# MEDICAL INVENTORY MANAGEMENT- (Developer)

College Name: BHARATHIDASAN COLLEGE OF ARTS AND SCIENCE

College Code:

bru3d

**TEAM ID:** NM2025TMID24686

**TEAM MEMBERS:**

Team Leader Name: AHAMED ASLAM.J

Email: [ahamedaslam861@gmail.com](mailto:ahamedaslam861@gmail.com)

Team Member 1: NANTHAKUMAR.C

Email: [nanthakumar5856@gmail.com](mailto:nanthakumar5856@gmail.com)

Team Member 2: ASHOK KUMAR. V

Email: [ak77296166@gmail.com](mailto:ak77296166@gmail.com)

Team Member 2: DHAMODHARAN.N

Email: [dharandhamo390@gmail.com](mailto:dharandhamo390@gmail.com)

# 1.INTRODUCTION

The Medical Inventory Management System is a comprehensive Salesforce application designed to streamline and manage various operational aspects of the medical inventory. It can efficiently maintain supplier details, manage purchase orders, track product details and transactions, and monitor expiry dates of products, thereby improving operational efficiency, data accuracy, and reporting capabilities.

## 1.1 Project Overview

This project is a comprehensive Salesforce application to streamline and manage various operational aspects of medical inventory. The system aims to efficiently maintain supplier details, manage purchase orders, track product details and transactions, and monitor the expiry dates of products. Maintain detailed records of suppliers, including contact information. Catalog product information, including descriptions, stock levels. Monitor and track product expiry dates to avoid using expired items. Comprehensive reports to track supplier performance, and purchase orders.

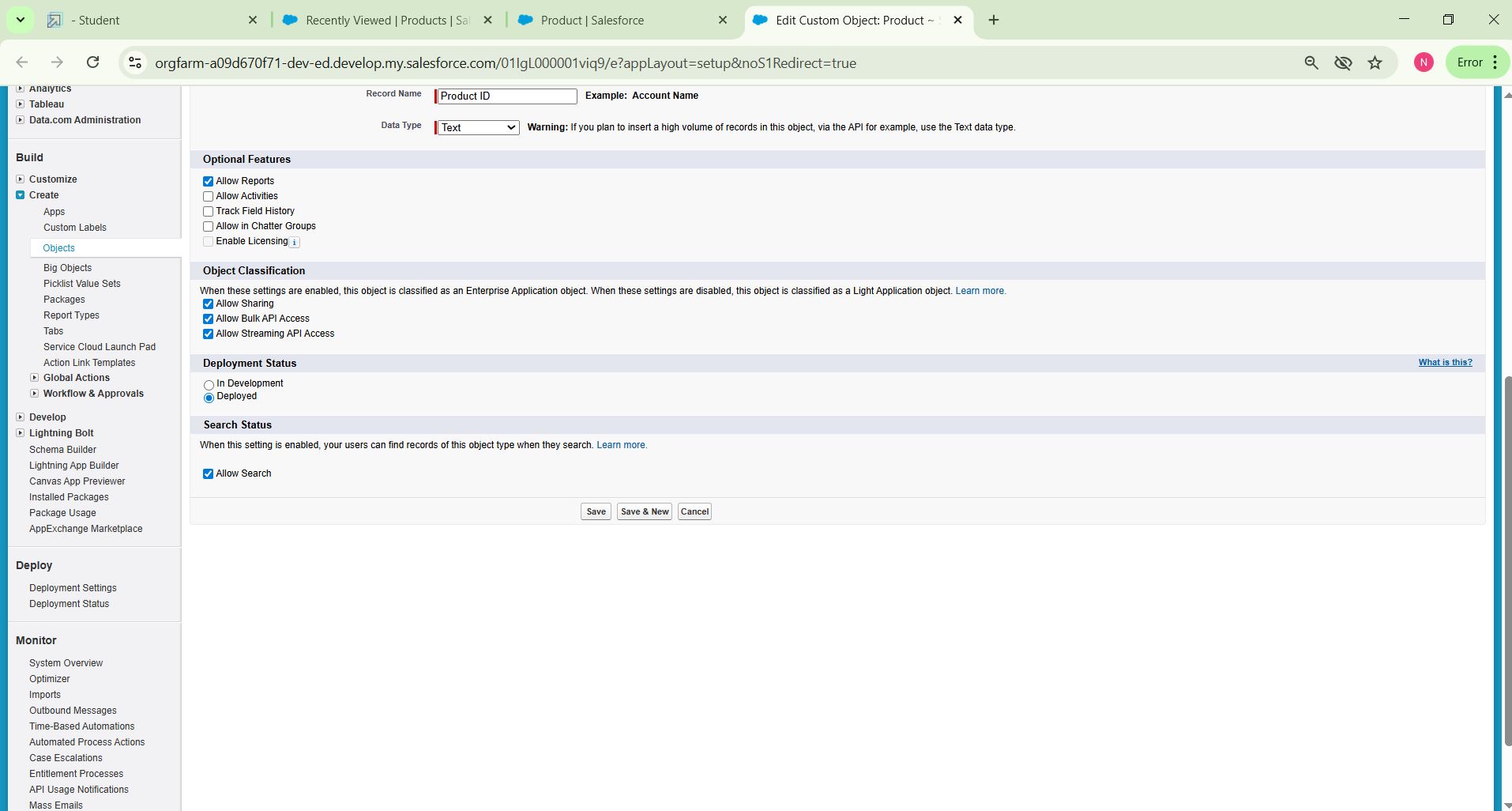
### Creating a Product Object

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.

