

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

Abstract

This project outlines the design and implementation of a custom-built Salesforce CRM system for HandsMen Threads, a premium menswear and tailoring company. The core mission was to transform their customer management and operational processes by creating a centralized, automated, and intelligent platform.

We built a robust data architecture centered on five key custom objects: Customer, Order, Product, Inventory, and Marketing Campaign. To bring this system to life, we automated essential business processes using a combination of Record-Triggered Flows, Scheduled Flows, automated Email Alerts, and custom Apex code. This automation handles everything from sending order confirmations and updating loyalty statuses to providing real-time inventory alerts.

Data integrity and security were paramount. We implemented strict validation rules and a role-based access control system to ensure information is both accurate and secure. Furthermore, we developed a scheduled Apex process to autonomously handle routine tasks like low-stock updates.

The result is an end-to-end CRM solution that not only deepens customer relationships through personalized engagement but also drives operational efficiency through automation, all built on a scalable Salesforce foundation ready for future growth.

Objective

The primary objective of this initiative was to architect and deploy a bespoke Salesforce CRM for HandsMen Threads. The solution is designed to streamline core business operations, ensure unparalleled data consistency, and significantly elevate the customer experience.

By creating a unified system to manage customers, orders, products, inventory, and marketing campaigns, the project achieves the following goals:

- Automate critical workflows such as order confirmations, loyalty program advancements, and inventory notifications.
- Ensure data integrity by enforcing validation rules for precise and consistent record-keeping.
- Enable real-time visibility into stock levels and key customer engagement metrics.
- Improve cross-team collaboration through defined role-based permissions, streamlining decision-making.
- Cultivate personalized client relationships through targeted communication and dynamic loyalty rewards.

Technology Stack

Salesforce:

Salesforce is a leading cloud-based Customer Relationship Management (CRM) platform. It empowers businesses to centralize customer data, automate complex processes, and enhance every facet of their sales, service, and marketing efforts. Its powerful yet user-friendly environment combines intuitive point-and-click tools with advanced development capabilities, making it ideal for creating tailored business solutions.

Custom Objects:

These act like specialized database tables designed to store unique business data. For this project, we created:

- Customer_c: Stores comprehensive client profiles and interaction history.
 - Product_c: A centralized catalog for all product details, specifications, and pricing.
 - Order_c: Tracks all customer transactions and manages the entire order lifecycle.
 - Inventory_c: Monitors stock levels across all warehouse locations.
 - Marketing_Campaign_c: Manages promotional initiatives and tracks their performance.
- **Tabs:** Provide intuitive navigation links on the Salesforce dashboard, allowing users to easily access and manage the data within each custom object (e.g., a Product Tab for catalog management).
- **Profiles & Roles:** Profiles define a user's baseline permissions to view and edit data. Roles establish a hierarchy that controls data visibility across the organization (e.g., a sales representative vs. a sales manager).
- **Permission Sets:** These grant additional, granular access rights to users on top of their profile, allowing for flexible permission management without altering core profiles.
- **Validation Rules:** Business rules that enforce data quality. Examples include:
 - Ensuring email addresses contain a valid domain (e.g., "@gmail.com").
 - Preventing the entry of negative values in inventory quantity fields.
- **Email Templates & Alerts:** Pre-designed email formats (templates) that are automatically sent (alerts) by the system based on specific triggers, such as an order confirmation or a low-stock warning.

- **Flows:** Visual, low-code automation tools that execute business logic. They can create records, update fields, and send alerts without writing code.
- **Apex:** Salesforce's powerful, object-oriented programming language used to create complex custom logic and automation that extends beyond declarative tools, such as calculating order totals or managing real-time inventory deductions.

Project Execution: A Step-by-Step Journey

1. Setting Up the Developer Org

We began by provisioning a dedicated Salesforce Developer Edition org via the official developer portal. This sandbox environment was configured with administrative privileges, providing a secure space for building and testing the entire solution.

