

Medical Inventory Management

COLLEGE NAME : Sri Krishna Adithya College of Arts and Science

TEAM ID : NM2025TMID22550

TEAM SIZE : 4

TEAM LEADER : BHARATH V

Email: 23bsit209bharathv@skacas.ac.in

Team Member 1: AKASH K

Email : 23bsit203akashk@skacas.ac.in

Team Member 2: ATCHAYA S

Email: 23bsit206atchayas@skacas.ac.in

Team Member 3: DEEPAK A

Email: 23bsit212deepaka@skacas.ac.in

Title: Medical Inventory Management



Project Overview

The **Medical Inventory Management System** is a Salesforce CRM-based application created to handle medical supplies, medicines, and equipment with accuracy and efficiency. It simplifies operations such as purchase order handling, supplier management, and inventory tracking. By automating these processes, the system lowers manual effort, reduces mistakes, and ensures that healthcare organizations always have essential medical resources available.

Objectives

The system has been developed with the following goals:

- Keep all medical inventory information in a single, centralized place.
- Make the purchase order process faster through automation and approvals.
- Record supplier details and track their order history for better accountability.
- Alert users whenever stock is low or products are nearing expiry.
- Provide managers with useful reports and dashboards for smarter decision-making.
- Apply Salesforce features like workflows, validation rules, and triggers.
- Ensure data security by giving different access levels to Admins, Inventory Managers, and Staff members.

Scope of the Project

What is included:

- Tracking of medicines and medical equipment.
- Supplier records and purchase order processing.
- Role-based security for users.
- Salesforce automation (triggers, workflows, approval processes).
- Reports and dashboards for monitoring and insights.

What is not included:

- Billing or payment system integration with hospitals.
- Advanced demand forecasting using AI/ML.
- Offline mobile application development.

Key Modules

1. **Inventory Management** – Add, update, and view medical items.
2. **Supplier Management** – Save supplier details and monitor past orders.
3. **Purchase Order Management** – Automate creation, approvals, and cost calculations.
4. **Stock Monitoring** – Notifications for low quantity or expired items.
5. **Reports & Dashboards** – Summaries and visual reports for managers.
6. **User Management** – Roles and profiles for secure access.

Benefits

- Reduced wastage and better stock control.
- Less manual effort in creating and approving purchase orders.
- Stronger supplier accountability.
- Real-time access to inventory and cost details.

Student Learning Outcomes

1. **Knowledge of Salesforce CRM** – Hands-on customization for practical applications.
2. **Cloud Development Skills** – Creating custom fields, objects, layouts, and relationships.
3. **Business Automation Experience** – Building triggers, workflows, and approval processes.
4. **Data Analytics Skills** – Developing dashboards and reports for inventory insights.
5. **Security Management** – Implementing user profiles with different access permissions.
6. **Problem-Solving Ability** – Addressing real challenges in healthcare inventory.
7. **Documentation & Presentation Skills** – Preparing clear technical reports.

8. **Collaboration & Project Management** – Working effectively with peers to meet deadlines (for team projects).

System Requirements

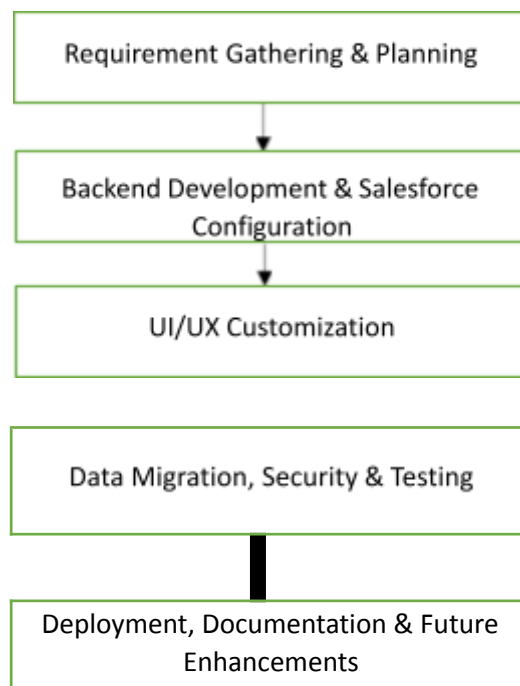
Hardware

- **Processor:** Intel Core i3 / AMD equivalent or higher
- **RAM:** 4 GB minimum (8 GB preferred)
- **Storage:** At least 20 GB free space
- **Display:** 1024×768 or higher
- **Connectivity:** Reliable internet connection

Software

- **OS:** Windows 10/11, Linux, or macOS
- **Browser:** Chrome, Firefox, or Edge (latest versions)
- **Salesforce Edition:** Developer Edition (*Naan Mudhalvan provided*)
- **Optional Tools:** Developer Console, VS Code with Salesforce extensions
- **Documentation Tools:** MS Office, Google Docs

PROJECT PHASES



Phase 1: Requirement Analysis & Planning

-Clear understanding of the requirements collected from healthcare staff, administrators, and inventory managers.

- Well-defined project objectives and scope.
- Drafted preliminary data model and workflow diagrams for suppliers, purchase orders, and medical items.
- Structured project roadmap prepared for smooth execution in later phases.

Phase 2: Salesforce Development – Backend & Configurations

Milestone 1: Salesforce Developer Account Setup

Activity 1: Creating a Developer Account

- Registered for a Salesforce Developer Edition account through the official signup page: <https://developer.salesforce.com/signup>
- Configured the developer org to serve as the working environment for backend customization and application development.

Milestone 2: Creating Custom Objects

Activity 1: Creating a Product Object **Steps**

to create the custom object *Product*:

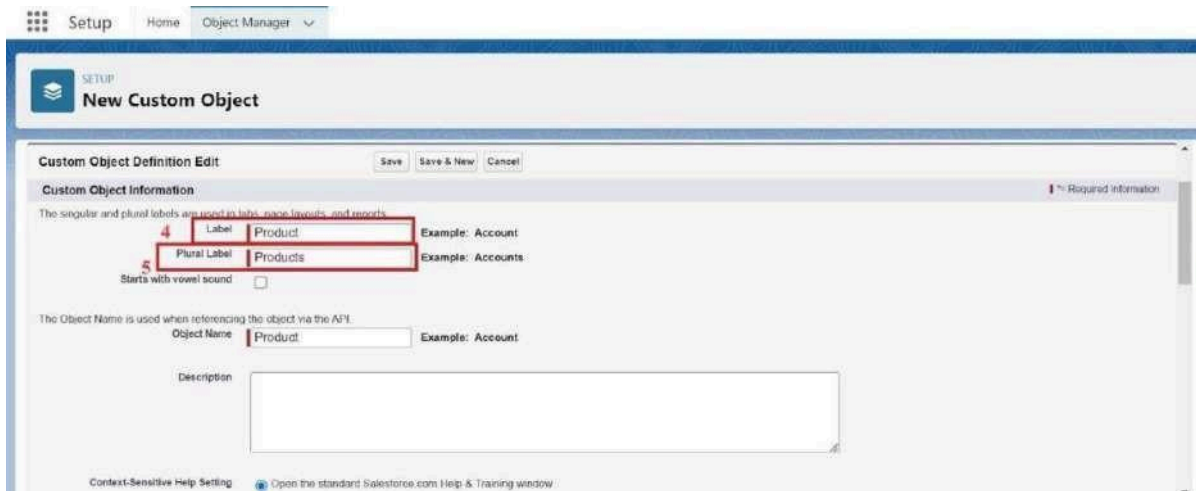
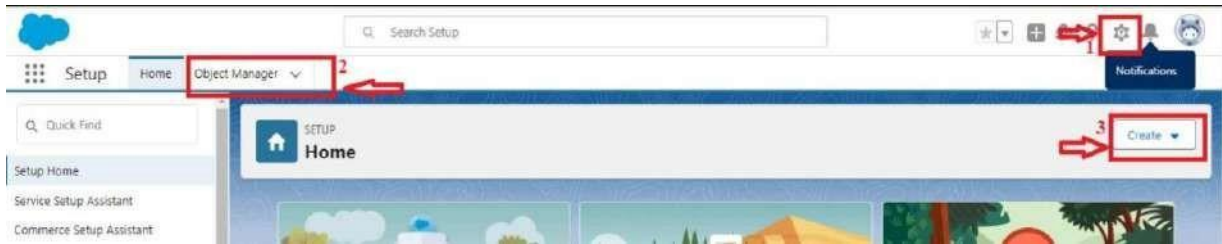
- Navigate to Setup in Salesforce.
- Select Object Manager from the menu.
- Click Create → Custom Object.

