

MEDICAL INVENTORY MANAGEMENT SYSTEM

College Name: Government Arts college in Udhagamandalam

College Code: Bru11

TEAM MEMBERS:

TEAM ID : NM2025TMID27278

TEAM SIZE : 4

TEAM LEADER : NALLAPERUMAL P

TEAM MEMBER : ELANGO S

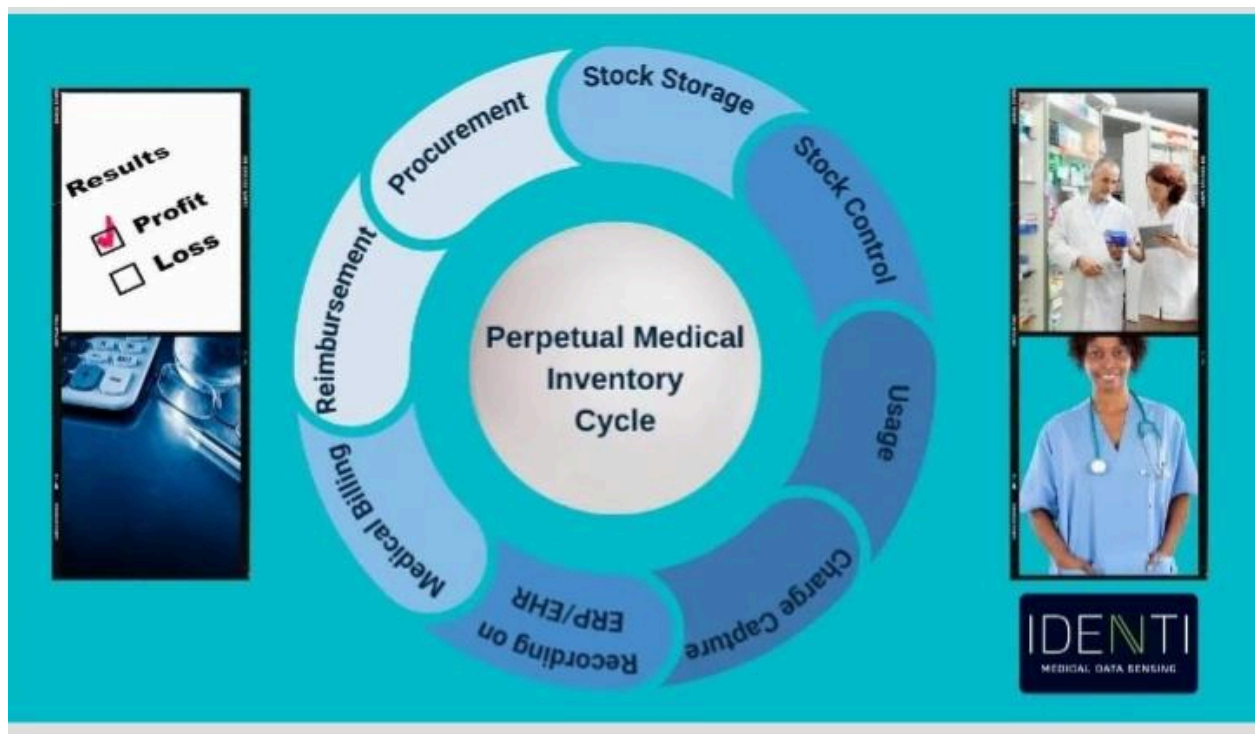
TEAM MEMBER : DINESH B

TEAM MEMBER : EBENEZER G

1. INTRODUCTION

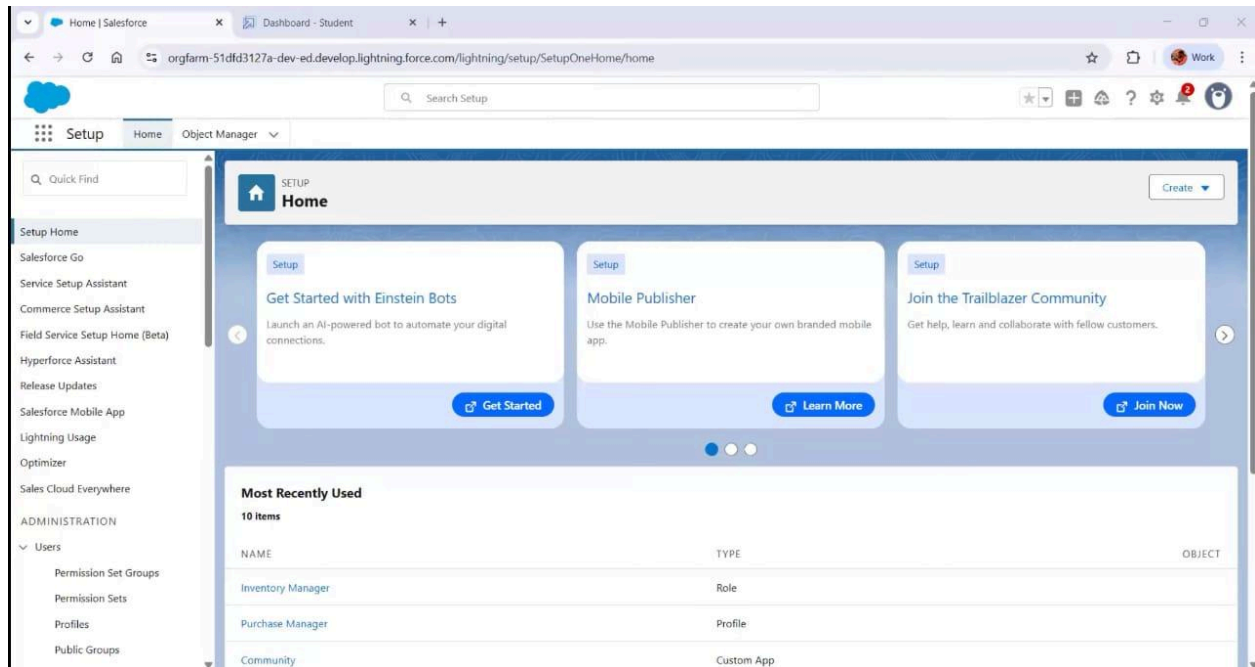
Project Overview

The Medical Inventory Management System is a Salesforce-based application designed to streamline the management of medical supplies, medicines, and equipment in hospitals, clinics, and