

NAAN MUDHALVAN SKILL DEVELOPMENT PROGRAM

SB8067-SALESFORCE DEVELOPER

A SKILL TRAINING REPORT

Submitted By

SUBBULAKSHMI S	952422104046
SWATHI S	952422104050
ESAIVANI S	952422104008
SUBA LAKSHMI G	952422104045

in recognition of the completion of the Skill Training course as part of the curriculum
during Semester Seven

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING**

**PSN INSTITUTE OF TECHNOLOGY & SCIENCE,
TIRUNELVELI**

ANNA UNIVERSITY : CHENNAI – 600025

NOVEMBER - 2025

MEDICAL INVENTORY MANAGEMENT

PROJECT REPORT

Medical Inventory Management

User Story: The MedicalInventoryManagementSystemisacomprehensive 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.

ProjectFlow:

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 & Pthaegire Lraeylaotuiot n4s.h iVp alidna tiSoanl eRsfuolercse
3. 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, analyse, 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.

1. Go to <https://developer.salesforce.com/signup>
2. On the sign up form, enter the following details:
 1. First name & Last name
 2. Email
 3. Role: Developer

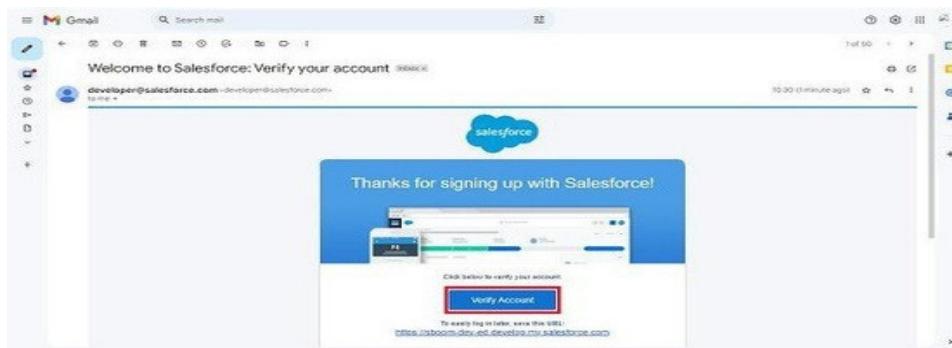
4. Company: College Name
5. County: India
6. Postal Code: pin code
7. Username: should be a combination of your name and company

This need not be an actual email id; you can give anything in the format: username@organization.com

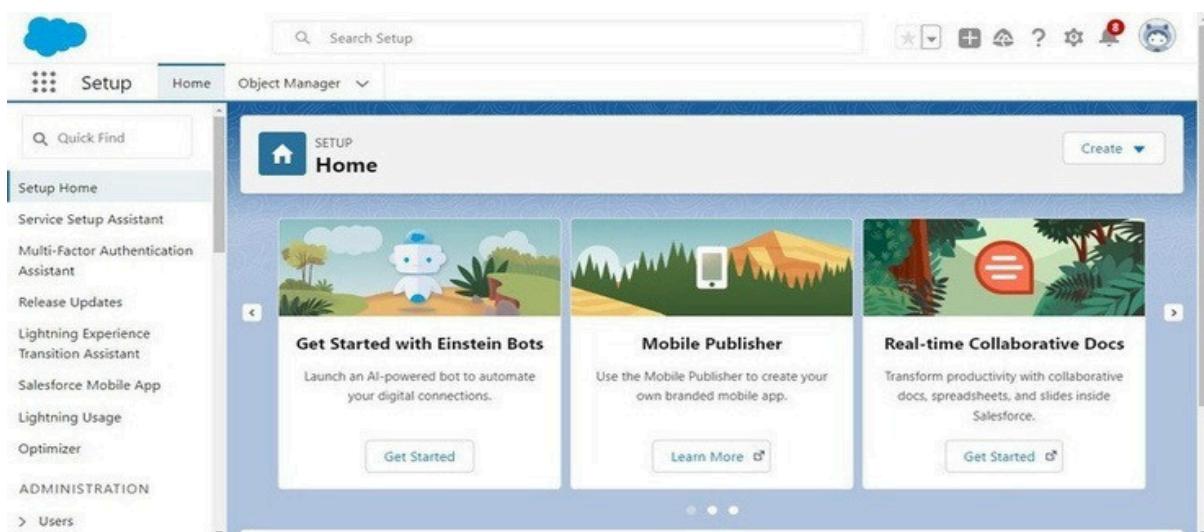
Click on sign me up after filling these.

Activity 2: Account Activation

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



2. Click on Verify Account
3. Give a password and answer a security question and click on change password.
4. Then you will redirect to your salesforce setup page.



