HandsMen Threads - Salesforce CRM Implementation

### **Project Overview**

This project involves the development of a custom **Customer Relationship Management (CRM)** system using the Salesforce platform. The CRM was designed to streamline the process of managing client information, booking records, and automate business processes using low-code tools like Flows, Validation Rules, Approval Processes, and Reports. Key features include contact management, service bookings, lead status tracking, automated tasks, and analytical dashboards to provide business insights.

### **Objectives**

* Automate the booking and customer engagement process.
* Improve client information tracking through custom objects and relationships.
* Enable task automation using Flows and Process Builder.
* Generate visual reports and dashboards for performance review.
* Establish user access controls and maintain security compliance.

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

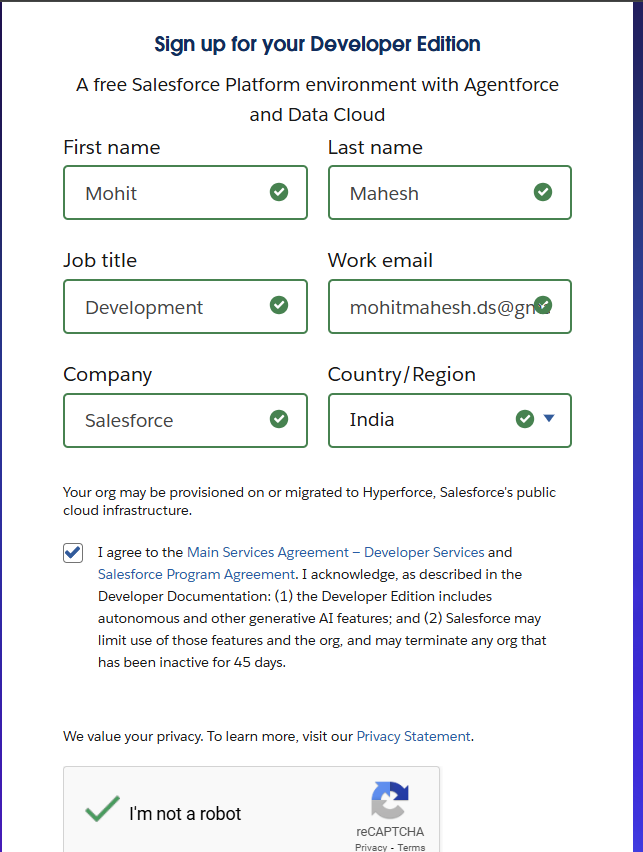
**Understanding Business Requirements:** - The CRM addresses the needs of service-based businesses where regular bookings and customer follow-ups are vital. - Requirement was to streamline bookings, manage customer profiles, and assign tasks to employees.

**Defining Project Scope and Objectives:** - Scope includes creation of a Booking System, automated task assignments, lead tracking, and report generation. - Objective is to reduce manual workload and improve response time to customer inquiries.

