

## 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:

## 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.

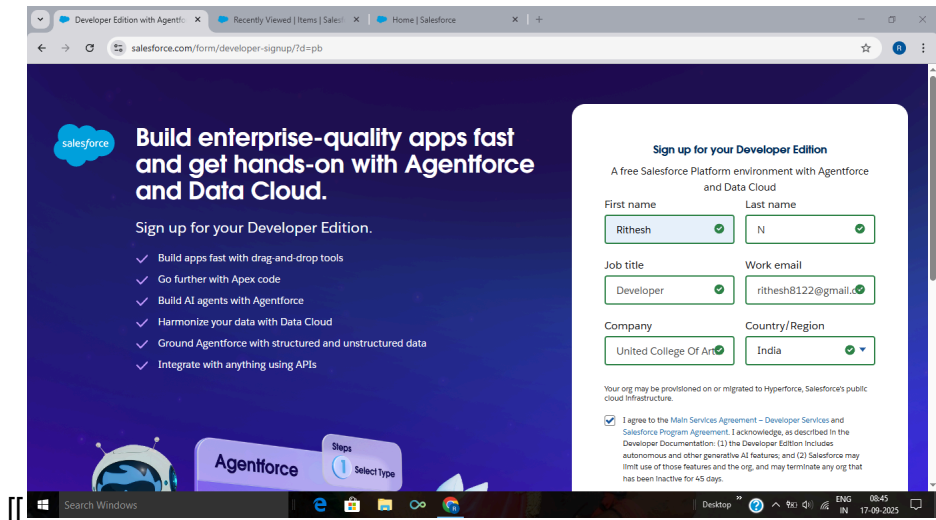
### 1.2 Purpose

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

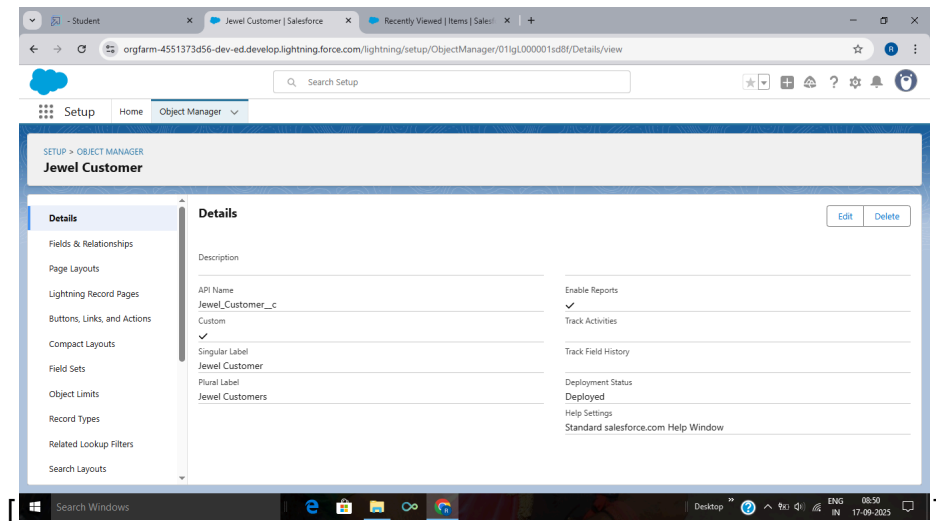
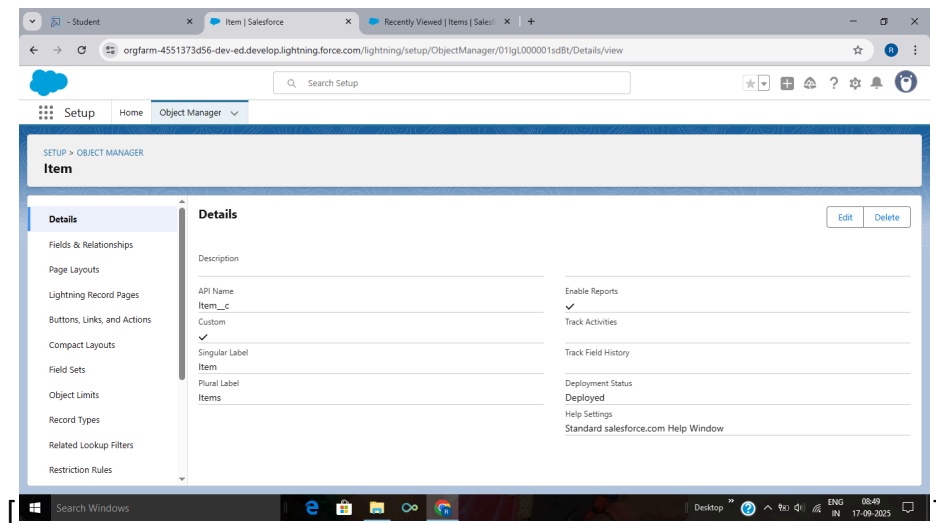
## DEVELOPMENT PHASE

