

**SmartBridge - Salesforce Virtual Internship Program 2025**

*(in collaboration with AICTE)*

## **Project Documentation**

### **HandsMen Threads: Elevating the Art of Sophistication in Men's Fashion**

**Intern Name:** Bhavnita Oinam

**University:** Amity University, Noida

**Course:** B.Tech – Computer Science and Engineering

**Project Domain:** Salesforce CRM Development

**Project Duration:** 21 May 2025 – 18th July 2025

**Internship Type:** Virtual

**Submitted To:**

Team Smartbridge

# INDEX

<b>I. Project Overview.....</b>	<b>3</b>
<b>II. Objectives.....</b>	<b>4</b>
<b>III. Phase 1: Requirement Analysis &amp; Planning.....</b>	<b>5</b>
Understanding Business Requirements.....	5
Defining Project Scope and Objectives.....	5
Design of Data Model and Security Model.....	6
<b>IV. Phase 2: Salesforce Development - Backend &amp; Configurations.....</b>	<b>7</b>
Setup Environment & DevOps Workflow.....	7
Automation: Workflow Rules, Process Builder, Flows, and Approval Processes.....	10
Order Confirmation Flow (Record-Triggered Flow).....	10
Loyalty Status Update Flow (Scheduled Flow).....	11
Low Stock Alert Flow (Record-Triggered Flow).....	11
Apex Classes, Triggers & Asynchronous Apex.....	12
InventoryBatchJob Apex Class.....	13
<b>V. Phase 3: UI/UX Development &amp; Customization.....</b>	<b>14</b>
1. Unit Testing of Individual Components.....	14
2. Performance & Load Testing.....	14
3. Security Testing.....	15
4. Bug Fixes and Improvements.....	15
<b>VI. Phase 4: Data Migration, Testing &amp; Security.....</b>	<b>17</b>
1. Deployment to Production.....	17
2. End-User Training.....	17
<b>VII. Phase 5: Deployment, Documentation &amp; Maintenance.....</b>	<b>20</b>
1. Deployment Strategy.....	20
2. Documentation.....	21
3. Maintenance & Monitoring.....	21
<b>VIII. Conclusion.....</b>	<b>23</b>
<b>IX. Future Enhancements.....</b>	<b>24</b>

## I. Project Overview

The Salesforce CRM implementation project was undertaken for *HandsMen Threads*, a premium men's fashion brand specializing in bespoke tailoring. The project's main goal was to use Salesforce's capabilities to digitally revolutionize the company's customer relationship management procedures. The CRM system was created to facilitate customized customer interactions, effective order monitoring, and proactive inventory management because of the brand's emphasis on personalization, high-quality service, and operational accuracy.

By centralizing client data, automating tedious operations, and improving departmental visibility, the system was designed to meet the expanding needs of the company. Automated order confirmations, dynamic loyalty status updates based on past purchases, inventory alert notifications when stock levels drop below a certain threshold, and daily scheduled batch processing for large orders are some of the key features that have been put into place. Through prompt communication and individualized services, these features enhanced client engagement and retention in addition to streamlining daily operations.

The project also placed a strong emphasis on user experience, system scalability, and data quality. To facilitate role-based access, avoid data duplication, and preserve the integrity of all company records, a robust data and security model was created. Additionally, the CRM had dashboards for real-time analytics, user-friendly interfaces, and customized objects that closely matched the fashion brand's particular business procedures. Order processing, tracking customer interactions, and business reporting are now all supported by this all-inclusive system.

This project essentially shows how Salesforce is used in practice as an enterprise-grade CRM technology that supports both customer management and end-to-end business operations. It gives HandsMen Threads the flexibility and automation required to expand sustainably in a cutthroat, fashion-forward market.

## II. Objectives

This Salesforce CRM project's main goal was to create and deploy a dependable, expandable, and intuitive customer relationship management system that was specifically suited to the requirements of HandsMen Threads, a high-end handmade menswear brand. By utilizing Salesforce's robust tools and automation capabilities, the project sought to automate and optimize the whole customer journey, from the time an order is placed to post-purchase engagement.

One of the main objectives was to reduce reliance on manual procedures and eliminate dispersed records by centralizing all inventory, order, and customer data into a single platform. By guaranteeing a single source of truth for the company, this would facilitate improved decision-making, precise reporting, and smooth cooperation between divisions such as operations, marketing, and sales.

Reducing the overhead of repeated work and guaranteeing prompt, reliable client communication were two further key goals of automation. Features like automated order confirmation emails, customer history-based loyalty program updates, and stock alerts for warehouse staff when inventory levels dropped below a certain threshold were all part of this. In order to increase internal efficiency and reduce human error, the system was intended to integrate logic using Flows, Apex Triggers, and Batch Apex.

With personalization, the CRM aims to improve client engagement. The approach may be able to facilitate more focused loyalty programs, better customer service encounters, and perhaps even tailored suggestions by monitoring consumer behavior and purchase trends. Increasing client happiness, encouraging brand loyalty, and encouraging repeat business are the three main ways that businesses can maximize their value.

On a technical level, the goal was to develop a system that was easy to maintain, with clear data models, safe access restrictions, and an intuitive user interface. The initiative functioned as a strategic digital transformation endeavor for the brand as well as a practical learning experience.

### **III. Phase 1: Requirement Analysis & Planning**

Understanding HandsMen Threads' business challenges and matching the CRM solution to their operational objectives and customer interaction strategies were the main aims of the first phase of the Salesforce CRM deployment. This stage ensured that all subsequent development would be scalable and purpose-driven by laying the groundwork for the system's architecture, data model, automation logic, and security design.

#### **Understanding Business Requirements**

HandsMen Threads provides individualized tailoring services and is a retailer of high-end fashion. The business required a solution that could automate important interactions, optimize internal operations, and handle consumer data. Among the main business pain points found were:

- Service delivery is fragmented due to a lack of centralized customer and order data.
- Order confirmations, inventory tracking, and loyalty status updates are all done manually.
- There is no automatic method for managing large orders or anticipating low stock conditions.
- Creating performance reports for sales, inventory, and customer interaction can be challenging.

