

Project Overview

HandsMen Threads is a men's fashion brand focused on delivering elegance and quality. To manage their growing demands and improve customer experiences, they decided to build a Salesforce CRM system. The goal is to organize their business data—such as orders, inventory, customer details, and more—in one place while automating key processes like order confirmations, loyalty programs, and stock alerts. The system is designed to help the company work smarter, faster, and more efficiently, ensuring that everything operates under trusted hands. can increase the material.

Objective

The primary objective of the CRM implementation is to **unify and manage critical business data**—including **customer profiles, product catalogs, inventory levels, orders, and marketing campaigns**—within a single, secure platform. This not only ensures data consistency but also enables better decision-making through real-time insights.

In addition to data centralization, the system also **automates key business processes** to reduce manual work and minimize errors. Some of the major automations include:

- **Order Confirmation Alerts:** Customers receive instant email confirmations upon successful order placement.
- **Loyalty Program Management:** Customer loyalty status is dynamically updated based on purchase history, and personalized emails are triggered for rewards and offers.
- **Inventory Stock Alerts:** Automatic notifications are sent when stock levels fall below threshold, helping inventory managers replenish items proactively.
- **Order Validation:** Ensures product availability before confirming orders, reducing fulfillment issues.

The CRM is built using a **modular architecture** involving custom objects like HandsMen_Customer__c, HandsMen_Product__c, HandsMen_Order__c, Inventory__c, and Marketing_Campaign__c. These are supported by **automation tools such as Apex triggers, batch jobs, flows, validation rules, and email templates**.

This solution not only enhances customer satisfaction and operational efficiency but also empowers different teams—such as Sales, Inventory, and Marketing—with the tools and data they need to perform at their best. Ultimately, the system ensures that **every transaction, every alert, and every interaction happen under trusted hands—HandsMen Threads**.

Phase 1: Requirement Analysis & Planning

Understanding the Business

The **HandsMen Threads** team is looking for a solution that reduces manual work and automates most business processes, minimizing the chances of errors and risks. They aim for improved tracking of stock levels, customer loyalty, and campaign performance. The team needs a system that can scale with growing demands, provide real-time alerts, and ensure faster updates as a top priority.

Scope & Key Features

- Design **custom objects** for **Products, Orders, Customers, Inventory, and Campaigns**.
- Set up **Flows, Formulas**, and **Validation Rules** to maintain accurate and reliable data as per business requirements.
- Implement **Automation** for **daily tasks** and **scheduled updates** to enhance efficiency and reduce manual effort.

Data & Security Design

1. Data Relationships:

Custom objects are connected through **lookup** and **master-detail relationships** to maintain data integrity and enforce business logic. For example, **validation rules** ensure that orders cannot be placed if the requested quantity exceeds available inventory.

2. Role Hierarchy:

Defined key roles such as **Sales, Inventory, and Marketing** to segregate responsibilities and support efficient collaboration while preventing potential operational conflicts.

3. Profile-Based Access Control:

Set up **permissions** and **field-level security** for each profile to control data visibility and edit access, ensuring that users only see and modify the data relevant to their roles.

Phase 2: Backend Setup & Automation

Object & Field Setup

Custom objects were created using **Object Manager** based on the specific business needs:

Custom Objects:

- HandsMen_Customer__c
- HandsMen_Product__c
- HandsMen_Order__c
- Inventory__c
- Marketing_Campaign__c

Fields Created (Examples):

❖ In HandsMen Customer:

The screenshot shows the 'Fields & Relationships' section of the HandsMen Customer object setup. It lists 11 fields, each with its field label, name, data type, controlling field, and indexed status. The fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
First Name	First_Name__c	Text(40)		
Full Name	Full_Name__c	Formula (Text)		
HandsMen Customer Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Last Name	Last_Name__c	Text(30)		
Loyalty Status	Loyalty_Status__c	Picklist		
Owner	OwnerId	Lookup(User/Group)		✓
Phone	Phone__c	Phone		
Total Purchases	Total_Purchases__c__c	Number(18, 0)		

❖ In HandsMen Product:

The screenshot shows the 'Fields & Relationships' section of the HandsMen Product object setup. It lists 8 fields, each with its field label, name, data type, controlling field, and indexed status. The fields are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
HandsMen Product Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Order	Order__c	Lookup(HandsMen Order)		✓
Owner	OwnerId	Lookup(User/Group)		✓
Price	Price__c	Currency(18, 0)		
SKU	SKU__c	Text(40)		
Stock Quantity	Stock_Quantity__c	Number(18, 0)		

