

Medical Inventory Management

COLLEGE NAME :VET INSTITUTE OF ARTS AND SCIENCE(CO-EDUCATION) COLLEGE.

COLLEGE CODE : brubb

TEAM ID : NM2025TMID25637

TEAM SIZE : 5

TEAM LEADER : KAVINNESH R V

Email: Kavinneshrv23cse@vetias.ac.in

Team Member 1: GOKUL KANNAN V

Email: gokulkannanv23cse@vetias.ac.in

Team Member 2: ASHWANTH N K

Email:ashwanthnk23cse@vetias.ac.in

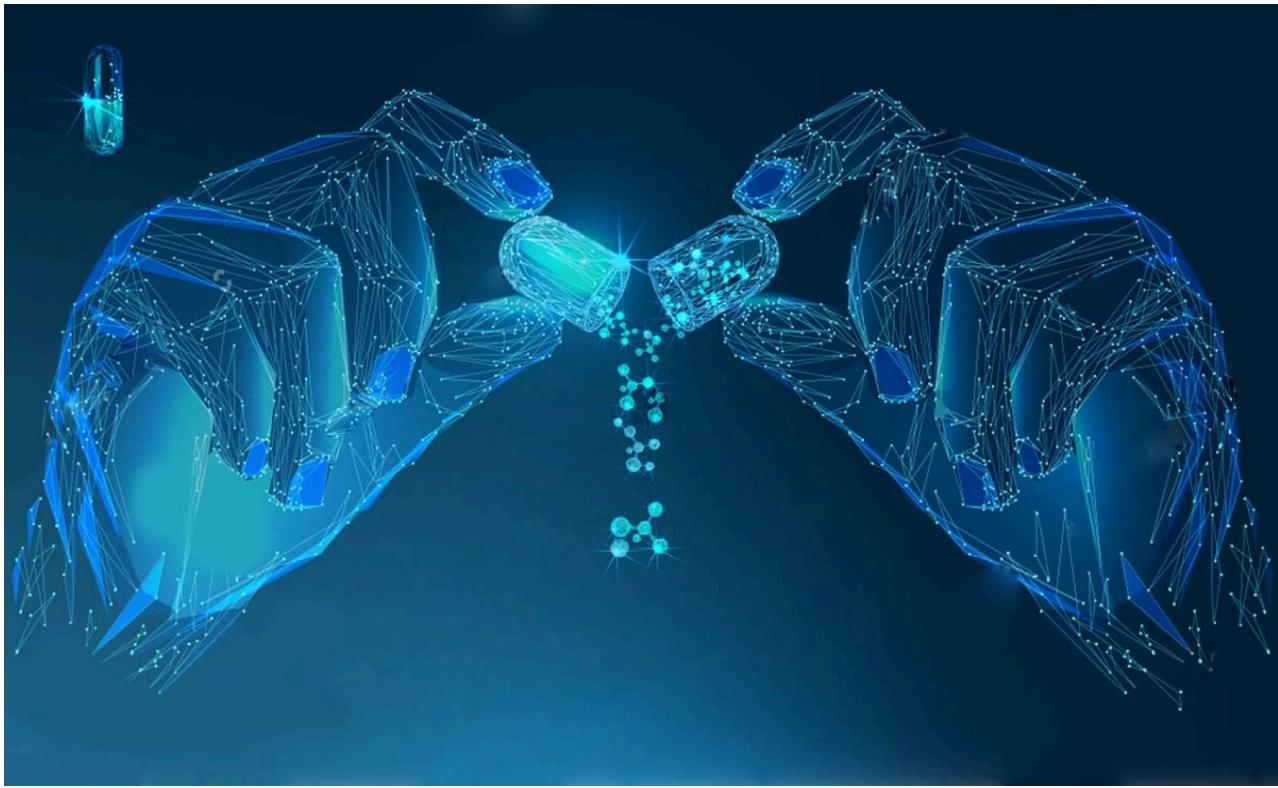
Team Member 3: SANDHIYA S

Email: sandhiyas23cse@vetias.ac.in

Team Member 4: SIVAPRIYAN S

Email: Sivapriyans23cse@vetias.ac.in

Title: Medical Inventory Management



Project Overview:

The **Medical Inventory Management System** is developed on Salesforce CRM to manage the flow of medical supplies, medicines, and equipment efficiently. It streamlines purchase order processing, supplier management, and stock tracking. The system reduces manual effort, minimizes errors, and ensures the availability of essential medical items in healthcare organizations.

Objectives:

- To maintain a centralized record of all medical items.
- To automate purchase order creation and approvals.
- To track supplier details and order histories.
- To monitor stock levels and trigger alerts for low/expired items.
- To generate reports and dashboards for decision-making.
- To implement Salesforce automation (Triggers, Validation Rules, Workflows).
- To ensure secure, role-based access for different users (Admin, Inventory Manager, Staff).

Scope of the Project

- **In Scope:**
 - Inventory tracking (medicines, equipment).
 - Supplier and purchase order management.
 - Role-based security profiles.
 - Salesforce automation (triggers, workflows, approval processes).
 - Reports & dashboards for insights.
- **Out of Scope:**
 - Integration with external hospital billing systems.
 - Advanced AI/ML prediction of demand.
 - Mobile app for offline inventory management.

System Modules

1. **Inventory Management** – Add, update, and track medical items.
2. **Supplier Management** – Maintain supplier details and history.
3. **Purchase Order Management** – Automate order creation, approvals, and cost calculation.
4. **Stock Monitoring** – Alerts for stock shortage or expiry.
5. **Reports & Dashboards** – Generate summary and detailed reports.
6. **User Management** – Roles and profiles for secure access.

Deliverables

- Salesforce Application (Medical Inventory Management App).
- Custom Objects: Medical Item, Supplier, Purchase Order, Order Item.
- Automation Components: Triggers, Validation Rules, Workflows.
- Reports & Dashboards for purchase orders and inventory tracking.
- User Profiles: Admin, Inventory Manager, Standard User.

Benefits

- Efficient stock management with reduced wastage.
- Automated purchase order process reduces manual work.
- Improved supplier accountability.

- Real-time monitoring of stock and costs.

Student Learning Outcomes

By completing this project, students were able to:

1. **Understand Salesforce CRM Platform**
 - Gained hands-on experience in customizing Salesforce for real-world applications.
2. **Apply Cloud Application Development Skills**
 - Learned how to create custom objects, fields, page layouts, and relationships.
3. **Implement Business Automation**
 - Designed and implemented triggers, workflows, validation rules, and approval processes.
4. **Build Reports and Dashboards**
 - Developed analytical reports and dashboards to monitor purchase orders, suppliers, and stock levels.
5. **Work with Role-Based Access Control**
 - Configured user profiles and permissions for Admin, Manager, and Staff.
6. **Develop Problem-Solving & Analytical Thinking**
 - Understood challenges in healthcare inventory management and applied Salesforce solutions.
7. **Improve Documentation and Presentation Skills**
 - Prepared structured documentation of the project with technical and functional details.
8. **Teamwork and Project Management** (*if done as a group*)
 - Learned to collaborate with peers and manage tasks effectively within deadlines.

System Requirements

1. Hardware Requirements

- **Processor:** Intel Core i3 / AMD equivalent or higher
- **RAM:** Minimum 4 GB (8 GB recommended)

- **Hard Disk:** At least 20 GB of free space
- **Display:** 1024 × 768 resolution or higher
- **Internet Connection:** Stable broadband connection for Salesforce access

2. Software Requirements

- **Operating System:** Windows 10 / Windows 11 / macOS / Linux (any modern OS)
- **Web Browser:** Google Chrome (latest version) / Mozilla Firefox / Microsoft Edge
- **Salesforce Platform:** Salesforce Developer Edition (provided by Naan Mudhalvan)
- **IDE/Tools (Optional):**
 - Salesforce Developer Console
 - VS Code with Salesforce Extensions (for Apex development)
- **PDF/Doc Viewer:** MS Office / LibreOffice / Google Docs for documentation

3. User Requirements

- Basic knowledge of Salesforce navigation.
- Internet connectivity to log in and use Salesforce.
- Valid Salesforce Developer account (provided for students).

Phase se No.	Phase Name	Description	Page e No.
1	Requirement Analysis & Planning	Gathering requirements from healthcare staff, administrators, and inventory managers; defining project scope and objectives; planning the data model and workflows for suppliers, purchase orders, and medical items.	

2	Salesforce Development – Backend & Configurations	Creating custom objects (Medical Item, Supplier, Purchase Order, Order Item), fields, and relationships; setting up Flows and Apex Triggers for purchase order automation and total amount calculations.
3	UI/UX Development & Customization	Building the Medical Inventory Management App, customizing page layouts, compact layouts, and record pages; implementing Lightning pages and UI logic with Flows for purchase order approvals and stock management.
4	Data Migration, Testing & Security	Creating Users, Profiles (Admin, Inventory Manager, Standard User), Permission Sets, Public Groups, and Sharing Rules; configuring Report Types, Reports, and Dashboards; testing functionalities for accuracy and ensuring data security.
5	Deployment, Documentation & Maintenance	Finalizing the Home Page and App Navigation, deploying the Medical Inventory Management system, preparing project documentation, writing conclusion, and planning for future enhancements and maintenance.

Phase 1: Requirement Analysis & Planning

In this phase, the requirements for the Medical Inventory Management System were gathered, analyzed, and documented. The main objective of this phase was to clearly understand the needs of healthcare staff, administrators, and inventory managers to design a system that ensures efficient stock management and reduces manual errors.

Activities Performed:

1. Requirement Gathering:

- Collected inputs from hospital administrators, inventory managers, and staff about the problems faced in managing medicines and medical equipment.
- Identified challenges such as stock shortage, manual purchase orders, lack of supplier records, and delays in approvals.

2. Defining Scope and Objectives:

- Established the key goals of the project: inventory tracking, supplier management, purchase order automation, and stock alerts.
- Defined boundaries by excluding features such as billing system integration and AI-based demand forecasting.

3. Data Model Planning:

- Identified entities such as Medical Item, Supplier, Purchase Order, and Order Item.
- Defined relationships among these objects to ensure smooth workflow.

4. Workflow Planning:

- Designed workflows for common tasks like creating purchase orders, approving suppliers, and tracking item expiry.
- Planned automation using Salesforce features (Validation Rules, Triggers, Flows).

5. Project Planning:

- Estimated effort required: 31 hours.
- Divided project into 5 phases (Requirement Analysis, Development, UI Customization, Testing & Security, Deployment & Documentation). Assigned timelines and deliverables for each phase.

Outcome of Phase 1:

- A clear understanding of the project requirements.
- A well-defined project scope and objectives.
- A preliminary data model and workflow plan.
- A structured project roadmap for execution.