The stakeholders, which included customer service representatives, warehouse managers, and sales teams, emphasized the necessity of better customer communication, less manual labor, and real-time data visibility.

#### **Defining Project Scope and Objectives**

The Salesforce CRM implementation's scope was delineated based on the requirements that were gathered. Among them were:

- To appropriately portray business activities, custom objects like orders, products, customers, loyalty points, and inventory are created.
- Automating processes for stock notifications, loyalty point updates, and order confirmations.
- Establishing batch procedures to manage financial tracking and bulk order modifications.
- Constructing management dashboards to track consumer behavior, inventory fluctuations, and sales.
- Creating security measures that guarantee role-based data access, safeguarding operational and sensitive customer data.
- Allowing for scalability in order to include future features and grow with the company.

Early definition of these components allowed the project to move forward with precise objectives, quantifiable results, and stakeholder expectations that were all in line.

## **Design of Data Model and Security Model**

To make sure that the links between different entities (such as customers, orders, and products) were clearly defined, a strong data model was necessary. The following design factors were taken into account:

- **Entity-Relationship Diagram:** The purpose of an entity-relationship diagram is to show and verify the structure of linked objects. For instance, a single customer may have several orders, each of which is connected to one or more products.
- **Relationships with Custom Fields:** Logical connections between records were made using lookup and master-detail relationships. Totals, order status, and loyalty point summaries were computed using formula fields.
- **Record kinds and Page Layouts:** Depending on the user's job (e.g., Admin, Sales Rep, Warehouse Team), several layouts and kinds were established.
- **Validation rules:** They are put in place to stop inaccurate or partial data entry; for example, they prohibit orders without specified products from being submitted.
- **Security Model:**
  - To guarantee that users could only access information pertinent to their job function, profiles and roles were assigned.
  - When more adaptable access control was required, Permission Sets were developed.
  - Record-level sharing between teams was made possible by Sharing Rules, which also protected sensitive data.

## IV. Phase 2: Salesforce Development - Backend & Configurations

Phase 2 concentrated on the technical construction of the Salesforce CRM system for HandsMen Threads following the completion of the planning and requirement analysis. During this stage, the development environment was built up, objects and fields were customized, automation tools were configured, and Apex code was written as needed. The objective was to convert the Phase 1 business logic into a functional, expandable Salesforce solution.

### Setup Environment & DevOps Workflow

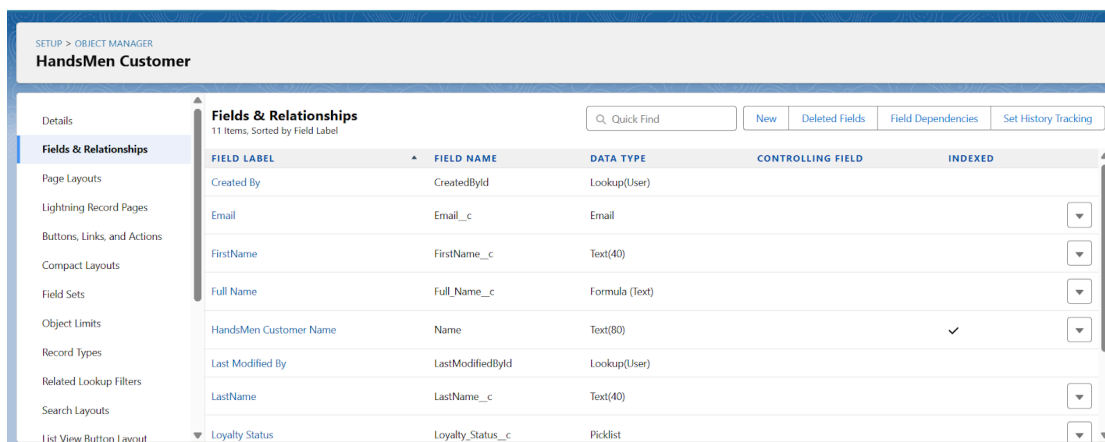
To build and test every component in a controlled context, the development process started with the setup of a Salesforce Developer Org. Important setup procedures included:

- Use Lightning App Builder to create a project-specific application by grouping similar tabs and components.
- Development logs were arranged and metadata components were backed up to emulate version control procedures.
- Validated components were to be migrated from the development environment to the production organization using Change Sets.
- Advanced feature testing included sandbox testing to simulate production conditions.

### Customization of Objects, Fields & Relationships

To represent the fundamental functions of HandsMen Threads, custom objects were made. These comprised:

- **Customer\_\_c**: To store client information such as name, contact, preferences, and loyalty tier.



FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
FirstName	FirstName__c	Text(40)		
Full Name	Full_Name__c	Formula (Text)		
HandsMen Customer Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
LastName	LastName__c	Text(40)		
Loyalty Status	Loyalty_Status__c	Picklist		

- **Order\_\_c:** Linked to Customer and Product objects to track individual purchases.

SETUP > OBJECT MANAGER  
**HandsMen Order**

Details

**Fields & Relationships**  
8 Items, Sorted by Field Label

Q, Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(HandsMen Customer)		✓
HandsMen OrderNumber	Name	Auto Number		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Quantity	Quantity__c	Number(8, 0)		
Status	Status__c	Picklist		
Total Amount	Total_Amount__c	Number(16, 2)		

Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout

- **Product\_\_c:** Maintained inventory details and product variants.

SETUP > OBJECT MANAGER  
**HandsMen Product**

Details

**Fields & Relationships**  
8 Items, Sorted by Field Label

Q, Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
HandsMen Product Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Order	Order__c	Lookup(HandsMen Order)		✓
Owner	OwnerId	Lookup(User,Group)		✓
Price	Price__c	Currency(16, 2)		
SKU	SKU__c	Text(40)		
Stock Quantity	Stock_Quantity__c	Number(8, 0)		

Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout

- **Inventory\_\_c:** Managed stock levels for each product.

