Title:

Salesforce-Based CRM and Inventory Automation for HandsMen Threads

1. Introduction

HandsMen Threads, a growing enterprise in the fashion domain, has embarked on a digital transformation journey through Salesforce to streamline its operations. This project involved implementing a customized CRM and inventory management system with automation at its core. By leveraging Salesforce tools such as Lightning App Builder, Apex, Flows, and Process Automation, the system is designed to improve operational efficiency, enhance customer relations, and ensure real-time inventory tracking.

2. Objective

- Establish a robust CRM and inventory solution using Salesforce.
- Design scalable and secure data models for customer, product, order, and inventory records.
- Implement business logic automation using Flows, Apex, Triggers, and Email Alerts.
- Enable loyalty tracking and automated customer engagement.
- Train users and deploy a fully functional application.

3. Tools and Technologies Used

- **Salesforce CRM Platform**
- **Lightning App Builder**
- **Apex Programming Language**
- **Flows & Process Automation**
- **Validation Rules**
- **Email Templates and Alerts**
- **Profiles, Roles & Permission Sets**
- **Batch Apex for Scheduled Jobs**

4. Step-by-Step Implementation

Step 1: Creating a Salesforce Developer Account

To begin the implementation, a Salesforce Developer Account is required.

Procedure:

- Visit https://developer.salesforce.com/signup
- Fill in the signup form:
    - First Name, Last Name
    - Email
    - Role: Developer
    - Company: (e.g., College Name)
    - Country: India
    - Postal Code
    - Username: any format like name@org.com
- Click **Sign Me Up**
- Activate the account via the verification email.

Step 2: Initial Setup and Navigation

- After activation, log in and go to **Setup**.
- Navigate using the gear icon → **Setup**
- Familiarize yourself with Object Manager, Flows, Users, and App Builder.

Step 3: Creating Custom Objects

Created the following custom objects for data modeling:

1. **HandsMen Customer**
2. **HandsMen Product**
3. **HandsMen Order**
4. **Inventory**
5. **Marketing Campaign**

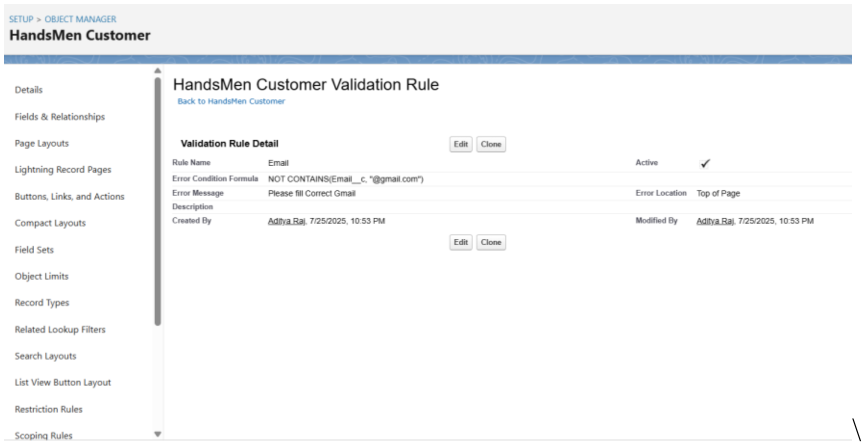
Each object includes specific fields, such as Name, Email, Phone, SKU, Price, Quantity, Loyalty Status, etc.

Step 4: Creating Custom Fields

For each object, created relevant fields such as:

- Email, Phone, Loyalty Status (Picklist)
- SKU, Price, Stock Quantity
- Order Number (Auto-Number), Status, Total Amount
- Warehouse, Campaign Dates

Also created **formula fields** and **validation rules** for data consistency.

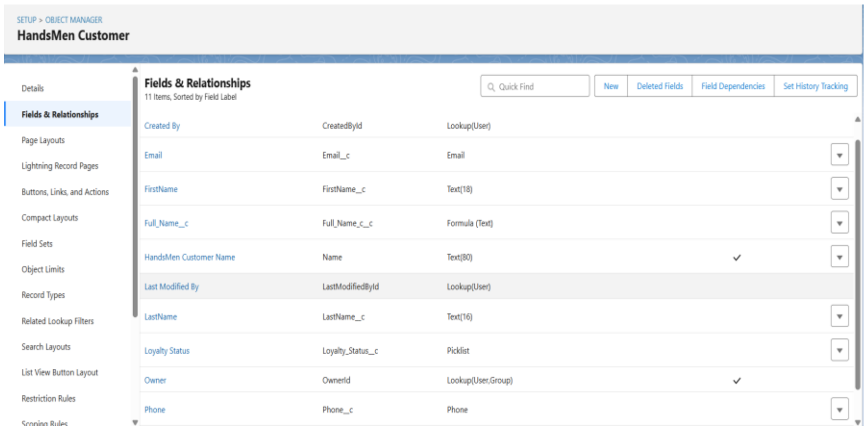


HandsMen Customer Validation Rule

Validation Rule Detail

Rule Name	Email	Active	✓
Error Condition Formula	NOT CONTAINS(Email__c, "@gmail.com")		
Error Message	Please fill Correct Gmail	Error Location	Top of Page
Description			
Created By	Aditya Raj, 7/25/2025, 10:53 PM	Modified By	Aditya Raj, 7/25/2025, 10:53 PM

\



Fields & Relationships
11 Items, Sorted by Field Label

Created By	CreatedById	Lookup(User)	
Email	Email__c	Email	
FirstName	FirstName__c	Text(18)	
Full_Name__c	Full_Name_c__c	Formula (Text)	
HandsMen Customer Name	Name	Text(80)	✓
Last Modified By	LastModifiedById	Lookup(User)	
LastName	LastName__c	Text(16)	
Loyalty Status	Loyalty_Status__c	Picklist	
Owner	OwnerId	Lookup(User,Group)	✓
Phone	Phone__c	Phone	

