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

HandsMen Threads: Salesforce Automation for Premium Men’s Fashion Operations

---

Project Overview

HandsMen Threads, a growing organization in the premium men’s fashion industry, is undertaking a Salesforce-based digital transformation to modernize its data management and strengthen customer relationships. The objective of this project is to design and implement a robust Salesforce solution that centralizes business data and enables a smooth, reliable flow of information across all departments.

A core focus of the implementation is maintaining **data integrity directly at the user interface (UI) level**. By enforcing validations, structured data entry, and controlled automation, the system ensures that all business data remains accurate, consistent, and trustworthy—supporting better decision-making and dependable day-to-day operations.

To improve customer experience and operational efficiency, the project integrates several automation-driven business processes into the workflow:

• **Automated Order Confirmations** – Customers automatically receive confirmation emails once an order is finalized, improving transparency and engagement.

• **Dynamic Loyalty Program** – Customer loyalty status is updated based on purchase history, enabling personalized rewards and encouraging repeat business.

• **Proactive Stock Alerts** – When inventory levels fall below a defined threshold, automated email alerts notify the warehouse team to ensure timely restocking and prevent stockouts.

• **Scheduled Bulk Order Processing** – The system processes bulk orders at scheduled intervals, updating financial records and inventory levels to maintain accurate daily stock visibility.

Through this project, Salesforce is used not only as a CRM platform but as an **end-to-end business automation system** that supports scalability, operational efficiency, and a premium customer experience.

Objectives

• **Centralize Business Operations**
Create a single Salesforce platform to manage customers, products, orders, inventory, and campaigns.

• **Automate Order & Inventory Processes**
Reduce manual intervention by automating order confirmations, stock monitoring, and inventory updates.

• **Enable Loyalty-Based Personalization**
Automatically update customer loyalty status based on purchase history to support personalized engagement.

• **Improve Data Accuracy & Integrity**
Enforce validations, formula fields, and controlled data entry at the UI level.

• **Ensure Secure Role-Based Access**
Protect sensitive data using Salesforce roles, profiles, permission sets, and sharing rules.

• **Build a Scalable Architecture**
Design the system to support future growth, analytics, and advanced automation.

Key Features

• Custom objects for customers, products, orders, inventory, and marketing campaigns
• Automated order confirmation emails
• Real-time inventory tracking with low-stock alerts
• Loyalty status automation using scheduled processes
• Apex triggers for enforcing complex business rules
• Batch and scheduled Apex for bulk and time-based operations
• Secure data access using Salesforce security model

System Requirements

Hardware Requirements

● Computer with minimum 4 GB RAM
● Dual-core processor or higher
● Stable internet connection

Software Requirements

- Salesforce Developer Edition Org
- Modern Web Browser (Google Chrome / Mozilla Firefox)

Skills Required

- Salesforce Platform Fundamentals
- Salesforce Data Modeling and Relationships
- Salesforce Security (Roles, Profiles, Permission Sets)
- Flow Builder and Automation Concepts
- Apex Programming and Triggers
- Batch Apex and Scheduled Apex
- Reporting and Dashboards

Milestone 1: Salesforce Account Setup

Introduction

Salesforce is a leading cloud-based platform used to build scalable CRM and business automation solutions. In the HandsMen Threads project, Salesforce serves as the foundation for managing customer relationships, automating business workflows, and ensuring secure data handling.

This milestone focuses on setting up a **Salesforce Developer Edition account**, which provides a free and fully functional environment for configuration, automation, and Apex development. This account is used throughout the project lifecycle for building and testing the solution.

What Is Salesforce?

Salesforce is a customer success platform that helps organizations sell, service, market, analyze, and connect with their customers. It offers powerful tools for data management, automation, security, reporting, and application development—all delivered through the cloud.

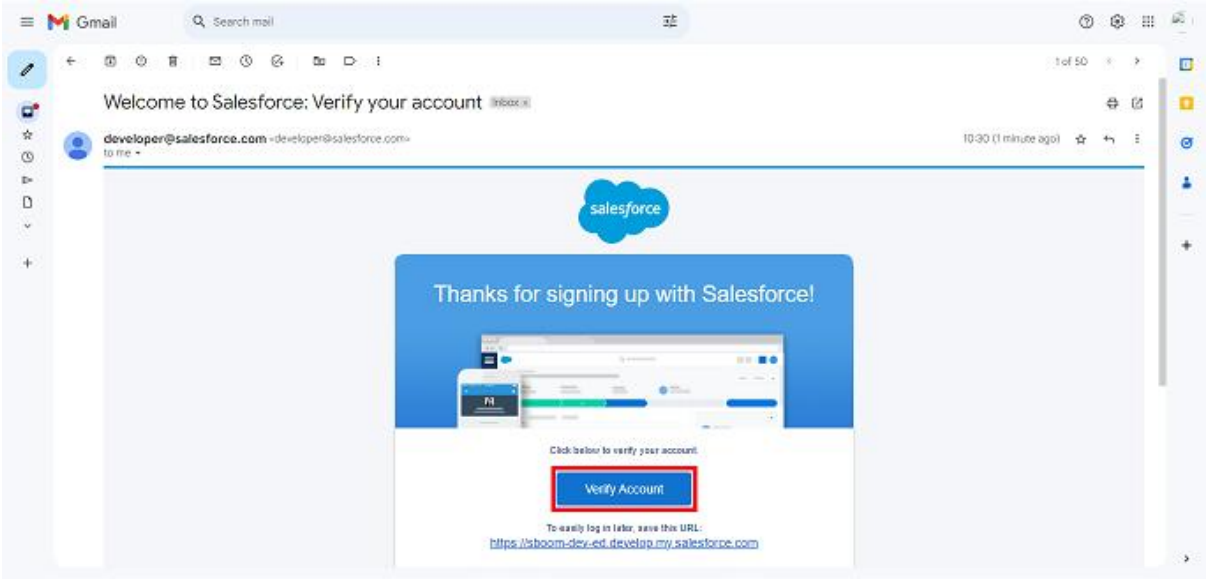
Activity 1: Creating a Salesforce Developer Account

Objective:
To create a Salesforce Developer Edition account that will be used for development and testing.

Steps:

1. Open a web browser and navigate to: https://developer.salesforce.com/signup
2. Fill in the registration form with the following details:
    - First Name & Last Name
    - Email Address
    - Job Title: Developer
    - Company: College Name
    - Country: India
    - Postal Code
    - Username: A unique username in the format name@organization.com

3. Click **Sign Me Up** after completing the form.



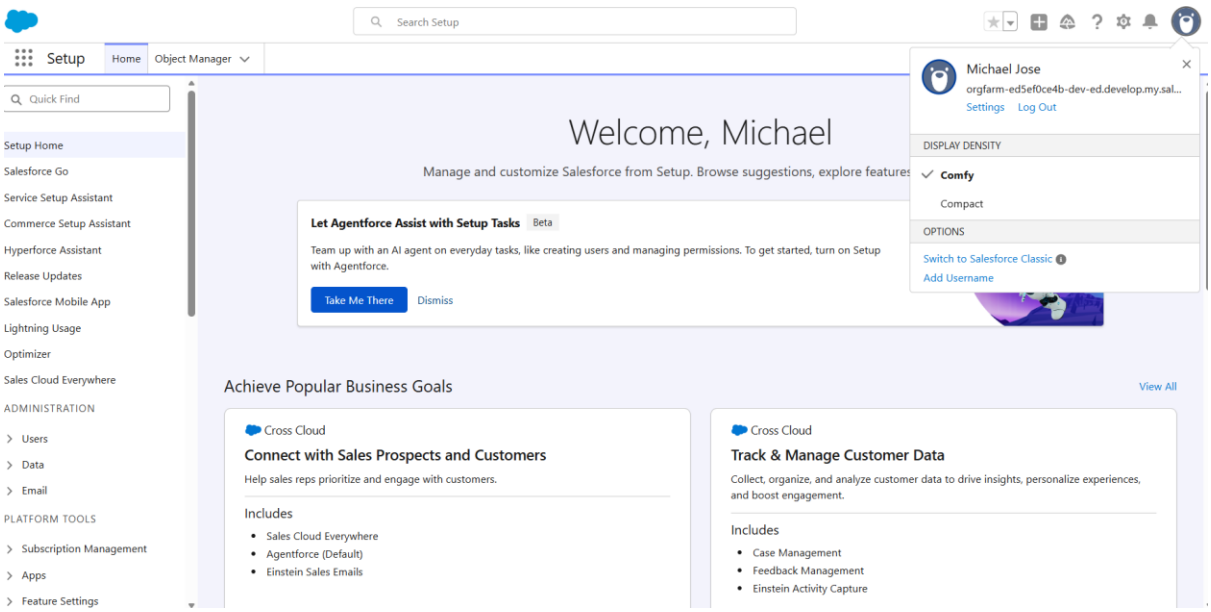
Activity 2: Account Activation

Objective:

To activate the Salesforce Developer Edition account and access the setup environment.

Steps:

1. Open the inbox of the email address used during registration.
2. Click the **Verify Account** link received from Salesforce.
3. Set a password and select a security question.
4. After successful activation, you will be redirected to the Salesforce Setup page.



