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

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

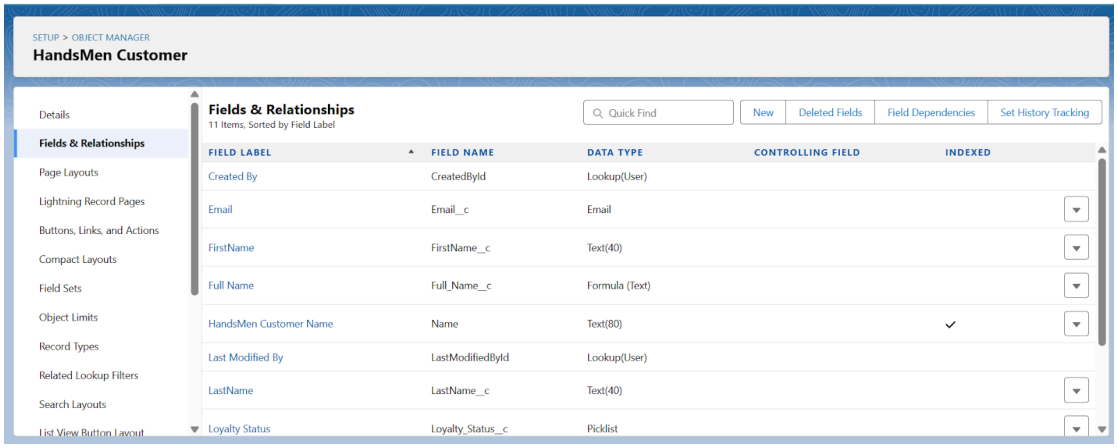
To build and test every component in a controlled context, the development process started with the setup of a Salesforce Developer Org. Important setup procedures included:

- 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.

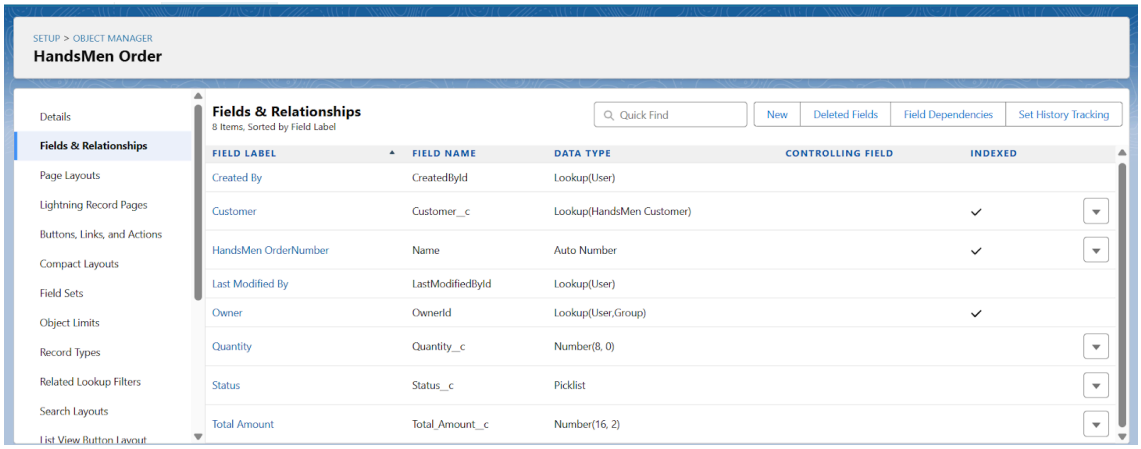
Customization of Objects, Fields & Relationships

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

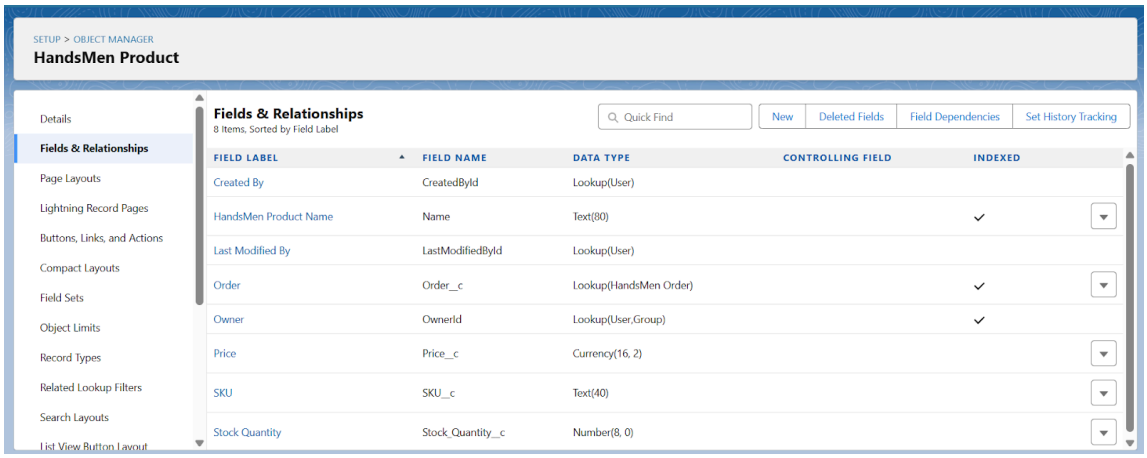
- **Customer__c:** To store client information such as name, contact, preferences, and loyalty tier.



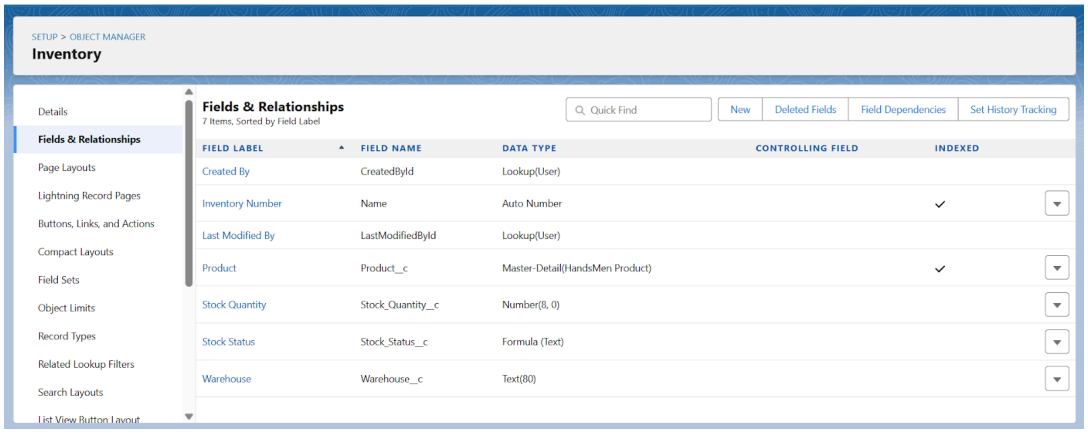
- **Order__c:** Linked to Customer and Product objects to track individual purchases.