pharmacies. It ensures accurate tracking of stock levels, supplier details, purchase orders, and usage records with automation features such as flows, approval processes, and alerts.



1. Purpose

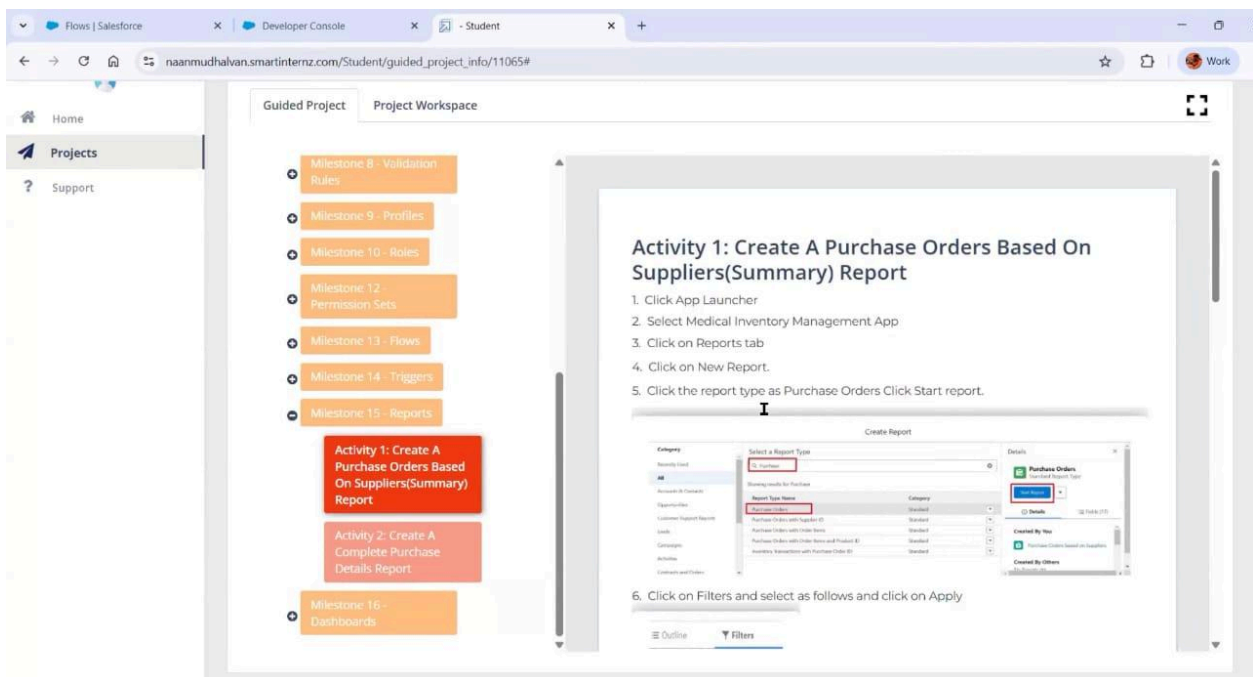
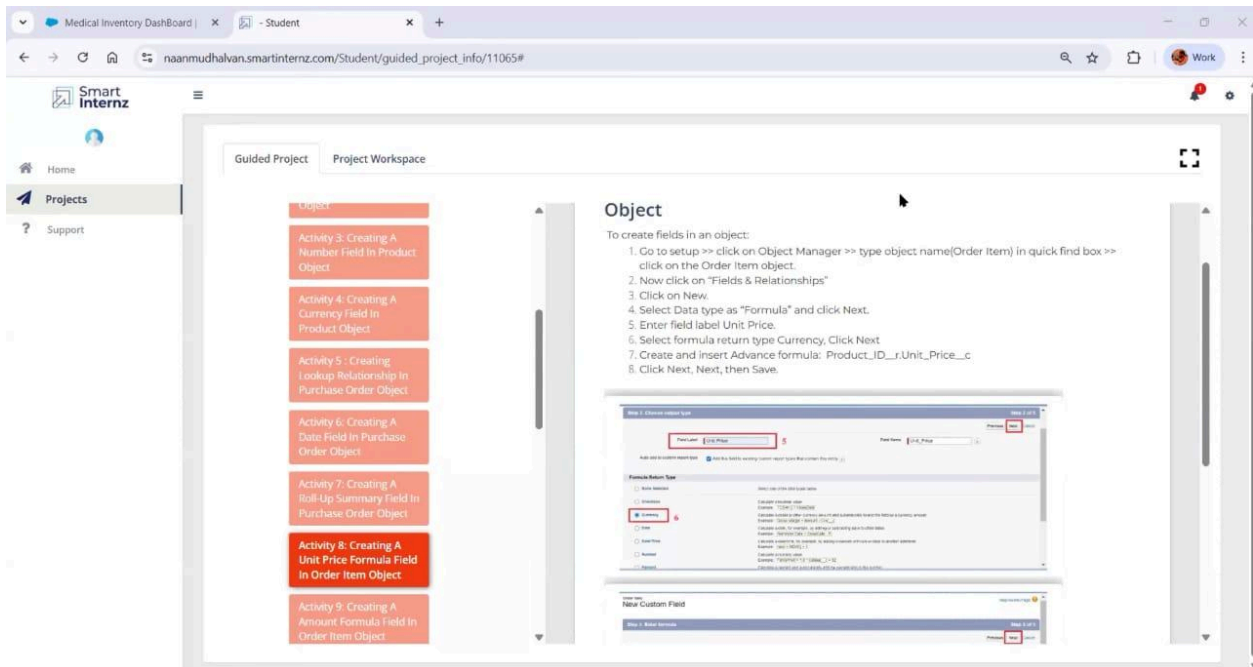
The main objective of this project is to enable healthcare organizations to efficiently manage inventory, avoid shortages/overstocking, improve accuracy, and ensure timely replenishment of medical supplies. It reduces manual intervention and enhances patient care by maintaining reliable stock availability.



