

MEDICAL INVENTORY MANAGEMENT SYSTEM

SALESFORCE PROJECT DOCUMENTATION REPORT

TEAM ID: NM2025TMID03981

Team Size: 3

Team Leader: Kaviya K

Team Members: Devaki V, Elakkia M J

GitHub Repository: <https://github.com/Elakkia-19/Medical-Inventory-Management-System---SF>

ABSTRACT

- The Medical Inventory Management System is a cloud-based application built on the Salesforce platform to automate and simplify the handling of medical supplies. The system efficiently records and manages supplier details, purchase orders, product information, inventory transactions, and expiry dates.
- It utilizes various Salesforce features such as custom objects, validation rules, flows, triggers, and dashboards to ensure seamless operations and data accuracy.
- The primary goal is to eliminate manual tracking, enhance efficiency, and provide real-time visibility into inventory levels and supplier performance through automation and analytics.
- Technologies Used:
- Salesforce Lightning Platform
- Apex Trigger and Handler Classes
- Validation Rules & Automation Flows
- Reports and Dashboards
- This project serves as a complete Salesforce-based solution for medical inventory control, offering accuracy, transparency.

CHAPTER 1 – INTRODUCTION

1.1 Overview

The Medical Inventory Management System aims to simplify the management of medical items by automating the entire process using Salesforce CRM tools.

It allows users to record suppliers, purchase orders, inventory transactions, and product details, along with real-time tracking of stock and expiry dates.

This project provides an end-to-end automated solution, reducing the risk of human errors.

1.2 Objectives

- To automate the medical inventory process through Salesforce.
- To create and interlink objects for managing products, suppliers, and transactions.
- To build validation rules and flows for data consistency.
- To develop visual dashboards and reports for analysis and decision-making.

1.3 Scope

The system supports:

- Managing suppliers and purchase orders.
- Monitoring product quantities and expiry dates.
- Automated order tracking through flows and triggers.
- Visual dashboards for instant insights into business performance.

1.4 Existing vs Proposed System

Existing System	Proposed System (Salesforce)
Manual data entry and record-keeping	Fully automated record system
Data errors and missing records	High accuracy with validation rules
No expiry monitoring	Automatic expiry tracking
Time-consuming analysis	Instant dashboards and reports

CHAPTER 2 – PROJECT IDEATION (Phase 1)

2.1 Problem Identification

Traditional methods of maintaining medical inventory are time-consuming and error-prone. There is often no proper system to track supplier activity, expired items, or current stock levels.

2.2 Idea Description

This project focuses on developing a Salesforce application that automates all operations related to medical inventory. It provides a single platform where purchase orders, suppliers, and stock can be managed effectively, ensuring smooth coordination and data visibility.

2.3 Feasibility

Since Salesforce is a cloud-based CRM with prebuilt automation and reporting tools, the project is cost-efficient, scalable, and easy to implement without additional hardware requirements.

2.4 Innovation

- Use of flows to auto-update delivery dates.
- Trigger to calculate the total purchase amount dynamically.
- Dashboards to visualize supplier efficiency and purchase statistics.

CHAPTER 3 – REQUIREMENT ANALYSIS (Phase 2)

3.1 Functional Requirements

- Add and manage suppliers, products, and purchase orders.
- Update and validate order and transaction data automatically.
- Generate detailed analytical reports and dashboards.
- Assign roles and permissions for secure access.

3.2 Non-Functional Requirements

- Cloud-based and secure platform.
- Simple, intuitive user interface.
- Flexible and scalable structure for future features.

3.3 User Roles

- Administrator: Has complete control and configuration rights.
- Purchase Manager: Handles purchase orders and supplier data.
- Inventory Manager: Manages stock, expiry, and transactions.

3.4 Data Flow

Relationships:

Supplier → Purchase Order → Order Item → Product → Inventory Transaction

The entities are connected using Lookup and Master-Detail relationships for effective data flow between modules.

CHAPTER 4 – PROJECT DESIGN (Phase 3)

4.1 Schema Builder

Displays all custom objects and their interrelationships visually, helping to understand data flow and dependencies.

4.2 Process Flow

- Supplier data is added.
- Products and purchase orders are created.
- Order items and transactions are updated.
- Delivery and cost details are auto-calculated.
- Reports and dashboards display key performance metrics.

4.3 Interface Design

Created a custom Lightning App with separate tabs for each module—Suppliers, Products, Orders, Transactions, Reports, and Dashboards—ensuring smooth navigation.

CHAPTER 5 – PROJECT DEVELOPMENT (Phase 4)

5.1 Milestones Summary

The development process followed 17 milestones covering setup to conclusion:

- M1: Salesforce Account Creation
- M2: Object Creation
- M3: Tabs
- M4: Lightning App
- M5: Fields
- M6: Page Layouts
- M7: Compact Layouts
- M8: Validation Rules
- M9: Profiles
- M10: Roles
- M11: Users
- M12: Permission Sets
- M13: Flows
- M14: Triggers
- M15: Reports
- M16: Dashboards
- M17: Conclusion

5.2 Key Apex Code (Handler and Trigger)

A trigger was implemented to calculate total order cost in the Purchase Order object whenever Order Items are inserted or updated.

5.3 Flows

Flows were created to auto-update Actual Delivery Date and ensure dynamic updates across related records.

5.4 Reports

Two reports were designed:

1. Purchase Orders based on Suppliers
2. Complete Purchase Details Report

CHAPTER 6 – PROJECT DOCUMENTATION (Phase 5)

6.1 Execution Steps

- Created a Salesforce Developer Org.
- Defined custom objects and established relationships.
- Added fields, validation rules, and triggers.
- Designed reports and dashboards for visualization.
- Uploaded deliverables and documentation to GitHub.

6.2 GitHub Repository

Link: <https://github.com/Elakkia-19/Medical-Inventory-Management-System---SF>

6.3 Fixes & Adjustments

Issue	Resolution
Missing data in reports	Added test data for validation
Layout misalignment	Rearranged fields in Page Layout
Validation error	Corrected formula logic

CHAPTER 7 – PROJECT DEMONSTRATION & RESULTS

7.1 Demo Process

- The team demonstrated the complete workflow:
- Creating records for suppliers and products.
- Generating purchase orders and order items.
- Auto-updating delivery and total cost via triggers and flows.
- Analyzing data using reports and dashboards.

7.2 Outputs

- Data is automatically validated and updated.
- Reports generate accurate insights.
- Dashboards reflect real-time summary of purchase and supplier activity.

CHAPTER 8 – CONCLUSION & FUTURE ENHANCEMENT

8.1 Conclusion

The Medical Inventory Management System provides an automated approach to managing medical supplies using Salesforce CRM. It ensures accuracy, reduces manual errors, and improves transparency in stock and order tracking.

The project demonstrates the use of Salesforce tools like triggers, flows, and dashboards to simplify real-world inventory management.

8.2 Limitations

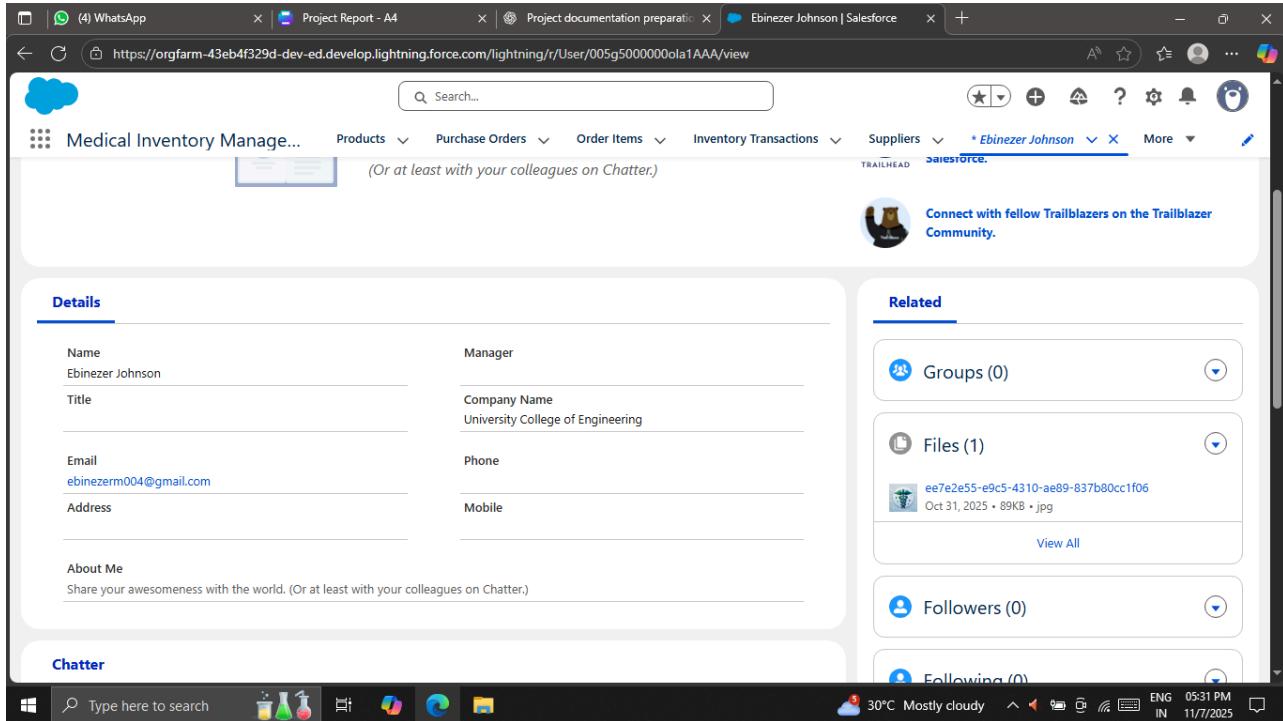
- Limited to Salesforce environment.
- Internet access required for operation.

