HandsMen Threads: Elevating the Art of Sophistication in Men's Fashion

Abstract

The core of this project was the custom-built Salesforce CRM designed for HandsMen Threads. It was developed to elevate customer engagement, streamline data movement, and automate vital business functions. The system integrates features like instant order updates, loyalty program management, inventory alerts, and automatic finance reporting. A central design principle ensures data integrity is maintained directly via the UI, which is key for consistent, reliable business performance.

Introduction

Given that HandsMen Threads operates in a dynamic, customer-centric market, a unified system for handling key business areas—specifically customer data, sales, inventory, and communication—was essential for continued success. They opted for Salesforce CRM based on its reputation for flexibility and scalability. The project involved deep utilization of the platform, incorporating features like standard objects (reports, dashboards), custom objects, various automation tools (validation rules, triggered and scheduled flows), and robust communication features (email alerts and templates).

Objective

The project will integrate several new processes into the business workflow to improve customer service and operational efficiency:

1. **Automated Order Confirmations:** Post-order confirmation, customers will receive an email update, fostering engagement and strengthening customer relations.

2. **Dynamic Loyalty Program:** Customer loyalty statuses will be updated based on purchase history, enabling personalized rewards and promoting repeat business.

3. **Proactive Stock Alerts:** When stock levels drop below five units, automatic emails will notify the warehouse team, ensuring timely restocking and preventing runout.

4. **Scheduled Bulk Order Updates:** Daily at midnight, the system will process bulk orders, updating financial records and adjusting inventory, ensuring accurate stock levels for daily operations."

Methodology / Techniques Used

Salesforce

Salesforce functions as a unified, cloud-based platform that equips businesses with the necessary tools—spanning sales, service, marketing, and analytics—to manage customer relationships, streamline processes, and achieve growth. The main offering, Salesforce CRM, is a vital tool for tracking customer interactions, automating key tasks, and personalizing engagement. By centralizing data across the organization, it facilitates easier lead conversion, efficient deal management, and the provision of superior service.

Custom Data Modelling: Created custom objects and relationships to handle unique business data. Some of the custom objects that were required and were created are,

- Handsmen Customers
- HandsMen Orders
- HandsMen Products
- Intventory
- Marketing Campaign

Validation Rules: Implemented directly in the UI to maintain data accuracy. We used multiple validation rules that will be discussed later in the report.

Process Builder / Flow Builder: Automated workflows for order confirmation, stock alerts, and loyalty program management.

Email Alerts: Configured with workflows to send real-time notifications. These Emails were triggered by using the Triggered floes and by the scheduled triggered flows, which when activated will send the email to the customer or to the inventory owner.

Scheduled Apex Jobs / Scheduled Flows: Set to execute daily tasks like bulk order processing.

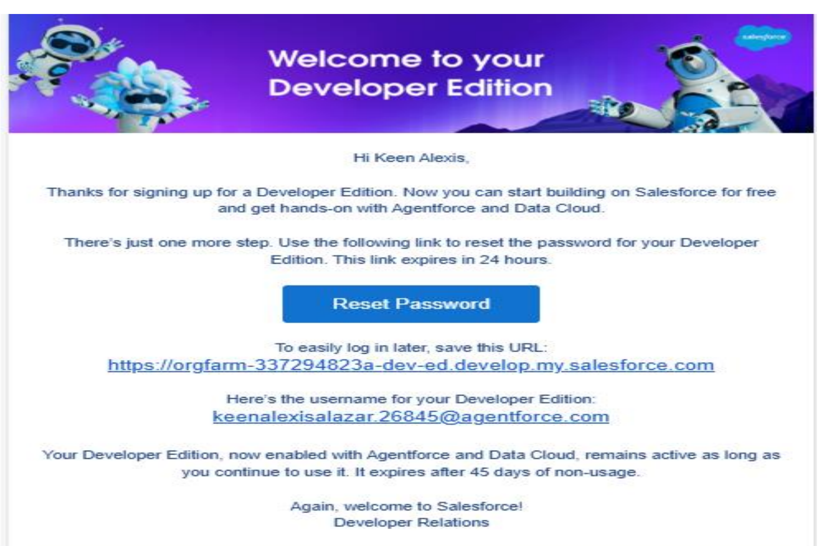
Reports & Dashboards: Visual tools for monitoring orders, stock status, and customer trends.

Detailed Execution of Project Plan

Developer Account Creation/ Setup

Salesforce org was created by the link provided, https://developer.salesforce.com/signup,

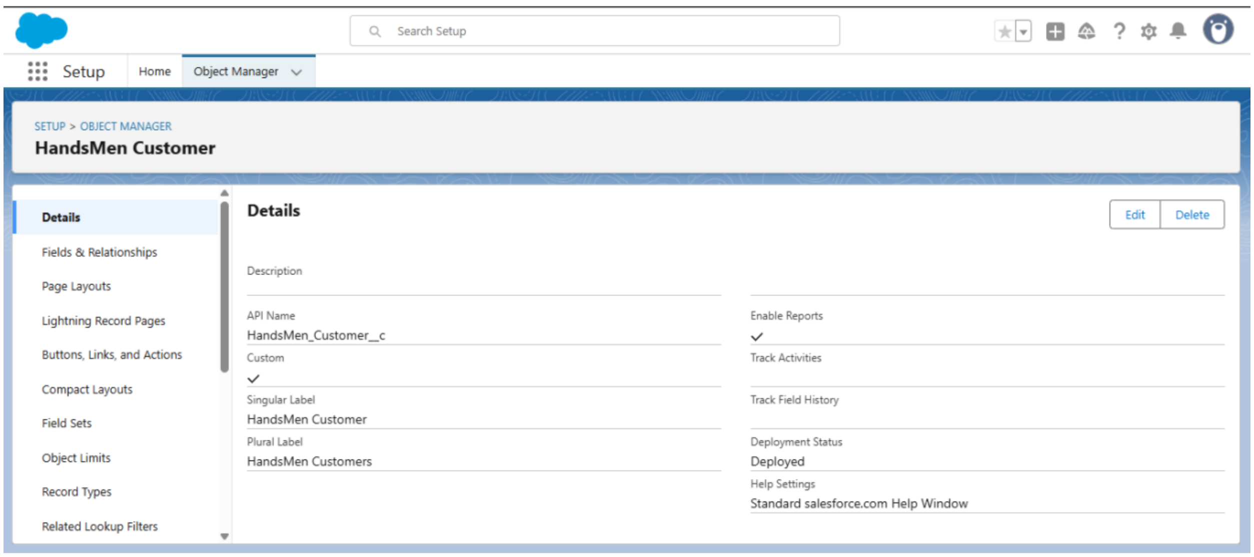
The account was verified by resetting the password, and our Developer Edition Org was successfully created.



Data Management - Objects

We were required to create some custom objects in our org, the custom objects were,

- HandsMen Customers
- Hands Men Orders
- HandsMen Products
- Inventory
- Marketing Campaign



Handsmen Customers:- This custom object was made to have the information about the customers we have.

HandsMen Orders:- This custom object was created to track and manage the orders ordered by the respective customers.

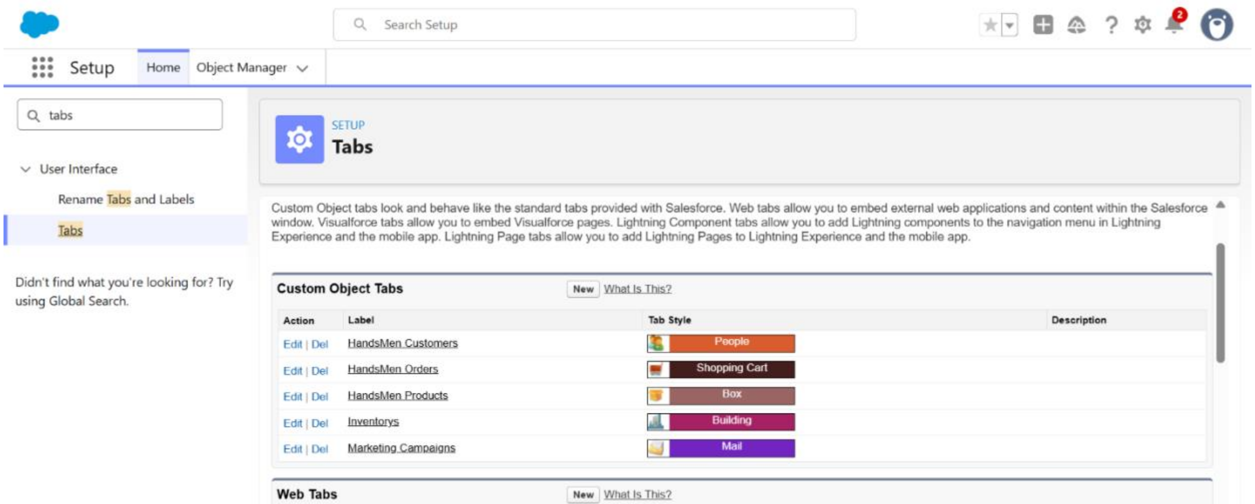
HandsMen Products:- This custom object was created to manage the available products.

Inventory:- This custom object was created to see and track the stock of the products and to track how much and where the stock is stored in the warehouse.

Marketing Campaign:- This custom object was created to manage the marketing events.

Data Management – Tabs

The HandsMen Threads Salesforce CRM utilizes custom tabs to offer easy and direct access to the specialized data models and custom objects developed for the brand. These tabs specifically serve as gateways for managing core operational entities like orders, customer loyalty status, inventory levels, and critical warehouse alerts.



