

PROJECT TITLE: Medical Inventory Management System

College Name: TERF's Academy College of Arts and Science

College Code: bru5e

TEAM ID: NM2025TMID27527

TEAM MEMBERS:

- **Team Leader Name:** DEEPAK N – deepakdeepak01452@gmail.com
- **Team Member 1:** ABDUL KALAM AZATH A – akazath277@gmail.com
- **Team Member 2:** VISHNU C R – smilycrd@gmail.com
- **Team Member 3:** DEVADHARSHINI M – ddharshini820@gmail.com

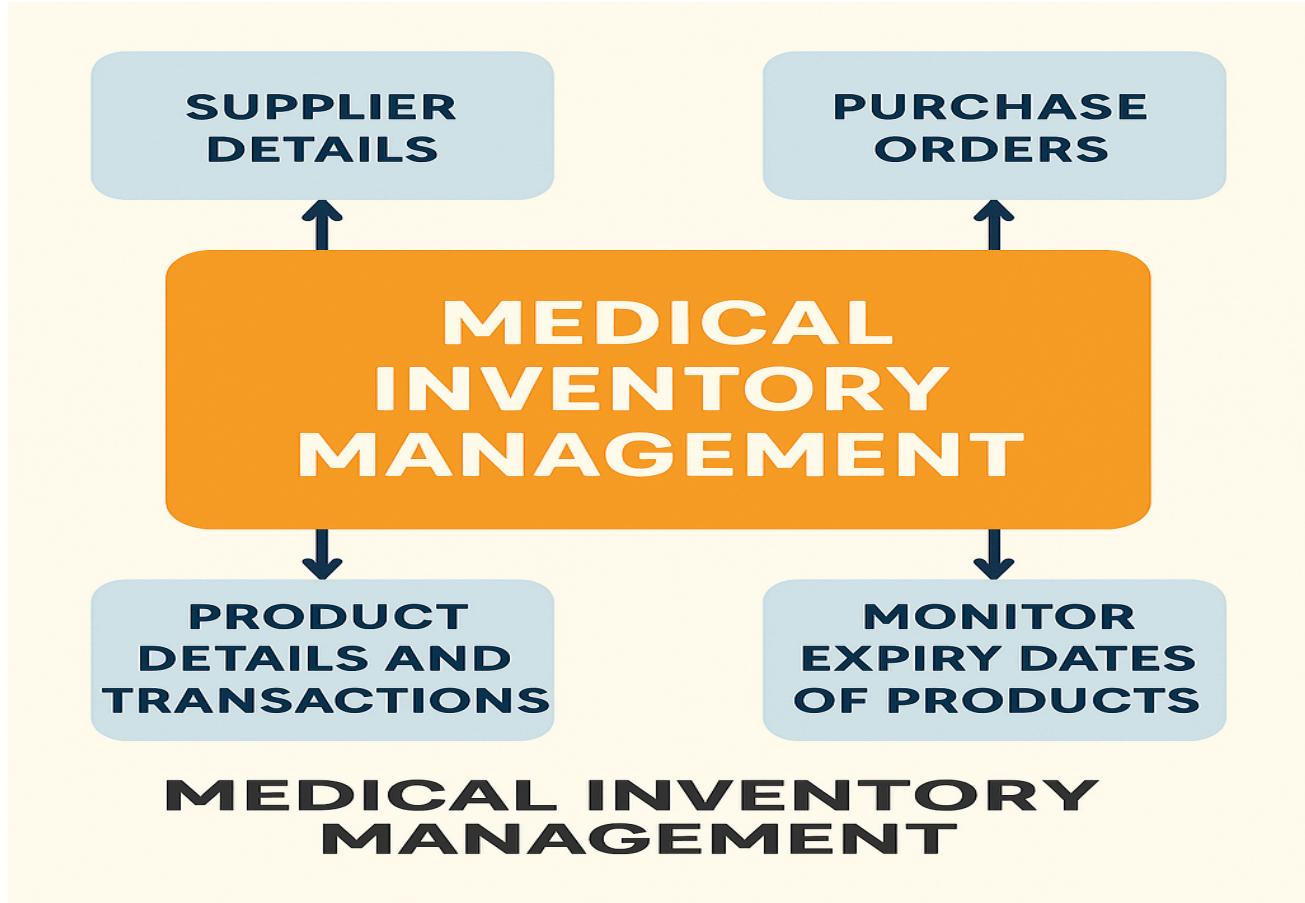
DEMO VIDEO LINK:

 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.



