

Name : Salsabila Nida Azzahra
Department : Computer Engineering
Institution : Indonesian University of Education (Kampus Daerah Cibiru)

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

Abstract

This document outlines the design and implementation of a customized Salesforce CRM system for HandsMen Threads, a leading brand in the men's fashion sector. As the organization expanded, the project focused on optimizing customer relationship management, improving order handling efficiency, enhancing inventory management, and reinforcing marketing strategies.

By leveraging the full potential of Salesforce's cloud-based platform, essential business processes were effectively automated, data accuracy and consistency were ensured, and real-time operational insights became accessible to all departments. The developed solution integrates custom objects, advanced automation flows, robust data validation mechanisms, scheduled email alerts, and customized Apex programming. Together, these components support better customer engagement, boost sales productivity, minimize inventory risks, and provide a comprehensive and transparent view of overall business operations.

Objective

This project aimed to develop and deploy a tailored Salesforce CRM system for HandsMen Threads to support daily operational efficiency and improve overall customer experience.

The specific objectives of this project include:

- Automating repetitive operational tasks such as order notifications, loyalty status updates, and low-stock warnings.
- Ensuring accurate and consistent data entry through robust validation mechanisms.
- Delivering real-time access to essential business information, including stock availability, customer interaction history, and sales analytics.
- Strengthening collaboration among sales, inventory, and marketing teams.
- Creating personalized customer interactions through automated messaging and integrated loyalty programs.

Technology Description

The system was developed using several core Salesforce technologies, described as follows:

- **Salesforce Platform**

Salesforce functions as the central operational platform, providing a cloud-based CRM environment to manage customer data and business processes. The platform supports automation, customer service enhancement, marketing optimization, and sales performance improvement. Both no-code tools such as drag-and-drop features and advanced programming tools like Apex and Flow were utilized to accommodate different system requirements.

- **Custom Objects**

Custom objects were created to store and manage business-specific data, including:

- **Customer_c**: Stores customer profiles, including personal data, contact information, and loyalty status.
- **Order_c**: Records order transactions, linking customers with products while capturing order date, total cost, and quantity.
- **Product_c**: Contains detailed product data such as product name, description, pricing, and stock levels.
- **Inventory_c**: Maintains stock availability information, including stock quantity, status, warehouse location, and product linkage.
- **Marketing_Campaign_c**: Manages promotional campaigns and tracks marketing effectiveness.

- **Tabs and Custom App**

Custom navigation tabs were created for efficient data access. A dedicated Lightning App named “**HandsMen Threads**” serves as a centralized workspace, integrating all major modules, including Customer, Order, Product, Inventory, Campaign, and Reports, to support smooth operational workflows.

- **Profiles, Roles, and Permission Sets**

User access control was implemented through role-based profiles and permission sets:

- **Profiles** define user privileges, accessible objects, and interface layouts.
- **Roles** regulate data visibility based on organizational hierarchy.
- **Permission Sets** provide additional access rights without modifying core profile settings.

- **Validation Rules**

Validation rules were implemented to ensure data integrity, including:

- Preventing negative or zero transaction values.
- Blocking order submissions that result in insufficient stock.
- Enforcing proper email formatting.

- **Email Templates and Alerts**

Pre-designed email templates and automated alerts were developed to facilitate consistent communication, including order confirmations, low-stock notifications, and loyalty status updates.

- **Flows**

Salesforce Flow was used to automate business processes, such as:

- Automatically sending order confirmation emails.
- Updating inventory levels after order transactions.
- Running scheduled evaluations to update customer loyalty status.

- **Apex Triggers**

Apex triggers were implemented for advanced automation, including:

- Automatic calculation of order totals.
- Real-time deduction of inventory stock.
- Instant loyalty status updates based on purchase history.

Project Implementation

1. Salesforce Developer Environment Setup

A Salesforce Developer Org was configured as the primary development environment for system design and testing.

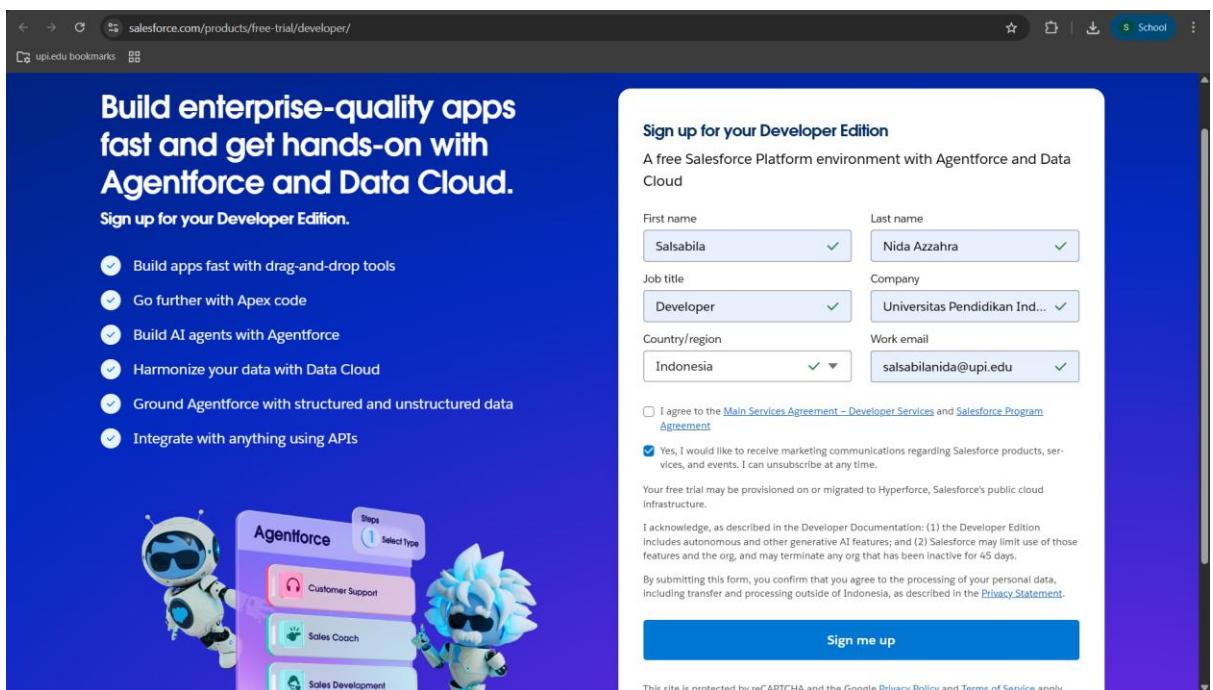


Figure 1. Creating a Salesforce Developer Org

2. Custom Object

The core data architecture was developed through the creation of custom objects and their associated fields to systematically store all critical business information, including:

- HandsMen Customer (Customer__c)
This object is designed to maintain detailed customer records, encompassing essential information such as Name, Email, Phone Number, Address, and Loyalty_Status__c.
- HandsMen Order (Order__c)
This object is responsible for managing customer order data, including Date, Quantity__c, and Total_Amount__c, while establishing relationships between orders, customers, and products.
- HandsMen Product (Product__c)
It stores comprehensive product details, such as Name, Description, Unit_Price__c, and Stock__c.
- Inventory (Inventory__c)
This object tracks stock availability, including Stock Quantity, Stock Status, Warehouse location, and its association with the corresponding HandsMen Product.
- Marketing Campaign (Marketing_Campaign__c)
This object is structured to document and manage marketing and promotional initiatives.

