

**VEERAMMAL ENGINEERING COLLEGE**

**PVP NAGAR, K.SINGARAKOTTAI (Post),**

**DINDIGUL-624708.**



**Salesforce Developer-SB8067**

**RECORD NOTE BOOK**

**NAME** : .....

**REGISTER NO** : .....

**BRANCH** : .....

**DEGREE/YEAR/SEM** : .....



## VEERAMMAL ENGINEERING COLLEGE

PVP NAGAR, K.SINGARAKOTTAI (Post),

DINDIGUL-624708.

### BONAFIDE CERTIFICATE

**NAME** : .....

**REGISTER NO** : .....

**BRANCH/YEAR/SEM** : .....

**SUBJECT** : .....

Certified that this is the bonafide record of work done by the above student  
in ..... Laboratory during the year 20 - 20

**Lab-In Charge**

**Head of the Department**

Submitted for the Practical Examination held on .....

**Internal Examiner**

**External Examiner**



## **Naan Mudhalvan**

### **SalesForce Developer Course**

**College Code:** 9212

**College Name:** Veerammal Engineering College

**Project Title : -Medical inventory Management**

**Number of students in group:** 2

**Team ID :** NM2025TMID01779

**Group Members:**

**B.BAGAVATHI ARULESWARI (CSE IV )**

**RESHMA.JT (CSE IV )**

**SUBMITTED BY:**

**BAGAVATHI ARULESWARI (CSE IV )& RESHMA J T**



# Medical inventory management

# **MEDICAL INVENTORY MANAGEMENT SYSTEM USING SALESFORCE**

**College Name:** VEERAMMAL ENGINEERING  
COLLEGE ,DINDUGAL

**College Code:**9212

**Team Size:**2

**TEAM ID :** NM2025TMID01779

**Team Leader Name:** RESHMA

**Team Member:**ARULESHWERI BHAGAVATHY

**PROJECT NAME:** MEDICAL INVENTORY MANAGEMENT

# **Title: Medical Inventory Management System Using Salesforce....**

## **Project Overview:**

- Healthcare institutions often face challenges in managing their medical supplies, such as shortages, overstocking, and wastage due to expired medicines. To address these issues, this project leverages Salesforce CRM to develop a Medical Inventory Management System. The application streamlines the tracking, procurement, and distribution of medicines, equipment, and consumables across hospitals, clinics, and pharmacies.
- By using custom objects, workflows, automation, and reports, the system ensures real-time monitoring of stock levels, expiry alerts, and efficient supplier management. It also enables role-based access for administrators, pharmacists, and medical staff to maintain security and accountability. The project highlights how

technology can improve operational efficiency, reduce wastage, and enhance patient care in the healthcare sector.

## Objectives:

- **Maintain Accurate Inventory Records:** Track stock levels of medicines, surgical items, and consumables in real time.
- **Prevent Shortages and Wastage:** Set automated alerts for low stock and near-expiry items to minimize losses.
- **Automate Procurement Processes:** Generate purchase orders and streamline supplier management.
- **Improve Resource Utilization:** Optimize allocation of medical supplies across departments and branches.
- **Enhance Operational Efficiency:** Reduce manual errors through workflows, triggers, and automated reports.
- **Enable Role-Based Access:** Provide secure access for pharmacists, nurses, and administrators.

- **Support Data-Driven Decisions:** Generate analytics and reports for forecasting and financial planning.
- **Promote Patient Safety:** Ensure timely availability of essential medicines and equipment.

## **Student Outcomes:**

- **Hands-on Experience with Healthcare Automation:** Students gain practical skills in configuring Salesforce objects, automating workflows, and managing real-time medical inventory tracking.
- **Understanding of Project Lifecycle in Healthcare CRM:** Students learn the complete process from requirement gathering to deployment, enhancing their ability to implement Salesforce projects in the healthcare sector.

- **Enhanced Analytical and Problem-Solving Skills:** Students develop the ability to identify stock management challenges, design automated solutions, and troubleshoot issues effectively.
- **Improved Collaboration Skills:** Students gain experience working in teams, coordinating roles such as requirement analysis, development, testing, and deployment.
- **Industry-Relevant Exposure:** Students get exposure to real-world use cases of Salesforce CRM in healthcare management, preparing them for careers in health IT and enterprise CRM solutions.

## **System Requirements:**

### **Hardware Requirements:**

- Computer with minimum 4 GB RAM, Dual-core processor (8 GB recommended for smooth performance)
- Stable internet connection

## **Software Requirements:**

- Salesforce Developer Edition Org
- Modern Web Browser (e.g., Google Chrome, Firefox, Microsoft Edge)

## **Phases Overview:**

Phase no	Phase Name	Phase Description	Page no
1.	Requirement Analysis & Planning	Gathering requirements from hospital staff, pharmacists, and administrators; defining scope, goals,	

		and planning data model & workflows.	
2.	Salesforce Development – Backend & Configurations	Creating custom objects (Medicines, Equipment, Suppliers, Requests), fields, relationships; setting up Flows and Apex Triggers for inventory automation.	
3.	UI/UX Development & Customization	Building Lightning App for medical inventory, customizing page layouts, record pages, adding fields, and developing UI logic for users.	
4.	Data Migration, Testing & Security	Creating Users, Profiles, Permission Sets; configuring Sharing Rules, Report Types, Reports, and Dashboards; testing functionalities and ensuring HIPAA-compliant data security.	

5.	Deployment, Documentation & Maintenance	Designing and finalizing Home Page, deploying the solution to production, preparing documentation, training users, and ensuring ongoing system maintenance.	
----	---	---	--

## Phase 1: Requirement Analysis & Planning

### Project Goal:

To manage and track medical inventory (medicines, equipment, and supplies) using Salesforce, ensuring availability, transparency, and efficient distribution.

### Key Objectives:

- Monitor stock levels of medicines and supplies
- Automate reorder alerts when stock is low

- Track donors, suppliers, and receivers (patients/NGOs/hospitals)
- Ensure expiry-date tracking for medicines
- Generate reports for usage, wastage, and distribution efficiency

## **Phase 2: Salesforce Development – Backend & Configurations**

### **Milestone 1: Salesforce Developer Account Creation**

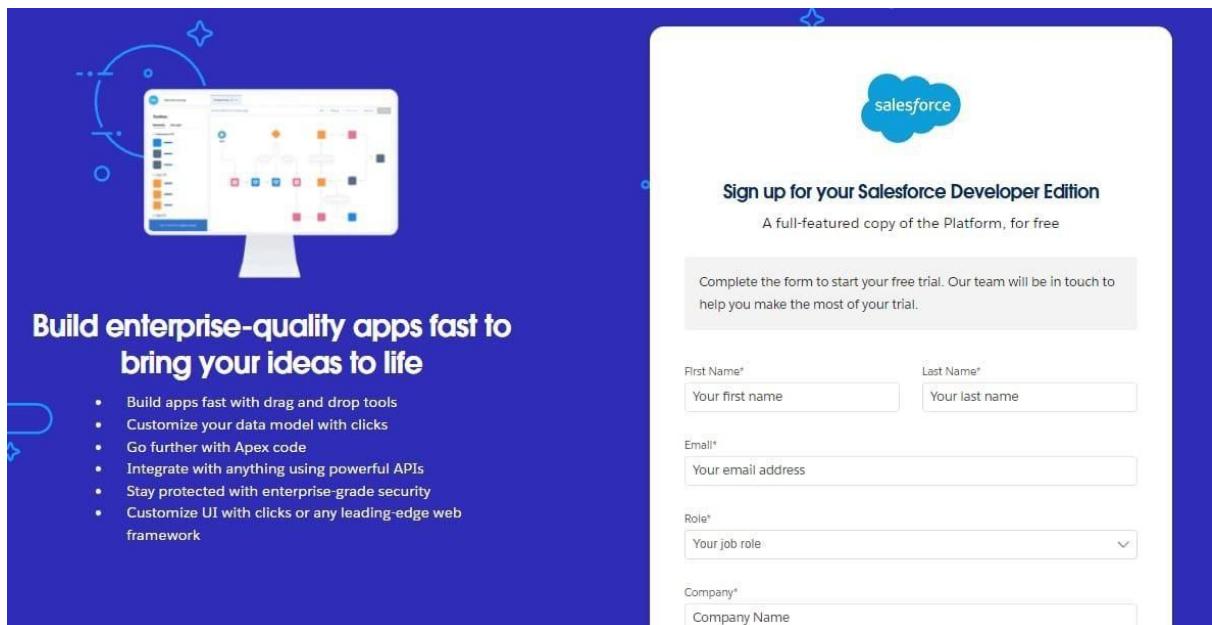
#### **Activity 1: Creating Developer Account**

To build and configure the system, we need a Salesforce Developer Org.

Creating a developer org in salesforce.

