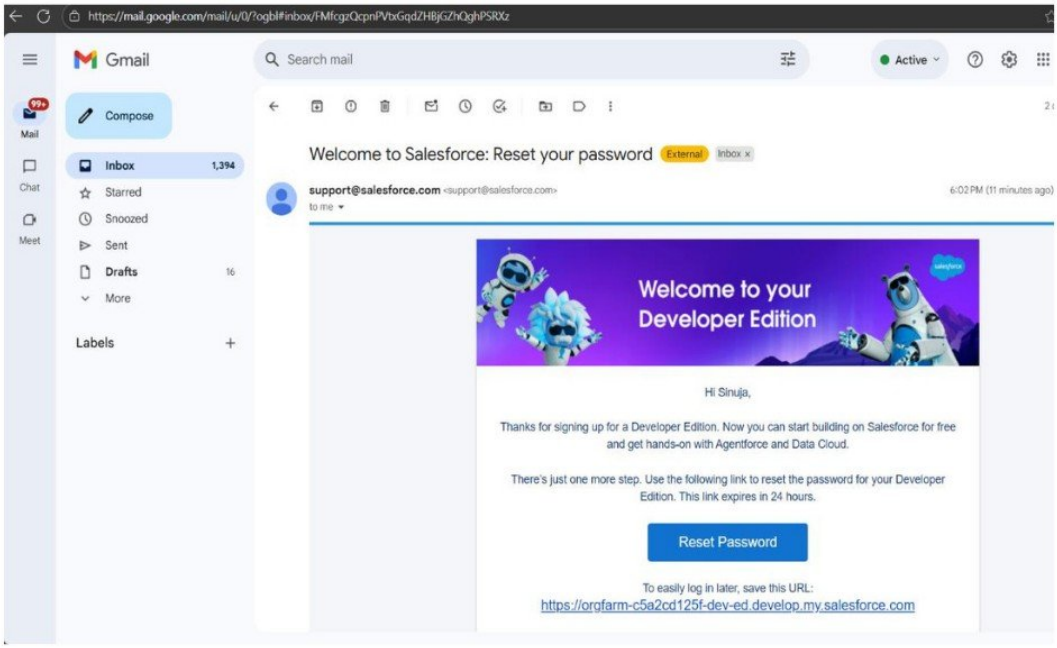
2.DEVELOPMENT PHASE

2.1 Creating Developer Account:

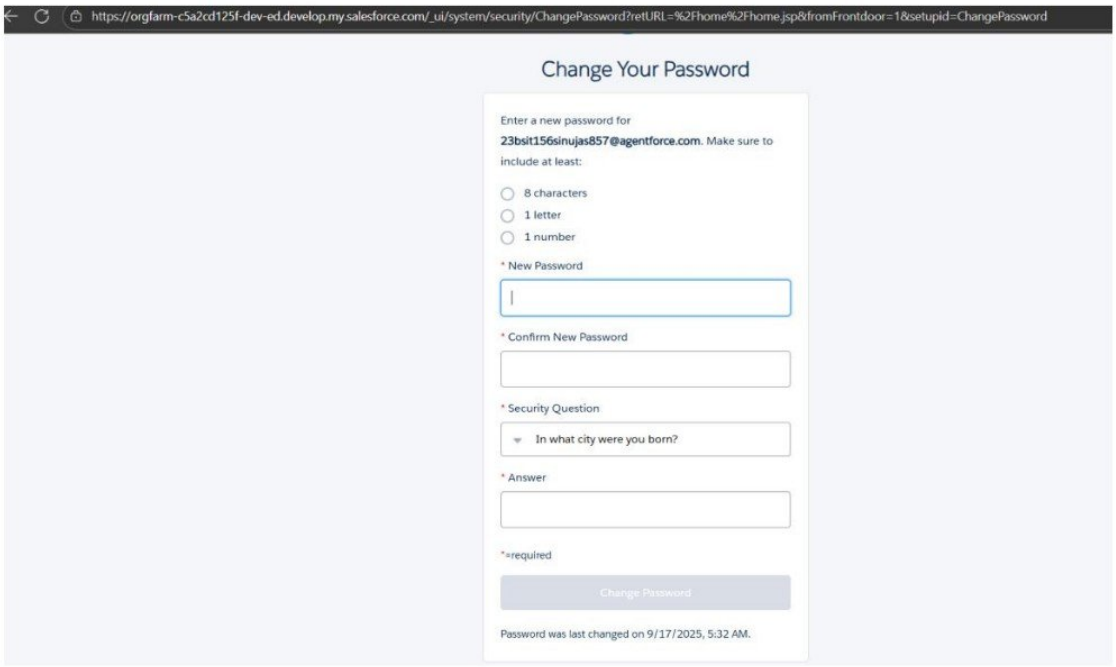
- Registered on Salesforce Developer Edition using **https://www.salesforce.com/form/developer-signup/?d=pb**
- Configured the org with profiles, permission sets, and security settings.



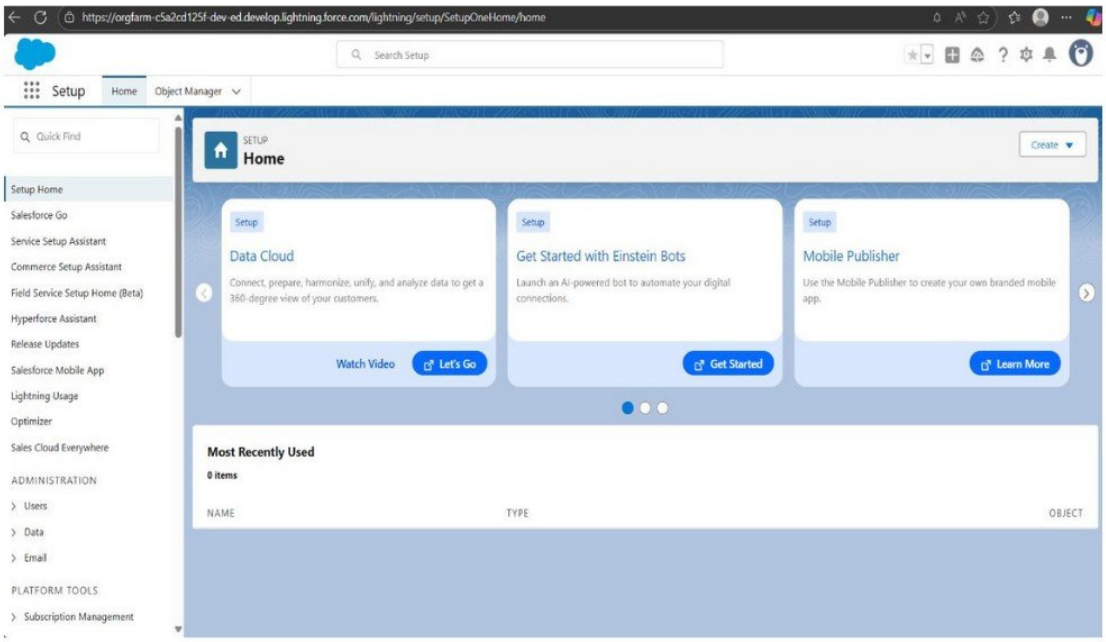
2.2 Account Activation:



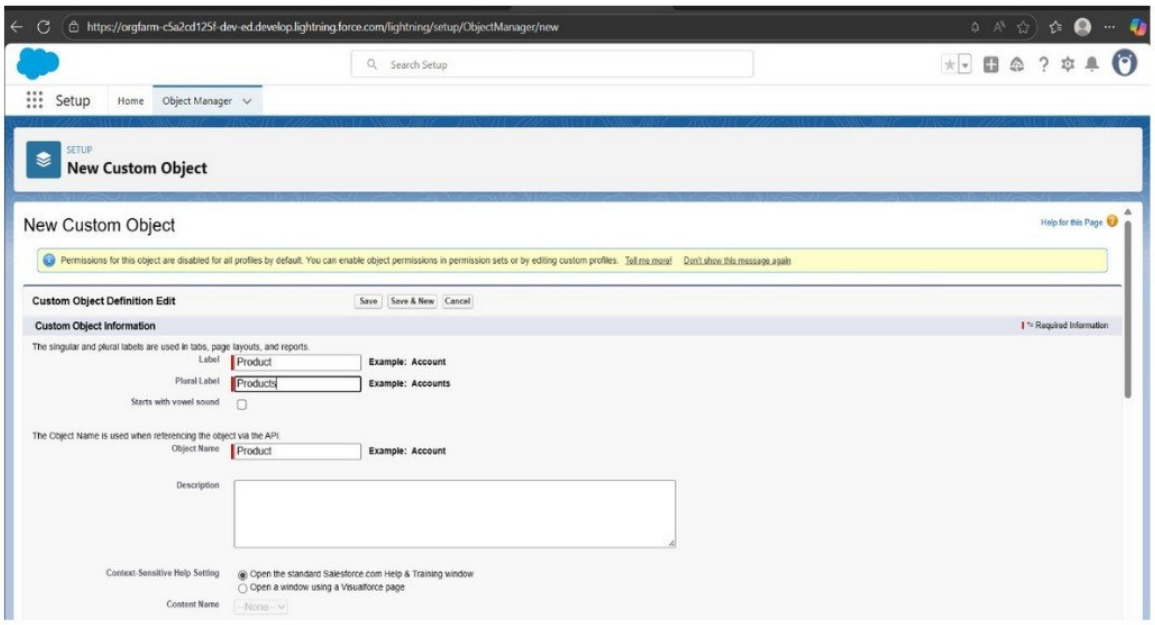
2.3 Verify Account

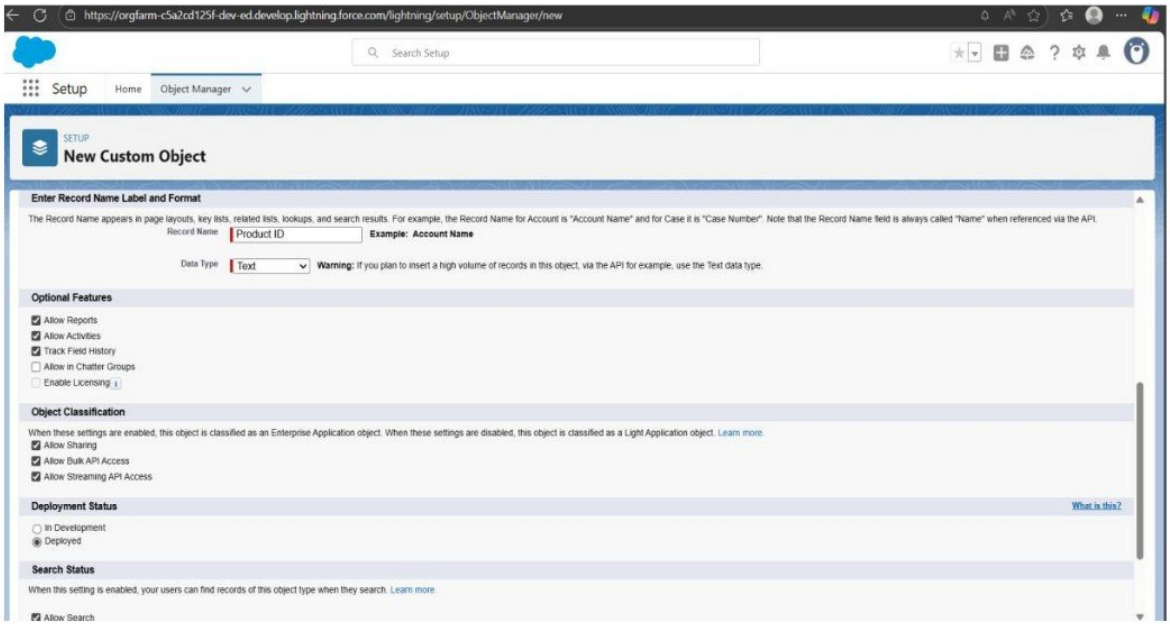


2.4Salesforce Setup Page

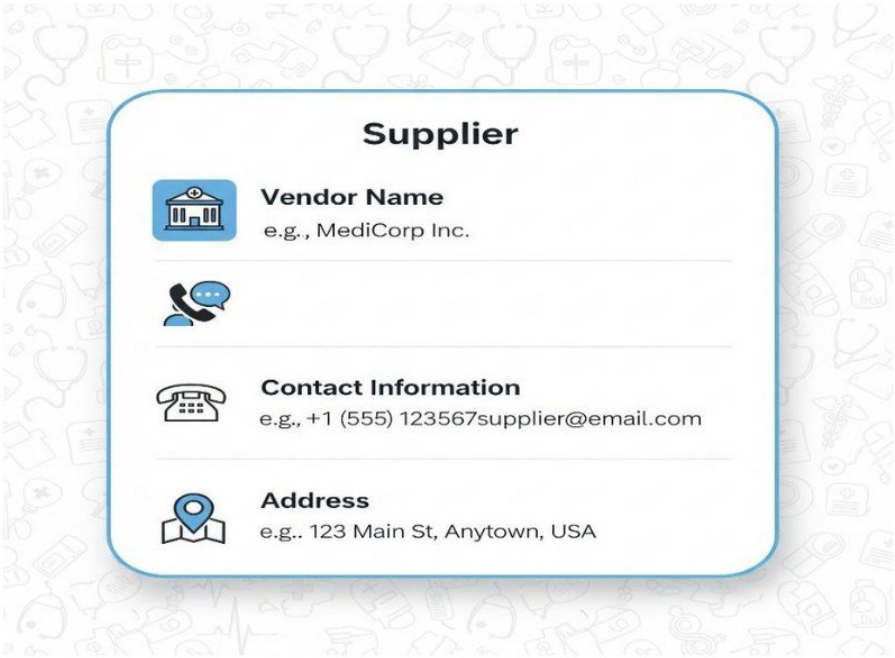


2.5Objects Created: Product

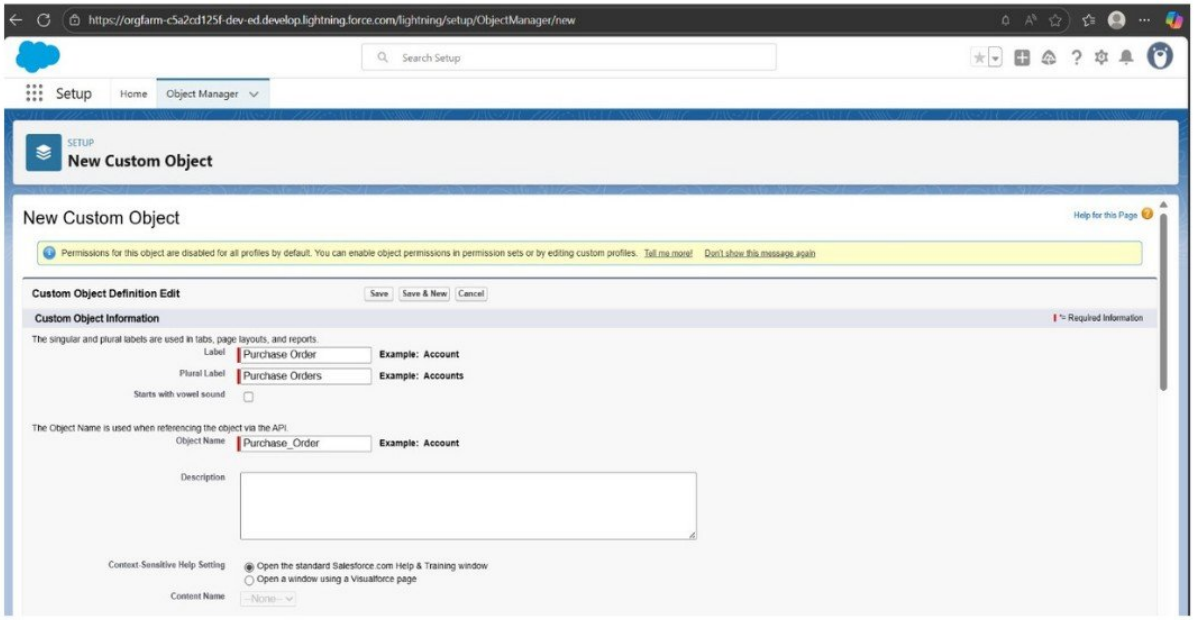




- Supplier: Contains information about vendors who supply the medicines.



- Purchase Order: Tracks orders placed with suppliers, linking themtospecific medicines and quantities.



2.3 Configurations Done

A series of configurations were implemented to automate processes and enforce business rules:

Creating Remaining Tabs

Procedure:

-Create tabs for the following objects:

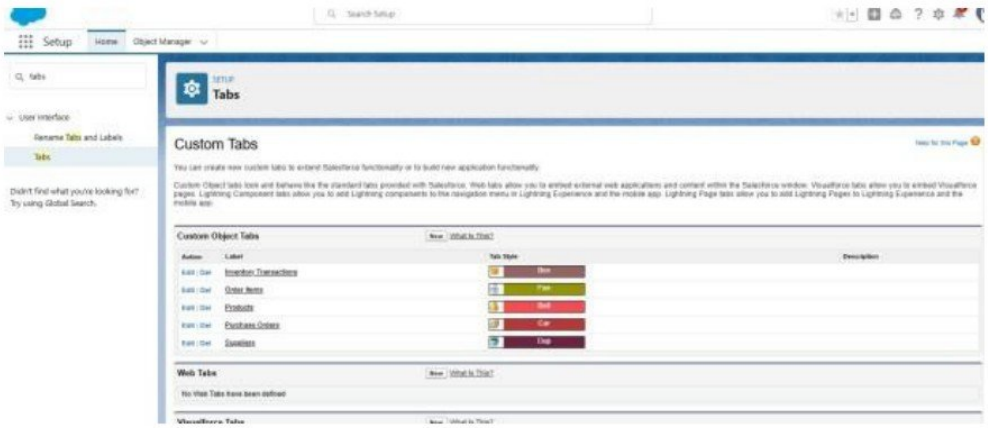
-Purchase Order

-Order Item

-Inventory Transaction

-Supplier

Follow the same steps as described in **Activity 1 (Creating a Tab for the Product Object)** to complete the process for each object.



The Lightning App

Activity 1: Creating a Lightning App for Medical Inventory Management

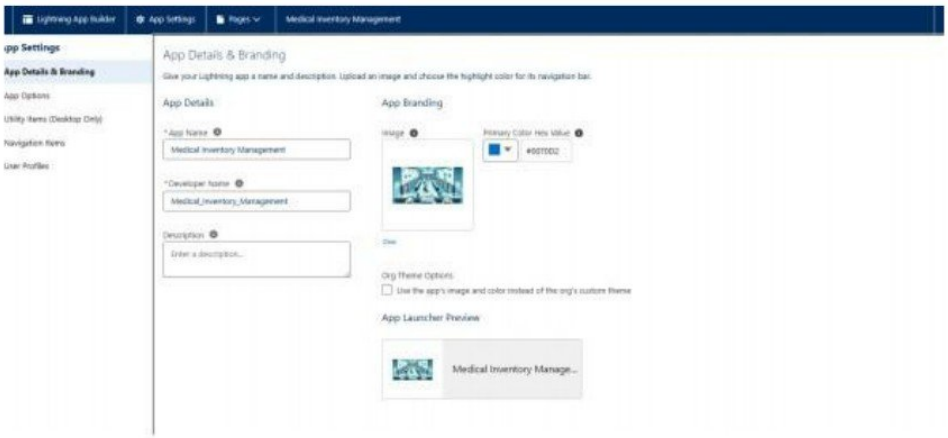
Procedure:

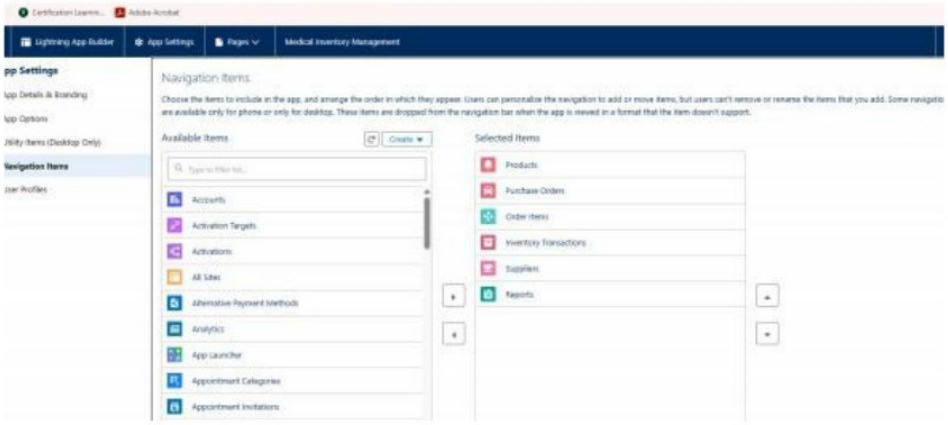
1. From **Setup**, enter **App Manager** in the Quick Find bar and select App Manager.
2. Click **New Lightning App**.
3. Enter **Medical Inventory Management** as the **App Name**.

   -Optionally, upload an image related to medical inventory.

   -Click **Next**.

4. Under **App Options**, leave the default selections and click **Next**.
5. Under **Utility Items**, retain the default configuration and click **Next**.



