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

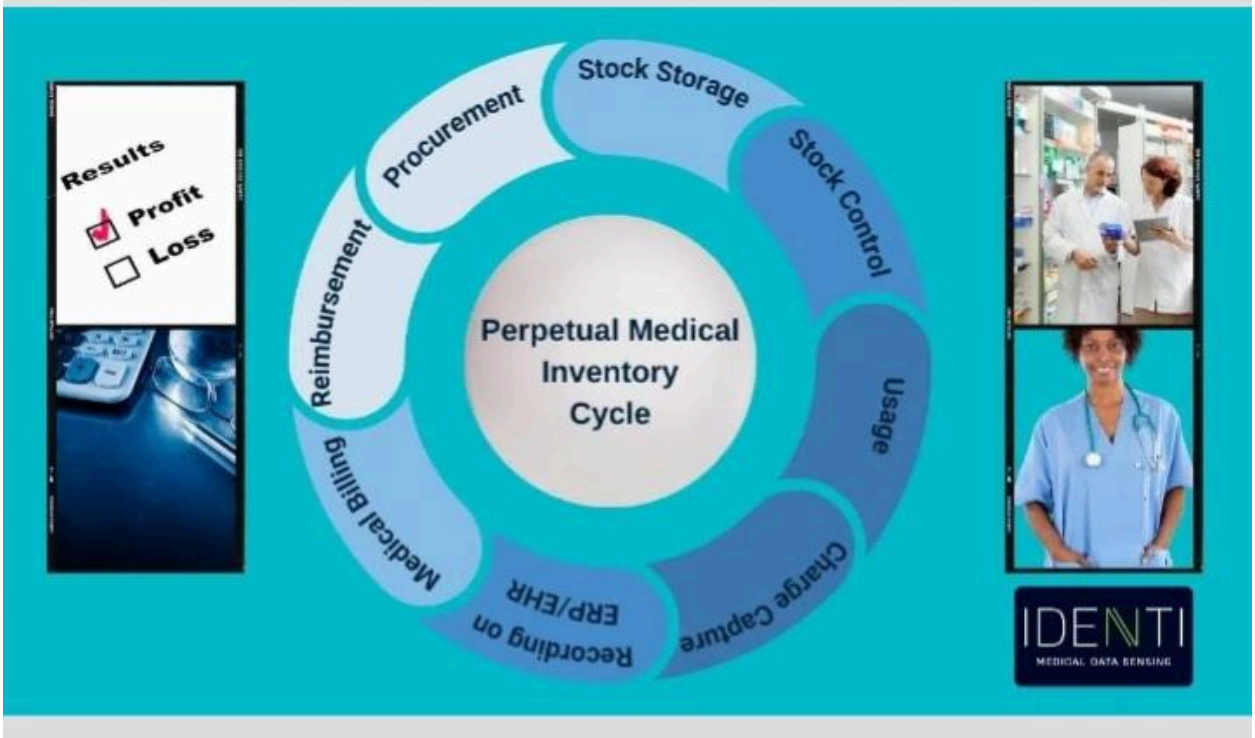
TEAM MEMBER : DINESH B

TEAM MEMBER : EBENEZER G

1. INTRODUCTION

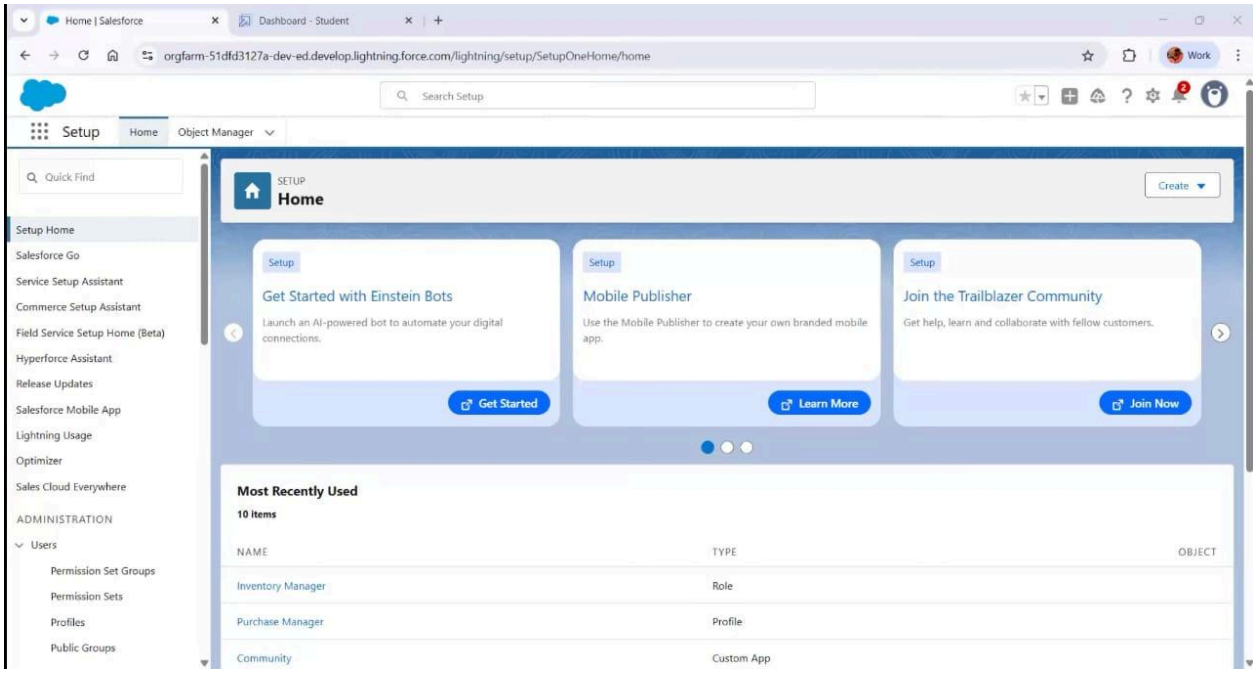
Project Overview

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

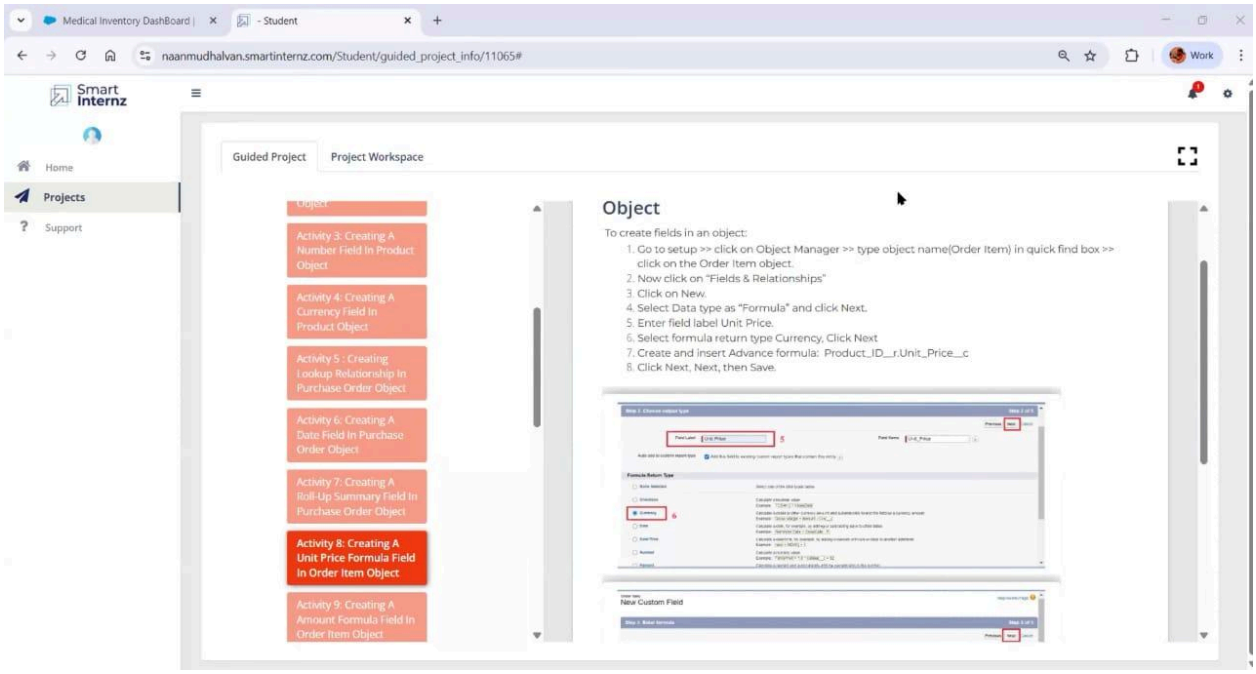
The main objective of this project is to enable healthcare organizations to efficiently manage inventory, avoid shortages/overstocking, improve accuracy, and ensure timely replenishment of medical supplies. It reduces manual intervention and enhances patient care by maintaining reliable stock availability.