8.3 Future Enhancements

- Enable automatic email alerts for expired items.
- Integrate barcode or QR scanning for inventory updates.
- Develop a mobile-friendly version of the application.

APPENDIX – SCREENSHOTS

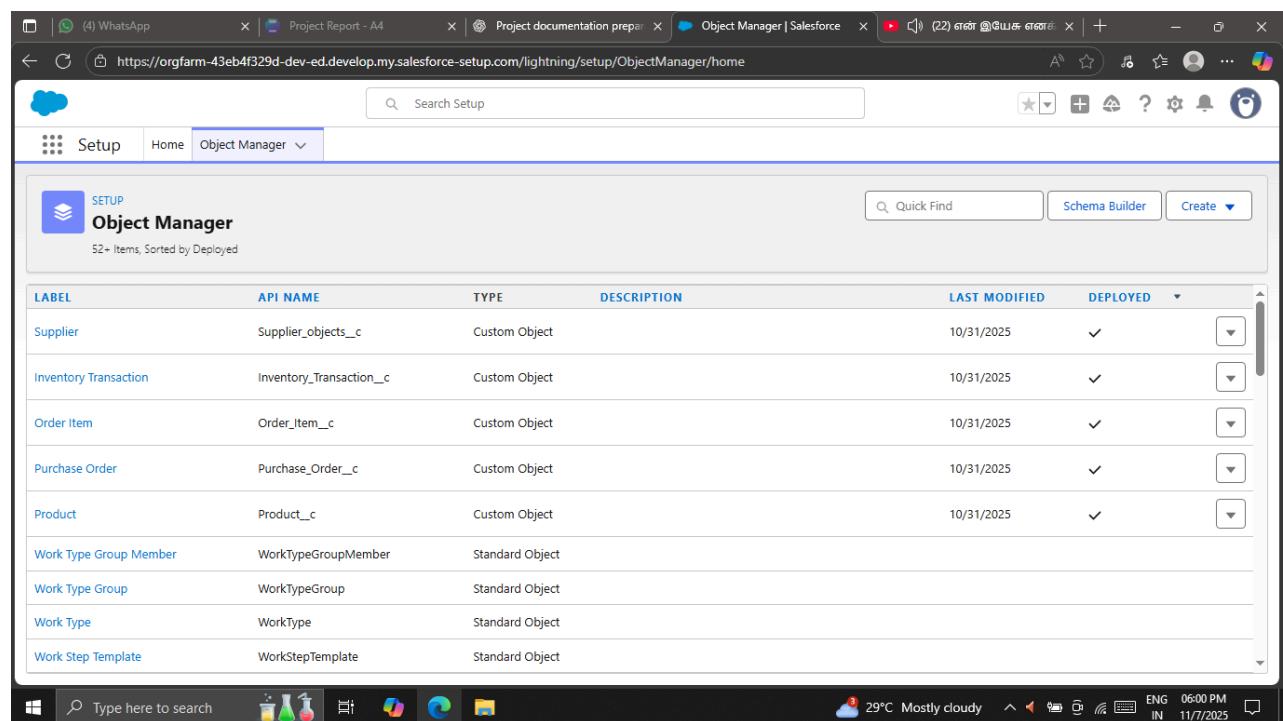
1) Salesforce Developer Account Setup



The screenshot shows the Salesforce Lightning Experience interface. The top navigation bar includes tabs for WhatsApp, Project Report - A4, Project documentation preparation, and Ebenezer Johnson | Salesforce. The main content area displays the profile details for Ebenezer Johnson, including Name, Title, Email (ebinezerm004@gmail.com), Address, Manager (University College of Engineering), Company Name, Phone, and Mobile. Below the details, there's a "About Me" section with a placeholder message. To the right, a "Related" sidebar lists Groups (0), Files (1) containing a file named "ee7e2e55-e9c5-4310-ae89-837b80cc1f06" (Oct 31, 2025, 89KB), Followers (0), and Following (0). The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

Salesforce Developer Account successfully created and activated.

2) Custom Objects Created



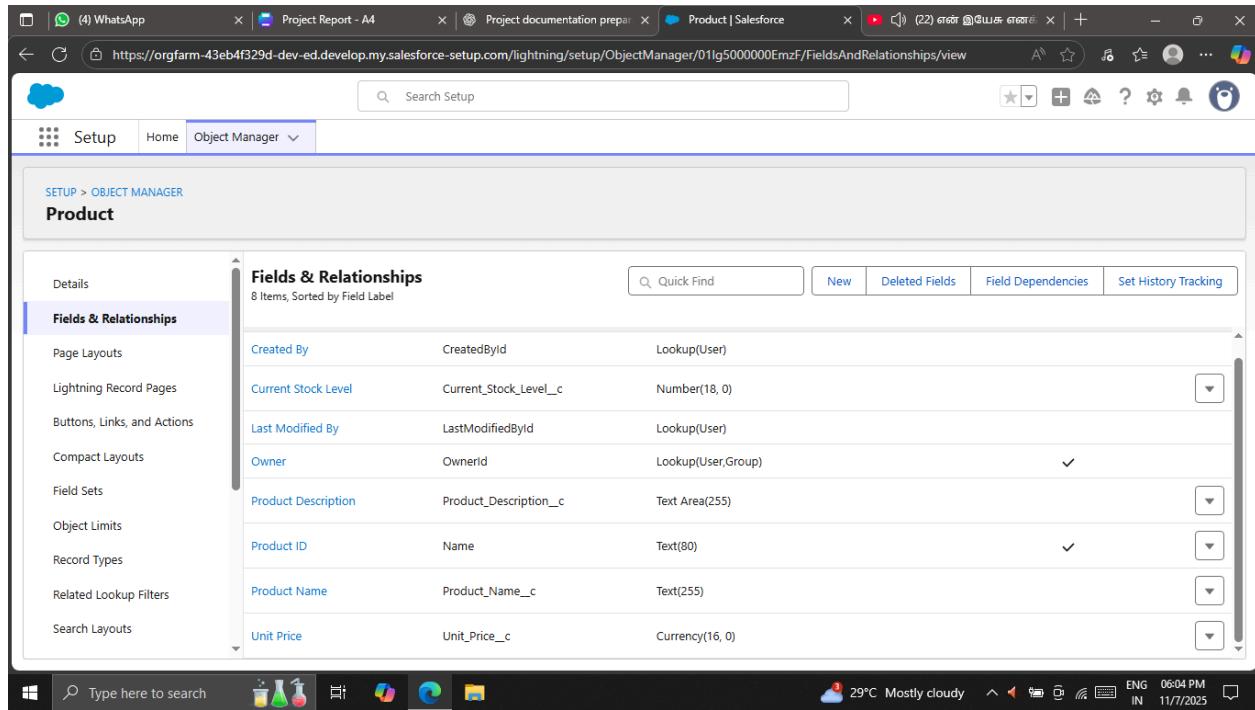
The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for Setup, Home, and Object Manager. The main content area displays a table of custom objects, each with a Label, API Name, Type, Description, Last Modified, and Deployed status. The objects listed are Supplier, Inventory Transaction, Order Item, Purchase Order, Product, Work Type Group Member, Work Type Group, Work Type, and Work Step Template. All objects were last modified on 10/31/2025 and are deployed. The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Supplier	Supplier_objects_c	Custom Object		10/31/2025	✓
Inventory Transaction	Inventory_Transaction_c	Custom Object		10/31/2025	✓
Order Item	Order_Item_c	Custom Object		10/31/2025	✓
Purchase Order	Purchase_Order_c	Custom Object		10/31/2025	✓
Product	Product_c	Custom Object		10/31/2025	✓
Work Type Group Member	WorkTypeGroupMember	Standard Object			
Work Type Group	WorkTypeGroup	Standard Object			
Work Type	WorkType	Standard Object			
Work Step Template	WorkStepTemplate	Standard Object			

Custom Objects created for Medical Inventory Management System.

