PROJECT TITLE:

Medical Inventory Management

College Name:

Sri Vasavi College Self Finance Wing, Erode

College Code:

bru17

Team ID: NM2025TMID23361

TEAM MEMBERS:

Team Leader Name: SIVACHANDRAN S

Email: shivask0332@gmail.com

Team Member: HARIHARAN M

Email: harithiva1234@gmail.com

Team Member: SHRIL S

Email: johnpalshril408@gmail.com

Team Member: MADHUMITHA R

Email: madhued06@gmail.com

Medical Inventory Management Project Documentation

---

1. Project Overview

The **Medical Inventory Management System** is a Salesforce application designed to streamline the process of managing medical inventory.
It helps track suppliers, products, purchase orders, order items, and inventory transactions in a single platform.

This system improves:

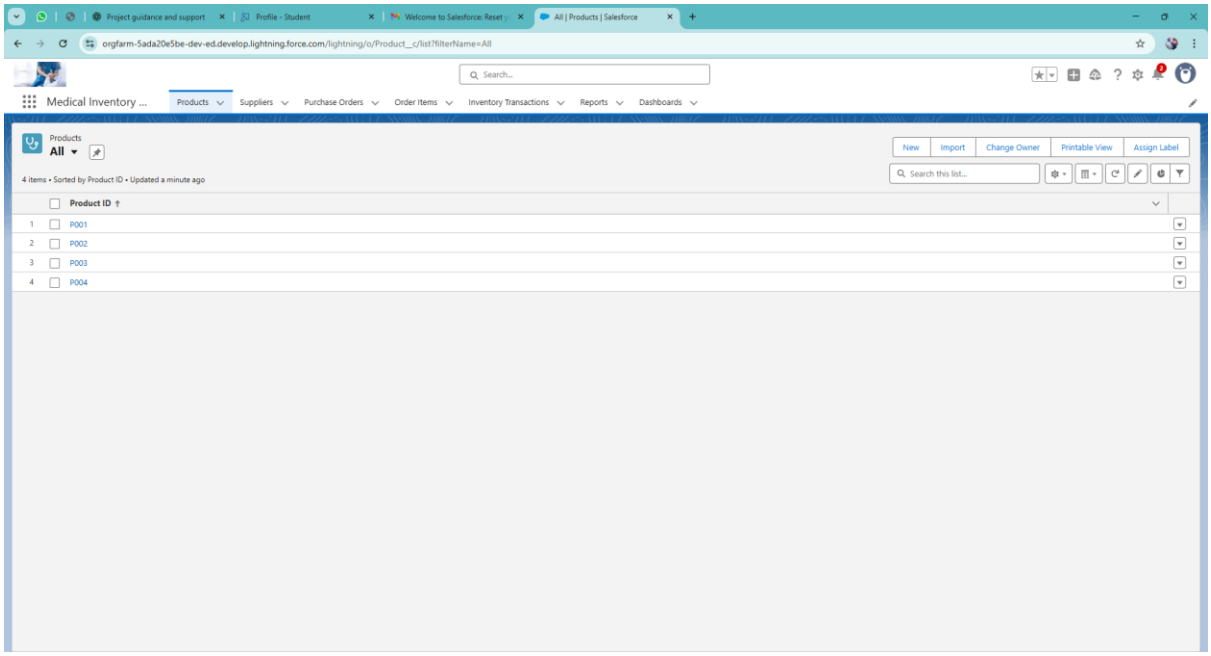
- Supplier tracking
- Stock level monitoring
- Order management
- Expiry date tracking
- Report generation and analytics

2. Objectives

The main objectives of this project are:

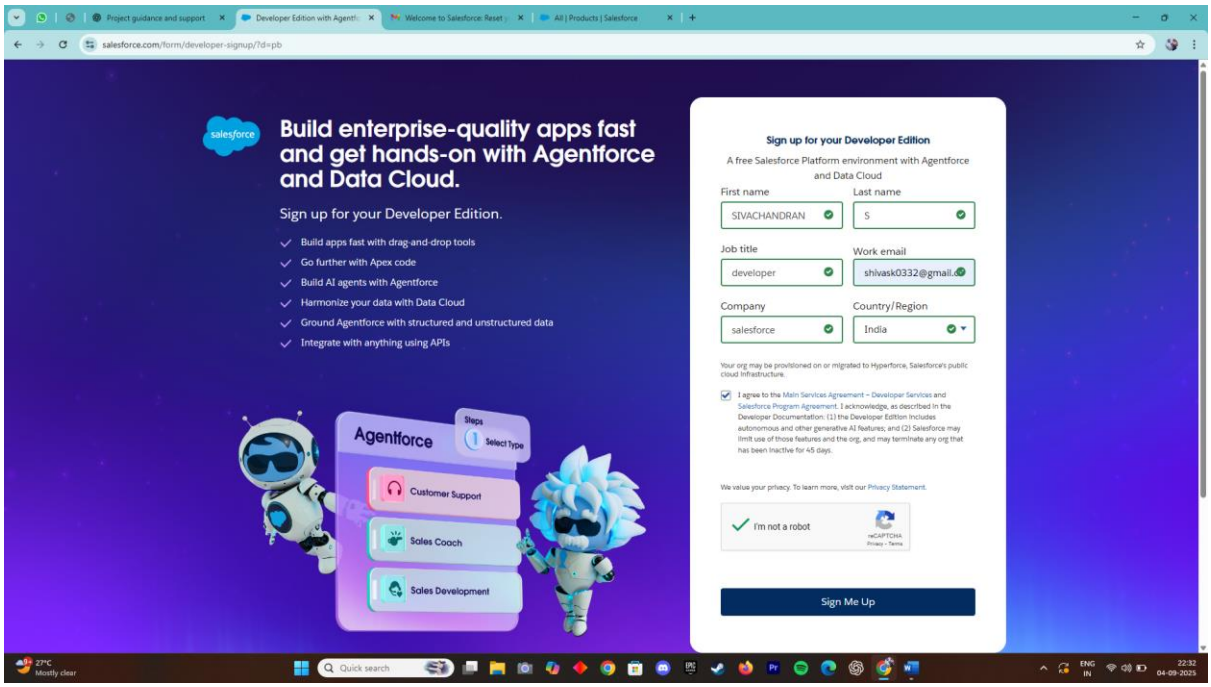
1. To efficiently manage and track **medical products** and their **stock levels**.
2. To streamline the **purchase order process** by linking suppliers and products.
3. To ensure **real-time monitoring** of stock and expiry dates.
4. To generate **insightful reports and dashboards** for decision-making.
5. To automate repetitive tasks using Salesforce tools like **flows and validation rules**.

Medical Inventory Management App.



3.Created a developer org in salesforce.

1. Go to https://developer.salesforce.com/signup
2. On the sign up form, enter the following details :



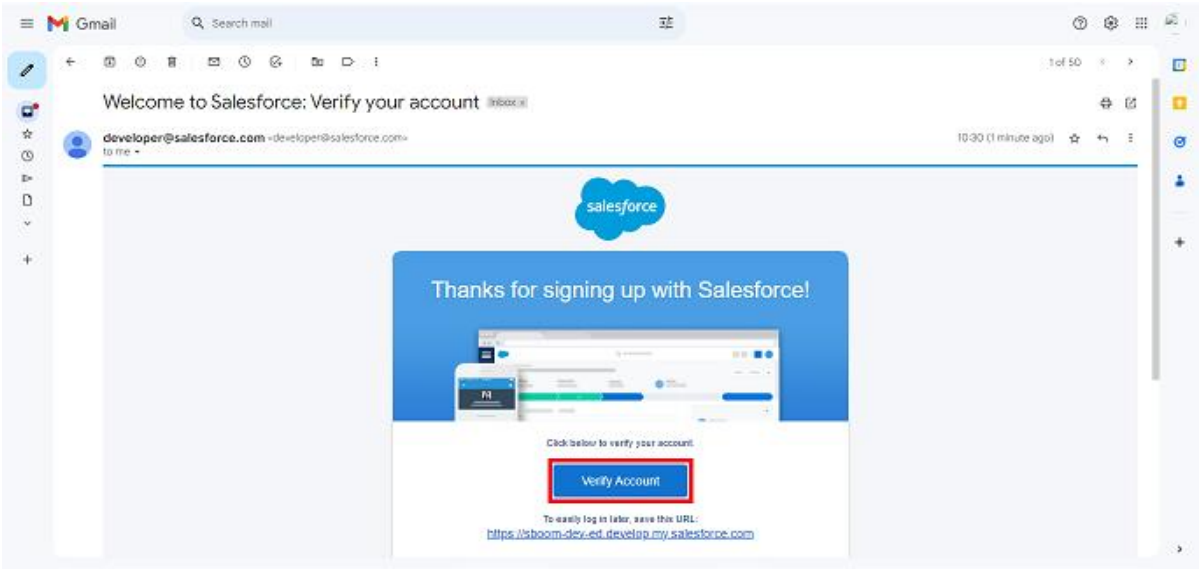
1. First name & Last name
2. Email

3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code

Click on sign me up after filling these.

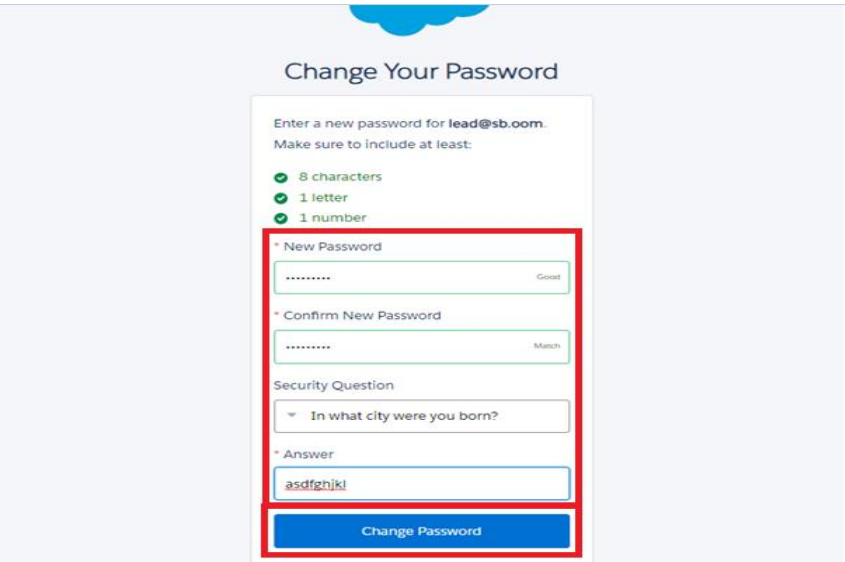
4.Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.

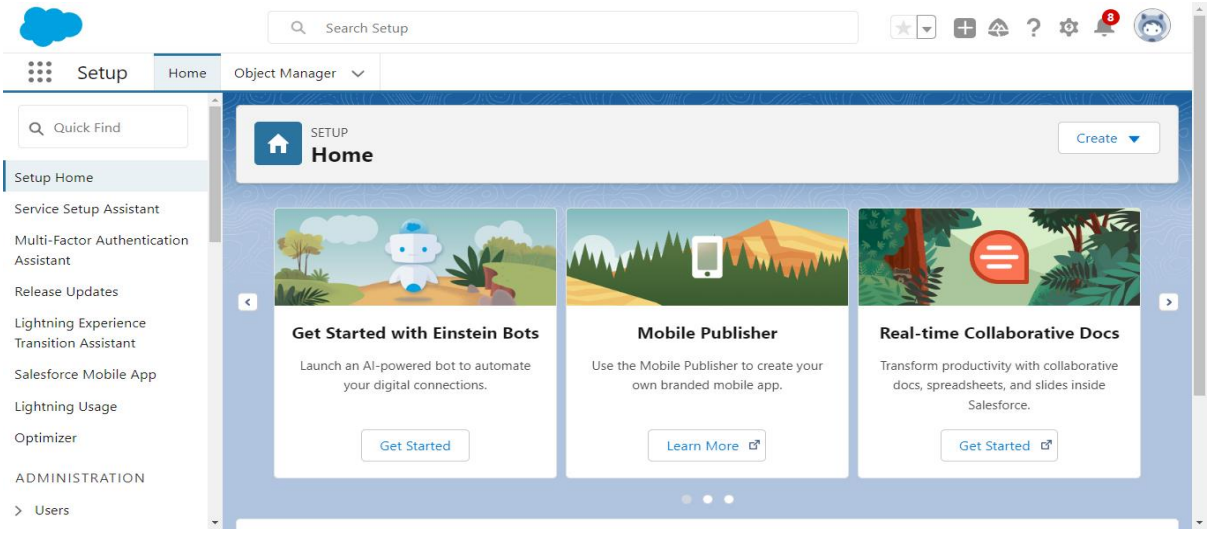


2. Click on Verify Account

3. Give a password and answer a security question and click on change password.



4. Then you will redirect to your salesforce setup page.



---

5. Objects Created

The following **custom objects** were created to manage the system:

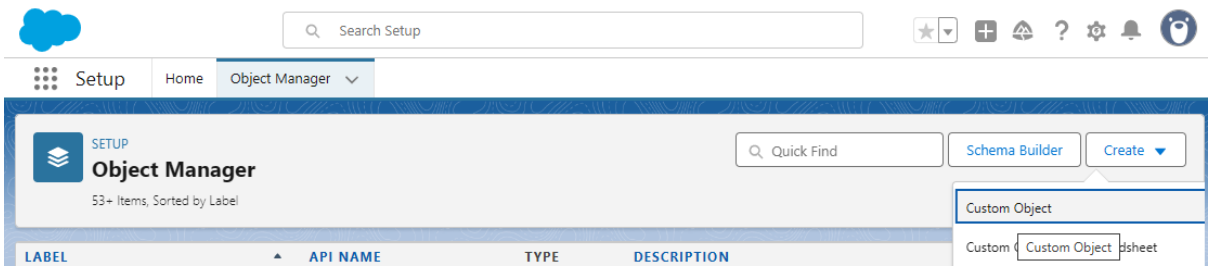
Object Name	Purpose
Supplier	To store supplier details like name, contact person, phone, and address.

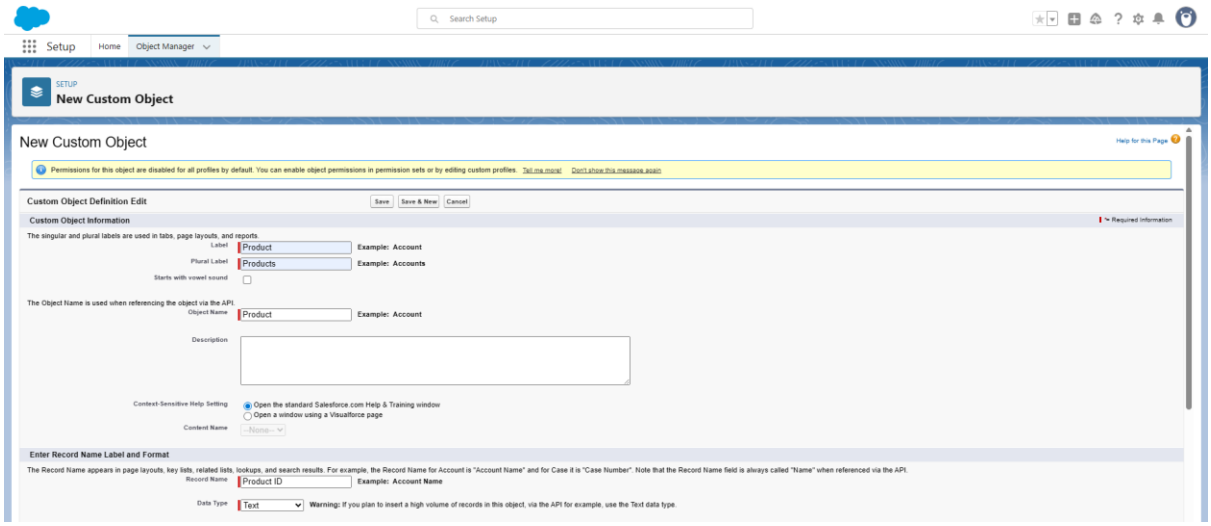
Object Name	Purpose
Product	To store product information like unit price, current stock level, and minimum stock level.
Purchase Order	To record order details from suppliers.
Order Item	To link products with each purchase order and track quantities and costs.
Inventory Transaction	To track stock movement (Inbound/Outbound).

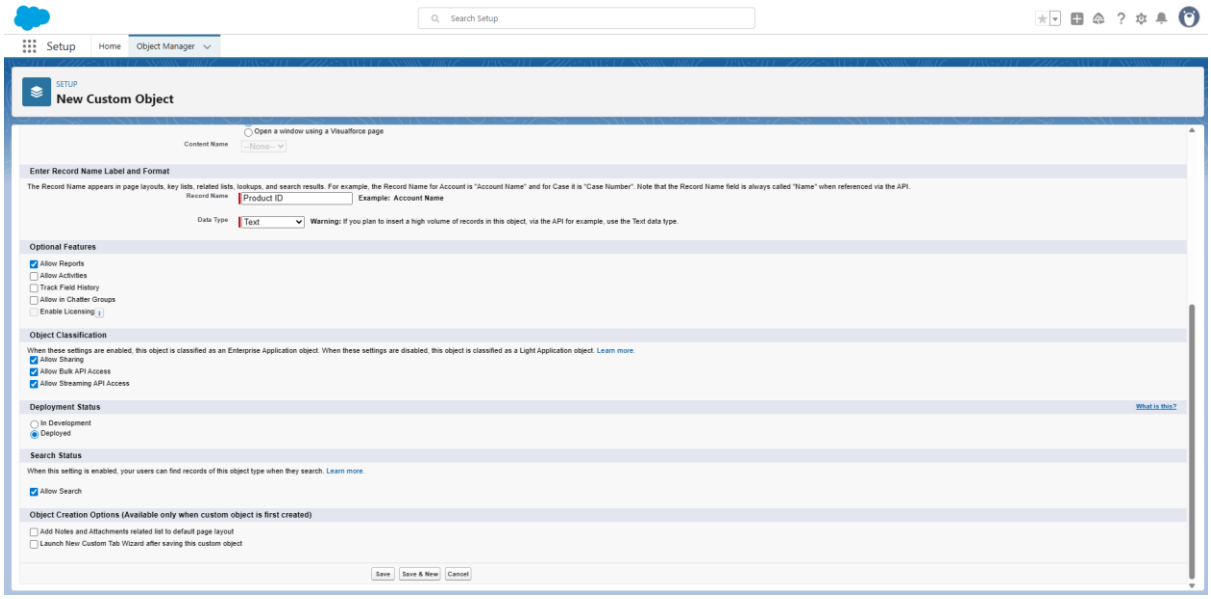
In Salesforce, objects are database tables that allow you to store data specific to your organization.