Step 5: Creating Relationships Between Objects

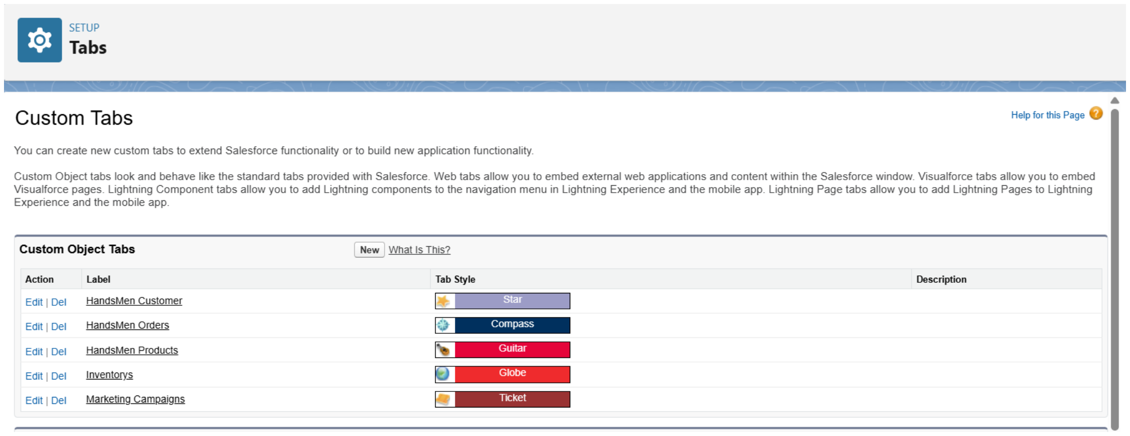
- **Lookup Relationships:**
  - Marketing Campaign → HandsMen Customer
  - HandsMen Product → HandsMen Order
  - HandsMen Order → HandsMen Customer

- **Master-Detail:**
    - Inventory → HandsMen Product

Step 6: Creating Tabs for Custom Objects

Created tabs for:

- HandsMen Customer
- HandsMen Product
- HandsMen Order
- Inventory



Custom Tabs

Help for this Page

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

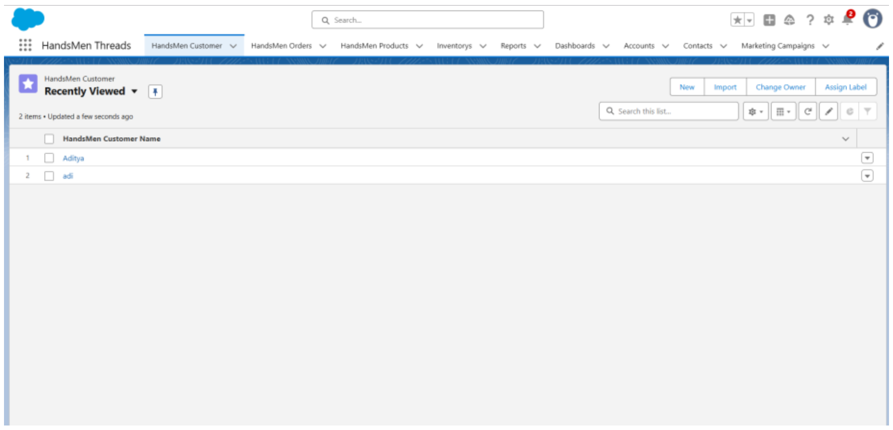
Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs New What Is This?

Action	Label	Tab Style	Description
Edit | Del	HandsMen Customer	Star	
Edit | Del	HandsMen Orders	Compass	
Edit | Del	HandsMen Products	Guitar	
Edit | Del	Inventorys	Globe	
Edit | Del	Marketing Campaigns	Ticket	

Step 7: Creating Lightning App: "HandsMen Threads"

- Used App Manager to create a new app
- Added custom objects and navigation items
- Assigned to System Administrator profile

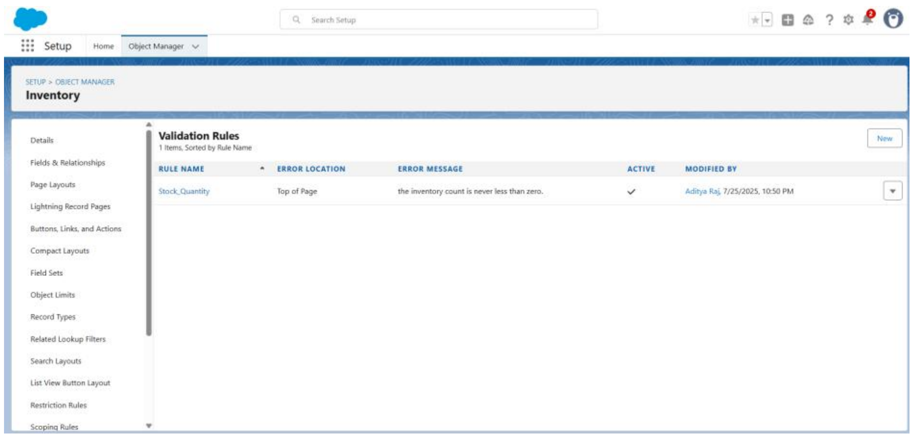


2 items • Updated a few seconds ago

	HandsMen Customer Name
1	Aditya
2	adi

Step 8: Implementing Business Logic & Validations

- Added **Validation Rules**:
    - Total_Amount ≤ 0 (Order)
    - Stock_Quantity ≤ 0 (Inventory)
    - Email must contain "@gmail.com" (Customer)