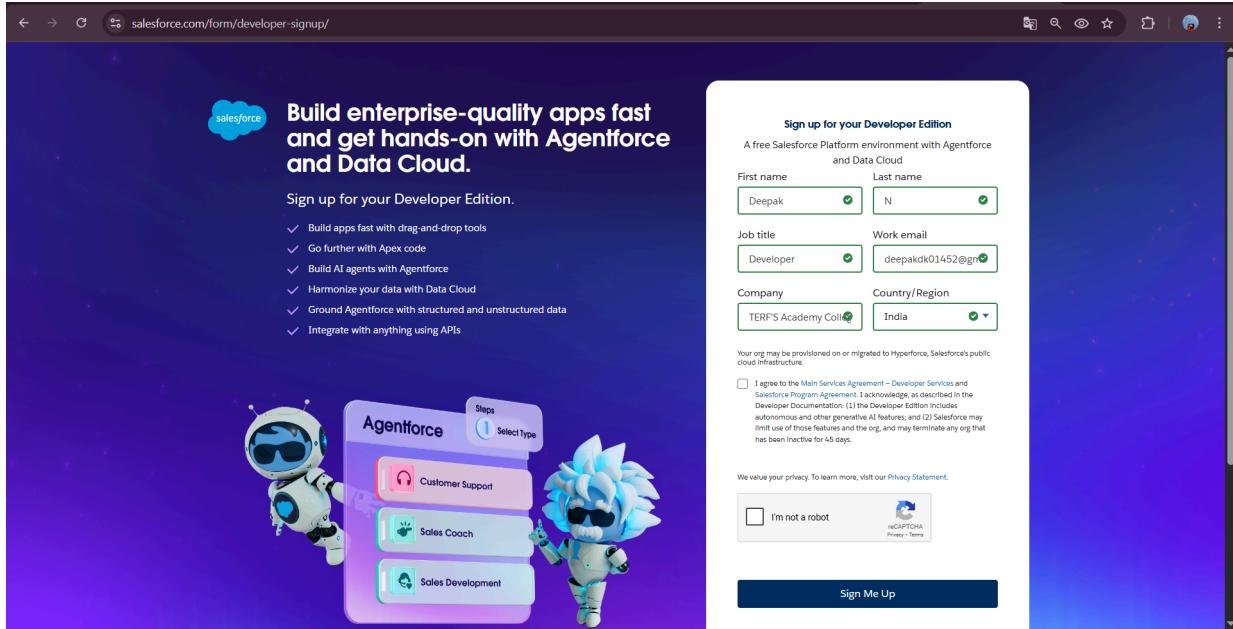
Purpose

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	Suppliers	Truck	

Web Tabs

No Web Tabs have been defined.

Visualforce Tabs

No Visualforce Tabs have been defined.

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...	Ap...	Vi...
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 Experi...	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	MSJApp	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, oppo...	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...	8/31/2025, 4:27 PM	Lightning	✓
22 Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profil...	8/31/2025, 4:27 PM	Classic	✓

Creating a Text Field in Product Object

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)		
Restriction Rules					
Scoping Rules					

Creating Lookup Relationship in Purchase Order Object

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)		
Restriction Rules					
Scoping Rules					

Creating a Unit Price Formula Field in Order Item object

Object Manager

Order Item

Fields & Relationships

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

Object Manager

Inventory Transaction

Fields & Relationships

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 Setup interface for the Product object. The left sidebar lists various configuration options like 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 Page Layout Properties for the Product Detail layout. It includes a Fields section with fields like Expiry Date, Section, Blank Space, Last Modified By, Created By, Minimum Stock Level, Current Stock Level, Product ID, Product Name, and Owner. Below this is a Product Detail section with a table showing information such as Product ID, Product Name, Product Description, Unit Price (\$123.45), Current Stock Level (81,681), Minimum Stock Level (62,537), and Owner (Sample Text). There are also sections for System Information (Created By, Last Modified By) and Custom Links.

