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

Abstract:

This document outlines a Salesforce project for HandsMen Threads, a fashion industry organization, focusing on transforming their data management and customer relations. The core objective is to establish a **robust data model** to ensure data integrity and facilitate a seamless flow of information throughout the organization.

The project emphasizes maintaining **data quality** directly from the user interface, crucial for accurate decision-making. Key integrations include **automated order confirmations** to enhance customer engagement, a **dynamic loyalty program** that personalizes rewards based on purchase history, and **proactive stock alerts** for efficient inventory management. Additionally, **scheduled bulk order updates** will ensure accurate financial and inventory records.

The project phases encompass Architecture & Planning, Development, Testing & QA, and Deployment & Training. Deliverables include a comprehensive Solution Design

Document, detailing the Object Model, Entity-Relationship Diagram (ERD), and Automation Strategy. This implementation leverages Salesforce capabilities such as **Lightning App Builder**, **Record Triggered Flows**, **Apex and Apex Triggers**, and **Asynchronous Apex** to achieve these transformative goals.

Objective:

The primary objective of this Salesforce project for HandsMen Threads is to **revolutionize data management and significantly enhance customer relations** by implementing a robust and integrated Salesforce platform. This will ensure data integrity, streamline business processes, and improve operational efficiency through automation, ultimately fostering stronger customer engagement and enabling informed decision-making.

Technology Description:

1. **Data Modelling:** Designing the structure of data within Salesforce, including custom objects, fields, and relationships, to accurately represent HandsMen Threads' business information.
2. **Data Quality:** Implementing mechanisms (such as validation rules) and processes to ensure the accuracy, consistency, and reliability of data within the Salesforce system, directly from the user interface.

3. **Lightning App Builder:** A declarative tool within Salesforce used to create custom user interfaces (apps, record pages, home pages) with drag-and-drop functionality, enhancing the user experience for HandsMen Threads' employees.
4. **Record Triggered Flows:** Automation tools that execute actions (e.g., sending emails, updating records) automatically when a Salesforce record is created, updated, or deleted, used for processes like order confirmations and loyalty program updates.
5. **Apex and Apex Triggers:** Salesforce's proprietary programming language (Apex) and event-driven code (Apex Triggers) used for complex business logic, custom validations, and advanced automation that cannot be achieved with declarative tools.
6. **Asynchronous Apex:** Apex code executed in the background, allowing for long-running operations (like scheduled bulk order updates) without impacting the performance of the user interface or hitting governor limits for real-time transactions.

Detailed execution of project phases:

Phase 1: Architecture & Planning

This phase focuses on defining the "what" and "how" of the solution, laying a solid foundation for development.

- **1.1 Requirements Elicitation & Analysis:**
  - Conduct in-depth workshops and interviews with key stakeholders from Sales, Customer Service, Warehouse, and Finance departments to understand current processes, pain points, and desired future states.
  - Document detailed functional and non-functional requirements for data management, customer relations, and specific automation needs (Order Confirmations, Loyalty Program, Stock Alerts, Bulk Order Updates).
  - Identify key performance indicators (KPIs) for success.
- **1.2 Data Model Definition:**
  - **Object Identification:** Define all necessary standard and custom objects (e.g., Customer__c, Order__c, Product__c, Loyalty_Status__c, Inventory__c, Bulk_Order_Log__c).
  - **Field Definition:** For each object, specify fields including data type, length, required status, default values, and help text.
  - **Relationship Mapping:** Establish relationships (Master-Detail, Lookup) between objects (e.g., Order__c to Customer__c, Inventory__c to Product__c).

    - **Formula & Roll-up Summary Fields:** Define calculated fields (e.g., Order__c.Total_Amount__c) and roll-up summaries (e.g., Customer__c.Total_Purchases__c) to derive insights.
- **1.3 Entity-Relationship Diagram (ERD) Creation:**
    - Visually represent the defined data model, illustrating all objects, their fields, and the relationships between them, ensuring clarity and accuracy.
- **1.4 Automation Strategy Design:**
    - **Automated Order Confirmations:** Design a **Record-Triggered Flow** (After Save on Order__c object) to trigger an email alert to the customer upon order status change to 'Confirmed'.
    - **Dynamic Loyalty Program:** Design a **Record-Triggered Flow** (After Save on Order__c object) to update the Loyalty_Status__c on the Customer__c record based on accumulated purchase history. Complex loyalty tier logic might necessitate an **Apex Trigger** called by the Flow.
    - **Proactive Stock Alerts:** Design a **Record-Triggered Flow** (After Save on Inventory__c or Product__c object) to check Stock_Level__c and send an email notification to the warehouse team if it drops below 5 units.

    - **Scheduled Bulk Order Updates:** Design the logic for an **Apex Batch Job** to run daily at midnight. This job will query pending bulk orders, process them, update Order__c and Inventory__c records, and log details in Bulk_Order_Log__c.
- **1.5 Data Quality & Validation Strategy:**
    - Define **Validation Rules** for critical fields to ensure data integrity at the point of entry (e.g., Order_Date__c cannot be in the future, Stock_Level__c must be a positive number).
    - Outline data cleansing and migration considerations (if applicable).
- **1.6 Email Template Design:**
    - Develop detailed content and visual designs for all automated email templates (Order Confirmation, Low Stock Alert).
    - Identify necessary merge fields for personalization.
- **1.7 Security & Sharing Model Planning:**
    - Define Profiles and Permission Sets to control object and field-level access based on user roles (e.g., Sales, Warehouse, Admin).
    - Determine Organization-Wide Defaults (OWD) and design Sharing Rules to manage record-level visibility (e.g., sales reps only see their own accounts, but managers see all).

- **1.8 Solution Design Document (SDD) Creation:**
    - Compile all architectural decisions, detailed object models, ERD, automation strategies, security plans, and email template designs into a comprehensive SDD. This document will serve as the blueprint for development.

Phase 2: Development

This phase involves building and configuring the Salesforce solution based on the approved design.

- **2.1 Salesforce Environment Setup:**
    - Provision a development sandbox (e.g., Developer Pro or Full Sandbox) for the project.
    - Enable necessary Salesforce features and settings.
- **2.2 Object and Field Creation:**
    - Create all custom objects and fields as defined in the SDD.
    - Establish relationships between objects.
    - Configure formula fields and roll-up summary fields.
- **2.3 Validation Rule Implementation:**
    - Implement and activate all defined validation rules to enforce data quality.
- **2.4 Flow Development & Activation:**

- Build and test the **Record-Triggered Flow** for Automated Order Confirmations.
- Develop and test the **Record-Triggered Flow** for Dynamic Loyalty Program updates.
- Create and test the **Record-Triggered Flow** for Proactive Stock Alerts.
- Ensure all Flows are activated and correctly configured to fire on the specified record events.

- **2.5 Apex and Apex Trigger Development:**
  - Develop any required **Apex Triggers** for complex logic (e.g., if loyalty logic is too complex for Flow).
  - Develop the **Apex Batch Job** for Scheduled Bulk Order Updates, including the execute and finish methods.
  - Write comprehensive **Apex Test Classes** for all Apex code, ensuring high code coverage (minimum 75%) and testing various scenarios, including positive, negative, and bulk data processing.
- **2.6 Data Security and Sharing Rule Configuration:**
  - Create and configure Profiles and Permission Sets.
  - Set up Organization-Wide Defaults (OWD) and implement Sharing Rules to enforce the designed data visibility.

- **2.7 Email Template Configuration:**
    - Create and configure Email Templates within Salesforce, incorporating merge fields for dynamic content.
    - Associate templates with the respective Flows.
- **2.8 Lightning App Builder Configuration:**
    - Design and build the main "HandsMen Threads" Lightning App.
    - Customize Lightning Record Pages for key objects (e.g., Customer, Order, Product) to display relevant information, related lists, and quick actions.
    - Configure Lightning Home Pages and Utility Bars as needed for user productivity.

Phase 3: Testing & QA

This phase ensures the developed solution is robust, functional, and meets all requirements.

- **3.1 Unit Testing:**
    - **Apex Unit Tests:** Run all Apex test classes to ensure code coverage and functional correctness of Apex triggers and batch jobs. Review test results for failures.
    - **Flow Unit Tests:** Individually test each Flow using the Flow Debugger and by manually

creating/updating records to verify correct execution paths and outcomes.

- **3.2 Integration Testing:**
    - Test the end-to-end flow of data and automation across multiple components. For example, create a new order, confirm it, and verify that the customer receives an email, and their loyalty status is updated.
    - Test the interaction between Flows and Apex (if an Apex trigger is invoked by a Flow).
- **3.3 End-to-End Testing (User Acceptance Testing - UAT):**
    - Prepare a comprehensive UAT test plan with detailed test cases covering all business processes and user scenarios.
    - Populate the UAT sandbox with realistic sample data.
    - Engage key business users from HandsMen Threads to execute the test cases.
    - Gather user feedback and meticulously log all defects, prioritizing them based on severity.
- **3.4 Performance Testing:**
    - Test the performance of the Apex Batch Job with a simulated large volume of bulk orders to ensure it completes within governor limits and acceptable timeframes.

    - Monitor the performance of Flows and UI responsiveness.
- **3.5 Security Testing:**
    - Verify that the implemented security model (Profiles, Permission Sets, OWD, Sharing Rules) correctly restricts or grants access to data and functionalities as designed.
    - Test with different user profiles to ensure data visibility is as expected.
- **3.6 Defect Resolution & Retesting:**
    - Address all identified bugs and issues promptly.
    - Perform regression testing to ensure that fixes do not introduce new issues.
    - Obtain sign-off from business users on the UAT results.

Phase 4: Deployment & Training

This final phase involves moving the solution to production and preparing users for the new system.

- **4.1 Deployment Planning:**
    - Create a detailed deployment plan, outlining the sequence of steps, dependencies, and rollback procedures.
    - Identify all metadata components to be deployed (objects, fields, flows, Apex classes/triggers,

email templates, profiles, permission sets, sharing rules, Lightning App/pages).

- Choose the deployment tool (e.g., Change Sets, Salesforce DX, or a third-party CI/CD tool).

- **4.2 Pre-Deployment Activities:**
  - Ensure all tests (unit, integration, UAT) have passed in the staging sandbox.
  - Perform a final code review and configuration audit.
  - Communicate the deployment schedule to all stakeholders.
- **4.3 Production Deployment:**
  - Execute the deployment plan, carefully monitoring the process for any errors or failures.
  - Perform a quick "smoke test" in the production environment to verify core functionalities are working immediately after deployment.
- **4.4 User Training:**
  - Develop comprehensive training materials, including user manuals, quick reference guides, and FAQs, tailored to HandsMen Threads' specific processes.
  - Conduct hands-on training sessions for all affected user groups, demonstrating the new Salesforce functionalities and emphasizing the

new automated processes (order confirmations, loyalty updates, stock alerts).
    - Provide opportunities for users to practice in a training environment.
- **4.5 Post-Go-Live Support & Monitoring:**
    - Provide dedicated support during the initial period after go-live to address any immediate user questions or issues.
    - Monitor system performance, error logs, and user adoption rates.
    - Establish a feedback mechanism for continuous improvement.
    - Plan for ongoing maintenance and future enhancements.

Project explanation with real-world example:

Scenario: Meet **Sarah**, a new customer who just discovered HandsMen Threads.

1. **Sarah Places an Order (Data Model & Data Quality in Action):**
    - Sarah visits the HandsMen Threads website and places an order for a new designer dress.

- When her order comes into the system (or is manually entered by a sales associate using the new **Lightning App Builder** interface), Salesforce creates a new Order record and, if she's a new customer, a Customer record.
- **Data Quality:** As the sales associate enters the details, a **Validation Rule** (part of Data Quality) immediately flags if they try to enter a future date for the order, preventing errors right away. The Order record is automatically linked to Sarah's Customer record (a **relationship** in the Data Model).

2. **Automated Order Confirmation (Record-Triggered Flow):**
    - The moment Sarah's order status changes to "Confirmed" in Salesforce, a **Record-Triggered Flow** automatically springs into action.
    - **Real-world impact:** Sarah instantly receives a beautifully branded email confirming her purchase, tracking number, and a thank-you message. This makes her feel valued and informed, without any manual effort from HandsMen Threads staff.
3. **Dynamic Loyalty Program (Record-Triggered Flow & potentially Apex):**

- Sarah loves her dress and buys another one a month later. Her second purchase pushes her total spending over a certain threshold.
- Another **Record-Triggered Flow** (or an **Apex Trigger** if the loyalty logic is very complex) automatically updates her Loyalty_Status__c field on her Customer record from "Bronze" to "Silver."
- **Real-world impact:** HandsMen Threads can now automatically send Sarah a personalized email with a "Silver Member" discount code, encouraging her next purchase. This fosters loyalty and repeat business without anyone having to manually track her spending or change her status.