Outcome:

A fully activated Salesforce Developer Edition account, ready for HandsMen Threads project implementation.

Milestone 2: Object Creation & Data Modeling

Introduction

In Salesforce, objects represent database tables used to store business data. For the HandsMen Threads project, custom objects are created to accurately model real-world entities such as customers, products, orders, inventory, and marketing campaigns. A well-structured object model is essential for enabling automation, maintaining data integrity, and supporting reporting and analytics.

This milestone focuses on navigating the Salesforce Setup environment and creating the core custom objects required for the HandsMen Threads Salesforce Automation System.

Navigating to Salesforce Setup

To access the configuration environment in Salesforce:

1. Click on the **Gear icon (⚙)** located at the top-right corner of the Salesforce interface.
2. Select **Setup** from the dropdown menu.

The Setup page provides access to Object Manager, automation tools, security settings, and development features.

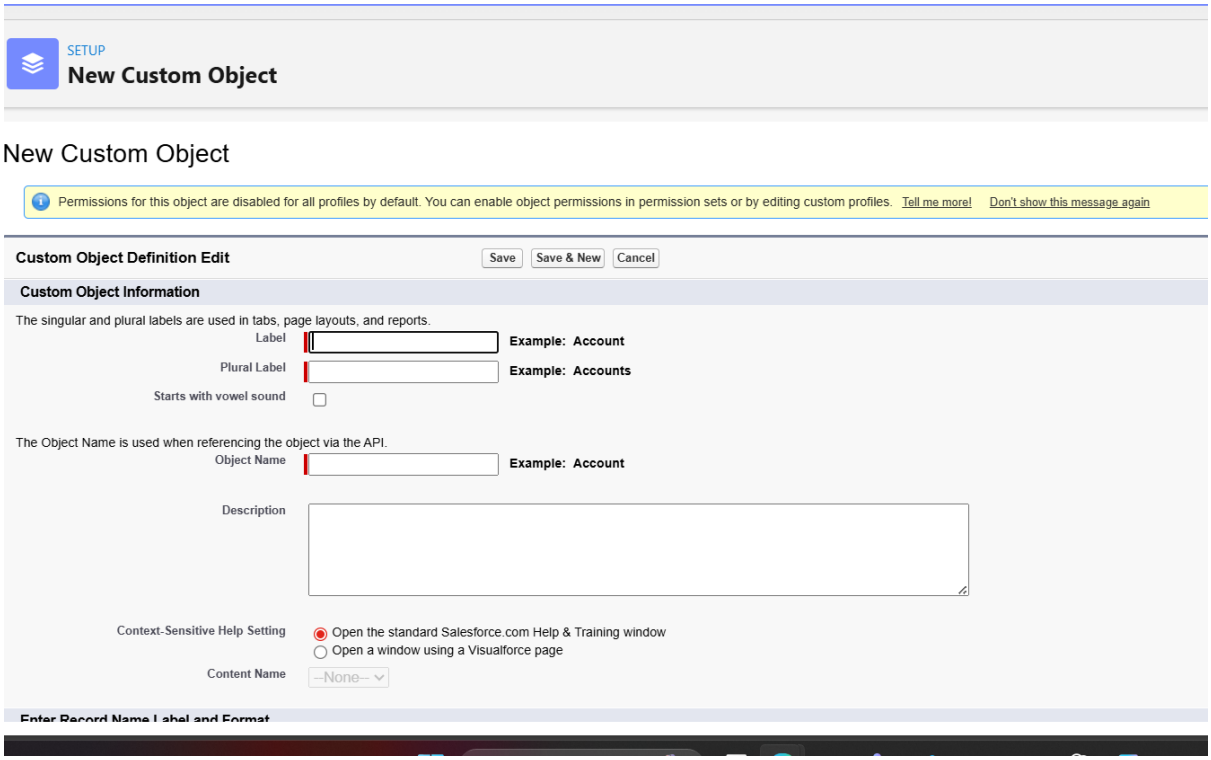
Activity 1: Create HandsMen Customer Object

Purpose:
The HandsMen Customer object is used to store customer-related information and serves as the foundation for order management and loyalty tracking.

Steps to Create the Object:

1. From the Setup page, click on **Object Manager**.
2. Click **Create** → **Custom Object**.
3. Enter the following details:
    - **Label:** HandsMen Customer
    - **Plural Label:** HandsMen Customers
    - **Record Name:** HandsMen Customer Name
    - **Data Type:** Text

4. Enable the following options:
    - ☑ Allow Reports
    - ☑ Allow Search
5. Click **Save**.



Activity 2: Create HandsMen Product Object

Purpose:

The HandsMen Product object stores information about the product catalogue, including items offered by the brand and their pricing and stock details.

Steps to Create the Object:

1. Go to **Setup** → **Object Manager**.
2. Click **Create** → **Custom Object**.

3. Enter the following details:
    - **Label:** HandsMen Product
    - **Plural Label:** HandsMen Products
    - **Record Name:** HandsMen Product Name
    - **Data Type:** Text
4. Enable:
    - ☑ Allow Reports
    - ☑ Allow Search
5. Click **Save**.

Activity 3: Create HandsMen Order Object

Purpose:
The HandsMen Order object is used to track customer orders and their lifecycle, including status, quantity, and order identification.

Steps to Create the Object:

1. Navigate to **Setup** → **Object Manager**.
2. Click **Create** → **Custom Object**.
3. Enter the following details:
    - **Label:** HandsMen Order
    - **Plural Label:** HandsMen Orders
    - **Record Name:** HandsMen Order Number

- **Data Type:** Auto Number
- **Display Format:** O-{0000}
- **Starting Number:** 001

4. Enable:
    - ☑ Allow Reports
    - ☑ Allow Search
5. Click **Save**.

Activity 4: Create Inventory Object

Purpose:
The Inventory object tracks stock availability for products and supports inventory monitoring and automation.

Steps to Create the Object:

1. Go to **Setup** → **Object Manager**.
2. Click **Create** → **Custom Object**.
3. Enter the following details:
    - **Label:** Inventory
    - **Plural Label:** Inventories
    - **Record Name:** Inventory Number
    - **Data Type:** Auto Number
    - **Display Format:** I-{0000}
    - **Starting Number:** 001

4. Enable:
    - ☑ Allow Reports
    - ☑ Allow Search
5. Click **Save**.

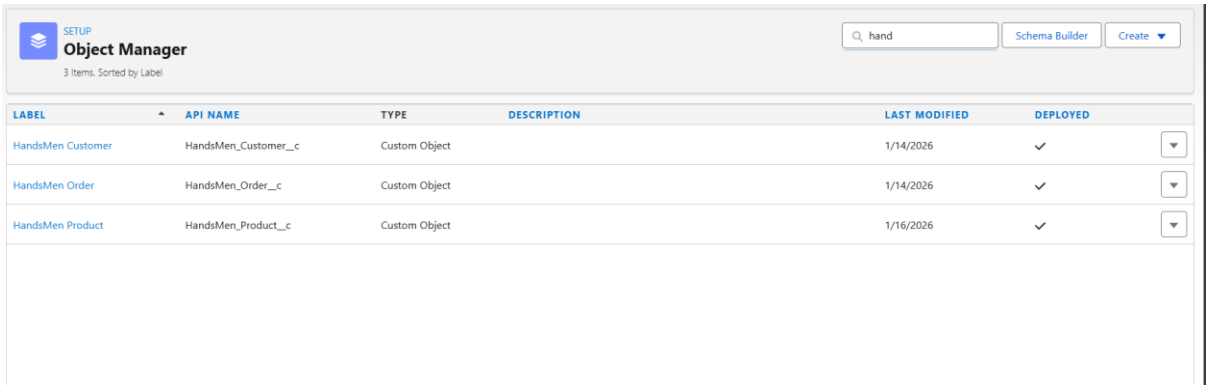
Activity 5: Create Marketing Campaign Object

Purpose:
The Marketing Campaign object is used to manage promotional activities and customer engagement initiatives.

Steps to Create the Object:

1. Navigate to **Setup** → **Object Manager**.
2. Click **Create** → **Custom Object**.
3. Enter the following details:
    - **Label:** Marketing Campaign
    - **Plural Label:** Marketing Campaigns
    - **Record Name:** Marketing Campaign Number
    - **Data Type:** Auto Number
    - **Display Format:** MC-{0000}
    - **Starting Number:** 001
4. Enable:
    - ☑ Allow Reports
    - ☑ Allow Search

5. Click **Save**.



Outcome of Milestone 2:

All core custom objects required for the HandsMen Threads Salesforce Automation System are successfully created. These objects form the foundation for data storage, automation, security configuration, and reporting in subsequent milestones.

Milestone 3: Custom Tab Creation

Introduction

Custom tabs in Salesforce provide users with direct access to object records through the navigation bar. Creating custom tabs for HandsMen Threads objects improves usability by allowing users to quickly access and manage records without navigating through Object Manager or related lists.

This milestone focuses on creating custom object tabs for all core HandsMen Threads objects to enhance user experience and operational efficiency.

Activity: Creating a Custom Tab – HandsMen Customer

Purpose:
To provide a dedicated navigation tab for managing HandsMen Customer records.