SETUP > OBJECT MANAGER  
**Inventory**

Details

**Fields & Relationships**  
7 Items, Sorted by Field Label

Q, Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Inventory Number	Name	Auto Number		✓
Last Modified By	LastModifiedById	Lookup(User)		
Product	Product__c	Master-Detail(HandsMen Product)		✓
Stock Quantity	Stock_Quantity__c	Number(8, 0)		
Stock Status	Stock_Status__c	Formula (Text)		
Warehouse	Warehouse__c	Text(80)		

Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout



- **Marketing\_Campaign\_\_c:**

SETUP > OBJECT MANAGER

### Marketing Campaign

Details

**Fields & Relationships**  
7 Items, Sorted by Field Label

Q Quick Find New Deleted Fields Field Dependencies Set History Tracking

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

Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets  
Object Limits  
Record Types  
Related Lookup Filters  
Search Layouts  
List View Button Layout

- **Validation Rules:**

SETUP > OBJECT MANAGER

### HandsMen Customer

Details

**HandsMen Customer Validation Rule**  
[Back to HandsMen Customer](#)

Help for this Page

**Validation Rule Detail** Edit Clone

Rule Name	Email	Active	✓
Error Condition Formula	NOT CONTAINS( Email__c, "@gmail.com")		
Error Message	Please fill Correct Email	Error Location	Top of Page
Description			
Created By	Rhavyrita Oinam, 7/11/2025, 5:31 AM	Modified By	Rhavyrita Oinam, 7/11/2025, 5:32 AM

Edit Clone

Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets

SETUP > OBJECT MANAGER

### HandsMen Order

Details

**HandsMen Order Validation Rule**  
[Back to HandsMen Order](#)

Help for this Page

**Validation Rule Detail** Edit Clone

Rule Name	Total_Amount	Active	✓
Error Condition Formula	Total_Amount__c <= 0		
Error Message	Please Enter Correct Amount	Error Location	Total Amount
Description			
Created By	Rhavyrita Oinam, 7/11/2025, 5:26 AM	Modified By	Rhavyrita Oinam, 7/11/2025, 5:26 AM

Edit Clone

Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets

SETUP > OBJECT MANAGER

### Inventory

Details

**Inventory Validation Rule**  
[Back to Inventory](#)

Help for this Page

**Validation Rule Detail** Edit Clone

Rule Name	Stock_Quantity	Active	✓
Error Condition Formula	Stock_Quantity__c <= 0		
Error Message	The inventory count is never less than zero.	Error Location	Top of Page
Description			
Created By	Rhavyrita Oinam, 7/11/2025, 5:34 AM	Modified By	Rhavyrita Oinam, 7/11/2025, 5:34 AM

Edit Clone

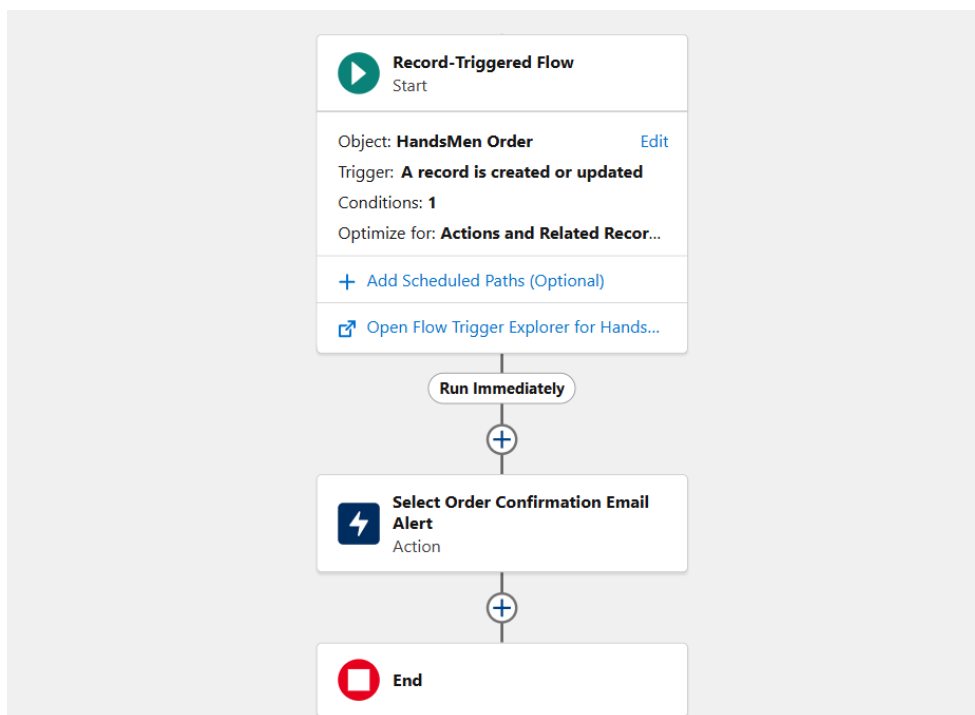
Page Layouts  
Lightning Record Pages  
Buttons, Links, and Actions  
Compact Layouts  
Field Sets

## Automation: Workflow Rules, Process Builder, Flows, and Approval Processes

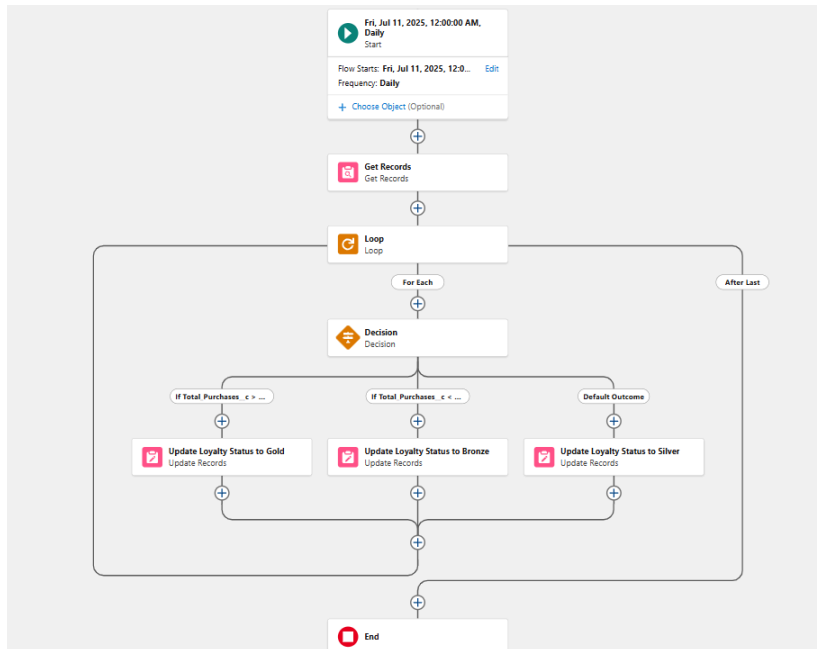
Automation was a key component in aligning CRM functionality with business goals:

- **Record-Triggered Flows** (used instead of Process Builder due to best practice recommendations) to:
  - Send automatic email confirmations when an Order is created.
  - Update LoyaltyPoints\_\_c when an Order is marked “Completed.”
  - Trigger Inventory Alerts when Product stock < 5.
- **Scheduled Flows** to process daily batch updates for bulk orders and adjust inventory levels at midnight.
- **Approval Process** (if applicable) to simulate real-world review workflows such as approving large bulk orders or loyalty redemptions. Though not mandatory, it can be introduced for bonus features.

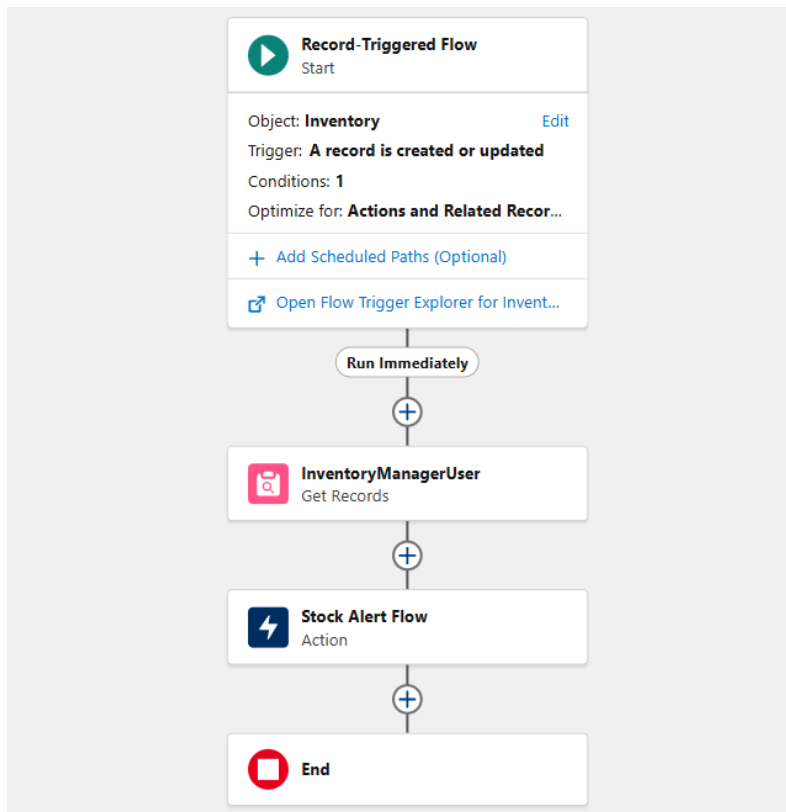
### Order Confirmation Flow (Record-Triggered Flow)



## Loyalty Status Update Flow (Scheduled Flow)



## Low Stock Alert Flow (Record-Triggered Flow)

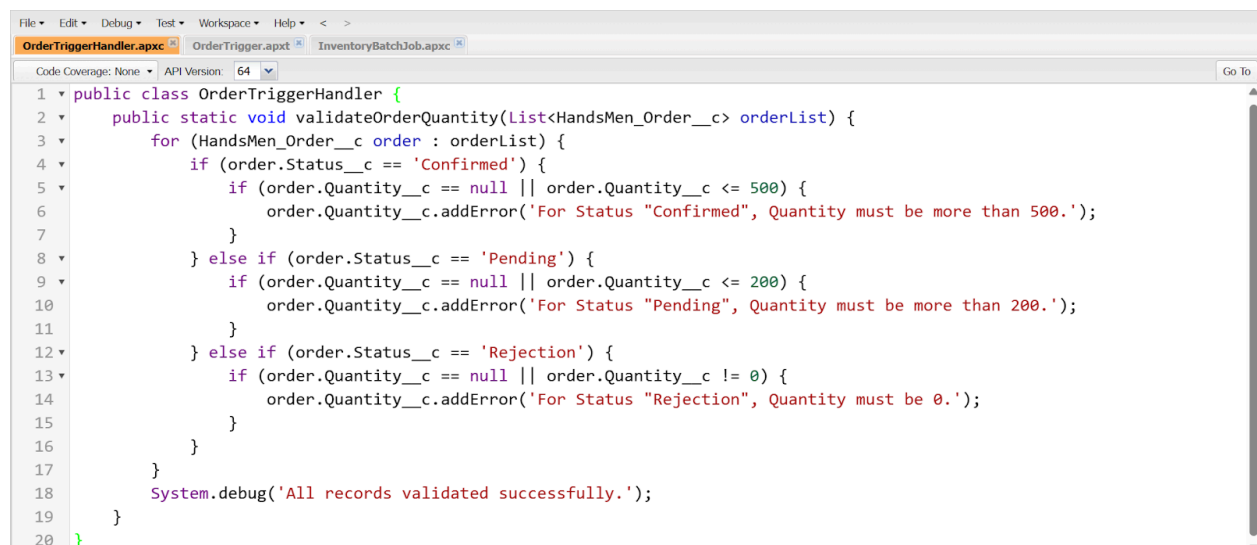


## Apex Classes, Triggers & Asynchronous Apex

While most logic was handled through declarative tools, code-based automation was introduced for more complex operations:

- **Apex Triggers:**
  - Trigger on **Order\_\_c** to calculate and update **LoyaltyPoints\_\_c**.
  - Trigger on **Inventory\_\_c** to flag low stock and notify warehouse via email.
- **Apex Classes:**
  - Utility classes to encapsulate common logic like point calculation or status checks.
- **Batch Apex (Asynchronous):**
  - A scheduled **InventoryBatchJob** class was created to run at midnight every day.
  - It iterates over **Order\_\_c** records with a “Pending” status, marks them as “Processed,” updates totals, and triggers inventory adjustment.
  - This helped simulate real-world scheduled tasks with large data volumes.

