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

## PROJECT OBJECTIVE:

HandsMen Threads is a premium men’s fashion brand specializing in bespoke tailoring and personalized styling. To enhance customer experience and streamline internal operations, the organization is implementing a Salesforce-based solution.

This project focuses on building a **robust data model**, enforcing **data integrity at the UI level**, and introducing **automation-driven business processes** such as order confirmations, loyalty management, stock alerts, and scheduled bulk order processing.

The Salesforce platform enables HandsMen Threads to centralize data, automate workflows, and deliver a seamless, personalized customer journey while ensuring operational efficiency.

## SCOPE :

* Custom object creation and relationship modeling
* Validation rules and formula fields
* Record-triggered flows and Apex triggers
* Email templates and automated notifications
* Asynchronous Apex and batch processing
* Data security and sharing configuration
* Testing, deployment, and user training

## TECHNOLOGY STACK

| **Category** | **Technology** |
| --- | --- |
| CRM Platform | Salesforce |
| Automation | Record-Triggered Flows, Apex Triggers |
| Backend Logic | Apex |
| Async Processing | Batch Apex, Scheduled Apex |
| UI Configuration | Lightning App Builder |
| Notifications | Salesforce Email Templates |
| Security | Profiles, Permission Sets, Sharing Rules |

## 

**System Requirements:**

**Hardware Requirements:**

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

**Software Requirements**:

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

**Skills Required:**

* Salesforce CRM fundamentals
* Custom Object creation and relationship modeling
* Field creation (Text, Number, Picklist, Formula, Lookup, Master-Detail)
* Validation Rules for data integrity
* Record-Triggered Flows and Scheduled Flows
* Apex programming (Classes, Triggers)
* Asynchronous Apex (Batch Apex, Scheduled Apex)
* Email Templates and Email Alerts
* Data Security (Profiles, Roles, Permission Sets, Sharing Rules)
* Testing and debugging in Salesforce

## PHASES OVERVIEW:

| **Phase No.** | **Phase Name** | **Description** | **Page Numbers** |
| --- | --- | --- | --- |
| 1 | Requirement Analysis & Planning | Define business requirements, project scope, Salesforce architecture, and data model design. | 5-9 |
| 2 | Salesforce Development & Configuration | Create custom objects, fields, relationships, validation rules, flows, Apex triggers, batch jobs, and email alerts. | 10 –14  16-22 |
| 3 | UI/UX Development & Customization | Design intuitive interfaces using Lightning Components and Flows. | 15-16 |
| 4 | Testing & Data Validation | Perform unit testing, functional testing, validation testing, and automation verification. | 23 – 24 |
| 6 | Data Security & Access Control | Configure profiles, roles, permission sets, and sharing rules to secure Salesforce data. | 25 – 52 |
| 5 | Deployment, Documentation & Maintenance | Deploy the solution, prepare documentation, train users, and provide post-deployment support. | 53 – 56 |

## PROJECT OVERVIEW:

HandsMen Threads, a dynamic organization in the fashion industry, is embarking on a Salesforce project designed to revolutionize their data management and enhance customer relations. The project involves building a robust data model tailored to store all pertinent business data, ensuring a seamless flow of information across the organization.

A key aspect of this project is the maintenance of data integrity directly from the user interface (UI). This feature will safeguard the accuracy and consistency of the data, which is crucial for informed decision-making and reliable business operations. The project will integrate several new processes into the business workflow to improve customer service and operational efficiency:

Automated Order Confirmations: Post-order confirmation, customers will receive an email update, fostering engagement and strengthening customer relations.

Dynamic Loyalty Program: Customer loyalty statuses will be updated based on purchase history, enabling personalized rewards and promoting repeat business.

Proactive Stock Alerts: When stock levels drop below five units, automatic emails will notify the warehouse team, ensuring timely restocking and preventing stockouts.

Scheduled Bulk Order Updates: Daily midnight, the system will process bulk orders, updating financial records and adjusting inventory, ensuring accurate stock levels for daily operations.

## PROJECT OBJECTIVES :

* Design a scalable Salesforce data model to store all business-critical data.
* Maintain high data quality and integrity through validations and automation.
* Automate customer communication and internal operational workflows.
* Improve inventory visibility and prevent stock shortages.
* Enable loyalty-based personalization to boost customer retention.
* Ensure secure, reliable, and high-performing system operations.

## PHASE 1: ARCHITECTURE & PLANNING

* Define custom and standard objects
* Design fields, relationships, and formula fields
* Establish validation rules for data integrity
* Plan automation using flows and Apex
* Design email templates for customer and internal communication

## PHASE 2: DEVELOPMENT

* Create objects and fields
* Implement record-triggered flows
* Develop Apex triggers and batch jobs
* Configure data security and sharing rules
* Set up email alerts and notification workflows

## PHASE 3: TESTING & QUALITY ASSURANCE

* Unit testing of Apex and flows
* End-to-end testing using sample business data
* Performance testing for batch processes
* Security validation and access testing

## PHASE 4: DEPLOYMENT & TRAINING

* Deploy components to production environment
* Conduct end-user training sessions
* Monitor system post-go-live
* Provide post-deployment support

## MILESTONE 1: SALESFORCE ACCOUNT

**Introduction:**

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

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

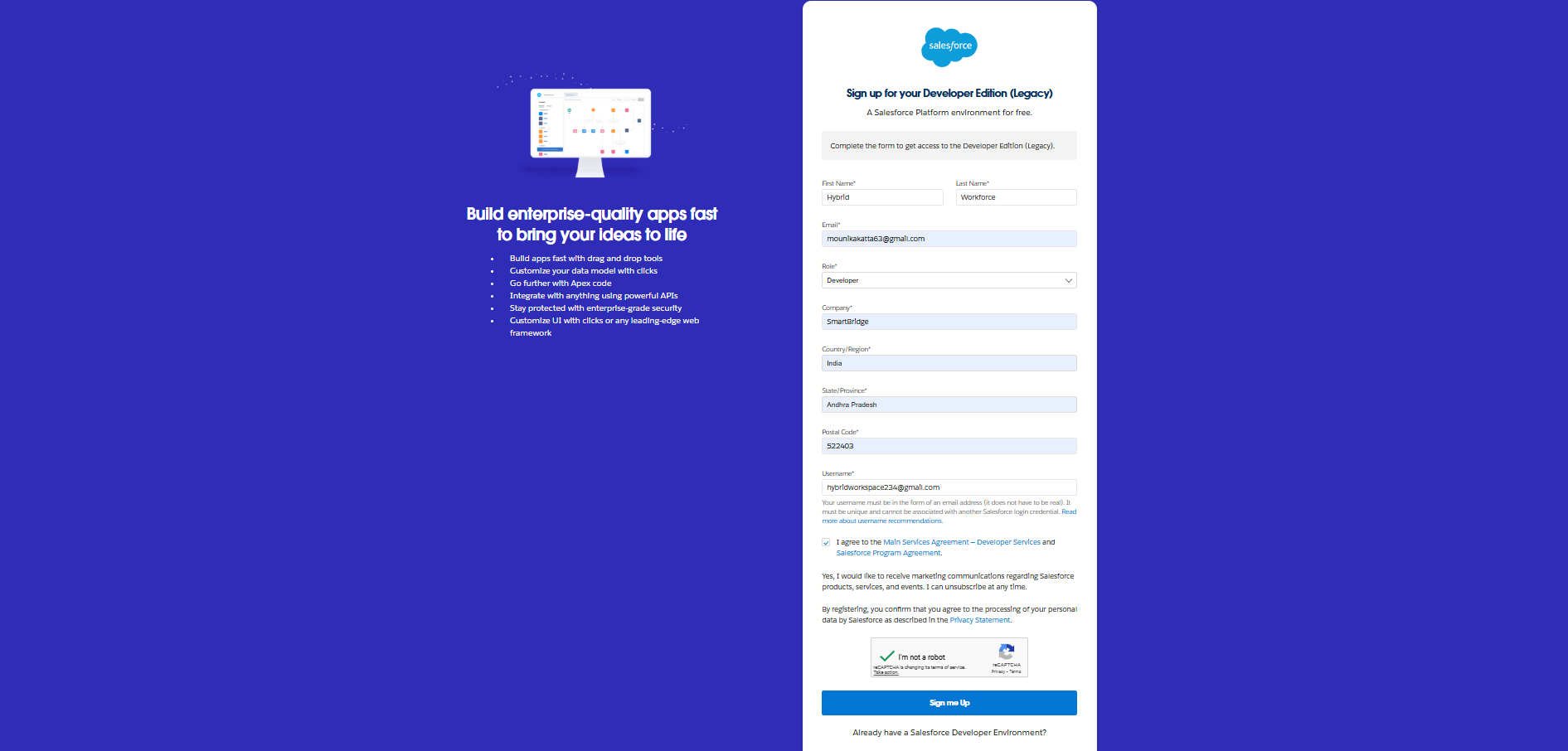
**What Is Salesforce?**

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

### **Activity 1: Creating Developer Account:**

Creating a developer org in salesforce.

1. Go to <https://developer.salesforce.com/signup>
2. On the sign-up form, enter the following details:

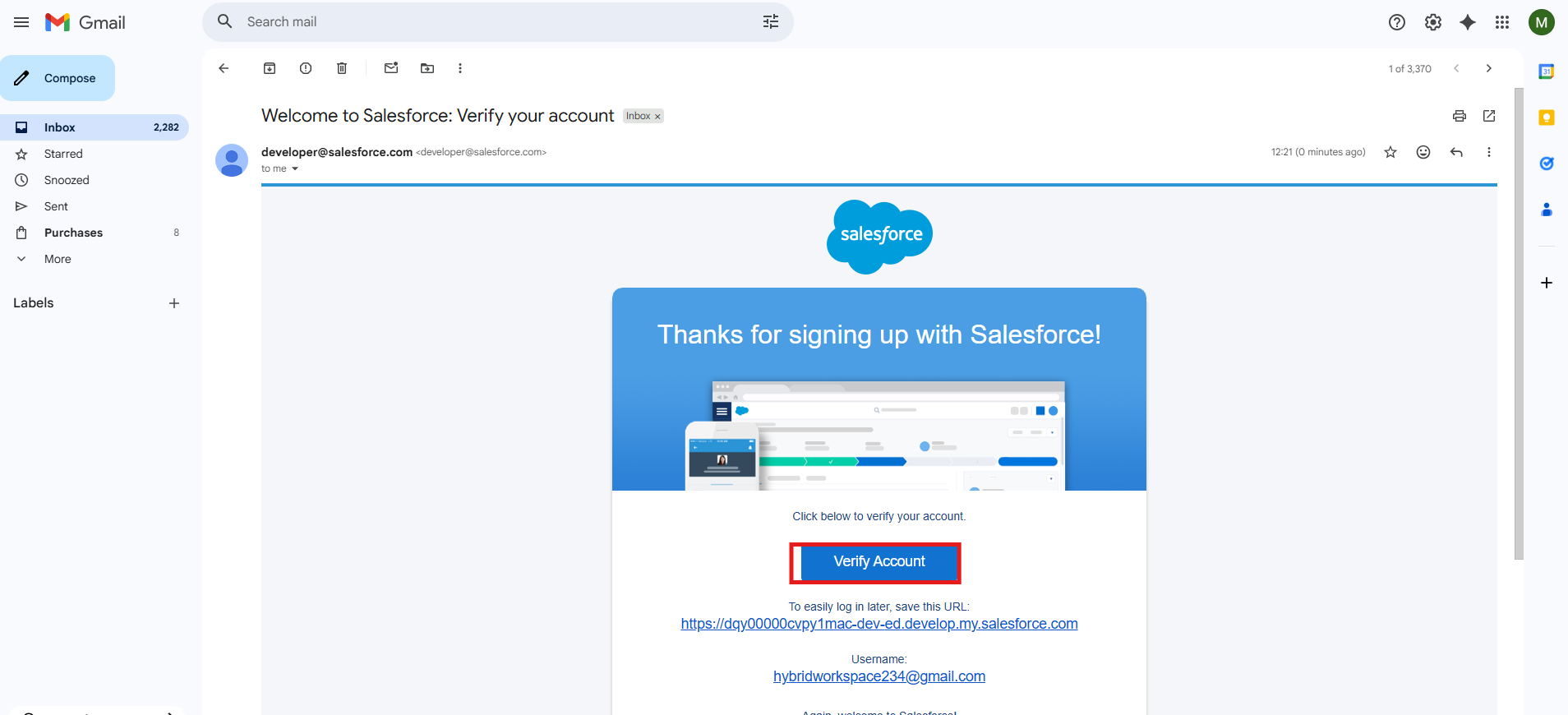


1. First name & Last name
2. Email
3. Job Title: Developer
4. Company: College Name
5. Country: India
6. Postal Code
7. Username : should be a combination of your name and company
8. This need not be an actual email id, you can give anything in the format : [username@organization.com](mailto:username@organization.com)

Click on sign me up after filling these.

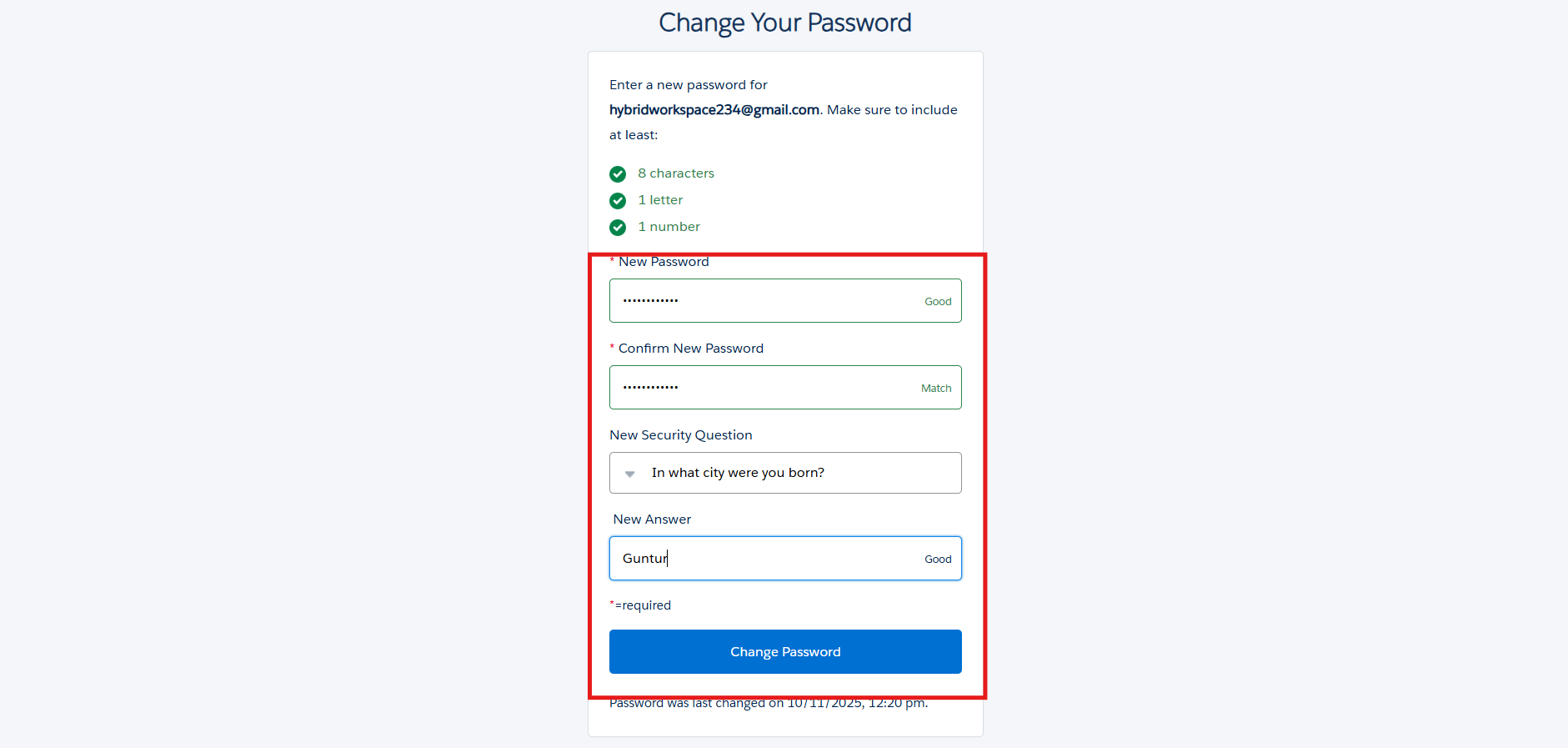
### **Activity 2: Account Activation**

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 10-30mins and sometimes 2 hours.

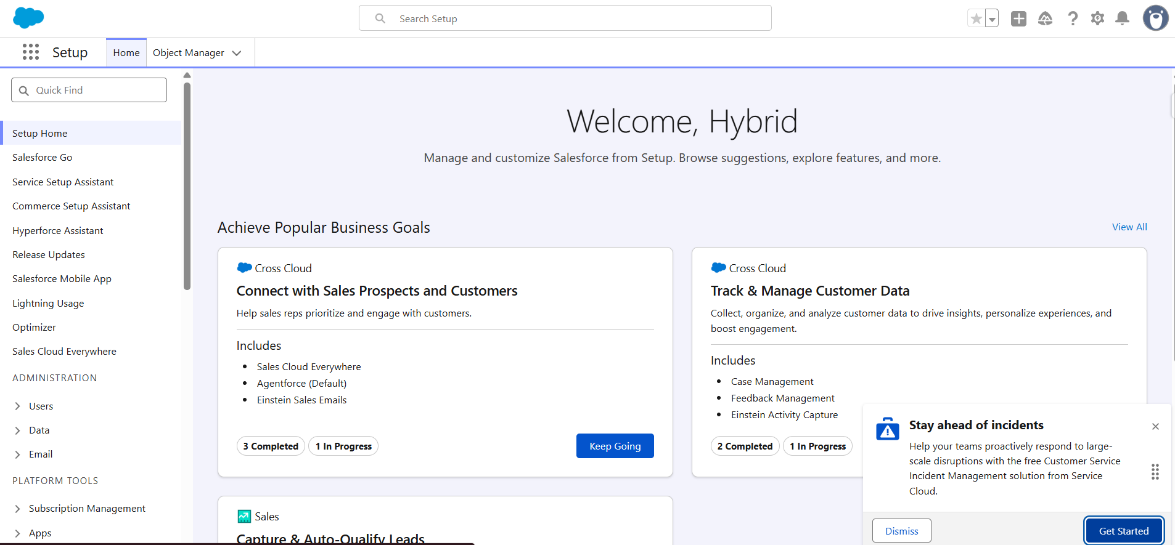


2.Click on Verify Account

3.Give a password and answer a security question and click on change password.

