**PROJECT DEVELOPMENT PHASE**

**Date:** 18-06-2025

**Team ID:** LTVIP2025TMID28927

**Project Name**: Medical Inventory Management – (Developer)

**Maximum Marks:** *(To be filled by evaluator)*

**1. OVERVIEW**

This document captures the technical deliverables implemented as part of the **Medical Inventory Management System** project in Salesforce. It includes details of **custom configurations, automation logic, sample datasets, and screenshots of functional outputs**. The goal is to demonstrate the executable components of the solution and provide visual evidence that the **core functionality has been correctly built and deployed**.

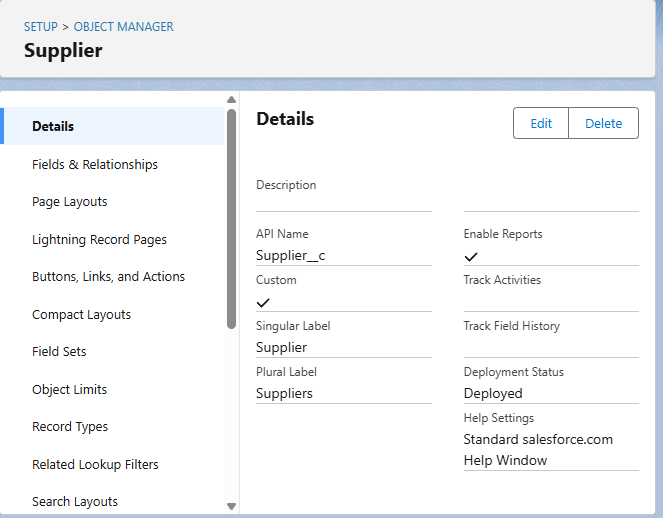
This section serves as a proof of execution, showcasing that the development aligns with the problem statement, solution design, and project objectives defined in the earlier phases for the **efficient management of medical inventory**.

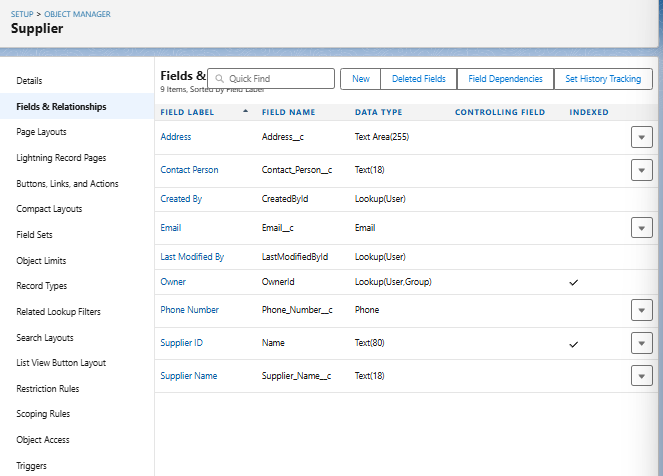
**2. REQUIRED PROJECT FILES**

### A. Custom Objects and Their Roles

#### 1. Supplier\_\_c

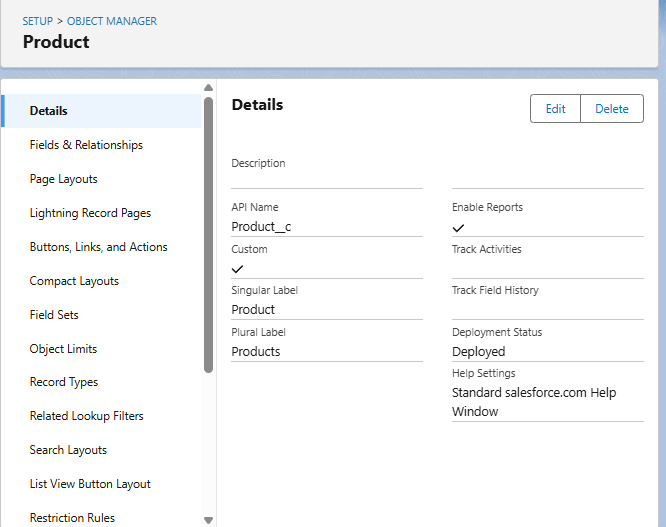
* **Purpose**: Stores comprehensive information about medical product suppliers, including their name, contact person, phone number, email address, physical address, and any specific terms or notes related to their partnership.
* **Usage**: This object serves as the central point for all supplier-related data, enabling efficient management of procurement and communication. It is critical for maintaining accurate records of where medical products are sourced.
* **Key Fields**:
  + Supplier\_Name\_\_c (Text)
  + Contact\_Person\_\_c (Text)
  + Phone\_Number\_\_c (Phone)
  + Email\_\_c (Email)
  + Address\_\_c (Text Area)

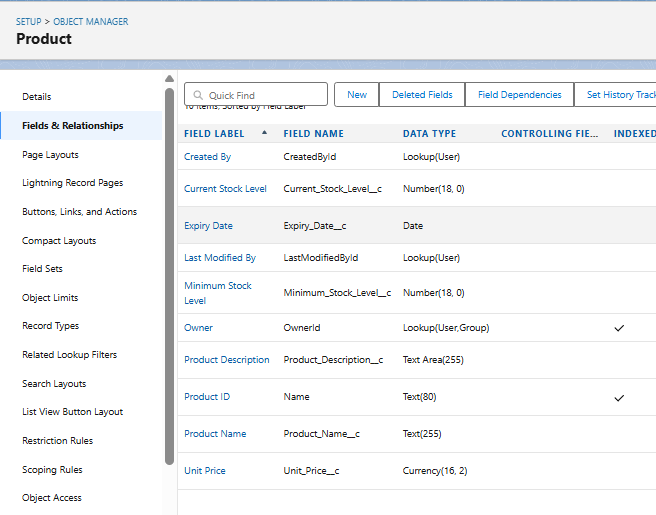
****

****

#### 2. Product\_\_c

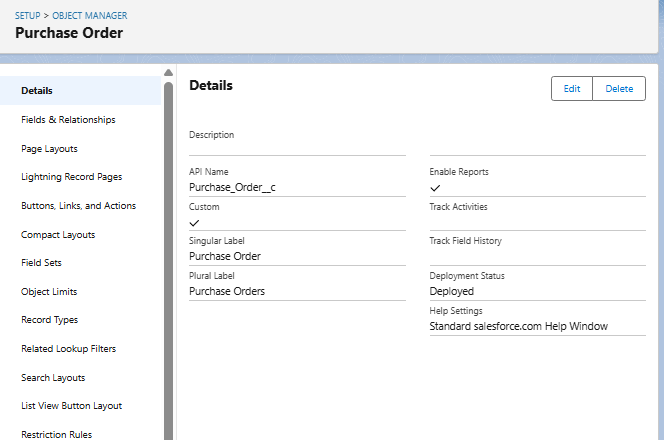
* **Purpose**: Stores detailed information about each medical product in the inventory. This includes crucial attributes like product name, description, unique identifiers, current stock levels, minimum reorder levels, unit price, and vital expiry dates.
* **Usage**: This object is fundamental for tracking all individual items within the medical inventory. It allows for efficient stock management, helps prevent stockouts, and ensures that expired products are identified and managed appropriately.
* **Key Fields**:
  + Product\_Name\_\_c (Text)
  + Product\_Description\_\_c (Text Area)
  + Minimum\_Stock\_Level\_\_c (Number)
  + Current\_Stock\_Level\_\_c (Number)
  + Unit\_Price\_\_c (Currency)
  + Expiry\_Date\_\_c (Date)

