

PROJECT TITLE: Medical Inventory Management System

College Name: KG College of Arts and Science

College Code: bru4y

TEAM ID: NM2025TMID23506

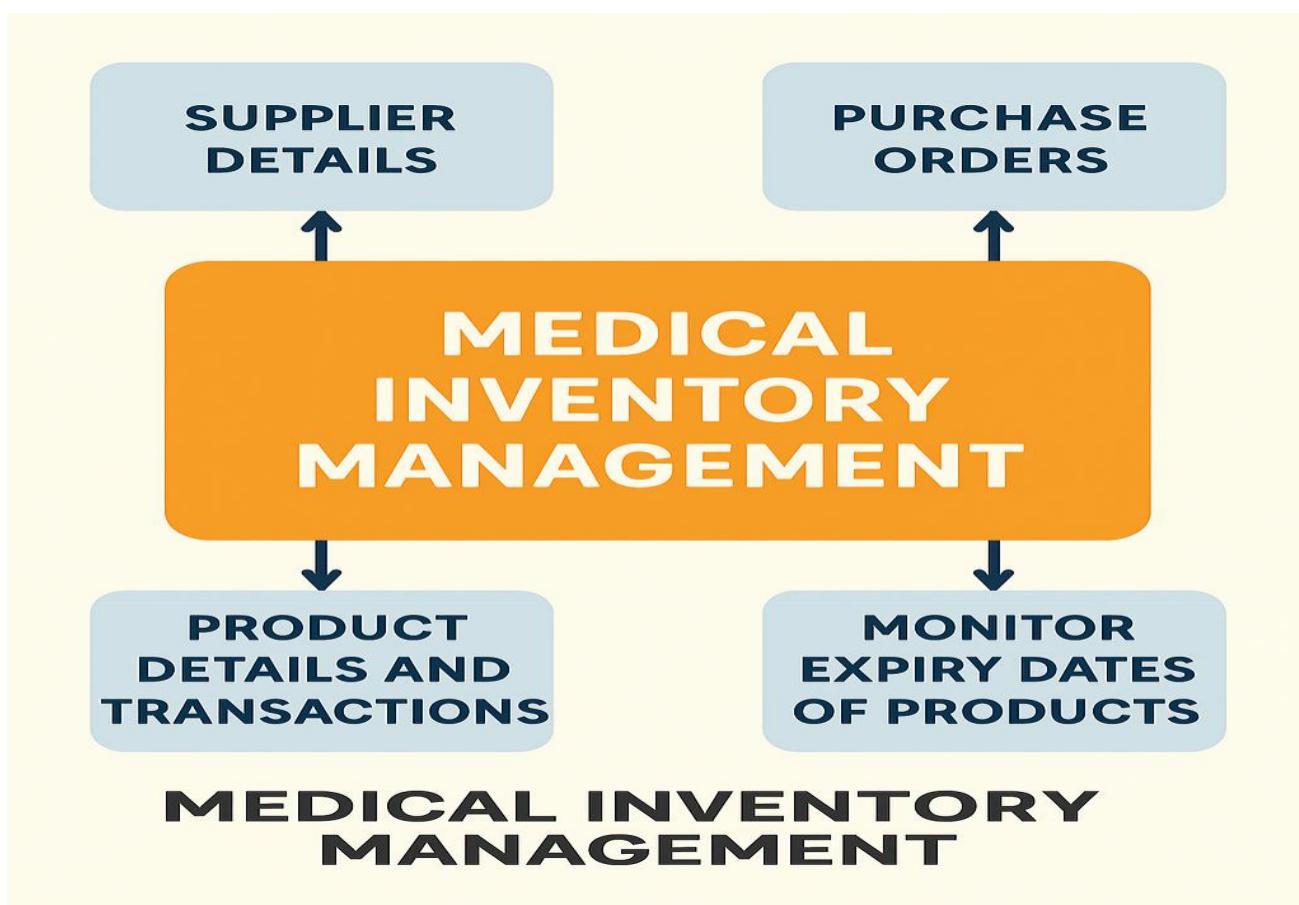
TEAM MEMBERS:

- **Team Leader Name:** PERUMAL PRADEEP S – 2322ja33@kgcas.com
- **Team Member 1:** PRATHVIRAJ D – 2322ja34@kgcas.com
- **Team Member 2:** RAHUL V – 2322ja35@kgcas.com
- **Team Member 3:** RANJITHA D – 2322ja36@kgcas.com

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.



