

P.S.R.R. COLLEGE OF ENGINEERING

SIVAKASI - 626140

VIRUDHUNAGAR DISTRICT

**DEPARTMENT
OF
COMPUTER SCIENCE AND ENGINEERING**



Submitted By,

- **Kaleeswari B**
- **Gunavarsha M**
- **Maunika MS**
- **Nika J**

Medical Inventory Management

Project Overview

Project Type

Database-driven Web / Desktop / Mobile Application .

Objective

The goal of the Medical Inventory Management System (MIMS) is to efficiently manage the storage, distribution, and tracking of medical supplies, drugs, and equipment in hospitals, clinics, and pharmacies. The system automates the manual inventory process to ensure real-time monitoring, reduce human error, prevent stockouts or overstocking, and improve overall healthcare logistics.

Problem Statement

In many healthcare institutions, managing medical supplies manually results in inefficiency, errors, and stock mismanagement. Lack of accurate records often causes delays in treatment, wastage of medicines, and financial losses. There is a need for a digital system that can provide:

- Real-time inventory updates
- Automated alerts for low stock or expiry
- Supplier and purchase order management
- User access control for staff and administrators

User Story:

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.

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.

System Objectives:

The Medical Inventory Management System aims to:

1. Maintain detailed records of all medical items.
2. Automate stock-in and stock-out processes.
3. Track item expiry dates and send alerts.
4. Generate reports on inventory usage and trends.
5. Manage supplier and purchase information.
6. Provide secure login and role-based access.

System Scope

The system will handle:

- Medicines, surgical tools, and consumables.
- Inventory levels, batch numbers, and expiry dates.
- Departmental requests and stock issuance.
- Purchase orders and supplier management.
- Data analytics and reporting.

Project Flow:

Milestone 1 : Creation of developer account

Milestone 2 : Object Creation

Milestone 3 : Tabs

Milestone 4 : The Lightning App
Milestone 5 : Fields
Milestone 6 : Updating of Page Layouts
Milestone 7 : Compact Layouts
Milestone 8 : Validation rules
Milestone 9 : Profiles
Milestone 10 : Roles
Milestone 11 : Users
Milestone 12 : Permission Sets
Milestone 13 : Flows
Milestone 14 : Triggers
Milestone 15 : Reports
Milestone 16 : Dashboards
Milestone 17 : Conclusion

What you'll learn:

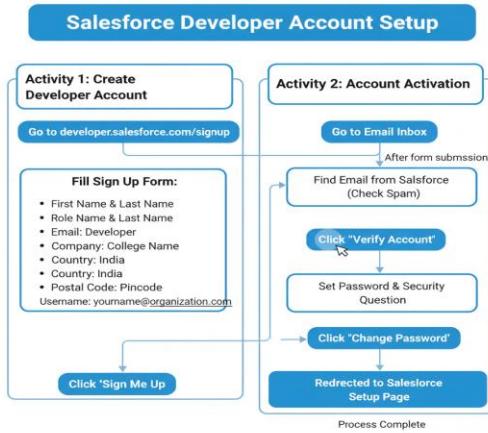
1. Real Time Salesforce Project
2. Object & their relationship in Salesforce
3. Page Layout
4. Validation Rules
5. Compact Layouts
6. Profiles
7. Roles
8. Users
9. Permission Sets
10. Triggers
11. Flows
12. Reports
13. Dashboards

Milestone 1-Salesforce Account

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?".



What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud. So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

<https://youtu.be/r9EX3lGde5k>

Activity 1: Creating Developer Account

Creating a developer org in salesforce.

Go to <https://developer.salesforce.com/signup>

On the sign up form, enter the following details :

First name & Last name

Email

Role : Developer

Company : College Name

County : India

Postal Code : pin code

Username : should be a combination of your name and company

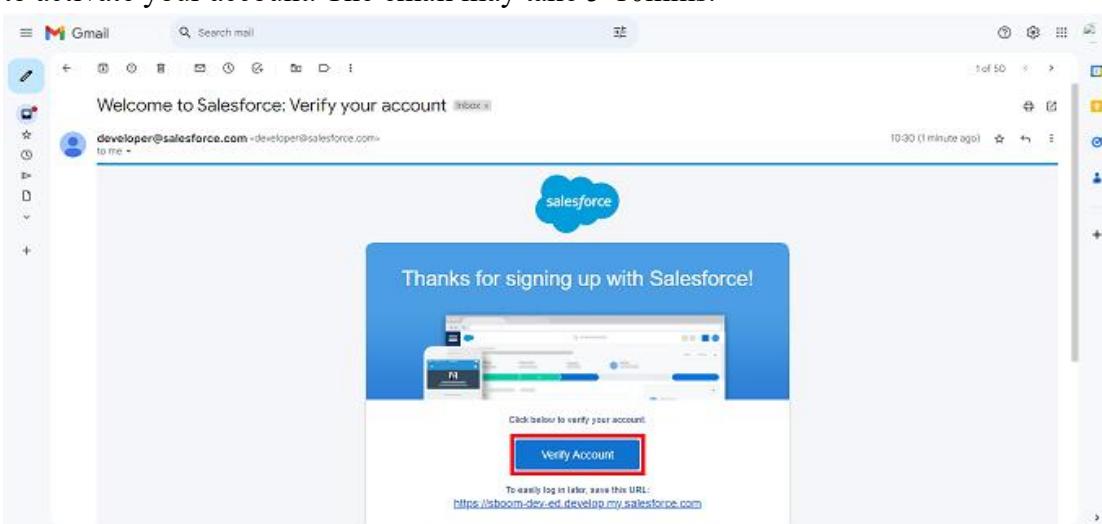
This need not be an actual email id, you can give anything in the format :

username@organization.com

Click on sign me up after filling these.

Activity 2: Account Activation

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.



Click on Verify Account

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

Change Your Password

Enter a new password for lead@sb.oom.
Make sure to include at least:

- 8 characters
- 1 letter
- 1 number

* New Password
..... Good

* Confirm New Password
..... Match

Security Question
In what city were you born?

* Answer
asdfghjkl

Change Password

Then you will redirect to your salesforce setup page.

Setup Home

Object Manager

SETUP Home

Get Started with Einstein Bots

Launch an AI-powered bot to automate your digital connections.

Get Started

Mobile Publisher

Use the Mobile Publisher to create your own branded mobile app.

Learn More

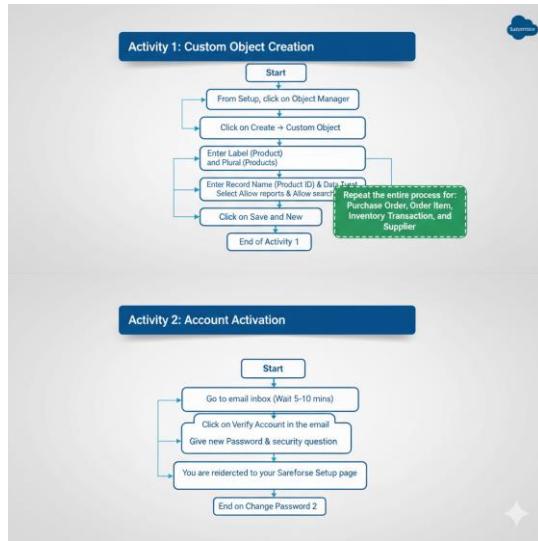
Real-time Collaborative Docs

Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce.

Create

Milestone 2- Objects

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



Activity 1: Creating a Product Object

To create an object:

From the setup page

Click on Object Manager

Click on Create >> Click on Custom Object.

Enter the label name as Product

Enter Plural label name as Products

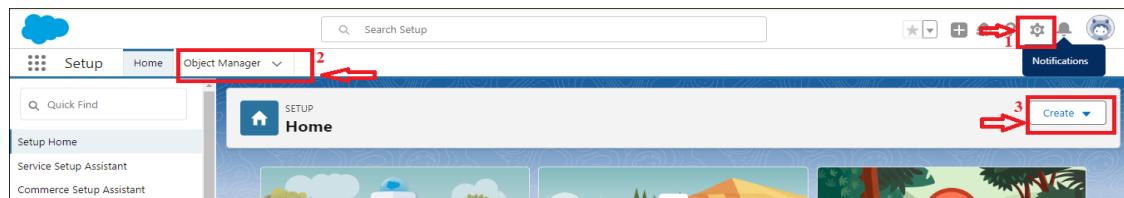
Enter Record Name as Product ID

Select Data Type as Text.

Select Allow reports.

Select Allow search.

Click on Save and New

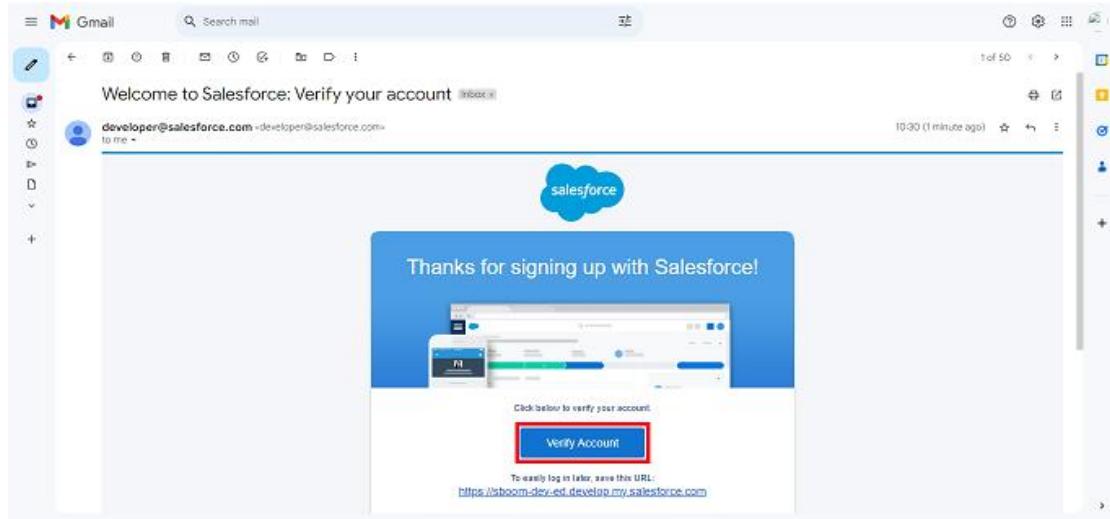


The screenshot shows the 'New Custom Object' setup page. The 'Label' field is set to 'Product' (4) and the 'Plural Label' field is set to 'Products' (5). The 'Object Name' field also contains 'Product'. The 'Record Name' field is set to 'Product ID' (6), and the 'Data Type' is 'Text' (7). Under 'Optional Features', 'Allow Reports' is checked (8). In the 'Deployment Status' section, 'Deployed' is selected. In the 'Search Status' section, 'Allow Search' is checked (9). At the bottom, there are 'Save', 'Save & New', and 'Cancel' buttons (10).