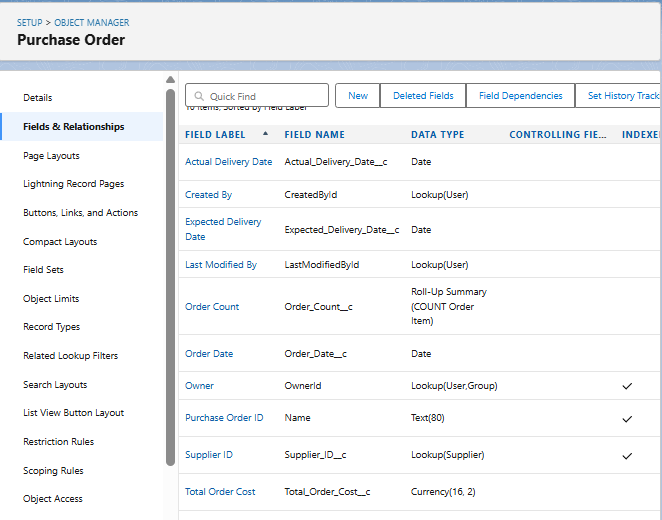




#### 3. Purchase\_Order\_\_c

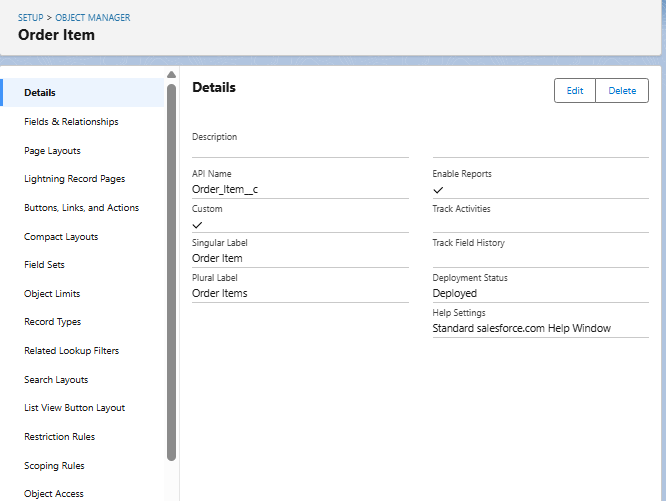
* **Purpose**: Represents a formal request made to a Supplier\_\_c for a specific quantity of Product\_\_cs. It captures details such as the order date, expected and actual delivery dates, the associated supplier, the total cost of the order, and the number of distinct items ordered.
* **Usage**: This object is essential for managing the procurement process. It allows for tracking the status of incoming inventory, linking orders to specific suppliers, and providing an overview of pending and completed purchases.
* **Key Fields**:
  + Supplier\_ID\_\_c (Lookup to Supplier\_\_c)
  + Order\_Date\_\_c (Date)
  + Expected\_Delivery\_Date\_\_c (Date)
  + Actual\_Delivery\_Date\_\_c (Date)
  + Order\_Count\_\_c (Roll-Up Summary, COUNT of Order\_Item\_\_c)
  + Total\_Order\_Cost\_\_c (Currency)

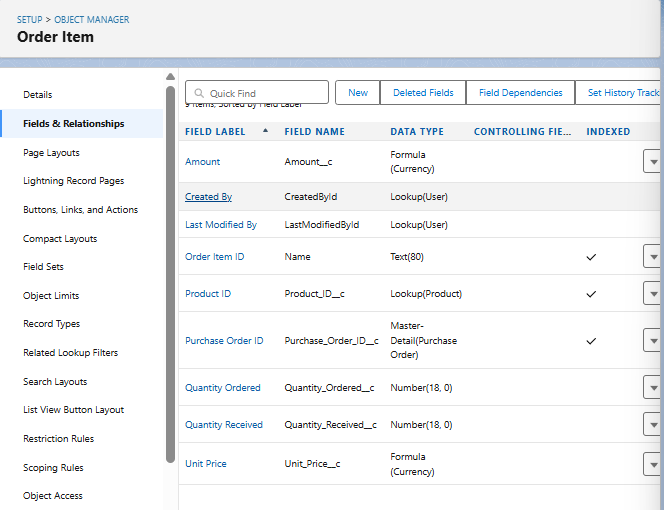




#### 4. Order\_Item\_\_c

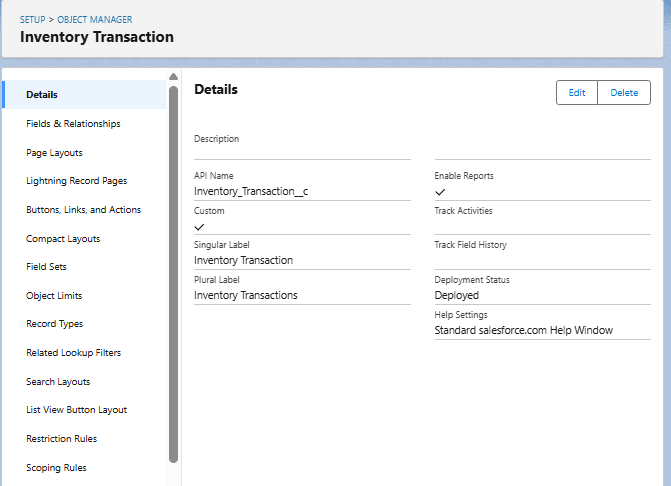
* **Purpose**: Represents a specific line item within a Purchase\_Order\_\_c. Each record details a particular Product\_\_c being ordered, the quantity requested, the quantity received, and the calculated amount for that item.
* **Usage**: This object forms the bridge between Purchase\_Order\_\_c and Product\_\_c, breaking down a single purchase order into its constituent products. It's crucial for granular tracking of what products are part of each order and their individual quantities and costs.
* **Key Fields**:
  + Product\_ID\_\_c (Lookup to Product\_\_c)
  + Purchase\_Order\_ID\_\_c (Master-Detail to Purchase\_Order\_\_c)
  + Quantity\_Ordered\_\_c (Number)
  + Quantity\_Received\_\_c (Number)
  + Unit\_Price\_\_c (Formula, Currency)
  + Amount\_\_c (Formula, Currency)

