**CRM APPLICATION FOR JEWEL MANAGEMENT - DEVELOPER**

**College Name:** **Karuppannan Mariappan College**

**College Code: bru3s**

**TEAM ID**: NM2025TMID28556

**TEAM MEMBERS**:

Team Leader Name: MAYILMURUGAN S

Email: mayilm968@gmail.com

Team Member1: SATHISH M

Email: sathishh1117@gmail.com

Team Member2: SIVAKUMAR R

Email: sivakumar061205@gmail.com

Team Member3: SUKUMAR S

Email: sukumar20052005@gmail.com

**1.INTRODUCTION:**

**1.1 Project Overview:**

The **CRM Application for Jewel Management** is a Salesforce based solution designed to simplify and optimize jewel business operations. It provides an integrated platform to manage customer relationships, sales, inventory, billing after-sales services. The system ensures seamless interaction with customers by storing purchase history. It allowing jewellers to deliver personalized services and build long-term customer loyalty.

**1.2 Purpose:**

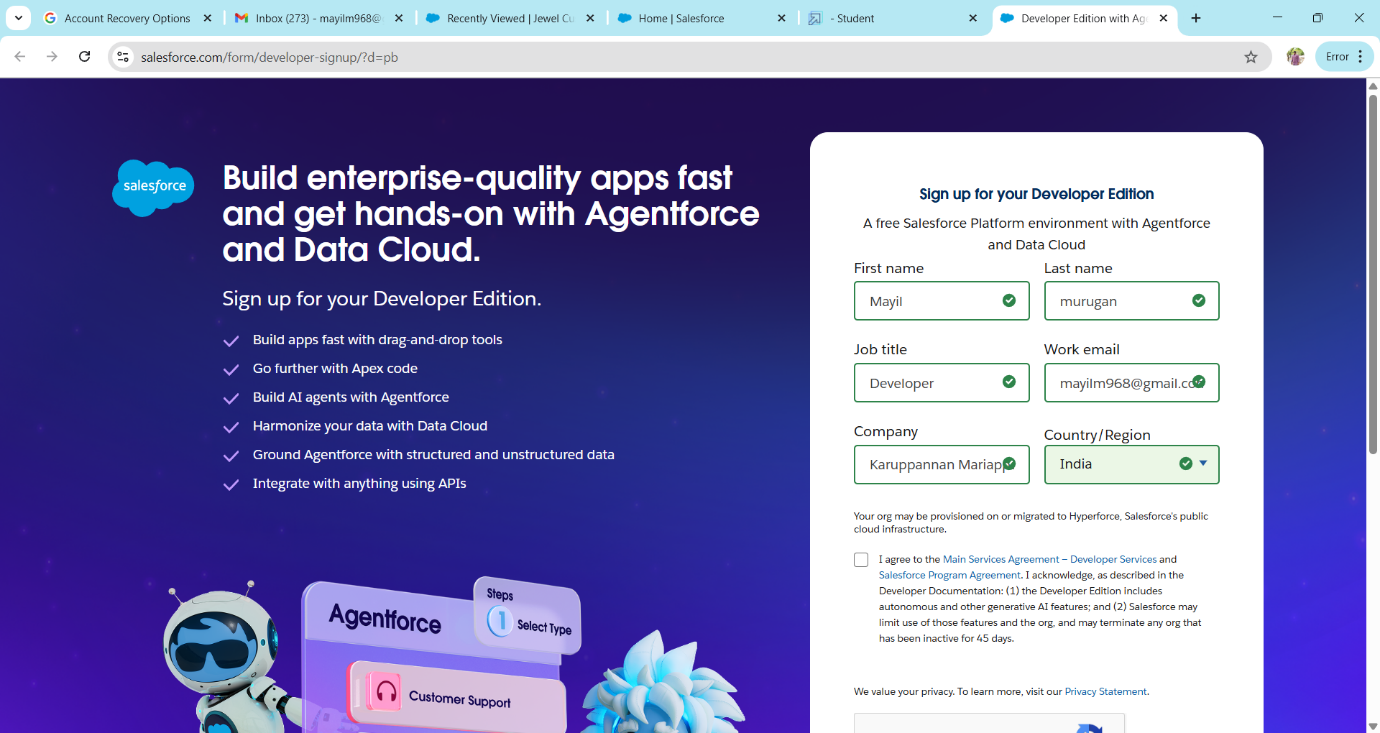
The primary purpose of the **Jewel Management CRM Application** is to enable jewel businesses to manage their operation and customer interactions more effectively.

The system enhance customer satisfaction, strengthens business efficiency, and ensures better communication across the organization.

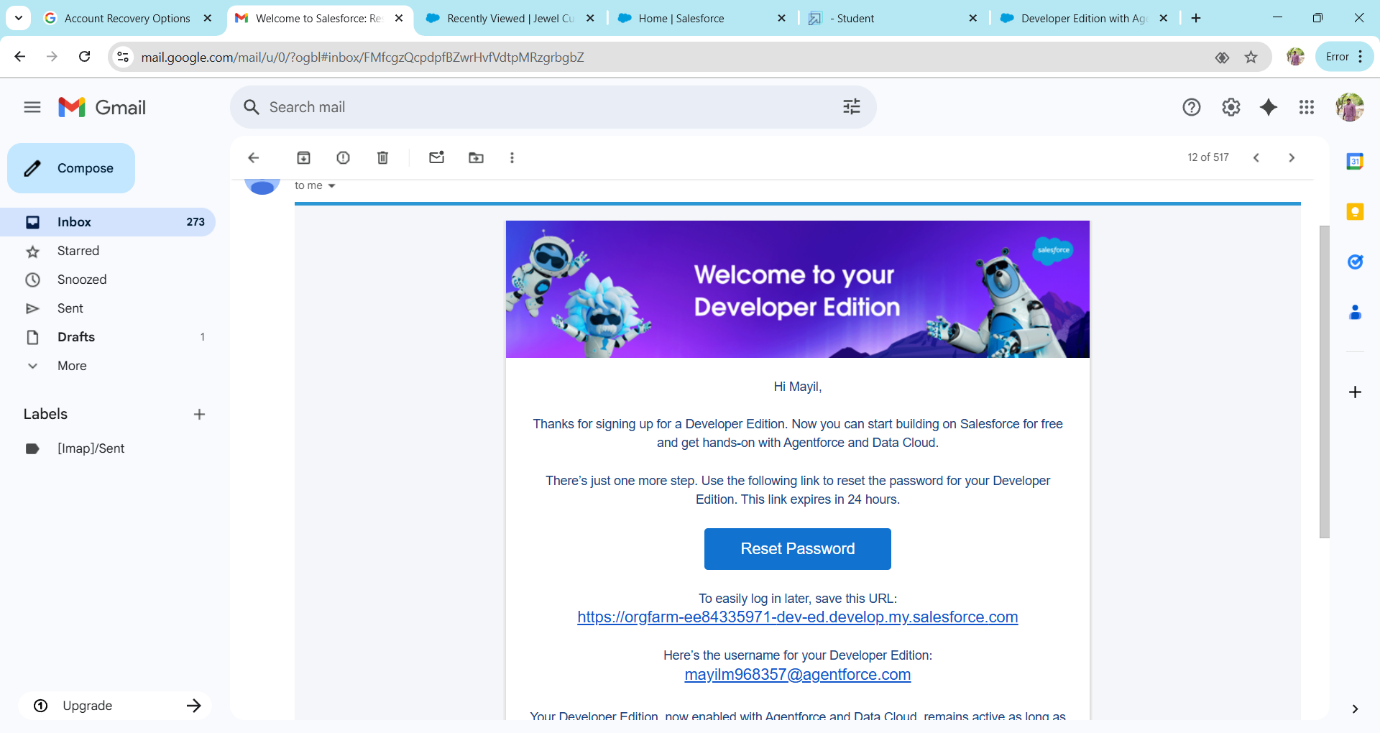
**DEVELOPMENT PHASE**

**Creating Developer Account:**

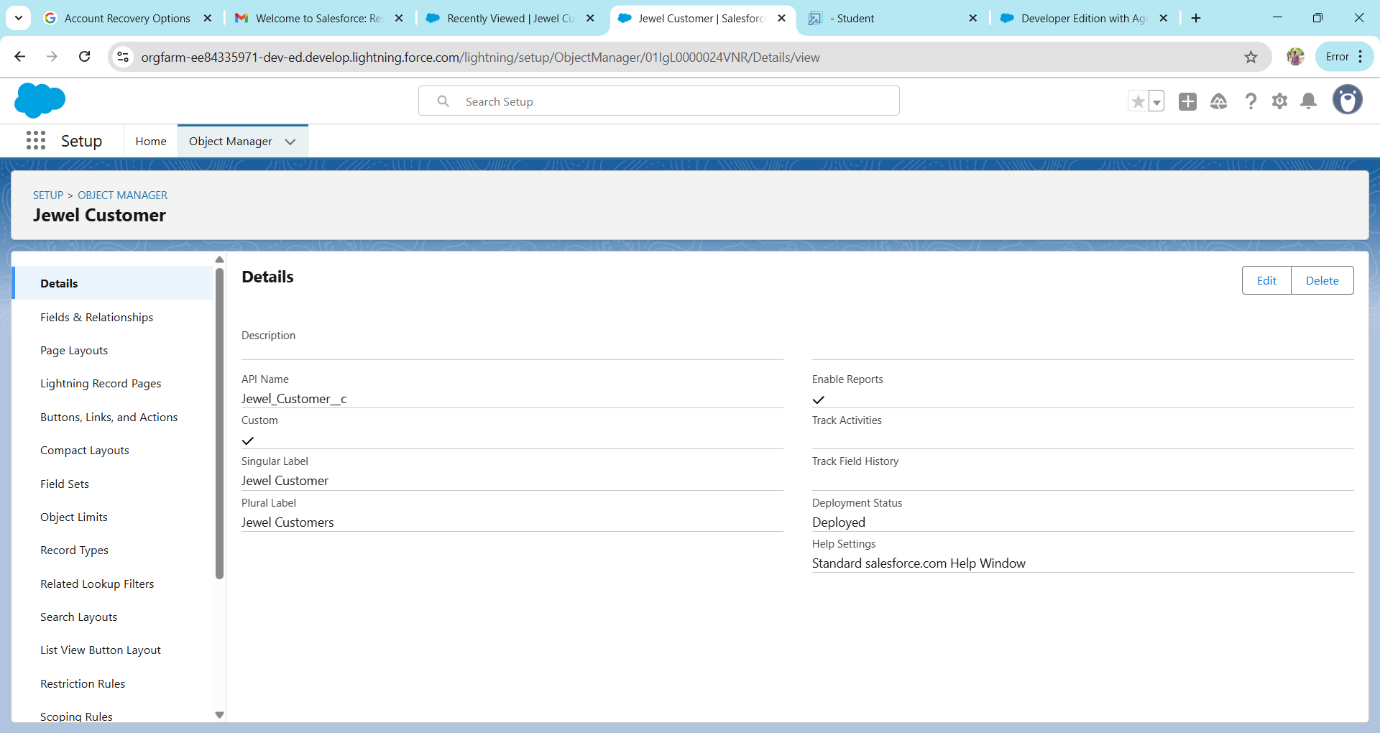
By using this URL <https://developer.salesforce.com/signup>