APPENDIX – SCREENSHOTS

3) Product Object – Fields & Relationships

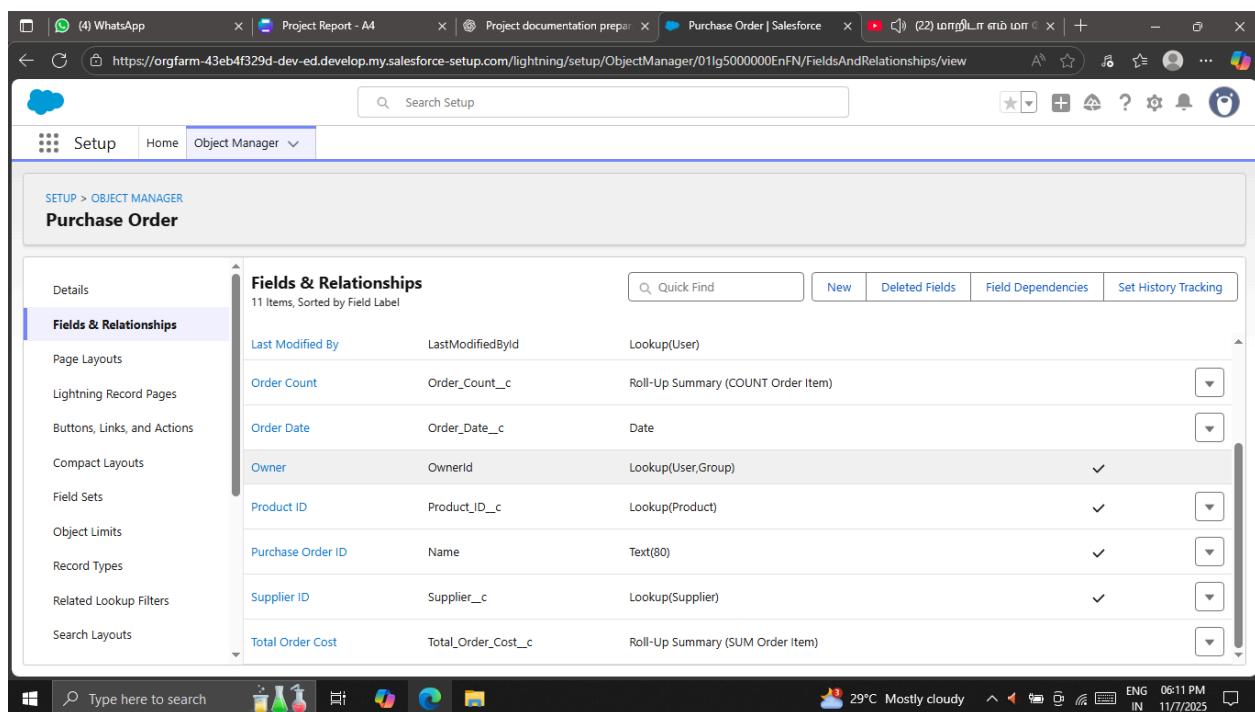


The screenshot shows the Salesforce Object Manager interface for the 'Product' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area is titled 'Fields & Relationships' and displays 8 items, sorted by Field Label. The fields listed are:

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

Fields and Relationships defined for Product Object.

4) Purchase Order Object – Lookup & Roll-Up Fields



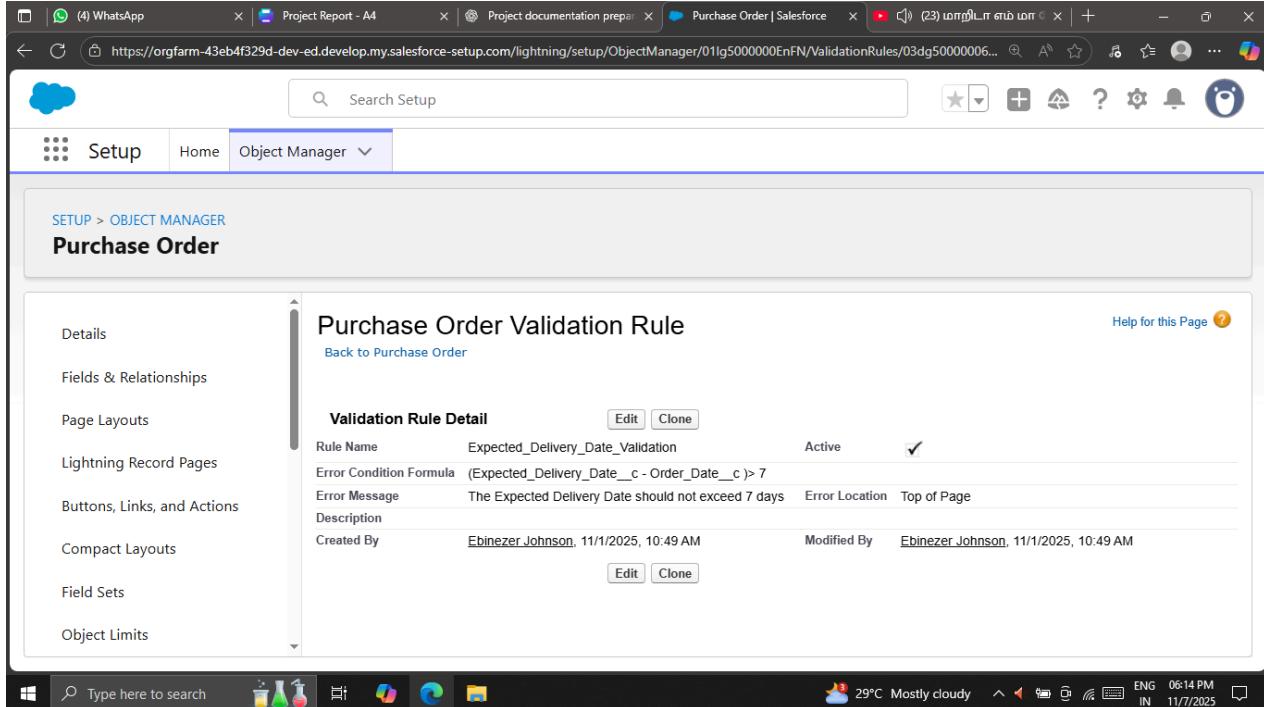
The screenshot shows the Salesforce Object Manager interface for the 'Purchase Order' object. The left sidebar lists various setup options. The main area is titled 'Fields & Relationships' and displays 11 items, sorted by Field Label. The fields listed are:

Field Name	Field Type	Description
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)
Product ID	Product_ID_c	Lookup(Product)
Purchase Order ID	Name	Text(80)
Supplier ID	Supplier_c	Lookup(Supplier)
Total Order Cost	Total_Order_Cost_c	Roll-Up Summary (SUM Order Item)

Lookup and Roll-Up Summary fields configured in Purchase Order Object.

APPENDIX – SCREENSHOTS

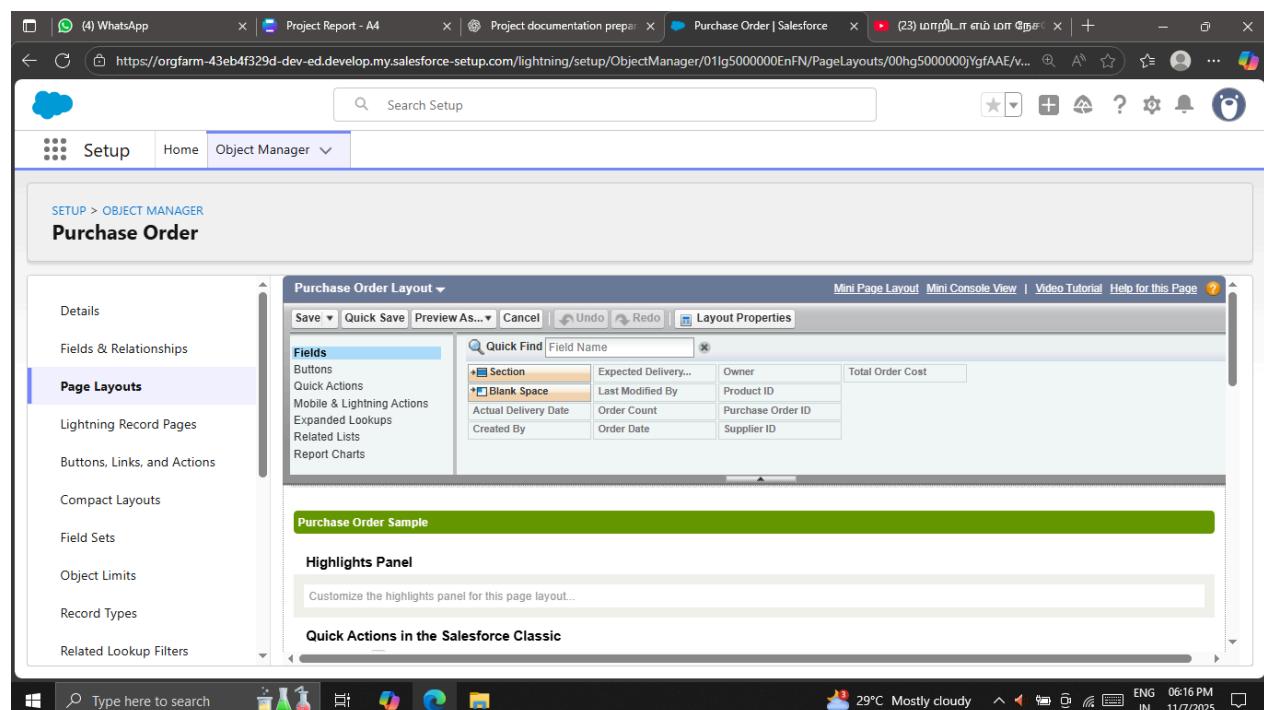
5) Validation Rule



The screenshot shows the 'Purchase Order Validation Rule' configuration in the Salesforce Object Manager. The validation rule is named 'Expected_Delivery_Date_Validation'. The error condition formula is '(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 Ebinezer Johnson on 11/1/2025 at 10:49 AM.