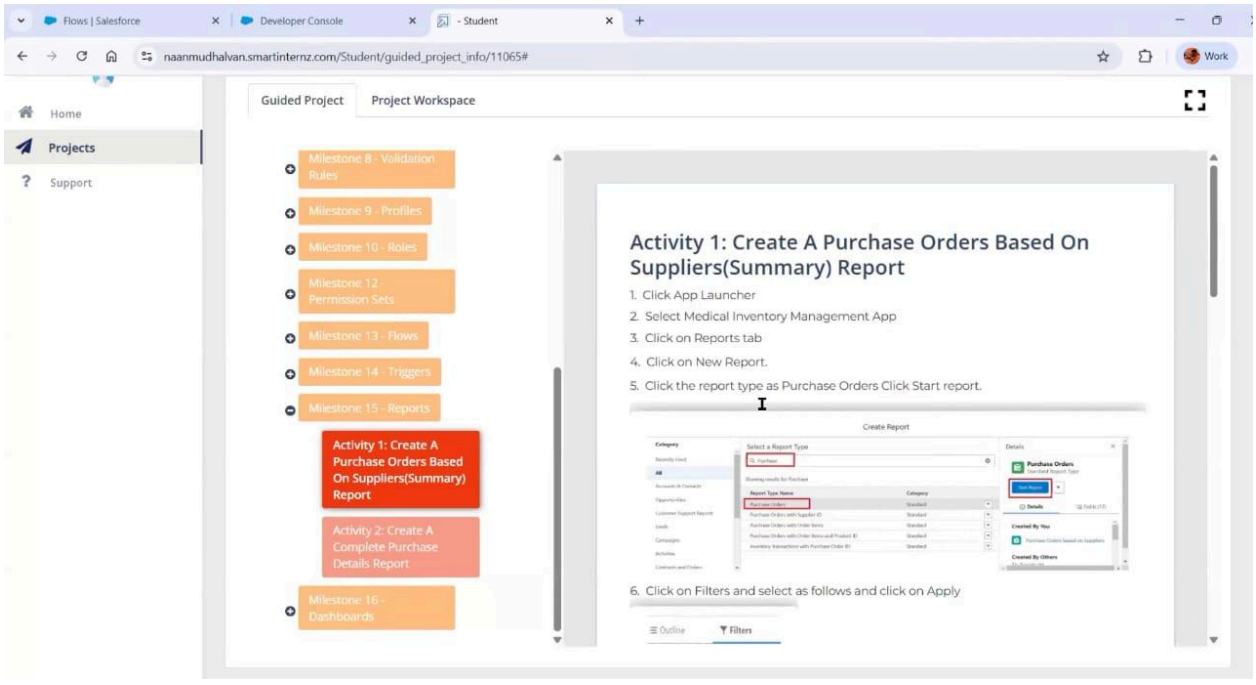


2. DEVELOPMENT PHASE

Creating Developer Account:

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





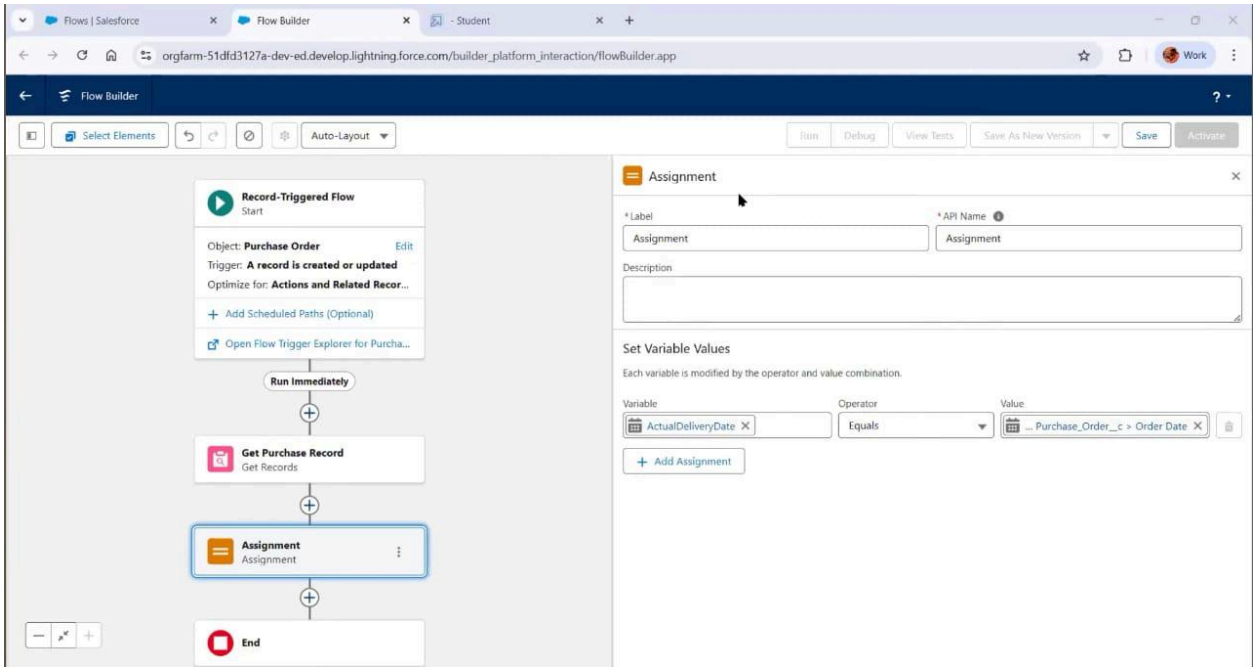
Created Objects:

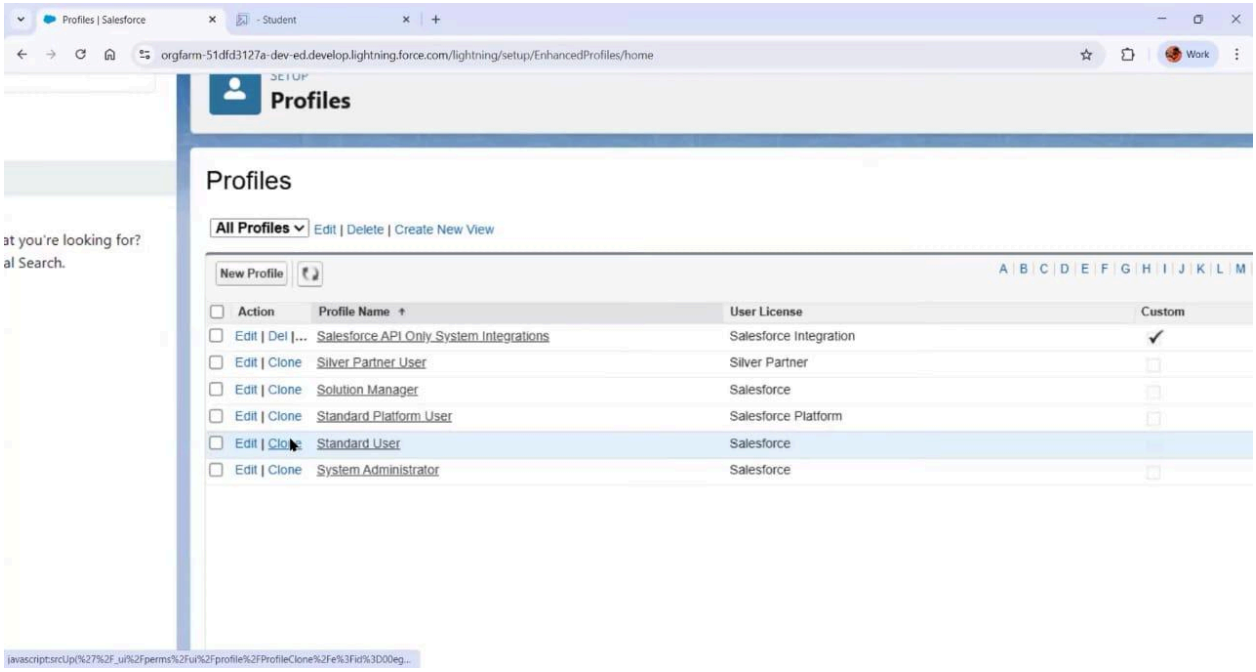
Medicine (Name, Category, Expiry Date, Stock Quantity, Price)

Supplier (Name, Contact, Email, Address)