4. **Proactive Stock Alert (Record-Triggered Flow):**
    - Sarah's purchase of the designer dress, along with other recent sales, causes the Stock_Level__c for that specific dress style to drop to 4 units in the Inventory object.
    - A **Record-Triggered Flow** monitoring the Inventory object detects this low stock level.
    - **Real-world impact:** An automated email is immediately sent to the warehouse team, alerting them that the "Designer Dress - Style X" needs to be reordered. This prevents stockouts,

ensures popular items are always available, and avoids disappointed customers.

5. **Scheduled Bulk Order Updates (Asynchronous Apex - Batch Job):**
    - HandsMen Threads also deals with large corporate clients who place bulk orders. These orders often involve complex pricing and inventory adjustments.
    - Every night at midnight, an **Apex Batch Job** (a form of **Asynchronous Apex**) automatically runs. It processes all bulk orders that were marked "Pending" during the day, updates the financial records, and adjusts the inventory for hundreds of items simultaneously.
    - **Real-world impact:** The finance team wakes up to accurate, reconciled records, and the inventory team has precise stock levels reflected for the start of the new business day. This eliminates manual reconciliation, reduces errors, and ensures efficient daily operations without slowing down the system during peak hours.

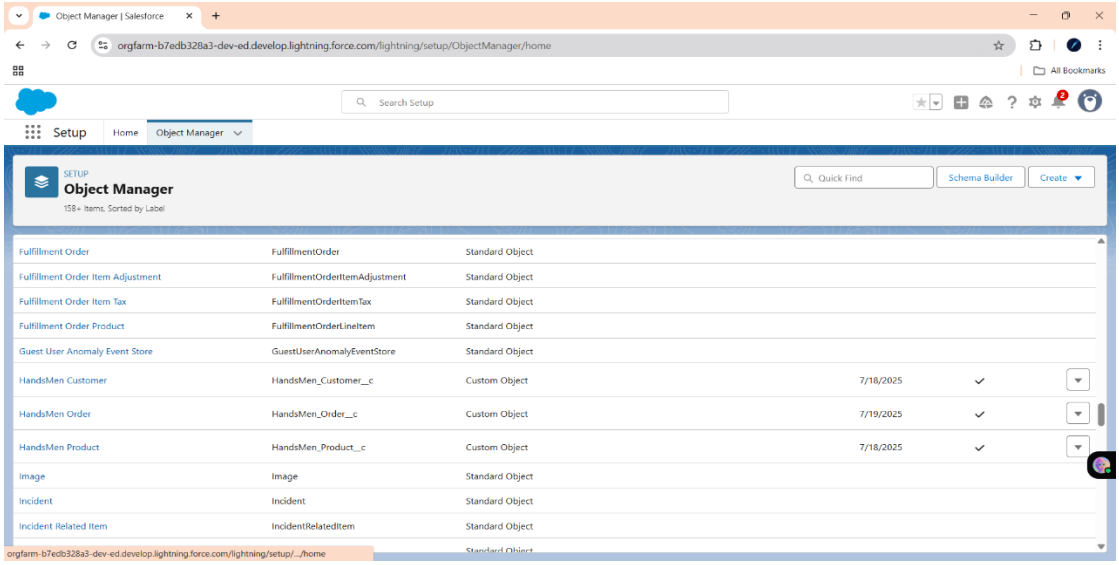
Overall Impact for HandsMen Threads:

- **Single Source of Truth:** All customer, order, and inventory data is now in one place, accessible to everyone who needs it.

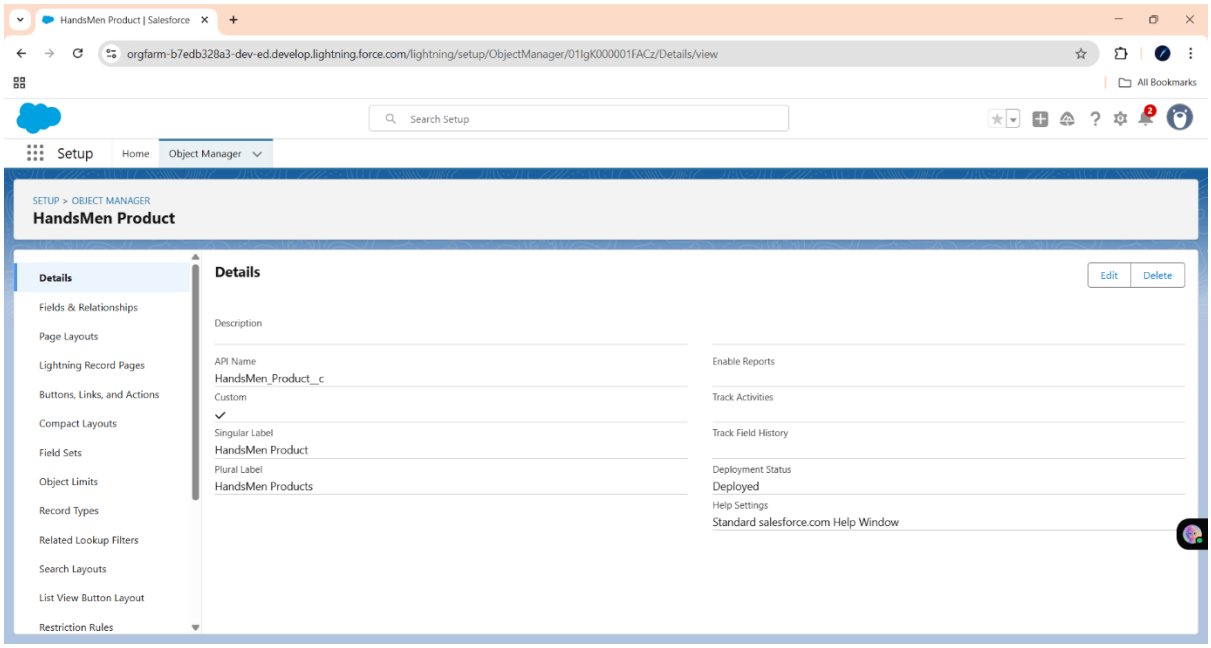
- **Improved Customer Experience:** Customers receive timely, personalized communication, making them feel valued.
- **Increased Efficiency:** Manual tasks are eliminated, freeing up staff to focus on more strategic activities.
- **Better Decision-Making:** Accurate, real-time data on stock and sales allows for smarter purchasing and marketing strategies.
- **Scalability:** The system is built to handle future growth and increasing order volumes.

This project empowers HandsMen Threads to operate more efficiently, serve their customers better, and make data-driven decisions, all powered by the robust capabilities of the Salesforce platform.

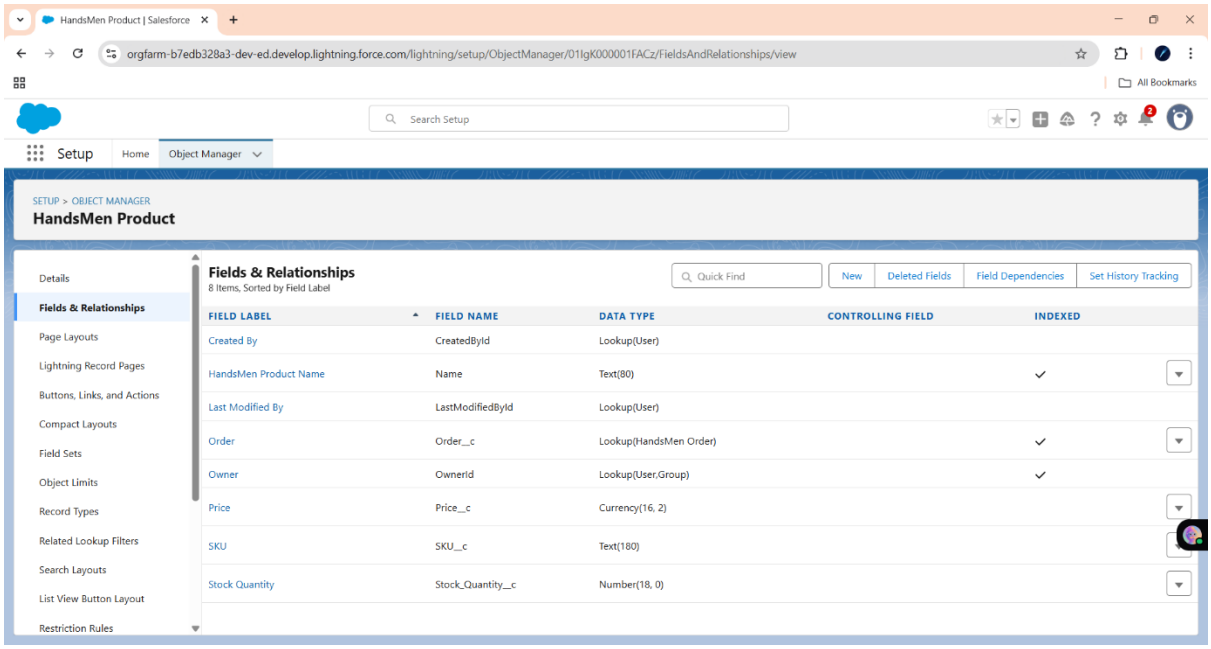
Screenshots of each phase:



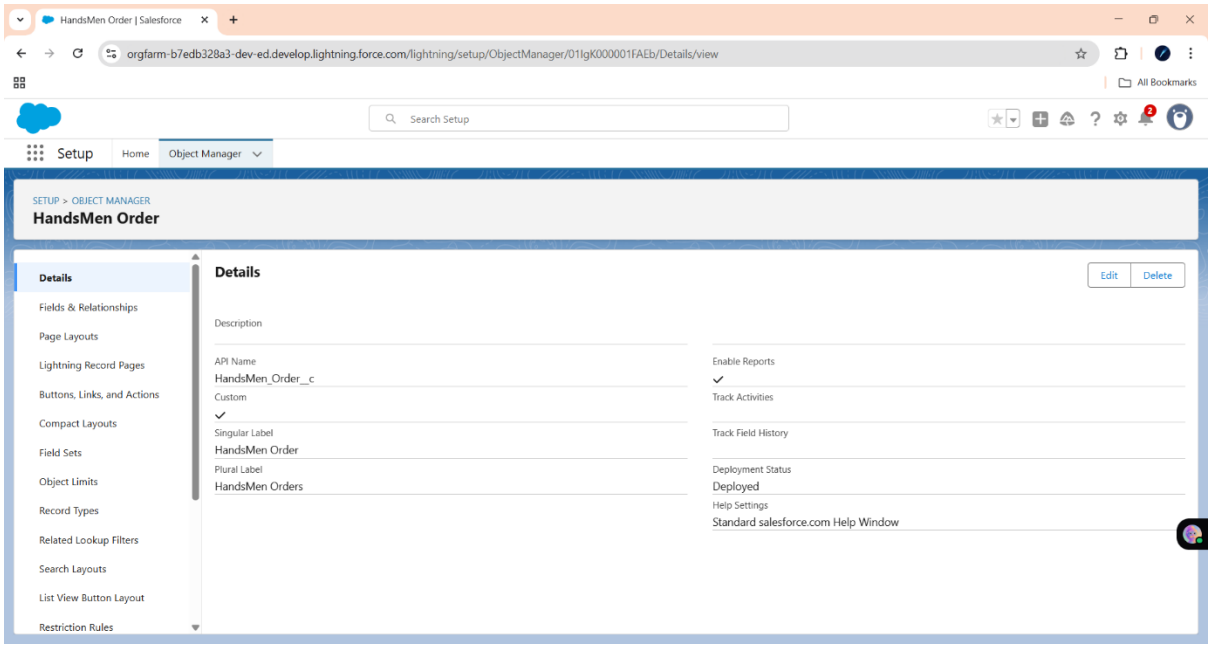
Go to the object manager and create 3 objects as Customer, product and order