Validation Rule to restrict Expected Delivery Date beyond 7 days.

6) Page Layout Editing

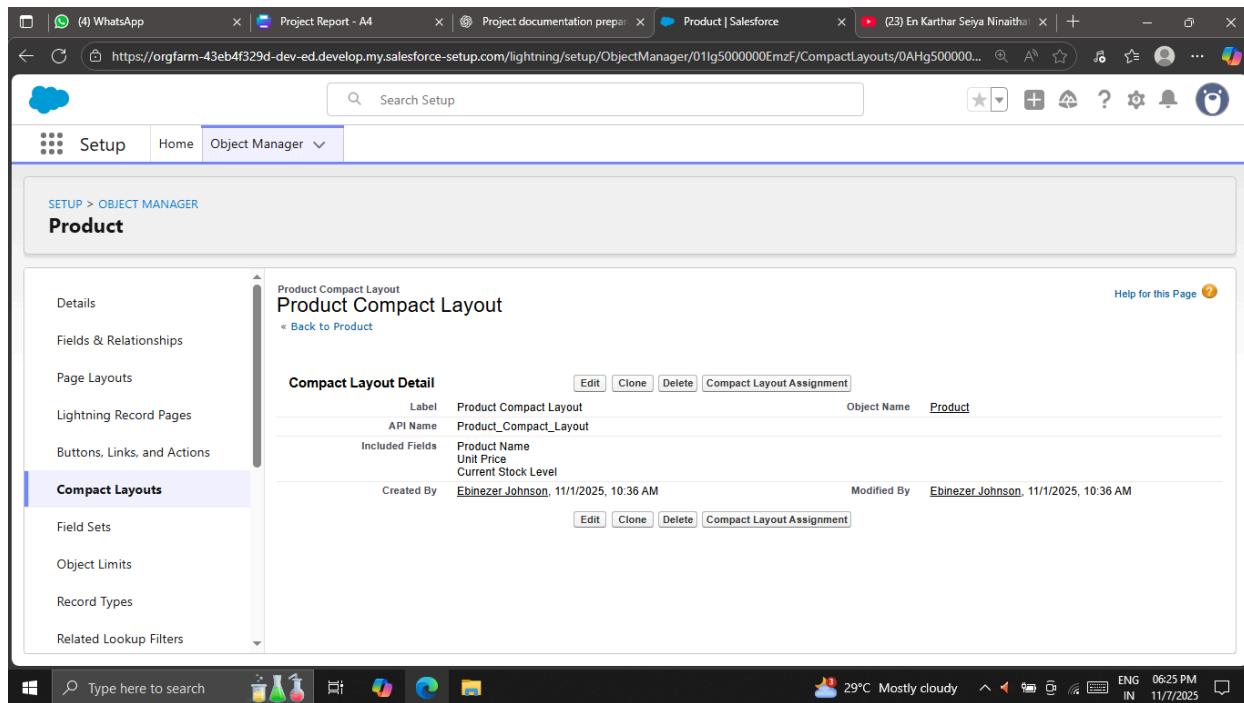


The screenshot shows the 'Purchase Order Layout' configuration in the Salesforce Object Manager. A new page layout is being created, featuring fields for Expected Delivery Date, Last Modified By, Product ID, Order Count, Purchase Order ID, and Order Date. The layout also includes sections for 'Purchase Order Sample' and 'Highlights Panel'.

Customized Page Layout for Purchase Order Object with required fields.

APPENDIX – SCREENSHOTS

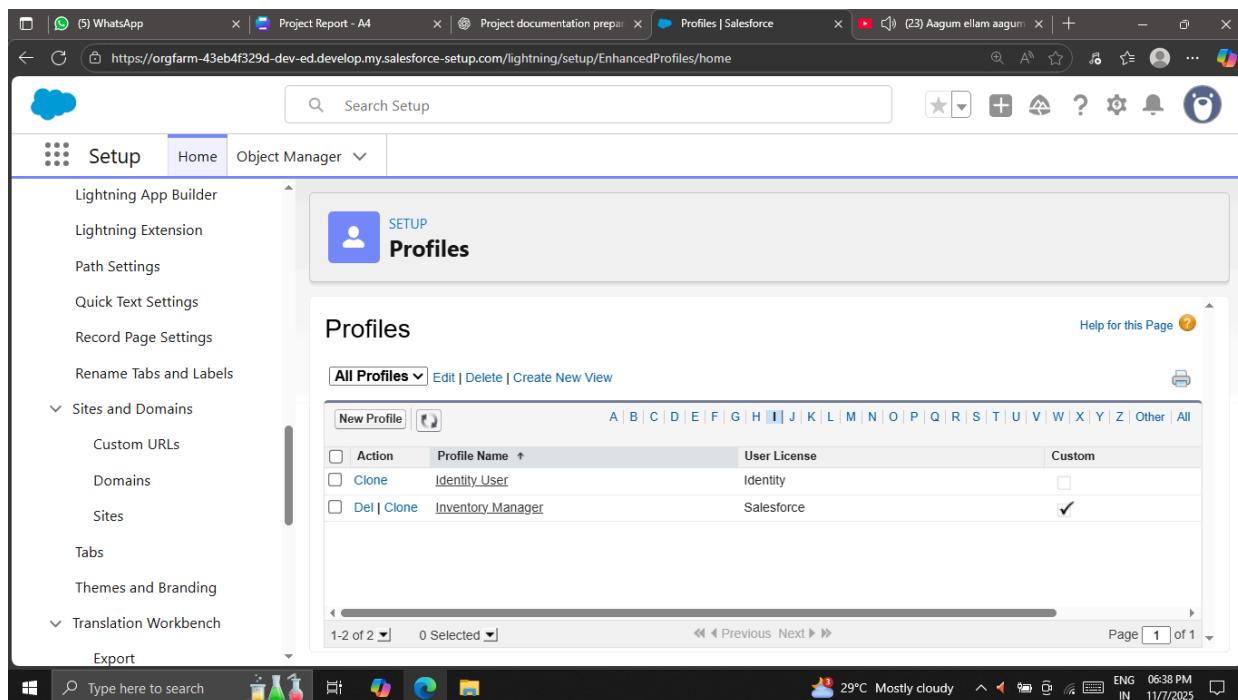
7) Compact Layout



The screenshot shows the Salesforce Setup interface for creating a new Compact Layout. The left sidebar is titled 'Object Manager' and has a 'Compact Layouts' section selected. The main content area is titled 'Product Compact Layout' and displays the 'Compact Layout Detail' for 'Product Compact Layout'. The layout includes fields for 'Label' (Product Compact Layout), 'API Name' (Product_Compact_Layout), and 'Included Fields' (Product Name, Unit Price, Current Stock Level). The layout was created by Ebenezer Johnson on 11/1/2025 at 10:36 AM and modified by Ebenezer Johnson on 11/1/2025 at 10:36 AM.

Compact Layout created to display key fields in record highlights panel.

8) Profiles and Roles

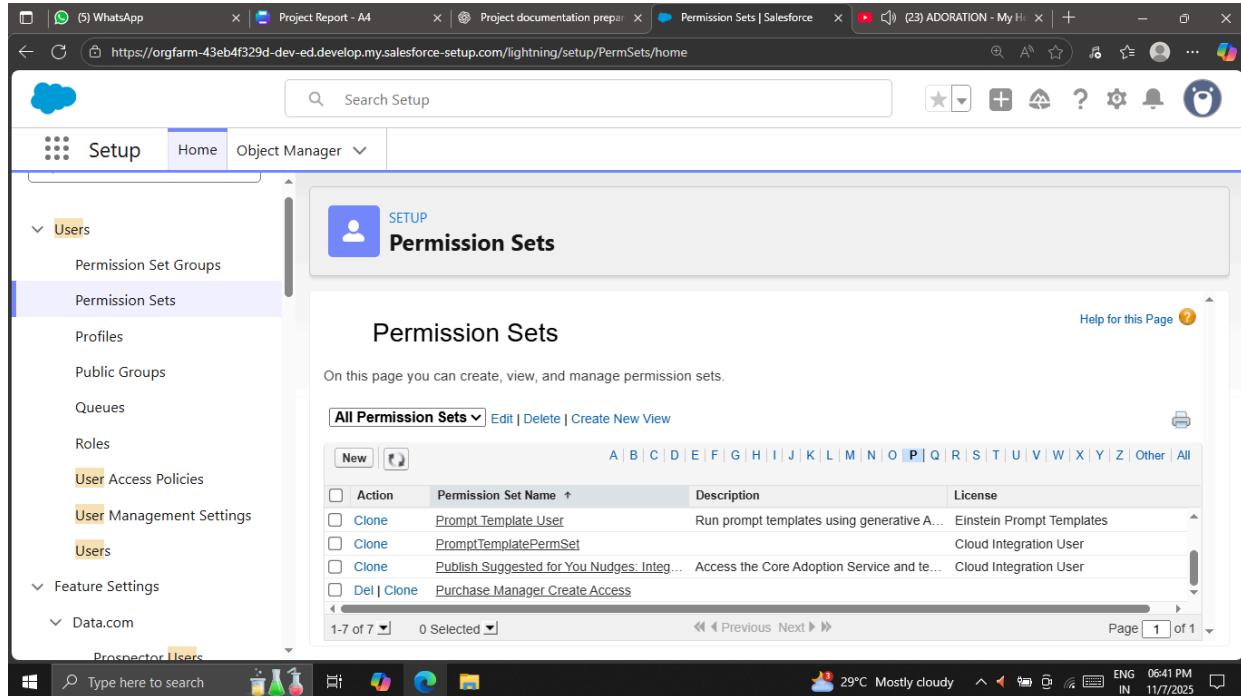


The screenshot shows the Salesforce Setup interface for managing Profiles. The left sidebar has sections like Lightning App Builder, Lightning Extension, Path Settings, Quick Text Settings, Record Page Settings, Rename Tabs and Labels, Sites and Domains (with sub-options for Custom URLs, Domains, and Sites), Tabs, Themes and Branding, Translation Workbench, and Export. The main content area is titled 'Profiles' and shows a table of existing profiles. The table includes columns for Action, Profile Name, User License, and Status. Two profiles are listed: 'Identity_User' (User License: Identity, Status: Custom) and 'Inventory_Manager' (User License: Salesforce, Status: checked).