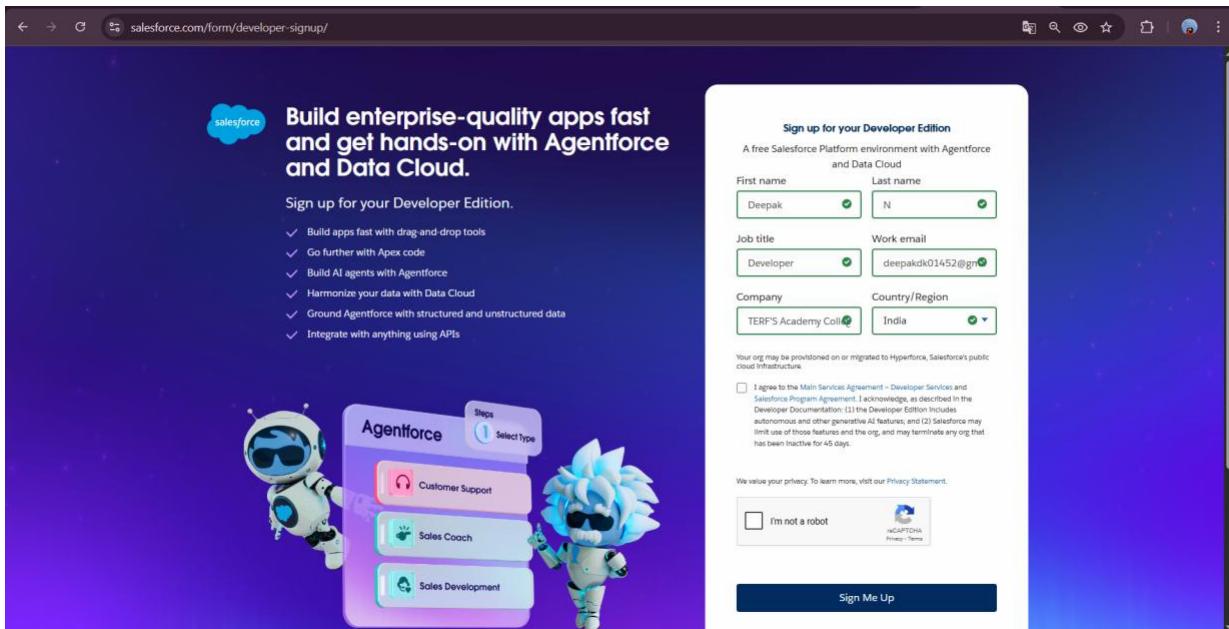
Purpose

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

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-58f3443d53-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK000002pi1/Details/view>. The page title is "Product". On the left, there's a sidebar with "Details" selected under "Fields & Relationships". The main area shows the "Details" tab for the Product object. Fields include:

- Description: (empty)
- API Name: Product_c (Custom checked)
- Singular Label: Product
- Plural Label: Products
- Enable Reports: ✓
- Track Activities: ✓
- Track Field History: (empty)
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

Buttons at the top right: Edit, Delete.

Creating a tab for Product Object and Remaining Tab

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-58f3443d53-dev-ed.lightning.force.com/lightning/setup/CustomTabs/home>. The page title is "Custom Tabs". The sidebar shows "User Interface" expanded with "Tabs" selected. The main area displays "Custom Tabs" with the following information:

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.

Custom Object Tabs

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

New | What Is This?

No Web Tabs have been defined.

Visualforce Tabs

New | What Is This?

No Visualforce Tabs have been defined.