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

Activity 2: Account Activation

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.

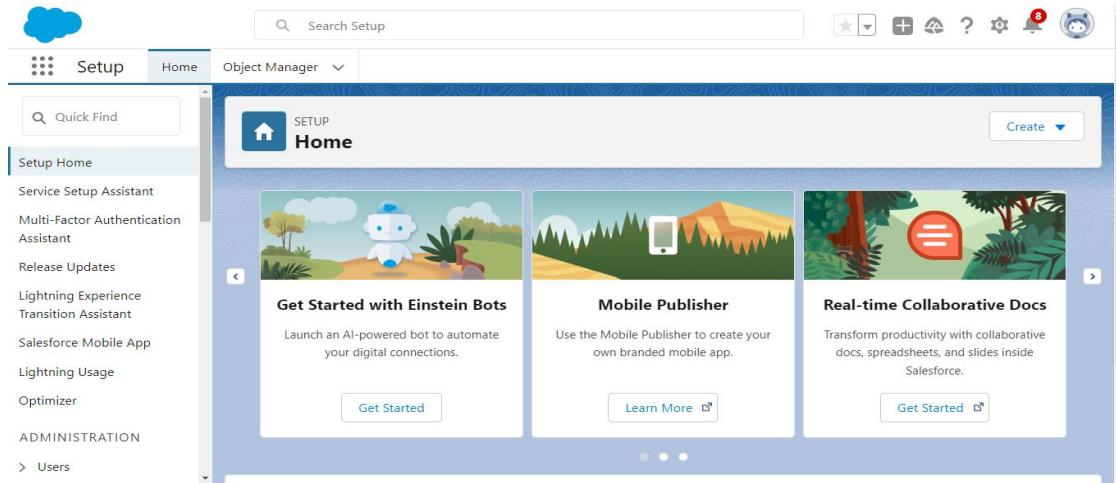


Click on Verify Account

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

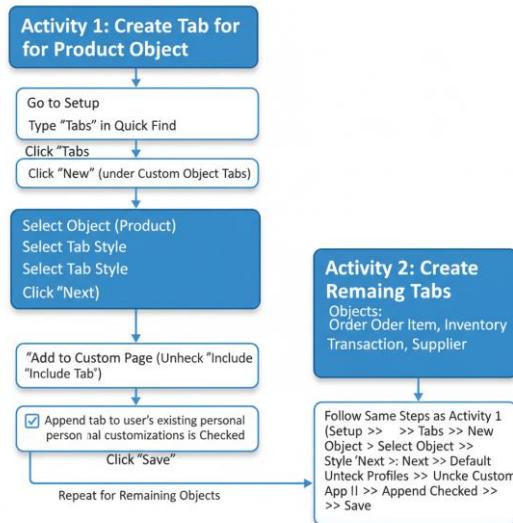
A screenshot of a "Change Your Password" form. The title is "Change Your Password". It asks for a new password for "lead@sb.oom" and specifies that it must include at least 8 characters, 1 letter, and 1 number. The "New Password" field contains "Good" and the "Confirm New Password" field contains "Match". A "Security Question" field is set to "In what city were you born?" and an "Answer" field contains "asdfghjk!". The "Change Password" button at the bottom is highlighted with a red box.

Then you will redirect to your salesforce setup page.



Milestone 3- 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.



Activity 1: Creating a tab for Product Object

Go to the setup page >> type Tabs in Quick Find bar

Click on tabs

Click on New (under custom object tab).

Select Object(Product) >> Select the tab style

Click on Next >> (Add to profiles page) keep it as default >> Click on Next (Add to Custom App) uncheck the include tab .

Make sure that the Append tab to user's existing personal customizations is checked.

Click save

The image consists of two vertically stacked screenshots of the Salesforce Setup interface.

Screenshot 1: Custom Tabs Overview

This screenshot shows the "Custom Tabs" section of the Setup page. A red box labeled "1" highlights the search bar at the top. A red box labeled "2" highlights the "Tabs" link in the sidebar under "User Interface". A red box labeled "3" highlights the "New" button in the "Custom Object Tabs" section.

Screenshot 2: Step 1 of 3 - Enter the Details

This screenshot shows the first step of a wizard for creating a new custom tab. A red box labeled "4" highlights the "Object" dropdown menu set to "Product". A red box labeled "5" highlights the "Next" button at the bottom right of the screen.

Activity 2: Creating Remaining Tabs

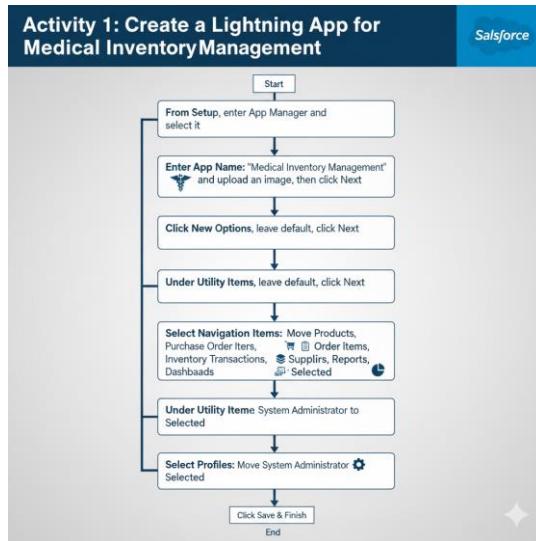
Now create the Tabs for the remaining Objects, they are “Purchase Order, Order Item, Inventory Transaction, Supplier”.

Follow the same steps as mentioned in Activity -1 .

Milestone 4- 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.



Activity 1: Create a Lightning App for Medical Inventory Management

From Setup, enter App Manager in the Quick Find and select App Manager.

Click New Lightning App.

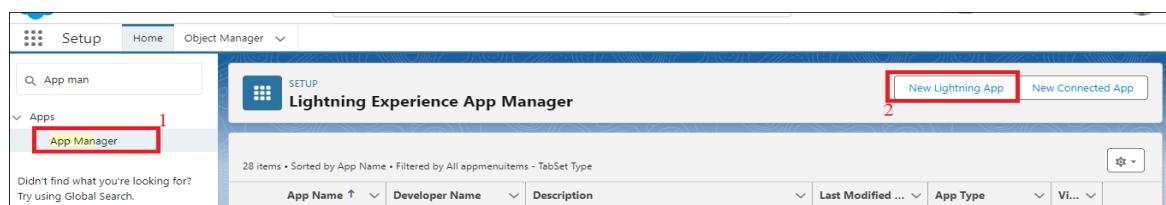
Enter Medical Inventory Management as the App Name >> Click on upload image and add an image related to Medical Inventory then click next

Under App Options, leave the default selections and click next.

Under Utility Items, leave as is and click Next.

From Available Items, select Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, Reports, and Dashboards and move them to Selected Item and Click Next.

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



New Lightning App

App Details

* App Name Medical Inventory Management 3

* Developer Name Medical_Inventory_Management

Description 1

App Branding

Image 1 3

Primary Color Hex
Value #0070D2

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

<input type="text" value="Dash"/>	Create ▼
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.

Available Profiles

<input type="text" value="system"/>	▶
System Administrator	◀

Selected Profiles

	Save & Finish
--	---------------

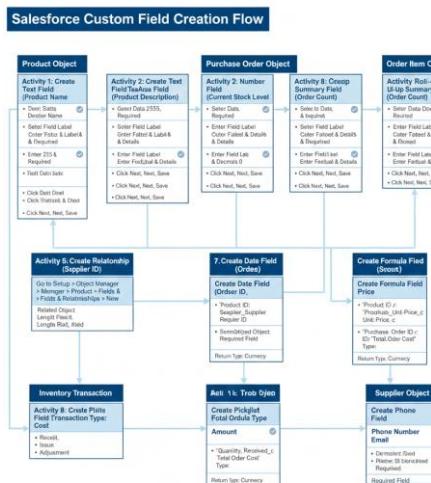
7

8

Milestone 5- Fields

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



Activity 1: Creating a Text Field in Product Object

To create fields in an object:

Click the gear icon and select Setup. This launches Setup in a new tab.

Click the Object Manager tab next to Home.

Select Product custom object.

Select Fields & Relationships from the left navigation

Click on New

Select Text field, click Next

Enter Field Label as “Product Name” and Length 255.

Select Required Field.

Click Next, Next, then Save & New.

The screenshot shows the Salesforce Object Manager. A red box highlights the 'Object Manager' tab in the top navigation bar. Another red box highlights the 'Product' row in the list, which has an API name of 'Product__c'. A third red box highlights the 'New' button at the bottom right of the list table.

The screenshot shows the 'Fields & Relationships' section for the Product object. A red box highlights the 'Fields & Relationships' tab in the left sidebar. Another red box highlights the 'New' button at the top right of the list table.

The screenshot shows the configuration for a new field. A red box highlights the 'Text' option in the list of field types. Another red box highlights the 'Next' button at the bottom right of the page.

Step 2. Enter the details

Field Label 7

Please enter the maximum length for a text field below.
Length 7

Field Name 7

Description

Help Text

Required Always require a value in this field in order to save a record 8

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 1

Step 2 of 4

Previous Next Cancel

9

Activity 2: Creating a TextArea Field in Product Object

To create fields in an object:

Click the gear icon and select Setup. This launches Setup in a new tab.