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:

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

The screenshot shows the 'New Custom Object' setup page in Salesforce. The 'Custom Object Definition Edit' screen has the following visible fields and settings:

- Custom Object Information:** Fields include 'Label' (Product) and 'Plural Label' (Products).
- Record Name:** 'Object Name' is set to 'Product'.
- Description:** A large text input field.
- Context-Sensitive Help Setting:** Options to open the standard Salesforce.com Help & Training window or view as word-free version in Microsoft Word.
- Enter Record Name Label and Format:** Shows 'Record Name' as 'Product ID' and 'Data Type' as 'Text'. A note says: "The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API."
- Optional Features:** A section with checkboxes:
 - Allow Reports** (checked)
 - Allow Activities**
 - Track Field History**
 - Allow in Chatter Groups**
 - Enable Licensing**

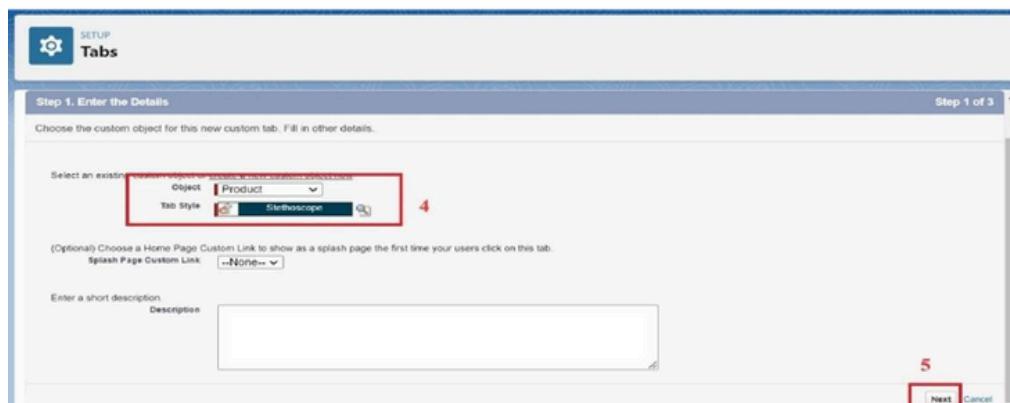
In the same way Create Purchase Order, Order Item, Inventory Transaction and Supplier objects.

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

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



Activity 2: Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are “Purchase Order, Order Item, Inventory Transaction, Supplier”.
2. 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

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

New Lightning App

App Details	App Branding
*App Name <input type="text" value="Medical Inventory Management"/> 3	Image <input type="file" value="Medical_Inventory_Management.jpg"/> 3
*Developer Name <input type="text" value="Medical_Inventory_Management"/>	Primary Color Hex Value <input type="text" value="#007002"/>
Description <input type="text" value="Enter a description..."/>	Org Theme Options <input type="checkbox"/> Use the app's image and color instead of the org's custom theme

App Launcher Preview

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	Selected Items
<input type="checkbox"/> Dash	<input checked="" type="checkbox"/> Products
<input checked="" type="checkbox"/> Dashboards	<input checked="" type="checkbox"/> Purchase Orders
	<input checked="" type="checkbox"/> Order Items
	<input checked="" type="checkbox"/> Inventory Transactions
	<input checked="" type="checkbox"/> Suppliers
	<input checked="" type="checkbox"/> Reports

6

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)
Purchase Order	Expiry Date	Date
	Purchase Order ID(Standard)	Text
	Supplier ID	Lookup (Supplier)
	Order Date	Date
	Expected Delivery Date	Date
	Actual Delivery Date	Date
Order Item	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:

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.

Activity 2: Creating a TextArea 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 TextArea field, click Next
7. Enter Field Label as "Product Description".
8. Click Next, Next, then Save & New.

Activity 3: Creating a Number Field in Product object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product custom object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Number” and click Next.
5. Enter Field Label as “ Current Stock Level”.
6. Length - 18, Decimal Places - 0.
7. 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 'Current Stock Level'. The 'Length' is set to 18 and the 'Decimal Places' is set to 0. Other settings like 'Required', 'Unique', and 'External ID' are shown at the bottom.

Activity 4: Creating a Currency Field in Product object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product custom object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Currency” and click Next.
5. Enter Field Label as “ Unit Price”.
6. Length - 16, Decimal Places - 2.
7. Select Required Field.
8. Click on Next, Next and Save.

Step 2. Enter the details Step 2 of 4

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 Number of digits to the right of the decimal point

Field Name 7 Description

Help Text

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

Auto-add to custom report type Add this field to existing custom report types that contain this entity 9

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.

1. 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.
2. Click on Fields & Relationship
3. Click on New.
4. Select “Lookup relationship” as data type and click Next.
5. Select the related object “Supplier”.
6. Click on Next.
7. Give Field Label as “Supplier ID”.
8. Select Required Field.
9. Click on Next ,Next , Next , Save.

Field Label 7

Field Name 8

Description

Help Text

Child Relationship Name 9

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 10

Activity 6: Creating a Date Field in Purchase Order object

To create fields in an object:

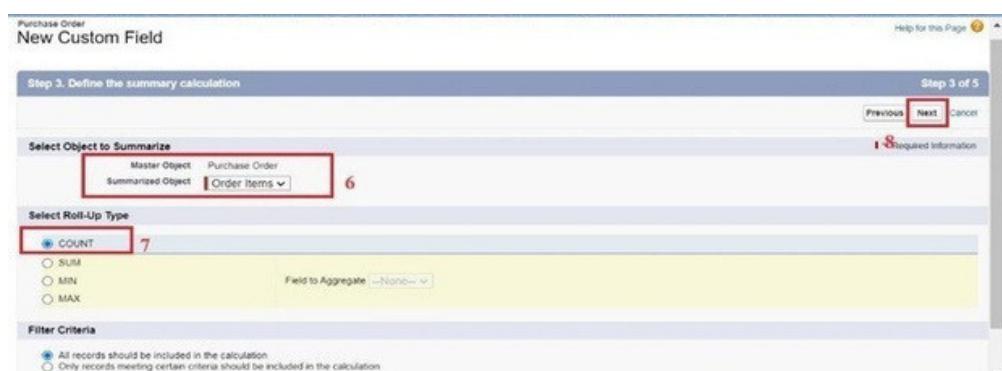
1. Go to setup >> click on Object Manager >> type object name (Purchase Order) in quick find box>> click on the Purchase Order object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Date” and click Next.
5. Enter Field Label as “Order Date”.
6. Click on Next, Next and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