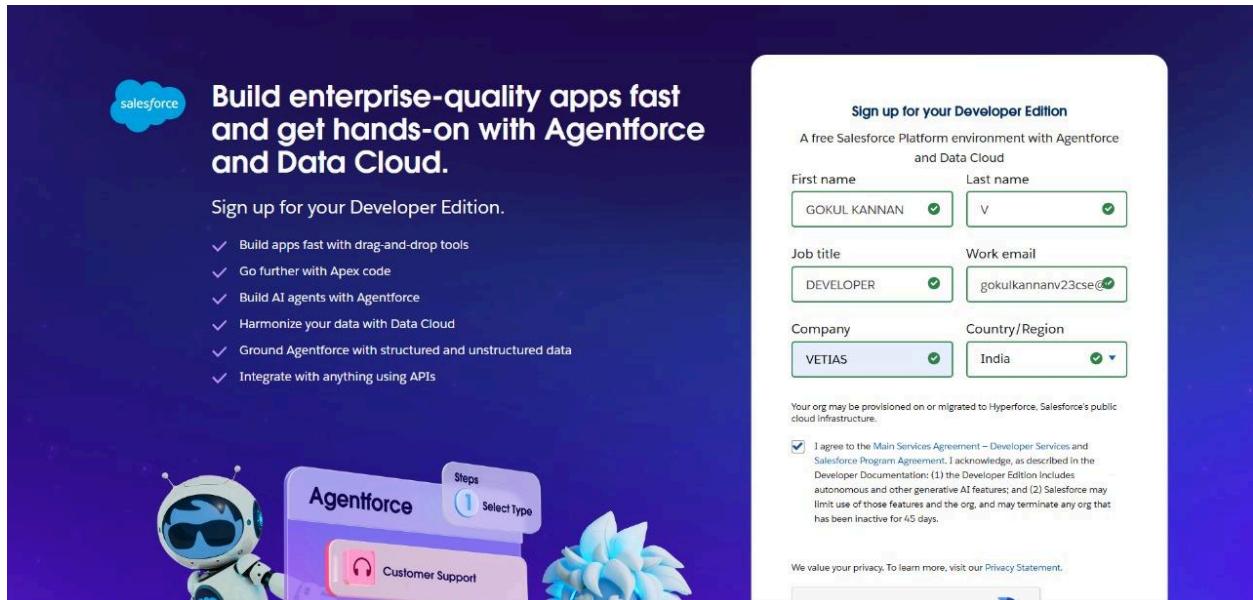
Phase 2: Salesforce Development – Backend & Configurations

Milestone 1: Salesforce developer account creation.

ACTIVITY 1: Creating Developer Account

Creating a developer org in salesforce.

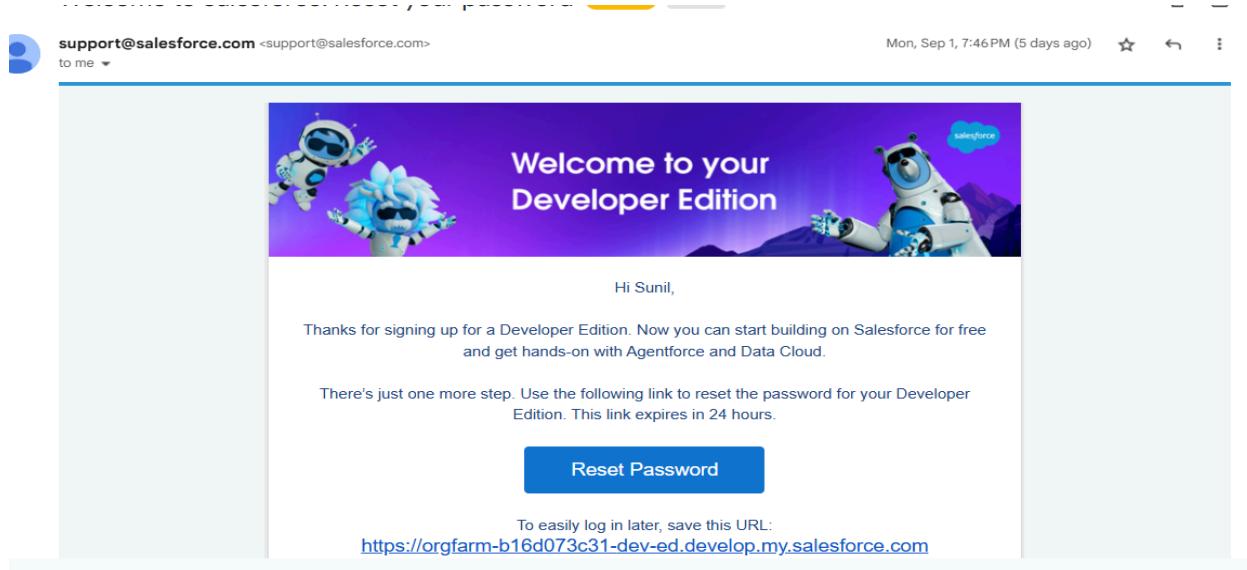
1.<https://developer.salesforce.com/signup>



1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code
7. Username : should be a combination of your name and company

ACTIVITY 2:Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the Reset Password to activate your account. The email may take 5-10mins.
2. Click on Reset Password
3. Give a password and answer a security question and click on change password.



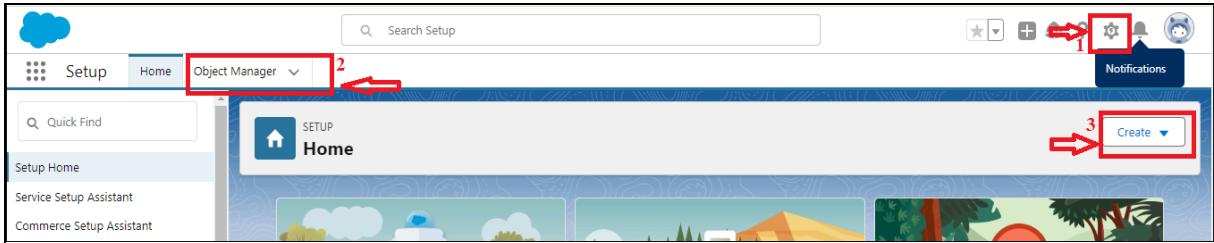
4. Then you will redirect to your salesforce setup page.

Milestone 2- Objects

Activity 1: Creating a Product Object

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object.
4. Enter the label name as Product
5. Enter Plural label name as Products
6. Enter Record Name as Product ID
7. Select Data Type as Text.
8. Select Allow reports.
9. Select Allow search.
10. Click on Save and New



New Custom Object

Custom Object Definition Edit

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label	Product	Example: Account
Plural Label	Products	Example: Accounts

Starts with vowel sound

The Object Name is used when referencing the object via the API.

Object Name	Product	Example: Account
-------------	---------	------------------

Description

Context-Sensitive Help Setting

- Open the standard Salesforce.com Help & Training window
- Open a window using a Visualforce page

Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name	Product ID	Example: Account Name
-------------	------------	-----------------------

Data Type Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Optional Features

Allow Reports 8

Allow Activities

Track Field History

Allow in Chatter Groups

Enable Licensing i

Deployment Status

In Development

Deployed

Search Status

When this setting is enabled, your users can find records of this object type when they search. [Learn more](#).

Allow Search 9

Object Creation Options (Available only when custom object is first created)

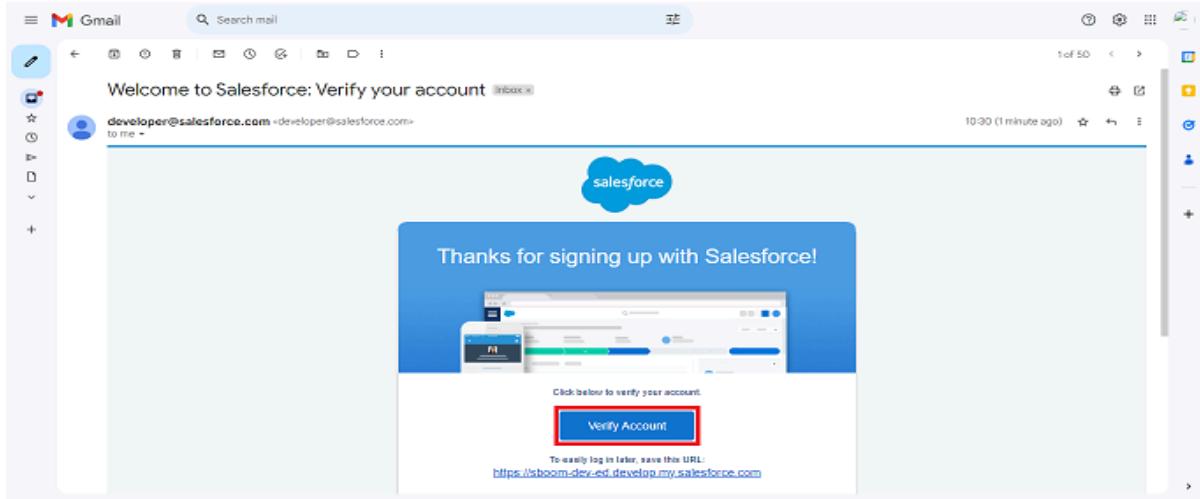
Add Notes and Attachments related list to default page layout

Launch New Custom Tab Wizard after saving this custom object 10

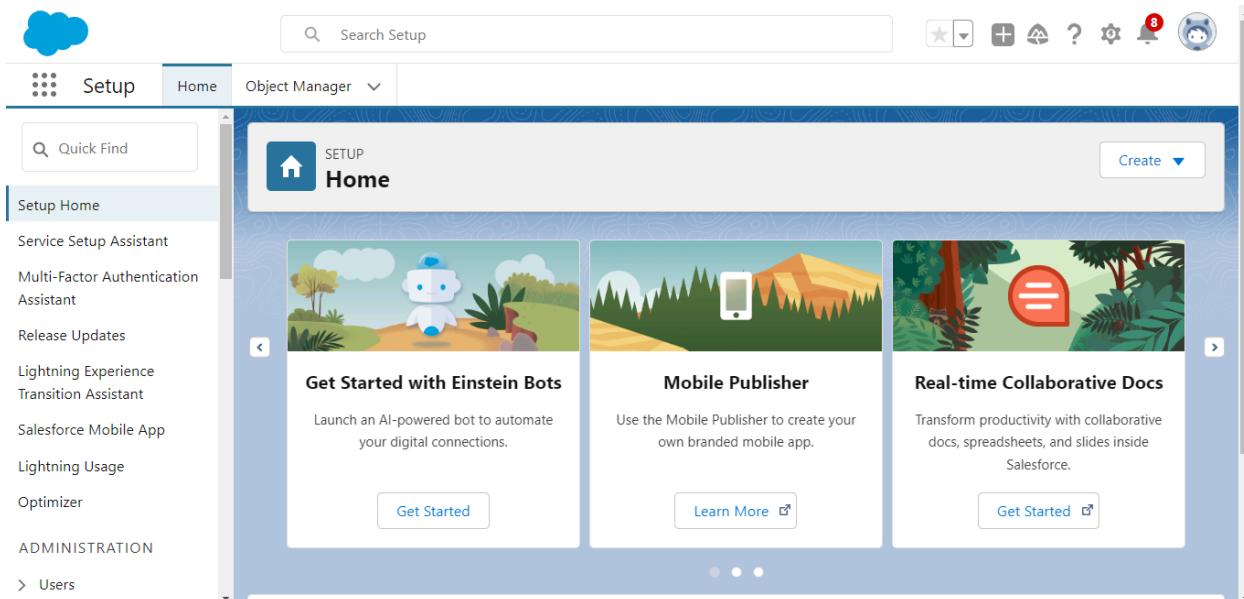
In the same way Create Purchase Order, Order Item, Inventory Transaction and Supplier objects.

