

Medical Inventory Management

College Name: Sri Krishna Adithya College of Arts and Science

College Code: Bruag

TEAM ID: NM2025TMID22191

TEAM MEMBERS:

Team Leader Name: DEEPIKA G

Email: 23bsit112deepikag@skacas.ac.in

Team Member1: BARATHI S

Email: 23bsit110barathis@skacas.ac.in

Team Member: BHUVANESHWARI V

Email: 23bsit111bhuvaneshwariv@skacas.ac.in

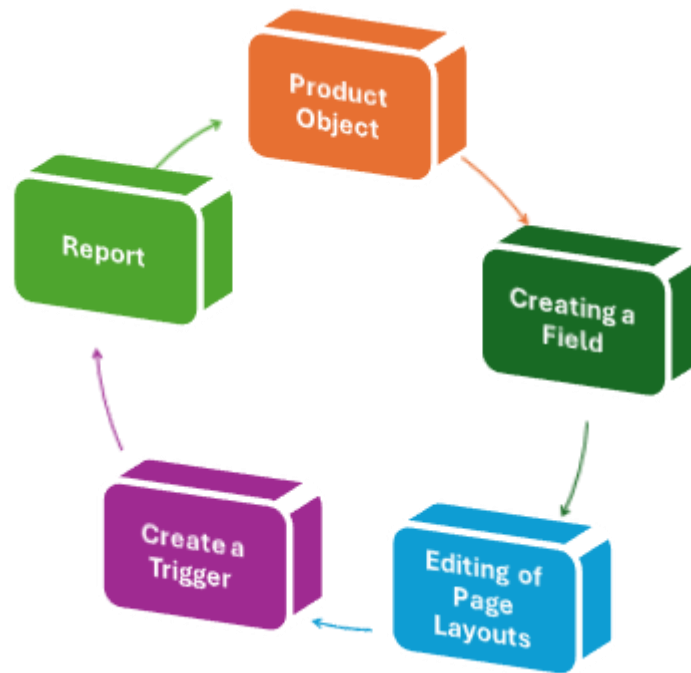
Team Member: DIVYA R

Email: 23bsit116divyar@skacas.ac.in

1.INTRODUCTION

1.1 Project Overview

The Medical Management System 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.

salesforce.com/form/developer-signup/?d=pb

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

Job title Work email

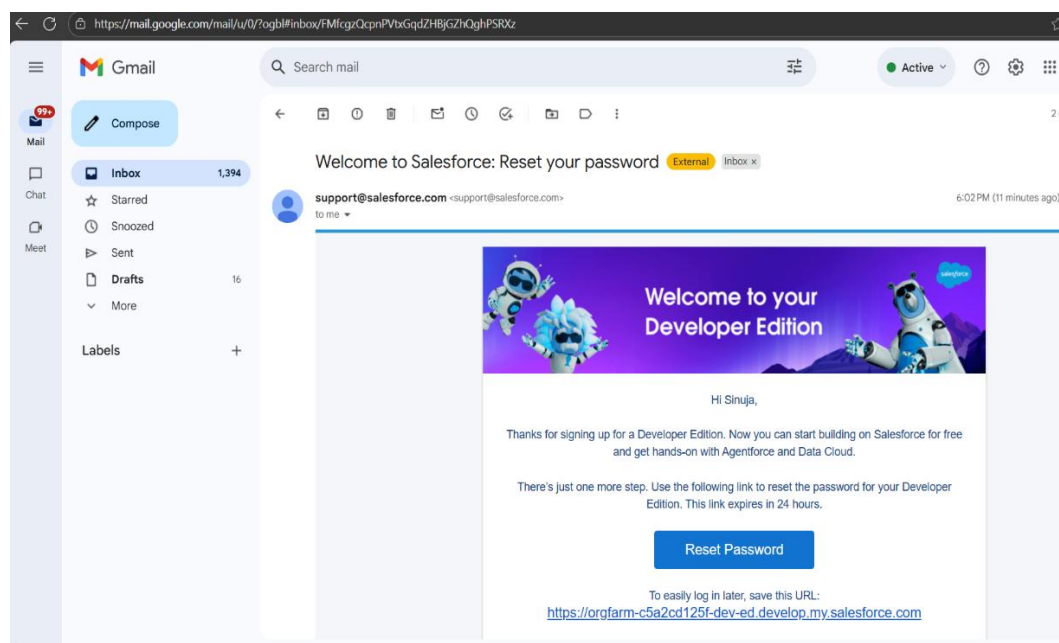
Company Country/Region

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.

By registering I confirm that I have read and agree to the Privacy Statement.

2.2 Account Activation:



2.3 Verify Account

← ↻ 🔒 https://orgfarm-c5a2cd125f-dev-ed.develop.my.salesforce.com/_ui/system/security/ChangePassword?retURL=%2Fhome%2Fhome.jsp&fromFrontdoor=1&setupid=ChangePassword

Change Your Password

Enter a new password for
23bsit156sinujas857@agentforce.com. Make sure to include at least:

- ☐ 8 characters
- ☐ 1 letter
- ☐ 1 number

* New Password

* Confirm New Password

* Security Question
▼ In what city were you born?

* Answer

*=required

Change Password

Password was last changed on 9/17/2025, 5:32 AM.

2.4 Salesforce Setup Page

← ↻ 🔒 https://orgfarm-c5a2cd125f-dev-ed.develop.lightning.force.com/lightning/setup/SetupOneHome/home

Setup Home Object Manager

Quick Find

Setup Home

- Salesforce Go
- Service Setup Assistant
- Commerce Setup Assistant
- Field Service Setup Home (Beta)
- Hyperforce Assistant
- Release Updates
- Salesforce Mobile App
- Lightning Usage
- Optimizer
- Sales Cloud Everywhere

ADMINISTRATION

- > Users
- > Data
- > Email

PLATFORM TOOLS

- > Subscription Management

SETUP Home Create

Setup Data Cloud Watch Video Let's Go

Setup Get Started with Einstein Bots Get Started

Setup Mobile Publisher Learn More

Most Recently Used

0 items

NAME	TYPE	OBJECT
------	------	--------

2.5 Objects Created: Product

Setup Home Object Manager

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more!](#) [Don't show this message again](#)

Custom Object Definition Edit

Save Save & New Cancel

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label Example: Account

Plural Label Example: Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name Example: Account

Description

Context-Sensitive Help Setting ☒ Open the standard Salesforce.com Help & Training window
☐ Open a window using a Visualforce page

Content Name

Setup Home Object Manager

New Custom Object

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name Example: Account Name

Data Type Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

☒ Allow Reports
☒ Allow Activities
☒ Track Field History
☐ Allow in Chatter Groups
☐ Enable Licensing

