HandsMen Threads:

Elevating the Art of Sophistication in Men's Fashion.

By:

K.Abhijith Reddy

Abstract:

The capstone project titled *"HandsMen Threads: Elevating the Art of Sophistication in Men's Fashion"* is developed as part of the Salesforce SmartBridge 2025 program. This project showcases the practical implementation of Salesforce's automation and customization features to streamline operations in a fictional men's fashion business.

The project revolves around the setup and configuration of a Salesforce Developer Edition Org. Five custom objects—**Handsman Customer**, **Handsman Product**, **Handsman Order**, **Inventory**, and **Marketing Campaign**—form the backbone of the system, each integrated into a custom Lightning App named *"Handsman Threads."* Lookup and master-detail relationships are strategically implemented to ensure efficient data connections and integrity.

Business automation is achieved through a combination of **validation rules, formula fields, email alerts, flows, and Apex triggers**. Notable automation includes real-time stock deduction from inventory upon order confirmation, dynamic calculation of order totals based on product price and quantity, and scheduled updates to customer loyalty status with corresponding email notifications. Validation rules enforce essential data quality, such as ensuring valid Gmail addresses and positive values for stock and order amounts.

A tailored data security model is built using custom **profiles, roles, and permission sets**, with specific access levels granted to users in Sales, Inventory, and Marketing roles. Email automation features include custom templates and alerts for order confirmations, low stock warnings, and loyalty program communications.

Advanced automation is further enhanced with **record-triggered flows**, **scheduled flows**, and **Apex batch jobs**, ensuring that the system not only runs efficiently but also scales with business needs.

This project demonstrates how Salesforce can be leveraged to automate retail workflows, improve customer engagement, and maintain operational efficiency in a fashion-centric business environment, providing a holistic solution aligned with modern digital business practices.

Objectives:

1. **To create a centralized system** using Salesforce Developer Edition that manages customers, products, orders, inventory, and marketing campaigns for a men's fashion business.

2. **To design and configure five custom objects** (Handsman Customer, Handsman Product, Handsman Order, Inventory, and Marketing Campaign) tailored to the operational needs of a fashion retail business.

3. **To implement data relationships** (Lookup and Master-Detail) among custom objects for accurate tracking of orders, stock levels, and customer interactions.

4. **To ensure data quality and integrity** by applying validation rules on fields such as stock quantity, order amount, and email format.

5. **To automate core business processes** using flows, email alerts, and Apex triggers—such as updating loyalty status, sending order confirmations, and deducting stock on order confirmation.

6. **To establish a secure data access structure** through custom profiles, roles, and permission sets based on user responsibilities (Sales, Inventory, Marketing).

7. **To enhance customer engagement** by automating personalized email communications like order confirmations, loyalty updates, and low stock alerts.

8. **To demonstrate the use of Apex programming** for advanced automation tasks including total amount calculation, inventory management, and batch processing for scalable operations.

Technical Description:

The *HandsMen Threads* project utilizes the Salesforce Platform to build a fully functional retail management solution for a fictional men's fashion brand. The project is implemented in a Salesforce Developer Edition Org and demonstrates proficiency in declarative tools, automation, and Apex programming.

Custom Objects and Schema Design:

- **Five custom objects** were created: `Handsman Customer`, `Handsman Product`, `Handsman Order`, `Inventory`, and `Marketing Campaign`.
- Objects are linked via **Lookup Relationships** (e.g., Handsman Order → Handsman Customer/Product) and a **Master-Detail Relationship** (Inventory → Product).
- Custom tabs and a **Lightning App** named *Handsman Threads* were created for centralized navigation and record management.

Field Customization and Validation:

- Key custom fields include picklists, currency, email, formula fields, and number types.
- **Validation rules** ensure proper data quality (e.g., preventing orders with zero amount, enforcing Gmail email format, disallowing zero inventory).

Process Automation using Flows and Email Alerts:

- **Record-Triggered Flows**:
  - **Order Confirmation Flow**: Sends an email when the order status is set to "confirmed."
  - **Low Stock Alert Flow**: Triggers an alert when stock quantity drops below 5.
- **Scheduled Flow**:
  - Evaluates customer purchase history and updates **Loyalty Status** (Gold, Silver, Bronze), sending loyalty emails accordingly.
- **Email Templates and Alerts**: Created using **Classic Letterhead** and linked with flows for real-time communication.

Apex Automation:

- **Order Trigger**: Automatically calculates `Total Amount` based on product price ×
  quantity.
- **Stock Trigger**: Reduces inventory on order confirmation.
- **Batch Class**: Implements a schedulable job to process inventory data for advanced
  scalability.

Security Model:

1. **Custom Profile**: `Platform One` with tailored object permissions.
2. **Roles** and **Permission Sets**: Defined for Sales, Inventory, and Marketing with field-
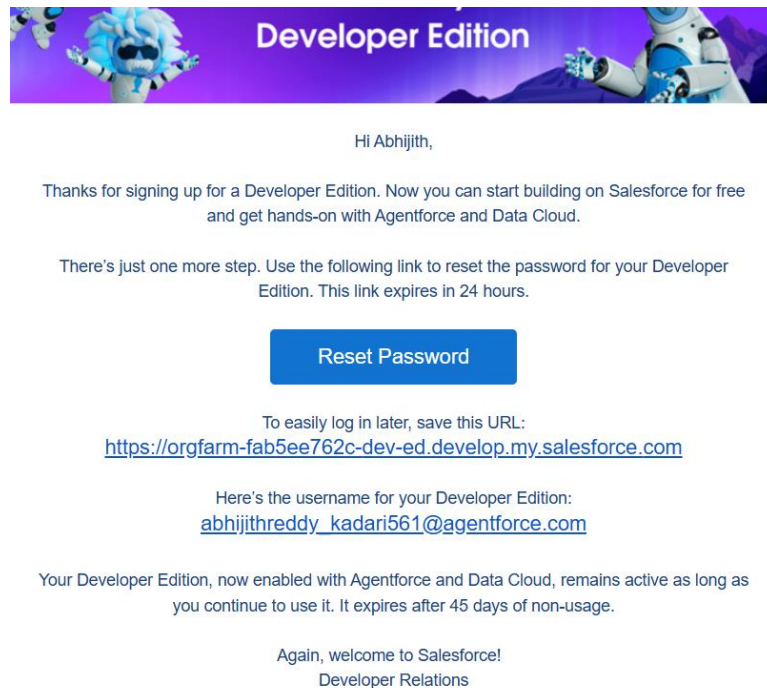   level access controls and user assignments.
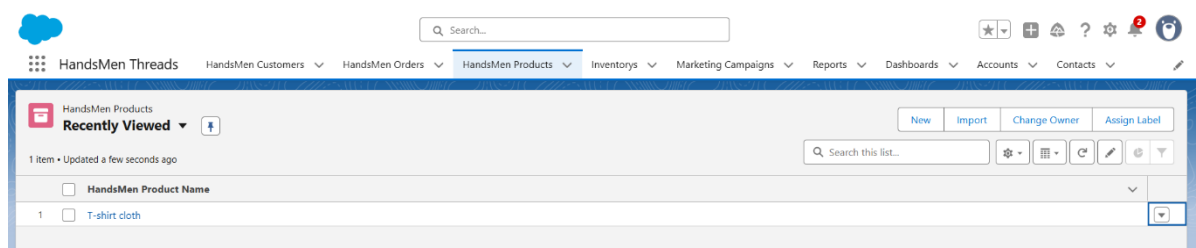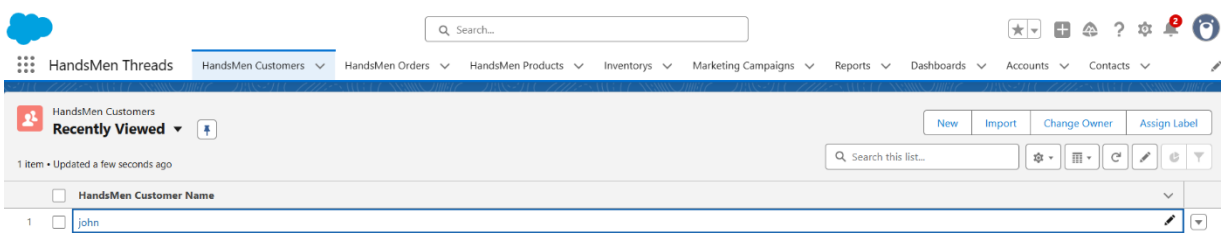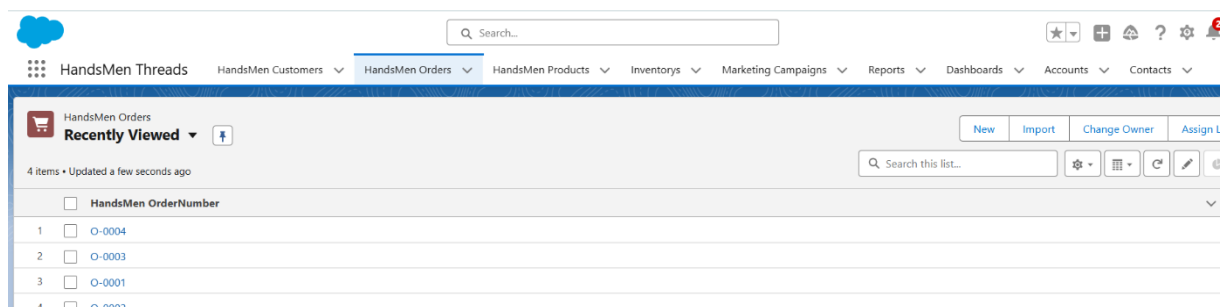
Project Execution:

1. Developer Org Setup

A Salesforce Developer Org was created using

https://developer.salesforce.com/signup.

The account was verified, password set, and access was granted to Salesforce Setup Page.

2. Custom Object Creation

Five custom objects were created to store business-critical data:
- Hands Men Customer Stores customer info like email, phone, loyalty status.
- HandsMen Product-Stores product catalog details like SKU, price, and stock.
- Hands Men Order-Stores orders placed by customers, including quantity and status.
- Inventory-Tracks stock quantity and warehouse location.
- Marketing Campaign Stores promotional campaigns and scheduling.
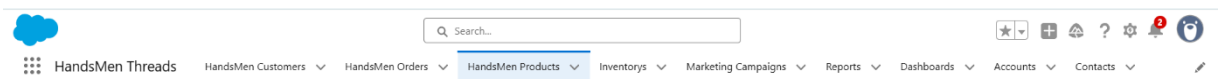
Steps followed:

- Navigated to Setup Object Manager Create Custom Object
- Provided label, name, and enabled reports/search
- Saved and created Tabs for each object

3. **Creating the Lightning App**
   - A custom Ligtning App named HandsMen Threads was created.

   - Included tabs: Hands Men Customer, Order, Product, Inventory, Campaign, Reports, etc.

   - Assigned to the System Administrator profile.

4. **Validation Rules**
   To ensure accurate data entry and enforce business logic, the following validation rules

   were applied:

   - Order Object: Prevents saving if Total Amount 0.
     Error: "Please Enter Correct Amount"
   - Customer Object: Validates email contains @gmail.com.
     Error: "Please fill Correct Gmail"

5. **User Role and Profile Setup**
   - Cloned the Standard User profile to a new profile named Platform1 and added access to necessary custom objects.
   - Created roles for different departments:
     Sales Manager, Inventory Management, Marketing Team.

6. **User Creation**
   Users were created in Salesforce and assigned appropriate roles and profiles to reflect their responsibilities:

   - Niklaus Mikaelson-Assigned the Sales role
   - Kol Mikaelson-Assigned the Inventory role
   - These role-based assignments help enforce proper data access and process control within the system.

7. **Email Template & Alerts**
   Created three email templates:
   - Order Confirmation Sent on order status Confirmed
   - Low Stock Alert-Sent when Inventory 5 units
   - Loyalty Program Email- Sent when loyalty status changes
   Corresponding Email Alerts were created using these templates and linked to automation flows.

OrgFarm EPIC <epic.orgfarm@salesforce.com>
to me ▾

Wed, Jul 23, 11:53 PM (19 hours ago) ☆ ☺ ↩ ⋮

Congratulations! You are now a Gold member and you are eligible for our Loyalty Rewards Program.
Enjoy exclusive discounts, early access to offers, and special member benefits.
Thank you for your continued Support.

Dear Inventory Manager,
This is to inform you that the stock for the following product is running low:
Product Name: Creta
Current Stock Quantity: 4
Please take the necessary steps to restock this item immediately.
Best Regards,
Inventory Monitoring System

8. Flows

a) Order Confirmation Flow
- Triggered when order is updated to Confirmed.
- Sends an Order Confirmation email to the related customer

b) Stock Alert Flow
- Triggered when Inventory stock drops below 5
- Sends Low Stock email to Inventory Manager

c) Scheduled Flow: Loyalty Update
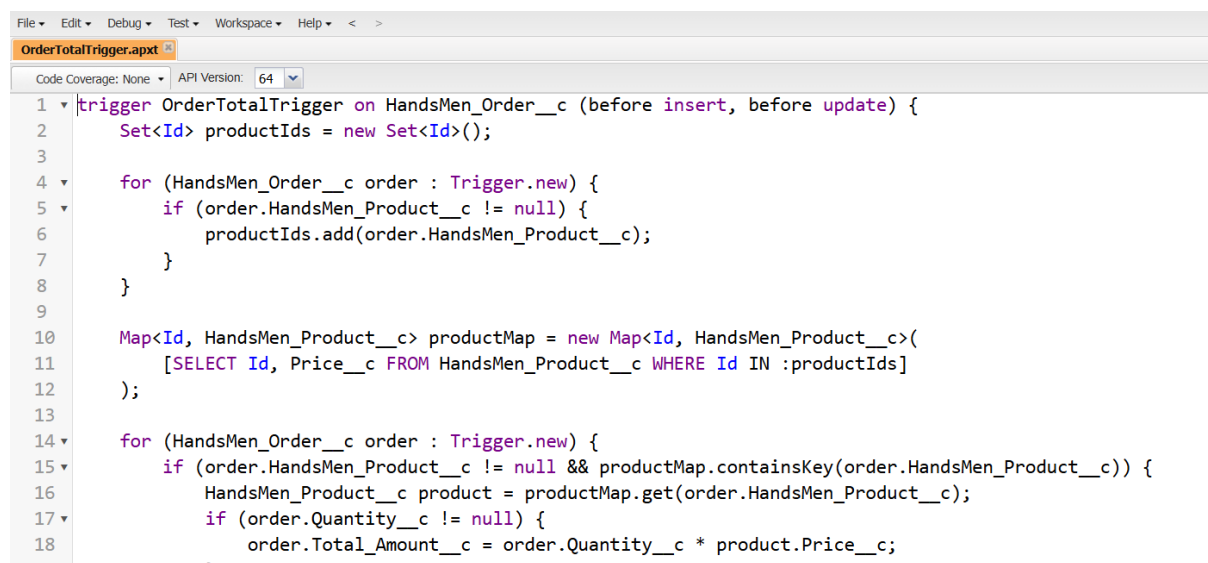- Runs daily at midnight
- Loops through customers and updates their Loyalty Status Based on total purchases
- Total Purchases>1000 → Gold
- Total Purchases<500 → Bronze
- Total Purchases>500 & <1000 → Silver(default)
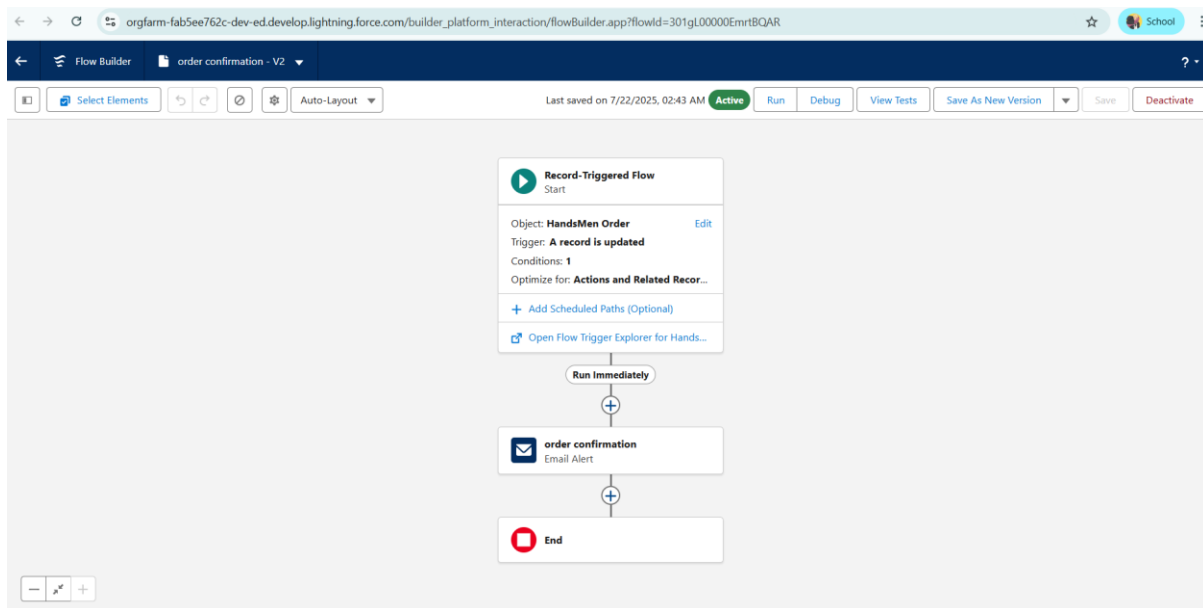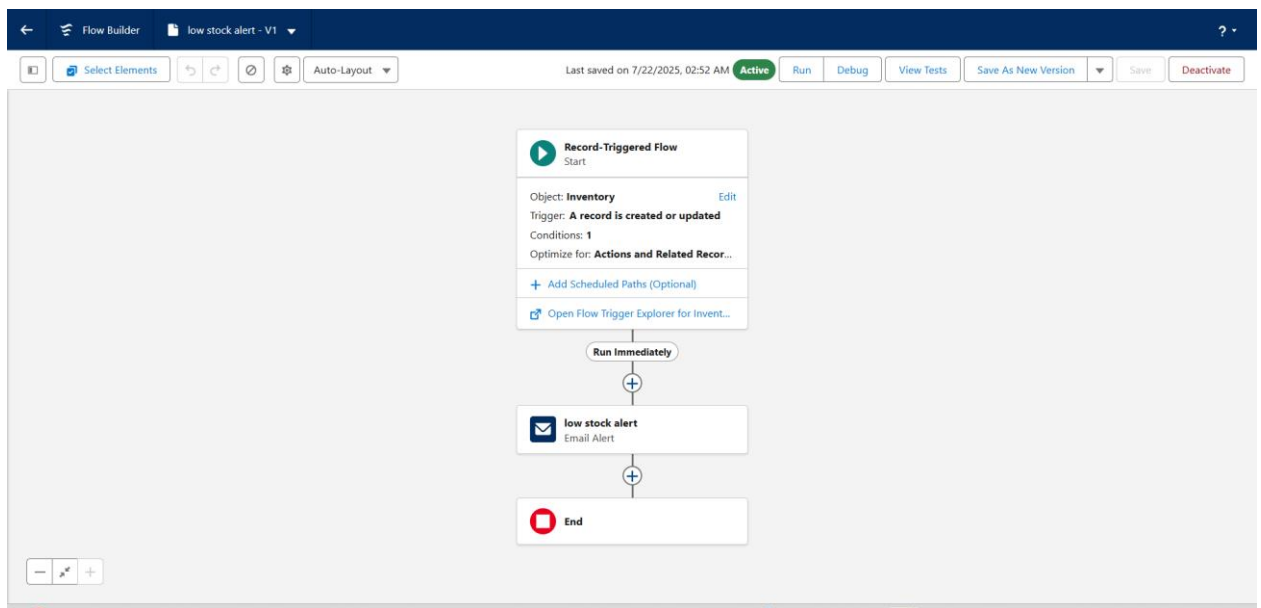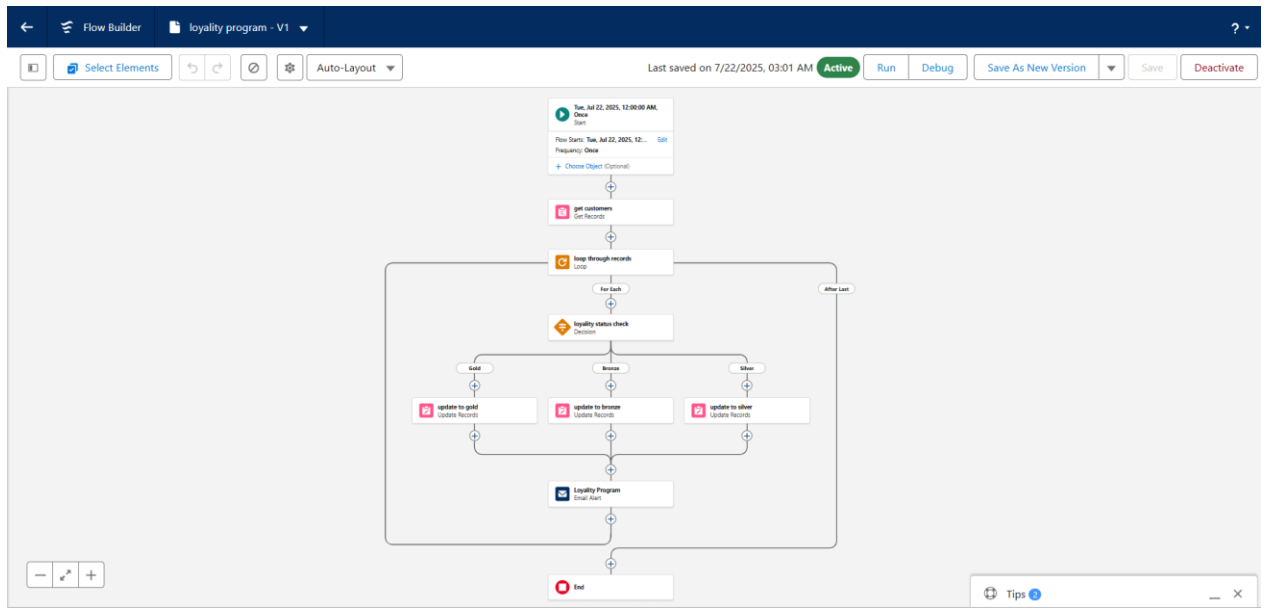
9. Apex Triggers

- **Order Total Trigger:** Auto-calculates Total Amount based on quantity and unit price.
- **Stock Deduction Trigger:** Reduces stock when an order is placed.
- **Loyalty Status Trigger:** Updates Loyalty Status based on total purchases.

File ▾   Edit ▾   Debug ▾   Test ▾   Workspace ▾   Help ▾   <   >

OrderTotalTrigger.apxt ✕

Code Coverage: None ▾   API Version: 64 ▾

```
 1 ▾ trigger OrderTotalTrigger on HandsMen_Order__c (before insert, before update) {
 2       Set<Id> productIds = new Set<Id>();
 3
 4 ▾     for (HandsMen_Order__c order : Trigger.new) {
 5 ▾         if (order.HandsMen_Product__c != null) {
 6               productIds.add(order.HandsMen_Product__c);
 7           }
 8       }
 9
10       Map<Id, HandsMen_Product__c> productMap = new Map<Id, HandsMen_Product__c>(
11           [SELECT Id, Price__c FROM HandsMen_Product__c WHERE Id IN :productIds]
12       );
13
14 ▾     for (HandsMen_Order__c order : Trigger.new) {
15 ▾         if (order.HandsMen_Product__c != null && productMap.containsKey(order.HandsMen_Product__c)) {
16               HandsMen_Product__c product = productMap.get(order.HandsMen_Product__c);
17 ▾             if (order.Quantity__c != null) {
18                   order.Total_Amount__c = order.Quantity__c * product.Price__c;
```

OrderTotalTrigger.apxt ⊠   **StockDeductionTrigger.apxt** ⊠

Code Coverage: None ▾ | API Version: 64 ▾

```
1   trigger StockDeductionTrigger on HandsMen_Order__c (after insert, after update) {
2       Set<Id> productIds = new Set<Id>();
3
4       for (HandsMen_Order__c order : Trigger.new) {
5           if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
6               productIds.add(order.HandsMen_Product__c);
7           }
8       }
9
10      if (productIds.isEmpty()) return;
11
12      // Query related inventories based on product
13      Map<Id, Inventory__c> inventoryMap = new Map<Id, Inventory__c>(
14          [SELECT Id, Stock_Quantity__c, HandsMen_Product__c
15           FROM Inventory__c
16           WHERE HandsMen_Product__c IN :productIds]
17      );
18
19      List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();
```

Detailed Project Explanation with Example

1. Customer Registration

- A customer, Jhon, visits the store or website.
- In Salesforce: A record is created in the Customer object with his name, phone, email, etc.
- Validation Rule: Ensures the email is valid (e.g., must contain gmail.com).

2. Product Setup

- The admin adds product T-Shirt Cloth, into the Product object.
- Each product has a price and other details.
- Inventory is also created to manage stock for these products.

3. Order Placement

- Jhon decides to buy 2 T-shirt Cloth ($250/- each). An order is placed.
- In Salesforce: A new Order record is created.
- Apex Trigger: Automatically calculates Total Amount=250*4=$1000

4. Inventory Update

As soon as the order is placed:

- Apex Trigger on Inventory: Reduces shirt stock by 4.
- Validation Rule: Ensures stock never goes below 0.

5. Loyalty Program

- Elijah 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, Jhon becomes a Silver member.

6. Email Notifications

- When a new order is placed or loyalty status is updated:
- Flow Email Alert is triggered.
- Jhon gets an email:

  "Thanks for your purchase! Your loyalty status is now Silver."

7. Users and Roles

Salesforce users like store staff are created:

- **Niklaus Mikaelson**-Sales Role (Platform 1 Profile)
- **Kol Mikaelson** Inventory Role (Platform 1 Profile)

Conclusion

The *HandsMen Threads* capstone project represents a comprehensive and well-structured approach to automating a men's fashion business using the Salesforce platform. Through the use of custom objects, relationships, validation rules, flows, Apex triggers, and email automation, the project demonstrates a complete digital transformation of core retail operations—covering customer management, order processing, inventory tracking, and marketing outreach.

The project showcases the practical application of Salesforce's declarative and programmatic tools in building real-world CRM solutions. Automated flows ensure prompt communication through email alerts, while Apex triggers handle backend calculations and inventory updates. Role-based access, profiles, and permission sets enforce secure data usage based on job roles. Scheduled flows like loyalty status evaluation enhance long-term customer engagement and retention strategies.

Overall, the project not only reinforces the technical concepts learned through Trailhead but also provides a scalable, efficient system that reflects the needs of a growing fashion retail brand. By blending usability with powerful backend automation, *HandsMen Threads* delivers a robust model for modern retail CRM practices. It stands as a solid demonstration of how Salesforce can streamline business processes and increase operational efficiency in any small-to-medium enterprise.

Future Scope:

In the future, the *HandsMen Threads* system can be expanded by integrating Salesforce Commerce Cloud for real-time online sales and customer interactions. Adding customer community portals will enable self-service order tracking, loyalty program redemption, and support. Integrating Einstein Analytics can provide advanced insights into customer behavior and sales trends. A mobile-friendly interface using Salesforce Mobile SDK could enhance on-the-go sales and inventory access. Additionally, integrating third-party payment gateways and shipping APIs will fully automate the order-to-delivery process, making the system production-ready for real-world retail deployment.