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object.
4. Enter the label name as Product
5. Enter Plural label name as Products
6. Enter Record Name as Product ID
7. Select Data Type as Text.
8. Select Allow reports.
9. Select Allow search.
10. Click on Save and New







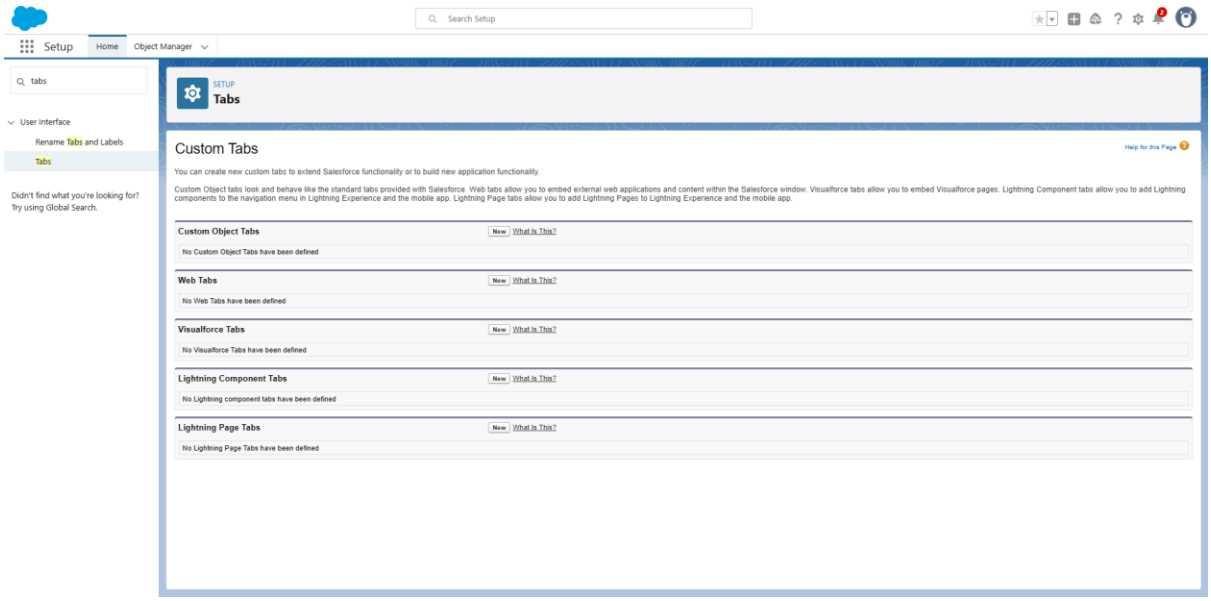
In the same way Created Purchase Order, Order Item, Inventory Transaction and Supplier objects.

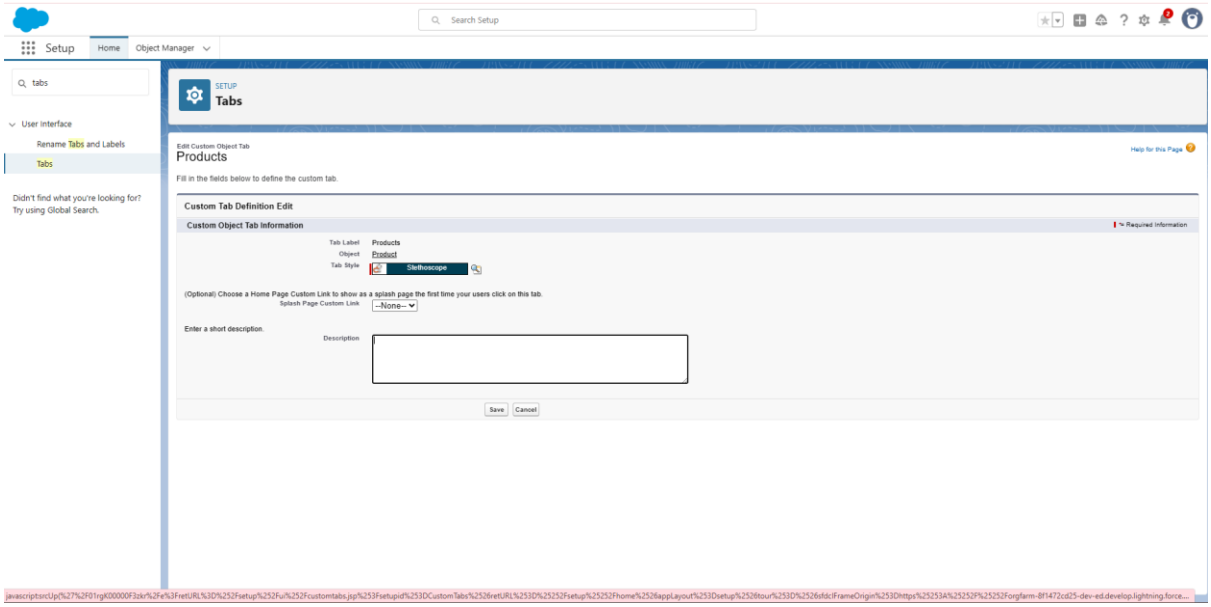
6.Tabs Created

In Salesforce, tabs are used to make the data stored in objects accessible to users through the user interface. Tabs are a fundamental part of the Salesforce interface, providing a way to navigate to different objects and records.

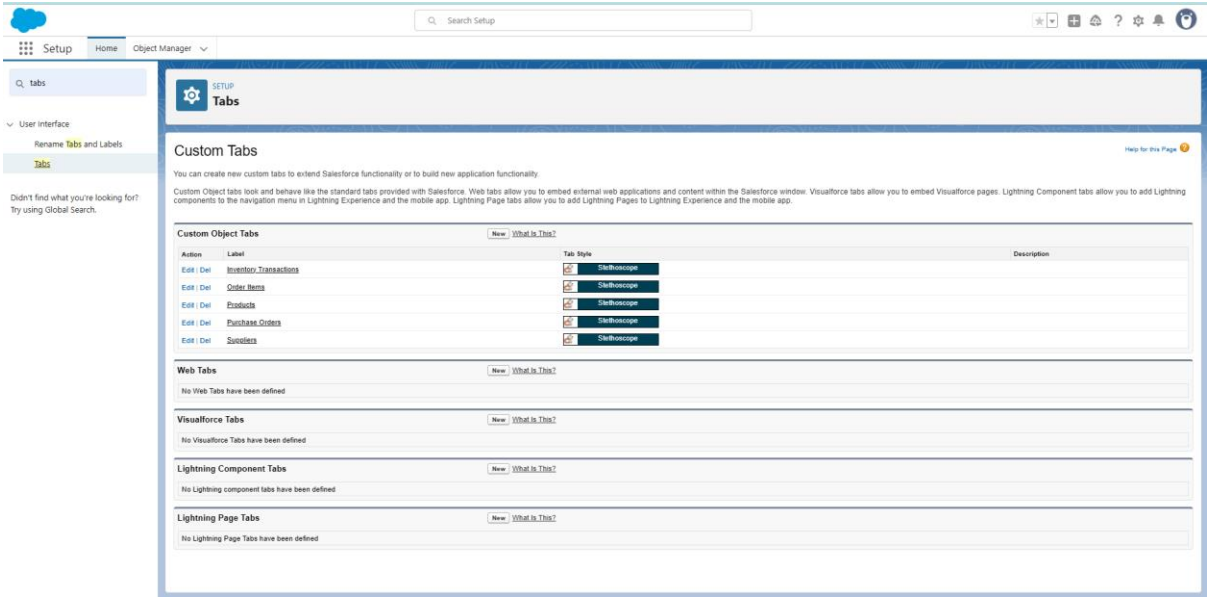
1. Go to the setup page >> type Tabs in Quick Find bar
2. Click on tabs

3. Click on New (under custom object tab).
4. Select Object(Product) >> Select the tab style
5. Click on Next >> (Add to profiles page) keep it as default >> Click on Next (Add to Custom App) uncheck the include tab .
6. Make sure that the Append tab to user's existing personal customizations is checked.
7. Click save





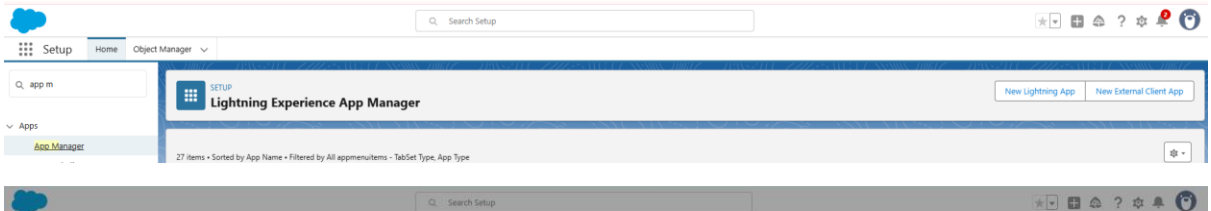
1. Now created the Tabs for the remaining Objects, they are “Purchase Order, Order Item, Inventory Transaction, Supplier”.
2. Followed the same steps as mentioned in above.

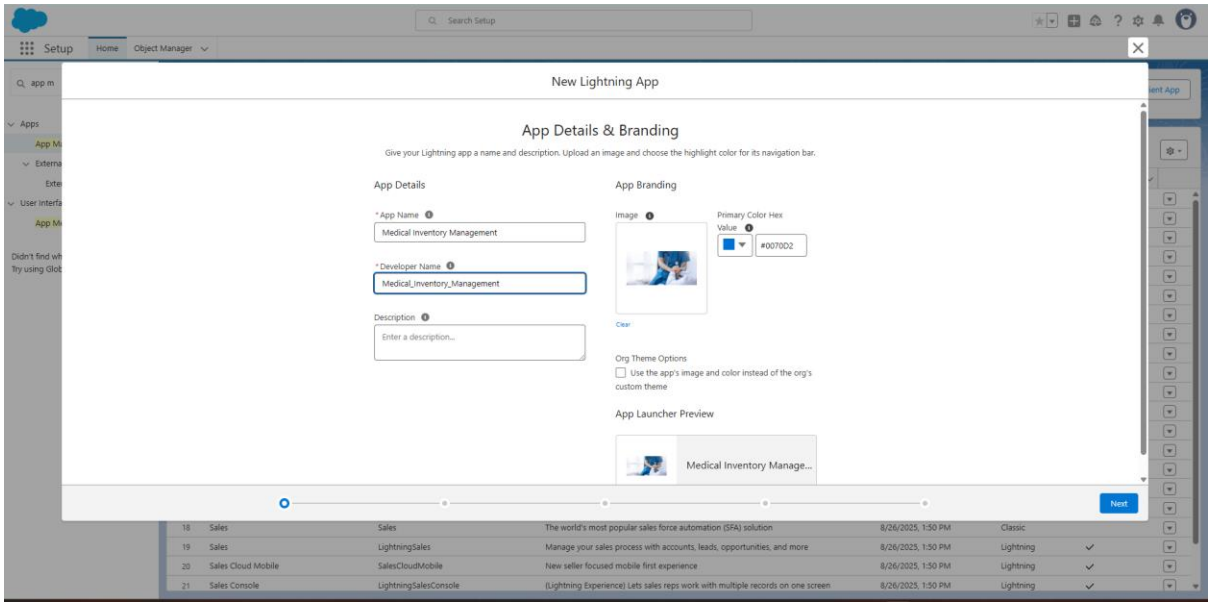


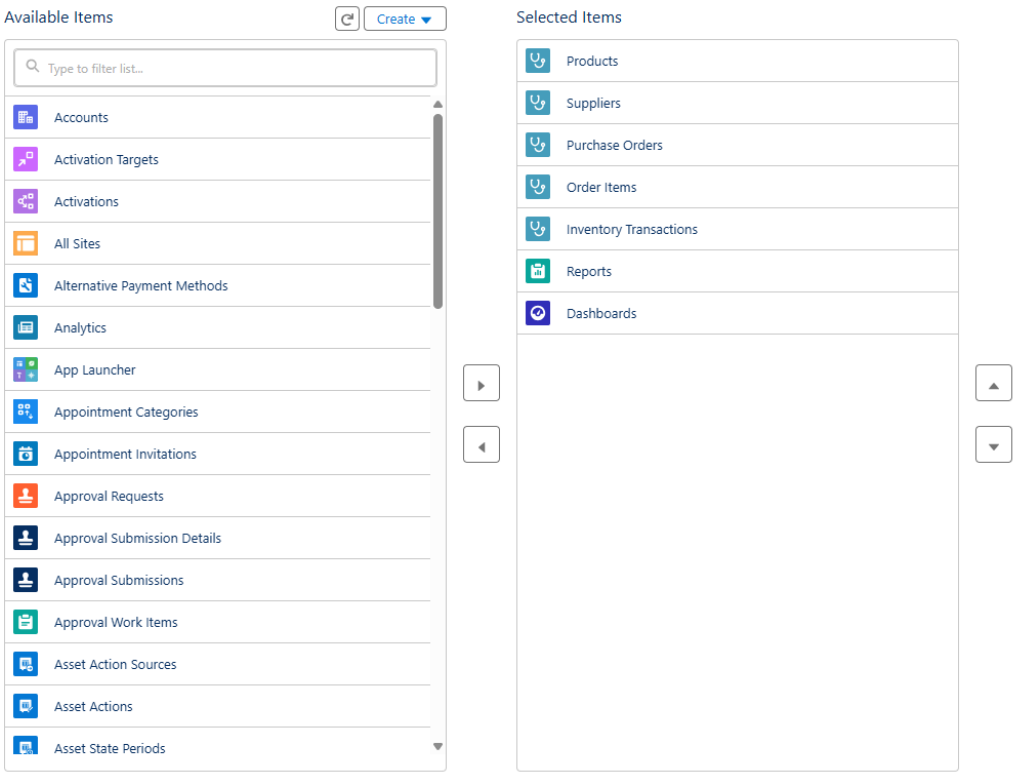
7.Created a Lightning App for Medical Inventory Management

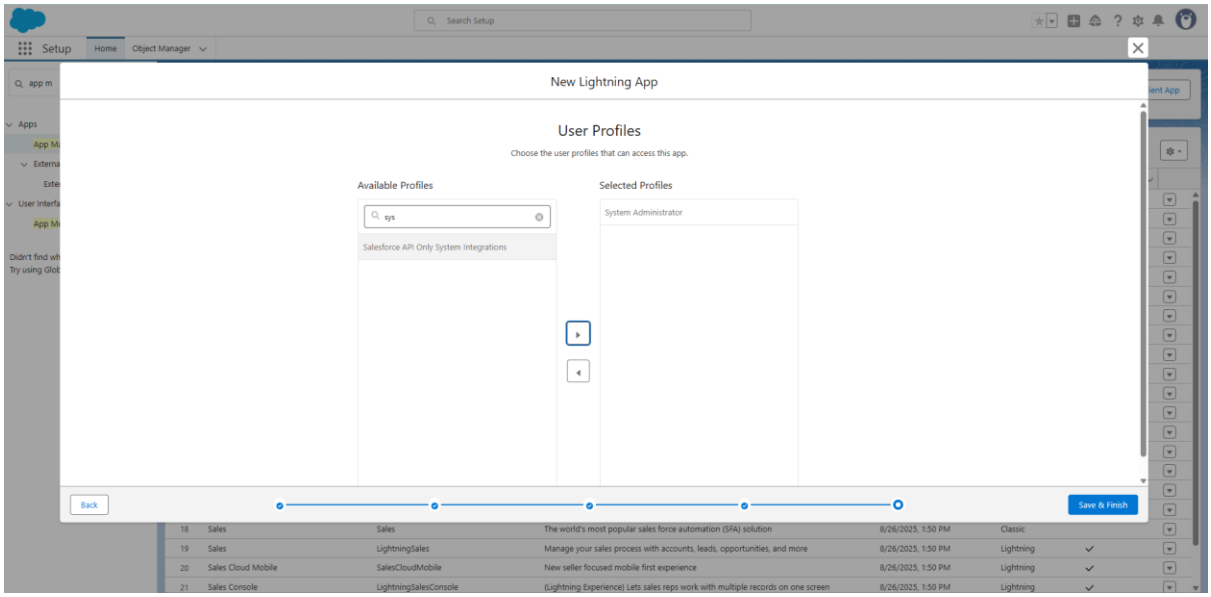
1. From Setup, enter App Manager in the Quick Find and select App Manager.
2. Click New Lightning App.
3. Enter Medical Inventory Management as the App Name >> Click on upload image and add an image related to Medical Inventory then click next
4. Under App Options, leave the default selections and click next.
5. Under Utility Items, leave as is and click Next.
6. From Available Items, select Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, Reports, and Dashboards and move them to Selected Item and Click Next.