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

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

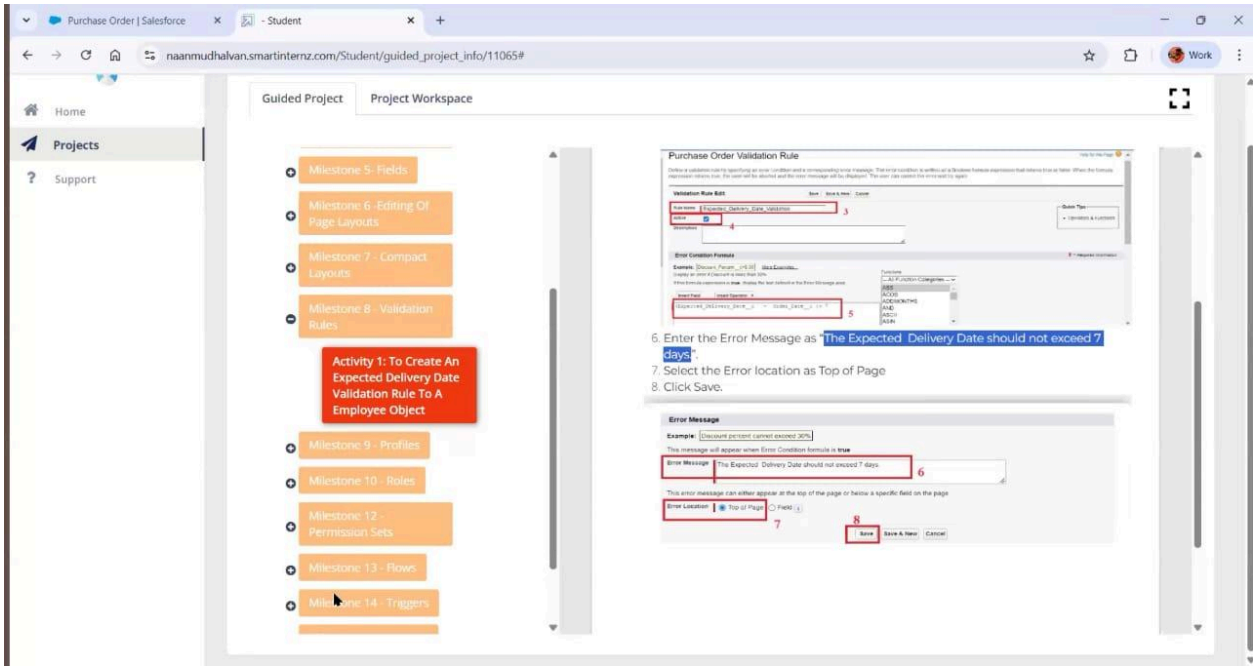


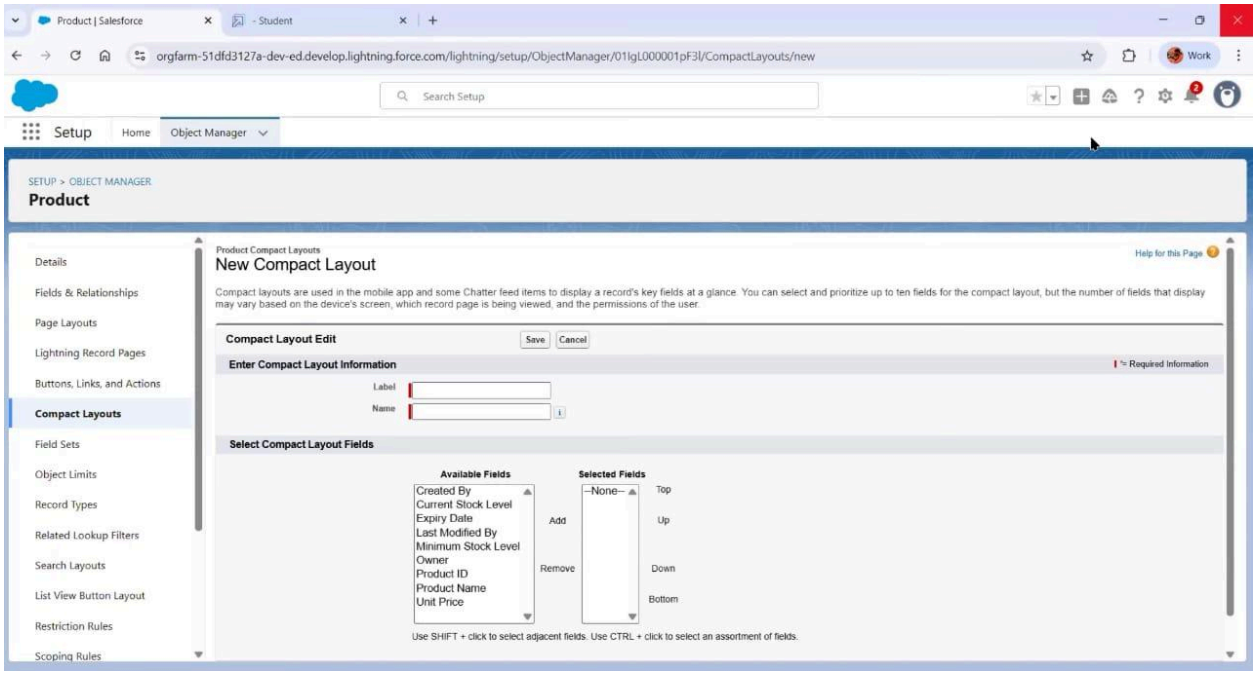


Configured Fields and Relationships:

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:

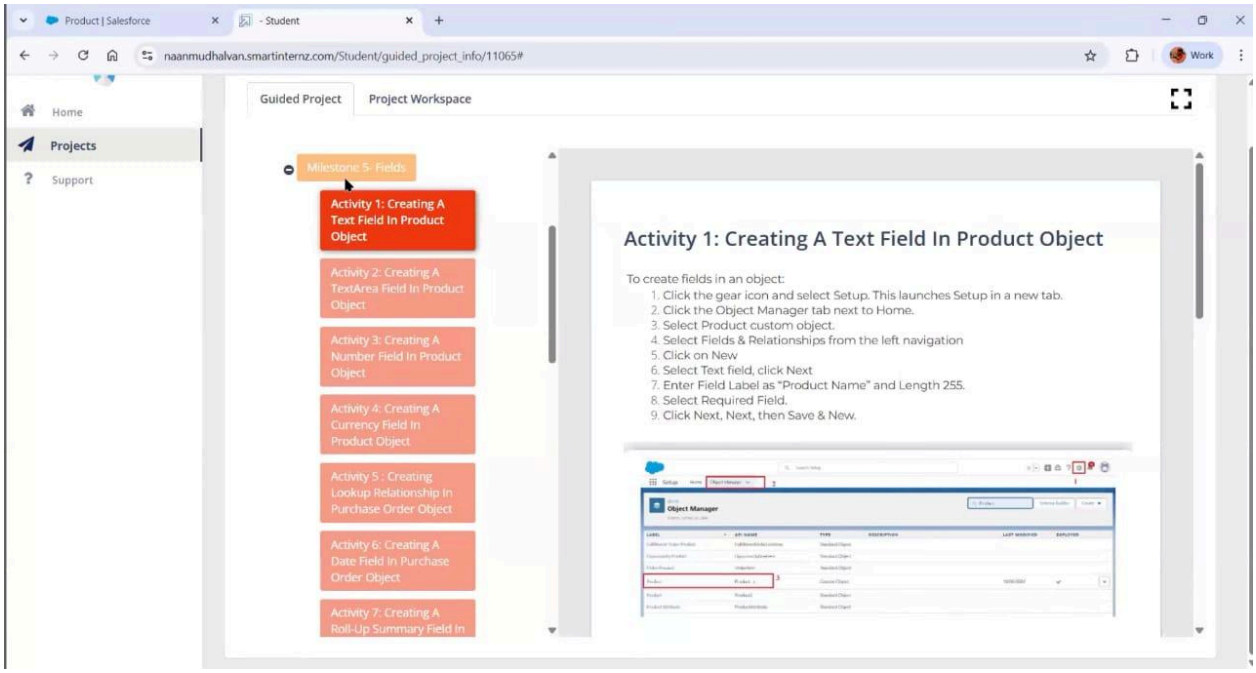
Automatic stock deduction when a transaction is recorded

