

Kaamadhenu arts and science college

Sathyamangalam-638503

Project Title: Medical Inventory Management

College Code:bru4p

Team ID:NM2025TMID21202

Team Leader : Tharani.P

Email:tharani.cs23@kascathy.ac.in

Team Member : Subhaharini.P

Email:subhaharini.cs23@kascathy.ac.in

Team Member : Srisakthi.G

Email:srisakthi.cs23@kascathy.ac.in

Team Member : Tamilarasi.M

Email:Tamilarasi.cs23@kascathy.ac.in

1.Introduction:

Project Overview:

The **Medical Inventory Management System** is a comprehensive Salesforce-based solution developed to digitize and streamline the management of medical supplies in healthcare facilities. The system addresses common inventory challenges such as manual errors, stockouts, and expired medicines by offering a real-time, automated approach to inventory control.

Designed with usability and efficiency in mind, the system provides features such as barcode scanning, real-time stock level tracking, automatic expiry alerts, and detailed inventory reporting. It supports role-based access, ensuring that pharmacists, doctors, and administrators can securely manage and view inventory according to their specific roles.

This solution enables healthcare providers to:

- Maintain accurate records of stock levels and item details.
- Monitor expiry dates to prevent the usage of outdated medical products.
- Manage supplier and purchase order information efficiently.
- Access inventory data from multiple devices through a user-friendly interface.

By integrating automation into medical inventory operations, this project aims to enhance accuracy, reduce waste, and improve the overall efficiency and safety of healthcare supply management.

Purpose

The primary purpose of the Medical Inventory Management System is to offer healthcare facilities an intelligent, automated platform for effectively managing their medical inventory. The system is developed to:

- Enable accurate and real-time tracking of medical stock levels.
- Prevent critical stockouts of essential medicines and supplies.
- Minimize wastage by promptly identifying items nearing expiration.
- Provide insightful reports and analytics to support informed decision-making.
- Simplify procurement through restock alerts and purchase order tracking.
- Promote patient safety by ensuring the timely availability of necessary medicines.

In essence, this project is intended to improve the operational workflows within healthcare settings, reduce inventory-related risks, and support high-quality healthcare service delivery through efficient supply chain management.

2.Ideation Phase

Brainstorming:

In the **Medical Inventory Management System** project, the ideation phase encourages the generation of a wide range of ideas without initially judging their feasibility. The focus is on quantity over perfection, embracing unconventional or out-of-the-box solutions. These raw ideas form the foundation for robust, scalable, and innovative Salesforce-based solutions. Team collaboration helps filter and refine ideas into practical implementations tailored for medical inventory need

Empathy Map:

The empathy map centers on understanding the core needs, emotions, and challenges faced by key stakeholders, such as pharmacists, inventory managers, and hospital staff who rely on the system.

- **Says:** Users emphasize the importance of accurate stock tracking, expiry date monitoring, low-stock alerts, efficient reporting, and easy data entry.
- **Thinks:** They are concerned about missing expiry dates, manual data errors, and compliance with healthcare regulations.
- **Does:** Regularly monitor inventory levels, update records, generate reports, communicate with suppliers, and prepare for audits.
- **Feels:** Anxious about shortages, frustrated with complex systems, relieved when alerts work properly, and confident when the system is efficient.

This empathy map helps align the solution to the real-world pain points of users, resulting in increased system usability, trust, and satisfaction

Problem Statement:

While building the Medical Inventory Management System using Salesforce, users often face difficulties in generating accurate and dynamic reports due to:

- Complex data relationships

- Limited native customization options

These limitations can lead to:

- Incomplete inventory visibility
- Delayed expiry tracking
- Missed stock alerts

Such challenges impact decision-making, regulatory compliance, and patient safety. Therefore, there is a critical need for a **streamlined, automated, and real-time reporting system** within Salesforce that ensures accurate tracking, timely notifications, and reliable inventory management.

3. Requirement Analysis:

Customer Journey map:

The customer journey map outlines the key stages a typical user (e.g., pharmacist, hospital staff) goes through when interacting with the Salesforce-based inventory system:

- 1. Awareness & Consideration:**
Users identify the limitations of manual inventory methods and explore the system's benefits in improving accuracy and automation.
- 2. Onboarding & Initial Use:**
Users are introduced to the Salesforce platform, undergo brief training, and begin using features like stock entry, reporting, and expiry alerts. Some initial learning challenges are common.
- 3. Issue Resolution & Support:**
Users may face issues like report generation errors or navigation difficulties and rely on support for resolution. Timely help builds user trust.
- 4. Adoption & Loyalty:**
As users gain confidence, they depend more on the system. Consistent performance and continuous improvements can lead to high satisfaction and advocacy for the platform.

Key Insights:

Seamless onboarding, intuitive interface, timely support, and reliable real-time data are essential to maximize user satisfaction and system adoption.

Solution Requirements:

1. Functional Requirements

- Real-time inventory tracking with batch numbers and expiry date monitoring



- Automated alerts for low stock levels and approaching expiries
- Custom real-time reports for inventory status, audits, and expiry tracking
- Barcode scanning for quick and accurate data entry
- Role-based access control for pharmacists, auditors, and managers
- Automated workflows for restocking and expiry item handling
- Mobile accessibility for updates and reporting
- Optional integration with supplier/vendor systems

2. Non-Functional Requirements

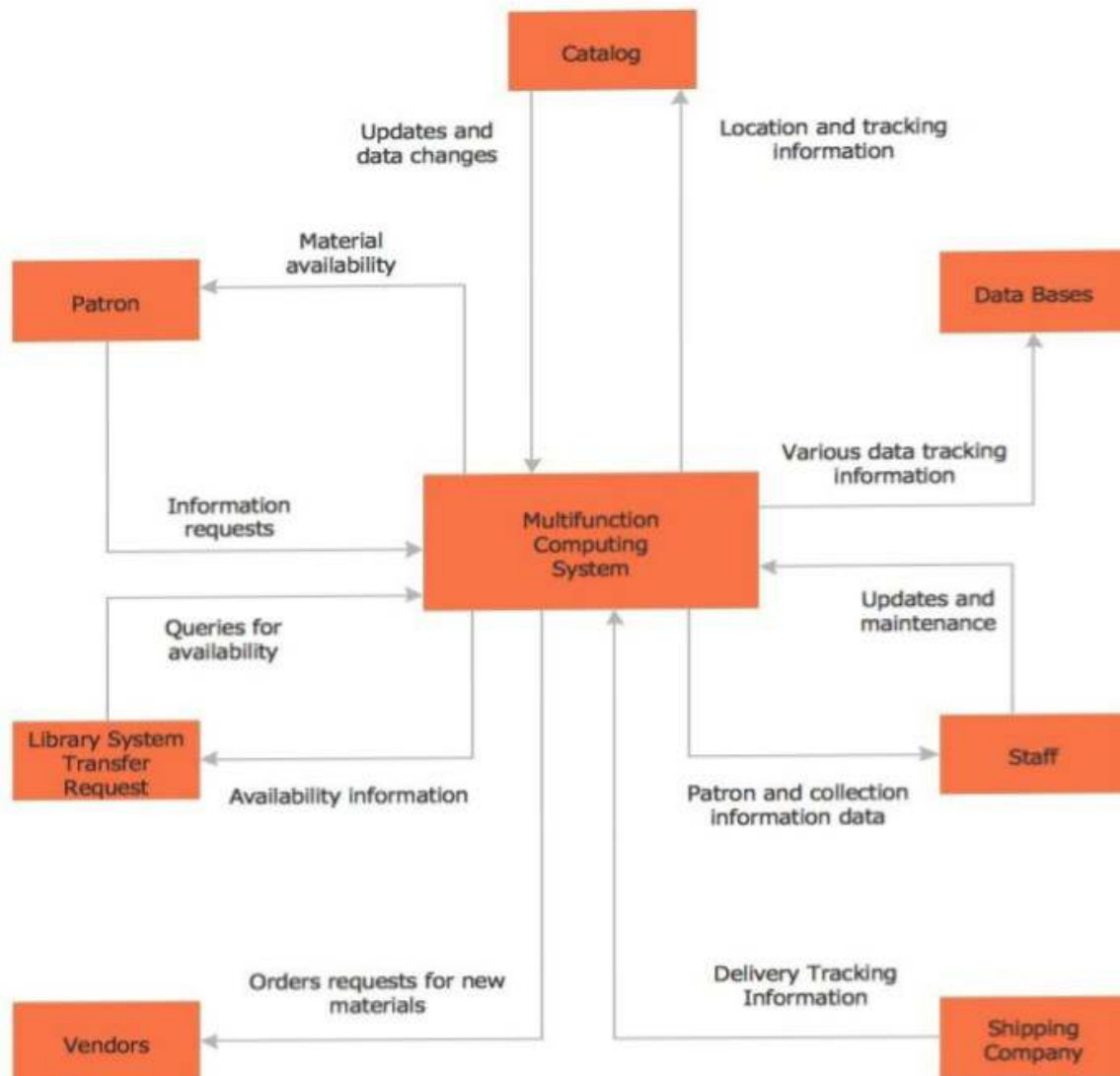
- Fast system performance and real-time response
- User-friendly interface with minimal training needed
- Scalability for expanding inventory and user base
- High system availability and minimal downtime
- Strong data protection and access security
- Easy to maintain and update with minimal technical intervention

Technology stack

Platform: Salesforce Developer Edition

Key Tools & Technologies:

- **Objects & Schema Builder** for data modeling
- **Process Builder / Flow** for workflow automation
- **Apex** for custom logic and backend processes
- **Visualforce / Lightning Components** for UI customization
- **Reports & Dashboards** for analytics
- **Data Loader** for bulk data operations
- **Salesforce Mobile App** for accessibility
- **External APIs (optional)** for supplier integration



4.Project Desing:

Problem Solution Fit:

Problem:

Healthcare facilities often face:

- Stock shortages.
- Expired medicines going unnoticed.
- Manual tracking errors.
- Inefficient inventory reporting.

A Medical Inventory Management System that:

- Tracks stock levels in real-time.
- Sends automated alerts for low stock and nearing expiry.
- Provides accurate reporting and analytics.
- Reduces manual errors with barcode scanning and digital records.

Problem-Solution Fit:

You achieve fit when:

- Healthcare staff find the system easy to use and it saves their time.
- Expiry alerts reduce medicine wastage.
- Stock shortage incidents drop significantly.
- Decision-makers trust the reports and use them for ordering and planning.

Problem-Solution Fit is when medical inventory system genuinely solves the pain points of medicine tracking, stock management, and reporting in healthcare settings, and the users (like pharmacists, nurses, and admins) are happy with the solution.

Proposed Solution:

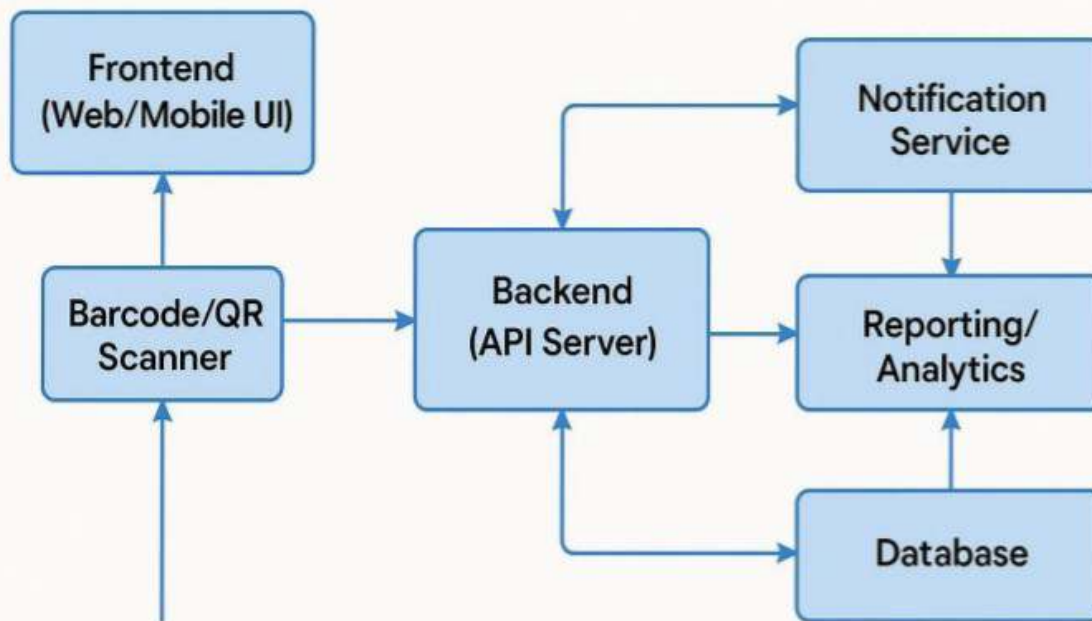
The Medical Inventory Management System is a smart, digital solution designed to address the common challenges faced by healthcare facilities in managing their medical supplies. Traditional manual tracking methods often lead to stock shortages, unnoticed expiry dates, and reporting errors that can impact patient safety and hospital efficiency. This system offers real-time inventory tracking with automated alerts for low stock and upcoming expiries, ensuring timely replenishment and reducing wastage. It integrates barcode scanning to speed up inventory updates and minimize manual entry errors.

