**CRM APPLICATION FOR JEWEL MANAGEMENT (DEVELOPER)**

**COLLEGE NAME : Tiruppur Kumaran College For Women**

**COLLEGE CODE : bru3j**

**TEAM ID : NM2025TMID28335**

**TEAM MEMBERS :**

* **TEAM LEADER : Dhanusree. A**

**EMAIL ID :** [**vanithaashok0303@gmail.com**](mailto:vanithaashok0303@gmail.com)

* **TEAM MEMBER : Dharshana R**

**EMAIL ID :** [**dharshanaram14@gmail.com**](mailto:dharshanaram14@gmail.com)

* **TEAM MEMBER : Harini K**

**EMAIL ID :** [**harinisuseela03@gmail.com**](mailto:harinisuseela03@gmail.com)

* **TEAM MEMBER : Guruchithra M**

**EMAIL ID :** [**kmarikmari918@gmail.com**](mailto:kmarikmari918@gmail.com)

**1.INTRODUCTION:**

**1.1 PROJECT OVERVIEW**

This CRM application is a software solution designed to streamline the processes of a jewel business. It manages customer relationships, sales, and inventory to improve efficiency and enhance the customer experience.

**1.2 PURPOSE**

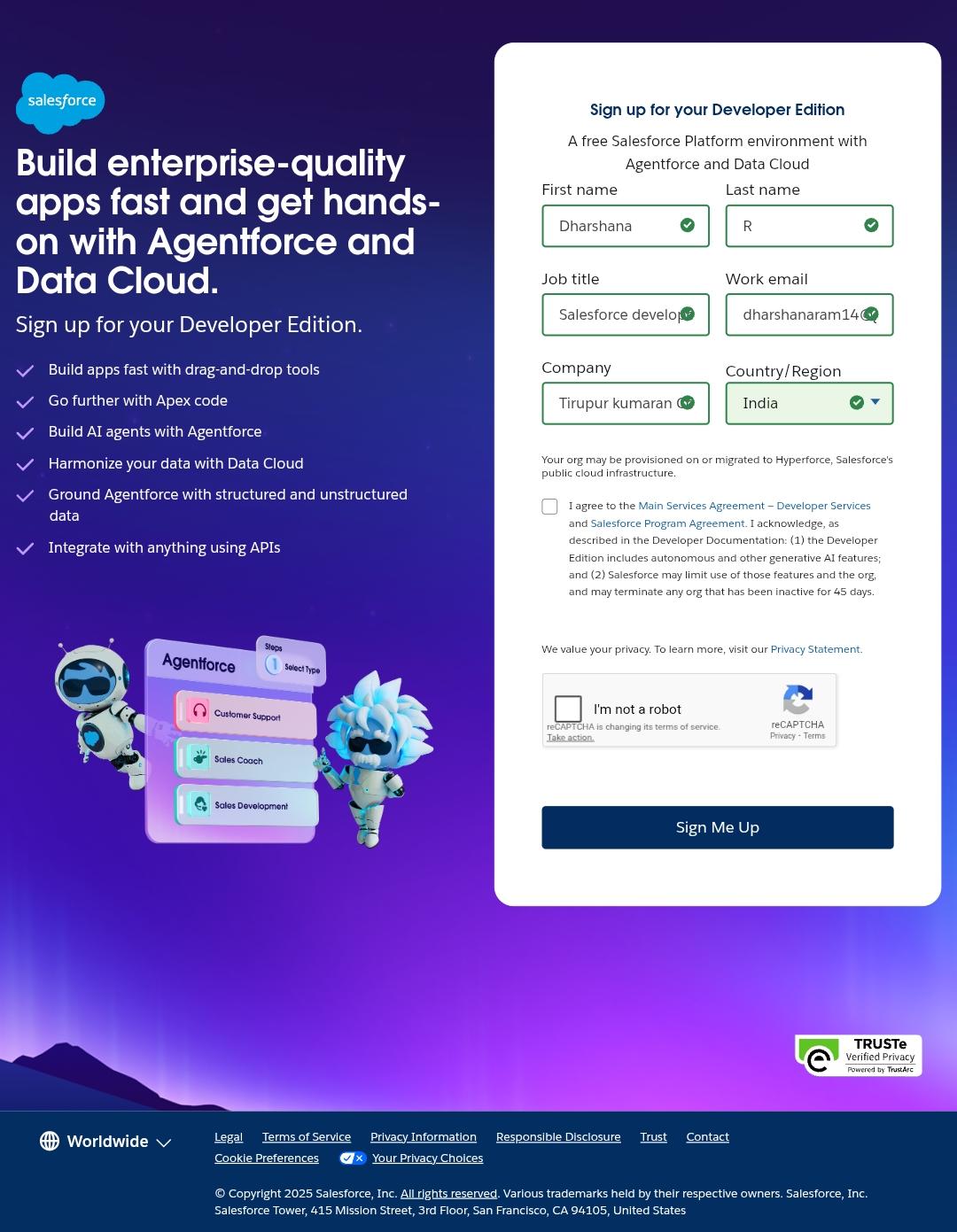
The main objective of the application is to:

* Increase customer satisfaction and loyalty.
* Provide data-driven insights for better decision-making.
* Reduce manual effort and human errors in daily operations.

DEVELOPMENT PHASE

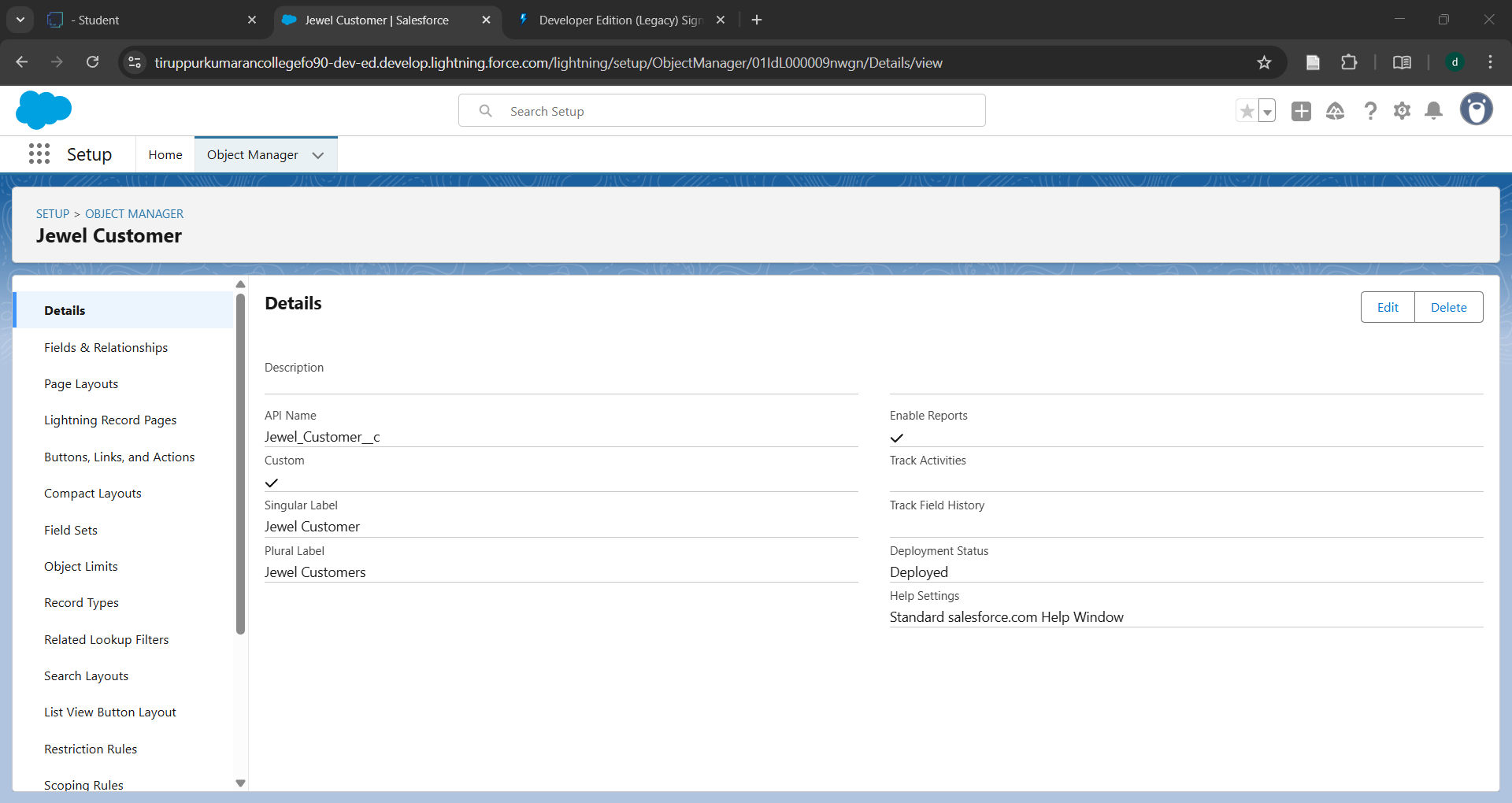
**Creating Development Phase**

By using this URL [**https://developer.salesforce.com/developer-legacy/signup**](https://developer.salesforce.com/developer-legacy/signup)

