

PROJECT TITLE

CRM Application For Jewel Management

College Name: United College of Arts and Science

College Code: asbrubl

TEAM ID: A4EE666A074383EA2CF453EDDCA35E27

TEAM MEMBERS: 5

Team Leader Name: Rithesh.N

Email: rithesh8122@gmail.com

Team Member 1: Naveen Kumar.G

Email: naveenk53059@gmail.com

Team Member 2: Siva Kumar.R

Email: itzmerohit.devil@gmail.com

Team Member 3: Viswajith.S

Email: kumarsenthil47492@gmail.com

Team member 4: Deepak

Email:deepak220615@gmail.com

1.INTRODUCTION

1.1 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.

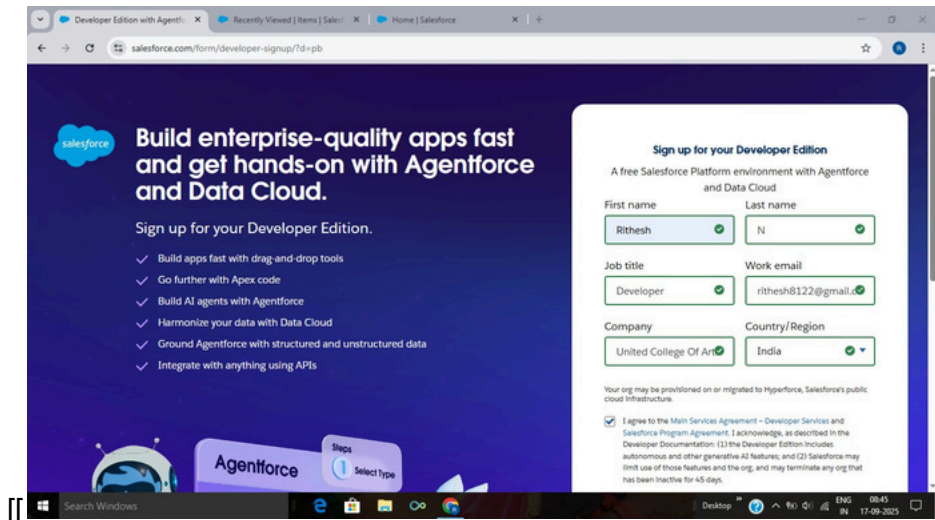
1.2 Purpose

The purpose of this project is to help jewelry businesses efficiently manage customers, sales, and inventory while building stronger customer relationships.