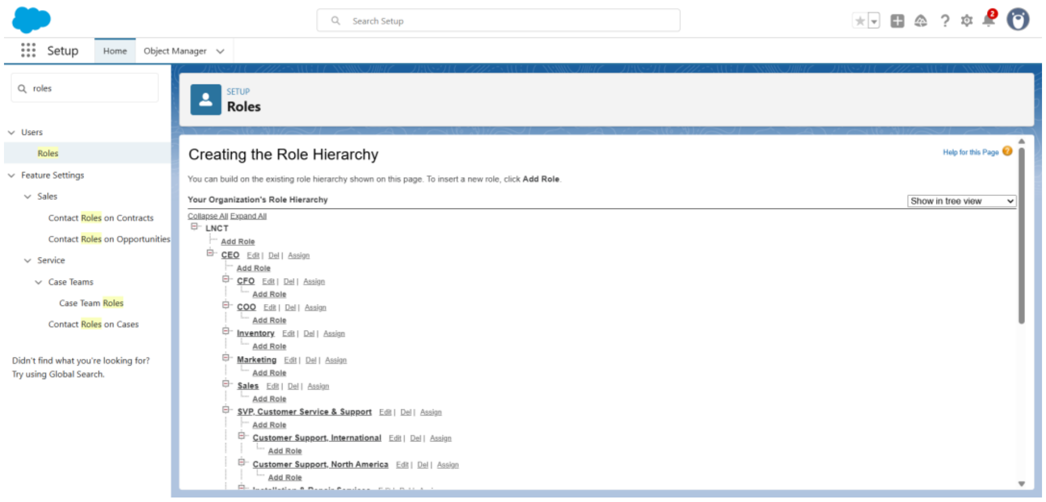
SETUP > OBJECT MANAGER
Inventory

Validation Rules
1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Stock_Quantity	Top of Page	the inventory count is never less than zero.	✓	Aditya Raj, 7/25/2025, 10:50 PM

Step 9: Creating Users, Roles, and Profiles

- Cloned “Standard User” profile → “Platform 1”
- Created Roles:
    - Sales
    - Inventory
    - Marketing
- Created Users: Niklaus, Kol, etc.
- Assigned Roles and Profiles accordingly



SETUP
Roles

Creating the Role Hierarchy

You can build on the existing role hierarchy shown on this page. To insert a new role, click **Add Role**.

Your Organization's Role Hierarchy — Show in tree view

```
Collapse All Expand All
LNCT
  Add Role
  CEO  Edit | Del | Assign
    Add Role
    CFO  Edit | Del | Assign
      Add Role
    COO  Edit | Del | Assign
      Add Role
    Inventory  Edit | Del | Assign
      Add Role
    Marketing  Edit | Del | Assign
      Add Role
    Sales  Edit | Del | Assign
      Add Role
    SVP, Customer Service & Support  Edit | Del | Assign
      Add Role
      Customer Support, International  Edit | Del | Assign
        Add Role
      Customer Support, North America  Edit | Del | Assign
        Add Role
```

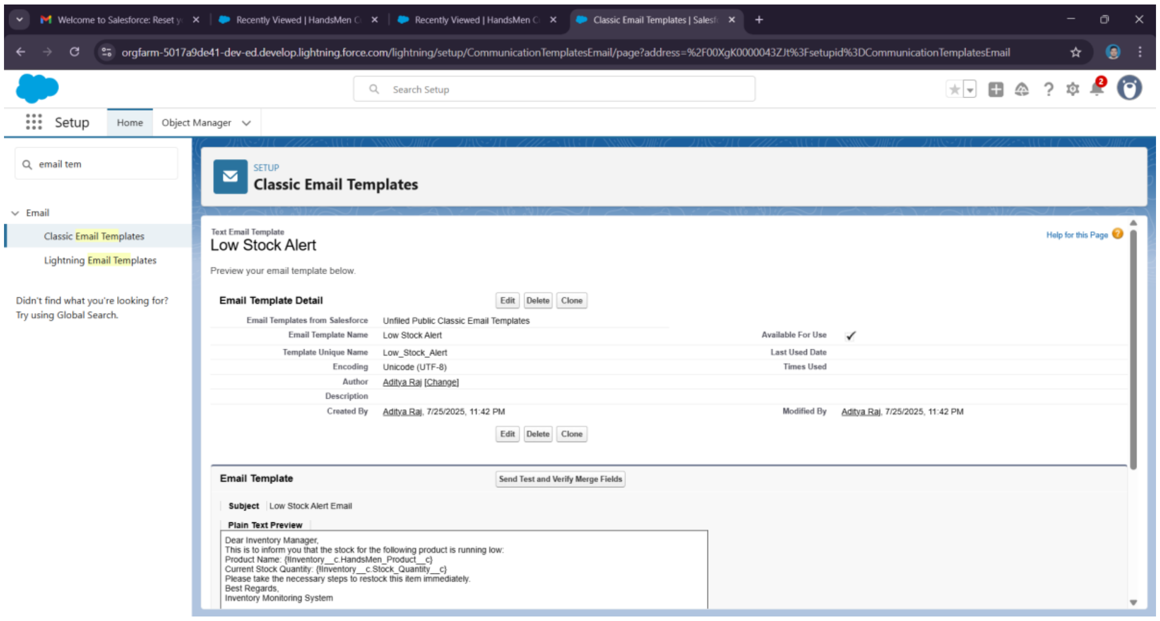
Step 10: Setting Up Permission Sets

- Created permission set “Permission_Platform_1”
- Granted CRUD access on custom objects
- Assigned to Platform 1 users

Step 11: Creating Email Templates

Created the following email templates:

1. **Order Confirmation**
2. **Low Stock Alert**
3. **Loyalty Program Update**



Welcome to Salesforce: Reset y
Recently Viewed | HandsMen C
Recently Viewed | HandsMen C
Classic Email Templates | Salesf
orgfarm-5017a9de41-dev-ed.develop.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/page?address=%2F00XgK0000043ZJt%3Fsetupid%3DCommunicationTemplatesEmail
Search Setup
Setup
Home
Object Manager
email tem
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for? Try using Global Search.
SETUP
Classic Email Templates
Help for this Page
Edit
Delete
Clone
Send Test and Verify Merge Fields

Text Email Template

Low Stock Alert

Preview your email template below.