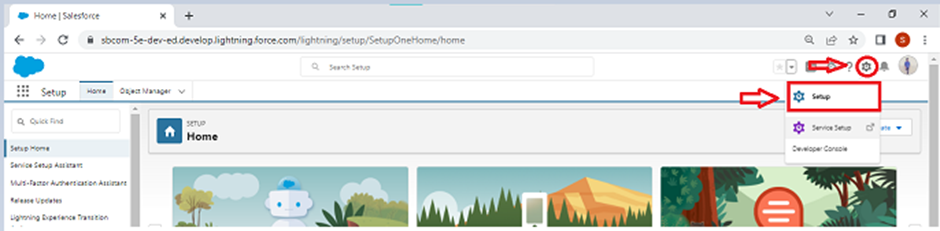


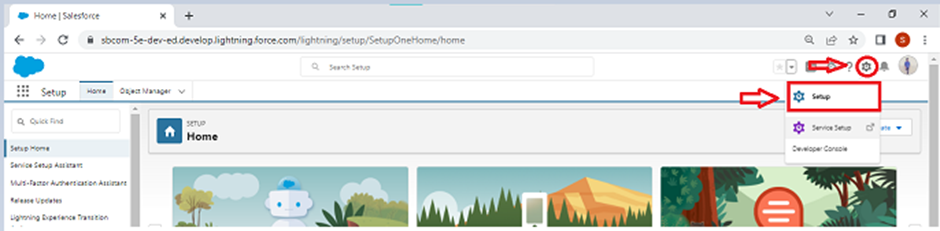
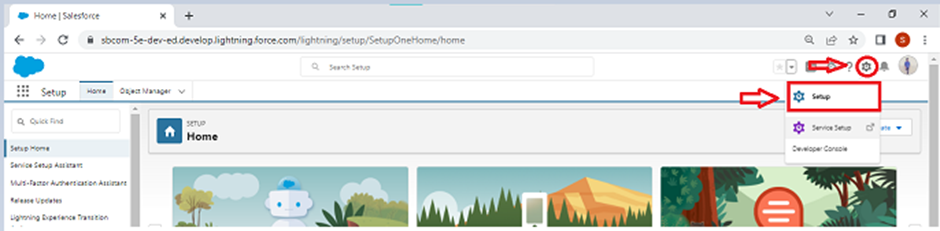
* 1. Then you will redirect to your salesforce setup page.



## MILESTONE 2: OBJECTS CREATION

**To Navigate to Setup page:**

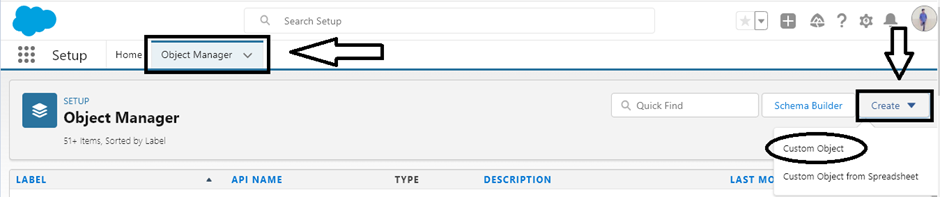
Click on gear icon → click setup. 



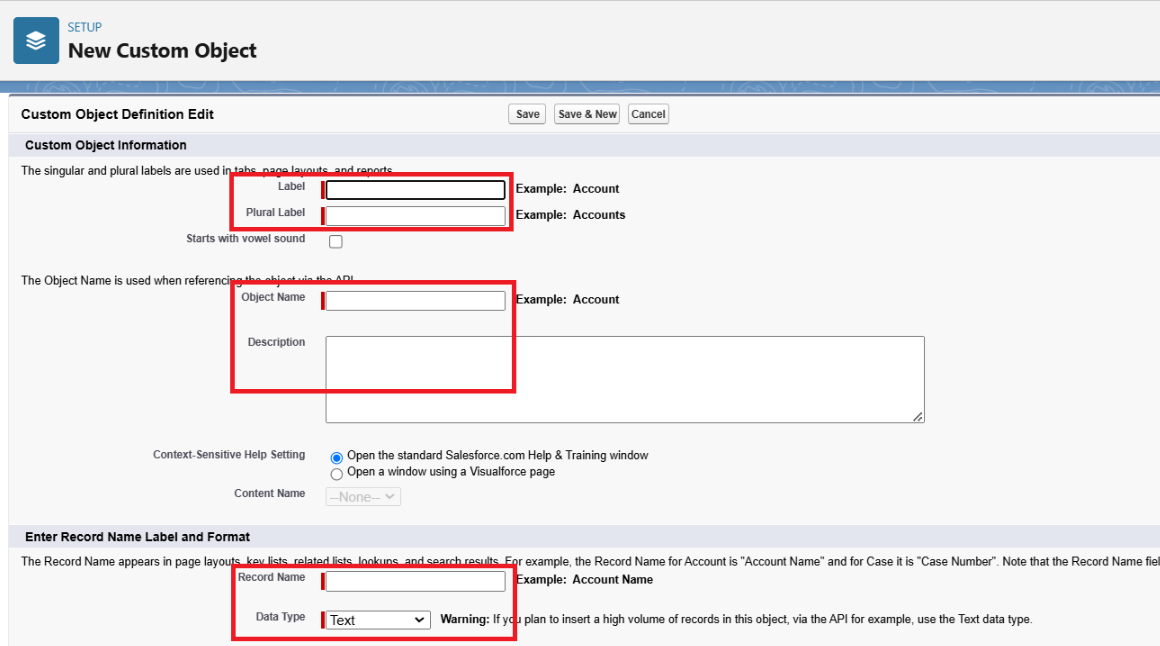
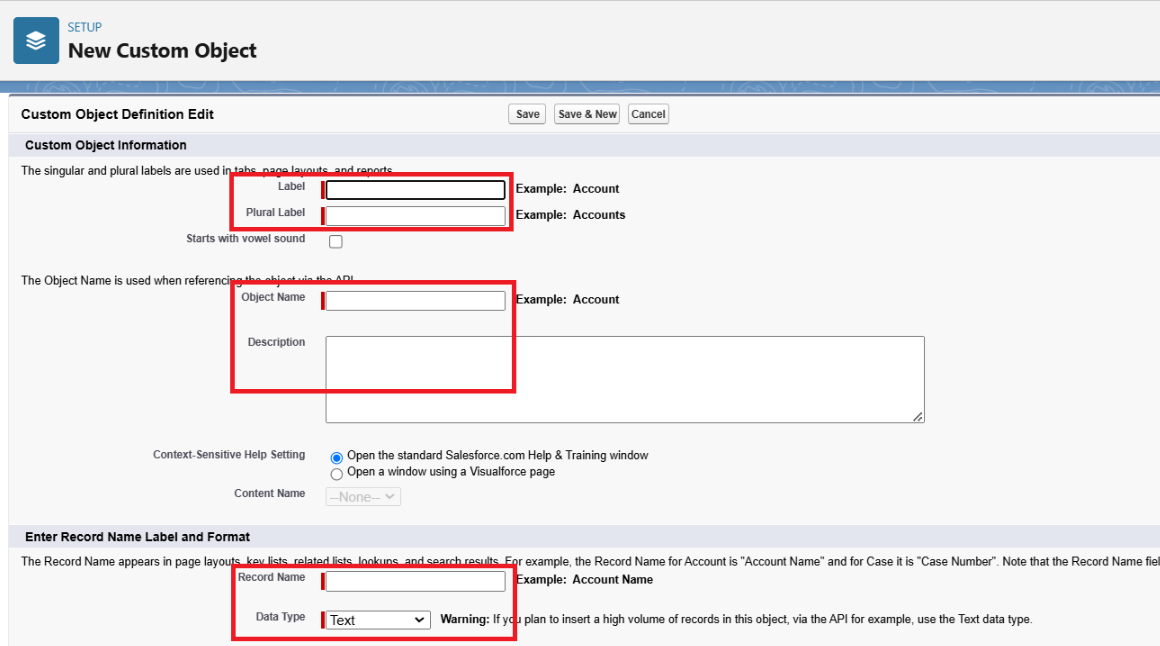
**Activity 1: Create HandsMen Customer Object**

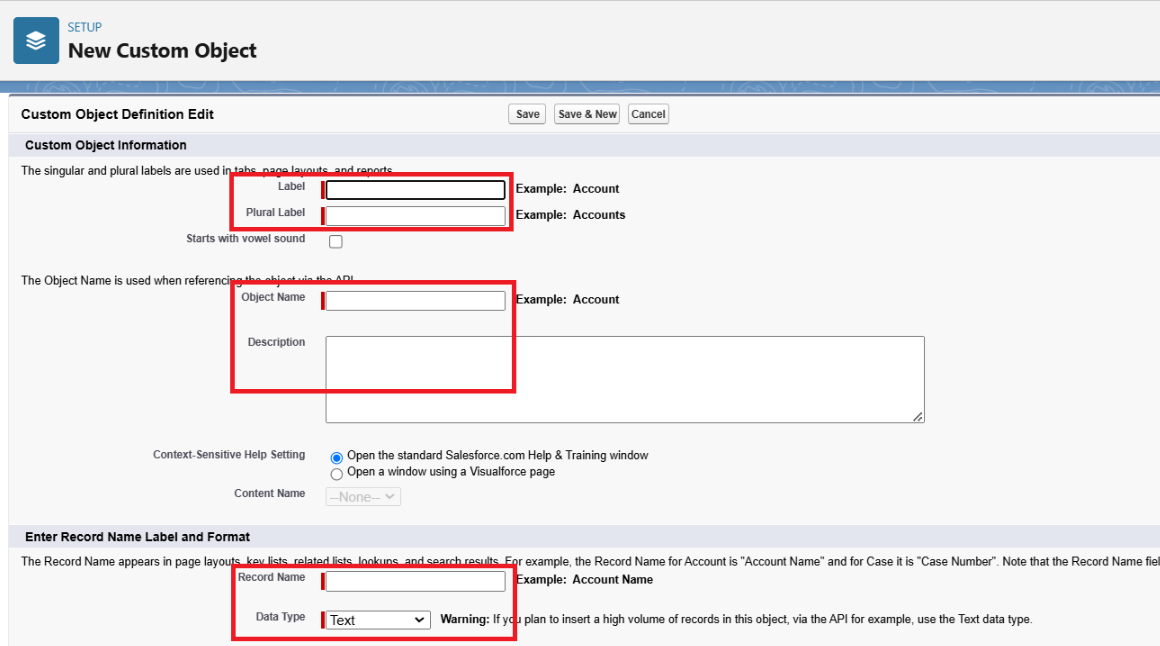
To create an object:

From the setup page → Click on Object Manager → Click on Create → Click on Custom Object.



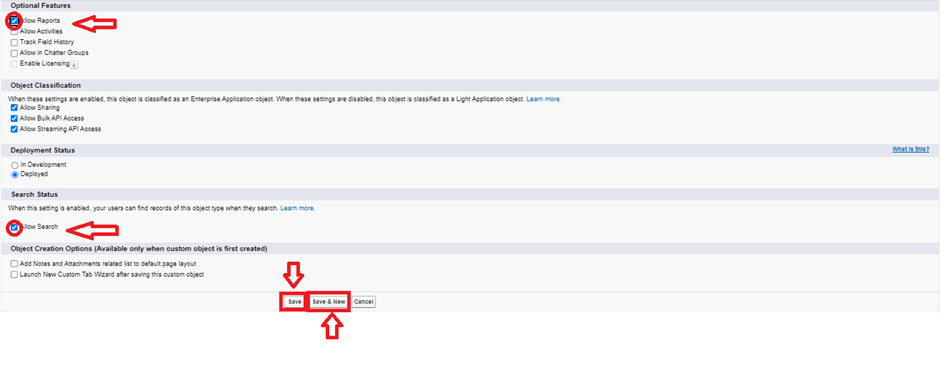
* Enter the label name→ HandsMen Customer
* Plural label name→ HandsMen Customer
* Enter Record Name Label and Format
* Record Name → HandsMen Customer Name
* Data Type → Text





Click on Allow reports,

Allow search → **Save.**



**Create HandsMen Product Object**

1. To create an object:
2. From the setup page → Click on Object Manager → Click on Create → Click on Custom Object.
3. Enter the label name→ HandsMen Product
4. Plural label name→ HandsMen Products
5. Enter Record Name Label and Format
6. Record Name → HandsMen Product Name
7. Data Type → Text
8. Click on Allow reports,
9. Allow search → **Save**

**Create HandsMen Order Object**

1. To create an object:
2. From the setup page → Click on Object Manager → Click on Create → Click on Custom Object.
3. Enter the label name→ HandsMen Order
4. Plural label name→ HandsMen Orders
5. Enter Record Name Label and Format
6. Record Name → HandsMen OrderNumber
7. Data Type → Auto Number
8. Display Format → O-{0000}
9. Starting Number → 001
10. Click on Allow reports,
11. Allow search → **Save**

**Create Inventory Object**

To create an object:

1. From the setup page → Click on Object Manager → Click on Create → Click on Custom Object.
2. Enter the label name→ Inventory
3. Plural label name→ Inventorys
4. Enter Record Name Label and Format
5. Record Name → Inventory Number
6. Data Type → Auto Number
7. Display Format → I -{0000}
8. Starting Number → 001
9. Click on Allow reports,
10. Allow search → **Save**

**Create Marketing Campaign Object**

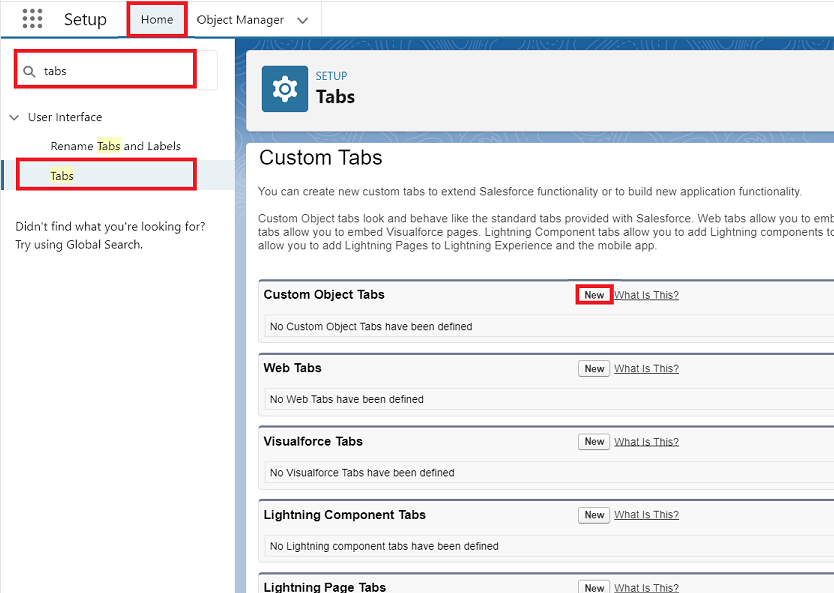
To create an object:

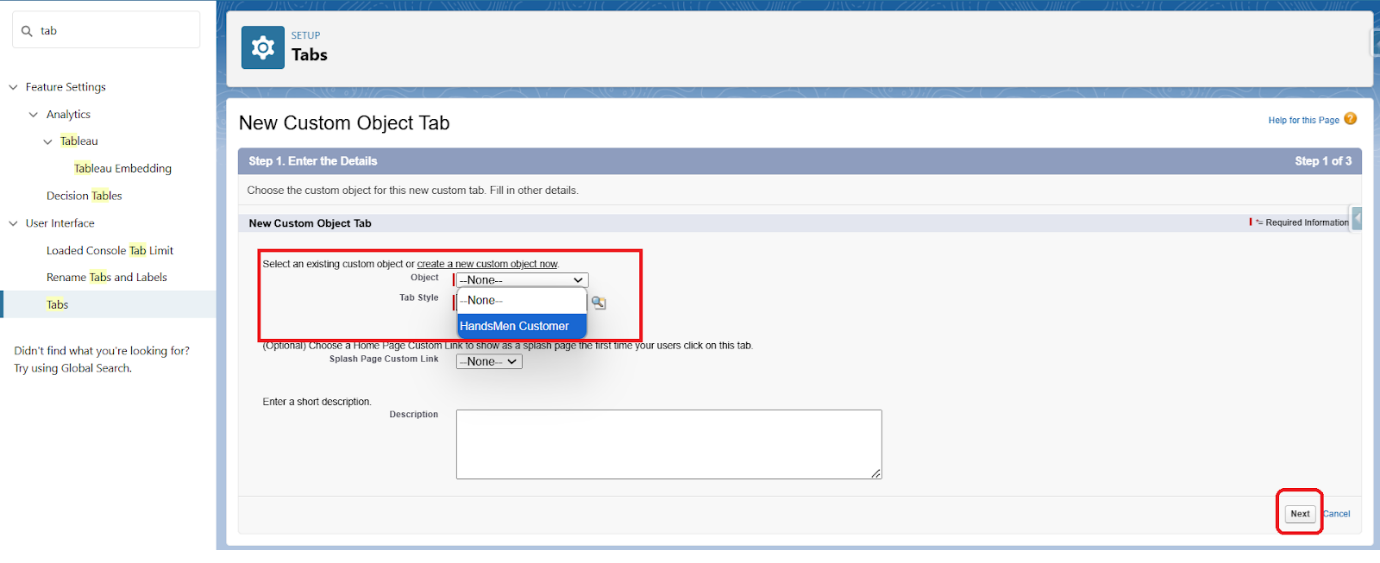
1. From the setup page → Click on Object Manager → Click on Create → Click on Custom Object.
2. Enter the label name→ Marketing Campaign
3. Plural label name→ Marketing Campaigns
4. Enter Record Name Label and Format
5. Record Name → Marketing CampaignNumber
6. Data Type → Auto Number
7. Display Format → MC -{0000}
8. Starting Number → 001
9. Click on Allow reports,
10. Allow search → **Save**

## MILESTONE 3: TAB CREATION

**Creating a Custom Tab(HandsMen Customer)**

**To create a Tab(HandsMen Customer )**

1. Go to setup page → type Tabs in Quick Find bar → click on tabs → New (under custom object tab)

Select Object(HandsMen Customer) → Select any tab style → Next (Add to profiles page) keep it as default → Next (Add to Custom App) keep it as default → Save.

1. Now create tabs for

* Object - HandsMen Customer
* Object - HandsMen Product
* Object - HandsMen Order
* Object - Inventory
* Object - Marketing Campaign

objects as well which we have created above Objects.

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

## The UI/UX Development & Customization phase focused on building an intuitive and responsive interface for the Handsmen Threads Project. The goal was to deliver a seamless user experience for company and Customers to access and manage customer relationship and product order sales.