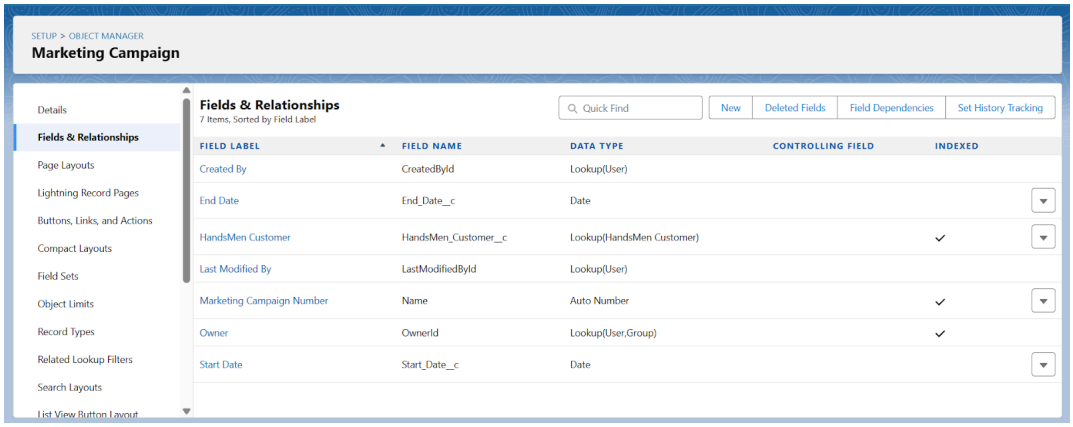
- **Product__c:** Maintained inventory details and product variants.



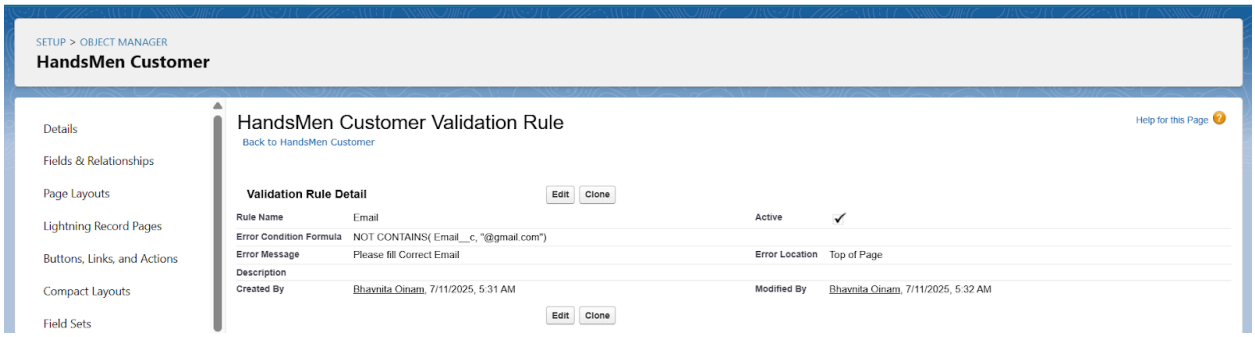
- **Inventory__c:** Managed stock levels for each product.