Email Template Detail

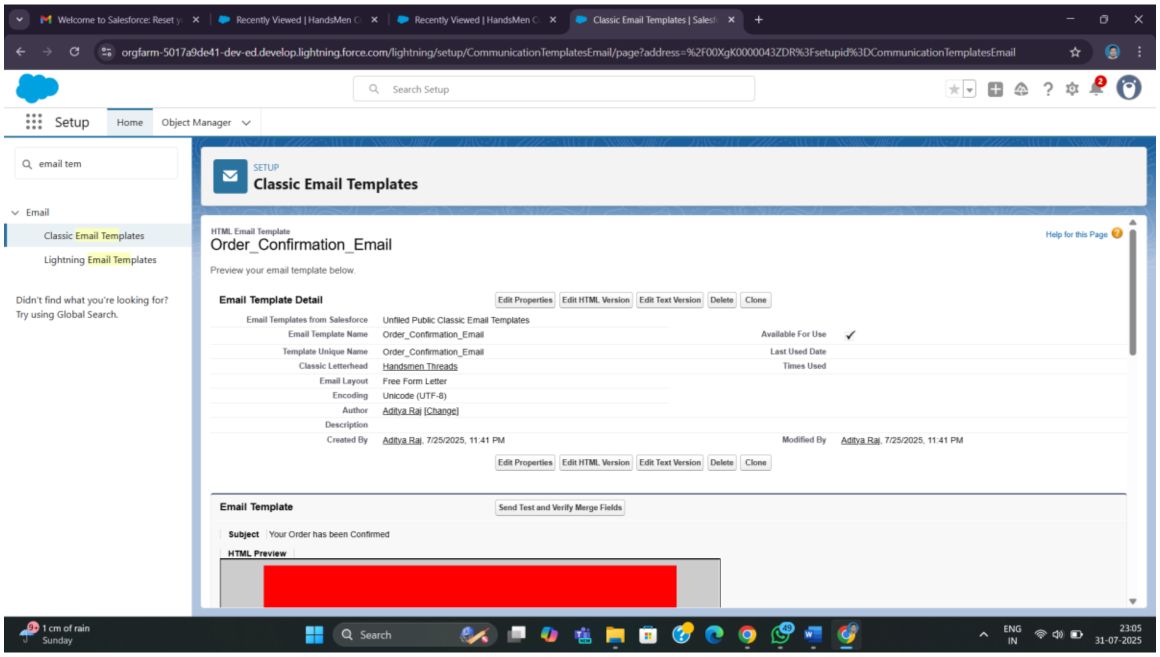
Email Templates from Salesforce	Unfiled Public Classic Email Templates		
Email Template Name	Low Stock Alert	Available For Use	✓
Template Unique Name	Low_Stock_Alert	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Aditya Raj [Change]		
Description			
Created By	Aditya Raj, 7/25/2025, 11:42 PM	Modified By	Aditya Raj, 7/25/2025, 11:42 PM

Email Template

Subject: Low Stock Alert Email

Plain Text Preview

```
Dear Inventory Manager,
This is to inform you that the stock for the following product is running low:
Product Name: {!Inventory__c.HandsMen_Product__c}
Current Stock Quantity: {!Inventory__c.Stock_Quantity__c}
Please take the necessary steps to restock this item immediately.
Best Regards,
Inventory Monitoring System
```



SETUP
Classic Email Templates

HTML Email Template
Order_Confirmation_Email

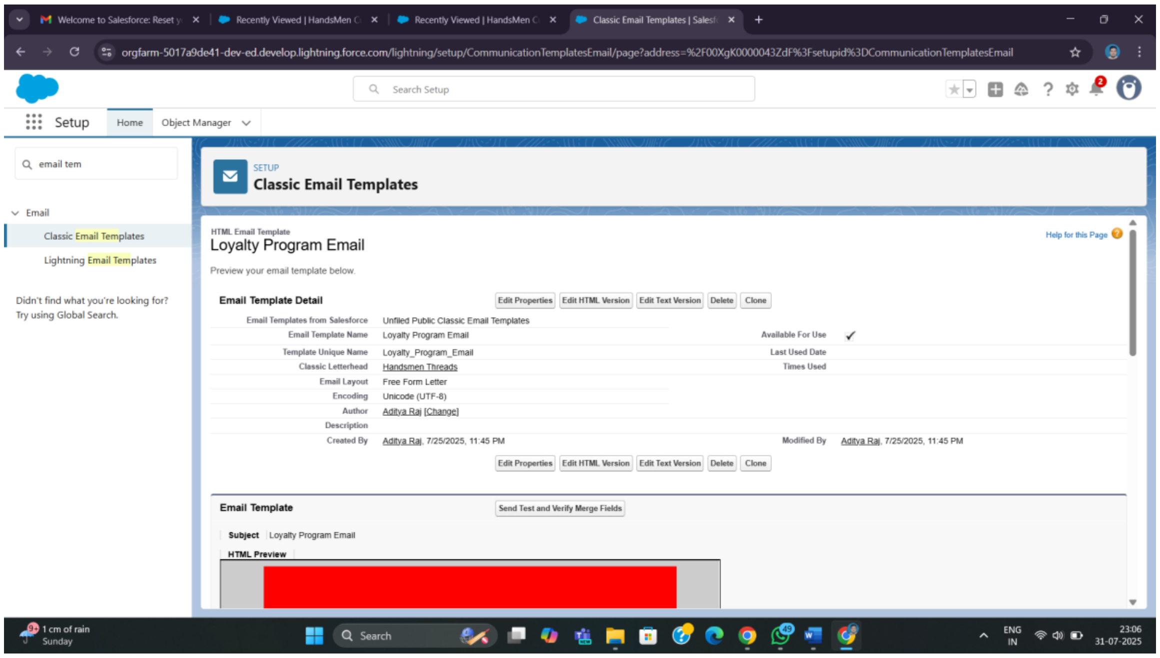
Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates		
Email Template Name	Order_Confirmation_Email	Available For Use	✓
Template Unique Name	Order_Confirmation_Email	Last Used Date	
Classic Letterhead	Handsmen Threads	Times Used	
Email Layout	Free Form Letter		
Encoding	Unicode (UTF-8)		
Author	Aditya Raj [Change]		
Description			
Created By	Aditya Raj, 7/25/2025, 11:41 PM	Modified By	Aditya Raj, 7/25/2025, 11:41 PM

Email Template

Subject: Your Order has been Confirmed



SETUP
Classic Email Templates

HTML Email Template
Loyalty Program Email

Preview your email template below.