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

To create fields in an object:

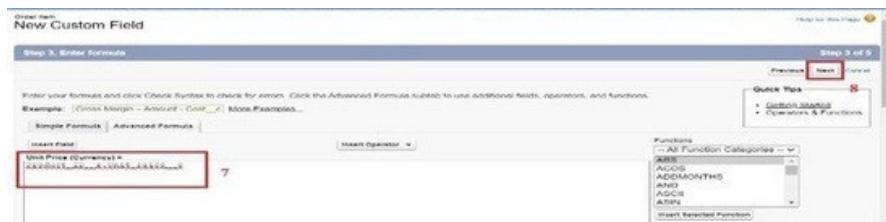
1. Go to setup >> click on Object Manager >> type object name (Purchase Order) in quick find box>> click on the Purchase Order object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Roll-Up Summary” and click Next.
5. Enter Field Label as “Order Count”.
6. Choose the Summarized Object as “Order Items”.
7. For Select Roll-Up Type select “Count”.
8. Click on Next, Next and Save.



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

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Order Item) in quick find box >> click on the Order Item object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Unit Price.
6. Select formula return type Currency, Click Next
7. Create and insert Advance formula: Product_ID__r.Unit_Price__c
8. Click Next, Next, then Save.



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

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Order Item) in quick find box >> click on the Order Item object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Amount.
6. Select formula return type Currency, Click Next
7. Create and insert Advance formula: Quantity_Received__c * Unit_Price__c
8. Click Next, Next, then Save.



Activity 10: Creating a Picklist Field in Inventory Transaction Object

To create fields in an object:

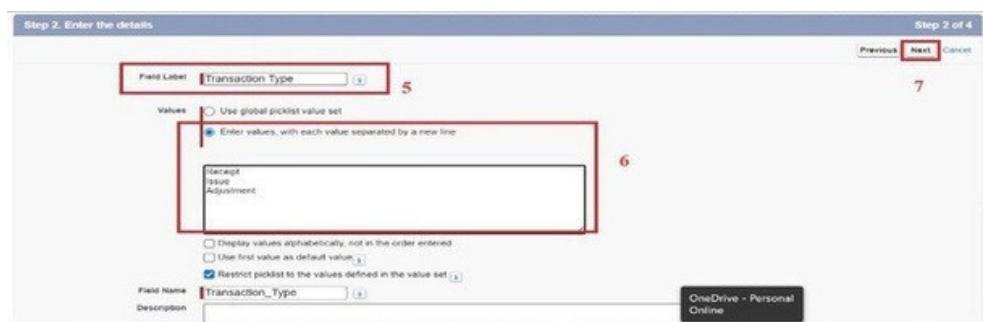
1. Go to setup >> click on Object Manager >> type object name (Inventory Transaction) in quick find box >> click on the Inventory Transaction Object.
2. Now click on “Fields & Relationships”.
3. Click on New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Transaction Type”.
6. In values select “Enter values, with each value separated by a new line” and enter values as shown below.

Receipt

Issue

Adjustment

7. Click on Next, Next and Save.



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

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Inventory Transaction) in quick find box >> click on the Order Item object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Total Order Cost.
6. Select formula return type Currency, Click Next
7. Create and insert Advance formula: Purchase_Order_ID__r.Total_Order_Cost__c
8. Click Next, Next, then Save.

Activity 12: 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.

The screenshot shows the 'Step 2, Enter the details' dialog box. It has fields for 'Field Label' (set to 'Phone Number'), 'Field Name' (set to 'Phone_Number'), 'Description', and 'Help Text'. Under 'Required', there are two checkboxes: 'Always require a value in this field in order to save a record' (checked) and 'Add this field to existing custom report types that contain this entity' (unchecked). A note at the bottom says 'Use Schema prefix: Choose left and quoted value API names in schema quotes ("Name") include numbers without quotes (e.g., when referencing an Account, "A-100"), and express date values as in the standard format ("MM/DD/YYYY"). Use value in field type a Custom Metadata field record type. Suffixes and prefixes are not supported for fields...'.

Activity 13: Creating a Email 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 “Email” and click Next.
5. Enter the Field Label as “Email”.
6. Click on Next, Next and Save.