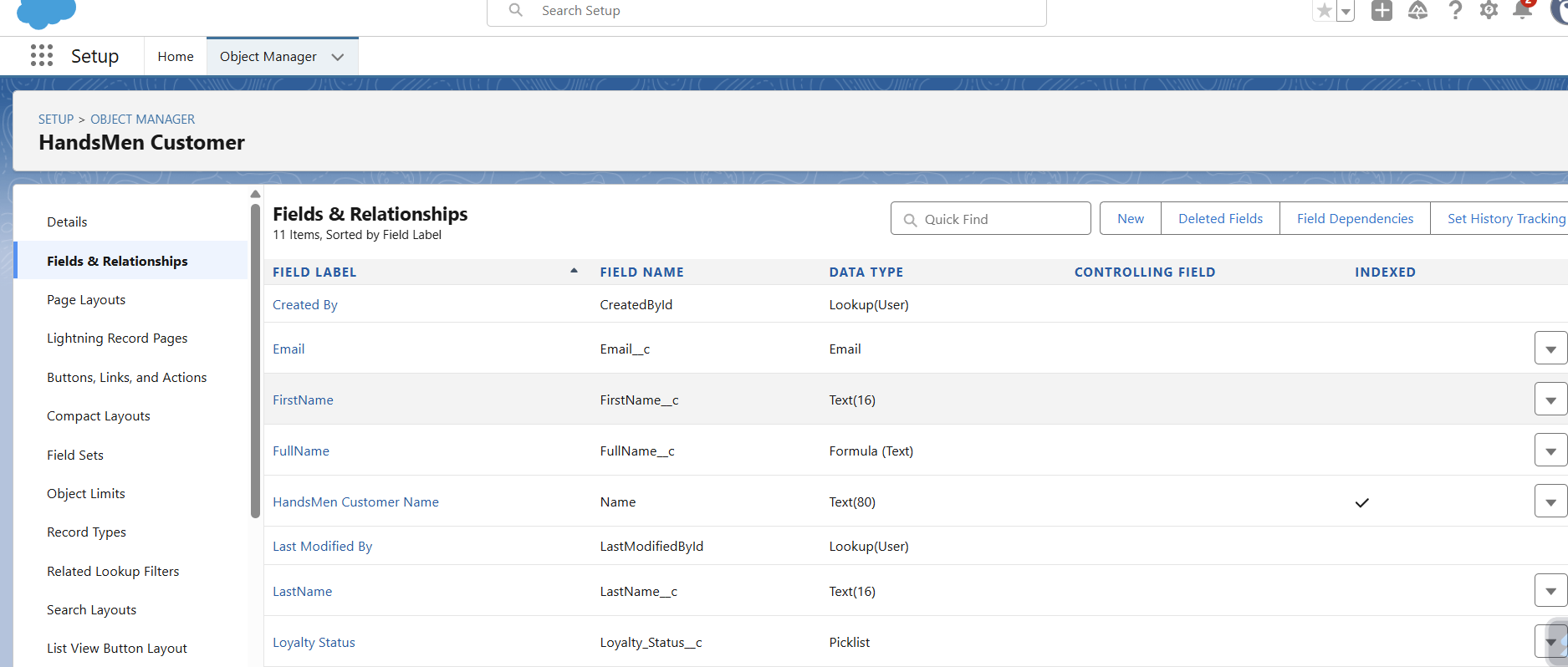
**Design Data Model and Security Model:** - Custom objects: Booking, Feedback, Service. - Relationships established: Master-detail (Service to Booking), Lookup (Feedback to Booking). - Security Model includes Profiles, Permission Sets, and Role Hierarchy.

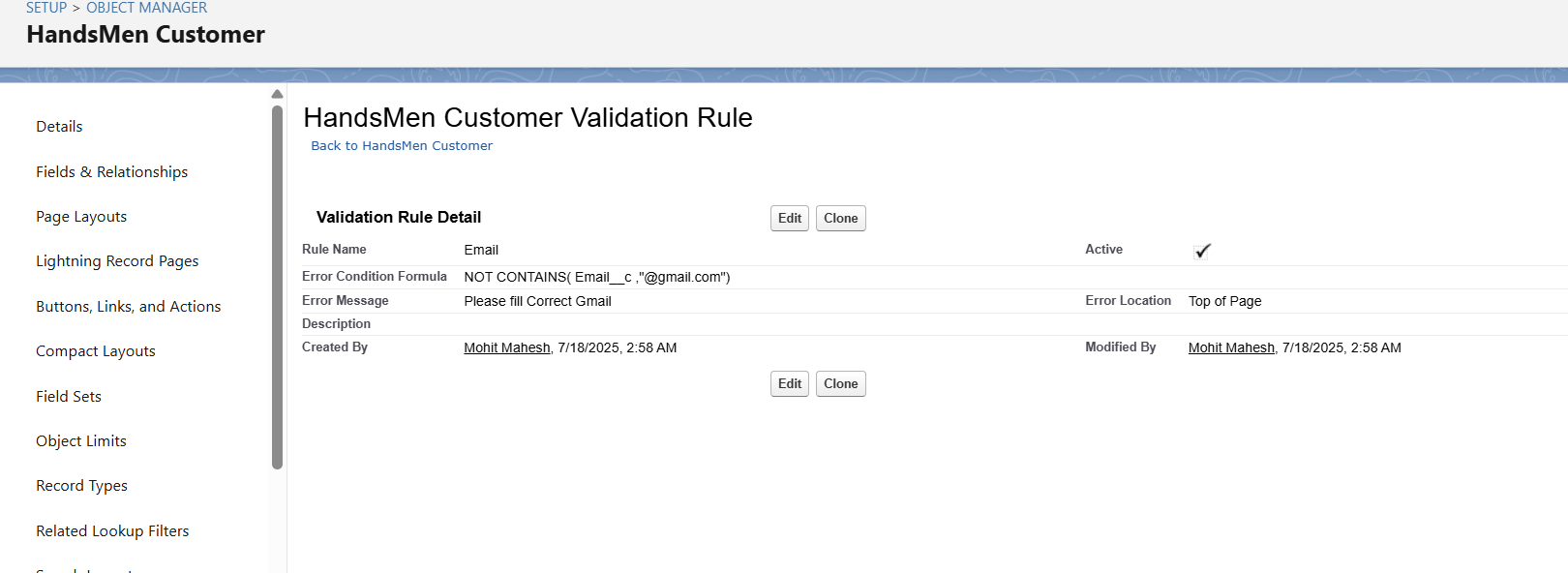
### **Phase 2: Salesforce Development - Backend & Configurations**