2. DEVELOPMENT PHASE

Creating Developer Account:

Used Salesforce Developer Org via <https://developer.salesforce.com/signup>



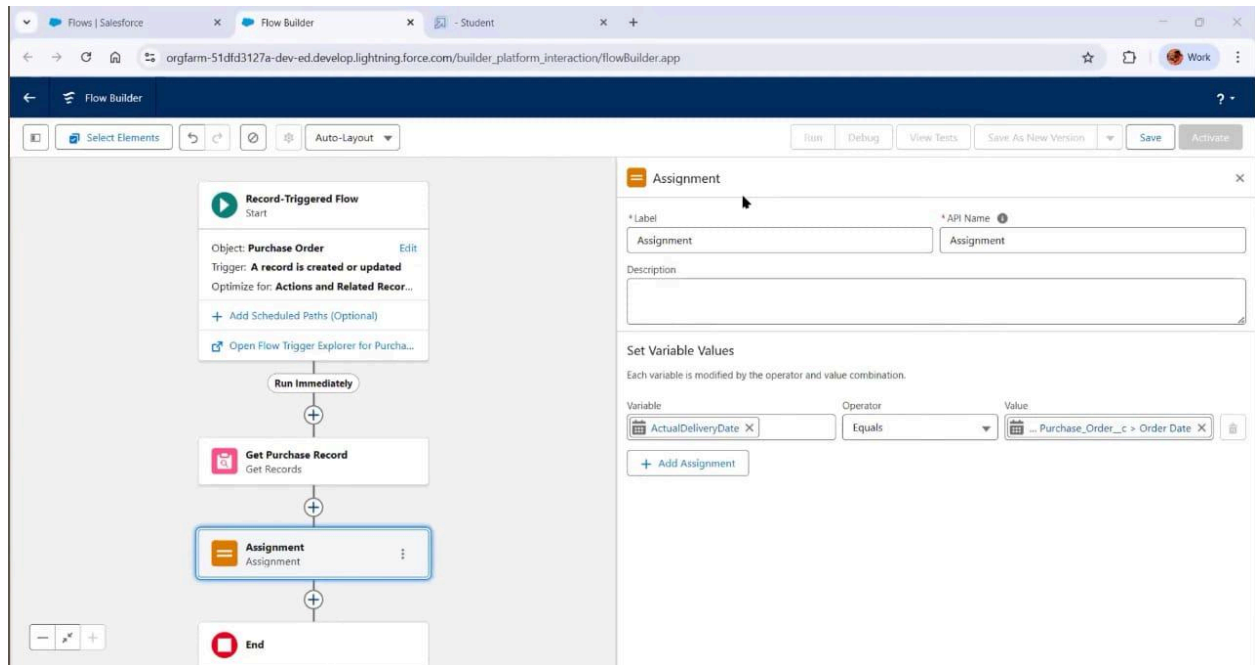
Created Objects:

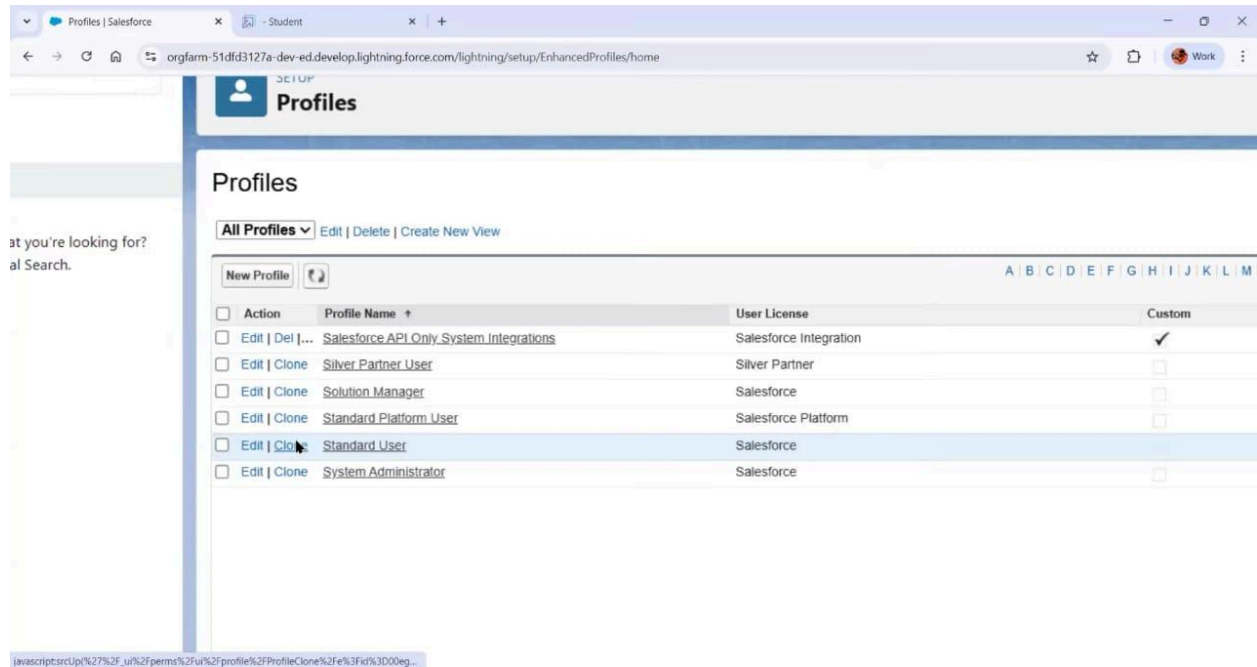
Medicine (Name, Category, Expiry Date, Stock Quantity, Price)

Supplier (Name, Contact, Email, Address)

Purchase Order (Supplier, Medicine, Quantity Ordered, Date, Status)

Stock Transaction (Medicine, Quantity Used, Department, Date)