Tabs Created for these custom objects,

- HandsMen Customers (People)
- Hands Men Orders (Shopping Cart)
- HandsMen Products (Box)
- Inventory (Building)
- Marketing Campaign (Mail)

Data Management – App Manager

To unify all CRM functionalities into a single, cohesive interface, a custom Lightning Application named "HandsMen Threads" was developed via the Salesforce App Manager. This application serves as the primary digital workspace for all HandsMen Threads departments (Sales, Inventory, Finance, and Customer Service) by integrating all necessary tabs, objects, and components.

Included Tabs:

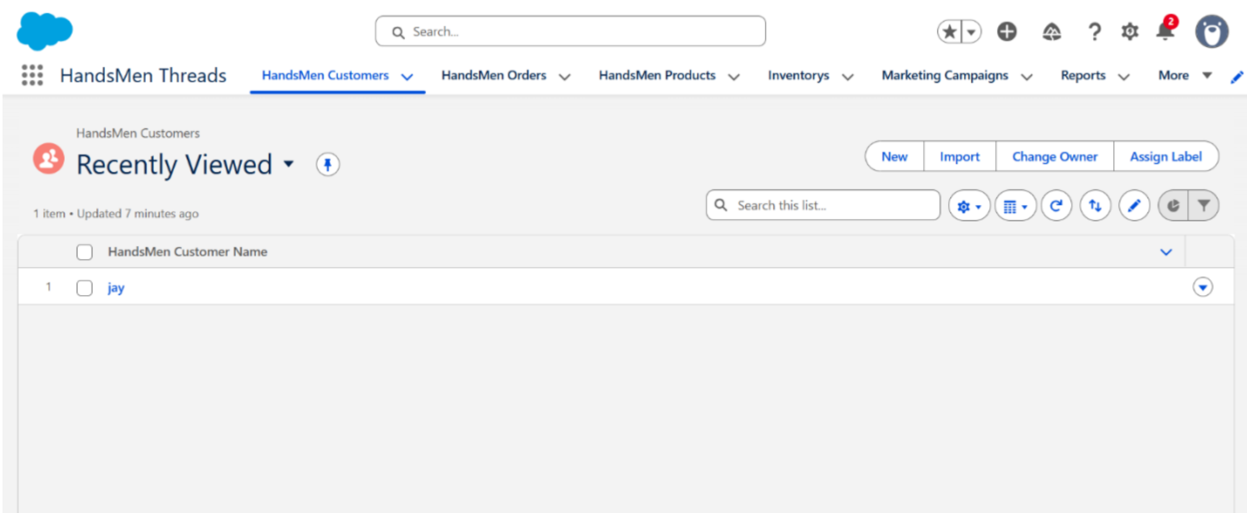
The tabs included were the combination of the custom objects I created and the standard objects received from salesforce. The list of all the objects are below: -

Custom objects

- HandsMen Customers
- Hands Men Orders
- HandsMen Products
- Inventory
- Marketing Campaign

Standard objects

- Reports
- Dashboards



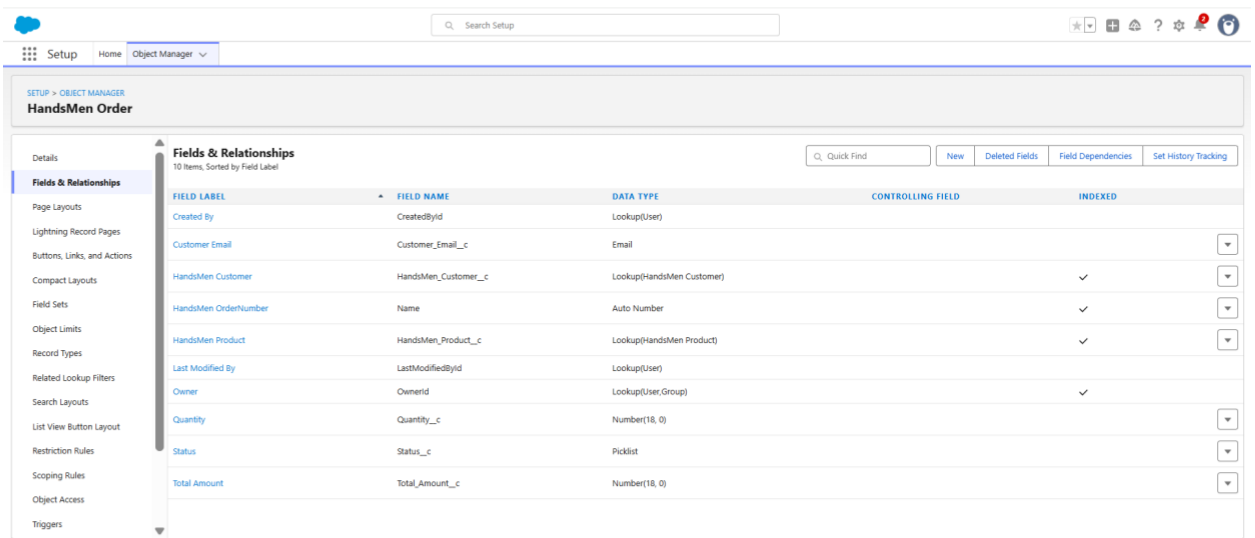
Data Management – Fields

I was supposed to create some fields into the respective custom objects.

The HandsMen Threads CRM system utilizes Custom Objects—specifically designed to represent core business entities like Customers, Orders, Products, and Inventory. To meet the organization's precise operational and data requirements, each object was configured with custom fields. This careful customization ensures that every business process can be accurately captured, tracked, and automated within the Salesforce environment.

Custom Fields: Object Name	**Key Fields**
HandsMen Customer_c	Name (Record Name), Email (Email), Phone (Phone), Loyalty_Status__c (Picklist: Bronze, Gold, Silver) Total_Purchases__c (Number)

HandsMen Product_c	Name (Record Name), SKU (Text), Price (Currency), Stock_Quantity__c (Number)
HandsMen Order_c	Order_Number (Record Name), Status (Picklist: Pending, Confirmed, Rejection), Quantity__c (Number), Total_Amount__c(Number)
Inventory_c	Auto Number (Record Name), Warehouse (Text), Stock_Quantity__c (Number)
Marketing Management_c	Campaign_Name (Record Name), Start_Date (Date), End_Date (Date)



Creation of Lookup Relationship

We established some lookup relationships between the custom objects to have a dependency between them. The lookup relationship are given bellow:

1. Creating lookup relationship between Marketing Campaign and HandsMen Customers.

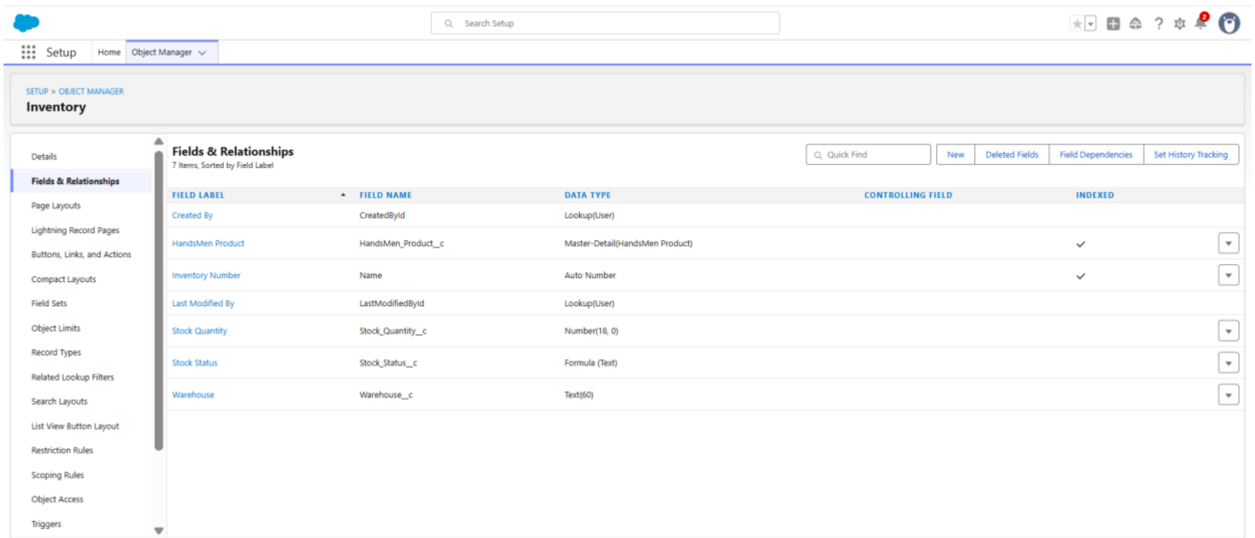
2. Creating lookup relationship between HandsMen order and HandsMen Product.

This relationship was created to automatically update the changes in the Product object when the customer confirms the order.

3. Creating lookup relationship between HandsMen Order and HandsMen Customers.

Creation of Master-Detail Relationship

We implemented a master-detail relationship between the custom objects to automatically synchronize changes between them upon order confirmation. Specifically, this relationship connects Inventory and HandsMen Product, allowing the system to accurately link and store inventory data to its respective product record.



