**CRM Application for Jewel Management - (Developer)**

College Name: Sri Vasavi College (self finance wing), Erode

College Code: bru17

TEAM ID: NM2025TMID23309

TEAM MEMBERS:

Team Leader Name: THAMARAIBHARANI P

Email: thamaraibharani0908@gmail.com

Team Member1: THAMARAI SELVAN R

Email: thamaraiselvant599@gmail.com

Team Member2: VENKATESH N

Email: venkateshvenkatesh17660@gmail.com

Team Member3: VEENA P

Email: hepziveena@gmail.com

Team Member 4: PRABHAKARAN K

Email: k.prabhakaran.2005@gmail.com

INTRODUCTION

Project Overview

The Jewel Inventory System is a comprehensive software Solution designed to streamline and manage the inventory and sales processes of a jewellery store or a jewellery manufacturer.

The system aims to provide an efficient and user-friendly solution to track and control the inventory of various jewellery items, maintain accurate records, and facilitate seamless sales transactions.

Purpose

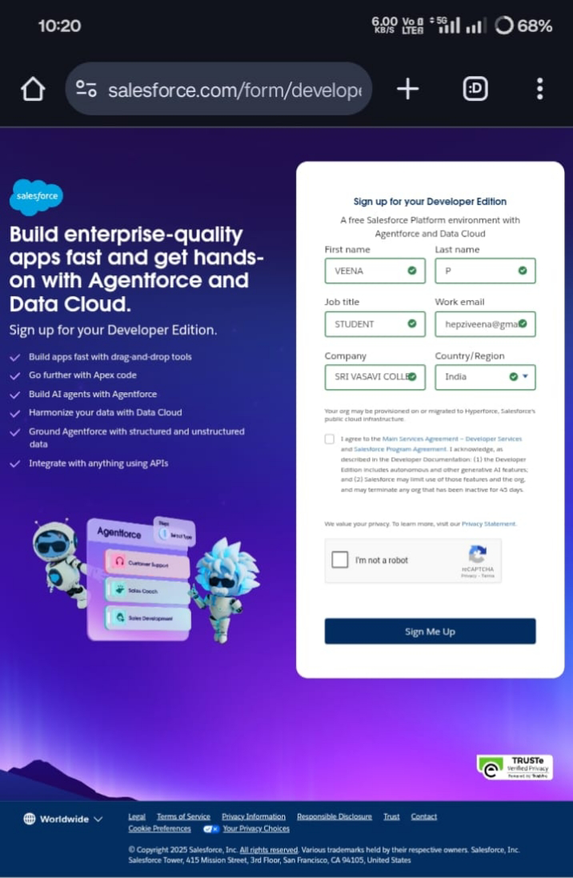
The purpose of the CRM Application for Jewel Management is to help jewelry businesses efficiently manage customer information, track sales and inventory, and improve customer relationships.

It allows for better organization, personalized services, faster communication, and data-driven decisions, ultimately increasing customer satisfaction and business growth.

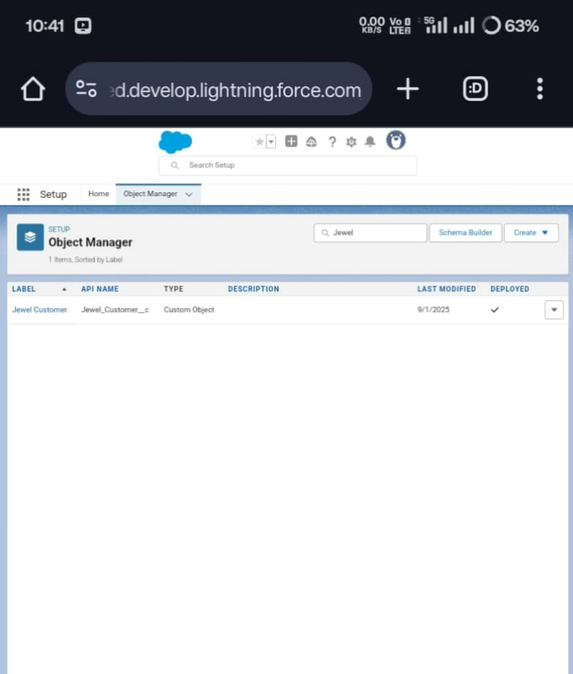
DEVELOPMENT PHASE

Creating Developer Account:

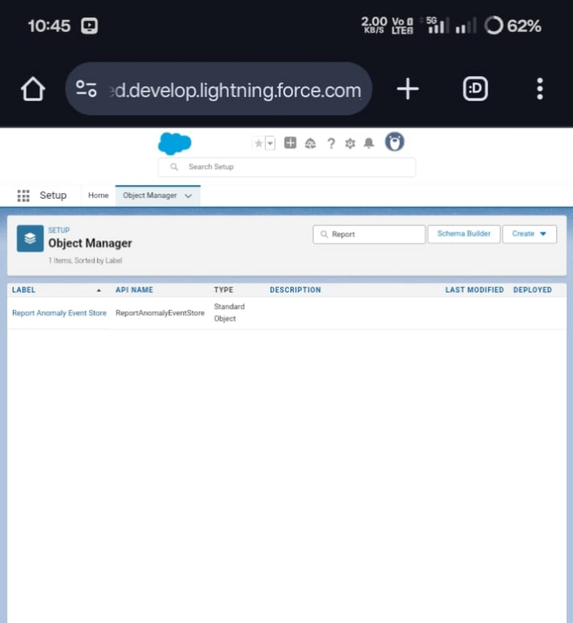
By using this URL - https://developer.salesforce.com/signup