Create a Lightning App for Medical Inventory Management

The screenshot shows the 'Lightning Experience App Manager' page in the Salesforce Setup interface. The left sidebar is collapsed, showing categories like 'Salesforce Mobile App', 'Data', 'Apps' (with 'App Manager' selected), 'Connected Apps', 'External Client Apps', and 'Lightning Bolt'. The main content area displays a table titled 'Lightning Experience App Manager' with 27 items. The columns are: App Name, Developer Name, Description, Last Modified, Type, and Action buttons. The table lists various apps such as Digital Experiences, Lightning Usage App, Marketing CRM Classic, and the newly created 'Medical Inventory Management' app.

The screenshot shows the 'Fields & Relationships' section for the 'Product' object in the Object Manager. The left sidebar shows options like 'Details', 'Fields & Relationships' (selected), 'Page Layouts', 'Lightning Record Pages', etc. 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 table lists 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.

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 like Page Layouts, Lightning Record Pages, and Field Sets. The main content area displays a table titled 'Fields & Relationships' with 10 items, sorted by Field Label. The columns include FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. Key fields shown include 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.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Actual Delivery Date	Actual_Delivery_Date__c	Date		
Created By	CreatedById	Lookup(User)		
Expected Delivery Date	Expected_Delivery_Date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Order Count	Order_Count__c	Roll-Up Summary (COUNT Order Item)		
Order Date	Order_Date__c	Date		
Owner	OwnerId	Lookup(User/Group)	✓	
Purchase Order ID	Name	Text(80)	✓	
Supplier ID	Supplier_ID__c	Lookup(Supplier)	✓	
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. The main content area displays a table titled 'Fields & Relationships' with 10 items, sorted by Field Label. The columns include FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. 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	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Formula (Currency)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Order Item ID	Name	Text(80)	✓	
Product ID	Product_ID__c	Lookup(Product)	✓	
Purchase Order ID	Purchase_Order_ID__c	Master-Detail(Purchase Order)	✓	
Quantity Ordered	Quantity_Ordered__c	Number(18, 0)		
Quantity Received	Quantity_Received__c	Number(18, 0)		
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. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Inventory Transaction'. On the left, a sidebar lists various setup options under 'Fields & Relationships'. The main content area is titled 'Fields & Relationships' with a sub-header '5 Items, Sorted by Field Label'. It displays a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table contains five rows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Inventory Transaction ID	Name	Text(80)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Product'. On the left, a sidebar lists various setup options under 'Page Layouts'. The main content area is titled 'Page Layouts' with a sub-header '00hgK000006ug5xQAA/view'. It displays a table of fields and their properties. At the bottom, there is a section for 'Product Detail' with buttons for 'Edit', 'Delete', 'Clone', 'Change Owner', 'Change Record Type', 'Printable View', 'Sharing', 'Sharing Hierarchy', and 'Edit Labels'.

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, 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 Layouts configuration for the Purchase Order Detail section. It includes a Fields panel with sections for Buttons, Quick Actions, Mobile & Lightning Actions, Expanded Lookups, Related Lists, and Report Charts. A Layout Properties bar at the top provides standard buttons (Save, Quick Save, Preview As...) and a Quick Find search bar. The main content area shows the Purchase Order Detail layout with sections for Information, System Information, and Custom Links, each containing sample data.

To edit a Page Layout in Order Item Object

The screenshot shows the Salesforce Setup interface for the Order Item object. The left sidebar lists various configuration options similar to the Purchase Order setup. The main area displays the Page Layouts configuration for the Order Item Detail section. It includes a Fields panel with sections for Buttons, Quick Actions, Mobile & Lightning Actions, Expanded Lookups, Related Lists, and Report Charts. A Layout Properties bar at the top provides standard buttons (Save, Quick Save, Preview As...) and a Quick Find search bar. The main content area shows the Order Item Detail layout with sections for Information, System Information, and Custom Links, each containing sample data. A note at the bottom of the layout section says "Drag expanded lookups and mobile-enabled Visualforce pages here to display them as mobile cards".

To edit a Page Layout in Inventory Transaction Object

The screenshot shows the Salesforce Setup interface for the Inventory Transaction object. The left sidebar lists various configuration options like Details, Fields & Relationships, 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 'Page Layouts' tab is selected. The main area displays the Page Layout editor for the Inventory Transaction object. It includes a 'Fields' section with a quick find bar, a 'Layout Properties' tab, and sections for Information, System Information, Custom Links, Mobile Cards (Salesforce mobile only), and Related Lists. A preview pane shows the layout structure with fields like Inventory Transaction ID, Transaction Type, Created By, Last Modified By, Owner, and Supplier Name.