### OrderTriggerHandler Apex Class



```
File Edit Debug Test Workspace Help < >
OrderTriggerHandler.apxc OrderTrigger.apxt InventoryBatchJob.apxc
Code Coverage: None API Version: 64 Go To

1 public class OrderTriggerHandler {
2     public static void validateOrderQuantity(List<HandsMen_Order__c> orderList) {
3         for (HandsMen_Order__c order : orderList) {
4             if (order.Status__c == 'Confirmed') {
5                 if (order.Quantity__c == null || order.Quantity__c <= 500) {
6                     order.Quantity__c.addError('For Status "Confirmed", Quantity must be more than 500.');
```

## OrderTrigger Apex Trigger

```
File Edit Debug Test Workspace Help < >
OrderTriggerHandler.apxc OrderTrigger.apxt InventoryBatchJob.apxc
Code Coverage: None API Version: 64 Go To

1 trigger OrderTrigger on HandsMen_Order__c (before insert, before update) {
2     if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
3         OrderTriggerHandler.validateOrderQuantity(Trigger.new);
4     }
5 }
```

## InventoryBatchJob Apex Class

```
File Edit Debug Test Workspace Help < >
OrderTriggerHandler.apxc OrderTrigger.apxt InventoryBatchJob.apxc
Code Coverage: None API Version: 64 Go To

1 global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {
2     global Database.QueryLocator start(Database.BatchableContext BC) {
3         return Database.getQueryLocator(
4             'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'
5         );
6     }
7     global void execute(Database.BatchableContext BC, List<SObject> records) {
8         List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();
9         // Cast SObject list to Product__c list
10        for (SObject record : records) {
11            HandsMen_Product__c product = (HandsMen_Product__c) record;
12            product.Stock_Quantity__c += 50; // Restock logic
13            productsToUpdate.add(product);
14        }
15        if (!productsToUpdate.isEmpty()) {
16            try {
17                update productsToUpdate;
18            } catch (DmlException e) {
19                System.debug('Error updating inventory: ' + e.getMessage());
20            }
21        }
22    }
23    global void finish(Database.BatchableContext BC) {
24        System.debug('Inventory Sync Completed');
25    }
26    // Scheduler Method
27    global void execute(SchedulableContext SC) {
28        InventoryBatchJob batchJob = new InventoryBatchJob();
29        Database.executeBatch(batchJob, 200);
30    }
31 }
```

## **V. Phase 3: UI/UX Development & Customization**

To guarantee a dependable, safe, and error-free implementation, it was crucial to rigorously test each component and automation logic after Phase 2 of the Salesforce system's basic construction was finished. Phase 3 was devoted to thorough system validation, modeling real-world use cases, and optimizing data security and performance.

### **1. Unit Testing of Individual Components**

To confirm intended behavior under both normal and edge conditions, each custom feature—objects, flows, Apex triggers, and scheduled jobs—was evaluated separately.

- Custom Fields & Objects:
  - Verified the accuracy of the formula and field-level validation standards.
  - Confirmed that when records were added or removed, lookup and master-detail connections worked as intended.
- Automation and Logic Testing:
  - Verified that, in response to record modifications, Record-Triggered Flows operated as intended.
  - Batch jobs were tested to ensure correct operation during planned runs.
  - Reviewed Apex Triggers' inventory notifications and loyalty point computations.
- Email Notifications:
  - To ensure proper formatting and delivery of automatically generated confirmation emails, an order creation simulation was conducted.
  - Confirmed that stock notifications are sent to the warehouse personnel at the appropriate threshold.

### **2. Performance & Load Testing**

Despite being a mock internship, consideration was given to how the system would function under actual business demands:

- Performance of Batch Apex:
  - Made certain that huge datasets (more than 100 records) were executed correctly.
  - Verified application of best practices and governor restrictions to prevent failures or timeouts.
- Flow Optimizations:
  - Effective use of decision elements.
  - Recursive flow calls were avoided.
  - Ensured that Record-Triggered Flows had the right entry conditions.

### 3. Security Testing

- Profile and Permission Set Configuration:
  - Configuring profiles and permission sets made that users could access only pertinent fields, records, and tabs.
  - Sensitive fields such as internal stock alerts or financial data are subject to read/write restrictions.
- Field-Level Security Testing:
  - Used report outputs or URL manipulation to confirm that concealed fields were unavailable.
- Validation of Sharing Rules:
- Made sure that various jobs, such as administrative, stylistic, and warehousing employees, have the right visibility.

### 4. Bug Fixes and Improvements

Minor errors and logical inconsistencies were found and fixed during testing:

- To avoid double execution, flow entrance criteria were modified.
- improved email formatting and template variable bindings.
- Fixed To prevent infinite recursion when updating loyalty points, use apex trigger logic.

### App Details and Branding

The screenshot shows the 'App Details & Branding' configuration page in the Lightning App Builder. The page is divided into two main sections: 'App Details' and 'App Branding'.

**App Details:**

- \* App Name:** HandsMen Threads
- \* Developer Name:** HandsMen\_Threads
- Description:** Designed and configured to serve as a centralized operational dashboard for the

**App Branding:**

- Image:** A blue shirt with a yellow tie.
- Primary Color Hex Value:** #025197
- Org Theme Options:** ☒ Use the app's image and color instead of the org's custom theme

**App Launcher Preview:**

The preview shows the app icon (blue shirt with yellow tie) and the app name 'HandsMen Threads' with the description 'Designed and configured to serve as a centralized oper...'

# Dashboard

← Lightning App Builder

Pages ▾

HandsMen Threads Operational Dashboard

?

Help

🔄 ↻ 🗂 📄 📁

Desktop ▾

Shrink To View ▾

🔍

Activation... Save

Components

Search...