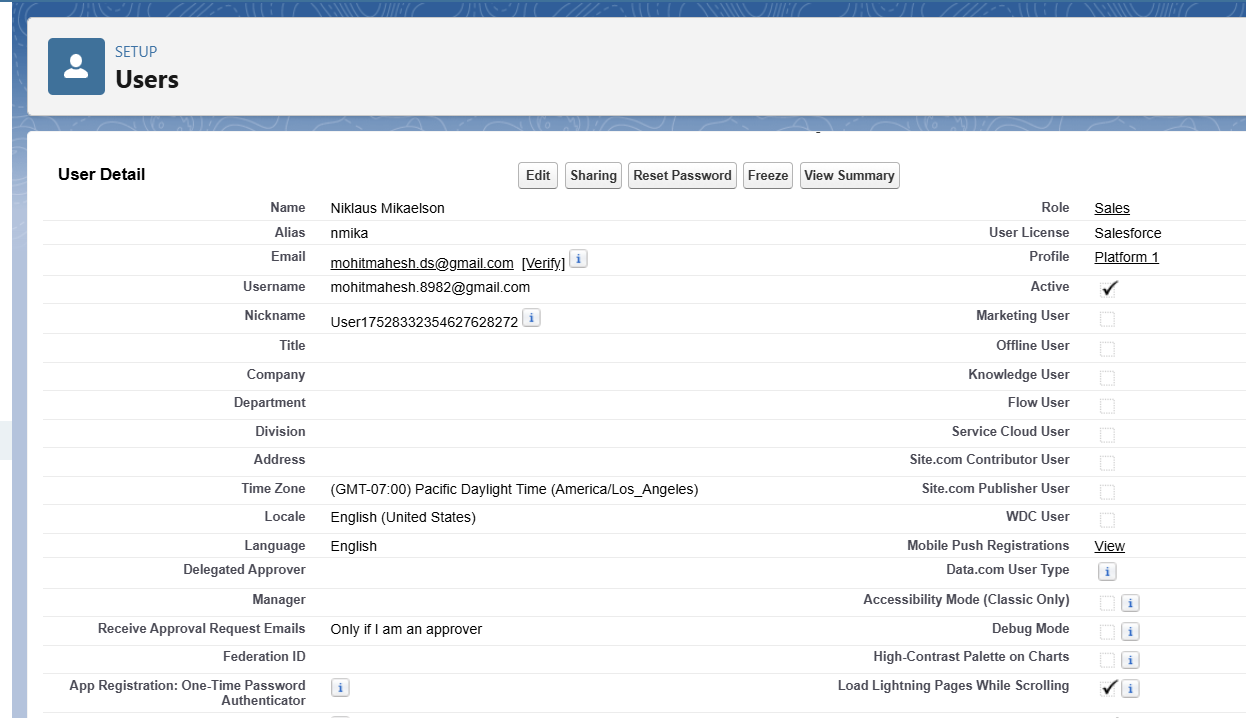
**Setup Environment & DevOps Workflow:** - Developer Edition Org configured. - Changes documented for future deployment via Change Sets.



**Customization & Automation:** - Created Custom Objects: Booking, Service, Feedback. - Implemented Fields, Validation Rules (e.g., date checks, email format). - Process Builder and Flow used to auto-create Tasks upon booking confirmation. - Approval Process used for booking cancellations or VIP status approvals.