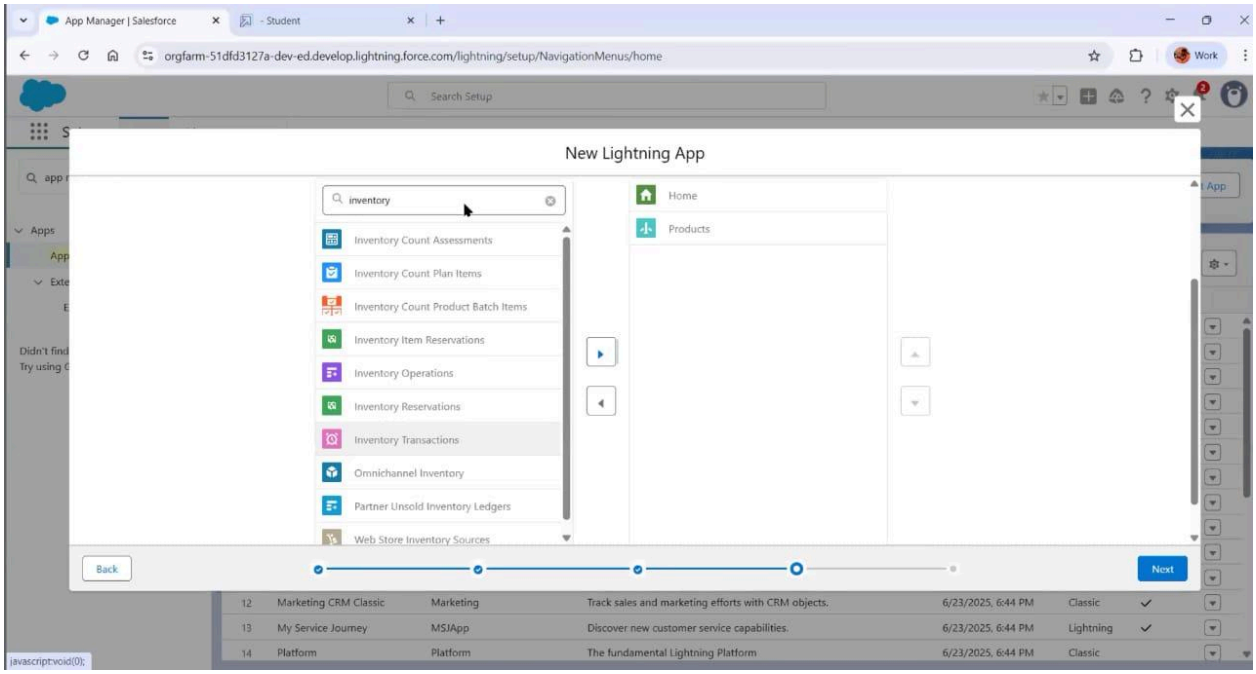
Reorder notification when stock falls below threshold

Validation Rules:

Expired medicines cannot be issued

Minimum order quantity must be greater than zero





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

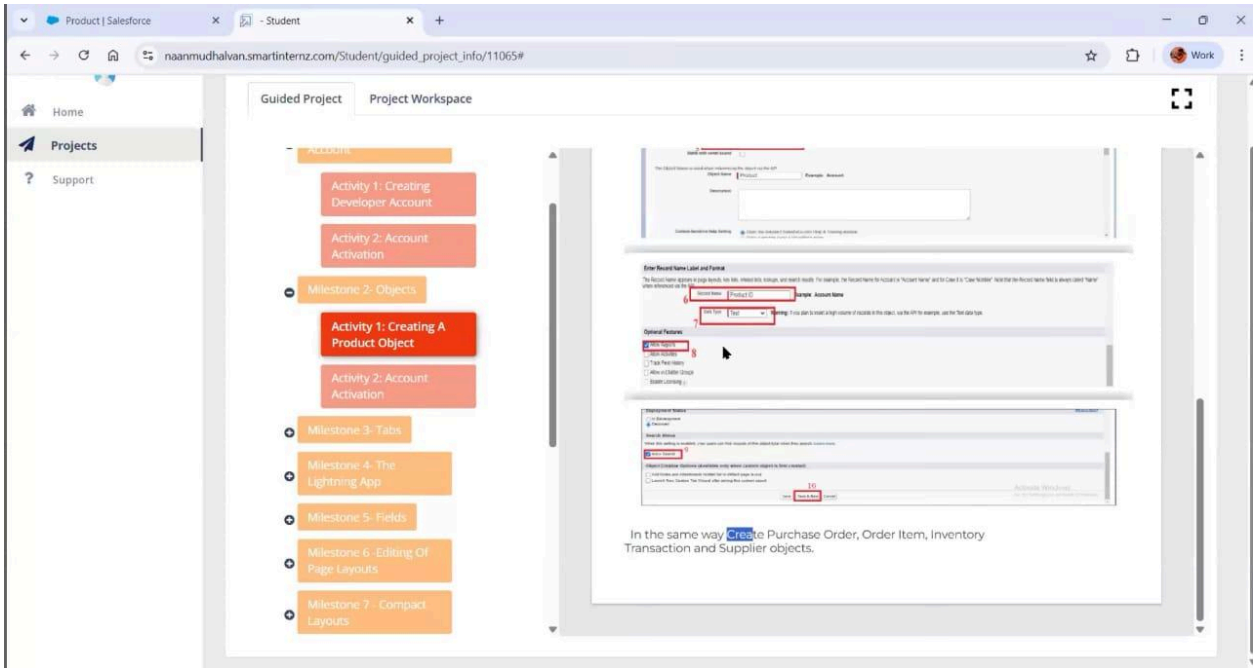
Email Templates Built:Low stock alert

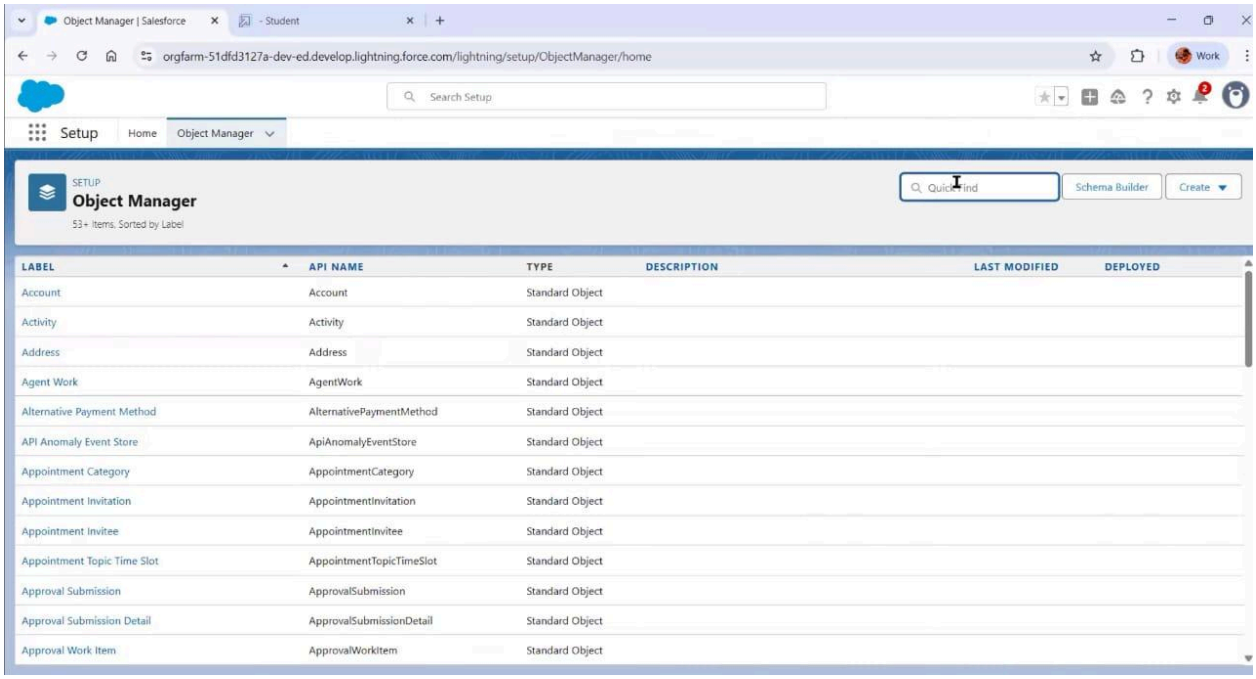
Expiry reminder

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)