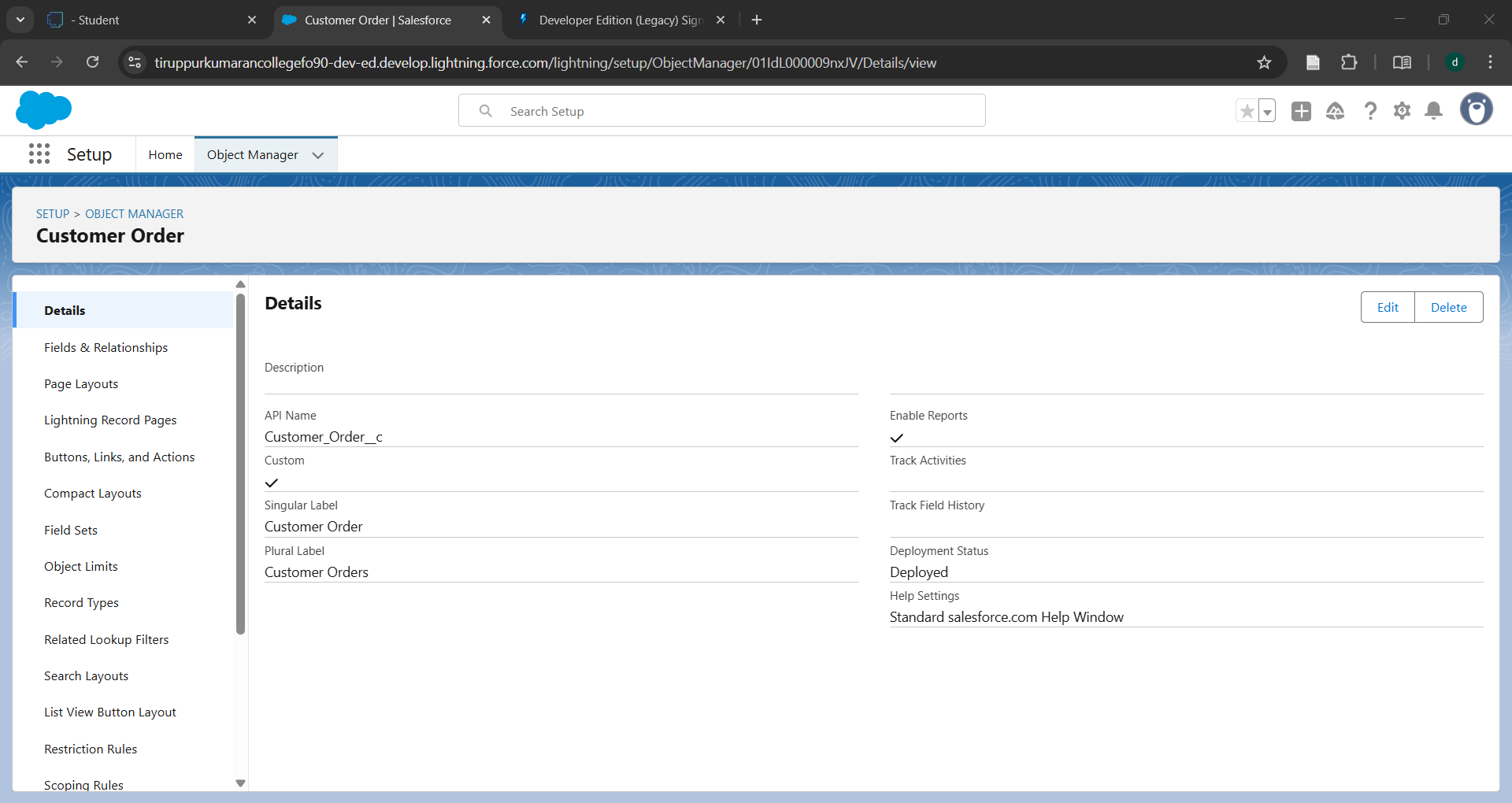
****

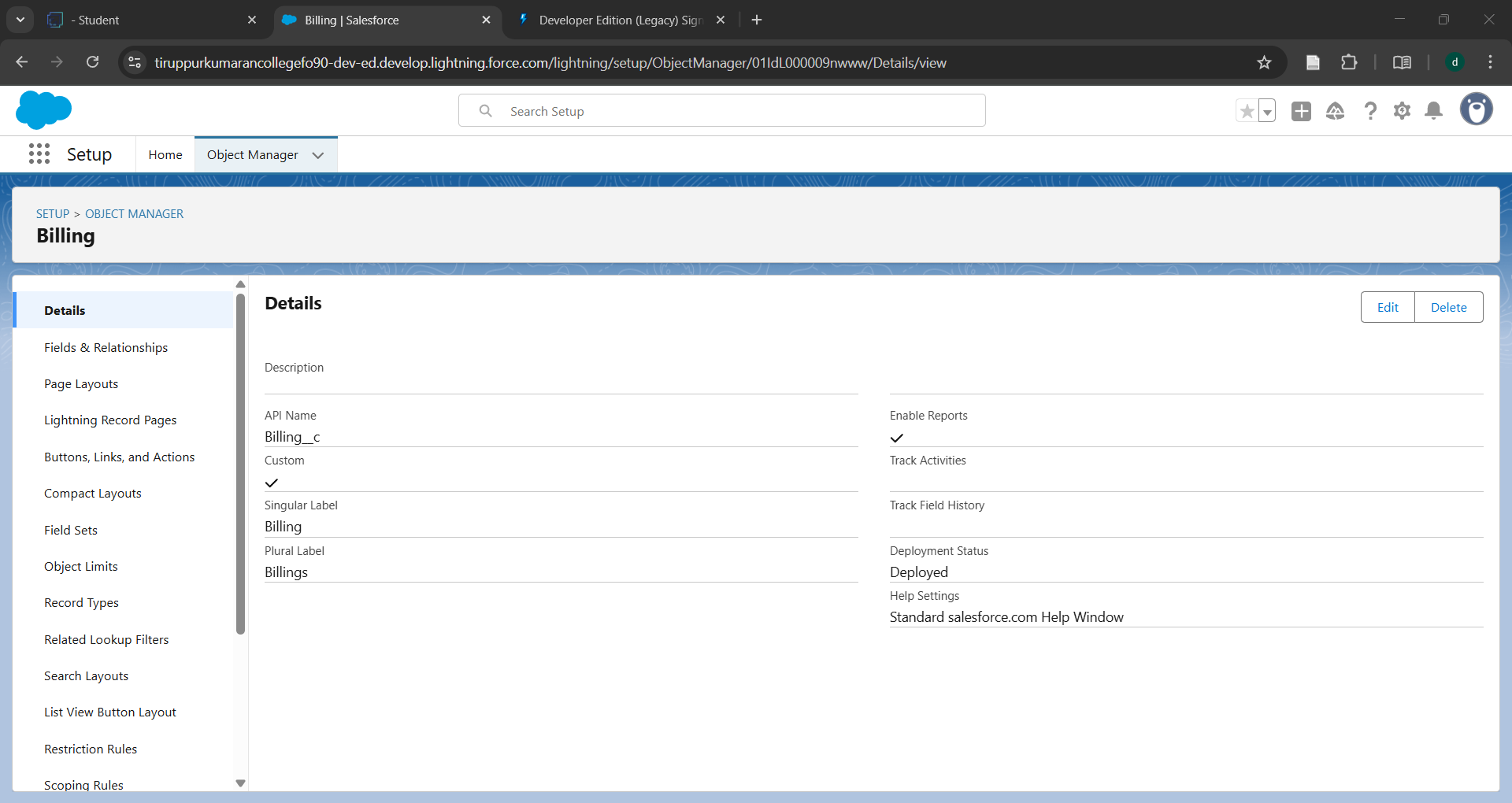