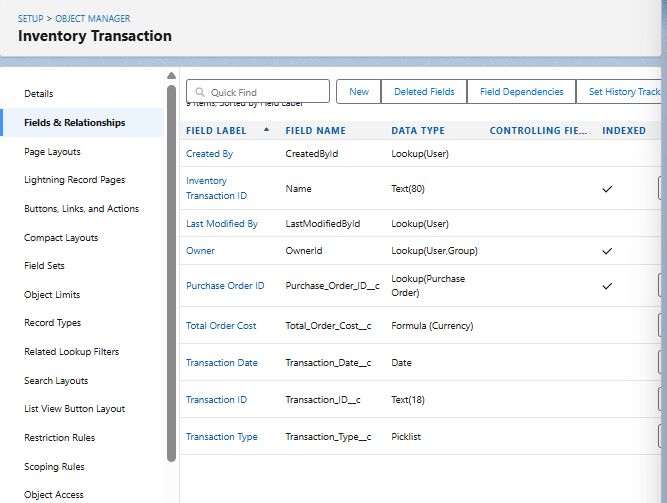




#### 5. Inventory\_Transaction\_\_c

* **Purpose**: Records all movements and adjustments of products within the inventory. This includes various transaction types such as receiving stock (from purchase orders), dispensing items, returns, or adjustments due to spoilage/damage. It links back to the relevant Purchase\_Order\_\_c if applicable and captures the date and type of transaction.
* **Usage**: This object provides an audit trail for all changes to product stock levels, allowing for accurate inventory reconciliation, identifying discrepancies, and understanding the flow of products in and out of storage.
* **Key Fields**:
  + Purchase\_Order\_ID\_\_c (Lookup to Purchase\_Order\_\_c)
  + Transaction\_Date\_\_c (Date)
  + Transaction\_Type\_\_c (Picklist - e.g., 'Received', 'Dispensed', 'Adjustment')
  + Total\_Order\_Cost\_\_c (Formula, Currency - Note: This field might be more relevant to a Purchase Order. For an inventory transaction, a 'Quantity\_Changed' or 'Value\_Changed' might be more appropriate depending on transaction type. Reconfirm if Total\_Order\_cost\_\_c makes sense here, perhaps it refers to the cost of the items in this specific transaction.)





**B. AUTOMATION ELEMENTS**

The Medical Inventory Management System leverages Salesforce's automation capabilities to streamline business processes, reduce manual interventions, and enhance operational accuracy. The automation comprises Apex Triggers, Record-Triggered Flows, and Validation Rules, all tailored to meet the critical needs of a medical inventory and procurement operations.

**1. Apex Trigger –**

* **Name:** CalculateTotalAmountTrigger
* **Trigger Type:** after insert, after update, after delete, after undelete
* **Description:**  
  This use case works for Amount Distribution for each Service the customer selected for the Vehicle
* **Core Logic:**

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);

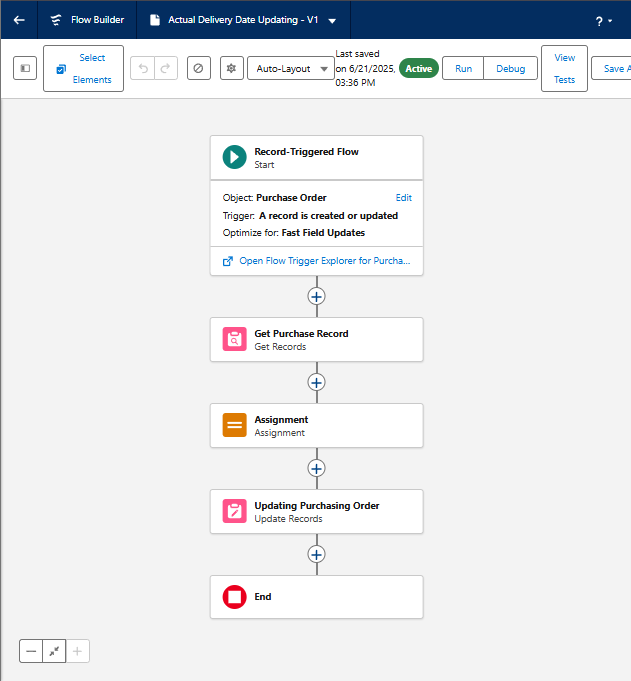
}

### 2. Flow Name: Actual\_Delivery\_Date\_Updating

* **Purpose**: This Record-Triggered Flow automates the initial estimation of the Actual\_Delivery\_Date\_\_c on a Purchase\_Order\_\_c record. Upon the creation or update of a Purchase Order, the flow automatically calculates and populates this field to be 3 days after the Order\_Date\_\_c. This provides an immediate, intelligent default for the actual delivery date, streamlining data entry and offering a preliminary timeline for incoming inventory.
* **Associated Object**: Purchase\_Order\_\_c
* **Trigger Event**: A record is **created or updated**.
* **Optimization for Related Records**: Fast Field Updates (The flow runs before the record is saved to the database).

#### Step-by-Step Flow Logic:

1. **Start Element (Trigger)**:
   * **Object**: Purchase\_Order\_\_c
   * **Trigger the Flow When**: A record is created or updated.
   * **Entry Conditions**: None (The flow should run for all new or updated Purchase Orders).
   * **Optimize the Flow for**: Fast Field Updates (This ensures the Actual\_Delivery\_Date\_\_c is updated before the record is officially committed, making it ideal for updating fields on the triggering record itself).
2. **Assignment Element**:
   * **Label**: Assignment (or more descriptive: Set Actual Delivery Date)
   * **Purpose**: This element calculates the new Actual\_Delivery\_Date\_\_c value by adding 3 days to the Order\_Date\_\_c.
   * **Variables/Fields Set**:
     + **Variable**: A Date type variable (e.g., vActualDeliveryDate)
     + **Operator**: Equals
     + **Value**: {!$Record.Order\_Date\_\_c} + 3
     + (Note: While your milestone used a variable ActualDeliveryDate and then an update element, for a "Fast Field Update" flow updating the same record, you can directly update the record's field without an explicit "Get Records" if the field is on the triggering record itself. However, sticking to your provided logic for clarity based on your Milestone 13 helps.)
3. **Update Records Element**:
   * **Label**: Updating Purchase Order (or more descriptive: Update Actual Delivery Date)
   * **Purpose**: This element takes the calculated Actual\_Delivery\_Date\_\_c and applies it to the Purchase\_Order\_\_c record that initiated the flow.
   * **How to Find Records to Update and Set Their Values**: Use the Purchase\_Order\_\_c record that triggered the flow.
   * **Set Field Values for the Purchase\_Order\_\_c Record**:
     + **Field**: Actual\_Delivery\_Date\_\_c
     + **Value**: {!vActualDeliveryDate} (Referring to the variable from the Assignment step).
4. **Save and Activate**:
   * Save the flow with the name Actual Delivery Date Updating.
   * Activate the flow to ensure it runs automatically whenever a Purchase\_Order\_\_c record is created or updated.