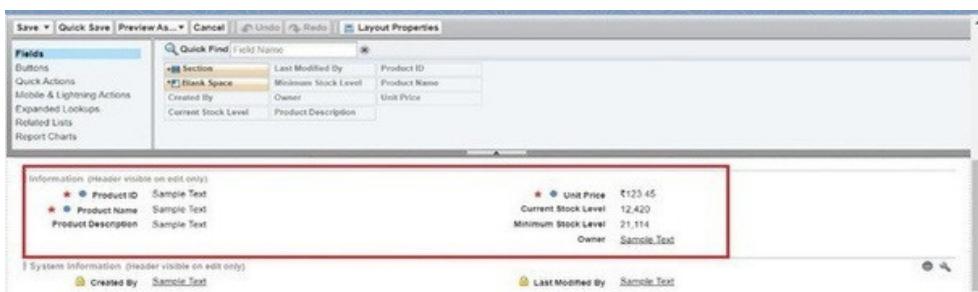
The screenshot shows the 'Step 2, Enter the details' dialog box. It has fields for 'Field Label' (set to 'Email'), 'Field Name' (set to 'Email'), 'Description', and 'Help Text'. Under 'Required', there are three checkboxes: 'Always require a value in this field in order to save a record' (unchecked), 'Do not allow duplicate values' (unchecked), and 'Set this field as the unique record identifier from an external system' (unchecked). A note at the bottom says 'Use Schema prefix: Choose left and quoted value API names in schema quotes ("Name") include numbers without quotes (e.g., when referencing an Account, "A-100"), and express date values as in the standard format ("MM/DD/YYYY"). Use value in field type a Custom Metadata field record type. Suffixes and prefixes are not supported for fields...'.

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

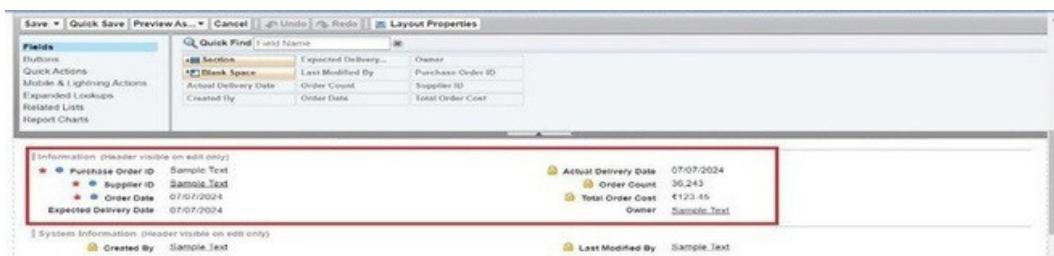
1. Go to setup >> click on Object Manager >> type object name (Product) in quick find box >> click on the Product object >> Page Layouts.
2. Click on the Product Layout.
3. Drag and arrange the field as shown below.



4. Click on Save.

Activity 2: To edit a Page Layout in 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 >> Page Layouts.
2. Click on the Purchase Order Layout
3. Drag and arrange the field as shown below



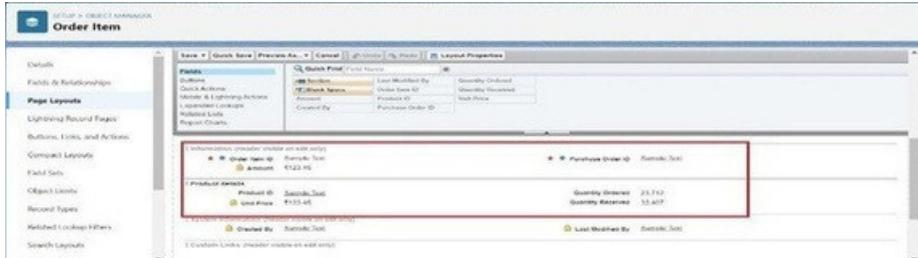
4. Click on field Order Date >> click on settings >> select Required and save it.
5. Click on field Total Order Cost >> click on settings >> select Read Only and save it.
6. Click Save.

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

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

2. Click on the Order Item Layout

3. Drag and arrange the field as shown below



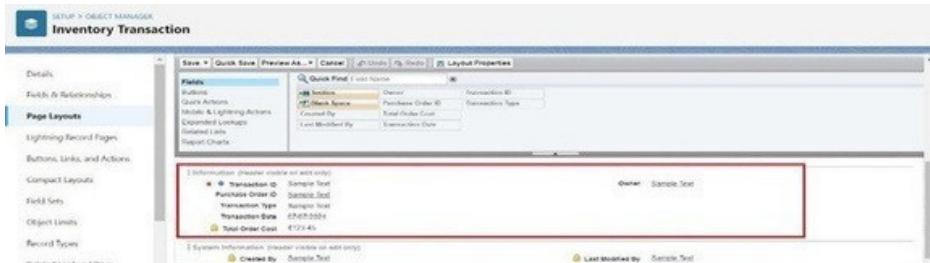
4. Click Save.

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

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

2. Click on the Inventory Transaction Layout

3. Drag and arrange the field as shown below



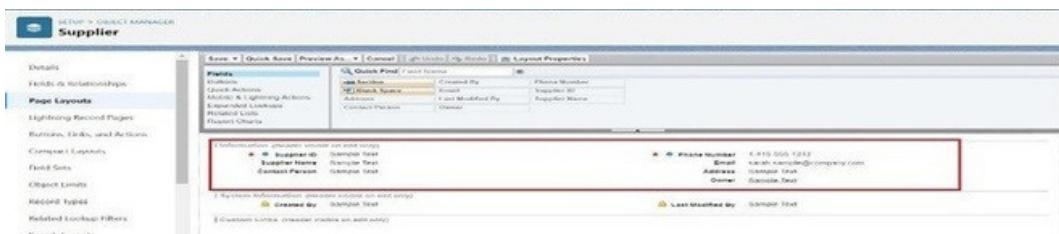
4. Click Save.

