PROJECT TITLE:

Medical Inventory Management

College Name:

Nandha Arts And Science College, Erode

College Code:

bru4j

TEAM MEMBERS:

Team Leader Name: ARUN KUMAR V

Email: arunvadivel734@gmail.com

Team Member1: BHARATH S

Email:bharathsundaramani@gmail.com

Team Member2:HEMANTH

KUMAR M

Email:hemanthm14321111@gmail.com

Team Member3: JEEVA T

Email: jeeva.9043884835@gmail.com

Team Member4: ARUN KUMAR V

Email: arunvadivel734@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:

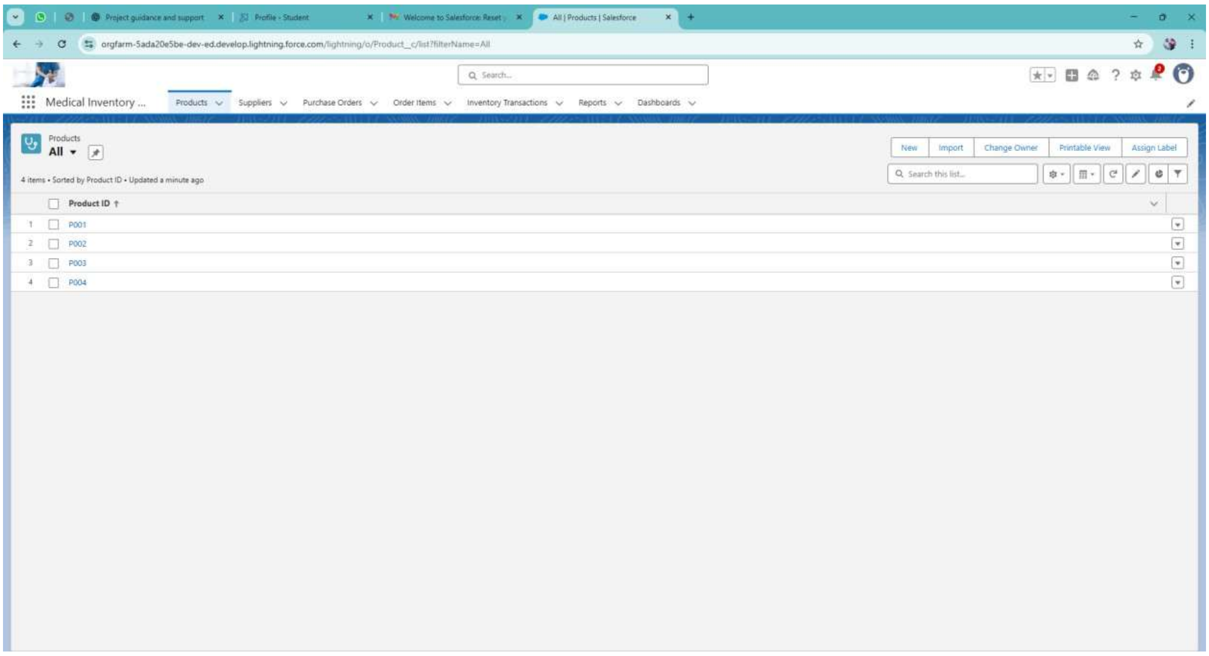
- Supplier tracking
- Stock level monitoring
- Order management
- Expiry date tracking
- Report generation and analytics

2. Objectives

The main objectives of this project are:

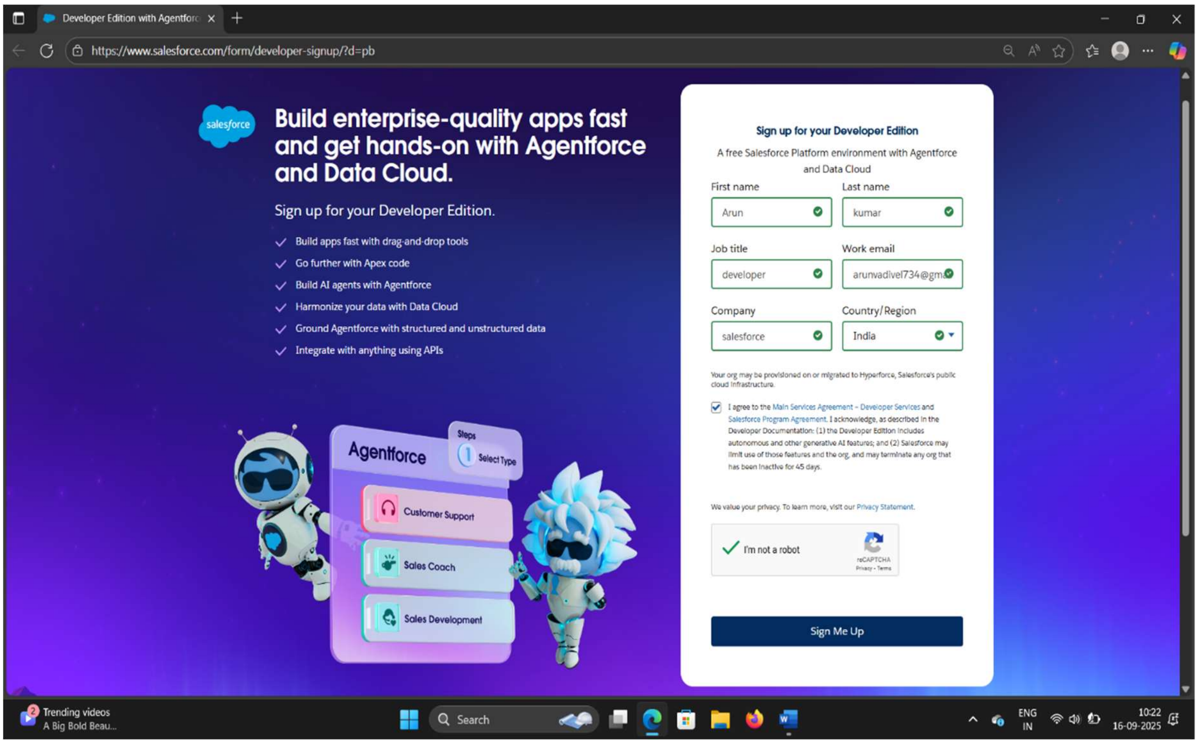
1. To efficiently manage and track **medical products** and their **stock levels**.
2. To streamline the **purchase order process** by linking suppliers and products.
3. To ensure **real-time monitoring** of stock and expiry dates.
4. To generate **insightful reports and dashboards** for decision-making.
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 :
3. 

1. First name & Last name
2. Email

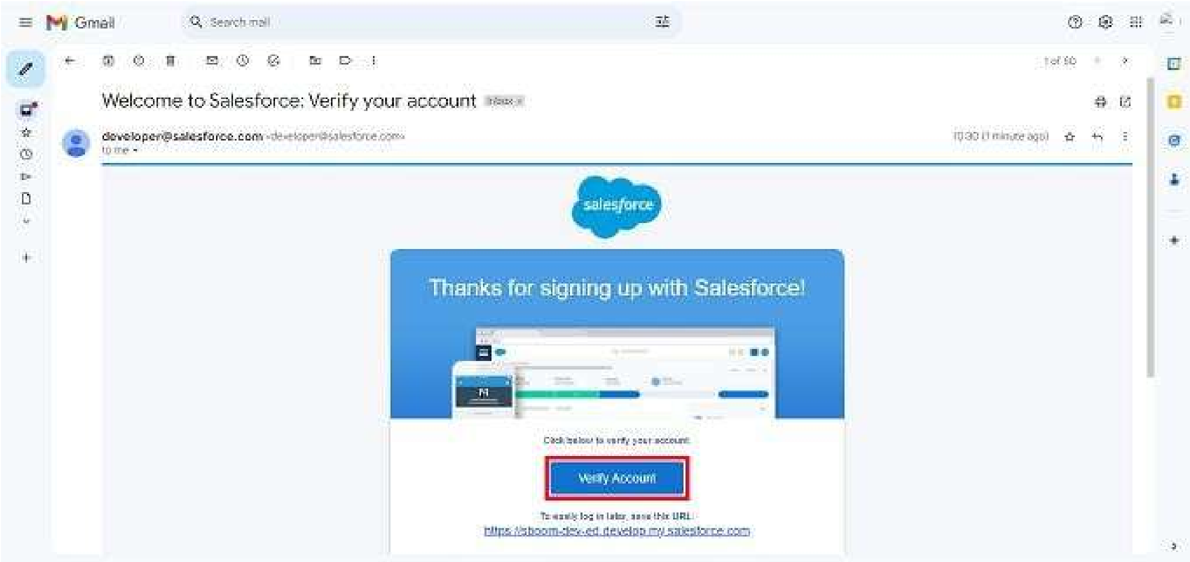
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code

Click on sign me up after filling these.

---

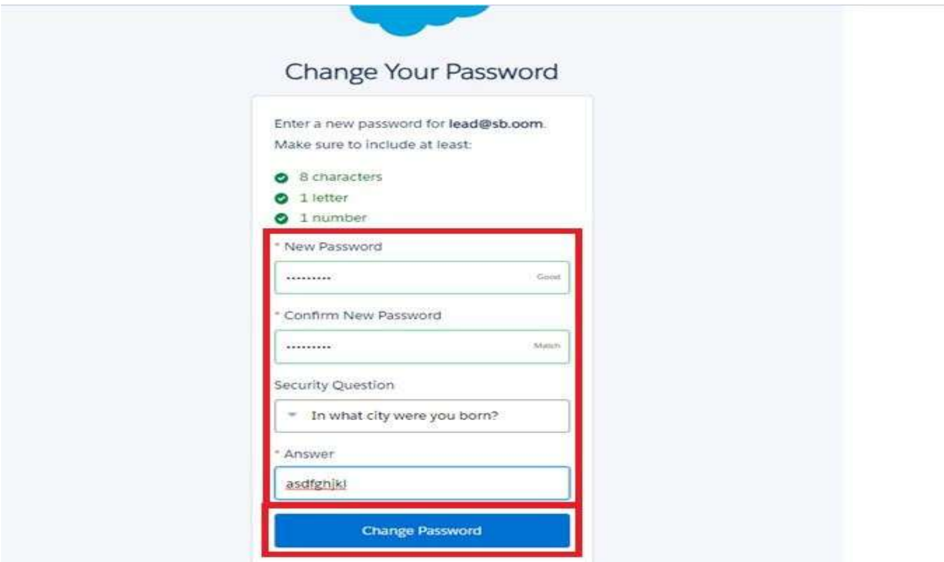
4. Account Activation

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.

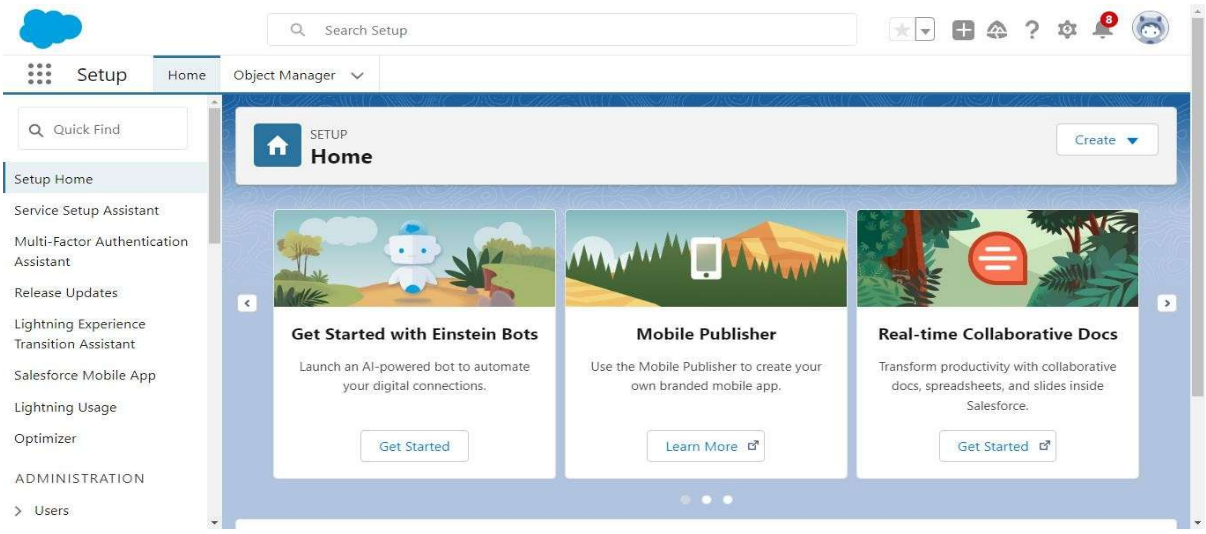


2. Click on Verify Account

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:

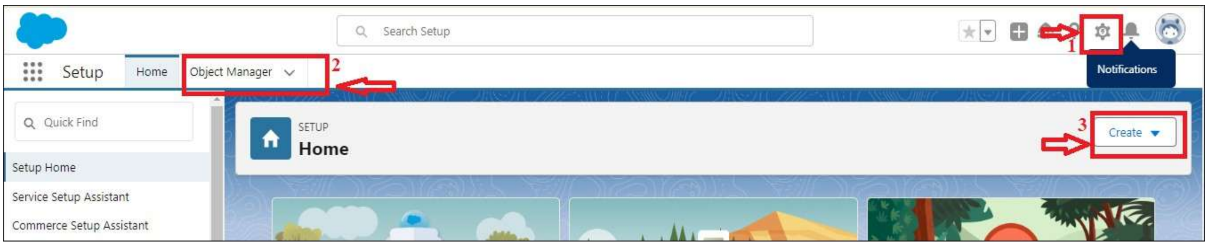
Object Name	Purpose
Supplier	To store supplier details like name, contact person, phone, and address.

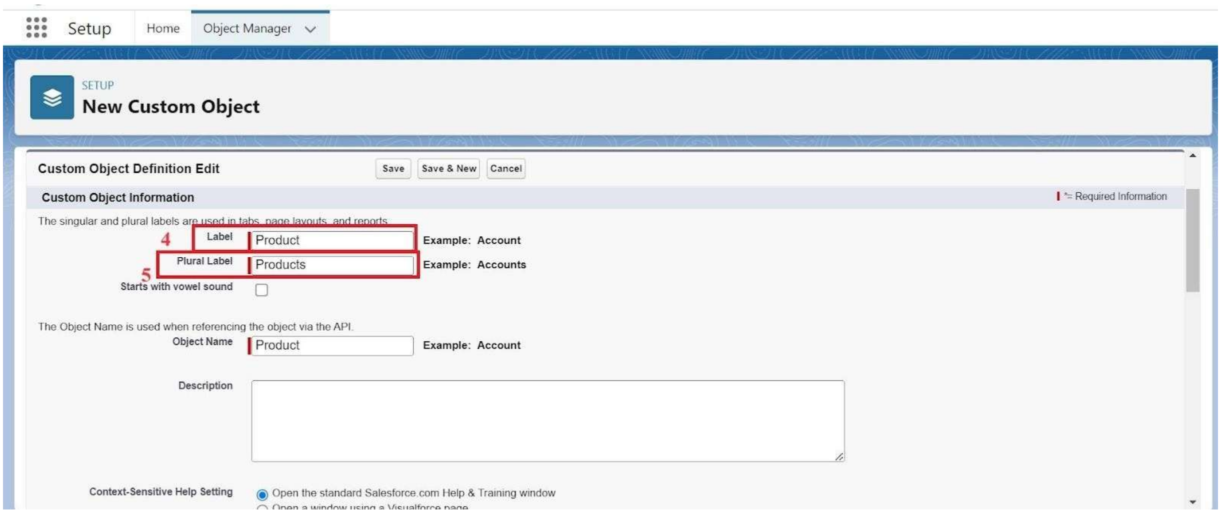
Object Name	Purpose
Product	To store product information like unit price, current stock level, and minimum stock level.
Purchase Order	To record order details from suppliers.
Order Item	To link products with each purchase order and track quantities and costs.
Inventory Transaction	To track stock movement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

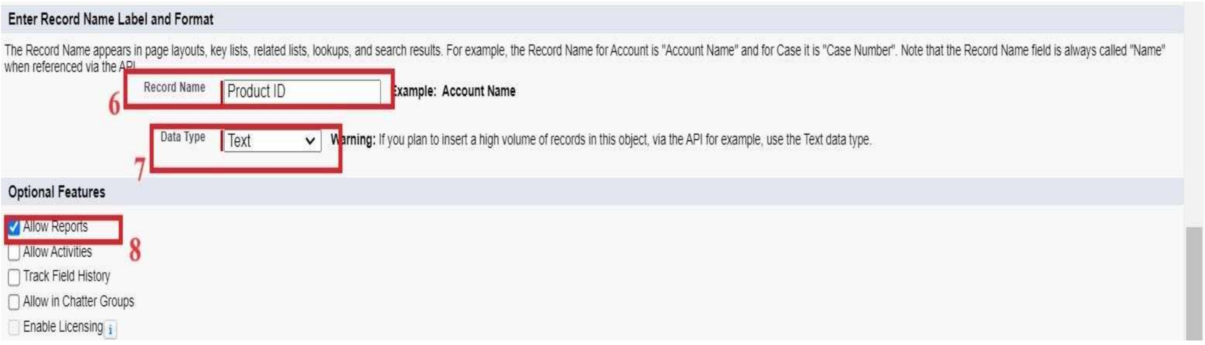
In Salesforce, objects are database tables that allow you to store data specific to your organization.