ACTIVITY 2:Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.



2. Click on Verify Account
3. Give a password and answer a security question and click on change password.
4. Then you will redirect to your salesforce setup page.



Milestone 3- Tabs

Activity 1: Creating a tab for Product Object

1. Go to the setup page >> type Tabs in Quick Find bar
2. Click on tabs
3. Click on New (under custom object tab).
4. Select Object(Product) >> Select the tab style
5. Click on Next >> (Add to profiles page) keep it as default >> Click on Next (Add to Custom App) uncheck the include tab .
6. Make sure that the Append tab to user's existing personal customizations is checked.
7. Click save

Activity 2: Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are "Purchase Order, Order Item, Inventory Transaction, Supplier"
2. Follow the same steps as mentioned in Activity -1 .

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected. On the left, there's a sidebar with 'User Interface' and 'Tabs' sections. The main area is titled 'Custom Tabs' and contains three sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section lists tabs for 'Inventory Transactions' (Box), 'Order Items' (Fan), 'Products' (Bell), 'Purchase Orders' (Car), and 'Suppliers' (Cup). Each row has 'Edit | Del' links and a 'New' button.

Milestone 4- The Lightning App

Activity 1: Create a Lightning App for Medical Inventory Management

1. From Setup, enter App Manager in the Quick Find and select App Manager.
2. Click New Lightning App.
3. Enter Medical Inventory Management as the App Name >> Click on upload image and add an image related to Medical Inventory then click next
4. Under App Options, leave the default selections and click next.
5. Under Utility Items, leave as is and click Next.

6. From Available Items, select Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, Reports, and Dashboards and move them to Selected Item and Click Next.
7. From Available Profiles, select System Administrator and move it to Selected Profiles.
8. Click Save & Finish.

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

- * App Name: Medical Inventory Management
- * Developer Name: Medical_Inventory_Management

App Branding

- Image:
- Primary Color Hex Value: #0070D2

Description

Enter a description...

Org Theme Options

Use the app's image and color instead of the org's custom theme

App Launcher Preview

Medical Inventory Manage...

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

- Accounts
- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests
- Dashboards

Selected Items

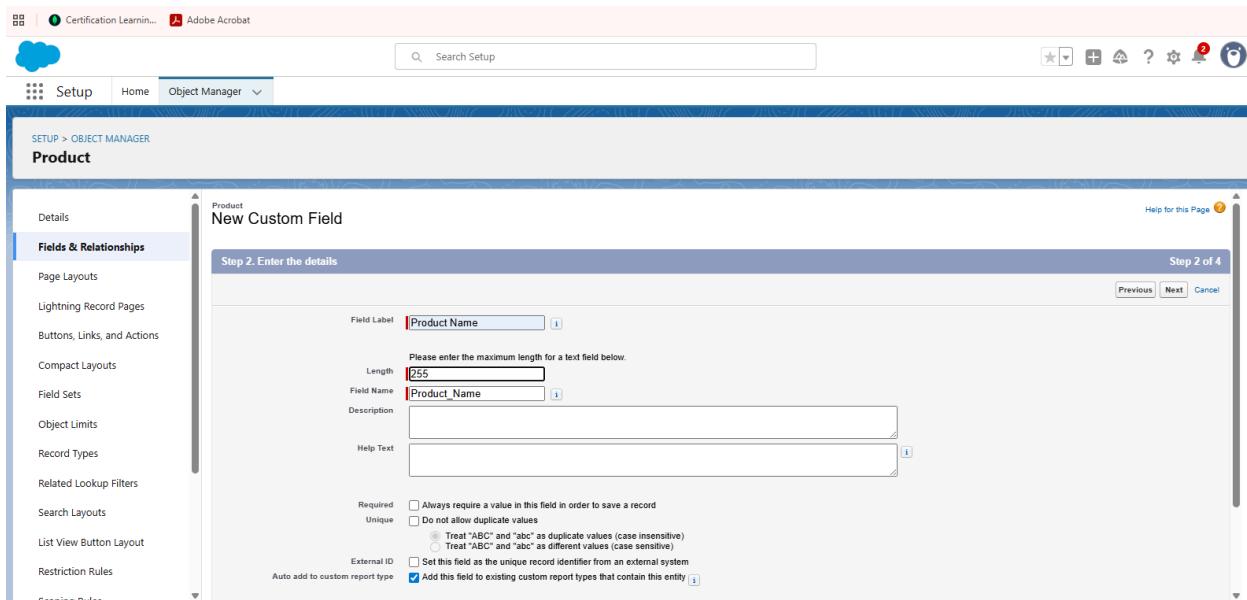
- Products
- Purchase Orders
- Order Items
- Inventory Transactions
- Suppliers
- Reports

Milestone 5- Fields

Activity 1: Creating a Text Field in Product Object

To create fields in an object:

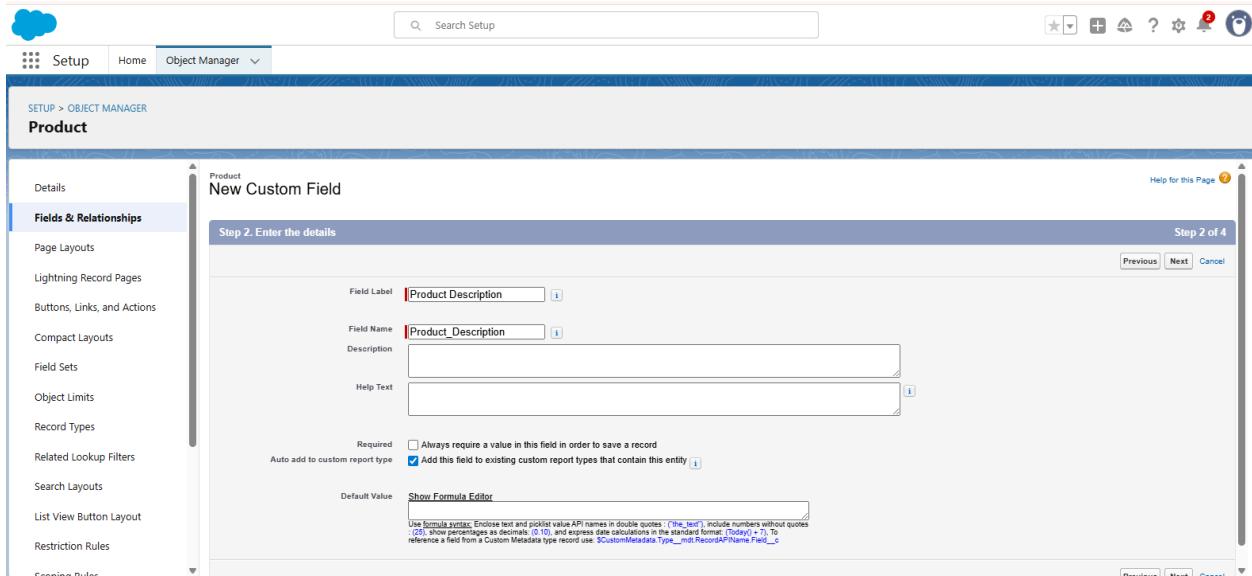
1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select Product custom object.
4. Select Fields & Relationships from the left navigation
5. Click on New
6. Select Text field, click Next
7. Enter Field Label as “Product Name” and Length 255.
8. Select Required Field.
9. Click Next, Next, then Save & New.



Activity 2: Creating a TextArea Field in Product Object

To create fields in an object:

1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select Product custom object.
4. Select Fields & Relationships from the left navigation
5. Click on New
6. Select TextArea field, click Next
7. Enter Field Label as “Product Description” .
8. Click Next, Next, then Save & New



The screenshot shows the Salesforce Setup interface with the 'Object Manager' for the 'Product' custom object. The 'Fields & Relationships' tab is selected. On the left, a sidebar lists various setup categories like Details, Page Layouts, Lightning Record Pages, etc. On the right, a list of field types is shown with descriptions:

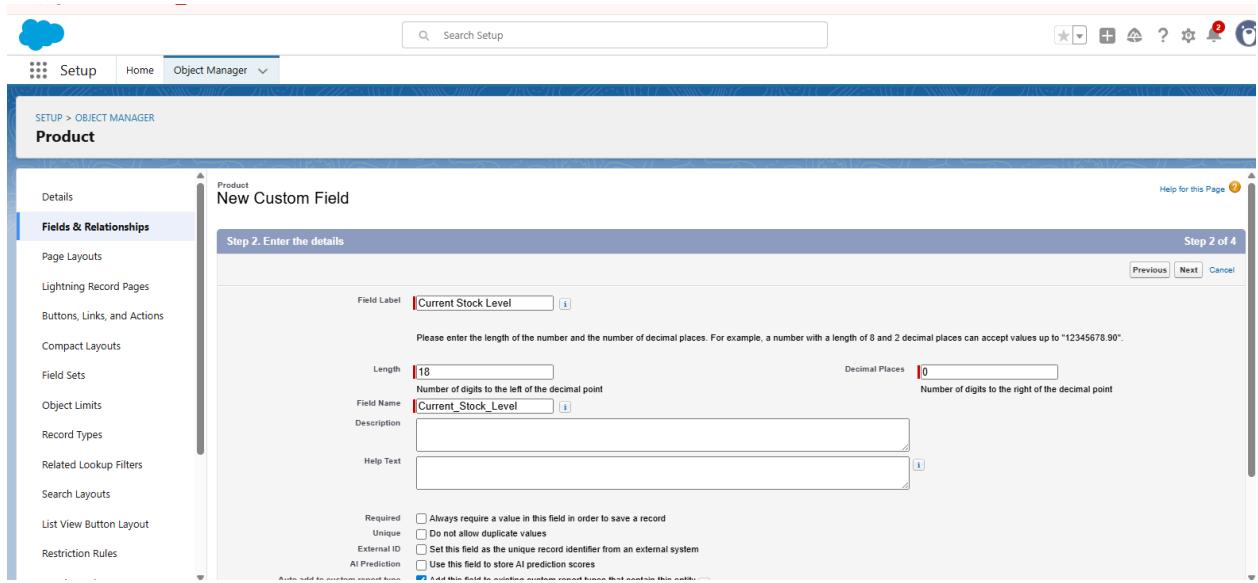
- Currency
- Date
- DateTime
- Email
- Geolocation
- Number
- Percent
- Phone
- Picklist
- Picklist (Multi-Select)
- Text
- Text Area**
- Text Area (Long)
- Text Area (Rich)
- Text (Encrypted) 1
- Time
- URL

Descriptions for each field type are provided to the right of the list. The 'Text Area' and 'Text' types are highlighted with blue ovals.

