

Medical Inventory Management

1. Ideation

The **Medical Inventory Management System** is an innovative Salesforce-based application conceptualized to streamline and manage the critical operations involved in medical inventory management. The idea originated from the growing need for **automation, accuracy, and efficiency** in handling medical supplies within hospitals, pharmacies, and healthcare institutions.

1.1 Problem Identification

The medical industry relies on timely access to essential medicines and equipment. However, traditional inventory management systems — often based on manual record-keeping or spreadsheets — are prone to errors such as **data duplication, stock mismatches, and delays in procurement**. These issues can lead to **overstocking, shortages, or accidental use of expired items**, ultimately affecting patient safety and operational productivity.

1.2 Proposed Solution

To overcome these challenges, the proposed system leverages **Salesforce's cloud-based platform** to create a centralized and automated solution. The **Medical Inventory Management System** enables users to:

- Maintain comprehensive **supplier profiles** with contact and product information.
- Create and manage **purchase orders** with detailed tracking of order items.
- Maintain **real-time product catalogs**, including stock levels and expiry dates.
- Generate **automated reports and dashboards** for performance insights and decision-making.
- Improve **accuracy, transparency, and data integrity** across all inventory operations.

1.3 Innovative Approach

The ideation behind the project lies in integrating **Salesforce CRM capabilities** with inventory management needs, combining **data automation, analytical reporting, and customizable workflows**. Unlike traditional methods, this application ensures **real-time visibility**, automated updates through **Apex triggers and Flows**, and visual analytics via **Reports and Dashboards** — empowering medical organizations to make smarter, faster, and safer decisions.

Medical Inventory Management

1.4 Expected Impact

This solution enhances:

- **Operational efficiency** through automation.
- **Data reliability** with centralized storage.
- **Timely decision-making** using analytics dashboards.
- **Regulatory compliance** by tracking expiry dates and purchase details.

2. Requirement Analysis

The **Medical Inventory Management System** is designed to automate and simplify medical inventory operations using Salesforce's robust CRM features. The requirement analysis phase identifies both **functional and non-functional requirements**, ensuring that the system meets business objectives efficiently while maintaining scalability, accuracy, and usability.

2.1 Functional Requirements

1. Supplier Management

- The system must allow creation, editing, and tracking of supplier records.
- Supplier details should include **Name, Contact Information, Address, and Product Categories** supplied.
- Each supplier record must be associated with **Purchase Orders** for transparency and traceability.

2. Product Management

- Maintain detailed product catalogs containing **Product Name, Category, Stock Quantity, and Expiry Date**.
- Allow linking products with suppliers and purchase orders.
- Enable alerts or visual indicators for products nearing expiry or low stock.

3. Purchase Order Management

- Create, update, and manage **Purchase Orders (POs)** linked to suppliers.
- Each PO should include **Order ID, Product Details, Quantity, Order Date, Expected Delivery Date, and Actual Delivery Date**.
- Implement validation rules to prevent incomplete or duplicate purchase orders.
- Provide automatic calculation of **Total Order Cost** using formula fields.

Medical Inventory Management

4. Inventory Transaction Tracking

- Record every transaction related to product movement (incoming and outgoing).
- Capture details such as **Transaction Date, Product Name, Quantity, and Type of Transaction** (Received / Issued).
- Maintain relationships between **Order Items, Products, and Suppliers** to ensure end-to-end traceability.

5. Reporting and Analytics

- Generate **custom reports** for supplier performance, stock status, and purchase order summary.
- Provide **dashboard views** showing visual insights into stock availability, order timelines, and supplier efficiency.
- Allow export of reports for management review and auditing purposes.

2.2 Non-Functional Requirements

1. Usability:

The application interface must be intuitive and easy to navigate with user-friendly page layouts.

2. Scalability:

The system design should support future enhancements such as automatic re-ordering or integration with hospital management systems.

3. Performance:

Queries and record operations must execute quickly, even with large datasets.

4. Security:

Role-based access should be enforced to control visibility and data modification privileges. Field-level security and profile settings must protect sensitive information.

5. Maintainability:

The schema and automation design should be modular, allowing easy updates and debugging.

2.3 Dependencies and System Constraints

- Relies on **Salesforce standard and custom objects** for implementation (Supplier, Product, Purchase Order, Order Item, Inventory Transaction).
- Requires appropriate **user permissions** and **page layouts** for data access.
- Automation depends on **Flows, Validation Rules, and Formula Fields**.

Medical Inventory Management

- Report generation depends on **Salesforce Reporting & Dashboard Tools**.

2.4 Alignment with Business Goals

The proposed system directly aligns with the operational goals of healthcare organizations by:

- Reducing manual effort and data errors in inventory tracking.
- Ensuring compliance through expiry-date monitoring.
- Enhancing procurement decision-making using real-time data insights.
- Strengthening supplier relationships through performance analysis.

2.5 Outcome of Requirement Analysis

After thorough analysis, all business and technical requirements were clearly defined, dependencies identified, and object relationships mapped. This phase ensured the foundation for **efficient design, automation, and implementation** within Salesforce while maintaining scalability and reliability.

3. Project Design

The **Medical Inventory Management System** is thoughtfully designed in Salesforce to offer a centralized, automated, and visually intuitive platform for managing all medical inventory activities. The design ensures a seamless blend of backend data automation and a user-friendly Lightning interface that enhances usability, scalability, and data accuracy.

3.1 Object and Relationship Design

The system architecture revolves around five key custom objects:

- **Supplier:** Stores supplier details such as name, contact, and company information.
- **Product:** Contains product details, including name, description, quantity, and expiry date.
- **Purchase Order:** Records supplier orders and their statuses.
- **Order Item:** Tracks individual products ordered under a purchase order.
- **Inventory Transaction:** Monitors product movement and updates stock levels.

Relationships are established through **lookup** and **master-detail** links to ensure data consistency. For example, each *Order Item* is linked to both *Purchase Order* and *Product*, enabling accurate cost and stock tracking. The **Schema Builder** was used to visualize these relationships for clarity and validation.

Medical Inventory Management

3.2 Automation and Logic

To reduce manual effort and enhance data accuracy, **Salesforce Flows** and **Validation Rules** were implemented:

- **Record-Triggered Flows** automatically update the *Actual Delivery Date* based on the *Order Date* and predefined conditions.
- **Validation Rules** prevent incorrect data entries (e.g., ensuring expiry date is after the manufacturing date).
- **Formula Fields** calculate *Total Order Cost* and dynamically display real-time inventory status. This ensures that all automation is admin-friendly, maintainable, and compliant with Salesforce best practices.

3.3 Reports and Dashboards

Comprehensive **Reports and Dashboards** were developed to provide real-time insights:

- **Reports** such as *Purchase Orders by Supplier* and *Complete Purchase Details* summarize key operational data.
- The **Medical Inventory Dashboard** consolidates these reports into interactive visualizations like bar charts, pie charts, and tables, making it easier to analyze supplier performance, purchase trends, and inventory levels. This enhances decision-making and transparency within the organization.

3.4 User Interface Design

The system uses **Lightning App Builder**, **Dynamic Forms**, and **Role-based Visibility** to create a responsive and intuitive interface.

- Each page layout is cleanly structured, grouping relevant fields and actions for easy access.
- **Dynamic Pages** ensure that users only see the sections relevant to their roles (e.g., Managers vs. Staff).
- Navigation is simplified with custom tabs and the **Medical Inventory App**, ensuring a consistent and efficient user experience.

3.5 Innovation and Scalability

The design integrates multiple Salesforce features innovatively to deliver a robust, future-ready system.

- Combines **Flows, Reports, and Dashboards** for complete automation and visualization.

Medical Inventory Management

- Enhances usability with **Dynamic Forms**, **Lightning Pages**, and **automated notifications**.
- Scalable to include future enhancements such as **AI-based demand forecasting**, **automated reorder levels**, or **integration with hospital management systems**.

4. Project Development

The **Medical Inventory Management System** was developed on the Salesforce platform using a combination of **declarative tools** (click-based automation) and **programmatic components** (Apex triggers, classes, and validation logic). The development process followed a systematic and modular approach to ensure efficiency, maintainability, and scalability.

4.1 Development Environment

- The project was built in **Salesforce Developer Edition**, which provided a secure and flexible environment to create custom objects, fields, and automation.
- The **Lightning App Builder**, **Object Manager**, **Schema Builder**, and **Developer Console** were the primary tools used for configuration and coding.
- A dedicated **Medical Inventory Management App** was created to organize all related tabs and features under one workspace, ensuring smooth navigation for users.

4.2 Object Creation and Configuration

- Custom objects such as **Supplier**, **Product**, **Purchase Order**, **Order Item**, and **Inventory Transaction** were created to model real-world medical inventory processes.
- Each object included relevant custom fields (e.g., *Actual Delivery Date*, *Total Order Cost*, *Quantity Received*) and **lookup relationships** to establish data connectivity.
- **Page layouts**, **record types**, and **field-level security** were configured to ensure that users can view or edit only relevant data according to their roles.

4.3 Automation and Logic Implementation

Automation played a key role in improving operational accuracy and reducing manual workload:

- **Apex Trigger:** A trigger named *CalculateTotalAmountTrigger* was developed to automatically compute and update the total cost in *Purchase Orders* whenever related *Order Items* are inserted, updated, or deleted.

Medical Inventory Management

- **Handler Class:** The `CalculateTotalAmountHandler` Apex class was used to separate business logic from trigger code, improving reusability and maintainability.
- **Flows and Validation Rules:**
 - Record-triggered flows were designed to auto-populate dates and statuses.
 - Validation rules ensured that incorrect or incomplete data (like expired dates or missing supplier references) could not be saved.
- **Formula Fields:** Used for dynamic calculations such as total order cost and remaining stock.

This structured approach ensured clean, efficient code following **Salesforce Apex Best Practices**, including error handling, and separation of concerns.

4.4 Reports and Dashboards Development

- Two summary reports were developed:
 1. **Purchase Orders Based on Suppliers Report** – summarizes supplier performance and order statistics.
 2. **Complete Purchase Details Report** – provides a comprehensive view of all purchase transactions, product details, and costs.
- The **Medical Inventory Dashboard** integrates these reports using widgets and charts to visually represent real-time data such as supplier order count, stock trends, and total purchase costs.

These tools allow management to analyze data efficiently and make informed decisions without manual data compilation.

4.5 Testing and Validation

- **Unit Testing:** Conducted for Apex triggers and classes to ensure accuracy and prevent runtime errors.
- **Functional Testing:** Verified that each automation (flows, formulas, validation rules) worked as intended under multiple use cases.
- **User Acceptance Testing (UAT):** Simulated real-world scenarios to validate usability and confirm that users (like Purchase Managers) could perform end-to-end operations smoothly.
- **Error Handling:** Proper exception handling in code ensured that errors are logged and manageable without disrupting processes.

Medical Inventory Management

4.6 Implementation Best Practices

The development followed Salesforce's **Low-Code + Pro-Code** philosophy to balance automation and customization effectively:

- **Reusability:** Handler classes, formulas, and flows were designed for modular use.
- **Scalability:** The structure allows easy addition of new modules (e.g., product return tracking or supplier evaluation).
- **Security:** Applied proper field-level and object-level permissions through profiles and permission sets.
- **Documentation:** Each customization (trigger, flow, field) was documented for future reference and maintenance.

5. Project Documentation

5.1 : Creation of developer account

5.1.1 Salesforce Developer Account Creation

Objective: Create a Salesforce Developer account to access a personal development environment.

Description: This account allows users to explore and practice Salesforce features without affecting live data.

Steps:

1. Go to [Salesforce Developer Signup](#).
2. Fill in the form:
 - First & Last Name
 - Email
 - Role: Developer
 - Company: College Name
 - Country: India
 - Postal Code
 - Username: name@organization.com
3. Click Sign Me Up.

Medical Inventory Management

Outcome: Developer account request submitted.