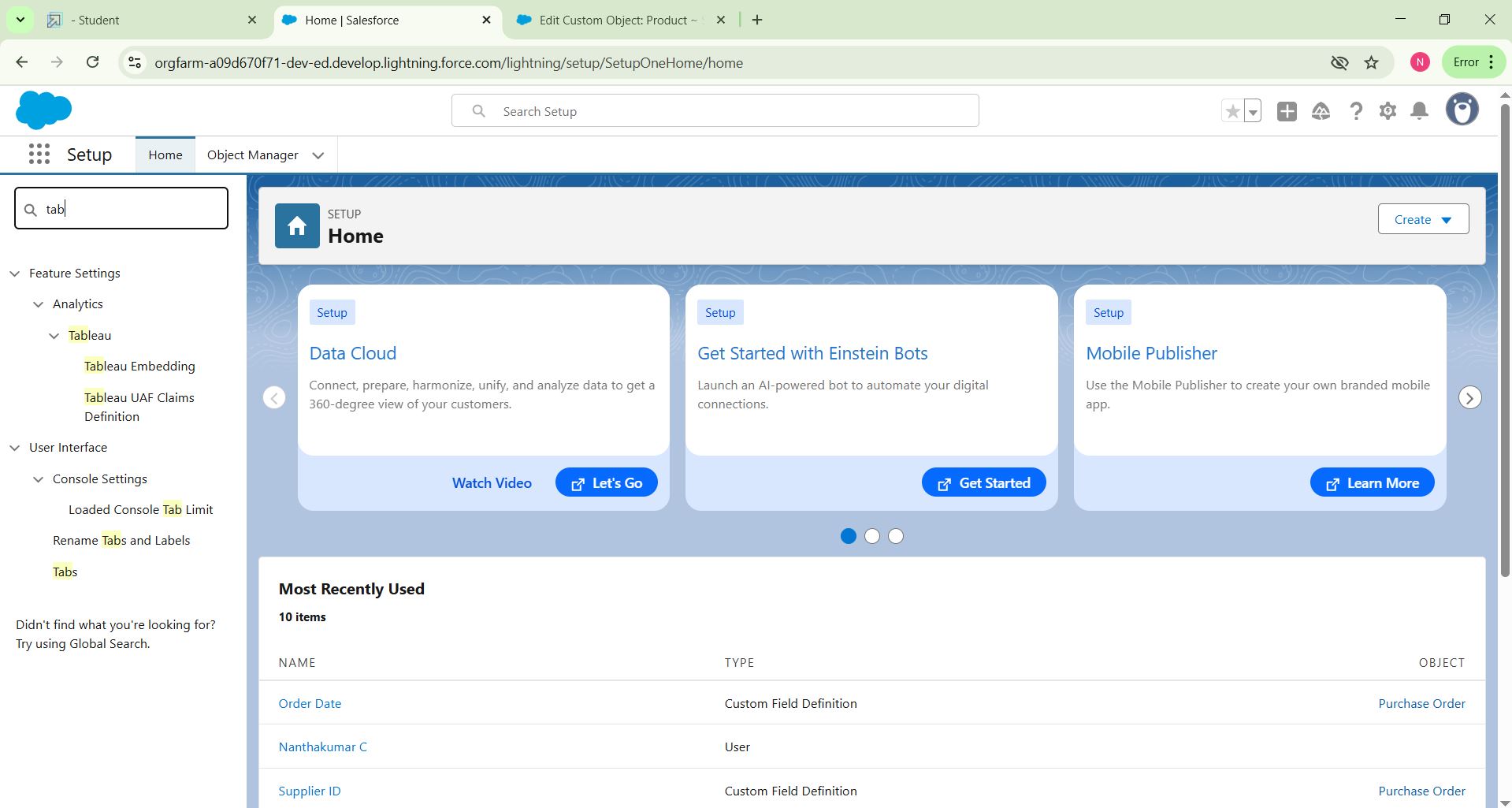
10.Click on Save and New

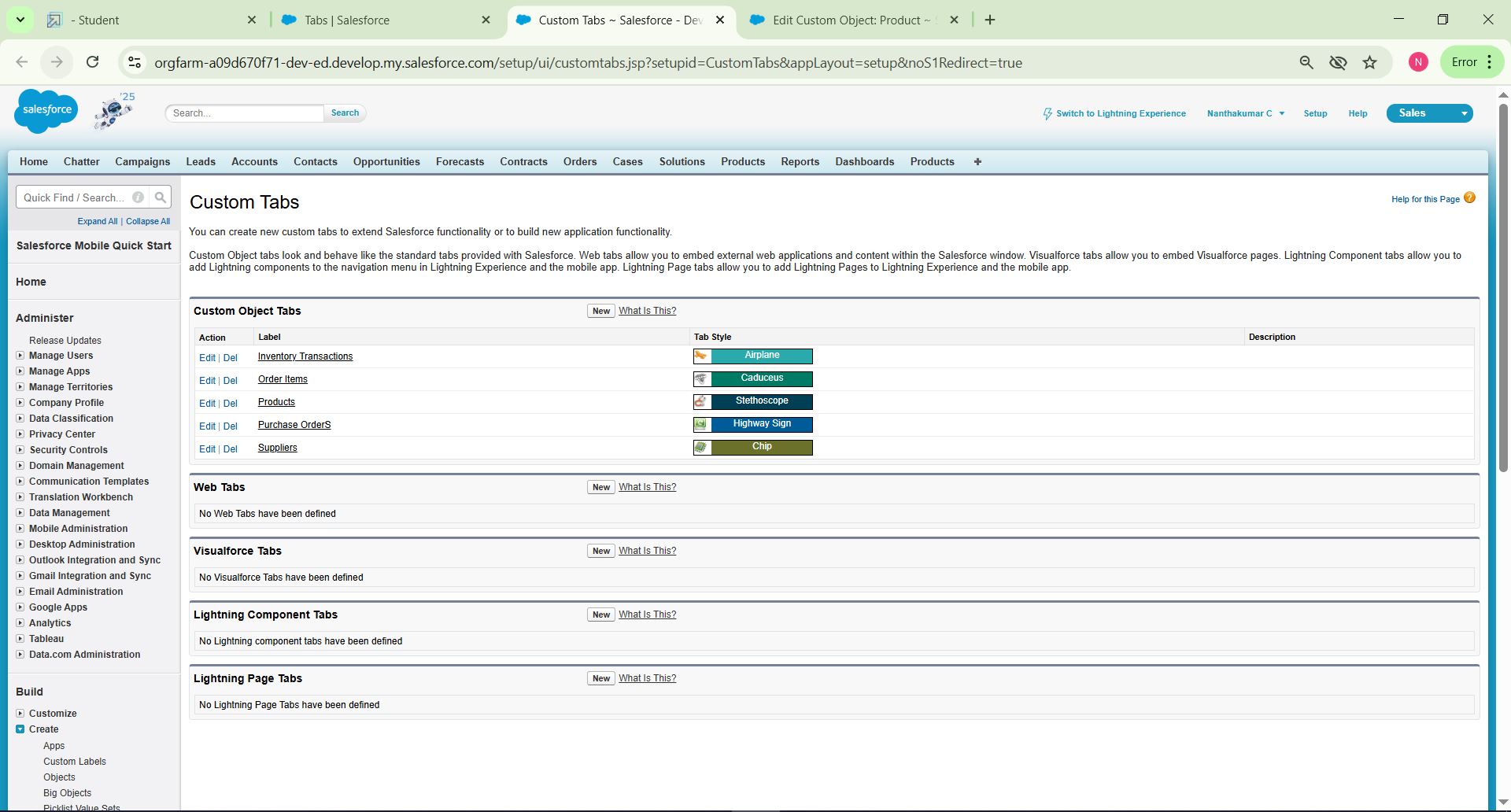
### 



### Tabs

### 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.



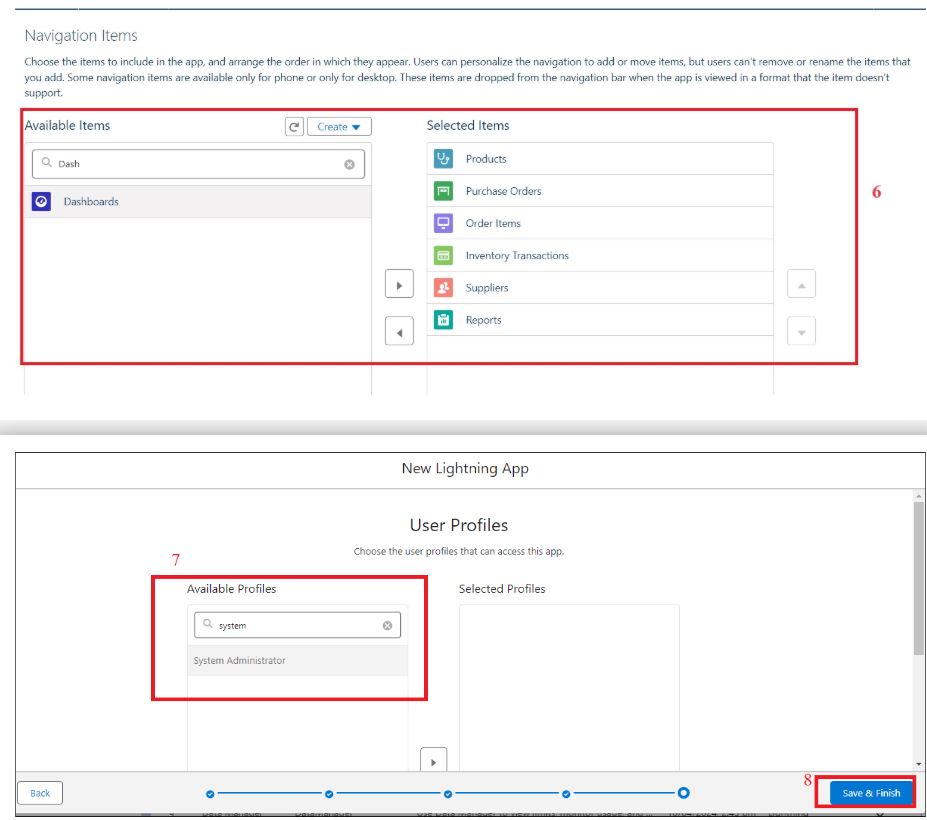


Now create the Tabs for the remaining Objects, they are “Purchase Order, Order Item, Inventory Transaction, Supplier”.

### The Lightning App

### A Lightning App in Salesforce is a collection of items that work together to serve a particular function for the end-users. These items can include standard and custom objects, tabs, utilities, and other productivity tools. Lightning Apps are designed to provide a more intuitive and efficient user experience compared to traditional Salesforce apps.

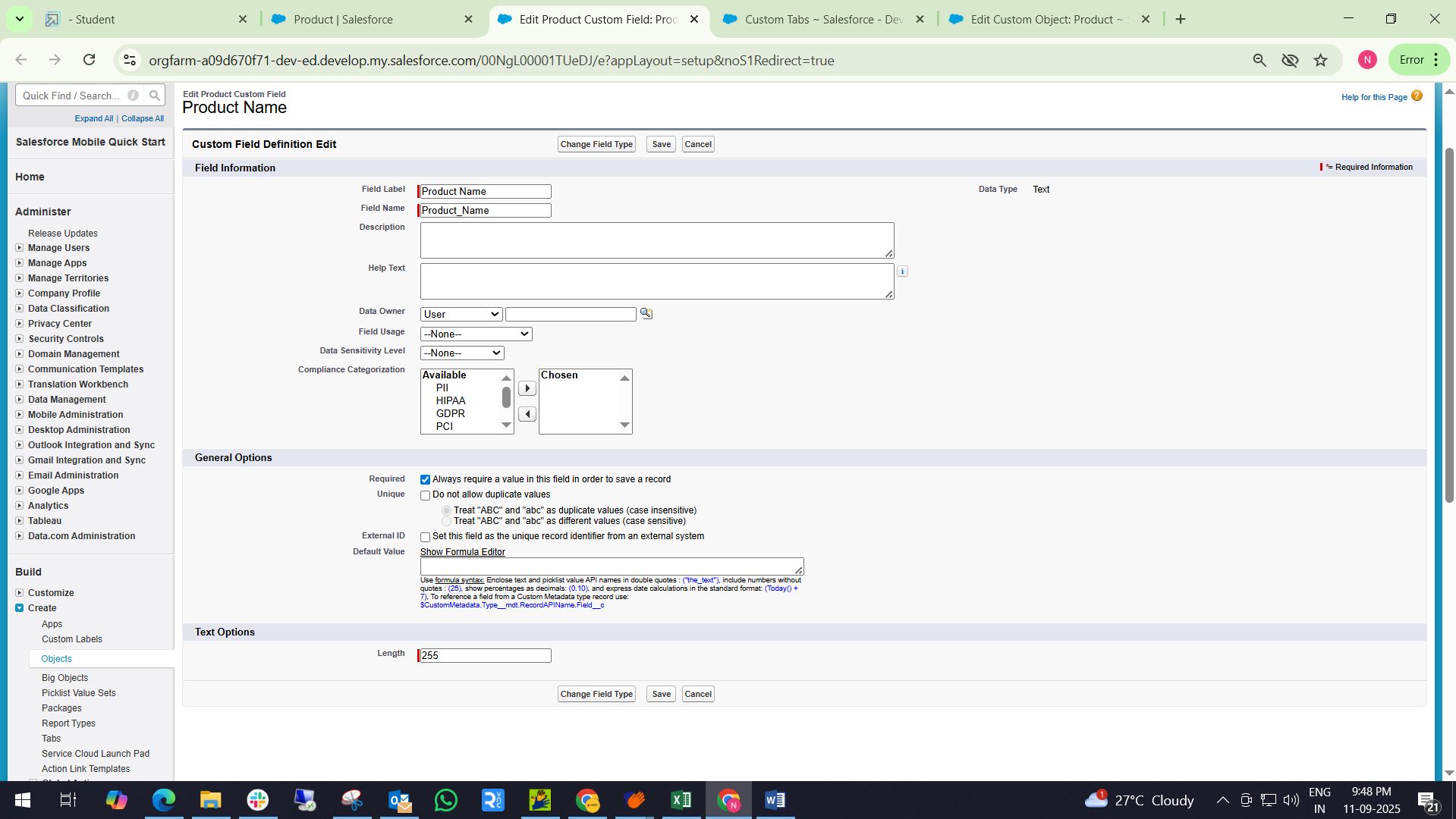
# 



### Creating a Text Field in Product Object

To create fields in an object:

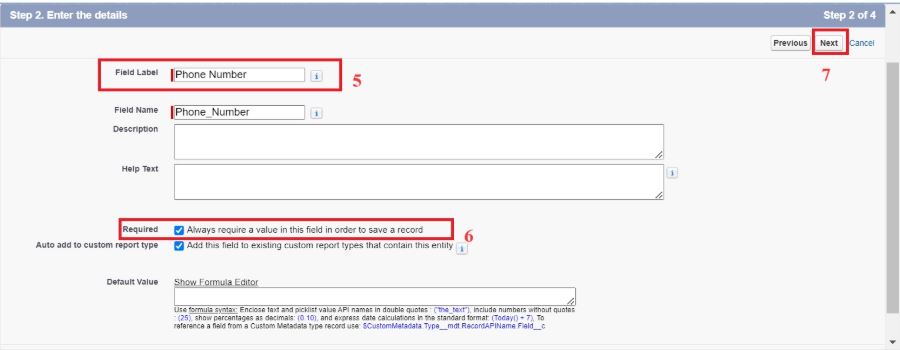
1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select Product custom object.
4. Select Fields & Relationships from the left navigation
5. Click on New
6. Select Text field, click Next
7. Enter Field Label as “Product Name” and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



### Creating a Phone Field in Supplier object

 To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box>> click on the Supplier object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Phone” and click Next.
5. Enter the Field Label as “ Phone Number”.
6. Select Required Field.
7. Click on Next, Next and Save.

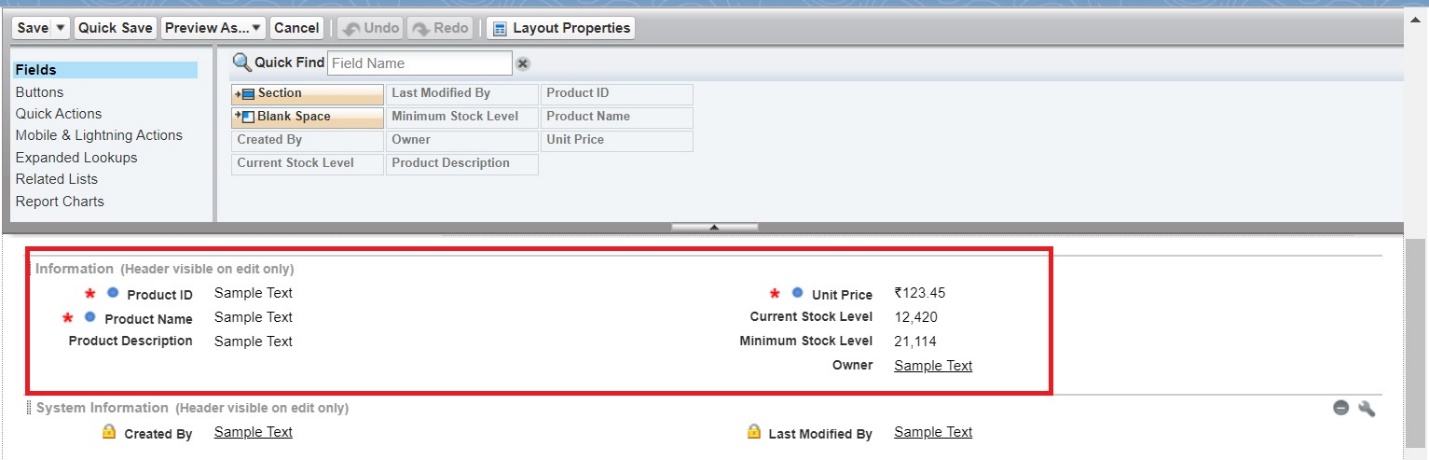


### Editing of Page Layouts

Page layouts in Salesforce are used to customize the organization, structure, and content of pages for viewing and editing records. They determine which fields, related lists, and custom links are visible to users, as well as the order and grouping of those elements.