Activity 3: Creating a Number Field in Product object

To create fields in an object:

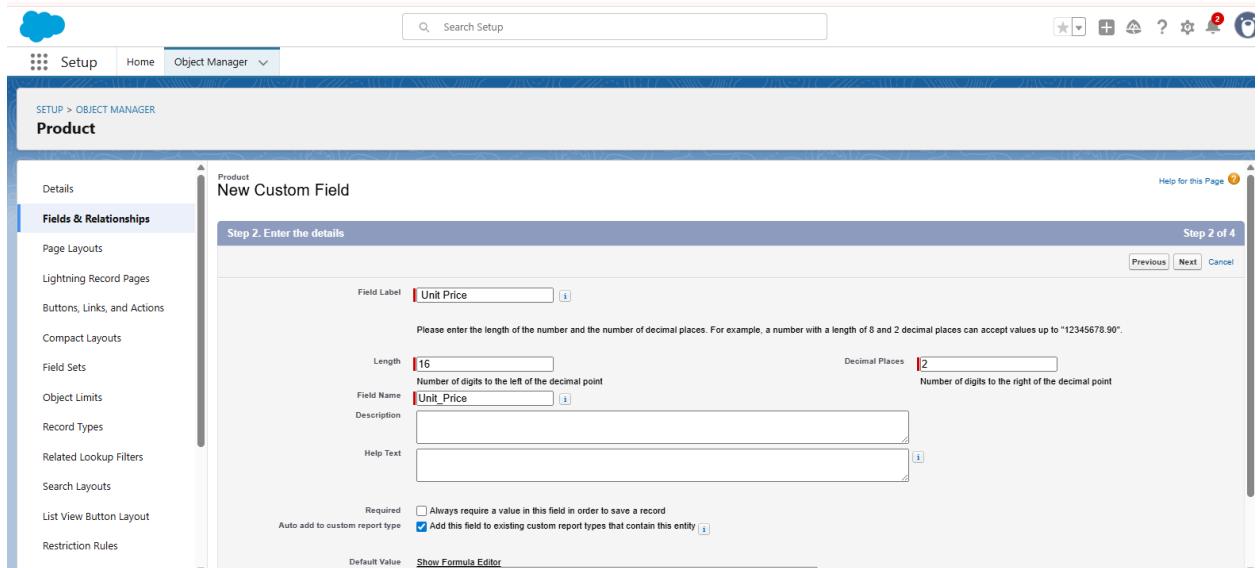
1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product custom object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Number” and click Next.
5. Enter Field Label as “ Current Stock Level”.
6. Length - 18, Decimal Places - 0.
7. Click on Next, Next and Save.



Activity 4: Creating a Currency Field in Product object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product custom object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Currency” and click Next.
5. Enter Field Label as “ Unit Price”.
6. Length - 16, Decimal Places - 2.
7. Select Required Field.
8. Click on Next, Next and Save



Activity 5 : Creating Lookup Relationship in Purchase Order Object

A Lookup relationship is a type of relationship in Salesforce that connects two objects together based on a field known as the Lookup field. It establishes a relationship between a child object and a parent object, allowing the child object to reference the parent object.

To Create a relationship from Purchase Order to Supplier .

1. Go to the Setup page >> click on Object manager >> type object name(Purchase Order) in the quick find bar >> click on the Purchase Order object.
2. Click on Fields & Relationship
3. Click on New.
4. Select “Lookup relationship” as data type and click Next.
5. Select the related object “Supplier”.
6. Click on Next.
7. Give Field Label as “Supplier ID” .
8. Select Required Field.
9. Click on Next , Next, Next , Save.

Activity 6: Creating a Date Field in Purchase Order object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Date” and click Next.
5. Enter Field Label as “Order Date”.
6. Click on Next, Next and Save

Step 2. Enter the details

Field Label: Order Date 5

Field Name: Order_Date

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 6

Default Value: Show Formula Editor

Use formula syntax. Enclose text and picklist 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__mdt.RecordAPIName.Field__c

Activity 7: Creating a Roll-Up Summary Field in Purchase Order object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Roll-Up Summary” and click Next.
5. Enter Field Label as “ Order Count”.
6. Choose the Summarized Object as “Order Items”.
7. For Select Roll-Up Type select “Count”.
8. Click on Next, Next and Save

Data Type

None Selected Select one of the data types below.

Auto Number A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

Formula A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

Roll-Up Summary 4 A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

Purchase Order Help for this Page

New Custom Field

Step 2. Enter the details Step 2 of 5

Field Label 5

Field Name

Description

Help Text

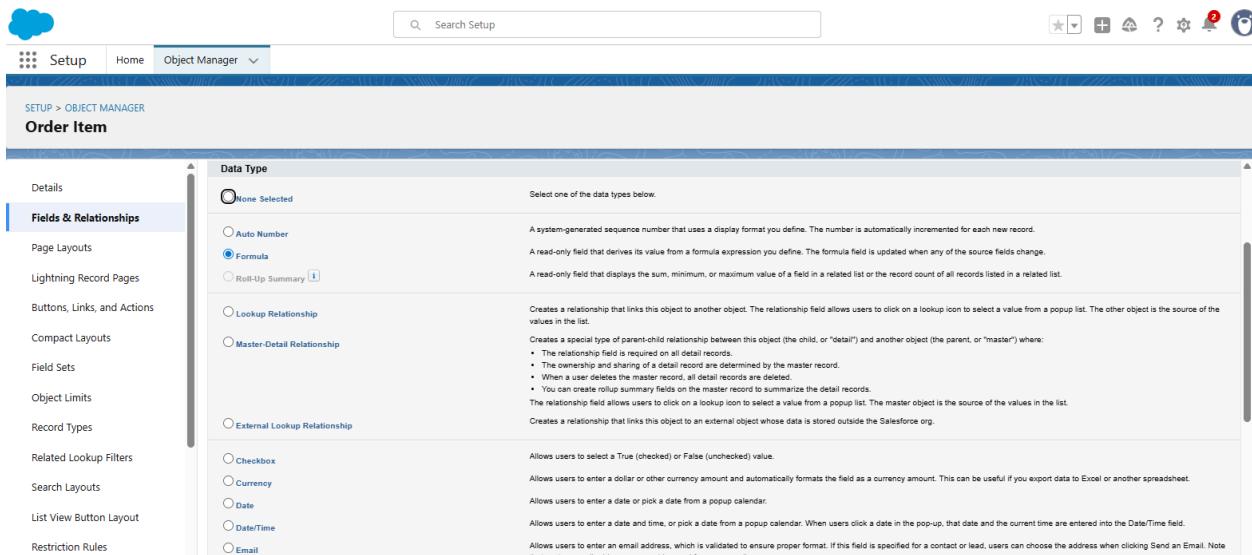
Auto add to custom report type Add this field to existing custom report types that contain this entity

Previous **Next** **Cancel**

Activity 8: Creating a Unit Price Formula Field in Order Item object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Order Item) in quick find box >> click on the Order Item object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Unit Price.
6. Select formula return type Currency, Click Next
7. Create and insert Advance formula: Product_ID__r.Unit_Price__c
8. Click Next, Next, then Save.

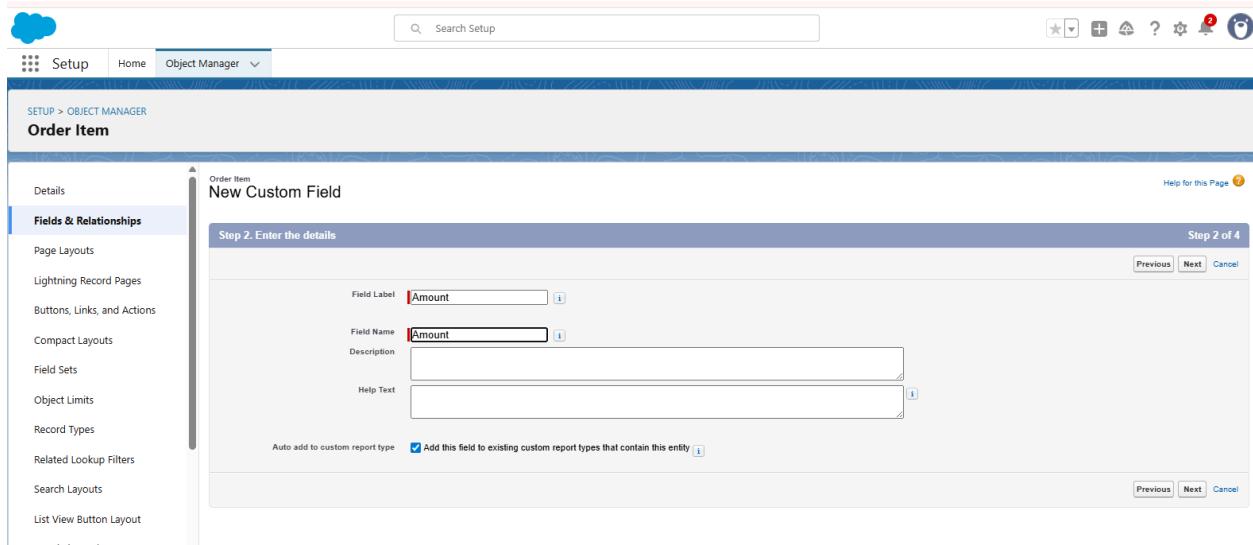


Activity 9: Creating a Amount Formula Field in Order Item object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Order Item) in quick find box >> click on the Order Item object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Amount.
6. Select formula return type Currency, Click Next

7. Create and insert Advance formula: Quantity_Received__c * Unit_Price__c
8. Click Next, Next, then Save.



Activity 10: Creating a Picklist Field in Inventory Transaction Object

To create fields in an object:

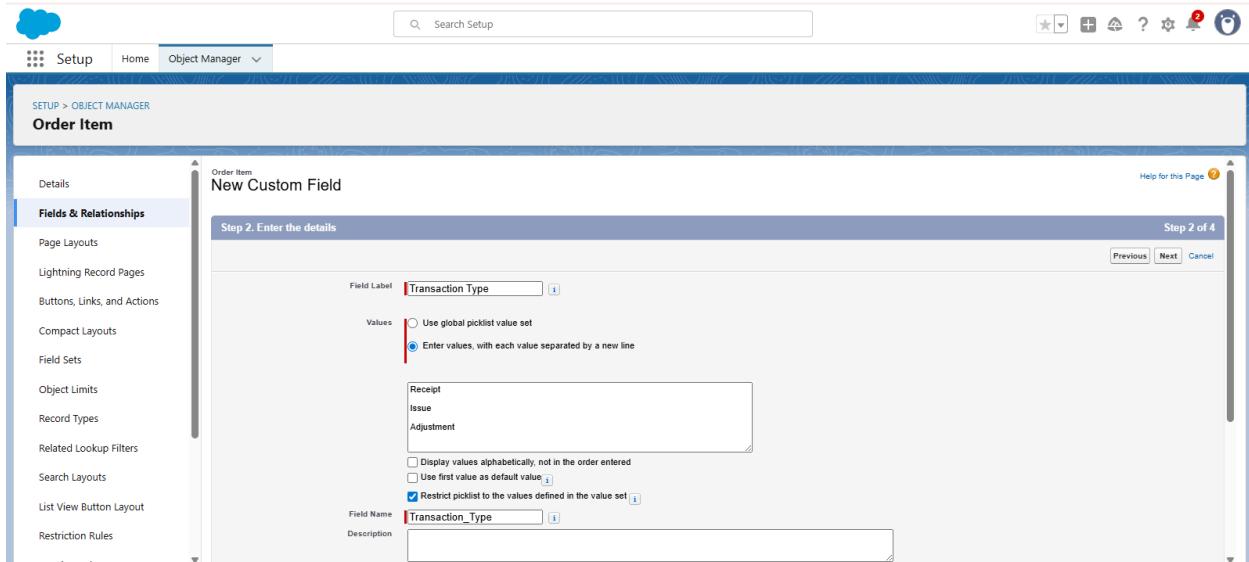
1. Go to setup >> click on Object Manager >> type object name(Inventory Transaction) in quick find box>> click on the Inventory Transaction Object.
2. Now click on “Fields & Relationships” .
3. Click on New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Transaction Type”.
6. In values select “Enter values, with each value separated by a new line” and enter values as shown below.