A user-friendly interface makes it simple for healthcare staff, including pharmacists, nurses, and inventory managers, to use the system with minimal training. It also provides detailed reporting and analytics, helping administrators make quick, data-driven decisions regarding procurement and inventory levels. The system supports multi-user access with role-based permissions, ensuring that each user interacts with the system according to their responsibilities. This enhances data security and workflow efficiency.

By digitizing the inventory process, the proposed solution significantly improves accuracy, reduces human error, and helps maintain a safe and well-stocked medical supply chain in healthcare facilities.

Solution Architecture:

Solution Architecture



Medical Inventory Management System

The solution architecture of the Medical Inventory Management System is designed to ensure seamless, secure, and real-time management of medical inventories across healthcare facilities. It is structured into three main layers:

1. Frontend Layer:

The user interface is accessible via both web and mobile applications, providing role-based dashboards for pharmacists, doctors, and administrators. It allows users to view stock levels, receive alerts, and generate reports in a user-friendly environment.

2. Backend Layer:

The backend consists of a RESTful API server that manages business logic, inventory operations, and user authentication. It handles all communication between the frontend and the database and integrates services like email and SMS notifications for timely stock and expiry alerts.

3. Database Layer:

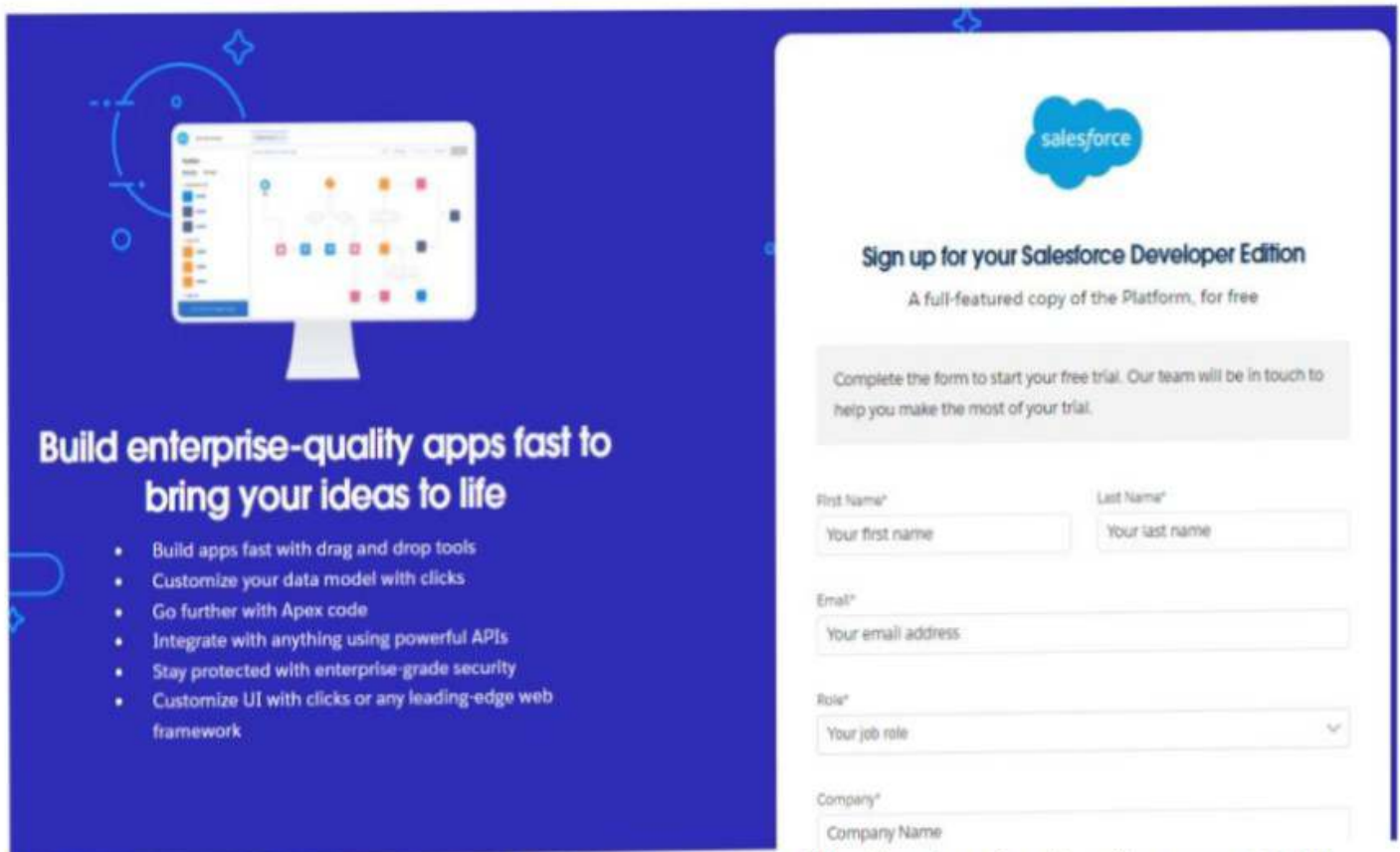
The database securely stores all medical inventory data, including stock details, expiry dates, user roles, and transaction logs. It supports real-time updates and retrievals to ensure accuracy and quick response times.

5. Project Planning & Scheduling

Project Planning:

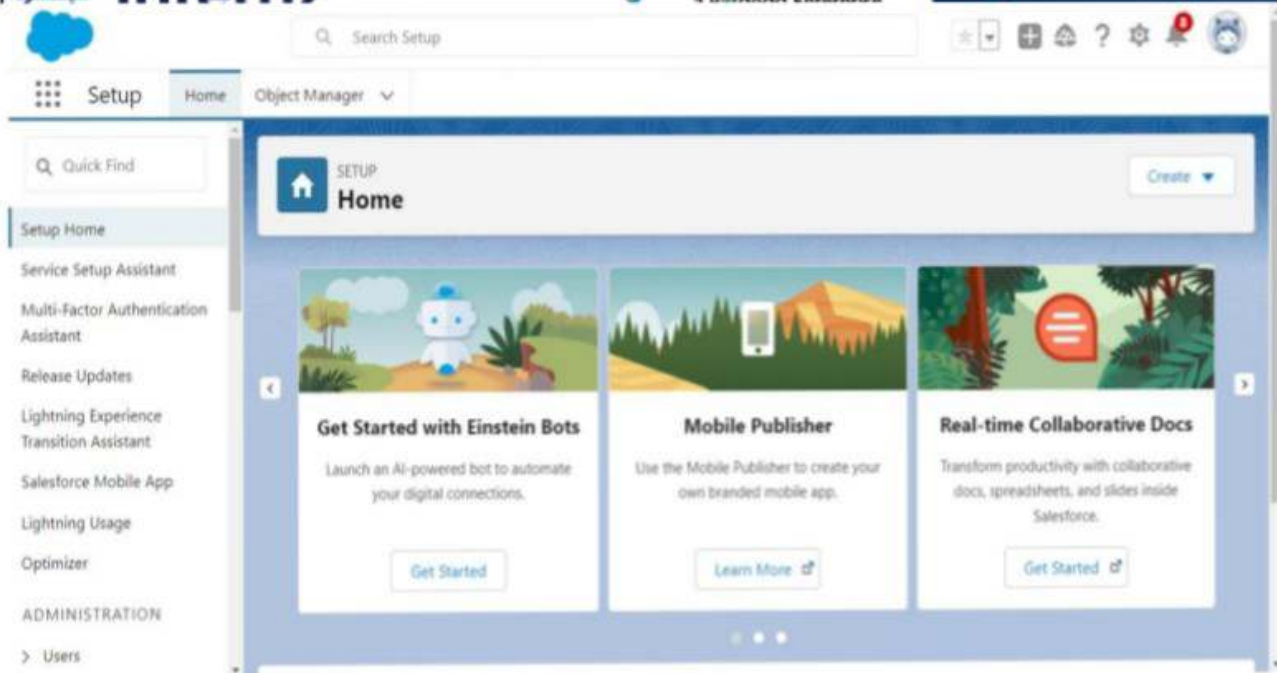
➤ Initiation:

- Creating a developer org in salesforce.
- On the signup form, entered the details.



The image shows a promotional banner for Salesforce Developer Edition on the left and a sign-up form on the right. The banner features a computer monitor displaying a Salesforce interface with various app icons. The text on the banner reads: "Build enterprise-quality apps fast to bring your ideas to life". Below this, a list of bullet points highlights key features: "Build apps fast with drag and drop tools", "Customize your data model with clicks", "Go further with Apex code", "Integrate with anything using powerful APIs", "Stay protected with enterprise-grade security", and "Customize UI with clicks or any leading-edge web framework". The sign-up form on the right is titled "Sign up for your Salesforce Developer Edition" and includes the subtext "A full-featured copy of the Platform, for free". It contains a message: "Complete the form to start your free trial. Our team will be in touch to help you make the most of your trial." The form fields include: "First Name*" (placeholder: "Your first name"), "Last Name*" (placeholder: "Your last name"), "Email*" (placeholder: "Your email address"), "Role*" (placeholder: "Your job role" with a dropdown arrow), and "Company*" (placeholder: "Company Name").

- After Activating the account, we will be having the developer account to create the project.



➤ Planning:

During the planning phase of the **Medical Inventory Management System**, key components and configurations were carefully outlined and implemented to ensure a smooth development and deployment process. The following activities were undertaken:

- **Custom Object Creation:**
Developed essential custom objects including **Product**, **Purchase Order**, **Order Item**, **Inventory Transaction**, and **Supplier** to manage various aspects of medical inventory.
- **Tab Configuration:**
Created custom tabs for the newly developed objects to enable easy navigation and accessibility within the Salesforce interface.
- **Lightning App Setup:**
Designed and configured a dedicated **Lightning App** named *Medical Inventory Management* to group all relevant components, streamlining user access.
- **Field and Layout Customization:**
Added custom fields to each object based on functional requirements and tailored **page layouts** to improve user experience and data input efficiency.
- **Compact Layouts:**
Defined **compact layouts** for key objects to display critical information in highlights panels, enhancing user visibility in Lightning Experience.
- **Validation Rules:**
Implemented **validation rules** in relevant objects (e.g., Employee) to ensure data integrity and prevent invalid entries during record creation or updates.
- **Security & Access Configuration:**
Established **Profiles**, **Roles**, **Users**, and **Permission Sets** to control access based on user roles such as Pharmacist, Admin, and Inventory Manager.
- **Automation and Reporting:**
Developed **Flows** and **Triggers** to automate processes like stock updates and expiry

Created **Reports and Dashboards** to provide real-time insights into inventory status, stock movement, and supplier performance.

➤ Development:

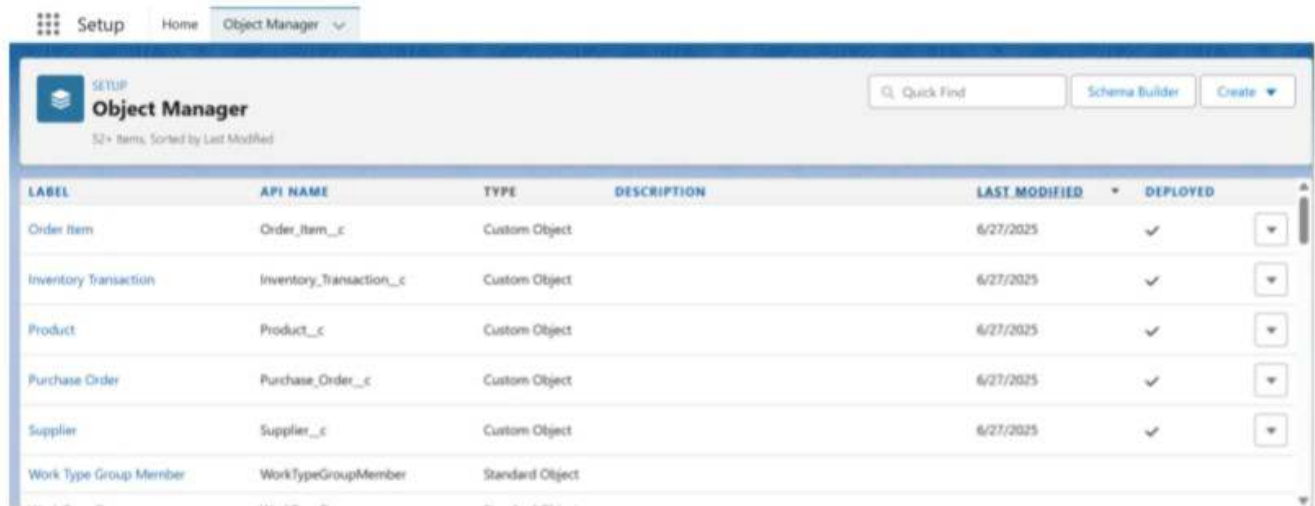
Creation of Objects:

To create a custom object in Salesforce, navigate to **Setup** from the Salesforce homepage. From the **Object Manager** tab, click on **"Create New Custom Object"**. Fill in the necessary details such as the **Label**, **Object Name**, and **Record Name**, and configure options like **Allow Reports**, **Track Field History**, and **Deployment Status**. Once the details are complete, click **Save** to create the object.