7. From Available Profiles, select System Administrator and move it to Selected Profiles.
8. Click Save & Finish.



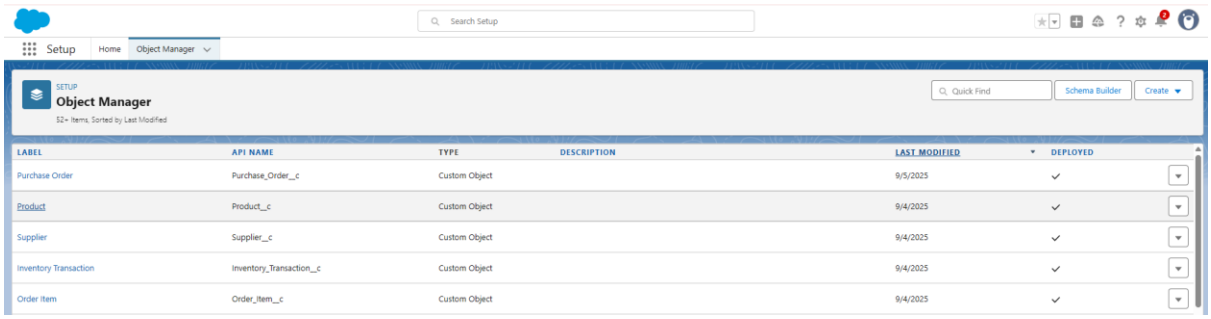


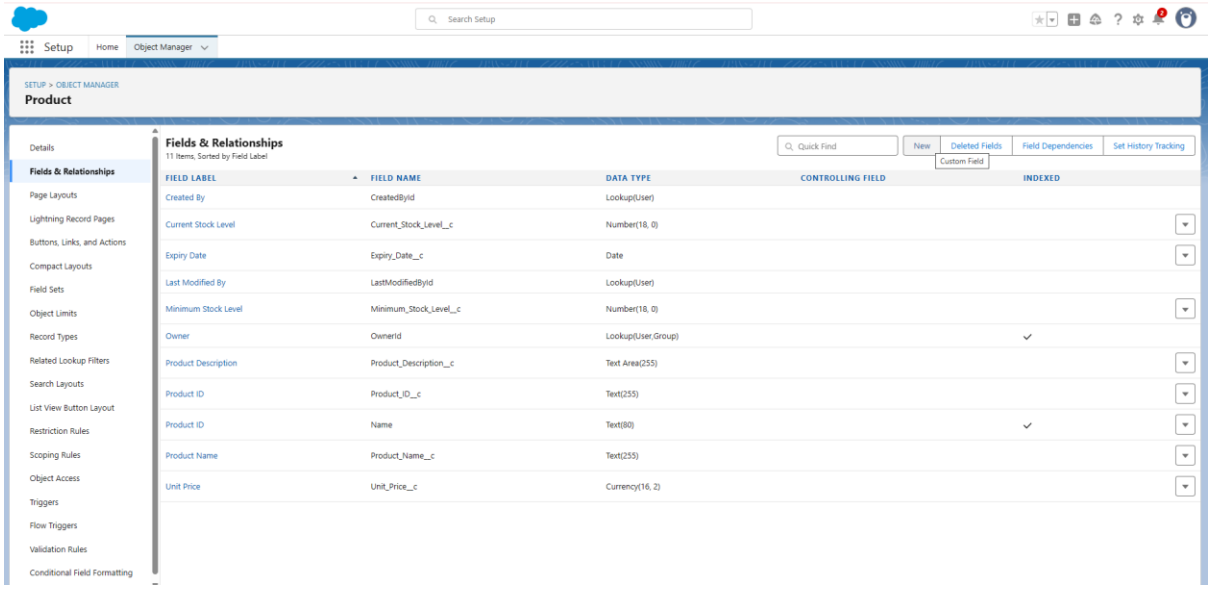


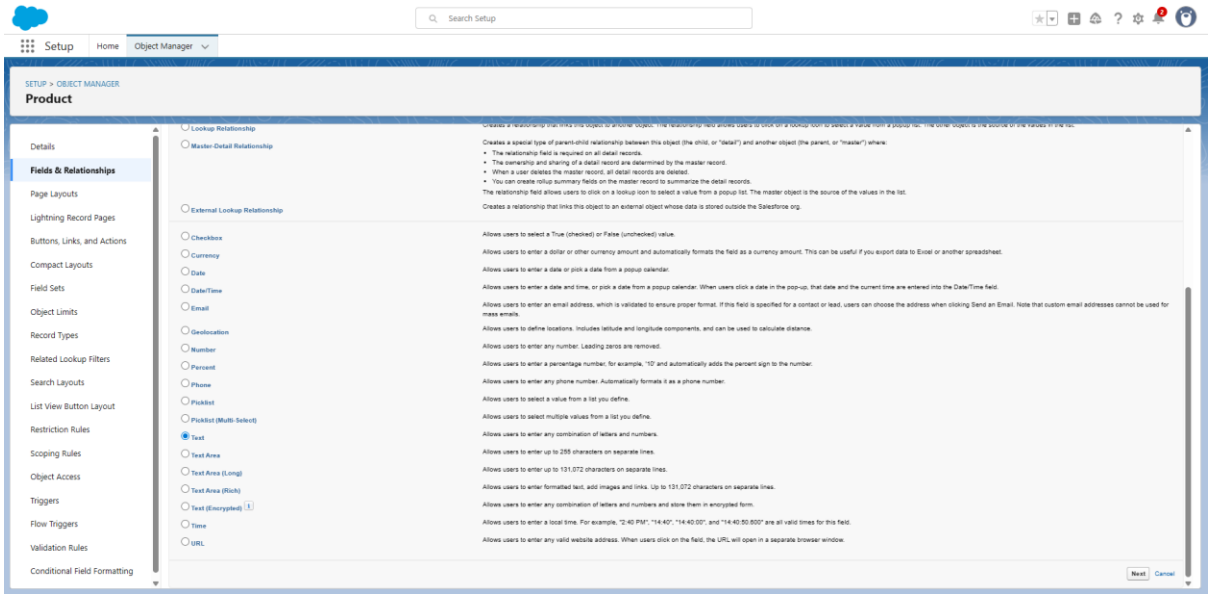


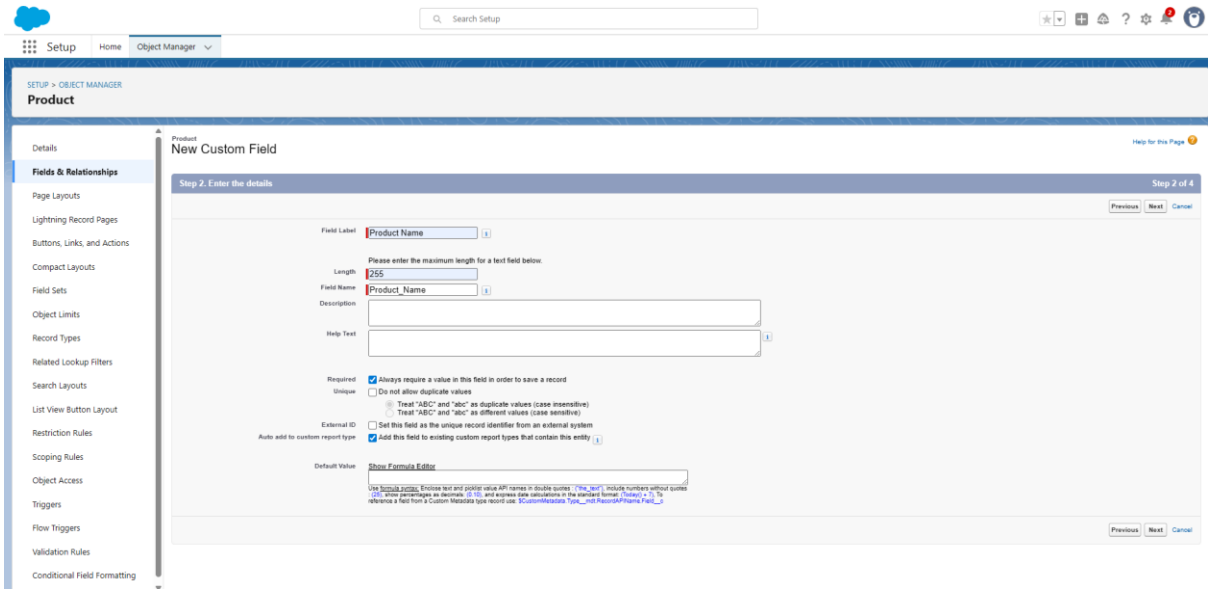
8.Created a Text Field in Product Object

1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select Product custom object.
4. Select Fields & Relationships from the left navigation
5. Click on New
6. Select Text field, click Next
7. Enter Field Label as “Product Name” and Length 255.
8. Select Required Field.
9. Click Next, Next, then Save & New.




Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Product
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
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Fields & Relationships
11 Items, Sorted by Field Label
Quick Find
New
Deleted Fields
Field Dependencies
Set History Tracking
Custom Field
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED
Created By
CreatedById
Lookup(User)
Current Stock Level
Current_Stock_Level__c
Number(18, 0)
Expiry Date
Expiry_Date__c
Date
Last Modified By
LastModifiedById
Lookup(User)
Minimum Stock Level
Minimum_Stock_Level__c
Number(18, 0)
Owner
OwnerId
Lookup(User,Group)
Product Description
Product_Description__c
Text Area(255)
Product ID
Product_ID__c
Text(255)
Product ID
Name
Text(80)
Product Name
Product_Name__c
Text(255)
Unit Price
Unit_Price__c
Currency(16, 2)


Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Product
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
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Lookup Relationship
Master-Detail Relationship
Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:
The relationship field is required on all detail records.
The ownership and sharing of a detail record are determined by the master record.
When a user deletes the master record, all detail records are deleted.
You can create rollup summary fields on the master record to summarize the detail records.
The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.
External Lookup Relationship
Creates a relationship that links this object to an external object whose data is stored outside the Salesforce org.
Checkbox
Allows users to select a True (checked) or False (unchecked) value.
Currency
Allows users to enter a dollar or other currency amount and automatically formats the field as a currency amount. This can be useful if you export data to Excel or another spreadsheet.
Date
Allows users to enter a date or pick a date from a popup calendar.
Date/Time
Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.
Email
Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.
Geolocation
Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.
Number
Allows users to enter any number. Leading zeros are removed.
Percent
Allows users to enter a percentage number, for example, "10" and automatically adds the percent sign to the number.
Phone
Allows users to enter any phone number. Automatically formats it as a phone number.
Picklist
Allows users to select a value from a list you define.
Picklist (Multi-Select)
Allows users to select multiple values from a list you define.
Text
Allows users to enter any combination of letters and numbers.
Text Area
Allows users to enter up to 255 characters on separate lines.
Text Area (Long)
Allows users to enter up to 131,072 characters on separate lines.
Text Area (Rich)
Allows users to enter formatted text, add images and links. Up to 131,072 characters on separate lines.
Text (Encrypted)
Allows users to enter any combination of letters and numbers and store them in encrypted form.
Time
Allows users to enter a local time. For example, "2:40 PM", "14:40", "14:40:00", and "14:40:50.600" are all valid times for this field.
URL
Allows users to enter any valid website address. When users click on the field, the URL will open in a separate browser window.
Next
Cancel


Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Product
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
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting
Product
New Custom Field
Help for this Page
Step 2. Enter the details
Step 2 of 4
Previous
Next
Cancel
Field Label
Product Name
Please enter the maximum length for a text field below.
Length
255
Field Name
Product_Name
Description
Help Text
Required
Always require a value in this field in order to save a record
Unique
Do not allow duplicate values
Treat "ABC" and "abc" as duplicate values (case insensitive)
Treat "ABC" and "abc" as different values (case sensitive)
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
Previous
Next
Cancel

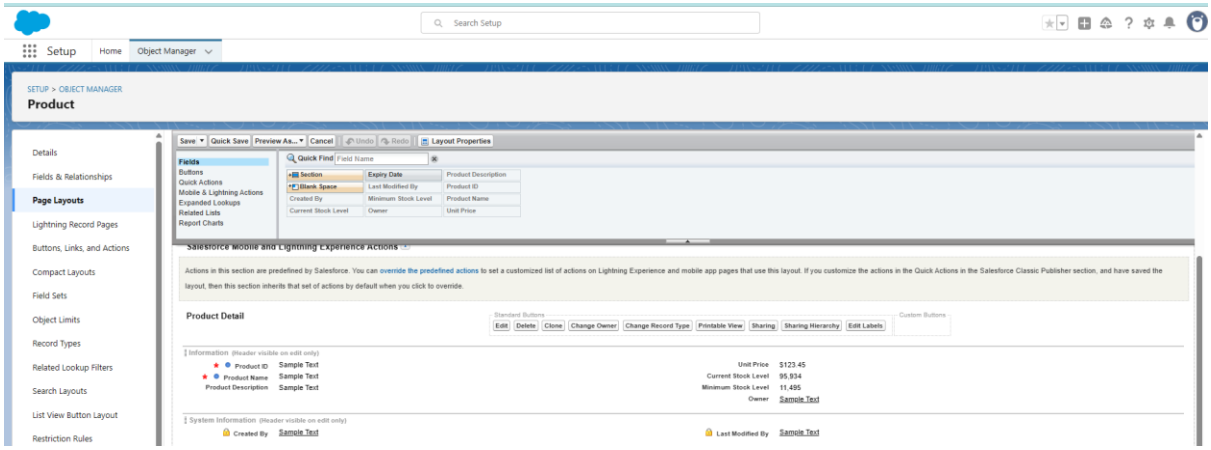
Field Added in Object:

Object	Field Name	Data Type
Product	Product ID(Standard)	Text
	Product Name	Text
	Product Description	Text Area
	Minimum Stock Level	Number(18, 0)
	Current Stock Level	Number(18, 0)
	Unit Price	Currency(16, 2)
	Expiry Date	Date
Purchase Order	Purchase Order ID(Standard)	Text
	Supplier ID	Lookup(Supplier)
	Order Date	Date
	Expected Delivery Date	Date
	Actual Delivery Date	Date
	Order Count	Roll-Up Summary (COUNT Order Item)
	Total Order Cost	Currency(16, 2)
Order Item	Order Item ID(Standard)	Text
	Product ID	Lookup(Product)
	Purchase Order ID	Master-Detail(Purchase Order)
	Quantity Ordered	Number(18, 0)

	Quantity Received	Number(18, 0)
	Unit Price	Formula(Currency)
	Amount	Formula(Currency)
Inventory Transaction	Transaction ID(Standard)	Text
	Purchase Order ID	Lookup(Purchase Order)
	Transaction Date	Date
	Transaction Type	Picklist
	Total Order Cost	Formula(Currency)
Supplier	Supplier ID(Standard)	Text
	Supplier Name	Text
	Contact Person	Text
	Phone Number	Phone
	Email	Email
	Address	TextArea

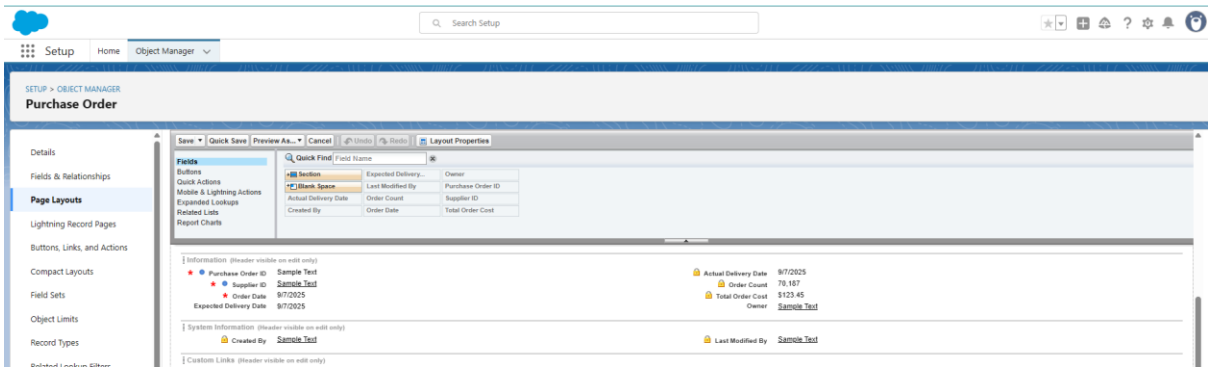
9.Edited a Page Layout in Product Object

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object >> Page Layouts .
2. Click on the Product Layout.
3. Drag and Arrange the field as shown below.

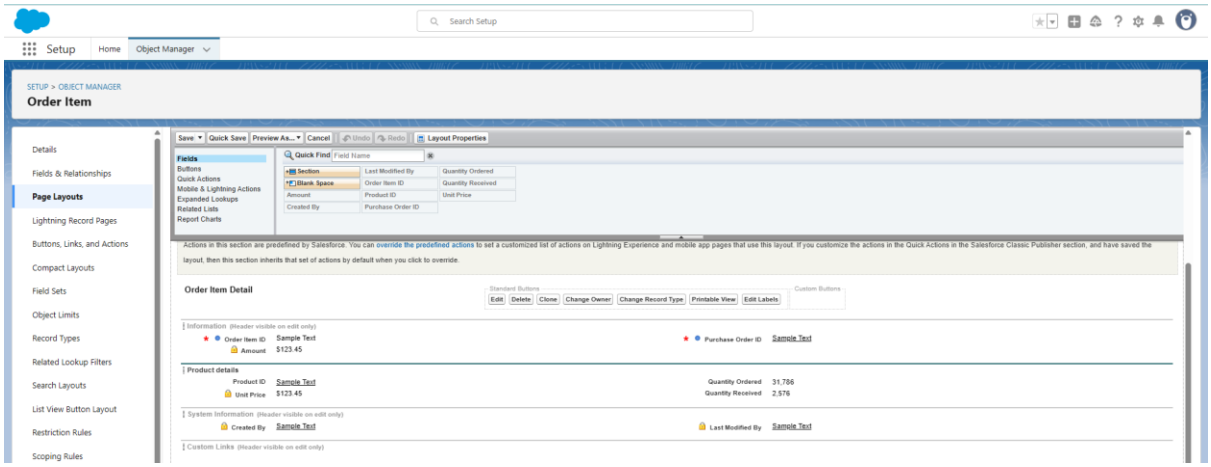


4. Click on Save.

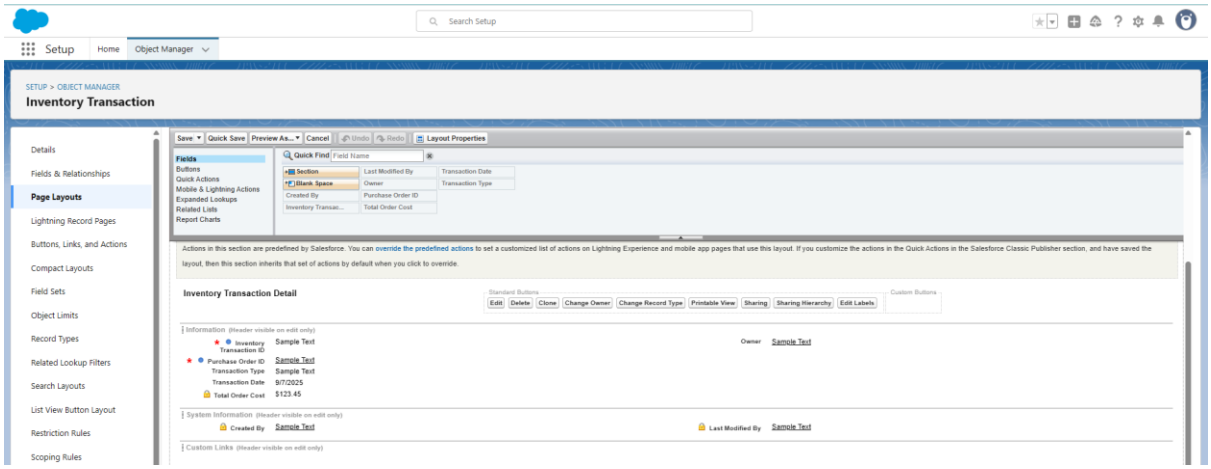
9.1Edited a Page Layout in Purchase Order Object