Click the Object Manager tab next to Home.

Select Product custom object.

Select Fields & Relationships from the left navigation

Click on New

Select TextArea field, click Next

Enter Field Label as “Product Description” .

Click Next, Next, then Save & New.

clicking Send an Email. Note that custom email addresses cannot be used for mass emails.

<input type="radio"/> Geolocation	Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.
<input type="radio"/> Number	Allows users to enter any number. Leading zeros are removed.
<input type="radio"/> Percent	Allows users to enter a percentage number, for example, '10' and automatically adds the percent sign to the number.
<input type="radio"/> Phone	Allows users to enter any phone number. Automatically formats it as a phone number.
<input type="radio"/> Picklist	Allows users to select a value from a list you define.
<input type="radio"/> Picklist (Multi-Select)	Allows users to select multiple values from a list you define.
<input type="radio"/> Text	Allows users to enter any combination of letters and numbers.
<input checked="" type="radio"/> Text Area	Allows users to enter up to 255 characters on separate lines. 6
<input type="radio"/> Text Area (Long)	Allows users to enter up to 131,072 characters on separate lines.
<input type="radio"/> Text Area (Rich)	Allows users to enter formatted text, add images and links. Up to 131,072 characters on separate lines.
<input type="radio"/> Text (Encrypted)	Allows users to enter any combination of letters and numbers and store them in encrypted form.
<input type="radio"/> 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.
<input type="radio"/> 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

Step 2. Enter the details

Step 2 of 4

Previous **Next** Cancel

Field Label	Product Description	7
Field Name	Product_Description	8
Description		
Help Text		
Required	<input type="checkbox"/> Always require a value in this field in order to save a record	
Auto add to custom report type	<input checked="" type="checkbox"/> Add this field to existing custom report types that contain this entity	
Default Value	Show Formula Editor	
<small>Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimal : (0.10), and express date calculations in the standard format : (Today() + 7). To reference a field from a Custom Metadata type record use: \$CustomMetadata.Type__mdt.RecordAPIName.Field__c</small>		

Activity 3: Creating a Number Field in Product object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product custom object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Number” and click Next.

Enter Field Label as “ Current Stock Level”.

Length - 18, Decimal Places - 0.

Click on Next, Next and Save.

Step 2. Enter the details

Step 2 of 4

Previous **Next** Cancel

Field Label	Current Stock Level	5
<small>Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".</small>		
Length	18	Decimal Places 0
Field Name	Current_Stock_Level	Number of digits to the right of the decimal point
Description		
Help Text		
Required	<input type="checkbox"/> Always require a value in this field in order to save a record	
Unique	<input type="checkbox"/> Do not allow duplicate values	
External ID	<input type="checkbox"/> Set this field as the unique record identifier from an external system	

Activity 4: Creating a Currency Field in Product object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product custom object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Currency” and click Next.

Enter Field Label as “ Unit Price”.

Length - 16, Decimal Places - 2.

Select Required Field.

Click on Next, Next and Save.

Step 2. Enter the details

Field Label 5

Please enter the length of the number and the number of decimal places. For example, a number with a length of 8 and 2 decimal places can accept values up to "12345678.90".

Length 6 Decimal Places

Number of digits to the left of the decimal point
Field Name 7 Number of digits to the right of the decimal point

Description

Help Text

Required Always require a value in this field in order to save a record 7

Auto add to custom report type Add this field to existing custom report types that contain this entity 8

Activity 5 : Creating Lookup Relationship in Purchase Order Object

A Lookup relationship is a type of relationship in Salesforce that connects two objects together based on a field known as the Lookup field. It establishes a relationship between a child object and a parent object, allowing the child object to reference the parent object.

To Create a relationship from Purchase Order to Supplier .

Go to the Setup page >> click on Object manager >> type object name(Purchase Order) in the quick find bar >> click on the Purchase Order object.

Click on Fields & Relationship

Click on New.

Select “Lookup relationship” as data type and click Next.

Select the related object “ Supplier”.

Click on Next.

Give Field Label as “Supplier ID” .

Select Required Field.

Click on Next , Next, Next , Save.

The image consists of three vertically stacked screenshots from the Salesforce Setup interface.

Screenshot 1: Data Type Selection

- Header: "Specify the type of information that the custom field will contain."
- Buttons: "Next" (highlighted with a red box) and "Cancel".
- Section: "Data Type"
 - None Selected
 - Auto Number
 - Formula
 - Roll-Up Summary
 - Lookup Relationship 4
 - Master-Detail Relationship
 - External Lookup Relationship
- Description for "Lookup Relationship": "Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list."

Screenshot 2: Step 2 - Choose the related object

- Header: "Purchase Order" and "New Relationship".
- Section: "Step 2. Choose the related object"
- Buttons: "Previous", "Next" (highlighted with a red box), and "Cancel".
- Description: "Select the other object to which this object is related."
- Input: "Related To" dropdown set to "Supplier" 5
- Buttons: "Previous", "Next" (highlighted with a red box), and "Cancel".

Screenshot 3: Field Definition

- Buttons: "Previous", "Next" (highlighted with a red box), and "Cancel".
- Fields:
 - Field Label: "Supplier ID" 7
 - Field Name: "Supplier_ID"
 - Description: (empty)
 - Help Text: (empty)
- Child Relationship Name: "Purchase_Orders" 9
- Required: Always require a value in this field in order to save a record 8
- What to do if the lookup record is deleted? Clear the value of this field. You can't choose this option if you make this field required. Don't allow deletion of the lookup record that's part of a lookup relationship.
- Auto add to custom report type: Add this field to existing custom report types that contain this entity

Activity 6: Creating a Date Field in Purchase Order object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Date” and click Next.

Enter Field Label as “ Order Date”.

Click on Next, Next and Save.

Step 2. Enter the details Step 2 of 4

Field Label 5

Field Name 6

Description

Help Text

Required Always require a value in this field in order to save a record

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

Default Value 7

Use formula syntax: Enclose text and picklist value API names in double quotes : ("the_text"), include numbers without quotes : (25), show percentages as decimals: (0.10), and express date calculations in the standard format: (Today) + 7. To reference a field from a Custom Metadata type record use: \$CustomMetadata.Type__mdt.RecordAPIName.Field__c

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

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Roll-Up Summary” and click Next.

Enter Field Label as “ Order Count”.

Choose the Summarized Object as “Order Items”.

For Select Roll-Up Type select “Count”.

Click on Next, Next and Save.

Data Type

None Selected

Auto Number

Formula

Roll-Up Summary 4

Select one of the data types below.

A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

Purchase Order
New Custom Field

Help for this Page

Step 2. Enter the details Step 2 of 5

Field Label 5

Field Name 6

Description

Help Text

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

Purchase Order
New Custom Field

Step 3. Define the summary calculation Step 3 of 5

Select Object to Summarize
Master Object Purchase Order
Summarized Object Order Items **6**

Select Roll-Up Type
 COUNT **7**
 SUM
 MIN
 MAX
Field to Aggregate —None—

Filter Criteria
 All records should be included in the calculation
 Only records meeting certain criteria should be included in the calculation

Required Information **8**

Previous Next Cancel

Activity 8: Creating a Unit Price Formula Field in Order Item object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Order Item) in quick find box >> click on the Order Item object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Formula” and click Next.

Enter field label Unit Price.

Select formula return type Currency, Click Next

Create and insert Advance formula: Product_ID__r.Unit_Price__c

Click Next, Next, then Save.

Step 2. Choose output type Step 2 of 5

Field Label Unit Price **5** Field Name Unit_Price

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

Formula Return Type
 None Selected Select one of the data types below.
 Currency **6**
 Checkbox Calculate a boolean value.
 Date Calculate a date, for example, by adding or subtracting days to other dates.
 Date/Time Calculate a date/time, for example, by adding a number of hours or days to another date/time.
 Number Calculate a numeric value.
 Percent Calculate a percent and automatically add the percent sign to the number.

Previous Next Cancel



Activity 9: Creating a Amount Formula Field in Order Item object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Order Item) in quick find box >> click on the Order Item object.

Now click on “Fields & Relationships”

Click on New.

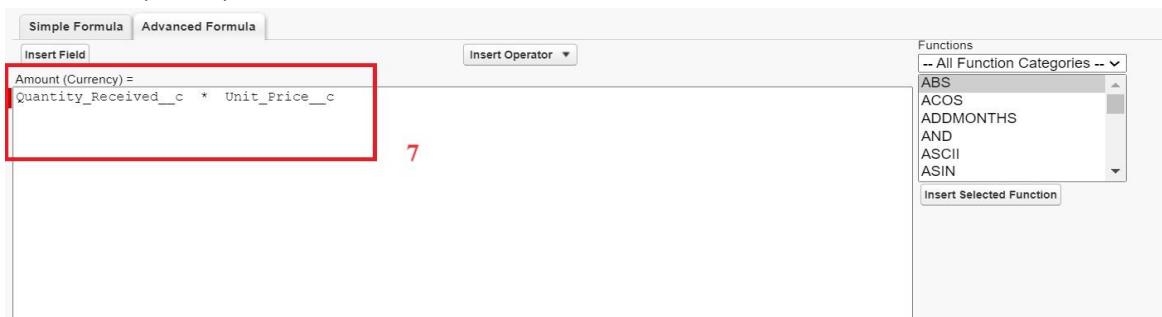
Select Data type as “Formula” and click Next.

Enter field label Amount.

Select formula return type Currency, Click Next

Create and insert Advance formula: Quantity_Received__c * Unit_Price__c

Click Next, Next, then Save.



Activity 10: Creating a Picklist Field in Inventory Transaction Object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Inventory Transaction) in quick find box>> click on the Inventory Transaction Object.

Now click on “Fields & Relationships” .

Click on New.

Select Data type as “Picklist” and click Next.

Enter Field Label as “Transaction Type”.

In values select “Enter values, with each value separated by a new line” and enter values as shown below.

