

CRM APPLICATION FOR Medical Inventory Management

College Name :VET INSTITUTE OF ARTS AND SCIENCE

College Code:65651

Team ID:NM2025TMID25710

Team Size: 5

Team Leader: PRAKASH MEENASHISUNDARAM

EMail : prakashmeenashisundaram23sit@vetias.ac.in

Team member: GOWTHAM R

EMail : gowthamr23sit@vetias.ac.in

Team member: NIRMALKUMAR B

Email : nirmalkumarb23sit@vetias.ac.in

Team member: INDHIRAJITH P V

EMail : indhirajithpv23sit@vetias.c.in

Team member: DEEPAN P

EMail : deepanp23sit@vetias.ac.in

Medical Inventory Management

[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:

The screenshot shows the 'Sign up for your Developer Edition' page. The header features the Salesforce logo and the text 'Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.' Below this, there's a list of benefits: 'Build apps fast with drag-and-drop tools', 'Go further with Apex code', 'Build AI agents with Agentforce', 'Harmonize your data with Data Cloud', 'Ground Agentforce with structured and unstructured data', and 'Integrate with anything using APIs'. A cartoon robot character is on the left, pointing towards a 'Customer Support' button. The main form area contains fields for First name (Prakash), Last name (Meenashisundaram), Job title (student), Work email (prakashmeenashisun@), Company (VET Institute of Arts), and Country/Region (India). A checkbox for agreeing to the Main Services Agreement is checked. At the bottom, there's a reCAPTCHA field with the message 'We value your privacy. To learn more, visit our Privacy Statement.'

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 neednot be an actual email id, you can give anything in the format

: username@organization.com

Click on sign me up after filling these.

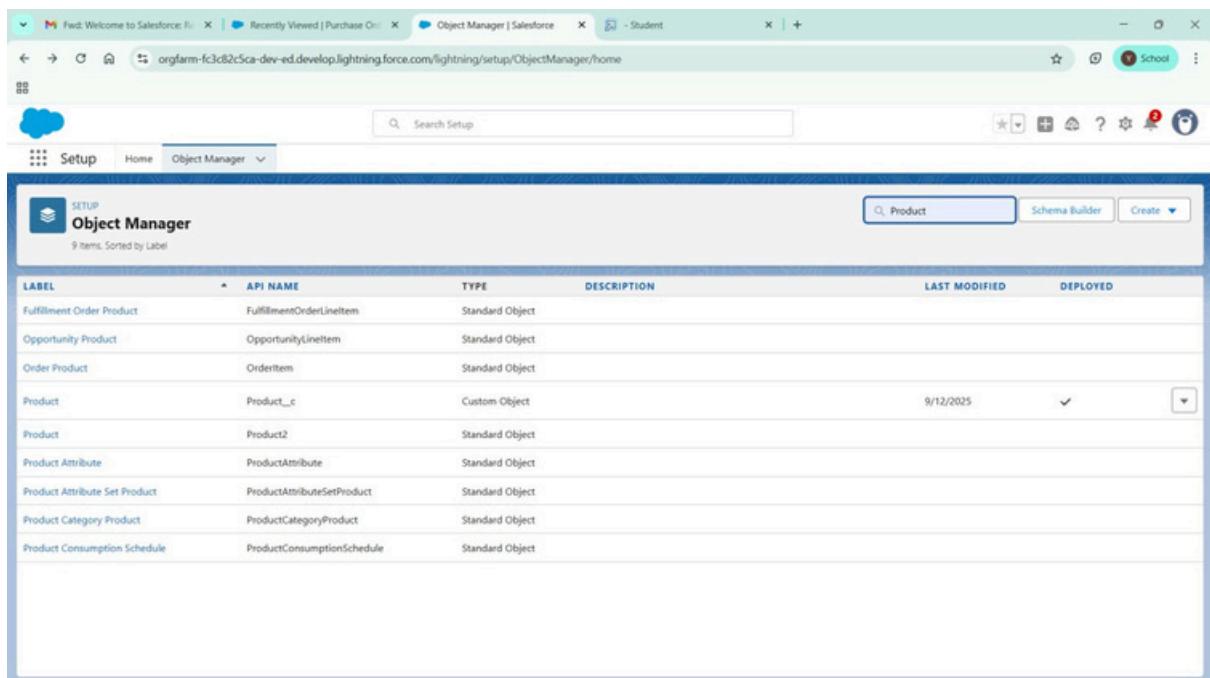
Creating a Product Object

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 Salesforce Object Manager interface. At the top, there's a navigation bar with links for Setup, Home, and Object Manager. The main title is "Object Manager" with a "SETUP" icon. Below the title, it says "9 items. Sorted by Label". There's a search bar with "Product" typed in and a "Create" button. A table lists the custom objects:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Fulfillment Order Product	FulfillmentOrderLineItem	Standard Object			
Opportunity Product	OpportunityLineItem	Standard Object			
Order Product	OrderItem	Standard Object			
Product	Product_c	Custom Object		9/12/2025	✓
Product	Product2	Standard Object			
Product Attribute	ProductAttribute	Standard Object			
Product Attribute Set Product	ProductAttributeSetProduct	Standard Object			
Product Category Product	ProductCategoryProduct	Standard Object			
Product Consumption Schedule	ProductConsumptionSchedule	Standard Object			

The screenshot shows the Salesforce Object Manager interface. The left sidebar is titled 'Object Manager' and lists various setup options: 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, Restriction Rules, and Scoping Rules. The main content area is titled 'Fields & Relationships' and displays a table of 12 items, sorted by Field Label. The table columns are Field Label, API Name, and Data Type. The fields listed are:

Field Label	API Name	Data Type
Expected Delivery Date	Expected_Delivery_Date__c	Date
Expiry Date	Expiry_Date__c	Date
Last Modified By	LastModifiedById	Lookup(User)
Minimum Stock Level	Minimum_Stock_Level__c	Number(10, 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)
Transaction Type	Transaction_Type__c	Picklist
Unit Price	Unit_Price__c	Currency(16, 2)

The screenshot shows the 'Product Id' field details page. The left sidebar is identical to the previous screenshot. The main content area is titled 'Product Id' and includes a 'Help for this Page' link. It has two sections: 'Field Information' and 'Validation Rules'. The 'Field Information' section shows the field's properties: Field Label is 'Product Id', Data Type is 'Text(80)', and it is the primary key ('Name'). The 'Validation Rules' section indicates 'No validation rules defined.'

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

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

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page open. The left sidebar is collapsed, and the main content area displays the 'Custom Tabs' section. The 'Custom Object Tabs' table lists five tabs:

Action	Label	Tab Style	Description
Edit Del	Inventory Transactions	Alarm clock	
Edit Del	Order Items	Airplane	
Edit Del	Products	Telescope	
Edit Del	Purchase Orders	Airplane	
Edit Del	Suppliers	Alarm clock	

Below this, there are sections for 'Web Tabs' and 'Visualforce Tabs', both of which currently have no entries defined.

The screenshot shows the Salesforce Setup interface. On the left, a sidebar lists various settings under 'Feature Settings' (Analytics, Tableau, User Interface) and 'Tableau' (Tableau Embedding, Tableau UAF Claims Definition). Under 'User Interface', 'Console Settings' is expanded, showing 'Loaded Console Tab Limit' and 'Rename Tabs and Labels'. The 'Tabs' option is selected. In the main area, a search bar at the top has 'tab' typed into it. Below the search bar, a 'SETUP' button and a 'Tabs' icon are visible. The main content area is titled 'Edit Custom Object Tab Products'. It contains a 'Custom Tab Definition Edit' section with 'Custom Object Tab Information'. The 'Tab Label' is set to 'Products', 'Object' to 'Product', and 'Tab Style' is 'Stethoscope'. A note says '(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.' A dropdown menu shows 'None'. Below this is a 'Description' field with a large empty text area. At the bottom are 'Save' and 'Cancel' buttons.

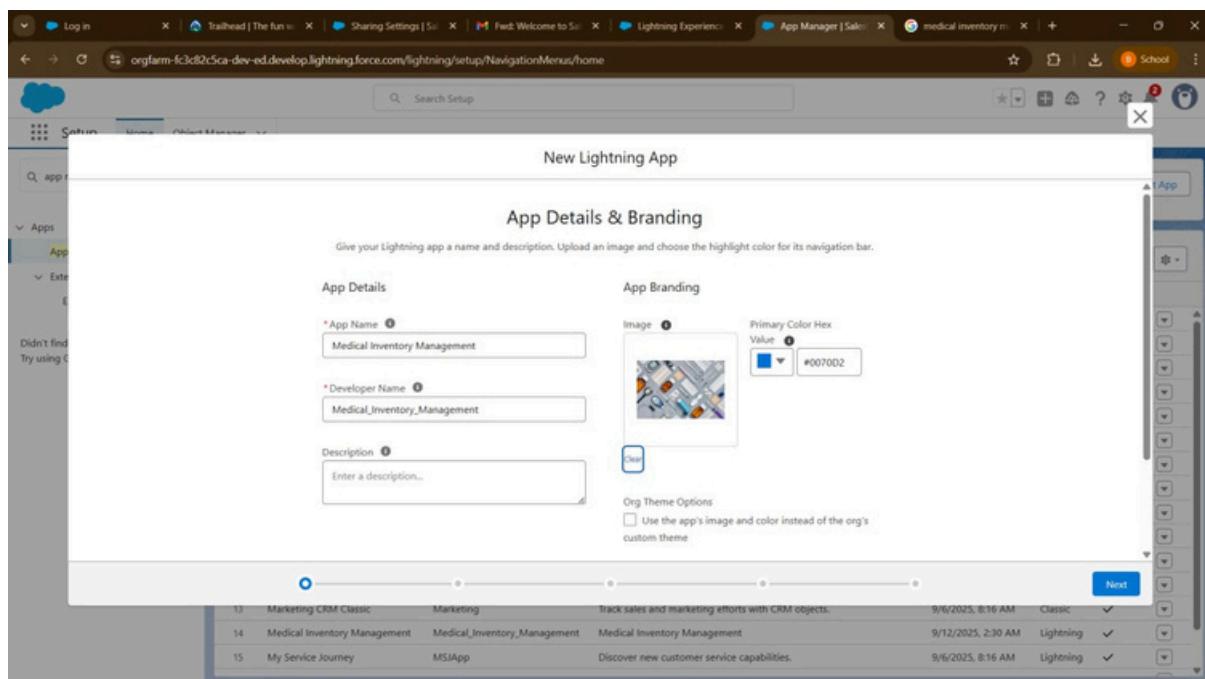
This screenshot shows the same Salesforce Setup interface after the tab has been created. The sidebar and search bar are identical to the previous screenshot. The main content area now displays the 'Custom Object Tab Products' page. It shows a 'Custom Tab Definition Detail' section with the same information: Tab Label 'Products', Object 'Product', Tab Style 'Stethoscope', and a note about the optional splash page link. The 'Created By' field shows 'Prakash Meenashisundaram' and the date '9/6/2025, 11:27 PM'. The 'Modified By' field also shows 'Prakash Meenashisundaram' and the same date. The 'Edit' and 'Delete' buttons are visible above the table.