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

2. On the sign up form, enter the following details :



First name & Last name

Email

Role : Developer

Company : College Name

Country : India

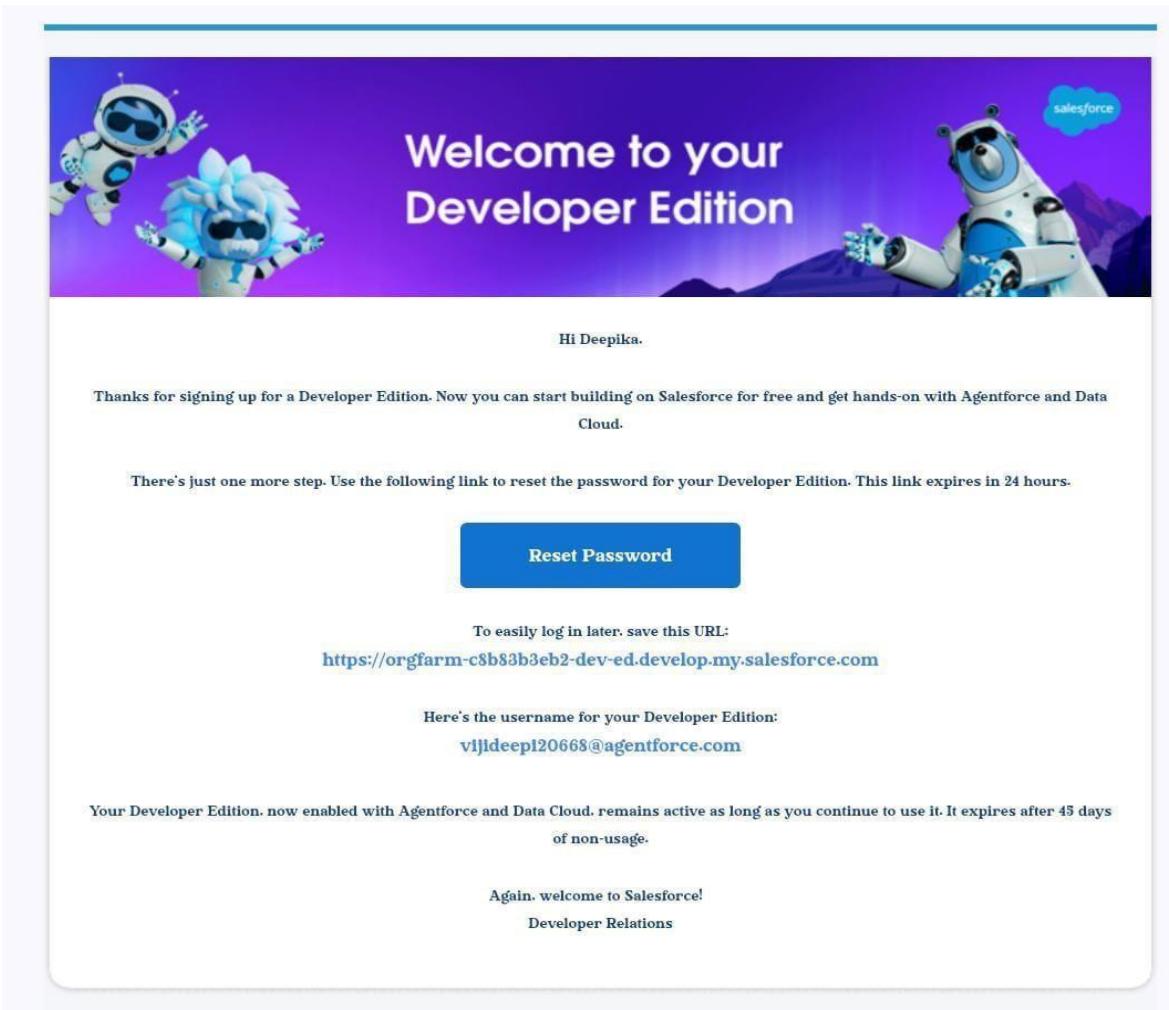
Postal Code : pin code

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**



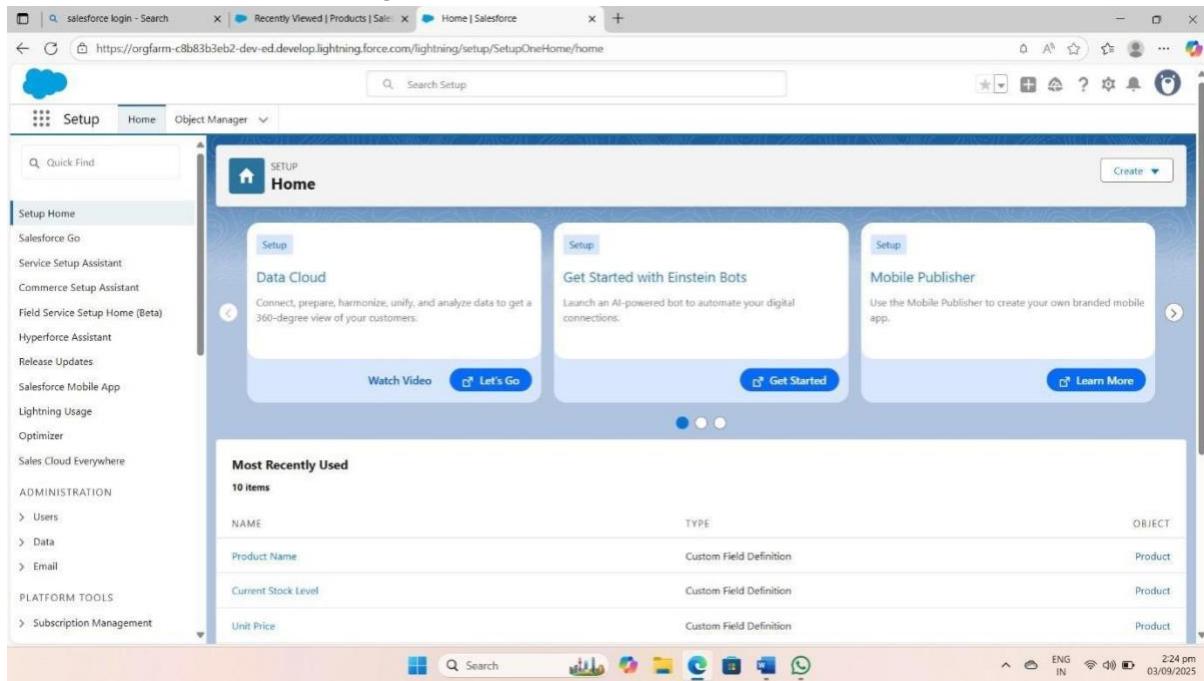
1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.

## 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.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for Setup, Home, and Object Manager. A search bar labeled "Search Setup" is present. The main content area is titled "Object Manager" and displays a table of 53+ items sorted by Label. The columns are labeled: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. Standard objects listed include Account, Activity, Address, Agent Work, Alternative Payment Method, API Anomaly Event Store, Appointment Category, Appointment Invitation, Appointment Invitee, Appointment Topic Time Slot, Approval Submission, Approval Submission Detail, and Approval Work Item.

The screenshot shows the "New Custom Object" page. The title is "Custom Object Definition Edit". The "Custom Object Information" section contains fields for "Label" (Account) and "Plural Label" (Accounts). There is also a checkbox for "Starts with vowel sound". The "Object Name" field is set to "Account" with an example of "Account". The "Description" field is empty. Below these fields are "Context Sensitive Help Setting" options and a "Content Name" dropdown set to "None". At the bottom, there are "Save", "Save & New", and "Cancel" buttons.

## 4. Enter the label name as Product

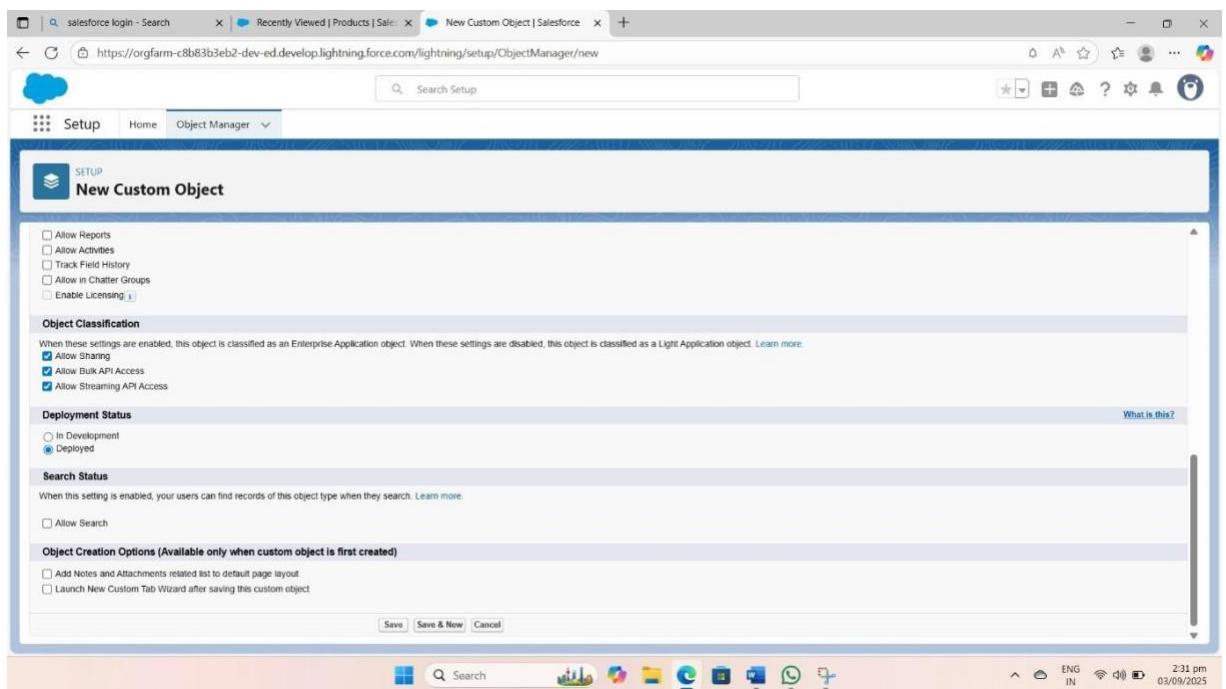
## 5. Enter Plural label name as Products

6. Enter Record Name as Product ID

7. Select Data Type as Text.

8. Select Allow reports.

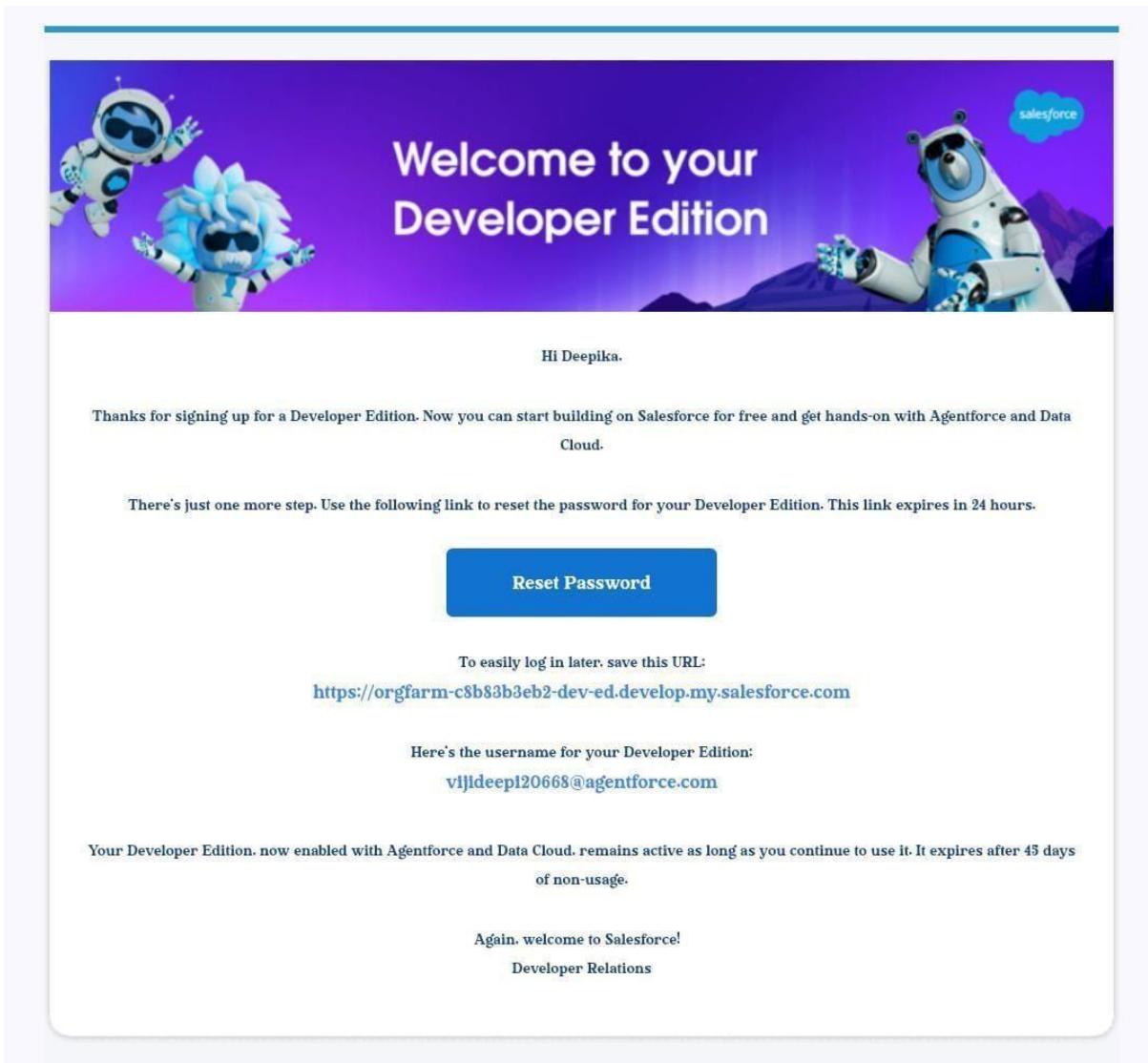
9. Select Allow search.