Activity 5: 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



4. 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

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.

The screenshot shows the 'Enter Compact Layout Information' page. Step 4 is highlighted in red around the 'Label' field containing 'Product Compact Layout'. Step 5 is highlighted in red around the 'Selected Fields' list which includes 'Product Name', 'Unit Price', and 'Current Stock Level'. Step 6 is highlighted in red around the 'Save' button at the bottom.

Product Compact Layouts

Compact Layout Assignment

The screenshot shows the 'Compact Layout Assignment' screen for the 'Product Compact Layouts' object. At the top, there are 'Save' and 'Cancel' buttons. Below them, a section titled 'Primary Compact Layout' contains the instruction: '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: Product Compact Layout' is shown, with the number '9' next to it. At the bottom of this section is another set of 'Save' and 'Cancel' buttons, with the number '10' next to the 'Save' button.

Activity 2: 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.

The screenshot shows the 'Compact Layout Edit' screen for creating a new compact layout. It has sections for 'Enter Compact Layout Information' (Label: 'Purchase Order Compact L...', Name: '(Purchase_Order_Compact...)') and 'Select Compact Layout Fields'. The 'Available Fields' list includes: Actual Delivery Date, Created By, Expected Delivery Date, Last Modified By, Owner, and Order Count. The 'Selected Fields' list includes: Purchase Order ID, Order Date, Total Order Cost, and Supplier ID. Between the two lists are 'Add' and 'Remove' buttons. At the bottom are 'Save' and 'Cancel' buttons, with the number '6' next to the 'Save' button.

Purchase Order Compact Layouts

Compact Layout Assignment

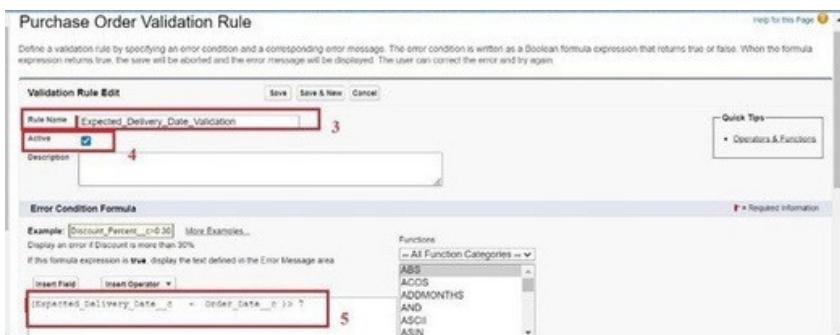
The screenshot shows the 'Compact Layout Assignment' screen for the 'Purchase Order Compact Layouts' object. It has a 'Primary Compact Layout' section with the instruction: 'Select the compact layout to use when this object's records appear as list items in the mobile app.' A dropdown menu labeled 'Primary Compact Layout: Purchase Order Compact Layout' is shown, with the number '9' next to it. At the bottom are 'Save' and 'Cancel' buttons, with the number '10' next to the 'Save' button.

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

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



Purchase Order Validation Rule

Help for this Page

Validation Rule Edit

Rule Name: Expected_Delivery_Date_Validation 3

Active 4

Description:

Error Condition Formula

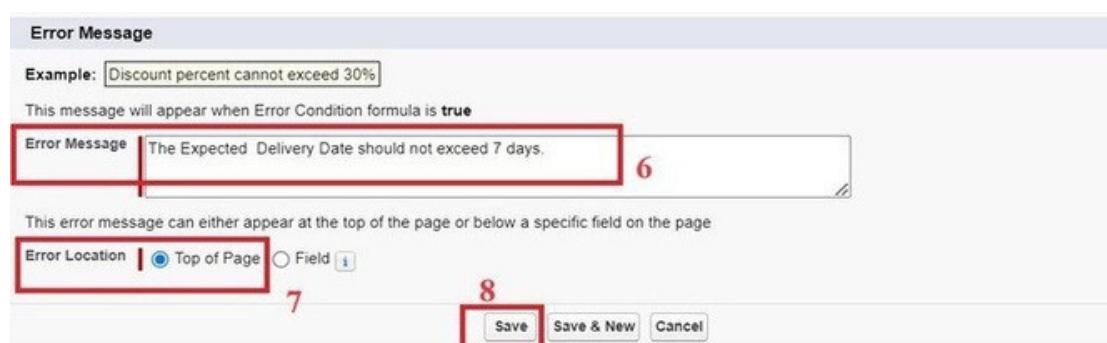
Example: Discount_Percent_c<0.30 More Examples...
Display an error if Discount is more than 30%
If this formula expression is true, display the text defined in the Error Message area

Insert Field Insert Operator Functions

Functions: ABS, ACOS, ADDMONTHS, AND, ASCII, ASN.

(Expected_Delivery_Date_c - Order_Date_c) > 7 5

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



Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is true

Error Message: The Expected Delivery Date should not exceed 7 days. 6

This error message can either appear at the top of the page or below a specific field on the page

Error Location: Top of Page Field 7

Save 8 Save & New Cancel

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

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



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

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

5. Change the password policies as mentioned :
6. User passwords expire in should be “never expires” .
7. 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:
- Require a minimum 1 day password lifetime:
- Don't immediately expire links in forgot password emails:

Buttons: Save, Save & New, Cancel

Activity 2: 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.