Creation of Formula Fields

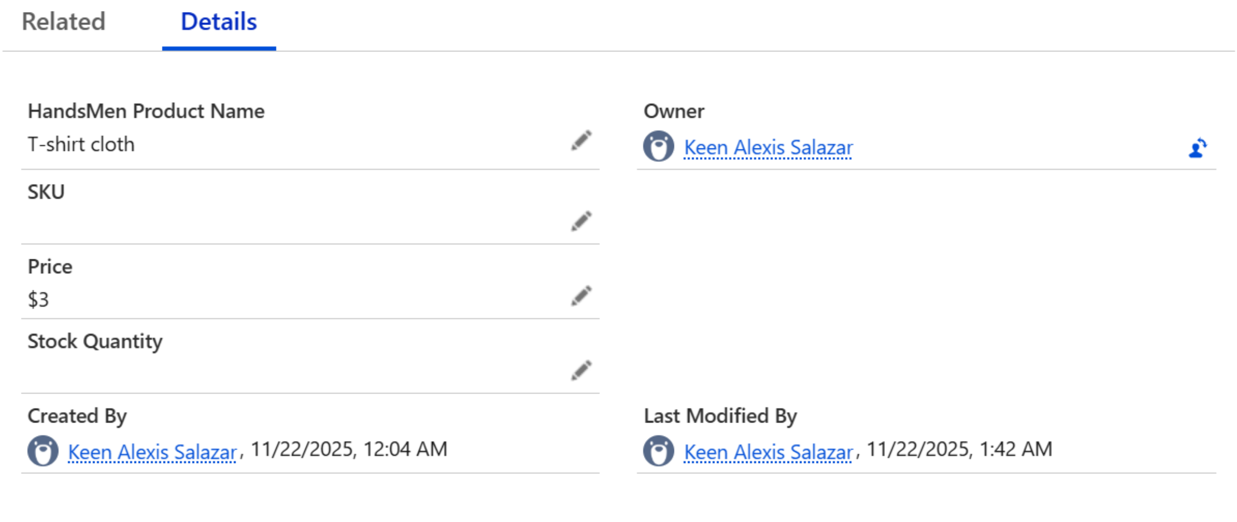
Formula fields were incorporated into the custom objects to enhance real-time data visibility and eliminate the need for manual calculations. These fields automatically compute and display values based on predefined logic, ensuring greater clarity and automation across all business processes.

1. Inventory__c → Stock_Status__c

To provide a quick visual status of item availability without requiring users to check the numeric stock count manually.

Formula:

Stock_Quantity__c <= 0

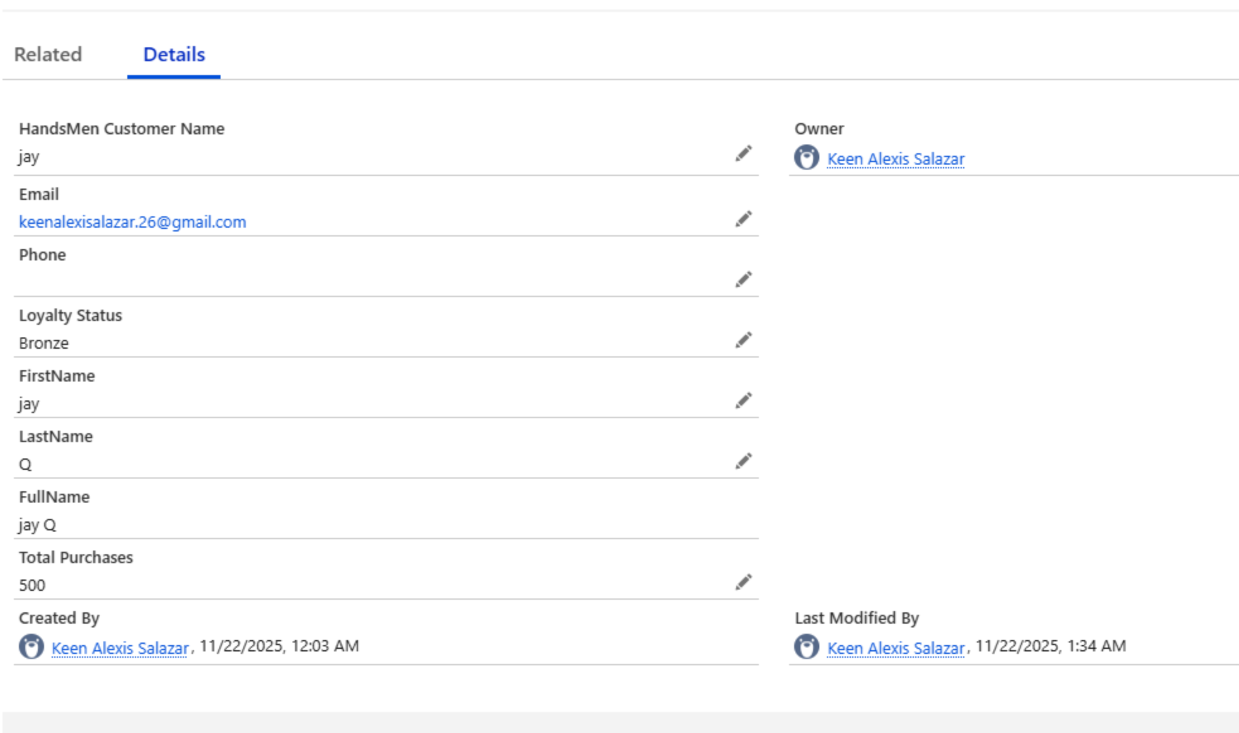


2. HandsMen_Customer__c → Full_Name__c

To combine a customer's first and last names into one clean, display-friendly field.

Formula:

FirstName__c + " " + LastName__c



Data Configuration – Validation Rules.

To guarantee data accuracy, enforce business logic, and proactively prevent errors, multiple Validation Rules were deployed across the custom objects within the HandsMen Threads CRM. These rules are crucial for maintaining clean, reliable data and ensuring that incorrect or illogical records are not processed.1. HandsMen_Order__c → Total_Amount__c

Validation Rule: **"Total Amount"**

Total_Amount__c <= 0

Purpose: Prevents order entries with zero or negative total amounts.

2. Inventory__c → Stock_Quantity__c

Validation Rule: **"Stock Quantity"**

Stock_Quantity__c <= 0

Purpose: Blocks the saving of inventory records where the stock count is zero or negative.

3. HandsMen_Customer__c → Email

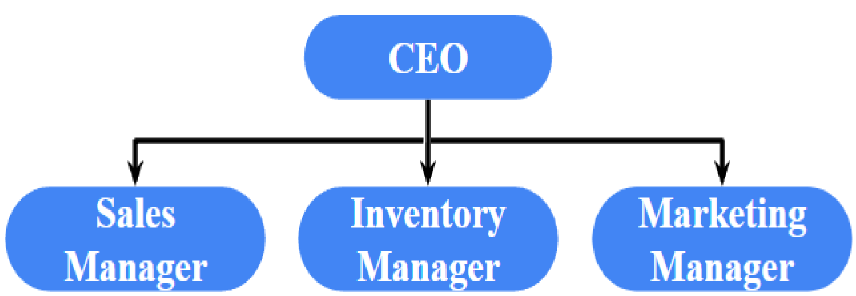
Validation Rule: **"Email"**

NOT CONTAINS (Email, "@gmail.com")

Purpose: Restricts entries to only Gmail addresses (or blocks non-Gmail ones, depending on the requirement).

Data Security – Roles

A clear role hierarchy was implemented in the HandsMen Threads CRM to precisely define data visibility, access controls, and the organizational reporting structure. This structure was designed to replicate the company's real-world operations, enabling efficient team collaboration while strictly maintaining data segregation and proper oversight.



Structure Implemented

At the top of the hierarchy:

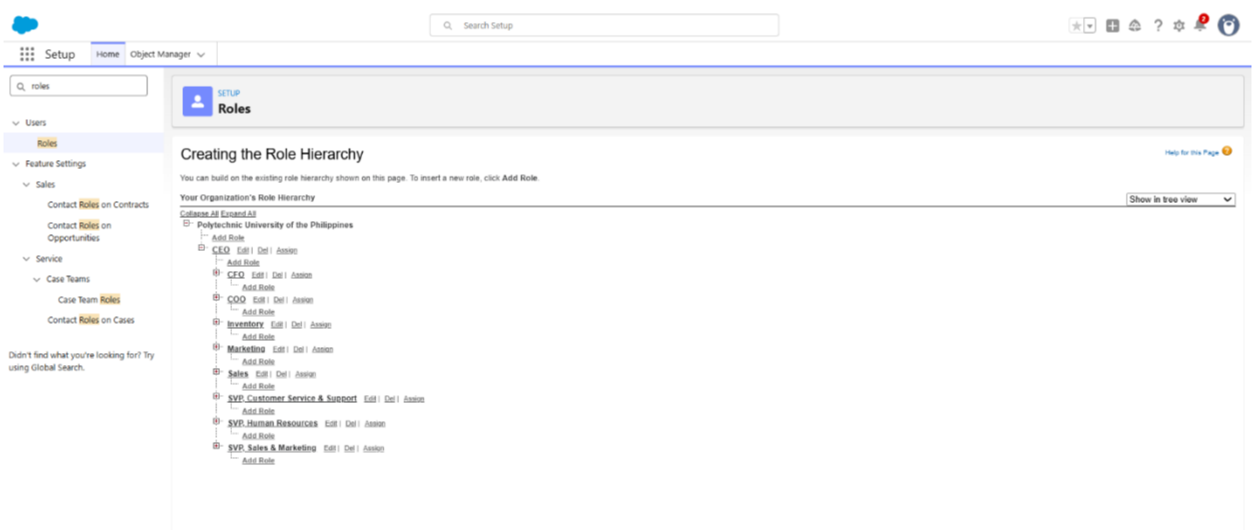
 CEO (Chief Executive Officer) – Has visibility into all records and activities across departments.

Reporting directly to the CEO:

1. **Sales Manager (Label: Sales)** Responsible for overseeing customer orders, revenue reports, and loyalty programs.

2. **Marketing Manager (Label: Marketing)** Handles customer engagement, loyalty campaigns, and customer communications.

3. **Inventory Manager (Label: Inventory)** Oversees stock levels, restocking workflows, and warehouse notifications.



