

# Medical Inventory Management

COLLEGE NAME: KUMARAGURU COLLEGE OF LIBRAL ARTS AND SCIENCE.

COLLEGE CODE: bruax

TEAM ID: NM2025TMID25550

TEAM SIZE: 4

**TEAM LEADER:** DEVADARSHINI P  
Email: [devadarshini027@gmail.com](mailto:devadarshini027@gmail.com)

**Team Member 1:** BHAVADHARANI D  
Email: [bhavadharani.23bds@kclas.ac.in](mailto:bhavadharani.23bds@kclas.ac.in)

**Team Member 2:** NETHRA K  
Email: [nethravamsi05@gmail.com](mailto:nethravamsi05@gmail.com)

**Team Member 3:** MAHALAKSHMI M  
Email: [mahalakshmi.23bds@klclas.ac.in](mailto:mahalakshmi.23bds@klclas.ac.in)

# Title: Medical Inventory Management



## Project Overview:

The **Medical Inventory Management System**, built on Salesforce CRM, is designed to handle the movement of medical equipment, drugs, and supplies with greater efficiency. It simplifies tasks such as tracking inventory levels, managing vendor relationships, and processing purchase orders. By automating these operations, the system helps cut down on manual work, reduces the chances of mistakes, and guarantees that critical healthcare resources remain available when needed.

## Objectives:

- Maintain a **single, unified database** of all medical supplies and equipment.
- **Automate** the process of creating purchase orders and routing them for approval.
- Keep detailed records of **suppliers, past transactions, and order history**.

- **Track inventory levels** in real time and send notifications for items that are low in stock or nearing expiration.
- Provide **interactive dashboards and reports** to support management decisions.
- Leverage **Salesforce features** such as workflows, triggers, and validation rules for automation.
- Ensure **secure, role-based permissions** tailored to different users (Administrators, Inventory Supervisors, and Staff).

## **Project Scope**

### **Included in Scope:**

- Monitoring and managing stock levels for medical supplies, drugs, and equipment.
- Handling supplier information along with purchase order creation and approvals.
- Implementing user access controls through role-based security profiles.
- Utilizing Salesforce automation tools (workflows, triggers, approval mechanisms) to streamline operations.
- Building dashboards and analytical reports to provide real-time insights.

### **Excluded from Scope:**

- Integration with third-party hospital billing or financial systems.
- Use of advanced artificial intelligence or machine learning techniques for demand forecasting.
- Development of a mobile application to support offline inventory management.

## **System Modules**

1. Inventory Control – Enables adding, updating, and tracking of all medical supplies and equipment.
2. Vendor Management – Stores supplier information and maintains transaction history.

3. Order Processing – Supports automated purchase order generation, approval workflows, and expense calculation.
4. Stock Oversight – Provides notifications for low inventory levels and items nearing expiration.
5. Analytics & Reporting – Offers both detailed and summary reports through interactive dashboards.
6. Access Management – Implements secure role-based permissions for administrators, managers, and staff.

## **Project Deliverables**

- A Salesforce-based Medical Inventory Management application.
- Creation of custom objects such as Medical Items, Suppliers, Purchase Orders, and Order Line Items.
- Implementation of automation features including validation rules, workflows, and triggers.
- Development of dashboards and reports for monitoring inventory levels and purchase order activities.
- Configuration of user roles and profiles (Administrator, Inventory Supervisor, and General User).

## **Key Benefits**

- Optimized inventory control minimizes waste and ensures timely availability of medical supplies.
- Automation of purchase orders, cutting down repetitive manual tasks and saving staff time.
- Enhanced supplier transparency and accountability through detailed records and order history. Instant visibility into stock levels and expenses with real-time tracking and reporting.

## **Student Learning Outcomes**

Upon completing this project, students were able to:

1. Gain Knowledge of Salesforce CRM
  - Acquired practical exposure in tailoring Salesforce to address real-world scenarios.
2. Apply Cloud Development Techniques
  - Built custom objects, fields, page layouts, and defined relationships among them.
3. Implement Process Automation
  - Designed and applied workflows, validation rules, triggers, and approval mechanisms.
4. Create Reports and Dashboards
  - Developed visual dashboards and analytical reports to track suppliers, stock status, and purchase orders.
5. Configure Role-Based Security
  - Set up user profiles and permissions for administrators, managers, and general staff.
6. Enhance Analytical & Problem-Solving Skills
  - Explored challenges in healthcare inventory management and provided Salesforce-driven solutions.
7. Strengthening Documentation & Presentation Abilities
  - Produced well-structured technical and functional project documentation.
8. Improve Teamwork & Project Coordination (*for group projects*)

- Practiced collaboration, task allocation, and meeting project timelines effectively.

## 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).

| <b>Phase No.</b> | <b>Phase Name</b>                                 | <b>Description</b>   | <b>Page No.</b> |
|------------------|---|--|-----------------|
| <b>1</b>         | 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. |                 |
| <b>2</b>         | 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.               |                 |
| <b>3</b>         | 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.  |                 |

|          |   |   |  |
|----------|---|---|--|
| <b>4</b> | Data Migration, Testing & Security      | Creating Users, Profiles (Admin, Inventory Testing & Security Manager, Standard User), Permission Sets, Public Groups and Sharing Rules; configuring Report Types, Reports and Dashboards; testing functionalities for accuracy |  |
| <b>5</b> | 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

During this stage, the project team focused on collecting, reviewing, and documenting the requirements for the Medical Inventory Management System. The primary goal was to gain a clear understanding of the expectations of healthcare professionals, administrators, and inventory coordinators. This analysis served as the foundation for designing a solution that improves inventory control, minimizes manual effort, and ensures accuracy in stock management.

## **Activities Performed:**

### **1. Requirement Gathering:**

- Gathered feedback from hospital staff, administrators, and inventory supervisors regarding difficulties in handling medical supplies and equipment.
- Highlighted issues such as frequent stockouts, manual handling of purchase orders, missing supplier history, and slow approval processes.

### **2. Defining Scope and Objectives**

- Clarified the main objectives of the project, which include efficient inventory tracking, supplier management, automation of purchase orders, and timely stock notifications.
- Specified exclusions such as billing system integration and AI-based demand prediction to clearly mark the project boundaries.

### **3. Data Model Planning**

- Determined the key entities: Medical Item, Supplier, Purchase Order, and Order Item.
- Established relationships among these objects to enable seamless system operations.

### **4. Workflow Planning**

- Outlined workflows for routine activities like generating purchase orders, approving supplier records, and monitoring item expiry.
- Planned automation through Salesforce tools such as Validation Rules, Triggers, and Flows.

### **5. Project Planning**

- Estimated the total effort as 31 hours.
- Divided the work into five phases: Requirement Analysis, Development, UI Customization, Testing & Security, and Deployment & Documentation.

- Defined deliverables and timelines for each phase to ensure structured execution.

## Outcome of Phase 1

- Achieved a comprehensive understanding of the system requirements.
- Established a clearly defined scope along with specific project objectives.
- Prepared an initial draft of the data model and workflow structure.
- Created a structured roadmap to guide the project's execution.

## Phase 2: Salesforce Development – Backend & Configurations

### Milestone 1: Salesforce developer account creation.

#### Activity 1: Setting Up a Salesforce Developer Account

To begin working on the project, a Salesforce Developer Org needs to be created. This provides a free environment for building and testing applications.

**Sign up for your Developer Edition**

A free Salesforce Platform environment with Agentforce and Data Cloud

|                    |                           |
|--------------------|---------------------------|
| First name         | Last name                 |
| Devadarshini       | P                         |
| Job title          | Work email                |
| Student            | devadarshini027@gmail.com |
| Company            | Country/Region            |
| Kumaraguru college | India                     |

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

**Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.**

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

#### Steps to create the account:

- Visit: <https://developer.salesforce.com/signup>
- Fill in the required details:

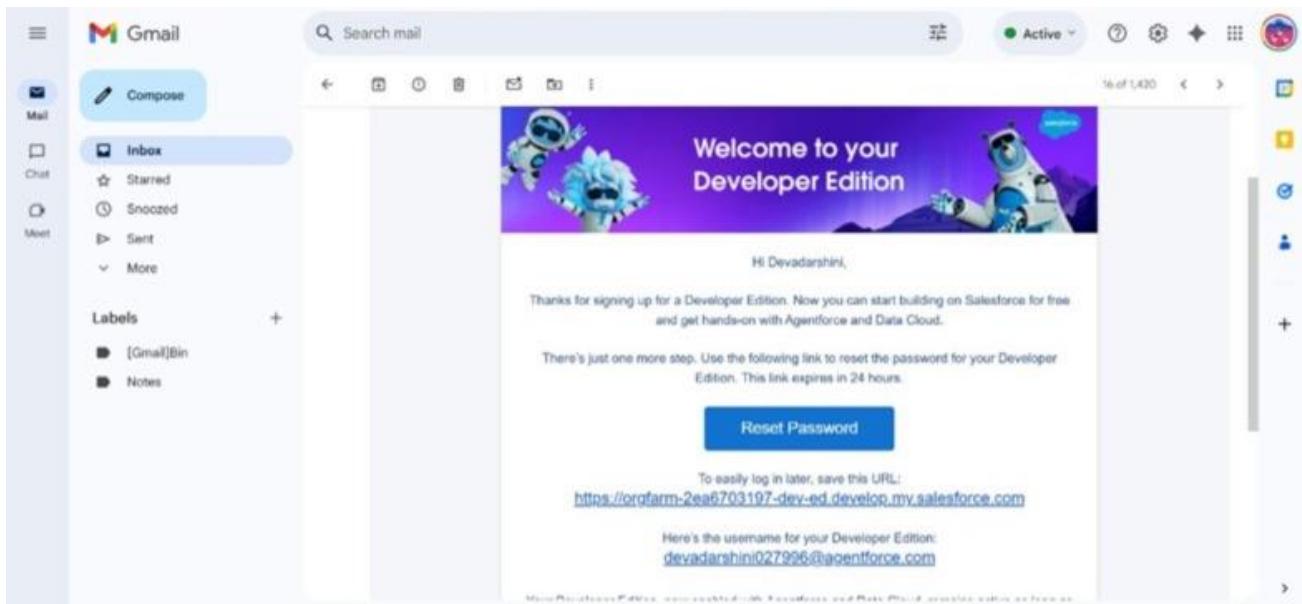
- **First Name & Last Name** – Enter your full name.
- **Email** – Provide a valid email ID.
- **Role** – Select *Developer*.
- **Company** – Enter your college name.
- **Country** – Choose *India*.
- **Postal Code** – Enter your area PIN code.
- **Username** – Must be unique; use a format combining your name and institution

## ACTIVITY 2: Account Activation

After signing up, the developer account must be activated to gain access.