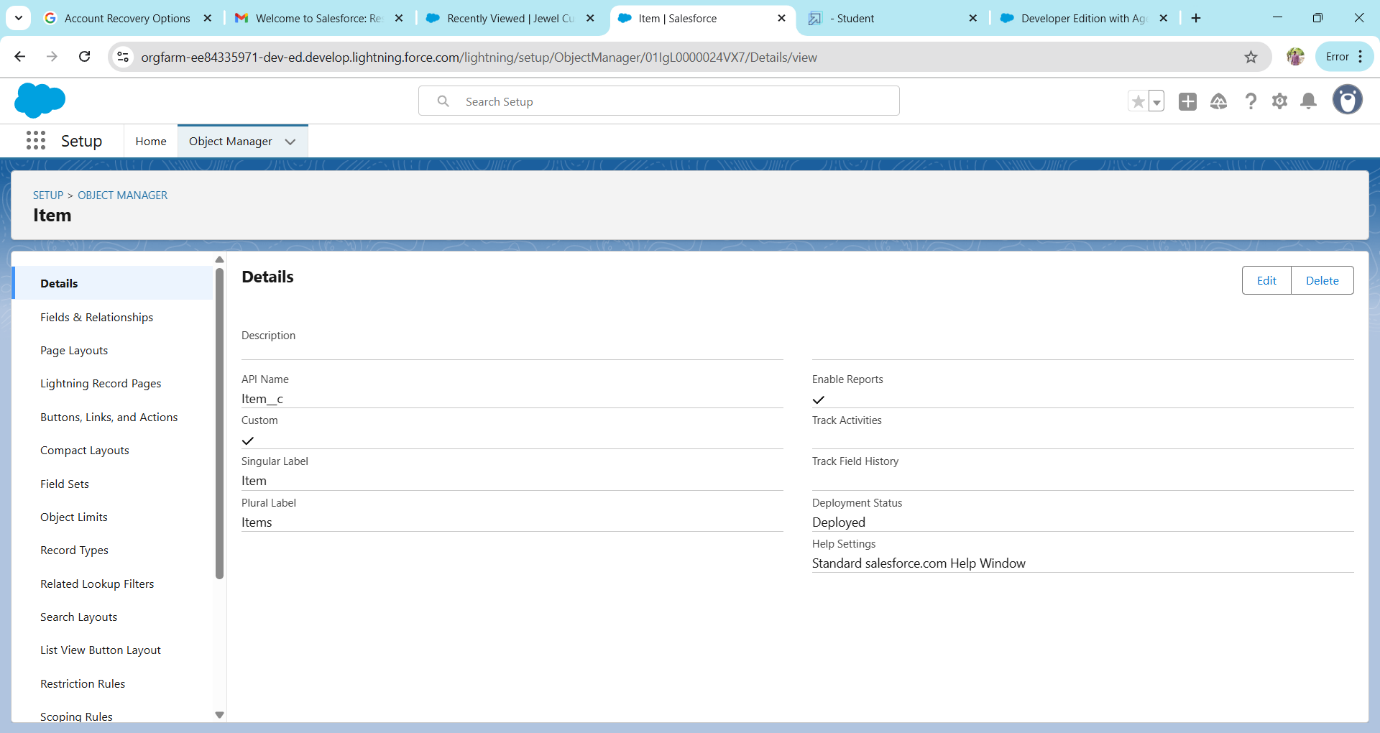
****

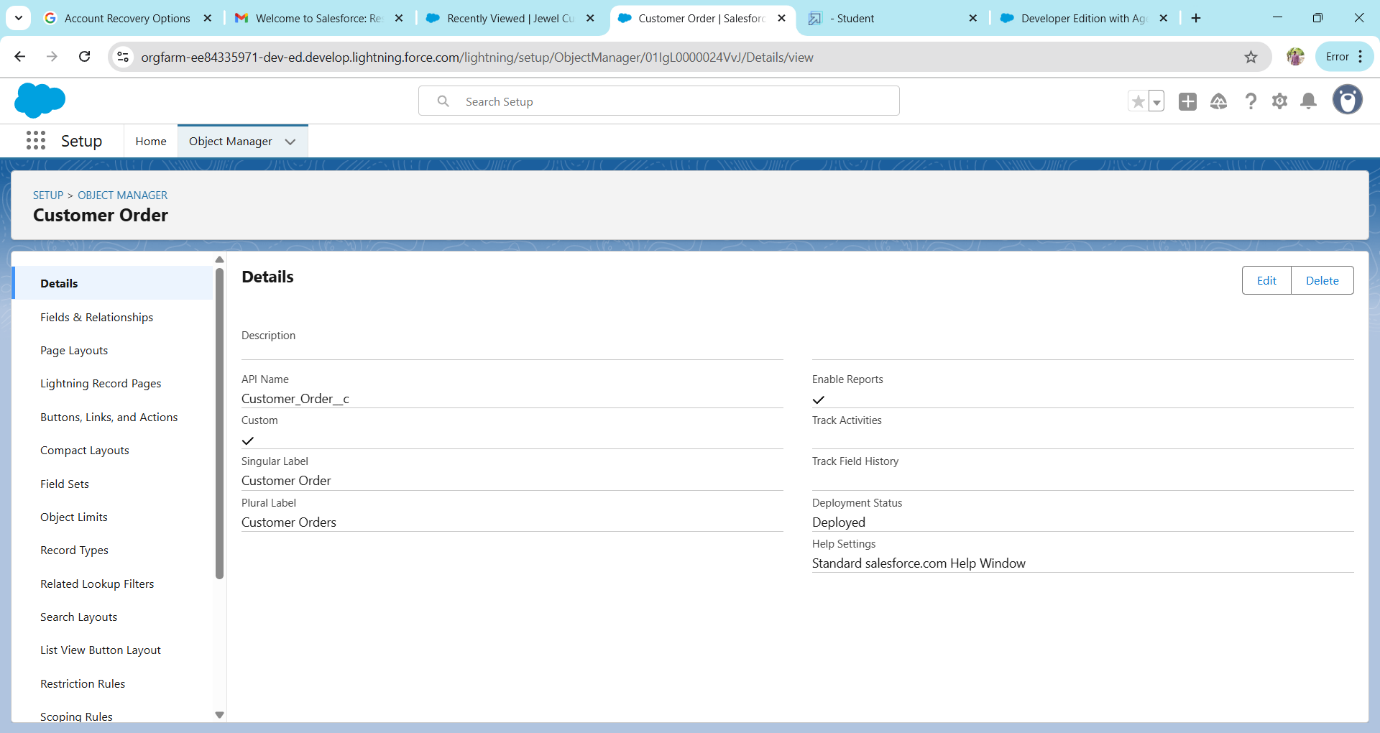
Account activation

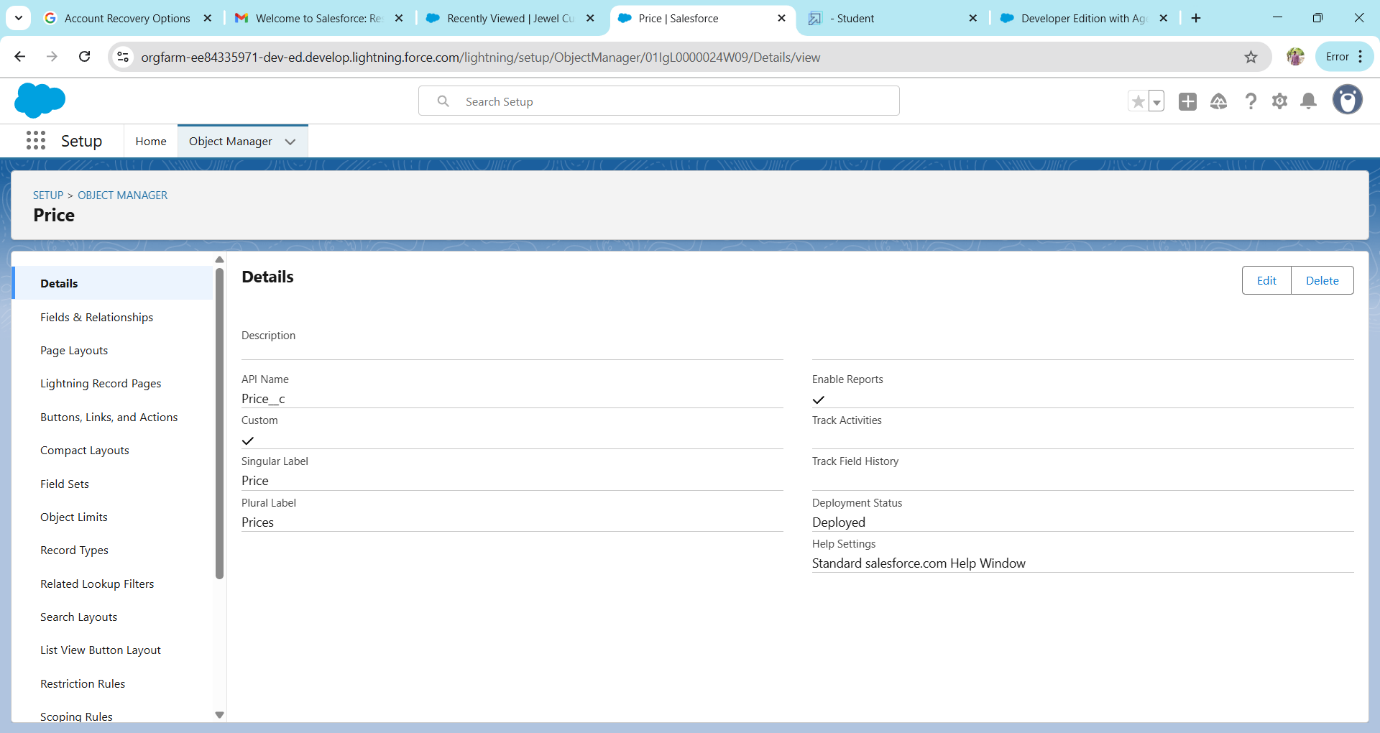


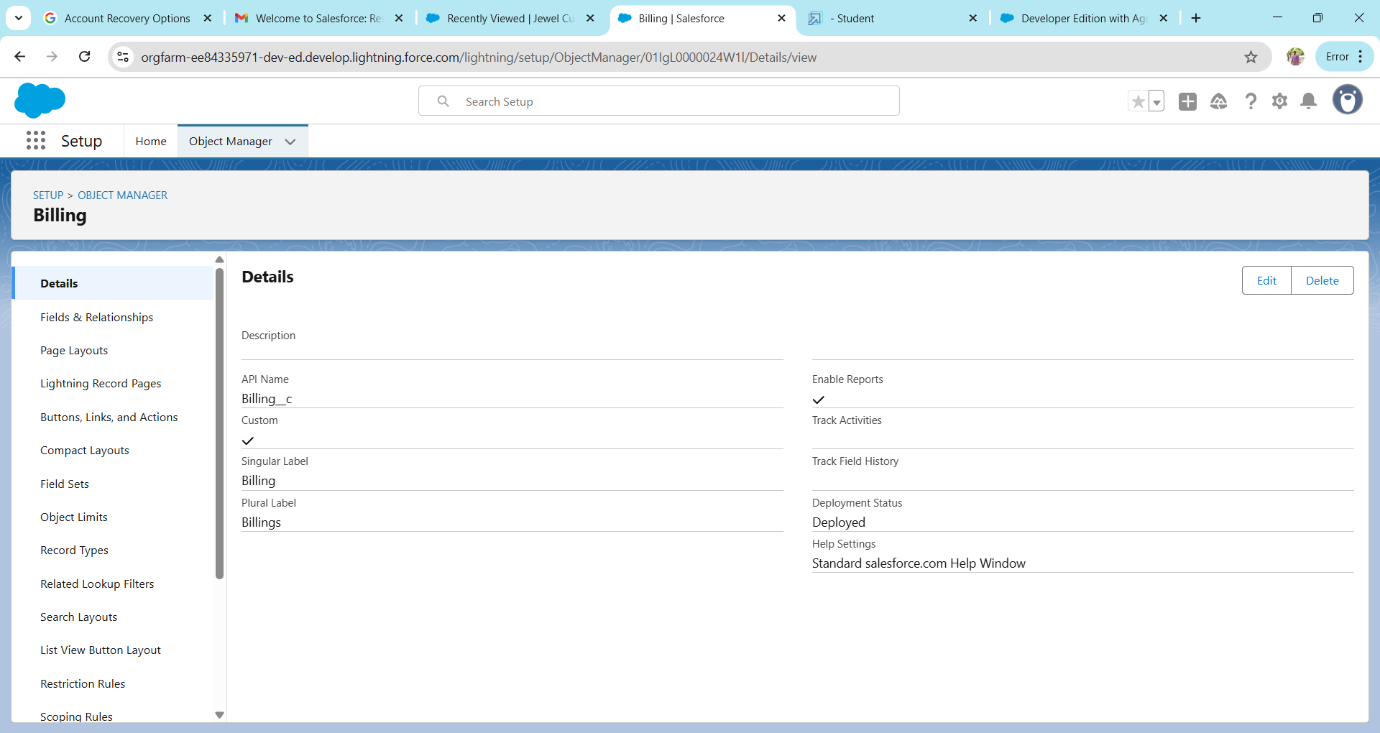
Create objects: Jewel Customer, Item, Customer Order, Price and Billing