Data Security – Users

To assign the respective roles we created some demo users that were asked to create in the user story.

First user:

Name: Niklaus Mikaelson

Role: Sales

User Licence: Salesforce Platform

Profile: Platform 1

Second user:

Name: Kol Mikaelson

Role: Inventory

User Licence: Salesforce Platform

Profile: Platform 1

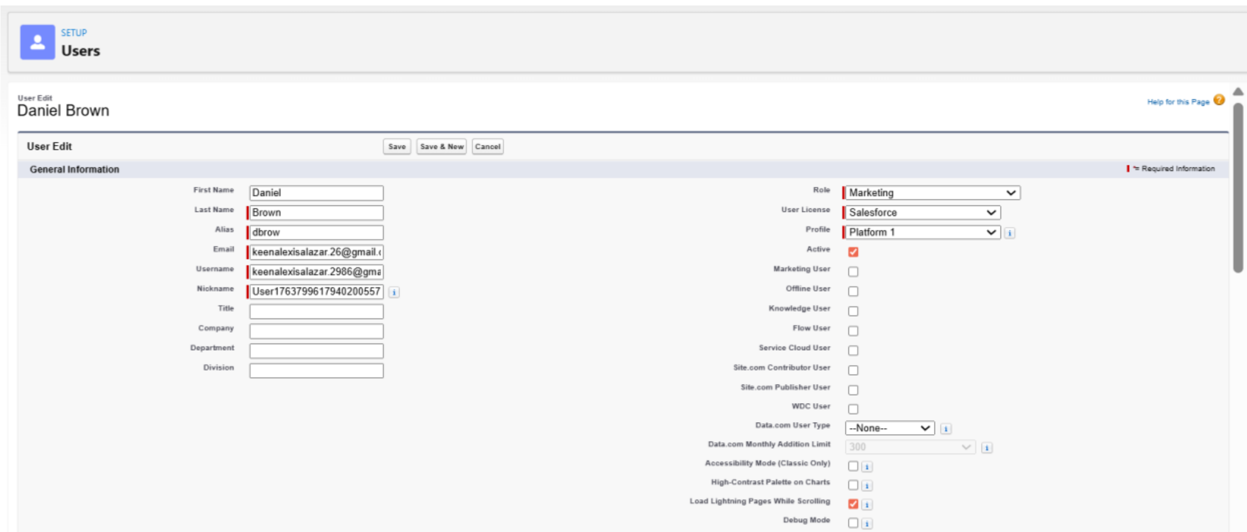
Third user:

Name: Daniel Brown

Role: Marketing

User Licence: Salesforce Platform

Profile: Platform 1



Data Security – Permission Sets

To ensure flexible and granular control over data access within the HandsMen Threads CRM, Permission Sets were strategically employed to define object-level permissions for various departmental roles. This methodology eliminated the administrative complexity of managing numerous custom profiles, ensuring each role has secure, role-specific access granted only through its dedicated permission set.

I created 3 permission sets for various reasons, the three permission sets are below:

Sales Permission Sets

Full access (Read, Create, Edit, Delete) to Customer**s** and Orders.

Inventory Permission Sets

Read and Edit access on Inventory and HandsMen Products via a custom permission sets.

Marketing Permission Sets

Read access on Customers and Edit access on Marketing Campaigns.

Email Template

Here are a few ways to paraphrase the text:

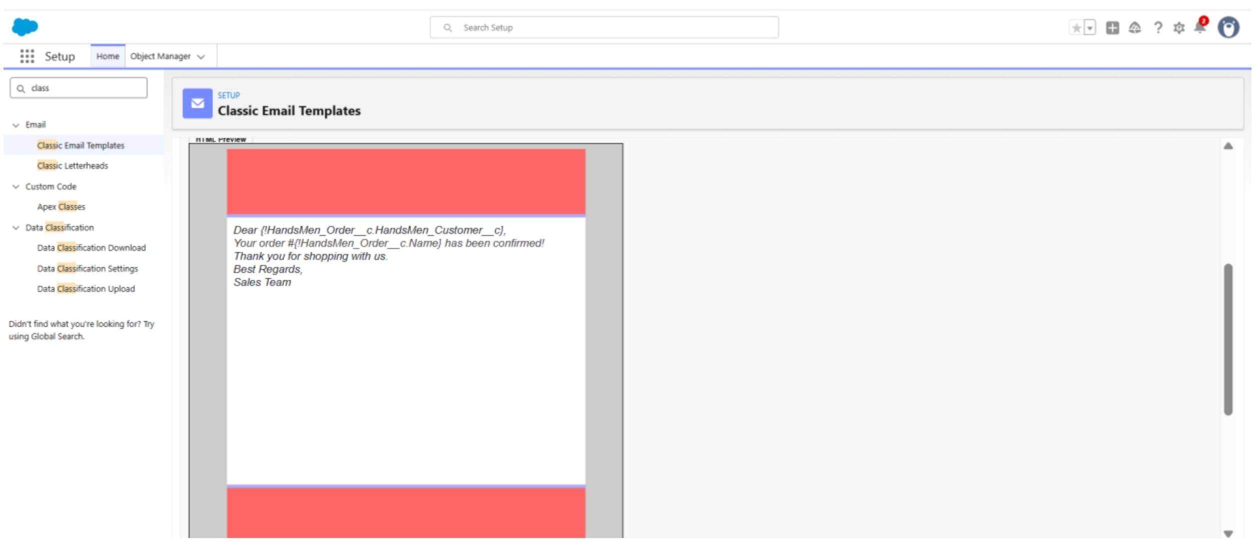
Option 1: Emphasizing Customer Experience and Automation Tool

A key automated feature in the HandsMen Threads CRM is the instant dispatch of order confirmation emails, which significantly improves the user experience and fosters brand trust. This functionality was achieved by designing the email template using Salesforce's Classic Email Templates and integrating it with an automated process, such as a Flow or Process Builder, for immediate triggering.

We created 3 major email templates for our usage:

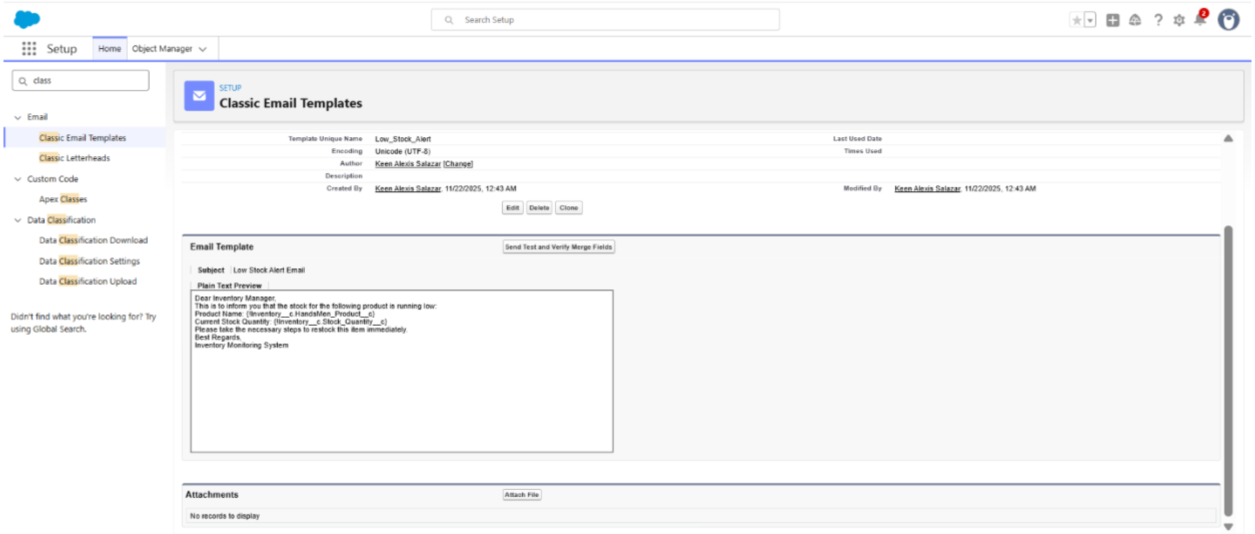
1 . **Order Confirmation Email**

The Order Confirmation Email Template, built using Salesforce's Classic Email Templates, was integrated with an automation tool (such as a Workflow Rule, Process Builder, or Flow) to ensure the email is automatically sent whenever a new record is created in the HandsMen Order__c object.