### Steps to activate:

1. Open the inbox of the email address used during registration. Look for the Salesforce account activation email (this may take **5–10 minutes** to arrive).
2. Click on the **Reset Password** link provided in the email.
3. Set a **new password**, choose and answer a **security question**, then confirm by clicking **Change Password**.
4. Then you will redirect your **salesforce setup page**.



## **Milestone 2 – Objects**

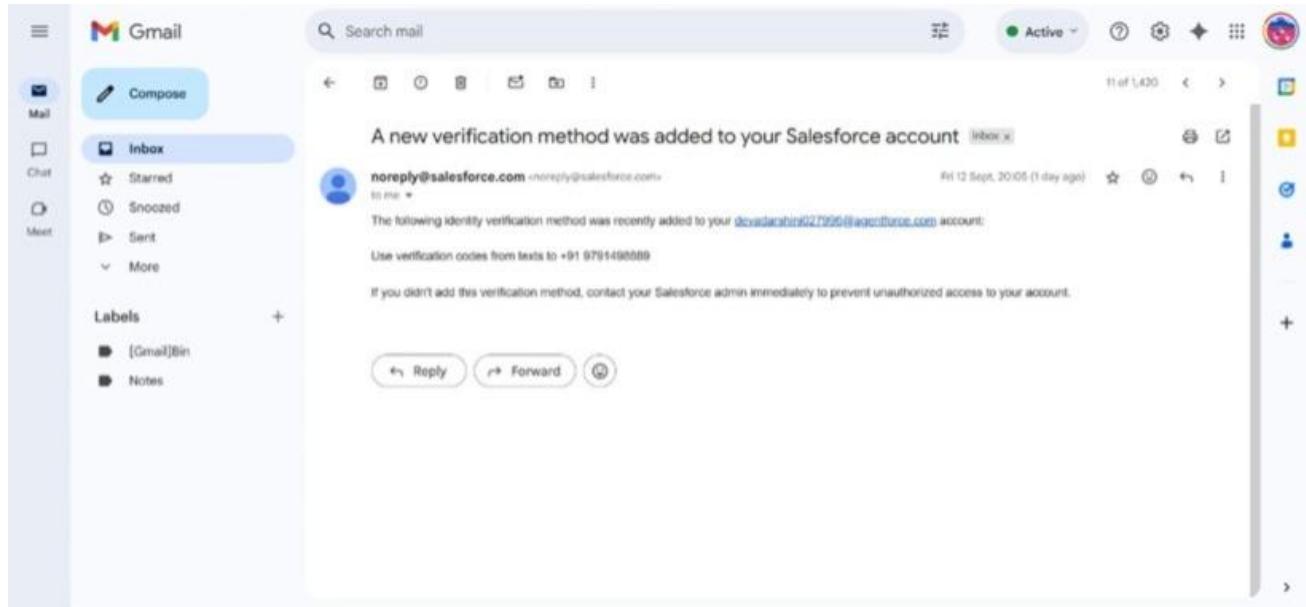
### **Activity 1: Creating a Product Object**

**To create a new custom object in Salesforce, follow these steps:**

1. Navigate to the Setup page.
2. Click on Object Manager.
3. Select Create → Custom Object.
4. Enter the Label Name as *Product*.
5. Enter Plural Label as *Products*.
6. Set the Record Name field to *Product ID*.
7. Choose Data Type as *Text*.
8. Check the option Allow Reports.
9. Check the option Allow Search.
10. Click Save & New to proceed.

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 your salesforce setup page.

A screenshot of the Salesforce Setup Home page. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar labeled 'Search Setup'. The main content area is titled 'SETUP Home' and features three cards: 'Get Started with Einstein Bots' (with a 'Get Started' button), 'Mobile Publisher' (with a 'Learn More' button), and 'Real-time Collaborative Docs' (with a 'Get Started' button). On the left, a sidebar lists various setup tools: 'Setup Home', 'Service Setup Assistant', 'Multi-Factor Authentication Assistant', 'Release Updates', 'Lightning Experience Transition Assistant', 'Salesforce Mobile App', 'Lightning Usage', 'Optimizer', and 'ADMINISTRATION' (which has a 'Users' sub-item). The top right corner of the screen shows a notification badge with the number '8'.

## Milestone 3- Tabs

### Activity 1: Creating a Tab for the Product Object

To make the Product object easily accessible, create a custom tab by following these steps:

1. Go to the Setup page and type Tabs in the Quick Find bar.
2. Click on Tabs.
3. Under the *Custom Object Tabs* section, click New.
4. Select the Product object and choose a suitable Tab Style.
5. Click Next. On the *Add to Profiles* page, leave the default settings and continue.
6. On the *Add to Custom App* page, uncheck Include Tab.
7. Ensure that the option Append tab to user's existing personal customizations is selected.
8. Click Save to complete the setup.

### 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. The left sidebar has 'User Interface' expanded, with 'Tabs' selected. The main content area has three sections:

- Custom Tabs**: A table showing custom object tabs. One row is selected for 'Inventory\_Transactions' with a 'Box' style icon.
- Web Tabs**: A table showing no tabs have been defined.
- Visualforce Tabs**: A table showing no tabs have been defined.

## Milestone 4- The Lightning App

### Activity 1: Creating a Lightning App for Medical Inventory Management

To set up a custom Lightning App for the project, follow these steps:

1. Go to Setup, type App Manager in the Quick Find box, and select App Manager.
2. Click New Lightning App.
3. Enter Medical Inventory Management as the App Name. Upload an appropriate image related to medical inventory, then click Next.
4. In App Options, leave the default selections and proceed by clicking Next.
5. For Utility Items, keep the defaults and click Next.
6. From Available Items, select the following and move them to Selected Items:
  - o Products
  - o Purchase Orders
  - o Order Items
  - o Inventory Transactions
  - o Suppliers
  - o Reports
  - o Dashboards
 Click Next.

7. From Available Profiles, choose System Administrator and move it to Selected Profiles.
8. Click Save & Finish to complete the app creation.

The screenshot shows two overlapping pages from the Salesforce Lightning App Builder.

**New Lightning App Configuration:**

- App Name:** Medical Inventory Management
- Developer Name:** Medical\_Inventory\_Management
- Description:** Enter a description...
- Image:** A placeholder image showing medical supplies.
- Primary Color Hex Value:** #0070D2
- Org Theme Options:** Use the app's image and color instead of the org's custom theme (unchecked).
- App Launcher Preview:** Shows the app's entry in the App Launcher.
- Next:** A blue button at the bottom right.

**Navigation Items Configuration:**

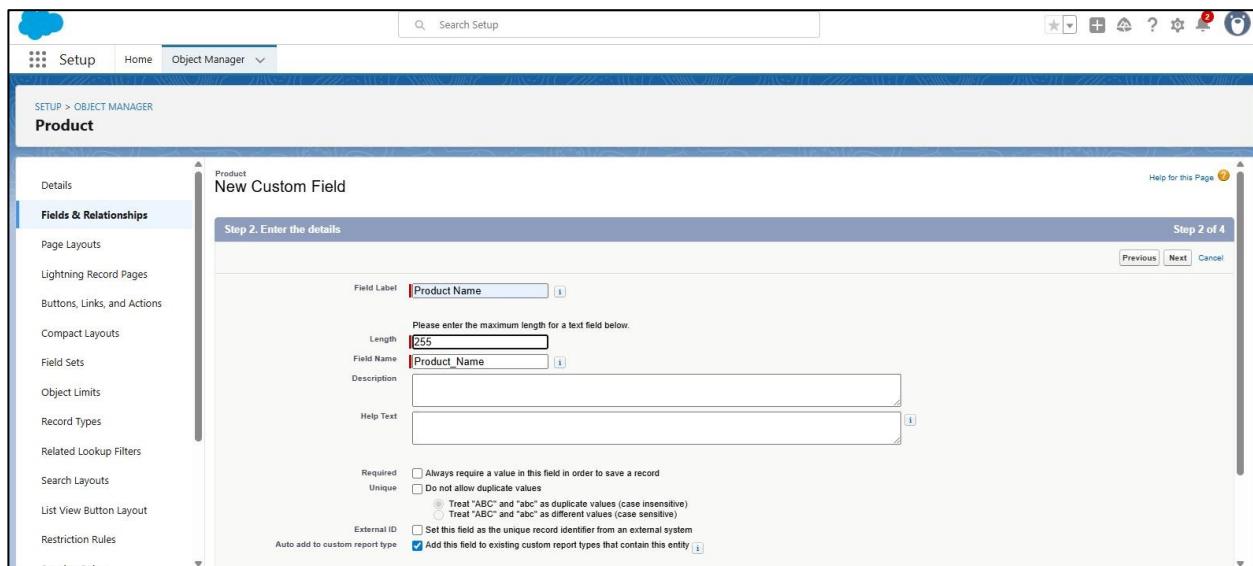
- Available Items:** A list of navigation items including Accounts, Activation Targets, Activations, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests, and more.
- Selected Items:** A list of items currently assigned to the app, including Products, Purchase Orders, Order Items, Inventory Transactions, Suppliers, and Reports.
- Buttons:** Up and down arrows to rearrange the selected items.