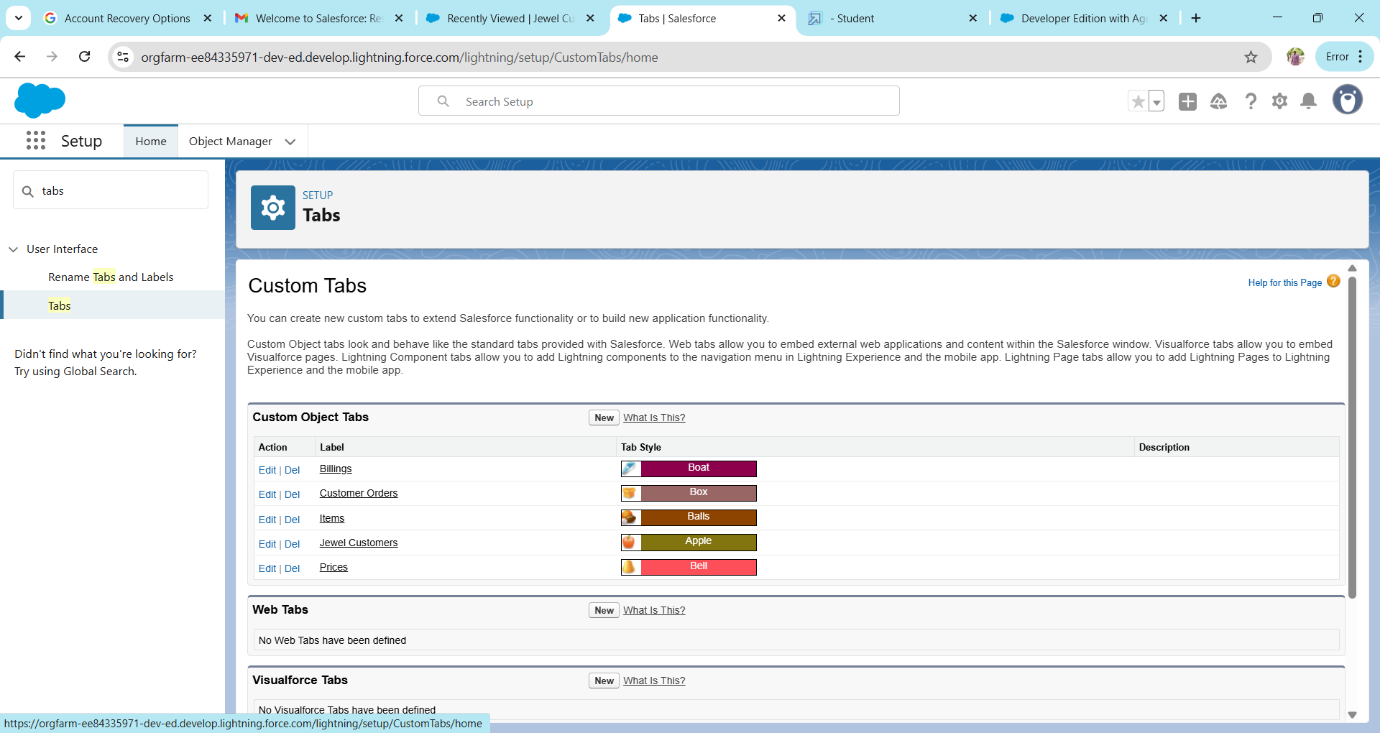




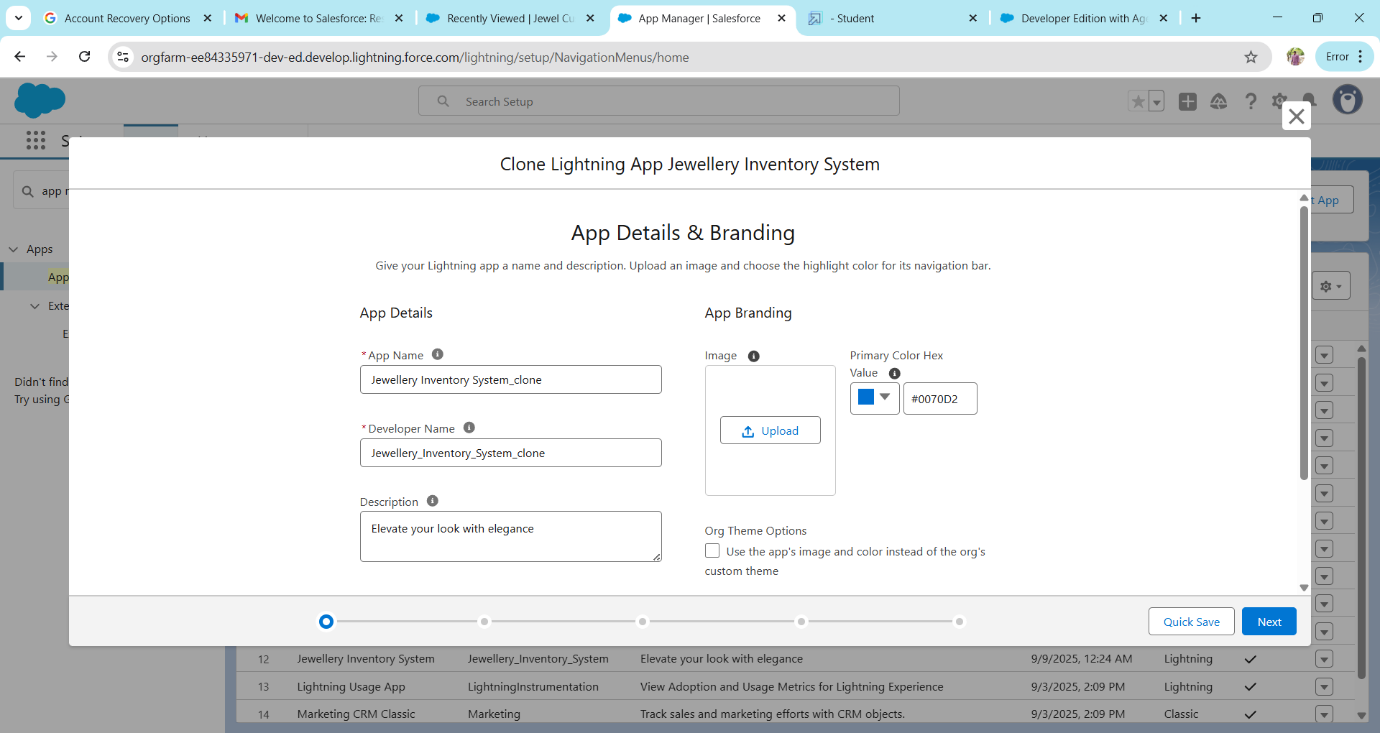


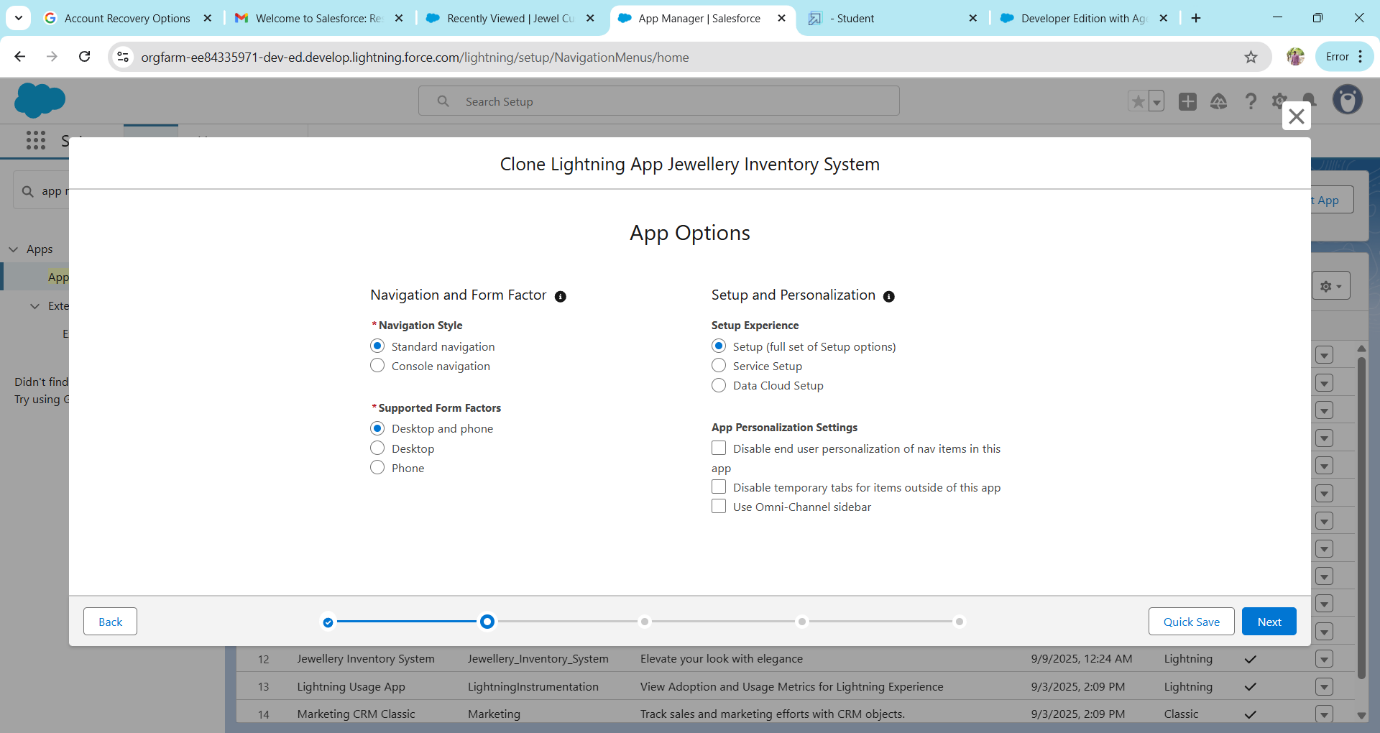


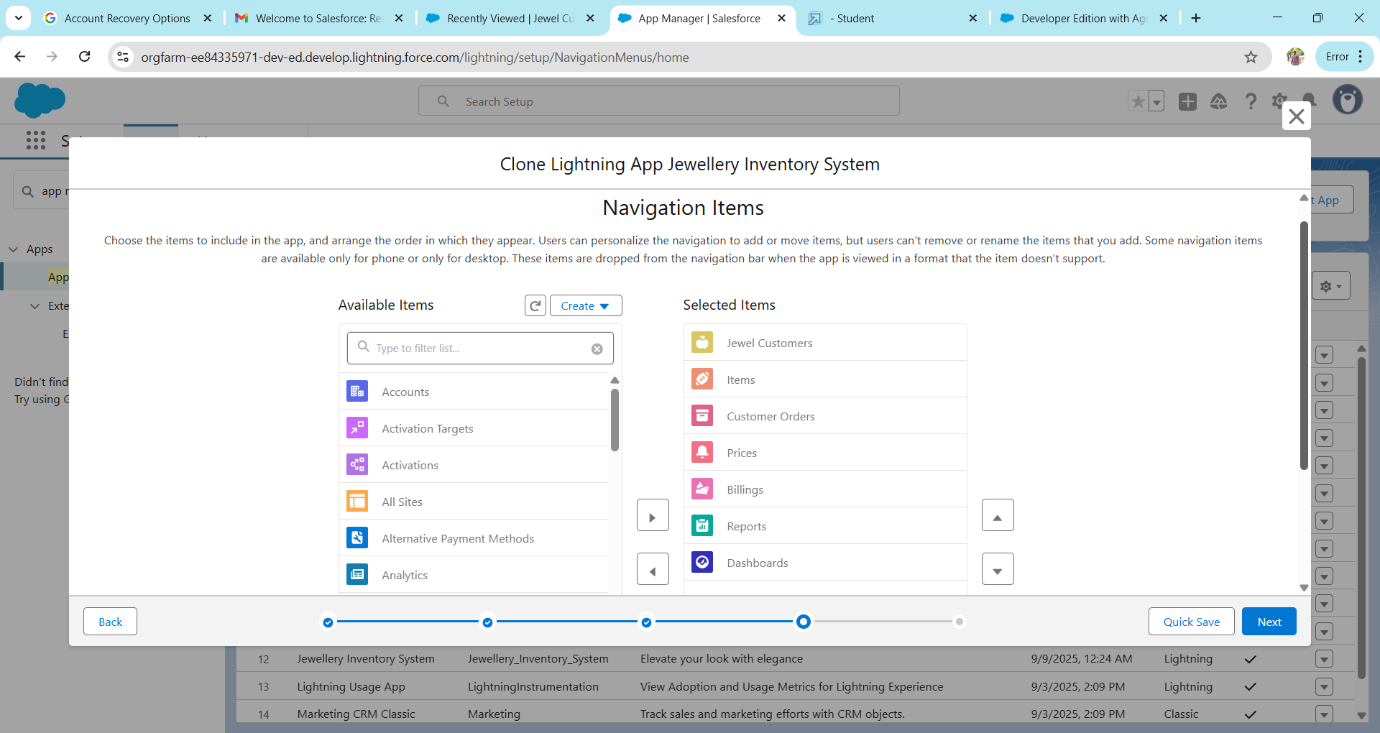
Create a Tabs

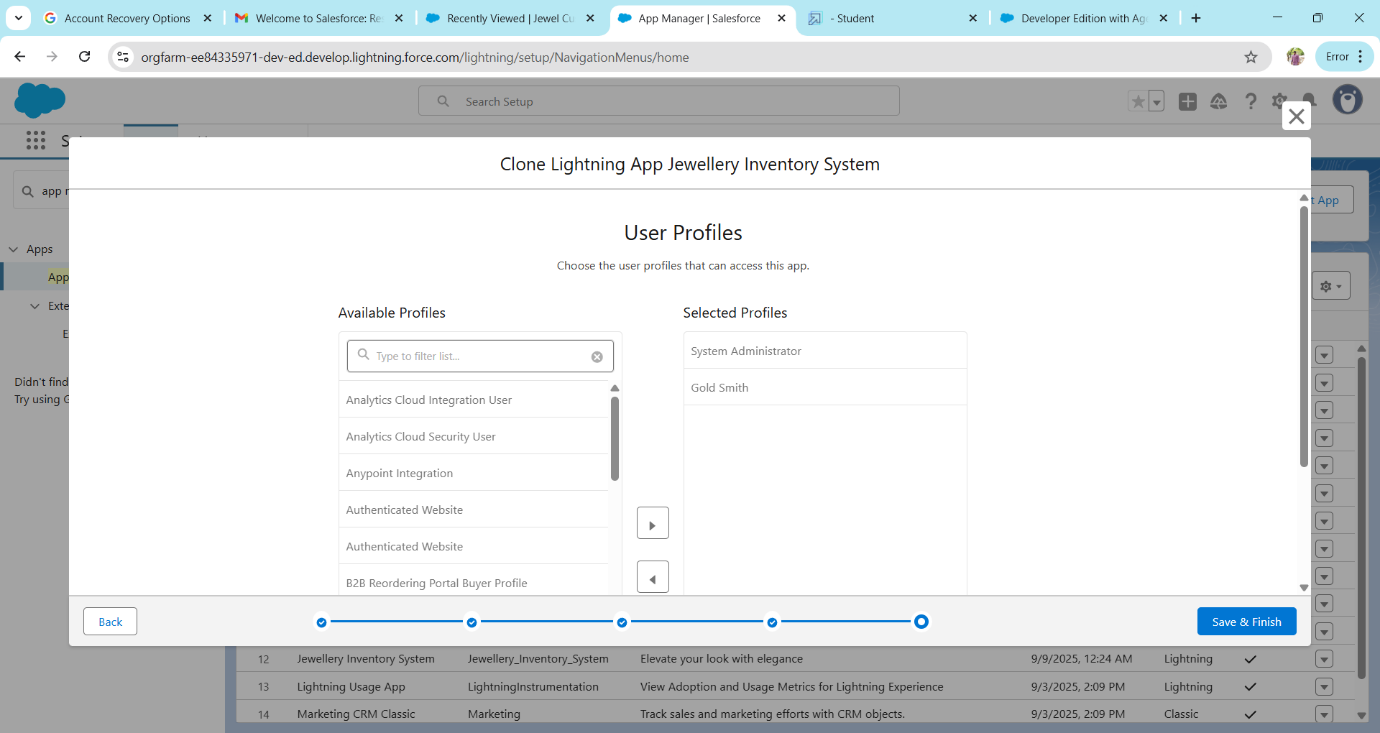


Create a Lightning app



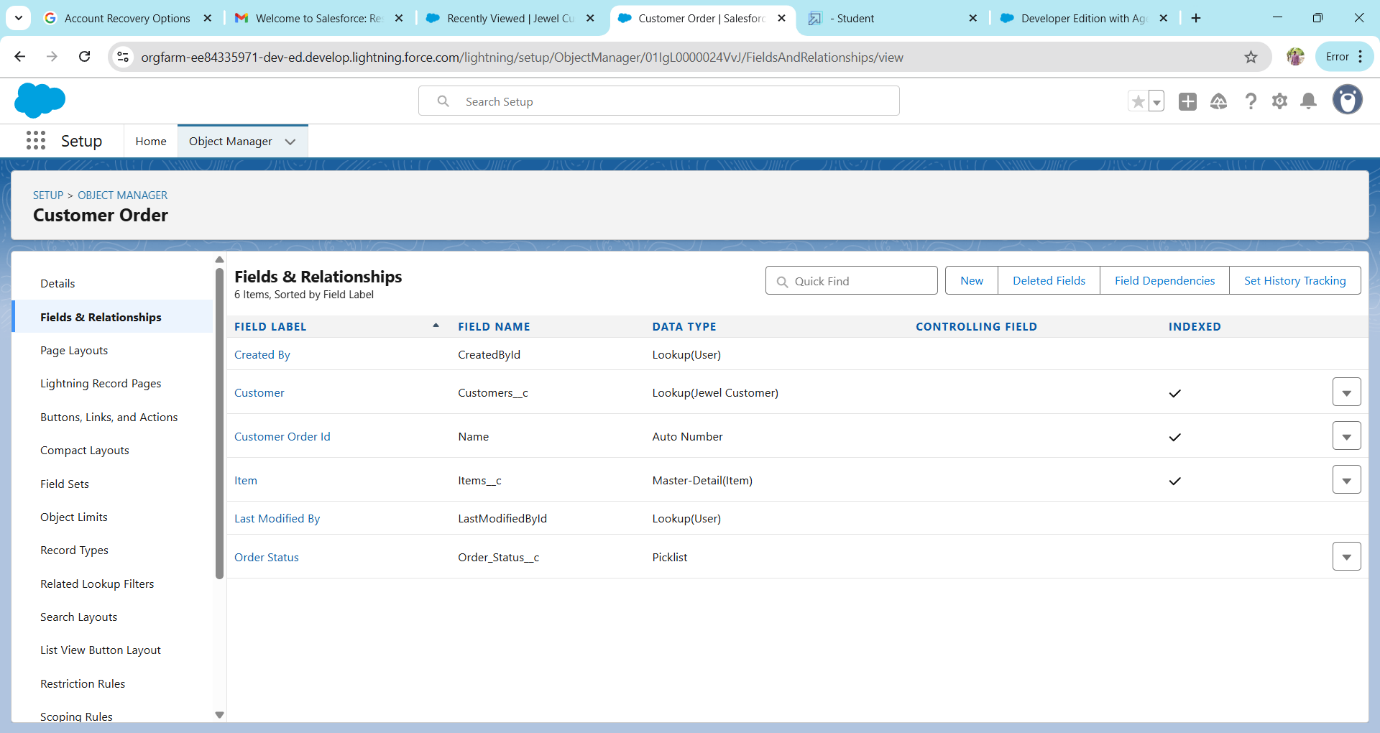




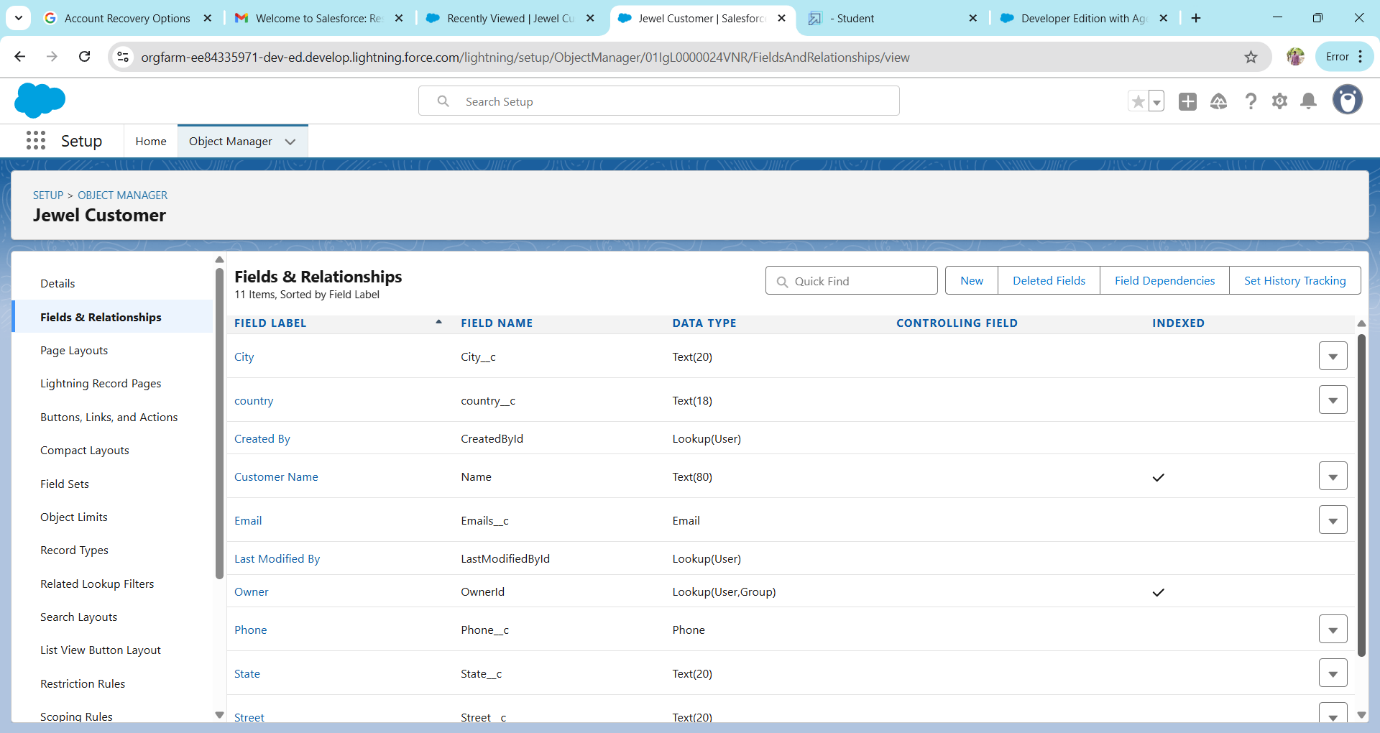


**Creating a Field**

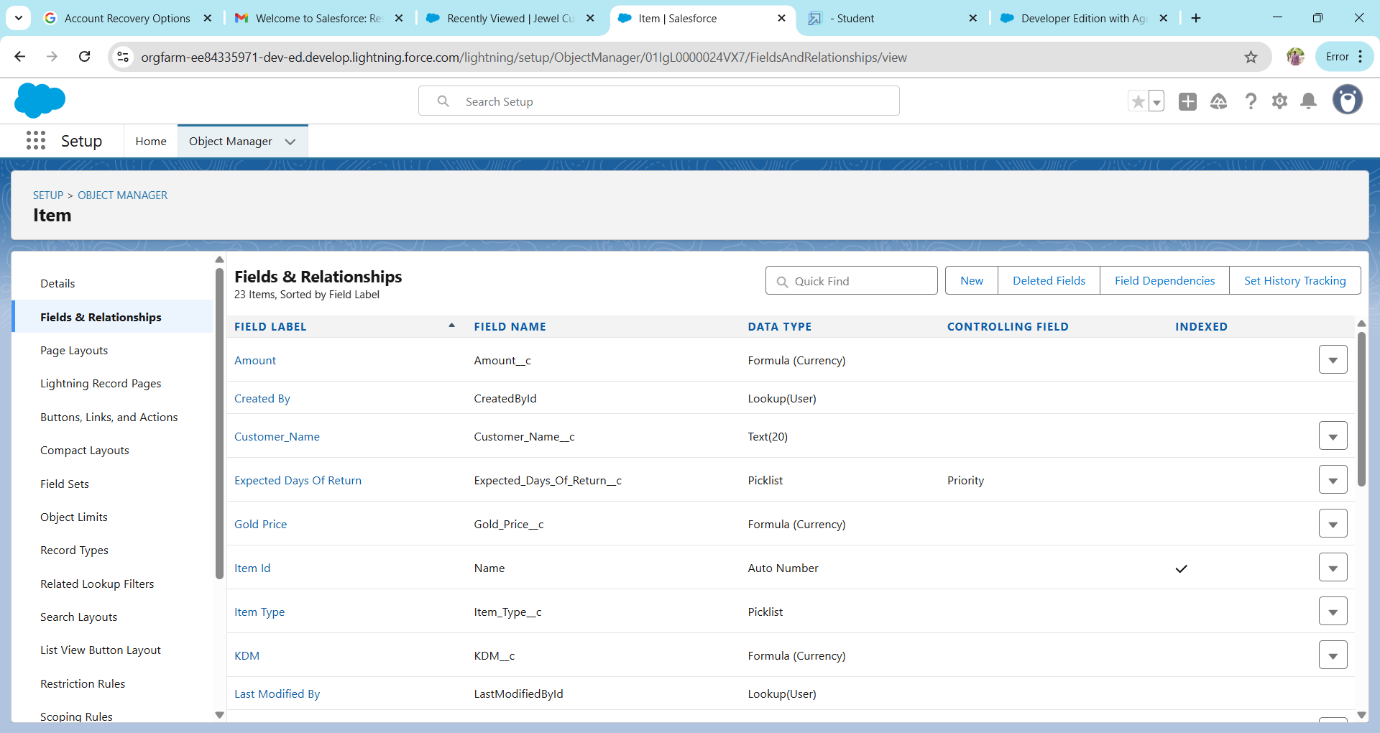
Creating a field in Customer Order Object



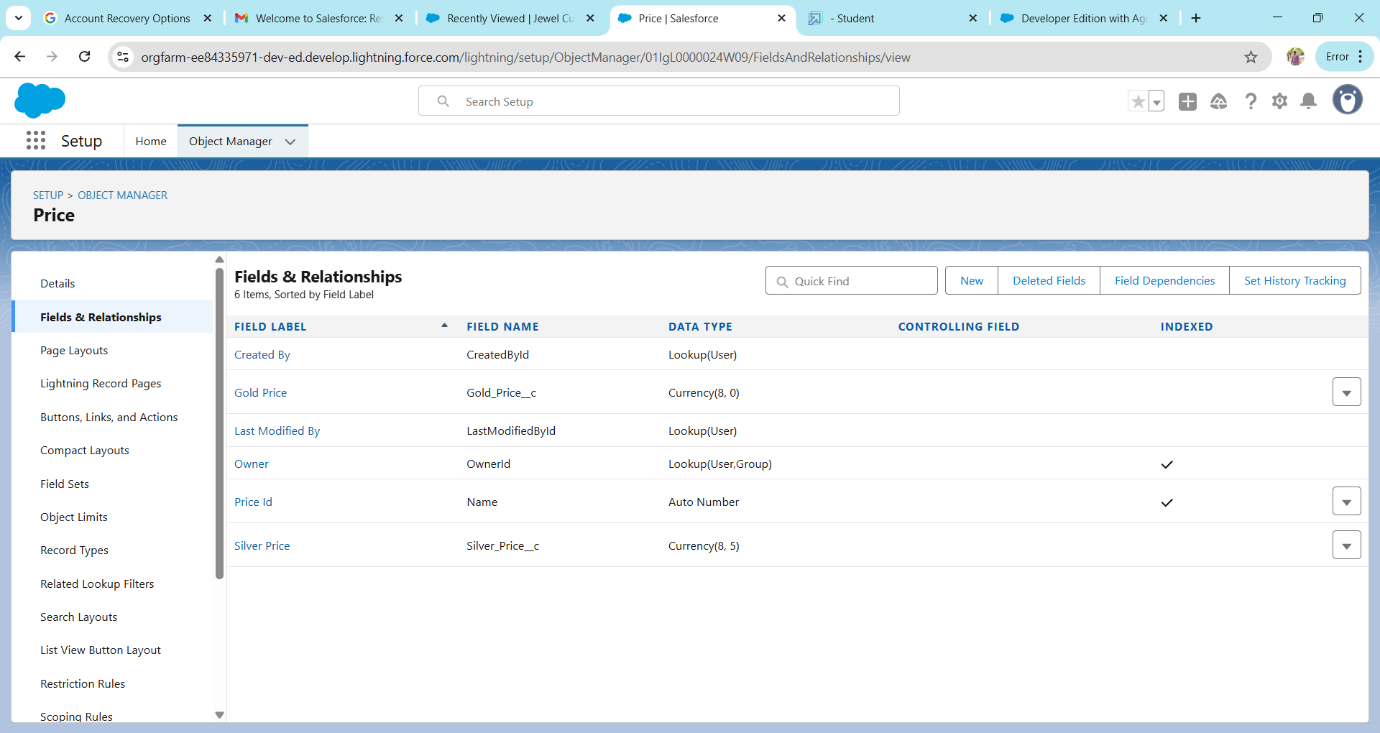
Create a field in Jewel Customer Objects