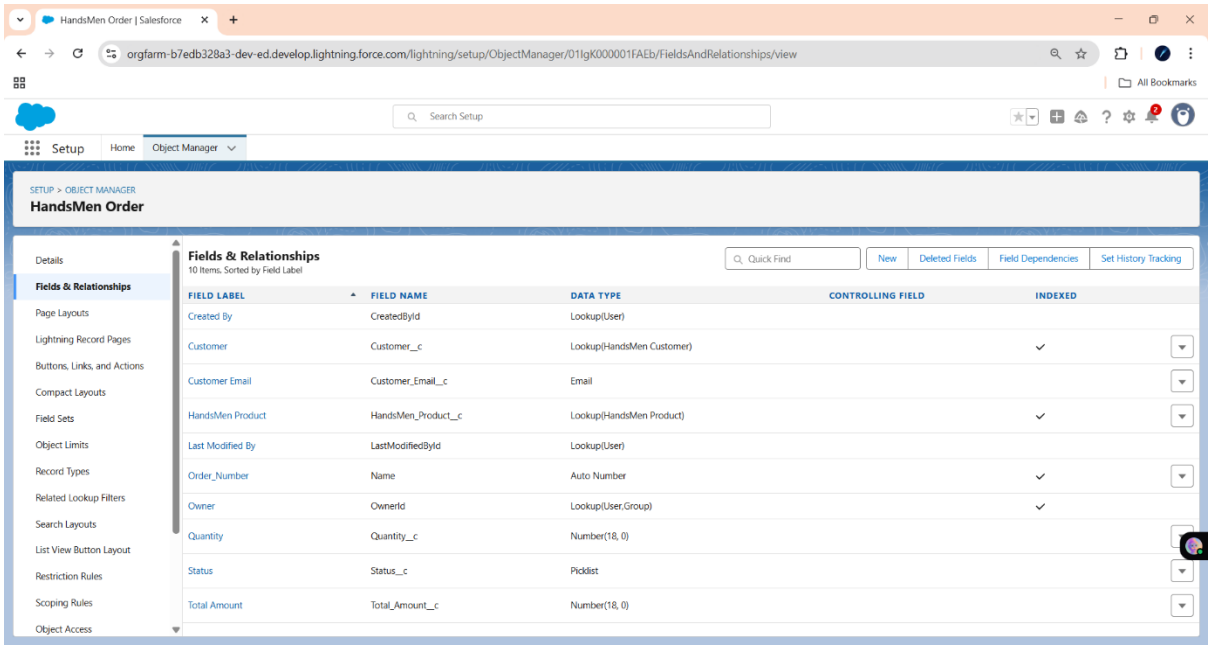
Create the product object accordingly



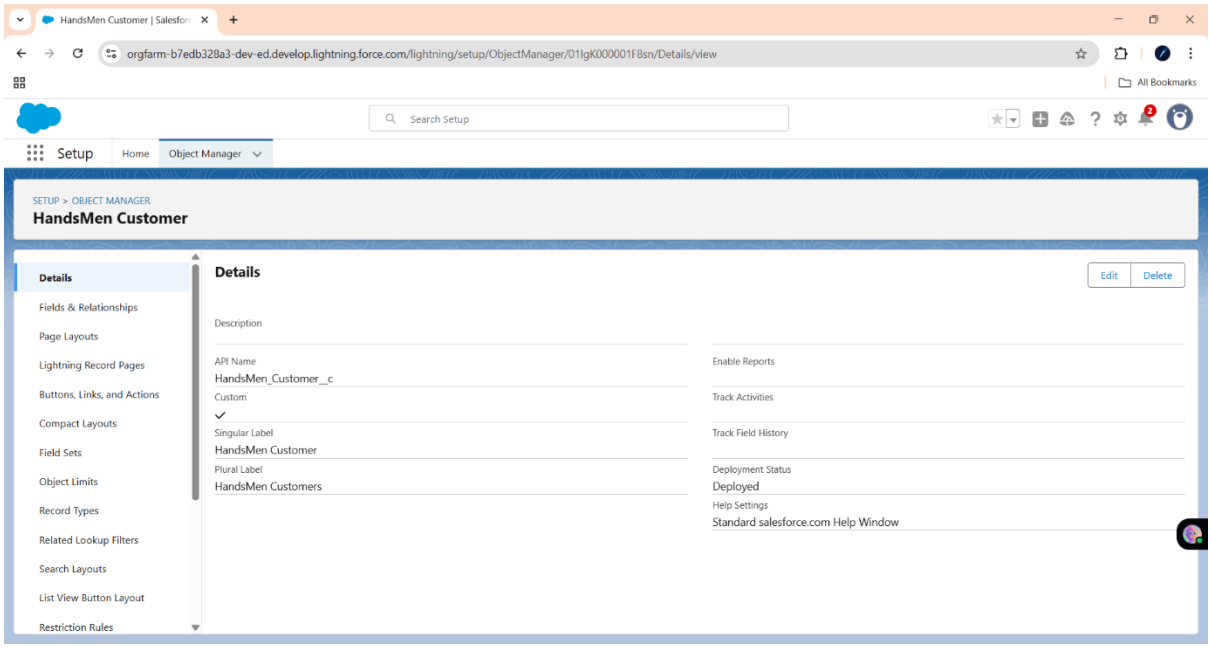
Define the following fields and relationships for product object



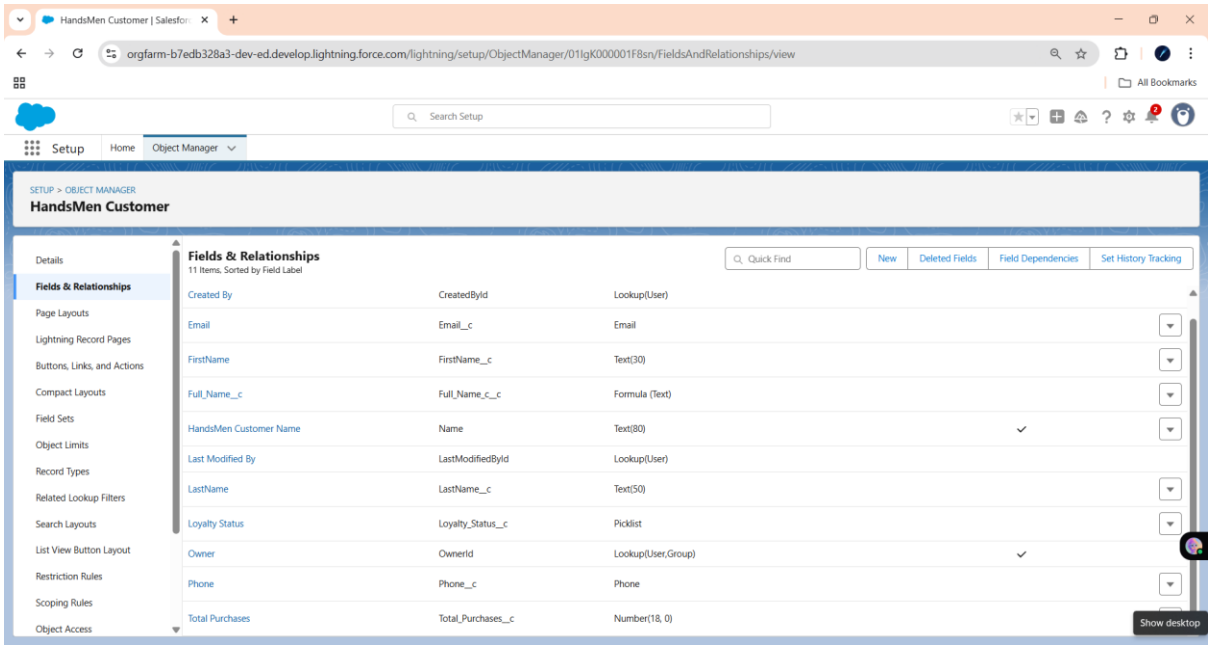
Create the order object accordingly



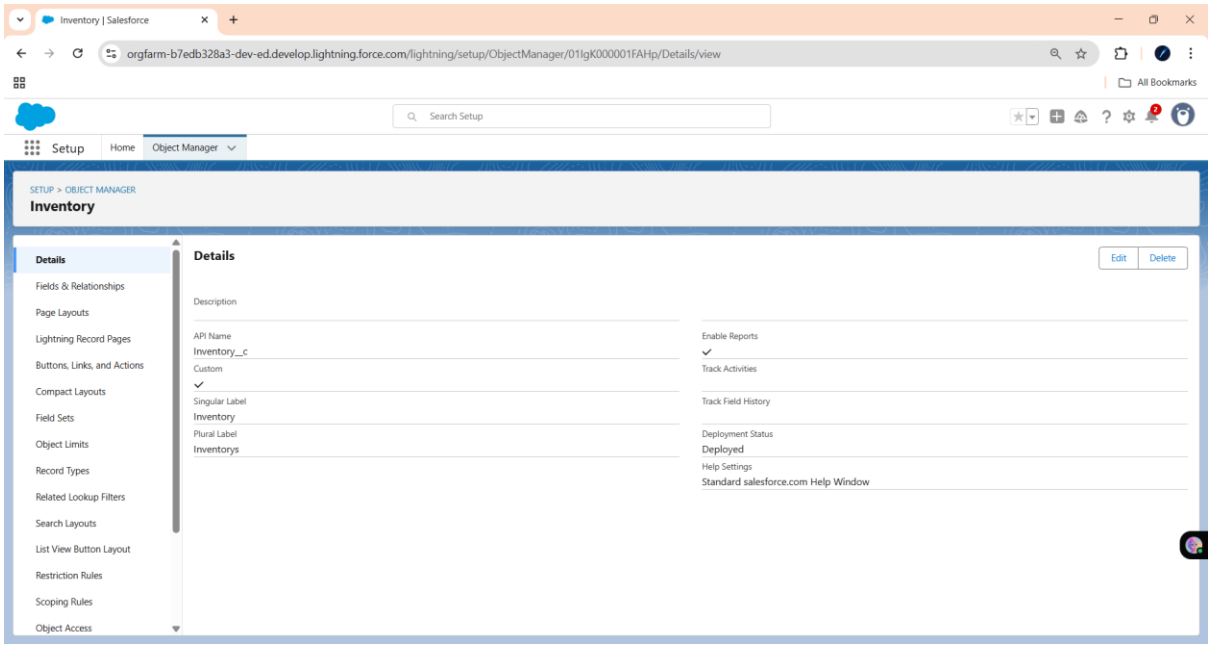
Define the following fields and relationships for order object