## 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 for creating a new custom field. The left sidebar is titled 'Fields & Relationships' under 'Object Manager'. The main area is titled 'New Custom Field' for the 'Product' object. The step being completed is 'Step 2. Enter the details'. The field is named 'Product\_Description' and is of type 'Text'. It is marked as required and checked for 'Auto add to custom report type'. A formula editor is available for default values.

The screenshot shows the list of field types available for the Product object. The 'Text Area' type is selected. Other options include Currency, Date, DateTime, Email, Golocation, Number, Percent, Phone, Picklist, Picklist (Multi-Select), Text, Text Area, Text Area (Long), Text Area (Rich), Text (Encrypted), Time, and URL. Each field type has a brief description of its functionality.

## 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”

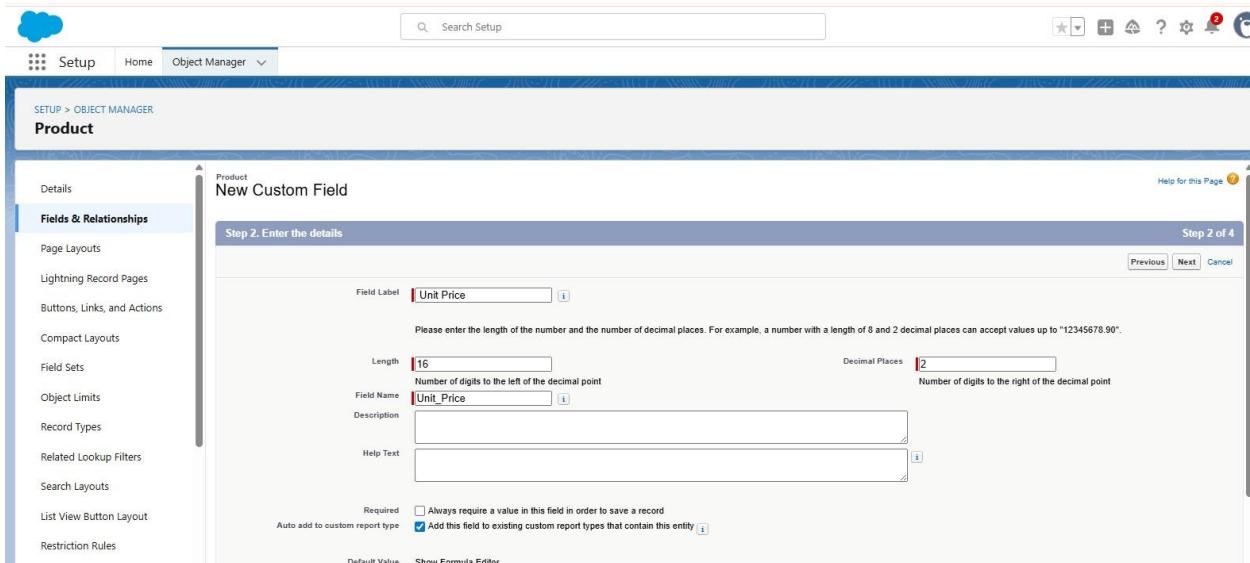
3. Click on New.
4. Select Data type as “Number” and click Next.
5. Enter Field Label as “Current Stock Level”.
6. Length - 18, Decimal Places - 0.
7. Click on Next, Next and Save.

The screenshot shows the Salesforce Setup interface for creating a new custom field. The left sidebar shows the 'Fields & Relationships' section selected. The main area is titled 'New Custom Field' under 'Step 2. Enter the details'. The 'Field Label' is set to 'Current Stock Level'. The 'Length' is set to 18, and 'Decimal Places' is set to 0. The 'Field Name' is 'Current\_Stock\_Level'. There are sections for 'Description' and 'Help Text', both of which are currently empty. At the bottom, there are several optional checkboxes under categories like 'Required', 'Unique', 'External ID', and 'AI Prediction'. A note at the bottom says 'Add this field to existing custom record types that contain this object'.

## 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 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 Next, Next and Save

## 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

### **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.

**SETUP > OBJECT MANAGER**

### Order Item

**Fields & Relationships**

**Data Type**

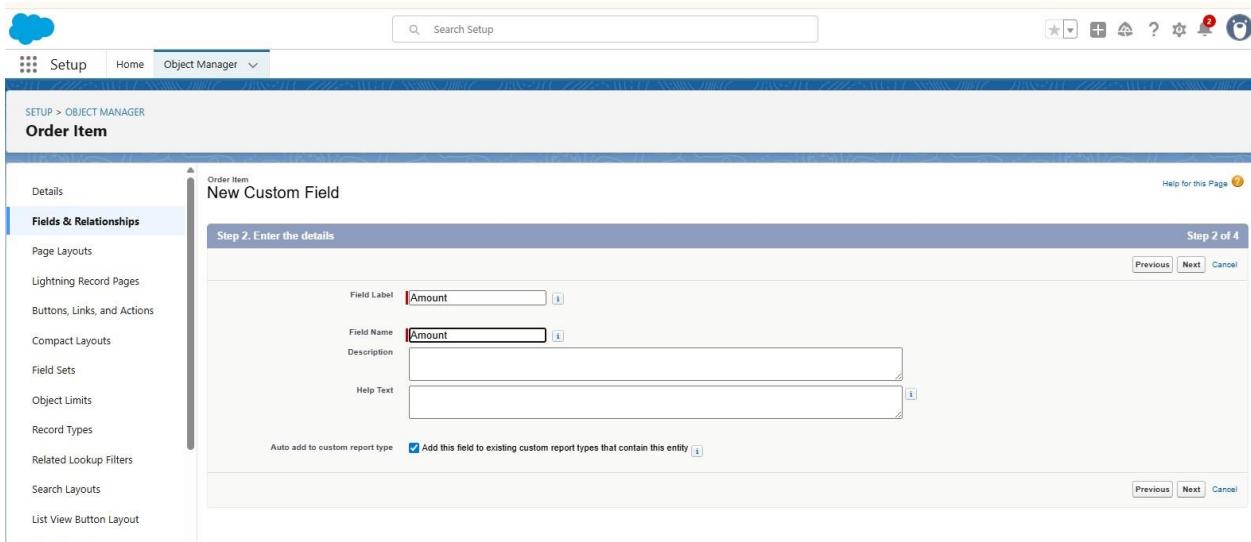
- None Selected Select one of the data types below.
- Auto Number A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.
- Formula A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.
- Roll-Up Summary 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.
- Lookup Relationship Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.
  - The relationship field is required on all detail records.
  - The ownership and sharing of a detail record are determined by the master record.
  - When a user deletes the master record, all detail records are deleted.
  - You can create rollup summary fields on the master record to summarize the detail records.
- Master-Detail Relationship Creates a specific type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:
  - The relationship field is required on all detail records.
  - The ownership and sharing of a detail record are determined by the master record.
  - When a user deletes the master record, all detail records are deleted.
  - You can create rollup summary fields on the master record to summarize the detail records.
- External Lookup Relationship Creates a relationship that links this object to an external object whose data is stored outside the Salesforce org.

- Checkbox Allows users to select a True (checked) or False (unchecked) value.
- Currency Allows users to enter a dollar or other currency amount and automatically formats the field as a currency amount. This can be useful if you export data to Excel or another spreadsheet.
- Date Allows users to enter a date or pick a date from a popup calendar.
- DateTime Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.
- Email Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.

## 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:

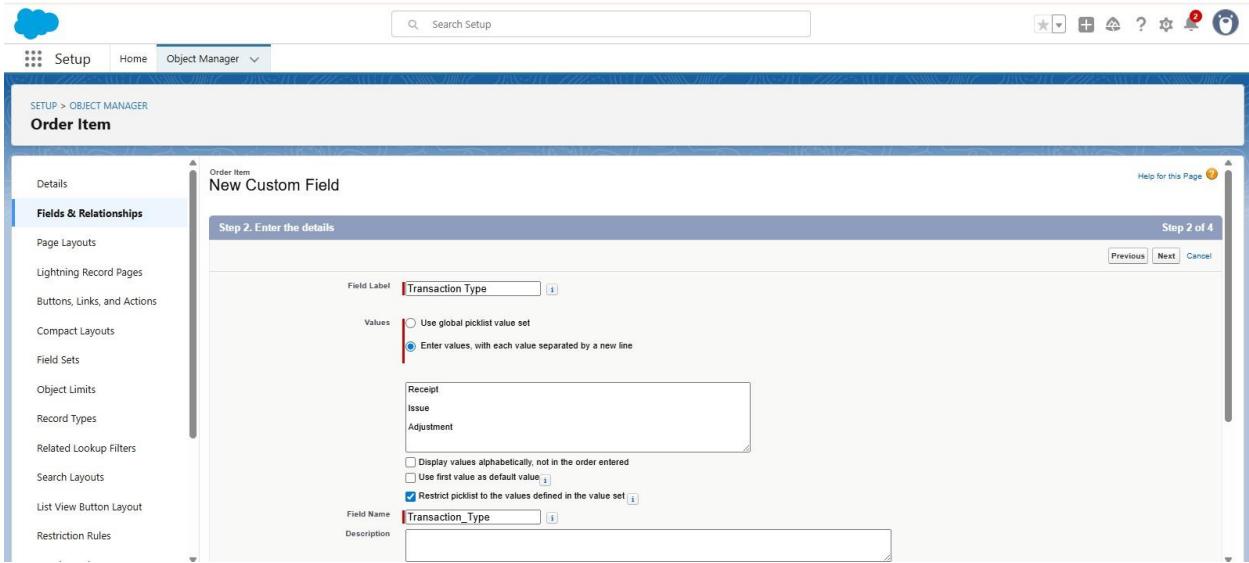
1. Go to setup >> click on Object Manager >> type object name (Inventory Transaction) in quick find box>> click on the Inventory Transaction Object.
2. Now click on “Fields & Relationships”.
3. Click on New.
4. Select Data type as “Picklist” and click Next.
5. Enter Field Label as “Transaction Type”.
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. In the top left, there's a blue cloud icon. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar labeled 'Search Setup'. On the right side of the header are various icons for account management and help.