Receipt

Issue

Adjustment

Click on Next, Next and Save.

The screenshot shows the 'Step 2. Enter the details' screen for creating a new field. The 'Field Label' is set to 'Transaction Type' (5). Under 'Values', the 'Enter values, with each value separated by a new line' option is selected (6), and the input field contains the values 'Receipt', 'Issue', and 'Adjustment'. The 'Next' button is highlighted with a red box (7).

Activity 11: Creating a Total Order Cost Formula Field in Inventory Transaction object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Inventory Transaction) in quick find box >> click on the Order Item object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Formula” and click Next.

Enter field label Total Order Cost.

Select formula return type Currency, Click Next

Create and insert Advance formula: Purchase_Order_ID__r.Total_Order_Cost__c

Click Next, Next, then Save.

Activity 12: Creating a Phone Field in Supplier object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box>> click on the Supplier object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Phone” and click Next.

Enter the Field Label as “ Phone Number”.

Select Required Field.
Click on Next, Next and Save.

Step 2. Enter the details

Field Label: **Phone Number** 5

Required: Always require a value in this field in order to save a record 6

Next 7

Activity 13: Creating a Email Field in Supplier object

To create fields in an object:

Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box>> click on the Supplier object.

Now click on “Fields & Relationships”

Click on New.

Select Data type as “Email” and click Next.

Enter the Field Label as “ Email”.

Click on Next, Next and Save.

Step 2. Enter the details

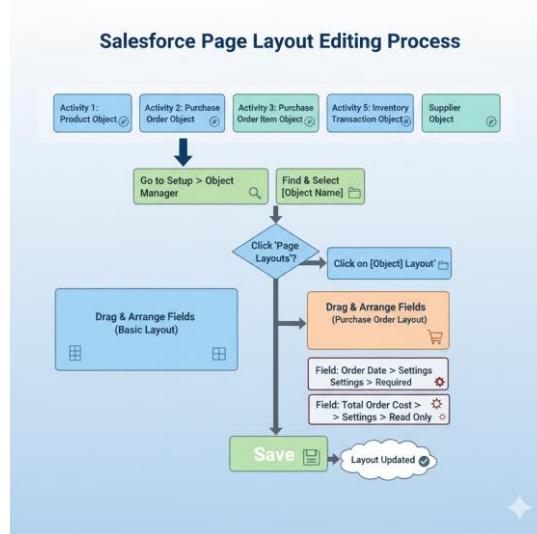
Field Label: **Email** 5

Required: Always require a value in this field in order to save a record 6

Next 7

Milestone 6 -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.



Activity 1: To edit a Page Layout in Product Object

Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object >> Page Layouts .

Click on the Product Layout.

Drag and Arrange the field as shown below.

The screenshot shows the 'Layout Properties' window for the Product object. The 'Fields' sidebar lists various actions like Buttons, Quick Actions, etc. The main area shows the layout structure with fields grouped into sections: 'Information' (Header visible on edit only) and 'System Information' (Header visible on edit only). The 'Information' section contains fields: Product ID (required), Product Name (required), and Product Description (optional). The 'System Information' section contains fields: Created By, Last Modified By, and Owner. A red box highlights the 'Information' section.

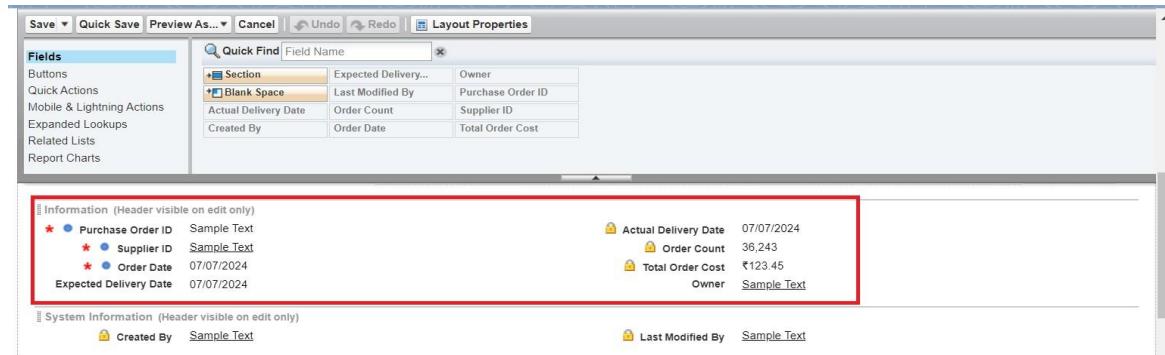
Click on Save.

Activity 2: To edit a Page Layout in Purchase Order Object

Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box >> click on the Purchase Order object >> Page Layouts.

Click on the Purchase Order Layout

Drag and Arrange the field as shown below



Click on field Order Date >> click on settings >> select Required and save it.

Click on field Total Order Cost >> click on settings >> select Read Only and save it.

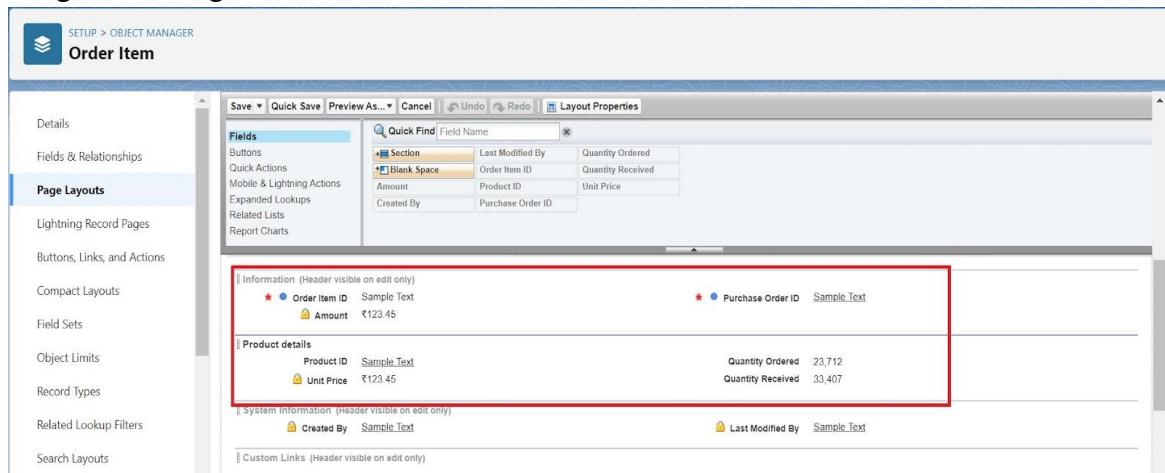
Click Save.

Activity 3: To edit a Page Layout in Order Item Object

Go to setup >> click on Object Manager >> type object name(Order Item) in quick find box >> click on the Order Item object >> Page Layouts.

Click on the Order Item Layout

Drag and Arrange the field as shown below



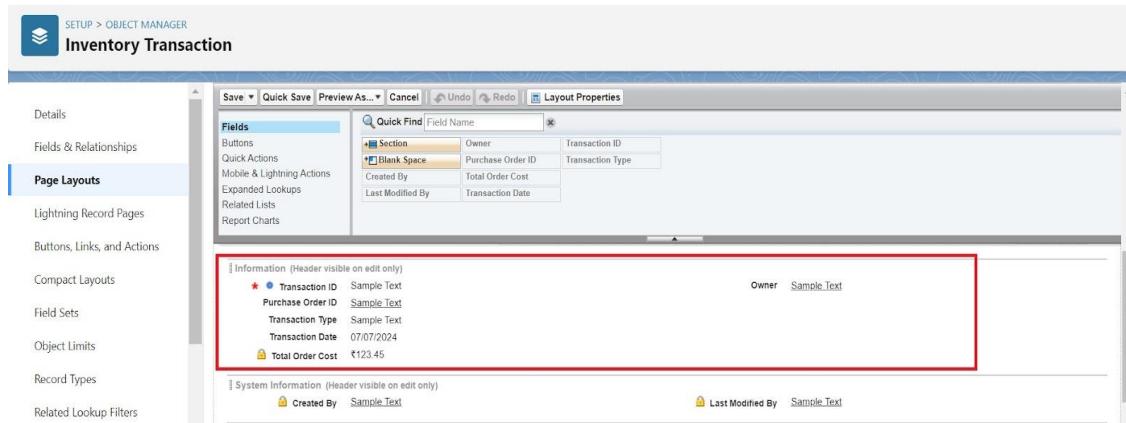
Click Save.

Activity 4: To edit a Page Layout in Inventory Transaction Object

Go to setup >> click on Object Manager >> type object name(Inventory Transaction) in quick find box >> click on the Inventory Transaction object >> Page Layouts.

Click on the Inventory Transaction Layout

Drag and Arrange the field as shown below



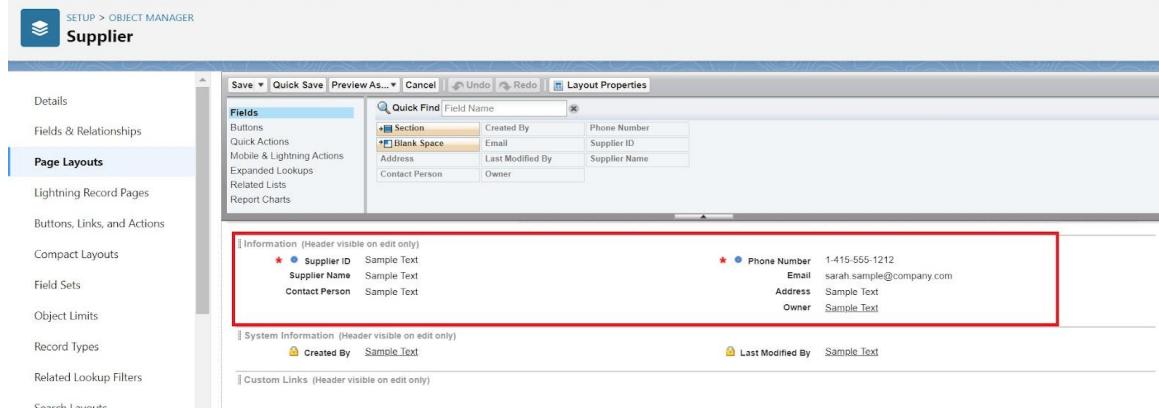
Click Save.