❖ In HandsMen Order:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(HandsMen Customer)		
HandsMen Order Number	Name	Auto Number		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		
Quantity	Quantity__c	Number(10, 0)		
Status	Status__c	Picklist		
Total Amount	Total_Amount__c	Number(18, 0)		

❖ In Inventory:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Inventory Number	Name	Auto Number		
Last Modified By	LastModifiedById	Lookup(User)		
Product	HandsMen_Product__c	Master-Detail(HandsMen Product)		
Record Name	Record_Name__c	Auto Number		
Stock Quantity	Stock_Quantity__c	Number(18, 0)		
Stock Status	Stock_Status__c	Formula (Text)		
Warehouse	Warehouse__c	Text(30)		

❖ In Marketing Campaign:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Campaign Name	Name	Auto Number		
Created By	CreatedById	Lookup(User)		
End Date	End_Date__c	Date		
HandsMen Customer	HandsMen_Customer__c	Lookup(HandsMen Customer)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		
Start Date	Start_Date__c	Date		

Validation Rules Implemented:

- **HandsMen Order:**

Total_Amount_c <= 0

Error Message: "Please enter a correct amount."

The screenshot shows the 'HandsMen Order Validation Rule' setup page. The validation rule is named 'Total_Amount' and has the formula 'Total_Amount_c <= 0'. The error message is 'Please Enter Correct Amount.' The rule is active and located at the 'Total Amount' field. It was created by Shriyansh Negi on 7/9/2025, 11:28 AM and modified by him on 7/9/2025, 11:28 AM.

- **Inventory:**

Stock_Quantity_c <= 0

Error Message: "The inventory count must be greater than zero."

The screenshot shows the 'Inventory Validation Rule' setup page. The validation rule is named 'Stock_Quantity' and has the formula 'Stock_Quantity_c <= 0'. The error message is 'the inventory count is never less than zero.' The rule is active and located at the 'Top of Page'. It was created by Shriyansh Negi on 7/9/2025, 11:30 AM and modified by him on 7/9/2025, 11:33 AM.

- **HandsMen Customer:**

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

Error Message: "Please enter a valid Gmail address."

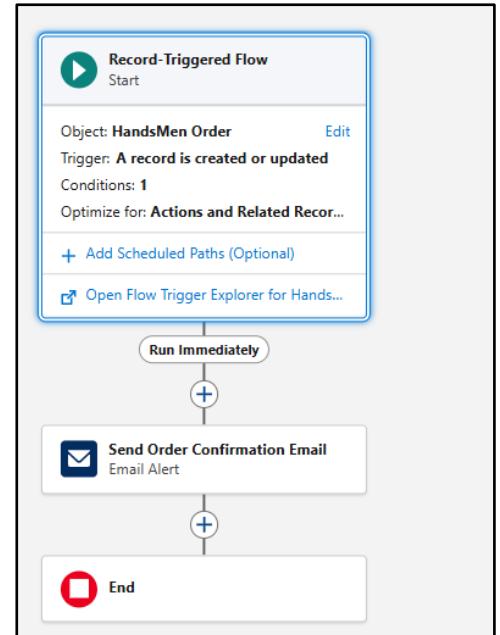
The screenshot shows the 'HandsMen Customer Validation Rule' setup page. The validation rule is named 'Email' and has the formula 'NOT CONTAINS>Email_c, '@gmail.com')'. The error message is 'Please fill correct Gmail'. The rule is active and located at the 'Top of Page'. It was created by Shriyansh Negi on 7/9/2025, 11:26 AM and modified by him on 7/9/2025, 11:34 AM.

Flows & Email Alerts

To automate key business processes and reduce manual effort, the following flows and email alerts were implemented:

1. Order Confirmation Flow

- Trigger:** When an order is updated and Status = Confirmed.
- Action:** Sends a confirmation email to the customer using the "Order Confirmation" email template.
- Purpose & Benefits:**
 - ✓ **Providing immediate assurance** to the customer that their order has been successfully received and processed.
 - ✓ **Eliminating delays** in manual follow-up by the sales or support team.
 - ✓ **Reducing human errors** and maintaining consistency in customer communication.
 - ✓ **Building trust and satisfaction**, which increases the chances of repeat business and customer loyalty.



Classic Email Templates

Custom Email Template
Order_Confirmation_Email

Preview your email template below.

Email Template Detail

Email Templates from: Salesforce	Unfilled Public Classic Email Templates
Email Template Name	Order_Confirmation_Email
Template Unique Name	Order_Confirmation_Email
Encoding	Unicode (UTF-8)
Author	Shriyansh Negi [Change]
Description	Created By Shriyansh Negi, 7/24/2025, 11:10 PM Modified By Shriyansh Negi, 7/24/2025, 11:10 PM

Email Template

Subject: Your Order has been Confirmed!