The main content area is titled 'SETUP > OBJECT MANAGER' and 'Order Item'. A sidebar on the left lists options like 'Details', 'Fields & Relationships', and various layout and action types. The 'Fields & Relationships' section is currently selected.

The main panel is titled 'New Custom Field' and is on 'Step 2. Choose output type'. It shows a field label 'Order Cost' and a field name 'Order\_Cost'. There's a checkbox 'Auto add to custom report type' with a note 'Add this field to existing custom report types that contain this entity'. Below this is a 'Formula Return Type' section with several options: 'None Selected' (selected), 'Checkbox', 'Currency' (selected), 'Date', 'DateTime', 'Number', and 'Percent'. Each option has a brief description and examples.

## 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.

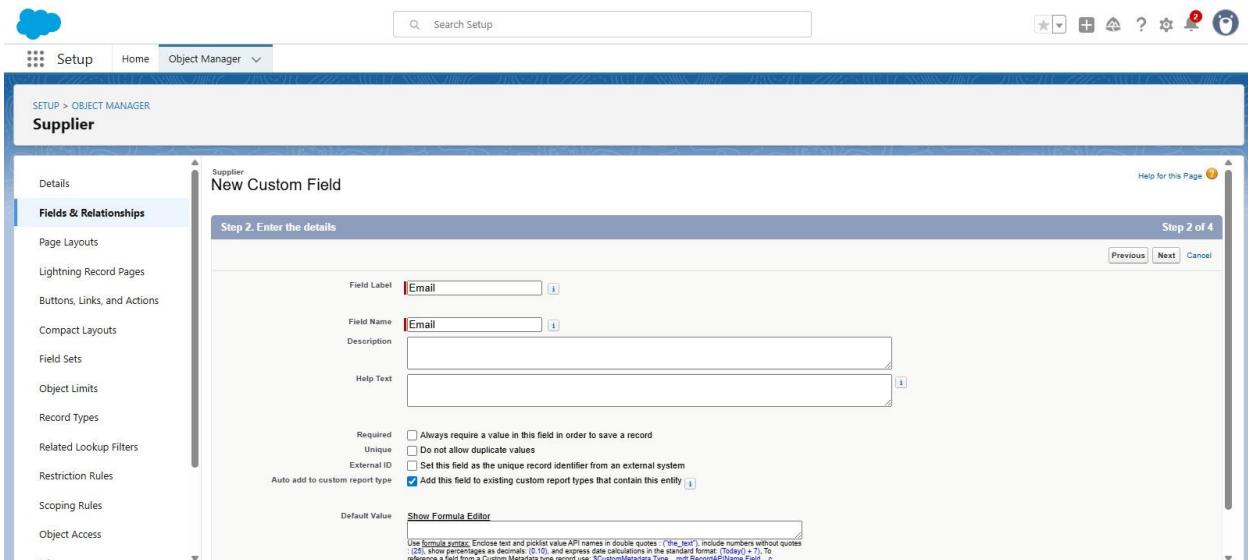
This screenshot shows the same Salesforce Setup interface as the previous one, but for the 'Supplier' object instead of 'Order Item'. The steps are identical: creating a new custom field, choosing the 'Text' data type, and entering 'Phone Number' as the field label and name.

The main panel is titled 'Supplier New Custom Field' and is on 'Step 2. Enter the details'. It shows a field label 'Phone Number' and a field name 'Phone\_Number'. There are fields for 'Description' and 'Help Text'. Under 'Required', there's a checkbox 'Always require a value in this field in order to save a record' which is checked. There's also a checkbox 'Add this field to existing custom report types that contain this entity'. At the bottom, there's a 'Show Formula Editor' link and a note about formula syntax.

## Activity 13: Creating a Email Field in Supplier object

To create fields in an object:

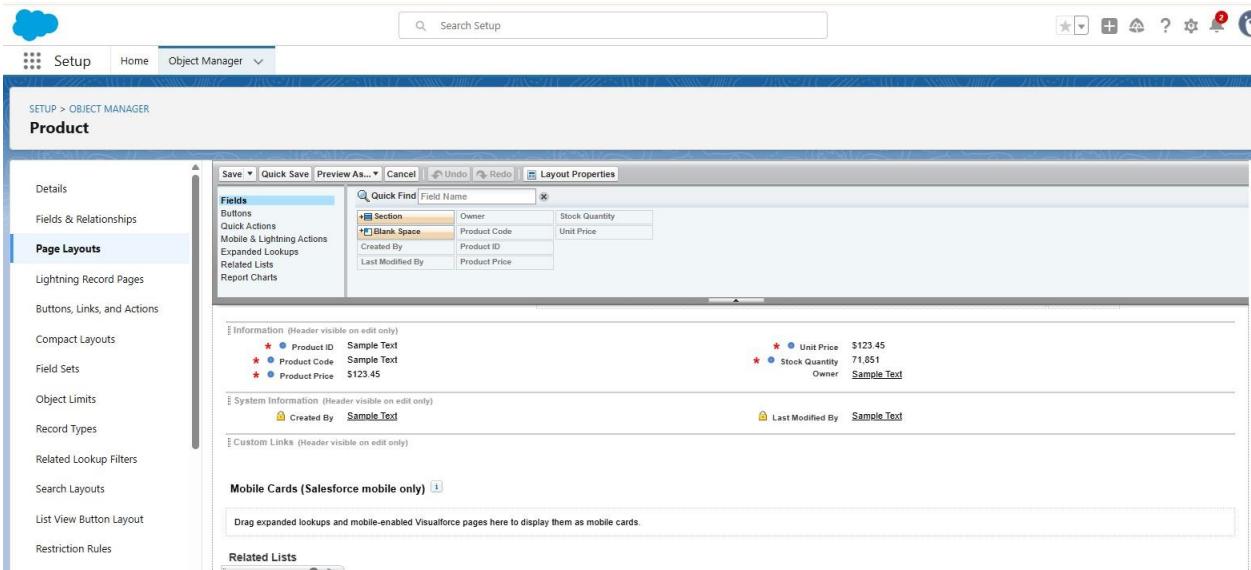
1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box>> click on the Supplier object.
2. Now click on “Fields & Relationships”
3. Click on New.
4. Select Data type as “Email” and click Next.
5. Enter the Field Label as “ Email”.
6. Click on Next, Next and Save.



## Milestone 6 -Editing of Page Layouts

### Activity 1: To edit a Page Layout in Product Object

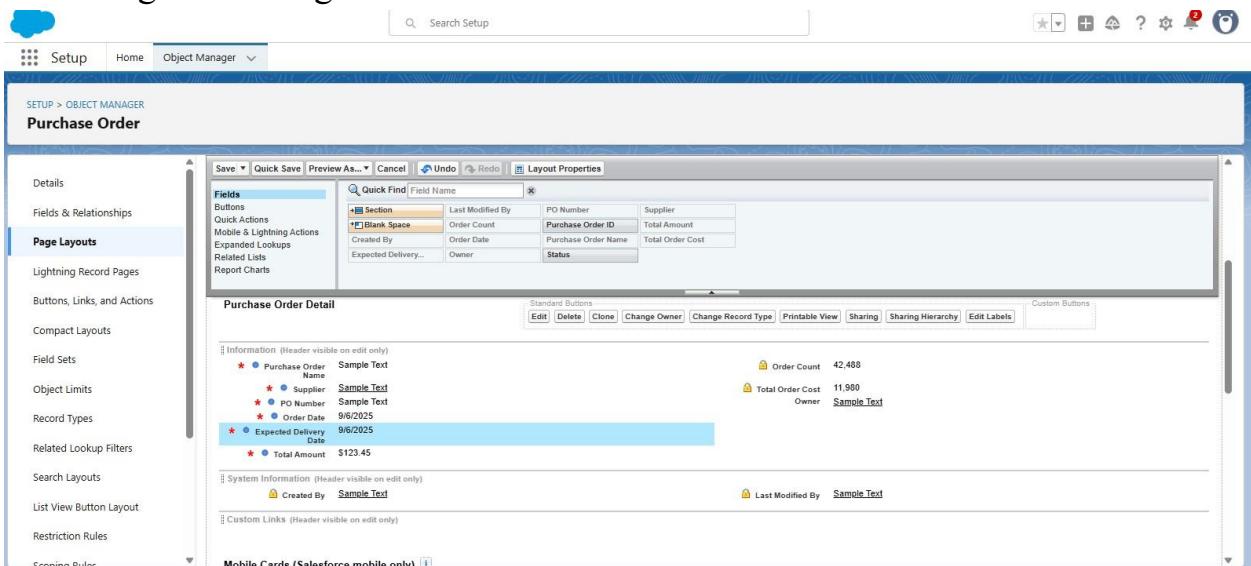
1. Go to setup >> click on Object Manager >> type object name(Product) in quick find box >> click on the Product object >> Page Layouts .
2. Click on the Product Layout.
3. Drag and arrange the field as shown below.