To edit a Page Layout in Purchase Order Object

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 (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 Page Layout Properties for the Purchase Order Detail layout. It includes a Fields section with fields like Expected Delivery..., Section, Blank Space, Last Modified By, Purchase Order ID, Actual Delivery Date, Order Count, Supplier ID, Created By, Order Date, and Total Order Cost. Below this is a Purchase Order Detail section with a table showing information such as Purchase Order ID, Supplier ID, Order Date (9/8/2025), Expected Delivery Date (9/8/2025), Actual Delivery Date (9/8/2025), Order Count (76,264), Total Order Cost (\$123.45), and Owner (Sample Text). There are also sections for System Information (Created By, Last Modified By) and Custom Links.

To edit a Page Layout in Order Item Object

SETUP > OBJECT MANAGER
Order Item

Save Quick Save Preview As... Cancel Undo Redo Layout Properties

Fields

Section	Last Modified By	Quantity Ordered
Blank Space	Order Item ID	Quantity Received
Amount	Product ID	Total Order Cost
Created By	Purchase Order ID	Unit Price

Information (Header visible on edit only)

Order Item ID Sample Text
Amount \$123.45
Product ID Sample Text
Unit Price \$123.45

Purchase Order ID Sample Text
Quantity Ordered 24,404
Quantity Received 68,497

System Information (Header visible on edit only)

Created By Sample Text
Last Modified By Sample Text

Custom Links (Header visible on edit only)

Mobile Cards (Salesforce mobile only) *i*

Drag expanded lookups and mobile-enabled Visualforce pages here to display them as mobile cards.

Related Lists

To edit a Page Layout in Inventory Transaction Object

SETUP > OBJECT MANAGER
Inventory Transaction

Save Quick Save Preview As... Cancel Undo Redo Layout Properties

Fields

Section	Last Modified By	Owner
Blank Space	Created By	Owner
Transaction ID	Owner	Sample Text
Transaction Type	Sample Text	

Information (Header visible on edit only)

Inventory Transaction ID Sample Text
Transaction Type Sample Text
Owner Sample Text

System Information (Header visible on edit only)

Created By Sample Text
Last Modified By Sample Text

Custom Links (Header visible on edit only)

Mobile Cards (Salesforce mobile only) *i*

Drag expanded lookups and mobile-enabled Visualforce pages here to display them as mobile cards.

Related Lists

To edit a Page Layout in Supplier Object

To create a Compact Layout to a Product Object

To create a Compact Layout to a Purchase Order Object

orgfarm-58f3443d53-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK000002plF/CompactLayouts/0AHgK00000GxuT/edit

The screenshot shows the 'Purchase Order Compact Layout' configuration page. On the left, a 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 contains a sub-section 'Purchase Order Compact Layout'. It includes fields for 'Label' (set to 'Purchase Order Compact L') and 'Name' (set to 'Purchase_Order_Compact'). Below this is a 'Select Compact Layout Fields' section with two columns: 'Available Fields' (listing fields like Actual Delivery Date, Created By, Expected Delivery Date, Last Modified By, Order Count, Owner) and 'Selected Fields' (listing fields like Purchase Order ID, Order Date, Total Order Cost, Supplier ID). A note at the bottom says 'Use SHIFT + click to select adjacent fields. Use CTRL + click to select an assortment of fields.'