Activity 5: To edit a Page Layout in Supplier Object

Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box >> click on the Supplier object >> Page Layouts.

Click on the Supplier Layout

Drag and Arrange the field as shown below



Click Save.

Milestone 7 - Compact Layouts

Compact layouts display a record's key fields at a glance, providing important information quickly without needing to open the record.



Activity 1: To create a Compact Layout to a Product Object

Go to setup >> click on Object Manager >> type object name(Product) in quick find box
 >> click on the Product object

Click on Compact Layouts in the sidebar .

Click on New.

Enter the Label as “Product Compact Layout”.

Select the Compact Layout Fields : Select Product name, Unit Price, Current Stock Level.

Click Save.

Click Compact Layout Assignment.

Click Edit Assignment.

Choose "Product Compact Layout" from the dropdown.

Click Save.



Enter Compact Layout Information

Label: Product Compact Layout **4**
 Name: Product_Compact_Layout

Select Compact Layout Fields

Available Fields: Created By, Last Modified By, Minimum Stock Level, Owner, Product ID
 Selected Fields: Product Name, Unit Price, Current Stock Level **5**

Use SHIFT + click to select adjacent fields. Use CTRL + click to select an assortment of fields.

6 Save Cancel

Product Compact Layouts Compact Layout Assignment

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 **9**

10 Save Cancel

Activity 2: To create a Compact Layout to a Purchase Order Object

Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box >> click on the Purchase Order object

Click on Compact Layouts in the sidebar .

Click on New.

Enter the Label as “Purchase Order Compact Layout”.

Select the Compact Layout Fields : Select Purchase Order ID, Order Date, Total Order Cost, Supplier ID.

Click Save.

Click Compact Layout Assignment.

Click Edit Assignment.

Choose "Purchase Order Compact Layout" from the dropdown.
Click Save.

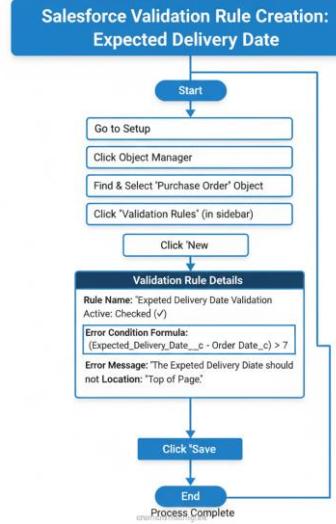
The screenshot shows the 'Compact Layout Edit' window. In the 'Enter Compact Layout Information' section, the 'Label' field is set to 'Purchase Order Compact L' (step 4). In the 'Select Compact Layout Fields' section, several fields are listed under 'Available Fields': Actual Delivery Date, Created By, Expected Delivery Date, Last Modified By, Owner, and Order Count. Under 'Selected Fields', 'Purchase Order ID', 'Order Date', 'Total Order Cost', and 'Supplier ID' are listed. A red box highlights the 'Selected Fields' list, and a red number '5' is placed to its right. Below the lists is a note: 'Use SHIFT + click to select adjacent fields. Use CTRL + click to select an assortment of fields.' At the bottom, there are 'Save' and 'Cancel' buttons, with 'Save' highlighted by a red box and a red number '6' to its left.

Purchase Order Compact Layouts Compact Layout Assignment

The screenshot shows the 'Compact Layout Assignment' window. It asks to 'Select the compact layout to use when this object's records appear as list items in the mobile app.' A dropdown menu labeled 'Primary Compact Layout' contains 'Purchase Order Compact Layout' (step 9). At the bottom are 'Save' and 'Cancel' buttons, with 'Save' highlighted by a red box and a red number '10' to its left.

Milestone 8 - 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.



Activity 1: To create an Expected Delivery Date Validation rule to a Employee Object

Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object

Click on the validation rule >> click on New.

Enter the Rule name as “Expected Delivery Date Validation”.

Select Active

Insert the Error Condition Formula as :

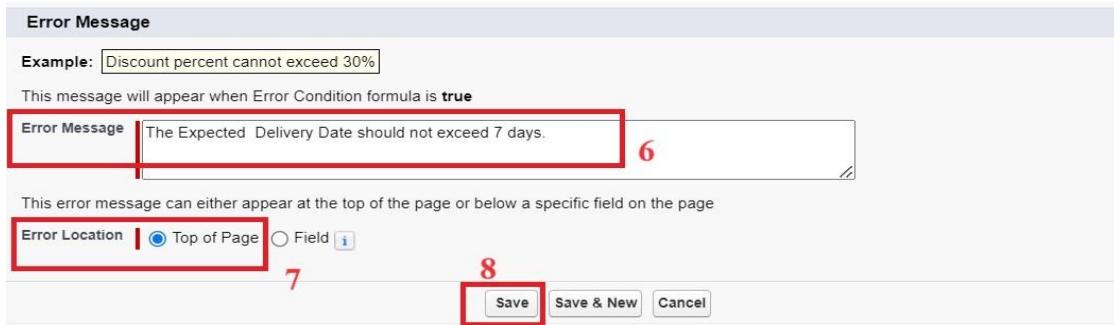
$(\text{Expected_Delivery_Date_c} - \text{Order_Date_c}) > 7$

The screenshot shows the 'Validation Rule Edit' page. The 'Rule Name' field is filled with 'Expected_Delivery_Date_Validation' (labeled 3). The 'Active' checkbox is checked (labeled 4). The 'Error Condition Formula' field contains the formula $(\text{Expected_Delivery_Date_c} - \text{Order_Date_c}) > 7$ (labeled 5). The 'Functions' dropdown menu is open, showing options like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. A 'Quick Tips' box is visible on the right, and a note at the bottom right indicates that fields marked with a red asterisk are required.

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

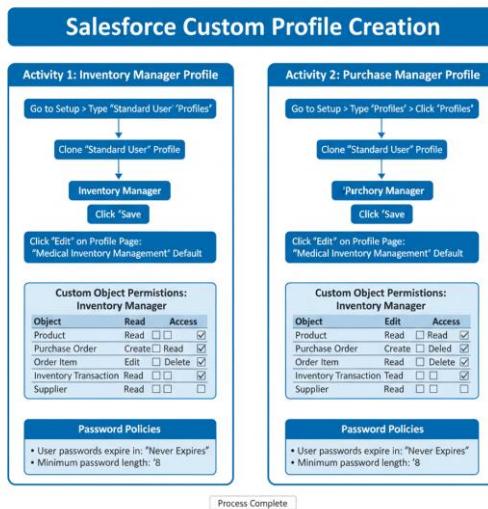
Select the Error location as Top of Page

Click Save.



Milestone 9 - 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.



Activity 1: To create an Inventory Manager Profile

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

Clone Profile

Enter the name of the new profile.

You must select an existing profile to clone from.

Existing Profile	Standard User
User License	Salesforce
Profile Name	<input type="text" value="Inventory Manager"/>

While still on the profile page, then click Edit.

Select the Custom App settings as default for the Medical Inventory Management.

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

Change the password policies as mentioned :

User passwords expire in should be “ never expires ”.

Minimum password length should be “ 8 ”, and click save.

Password Policies

User passwords expire in	Never expires
Enforce password history	3 passwords remembered
Minimum password length	8
Password complexity requirement	Must include alpha and numeric characters
Password question requirement	Cannot contain password
Maximum invalid login attempts	10
Lockout effective period	15 minutes
Obscure secret answer for password resets	<input type="checkbox"/>
Require a minimum 1 day password lifetime	<input type="checkbox"/>
Don't immediately expire links in forgot password emails	<input type="checkbox"/> 

Activity 2: To create an Purchase Manager Profile

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

While still on the profile page, then click Edit.

Select the Custom App settings as default for the Medical Inventory Management.

SETUP Profiles

Set the permissions and page layouts for this profile.

Profile Edit		Save	Save & New	Cancel	
Name	Purchase Manager				
User License	Salesforce				
Description					
Custom Profile <input checked="" type="checkbox"/>					
Custom App Settings Required information					
	Visible	Default	Visible	Default	
All Tabs (standard__AllTabSet)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sales (standard__LightningSales)	<input checked="" type="checkbox"/>	<input type="radio"/>
Analytics Studio (standard__Insights)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sales (standard__Sales)	<input checked="" type="checkbox"/>	<input type="radio"/>
App Launcher (standard__AppLauncher)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sales Console (standard__LightningSalesConsole)	<input checked="" type="checkbox"/>	<input type="radio"/>
Bolt Solutions (standard__LightningBolt)	<input checked="" type="checkbox"/>	<input type="radio"/>	Salesforce Chatter (standard__Chatter)	<input checked="" type="checkbox"/>	<input type="radio"/>
Community (standard__Community)	<input checked="" type="checkbox"/>	<input type="radio"/>	Salesforce Scheduler Setup (standard__LightningScheduler)	<input type="checkbox"/>	<input type="radio"/>
Content (standard__Content)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sample Console (standard__ServiceConsole)	<input type="checkbox"/>	<input type="radio"/>
Data Manager (standard__DataManager)	<input checked="" type="checkbox"/>	<input type="radio"/>	Service (standard__Service)	<input checked="" type="checkbox"/>	<input type="radio"/>
Digital Experiences (standard__SalesforceCMS)	<input checked="" type="checkbox"/>	<input type="radio"/>	Service Console (standard__LightningService)	<input checked="" type="checkbox"/>	<input type="radio"/>
Lightning Usage App (standard__LightningUsage)	<input checked="" type="checkbox"/>	<input type="radio"/>	Site.com (standard__Sites)	<input checked="" type="checkbox"/>	<input type="radio"/>
Marketing CRM Classic (standard__Marketing)	<input checked="" type="checkbox"/>	<input type="radio"/>	Subscription Management (standard__RevenueCloudConsole)	<input checked="" type="checkbox"/>	<input type="radio"/>
Medical Inventory Management (Medical_Inventory_Management)	<input type="checkbox"/>	<input checked="" type="radio"/>	WDC (standard__Work)	<input checked="" type="checkbox"/>	<input type="radio"/>
Queue Management (standard__QueueManagement)	<input checked="" type="checkbox"/>	<input type="radio"/>			

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

Custom Object Permissions										
	Basic Access				Data Administration					
	Read	Create	Edit	Delete	View All	Modify All	Read	Create	Edit	Delete
Inventory Transactions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Order Items	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Products	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Purchase Orders	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Suppliers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Change the password policies as mentioned :

User passwords expire in should be “ never expires ”.