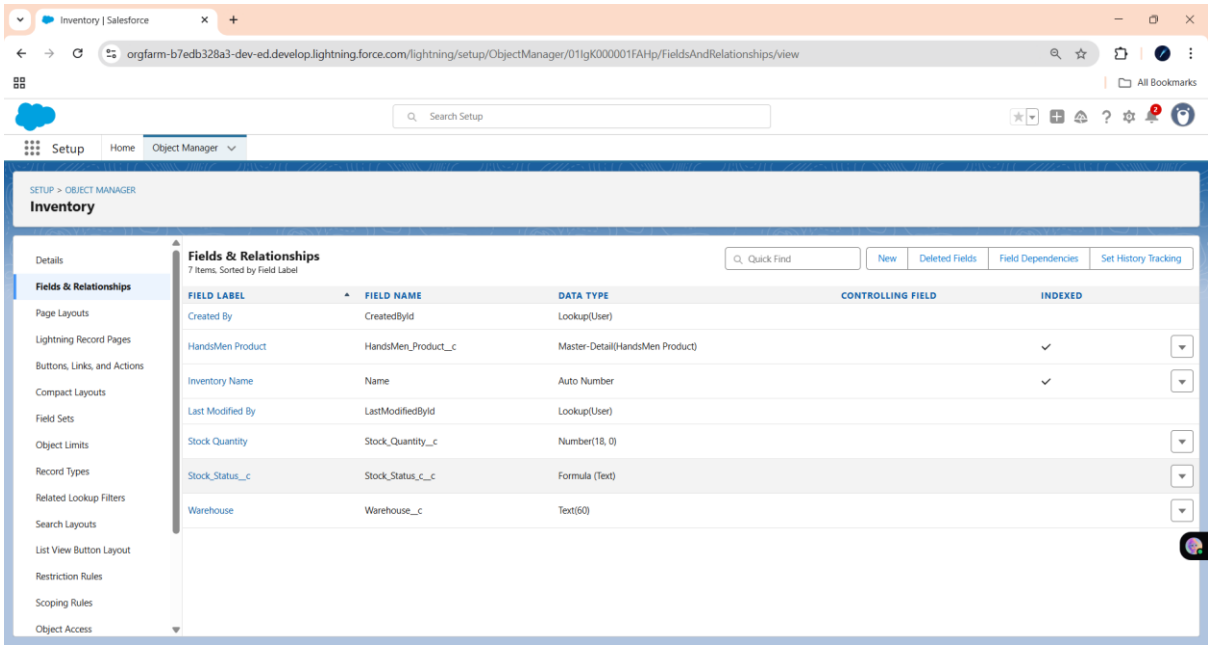
Create the Customer object accordingly



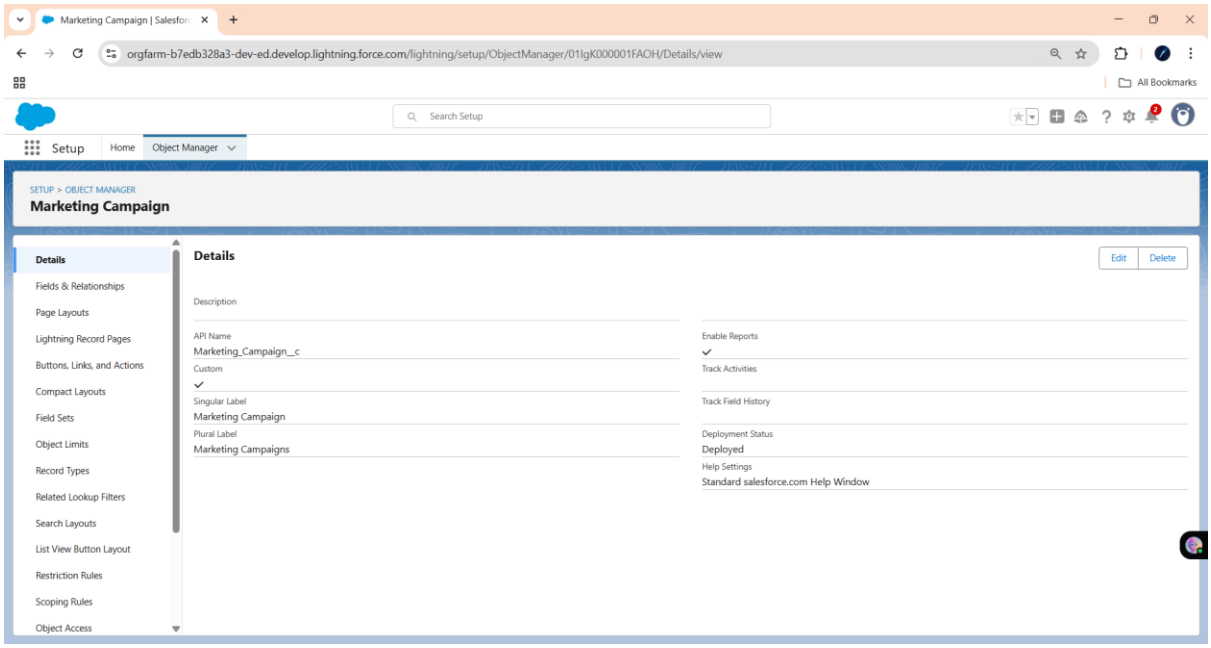
Define the following fields and relationships for the customer object



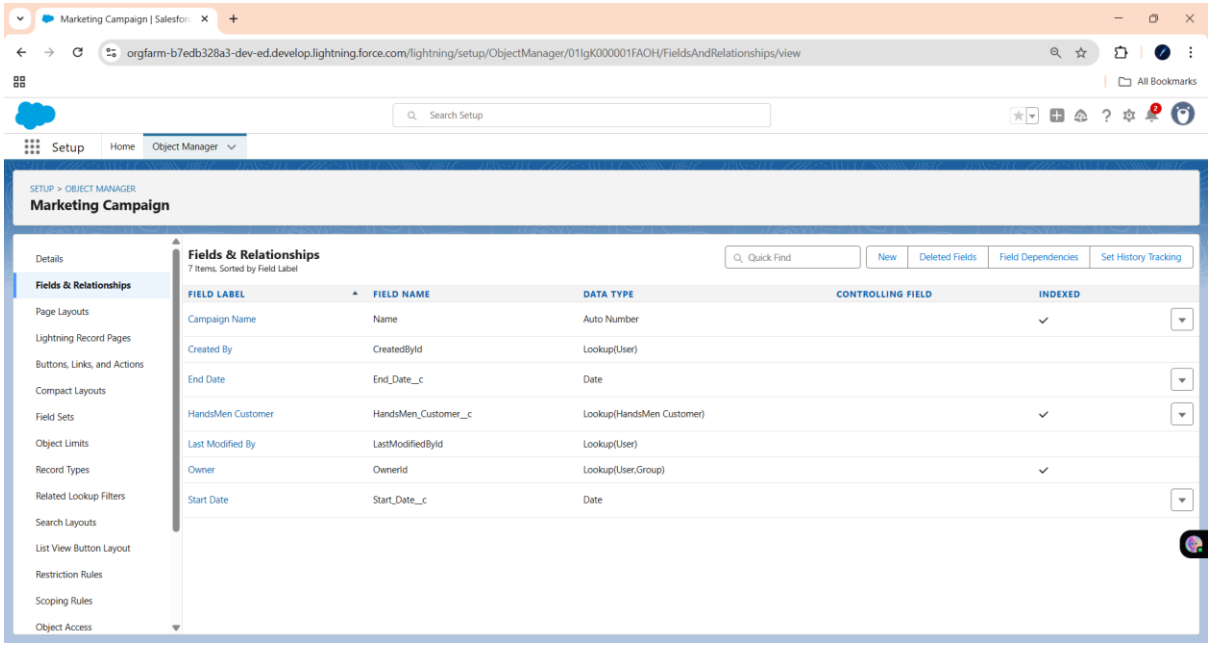
Create the inventory object accordingly



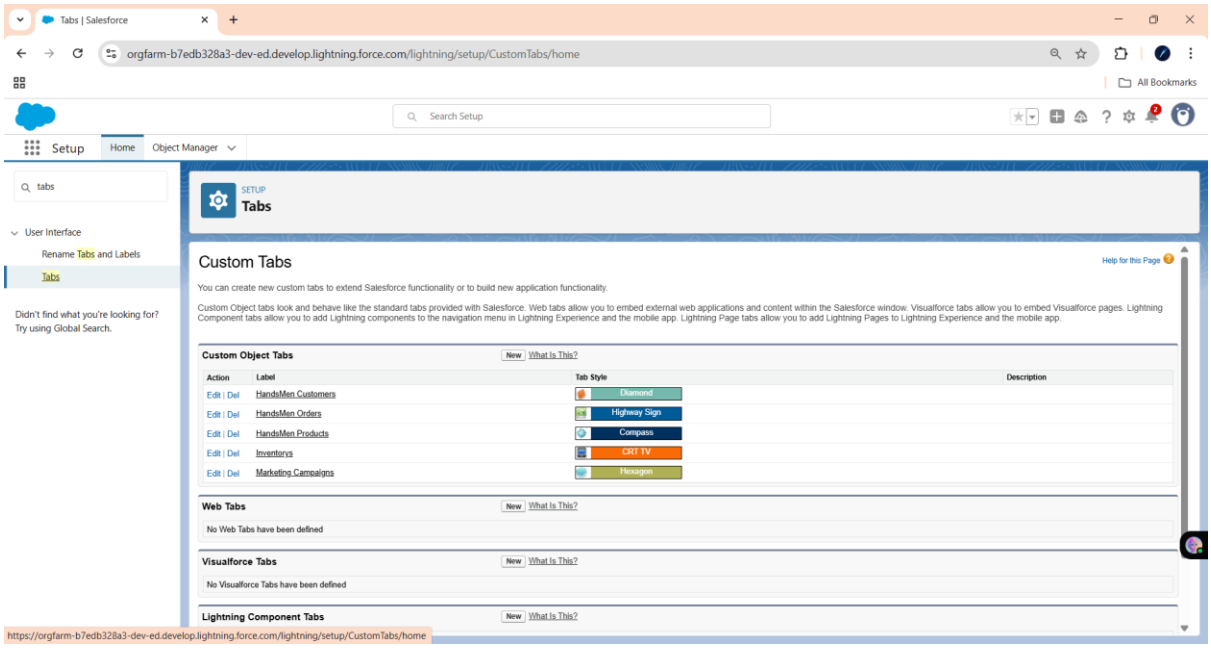
Define the following fields and relationships for inventory object



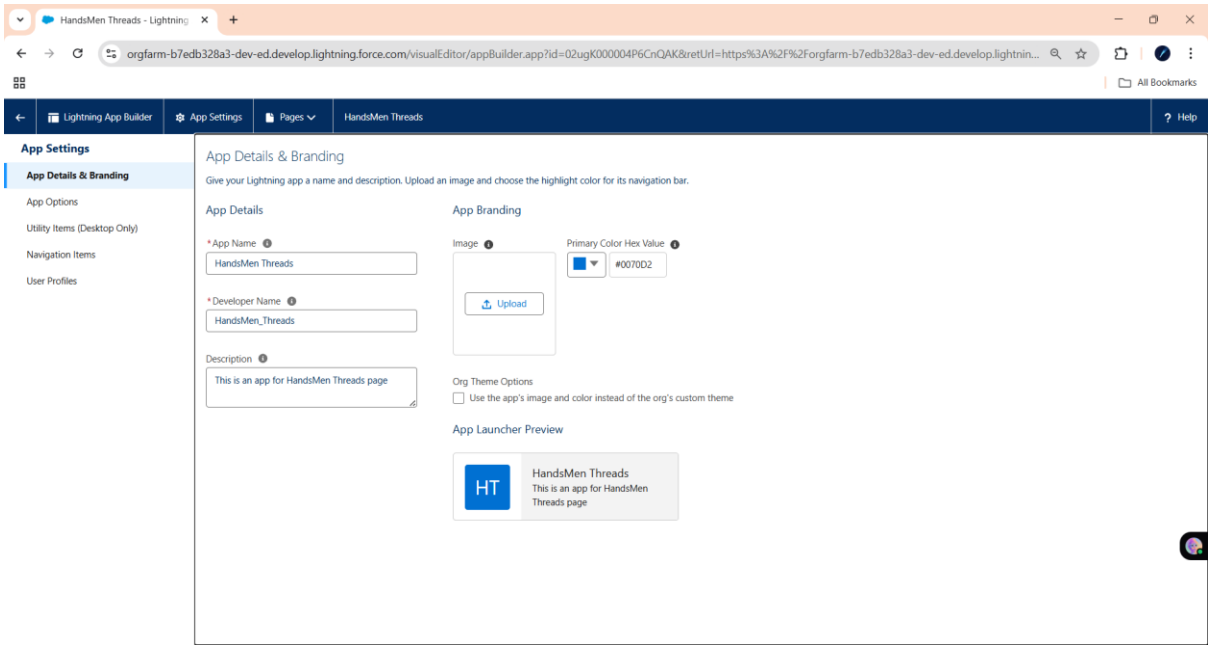
Create the marketing campaign object accordingly



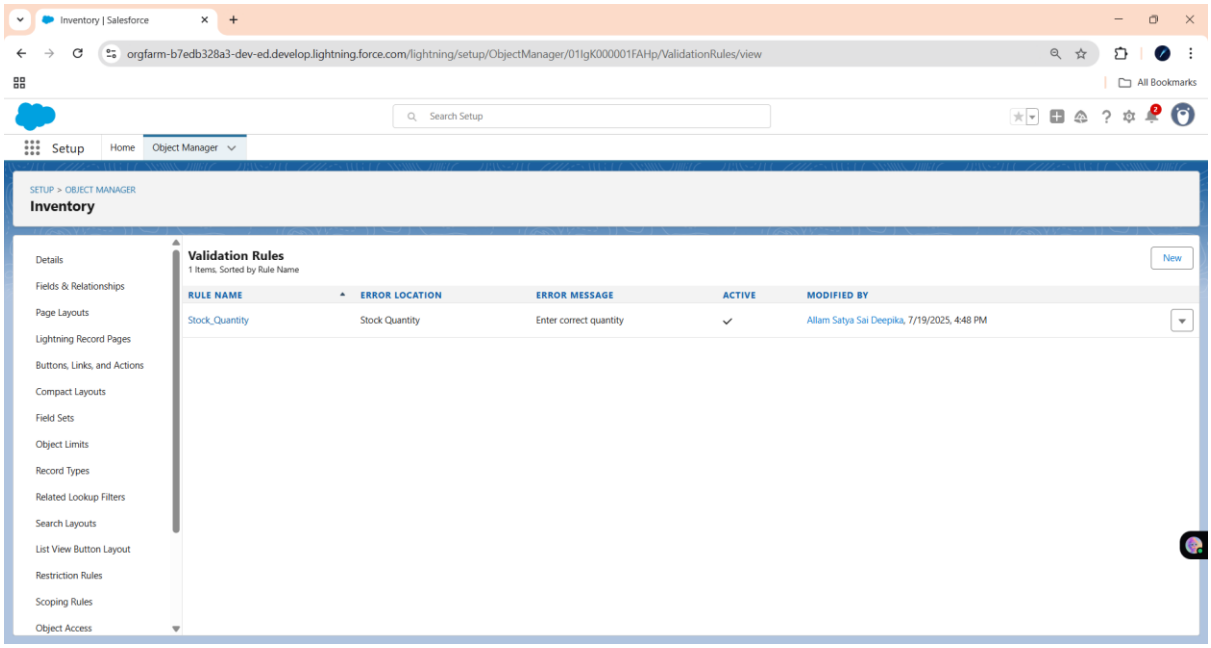
Define the following fields and relationships for the marketing campaign object