10. Click on Save and New

## 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.

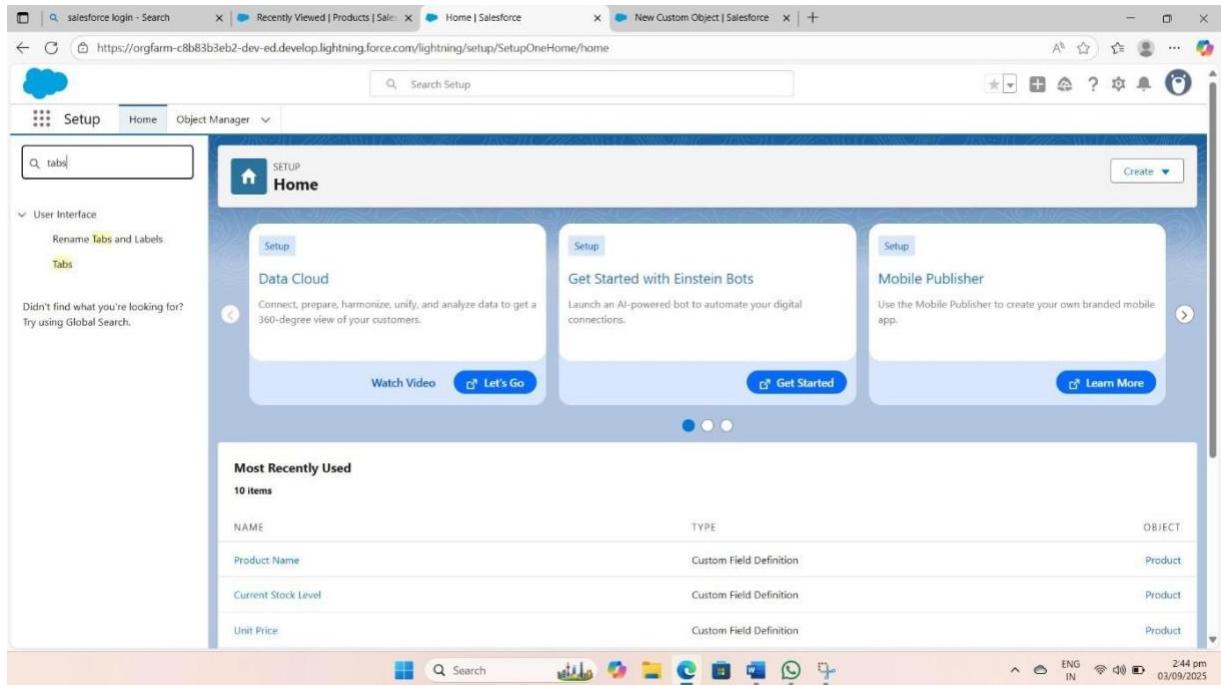
## **Milestone 3- Tabs**

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



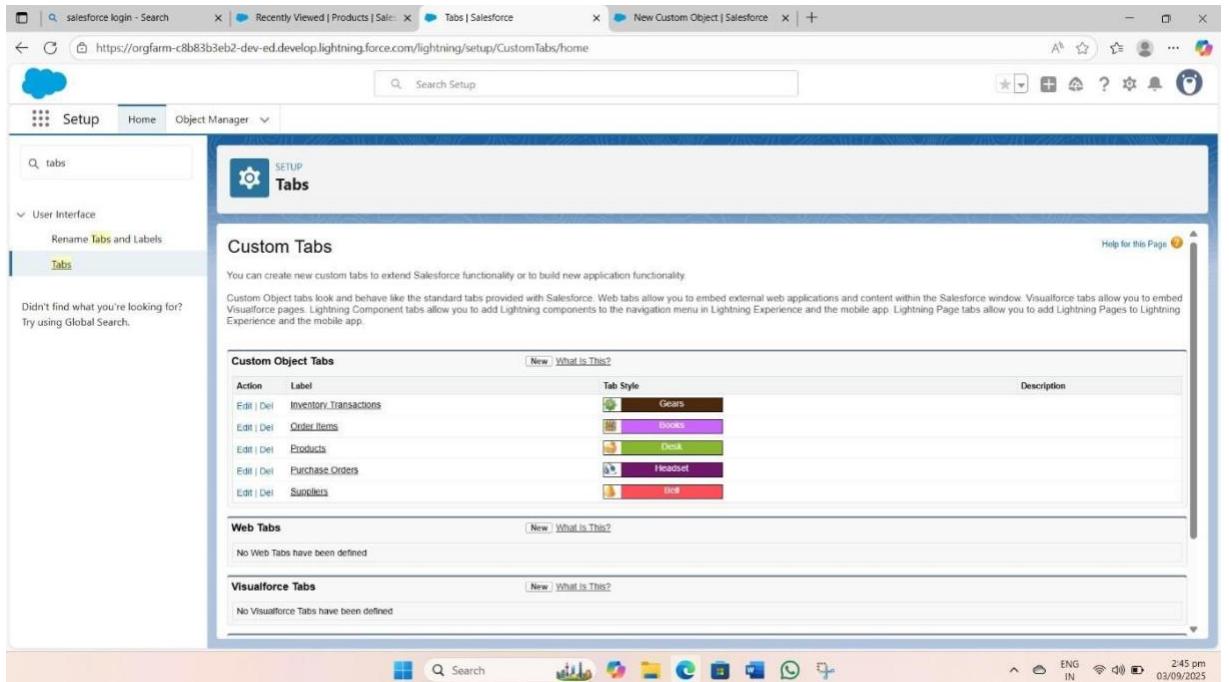
3. Click on New (under custom object tab).

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 .

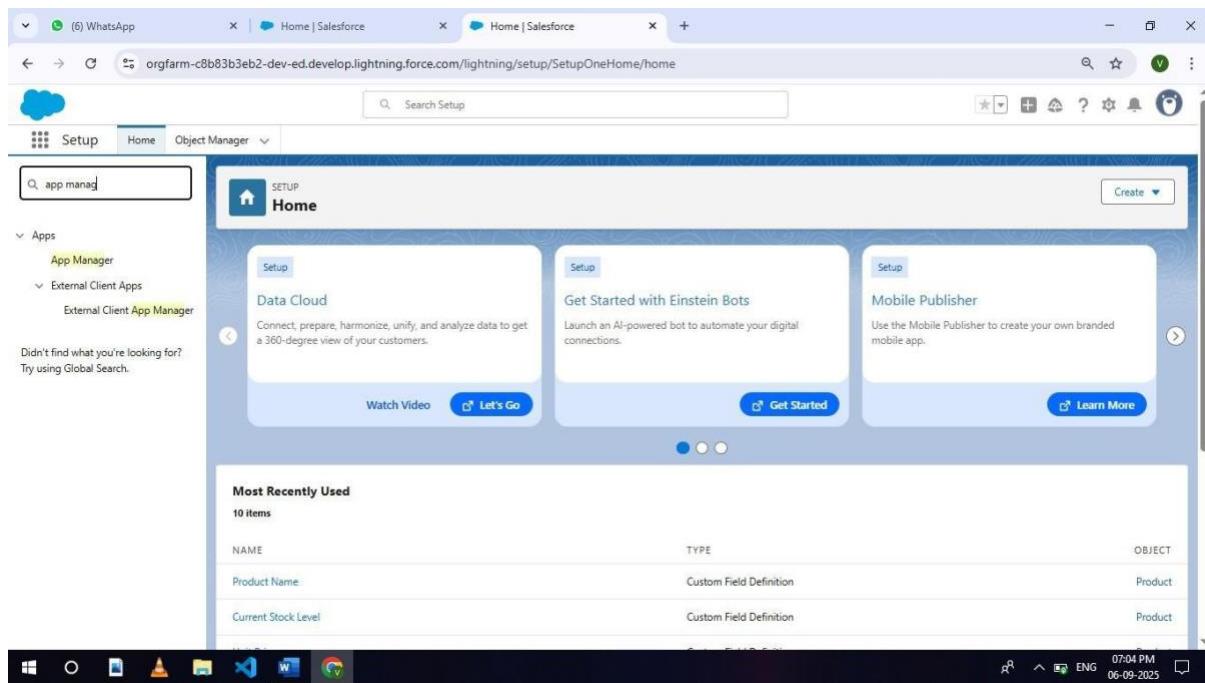
## Milestone 4- The Lightning App

- 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

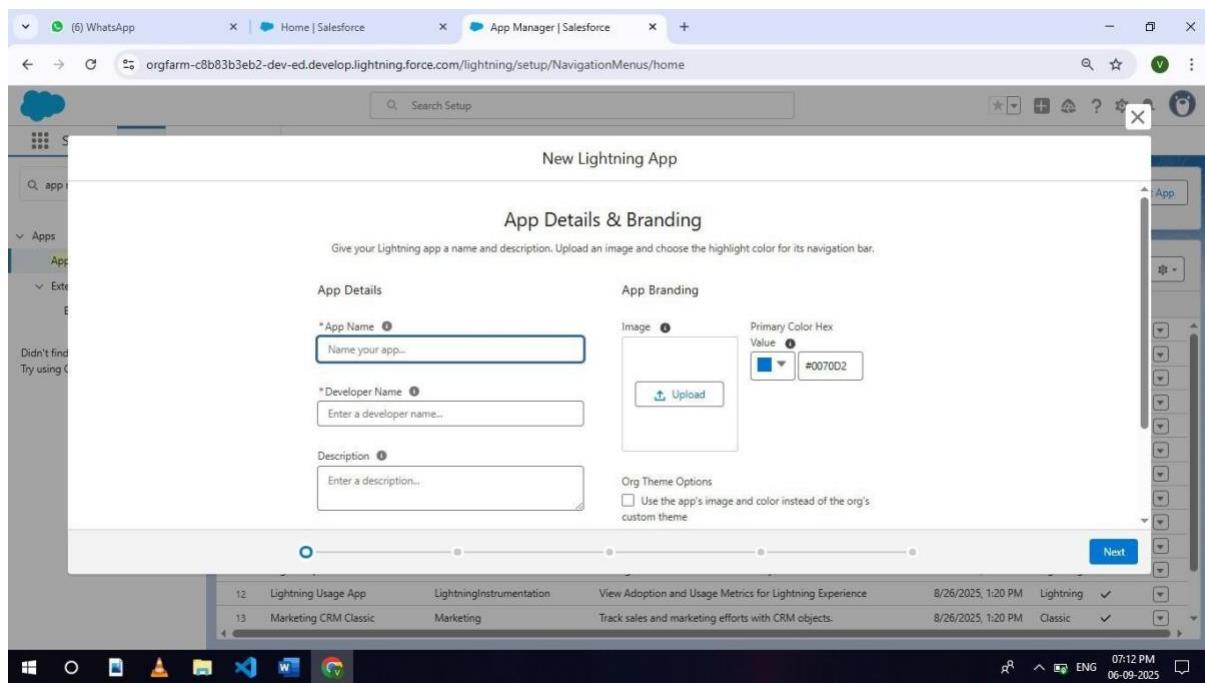
1. From Setup, enter App Manager in the Quick Find and select App Manager.



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.

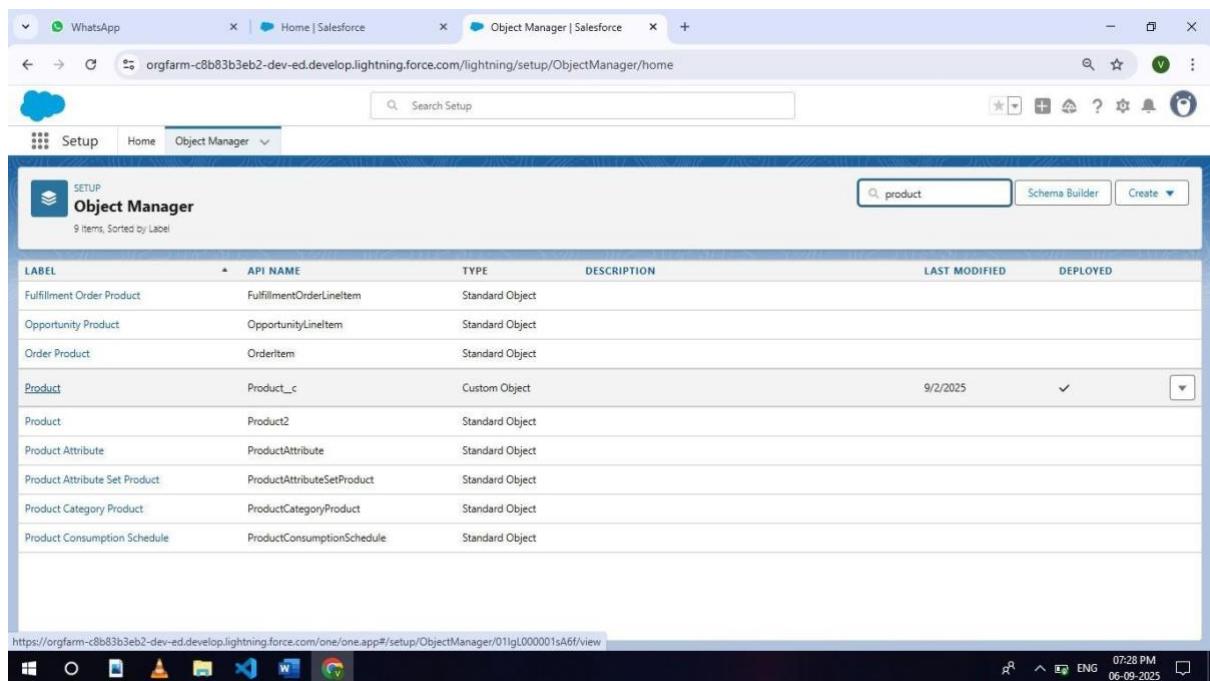
7. From Available Profiles, select System Administrator and move it to Selected Profiles.

## 8. Click Save & Finish.

# Milestone 5- Fields Object

# 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.

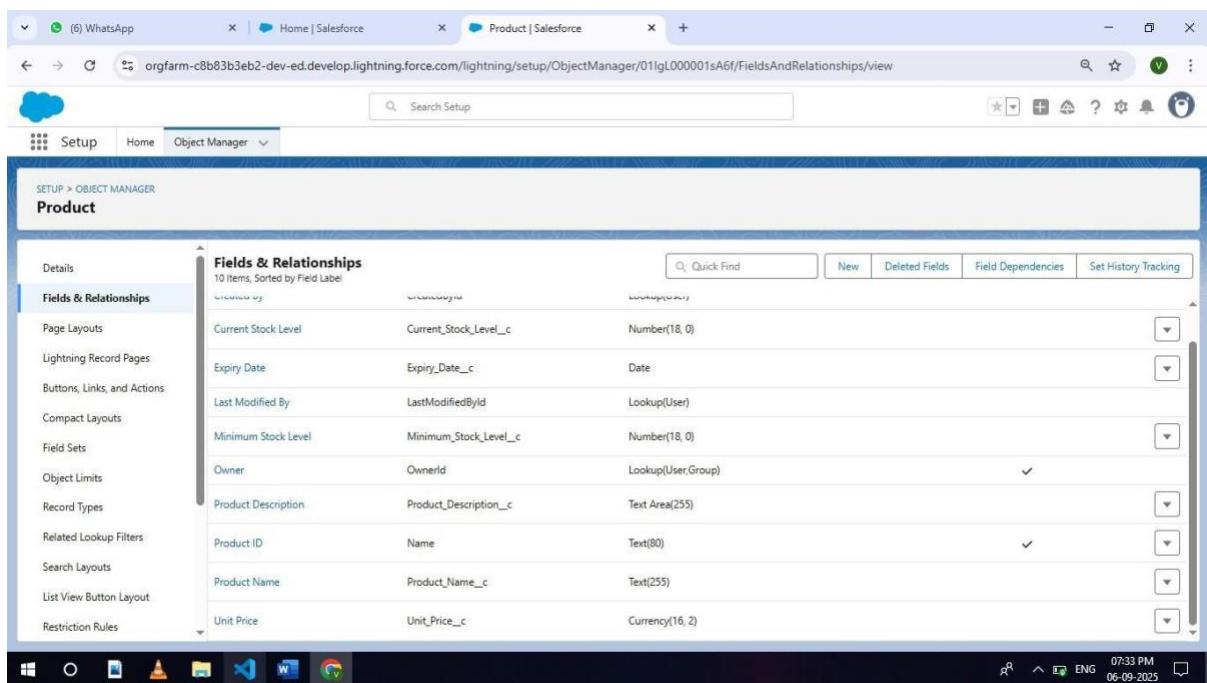
2. Click the Object Manager tab next to Home.