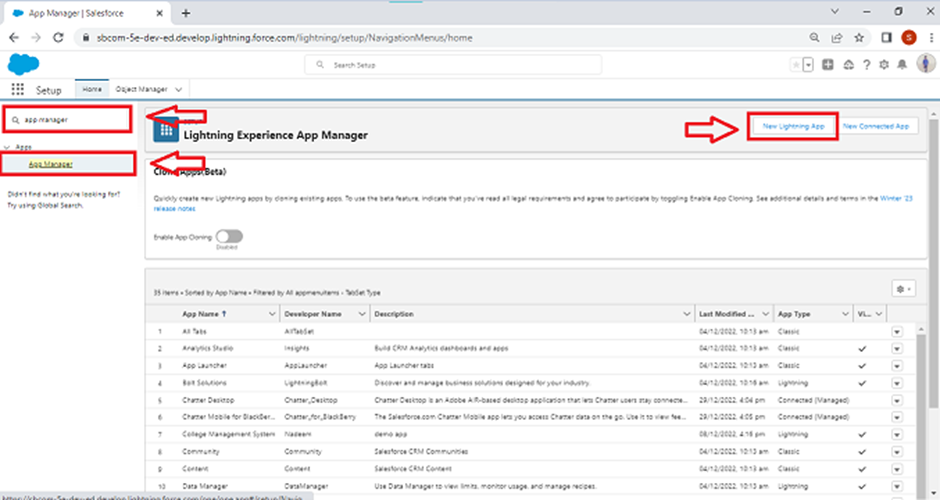
## During this phase, Lightning Application and Page Layouts were created and customized to ensure smooth navigation and a connected design aligned with organizational branding.

## MILESTONE : 4 CREATE A LIGHTNING APP

**Create a Lightning App**

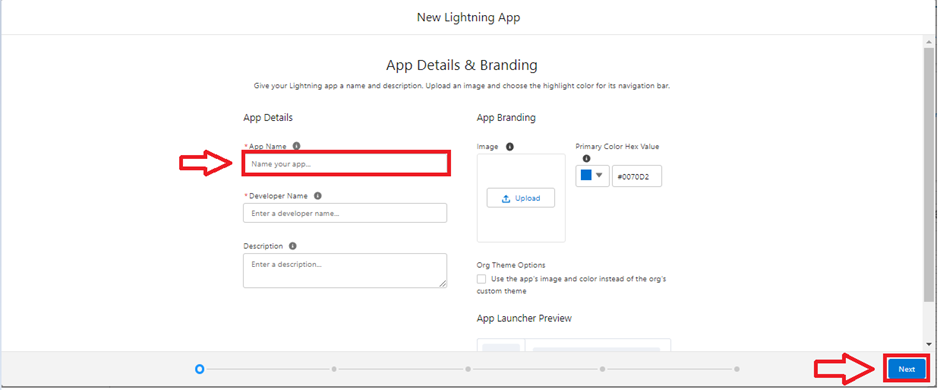
**To create a lightning app page:**

Go to setup page → search “app manager” in quick find → select “app manager” → click on New lightning App.

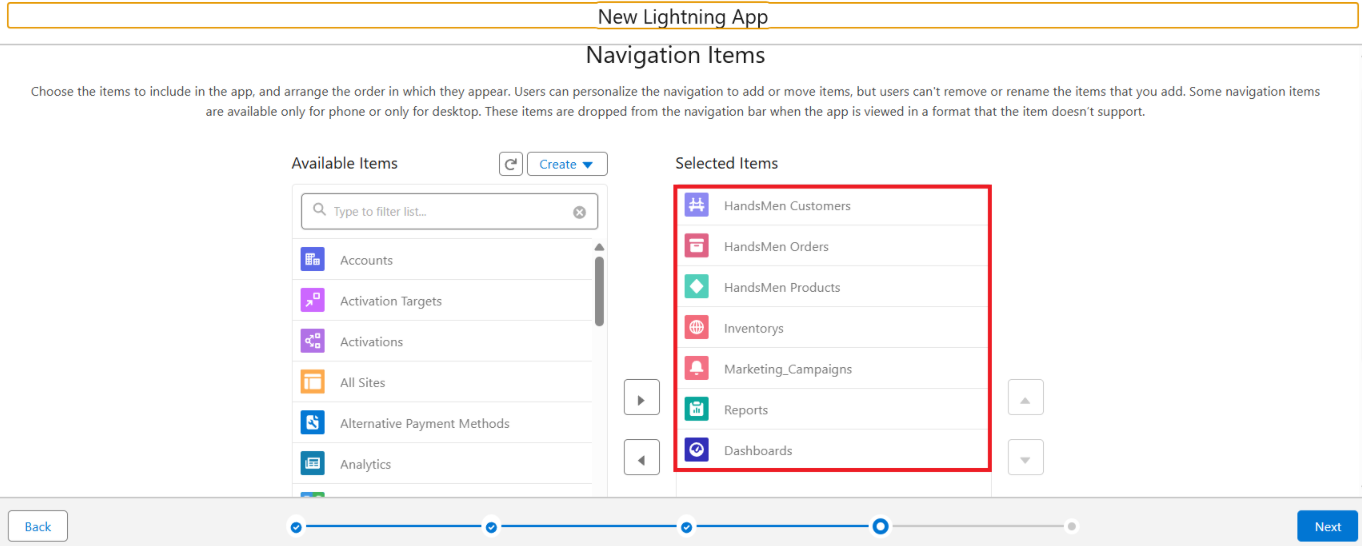


1. Fill the app name in app details and branding as follow  
   App Name : HandsMen Threads  
   Developer Name : this will auto populated  
   Description : Give a meaningful description  
   Image : optional (if you want to give any image you can otherwise not mandatory)  
   Primary color hex value : keep this default

1. Then click Next  → (App option page) keep it as default → Next → (Utility Items) keep it as default → Next.

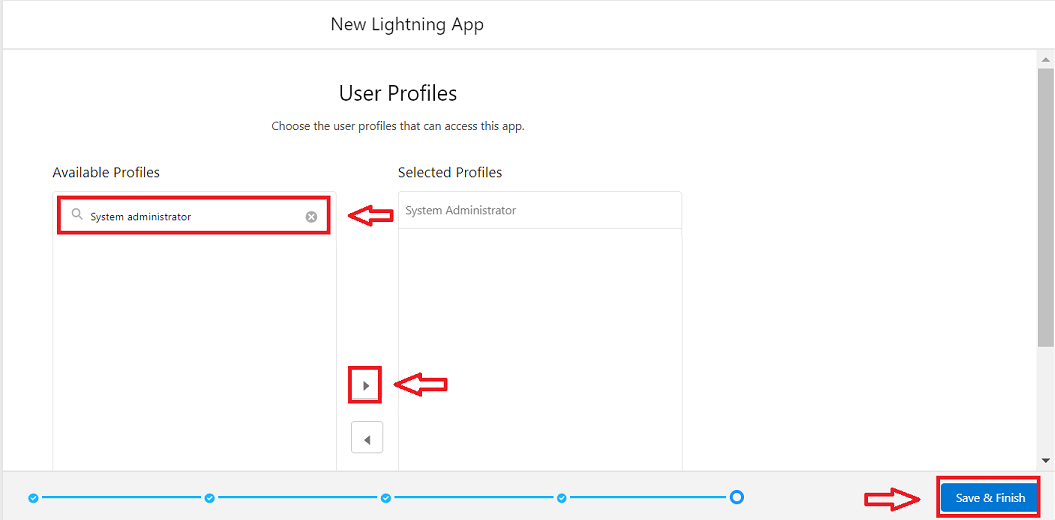


To Add Navigation Items:



Search the items in the search bar(HandsMen Customer, HandsMen Order, Inventory, HandsMen Product, Reports, Dashboard, Account, Contact , Marketing Campaign) from the search bar and move it using the arrow button → Next.  
**Note**: select the custom object which we have created in the previous activity.

1.To Add User Profiles:

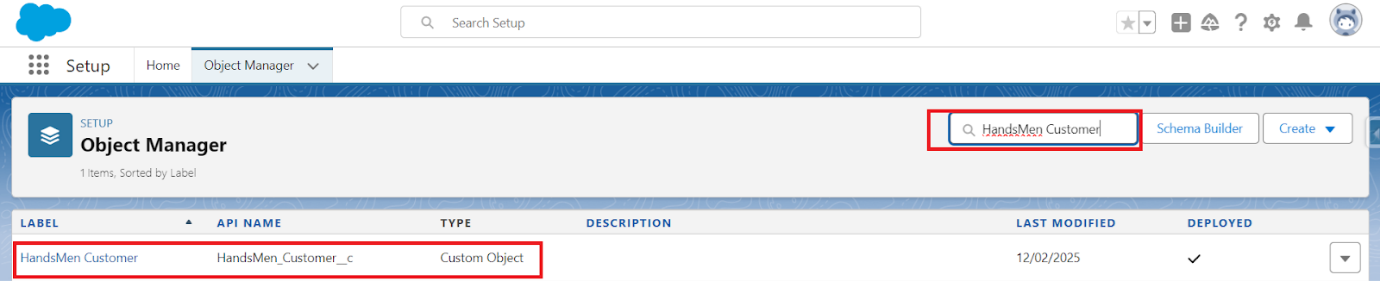


Search profiles (System administrator) in the search bar → click on the arrow button → save & finish.

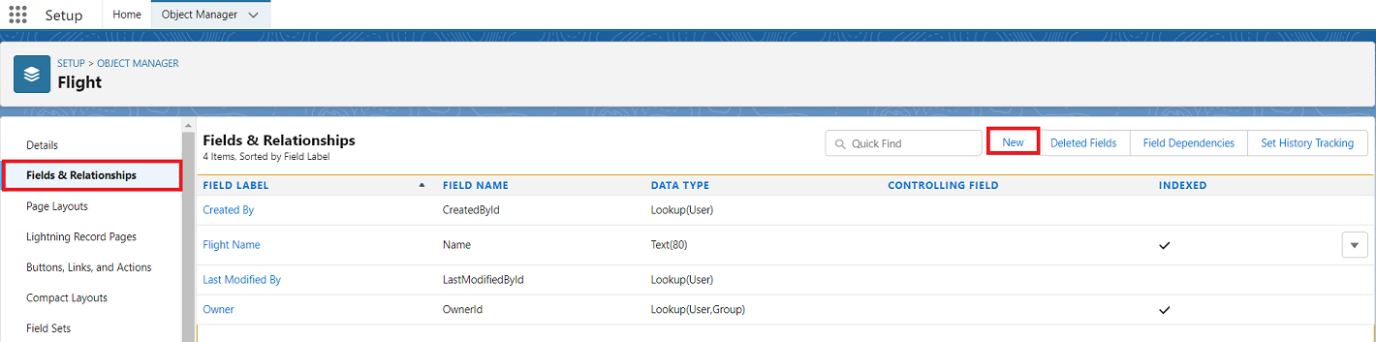
## MILESTONE : 5 FIELD CREATION

**Creating Field in HandsMen Customer Object**

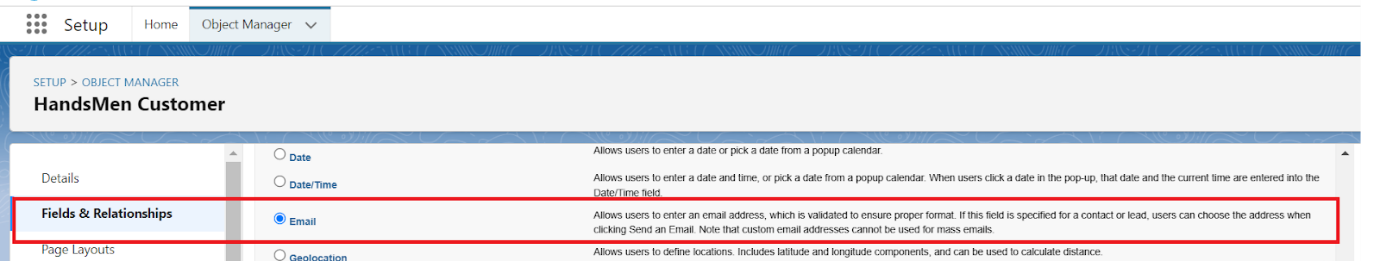
1. To create fields in an object:
2. Go to setup → click on Object Manager → type object name(HandsMen Customer) in quick find bar→ click on the object.



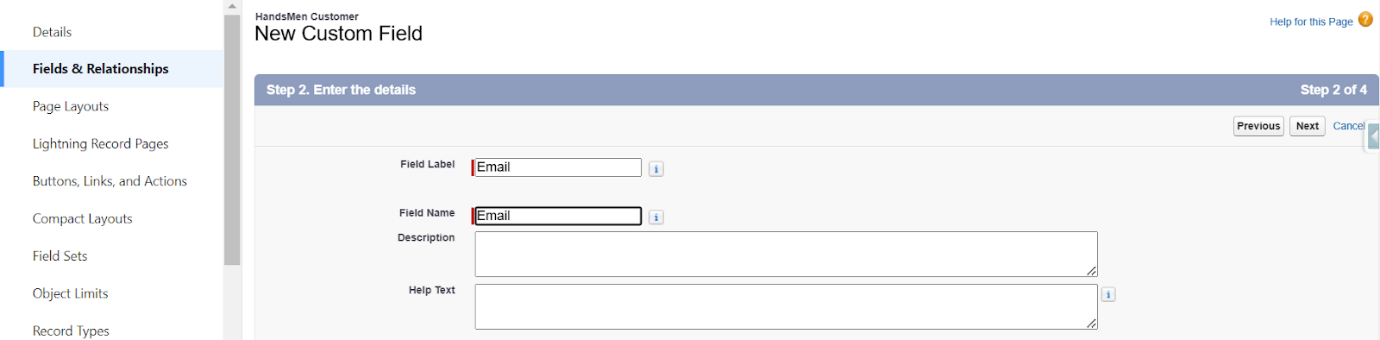
1. Now click on “Fields & Relationships” → New



1. Select Data type as “Email”.



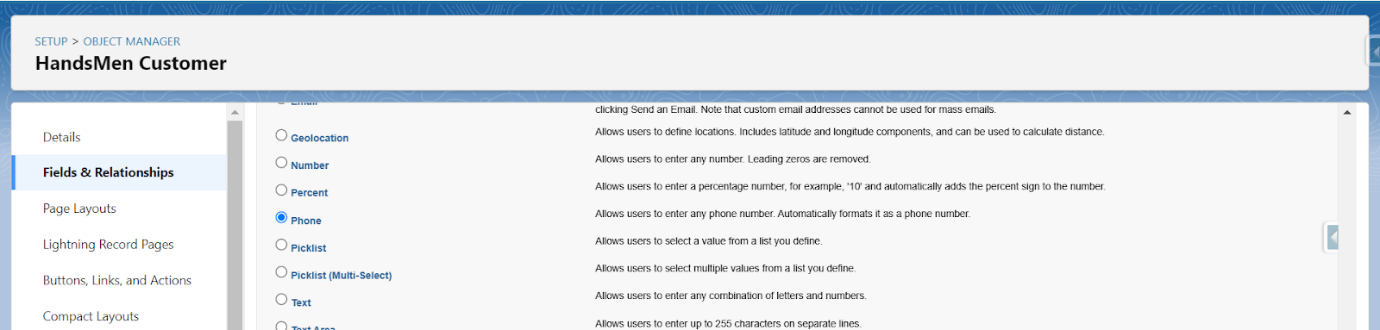
1. Click on Next



1. Fill the above as following:
2. Field Label: Email
3. Field Name : gets auto generated
4. Click on Next → Next → Save and new.

**Creating Phone on HandsMen Customer Object**

1. Go to setup → click on Object Manager → type object name(HandsMen Customer) in quick find bar→ click on the object.
2. Now click on “Fields & Relationships” → New.
3. Select Data type as “Phone” and click Next.

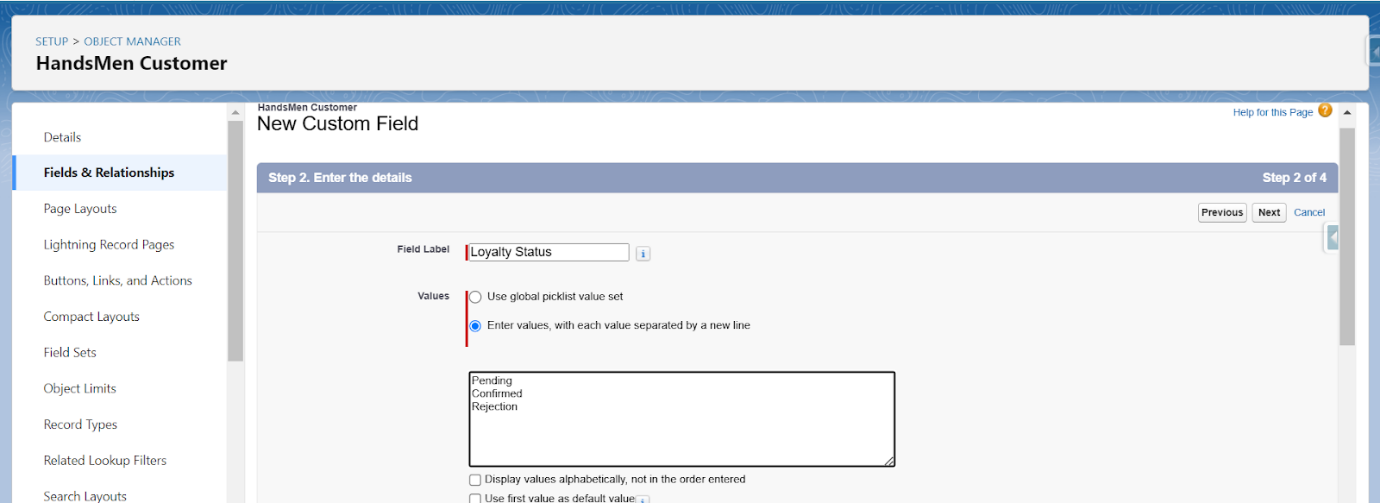


1. Click on Next.
2. Fill the above as following:
3. Field Label: Phone.
4. Field Name : gets auto generated.
5. Click on Next → Next → Save and new.

**Creating Picklist field on HandsMen Customer object.**

1. Go to setup → click on Object Manager → type object name(HandsMen Customer) in quick find bar→ click on the object.
2. Now click on “Fields & Relationships” → New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Loyalty Status”, under values select “Enter values, with each value separated by a new line" and enter values as shown below:

* Gold
* Silver
* Bronze



1. Click Next→ Next → Next → Save & New.

**Unit 1 : Creating Lookup Relationship between Marketing Campaign and HandsSome Customer**

1. To Create a lookup relationship
2. Go to the setup page → click on object manager → type object name(Marketing Campaign) in the quick find bar→ click on the object.
3. Click on fields & relationship → click on New.
4. Select “lookup relationship” as data type and click Next.
5. For field label related to: select “HandsMen Customer” object and click Next.
6. Give Field Label as “HandsMen Customer” and click Next.
7. Next → Next → Save.

