

Name : Agistia Ronna Aniqa

Departement : Computer Engineering

Institution : Indonesian University of Education (Kampus Daerah Cibiru)

## **HANDSMEN THREADS: ELEVATING THE ART OF SOPHISTICATION IN MEN'S FASHION**

### **Abstract**

This document explains the development of a custom Salesforce CRM solution for HandsMen Threads, a prominent brand in the men's fashion industry. As the company grew, the project aimed to simplify customer management, streamline order processing, improve inventory control, and strengthen marketing initiatives.

By fully utilizing the capabilities of Salesforce's cloud-based platform, critical business workflows were successfully automated, data integrity was consistently maintained, and real-time operational visibility was made available to all teams. The implemented solution consists of custom objects, intelligent automation flows, strict data validation rules, scheduled email notifications, and tailored Apex code, all working together to enhance customer engagement, increase sales performance, prevent inventory-related issues, and deliver a clear and comprehensive overview of the entire business.

### **Objective**

The project was driven by a clear objective to design and deploy a customized Salesforce CRM for HandsMen Threads, aimed at streamlining everyday operations while enhancing overall customer satisfaction.

More specifically, the objectives of this initiative were to:

- Automate repetitive operational activities, including order confirmation notifications, customer loyalty level updates, and low-stock alerts.
- Ensure highly accurate and consistent data input through the implementation of strong and reliable validation rules.

- Provide teams with real-time access to key insights, such as inventory availability, customer interaction histories, and sales performance metrics.
- Enhance collaboration and communication among sales, inventory, and marketing departments.
- Create a personalized customer experience by utilizing automated messaging systems and integrated loyalty programs.

## **Technology Description**

Let's explore the Salesforce capabilities that powered this project.

- **Salesforce Platform**

Serving as the core system of our operations, Salesforce acts like a central hub for managing all customer-related data through its cloud-based CRM environment. The platform offers a rich set of tools that support task automation, enhanced customer service, optimized marketing campaigns, and increased sales performance. To address different levels of complexity, we utilized both intuitive drag-and-drop functionalities and advanced development tools such as Apex and Flows.

- **Custom Objects**

These custom objects function as purpose-built digital storage units within Salesforce, created to manage and organize HandsMen Threads' specific business data, including:

- Customer\_\_c

This object stores complete customer profiles, covering personal information, contact details, and current loyalty status.

- Order\_\_c

This object systematically records every customer order, linking each transaction to the relevant customer and products, along with details such as order date, total value, and quantity purchased.

- Product\_\_c

It is used to manage detailed product information, including product names, descriptions, pricing, and how many available stock.

- Inventory (Inventory\_\_c)

For stock availability information, including Stock Quantity, Stock Status, Warehouse, and linked to HandsMen Product.

- Marketing\_Campaign\_\_c

It is designed to monitor and document marketing and promotional activities, making it easier to evaluate which campaigns deliver the best results.

- **Tabs**

Tabs function as convenient navigation aids that allow users to locate information more easily. To support efficient data handling, custom tabs were created for each custom object, such as Products and Orders, enabling the team to quickly access, create, and manage records directly within the “HandsMen Threads” application.

- **Custom App**

A custom Lightning App named “HandsMen Threads” was developed to function as the central workspace for the team. This application consolidates all essential tabs including Customer, Order, Product, Inventory, Campaign, and Reports into a single interface, enabling a more efficient and seamless workflow.

- **Profiles & Roles**

These components are used to regulate user access and system capabilities, determining what users can view and what actions they are allowed to perform.

- **Profiles**

Profiles are responsible for defining user permissions, including accessible objects and fields, as well as the layout of user interfaces. To align with specific responsibilities, we created custom profiles, such as the “HandsMen Sales Profile,” tailored to different job roles.

- **Roles**

Roles control data visibility based on the organizational hierarchy. For instance, a Sales Manager is granted access to all sales-related data, whereas a Sales Representative can only view their own records, ensuring a clear reporting structure within the organization.

- **Permission Sets**

These function as additional permission layers that allow users to access certain features or tools beyond the limits of their default profiles. By applying these extra permissions, we can fine-tune user access without altering or disrupting their primary profile configurations.

- **Validation Rules**

Validation rules act as protectors of data quality, ensuring that any information entered by users complies with predefined standards before being saved in the system. For example:

- Email fields are validated to follow proper email formats, such as requiring the presence of “@gmail.com”.
- Orders are blocked from being saved if the transaction would cause the product’s stock quantity to drop below zero.

- **Email Templates & Alerts**

These features are designed to support automated and consistent communication across the system:

- Email Templates

These consist of preformatted email designs, such as the Order Confirmation template, which help ensure that all outgoing messages maintain a professional appearance and consistent format.

- Email Alerts

These are system-generated email notifications that are triggered automatically when specific conditions are met, for example when a new order is created or when a customer’s loyalty status is updated.

- **Flows**

Flows function as visual automation tools that enable the creation of complex business logic without requiring any coding. Several important flows were implemented as part of this system, including:

- Order Confirmation Flow

This flow is designed to automatically trigger and send an order confirmation email to the customer immediately after a new order is successfully created.

- Stock Alert Flow

This automation is designed to automatically adjust the available product inventory whenever an order is created or updated, ensuring stock levels always reflect the latest transactions.

- Loyalty Program Flow

This automated flow is scheduled to run daily, for example at midnight, to evaluate customers’ total purchase values and update their loyalty status accordingly.

- **Apex**

Apex serves as Salesforce's programming language, applied when advanced or highly specific business logic cannot be achieved using standard declarative tools. In this project, several Apex triggers were implemented, including:

- Order Total Trigger

This trigger automatically computes and updates the `Total_Amount__c` field on each order by calculating the quantity against the unit price, ensuring accurate order totals.

- Stock Deduction Trigger

Operating in the background, this trigger automatically decreases product stock levels once an order is confirmed. It works in conjunction with validation rules to prevent overselling and maintain inventory accuracy.

- Loyalty Status Trigger

This trigger updates the `Loyalty_Status__c` field in real time, adjusting a customer's loyalty status based on their accumulated purchase history.

## Detailed Project Execution

### 1. Salesforce Developer Org Setup

A dedicated Salesforce Developer Org was created through the Salesforce developer signup portal and used as the primary environment for all development activities throughout the project.

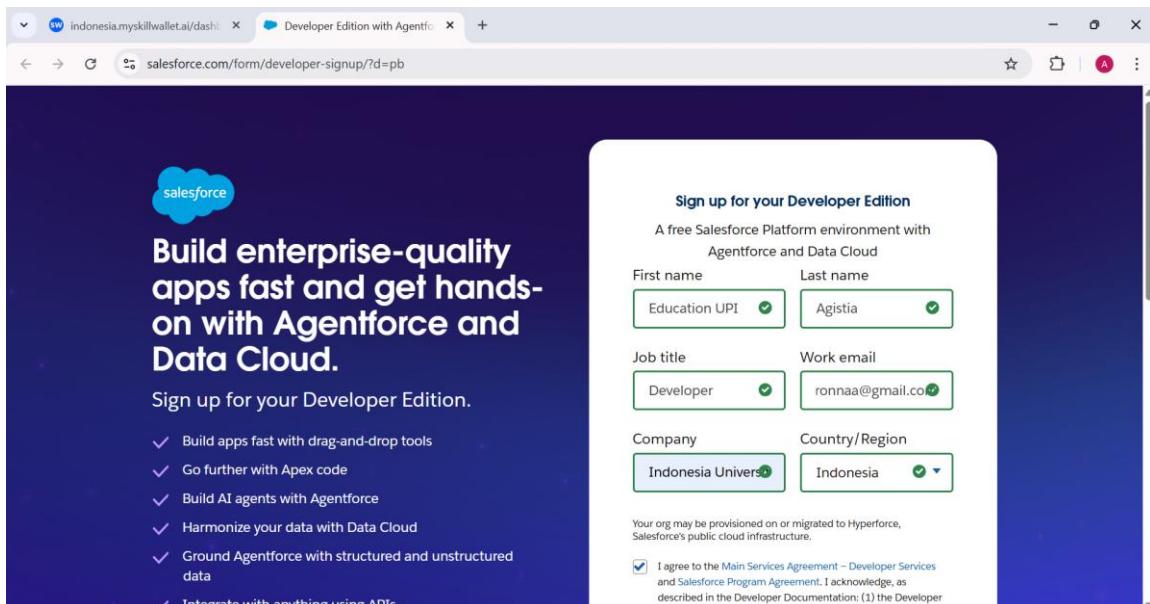


Figure 1. Creating a Salesforce Developer Org

## **2. Custom Object**

The foundational data architecture was established by designing custom objects along with their respective fields to store all essential business data, including:

- HandsMen Customer (Customer\_\_c)

This object is used to store comprehensive customer information, such as Name, Email, Phone, Address, and Loyalty\_Status\_\_c.

- HandsMen Order (Order\_\_c)

This object manages customer order records, capturing Date, Quantity\_\_c, and Total\_Amount\_\_c, while linking each order to the corresponding Customer and Product.

- HandsMen Product (Product\_\_c)

It holds detailed product data, including Name, Description, Unit\_Price\_\_c, and Stock\_\_c.

- Inventory (Inventory\_c)

For stock availability information, including Stock\_Quantity, Stock\_Status, Warehouse, and linked to HandsMen Product.

- Marketing Campaign (Marketing\_Campaign\_\_c)

It is designed to record and manage promotional and marketing activities.

## **3. Custom Lightning App**

The “HandsMen Threads” Lightning App was developed to provide teams with a centralized and user-friendly workspace. This application consolidates all essential tabs such as Customer, Order, Product, Inventory, Campaign, and Reports into a single interface, allowing users to work more efficiently and seamlessly.

## **4. Validation Rules**

To maintain data quality, we put validation rules in place with the following purposes:

- HandsMen Order

Prevents Total\_Amount\_c values from being less than or equal to 0.

- Stock\_Quantity

Prevents Stock\_Quantity\_e values from being less than or equal to 0.

- Email

The system requires the Email field to include “@gmail.com”, ensuring that customer email entries follow the specified format.

## **5. User Profiles & Roles**

User access and data visibility were managed by configuring profiles and role structures tailored to organizational needs:

- Profiles

The default Standard User profile was duplicated and customized to create role-specific profiles, such as the “HandsMen Sales Profile.”

- Roles

An organizational hierarchy was established, including roles like Sales Manager and Inventory Manager, to regulate data visibility according to each user’s responsibilities.

## 6. User Creation

To showcase the system’s multi-user functionality, we added several users and assigned each of them to appropriate profiles and roles, such as Niklaus Mikaelson in the Sales role and Kol Mikaelson in the Inventory role.

The screenshot shows the Salesforce 'Users' page under the 'Setup' tab. The left sidebar has 'Users' selected. The main area displays a table of users with columns for Action, Full Name, Alias, Username, Role, Active, and Profile. The table includes rows for various users like Agustina, Chatter Expert, EPIC OrgFarm, Mikaelson Daniel, Mikaelson Kol, Mikaelson Niklaus, User Integration, and User Security, each assigned specific roles like System Administrator, Marketing, Inventory, Sales, etc., and profiles like Chatter Free User, Platform 1, and Analytics Cloud Security User.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>   Edit	Agustina_Education_UPI	agi	agistaronna463@agentforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>   Edit	Chatter_Expert	Chatty	chatty_009g000000g6juad.vnxv3cwoq1e@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>   Edit	EPIC_OrgFarm	OEPIC	epic.134d90000046@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>   Edit	Mikaelson_Daniel	dmika	agistaronna04@gmail.com	Marketing	<input checked="" type="checkbox"/>	Platform 1
<input type="checkbox"/>   Edit	Mikaelson_Kol	kmika	agistaronna00@gmail.com	Inventory	<input type="checkbox"/>	Platform 1
<input type="checkbox"/>   Edit	Mikaelson_Niklaus	rnmika	agistaronna19@gmail.com	Sales	<input checked="" type="checkbox"/>	Platform 1
<input type="checkbox"/>   Edit	User_Integration	integ	integration@00dgk00000g6juad.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>   Edit	User_Security	sec	insightsssecurity@00dgk00000g6juad.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

Figure 2. User List

## 7. Email Templates

To support consistent and automated communication, we developed a set of custom email templates, including templates for Order Confirmation Email, Low Stock Alert Email, and Loyalty Program Email.

## 8. Flow Implementations

- Order Confirmation Flow (Record-Triggered Flow)

This flow is configured to automatically deliver an order confirmation email to customers immediately after a new order record is created.

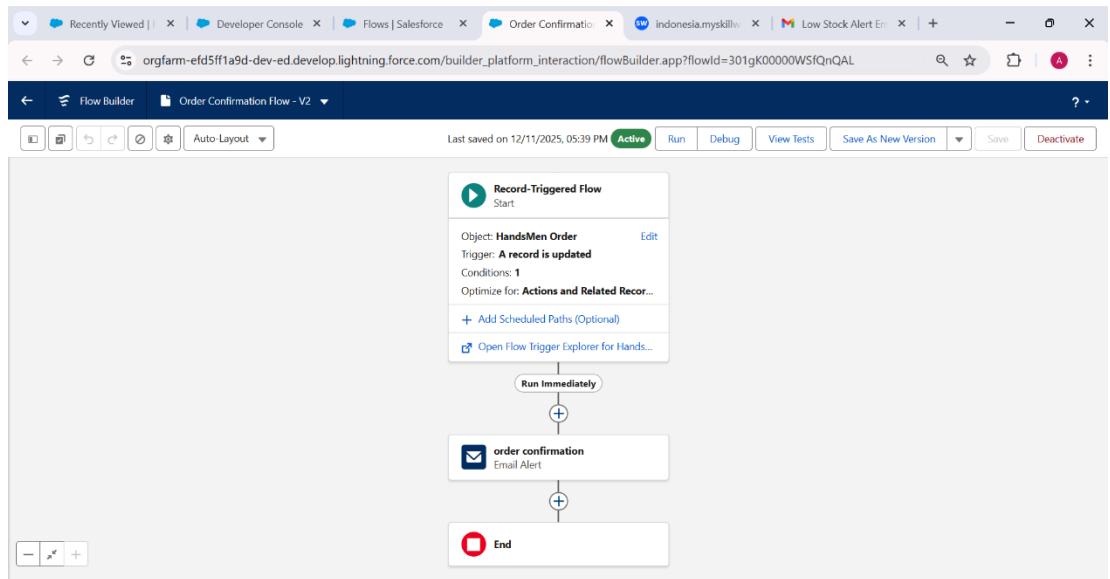


Figure 3. Order Confirmation Flow

- Stock Alert Flow (Record-Triggered Flow)

Product inventory are updated automatically by this flow whenever an order is created or modified.

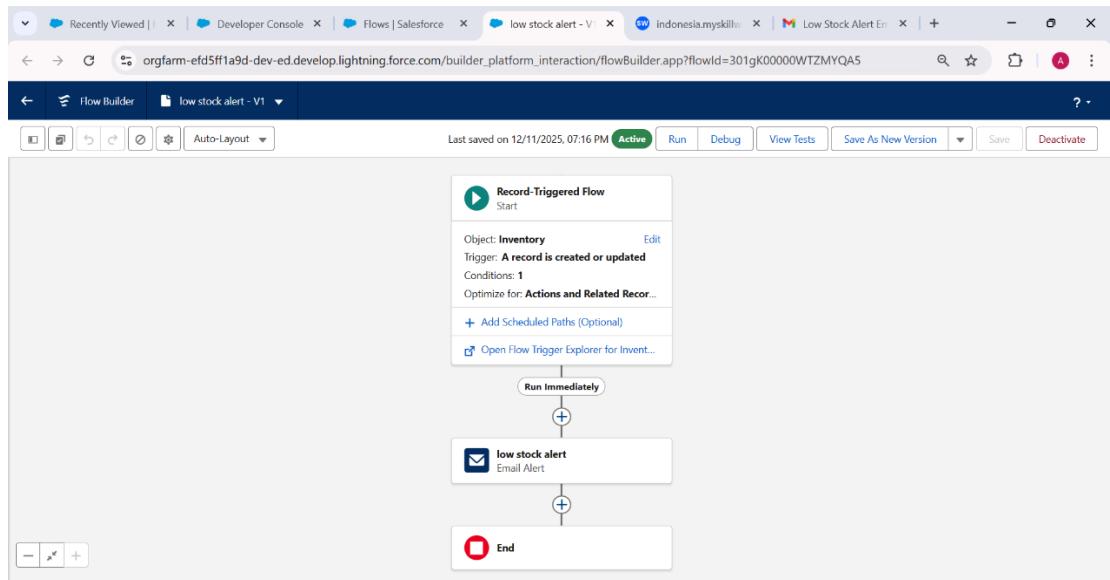


Figure 4. Stock Alert Flow

- Loyalty Program Flow (Scheduled-Triggered)

Runs daily to update customer loyalty status based on their total purchase amounts.

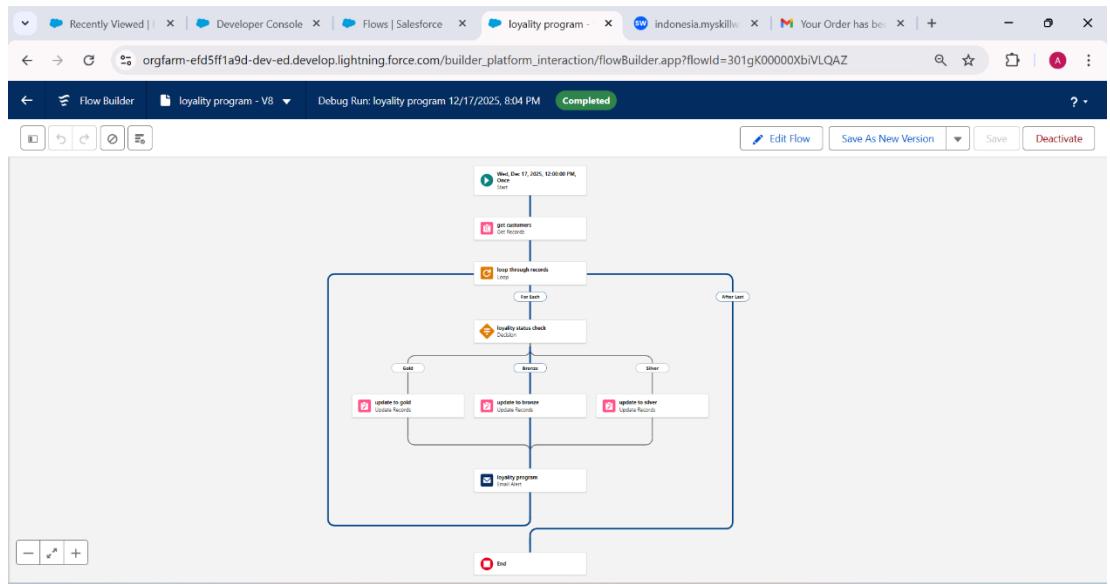


Figure 5. Loyalty Program Flow

## 9. Apex Triggers

- Order Total Trigger (OrderTotalTrigger)

This trigger is responsible for calculating the total value of an order by using the quantity and unit price, then automatically updating the order's total amount field.

```

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]
    );
}

```

User	Application	Operation	Time	Status	Read	Size
Education UPI Agista	Unknown	/services/data/v65.0/tooling/ex...	12/17/2025, 5:59:11 PM	Success	Unread	2.87 KB
Education UPI Agista	Browser	/aura	12/17/2025, 5:52:59 PM	Success	Unread	13.81 KB
Education UPI Agista	Unknown	common.api.soap.DirectSoap	12/17/2025, 5:52:59 PM	Success	Unread	521 bytes
Education UPI Agista	Browser	/aura	12/17/2025, 5:51:25 PM	Success	Unread	8.36 KB
Education UPI Agista	Unknown	common.api.soap.DirectSoap	12/17/2025, 5:51:24 PM	Success	Unread	526 bytes
Education UPI Agista	Browser	/aura	12/17/2025, 5:49:58 PM	Success	Unread	1.46 KB

Figure 6. Order Total Trigger

- Stock Deduction Trigger (StockDeductionTrigger)

Once an order is processed, this trigger accurately adjusts the product's stock level, ensuring inventory is reduced correctly after each order.

```

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
}

```

User	Application	Operation	Time	Status	Read	Size
Education UPI Agista	Unknown	/services/data/v65.0/tooling/ex...	12/17/2025, 5:59:11 PM	Success	Unread	2.87 KB
Education UPI Agista	Browser	/aura	12/17/2025, 5:52:59 PM	Success	Unread	13.81 KB
Education UPI Agista	Unknown	common.api.soap.DirectSoap	12/17/2025, 5:52:59 PM	Success	Unread	521 bytes
Education UPI Agista	Browser	/aura	12/17/2025, 5:51:25 PM	Success	Unread	8.36 KB
Education UPI Agista	Unknown	common.api.soap.DirectSoap	12/17/2025, 5:51:24 PM	Success	Unread	526 bytes
Education UPI Agista	Browser	/aura	12/17/2025, 5:49:58 PM	Success	Unread	1.46 KB

Figure 7. Stock Deduction Trigger

- Loyalty Status Trigger (InventoryBatchJob)

By analyzing a customer's accumulated purchase history, this trigger updates the loyalty status instantly and in real time.

```

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'
        );
    }
}

```

User	Application	Operation	Time	Status	Read	Size
Education UPI Agista	Unknown	/services/data/v65.0/tooling/ex...	12/17/2025, 5:59:11 PM	Success	Unread	2.87 KB
Education UPI Agista	Browser	/aura	12/17/2025, 5:52:59 PM	Success	Unread	13.81 KB
Education UPI Agista	Unknown	common.api.soap.DirectSoap	12/17/2025, 5:52:59 PM	Success	Unread	521 bytes
Education UPI Agista	Browser	/aura	12/17/2025, 5:51:25 PM	Success	Unread	8.36 KB
Education UPI Agista	Unknown	common.api.soap.DirectSoap	12/17/2025, 5:51:24 PM	Success	Unread	526 bytes
Education UPI Agista	Browser	/aura	12/17/2025, 5:49:58 PM	Success	Unread	1.46 KB

Figure 8. Inventory Batch Job

## Project Demonstration

This section illustrates the overall workflow and functionality of the HandsMen Threads system.

### 1. Customer Registration

A new customer named garvin reviano registered as a customer in the HandsMen Threads system. During the registration process, customer information such as HandsMen Customer Name, First Name, and Last Name was entered and stored in the

HandsMen Customer record. Once all required details were completed, garvin reviano was officially registered as a customer of HandsMen Threads.

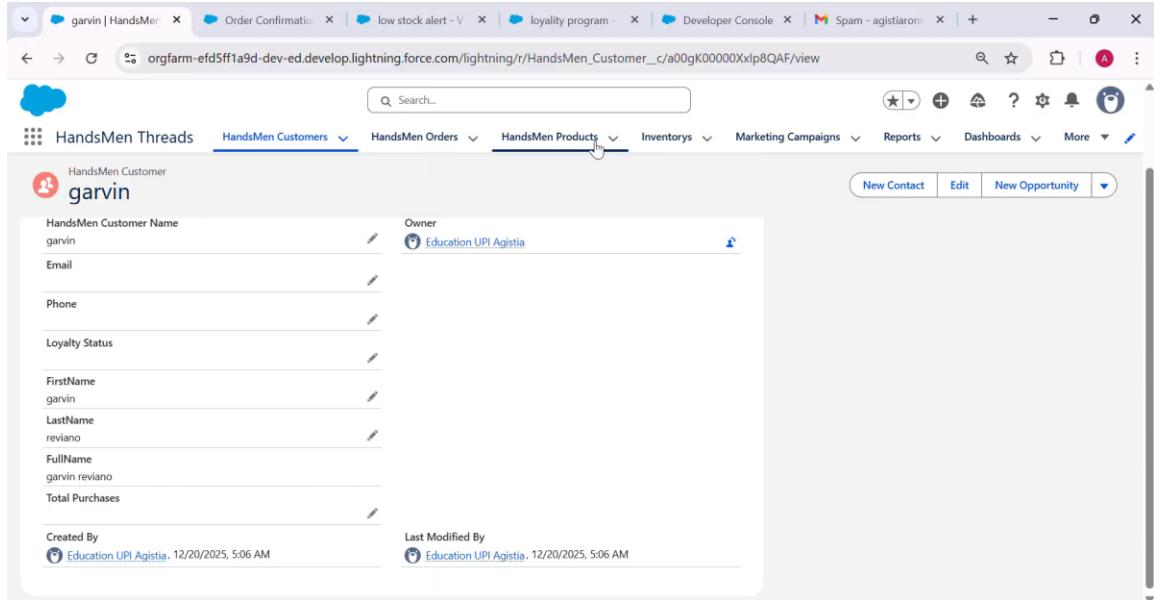


Figure 9. HandsMen Customer Registration

## 2. Product Setup

To register a new product in HandsMen Product, essential details such as the product name, pricing, and stock quantity are entered into the system. In this scenario, a belt item is added with a unit price of \$5 and an initial stock of 4 units.

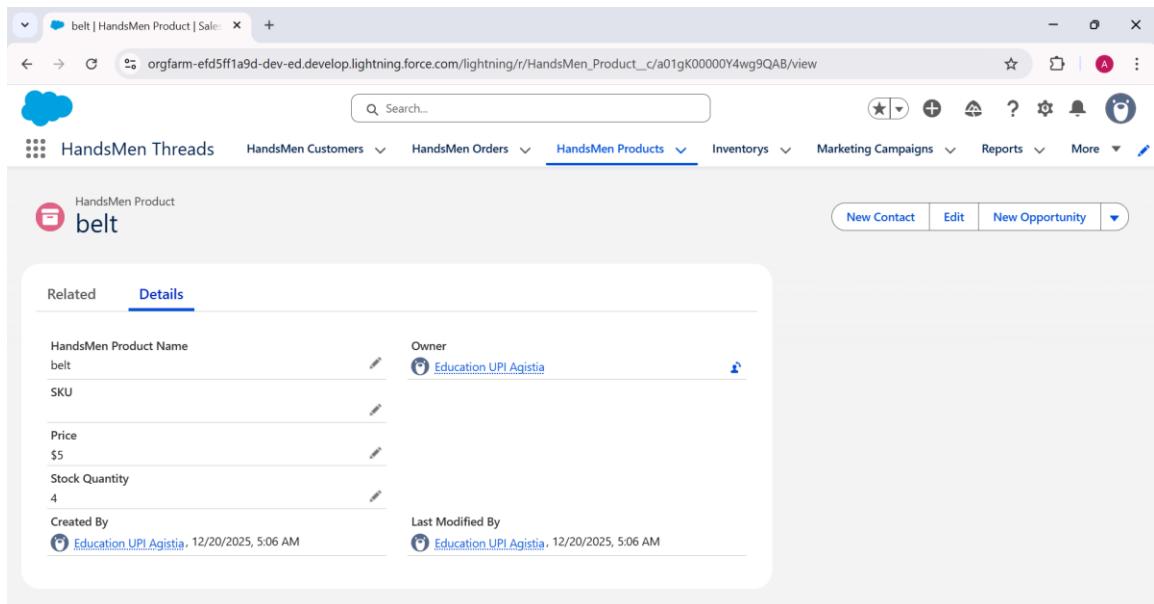


Figure 10. Detail HandsMen Product

## 3. Inventory Setup

Inventory are created and linked directly to HandsMen Product data. During the initial setup, the stock quantity is intentionally set to 3 units to allow the system to identify a low stock condition.

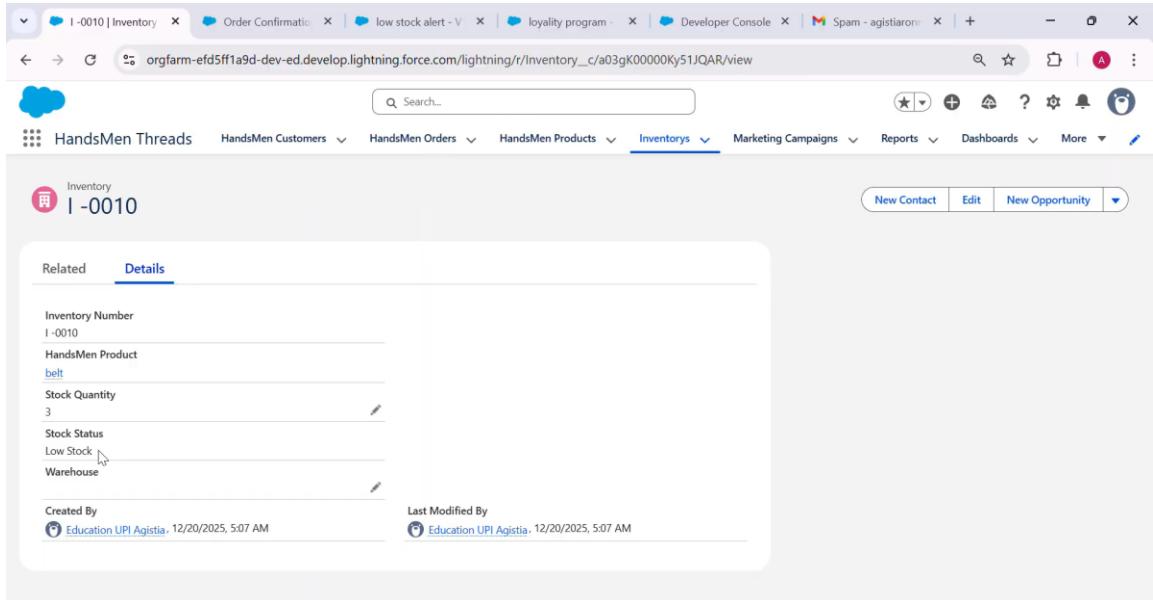


Figure 11. Inventorys Creation

The system automatically triggers a Low Stock Alert email that is sent to the Inventory Manager, informing them that the product's stock level has reached a low threshold.

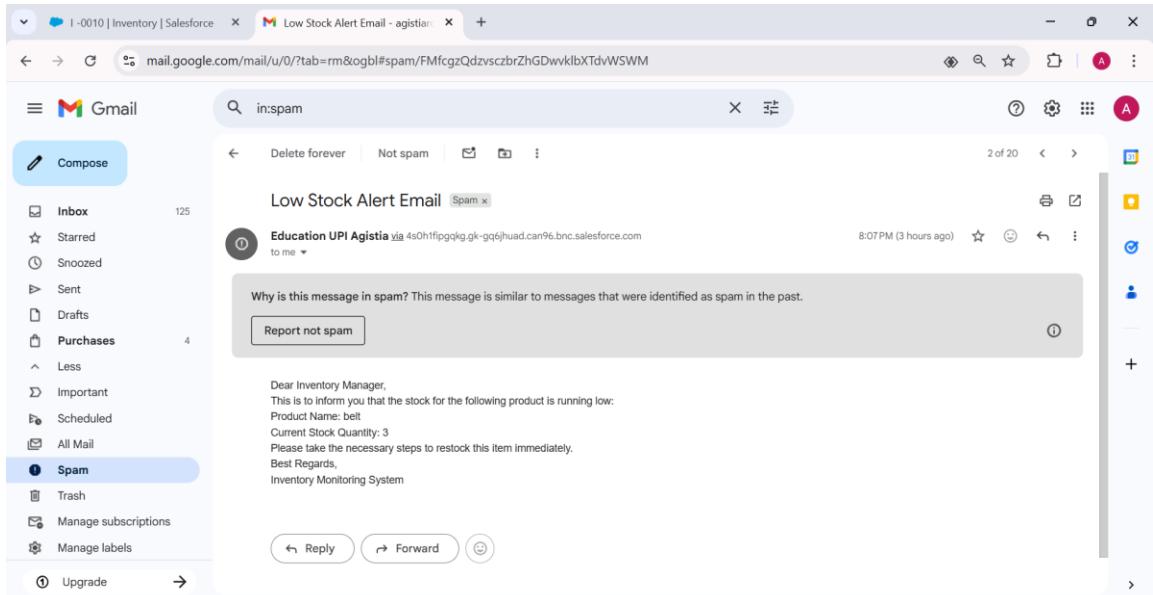


Figure 12. Low Stock Alert Email Notification

After the stock quantity was increased to 300 units, the system automatically updated the stock status to “Available,” indicating that the product had been successfully restocked and was ready for sale again.

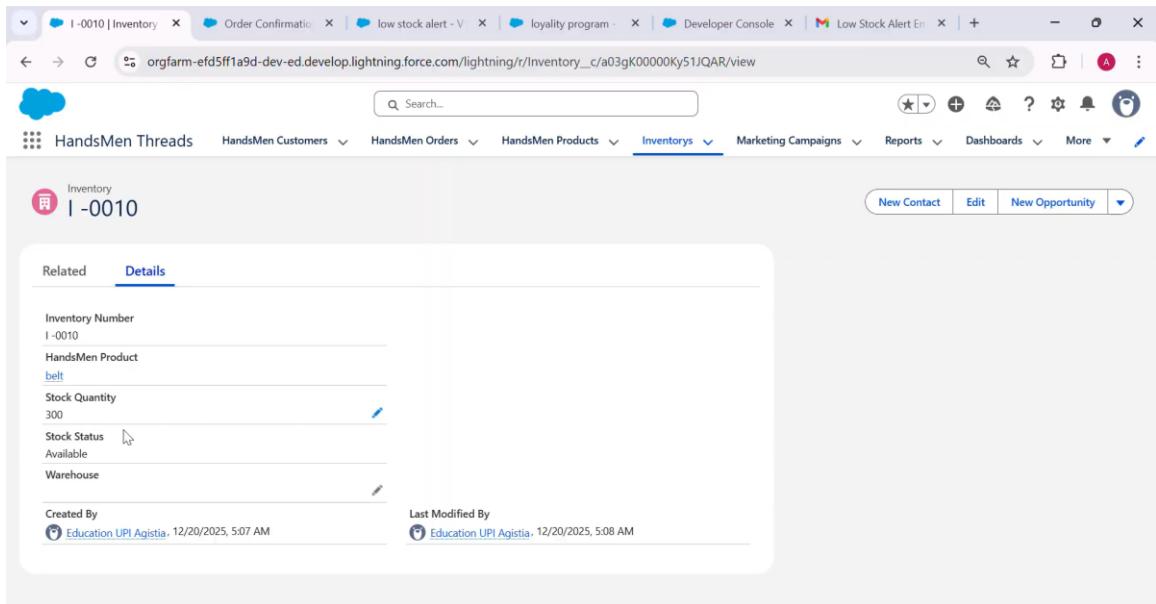


Figure 13. Detail Inventory

#### 4. Order Setup

At this stage, an order record is created to link HandsMen Customer with HandsMen Product. In this example, garvin places an order for belt item, setting the initial order status to Pending, specifying an order quantity of 100 units, and providing the customer email information.

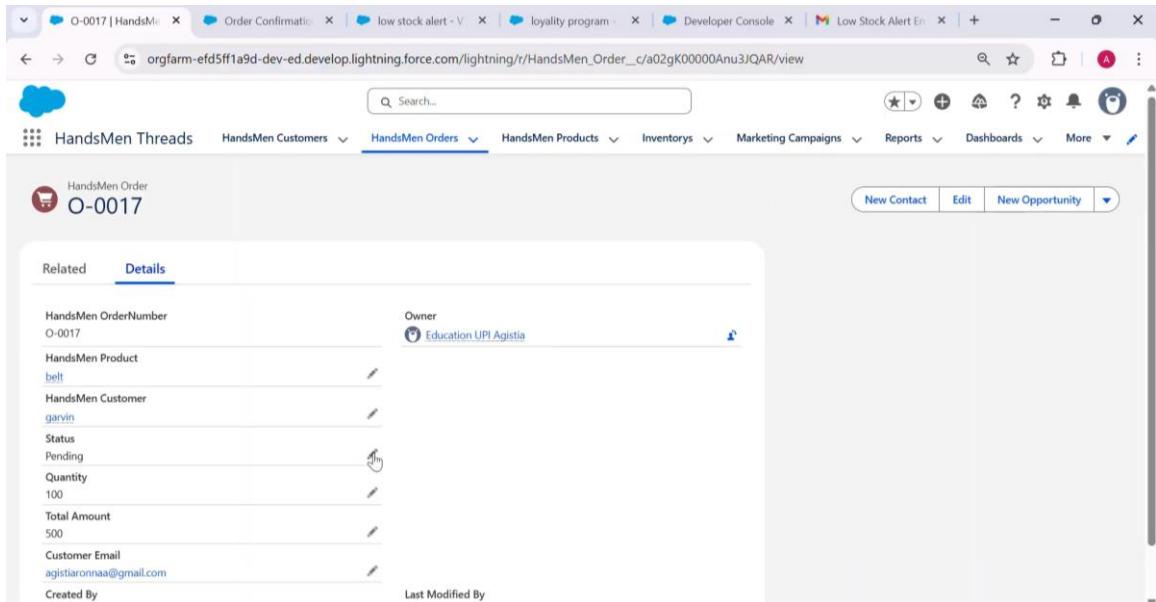


Figure 14. Order Setup

Once the order record is created, updating its status to Confirmed activates an automated process that sends an order confirmation email directly to the customer.

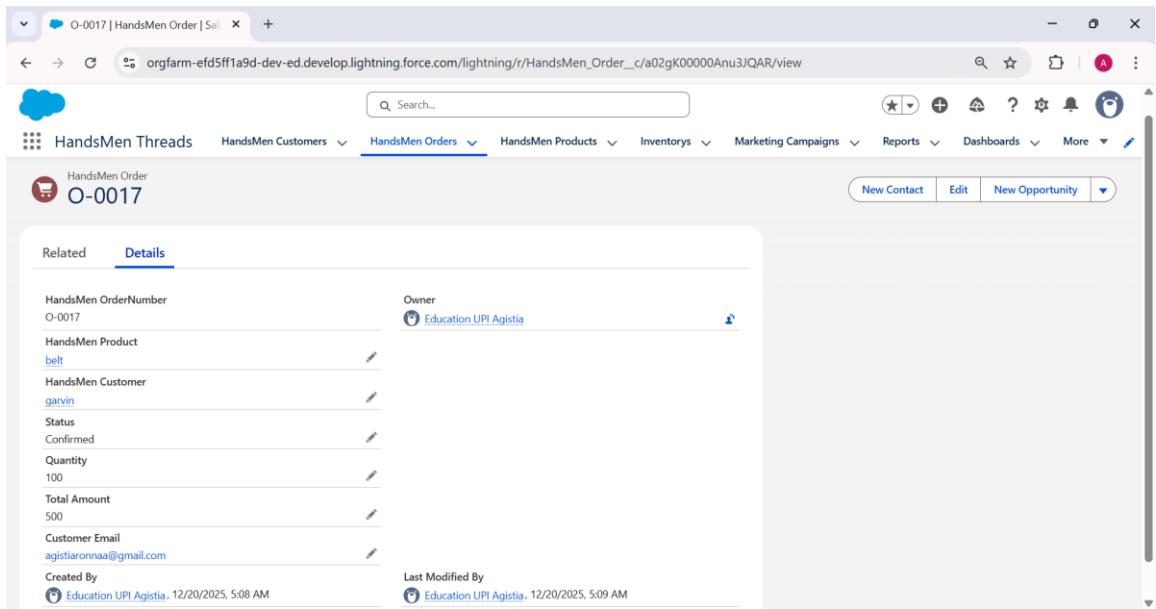


Figure 15. Update Order Status to Confirmed

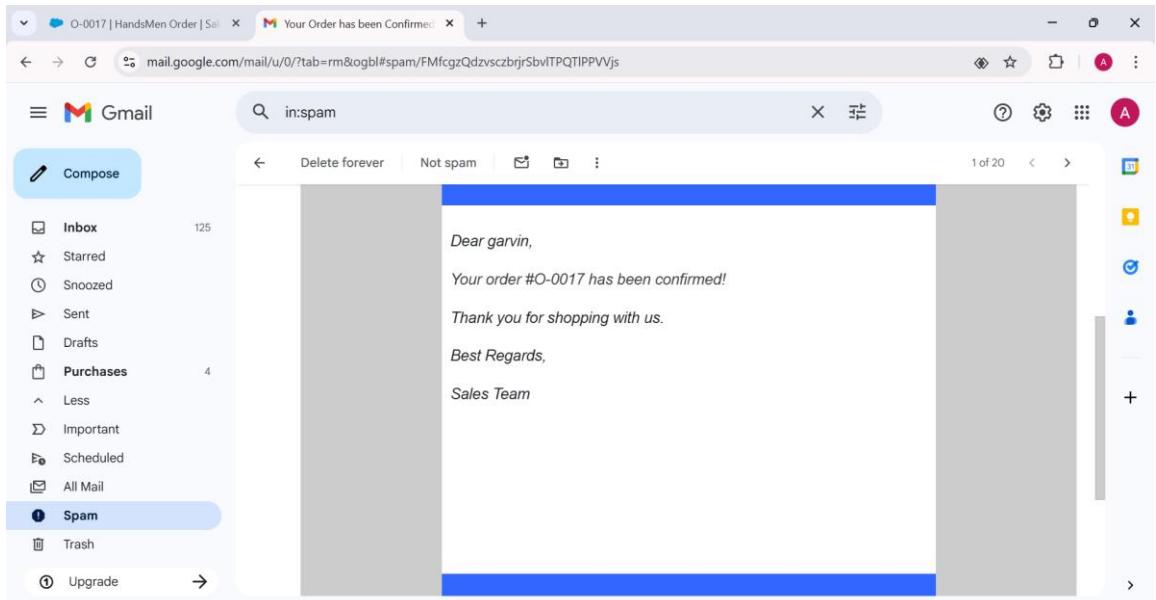


Figure 16. Order Confirmation Email Notification

## 5. Loyalty Status Update

The HandsMen Customer record is updated by modifying the Customer Email and Total Purchase fields. Afterward, the Loyalty Status Update Flow is debugged to verify that the customer's loyalty status is updated correctly based on the latest data.

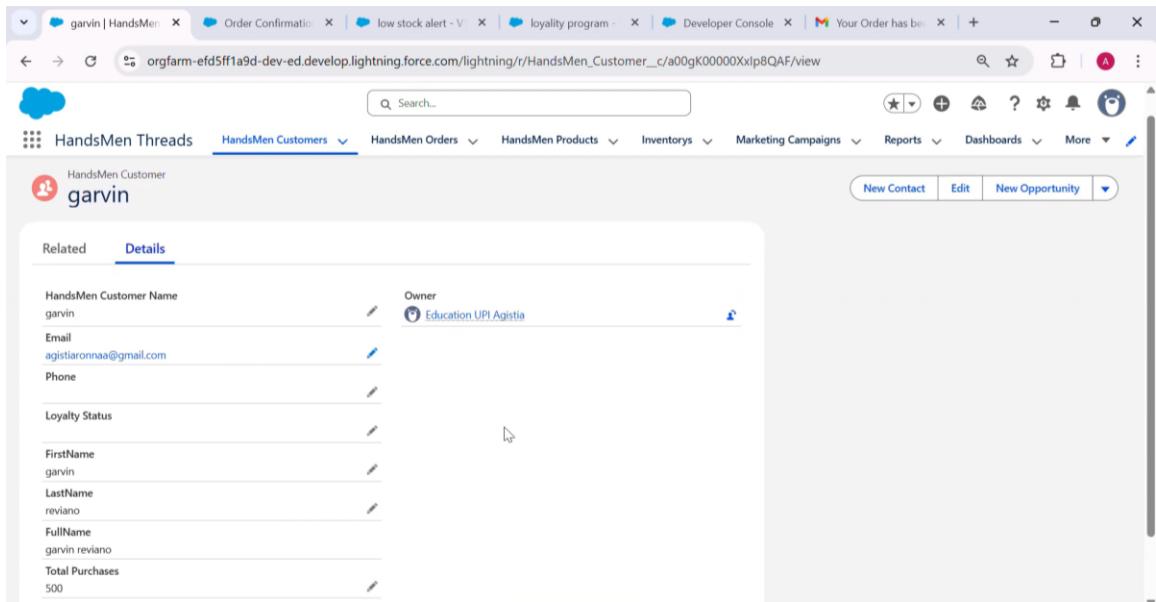


Figure 17. Update HandsMen Customer Data

The customer's Loyalty Status is automatically adjusted by the system according to the total purchase value recorded in the HandsMen Customer data.

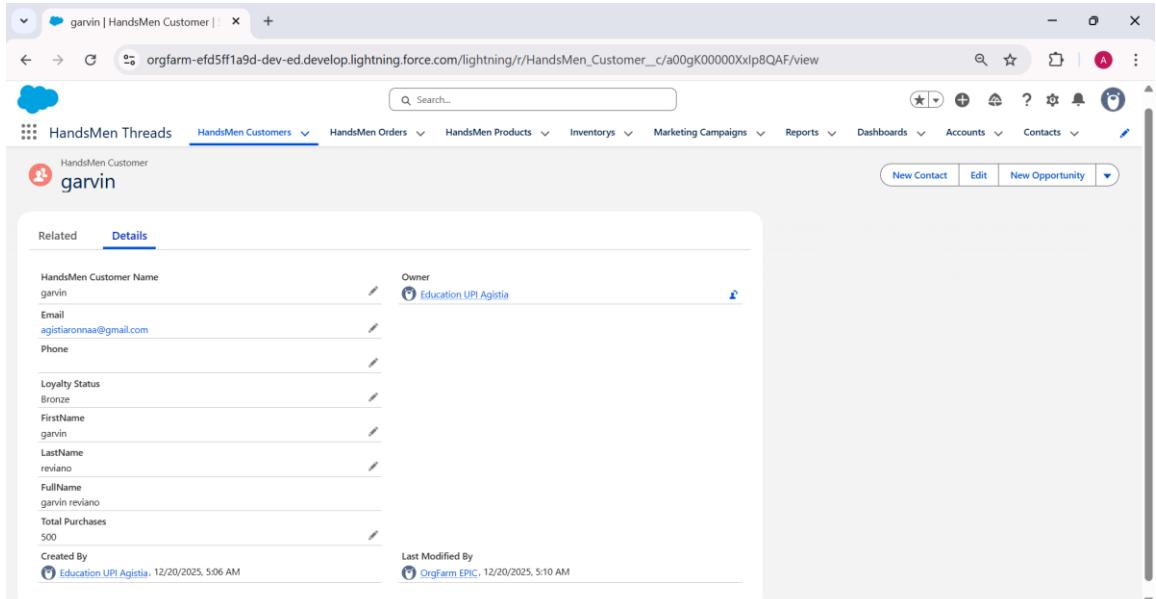


Figure 18. Update Loyalty Status

## 6. Apex Trigger Execution

Once an order reaches the Confirmed status, an Apex Trigger automatically calculates the Total Amount value. For example, when an order includes 100 items priced at \$5 each, the trigger processes this information and updates the Total Amount field to 500 without manual input.

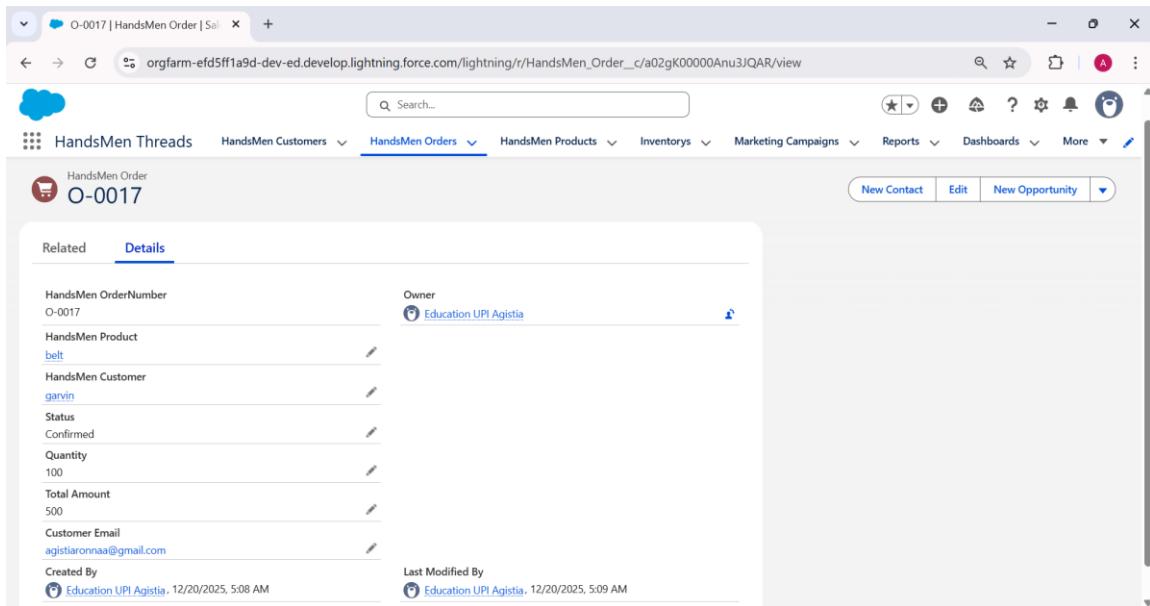


Figure 19. Total Amount in HandsMen Order

Additionally, the Stock Quantity value in Inventory is automatically adjusted through the execution of the Stock Deduction Trigger. Based on the quantity of confirmed orders, the stock amount previously set at 300 units is reduced to 200 units. This automated process helps maintain consistency and accuracy between order records and available inventory.

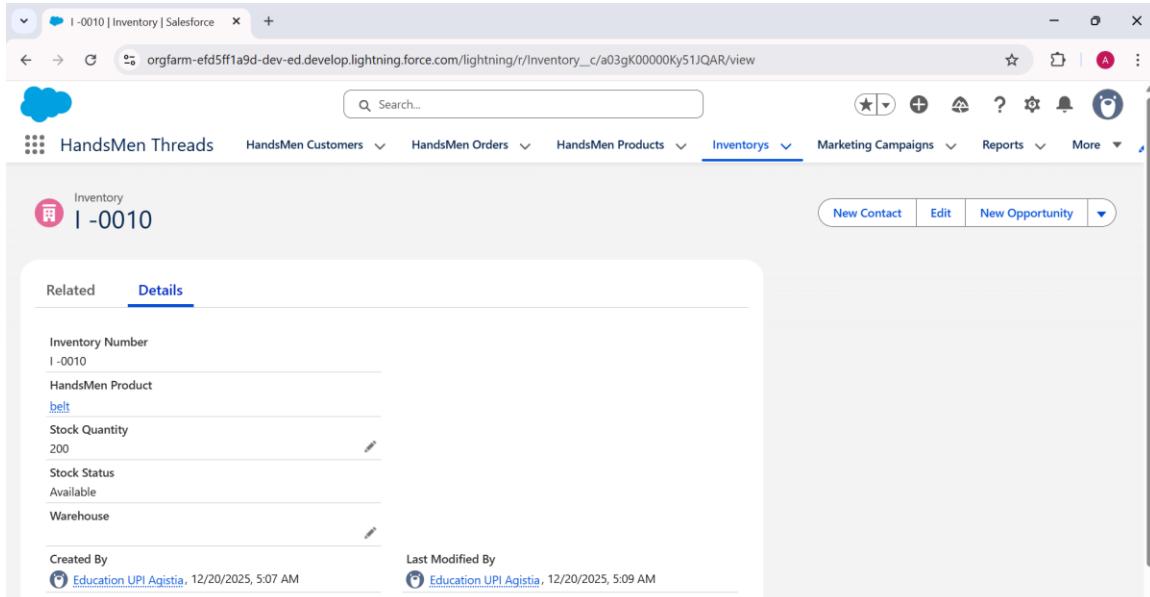


Figure 20. Stock Quantity in Inventory

## Conclusion

Based on the design, development, testing, and implementation phases, it can be concluded that the Salesforce CRM system developed for HandsMen Threads has successfully delivered

a well-structured and integrated solution for data management and business process automation. The system effectively centralizes customer, order, product, and inventory data into a single platform, leading to improved operational efficiency and data accuracy. The implementation of custom objects, validation rules, Salesforce Flows, and Apex Triggers has proven effective in maintaining data integrity, reducing manual processes, and supporting automated business workflows. Furthermore, the integration of automated email notifications enables faster, more consistent, and real-time communication with both customers and internal teams. Overall, the developed system meets the project's objectives and provides a solid foundation for future Salesforce-based system enhancements.