HTML Preview:

Dear {!HandsMen_Order__c.Customer__c},
Your order # {!HandsMen_Order__c.Name} has been confirmed!
Thank you for shopping with us.
Best Regards,
Sales Team

Plain Text Preview

Email Alerts

Email Alert
Order_Confirmation_Email_Alert

Rules Using This Email Alert

This alert is currently not used by any rules

Approval Processes Using This Email Alert

This alert is currently not used by any approval processes

Entitlement Processes Using This Email Alert

This alert is currently not used by any entitlement processes

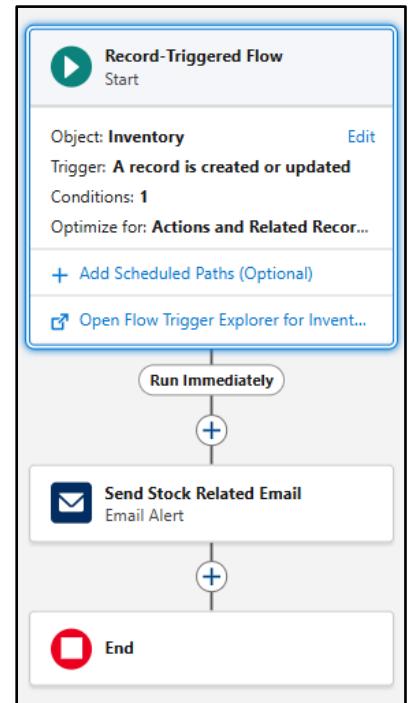
Flows Using This Email Alert

Flow Name	Version	Description	Object	Active
Order Confirmation Flow	1, 2		01gL0000015XM9	✓

Always show me ▾ more records per related list

2. Low Stock Alert Flow

- Trigger:** When Stock_Quantity__c < 5 in the **Inventory** object.
- Action:** Sends an email to the inventory management team using the "**Low Stock Alert**" email template.
- Purpose & Benefits:**
 - ✓ **Real-time visibility** into low inventory levels across all product lines.
 - ✓ **Faster response and restocking**, reducing the risk of lost sales or customer dissatisfaction.
 - ✓ **Minimization of manual monitoring**, allowing warehouse staff to focus on fulfillment and logistics.
 - ✓ **Support for just-in-time inventory strategies**, improving cash flow and reducing storage costs.
 - ✓ **Consistency and reliability** in inventory management practices.



SETUP Classic Email Templates

Custom Email Template
Low Stock Alert

Preview your email template below.

Email Template Detail					
Email Templates from Salesforce	Unified Public Classic Email Templates				
Email Template Name	Low Stock Alert	Available For Use	✓		
Template Unique Name	Low_Stock_Alert	Last Used Date			
Encoding	Unicode (UTF-8)	Times Used			
Author	Shriyansh Negi [Change]				
Description					
Created By	Shriyansh Negi, 7/24/2025, 11:38 PM	Modified By	Shriyansh Negi, 7/26/2025, 10:11 AM		
Edit Properties		Edit HTML Version	Edit Text Version	Delete	Clone

Email Template [Send Test and Verify Merge Fields](#)

Subject: Low Stock Alert: {!HandsMen_Product__c.Name}

HTML Preview

Dear Inventory Team,

This is a system alert regarding low inventory for the following product:

- Product Name: {!HandsMen_Product__c.Name}
- SKU: {!HandsMen_Product__c.SKU__c}
- Current Stock: {!HandsMen_Product__c.Stock_Quantity__c}

Please restock the item as soon as possible to avoid outages.

Regards,
Inventory Management System

[Plain Text Preview](#)

SETUP Email Alerts

Email Alert
For alerting the customer

[Rules Using This Email Alert](#) | [Approval Processes Using This Email Alert](#) | [Entitlement Processes Using This Email Alert](#)

Email Alert Detail			
Description	For alerting the customer		
Unique Name	Low_Stock_Email_Alert	Email Template	Low Stock Alert
From Email Address	Current User's email address	Object	Inventory
Recipients	Role: Inventory		
Additional Emails			
Created By	Shriyansh Negi, 7/10/2025, 3:58 AM	Modified By	Shriyansh Negi, 7/29/2025, 2:26 AM
Edit		Delete	Clone

Rules Using This Email Alert [Rules Using This Email Alert Help](#)

This alert is currently not used by any rules

Approval Processes Using This Email Alert [Approval Processes Using This Email Alert Help](#)