Object Classification

When these settings are enabled, this object is classified as an Enterprise Application object. When these settings are disabled, this object is classified as a Light Application object. [Learn more](#)

☒ Allow Sharing
☒ Allow Bulk API Access
☒ Allow Streaming API Access

Deployment Status

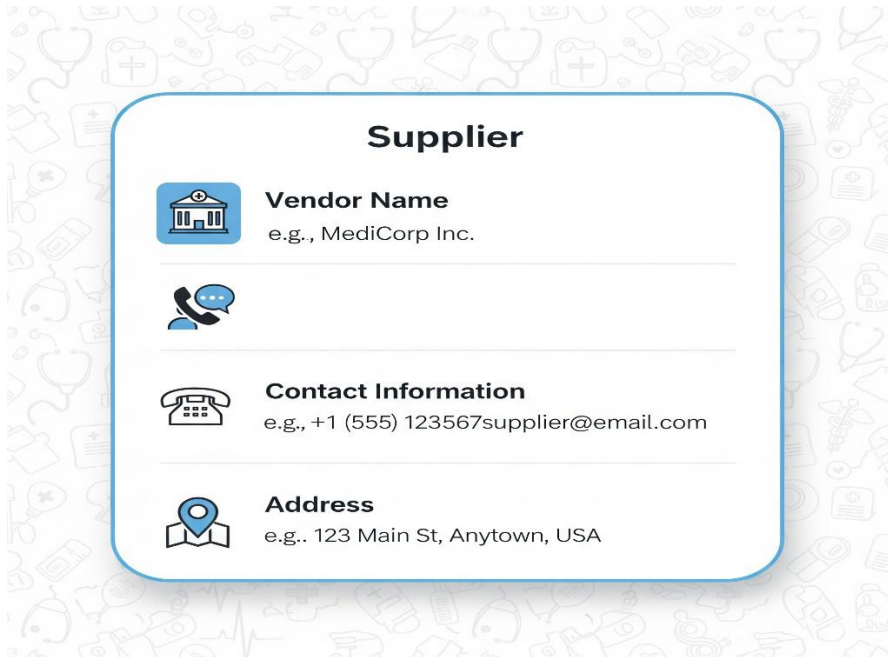
☐ In Development
☒ Deployed [What is this?](#)

Search Status


When this setting is enabled, your users can find records of this object type when they search. [Learn more](#)


☒ Allow Search


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




Supplier

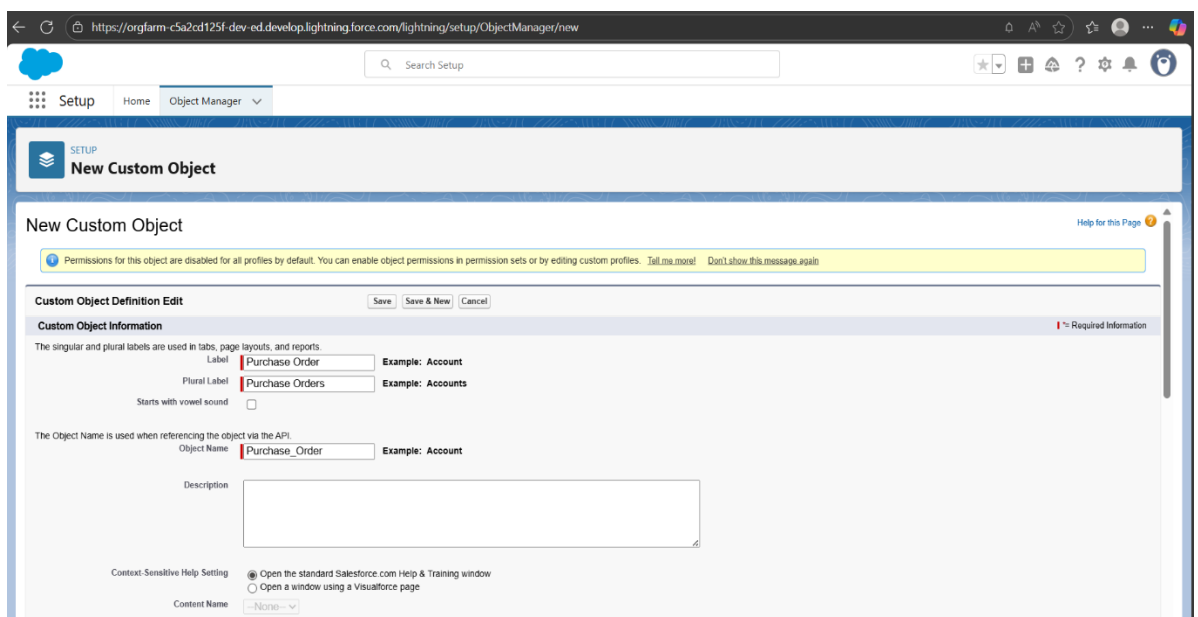
 **Vendor Name**
e.g., MediCorp Inc.

 _____

 **Contact Information**
e.g., +1 (555) 123567supplier@email.com

 **Address**
e.g., 123 Main St, Anytown, USA

- **Purchase Order:** Tracks orders placed with suppliers, linking them to specific medicines and quantities.



https://orgfarm-c5a2cd125f-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new

Setup Home Object Manager

SETUP
New Custom Object

New Custom Object Help for this Page

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more!](#) [Don't show this message again](#)

Custom Object Definition Edit Save Save & New Cancel

Custom Object Information Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label **Example:** Account

Plural Label **Example:** Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name **Example:** Account

Description

Context: Sensitive Help Setting ☒ Open the standard Salesforce.com Help & Training window
☐ Open a window using a Visualforce page

Content Name

2.3 Configurations Done

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

Creating Remaining Tabs

Procedure:

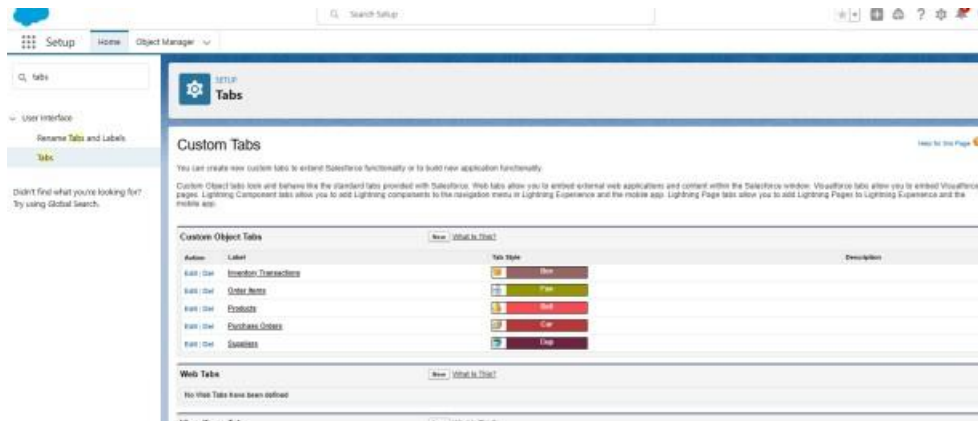
- Create tabs for the following objects:
- Purchase Order