Profile Edit

Name: Purchase Manager
User License: Salesforce
Description:

Custom App Settings

Available	Default	Available	Default
All Tabs (standard, Admin Only)	<input checked="" type="radio"/>	Sales (standard, Lightning Sales)	<input checked="" type="radio"/>
Analytics Studio (standard, Insights)	<input checked="" type="radio"/>	Sales Console (standard, Lightning Sales Console)	<input checked="" type="radio"/>
App Launcher (standard, App switcher)	<input checked="" type="radio"/>	Salesforce Chat (standard, Chat)	<input checked="" type="radio"/>
Gift Solutions (standard, LightningGift)	<input checked="" type="radio"/>	Salesforce Scheduler Setup (standard, Lightning Scheduler)	<input type="radio"/>
Community (standard, Community)	<input checked="" type="radio"/>	Service Console (standard, Service Console)	<input type="radio"/>
Content (standard, Content)	<input checked="" type="radio"/>	Service Console (standard, Lightning Services)	<input checked="" type="radio"/>
Data Manager (standard, Data Manager)	<input type="radio"/>	Site.com (standard, Site)	<input checked="" type="radio"/>
Digital Experiences (standard, SalesforceDX)	<input checked="" type="radio"/>	Subscription Management (standard, Subscription Management)	<input type="radio"/>
Lightning Image API (standard, LightningImageAPI)	<input checked="" type="radio"/>	WMC (standard, WMC)	<input type="radio"/>
Marketing CRM Classic (standard, Marketing)	<input checked="" type="radio"/>		
Medical Inventory Management (Medical Inventory Management)	<input checked="" type="radio"/>		
Generic Management (standard, Generic Management)	<input type="radio"/>		

4. Scroll down to Custom Object Permissions and Give access permissions as mentioned in the below diagram.
5. Change the password policies as mentioned :
6. User passwords expire in should be “ never expires ”.
7. 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:
- Require a minimum 1 day password lifetime:
- Don't immediately expire links in forgot password emails:

Buttons: Save, Save & New, Cancel

Milestone 10 - Roles

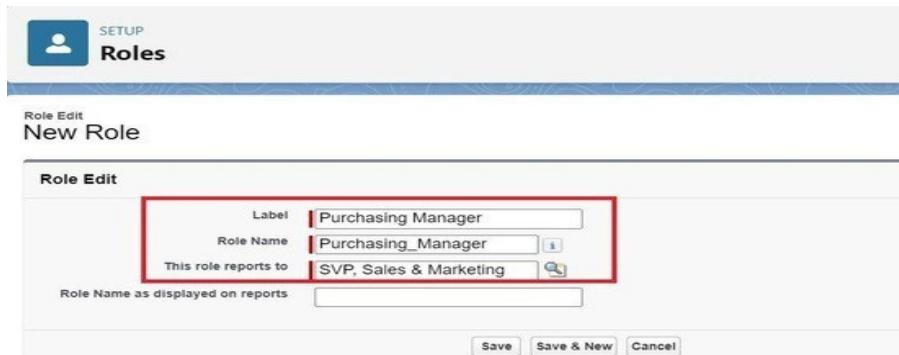
Roles in Salesforce are used to control record-level access and define the hierarchy of an organization, determining the level of visibility and sharing of records among users. Roles work in conjunction with profiles to provide a robust security model. While profiles control what actions users can perform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

Activity1 : Create Purchasing Manager Role.

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



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



Activity 2 : Create a Purchasing Manager Role.

1. Go to quick find >> Search for Roles >> click on Set Up Roles.
2. Click on Expand All and click on add role under SVP, Sales & Marketing role.
3. Give Label as “Inventory Manager” and the Role name gets auto populated. Then click on Save.

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.

1. 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 'Permission' selected under 'Users'. The main area displays a table of permission sets with columns for Action, Permission Set Label, Description, and License. The 'Label' column is sorted. The first few rows include 'Guest', 'Buyer Manager', 'Cloud Integration User', and 'Commerce Admin'. A 'New' button is visible at the top left of the table.

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

The screenshot shows the 'Create Permission Set' dialog. It has fields for 'Label' (set to 'Purchase Manager Create Access'), 'API Name' (set to 'Purchase_Manager'), and 'Description'. A 'Save' button is highlighted with a red box. The 'Session Activation Required' checkbox is unchecked.

3. 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 'Permission Sets' page in Salesforce. A specific permission set named 'Purchase Manager Create Access' is selected. Under the 'Order Items' tab, the 'Available' section has a checked checkbox, and the 'Visible' section also has a checked checkbox. In the 'Object Permissions' section, under 'Enabled', the 'Read' and 'Create' checkboxes are checked. There are other checkboxes for 'Edit', 'Delete', 'View All', and 'Modify All' which are not checked.

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

The screenshot shows the 'Select Users to Assign' screen. Under the 'Active Users' section, a user named 'John PurchaseM' is selected, indicated by a checked checkbox. The 'Next' button at the bottom right is highlighted with a red box.

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

The screenshot shows the 'Select an Expiration Option For Assigned Users' screen. 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.