Email Template Detail

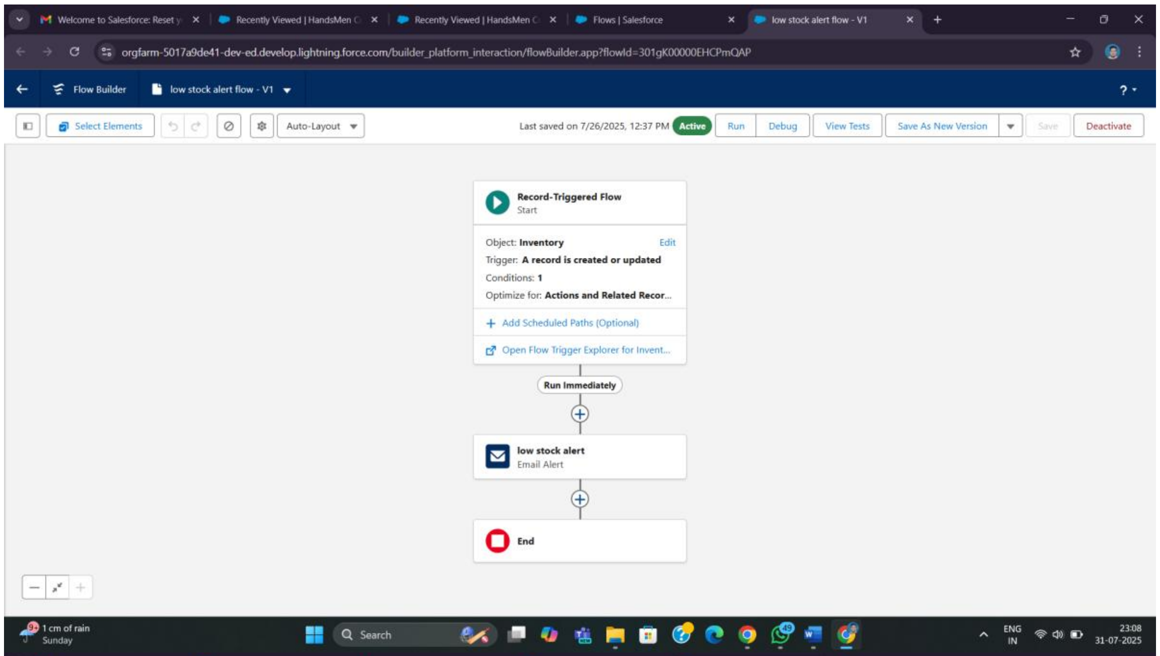
Email Templates from Salesforce	Unfiled Public Classic Email Templates		
Email Template Name	Loyalty Program Email	Available For Use	✓
Template Unique Name	Loyalty_Program_Email	Last Used Date	
Classic Letterhead	Handsmen Threads	Times Used	
Email Layout	Free Form Letter		
Encoding	Unicode (UTF-8)		
Author	Aditya Raj [Change]		
Description			
Created By	Aditya Raj, 7/25/2025, 11:45 PM	Modified By	Aditya Raj, 7/25/2025, 11:45 PM