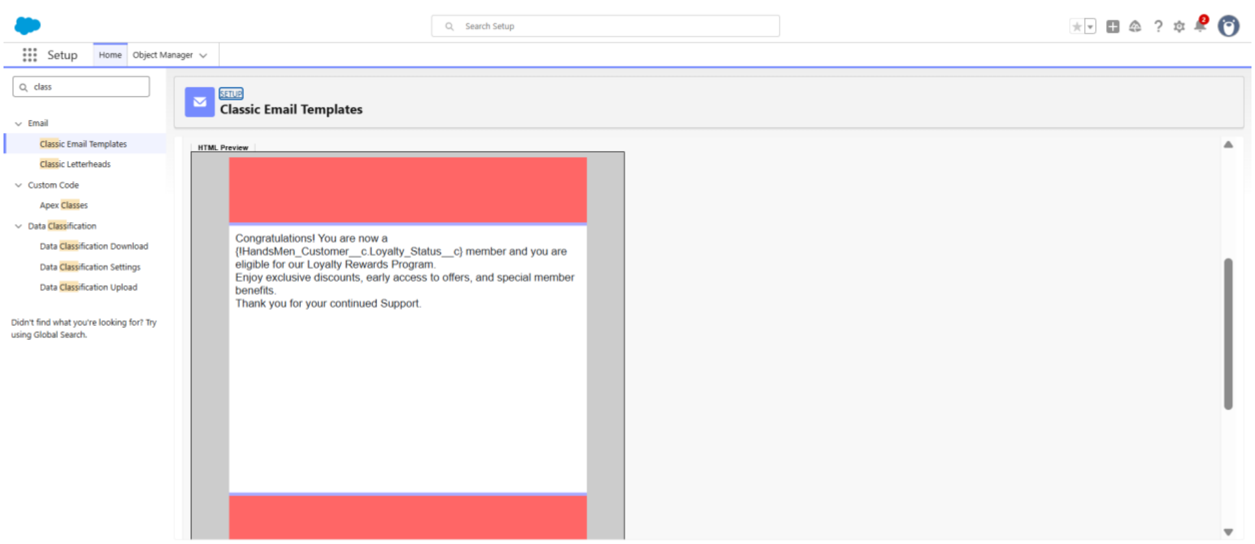
2 . Low Stock Alert:

To support seamless inventory operations in the HandsMen Threads CRM, we implemented a Low Stock Alert Email system. This system proactively notifies inventory managers when stock dips below a certain limit, thereby facilitating timely replenishment and avoiding any delays in fulfilling customer orders.



3 . Loyalty Program Email:

As a core component of HandsMen Threads' customer retention strategy, a Loyalty Program Email System was deployed in the Salesforce CRM. This automated system recognizes and rewards loyal customers by automatically tiering them into Gold, Silver, or Bronze based on their purchase history, and then sends personalized emails notifying them of their current status.



Email Alerts

The automation layer of the HandsMen Threads CRM utilizes Email Alerts (built upon the existing email templates) to ensure timely, real-time communication with the correct recipients. These alerts are critical for notifying users and customers about major events like order confirmations, low stock warnings, and changes in loyalty program status.

Flows

Salesforce Flows were implemented to automate crucial business processes and reduce manual intervention. By enabling actions to be triggered based on specific conditions, these flows ensure that customers and users receive necessary updates and alerts in a timely manner.

I created 3 flows out of which 2 were record triggered flows and one was schedule triggered flow.

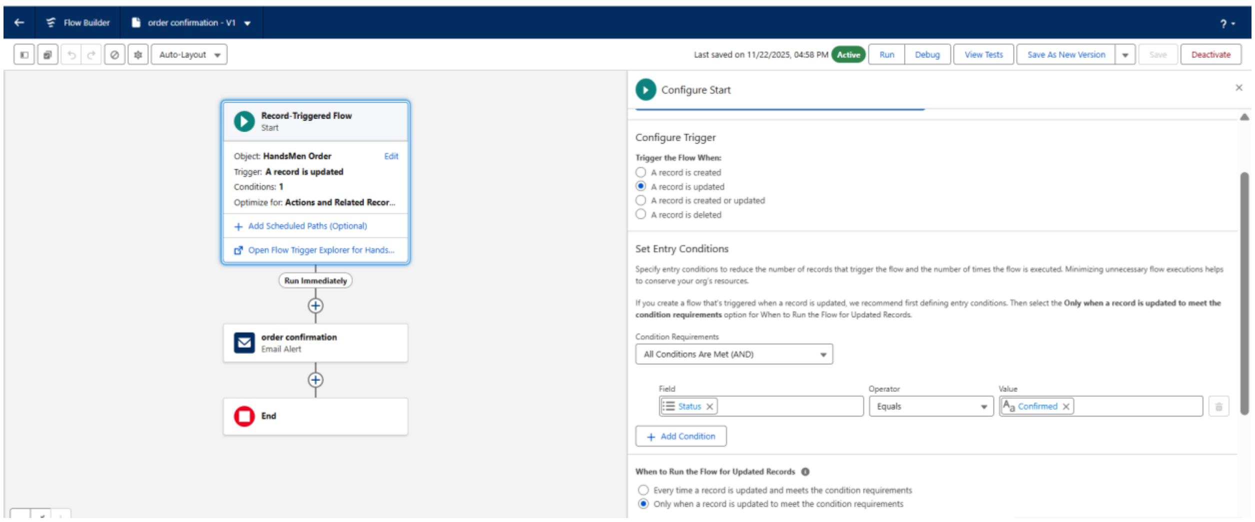
1 . Order Confirmation Flow

? **Type:** Record-Triggered Flow

? **Purpose:** Sends an automated confirmation email when an order is marked as “Confirmed”.

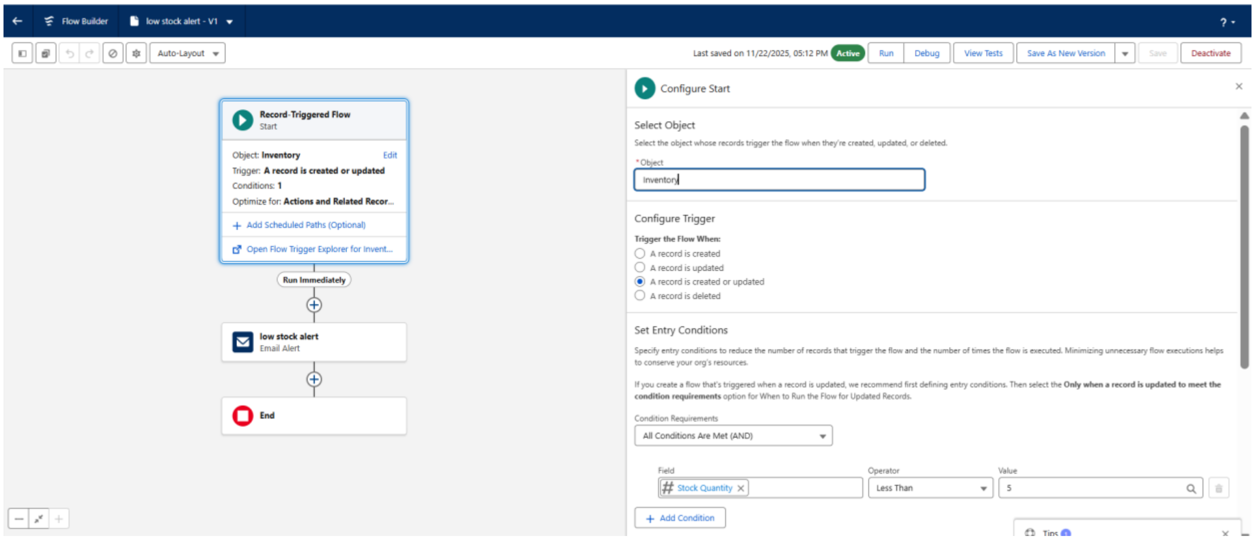
? **Business Impact:** Enhances customer experience by acknowledging orders

instantly and sending them an email alert saying your order has been confirmed.



