

## **PROJECT TITLE: Medical Inventory Management System**

**College Name:** United College of Arts and Science

**College Code:** brubl

**TEAM ID:** NM2025TMID27775

### **TEAM MEMBERS:**

- **Team Leader Name:** Akash akashsumesh9489@gmail.com
- **Team Member 1:** AMAL XAVIER E xavieramal922@gmail.com
- **Team Member 2:** BRIGIN S J johnbrigin@gmail.com
- **Team Member 3:** SHRIJ C shrijsrij@gmail.com

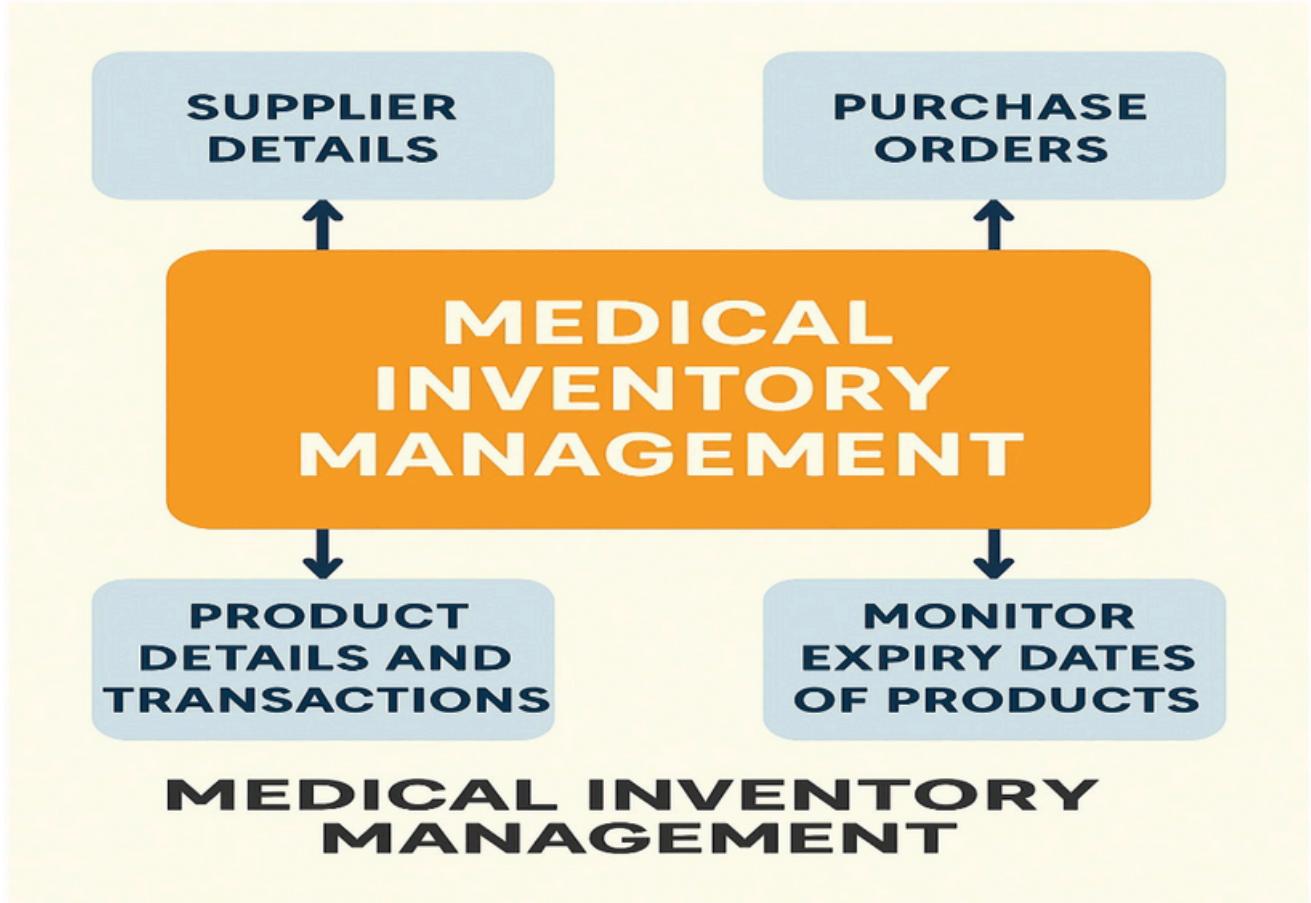
### **DEMO VIDEO LINK:**

 NM\_Demo video(Medical inventory management).mp4

## **INTRODUCTION**

### **Project Overview**

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



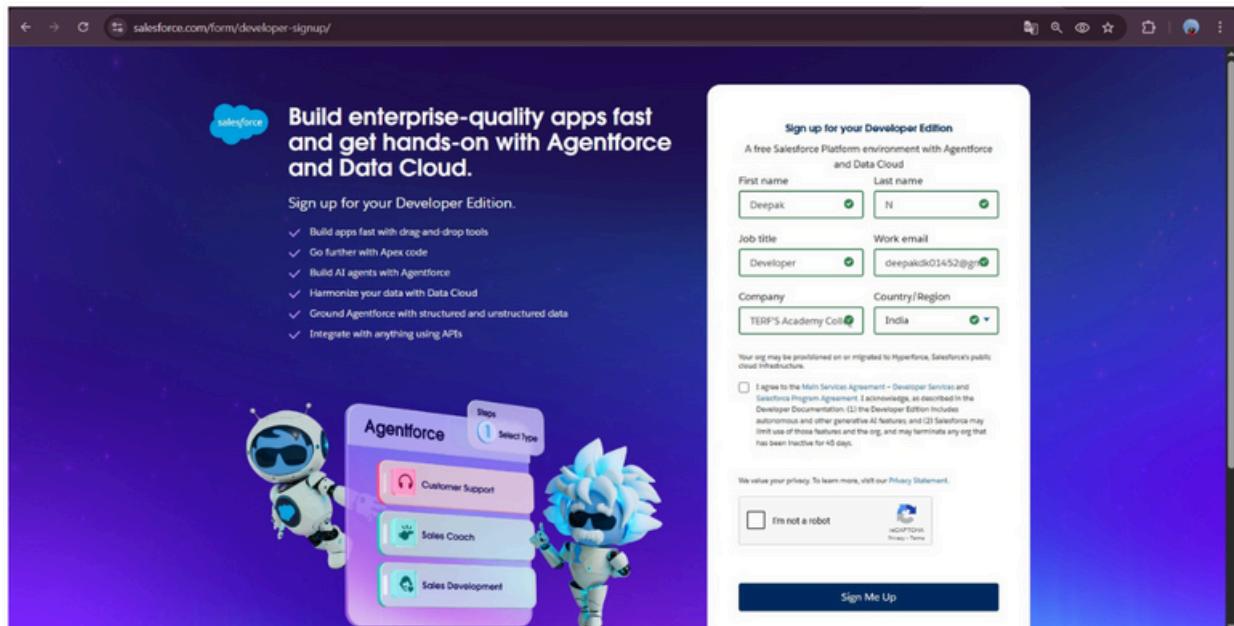
## Purpose

The system aims to efficiently maintain supplier details, manage purchase orders, track product details and transactions, and monitor the expiry dates of products. Maintain detailed records of suppliers, including contact information. Catalog product information, including descriptions, stock levels. Monitor and track product expiry dates to avoid using expired items. Comprehensive reports to track supplier performance, and purchase orders.

## DEVELOPMENT PHASE

### Creation of Developer Account

- A Salesforce Developer account was created using the signup link:  
<https://www.salesforce.com/form/developer-signup>



## Creating a Product Object

## Creating a tab for Product Object and Remaining Tabs

**Custom Tabs**

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Action	Label	Tab Style	Description
Edit   Del	Inventory Transactions	Hands	
Edit   Del	Order Items	Box	
Edit   Del	Products	Stethoscope	
Edit   Del	Purchase Orders	Shopping Cart	
Edit   Del	Suspects	Track	