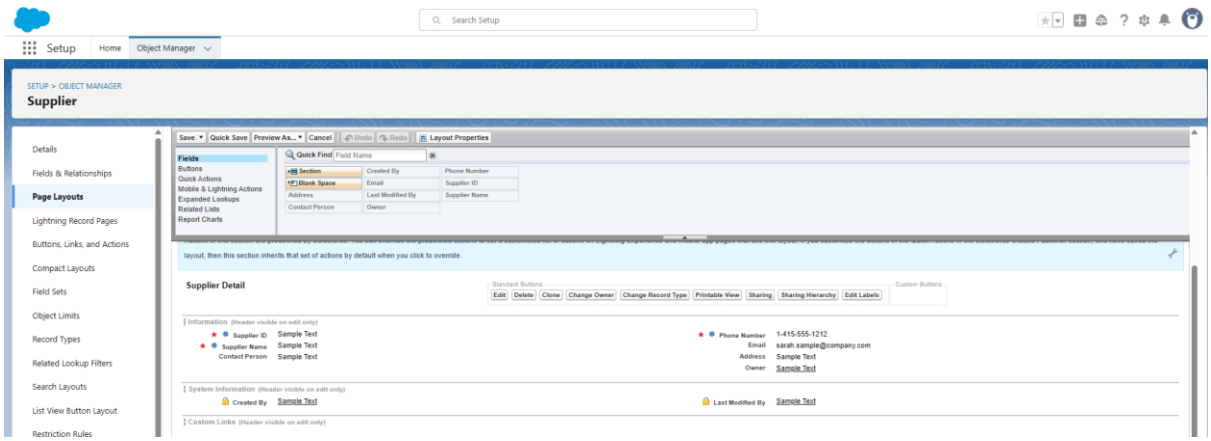
9.2Edited a Page Layout in Order Item Object



9.3Edited a Page Layout in Inventory Transaction Object

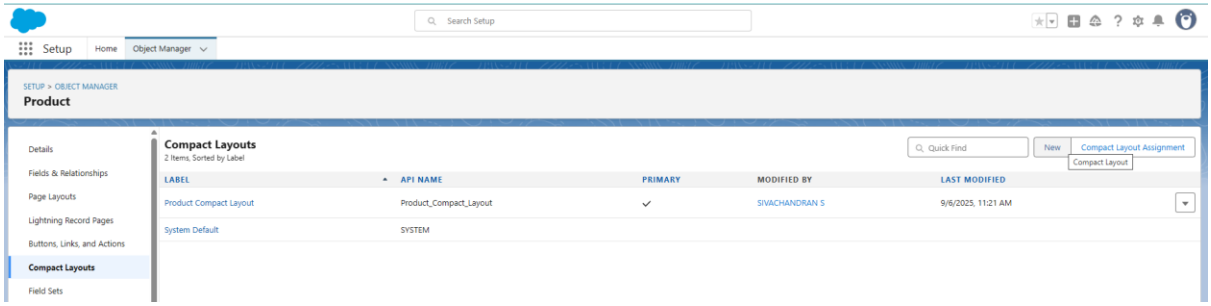


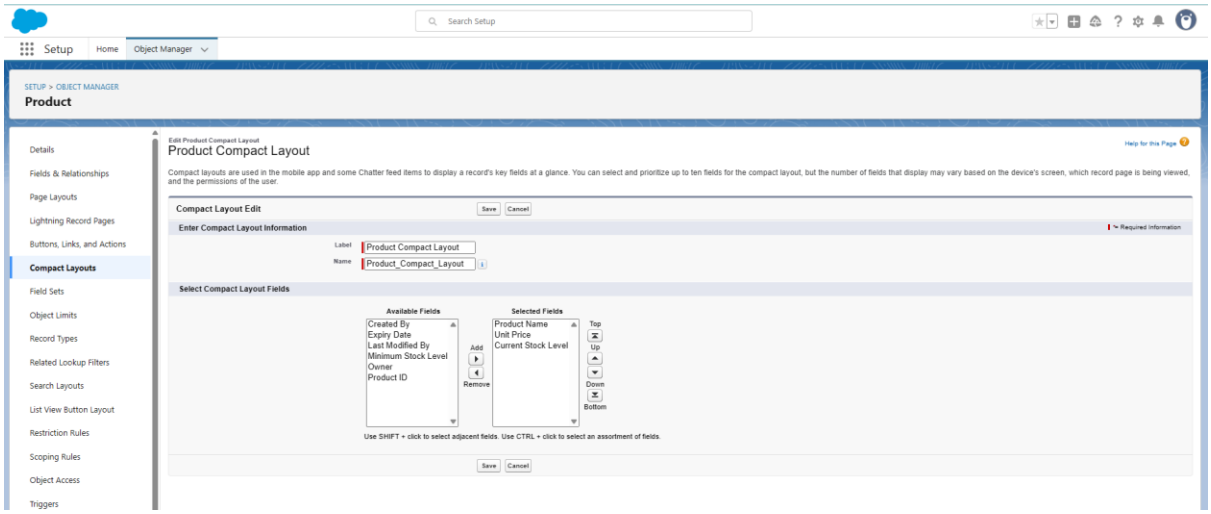
9.4Edited a Page Layout in Supplier Object

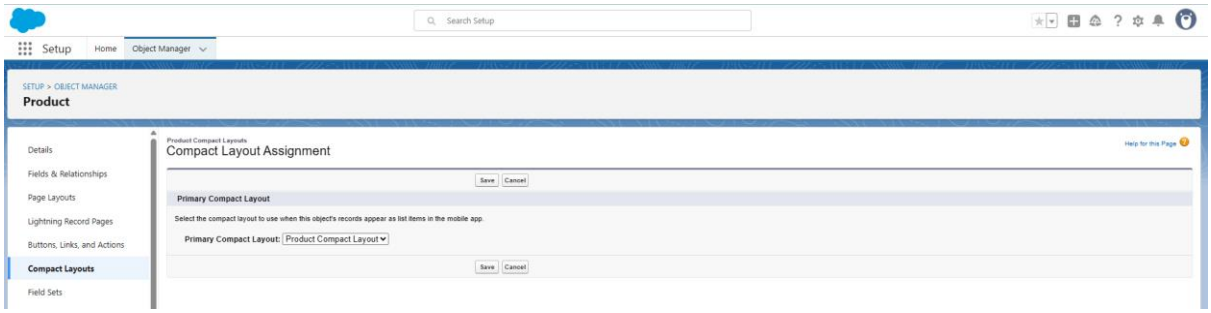


10.Created a Compact Layout to a Product Object

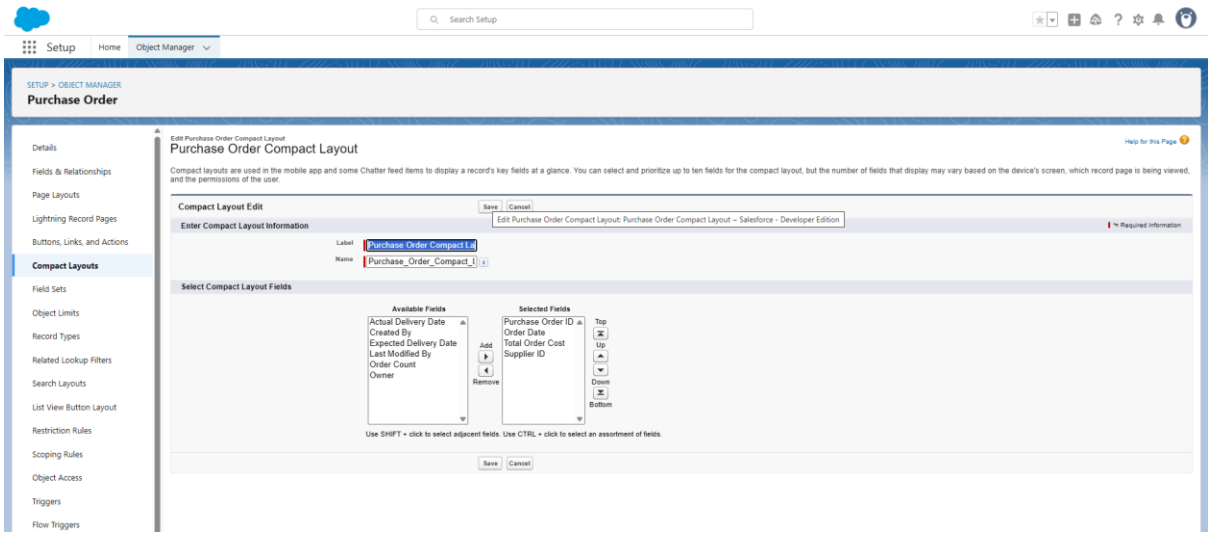
1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object
2. Click on Compact Layouts in the sidebar .
3. Click on New.
4. Enter the Label as “Product Compact Layout”.
5. Select the Compact Layout Fields : Select Product name, Unit Price, Current Stock Level.
6. Click Save.
7. Click Compact Layout Assignment.
8. Click Edit Assignment.
9. Choose "Product Compact Layout" from the dropdown.
10. Click Save.

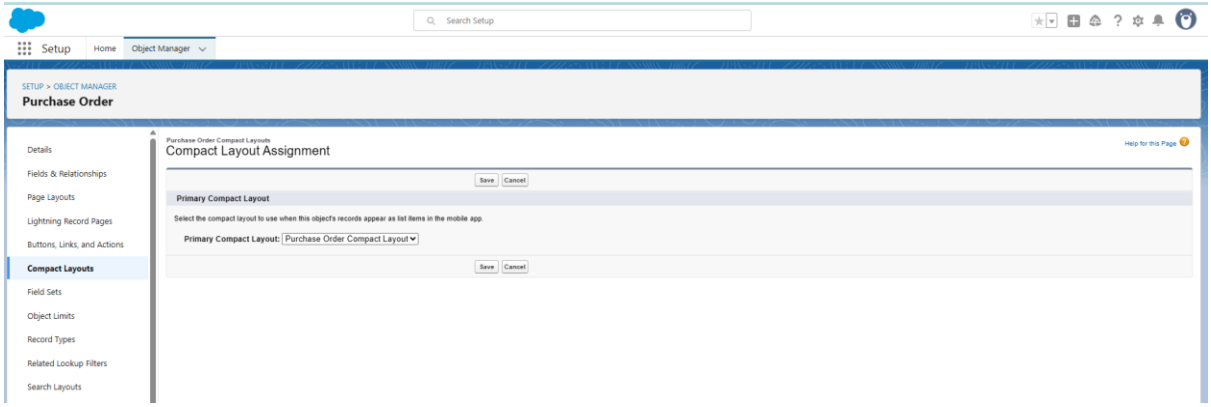

Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Product
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Compact Layouts
2 Items, Sorted by Label
Quick Find
New
Compact Layout Assignment
Compact Layout
LABEL
API NAME
PRIMARY
MODIFIED BY
LAST MODIFIED
Product Compact Layout
Product_Compact_Layout
SIVACHANDRAN S
9/6/2025, 11:21 AM
System Default
SYSTEM


Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Product
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
Object Access
Triggers
Edit Product Compact Layout
Product Compact Layout
Help for this Page
Compact layouts are used in the mobile app and some Chatter feed items to display a record's key fields at a glance. You can select and prioritize up to ten fields for the compact layout, but the number of fields that display may vary based on the device's screen, which record page is being viewed, and the permissions of the user.
Compact Layout Edit
Save
Cancel
Enter Compact Layout Information
= Required Information
Label
Product Compact Layout
Name
Product_Compact_Layout
Select Compact Layout Fields
Available Fields
Created By
Expiry Date
Last Modified By
Minimum Stock Level
Owner
Product ID
Add
Remove
Selected Fields
Product Name
Unit Price
Current Stock Level
Top
Up
Down
Bottom
Use SHIFT + click to select adjacent fields. Use CTRL + click to select an assortment of fields.
Save
Cancel


Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Product
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Product Compact Layouts
Compact Layout Assignment
Help for this Page
Save
Cancel
Primary Compact Layout
Select the compact layout to use when this object's records appear as list items in the mobile app.
Primary Compact Layout:
Product Compact Layout
Save
Cancel

10.1Created a Compact Layout to a Purchase Order Object





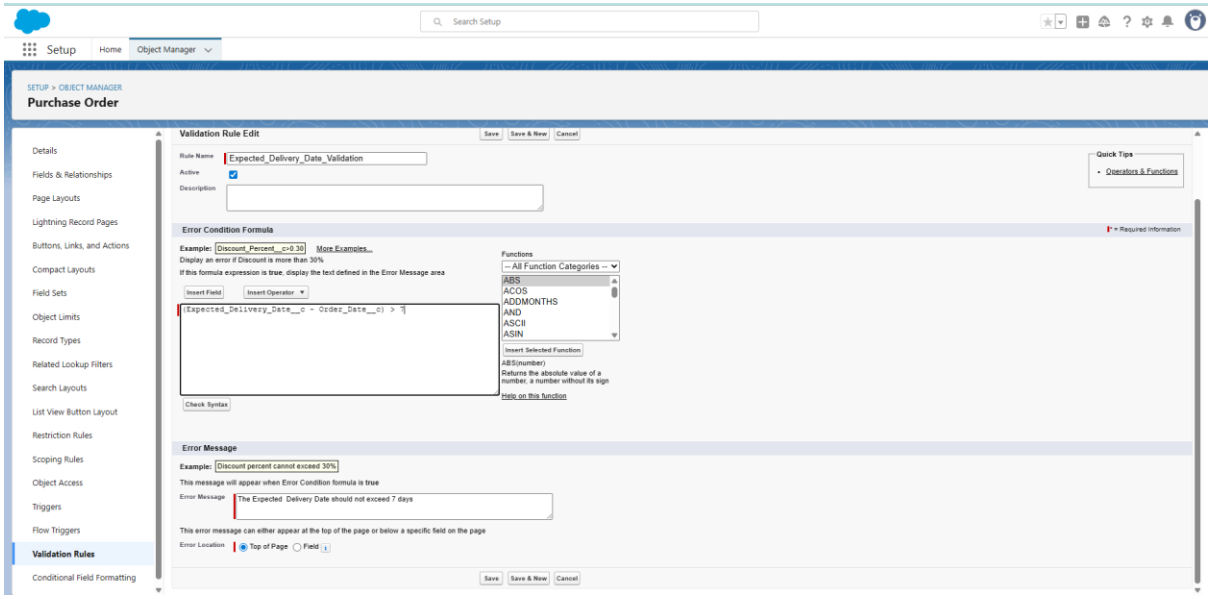
11.Create an Expected Delivery Date Validation rule to a Purchase Order Object

1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Expected Delivery Date Validation”.
4. Select Active
5. Insert the Error Condition Formula as :
   (Expected_Delivery_Date__c - Order_Date__c)> 7

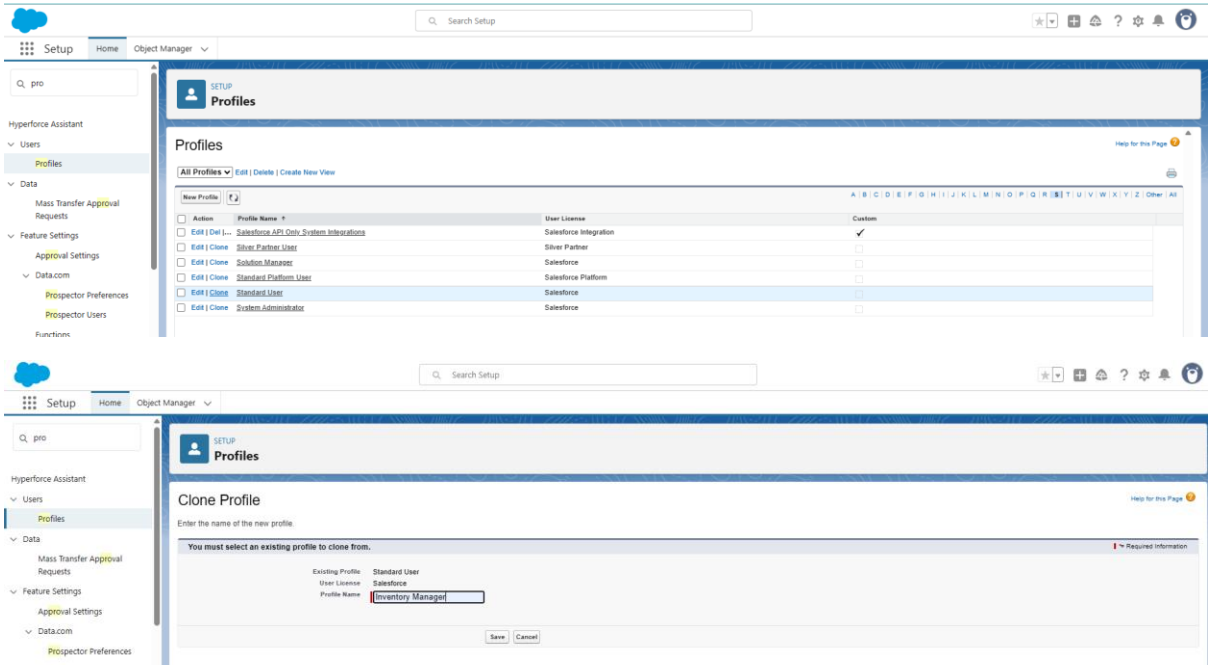
6. Enter the Error Message as “The Expected Delivery Date should not exceed 7 days.”.