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

The screenshot shows the Salesforce Object Manager page. At the top, there are three tabs: WhatsApp, Home | Salesforce, and Object Manager | Salesforce. The Object Manager tab is active. Below the tabs, there's a search bar with the placeholder "Search Setup" and a dropdown menu with options like "Schema Builder" and "Create". The main area is titled "Object Manager" and shows a table with 9 items, sorted by Label. The columns are labeled: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The table includes rows for Fulfillment Order Product, Opportunity Product, Order Product, Product (selected), Product2, Product Attribute, Product Attribute Set Product, Product Category Product, and Product Consumption Schedule. The Product row is highlighted with a blue border. The status bar at the bottom shows the URL "https://orgfarm-c8b83b3eb2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgL000001sA6f/view", the date "06-09-2025", and the time "07:28 PM".

## 5. Click on New

## 6. Select TextArea field, click Next

## 7. Enter Field Label as “Product Description”

The screenshot shows the Salesforce Product Fields & Relationships page. The left sidebar has links for Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main area is titled "Fields & Relationships" and shows a table with 10 items, sorted by Field Label. The columns are labeled: FIELD LABEL, API NAME, and DATA TYPE. The table includes rows for Current Stock Level, Expiry Date, Last Modified By, Minimum Stock Level, Owner, Product Description (selected), Product ID, Product Name, and Unit Price. The Product Description row is highlighted with a blue border. The status bar at the bottom shows the URL "https://orgfarm-c8b83b3eb2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgL000001sA6f/FieldsAndRelationships/view", the date "06-09-2025", and the time "07:33 PM".

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”

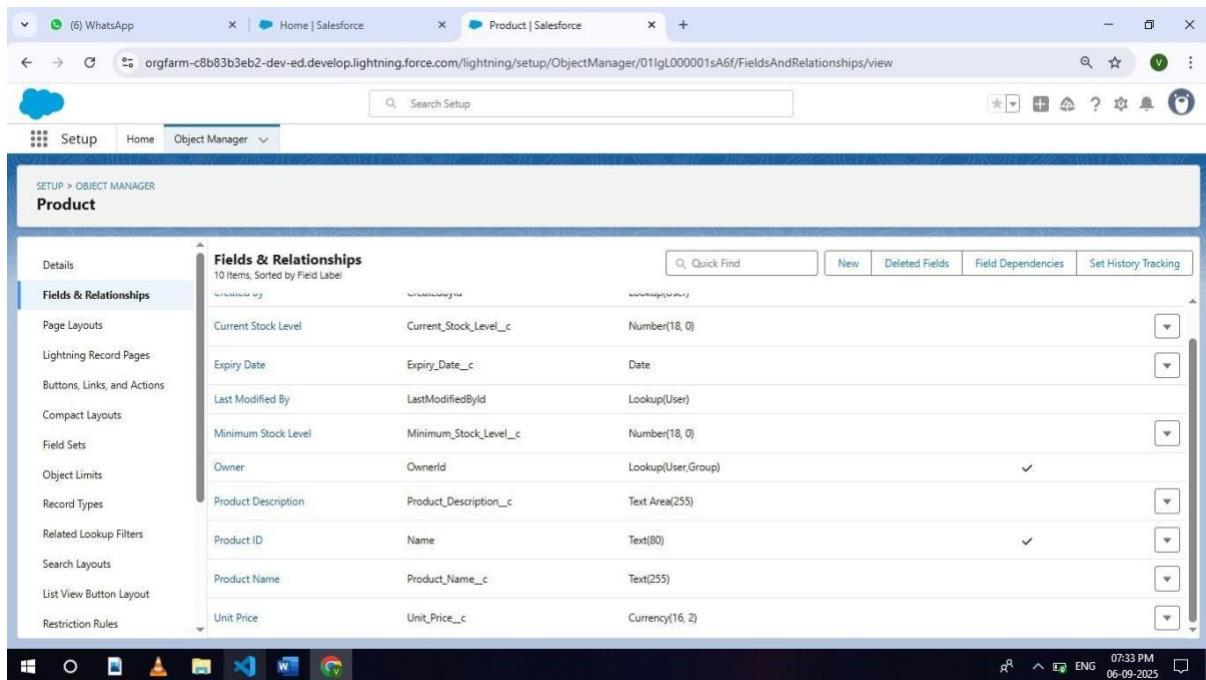
3. Click on New.

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.



## 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.

The screenshot shows the Salesforce Object Manager interface for the 'Product' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Fields & Relationships' and displays 10 items, sorted by Field Label. The table shows the following fields:

Field Label	Name	Type
Current Stock Level	Current_Stock_Level_c	Number(18, 0)
Expiry Date	Expiry_Date_c	Date
Last Modified By	LastModifiedById	Lookup(User)
Minimum Stock Level	Minimum_Stock_Level_c	Number(18, 0)
Owner	OwnerId	Lookup(User,Group)
Product Description	Product_Description_c	Text Area(255)
Product ID	Name	Text(80)
Product Name	Product_Name_c	Text(255)
Unit Price	Unit_Price__c	Currency(16, 2)

## 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.

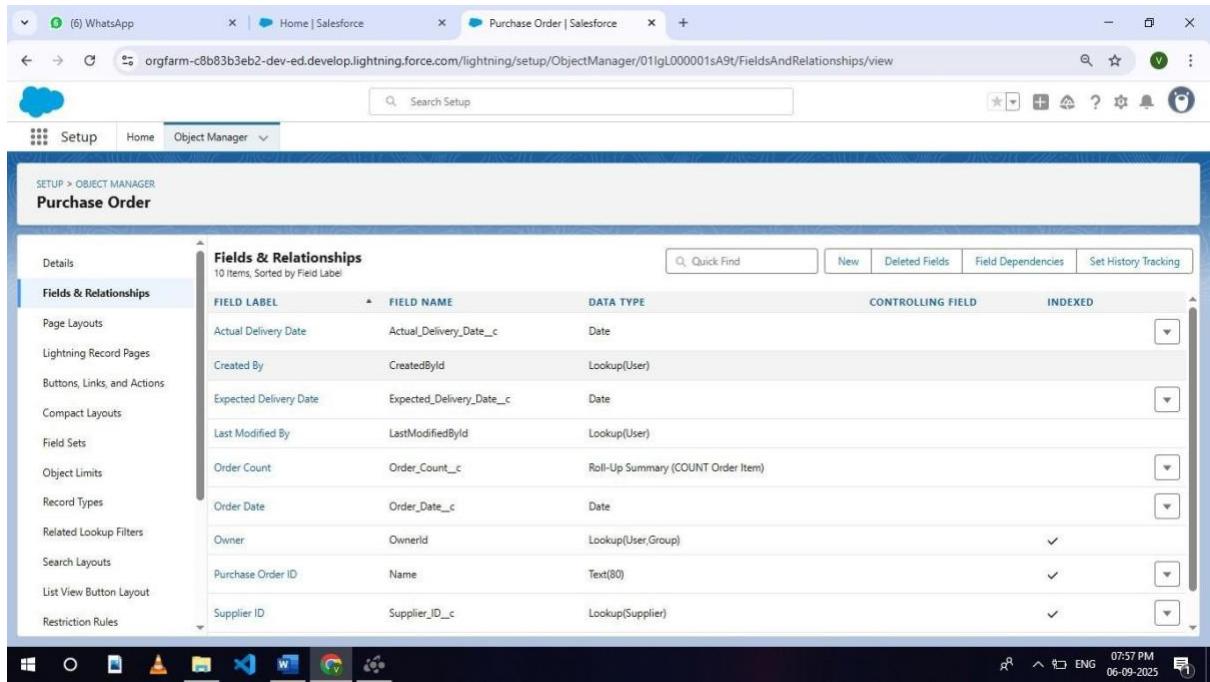
The screenshot shows the Salesforce Object Manager Fields & Relationships page for the Purchase Order object. The left sidebar lists various setup options like Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, etc. The main area displays a table of fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Actual Delivery Date	Actual_Delivery_Date__c	Date		
Created By	CreatedById	Lookup(User)		
Expected Delivery Date	Expected_Delivery_Date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Order Count	Order_Count__c	Roll-Up Summary (COUNT Order Item)		
Order Date	Order_Date__c	Date		
Owner	OwnerId	Lookup(User/Group)		
Purchase Order ID	Name	Text(80)		
Supplier ID	Supplier_ID__c	Lookup(Supplier)		

## **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.

The screenshot shows the Salesforce Setup interface for the Purchase Order object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the 'Fields & Relationships' section with a table of fields. The table includes columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. A new field, 'Order Count', has been added with the field name 'Order\_Count\_\_c', data type 'Roll-Up Summary (COUNT Order Item)', and no controlling field or index.

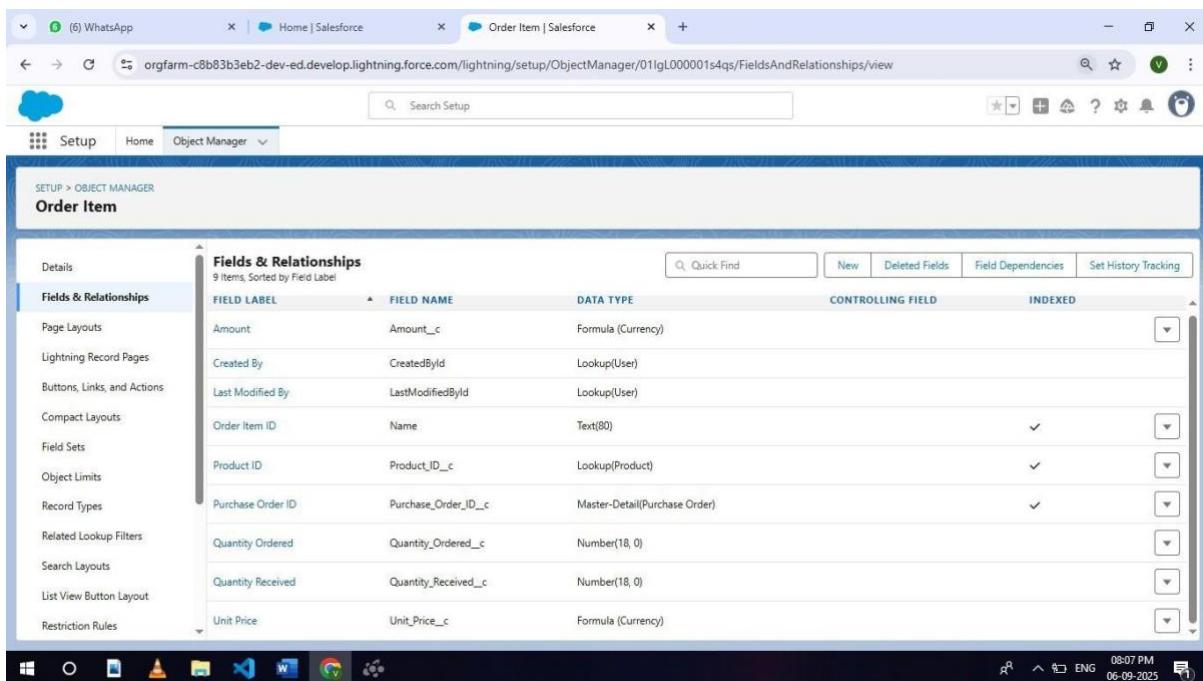
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Actual Delivery Date	Actual_Delivery_Date__c	Date		
Created By	CreatedById	Lookup(User)		
Expected Delivery Date	Expected_Delivery_Date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Order Count	Order_Count__c	Roll-Up Summary (COUNT Order Item)		
Order Date	Order_Date__c	Date		
Owner	OwnerId	Lookup(User,Group)		✓
Purchase Order ID	Name	Text(80)		✓
Supplier ID	Supplier_ID__c	Lookup(Supplier)		✓

## **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.



The screenshot shows the Salesforce setup interface. The top navigation bar includes tabs for WhatsApp, Home, and Order Item. The main title is "Object Manager" under "SETUP". The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, Buttons, etc. The main content area is titled "Fields & Relationships" for the "Order Item" object. It displays a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table lists nine fields: Amount, Created By, Last Modified By, Order Item ID, Product ID, Purchase Order ID, Quantity Ordered, Quantity Received, and Unit Price. The "Amount" field is a Formula (Currency) type, while others are Lookup types. The "Order Item ID" field is indexed. The bottom status bar shows system information: 08:07 PM, 06-09-2025, ENG, and battery level.