Receipt

Issue

Adjustment

7. Click on Next, Next and Save.



Activity 11: Creating a Total Order Cost Formula Field in Inventory Transaction object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Inventory Transaction) in quick find box >> click on the Order Item object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Formula” and click Next.
5. Enter field label Total Order Cost.
6. Select formula return type Currency, Click Next
7. Create and insert Advance formula:
Purchase_Order_ID__r.Total_Order_Cost__c
8. Click Next, Next, then Save.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar. The main area is titled 'SETUP > OBJECT MANAGER' and 'Order Item'. On the left, a sidebar lists various setup options like 'Details', 'Fields & Relationships', and 'Page Layouts'. The current step is 'Step 2. Choose output type' of a 'New Custom Field' for 'Order Item'. The field label is 'Order Cost' and the field name is 'Order_Cost'. The 'Formula Return Type' section is expanded, showing options for 'None Selected', 'Checkbox', 'Currency' (selected), 'Date', 'DateTime', 'Number', and 'Percent'. A note indicates that 'Currency' calculates a dollar or other currency amount and automatically formats the field as a currency amount.

Activity 12: Creating a Phone Field in Supplier object

To create fields in an object:

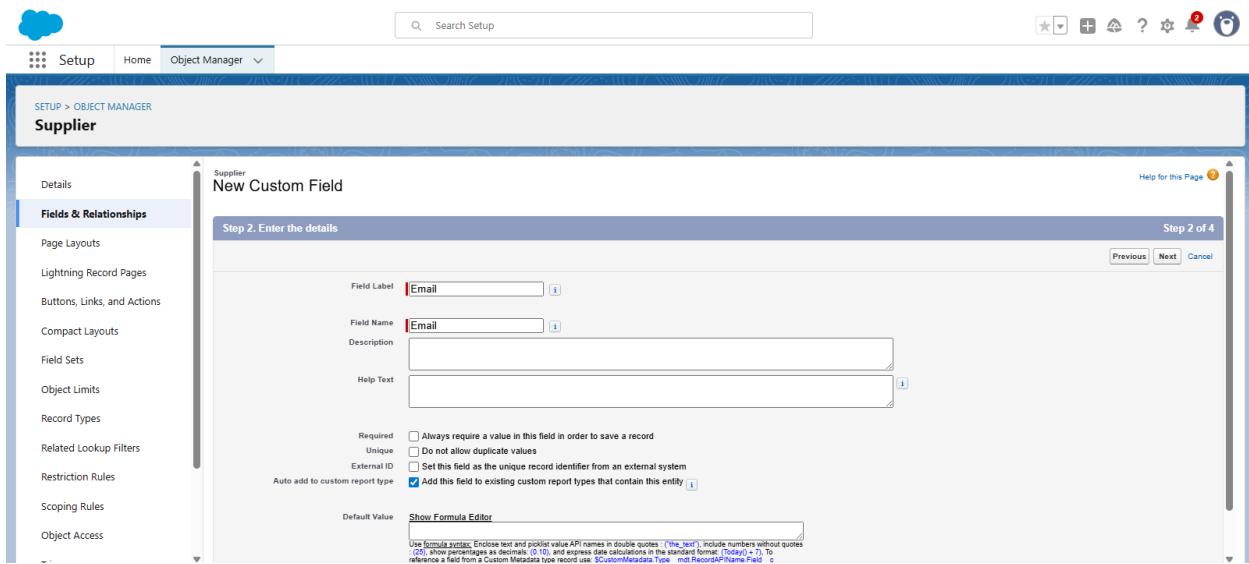
1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box>> click on the Supplier object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Phone” and click Next.
5. Enter the Field Label as “ Phone Number”.
6. Select Required Field.
7. Click on Next, Next and Save.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar. The main area is titled 'SETUP > OBJECT MANAGER' and 'Supplier'. On the left, a sidebar lists various setup options like 'Details', 'Fields & Relationships', and 'Page Layouts'. The current step is 'Step 2. Enter the details' of a 'New Custom Field' for 'Supplier'. The field label is 'Phone Number' and the field name is 'Phone_Number'. The 'Required' checkbox is checked. A note indicates that 'Always require a value in this field in order to save a record' and 'Add this field to existing custom report types that contain this entity'. The 'Show Formula Editor' link is visible at the bottom.

Activity 13: Creating a Email Field in Supplier object

To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box>> click on the Supplier object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Email” and click Next.
5. Enter the Field Label as “ Email”.
6. Click on Next, Next and Save.



Milestone 6 -Editing of Page Layouts

Activity 1: To edit a Page Layout in Product Object

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object >> Page Layouts .
2. Click on the Product Layout.
3. Drag and Arrange the field as shown below.

- Click on Save.

Activity 2: To edit a Page Layout in Purchase Order Object

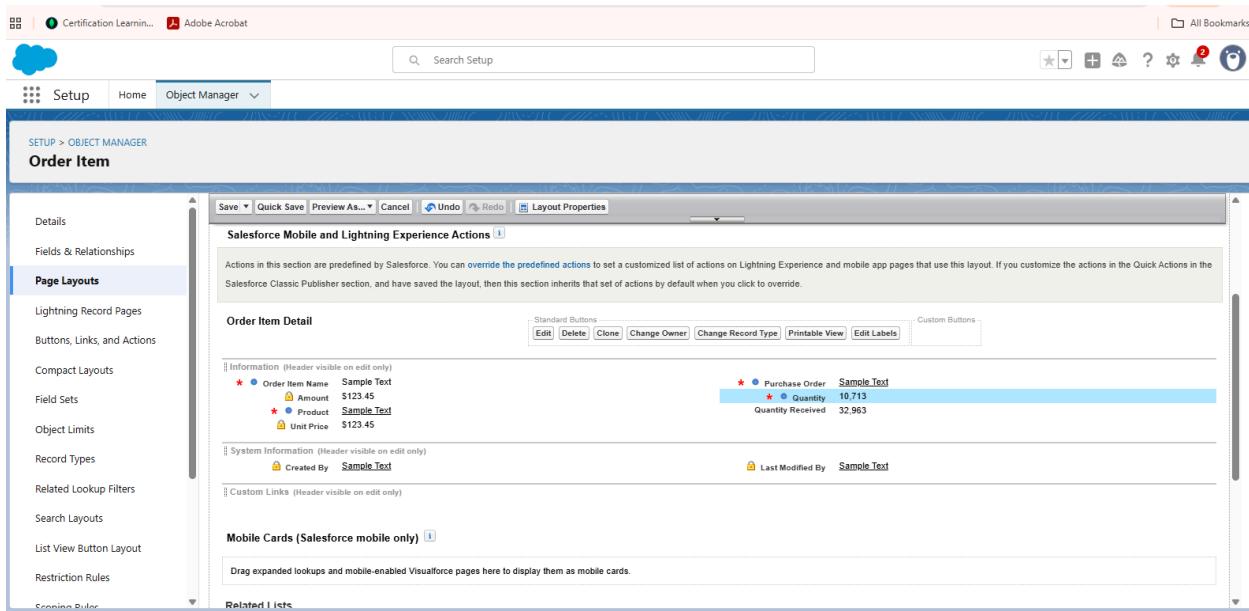
- Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box >> click on the Purchase Order object >> Page Layouts.
- Click on the Purchase Order Layout
- Drag and Arrange the field as shown below

- Click on field Order Date >> click on settings >> select Required and save it.

5. Click on field Total Order Cost >> click on settings >> select Read Only and save it.
6. Click Save.

Activity 3: To edit a Page Layout in Order Item Object

1. Go to setup >> click on Object Manager >> type object name(Order Item) in quick find box >> click on the Order Item object >> Page Layouts.
2. Click on the Order Item Layout
3. Drag and Arrange the field as shown below



4. Click Save.

Activity 4: To edit a Page Layout in Inventory Transaction Object

1. Go to setup >> click on Object Manager >> type object name(Inventory Transaction) in quick find box >> click on the Inventory Transaction object >> Page Layouts.
2. Click on the Inventory Transaction Layout
3. Drag and Arrange the field as shown below

4. Click Save.

Activity 5: To edit a Page Layout in Supplier Object

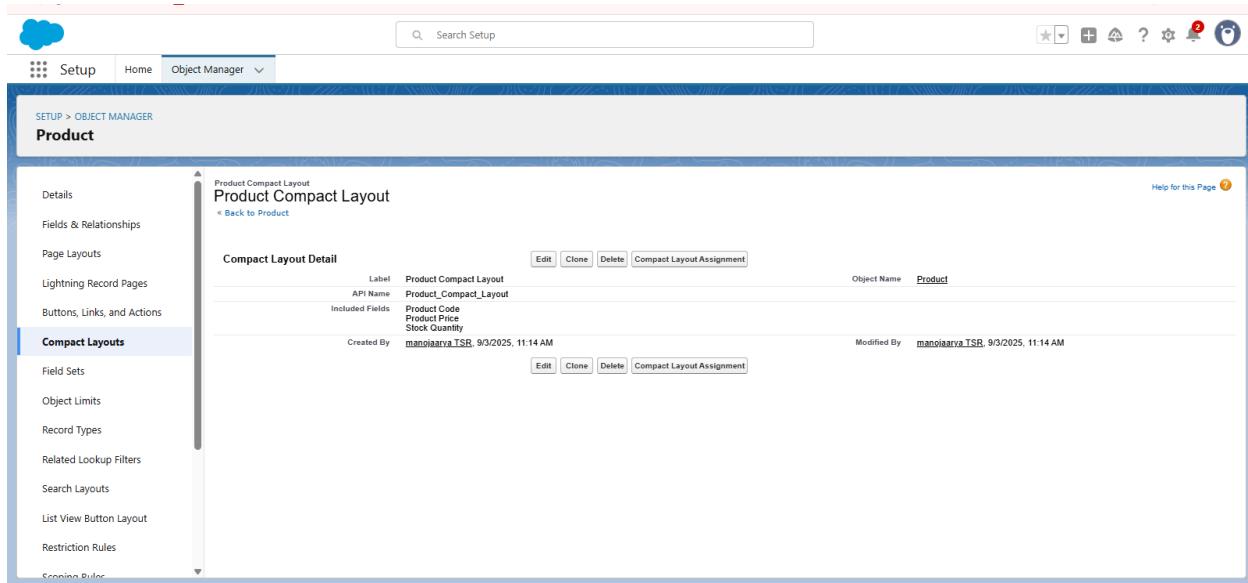
1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box >> click on the Supplier object >> Page Layouts.
2. Click on the Supplier Layout
3. Drag and Arrange the field as shown below

Click Save

Milestone 7 - Compact Layouts

Activity 1: To create a Compact Layout to a Product Object

1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object
2. Click on Compact Layouts in the sidebar .
3. Click on New.
4. Enter the Label as "Product Compact Layout".
5. Select the Compact Layout Fields : Select Product name, Unit Price, Current Stock Level.
6. Click Save.
7. Click Compact Layout Assignment.
8. Click Edit Assignment.
9. Choose "Product Compact Layout" from the dropdown.
10. Click Save



Activity 2: To create a Compact Layout to a Purchase Order Object

1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box >> click on the Purchase Order object

2. Click on Compact Layouts in the sidebar .
3. Click on New.
4. Enter the Label as “Purchase Order Compact Layout”.
5. Select the Compact Layout Fields : Select Purchase Order ID, Order Date, Total Order Cost, Supplier ID.
6. Click Save.
7. Click Compact Layout Assignment.
8. Click Edit Assignment.
9. Choose "Purchase Order Compact Layout" from the dropdown.
10. Click Save.