Creating Developer Account

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

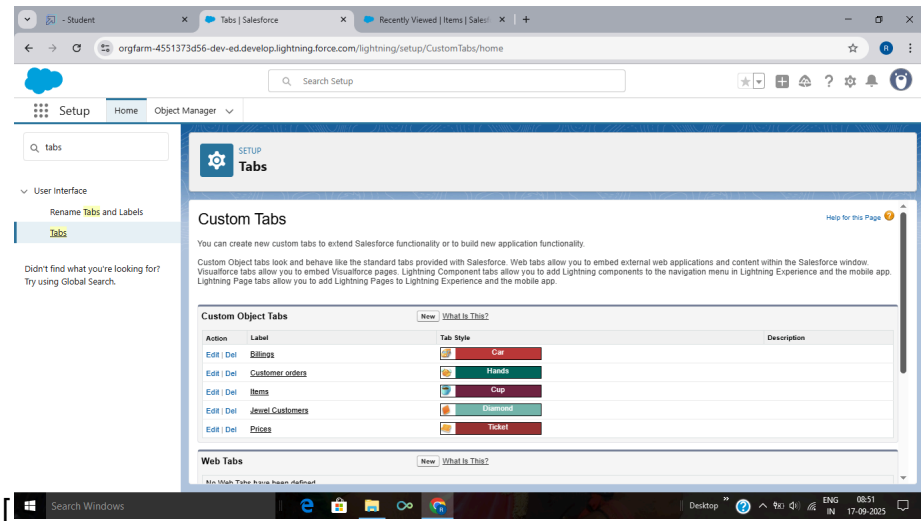


## 2.Created objects: Item, Jewel, Customer.

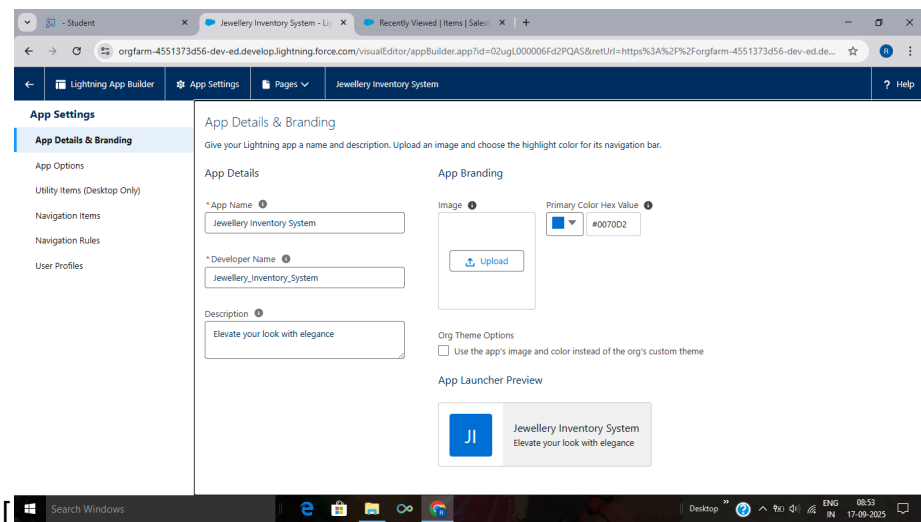


### 3. Create custom Tabs For

- 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

Student Customer order | Salesforce Recently Viewed | Items | Sales | +

orgfarm-4551373d56-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gl000001zFc/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER

### Customer order

Details

**Fields & Relationships** 6 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

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

Search Windows Desktop 08:55 17-09-2025

## • Jewel Customers

Student Jewel Customer | Salesforce Recently Viewed | Items | Sales | +

orgfarm-4551373d56-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gl000001sd8/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER

### Jewel Customer

Details

**Fields & Relationships** 11 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
City	City__c	Text(20)		
Country	Country__c	Text(18)		
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
Jewel Customer Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	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(18)		

Search Windows Desktop 08:55 17-09-2025

## • Items

Student Item | Salesforce Recently Viewed | Items | Sales | +

orgfarm-4551373d56-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gl000001sd8/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER

### Item

Details

**Fields & Relationships** 22 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Formula (Currency)		
Created By	CreatedById	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	LastModifiedById	Lookup(User)		
Making Charges	Making_Charges__c	Formula (Currency)		

Search Windows Desktop 08:56 17-09-2025