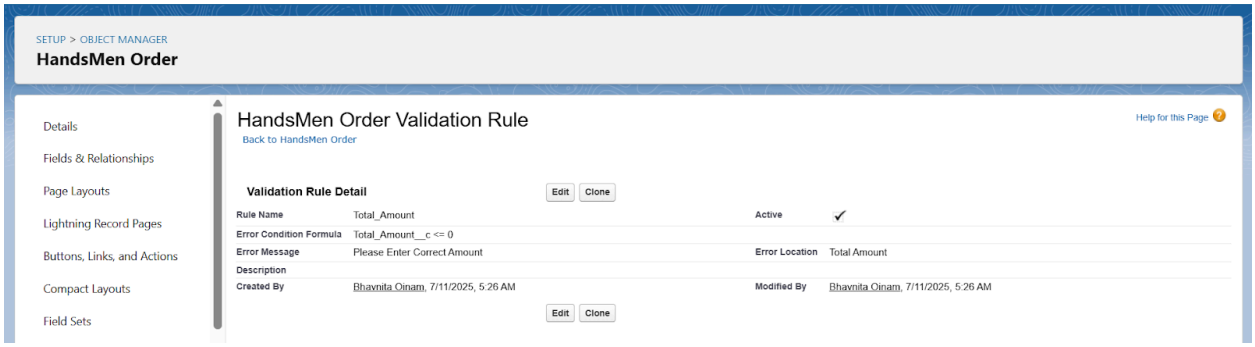


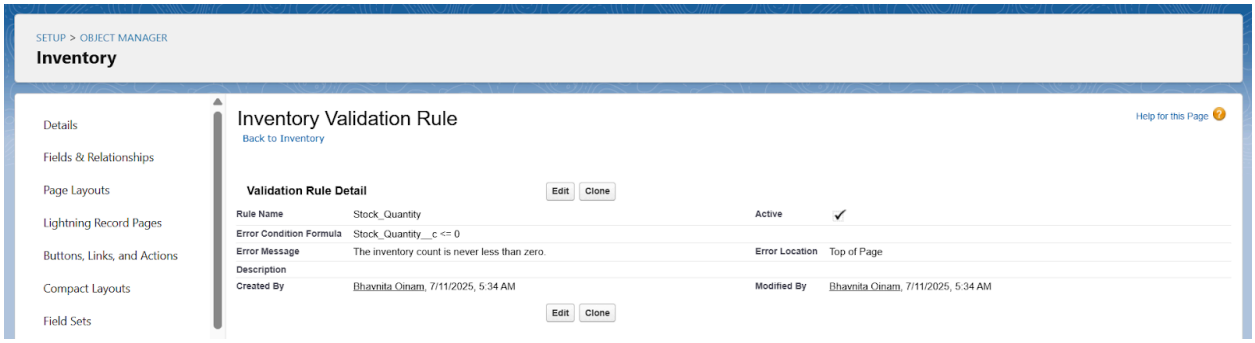
- **Marketing_Campaign__c:**



- **Validation Rules:**







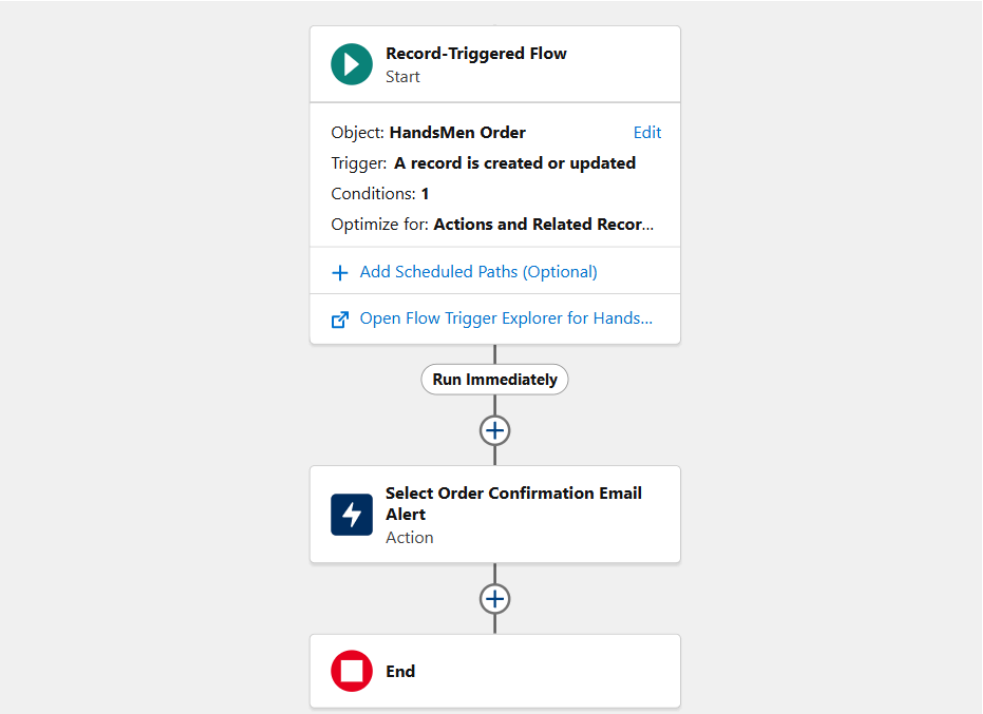
Automation: Workflow Rules, Process Builder, Flows, and Approval Processes

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

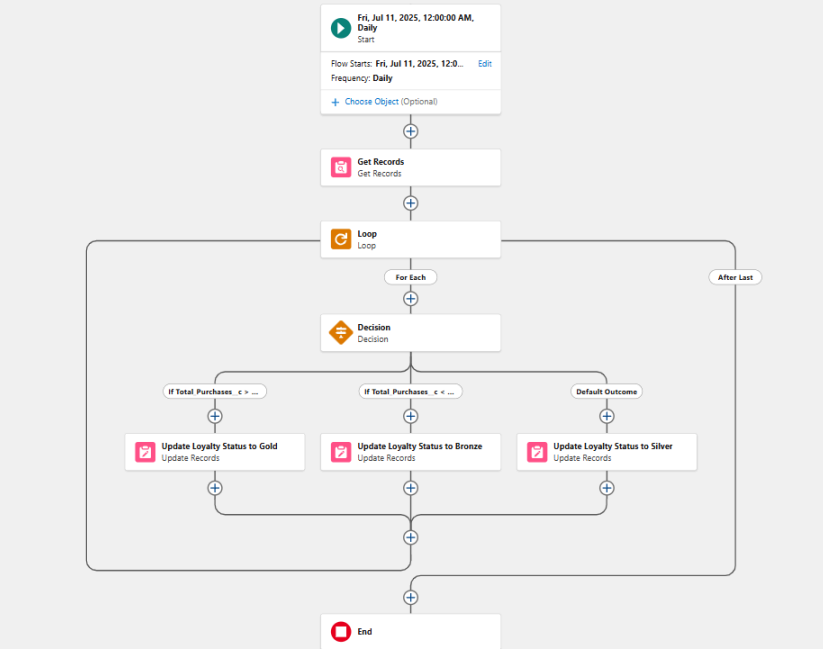
- **Record-Triggered Flows** (used instead of Process Builder due to best practice recommendations) to:
    - Send automatic email confirmations when an Order is created.
    - Update LoyaltyPoints__c when an Order is marked "Completed."
    - Trigger Inventory Alerts when Product stock < 5.

- **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.