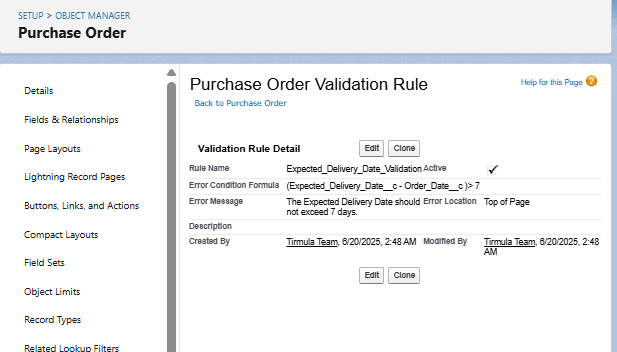


## 3. Validation Rules

Validation Rules are essential for maintaining data quality and consistency within the Medical Inventory Management System. They enforce business logic by preventing users from saving records that do not meet specified criteria.

### Validation Rule Name: Expected\_Delivery\_Date\_Validation

* **Associated Object**: Purchase\_Order\_\_c
* **Purpose**: This Validation Rule ensures that the Expected\_Delivery\_Date\_\_c on a Purchase\_Order\_\_c record is set within a reasonable timeframe relative to the Order\_Date\_\_c. Specifically, it prevents the expected delivery date from being more than 7 days after the order date. This helps in managing supplier expectations and ensuring timely procurement of medical supplies.
* **Formula Logic**:
* (Expected\_Delivery\_Date\_\_c - Order\_Date\_\_c) > 7
  + **Explanation**: This formula checks if the number of days between the Expected\_Delivery\_Date\_\_c and the Order\_Date\_\_c is greater than 7. If this condition evaluates to TRUE, the validation rule fires, preventing the record from being saved.
* **Error Message**: "The Expected Delivery Date should not exceed 7 days."
* **Error Location**: Top of Page
* **Usage Scenario**:
  + When a user creates a new Purchase\_Order\_\_c record or modifies an existing one, they are required to input an Order\_Date\_\_c and an Expected\_Delivery\_Date\_\_c.
  + If the user enters an Expected\_Delivery\_Date\_\_c that is, for example, 10 days after the Order\_Date\_\_c, this validation rule will trigger.
  + The system will display the error message "The Expected Delivery Date should not exceed 7 days." at the top of the record page, preventing the user from saving the invalid data.
  + The user must then adjust the Expected\_Delivery\_Date\_\_c to be within 7 days of the Order\_Date\_\_c to successfully save the record.



**C. UI COMPONENTS**

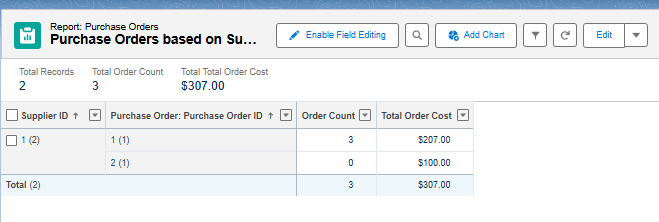
**Reports Configuration**

**Use Case:**

Reports in the Medical Inventory Management System provide valuable insights into supplier performance, purchase orders, and overall inventory status. They are essential tools for tracking key metrics and enabling data-driven decision-making.

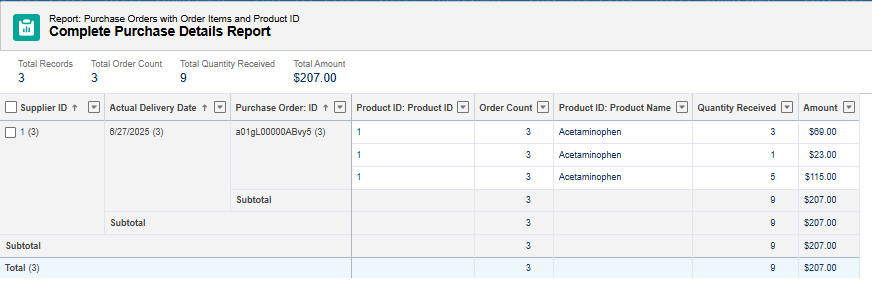
**Activity 1: Create a "Purchase Orders based on Suppliers" Report**

1. **Navigate to Reports**:
   * From App Launcher, select Medical Inventory Management App.
   * Click on the Reports tab.
   * Click New Report.
2. **Select Report Type**: Choose Purchase Orders. Click Start Report.
3. **Configure Filters**:
   * Show Me: All Purchase Orders.
   * Date Field: Order Date.
   * Range: All Time. Click Apply.
4. **Add Groupings & Columns**:
   * **Group Rows**: Supplier ID, Purchase Order: Purchase Order ID.
   * **Columns**: Order Count, Total Order Cost.
5. **Save Report**:
   * Click Save & Run.
   * **Report Name**: Purchase Orders based on Suppliers.
   * Click Save.



**Activity 2: Create a "Complete Purchase Details Report"**

1. **Navigate to Reports**:
   * From App Launcher, select Medical Inventory Management App.
   * Click on the Reports tab.
   * Click New Report.
2. **Select Report Type**: Choose Purchase Orders with Order Items and Product ID. Click Start Report.
3. **Configure Filters**:
   * Show Me: All Purchase Orders.
   * Order Item Date: All Time. Click Apply.
4. **Add Groupings & Columns**:
   * **Group Rows**: Supplier ID, Actual Delivery Date, Purchase Order: Purchase Order ID.
   * **Columns**: Product ID: Product ID, Product ID: Product Name, Order Item: Quantity Ordered, Order Item: Quantity Received, Order Item: Amount.
5. **Save Report**:
   * Click Save & Run.
   * **Report Name**: Complete Purchase Details Report.
   * Click Save.