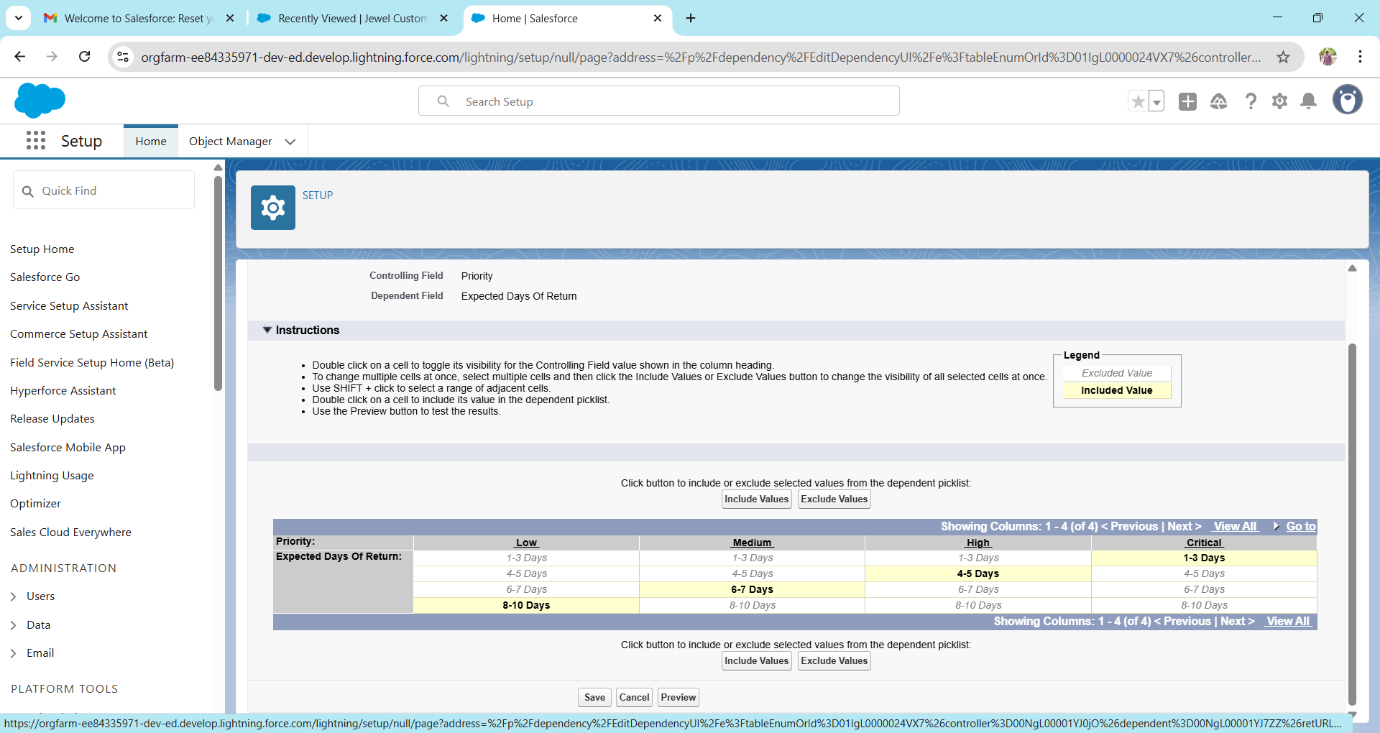
Create a field In Item Objects



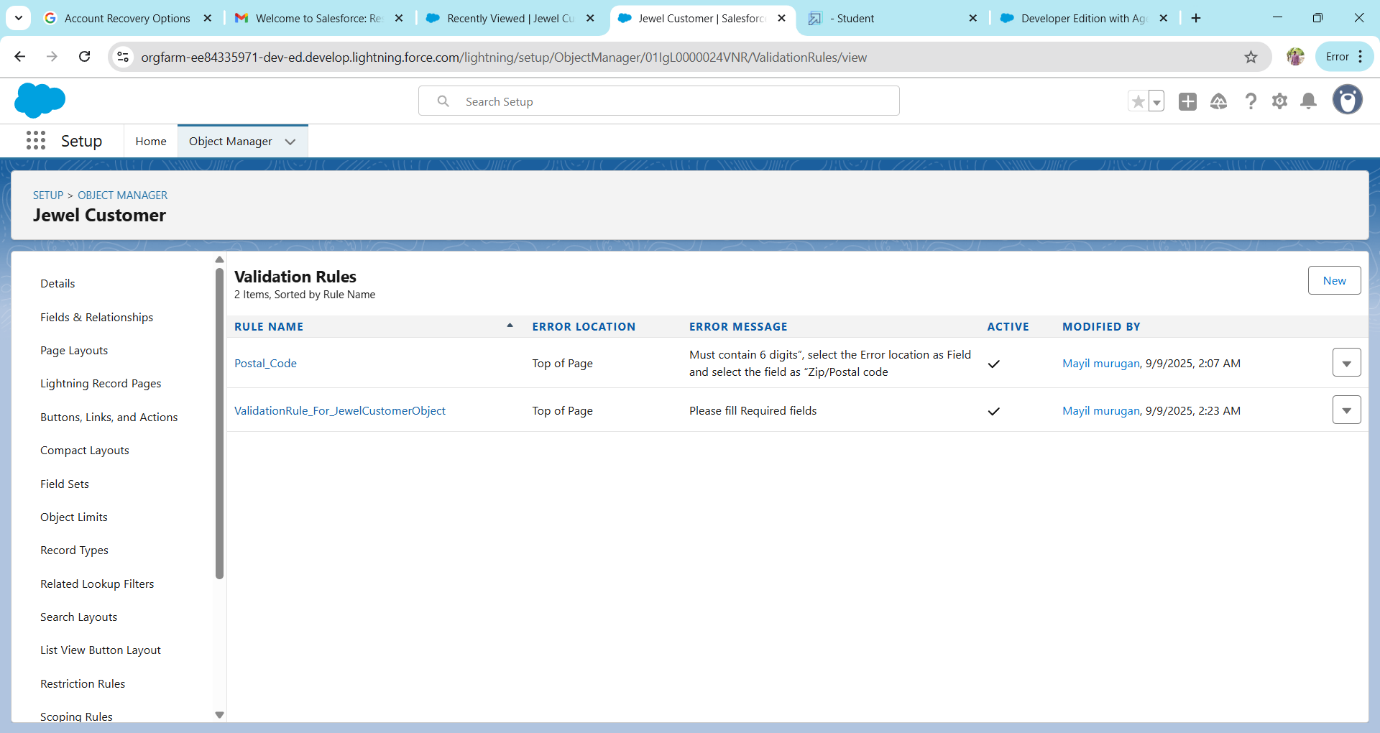
Create a field in Price Object



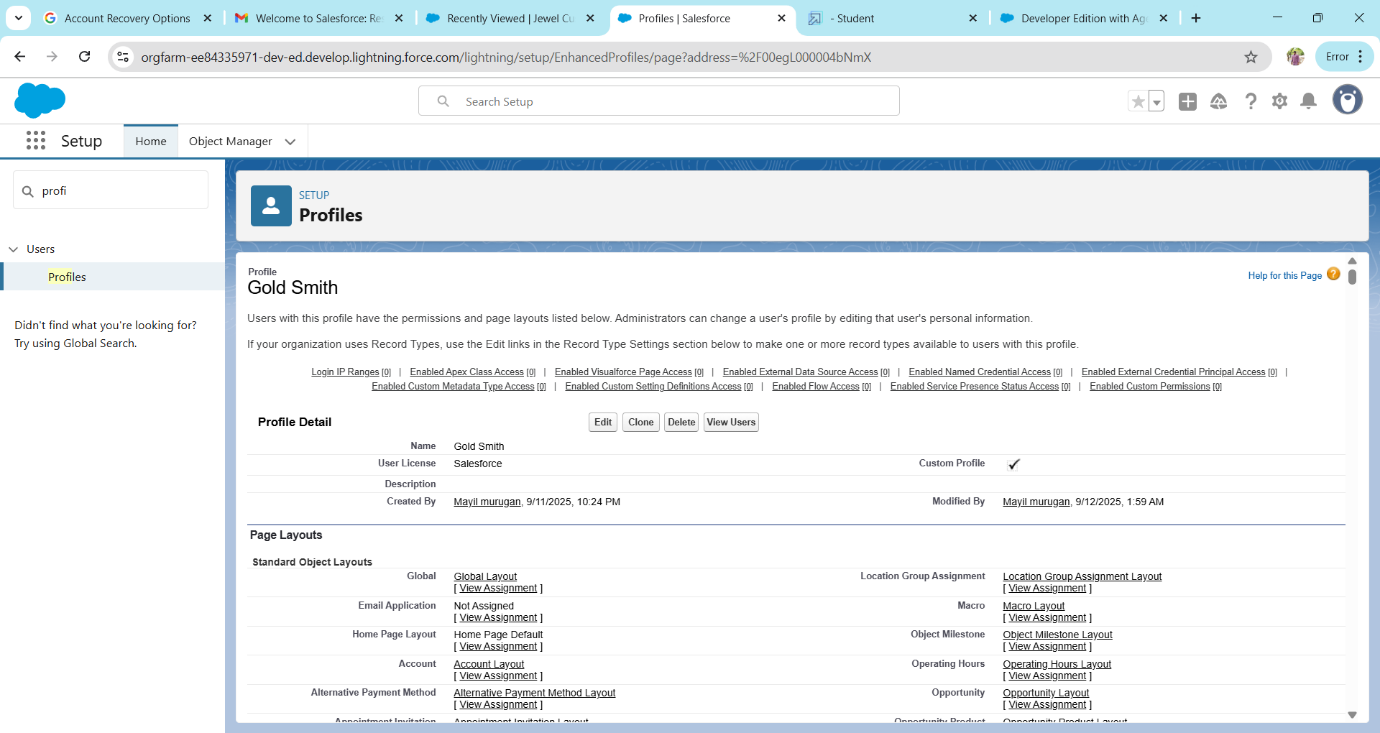
Creating Field dependencies

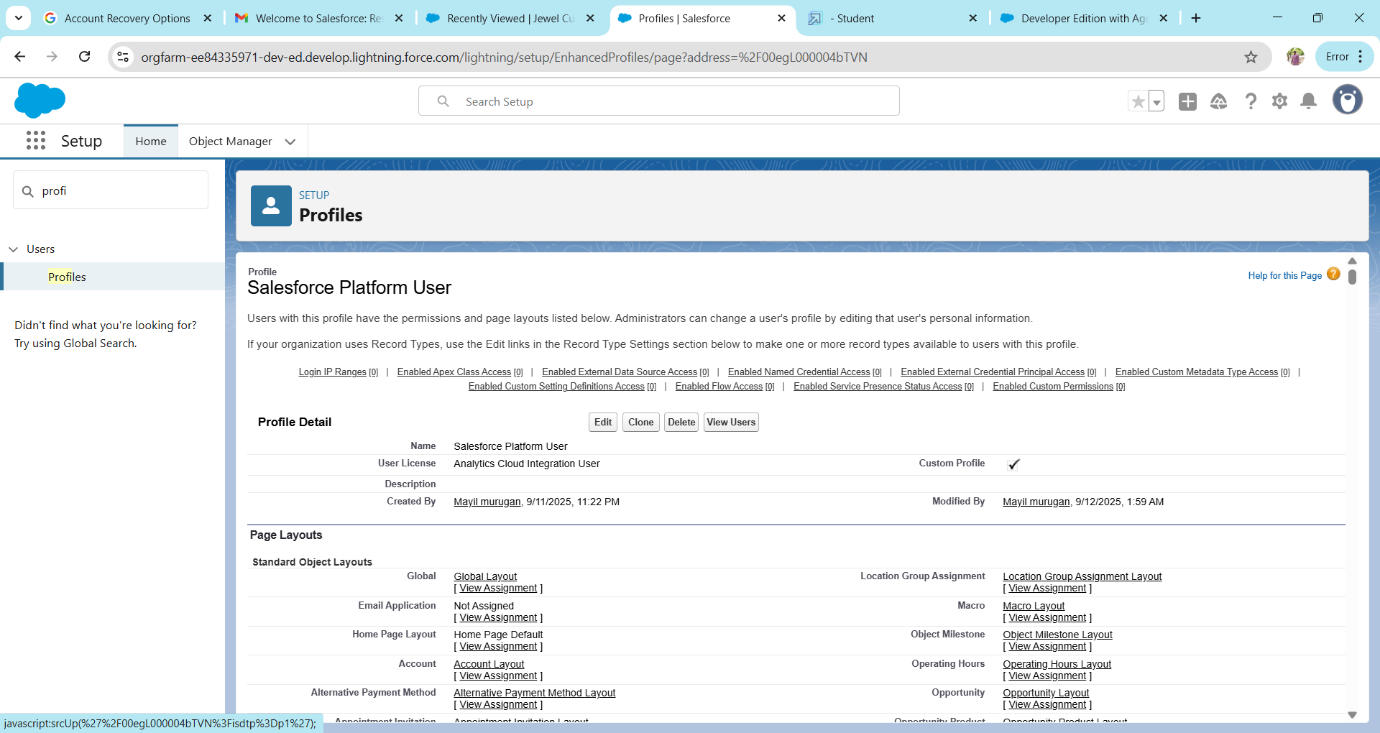


Creating a Validation Rule under a Jewel Customer object

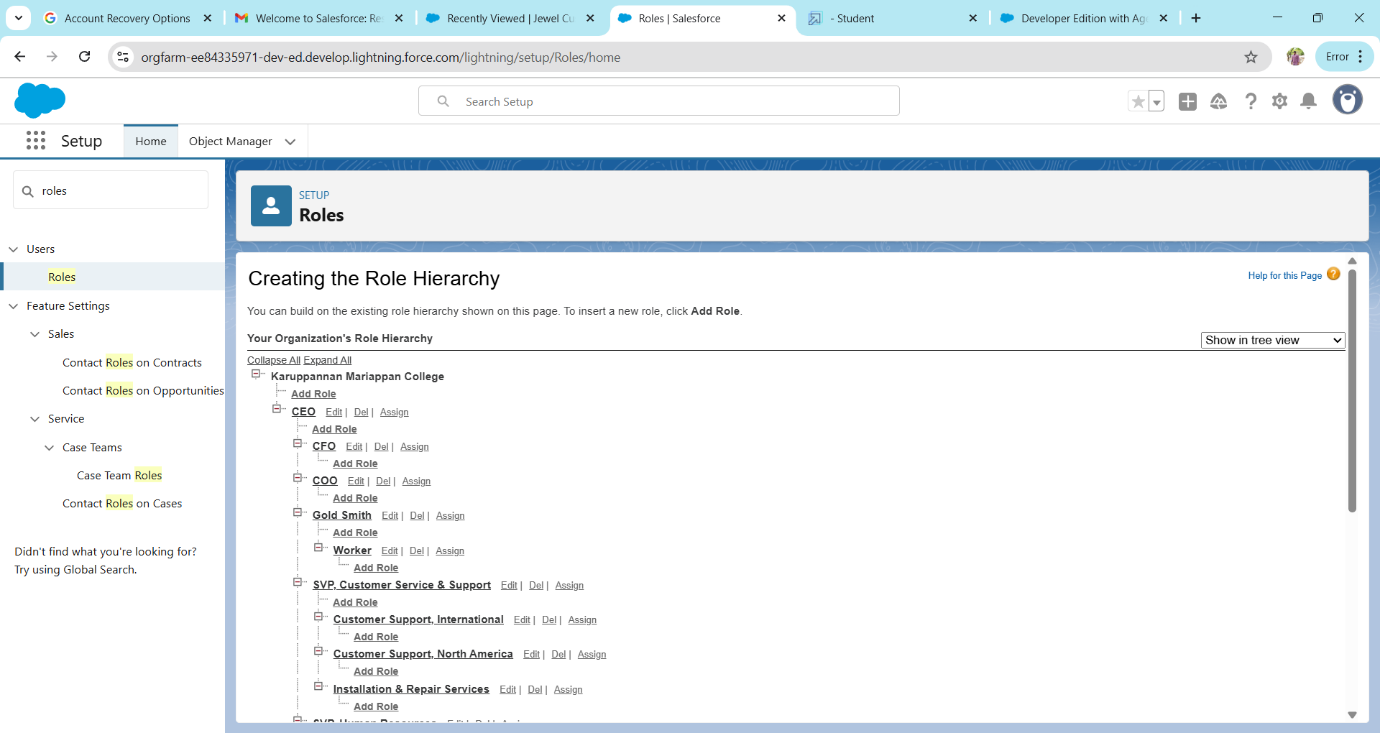


Creating a Gold Smith and Worker profiles

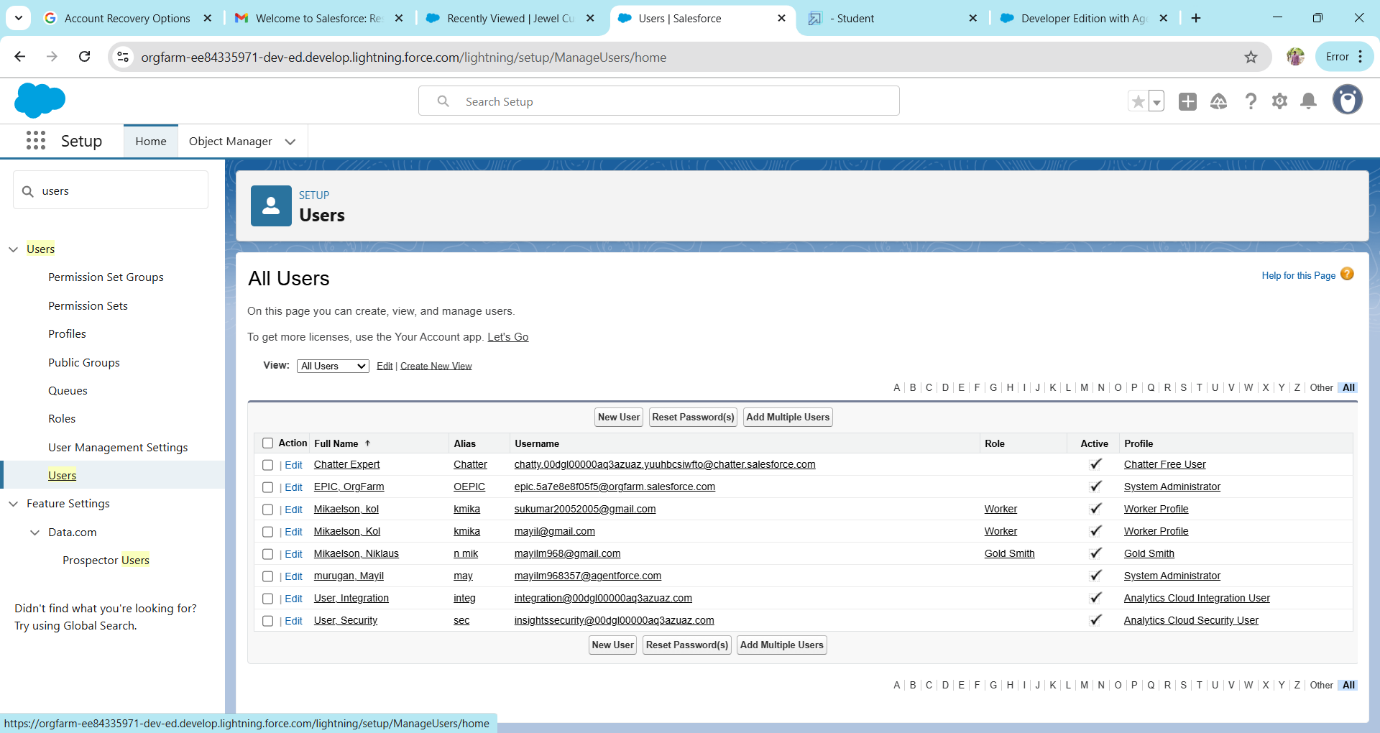