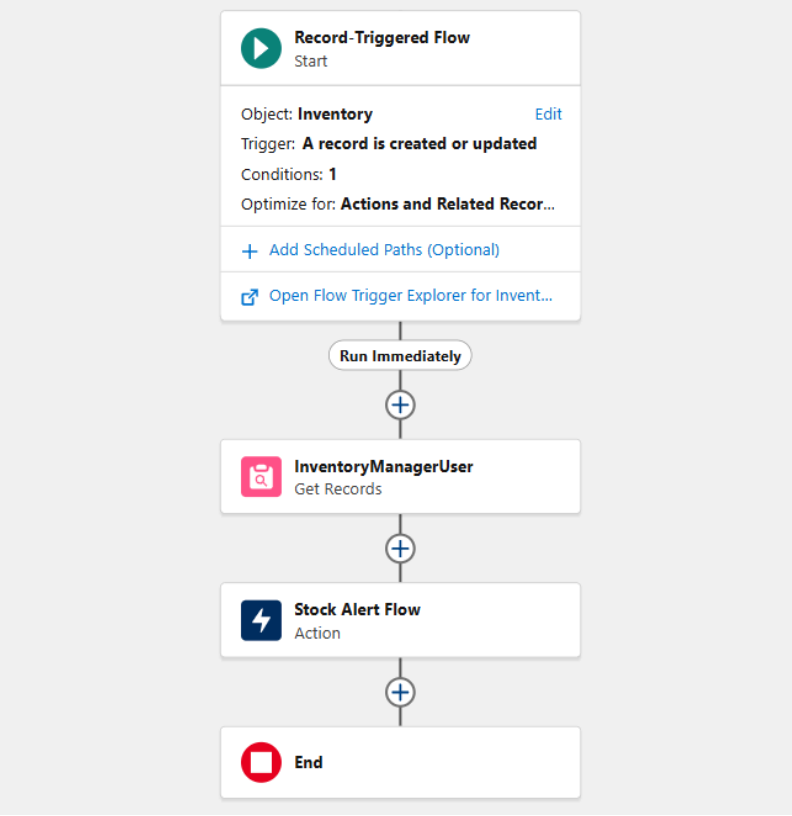
Order Confirmation Flow (Record-Triggered Flow)



Loyalty Status Update Flow (Scheduled Flow)



Low Stock Alert Flow (Record-Triggered Flow)



Apex Classes, Triggers & Asynchronous Apex

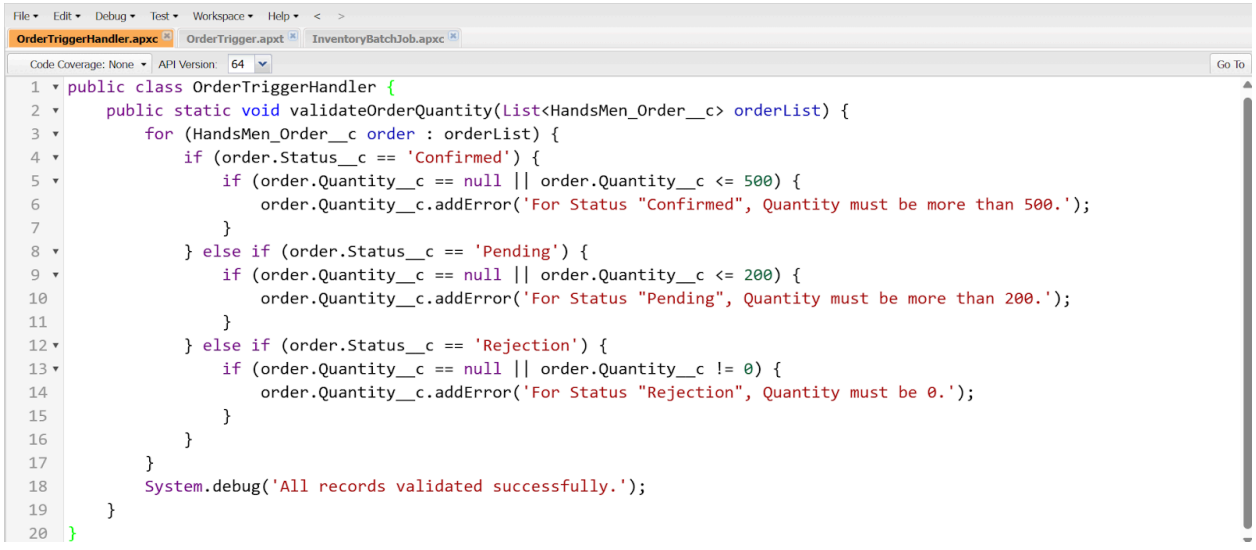
While most logic was handled through declarative tools, code-based automation was introduced for more complex operations:

- **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.

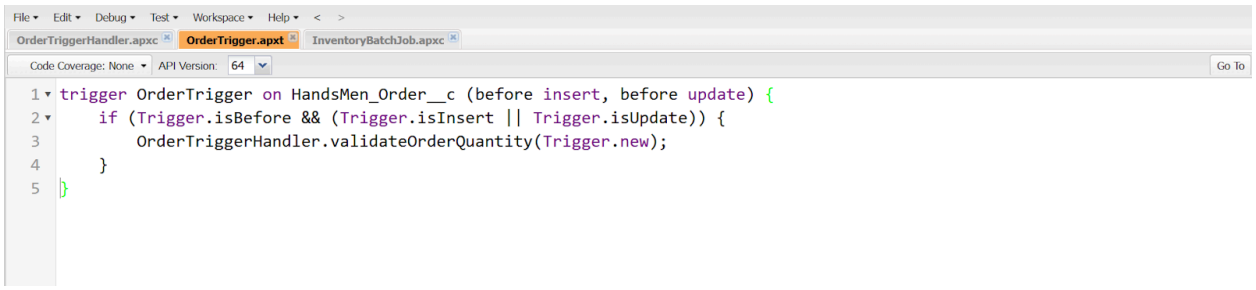
- **Batch Apex (Asynchronous):**
  - A scheduled `InventoryBatchJob` class was created to run at midnight every day.
  - It iterates over `Order__c` records with a "Pending" status, marks them as "Processed," updates totals, and triggers inventory adjustment.
  - This helped simulate real-world scheduled tasks with large data volumes.