The screenshot shows the Salesforce Object Manager interface for the Purchase Order object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, and Compact Layouts. The Compact Layouts option is selected. In the main content area, a new compact layout is being created with the label 'Purchase Order Compact Layout' and API name 'Purchase_Order_Compact_Layout'. It includes fields for Purchase Order ID, Order Date, Total Order Cost, and Supplier. The layout is assigned to the Purchase Order object. The status bar at the bottom indicates the URL as orgfarm-acfbeac20d-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager.../view.

Milestone 8 - Validation Rules

Activity 1: To create an Expected Delivery Date Validation rule to a Employee Object

1. Go to setup >> click on Object Manager >> type object name(Purchase Order) in quick find box>> click on the Purchase Order object
2. Click on the validation rule >> click on New.
3. Enter the Rule name as “Expected Delivery Date Validation”.
4. Select Active
5. Insert the Error Condition Formula as :
(Expected_Delivery_Date__c - Order_Date__c)> 7

The screenshot shows the Salesforce Setup interface for the Purchase Order object. On the left, a sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area is titled "Purchase Order Validation Rule" and displays a single rule named "Expected_Delivery_Date_Validation". The rule's formula is "(Expected_Delivery_Date__c - Order_Date__c) > 7". The error message is "The Expected Delivery Date should not exceed 7 days." The rule is marked as "Active". The "Error Location" is set to "Top of Page". The "Created By" and "Modified By" fields both show "manojarya TSR" with the timestamp "9/6/2025, 1:34 AM". There are "Edit" and "Clone" buttons at the bottom of the rule card.

6. Enter the Error Message as “The Expected Delivery Date should not exceed 7 days.”.
7. Select the Error location as Top of Page
8. Click Save.

Milestone 9 - Profiles

Activity 1: To create an Inventory Manager Profile

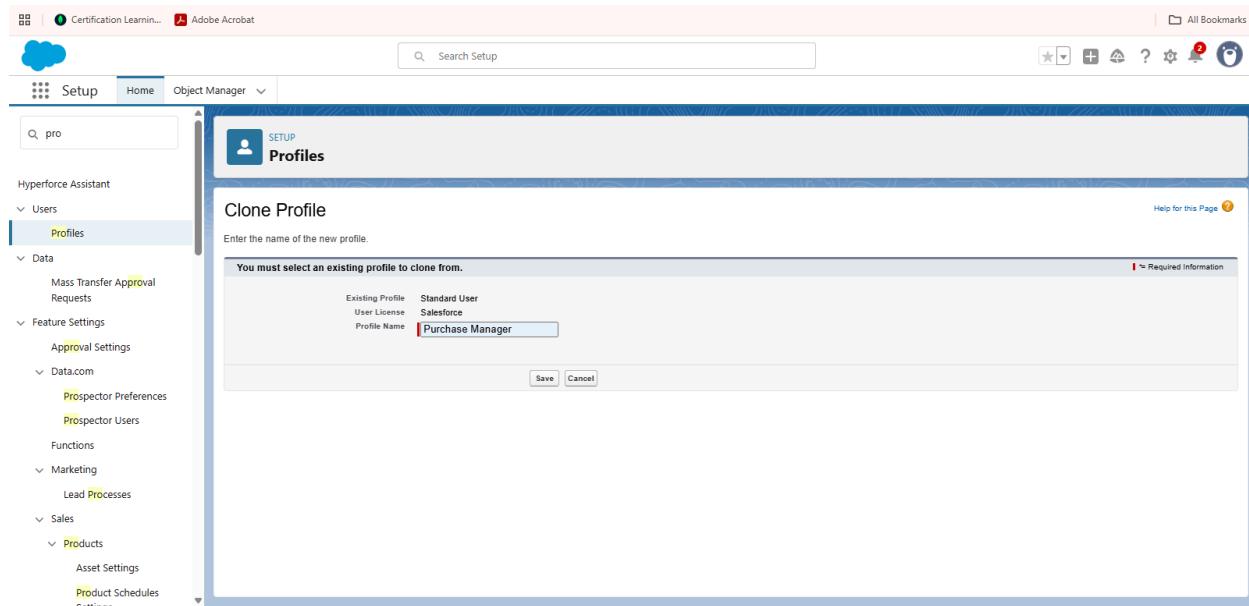
1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Inventory Manager) >> Save.

The screenshot shows the Salesforce Setup interface under the "Profiles" section. The sidebar includes categories like Users, Data, Marketing, Sales, and Products. The main area is titled "Clone Profile" and asks for the name of the new profile. It says "You must select an existing profile to clone from." and provides three options: "Existing Profile", "Standard User", and "Salesforce". The "Profile Name" field contains "Inventory Manager". At the bottom are "Save" and "Cancel" buttons.

2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Medical Inventory Management
4. Change the password policies as mentioned :
5. User passwords expire in should be “ never expires ”.
6. Minimum password length should be “ 8 ”, and click save.

Activity 2: To create an Purchase Manager Profile

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Purchase Manager) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Medical Inventory Management.



4. Scroll down to Custom Object Permissions and Give access permissions as mentioned in the below diagram.
5. Change the password policies as mentioned :
6. User passwords expire in should be “ never expires ”.
7. Minimum password length should be “ 8 ”, and click save.

Milestone 10 - Roles

Activity 1 : Create a Purchasing Manager Role.

1. Go to quick find >> Search for Roles >> click on Set Up Roles.

The screenshot shows the Salesforce Setup interface for Roles. On the left, a sidebar navigation includes 'Users' and 'Roles'. Under 'Roles', there are sections for 'Feature Settings', 'Sales' (with 'Contact Roles on Contracts' and 'Contact Roles on Opportunities'), 'Service' (with 'Case Teams' and 'Case Team Roles'), and 'Case' (with 'Contact Roles on Cases'). A note at the bottom says 'Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Understanding Roles' and displays a sample role hierarchy. At the top, it says 'Sample Role Hierarchy' and 'View other sample Role Hierarchies: Territory-based Sample'. The hierarchy diagram shows:

- Executive Staff**: CEO, President, CFO, VP, Sales. Description: * View & edit data, roll up forecasts, & generate reports for all users below. * Can't access data of other Executive Staff.
- Western Sales Director**: Director of W. Sales. Description: * View & edit data, roll up forecasts, & generate reports directly below. * Can't access data of others at same level.
- Eastern Sales Director**: Director of E. Sales. Description: * View & edit data, roll up forecasts, & generate reports directly below. * Can't access data of others at same level.
- International Sales Director**: Director of Int'l Sales. Description: * View & edit data, roll up forecasts, & generate reports directly below. * Can't access data of others above or at same level.
- Western Sales Rep**: CA Sales Rep, OR Sales Rep. Description: * View & edit data, roll up forecasts, & generate reports directly below. * Can't access data of others at same level.
- Eastern Sales Rep**: NY Sales Rep, MA Sales Rep. Description: * View & edit data, roll up forecasts, & generate reports directly below. * Can't access data of others at same level.
- International Sales Rep**: Asian Sales Rep, European Sales Rep. Description: * View & edit data, roll up forecasts, & generate reports directly below. * Can't access data of others above or at same level.

At the bottom right of the main content area are 'Set Up Roles' and 'Don't show this page again' buttons.

2. Click on Expand All and click on add role under SVP, Sales & Marketing role.
3. Give Label as "Purchasing Manager" and Role name gets auto populated. Then click on Save.

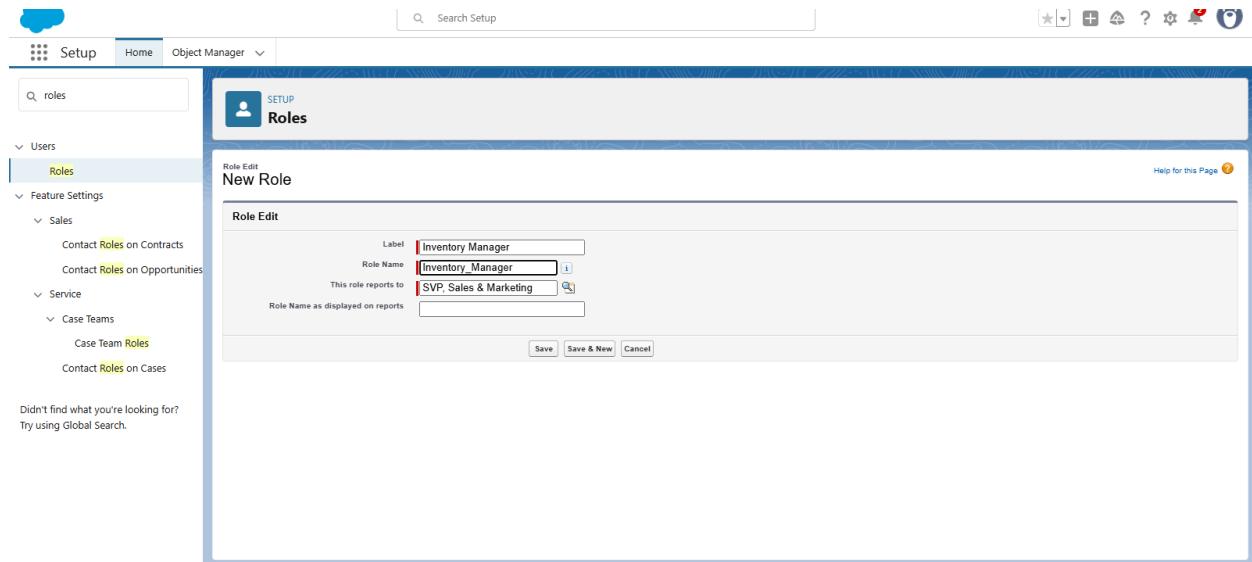
The screenshot shows the 'Role Edit' screen for creating a new role. The title is 'New Role'. The 'Role Edit' section contains the following fields:

- Label**: Purchasing Manager
- Role Name**: Purchasing_Manager
- This role reports to**: SVP, Sales & Marketing
- Role Name as displayed on reports**: (empty field)

At the bottom are 'Save', 'Save & New', and 'Cancel' buttons.

Activity 2 : Create a Purchasing Manager Role.

1. Go to quick find >> Search for Roles >> click on Set Up Roles.
2. Click on Expand All and click on add role under SVP, Sales & Marketing role.
3. Give Label as “Inventory Manager” and the Role name gets auto populated. Then click on Save.



Milestone 12 - Permission Sets

Activity 1 : Create a Permission Set.