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow >> Start From Scratch .
2. Select the record Triggered flow.Click on create.
3. Under Object select “Purchase Order”
4. Select A record is created or updated



5. Set EntryConditions:None
6. SelectFastFieldUpdates and click on Done
7. Underthererecordtrigger flow click on the “+” icon and select Get Records.
8. EnterLabelas“GetPurchase Record ”.
9. ForObjectselectPurchase Order.
10. ForConditionRequirements , select All Conditions are Met(AND)

For the first condition select as follows:

Field:Id
Operator:Equals
Value:{!\$Record.Id}



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



13. In the Flow Builder, click on the Manager tab on the left-hand side >> Click on New Resource >> In the Resource Type dropdown, select Variable.
14. Enter API name as ActualDeliveryDate >> Select Data type as Date >> Click on Done.
15. From the Toolbox drag and drop Assignment element.
16. Enter the label as "Assignment".
17. Set Variable Values:

a) Variable : {!ActualDeliveryDate}

Operator : Equals

Value : {!\$Record.Order_Date__c}

b) Variable : {!ActualDeliveryDate}

Operator : Add

Value : 3

* Label **Assignment**

* API Name **Assignment_1**

Description

Set Variable Values
Each variable is modified by the operator and value combination.

Variable <input type="text" value="ActualDeliveryDate"/>	Operator <input type="button" value="Equals"/>	Value <input type="text" value="\$Record > Order Date"/>
Variable <input type="text" value="ActualDeliveryDate"/>	Operator <input type="button" value="Add"/>	Value <input type="text" value="3"/>

+ Add Assignment

18. Click Done

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

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

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

22. Set Filter Conditions : None -Always Update Record

23. Set Field Values for the Trip Record as

Field : Actual_Delivery_Date__c

Value : {!ActualDeliveryDate}

How to Find Records to Update and Set Their Values

None—Always Update Record

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

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

Set Filter Conditions

Condition Requirements to Update Record

Set Field Values for the Purchase Order Record

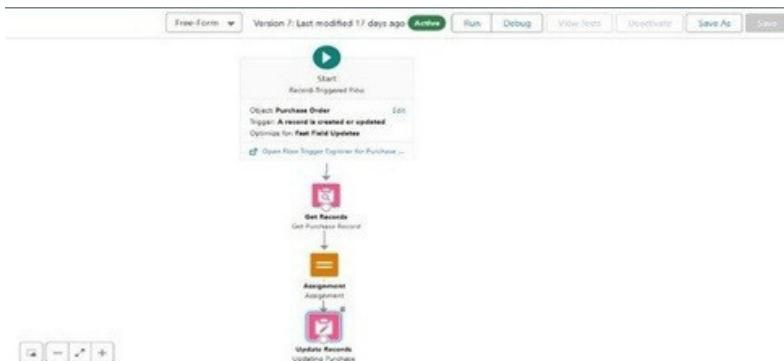
Field <input type="text" value="Actual_Delivery_Date__c"/>	Value <input type="text" value="ActualDeliveryDate"/>
---	--

+ Add Field

24. Click Done

25. Save the flow as “Actual Delivery Date Updating”.

26. Activate the flow.



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:

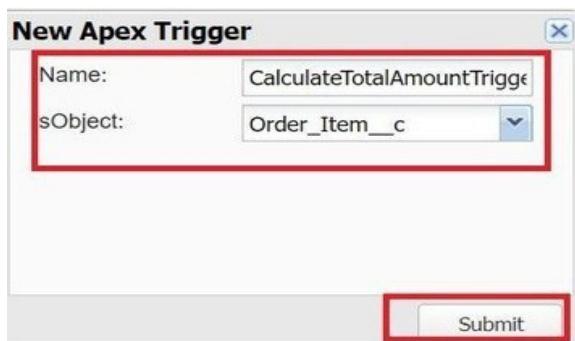
Log in to your Salesforce account with administrative privileges.

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.update, Trigger.delete, 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.

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.

The screenshot shows the 'Create Report' interface in Salesforce. On the left, there's a sidebar with categories like 'Recently Used', 'All', 'Accounts & Contacts', 'Opportunities', etc. The main area has a search bar 'Select a Report Type' with 'Q. Purchase' typed in. Below it, a list of 'Report Type Name' and 'Category' is shown, with 'Purchase Orders' selected. To the right, a 'Details' panel is open for the 'Purchase Orders' report, showing its icon, name, and a large red box around the 'Start Report' button. Other details like 'Fields (17)', 'Created By You', and 'Created By Others' are also visible.

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”)

The screenshot shows the 'Purchase Orders based on Suppliers' report interface. On the left, there's a sidebar with 'Fields' and 'Columns' sections, both containing 'GROUP BY' options like 'Supplier ID'. The main area displays a table with columns: 'Supplier ID', 'Purchase Order/Purchase Order ID', 'Order Count', and 'Total Order Cost'. The data shows records for Supplier 001 (4) and Supplier 002 (1), with a total of 14 purchase orders and a total cost of \$16,325.00.

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)	4	\$2,075.00 \$2,250.00 \$2,000.00 \$3,500.00
Supplier 002 (1)	Purchase-0005 (1)	1	\$4,500.00
Total (14)		14	\$16,325.00

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.