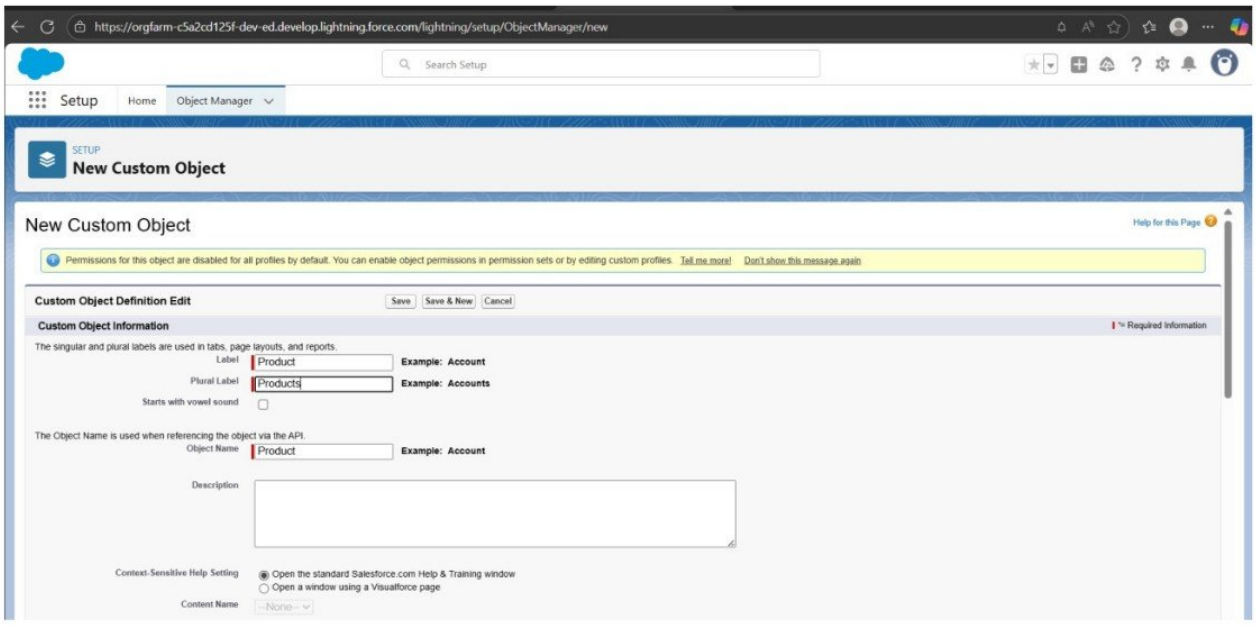


Fields

Creating a Text Field in the Product Object

Steps:

1.
2.
3. Click the **gear icon** and select **Setup** (opens in a new tab).
4. In Setup, go to the **Object Manager** tab.
5. Select the **Product** custom object.
6. From the left navigation, click **Fields & Relationships**.
7. Click **New**.
   Choose **Text** as the field type and click **Next**.
   Enter the following details:
   - **Field Label:** Product Name
   - **Length:** 255
8. Select the **Required Field** checkbox.
9. Click **Next → Next → Save & New** to create the field.



Creating a Number Field in the Product Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Product and select the Product custom object.

-From the left panel, click Fields & Relationships.

-Click New.

-ChooseNumber as the data type and click Next.

Enter the details:

Field Label: Current Stock Level

Length: 18

Decimal Places: 0

ClickNext→Next→ Save to finish creating the field.

Creating a Currency Field in the Product Object Steps:

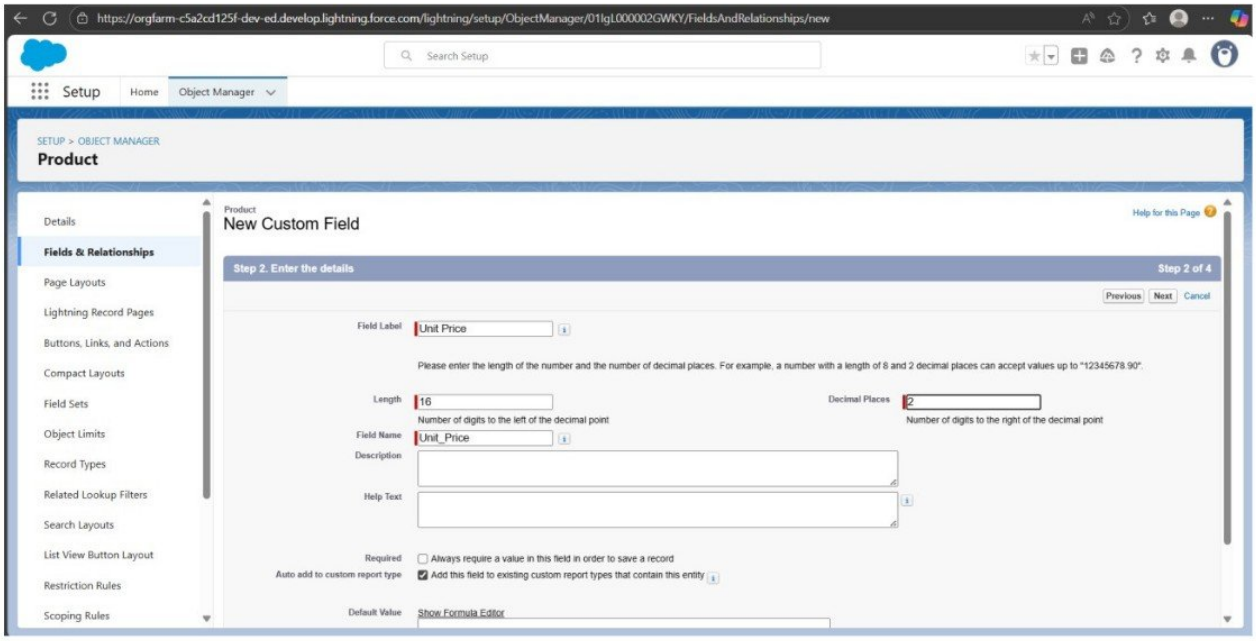
-Go to Setup → click on Object Manager. -In the Quick Find box, type Product and select the Product custom object. -From the left-hand menu, select Fields & Relationships. -Click New. -Choose Currency as the data type and click Next. **Enter the details:** Field

Label: Unit Price Length: 16

Decimal Places: 2

Mark the field as Required.

Click Next → Next → Save.



Creating a Lookup Relationship in the Purchase Order Object

A Lookup Relationship in Salesforce links two objects together, where one object (child) references another (parent). This helps maintain relational data integrity and allows easy navigation between related records.

In this activity, we'll establish a relationship from Purchase Order (child) to Supplier (parent).

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Purchase Order and select the Purchase Order custom object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Lookup Relationship as the data type and click Next.

-For the related object, select Supplier.

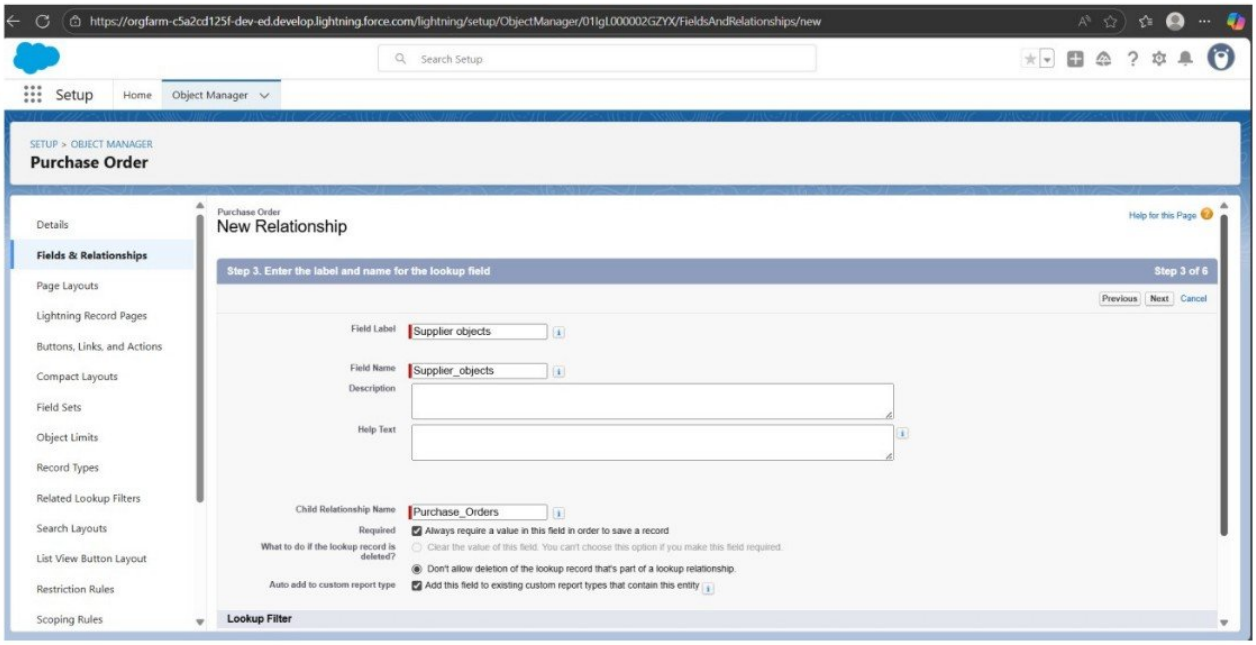
-Click Next.

Enter the details:

Field Label: Supplier ID

Mark the field as Required.

Continue by clicking Next → Next → Next → Save.



Creating a Date Field in the Purchase Order Object

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Purchase Order and select the Purchase Order custom object.

-From the left-hand menu, click Fields & Relationships.

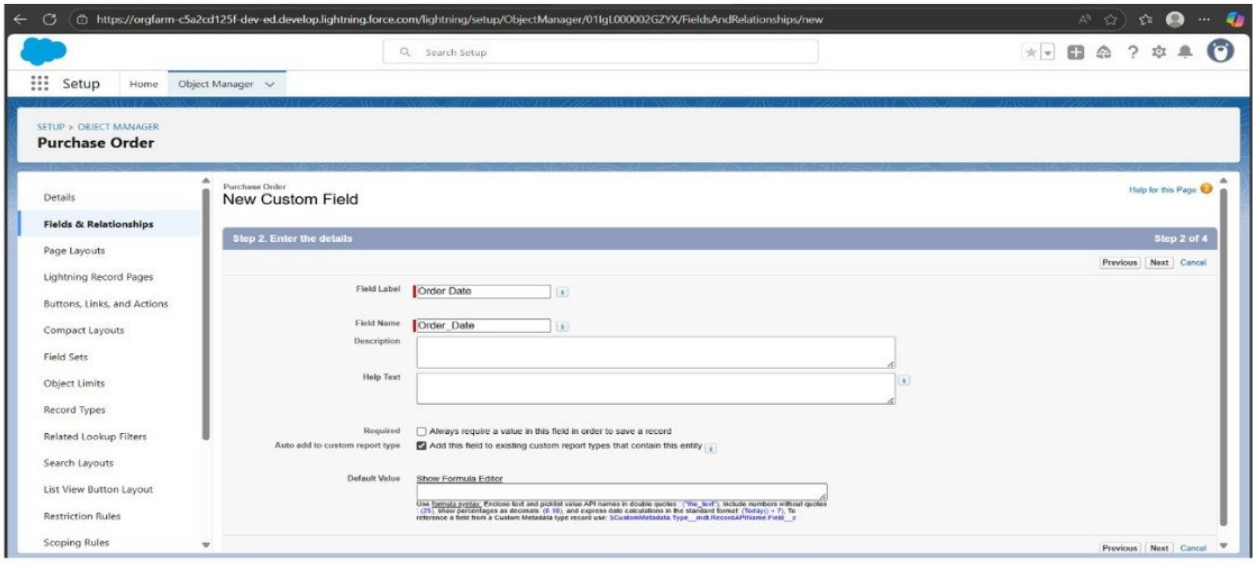
-Click New.