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.

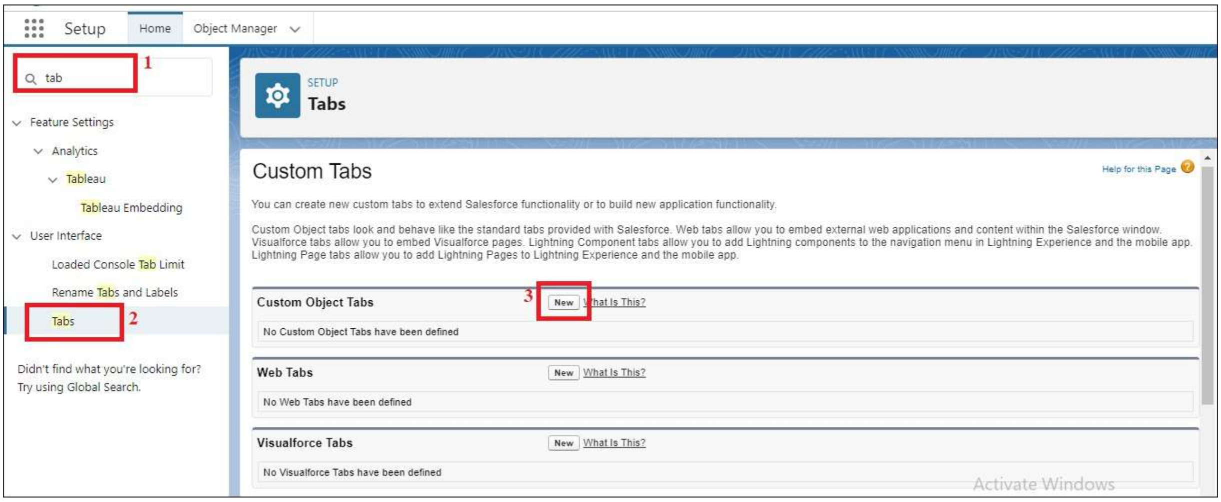
---

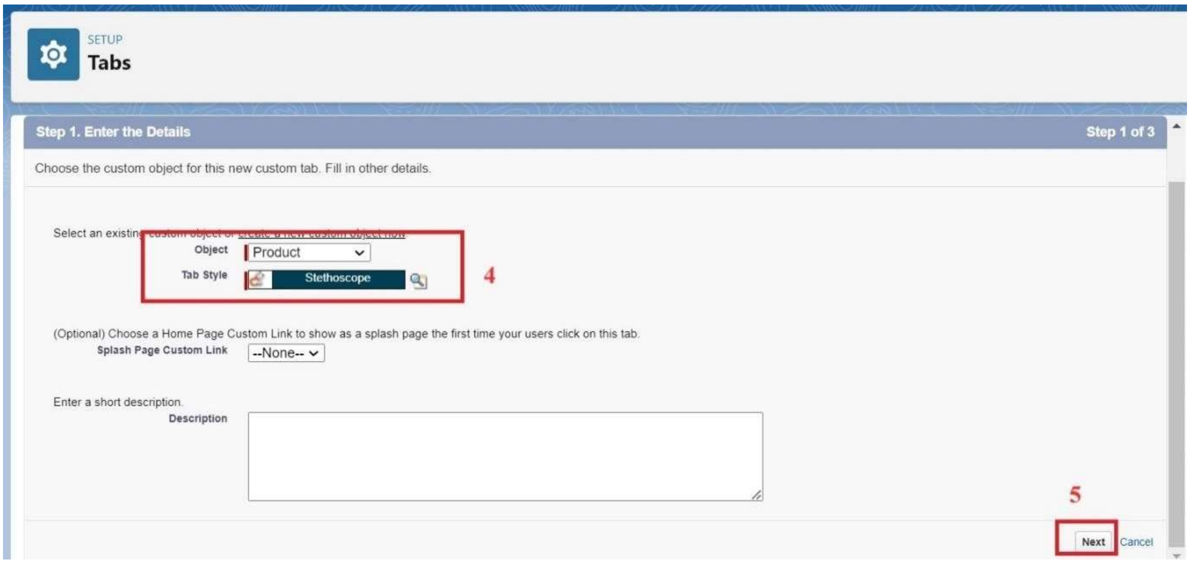
6. Tabs Created

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.

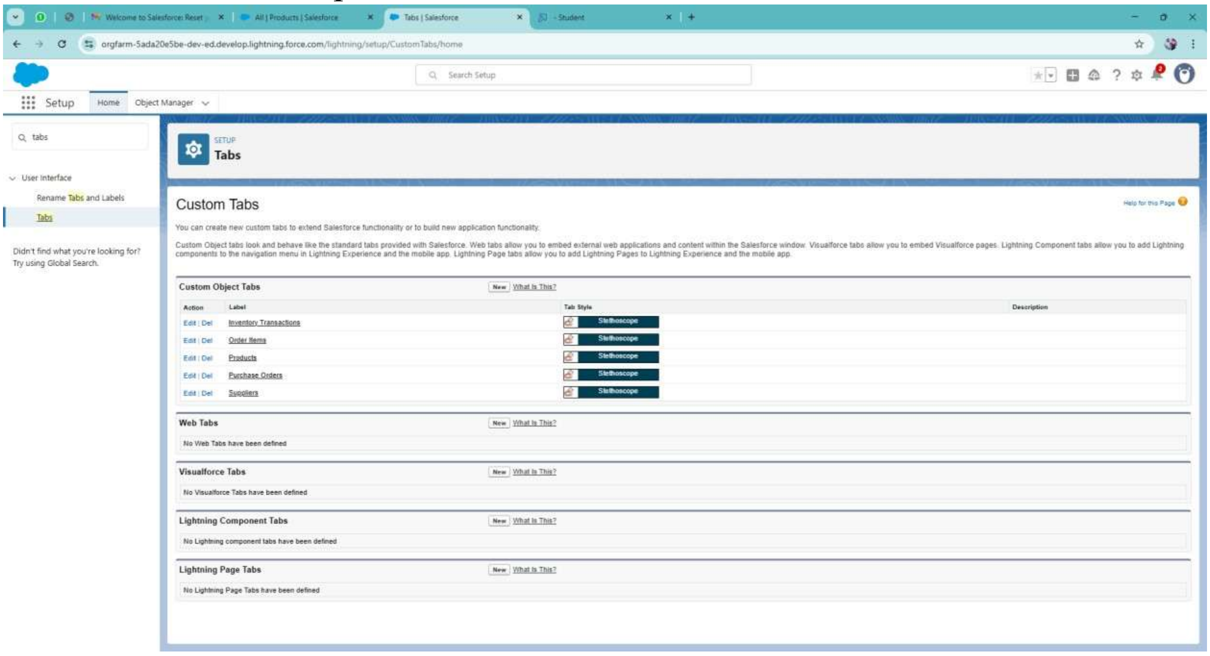
1. Go to the setup page >> type Tabs in Quick Find bar
2. Click on tabs
3. Click on New (under custom object tab).

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





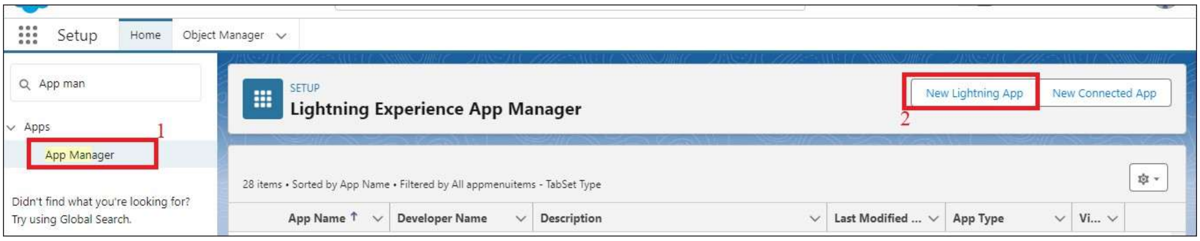
1. Now created the Tabs for the remaining Objects, they are “Purchase Order, Order Item, Inventory Transaction, Supplier”.
2. Followed the same steps as mentioned in above.

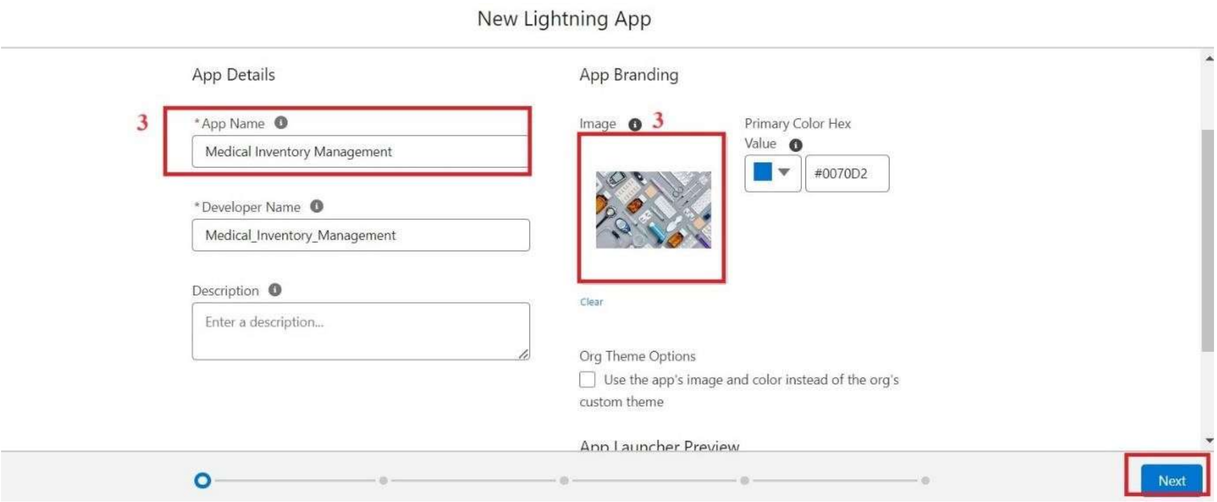


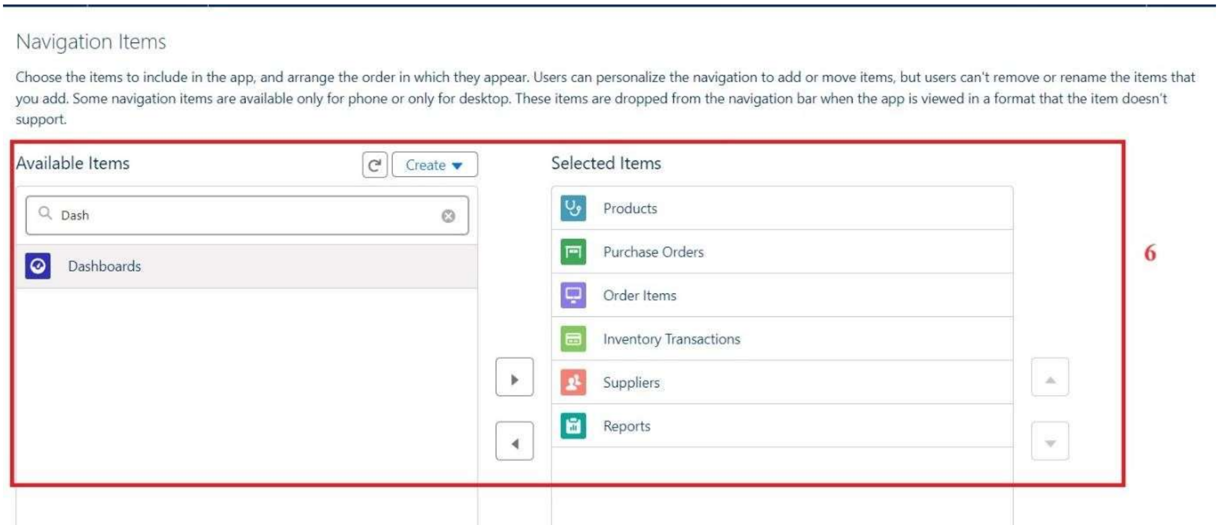
---

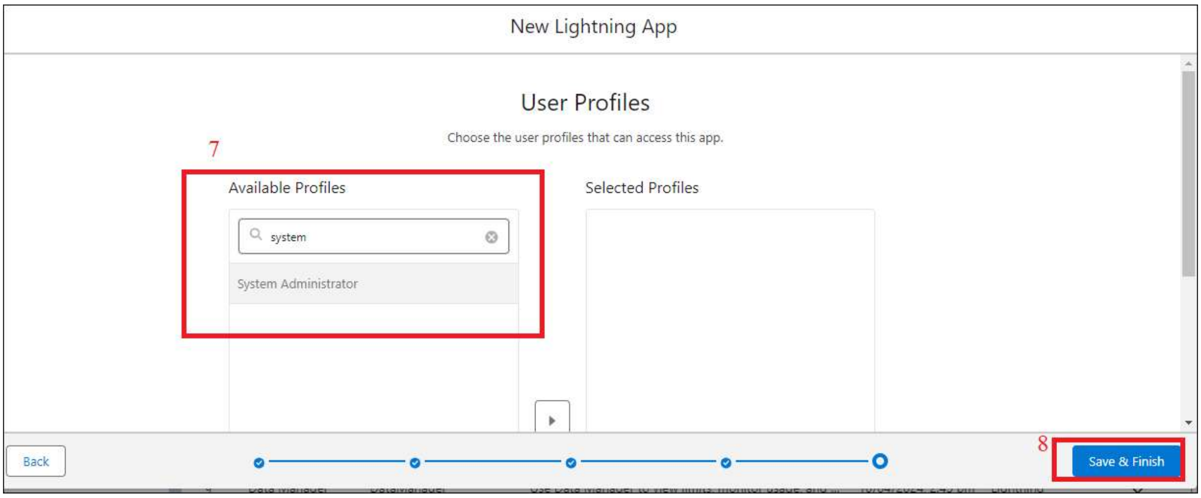
7. Created a Lightning App for Medical Inventory Management

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.


Setup
Home
Object Manager
App man
Apps
App Manager
1
Didn't find what you're looking for? Try using Global Search.
SETUP
Lightning Experience App Manager
New Lightning App
New Connected App
2
28 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type
App Name
Developer Name
Description
Last Modified ...
App Type
Vi...