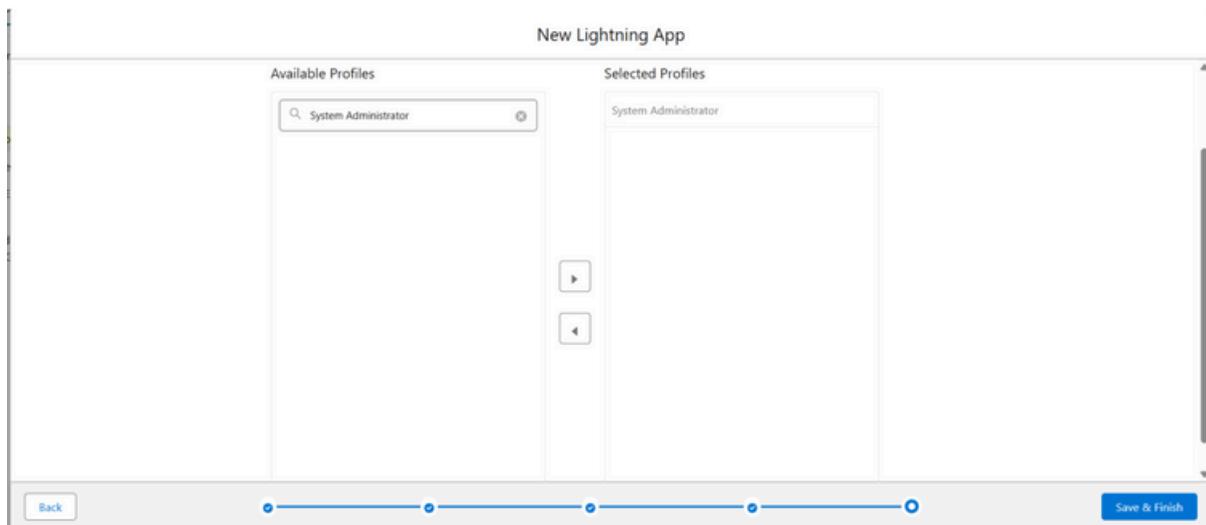
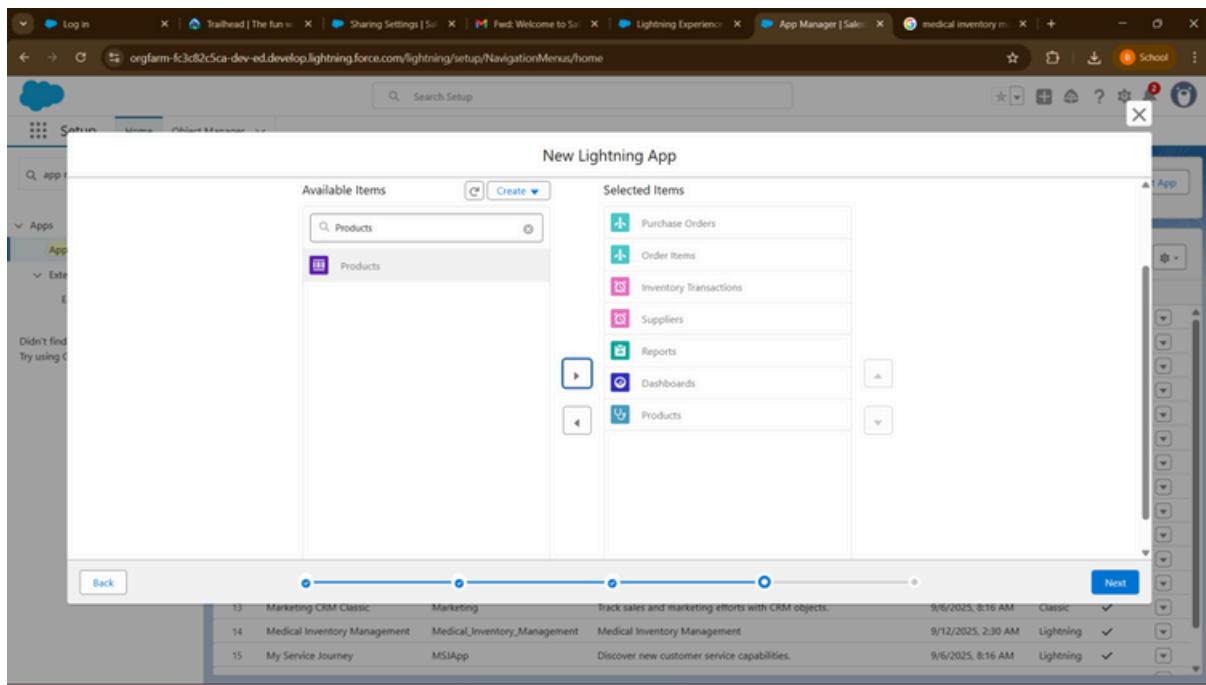
Creating Remaining Tabs

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 .

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.





Creating a Text Field in Product Object

To create fields in an object:

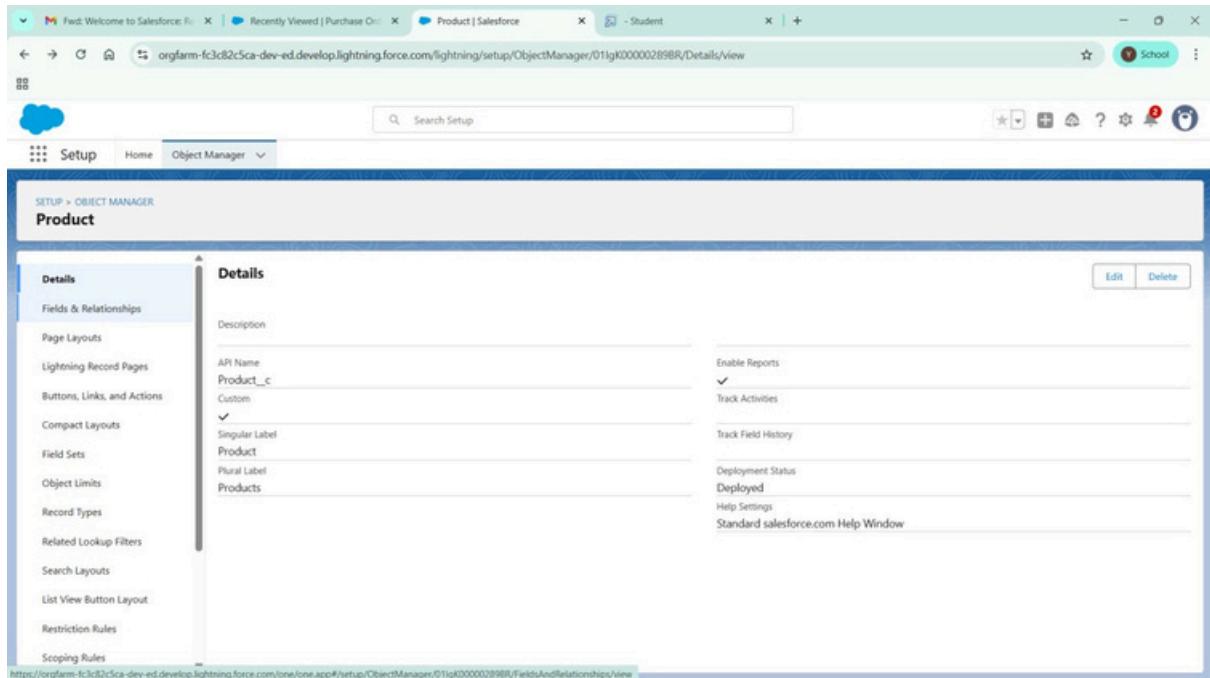
1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select Product custom object.
4. Select Fields & Relationships from the left navigation
5. Click on New
6. Select Text field, click Next
7. Enter Field Label as “Product Name” and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

The screenshot shows the Salesforce Object Manager page. At the top, there's a navigation bar with tabs for Setup, Home, and Object Manager. The main area is titled "Object Manager" and displays a list of 158+ items, sorted by Label. The list includes various standard and custom objects such as Price Book Entry, Privacy RTBF Request, Problem, Problem Related Item, Process Exception, Product, Product2, Product Attribute, Product Attribute Set Product, Product Category Product, Product Consumption Schedule, Promotion, Promotion Market Segment, and Promotion Qualifier. Each item in the list has three columns: Name, Type, and Status (with a dropdown arrow). The "Product" row is highlighted, showing it is a Custom Object. The status dropdown for "Product" is set to "9/12/2025".