To edit a Page Layout in Supplier Object

The screenshot shows the Salesforce Setup interface for the Supplier object. The left sidebar lists various configuration options like Details, Fields & Relationships, 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 'Page Layouts' tab is selected. The main area displays the Page Layout editor for the Supplier object. It includes a 'Fields' section with a quick find bar, a 'Layout Properties' tab, and sections for Supplier Detail, Information, System Information, Custom Links, and Mobile Cards (Salesforce mobile only). A preview pane shows the layout structure with fields like Supplier ID, Supplier Name, Contact Person, Phone Number, Email, Address, and Owner. It also shows standard buttons like Edit, Delete, Clone, Change Owner, Change Record Type, Printable View, Sharing, Sharing Hierarchy, and Edit Labels.

To create a Compact Layout to a Product Object

The screenshot shows the Salesforce Setup interface for creating a compact layout for the Product object. The left sidebar navigation includes: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts (which is selected), 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 "Edit Product Compact Layout" and "Product Compact Layout". It contains a "Compact Layout Edit" section with "Enter Compact Layout Information" fields for Label ("Product Compact Layout") and Name ("Product_Compact_Layout"). Below this is a "Select Compact Layout Fields" section. On the left is a list of "Available Fields": Created By, Expiry Date, Last Modified By, Minimum Stock Level, Owner, and Product ID. On the right is a list of "Selected Fields": Unit Price, Product Name, Current Stock Level, Top, Up, Down, and Bottom. A note at the bottom says "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 Salesforce Setup interface for creating a compact layout for the Purchase Order object. The left sidebar navigation includes: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts (which is selected), 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 "Edit Purchase Order Compact Layout" and "Purchase Order Compact Layout". It contains a "Compact Layout Edit" section with "Enter Compact Layout Information" fields for Label ("Purchase Order Compact L") and Name ("Purchase_Order_Compact_"). Below this is a "Select Compact Layout Fields" section. On the left is a list of "Available Fields": Actual Delivery Date, Created By, Expected Delivery Date, Last Modified By, Order Count, and Owner. On the right is a list of "Selected Fields": Purchase Order ID, Order Date, Total Order Cost, Supplier ID, Top, Up, Down, and Bottom. 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

The screenshot shows the Salesforce Setup interface for creating a validation rule. The left sidebar lists various object settings 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" and has the formula "(Expected_Delivery_Date__c - Order_Date__c) > 7". The error message is "The Expected Delivery Date should not exceed 7 days." The rule is active and was created by Deepak.N on 9/5/2025 at 11:25 PM.

To create an Purchase Manager Profile

The screenshot shows the Salesforce Setup interface for creating a user profile. The left sidebar shows "Users" and "Profiles". The main content area is titled "Purchase Manager" and displays the "Profile Detail" section. The profile name is "Purchase Manager", assigned to the "Salesforce" user license, and is a "Custom Profile". The "Page Layouts" section shows standard object layouts for various objects like Global, Email Application, Home Page Layout, Account, Alternative Payment Method, and Appointment Invitation, each mapped to specific page layouts. The "Location Group Assignment" section shows assignments for Macro Layout, Object Milestone, Operating Hours, Opportunity, and Opportunity Product.

Create a Purchasing Manager Role.

The screenshot shows the Salesforce Setup interface for creating a new role. The URL in the browser is <https://orgfarm-58f3443d53-dev-ed.lightning.force.com/lightning/setup/Roles/page?address=%2F00EgK000003xbcD%3fsetupid%3DRoles>.