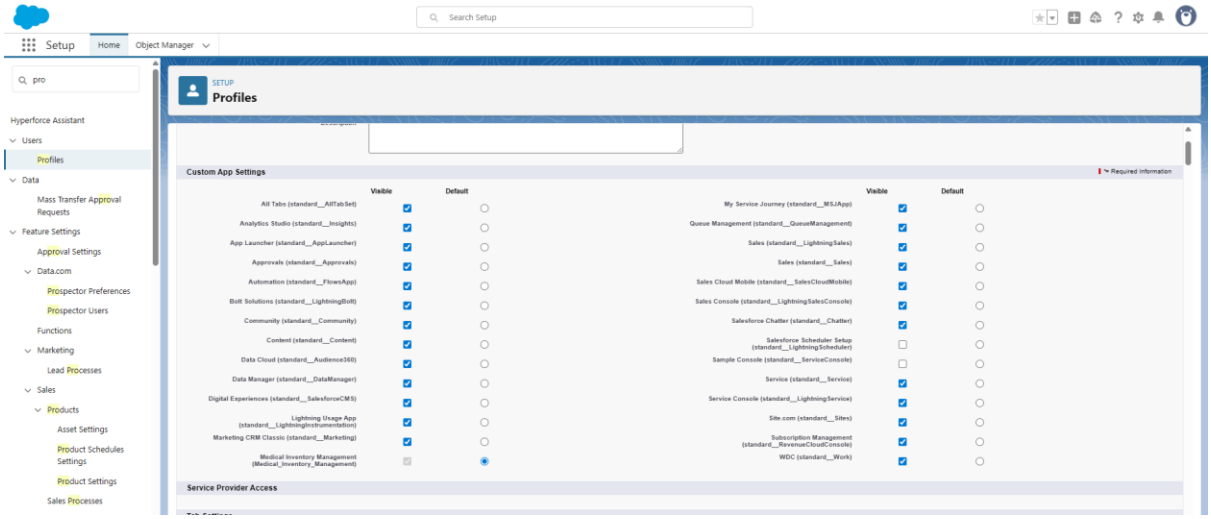
7. Select the Error location as Top of Page

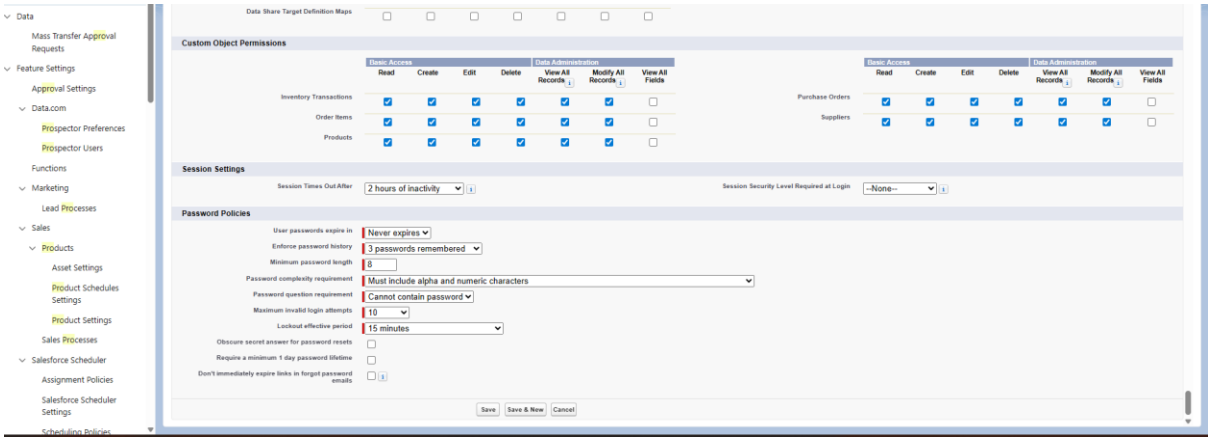
8. Click Save.



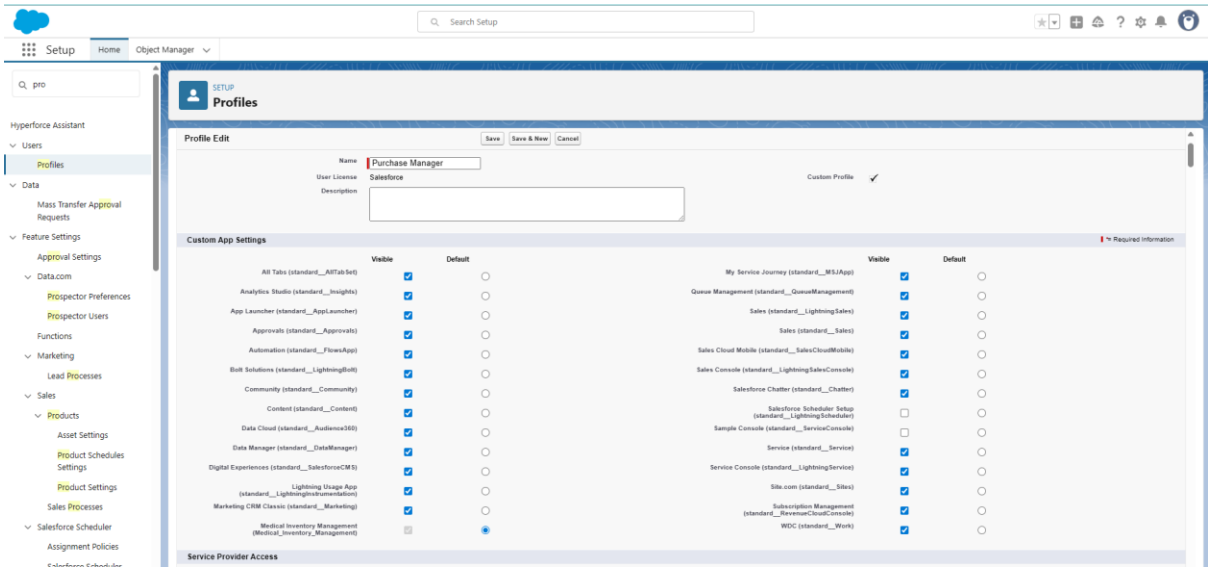
12.Created an Inventory Manager Profile

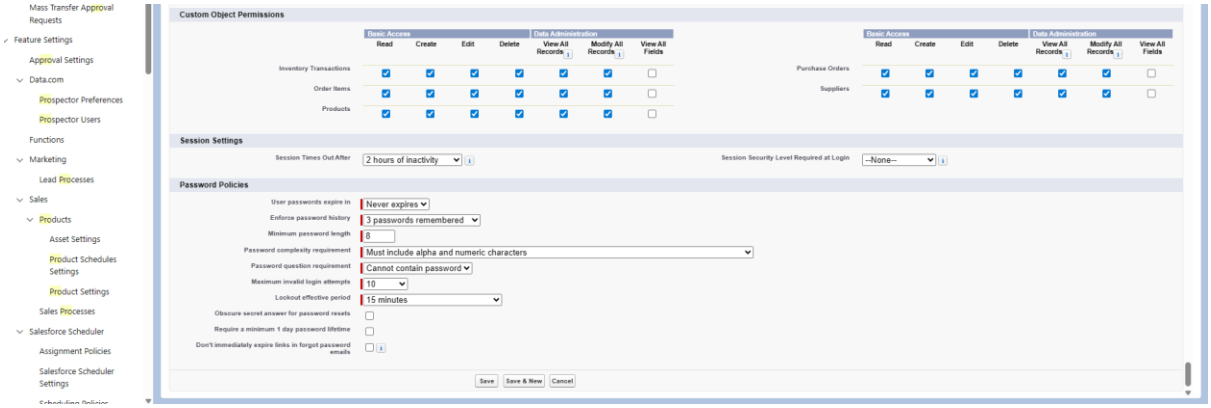




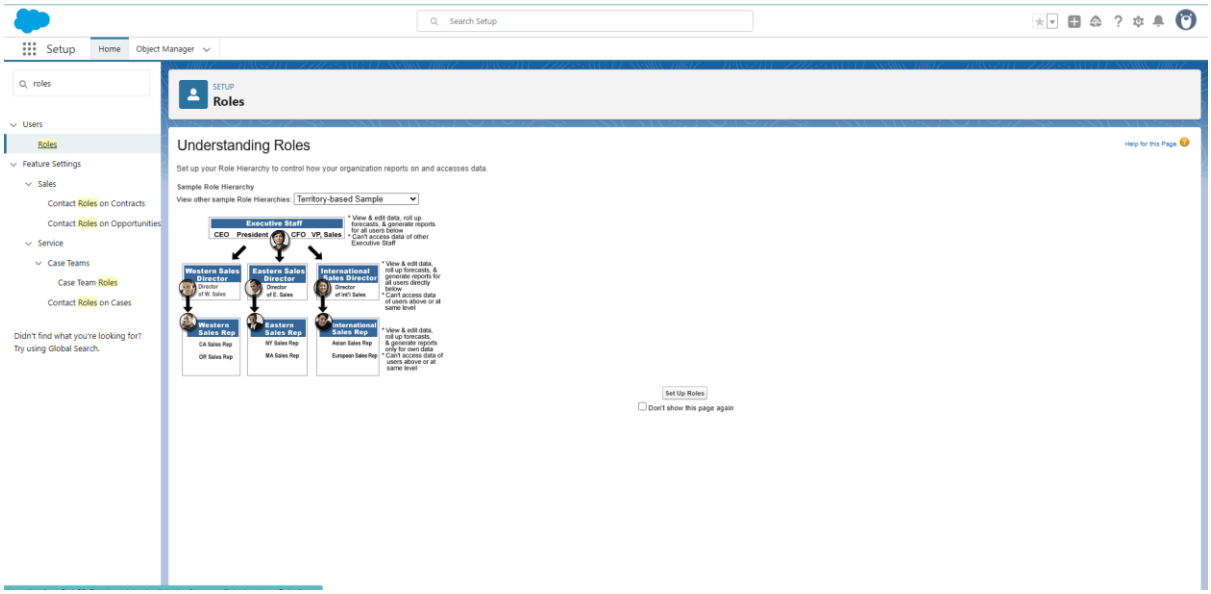


12.1Created an Purchase Manager Profile

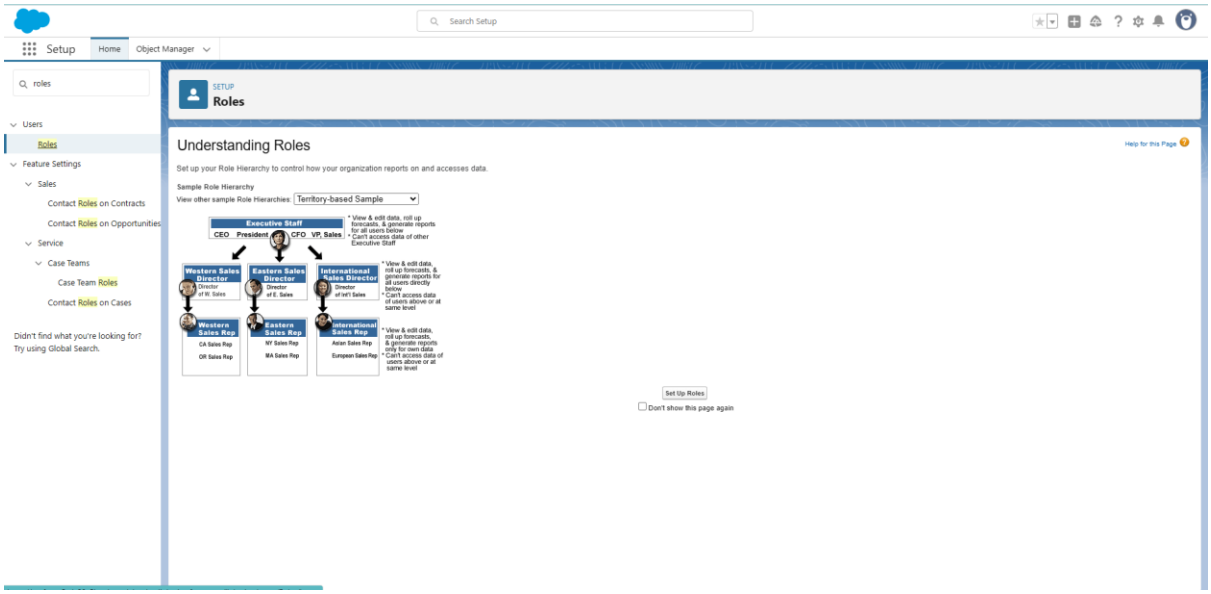


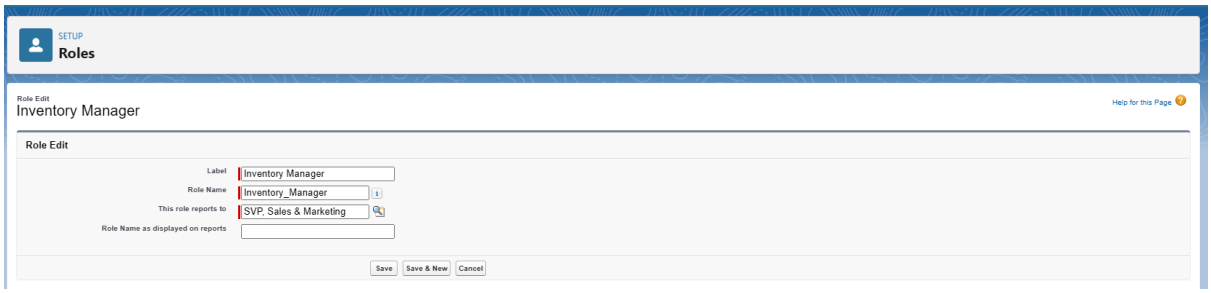


13.Created a Purchasing Manager Role.



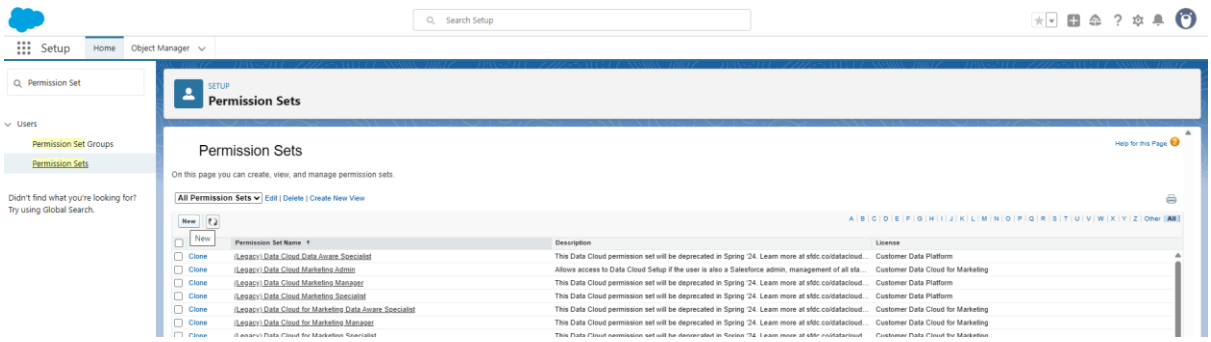
13.1Created a Inventory Manager Role.



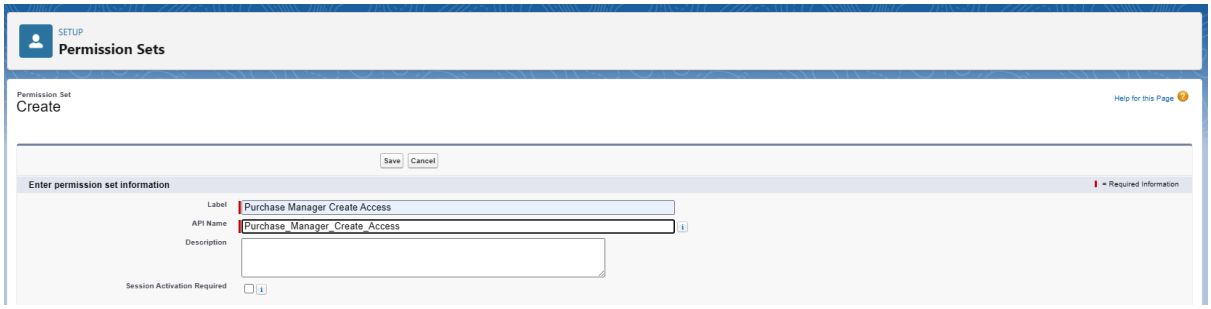


14.Created a Permission Set.

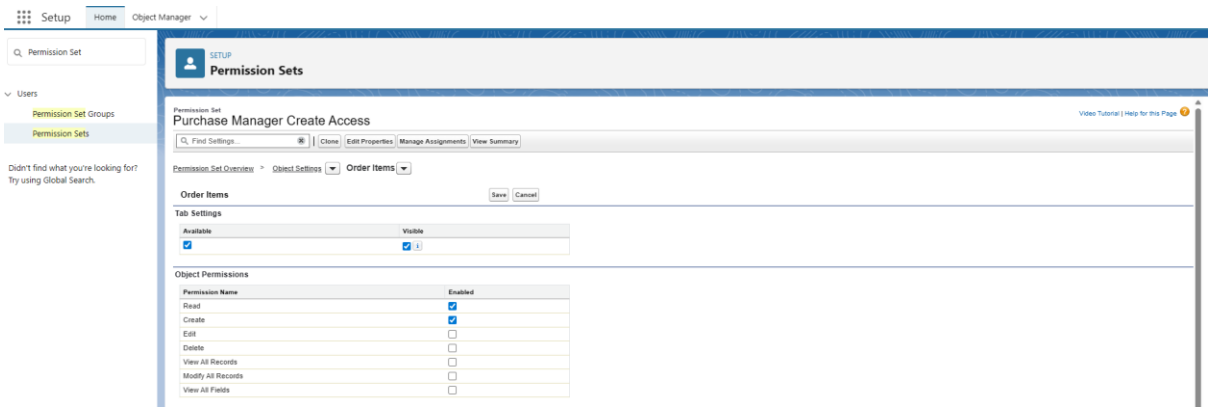
1. Go to setup >> type Permission in quick find box >> Select Permission Set >> click on New.



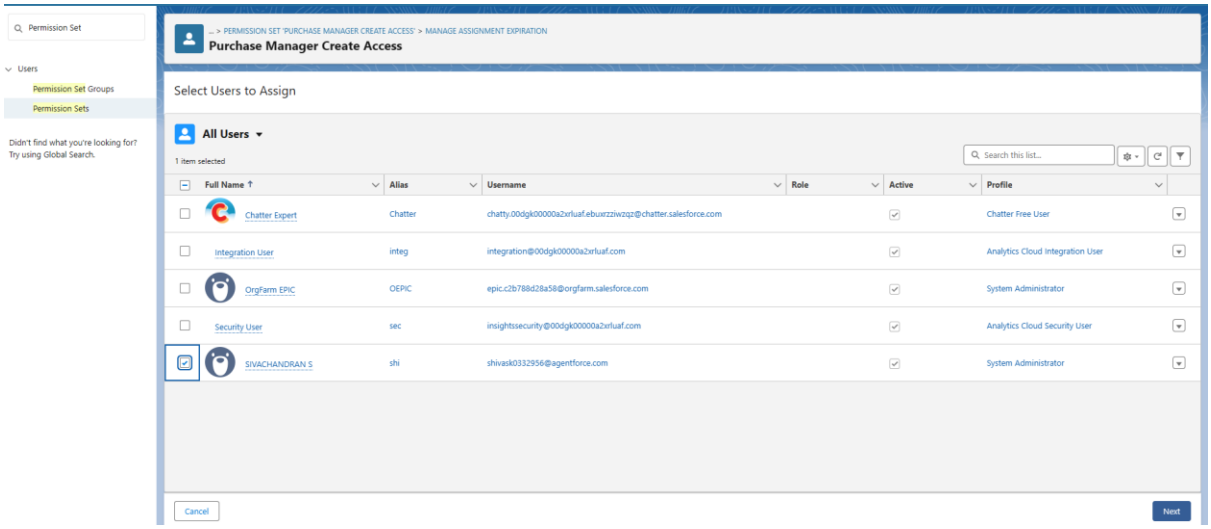
2. Enter Label as Purchase Manager Create Access >> Click on Save.