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.



The screenshot shows the Salesforce Object Manager interface for the 'Product' object. The left sidebar lists various setup categories like Details, 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, Restriction Rules, and Scoping Rules. The main content area is titled 'Fields & Relationships' and displays 12 items sorted by Field Label. Each item includes the field name, its label, and its type. The fields listed are: Expected Delivery Date (Expected_Delivery_Date__c, Date), Expiry Date (Expiry_Date__c, Date), Last Modified By (LastModifiedById, Lookup(User)), Minimum Stock Level (Minimum_Stock_Level__c, Number(10, 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)), Transaction Type (Transaction_Type__c, Picklist), and Unit Price (Unit_Price__c, Currency(16, 2)). A search bar at the top right allows users to quickly find specific fields.

The screenshot shows the Salesforce Setup interface for the Object Manager. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Object Limits. The main content area is titled "Product Description" under "Product Custom Field". It displays the "Custom Field Definition Detail" for the "Product Description" field. The field information includes:

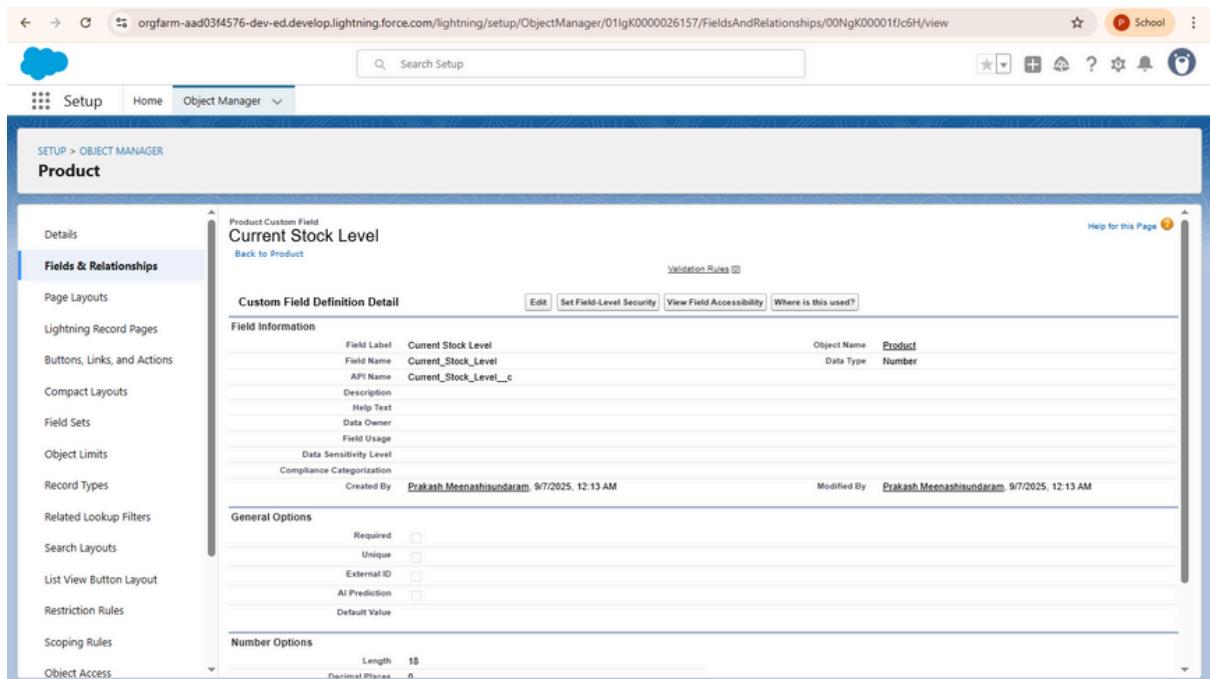
	Field Label	Product Description	Object Name	Product
Field Name	Product_Description	Data Type	Text Area	
API Name	Product_Description__c			
Description				
Help Text				
Data Owner				
Field Usage				

Below this, the "General Options" section shows "Required" checked and "Default Value" as an empty input field. The "Validation Rules" section indicates "No validation rules defined." At the bottom, there are links to "Back To Top" and "Always show me more records per related list".

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.



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. Select Data type as “Currency”
and click Next. Enter Field Label as “ Unit
- 4 Price”. Length - 16, Decimal Places - 2. Select
Required Field. Click on Next, Next and Save.

Product Custom Field
Unit Price

[Back to Product](#)

[Help for this Page](#)

Custom Field Definition Detail

[Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#) [Where is this used?](#)

Field Information	Object Name	Product
Field Label: Unit Price	Data Type:	Currency
Field Name: Unit_Price		
API Name: Unit_Price_c		
Description:		
Help Text:		
Data Owner:		
Field Usage:		
Data Sensitivity Level:		
Compliance Categorization:		
Created By: Prakash Meenashisundaram, 9/7/2025, 12:15 AM	Modified By: Prakash Meenashisundaram	9/7/2025, 12:15 AM

General Options

Required:

Default Value:

Currency Options

Length: 16
Decimal Places: 2

Validation Rules

[New](#)

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.

Purchase Order

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Actual_Delivery_Date	Actual_Delivery_Date_c	Date/Time		
Created_By	CreatedById	Lookup(User)		
Last_Modified_By	LastModifiedById	Lookup(User)		
Order_Date	Order_Date_c	Date		
Product	Product_c	Master-Detail(Product)	✓	
Purchase_Order_Name	Name	Text(80)	✓	
Supplier_ID	Supplier_ID_c	Lookup(Supplier)	✓	
Total_Order_Cost	Total_Order_Cost_c	Currency(10, 2)		

Purchase Order Custom Field

Supplier ID

Custom Field Definition Detail

Field Information	Object Name	Purchase_Order
Field Label: Supplier ID	Object Name:	Purchase_Order
Field Name: Supplier_ID	Data Type:	Lookup
API Name: Supplier_ID_c		
Description:		
Help Text:		
Data Owner:		
Field Usage:		
Data Sensitivity Level:		
Compliance Categorization:		
Created By: Prakash Meenashisundaram	Modified By:	Prakash Meenashisundaram

Lookup Options

Related To	Related List Label	Child Relationship Name	Object Name
Supplier	Purchase Orders	Purchase_Orders	Purchase_Order
	Required		
	What to do if the lookup record is deleted?	Dont allow deletion of the lookup record that's part of a lookup relationship.	
	Lookup Filter	No lookup filters defined.	

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.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Setup, Home, and Object Manager. The main title is "Purchase Order". On the left, a sidebar lists various configuration options under "Fields & Relationships". The main content area displays the "Purchase Order Custom Field" for the "Order Date" field. The "Custom Field Definition Detail" section shows the following details:

Field Label	Order Date	Object Name	Purchase_Order
Field Name	Order_Date	Data Type	Date
API Name	Order_Date__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Prakash Meenashisundaram	Modified By	Prakash Meenashisundaram
	9/7/2025, 12:20 AM		9/7/2025, 12:20 AM

Below this, the "General Options" section has "Required" checked. The "Validation Rules" section indicates "No validation rules defined." At the bottom, there are links for "Back To Top" and "Always show me ▾ more records per related list".

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.

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 for creating a custom field. The URL is orgfarm-aad03f4576-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000261JR/FieldsAndRelationships/00NgK00001fgW1n/view. The page title is "Purchase Order". The left sidebar shows "Fields & Relationships" is selected. The main content area displays the "Purchase Order Custom Field" named "Unit Price". The "Field Information" section includes the following details:

- Field Label: Unit Price
- Field Name: Unit_Price
- API Name: Unit_Price__c
- Description: Help Text
- Data Owner: Field Usage
- Data Sensitivity Level: Default
- Compliance Categorization: Not Specified
- Created By: Prakash Meenashisundaram, 9/8/2025, 2:25 AM
- Modified By: Prakash Meenashisundaram, 9/8/2025, 2:25 AM

The "General Options" section has "Required" checked. The "Currency Options" section shows Length: 10 and Decimal Places: 2. The "Validation Rules" section has a "New" button. A "Help for this Page" link is also present.

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 Setup interface for creating a custom field. The URL in the browser is <https://orgfarm-aad03f4576-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK0000026113/FieldsAndRelationships/00NgK00001fviaT/view>. The page title is "SETUP > OBJECT MANAGER Order Item". The left sidebar shows navigation options like Details, Fields & Relationships (which is 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, Restriction Rules, Scoping Rules, and Object Access. The main content area displays the "Custom Field Definition Detail" for the "Amount" field. The "Field Information" section includes the field label "Amount", field name "Amount", API name "Amount_c", and a formula "Quantity_Received__c * Unit_Price__c". The "Formula Options" section shows the data type as "Currency" and decimal places as "2". The "Object Name" is listed as "Order Item". The "Created By" is "Prakash Meenashisundaram" and the "Modified By" is also "Prakash Meenashisundaram". There are tabs for "Edit", "Set Field-Level Security", "View Field Accessibility", and "Where is this used?". A "Help for this Page" link is also present.

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.
7. Click on Next, Next and Save.

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Inventory Transaction Custom Field' and shows a field named 'Transaction Type'. The 'Field Information' section includes the field label 'Transaction Type', field name 'Transaction_Type', API name 'Transaction_Type__c', and data type 'Picklist'. The 'General Options' section has 'Required' checked and 'Default Value' set to null. The 'Picklist Options' section has 'Restrict picklist to the values defined in the value set' checked and 'Controlling Field' set to 'New'. The 'Validation Rules' and 'Help for this Page' buttons are also visible.