Provide the following details:

- Label Name: Product
- Plural Label: Products
- Record Name: Product ID
- Data Type: Text

Enable the following options:

- Allow Reports
- Allow Search
- Click Save and New to complete the process.



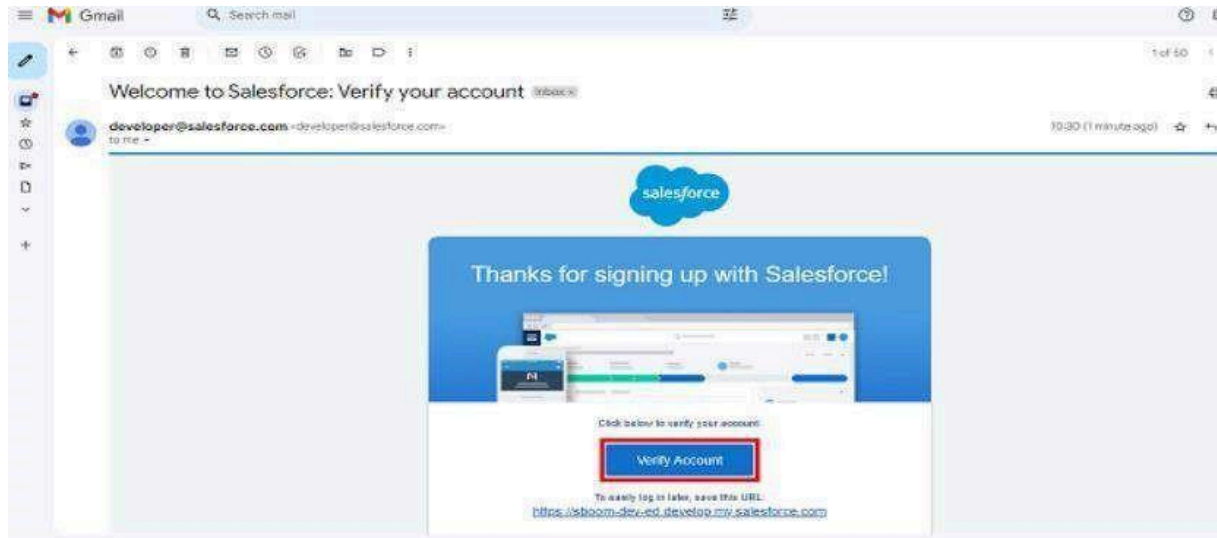
Activity 2: Account Activation

Steps to verify and activate the Salesforce Developer Account:

-Open the inbox of the email address used during signup and look for the verification email (may take 5–10 minutes to arrive).

-Click on the “Verify Account” link provided in the email.

-Your Salesforce Developer account will be successfully activated and ready for use.

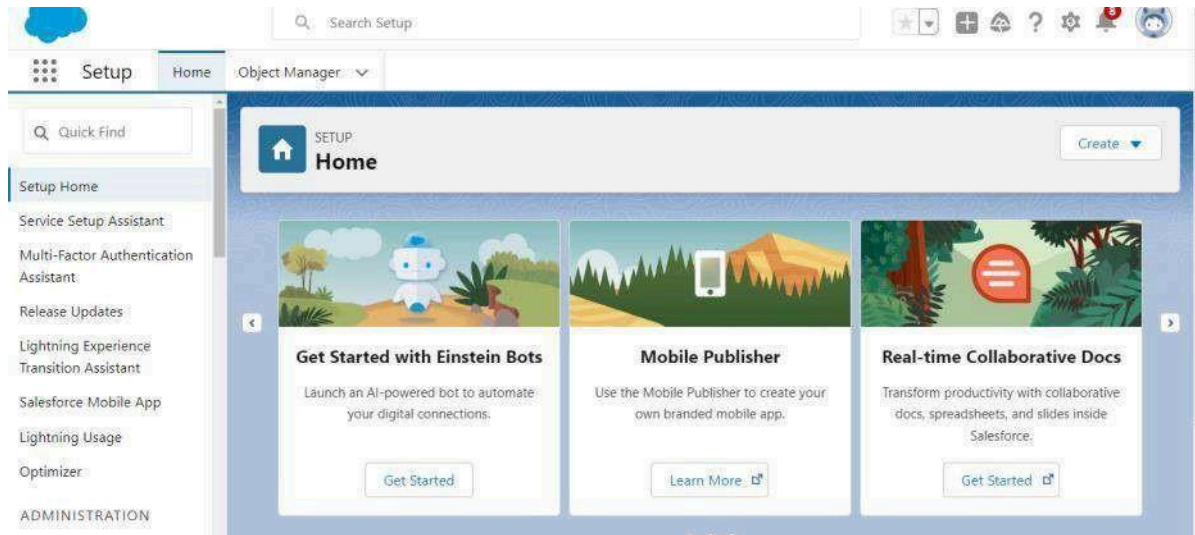


-Open the inbox of the email address used during signup (the email may take 5–10 minutes to arrive).

-Click on the “Verify Account” link provided in the email.

-Set a new password, choose and answer a security question, then click Change Password.

-After completing the setup, you will be redirected to your Salesforce Setup Home Page.