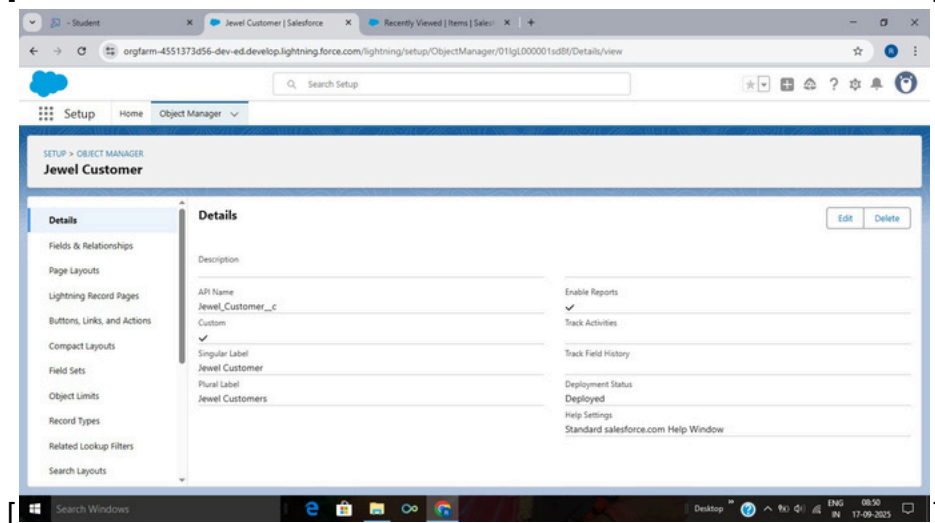
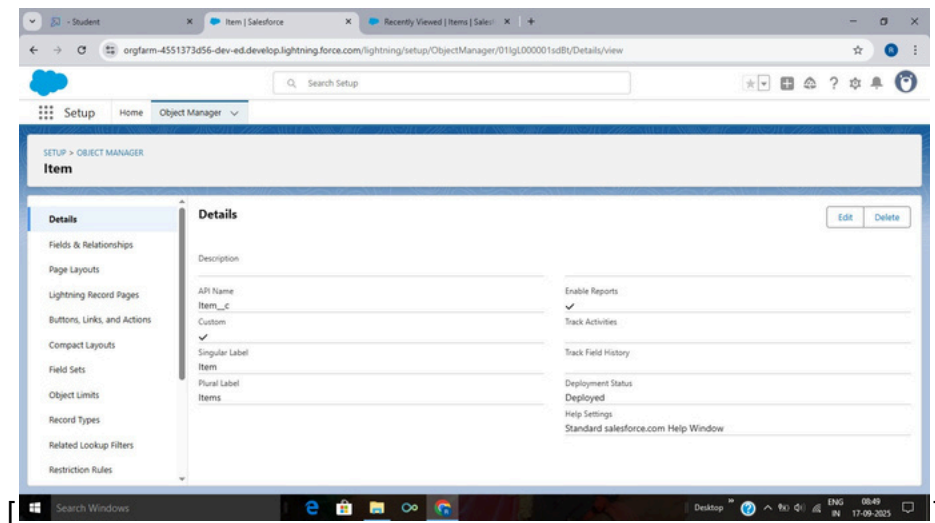
DEVELOPMENT PHASE

Creating Developer Account

1.By using this URL - <https://www.salesforce.com/form/developer-signup/?d=pb>

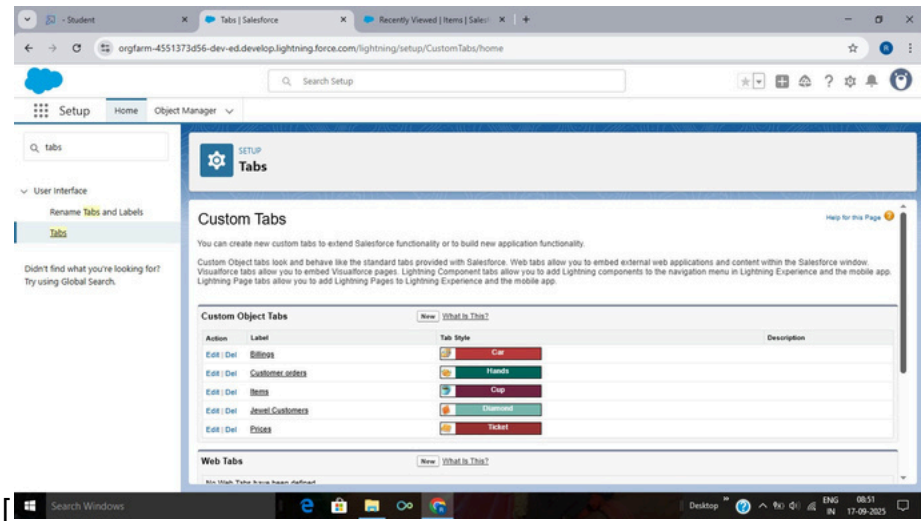


2.Created objects: Item, Jewel, Customer.