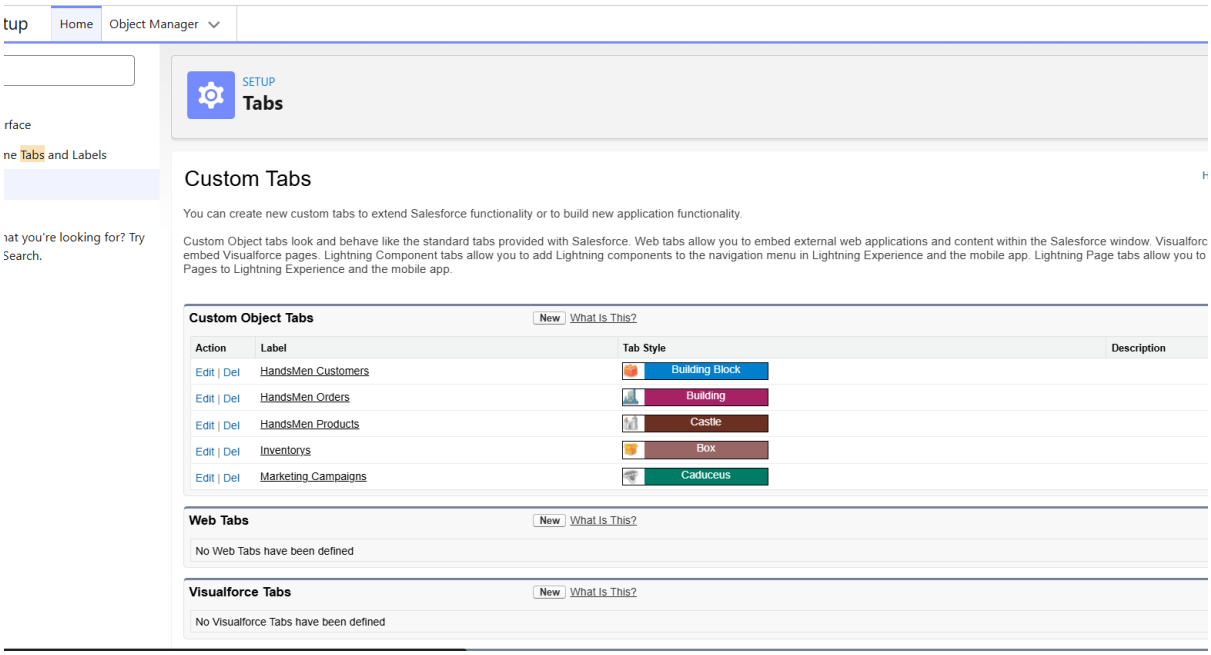
Steps to Create the Tab:

1. Go to the **Setup** page.
2. In the **Quick Find** search box, type **Tabs** and select **Tabs**.
3. Under **Custom Object Tabs**, click **New**.
4. Select the object **HandsMen Customer**.
5. Choose any preferred tab style and click **Next**.
6. On the **Add to Profiles** page, keep the default selection and click **Next**.
7. On the **Add to Custom Apps** page, keep the default selection and click **Save**.

Creating Tabs for Other Objects

Repeat the same steps to create custom tabs for the following objects:

- HandsMen Product
- HandsMen Order
- Inventory
- Marketing Campaign



Outcome of Milestone 3:

Custom tabs are successfully created for all HandsMen Threads objects, enabling easy navigation and efficient record management within the Salesforce application.

Milestone 4: Create a Lightning App

Activity: Creating a Lightning App – HandsMen Threads

A Lightning App provides a centralized workspace where users can access all required objects, reports, and dashboards related to the HandsMen Threads business.

Steps to Create a Lightning App:

1. Go to the **Setup** page in Salesforce.

2. In the **Quick Find** search box, type **App Manager** and select **App Manager**.
3. Click on **New Lightning App**.

App Details & Branding

4. Enter the following details:
    - **App Name:** HandsMen Threads
    - **Developer Name:** Auto populated by Salesforce
    - **Description:** Enter a meaningful description related to men's fashion operations and customer management
    - **Image:** Optional (can be skipped)
    - **Primary Color:** Keep the default value
5. Click **Next**.

App Options

6. On the **App Options** page, keep all options as default and click **Next**.

Utility Items

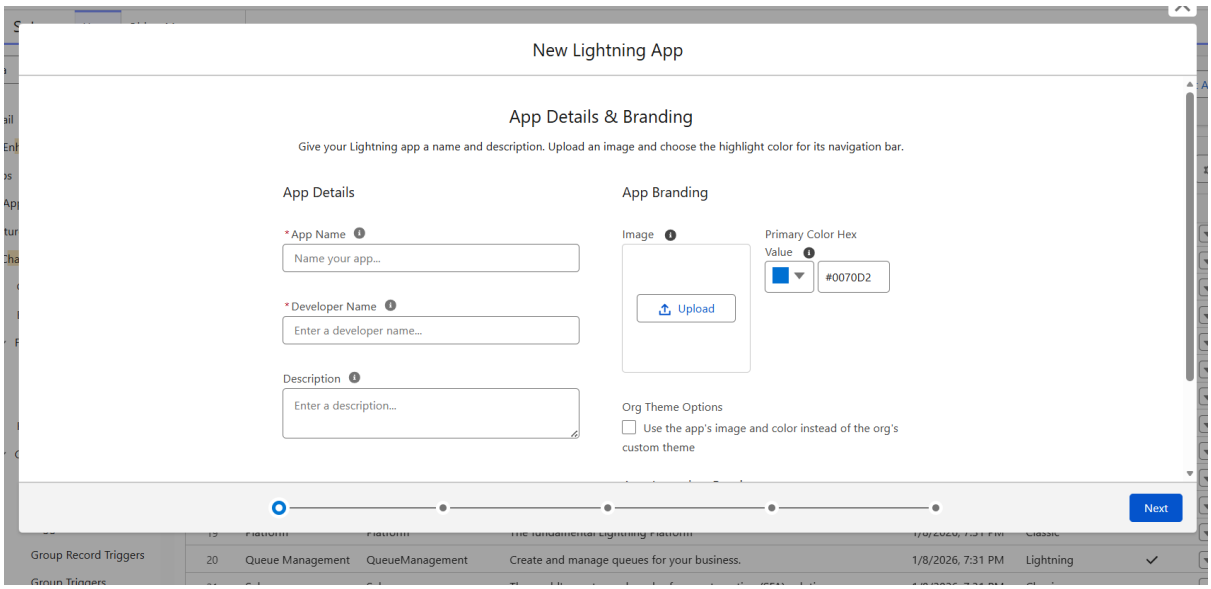
7. On the **Utility Items** page, keep the default settings and click **Next**.

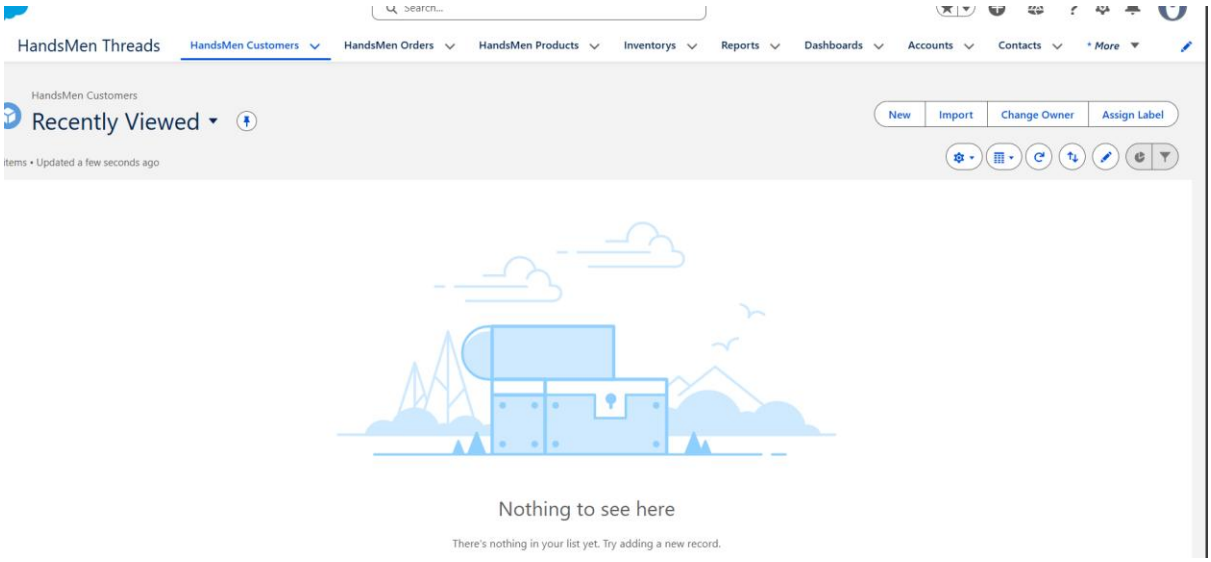
Navigation Items

8. In the **Available Items** search bar, search and add the following items using the arrow button:
    - HandsMen Customer
    - HandsMen Order
    - Inventory
    - HandsMen Product
    - Marketing Campaign
    - Reports
    - Dashboards
    - Account
    - Contact

(Note: Select the custom objects created in the previous activities.)