-Select Date as the data type and click Next.

-Enter the following details:

-Field Label: Order Date

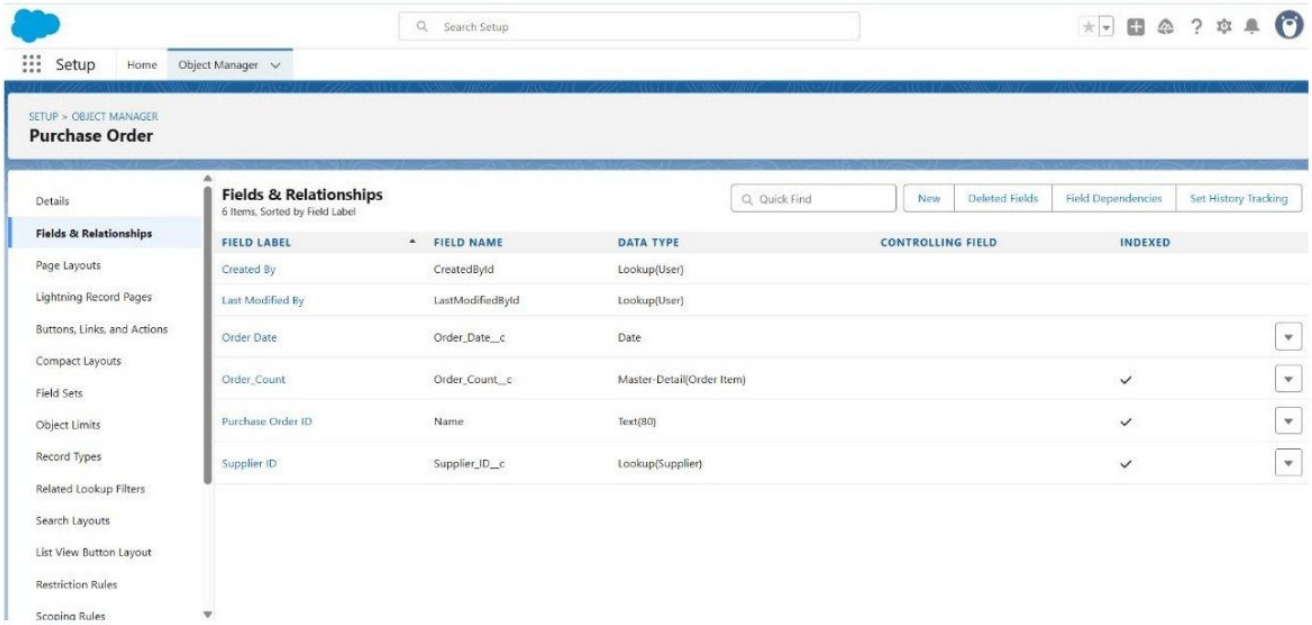
-Click Next → Next → Save to complete the creation of the date field.



Creating a Roll-Up Summary Field in Purchase Order object To create fields in an object:

1. Gotosetup>>clickonObjectManager>>typeobjectname(PurchaseOrder)

in quick find box>> click on the Purchase Order object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Roll-Up Summary” and click Next.
5. Enter Field Label as “ Order Count”.
6. Choose the Summarized Object as “Order Items”.
7. For Select Roll-Up Type select “Count”.
8. Click on Next, Next and Save



Creating a Unit Price Formula Field in the Order Item Object

Steps:

-Go to **Setup** → click on **Object Manager**.

-In the Quick Find box, type **OrderItem** and select the **Order Item** custom object.

-From the left-hand menu, click **Fields&Relationships**.

-Click **New**.

-Select **Formula** as the data type and click **Next**. **Enter**

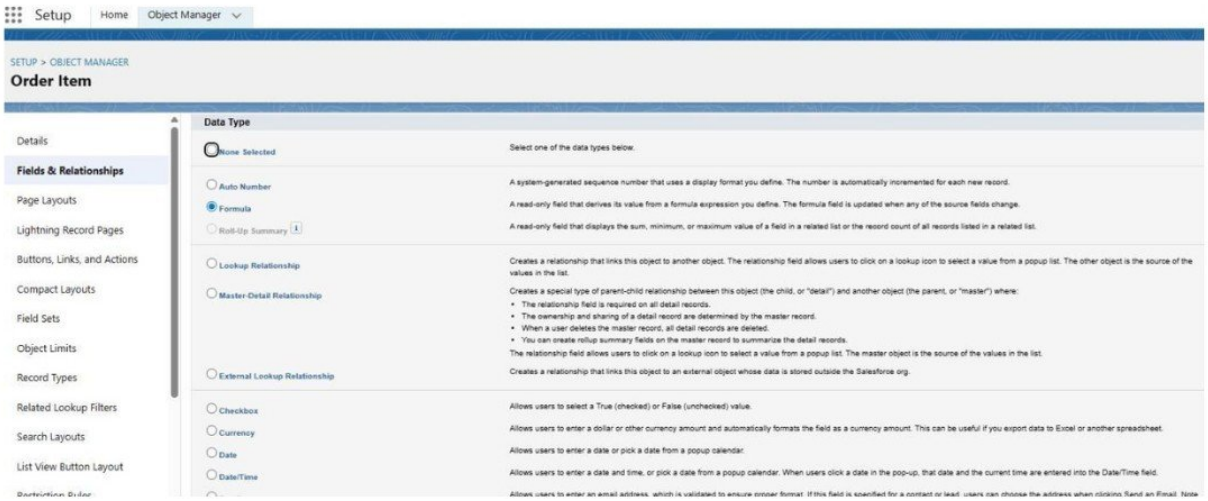
the following details:

Field Label: Unit Price

Formula ReturnType: Currency

In the formula editor, enter the advanced formula:
Product_ID__r.Unit_Price__c

This pulls the **Unit Price** directly from the related **Product** object. Click **Next** → **Next** → **Save** to complete the field creation



Creating an Amount Formula Field in the Order Item Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Order Item and select the Order Item custom object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Formulaasthedata type and click Next.

Enter the following details:

Field Label: Amount

Formula Return Type: Currency

In the formula editor, enter the advanced formula:

Quantity_Received__c * Unit_Price__c

This calculates the total price for each Order Item automatically.

Click Next → Next → Save to complete the field creation.

Creating a Picklist Field in the Inventory Transaction Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Inventory Transaction and select the object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Picklist as the data type and click Next.

Enter the following details:

Field Label: Transaction Type

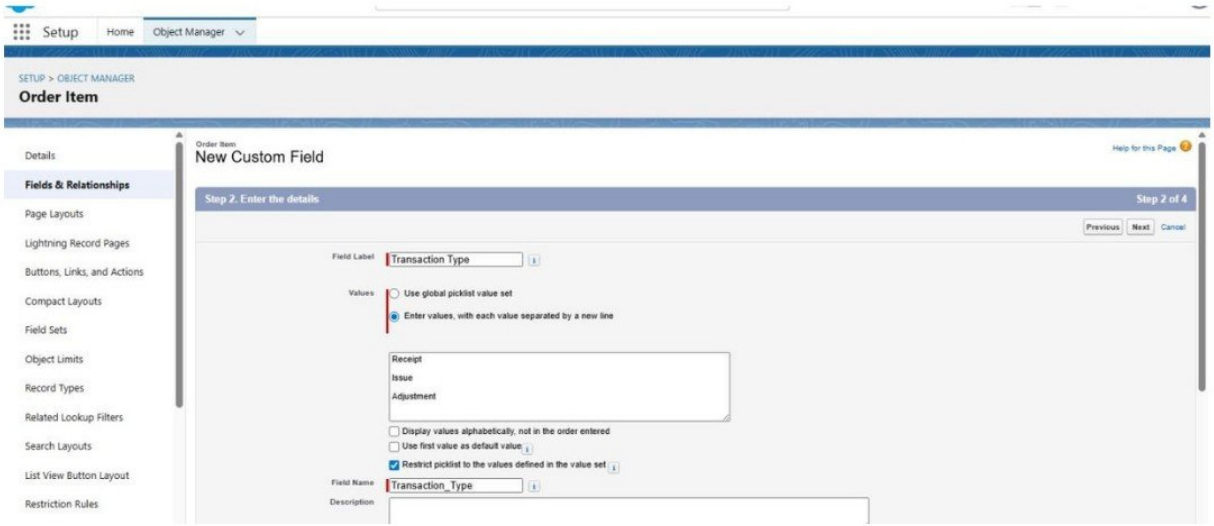
Values: Enter manually, each on a new line:

Receipt

Issue

Adjustment

Click Next → Next → Save to complete the picklist creation.



Creating a Total Order Cost Formula Field in the Inventory Transaction Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Inventory Transaction and select the object.

-From the left-hand menu, click Fields & Relationships.

-Click New.

-Select Formula as the data type and click Next.

-Enter the following details:

Field Label: Total Order Cost

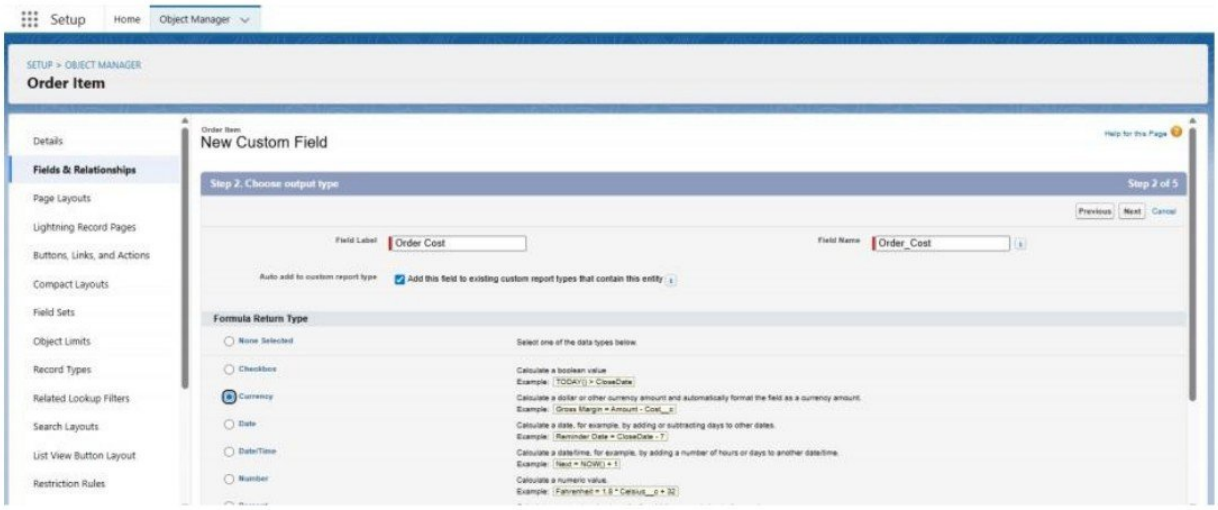
Formula Return Type: Currency

-0In the formula editor, enter the advanced formula:

Purchase_Order_ID r.Total_Order_Cost c_

-This formula pulls the total cost from the related Purchase Order, ensuring accurate cost tracking for inventory transactions.