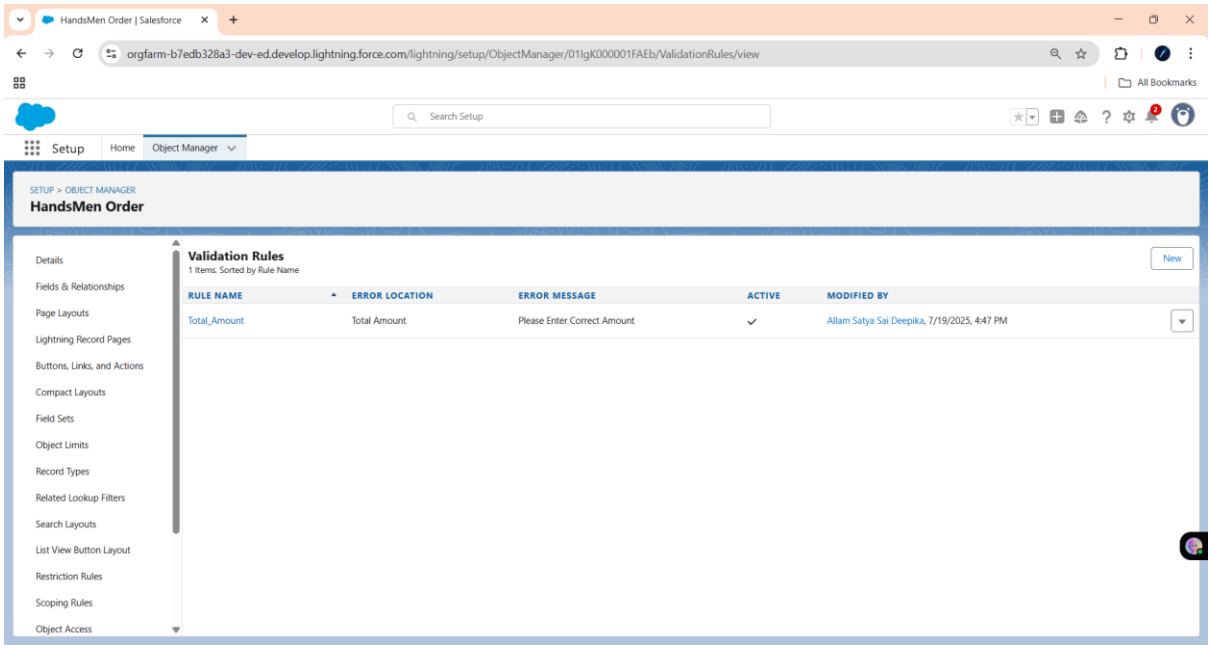
Create custom tabs for all the objects created above



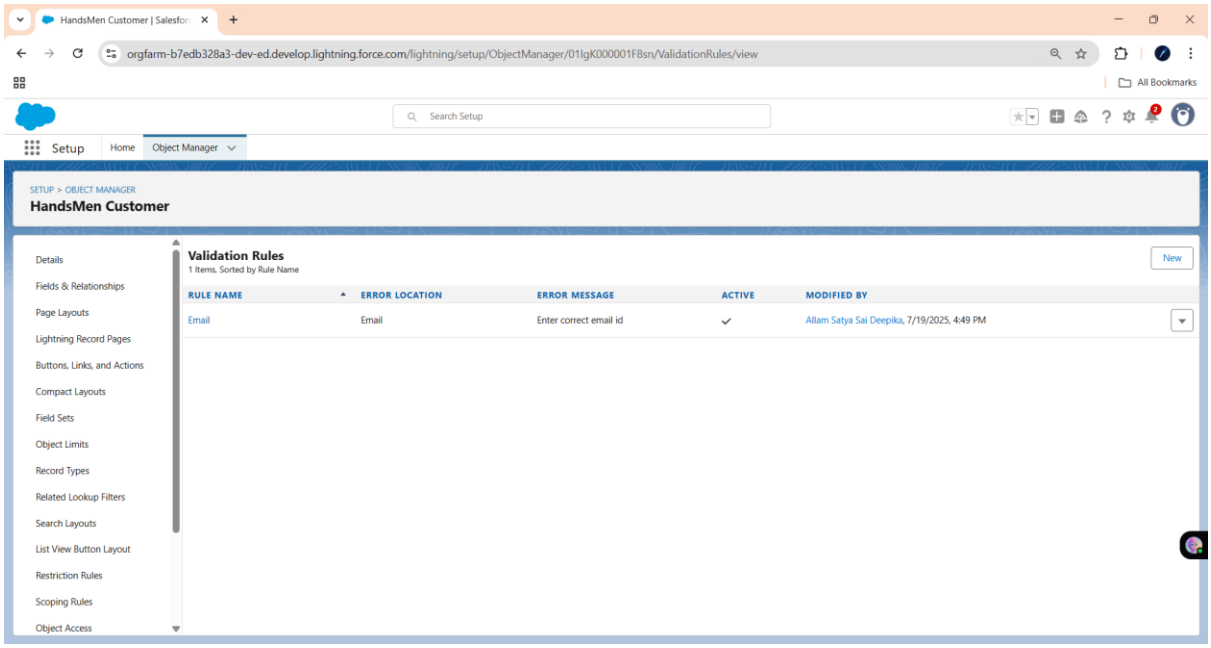
Create a lightning page app named HandsMen Threads in the app manager



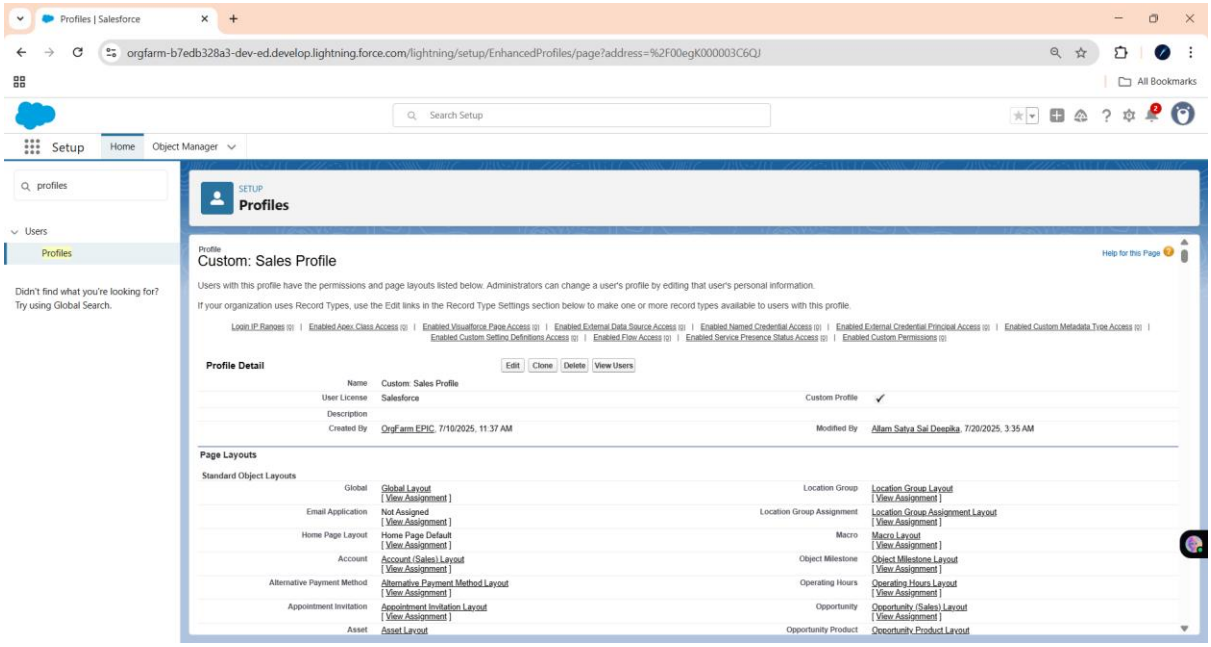
Create a validation rule for the inventory object



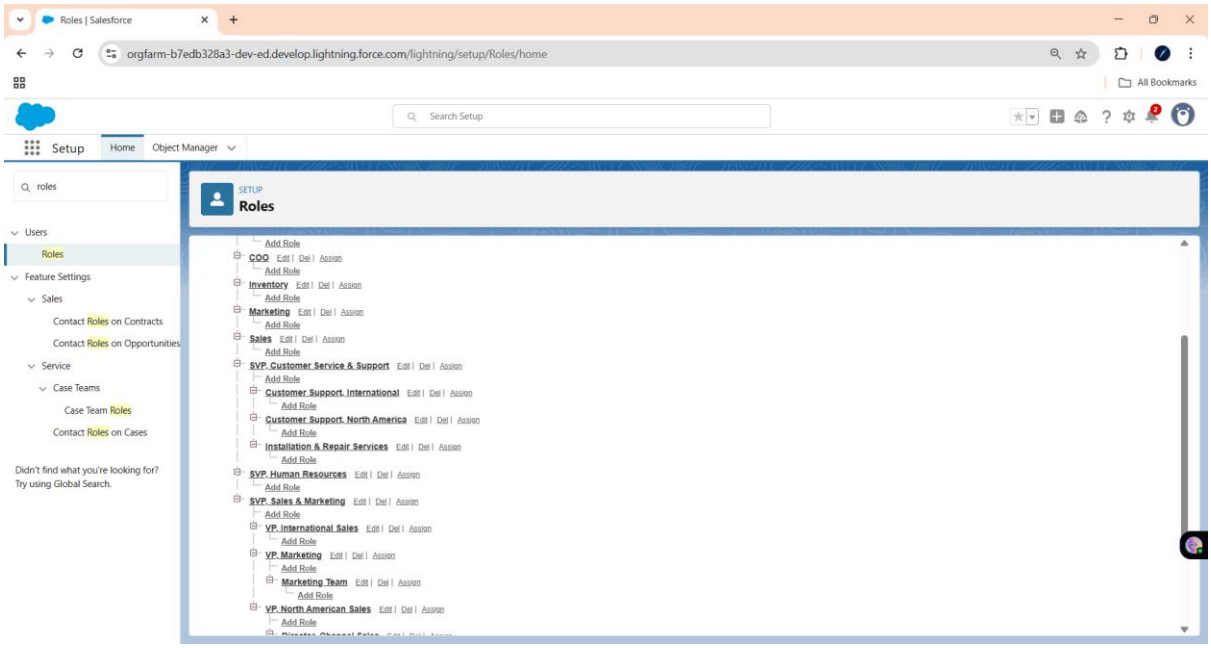
Create a validation rule for the order object



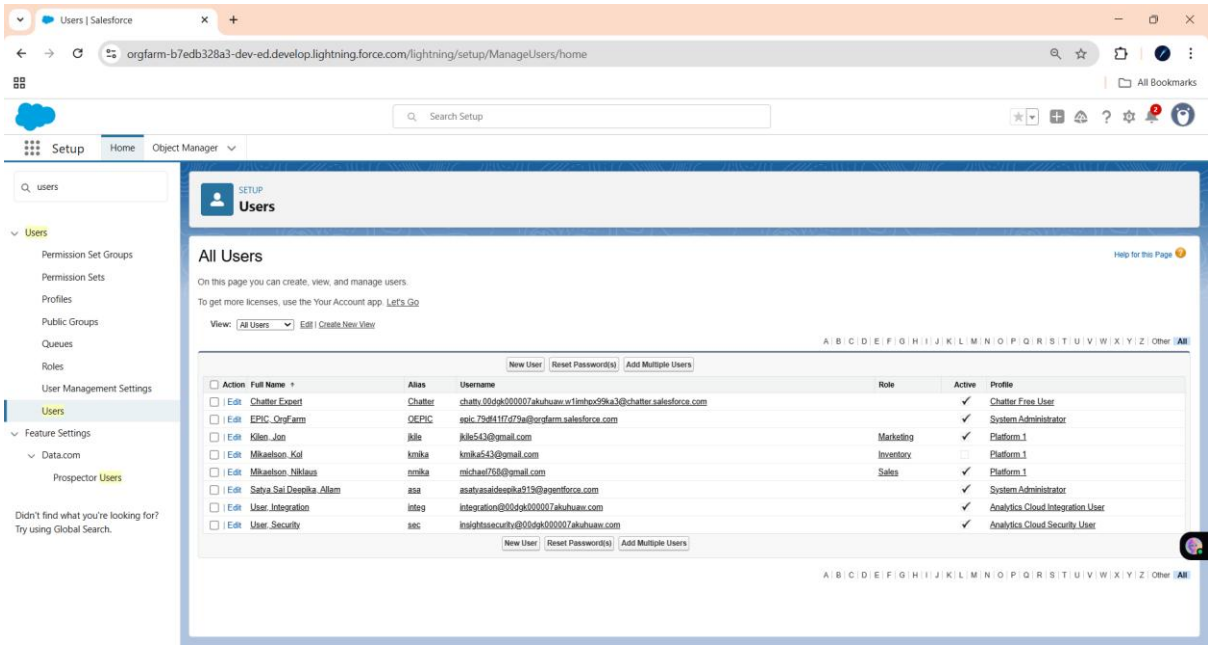
Create a validation rule for the customer object