## 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.

The screenshot shows the Salesforce Object Manager interface for the 'Order Item' object. On the left, there's a sidebar with various setup options like Page Layouts, Lightning Record Pages, Buttons, etc. The main area is titled 'Fields & Relationships' and lists nine fields. The fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Formula (Currency)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Order Item ID	Name	Text(80)		✓
Product ID	Product_ID__c	Lookup(Product)		✓
Purchase Order ID	Purchase_Order_ID__c	Master-Detail(Purchase Order)		✓
Quantity Ordered	Quantity_Ordered__c	Number(18, 0)		
Quantity Received	Quantity_Received__c	Number(18, 0)		
Unit Price	Unit_Price__c	Formula (Currency)		

## 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.

The screenshot shows the Salesforce setup interface for the 'Object Manager'. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Fields & Relationships' for the 'Inventory Transaction' object. It displays a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table contains the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Inventory Transaction ID	Name	Text(80)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Purchase Order ID	Purchase_Order_ID_c	Lookup(Purchase Order)		✓
Total Order Cost	Total_Order_Cost_c	Formula (Currency)		
Transaction Date	Transaction_Date_c	Date		
Transaction Type	Transaction_Type_c	Picklist		

## 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.

The screenshot shows the Salesforce Setup interface with the following details:

- Page Header:** (6) WhatsApp, Home | Salesforce, orgfarm-c8b83b3eb2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001sABV/FieldsAndRelationships/view
- Search Bar:** Search Setup
- Navigation:** Setup, Home, Object Manager
- Section:** SETUP > OBJECT MANAGER, Inventory Transaction
- Table:** Fields & Relationships (8 items, Sorted by Field Label)
- Table Headers:** FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, INDEXED
- Table Data:**

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Inventory Transaction ID	Name	Text(80)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Purchase Order ID	Purchase_Order_ID_c	Lookup(Purchase Order)		✓
Total Order Cost	Total_Order_Cost_c	Formula (Currency)		
Transaction Date	Transaction_Date_c	Date		
Transaction Type	Transaction_Type_c	Picklist		
- System Status:** 08:12 PM, ENG, 06-09-2025

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 Salesforce Object Manager interface. The top navigation bar has tabs for Home, Object Manager, and Supplier. The main content area is titled 'Supplier' under 'OBJECT MANAGER'. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The 'Fields & Relationships' section is selected and displays a table with 9 items. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are Address, Contact Person, Created By, Email, Last Modified By, Owner, Phone Number, Supplier ID, and Supplier Name. The 'Supplier ID' field is highlighted with a yellow background.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text Area(255)		
Contact Person	Contact_Person_c	Text(10)		
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone Number	Phone_Number_c	Phone		
Supplier ID	Name	Text(80)		✓
Supplier Name	Supplier_Name_c	Text(10)		

## 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 Salesforce Object Manager interface for the 'Supplier' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area is titled 'Fields & Relationships' and displays a table of fields for the Supplier object. The table columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are Address, Contact Person, Created By, Email, Last Modified By, Owner, Phone Number, Supplier ID, and Supplier Name. The 'Email' field is highlighted, showing it has a data type of 'Email'. The status bar at the bottom indicates the date and time as 06-09-2025 08:20 PM.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text Area(255)		
Contact Person	Contact_Person_c	Text(10)		
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone Number	Phone_Number_c	Phone		
Supplier ID	Name	Text(80)		✓
Supplier Name	Supplier_Name_c	Text(10)		

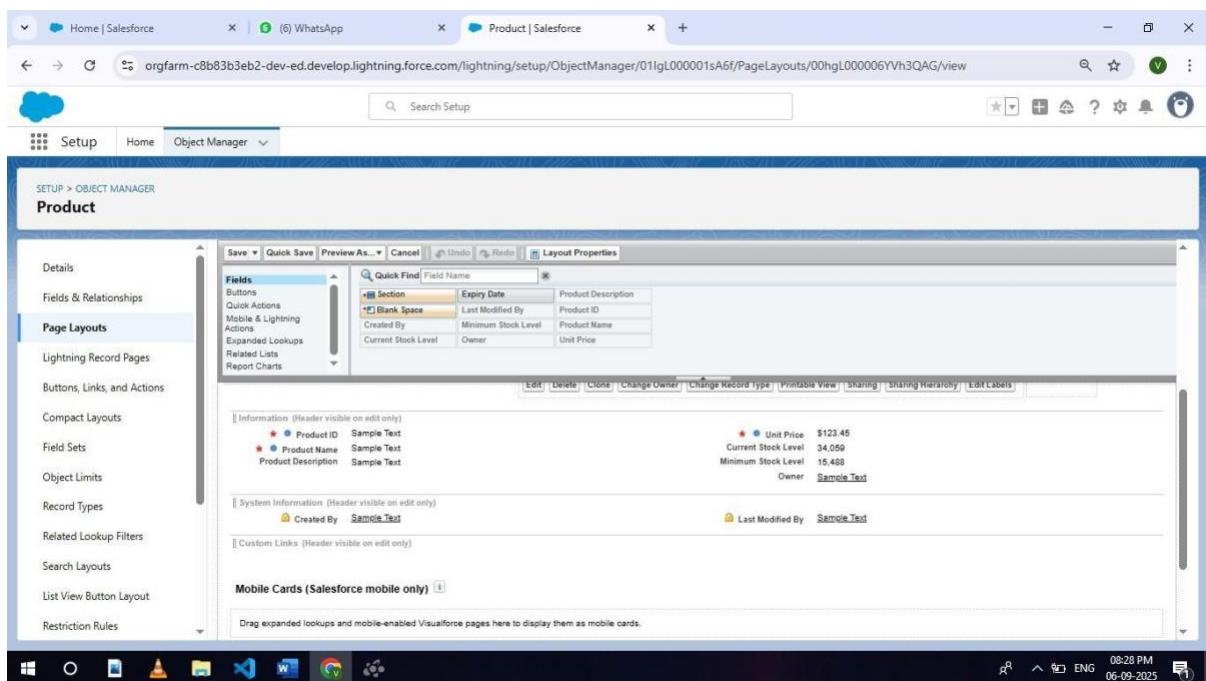
## Milestone 6 -Editing of Page Layouts

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.

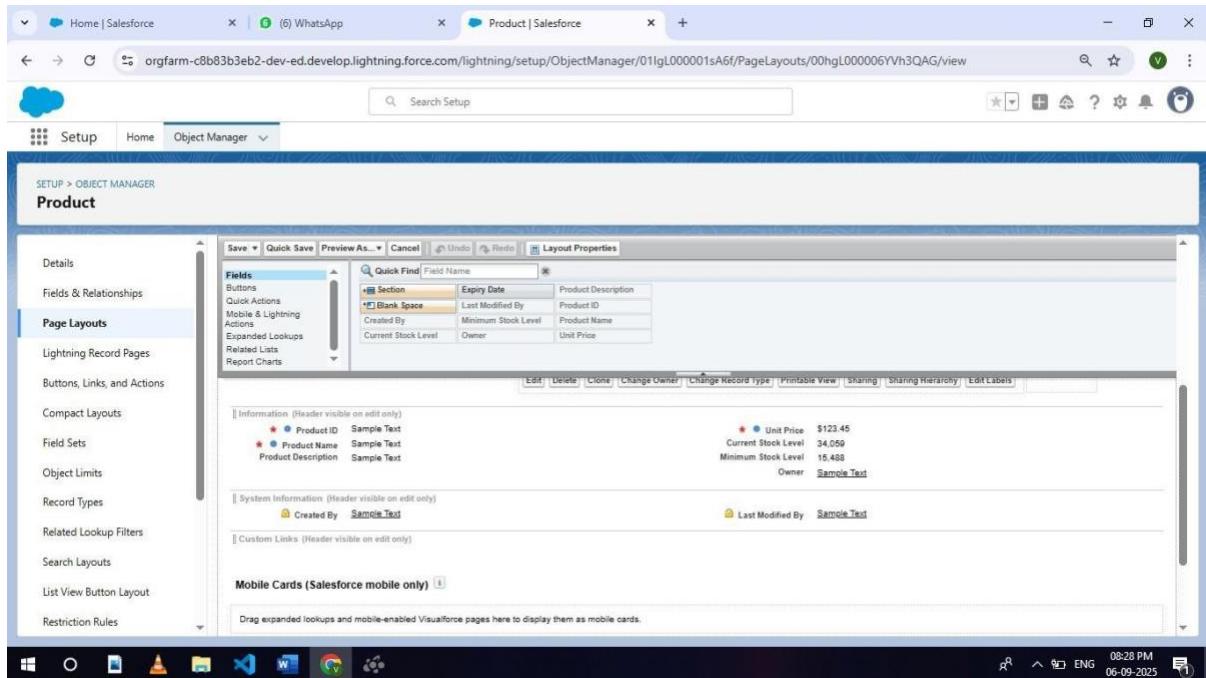
2. Click on the Purchase Order Layout

3. Drag and Arrange the field as shown below

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

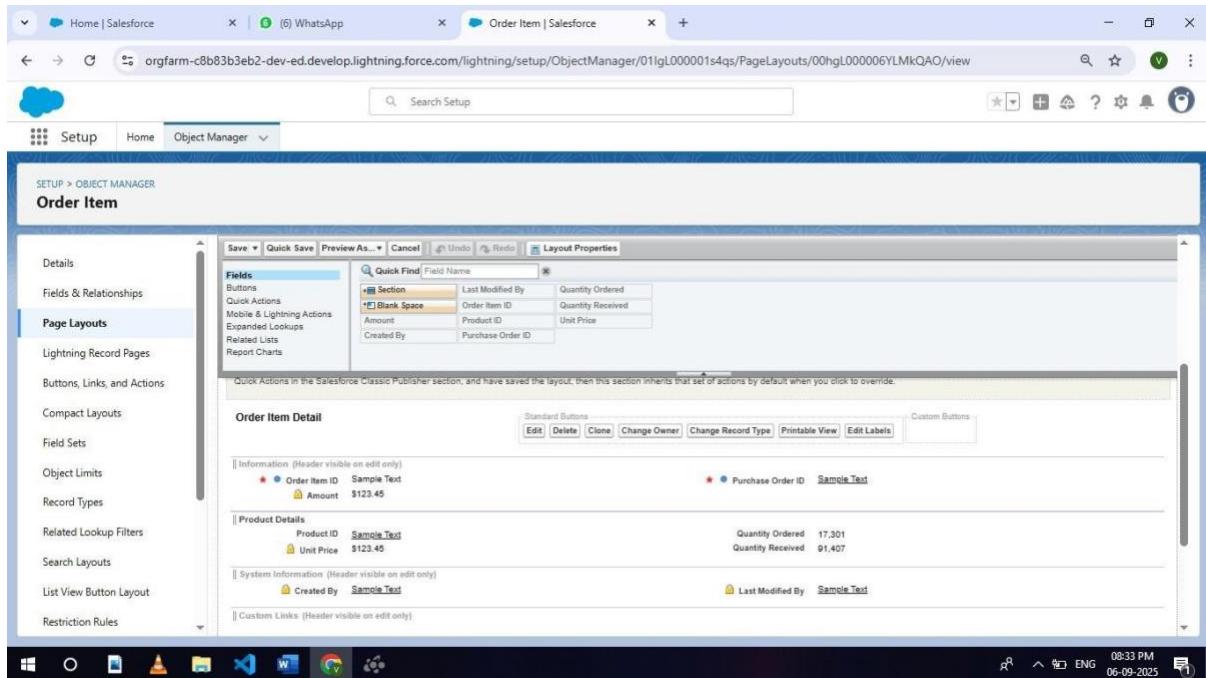
5. Click on field Total Order Cost >> click on settings >> select Read Only and save it.

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



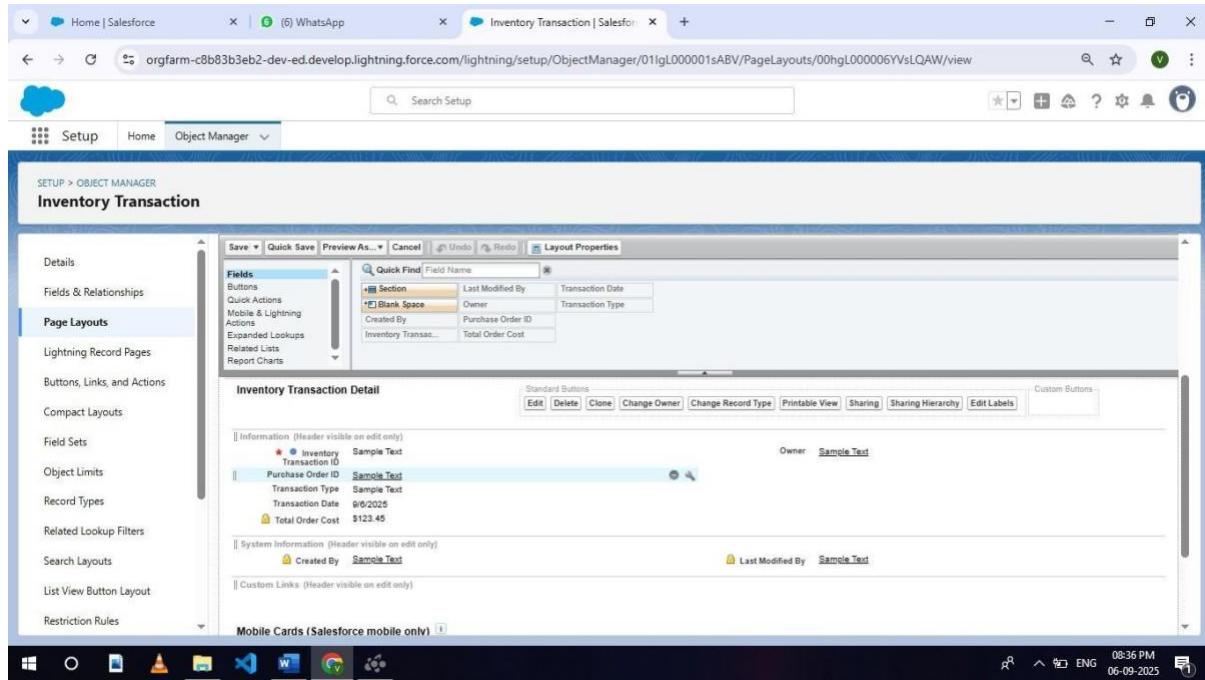
## Activity 4: To edit a Page Layout in Inventory Transaction Object

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.