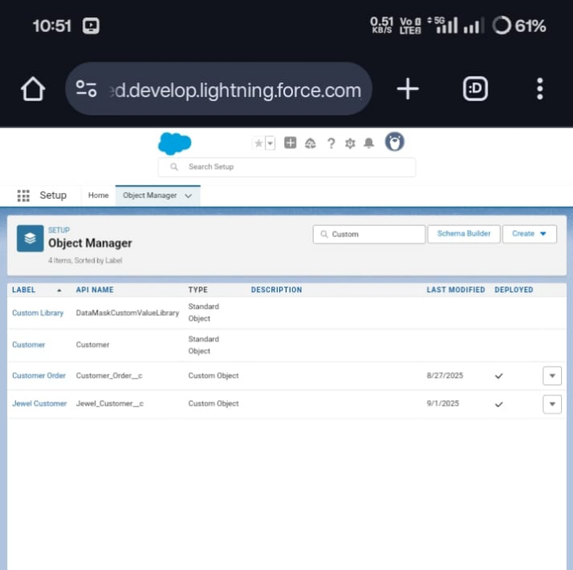




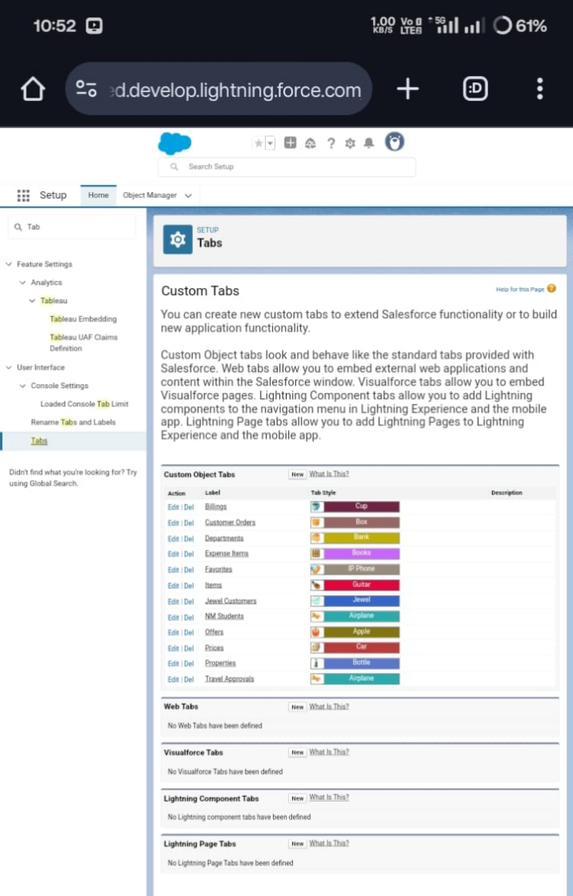
Create Jewel Customer Object

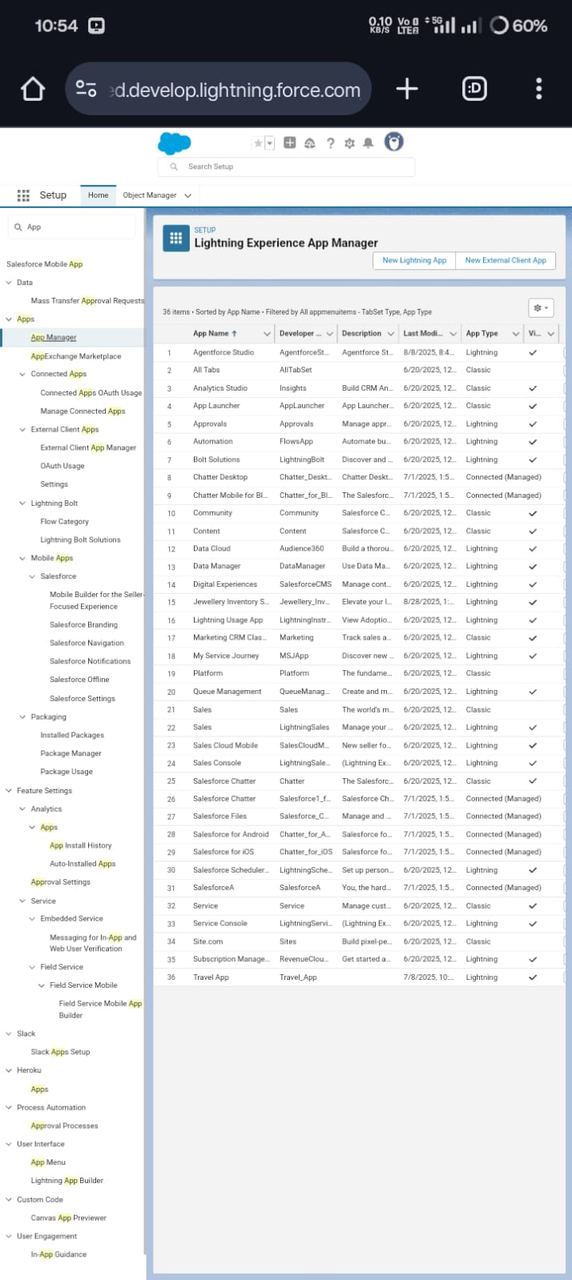


The purpose of creating a Item object is to manage the inventory of gold and silver items.

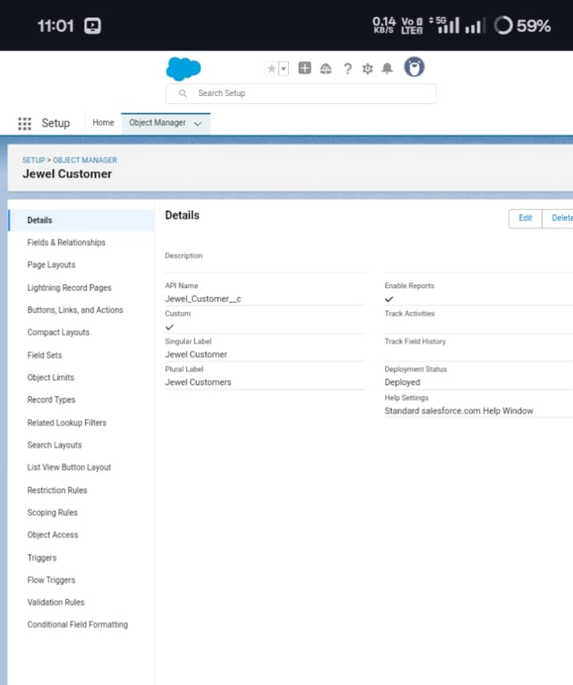


Creating a Custom Tab for an item , customer orders, billing objects.

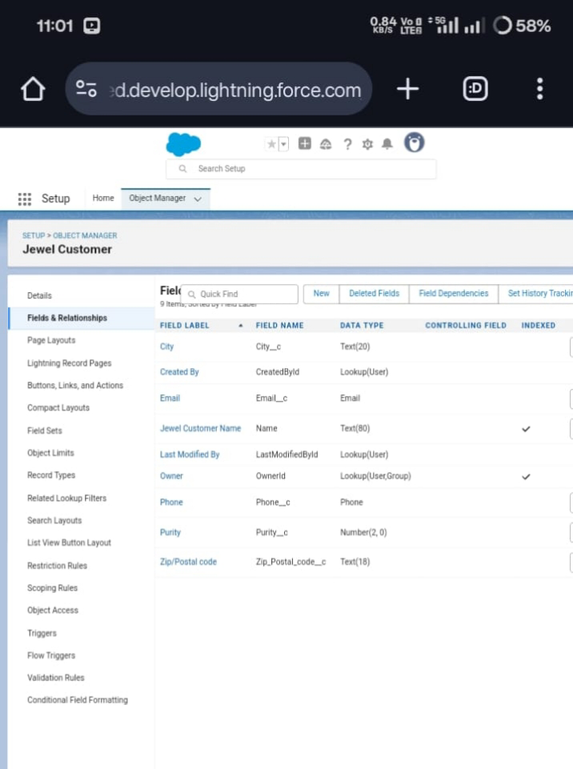




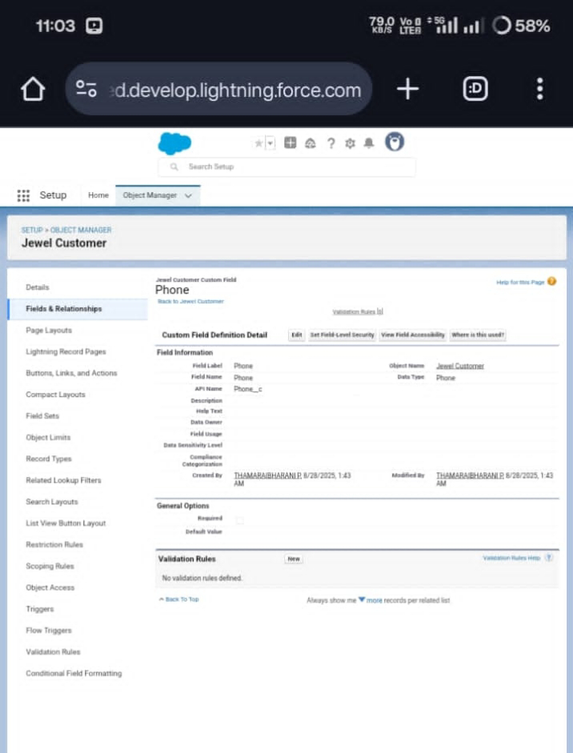
Create a Lightning App : Jewellery inventory system