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

orgfarm-58f3443d53-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK000002plF/ValidationRules/03dgK00000OdHxQAK/view

The screenshot shows the 'Purchase Order Validation Rule' configuration page. The sidebar on the left is identical to the previous screenshot. The main area is titled 'Purchase Order Validation Rule' and contains a 'Validation Rule Detail' section. The rule details are as follows:

Rule Name	Expected_Delivery_Date_Validation	Active
Error Condition Formula	(Expected_Delivery_Date__c - Order_Date__c) > 7	
Error Message	The Expected Delivery Date should not exceed 7 days.	Error Location
Description		Top of Page
Created By	Deepak.N. 9/5/2025, 11:25 PM	Modified By
		Deepak.N. 9/5/2025, 11:25 PM

To create an Inventory Manager Profile

The screenshot shows the Salesforce Setup interface under the Profiles section. The profile name is 'Inventory Manager'. The 'Profile Detail' section includes fields for Name (Inventory Manager), User License (Salesforce), and Description. The 'Custom Profile' checkbox is checked. The 'Page Layouts' section shows assignments for various object layouts, such as Global Layout for Home Page Layout and Object Milestone Layout for Account.

To create an Purchase Manager Profile

The screenshot shows the Salesforce Setup interface under the Profiles section. The profile name is 'Purchase Manager'. The 'Profile Detail' section includes fields for Name (Purchase Manager), User License (Salesforce), and Description. The 'Custom Profile' checkbox is checked. The 'Page Layouts' section shows assignments for various object layouts, such as Global Layout for Home Page Layout and Object Milestone Layout for Account.

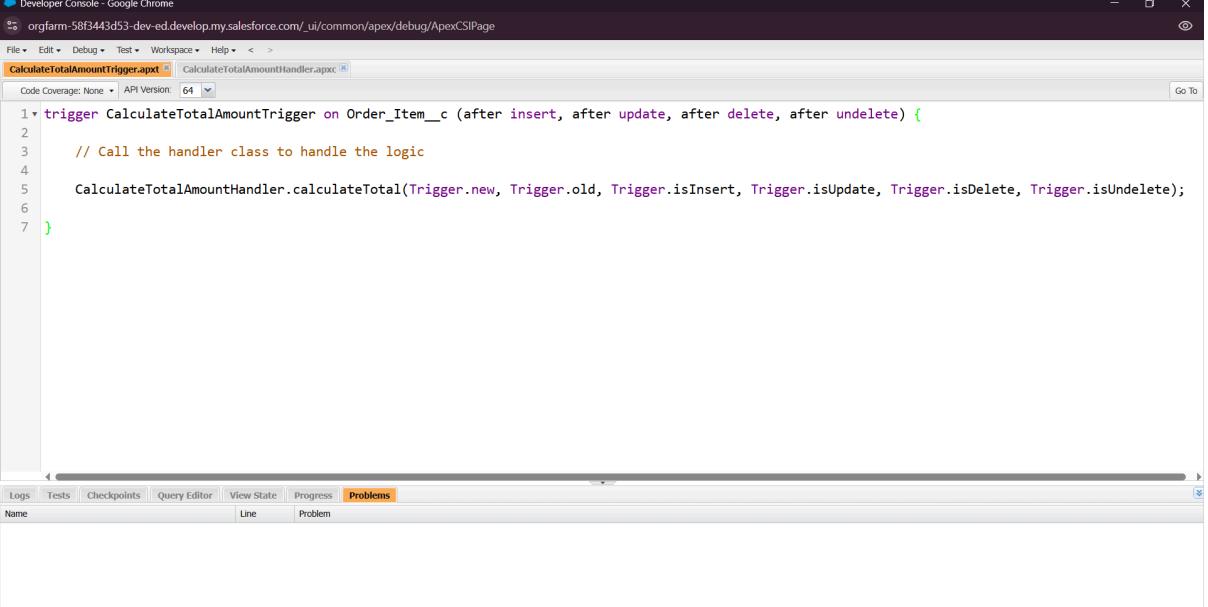
Create a Purchasing Manager Role.

The screenshot shows the Salesforce Setup Roles page. On the left, there's a sidebar with various setup links like Setup Home, Salesforce Go, Service Setup Assistant, etc. The main area is titled 'Roles' and shows a 'Purchasing Manager' role. The 'Role Detail' section includes fields for Label (Purchasing Manager), This role reports to (SVP, Sales & Marketing), Modified By (Deepak N), Opportunity Access (Users can edit all opportunities), Case Access (Users can edit all cases), Role Name (Purchasing_Manager), and Role Name as displayed on reports (SVP, Sales & Marketing). Below this is a table titled 'Users in Purchasing Manager Role' with one row for 'John.PurchaseM'. A 'Help for this Page' link is at the top right.