The screenshot shows the Salesforce Object Manager interface for an 'Item' object. The 'Fields & Relationships' tab is active, displaying a list of 22 fields. The fields are organized into three columns: Name, API Name, and Data Type. The fields include Ornament, Owner, Percentage, Prices, Priority, Purity, Purity Gold Price, Record Type, Silver Price, and Stone Weight. Each field has a corresponding API name and data type, and a dropdown menu to select the field's behavior (e.g., Text, Number, Formula, Picklist).

Field Name	API Name	Data Type	Behavior
Ornament	Ornament__c	Text(20)	
Owner	OwnerId	Lookup(User Group)	✓
Percentage	Percentage__c	Number(2, 0)	
Prices	Prices__c	Lookup(Price)	✓
Priority	Priority__c	Picklist	
Purity	Purity__c	Number(2, 0)	
Purity Gold Price	Purity_Gold_Price__c	Formula (Currency)	
Record Type	RecordTypeId	Record Type	✓
Silver Price	Silver_Price__c	Formula (Number)	
Stone Weight	Stone_Weight__c	Number(5, 5)	

The screenshot shows the Salesforce Object Manager interface for an 'Item' object. The 'Fields & Relationships' tab is active, displaying a list of 22 fields. The fields are organized into three columns: Name, API Name, and Data Type. The fields include Priority, Purity, Purity Gold Price, Record Type, Silver Price, Stone Weight, Stone/Other Price, Total Weight, and Weight. Each field has a corresponding API name and data type, and a dropdown menu to select the field's behavior (e.g., Text, Number, Formula, Picklist).

Field Name	API Name	Data Type	Behavior
Priority	Priority__c	Picklist	
Purity	Purity__c	Number(2, 0)	
Purity Gold Price	Purity_Gold_Price__c	Formula (Currency)	
Record Type	RecordTypeId	Record Type	✓
Silver Price	Silver_Price__c	Formula (Number)	
Stone Weight	Stone_Weight__c	Number(5, 5)	
Stone/Other Price	Stone_Other_Price__c	Currency(8, 2)	
Total Weight	Total_Weight__c	Formula (Number)	
Weight	Weight__c	Number(8, 5)	