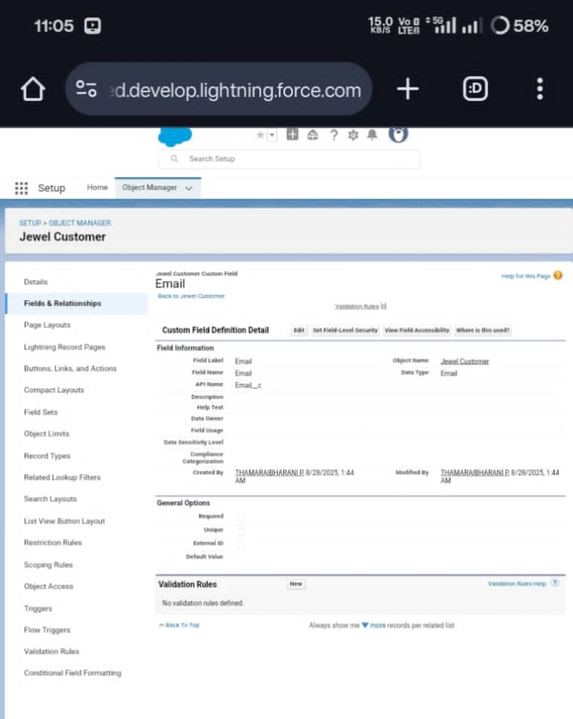
To Create a relationship between Jewel Customer & Customer Order Objects.



Creating the Phone field in object Jewel Customer



Creating the Email field in object Jewel Customer



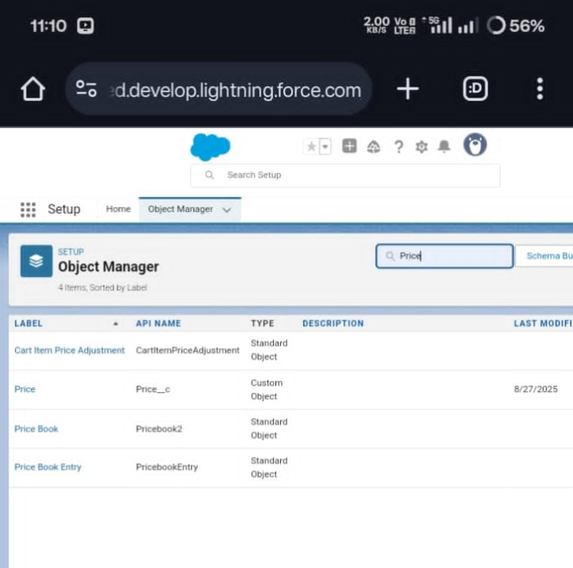
Creating the number field in Item object