OrderTriggerHandler Apex Class



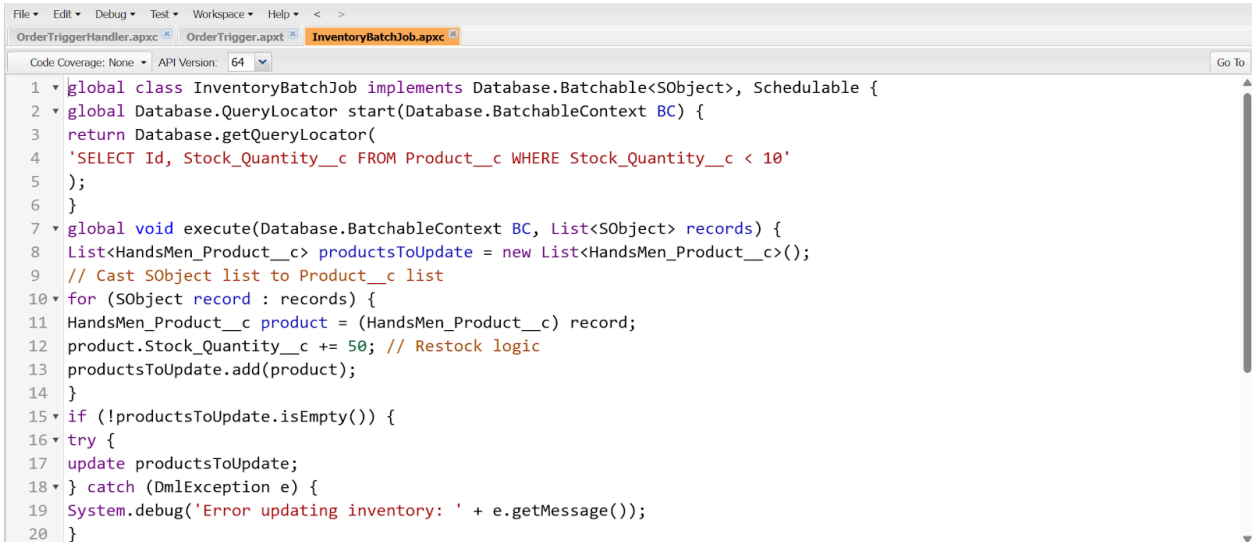

```apex
public class OrderTriggerHandler {
    public static void validateOrderQuantity(List<HandsMen_Order__c> orderList) {
        for (HandsMen_Order__c order : orderList) {
            if (order.Status__c == 'Confirmed') {
                if (order.Quantity__c == null || order.Quantity__c <= 500) {
                    order.Quantity__c.addError('For Status "Confirmed", Quantity must be more than 500.');
                }
            } else if (order.Status__c == 'Pending') {
                if (order.Quantity__c == null || order.Quantity__c <= 200) {
                    order.Quantity__c.addError('For Status "Pending", Quantity must be more than 200.');
                }
            } else if (order.Status__c == 'Rejection') {
                if (order.Quantity__c == null || order.Quantity__c != 0) {
                    order.Quantity__c.addError('For Status "Rejection", Quantity must be 0.');
                }
            }
        }
        System.debug('All records validated successfully.');
    }
}
```

OrderTrigger Apex Trigger




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

InventoryBatchJob Apex Class




```
global class InventoryBatchJob implements Database.Batchable<SObject>, Schedulable {
global Database.QueryLocator start(Database.BatchableContext BC) {
return Database.getQueryLocator(
'SELECT Id, Stock_Quantity__c FROM Product__c WHERE Stock_Quantity__c < 10'
);
}
global void execute(Database.BatchableContext BC, List<SObject> records) {
List<HandsMen_Product__c> productsToUpdate = new List<HandsMen_Product__c>();
// Cast SObject list to Product__c list
for (SObject record : records) {
HandsMen_Product__c product = (HandsMen_Product__c) record;
product.Stock_Quantity__c += 50; // Restock logic
productsToUpdate.add(product);
}
if (!productsToUpdate.isEmpty()) {
try {
update productsToUpdate;
} catch (DmlException e) {
System.debug('Error updating inventory: ' + e.getMessage());
}
```

```
}
}
global void finish(Database.BatchableContext BC) {
System.debug('Inventory Sync Completed');
}
// Scheduler Method
global void execute(SchedulableContext SC) {
InventoryBatchJob batchJob = new InventoryBatchJob();
Database.executeBatch(batchJob, 200);
}
}
```

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

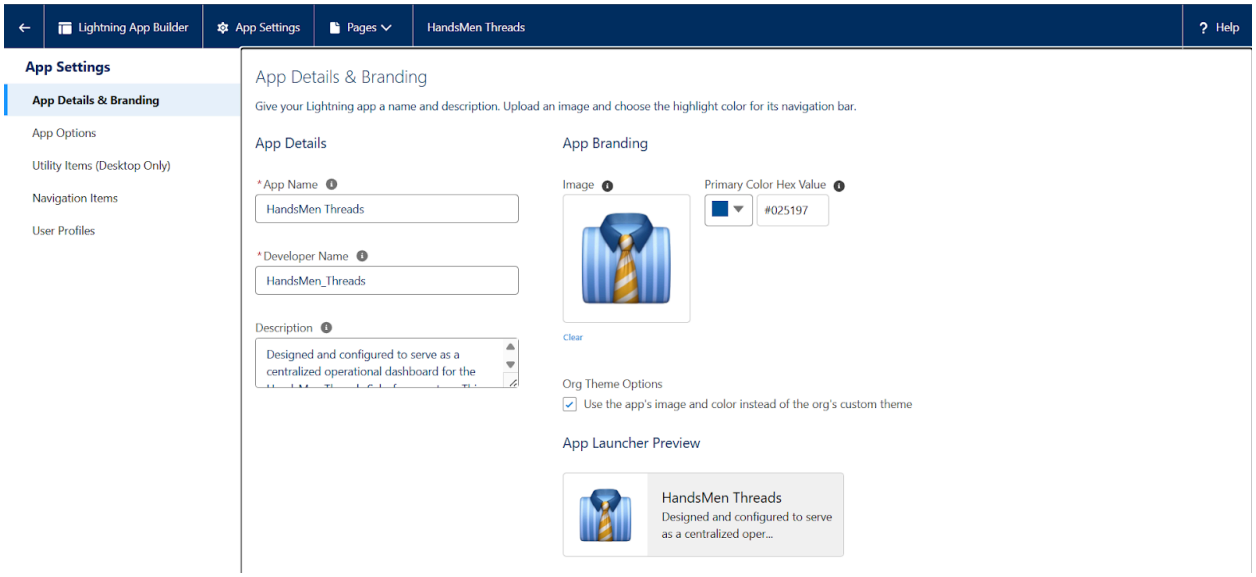
- Profile and Permission Set Configuration:
    - Configuring profiles and permission sets made that users could access only pertinent fields, records, and tabs.
    - Sensitive fields such as internal stock alerts or financial data are subject to read/write restrictions.
- Field-Level Security Testing:
    - 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