****

**Dashboards**

**Use Case:**

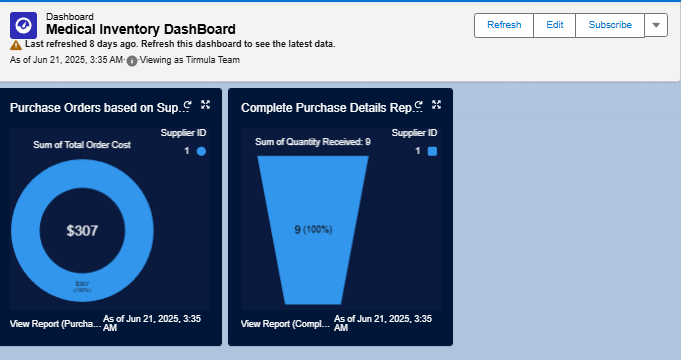
Dashboards provide visual representations of key data from your reports, offering a quick overview of critical metrics within the Medical Inventory Management System.

**Activity 1: Create "Medical Inventory DashBoard"**

1. **Navigate to Dashboards**:
   * From the App Launcher, select the Medical Inventory Management application.
   * Click on the Dashboards tab.
2. **Create New Dashboard**:
   * Click New Dashboard.
   * **Dashboard Name**: Medical Inventory DashBoard.
   * Click Create.
3. **Add Widget (Component)**:
   * Click +Widget.
   * Select the report: Purchase Orders based on Suppliers.
   * Choose a desired chart type (e.g., Bar Chart, Pie Chart) or table for data visualization.
   * Click Add.
4. **Save Dashboard**:
   * Click Save.

**Activity 2: View Dashboard**

1. **Navigate to Dashboards**:
   * From the App Launcher, search for and select the Medical Inventory Management App.
   * Click on the Dashboards Tab.
2. **Open Dashboard**:
   * Click on Medical Inventory DashBoard from the list to view its graphical representation of records.



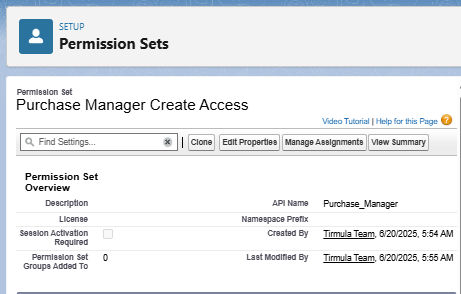
**Users and Permission Set Management**

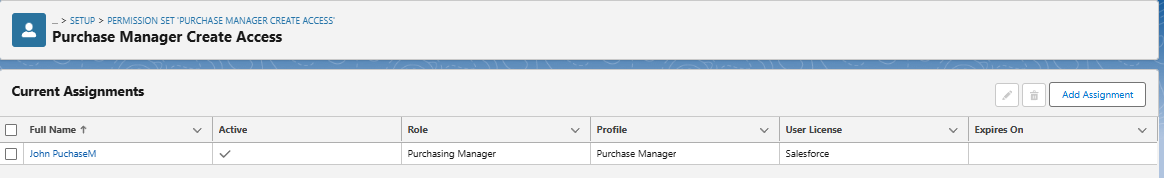
**Permission Sets:**

Permission Sets are used to grant additional permissions and access settings to users, extending their capabilities beyond what is defined in their assigned profile. This provides a flexible way to manage user access without cloning profiles unnecessarily.

**Activity 1: Create "Purchase Manager Create Access" Permission Set**

1. **Navigate to Permission Sets**:
   * From Setup, enter Permission Sets in the Quick Find box.
   * Select Permission Sets.
   * Click New.
2. **Define Permission Set**:
   * **Label**: Purchase Manager Create Access
   * Click Save.
3. **Configure Object Permissions (Order Item)**:
   * From the Permission Set detail page, click Object Settings.
   * Select Order Item.
   * Click Edit.
   * Enable Tab Available and Visible for Order Item.
   * Enable Read and Create under Object Permissions for Order Item.
   * Click Save.
4. **Assign to User**:
   * From the Permission Set detail page, click Manage Assignments.
   * Click Add Assignments.
   * Select the user John PurchaseM.
   * Click Next.
   * Select No Expiration Date.
   * Click Assign.

****

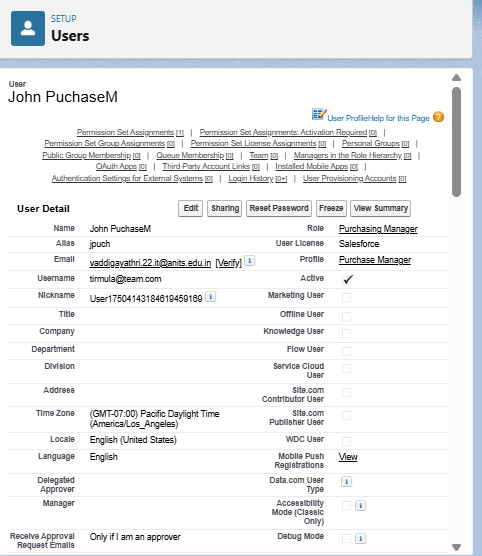
****

**Users:**

Users are the individuals who log into your Salesforce organization to work with the Medical Inventory Management System. Each user requires a unique username, email address, user license, profile, and optionally a role.

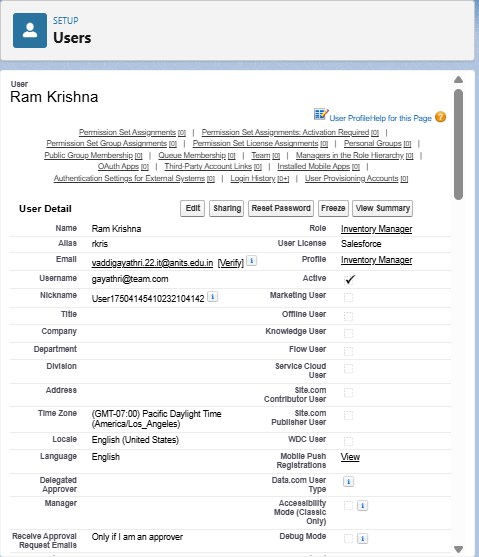
**Activity 1: Create "Purchasing Manager" User**

1. **Navigate to Users**:
   * From Setup, enter Users in the Quick Find box.
   * Select Users.
   * Click New User.
2. **Enter User Details**:
   * **First Name**: John
   * **Last Name**: PurchaseM
   * **Email**: (Your email address)
   * **Username**: (Unique username, e.g., john.purchasem@yourcompany.com)
   * **Role**: Purchasing Manager
   * **User License**: Salesforce
   * **Profile**: Purchase Manager
   * Check Generate new password and notify the user immediately.
3. **Save User**:
   * Click Save.



**Activity 2: Create "Inventory Manager" User**

1. **Navigate to Users**:
   * From Setup, enter Users in the Quick Find box.
   * Select Users.
   * Click New User.
2. **Enter User Details**:
   * **First Name**: Ram
   * **Last Name**: Krishna
   * **Email**: (Your email address)
   * **Username**: (Unique username, e.g., ram.krishna@yourcompany.com)
   * **Role**: Inventory Manager
   * **User License**: Salesforce
   * **Profile**: Inventory Manager
   * Check Generate new password and notify the user immediately.
3. **Save User**:
   * Click Save.