Email Template

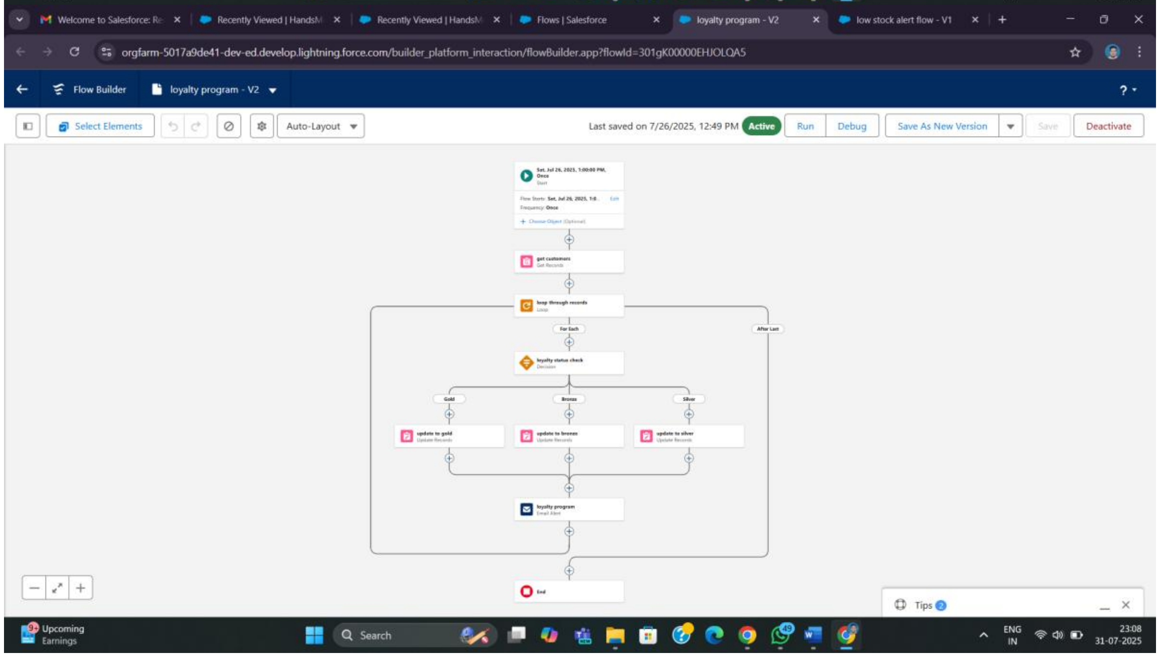
Subject: Loyalty Program Email

Step 12: Building Flows for Automation

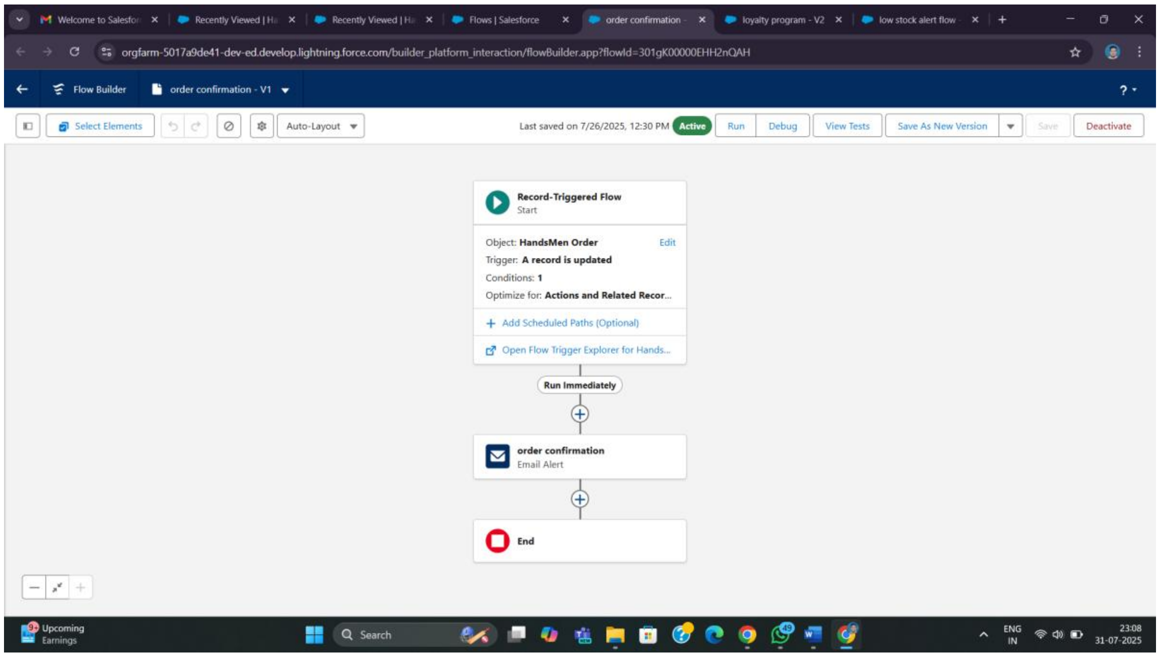
- **Order Confirmation Flow**: Sends email when order status is "Confirmed"
- **Low Stock Flow**: Alerts warehouse when inventory < 5
- **Loyalty Status Update Flow**: Updates loyalty based on total purchases daily


Welcome to Salesforce: Reset y
Recently Viewed | HandsMen C
Recently Viewed | HandsMen C
Flows | Salesforce
low stock alert flow - V1
orgfarm-5017a9de41-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301gK00000EHCPmQAP
Flow Builder
low stock alert flow - V1
Select Elements
Auto-Layout
Last saved on 7/26/2025, 12:37 PM
Active
Run
Debug
View Tests
Save As New Version
Save
Deactivate
Record-Triggered Flow
Start
Edit
Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Invent...
Run Immediately
low stock alert
Email Alert
End
1 cm of rain
Sunday
Search
ENG
IN
23:08
31-07-2025

Object: **Inventory**
Trigger: **A record is created or updated**
Conditions: **1**
Optimize for: **Actions and Related Recor...**



Flow Starts: **Sat, Jul 26, 2025, 1:0...**
Frequency: **Once**



Step 13: Writing Apex Code

Apex Trigger & Handler:

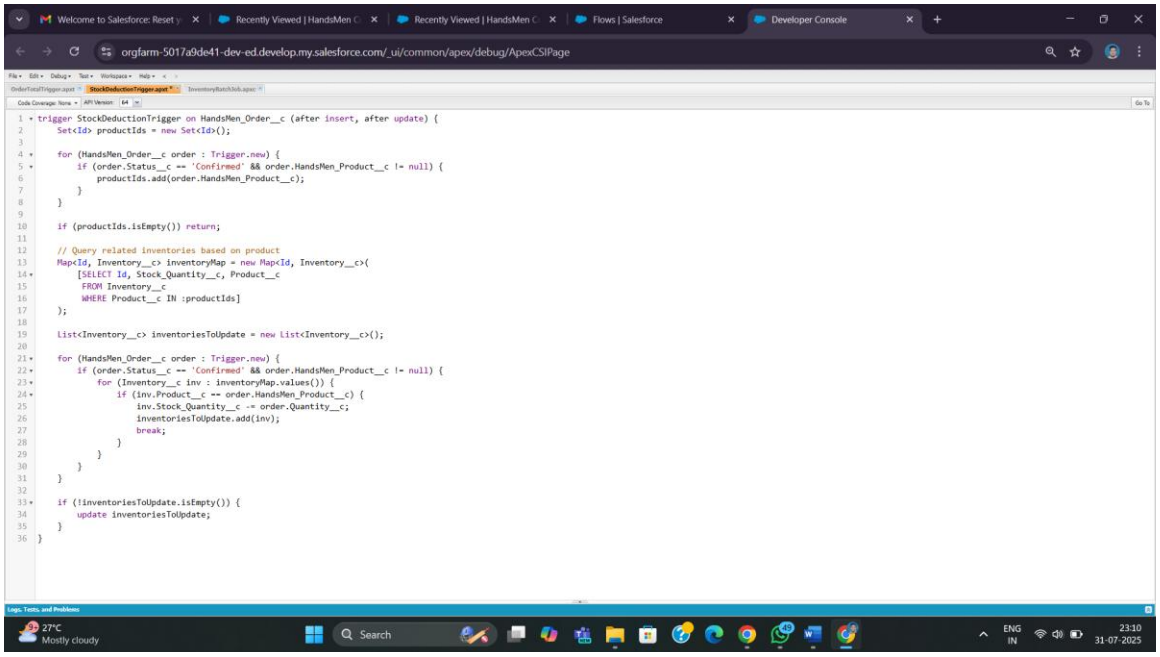
- Validates order quantity based on order status.