⚙

Standard (25)

Accordion

Chatter Feed

Chatter Publisher

CRM Analytics Collection

CRM Analytics Dashboard

Dashboard

Einstein Next Best Action

Flow

Launchpad

List View

Lobby Management

LWC CRM Analytics Dashboard

Get more on the AppExchange

HandsMen Threads Dashboard

Welcome to the centralized operations panel. Monitor orders, stock, and customer insights in one place.

Low Stock Products

Sum of Stock Quantity

HandsMen Product Hands...

Alerts

3

View Report

As of Today at 10:20 AM

Monthly Sales Summary

Recent Chart

7/14/2025

HandsMen Order Created Date

View Report

As of Today at 10:20 AM

Loyalty Status Breakdown

Sum of Total Purchases

Loyalty Status

1.2k

15k

View Report

As of Today at 10:20 AM

All

3 Items, sorted by Inventory Number

1-0001

1-0004

1-0006

Standard Recent Items (1)

Q-0003

Q-0002

Q-0001

All

3+ Items, sorted by HandsMen Customer Name

Aanya Verma

Pooni Verma

Saee Singh

Page

\* Label

HandsMen Threads Operational Dashboard

\* API Name

HandsMen\_Threads\_Operational\_Dashboard

\* Page Type

App Page

Template

Header and Three Regions

Change

Description

Actions

Select...



## VI. Phase 4: Data Migration, Testing & Security

The successful deployment of the solution into a simulated production environment and providing end users with the necessary skills to operate the system efficiently were the main objectives of the last stage of the Salesforce virtual internship project. This stage made sure that the data model, automation, logic, and communication flows—all of which had been developed in earlier stages—were smoothly and minimally disruptedly moved into a stable, functional environment. In order to promote acceptance and guarantee long-term success, user onboarding and support systems were also put in place.

### 1. Deployment to Production

- **Metadata Migration:**
  - Change Sets were used to bundle and transfer all custom objects, fields, validation rules, flows, Apex code, and email templates from the development environment (sandbox) to production.
  - To make sure nothing broke during transfer, dependencies and object relationships were checked before deployment.
- **Post-Deployment Validation:**
  - To ensure that every component was correctly delivered, a sanity check was carried out in the production environment following the migration.
  - To ensure that flows, triggers, and email notifications were operating properly, sample records were made.
- **Deployment Checklist Completed:**
  - To guarantee functionality and completeness in the new environment, every item (object structure, automation, templates, access permissions, etc.) was examined.

### 2. End-User Training

Customized training was necessary because HandsMen Threads is a company with many jobs, including as sales executives, warehouse managers, stylists, and administrators:

- **Conduct of Training Sessions:**
  - To demonstrate how each user job may use the new Salesforce interface to carry out their everyday responsibilities, simulated walkthroughs and demo sessions were developed.
- **User manuals based on roles:**
  - Sales Team: Helped with order placement, loyalty tier viewing, and customer record creation.
  - The warehouse staff: They received training on keeping an eye on stock alerts and restocking messages.

- Admins: Taught how to manage support tickets, make updates, and keep an eye on batch jobs.
- Hands-on Practice:
  - To get acquainted with the system's functionality and navigation, test users were urged to experiment with it and mimic actual tasks.

## Field History Tracking

The screenshot shows the Salesforce Setup page for Field History Tracking. The left sidebar contains a search bar with 'field' and a list of navigation items: Field Service Setup Home (Beta), Feature Settings, Field History Tracking, Marketing, LinkedIn Lead Gen, Lead Gen Fields, Service, Field Service, Field Service Mobile, Field Service Mobile App Builder, Field Service Settings, Objects and Fields, Object Manager, Picklist Value Sets, and Schema Builder. The main content area is titled 'Field History Tracking' and includes a search bar, a checkbox for 'Show only tracked objects', and a table listing objects and their tracked fields.

Object	Number of Tracked Fields	
Case	10	<a href="#">View</a>
Contract	18	<a href="#">View</a>
Order	17	<a href="#">View</a>

Page 1/1 | < Previous | Next >

## Duplicate Rules

The screenshot shows the Salesforce Setup page for Duplicate Rules. The left sidebar contains a search bar with 'duplicate' and a list of navigation items: Data, Duplicate Management, Duplicate Error Logs, Duplicate Rules, and Matching Rules. The main content area is titled 'Duplicate Rules' and includes a section for 'All Duplicate Rules' with a 'What Are Duplicate Rules?' link. Below this is a table listing duplicate rules.

Rule Name	Description	Object	Matching Rule	Active	Last Modified By	Last Modified Date
Standard Account Duplicate Rule	Identify accounts that duplicate other accounts.	Account	Standard Account Matching Rule	✓	OEPIC	7/3/2025
Standard Contact Duplicate Rule	Identify contacts that duplicate other contacts and leads.	Contact	Standard Lead Matching Rule Standard Contact Matching Rule	✓	OEPIC	7/3/2025
Standard Lead Duplicate Rule	Identify leads that duplicate other leads and contacts.	Lead	Standard Lead Matching Rule Standard Contact Matching Rule	✓	OEPIC	7/3/2025

## Matching Rules

**Setup**
Home   Object Manager ▾

---

▾ Data
 

▾ Duplicate Management
 

Duplicate Error Logs
 Duplicate Rules
 **Matching Rules**

Didn't find what you're looking for?  
Try using Global Search.

**SETUP**  
**Matching Rules**

[Help for this Page](#)

## All Matching Rules

**What Are Matching Rules?**
[\[ Expand \]](#)

View: All Matching Rules ▾ [Create New View](#)

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

Action	Rule Name +	Object	Status	Description	Last Modified Date	Last Modified By
<a href="#">Deactivate</a>	Standard Account Matching Rule	Account	Active	Matching rule for account records. <a href="#">More info</a>	7/3/2025	OEPIC
<a href="#">Deactivate</a>	Standard Contact Matching Rule	Contact	Active	Matching rule for contact records. <a href="#">More info</a>	7/3/2025	OEPIC
<a href="#">Deactivate</a>	Standard Lead Matching Rule	Lead	Active	Matching rule for lead records. <a href="#">More info</a>	7/3/2025	OEPIC

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

## Roles

Setup

Home

Object Manager ▾

Q roles

Users

Roles

Feature Settings

Sales

- Contact Roles on Contracts
- Contact Roles on Opportunities

Service

- Case Teams
  - Case Team Roles
- Contact Roles on Cases

Didn't find what you're looking for?  
Try using Global Search.