This alert is currently not used by any approval processes

Entitlement Processes Using This Email Alert

This alert is currently not used by any entitlement processes

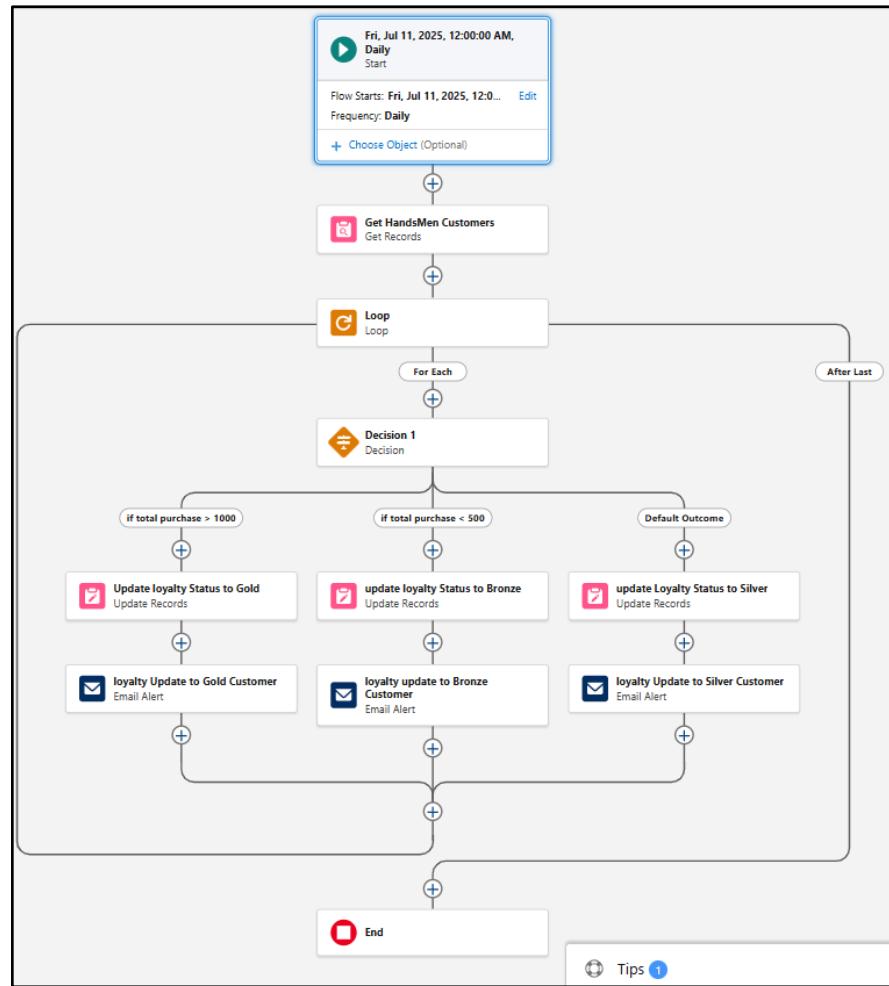
Flows Using This Email Alert

Flow Name	Version	Description	Object	Active
Stock Alert Flow	1, 2		01lgL0000015XNI	✓

[^ Back To Top](#) Always show me [more](#) records per related list

3. Loyalty Status Update Flow

- Type:** Scheduled Flow (runs daily).
- Action:** Updates the Loyalty_Status__c field in HandsMen_Customer__c based on their Total_Purchases__c.
- Purpose:** To keep customer loyalty tiers up to date without manual intervention. This helps in automating reward programs, targeted campaigns, and personalized offers.



Classic Email Templates

Custom Email Template Loyalty Program Email

Preview your email template below.

Email Template Detail	Edit Properties	Edit HTML Version	Edit Text Version	Delete	Clone	
Email Templates from Salesforce	Unified Public Classic Email Templates					
Email Template Name	Loyalty Program Email	Available For Use				
Template Unique Name	Loyalty_Program_Email	Last Used Date				
Encoding	Unicode (UTF-8)	Times Used				
Author	Shriyansh Negi [Change]					
Description						
Created By	Shriyansh Negi, 7/25/2025, 12:47 AM	Modified By	Shriyansh Negi, 7/29/2025, 12:54 AM			
		Edit Properties	Edit HTML Version	Edit Text Version	Delete	Clone

Email Template

Subject: Congratulations! You've Reached a New Loyalty Tier!

HTML Preview:

Dear {HandsMen_Customer__c.Full_Name__c},

We're excited to reward our valued customers with exclusive benefits through our new Loyalty Program!

Your current loyalty status is: {!HandsMen_Customer__c.Loyalty_Status__c}

As part of the program, you can earn points on every purchase and unlock amazing rewards.

Thank you for being a loyal customer!