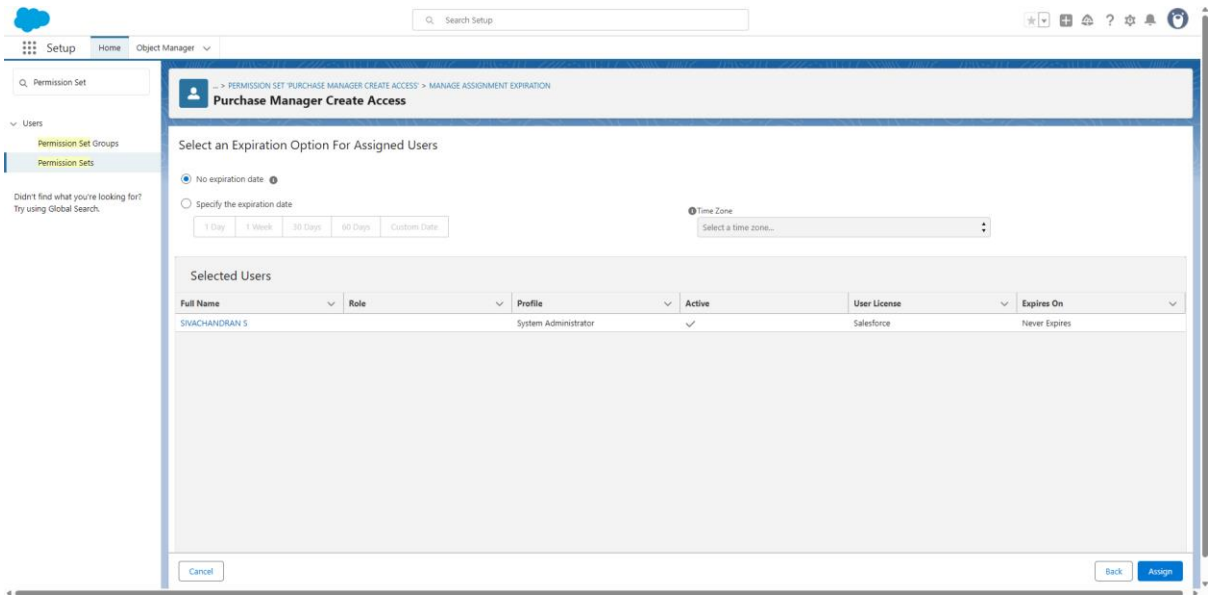
3. From Object Settings >> Select Order Item >> Enable for both Tab Available and Visible >> Enable Read and Create in Object Permissions >> Click on Save.



4. Navigate to the Permission Set detail page >> Click Manage Assignments >> Click Add Assignments >> Select the user SIVACHANDRAN S to assign the permission set to and click Next.



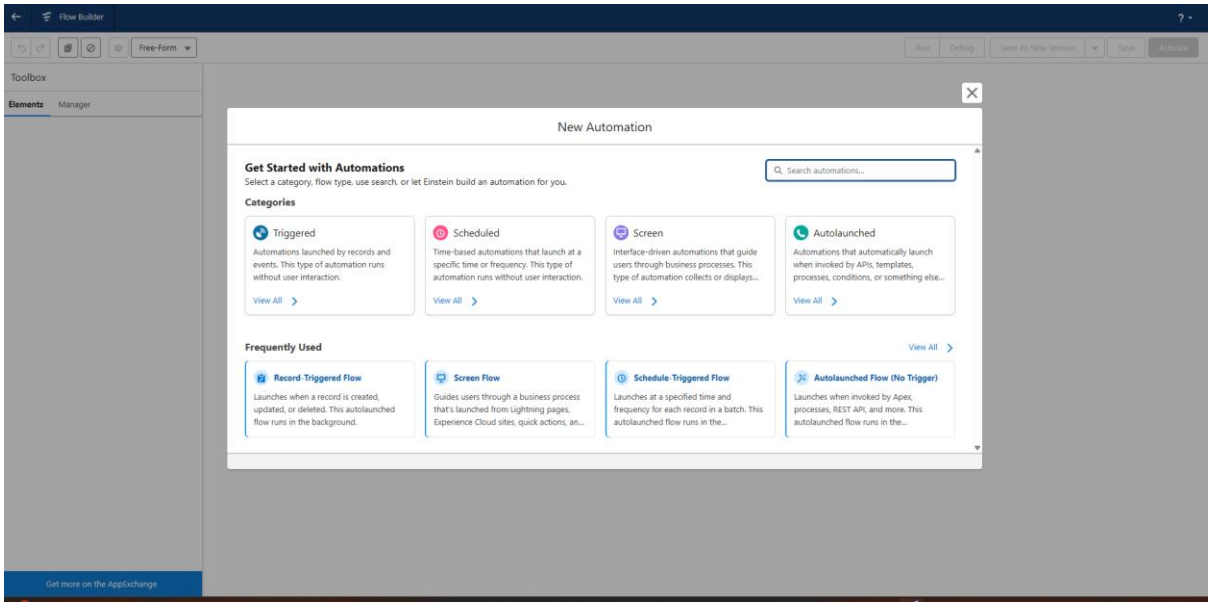
5. Select No Expiration date >> Click on Assign.



15.Created Flow to update the Actual Delivery Date.

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow >> Start From Scratch .

2. Select the record Triggered flow.Click on create.

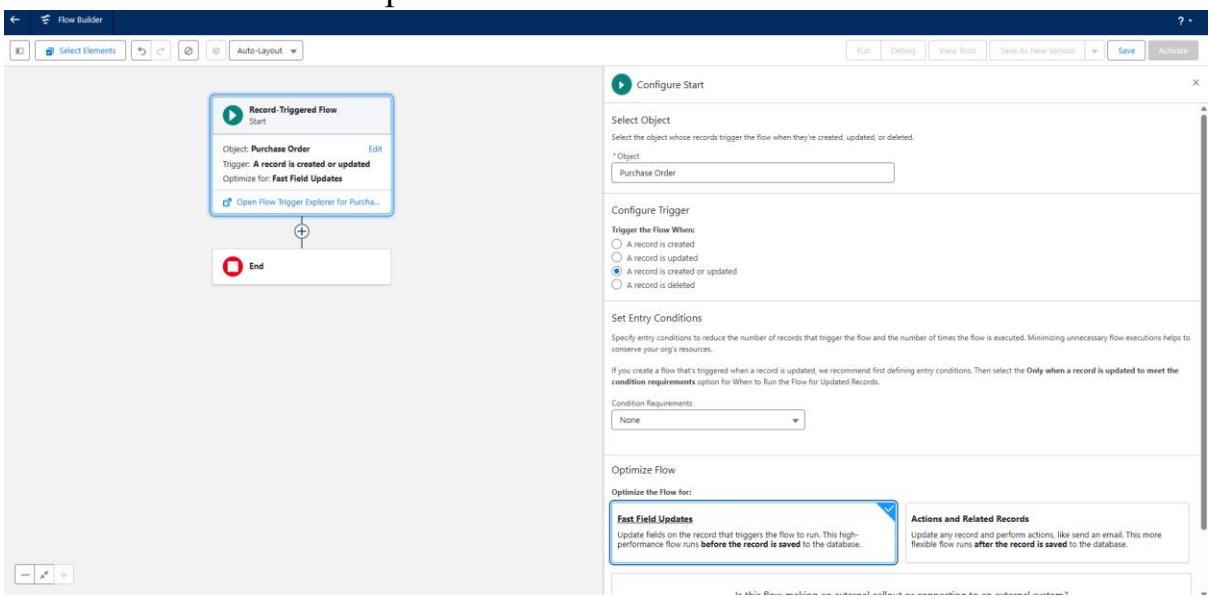


3. Under Object select “Purchase Order”

4. Select A record is created or updated

5. Set Entry Conditions : None

6. Select Fast Field Updates and click on Done



7. Under the record trigger flow click on the “+” icon and select Get Records.

8. Enter Label as “ Get Purchase Record ”.

9. For Object select Purchase Order.

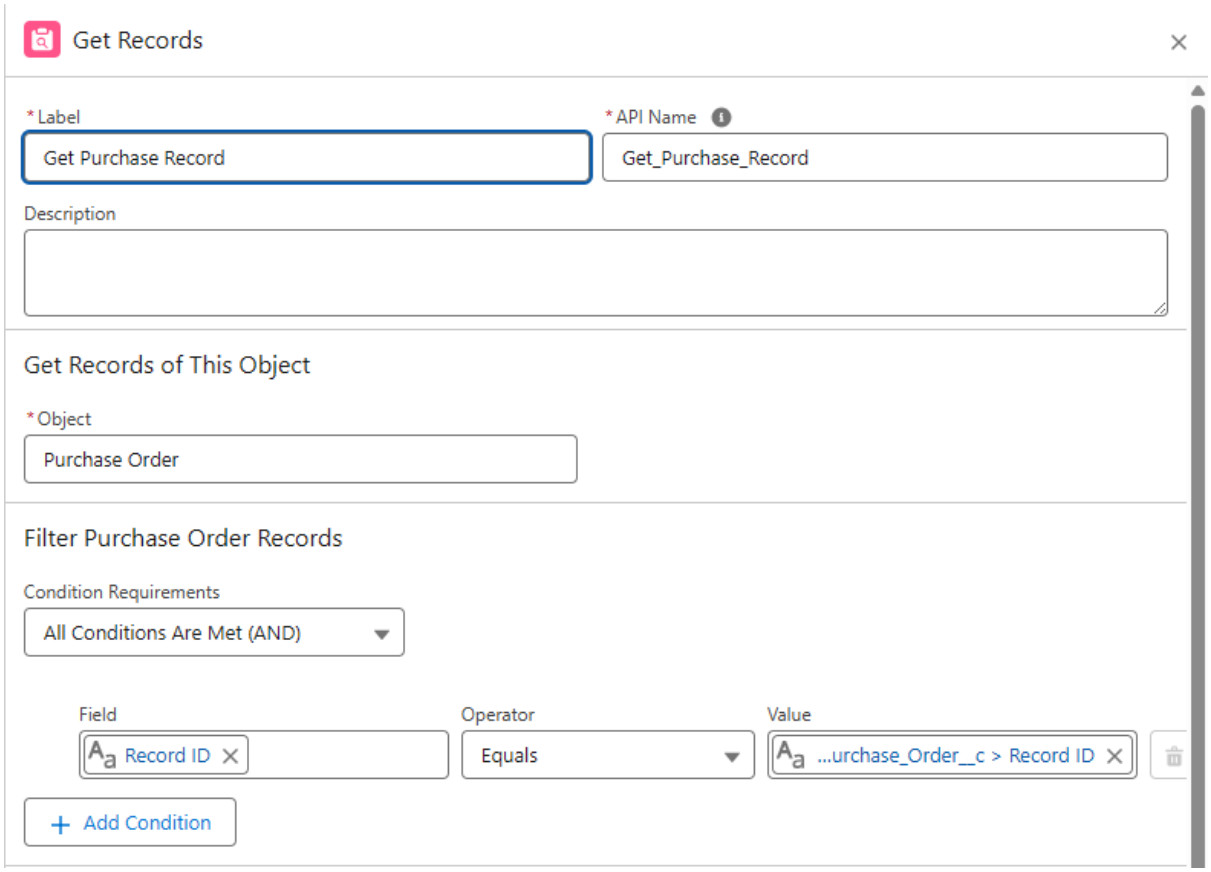
10. For Condition Requirements , select All Conditions are Met(AND)

For the first condition select as follows:

Field: Id

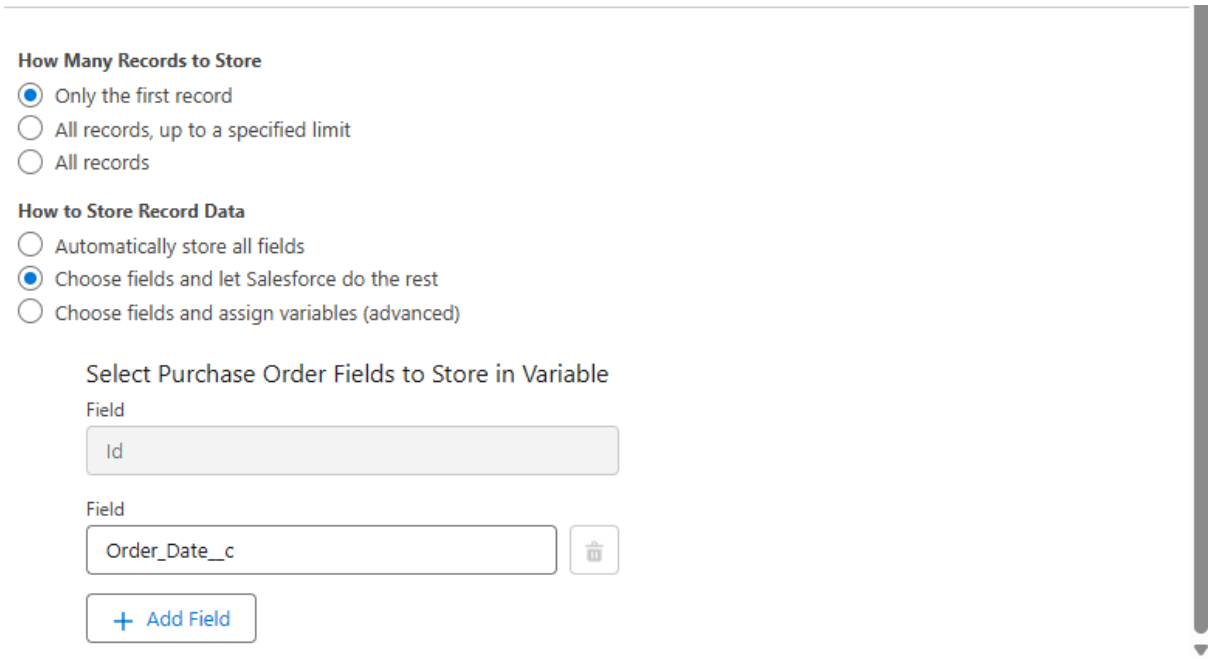
Operator: Equals

Value: {!$Record.Id}



11. For How many Records to store Select Only the First Record.

12. For How to Store Record Data select Choose fields and let Salesforce do the rest. Select Field: Order_Date__c. Click on Done.



13. In the Flow Builder, click on the Manager tab on the left-hand side >> Click on New Resource >> In the Resource Type dropdown, select Variable.