This screenshot shows the same 'Purchase Orders based on Suppliers' report as the previous one, but with a different view. It includes summary rows at the top for 'Total Records', 'Total Order Count', and 'Total Total Order Cost'. The data table below shows the same 14 purchase orders from Supplier 001 and Supplier 002, totaling \$16,325.00.

Total Records	Total Order Count	Total Total Order Cost	
14	14	\$16,325.00	
Supplier ID - 1	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)	4	\$2,075.00 \$2,250.00 \$2,000.00 \$3,500.00
Supplier 002 (1)	Purchase-0005 (1)	1	\$4,500.00
Total (14)		14	\$16,325.00

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

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 > **Purchase Orders with Order Items and Product ID**

Groups: Supplier ID, Purchase Order ID

Columns: Product ID: Product ID, Product ID: Product Name, Order Count, Quantity Received, Amount

Report Preview: Update Preview Automatically

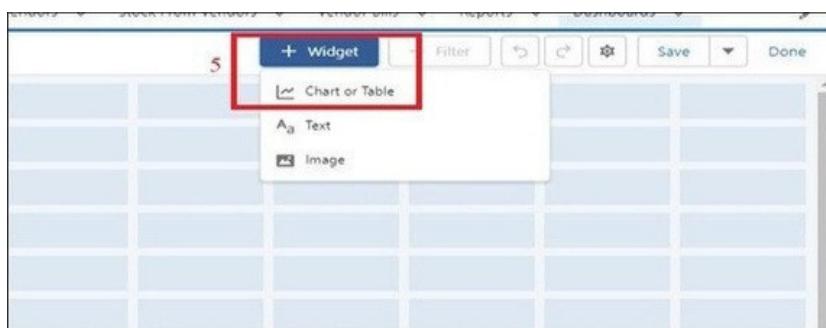
Supplier ID	Actual Delivery Date	Purchase Order: Purchase Order ID	Product ID: Product ID	Product ID: Product Name	Order Count	Quantity Received	Amount
Supplier A (1)	2023-01-01	Purchase-0001 (1)	Syr-0001	Syringe	2	50	\$100.00
			Cap-0001	Capsule	2	100	\$200.00
					2	200	\$300.00
Supplier B (2)	2023-01-02	Purchase-0002 (2)	Syr-0001	Syringe	3	50	\$150.00
			Cap-0001	Capsule	3	100	\$300.00
			Cap-0001	Capsule	1	50	\$100.00
Supplier C (3)	2023-01-03	Purchase-0003 (3)	Syr-0001	Syringe	3	50	\$150.00
			Cap-0001	Capsule	3	100	\$300.00
			Cap-0001	Capsule	1	50	\$100.00
Supplier D (4)	2023-01-04	Purchase-0004 (4)	Syr-0001	Syringe	4	50	\$200.00
			Cap-0001	Capsule	4	100	\$400.00
			Cap-0001	Capsule	2	50	\$100.00
			Cap-0001	Capsule	1	50	\$100.00

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.

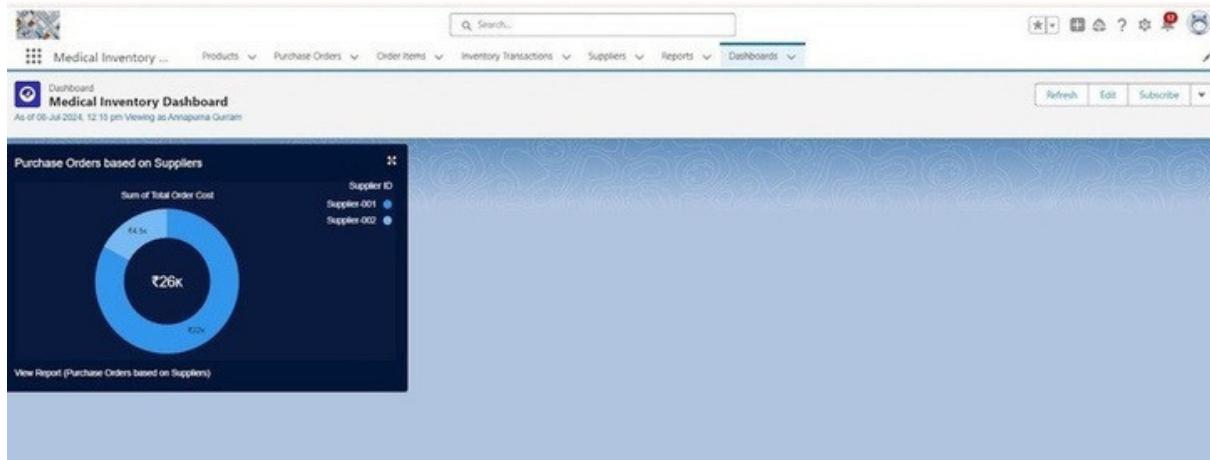
Activity 1: - Create Dashboard

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.



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 is a Salesforce-based solution designed to efficiently manage suppliers, products, purchase orders, and stock details. Through features like custom objects, validation rules, automation (flows & triggers), and dashboards, the project improves accuracy, efficiency, and reporting in medical inventory operations.

It provides practical experience in Salesforce administration and development, including user management, automation, and data analysis. Overall, the project results in a scalable and reliable system for effective medical inventory tracking and management.