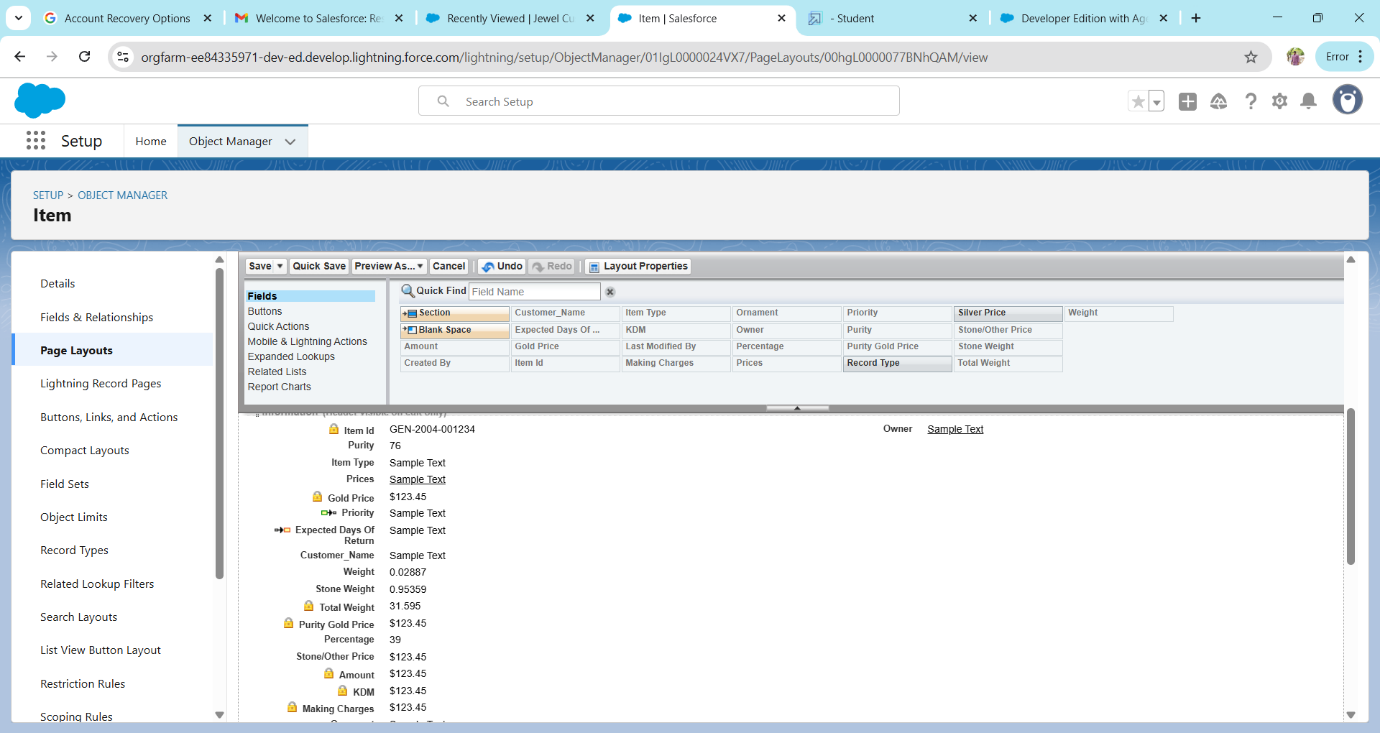
Creating Gold Smith Role



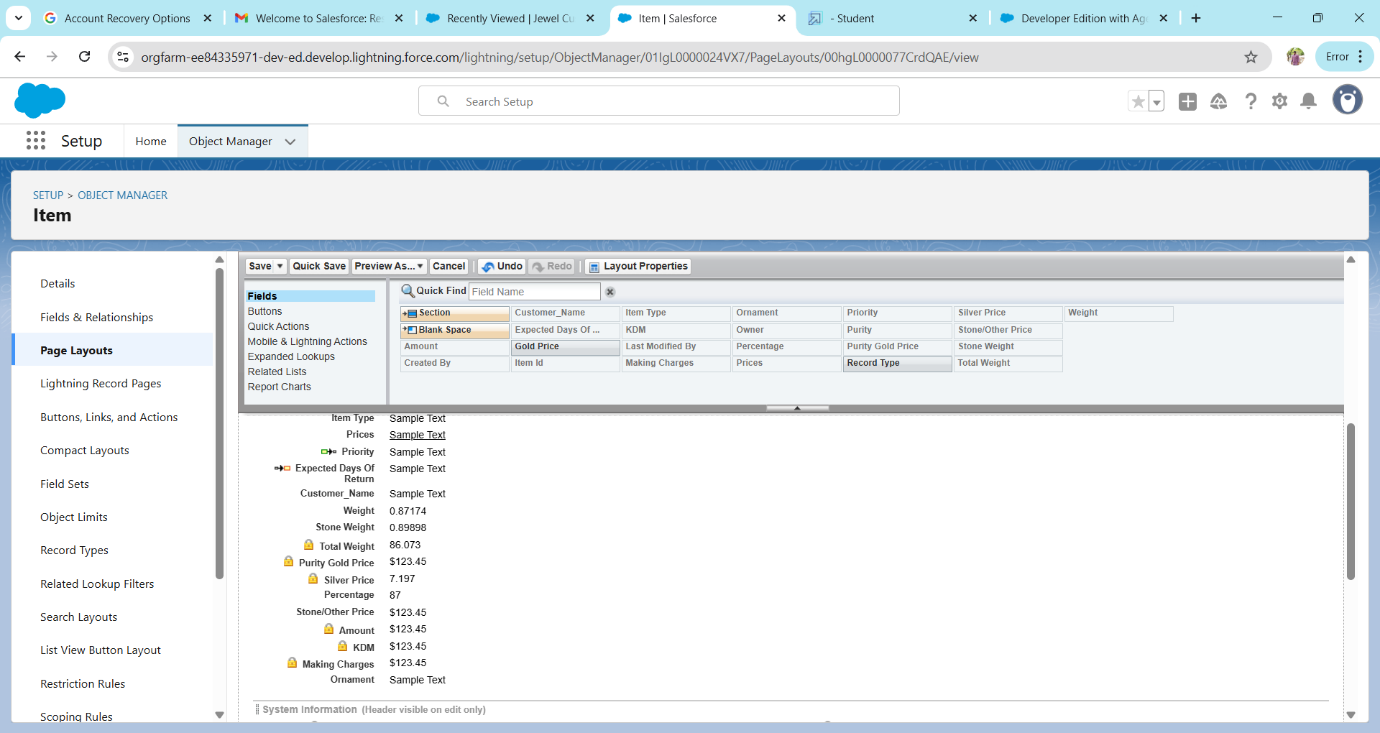
Create User



To Create a Gold page layout

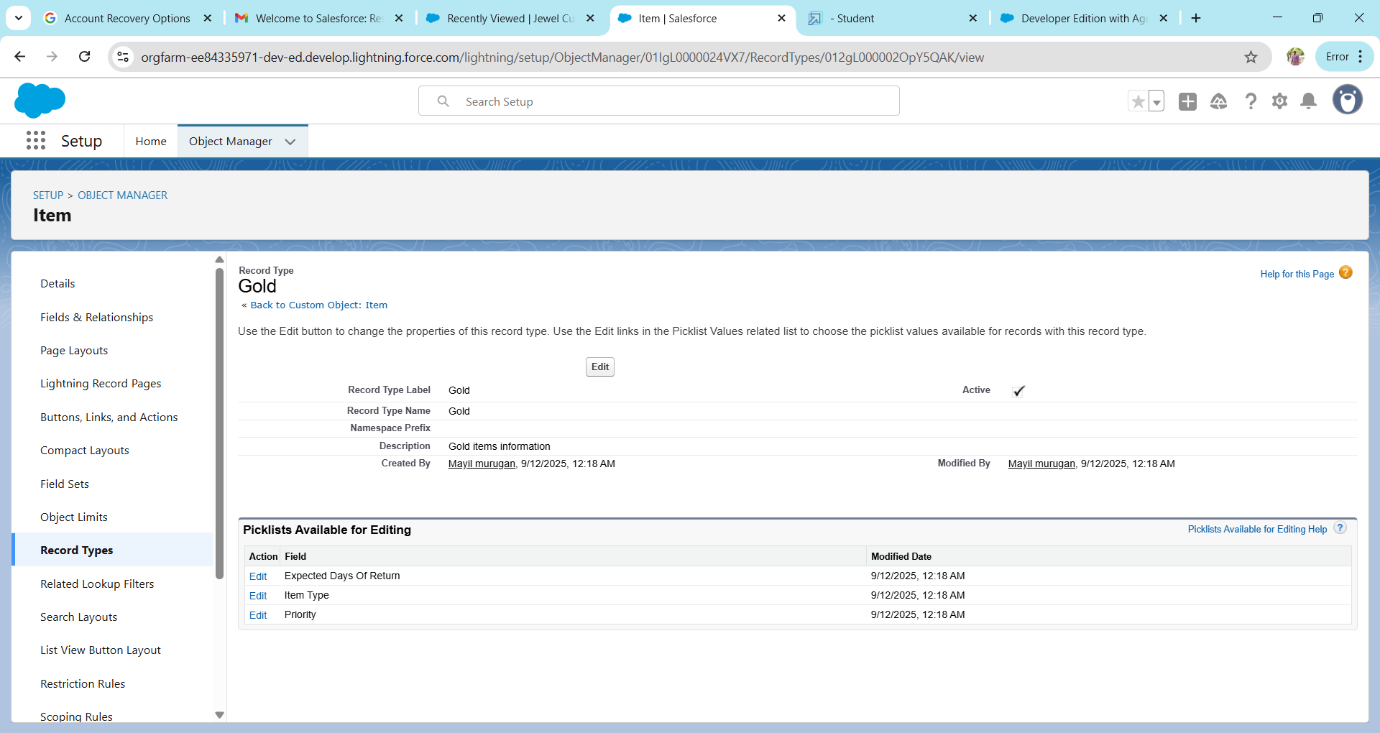


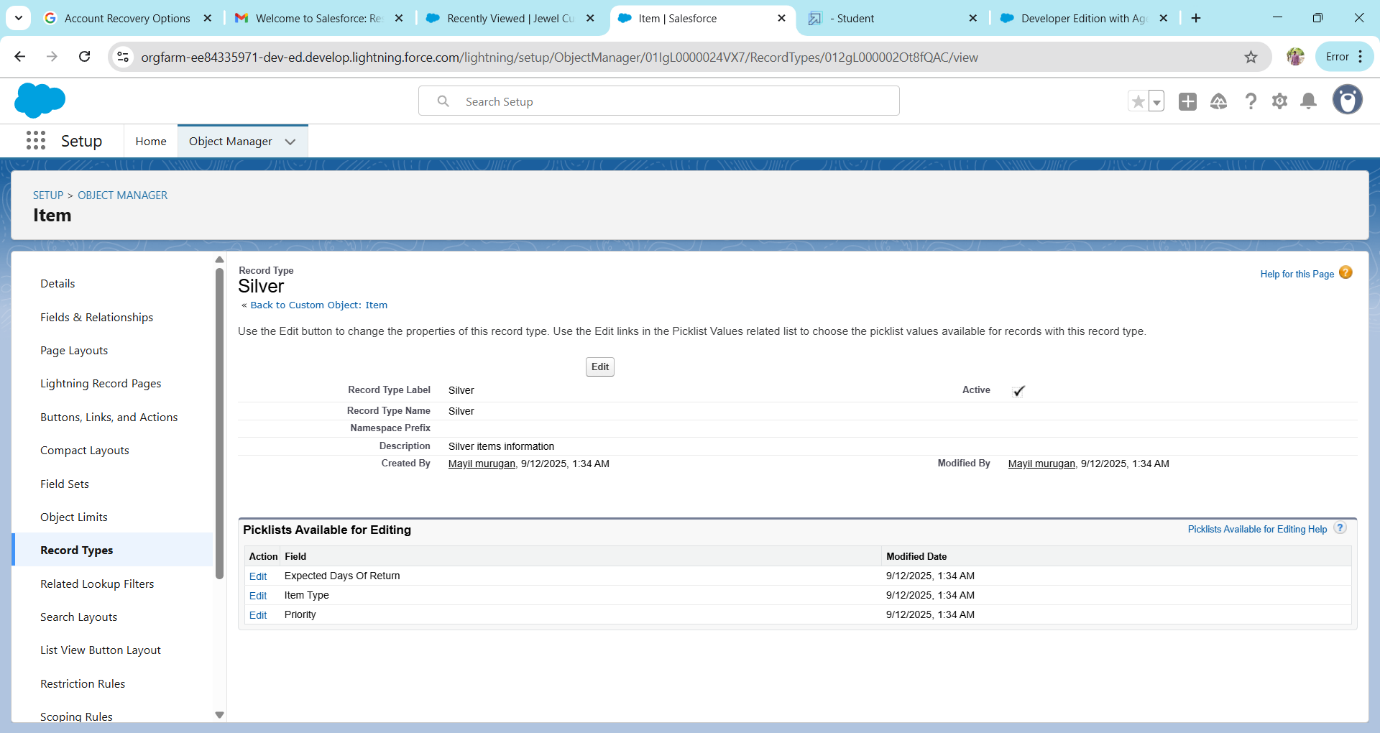
To Create a Silver page layout



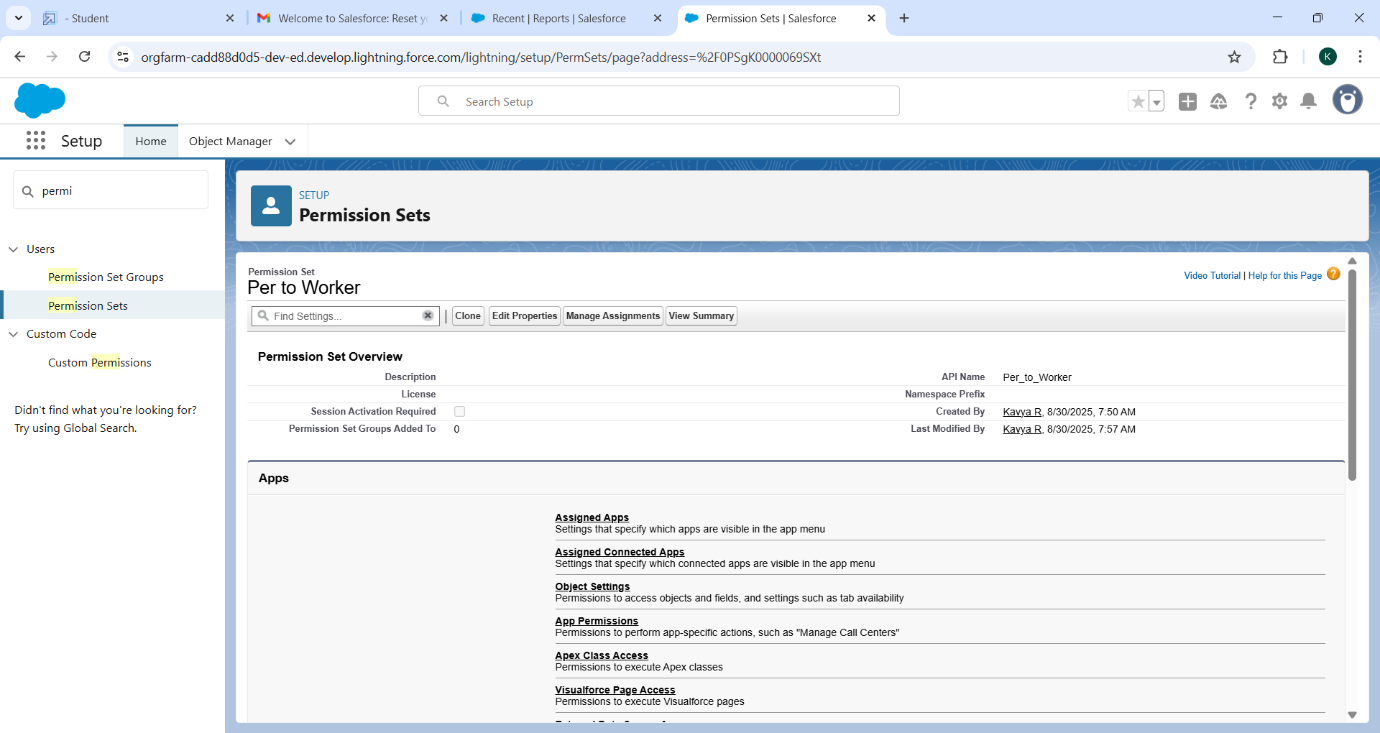
**Record Type**

Create a gold and silver record types under the Item object



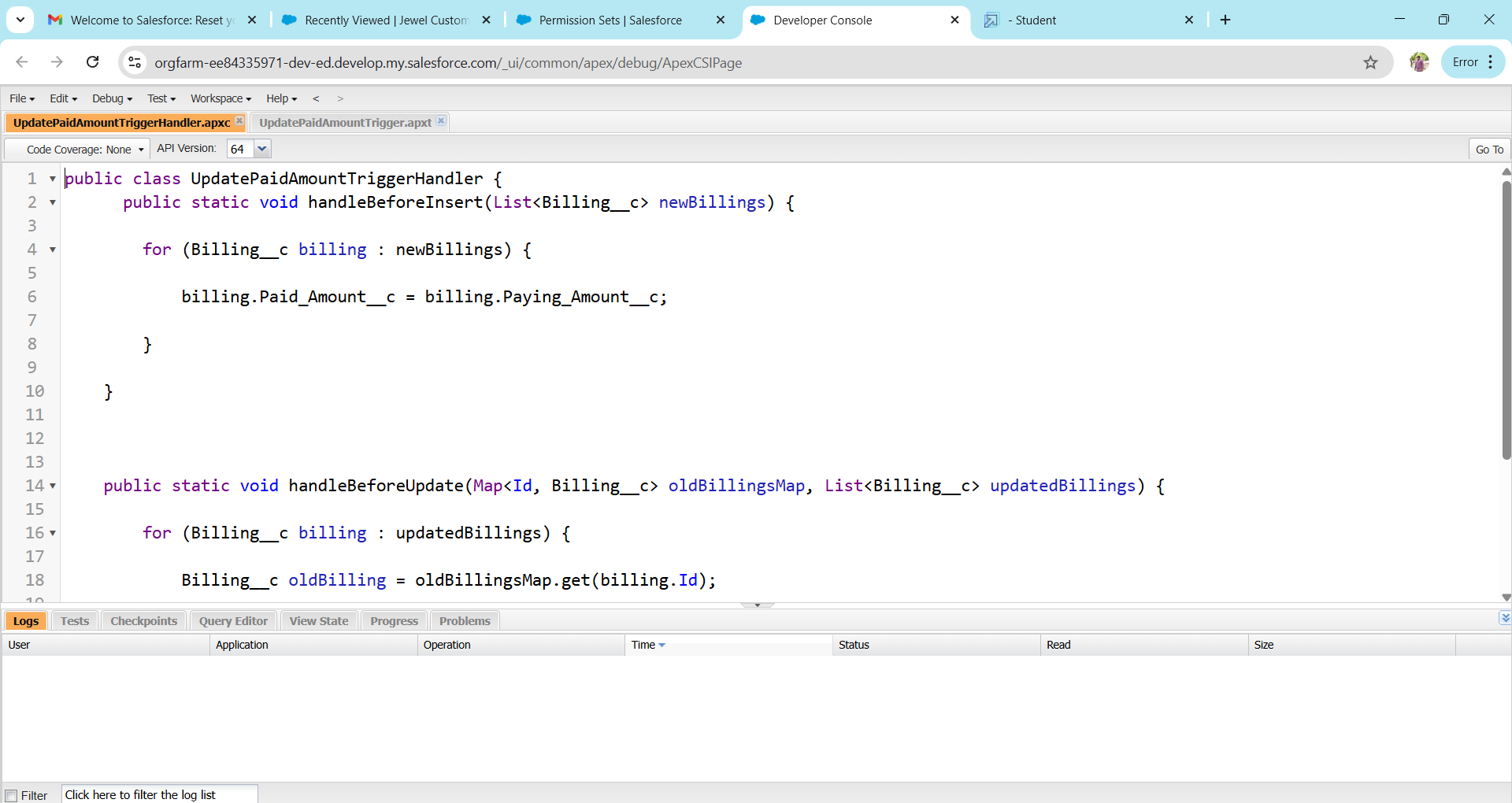