Minimum password length should be “ 8 ”, and click save.

The screenshot shows the 'Password Policies' configuration page. The 'User passwords expire in' dropdown is set to 'Never expires' (highlighted with a red box). The 'Minimum password length' input field is set to '8'. At the bottom, the 'Save' button is highlighted with a red box.

Milestone 10 - 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.

Salesforce Custom Role Creation



Process Complete

Activity 1 : Create a Purchasing Manager Role.

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

The screenshot shows the Salesforce Setup interface. On the left, there's a navigation sidebar with links like 'Setup', 'Home', 'Object Manager', 'Users', 'Roles' (which is highlighted), and various 'Feature Settings' and 'Case Teams' sections. The main content area is titled 'SETUP Roles' and contains a diagram illustrating a role hierarchy. At the top of the hierarchy is 'Executive Staff' (CEO, President, CFO, VP Sales). Below them are 'Western Sales Director' (CA Sales Rep, OR Sales Rep) and 'Eastern Sales Director' (NY Sales Rep, MI Sales Rep). Further down is 'International Sales Director' (Asian Sales Rep, European Sales Rep). The 'Set Up Roles' button is located at the bottom right of the main content area.

Click on Expand All and click on add role under SVP, Sales & Marketing role.

Give Label as “Purchasing Manager” and Role name gets auto populated. Then click on Save.

Role Edit
New Role

Role Edit

Label	Purchasing Manager
Role Name	Purchasing_Manager
This role reports to	SVP, Sales & Marketing
Role Name as displayed on reports	

Save Save & New Cancel

Activity 2 : Create a Purchasing Manager Role.

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

Setup Home Object Manager

roles

Users Roles

Feature Settings Sales Contact Roles on Contracts Contact Roles on Opportunities Service Case Teams Case Team Roles Contact Roles on Cases

Didn't find what you're looking for? Try using Global Search.

Help for this Page

SETUP Roles

Understanding Roles

Set up your Role Hierarchy to control how your organization reports on and accesses data.

Sample Role Hierarchy
View other sample Role Hierarchies: Territory-based Sample

```

graph TD
    CEO[CEO - President] --> CFO[CFO - VP, Sales]
    CEO --> CIO[CIO - VP, IT]
    CFO --> WesternSales[Western Sales Director]
    CFO --> EasternSales[Eastern Sales Director]
    CIO --> InternationalDir[International Director]
    WesternSales --> WestRep1[West Rep]
    WesternSales --> WestRep2[East Sales Rep]
    EasternSales --> EastRep1[NY Sales Rep]
    EasternSales --> EastRep2[NA Sales Rep]
    InternationalDir --> EuroRep1[European Sales Rep]
    InternationalDir --> EuroRep2[Asian Sales Rep]
  
```

* View & edit data, roll up forecasts, & generate reports
* View & edit data, & roll up forecasts directly for all users directly beneath them
* Can't access data of other Executive Staff members

* View & edit data, & roll up forecasts directly for all users directly beneath them
* Can't access data of other Sales Director members

* View & edit data, & roll up forecasts directly for all users directly beneath them
* Can't access data of other Sales Director members

* View & edit data, & roll up forecasts directly for all users directly beneath them
* Can't access data of other Sales Director members

Set Up Roles

Don't show this page again

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

Role Edit
New Role

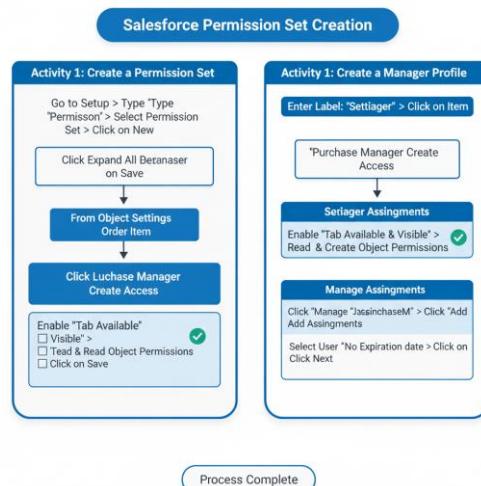
Role Edit

Label	Inventory Manager
Role Name	Inventory_Manager
This role reports to	SVP, Sales & Marketing
Role Name as displayed on reports	

Save **Save & New** **Cancel**

Milestone 12 - 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.



Activity 1 : Create a Permission Set.

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

The screenshot shows the 'Permission Sets' page in the Salesforce Setup. The left sidebar has sections for 'Users' (with 'Permission Sets' highlighted), 'Permission Set Groups', and 'Custom Code'. The main area is titled 'Permission Sets' and contains a list of existing permission sets like 'Buyer', 'CRM User', and 'Commerce Admin'. At the top right, there's a 'New' button, which is also highlighted with a red box. The bottom right corner shows 'Page 1 of 2'.

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

The screenshot shows the 'Permission Set Create' page. It has fields for 'Label' (set to 'Purchase Manager Create Access'), 'API Name' (set to 'Purchase_Manager'), and 'Description'. There's also a 'Session Activation Required' checkbox. The 'Save' button is highlighted with a red box at the top right of the form.

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

The screenshot shows the 'Purchase Manager Create Access' object settings. Under 'Tab Settings', the 'Available' and 'Visible' checkboxes are checked and highlighted with a red box. Under 'Object Permissions', the 'Read' and 'Create' checkboxes are checked and highlighted with a red box. The 'Save' button is visible at the top right of the form.

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.

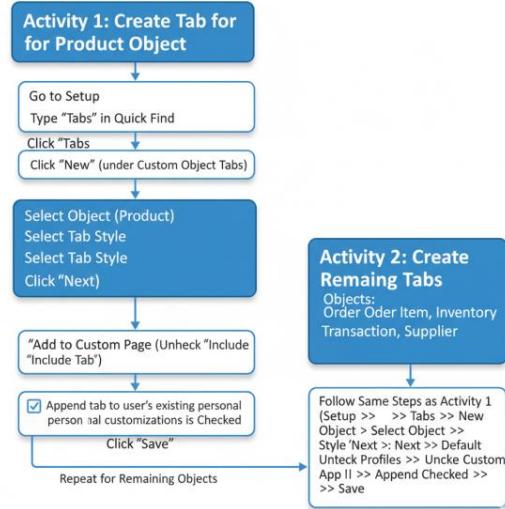
The screenshot shows the 'Purchase Manager Create Access' page. In the 'Select Users to Assign' section, a user named 'John PurchaseM' is selected, indicated by a checked checkbox and highlighted with a red box. The 'Next' button at the bottom right is also highlighted with a red box.

Select No Expiration date >> Click on Assign.

The screenshot shows the 'Purchase Manager Create Access' page. In the 'Select an Expiration Option For Assigned Users' section, the 'No expiration date' radio button is selected and highlighted with a red box. The 'Assign' button at the bottom right is also highlighted with a red box.

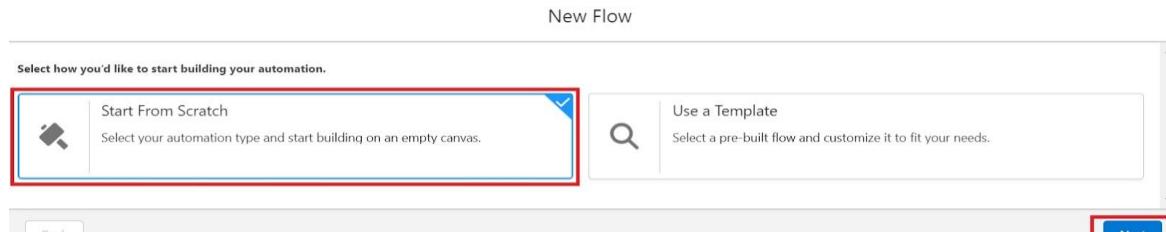
Milestone 13 - Flows

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.

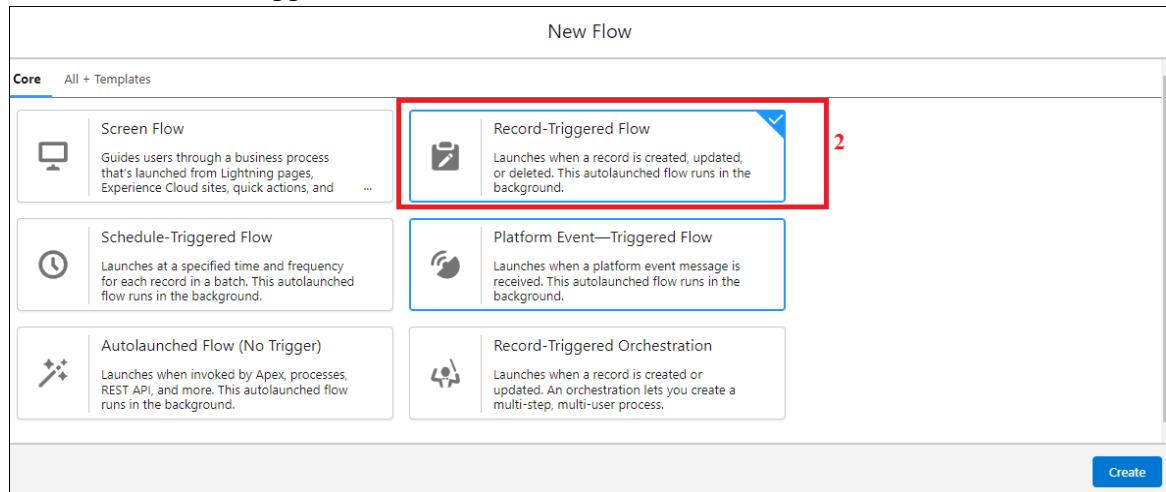


Activity 1 : Create Flow to update the Actual Delivery Date.