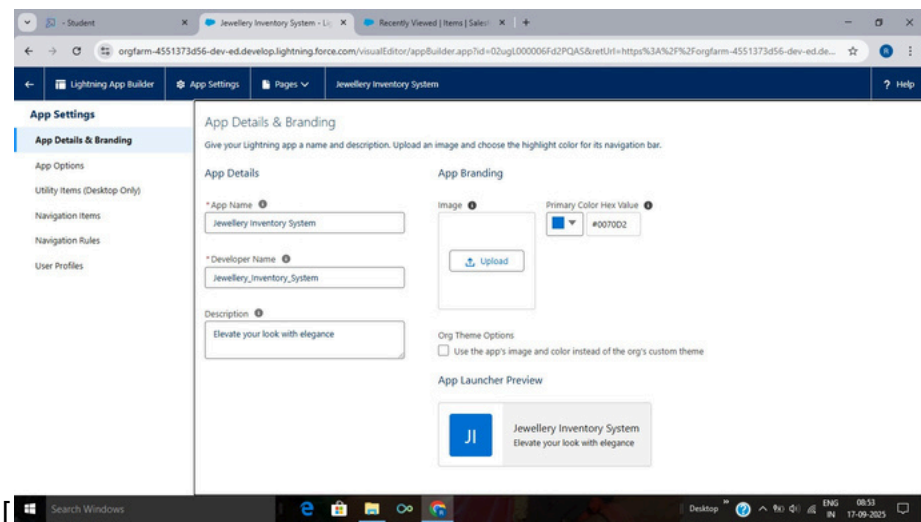


3. Create custom Tabs For

- Billing
- Customer Orders
- Items
- Jewel Customers
- Price



4. Developed Lightning App with relevant tabs App Name: Jewellery Inventory System



5. Create Fields And Relationship For

- Customer Orders

Customer order

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Customer	Customer__c	Lookup(Jewel Customer)		✓
Customer object id	Name	Auto Number		✓
Item	Item__c	Master-Detail(Item)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Order Status	Order_Status__c	Picklist		

• Jewel Customers

Jewel Customer

Fields & Relationships

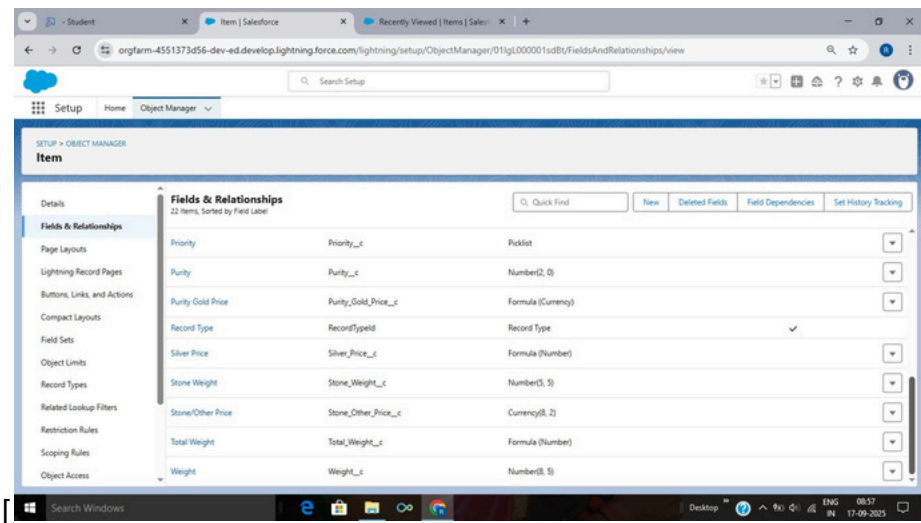
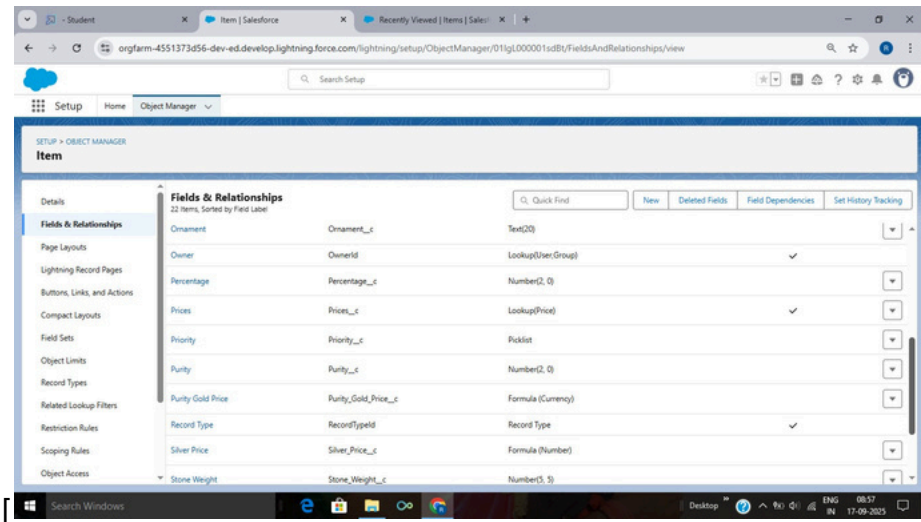
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
City	City__c	Text(20)		
Country	Country__c	Text(16)		
Created By	CreatedBy	Lookup(User)		
Email	Email__c	Email		
Jewel Customer Name	Name	Text(80)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User Group)		✓
Phone	Phone__c	Phone		
State	State__c	Text(20)		
Street	Street__c	Text(20)		
Zip/Postal code	Zip_Postal_code__c	Text(16)		