This process was repeated to create all the required objects for the **Medical Inventory Management System**. The following custom objects were created to support the application's functionality:

- **Product** – Stores information about medical items available in inventory.
- **Purchase Order** – Tracks orders placed to suppliers for inventory restocking.
- **Order Item** – Represents individual items within a purchase order.
- **Inventory Transaction** – Logs stock movement such as additions, usage, or removals.
- **Supplier** – Maintains details of vendors supplying medical products.

These objects form the core structure of the system and were further customized with fields, relationships, and automation to suit the medical inventory workflow.



The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main header area displays 'SETUP Object Manager' and '52+ Items, Sorted by Last Modified'. A table lists the following objects:

| LABEL | API NAME | TYPE | DESCRIPTION | LAST MODIFIED | DEPLOYED |
|------------------------|--------------------------|-----------------|-------------|---------------|----------|
| Order Item | Order_Item__c | Custom Object | | 6/27/2025 | ✓ |
| Inventory Transaction | Inventory_Transaction__c | Custom Object | | 6/27/2025 | ✓ |
| Product | Product__c | Custom Object | | 6/27/2025 | ✓ |
| Purchase Order | Purchase_Order__c | Custom Object | | 6/27/2025 | ✓ |
| Supplier | Supplier__c | Custom Object | | 6/27/2025 | ✓ |
| Work Type Group Member | WorkTypeGroupMember | Standard Object | | | |



Setup Home Object Manager

Search Setup

Setup > OBJECT MANAGER

Purchase Order

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

Details

Description

API Name
Purchase_Order__c

Custom

✓

Singular Label
Purchase Order

Plural Label
Purchase Orders

Enable Reports
✓

Track Activities

Track Field History

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

Edit Delete

Creation of Tabs:

Tabs in Salesforce are used to make the data stored within custom or standard objects easily accessible to users through the user interface. They serve as navigation elements that allow users to view, create, and manage records related to a specific object.

Tabs are a fundamental part of the Salesforce experience, enabling seamless access to different objects without needing to navigate through complex menus. By creating custom tabs for each of the objects—such as **Product**, **Purchase Order**, **Order Item**, **Inventory Transaction**, and **Supplier**—users can quickly access and interact with the data relevant to the **Medical Inventory Management System**.

Tabs not only improve user efficiency but also enhance the overall usability of the application by organizing key modules in a structured, intuitive layout.

Setup Home Object Manager

Q tab

Feature Settings

Analytics

Tableau

Tableau Embedding

Tableau UAF Claims Definition

User Interface

Console Settings

Loaded Console Tab Limit

Rename Tabs and Labels

Tabs

Setup Tabs

Custom Object Tab

Products

Below is the information for the custom tab. Click Edit to change the custom tab.

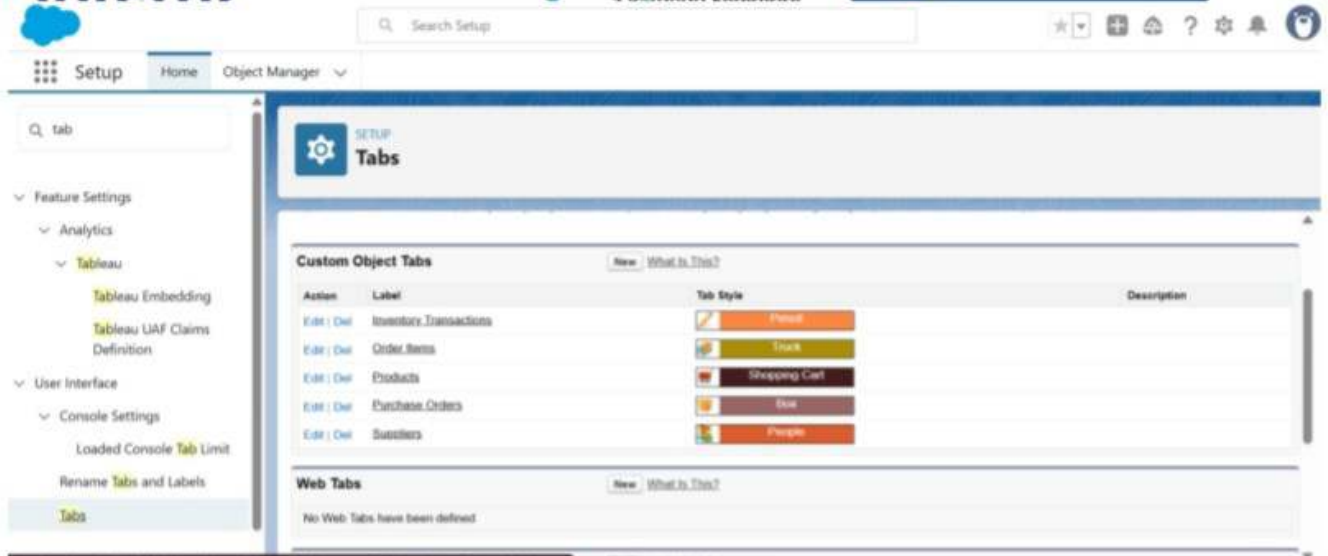
Custom Tab Definition Detail

Edit Delete

| | | | |
|-------------|----------|-------------------------|---------------|
| Tab Label | Products | Tab Style | Shopping Cart |
| Object | Product | Splash Page Custom Link | |
| Description | | | |
| Created By | Admin | Modified By | Admin |

Created: 6/27/2025, 6:28 AM

Modified: 6/27/2025, 6:28 AM



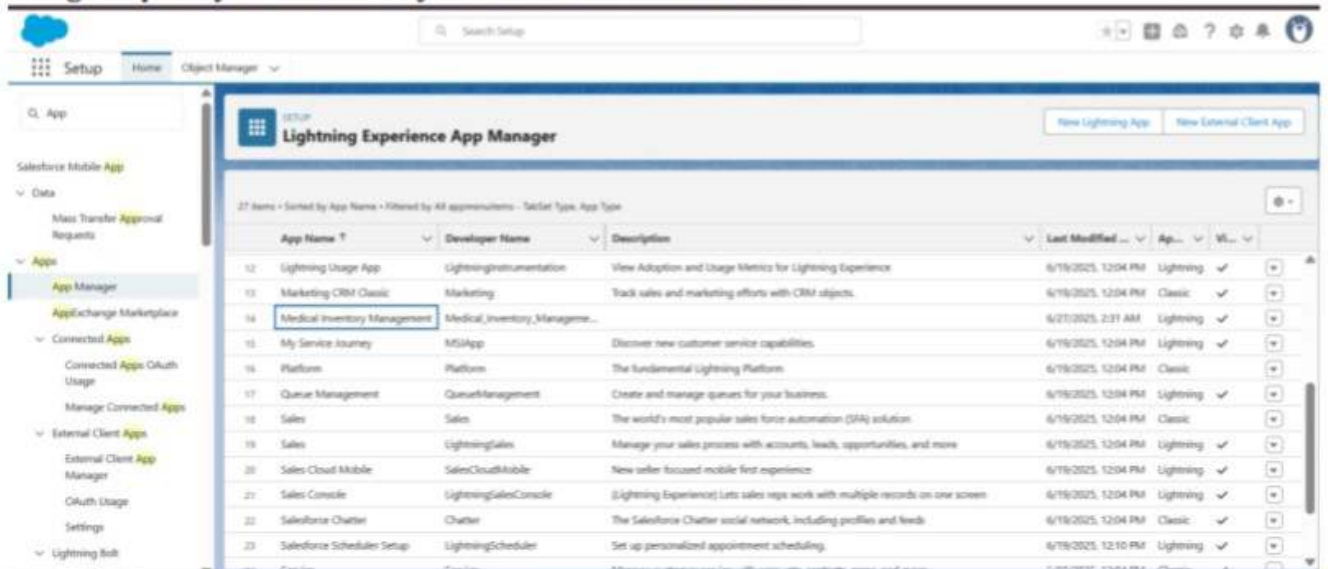
Creating the Lightning App:

A **Lightning App** in Salesforce is a customized collection of components—such as standard and custom objects, tabs, utilities, and tools—designed to streamline workflows and enhance the user experience for a specific business function. Lightning Apps provide a more modern, efficient, and role-specific interface compared to traditional Salesforce apps. For the **Medical Inventory Management System**, a dedicated Lightning App was created to bring together all relevant components into a unified workspace.

Steps to Create the Lightning App:

1. Go to **Setup**, and in the **Quick Find** box, type **App Manager**, then select it.
2. Click **New Lightning App**.
3. Enter the **App Name** as **Medical Inventory Management**.
4. Configure the app settings, including the branding, navigation style, and utility bar (if needed).
5. Add the required **objects and tabs** such as Product, Purchase Order, Inventory Transaction, Order Item, and Supplier.
6. Assign the app to relevant user profiles to ensure appropriate access.

This app serves as the central hub for all medical inventory operations, allowing users to navigate quickly and efficiently between various modules.





Smart
Internz

Creating Fields in Objects:

Creating a Text Field in Product Object:



SMARTBRIDGE

Let's Bridge the Gap

a Valeranda Enterprise



Setup Home Object Manager

Setup > OBJECT MANAGER

Product

Details

Fields & Relationships

4 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------|------------------|--------------------|-------------------|---------|
| Created By | CreatedById | Lookup(User) | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Owner | OwnerId | Lookup(User/Group) | | ✓ |
| Product ID | Name | Text(18) | | ✓ |

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

Step 2: Enter the details

Step 2 of 4

Previous **Next** Cancel

Field Label **Product Name**

Please enter the maximum length for a text field below

Length **255**

Field Name **Product**

Description

Help Text

Required ☒ Always require a value in this field in order to save a record

Unique ☐ Do not allow duplicate values

☐ Treat "ABC" and "abc" as duplicate values (case insensitive)

☐ Treat "ABC" and "abc" as different values (case sensitive)

External ID ☐ Set this field as the unique record identifier from an external system

Auto add to custom report type ☒ Add this field to existing custom report types that contain this entity

Creating a Text area Field in Product Object:

Step 2. Enter the details Step 2 of 4

Previous **Next** Cancel

Field Label: **Product Description** 7

Field Name: **Product_Description** 8

Description:

Help Text:

Required: ☐ Always require a value in this field in order to save a record

Auto add to custom report type: ☒ Add this field to existing custom report types that contain this entity

Default Value: [Show Formula Editor](#)

Use formula syntax. Enclose text and point value API names in double quotes ("the_text"). Include numbers without quotes (25). Show percentages as decimals (0.10), and express date calculations in the standard format (Today) + 7. To reference a field from a Custom Metadata type record use \$CustomMetadata.Type__c.RecordAPIName.Field__c

Created fields in Product Object:

Setup > OBJECT MANAGER > Product

Search Setup

Setup Home Object Manager

Details

Fields & Relationships 8 items, sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

| | FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------------------|---------------------|------------------------|---------------------|-------------------|---------|
| Page Layouts | Created By | CreatedBy | Lookup(126) | | |
| Lightning Record Page | Current Stock Level | Current_Stock_Level__c | Number(18, 0) | | |
| Buttons, Links, and Actions | Last Modified By | LastModifiedBy | Lookup(126) | | |
| Compact Layouts | Minimum Stock Level | Minimum_Stock_Level__c | Number(18, 0) | | |
| Field Sets | Owner | OwnerId | Lookup(Owner Group) | | ✓ |
| Object Links | Product Description | Product_Description__c | Text Area(255) | | |
| Record Types | Product ID | Name | Text(30) | | ✓ |
| Related Lookup Filters | Product Name | Product_Name__c | Text(255) | | |
| Search Layouts | Unit Price | Unit_Price__c | Currency(16, 2) | | |
| List View Button Layout | | | | | |
| Restriction Rules | | | | | |
| Scoping Rules | | | | | |
| Object Access | | | | | |
| Triggers | | | | | |
| Flow Triggers | | | | | |
| Validation Rules | | | | | |
| Conditional Field Formatting | | | | | |



Smart
Internz

Creating Lookup Relationship in Purchase Order Object:



SMARTBRIDGE
Let's Bridge the Gap
a Valanda Enterprise



Purchase Order: New Relationship

Step 2: Choose the related object

Select the other object to which this object is related:

Related To: **Supplier**

Field Label: **Supplier ID**

Field Name: **Supplier_ID**

Description:

Help Text:

Relationship Name: **Purchase Orders**

Required: ☒ Always require a value in this field in order to save a record.