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

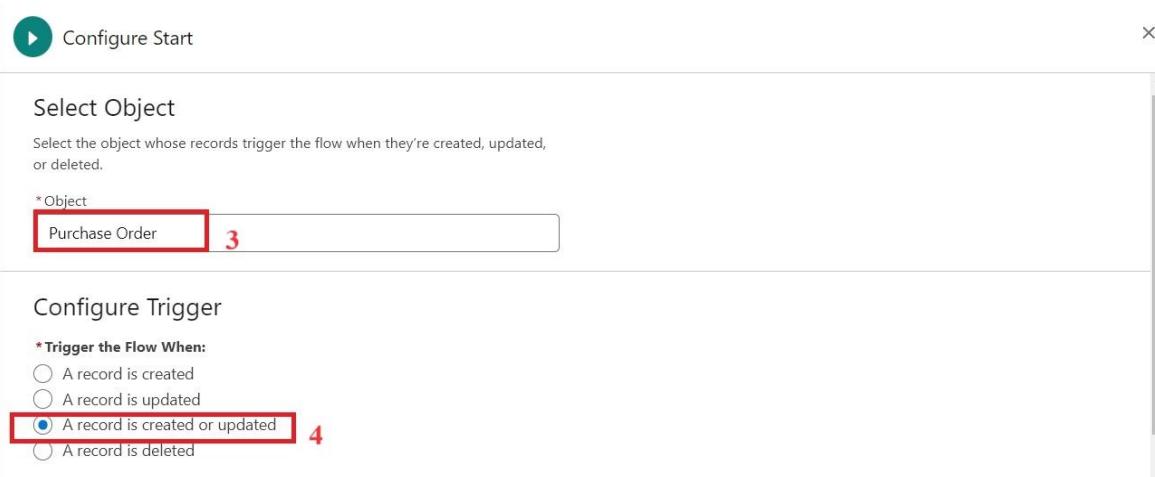


Select the record Triggered flow.Click on create.



Under Object select “Purchase Order”

Select A record is created or updated



Set Entry Conditions : None

Select Fast Field Updates and click on Done

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

None 5

* Optimize the Flow for:

Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs *before* the record is saved to the database.

6

Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.

Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed

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

Enter Label as “ Get Purchase Record ”.

For Object select Purchase Order.

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

For the first condition select as follows:

Field: Id

Operator: Equals

Value: {!\$Record.Id}

Get Records

*Label: Get Purchase Record (8)

*API Name: Get_Purchase_Record

Description:

Get Records of This Object

*Object: Purchase Order (9)

Filter Purchase Order Records (10)

Condition Requirements: All Conditions Are Met (AND)

Field	Operator	Value
Id	Equals	\$Record > Record ID

+ Add Condition

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

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

How Many Records to Store

Only the first record

All records

How to Store Record Data

Automatically store all fields

Choose fields and let Salesforce do the rest

Choose fields and assign variables (advanced)

Select Purchase Order Fields to Store in Variable

Field	
ID	
Order_Date_c	[Delete]

+ Add Field

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.

Enter API name as ActualDeliveryDate >> Select Data type as Date >> Click on Done. From the Toolbox drag and drop Assignment element.

Enter the label as “Assignment”.

Set Variable Values:

a) Variable : {!ActualDeliveryDate}

Operator : Equals

Value : {!\$Record.Order_Date_c}

b) Variable : {!ActualDeliveryDate}
Operator : Add
Value : 3

* Label * API Name
Description
Set Variable Values
Each variable is modified by the operator and value combination.
Variable Operator Value
ActualDeliveryDate Equals \$Record > Order Date
Variable Operator Value
ActualDeliveryDate Add 3
+ Add Assignment

Click Done

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

Enter the label as “Updating Purchasing Order”.

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

Set Filter Conditions : None -Always Update Record

Set Field Values for the Trip Record as

Field : Actual_Delivery_Date__c

Value : {!ActualDeliveryDate}

Milestone 14 – 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.

Activity 1 : Create a Trigger to Calculate total amount on Order Item.

Step 1 : Login to Salesforce:

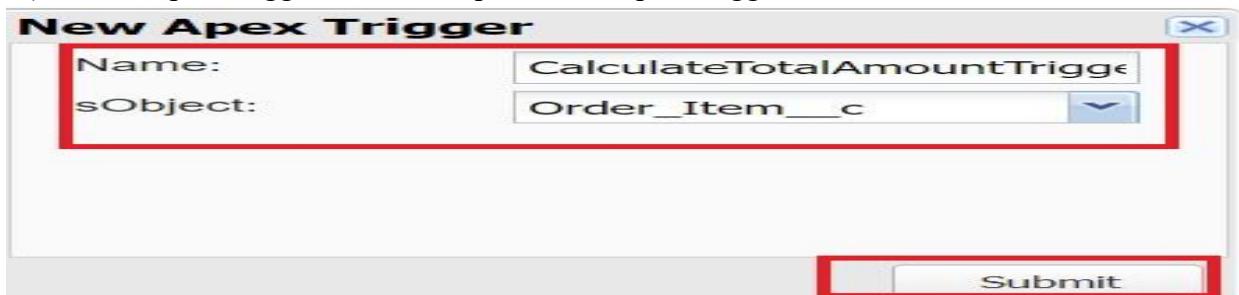
Log in to your Salesforce account with administrative privileges.