**Unit 2 : Creating Lookup Relationship between HandsMen Product and HandsMen Order**

1. To Create a lookup relationship
2. Go to the setup page → click on object manager → type object name(HandsMen Product) in the quick find bar→ click on the object.
3. Click on fields & relationship → click on New.
4. Select “lookup relationship” as data type and click Next.
5. For field label related to: select “HandsMen Order” object and click Next.
6. Give Field Label as “Order” and click Next.
7. Next → Next → Save.

**Unit 3 : Creating Lookup Relationship between HandsMen Order and HandsMen Customer**

1. To Create a lookup relationship
2. Go to the setup page → click on object manager → type object name(HandsMen Order) in the quick find bar→ click on the object.
3. Click on fields & relationship → click on New.
4. Select “lookup relationship” as data type and click Next.
5. For field label related to: select “**HandsMen Customer**” object and click Next.
6. Give Field Label as “Customer” and click Next.
7. Next → Next → Save.

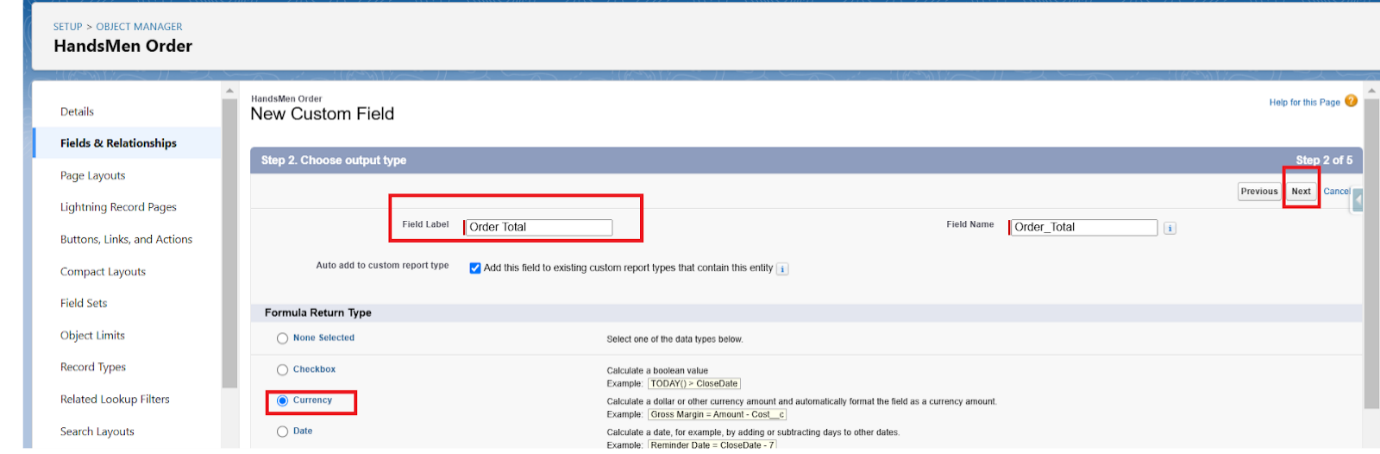
**Unit 4 : Creating Master-Detail Relationship between Inventory and HandsSome Product**

1. To Create a Master-Detail relationship
2. Go to the setup page → click on object manager → type object name(Inventory) in the quick find bar→ click on the object.
3. Click on fields & relationship → click on New.
4. Select “**Master-Detail** relationship” as data type and click Next.
5. For field label related to: select “**HandsMen Product**” object and click Next.
6. Give Field Label as “Product” and click Next.
7. Next → Next → Save.

|  |  |  |  |
| --- | --- | --- | --- |
| **Object** | **Field** | **Formula Return type** | **Formula** |
| Inventory\_\_c | Stock\_Status\_\_c | text | IF(Stock\_Quantity\_\_c > 10, "Available", "Low Stock") |
| HandsMen Customer\_\_c | Full\_Name\_\_c | text | FirstName & " " & LastName |

**Note :** Before creating the “FullName” formula field in the HandsMen Customer object, create 2 Custom fields with the name “FirstName” and “LastName”.

1. Go to the setup page → click on object manager → type object name(HandsMen\_Customer\_\_c) in the quick find bar→ click on the object.
2. Click on fields & relationship → click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “Full\_Name\_\_c” and select formula return type as “Text” and click next.



1. Under Advanced Formula write down the formula and click “Check Syntax” and Next→ Next→ Save & New.

**Relationships**

* Customer → Order (One-to-Many)
* Order → Order Item (One-to-Many)
* Product → Inventory (One-to-One)
* Customer → Loyalty Program (One-to-One)

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Type** | **Description** | **Key Fields** |
| **HandsMen Customer\_\_c** | Custom Object | Stores customer details | Name (Record Name),  Email (Email),  Phone (Phone),  Loyalty\_Status\_\_c (Picklist: Bronze, Gold, Silver) [Loyalty status field is already created in previous activity do not create it again, Total\_Purchases\_\_c (Number) |
| **HandsMen Product\_\_c** | CustomObject | Stores product catalog | Name (Record Name),  SKU ( Text ),  Price (Currency), Stock\_Quantity\_\_c (Number) |
| **HandsMen Order\_\_c** | CustomObject | Stores customer orders | Order\_Number (Record Name),  Status (Picklist: Pending, Confirmed, Rejection), Quantity\_\_c (Number), Total\_Amount\_\_c(Number) |
| **Inventory\_\_c** | CustomObject | Tracks inventory levels | Auto Number (Record Name), Warehouse (Text),  Stock\_Quantity\_\_c (Number) |
| **Marketing\_Campaign\_\_c** | CustomObject | Manages promotions & campaigns | Campaign\_Name (Record Name),  Start\_Date (Date),  End\_Date (Date) |

## MILESTONE : 6 VALIDATION RULES

**Validation Rules**

|  |  |  |
| --- | --- | --- |
| **Object** | **Field** | **Validation Rule** |
| HandsMen Order\_\_c | Total\_Amount\_\_c | Total\_Amount\_\_c <= 0 |
| Inventory\_\_c | Stock\_Quantity\_\_c | Stock\_Quantity\_\_c < = 0 |
| HandsMen Customer\_\_c | Email | NOT CONTAINS(Email, "@[gmail.com](http://gmail.com/)") |

**Activity 1 : Creating the validation rule**

**Creating the validation rule for Postal Code field in HandsMen Order\_\_c object**

**Note :** check whether the fields mentioned in the formula field are created or not , if not create those fields mentioned in **HandsMen Order\_\_c** object.

1. Go to setup → click on Object Manager → type object name(**HandsMen Order\_\_c** ) in quick find bar→ click on the object.
2. Click on the validation rule → click New.

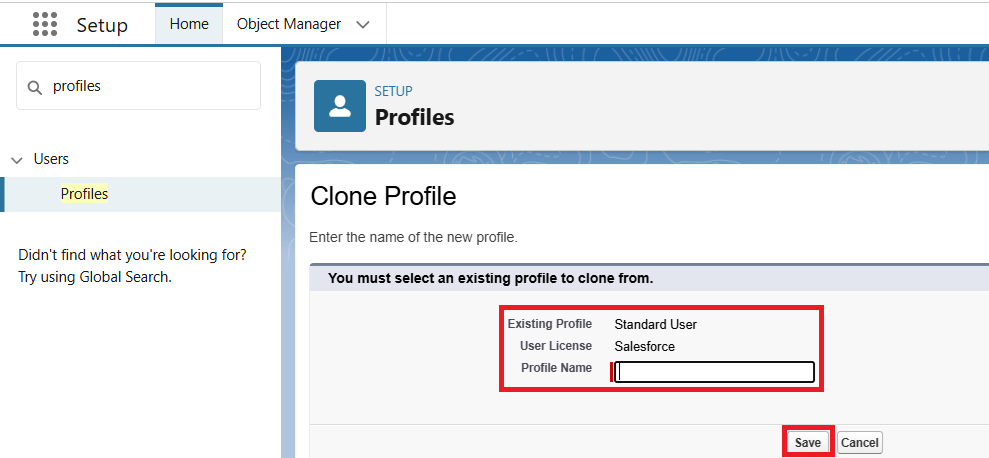


1. Enter the Rule name as “Total Amount“.
2. Insert the Error Condition Formula as : -
3. Total\_Amount\_\_c <= 0
4. Enter the Error Message as “Please Enter Correct Amount”, select the Error location as Field and select the field as “Total Amount”, and click Save.

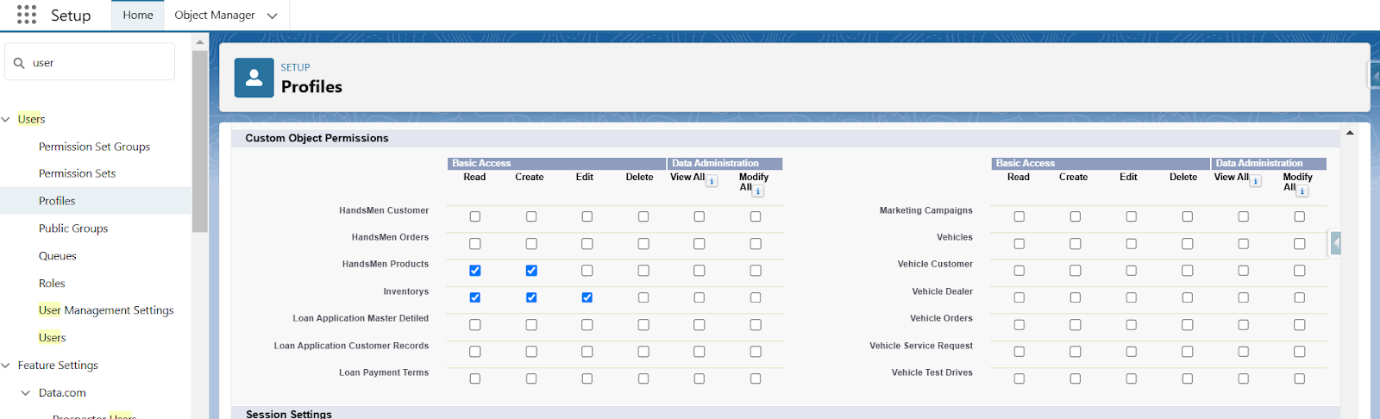
## MILESTONE : 7 DATA SECURITY

**Activity 1: Creating Profile**

1. Go to setup → type profiles in quick find box → click on profiles → clone the desired profile (Standard user) → enter profile name (Platform 1) → Save.



1. While still on the profile page, then click Edit.
2. Scroll down to Custom Object Permissions and Give access permissions for HandsMen products and Inventory objects.

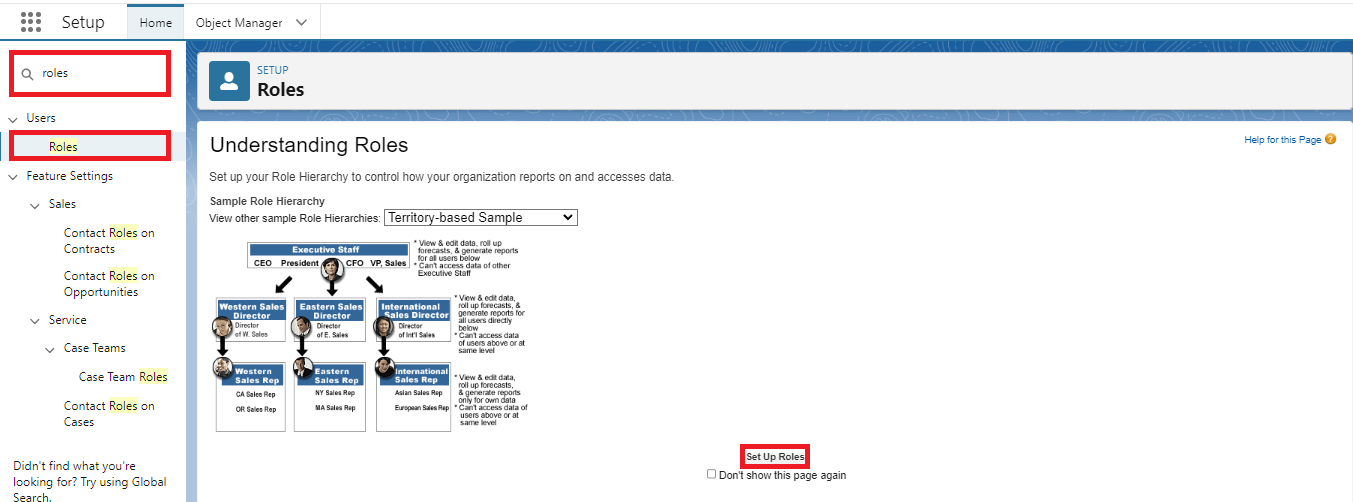


**Creating Profile**

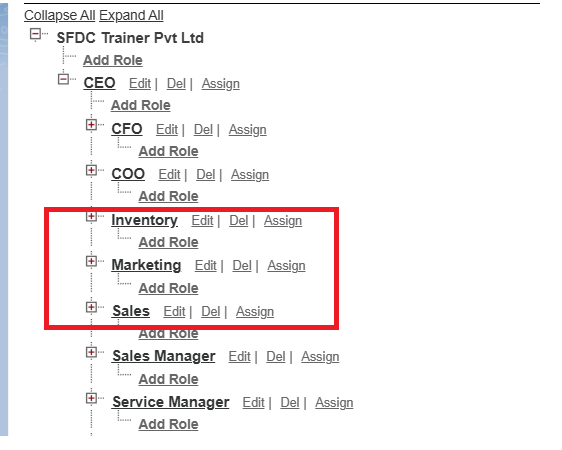
1. Go to setup → type profiles in quick find box → click on profiles → clone the desired profile (Standard user) → enter profile name (Platform 1) → Save.
2. While still on the profile page, then click Edit.
3. Scroll down to Custom Object Permissions and Give access permissions for HandsMen products and Inventory objects.
4. Scroll down and Click on Save.

**Activity 2: Creating Sales Manager Role:**

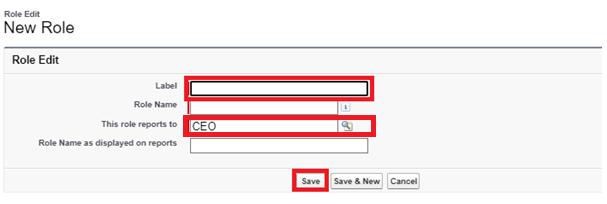
1. Go to quick find → Search for Roles → click on set up roles.



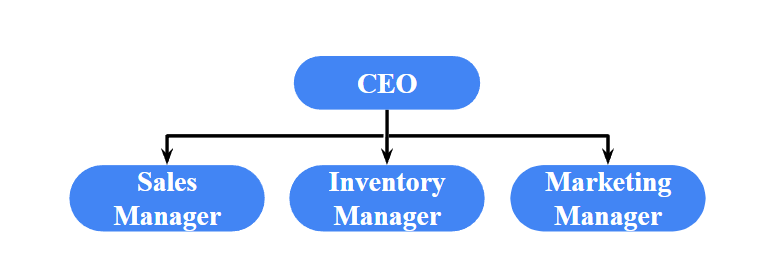
1. Click on Expand All and click on add role under whom this role works(Here Click Add Role Under CEO role).



1. Give Label as “Sales” and Role name gets auto populated. Check to whom this role (Sales) reports. Then click on Save



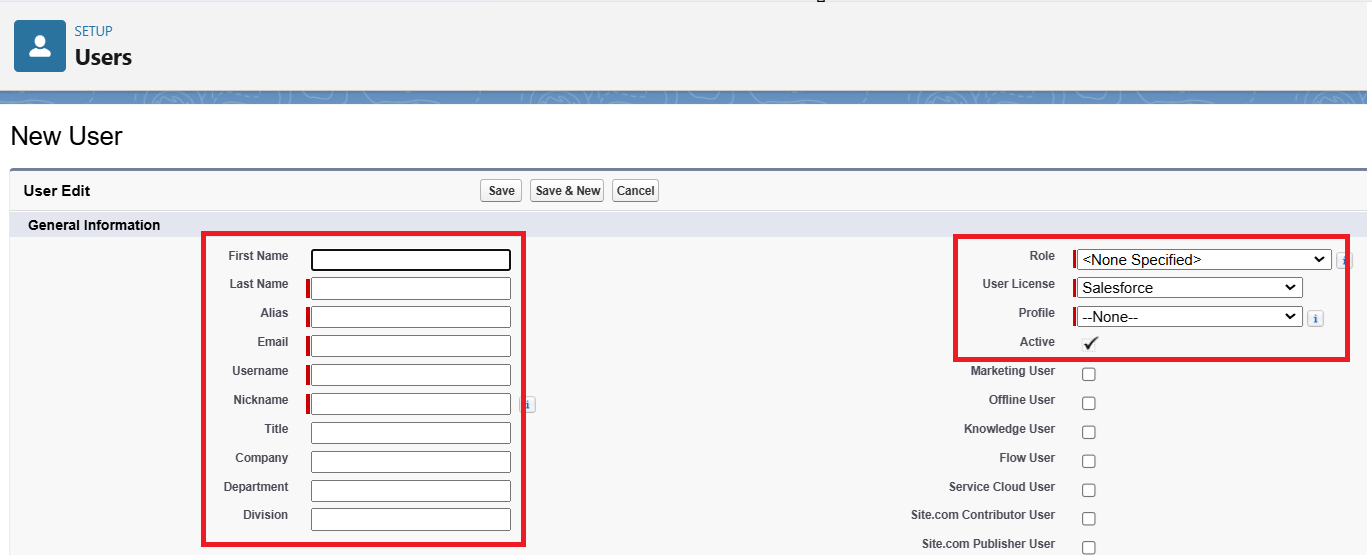
Refer the below diagram to understand which role reports to which role.



Create remaining two roles : Inventory and Marketing same as above.

**Activity 3: Create User**

1. Go to setup → type users in quick find box → select users → click New user.
2. Fill in the fields
3. First Name : Niklaus
4. Last Name : Mikaelson
5. Alias : Give an Alias Name
6. Email id : Give your Personal Email id
7. Username : Username should be in this form: text@text.text
8. Nick Name : Give a Nickname
9. Role : Sales
10. User license : Salesforce Platform
11. Profiles : Platform 1
12. Save.



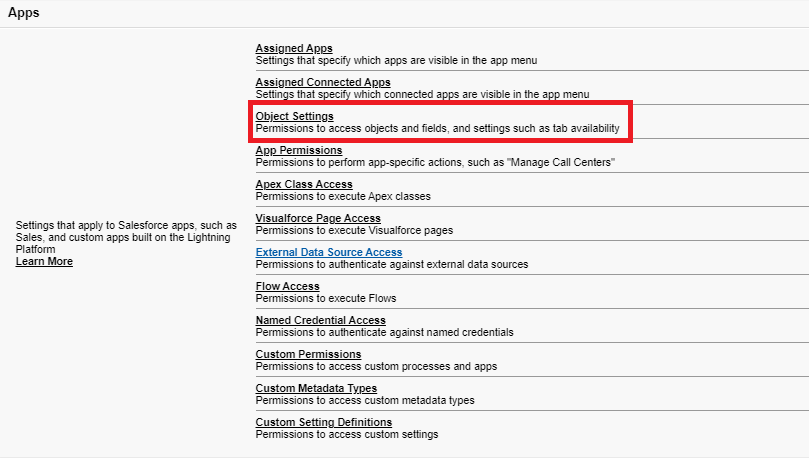
1. Go to setup → type users in quick find box → select users → click New user.
2. Fill in the fields
3. First Name : Kol
4. Last Name : Mikaelson
5. Alias : Give a Alias Name
6. Email id : Give your Personal Email id
7. Username : Username should be in this form: text@text.text
8. Nick Name : Give a Nickname
9. Role : Inventory
10. User license : Salesforce Platform
11. Profiles : Platform 1
12. Save.