**OBJECTS**

* **Created Objects : Jewel Customer, Item, Customer, Order, Price and Billing**

****

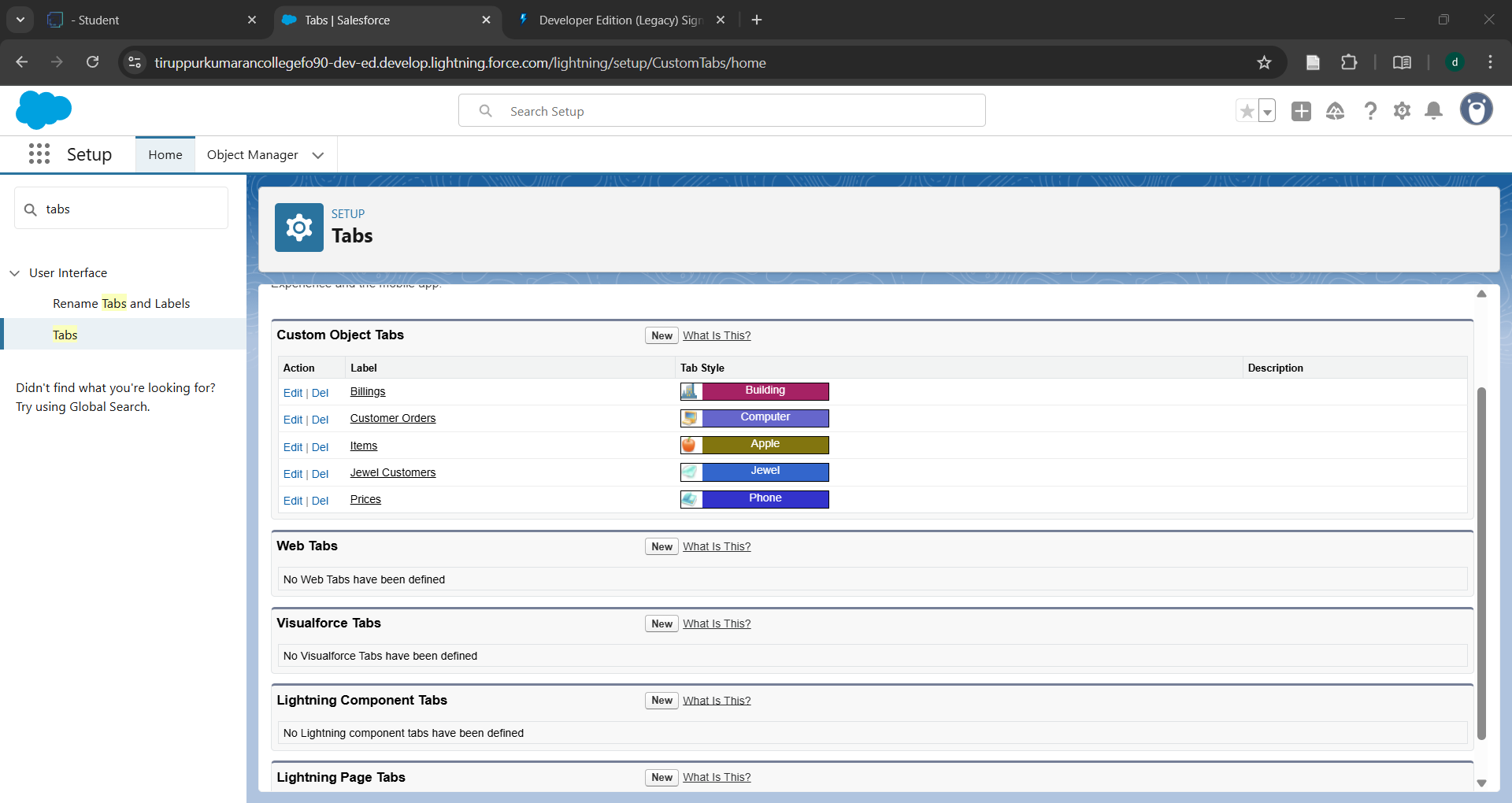
****

****

****