What to do if the lookup record is deleted?: ☒ Don't allow deletion of the lookup record that's part of a lookup relationship.

Auto add to custom report type: ☒ Add this field to existing custom report types that contain this entity.

Created Fields in Purchase Order Object:

Setup > OBJECT MANAGER

Purchase Order

Fields & Relationships

10 Items, Sorted by Field Label

| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------------|---------------------------|-------------------------------------|-------------------|---------|
| Actual Delivery Date | Actual_Delivery_Date__c | Date | | |
| Created By | CreatedById | Lookup(User) | | |
| Expected Delivery Date | Expected_Delivery_Date__c | Date | | |
| Last Modified By | LastModifiedById | Lookup(User) | | |
| Order Count | Order_Count__c | Roll-up Summary (COUNT Order Items) | | |
| Order Date | Order_Date__c | Date | | |
| Owner | OwnerId | Lookup(Owner Group) | | ✓ |
| Purchase Order ID | Name | Text(80) | | ✓ |
| Supplier ID | Supplier_ID__c | Lookup(Supplier) | | ✓ |
| Total Order Cost | Total_Order_Cost__c | Currency(16, 0) | | |

Creating a Unit Price Formula Field in Order Item object:

Step 2: Choose output type Step 2 of 5

Field Label: Unit Price Field Name: Unit_Price

Auto add to custom report type: ☒ Add this field to existing custom report types that contain this entity.

Formula Return Type

☐ None Selected
☐ Checkbox
☒ **Currency** 6
☐ Date
☐ Date/Time
☐ Number
☐ Percent

Select one of the data types below:

Calculate a boolean value.
Example: `TODAY() > CDATE()`

Calculate a dollar or other currency amount and automatically format the field as a currency amount.
Example: `Order_Mtmt * Amount * CDATE`

Calculate a date, for example, by adding or subtracting days to other dates.
Example: `RemDate Date + CDATE(7)`

Calculate a datetime, for example, by adding a number of hours or days to another datetime.
Example: `Next + NOW() + 1`

Calculate a numeric value.
Example: `Percent * 100 + CDATE(1) + 30`

Calculate a percent and automatically add the percent sign to the number.

Order Item
New Custom Field

Step 3: Enter Formula Step 3 of 5

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.

Example: `(Screen Margin * Amount * CDATE) / More Examples`

Simple Formula | **Advanced Formula**

Insert Field: Unit Price (Currency) = Product ID * Sales Price 7

Insert Operator: *

Functions: All Function Categories

ABS
ACOS
ADDMONTHS
AND
ASCE
ASIN
Insert Selected Function

Quick Tip

- Getting Started
- Operations & Functions

Created Fields in Order Item Object:

Setup | Home | **Object Manager** | Search Setup

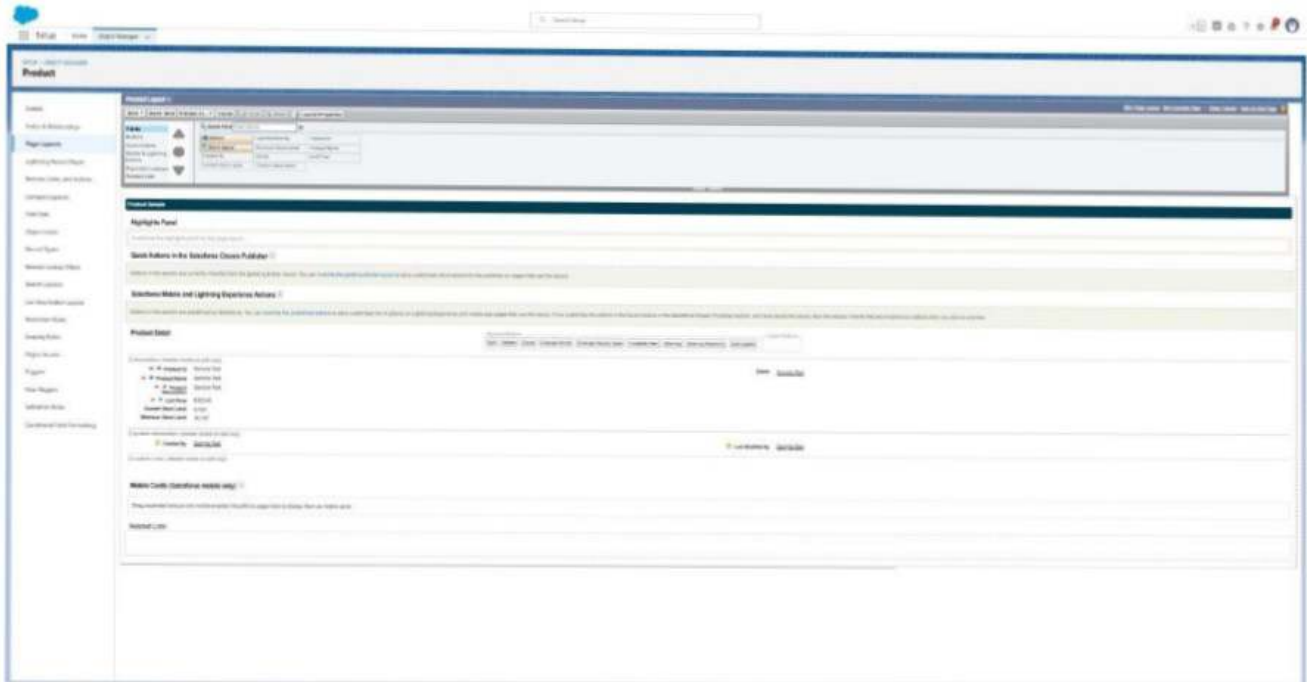
SETUP > OBJECT MANAGER
Order Item

Details | **Fields & Relationships** | 10 Items, Sorted by Field Label | Quick Find | New | Deleted Fields | Field Dependencies | Set History Tracking

| | FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
|------------------------------|-------------------|---------------------|-------------------------------|-------------------|---------|
| Page Layouts | Amount | Amount_c | Formula (Currency) | | |
| Lightning Record Pages | Created By | CreatedBy | Lookup(User) | | |
| Buttons, Links, and Actions | Last Modified By | LastModifiedDate | Lookup(User) | | |
| Compact Layouts | Order Item ID | Name | Text(80) | | ✓ |
| Field Sets | Product ID | Product_ID_c | Lookup(Product) | | ✓ |
| Object Limits | Product Lookup | Product_Lookup_c | Lookup(Product) | | ✓ |
| Record Types | Purchase Order | Purchase_Order_c | Master-Detail(Purchase Order) | | ✓ |
| Related Lookup Filters | Quantity Ordered | Quantity_Ordered_c | Number(18, 0) | | |
| Search Layouts | Quantity Received | Quantity_Received_c | Number(18, 0) | | |
| List View Button Layout | Unit Price | Unit_Price_c | Formula (Currency) | | |
| Restriction Rules | | | | | |
| Scoping Rules | | | | | |
| Object Access | | | | | |
| Triggers | | | | | |
| Flow Triggers | | | | | |
| Validation Rules | | | | | |
| Conditional Field Formatting | | | | | |

Created Fields in Inventory Transaction Object:

Created Page layout in Product object:



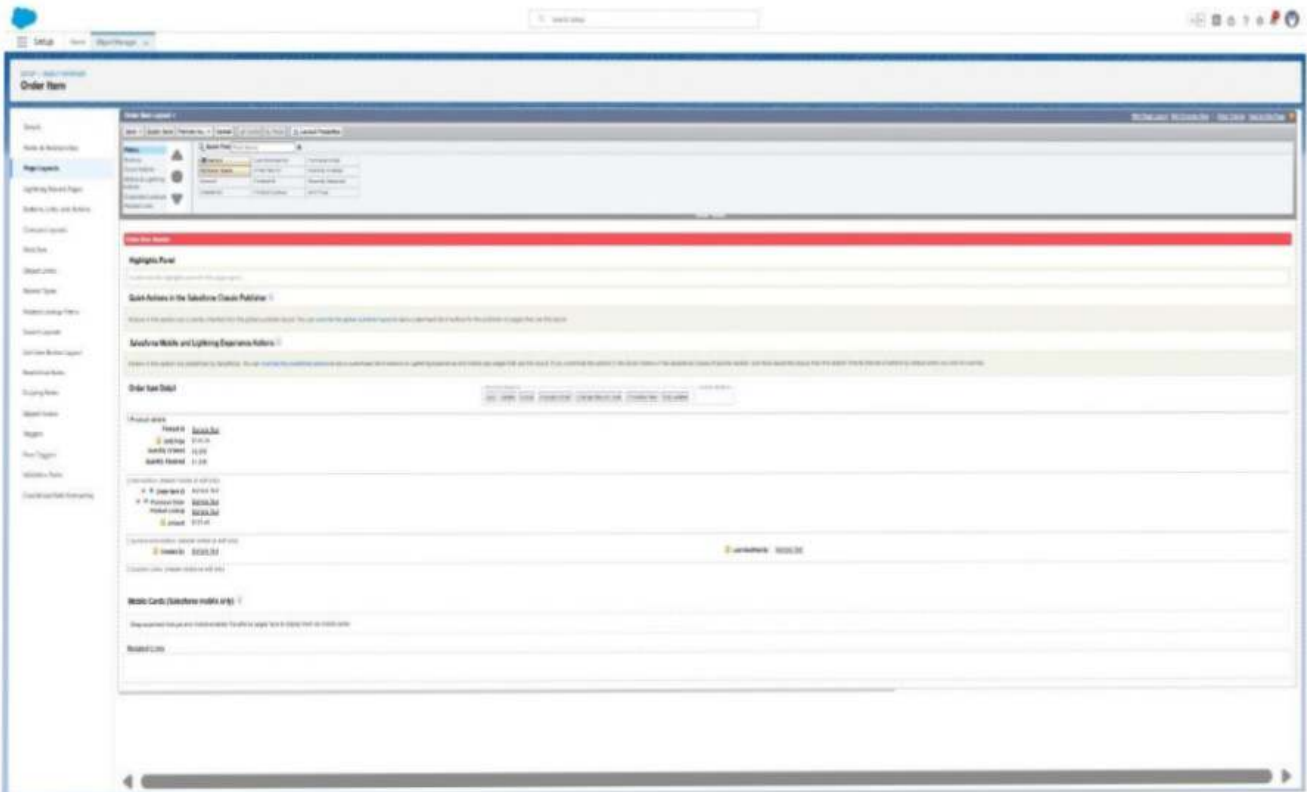
The screenshot shows the Salesforce Product object page layout. The left sidebar contains navigation links: Home, Setup, and a search bar. The main content area is titled "Product" and includes a "Product Detail" section with a table of product information. The table has columns for Product Name, Product Code, Product Type, and Product Status. The table contains one row with the following data:

| Product Name | Product Code | Product Type | Product Status |
|--------------|--------------|--------------|----------------|
| Product A | 12345 | Product Type | Active |

Below the table, there is a "Related Lists" section with a table of related records. The table has columns for Name, Date, and Status. The table contains one row with the following data:

| Name | Date | Status |
|-------------------|------------|--------|
| Related List Item | 2023-10-27 | Active |

Created Page layout in Order Item object:



The screenshot shows the Salesforce Order Item object page layout. The left sidebar contains navigation links: Home, Setup, and a search bar. The main content area is titled "Order Item" and includes a "Order Item Detail" section with a table of order item information. The table has columns for Order Item Name, Order Item Code, Order Item Type, and Order Item Status. The table contains one row with the following data:

| Order Item Name | Order Item Code | Order Item Type | Order Item Status |
|-----------------|-----------------|-----------------|-------------------|
| Order Item A | 12345 | Order Item Type | Active |

Below the table, there is a "Related Lists" section with a table of related records. The table has columns for Name, Date, and Status. The table contains one row with the following data:

| Name | Date | Status |
|-------------------|------------|--------|
| Related List Item | 2023-10-27 | Active |