-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

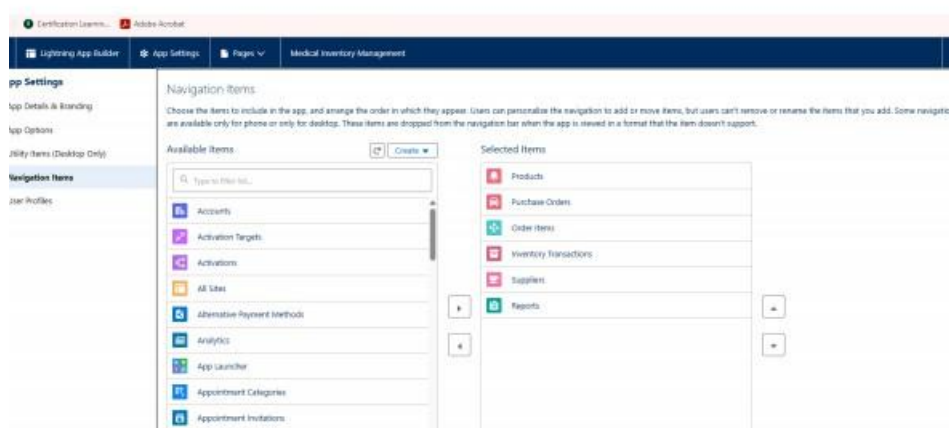
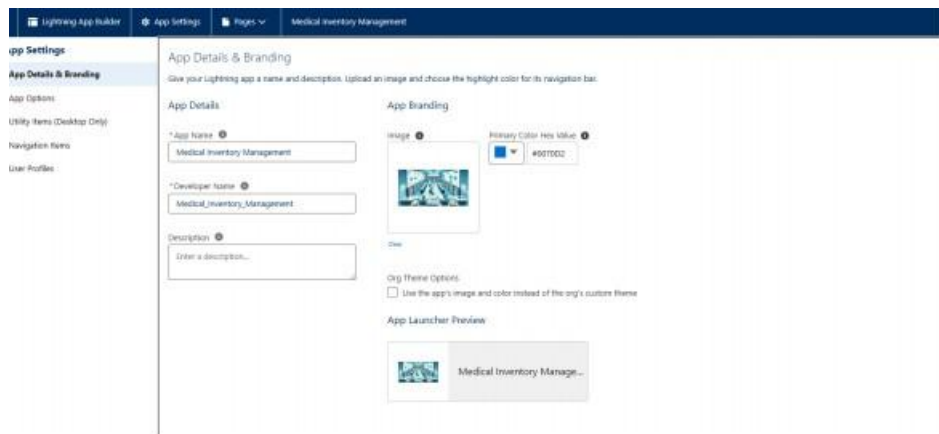
Procedure:

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**.

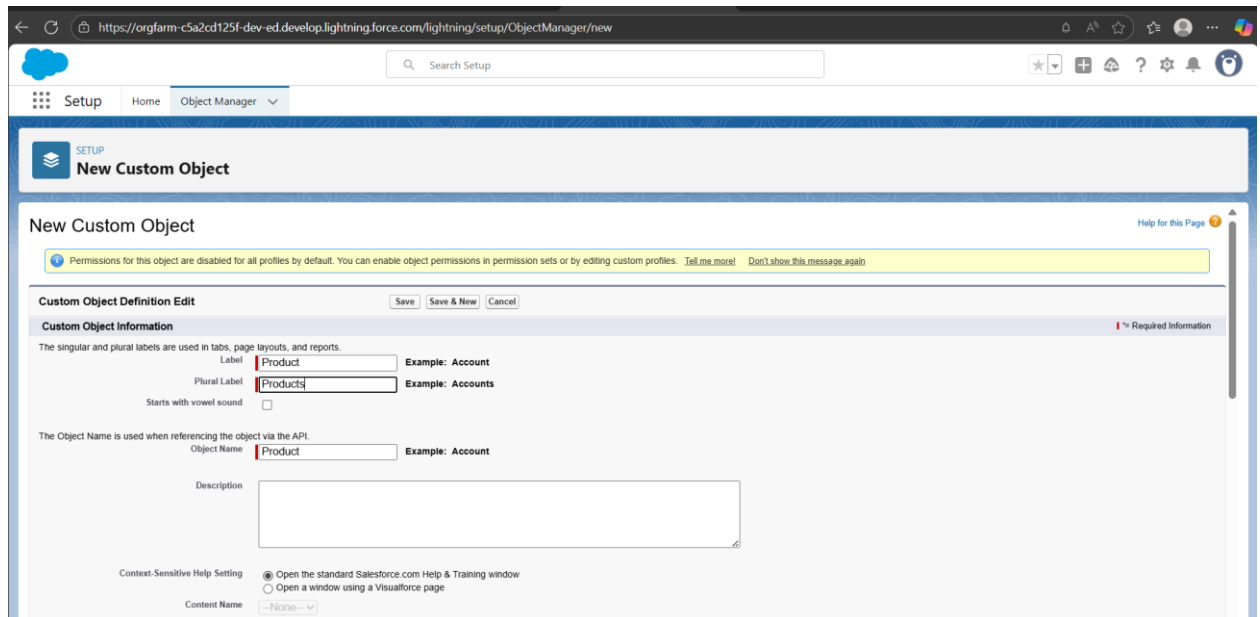


Fields

Creating a Text Field in the Product Object

Steps:

1. 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.



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.

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:

Decimal Places: 2

Mark the field as Required.

Click Next → Next → Save.

The screenshot shows the Salesforce Setup interface. The left-hand navigation menu is open, showing 'Fields & Relationships' under the 'Product' object. The main content area is titled 'New Custom Field' and shows 'Step 2. Enter the details'. The field label is 'Unit Price'. The length is set to 16, and the decimal places are set to 2. The field name is 'Unit_Price'. The description is empty. The help text is empty. The 'Required' checkbox is checked. The 'Auto add to custom report type' checkbox is checked. The default value is 'Show Formula Editor'.

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.

Setup > OBJECT MANAGER
Purchase Order

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules

Purchase Order
New Relationship

Step 3. Enter the label and name for the lookup field

Field Label: Supplier objects

Field Name: Supplier_objects

Description:

Help Text:

Child Relationship Name: Purchase_Orders

Required: ☒ Always require a value in this field in order to save a record
☐ Clear the value of this field. You can't choose this option if you make this field required.