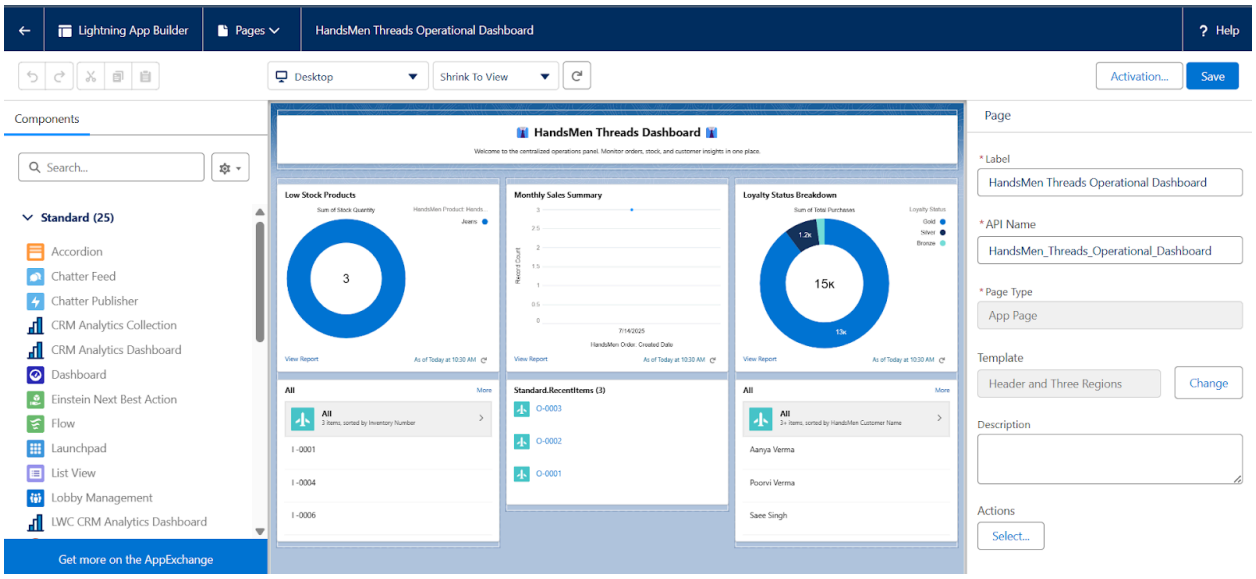
Minor errors and logical inconsistencies were found and fixed during testing:

- 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



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
Low Stock Products
Monthly Sales Summary
Loyalty Status Breakdown
Standard.RecentItems (3)
O-0003
O-0002
O-0001
Aanya Verma
Poorvi 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

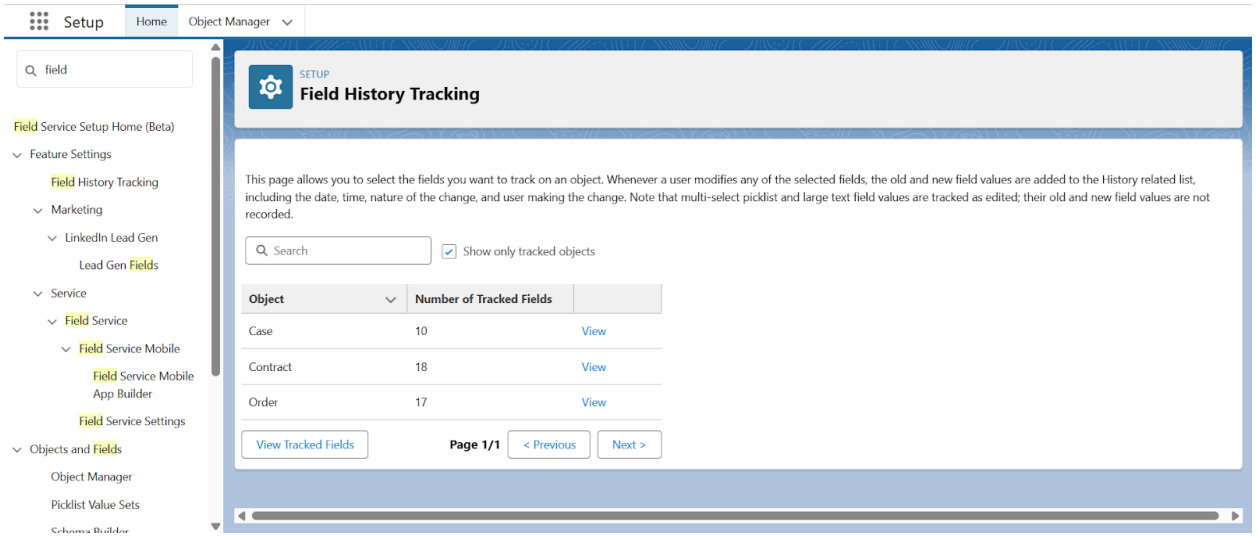
Customized training was necessary because HandsMen Threads is a company with many jobs, including as sales executives, warehouse managers, stylists, and administrators:

- 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.

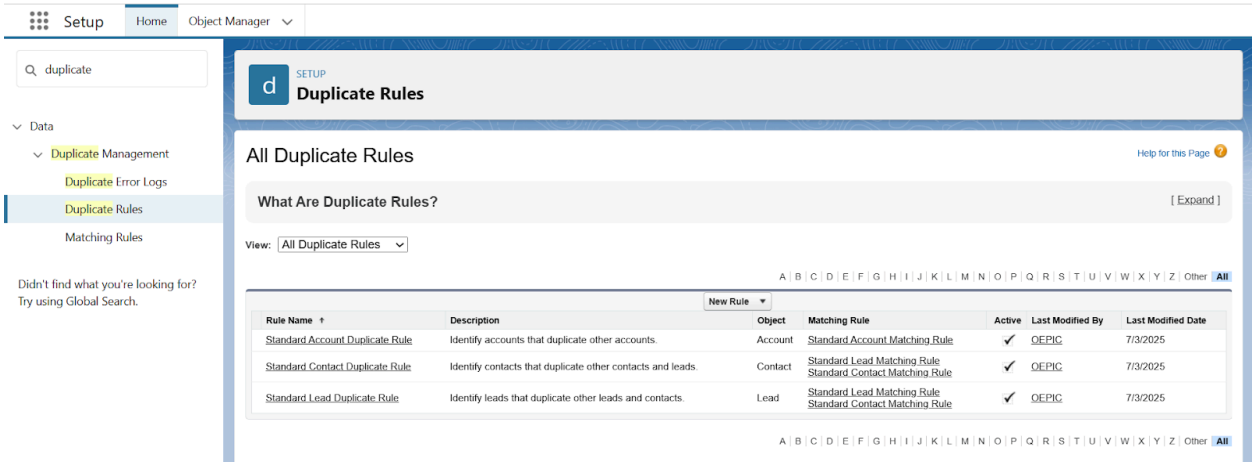
- Admins: Taught how to manage support tickets, make updates, and keep an eye on batch jobs.