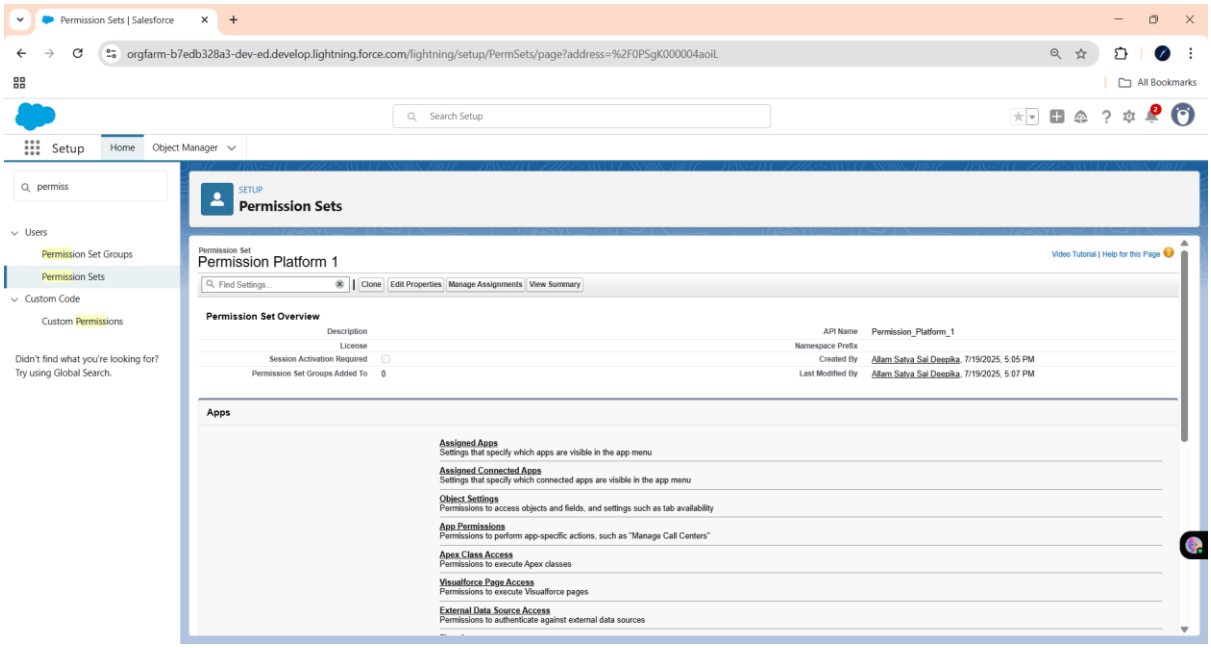
Create custom sales profile by cloning the standard profile



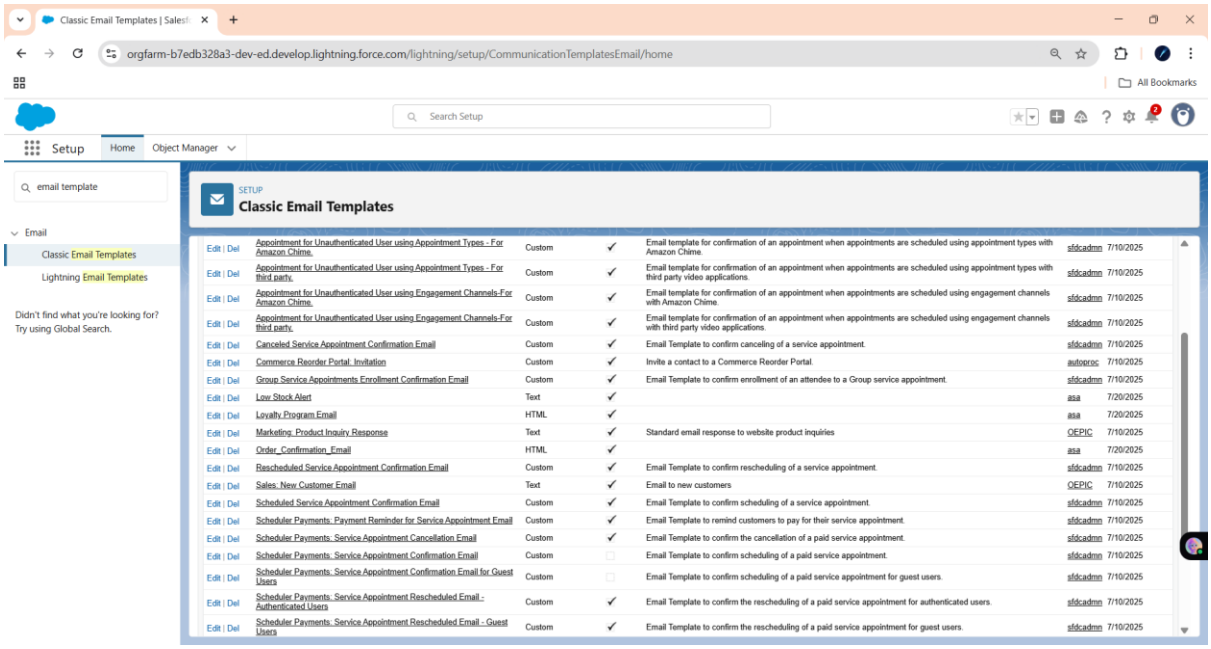
Create three roles: Sales, Inventory, and Marketing



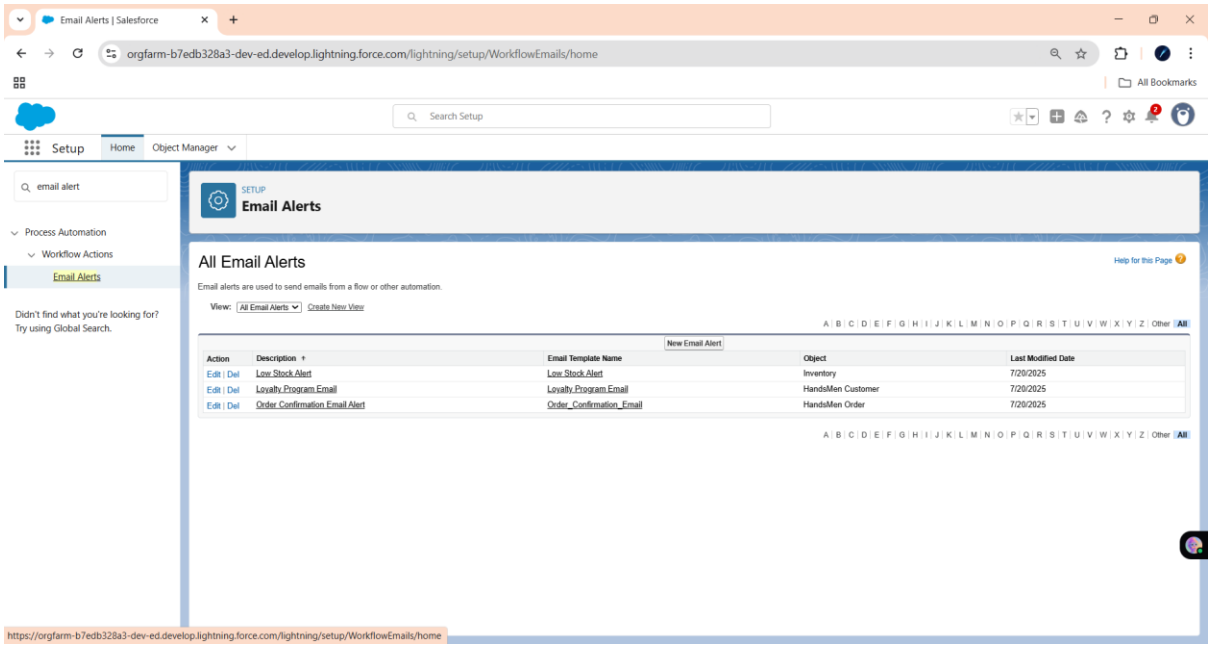
Create users Niklaus, Kol and 2 more of your choice



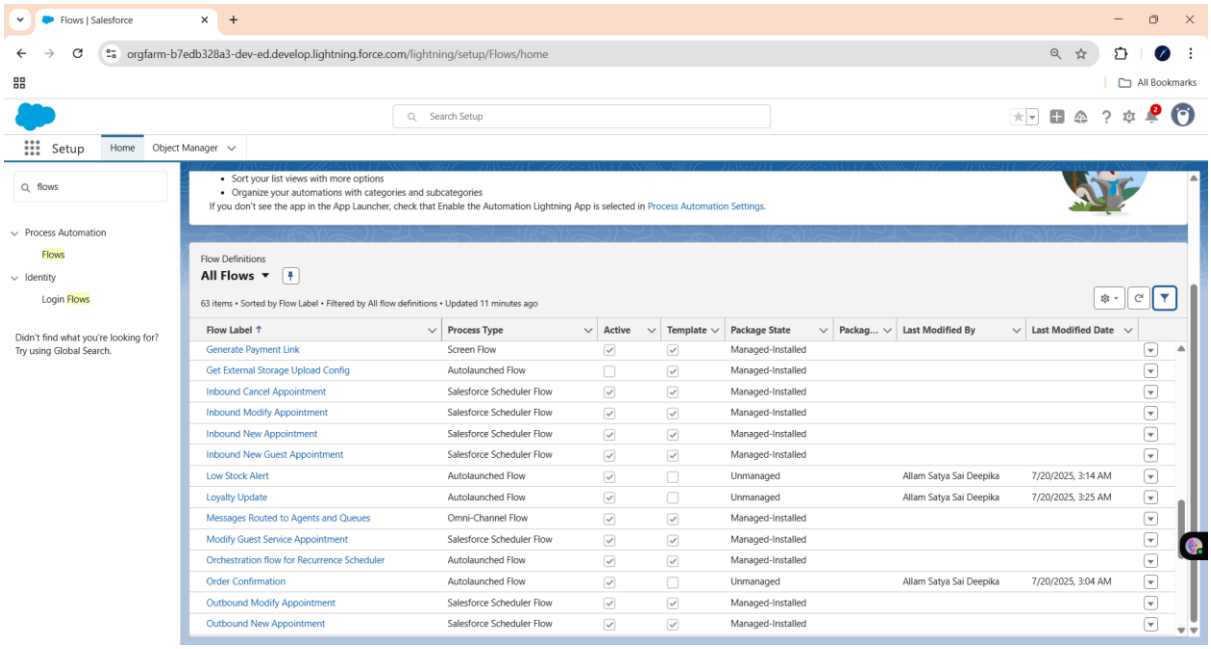
Create permission set Permission Platform 1



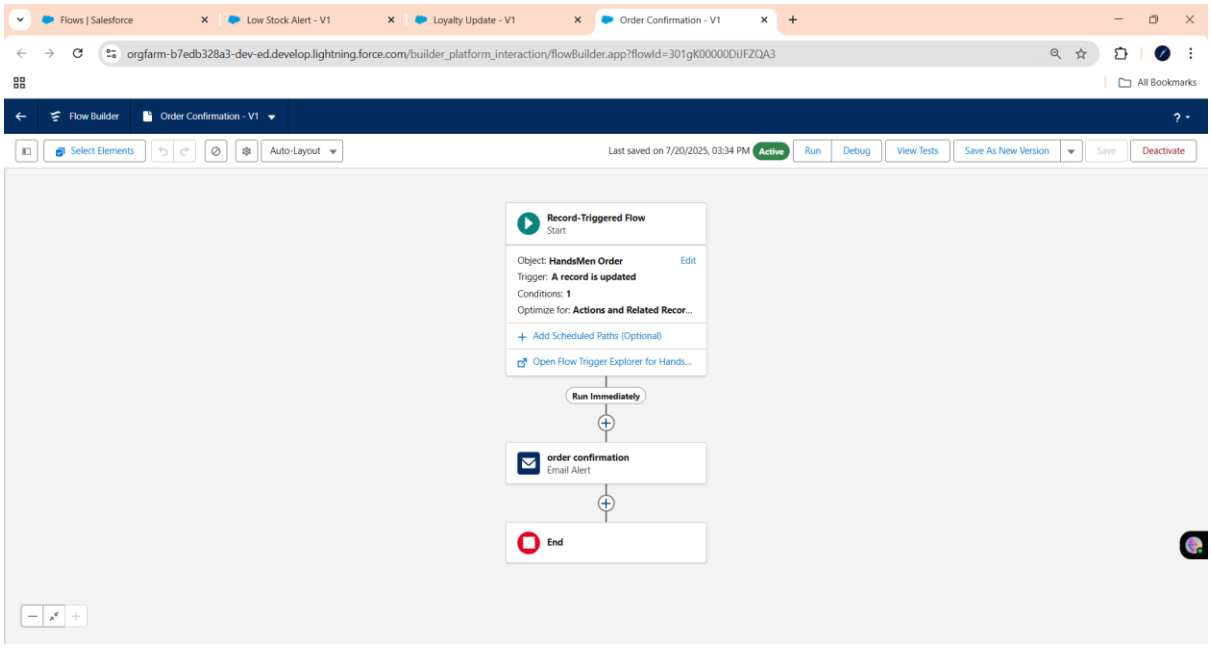
Create Email Templates with the names “**Order Confirmation**”, "**Low Stock Alert**", and "**Loyalty Program Email**" as mentioned in the Email Template description.