Best regards,
HandsMen Threads Team

Email Alerts

Tells the loyalty to the customer

Rules Using This Email Alert | Approval Processes Using This Email Alert | Entitlement Processes Using This Email Alert

Email Alert Detail	Edit	Delete	Clone	
Description	Tells the loyalty to the customer			
Unique Name	Loyalty_Status			
From Email Address	Current User's email address			
Recipients	Email Field: Email			
Additional Emails				
Created By	Shriyansh Negi, 7/25/2025, 12:52 AM			
Modified By	Shriyansh Negi, 7/29/2025, 12:56 AM			
		Edit	Delete	Clone

Rules Using This Email Alert

This alert is currently not used by any rules

Approval Processes Using This Email Alert

This alert is currently not used by any approval processes

Entitlement Processes Using This Email Alert

This alert is currently not used by any entitlement processes

Flows Using This Email Alert

Flow Name	Version	Description	Object	Active
Loyalty_Status_Update_Flow	4, 4, 4, 5, 5, 5		HandsMen Customer	✓

^ Back To Top Always show me ▾ more records per related list

Apex Trigger & Class

To implement advanced logic that couldn't be handled through flows or validation rules, Apex Triggers and Classes were developed as follows:

Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.1%
You are currently using 5,955 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage [?](#)

Compile all classes [?](#)

View: All [Create New View](#)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other [All](#)

Action	Name +	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	InventoryBatchJob		64.0	Active	1,156	Shriyansh Negi, 7/28/2025, 10:12 AM	<input type="checkbox"/>
Edit Del	InventoryBatchJobTest		64.0	Active	1,526	Shriyansh Negi, 7/28/2025, 10:13 AM	<input type="checkbox"/>
Edit Del Security	LoyaltyStatusHandler		64.0	Active	561	Shriyansh Negi, 7/30/2025, 11:28 PM	<input type="checkbox"/>
Edit Del	LoyaltyStatusHandlerTest		64.0	Active	2,178	Shriyansh Negi, 7/30/2025, 10:27 AM	<input type="checkbox"/>
Edit Del Security	OrderTotalHandler		64.0	Active	896	Shriyansh Negi, 7/30/2025, 9:00 AM	<input type="checkbox"/>
Edit Del	OrderTotalHandlerTest		64.0	Active	1,784	Shriyansh Negi, 7/30/2025, 9:05 AM	<input type="checkbox"/>
Edit Del Security	OrderTriggerHandler		64.0	Active	1,014	Shriyansh Negi, 7/28/2025, 9:55 AM	<input type="checkbox"/>
Edit Del	OrderTriggerHandlerTest		64.0	Active	2,690	Shriyansh Negi, 7/28/2025, 10:00 AM	<input type="checkbox"/>
Edit Del Security	StockDeductionHandler		64.0	Active	1,230	Shriyansh Negi, 7/30/2025, 11:11 AM	<input type="checkbox"/>
Edit Del	StockDeductionHandlerTest		64.0	Active	861	Shriyansh Negi, 7/30/2025, 11:10 AM	<input type="checkbox"/>

Dynamic Apex Classes

Dynamic Apex extends your programming reach by interacting with Lightning Platform components.

Apex Triggers

This page allows you to view and modify all the triggers in your organization. To create a new trigger, navigate to the appropriate sObject triggers page.

Percent of Apex Used: 0.1%
You are currently using 5,955 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Compile all triggers [?](#)

View: All [Create New View](#)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R

Action	Name +	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By
Edit Del	CustomerLoyaltyTrigger		HandsMen_Customer	64.0	Active	229	Shriyansh Negi, 7/30/2025, 10:30 AM
Edit Del	HandsMenOrderTrigger		HandsMen_Order	64.0	Active	196	Shriyansh Negi, 7/30/2025, 9:01 AM
Edit Del	OrderTrigger		HandsMen_Order	64.0	Active	217	Shriyansh Negi, 7/28/2025, 10:40 AM
Edit Del	StockDeductionTrigger		HandsMen_Order	64.0	Active	456	Shriyansh Negi, 7/30/2025, 11:11 AM

Apex Jobs

Monitor the status of all Apex jobs, and optionally, abort jobs that are in progress.

Percent of Asynchronous Apex Used: 0%

You have currently used 2 asynchronous Apex operations out of an allowed 24-hour organization limit of 250,000. To learn about how this limit is calculated and what contributes to it, see the [Lightning Platform Apex Limits](#) topic.

View: All [Create New View](#)

Action	Submitted Date +	Job Type	Status	Status Detail	Total Batches	Batches Processed	Failures	Submitted By	Completion Date	Apex Class	Apex Method	Apex Job ID
	7/29/2025, 12:00 AM	Batch Apex	Completed		0	0	0	Negi_Shriyansh	7/29/2025, 12:00 AM	InventoryBatchJob		707gl00000APh4e
	7/28/2025, 10:31 AM	Batch Apex	Completed		1	1	0	Negi_Shriyansh	7/28/2025, 10:31 AM	InventoryBatchJob		707gl00000ALXIS
	7/28/2025, 10:22 AM	Scheduled Apex	Queued		0	0	0	Negi_Shriyansh		InventoryBatchJob		707gl00000ALZSP
	7/28/2025, 12:02 AM	Batch Apex	Completed		1	1	0	Negi_Shriyansh	7/28/2025, 12:02 AM	InventoryBatchJob		707gl00000AJlwD
	7/27/2025, 12:01 AM	Batch Apex	Completed		1	1	0	Negi_Shriyansh	7/27/2025, 12:01 AM	InventoryBatchJob		707gl00000AEQ2
	7/26/2025, 12:01 AM	Batch Apex	Completed		0	0	0	Negi_Shriyansh	7/26/2025, 12:01 AM	InventoryBatchJob		707gl00000AA3ot
	7/25/2025, 12:01 AM	Batch Apex	Completed		0	0	0	Negi_Shriyansh	7/25/2025, 12:02 AM	InventoryBatchJob		707gl00000A4uVo
	7/24/2025, 12:02 AM	Batch Apex	Completed		0	0	0	Negi_Shriyansh	7/24/2025, 12:02 AM	InventoryBatchJob		707gl000009xx4
	7/23/2025, 12:00 AM	Batch Apex	Completed		0	0	0	Negi_Shriyansh	7/23/2025, 12:00 AM	InventoryBatchJob		707gl000009qeSw

1. Apex Trigger: OrderTrigger on HandsMen_Order__c

This trigger is responsible for handling order validation logic during record insertions and updates. It delegates logic to a handler class for better code maintainability and scalability.

Apex trigger: -

```
trigger OrderTrigger on HandsMen_Order__c (before insert, before update) {
    if(Trigger.isBefore) {
        if(Trigger.isInsert || Trigger.isUpdate) {
            OrderTriggerHandler.validateOrderQuantities(Trigger.new);
        }
    }
}
```

2. Handler Class: OrderTriggerHandler

This class contains the business logic that checks the order quantity based on its status.

Logic: -

```
public class OrderTriggerHandler {
    public static void validateOrderQuantities(List<HandsMen_Order__c> orders) {
        for(HandsMen_Order__c order : orders) {
            if(order.Status__c == 'Confirmed' && order.Quantity__c > 500) {
                order.addError('Confirmed orders cannot exceed 500 units.');
            } else if(order.Status__c == 'Pending' && order.Quantity__c > 200) {
                order.addError('Pending orders cannot exceed 200 units.');
            } else if(order.Status__c == 'Rejected' && order.Quantity__c != 0) {
                order.addError('Rejected orders must have a quantity of 0.');
            }
        }
    }
}
```

Purpose:

To ensure business rules around order quantities are strictly enforced based on the order status, reducing manual errors and maintaining operational standards.

3. Batch Job: InventoryBatchJob

A scheduled batch class that automatically adds stock to products with low inventory.

Logic: -

```
global class InventoryBatchJob implements Database.Batchable<SObject> {
    global Database.QueryLocator start(Database.BatchableContext BC) {
        return Database.getQueryLocator([
            SELECT Id, Stock_Quantity__c
        ]);
    }
}
```

```

        FROM Inventory__c
        WHERE Stock_Quantity__c < 10
    ]);
}

global void execute(Database.BatchableContext BC, List<Inventory__c> scope) {
    for(Inventory__c inv : scope) {
        inv.Stock_Quantity__c += 50;
    }
    update scope;
}

global void finish(Database.BatchableContext BC) {}
}

```

Scheduled Daily at Midnight:

Logic: -

```
System.schedule('Daily Inventory Update', '0 0 0 * * ?', new InventoryBatchJob());
```

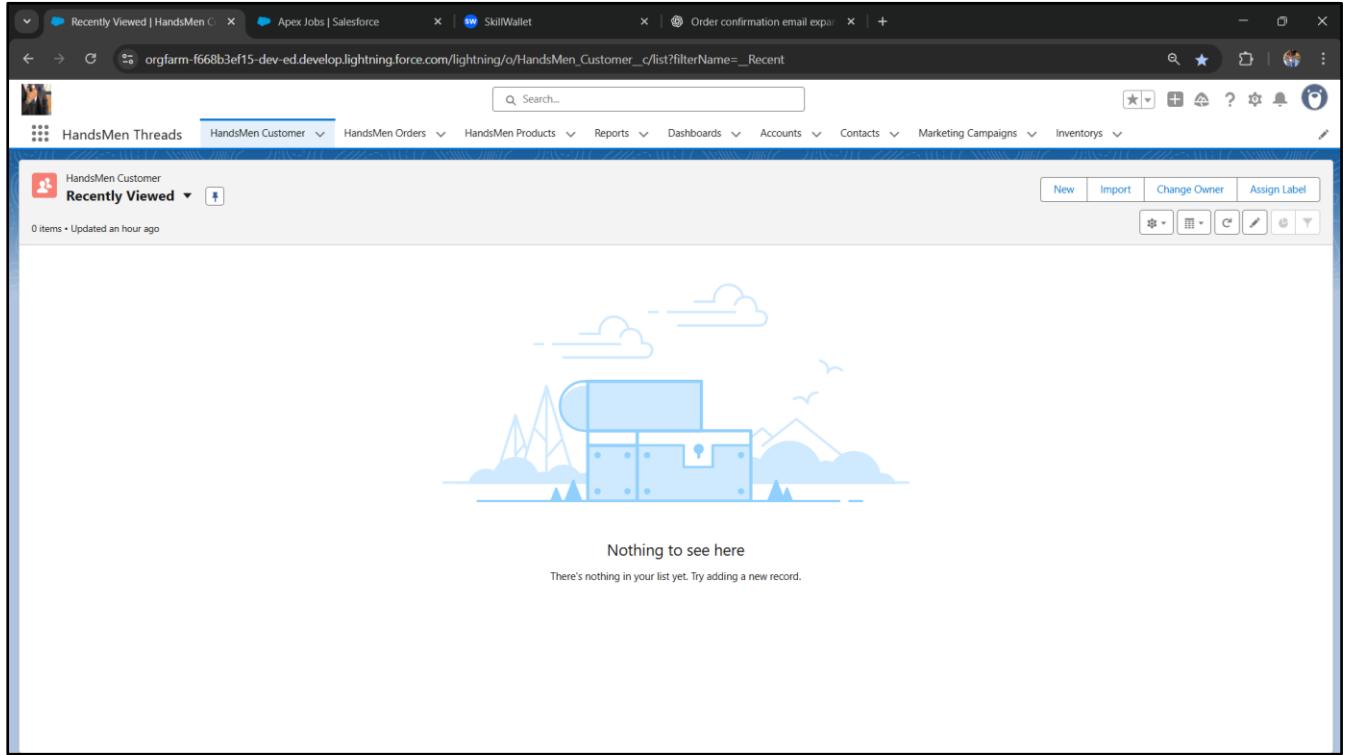
Purpose:

To maintain minimum stock levels by automatically replenishing inventory for items running low, thus avoiding stock shortages and improving operational efficiency.

Trigger Name	Object	Purpose
OrderTrigger	HandsMen_Order__c	Validate quantity based on status (Confirmed, Pending, Rejected)
Update Order Total	HandsMen_Order__c	Auto-calculate and update Total_Amount__c when order is created or edited
Stock Deduction	HandsMen_Order__c	Deduct stock when an order with Confirmed status is placed
Loyalty Status Update	HandsMen_Customer__c	Automatically upgrade customer loyalty level based on total purchases

Phase 3: UI/UX Setup & App Configuration

Lightning App Creation



- Created a custom Lightning App named "**HandsMen Threads**".
- **Navigation Tabs** included:
Customers, Orders, Products, Inventory, Campaigns, Reports, Dashboards, Accounts, Contacts.

User Roles & Permissions

1. Roles & Access

- **Sales**
→ Full access to Customers and Orders
- **Inventory**
→ Read & Edit access to Inventory and Products
- **Marketing**
→ Read access to Customers, Edit access to Marketing Campaigns

A screenshot of the Salesforce Setup Roles page. The title is "Creating the Role Hierarchy". It shows a tree structure of roles under "Your Organization's Role Hierarchy".

```
graph TD; Root[Your Organization's Role Hierarchy] --> GHEH[Graphic Era Hill University]; GHEH --> CEO[CEO]; GHEH --> COO[COO]; GHEH --> CFO[CFO]; GHEH --> Inv[Inventory]; GHEH --> Marketing[Marketing]; GHEH --> Sales[Sales]; GHEH --> SVCSS[SVP, Customer Service & Support]; GHEH --> CSR[Customer Support, International]; GHEH --> CSNA[Customer Support, North America]; GHEH --> IRS[Installation & Repair Services]; GHEH --> SHR[SVP, Human Resources]; GHEH --> SSM[SVP, Sales & Marketing]; GHEH --> VIPS[VP, International Sales]; CEO --> AddRole1[Add Role]; COO --> AddRole2[Add Role]; CFO --> AddRole3[Add Role]; Inv --> AddRole4[Add Role]; Marketing --> AddRole5[Add Role]; Sales --> AddRole6[Add Role]; SVCSS --> AddRole7[Add Role]; CSR --> AddRole8[Add Role]; CSNA --> AddRole9[Add Role]; IRS --> AddRole10[Add Role]; SHR --> AddRole11[Add Role]; SSM --> AddRole12[Add Role]; VIPS --> AddRole13[Add Role];
```

The "Add Role" buttons are located next to each role entry in the hierarchy.