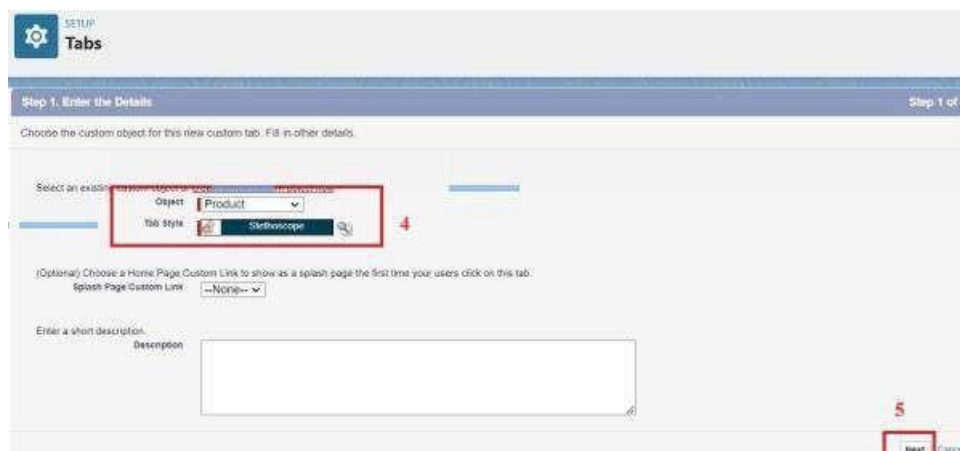
Milestone 3: Creating Tabs

Activity 1: Creating a Tab for the Product Object

Procedure:

1. Navigate to **Setup** and type **Tabs** in the Quick Find search bar.

2. Select **Tabs** from the available options.
3. Under **Custom Object Tabs**, click **New**.
4. From the list, choose **Product** as the object and select an appropriate **Tab Style**.
5. Click **Next**. On the **Add to Profiles** page, retain the default settings and click **Next** again.
6. On the **Add to Custom Apps** page, uncheck the option **Include Tab**.
7. Make sure the option **Append tab to user's existing personal customizations** is selected.
8. Finally, click **Save** to complete the tab creation process.



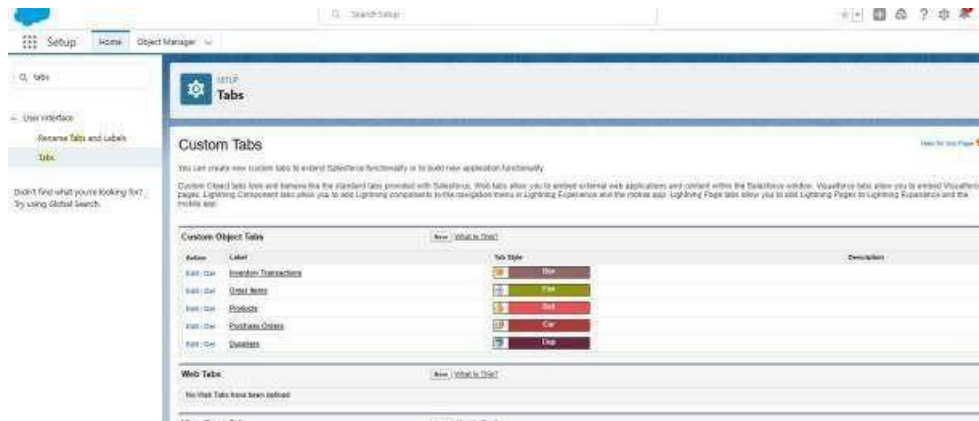
Milestone 3: Creating Tabs

Activity 2: 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.



Milestone 4: The Lightning App

Activity 1: Creating a Lightning App for Medical Inventory Management

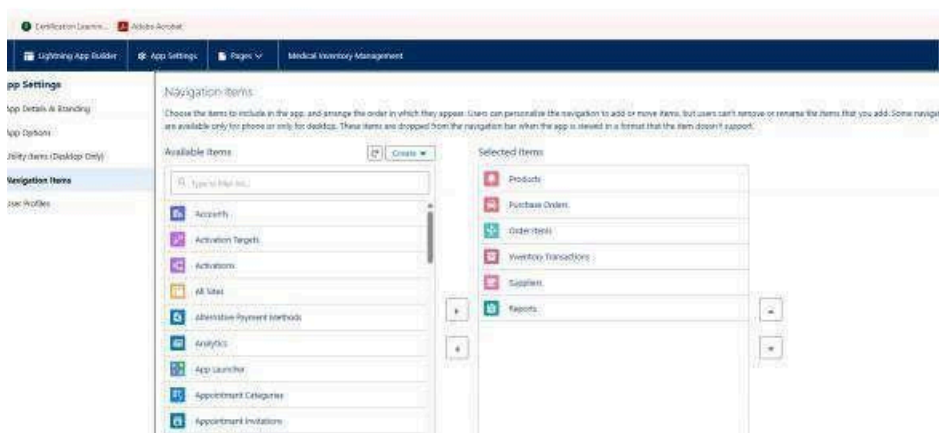
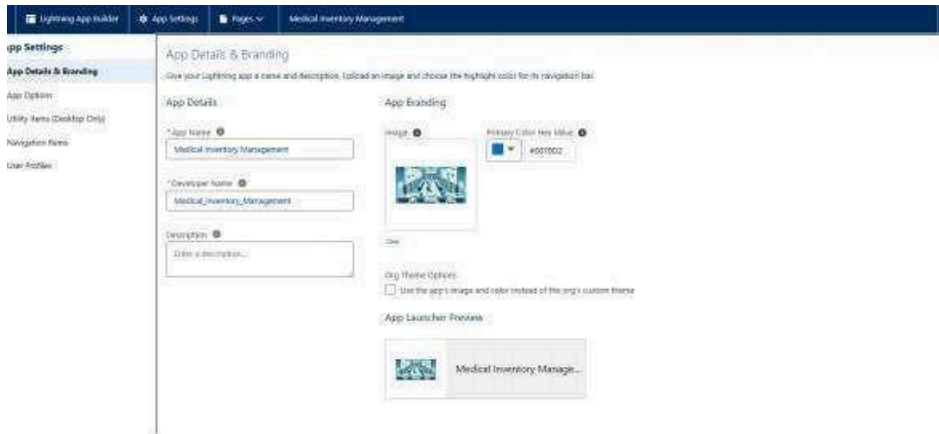
Procedure:

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**.



Milestone 5: Fields

Activity 1: Creating a Text Field in the Product Object

Steps:

1. Click the **gear icon** and select **Setup** (opens in a new tab).
2. In Setup, go to the **Object Manager** tab.
3. Select the **Product** custom object.
4. From the left navigation, click **Fields & Relationships**.
5. Click **New**.
6. Choose **Text** as the field type and click **Next**.
7. 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.

Setup - Object Manager - Product - New Custom Field

Step 2: Enter the details

Field Label: Product Name

Length: 255

Field Name: Product_Name

Description:

Help Text:

Required: ☐ Always requires a value in this field in order to save a record

Auto add to custom report type: ☒ Add this field to existing custom report types that contain this entity

Milestone 5: Fields

Activity 2: Creating a Text Area Field in the Product Object

Steps:

- Click the gear icon and select Setup (opens in a new tab).
- In Setup, go to the Object Manager tab.
- Select the Product custom object.
- From the left navigation, click Fields & Relationships.
- Click New.
- Choose Text Area as the field type and click Next.