What to do if the lookup record is deleted?:
☒ Don't allow deletion of the lookup record that's part of a lookup relationship.
☐ Add this field to existing custom report types that contain this entity.

Auto add to custom report type: ☒

Lookup Filter

Previous Next Cancel

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.

Setup > OBJECT MANAGER
Purchase Order

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules

Purchase Order
New Custom Field

Step 2. Enter the details

Field Label: Order Date

Field Name: Order_Date

Description:

Help Text:

Required: ☐ Always require a value in this field in order to save a record
☒ Add this field to existing custom report types that contain this entity.

Auto add to custom report type: ☒

Default Value: Show Formula Editor

Use formula syntax. Enclose text and picklist value API names in double quotes. "New_Text" includes numbers without quotes. (20) shows percentages as decimals. (0.10), and express date calculations in the standard format. (Today) + 7. To reference a field from a Custom Metadata type record user: SCustomMetadataType__mdc RecordAPIName Field__c

Previous Next Cancel

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

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The 'Object Manager' dropdown is selected, showing 'Purchase Order'. The left sidebar contains a list of setup categories, with 'Fields & Relationships' selected. The main content area displays the 'Fields & Relationships' section for the 'Purchase Order' object, showing 6 items sorted by Field Label. A table lists the fields with their labels, names, data types, controlling fields, and indexed status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Order Date	Order_Date__c	Date		
Order_Count	Order_Count__c	Master-Detail(Order Item)		✓
Purchase Order ID	Name	Text(80)		✓
Supplier ID	Supplier_ID__c	Lookup(Supplier)		✓

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:

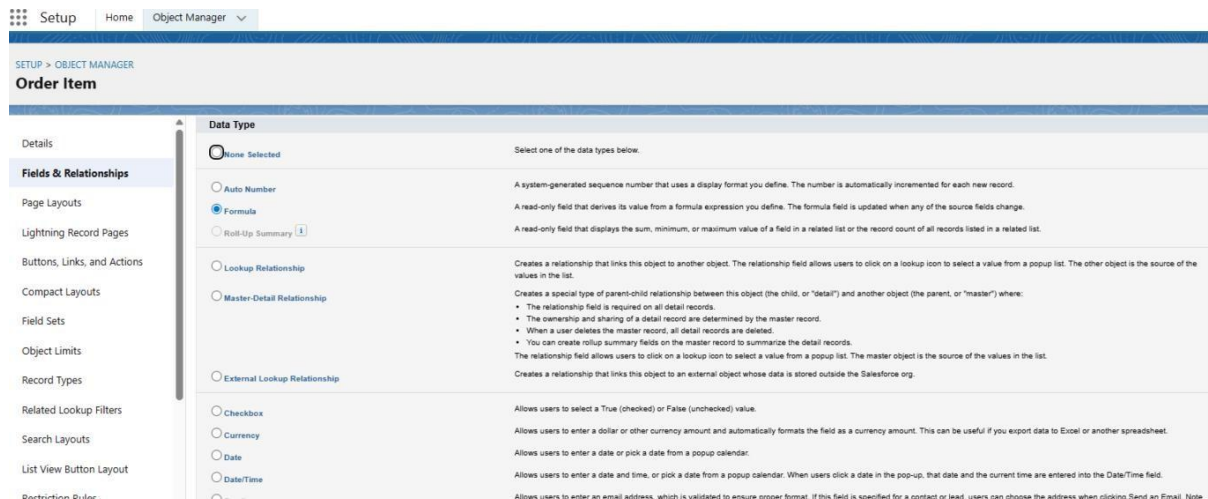
Field Label: Unit Price

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



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.

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 Salesforce 'New Custom Field' configuration page for the 'Order Item' object. The interface is in 'Step 2 of 4: Enter the details'. On the left, a navigation menu lists various setup options, with 'Fields & Relationships' currently selected. The main form area contains the following details: 'Field Label' is 'Transaction Type'; 'Values' are entered manually as 'Receipt', 'Issue', and 'Adjustment' on separate lines; the checkbox 'Restrict picklist to the values defined in the value set' is checked; and the 'Field Name' is 'Transaction_Type'. Navigation buttons 'Previous', 'Next', and 'Cancel' are visible at the top right of the form area.

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

-In 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' dialog for the 'Order Item' object. The 'Field Label' is 'Order Cost' and the 'Field Name' is 'Order_Cost'. The 'Formula Return Type' is set to 'Currency'. The 'Auto add to custom report type' checkbox is checked. The dialog is at 'Step 2 of 5'.

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' dialog for the 'Supplier' object. The 'Field Label' is 'Phone Number' and the 'Field Name' is 'Phone_Number'. The 'Description' is empty. The 'Help Text' is empty. The 'Required' checkbox is checked. The 'Auto add to custom report type' checkbox is checked. The 'Default Value' is 'Show Formula Editor'. The dialog is at 'Step 2 of 4'.

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.

The screenshot shows the 'New Custom Field' setup page for the 'Supplier' object in Salesforce. The page is titled 'Step 2: Enter the details' and shows the following fields and options:

- Field Label:** Email
- Field Name:** Email
- Description:** (empty text box)
- Help Text:** (empty text box)
- Required:** ☐ Always require a value in this field in order to save a record
- Unique:** ☐ Do not allow duplicate values
- External ID:** ☐ Set this field as the unique record identifier from an external system
- Auto add to custom report type:** ☒ Add this field to existing custom report types that contain this entity
- Default Value:** [Show Formula Editor](#)

At the bottom, there is a note: 'Use formula syntax. Enclose text and picklist value API names in double quotes ("the_text"). Include numbers without quotes. (For more information, see the formula syntax guide.)'

Page Layout Customization

Editing a Page Layout in the Product Object

Steps:

-Go to Setup → click on Object Manager.

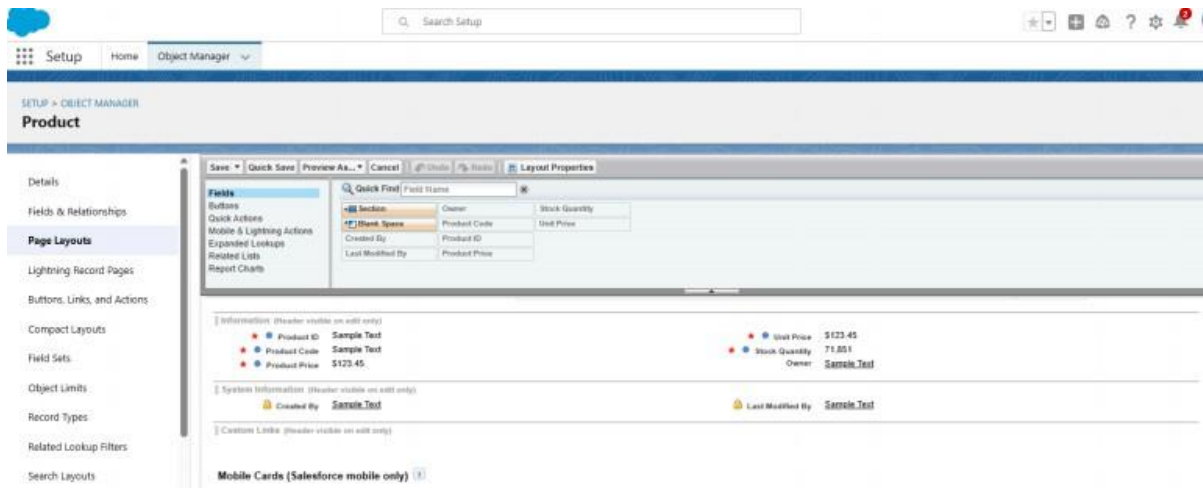
-In the Quick Find box, type Product and select the Product custom object.