Custom Profiles and Roles created for user access management.

APPENDIX – SCREENSHOTS

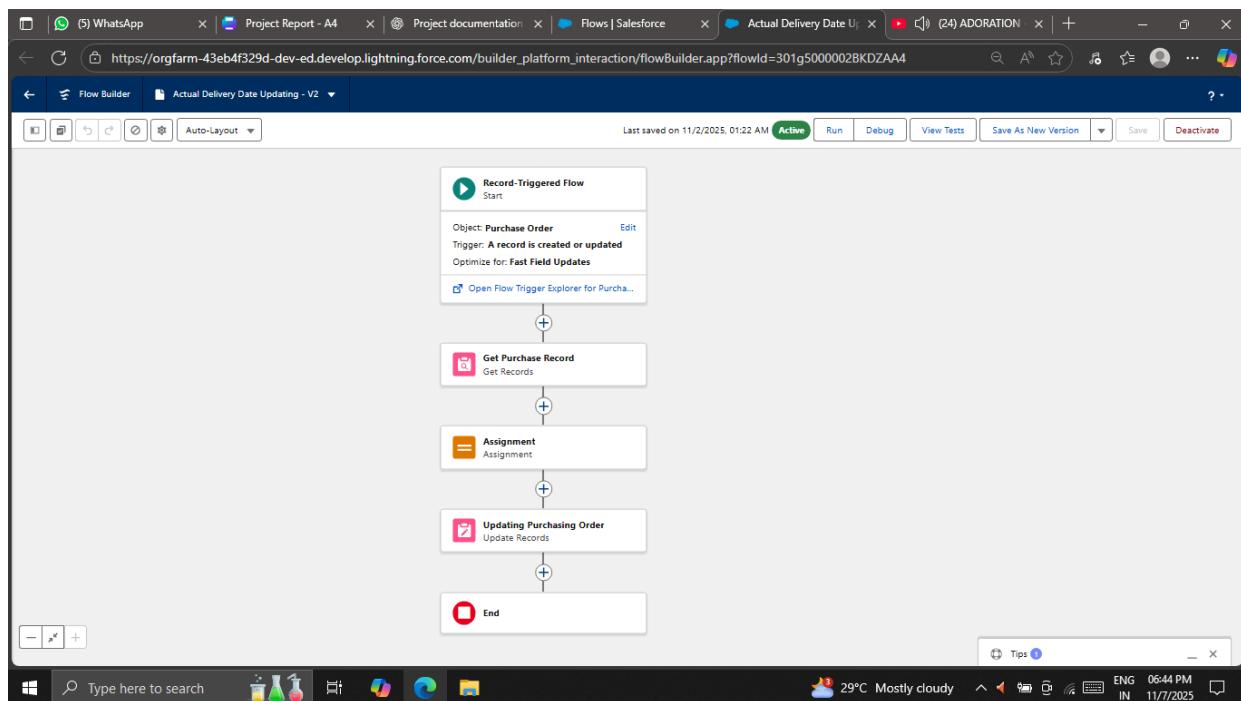
9) Permission Set



The screenshot shows the Salesforce Setup interface under the 'Permission Sets' section. The left sidebar includes options like 'Users', 'Profiles', 'Public Groups', 'Queues', 'Roles', 'User Access Policies', 'User Management Settings', and 'Users'. The main content area displays a table of permission sets with columns for Action, Permission Set Name, Description, and License. The table lists several entries, including 'Prompt Template User' (Einstein Prompt Templates), 'PromptTemplatePermSet' (Cloud Integration User), and 'Purchase Manager Create Access' (Cloud Integration User). A navigation bar at the bottom indicates 1-7 of 7 results.

Permission Set assigned for creating and accessing Order Item records.

10) Flow Builder

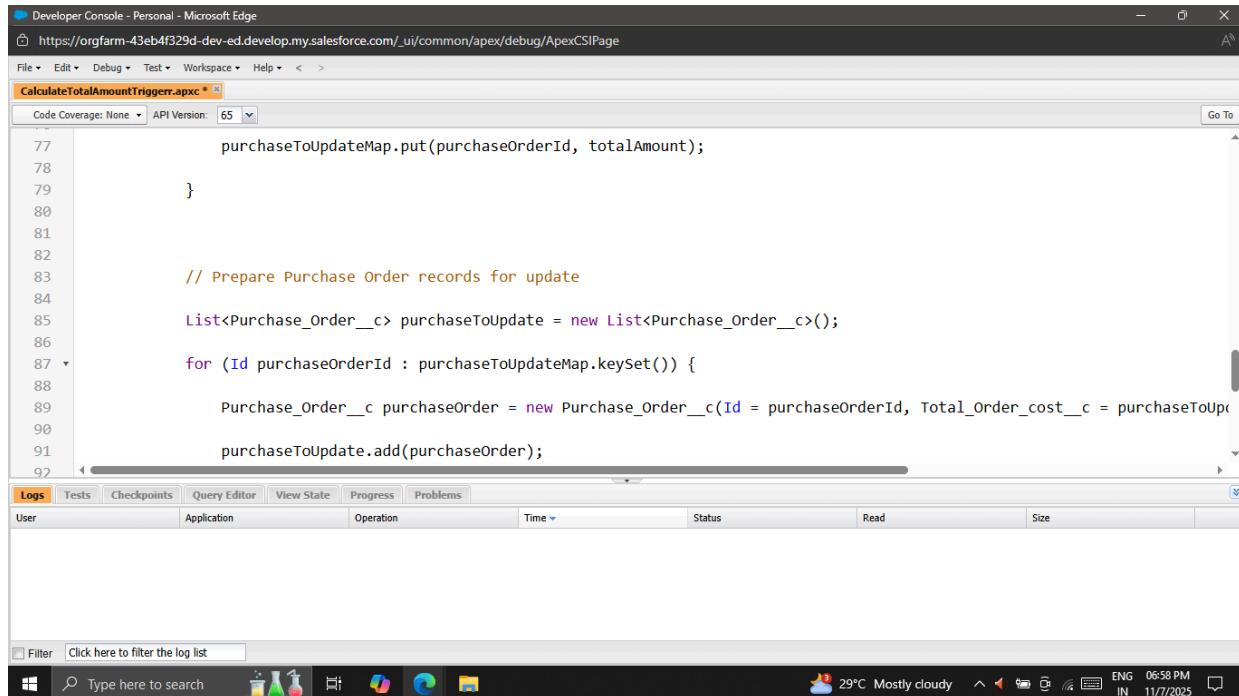


The screenshot shows the Salesforce Flow Builder interface for a 'Record-Triggered Flow'. The flow starts with a 'Record-Triggered Flow Start' step for 'Purchase Order' objects triggered by a record creation or update. This is followed by a 'Get Purchase Record' step, an 'Assignment' step, and an 'Updating Purchasing Order' step. The flow concludes with an 'End' step. The status bar at the bottom indicates the flow was last saved on 11/2/2025 at 01:22 AM and is currently active.

Record-Triggered Flow created to auto-update Actual Delivery Date in Purchase Orders.