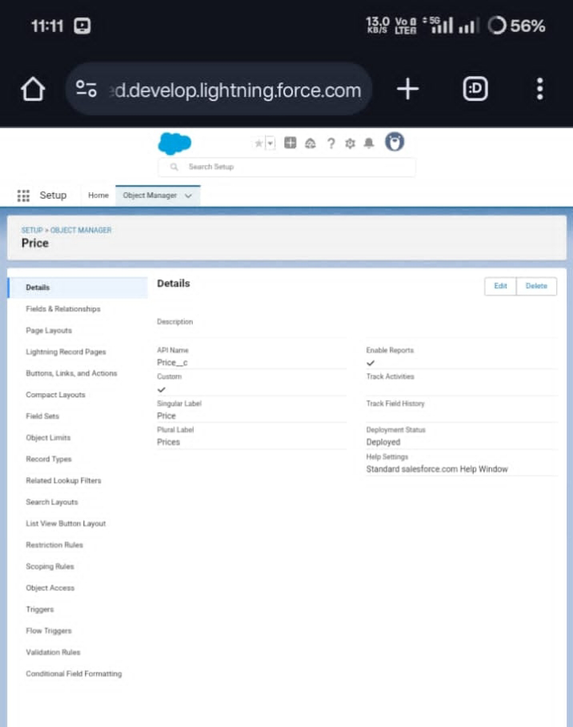


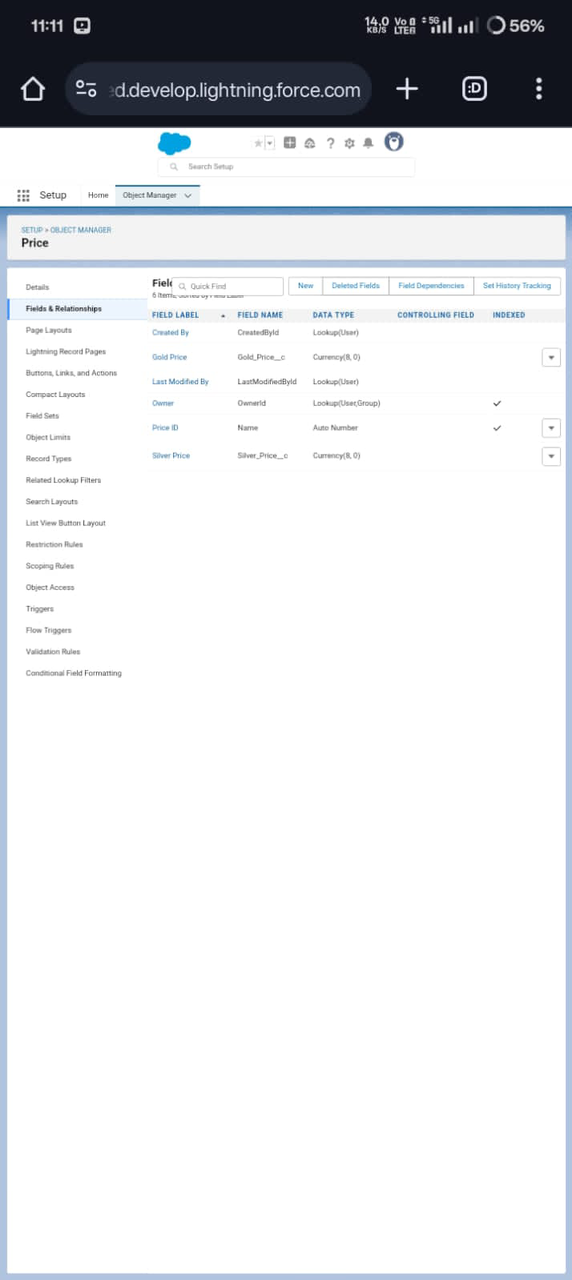
Creating Picklist Field in Item Object



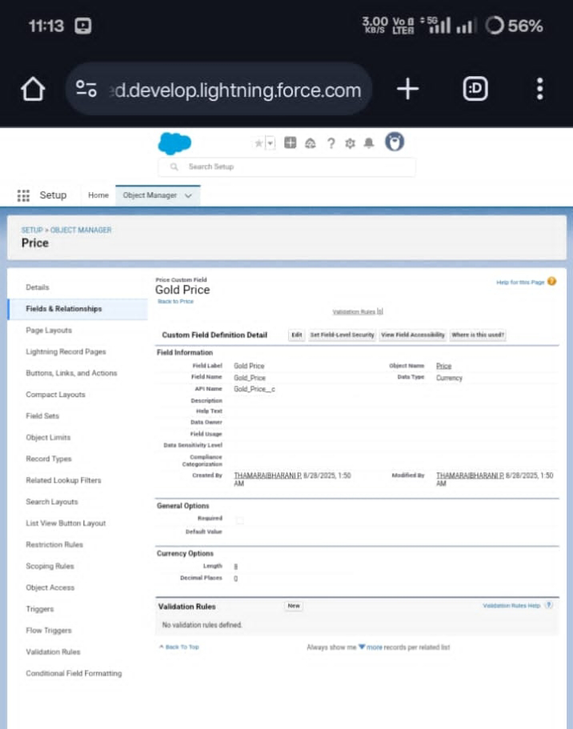
Creating Currency Field in Price Object



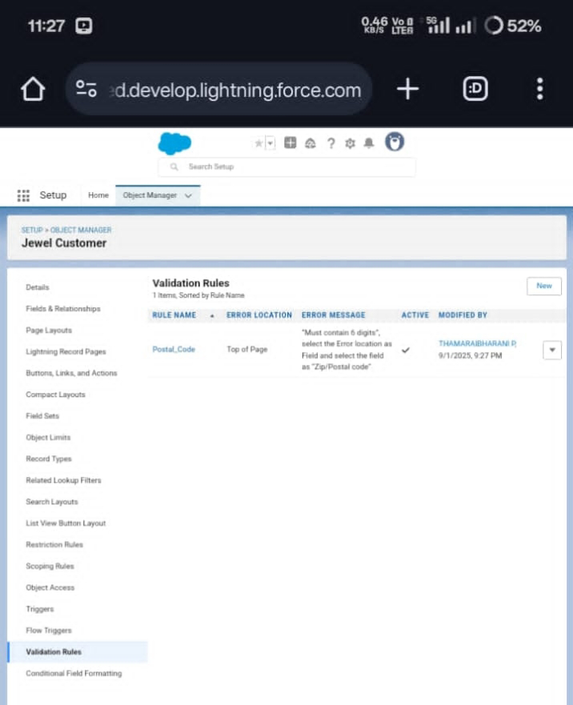


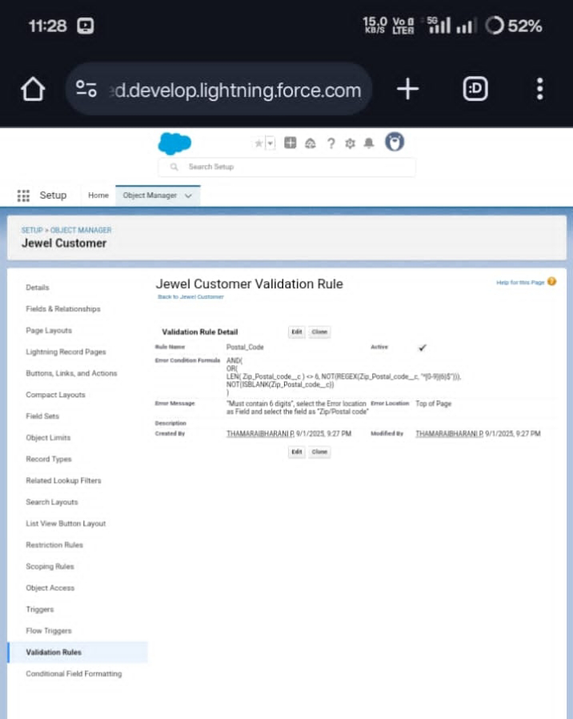


Field Label and Field Name as “Gold Price” and select formula return type as “Currency”