Creating Permission set

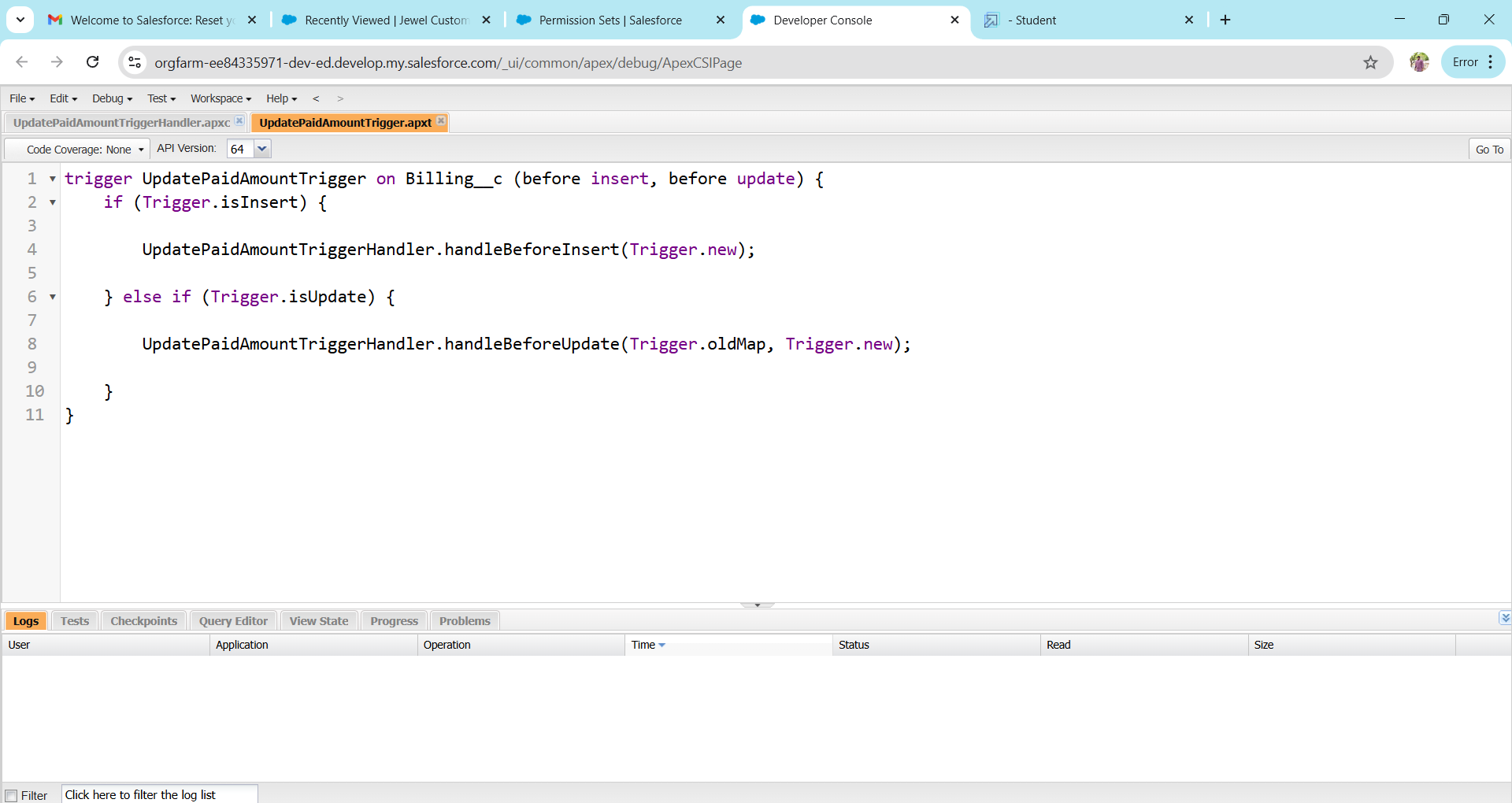


**Trigger**

Create a Trigger Handler

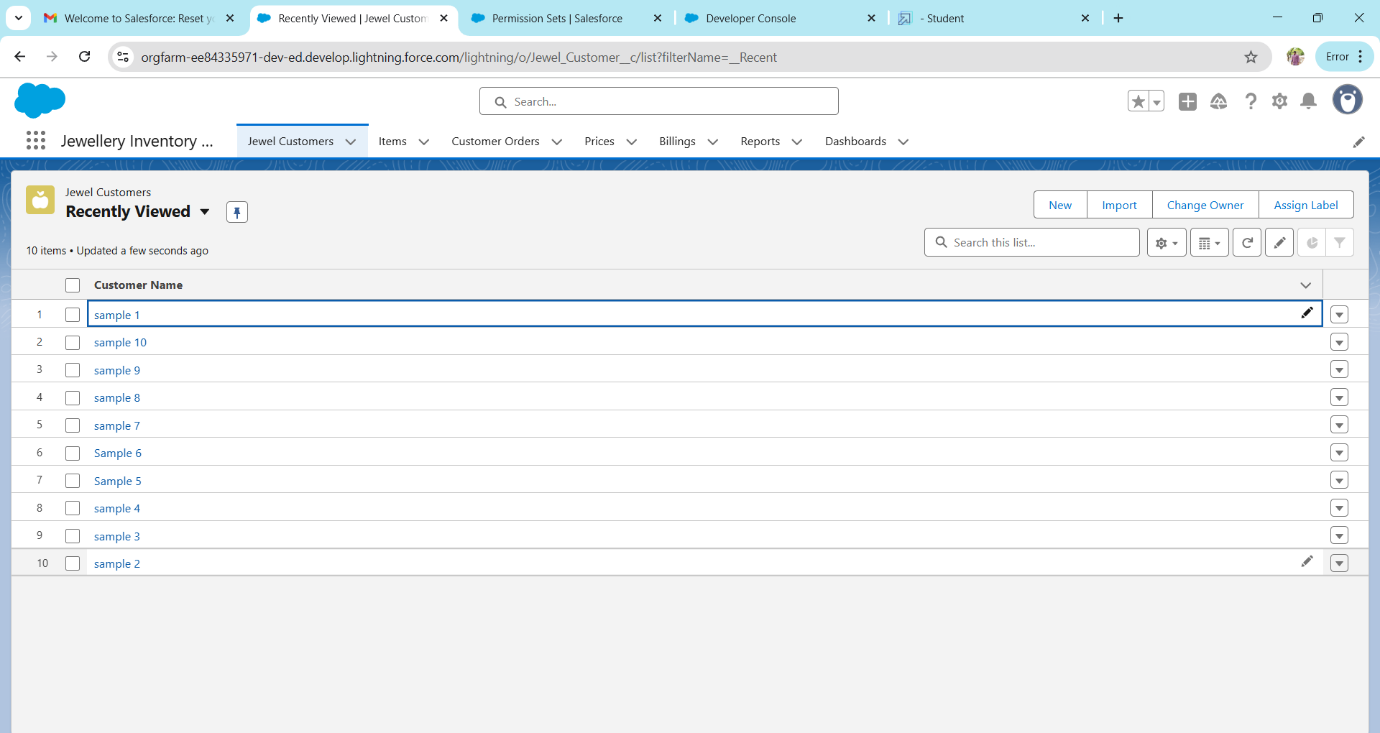


Create a UpdatePaidAmountTrigger on Billing object

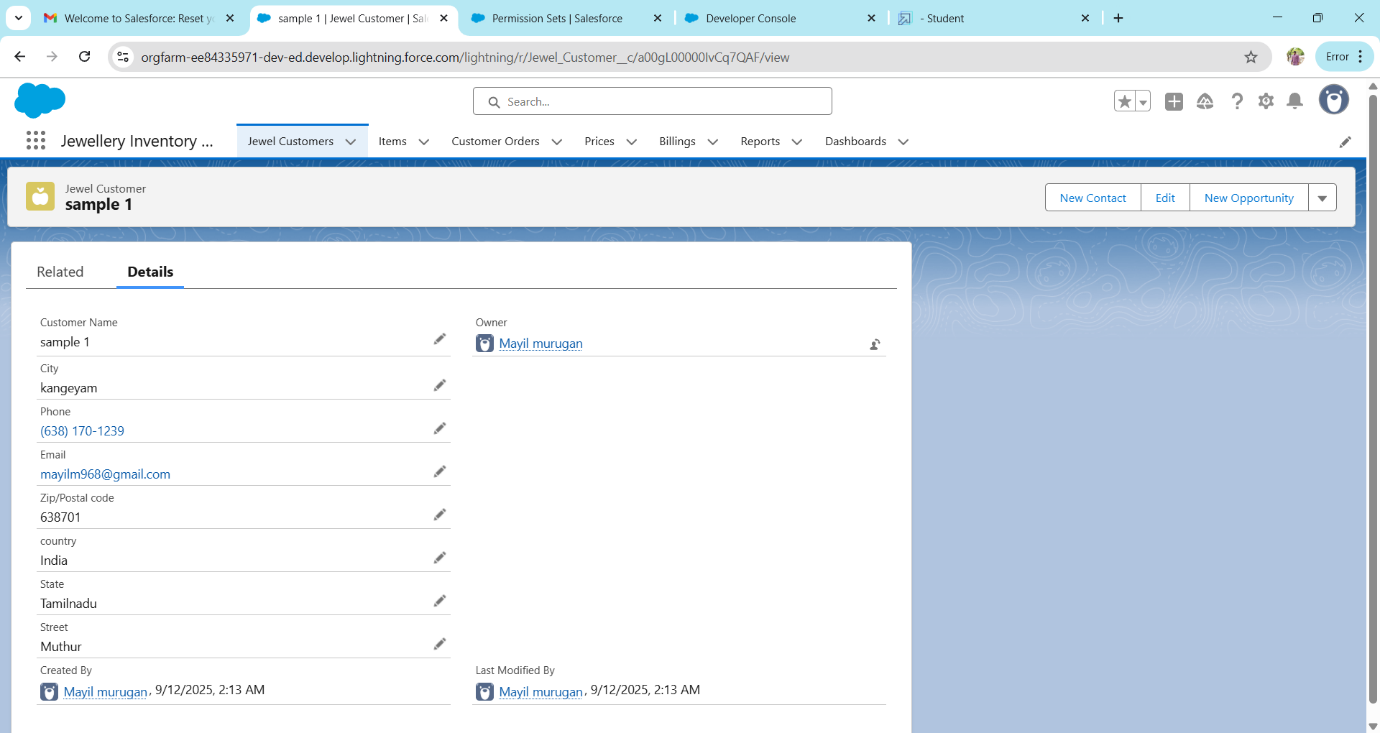


**User Adoption**

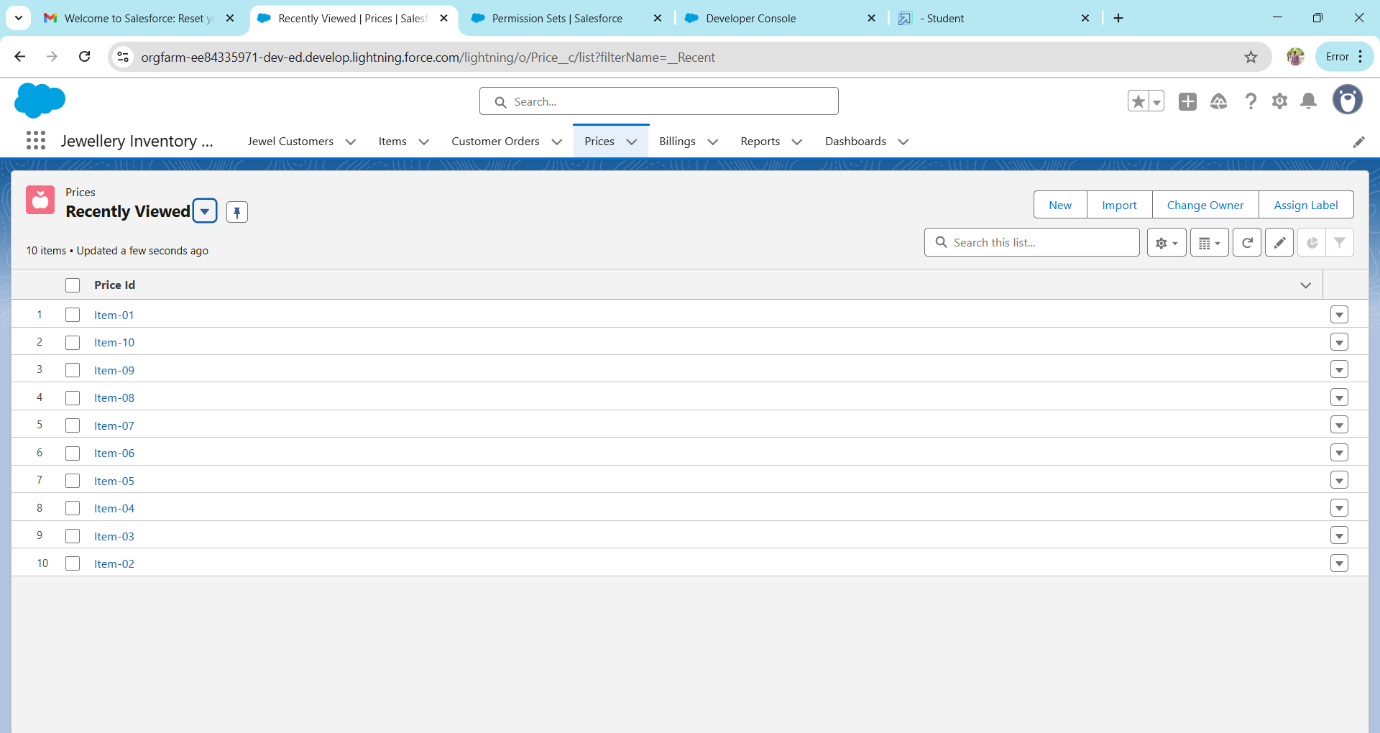
Create a Jewel Customer Record



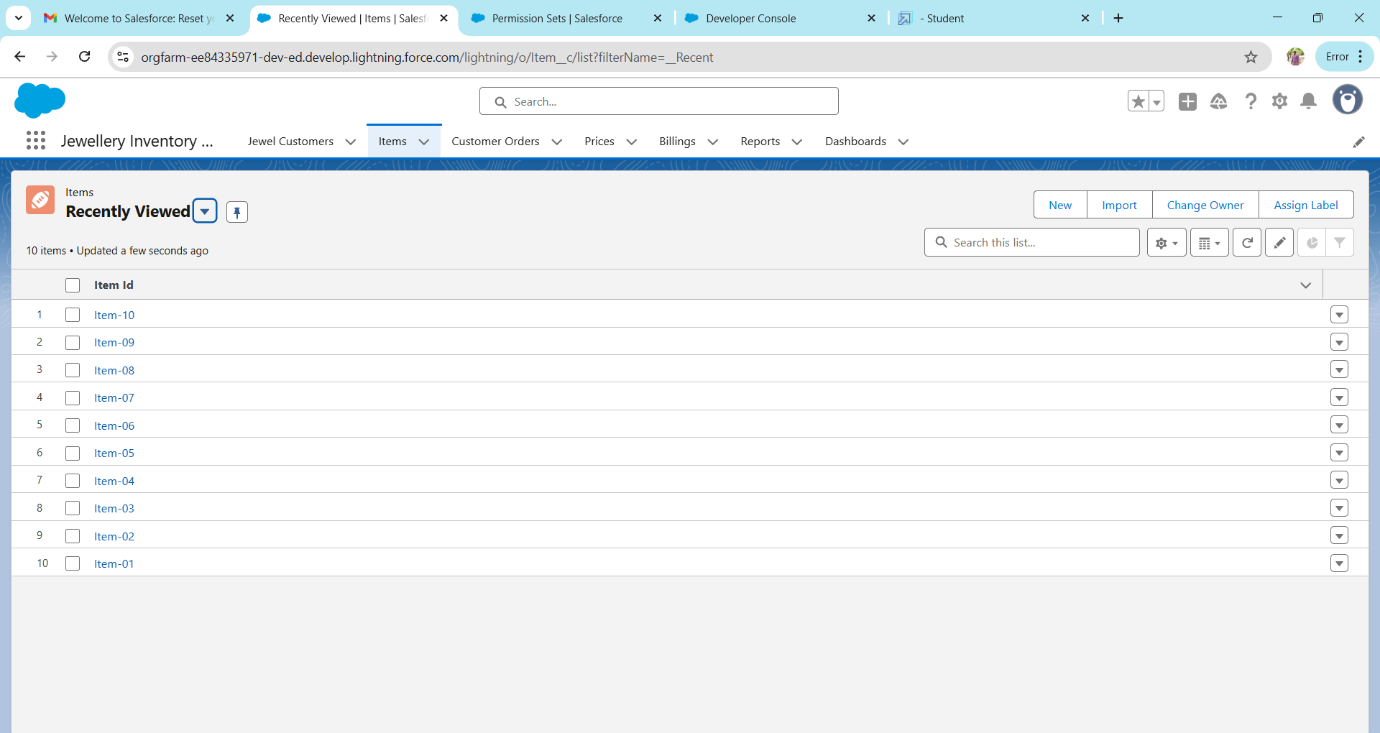
View a jewel customer record



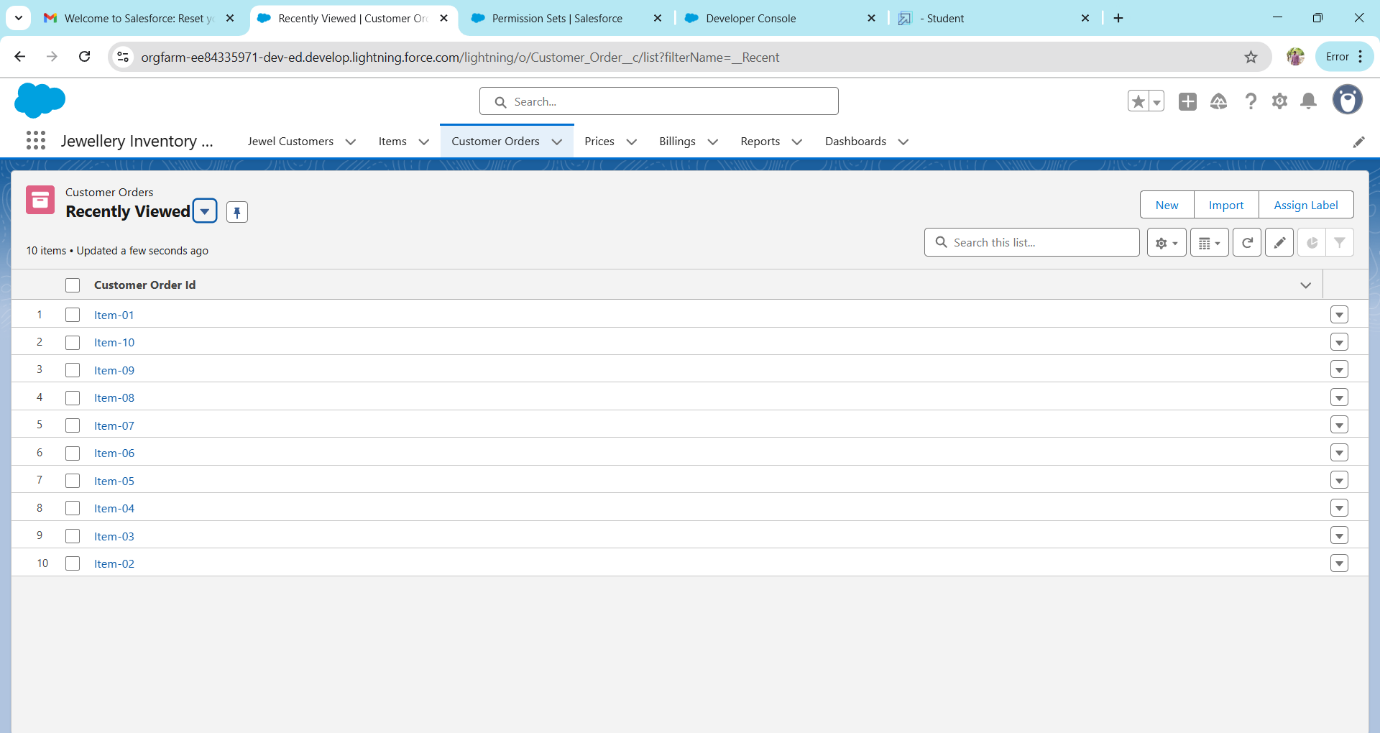
