**Title: Medical Inventory Management Using Salesforce**

Using Salesforce…

Project Overview: Medical Inventory Management ensures efficient tracking of medicines, suppliers, purchase orders, and sales orders. Using Salesforce CRM, this project automates the from procurement to entire process distribution – helping hospitals and pharmacies reduce wastage, prevent stock outs, and improve patient care.

**Objectives**:

* Ensure Medicine Availability: Prevent stock outs by providing real-time updates on available medicines and supplies.
* Track Expiry and Wastage: Maintain accurate records of expiry dates to minimize wastage of unused medicines.
* Improve Supplier Coordination: Automate purchase order creation, supplier communication, and delivery tracking.
* Enhance Transparency: Keep detailed logs of stock movement, purchases, and distribution for accountability.
* Optimize Resource Allocation: Monitor inventory levels, forecast demand, and automate replenishment to reduce overstocking.
* Improve Decision Making: Generate analytical reports and dashboards for administrators to plan and allocate resources effectively.
* Promote Patient Care: Build a reliable and efficient inventory system that ensures timely delivery of essential medicines to patients.

**Student Outcomes**:

* Practical Salesforce Experience: Students gain real-world skills in configuring objects like Suppliers, Purchase Orders, Order Items, and Products.
* Understanding of Healthcare Processes: Learn how inventory management applies to the medical field, bridging technology with healthcare needs.
* Workflow and Automation Skills: Hands-on practice with Flows, Approval Processes, Validation Rules, and Triggers.
* Reporting and Analytics: Ability to design reports and dashboards for monitoring stock, supplier performance, and order details.
* Team Collaboration: Students practice working as a team — handling requirement gathering, development, testing, and deployment.
* Problem-Solving Skills: Learn to identify challenges like shortage, expiry, duplicate orders and design Salesforce-based solutions.
* Industry-Relevant Exposure: Prepares students for careers in healthcare IT and CRM-based project implementation.

**System Requirements**:

Hardware Requirements:

* Computer with minimum 4 GB RAM, Dual-Core Processor
* Stable Internet Connection

Software Requirements:

* Salesforce Developer Edition Org
* Modern Web Browser (Google Chrome, Firefox, Edge)

Project Duration : 31 Hours;

Phases Overview :

Phase

No.

Phase Name Description Page

Numbers

1 Requirement

Analysis & Planning

Gathering requirements from

Donors, volunteers, and receivers;

Defining scope and goals; planning

Data model and workflows.

2 Salesforce

Development –

Backend &

Configurations

Creating custom objects, fields,

Relationships; setting up Flows

And Apex Triggers for

Automation.

4 – 11

3 UI/UX Development

& Customization

Building Lightning App,

Customizing layouts, adding fields,

Implementing Flows, and

Developing UI logic.

11 – 28

4 Data Migration,

Testing & Security

Creating Users, Profiles, Public

Groups, Sharing Rules;

Configuring Report Types, Reports, Dashboards; testing functionalities and ensuring data

Security.

28 – 37

5 Deployment,

Documentation &

Maintenance

Designing and finalizing Home

Page, deploying solution to live

Environment, preparing

Documentation, conclusion, and

Ongoing system maintenance.

37 – 40

Phase 1: Requirement Analysis & Planning:-

Medical Inventory Management:

Utilizing Salesforce, our project focuses on managing medicines, suppliers, purchase orders, and sales orders within hospitals, clinics, and pharmacies. The system streamlines stock tracking, supplier coordination, and distribution of medicines, ensuring efficiency, transparency, and minimal wastage.

Phase 2: Salesforce Development – Backend & Configurations:-

Milestone 1-Salesforce Account

Activity 1: Creating Developer Account

Creating a developer org in salesforce.

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

2.On the sign up form, enter the following details :

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

This need not be an actual email id, you can give anything in the format : [username@organization.com](mailto:username@organization.com)

Click on sign me up after filling these.

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 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

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 .

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.

Milestone 5- Fields

Activity 1: Creating A Text Field In Product Object

1.To create fields in an object:

2.Click the gear icon and select Setup. This launches Setup in a new tab.

3.Click the Object Manager tab next to Home.

4.Select Product custom object.

5.Select Fields & Relationships from the left navigation

6.Click on New

7.Select Text field, click Next

8.Enter Field Label as “Product Name” and Length 255.

9.Select Required Field.

10.Click Next, Next, then Save & New.

Activity 2: Creating A Text Area 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 Area field, click Next

7.Enter Field Label as “Product Description” .

8.Click Next, Next, then Save & New.

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.

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.

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 Pick list 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 “Pick list” 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.

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.

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.

Activity 2: To Edit A Page Layout In 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 >> Page Layouts.

2.Click on the Purchase Order Layout

3.Drag and Arrange the field as shown below.

4.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

4.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.

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

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.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.

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

Activity 1 : 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 “Purchasing Manager” and Role name gets auto populated. Then click on Save.

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

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.

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 Purchase M to assign the permission set to and click Next.

5.Select No Expiration date >> Click on Assign.

Milestone 13

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 .

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 follow:

* Field: Id
* Operator: Equals
* Value: {!$Record. Id}

11.For How many Records to store Select Only the First Record.