## Create a Lightning App for Medical Inventory Management

**Lightning Experience App Manager**

27 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type, App Type

App Name	Developer Name	Description	Last Modified	App Type	Visibility
11 Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	8/31/2025, 4:27 PM	Lightning	✓
12 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience.	8/31/2025, 4:27 PM	Lightning	✓
13 Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	8/31/2025, 4:27 PM	Classic	✓
14 Medical Inventory Management	Medical_Inventory_Manageme...		9/5/2025, 5:13 AM	Lightning	✓
15 My Service Journey	MSIApp	Discover new customer service capabilities.	8/31/2025, 4:27 PM	Lightning	✓
16 Platform	Platform	The fundamental Lightning Platform	8/31/2025, 4:27 PM	Classic	✓
17 Queue Management	QueueManagement	Create and manage queues for your business.	8/31/2025, 4:27 PM	Lightning	✓
18 Sales	Sales	The world's most popular sales force automation (SFA).	8/31/2025, 4:27 PM	Classic	✓
19 Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more.	8/31/2025, 4:27 PM	Lightning	✓
20 Sales Cloud Mobile	SalesCloudMobile	New seller focused mobile first experience	8/31/2025, 4:27 PM	Lightning	✓
21 Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multi-channel data in one place.	8/31/2025, 4:27 PM	Lightning	✓
22 Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles, posts, and comments.	8/31/2025, 4:27 PM	Classic	✓

## Creating a Text Field in Product Object

The screenshot shows the Salesforce Object Manager interface for the 'Product' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main content area displays a table titled 'Fields & Relationships' with 10 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes fields such as Created By, Current Stock Level, Expiry Date, Last Modified By, Minimum Stock Level, Owner, Product Description, Product ID, Product Name, and Unit Price.

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedById	Lookup(User)		
Lightning Record Pages	Current Stock Level	Current_Stock_Level__c	Number(18, 0)		
Buttons, Links, and Actions	Expiry Date	Expiry_Date__c	Date		
Compact Layouts	Last Modified By	LastModifiedById	Lookup(User)		
Field Sets	Minimum Stock Level	Minimum_Stock_Level__c	Number(18, 0)		
Object Limits	Owner	OwnerId	Lookup(User,Group)		
Record Types	Product Description	Product_Description__c	Text Area(255)		
Related Lookup Filters	Product ID	Name	Text(80)		
Search Layouts	Product Name	Product_Name__c	Text(255)		
List View Button Layout	Unit Price	Unit_Price__c	Currency(16, 2)		

## Creating Lookup Relationship in Purchase Order Object

The screenshot shows the Salesforce Object Manager interface for the 'Purchase Order' object. The left sidebar lists various setup categories. The main content area displays a table titled 'Fields & Relationships' with 10 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes fields such as Actual Delivery Date, Created By, Expected Delivery Date, Last Modified By, Order Count, Order Date, Owner, Purchase Order ID, Supplier ID, and Total Order Cost.

Fields & Relationships					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Actual Delivery Date	Actual_Delivery_Date__c	Date		
Lightning Record Pages	Created By	CreatedById	Lookup(User)		
Buttons, Links, and Actions	Expected Delivery Date	Expected_Delivery_Date__c	Date		
Compact Layouts	Last Modified By	LastModifiedById	Lookup(User)		
Field Sets	Order Count	Order_Count__c	Roll-Up Summary (COUNT Order Item)		
Object Limits	Order Date	Order_Date__c	Date		
Record Types	Owner	OwnerId	Lookup(User,Group)		
Related Lookup Filters	Purchase Order ID	Name	Text(80)		
Search Layouts	Supplier ID	Supplier_ID__c	Lookup(Supplier)		
List View Button Layout	Total Order Cost	Total_Order_Cost__c	Currency(16, 2)		

## Creating a Unit Price Formula Field in Order Item object

The screenshot shows the Salesforce Object Manager interface for the 'Order Item' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main content area is titled 'Fields & Relationships' and displays 10 items, sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, and various configuration options. Key fields shown include Amount, Created By, Last Modified By, Order Item ID, Product ID, Purchase Order ID, Quantity Ordered, Quantity Received, Total Order Cost, and Unit Price.