14. Enter API name as ActualDeliveryDate >> Select Data type as Date >> Click on Done.

15. From the Toolbox drag and drop Assignment element.

16. Enter the label as “Assignment”.

17. Set Variable Values:

    a) Variable : {!ActualDeliveryDate}

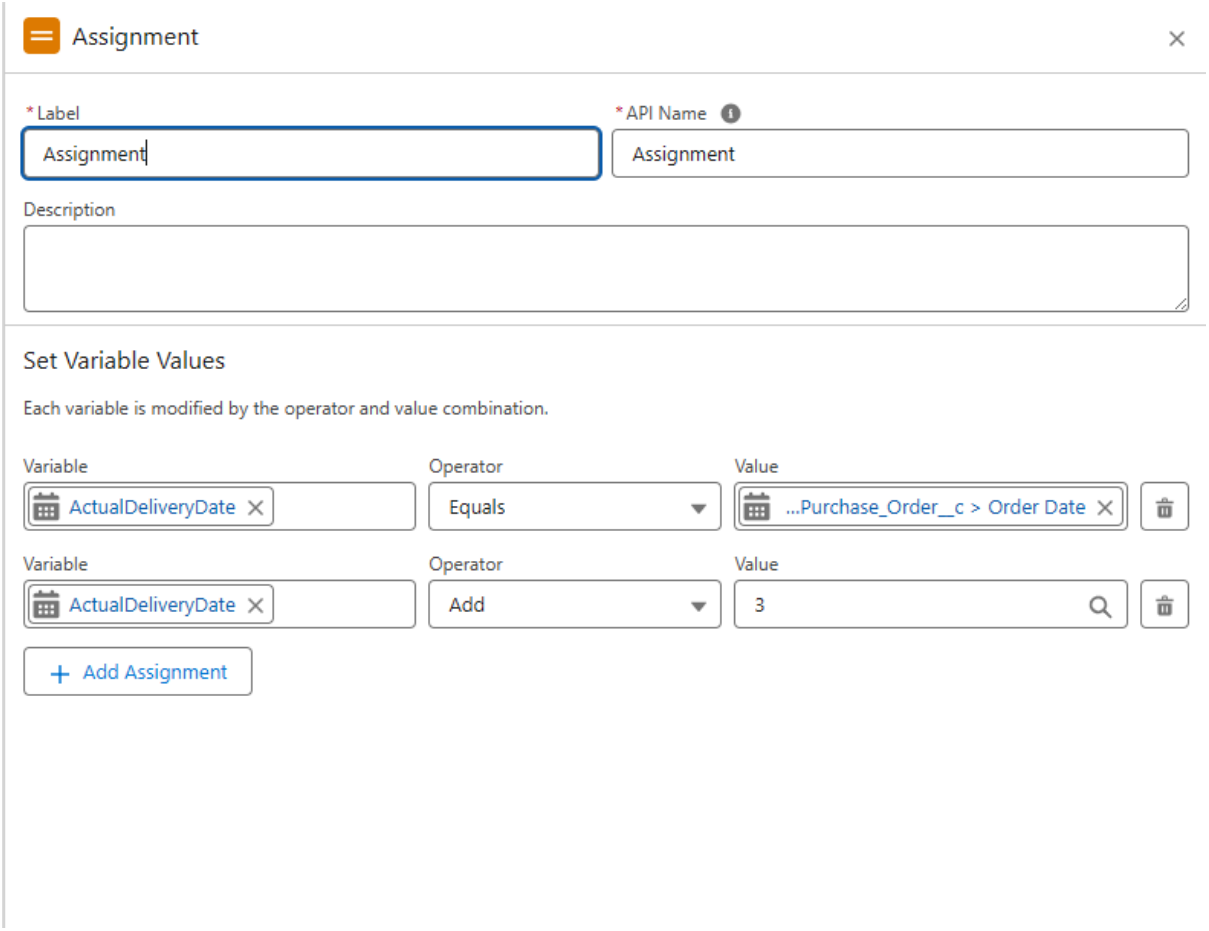
       Operator : Equals

       Value : {!$Record.Order_Date__c}

    b) Variable : {!ActualDeliveryDate}

       Operator : Add

       Value : 3



18. Click Done

19. From the Toolbox drag and drop Update Records element and connect to the Assignment element.

20. Enter the label as “Updating Purchasing Order”.

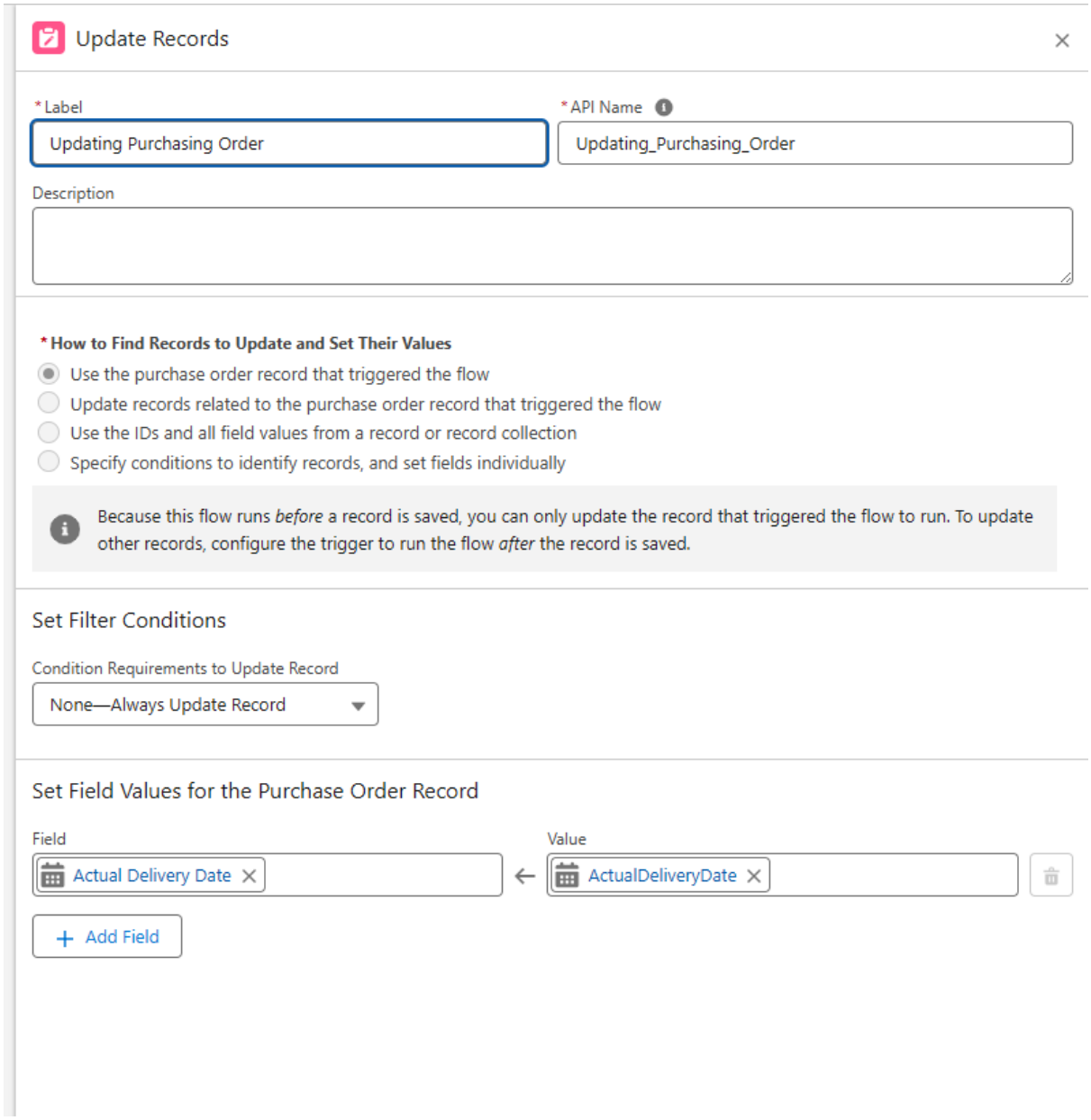
21. How to Find Records to Update and Set Their Values : Use the Purchase Order record that triggered the flow

22. Set Filter Conditions : None -Always Update Record

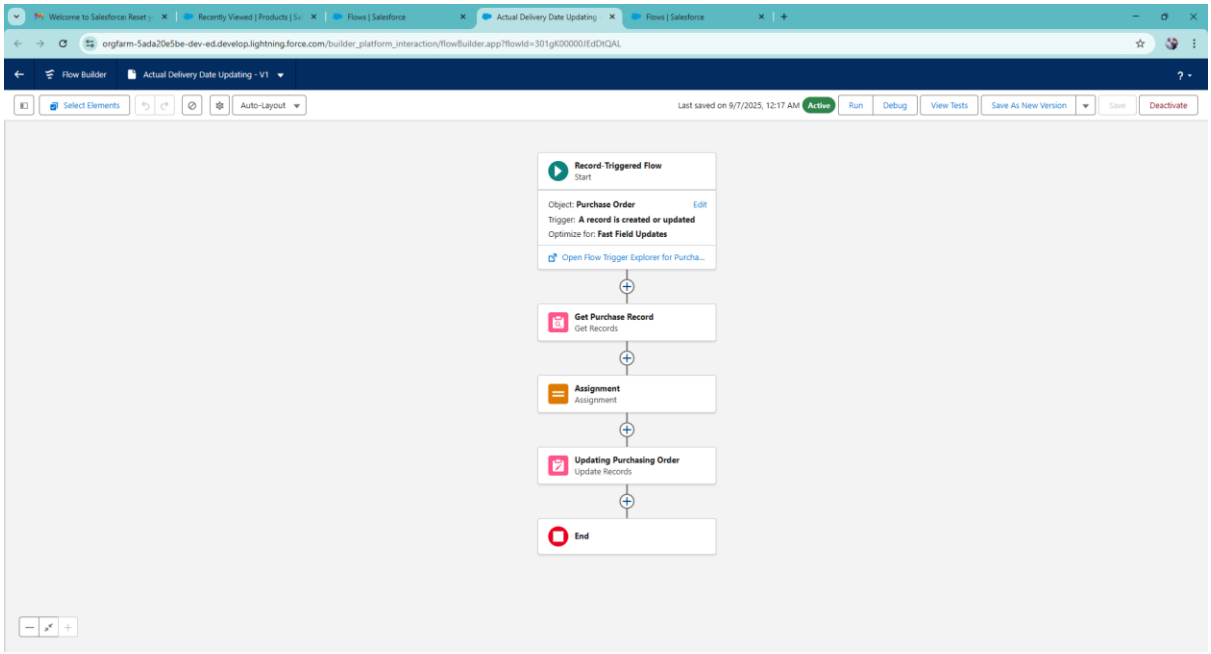
23. Set Field Values for the Trip Record as

Field : Actual_Delivery_Date__c

Value : {!ActualDeliveryDate}



24. Click Done
25. Save the flow as “Actual Delivery Date Updating”.
26. Activate the flow.



16.Created 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:

i)Navigate to Setup: Once logged in, click on the gear icon ?? (Setup) located at the top-right corner of the page. This will open the Setup menu.

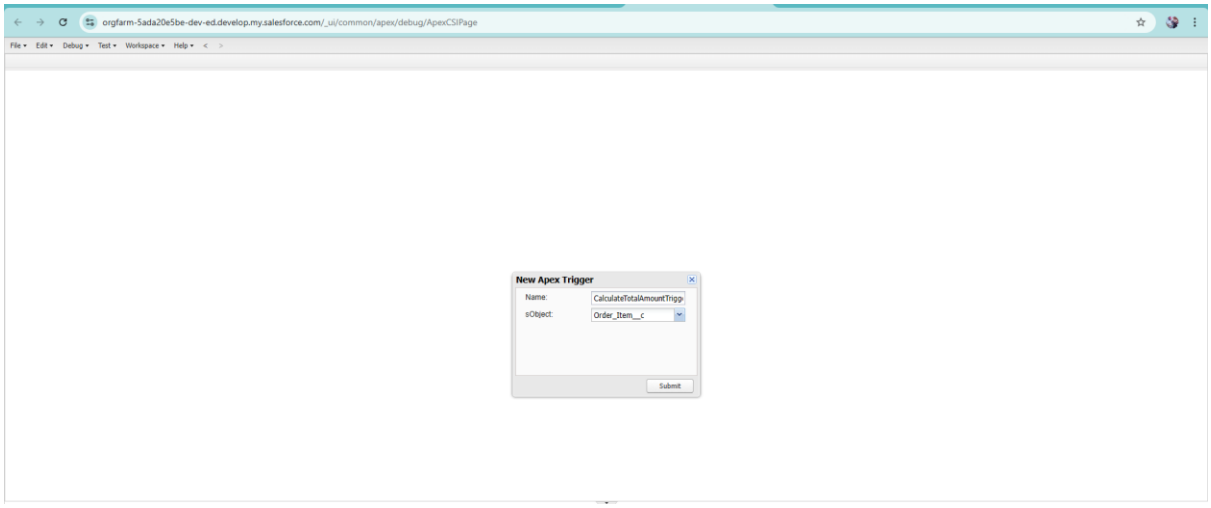
ii)Click on Developer Console: Click on the "Developer Console" option from the Setup menu. This will open the Developer Console in a new browser tab or window.

Step 3:

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

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

iii)Choose Apex Trigger: This will open a new Apex Trigger editor tab.



Create an Apex Trigger:

```
trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {
    // Call the handler class to handle the logic
    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);
}
```

Step 4:

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

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

iii)Choose Apex Class: Name it as CalculateTotalAmountHandler

```
public class CalculateTotalAmountHandler {
```

```
    // Method to calculate the total amount for Purchase Orders based on related Order Items
    public static void calculateTotal(List<Order_Item__c> newItems, List<Order_Item__c> oldItems, Boolean isInsert, Boolean isUpdate, Boolean isDelete, Boolean isUndelete) {

        // Collect Purchase Order IDs affected by changes in Order_Item__c records
        Set<Id> parentIds = new Set<Id>();

        // For insert, update, and undelete scenarios
        if (isInsert || isUpdate || isUndelete) {
            for (Order_Item__c ordItem : newItems) {
                parentIds.add(ordItem.Purchase_Order_Id__c);
            }
        }

        // For update and delete scenarios
        if (isUpdate || isDelete) {
            for (Order_Item__c ordItem : oldItems) {
                parentIds.add(ordItem.Purchase_Order_Id__c);
            }
        }

        // Calculate the total amounts for affected Purchase Orders
        Map<Id, Decimal> purchaseToUpdateMap = new Map<Id, Decimal>();

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

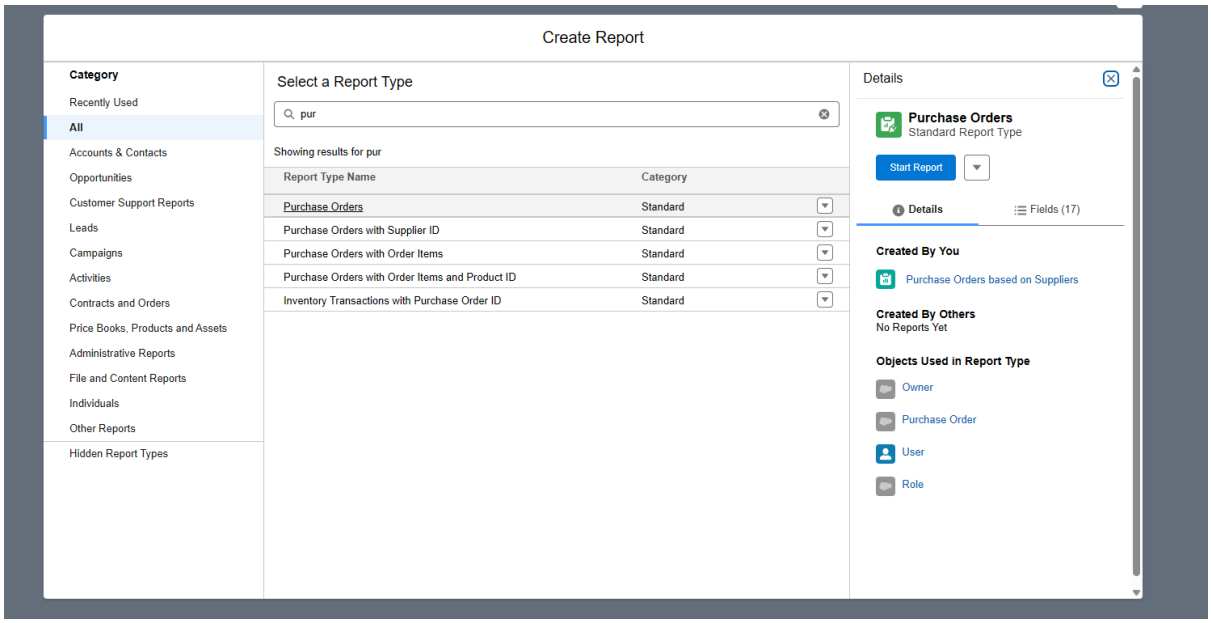
```
        // Perform an aggregate query to sum the Amount__c for each Purchase
Order
        List<AggregateResult> aggrList = [
            SELECT Purchase_Order_Id__c, SUM(Amount__c) totalAmount
            FROM Order_Item__c
            WHERE Purchase_Order_Id__c IN :parentIds
            GROUP BY Purchase_Order_Id__c
        ];

        // Map the result to Purchase Order IDs
        for (AggregateResult aggr : aggrList) {
            Id purchaseOrderId = (Id)aggr.get('Purchase_Order_Id__c');
            Decimal totalAmount = (Decimal)aggr.get('totalAmount');
            purchaseToUpdateMap.put(purchaseOrderId, totalAmount);
        }

        // Prepare Purchase Order records for update
        List<Purchase_Order__c> purchaseToUpdate = new
List<Purchase_Order__c>();
        for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {
            Purchase_Order__c purchaseOrder = new Purchase_Order__c(Id =
purchaseOrderId, Total_Order_cost__c =
purchaseToUpdateMap.get(purchaseOrderId));
            purchaseToUpdate.add(purchaseOrder);
        }

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

```
        }
    }
  }
}
```

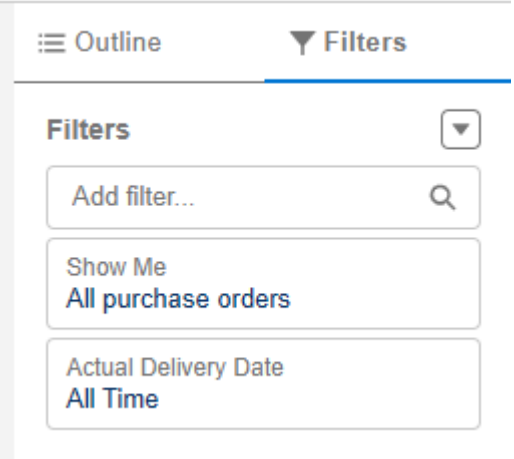
Save all.

17.Created 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).

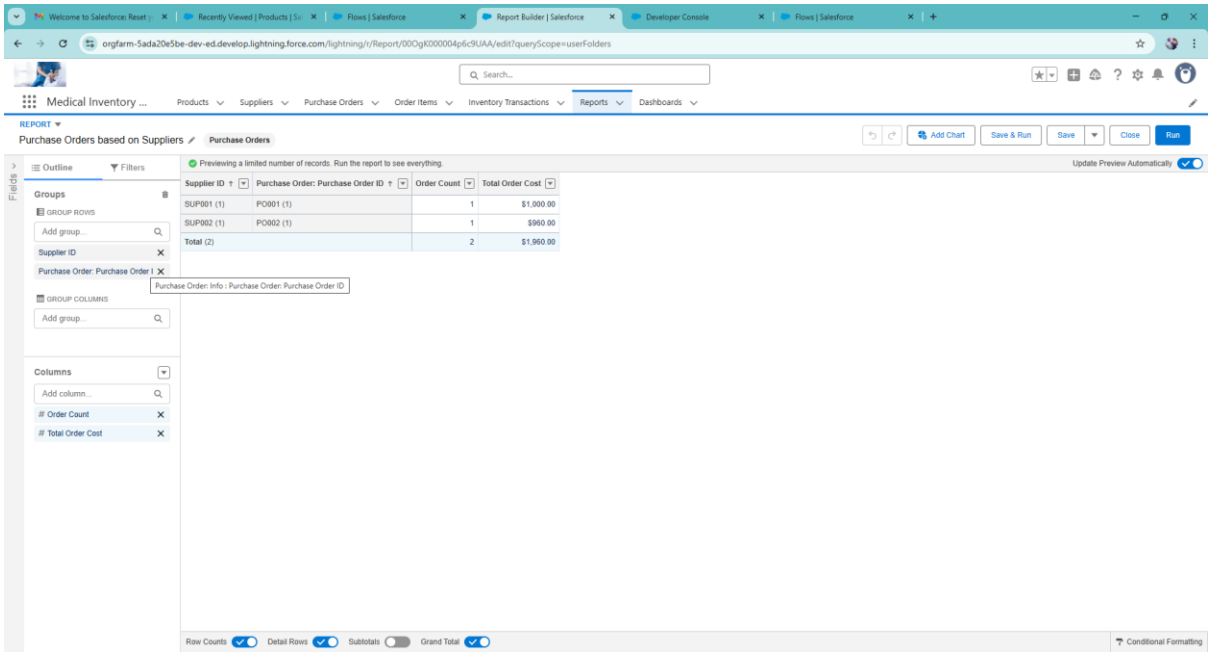
8. Click save and run

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”)



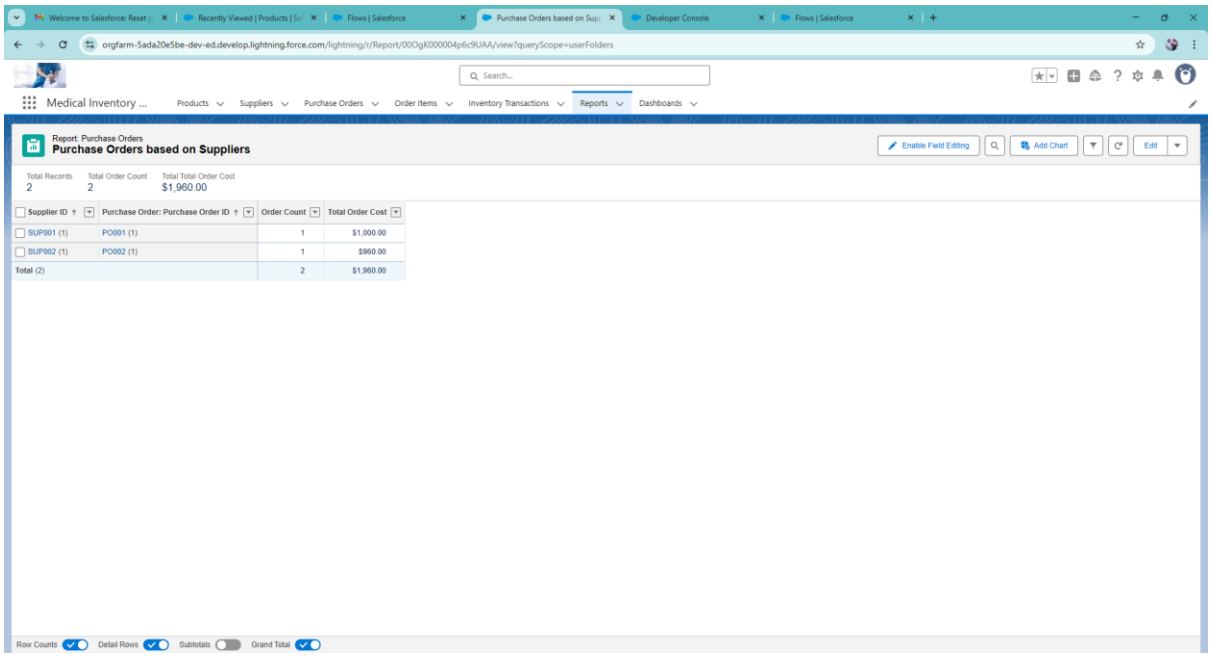
View Report

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.