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.

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 includes links for Setup, Home, and Object Manager. The main title is "SETUP > OBJECT MANAGER" followed by "Supplier". On the left, a sidebar lists various configuration options under "Fields & Relationships". The main content area displays the "Supplier Custom Field" named "Phone Number". The "Custom Field Definition Detail" section shows the following details:

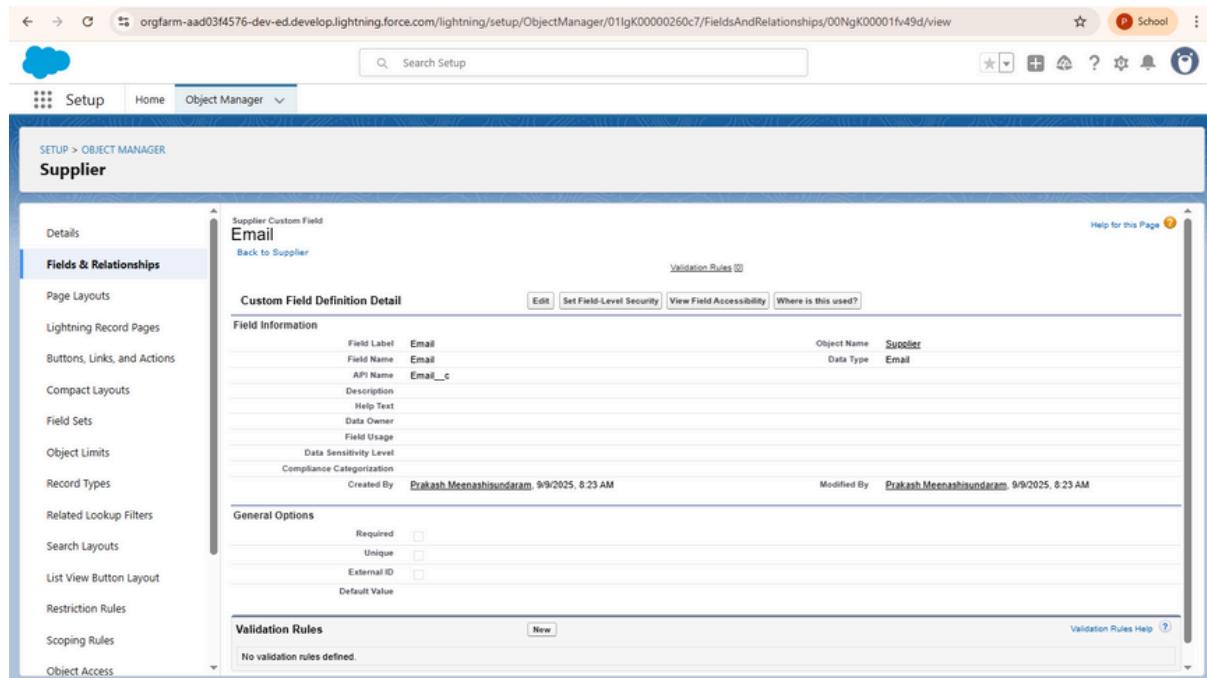
- Field Information:** Field Label: Phone Number, Field Name: Phone_Number, API Name: Phone_Number_c, Object Name: Supplier, Data Type: Phone.
- General Options:** Required is checked.
- Validation Rules:** A note states "No validation rules defined."

At the bottom of the page, there are links for "Back To Top" and "Always show me ▾ more records per related list".

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.



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.

The image contains two screenshots of the Salesforce Setup interface, illustrating the process of editing a Page Layout for the Product object.

Screenshot 1: Product Object Manager

This screenshot shows the "Object Manager" page for the "Product" object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, etc. The "Page Layouts" section is selected. The main area displays a table titled "Page Layouts" with one item: "Product Layout". The table includes columns for "PAGE LAYOUT NAME", "CREATED BY", and "MODIFIED BY". The "Product Layout" was created by "Aathi Hari Hara D" on 9/10/2025 at 1:49 AM, and last modified by "Aathi Hari Hara D" on 9/12/2025 at 10:23 AM.

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Product Layout	Aathi Hari Hara D, 9/10/2025, 1:49 AM	Aathi Hari Hara D, 9/12/2025, 10:23 AM

Screenshot 2: Product Page Layout Editor

This screenshot shows the "Page Layout" editor for the "Product Layout". The left sidebar is identical to the first screenshot. The main area has a toolbar with "Save", "Quick Save", "Preview As...", "Cancel", "Undo", "Redo", and "Layout Properties". The "Fields" tab is selected, showing a list of fields: "Section", "Blank Space", "Expected Delivery...", "Expiry Date", "Owner", "Product Description", "Transaction Type", "Created By", "Last Modified By", "Product ID", "Current Stock Level", "Minimum Stock Level", and "Product Name". Below the fields, the "Product Detail" section is visible, containing "Information" (Product ID, Product Name, Product Description, Expected Delivery Date), "System Information" (Created By, Last Modified By), and "Custom Links". A "Mobile Cards (Salesforce mobile only)" section is also present.

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

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

- Page: SETUP > OBJECT MANAGER > Purchase Order
- Section: Page Layouts
- Table Headers: PAGE LAYOUT NAME, CREATED BY, MODIFIED BY
- Table Data:

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Purchase Order Layout	Prakash Meenashisundaram, 9/6/2025, 11:38 PM	Prakash Meenashisundaram, 9/9/2025, 10:54 PM

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

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

- Page: SETUP > OBJECT MANAGER > Purchase Order
- Section: Page Layouts
- Panel: Layout Properties (Fields tab)
- Fields listed in the sidebar: Buttons, Quick Actions, Mobile & Lightning Actions, Expanded Lookups, Related Lists, Report Charts.
- Fields listed in the main area:

Field Name	Type	Label	Value
Section	Section	Last Modified By	Supplier ID
Blank Space	Text	Order Date	Total Order Cost
Actual Delivery Date	Date	Actual Delivery Date	Product
Created By	User	Created By	Purchase Order Name
- Buttons at the top: Save, Quick Save, Preview As..., Cancel, Undo, Redo.

To edit a Page Layout in Order Item Object

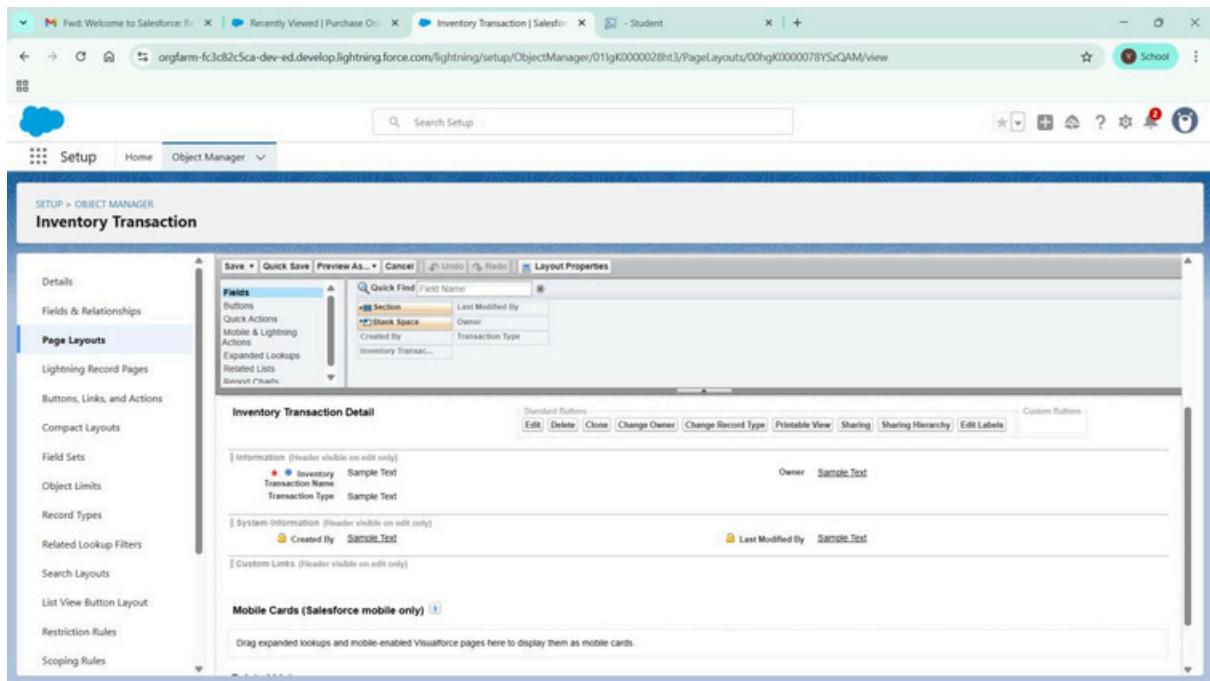
1. Go to setup >> click on Object Manager >> type object name(Order Item) in quick find box >> click on the Order Item object >> Page Layouts.
2. Click on the Order Item Layout
3. Drag and Arrange the field as shown below

The screenshot shows the Salesforce Setup interface for the Order Item object. The left sidebar has a tree view with 'Page Layouts' selected. The main content area displays the 'Order Item Detail' page layout. At the top, there's a 'Fields' section containing a 'Quick Find' bar and a list of fields: Product ID, Unit Price, Last Modified By, Purchase Order, Quantity Received, Order, Amounts, Order Item Name, and Total Order Cost. Below this is the 'Order Item Detail' section with fields for Order Item Name, Order, Amount, and Purchase Order. There are also sections for System Information (Created By, Last Modified By) and Custom Links. A note at the bottom says 'Drag expanded lookups and mobile-enabled Visualforce pages here to display them as mobile cards.'