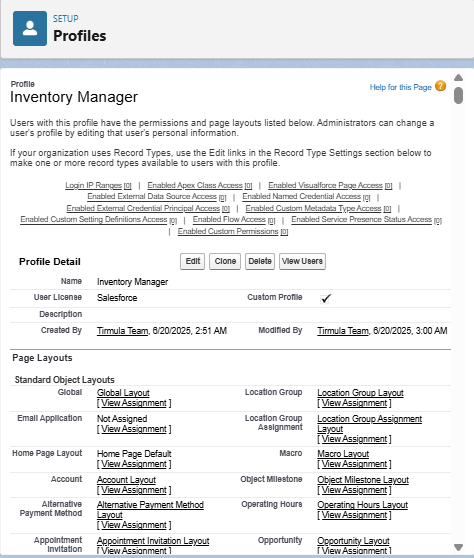
**D. Implementation Activities**

## Profiles

Profiles define a user's permissions for performing different operations in Salesforce, such as creating records, editing records, viewing tabs, and accessing specific apps. They are a fundamental part of controlling data access and user capabilities.

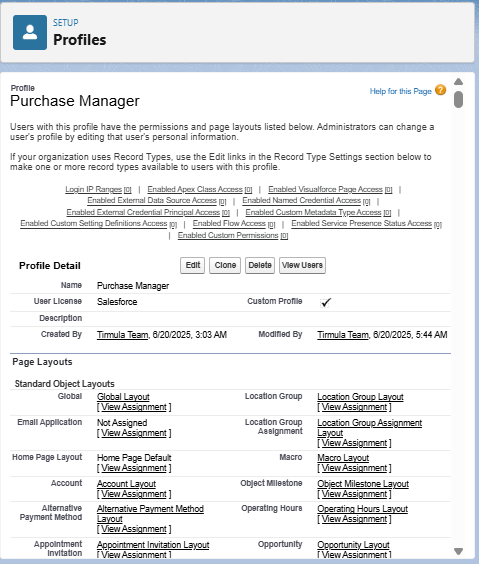
### Activity 1: Create "Inventory Manager" Profile

1. **Navigate to Profiles**:
   * From Setup, enter Profiles in the Quick Find box.
   * Select Profiles.
2. **Clone Standard User Profile**:
   * Find and click Standard User profile.
   * Click Clone.
   * **Profile Name**: Inventory Manager
   * Click Save.
3. **Edit Profile Settings**:
   * While on the Inventory Manager profile page, click Edit.
   * **Custom App Settings**: Set Medical Inventory Management App to Default.
   * **Custom Object Permissions**: Configure permissions for custom objects relevant to an Inventory Manager.
     + For Product\_\_c: Read, Create, Edit, Delete (as needed for managing products)
     + For Inventory\_Transaction\_\_c: Read, Create, Edit, Delete (for managing stock movements)
     + For Purchase\_Order\_\_c: Read (may not need create/edit if Purchase Manager handles it)
     + For Order\_Item\_\_c: Read (similarly)
     + For Supplier\_\_c: Read (similarly)
   * **Password Policies**:
     + User passwords expire in: Never expires
     + Minimum password length: 8
   * Click Save.



### Activity 2: Create "Purchase Manager" Profile

1. **Navigate to Profiles**:
   * From Setup, enter Profiles in the Quick Find box.
   * Select Profiles.
2. **Clone Standard User Profile**:
   * Find and click Standard User profile.
   * Click Clone.
   * **Profile Name**: Purchase Manager
   * Click Save.
3. **Edit Profile Settings**:
   * While on the Purchase Manager profile page, click Edit.
   * **Custom App Settings**: Set Medical Inventory Management App to Default.
   * **Custom Object Permissions**: Configure permissions for custom objects relevant to a Purchase Manager.
     + For Purchase\_Order\_\_c: Read, Create, Edit, Delete (for managing purchase orders)
     + For Order\_Item\_\_c: Read, Create, Edit, Delete (for managing items within orders)
     + For Supplier\_\_c: Read, Create, Edit, Delete (for managing supplier details)
     + For Product\_\_c: Read (for viewing product details when creating orders)
     + For Inventory\_Transaction\_\_c: Read (for viewing transaction history)
   * **Password Policies**:
     + User passwords expire in: Never expires
     + Minimum password length: 8
   * Click Save.



**Role Hierarchy and Record-Level Access Control**

## Roles

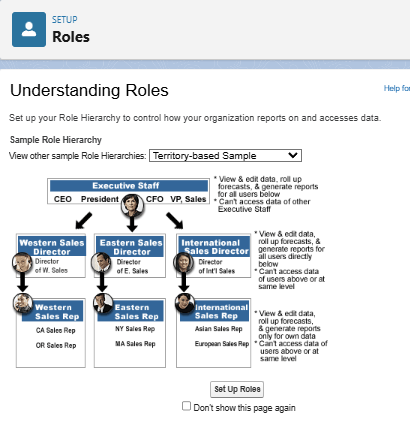
Roles in Salesforce are used to control the level of visibility users have to an organization's data based on their position in the role hierarchy. Users at higher levels in the hierarchy can view, edit, and report on all data owned by or shared with users below them in the hierarchy.

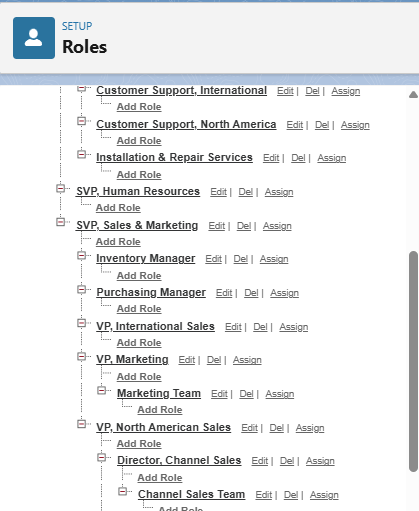
### Activity 1: Create "Purchasing Manager" Role

1. **Navigate to Roles**:
   * From Setup, enter Roles in the Quick Find box.
   * Click Set Up Roles.
2. **Add Role to Hierarchy**:
   * Click Expand All.
   * Locate SVP, Sales & Marketing (or a suitable parent role in your org structure).
   * Click Add Role beneath it.
3. **Define Role Details**:
   * **Label**: Purchasing Manager
   * (Role Name will auto-populate)
   * Click Save.

### Activity 2: Create "Inventory Manager" Role

1. **Navigate to Roles**:
   * From Setup, enter Roles in the Quick Find box.
   * Click Set Up Roles.
2. **Add Role to Hierarchy**:
   * Click Expand All.
   * Locate SVP, Sales & Marketing (or the same parent role as Purchasing Manager for peer relationship, or a different one if a different hierarchy is desired).
   * Click Add Role beneath it.
3. **Define Role Details**:
   * **Label**: Inventory Manager
   * (Role Name will auto-populate)
   * Click Save.



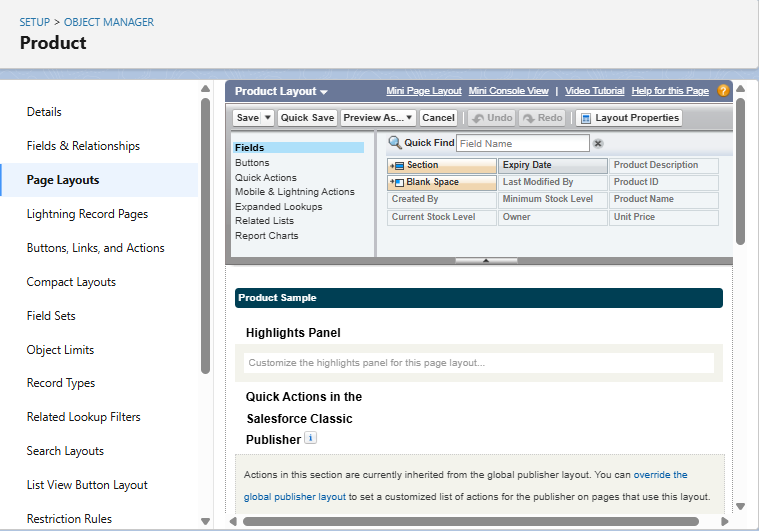


## E. Page Layouts

Page Layouts control the arrangement of fields, custom links, and related lists on record detail and edit pages in Salesforce. They help present information clearly and efficiently to users.

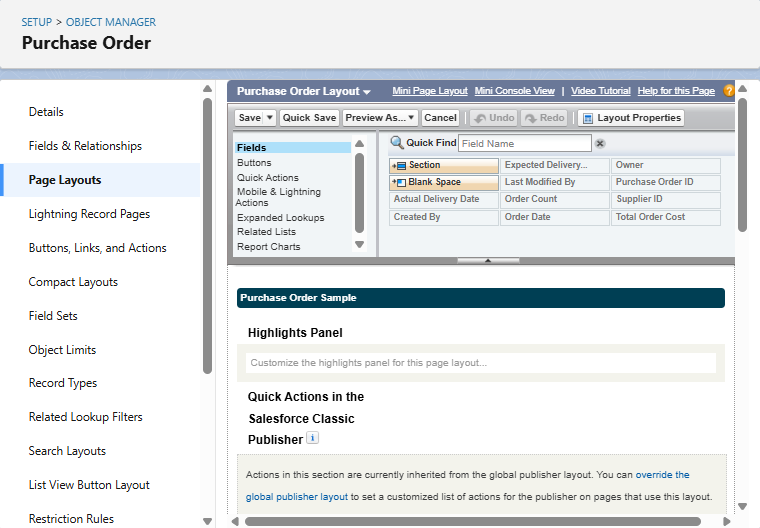
### Activity 1: Edit "Product" Object Page Layout

1. **Navigate to Object Manager**:
   * From Setup, click Object Manager.
   * In the Quick Find box, type Product and click on the Product object.
2. **Access Page Layouts**:
   * Click Page Layouts in the sidebar.
   * Click on Product Layout.
3. **Arrange Fields**:
   * Drag and arrange fields (e.g., Product Name, Product Description, Unit Price, Minimum Stock Level, Current Stock Level, Expiry Date) into a logical order on the layout.
   * Click Save.



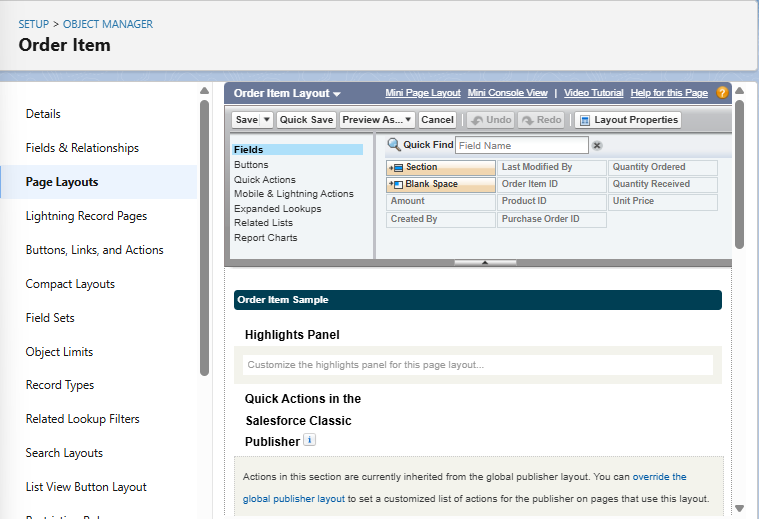
### Activity 2: Edit "Purchase Order" Object Page Layout

1. **Navigate to Object Manager**:
   * From Setup, click Object Manager.
   * In the Quick Find box, type Purchase Order and click on the Purchase Order object.
2. **Access Page Layouts**:
   * Click Page Layouts in the sidebar.
   * Click on Purchase Order Layout.
3. **Arrange Fields and Configure Properties**:
   * Drag and arrange fields (e.g., Supplier ID, Order Date, Expected Delivery Date, Actual Delivery Date, Order Count, Total Order Cost) into a logical order.
   * **Configure Field Properties**:
     + Click the wrench icon (Settings) next to Order Date. Select Required. Click OK.
     + Click the wrench icon (Settings) next to Total Order Cost. Select Read Only. Click OK.
   * Click Save.