12.For How to Store Record Data select Choose fields and let Salesforce do the rest. Select Field: Order \_Date \_\_c. Click on Done.

13.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.

14. Enter API name as ActualDeliveryDate >> Select Data type as Date >> Click on Done.

15.From the Toolbox drag and drop Assignment element.

16.Enter the label as “Assignment”.

17.Set Variable Values:

1. Variable : {!ActualDeliveryDate}

Operator : Equals

Value : {!$Record. Order \_Date \_\_c}

1. Variable : {!ActualDeliveryDate}

Operator : Add

Value : 3

18.Click Done

19.From the Toolbox drag and drop Update Records element and connect to the Assignment element.

20.Enter the label as “Updating Purchasing Order”.

21.How to Find Records to Update and Set Their Values : Use the Purchase Order record that triggered the flow

22.Set Filter Conditions : None -Always Update Record

23.Set Field Values for the Trip Record as

Field : Actual \_Delivery \_Date \_\_c

Value : {!ActualDeliveryDate}

24.Click Done

25.Save the flow as “Actual Delivery Date Updating”.

26.Activate the flow.

Milestone 14

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. Calculate Total(Trigger. new, Trigger. old, Trigger. Is Insert, Trigger. is Update, Trigger. is Delete, Trigger. is Undelete);

}

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 calculate Total(List<Order \_Item \_\_c> new Items, List<Order \_Item \_\_c> old Items, Boolean is Insert, Boolean is Update, Boolean is Delete, Boolean is Undelete) {

// Collect Purchase Order IDs affected by changes in Order \_Item \_\_c records

Set<Id> parent Ids = new Set<Id>();

// For insert, update, and undelete scenarios

If (is Insert || is Update || is Undelete)

For (Order \_ Item \_\_c order Item : new Items) {

Parent Ids. add(order Item. Purchase \_Order \_Id \_\_c);

}

}

// For update and delete scenarios

If (is Update || is Delete) {

For (Order \_Item \_\_c ordItem : old Items) {

Parent Ids. Add (ordItem. Purchase \_Order \_Id \_\_c);

}

}

// Calculate the total amounts for affected Purchase Orders

Map<Id, Decimal> purchase To Update Map = new Map<Id, Decimal>();

If (!parent Ids. Is Empty ()) {

// Perform an aggregate query to sum the Amount \_\_c for each Purchase Order

List<Aggregate Result> aggrList = [

SELECT Purchase \_Order \_Id \_\_c, SUM(Amount \_\_c) total Amount

FROM Order \_Item \_\_c

WHERE Purchase \_Order \_Id \_c IN : parent Ids

GROUP BY Purchase \_Order \_Id \_c

];

// Map the result to Purchase Order IDs

For (Aggregate Result aggr : aggrList) {

Id purchase Order Id = (Id) aggr . get(‘Purchase \_Order \_Id \_\_c’);

Decimal total Amount = (Decimal) aggr. get(‘total Amount’);

Purchase To Update Map. put(purchase Order Id, total Amount);

}

// Prepare Purchase Order records for update

List<Purchase \_Order \_\_c> purchase To Update = new List<Purchase \_Order \_\_c>();

For (Id purchase Order Id : purchase To Update Map. key Set()) {

Purchase \_Order \_\_c purchase Order = new Purchase \_Order \_\_c(Id = purchase Order Id, Total \_Order \_cost \_\_c = purchase To Update Map. get (purchase Order Id));

Purchase To Update. add(purchase Order);

}

// Update Purchase Orders if there are any changes

If (!purchase To Update .is Empty()) {

Update purchase To Update;

}

}

}

}

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

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.

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 Dash Board see graph view of records

**CONCLUSION:**

By leveraging the Salesforce platform, the project successfully established a transparent and efficient system for managing medical inventory in hospitals and healthcare centers. Through real-time tracking of medicines, medical equipment, and supplies, the project ensured timely availability of resources, reduced wastage, and improved patient care outcomes.

The project “Medical Inventory Management using Salesforce” has been successfully implemented and demonstrates the practical use of Salesforce CRM for healthcare optimization.

Project Achievements:

* Streamlined the process of tracking and managing medicines, consumables, and equipment.
* Ensured real-time updates of stock availability, expiries, and re-order levels.
* Automated workflows for purchase orders, supplier coordination, and stock replenishment.
* Improved transparency using dashboards, reports, and alerts.
* Enhanced usability with Lightning App, Home Page, and role-based security for doctors, pharmacists, and administrators.

Student Learning Outcomes:

* Hands-on skills in Salesforce CRM customization for healthcare operations.
* Improved problem-solving through real-time inventory management use cases.
* Experience in developing automated workflows for critical healthcare processes.
* Collaboration and teamwork in handling requirement analysis, development, and testing.
* Exposure to industry-relevant tools and healthcare project lifecycle management.

Future Scope:

* Integration with To devices to track medical equipment usage and maintenance.
* Use of AI and predictive analytics to forecast medicine demand and prevent shortages.
* Collaboration with suppliers, pharmacies, and distributors for seamless inventory flow.
* Expansion to a multi-hospital and multi-region system for large-scale healthcare networks.
* Integration with mobile apps for doctors, pharmacists, and patients to improve accessibility.