SETUP  
**Roles**

Collapse All Expand All

- Amity University Noida
  - Add Role
  - CEO Edit | Del | Assign
    - Add Role
  - CFO Edit | Del | Assign
    - Add Role
  - COO Edit | Del | Assign
    - Add Role
  - Inventory Edit | Del | Assign
    - Add Role
  - Marketing Edit | Del | Assign
    - Add Role
  - Sales Edit | Del | Assign
    - Add Role
  - SVP\_Customer Service & Support Edit | Del | Assign
    - Add Role
  - Customer Support, International Edit | Del | Assign
    - Add Role
  - Customer Support, North America Edit | Del | Assign
    - Add Role
  - Installation & Repair Services Edit | Del | Assign

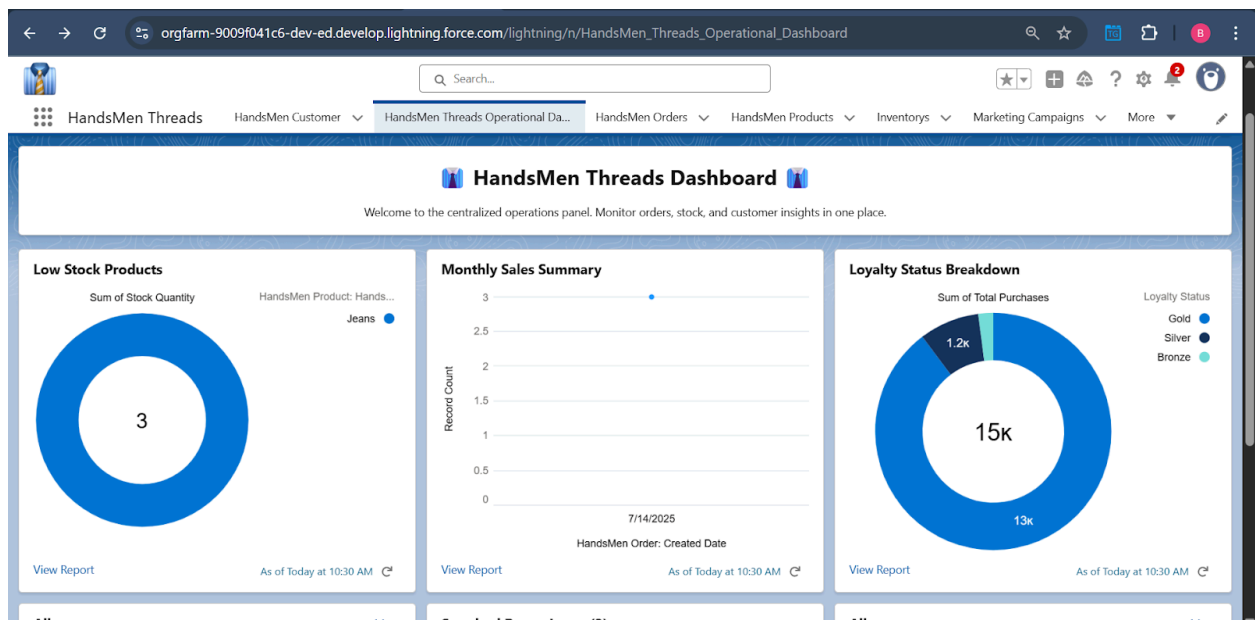
## VII. Phase 5: Deployment, Documentation & Maintenance

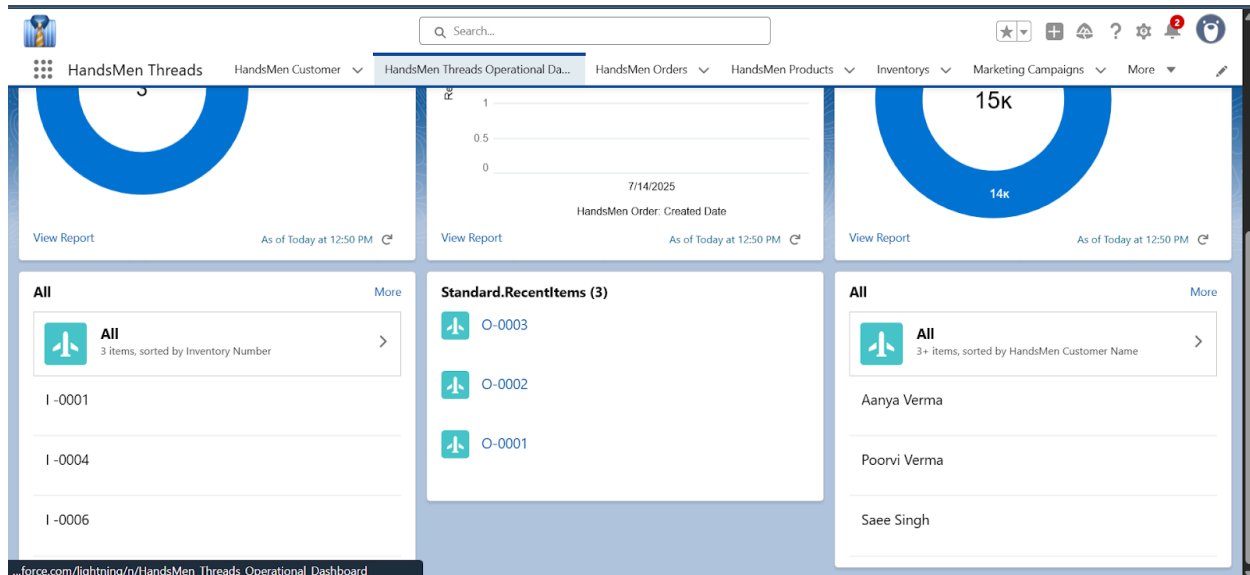
Assuring a seamless transition from development to production, preserving system stability, and providing stakeholders with the information and resources required for long-term sustainability were the key objectives of the project's last phase. By recording every step of the build and creating a plan for post-deployment monitoring, support, and improvement, this phase strengthened the CRM installation.