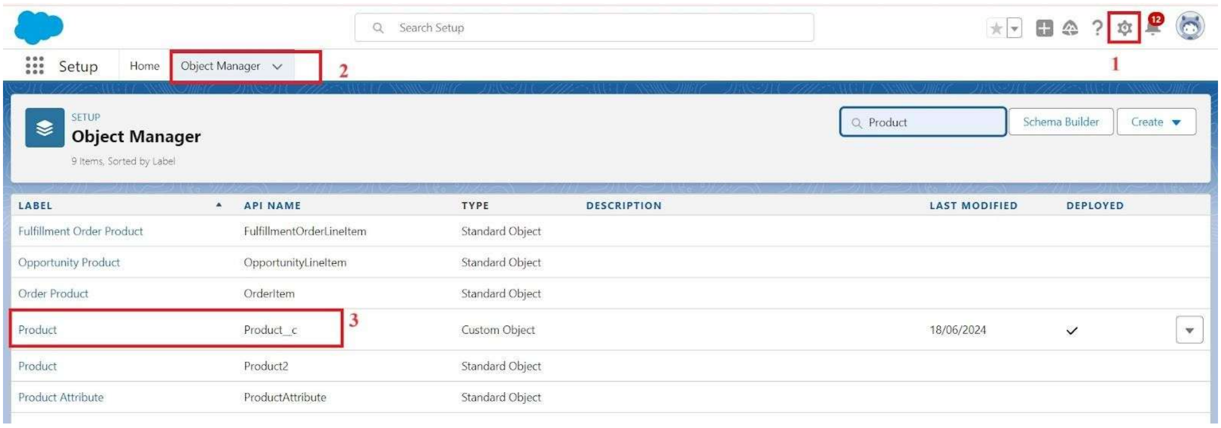

New Lightning App
App Details
3
*App Name
Medical Inventory Management
*Developer Name
Medical_Inventory_Management
Description
Enter a description...
App Branding
Image
3
Primary Color Hex Value
#0070D2
Clear
Org Theme Options
Use the app's image and color instead of the org's custom theme
App Launcher Preview
Next

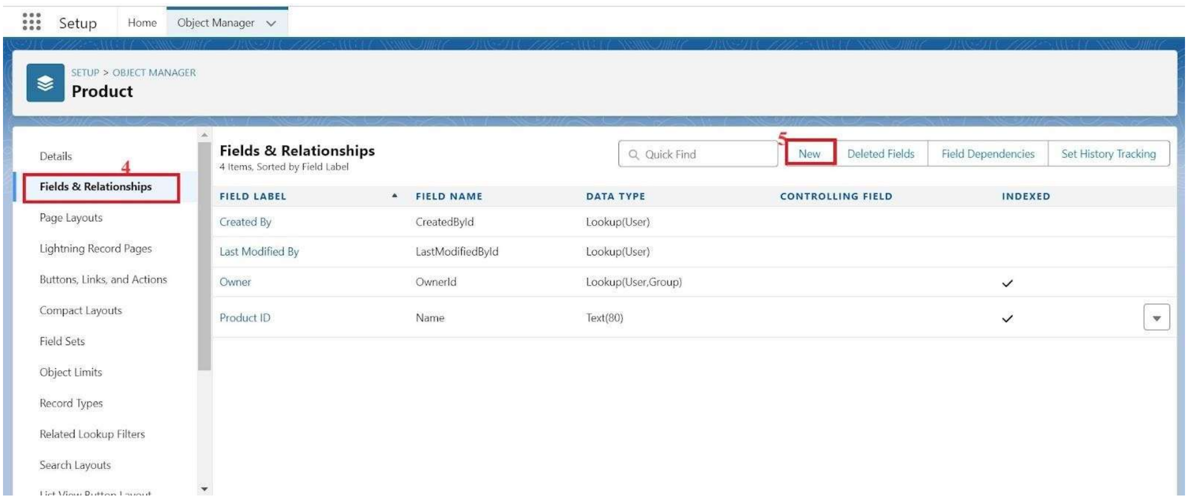

Navigation Items
Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.
Available Items
Create
Dash
Dashboards
Selected Items
Products
Purchase Orders
Order Items
Inventory Transactions
Suppliers
Reports
6

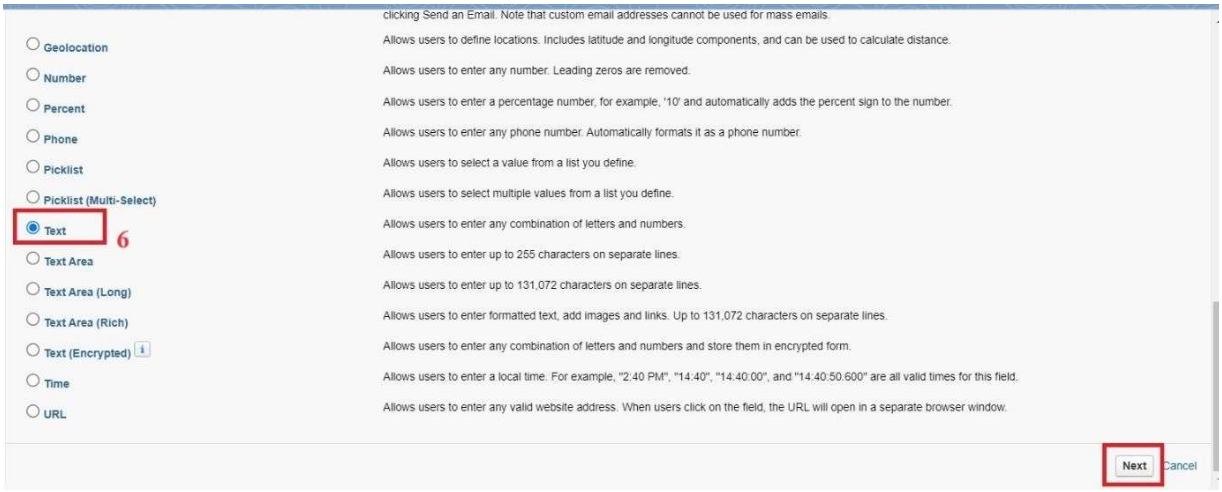

New Lightning App
User Profiles
Choose the user profiles that can access this app.
7
Available Profiles
system
System Administrator
Selected Profiles
Back
8
Save & Finish

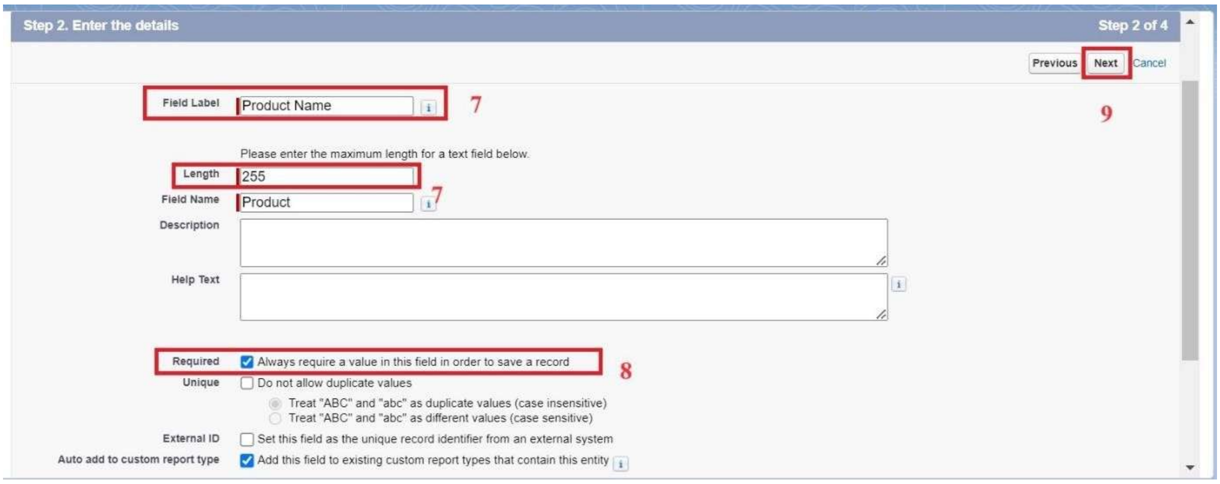
8. Created a Text Field in Product Object

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.









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

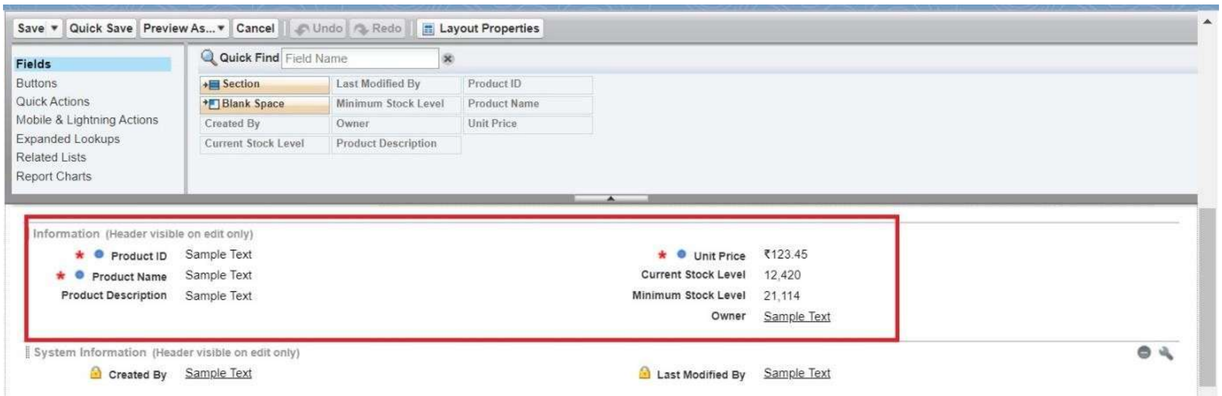
Purchase Order	Purchase Order ID(Standard)	Text
	Supplier ID	Lookup(Supplier)
	Order Date	Date
	Expected Delivery Date	Date
	Actual Delivery Date	Date
	Order Count	Roll-Up Summary (COUNT Order Item)
	Total Order Cost	Currency(16, 2)
Order Item	Order Item ID(Standard)	Text
	Product ID	Lookup(Product)
	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

---

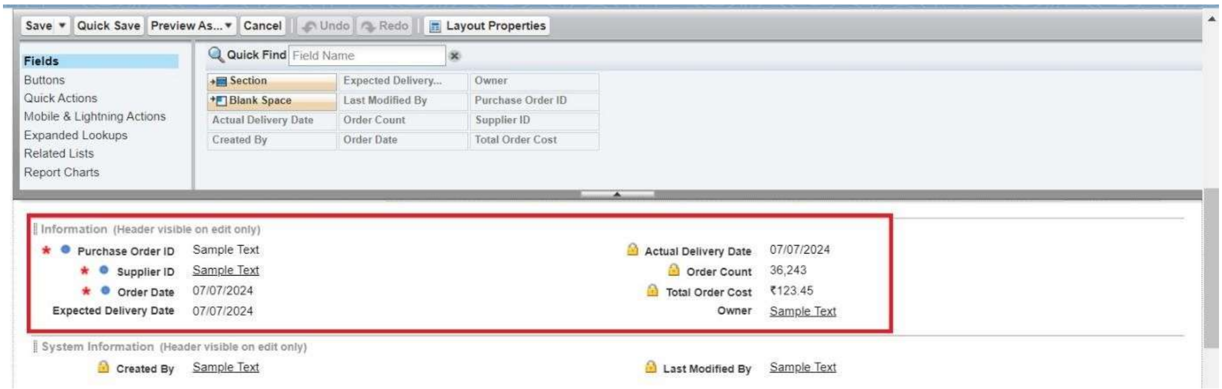
9. Edited a Page Layout in Product Object

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.

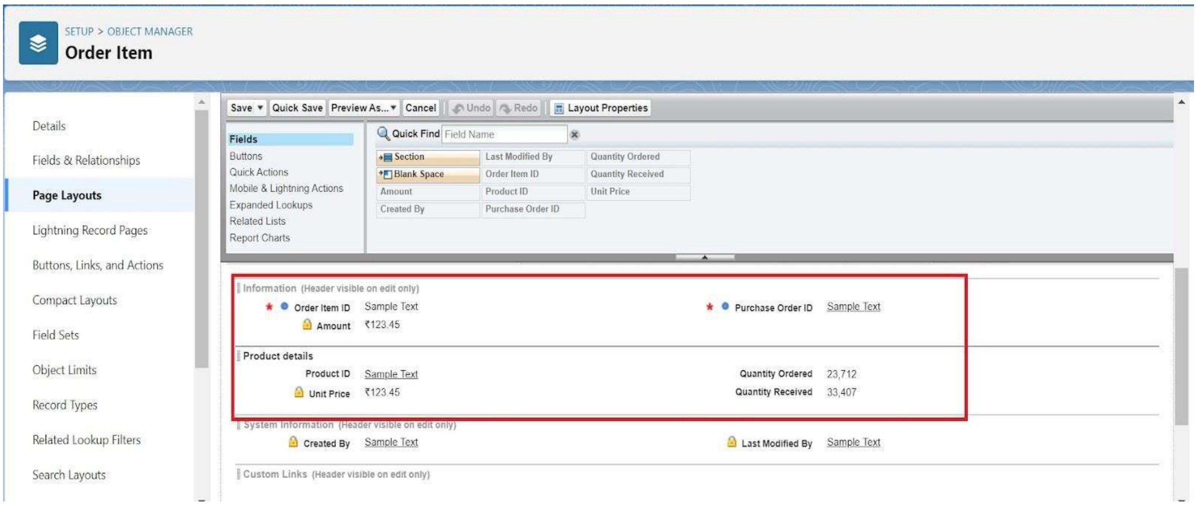


4. Click on Save.

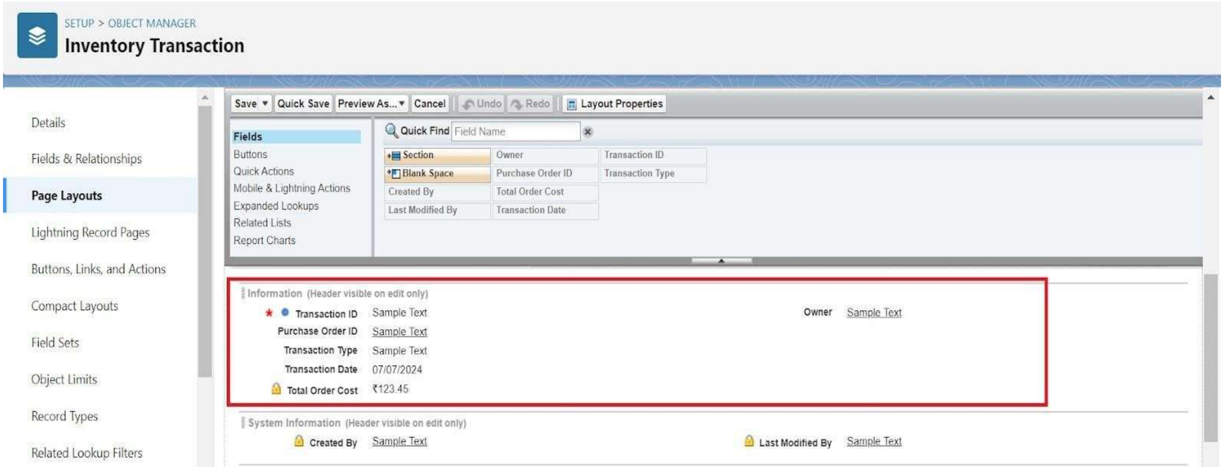
9.1 Edited a Page Layout in Purchase Order Object



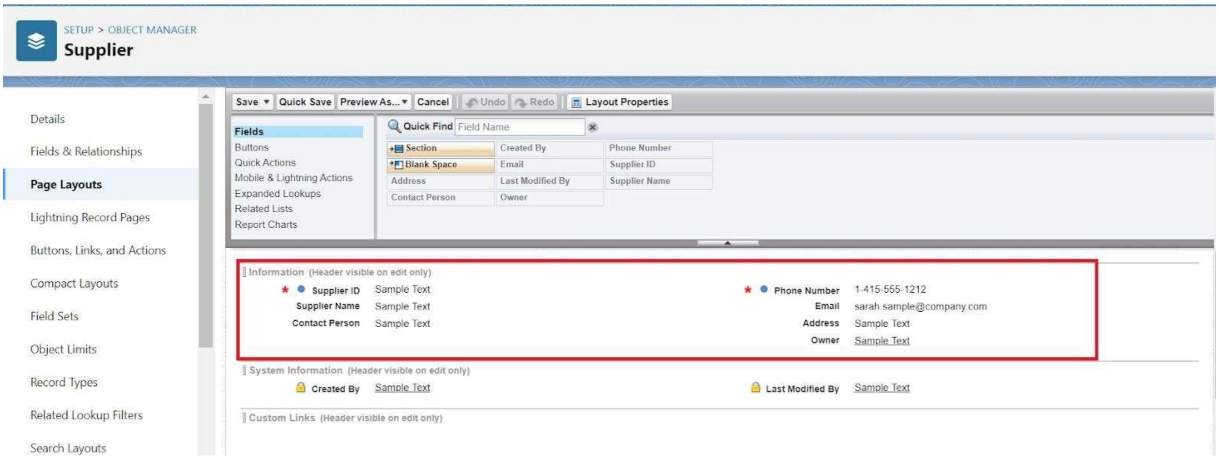
9.2 Edited a Page Layout in Order Item Object