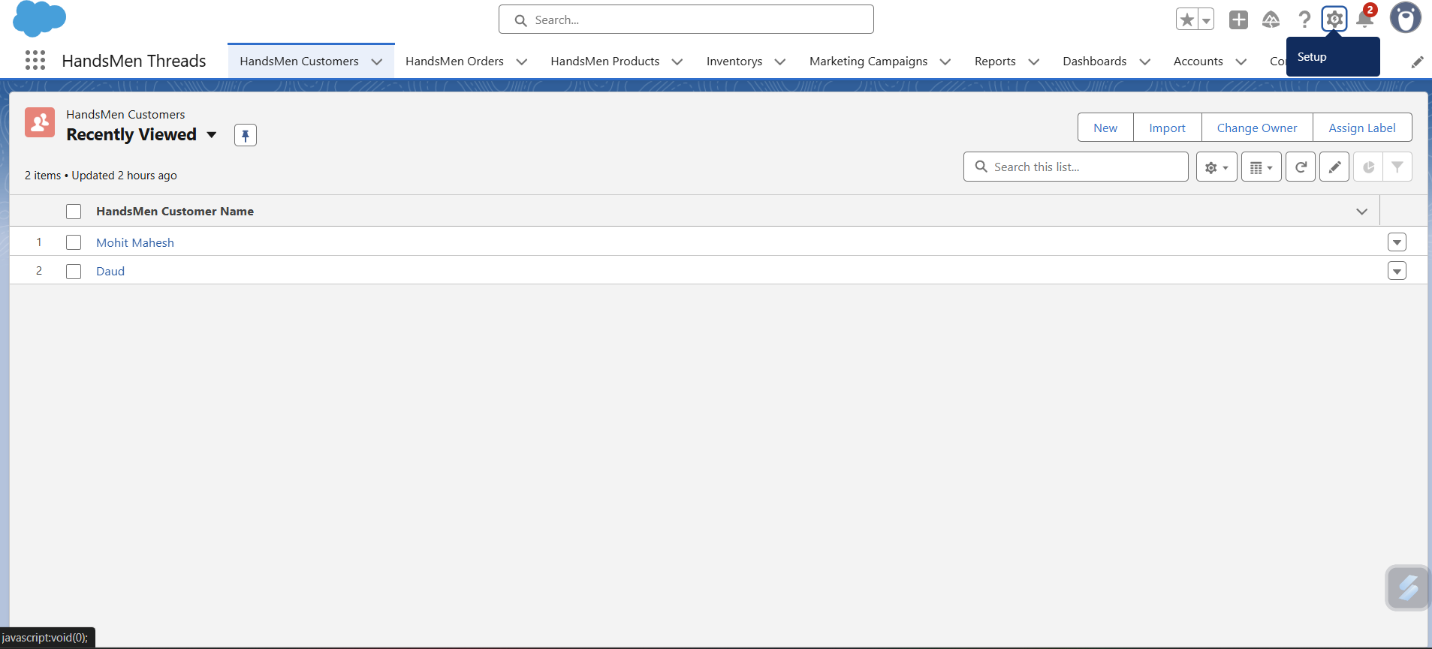






**Apex Development (if any):** - Triggers for field updates and duplicate checks. - Asynchronous Apex used for bulk email reminders.

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

* App created using Lightning App Builder.
* Page Layouts configured with Dynamic Forms.
* Tabs organized for intuitive navigation.
* Profiles for Admin, Manager, and Staff roles.
* Reports on Booking by Month, Feedback Ratings.
* Dashboards with KPIs like Booking Count, Revenue Forecast.
* Lightning Pages for Booking and Feedback.
* No LWC development was done for this project.
* 

### 

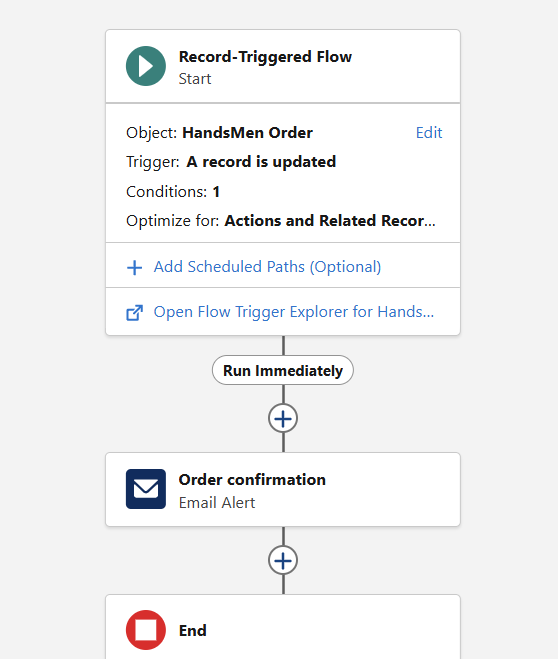
### **Phase 4: Data Migration, Testing & Security**