• Items

Item

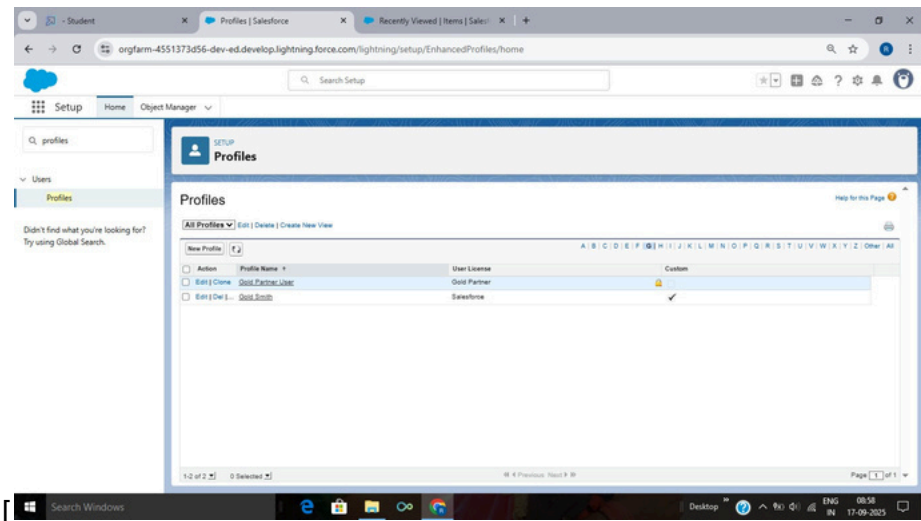
Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Formula (Currency)		
Created By	CreatedBy	Lookup(User)		
Expected Days Of Return	Expected_Days_Of_Return__c	Picklist	Priority	
Gold Price	Gold_Price__c	Formula (Currency)		
Item Id	Name	Auto Number		✓
Item Type	Item_Type__c	Picklist		
KDM	KDM__c	Formula (Currency)		
Last Modified By	LastModifiedBy	Lookup(User)		
Making Charges	Making_Charges__c	Formula (Currency)		