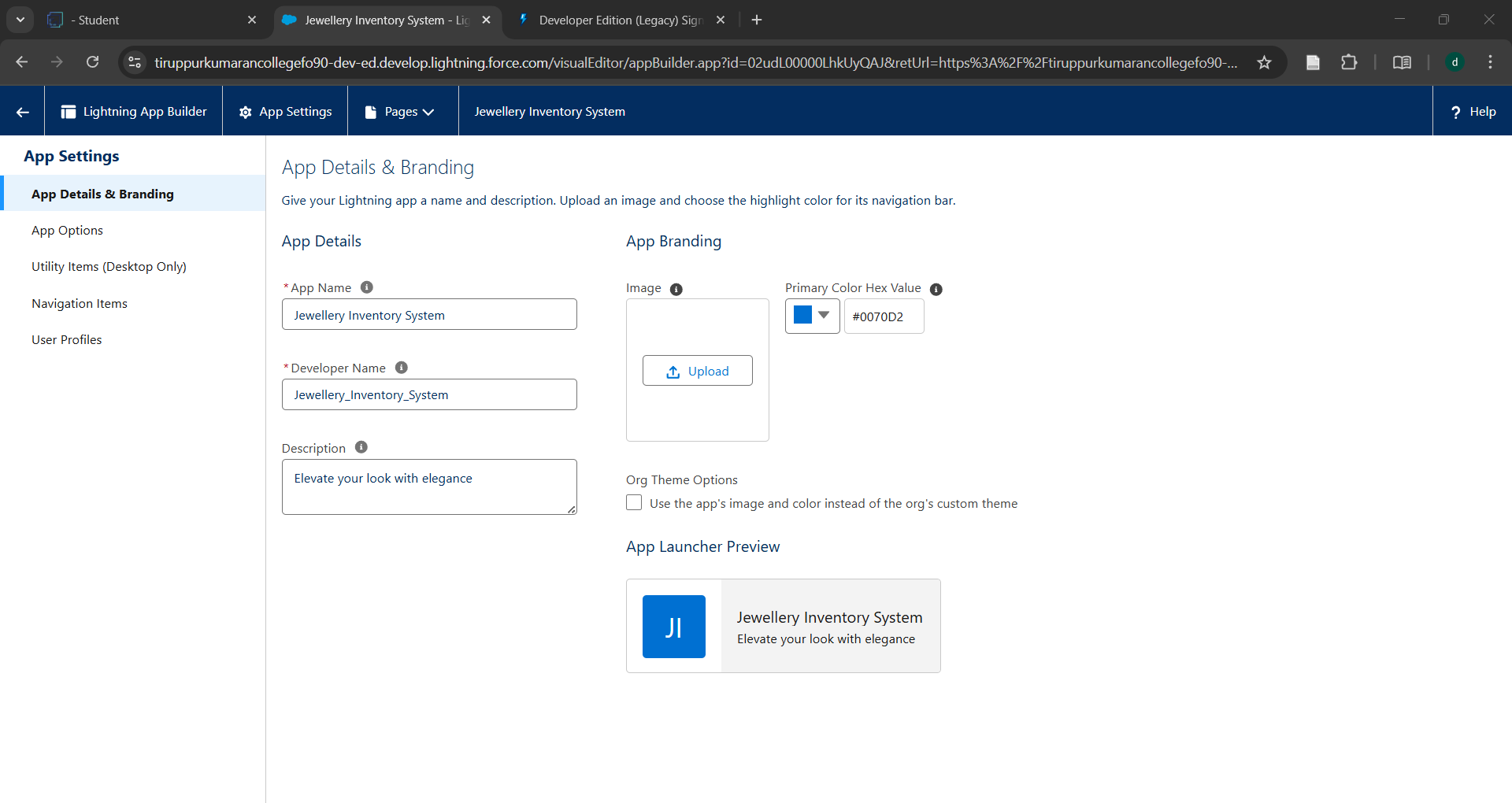
**TABS**

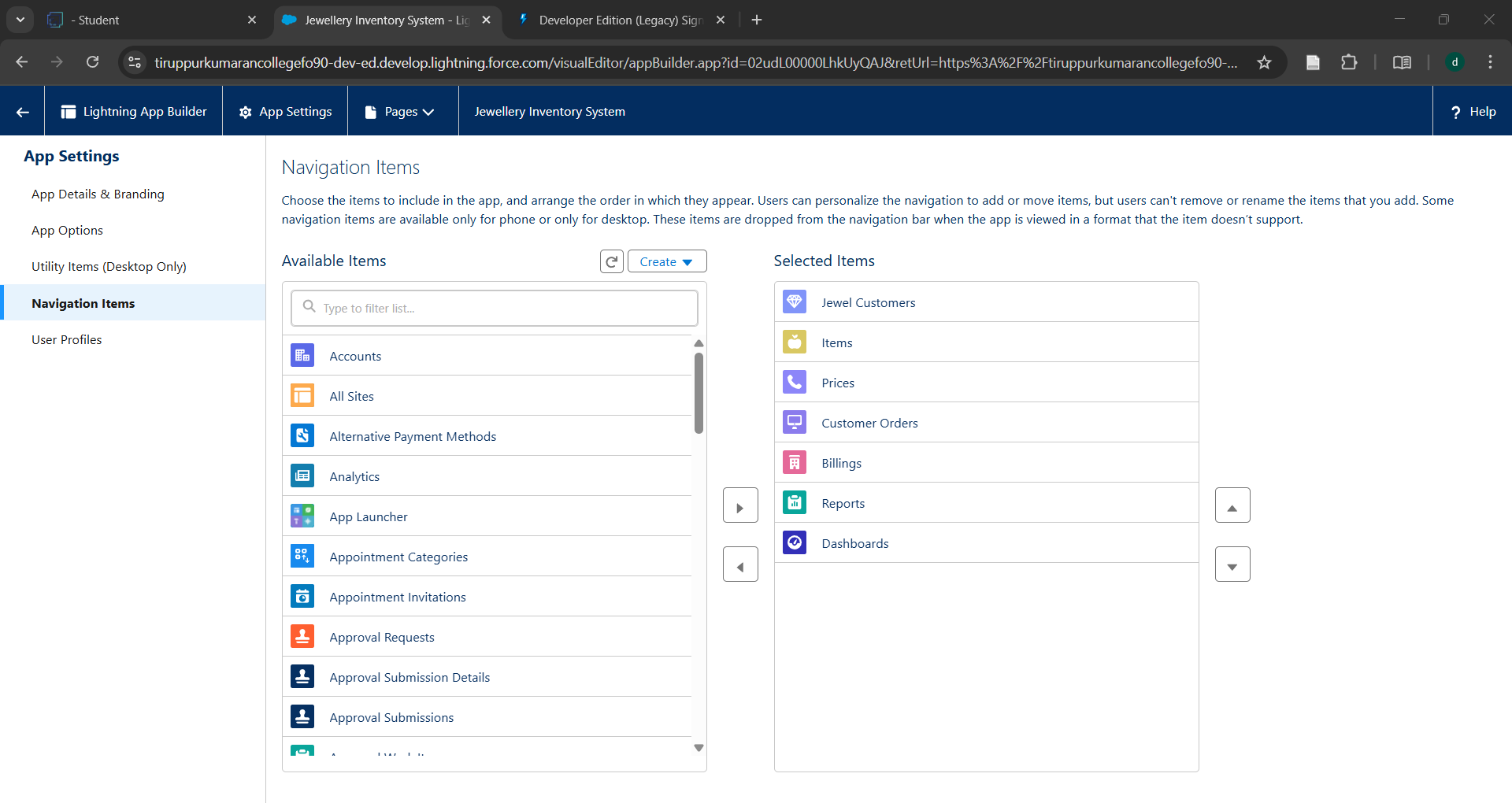
* **Created tabs : Jewel Customer, Item, Customer, Order, Price and Billing**