Enter the following details:

Field Label: Product Description

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

Setup - Object Manager - Product - New Custom Field

Step 2: Enter the details

Field Label: Product Description

Field Name: Product_Description

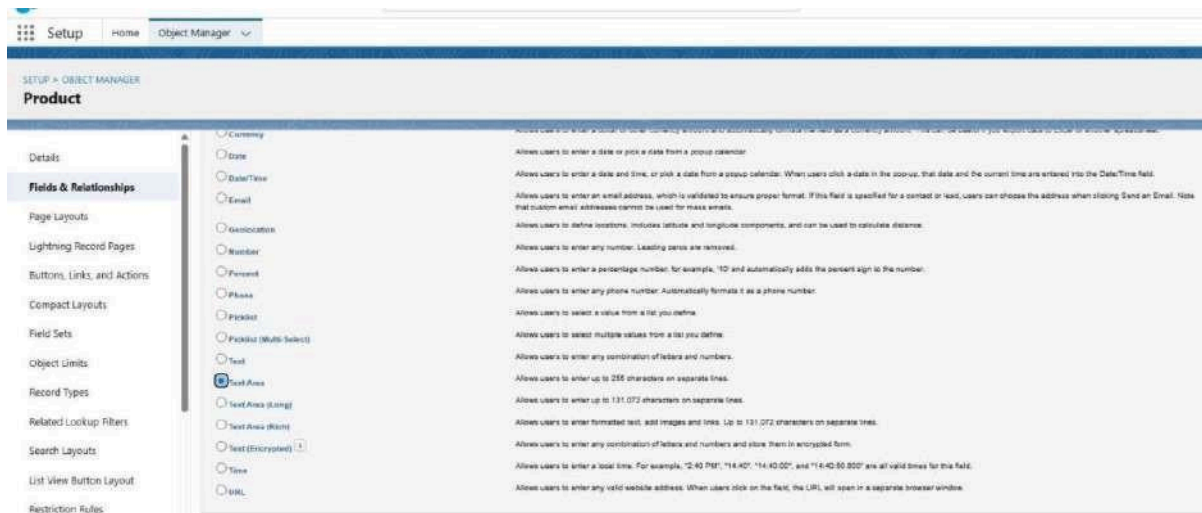
Description:

Help Text:

Required: ☐ Always requires a value in this field in order to save a record

Auto add to custom report type: ☒ Add this field to existing custom report types that contain this entity

Default Value: [Show Formula Editor](#)



Activity 3: 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.
- Choose Number as the data type and click Next.

Enter the details:

Field Label: Current Stock Level

Length: 18

Decimal Places: 0

Click Next → Next → Save to finish creating the field.

Activity 4: 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.

The screenshot shows the Salesforce 'New Custom Field' setup page for a Product object. The page is titled 'Step 2. Enter the details' and shows fields for 'Field Label' (Unit Price), 'Length' (16), 'Decimal Places' (2), 'Field Name' (Unit_Price), 'Description', and 'Help Text'. The 'Required' checkbox is checked. The left sidebar shows the 'Fields & Relationships' menu.

Activity 5: 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.

The screenshot shows the Salesforce Setup interface. The left sidebar has a menu with 'Fields & Relationships' selected. The main content area is titled 'Product' and 'New Custom Field'. It shows 'Step 2 of 4: Enter the details'. The 'Field Label' is 'Unit Price', 'Field Name' is 'Unit_Price', and 'Length' is 16. The 'Required' checkbox is checked. The 'Next' button is highlighted.

Activity 6: Creating a Date Field 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 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.

The screenshot shows the Salesforce Setup interface. The left sidebar has a menu with 'Fields & Relationships' selected. The main content area is titled 'Purchase Order' and 'New Custom Field'. It shows 'Step 2 of 4: Enter the details'. The 'Field Label' is 'Order Date', 'Field Name' is 'Order_Date', and 'Length' is 10. The 'Required' checkbox is checked. The 'Next' button is highlighted.

Activity 7: Creating a Roll-Up Summary Field in Purchase Order object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Purchase Order) 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

Data Type

Select one of the data types below.

☐ None Selected

☐ Auto Number

☐ Formula

☒ Roll-Up Summary 4

Purchase Order
New Custom Field

Help for this Page

Step 2. Enter the details Step 2 of 5

Previous **Next** Cancel

Field Label: Order Count 5

Field Name: Order_Count

Description:

Help Text:

Auto add to custom report type ☒ Add this field to existing custom report types that contain this entity

Previous Next Cancel

Activity 8: 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 **Order Item** 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

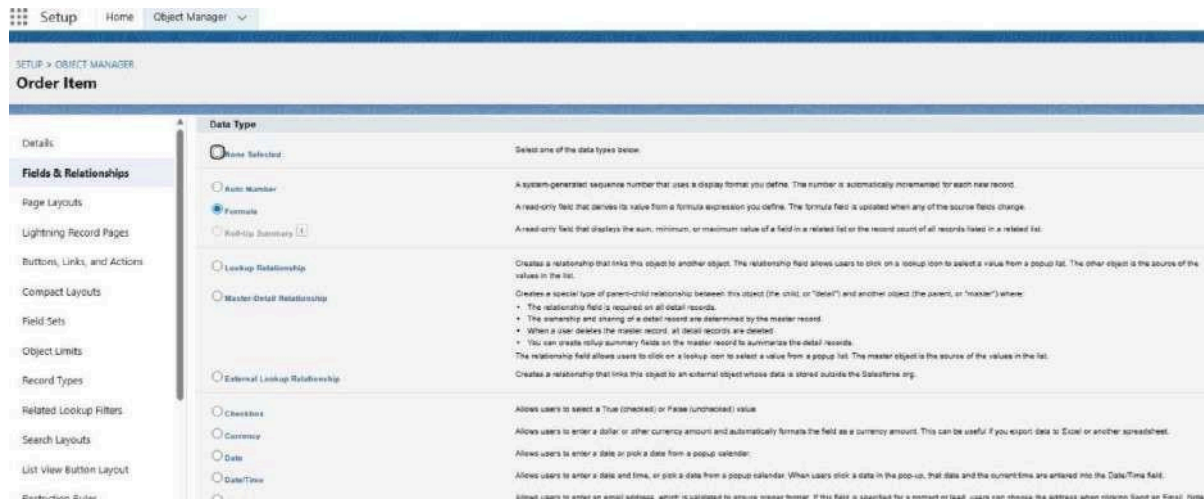
Formula Return Type: Currency

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.



Activity 9: 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 Formula as the data 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.

Activity 10: 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.

The screenshot shows the 'New Custom Field' wizard in Salesforce, specifically for the 'Order Item' object. The wizard is at 'Step 2: Enter the details'. The 'Field Label' is 'Transaction Type'. The 'Values' section shows a list of values: 'Receipt', 'Issue', and 'Adjustment'. The 'Restrict picklist to the values defined in the value set' checkbox is checked. The 'Field Name' is 'Transaction_Type'.

Activity 11: 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.

The screenshot shows the 'New Custom Field' wizard for the 'Order Item' object. The left sidebar lists navigation 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, Search Layouts, List View Button Layout, and Restriction Rules. The main content area is titled 'Step 2: Choose output type' and 'Step 2 of 5'. It shows 'Field Label' as 'Order Cost' and 'Field Name' as 'Order_Cost'. There is a checkbox 'Add this field to existing custom report types that contain this entity' which is checked. Below this is the 'Formula Return Type' section with radio buttons for 'None Selected', 'Checkbox', 'Currency' (selected), 'Date', 'DateTime', and 'Number'. To the right of these options is a list of formulas and their descriptions, such as 'Calculate a boolean value', 'Calculate a dollar or other currency amount', 'Calculate a date', 'Calculate a datetime', and 'Calculate a numeric value'.