-Click Next → Next → Save to complete the field creation.



Creating a Phone Field in the Supplier Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Supplier and select the Supplier custom object.

-From the left-hand menu, click Fields & Relationships.

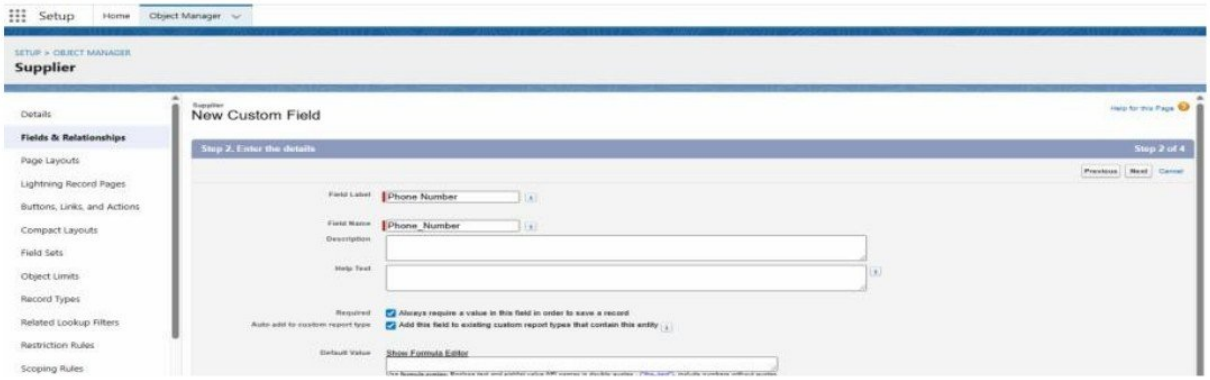
-Click New.

-Select Phone as the data type and click Next.

Enter the following details:

Field Label: Phone Number -Mark the field as Required. -

Click Next → Next → Save to complete the field creation.



Creating an Email Field in the Supplier Object

Steps:

Go to **Setup** → click on **Object Manager**.

In the Quick Find box, type **Supplier** and select the **Supplier** custom object. From the left-hand menu, click **Fields & Relationships**.

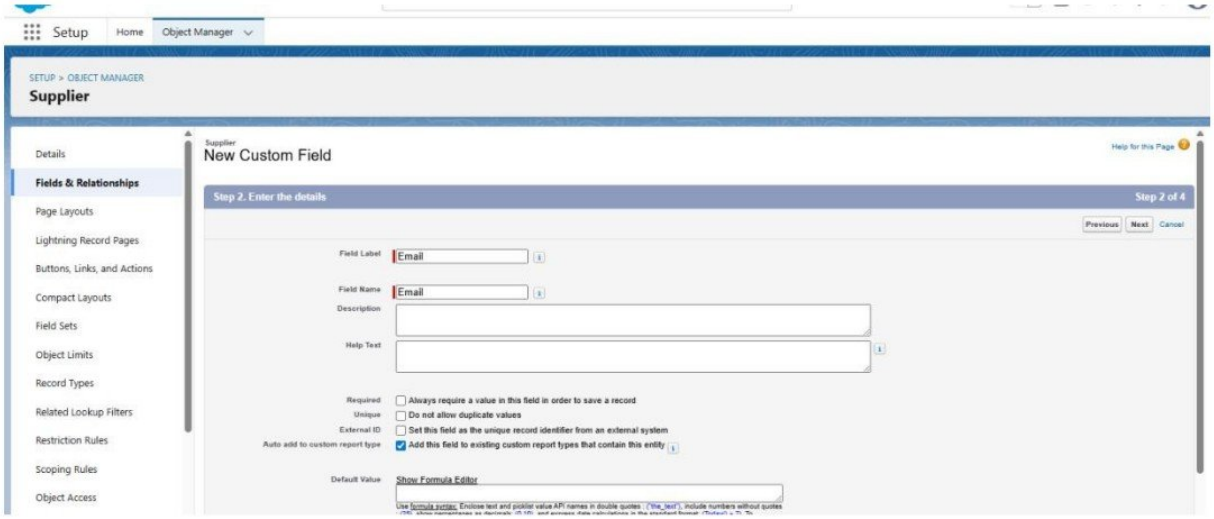
Click **New**.

Select **Email** as the data type and click **Next**.

Enter the following details:

Field Label: Email

Click **Next** → **Next** → **Save** to complete the field creation.



Page Layout Customization

Editing a Page Layout in the Product Object

Steps:

-Go to Setup → click on Object Manager.

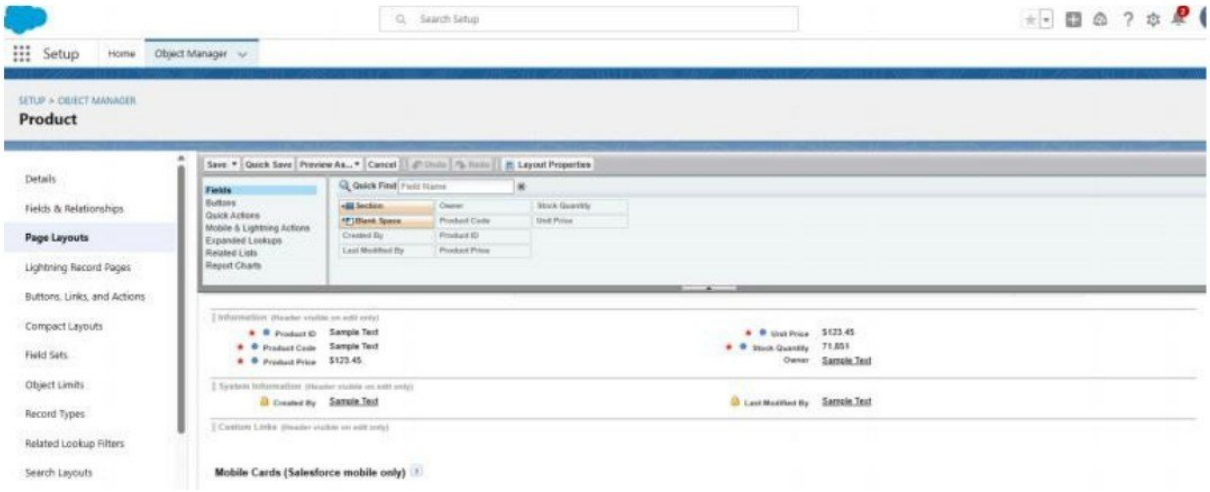
-In the Quick Find box, type Product and select the Product custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Product Layout.

-Drag and arrange the fields on the page layout as required to optimize data entry and display.

- Save it



Editing a Page Layout in the Purchase Order Object

Steps:

-Go to Setup → click on Object Manager. -In the Quick Find box, type Purchase Order and select the Purchase Order custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Purchase Order Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

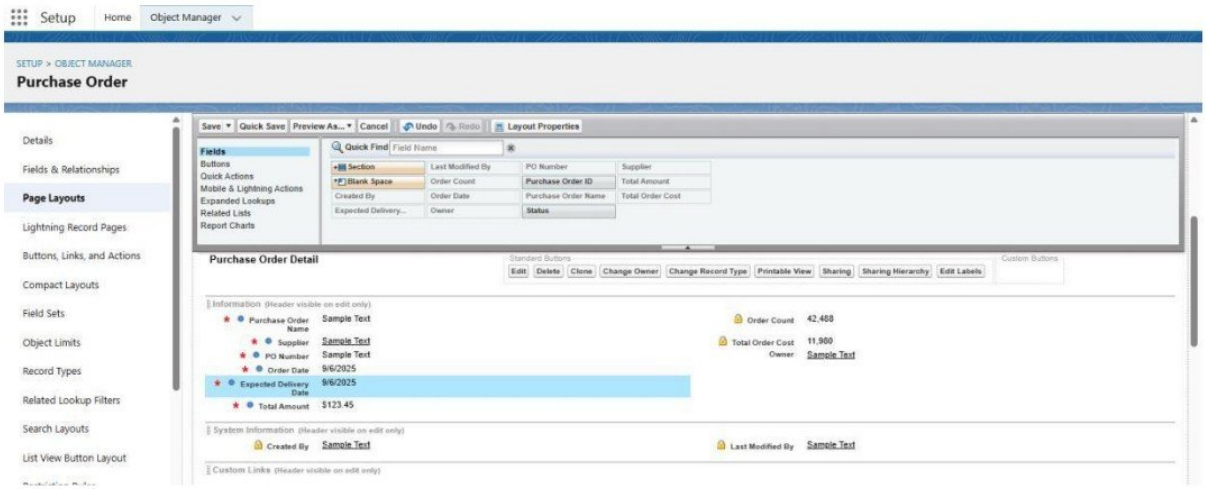
For the Order Date **field:**

-Click on the field → click Settings → select Required → save.

-For the Total Order Cost field:

-Click on the field → click Settings → select Read-Only → save.

-Click Save to finalize the layout changes.



Editing a Page Layout in the Order Item Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Order Item and select the Order Item custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Order Item Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

-Click Save to finalize the layout changes.

Editing a Page Layout in the Inventory Transaction Object

Steps:

-Go to Setup → click on Object Manager.