4. Click Save.

To edit a Page Layout in Inventory Transaction Object

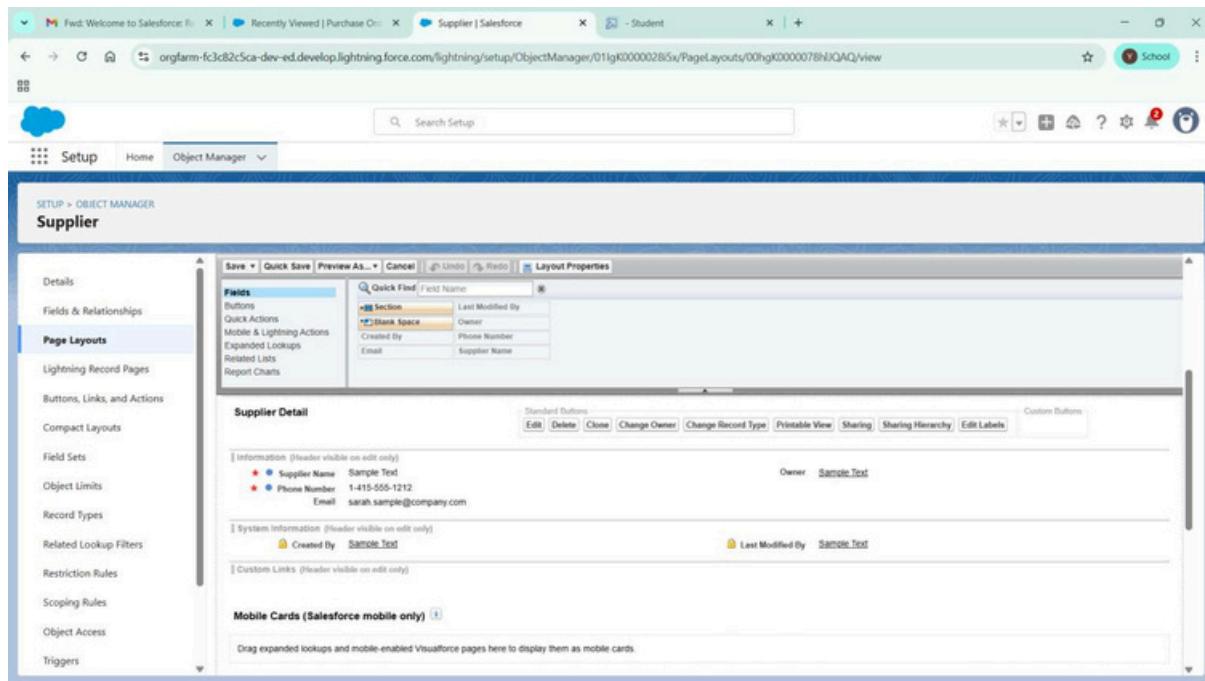
1. Go to setup >> click on Object Manager >> type object name(Inventory Transaction) in quick find box >> click on the Inventory Transaction object >> Page Layouts.
2. Click on the Inventory Transaction Layout
3. Drag and Arrange the field as shown below



4. Click Save.

To edit a Page Layout in Supplier Object

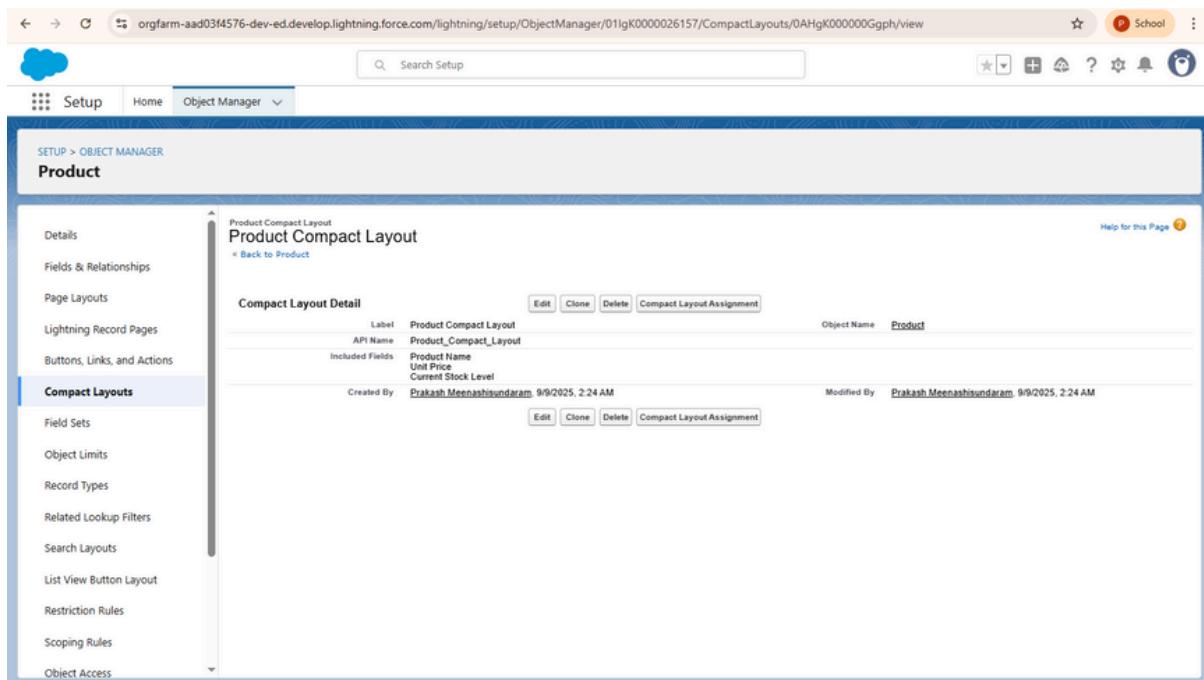
1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box >> click on the Supplier object >> Page Layouts.
2. Click on the Supplier Layout
3. Drag and Arrange the field as shown below



4. Click Save.

To create a Compact Layout to a Product Object

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object
2. Click on Compact Layouts in the sidebar .
3. Click on New.
4. Enter the Label as “Product Compact Layout”.
5. Select the Compact Layout Fields : Select Product name, Unit Price, Current Stock Level.
6. Click Save.
7. Click Compact Layout Assignment.
8. Click Edit Assignment.
9. Choose "Product Compact Layout" from the dropdown.
10. Click Save.



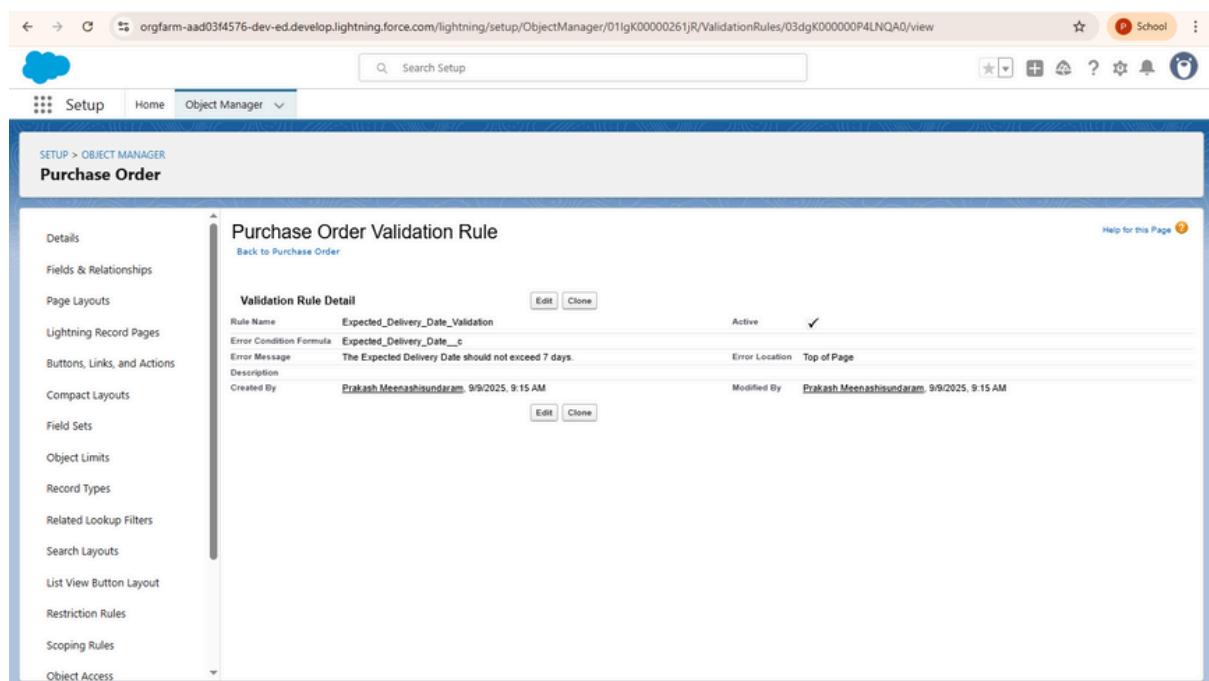
To create a Compact Layout to a Purchase Order Object

1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box >> click on the Purchase Order object
2. Click on Compact Layouts in the sidebar .
3. Click on New.
4. Enter the Label as “Purchase Order Compact Layout”.
5. Select the Compact Layout Fields : Select Purchase Order ID, Order Date, Total Order Cost, Supplier ID.
6. Click Save.
7. Click Compact Layout Assignment.
8. Click Edit Assignment.
9. Choose "Purchase Order Compact Layout" from the dropdown.
10. Click Save.