### 1. Deployment Strategy

- Change Set Deployment:
  - Outbound Change Sets were used to transfer all customizations from the sandbox to the production environment, including objects, fields, flows, Apex classes, validation rules, and dashboards.
  - To prevent component failure or incomplete transfers, every deployment package was tested and its dependencies checked before being pushed.
- Pre-Deployment Checklist:
  - To confirm all configurations, test coverage for Apex classes, and user role access settings, a thorough checklist was kept up to date.
- Post-Deployment Testing:
  - To make sure that the automation processes (Flows, Approval Processes, and Workflow Rules) were operating properly, functional testing was done in production.
  - Apex triggers ran without any issues.
  - Reports and dashboards showed precise, up-to-date data.

### Dashboard





## 2. Documentation

Comprehensive documentation was created to support users, developers, and administrators. It included:

- **Component-Level Documentation:**
  - **Custom Objects & Fields:** Purpose, field types, and relationships.
  - **Flows & Automation:** Process maps with entry points, logic, and expected outputs.
  - **Apex Triggers & Classes:** Description of functionality, invocation logic, and test coverage.
  - **Validation Rules & Approval Processes:** Condition logic and user impact.
- **User Role Matrix:**  
Summary of permissions assigned via Profiles and Permission Sets, including access control via Role Hierarchy and Sharing Rules.
- **Testing Results:**  
Summary of test cases with screenshots and outcomes for key functionalities such as:
  - Booking creation
  - Loyalty points allocation
  - Automatic task generation
  - Approval workflows
  - Record updates via Flows

## 3. Maintenance & Monitoring

- **Monitoring Strategy:**
  - Scheduled manual log checks for Apex errors or failed automation.

- Creation of dummy transactions to confirm functionality at regular intervals.
- **Issue Resolution Plan:**  
A lightweight issue tracker was proposed for the team to log bugs or enhancement requests.
- **Backup & Recovery:**  
Although an actual Salesforce organization would have automated backups, effective practices—like exporting reports and routinely backing up metadata via packages—were documented in this simulation.

## Scheduled Jobs

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

**Percentage of Scheduled Jobs Used: 1%**  
You have currently used 1 scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the [Lightning Platform Apex Limits](#) topic.

View: [All Scheduled Jobs](#) [Create New View](#)

Action	Job Name ↑	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID
<a href="#">Manage</a>   <a href="#">Del</a>   <a href="#">Pause Job</a>	Daily Inventory Sync	<a href="#">Qinam_Bhavrita</a>	7/11/2025, 9:40 AM	7/18/2025, 12:01 AM	7/19/2025, 12:00 AM	Scheduled Apex	08egl.0000070cpi
<a href="#">Del</a>	Loyalty_Status_Update_Flow-1	<a href="#">Qinam_Bhavrita</a>	7/11/2025, 9:30 AM	7/18/2025, 12:01 AM	7/19/2025, 12:00 AM	Scheduled Flow	08egl.0000070ZYY
<a href="#">Del</a>	Metalytics Data Loader Job for Org : 00Dgl.0000006ZZh0	<a href="#">User_Integration</a>	7/3/2025, 4:54 AM	7/17/2025, 6:16 PM	7/18/2025, 6:16 PM	Autonomous Data Loader Job	08egl.0000006aQwj
	Program Milestone Computation Cron Job	Process, Automated	7/3/2025, 4:54 AM	7/18/2025, 12:00 PM	7/18/2025, 4:59 PM	Program Milestone Computation Cron Job	08egl.0000006aQwh
	Program Status Update Cron Job	Process, Automated	7/3/2025, 4:54 AM	7/18/2025, 5:01 AM	7/18/2025, 8:00 PM	Program Status Update Cron Job	08egl.0000006aQwi

## **VIII. Conclusion**

The SmartBridge Salesforce Virtual Internship provided an invaluable opportunity to apply theoretical knowledge in a practical, industry-relevant context. I obtained practical experience in creating scalable data models, automating processes with Flows and Apex, and putting real-time business logic into place for more efficient operations by creating a CRM solution for HandsMen Threads. The project reinforced best practices in data integrity, user-centric design, and system maintenance while deepening my grasp of Salesforce's low-code and programmatic capabilities from the very beginning through deployment and documentation.

In addition to honing my technical abilities, this internship enhanced my problem-solving methodology, meticulousness, and comprehension of CRM-driven business processes, giving me the assurance I need to make a significant contribution to actual Salesforce implementations in the future.

## IX. Future Enhancements

The CRM can be expanded with the following features to increase automation, personalization, and analytical capabilities as the company grows and customer expectations change:

- **Integration of Einstein Bots with Chatbots**
  - The user experience can be greatly enhanced and the amount of manual support work reduced by implementing AI-powered chatbots to answer frequent customer inquiries like order tracking, appointment scheduling, and size recommendations.
- **AI-Powered Suggestions for Products**
  - Utilizing Salesforce Einstein AI to examine consumer preferences and past purchases in order to produce customized upsell/cross-sell possibilities or personalized style recommendations.
- **Integration of Mobile Apps**
  - Using Heroku or the Salesforce Mobile SDK, create a customer-facing mobile application that is integrated with the CRM to provide easy access to order history, loyalty status, and support services.
- **Improved Customer Feedback System:**
  - To collect data and make ongoing improvements to goods and services, automated post-purchase surveys and feedback forms connected to customer records are being introduced.
- **Omnichannel Support Integration:**
  - To guarantee consistent client communication and support, channels such as Facebook Messenger, Instagram direct messages, and WhatsApp are being integrated into the Service Console.
- **Management of Custom Loyalty Programs:**
  - Enhancing long-term client engagement by adding tiered memberships, incentive points, referral tracking, and redemption workflows to loyalty programs.
- **Automated Email Journeys and Campaigns:**
  - Using Pardot or Salesforce Marketing Cloud integration to build personalized email campaigns, abandoned basket alerts, and user behavior-based email journeys.
- **Integration of E-Commerce or Third-Party ERP:**
  - Integrating Salesforce with third-party platforms (such as Shopify, SAP, and Razorpay) to guarantee complete platform synchronization of orders, payments, and inventories.