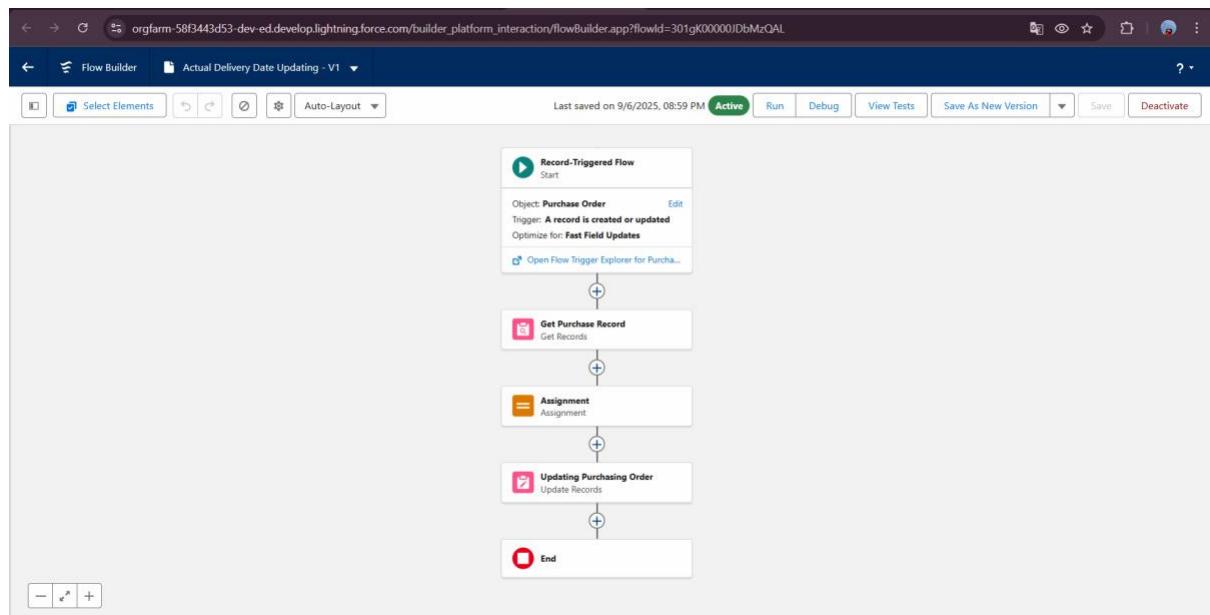
Role Detail:

Label	Purchasing Manager	Role Name	Purchasing_Manager
This role reports to	SVP_Sales & Marketing	Role Name as displayed on reports	
Modified By	Deepak.N 9/5/2025, 11:50 PM	Sharing Groups	Role, Role and Internal Subordinates
Opportunity Access	Users in this role can edit all opportunities associated with accounts that they own, regardless of who owns the opportunities		
Case Access	Users in this role can edit all cases associated with accounts that they own, regardless of who owns the cases		

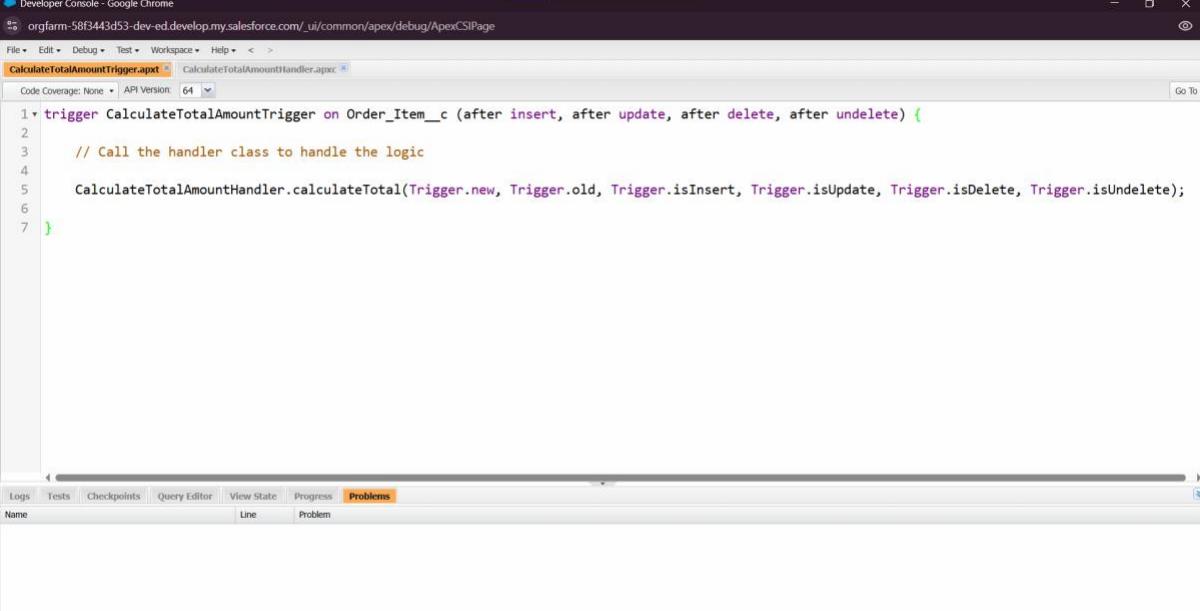
Users in Purchasing Manager Role:

Action	Full Name	Alias	Username	Active
Edit	John PurchaseM	JohnP	johnpurchaseM@test.com	✓

Create Flow to update the Actual Delivery Date.



Create a Trigger to Calculate total amount on Order Item.

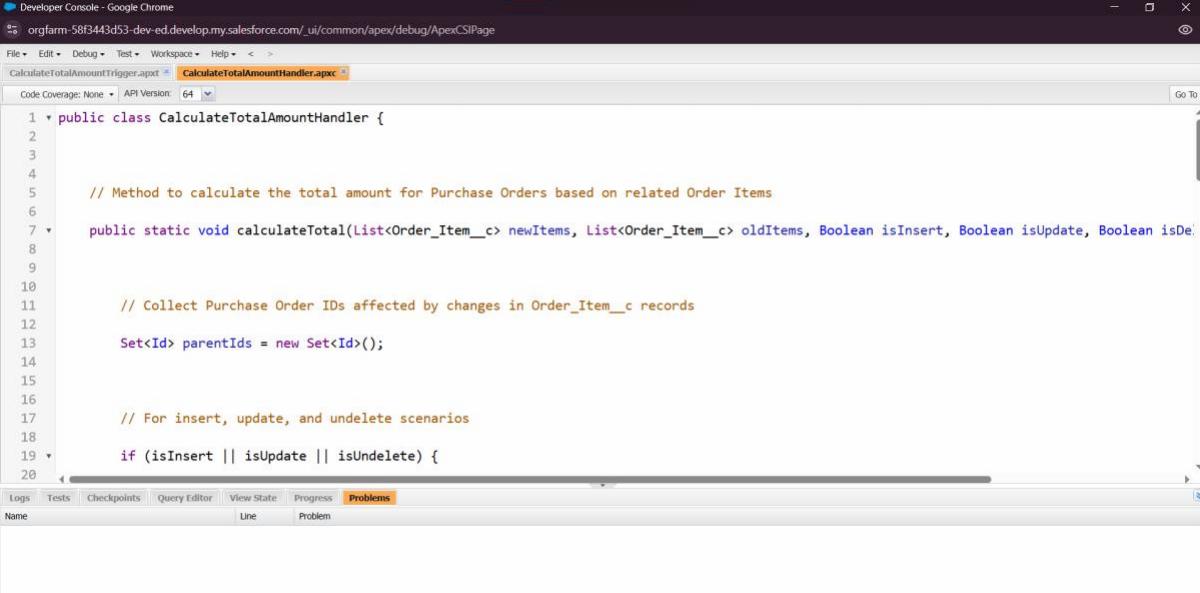


The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-58f3443d53-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage. The tab bar has two tabs: 'CalculateTotalAmountTrigger.apxc' and 'CalculateTotalAmountHandler.apxc'. The 'CalculateTotalAmountTrigger.apxc' tab is active. The code editor contains the following Apex trigger:

```
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 bottom navigation bar shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected. There are no problems listed.

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/ApexCSPage. The tab bar has two tabs: 'CalculateTotalAmountTrigger.apxc' and 'CalculateTotalAmountHandler.apxc'. The 'CalculateTotalAmountHandler.apxc' tab is active. The code editor contains 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(List<Order_Item__c> newItems, List<Order_Item__c> oldItems, Boolean isInsert, Boolean isUpdate, Boolean isDelete) {
7
8
9         // Collect Purchase Order IDs affected by changes in Order_Item__c records
10
11         Set<Id> parentIds = new Set<Id>();
12
13
14
15
16         // For insert, update, and undelete scenarios
17
18         if (isInsert || isUpdate || isDelete) {
19
20 }
```

The bottom navigation bar shows tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected. There are no problems listed.