-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



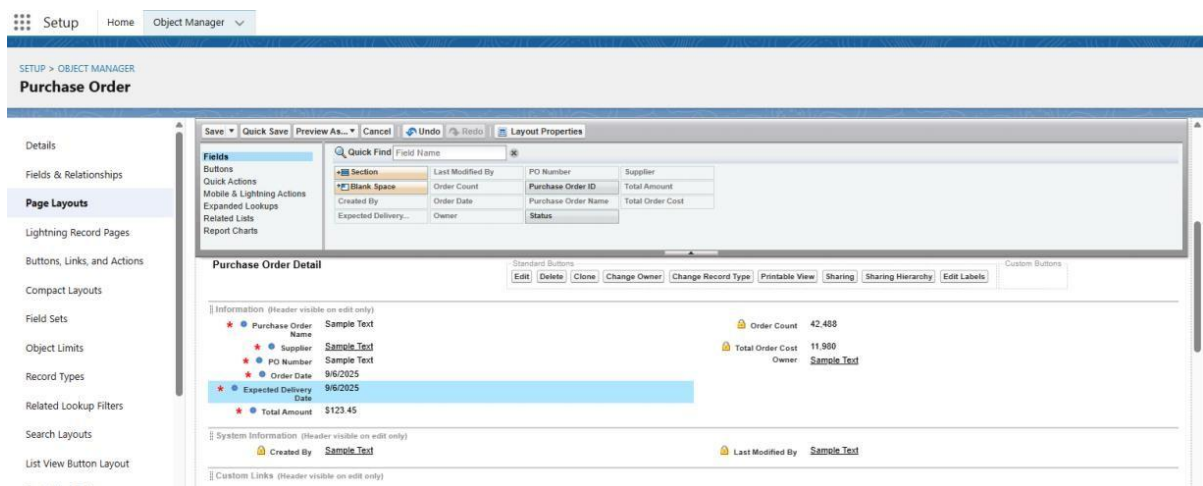
Editing a Page Layout in the Purchase Order Object

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.

The screenshot displays the Salesforce Object Manager interface for the 'Inventory Transaction' object. The left-hand navigation menu is open, showing the 'Page Layouts' section. The main content area shows the 'Inventory Transaction Detail' page layout. The layout includes a 'Fields' section with a 'Quick Find' box and a list of fields: 'Section', 'Blank Space', 'Created By', 'Inventory Transac...', 'Last Modified By', 'Order Date', 'Owner', and 'Purchase Order'. The 'Inventory Transaction Detail' section shows a table with columns for 'Information (header visible on edit only)', 'Owner', and 'Sample Text'. The table contains the following data:

Information (header visible on edit only)	Owner	Sample Text
Inventory Transaction Name		Sample Text
Purchase Order		Sample Text
Transaction Type		Sample Text
Total Order Cost		\$123.45
Order Date		9/6/2025
Supplier ID		Sample Text

The 'System Information' section shows the 'Created By' field with the value 'Sample Text' and the 'Last Modified By' field with the value 'Sample Text'. The 'Custom Links' section is also visible at the bottom.

Editing a Page Layout in the Supplier Object

Steps:

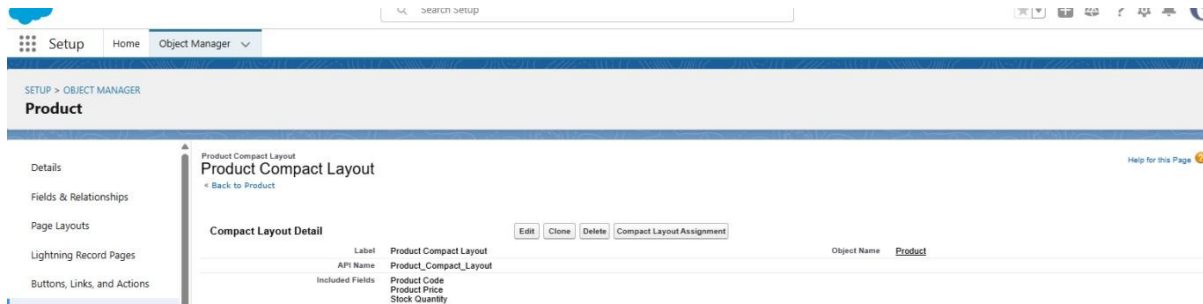
- Go to Setup → click on Object Manager.
- In the Quick Find box, type Supplier and select the Supplier custom object.
- From the left-hand menu, click Page Layouts.
- Select the layout named Supplier Layout.
- Drag and arrange the fields on the layout as required to optimize data entry and display.
- Click Save to finalize the layout changes.

Compact Layouts

Creating a Compact Layout for the Product Object

Steps:

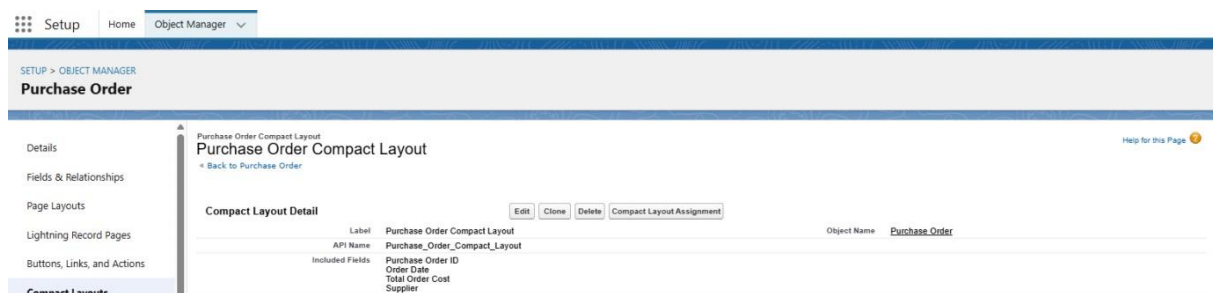
- Go to Setup → click on Object Manager.
- In the Quick Find box, type Product and select the Product custom object.
- From the sidebar, click Compact Layouts.
- Click New.
- Enter the following details:
Label: Product Compact Layout
- Select the fields to display in the compact layout:
Product Name
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. 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**.