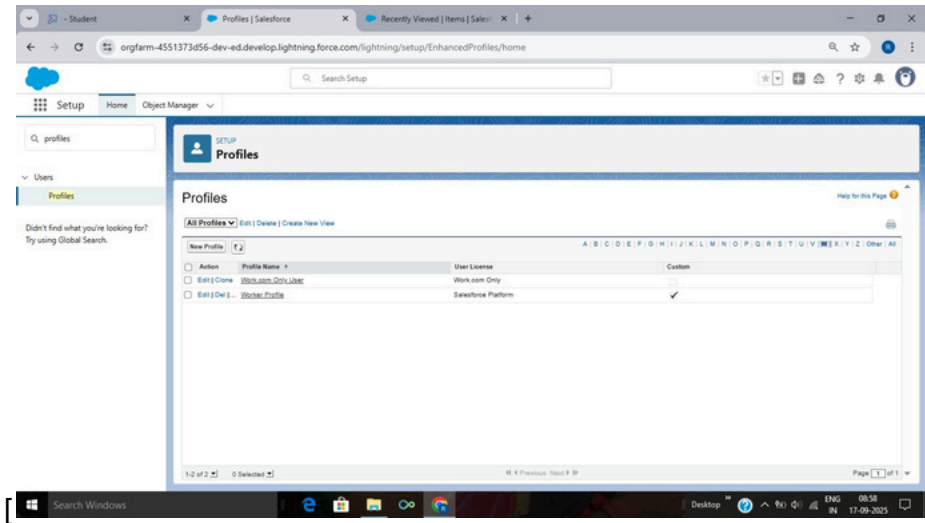


6.Create Profiles For Goldsmith and Worker

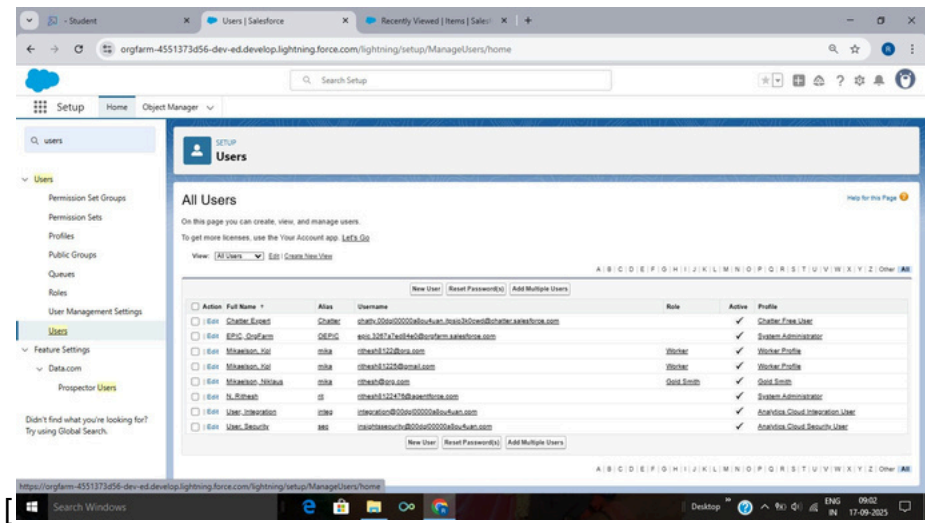
● Gold Smith Profile



- Worker Profile