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 table includes:

Total Records	Total Order Count	Total Total Order Cost
5	14	\$26,325.00

The main table displays data grouped by Supplier ID, with rows for Purchase-0001 (4), Purchase-0002 (1), Purchase-0003 (1), Purchase-0004 (1), and Purchase-0005 (1). The columns are Supplier ID, Purchase Order: Purchase Order ID, Order Count, and Total Order Cost.

At the bottom, there are checkboxes for Row Counts, Detail Rows, Subtotals, and Grand Total.

Create a Complete Purchase Details Report

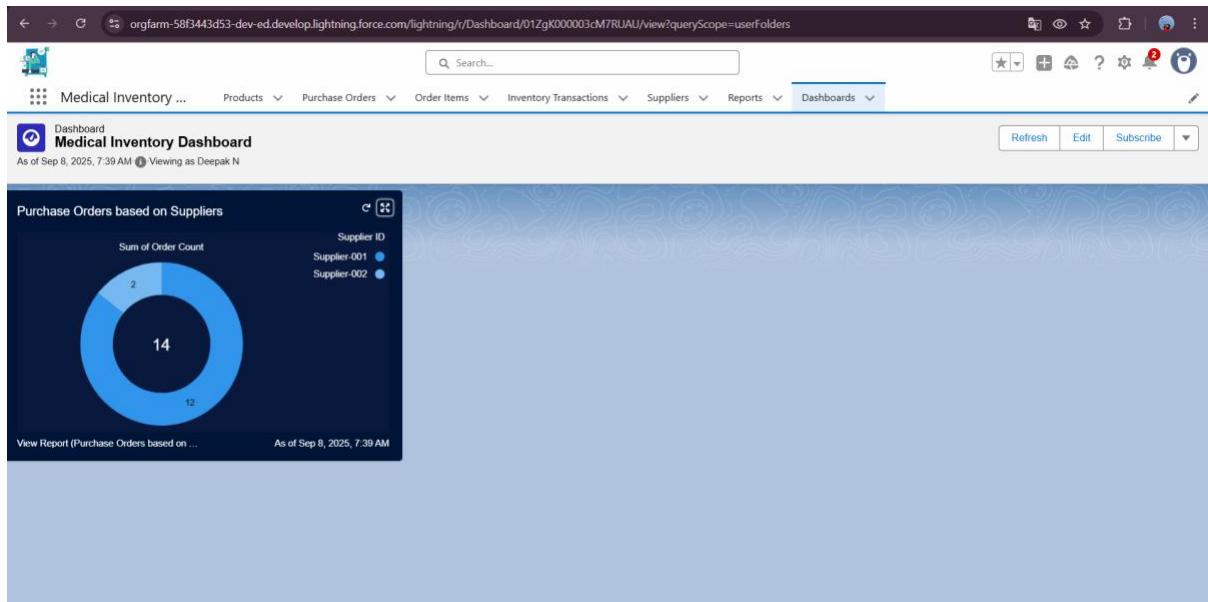
The screenshot shows a Salesforce Lightning report titled "Complete Purchase Details Report". The report summary table includes:

Total Records	Total Order Count	Total Quantity Received	Total Amount
14	14	276	\$26,325.00

The main table displays data grouped by Supplier ID and Actual Delivery Date, with rows for Purchase-0004 (4), Purchase-0001 (3), Purchase-0002 (2), and Purchase-0003 (3). The columns include Supplier ID, Actual Delivery Date, Purchase Order: Purchase Order ID, Product ID: Product ID, Order Count, Product ID: Product Name, Quantity Received, and Amount.

Subtotal rows are present for each group of delivery dates. At the bottom, there are checkboxes for Row Counts, Detail Rows, Subtotals, and 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.