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.



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.

The screenshot shows the Salesforce Setup interface. On the left, the navigation menu is visible with 'Profiles' selected under the 'Users' section. The main content area is titled 'Clone Profile' and contains a form to create a new profile by cloning an existing one. The form includes a search bar for the existing profile, a table of available profiles, and a 'Profile Name' field where 'Purchase Manager' has been entered. The 'Existing Profile' column shows 'Standard User' and 'Salesforce' as options. The 'User License' column shows 'Salesforce'. The 'Profile Name' column shows 'Purchase Manager'. At the bottom of the form are 'Save' and 'Cancel' buttons. A red error message 'You must select an existing profile to clone from.' is displayed at the top of the form area.

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

The screenshot shows the Salesforce Setup interface for creating a new role. The left navigation menu has 'Roles' selected under the 'Users' section. The main content area is titled 'New Role' and contains a 'Role Edit' form. The form includes fields for 'Label' (Purchasing Manager), 'Role Name' (Purchasing_Manager), and 'This role reports to' (SVP, Sales & Marketing). The 'Role Name as displayed on reports' field is empty. At the bottom of the form are 'Save', 'Save & New', and 'Cancel' buttons. The 'Role Edit' form is also titled 'New 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.

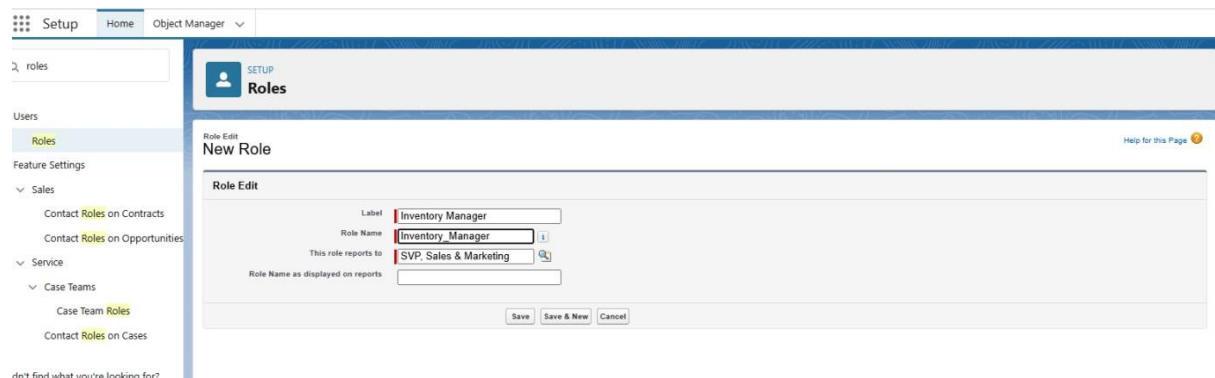
Under the SVP, Sales & Marketing role, click Add Role.

Enter the following details:

Label: Inventory Manager

The Role Name will auto-populate.

Click Save to create the role.



The screenshot shows the Salesforce Setup interface. The left sidebar contains a search bar with 'roles' entered, and a list of navigation items including 'Users', 'Roles', 'Feature Settings', 'Sales', 'Service', and 'Case Teams'. The main content area is titled 'Roles' and shows a 'New Role' form. The form has the following fields: 'Label' with the value 'Inventory Manager', 'Role Name' with the value 'Inventory_Manager', 'This role reports to' with the value 'SVP, Sales & Marketing', and 'Role Name as displayed on reports' which is empty. At the bottom of the form are three buttons: 'Save', 'Save & New', and 'Cancel'.

Permission Sets Activity 1:

Creating a Permission Set Steps:

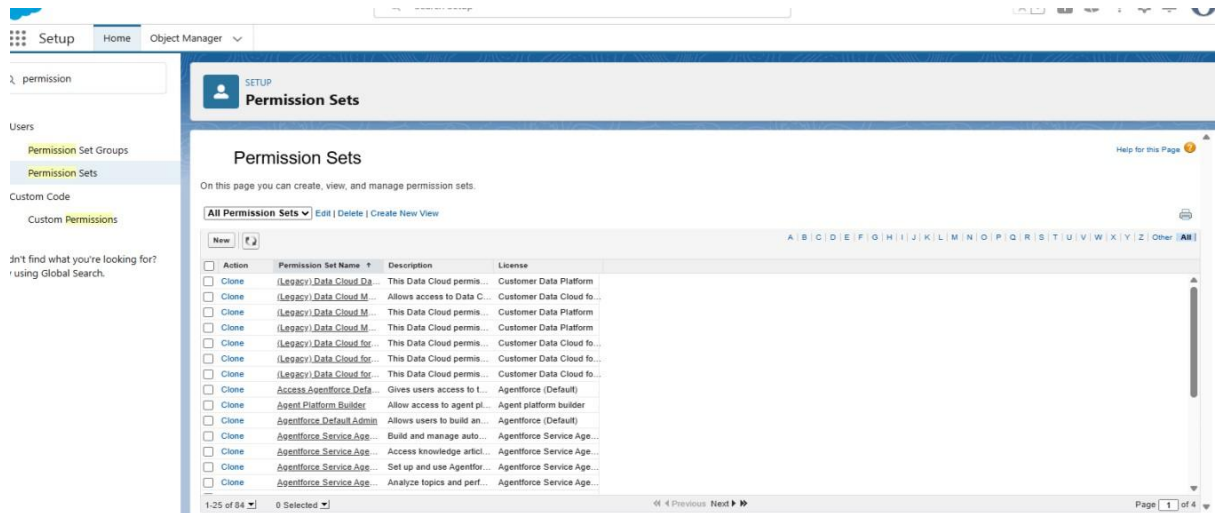
Go to Setup → type Permission in the Quick Find box → select Permission Sets.