17.1Created a Complete Purchase Details Report

1. Click App Launcher

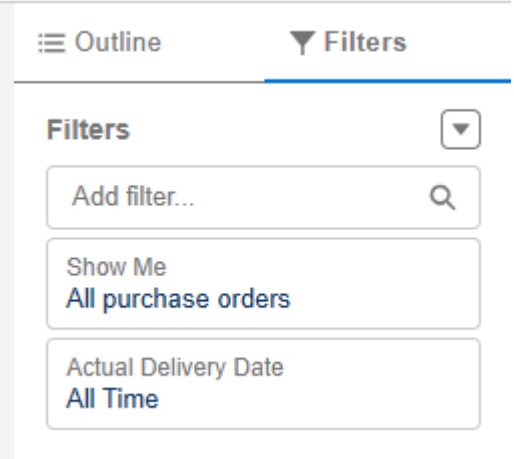
2. Select Medical Inventory Management App

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

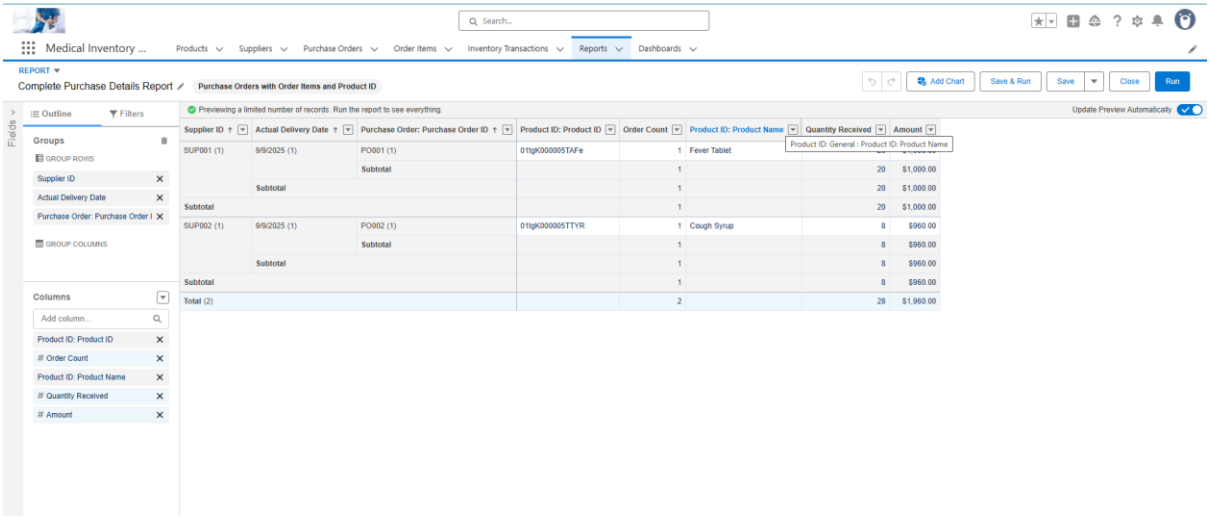


7. Customize your report, in group rows select – Supplier ID, Actual Delivery Date, Purchase Order: Purchase Order ID, for columns Product ID : Product ID, Product ID : Product Name, Order Count, Quantity Received, Amount (In this way we are making a Summary Report).

8. Click save and run

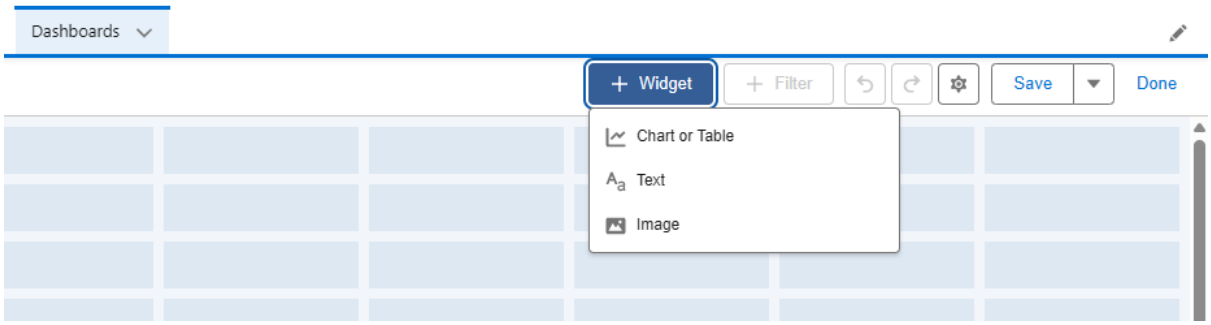
9. Give report name – Complete Purchase Details Report

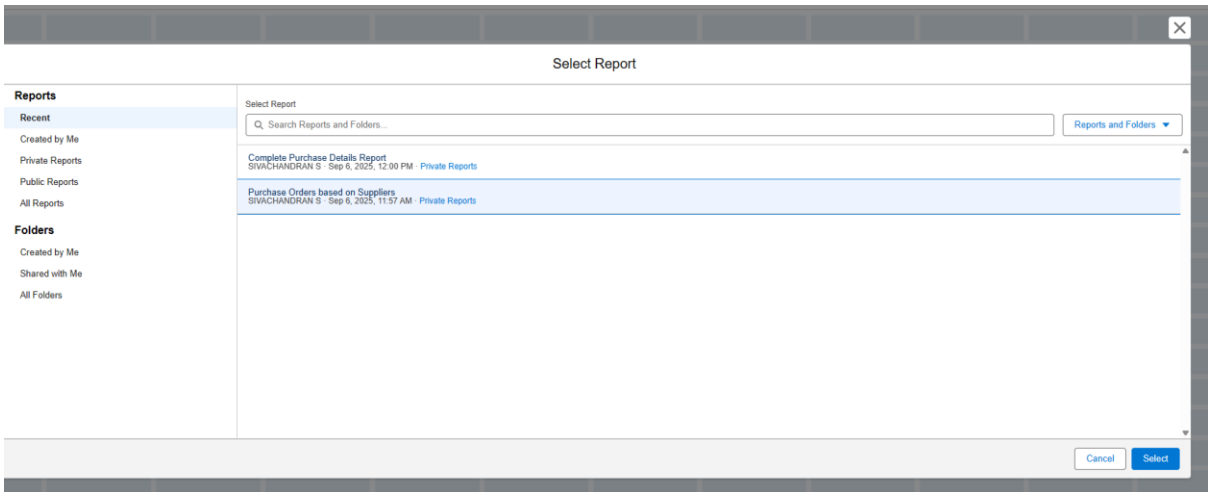
10. Click Save

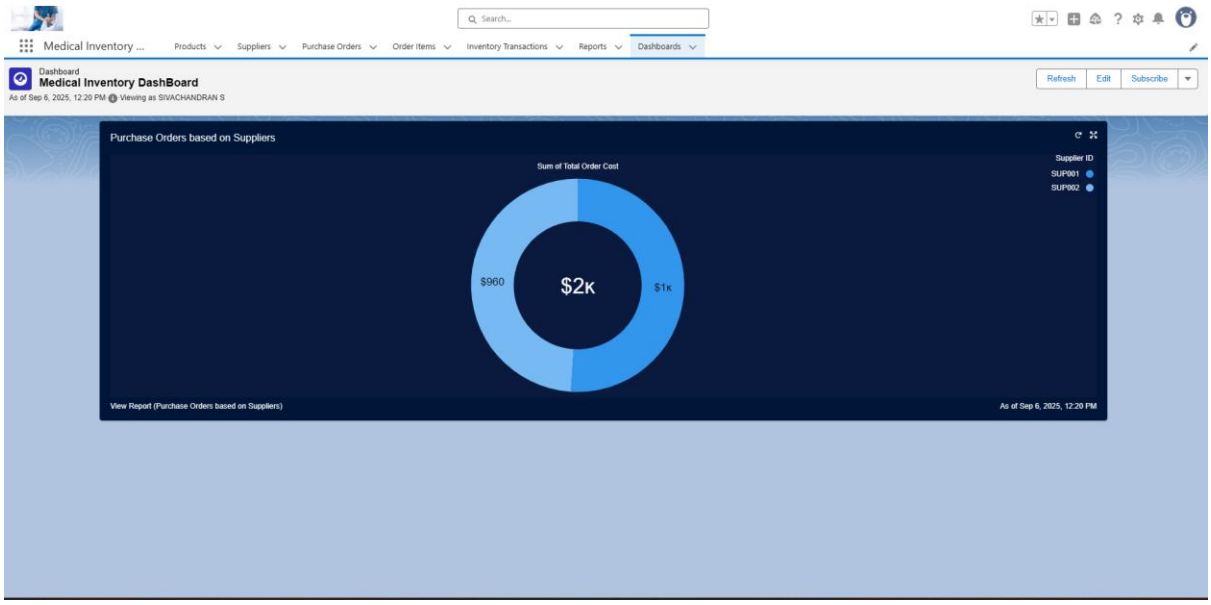


18. Created Dashboard

1. Click on the Dashboards tab from the Medical Inventory Management application.
2. Click on the new dashboard.
3. Give name - Medical Inventory DashBoard
4. Click create
5. Click on +widget
6. Select the Purchase Orders based on Suppliers Report
7. For the data visualization select any of the charts, tables etc. as per your choice/requirement
8. Click add.
9. Click save.





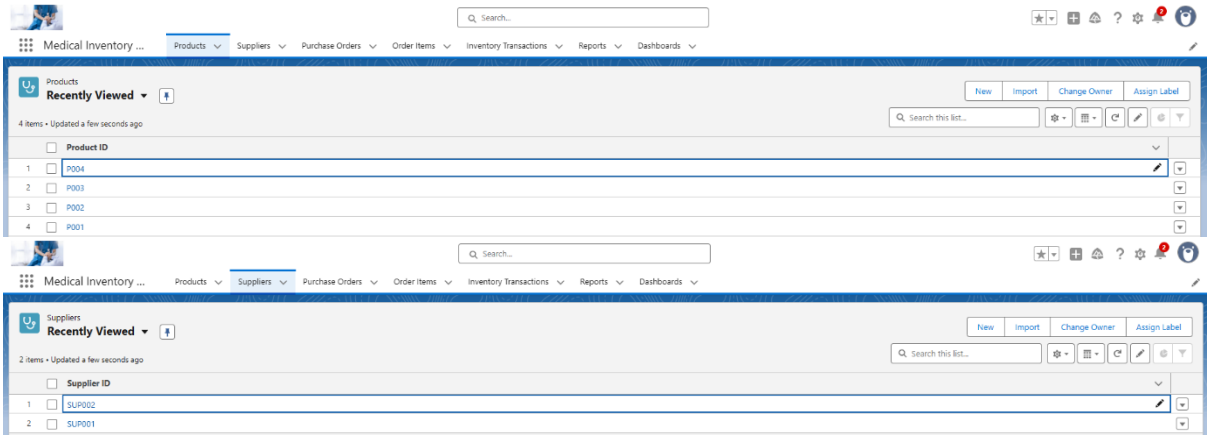


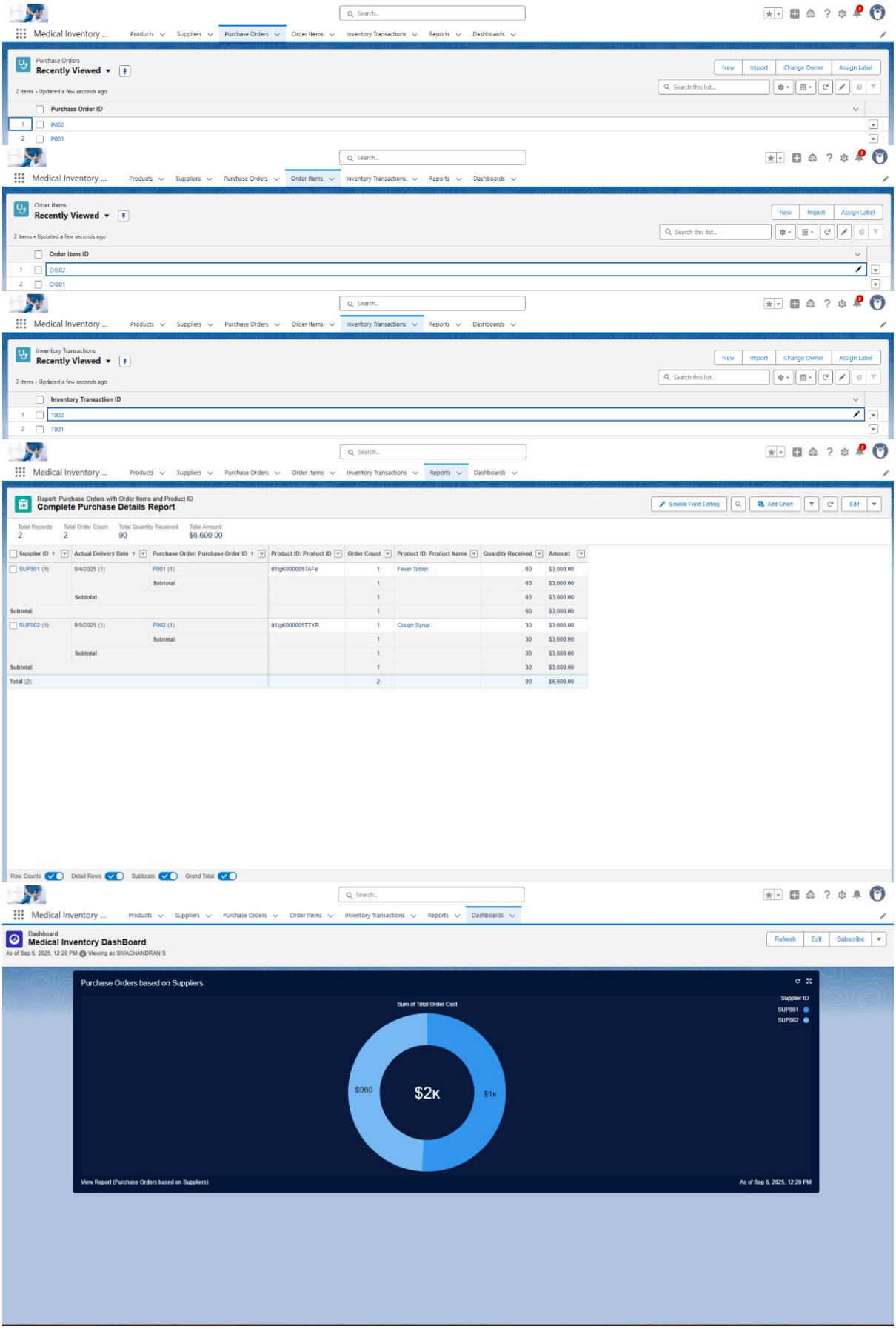
---

19. Testing Approach

To ensure the app works as intended:

- **Step 1:** Created sample suppliers, products, purchase orders, and transactions.
- **Step 2:** Verified validations by entering incorrect data.
- **Step 3:** Checked automation by adding inbound transactions to see if stock updated correctly.
- **Step 4:** Reviewed reports and dashboards to ensure accurate totals.




Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Purchase Orders
Recently Viewed
New
Import
Change Owner
Assign Label
Purchase Order ID
P002
P001
Order Items
Recently Viewed
Order Item ID
OI002
OI001
Inventory Transactions
Recently Viewed
Inventory Transaction ID
T002
T001
Report: Purchase Orders with Order Items and Product ID
Complete Purchase Details Report
Total Records 2
Total Order Count 2
Total Quantity Received 90
Total Amount $6,600.00
Supplier ID
Actual Delivery Date
Purchase Order: Purchase Order ID
Product ID: Product ID
Order Count
Product ID: Product Name
Quantity Received
Amount
SUP001 (1)
9/4/2025 (1)
P001 (1)
01tgK000005TAFe
1
Fever Tablet
60
$3,000.00
Subtotal
SUP002 (1)
9/5/2025 (1)
P002 (1)
01tgK000005TTYR
Cough Syrup
30
$3,600.00
Total (2)
Row Counts
Detail Rows
Subtotals
Grand Total
Dashboard
Medical Inventory DashBoard
Refresh
Edit
Subscribe
Purchase Orders based on Suppliers
Sum of Total Order Cost
Supplier ID
SUP001
SUP002
$960
$2K
$1K
View Report (Purchase Orders based on Suppliers)
As of Sep 6, 2025, 12:20 PM

20. Conclusion

The **Medical Inventory Management System** successfully demonstrates how Salesforce can be used to manage medical products, suppliers, and stock efficiently.
By using custom objects, validation rules, flows, reports, and dashboards, the application provides a complete solution for inventory management with real-time tracking and analytics.