Activity 12: 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.

The screenshot shows the 'New Custom Field' wizard for the 'Supplier' object. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Step 2: Enter the details' and 'Step 2 of 4'. It shows 'Field Label' as 'Phone Number' and 'Field Name' as 'Phone_Number'. There is a 'Description' field and a 'Help Text' field. Below these are checkboxes for 'Required' (checked) and 'Add this field to existing custom report types that contain this entity' (checked). At the bottom, there is a 'Default Value' field with a 'Show Formula Editor' link.

Activity 13: 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.

Milestone 6: Page Layout Customization

Activity 1: Editing a Page Layout in the Product Object

Steps:

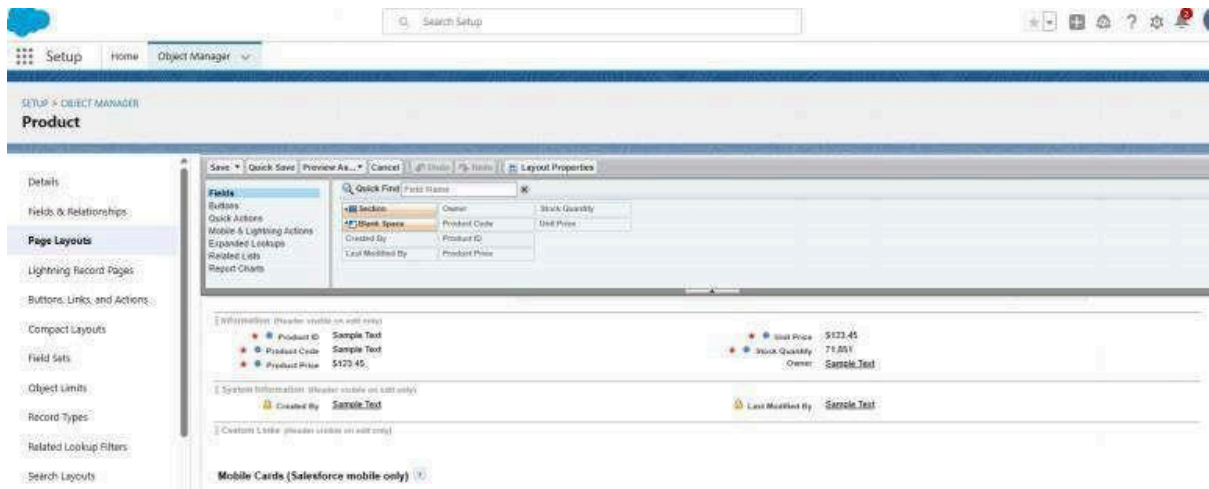
-Go to Setup → click on Object Manager.

-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



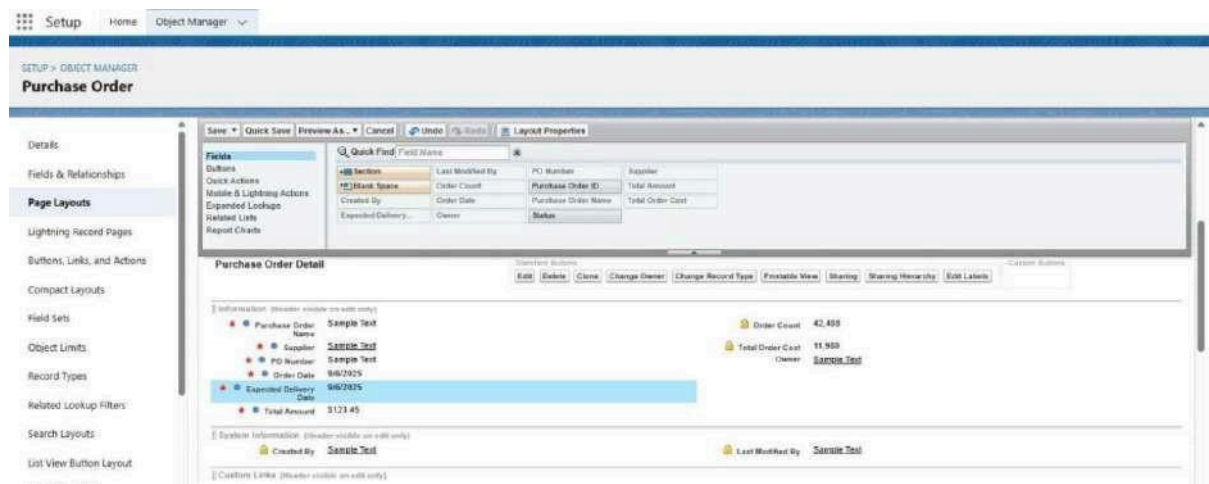
Activity 2: Editing a Page Layout in the Purchase Order Object

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.



Activity 3: Editing a Page Layout in the Order Item Object

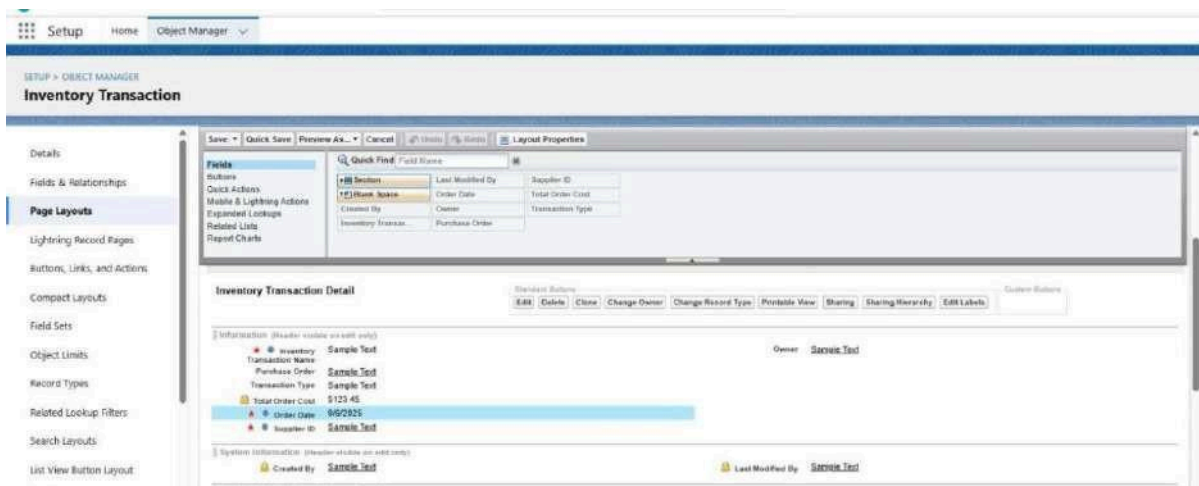
Steps:

- 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.

Activity 4: 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.



Activity 5: 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.

Milestone 7: Compact Layouts

Activity 1: 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.



Activity 2: Creating a Compact Layout for the Purchase Order Object

Steps:

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



Milestone 8: Validation Rules

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

Steps:

1. Go to **Setup** → click on **Object Manager**.
2. In the Quick Find 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. In the formula editor, enter the error condition formula:
6. $(\text{Expected_Delivery_Date_c} - \text{Order_Date_c}) > 7$

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

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



Milestone 9: Profiles

Activity 1: 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.