Batch Apex (InventoryBatchJob):

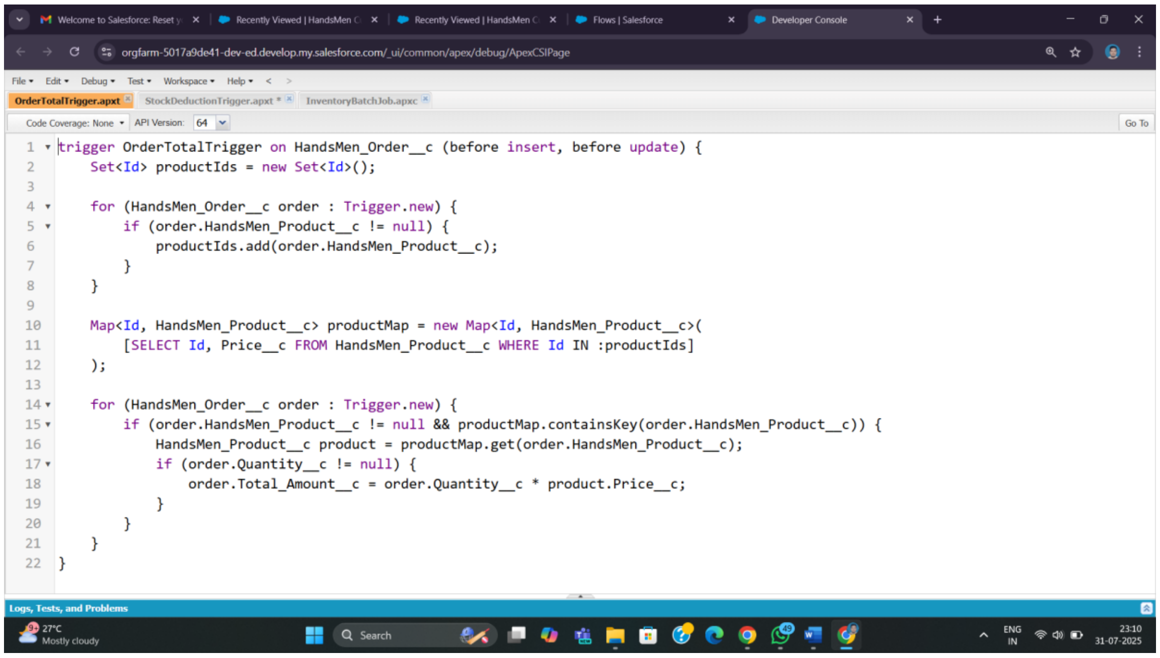
- Restocks items with low quantity.
- Scheduled to srun at midnight daily.

Execute Anonymous Code:

- Used to schedule the batch job.



```
trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
    Set<Id> productIds = new Set<Id>();

    for (HandsMen_Order__c order : Trigger.new) {
        if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
            productIds.add(order.HandsMen_Product__c);
        }
    }

    if (productIds.isEmpty()) return;

    // Query related inventories based on product
    Map<Id, Inventory__c> inventoryMap = new Map<Id, Inventory__c>(
        [SELECT Id, Stock_Quantity__c, Product__c
         FROM Inventory__c
         WHERE Product__c IN :productIds]
    );

    List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();

    for (HandsMen_Order__c order : Trigger.new) {
        if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
            for (Inventory__c inv : inventoryMap.values()) {
                if (inv.Product__c == order.HandsMen_Product__c) {
                    inv.Stock_Quantity__c -= order.Quantity__c;
                    inventoriesToUpdate.add(inv);
                    break;
                }
            }
        }
    }

    if (!inventoriesToUpdate.isEmpty()) {
        update inventoriesToUpdate;
    }
}
```

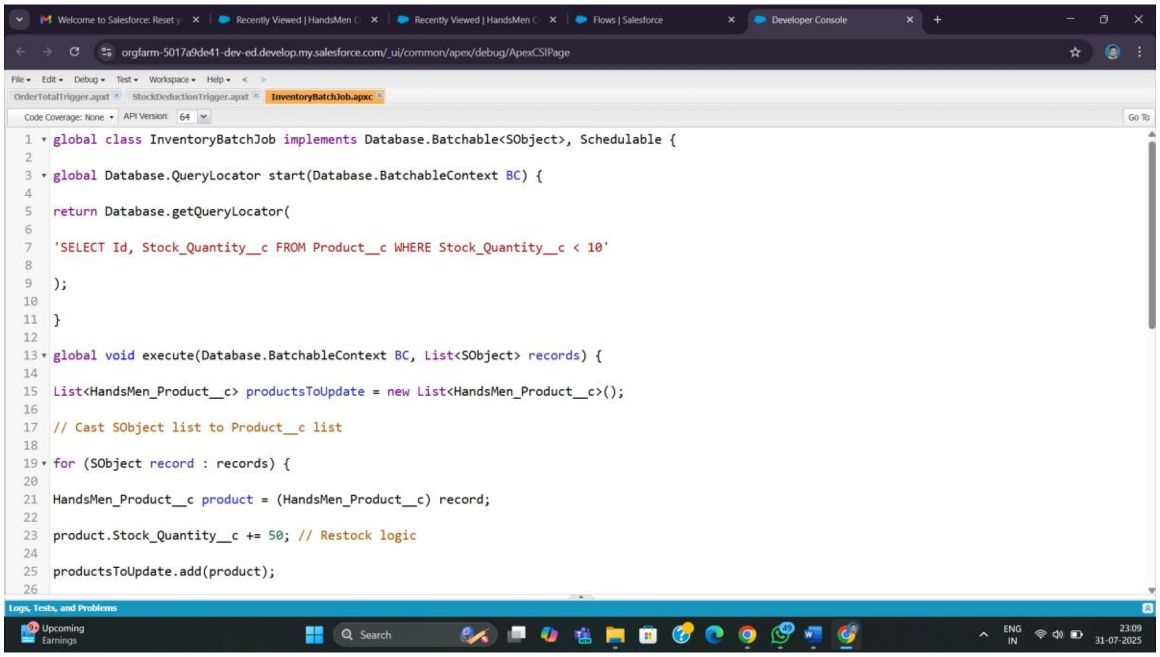


```
trigger OrderTotalTrigger on HandsMen_Order__c (before insert, before update) {
    Set<Id> productIds = new Set<Id>();

    for (HandsMen_Order__c order : Trigger.new) {
        if (order.HandsMen_Product__c != null) {
            productIds.add(order.HandsMen_Product__c);
        }
    }

    Map<Id, HandsMen_Product__c> productMap = new Map<Id, HandsMen_Product__c>(
        [SELECT Id, Price__c FROM HandsMen_Product__c WHERE Id IN :productIds]
    );

    for (HandsMen_Order__c order : Trigger.new) {
        if (order.HandsMen_Product__c != null && productMap.containsKey(order.HandsMen_Product__c)) {
            HandsMen_Product__c product = productMap.get(order.HandsMen_Product__c);
            if (order.Quantity__c != null) {
                order.Total_Amount__c = order.Quantity__c * product.Price__c;
            }
        }
    }
}
```