**Activity 3: Permission Set**

1. Go to setup → type “permission sets” in quick search → select permission sets → New.



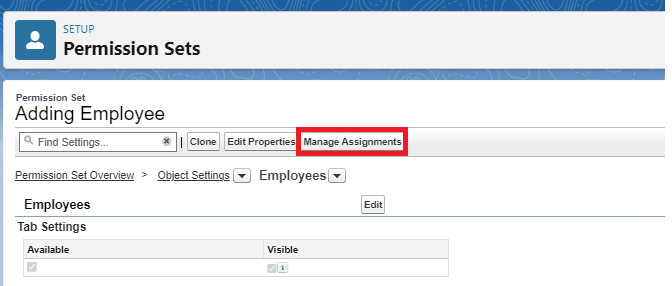
1. Enter the label name as “Permission\_Platform\_1” → save.
2. Under Apps Select object settings.



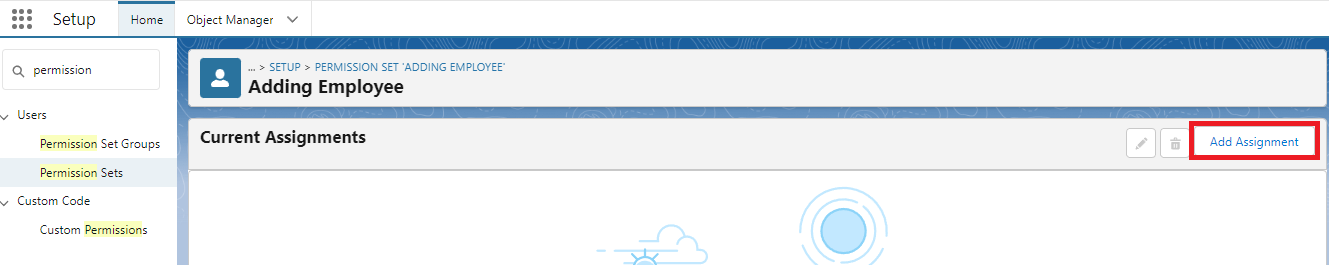
1. Click on Customer object → click on Edit → under object permission check for read, create, edit, delete for HandsMen Customer object.



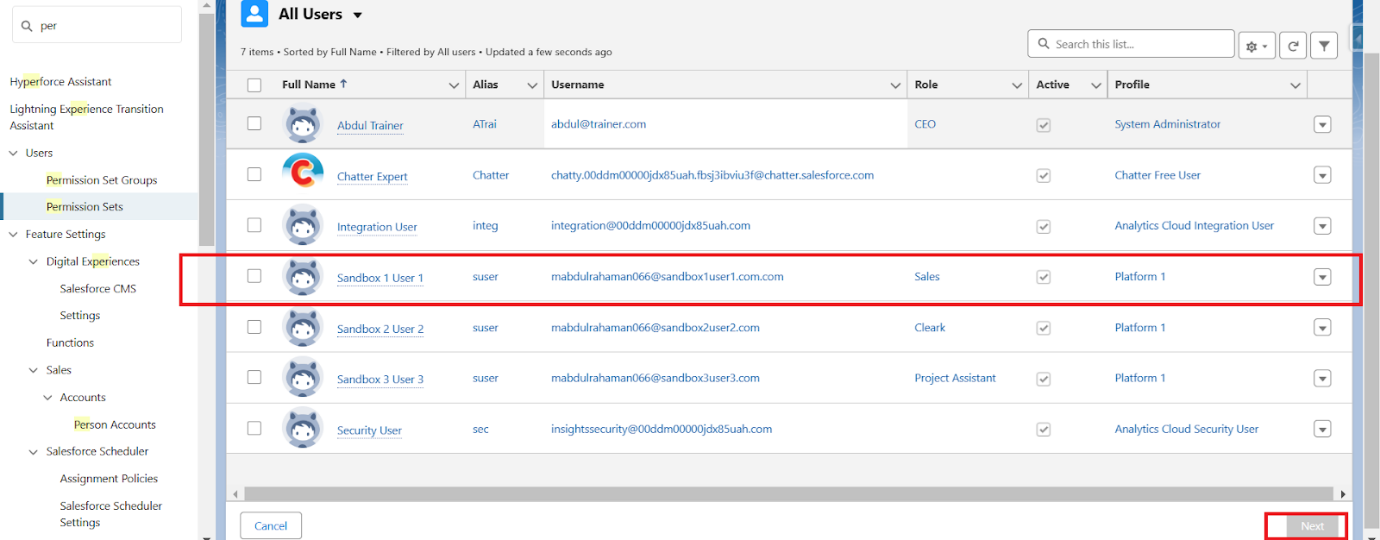
1. Click on Save.
2. Repeat from step 3 and give permission for read, create, edit and delete on order object.
3. After saving the permission click on the Manage assignment



1. Now click on the Manage Assignment.



1. Click on Add Assignment.



1. Now select the users(any one user with the profile “Platform 1”) and click on Next.
2. Click on Assign
3. Click on Done.

## MILESTONE : 8 EMAIL ALERT

**Description**

Creating the following email templates:

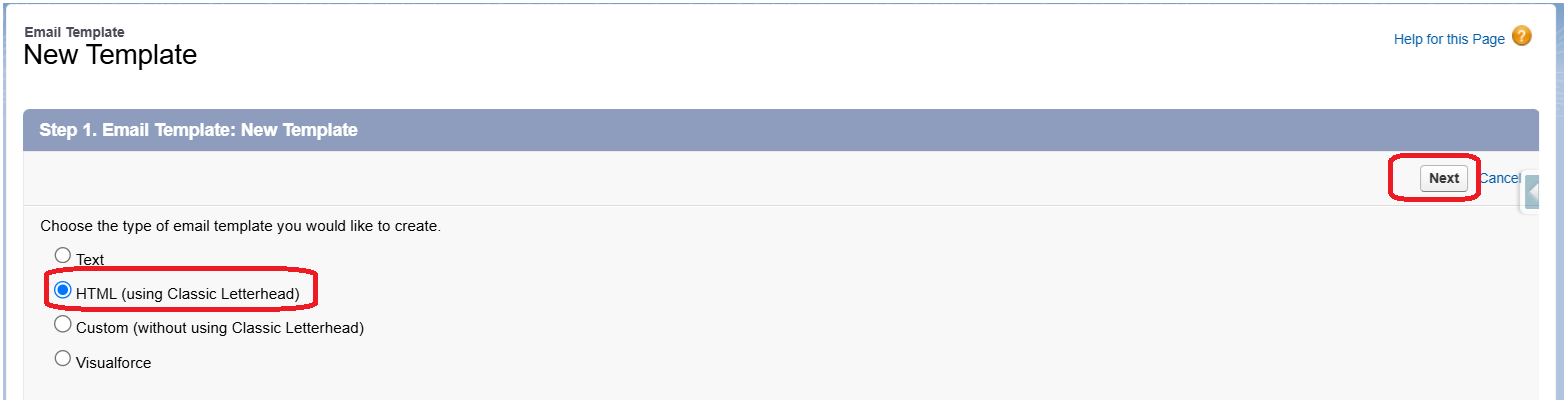
|  |  |  |
| --- | --- | --- |
| **Template Name** | **Type** | **Trigger** |
| **Order Confirmation** | HTML | Sent when Order is placed |
| **Low Stock Alert** | Text | Sent when Inventory\_\_c.Stock\_Quantity\_\_c < 5 |
| **Loyalty Program Email** | HTML | Sent when customer qualifies for loyalty rewards |

**Activity 1: Create an Order Confirmation Email Template**

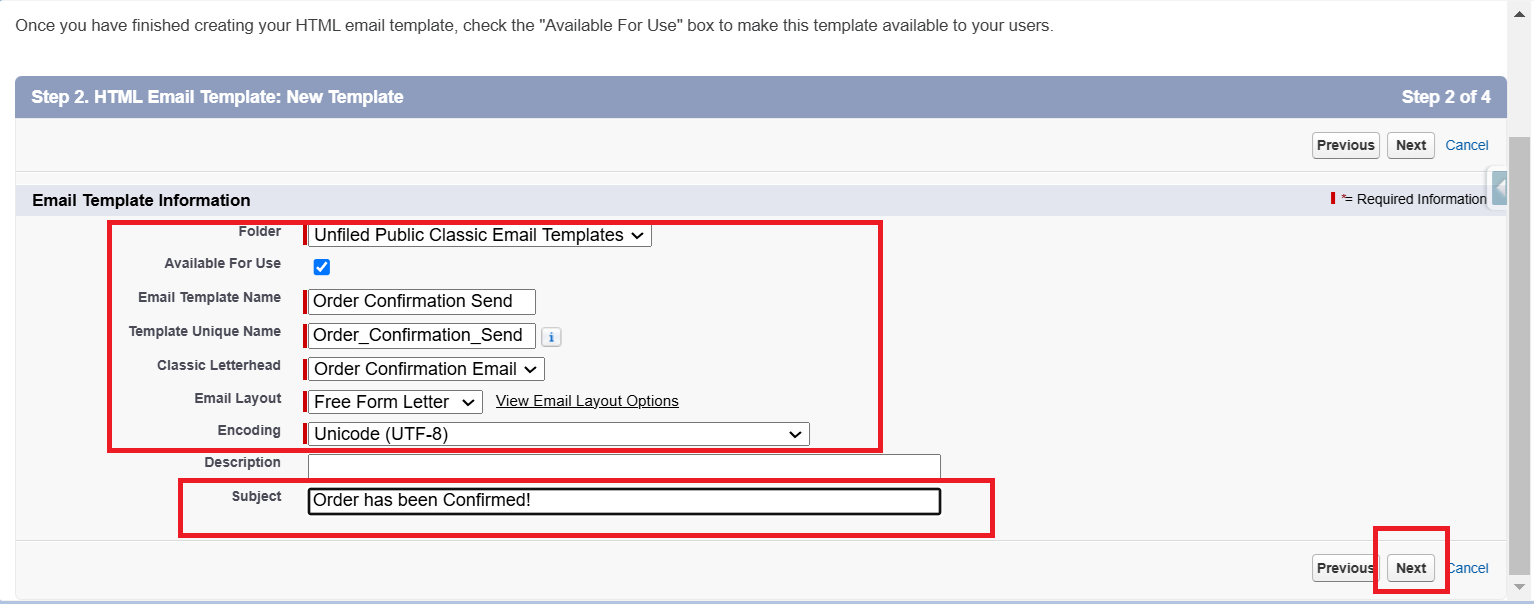
1. Steps to Create a Classic Email Template
2. Go to Salesforce Setup
3. Click on the Gear Icon (⚙) in the top-right corner and select Setup.
4. Navigate to Classic Email Templates
5. In Quick Find, search for Classic Email Templates and click on it.
6. Click "New Template"



1. Choose Text, HTML (with Classic Letterhead), Custom (without Classic Letterhead), or Visualforce.
2. Select HTML (with Classic Letterhead) for a formatted email**.**



1. Fill in Template Details
2. Folder: Select "Unfiled Public Email Templates" (or create a new folder).
3. Available for Use: ✅ (Check this box)
4. Email Template Name: Order\_Confirmation\_Email (or appropriate name).
5. Encoding: UTF-8 (default).
6. Subject: Your Order has been Confirmed!



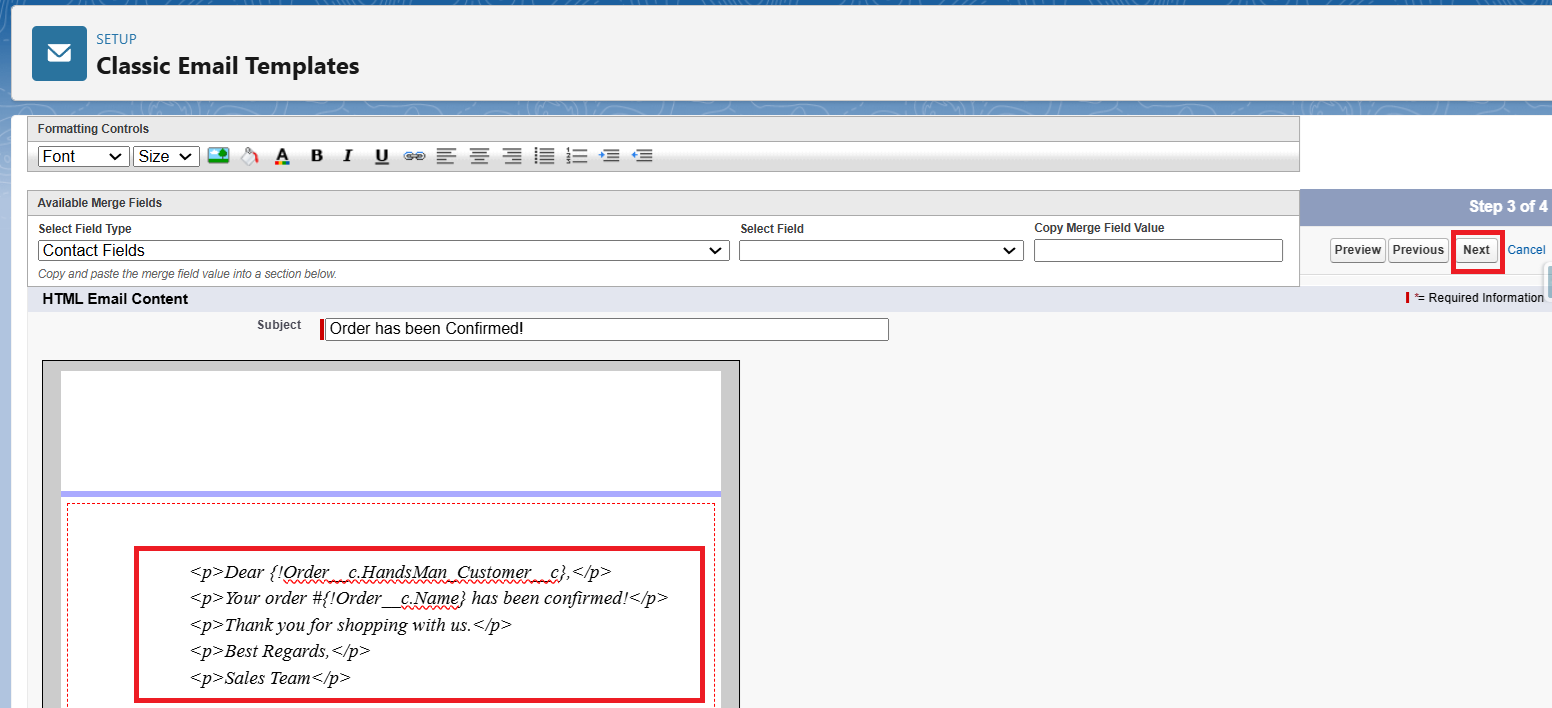
**HTML Body:  
  
*<p>Dear {!Order\_\_c.Customer\_\_c},</p>***

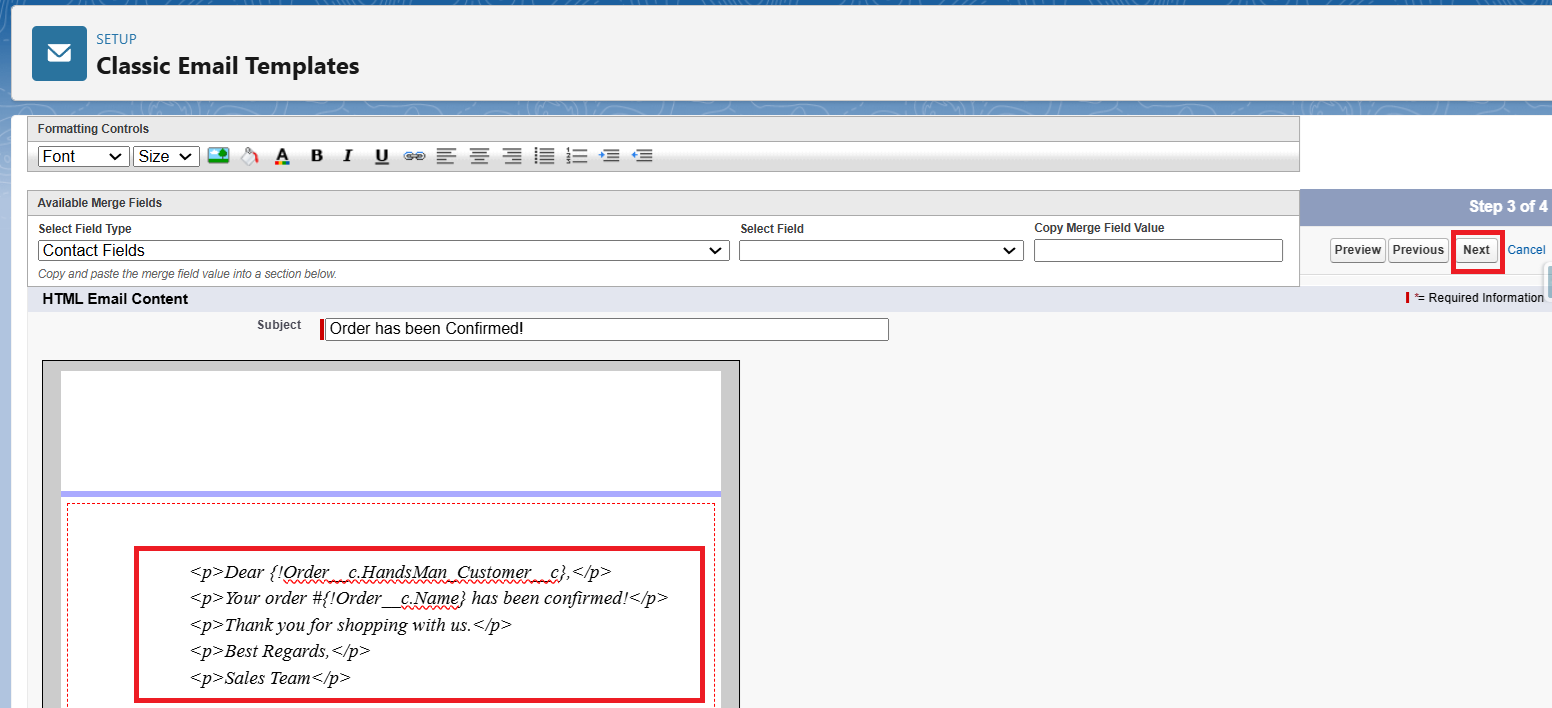
***<p>Your order #{!Order\_\_***[***c.Name***](http://c.name/)***} has been confirmed!</p>***

***<p>Thank you for shopping with us.</p>***

***<p>Best Regards,</p>***

***<p>Sales Team</p>***

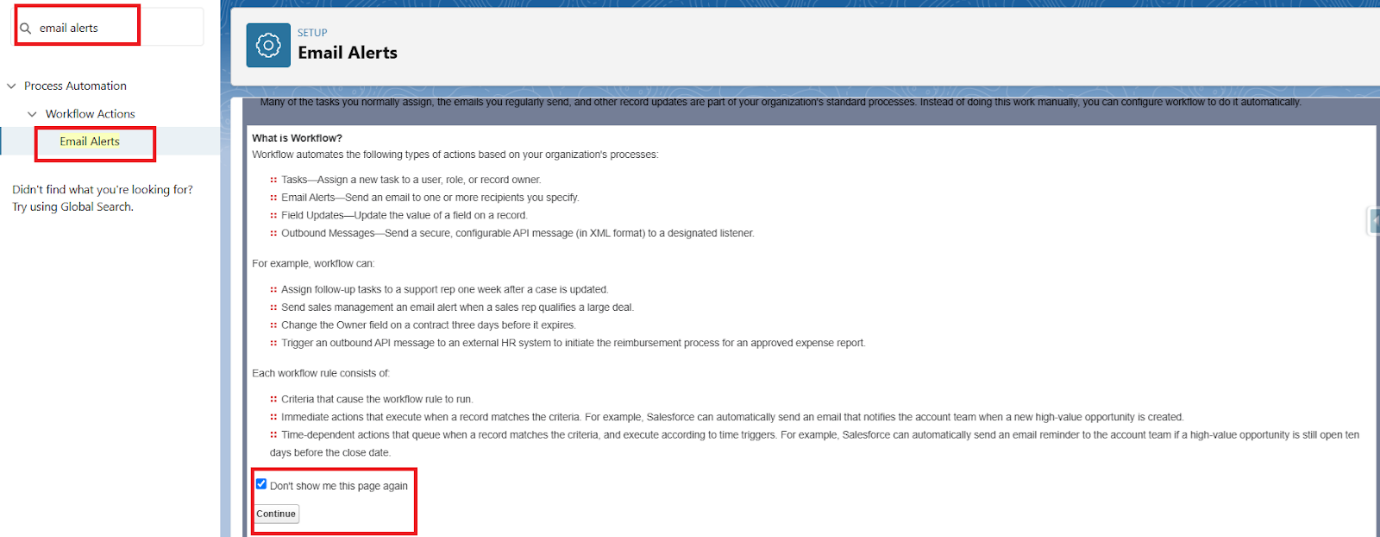




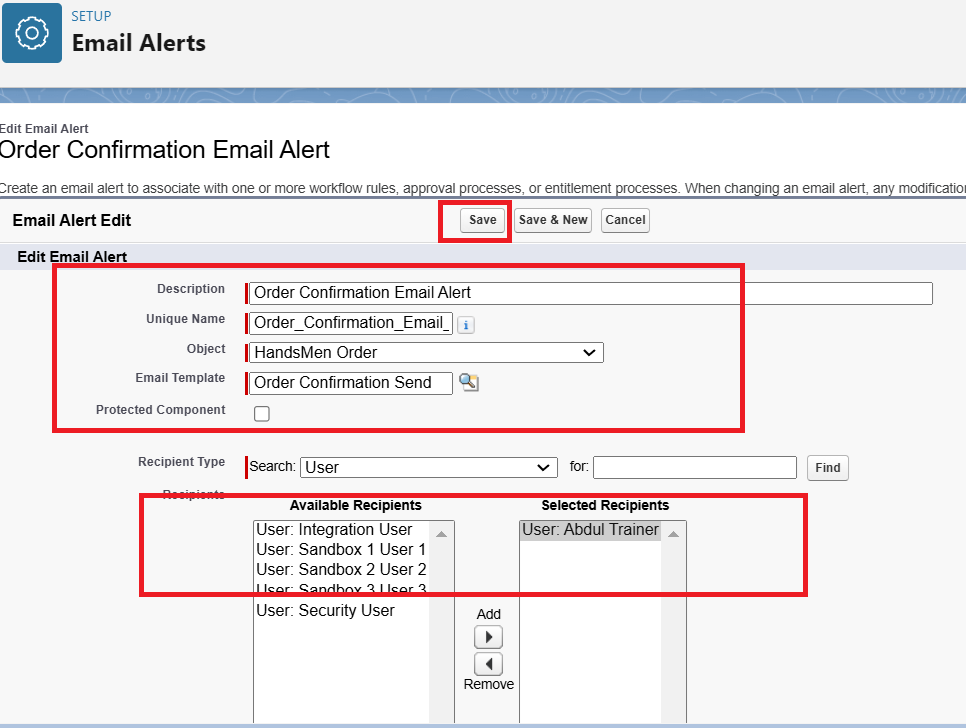
**Save the Template.**

**Activity 2:  
Create an email alert to send an email when an order is confirmed.**

1. Steps to Create an Email Alert
2. Go to Setup
3. In Quick Find, search for Email Alerts and click on it.



1. Click "New Email Alert"
2. Description: Order Confirmation Email Alert
3. Object: Order\_\_c
4. Email Template: Select the one created earlier.
5. Recipient Type: Select Related Record → Customer\_\_c
6. Save the email alert.



## MILESTONE : 9 FLOWS

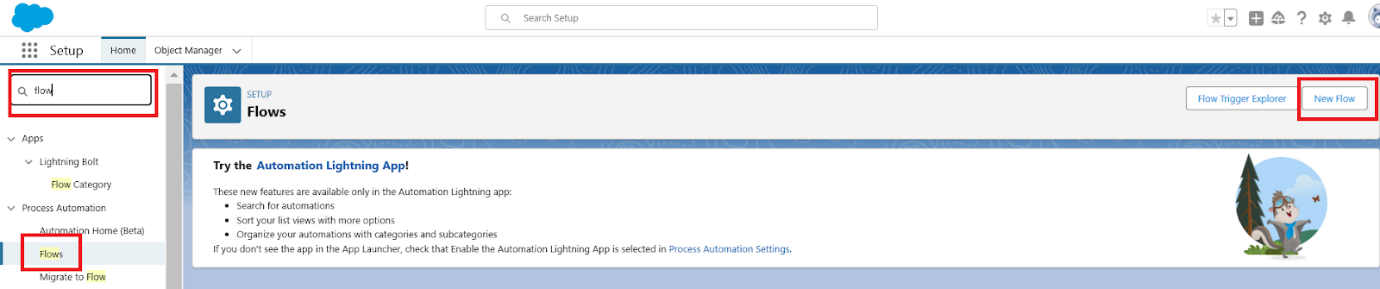
**Create Record-Triggered Flows**

**Flow Summary**

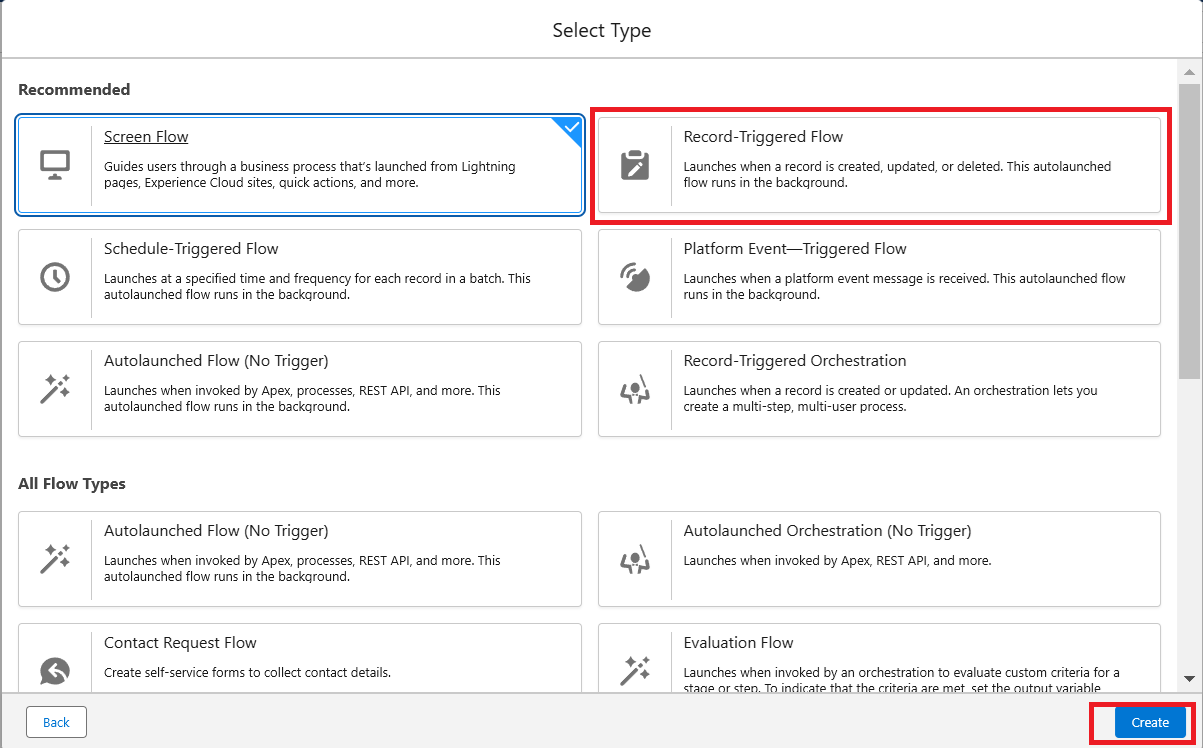


**Activity 1: Order Confirmation Email (Record-Triggered)**

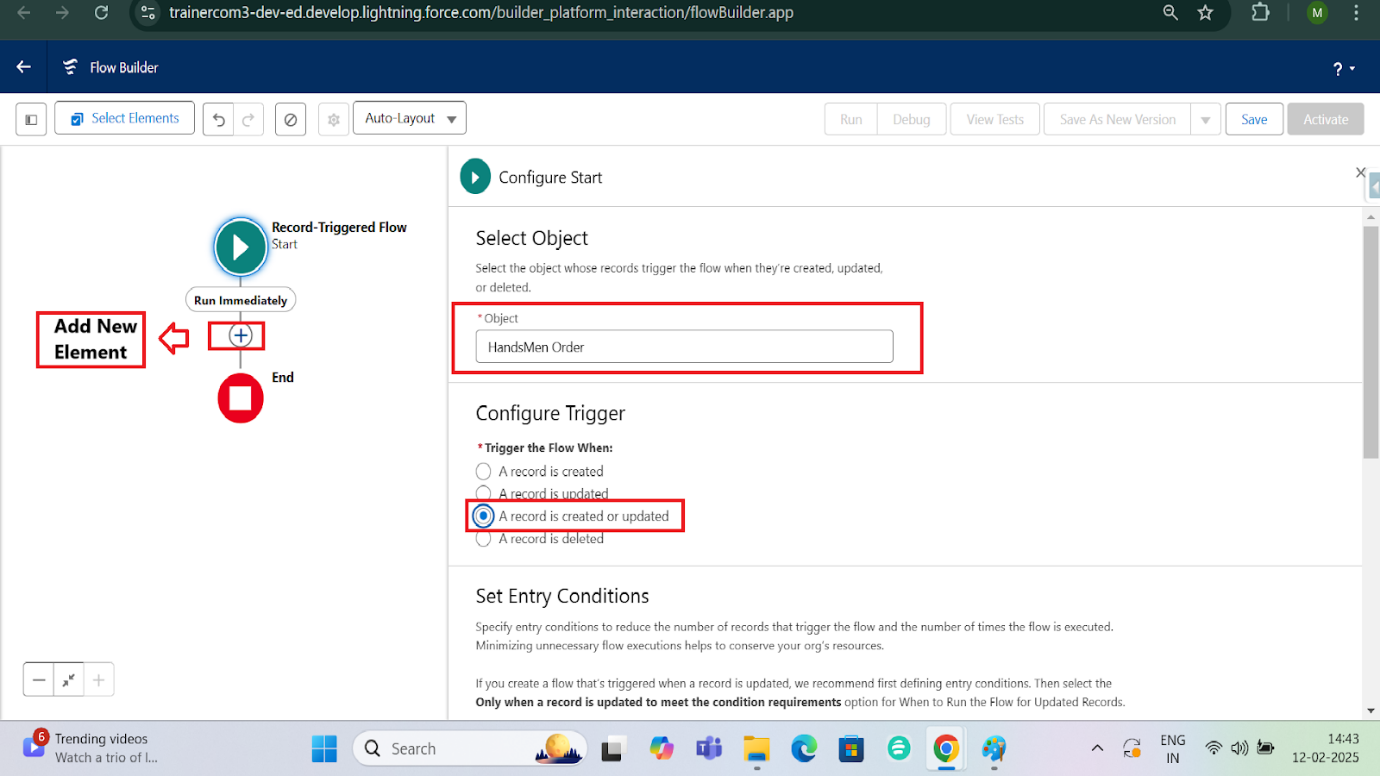
1. Go to Setup → Flow
2. In Quick Find, search for Flows and click on it.



1. Click New Flow → Select Record-Triggered Flow → Click Create.



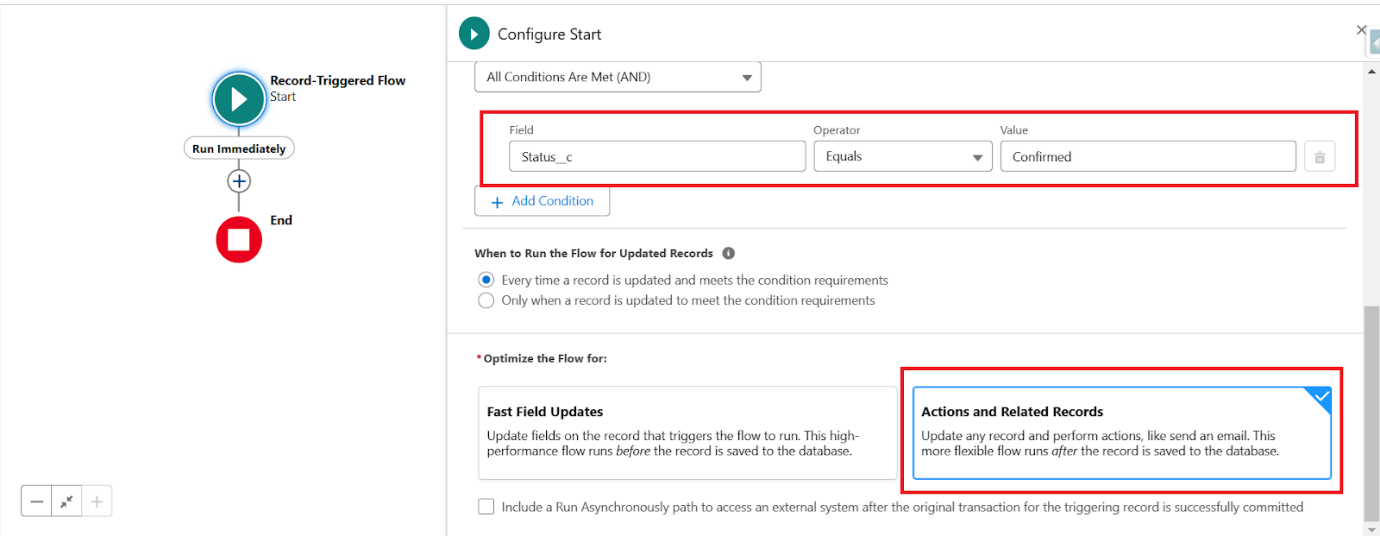
1. Set Flow Trigger Details
2. Object: Order\_\_c
3. Trigger: When a record is updated



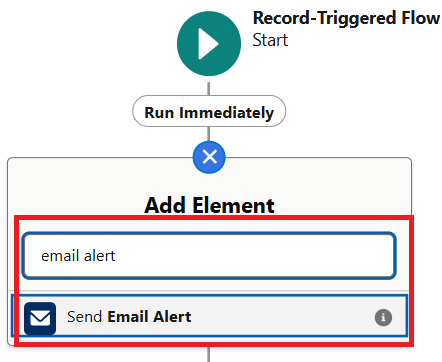
1. Condition:

Field: Order\_\_c.Status\_\_c = "Confirmed"

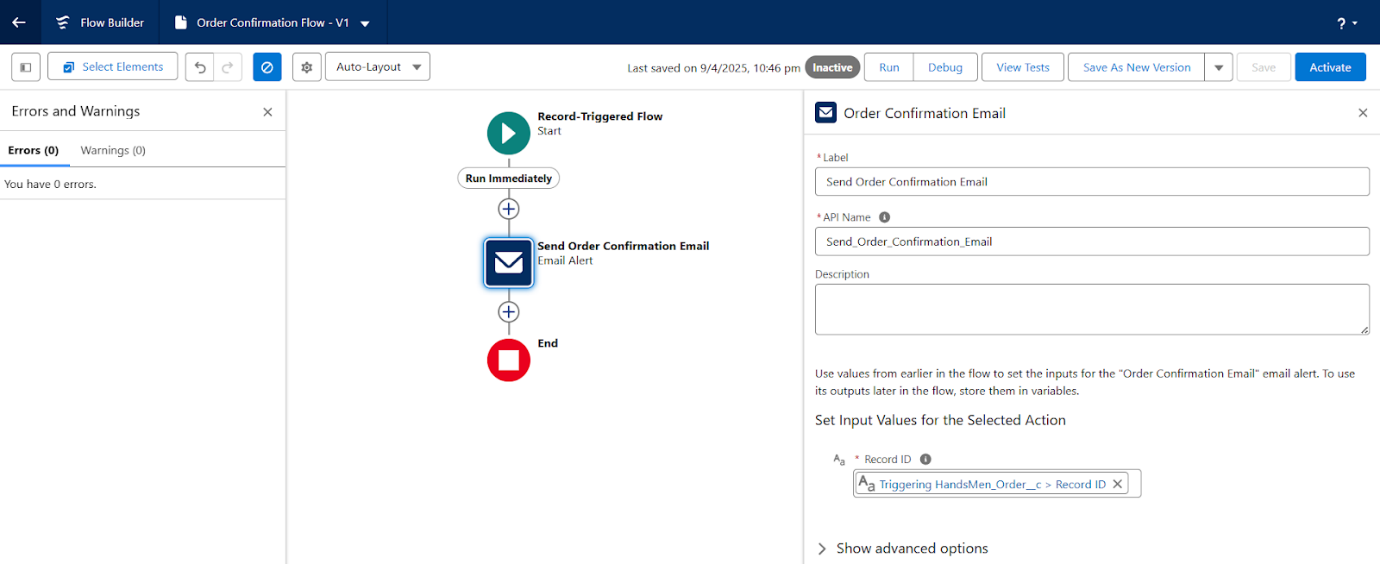
Select Only when a record is updated to meet the condition