9. Click **Next**.





Milestone 5: Field Creation & Relationships

Creating Fields in HandsMen Customer Object

Fields are created to store essential customer details such as email, phone number, and loyalty status.

Steps to Create Fields:

1. Go to **Setup**.
2. Click on **Object Manager**.
3. Search for and select **HandsMen Customer**.
4. Click on **Fields & Relationships**.
5. Click **New**.

Creating Email Field

1. Select **Data Type** as **Email**.
2. Click **Next**.

3. Enter the following details:
    - **Field Label:** Email
    - **Field Name:** Auto-generated by Salesforce
4. Click **Next** → **Next** → **Save & New**.

Creating Phone Field

1. Select **Data Type** as **Phone**.
2. Click **Next**.
3. Enter the following details:
    - **Field Label:** Phone
    - **Field Name:** Auto-generated by Salesforce
4. Click **Next** → **Next** → **Save & New**.

Creating Picklist Field – Loyalty Status

1. Select **Data Type** as **Picklist**.
2. Click **Next**.
3. Enter:
    - **Field Label:** Loyalty Status
4. Under values, select **Enter values, with each value separated by a new line**.
5. Enter the following values:
    - Gold

- Silver
- Bronze

6. Click **Next** → **Next** → **Next** → **Save & New**.

Creating Relationships

Unit 1: Lookup Relationship between Marketing Campaign and HandsMen Customer

1. Go to **Setup → Object Manager**.
2. Select **Marketing Campaign**.
3. Click **Fields & Relationships** → **New**.
4. Select **Lookup Relationship** and click **Next**.
5. For **Related To**, select **HandsMen Customer**.
6. Click **Next**.
7. Enter:
   - **Field Label:** HandsMen Customer
8. Click **Next** → **Next** → **Save**.

Unit 2: Lookup Relationship between HandsMen Product and HandsMen Order

1. Go to **Setup → Object Manager**.
2. Select **HandsMen Product**.
3. Click **Fields & Relationships** → **New**.

4. Select **Lookup Relationship** and click **Next**.
5. For **Related To**, select **HandsMen Order**.
6. Enter:
    - **Field Label:** Order
7. Click **Next** → **Next** → **Save**.

Unit 3: Lookup Relationship between HandsMen Order and HandsMen Customer

1. Go to **Setup → Object Manager**.
2. Select **HandsMen Order**.
3. Click **Fields & Relationships** → **New**.
4. Select **Lookup Relationship** and click **Next**.
5. For **Related To**, select **HandsMen Customer**.
6. Enter:
    - **Field Label:** Customer
7. Click **Next** → **Next** → **Save**.

Unit 4: Master-Detail Relationship between Inventory and HandsMen Product

1. Go to **Setup → Object Manager**.
2. Select **Inventory**.
3. Click **Fields & Relationships** → **New**.

4. Select **Master-Detail Relationship** and click **Next**.
5. For **Related To**, select **HandsMen Product**.
6. Enter:
    - **Field Label:** Product
7. Click **Next** → **Next** → **Save**.

Formula Fields

Formula Field: Stock Status (Inventory)

Object	Field	Return Type	Formula
Inventory__c	Stock_Status__c	Text	IF(Stock_Quantity__c > 10, "Available", "Low Stock")

This formula displays inventory availability based on stock quantity.

Formula Field: Full Name (HandsMen Customer)

Note: Before creating this formula field, ensure that two custom fields exist:

- **FirstName__c**
- **LastName__c**

Steps to Create Full Name Formula Field:

1. Go to **Setup → Object Manager**.

2. Select **HandsMen Customer**.
3. Click **Fields & Relationships** → **New**.
4. Select **Data Type** as **Formula** and click **Next**.
5. Enter:
    - **Field Label:** Full_Name__c
    - **Return Type:** Text
6. Click **Next**.
7. Under **Advanced Formula**, enter:

```
FirstName__c + " " + LastName__c
```

8. Click **Check Syntax**.
9. Click **Next** → **Next** → **Save & New**.

Object Summary

Object Name	**Type**	**Description**	**Key Fields**
HandsMen Customer__c	Custom Object	Stores customer details	Name, Email, Phone, Loyalty_Status__c, Total_Purchases__c

Object Name	Type	Description	Key Fields
HandsMen Product__c	Custom Object	Stores product catalog	Name, SKU, Price, Stock_Quantity__c
HandsMen Order__c	Custom Object	Stores customer orders	Order Number, Status, Quantity, Total_Amount__c
Inventory__c	Custom Object	Tracks inventory levels	Inventory Number, Warehouse, Stock_Quantity__c
Marketing_Campaign__c	Custom Object	Manages campaigns	Campaign Name, Start Date, End Date

Milestone 6: Validation Rules

Validation rules are used to maintain data accuracy by preventing users from saving incorrect or invalid data in Salesforce records.

Validation Rules Summary

Object	Field	Validation Rule
HandsMen Order__c	Total_Amount__c	Total_Amount__c <= 0
Inventory__c	Stock_Quantity__c	Stock_Quantity__c <= 0
HandsMen Customer__c	Email	NOT(CONTAINS(Email, "@gmail.com"))

Activity 1: Creating Validation Rules

Validation Rule 1: Total Amount Validation (HandsMen Order)

This rule ensures that an order cannot be saved with zero or negative total amount.

Steps:

1. Go to **Setup**.
2. Click **Object Manager**.
3. Search and select **HandsMen Order__c**.
4. Click **Validation Rules**.
5. Click **New**.

Rule Details:

- **Rule Name:** Total_Amount
- **Error Condition Formula:**

Total_Amount__c <= 0

- **Error Message:** Please Enter Correct Amount
- **Error Location:** Field
- **Field:** Total Amount

6. Click **Save**.

Validation Rule 2: Stock Quantity Validation (Inventory)

This rule prevents inventory records from having zero or negative stock values.

Steps:

1. Go to **Setup → Object Manager**.
2. Select **Inventory__c**.
3. Click **Validation Rules** → **New**.

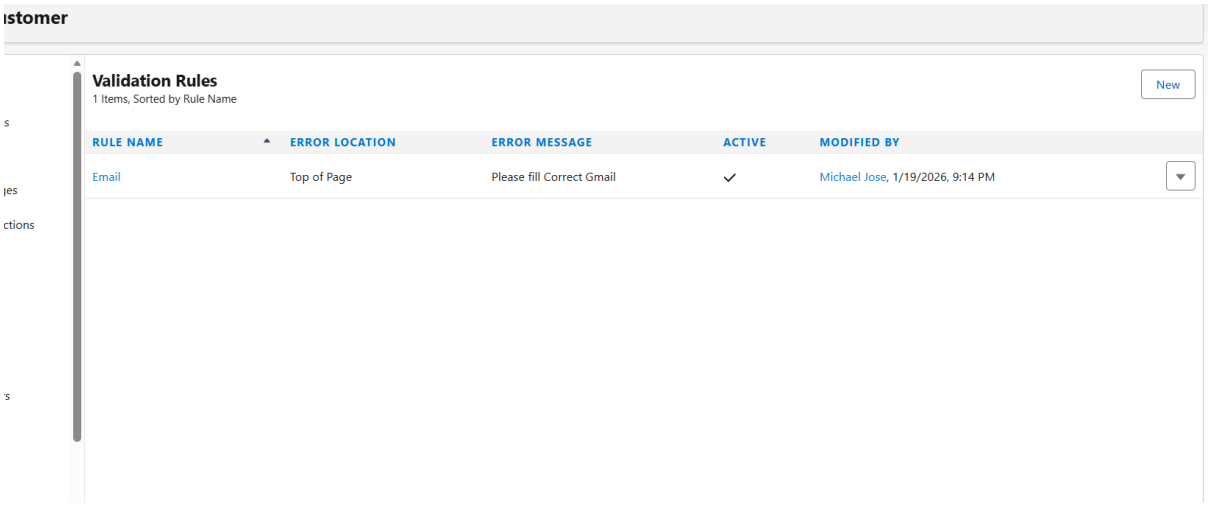
Rule Details:

- **Rule Name:** Stock_Quantity
- **Error Condition Formula:**

Stock_Quantity__c <= 0

- **Error Message:** The inventory count is never less than zero.
- **Error Location:** Top of Page

4. Click **Save**.



Validation Rule 3: Email Validation (HandsMen Customer)

This rule ensures that customers enter a valid Gmail email address.

Steps:

1. Go to **Setup → Object Manager**.
2. Select **HandsMen Customer__c**.
3. Click **Validation Rules** → **New**.

Rule Details:

- **Rule Name:** Email
- **Error Condition Formula:**

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

- **Error Message:** Please fill Correct Gmail
- **Error Location:** Top of Page

4. Click **Save**.

Outcome:

All validation rules are successfully implemented, ensuring:

- Correct order amount entry
- Valid inventory stock levels
- Proper customer email format

Milestone 7: Data Security – Profiles, Roles, and Users

Data security in Salesforce ensures that users can access only the data relevant to their roles and responsibilities. This milestone covers profile creation, role hierarchy setup, and user creation.

Activity 1: Creating a Custom Profile

Profiles control object-level permissions and user access within Salesforce.