Smart
Internz



SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

salesforce

PARTNER

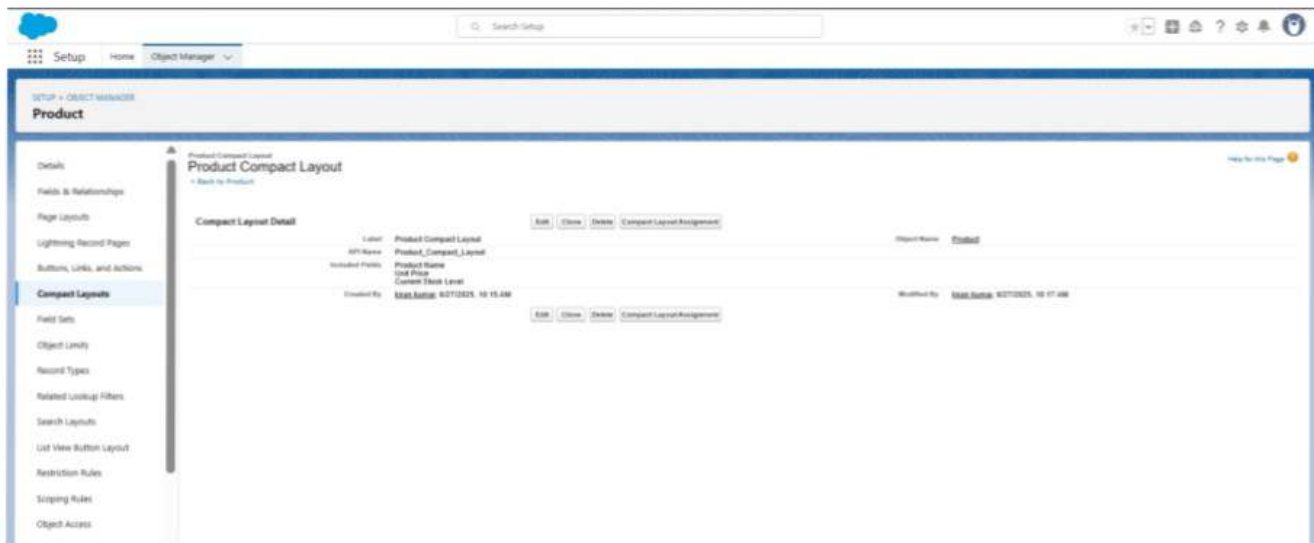
Created Page layout in Inventory Transaction object:

Created Page layout in Supplier object:

Compact Layouts:

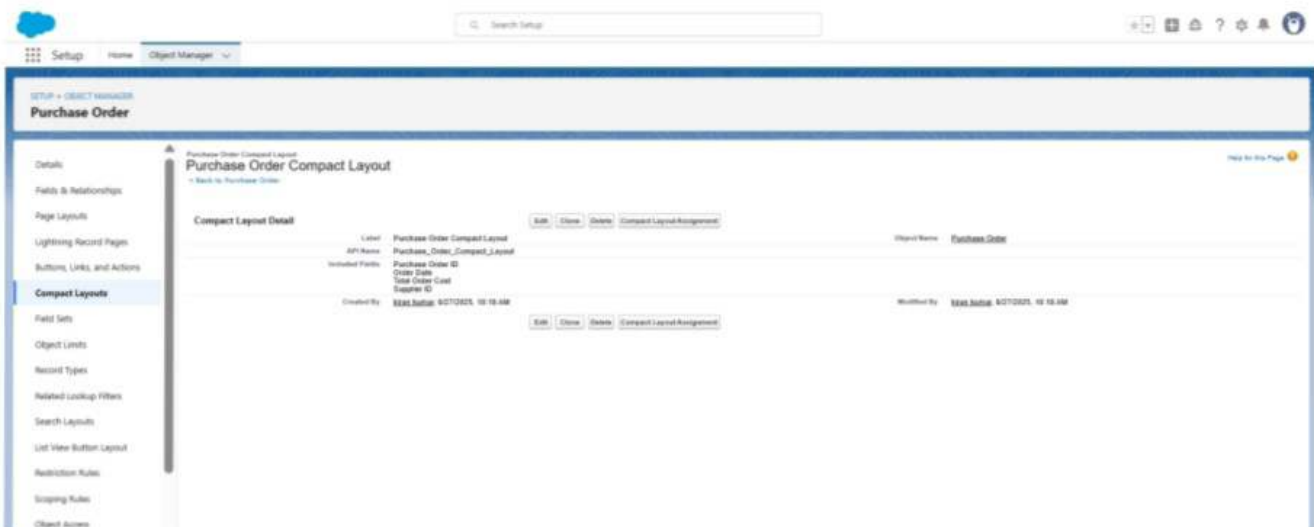
Compact layouts display a record's key fields at a glance, providing important information quickly without needing to open the record.

Created Compact layout in Product object:



The screenshot shows the Salesforce Setup interface for the 'Product' object. The left sidebar lists various configuration options, with 'Compact Layouts' selected. The main content area displays the 'Product Compact Layout' configuration page. It includes a 'Compact Layout Detail' section with fields for 'Label' (Product Compact Layout), 'API Name' (Product_Compact_Layout), and 'Included Fields' (Product Name, Unit Price, Current Stock Level). The 'Created By' field is populated with 'SAS.Sathya' and the 'Created' timestamp is '6/27/2025, 10:15:48 AM'. The 'Modified By' field is also populated with 'SAS.Sathya' and the 'Modified' timestamp is '6/27/2025, 10:17:48 AM'. The page includes buttons for 'Edit', 'Clone', 'Delete', and 'Compact Layout Assignment'.

Created Compact layout in Purchase Order object:



The screenshot shows the Salesforce Setup interface for the 'Purchase Order' object. The left sidebar lists various configuration options, with 'Compact Layouts' selected. The main content area displays the 'Purchase Order Compact Layout' configuration page. It includes a 'Compact Layout Detail' section with fields for 'Label' (Purchase Order Compact Layout), 'API Name' (Purchase_Order_Compact_Layout), and 'Included Fields' (Purchase Order ID, Order Date, Total Order Cost, Supplier ID). The 'Created By' field is populated with 'SAS.Sathya' and the 'Created' timestamp is '6/27/2025, 10:18:48 AM'. The 'Modified By' field is also populated with 'SAS.Sathya' and the 'Modified' timestamp is '6/27/2025, 10:18:48 AM'. The page includes buttons for 'Edit', 'Clone', 'Delete', and 'Compact Layout Assignment'.

creating an Expected Delivery Date Validation rule to Employee Object:

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.

Purchase Order Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Validation Rule Edit
Save Save & New Cancel

Rule Name: Expected_Delivery_Date_Validation

Active: ☒

Description:

Error Condition Formula

Example: `Discount_Percent_c < 0.30`
More Examples...

Display an error if Discount is more than 30%.
If this formula expression is **true**, display the text defined in the Error Message area.

Insert Field
Insert Operator

`(Expected_Delivery_Date_c - Order_Date_c) > 7`

Functions

-- All Function Categories --
ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN

Quick Tips

- Operators & Functions

Created Validation Rule in Purchase Order object:


Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Purchase Order

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts

Purchase Order Validation Rule
[Back to Purchase Order](#)

Validation Rule Detail

Rule Name: Expected_Delivery_Date_Validation

Error Condition Formula: (Expected_Delivery_Date__c - Order_Date__c) > 7

Error Message: The Expected Delivery Date should not exceed 7 days.

Description:

Created By: Admin.Bhatia: 6/27/2025, 10:20 AM

Active: ☒

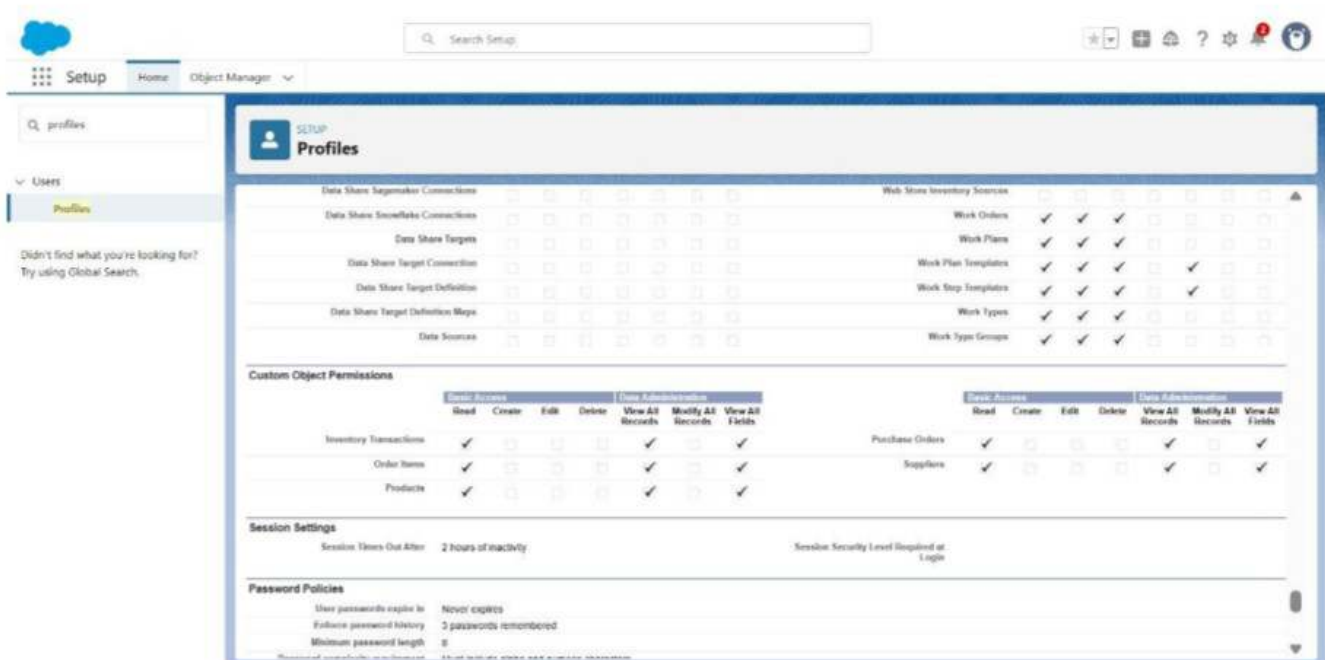
Error Location: Top of Page

Modified By: Admin.Bhatia: 6/27/2025, 10:20 AM

Profiles:

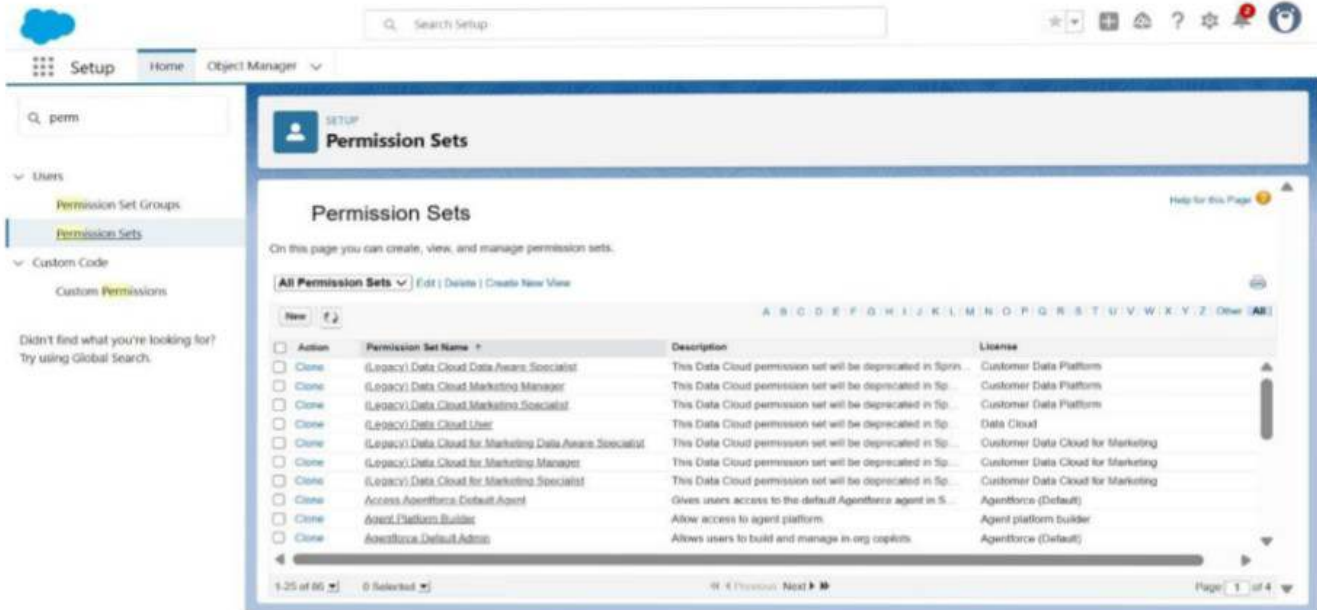
Profiles in Salesforce are fundamental to the platform's security model, defining what users can do within the organization. Profiles control a user's permissions to objects, fields, tabs, apps, and other settings. Each user in Salesforce must be assigned a profile, and the profile assigned to a user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