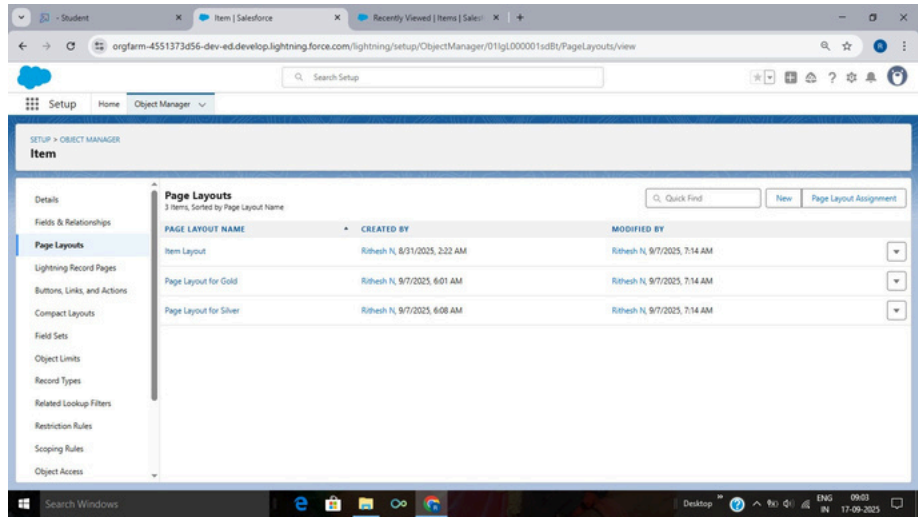


7.Create Users



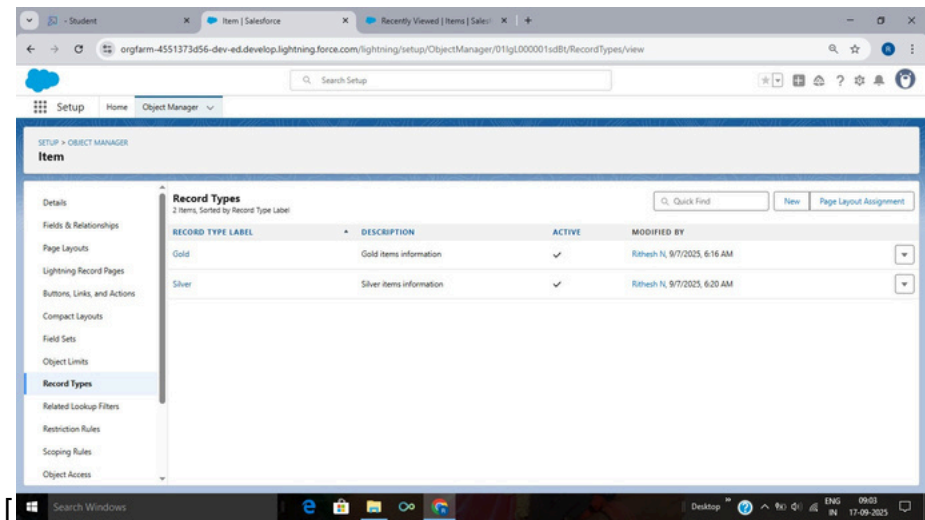
8. Create Page Layouts for

- Page Layouts for gold
- Page Layouts for silver
- Item Layouts