The screenshot shows the Salesforce Object Manager interface for creating a compact layout for the Product object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc., with 'Compact Layouts' selected. The main content area displays the 'Product Compact Layout' detail page. It shows the label 'Product Compact Layout', API name 'Product_Compact_Layout', and included fields 'Product Name', 'Unit Price', and 'Current Stock Level'. The 'Compact Layout Assignment' section indicates it's assigned to the 'Product' object. The page includes standard Salesforce navigation and action buttons.

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



The screenshot shows the Salesforce Object Manager interface for the Purchase Order object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area is titled "Purchase Order Validation Rule" and displays the "Validation Rule Detail" section. The rule is named "Expected_Delivery_Date_Validation". The "Error Condition Formula" is set to `Expected_Delivery_Date__c`. The "Error Message" is "The Expected Delivery Date should not exceed 7 days." The "Active" checkbox is checked. The "Error Location" is set to "Top of Page". The "Created By" field shows "Prakash Meenashisundaram" with a timestamp of "9/9/2025, 9:15 AM". The "Modified By" field also shows "Prakash Meenashisundaram" with the same timestamp. There are "Edit" and "Clone" buttons at the top right of the detail section.

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.

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.

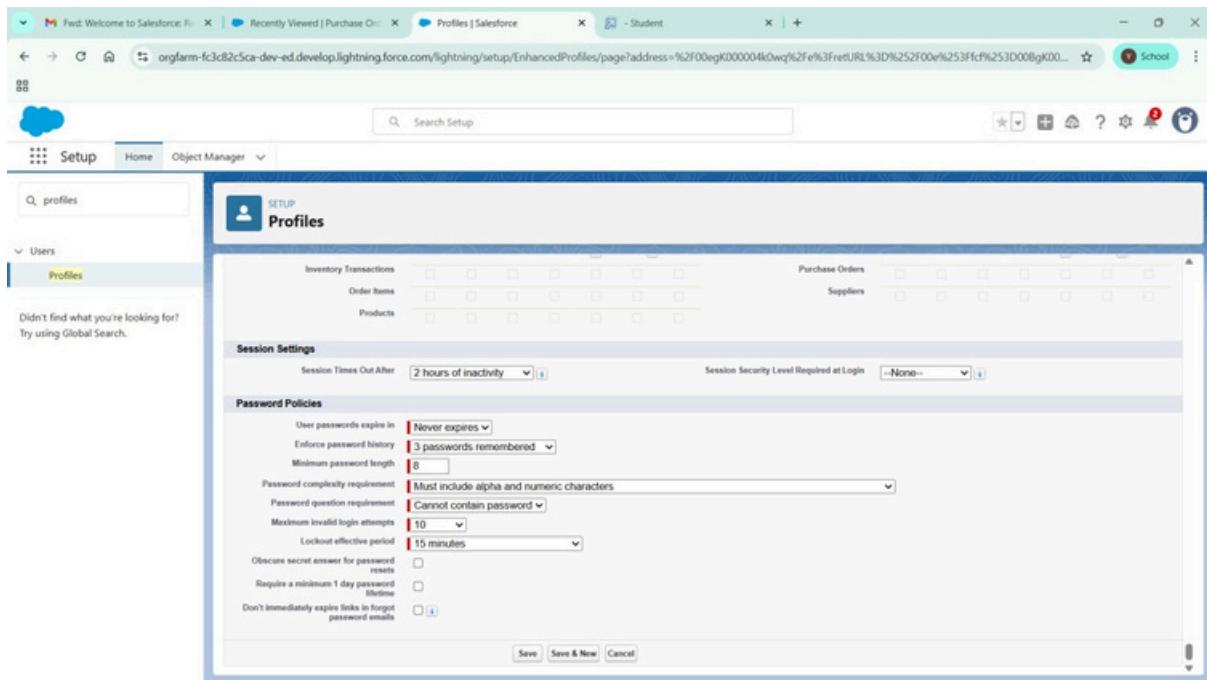
The screenshot shows the Salesforce Setup Profiles page. The URL is <https://orgfarm-fc3c82c5ca-dev-ed.lightning.force.com/lightning/setup/EnhancedProfiles/home>. The page title is "Profiles". A search bar at the top right contains "Search Setup". The left sidebar has "Setup" selected under "Users". The main content area shows a table titled "Profiles" with columns: Action, Profile Name, User License, and Custom. The "Custom" column has a checked checkbox next to the "Salesforce" profile. Other profiles listed include "Action", "Profile Name", "User License", and "Custom". The "Custom" column for most profiles has an unchecked checkbox.

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

The screenshot shows the "Custom App Settings" section of the Salesforce Setup Profiles page. The URL is <https://orgfarm-fc3c82c5ca-dev-ed.lightning.force.com/lightning/setup/EnhancedProfiles/page?address=%2F00egK000004k0wq%2Fe%3FretURL%3D%252F00e%253Fcf%253D008gK00...>. The page title is "Profiles". The left sidebar has "Setup" selected under "Users". The main content area shows a table titled "Custom App Settings" with columns: Visible and Default. The "Default" column has a checked radio button next to the "Medical Inventory Management" app. Other apps listed include "All Tabs (standard__AltTabSet)", "Analytics Studio (standard__Insights)", "App Launcher (standard__AppLauncher)", "Approvals (standard__Approvals)", "Automation (standard__FlowsApp)", "Bolt Solutions (standard__LightningDolt)", "Community (standard__Community)", "Content (standard__Content)", "Data Cloud (standard__Audience360)", "Data Manager (standard__DataManager)", "Digital Experiences (standard__ExperienceMS)", "Lightning Experience App (standard__LightningInstrumentation)", "Marketing CRM Classic (standard__Marketing)", "Medical Inventory Management (Medical_Inventory_Management)", and "Medical Inventory Management (Medical_Inventory_Managements)".

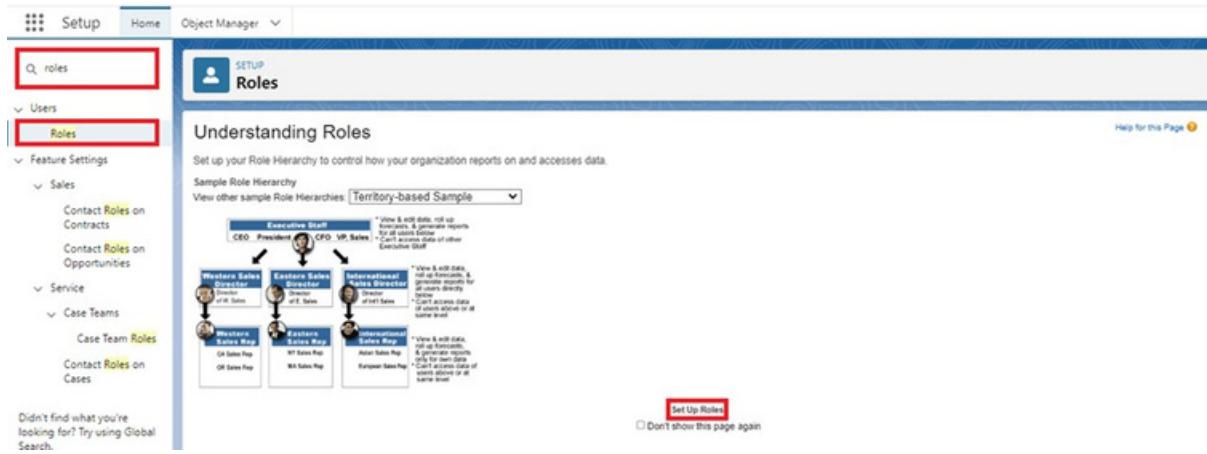
4. Scroll down to Custom Object Permissions and Give access permissions as mentioned in the below diagram.
5. Change the password policies as mentioned :
6. User passwords expire in should be “ never expires ”.

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



Create a Purchasing Manager Role.

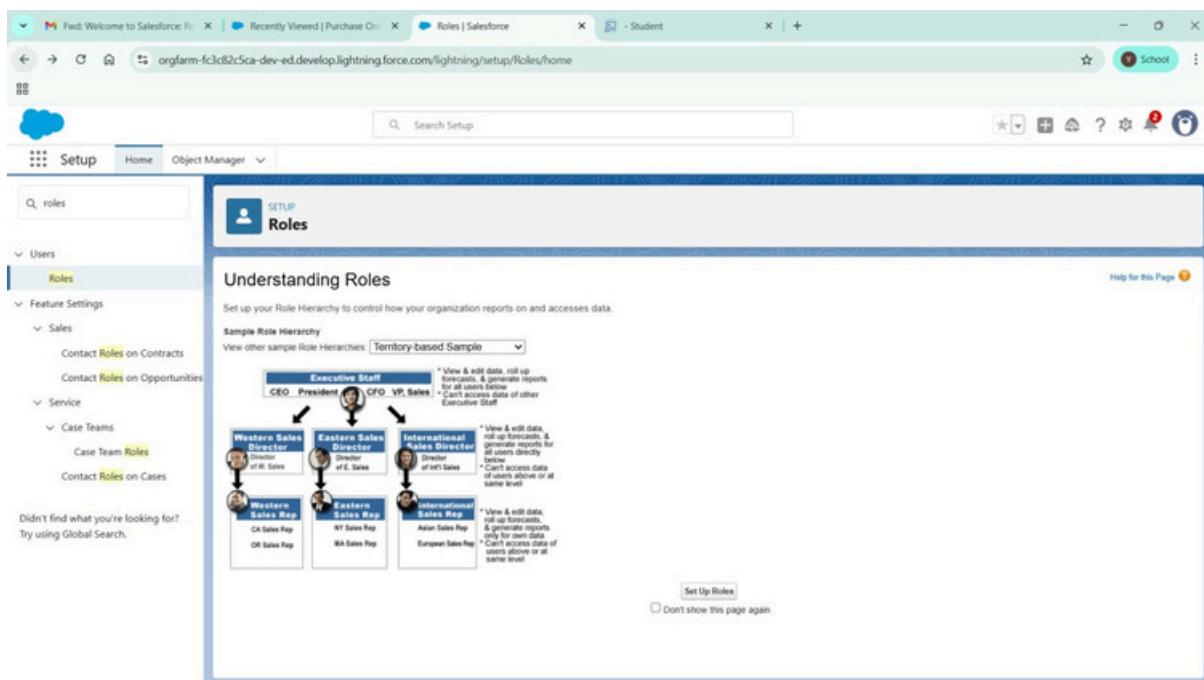
1. Go to quick find >> Search for Roles >> click on Set Up Roles.



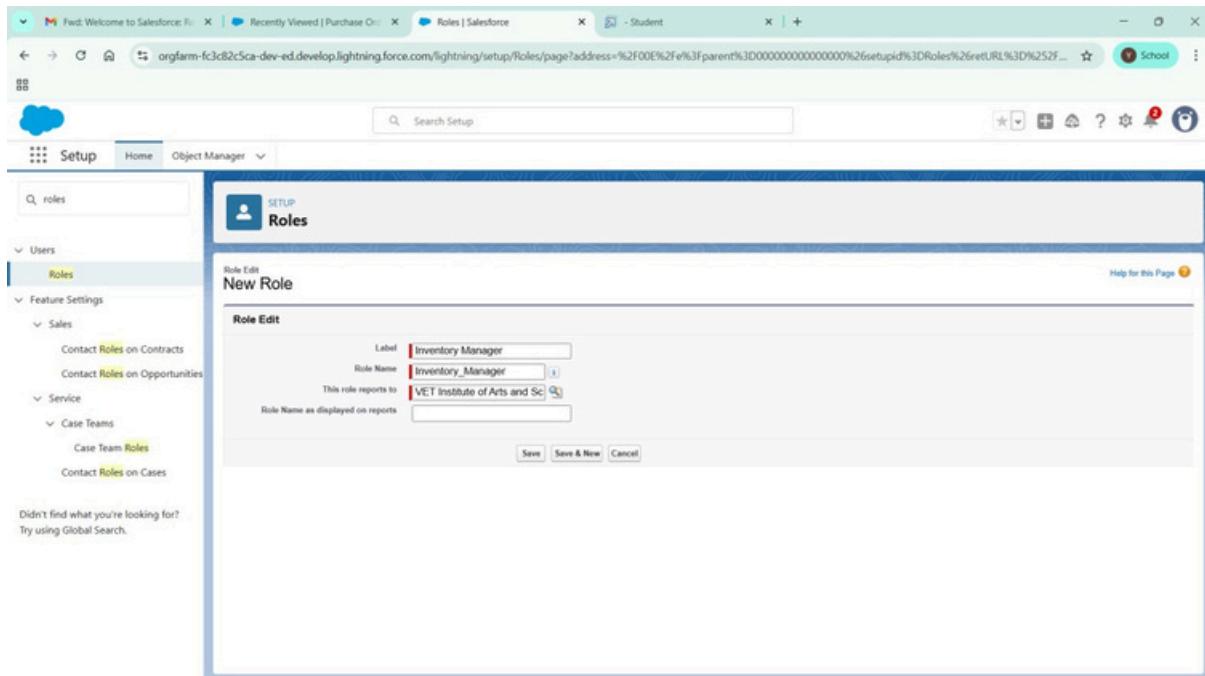
2. Click on Expand All and click on add role under SVP, Sales & Marketing role.
3. Give Label as “Purchasing Manager” and Role name gets auto populated. Then click onSave.

Create a Purchasing Manager Role.

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

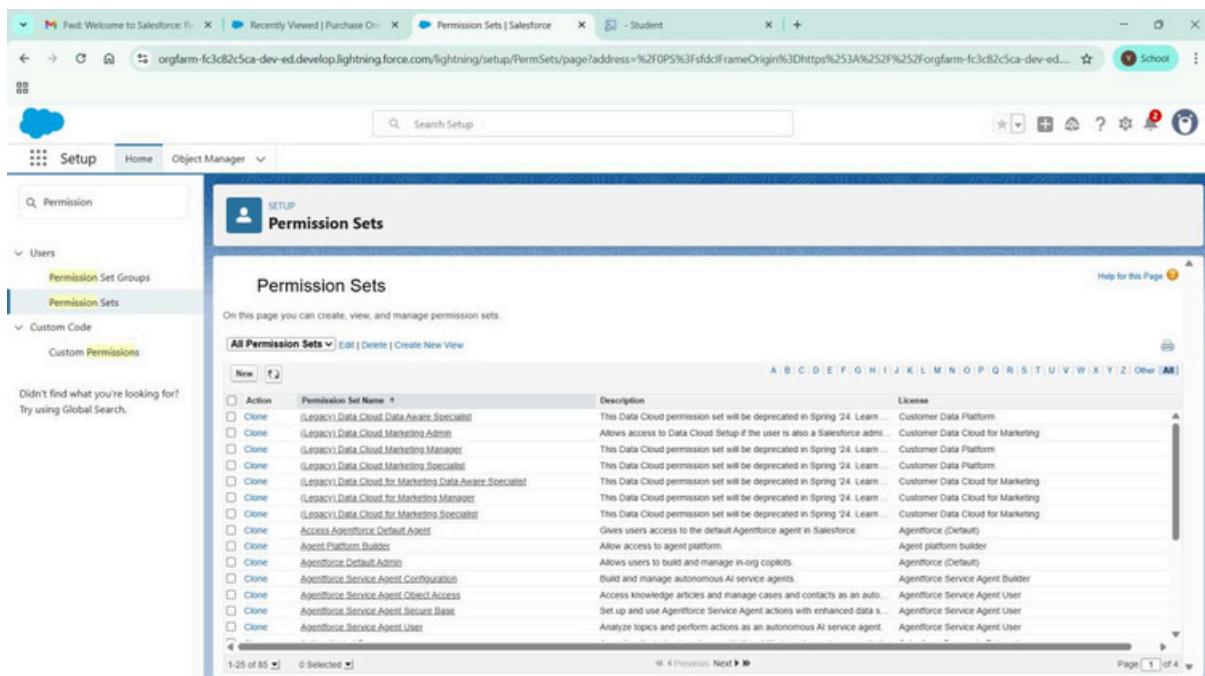


2. Click on Expand All and click on add role under SVP, Sales & Marketing role.
3. Give Label as “Inventory Manager” and the Role name gets auto populated. Then click on Save.

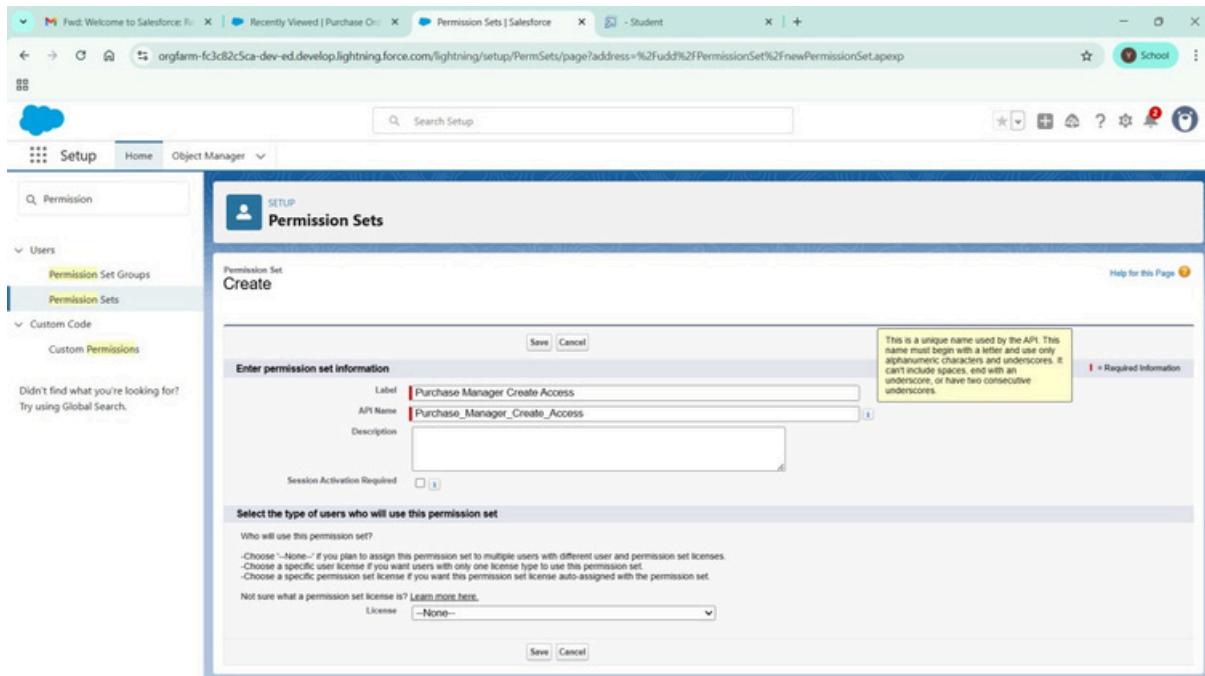


Create a Permission Set.

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 Purchase to assign the permission set to and click Next.

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

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.