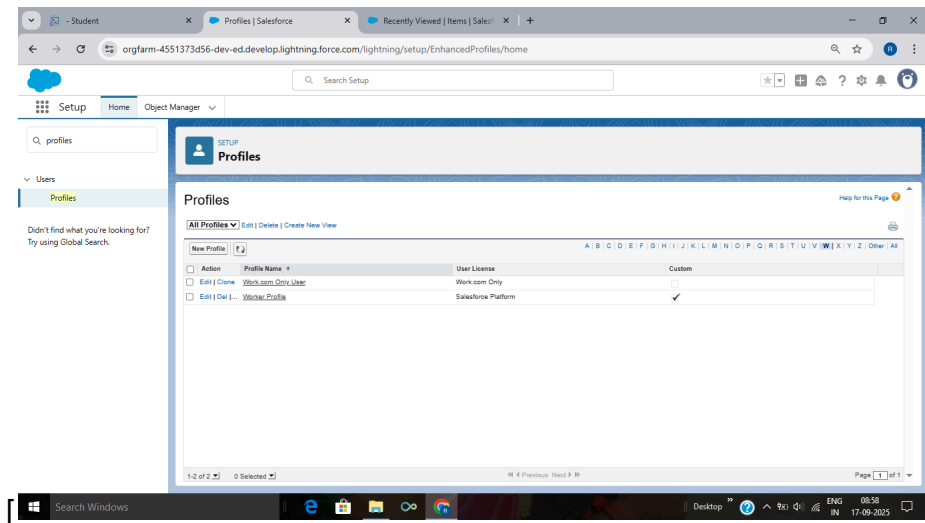
## 6. Create Profiles For Goldsmith and Worker

- Gold Smith Profile

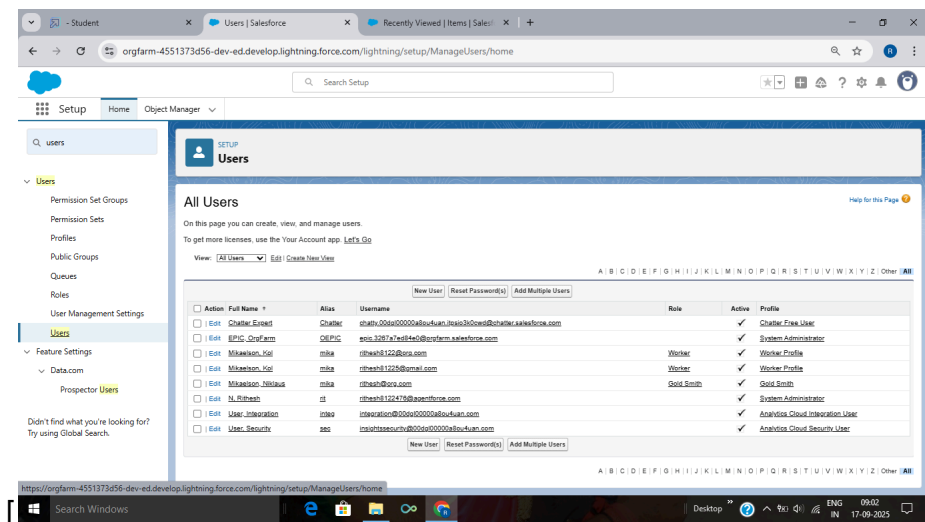
The screenshot shows the Salesforce Profiles page. The 'Profiles' tab is active, displaying a list of profiles. The 'Gold Smith' profile is selected, showing its details. The profile is named 'Gold Smith' and is associated with the 'Salesforce' user license. The profile is currently in the 'Draft' state.