Create a record in Price object



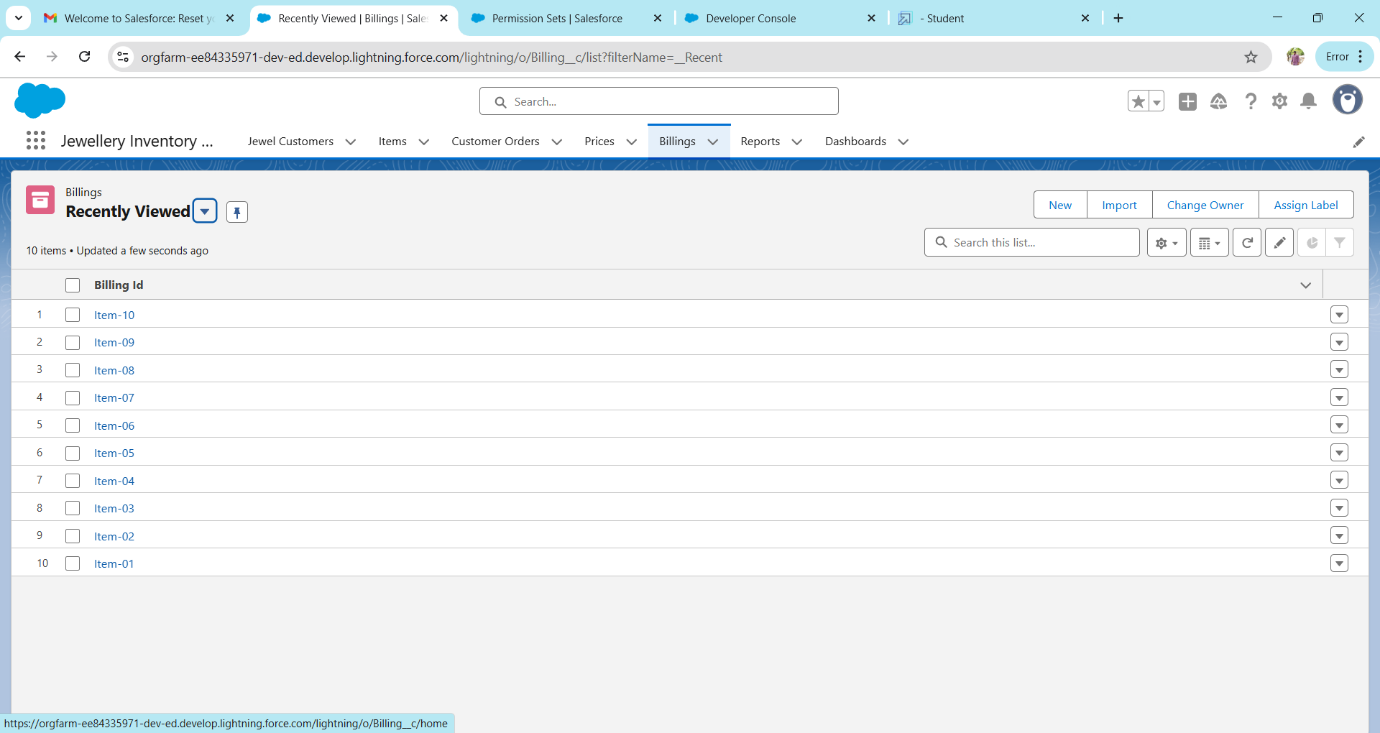
Create a record in Item object



Create a record in customer order

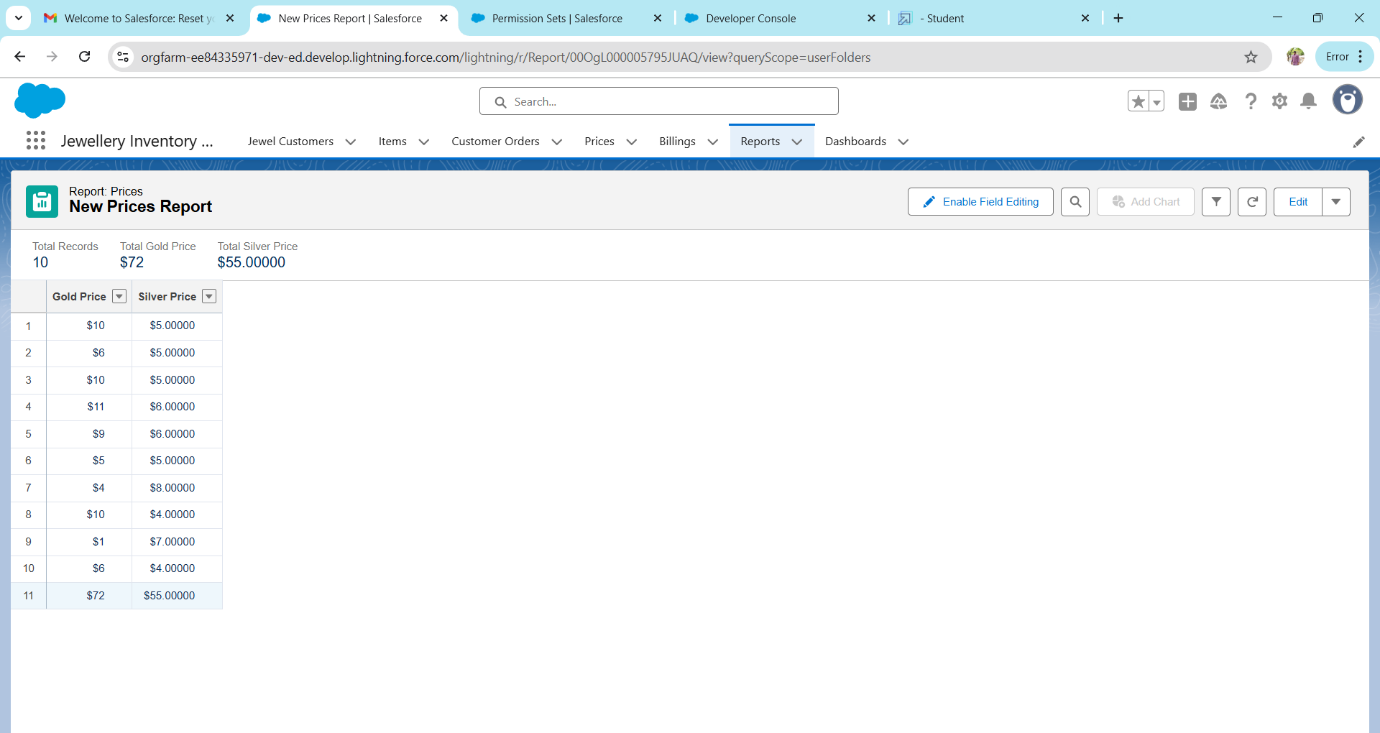


Create a record in Billing object

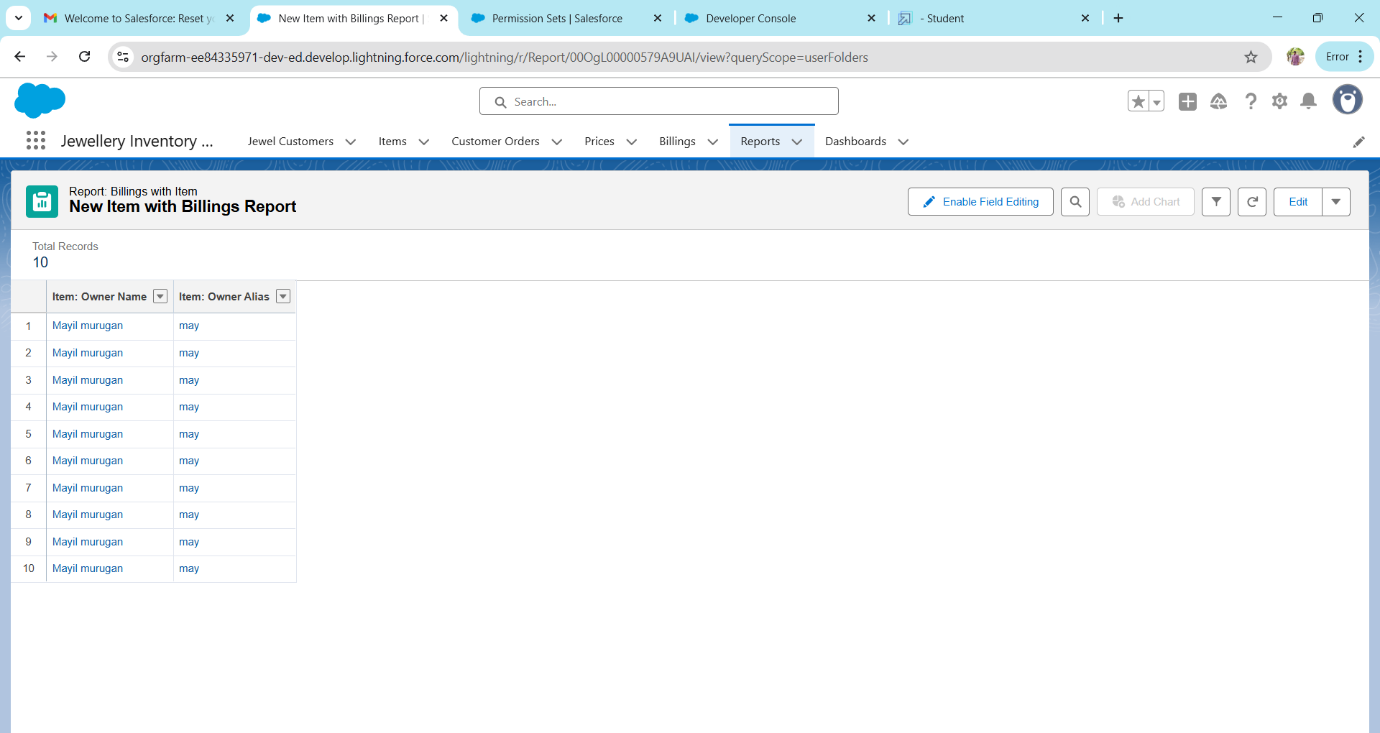


**Reports**

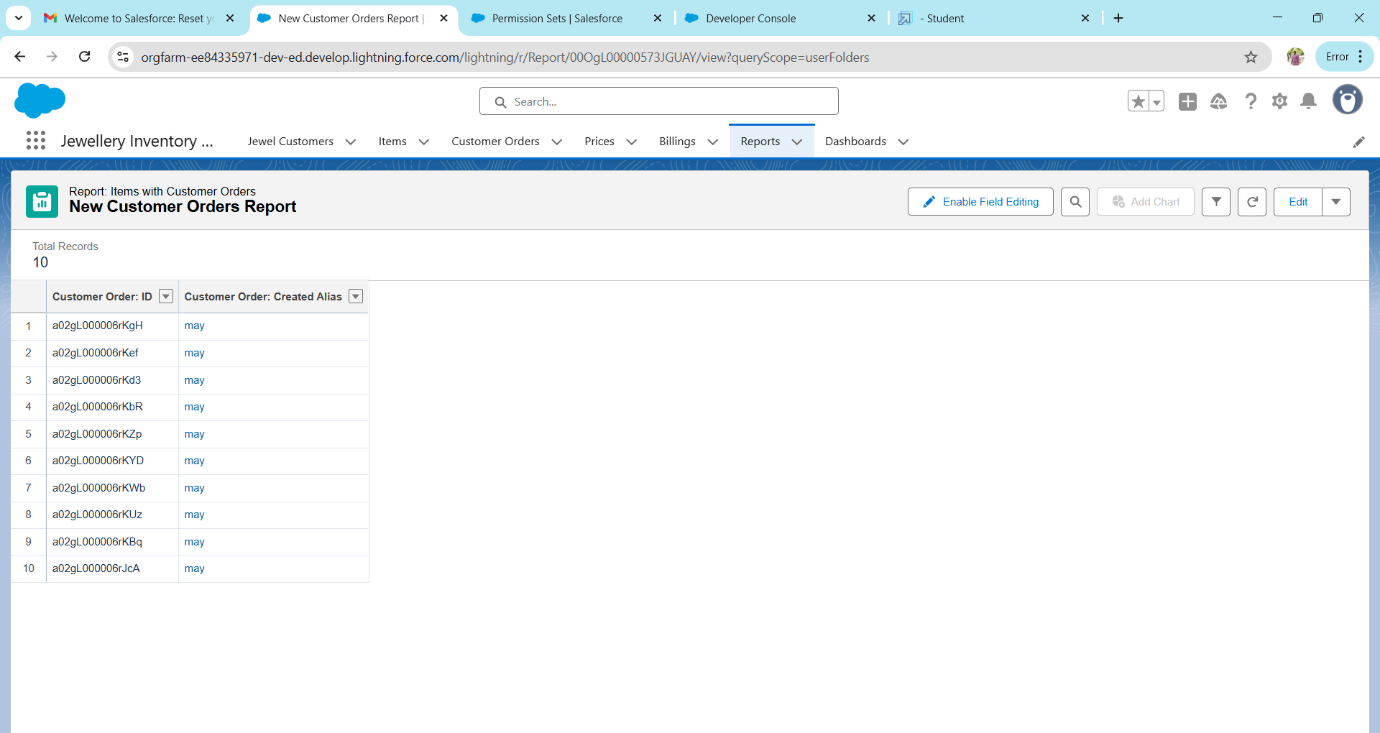
Creating a record



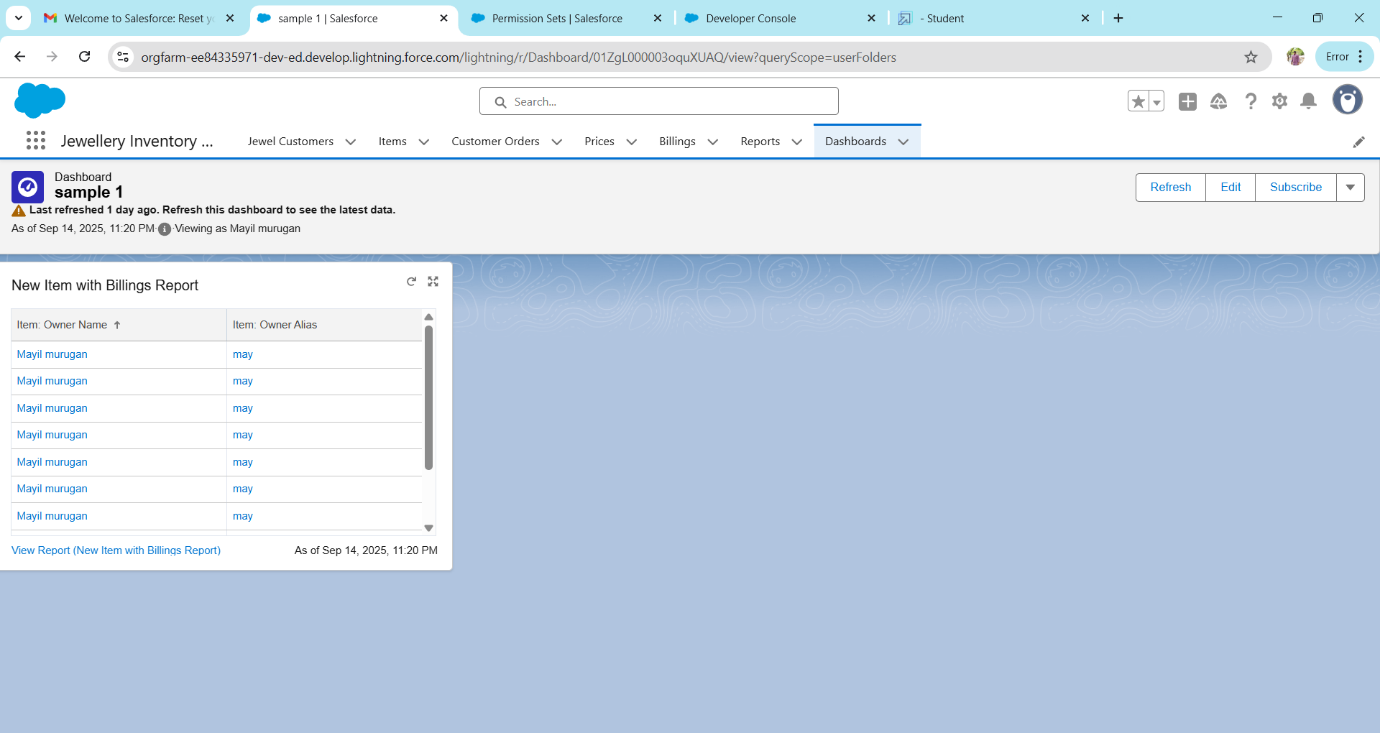
Creating a record type Item with Billings