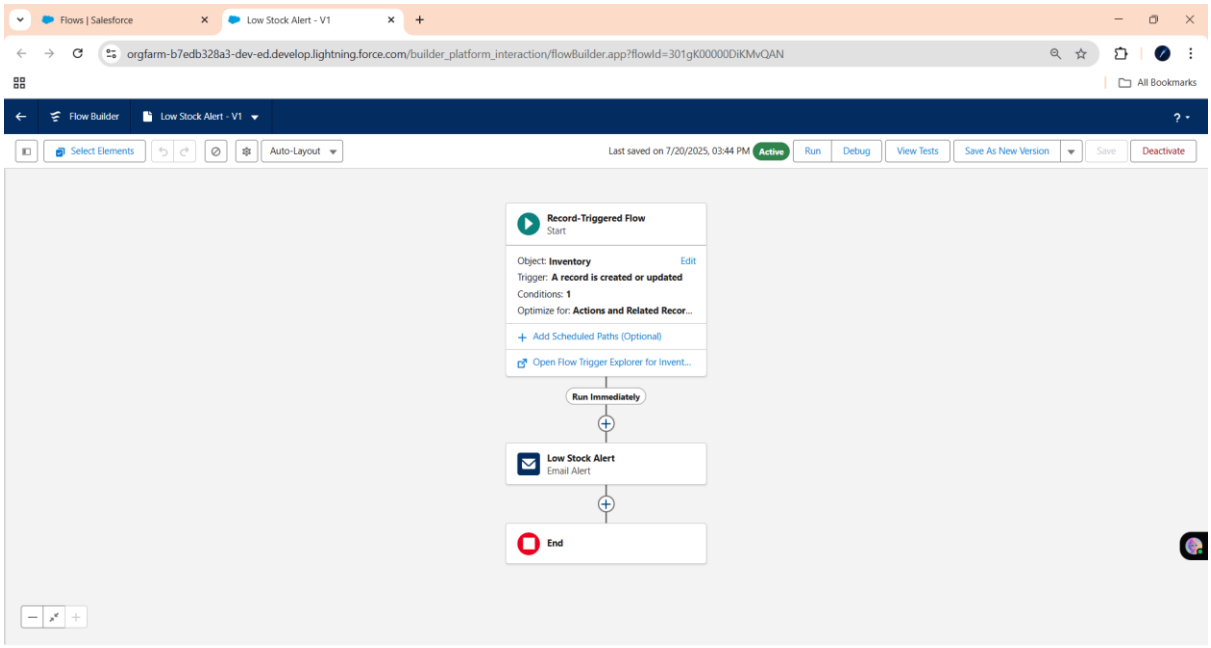
Create the following email alerts as shown in the image



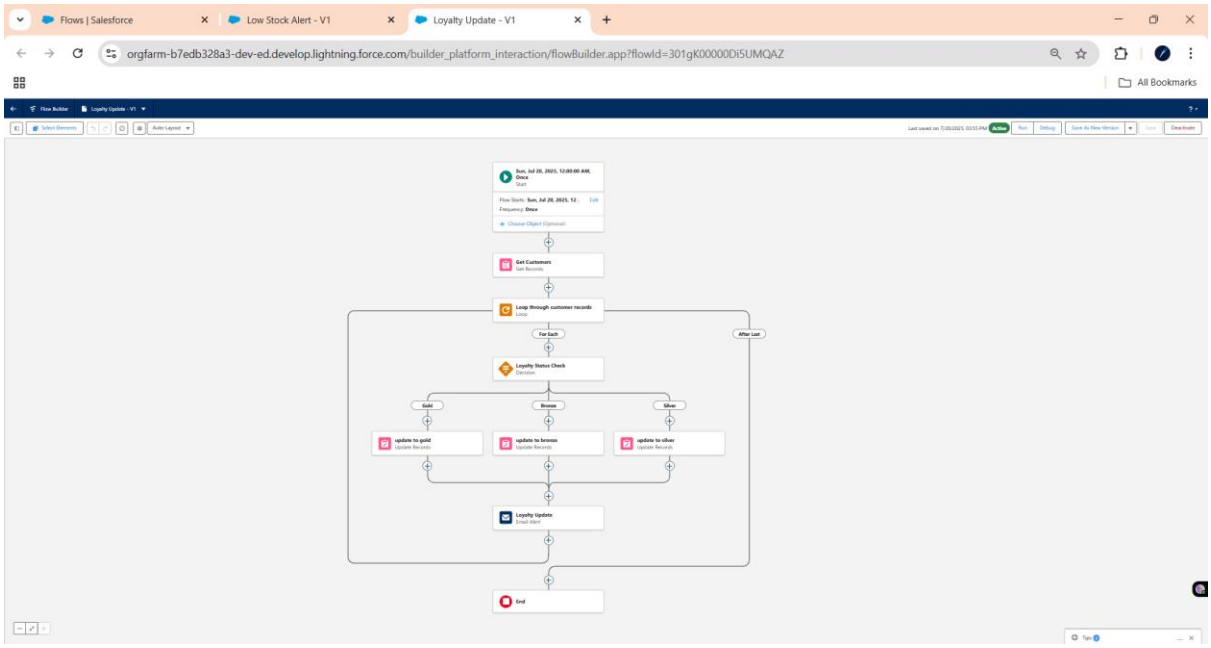
Create 3 flows



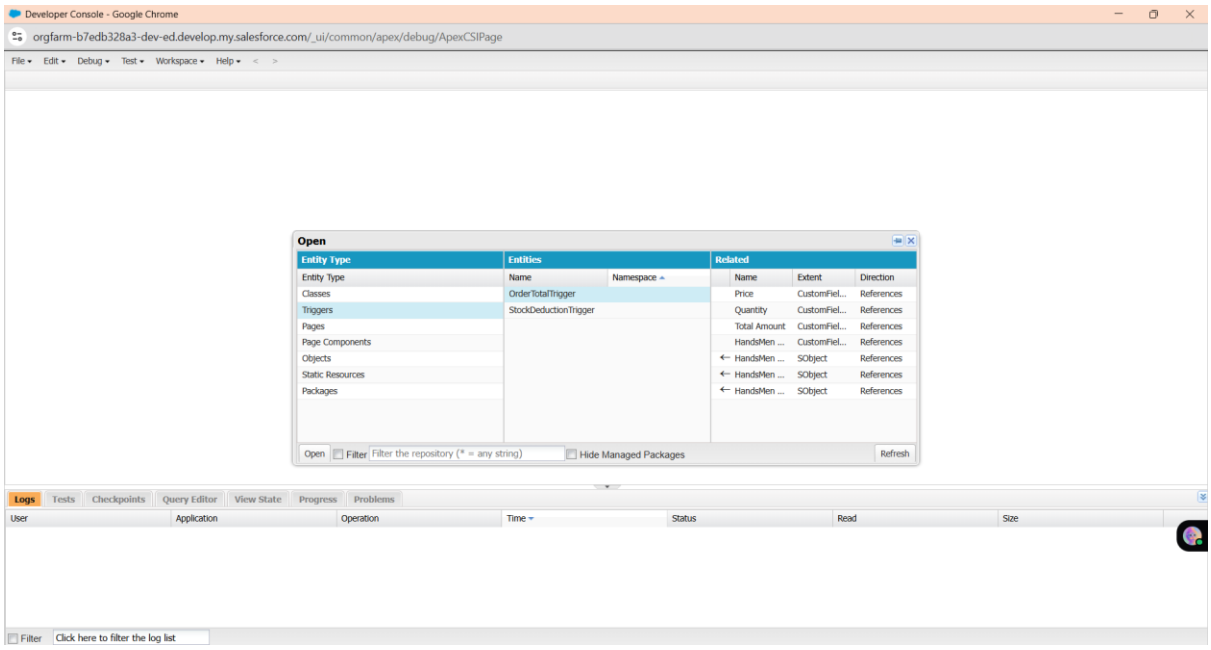
Create Order Triggered flow

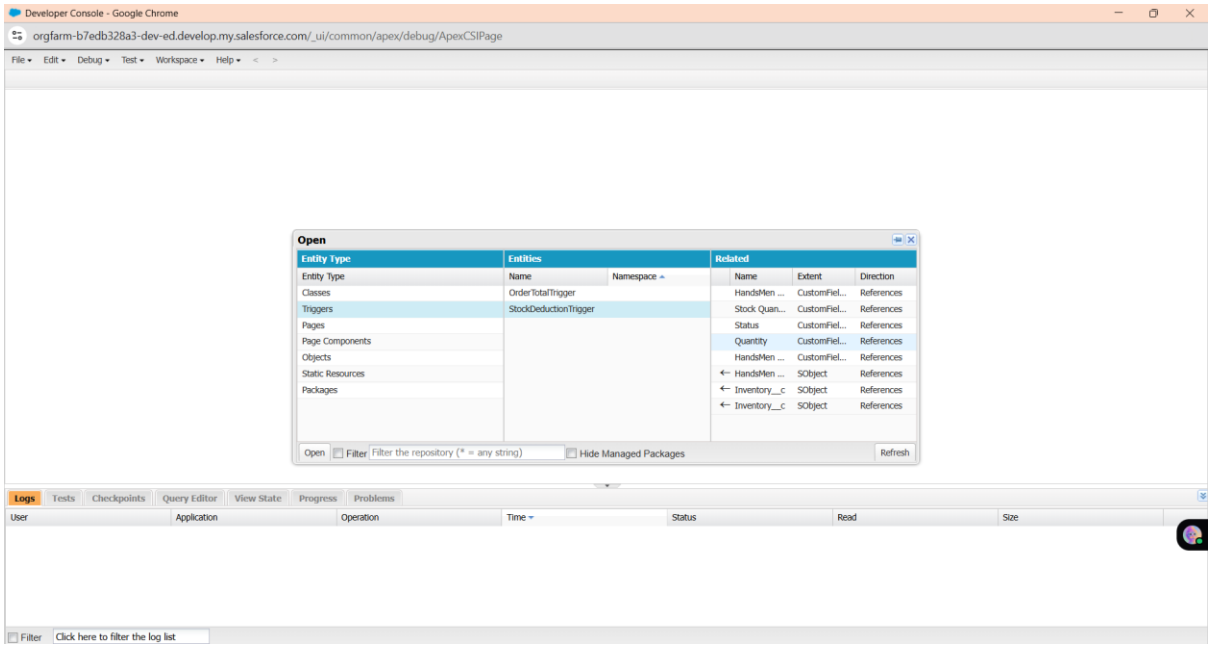


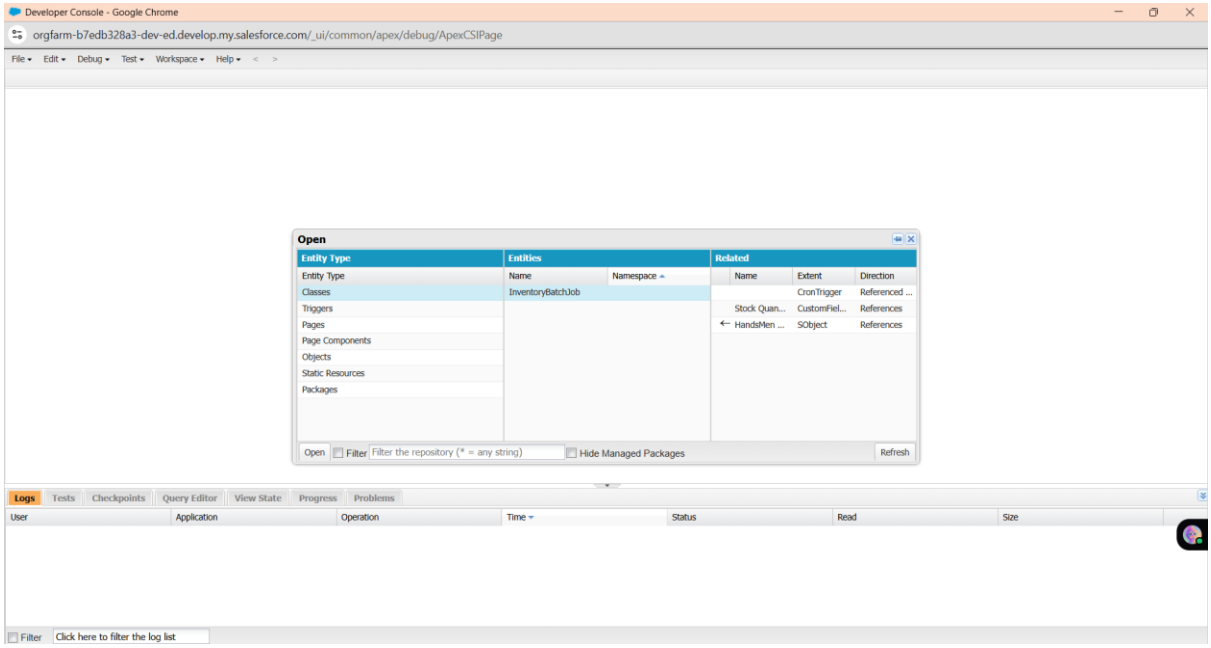
Create a Record- Triggered flow



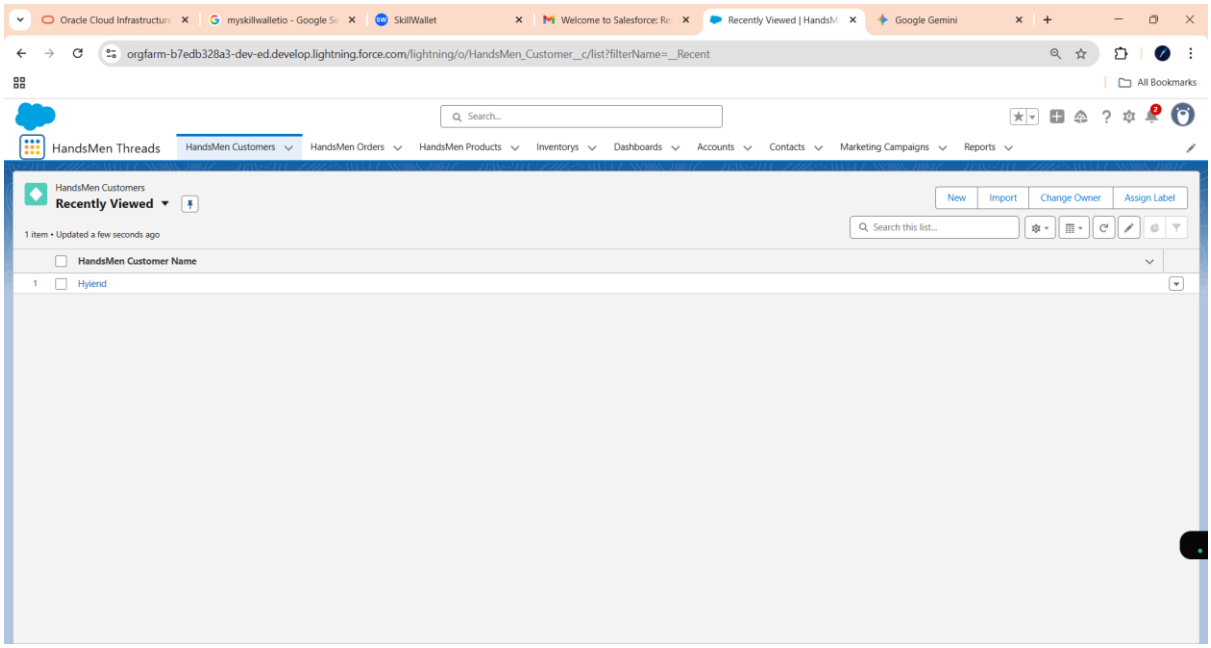
Create a Scheduled Flow



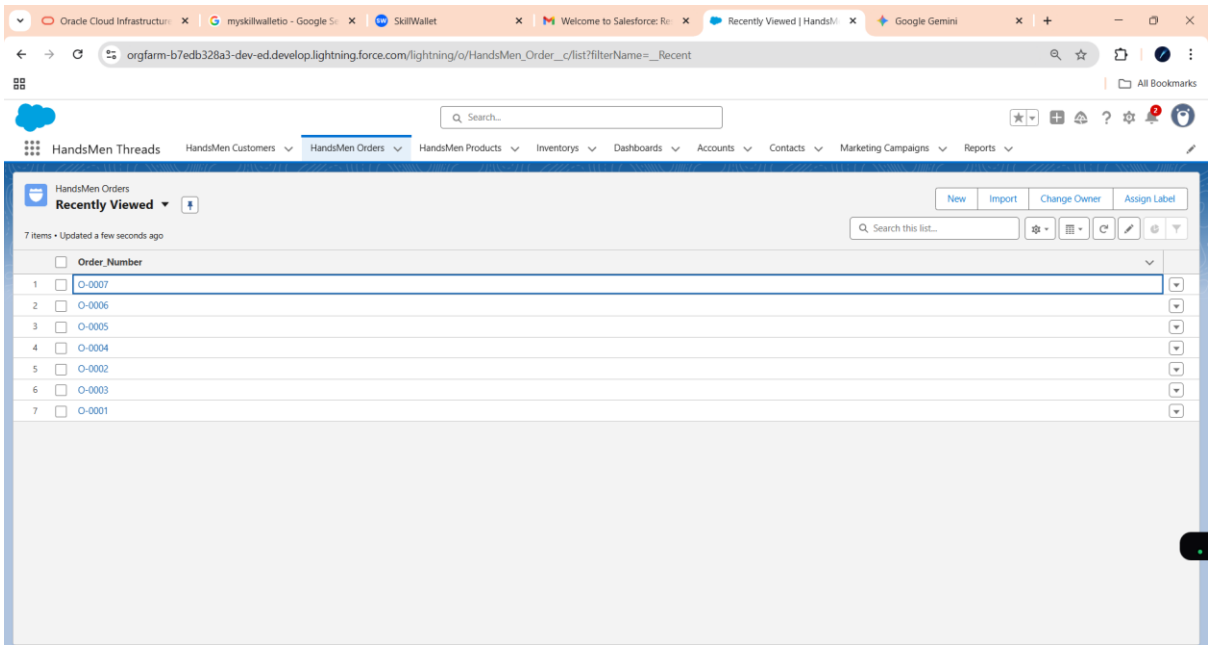




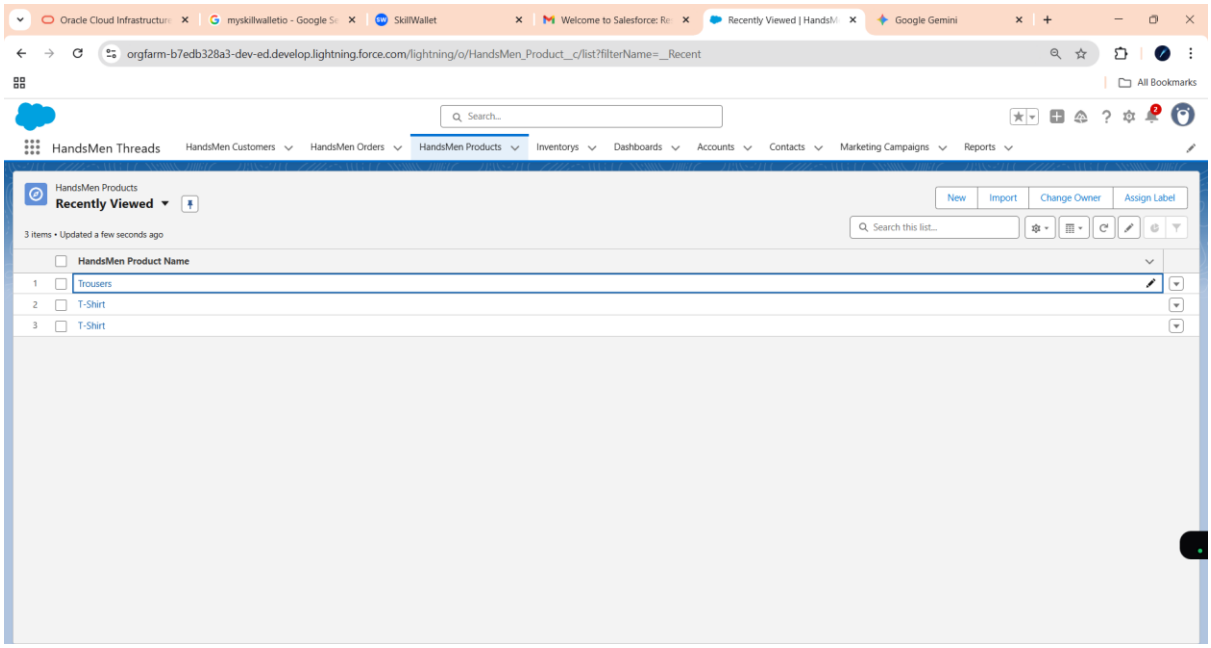
Create the following apex classes, batchjobs, and triggers accordingly



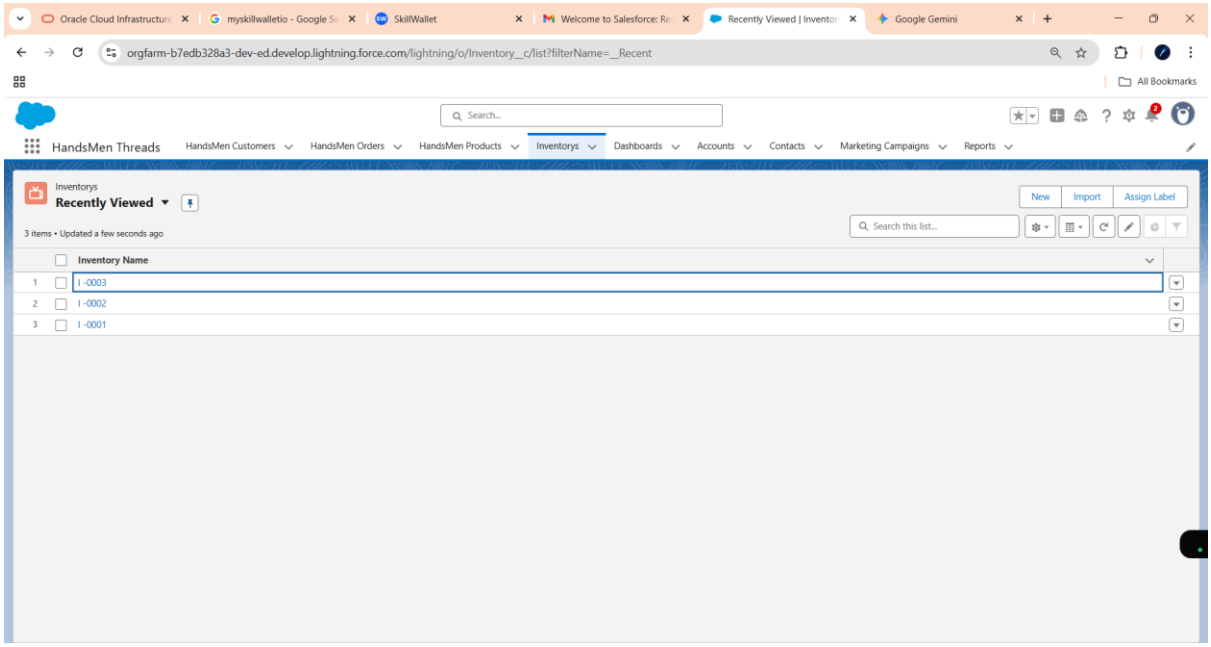
The customer will be shown here



The orders placed will be shown here



The products will be shown here



The inventories list will be visible here

Conclusion:

This Salesforce project is poised to be a transformative initiative for HandsMen Threads. By meticulously designing a robust data model, enforcing data quality from the user interface, and integrating key automated

processes like order confirmations, dynamic loyalty updates, proactive stock alerts, and scheduled bulk order processing, the organization will achieve unparalleled efficiency and significantly enhance its customer relationships.

Leveraging Salesforce technologies such as Lightning App Builder, Record Triggered Flows, Apex, and Asynchronous Apex, HandsMen Threads will establish a single, reliable source of truth for their business data. This will not only streamline operations and reduce manual effort but also empower them with accurate, real-time insights for informed decision-making. Ultimately, this project will position HandsMen Threads for sustainable growth, improved customer satisfaction, and a stronger competitive edge in the dynamic fashion industry.