Profile Name	User License	State
Gold Smith	Salesforce	Draft

- Worker Profile

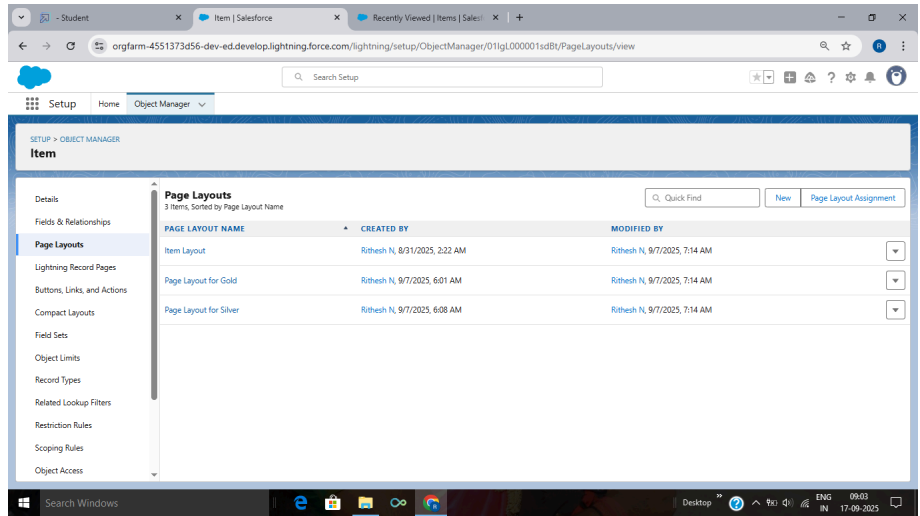


## 7.Create Users



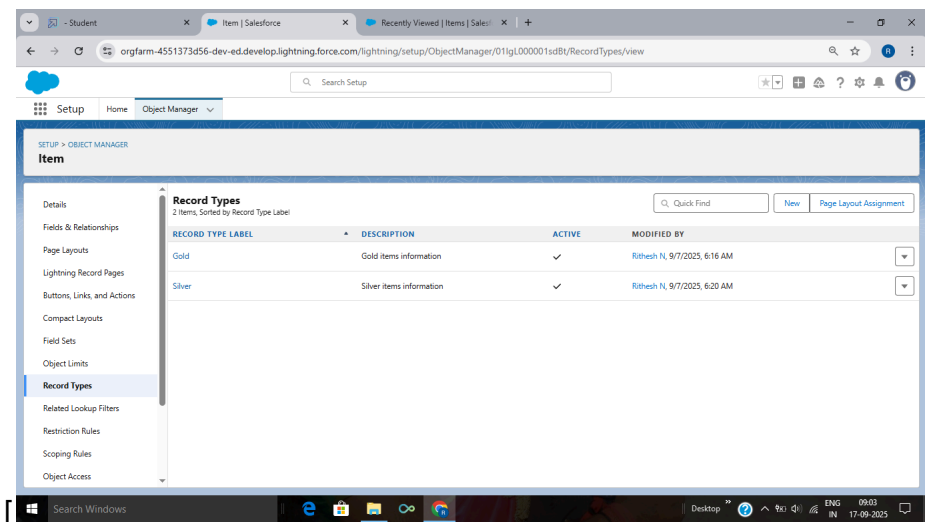
## 8.Create Page Layouts for

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