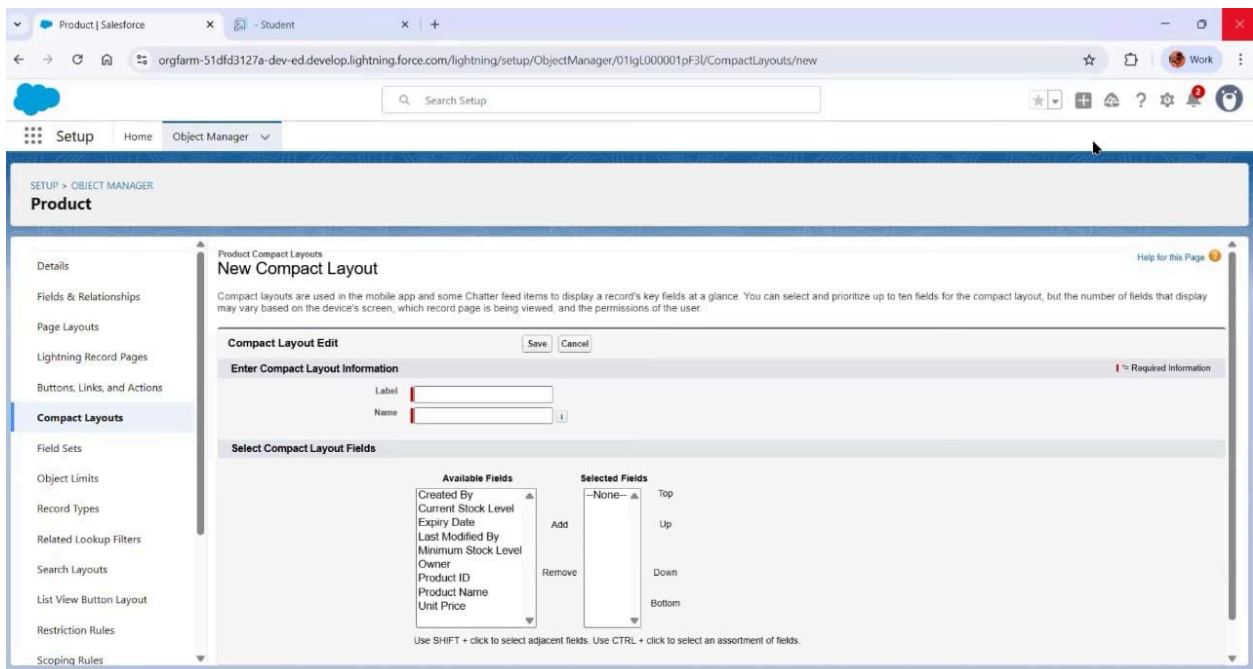
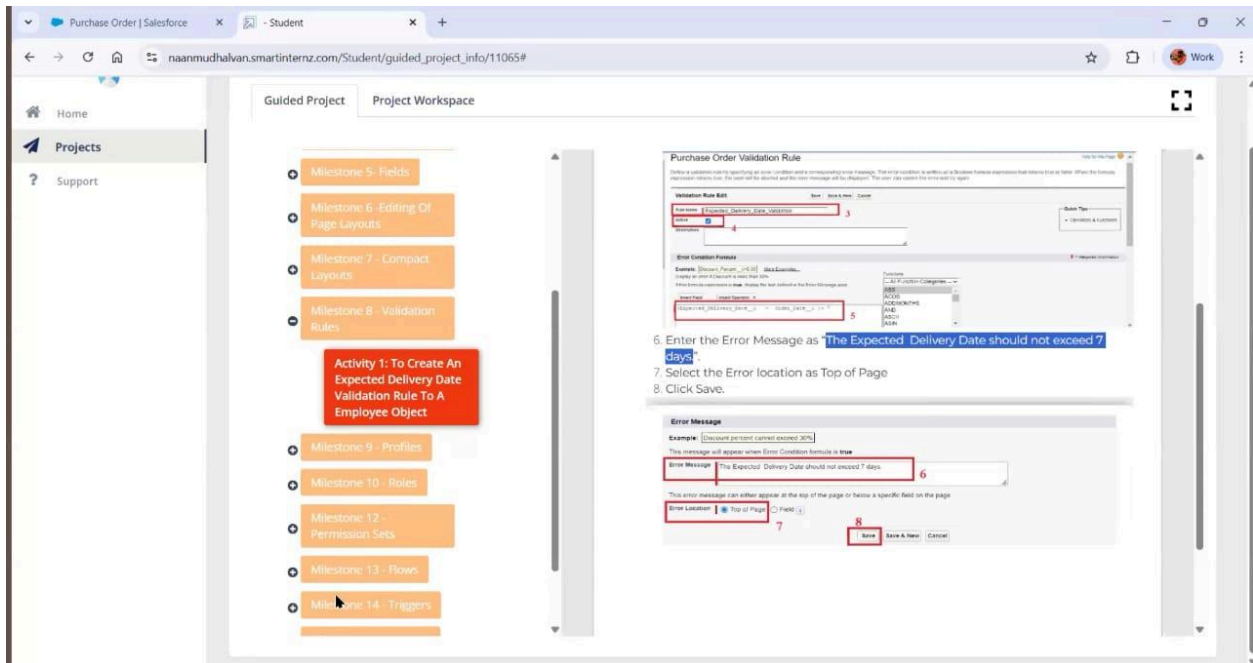




Configured Fields and Relationships:

Lookup relationships between Supplier Medicine, Medicine Purchase Order

Validation rules to prevent expired medicine usage



Developed Lightning App with Tabs:

Medicine, Supplier, Purchase Order, Stock Transaction

Implemented Flows:

Automatic stock deduction when a transaction is recorded

Reorder notification when stock falls below threshold

Validation Rules:

Expired medicines cannot be issued

Minimum order quantity must be greater than zero

The screenshot displays a Salesforce Guided Project workspace. On the left, a sidebar shows navigation options: Home, Projects, and Support. The main area is titled 'Guided Project' and 'Project Workspace'. A vertical list of activities is shown, starting with 'Milestone 5: Fields' and followed by seven activities related to creating fields in the Product Object. The first activity, 'Activity 1: Creating A Text Field In Product Object', is selected and expanded. The expanded view shows a list of steps to create fields in an object, followed by a screenshot of the Salesforce Object Manager interface. The Object Manager interface shows a table of objects with columns for Name, Type, and Description. The 'Product' object is highlighted, and the 'Fields' tab is selected, showing a list of fields with columns for Name, Type, and Description.