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Nogeline ✓ Last name: Bayor ✓

Job title: Developer ✓ Work email: nogelinebayor@gmail.com ✓

Company: North Eastern Mindanao ✓ Country/Region: Philippines ✓

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☐ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation, (1) the Developer Edition includes

2. Building the Data Foundation

To solve the problem of having our critical business information scattered across different systems, we built a custom, centralized hub right within our Salesforce platform. We created five core building blocks, each designed to manage a specific part of our operations:

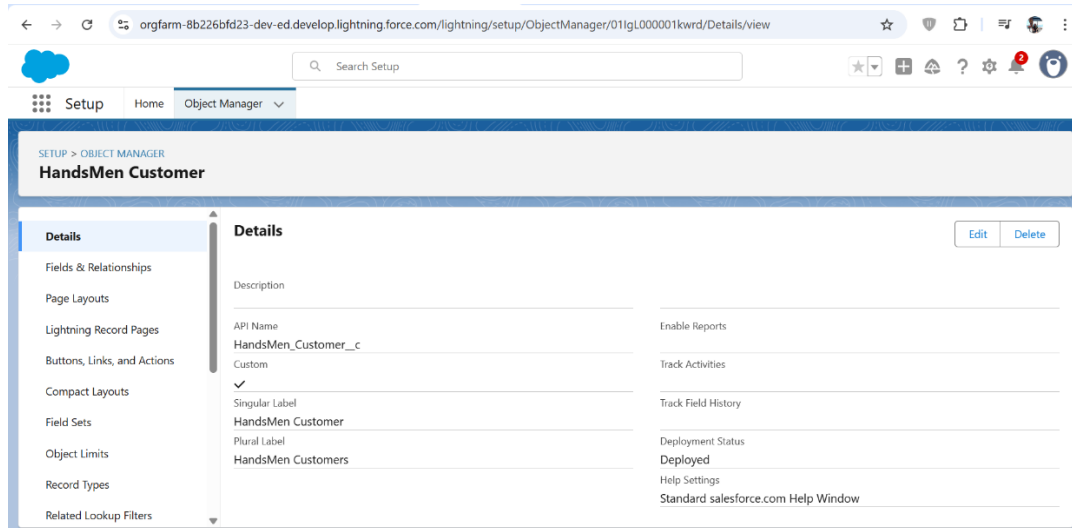
- **The HandsMen Customer Hub:** This is our single source of truth for every client. It neatly stores all contact details like emails and phone numbers, and also tracks how engaged they are with us through their loyalty tier status. Now, everyone from sales to support has a complete, 360-degree view of the customer.
- **The HandsMen Product Catalog:** This is our digital showroom. It holds our entire product lineup, complete with SKUs, pricing, and—most importantly—real-time inventory levels. This ensures our team and our customers always know what's available.
- **The HandsMen Order Tracker:** This object is all about the transaction. It captures every customer purchase, records what they bought and how much, and lets us follow its journey from "order placed" to "order delivered."

- **The Inventory Manager:** This works hand-in-hand with the Product Catalog to monitor our actual stock. It gives us a clear picture of what products are on hand and where they're located across our various warehouses.
- **The Marketing Campaign Center:** This is our command post for all promotional activities. We can plan campaigns, schedule them out, and then measure their performance with built-in analytics, all from one place.

How We Built It

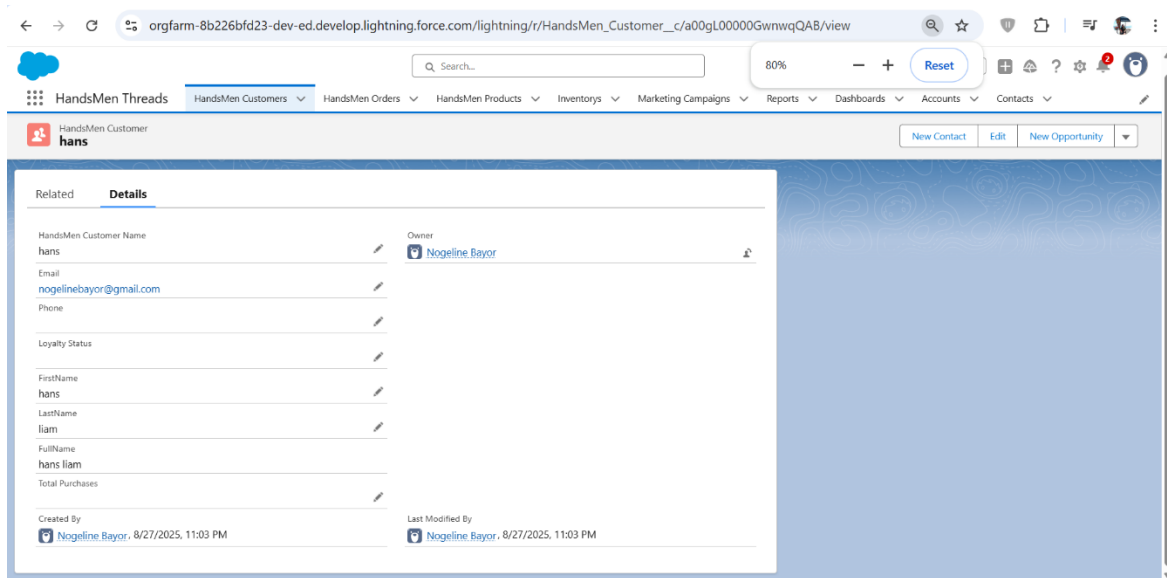
The process was straightforward and done entirely within Salesforce's powerful setup tools:

1. **Laying the Foundation:** We started in the Setup menu, navigating to **Object Manager** and clicking "**Create New Custom Object**" for each of our five building blocks.
2. **Giving Them an Identity:** For each object, we gave it a clear name (like "HandsMen Product"), assigned it a unique ID for the system to use, and made sure to turn on features for reporting and global search. This ensures our teams can easily find and create reports on all this new data.
3. **Making Them Accessible:** The final step was to make these new hubs easy for everyone to use. We generated dedicated tabs for each one, which now appear in our main Salesforce navigation. This gives our team seamless, one-click access to all our customer, product, order, inventory, and campaign data.



4. Creating the Unified Application Interface

To give users a seamless experience, we built a custom Lightning App named “HandsMen Threads.” This app brings all the key components—Customers, Orders, Products, Inventory, Campaigns, and Reports—into one centralized workspace, eliminating the need to navigate between different modules.

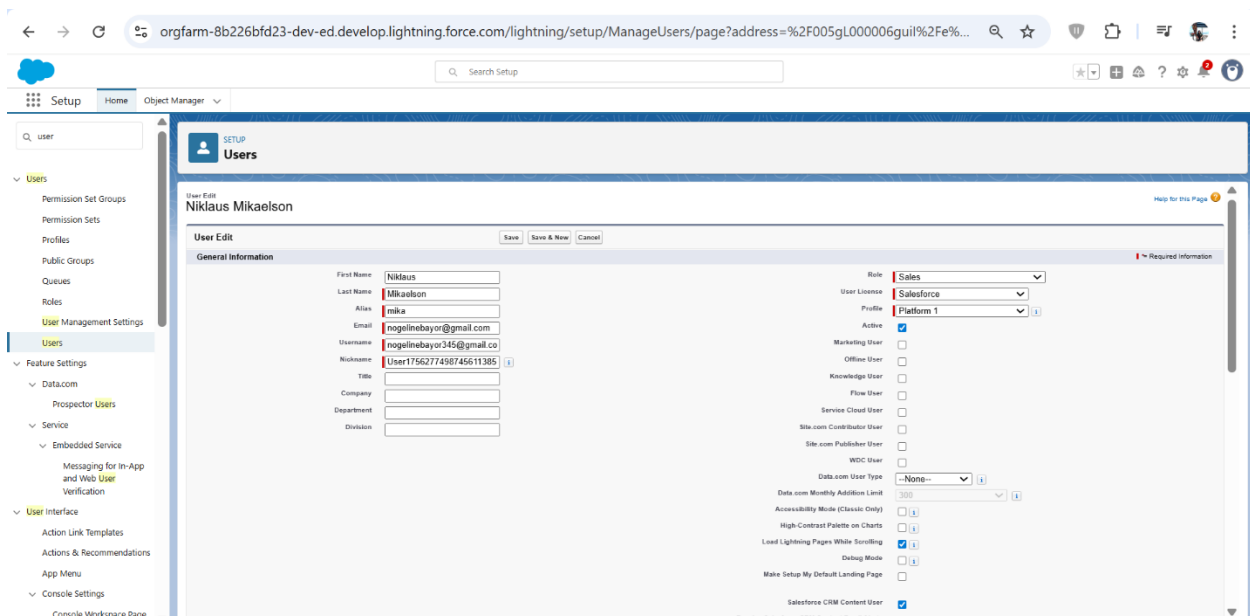


5. Configuring Security and Access

We established a structured security model by creating a custom user profile and defining hierarchical roles (e.g., Sales Manager, Inventory Manager). This ensures that team members only see the data and have the permissions relevant to their job functions, balancing efficiency with security.

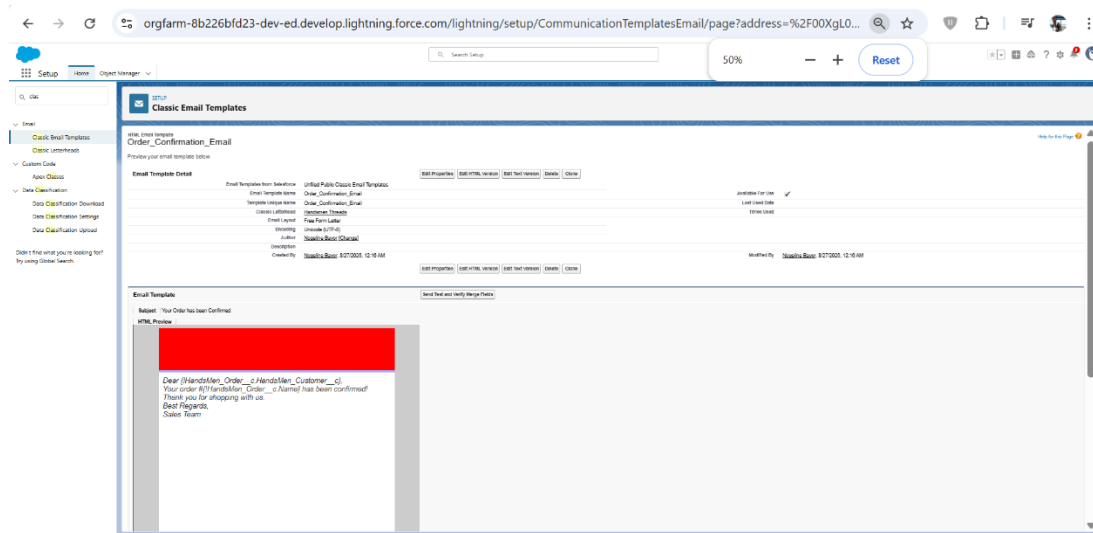
6. Onboarding Users

Example users were created and assigned to their respective roles and profiles, demonstrating how the security model works in practice and ensuring that everyone has the appropriate level of system access.



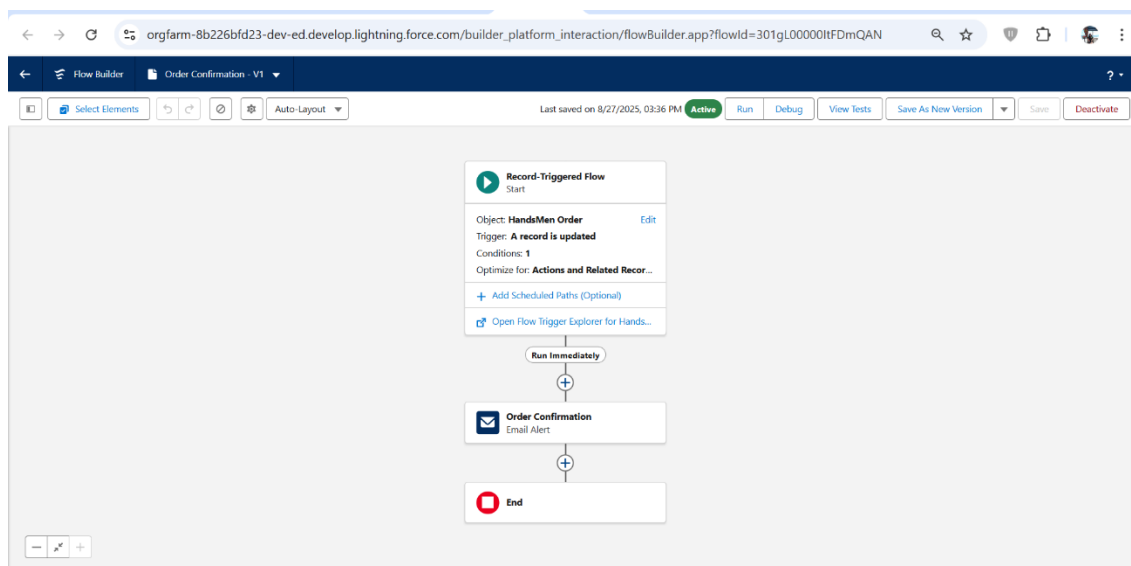
7. Automating Customer Communications

We designed three key email templates (Order Confirmation, Low Stock Alert, Loyalty Program Update) and configured automated alerts. These are triggered by specific events in the system, ensuring timely and consistent communication with customers and internal teams.

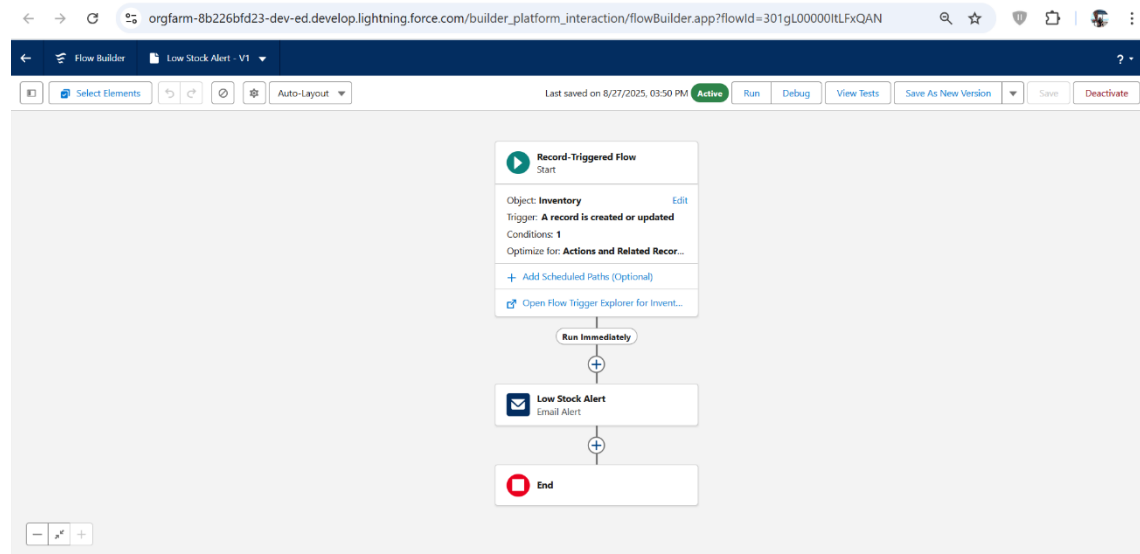


8. Designing Automated Workflows (Flows)

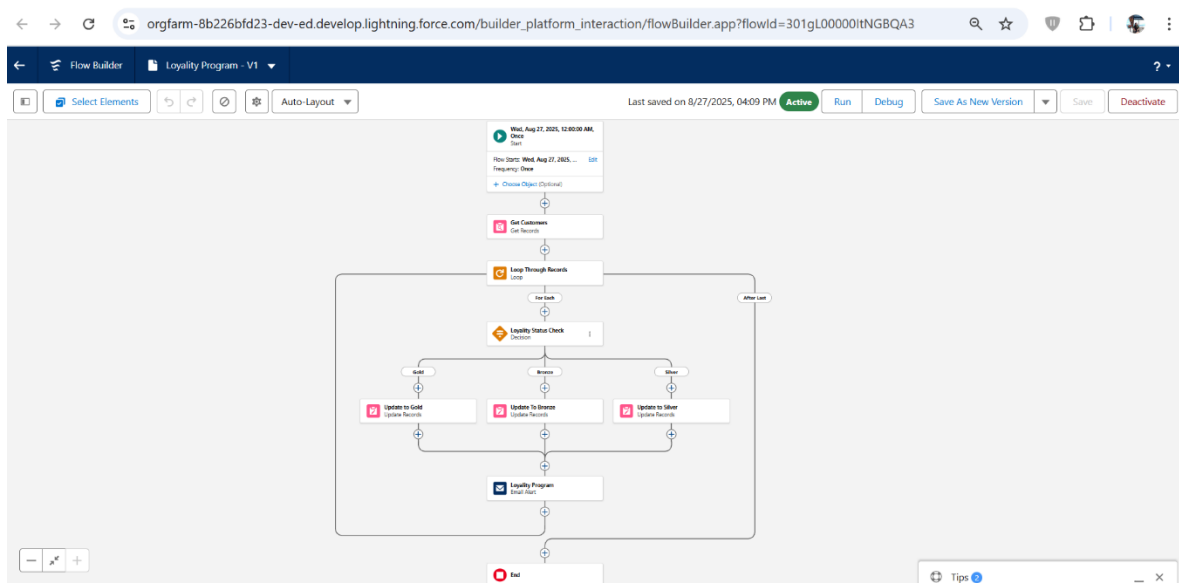
- **Order Confirmation Flow:** Triggered the moment an order's status is set to "Confirmed," this flow automatically sends a personalized confirmation email to the customer.



- **Low Stock Alert Flow:** Monitors inventory levels and instantly notifies the Inventory Manager when stock for any item falls below a predefined threshold.



- **Scheduled Loyalty Flow:** Runs every night to evaluate all customer purchase histories and automatically promote them to new loyalty tiers (Silver, Gold) based on their total spending.



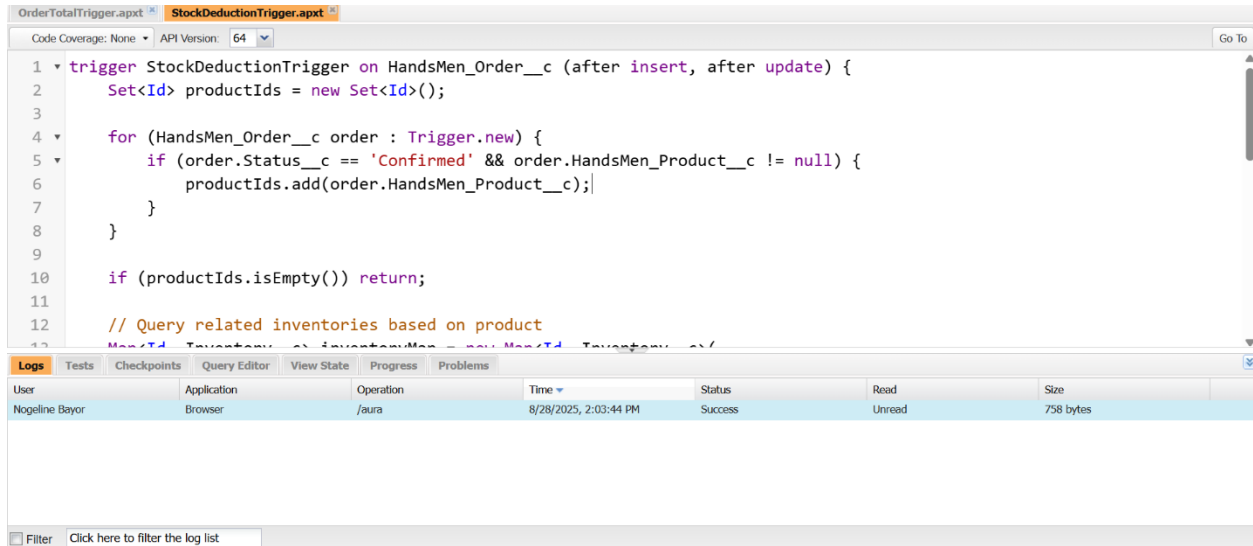
9. Developing Custom Business Logic (Apex)

For more complex automation, we wrote Apex triggers:

- One trigger automatically calculates the total amount of an order by multiplying quantity by unit price.
- Another trigger updates inventory levels in real-time when an order is completed.

- A third trigger dynamically adjusts a customer's loyalty status based on their accumulated purchase value.

OrderTotalTrigger



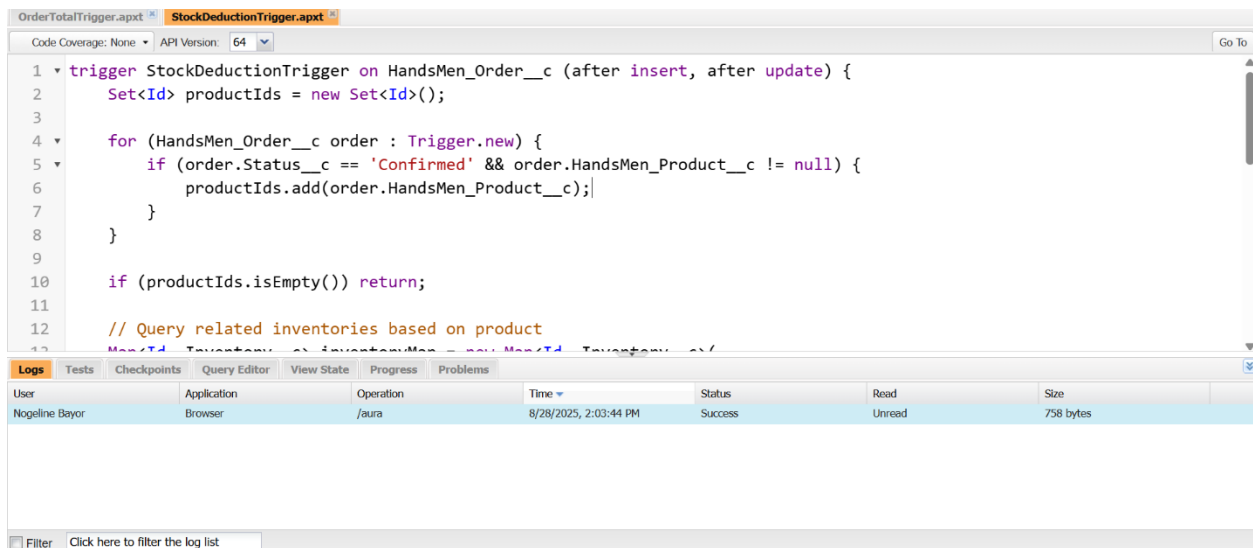
The screenshot shows the code editor for OrderTotalTrigger.apxt. The code is as follows:

```
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> inventoryMap = new Map<Id, Inventory>();
```

The logs table below the code editor shows a single log entry:

User	Application	Operation	Time	Status	Read	Size
Nogeline Baylor	Browser	/aura	8/28/2025, 2:03:44 PM	Success	Unread	758 bytes

StockDeductionTrigger



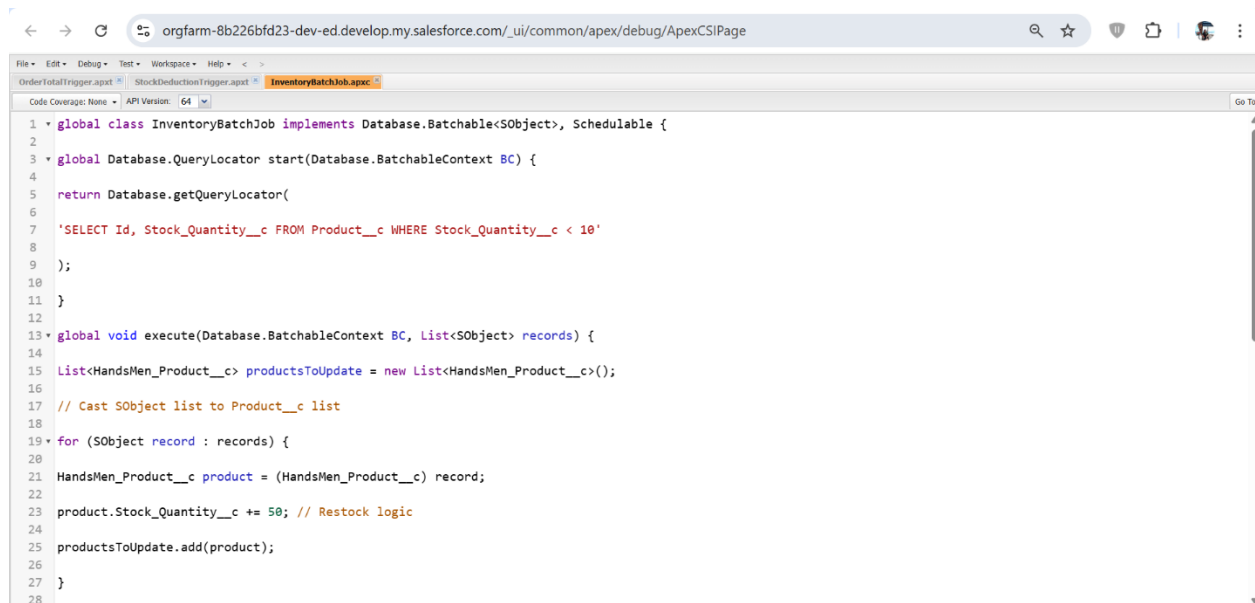
The screenshot shows the code editor for StockDeductionTrigger.apxt. The code is as follows:

```
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> inventoryMap = new Map<Id, Inventory>();
```

The logs table below the code editor shows a single log entry:

User	Application	Operation	Time	Status	Read	Size
Nogeline Baylor	Browser	/aura	8/28/2025, 2:03:44 PM	Success	Unread	758 bytes

InventoryBatchJob



The screenshot shows an IDE window with the URL `orgfarm-8b226bfd23-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The editor displays the `InventoryBatchJob.apex` file. The code is as follows:

```
1 global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {
2
3     global Database.QueryLocator start(Database.BatchableContext BC) {
4
5         return Database.getQueryLocator(
6
7             'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'
8
9         );
10    }
11
12    global void execute(Database.BatchableContext BC, List<SObject> records) {
13
14        List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();
15
16        // Cast SObject list to Product__c list
17
18        for (SObject record : records) {
19
20            HandsMen_Product__c product = (HandsMen_Product__c) record;
21
22            product.Stock_Quantity__c += 50; // Restock logic
23
24            productsToUpdate.add(product);
25
26        }
27    }
28}
```

A Real-World Scenario: Niklaus Customer Journey

To see the system in action, let's follow a customer named Niklaus Mikaelson:

1. **Registration:** Niklaus visits a store and is added to the CRM. A validation rule immediately checks that his email address is entered correctly.
2. **Product Setup:** An admin adds a new shirt to the Product catalog, setting its price and initial stock level.
3. **Placing an Order:** Niklaus buys two shirts. The system:
 - Creates an Order record.
 - An Apex trigger automatically calculates the correct total amount.
4. **Inventory Update:** Upon order confirmation, another trigger deducts the two shirts from inventory in real-time.
5. **Loyalty Reward:** The system adds Elijah's purchase to his lifetime total. An overnight Flow evaluates his new total and automatically upgrades his loyalty status from Bronze to Silver.
6. **Automated Communication:** Instantly, Niklaus receives a personalized email confirming his order and congratulating him on his new Silver status.
7. **Internal Coordination:** Based on their roles, the Sales team can view Niklaus order, while the Inventory team receives alerts if stock runs low, all within their controlled access views.

Conclusion

The HandsMen Threads CRM system successfully modernizes the brand's operations on the Salesforce platform. By automating core workflows—from customer management and order processing to inventory control and loyalty programs—the solution ensures data accuracy, provides real-time insights, and delivers a superior customer experience. The strategic use of automation and role-based security minimizes manual effort, reduces errors, and creates a scalable foundation for the company's continued growth.

Future Enhancements

Looking ahead, the platform can be extended with:

1. A **Customer Portal** for self-service account and order tracking.
2. A **Mobile App** for associates using the Salesforce SDK to manage operations on-the-go.
3. **Advanced Analytics Dashboards** for leadership to gain deeper insights into sales trends and inventory forecasting.
4. **AI-Powered Personalization** using Salesforce Einstein to generate intelligent product recommendations.
5. **Omnichannel Communications** to integrate notifications via WhatsApp and SMS, keeping customers informed through their preferred channels.