Create Flow to update the Actual Delivery Date.

The screenshot shows the Salesforce Flow Builder interface. The flow is titled 'Actual Delivery Date Updating - V1' and is a 'Record-Triggered Flow' for 'Purchase Order'. It starts with a 'Get Purchase Record' action, followed by an 'Assignment' action, then an 'Updating Purchasing Order' action, and ends with an 'End' action. The flow was last saved on 9/6/2025, 08:59 PM. The toolbar at the top includes buttons for Select Elements, Auto-Layout, and various flow controls.

Create a Trigger to Calculate total amount on Order Item.

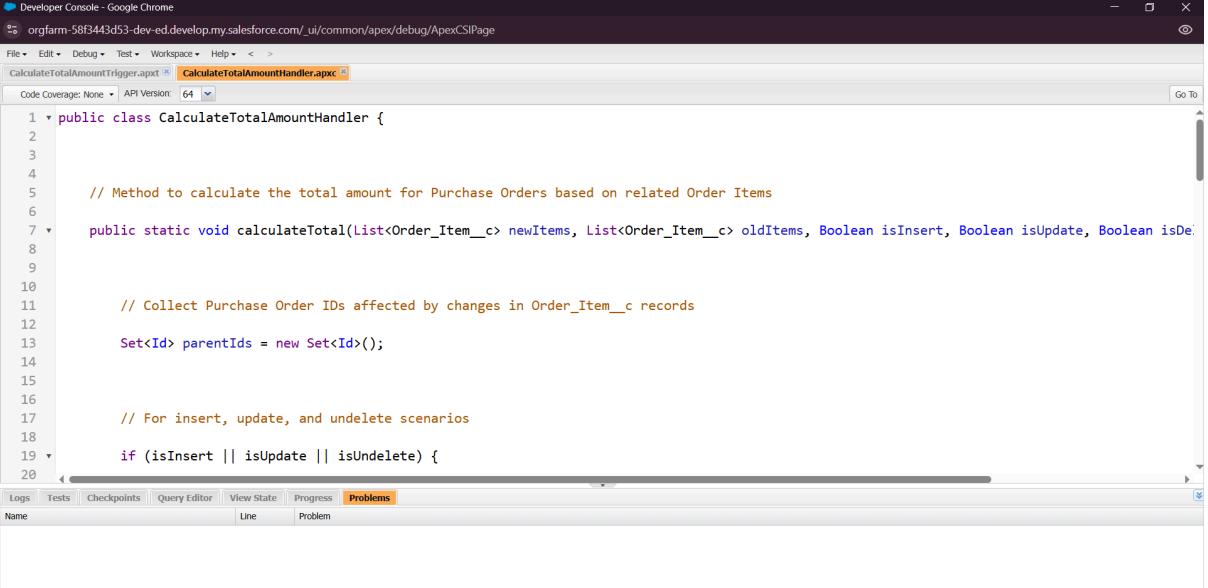


The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-58f3443d53-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is CalculateTotalAmountTrigger.apxc. The code editor contains the following Apex trigger:

```
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);
}
```

The console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing no errors or warnings.