- 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.

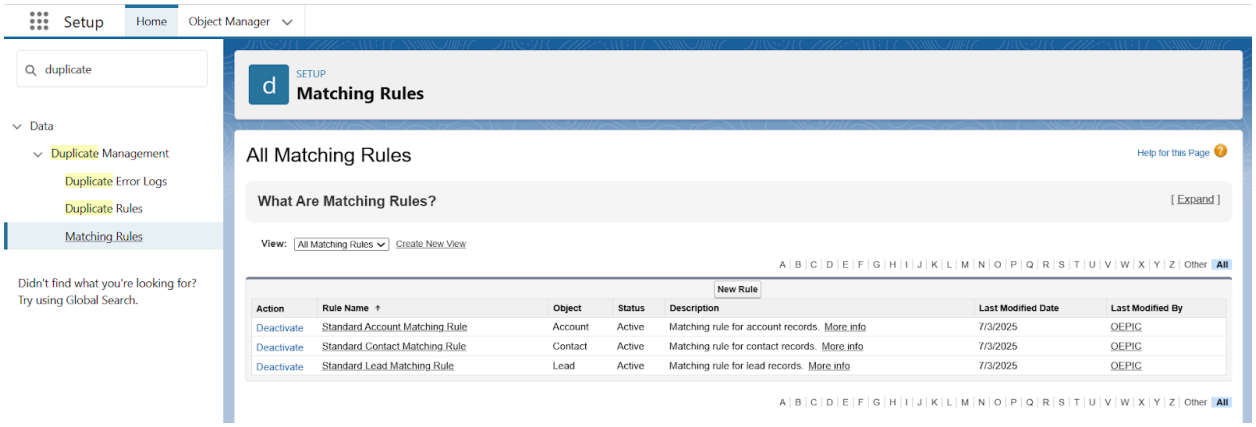
Field History Tracking



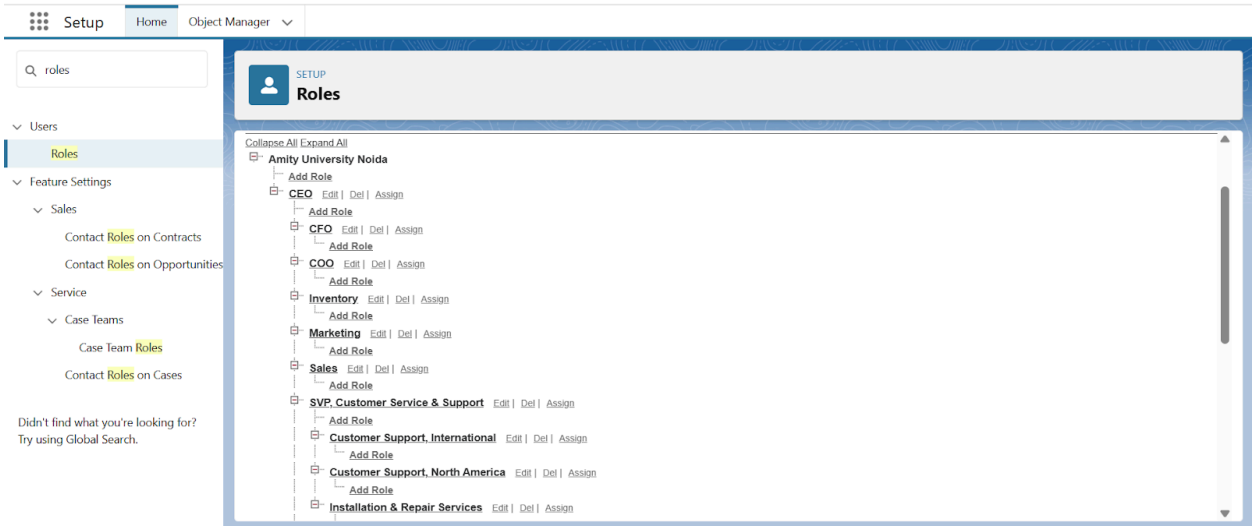
Duplicate Rules



Matching Rules



Roles



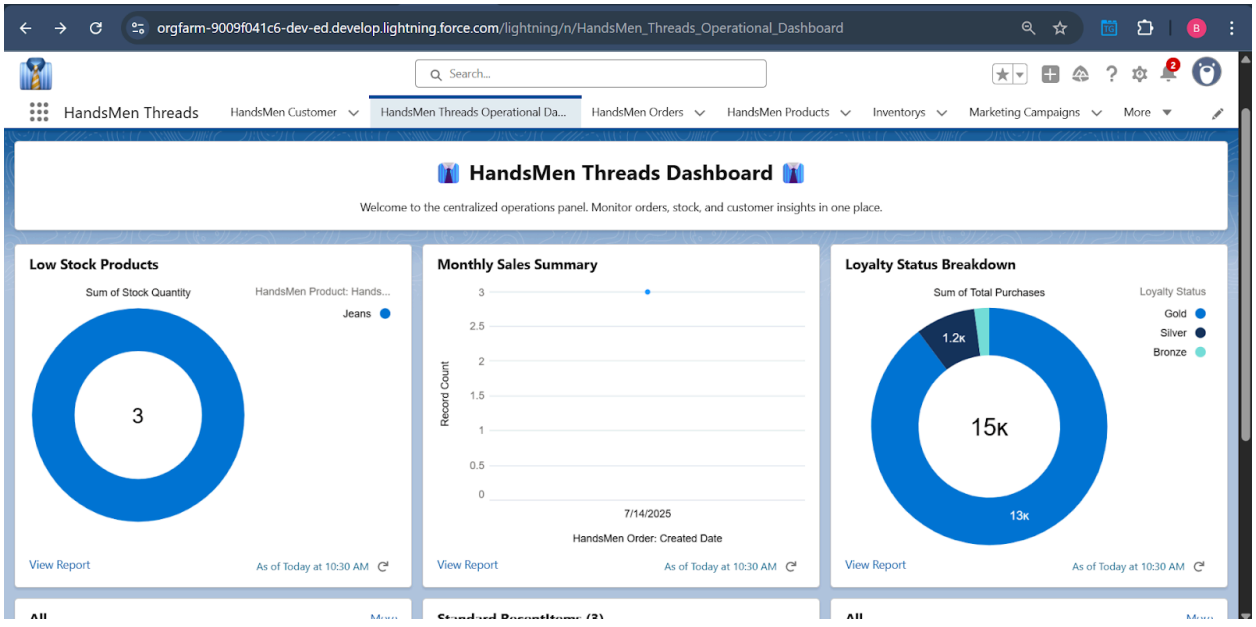
VII. Phase 5: Deployment, Documentation & Maintenance

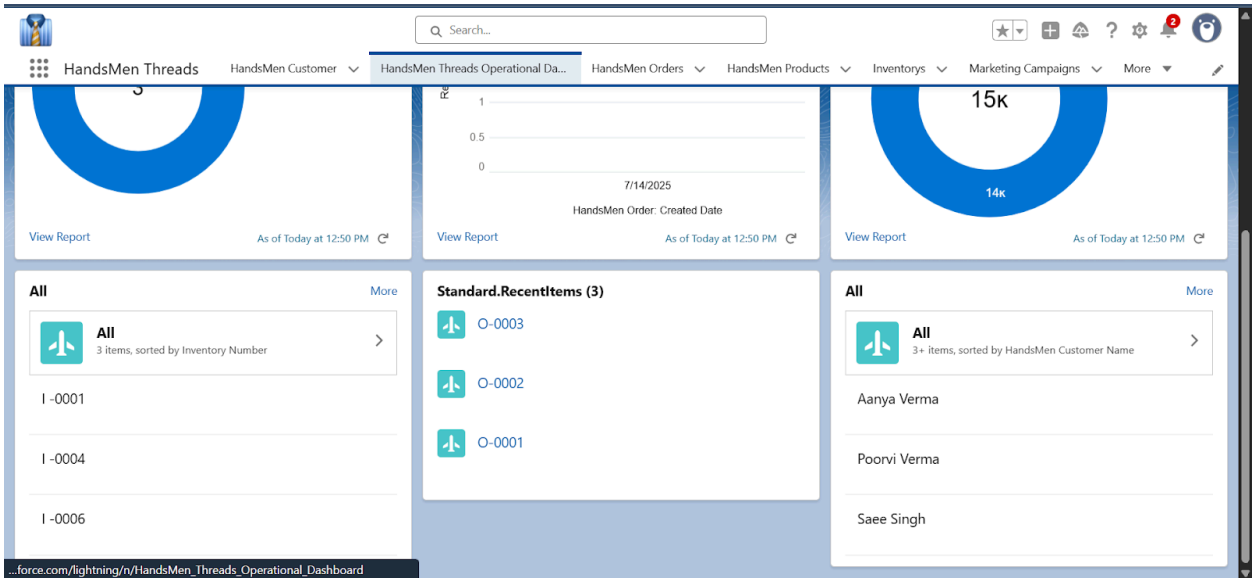
Assuring a seamless transition from development to production, preserving system stability, and