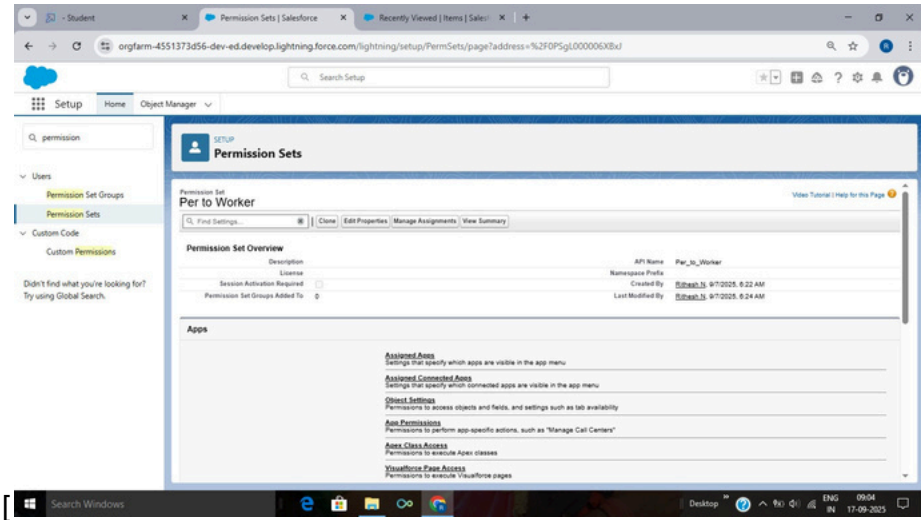
9. Create Record Types for

- Gold
- Silver



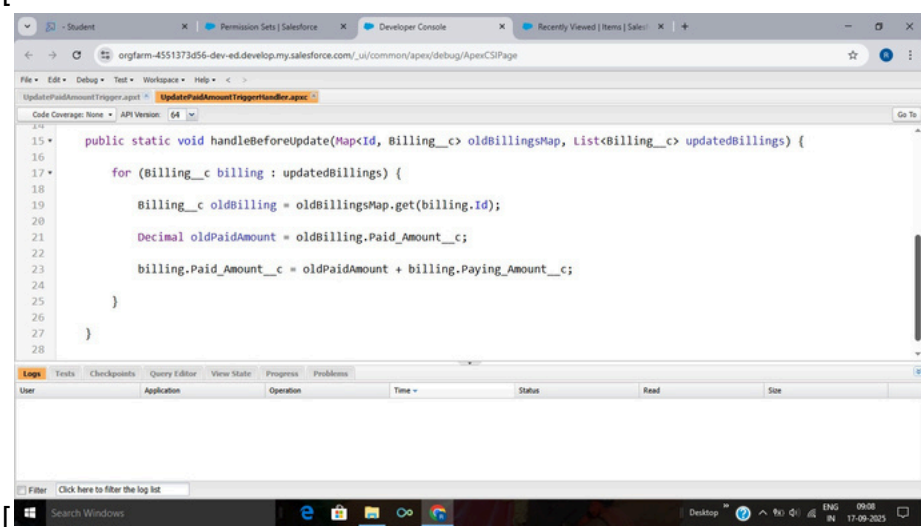
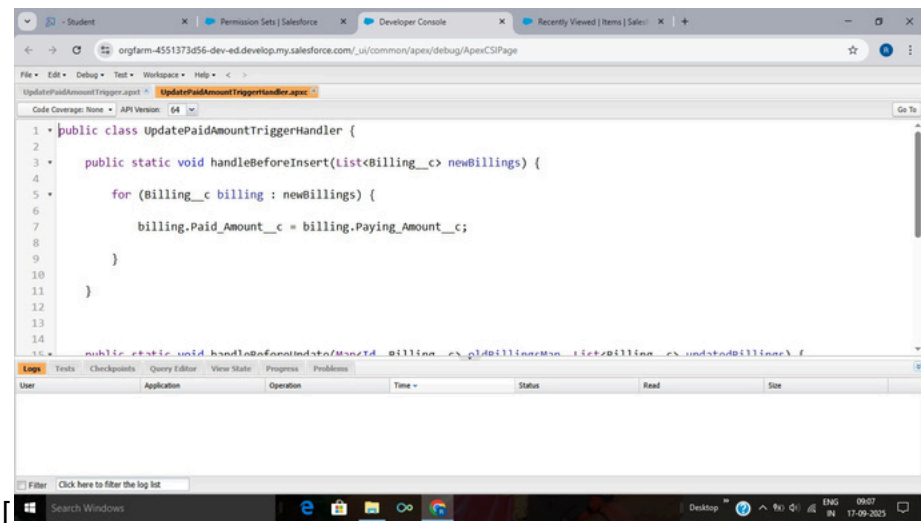
10. Create Permission Set for