- 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

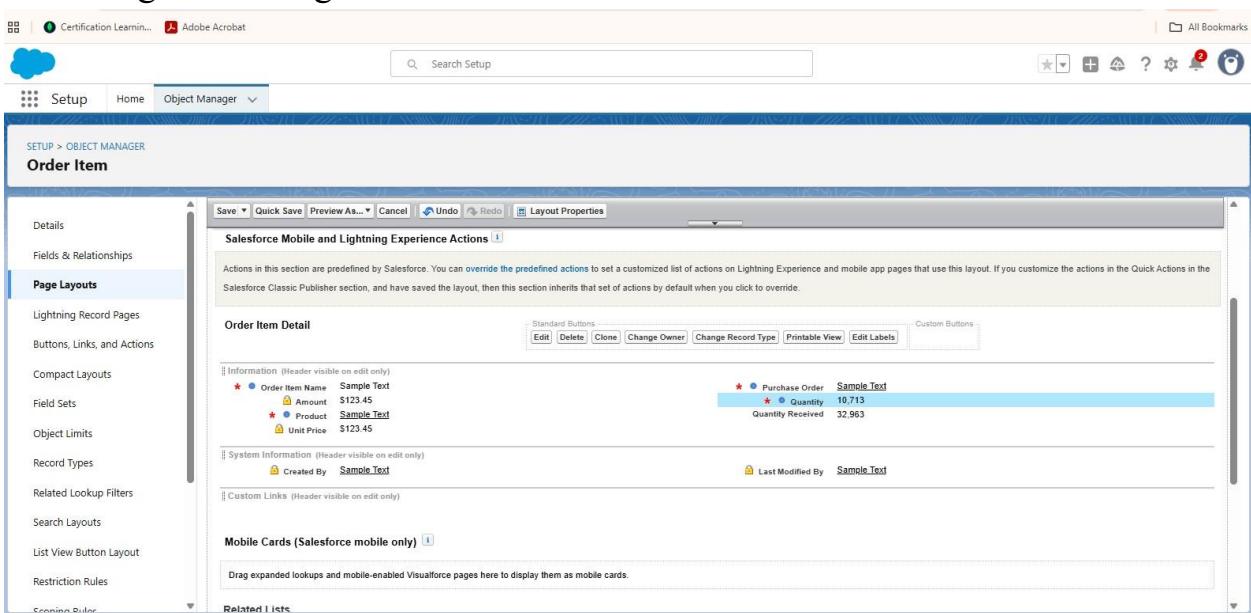


- Click on field Order Date >> click on settings >> select Required and save it.

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

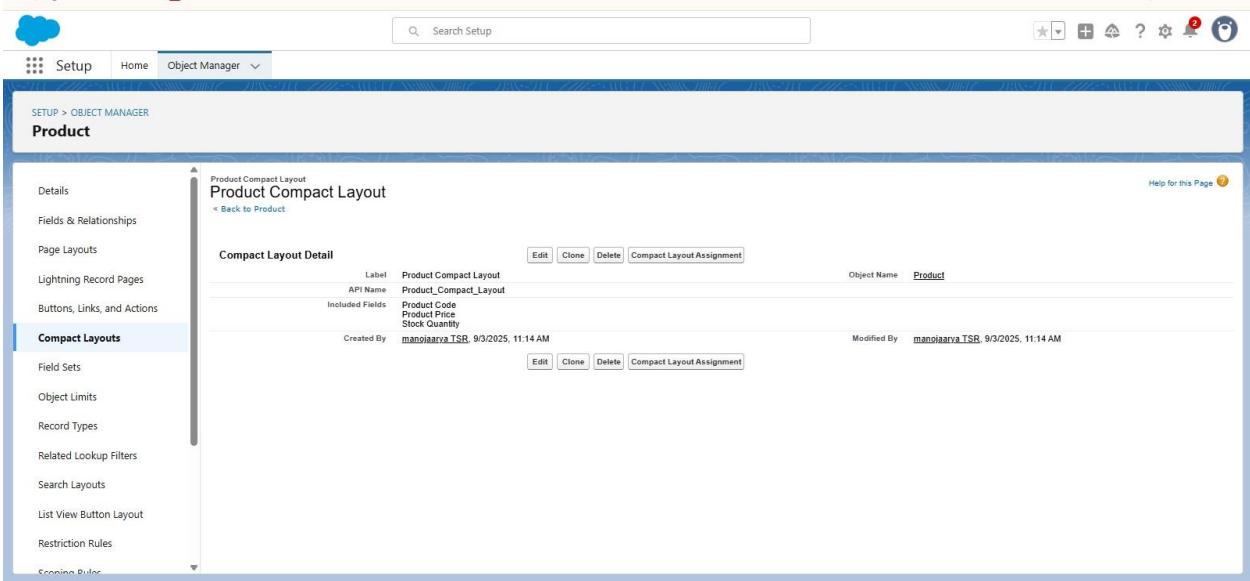
1. Go to setup >> click on Object Manager >> type object name(Supplier) in quick find box >> click on the Supplier object >> Page Layouts.
2. Click on the Supplier Layout
3. Drag and Arrange the field as shown below

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 set up >> 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 side bar.
3. Click on New.
4. Enter the Label as “Purchase Order Compact Layout”.
5. Select 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. The left sidebar is expanded to show various configuration options under the 'Compact Layouts' tab. The main area displays the 'Purchase Order Compact Layout' record, which has been created. The layout details include the label 'Purchase Order Compact Layout', API name 'Purchase\_Order\_Compact\_Layout', and included fields: Purchase Order ID, Order Date, Total Order Cost, and Supplier. The layout was created by 'manojaarya\_TSR' on 9/6/2025 at 2:13 AM and modified by the same user on the same date and time. Action buttons for Edit, Clone, Delete, and Compact Layout Assignment are visible at the top and bottom of the layout detail page.

## 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 :

$(\text{Expected\_Delivery\_Date\_c} - \text{Order\_Date\_c}) > 7$

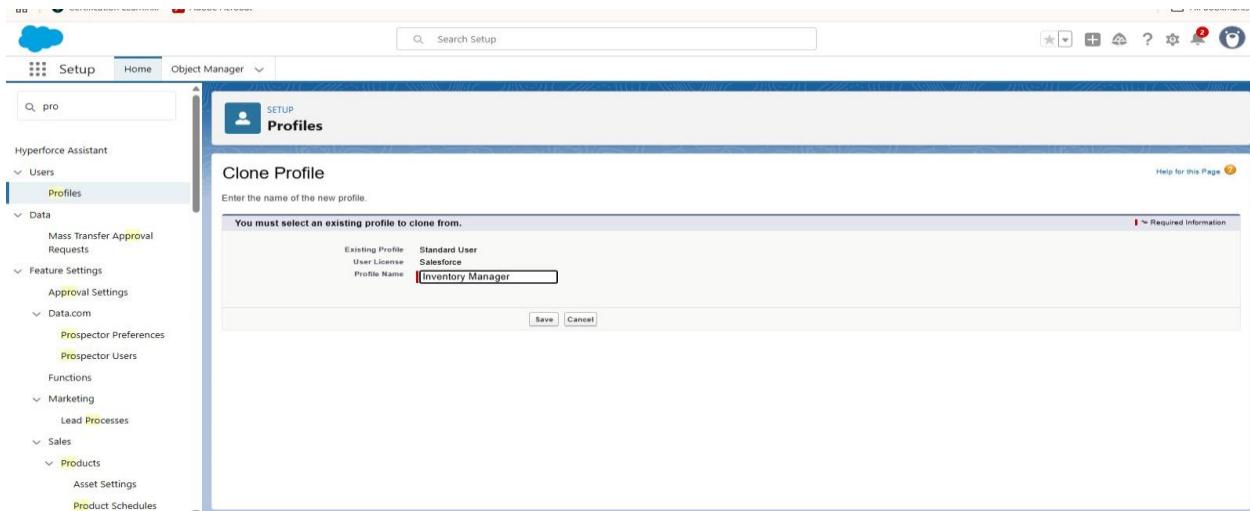
The screenshot shows the Salesforce Setup interface for the Purchase Order object. On the left, a sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area is titled "Purchase Order Validation Rule" and displays the "Validation Rule Detail" for a rule named "Expected\_Delivery\_Date\_Validation". The rule's formula is  $(\text{Expected\_Delivery\_Date\_c} - \text{Order\_Date\_c}) > 7$ . 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 "manojaarya.TSR" with the timestamp "9/6/2025, 1:34 AM". There are "Edit" and "Clone" buttons at the bottom of the rule detail section.