FIELD LABEL	FIELD NAME	DATA TYPE
Amount	Amount__c	Formula (Currency)
Created By	CreatedBy	Lookup(User)
Last Modified By	LastModifiedBy	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)
Total Order Cost	Total_Order_Cost__c	Formula (Currency)
Unit Price	Unit_Price__c	Formula (Currency)

## Creating a Picklist Field in Inventory Transaction Object

The screenshot shows the Salesforce Object Manager interface for the 'Inventory Transaction' object. The left sidebar lists various setup categories. The main content area is titled 'Fields & Relationships' and displays 5 items, sorted by Field Label. The table includes columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status. Key fields shown include Created By, Inventory Transaction ID, Last Modified By, Owner, and Transaction Type. The 'Transaction Type' field is explicitly listed as a Picklist type.

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)		
Transaction Type	Transaction_Type__c	Picklist		

## To edit a Page Layout in Product Object

The screenshot shows the Salesforce Object Manager interface for the 'Product' object. The left sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts (which is selected), 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 area displays the 'Product Detail' page layout. The 'Fields' section includes fields like Section, Expiry Date, Last Modified By, Product ID, Mobile & Lightning Actions, Blank Space, Created By, Minimum Stock Level, Product Name, Current Stock Level, Owner, and Unit Price. Below this is the 'Actions' section with standard buttons: Edit, Delete, Clone, Change Owner, Change Record Type, Printable View, Sharing, Sharing Hierarchy, Edit Labels, and Custom Buttons. The 'Information' section contains sample data for Product ID, Product Name, Product Description, Unit Price, Current Stock Level, Minimum Stock Level, and Owner. The 'System Information' section includes Created By and Last Modified By. The 'Custom Links' section is empty. At the bottom is a 'Mobile Cards' section.

## To edit a Page Layout in Purchase Order Object

The screenshot shows the Salesforce Object Manager interface for the 'Purchase Order' object. The left sidebar lists the same configuration options as the Product layout. The main area displays the 'Purchase Order Detail' page layout. The 'Fields' section includes fields like Section, Expected Delivery..., Owner, Purchase Order ID, Blank Space, Last Modified By, Actual Delivery Date, Order Count, Supplier ID, Created By, Order Date, and Total Order Cost. The 'Actions' section is identical to the Product layout. The 'Information' section contains sample data for Purchase Order ID, Supplier ID, Order Date, Expected Delivery Date, Actual Delivery Date, Order Count, Total Order Cost, and Owner. The 'System Information' section includes Created By and Last Modified By. The 'Custom Links' section is empty. The 'Mobile Cards' section is also present at the bottom.

## To edit a Page Layout in Order Item Object

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-50f344d53-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK0000022jtu/PageLayouts/00hgK000006ugyinQAA/view>. The page title is "SETUP > OBJECT MANAGER Order Item". The left sidebar has "Page Layouts" selected. The main area shows the layout editor for the Order Item object. Fields include Order Item ID, Amount, Unit Price, Purchase Order ID, and Total Order Cost. A preview pane shows sample data for these fields.

## To edit a Page Layout in Inventory Transaction Object

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-50f344d53-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK0000022q2z/PageLayouts/00hgK000006ugyinQAA/view>. The page title is "SETUP > OBJECT MANAGER Inventory Transaction". The left sidebar has "Page Layouts" selected. The main area shows the layout editor for the Inventory Transaction object. Fields include Transaction Type, Owner, and Last Modified By. A preview pane shows sample data for these fields.

## To edit a Page Layout in Supplier Object

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, Lightning Record Pages, etc. The 'Page Layouts' section is selected. The main area displays the 'Supplier Detail' page layout configuration. It includes sections for 'Information', 'System Information', and 'Custom Links'. Fields visible in the layout include Section, Created By, Phone Number, Email, Supplier ID, Address, Last Modified By, Contact Person, and Owner. A 'Layout Properties' bar at the top provides standard save options.