Guided Project Project Workspace

Milestone 5: Fields

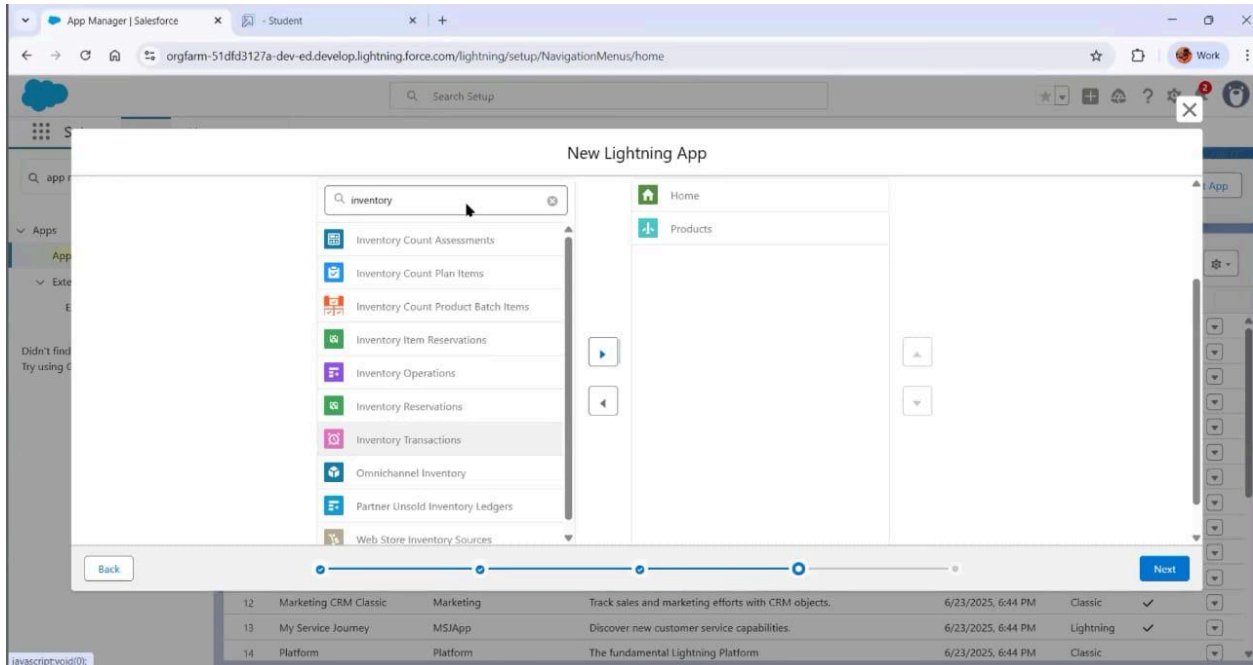
- Activity 1: Creating A Text Field In Product Object
- Activity 2: Creating A TextArea Field In Product Object
- Activity 3: Creating A Number Field In Product Object
- Activity 4: Creating A Currency Field In Product Object
- Activity 5: Creating Lookup Relationship In Purchase Order Object
- Activity 6: Creating A Date Field In Purchase Order Object
- Activity 7: Creating A Roll-Up Summary Field In

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.

NAME	TYPE	DESCRIPTION	LAST MODIFIED	DELETED
Product Name	Text (255)	Product Name		
Product Price	Text (255)	Product Price		
Product Description	Text (255)	Product Description		
Product Status	Text (255)	Product Status		
Product Category	Text (255)	Product Category		
Product Manufacturer	Text (255)	Product Manufacturer		
Product Weight	Text (255)	Product Weight		
Product Height	Text (255)	Product Height		
Product Width	Text (255)	Product Width		
Product Volume	Text (255)	Product Volume		
Product Weight (kg)	Text (255)	Product Weight (kg)		
Product Height (cm)	Text (255)	Product Height (cm)		
Product Width (cm)	Text (255)	Product Width (cm)		
Product Volume (cm³)	Text (255)	Product Volume (cm³)		



Apex Triggers:

Prevent negative stock updates

Auto-update "Stock Status" (In Stock, Low Stock, Out of Stock)

Scheduled Apex Class:

Monthly email reminders for upcoming expiries and low stock

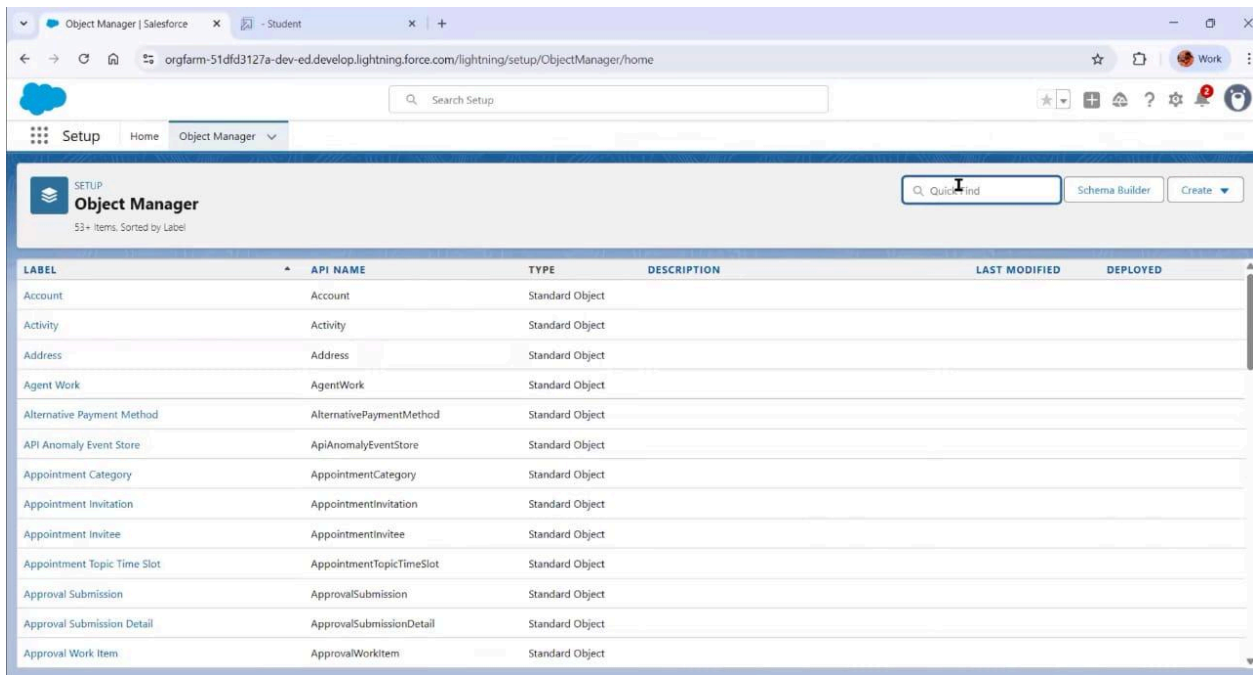
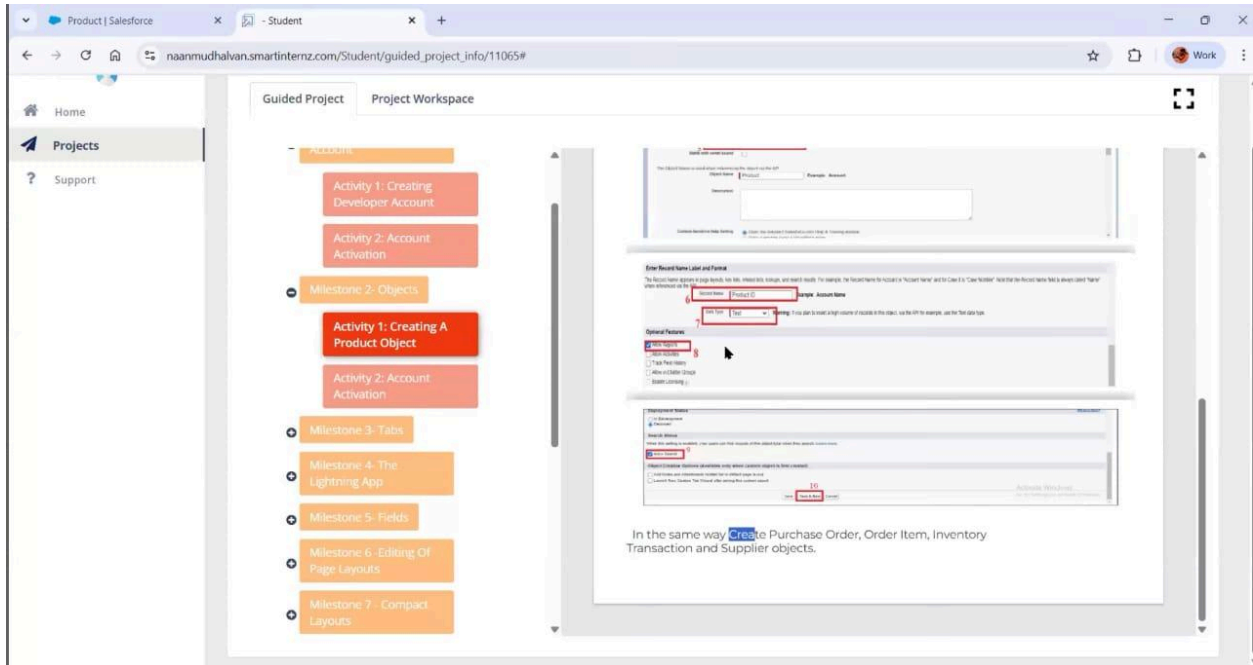
Email Templates Built: Low stock alert

Expiry reminder

Purchase order approval/rejection

Approval Process:

For approving new purchase orders before confirming with suppliers



3. FUNCTIONAL AND PERFORMANCE TESTING

Validated stock level updates after issuing medicine

Trigger validation for expired medicines

Tested low-stock flow automation

Approval process validated through email alerts

4. RESULTS (Sample Outputs)

Tabs: Medicine, Supplier, Purchase Order, Stock Transaction

Email alerts for low stock and expiry reminders

Purchase order approval/rejection workflows

Error messages for expired medicines

Flow execution for stock deduction

The screenshot displays a web browser window with multiple tabs, including 'Home | Salesforce', '- Student', and 'Developer Edition with Agentf'. The address bar shows the URL: naanmudhalvan.smartinternz.com/Student/guided_project_info/11065#. The main content area is titled 'Guided Project' and 'Project Workspace'. On the left, a sidebar menu includes 'Home', 'Projects', and 'Support'. The main workspace shows a project titled 'Medical Inventory Management' with a list of milestones: 'Milestone 1: Salesforce Account', 'Activity 1: Creating Developer Account', 'Activity 2: Account Activation', 'Milestone 2: Objects', 'Milestone 3: Tabs', 'Milestone 4: The Lightning App', 'Milestone 5: Fields', 'Milestone 6: Editing Of Page Layouts', and 'Milestone 7: Compact Layouts'. To the right, there is a sign-up form for 'Salesforce Developer Edition' with fields for 'First name', 'Last name', 'Email', 'Role', 'Company', 'Country', 'Postal Code', and 'Username'. Below the form, there is a list of instructions for the sign-up process.