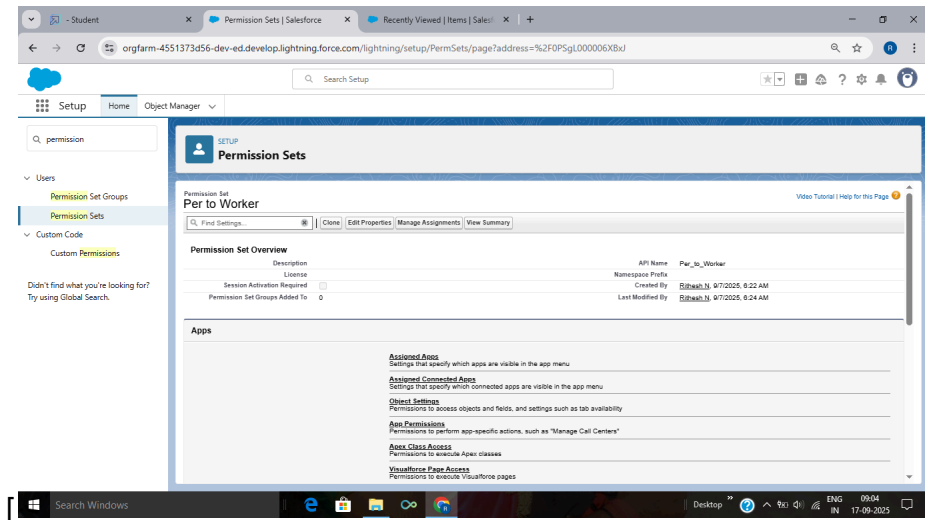
## 9. Create Record Types for

- Gold
- Silver



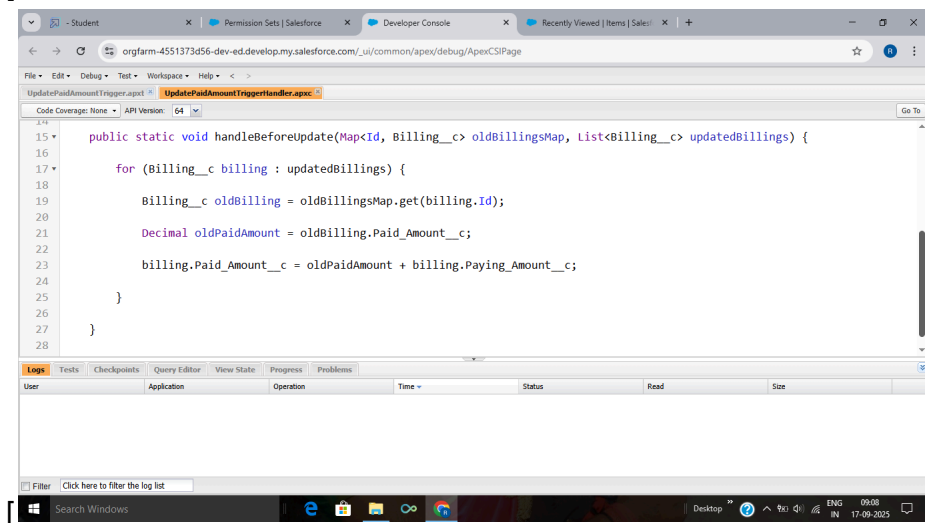
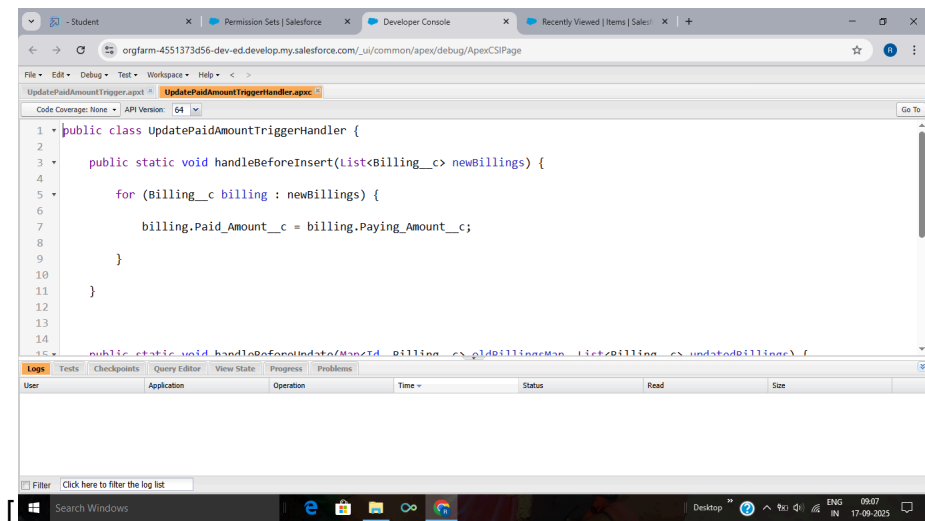
## 10. Create Permission Set for