## To create a Compact Layout to a Product Object

The screenshot shows the Salesforce Object Manager interface for the 'Product' object. The left sidebar lists various setup options, with 'Compact Layouts' selected. The main area shows the 'Edit Product Compact Layout' screen. It includes fields for 'Label' (set to 'Product Compact Layout') and 'Name' (set to 'Product\_Compact\_Layout'). Below this, the 'Select Compact Layout Fields' section allows users to choose fields from a list. Available fields include Created By, Expiry Date, Last Modified By, Minimum Stock Level, Owner, and Product ID. Selected fields include Unit Price, Product Name, Current Stock Level, and Top (selected by default). Buttons for 'Add', 'Remove', and 'Move' (Up, Down, Bottom) are available for managing the field order. A note at the bottom states: 'Use SHIFT + click to select adjacent fields. Use CTRL + click to select an assortment of fields.'

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

The screenshot shows the 'Purchase Order Compact Layout' configuration page in the Salesforce Setup. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The 'Compact Layouts' section is selected. The main area is titled 'Edit Purchase Order Compact Layout' and shows the 'Purchase Order Compact Layout'. It includes fields for Label ('Purchase Order Compact L') and Name ('Purchase\_Order\_Compact'). Below is a 'Select Compact Layout Fields' section with two columns: 'Available Fields' (Actual Delivery Date, Created By, Expected Delivery Date, Last Modified By, Order Count, Owner) and 'Selected Fields' (Purchase Order ID, Order Date, Total Order Cost, Supplier ID). A vertical toolbar on the right allows sorting fields by Top, Up, Down, and Bottom.

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

The screenshot shows the 'Purchase Order Validation Rule' configuration page in the Salesforce Setup. The left sidebar lists various setup categories. The 'Validation Rule Detail' section displays a rule named 'Expected\_Delivery\_Date\_Validation' with the formula '(Expected\_Delivery\_Date\_\_c - Order\_Date\_\_c) > 7'. The rule is active and located at the top of the page. It was created by 'Dessaa.N' on 5/5/2025, 11:25 PM and modified by 'Dessaa.N' on the same date and time.

## To create an Inventory Manager Profile

The screenshot shows the Salesforce Setup interface for creating a new profile. The left sidebar has 'Profiles' selected under 'Object Manager'. The main content area is titled 'Inventory Manager'. It displays the profile's details: Name (Inventory Manager), User License (Salesforce), Description (empty), Created By (Deepak.N), and Modified By (Deepak.N). The 'Profile Detail' section includes edit, close, delete, and view user buttons. Below this is the 'Page Layouts' section, which lists standard object layouts for various objects like Global, Email Application, Home Page Layout, Account, Alternative Payment Method, and Appointment Invitation. Each layout is associated with a specific assignment type (e.g., Global Layout [View Assignment], Not Assigned [View Assignment]). The right side shows location group assignments for objects like Location Group Assignment Layout, Macro Layout, Object Milestone Layout, Operating Hours, Opportunity, and Opportunity Product.

## To create an Purchase Manager Profile

The screenshot shows the Salesforce Setup interface for creating a new profile. The left sidebar has 'Profiles' selected under 'Object Manager'. The main content area is titled 'Purchase Manager'. It displays the profile's details: Name (Purchase Manager), User License (Salesforce), Description (empty), Created By (Deepak.N), and Modified By (Deepak.N). The 'Profile Detail' section includes edit, close, delete, and view user buttons. Below this is the 'Page Layouts' section, which lists standard object layouts for various objects like Global, Email Application, Home Page Layout, Account, Alternative Payment Method, and Appointment Invitation. Each layout is associated with a specific assignment type (e.g., Global Layout [View Assignment], Not Assigned [View Assignment]). The right side shows location group assignments for objects like Location Group Assignment Layout, Macro Layout, Object Milestone Layout, Operating Hours, Opportunity, and Opportunity Product.

## Create a Purchasing Manager Role.

The screenshot shows the Salesforce Setup Roles page. On the left, there's a sidebar with various setup options like Setup Home, Service Setup Assistant, and Administration. The main area is titled 'Role Purchasing Manager'. It displays the role detail for 'Purchasing Manager', which reports to 'SVP Sales & Marketing'. It was modified by 'Deepak.N' on 9/5/2025 at 11:50 PM. Under 'Opportunity Access', it says users can edit all opportunities associated with accounts they own. Under 'Case Access', it says users can edit all cases associated with accounts they own. The 'Role Name' is 'Purchasing\_Manager'. In the 'Users in Purchasing Manager Role' section, there is one user assigned: 'John.Purchaser' (Full Name: John.Purchaser, Alias: JohnP, Username: johnpurchaser@test.com). The status is 'Active'.