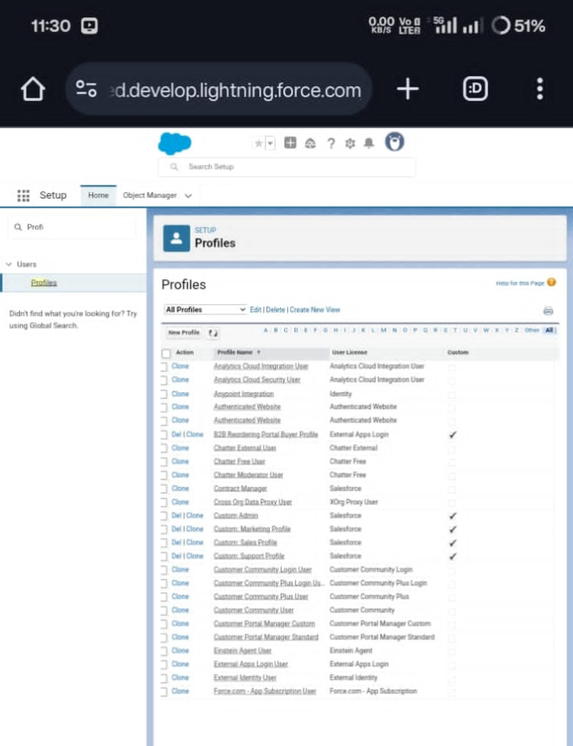


Creating the validation rule for Postal Code field in Jewel Customer object

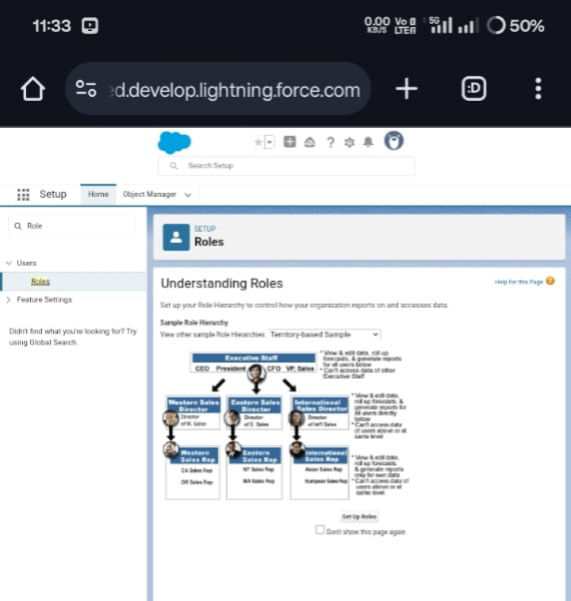


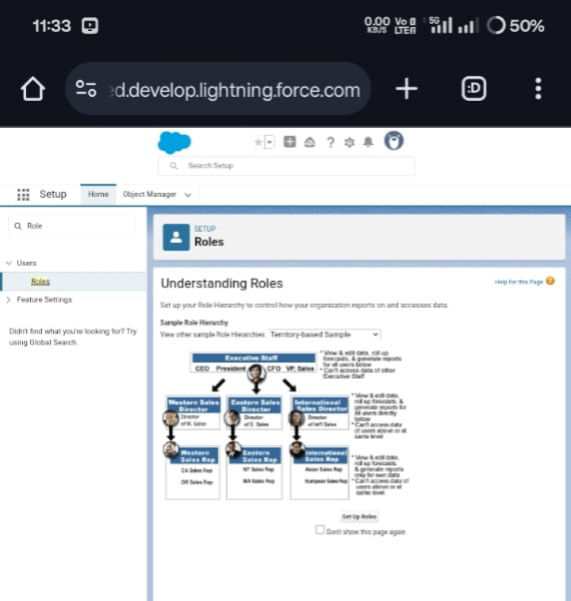


Gold Smith Profile and Worker Profile

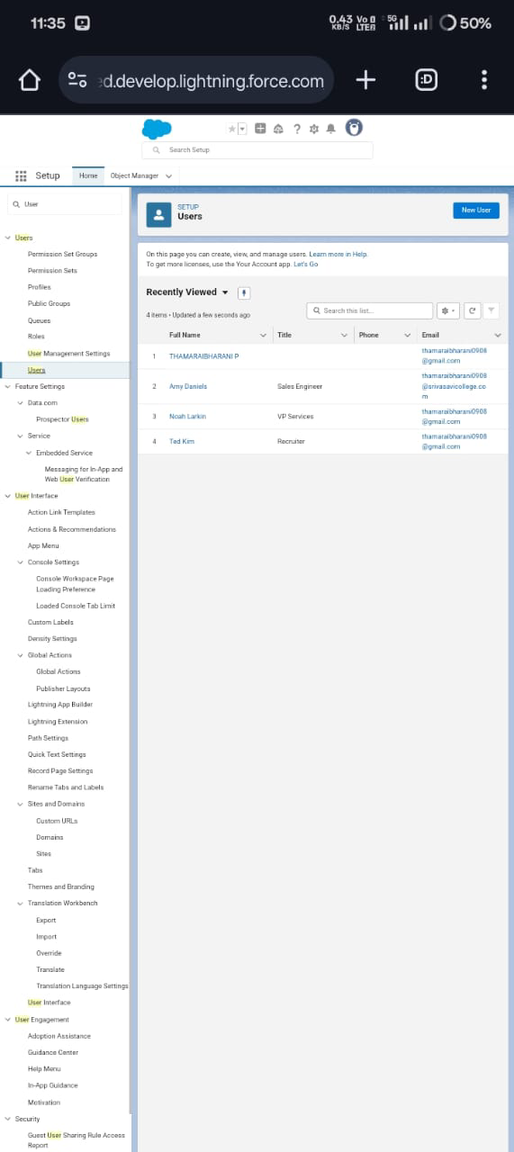


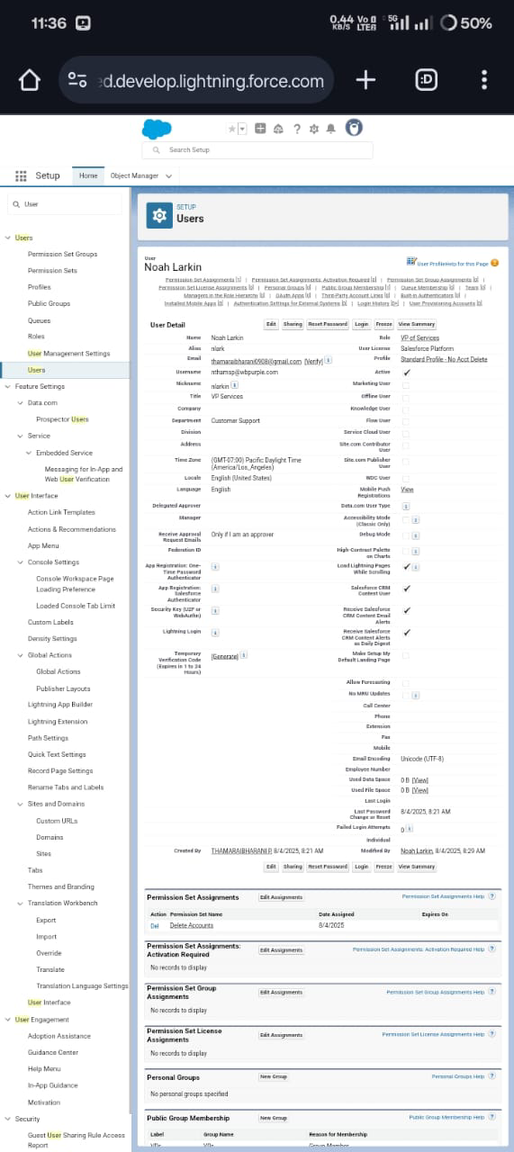
Creating Gold Smith Role and Create one more role as Worker which reports to Gold Smith.



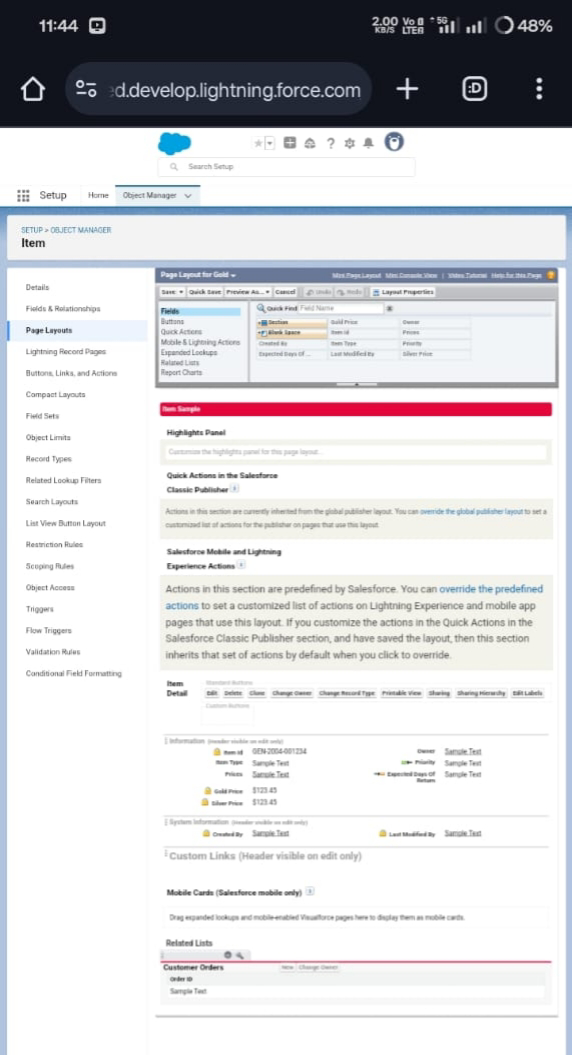


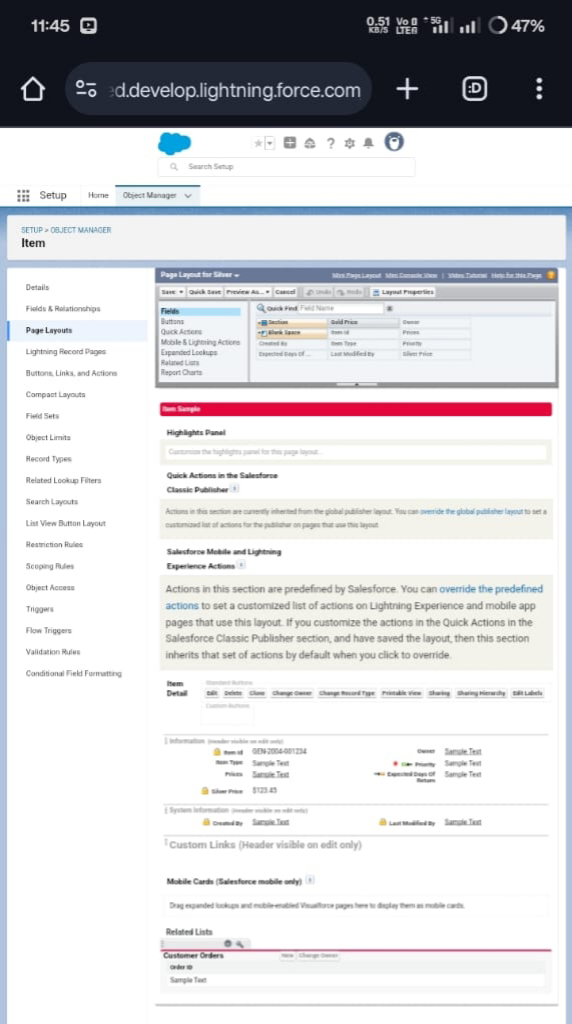
Create Users





To Create a Gold Page layout and To Create a Gold Page layout.