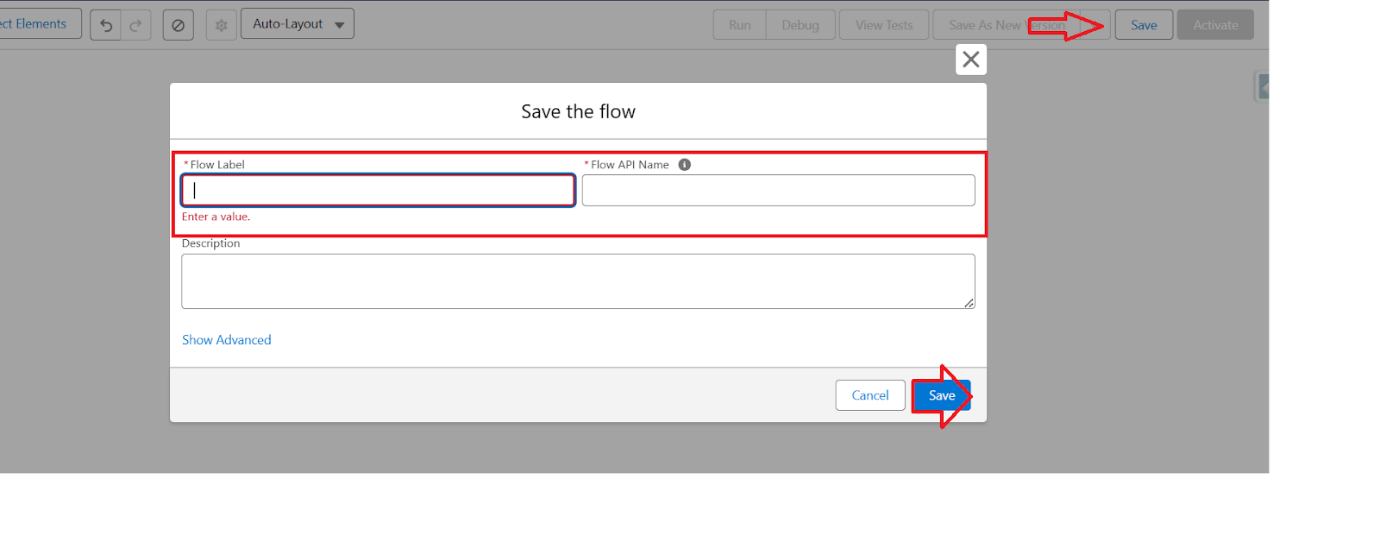
1. Click Done.
2. Add an "Action" Element
3. Click the "+" icon → Select Action.
4. Action Type: Send Email Alert



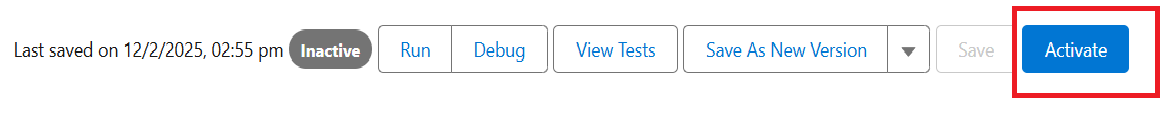
1. Email Alert: Select Order Confirmation Email Alert
2. Label : Send Order Confirmation Email
3. Record ID : {!$[Record.Id](http://record.id/)}
4. Click Save.



1. Save & Activate the Flow
2. Name: Order Confirmation Flow



1. Click Save → Activate.



Activity 2: **Create Stock Alert Email (Record-Triggered)**

1. Go to Setup → Flows → New Flow
2. Select Record-Triggered Flow → Click Create.
3. Set Flow Trigger Details
4. Object: Inventory\_\_c
5. Trigger: When a record is created or updated
6. Condition:
7. Field: Stock\_Quantity\_\_c < 5
8. Select: Every time a record is updated and meets the condition requirements
9. Click Done.
10. Add an "Action" Element
11. Click the "+" icon → Select Action.
12. Action Type: Send Email Alert
13. Create a new Email Alert (Similar to the Order Confirmation setup).
14. Recipient: Inventory Manager.
15. Save & Activate the Flow
16. Name: Stock Alert Flow
17. Click Save → Activate.

Activity : 3 **Loyalty Status Update (Scheduled Flow)**

1.Go to Setup → Flows → New Flow

* Select Schedule-Triggered Flow → Click Create.

2.Set Schedule Frequency

* Set Start Date & Time: Choose Time to run daily.
* Frequency: Select Daily.
* Click Done.

Add "Get Records" Element

* Click the "+" icon → Select Get Records.
* Object: HandsMen\_Customer\_\_c
* Filter: Retrieve all records.
* Sort Order: None.
* Click Done.

Add "Loop" Element

* Click the "+" icon → Select Loop.
* Collection: {!Get\_Records}
* Direction: First to Last.
* Click Done.

Inside Loop -

* Click the "+" inside the loop → Select Decision.

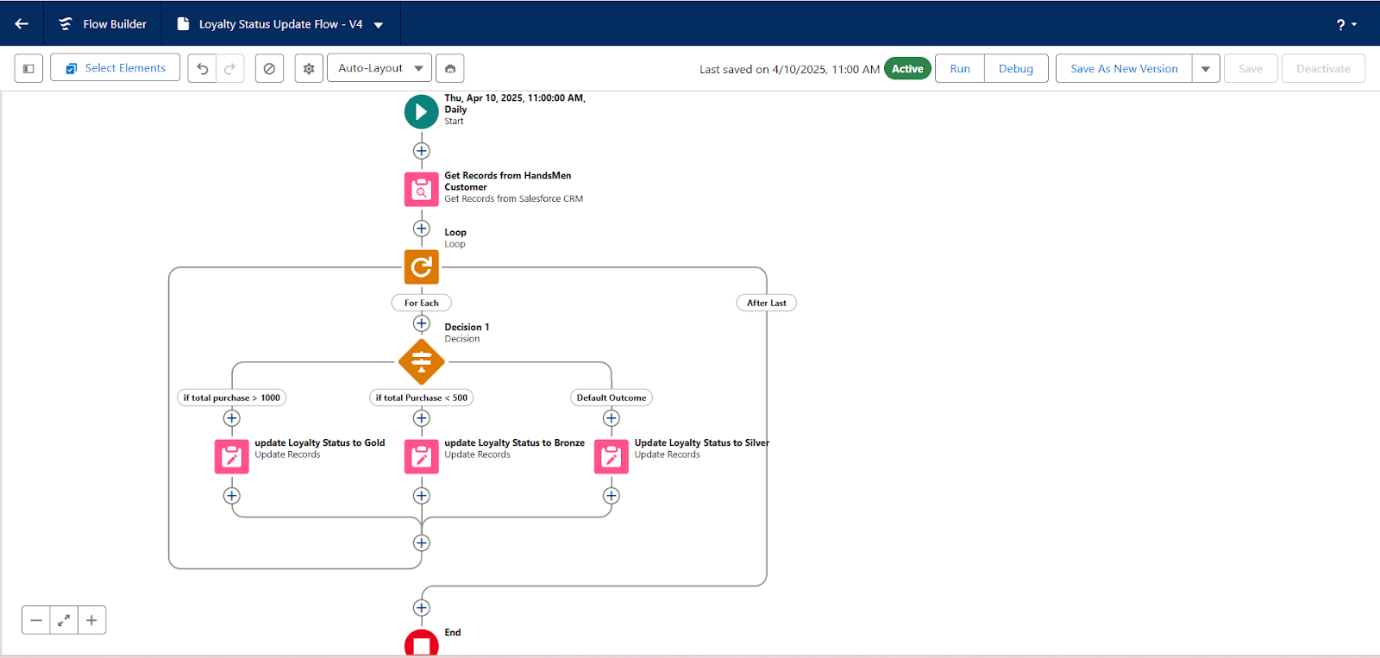
Set Conditions:

* If Total\_Purchases\_\_c > 1000, Set Loyalty\_Status\_\_c = Gold.
* Click the "+" Add Update Records
* Records to Update: Select Specify Condition
* Object: HandsMen Customer
* Set Field Values: Loyalty\_Status\_\_c = Gold
* Else if Total\_Purchases\_\_c < 500,
* Set Loyalty\_Status\_\_c = Bronze.
* Else,  (Default Outcome) Set Loyalty\_Status\_\_c =Silver.
* Click Done.

Save & Activate the Flow

Name: Loyalty Status Update Flow

Click Save → Activate.



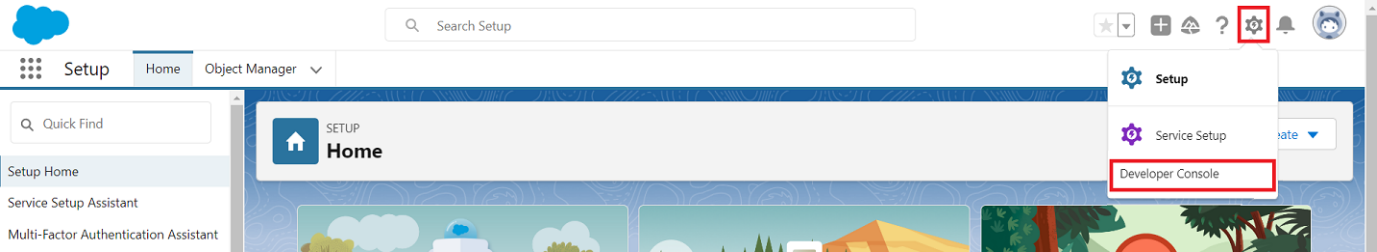
## MILESTONE : 10 APEX TRIGGERS

**Apex Triggers**

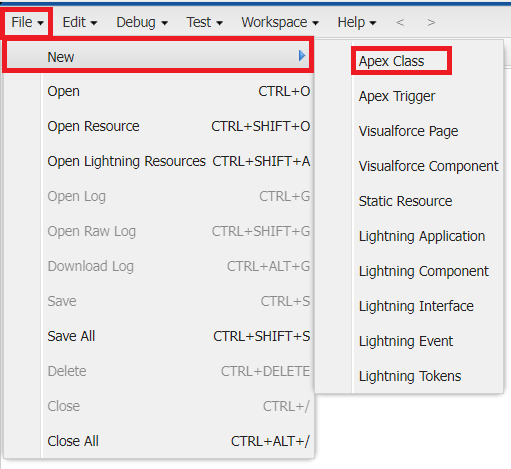
|  |  |  |
| --- | --- | --- |
| **Trigger Name** | **Object** | **Purpose** |
| **Update Order Total** | Order\_\_c | Auto-update Total\_Amount\_\_c on order save |
| **Stock Deduction** | Inventory\_\_c | Reduce stock when an order is placed |
| **Loyalty Status Update** | Customer\_\_c | Upgrade loyalty status based on purchases |

**Activity 1: Create Apex Class**

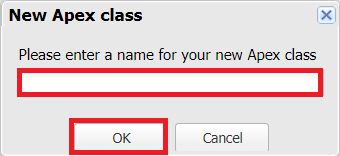
1. **Create an Apex Class**
2. **Go to Setup → Click on the gear icon → Select Developer Console.**



1. **Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.**
2. **To create a new Apex Class follow the below steps:   
   Click on the file → New → Apex Class.**

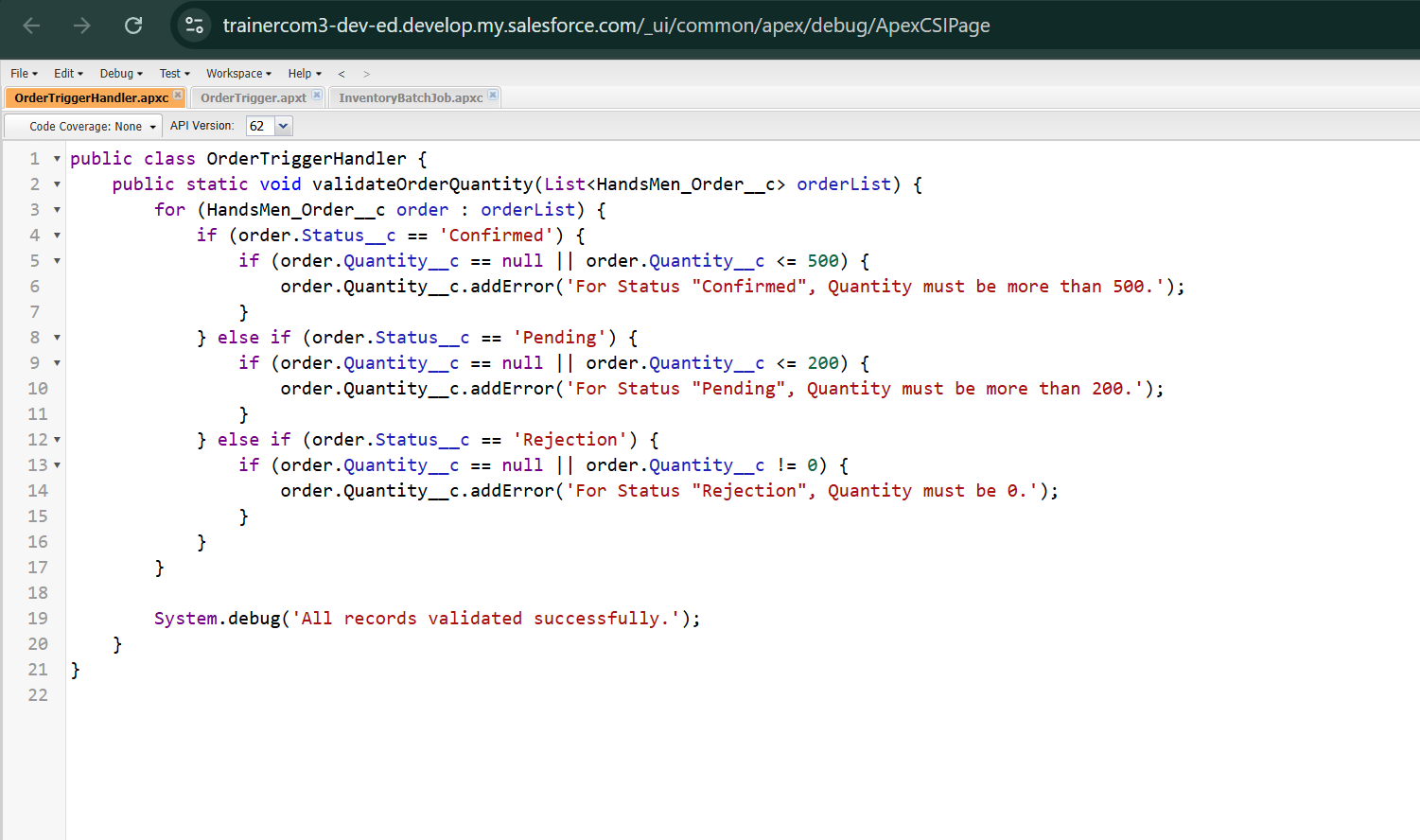


**Give the Apex Class name as “OrderTriggerHandler”.**



**Click ok.**

**Now write the code logic here**



**Source Code:**

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.');

    }

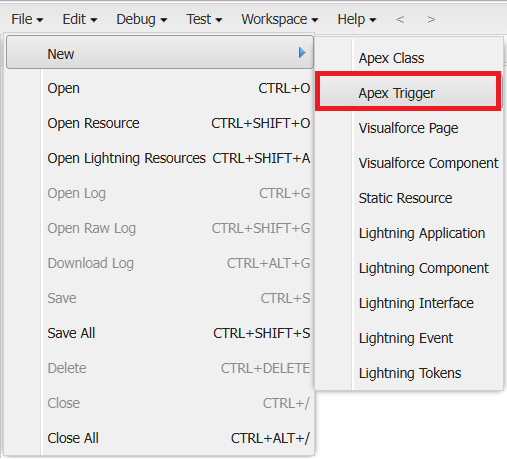
}

Save the code.(click on file → Save)

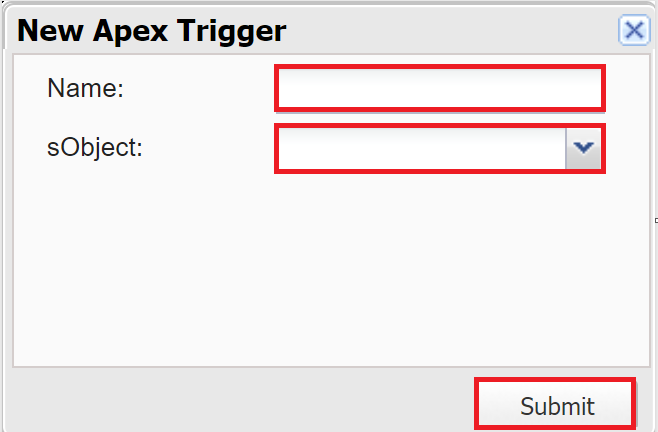
**Activity 2: Create an Apex Trigger**

To create a new Apex Trigger follow the below steps:

Click on the file → New → Apex Class.

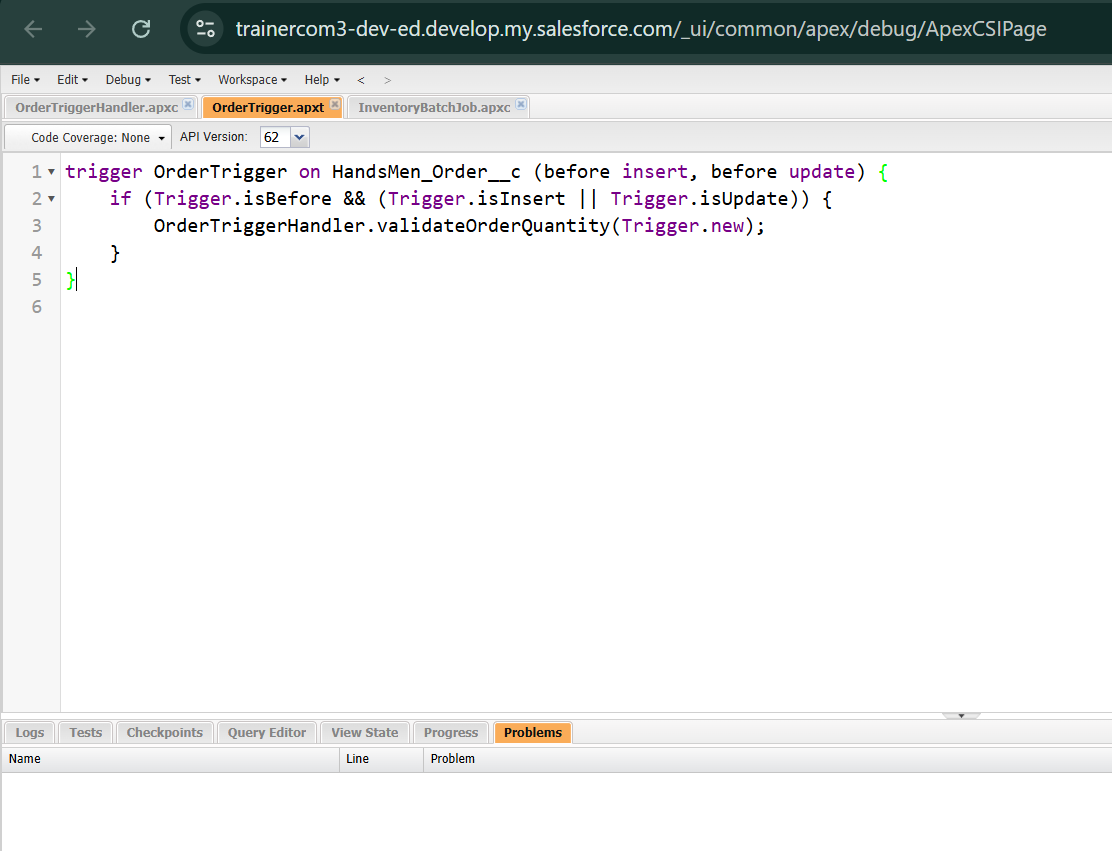


Give the Apex Trigger name as “OrderTrigger”, and select “HandsMen\_Order\_\_c” from the dropdown for sObject.