## Create Flow to update the Actual Delivery Date.

The screenshot shows the Salesforce Flow Builder interface for a flow named 'Actual Delivery Date Updating - V1'. The flow is triggered by a record being created or updated on a 'Purchase Order' object. It consists of the following steps:

- Start
- Get Purchase Record (Get Records)
- Assignment (Assignment)
- Updating Purchasing Order (Update Records)
- End

The flow is currently active and was last saved on 9/6/2025 at 08:59 PM.

**Create a Trigger to Calculate total amount on Order Item.**

Developer Console - Google Chrome

File • Edit • Debug • Test • Workspace • Help • < >

CalculateTotalAmountTrigger.apot CalculateTotalAmountHandler.apoc

Code Coverage: None API Version: 64 Go To

```
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 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------

**Choose Apex Class: Name it as CalculateTotalAmountHandler**

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

The screenshot shows a Salesforce Lightning report titled "Purchase Orders based on Suppliers". The report summary at the top indicates 5 total records, 14 total order count, and a total order cost of \$26,325.00. The main table lists purchase orders grouped by supplier. Supplier 001 has 4 entries: Purchase-0001 (1), Purchase-0002 (1), Purchase-0003 (1), and Purchase-0004 (1). Supplier 002 has 1 entry: Purchase-0005 (1). The table includes columns for Supplier ID, Purchase Order ID, Order Count, and Total Order Cost.

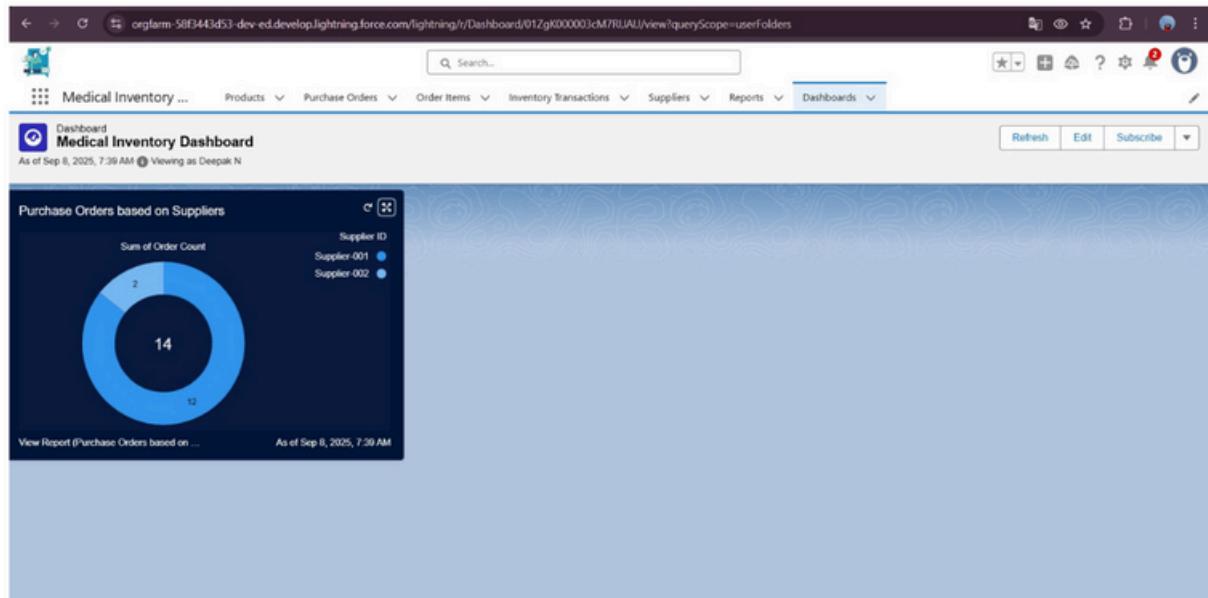
Supplier ID	Purchase Order ID	Order Count	Total Order Cost
Supplier-001 (4)	Purchase-0001 (1)	3	\$2,075.00
	Purchase-0002 (1)	2	\$3,250.00
	Purchase-0003 (1)	3	\$7,000.00
	Purchase-0004 (1)	4	\$9,500.00
Supplier-002 (1)	Purchase-0005 (1)	2	\$4,500.00
<b>Total (5)</b>		<b>14</b>	<b>\$26,325.00</b>

## Create a Complete Purchase Details Report

The screenshot shows a Salesforce Lightning report titled "Complete Purchase Details Report". The report summary at the top indicates 14 total records, 276 total quantity received, and a total amount of \$26,325.00. The main table lists purchase items grouped by supplier and delivery date. Supplier 001 has 12 entries across two delivery dates: 9/10/2025 and 9/11/2025. Supplier 002 has 2 entries. Supplier 003 has 3 entries. The table includes columns for Supplier ID, Actual Delivery Date, Purchase Order ID, Product ID, Order Count, Product Name, Quantity Received, and Amount.

Supplier ID	Actual Delivery Date	Purchase Order ID	Product ID	Order Count	Product Name	Quantity Received	Amount