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

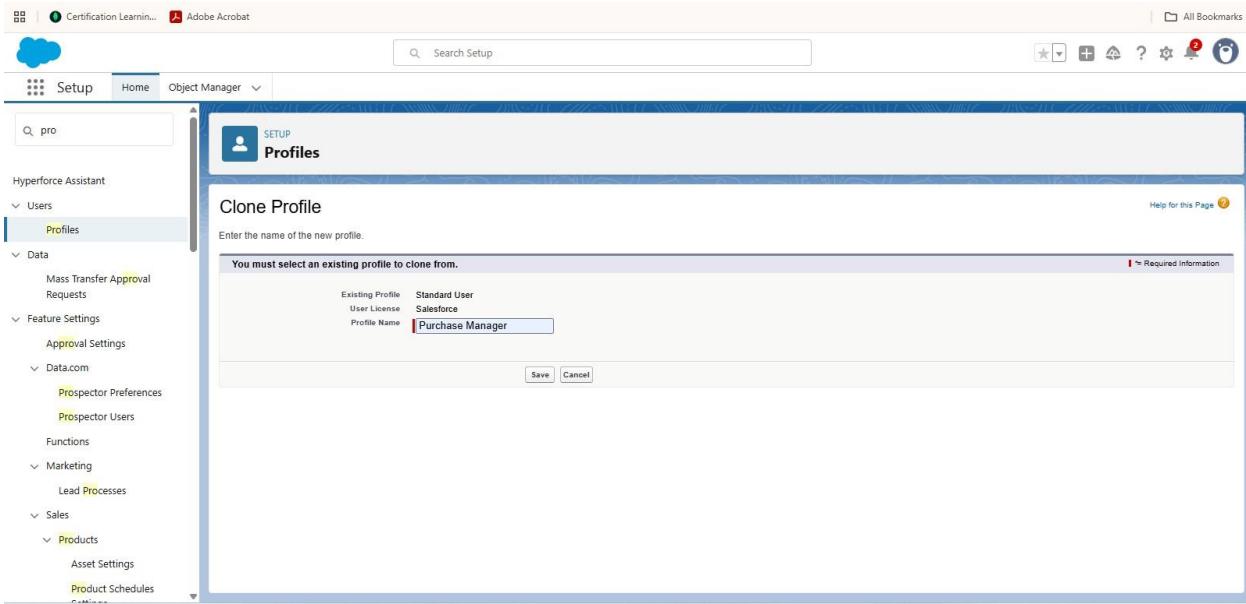
1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Inventory Manager) >> Save.



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 Roles page. On the left, there's a sidebar with navigation links like 'Users', 'Roles' (which is selected), '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 search bar at the top says 'Search Setup'. The main content area has a title 'Understanding Roles' and a sub-section 'Sample Role Hierarchy' with a dropdown set to 'Territory-based Sample'. It shows a hierarchy from 'Executive Staff' down to 'Western Sales Rep' and 'Eastern Sales Rep'. Each role has a brief description of its permissions. At the bottom right of the main content area is a 'Set Up Roles' button.

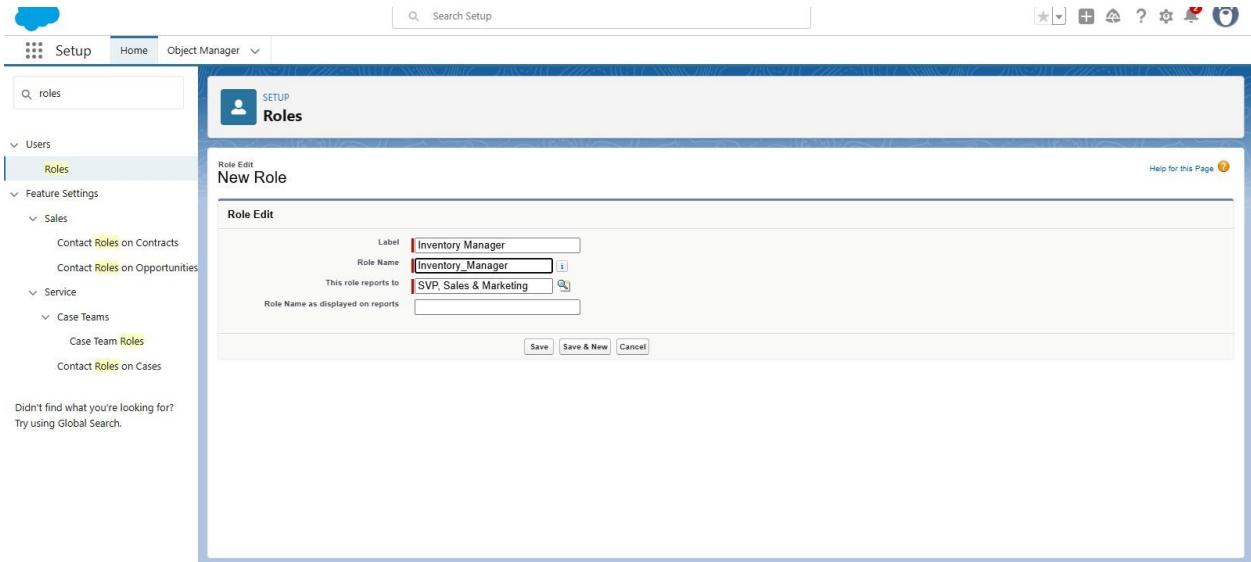
2. Click on Expand All and click on add role under SVP, Sales & Marketing role.
3. Give Label as “Purchasing Manager” and Role name gets auto populated. Then click on Save.

This screenshot shows the 'Role Edit New Role' form. The 'Label' field is set to 'Purchasing Manager', the 'Role Name' field is also 'Purchasing\_Manager', and the 'This role reports to' field is 'SVP, Sales & Marketing'. There's a note below stating 'Role Name as displayed on reports' followed by a text input field which is empty. At the bottom of the form 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 top screenshot shows the 'Permission Sets' page in the Salesforce Setup. The sidebar includes 'Users', 'Permission Set Groups', and 'Custom Code'. The main area displays a list of permission sets with columns for Action, Permission Set Name, Description, and License. The bottom screenshot shows the 'Create' dialog for a new permission set, with fields for Label ('Purchase Manager Create Access'), API Name ('Purchase\_Manager\_Create\_Access'), and Description.

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.

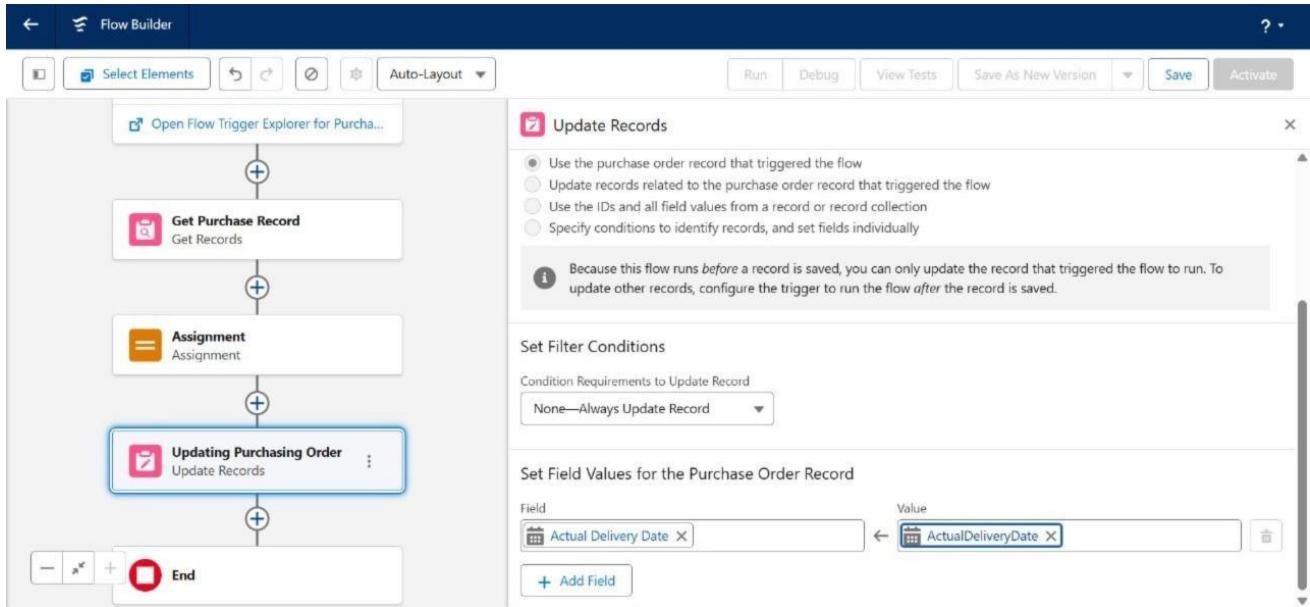
## Milestone 13 - Flows

### 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 'Flow Definitions' under the heading 'All Flows'. The list includes various flows such as 'Add or Modify Service Appointment Attendees', 'Approvals Workflow: Evaluate Approval Requests', and 'Approvals Workflow: Process Approval Submission'. Each row in the table provides details like 'Process Type', 'Active status', and 'Package State'.

The screenshot shows the 'Flow Builder' interface. On the left, there's a visual representation of the flow structure with nodes for 'Record-Triggered Flow Start' (Purchase Order), 'End', and a central node. The right side contains configuration panels. The 'Configure Start' panel shows the trigger is 'A record is created or updated' for 'Purchase Order'. The 'Optimize Flow' panel has sections for 'Fast Field Updates' (selected) and 'Actions and Related Records'. Buttons at the top include 'Run', 'Debug', 'View Tests', 'Save As New Version', 'Save', and 'Activate'.



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.