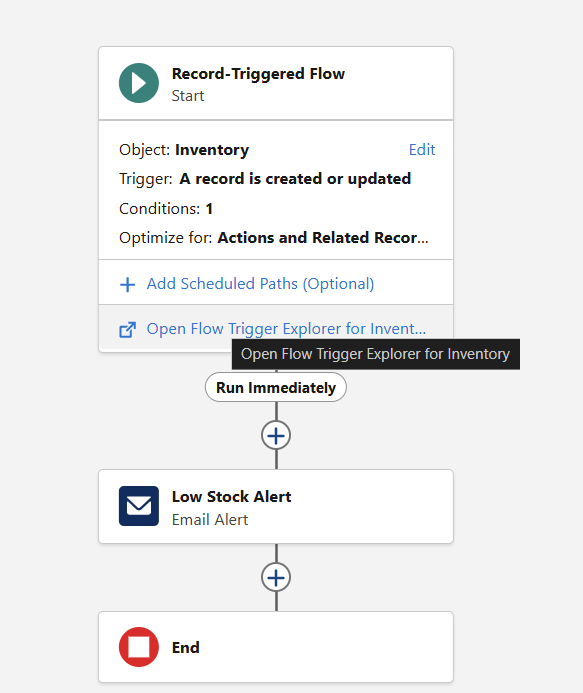
**Data Migration:** - Used Data Import Wizard for initial Service and Customer data. - Ensured proper record assignment via Owner field.

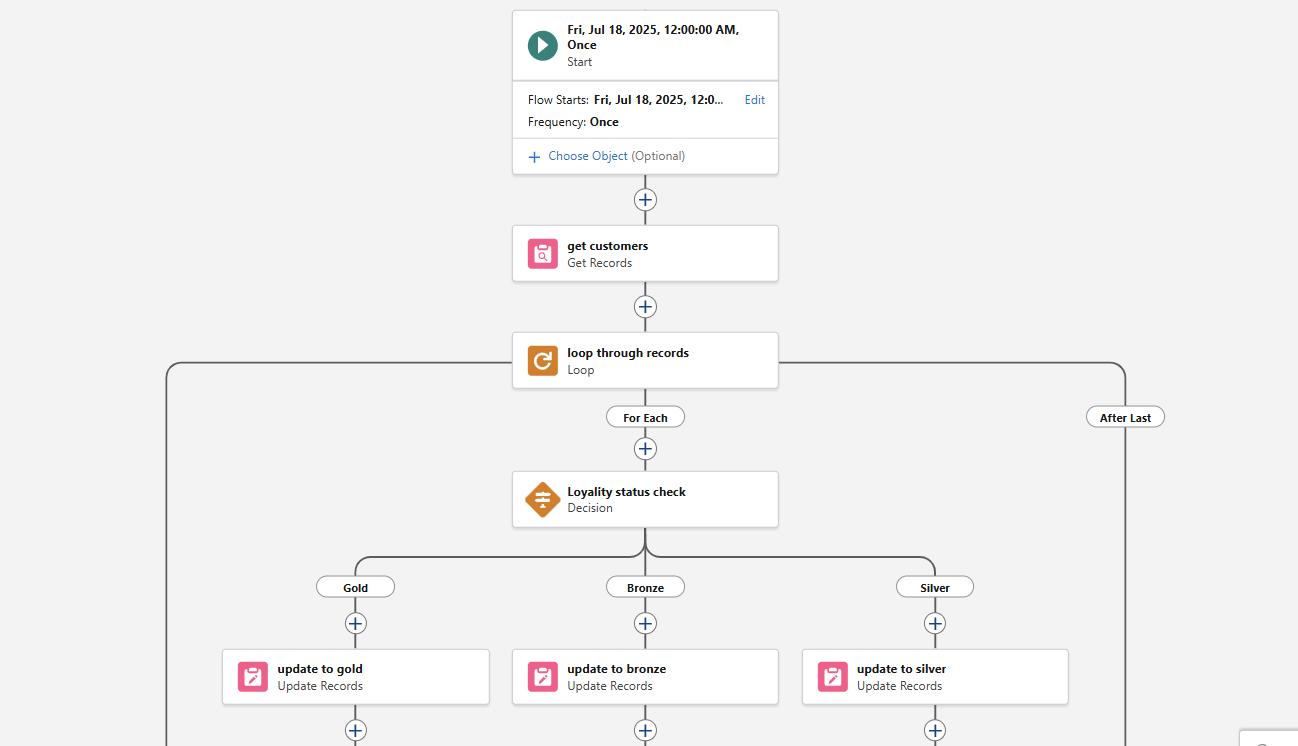
**Tracking & Rules:** - Field History Tracking enabled on Booking and Feedback. - Duplicate Rules and Matching Rules setup for Contact email.