Steps to Create Profile (Platform 1):

1. Go to **Setup**.
2. In the **Quick Find** box, type **Profiles** and click on **Profiles**.
3. Click **Clone** next to the **Standard User** profile.
4. Enter the following details:
    - **Profile Name:** Platform 1
5. Click **Save**.

Assign Object Permissions:

6. While still on the **Platform 1** profile page, click **Edit**.

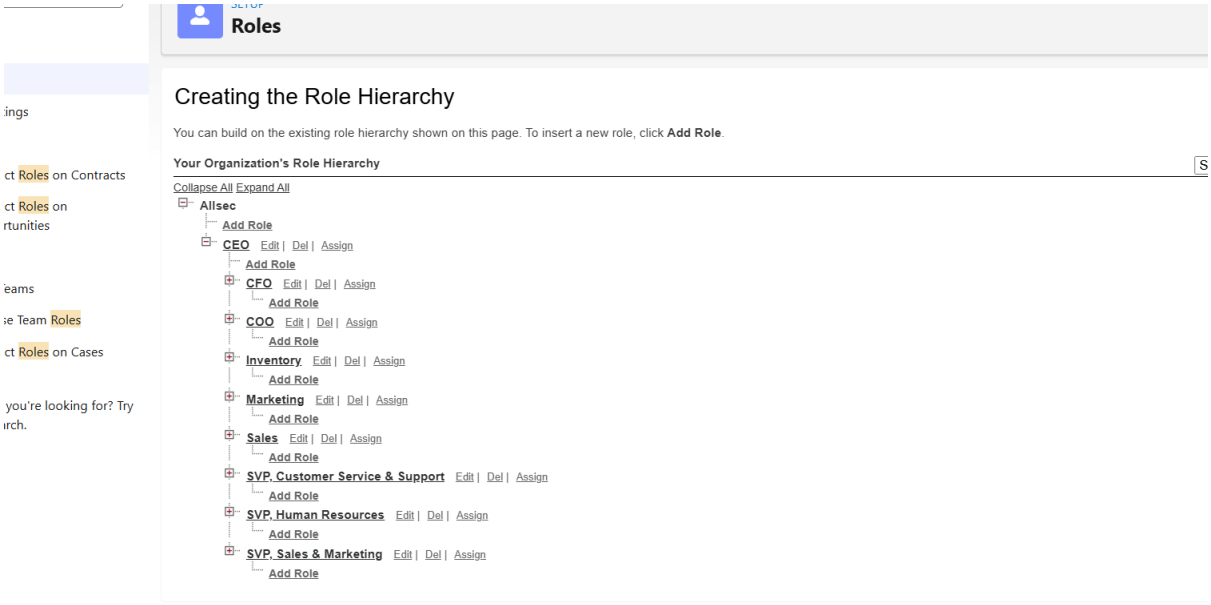
7. Scroll down to **Custom Object Permissions**.
8. Provide required access permissions for:
    - **HandsMen Product**
    - **Inventory**
9. Scroll down and click **Save**.

Activity 2: Creating Roles

Roles define record-level visibility and reporting hierarchy within the organization.

Creating Sales Role

1. Go to **Setup**.
2. In the **Quick Find** box, search for **Roles** and click **Set Up Roles**.
3. Click **Expand All**.
4. Click **Add Role** under the **CEO** role.
5. Enter the following details:
    - **Label:** Sales
    - **Role Name:** Auto populated
    - **This Role Reports To:** CEO
6. Click **Save**.



Creating Additional Roles

Repeat the above steps to create the following roles:

- **Inventory**
- **Marketing**

Activity 3: Creating Users

Users are created and assigned appropriate roles, profiles, and licenses to access the Salesforce system.

User 1: Niklaus Mikaelson

Steps to Create User:

1. Go to **Setup**.
2. In the **Quick Find** box, type **Users** and select **Users**.
3. Click **New User**.
4. Fill in the following details:

- **First Name:** Niklaus
- **Last Name:** Mikaelson
- **Alias:** Enter an alias
- **Email:** Enter your personal email ID
- **Username:** text@text.text
- **Nick Name:** Enter a nickname
- **Role:** Sales
- **User License:** Salesforce Platform
- **Profile:** Platform 1

5. Click **Save**.

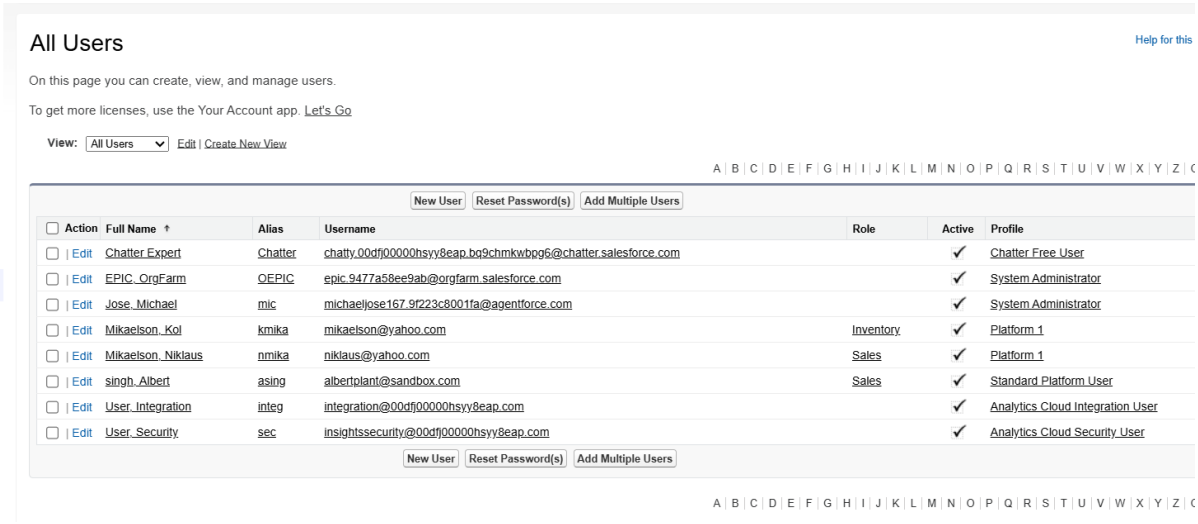
User 2: Kol Mikaelson

Steps to Create User:

1. Go to **Setup → Users**.
2. Click **New User**.
3. Enter the following details:
    - **First Name:** Kol
    - **Last Name:** Mikaelson
    - **Alias:** Enter an alias
    - **Email:** Enter your personal email ID
    - **Username:** text@text.text
    - **Nick Name:** Enter a nickname

- **Role:** Inventory
- **User License:** Salesforce Platform
- **Profile:** Platform 1

4. Click **Save**.



Additional Users

Create **two more users** by repeating the above steps, assigning roles and profiles as mentioned in **Activity 2**, based on business requirements.

Outcome:

- Custom profile **Platform 1** is created and configured
- Role hierarchy (Sales, Inventory, Marketing) is established
- Users are successfully created with appropriate roles, profiles, and licenses

This ensures **secure, role-based access control** within the HandsMen Threads Salesforce application.

Activity 4: Creating and Assigning a Permission Set

Permission Sets are used to grant additional permissions to users without changing their profiles. In this activity, a permission set is created to provide extended access to HandsMen Customer and HandsMen Order objects.

Creating Permission Set – Permission_Platform_1

Steps to Create Permission Set:

1. Go to **Setup**.
2. In the **Quick Find** box, type **Permission Sets** and click on **Permission Sets**.
3. Click **New**.
4. Enter the following details:
    - **Label:** Permission_Platform_1
    - **API Name:** Auto populated
5. Click **Save**.

Assign Object Permissions

6. After saving, navigate to **Apps** → select **Object Settings**.
7. Click on **HandsMen Customer** object.

8. Click **Edit**.
9. Under **Object Permissions**, enable the following:
    - Read
    - Create
    - Edit
    - Delete
10. Click **Save**.

Assign Permissions for HandsMen Order Object

11. Go back to **Object Settings**.
12. Select **HandsMen Order** object.
13. Click **Edit**.
14. Enable the following permissions:

- Read
- Create
- Edit
- Delete

15. Click **Save**.

Assign Permission Set to Users

16. From the permission set page, click **Manage Assignments**.

17. Click **Add Assignment**.
18. Select a user (any one user with **Profile: Platform 1**).
19. Click **Next**.
20. Click **Assign**.
21. Click **Done**.

Outcome:

- Permission set **Permission_Platform_1** is successfully created
- Additional object permissions are granted without modifying profiles
- Users with **Platform 1** profile now have enhanced access to Customer and Order objects

This ensures **flexible and secure access control** within the HandsMen Threads Salesforce application.

Milestone 8: Email Templates and Email Alerts

Email templates and alerts are used to automate communication with customers and internal teams. In this milestone, email templates are created and linked with email alerts to trigger notifications automatically.

Activity 1: Create Order Confirmation Email Template

Steps to Create a Classic Email Template

1. Go to **Salesforce Setup**.
2. Click the **Gear Icon (⚙)** in the top-right corner and select **Setup**.
3. In the **Quick Find** box, search for **Classic Email Templates** and click on it.
4. Click **New Template**.
5. Choose **HTML (with Classic Letterhead)** for a formatted email template.