providing stakeholders with the information and resources required for long-term sustainability were the key objectives of the project's last phase. By recording every step of the build and creating a plan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

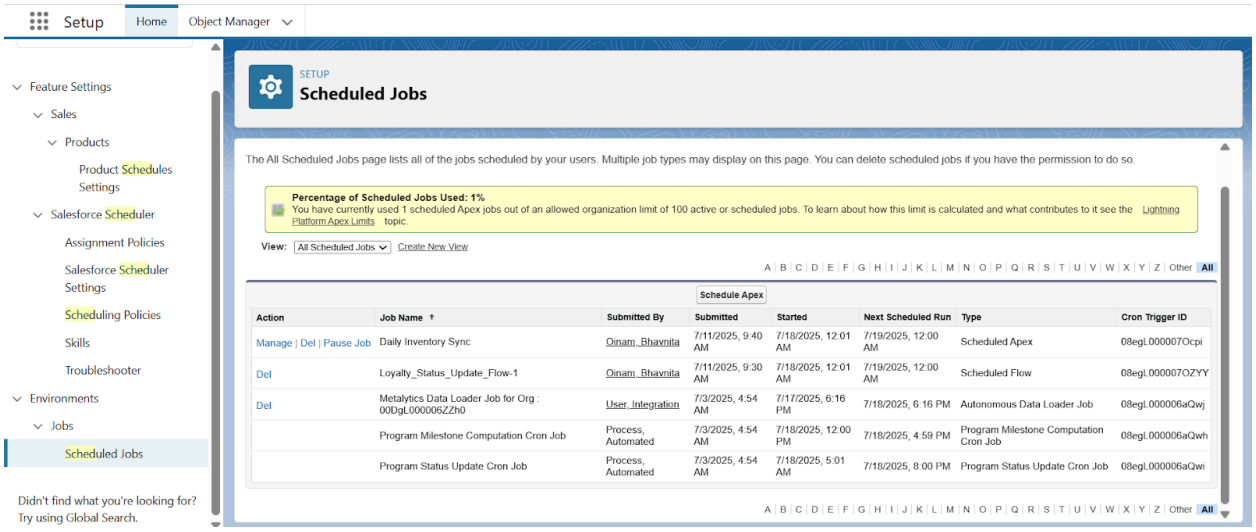
3. Maintenance & Monitoring

- **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



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.