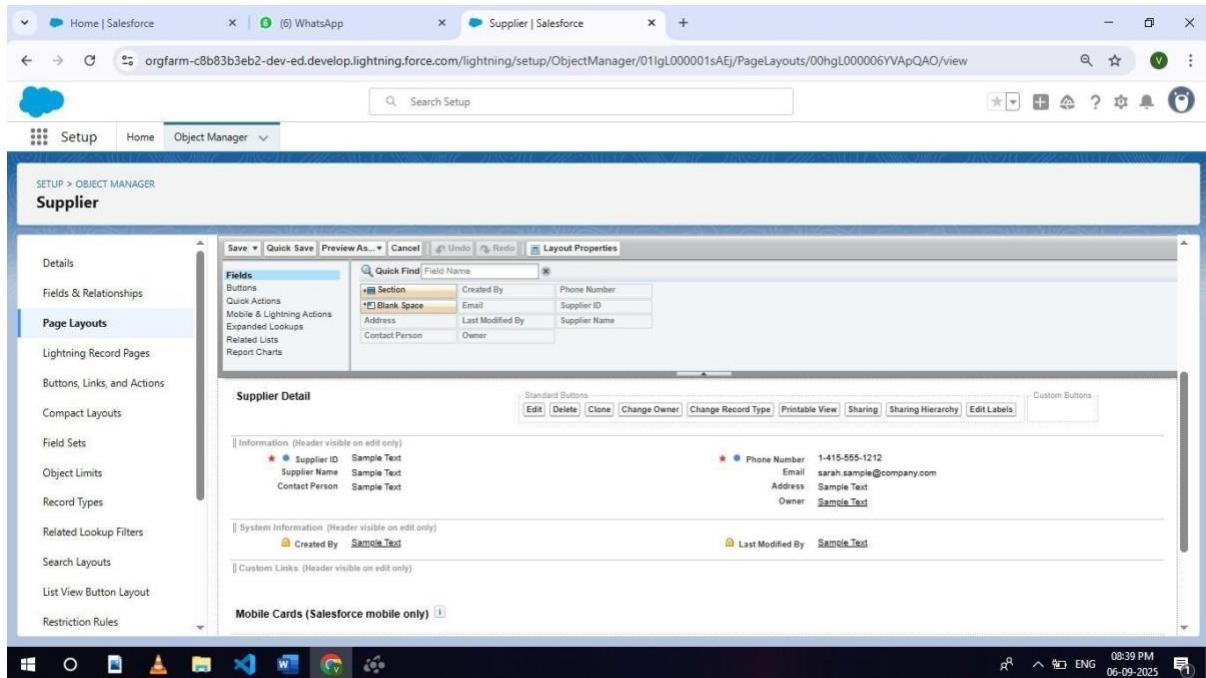
## Activity 5: To edit a Page Layout in Supplier Object

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.



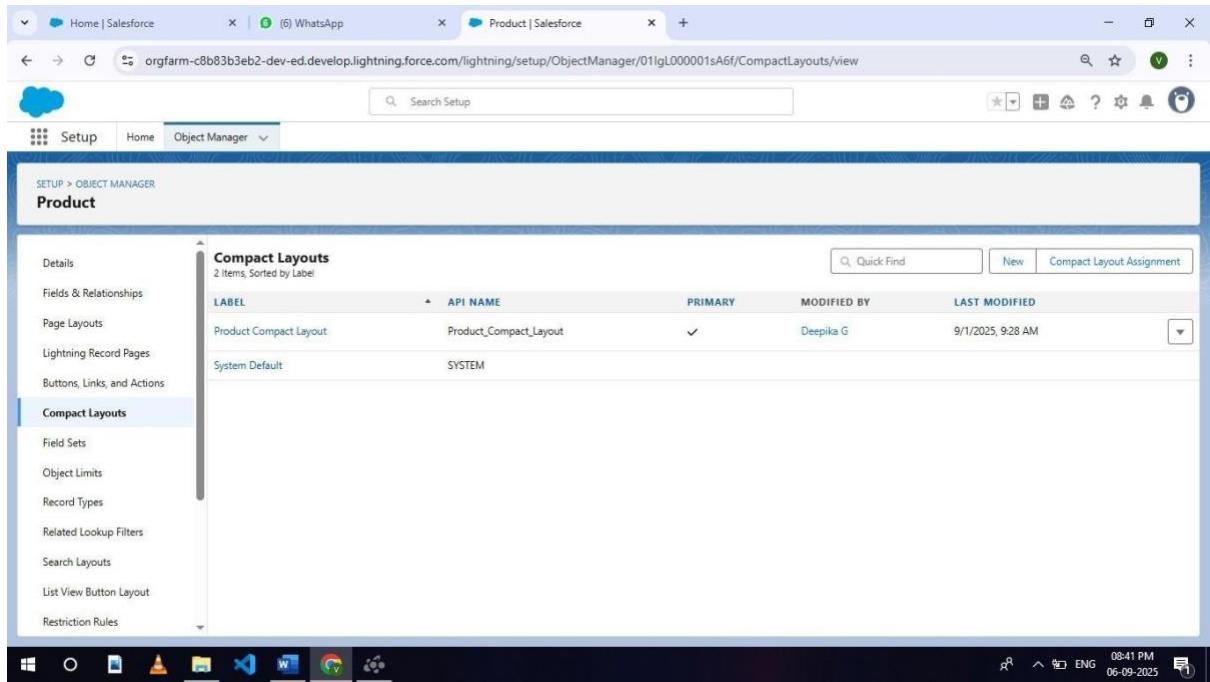
## Milestone 7 - Compact Layouts

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.



## 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 Salesforce Setup interface for the Purchase Order object. The 'Compact Layouts' tab is selected. The table displays the following data:

LABEL	API NAME	PRIMARY	MODIFIED BY	LAST MODIFIED
Purchase Order Compact Layout	Purchase_Order_Compact_Layout	✓	Deepika G	9/1/2025, 9:31 AM
System Default	SYSTEM			

## **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 :

(Expected\_Delivery\_Date\_\_c - Order\_Date\_\_c) > 7

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

7. Select the Error location as Top of Page

8. Click Save.

The screenshot shows the Salesforce Object Manager interface for the Purchase Order object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Validation Rules' and shows one item: 'Expected\_Delivery\_Date\_Validation'. This rule has 'Top of Page' selected as the error location and the message 'The Expected Delivery Date should not exceed 7 days.' It is marked as active and modified by Deepika G on 9/1/2025 at 9:34 AM. A 'New' button is visible in the top right corner of the validation rules table.

## **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.

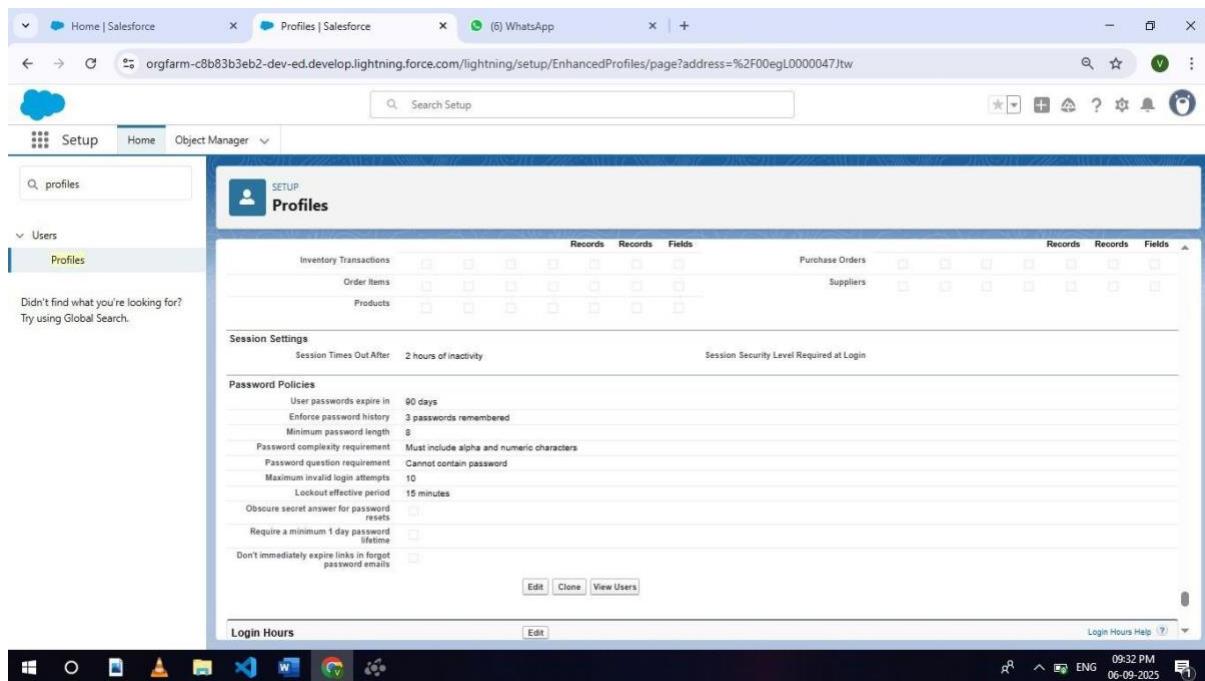
### **Activity 1: To create an Inventory Manager Profile**

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.

5. Change the password policies as mentioned :

6. User passwords expire in should be “ never expires ”.

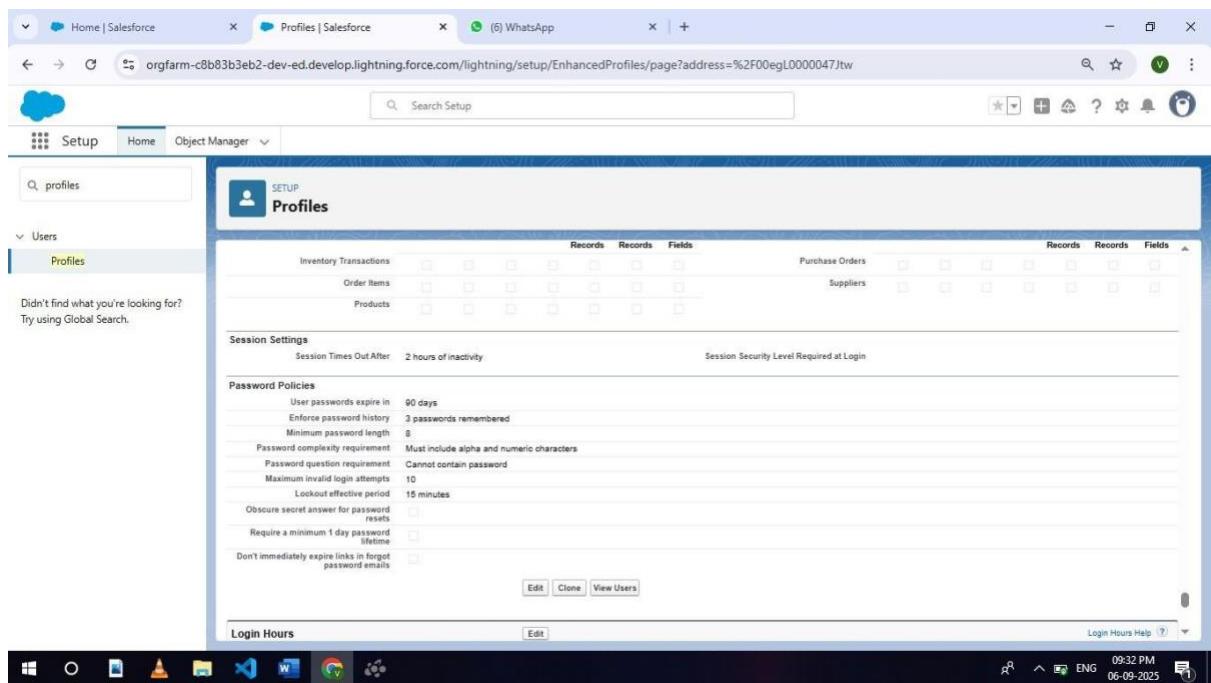
7. Minimum password length should be “ 8 ”, and click save.



## 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.
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.

## **Milestone 10 - Roles**

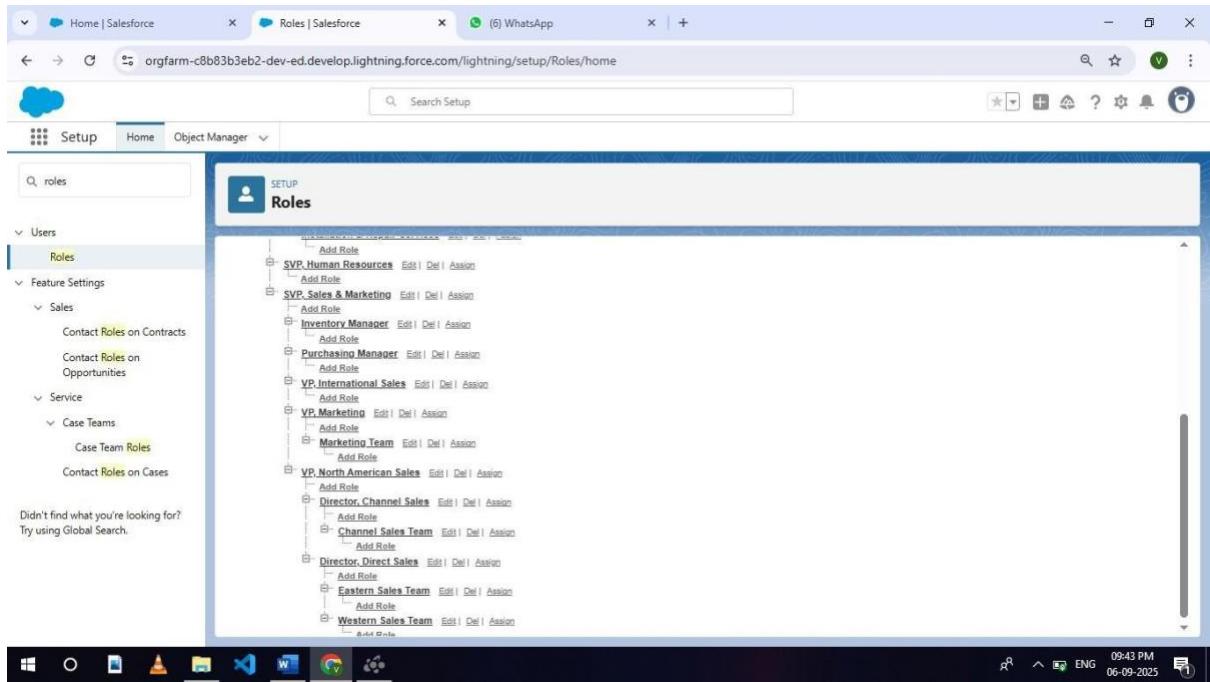
Roles in Salesforce are used to control record-level access and define the hierarchy of an organization, determining the level of visibility and sharing of records among users. Roles work in conjunction with profiles to provide a robust security model. While profiles control what actions users can perform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

### **Activity 1 : Create a Purchasing Manager Role.**

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

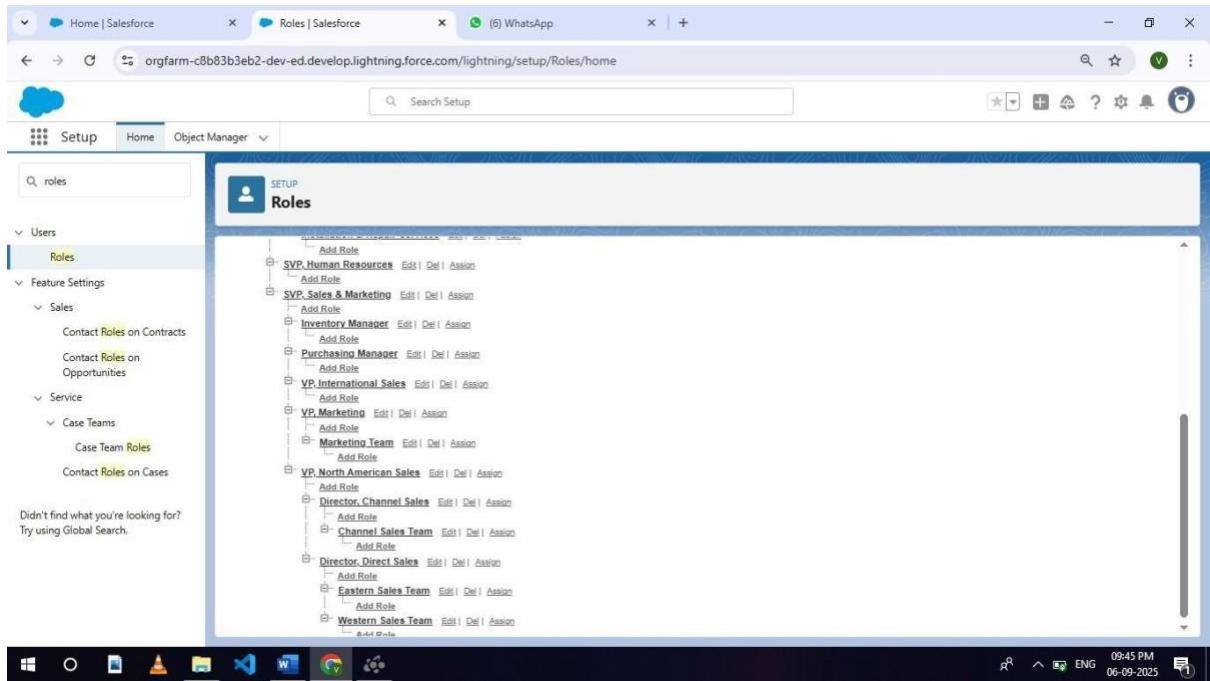
2. Click on Expand All and click on add role under SVP, Sales & Marketing role.

3. Give Label as “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.

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

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

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

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

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Home | Salesforce, Permission Sets | Salesforce, (6) WhatsApp
- Page Title:** Permission Sets
- Left Sidebar:** Q permission, Users, Permission Set Groups (Permission Sets is selected), Custom Code, Custom Permissions.
- Middle Content:** **Permission Sets** table:

Action	Permission Set Name	Description	License
<input type="checkbox"/>	Partner Connect Partner Admin Setup	Set up Partner Connect from a partner org. Partner Connect is a Par...	Salesforce
<input type="checkbox"/>	Payments Administrator	Has all the user permissions to gate access to APIs that are availab...	Salesforce Payments Internal
<input type="checkbox"/>	Prompt Template Manager	Manage prompt templates using Prompt Builder and run them using ...	Einstein Prompt Templates
<input type="checkbox"/>	Prompt Template User	Run prompt templates using generative AI features.	Einstein Prompt Templates
<input type="checkbox"/>	PromptTemplatePermSet		Cloud Integration User
<input type="checkbox"/>	Publish Suggested for You Nudges: Integration User	Access the Core Adoption Service and tenant orgs, which are used ...	Cloud Integration User
<input type="checkbox"/>	Purchase Manager Create Access		
- Bottom:** Page 1 of 1, 0 Selected, Previous, Next, 09:49 PM, 06-09-2025.

## 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 Entry Conditions : None
6. Select Fast Field Updates and click on Done

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

8.Enter Label as “ Get Purchase Record ”.

9.For Object select Purchase Order.

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

For the first condition select as follows:

Field: Id

Operator: Equals

Value: {!\$Record.Id}

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

12. For How to Store Record Data select Choose fields and let Salesforce do the rest. Select Field: Order\_Date\_\_c. Click on Done.

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

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

15. From the Toolbox drag and drop Assignment element.

16. Enter the label as “Assignment”.

17. Set Variable Values:

a) Variable : {!ActualDeliveryDate}

Operator : Equals

Value : {!\$Record.Order\_Date\_\_c}

b) Variable : {!ActualDeliveryDate}

Operator : Add

Value : 3

18. Click Done

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

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

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

22. Set Filter Conditions : None -Always Update Record

23. Set Field Values for the Trip Record as

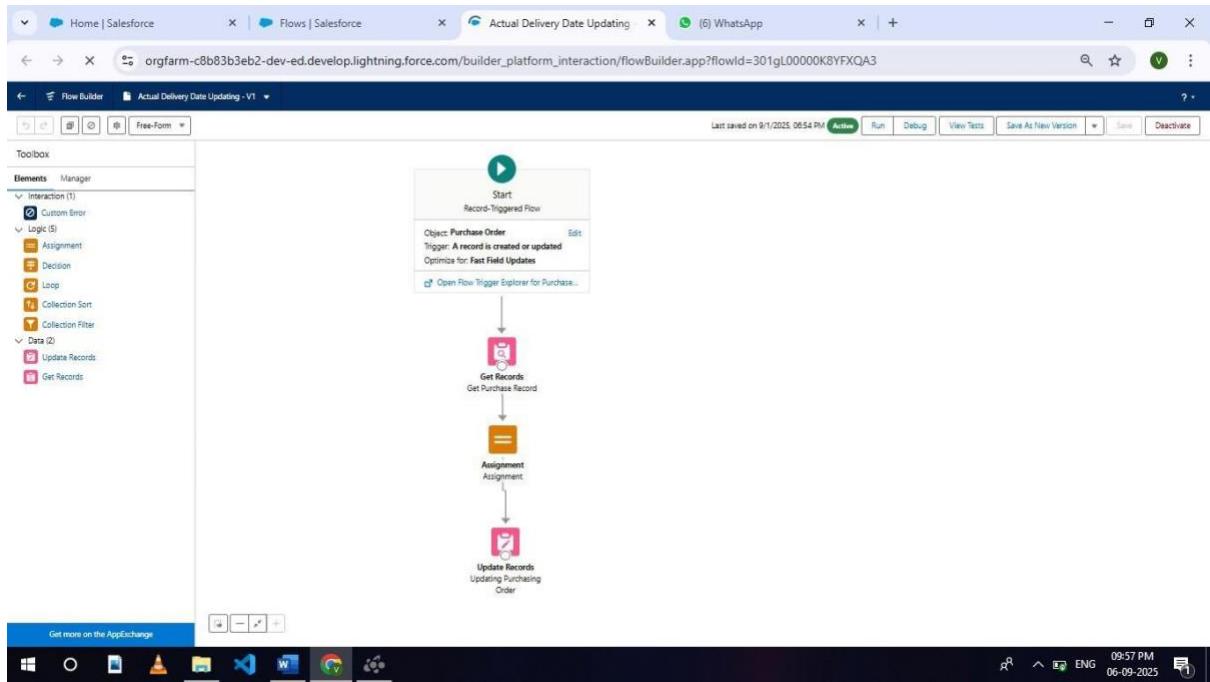
Field : Actual\_Delivery\_Date\_\_c

Value : {!ActualDeliveryDate}

24. Click Done

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

26. Activate the flow.



## 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.ne  
w, 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;
```

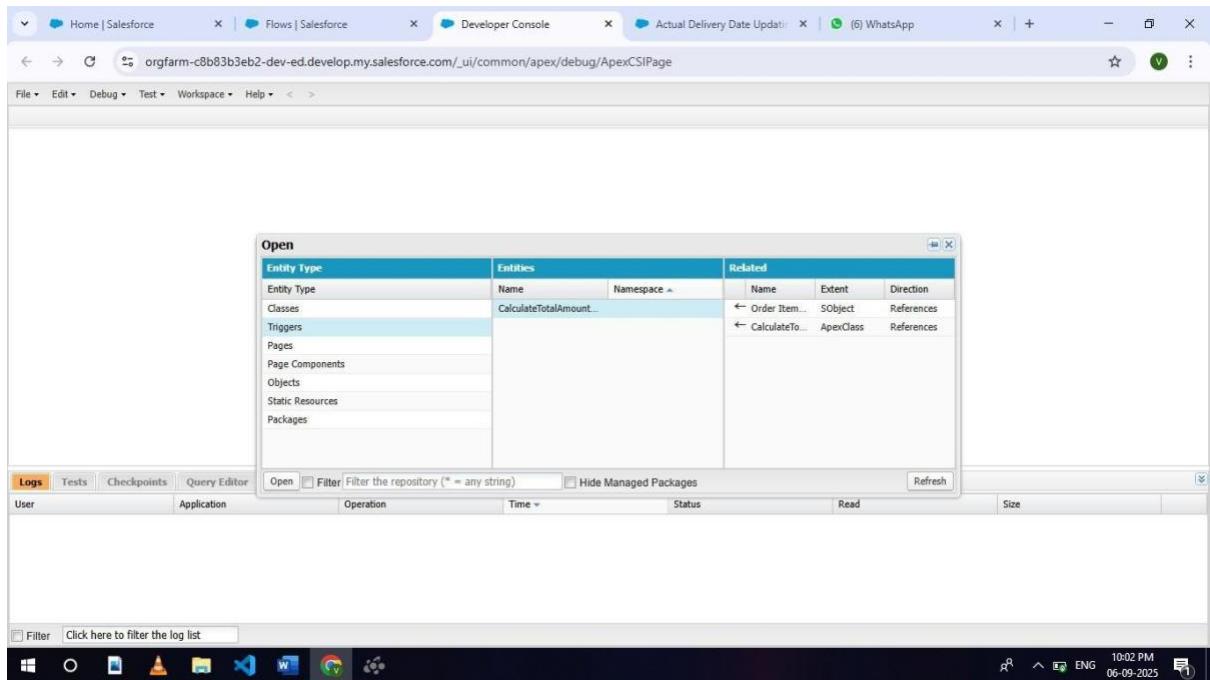
```
}
```

```
}
```

}

}

Save it.



```
1 public class CalculateTotalAmountHandler {  
2  
3  
4 // Method to calculate the total amount for Purchase Orders based on related Order Items  
5  
6 public static void calculateTotal(List<Order_Item__c> newItems, List<Order_Item__c> oldItems, Boolean isInsert, Boolean isUpdate)  
7  
8  
9  
10 // Collect Purchase Order IDs affected by changes in Order_Item__c records  
11  
12 Set<Id> parentIds = new Set<Id>();  
13  
14 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size

Filter Click here to filter the log list

## 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.

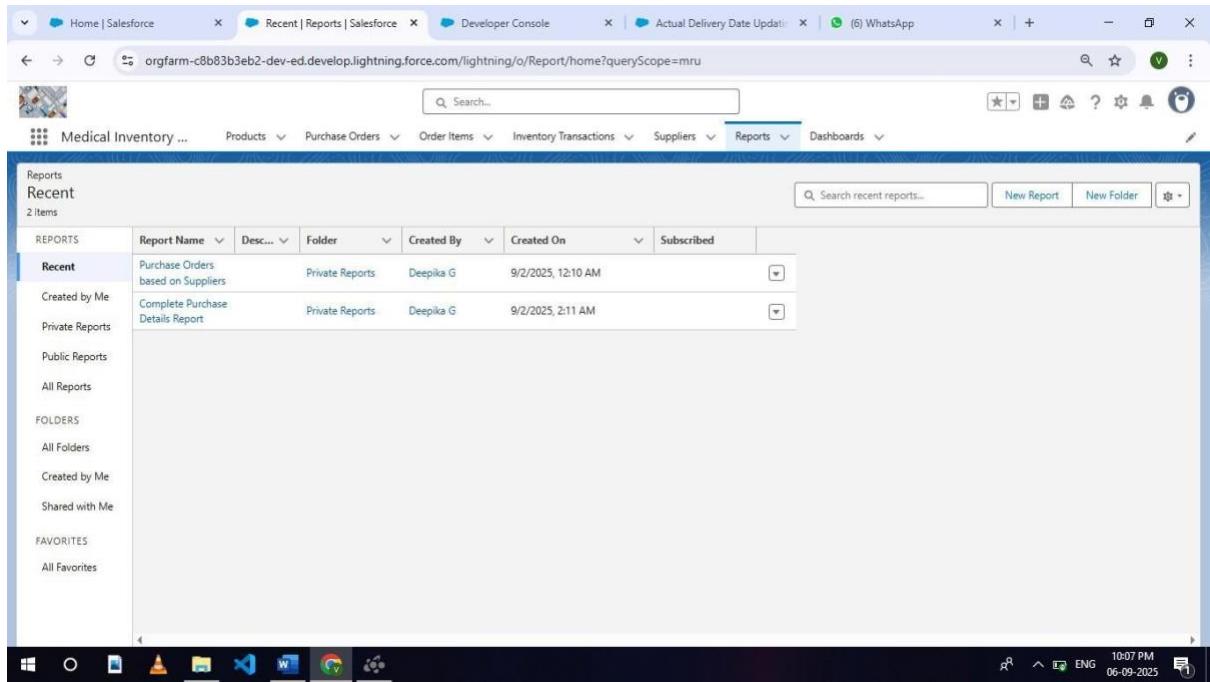
## 10. Click Save

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”)

### View Report

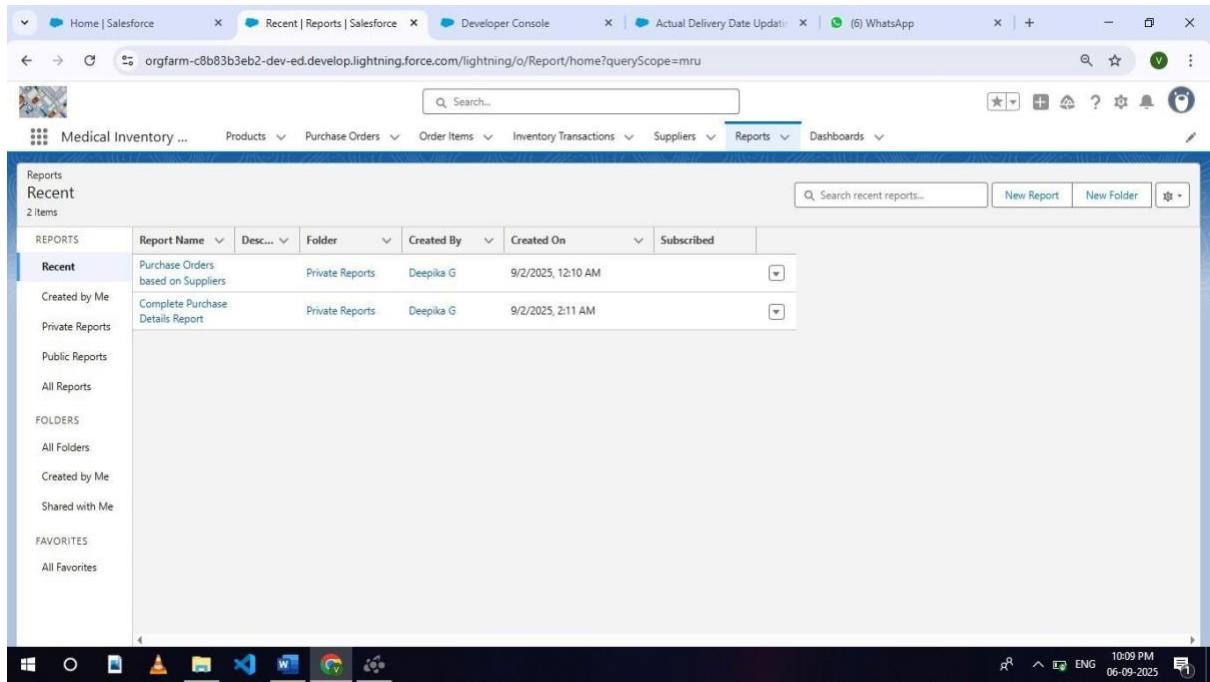
1. Click on App Launcher on the left side of the screen.
2. Search Medical Inventory Management App & click on it.
3. Click on Reports Tab.
4. Click on Purchase Orders based on Suppliers and see records.



## 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
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.



## 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

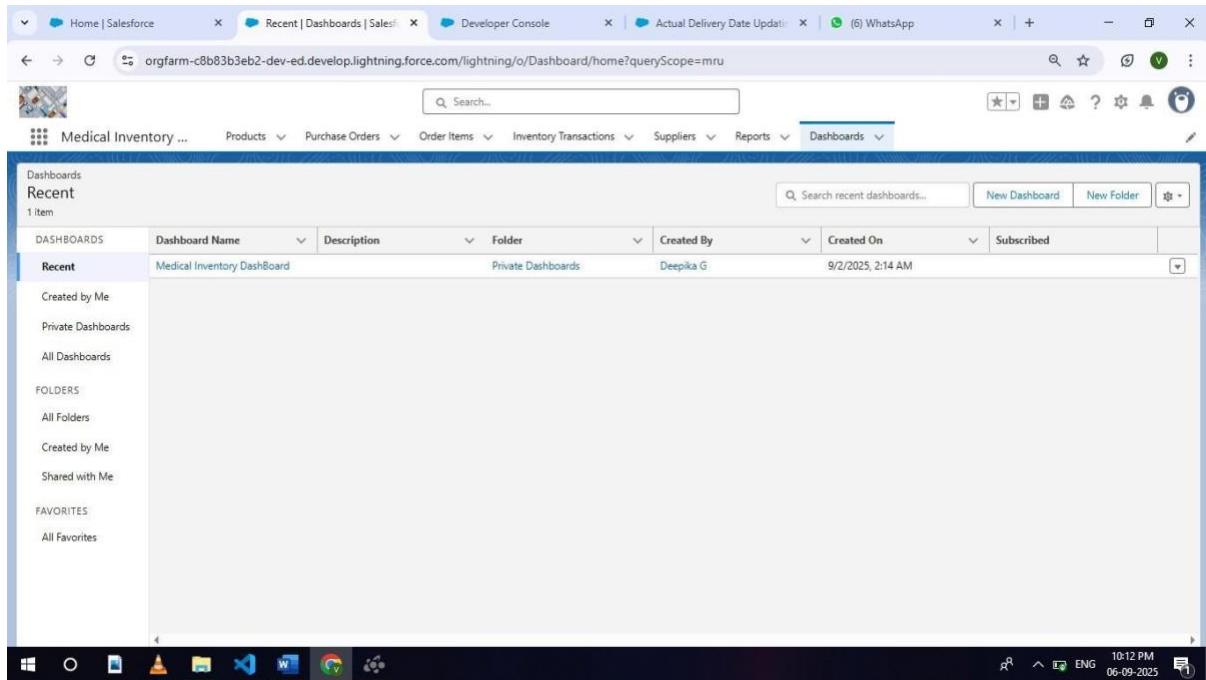
5. Click on +widget

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

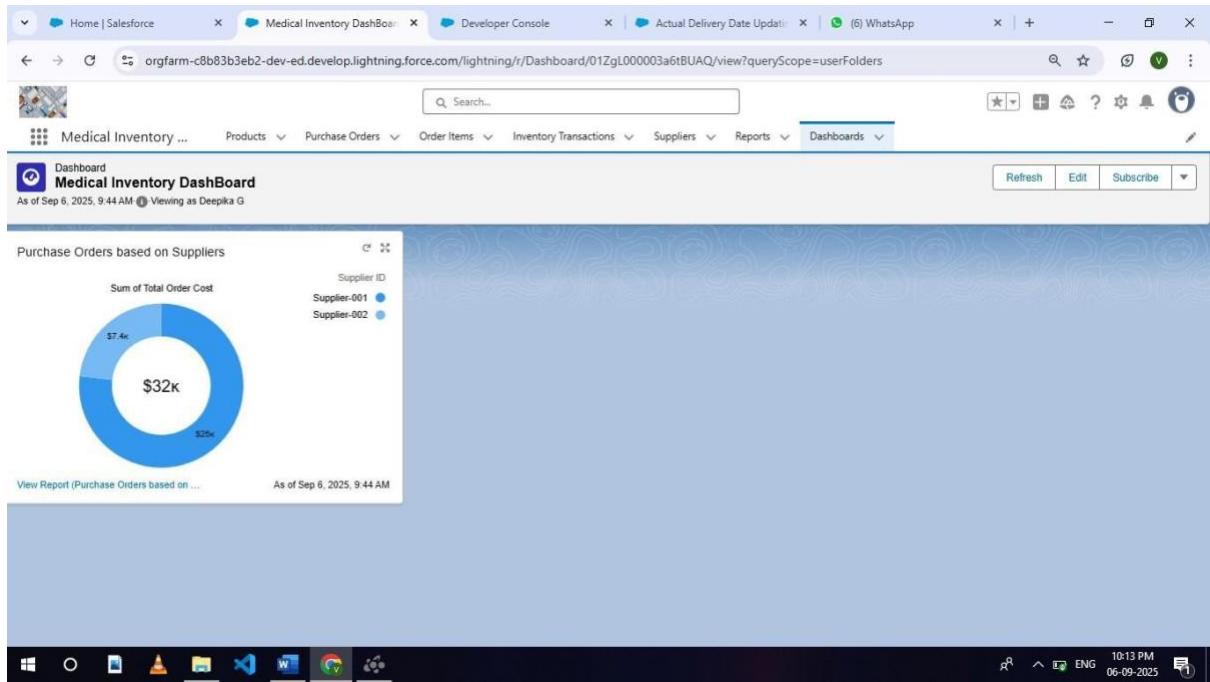
8. Click add.

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 :-

By leveraging the Salesforce platform, the project successfully created a robust and transparent system for managing medical inventory. Through streamlined tracking of medicines, equipment, and supplies, the solution enabled healthcare providers, donors, and NGOs to coordinate efficiently, thereby reducing wastage and ensuring timely availability of critical resources.

The project “Medical Inventory Management using Salesforce” has been successfully implemented and

demonstrates how Salesforce CRM can be utilized to address real-world healthcare challenges by ensuring efficiency, accountability, and scalability.

- **Project Achievements:**

- Designed a centralized system for tracking medical supplies, donations, and usage.
- Enabled real-time monitoring of stock levels with alerts for low or expiring inventory.
- Automated workflows using custom objects, Flows, and Apex triggers to handle requests and approvals.
- Improved transparency with reports and dashboards for stock movement and utilization.
- Enhanced security with role-based access, ensuring data privacy and compliance.

- **Student Learning Outcomes:**

- Hands-on experience in Salesforce development with a healthcare-focused use case.

- Enhanced analytical and problem-solving skills for inventory optimization.
- Practical understanding of requirement gathering, system design, and testing.
- Team collaboration across different stages of the project lifecycle.
- Exposure to CRM customization and industry-relevant best practices.

## **Future Scope:**

- Integration with IoT devices for real-time inventory tracking in warehouses and hospitals.
- Use of AI/ML for demand forecasting and automatic replenishment of medical supplies.
- Expansion to integrate with external healthcare systems (ERP/Hospital Management Systems).
- Mobile application support for on-the-go updates and approvals.
- Collaboration with pharmaceutical companies, NGOs, and government bodies for larger reach.