Supplier-001 (12)	9/10/2025 (4)	Purchase-0004 (4)	Antibiotic Tablets	4	Antibiotic Tablets	40	\$6,000.00
			Paracetamol	4	Paracetamol 500mg	20	\$500.00
			Vitamin C Tablets	4	Vitamin C Tablets	10	\$500.00
			Antacid	4	Antacid	25	\$2,500.00
				4		95	\$9,500.00
				4		95	\$9,500.00
			Subtotal				
	9/11/2025 (8)	Purchase-0001 (3)	Antibiotic Tablets	3	Antibiotic Tablets	10	\$1,500.00
			Paracetamol	3	Paracetamol 500mg	7	\$175.00
			Antacid	3	Antacid	4	\$400.00
			Subtotal	3		21	\$2,075.00
			Purchase-0002 (2)	2	Antibiotic Tablets	15	\$2,250.00
				2	Antacid	10	\$1,000.00
			Subtotal	2		25	\$3,250.00
			Purchase-0003 (3)	3	Antibiotic Tablets	24	\$3,600.00
				3	Paracetamol	56	\$1,400.00

## **View Dashboard**



## **RESULTS**

- Tabs for Product, Supplier, Purchase Order, Inventory.
- Reports for Expired Products and Supplier Performance.
- Dashboard showing Stock Levels and Purchase Order Summary.
- Trigger execution results (auto-calculated total order amount).
- Validation Rule error messages (when wrong data is entered).

## **ADVANTAGES & DISADVANTAGES**

### **Advantages**

- Accurate tracking of products and expiry dates.
- Easy management of supplier and purchase orders.
- Reduced manual work with automation (flows and triggers).
- Visual dashboards for quick decision-making.

## Disadvantages

- Requires Salesforce knowledge for customization.
- Limited offline functionality.
- Integration with external systems (e.g., hospital management software) not implemented yet.

## APPENDIX

### Create an Apex Trigger:

```
trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {
```

```
    // Call the handler class to handle the logic
    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert,
    Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);
}
```

### Create Apex Class:

```
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;
}
}
}
}

```

### **Future Enhancements**

- Add **barcode scanning** for products to make stock entry faster.
  - Implement **email or SMS alerts** for products nearing expiry.
  - Create **mobile-friendly pages** for quick access by staff.
  - Add **AI predictions** for stock demand and reordering.
  - Integrate with **external hospital systems** for real-time updates.

## **CONCLUSION**

The Medical Inventory Management System successfully streamlines the operations of managing medical supplies using Salesforce. It ensures better accuracy, reduces errors, and improves efficiency in handling suppliers, purchase orders, and products. With features like validation rules, flows, triggers, reports, and dashboards, the project demonstrates the practical use of Salesforce in real-time business scenarios.