- Per to Worker



## 11. Create a Trigger Handler Class

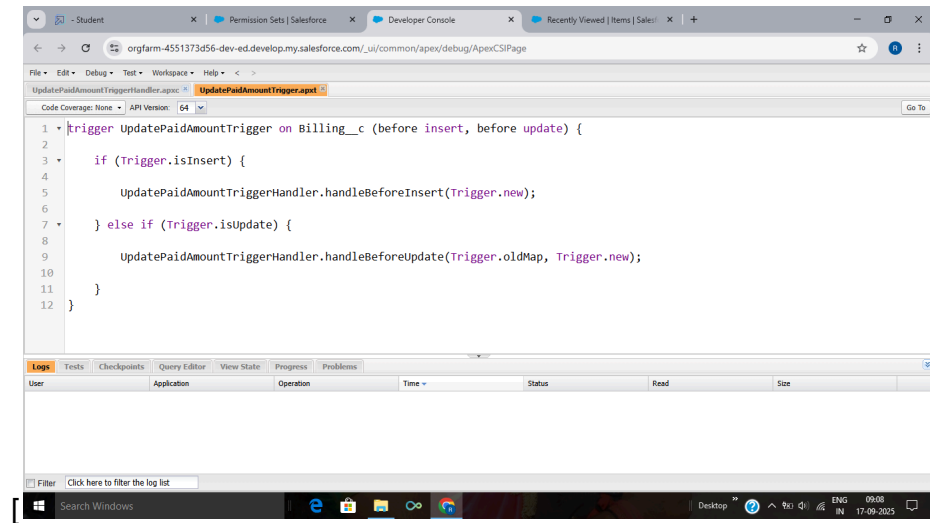
- UpdatePaidAmountTriggerHandler



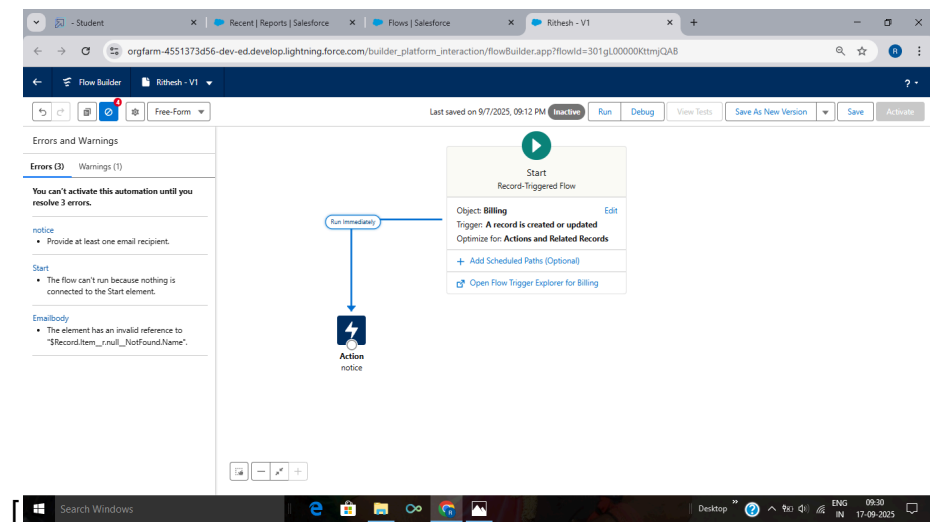


## 12.Create a Apex Trigger

- 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;

}

}

```

```

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);

        Decimal oldPaidAmount = oldBilling.Paid_Amount__c;

        billing.Paid_Amount__c = oldPaidAmount + billing.Paying_Amount__c;

    }

}

}

```

Code For Apex Trigger(UpdatePaidAmountTrigger)  
CODE:

```

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);

    }

}

```