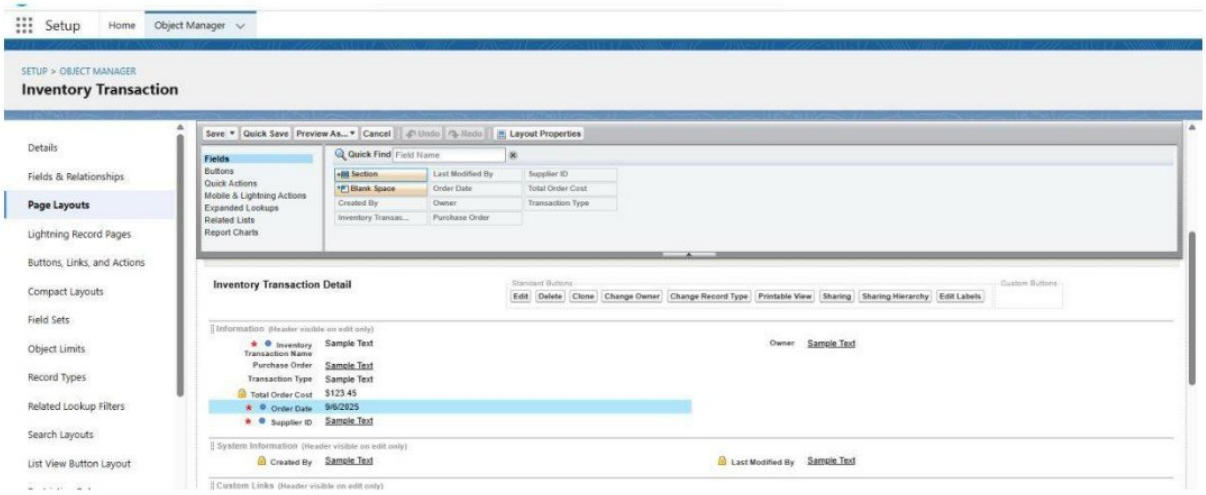
-In the Quick Find box, type Inventory Transaction and select the Inventory Transaction custom object.

-From the left-hand menu, click Page Layouts.

-Select the layout named Inventory Transaction Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

-Click Save to finalize the layout changes.



Editing a Page Layout in the Supplier Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Supplier and select the Supplier custom object.

-From the left-hand menu, click Page Layouts.

--Select the layout named Supplier Layout.

-Drag and arrange the fields on the layout as required to optimize data entry and display.

-Click Save to finalize the layout changes.

Compact Layouts

Creating a Compact Layout for the Product Object

Steps:

-Go to Setup → click on Object Manager.

-In the Quick Find box, type Product and select the Product custom object.

-From the sidebar, click Compact Layouts.

-Click New.

-Enter the following details:

Label: Product Compact Layout

-Select the fields to display in the compact layout:

Product Name

Unit Price

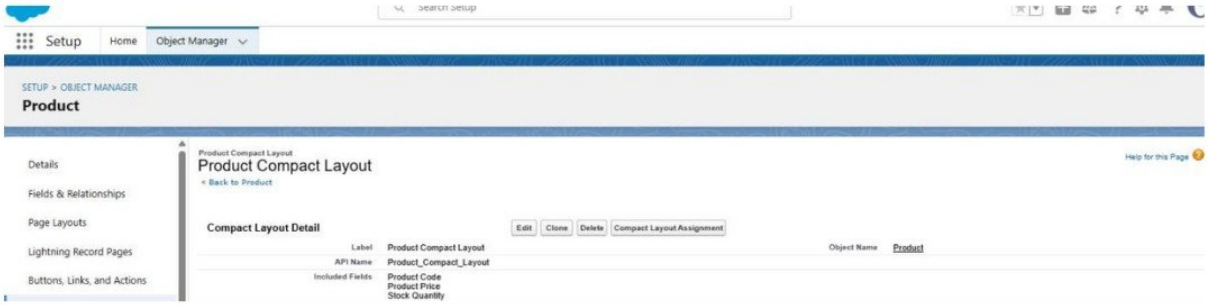
Current Stock Level

-Click Save.

-Click Compact Layout Assignment.

-Click Edit Assignment.

-Choose Product Compact Layout from the dropdown and click Save



Creating a Compact Layout for the Purchase Order Object

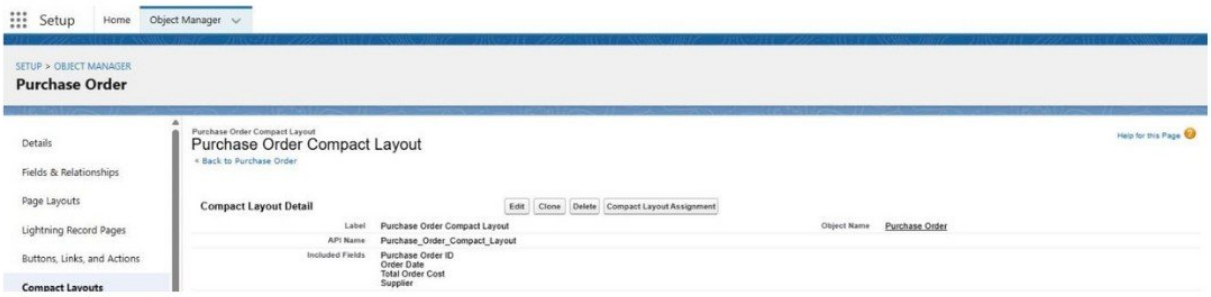
Steps:

1. 2.

3. Go to **Setup** → click on **Object Manager**.
4. In the Quick Find box, type **Purchase Order** and select the**Purchase Order**
5. custom object.
   From the sidebar, click **Compact Layouts**.
6. Click **New**.
   Enter the following details:
   - **Label:** Purchase Order Compact Layout

   Select the fields to display in the compact layout:
   - Purchase Order ID
   - Order Date
   - Total Order Cost
   - Supplier ID

7. Click**Save**.
8. Click Compact Layout Assignment → Edit Assignment.
9. Choose **Purchase Order Compact Layout** from the dropdown.
10. Click**Save**.



ValidationRules

Activity 1: Creating an Expected Delivery Date Validation Rule for the Purchase Order Object

Steps:

1. 2.

Go to **Setup** → click on **Object Manager**.
In the QuickFind box, type**Purchase Order** and select the**Purchase Order** custom object.

3. From the left-hand menu, click **Validation Rules → New**.
4. Enter the following details:
   - **Rule Name:** Expected Delivery Date Validation
   - **Active:**Checked
5. Intheformulaeditor,enter the errorcondition formula:
6. (Expected_Delivery_Date_c - Order_Date_____c)> 7

   This ensures that the expected delivery date cannot exceed 7 days from the order date.

7. Click **Save** to activate the validation rule.

Profiles

Creating an Inventory Manager Profile Steps:

Go to Setup → type Profiles in the Quick Find box → click Profiles.

Locate Standard User → click Clone.

Enter the Profile Name: Inventory Manager → click Save.

On the newly created profile page, click Edit.

Configure the following settings:

Custom App Settings: Set Medical Inventory Management as default.

Password Policies:

User passwords expire in: Never Expires

Minimum password length: 8

Click Save.

Creating a Purchase Manager Profile Steps:

Go to Setup → type Profiles in the Quick Find box → click Profiles.

Locate Standard User → click Clone.

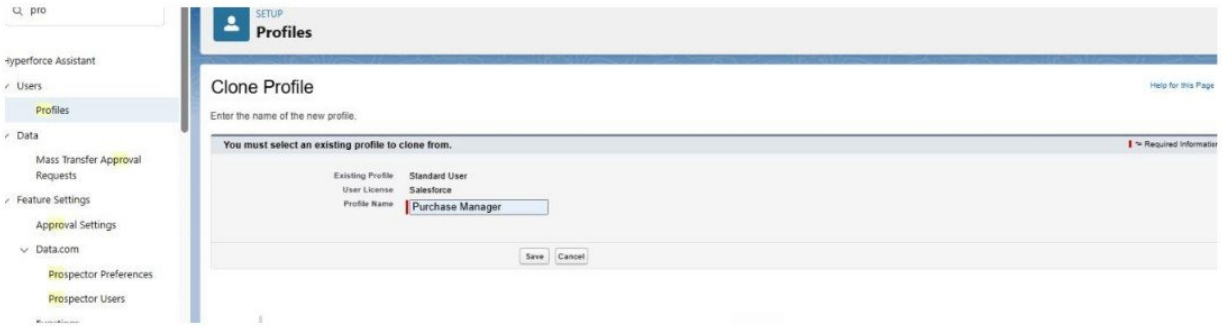
Enter the Profile Name: Purchase Manager → click Save.

On the newly created profile page, click Edit.

Configure the following settings:

Custom App Settings: Set Medical Inventory Management as default.

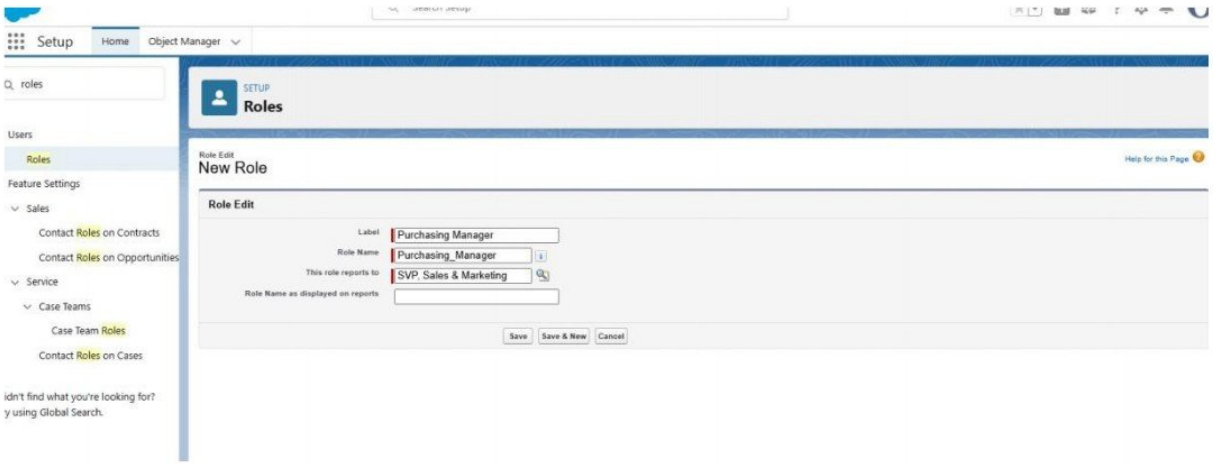
Click Save.



Roles

Creating a Purchasing Manager Role Steps:

1. Go to Setup → type Roles in the Quick Find box → click Set Up Roles.
2. Click Expand All to view the role hierarchy.
3. Under the SVP, Sales & Marketing role, click Add Role.
4. Enter the following details:
    - o Label: Purchasing Manager
    - o The Role Name will auto-populate.
5. Click Save to create the role.



Creating an Inventory Manager Role Steps:

Go to Setup → type Roles in the Quick Find box → click Set Up Roles.

Click Expand All to view the role hierarchy.