Activity 2: 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.

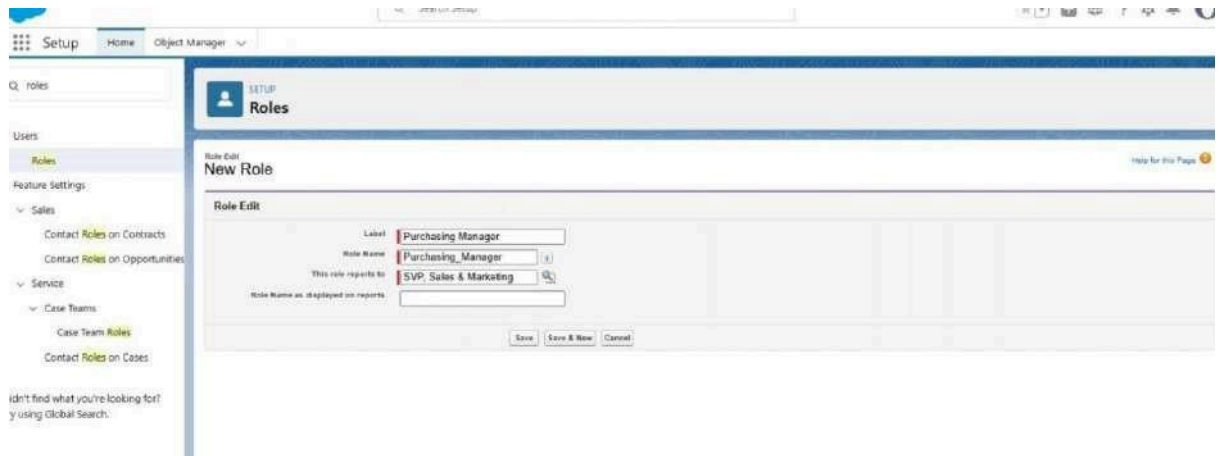


Milestone 10: Roles

Activity 1: 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:
 - Label: Purchasing Manager
 - The Role Name will auto-populate.
5. Click Save to create the role.



Activity 2: 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.

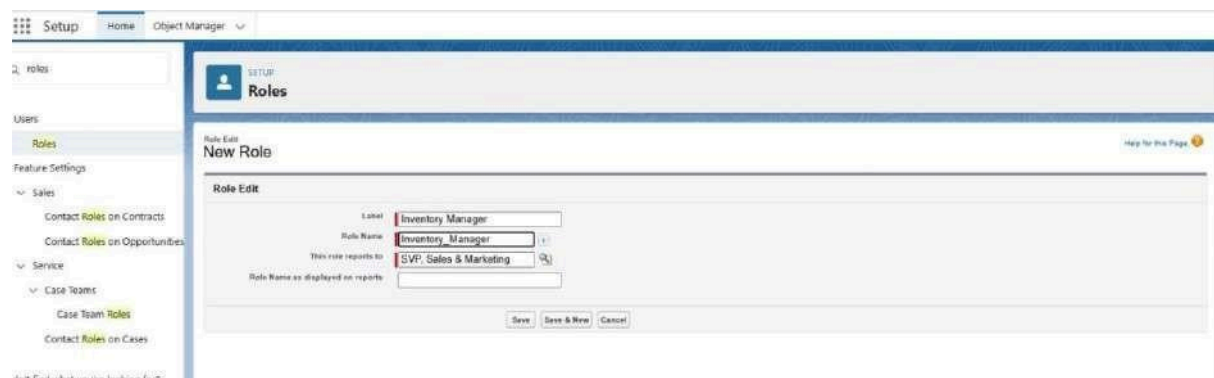
Under the SVP, Sales & Marketing role, click Add Role.

Enter the following details:

Label: Inventory Manager The

Role Name will auto-populate.

Click Save to create the role.



Milestone 12: Permission Sets

Activity 1: Creating a Permission Set

Steps:

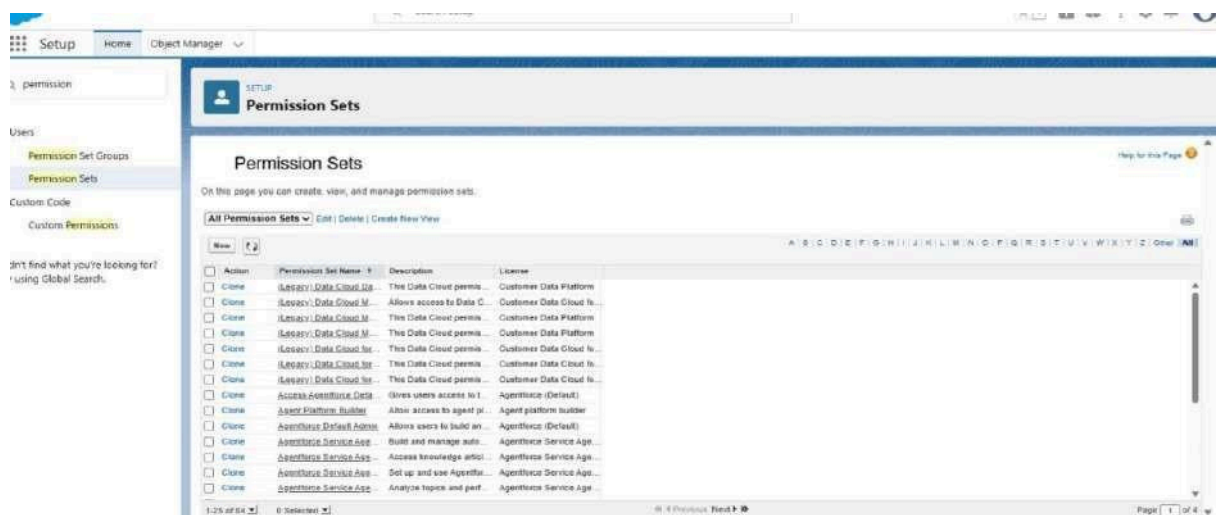
Go to Setup → type Permission in the Quick Find box → select Permission Sets.

Click New.

Enter the following details:

Label: Purchase Manager Create Access

Click Save to create the permission set.



Milestone 13: Flows

Activity 1: 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

7. Click the “+” icon → select Get Records.

8. Enter Label: Get Purchase Record.

9. Select Object: Purchase Order.

10. For Condition Requirements, choose All Conditions Are Met (AND).

11. 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. In Flow Builder, click Manager → New Resource.

15. Resource Type: Variable

16. API Name: ActualDeliveryDate

17. Data Type: Date → click Done.

Assignment Element

18. Drag and drop Assignment from the 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. Drag and drop Update Records → connect it to the Assignment element.

23. Enter Label: Updating Purchase Order.

24. How to Find Records to Update: Use the Purchase Order record that triggered the flow.

25. Filter Conditions: None – Always Update Record.

26. Set Field Values:

Field: Actual_Delivery_Date_c

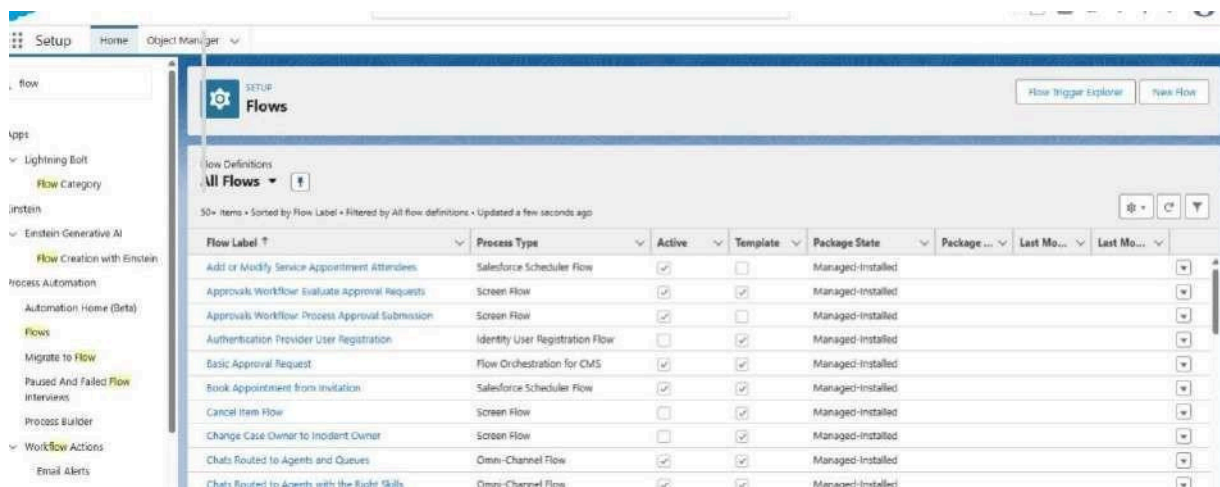
Value: {!ActualDeliveryDate}

Click Done.

Save and Activate Flow

28. Save the flow as Actual Delivery Date Updating.

29. Activate the flow.



Milestone 14: Triggers

Activity 1: 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

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

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

Step 3: Create the Apex Trigger

In Developer Console, go to **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 to **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)
```

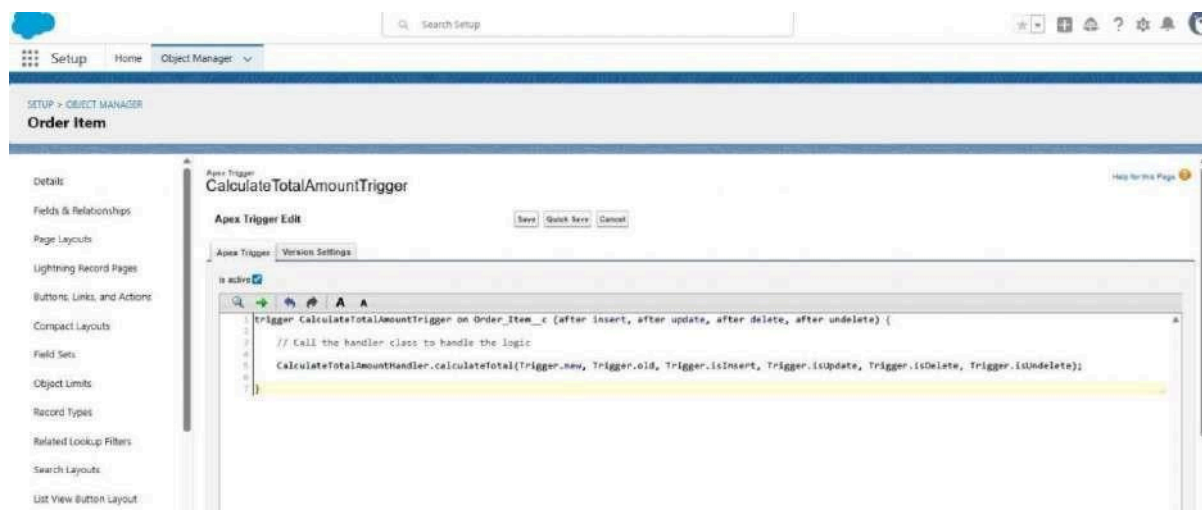
```
;
purchaseToUpdate.add(purchaseOrder);
}

// 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.



Milestone 15 - Reports

Activity 1: Create a Purchase Orders based on Suppliers(Summary) Report

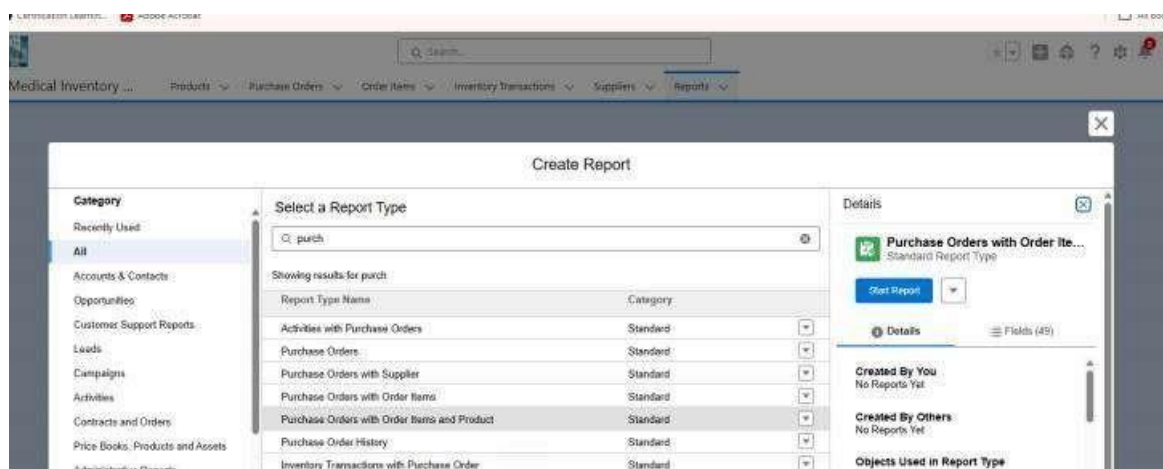
1. Click App Launcher
2. Select Medical Inventory Management App
3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders Click Start report.

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.

Activity 2: 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



Milestone 16: Dashboards

Activity 1: Create Dashboard

Open the Dashboards tab within the Medical Inventory Management application.

Click New Dashboard.