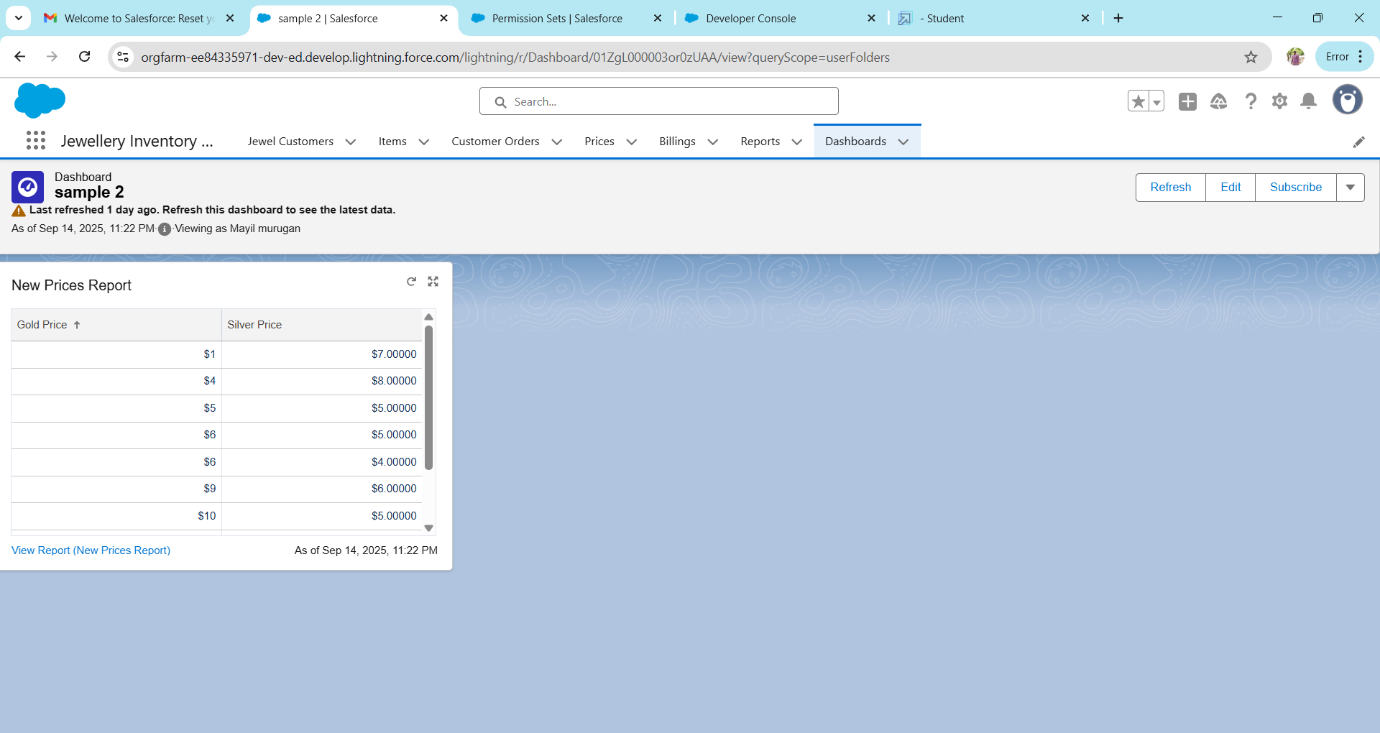


Creating a record with record type Billings with item and customer order

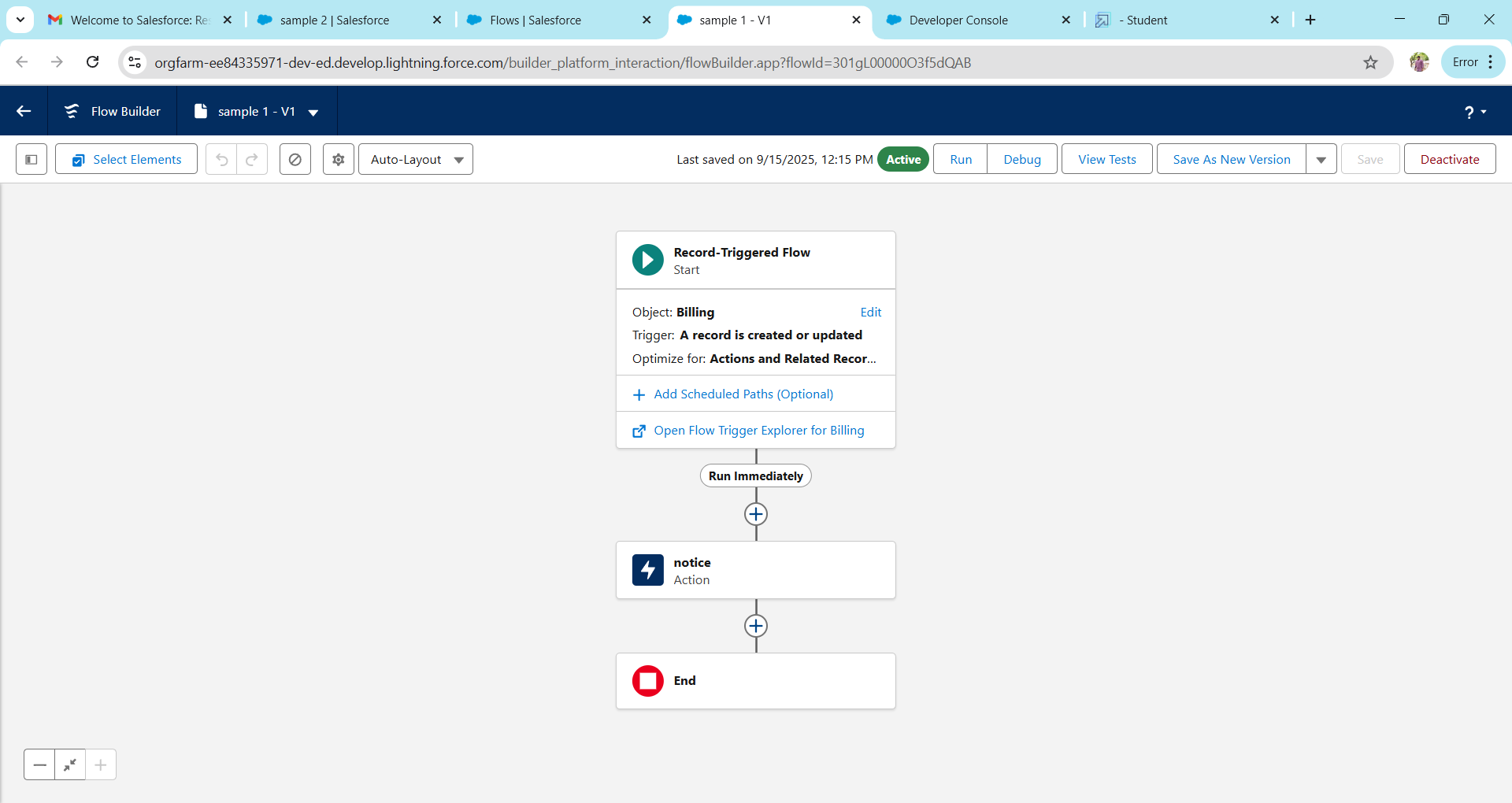


Dashboards



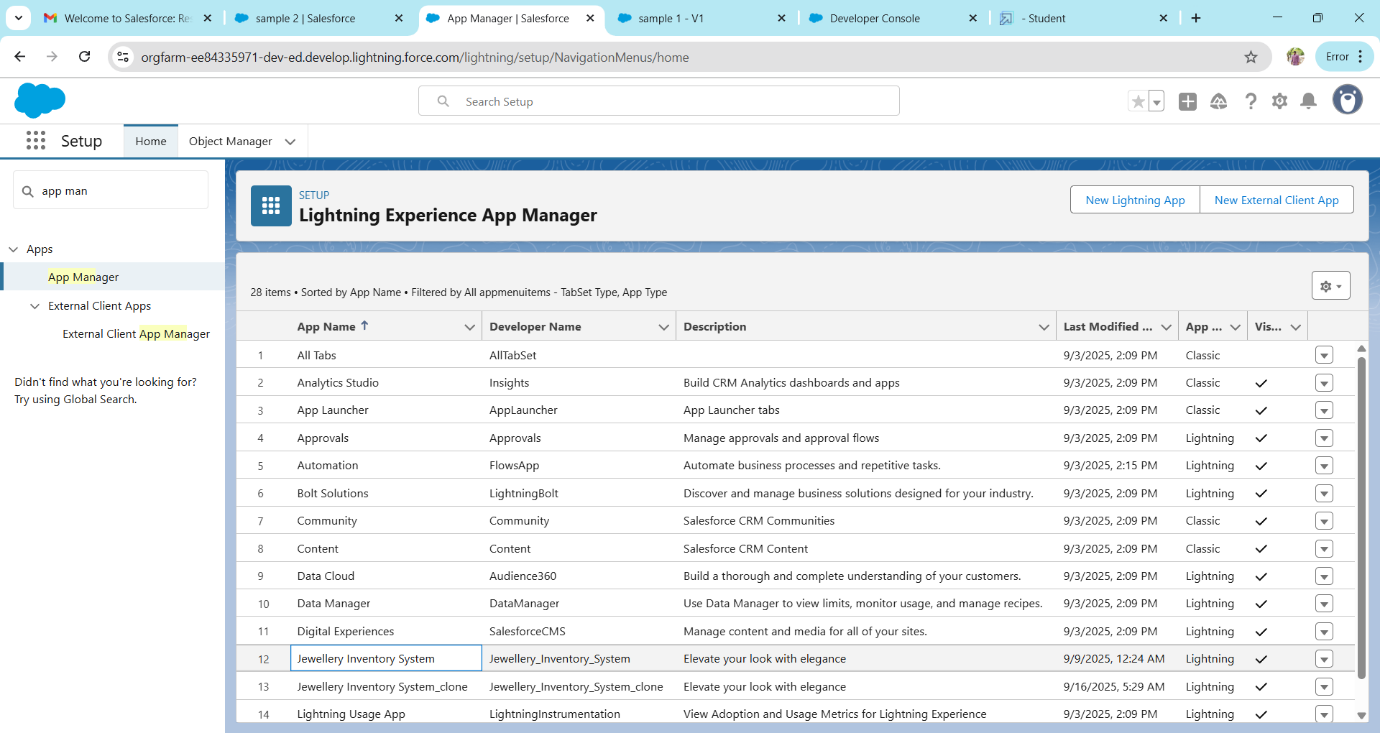


Flows creating

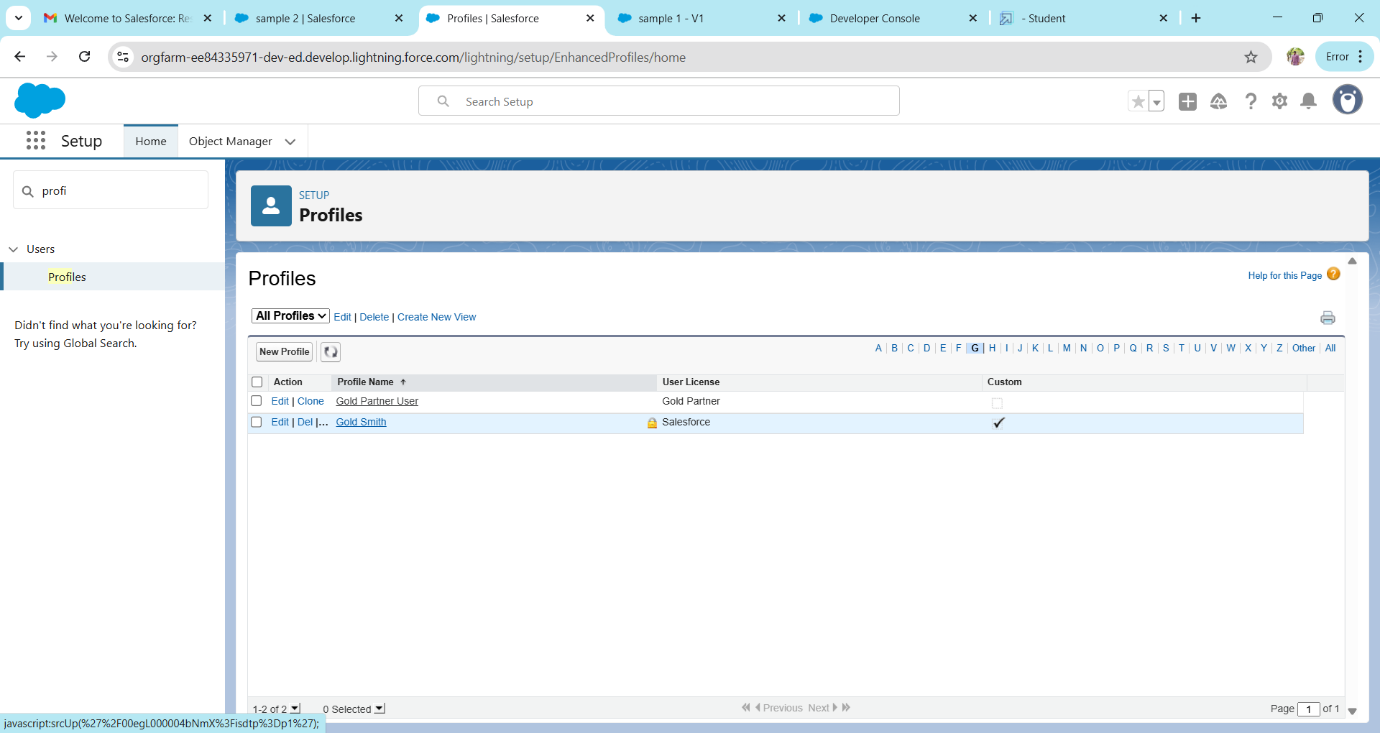


**RESULTS AND OUTPUT**

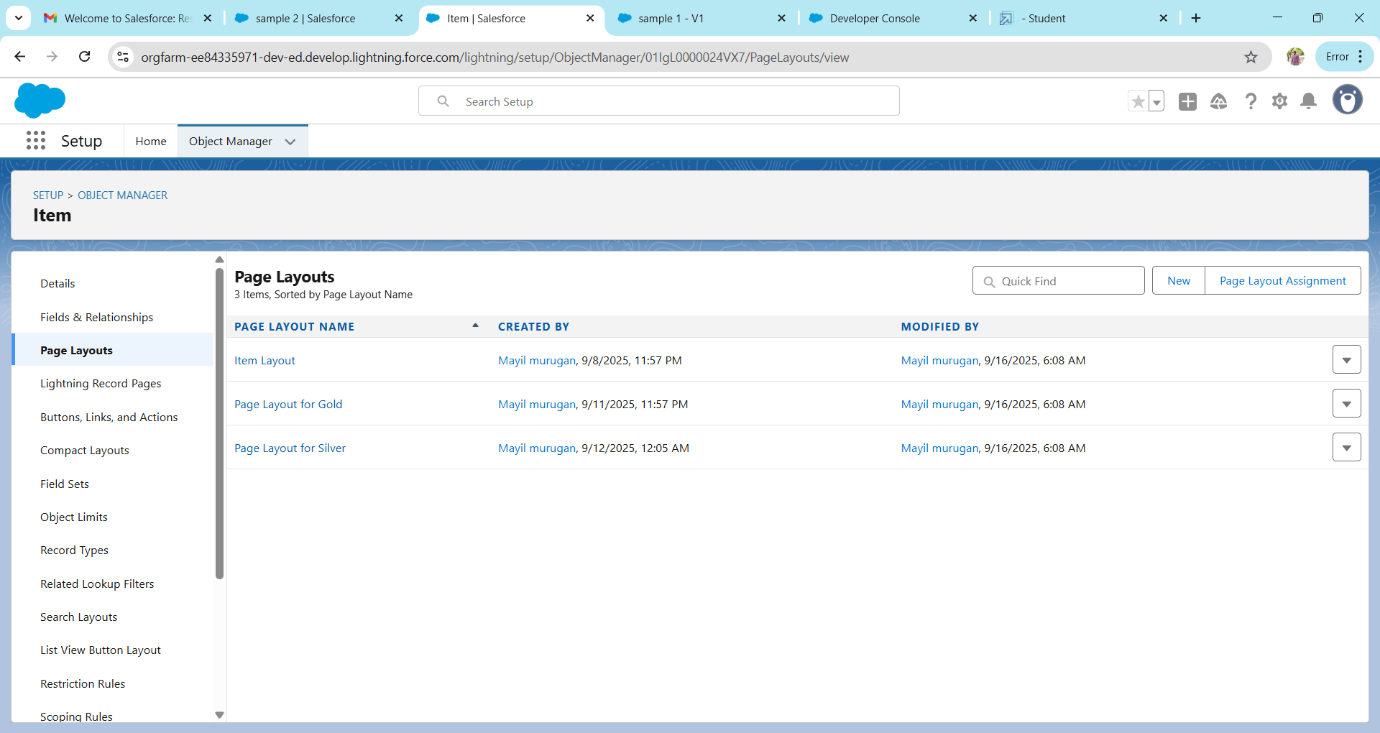
Lightning app

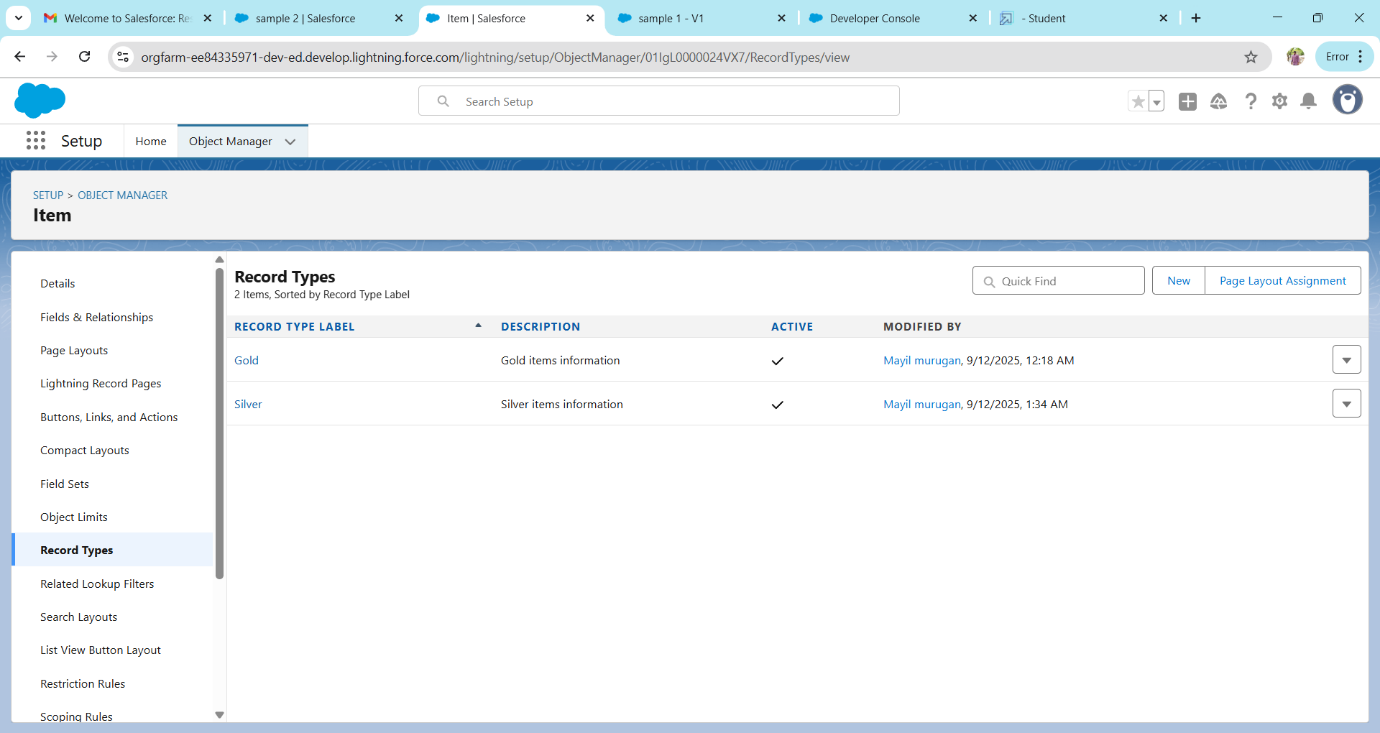


Profiles

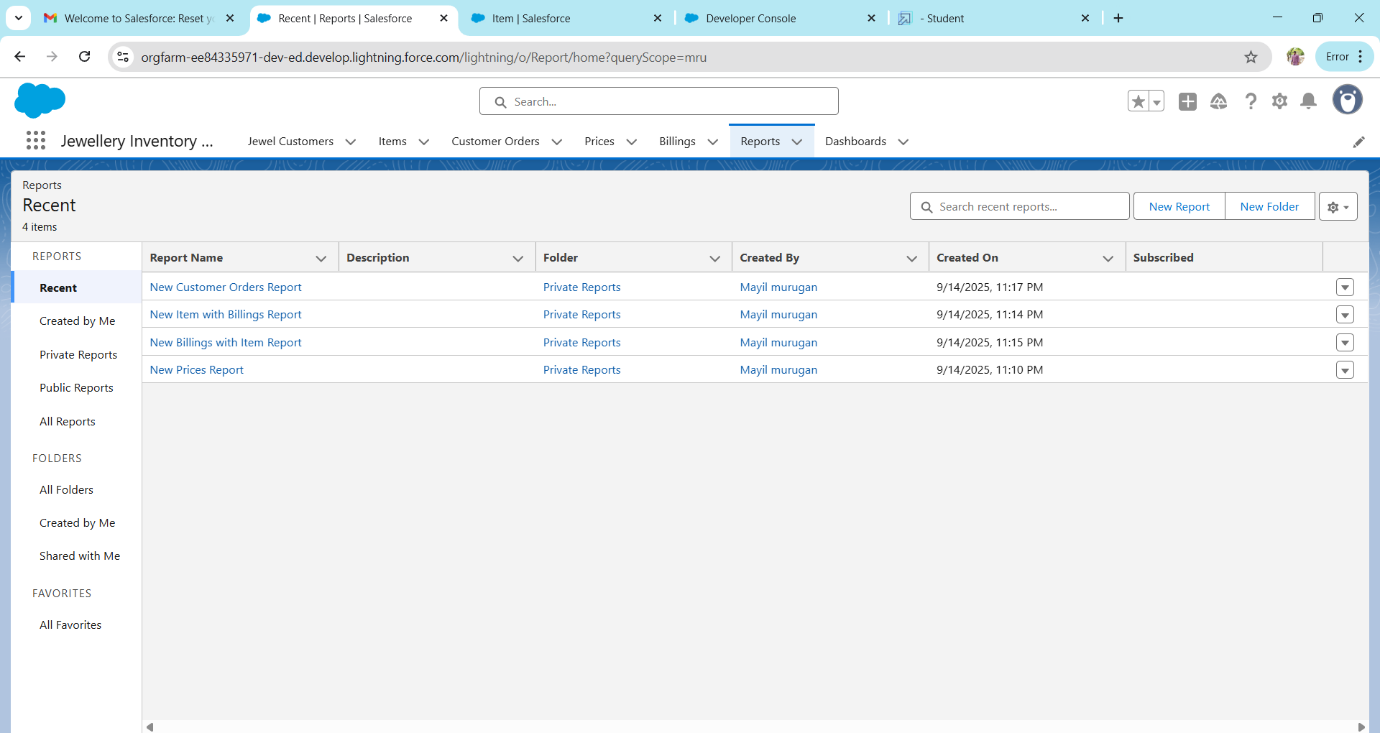


Page Layouts



Record Types

Records



**ADVANTAGES:**

Enhanced customer relationships.

Streamlined sales and Inventory management.

Better data management and insights.

Automation of routine tasks.

Improved communication and collaboration.

**DISADVANTAGES:**

Implementation and training costs.

Data security and privacy concerns.

User adoption challenges.

Customization and Complexity.

Ongoing maintenance and updates.

**CONCLUSION:**

A CRM application for jewel management aims to streamline customer relationship, sales, inventory management, and communication in the jewellery business. In conclusion, a well-configured CRM application leveraging salesforce capabilities can be a valuable tool for jewellers to manage customer relationships, optimize sales and inventory processes, and drive business growth.

**APPENDIX**

**Source Code:** Provided I Apex Classes and Triggers

**UpdatePaidAmountTriggerHandler.apxc**

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;

        }

    }

**UpdatePaidAmountTrigger.apxt:**

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);

    }

}