Click Submit.

Now write the code logic here



**Source Code:**

trigger OrderTrigger on HandsMen\_Order\_\_c (before insert, before update) {

    if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {

        OrderTriggerHandler.validateOrderQuantity([Trigger.new](http://trigger.new/));

    }

}

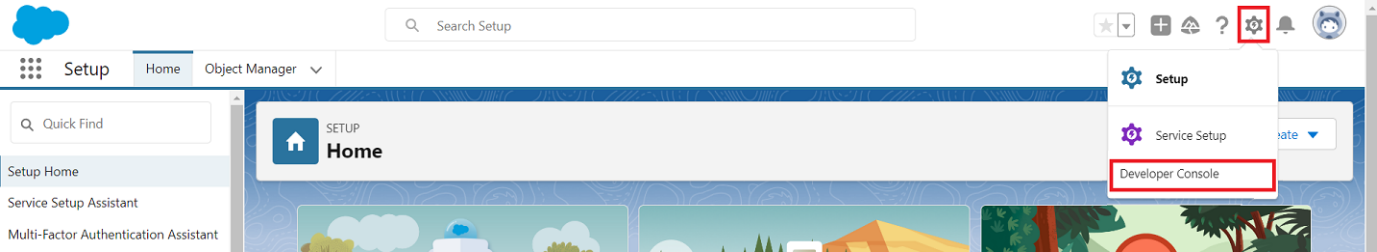
Save the code.(click on file → Save)

## MILESTONE : 11 BATCH JOBS

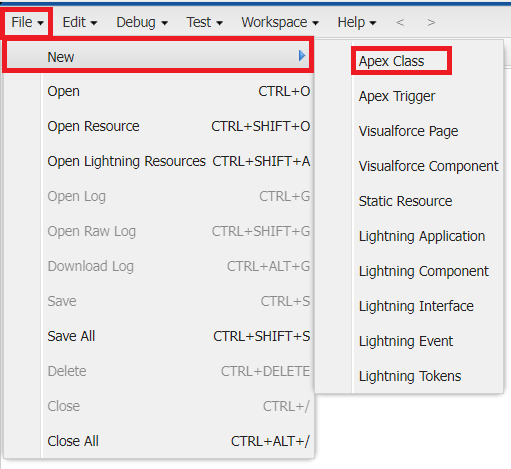
|  |  |  |
| --- | --- | --- |
| **Batch Job** | **Purpose** | **Schedule** |
| **Loyalty Points Calculation** | Updates customer points weekly | Every Sunday 12 AM |
| **Inventory Sync** | Syncs stock levels with external warehouse system | Daily at 2 AM |

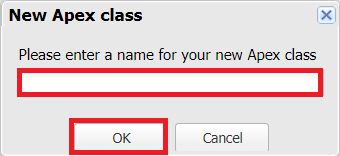
**Activity 1: Create Batch Apex**

1. **Create an Apex Class**
2. **Go to Setup → Click on the gear icon → Select Developer Console.**



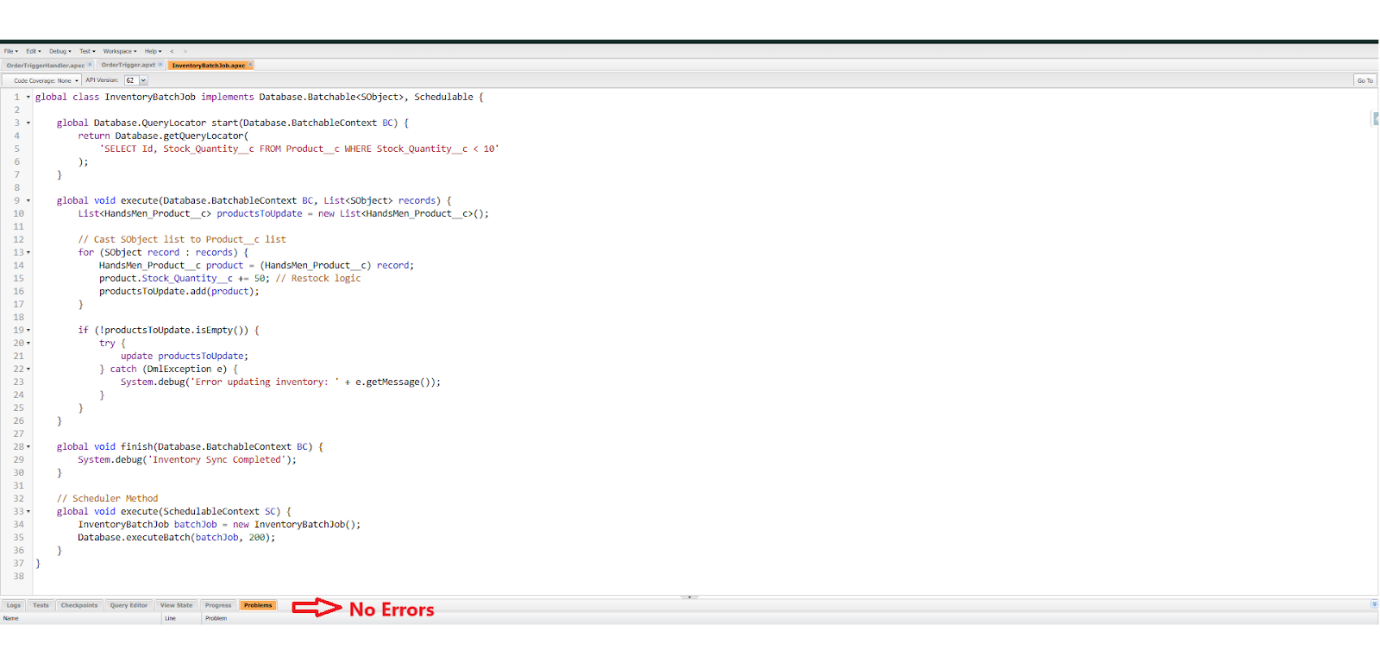
1. **Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.**
2. **To create a new Apex Class follow the below steps:   
   Click on the file → New → Apex Class.**



**Give the Apex Class name as “InventoryBatchJob”.**

**Click ok.**

**Now write the code logic here**



**Source Code:**

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);

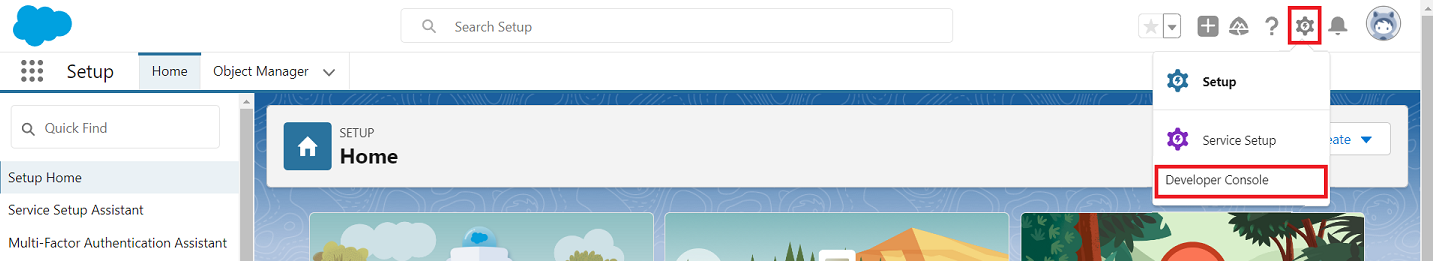
}

}

Save the code.(click on file → Save)

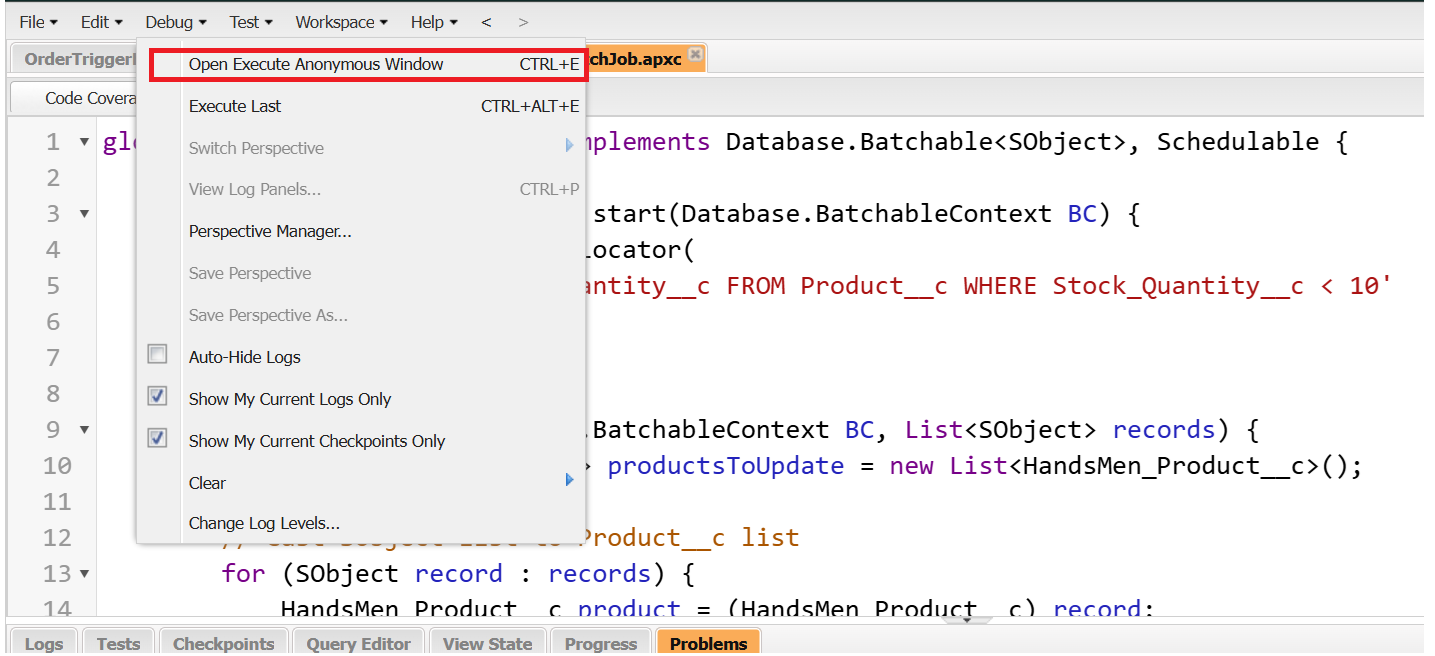
**Activity 2 :**

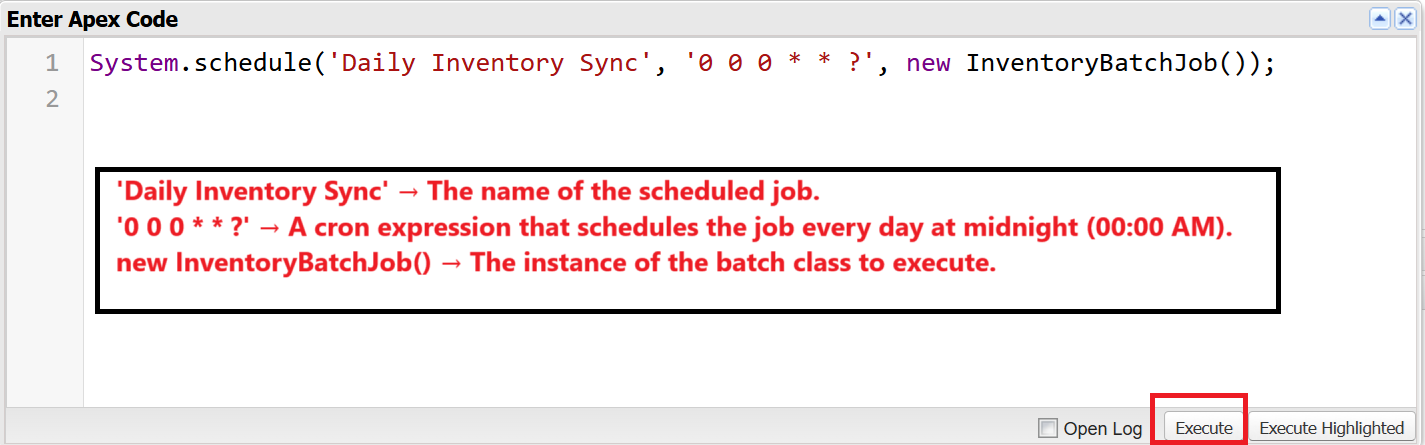
Go to Setup → Click on the gear icon → Select Developer Console.



Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.

To create a new Apex Class follow the below steps:   
Click on the file → Debug → Open Execute Anonymous Window



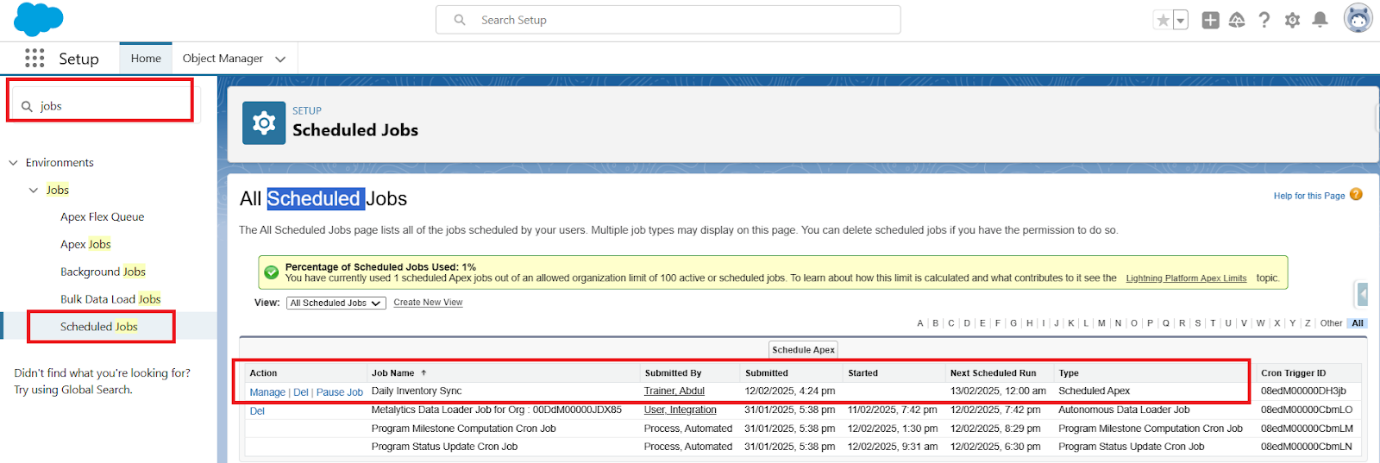


**Source Code :** System.schedule(‘Daily Inventory Sync’ , ‘0 0 0 \* \* ?’ , new InventoryBatchJob());

**Schedule the Apex class:**

From the Setup page search for “Jobs” in quick search.

Click on “Schedule Jobs” as shown below.



MILESTONE : 12 PROJECT DOCUMENTATION:  
Documentation ensures all objects, workflows, profiles, roles, and validations are recorded. It guides:

* Future maintenance
* User training
* Audit compliance
* Change management

**1. Data Integrity & Quality**

Data integrity is enforced directly from the **Salesforce UI** using:

* Validation rules to prevent incomplete or incorrect data entry
* Required fields and picklists
* Formula fields for calculated values
* Controlled record access via profiles and permission sets

**2. Automation Strategy**

**2.1 Automated Order Confirmations**

* Triggered after order confirmation
* Sends personalized email to customers
* Improves transparency and customer trust

**2.2 Dynamic Loyalty Program**

* Loyalty status updated based on purchase history
* Implemented using record-triggered flows and Apex
* Enables personalized offers and rewards

**2.3 Proactive Stock Alerts**

* Triggered when inventory drops below five units
* Automatic email sent to warehouse team
* Prevents stockouts and operational delays

**2.4 Scheduled Bulk Order Updates**

* Batch Apex runs daily at midnight
* Processes bulk orders
* Updates financial records and inventory levels
* Ensures accurate daily stock visibility

**3 . Security & Access Control**

* Role hierarchy for organizational access
* Permission sets for functional access
* Sharing rules to control record visibility
* Field-level security to protect sensitive data

**4. Testing Strategy**

* **Unit Testing:** Apex classes and triggers
* **Functional Testing:** Business workflows and automation
* **Integration Testing:** Data flow across objects
* **User Acceptance Testing (UAT):** Business user validation

**5. Deployment Strategy**

* Validate components in sandbox
* Deploy using Change Sets
* Post-deployment verification
* Monitor logs and automation performance

**6. Deliverables**

* Solution Design Document (SDD)
* Object Model & ER Diagram
* Automation Strategy Documentation
* Email Templates
* Apex Classes & Triggers
* Test Results & Deployment Checklist

**7. Learning Outcomes**

Through this project, the following Salesforce skills are gained:

* Data Modelling
* Data Quality Management
* Lightning App Builder
* Record-Triggered Flows
* Apex & Apex Triggers
* Asynchronous Apex (Batch & Scheduled)

**8. Conclusion**

The HandsMen Threads Salesforce implementation delivers a scalable, automated, and secure CRM solution that enhances customer engagement, optimizes inventory management, and ensures data accuracy. The system lays a strong foundation for future enhancements such as analytics, AI-driven personalization, and external integrations.