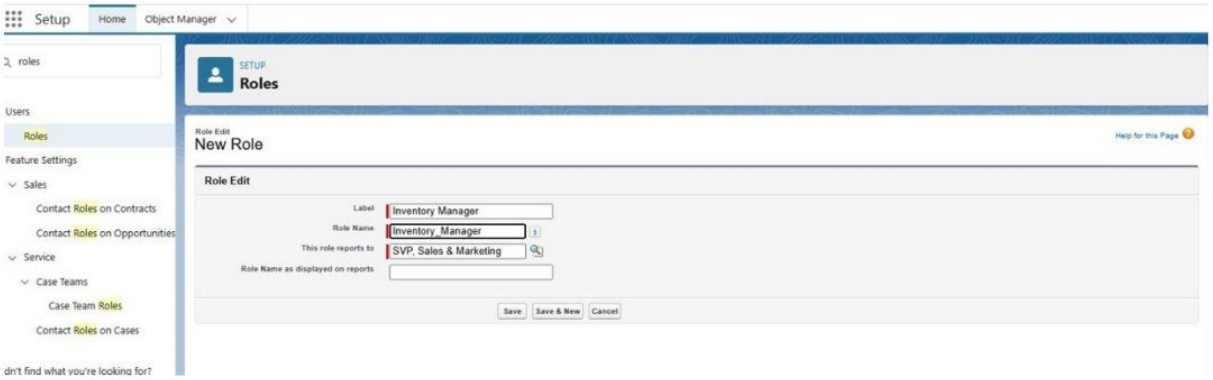
Under the SVP, Sales & Marketing role, click Add Role.

Enter the following details:

Label: Inventory Manager

The Role Name will auto-populate.

Click Save to create the role.



Permission Sets Activity 1:

Creating a Permission Set Steps:

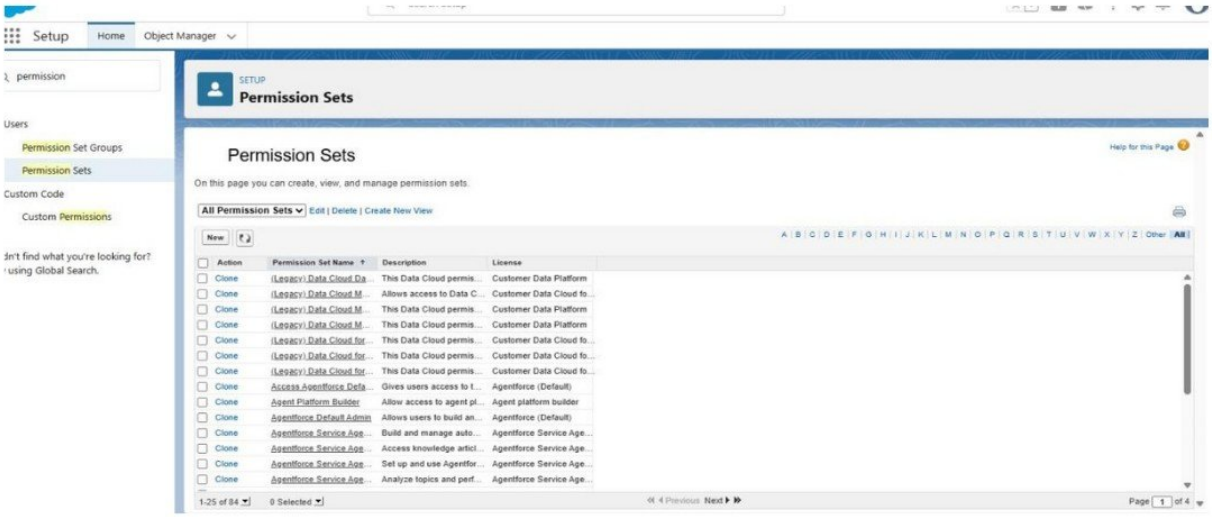
Go to Setup → type Permission in the Quick Find box → select Permission Sets.

Click New.

Enter the following details:

Label: Purchase Manager Create Access

Click Save to create the permission set.



Flows

Creating a Flow to Update the Actual Delivery Date Steps:

Go to Setup → type Flow in the Quick Find box → click Flows → New Flow → select Start From Scratch.

Choose Record-Triggered Flow → click Create.

Under Object, select Purchase Order.

Configure the trigger: A record is created or updated.

Set Entry Conditions: None.

Select Fast Field Updates → click Done.

Get Records Element

1. Click the "+" icon → select Get Records.
2. Enter Label: Get Purchase Record.
3. Select Object: Purchase Order.
4. For Condition Requirements, choose All Conditions Are Met (AND).
5. Set the condition:

Field: Id

Operator: Equals

Value: {!$Record.Id}

How Many Records to Store: Only the First Record.

How to Store Record Data: Choose fields and let Salesforce do the rest → select Order_Date__c → click Done.

Create a Variable

14. InFlowBuilder,clickManager→ New Resource.
15. Resource Type: Variable
16. API Name: ActualDeliveryDate
17. Data Type: Date → click Done.

Assignment Element

18. DraganddropAssignmentfromthe Toolbox.
19. Enter Label: Assignment.
20. Set Variable Values:

Variable: {!ActualDeliveryDate}

Operator: Equals

Value: {!$Record.Order_Date_c}

Variable: {!ActualDeliveryDate}

Operator: Add

Value: 3

Click Done.

Update Records Element

22. DraganddropUpdateRecords → connect it to the Assignment element.
23. EnterLabel:UpdatingPurchase Order.
24. HowtoFindRecordstoUpdate: Use the Purchase Order record that triggered the flow.
25. FilterConditions:None–Always Update Record.
26. Set Field Values:

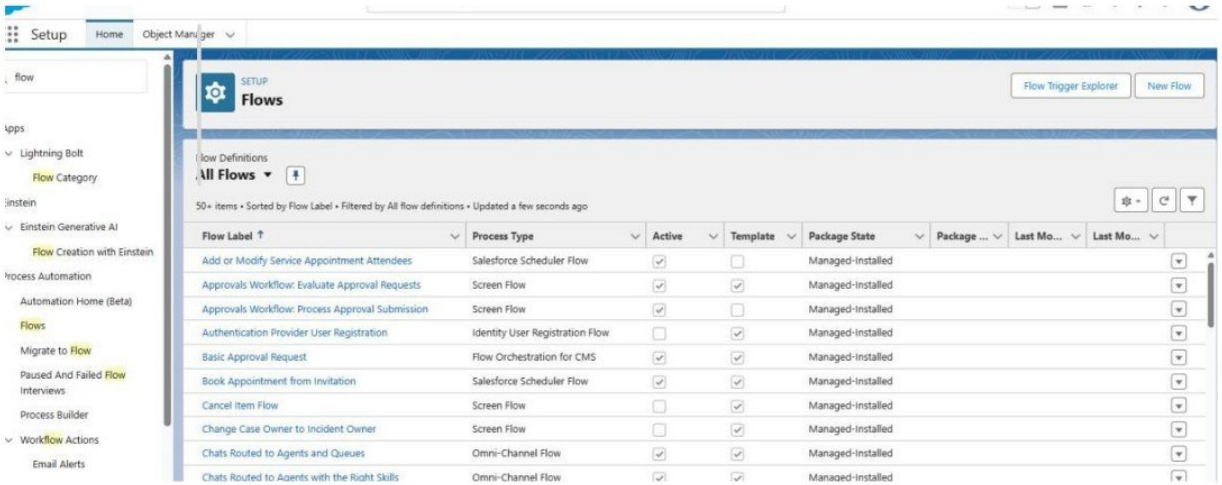
Field: Actual_Delivery_Date_c

Value: {!ActualDeliveryDate}

Click Done.

Save and Activate Flow

28. SavetheflowasActualDeliveryDate Updating.
29. Activate the flow.



Triggers

Creating a Trigger to Calculate Total Amount on Order Item

Step 1: Login to Salesforce

Log in to your Salesforce account with administrative privileges.

Step 2: Navigate to Developer Console

Clickthe **gear icon** (Setup)at the top-right corner → open th**Setup menu**.

Click **Developer Console** → opens in a new browser tab/window.

Step 3: Create the Apex Trigger

In Developer Console, go t**File** → **New** → **Apex Trigger**.

Name the trigger: CalculateTotalAmountTrigger.

Paste the following code:

```
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: Create the Apex Handler Class

In Developer Console, go t**File** → **New** → **Apex Class**.

Name it CalculateTotalAmountHandler.

Paste the following code:

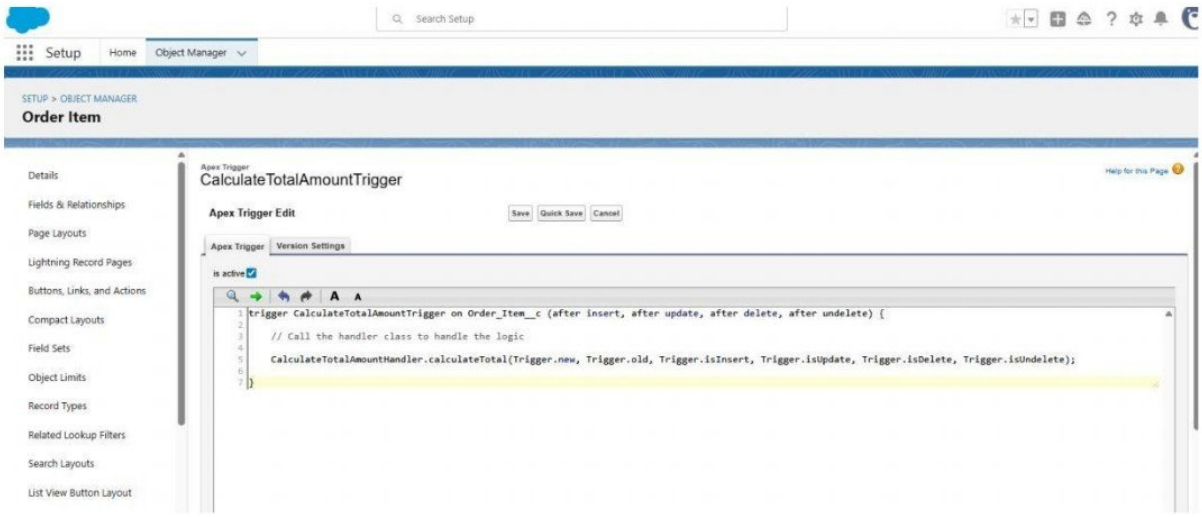
```
public class CalculateTotalAmountHandler {