Choose Apex Class: Name it as CalculateTotalAmountHandler



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-58f3443d53-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab title is CalculateTotalAmountHandler.apxc. The code editor contains the following 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) {
        // 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 || isDelete) {
    }
```

The console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing no errors or warnings.

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 total order cost of \$26,325.00. The main table lists purchase orders grouped by supplier. Supplier-001 has 4 entries: Purchase-0001 (\$2,075.00), Purchase-0002 (\$3,250.00), Purchase-0003 (\$7,000.00), and Purchase-0004 (\$9,500.00). Supplier-002 has 1 entry: Purchase-0005 (\$4,500.00). A "Total (5)" row shows the grand total of \$26,325.00. The report interface includes standard Salesforce navigation and reporting tools like "Enable Field Editing", "Add Chart", and "Edit".

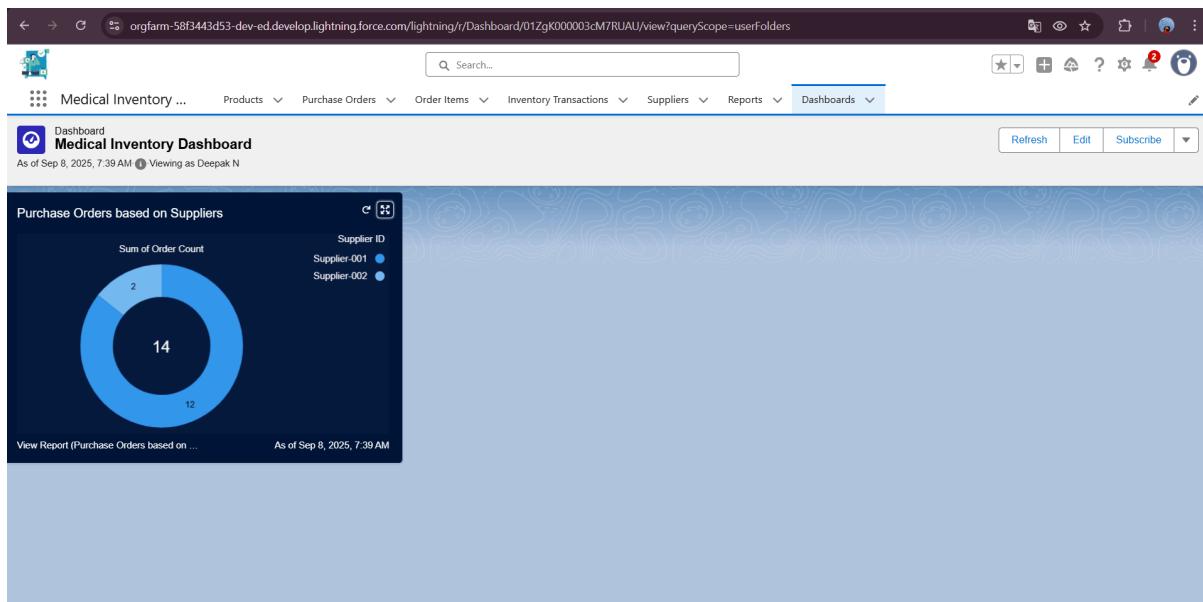
Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
Supplier-001 (4)	Purchase-0001 (1)	3	\$2,075.00
	Purchase-0002 (1)	2	\$3,250.00
	Purchase-0003 (1)	3	\$7,000.00
	Purchase-0004 (1)	4	\$9,500.00
Supplier-002 (1)	Purchase-0005 (1)	2	\$4,500.00
Total (5)		14	\$26,325.00

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, 14 total order count, 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 (12 entries) includes purchases on 9/10/2025 (4 items) and 9/11/2025 (8 items). Supplier-002 (2 entries) includes purchases on 9/11/2025 (3 items) and 9/12/2025 (2 items). Supplier-003 (3 entries) includes purchases on 9/12/2025 (2 items) and 9/13/2025 (1 item). Subtotals are shown for each group of items. The report interface includes standard Salesforce navigation and reporting tools like "Enable Field Editing", "Add Chart", and "Edit".

Total Records	Total Order Count	Total Quantity Received	Total Amount				
14	14	276	\$26,325.00				
Supplier ID	Actual Delivery Date	Purchase Order: Purchase Order ID	Product ID: Product ID	Order Count	Product ID: Product Name	Quantity Received	Amount
Supplier-001 (12)	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						
	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
				3		21	\$2,075.00
				2		15	\$2,250.00
				2		10	\$1,000.00
				2		25	\$3,250.00
				3		24	\$3,600.00
				3		56	\$1,400.00
Row Counts	Detail Rows	Subtotals	Grand Total				

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.