Step 14: Scheduling Batch Job

Scheduled the InventoryBatchJob using System.schedule:

System.schedule('Daily Inventory Sync', '0 0 0 * * ?', new InventoryBatchJob());


orgfarm-5017a9de41-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
OrderTotalTrigger.apxt
StockDeductionTrigger.apxt
InventoryBatchJob.apxc
Developer Console
Logs, Tests, and Problems

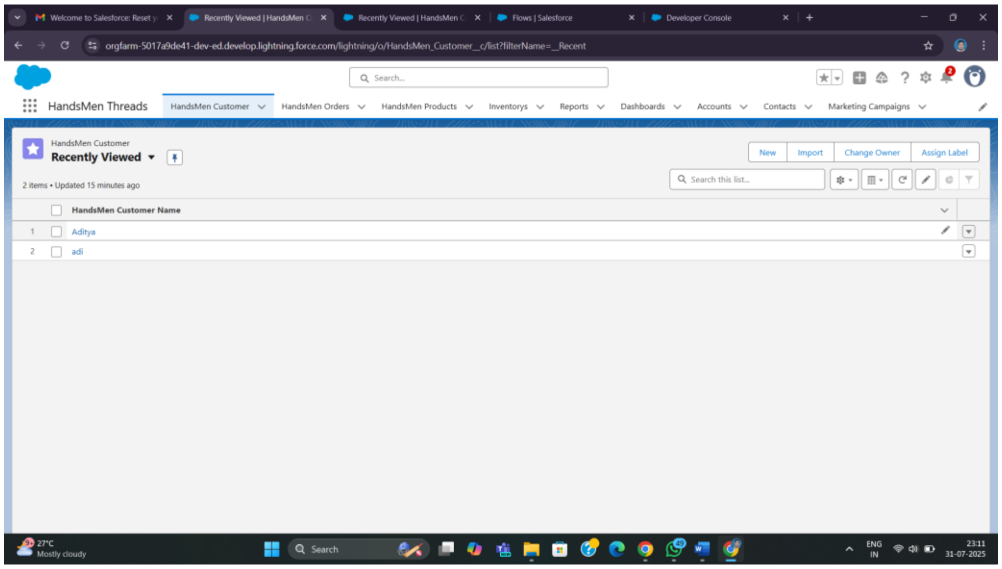
```
global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {

global Database.QueryLocator start(Database.BatchableContext BC) {

return Database.getQueryLocator(

'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'

);

}

global void execute(Database.BatchableContext BC, List<SObject> records) {

List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();

// Cast SObject list to Product__c list

for (SObject record : records) {

HandsMen_Product__c product = (HandsMen_Product__c) record;

product.Stock_Quantity__c += 50; // Restock logic

productsToUpdate.add(product);

```

Step 15: Testing & Debugging

- Tested validation rules with negative scenarios.
- Used sample data for Orders, Products, and Customers.
- Performed Flow Debugging and Log Tracing.

Step 16: Deployment & User Training

- Deployed the solution in developer org.
- Walked users through:
  - App navigation
  - Flow triggers
  - Email responses
- Conducted post-deployment monitoring.



HandsMen Customer
Recently Viewed

2 items • Updated 15 minutes ago

	HandsMen Customer Name
1	Aditya
2	adi

5. Deliverables

- Custom Salesforce App with all objects
- Apex Classes, Triggers, Flows, and Scheduled Jobs
- Email Templates and Alerts
- Fully Documented Solution Design
- User Roles, Profiles, and Training Guide

6. Learning Outcomes

- Real-world Salesforce app development lifecycle
- Data modeling and entity relationships
- Business automation using Flow and Apex
- Apex triggers and asynchronous logic
- User access management
- Email automation and customer engagement
- Deployment and troubleshooting

7. Future Scope

The current solution serves as a foundation for scaling business operations at HandsMen Threads. Future developments may include:

- **Customer Portal Integration:** Creating a self-service portal for order tracking, loyalty point redemption, and personalized offers.
- **Advanced Analytics & Reporting:** Implement dashboards with AI-powered insights using Salesforce Einstein.
- **Mobile App Extension:** Build a mobile-first interface using Salesforce Mobile SDK for on-the-go access by sales and warehouse teams.
- **Payment Gateway Integration:** Automate online transactions and payment confirmations within the platform.
- **Third-Party Integration:** Link with ERP systems or e-commerce platforms like Shopify or WooCommerce for end-to-end automation.
- **Multi-location Inventory Tracking:** Track stock movement across various warehouse locations and apply geolocation intelligence.
- **WhatsApp/SMS Notifications:** Add multi-channel communication for order updates and low-stock alerts.

8. Conclusion

This project has effectively transformed the manual order and inventory operations of HandsMen Threads into a modern, automated Salesforce system. By integrating customer management, order processing, inventory control, and scheduled automation, this solution not only improves operational efficiency but also elevates customer satisfaction through timely communication and personalized rewards.

Prepared By:
Aditya Raj
MCA 2nd Semester
Lakshmi Narain
College of
Technology