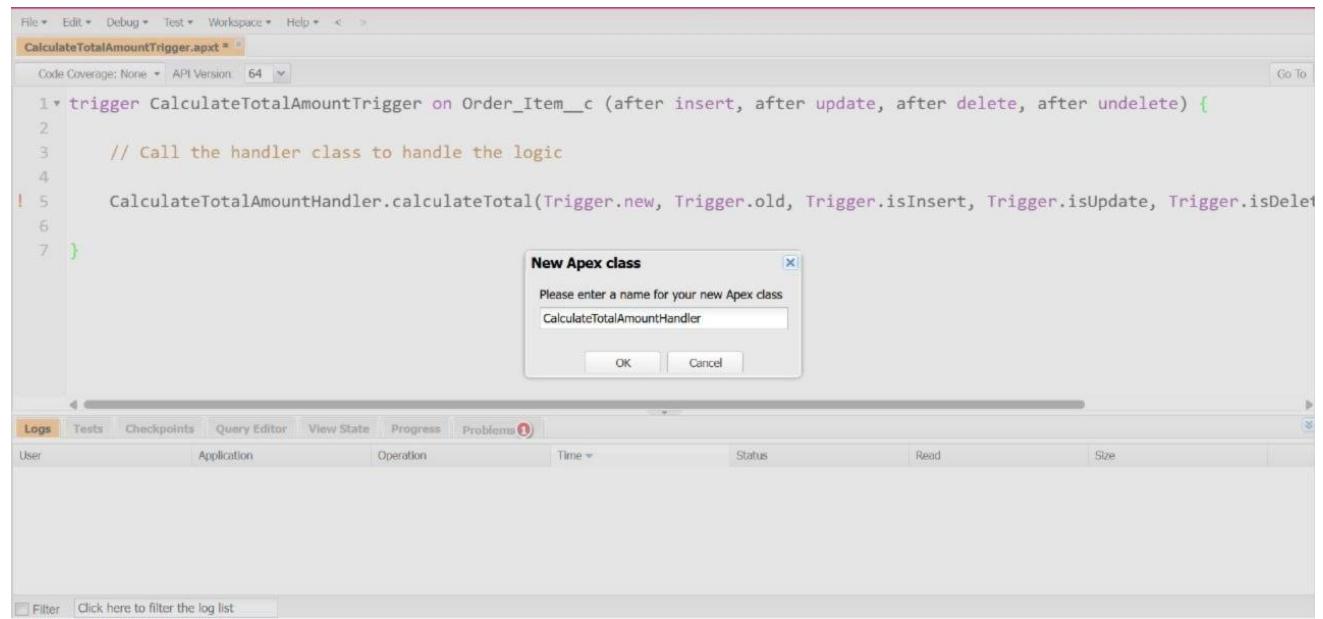
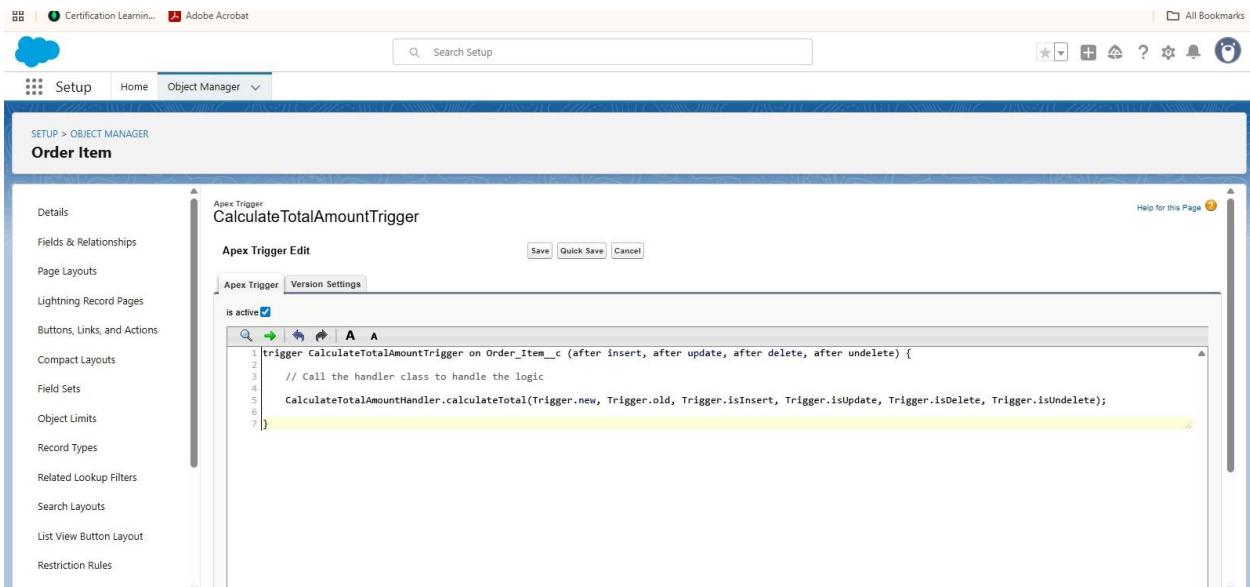
Step 2:

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.

The screenshot shows the 'Create Report' dialog box in the Medical Inventory software. The dialog box has a title 'Create Report' at the top. On the left, there is a sidebar with a tree view of categories: 'Recently Used' (Accounts & Contacts, Opportunities, Customer Support Reports, Leads, Campaigns, Activities, Contracts and Orders, Price Books, Products and Assets, Administrative Reports, File and Content Reports) and 'All' (Accounts & Contacts, Opportunities, Customer Support Reports, Leads, Campaigns, Activities, Contracts and Orders, Price Books, Products and Assets, Administrative Reports, File and Content Reports). The main area is titled 'Selected a Report Type' and contains a search bar with 'Search Report Types...' placeholder text. Below the search bar is a table with columns 'Report Type Name' and 'Category'. The table lists various report types: Accounts (Standard), Contacts & Accounts (Standard), Accounts with Partners (Standard), Account with Account Teams (Standard), Accounts with Contact Roles (Standard), Accounts with Assets (Standard), Contacts with Assets (Standard), Account History (Standard), and Contact History (Standard). At the bottom of the dialog box, there is a 'Recent Items' section with a history icon and a 'Recent Items' link.

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

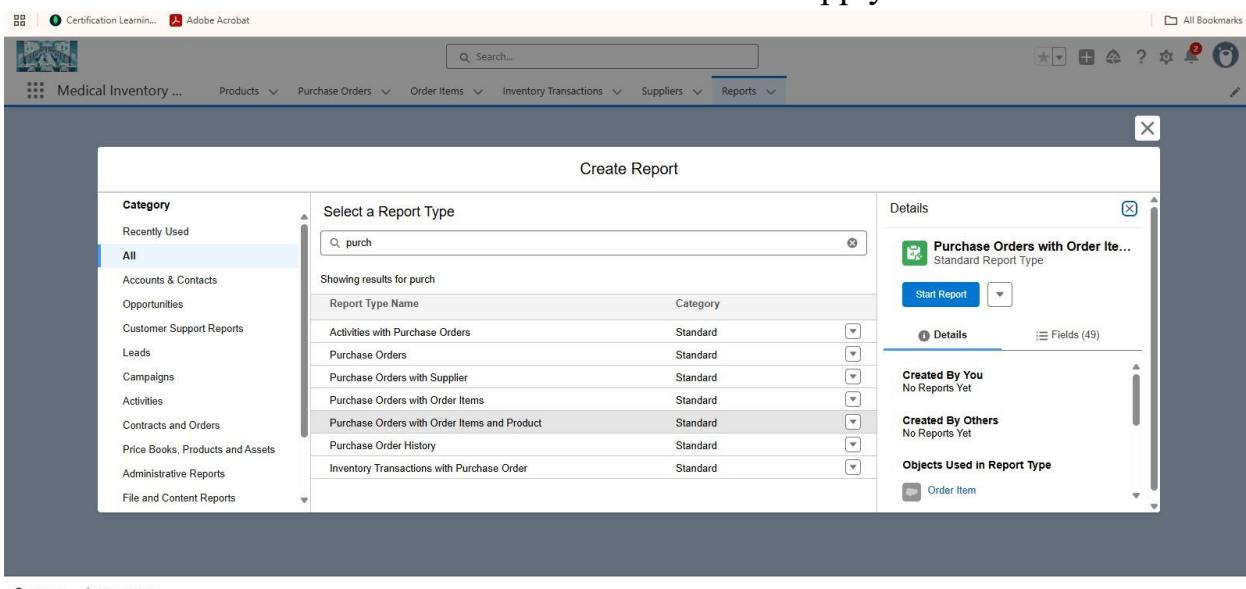
What you selects in “Select a report type option”)

(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



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.