Created Profiles:

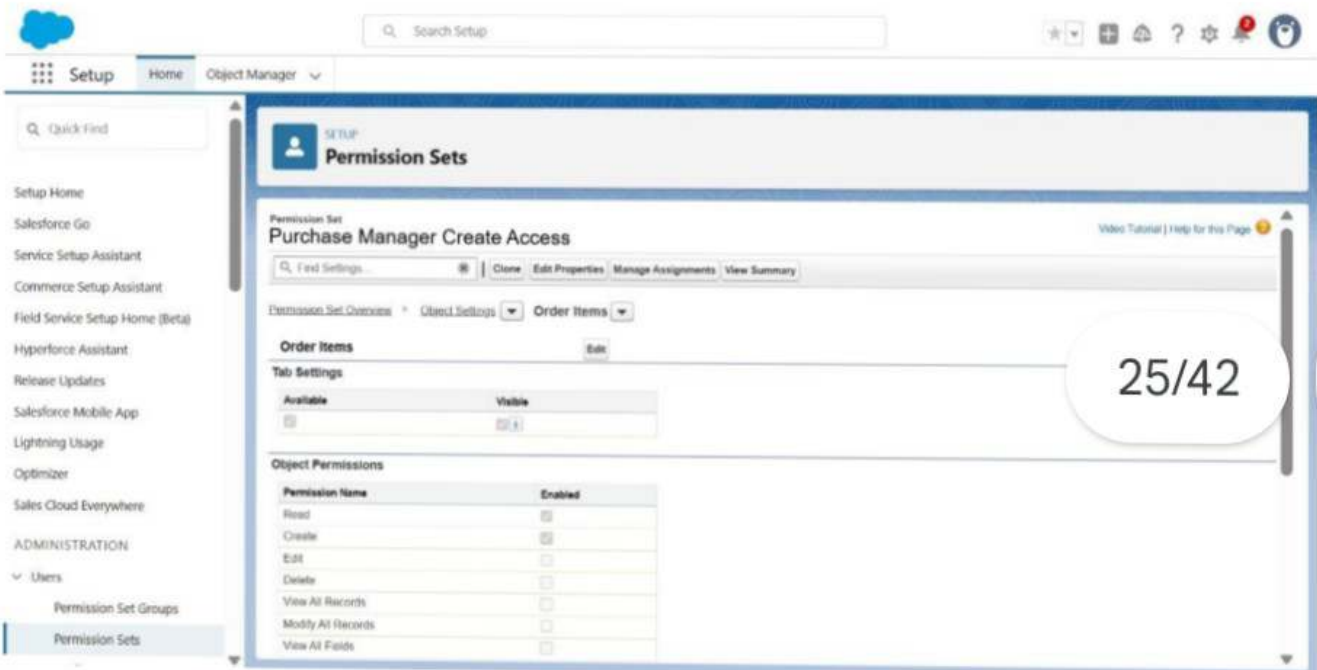


Created Roles:

for providing specialized permissions to specific users without the need to create multiple profiles.



Created Permission set:



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.

Created Flow to update the Actual Delivery Date:

Setup Home Object Manager

Search Setup

Flows

Process automation

Flows

Identity

Login Flows

Didn't find what you're looking for? Try using Global Search.

Flow Definitions

All Flows 1

38 items • Sorted by Flow Label • Filtered by All flow definitions • Updated a minute ago

| Flow Label | Process Type | Active | Template | Package State | Package No. | Last Modified By | Last Modified Date |
|---|---------------------------------|-------------------------------------|-------------------------------------|-------------------|-------------|------------------|--------------------|
| Actual Delivery Date Updating | Autolaunched Flow | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Unmanaged | | Kiran Kumar | 6/28/2023, 9:26 AM |
| Add or Modify Service Appointment Attendees | Salesforce Scheduler Flow | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Managed-Installed | | | |
| Approvals Workflow: Evaluate Approval Requests | Screen Flow | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |
| Approvals Workflow: Process Approval Submission | Screen Flow | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Managed-Installed | | | |
| Authentication Provider User Registration | Identity User Registration Flow | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |
| Basic Approval Request | Flow Orchestration for CMS | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |
| Book Appointment from Invitation | Salesforce Scheduler Flow | <input checked="" 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 | | | |
| Chats Routed to Agents and Queues | Omni-Channel Flow | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |
| Chats Routed to Agents with the Right Skills | Omni-Channel Flow | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |
| Check Flow API Name | Autolaunched Flow | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Managed-Installed | | | |
| Check Service Plan Eligibility | Autolaunched Flow | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |
| Close Change Request & Related Issues | Screen Flow | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |
| CMS Check whether Any Step is Completed | Evaluation Flow | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Managed-Installed | | | |

Flow Builder Actual Delivery Date Updating - V1

Last saved on 6/28/2023, 03:56 PM

Record Triggered Flow

Start

Object: Purchase Order

Trigger: A record is created or updated

Optimize for: Fast Field Updates

Open Flow Trigger Explorer for Purch...

Get Purchase Record

Get Records

Assignment

Assignment

Updating Purchasing Order

Update Records

End

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.



Smart
Internz



SMARTBRIDGE

Let's Bridge the Gap

a Veranda Enterprise



Created a Trigger to Calculate total amount on Order Item:

New Apex Trigger

Name:

CalculateTotalAmountTrigg

sObject:

Order_Item__c

Submit

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.

Created Reports:

Medical Inventory ...

Products

Purchase Orders

Order Items

Inventory Transactions

Supplier

Reports

Dashboards

Report: Purchase Orders

Purchase Orders based on Suppliers

Enable Field Editing

Search

Add Chart

Filter

Refresh

Edit

Total Records

5

Total Order Count

26

Total Total Order Cost

\$48,150

| Supplier ID | Purchase Order: Purchase Order ID | Order Count | Total Order Cost |
|------------------|-----------------------------------|-------------|------------------|
| Supplier 001 (4) | Purchase-0001 (1) | 3 | \$2,075 |
| | Purchase-0002 (1) | 2 | \$3,250 |
| | Purchase-0003 (1) | 3 | \$7,000 |
| | Purchase-0004 (1) | 4 | \$9,500 |
| Supplier-002 (1) | Purchase-0005 (1) | 14 | \$26,325 |
| Total (5) | | 26 | \$48,150 |

Row Counts

Detail Rows

Subtotals

Grand Total

- Setup Home
- Salesforce Go
- Service Setup Assistant
- Commerce Setup Assistant
- Field Service Setup Home (Beta)
- Hyperforce Assistant
- Release Updates
- Salesforce Mobile App
- Lightning Usage
- Optimizer
- Sales Cloud Everywhere
- ADMINISTRATION
- Users
 - Permission Set Groups
 - Permission Sets**

SETUP Permission Sets

Permission Set Purchase Manager Create Access

[Video Tutorial](#) | [Help for this Page](#)
 | [Clone](#) | [Edit Properties](#) | [Manage Assignments](#) | [View Summary](#)
[Permission Set Overview](#) > [Object Settings](#) > [Order Items](#)

Order Items

[Edit](#)

Tab Settings

| Available | Visible |
|--------------------------|-------------------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Object Permissions

| Permission Name | Enabled |
|--------------------|-------------------------------------|
| Read | <input checked="" type="checkbox"/> |
| Create | <input checked="" type="checkbox"/> |
| Edit | <input type="checkbox"/> |
| Delete | <input type="checkbox"/> |
| View All Records | <input type="checkbox"/> |
| Modify All Records | <input type="checkbox"/> |
| View All Fields | <input type="checkbox"/> |

SETUP > OBJECT MANAGER

Purchase Order

- Details
- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Restriction Rules

Purchase Order Validation Rule

[Help for this Page](#)
[Back to Purchase Order](#)

Validation Rule Detail

[Edit](#) | [Clone](#)

| | | | |
|-------------------------|--|----------------|-------------------------------------|
| Rule Name | Expected_Delivery_Date_Validation | Active | <input checked="" type="checkbox"/> |
| Error Condition Formula | (Expected_Delivery_Date__c - Order_Date__c) > 7 | | |
| Error Message | The Expected Delivery Date should not exceed 7 days. | Error Location | Top of Page |
| Description | | | |

 Created By: [kiran kumar](#), 6/27/2025, 10:20 AM | Modified By: [kiran kumar](#), 6/27/2025, 10:20 AM

[Edit](#) | [Clone](#)



| Open | | | | | |
|------------------|--------------------------|-----------|-----------------|---------|------------|
| Entity Type | Entities | | Related | | |
| Entity Type | Name | Namespace | Name | Extent | Direction |
| Classes | CalculateTotalAmountt... | | ← Order Item... | SObject | References |
| Triggers | | | | | |
| Pages | | | | | |
| Page Components | | | | | |
| Objects | | | | | |
| Static Resources | | | | | |
| Packages | | | | | |

Open ☐ Filter Filter the repository (* = any string) ☐ Hide Managed Packages Refresh

Logs Tests

User

Filter Click here to filter the log list

Flow Builder Actual Delivery Date Updating - V1

Free-Form

Last saved on 6/26/2023, 02:10 PM Active Run Debug View Tests Save As New Version Deactivate

Toolbox

Elements Manager

Interaction (1)

Custom Error

Logic (5)

Assignment

Decision

Loop

Collection Sort

Collection Filter

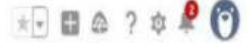
Data (2)

Update Records

Get Records







Setup Home Object Manager

Users

Profiles

Didn't find what you're looking for?
Try using Global Search.

Profiles

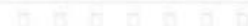
Data Share Target Definition Maps



Work Types



Data Sources



Work Type Groups



Custom Object Permissions

| | Basic Access | | | | Data Administration | | |
|------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | Read | Create | Edit | Delete | View All Records | Modify All Records | View All Fields |
| Inventory Transactions | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Order Items | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Products | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

| | Basic Access | | | | Data Administration | | |
|-----------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | Read | Create | Edit | Delete | View All Records | Modify All Records | View All Fields |
| Purchase Orders | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Suppliers | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Session Settings

Session Times Out After 2 hours of inactivity

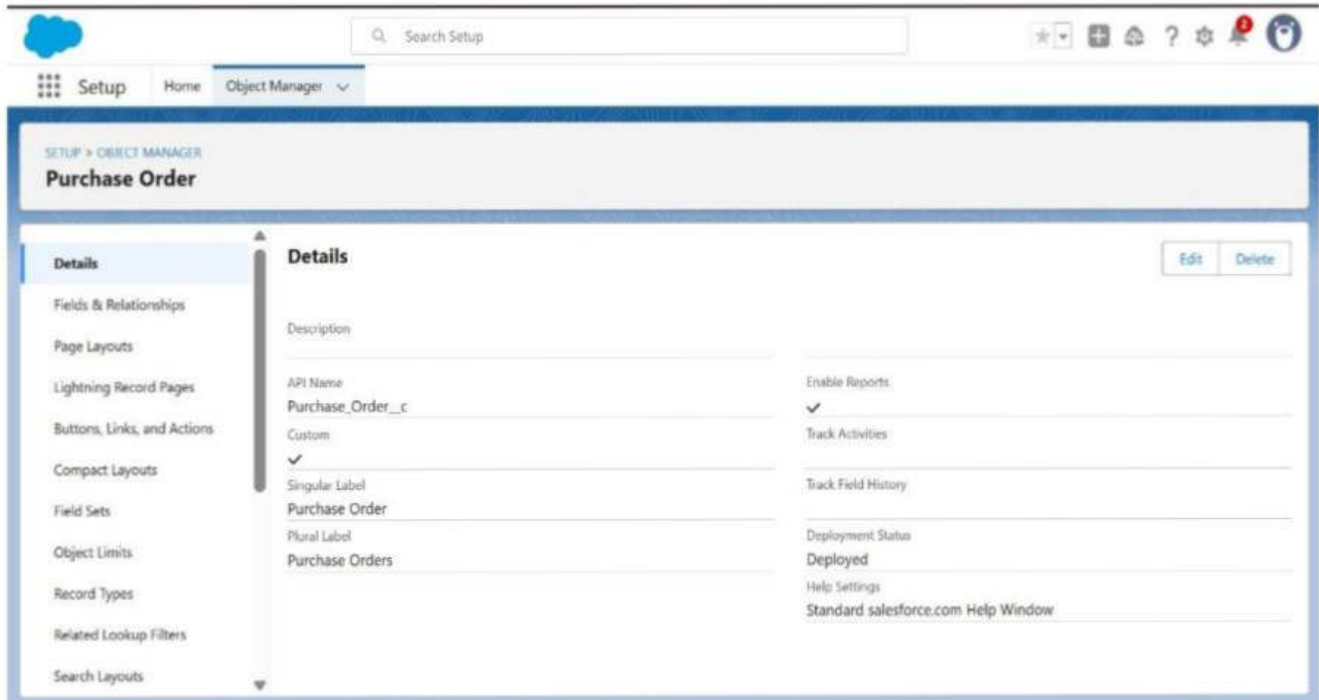
Session Security Level Required at Login