Fill in Email Template Details

6. Enter the following details:
    - **Folder:** Unfiled Public Email Templates (or create a new folder)
    - **Available for Use:** ✅ Checked
    - **Email Template Name:** Order_Confirmation_Email
    - **Encoding:** UTF-8 (Default)
    - **Subject:** Your Order has been Confirmed!

HTML Body Content

Dear {!Order__c.Customer__c},

Your order #{!Order__c.Name} has been confirmed!

Thank you for shopping with us.

Best Regards,

Sales Team

7. Click **Save**.

Create Remaining Email Templates

Repeat the above steps to create the following email templates:

- **Low Stock Alert**
- **Loyalty Program Email**

(Use appropriate subject lines and content based on the email purpose.)

Activity 2: Create an Email Alert

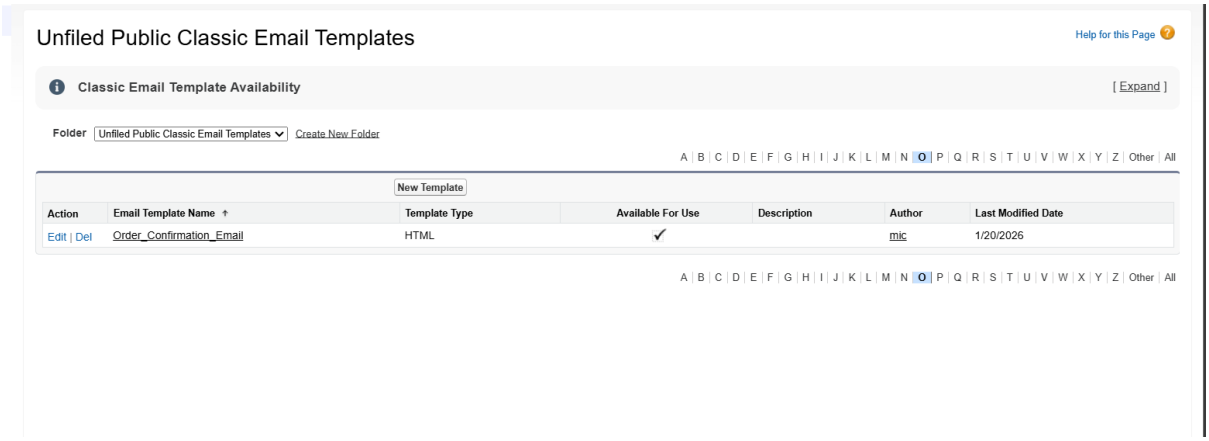
Email alerts are used to automatically send emails when specific conditions are met.

Steps to Create Email Alert – Order Confirmation

1. Go to **Setup**.
2. In the **Quick Find** box, search for **Email Alerts** and click on it.
3. Click **New Email Alert**.

Email Alert Details

4. Enter the following details:
    - **Description:** Order Confirmation Email Alert
    - **Object:** Order__c
    - **Email Template:** Select *Order_Confirmation_Email*
    - **Recipient Type:** Related Record
    - **Related Record:** Customer__c
5. Click **Save**.



Outcome:

- Order confirmation email template is created
- Low stock and loyalty email templates are prepared
- Email alert is configured to notify customers automatically when an order is confirmed

This automation improves **customer communication** and ensures timely notifications without manual effort.

Milestone 9: Flows (Automation)

Flows are used to automate business processes such as sending emails, updating records, and maintaining loyalty status without manual intervention.

Activity 1: Order Confirmation Email (Record-Triggered Flow)

This flow sends an email confirmation to the customer when an order status is updated to **Confirmed**.

Steps to Create the Flow:

1. Go to **Setup**.
2. In the **Quick Find** box, search for **Flows** and click on **Flows**.
3. Click **New Flow**.
4. Select **Record-Triggered Flow** and click **Create**.

Set Flow Trigger Details:

5. **Object:** Order__c
6. **Trigger:** When a record is updated

Condition Requirements:

7. Set the condition as:
    - **Field:** Status__c
    - **Operator:** Equals

- **Value:** Confirmed

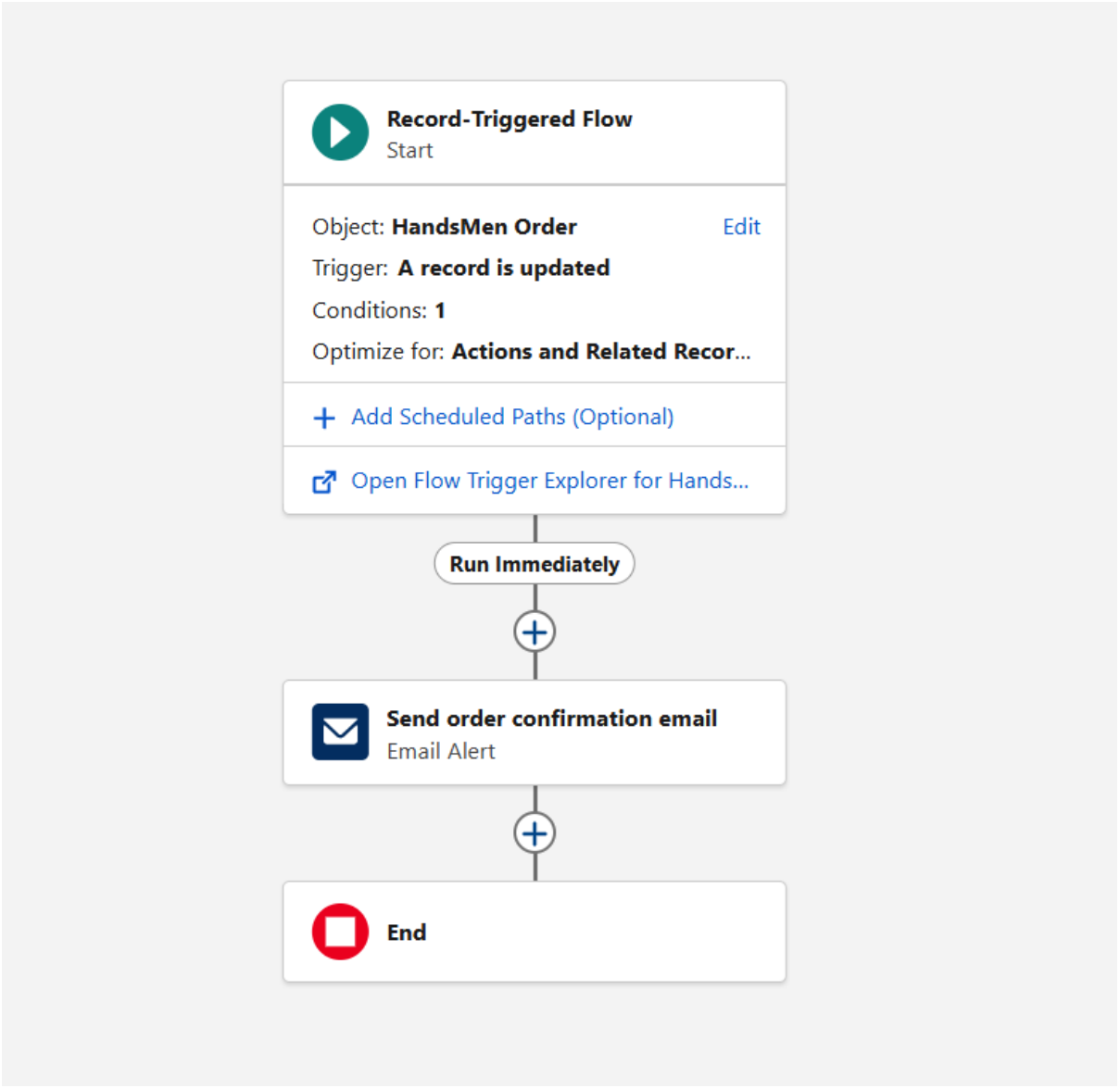
8. Select **Only when a record is updated to meet the condition**.
9. Click **Done**.