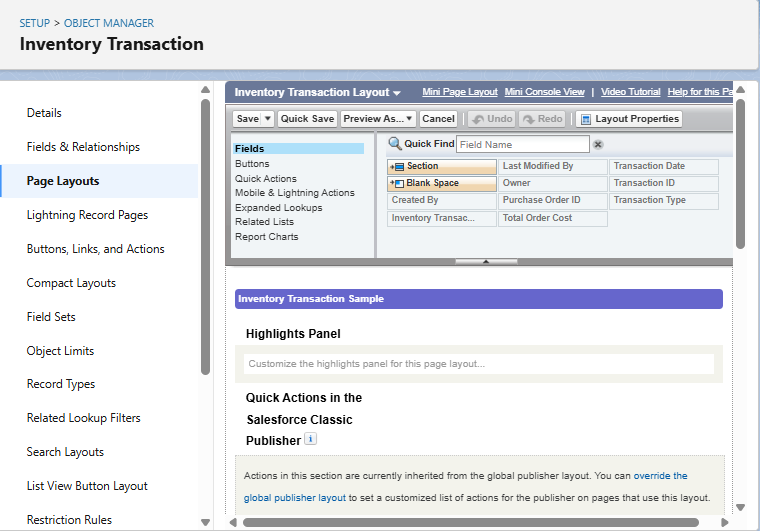
### Activity 3: Edit "Order Item" Object Page Layout

1. **Navigate to Object Manager**:
   * From Setup, click Object Manager.
   * In the Quick Find box, type Order Item and click on the Order Item object.
2. **Access Page Layouts**:
   * Click Page Layouts in the sidebar.
   * Click on Order Item Layout.
3. **Arrange Fields**:
   * Drag and arrange fields (e.g., Product ID, Purchase Order ID, Quantity Ordered, Quantity Received, Unit Price, Amount) into a logical order.
   * Click Save.



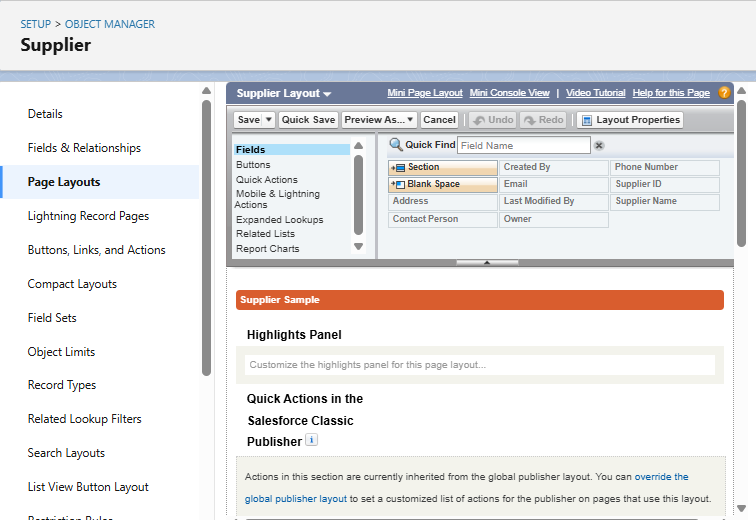
### Activity 4: Edit "Inventory Transaction" Object Page Layout

1. **Navigate to Object Manager**:
   * From Setup, click Object Manager.
   * In the Quick Find box, type Inventory Transaction and click on the Inventory Transaction object.
2. **Access Page Layouts**:
   * Click Page Layouts in the sidebar.
   * Click on Inventory Transaction Layout.
3. **Arrange Fields**:
   * Drag and arrange fields (e.g., Purchase Order ID, Transaction Date, Transaction Type, Total Order Cost - *note: ensure this field's relevance for this object's purpose as discussed before*) into a logical order.
   * Click Save.



### Activity 5: Edit "Supplier" Object Page Layout

1. **Navigate to Object Manager**:
   * From Setup, click Object Manager.
   * In the Quick Find box, type Supplier and click on the Supplier object.
2. **Access Page Layouts**:
   * Click Page Layouts in the sidebar.
   * Click on Supplier Layout.
3. **Arrange Fields**:
   * Drag and arrange fields (e.g., Supplier Name, Contact Person, Phone Number, Email, Address) into a logical order.
   * Click Save.

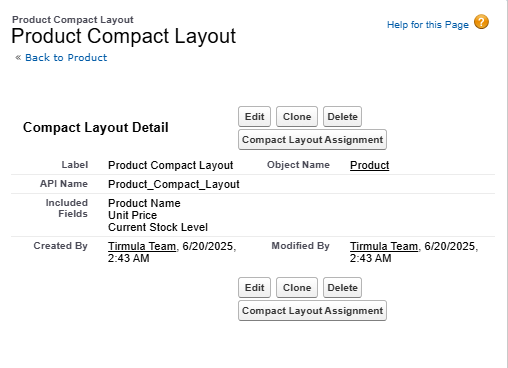


**F.** **Compact Layouts**

Compact Layouts control the fields that appear in the highlight panel at the top of a record page in the Salesforce mobile app and Lightning Experience, as well as in record previews. They are essential for providing a quick glance at key information for each record.

### Activity 1: Create Compact Layout for "Product" Object

1. **Navigate to Product Object**:
   * From Setup, click Object Manager.
   * In the Quick Find box, type Product and click on the Product object.
2. **Create New Compact Layout**:
   * Click Compact Layouts in the sidebar.
   * Click New.
   * **Label**: Product Compact Layout
   * **Selected Fields**: Select Product Name, Unit Price, Current Stock Level.
   * Click Save.
3. **Assign as Primary Compact Layout**:
   * Click Compact Layout Assignment.
   * Click Edit Assignment.
   * From the dropdown, choose Product Compact Layout.
   * Click Save.



### Activity 2: Create Compact Layout for "Purchase Order" Object

1. **Navigate to Purchase Order Object**:
   * From Setup, click Object Manager.
   * In the Quick Find box, type Purchase Order and click on the Purchase Order object.
2. **Create New Compact Layout**:
   * Click Compact Layouts in the sidebar.
   * Click New.
   * **Label**: Purchase Order Compact Layout
   * **Selected Fields**: Select Purchase Order ID, Order Date, Total Order Cost, Supplier ID.
   * Click Save.
3. **Assign as Primary Compact Layout**:
   * Click Compact Layout Assignment.
   * Click Edit Assignment.
   * From the dropdown, choose Purchase Order Compact Layout.
   * Click Save.

### 

### G. The Lightning App Configuration

**Overview:** A Lightning App in Salesforce provides a branded, organized workspace for users to access the objects, tabs, and features relevant to their tasks. It serves as the central hub for the Medical Inventory Management System.

**Activity 1: Create "Medical Inventory Management" Lightning App**

1. **Navigate to App Manager**:
   * From Setup, enter App Manager in the Quick Find box.
   * Select App Manager.
2. **Start New Lightning App**:
   * Click New Lightning App.
3. **Define App Details**:
   * **App Name**: Medical Inventory Management
   * (Optional: Upload an image related to Medical Inventory if desired).
   * Click Next.
4. **Configure App Options**:
   * Leave default selections for App Options and Utility Items. Click Next for each.
5. **Add Navigation Items**:
   * From Available Items, select and move the following to Selected Items:
     + Products
     + Purchase Orders
     + Order Items
     + Inventory Transactions
     + Suppliers
     + Reports
     + Dashboards
   * Click Next.
6. **Assign to Profiles**:
   * From Available Profiles, select System Administrator and move it to Selected Profiles. (Add other relevant profiles like Inventory Manager and Purchase Manager as well for broader access).
   * Click Save & Finish.