### To edit a Page Layout in Product Object

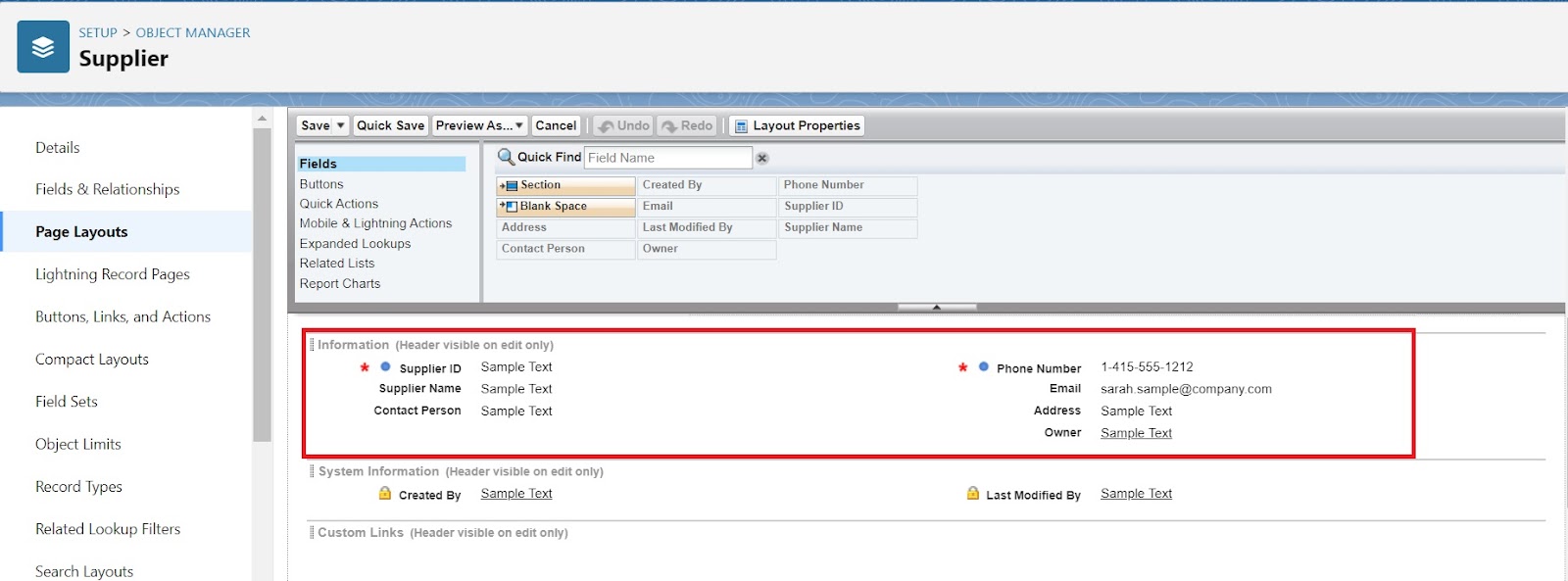
1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object >> Page Layouts .
2. Click on the Product Layout.
3. Drag and Arrange the field as shown below.



1. Click on Save.

### To edit a Page Layout in Supplier Object

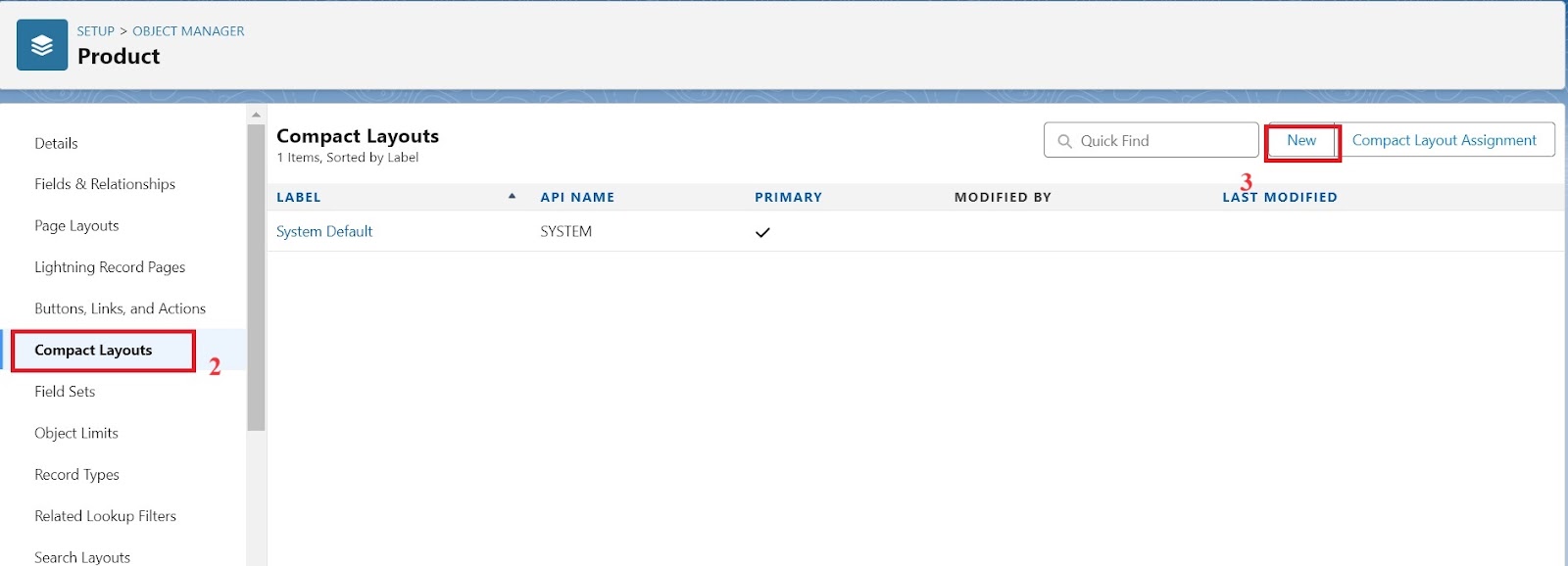
1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box >> click on the Supplier object >> Page Layouts.
2. Click on the Supplier Layout
3. Drag and Arrange the field as shown below

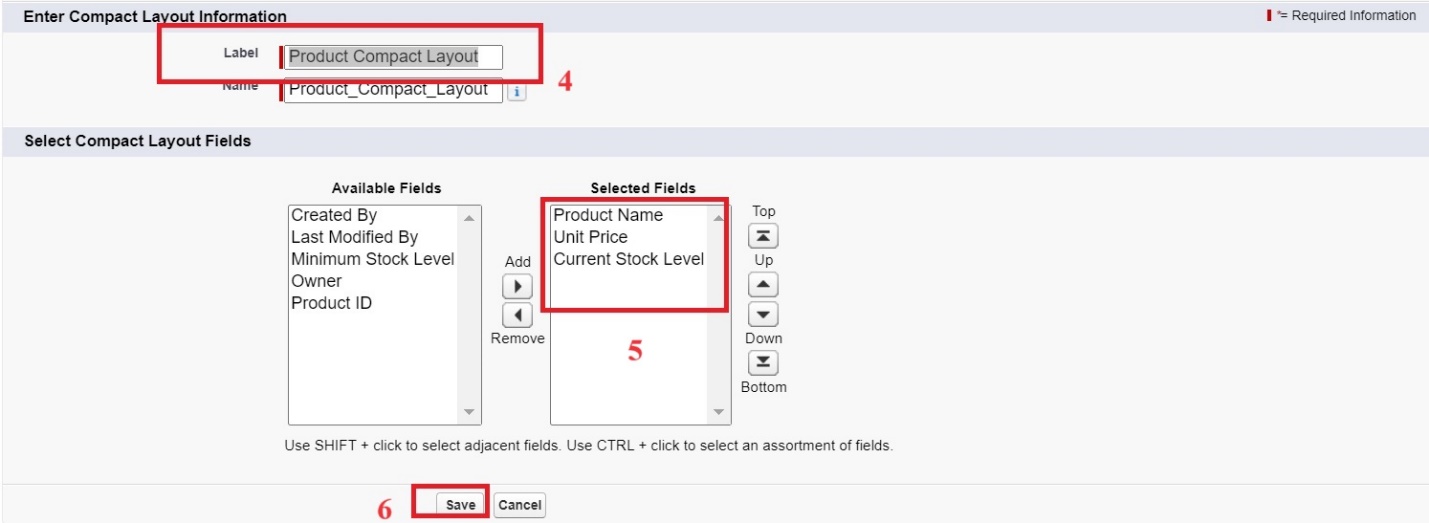


1. Click Save

### To create 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.

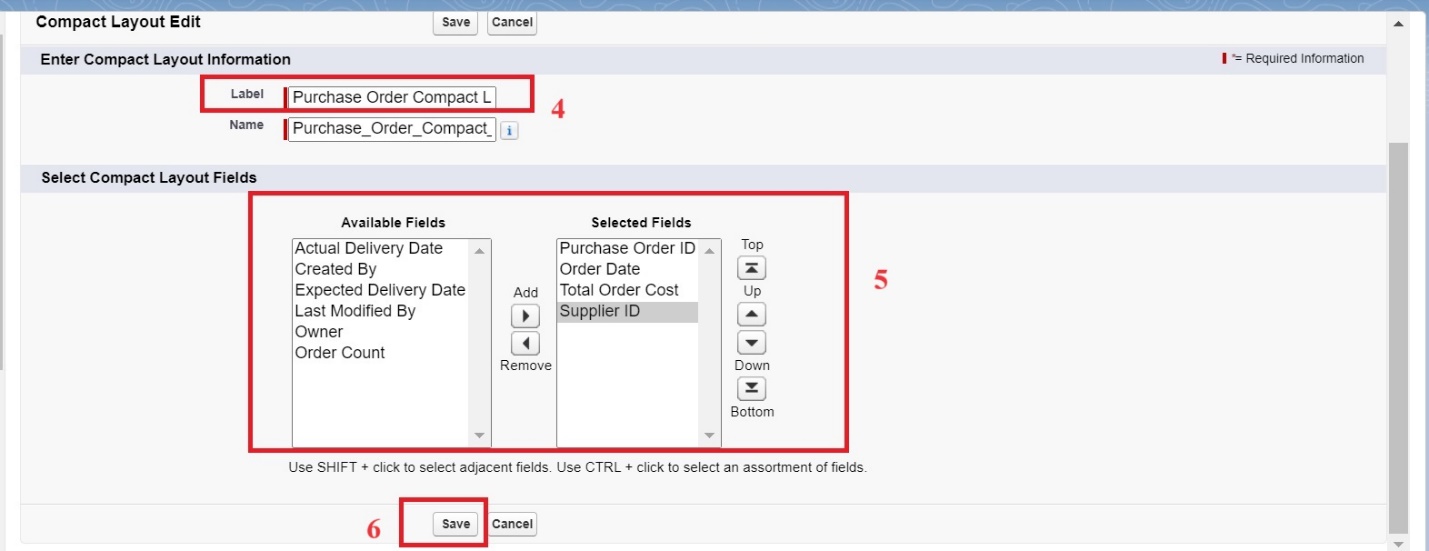


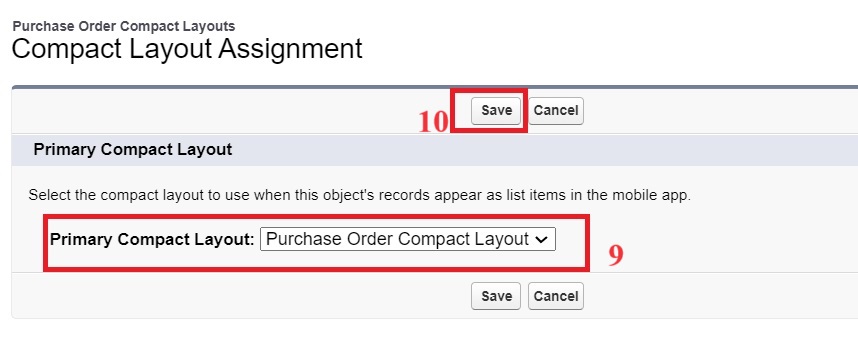




### To create a Compact Layout to a Purchase Order Object

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 Compact Layouts in the sidebar .
3. Click on New.
4. Enter the Label as “Purchase Order Compact Layout”.
5. Select the Compact Layout Fields : Select Purchase Order ID, Order Date, Total Order Cost, Supplier ID.
6. Click Save.
7. Click Compact Layout Assignment.
8. Click Edit Assignment.
9. Choose "Purchase Order Compact Layout" from the dropdown.
10. Click Save.