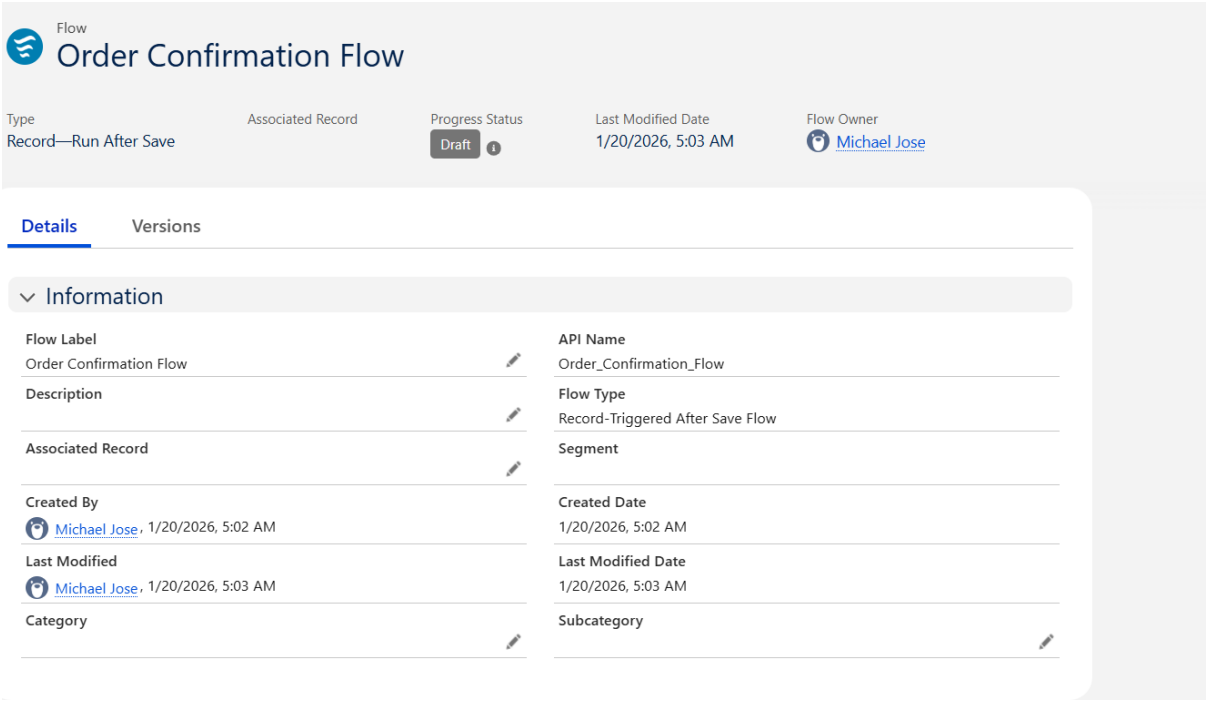
Add Action – Send Email Alert:

10. Click the **+ (Plus)** icon.
11. Select **Action**.
12. Choose **Action Type:** Send Email Alert.
13. Select **Email Alert:** Order Confirmation Email Alert.
14. Enter:
    - **Label:** Send Order Confirmation Email
    - **Record ID:** {!$Record.Id}
15. Click **Save**.

Save and Activate:

16. **Flow Name:** Order Confirmation Flow
17. Click **Save** → **Activate**.


Record-Triggered Flow
Start
Object: HandsMen Order
Edit
Trigger: A record is updated
Conditions: 1
Optimize for: Actions and Related Recor...
Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Hands...
Run Immediately
Send order confirmation email
Email Alert
End



Activity 2: Stock Alert Email (Record-Triggered Flow)

This flow sends an alert email when inventory stock falls below 5 units.

Steps to Create the Flow:

1. Go to **Setup → Flows**.

2. Click **New Flow**.

3. Select **Record-Triggered Flow** and click **Create**.

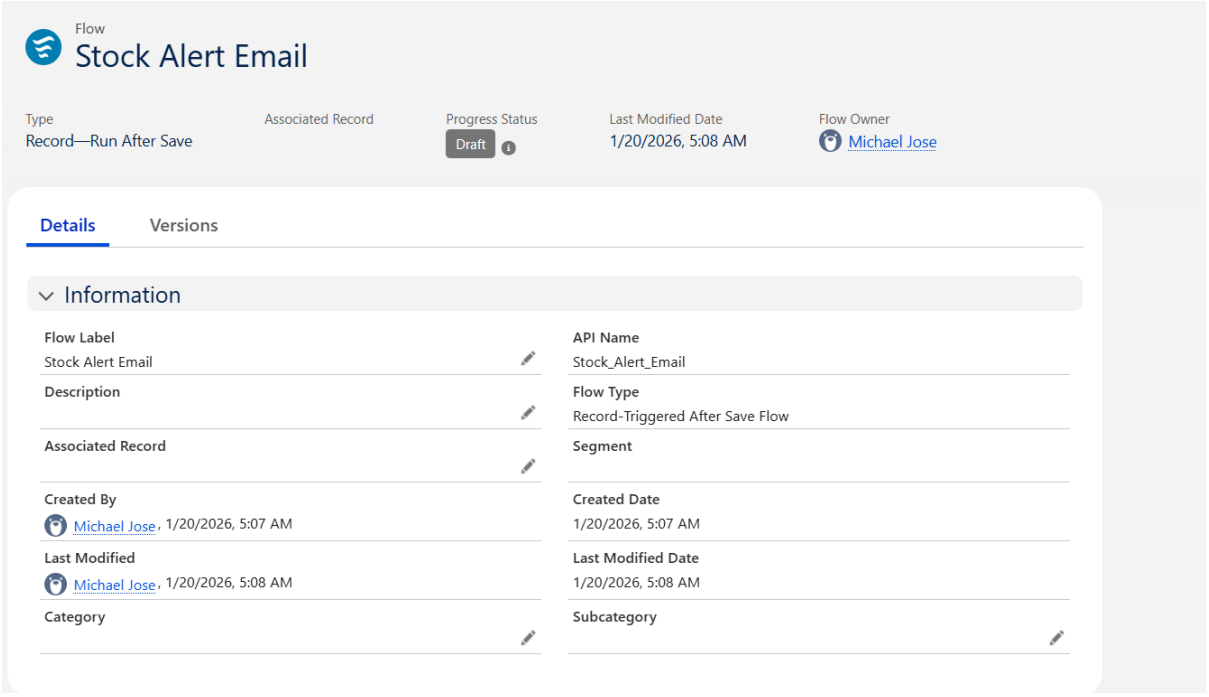
Set Flow Trigger Details:

4. **Object:** Inventory__c

5. **Trigger:** When a record is created or updated

Condition Requirements:

6. Set the condition:
    - **Field:** Stock_Quantity__c
    - **Operator:** Less Than
    - **Value:** 5
7. Select **Every time a record is updated and meets the condition requirements**.
8. Click **Done**.



Add Action – Send Email Alert:

9. Click the **+ (Plus)** icon.
10. Select **Action**.
11. Choose **Action Type:** Send Email Alert.
12. Create or select an email alert similar to Order Confirmation.

13. **Recipient:** Inventory Manager.
14. Click **Save**.

Save and Activate:

15. **Flow Name:** Stock Alert Flow
16. Click **Save** → **Activate**.

Activity 3: Loyalty Status Update (Scheduled Flow)

This scheduled flow updates customer loyalty status based on total purchases.

Steps to Create the Flow:

1. Go to **Setup → Flows**.
2. Click **New Flow**.
3. Select **Schedule-Triggered Flow** and click **Create**.

Schedule Configuration:

4. Set **Start Date & Time:** Choose a time to run daily.
5. **Frequency:** Daily.
6. Click **Done**.

Get Customer Records:

7. Click the **+ (Plus)** icon → select **Get Records**.

8. **Object:** HandsMen_Customer__c

9. **Filter:** Retrieve all records.

10. **Sort Order:** None.

11. Click **Done**.

Loop Through Records:

12. Click **+** → select **Loop**.

13. **Collection:** {!Get_Records}

14. **Direction:** First to Last.

15. Click **Done**.

Decision Logic Inside Loop:

16. Click **+** inside the loop → select **Decision**.

17. Set conditions:

- If **Total_Purchases__c > 1000**, set **Loyalty_Status__c = Gold**
- Else if **Total_Purchases__c < 500**, set **Loyalty_Status__c = Bronze**
- Else (Default): set **Loyalty_Status__c = Silver**

Update Records:

18. Add **Update Records** for each decision outcome.

19. **Object:** HandsMen Customer
20. Set **Loyalty_Status__c** accordingly.
21. Click **Done**.

Save and Activate:

22. **Flow Name:** Loyalty Status Update Flow
23. Click **Save** → **Activate**.

Outcome:

- Order confirmation emails are sent automatically
- Inventory managers receive low-stock alerts
- Customer loyalty status is updated daily based on purchase history

This automation improves **customer engagement**, **inventory control**, and **operational efficiency** within the HandsMen Threads Salesforce system.

Milestone 10: Automation Using Apex

Apex is used when business logic is complex and cannot be handled entirely using declarative tools like Flows. In this milestone, Apex triggers and handler classes are implemented to enforce business rules during order processing.

Apex Triggers Overview

Trigger Name	Object	Purpose
Update Order Total	Order__c	Auto-update Total_Amount__c on order save
Stock Deduction	Inventory__c	Reduce stock when an order is placed
Loyalty Status Update	Customer__c	Upgrade loyalty status based on purchases

---

Activity 1: Create Apex Class – OrderTriggerHandler

This Apex class contains the business logic to validate order quantity based on order status before saving the record.

Steps to Create Apex Class:

1. Go to **Setup**.
2. Click the **Gear Icon** → select **Developer Console**.
3. The Developer Console opens in a new window.
4. Click **File → New → Apex Class**.
5. Enter the class name as **OrderTriggerHandler**.