Password Policies

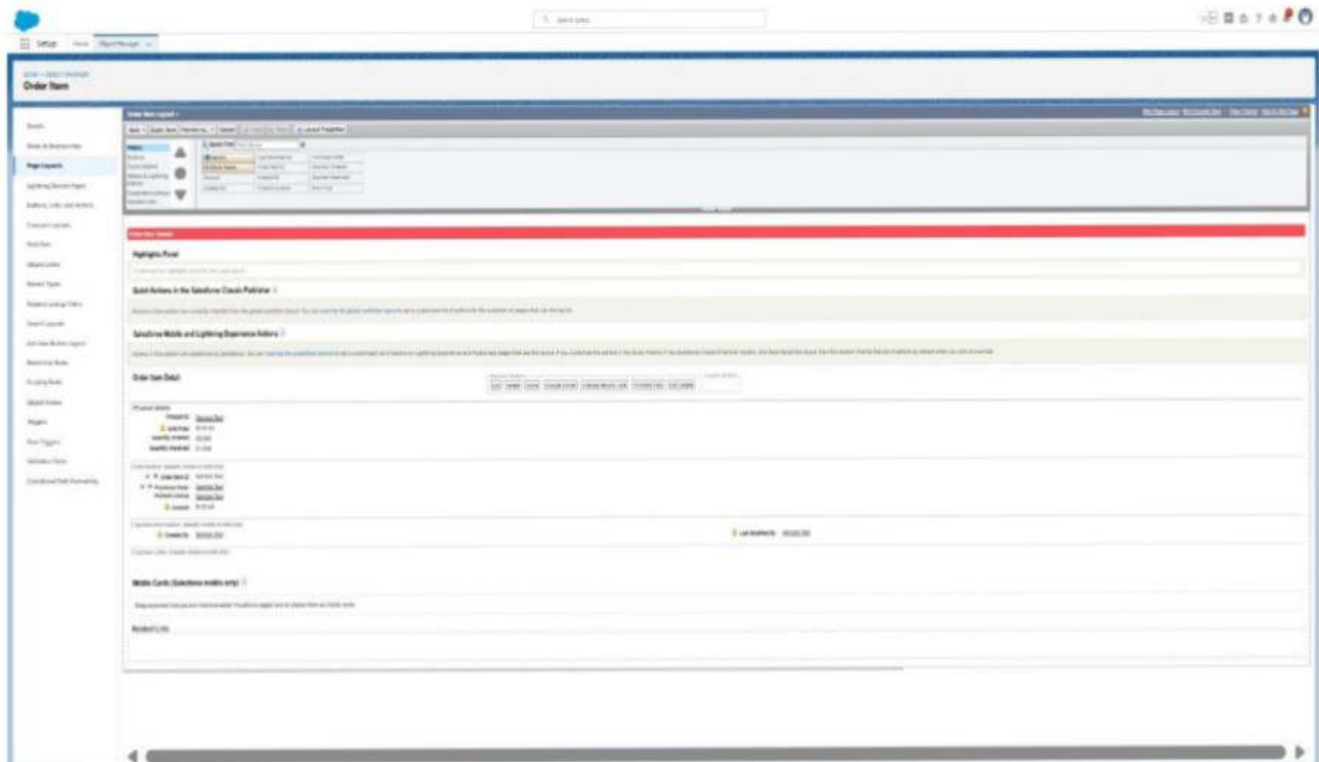
| | |
|---|---|
| User passwords expire in | Never expires |
| Enforce password history | 3 passwords remembered |
| Minimum password length | 8 |
| Password complexity requirement | Must include alpha and numeric characters |
| Password question requirement | Cannot contain password |
| Maximum invalid login attempts | 10 |
| Lockout effective period | 15 minutes |
| Obscure secret answer for password resets | <input type="checkbox"/> |
| Require a minimum 1 day password | <input type="checkbox"/> |

7. RESULTS

Output Screenshots:



This screenshot shows the Salesforce Setup interface for the 'Purchase Order' object. The left sidebar contains a navigation menu with options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main content area displays the 'Details' for the 'Purchase Order' object, including fields like Description, API Name (Purchase_Order__c), Custom (checked), Singular Label (Purchase Order), Plural Label (Purchase Orders), and various system settings like Enable Reports (checked), Track Activities, Track Field History, Deployment Status (Deployed), and Help Settings (Standard salesforce.com Help Window). Edit and Delete buttons are visible in the top right corner.



This screenshot shows the Salesforce Setup interface for the 'Order Item' object. The left sidebar contains a navigation menu with options like Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main content area displays the 'Details' for the 'Order Item' object, including fields like Description, API Name (Order_Item__c), Custom (checked), Singular Label (Order Item), Plural Label (Order Items), and various system settings like Enable Reports (checked), Track Activities, Track Field History, Deployment Status (Deployed), and Help Settings (Standard salesforce.com Help Window). Edit and Delete buttons are visible in the top right corner.

SETUP > OBJECT MANAGER
Inventory Transaction

| Details | Fields & Relationships | | | | |
|------------------------------|---|----------------------|------------------------|-------------------|---------|
| | 3 items Sorted by First Name | | | | |
| | <input type="text" value="Quick Find"/> <input type="button" value="Test"/> <input type="button" value="Default Fields"/> <input type="button" value="Field Dependencies"/> <input type="button" value="Set History Tracking"/> | | | | |
| Fields & Relationships | FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED |
| Page Layouts | Created By | CreatedBy | Lookup(User) | | |
| Lightning Record Pages | Inventory Transaction ID | Name | Text(80) | | ✓ |
| Buttons, Links, and Actions | Last Modified By | LastModifiedBy | Lookup(User) | | |
| Compact Layouts | Owner | OwnerId | Lookup(Person/Group) | | ✓ |
| Field Sets | Purchase Order ID | Purchase_Order_ID__c | Lookup(Purchase Order) | | ✓ |
| Object Limits | Total Order Cost | Total_Order_Cost__c | Formula (Currency) | | |
| Record Types | Transaction Date | Transaction_Date__c | Date | | |
| Related Lookup Filters | Transaction Type | Transaction_Type__c | Picklist | | |
| Search Layouts | | | | | |
| List View Button Layout | | | | | |
| Restriction Rules | | | | | |
| Grouping Rules | | | | | |
| Object Access | | | | | |
| Triggers | | | | | |
| Flow Triggers | | | | | |
| Validation Rules | | | | | |
| Conditional Field Formatting | | | | | |


 SETUP > OBJECT MANAGER
Product
Details

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts

Details

Description

API Name

Product__c

Custom

✓

Singular Label

Product

Plural Label

Products

Enable Reports

✓

Track Activities

Track Field History

Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

Didn't find what you're looking for?
Try using Global Search.

SETUP
Profiles

Data Share Target Definition Maps

Data Sources

Work Types

Work Type Groups

Custom Object Permissions

| | Basic Access | | | | Data Administration | | | | Basic Access | | | | Data Administration | | |
|------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | Read | Create | Edit | Delete | View All Records | Modify All Records | View All Fields | | Read | Create | Edit | Delete | View All Records | Modify All Records | View All Fields |
| Inventory Transactions | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Order Items | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Products | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Purchase Orders | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Suppliers | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

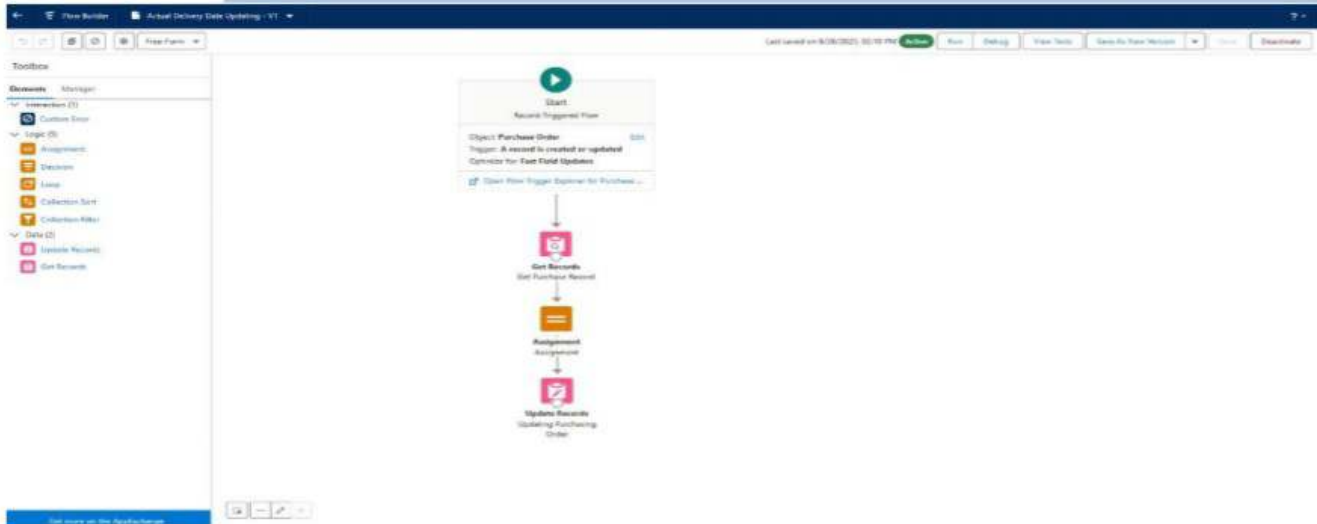
Session Settings

Session Times Out After: 2 hours of inactivity

Session Security Level Required at Login

Password Policies

| | |
|---|---|
| User passwords expire in | Never expires |
| Enforce password history | 3 passwords remembered |
| Minimum password length | 8 |
| Password complexity requirement | Must include alpha and numeric characters |
| Password question requirement | Cannot contain password |
| Maximum invalid login attempts | 10 |
| Lockout effective period | 15 minutes |
| Obscure secret answer for password resets | <input type="checkbox"/> |
| Require a minimum 1 day password | <input type="checkbox"/> |



Medical Inventory ... Products Purchase Orders Order Items Inventory Transactions Supplier Reports Dashboards

Report: Purchase Orders
Purchase Orders based on Suppliers

Enable Field Editing Add Chart Edit

| | | |
|---------------|-------------------|------------------------|
| Total Records | Total Order Count | Total Total Order Cost |
| 5 | 26 | \$48,150 |

| Supplier ID | Purchase Order: Purchase Order ID | Order Count | Total Order Cost |
|------------------|-----------------------------------|-------------|------------------|
| Supplier 001 (4) | Purchase-0001 (1) | 3 | \$2,075 |
| | Purchase-0002 (1) | 2 | \$3,250 |
| | Purchase-0003 (1) | 3 | \$7,000 |
| | Purchase-0004 (1) | 4 | \$9,500 |
| Supplier 002 (1) | Purchase-0005 (1) | 14 | \$26,325 |
| Total (5) | | 26 | \$48,150 |

Row Counts Detail Rows Subtotals Grand Total

8. ADVANTAGES & DISADVANTAGES

Here are the advantages and disadvantages of a Medical Inventory Management Project:

Advantages:

Advantages of the Medical Inventory Management System

- **Improved Accuracy**
- Minimizes human errors in recording stock levels, tracking expiry dates, and processing reorders.
- **Real-Time Inventory Tracking**
- Provides live updates on stock availability and movements, enabling faster and more informed decision-making.
- **Cost Efficiency**
- Prevents overstocking and understocking, helping to reduce waste and control unnecessary procurement costs.
- **Enhanced Patient Safety**
- Automatically flags expired or unavailable medications, ensuring safer treatment practices and reducing risk to patients.
- **Regulatory Compliance**
- Maintains precise records that align with healthcare standards, making it easier to pass audits and meet regulatory requirements.
- **Operational Efficiency**



- Automates repetitive tasks such as restock alerts and inventory updates, freeing up time for healthcare personnel.
- **Data-Driven Decision Making**
- Generates detailed reports and analytics that support accurate forecasting, budget planning, and resource allocation.
-

Disadvantages:

Disadvantages of the Medical Inventory Management System

- **High Initial Investment**
- Requires upfront costs for system setup, including licensing, hardware (e.g., barcode scanners), and staff training.
- **Risk of System Downtime**
- Technical failures or software outages can temporarily disrupt access to inventory data, affecting operations.
- **Integration Challenges**
- Incorporating the system with existing hospital or ERP systems may require additional development and customization efforts.
- **Training Requirements**
- Effective use of the system depends on proper training for pharmacists, nurses, and administrative staff.
- **Cybersecurity Threats**
- As a digital solution handling sensitive data, the system must be secured against potential cyber-attacks and data breaches.
- **Technology Dependence**
- Heavy reliance on the system means that any malfunction could significantly impact daily operations, especially in emergencies.