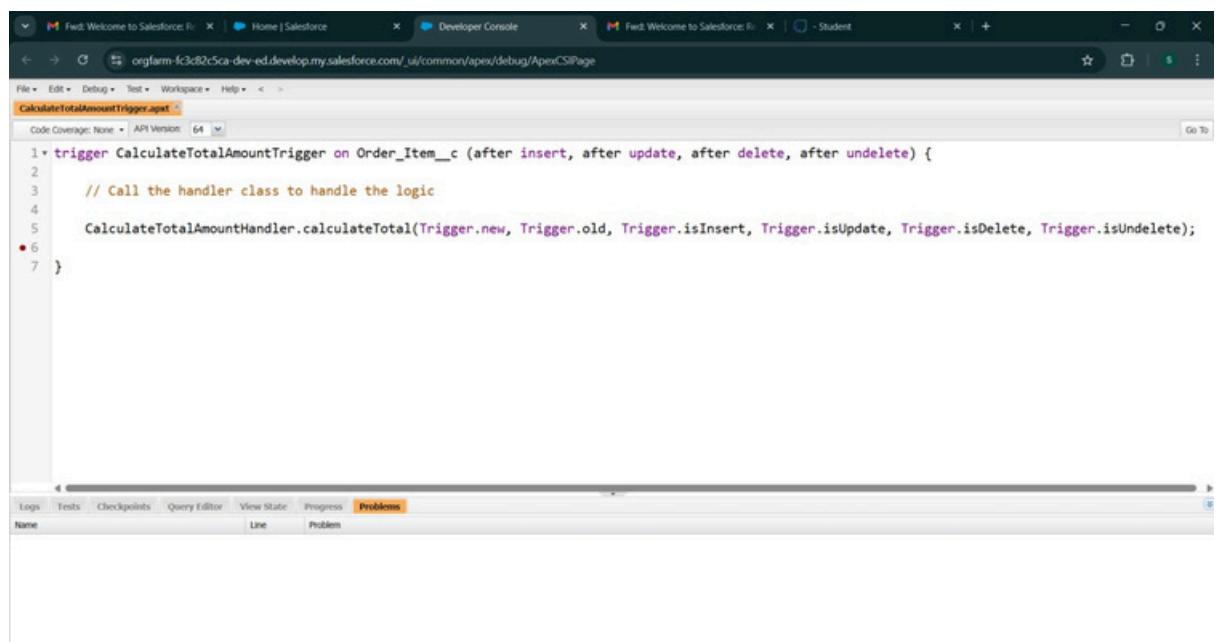
ii) Click on Developer Console: Click on the "Developer Console" option from the Setup menu. This will open the Developer Console in a new browser tab or window.

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.



The screenshot shows the Salesforce Developer Console interface. A tab titled "CalculateTotalAmountTrigger.apxt" is active, displaying the following Apex trigger code:

```
1 * trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {
2
3     // Call the handler class to handle the logic
4
5     CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);
6
7 }
```

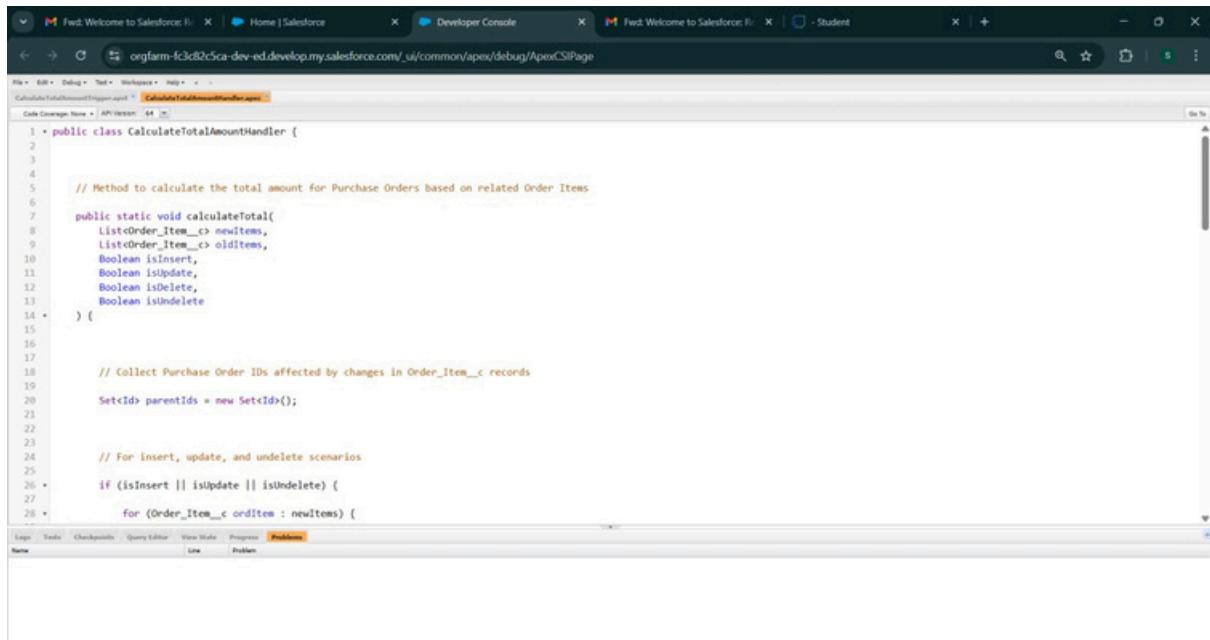
The code coverage is set to "None" and the API version is "64". Below the code editor, there is a "Logs" section and a "Problems" tab which is currently selected, showing no errors.

Step4:

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



The screenshot shows the Salesforce Developer Console with the URL <https://orgfarm-fc3cb2c5ca-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCsPage>. The tab title is "CalculateTotalAmountHandler.apex". The code editor displays the following Apex class:

```
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(  
7     List<Order_Item__c> newItems,  
8     List<Order_Item__c> oldItems,  
9     Boolean isInsert,  
10    Boolean isUpdate,  
11    Boolean isDelete,  
12    Boolean isUndelete  
13 ) {  
14  
15  
16  
17 // Collect Purchase Order IDs affected by changes in Order_Item__c records  
18 Set<Id> parentIds = new Set<Id>();  
19  
20  
21  
22  
23  
24 // For insert, update, and undelete scenarios  
25  
26 if (isInsert || isUpdate || isUndelete) {  
27  
28     for (Order_Item__c ordItem : newItems) {  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44 }
```

```
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__c);
```

```
}
```

```
}
```

```
//For update and delete scenarios
```

```
if(isUpdate || isDelete) {
```

```
    for(Order_Item__c ordItem : oldItems) {
```

```
        parentIds.add(ordItem.Purchase_Order__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__c, SUM(Amount__c)
        FROM Order_Item__c
        GROUP BY Purchase_Order__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;

    }

}

}

}
```

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.
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.

The screenshot shows the 'Medical Inventory ...' application's interface. At the top, there are navigation tabs: Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, Reports (which is currently selected), and Dashboards. Below the tabs is a search bar labeled 'Search...'. On the left, a sidebar titled 'Reports' lists categories: Recent, Created by Me, Private Reports, Public Reports, All Reports, Folders, All Folders, Created by Me, Shared with Me, and Favorites, All Favorites. The 'Recent' section contains one item: 'Purchase Orders based on Suppliers' under 'Private Reports', created by Prakash Meenashisundaram on 9/10/2025, 2:06 AM. The main area displays a table with columns: Report Name, Description, Folder, Created By, Created On, and Subscribed.

Report Name	Description	Folder	Created By	Created On	Subscribed
Purchase Orders based on Suppliers		Private Reports	Prakash Meenashisundaram	9/10/2025, 2:06 AM	

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

Purchase Orders based on Suppliers

	Order Count →	0	1	Total
Purchase Order Item: ID	Unit Price →	\$6.00	\$5.00	
<input type="checkbox"/> a05gK000007QD8r	Record Count	1	0	1
<input type="checkbox"/> a05gK000007QrmD	Record Count	0	1	1
Total	Record Count	1	1	2

Details (2 Rows) Click an intersection in the table above to filter details.

	Purchase Order Item: Purchase Order Item Name	Item Name	Quantity	Purchase Order Item: Owner Name	Purchase Order Item: Owner Alias	Purchase Order Item: Owner Role	Purchase Order Item: Created
1	Inventory Manager	paracetamol	100	Prakash Meenashisundaram	pra	-	Prakash Meenashisundaram
2	dolo	Dolo	2	Prakash Meenashisundaram	pra	-	Prakash Meenashisundaram
3							

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.

The screenshot shows the Salesforce App Launcher interface. On the left, there's a sidebar with categories: Dashboards, Recent (which has 1 item), DASHBOARDS, Recent, FOLDERS, and FAVORITES. The main area displays a table for the 'Recent' dashboard. The table columns are: Dashboard Name, Description, Folder, Created By, Created On, and Subscribed. One item is listed: 'Medical Inventory Dashboard' created by 'Aathi Hari Hara D' on '9/12/2025, 9:21 AM'. There are also search bars at the top and bottom of the main content area.

View Dashboard

1

- Click on App Launcher on the left side of the screen.
- 2 Search Medical Inventory Management & click on it.
- Click on Dashboard Tab.
- 3 Click on Medical Inventory DashBoard see graph view of records

4

The screenshot shows the 'Give name - Medical Inventory' dashboard. At the top, it says 'Give name - Medical Inventory' and 'Dashboard'. Below that, it says 'Give name - Medical Inventory DashBoard' and 'Last refreshed 14 days ago. Refresh this dashboard to see the latest data.' It also shows the date 'As of Sep 11, 2025, 1:29 AM' and the user 'Viewing as Prakash Meenashisundaram'. The main visual is a donut chart titled 'Purchase Orders based on Suppliers' with the value '2.00000'. To the right of the chart, it says 'Purchase Order Item: ID a05gK000007QD8r a05gK000007QrmD'. At the bottom, there are links for 'View Report (Purchase Orders based ...)' and 'As of Sep 11, 2025, 1:29 AM'.