**Security:** - Profiles created for different access levels. - Role Hierarchy from Admin → Manager → Staff. - Permission Sets for Reporting Access. - Sharing Rules to ensure object visibility.

**Testing:** - Created Test Classes for Apex Triggers (70%+ code coverage). - Manual test cases for Booking creation, Approval process. - Screenshots captured for Input Form, Approval Email, Task Automation, Flow Execution.





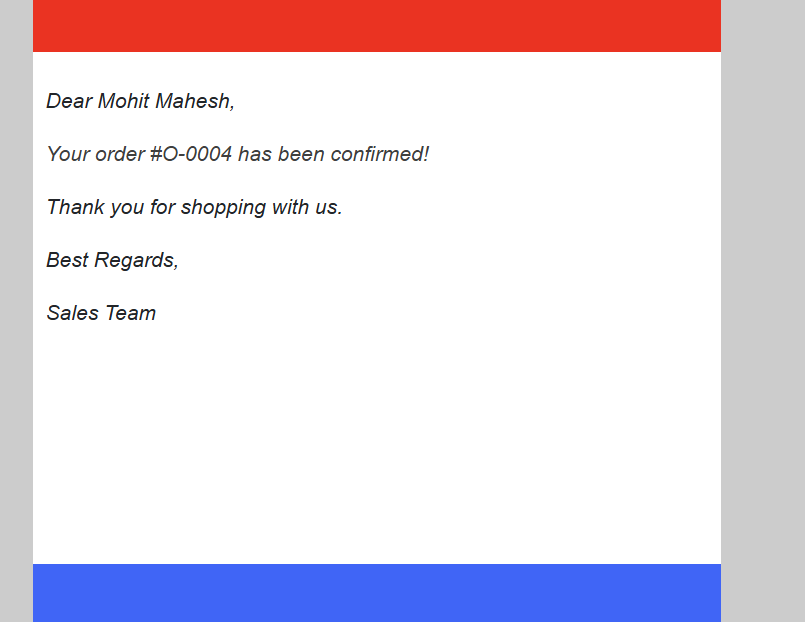


### **Phase 5: Deployment, Documentation & Maintenance**

**Deployment Strategy:** - Used Change Sets from sandbox to production. - Components migrated: Objects, Flows, Rules, Reports.

**Maintenance & Monitoring:** - Admin assigned to monitor user activity, errors, and feedback. - Backup of key data done weekly.

**Troubleshooting Approach:** - Field-level error messages to assist users. - Email notifications for failed automation.



# **Real-World Example**

Scenario: Retail Garment Store Chain (HandsMen Threads)

1. **Customer Registration:** A new customer signs up on the platform, providing contact and email details.

2. **Product Setup**: The product catalog is updated with new shirts using the HandsMen Product\_\_c object.

3. **Order Placement**: The customer places an order for five premium shirts.

4. **Order Total Calculation**: Upon saving the order, the Order Total is automatically calculated using an Apex Trigger.

5. **Inventory Update**: The stock count for shirts is reduced in real-time through a Stock Deduction Trigger.

6. **Stock Threshold Alert**: If stock goes below threshold, a Stock Alert Flow triggers an email to the Inventory Manager.

7. **Loyalty Evaluation**: A scheduled flow evaluates total order history and updates the customer’s loyalty status (e.g., Gold tier if orders > ₹10,000).

8. **Order Confirmation**: A confirmation email is sent to the customer using an Email Alert.

9. **Marketing Campaign**: The Marketing Team launches seasonal promotions using Marketing\_Campaign\_\_c object and targets loyal customers.

### **Conclusion**

This Salesforce-based CRM project has successfully transformed the manual customer engagement and booking process into a streamlined, automated system. It improves task efficiency, enhances customer experience, and provides decision-makers with real-time analytics. With scalable architecture and room for future enhancements, this CRM sets the foundation for digital transformation.

### **Future Enhancements**

* AI-based Lead Scoring using Einstein Analytics.
* Chatbot Integration for Booking and Feedback collection.
* Integration with Google Calendar for scheduling.
* Notification alerts via WhatsApp/SMS.

By:-Mohit Mahesh