APPENDIX – SCREENSHOTS

11) Apex Trigger and Handler Class



```

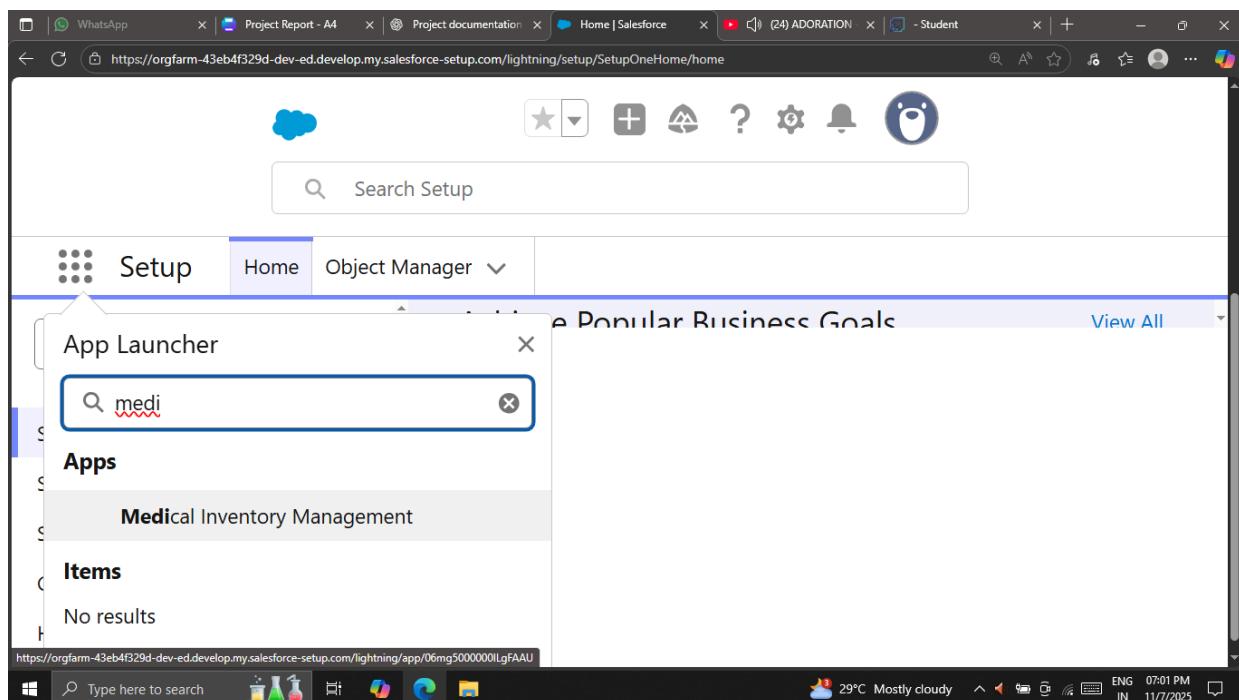
77     purchaseToUpdateMap.put(purchaseOrderId, totalAmount);
78 }
79
80
81
82
83     // Prepare Purchase Order records for update
84
85     List<Purchase_Order__c> purchaseToUpdate = new List<Purchase_Order__c>();
86
87     for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {
88
89         Purchase_Order__c purchaseOrder = new Purchase_Order__c(Id = purchaseOrderId, Total_Order_cost__c = purchaseToUpdateMap.get(purchaseOrderId));
90
91         purchaseToUpdate.add(purchaseOrder);
92

```

The screenshot shows the Salesforce Developer Console interface. The code editor displays an Apex trigger named "CalculateTotalAmountTrigger.apxc". The trigger logic involves a map to store purchase IDs and their total amounts, then iterates through the map to create new Purchase_Order__c records with the calculated total order cost. The developer console also includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems.

Apex Trigger and Handler implemented to calculate Total Order Cost.

12) Lightning App Creation

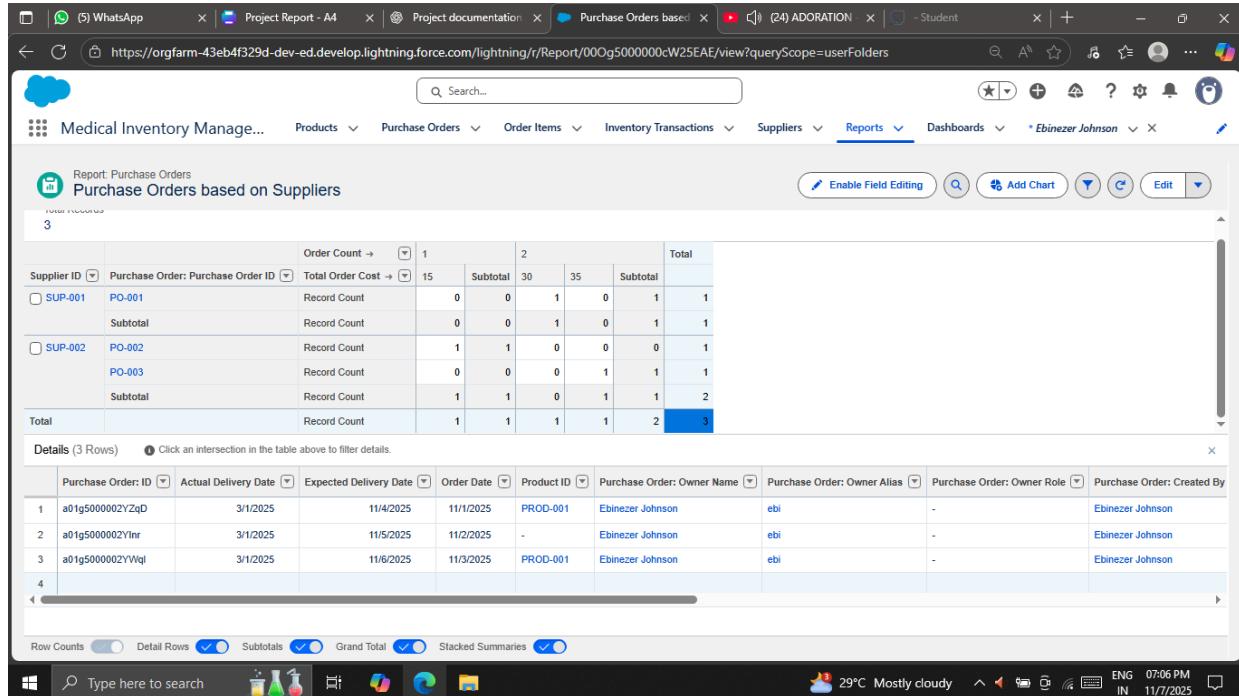


The screenshot shows the Salesforce Lightning App creation interface. The App Launcher search bar has "medi" typed into it. Under the "Apps" section, "Medical Inventory Management" is listed. Other sections like "Items" show "No results". The top navigation bar includes icons for Setup, Home, Object Manager, and other setup tools. The status bar at the bottom shows the URL https://orgfarm-43eb4f329d-dev-ed.develop.my.salesforce-setup.com/lightning/app/06mg5000000lgFAAU and system information like weather and time.

Lightning App created to manage all medical inventory modules.

APPENDIX – SCREENSHOTS

13) Reports



The screenshot shows a report titled "Purchase Orders based on Suppliers". The report displays a summary table and a detailed table below it.

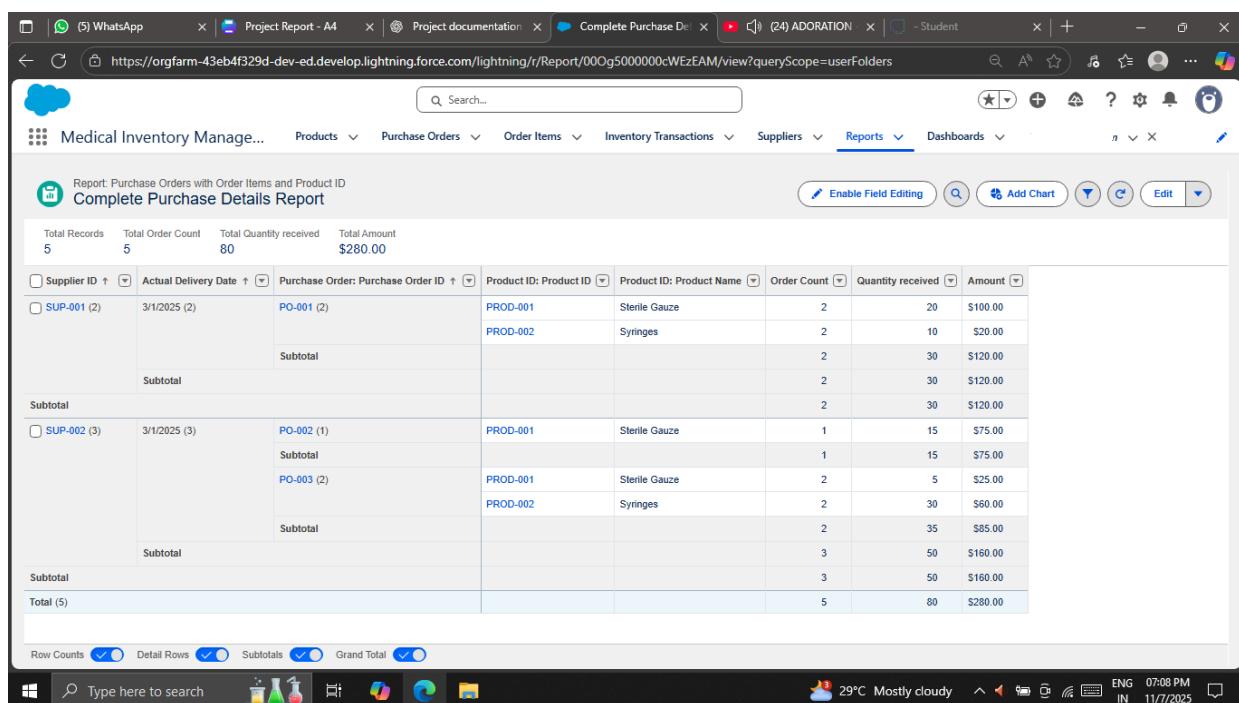
Summary Table:

Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost			Subtotal	Record Count	Total
			1	2	3			
SUP-001	PO-001	15	30	35	1	0	1	1
	Subtotal		0	1	0	1	1	1
SUP-002	PO-002		1	0	0	1	0	1
	PO-003		0	0	1	1	1	1
	Subtotal		1	1	0	1	1	2
	Total		1	1	1	2	3	3

Detailed Table:

Purchase Order: ID	Actual Delivery Date	Expected Delivery Date	Order Date	Product ID	Purchase Order: Owner Name	Purchase Order: Owner Alias	Purchase Order: Owner Role	Purchase Order: Created By
a01g5000002YzQD	3/1/2025	11/4/2025	11/1/2025	PROD-001	Ebinez Johnson	ebl	-	Ebinez Johnson
a01g5000002Ylnr	3/1/2025	11/5/2025	11/2/2025	-	Ebinez Johnson	ebl	-	Ebinez Johnson
a01g5000002YYwql	3/1/2025	11/6/2025	11/3/2025	PROD-001	Ebinez Johnson	ebl	-	Ebinez Johnson

Purchase Orders based on Suppliers



The screenshot shows a report titled "Complete Purchase Details Report". The report displays a summary table and a detailed table below it.