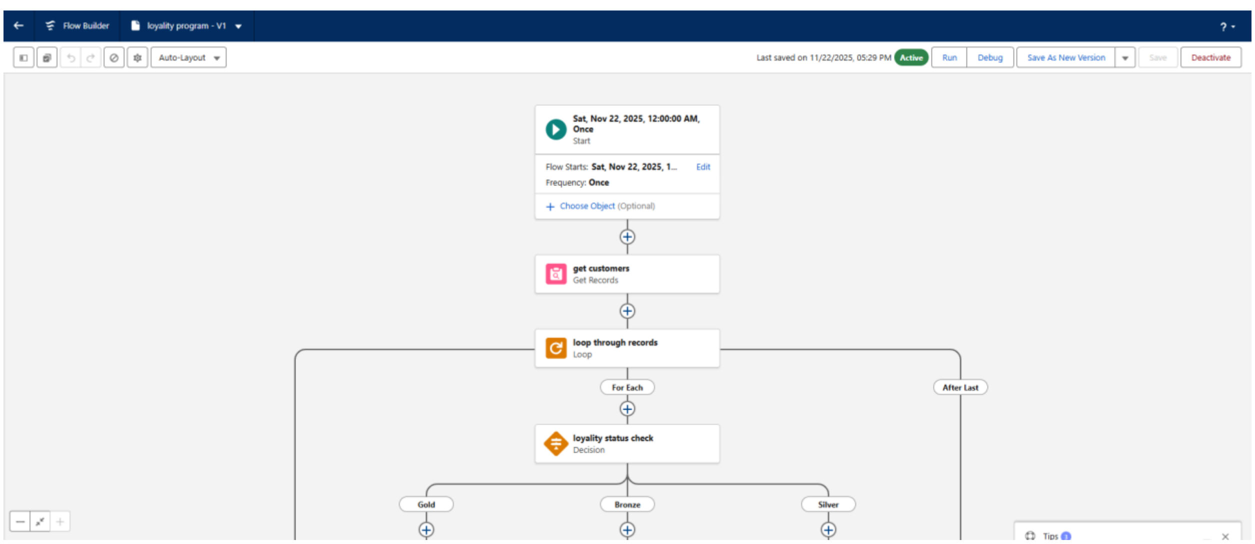
2 . Stock Alert Flow

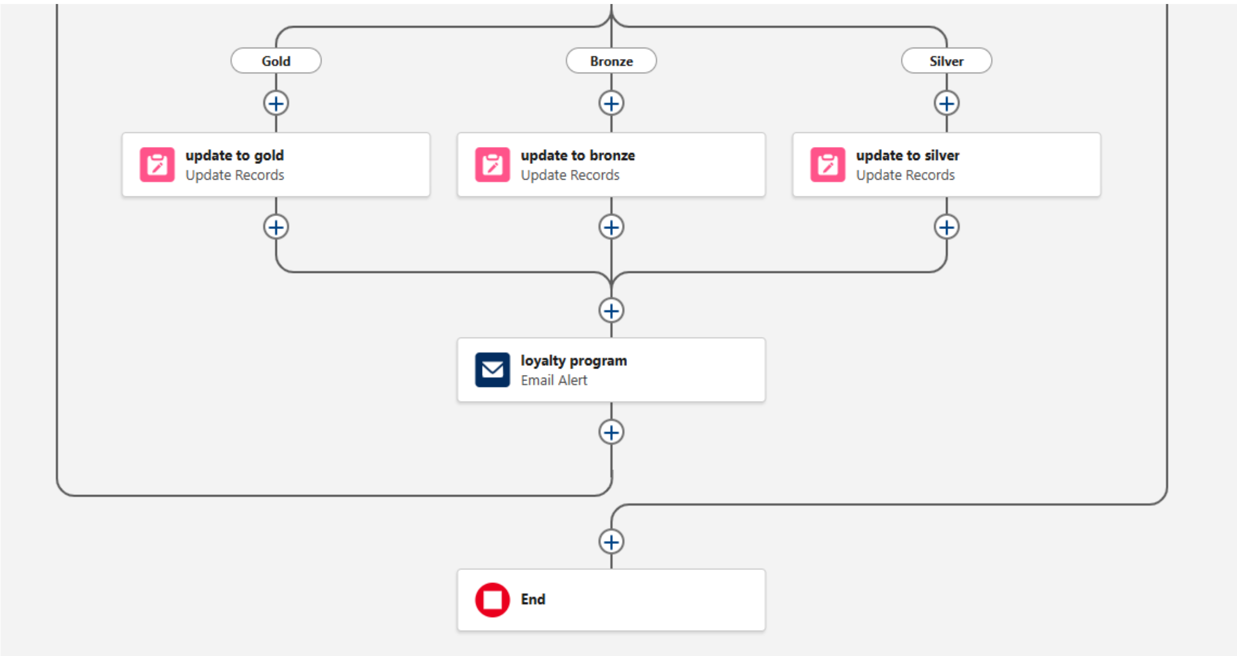
- **Type:** Record-Triggered Flow
- **Purpose:** Triggers a low stock alert when inventory drops below the critical threshold (less than 5 units).
- **Business Impact:** This system provides immediate notification to the inventory team, allowing them to take prompt replenishment action and prevent stockouts. Additionally, it instantly sends an email alert to the inventory or company owner, urging them to refill the stock before it is depleted.



3 . Loyalty Status Flow

- **Type:** Scheduled Flow
- **Purpose:** Evaluates the total purchase amount of customers and updates their loyalty tier (Gold, Silver, or Bronze).
- **Business Impact:** This system maintains up-to-date customer segmentation for highly targeted marketing efforts and personalized communication. It automatically categorizes customers into their respective loyalty domains and sends personalized congratulatory emails to notify them of their status.





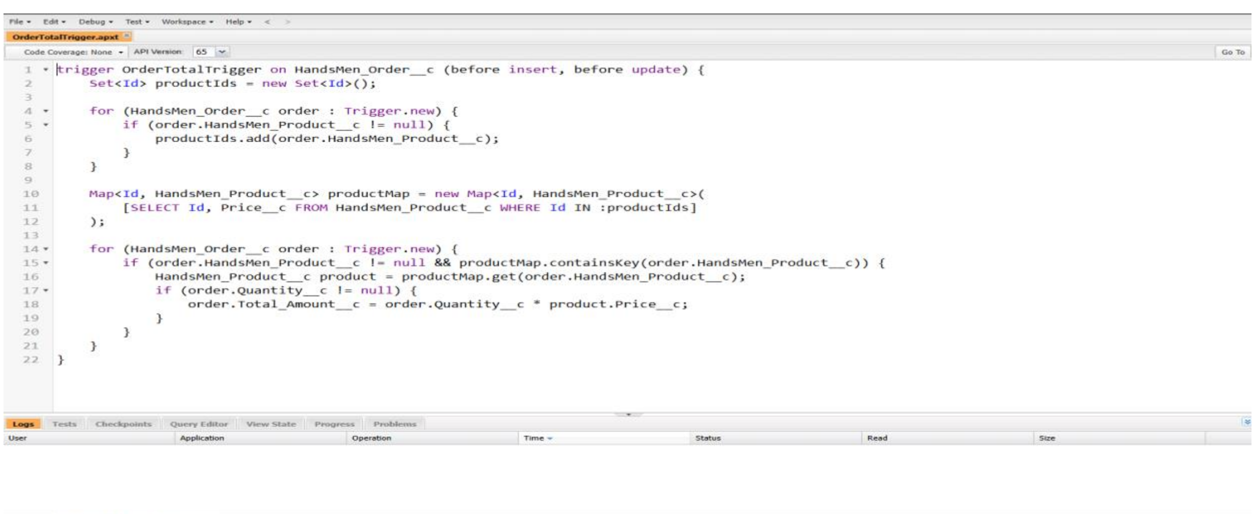
Automation using Apex

To guarantee efficient backend processing and data consistency, we deployed Apex Triggers on the custom objects. These triggers are essential for automating critical business logic at the data layer without needing any manual user input.

Two Apex triggers were created for 2 different automations.

1 . Update Order Total Trigger

- **Object:** HandsMen_Order__c
- **Purpose:** The Total_Amount__c field is automatically calculated and updated based on the line items or products included in the order.




```
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;
            }
        }
    }
}
```

2 . Stock Deduction Trigger

 Object: Inventory__c

 Purpose: Orders are monitored, and the system instantaneously deducts the correct quantity from the relevant inventory record upon order placement.

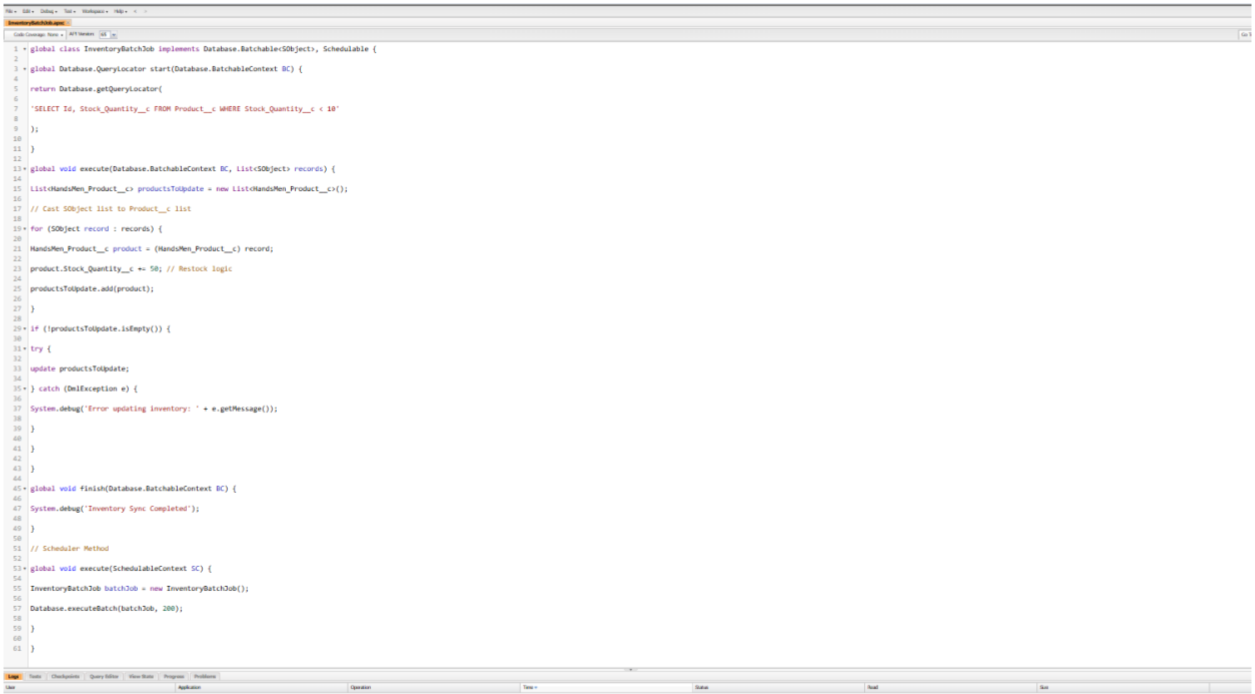
```
trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
    Set<Id> productIds = new Set<Id>();

    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, HandsMen_Product__c
         FROM Inventory__c
         WHERE HandsMen_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.HandsMen_Product__c == order.HandsMen_Product__c) {
                    inv.Stock_Quantity__c -= order.Quantity__c;
                    inventoriesToUpdate.add(inv);
                    break;
                }
            }
        }
    }

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

NOTE: Did not create the third and last Trigger because I already achieved that with the help of flows so it was totally useless for creating one.

Batch Jobs – Creating Batch Apex

Batch Apex Jobs were implemented to efficiently manage bulk data operations and backend processing. By executing asynchronously, these jobs are ideal for any time-based or large-scale data handling tasks. Loyalty Points Calculation

- **Purpose:** Loyalty points are automatically updated weekly following an evaluation of each customer's purchase history.
- **Impact:** The system accurately supports the loyalty program by rewarding customers in the Gold, Silver, and Bronze categories.