2. Profile Used:

- a. *Platform User – "Platform 1" with object-level access.*

The screenshot shows the Salesforce Setup Profiles page. The profile is named "Platform 1". It has a "User License" of "Salesforce Platform" and was created by "Shriyansh.Negi" on 7/9/2025 at 12:05 PM. The "Profile Detail" section includes links for Edit, Clone, Delete, and View Users. The "Page Layouts" section lists various standard object layouts for different objects like Global, Lead, Location, etc., each with a "View Assignment" link. The "Custom Profile" checkbox is checked.

3. Permission Sets:

- Used to assign specific object-level access.
- **CRUD access** granted as needed:
 - **SalesAccess** → Full on Customers, Orders
 - **InventoryAccess** → Read/Edit on Inventory, Products
 - **MarketingAccess** → Read on Customers, Edit on Campaigns
- Ensures flexible, role-based security without modifying profiles.

The screenshot shows the Salesforce Setup Users page. The "Active Users" section lists several users with their full names, aliases, and usernames. The users include Chatter Expert, EPIC_OrgFarm, Mikaelson Kol, Mikaelson Niklaus, Negi Abhimanyu, Negi Shriyansh, User Integration, and User Security. Each user has a checkbox and an "Edit" link. The table also includes columns for Role, Active status, and Profile, with checkboxes indicating active status for most roles and profiles. The bottom of the page shows navigation links for letters A-Z and a "All" link.

Phase 4: Testing & Security

Testing Process

- **Unit tests** have been written for all Apex classes and triggers to achieve the required **code coverage**;
 - ✓ for example, **InventoryBatchJobTest** for testing the **InventoryBatchJob** Apex class and **OrderTriggerHandlerTest** for testing the **OrderTriggerHandler** Apex class.

The screenshot shows the Salesforce Setup interface with the 'Apex Classes' tab selected. The page displays information about Apex code usage and a list of classes. A green banner at the top indicates 'Percent of Apex Used: 0.04%' and notes character usage. Below this, there are buttons for estimating coverage, compiling classes, and creating new views. The main table lists four classes:

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By
Edit Del Security	InventoryBatchJob		64.0	Active	1,156	Shriyansh Negi. 7/28/2025, 10:12 AM
Edit Del	InventoryBatchJobTest		64.0	Active	1,526	Shriyansh Negi. 7/28/2025, 10:13 AM
Edit Del Security	OrderTriggerHandler		64.0	Active	1,014	Shriyansh Negi. 7/28/2025, 9:55 AM
Edit Del	OrderTriggerHandlerTest		64.0	Active	2,690	Shriyansh Negi. 7/28/2025, 10:00 AM

- **Flows** tested using various scenarios to verify all logical branches.
- **Validation Rules** tested by inputting incorrect data to confirm proper error handling.
- **Email Alerts** verified using test email addresses to ensure correct delivery and template usage.

Security Features

- Defined **Role Hierarchies** for secure data access based on user responsibilities.
- Applied **Field-Level Security (FLS)** based on user profiles.
- Enabled **Duplicate Rules** for Email (on Customers) and SKU (on Products) to avoid redundant data.

Phase 5: Deployment & Maintenance

Deployment Strategy

- Used **Change Sets** to migrate metadata and configuration to the production environment.
- Components deployed included:
Custom Objects, Fields, Flows, Triggers, Email Templates, Validation Rules, and Profiles.

Scheduled Jobs

- InventoryBatchJob scheduled to run **daily at midnight** to automatically replenish low stock.

Troubleshooting & Monitoring

- Enabled **Debug Logs** for active monitoring and issue tracking.
- Configured **Error Email Alerts** to notify admins on flow or trigger failures.

Conclusion

The **Salesforce CRM system** for **HandsMen Threads** has successfully unified the organization's **Sales, Inventory, and Marketing** operations.

Key benefits include:

- Reduced manual work and human error.
- Improved customer experience through faster response and automation.
- Better data visibility, aiding smart decision-making.
- Scalability for future business growth.

This implementation provides a solid foundation for future enhancements in automation and intelligence.