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.

```

Update Records

*** How to Find Records to Update and Set Their Values**

- Use the purchase order record that triggered the flow
- Update records related to the purchase order record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

i Because this flow runs *before* a record is saved, you can only update the record that triggered the flow to run. To update other records, configure the trigger to run the flow *after* the record is saved.

Set Filter Conditions

Condition Requirements to Update Record

None—Always Update Record

Set Field Values for the Purchase Order Record

Field	Value
Actual_Delivery_Date__c	ActualDeliveryDate

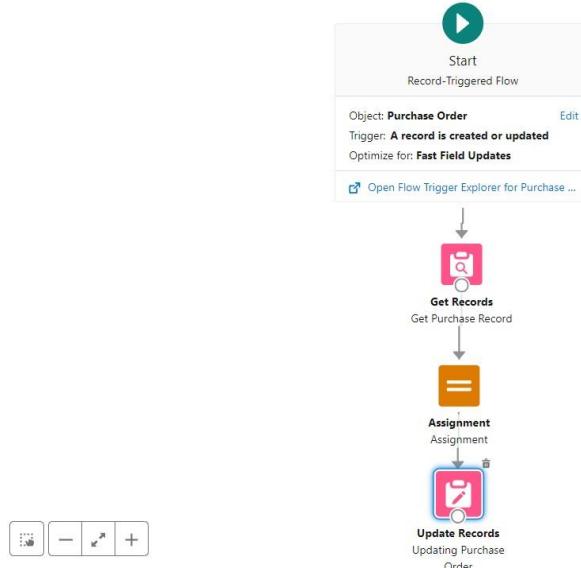
+ Add Field

Click Done

Save the flow as “Actual Delivery Date Updating”.

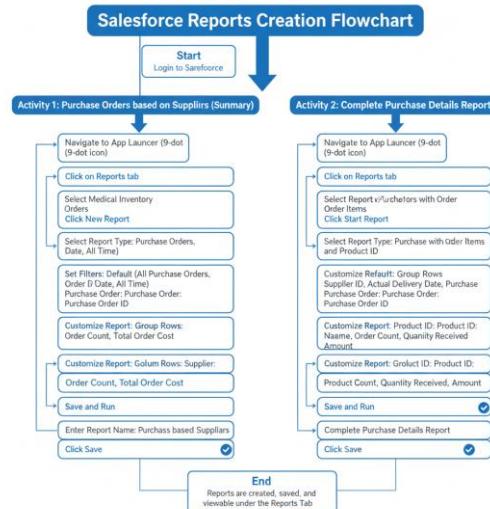
Activate the flow.

Free-Form ▾ Version 7: Last modified 17 days ago **Active** Run Debug View Tests Deactivate Save As Save



Milestone 15 - 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.



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

Report

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

6. Click on Filters and select as follows and click on Apply

7. Customize your report, in group rows select – Supplier ID, Purchase Order: Purchase Order ID, for columns Order Count, Total Order Cost (In this way we are making a Summary Report).

8. Click save and run

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

10. Click Save

NOTE: In this report you can see your all record of the object you selected for reporting (What you selects in “Select a report type option”)

Purchase Orders based on Suppliers

REPORT ▾ Purchase Orders

Fields

Groups

Supplier ID Purchase Order: Purchase Order ID Order Count Total Order Cost

Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
Supplier-001 (4)	Purchase-0001 (1) Purchase-0002 (1) Purchase-0003 (1) Purchase-0004 (1)	3 2 3 4	₹2,075.00 ₹3,250.00 ₹7,000.00 ₹9,500.00
Supplier-002 (1)	Purchase-0005 (1)	2	₹4,500.00
Total (5)		14	₹26,325.00

Columns

Add column... # Order Count # Total Order Cost

Row Counts Detail Rows Subtotals Grand Total Conditional Formatting

Save & Run Update Preview Automatically

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.

Report: Purchase Orders

Purchase Orders based on Suppliers

Total Records: 5 Total Order Count: 14 Total Total Order Cost: ₹26,325.00

Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
Supplier-001 (4)	Purchase-0001 (1) Purchase-0002 (1) Purchase-0003 (1) Purchase-0004 (1)	3 2 3 4	₹2,075.00 ₹3,250.00 ₹7,000.00 ₹9,500.00
Supplier-002 (1)	Purchase-0005 (1)	2	₹4,500.00
Total (5)		14	₹26,325.00

Row Counts Detail Rows Subtotals Grand Total

Activity 2: Create a Complete Purchase Details Report

1. Click App Launcher
2. Select Medical Inventory Management App

3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders with Order Items and Product ID >> Click Start report.
6. Click on Filters and select as follows and click on Apply

Filters

Add filter...

Show Me
All purchase orders

Actual Delivery Date
All Time

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

REPORT ▾

Complete Purchase Details Report ▾ Purchase Orders with Order Items and Product ID

Fields

Groups

- Supplier ID (2)
- Actual Delivery Date (2)
- Purchase Order: Purchase Order ID (2)

Columns

- Add column...
- Product ID: Product ID (X)
- # Order Count (X)
- Product ID: Product Name (X)
- # Quantity Received (X)
- # Amount (X)

Outline ▾ Filters

Previewing a limited number of records. Run the report to see everything.

Supplier ID	Actual Delivery Date	Purchase Order: Purchase Order ID	Product ID: Product ID	Order Count	Product ID: Product Name	Quantity Received	Amount
Supplier-001 (12)	19/06/2024 (2)	Purchase-0002 (2)	Gen-0001	2	Syringes	50	\$250.00
			Cap-0001	2	Dolo 650	150	\$3,000.00
				Subtotal		200	\$3,250.00
		22/06/2024 (3)	Purchase-0001 (3)			200	\$3,250.00
			Gen-0001	3	Syringes	5	\$25.00
			Gen-0001	3	Syringes	10	\$50.00
			Cap-0001	3	Dolo 650	100	\$2,000.00
				Subtotal		115	\$2,075.00
		23/06/2024 (3)	Purchase-0003 (3)			115	\$2,075.00
			Syr-0001	3	Calpol 120mg Syrup	100	\$4,000.00
			Cap-0001	3	Dolo 650	50	\$1,000.00
			Gen-0001	3	Syringes	400	\$2,000.00
				Subtotal		550	\$7,000.00
		11/07/2024 (4)	Purchase-0004 (4)			550	\$7,000.00
			Syr-0001	4	Calpol 120mg Syrup	100	\$4,000.00
			IV-0001	4	Saline	50	\$2,500.00
			Cap-0001	4	Dolo 650	100	\$2,000.00
				Subtotal		300	\$1,000.00

Save & Run

Save

Close

Run

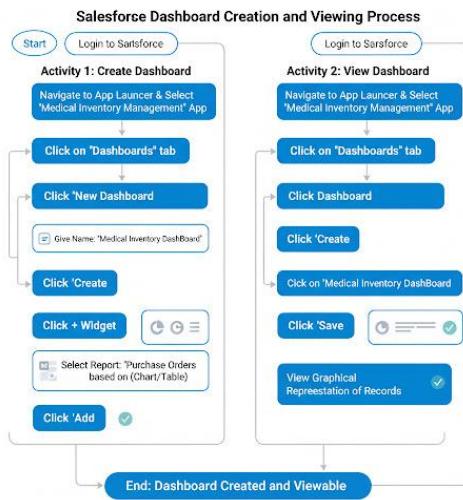
Update Preview Automatically

Row Counts Detail Rows Subtotals Grand Total

Conditional Formatting

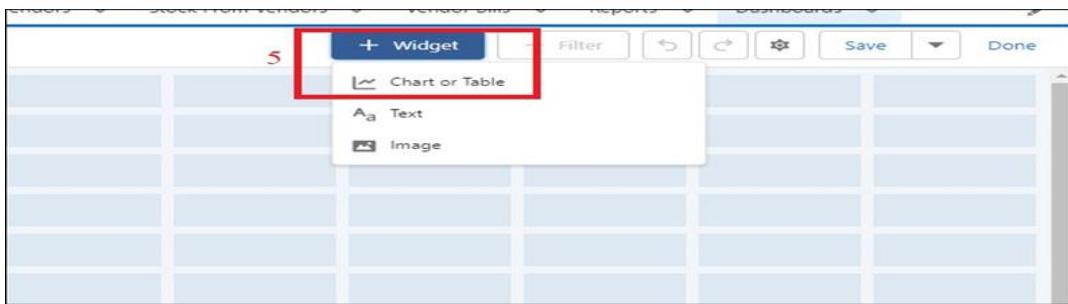
Milestone 16 - Dashboards

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.



Activity 1: - 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.



Select Report

Reports

- Recent:
- Created by Me
- Private Reports
- Public Reports
- All Reports

Folders

- Created by Me
- Shared with Me
- All Folders

Select Report

Q. Search Reports and Folders. Reports and Folders ▾

All Folders > Private Reports

Complete Purchase Details Report
Annapurna Gurram - 08-Jul-2024, 11:58 am - Private Reports

Purchase Orders based on Suppliers
Annapurna Gurram - 08-Jul-2024, 11:32 am - Private Reports

Cancel Select

A red box highlights the 'Purchase Orders based on Suppliers' report entry.

Add Widget

Report

Purchase Orders based on Supplier

Use chart settings from report i

Display As

Value

Sum of Total Order Cost

Sliced By

Supplier ID

Preview

Purchase Orders based on Suppliers

Sum of Total Order Cost

Supplier ID

Supplier-001

Supplier-002

₹26k

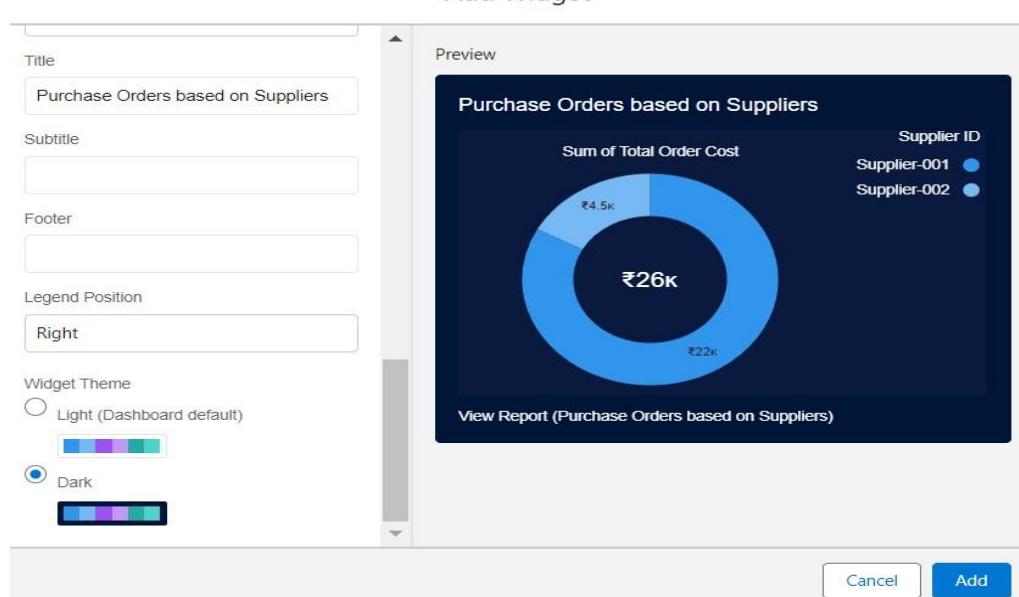
₹4.5k

₹22k

View Report (Purchase Orders based on Suppliers)

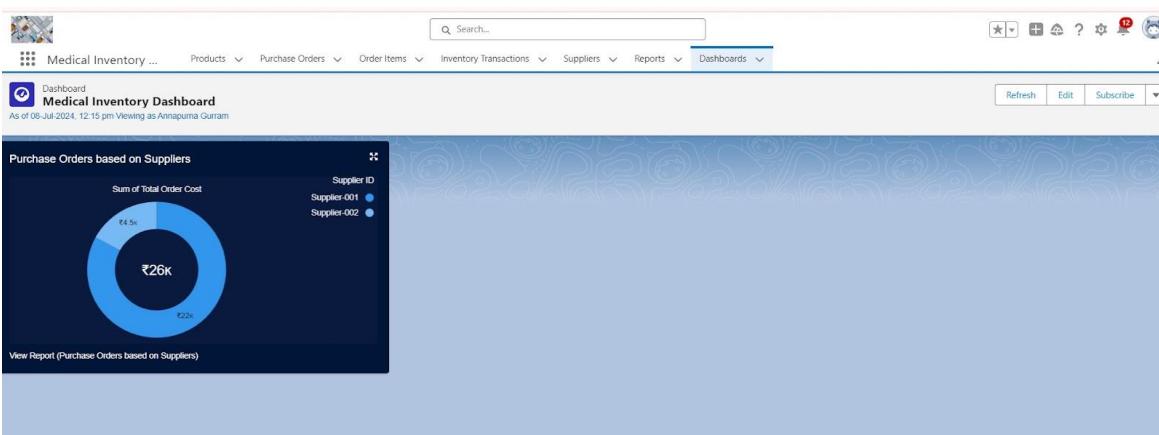
Cancel Add

A red box highlights the preview section of the 'Purchase Orders based on Suppliers' report.



Activity 2: 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



Conclusion:

The **Medical Inventory Management System** project successfully demonstrates the design and implementation of a comprehensive Salesforce application tailored for healthcare inventory operations. Through this project, we streamlined key business processes such as supplier management, purchase order tracking, product cataloging, and expiry monitoring—all within a single, integrated platform.

By leveraging Salesforce's powerful features—**Objects, Relationships, Page Layouts, Validation Rules, Profiles, Roles, Permission Sets, Flows, Triggers, Reports, and Dashboards**—we built a robust and user-friendly system that enhances efficiency, transparency, and data accuracy.

The generated **Reports and Dashboards** provide valuable real-time insights into supplier performance, stock levels, and purchasing trends, empowering decision-makers with actionable data. Overall, this project bridges the gap between manual inventory management and digital automation, ensuring smarter resource handling and improved operational control in medical environments.

In essence, the project not only strengthened our understanding of **Salesforce's declarative and programmatic tools** but also showcased how CRM platforms can be customized to solve real-world business problems in the healthcare industry.