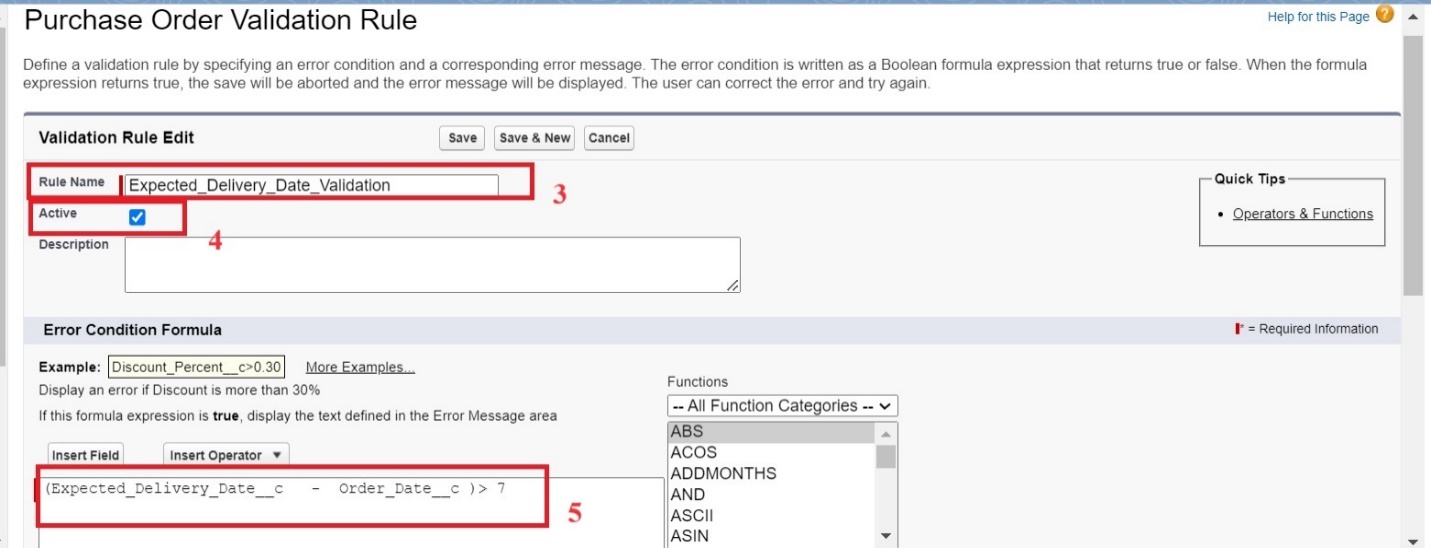


### Validation Rules

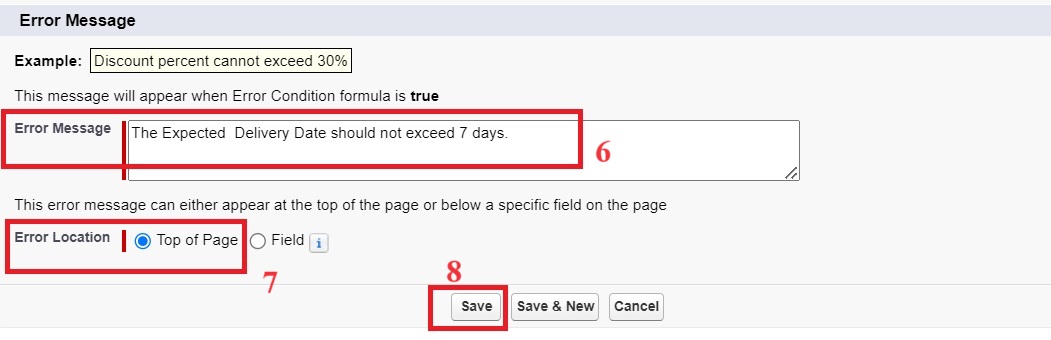
Validation rules in Salesforce are used to ensure data integrity by preventing users from saving invalid data in records. They consist of a formula or expression that evaluates the data in one or more fields and return a value of true or false. When the rule's criteria are met (i.e., the expression evaluates to true), an error message is displayed, and the user is prevented from saving the record until the issue is resolved

### To create an Expected Delivery Date Validation rule to a Employee 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



1. Enter the Error Message as “The Expected  Delivery Date should not exceed 7 days.”.
2. Select the Error location as Top of Page
3. Click Save.

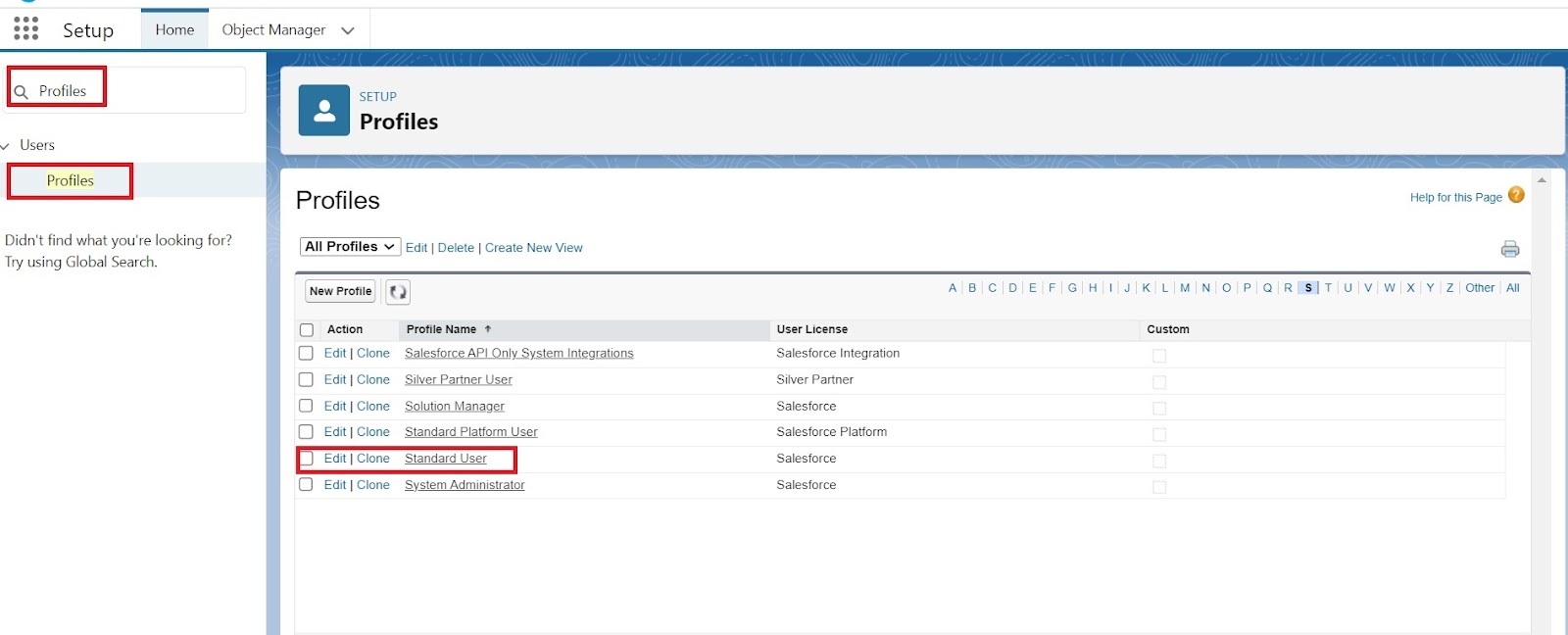


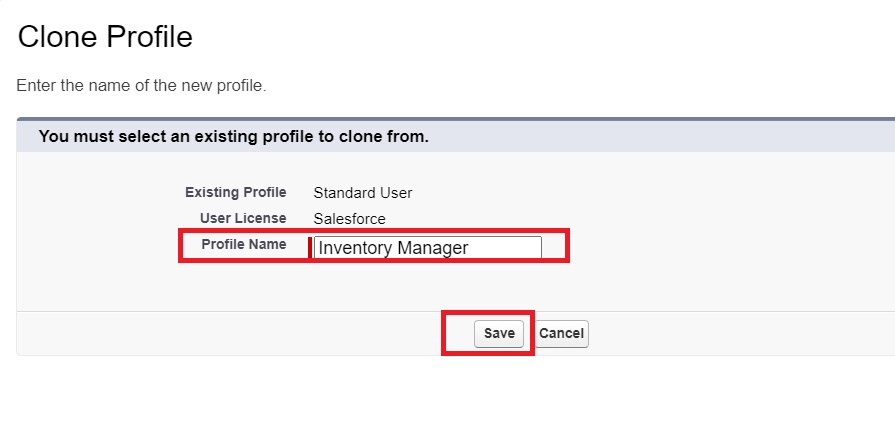
### Profiles

Profiles in Salesforce are fundamental to the platform's security model, defining what users can do within the organization. Profiles control a user’s permissions to objects, fields, tabs, apps, and other settings. Each user in Salesforce must be assigned a profile, and the profile assigned to a user determines what they can see and do in the system.

### To create an Inventory Manager Profile

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Inventory Manager) >> Save.





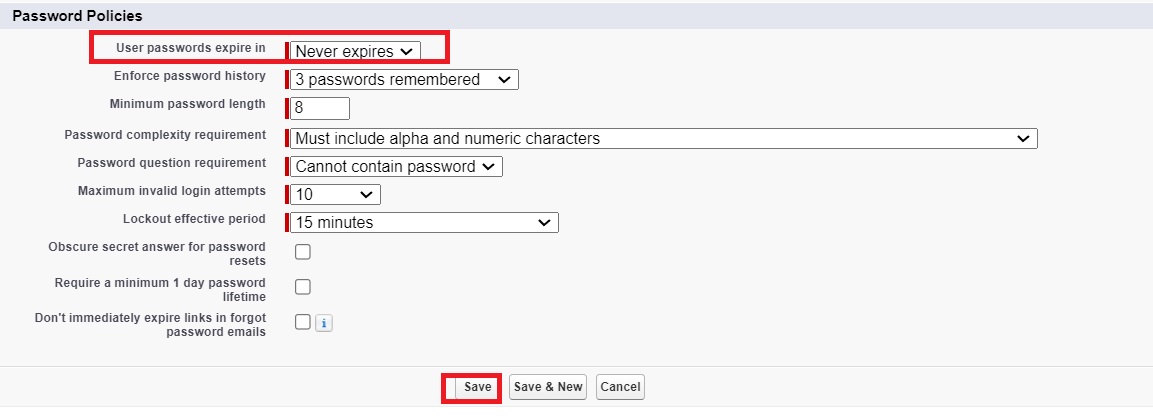
1. While still on the profile page, then click Edit.
2. Select the Custom App settings as default for the Medical Inventory Management.



1. Scroll down to Custom Object Permissions and Give access permissions as mentioned in the below diagram.

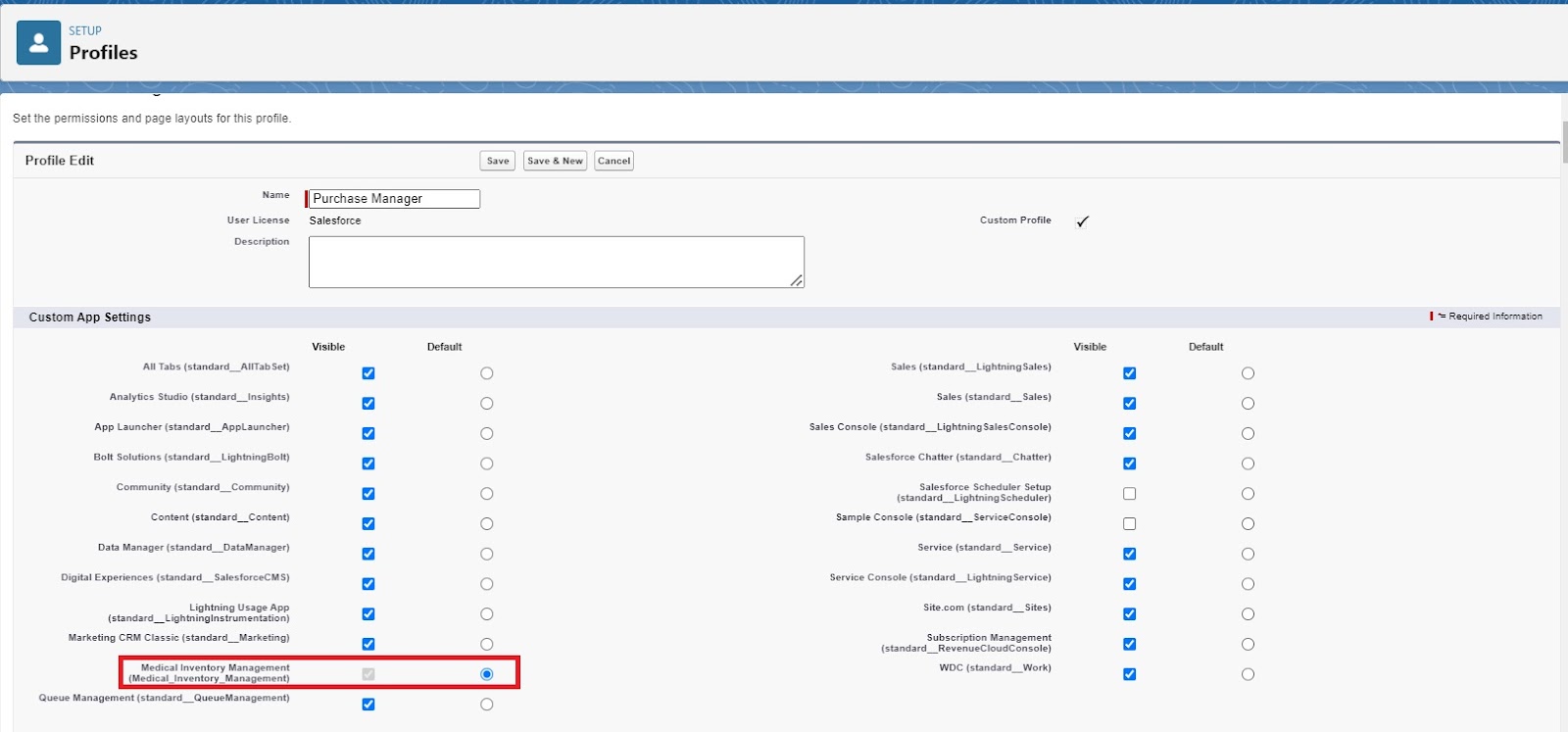


1. Change the password policies as mentioned :
2. User passwords expire in should be “ never expires ”.
3. Minimum password length should be “ 8 ”, and click save.