Summary Table:

Total Records	Total Order Count	Total Quantity received	Total Amount
5	5	80	\$280.00

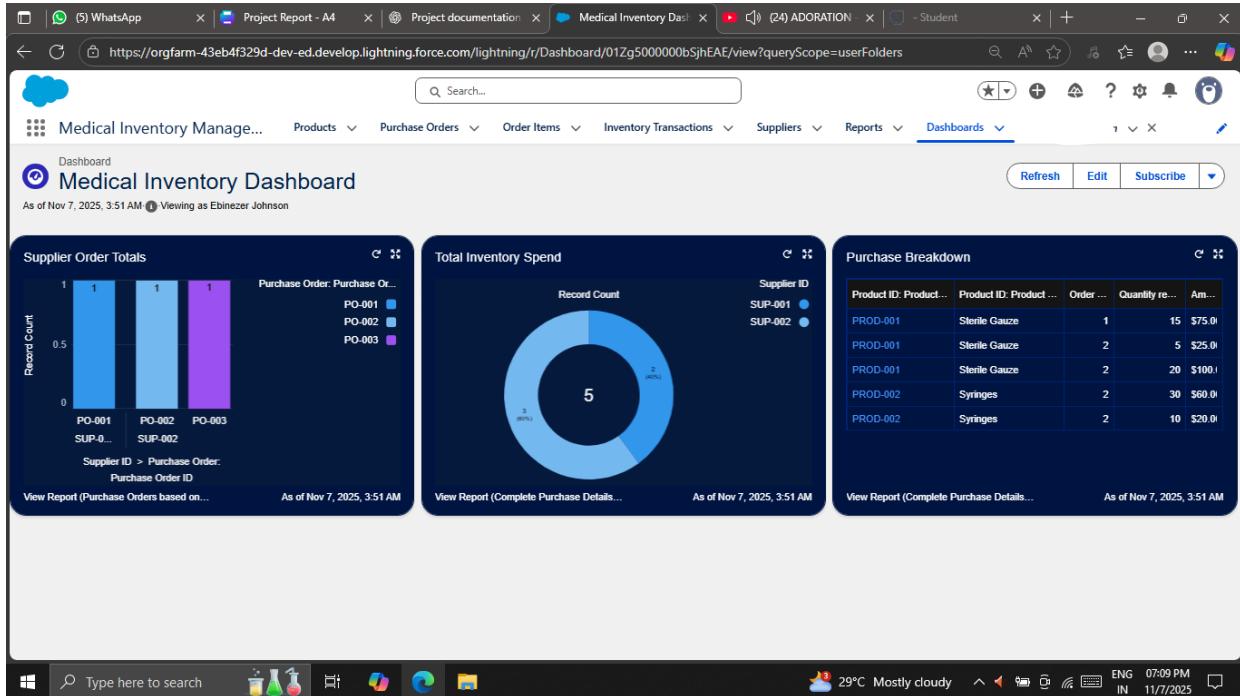
Detailed Table:

Supplier ID	Actual Delivery Date	Purchase Order: Purchase Order ID	Product ID: Product ID	Product ID: Product Name	Order Count	Quantity received	Amount
SUP-001 (2)	3/1/2025 (2)	PO-001 (2)	PROD-001	Sterile Gauze	2	20	\$100.00
			PROD-002	Syringes	2	10	\$20.00
					2	30	\$120.00
					2	30	\$120.00
					2	30	\$120.00
Subtotal							
SUP-002 (3)	3/1/2025 (3)	PO-002 (1)	PROD-001	Sterile Gauze	1	15	\$75.00
		Subtotal			1	15	\$75.00
		PO-003 (2)	PROD-001	Sterile Gauze	2	5	\$25.00
			PROD-002	Syringes	2	30	\$60.00
					2	35	\$85.00
					3	50	\$160.00
Subtotal					3	50	\$160.00
Total (5)					5	80	\$280.00

Complete Purchase Details Report

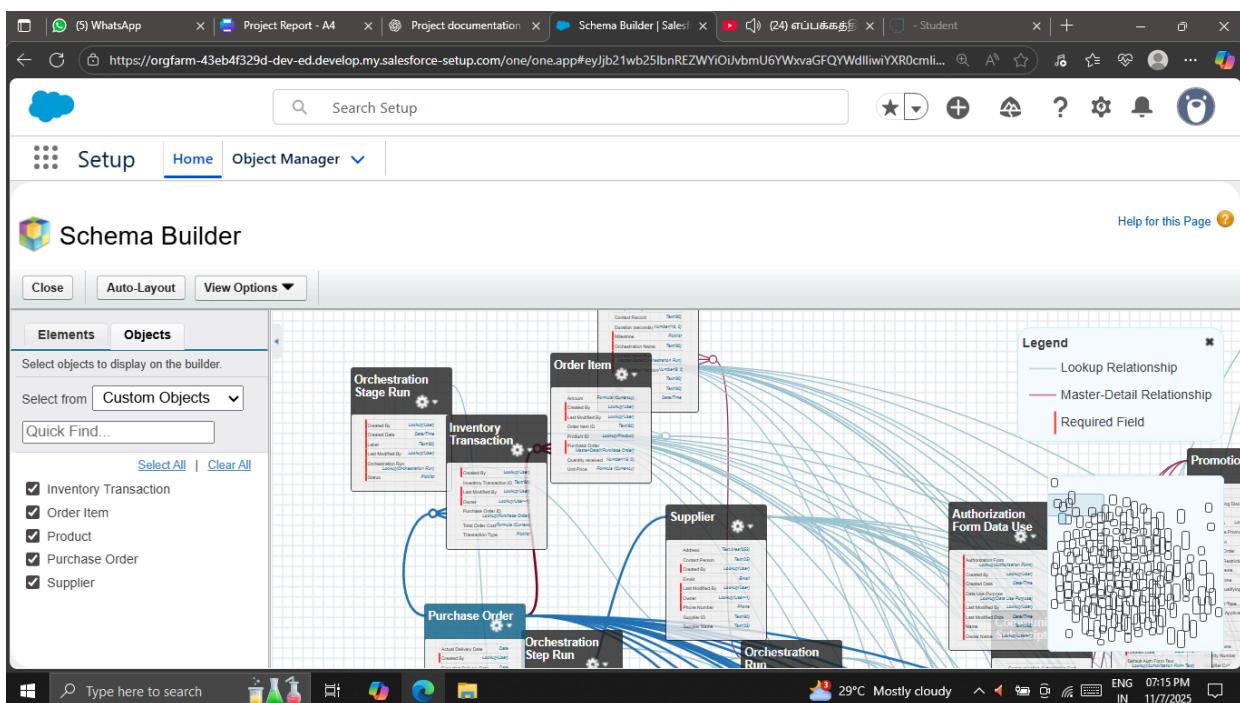
APPENDIX – SCREENSHOTS

14) Dashboard



Dashboard visualizing overall purchase performance and supplier data.