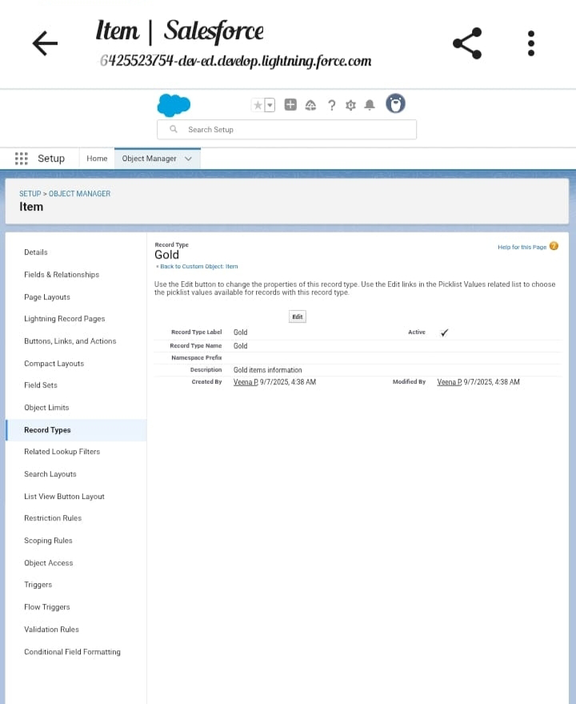




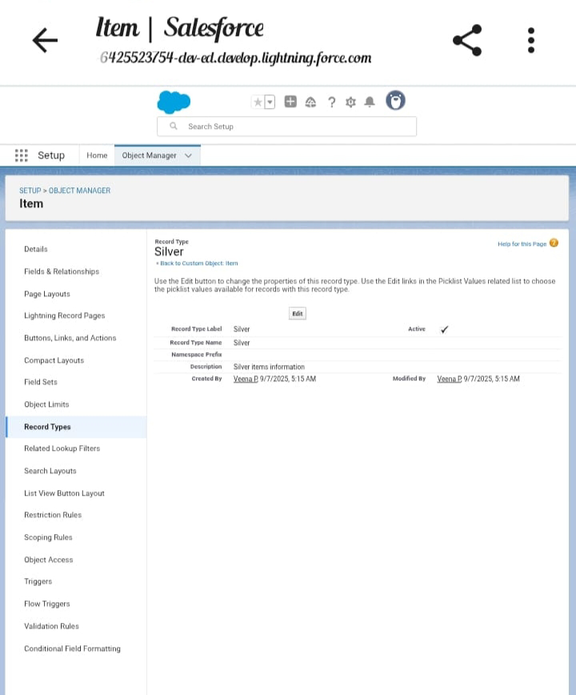
To create a Record Type



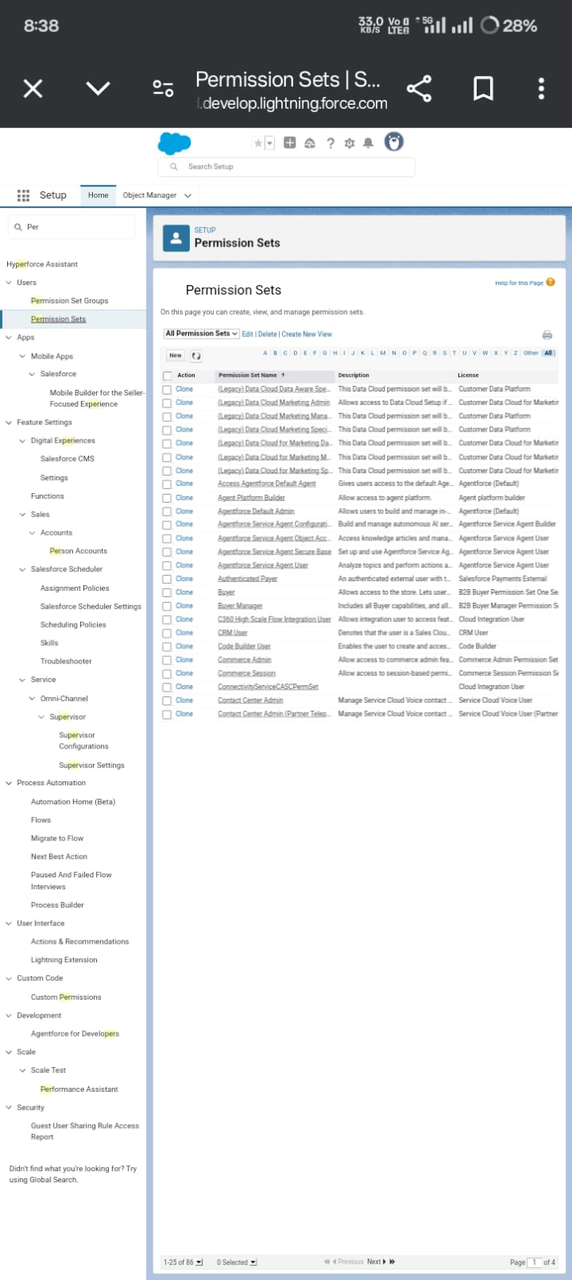
Gold

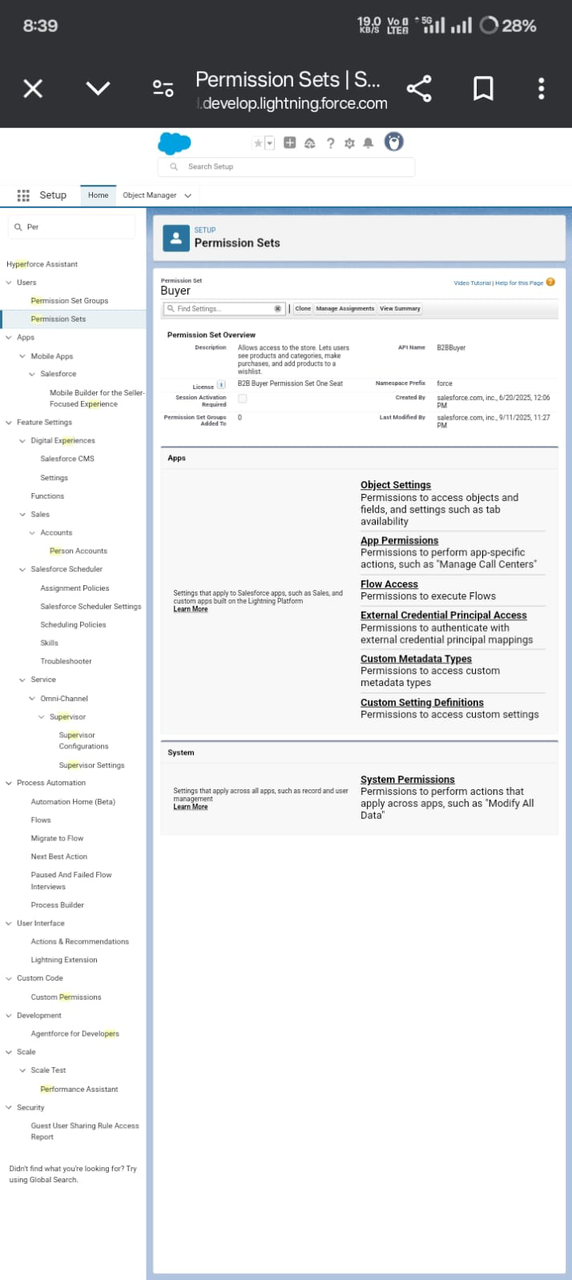


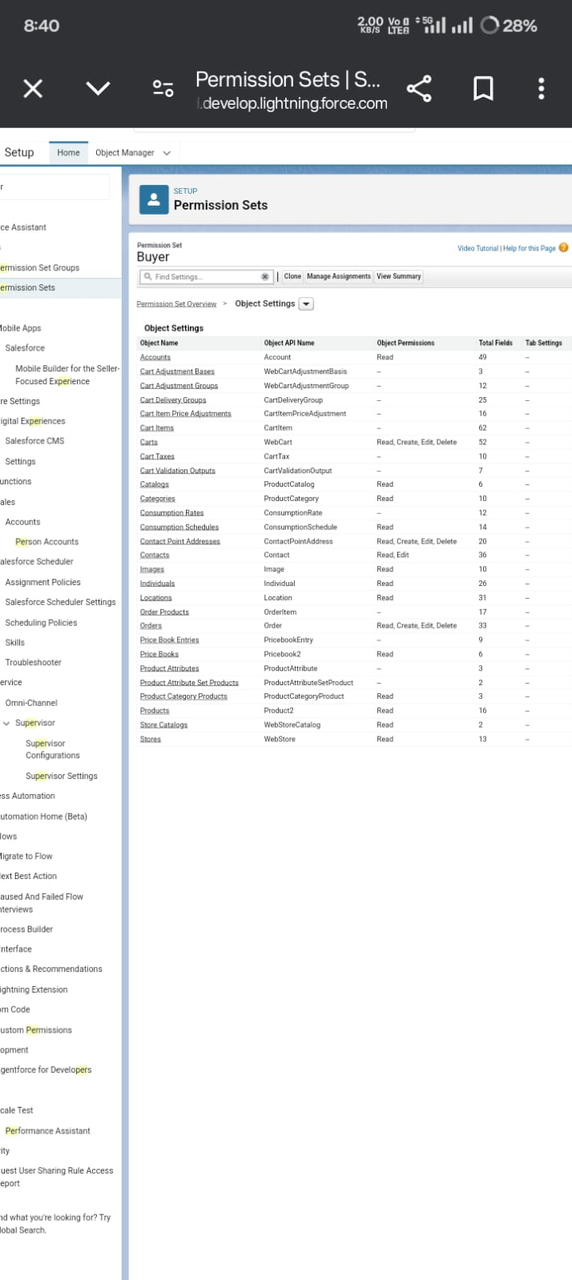
Silver



Creating permission set

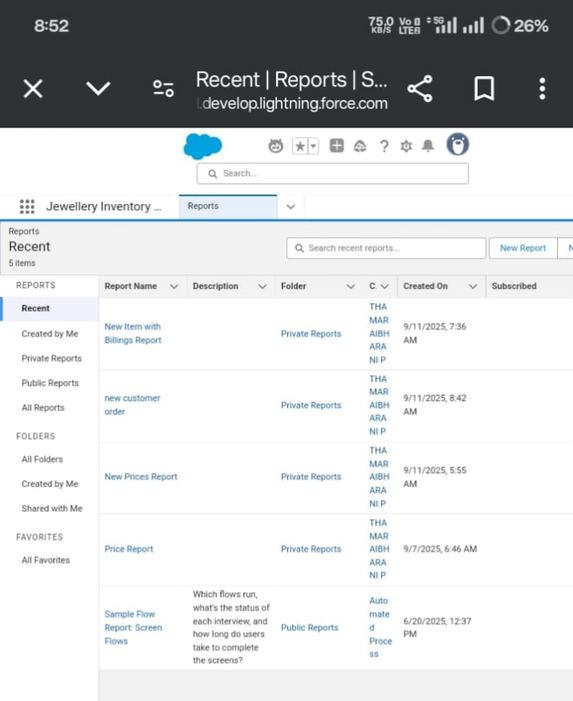