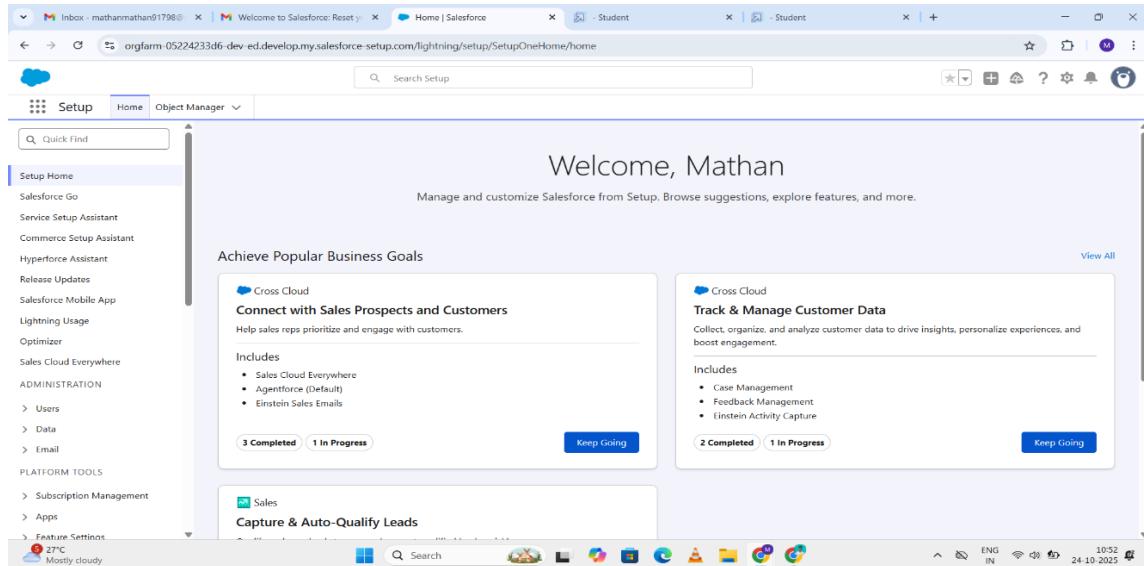
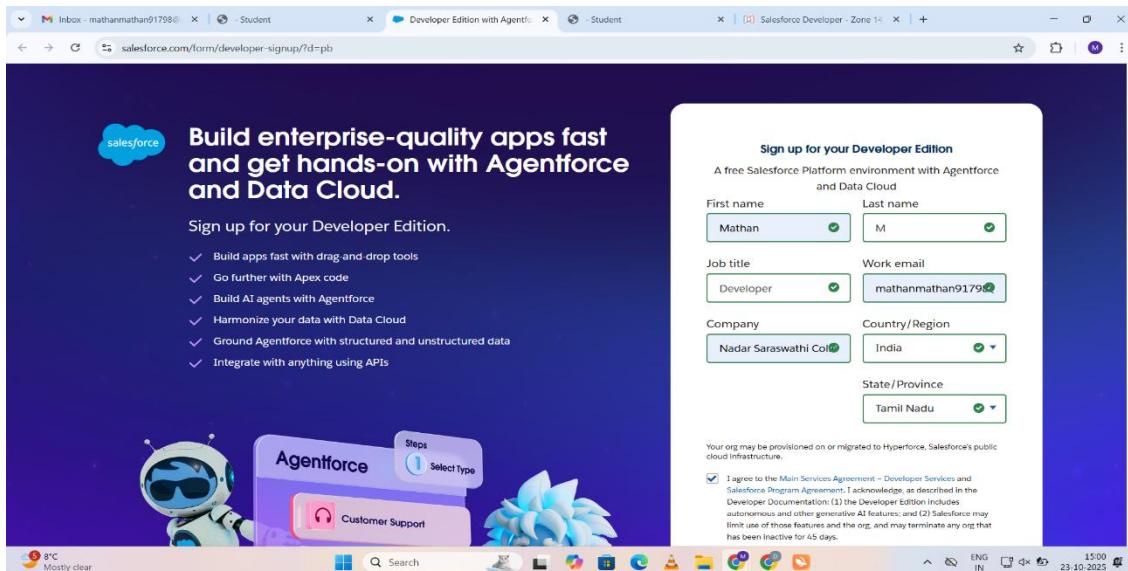
5.1.2 Salesforce Account Activation

Objective: Activate the Salesforce Developer account.

Steps:

1. Open the signup email and click Verify Account (may take 5–10 mins).
2. Set a password and security question.
3. Click Change Password and access the Salesforce setup page.

Outcome: Account activated and ready for use.



Medical Inventory Management

5.2: Objects

In Salesforce, objects are database tables that allow you to store data specific to your organization.

5.2.1 Creating a Product Object

Objective: Create custom objects in Salesforce to manage project data.

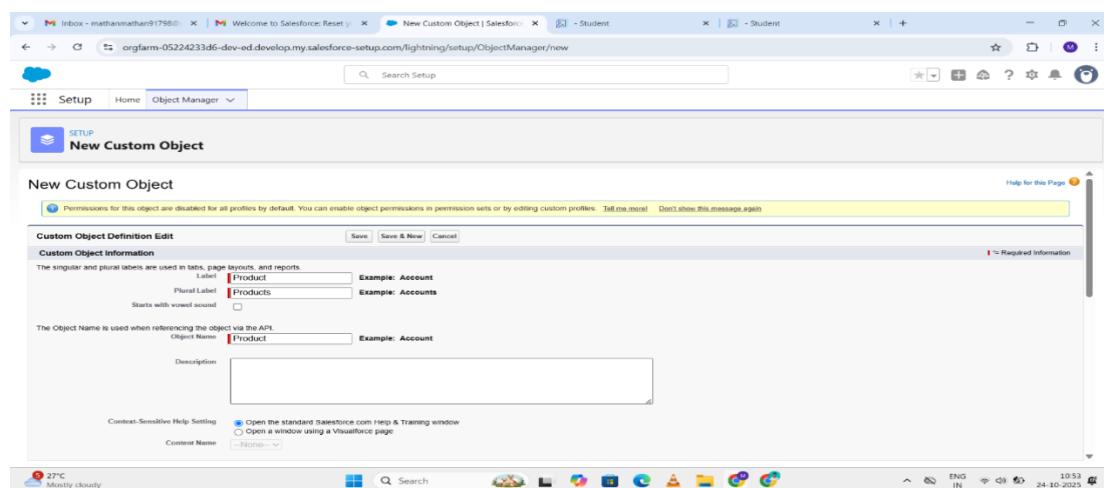
Description: Custom objects like Product, Purchase Order, and Supplier help structure project data for reporting, tracking, and inventory management.

Steps:

1. Go to Setup → Object Manager.
2. Click Create → Custom Object.
3. Enter the following details for the Product object:
 - **Label Name:** Product
 - **Plural Label:** Products
 - **Record Name:** Product ID
 - **Data Type:** Text
 - **Options:** Allow reports, Allow search
4. Click Save & New to create additional objects.

5. Similarly, create the objects: Purchase Order, Order Item, Inventory Transaction, Supplier.

Outcome: All required custom objects are created and ready for use in the project.



Medical Inventory Management

5.3: Tabs

In Salesforce, tabs are used to make the data stored in objects accessible to users through the user interface. Tabs are a fundamental part of the Salesforce interface, providing a way to navigate to different objects and records.

5.3.1 Creating a Tab for Product Object

Objective: Create a custom tab to access the Product object easily from Salesforce UI.

Steps:

1. Click New under Custom Object Tabs.
2. Select Object: Product, choose a Tab Style.
3. Click Next → Add to profiles (keep default) → Click Next → Add to Custom App (uncheck “Include Tab”).
4. Ensure Append tab to user’s existing personal customizations is checked.
5. Click Save.

Outcome: Product object tab is created and accessible from the Salesforce interface.

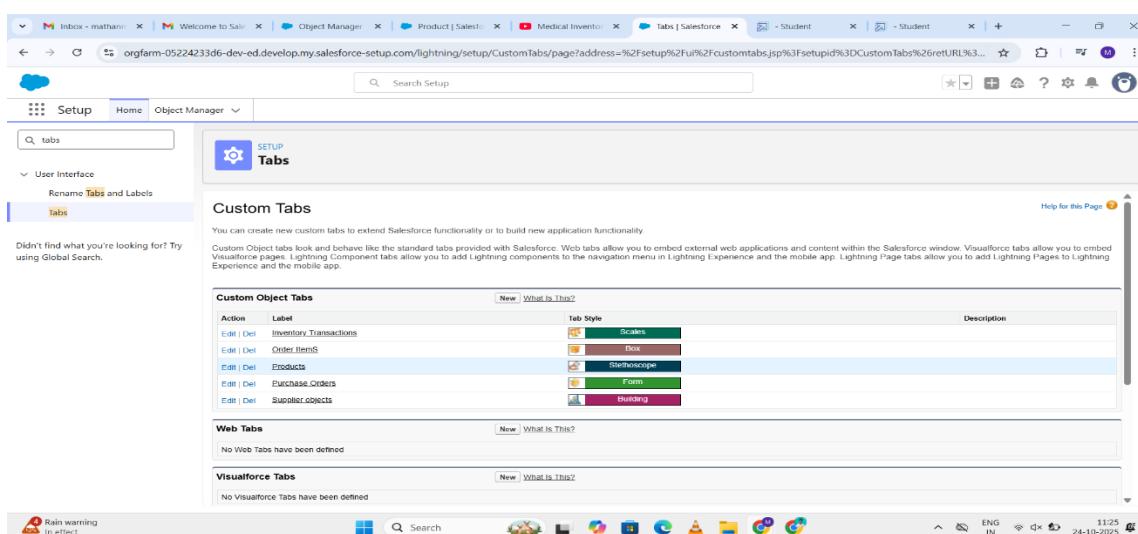
5.3.2 Creating Remaining Tabs

Objective: Create tabs for other custom objects to enable easy access.

Steps:

- Similarly, create tabs for Purchase Order, Order Item, Inventory Transaction, Supplier following the same procedure.

Outcome: All required custom object tabs are created and ready for use.



Medical Inventory Management

5.4:The Lightning App

A Lightning App in Salesforce is a collection of items that work together to serve a particular function for the end-users. These items can include standard and custom objects, tabs, utilities, and other productivity tools. Lightning Apps are designed to provide a more intuitive and efficient user experience compared to traditional Salesforce apps.

5.4.1 Creating a Lightning App for Medical Inventory Management

Objective: Create a custom Lightning App to manage all medical inventory-related objects in one place.

The Lightning App is designed to provide a seamless and intuitive experience for users managing the medical inventory. By consolidating all relevant objects, reports, and dashboards within one app, users can efficiently access, update, and analyze critical data without navigating through multiple tabs or applications.

The app also integrates automation features, such as **flows and triggers**, and enables access control through **profiles and roles**, ensuring that users have the right permissions to perform their tasks securely. This centralized approach improves productivity, reduces errors, and allows for better monitoring of inventory levels, supplier performance, and purchase order status.

Ultimately, the custom Lightning App serves as a **comprehensive platform** for managing the medical inventory, supporting data-driven decision-making and enhancing operational efficiency across the organization.

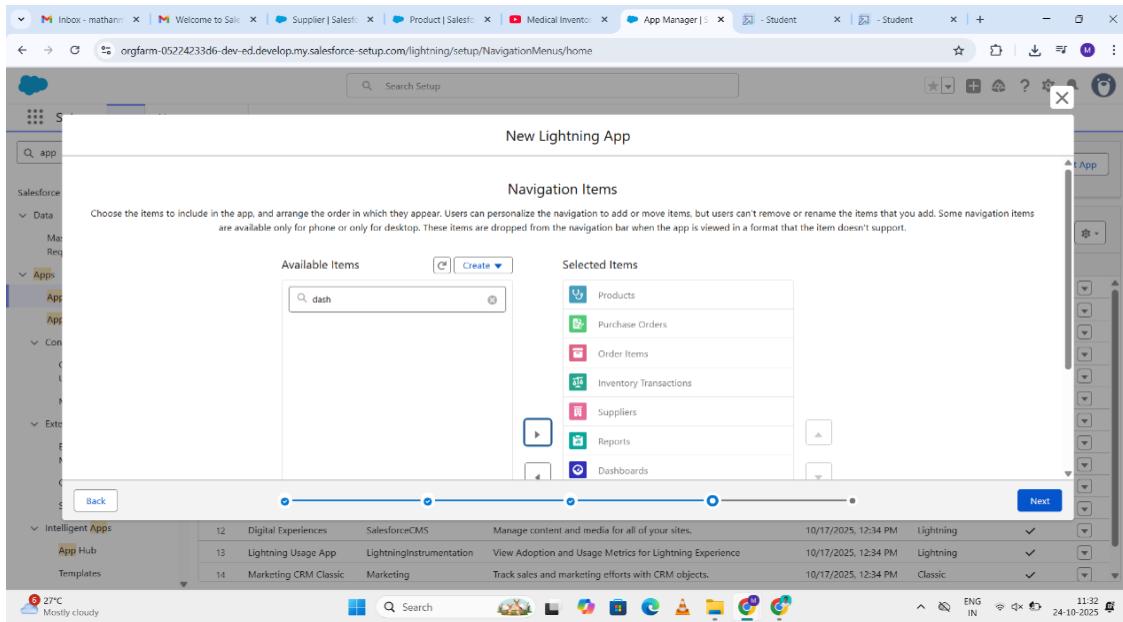
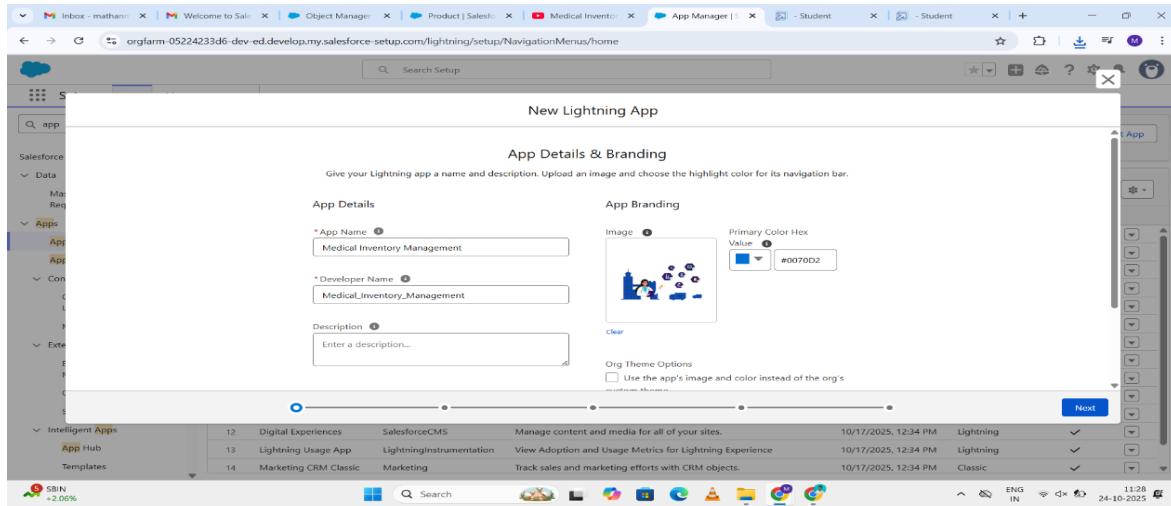
Steps:

1. From Setup, search and open App Manager.
2. Click New Lightning App.
3. Enter App Name: *Medical Inventory Management*.
4. Upload an image related to medical inventory and click Next.
5. Keep the default selections under App Options and click Next.
6. Leave Utility Items as is and click Next.
7. From Available Items, select *Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, Reports, and Dashboards*, then move them to Selected Items and click Next.
8. From Available Profiles, select *System Administrator* and move it to Selected Profiles.

Medical Inventory Management

9. Click Save & Finish.

Outcome: The *Medical Inventory Management* Lightning App is successfully created and ready for use.