1. Go to setup >> type Permission in quick find box >> Select Permission Set >> click on New.
2. Enter Label as Purchase Manager Create Access >> Click on Save.

The first screenshot shows the 'Permission Sets' page in the Setup interface. It lists various permission sets, including 'Purchase Manager Create Access' which is highlighted. The second screenshot shows the 'Create' page for a new permission set, where 'Purchase Manager Create Access' has been selected.

3. From Object Settings >> Select Order Item >> Enable for both Tab Available and Visible >> Enable Read and Create in Object Permissions >> Click on Save.
4. Navigate to the Permission Set detail page >> Click Manage Assignments >> Click Add Assignments >> Select the user John PurchaseM to assign the permission set to and click Next.
5. Select No Expiration date >> Click on Assign.

Milestone 13 - Flows

Activity 1 : Create Flow to update the Actual Delivery Date.

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow >> Start From Scratch .

The screenshot shows the Salesforce Setup interface with the 'Flow' category selected in the left sidebar. The main area displays a list of existing flow definitions, including various process types like Salesforce Scheduler Flow, Screen Flow, and Omni-Channel Flow. The list includes columns for Flow Label, Process Type, Active status, Template, Package State, and Last Modified date. A 'New Flow' button is visible in the top right corner of the main pane.

2. Select the record Triggered flow.Click on create.
3. Under Object select “Purchase Order”
4. Select A record is created or updated
5. Set Entry Conditions : None
6. Select Fast Field Updates and click on Done
7. Under the record trigger flow click on the “+” icon and select Get Records.
8. Enter Label as “ Get Purchase Record ”.
9. For Object select Purchase Order.
10. For Condition Requirements , select All Conditions are Met(AND)
For the first condition select as follows:
 - Field: Id
 - Operator: Equals
 - Value: {!\$Record.Id}
7. For How many Records to store Select Only the First Record.

8. For How to Store Record Data select Choose fields and let Salesforce do the rest. Select Field: Order_Date__c. Click on Done.
9. In the Flow Builder, click on the Manager tab on the left-hand side >> Click on New Resource >> In the Resource Type dropdown, select Variable.
10. Enter API name as ActualDeliveryDate >> Select Data type as Date >> Click on Done.
11. From the Toolbox drag and drop Assignment element.
12. Enter the label as “Assignment”.
13. Set Variable Values:
 - a) Variable : {!ActualDeliveryDate}
Operator : Equals
Value : {!\$Record.Order_Date__c}
 - b) Variable : {!ActualDeliveryDate}
Operator : Add
Value : 3
14. Click Done
15. From the Toolbox drag and drop Update Records element and connect to the Assignment element.
16. Enter the label as “Updating Purchasing Order”.
17. How to Find Records to Update and Set Their Values : Use the Purchase Order record that triggered the flow
18. Set Filter Conditions : None -Always Update Record
19. Set Field Values for the Trip Record as
 - Field : Actual_Delivery_Date__c
Value : {!ActualDeliveryDate}
20. Click Done
21. Save the flow as “Actual Delivery Date Updating”.
22. Activate the flow.

Milestone 14 - Triggers

Activity 1 : Create a Trigger to Calculate total amount on Order Item.

Step 1 : Login to Salesforce:

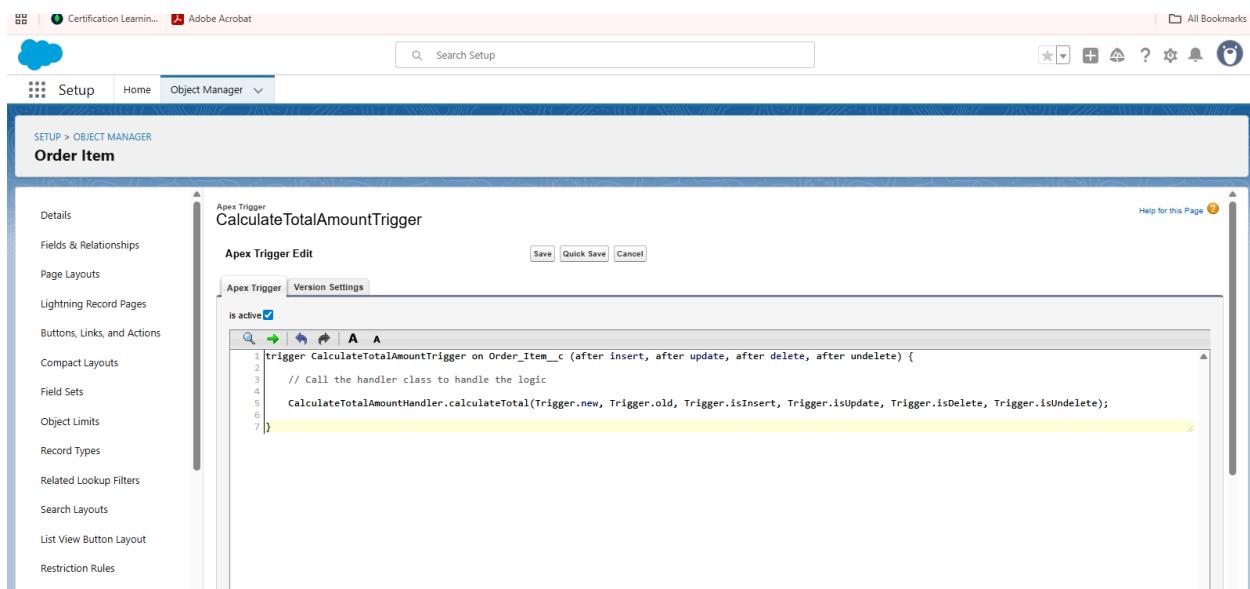
Log in to your Salesforce account with administrative privileges.

Step 2:

- i) Navigate to Setup: Once logged in, click on the gear icon ?? (Setup) located at the top-right corner of the page. This will open the Setup menu.
- ii) Click on Developer Console: Click on the "Developer Console" option from the Setup menu. This will open the Developer Console in a new browser tab or window.

Step 3:

- i) In the Developer Console window, go to the top menu and click on "File".
- ii) Select New: From the dropdown menu under "File", select "New".
- iii) Choose Apex Trigger: This will open a new Apex Trigger editor tab.



Create an Apex Trigger:

```
trigger CalculateTotalAmountTrigger on Order_Item__c (after insert, after update, after delete, after undelete) {
```

```
    // Call the handler class to handle the logic
    CalculateTotalAmountHandler.calculateTotal(Trigger.new, Trigger.old,
        Trigger.isInsert, Trigger.isUpdate, Trigger.isDelete, Trigger.isUndelete);
}
```

Step 4:

- i) In the Developer Console window, go to the top menu and click on "File".
- ii) Select New: From the dropdown menu under "File", select "New".
- iii) Choose Apex Class: Name it as CalculateTotalAmountHandler

```

public class CalculateTotalAmountHandler {

    // Method to calculate the total amount for Purchase Orders based on
    related Order Items
    public static void calculateTotal(List<Order_Item__c> newItems,
List<Order_Item__c> oldItems, Boolean isInsert, Boolean isUpdate, Boolean
isDelete, Boolean isUndelete) {

        // Collect Purchase Order IDs affected by changes in Order_Item__c
records
        Set<Id> parentIds = new Set<Id>();

        // For insert, update, and undelete scenarios
        if (isInsert || isUpdate || isUndelete) {
            for (Order_Item__c ordItem : newItems) {
                parentIds.add(ordItem.Purchase_Order_Id__c);
            }
        }

        // For update and delete scenarios
        if (isUpdate || isDelete) {
            for (Order_Item__c ordItem : oldItems) {
                parentIds.add(ordItem.Purchase_Order_Id__c);
            }
        }

        // Calculate the total amounts for affected Purchase Orders
        Map<Id, Decimal> purchaseToUpdateMap = new Map<Id, Decimal>();

        if (!parentIds.isEmpty()) {
            // Perform an aggregate query to sum the Amount__c for each
Purchase Order
            List<AggregateResult> aggrList = [
                SELECT Purchase_Order_Id__c, SUM(Amount__c) totalAmount
                FROM Order_Item__c
                WHERE Purchase_Order_Id__c IN :parentIds
                GROUP BY Purchase_Order_Id__c
            ];
        }
    }
}

```

```

];
// Map the result to Purchase Order IDs
for (AggregateResult aggr : aggrList) {
    Id purchaseOrderId = (Id)aggr.get('Purchase_Order_Id__c');
    Decimal totalAmount = (Decimal)aggr.get('totalAmount');
    purchaseToUpdateMap.put(purchaseOrderId, totalAmount);
}

// Prepare Purchase Order records for update
List<Purchase_Order__c> purchaseToUpdate = new
List<Purchase_Order__c>();
for (Id purchaseOrderId : purchaseToUpdateMap.keySet()) {
    Purchase_Order__c purchaseOrder = new Purchase_Order__c(Id =
purchaseOrderId, Total_Order_cost__c =
purchaseToUpdateMap.get(purchaseOrderId));
    purchaseToUpdate.add(purchaseOrder);
}

// Update Purchase Orders if there are any changes
if (!purchaseToUpdate.isEmpty()) {
    update purchaseToUpdate;
}
}
}
}

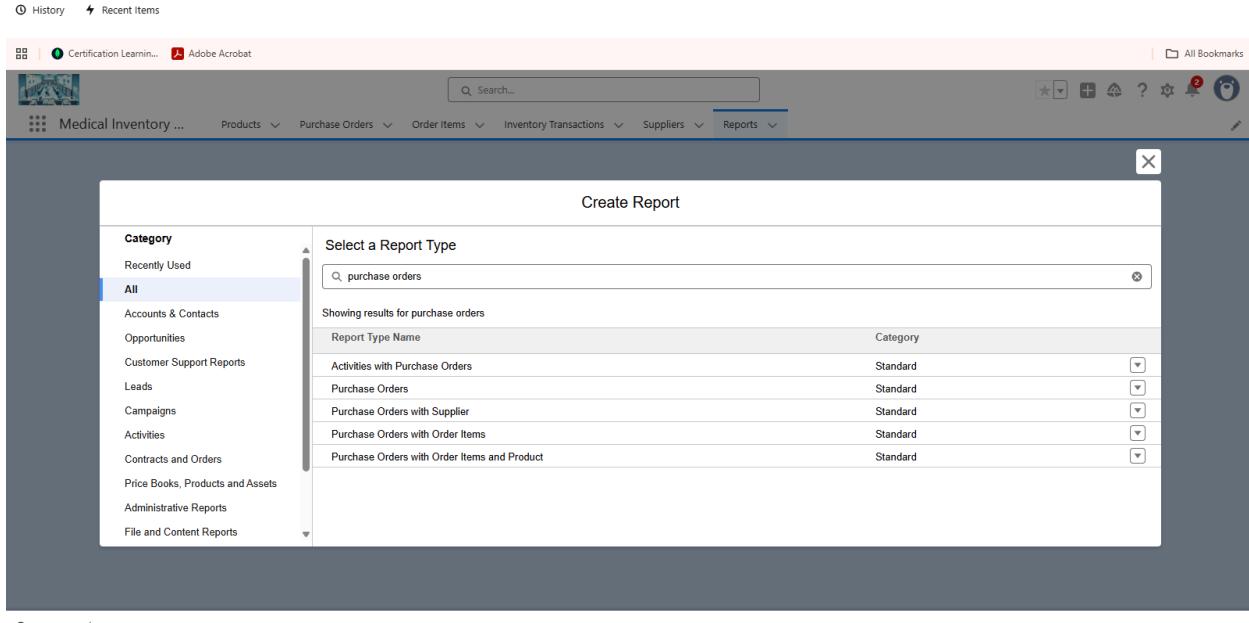
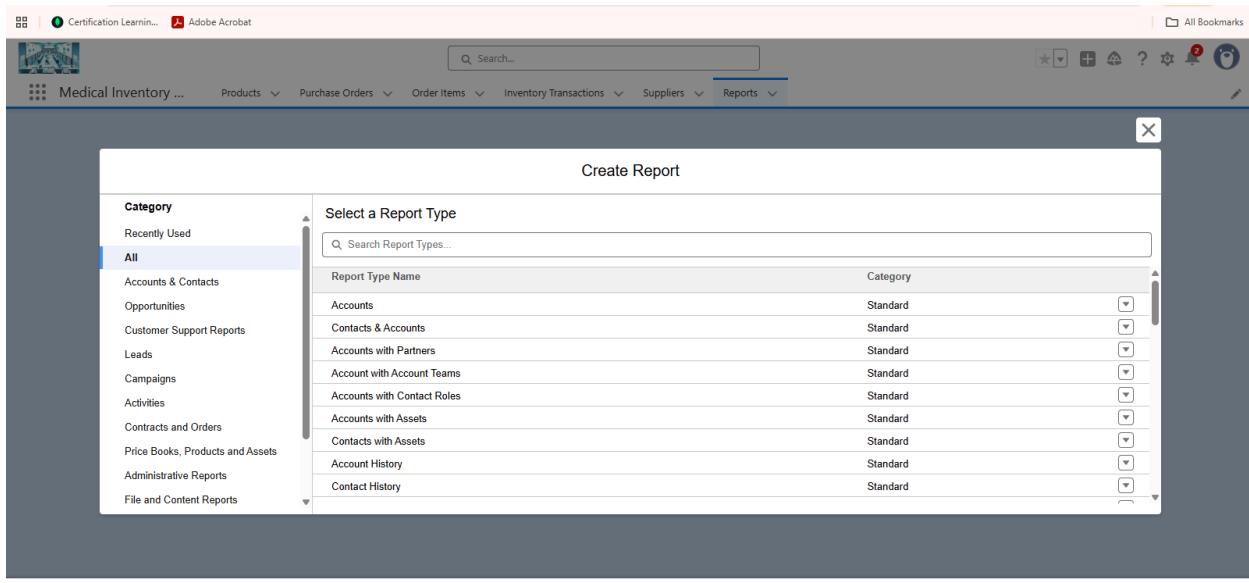
```