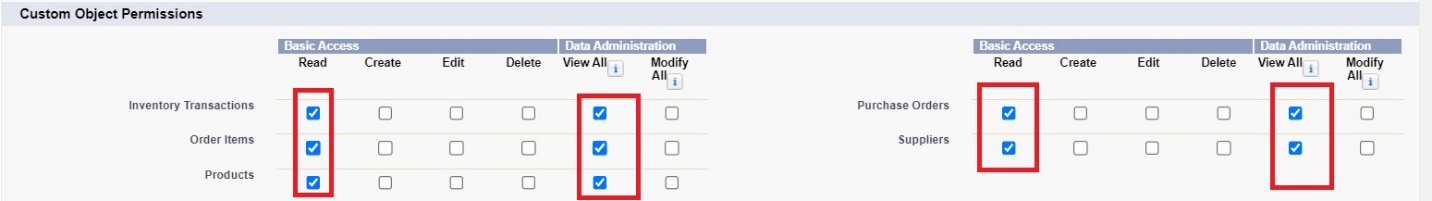


### To create an Purchase Manager Profile

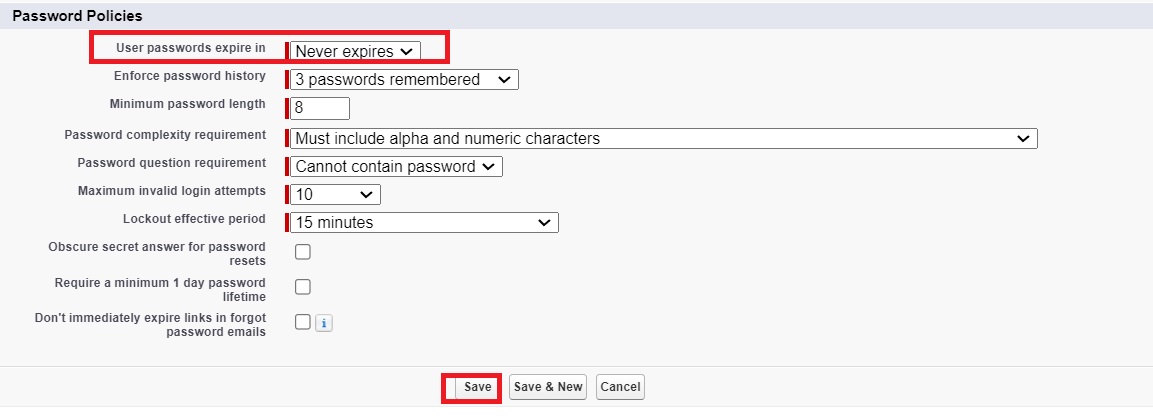
1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Purchase Manager) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Medical Inventory Management.



1. Scroll down to Custom Object Permissions and Give access permissions as mentioned in the below diagram.



1. Change the password policies as mentioned :
2. User passwords expire in should be “ never expires ”.
3. Minimum password length should be “ 8 ”, and click save.

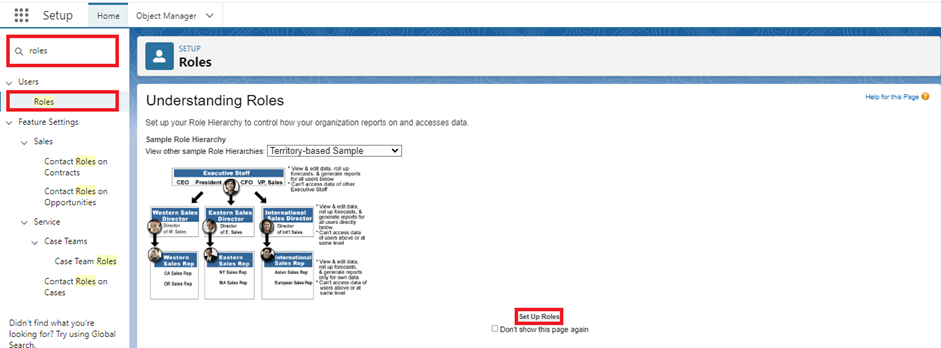


### Roles

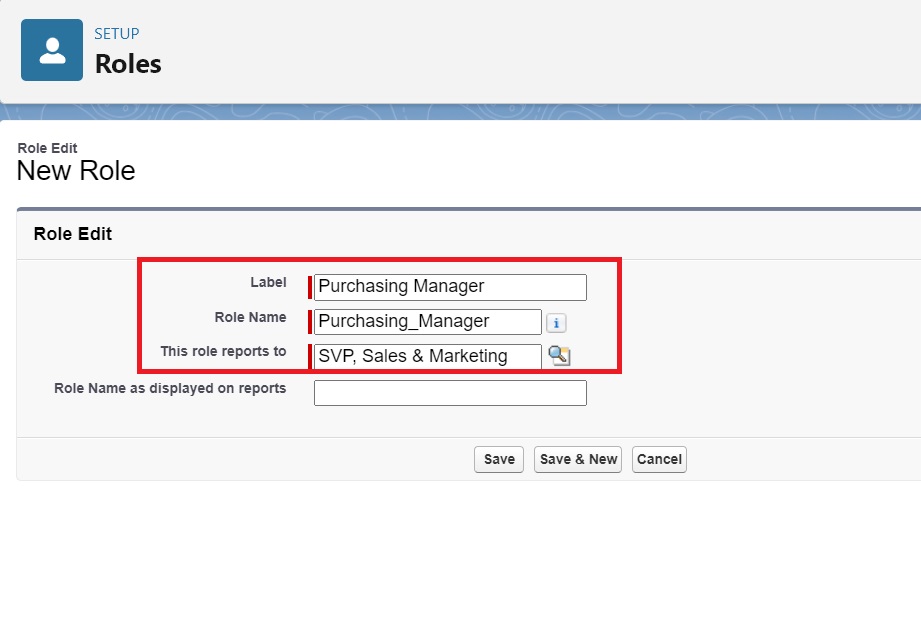
Roles in Salesforce are used to control record-level access and define the hierarchy of an organization, determining the level of visibility and sharing of records among users. Roles work in conjunction with profiles to provide a robust security model. While profiles control what actions users can perform (object and field permissions), roles control which records users can see based on their position in the hierarchy.

### Create a Purchasing Manager Role.

1. Go to quick find  >>  Search for Roles >> click on Set Up Roles.



1. Click on Expand All and click on add role under SVP, Sales & Marketing  role.
2. Give Label as “Purchasing Manager” and Role name gets auto populated. Then click on Save.

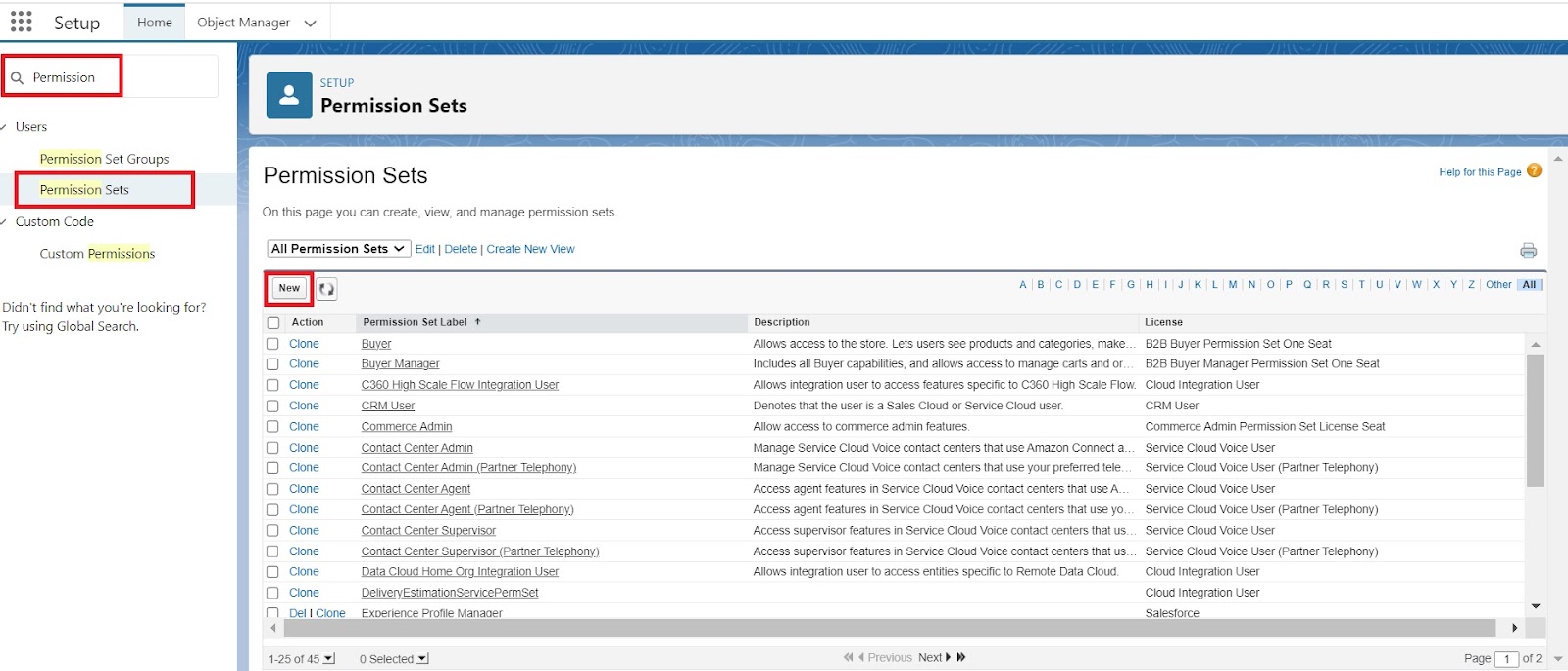


### Permission Sets

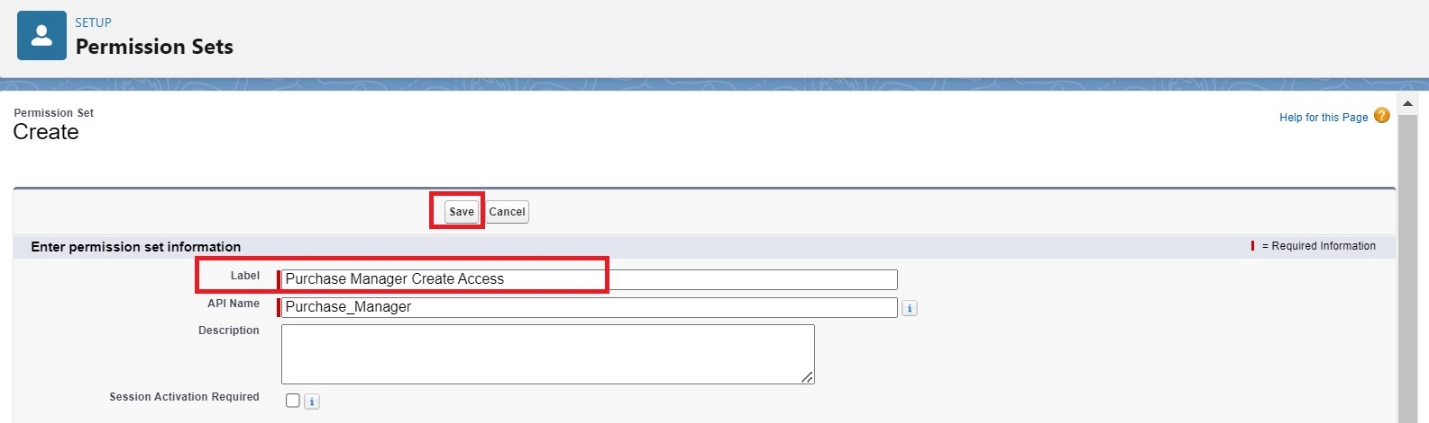
Permission Sets in Salesforce are a powerful tool to extend user permissions beyond what is defined in their profiles. They allow administrators to grant additional access to various tools and functions without altering the user's profile. Permission sets are particularly useful for providing specialized permissions to specific users without the need to create multiple profiles.

### Create a Permission Set.

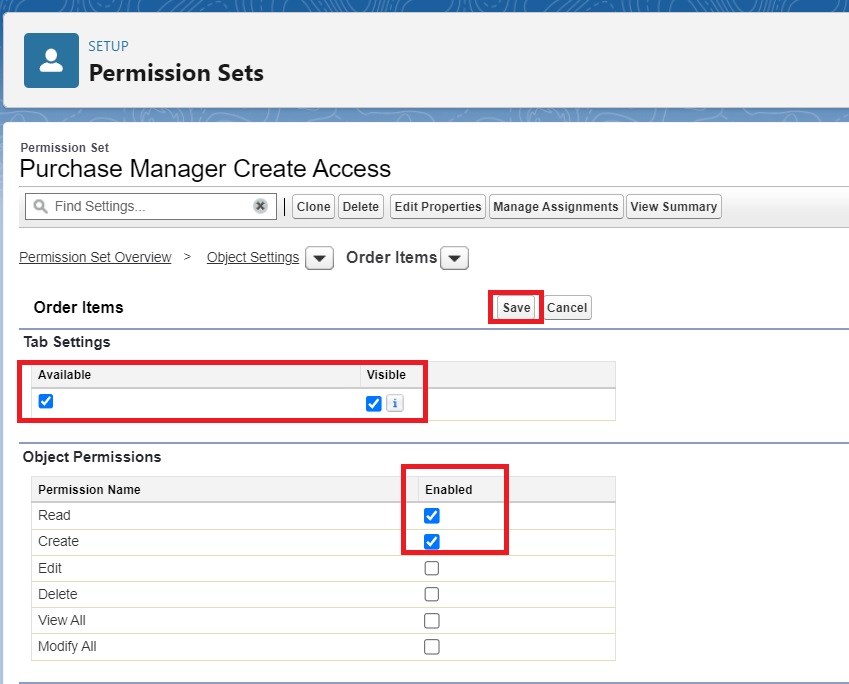
1. Go to setup >> type Permission in quick find box >> Select Permission Set >> click on New.



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



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



1. 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.



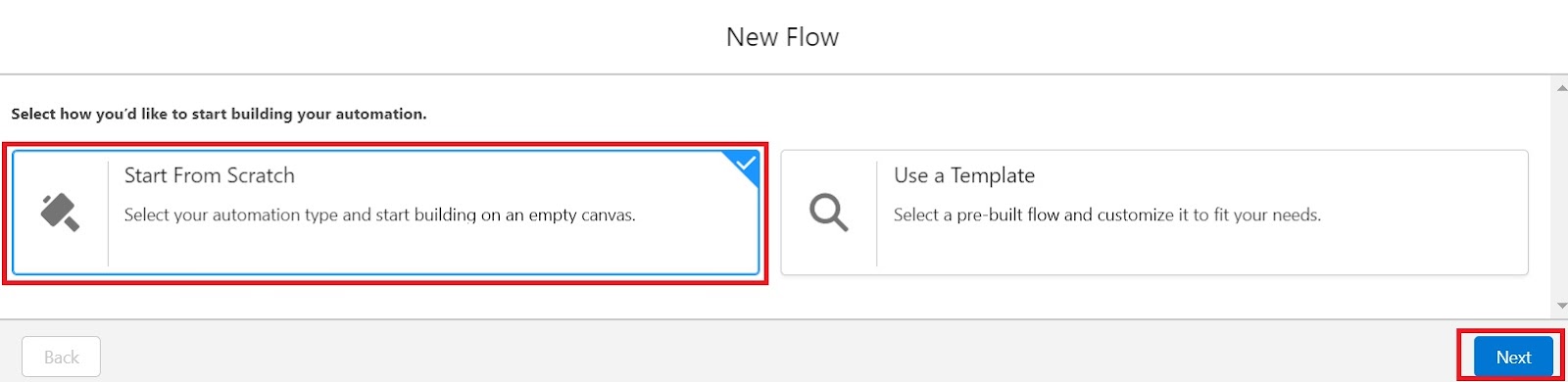
1. Select No Expiration date >> Click on Assign.

### Flows

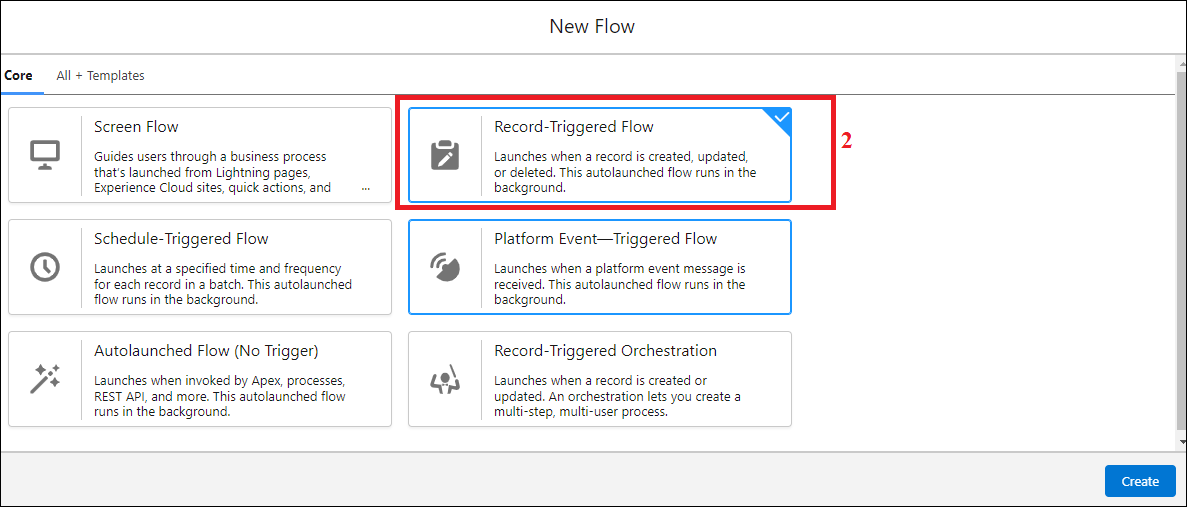
Flows in Salesforce, part of the Lightning Flow product, are powerful automation tools that help you collect data and perform actions in your Salesforce environment. Flows can be used to automate business processes, guide users through tasks, and integrate with external systems. They are highly versatile and can be configured to meet a wide range of business requirements without the need for custom code.

### Create Flow to update the Actual Delivery Date.

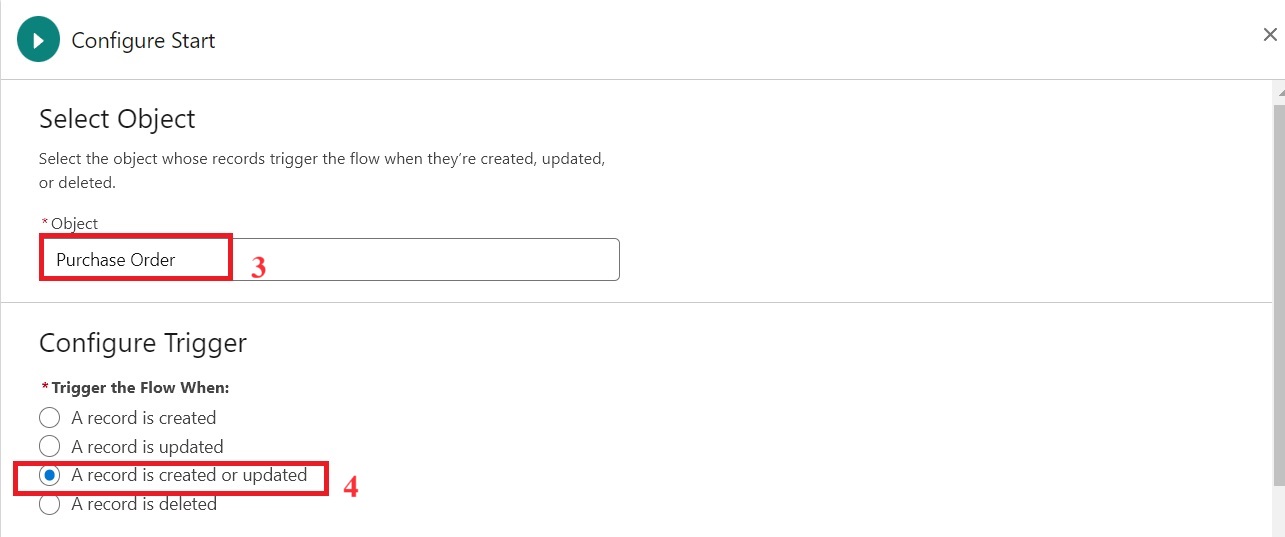
1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow >> Start From Scratch .



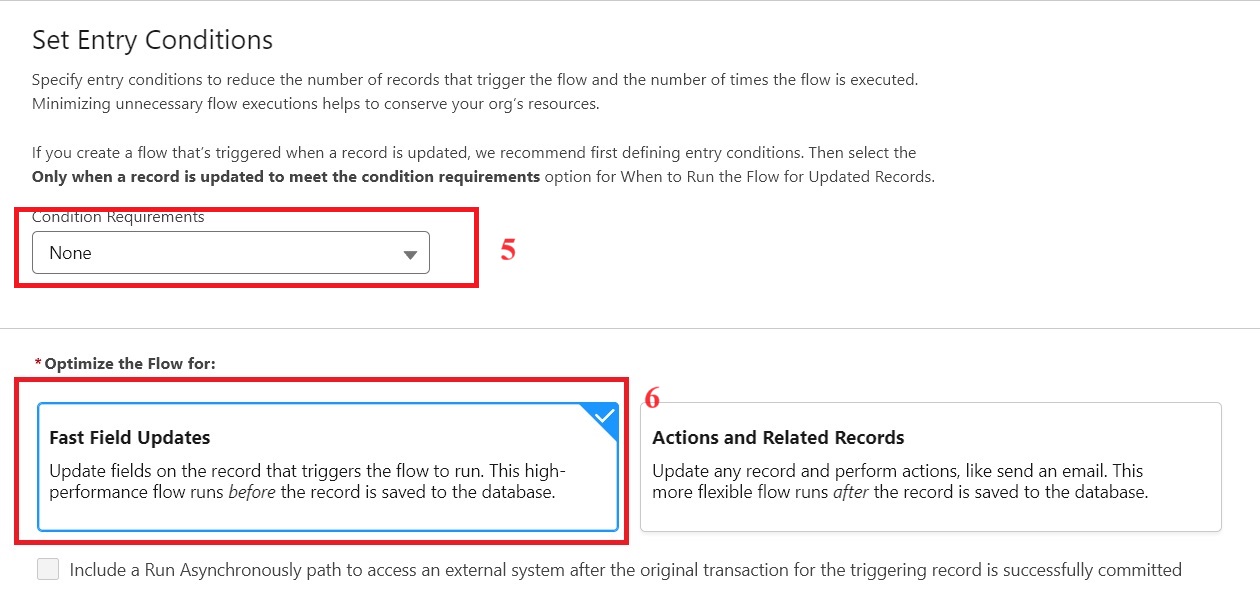
1. Select the record Triggered flow.Click on create.



1. Under Object select “Purchase Order”
2. Select  A record is created or updated



1. Set Entry Conditions : None
2. Select Fast Field Updates and click on Done



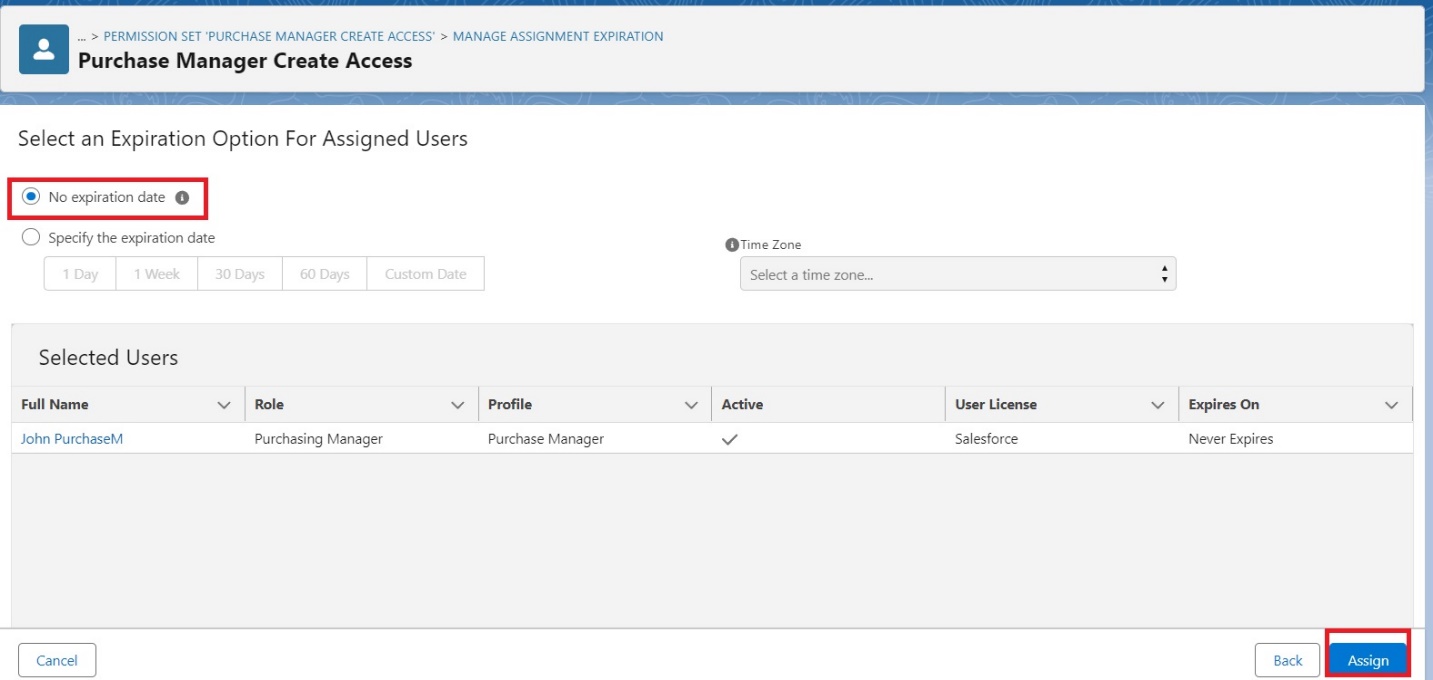
1. Under the record trigger flow click on the “+” icon and select Get Records.
2. Enter Label as “ Get Purchase Record ”.
3. For Object select Purchase Order.
4. For Condition Requirements , select All Conditions are Met(AND)

        For the first condition select as follows:

                                 Field: Id

                                 Operator: Equals

                          Value: {!$Record.Id}



### Triggers

Triggers in Salesforce are pieces of Apex code that execute before or after specific data manipulation events on Salesforce records, such as insertions, updates, deletions, and undeletions. They are powerful tools for automating complex business logic and ensuring data integrity by enforcing custom validation rules and workflows that cannot be achieved through declarative tools alone.

### Create 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.

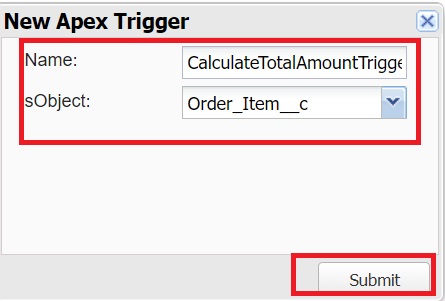
ii)Click on Developer Console: Click on the "Developer Console" option from the Setup menu.   This will open the Developer Console in a new browser tab or window.

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 it.

### Reports

Reports in Salesforce provide a powerful way to visualize and analyze data stored in your Salesforce organization. They allow users to create, customize, and share different types of reports based on data from standard and custom objects. Reports help organizations make informed decisions by providing insights into key metrics, trends, and performance indicators.

### Create a Purchase Orders based on Suppliers(Summary) Report

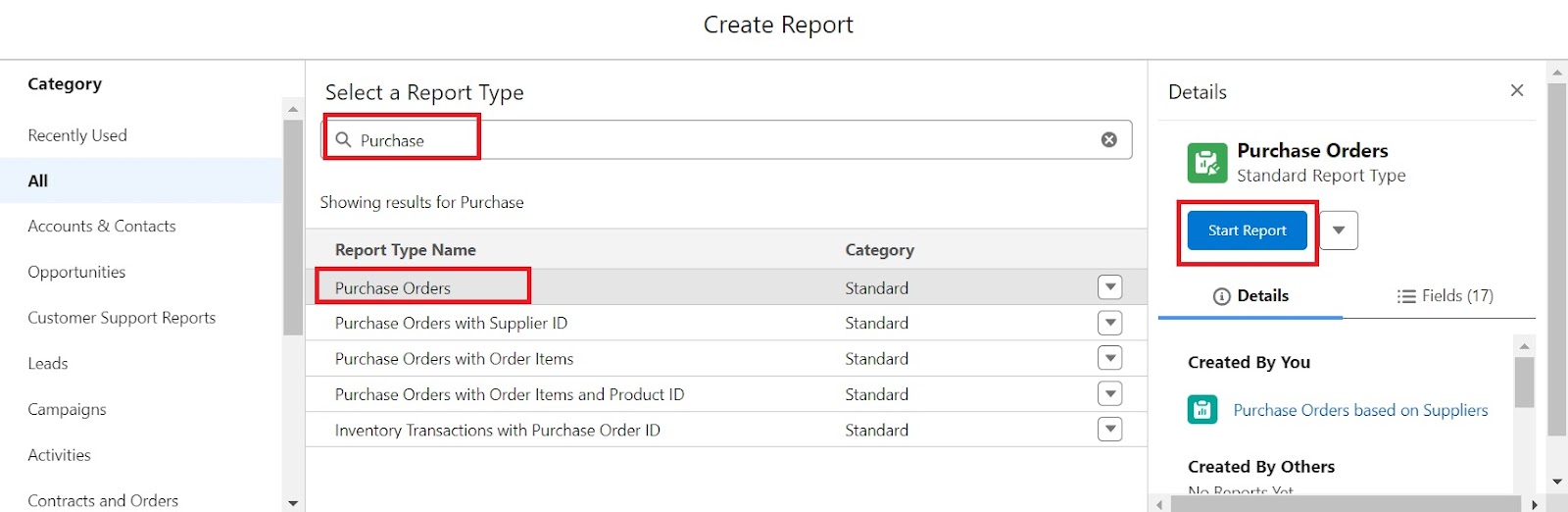
1.  Click App Launcher

2.  Select Medical Inventory Management App

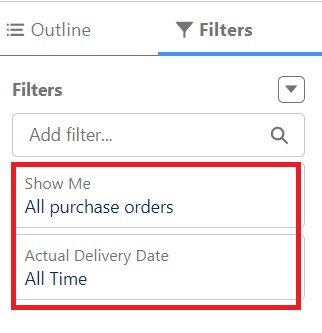
3.  Click on Reports tab

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).

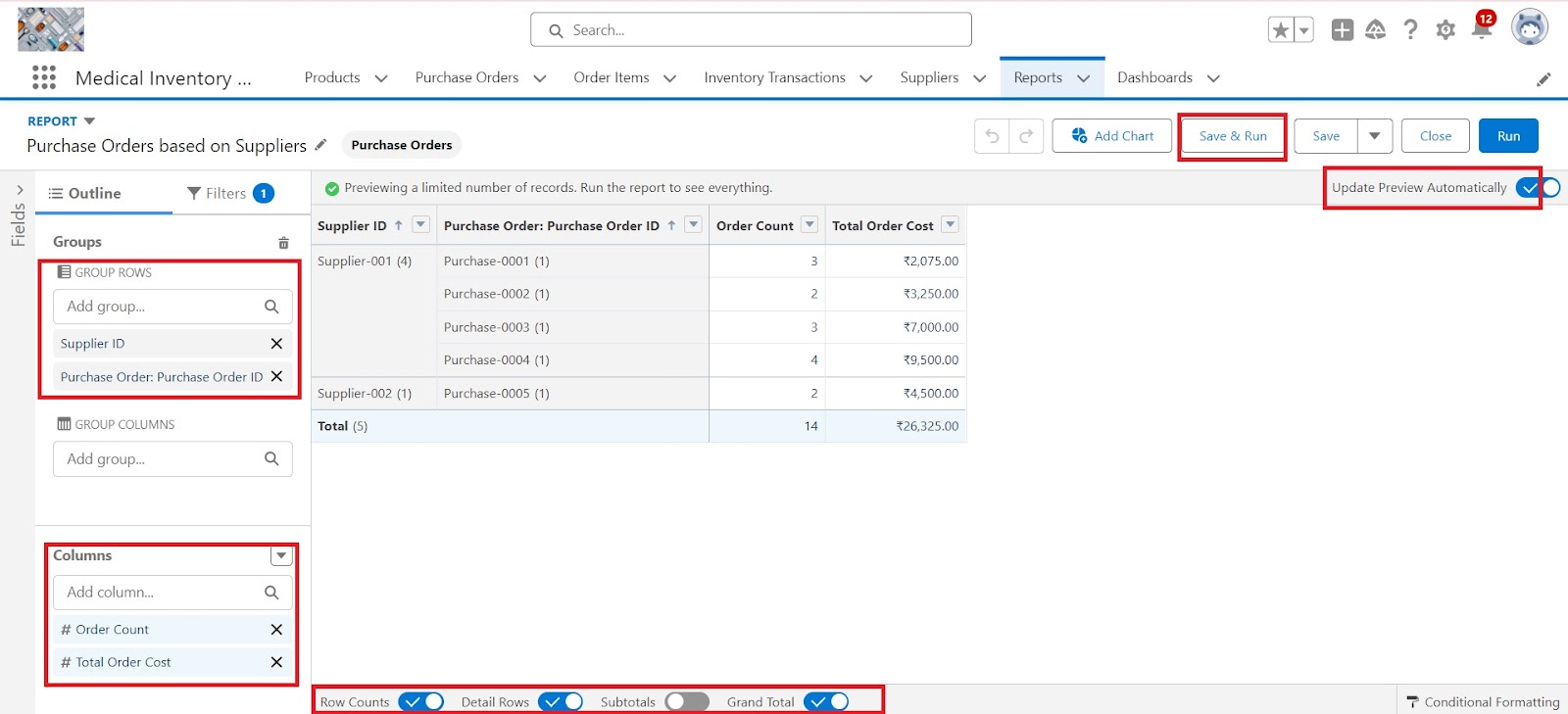
8.  Click save and run

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”)