1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. Country : 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
Click on sign me up after filling these.

5. ADVANTAGES & DISADVANTAGES

Advantages:

Reduces medicine wastage

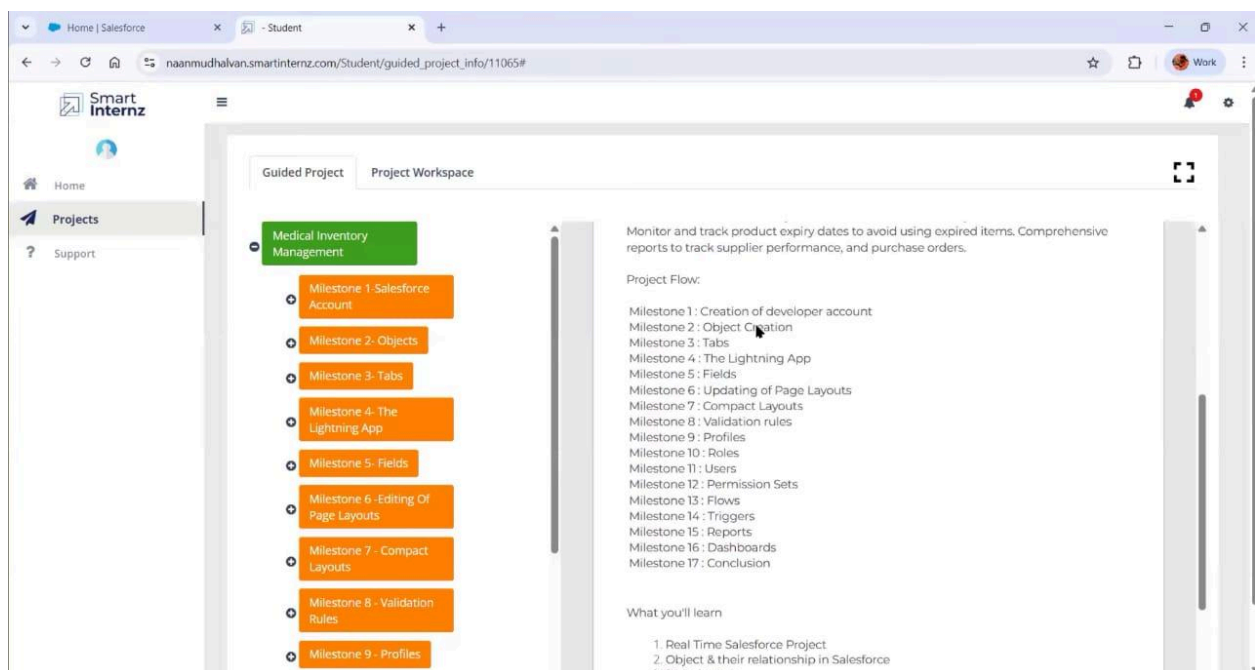
Prevents stock-outs

Improves patient safety

Automated alerts and reminders.

Disadvantages:

Requires internet and Salesforce access Initial setup cost



6. CONCLUSION

The Medical Inventory Management System successfully streamlines the operations of managing medicines and medical supplies through a structured, automated Salesforce application. It improves efficiency, reduces wastage, and ensures timely availability of essential medical items.

7. APPENDIX (Sample Apex Code)

// Trigger to prevent issuing expired medicine

trigger PreventExpiredIssue on Stock_Transaction_c (before insert) {

for (Stock_Transaction_ctxn: Trigger.new) {

Medicine_c med = [SELECT Expiry_Date_c FROM Medicine_c WHERE Id = :txn.Medicine_c];

if (med.Expiry_Date_c<Date.today()) {

txn.addError('Cannot issue expired medicine.');

}

// Scheduler for monthly expiry reminder

global class ExpiryReminderScheduler implements Schedulable {

global void execute(SchedulableContext sc) {

List<Medicine_c> meds = [SELECT Name, Expiry_Date_c FROM Medicine_c

WHERE Expiry_Date_c = NEXT_N_DAYS:30];

for (Medicine_c med: meds) {

Messaging.Single Email Message email = new Messaging.Single EmailMessage();

email.setToAddresses (new String[]{'admin@hospital.com'});

email.setSubject('Expiry Reminder: ' + med.Name);

email.setPlainTextBody(The medicine '+ med. Name + 'will expire soon. Please take action.");

Messaging.sendEmail (new Messaging.SingleEmailMessage[] (email));