5.5:Fields

Create custom fields in Salesforce objects to store specific data related to the medical inventory system. Each object such as Product, Purchase Order, Order Item, Inventory Transaction, and Supplier is customized by adding relevant fields like Quantity, Unit Price, and Total Cost. The created fields help capture and manage detailed business data efficiently within Salesforce.

Medical Inventory Management

5.5.1 Creating a Text Field in Product Object

Create a text field to store product names.

Steps: Select Product from Object Manager → Fields & Relationships → New → Choose Text → Enter *Field Label*: *Product Name*, Length 255 → Mark as Required → Click Next, Next, Save & New.

Outcome: The Product Name field is created successfully.

5.5.2 Creating a Text Area Field in Product Object

Create a text area field to store detailed product descriptions.

Steps: In Product → Fields & Relationships → New → Select Text Area → Enter *Field Label*: *Product Description* → Click Next, Next, Save & New.

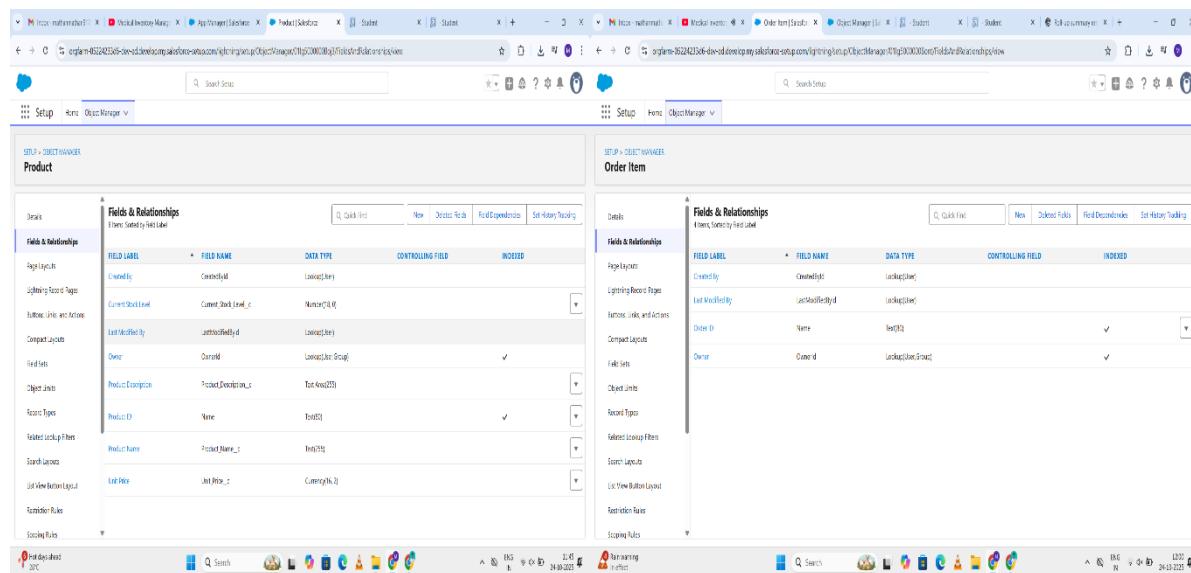
Outcome: The Product Description field is created successfully.

5.5.3 Creating a Number Field in Product Object

Create a number field to record the available stock quantity.

Steps: In Product → Fields & Relationships → New → Select Number → Enter *Field Label*: *Current Stock Level*, Length 18, Decimal 0 → Click Next, Next, Save.

Outcome: The Current Stock Level field is created successfully.



5.5.4 Creating a Currency Field in Product Object

Create a currency field to store the unit price of each product.

Steps: In Product → Fields & Relationships → New → Select Currency → Enter *Field*

Medical Inventory Management

Label: Unit Price, Length 16, Decimal 2 → Mark as Required → Click Next, Next, Save.

Outcome: The Unit Price field is created successfully.

5.5.5 Creating a Lookup Relationship in Purchase Order Object

A Lookup relationship connects related objects in Salesforce.

Steps: In Purchase Order → Fields & Relationships → New → Select Lookup Relationship → Choose related object Supplier → Enter Field Label: Supplier ID → Mark as Required → Click Next, Next, Next, Save.

Outcome: A lookup relationship between Purchase Order and Supplier is created successfully.

5.5.6 Creating a Date Field in Purchase Order Object

Create a date field to record when the order was placed.

Steps: In Purchase Order → Fields & Relationships → New → Select Date → Enter Field

Label: Order Date → Click Next, Next, Save.

Outcome: The Order Date field is created successfully.

5.5.7 Creating a Roll-Up Summary Field in Purchase Order Object

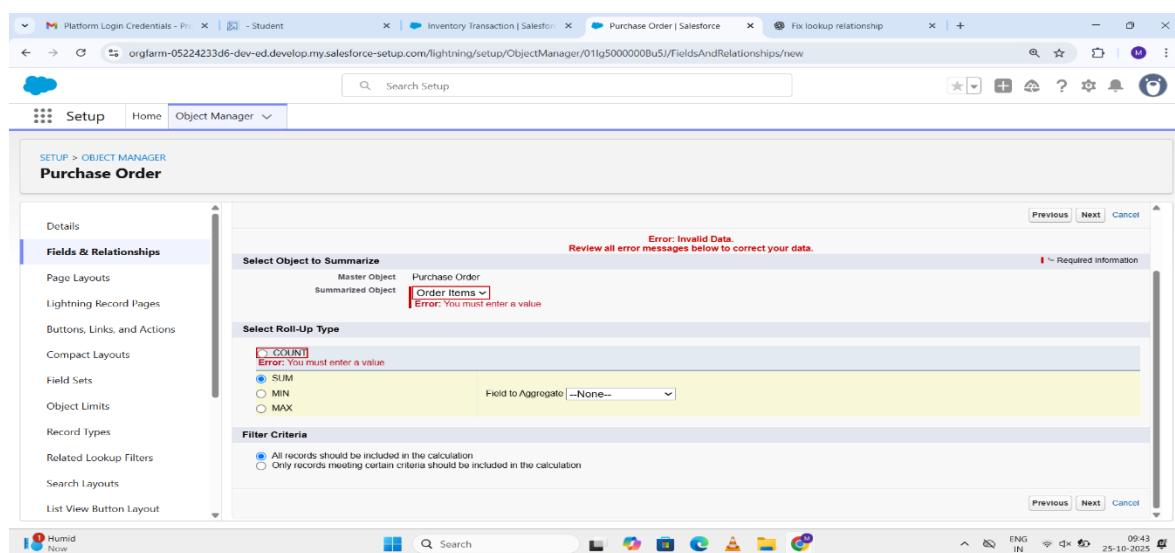
Create a roll-up summary field to count related order items.

Steps: In Purchase Order → Fields & Relationships → New → Select Roll-Up Summary →

Enter Field Label: Order Count → Choose Summarized Object: Order Items → Roll-Up

Type: Count → Click Next, Next, Save.

Outcome: The Order Count roll-up summary field is created successfully.



Medical Inventory Management

5.5.8 Creating a Unit Price Formula Field in Order Item Object

Create a formula field to fetch the product's unit price automatically.

Steps: In Order Item → Fields & Relationships → New → Select Formula → *Field Label: Unit Price* → Return Type *Currency* → Formula: *Product_ID__r.Unit_Price__c* → Click Next, Next, Save.

Outcome: The Unit Price formula field is created successfully.

5.5.9 Creating an Amount Formula Field in Order Item Object

Create a formula field to calculate the total amount of items received.

Steps: In Order Item → Fields & Relationships → New → Select Formula → *Field Label: Amount* → Return Type *Currency* → Formula: *Quantity_Received__c * Unit_Price__c* → Click Next, Next, Save.

Outcome: The Amount formula field is created successfully.

5.5.10 Creating a Picklist Field in Inventory Transaction Object

Create a picklist field to classify transaction types.

Steps: In Inventory Transaction → Fields & Relationships → New → Select Picklist → *Field Label: Transaction Type* → Enter values:

Receipt

Issue

Adjustment

Click Next, Next, Save.

Outcome: The Transaction Type picklist field is created successfully.

The screenshot shows the Salesforce Setup interface under the Object Manager. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Inventory Transaction' and shows the 'Fields & Relationships' section. A table lists fields such as Created By, Inventory Transaction ID, Last Modified By, Owner, Purchase Order ID, Total Order Cost, and Transaction Type. The 'Transaction Type' field is highlighted as a new picklist field. The top navigation bar includes tabs for Home, Object Manager, and a search bar.

Medical Inventory Management

5.5.11 Creating a Total Order Cost Formula Field in Inventory Transaction Object

Create a formula field to retrieve the total order cost from the Purchase Order object.

Steps: In Inventory Transaction → Fields & Relationships → New → Select Formula → *Field*

Label: *Total Order Cost* → Return Type *Currency* → Formula:

Purchase_Order_ID__r.Total_Order_Cost__c → Click Next, Next, Save.

Outcome: The Total Order Cost formula field is created successfully.

5.5.12 Creating a Phone Field in Supplier Object

Create a phone field to store supplier contact numbers.

Steps: In Supplier → Fields & Relationships → New → Select Phone → *Field Label: Phone Number* → Mark as Required → Click Next, Next, Save.

Outcome: The Phone Number field is created successfully.

5.5.13 Creating an Email Field in Supplier Object

Create an email field to store supplier email addresses.

Steps: In Supplier → Fields & Relationships → New → Select Email → *Field Label: Email* → Click Next, Next, Save.

Outcome: The Email field is created successfully.

5.6:Editing of Page Layouts

Page layouts in Salesforce are used to customize the organization, structure, and content of pages for viewing and editing records. They determine which fields, related lists, and custom links are visible to users, as well as the order and grouping of those elements.

5.6.1 To Edit a Page Layout in Product Object

Go to Setup → click on Object Manager → type Product in the Quick Find box → click on the Product object → Page Layouts.

Click on Product Layout.

Drag and arrange the fields as needed, then click Save.

5.6.2 To Edit a Page Layout in Purchase Order Object

Go to Setup → click on Object Manager → type Purchase Order in the Quick Find box → click on the Purchase Order object → Page Layouts.

Click on Purchase Order Layout.

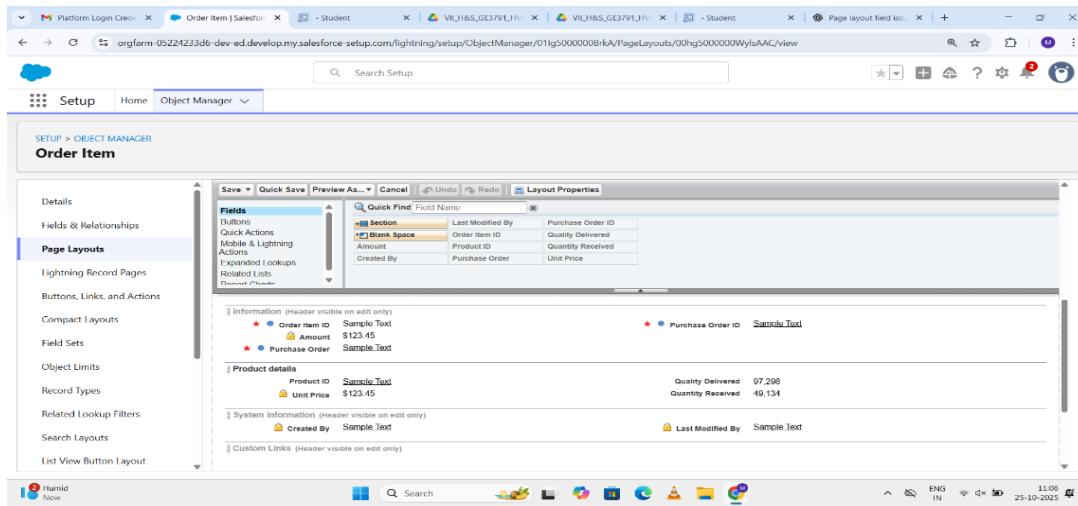
Drag and arrange the fields as required.

Click on the Order Date field → click on Settings → select Required and save it.

Click on the Total Order Cost field → click on Settings → select Read Only and save it.

Finally, click Save.

Medical Inventory Management



5.6.3 To Edit a Page Layout in Order Item Object

Go to Setup → click on Object Manager → type Order Item in the Quick Find box → click on the Order Item object → Page Layouts.

Click on Order Item Layout.

Drag and arrange the fields as required, then click Save.

5.6.4 To Edit a Page Layout in Inventory Transaction Object

Go to Setup → click on Object Manager → type Inventory Transaction in the Quick Find box → click on the Inventory Transaction object → Page Layouts.

Click on Inventory Transaction Layout.

Drag and arrange the fields as needed, then click Save.

5.6.5 Edit Page Layout in Supplier Object

Objective:

To customize the Supplier object page layout to display and arrange fields in a user-friendly and logical manner for efficient data entry and management.

Description:

Page layouts in Salesforce control the organization and visibility of fields, related lists, and buttons on a record page. Editing the Supplier page layout ensures that users can quickly access and update supplier information without confusion, improving usability and workflow efficiency.

Steps:

1. Go to Setup → Object Manager → Supplier → Page Layouts.
2. Click on Supplier Layout.

Medical Inventory Management

3. Drag and arrange the fields as required to suit business needs.
4. Click Save.

Outcome:

The Supplier page layout is customized to display essential fields clearly, enabling users to efficiently manage supplier data and maintain consistency across records.

This screenshot shows the Salesforce Object Manager interface for customizing the 'Inventory Transaction' page layout. The left sidebar lists various configuration options like Details, Fields & Relationships, and Page Layouts. The 'Page Layouts' option is selected. The main area displays the current page layout structure. It includes sections for 'Fields' (with a 'Quick Find' search bar), 'Information' (containing fields like Transaction ID, Purchase Order ID, Transaction Type, and Total Order Cost), and 'System Information' (containing Created By and Last Modified By). A 'Layout Properties' tab is visible at the top right of the layout editor.

This screenshot shows the Salesforce Object Manager interface for customizing the 'Supplier' page layout. Similar to the previous screenshot, it shows the 'Page Layouts' section selected in the sidebar. The main layout editor shows a different structure with sections for 'Fields' (including Section, Blank Space, and Contact Person), 'Supplier Detail' (containing fields like Supplier ID, Supplier Name, and Contact Person), and 'System Information'. The 'Layout Properties' tab is also present at the top right.

Medical Inventory Management

5.7: Compact Layouts

Compact layouts display a record's key fields at a glance, providing important information quickly without needing to open the record.

5.7.1 Product Object Compact Layout

Objective:

To create a compact layout for the Product object that highlights essential information, making it easier for users to quickly view key details on the record page.

Description:

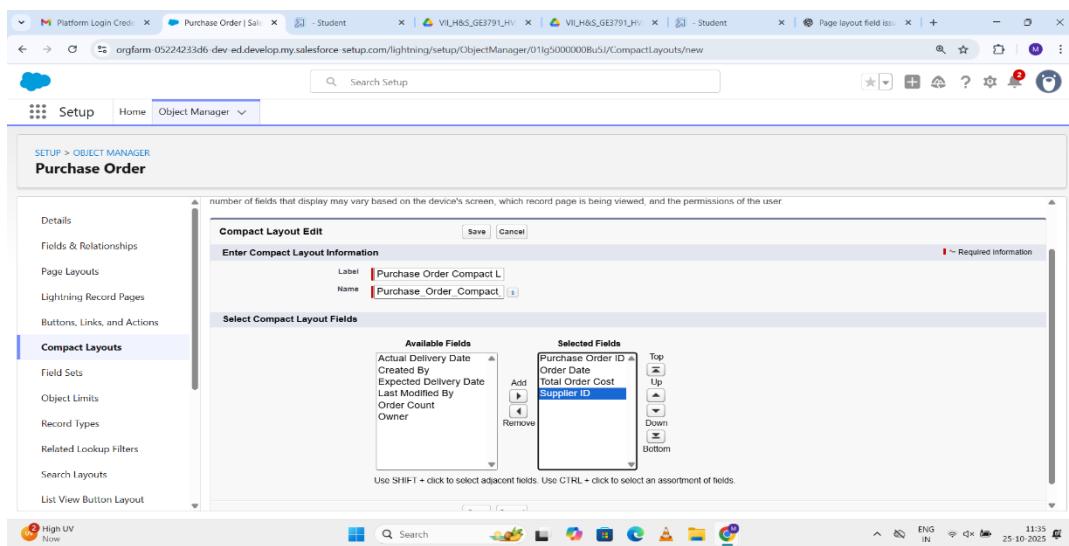
Compact layouts determine which fields appear in the record highlights section at the top of a Salesforce record. For products, displaying critical fields like Product Name, Unit Price, and Current Stock Level allows users to quickly access the most important data without scrolling through the entire record.

Steps:

1. Create a compact layout named “Product Compact Layout”.
2. Select the fields: Product Name, Unit Price, Current Stock Level.
3. Assign this compact layout as the primary layout for the Product object.

Outcome:

Users can quickly view key product details in the record highlights section, improving efficiency and data visibility.



Medical Inventory Management

5.7.2 Purchase Order Object Compact Layout

Objective:

To create a compact layout for the Purchase Order object that summarizes essential order information at a glance.

Description:

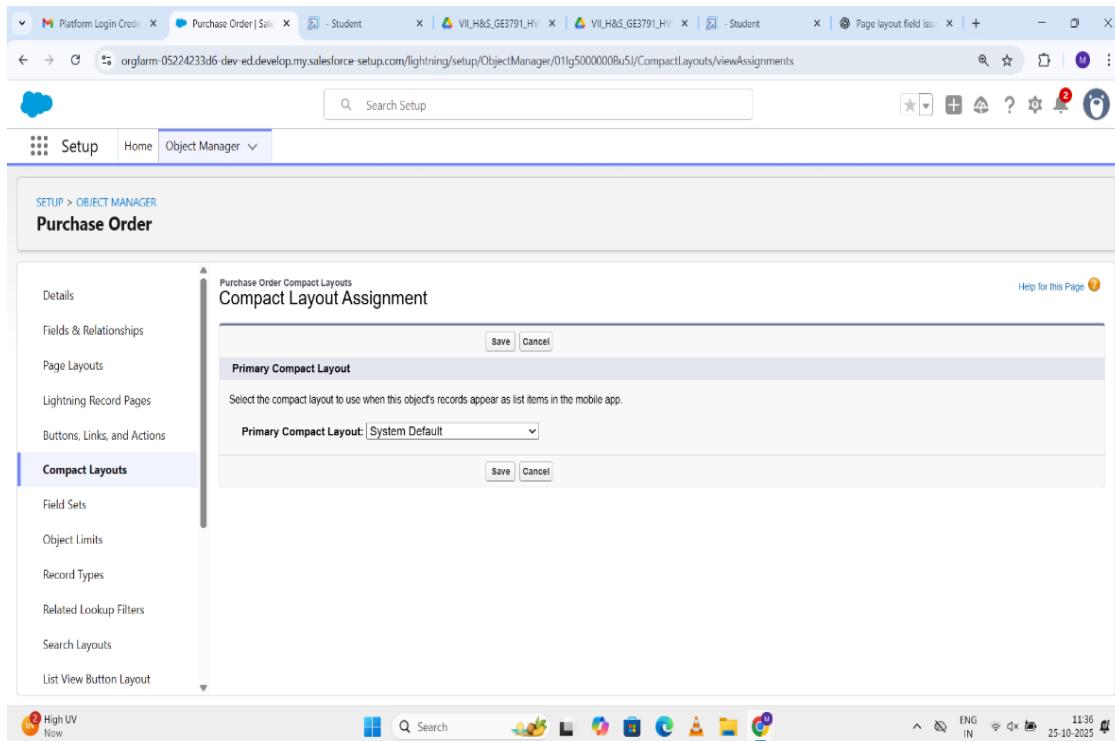
For purchase orders, the compact layout displays fields such as Purchase Order ID, Order Date, Total Order Cost, and Supplier ID. This allows users to immediately identify the order details without navigating through the full record.

Steps:

1. Create a compact layout named “Purchase Order Compact Layout”.
2. Select the fields: Purchase Order ID, Order Date, Total Order Cost, Supplier ID.
3. Assign this layout as the primary layout for the Purchase Order object.

Outcome:

Users can efficiently review key purchase order information, enhancing productivity and decision-making.



Medical Inventory Management

5.8: Validation Rules

Validation rules in Salesforce are used to ensure data integrity by preventing users from saving invalid data in records. They consist of a formula or expression that evaluates the data in one or more fields and return a value of true or false. When the rule's criteria are met (i.e., the expression evaluates to true), an error message is displayed, and the user is prevented from saving the record until the issue is resolved.

5.8.1 Expected Delivery Date Validation for Purchase Order Object

Objective:

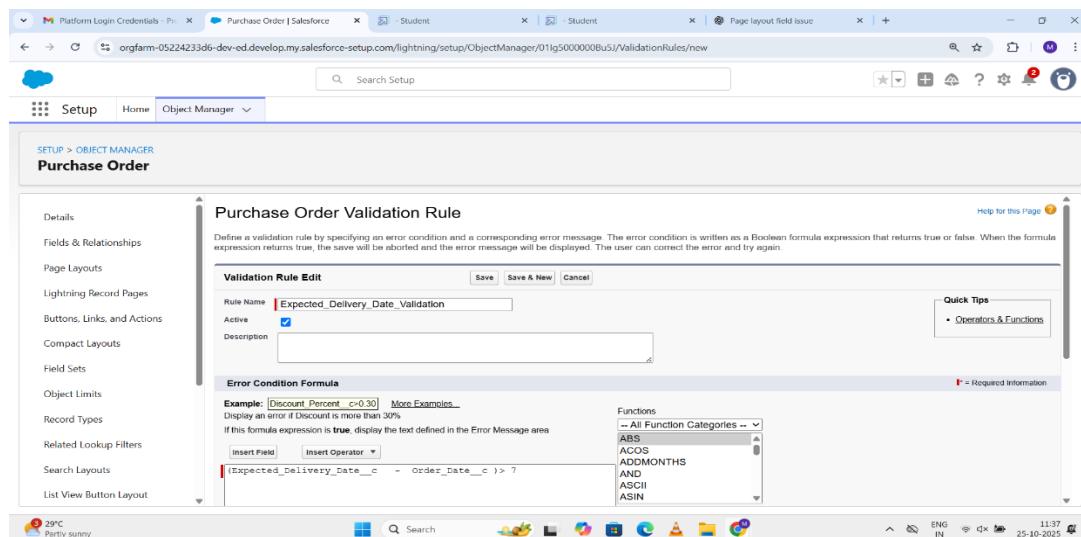
To create a validation rule that ensures the expected delivery date does not exceed 7 days from the order date.

Description:

Validation rules help maintain data integrity in Salesforce by enforcing business logic. This rule prevents users from entering an expected delivery date that is more than 7 days after the order date, ensuring timely order processing and accurate planning.

Steps:

1. Go to Setup → Object Manager → Purchase Order → Validation Rules → New.
2. Enter the Rule Name: *Expected Delivery Date Validation*.
3. Check Active.
4. Insert the Error Condition Formula:
 $(\text{Expected_Delivery_Date_c} - \text{Order_Date_c}) > 7$
5. Enter the Error Message: “*The Expected Delivery Date should not exceed 7 days.*”



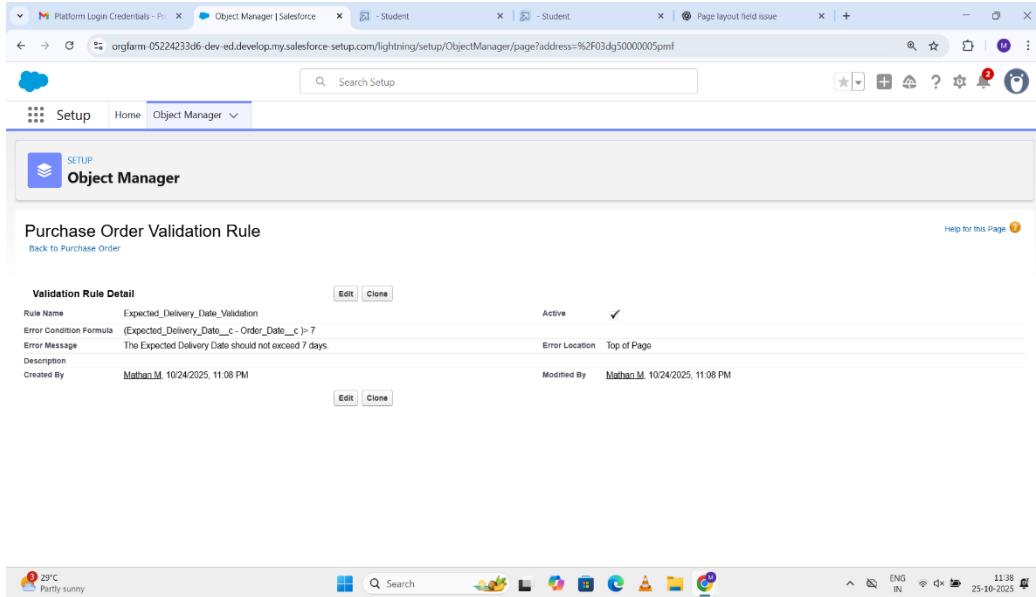
Medical Inventory Management

6. Select Error Location: Top of Page.

7. Click Save.

Outcome:

The system prevents users from entering an expected delivery date beyond 7 days from the order date, ensuring compliance with business rules.



5.9: Profiles

Profiles in Salesforce are fundamental to the platform's security model, defining what users can do within the organization. Profiles control a user's permissions to objects, fields, tabs, apps, and other settings. Each user in Salesforce must be assigned a profile, and the profile assigned to a user determines what they can see and do in the system.

5.9.1 Inventory Manager Profile

Objective:

To create a custom profile for the Inventory Manager with specific access permissions and password policies.

Description:

Profiles control the access level for users in Salesforce. The Inventory Manager profile ensures appropriate access to the **Medical Inventory Management** app and related custom objects, while enforcing secure password policies.

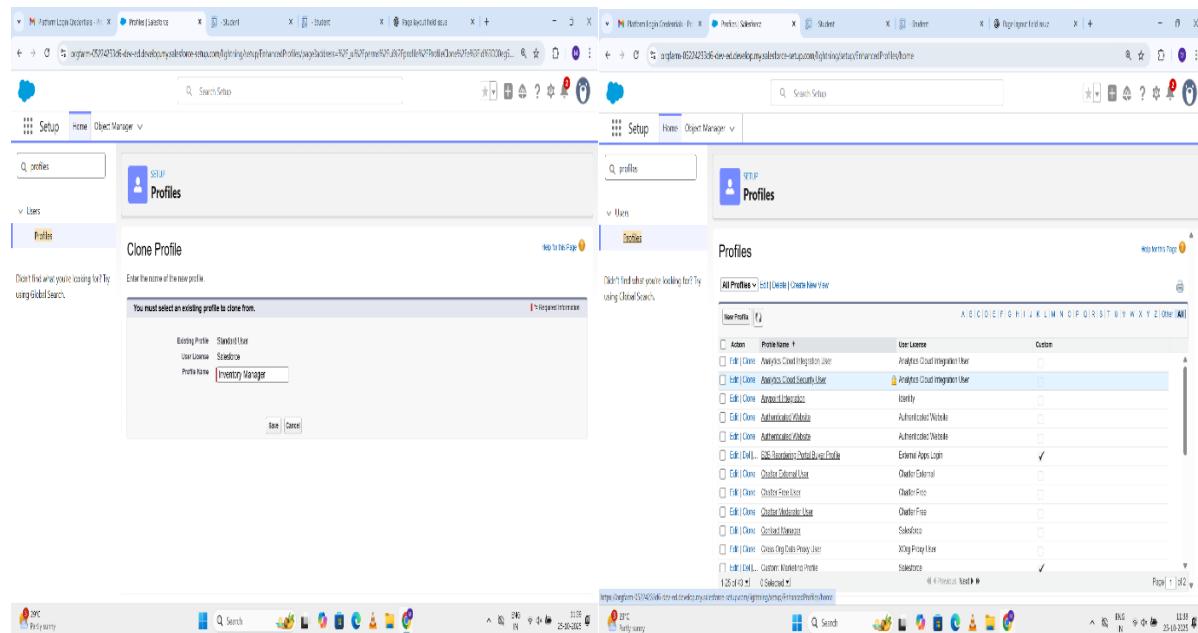
Medical Inventory Management

Steps:

1. Go to Setup → Profiles → Clone Standard User.
2. Enter Profile Name: Inventory Manager → Click Save.
3. Click Edit on the newly created profile.
4. Set the Custom App default to Medical Inventory Management.
5. Scroll down to Custom Object Permissions and assign access as required.
6. Update Password Policies:
 - User passwords expire: Never expires
 - Minimum password length: 8
7. Click Save.

Outcome:

The Inventory Manager profile is created with the necessary permissions and secure password settings.



5.9.2 Purchase Manager Profile

Objective:

To create a custom profile for the Purchase Manager with defined access to the app and objects, and proper password policies.

Medical Inventory Management

Description:

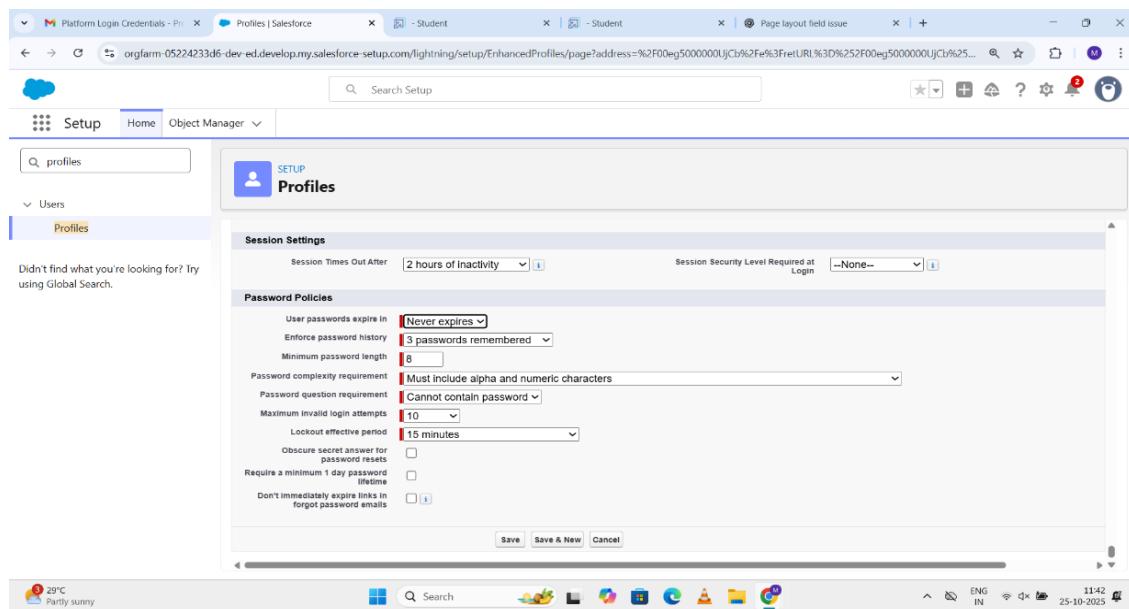
The Purchase Manager profile allows controlled access to purchase-related records and ensures secure login credentials within Salesforce.

Steps:

1. Go to Setup → Profiles → Clone Standard User.
2. Enter Profile Name: Purchase Manager → Click Save.
3. Click Edit on the newly created profile.
4. Set the Custom App default to Medical Inventory Management.
5. Scroll down to Custom Object Permissions and assign access as required.
6. Update Password Policies:
 - User passwords expire: Never expires
 - Minimum password length: 8
7. Click Save.

Outcome:

The Purchase Manager profile is successfully created with proper app access, object permissions, and password policies.



Medical Inventory Management

The screenshot shows the Salesforce Setup interface. The left sidebar is collapsed, and the main area is titled "Permission Sets". A search bar at the top right contains the text "Search Setup". Below the title, there's a sub-header "Purchase Manager Create Access". A toolbar with buttons for "Find Settings", "Clone", "Delete", "Edit Properties", "Manage Assignments", and "View Summary" is visible. Under "Object Permissions", a table lists permissions for "Order Items": Read (Enabled), Create (Enabled), Edit (Disabled), Delete (Disabled), View All Records (Disabled), and Modify All Records (Disabled). The status bar at the bottom right shows "12:06 25-10-2025".

The screenshot shows the "Assignment Summary" section of the "Purchase Manager Create Access" permission set. It displays a table with one row for "John PurchaseM" assigned to "Identity" with a "Success" status. The status bar at the bottom right shows "12:18 25-10-2025".

Medical Inventory Management

5.10: Roles

Roles in Salesforce are used to control record-level access and define the hierarchy of an organization, determining the level of visibility and sharing of records among users. Roles work in conjunction with profiles to provide a robust security model. While profiles control what actions users can perform (object and field permissions), role

5.10.1 Purchasing Manager Role

Objective:

To create a role for the Purchasing Manager that defines the hierarchy and access to purchase-related data.

Description:

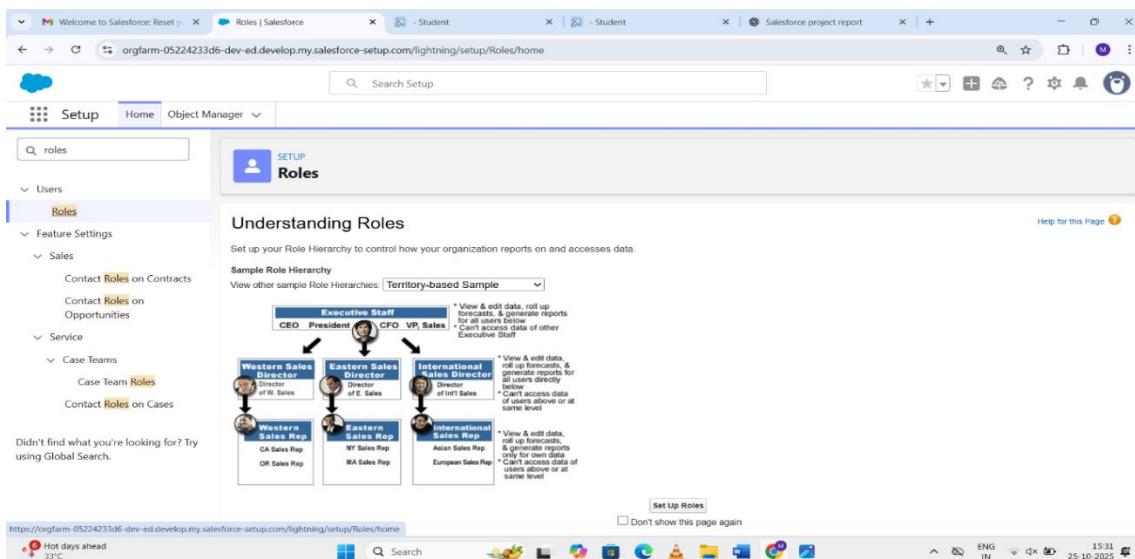
Roles in Salesforce control the level of visibility users have to records based on their position in the hierarchy. The Purchasing Manager role allows users to manage purchase operations while reporting to the SVP, Sales & Marketing.

Steps:

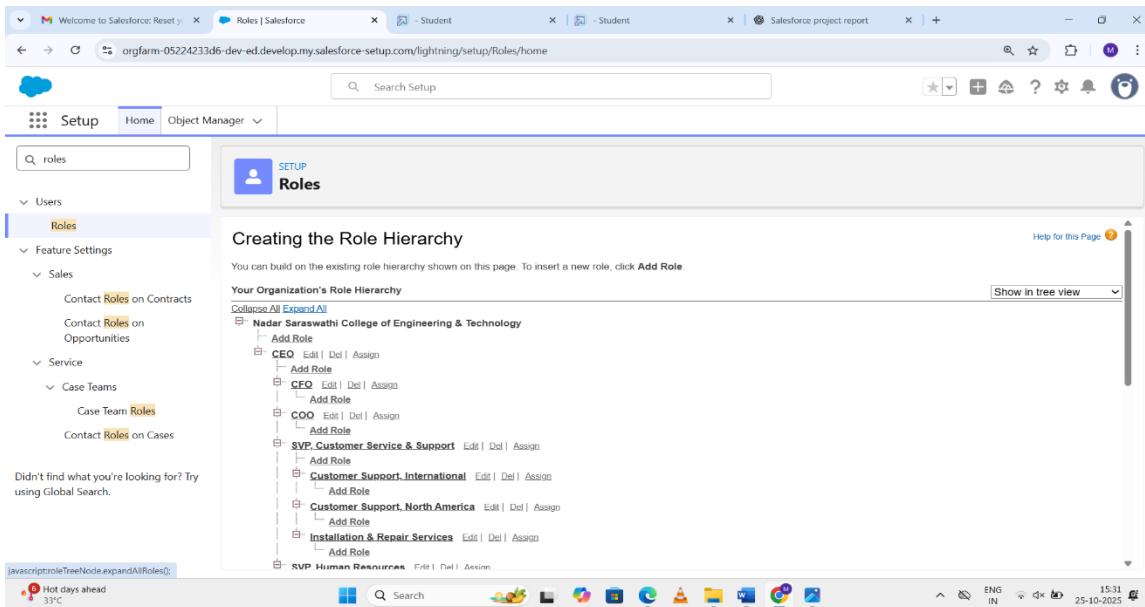
1. Go to Setup → Roles → Set Up Roles.
2. Click Expand All → click Add Role under SVP, Sales & Marketing.
3. Enter Label: Purchasing Manager → Role Name auto-populates.
4. Click Save.

Outcome:

The Purchasing Manager role is created and integrated into the Salesforce role hierarchy.



Medical Inventory Management



5.10.2 Inventory Manager Role

Objective:

To create a role for the Inventory Manager with hierarchical access to inventory-related records.

Description:

The Inventory Manager role allows controlled visibility and management of inventory data while reporting to the SVP, Sales & Marketing.

Steps:

1. Go to Setup → Roles → Set Up Roles.
2. Click Expand All → click Add Role under SVP, Sales & Marketing.
3. Enter Label: Inventory Manager → Role Name auto-populates.
4. Click Save.

Outcome:

The Inventory Manager role is successfully created in the Salesforce role hierarchy.

Medical Inventory Management

The screenshot shows the Salesforce Setup interface with the 'Roles' page selected. The 'Inventory Manager' role is displayed, which reports to the 'SVP Sales & Marketing' role. It has the label 'Inventory Manager' and the role name 'Inventory_Manager'. The 'Role Detail' section shows that users assigned to this role can edit all opportunities associated with accounts they own, regardless of who owns the opportunities. The 'Users in Inventory Manager Role' section is empty.

The screenshot shows the Salesforce Setup interface with the 'Roles' page selected. The 'Purchasing Manager' role is displayed, which reports to the 'SVP Sales & Marketing' role. It has the label 'Purchasing Manager' and the role name 'Purchasing_Manager'. The 'Role Detail' section shows that users assigned to this role can edit all opportunities associated with accounts they own, regardless of who owns the opportunities. The 'Users in Purchasing Manager Role' section shows one user named 'John PurchaseM' assigned to the role.

Medical Inventory Management

5.11: Permission Sets

Permission Sets in Salesforce are a powerful tool to extend user permissions beyond what is defined in their profiles. They allow administrators to grant additional access to various tools and functions without altering the user's profile. Permission sets are particularly useful for providing specialized permissions to specific users without the need to create multiple profiles.

5.11.1 Purchase Manager Permission Set

Objective:

To create a permission set that provides the Purchase Manager user with additional access to the **Order Item** object without modifying their profile.

Description:

Permission sets allow administrators to grant specific permissions to users without changing their profile. This ensures that John PurchaseM can access and manage **Order Item** records as required.

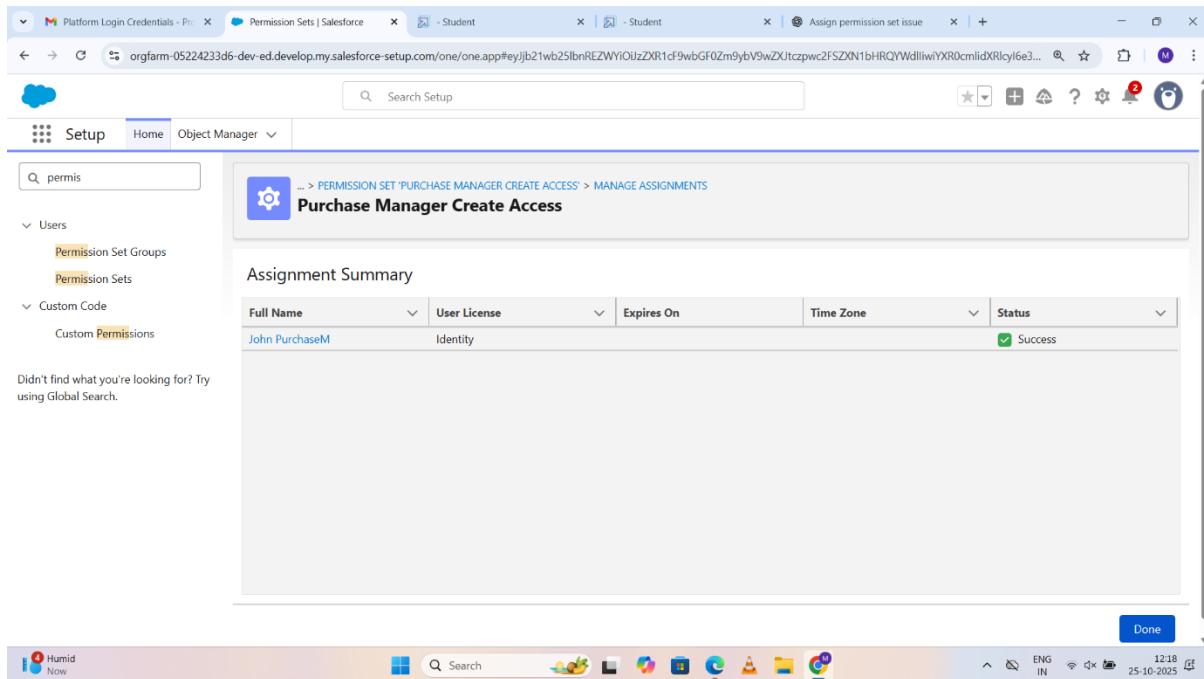
Steps:

1. Go to Setup → Permission Sets → New.
2. Enter Label: Purchase Manager Create Access → Click Save.
3. Under Object Settings → Order Item, enable:
 - Tab Available and Visible
 - Read and Create permissions
4. Click Save.
5. Navigate to the permission set detail page → Click Manage Assignments → Add Assignments.
6. Select the user John PurchaseM → Click Next.
7. Select No Expiration Date → Click Assign.

Outcome:

The Purchase Manager Create Access permission set is assigned successfully, granting John PurchaseM the required access to the Order Item object.

Medical Inventory Management



5.12: Flows

Flows in Salesforce, part of the Lightning Flow product, are powerful automation tools that help you collect data and perform actions in your Salesforce environment. Flows can be used to automate business processes, guide users through tasks, and integrate with external systems. They are highly versatile and can be configured to meet a wide range of business requirements without the need for custom code.

5.12.1 Flow to Update Actual Delivery Date

Objective:

To create a record-triggered flow that automatically updates the **Actual Delivery Date** of a purchase order based on its order date.

Description:

Flows automate business processes in Salesforce without the need for code. This flow retrieves the purchase order record, calculates the actual delivery date by adding 3 days to the order date, and updates the **Actual Delivery Date** field automatically when a record is created or updated.

Steps:

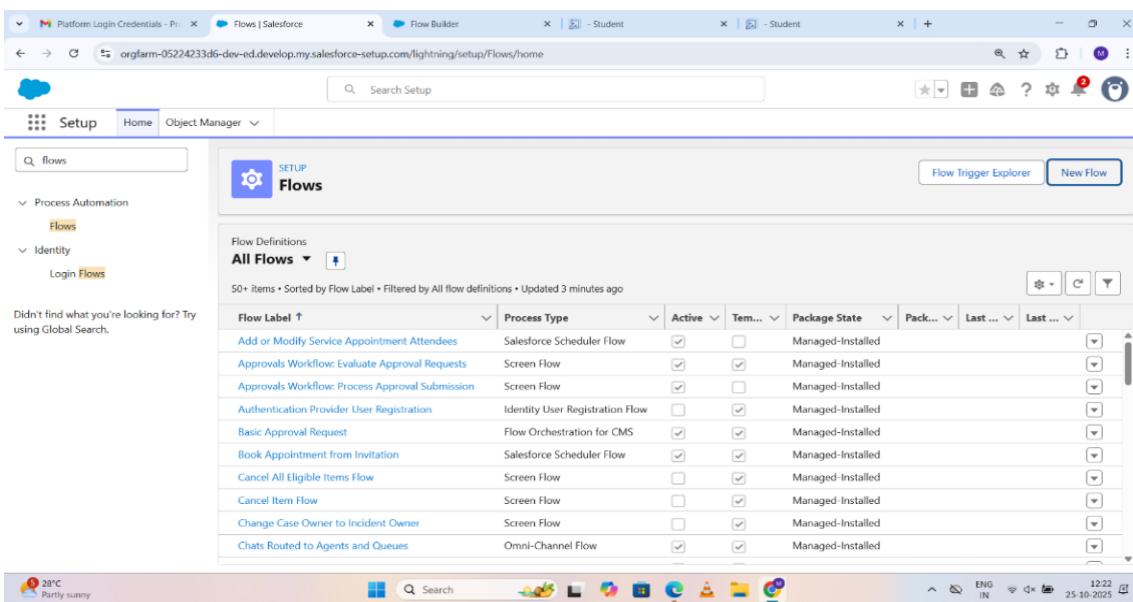
1. Go to Setup → Flow → New Flow → Start From Scratch.
2. Select Record-Triggered Flow → Create.

Medical Inventory Management

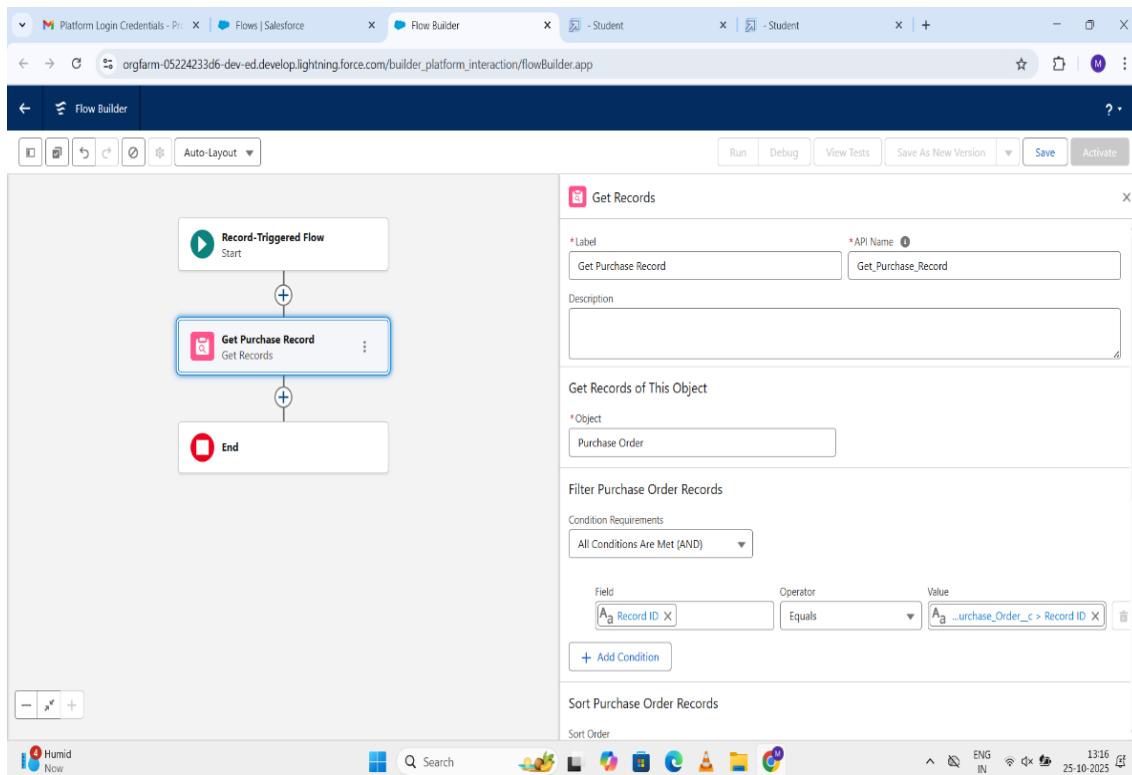
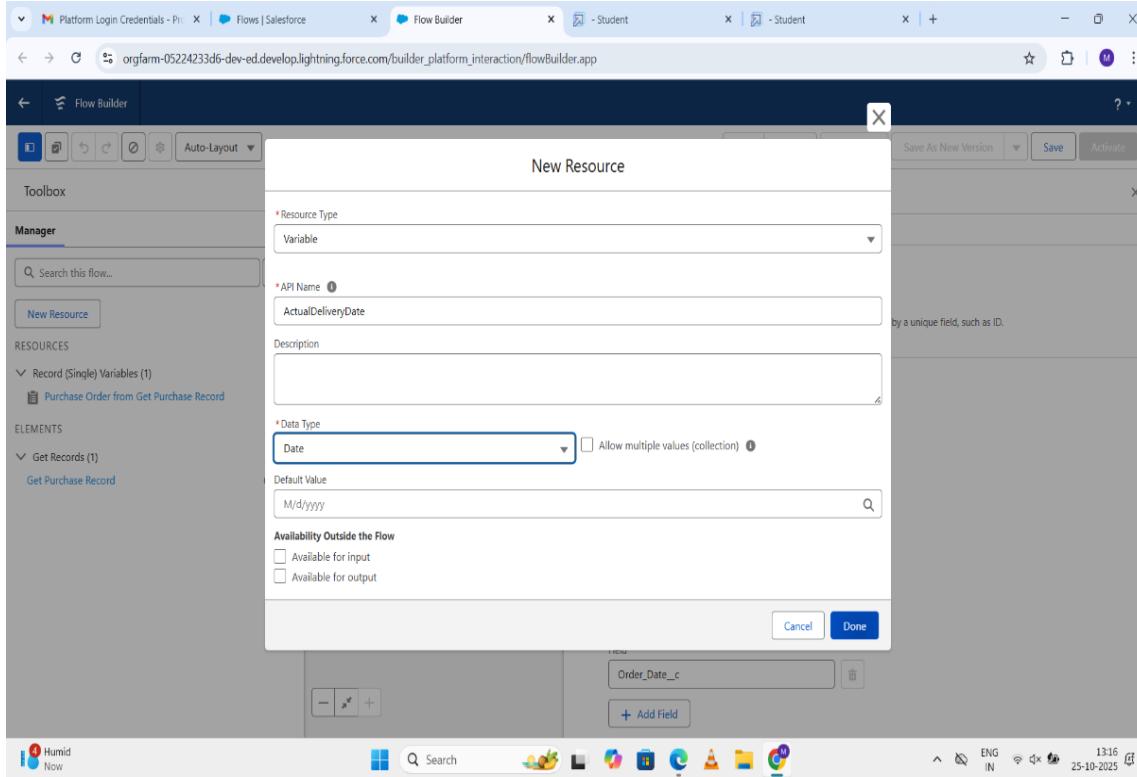
3. Under Object, select Purchase Order → Trigger: A record is created or updated → Entry Conditions: None → Fast Field Updates → Done.
4. Click the + icon → Select Get Records → Label: *Get Purchase Record* → Object: Purchase Order.
 - Condition: Id Equals {!\$Record.Id}
 - Store: Only the First Record → Fields: Order_Date__c → Done.
5. In Manager Tab → New Resource → Variable → API Name: *ActualDeliveryDate* → Data Type: Date → Done.
6. Drag Assignment element → Label: *Assignment* → Set Variable Values:
 - {!ActualDeliveryDate} = {!\$Record.Order_Date__c}
 - {!ActualDeliveryDate} + 3
→ Done.
7. Drag Update Records element → Label: *Updating Purchase Order* → Use the record that triggered the flow → Set Field: Actual_Delivery_Date__c = {!ActualDeliveryDate} → Done.
8. Save the flow as “Actual Delivery Date Updating” → Activate the flow.

Outcome:

The flow ensures that every new or updated purchase order automatically calculates and updates the **Actual Delivery Date**, improving process efficiency and reducing manual errors.

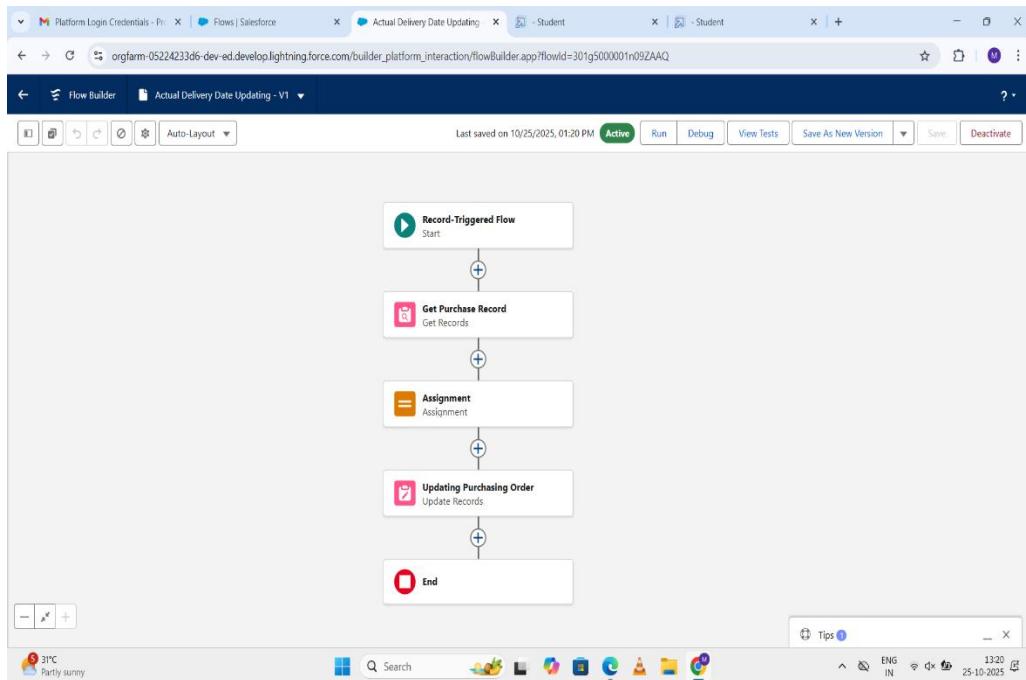


Medical Inventory Management



Medical Inventory Management

Final:



5.13: Triggers

Triggers in Salesforce are pieces of Apex code that execute before or after specific data manipulation events on Salesforce records, such as insertions, updates, deletions, and undeletions. They are powerful tools for automating complex business logic and ensuring data integrity by enforcing custom validation rules and workflows that cannot be achieved through declarative tools alone.

5.13.1 Trigger to Calculate Total Amount on Order Item

Objective:

To create an Apex trigger that automatically calculates and updates the **Total Order Cost** on the Purchase Order whenever related Order Items are inserted, updated, deleted, or undeleted.

Description:

Triggers in Salesforce allow automation of record changes using Apex code. This trigger ensures that the **Total Order Cost** of a Purchase Order reflects the sum of all associated Order Item amounts, maintaining accurate and real-time financial data.

Steps:

1. Login to Salesforce: Sign in with administrative privileges.
2. Open Developer Console: Click the gear icon → Developer Console.

Medical Inventory Management

3. Create Apex Trigger:

- o File → New → Apex Trigger.
- o Name: CalculateTotalAmountTrigger on **Order_Item__c**
- o Trigger Code:

```
trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {  
    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old, Trigger.isInsert,  
    Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);  
}
```

4. Create Apex Class:

- o File → New → Apex Class → Name: CalculateTotalAmountHandler
- o Code: Handles calculation of total amounts for Purchase Orders based on related Order Items (includes logic for insert, update, delete, and undelete scenarios).

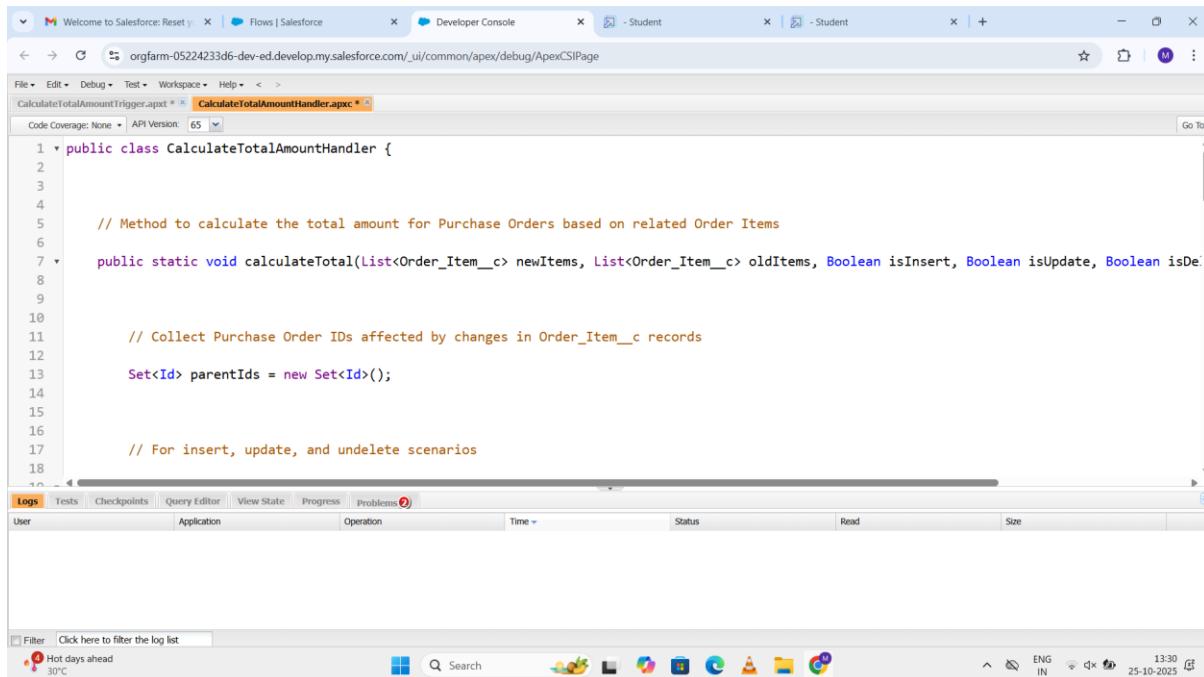
5. Save both the Trigger and Handler class.

Outcome:

The trigger automatically updates the **Total Order Cost** on Purchase Orders whenever Order Items are modified, ensuring accurate aggregation and reducing manual effort.

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes 'File', 'Edit', 'Debug', 'Test', 'Workspace', 'Help', and tabs for 'Flows | Salesforce', 'Developer Console', and 'Student'. Below the tabs, there are two code editors: 'CalculateTotalAmountTrigger.apxt' and 'CalculateTotalAmountHandler.apxc'. The 'CalculateTotalAmountTrigger.apxt' editor contains the trigger code shown in the text above. The 'CalculateTotalAmountHandler.apxc' editor is currently empty. The bottom of the screen features a toolbar with icons for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. A status bar at the bottom displays the date '25-10-2025' and time '13:30'.

Medical Inventory Management



```
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         // Collect Purchase Order IDs affected by changes in Order_Item__c records
9
10        Set<Id> parentIds = new Set<Id>();
11
12
13        // For insert, update, and undelete scenarios
14
15
16
17
18 }
```

5.14: Reports

Reports in Salesforce provide a powerful way to visualize and analyze data stored in your Salesforce organization. They allow users to create, customize, and share different types of reports based on data from standard and custom objects. Reports help organizations make informed decisions by providing insights into key metrics, trends, and performance indicators.

5.514.1 Purchase Orders Based on Suppliers (Summary) Report

Objective:

To create a summary report showing purchase orders grouped by suppliers, providing an overview of order counts and total order costs.

Description:

Summary reports in Salesforce allow aggregation of data to give insights at a glance. This report displays all purchase orders, grouped by **Supplier ID** and **Purchase Order ID**, with columns for **Order Count** and **Total Order Cost**, enabling easy tracking of supplier transactions.

Steps:

1. Open App Launcher → Medical Inventory Management App → Reports → New Report.
2. Select report type: Purchase Orders → Click Start Report.

Medical Inventory Management

3. Apply required Filters and click Apply.
4. Customize the report:
 - o Group Rows: Supplier ID, Purchase Order ID
 - o Columns: Order Count, Total Order Cost
5. Click Save & Run.
6. Enter Report Name: *Purchase Orders based on Suppliers* → Click Save.
7. To view report: Open App Launcher → Medical Inventory Management App → Reports → Click Purchase Orders based on Suppliers.

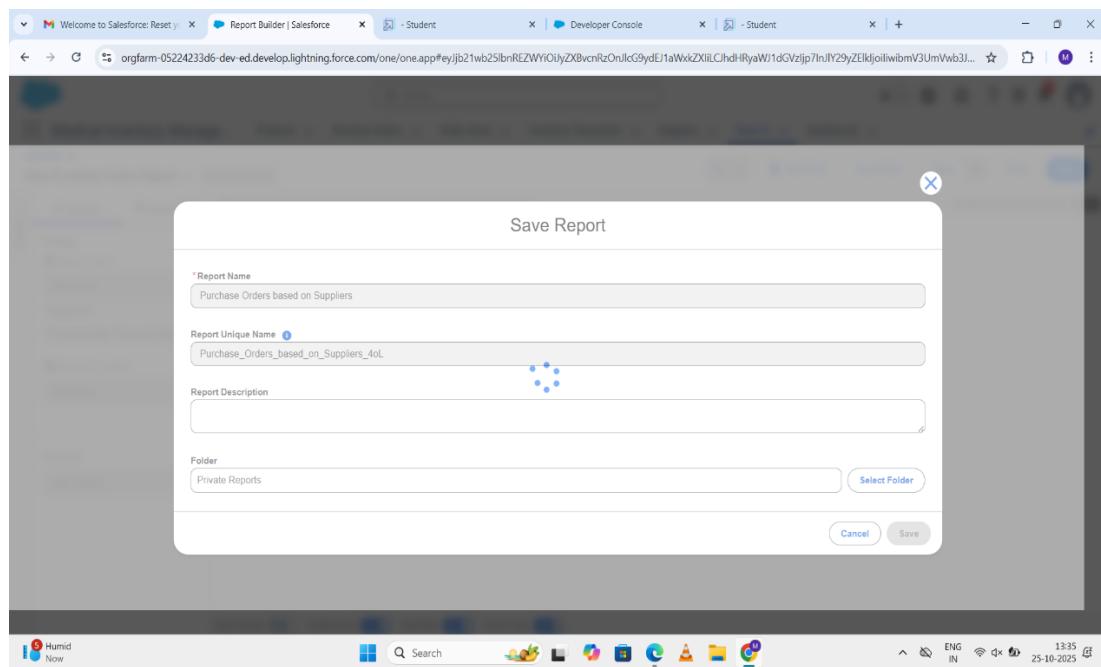
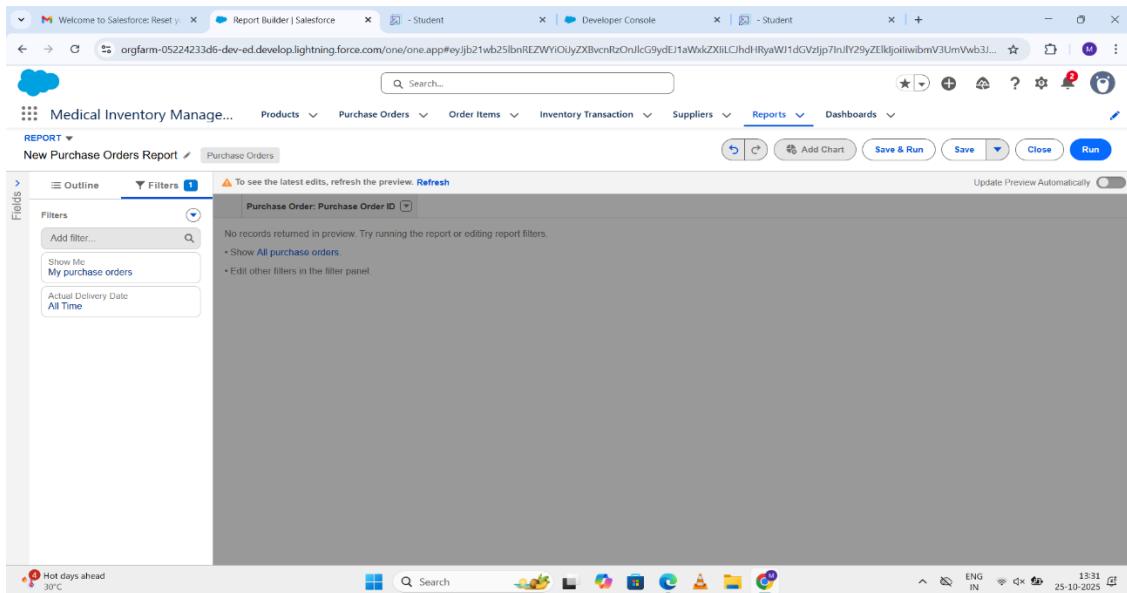
Outcome:

The report provides a summarized view of purchase orders per supplier, helping users monitor orders and financial totals efficiently.

The screenshot shows the Salesforce Flow Trigger Explorer interface. The top navigation bar includes tabs for 'Welcome to Salesforce', 'Flows | Salesforce', 'Developer Console', and 'Student'. Below the navigation is a search bar labeled 'Search Setup' and a toolbar with various icons. The main area is titled 'Flows' and displays a list of flows. The columns in the list are: Flow Label, Process Type, Active, Test, Package State, Pa..., Last..., and Last Modified Date. The flows listed are:

Flow Label	Process Type	Active	Test	Package State	Pa...	Last...	Last Modified Date
livery Date Updating	Autolaunched Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Unmanaged	Mathan M	10/25/2025, 12:50 AM	
odify Service Appointment Attendees	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-Installed			
Workflow: Evaluate Approval Requests	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Workflow: Process Approval Submission	Screen Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-Installed			
ation Provider User Registration	Identity User Registration Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
oval Request	Flow Orchestration for CMS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
ointment from Invitation	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Eligible Items Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Cancel Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Change Case Owner to Incident Owner	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			

Medical Inventory Management



5.14.2 Complete Purchase Details Report

Objective:

To create a detailed report showing complete purchase information, including order items and product details.

Description:

This report combines **Purchase Orders**, **Order Items**, and **Product details** to provide a

Medical Inventory Management

comprehensive overview of transactions, including quantities, product names, and amounts, facilitating detailed analysis.

Steps:

1. Open App Launcher → Medical Inventory Management App → Reports → New Report.
2. Select report type: Purchase Orders with Order Items and Product ID → Click Start Report.
3. Apply required Filters and click Apply.
4. Customize the report:
 - o Group Rows: Supplier ID, Actual Delivery Date, Purchase Order ID
 - o Columns: Product ID, Product Name, Order Count, Quantity Received, Amount
5. Click Save & Run.
6. Enter Report Name: *Complete Purchase Details Report* → Click Save.

Outcome:

The report provides detailed purchase information per supplier and product, enabling effective tracking and decision-making.

The screenshot shows a Salesforce Lightning page titled 'Complete Purchase Details Report'. The page displays a table of purchase order data. The columns include Supplier ID, Actual Delivery Date, Purchase Order ID, Order Count, Order Date, and Purchase Order: Owner Name. The data is grouped by Supplier ID and Actual Delivery Date, with subtotals for each group. The table has 5 total records. At the bottom, there are buttons for Row Counts, Detail Rows, Subtotals, and Grand Total, along with a 'Very high UV Now' status indicator. The top navigation bar includes links for Products, Purchase Orders, Order Items, Inventory Transaction, Suppliers, Reports, and Dashboards.

Supplier ID	Actual Delivery Date	Purchase Order: Purchase Order ID	Order Count	Order Date	Purchase Order: Owner Name
1 (2)	10/25/2025 (1)	1 (1)	a06g5000000Gsyv	0	10/22/2025
		Subtotal			
	11/1/2025 (1)	11 (1)	a06g5000000Gt0X	0	10/29/2025
		Subtotal			
	Subtotal				
2 (2)	10/4/2025 (1)	21 (1)	a06g5000000Gt8b	0	10/1/2025
		Subtotal			
	10/11/2025 (1)	25 (1)	a06g5000000GIAD	0	10/8/2025
		Subtotal			
	Subtotal				

Medical Inventory Management

The screenshot shows a Salesforce Lightning interface with the following details:

- Header:** Welcome to Salesforce: Reset, Medical Inventory Dashboard, Student, Purchase Orders based on Suppliers.
- Top Navigation:** Medical Inventory Manage..., Products, Purchase Orders, Order Items, Inventory Transaction, Suppliers, Reports, Dashboards.
- Report Title:** Report: Purchase Orders, Purchase Orders based on Suppliers.
- Table Data:** Total Records: 5

Supplier ID	Purchase Order: Purchase Order ID	Order Count	Total Order Cost
1 (2)	1 (1)	0	0
	11 (1)	0	0
2 (2)	21 (1)	0	0
	25 (1)	0	0
3 (1)	54 (1)	0	0
Total (5)			

- Report Settings:** Row Counts, Detail Rows, Subtotals, Grand Total.
- System Status:** Very high UV Now.
- System Icons:** Search, Home, File, Recent, Applications, Help.
- System Information:** ENG IN, 1405, 25-10-2025.

5.15: Dashboards

Dashboards in Salesforce are dynamic visual representations of key metrics and data from reports, providing a consolidated view of organizational performance and trends. They are powerful tools for monitoring real-time data, tracking progress towards goals, and gaining actionable insights at a glance. Dashboards consist of components such as charts, tables, metrics, and gauges that display data from underlying reports.

5.15.1 Create Dashboard

Objective:

To create a dashboard that visually represents purchase order data for quick analysis and monitoring.

Description:

Dashboards provide a graphical view of report data, enabling users to track key metrics and trends efficiently. This dashboard displays purchase orders by suppliers using charts or tables for better insights.

Steps:

1. Go to the Dashboards tab in the Medical Inventory Management App → Click New Dashboard.
2. Enter Name: *Medical Inventory Dashboard* → Click Create.
3. Click +Widget → Select the report: Purchase Orders based on Suppliers.

Medical Inventory Management

4. Choose a data visualization type (chart, table, etc.) as required → Click Add.
5. Click Save.

Outcome:

The dashboard provides a visual summary of purchase orders per supplier, enabling quick data interpretation and decision-making.

This screenshot shows the Salesforce Lightning interface. The top navigation bar includes links for 'Products', 'Purchase Orders', 'Order Items', 'Inventory Transaction', 'Suppliers', 'Reports', and 'Dashboards'. The 'Dashboards' link is currently selected. Below the navigation is a search bar and a toolbar with various icons. On the left, a sidebar titled 'Medical Inventory Management' lists sections for 'Dashboards', 'Recent', 'Folders', and 'Favorites'. The main content area displays a table titled 'Dashboards' with one item: 'Medical Inventory DashBoard' (Created by Mathan M on 10/25/2025, 1:23 AM). A 'Search recent dashboards...' input field and buttons for 'New Dashboard' and 'New Folder' are at the bottom of the table. The status bar at the bottom shows the URL, weather (33°C, mostly cloudy), system information (ENG IN), and date (25 10 2025).

This screenshot shows the 'Add Widget' dialog box overlaid on the Salesforce interface. The dialog has a title 'Add Widget' and a 'Report' section containing a report titled 'Purchase Orders based on Suppliers'. It includes options to 'Use chart settings from report' and 'Display As' (with a dropdown menu showing various chart types like bar, line, pie, etc.). Below these are fields for 'Value' (set to 'Record Count') and 'Sliced By' (set to 'Supplier ID'). To the right is a 'Preview' window showing a dark chart area with the message 'We can't draw this chart because there is no data' and a link 'View Report (Purchase Orders based on Suppliers)'. At the bottom of the dialog are 'Cancel' and 'Add' buttons. The background shows the same Salesforce interface as the previous screenshot, with the 'Medical Inventory DashBoard' visible in the sidebar. The status bar at the bottom shows the URL, weather (30°C, partly sunny), system information (ENG IN), and date (25-10-2025).

Medical Inventory Management

5.15.2 View Dashboard

Objective:

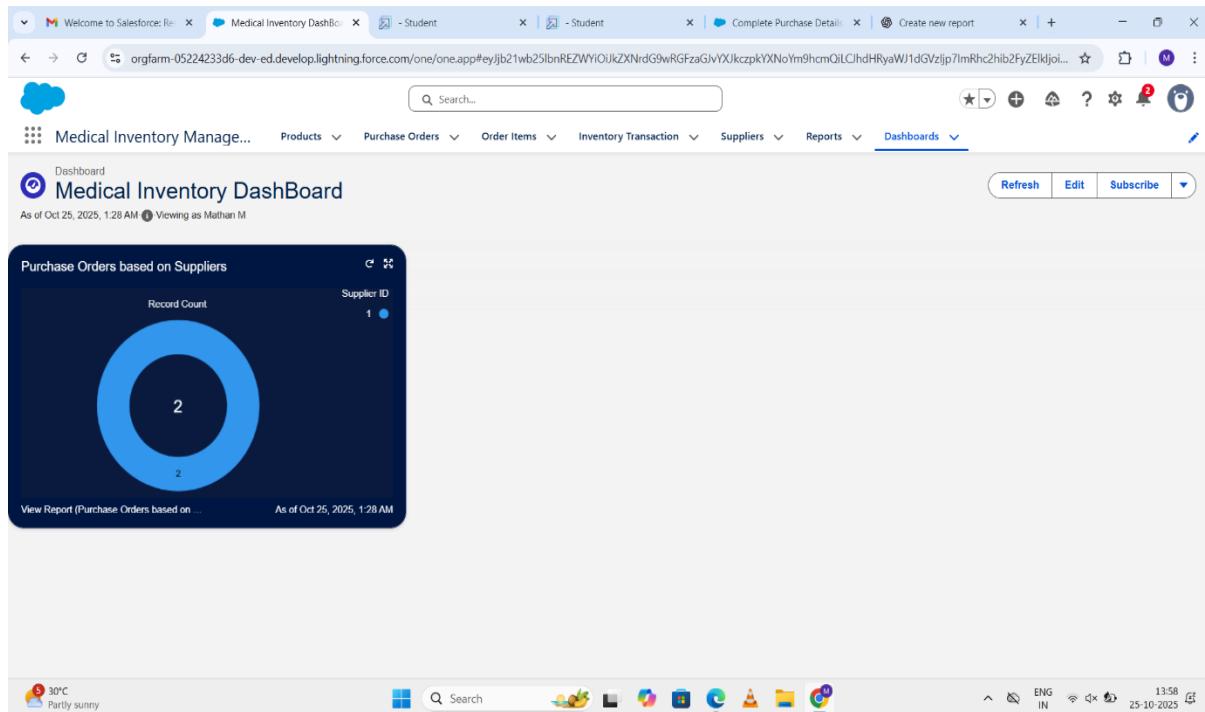
To view the created dashboard and analyze purchase order data visually.

Steps:

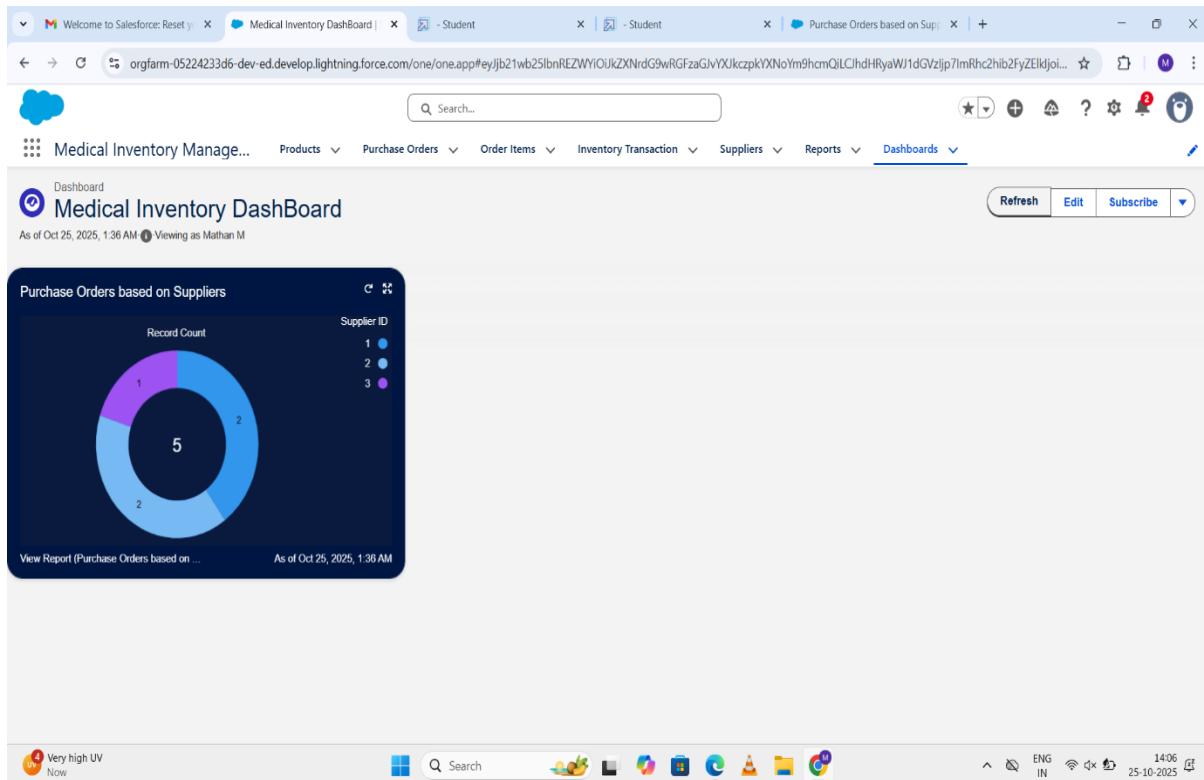
1. Go to App Launcher → Medical Inventory Management App → Dashboard Tab.
2. Click on Medical Inventory Dashboard to view the records in a graphical format.

Outcome:

Users can visually analyze purchase order trends and supplier performance using the dashboard



Medical Inventory Management



6. Outcome

The Medical Inventory Management System has been successfully developed as an end-to-end Salesforce-based solution that automates and simplifies the management of medical products, suppliers, and purchase operations. This system provides an integrated platform for efficiently maintaining all medical inventory-related activities in a centralized environment, reducing manual workload and minimizing the chances of data inconsistency.

Through the creation of custom objects, such as *Product*, *Purchase Order*, *Order Item*, *Inventory Transaction*, and *Supplier*, the system ensures that all aspects of the medical supply chain are well-structured and interlinked. Each object is enhanced with custom fields to capture essential business data such as product quantity, unit price, and total order cost. This customization allows organizations to maintain precise and real-time records of every inventory transaction.

The implementation of page layouts, compact layouts, and record relationships ensures that users can access relevant data quickly and in an organized manner. Validation rules enhance data accuracy by enforcing logical constraints, while automation through Flows and Apex Triggers streamlines updates—such as automatically calculating delivery dates and total order costs—without manual intervention.

Moreover, the integration of reports and dashboards empowers management with valuable insights into purchasing activities, supplier performance, and stock levels.

Medical Inventory Management

These visual tools aid in effective decision-making, enabling administrators to track order history, analyze procurement trends, and identify areas for cost optimization.

With the inclusion of user profiles, roles, and permission sets, the system provides a secure and hierarchical structure for data access, ensuring that only authorized users can perform specific operations. This contributes to better governance and accountability across the organization.

Overall, the project has resulted in a robust, scalable, and intelligent Salesforce application that enhances operational efficiency, data reliability, and real-time monitoring of medical inventory. The system not only improves the accuracy of inventory tracking but also supports better planning, reduces wastage, and ensures timely availability of medical supplies—making it a significant step toward digital transformation in healthcare inventory management.

7. Future Enhancements / Recommendations

While the Medical Inventory Management System efficiently manages suppliers, purchase orders, and inventory transactions, there are several opportunities for future enhancements to further improve automation, scalability, and user experience. The following recommendations outline potential directions for system advancement:

7.1. Integration with Hospital or Clinic Management Systems

- Connect the inventory system with hospital or clinic management software to ensure real-time synchronization of product usage, billing, and replenishment.
- This would create a seamless ecosystem between medical operations and supply chain management.

7.2. Barcode and QR Code Scanning

- Implement barcode or QR code scanning functionality for faster and more accurate product tracking during stock entry, dispatch, and verification processes.
- This feature would significantly reduce manual errors and speed up inventory handling.

7.3. Automated Expiry and Stock Notifications

- Add scheduled flows or email alerts to automatically notify users about upcoming product expirations or low stock levels.
- This ensures proactive decision-making and timely replenishment.

Medical Inventory Management

7.4. Supplier Performance Analytics

- Develop advanced analytical dashboards to measure supplier performance based on delivery time, quality, and cost efficiency.
- This will enable management to identify reliable suppliers and negotiate better contracts.

7.5. Enhanced Data Security and Audit Trails

- Implement additional security measures like field-level encryption and audit logs to monitor every data change.
- This would ensure full compliance with healthcare and data protection standards.

8. Conclusion

The Medical Inventory Management System project in Salesforce demonstrates the effective use of Salesforce features such as custom objects, fields, relationships, validation rules, flows, triggers, reports, and dashboards to automate and streamline inventory processes.

By creating objects like Product, Purchase Order, Order Item, Inventory Transaction, and Supplier, and adding relevant custom fields, the system captures detailed business data efficiently. Automation through flows and Apex triggers ensures real-time updates of critical information, such as Actual Delivery Dates and Total Order Costs, reducing manual errors and improving operational accuracy.

The creation of profiles, roles, and permission sets ensures secure and controlled access to data, maintaining proper governance and hierarchy. Reporting and dashboards provide actionable insights, enabling management to monitor supplier performance, purchase details, and inventory trends effectively.

Overall, this project highlights Salesforce's capabilities in customization, automation, and data visualization, resulting in a robust, user-friendly system that enhances productivity, transparency, and decision-making in managing medical inventory.