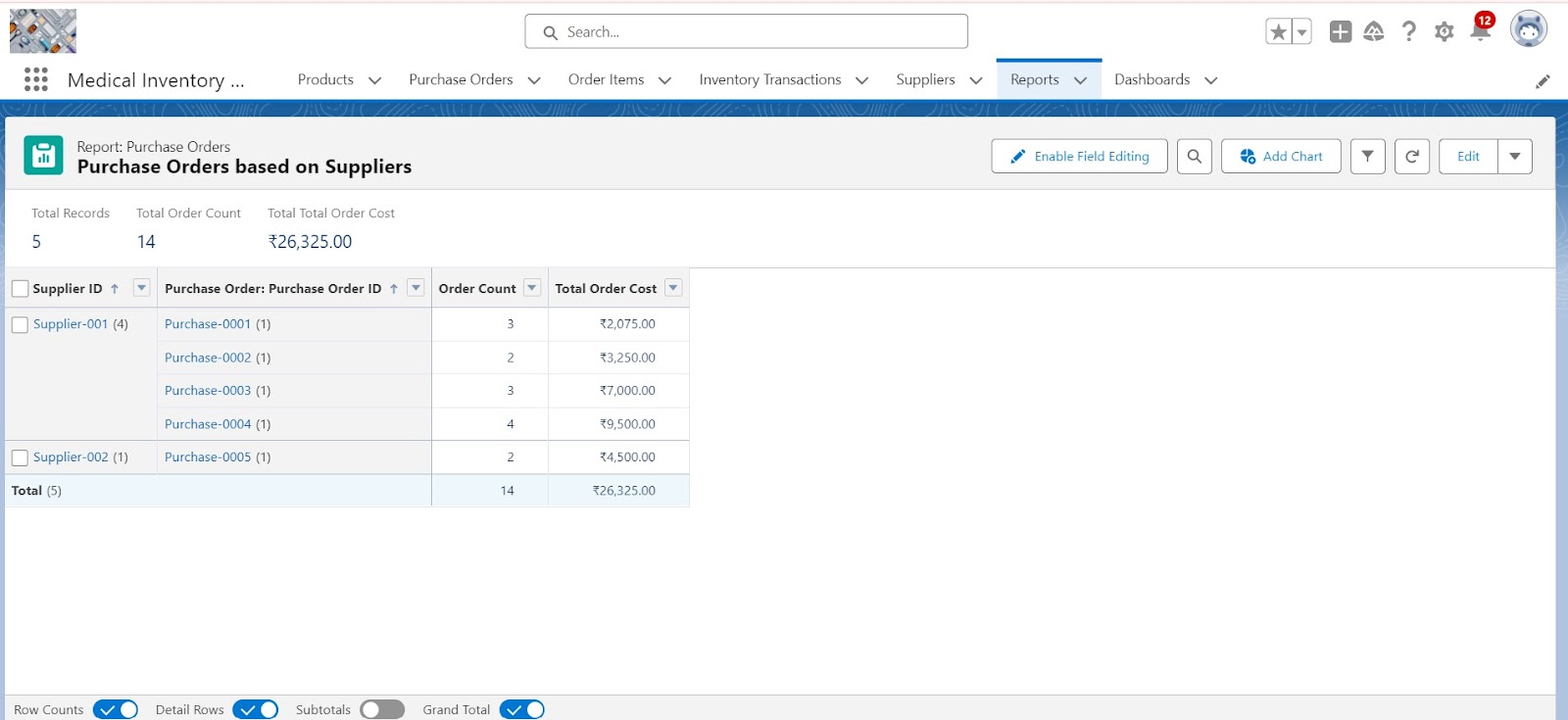
View Report

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

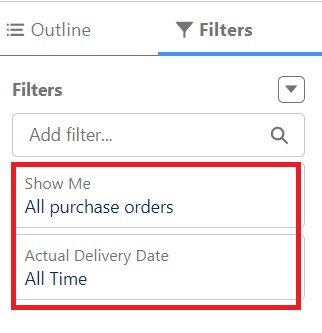
2.  Select Medical Inventory Management App

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

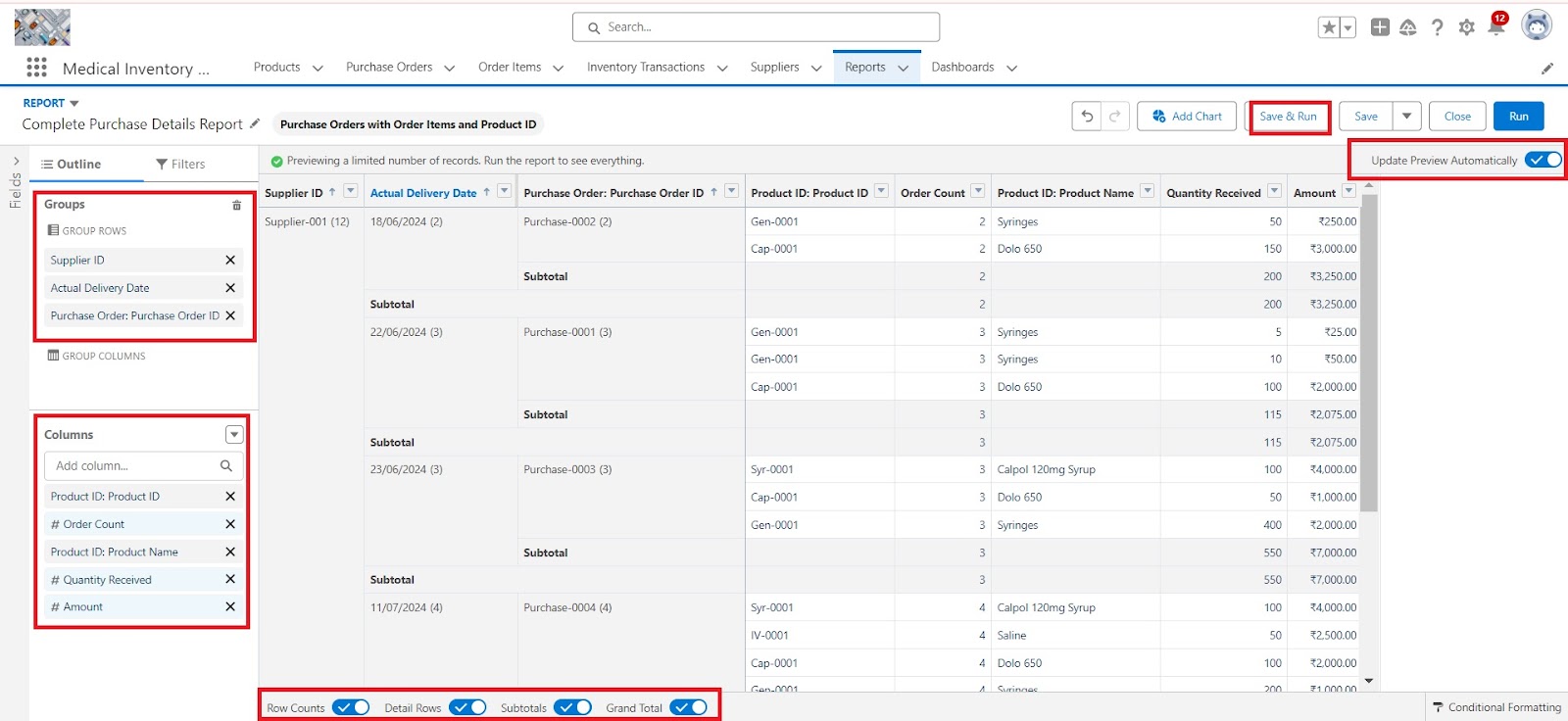


7. Customize your report, in group rows select – Supplier ID, Actual Delivery Date,  Purchase Order: Purchase Order ID, for columns Product ID : Product ID,  Product ID : Product Name, Order Count, Quantity Received, Amount  (In this way we are making a Summary Report).

8.  Click save and run

9.  Give report name – Complete Purchase Details Report

10.  Click Save

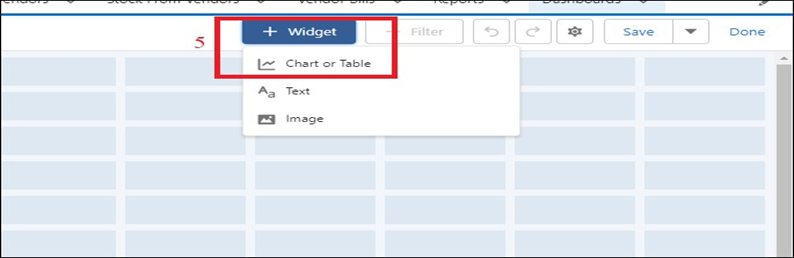


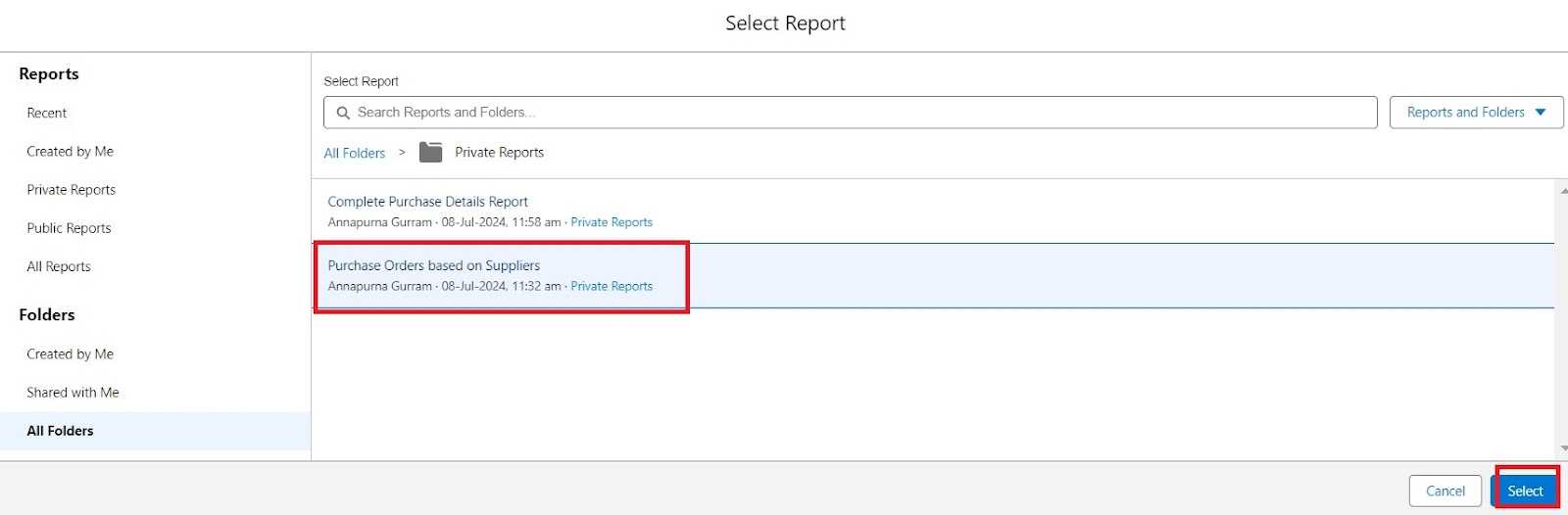
### Dashboards

Dashboards in Salesforce are dynamic visual representations of key metrics and data from reports, providing a consolidated view of organizational performance and trends. They are powerful tools for monitoring real-time data, tracking progress towards goals, and gaining actionable insights at a glance. Dashboards consist of components such as charts, tables, metrics, and gauges that display data from underlying reports.

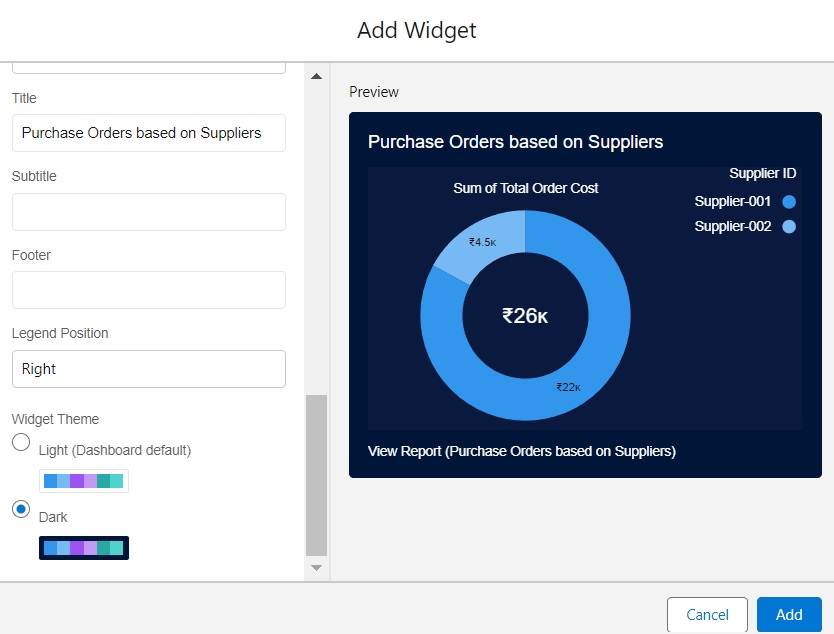
### Create Dashboard

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.









### View Dashboard

1.   Click on App Launcher on the left side of the screen.

2.   Search Medical Inventory Management & click on it.

3.   Click on Dashboard Tab.

4.   Click on Medical Inventory DashBoard see graph view of records