3. Custom Lightning Application

The “HandsMen Threads” Lightning App was designed to deliver a centralized and intuitive working environment for all teams. This application integrates key functional tabs, including Customer, Order, Product, Inventory, Campaign, and Reports, into a unified interface, enabling users to perform their tasks more efficiently, effectively, and seamlessly.

4. Validation Rules Configuration

To ensure data accuracy and consistency, a set of validation rules was implemented with the following objectives:

- HandsMen Order
Restricts the entry of Total_Amount__c values that are less than or equal to zero.
- Stock Quantity
Prevents Stock_Quantity__c values from being less than or equal to zero, thereby maintaining valid inventory records.
- Email
Enforces the inclusion of “@gmail.com” in the Email field to ensure that customer email addresses adhere to the specified format.

5. User Profile and Role Assignment

User access control and data visibility were governed through the configuration of customized profiles and role hierarchies aligned with organizational requirements:

- Profiles
The default Standard User profile was cloned and modified to create role-specific

profiles, such as the “HandsMen Sales Profile,” ensuring that each user group had appropriate permissions.

- Roles

A structured organizational role hierarchy was implemented, including positions such as Sales Manager and Inventory Manager, to control data access levels based on individual responsibilities and job functions.

6. User Creation

To demonstrate the system’s multi-user capabilities, multiple user accounts were created and assigned to appropriate profiles and roles, including Niklaus Mikaelson in the Sales role and Kol Mikaelson in the Inventory role.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter Expert	Chatter	chatty200dg5000003ja@ueas.fhc3601bpg@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	EPIC_OrgFarm	OEPIIC	oeic_a59649c749@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Mikaelson_Daniel	dmika	edumatters3654@gmail.com	Marketing	<input checked="" type="checkbox"/>	Platform_1
<input type="checkbox"/>	Mikaelson_Kol	kmika	edumatters35426@gmail.com	Inventory	<input type="checkbox"/>	Platform_1
<input type="checkbox"/>	Mikaelson_Niklaus	nmika	edumatters140804@gmail.com	Sales	<input checked="" type="checkbox"/>	Platform_1
<input type="checkbox"/>	Nida_Azzahra_Salsabila	sal	salsablananda.111301043cc6@agentforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	User_Integration	integ	integration@00dg5000003la@ueas.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User_Security	sec	Insightssecurity@00dg5000003je@ueas.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

Figure 2. User List

7. Email Template Development

Automated email templates were created for order confirmation, stock alerts, and loyalty program notifications.

8. Flow Automation

- Order Confirmation Flow (Record-Triggered Flow)

This automation flow is designed to automatically send an order confirmation email to customers immediately upon the creation of a new order record.

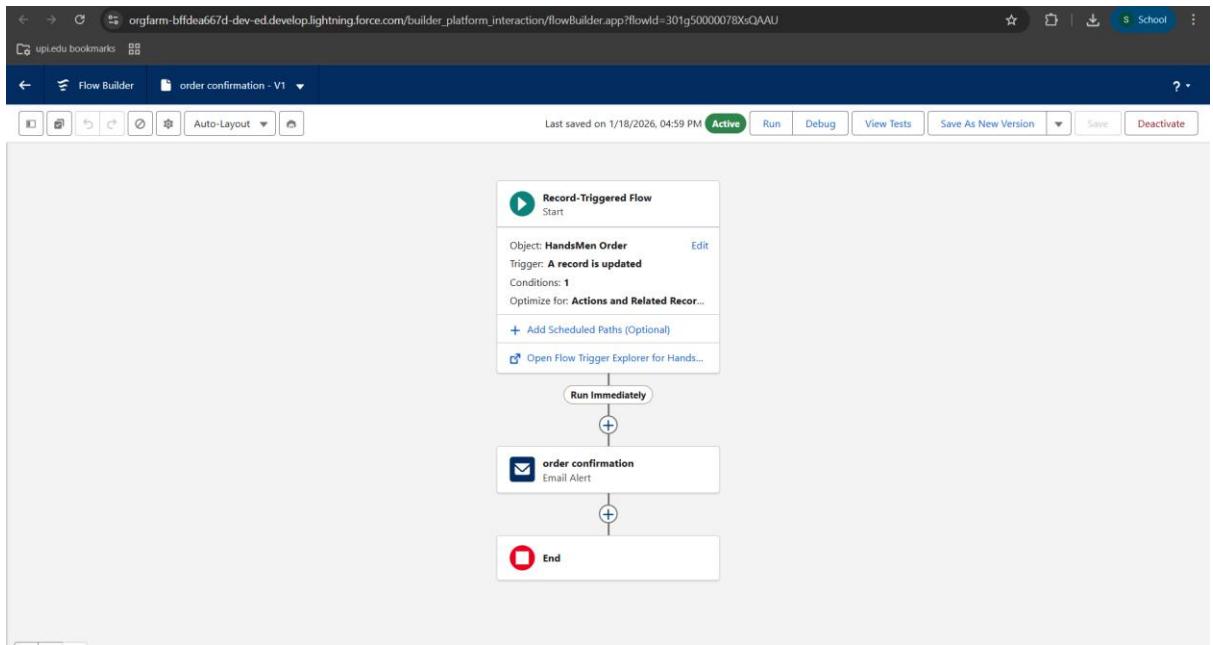


Figure 3. Order Confirmation Flow

- Stock Alert Flow (Record-Triggered Flow)

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

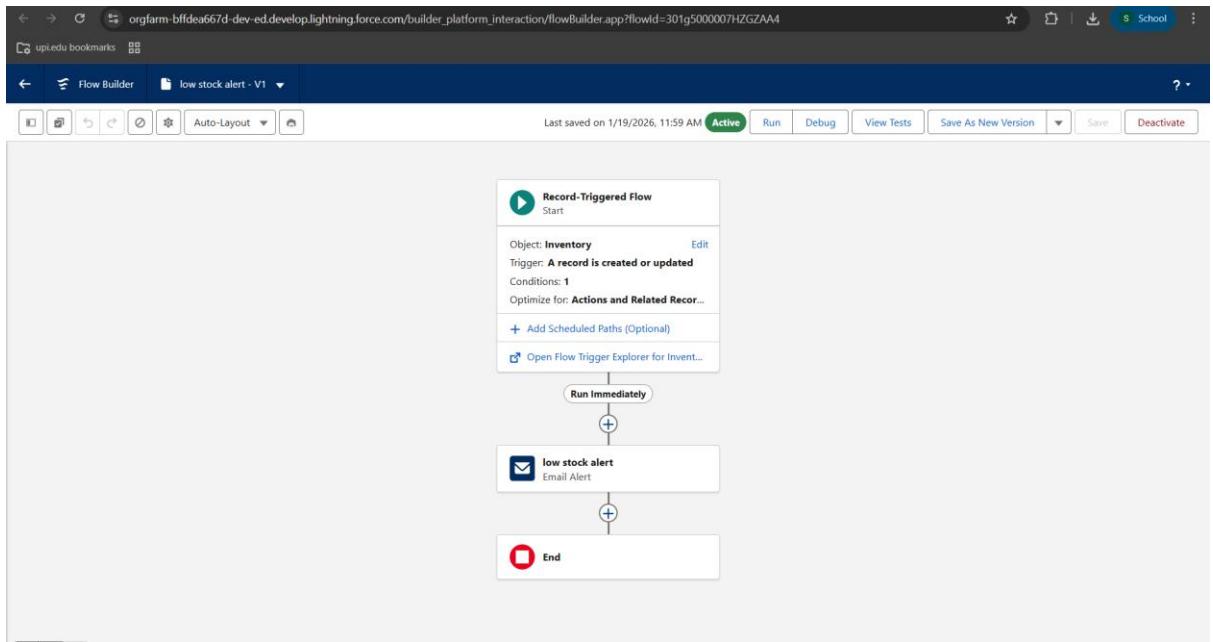


Figure 4. Stock Alert Flow

- Loyalty Program Flow (Scheduled-Triggered)

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

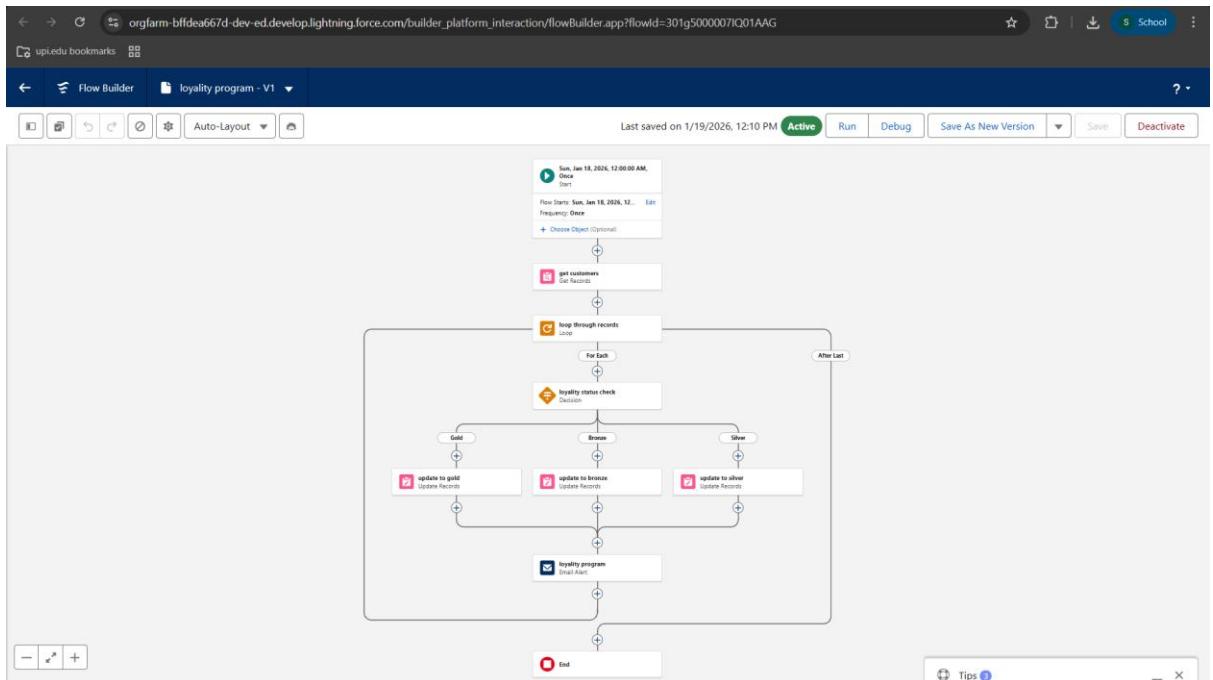


Figure 5. Loyalty Program Flow

9. Apex Trigger

- 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.

```

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;
19      }
20    }
21  }
22 }
  
```

User	Operation	Time	Status	Read	Size
Salsabila Nida Azzahra	/services/data/v65.0/tooling/executeA...	1/19/2026, 12:31:59 PM	Success	Unread	2.92 kB
Salsabila Nida Azzahra	/aura	1/19/2026, 12:28:43 PM	Success	Unread	13.73 kB
Salsabila Nida Azzahra	common.apisSoap.DirectSoap	1/19/2026, 12:28:43 PM	Success	Unread	536 bytes
Salsabila Nida Azzahra	/aura	1/19/2026, 12:28:10 PM	Success	Unread	8.35 kB
Salsabila Nida Azzahra	common.apisSoap.DirectSoap	1/19/2026, 12:28:09 PM	Success	Unread	536 bytes
Salsabila Nida Azzahra	/aura	1/19/2026, 12:26:56 PM	Success	Unread	1.49 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.

The screenshot shows the Salesforce IDE interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. Below the navigation bar, there are tabs for OrderTotalTrigger.apxc, StockDeductionTrigger.apxc, and InventoryBatchJob.apxc. The StockDeductionTrigger.apxc tab is active. The code editor contains the following Apex code:

```

16     WHERE HandsMen_Product__c IN :productIds]
17 );
18
19     List<Inventory__c> inventoriesToUpdate = new List<Inventory__c>();
20
21     for (HandsMen_Order__c order : Trigger.new) {
22         if (order.Status__c == 'Confirmed' && order.HandsMen_Product__c != null) {
23             for (Inventory__c inv : inventoryMap.values()) {
24                 if (inv.HandsMen_Product__c == order.HandsMen_Product__c) {
25                     inv.Stock_Quantity__c -= order.Quantity__c;
26                     inventoriesToUpdate.add(inv);
27                     break;
28                 }
29             }
30         }
31     }
32
33     if (!inventoriesToUpdate.isEmpty()) {
34         update inventoriesToUpdate;
35     }
36 }

```

Below the code editor is the Logs tab, which displays a table of log entries:

User	Application	Operation	Time	Status	Read	Size
Salsabila Nida Azzahra	Unknown	/services/data/v65.0/tooling/executeA...	1/19/2026, 12:31:59 PM	Success	Unread	2.92 kB
Salsabila Nida Azzahra	Browser	/aura	1/19/2026, 12:28:43 PM	Success	Unread	13.73 kB
Salsabila Nida Azzahra	Unknown	common.api.soap.DirectSoap	1/19/2026, 12:28:43 PM	Success	Unread	536 bytes
Salsabila Nida Azzahra	Browser	/aura	1/19/2026, 12:28:10 PM	Success	Unread	8.35 kB
Salsabila Nida Azzahra	Unknown	common.api.soap.DirectSoap	1/19/2026, 12:28:09 PM	Success	Unread	536 bytes
Salsabila Nida Azzahra	Browser	/aura	1/19/2026, 12:26:56 PM	Success	Unread	1.49 kB

At the bottom of the logs table is a "Filter Click here to filter the log list" input field.

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.

The screenshot shows the Salesforce IDE interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. Below the navigation bar, there are tabs for OrderTotalTrigger.apxc, StockDeductionTrigger.apxc, and InventoryBatchJob.apxc. The InventoryBatchJob.apxc tab is active. The code editor contains the following Apex code:

```

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        }
23
24    }
25
26    global void finish(Database.BatchableContext BC) {
27
28    }
29
30 }

```

Below the code editor is the Logs tab, which displays a table of log entries:

User	Application	Operation	Time	Status	Read	Size
Salsabila Nida Azzahra	Unknown	/services/data/v65.0/tooling/executeA...	1/19/2026, 12:31:59 PM	Success	Unread	2.92 kB
Salsabila Nida Azzahra	Browser	/aura	1/19/2026, 12:28:43 PM	Success	Unread	13.73 kB
Salsabila Nida Azzahra	Unknown	common.api.soap.DirectSoap	1/19/2026, 12:28:43 PM	Success	Unread	536 bytes
Salsabila Nida Azzahra	Browser	/aura	1/19/2026, 12:28:10 PM	Success	Unread	8.35 kB
Salsabila Nida Azzahra	Unknown	common.api.soap.DirectSoap	1/19/2026, 12:28:09 PM	Success	Unread	536 bytes
Salsabila Nida Azzahra	Browser	/aura	1/19/2026, 12:26:56 PM	Success	Unread	1.49 kB

At the bottom of the logs table is a "Filter Click here to filter the log list" input field.

Figure 8. Inventory Batch Job

Project Demonstration

The operational workflow of the system includes:

1. Customer Registration

New customers are registered and their data stored within the CRM system

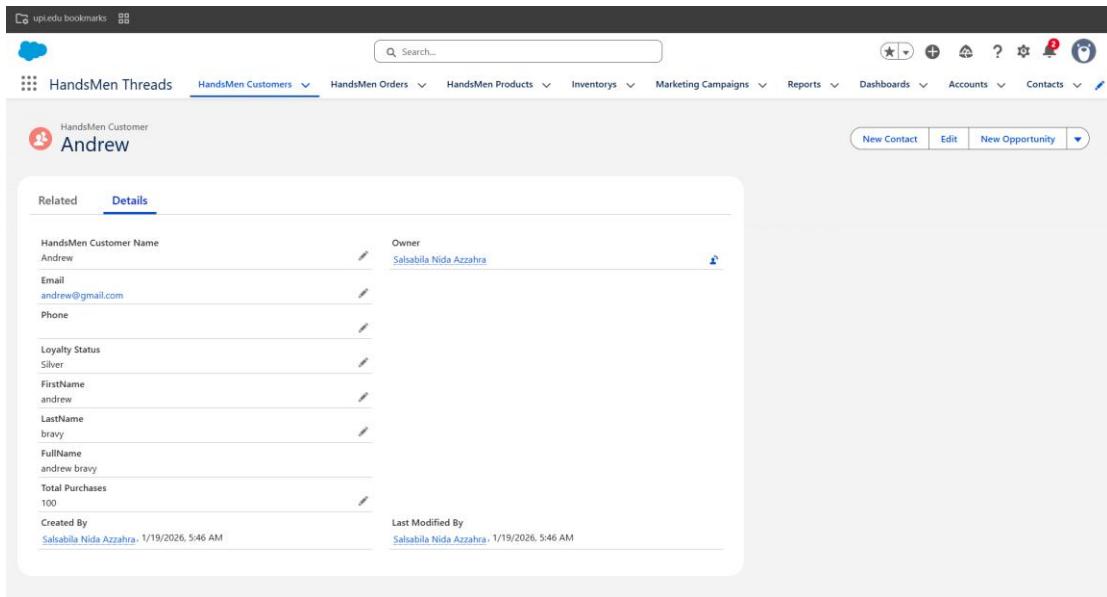


Figure 9. HandsMen Customer Registration

2. Product and Inventory Setup

Products and inventory records are configured, enabling automated stock tracking and low-stock alerts.

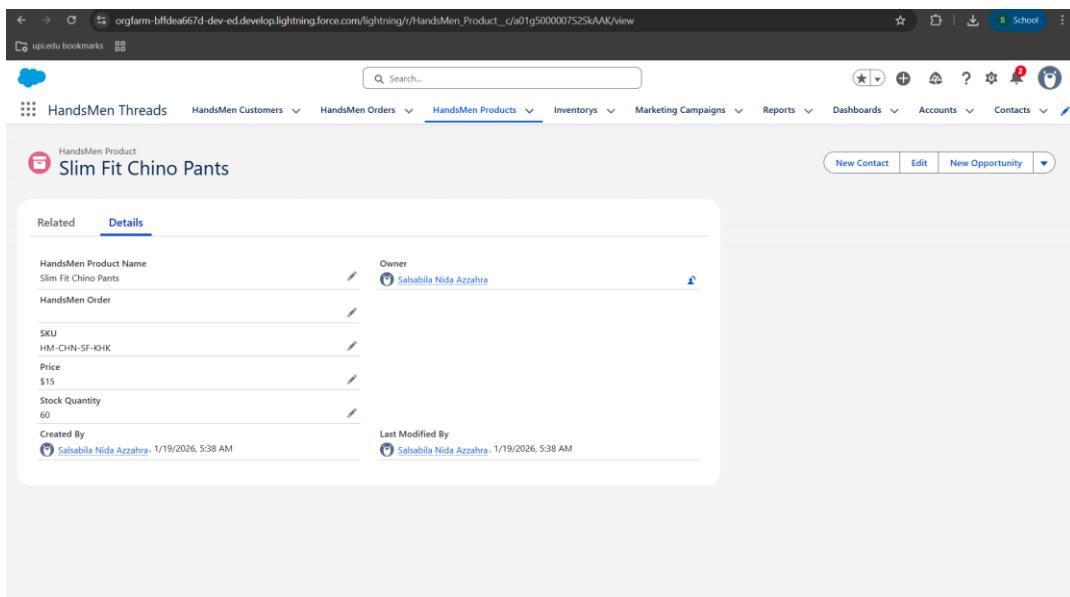


Figure 10. Detail HandsMen Product

3. Inventory Setup

Customer orders are created, validated, and confirmed, triggering automated email notifications and stock updates.

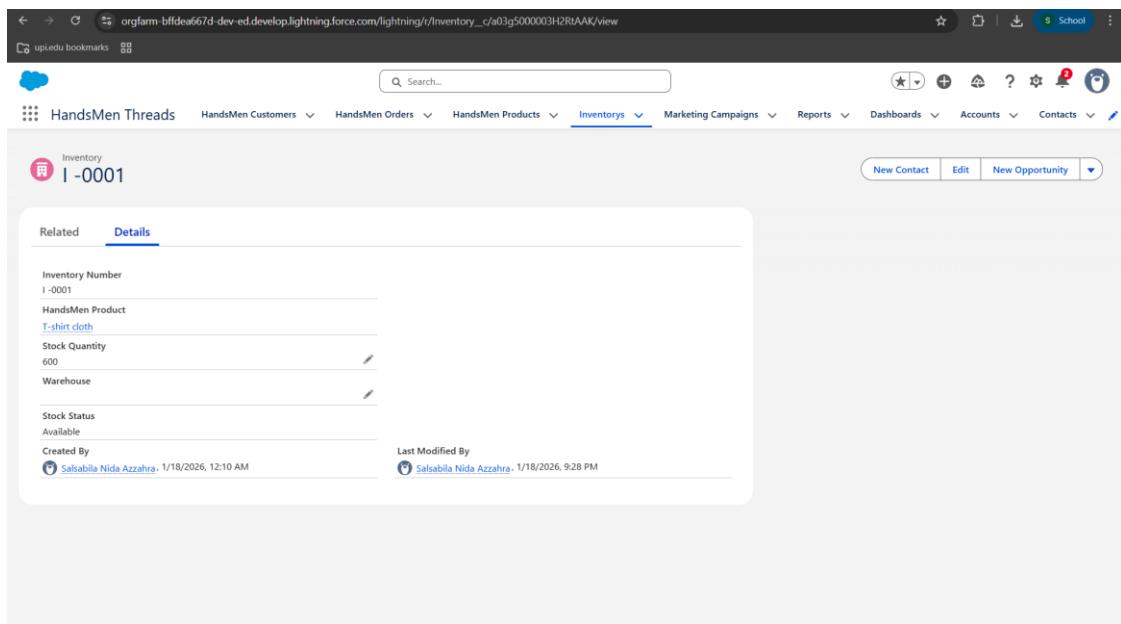


Figure 11. Inventorys Creation

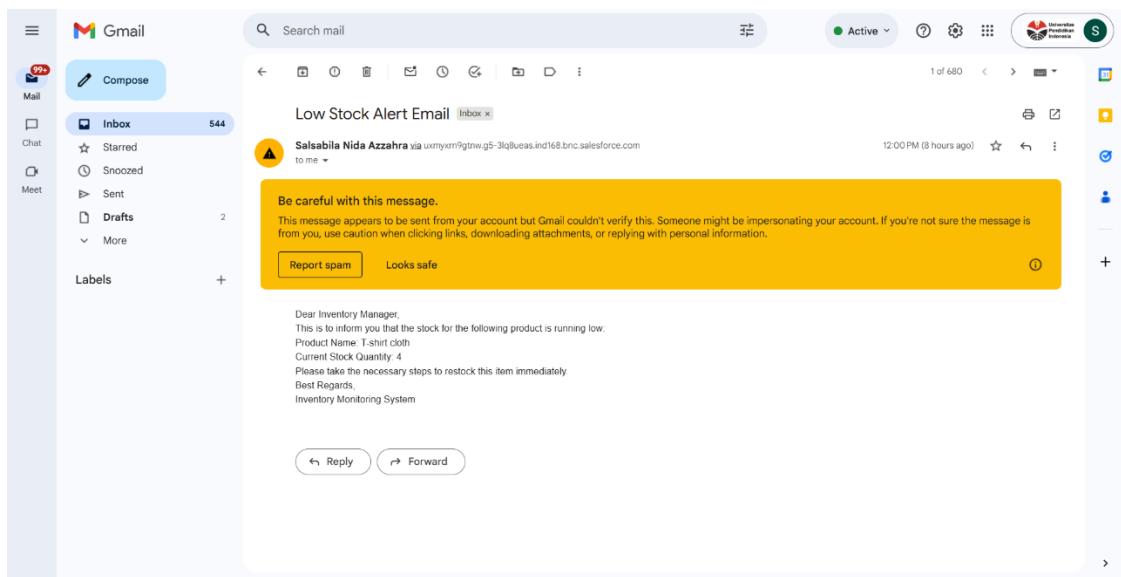


Figure 12. Low Stock Alert Email Notification

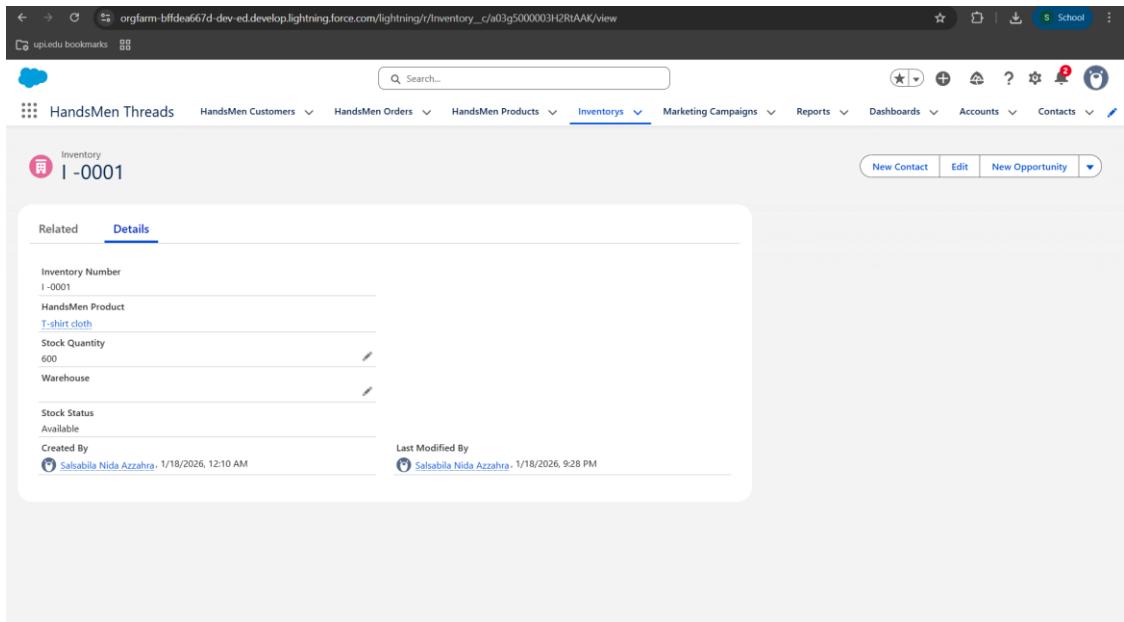


Figure 13. Detail Inventory

4. Order Setup

Customer loyalty levels are automatically evaluated and updated based on cumulative purchase value.

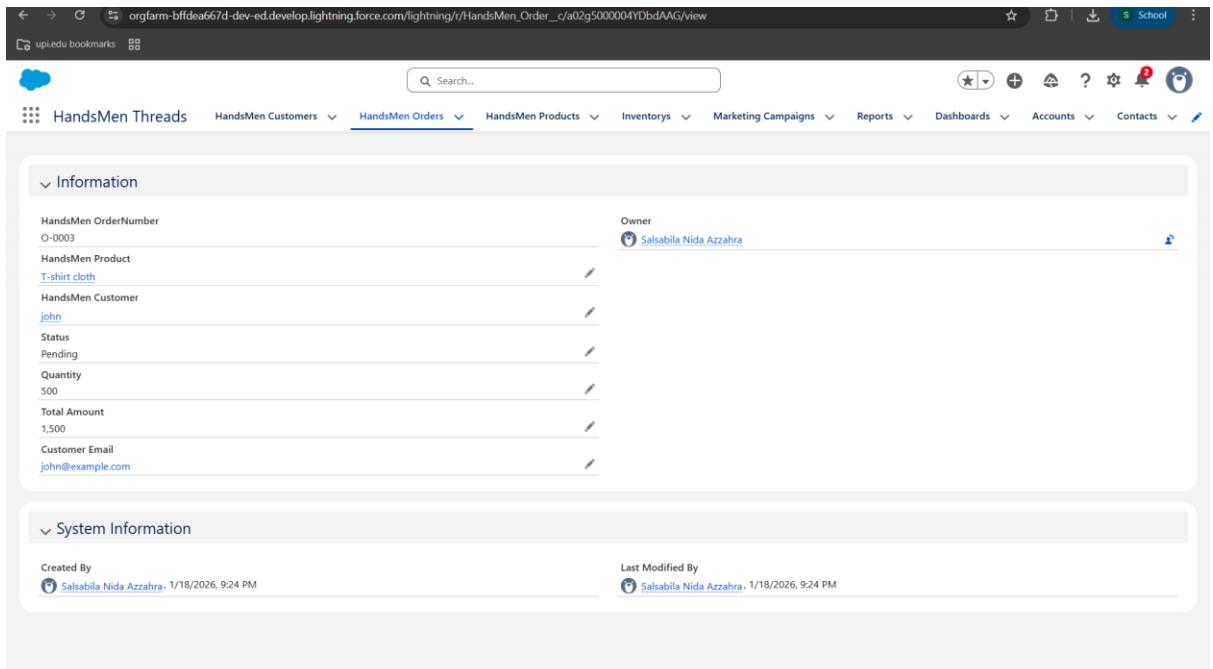


Figure 14. Order Setup

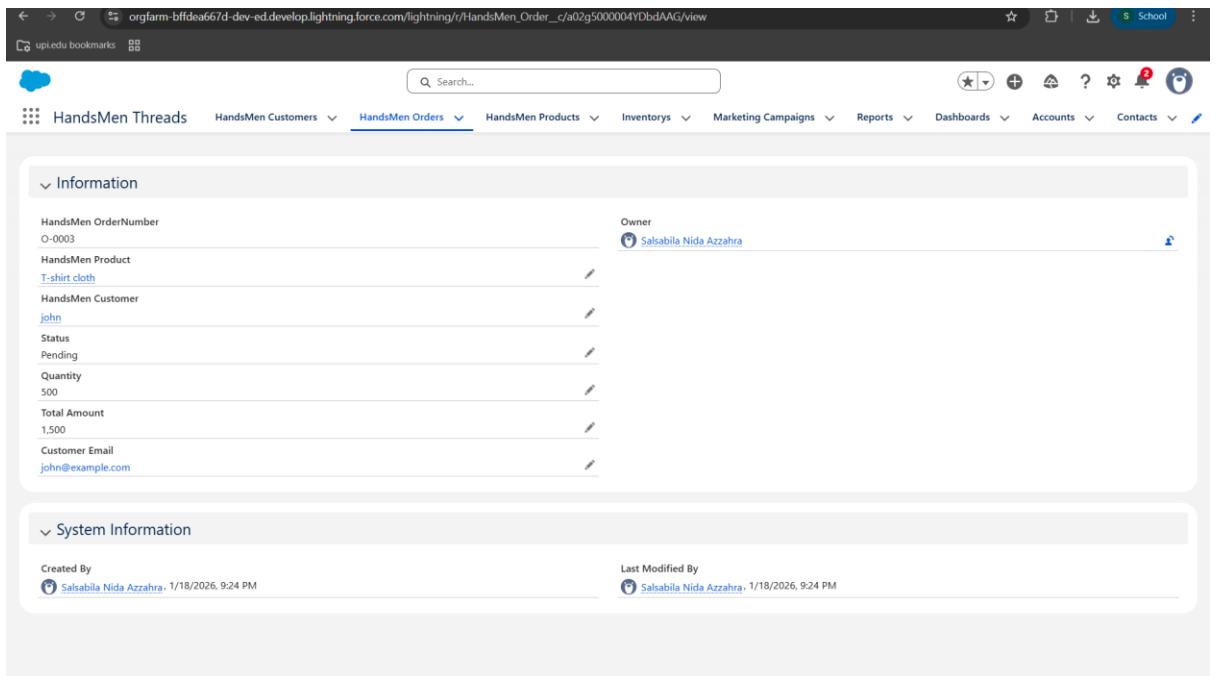


Figure 15. Update Order Status to Confirmed

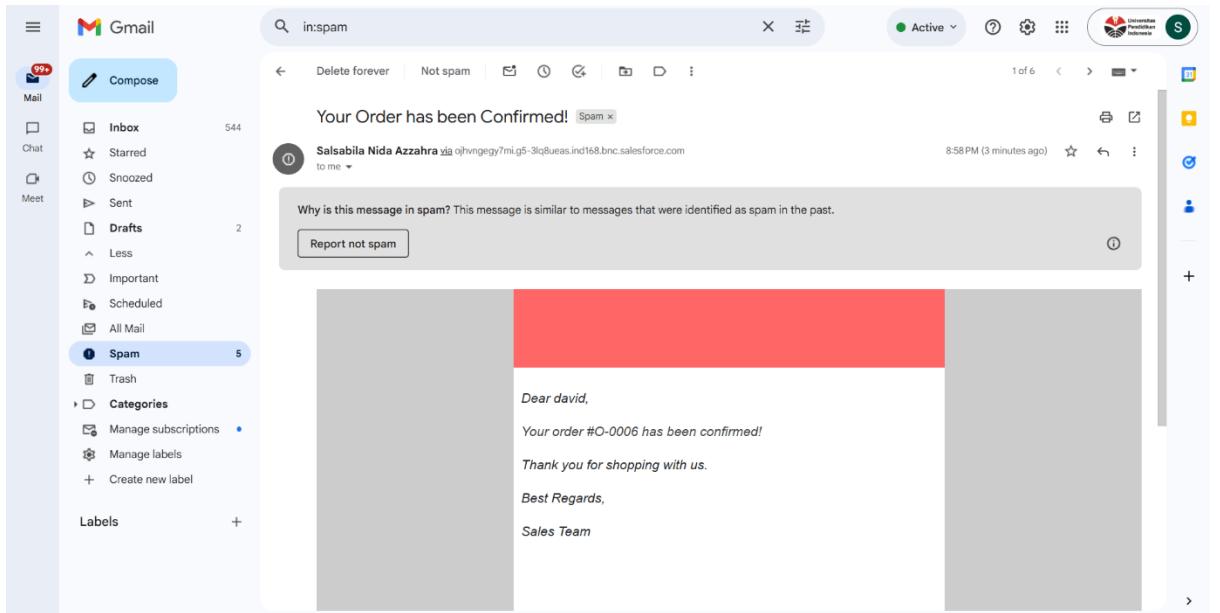


Figure 16. Order Confirmation Email Notification

5. Loyalty Status Update

Order totals and inventory deductions are processed automatically through backend triggers.

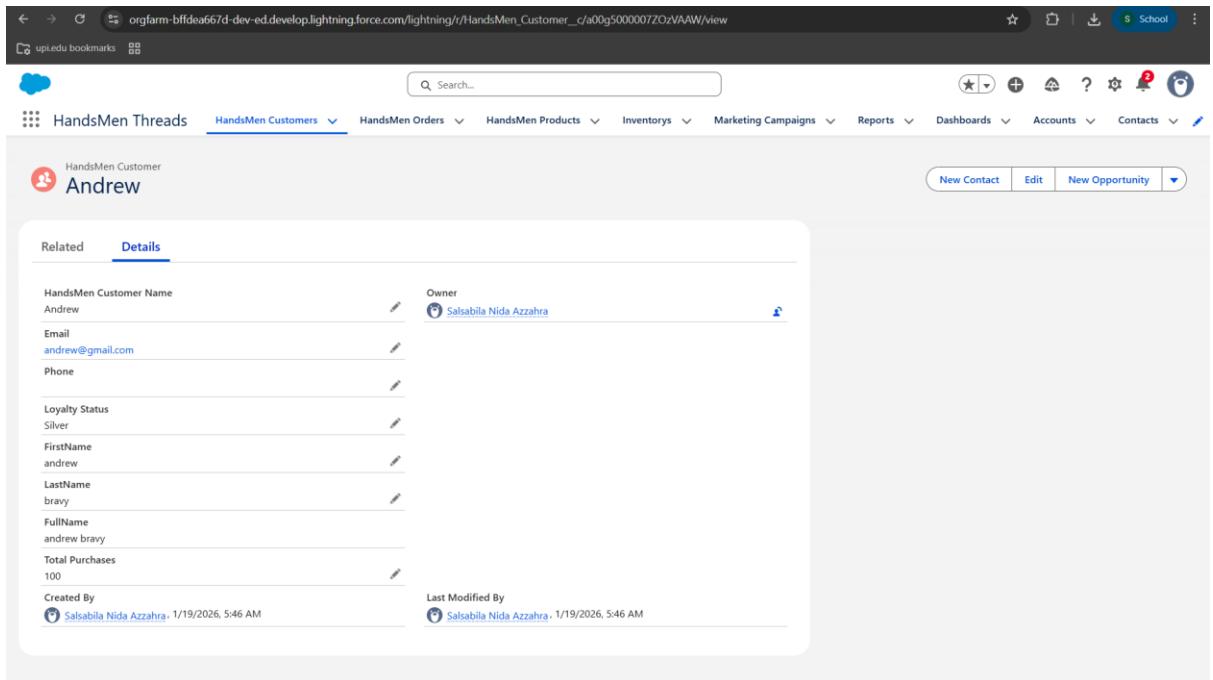


Figure 17. Update HandsMen Customer Data

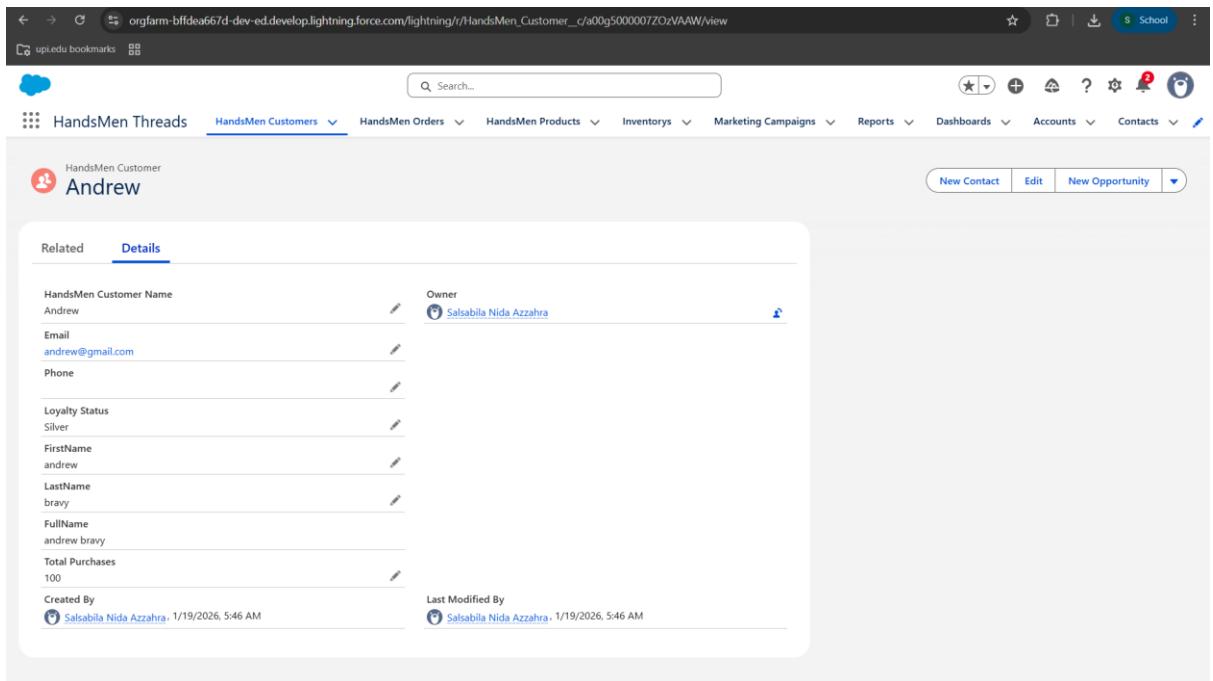


Figure 18. Update Loyalty Status

6. Apex Trigger Execution

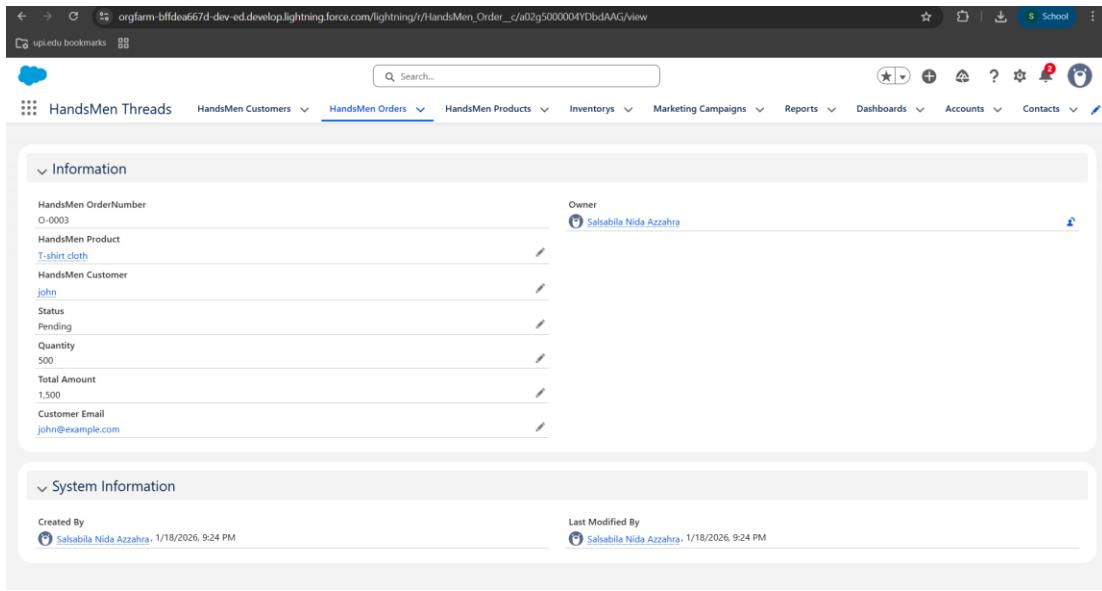


Figure 19. Total Amount in HandsMen Order

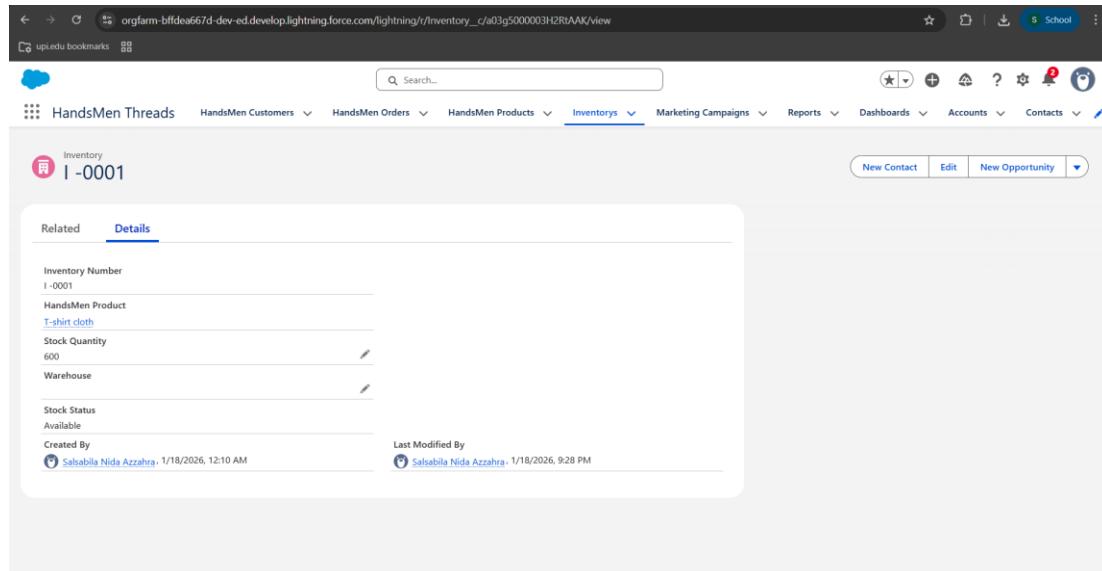


Figure 20. Stock Quantity in Inventory

Conclusion

Based on system design, development, testing, and implementation, it can be concluded that the Salesforce CRM solution developed for HandsMen Threads effectively delivers an integrated platform for business data management and workflow automation. The system successfully centralizes customer, order, product, and inventory information, resulting in improved operational efficiency, enhanced data accuracy, and streamlined business processes.

The integration of automation tools, validation rules, Salesforce Flows, and Apex triggers has significantly reduced manual workload while maintaining data integrity. Additionally, automated email notifications improve communication efficiency between customers and

internal teams. Overall, this system fulfills project objectives and establishes a strong foundation for future technological enhancements within the organization.