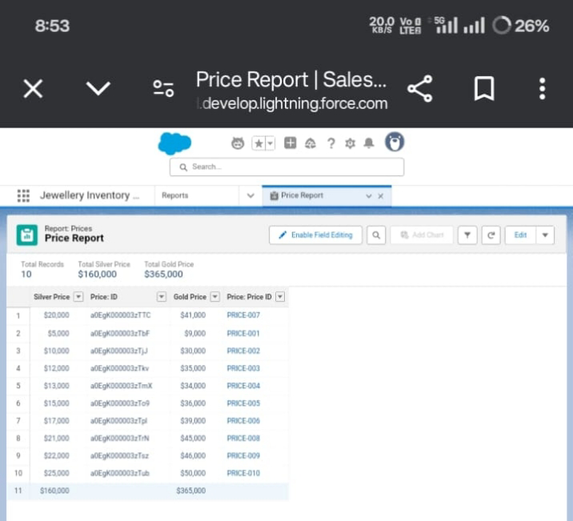


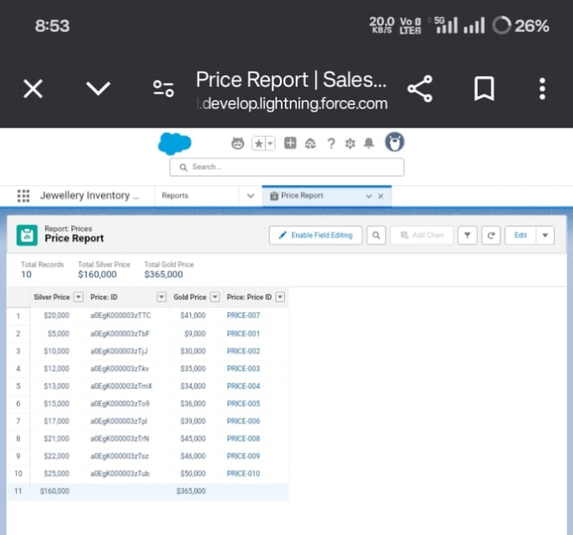


Create Reports : price report

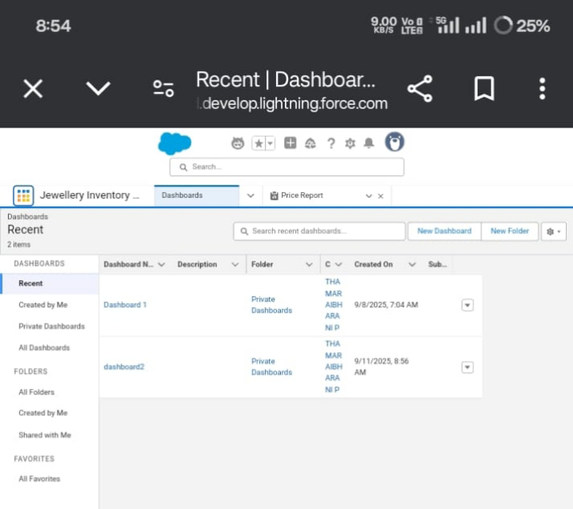
Create a report with report type: “Item with Billings”.Create a report with report type: “Billings with item and Customer order”.