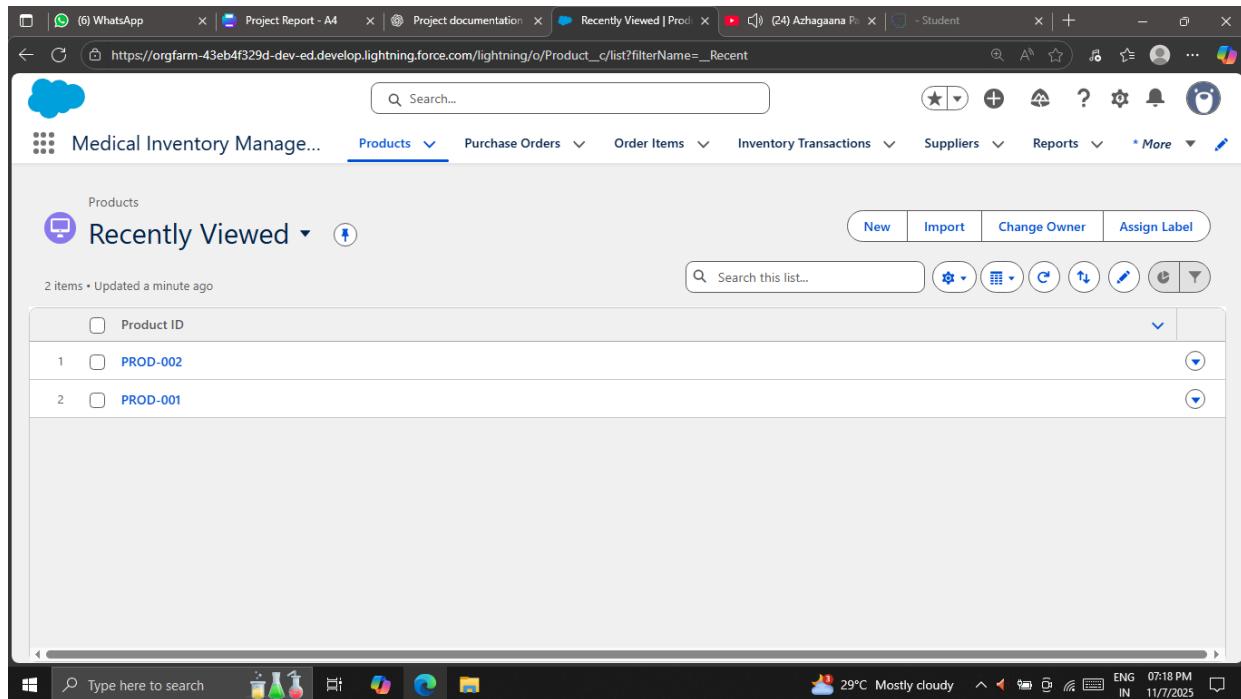
15) Schema Builder



Entity Relationship Diagram of Medical Inventory Management System.

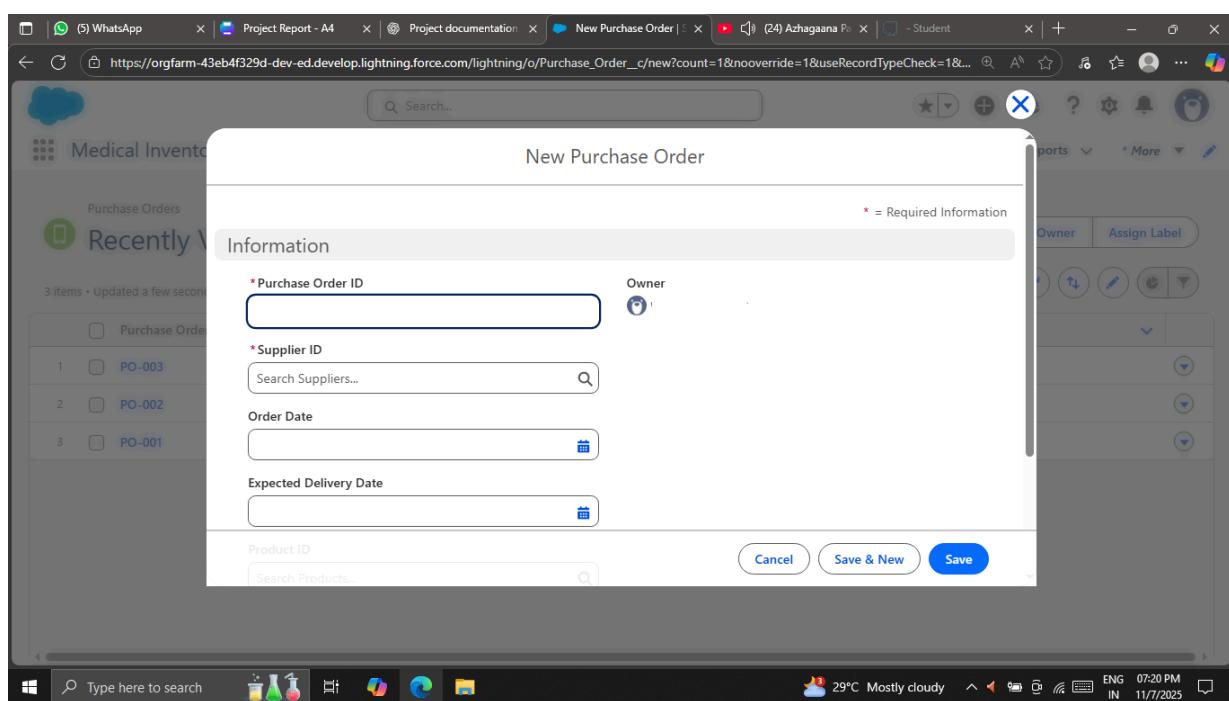
APPENDIX – SCREENSHOTS

16) Application Home Page (Final UI)



Medical Inventory Management Application Interface in Salesforce.

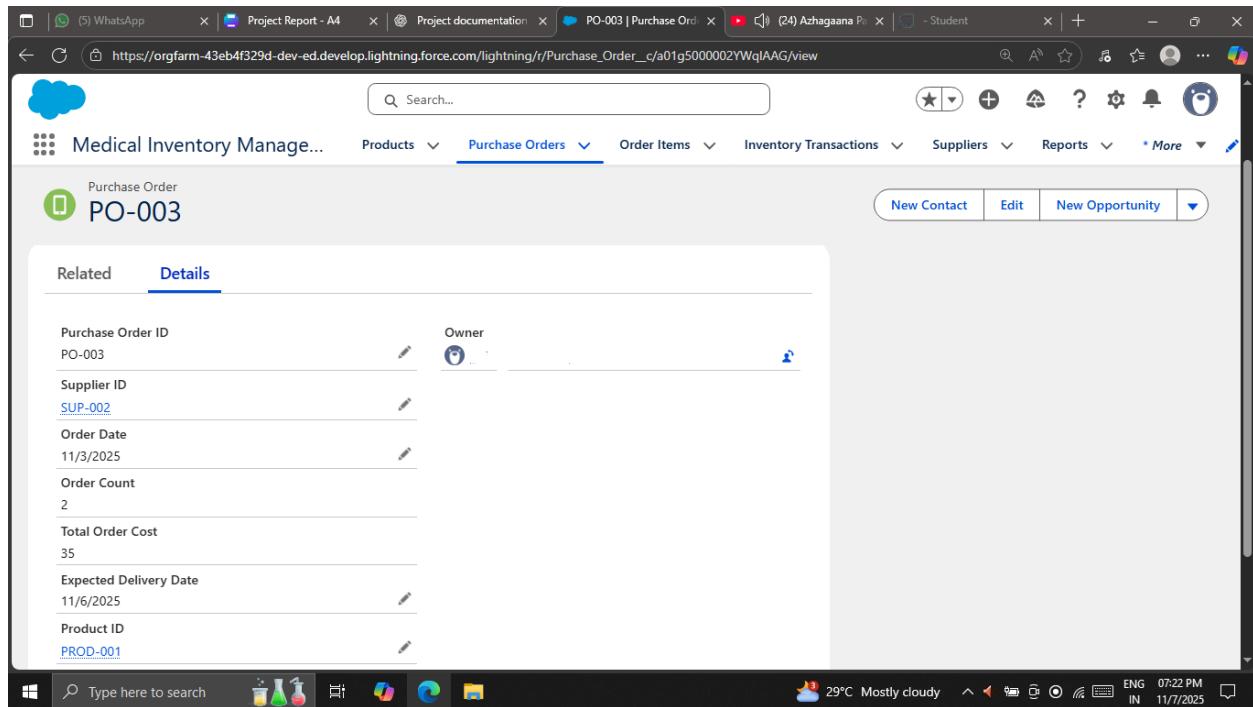
17) Record Creation Example



Sample record creation demonstrating system functionality.

APPENDIX – SCREENSHOTS

18) Sample Data Display



The screenshot shows a Salesforce Lightning interface for a Purchase Order. The top navigation bar includes links for WhatsApp, Project Report - A4, Project documentation, PO-003 | Purchase Ord, (24) Azhagaana Po, Student, and others. The main header displays "Medical Inventory Manage..." and the specific Purchase Order "PO-003". The "Purchase Orders" tab is currently selected. The "Details" section contains the following data:

Field	Value
Purchase Order ID	PO-003
Supplier ID	SUP-002
Order Date	11/3/2025
Order Count	2
Total Order Cost	35
Expected Delivery Date	11/6/2025
Product ID	PROD-001

Below the details, there are tabs for "Related" and "Purchase Orders". The bottom of the screen shows a Windows taskbar with a search bar, pinned icons for File Explorer, File History, Task View, and Edge, and system status indicators for weather (29°C, Mostly cloudy), battery (ENG IN 07:22 PM 11/7/2025), and connectivity.

Sample data added for testing Reports and Dashboards.

REFERENCES

- **Salesforce Trailhead Documentation —**
<https://trailhead.salesforce.com>
- **SmartInternz Naan Mudhalvan Portal —**
<https://nme.smartinternz.com>
- **Salesforce Developer Guide —**
<https://developer.salesforce.com>
- **GitHub Repository —** <https://github.com/Elakkia-19/Medical-Inventory-Management-System---SF>