9.3 Edited a Page Layout in Inventory Transaction Object



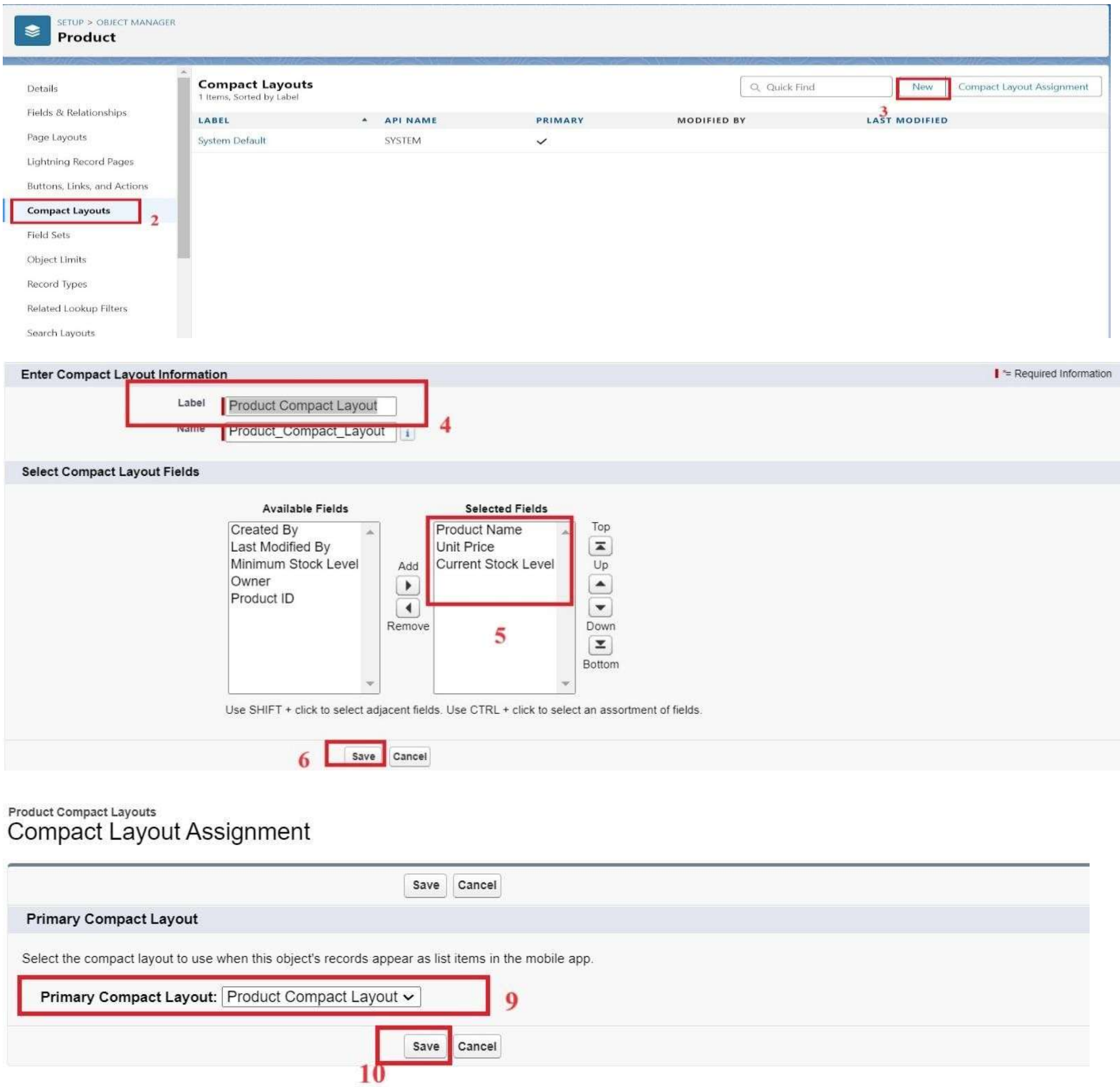
9.4 Edited a Page Layout in Supplier Object



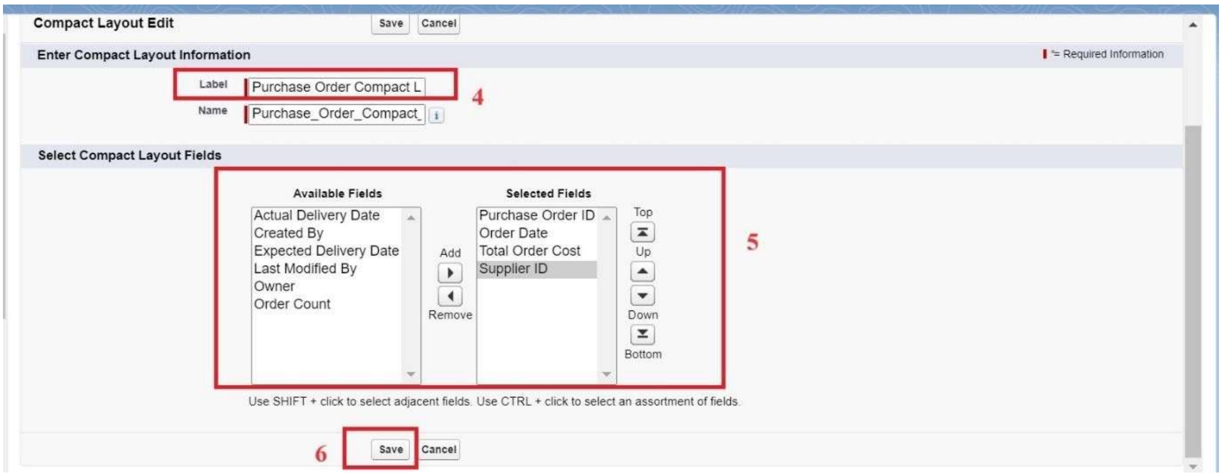
10. Created a Compact Layout to a Product Object

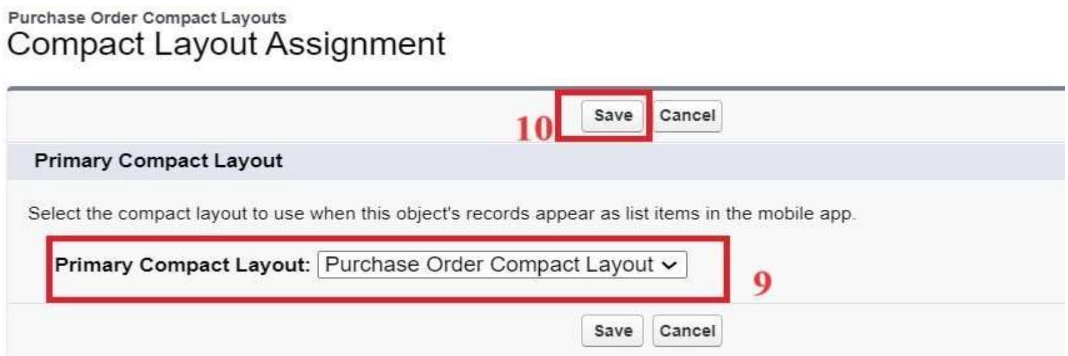
1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object
2. Click on Compact Layouts in the sidebar .
3. Click on New.

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.



10.1 Created a Compact Layout to a Purchase Order Object

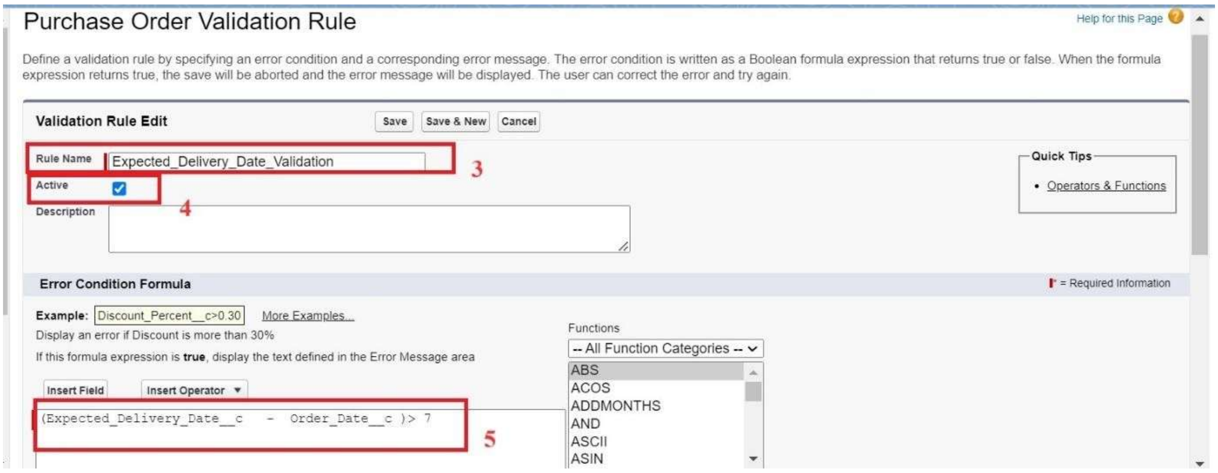




---

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

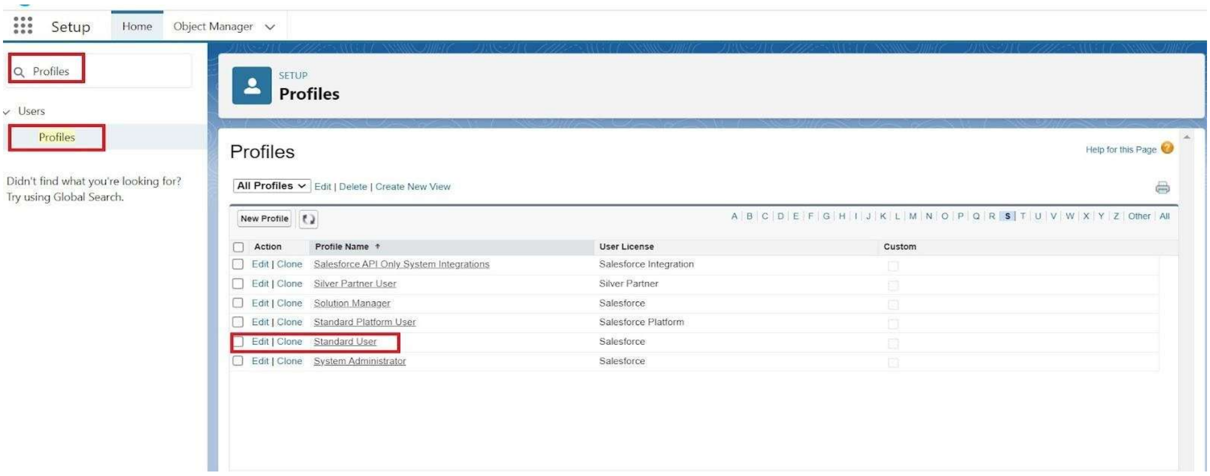


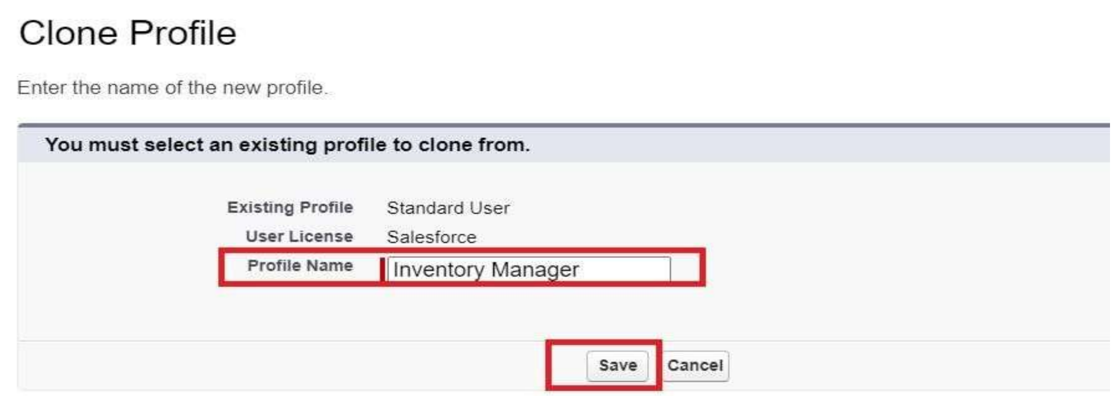
6. Enter the Error Message as “The Expected Delivery Date should not exceed 7 days.”.

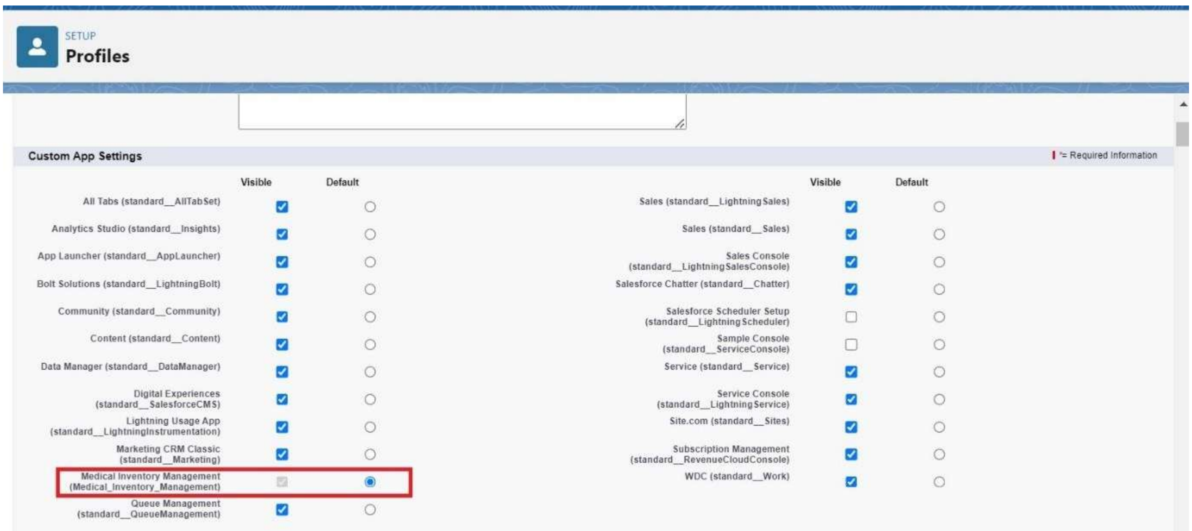
7. Select the Error location as Top of Page

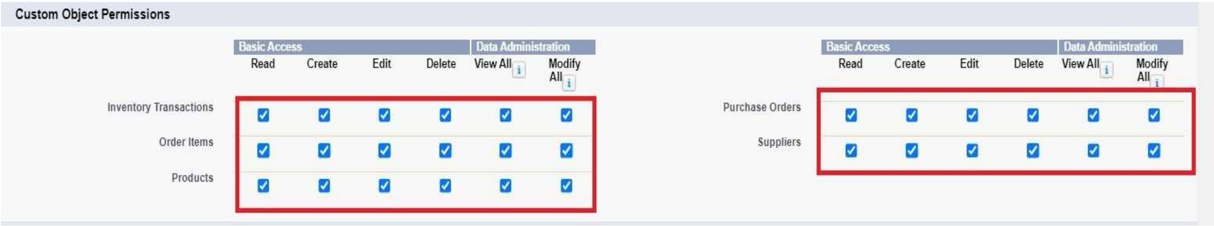
8. Click Save.

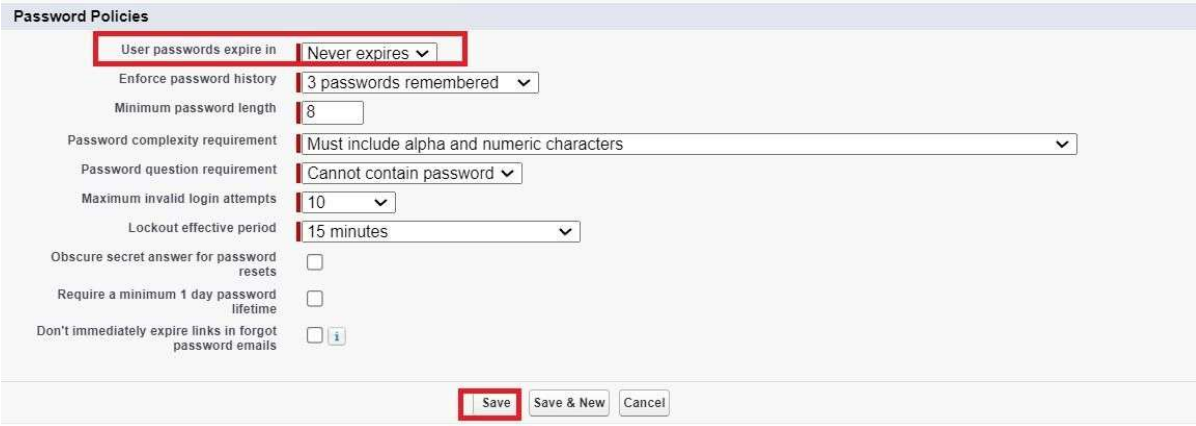
12. Created an Inventory Manager Profile



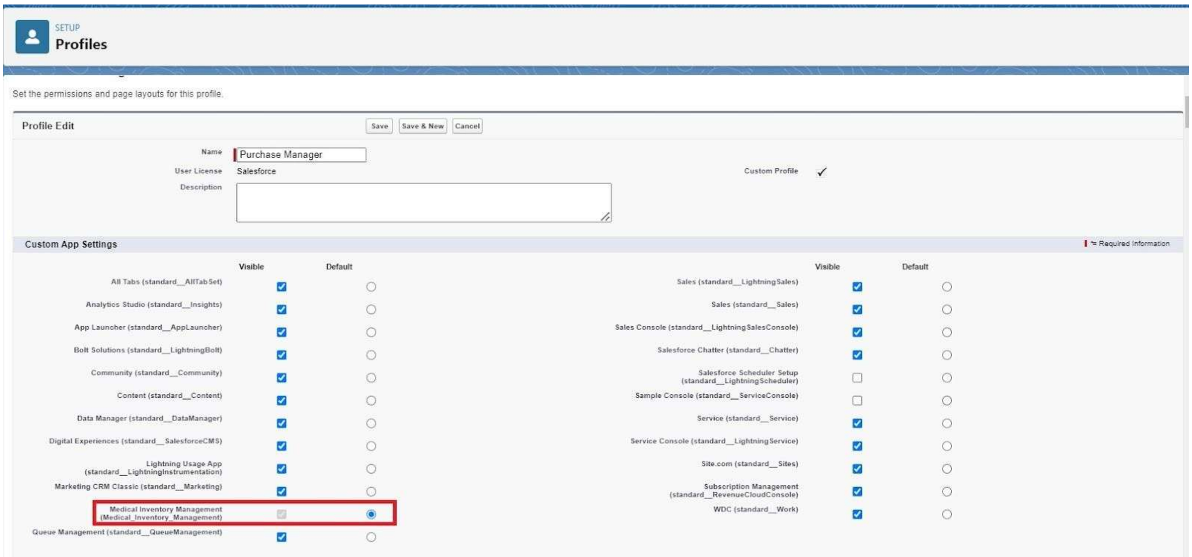


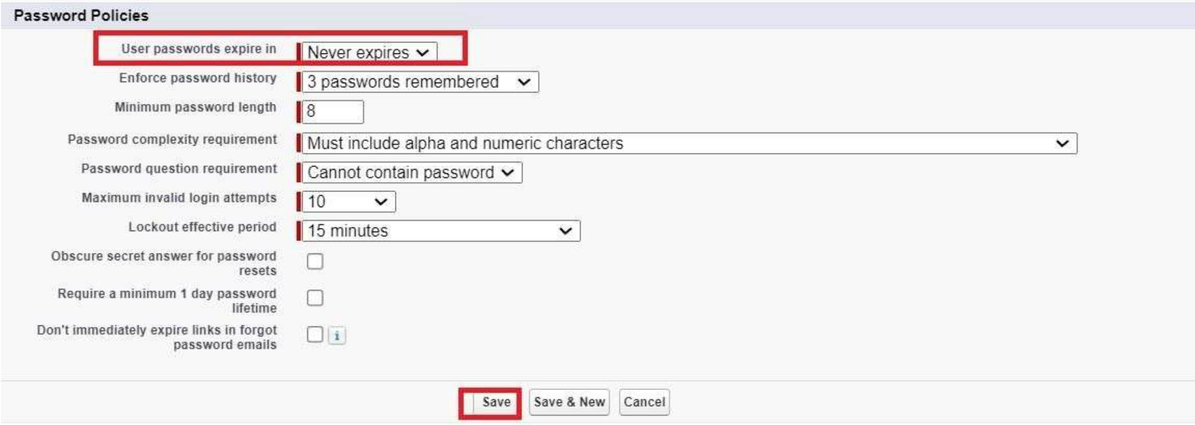




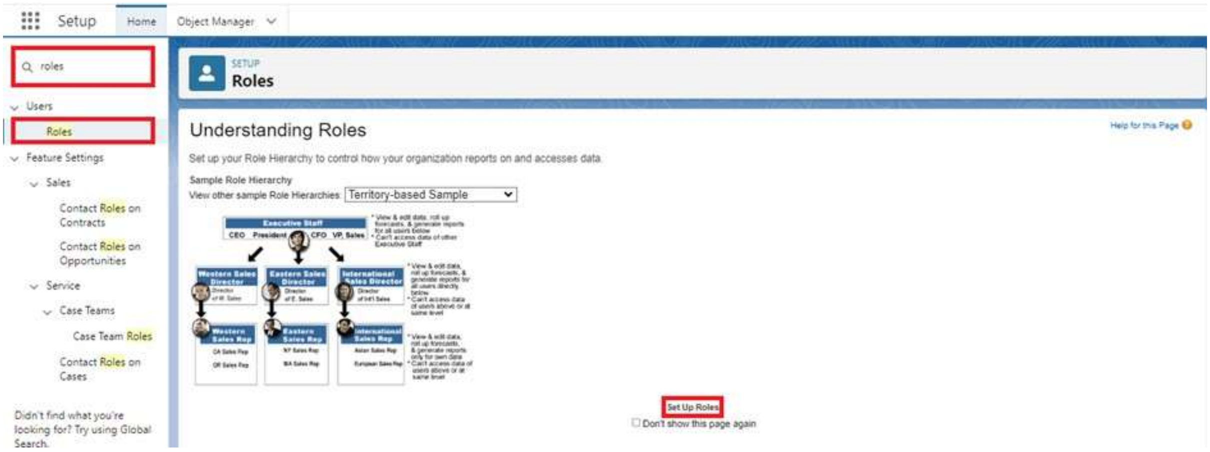


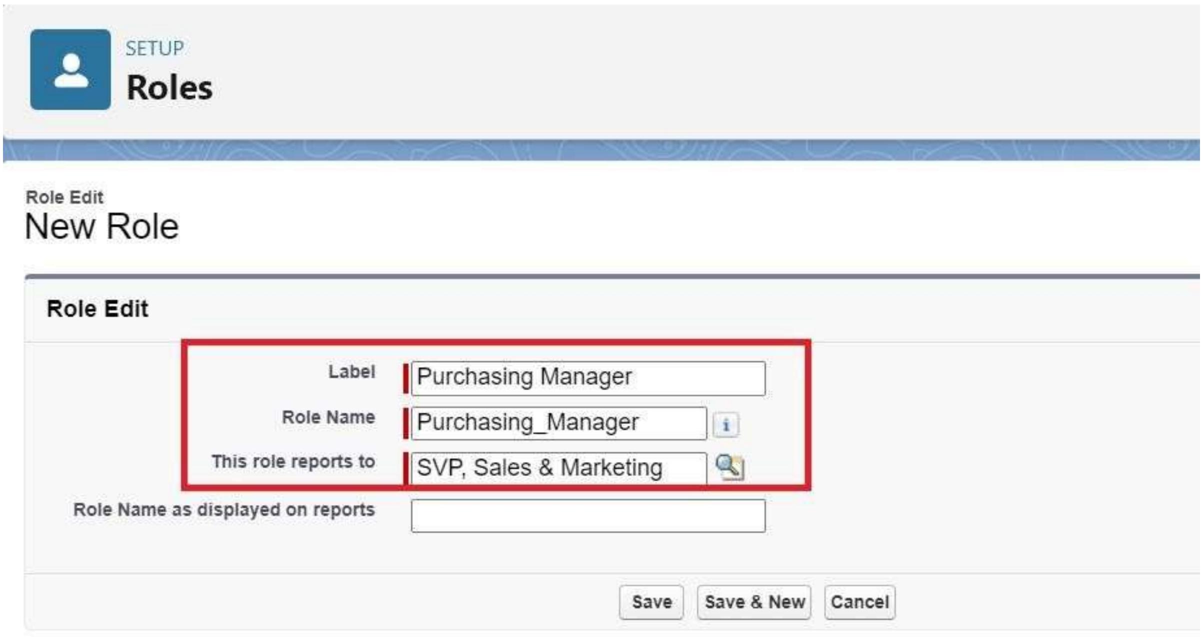
12.1 Created an Purchase Manager Profile



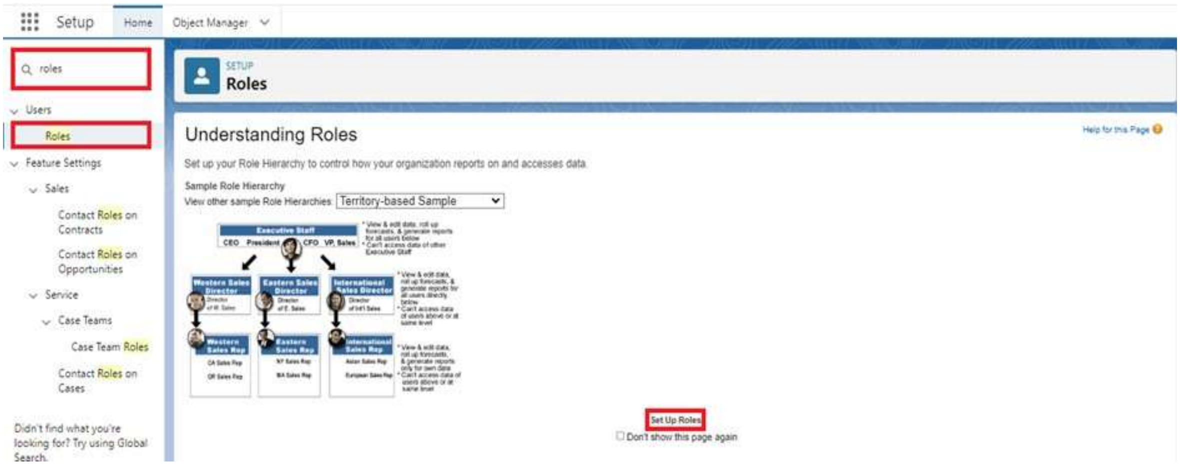


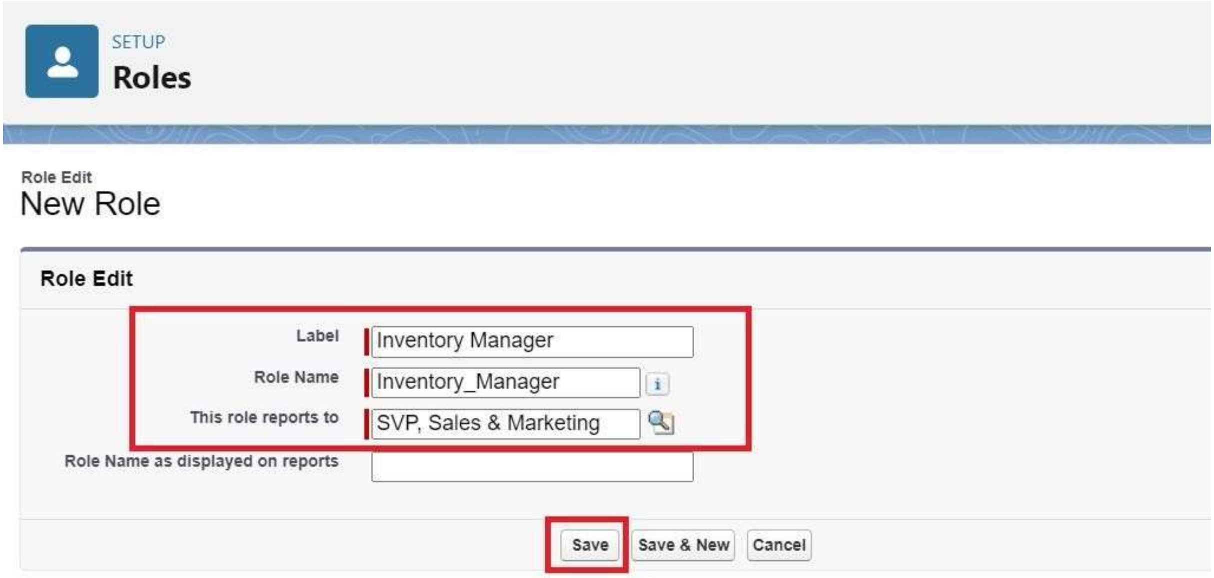
13. Created a Purchasing Manager Role.





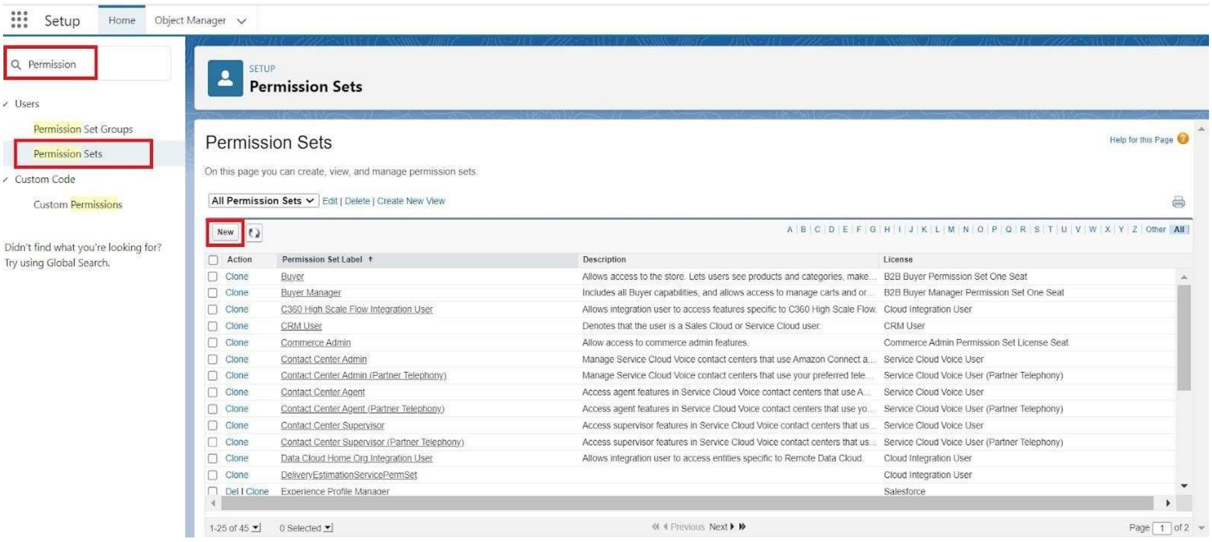
13.1 Created a Inventory Manager Role.



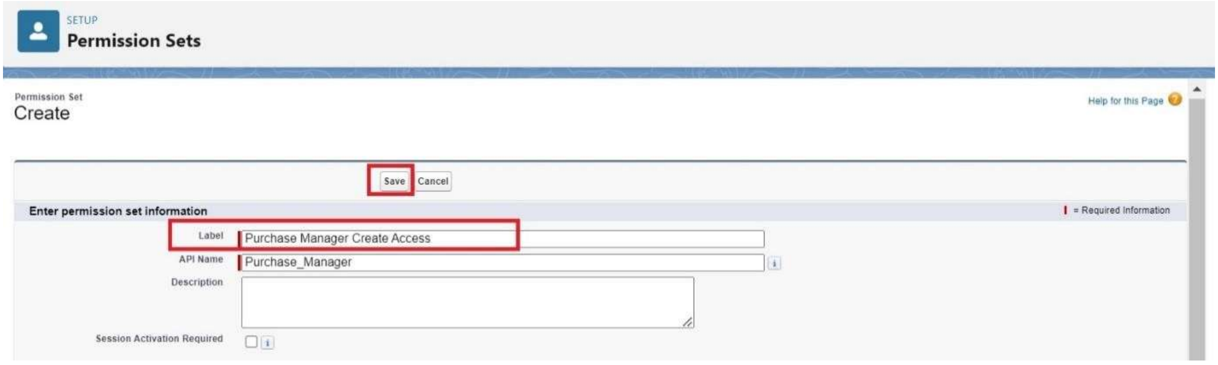


14. Created a Permission Set.

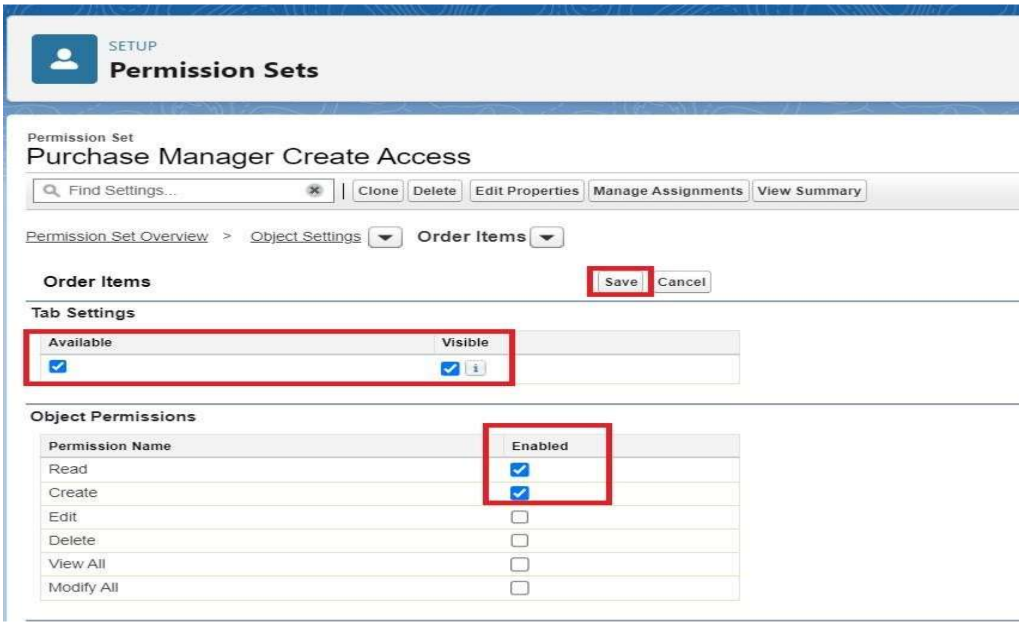
1. Go to setup >> type Permission in quick find box >> Select Permission Set >> click on New.



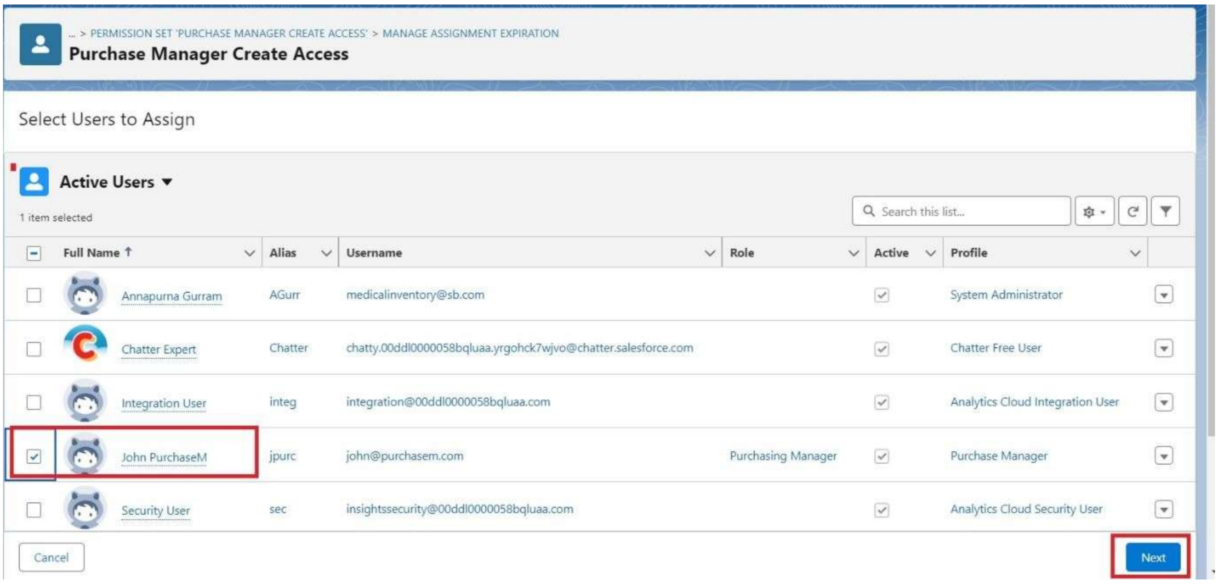
2. Enter Label as Purchase Manager Create Access >> Click on Save.



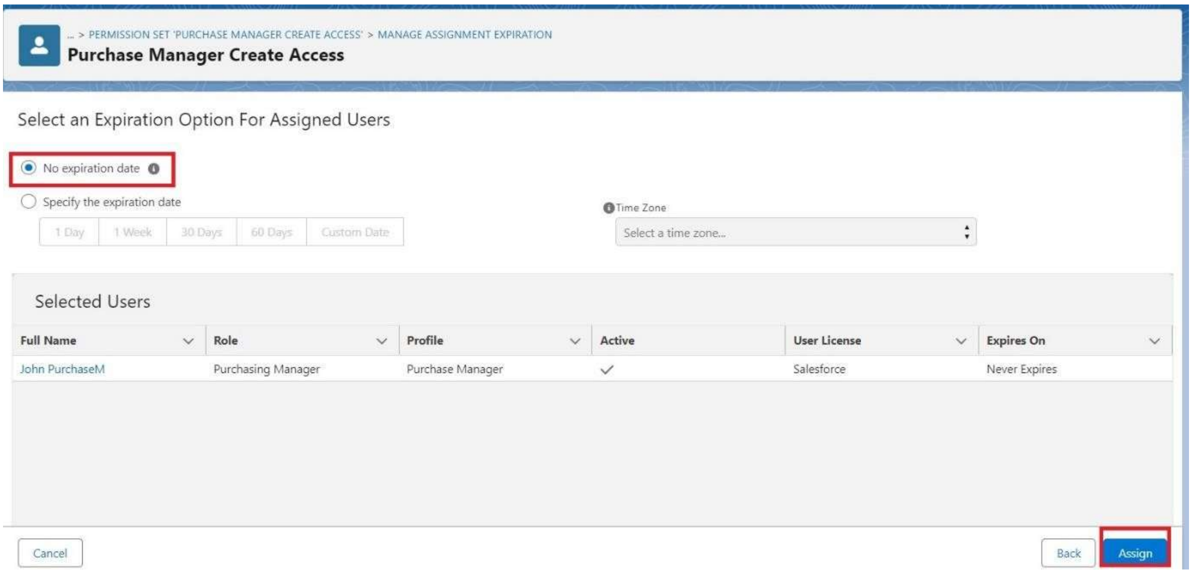
3. From Object Settings >> Select Order Item >> Enable for both Tab Available and Visible >> Enable Read and Create in Object Permissions >> Click on Save.



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

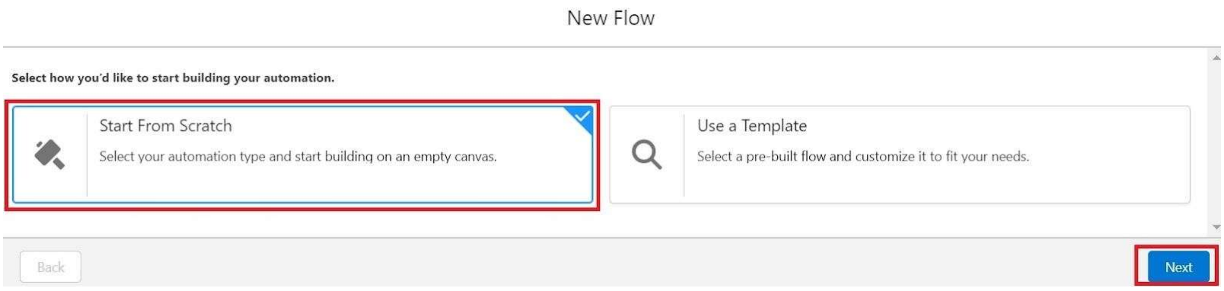


5. Select No Expiration date >> Click on Assign.

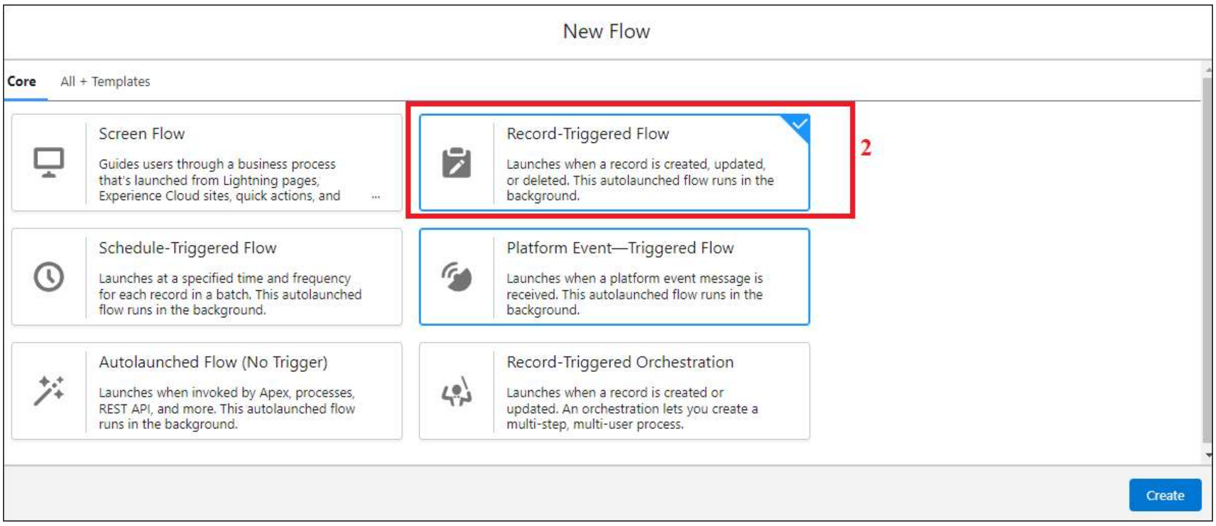


15. Created Flow to update the Actual Delivery Date.

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

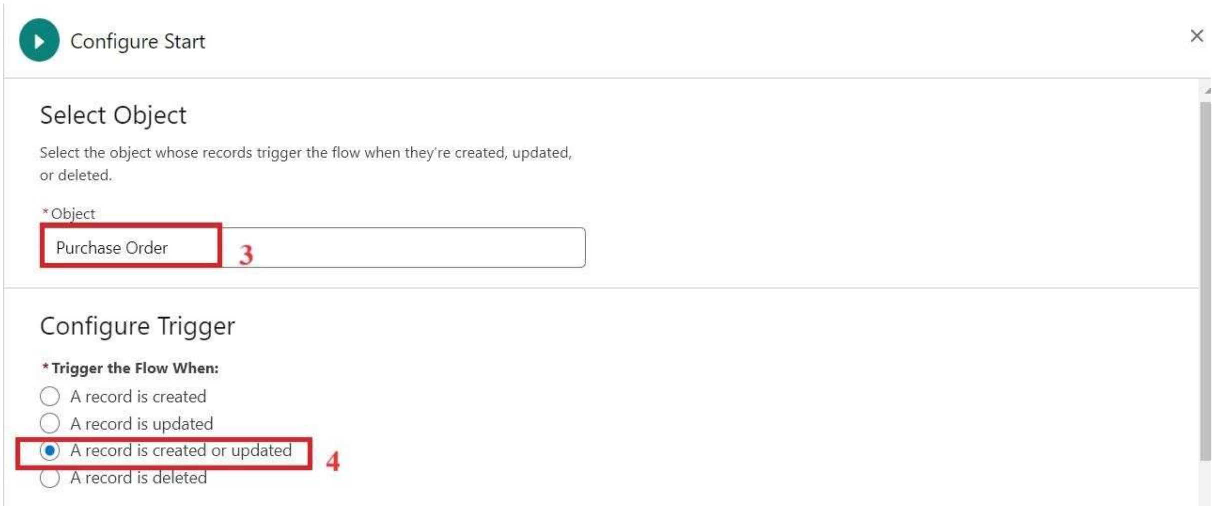


2. Select the record Triggered flow.Click on create.



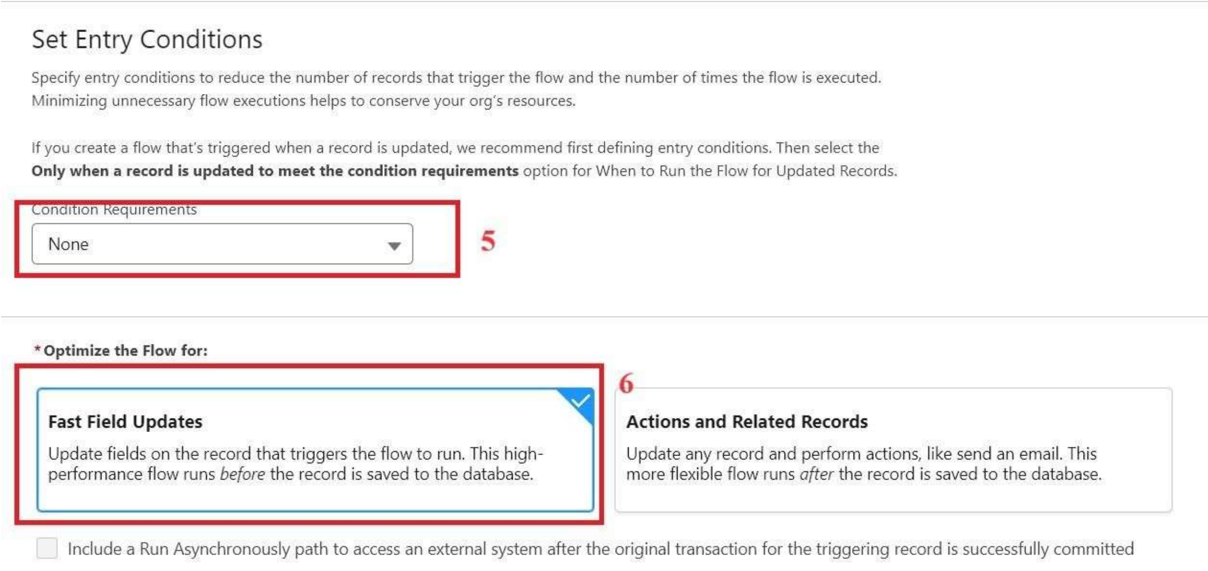
3. Under Object select "Purchase Order"

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.

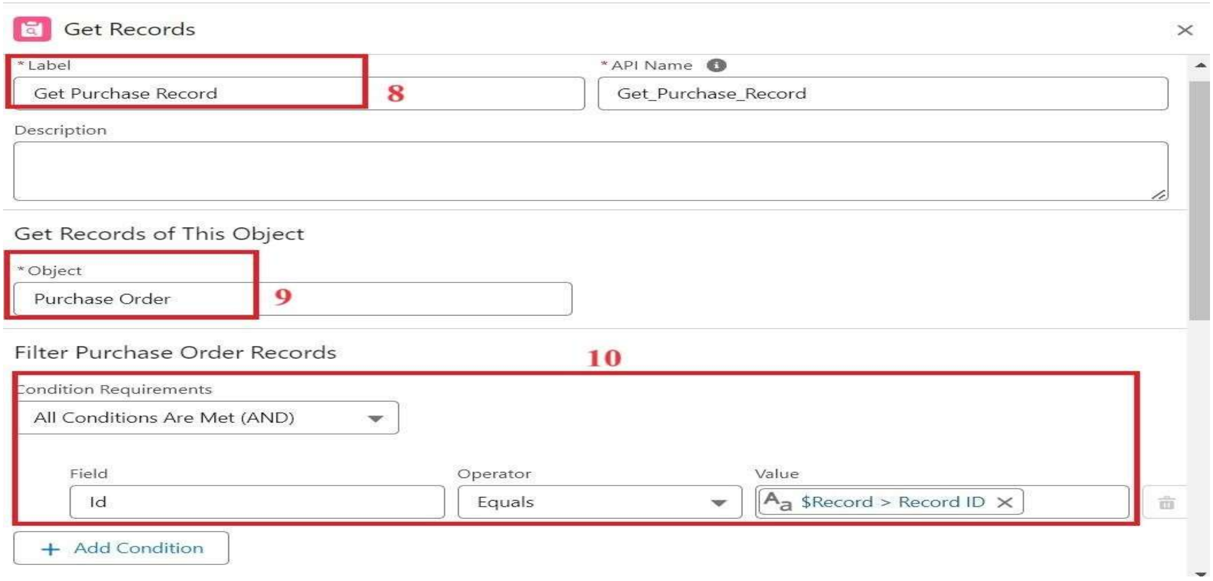
10. For Condition Requirements , select All Conditions are Met(AND)

For the first condition select as follows:

Field: Id

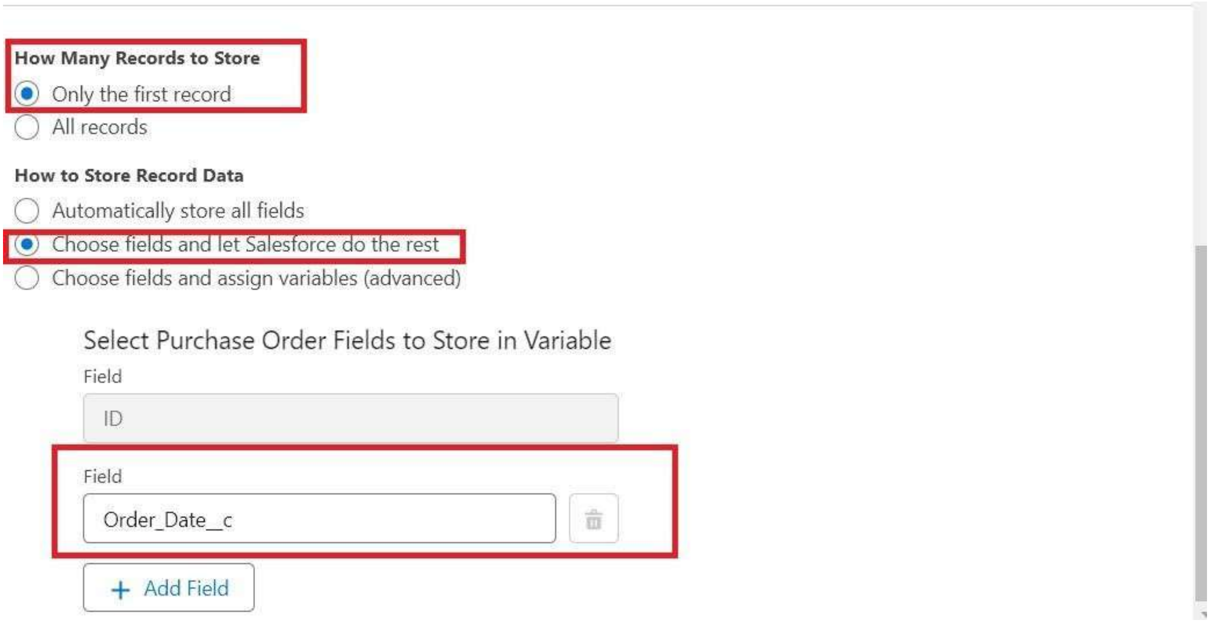
Operator: Equals

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}

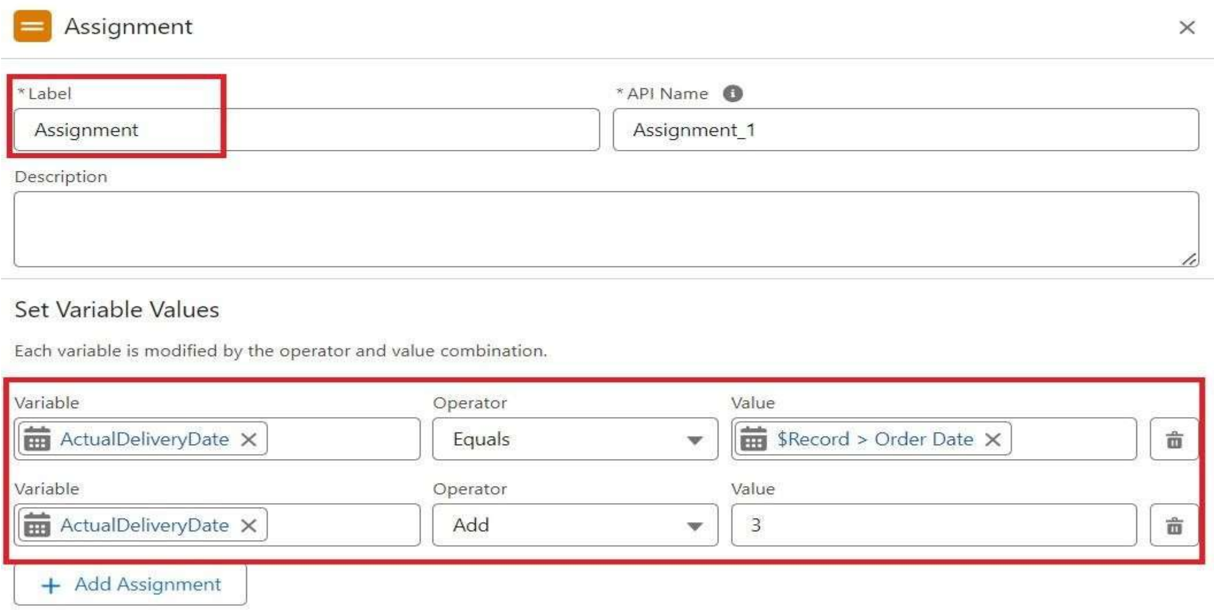
Operator : Equals

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".

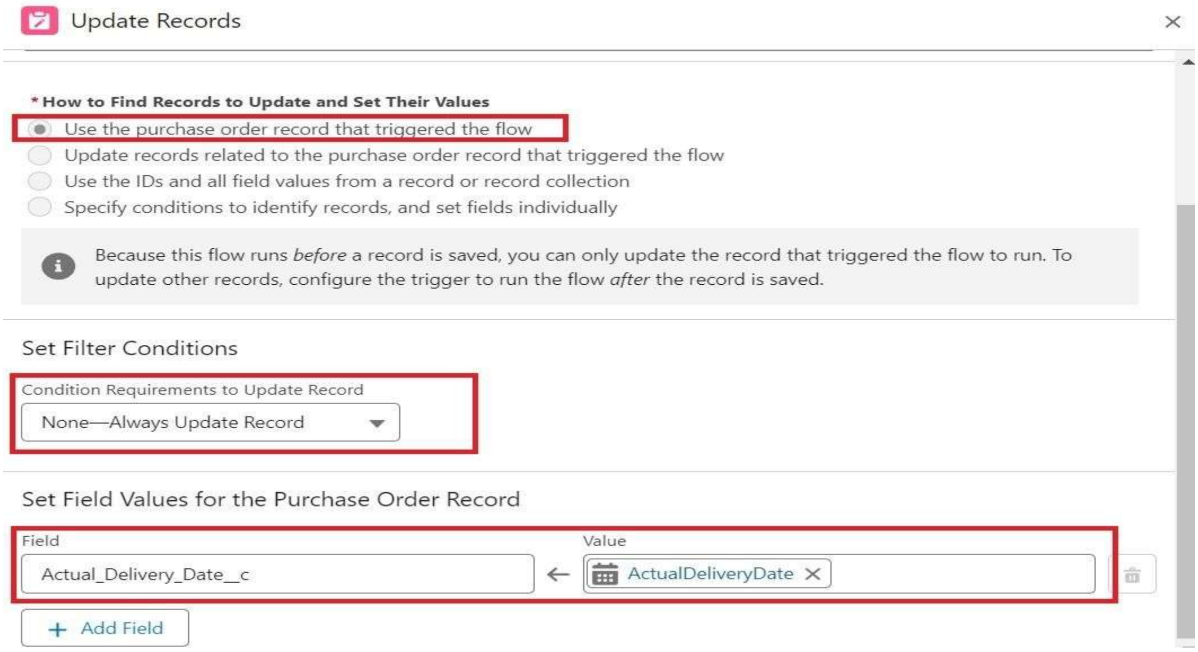
21. How to Find Records to Update and Set Their Values : Use the Purchase Order record that triggered the flow

22. Set Filter Conditions : None -Always Update Record

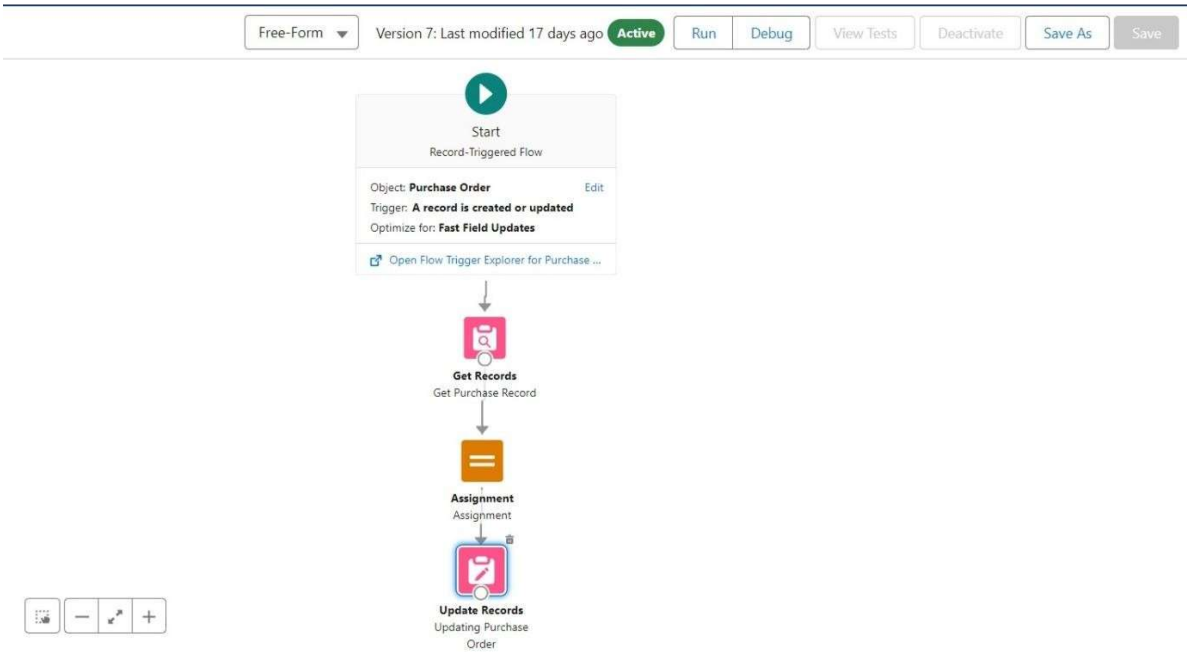
23. Set Field Values for the Trip Record as

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.

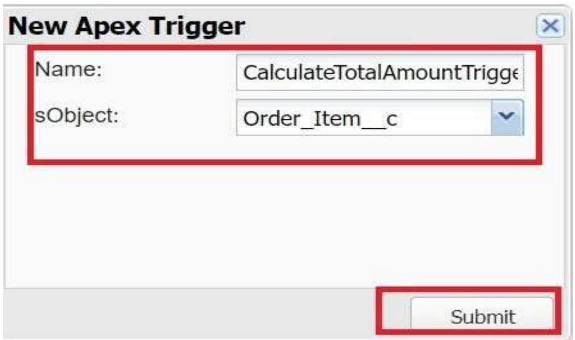
ii) Click on Developer Console: Click on the "Developer Console" option from the Setup menu. This will open the Developer Console in a new browser tab or window.

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);
            }
```

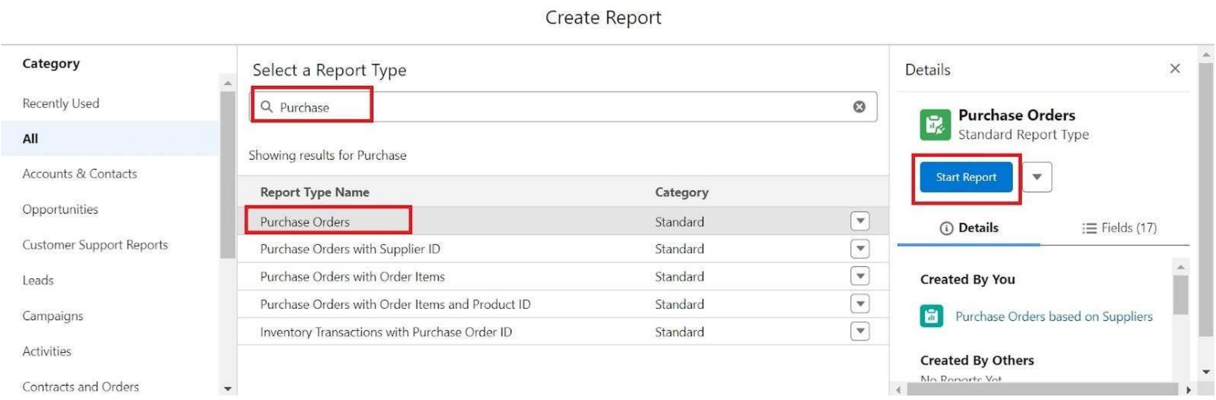
```
            // Prepare Purchase Order records for update
            List<Purchase_Order__c> purchaseToUpdate = new
List<Purchase_Order__c>();
            for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {
                Purchase_Order__c purchaseOrder = new Purchase_Order__c(Id =
purchaseOrderId,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

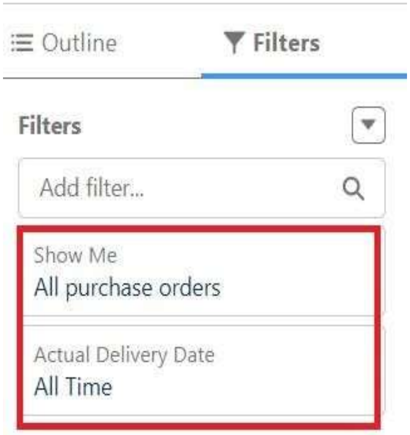
---

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).

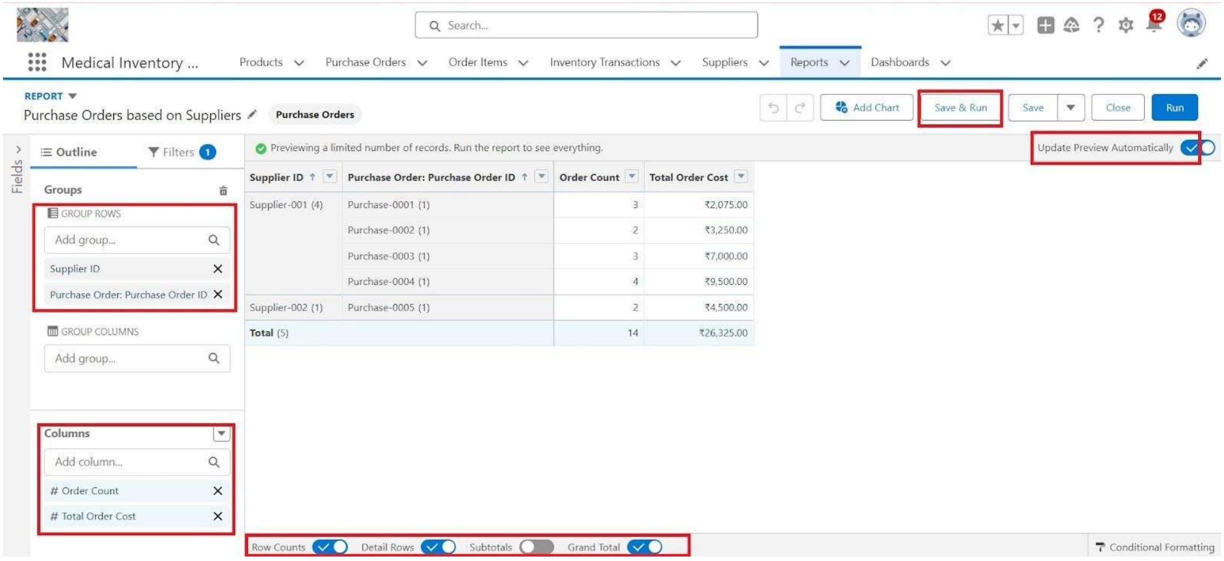
8. Click save and run

9. Give report name – Purchase Orders based on Suppliers.

10. Click Save

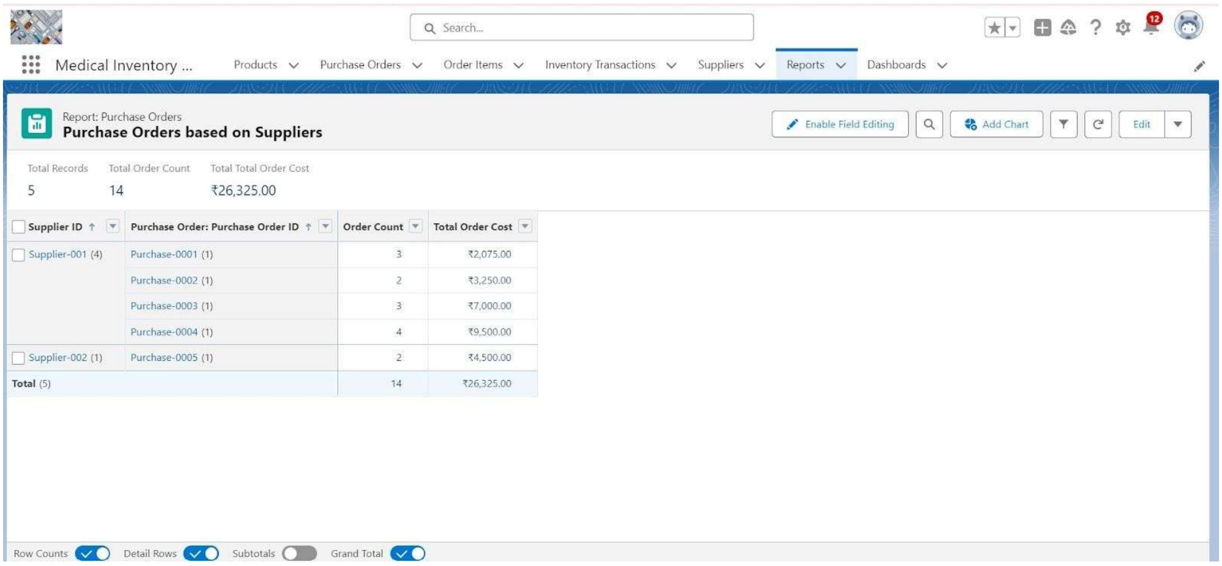
NOTE: In this report you can see your all record of the object you selected for reporting

(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.