```
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);
}
if (!productsToUpdate.isEmpty()) {
try {
update productsToUpdate;
} catch (DmlException e) {
System.debug('Error updating inventory: ' + e.getMessage());
}
}
}
global void finish(Database.BatchableContext BC) {
System.debug('Inventory Sync Completed');
}
// Scheduler Method
global void execute(SchedulableContext SC) {
InventoryBatchJob batchJob = new InventoryBatchJob();
Database.executeBatch(batchJob, 200);
}
}
```

Project Explanation with real world Example.

Let’s walk thorough it like a real world customer interaction.

1. Customer Registration

- A customer, Josh Guitarist, visit the store or website.
- In Salesforce: A record is created in the Customer object with his name, phone, email etc.
- A Validation Rule ensures the email format is correct - for example it must contain @gmail.com.

2. Product Setup

- The admin adds products (eg. shirts, jeans) into HandsMen_Product__c object.
- Each product includes pricing and other details.
- Inventory is also created to manage available stock for these products.

3. Order Placement

- John places an order for 2 shirts priced at ₹500 each.
- In Salesforce, a new record is created in the Order object.
- An Apex Trigger calculates the Total_Amount__c = 2 × 500 = ₹1000 automatically.

4. Inventory Update

As soon as the order is placed:

- **Apex Trigger on Inventory:** Reduces shirt stock by 2.
- **Validation Rule:** Ensures stock never goes below 0.

5. Loyalty Program

- John now has a total purchase of ₹1000.
- A trigger on Customer checks his total purchases. Based on the value: < ₹500 → Bronze ₹500–₹1000 → Silver ₹1000 → Gold
- So, John becomes a Silver member.

6. Email Notifications

- When a new order is placed or loyalty status is updated: Flow + Email Alert is triggered.
- John gets an email: “Thanks for your purchase! Your loyalty status is now Silver.”