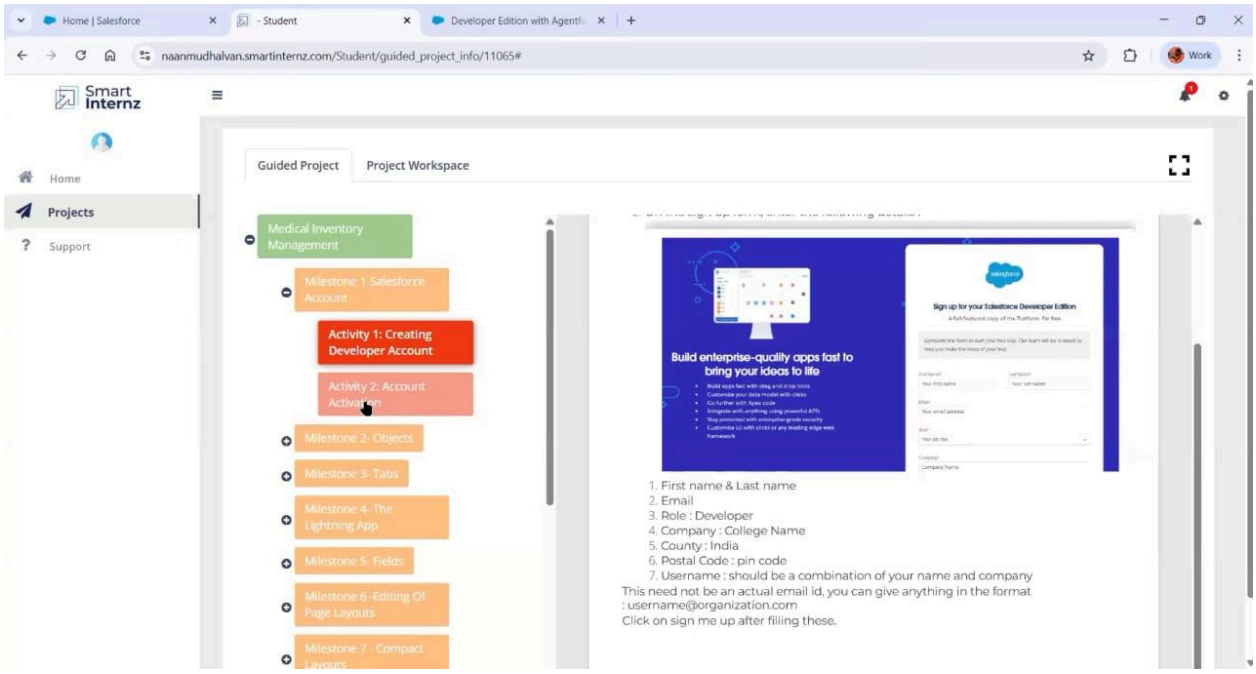
Tabs: Medicine, Supplier, Purchase Order, Stock Transaction

Email alerts for low stock and expiry reminders

Purchase order approval/rejection workflows

Error messages for expired medicines

Flow execution for stock deduction



5. ADVANTAGES & DISADVANTAGES

Advantages:

Reduces medicine wastage

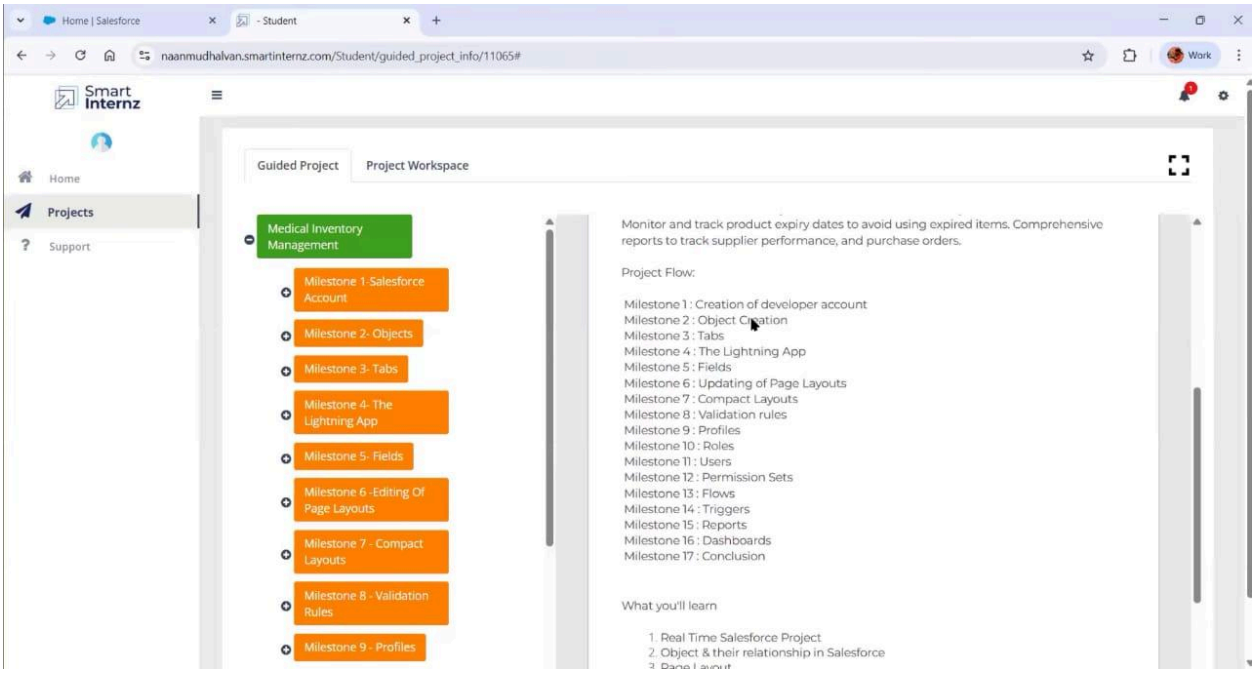
Prevents stock-outs

Improves patient safety

Automated alerts and reminders.

Disadvantages:

Requires internet and Salesforce accessInitial 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));
```