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. 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:

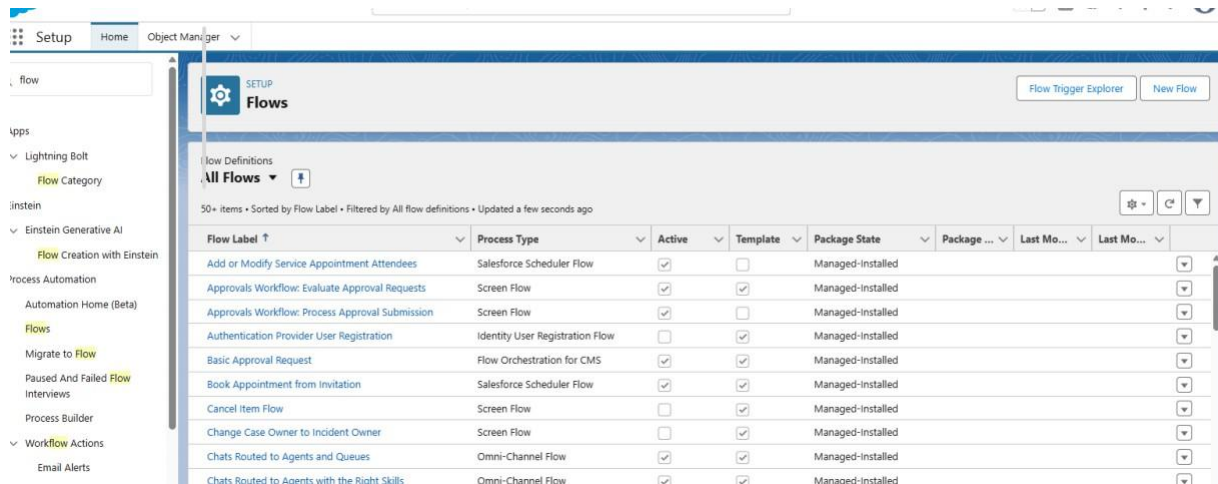
Field: Actual_Delivery_Date__c

Value: {!ActualDeliveryDate}

Click Done.

Save and Activate Flow

28. Save the flow as Actual Delivery Date Updating.
29. Activate the flow.



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

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);
    }

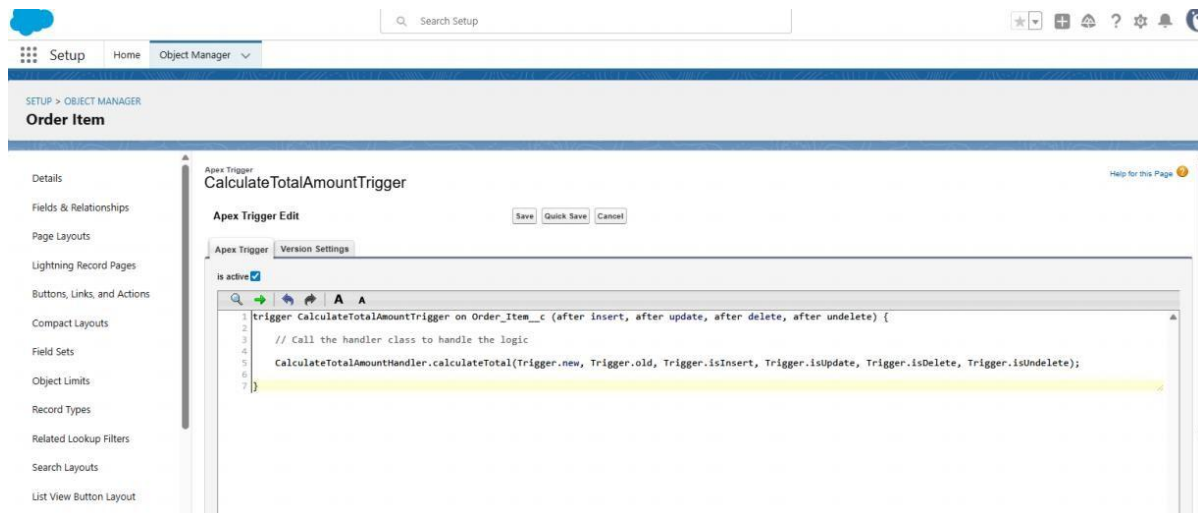
    // Update Purchase Orders if there are any changes
    if (!purchaseToUpdate.isEmpty()) {
        update purchaseToUpdate;
    }
}
}
}
}

```

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.



Reports

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.

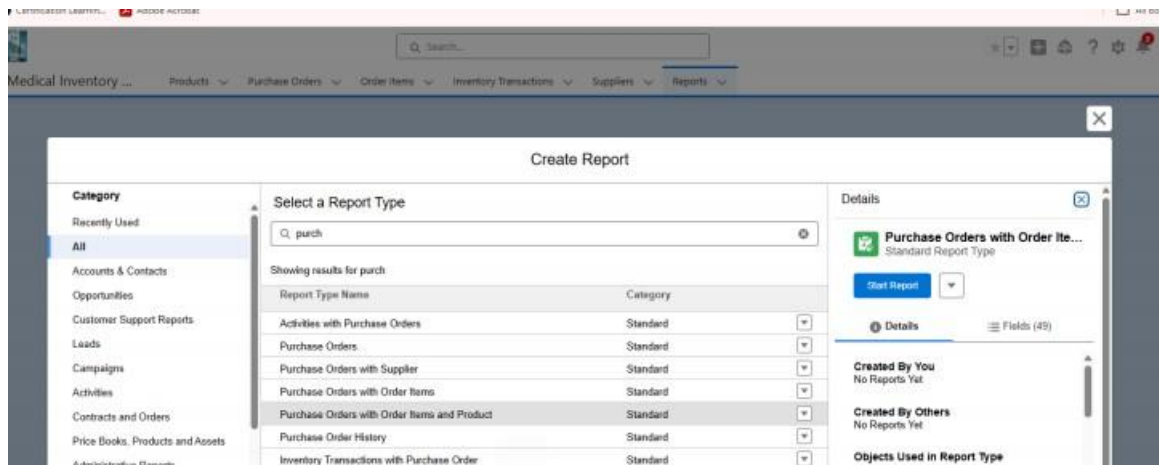
Create a Complete Purchase Details

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.

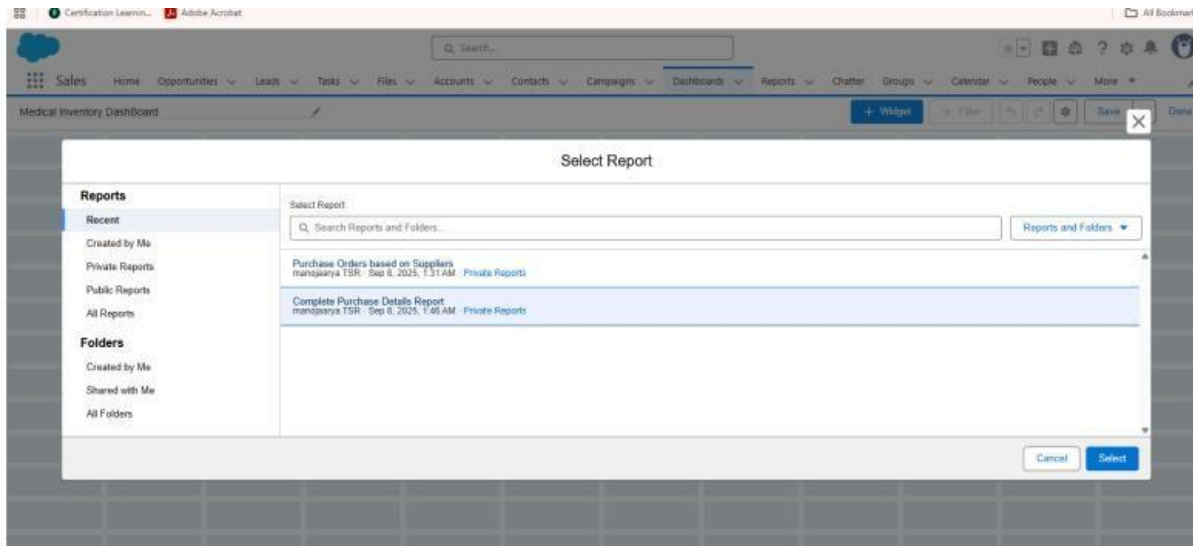
Enter the Name: Medical Inventory Dashboard → Click Create.

Click +Widget to add a component.

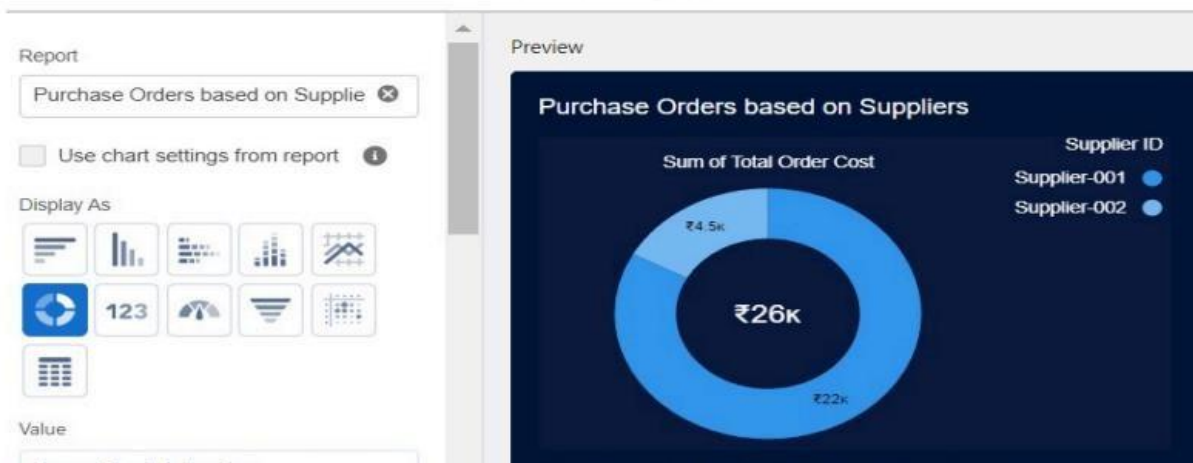
Select the Purchase Orders based on Suppliers report.

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

Click Add → then Save.



Add Widget



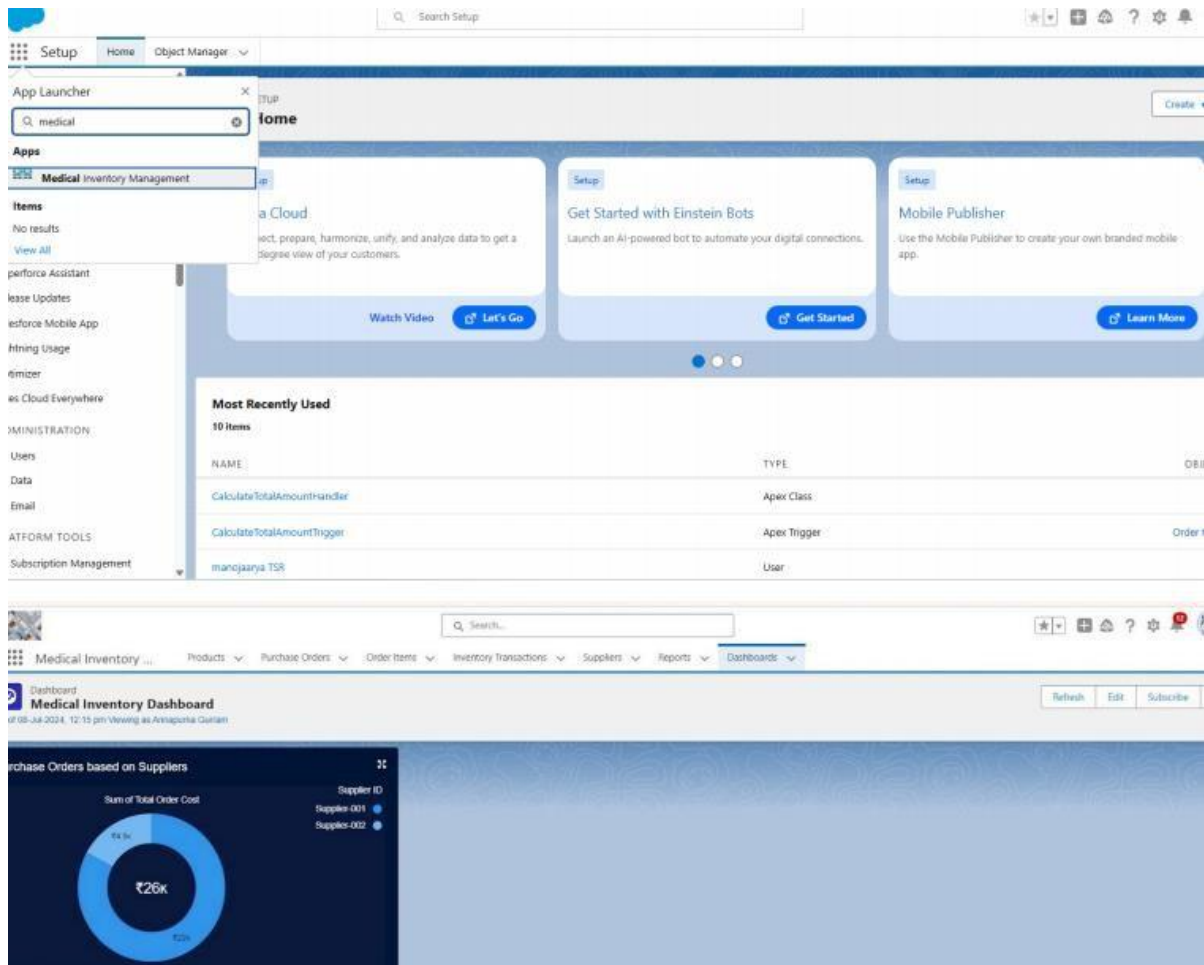
View Dashboard

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