Implementation & Conclusion

The implementation of a **Medical Inventory Management System** marks a critical advancement in modernizing healthcare operations. It provides a structured, automated, and scalable solution for efficiently managing the diverse and complex inventory requirements of healthcare facilities—including medicines, equipment, and consumables.

By enabling **real-time tracking, automated alerts** for low stock and expiration dates, and **comprehensive reporting**, the system significantly improves operational accuracy and efficiency. One of its most impactful advantages is the **reduction of manual errors**, which commonly lead to serious issues such as stockouts, overstocking, or the inadvertent use of expired medical supplies.

Improved inventory control ensures that the **right medical products are available at the right time**, directly supporting better patient care and safety outcomes.

In addition, the system enhances **regulatory compliance** by maintaining complete and accurate records, which can be easily retrieved during audits and inspections. This capability supports healthcare organizations in meeting both national and international standards. The availability of **data-driven reports and analytics** empowers administrators to make informed decisions regarding procurement, budgeting, and inventory optimization—leading to long-term operational cost savings.

Challenges and Considerations

Despite its many advantages, the implementation of such a system is not without challenges:

- **High initial investment** in software licenses, hardware (e.g., barcode scanners), and comprehensive staff training.
- **Technical integration requirements**, especially when aligning with pre-existing hospital or ERP systems.
- **Cybersecurity concerns**, as the system stores sensitive information related to medical supplies and patient safety.

Final Thoughts

Nonetheless, with proper **planning, system maintenance**, and **ongoing staff training**, these challenges can be effectively managed. Over time, the benefits of the system—such as streamlined operations, improved compliance, and enhanced patient safety—greatly outweigh the initial constraints.

Conclusion

In conclusion, the successful implementation of a **Medical Inventory Management System** is essential for healthcare institutions seeking to improve accuracy, efficiency, and accountability

in inventory operations. It not only optimizes internal workflows but also strengthens patient care quality and organizational compliance. In today's fast-evolving medical landscape, such systems are no longer optional—they are fundamental to delivering high-standard, reliable, and cost-effective healthcare services.

10. FUTURE SCOPE

Future Scope of Medical Inventory Management:

The Medical Inventory Management System has vast potential for future growth and development. As healthcare needs continue to evolve, the system can be expanded and enhanced to provide even greater efficiency, accuracy, and integration with emerging technologies. Some of the key future directions and opportunities include:

1. Integration with IoT (Internet of Things)

- Smart shelves, RFID tags, and IoT-enabled sensors can be used to automatically detect stock levels and environmental conditions like temperature and humidity, which are critical for sensitive medicines and vaccines.
- Real-time automated stock updates without manual intervention.

2. Artificial Intelligence (AI) and Predictive Analytics

- AI can be used for demand forecasting, helping hospitals predict future inventory needs based on patient inflow, disease outbreaks, and historical consumption patterns.
- Predictive alerts for possible shortages, stock expiration, and replenishment scheduling.

3. Mobile Accessibility

- Future systems can offer fully mobile applications for on-the-go inventory checks, approvals, and stock updates, improving accessibility for medical staff.
- Real-time push notifications for urgent stock alerts on mobile devices.

4. Blockchain Integration for Enhanced Security

- Blockchain can be used to track the entire supply chain of medical products, ensuring authenticity and reducing counterfeit risks.
- Immutable transaction records can improve traceability and trust.

5. Cloud-Based Solutions

- Moving the system to the cloud will allow scalability, remote access, and centralized data management.
- It will also reduce infrastructure costs and simplify system maintenance.

6. Automatic Supplier Reordering

- Integration with supplier systems to enable automated purchase orders when stock reaches minimum levels.

- Reduces delays in procurement and ensures continuous availability of critical items.

7. Advanced Reporting and Dashboards

- Future systems can offer customizable dashboards and real-time visual analytics to assist hospital administrators in making faster and more informed decisions.

8. Enhanced Cybersecurity Measures

- As more data becomes digitized, stronger cybersecurity frameworks will be essential to protect sensitive medical inventory and patient data from potential cyber threats.

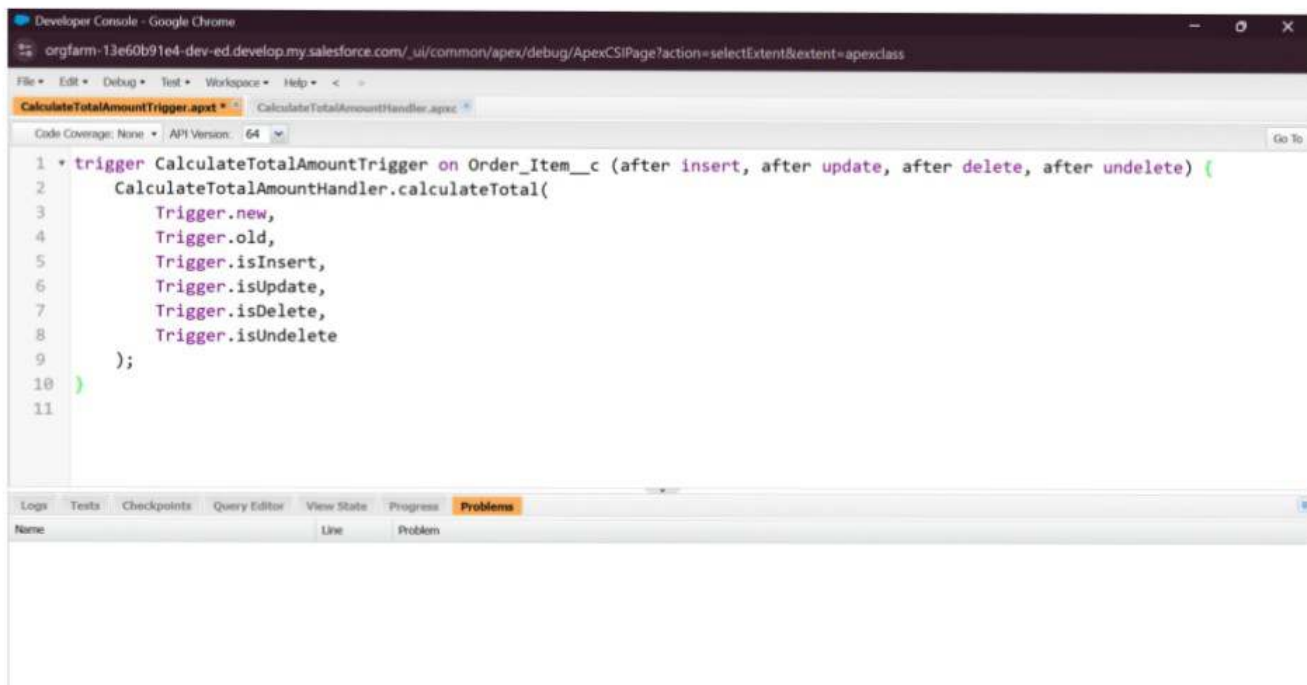
9. Multi-Location and Multi-Branch Inventory Synchronization

- Ability to manage inventory across multiple hospitals, clinics, or warehouses from a single system.
- Supports easy transfer of stock between locations based on demand.

10. Sustainability Tracking

- Future systems can track medical waste, expired items, and promote eco-friendly inventory practices in line with sustainable healthcare goals.

11. APPENDIX



```
Developer Console - Google Chrome
orgfarm-13e60b91e4-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apexclass
File • Edit • Debug • Test • Workspace • Help • < • >
CalculateTotalAmountTrigger.apxt • CalculateTotalAmountHandler.apex
Code Coverage: None • API Version: 64 • Go To
1 * trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {
2     CalculateTotalAmountHandler.calculateTotal(
3         Trigger.new,
4         Trigger.old,
5         Trigger.isInsert,
6         Trigger.isUpdate,
7         Trigger.isDelete,
8         Trigger.isUndelete
9     );
10 }
11
Logs Tests Checkpoints Query Editor View State Progress Problems
Name Line Problem
```

Source code:



Developer Console - Google Chrome

orgfarm-13e60b91e4-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```
1 public class SalesforceCSIPage {
2     public static void SalesforceCSIPageController(SalesforceCSIPage__c sfp, Session__c session, Session__c session2, Session__c session3, Session__c session4) {
3         Session__c session5 = new Session__c();
4         if (session5 != null) {
5             if (session5 != null) {
6                 Session__c session6 = new Session__c();
7                 session6.Session__c = session5;
8                 session6.Session__c = session5;
9             }
10         }
11         if (session5 != null) {
12             Session__c session6 = new Session__c();
13             session6.Session__c = session5;
14             session6.Session__c = session5;
15         }
16         Session__c session6 = new Session__c();
17         if (session6 != null) {
18             if (session6 != null) {
19                 Session__c session7 = new Session__c();
20                 session7.Session__c = session6;
21                 session7.Session__c = session6;
22                 session7.Session__c = session6;
23                 session7.Session__c = session6;
24                 session7.Session__c = session6;
25                 session7.Session__c = session6;
26                 session7.Session__c = session6;
27                 session7.Session__c = session6;
28                 session7.Session__c = session6;
29                 session7.Session__c = session6;
30                 session7.Session__c = session6;
31                 session7.Session__c = session6;
32                 session7.Session__c = session6;
33                 session7.Session__c = session6;
34                 session7.Session__c = session6;
35                 session7.Session__c = session6;
36                 session7.Session__c = session6;
37                 session7.Session__c = session6;
38                 session7.Session__c = session6;
39                 session7.Session__c = session6;
40                 session7.Session__c = session6;
41                 session7.Session__c = session6;
42                 session7.Session__c = session6;
43                 session7.Session__c = session6;
44                 session7.Session__c = session6;
45                 session7.Session__c = session6;
46                 session7.Session__c = session6;
47                 session7.Session__c = session6;
48                 session7.Session__c = session6;
49                 session7.Session__c = session6;
50                 session7.Session__c = session6;
51                 session7.Session__c = session6;
52                 session7.Session__c = session6;
53                 session7.Session__c = session6;
54                 session7.Session__c = session6;
55                 session7.Session__c = session6;
56                 session7.Session__c = session6;
57                 session7.Session__c = session6;
58                 session7.Session__c = session6;
59                 session7.Session__c = session6;
60                 session7.Session__c = session6;
61                 session7.Session__c = session6;
62                 session7.Session__c = session6;
63                 session7.Session__c = session6;
64                 session7.Session__c = session6;
65                 session7.Session__c = session6;
66                 session7.Session__c = session6;
67                 session7.Session__c = session6;
68                 session7.Session__c = session6;
69                 session7.Session__c = session6;
70                 session7.Session__c = session6;
71                 session7.Session__c = session6;
72                 session7.Session__c = session6;
73                 session7.Session__c = session6;
74                 session7.Session__c = session6;
75                 session7.Session__c = session6;
76                 session7.Session__c = session6;
77                 session7.Session__c = session6;
78                 session7.Session__c = session6;
79                 session7.Session__c = session6;
80                 session7.Session__c = session6;
81                 session7.Session__c = session6;
82                 session7.Session__c = session6;
83                 session7.Session__c = session6;
84                 session7.Session__c = session6;
85                 session7.Session__c = session6;
86                 session7.Session__c = session6;
87                 session7.Session__c = session6;
88                 session7.Session__c = session6;
89                 session7.Session__c = session6;
90                 session7.Session__c = session6;
91                 session7.Session__c = session6;
92                 session7.Session__c = session6;
93                 session7.Session__c = session6;
94                 session7.Session__c = session6;
95                 session7.Session__c = session6;
96                 session7.Session__c = session6;
97                 session7.Session__c = session6;
98                 session7.Session__c = session6;
99                 session7.Session__c = session6;
100                session7.Session__c = session6;
101            }
102        }
103    }
104}
```

GitHub:

Project Demo Link: