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

Project Overview

HandsMen Threads is a comprehensive Salesforce-based e-commerce platform designed to revolutionize the men's fashion retail experience. The system provides an end-to-end solution for managing premium men's fashion products, customer relationships, order processing, inventory management, and personalized shopping experiences.

Built using Salesforce Experience Cloud, Apex, Flows, and Lightning Web Components (LWC), the application delivers real-time visibility into product catalogs, customer preferences, order fulfillment, and sales analytics. It automates inventory management, order processing, customer engagement, and provides seamless integration between online shopping and business operations.

The platform enhances customer satisfaction and operational efficiency by enabling customers to browse curated collections, place orders, track shipments, submit reviews, and access personalized recommendations—all from an intuitive digital storefront. Sales teams and management can monitor inventory levels, sales performance, customer engagement, and market trends through automated reports and dashboards.

By combining automation, real-time analytics, secure payment processing, and role-based access control, HandsMen Threads helps fashion retailers maintain competitive advantage, ensure customer satisfaction, and build a data-driven retail culture that supports both growth and customer loyalty.

Objectives

- **Centralized Fashion Retail Management**: Provide a single platform to manage product catalogs, customer orders, inventory, and customer relationships.
- **Automation of Workflows**: Streamline order processing, inventory updates, customer notifications, and sales tracking using Flows and Apex.
- **Customer Empowerment**: Enable customers to browse products, place orders, track deliveries, and submit reviews through a self-service interface.
- **Real-Time Insights**: Equip sales and management teams with actionable dashboards for inventory management, sales analytics, and customer engagement analysis.
- **Inventory Management**: Automatically track stock levels, alert managers about low inventory, and manage product availability using Scheduled Apex and Flows.
- **Secure & Role-Based Access**: Ensure data privacy through Salesforce profiles, permission sets, and Experience Cloud login-based visibility.

Student Outcomes

- **Hands-On Salesforce Development**: Gain practical experience in building an e-commerce solution using Apex, Flows, and LWC.
- **End-to-End Application Design**: Learn to design and integrate data models, automation, and UI for a fashion retail management system.
- **Automation Expertise**: Implement real-time process automation for order processing, inventory management, and customer engagement.
- **Experience Cloud Implementation**: Understand how to build a customer-facing storefront for online fashion retail operations.
- **Analytical Thinking**: Create reports and dashboards for sales insights, reinforcing data-driven decision-making.
- **Asynchronous Apex Skills**: Learn to use Queueable or Scheduled Apex for large-scale inventory calculations and order processing updates.

System Requirements

Hardware Requirements:

- Computer with minimum 4 GB RAM, Dual-core processor
- Stable internet connection

Software Requirements:

- Salesforce Developer Edition Org
- Modern Web Browser (e.g., Google Chrome, Firefox)

Skills Required:

- Salesforce Configuration and Data Modeling
- Security and Access Management
- Apex Triggers, Classes, and Asynchronous Apex (Queueable, Scheduled)
- Flow Builder (Record-Triggered & Scheduled Flows)
- Lightning Web Components (LWC) Development
- Experience Cloud Site Configuration
- Reports and Dashboard Creation

Phases Overview

Phase No.	Phase Name	Description	Page Numbers
1	Requirement Analysis & Planning	Define business goals, gather requirements, and plan the architecture.	5-8
2	Salesforce Development - Backend & Configurations	Set up objects, fields, automation, and Apex logic.	9-53
3	UI/UX Development & Customization	Design intuitive interfaces using Lightning Components and Flows.	53-72
4	Data Migration, Testing & Security	Migrate data, test functionality, and enforce data security.	72-80
5	Deployment, Documentation & Maintenance	Deploy the solution, train users, maintain system health, resolve issues, Documentation.	80-86

Project Main Overview

The **HandsMen Threads** platform is a Salesforce-powered Experience Cloud application that streamlines men's fashion retail operations across multiple channels. It is designed to handle the complexities of managing product catalogs, customer orders, inventory, payments, and customer relationships in a sophisticated fashion retail environment.

The system's data model consists of key objects such as **Product__c**, **Order__c**, **Order_Item__c**, **Customer_Review__c**, **Inventory__c**, and **Shipping__c**. These objects store and interlink information about products, customer orders, inventory levels, reviews, and shipping details. Using Flows and Apex Triggers, the system automates tasks like processing orders, updating inventory, calculating totals, and sending notifications for order confirmations and shipping updates.

Customers interact through LWC components embedded in an Experience Site, allowing them to browse collections, add items to cart, place orders, track shipments, and submit product reviews. Sales managers and administrators can monitor all activity from dashboards and reports that present real-time data on sales performance, inventory levels, and customer engagement.

The system also includes approval processes for special discounts and bulk orders, validation rules to ensure data accuracy (e.g., preventing negative inventory, validating email formats), and email templates for order confirmations, shipping notifications, and promotional campaigns.

Ultimately, the HandsMen Threads platform enhances retail efficiency and customer satisfaction by creating a seamless, sophisticated shopping experience powered entirely by Salesforce.

Main Objectives

The primary goal of the HandsMen Threads platform is to establish an automated, transparent, and secure framework for managing men's fashion retail operations in a digital-first environment.

- **Automated Product Management**: Enable administrators to manage product catalogs, pricing, and availability through Flows and LWCs.
- **Order Processing**: Capture customer orders, calculate totals, apply discounts using Apex logic, and generate order summaries.
- **Inventory Tracking**: Monitor stock levels in real-time and trigger automatic alerts when inventory falls below threshold levels.
- **Customer Review Management**: Collect and display customer reviews using Flows and triggers to promote engagement and trust.
- **Shipping & Fulfillment**: Use Scheduled Apex to automatically update shipping status and notify customers about delivery progress.
- **Approval Process for Discounts**: Automate multi-step approvals for special pricing and promotional offers using Salesforce's approval process feature.
- **Real-Time Dashboards**: Provide sales teams and management with visual insights into sales trends, top-selling products, and customer satisfaction rates.
- **Data Security and Access Control**: Enforce field-level and role-based visibility using profiles, permission sets, and Experience Cloud login rules.
- **Email Notifications**: Notify customers automatically about order confirmations, shipping updates, promotional offers, and review requests.

Phase 1: Requirement Analysis & Planning

1. Understanding Business Requirements

Objective:

Understand how modern fashion retailers manage online storefronts, product catalogs, customer orders, inventory, and customer relationships—and identify challenges in maintaining customer satisfaction, operational efficiency, inventory accuracy, and competitive positioning.

The goal is to build a Salesforce-based solution that provides centralized visibility, automation, and analytics for managing a sophisticated men's fashion retail business efficiently.

Approach:

- Gather and analyze requirements from retail managers, sales teams, customers, inventory managers, and IT administrators to understand fashion retail practices and operational pain points.

- Study how product catalogs, customer orders, inventory levels, and customer feedback are currently tracked, and how sales teams monitor performance and customer satisfaction.
- Identify challenges such as manual order processing, lack of real-time inventory visibility, no structured review collection system, fragmented customer data, and limited data-driven insights.
- Conduct requirement study using multiple sources such as industry best practices, e-commerce platforms, Salesforce Documentation, and Trailhead, to design a scalable and secure fashion retail management system on the Salesforce platform.

Key Business Requirements Identified:

- Provide a Salesforce-based application for managing men's fashion product catalogs and customer orders.
- Automate order processing, inventory management, and customer engagement using Apex and Flows.
- Enable customers to browse products, place orders, and track deliveries through a self-service portal.
- Allow sales teams and managers to monitor inventory, sales performance, and customer satisfaction through dashboards.
- Collect and analyze customer reviews to measure product quality and customer satisfaction.
- Ensure secure, role-based visibility for customers, sales staff, and administrators.
- Generate analytical dashboards and reports for management to monitor retail performance and market trends.

---

2. Defining Project Scope & Objectives

Project Scope:

- Build a Salesforce-based HandsMen Threads E-Commerce Platform that automates product management, order processing, inventory tracking, and customer engagement.
- Integrate automation (Flows, Apex), UI (LWC), and analytics (Reports & Dashboards) for real-time insights.
- Provide self-service capabilities for customers to browse products, place orders, and submit reviews via an Experience Cloud site.
- Implement role-based access controls and security measures to protect customer and business data.
- Enable retail managers to configure pricing, manage inventory, and monitor sales performance trends.

Objectives Summary:

- Simplify retail operations through automation of order processing, inventory updates, and customer notification workflows.
- Empower customers with a self-service platform for product browsing, order placement, and review submission.

- Enhance visibility and accountability through real-time dashboards for sales teams and management.
- Improve customer satisfaction and loyalty tracking using custom objects and flows.
- Ensure secure access and data privacy through profile-based and login-based visibility settings.
- Support data-driven decision-making through analytics and sales performance reports.

---

3. Gathering & Analyzing User Needs

Users Involved:

- **Customers**: Browse products, add to cart, place orders, track shipments, and submit reviews.
- **Sales Representatives**: Manage customer inquiries, process orders, and track customer relationships.
- **Inventory Managers**: Monitor stock levels, update inventory, and manage product availability.
- **Retail Managers**: Approve discounts, monitor sales performance, and review customer satisfaction analytics.
- **System Administrator**: Configure user access, maintain security, and oversee automation processes.

Key Functional Needs:

- Intuitive customer storefront with product browsing and shopping cart capabilities.
- Ability for customers to view product details, pricing, sizes, and availability.
- Order placement form with payment processing and order confirmation.
- Inventory dashboard showing real-time stock levels and low-stock alerts.
- Automated inventory updates when orders are placed or products are restocked.
- Email alerts for order confirmations, shipping updates, and promotional offers.
- Review submission form for customers to rate products and provide feedback.
- Analytical dashboards to track sales trends, top products, customer demographics, and revenue.

Tools Used:

- **Google Forms**: To collect business and user requirements from retail managers and customers.
- **Miro Boards**: To visualize customer journey workflows and order processing flows.
- **User Personas**: To tailor experiences for customers, sales staff, and management teams.

Note: The tools used are mentioned considering real-time project implementation practices.

---

4. Identifying Key Salesforce Features & Tools Required

Salesforce Features Planned:

Custom Objects:

- **Product__c** → Stores product information including name, description, price, category, sizes, and images.
- **Order__c** → Tracks customer orders with order date, status, total amount, and customer information.
- **Order_Item__c** → Detail records for each product in an order (quantity, price, subtotal).
- **Customer_Review__c** → Captures product ratings, review text, and reviewer information.
- **Inventory__c** → Tracks stock levels, reorder points, and warehouse locations.
- **Shipping__c** → Manages shipping addresses, tracking numbers, and delivery status.

Standard Objects:

- **Contact** → Represents customers with contact details and preferences.
- **Account** → Represents corporate customers or retail partners.
- **User** → Represents sales staff, managers, and administrators.

Automations:

- Record-Triggered Flows, Approval Processes, and Scheduled Flows for inventory checks and order processing.

Apex:

- Triggers for validation, asynchronous classes for inventory calculations, and Apex controllers for LWC shopping cart functionality.

UI:

- Lightning App Pages, Dynamic Forms, and Lightning Web Components for product catalogs, shopping cart, and customer dashboards.

Email Services:

- Email Templates and Alerts for order confirmations, shipping notifications, and promotional campaigns.

Security:

- Profiles, Permission Sets, Role Hierarchy, Field-Level Security, and Login-Based Component Visibility.

---

5. Designing Data Model and Security Model

Data Model Includes:

Product__c (Custom Object)

1. Stores product catalog information including name, description, price, category, and availability.
2. Includes fields like Product_Code, Category (Formal, Casual, Accessories), Size, Color, and Stock_Available.
3. Used for displaying products in the storefront and inventory management.

Order__c (Custom Object)

1. Stores customer order information including order date, status, and total amount.
2. Linked to Contact for customer identification.
3. Contains fields like Order_Status (Pending, Processing, Shipped, Delivered), Payment_Method, and Shipping_Address.

Order_Item__c (Custom Object)

1. Detail records linking products to orders with quantity and pricing information.
2. Linked to both Order__c and Product__c via lookup relationships.
3. Includes fields like Quantity, Unit_Price, Subtotal, and Discount_Applied.

Customer_Review__c (Custom Object)

1. Captures customer feedback on products including ratings and review text.
2. Contains fields like Rating (1–5 stars), Review_Text, Review_Date, and Verified_Purchase.
3. Triggers email notifications to product managers for reviews below 3 stars.

Inventory__c (Custom Object)

1. Monitors stock levels for each product and warehouse location.
2. Automatically updated via Flows when orders are placed or products are restocked.
3. Contains fields like Current_Stock, Reorder_Level, Warehouse_Location, and Last_Restock_Date.

Shipping__c (Custom Object)

1. Manages shipping information for customer orders.
2. Linked to Order__c with tracking details and delivery status.
3. Contains fields like Shipping_Carrier, Tracking_Number, Estimated_Delivery, and Delivery_Status.

Contact (Standard Object)

1. Represents customers with contact information and shopping preferences.
2. Used for customer relationship management and order associations.

Account (Standard Object)

1. Represents corporate clients or wholesale partners.
2. Used for B2B sales and bulk order management.

User (Standard Object)

1. Represents sales staff, inventory managers, and administrators.
2. Used for role-based access, ownership, and internal user management.

Security Model Design:

- **Role Hierarchy**: Admin → Retail Manager → Sales Representative → Inventory Manager → Customer
- **Profiles**: System Administrator, Retail Manager Profile, Sales Representative Profile, Inventory Manager Profile, Customer Profile
- **Record-Level Security**: Implemented using Sharing Rules and Owner-based Access for internal users; customers can only view their own orders and reviews
- **Field-Level Security**: Protects sensitive information such as cost prices, vendor details, and customer payment information
- **Component Visibility**: Managed in Experience Builder using login-based component visibility for product catalogs, shopping cart, and order history

---

Summary

This phase established a comprehensive understanding of fashion retail business needs, challenges, and goals. Through detailed requirement analysis and user mapping, a clear project scope and feature set were defined. The data model and security framework were designed to ensure scalability, automation readiness, and compliance with retail data privacy standards and customer protection regulations.

Phase 2: Salesforce Development – Backend & Configurations

The Backend & Configuration phase established the functional foundation of the HandsMen Threads E-Commerce Platform.

This phase focused on configuring data models, creating automation through Flows and Apex, and ensuring secure handling of product management, order processing, inventory tracking, and customer engagement.

Using Salesforce declarative tools (Flows, Approval Processes, Validation Rules) alongside programmatic logic (Apex Classes, Triggers, and Async Apex), the system enables seamless coordination between product catalogs, customer orders, inventory management, and

fulfillment operations while maintaining visibility and operational efficiency for sales teams and management.

Milestone 1: Salesforce Account

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?".

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Activity 1: Creating Developer Account

Creating a developer org in Salesforce.

1. Go to https://developer.salesforce.com/signup
2. On the sign-up form, enter the following details:
    - First name & Last name
    - Email
    - Job Title: Developer
    - Company: College Name
    - Country: India
3. Click on "Sign me up" after filling these.

Activity 2: Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 10-30 mins and sometimes 2 hours.
2. Click on "Verify Account"
3. Give a password and answer a security question and click on "Change password."
4. Then you will be redirected to your Salesforce setup page.

Milestone 2: Objects Creation

Activity 1: Creating a Product Object

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object
4. Enter the label name as **Product**
5. Enter Plural label name as **Products**
6. Enter Record Name as **Product Code**
7. Select Data Type as **Auto Number: P-{00000}**, Starting with '1'
8. Select **Allow reports**
9. Select **Allow search**
10. Allow **Track Field History**
11. Click on **Save and New**

Activity 2: Creating an Order Object

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object
4. Enter the label name as **Order**
5. Enter Plural label name as **Orders**
6. Enter Record Name as **Order Number**
7. Select Data Type as **Auto Number: ORD-{00000}**, Starting with '1'
8. Select **Allow reports**
9. Select **Allow search**
10. Allow **Track Field History**
11. Click on **Save and New**

Activity 3: Creating an Order Item Object

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object
4. Enter the label name as **Order Item**
5. Enter Plural label name as **Order Items**
6. Enter Record Name as **Order Item ID**
7. Select Data Type as **Auto Number: OI-{0000}**, Starting with '1'
8. Select **Allow reports**
9. Select **Allow search**
10. Allow **Track Field History**
11. Click on **Save and New**

Activity 4: Creating a Customer Review Object

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object
4. Enter the label name as **Customer Review**
5. Enter Plural label name as **Customer Reviews**
6. Enter Record Name as **Review ID**
7. Select Data Type as **Auto Number: REV-{0000}**, Starting with '1'
8. Select **Allow reports**
9. Select **Allow search**
10. Allow **Track Field History**
11. Click on **Save and New**

Activity 5: Creating an Inventory Object

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object
4. Enter the label name as **Inventory**
5. Enter Plural label name as **Inventory**
6. Enter Record Name as **Inventory ID**
7. Select Data Type as **Auto Number: INV-{0000}**, Starting with '1'
8. Select **Allow reports**
9. Select **Allow search**
10. Allow **Track Field History**
11. Click on **Save and New**

Activity 6: Creating a Shipping Object

To create an object:

1. From the setup page
2. Click on Object Manager
3. Click on Create >> Click on Custom Object
4. Enter the label name as **Shipping**
5. Enter Plural label name as **Shipping Records**
6. Enter Record Name as **Shipping ID**
7. Select Data Type as **Auto Number: SHP-{0000}**, Starting with '1'
8. Select **Allow reports**
9. Select **Allow search**
10. Allow **Track Field History**
11. Click on **Save**

Milestone 3: Tabs

Activity 1: Creating a tab for Product Object

1. Go to the setup page → type **Tabs** in Quick Find bar
2. Click on **Tabs**
3. Click on **New** (under custom object tab)
4. Select Object (**Product**) >> Select the tab style
5. Click on **Next** >> (Add to profiles page) keep it as default >> Click on **Next** (Add to Custom App) uncheck the include tab
6. Make sure that the **Append tab to the user's existing personal customizations** is checked
7. Click **Save**
8. **Create tabs for every object created in Milestone-2**

Milestone 4: Fields & Relationships

Object	Field Name	Data Type	Required
Product__c	Product Code	Auto Number: P-{00000}	Yes
	Product Name	Text (80)	Yes
	Description	Long Text Area	No
	Category	Picklist (Formal Wear, Casual Wear, Accessories, Footwear)	Yes
	Price	Currency (2 decimal places)	Yes
	Size	Picklist (S, M, L, XL, XXL, XXXL)	Yes
	Color	Text (40)	No
	Brand	Text (80)	No
	Material	Text (80)	No
	Stock Available	Checkbox	Yes
	Image URL	URL	No
	Discount Percentage	Percent	No
Order__c	Order Number	Auto Number: ORD-{00000}	Yes
	Customer	Lookup (Contact)	Yes
	Order Date	Date	Yes
	Order Status	Picklist (Pending, Processing, Shipped, Delivered, Cancelled)	Yes
	Total Amount	Currency	Yes
	Payment Method	Picklist (Credit Card, Debit Card, UPI, Cash on Delivery)	Yes
	Shipping Address	Long Text Area	Yes
	Order Notes	Long Text Area	No
Order_Item__c	Order Item ID	Auto Number: OI-{0000}	Yes
	Order	Lookup (Order__c)	Yes
	Product	Lookup (Product__c)	Yes

Object	Field Name	Data Type	Required
	Quantity	Number	Yes
	Unit Price	Currency	Yes
	Subtotal	Formula (Currency): Quantity * Unit_Price	Auto-calculated
	Discount Applied	Currency	No
Customer_Review__c	Review ID	Auto Number: REV-{0000}	Yes
	Product	Lookup (Product__c)	Yes
	Customer	Lookup (Contact)	Yes
	Rating	Picklist (1, 2, 3, 4, 5)	Yes
	Review Text	Long Text Area	No
	Review Date	Date	Yes
	Verified Purchase	Checkbox	No
Inventory__c	Inventory ID	Auto Number: INV-{0000}	Yes
	Product	Lookup (Product__c)	Yes
	Current Stock	Number	Yes
	Reorder Level	Number	Yes
	Warehouse Location	Text (80)	Yes
	Last Restock Date	Date	No
Shipping__c	Shipping ID	Auto Number: SHP-{0000}	Yes
	Order	Lookup (Order__c)	Yes
	Shipping Carrier	Picklist (FedEx, DHL, Blue Dart, India Post)	Yes
	Tracking Number	Text (80)	No
	Estimated Delivery	Date	Yes
	Delivery Status	Picklist (Pending, In Transit, Out for Delivery, Delivered)	Yes

---

Activity 1: Creation of Picklist field for the Product object

Create Picklist Field in a Product object:

1. Go to setup >> click on Object Manager >> type object name (**Product**) in quick find box >> click on the **Product** Object
2. Now click on "**Fields & Relationships**"
3. Click on **New**
4. Select Data type as "**Picklist**" and click **Next**
5. Enter the Field Label as "**Category**"

6. Click on **Enter values** and Enter: **Formal Wear, Casual Wear, Accessories, Footwear**
7. Click on **Next**, **Next** and **Save**

Activity 2: Creation of Lookup field for the Order Object

Creating Lookup Relationship in Order Object:

1. Go to the Setup page >> click on Object manager >> type object name (**Order**) in the quick find bar >> click on the **Order** object
2. Click on **Fields & Relationship**
3. Click on **New**
4. Select "**Lookup relationship**" as data type and click **Next**
5. Select the related object "**Contact**"
6. Click on **Next**
7. Give Field Label as "**Customer**"
8. Click on **Next**, **Next**, **Next**, **Save**

Note: Create other fields from the above Fields table related to this object and choose the data types of the fields carefully.

Milestone 5: Validation Rules

Activity 1: To create a Validation rule for the Order object

Validation Rule on Order Total Amount:

1. Go to setup >> click on Object Manager >> type object name (**Order**) in search bar >> click on the **Order** object
2. Click on the **Validation Rules** >> click on **New**
3. Enter the Rule name as "**Positive Order Amount**"
4. Select **Active**
5. Insert the Error Condition Formula as: `Total_Amount__c <= 0`
6. Enter the Error Message as "**Order total must be greater than zero.**"
7. Select the Error location as **Top of the page**
8. Click **Save**

Activity 2: To create a Validation rule for the Product object

Validation Rule on Product Price:

1. Go to setup >> click on Object Manager >> type object name (**Product**) in search bar >> click on the **Product** object
2. Click on the **Validation Rules** >> click on **New**
3. Enter the Rule name as "**Valid Product Price**"
4. Select **Active**
5. Insert the Error Condition Formula as: `Price__c <= 0`
6. Enter the Error Message as "**Product price must be greater than zero.**"

7. Select the Error location as **Top of the page**
8. Click **Save**

Activity 3: To create a Validation rule for the Inventory object

Validation Rule on Inventory Stock Levels:

1. Go to setup >> click on Object Manager >> type object name (**Inventory**) in search bar >> click on the **Inventory** object
2. Click on the **Validation Rules** >> click on **New**
3. Enter the Rule name as "**Non-Negative Stock**"
4. Select **Active**
5. Insert the Error Condition Formula as: `Current_Stock__c < 0`
6. Enter the Error Message as "**Current stock cannot be negative.**"
7. Select the Error location as **Top of the page**
8. Click **Save**

Milestone 6: APPROVAL PROCESS

Activity 1: Create an Approval Process for Discount Approval

Note: Before Implementing Approval Process First Complete Profiles, Roles and Users Milestones.

1. Create Classic Email Template

- Go to **Setup** (gear icon on top-right)
- In the Quick Find box, type: **Email Templates**
- Click on **Classic Email Templates**
- Click "**New Template**"

Choose the type:

- **Text** – Plain text (no formatting)
- Click **Next**

Email Template Information:

Field	Value
Folder	Public or private folder (use "Unfiled Public Email Templates" for now)
Email Template Name	Product Discount Approval Notification
Template Unique Name	Auto-filled, or change it (no spaces)
Encoding	Leave default (UTF-8)
Subject	New Discount Request Awaiting Approval

Field	Value
Email Body	Dear {!Product__c.Approver__c}, A new product discount request has been submitted and is awaiting your approval. Please review the details below: **Request Details:** - Product Name: {!Product__c.Product_Name__c} - Current Price: {!Product__c.Price__c} - Requested Discount: {!Product__c.Discount_Percentage__c}% - Reason: {!Product__c.Discount_Reason__c} - Submitted By: {!Product__c.CreatedBy} - Submission Date: {!Product__c.CreatedDate} If you have any questions or need additional information, please contact the requester directly. Thank you for your prompt attention. Best regards, The HandsMen Threads Team

2. Create the Approval Process

- In the Quick Find box, type: **Approval Processes**
- Click on "**Approval Processes**" under Process Automation
- Click on the object name: **Product**
- Click **Create New Approval Process → Use Standard Setup Wizard**

Step 1: Basic Settings

- Name: **Approval Process for Product Discount**
- Unique Name: Auto-fills (you can keep it as is)
- Click **Next**

Specify Entry Criteria:

This defines when the process will be triggered.

- Choose "**Criteria are met**"

Set:

- Field: **Product → Approval Status**
- Operator: **Equals**
- Value: **Pending**

Step 2: Select Approver

- Select **Administrators ONLY can edit records during the approval process**

Step 3: Email Template

Approval Assignment Email Template:

- Select Email Template: **Product Discount Approval Notification**

Step 4: Select Fields to Display on Approval Page Layout

You'll see a dual list:

- Available Fields on the left
- Selected Fields on the right

Choose the fields you want approvers to see (typically key business info).

Suggested Fields:

- Product Code
- Product Name
- Owner
- Price
- Discount Percentage
- Discount Reason
- Use the **Add** button to move fields from left to right
- Use the **Up/Down** arrows to reorder how they appear
- Click **Next** to proceed

Step 5: Specify Initial Submitters

- Search for **owner** and add **Product owner** to the selected submitters from available submitters
- Click **Save**

Step 6: Final Approval Actions

- Click **Add New → Field Update**
- Select Field to Update as **Approval_Status**
- Name: **Set Approval Status to Approved**
- Field: **Approval_Status__c**
- New Value: **Approved**
- **Save**

Step 7: Final Rejection Actions

- Click **Add New → Field Update**
- Name: **Approval Status to Rejected**

- Field: **Approval_Status__c**
- New Value: **Rejected**
- **Save**

Step 8: Activate the Approval Process

- Click **View Approval Process Detail Page**
- Click the "**Activate**" button at the top

Activity 2: Create an Approval Process for Bulk Orders

Note: Before Implementing Approval Process First Complete Profiles, Roles and Users Milestones.

1. Create Classic Email Template

- Go to **Setup** (gear icon on top-right)
- In the Quick Find box, type: **Email Templates**
- Click on **Classic Email Templates**
- Click "**New Template**"

Choose the type:

- **Text** – Plain text (no formatting)
- Click **Next**

Email Template Information:

Field	Value
Folder	Public or private folder (use "Unfiled Public Email Templates" for now)
Email Template Name	Bulk Order Approval Notification
Template Unique Name	Auto-filled, or change it (no spaces)
Encoding	Leave default (UTF-8)
Subject	New Bulk Order Request for Approval
Email Body	

2. Create the Approval Process

- In the Quick Find box, type: **Approval Processes**
- Click on "**Approval Processes**" under Process Automation
- Click on the object name: **Order**
- Click **Create New Approval Process → Use Standard Setup Wizard**

Step 1: Basic Settings

- Name: **Approval Process for Bulk Orders**
- Unique Name: Auto-fills (you can keep it as is)
- Click **Next**

Specify Entry Criteria:

This defines when the process will be triggered.

- Choose "**Criteria are met**"

Set:

- Field: **Order → Total Amount**
- Operator: **Greater Than**
- Value: **50000** (or your bulk order threshold)

AND

- Field: **Order → Approval Status**
- Operator: **Equals**
- Value: **Pending**

Step 2: Select Approver

- Select **Administrators OR the currently assigned approver can edit records during the approval process**

Step 3: Email Template

Approval Assignment Email Template:

- Select Email Template: **Bulk Order Approval Notification**

Step 4: Select Fields to Display on Approval Page Layout

Suggested Fields:

- Order Number
- Owner
- Customer
- Total Amount
- Order Date
- Payment Method
- Use the **Add** button to move fields from left to right
- Use the **Up/Down** arrows to reorder how they appear
- Check **Display approval history information in addition to the fields selected above** checkbox

- Select **Allow approvers to access the approval page only from within the Salesforce application. (Recommended)**
- Click **Next** to proceed

Step 5: Specify Initial Submitters

- Search for **owner** and add **Order owner** to the selected submitters from available submitters
- Click **Save**

Step 6: Final Approval Actions

- Click **Add New → Field Update**
- Select Field to Update as **Approval_Status**
- Name: **Set Approval Status to Approved**
- Field: **Approval_Status__c**
- New Value: **Approved**
- **Save**

Step 7: Final Rejection Actions

- Click **Add New → Field Update**
- Name: **Approval Status to Rejected**
- Field: **Approval_Status__c**
- New Value: **Rejected**
- **Save**

Step 8: Activate the Approval Process

- Click **View Approval Process Detail Page**
- Click the "**Activate**" button at the top

Milestone 7: Apex Classes

Activity 1: Create an Apex class ProductController

Create Apex Class

Retrieves product catalog and manages product information.

1. Click **Gear Icon** and Select **Developer Console**
2. Click on **File** and Click **New** and Click on **Apex class**
3. Give name as **ProductController** and Click **Ok**
4. Write whole code in the class

Source Code:

apex

```
public class ProductController {
    @AuraEnabled(cacheable=true)
    public static List<Product__c> getProducts(String category) {
        if (String.isBlank(category)) {
            return [SELECT Id, Product_Name__c, Description__c, Price__c,
                    Category__c, Size__c, Color__c, Brand__c, Image_URL__c,
                    Stock_Available__c, Discount_Percentage__c
                    FROM Product__c
                    WHERE Stock_Available__c = true
                    ORDER BY Product_Name__c];
        } else {
            return [SELECT Id, Product_Name__c, Description__c, Price__c,
                    Category__c, Size__c, Color__c, Brand__c, Image_URL__c,
                    Stock_Available__c, Discount_Percentage__c
                    FROM Product__c
                    WHERE Category__c = :category AND Stock_Available__c = true
                    ORDER BY Product_Name__c];
        }
    }

    @AuraEnabled(cacheable=true)
    public static Product__c getProductDetails(Id productId) {
        return [SELECT Id, Product_Name__c, Description__c, Price__c,
                Category__c, Size__c, Color__c, Brand__c, Material__c,
                Image_URL__c, Stock_Available__c, Discount_Percentage__c
                FROM Product__c
                WHERE Id = :productId
                LIMIT 1];
    }
}
```

Save the class using CTRL+S

Activity 2: Create an Apex class OrderController

Create Apex Class

Handles order creation and order item management.

1. Click **Gear Icon** and Select **Developer Console**

2. Click on **File** and Click **New** and Click on **Apex class**
3. Give name as **OrderController** and Click **Ok**
4. Write whole code in the class

Source Code:

apex

```
public class OrderController {
    @AuraEnabled
    public static Id createOrder(Order__c order, List<Order_Item__c> orderItems) {
        try {
            // Insert the order
            insert order;

            // Associate order items with the order
            for (Order_Item__c item : orderItems) {
                item.Order__c = order.Id;
            }

            // Insert order items
            insert orderItems;

            // Update inventory
            updateInventory(orderItems);

            return order.Id;
        } catch (Exception e) {
            throw new AuraHandledException('Error creating order: ' + e.getMessage());
        }
    }

    @AuraEnabled(cacheable=true)
    public static List<Order__c> getCustomerOrders() {
        Id contactId = [SELECT ContactId FROM User WHERE Id = :UserInfo.getUserId() LIMIT 1].ContactId;

        return [SELECT Id, Order_Number__c, Order_Date__c, Order_Status__c,
                Total_Amount__c, Payment_Method__c
                FROM Order__c
                WHERE Customer__c = :contactId
                ORDER BY Order_Date__c DESC];
    }
```

```
    private static void updateInventory(List<Order_Item__c> orderItems) {
        Set<Id> productIds = new Set<Id>();
        Map<Id, Decimal> productQuantityMap = new Map<Id, Decimal>();

        for (Order_Item__c item : orderItems) {
            productIds.add(item.Product__c);
            productQuantityMap.put(item.Product__c, item.Quantity__c);
        }

        List<Inventory__c> inventoryList = [SELECT Id, Product__c, Current_Stock__c
                                            FROM Inventory__c
                                            WHERE Product__c IN :productIds];

        for (Inventory__c inv : inventoryList) {
            if (productQuantityMap.containsKey(inv.Product__c)) {
                inv.Current_Stock__c -= productQuantityMap.get(inv.Product__c);
            }
        }

        update inventoryList;
    }
}
```

Save the class using CTRL+S

Activity 3: Create an Apex class CustomerReviewController

Create Apex Class

Used for capturing and displaying customer reviews.

1. Click **Gear Icon** and Select **Developer Console**
2. Click on **File** and Click **New** and Click on **Apex class**
3. Give name as **CustomerReviewController** and Click **Ok**
4. Write whole code in the class

Source Code:

apex

```
public class CustomerReviewController {
    @AuraEnabled
```

```
    public static Id submitReview(Customer_Review__c review) {
        try {
            insert review;
            return review.Id;
        } catch (Exception e) {
            throw new AuraHandledException('Error submitting review: ' + e.getMessage());
        }
    }

    @AuraEnabled(cacheable=true)
    public static List<Customer_Review__c> getProductReviews(Id productId) {
        return [SELECT Id, Customer__r.Name, Rating__c, Review_Text__c,
                Review_Date__c, Verified_Purchase__c
                FROM Customer_Review__c
                WHERE Product__c = :productId
                ORDER BY Review_Date__c DESC];
    }

    @AuraEnabled(cacheable=true)
    public static Decimal getAverageRating(Id productId) {
        AggregateResult[] results = [SELECT AVG(Rating__c) avgRating
                                     FROM Customer_Review__c
                                     WHERE Product__c = :productId];

        if (results.size() > 0 && results[0].get('avgRating') != null) {
            return (Decimal)results[0].get('avgRating');
        }
        return 0;
    }
}
```

Save the class using CTRL+S

Milestone 8: Asynchronous Apex

Activity 1: Create an Async Apex class for Low Stock Alerts

Create an Apex class:

apex

```
public class SendLowStockAlerts implements Queueable {
    public void execute(QueueableContext context) {
        // Query inventory records where stock is below reorder level
        List<Inventory__c> lowStockItems = [SELECT Id, Product__r.Product_Name__c,
                                            Current_Stock__c, Reorder_Level__c,
                                            Warehouse_Location__c
                                            FROM Inventory__c
                                            WHERE Current_Stock__c < Reorder_Level__c];

        if (lowStockItems.isEmpty()) {
            return;
        }

        // Get inventory managers
        List<User> managers = [SELECT Id, Email FROM User
                               WHERE Profile.Name = 'Inventory Manager Profile'
                               AND IsActive = TRUE];

        Messaging.SingleEmailMessage[] emails = new List<Messaging.SingleEmailMessage>();

        for (User manager : managers) {
            Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
            mail.setToAddresses(new String[]{manager.Email});
            mail.setSubject('Low Stock Alert - Action Required');

            String emailBody = 'Dear Manager,\n\nThe following products are below their reorder levels:\n\n';

            for (Inventory__c item : lowStockItems) {
                emailBody += 'Product: ' + item.Product__r.Product_Name__c + '\n';
                emailBody += 'Current Stock: ' + item.Current_Stock__c + '\n';
                emailBody += 'Reorder Level: ' + item.Reorder_Level__c + '\n';
                emailBody += 'Location: ' + item.Warehouse_Location__c + '\n\n';
            }

            emailBody += 'Please take necessary action to restock these items.\n\nBest regards,\nHandsMen Threads
Inventory System';

            mail.setPlainTextBody(emailBody);
            emails.add(mail);
        }
```

```
        Messaging.sendEmail(emails);
    }
}
```

Save the class using CTRL+S

Milestone 9: Email Templates

Activity 1: Create an Order Confirmation Email Template

1. Go to **Setup** (gear icon on top-right)
2. In the Quick Find box, type **Email Templates**
3. Click on **Classic Email Templates**
4. Click **New Template**

Choose the type:

- **Text** – Plain text (no formatting)
- Click **Next**

Template Information:

Field	Value
Template Name	Order Confirmation
Subject	Your HandsMen Threads Order Confirmation - {!Order__c.Order_Number__c}
Email Body	Dear {!Order__c.Customer__c}, Thank you for your order with HandsMen Threads! **Order Details:** - Order Number: {!Order__c.Order_Number__c} - Order Date: {!Order__c.Order_Date__c} - Total Amount: ₹{!Order__c.Total_Amount__c} - Payment Method: {!Order__c.Payment_Method__c} - Order Status: {!Order__c.Order_Status__c} **Shipping Address:** {!Order__c.Shipping_Address__c} We are processing your order and will notify you once it ships. You can track your order status in your customer portal. Thank you for choosing HandsMen Threads - Where sophistication meets

Field	Value
	style. Best regards, HandsMen Threads Team

Click Save

Activity 2: Create a Shipping Notification Email Template

1. Go to **Setup** (gear icon on top-right)
2. In the Quick Find box, type **Email Templates**
3. Click on **Classic Email Templates**
4. Click **New Template**

Choose the type:

- **Text** – Plain text (no formatting)
- Click **Next**

Template Information:

Field	Value
Template Name	Shipping Notification
Subject	Your Order Has Shipped! - {!Order__c.Order_Number__c}
Email Body	Dear {!Order__c.Customer__c}, Great news! Your order has been shipped. **Shipping Details:** - Order Number: {!Order__c.Order_Number__c} - Shipping Carrier: {!Shipping__c.Shipping_Carrier__c} - Tracking Number: {!Shipping__c.Tracking_Number__c} - Estimated Delivery: {!Shipping__c.Estimated_Delivery__c} You can track your shipment using the tracking number provided above on the carrier's website. Expected delivery: {!Shipping__c.Estimated_Delivery__c} Thank you for shopping with HandsMen Threads! Best regards, HandsMen Threads Team

Click Save

Activity 3: Create a Product Review Request Email Template

1. Go to **Setup** (gear icon on top-right)
2. In the Quick Find box, type **Email Templates**
3. Click on **Classic Email Templates**
4. Click **New Template**

Choose the type:

- **Text** – Plain text (no formatting)
- Click **Next**

Template Information:

Field	Value
Template Name	Product Review Request
Subject	Share Your Experience with HandsMen Threads
Email Body	Dear {!Order__c.Customer__c}, We hope you're enjoying your recent purchase from HandsMen Threads! Your feedback is valuable to us and helps other customers make informed decisions. Would you take a moment to share your experience? **Your Order:** {!Order__c.Order_Number__c} **Order Date:** {!Order__c.Order_Date__c} Please visit our customer portal to leave a review for the products you purchased. As a token of our appreciation, you'll receive a 10% discount code on your next purchase after submitting your review! Thank you for being a valued customer. Best regards, HandsMen Threads Team

Click Save

Milestone 10: Declarative Automation (Flows)

Activity 1: Order Confirmation Flow

Objective: Automatically send order confirmation email when a new order is created.

Steps:

1. From Setup, in the Quick Find box, search for **Flows**
2. Click **New Flow**
3. Select Category: **Triggered**
4. Select Type: **Record-Triggered Flow**
5. Select Object: **Order__c**
6. Configure Trigger: **When a record is created**
7. Set Entry Conditions: **Order_Status__c = Pending**
8. Run **Every time a record is created and meets the condition requirements**
9. Optimize flow for **Actions and Related Records**
10. Click on **+** Icon and add an **Action Element**
11. For Label, enter: **Send Order Confirmation Email**
12. Select Email Template: **Order Confirmation**
13. Map **Customer__c** to the recipient field
14. **Save the Flow and Activate**

Activity 2: Inventory Update Flow

Objective: Automatically update product availability when inventory falls below threshold.

Steps:

1. From Setup, in the Quick Find box, search for **Flows**
2. Click **New Flow**
3. Select Category: **Triggered**
4. Select Type: **Record-Triggered Flow**
5. Select Object: **Inventory__c**
6. Configure Trigger: **When a record is updated**
7. Set Entry Conditions: **Current_Stock__c Is Changed = True AND Current_Stock__c < Reorder_Level__c**
8. Run **Every time a record is updated and meets the condition requirements**
9. Optimize flow for **Actions and Related Records**
10. Click on **+** Icon and add a **Get Records** element to fetch the related Product
11. Add an **Update Records** element
12. Update **Product__c.Stock_Available__c = False** when inventory is below reorder level
13. Add an **Action** element to send email notification to Inventory Manager
14. **Save the Flow and Activate**

Activity 3: Low Rating Alert Flow

Objective: Automatically notify product managers when a review with rating ≤ 2 is submitted.

Steps:

1. From Setup, in the Quick Find box, search for **Flows**
2. Click **New Flow**
3. Select Category: **Triggered**
4. Select Type: **Record-Triggered Flow**
5. Select Object: **Customer_Review__c**
6. Configure Trigger: **When a record is created**
7. Set Entry Conditions: **Rating__c Less Than or Equal 2**
8. Run **Every time a record is created and meets the condition requirements**
9. Optimize flow for **Actions and Related Records**
10. Click on + Icon and add an **Action Element**
11. For Label, enter: **Send Low Rating Alert**
12. Create email alert with product details and review information
13. Send to product managers and retail managers
14. **Save the Flow and Activate**

Summary

This phase successfully established the backend for managing men's fashion retail operations.

With well-defined data models, Apex logic, declarative automation, and security controls, the system now supports real-time product management, order processing, inventory tracking, customer reviews, and shipping management.

This foundation ensures that future phases (UI development with LWC and Experience Site deployment) will integrate seamlessly with existing logic and data structures, creating a comprehensive e-commerce platform for HandsMen Threads.

Phase 3: UI/UX Development & Customization

The UI/UX Development & Customization phase focused on building an intuitive and responsive interface for the HandsMen Threads E-Commerce Platform. The goal was to deliver a seamless shopping experience for customers and an efficient management interface for sales teams, inventory managers, and administrators to access and manage products, orders, inventory, reviews, and shipping information.

During this phase, Lightning Application and Page Layouts were created and customized to ensure smooth navigation and a connected design aligned with the HandsMen Threads brand identity and sophisticated aesthetic.

Milestone 11: The Lightning App

Activity 1: Create a Lightning App for "HandsMen Threads Portal"

1. From Setup, enter **App Manager** in the Quick Find and select **App Manager**
2. Click **New Lightning App**
3. Enter **HandsMen Threads Portal** as the App Name
4. Click on **upload image** and add a logo image related to men's fashion/HandsMen Threads brand
5. Click **Next**
6. Under App Options, leave the default selections and click **Next**
7. Under Utility Items, leave as is and click **Next**
8. From Available Items, select:
    - **Product__c**
    - **Order__c**
    - **Order_Item__c**
    - **Customer_Review__c**
    - **Inventory__c**
    - **Shipping__c**
    - **Contact**
    - **Account**
9. Move them to Selected Items and Click **Next**
10. From Available Profiles, select **System Administrator**, **Retail Manager Profile**, **Sales Representative Profile**, **Inventory Manager Profile** and move them to Selected Profiles
11. Click **Save & Finish**

Milestone 12: Editing of Page Layouts

Activity 1: To edit a Page Layout in Product__c Object

1. Go to setup >> click on Object Manager >> type object name (**Product**) in quick find box >> click on the **Product** object >> **Page Layouts**
2. Click on the **Product Layout**
3. Drag and arrange fields in logical sections:
    - **Product Information**: Product Code, Product Name, Brand, Category
    - **Pricing Details**: Price, Discount Percentage
    - **Product Specifications**: Size, Color, Material, Description
    - **Availability**: Stock Available, Image URL
4. Click on **Save**

Activity 2: To edit a Page Layout in Order__c Object

1. Go to setup >> click on Object Manager >> type object name (**Order**) in quick find box >> click on the **Order** object >> **Page Layouts**
2. Click on the **Order Layout**

3. Drag and arrange fields in logical sections:
    - **Order Information**: Order Number, Customer, Order Date, Order Status
    - **Financial Details**: Total Amount, Payment Method
    - **Shipping Information**: Shipping Address, Order Notes
4. Add **Related Lists**: Order Items, Shipping Records
5. Click **Save**

Activity 3: To edit a Page Layout in Order_Item__c Object

1. Go to setup >> click on Object Manager >> type object name (**Order Item**) in quick find box >> click on the **Order Item** object >> **Page Layouts**
2. Click on the **Order Item Layout**
3. Drag and arrange fields:
    - **Item Information**: Order Item ID, Order, Product
    - **Pricing Details**: Quantity, Unit Price, Subtotal, Discount Applied
4. Click **Save**

Activity 4: To edit a Page Layout in Customer_Review__c Object

1. Go to setup >> click on Object Manager >> type object name (**Customer Review**) in quick find box >> click on the **Customer Review** object >> **Page Layouts**
2. Click on the **Customer Review Layout**
3. Drag and arrange fields:
    - **Review Information**: Review ID, Product, Customer
    - **Review Details**: Rating, Review Date, Review Text, Verified Purchase
4. Click **Save**

Activity 5: To edit a Page Layout in Inventory__c Object

1. Go to setup >> click on Object Manager >> type object name (**Inventory**) in quick find box >> click on the **Inventory** object >> **Page Layouts**
2. Click on the **Inventory Layout**
3. Drag and arrange fields:
    - **Inventory Information**: Inventory ID, Product
    - **Stock Details**: Current Stock, Reorder Level, Warehouse Location, Last Restock Date
4. Click **Save**

Activity 6: To edit a Page Layout in Shipping__c Object

1. Go to setup >> click on Object Manager >> type object name (**Shipping**) in quick find box >> click on the **Shipping** object >> **Page Layouts**

2. Click on the **Shipping Layout**
3. Drag and arrange fields:
    - **Shipping Information**: Shipping ID, Order, Shipping Carrier
    - **Tracking Details**: Tracking Number, Estimated Delivery, Delivery Status
4. Click **Save**

Milestone 13: Dynamic Forms

Activity 1: To create a Dynamic Form in Product Object

1. Go to setup >> click on **App Launcher** >> Open "**HandsMen Threads Portal**" App >> click on the **Product** object tab >> Click on **New** and create a new record and save it
2. Click on the record created and click on the **Gear icon** on the top right corner and Select **Edit Page**
3. Click on the **Details** section and on the right pane click on **Upgrade Now** to enable Dynamic Forms for the object
4. Select **Product Layout** and Click on **Finish**
5. Click on **Save** and Click on **Activate**
6. Click on **Org as Default**, select **Desktop** and **Phone**, click **Next** and Click **Save**
7. **Do the same for remaining objects as well** (Order, Order Item, Customer Review, Inventory, Shipping)

Milestone 14: Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Activity 1: Create User - Retail Manager

Note: Before Implementing Users first complete Profiles and Roles Milestones.

1. Go to setup >> type **users** in quick find box >> select **users** >> click **new user**
2. Fill in the fields:
    - **First Name**: HandsMen
    - **Last Name**: Manager
    - **Alias**: Give an Alias Name
    - **Email id**: Give your Personal Email id
    - **Username**: Username should be in this form: text@text.text and it must be Unique
    - **Nick Name**: Give a Nickname
    - **Role**: Retail Manager
    - **User license**: Salesforce
    - **Profile**: Retail Manager Profile

3. **Save**

Activity 2: Creating Other Users

1. Repeat the steps and create other users:

Sales Representative:

- **Role**: Sales Representative
- **User license**: Salesforce
- **Profile**: Sales Representative Profile

Inventory Manager:

- **Role**: Inventory Manager
- **User license**: Salesforce
- **Profile**: Inventory Manager Profile

Customer User (Optional):

- **Role**: Customer
- **User license**: Customer Community Plus or Similar
- **Profile**: Customer Profile

Phase 4: Data Migration, Testing & Security

Data Migration, Testing & Security focuses on migrating data from legacy systems or spreadsheets into Salesforce, ensuring data integrity, and configuring security measures for the HandsMen Threads E-Commerce Platform. This phase also involves testing all functionalities, including product management, order processing, inventory tracking, customer reviews, and shipping management. Security settings such as role-based access, sharing rules, and field-level security are applied to protect sensitive customer and business information and ensure compliance with data protection regulations.

Performance Tuning & Optimization

This phase also focused on optimizing performance, ensuring smooth operation of the portal even with multiple concurrent users browsing products, placing orders, and accessing inventory information.

---

1. Data Migration

The HandsMen Threads Platform requires migration of product catalogs, customer data, order history, inventory records, customer reviews, and shipping information from legacy systems or spreadsheets into Salesforce.

Steps Followed:

- Imported customer records (Contact and Account data) using **Data Import Wizard** and **Data Loader**
- Imported related records: **Product__c**, **Order__c**, **Order_Item__c**, **Customer_Review__c**, **Inventory__c**, and **Shipping__c**
- Ensured unique identifiers (Product Code, Order Number, Customer Email) to prevent duplicates
- Validated lookup relationships between customers, orders, products, reviews, inventory, and shipping records
- Verified data integrity using reports and dashboards after migration

Data Validations:

- Mandatory fields (Product Name, Price, Customer Name, Order Total) verified for completeness
- Duplicate prevention using unique constraints on Product Code, Order Number, and Customer Email
- Validation Rules and Before Save Flows ensured accurate data entry for products, orders, inventory, and reviews
- Price validation to ensure all prices are positive values
- Stock level validation to prevent negative inventory

---

2. Testing

Testing ensured all objects, Flows, Apex logic, and user interfaces performed correctly and integrated seamlessly.

Testing Types Conducted:

Unit Testing

- Verified Apex classes and Flows for Product management, Order processing, Inventory updates, Review submission, and Shipping tracking
- Checked best practices and governor limits compliance
- Ensured test classes achieved 100% coverage for triggers and Apex handlers
- Tested edge cases like zero inventory, invalid pricing, and duplicate orders

Integration Testing

- Verified relationships between Contact/Account records and related **Order__c**, **Order_Item__c**, **Product__c**, **Customer_Review__c**, **Inventory__c**, **Shipping__c**
- Tested interaction between Flows, page layouts, and backend Apex logic
- Validated order total calculations and inventory deductions

- Tested approval processes for discounts and bulk orders

User Acceptance Testing (UAT)

Conducted with sample customer, sales representative, and manager users to simulate real-life scenarios:

Customer Scenarios:

- Browsing product catalog by category
- Adding products to cart and placing orders
- Tracking order status and shipments
- Submitting product reviews

Sales Representative Scenarios:

- Creating orders on behalf of customers
- Processing bulk orders
- Managing customer inquiries

Manager Scenarios:

- Approving discount requests
- Reviewing sales dashboards
- Monitoring inventory levels

Collected feedback and optimized page layouts, validation messages, and navigation for better usability.

Outcome:

All test cycles confirmed functional reliability, data accuracy, smooth order processing, and seamless user interaction across desktop and mobile devices.

---

3. Security Implementation

Security ensured only authorized personnel and customers could access appropriate records and sensitive information.

Role Hierarchy

- **Admin** – Full access to all records and configuration
- **Retail Manager** – Access to all products, orders, inventory, and sales analytics
- **Sales Representative** – Access to customer orders, products, and customer data
- **Inventory Manager** – Access to inventory records, product availability, and stock management
- **Customer** – Access limited to their own orders, reviews, and shipping information

Profiles & Permission Sets

- **Profiles** for Admin, Retail Manager, Sales Representative, Inventory Manager, Customer
- **Permission sets** for additional access to discount approvals, bulk order processing, and inventory management
- Field-level permissions to hide cost prices and vendor information from customers

Field-Level Security

- Sensitive fields (e.g., Cost Price, Vendor Information, Customer Payment Details) hidden for customers and sales representatives but visible to Managers and Admins
- Product pricing and discount details protected at field level
- Customer contact information secured for privacy compliance

Record-Level Security

- **OWD = Private** for **Order__c**, **Customer_Review__c**, **Shipping__c**
- **OWD = Public Read Only** for **Product__c**, **Inventory__c**
- **Sharing Rules**: Managers get read/write access to manage customer orders and reviews
- Customers can only access their own orders, reviews, and shipping records
- Sales representatives can view and edit orders for their assigned customers

---

Outcome of Phase 4

- Data security, access control, customer privacy protection, and compliance verified before deployment
- Platform ready for customer-facing operations with proper security controls
- All business logic tested and validated for accuracy

Performance Tuning & Optimization

Performance Tuning & Optimization focused on enhancing the overall speed and efficiency of the HandsMen Threads Portal, ensuring smooth performance even with multiple customers browsing products, placing orders, and accessing account information simultaneously.

1. Optimizing SOQL Queries

In Apex handlers and backend logic:

- Retrieved only required fields for product catalog queries
- Used selective filters based on product category and availability
- Applied **LIMIT** and **WHERE** clauses for efficiency
- Indexed frequently queried fields (Product Code, Order Number, Customer Email)

2. Reducing Loops and DML Operations

- Avoided SOQL/DML inside loops in order processing logic
- Batched inventory updates in Apex and Flows
- Optimized order item calculations to process in bulk

3. Page Layout Optimization

- Modular page layouts for Product, Order, Order Item, Customer Review, Inventory, and Shipping
- Minimized unnecessary queries by using formula fields for price calculations and order totals
- Used roll-up summary fields for order item totals and review averages

4. Security & Caching Enhancements

- **FLS checks** in Apex for secure data access
- **Platform Cache** for frequently accessed product catalog and category data
- Cached inventory availability to reduce real-time database queries
- Optimized product image loading for faster page rendering

Milestone 15: Duplicate and Matching Rules

Activity 1: Create a Custom Matching Rule for Products

1. Go to **Setup**
2. In Quick Find, search for **Matching Rules**
3. Click **New Rule**
4. Select Object: **Product** and Click **Next**
5. Enter Rule Name: **Unique Product Code**
6. In Matching Criteria: Select Field as **Product_Code__c**
7. In Matching Method: Select **Exact**
8. Check **Match Blank Fields** (optional - uncheck if you want to allow blank product codes temporarily)
9. Click **Next**
10. Click **Save & Activate**

Activity 2: Create a Duplicate Rule for Products

1. Go to **Setup**
2. Search for **Duplicate Rule**
3. Click **New Rule**
4. Select Object: Same as before (**Product**)
5. Enter Rule Name: **Prevent Duplicate Products**

6. Set **Action on Create** and **Action on Edit**: **Block** (to prevent duplicate product codes)
7. In Alert Text: **A product with this Product Code already exists. Please use a unique Product Code.**
8. In the Matching Rules section:
9. Click **Add Rule**
10. Select your previously created Matching Rule (**Unique Product Code**)
11. Click **Save & Activate**

Activity 3: Create a Custom Matching Rule for Customers

1. Go to **Setup**
2. In Quick Find, search for **Matching Rules**
3. Click **New Rule**
4. Select Object: **Contact** and Click **Next**
5. Enter Rule Name: **Unique Customer Email**
6. In Matching Criteria: Select Field as **Email**
7. In Matching Method: Select **Exact**
8. Check **Match Blank Fields**
9. Click **Next**
10. Click **Save & Activate**

Activity 4: Create a Duplicate Rule for Customers

1. Go to **Setup**
2. Search for **Duplicate Rule**
3. Click **New Rule**
4. Select Object: **Contact**
5. Enter Rule Name: **Prevent Duplicate Customers**
6. Set **Action on Create** and **Action on Edit**: **Allow and Report** (to warn but not block duplicate customer emails)
7. In Alert Text: **A customer with this email already exists. Please verify before creating.**
8. In the Matching Rules section:
9. Click **Add Rule**
10. Select your previously created Matching Rule (**Unique Customer Email**)
11. Click **Save & Activate**

Milestone 16: Profiles

A profile is a group/collection of settings and permissions that define what a user can do in Salesforce. Profile controls "Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges." You can define profiles by the user's job function.

For example, System Administrator, Retail Manager, Sales Representative, Inventory Manager.

Types of profiles in Salesforce:

1. Standard profiles:

By default, Salesforce provides below standard profiles:

- Contract Manager
- Read Only
- Marketing User
- Solutions Manager
- Standard User
- System Administrator

We cannot delete standard ones. Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

2. Custom Profiles:

Custom ones defined by us. They can be deleted if there are no users assigned with that particular one.

Activity 1: Retail Manager Profile Creation

To create a new profile:

1. Go to Setup >> type **Profiles** in Quick Find box >> click **Profiles** >> clone the desired profile (**Standard User**) >> enter profile name: **Retail Manager Profile** >> **Save**
2. While still on the profile page, click **Edit**
3. Select the **Custom App settings** as default for the **HandsMen Threads Portal**
4. Scroll down to **Custom Object Permissions** and give **Read/Create/Edit/Delete/View All/Modify All** access for the following objects:
    - **Product__c**
    - **Order__c**
    - **Order_Item__c**
    - **Customer_Review__c**
    - **Inventory__c**
    - **Shipping__c**
5. Give **Read/Create/Edit/View All** access for:
    - **Contact**
    - **Account**
6. Set **Session Timeout** to **2 hours** of inactivity
7. Configure **Password Policies** as:
    - Passwords expire in **90 days**
    - Minimum password length = **8**

    - Require at least **1 number and 1 uppercase letter**
8. Click **Save**

Activity 2: Sales Representative Profile Creation

1. Go to setup >> type **profiles** in quick find box >> click on **profiles** >> clone the desired profile (**Standard User Profile**) >> enter profile name (**Sales Representative Profile**) >> **Save**
2. While still on the profile page, click **Edit**
3. Select the **Custom App settings** as default for the **HandsMen Threads Portal**
4. Scroll down to **Custom Object Permissions** and give **Read/Create/Edit/View All** access for:
    - **Product__c** (Read Only)
    - **Order__c**
    - **Order_Item__c**
    - **Customer_Review__c** (Read Only)
    - **Inventory__c** (Read Only)
    - **Shipping__c** (Read Only)
5. Give **Read/Create/Edit** access for:
    - **Contact**
    - **Account**
6. Click **Save**

Activity 3: Inventory Manager Profile Creation

1. Go to setup >> type **profiles** in quick find box >> click on **profiles** >> clone the desired profile (**Standard User Profile**) >> enter profile name (**Inventory Manager Profile**) >> **Save**
2. While still on the profile page, click **Edit**
3. Select the **Custom App settings** as default for the **HandsMen Threads Portal**
4. Scroll down to **Custom Object Permissions** and give **Read/Create/Edit/Delete/View All** access for:
    - **Product__c** (Edit availability only)
    - **Inventory__c**
5. Give **Read Only** access for:
    - **Order__c**
    - **Order_Item__c**
    - **Shipping__c**
6. Click **Save**

Activity 4: Customer Profile Creation (For Experience Cloud)

1. Go to setup >> type **profiles** in quick find box >> click on **profiles** >> clone the desired profile (**Customer Community Plus User**) >> enter profile name (**Customer Profile**) >> **Save**
2. While still on the profile page, click **Edit**
3. Scroll down to **Custom Object Permissions** and give **Read Only** access for:
    - **Product__c**
4. Give **Read/Create** access for:
    - **Order__c** (own records only)
    - **Customer_Review__c** (own records only)
5. Give **Read Only** access for:
    - **Order_Item__c** (own records only)
    - **Shipping__c** (own records only)
    - **Inventory__c** (Product availability only)
6. Click **Save**

Milestone 17: Roles & Role Hierarchy

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Activity 1: Creation of Retail Manager Role

Creating Retail Manager Role:

1. Go to quick find >> Search for **Roles** >> click on **set up roles**
2. Click on **Expand All** and click on **add role** under **CEO**
3. Give Label as "**Retail Manager**" and the Role name gets auto populated
4. Then click on **Save**

Activity 2: Creating Sales Representative Role Under Retail Manager

1. Go to quick find >> Search for **Roles** >> click on **set up roles**
2. Click **plus** on **Retail Manager** role, and click **add role** under **Retail Manager**
3. Give Label as "**Sales Representative**" and Role name gets auto populated
4. Then click on **Save**

Activity 3: Creating Inventory Manager Role Under CEO

1. Go to quick find >> Search for **Roles** >> click on **set up roles**
2. Click on **Expand All** and click on **add role** under **CEO**
3. Give Label as "**Inventory Manager**" and Role name gets auto populated
4. Then click on **Save**

Activity 4: Creating Customer Role (If Applicable)

1. Go to quick find >> Search for **Roles** >> click on **set up roles**
2. Create a separate branch or leave customers without explicit roles (as Community users typically don't require role hierarchy for external access)
3. If creating a role, Give Label as "**Customer**" and Role name gets auto populated
4. Then click on **Save**

Phase 5: Deployment, Documentation & Maintenance

Deployment

In the scope of the HandsMen Threads E-Commerce Platform project implementation:

- The complete platform, including custom objects, page layouts, Flows, Apex classes, and Lightning Application, has been built and configured within a **Salesforce Developer Org**
- The primary goal of this phase is to understand how a men's fashion e-commerce platform can be developed, tested, and prepared for real-world deployment in an organization's Salesforce environment
- This includes configuring navigation items, access permissions, Lightning App, and deploying reusable pages for modules like:
    - **Product Catalog**
    - **Order Management**
    - **Inventory Tracking**
    - **Customer Reviews**
    - **Shipping Management**
    - **Customer Portal**
- Each page has been designed for easy usability, sophisticated aesthetics, and clear navigation to improve customer shopping experience and administrative efficiency

Note on Production Deployment:

- Actual deployment (migration from development to production) is not performed in this project because:
    - Developer Edition orgs are standalone and not connected to a production instance
    - Production deployment typically requires sandbox environments, change sets, or Salesforce DevOps tools (such as GitHub, Azure DevOps, or CI/CD pipelines) — which are part of enterprise-level projects
- However, all objects, page layouts, Flows, Apex classes, and the Lightning App have been structured in a way that they can easily be packaged and deployed to production in the future using standard Salesforce deployment tools

---

Maintenance, Monitoring & Troubleshooting

Maintenance ensures that product catalogs, order processing, inventory levels, customer reviews, and shipping records remain accurate, secure, and up to date.

Ongoing monitoring and maintenance include:

- Reviewing portal pages and Lightning App performance for responsiveness, load times, and mobile accessibility
- Ensuring security settings like profiles, permission sets, and sharing rules are updated according to employee role changes and customer access requirements
- Monitoring Flows for order processing, inventory updates, and email notifications for any failures or delays
- Maintaining duplicate rules and matching rules to prevent duplicate product entries or customer records
- Tracking inventory levels and ensuring automatic alerts for low stock items are functioning correctly
- Updating product catalogs, pricing, and promotional offers as per business requirements
- Monitoring customer review submissions and responding to low ratings promptly
- Tracking shipping status updates and ensuring delivery notifications are sent accurately
- Tracking user feedback and implementing UI or automation improvements for better customer and employee experience
- Reviewing sales dashboards and analytics to identify trends, top-selling products, and revenue patterns
- Regular database cleanup to archive old orders and maintain system performance
- Security audits to ensure customer data protection and compliance with privacy regulations

Although this project was implemented in a Salesforce Developer Org for demonstration and learning purposes, in a real-world enterprise deployment, these maintenance and monitoring activities are crucial to ensure the HandsMen Threads Platform remains stable, scalable, competitive, and aligned with fashion retail best practices and customer expectations.

Project Documentation

Project documentation in the HandsMen Threads E-Commerce Platform serves as a comprehensive record of the platform's purpose, design, development, and deployment.

It ensures that business requirements — such as managing product catalogs, processing customer orders, tracking inventory, collecting reviews, and managing shipping — are clearly defined and mapped to system functionality.

Documentation benefits:

- Acts as a **blueprint for developers and admins**, guiding consistent development and enabling future scalability and feature enhancements

- Supports **user training, troubleshooting, and maintenance** by detailing every object, automation, workflow, and configuration involved
- Assists in **audit readiness, change management, and knowledge transfer**, making it an essential asset for successful long-term sustainability and business growth
- Provides reference for **marketing teams** to understand product management features
- Helps **customer service teams** understand order processing and customer management workflows

---

Guidelines for HandsMen Threads E-Commerce Platform Documentation Submission

General Instructions:

- Submit in professional format (Word or PDF)
- Use clear headings, subheadings, and bullet points
- Maintain a consistent font (Times New Roman or Arial, size 12 or 13)
- Ensure zero grammatical or spelling mistakes and properly aligned sections
- Plagiarism is strictly prohibited
- Include screenshots of key features, dashboards, and user interfaces where applicable

---

Mandatory Sections to Include

Project Overview

The HandsMen Threads E-Commerce Platform is a centralized Salesforce application designed to manage sophisticated men's fashion retail operations. The platform handles product catalog management, customer order processing, inventory tracking, customer reviews, and shipping logistics.

Customers can:

- Browse curated fashion collections by category
- View detailed product information with images and specifications
- Add products to shopping cart and place orders
- Track order status and shipment delivery
- Submit product reviews and ratings
- Access order history and manage account preferences

Business Users can:

- Manage product catalogs, pricing, and availability
- Process customer orders and handle bulk orders
- Monitor inventory levels and receive low-stock alerts
- Review customer feedback and ratings
- Track shipping and delivery status
- Analyze sales performance and customer trends

The platform improves retail efficiency, customer satisfaction, and reduces dependency on manual order processing and inventory management.

Objectives

- **Empower customers** with seamless self-service capabilities for browsing, shopping, order tracking, and review submission
- **Reduce administrative overhead** through automated order processing, inventory updates, and shipping notifications
- **Ensure secure, role-based access** for customers, sales teams, inventory managers, and administrators
- **Improve customer engagement and loyalty** by providing an intuitive shopping experience and personalized recommendations
- **Enable data-driven business decisions** through real-time sales analytics, inventory insights, and customer feedback analysis
- **Maintain competitive advantage** in the men's fashion retail market through efficient operations and superior customer experience

---

Phase 1: Requirement Analysis & Planning

Understanding Business Requirements:

Identify the need for a centralized, sophisticated e-commerce platform for managing men's fashion products, customer orders, inventory, customer relationships, and shipping operations.

Defining Project Scope and Objectives:

Include modules:

- Product Catalog Management
- Order Processing & Management
- Inventory Tracking & Alerts
- Customer Review System
- Shipping & Delivery Management
- Customer Portal & Self-Service
- Sales Analytics & Reporting

Design Data Model and Security Model:

Create necessary custom objects (Product, Order, Order Item, Customer Review, Inventory, Shipping), define fields, establish relationships, and design record-level access based on user roles (customers, sales staff, managers, administrators).

Phase 2: Salesforce Development – Backend & Configurations

Setup Environment & DevOps Workflow:

Configure Developer Org and enable Lightning App settings for retail operations.

Customization of Objects & Fields:

Create objects for Product, Order, Order Item, Customer Review, Inventory, and Shipping.

Configure validation rules to:

- Prevent invalid pricing (negative or zero prices)
- Ensure positive order totals
- Validate inventory stock levels
- Prevent duplicate product codes

Automation:

Develop Flows for:

- Automated order confirmation emails
- Inventory updates when orders are placed
- Low stock alerts to inventory managers
- Shipping status notifications to customers
- Low rating alerts to product managers

Apex Classes / Triggers:

Backend logic for:

- Product catalog retrieval and filtering
- Order creation and processing
- Inventory calculations and updates
- Customer review management
- Order total calculations with discounts

Asynchronous Apex:

Queueable Apex for sending low stock alerts to inventory managers in batch.

Phase 3: UI/UX Development & Customization

Lightning App Setup:

Built **HandsMen Threads Portal** Lightning App for centralized navigation with sophisticated, fashion-focused branding.

Page Layouts:

Customized layouts for all objects with intuitive field organization:

- Product layouts showcasing specifications, pricing, and availability
- Order layouts displaying customer details, items, and totals
- Inventory layouts for stock management
- Review layouts for customer feedback

Navigation Items:

Added pages for:

- Product Catalog (with category filtering)
- Order Management
- Inventory Dashboard
- Customer Reviews
- Shipping Tracker
- Sales Analytics

User Management:

Created profiles, roles, and permission sets to control access for:

- Retail Managers (full access)
- Sales Representatives (customer and order management)
- Inventory Managers (product availability and stock control)
- Customers (self-service shopping and order tracking)

Phase 4: Data Migration, Testing & Security

Data Migration:

Used **Data Import Wizard** and **Data Loader** to load:

- Product catalog with images and specifications
- Customer records (existing customers)
- Historical order data
- Inventory levels
- Existing customer reviews

Field History Tracking:

Enabled for key fields to track:

- Product price changes
- Order status updates
- Inventory level changes

Duplicate & Matching Rules:

Configured to prevent:

- Duplicate product codes
- Duplicate customer email addresses
- Overlapping inventory records

Profiles, Roles, and Permission Sets:

Defined access levels for:

- **Retail Managers**: Full access to all retail operations
- **Sales Representatives**: Customer and order management
- **Inventory Managers**: Product availability and stock management
- **Customers**: Personal orders, reviews, and shipping info only

Testing:

Conducted:

- **Unit Testing** for Apex classes, triggers, and Flows
- **Integration Testing** for order processing workflow, inventory updates, and email notifications
- **UAT (User Acceptance Testing)** with sample customers, sales reps, and managers
- **Security testing** for profile-based access, field-level security, and customer data protection
- **Performance testing** for page load times and concurrent user handling

---

Phase 5: Deployment, Documentation & Maintenance

Deployment:

The platform was built in a **Salesforce Developer Org** to simulate real-world e-commerce deployment.

Actual production deployment was not performed due to Developer Org limitations, but the architecture supports migration via:

- **Change Sets**
- **Salesforce DevOps tools** (GitHub, Azure DevOps)
- **Unmanaged or Managed Packages**

Maintenance, Monitoring & Troubleshooting:

Ongoing activities include:

- Regular review of page layouts, Flows, and app performance

- Monitor order processing, inventory updates, and shipping notifications for accuracy
- Maintain security via profiles, permission sets, and sharing rules
- Update product catalogs, pricing, and seasonal collections
- Track sales analytics and customer trends
- Resolve errors via debug logs and update Flows/validations for new business rules
- Respond to customer reviews and feedback
- Conduct periodic security audits for data protection compliance

Project Documentation:

Documentation ensures all objects, workflows, profiles, roles, validations, and business logic are recorded comprehensively. It guides:

- Future maintenance and feature enhancements
- User training for employees and customers
- Audit compliance and data governance
- Change management for business process updates
- Knowledge transfer for new team members

Conclusion

The **HandsMen Threads E-Commerce Platform** project successfully demonstrates how Salesforce can be leveraged to create a sophisticated, customer-centric retail experience while streamlining internal operations for a men's fashion business.

Through a combination of custom objects, Apex classes, declarative automation (Flows), intuitive page layouts, and secure user configurations, the system enables seamless management of:

- Product catalogs with detailed specifications
- Customer order processing with automated workflows
- Real-time inventory tracking with low-stock alerts
- Customer review collection and display
- Shipping and delivery management
- Sales analytics and performance dashboards

The platform empowers customers with self-service capabilities for browsing, shopping, order tracking, and review submission, while providing business users with powerful tools for managing products, processing orders, monitoring inventory, and analyzing sales trends.

Although developed within a **Salesforce Developer Org** for training and demonstration purposes, the project reflects **real-world e-commerce and retail automation practices**. It highlights the importance of:

- User experience design for customer satisfaction
- Security setup for data protection and privacy compliance
- Process automation for operational efficiency

- Analytics for data-driven business decisions
- Ongoing maintenance for system reliability and competitiveness

The HandsMen Threads platform is architected for scalability and can be easily enhanced with additional features such as:

- AI-powered product recommendations
- Advanced search and filtering
- Loyalty programs and rewards
- Multi-channel integration (mobile app, social commerce)
- Advanced analytics and predictive insights
- Integration with payment gateways and logistics providers

This project demonstrates end-to-end Salesforce development capabilities and provides a solid foundation for building sophisticated e-commerce solutions that elevate fashion retail businesses to new levels of sophistication and customer satisfaction.

Steps for Demo Video Presentation

1. Introduction

- Introduce yourself and your project name: **"HandsMen Threads E-Commerce Platform using Salesforce."**
- Briefly explain the project's purpose: **"A sophisticated men's fashion retail platform that manages product catalogs, customer orders, inventory tracking, customer reviews, and shipping logistics."**

2. App Overview

- Show the **Lightning App** from the App Launcher
- Mention custom objects like **Product__c**, **Order__c**, **Order_Item__c**, **Customer_Review__c**, **Inventory__c**, and **Shipping__c**
- Show the app navigation and tab structure

3. User Interface Demonstration

Customer Experience:

- Show how customers browse the product catalog
- Demonstrate filtering products by category (Formal Wear, Casual Wear, Accessories)
- Show product detail page with specifications, pricing, and images
- Demonstrate adding products to cart and placing an order
- Show order confirmation and tracking
- Demonstrate submitting a product review

Sales Representative Experience:

- Show how sales reps create orders on behalf of customers

- Demonstrate order management dashboard
- Show customer interaction tracking

Inventory Manager Experience:

- Show inventory dashboard with stock levels
- Demonstrate low stock alerts
- Show restocking process

Manager Experience:

- Show approval process for discount requests
- Demonstrate bulk order approval workflow
- Show sales analytics dashboards

4. Business Process Automation

- Show the **Flows** created for:
    - Order confirmation emails
    - Inventory updates when orders are placed
    - Low stock alerts to inventory managers
    - Shipping notifications to customers
    - Low rating alerts for product reviews
- Explain email automation and notification system
- Open related backend logic (**Apex Classes**) to show:
    - Product catalog management
    - Order processing logic
    - Inventory calculation automation
    - Review management

5. User Management & Security

- Demonstrate **Profile** and **Permission Set** configuration for different user types (Retail Manager, Sales Representative, Inventory Manager, Customer)
- Show **role hierarchy** ensuring proper data visibility
- Demonstrate **sharing rules** and **field-level security** ensuring customer data privacy and business information protection
- Show how customers can only access their own orders while managers have broader visibility

6. Error Handling & Debugging

- Briefly show validation rules preventing:
    - Negative pricing
    - Invalid order totals
    - Negative inventory
    - Duplicate product codes
- Show debug logs or Flow error handling setup

7. Highlights

- Showcase the **Lightning App** design with sophisticated, fashion-focused branding
- Highlight page layouts optimized for product display and order management
- Mention unique features like:
    - Real-time inventory tracking
    - Automated customer notifications
    - Customer review system
    - Multi-level approval processes
    - Role-based access for security
    - Sales analytics dashboards

8. Conclusion

- Summarize the complete platform workflow and its business value:
    - **Enhanced customer experience** through intuitive self-service
    - **Operational efficiency** through automation
    - **Data-driven insights** through analytics
    - **Competitive advantage** in men's fashion retail
- Mention that it was thoroughly tested, documented, and is ready for production adaptation
- Highlight scalability for future enhancements like mobile app integration, AI recommendations, and loyalty programs

---

Final Conclusion

The **HandsMen Threads E-Commerce Platform** successfully illustrates how Salesforce can streamline sophisticated fashion retail operations while delivering an exceptional customer experience.

With comprehensive features including:

- **Product catalog management** with detailed specifications
- **Seamless order processing** with automated workflows
- **Real-time inventory tracking** with proactive alerts
- **Customer review system** for feedback and engagement
- **Shipping management** with delivery tracking
- **Sales analytics** for business intelligence
- **Multi-level security** for data protection
- **Role-based access control** for operational efficiency

The platform demonstrates how modern retail businesses can leverage Salesforce to:

- Enhance customer satisfaction and loyalty
- Improve operational efficiency through automation
- Make data-driven business decisions
- Maintain competitive advantage in the fashion industry
- Scale operations seamlessly as the business grows

Even though implemented in a **Developer Org** for educational and demonstration purposes, the project aligns with **real-world Salesforce e-commerce standards**, demonstrating comprehensive understanding of:

- Object-oriented data modeling
- Business process automation
- User experience design
- Security architecture
- Integration readiness
- Performance optimization
- Deployment best practices

The HandsMen Threads platform stands as a testament to the power of Salesforce in transforming traditional retail operations into sophisticated, data-driven, customer-centric digital experiences that elevate the art of men's fashion retail.