6. Click **OK**.

Apex Class Code: OrderTriggerHandler

```
public class OrderTriggerHandler {


    public static void
validateOrderQuantity(List<HandsMen_Order__c>
orderList) {


        for (HandsMen_Order__c order : orderList) {


            if (order.Status__c == 'Confirmed') {


                if (order.Quantity__c == null || order.Quantity__c
<= 500) {
                    order.Quantity__c.addError(
                        'For Status "Confirmed", Quantity must be
more than 500.'
                    );
                }


            } else if (order.Status__c == 'Pending') {
```

```
            if (order.Quantity__c == null || order.Quantity__c
<= 200) {
                order.Quantity__c.addError(
                    'For Status "Pending", Quantity must be more
than 200.'
                );
            }

        } else if (order.Status__c == 'Rejection') {

            if (order.Quantity__c == null || order.Quantity__c
!= 0) {
                order.Quantity__c.addError(
                    'For Status "Rejection", Quantity must be 0.'
                );
            }

        }
    }

    System.debug('All records validated successfully.');
  }
}
```

7. Save the code by clicking **File → Save**.

Purpose of the Apex Class:

- Validates order quantity before saving
- Applies different quantity rules based on order status
- Prevents invalid data from being committed to the database
- Centralizes business logic for reuse and maintainability

Activity 2: Create Apex Trigger – OrderTrigger

This trigger calls the handler class before inserting or updating an order record.

Steps to Create Apex Trigger:

1. In **Developer Console**, click **File → New → Apex Trigger**.
2. Enter:
    - **Trigger Name:** OrderTrigger
    - **sObject:** HandsMen_Order__c
3. Click **Submit**.

Apex Trigger Code: OrderTrigger

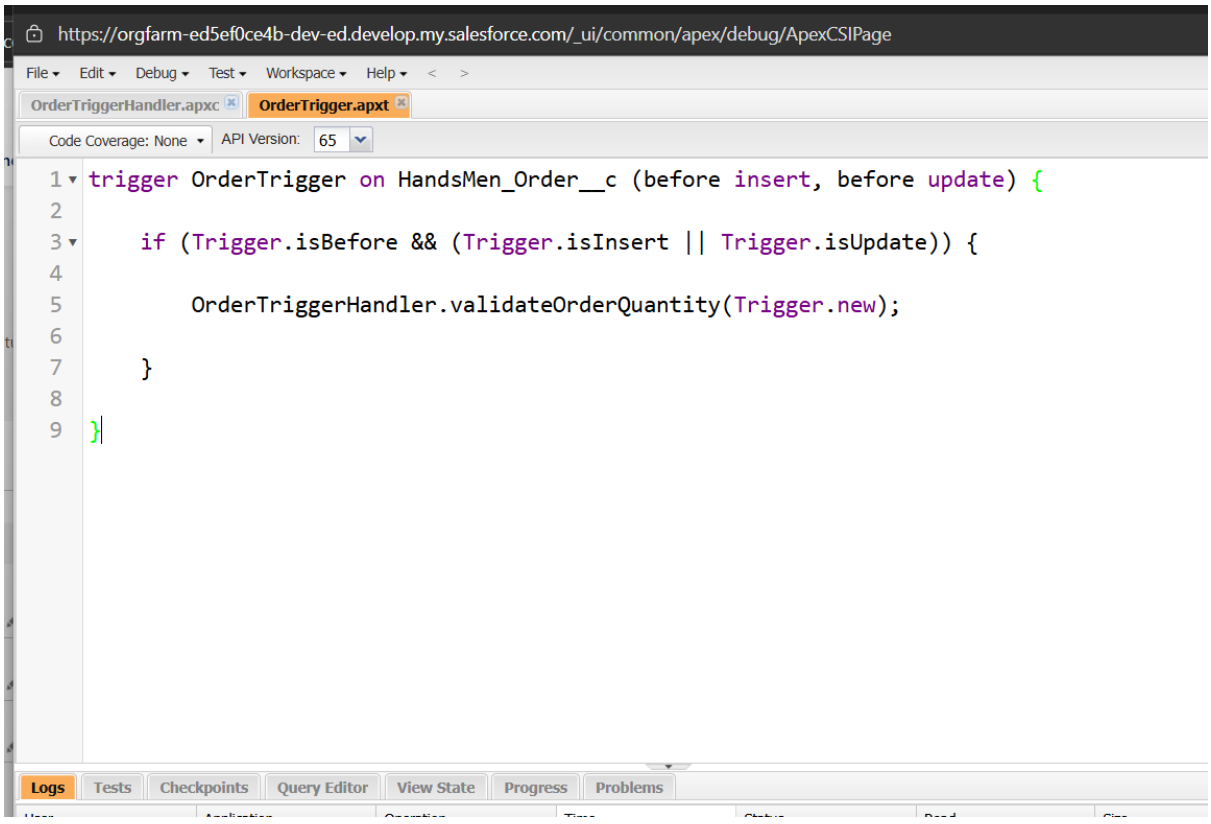
trigger OrderTrigger on HandsMen_Order__c (before insert, before update) {

```
    if (Trigger.isBefore && (Trigger.isInsert ||
Trigger.isUpdate)) {

OrderTriggerHandler.validateOrderQuantity(Trigger.new);
    }


}
```

4. Save the trigger by clicking **File → Save**.



Purpose of the Trigger:

- Executes before order records are inserted or updated

- Invokes the handler class to validate business rules
- Prevents invalid orders from being saved
- Ensures data integrity at the database level

Outcome:

- Apex automation successfully implemented
- Order quantity validation enforced based on status
- Business rules executed before database commit
- Improved data consistency and reliability

This Apex implementation complements Flows by handling **complex validation logic** that requires programmatic control.

Milestone 11: Batch Apex

Batch Apex is used to process large volumes of records asynchronously. In this activity, a batch job is created to automatically update inventory stock levels and is scheduled to run at a specific time.

Activity 1: Create Batch Apex Class – InventoryBatchJob

This batch job identifies products with low stock and automatically replenishes inventory.

Steps to Create Apex Class:

1. Go to **Setup**.

2. Click the **Gear Icon** → select **Developer Console**.

3. The Developer Console opens in a new window.

4. Click **File → New → Apex Class**.

5. Enter the class name as **InventoryBatchJob**.

6. Click **OK**.

Apex Class Code: InventoryBatchJob

```
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'
      );
   }


   global void execute(Database.BatchableContext BC,
List<SObject> records) {


      List<HandsMen_Product__c> productsToUpdate =
         new List<HandsMen_Product__c>();
```

```
    // Cast SObject list to Product__c list
    for (SObject record : records) {
        HandsMen_Product__c product =
            (HandsMen_Product__c) record;
        product.Stock_Quantity__c += 50; // Restock logic
        productsToUpdate.add(product);
    }

    if (!productsToUpdate.isEmpty()) {
        try {
            update productsToUpdate;
        } catch (DmlException e) {
            System.debug(
                'Error updating inventory: ' + e.getMessage()
            );
        }
    }
}

global void finish(Database.BatchableContext BC) {
    System.debug('Inventory Sync Completed');
```

```
    }

    // Scheduler Method
    global void execute(SchedulableContext SC) {
        InventoryBatchJob batchJob = new InventoryBatchJob();
        Database.executeBatch(batchJob, 200);
    }
}
```

7. Save the code by clicking **File → Save**.

Purpose of the Batch Apex:

- Identifies products with low stock
- Automatically replenishes inventory in bulk
- Handles large data volumes efficiently
- Runs asynchronously without impacting user performance

Activity 2: Schedule the Batch Apex Job

The batch job is scheduled to run automatically at a fixed time every day.

Steps to Schedule the Batch Job:

1. Go to **Setup**.

2. Click the **Gear Icon** → select **Developer Console**.
3. In the Developer Console, click **Debug → Open Execute Anonymous Window**.

Execute Anonymous Code:

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

4. Click **Execute**.

Verify Scheduled Job:

5. Go back to **Setup**.
6. In the **Quick Find** box, search for **Jobs**.
7. Click on **Scheduled Jobs**.
8. Verify that **Daily Inventory Sync** appears in the list.

Outcome:

- Batch Apex successfully created and scheduled
- Inventory stock is automatically updated in bulk

- System performance is maintained using asynchronous processing
- Daily inventory synchronization is fully automated

Conclusion

The **HandsMen Threads Salesforce implementation** successfully delivers a comprehensive, scalable, and automation-driven CRM solution tailored to the needs of a premium men's fashion business. Through the effective use of Salesforce's declarative and programmatic capabilities, the project transforms manual and fragmented processes into a centralized, secure, and intelligent system.

A well-structured data model was designed using custom objects and relationships to accurately represent customers, products, orders, inventory, and marketing campaigns. Data integrity was ensured through validation rules, required fields, and formula fields, preventing incorrect data entry and maintaining consistency across the system. Role-based access control using profiles, roles, permission sets, and sharing configurations ensured that users could access only the data relevant to their responsibilities, strengthening overall system security.

Automation played a key role in enhancing operational efficiency. Record-triggered flows automated order confirmation emails and proactive stock alerts, while scheduled flows dynamically updated customer loyalty status based on purchase history. Apex triggers were

implemented to enforce complex business rules such as order quantity validation, ensuring that critical validations occur before records are saved. Batch Apex and scheduled Apex further extended automation by handling bulk inventory updates asynchronously, maintaining system performance even with large data volumes.

The project provided strong hands-on experience with Salesforce core concepts including data modeling, Lightning App Builder, Flow Builder, Apex programming, asynchronous processing, and deployment practices. Overall, the HandsMen Threads Salesforce solution establishes a robust foundation for future enhancements such as advanced reporting, analytics, AI-driven personalization, and external system integrations, making it a future-ready platform that supports business growth and customer engagement.