- Per to Worker



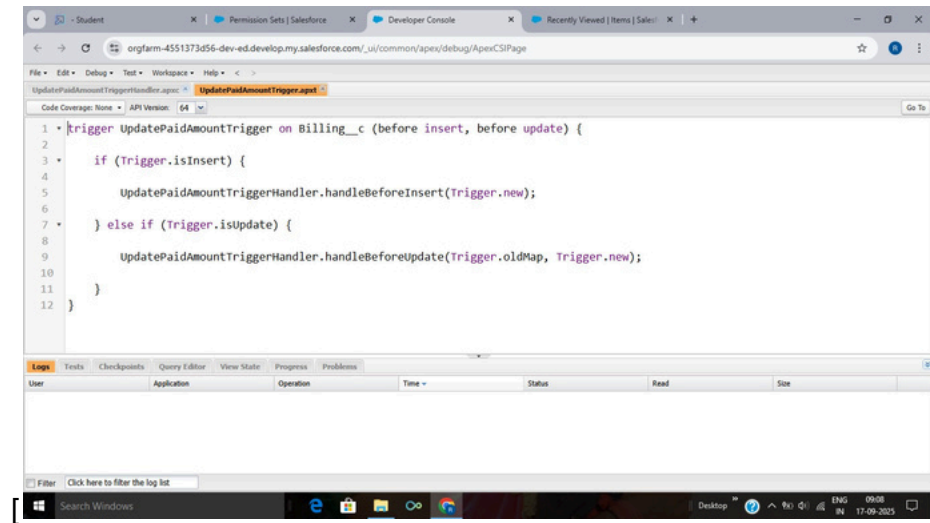
11. Create a Trigger Handler Class

- UpdatePaidAmountTriggerHandler

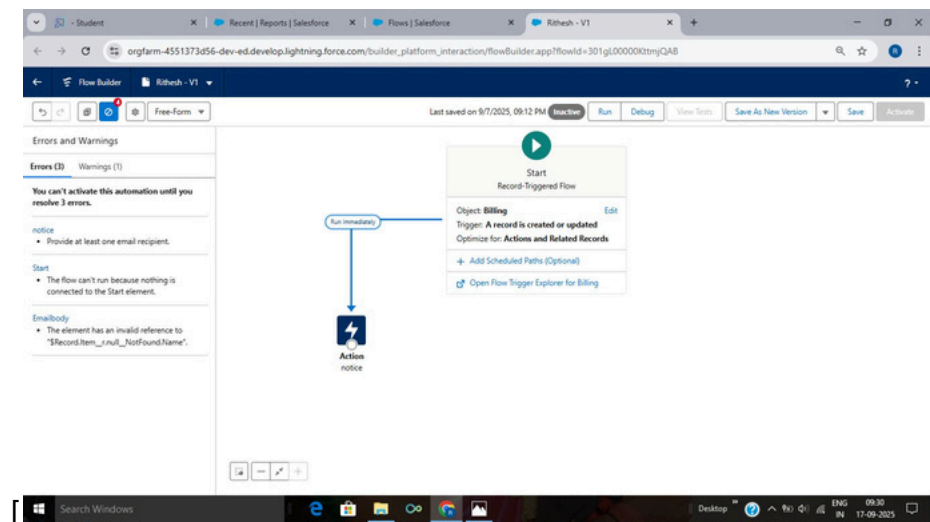


12.Create a Apex Trigger

● UpdatePaidAmountTrigger



13.Create a Flow



CONCLUSION.

The CRM Application for Jewel Management demonstrates how Salesforce can be tailored for jewelry retail. It integrates sales, inventory, and CRM into a single platform with automation features that improve efficiency, reduce errors, and enhance customer satisfaction.

APPENDIX.

Code for Trigger Handler Class(UpdatePaidAmountTriggerHandler)

● 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;

}

}

```

```

publicstatic void handleBeforeUpdate(Map<Id, Billing__c> oldBillingsMap, List<Billing__c>
updatedBillings) {

```

```

    for(Billing__c billing : updatedBillings) {

        Billing__c oldBilling = oldBillingsMap.get(billing.Id);

        Decimal oldPaidAmount = oldBilling.Paid_Amount__c;

        billing.Paid_Amount__c = oldPaidAmount + billing.Paying_Amount__c;

    }

}

}

```

CodeForApex Trigger(UpdatePaidAmountTrigger)
CODE:

```

triggerUpdatePaidAmountTrigger on Billing__c (before insert, before update) {

    if(Trigger.isInsert) {

        UpdatePaidAmountTriggerHandler.handleBeforeInsert(Trigger.new);

    }elseif(Trigger.isUpdate) {

        UpdatePaidAmountTriggerHandler.handleBeforeUpdate(Trigger.oldMap, Trigger.new);

    }

}

```