Save it.

Milestone 15 - Reports

Activity 1: Create a Purchase Orders based on Suppliers(Summary) Report

1. Click App Launcher
2. Select Medical Inventory Management App
3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders Click Start report.



6. Click on Filters and select as follows and click on Apply

7. Customize your report, in group rows select – Supplier ID, Purchase Order: Purchase Order ID, for columns Order Count, Total Order Cost (In this way we are making a Summary Report).
8. Click save and run
9. Give report name – Purchase Orders based on Suppliers.
10. Click Save

NOTE: In this report you can see your all record of the object you selected for reporting

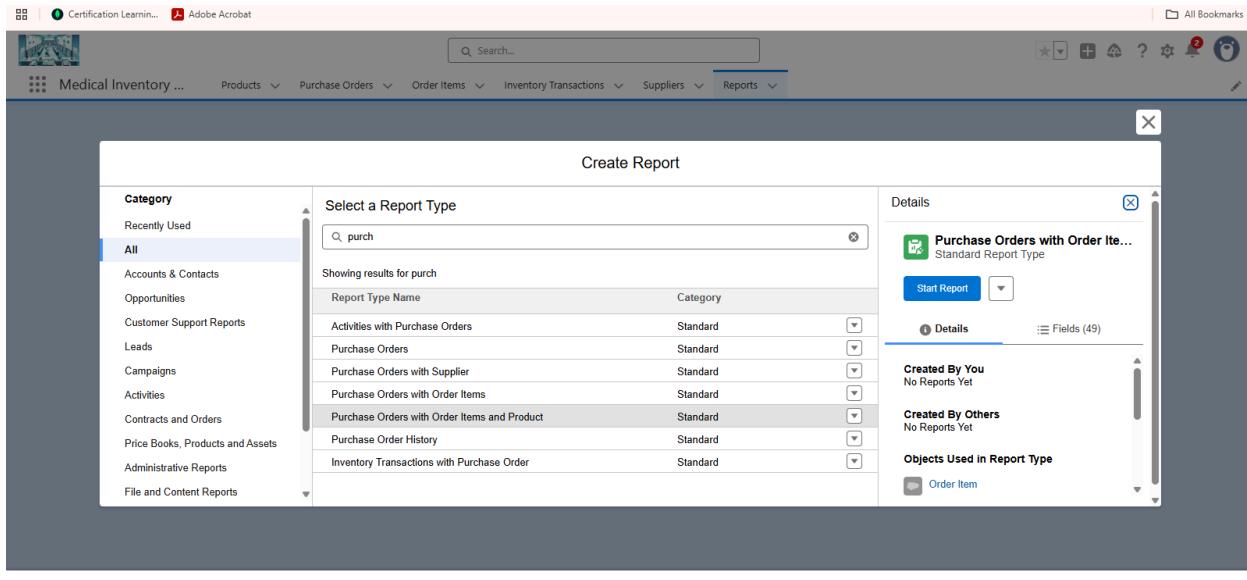
What you selects in “Select a report type option”)

(View Report

1. Click on App Launcher on the left side of the screen.
2. Search Medical Inventory Management App & click on it.
3. Click on Reports Tab.
4. Click on Purchase Orders based on Suppliers and see records.

Activity 2: Create a Complete Purchase Details Report

1. Click App Launcher
2. Select Medical Inventory Management App
3. Click on Reports tab
4. Click on New Report.
5. Click the report type as Purchase Orders with Order Items and Product ID
>> Click Start report.
6. Click on Filters and select as follows and click on Apply



① History ⏪ Recent items

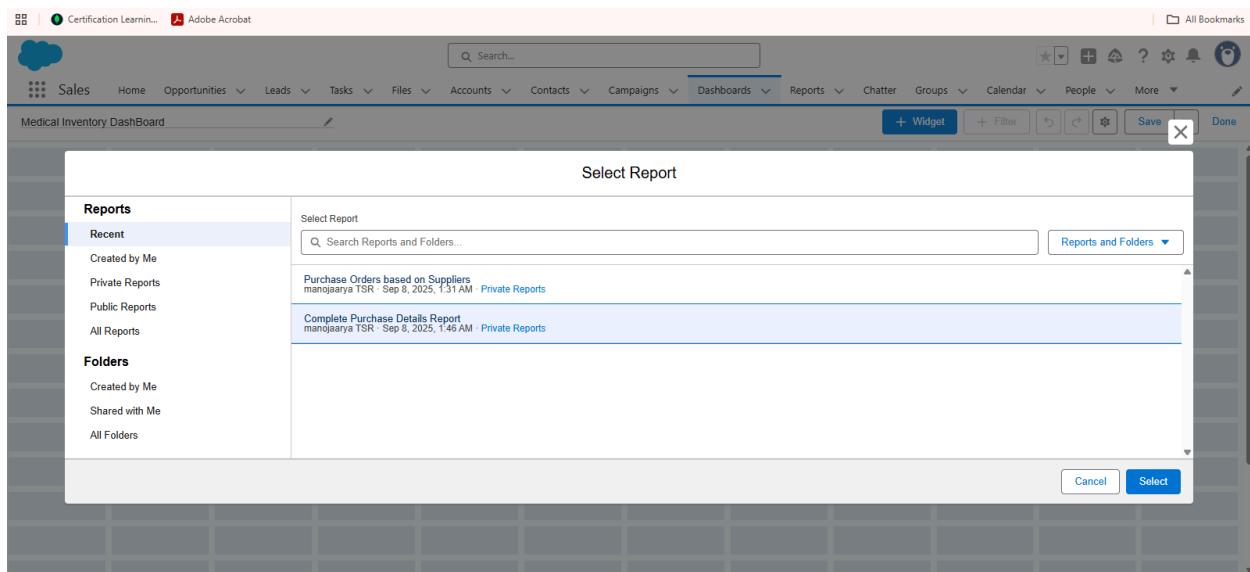
7. Customize your report, in group rows select – Supplier ID, Actual Delivery Date, Purchase Order: Purchase Order ID, for columns Product ID : Product ID, Product ID : Product Name, Order Count, Quantity Received, Amount (In this way we are making a Summary Report).
8. Click save and run
9. Give report name – Complete Purchase Details Report
10. Click Save.

Milestone 16 - Dashboards

Activity 1: - Create Dashboard

1. Click on the Dashboards tab from the Medical Inventory Management application.
2. Click on the new dashboard.
3. Give name - Medical Inventory DashBoard
4. Click create
5. Click on +widget
6. Select the Purchase Orders based on Suppliers Report
7. For the data visualization select any of the charts, tables etc. as per your choice/requirement

8. Click add.
9. Click save.



Add Widget

Report

Purchase Orders based on Supplie X

Use chart settings from report i

Display As

Value

Sum of Total Order Cost

Sliced By

Supplier ID

Display Units

Preview

Purchase Orders based on Suppliers

Sum of Total Order Cost

Supplier ID

Supplier-001

Supplier-002

View Report (Purchase Orders based on Suppliers)

Cancel
Add

Activity 2: View Dashboard

1. Click on App Launcher on the left side of the screen.
2. Search Medical Inventory Management & click on it.
3. Click on Dashboard Tab.
4. Click on Medical Inventory DashBoard see graph view of records

The top screenshot shows the Salesforce App Launcher search results for 'Medical Inventory Management'. The search bar at the top has 'medical' typed into it. Below the search bar, there is a list of apps under 'Most Recently Used' with 10 items. The list includes 'CalculateTotalAmountHandler' (Apex Class), 'CalculateTotalAmountTrigger' (Apex Trigger), and 'manojaarya TSR' (User). The bottom screenshot shows the 'Medical Inventory Dashboard'. At the top, there is a navigation bar with tabs for 'Products', 'Purchase Orders', 'Order Items', 'Inventory Transactions', 'Suppliers', 'Reports', and 'Dashboards'. The 'Dashboards' tab is currently selected. Below the navigation bar, there is a section titled 'Purchase Orders based on Suppliers' featuring a donut chart. The chart shows a total order cost of ₹26k, with a breakdown of ₹4.5k and ₹22k. It also lists 'Supplier ID' with entries for 'Supplier-001' and 'Supplier-002'.

Conclusion:

The **Medical Inventory Management System** successfully automates and streamlines inventory processes in a healthcare environment. By leveraging Salesforce CRM features, the project ensures efficiency, accuracy, and transparency in managing medical supplies. This project demonstrates the practical application of Salesforce in solving real-world problems under the **Naan Mudhalvan initiative**.