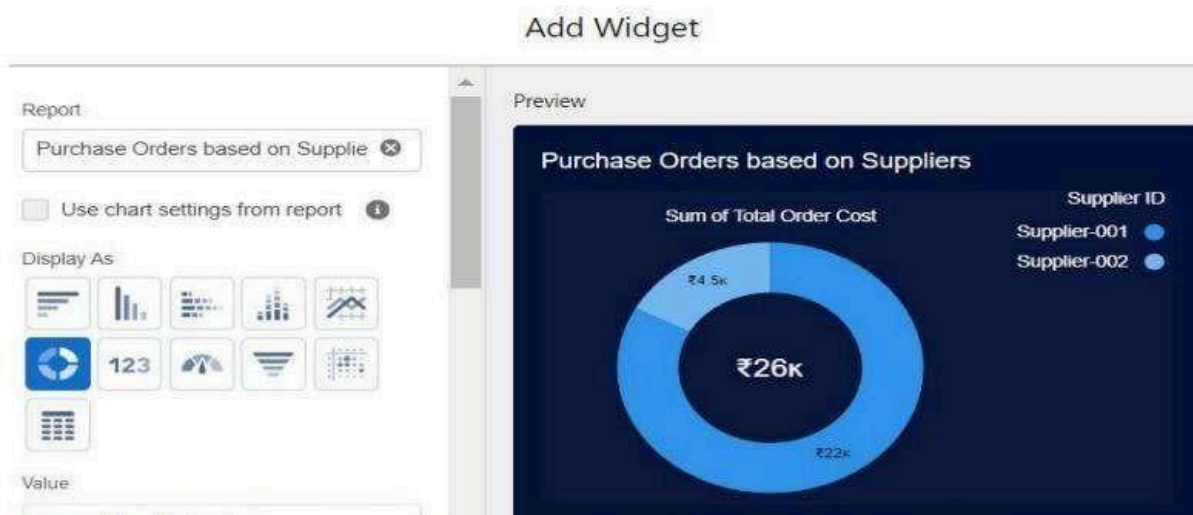
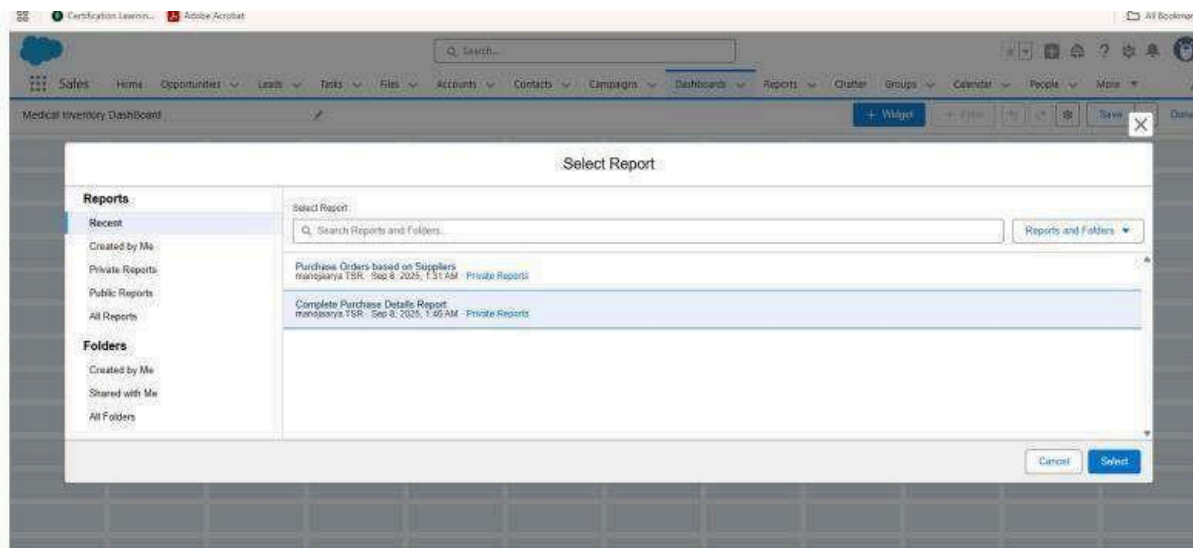
Enter the Name: Medical Inventory Dashboard → Click Create.

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.



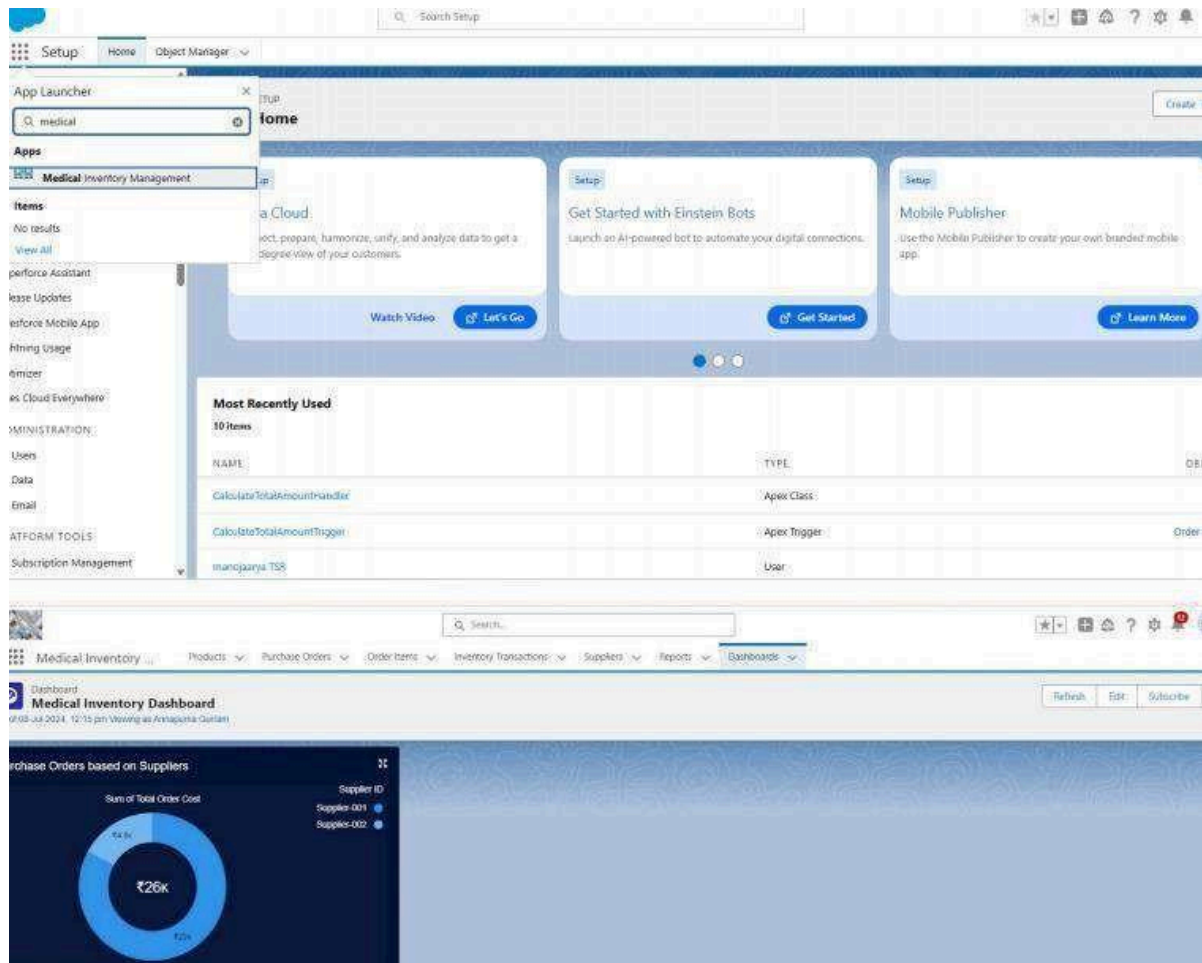
Activity 2: View Dashboard

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.