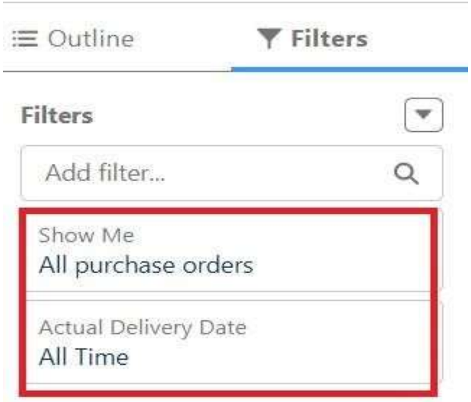
17.1 Created a Complete Purchase Details Report

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

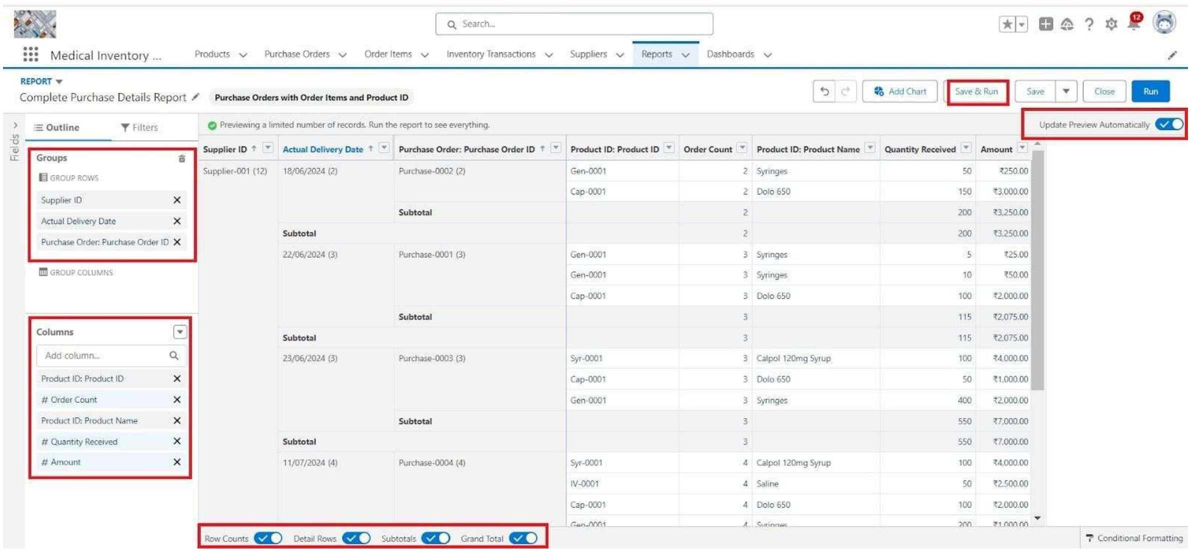


7. Customize your report, in group rows select – Supplier ID, Actual Delivery Date, Purchase Order: Purchase Order ID, for columns Product ID : Product ID, Product ID : Product Name, Order Count, Quantity Received, Amount (In this way we are making a Summary Report).

8. Click save and run

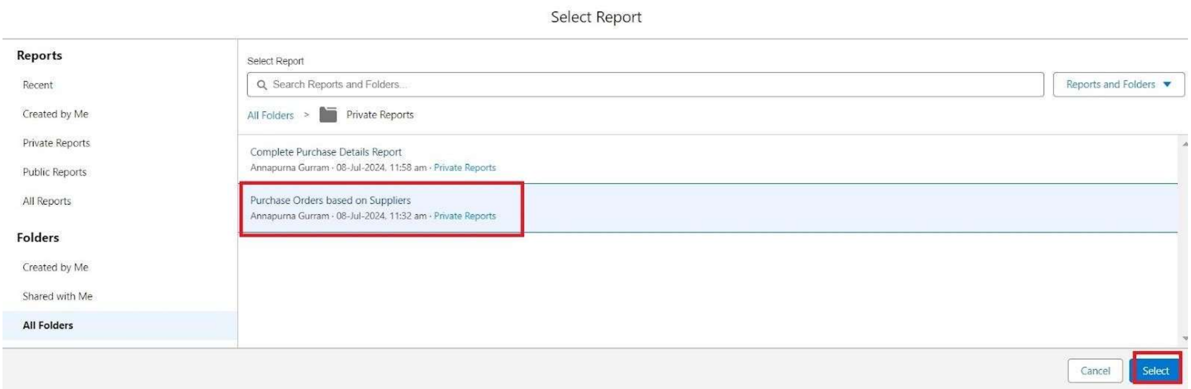
9. Give report name – Complete Purchase Details Report

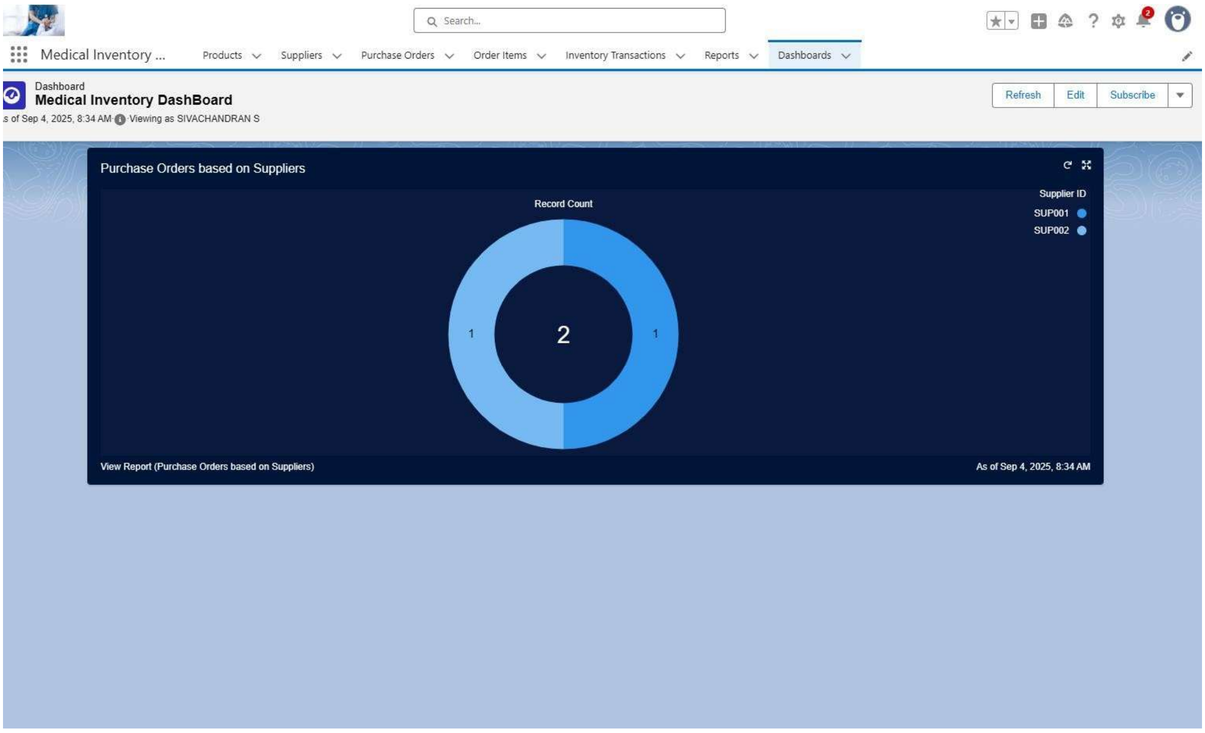
10. Click Save



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.

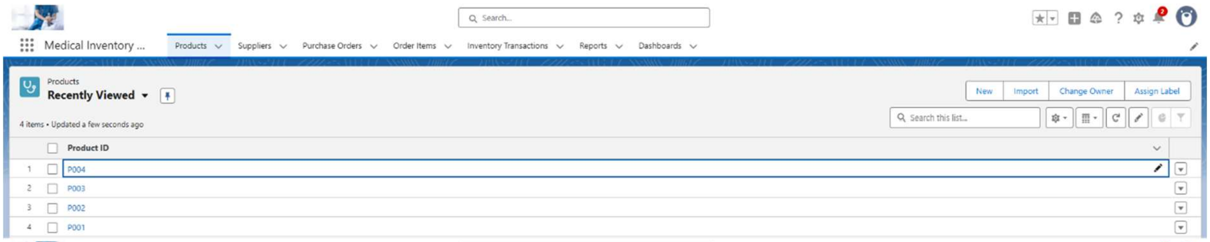
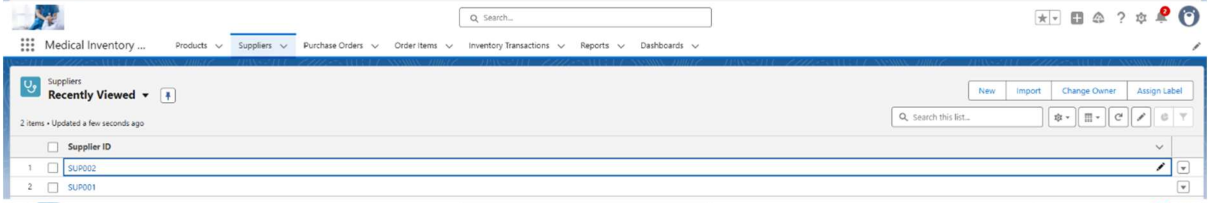
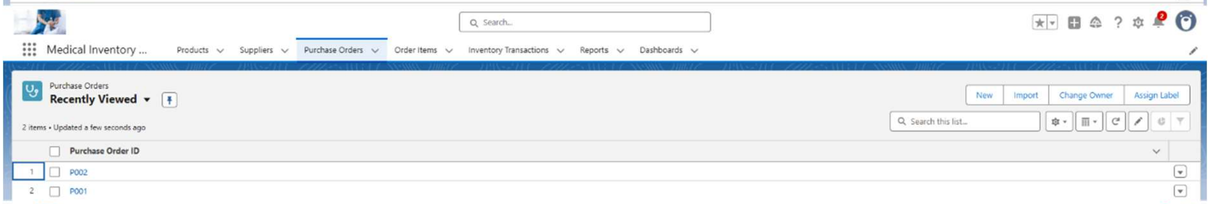
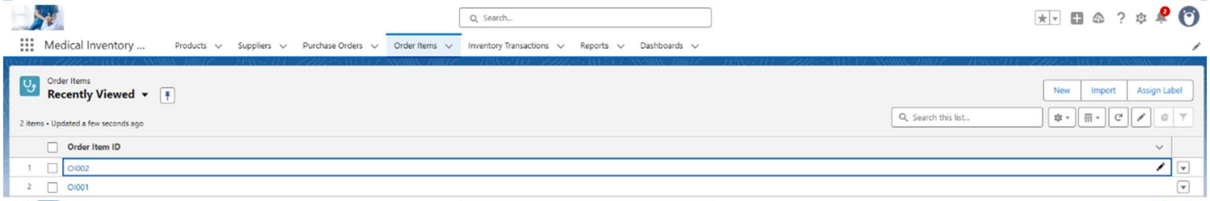
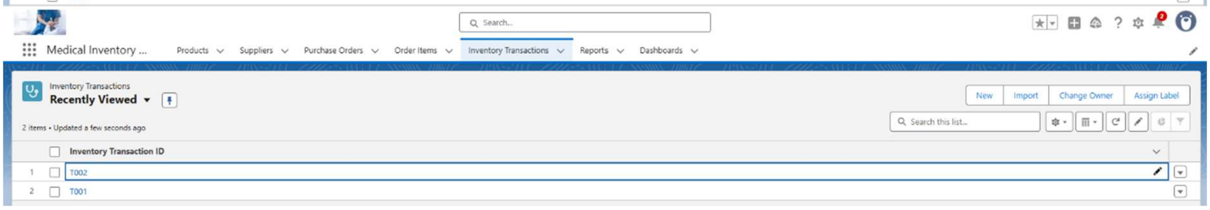


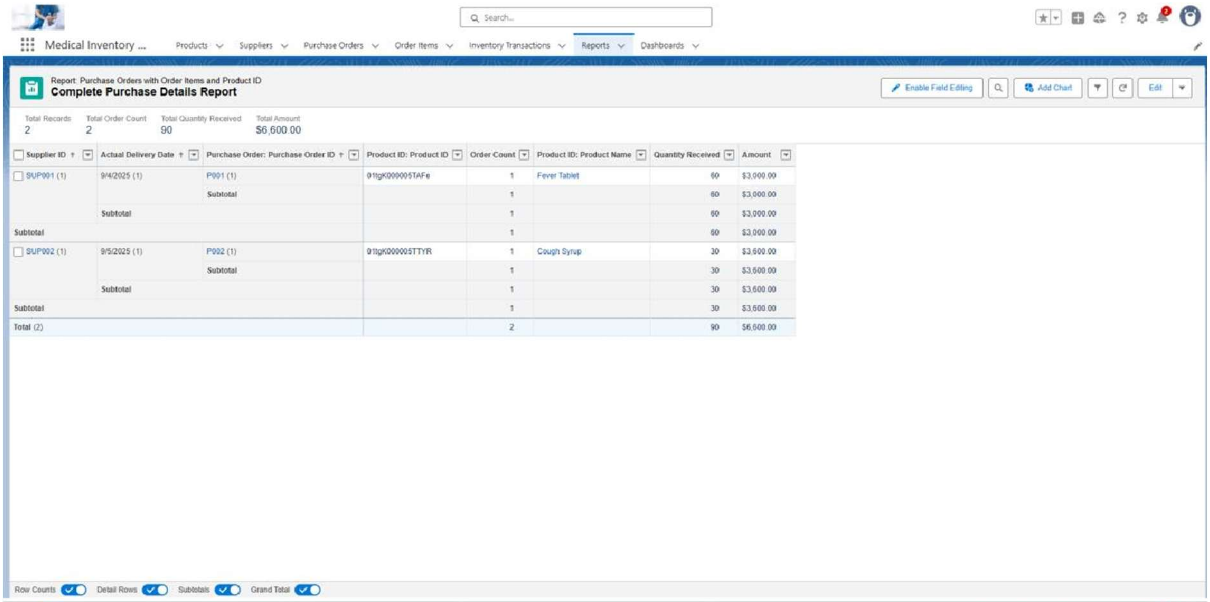


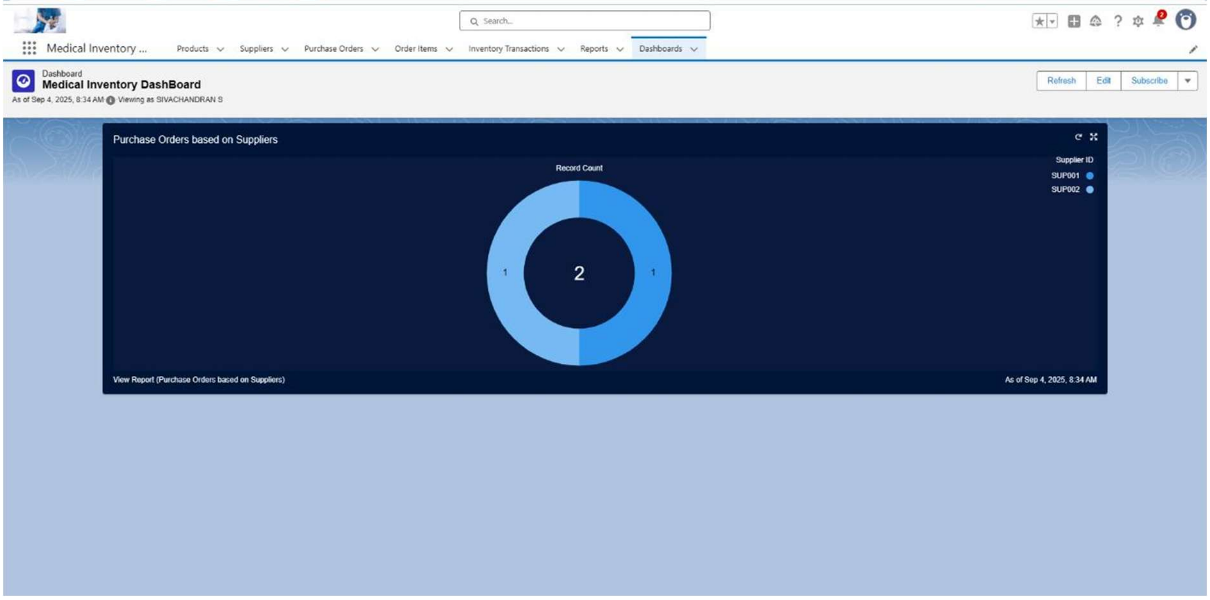
19. Testing Approach

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.


Search...
Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Products
Recently Viewed
New
Import
Change Owner
Assign Label
Search this list...
4 items • Updated a few seconds ago
Product ID
1 P004
2 P003
3 P002
4 P001

Search...
Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Suppliers
Recently Viewed
New
Import
Change Owner
Assign Label
Search this list...
2 items • Updated a few seconds ago
Supplier ID
1 SUP002
2 SUP001

Search...
Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Purchase Orders
Recently Viewed
New
Import
Change Owner
Assign Label
Search this list...
2 items • Updated a few seconds ago
Purchase Order ID
1 P002
2 P001

Search...
Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Order Items
Recently Viewed
New
Import
Assign Label
Search this list...
2 items • Updated a few seconds ago
Order Item ID
1 OI002
2 OI001

Search...
Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Inventory Transactions
Recently Viewed
New
Import
Change Owner
Assign Label
Search this list...
2 items • Updated a few seconds ago
Inventory Transaction ID
1 T002
2 T001


Search...
Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Report: Purchase Orders with Order Items and Product ID
Complete Purchase Details Report
Enable Field Editing
Add Chart
Edit
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
1
Cough Syrup
30
$3,600.00
Subtotal
Total (2)
2
90
$6,600.00
Row Counts
Detail Rows
Subtotals
Grand Total


Search...
Medical Inventory ...
Products
Suppliers
Purchase Orders
Order Items
Inventory Transactions
Reports
Dashboards
Dashboard
Medical Inventory DashBoard
As of Sep 4, 2025, 8:34 AM · Viewing as SIVACHANDRAN S
Refresh
Edit
Subscribe
Purchase Orders based on Suppliers
Record Count
Supplier ID
SUP001
SUP002
1
2
1
View Report (Purchase Orders based on Suppliers)
As of Sep 4, 2025, 8:34 AM

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.