Sales Home Opportunities Leads Tasks Files Accounts Contacts Campaigns Dashboards Reports Chatter Groups Calendar People More

Medical Inventory Dashboard

Select Report

**Reports**

**Recent**

- Created by Me
- Private Reports
- Public Reports
- All Reports

**Folders**

- Created by Me
- Shared with Me
- All Folders

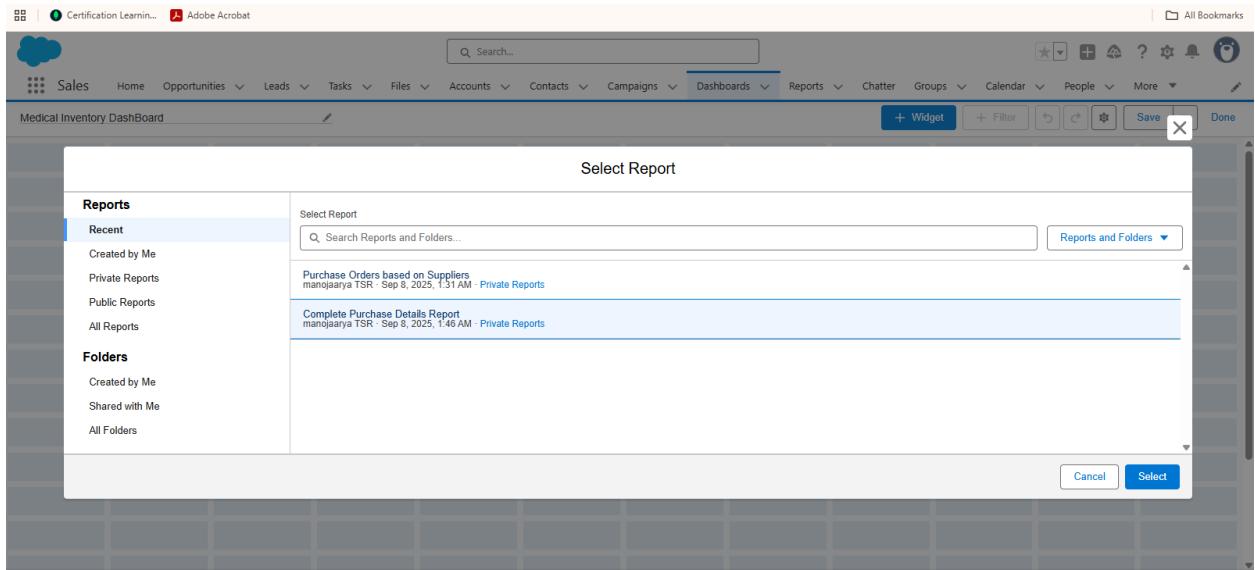
Select Report

Search Reports and Folders...

Purchase Orders based on Suppliers  
manojganya TSR - Sep 8, 2025, 1:31 AM · Private Reports

Complete Purchase Details Report  
manojganya TSR - Sep 8, 2025, 1:49 AM · Private Reports

Cancel Select



## Add Widget

Report

Purchase Orders based on Supplie

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

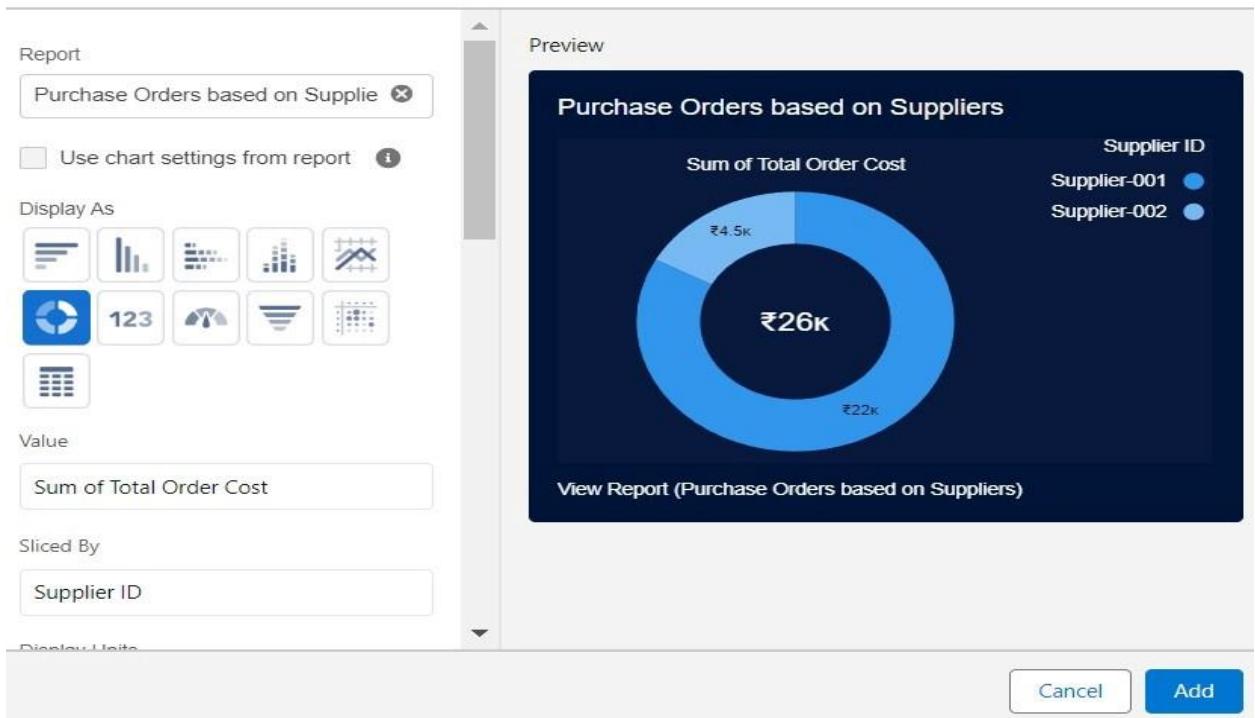
₹26k

₹4.5k

₹22k

View Report (Purchase Orders based on Suppliers)

Cancel Add

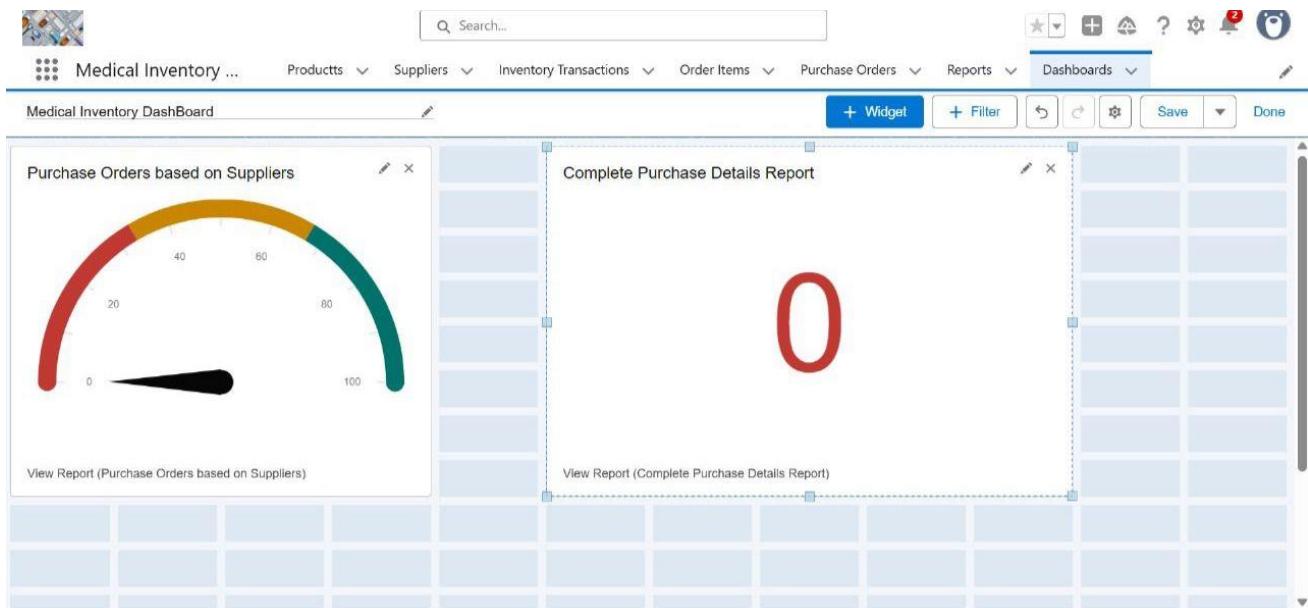


## Activity 2: View Dashboard

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 screenshot shows the Salesforce Setup Home page. On the left, the App Launcher sidebar is open, displaying the "Medical Inventory Management" app under the "Apps" section. The main content area features a "Get Started with Einstein Bots" card and a "Mobile Publisher" card. Below these cards, there is a section titled "Most Recently Used" which lists 10 items, including Apex classes, triggers, and a user record. The top navigation bar includes links for "Home", "Object Manager", and a search bar labeled "Search Setup".

The screenshot shows the Salesforce Dashboards creation interface. A modal dialog box titled "New Dashboard" is open in the center. It contains fields for "Name" (set to "Medical Inventory DashBoard"), "Description" (empty), and "Folder" (set to "Private Dashboards"). There are "Select Folder" and "Cancel" buttons at the bottom right of the dialog. The background shows a grid of dashboard slots and a toolbar with various dashboard-related icons.



## Conclusion

The Medical Inventory Management System effectively automates and optimizes inventory operations within healthcare settings. Utilizing Salesforce CRM capabilities, the system enhances efficiency, accuracy, and transparency in managing medical supplies and equipment.

### Key takeaways include:

- Reduced Manual Effort: Automation of purchase orders, stock tracking, and supplier management significantly decreases manual tasks.
- Improved Decision-Making: Real-time dashboards and analytical reports provide actionable insights for administrators and inventory managers.
- Error Minimization: Automated workflows and validation rules help prevent mistakes in stock management and procurement.
- Enhanced Accountability: Supplier and order histories ensure transparency and better tracking of resources.
- Practical Learning Experience: The project provided hands-on experience with Salesforce customization, automation, and role-based security, applying theoretical knowledge to a real-world problem.

- Alignment with Naan Mudhalvan Goals: Demonstrates the practical application of technology to improve operational efficiency in healthcare institutions.

Overall, this project not only delivers a robust solution for managing medical inventory but also equips students with essential skills in cloud-based application development, problem-solving, and project management.