Screenshots

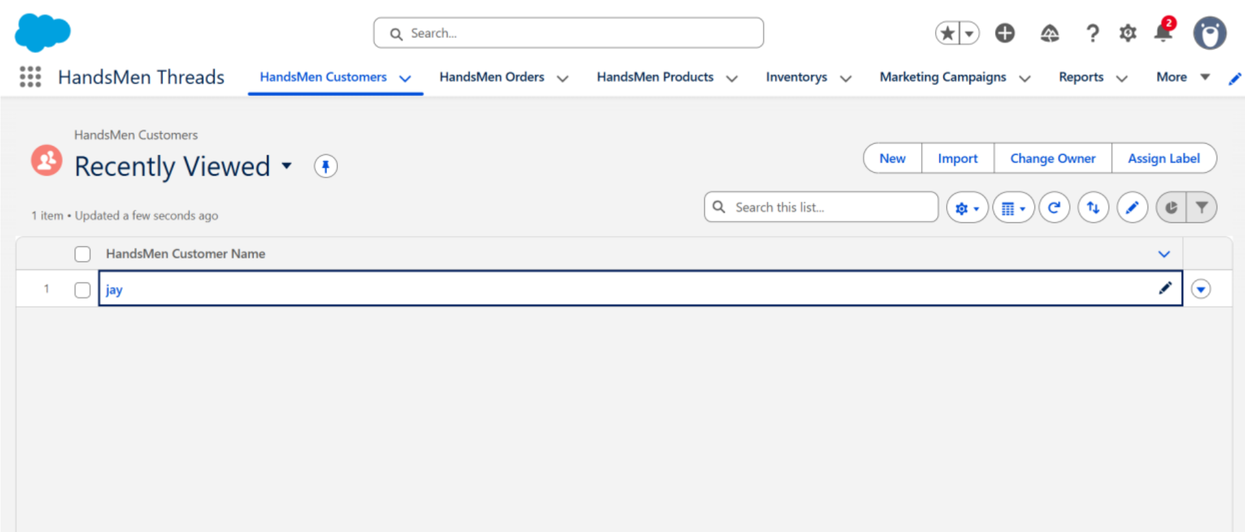


Fig. Custom App for HandsMen Threads

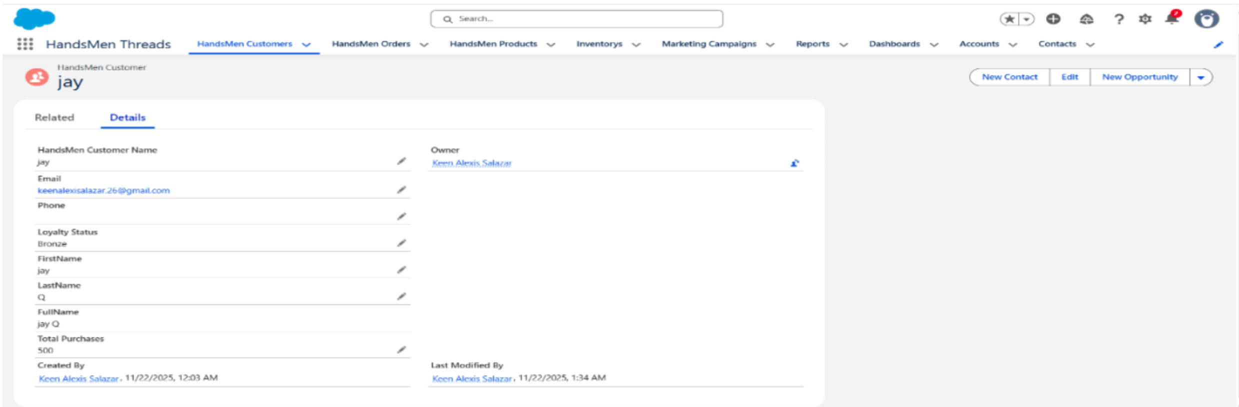


Fig. Customer creation

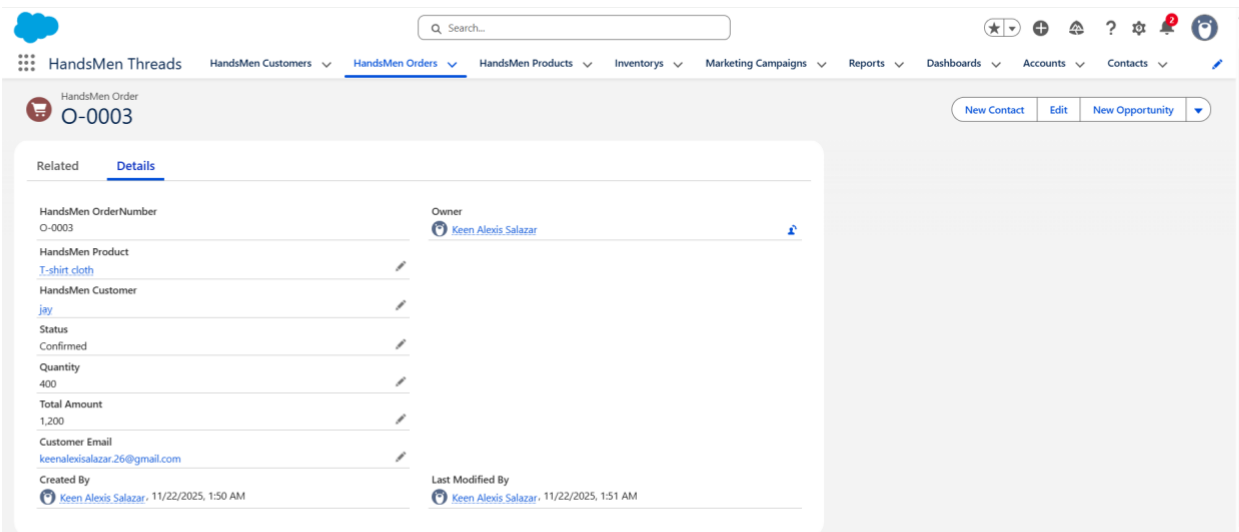


Fig. Order confirmation of the customer

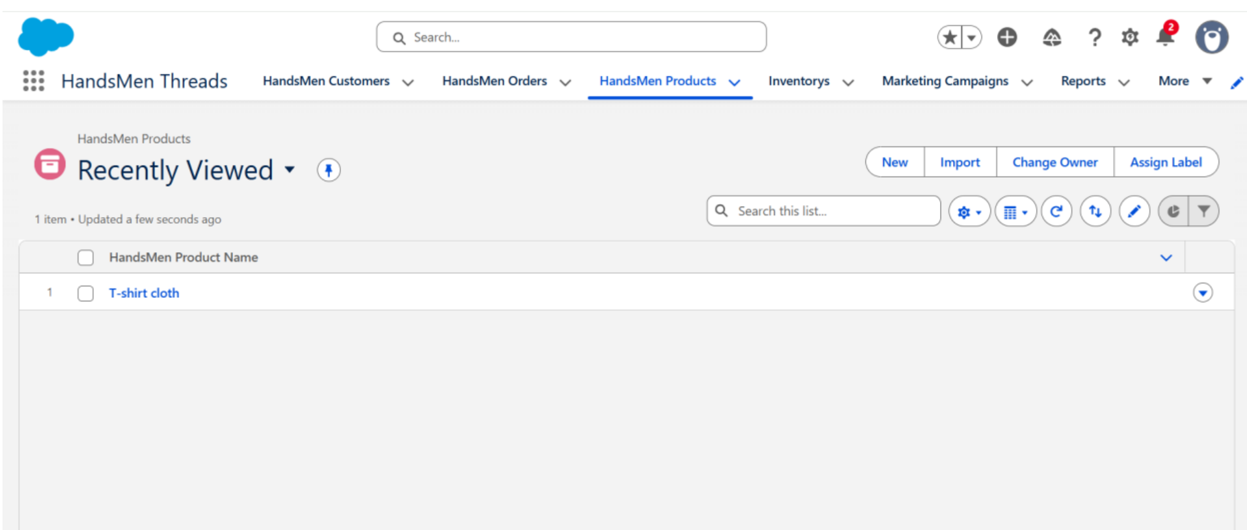


Fig. Collection of available product's in our store

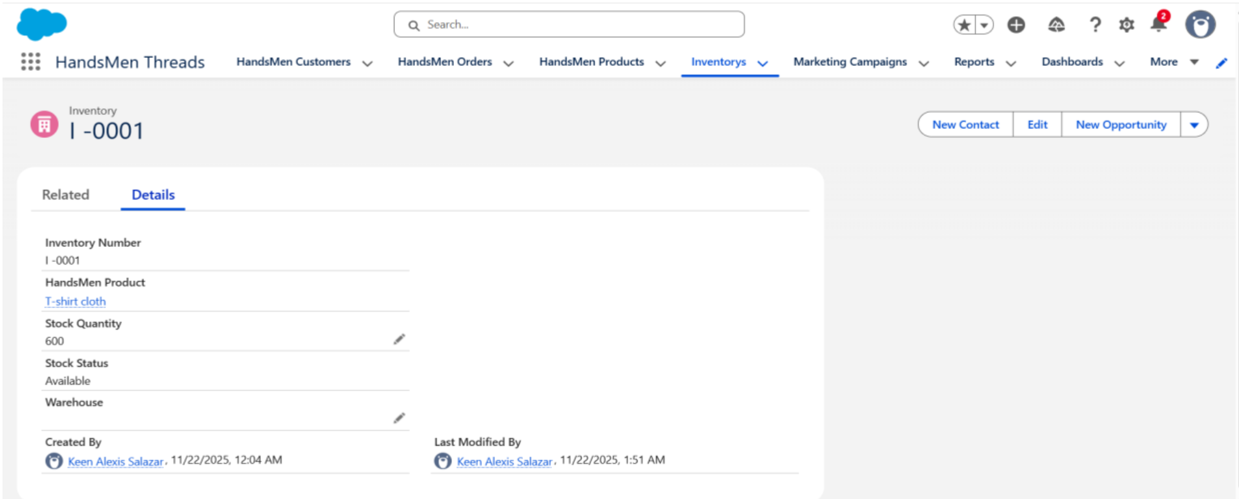


Fig. Inventory of HandsMen Threads

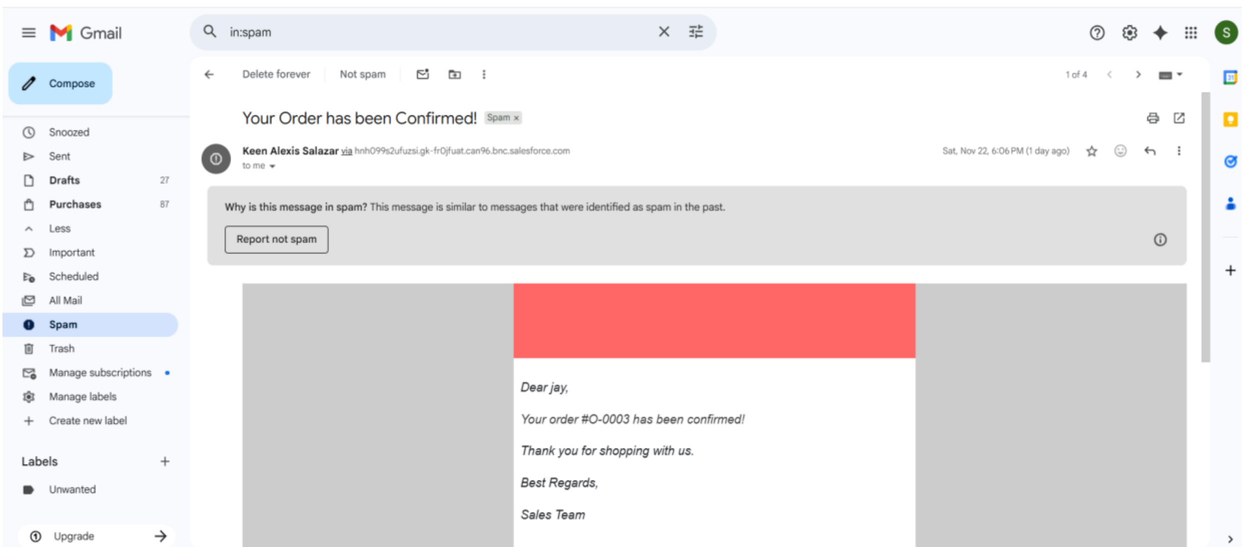


Fig. Order Confirmation Email

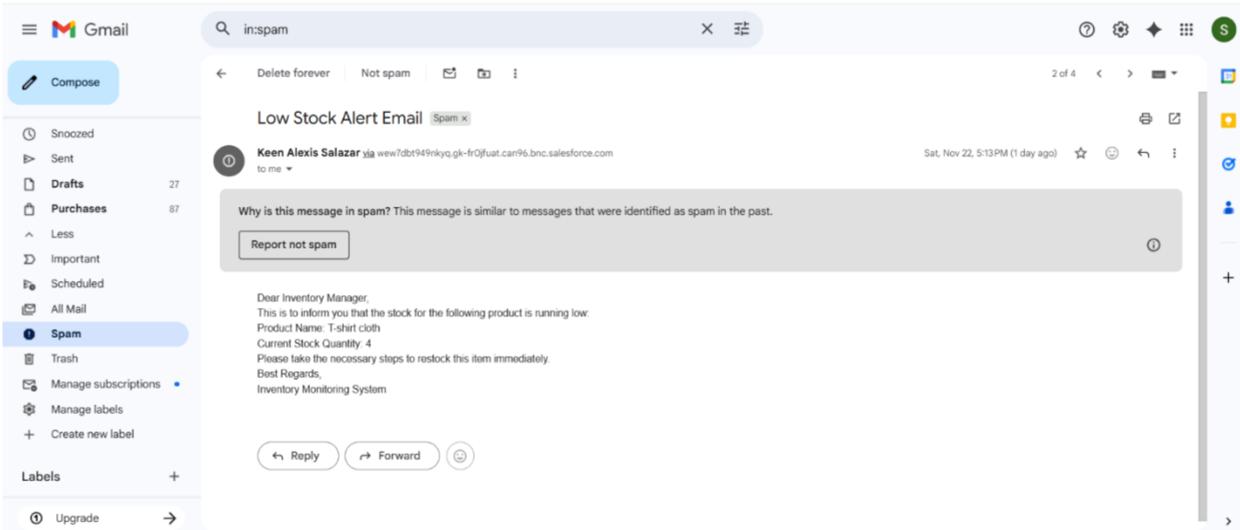


Fig. Low Stock Alert Email

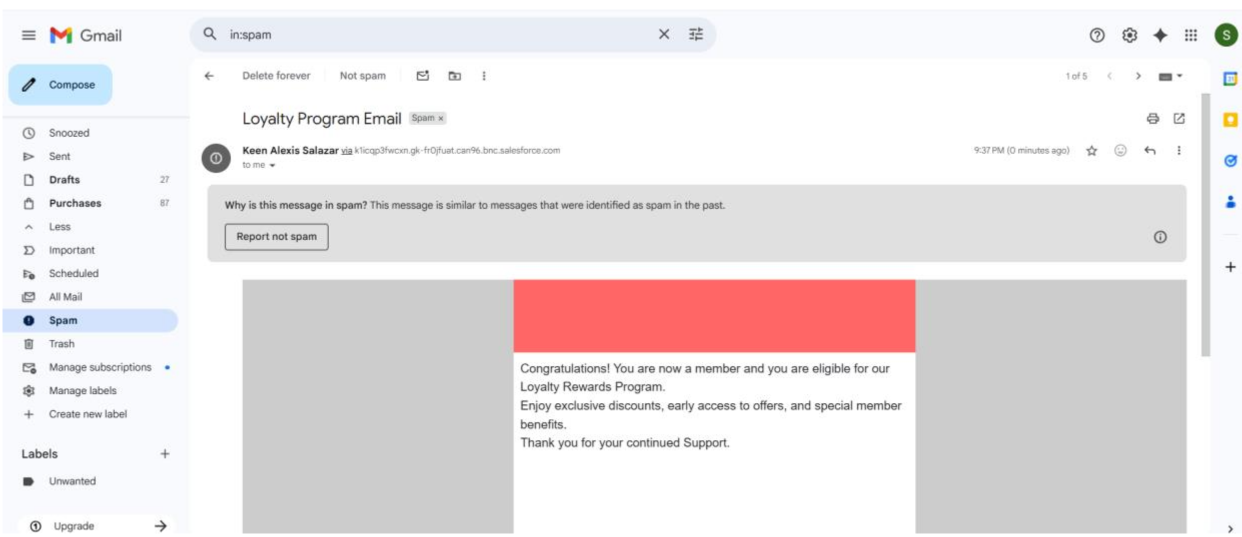


Fig. Loyalty Program Email

Conclusion

The resulting Salesforce CRM for HandsMen Threads is an integrated digital solution that comprehensively manages all customer and operational aspects (sales, inventory, marketing, loyalty). This system features a robust data model, intelligent automation, secure role-based access, and efficient workflows, perfectly suited for the needs of a high-speed retail brand.

Future Scope and Enhancements

1. Integration with Payment Gateways

- Payment status updates and refund workflows are automated by the system.

2. Customer Portal

- The system grants customers login access for order tracking, profile management, and viewing their loyalty status.

3. AI-Powered Recommendations

- Einstein AI is used for product suggestions and the analysis of purchase patterns.

4. **Mobile-Responsive Experience**

- Develop a Salesforce Mobile App for field sales and warehouse managers.

5. Enhanced Reporting

- Dashboards for revenue, stock aging, customer churn, and campaign ROI.

6. **Multi-Warehouse Support**

- Expand Inventory Sync to multiple warehouse locations and manage regional

stock levels.

7. Customer Feedback Automation

- After order delivery, auto-trigger surveys for experience rating and product feedback