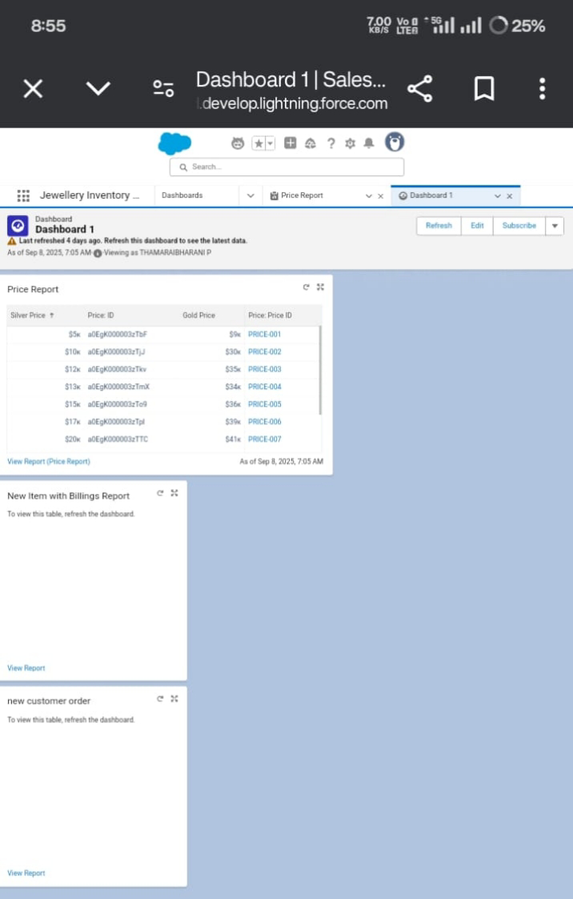


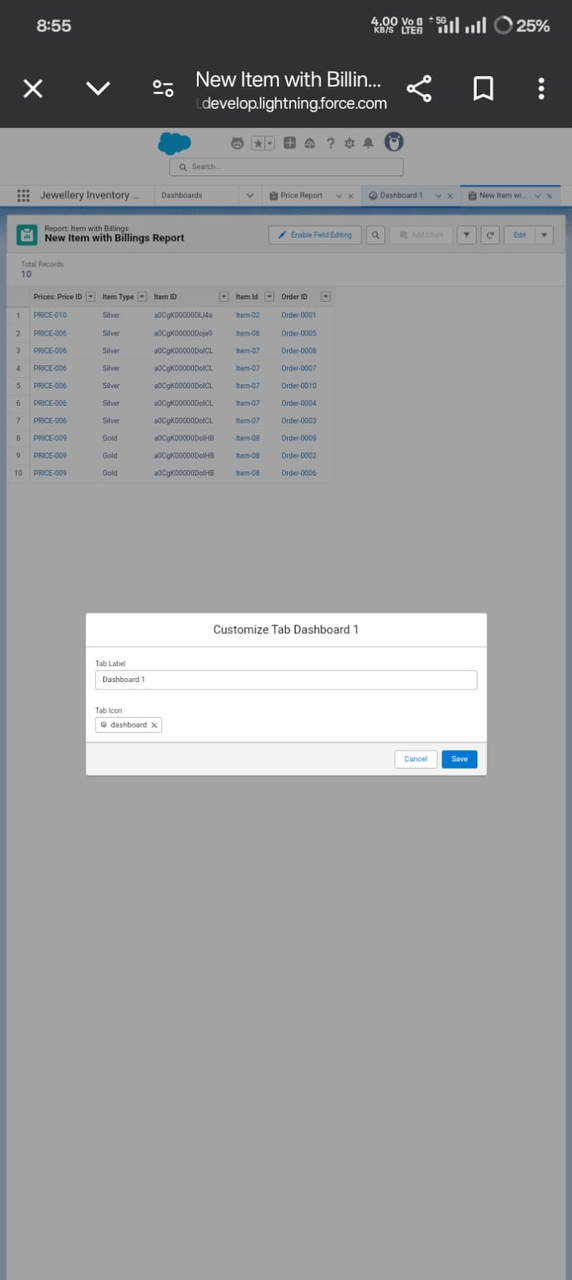




Create Dashboard

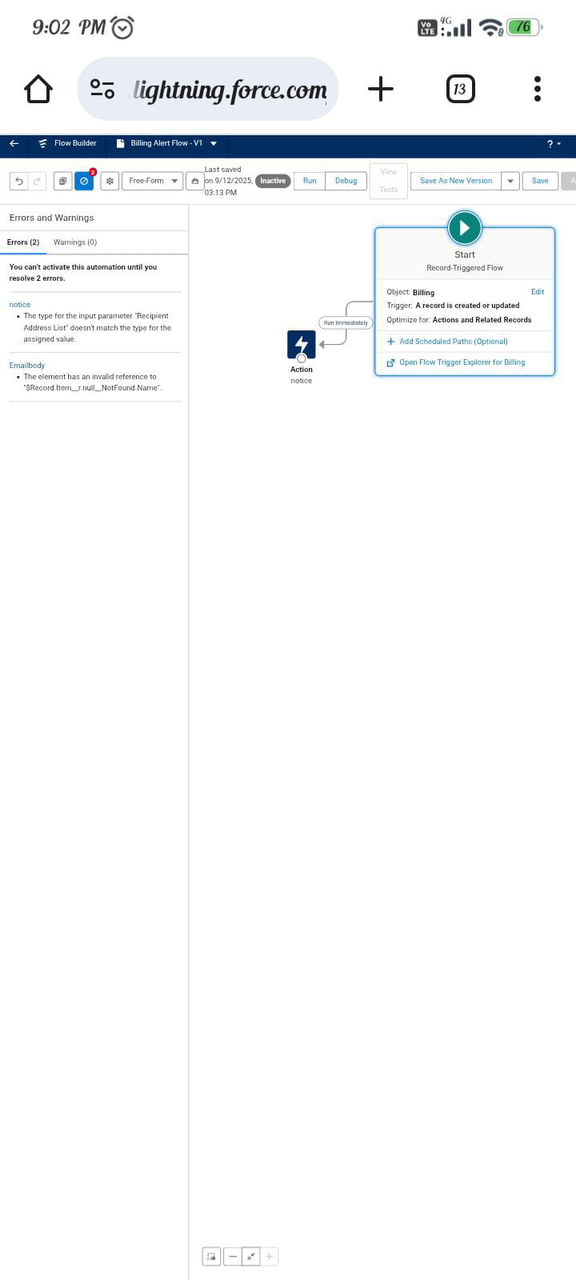






Create a Flow





ADVANTAGES & DISADVANTAGES

Advantages:

Organizes customer and sales data.Improves service and communication.Tracks inventory easily.Supports better decisions.

Disadvantages:

Setup and maintenance costs.Needs training.Risk of data breaches.Can reduce personal touch.

CONCLUSION:

The CRM application for jewel management helps businesses serve customers better, manage sales, and organize inventory efficiently. Though it has some costs and risks, it ultimately improves productivity and customer satisfaction, supporting long-term growth.

TRIGGER

Trigger handler:

CODE:

public class UpdatePaidAmountTriggerHandler {

public static void handleBeforeInsert(List<Billing\_\_c> newBillings) {

for (Billing\_\_c billing : newBillings) {

billing.Paid\_Amount\_\_c = billing.Paying\_Amount\_\_c;

}}

public static void handleBeforeUpdate(Map<Id, Billing\_\_c> oldBillingsMap, List<Billing\_\_c> updatedBillings) {

for (Billing\_\_c billing : updatedBillings) {

Billing\_\_c oldBilling = oldBillingsMap.get([billing.Id](http://billing.id));

Decimal oldPaidAmount = oldBilling.Paid\_Amount\_\_c;

billing.Paid\_Amount\_\_c = oldPaidAmount + billing.Paying\_Amount\_\_c;

}

}

}

Create the trigger:

trigger UpdatePaidAmountTrigger on Billing\_\_c (before insert, before update) {

if (Trigger.isInsert) {

UpdatePaidAmountTriggerHandler.handleBeforeInsert(Trigger.new);

} else if (Trigger.isUpdate) { UpdatePaidAmountTriggerHandler.handleBeforeUpdate(Trigger.oldMap, Trigger.new);

}

}