    // 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

        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()) {

            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);
        }

        // 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;
        }
    }
  }
}
```

Step 5: Save and Test

Click **Save** for both the Trigger and the Handler Class.

Test by creating, updating, or deleting **Order Items**. The **Total Order Cost** on the related Purchase Order should update automatically.



Reports

CreateaPurchase Orders based on Suppliers(Summary) Report

1. Click App Launcher
2. Select Medical Inventory Management App
3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders Click Start report.
6. Click on Filters and select as follows and click on Apply
7. Customize your report, in group rows select – Supplier ID, Purchase Order: Purchase Order ID, for columns Order Count, Total Order Cost (In this way we are making a Summary Report).
8. Click save and run
9. Give report name – Purchase Orders based on Suppliers.
10. Click Save

NOTE: In this report you can see your all record of the object you selected for reporting

What you selects in "Select a report type option")

(View Report

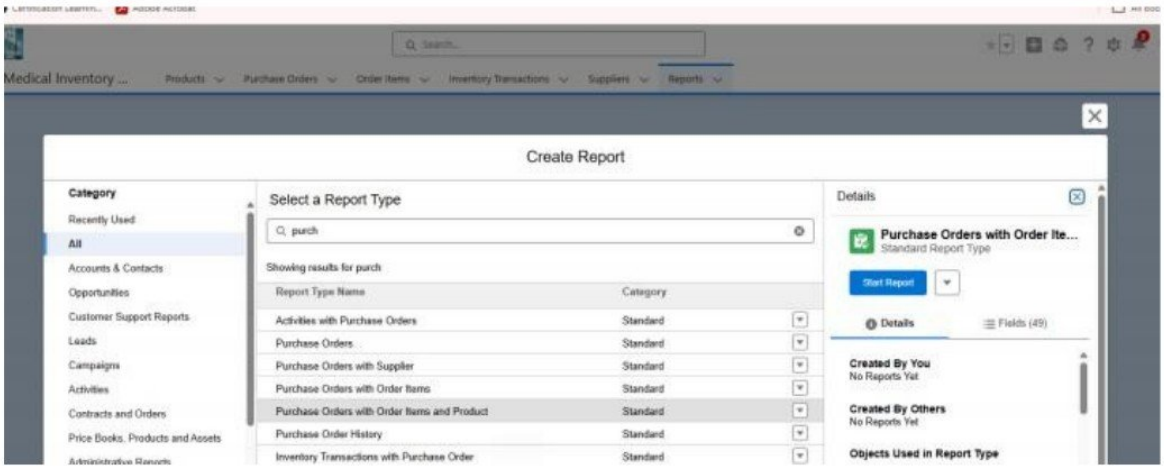
1. Click on App Launcher on the left side of the screen.
2. Search Medical Inventory Management App & click on it.
3. Click on Reports Tab.
4. Click on Purchase Orders based on Suppliers and see records.

Create a Complete Purchase Details Report

1. Click App Launcher
2. Select Medical Inventory Management App
3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders with Order Items and Product ID

>> Click Start report.

6. Click on Filters and select as follows and click on Apply



Dashboards

Create Dashboard

Open the Dashboards tab within the Medical Inventory Management application.

Click New Dashboard.

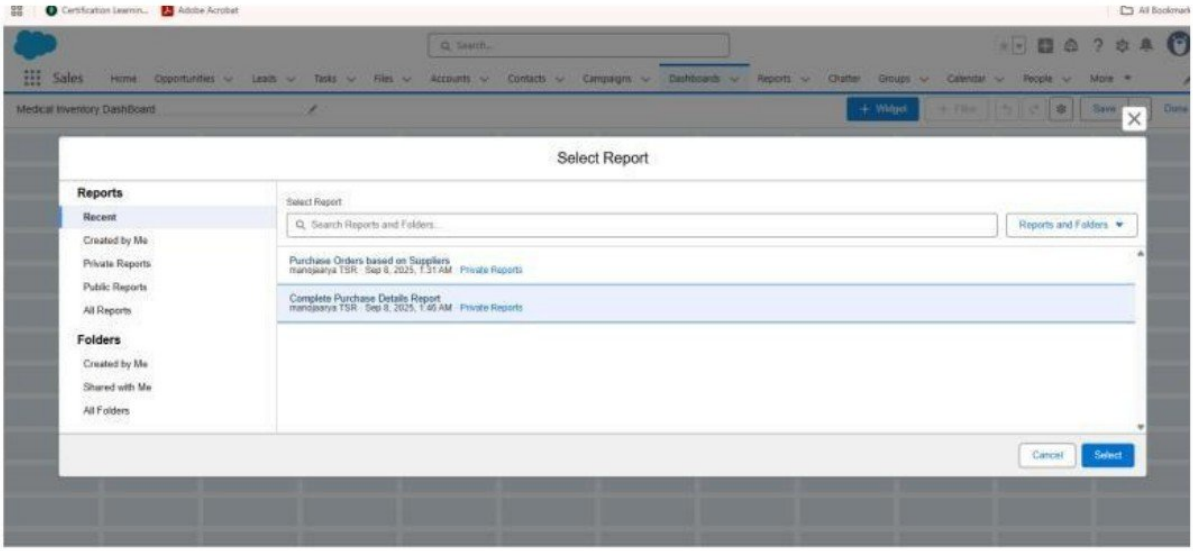
Enter the Name: Medical Inventory Dashboard → Click Create.

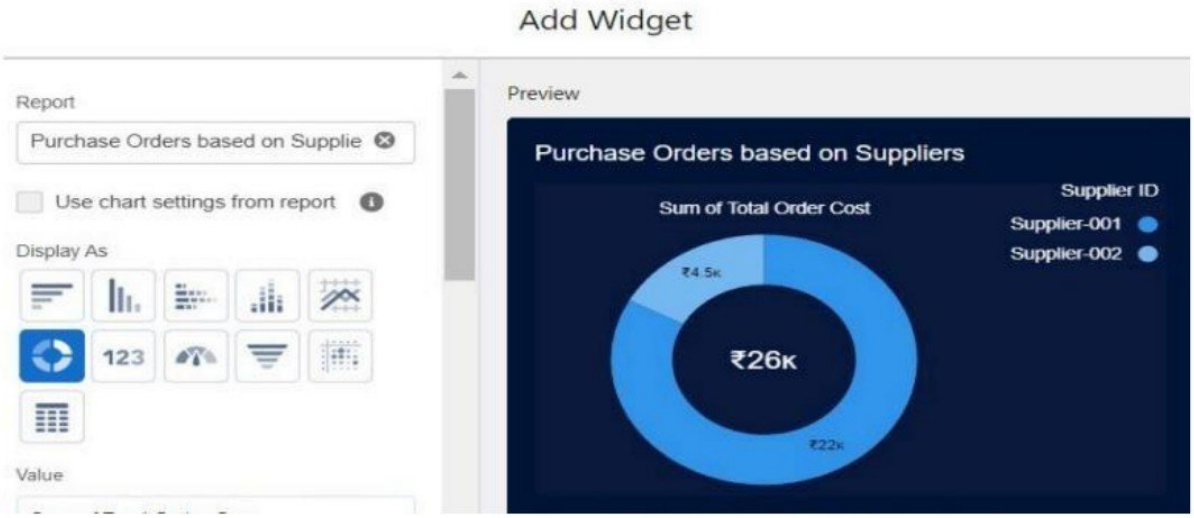
Click +Widget to add a component.

Select the Purchase Orders based on Suppliers report.

Choose a suitable data visualization type (chart, table, etc.) based on your requirement.

Click Add → then Save.





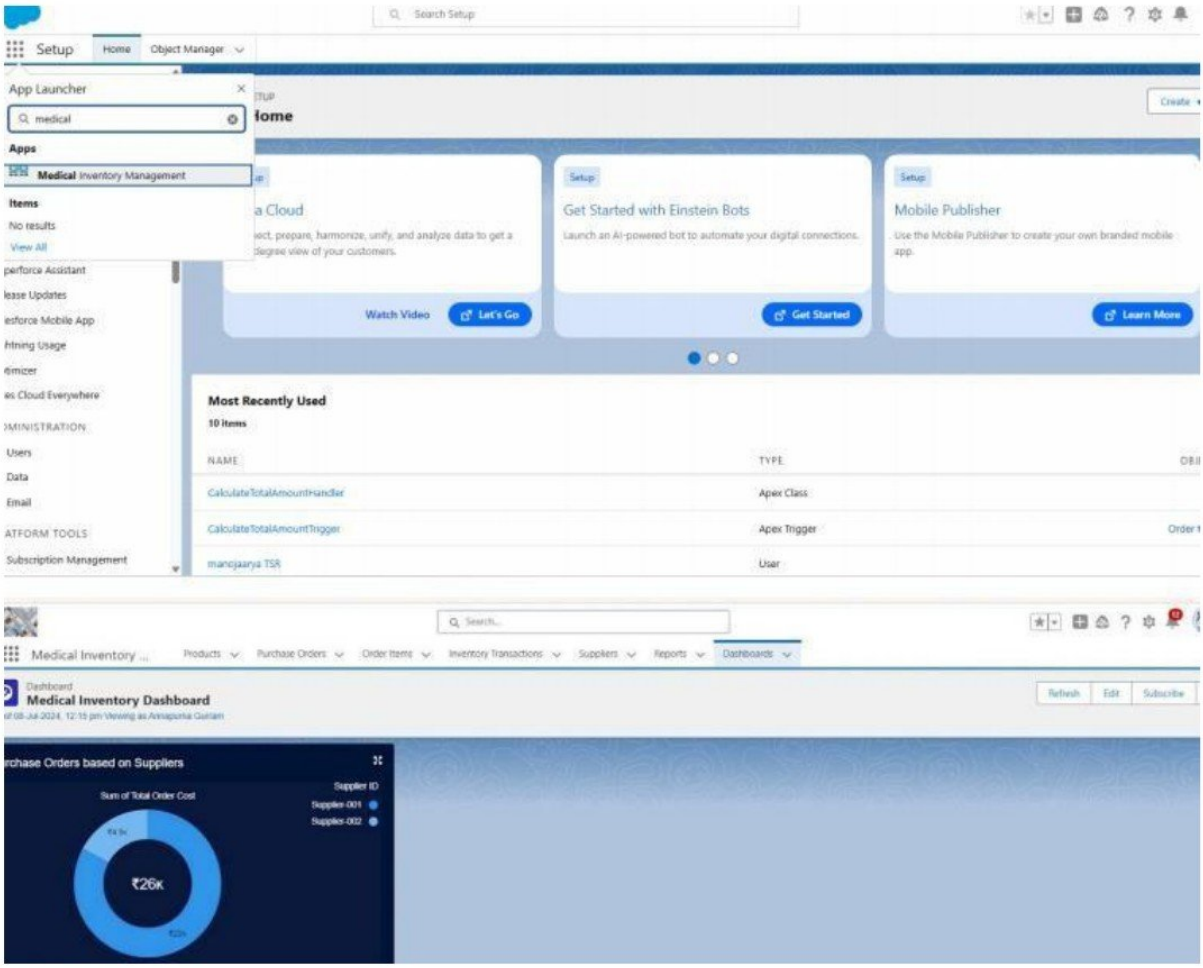
View Dashboard

Click on App Launcher (left-hand side of the screen).

Search for Medical Inventory Management → Click to open the app.

Go to the Dashboard tab.

Click on Medical Inventory Dashboard to view the graphical representation of records.



Conclusion

The Medical Inventory Management System effectively automates and streamlines inventory management in a healthcare environment. By leveraging Salesforce CRM features, the system improves efficiency, ensures data accuracy, and enhances transparency in managing medical supplies. This project highlights the practical application of Salesforce in addressing real-world challenges, as part of the Naan Mudhalvan initiative

Thank you