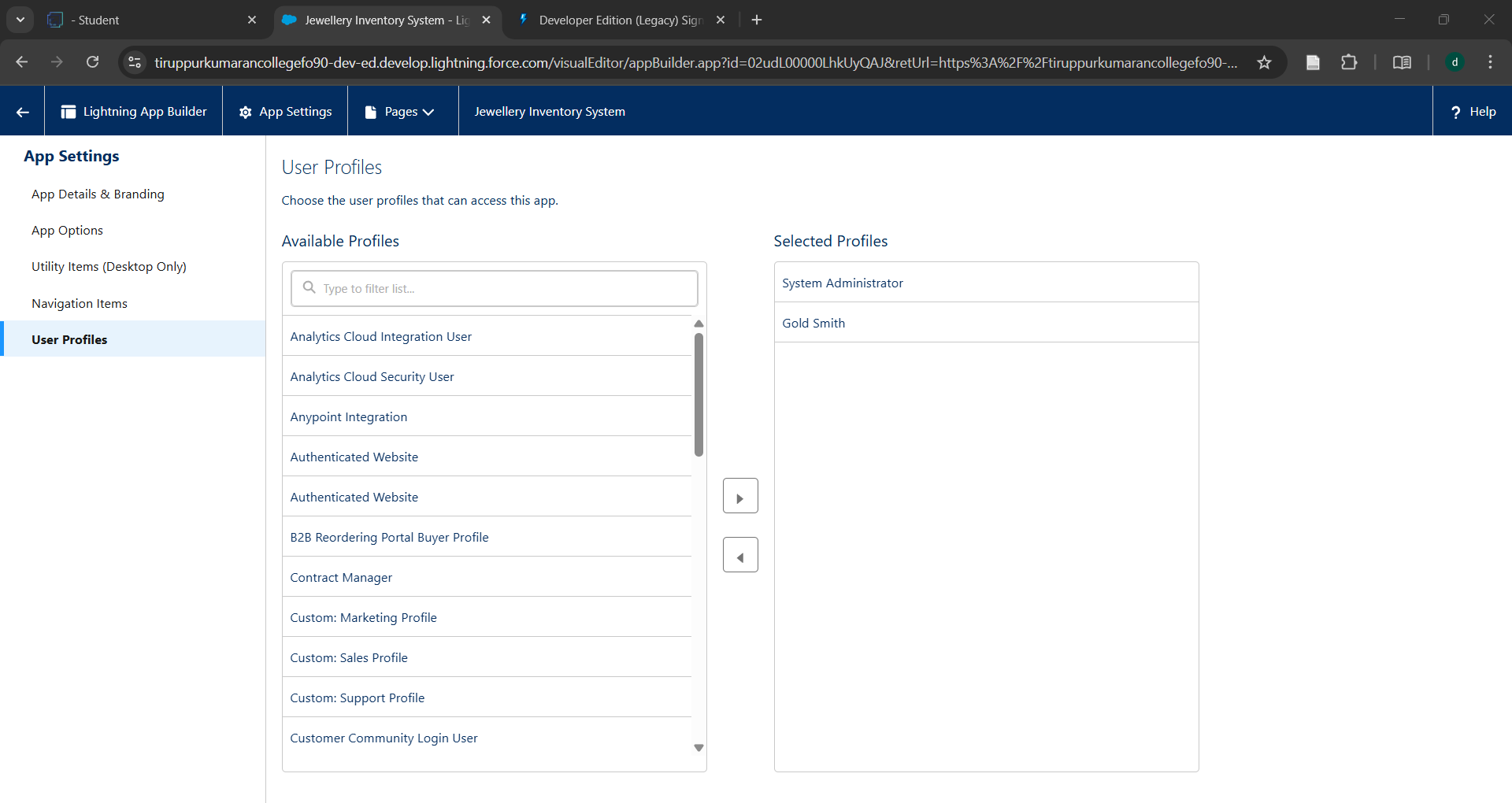


**THE LIGHTNING APPS**

* Developed lightning apps using relevant tabs

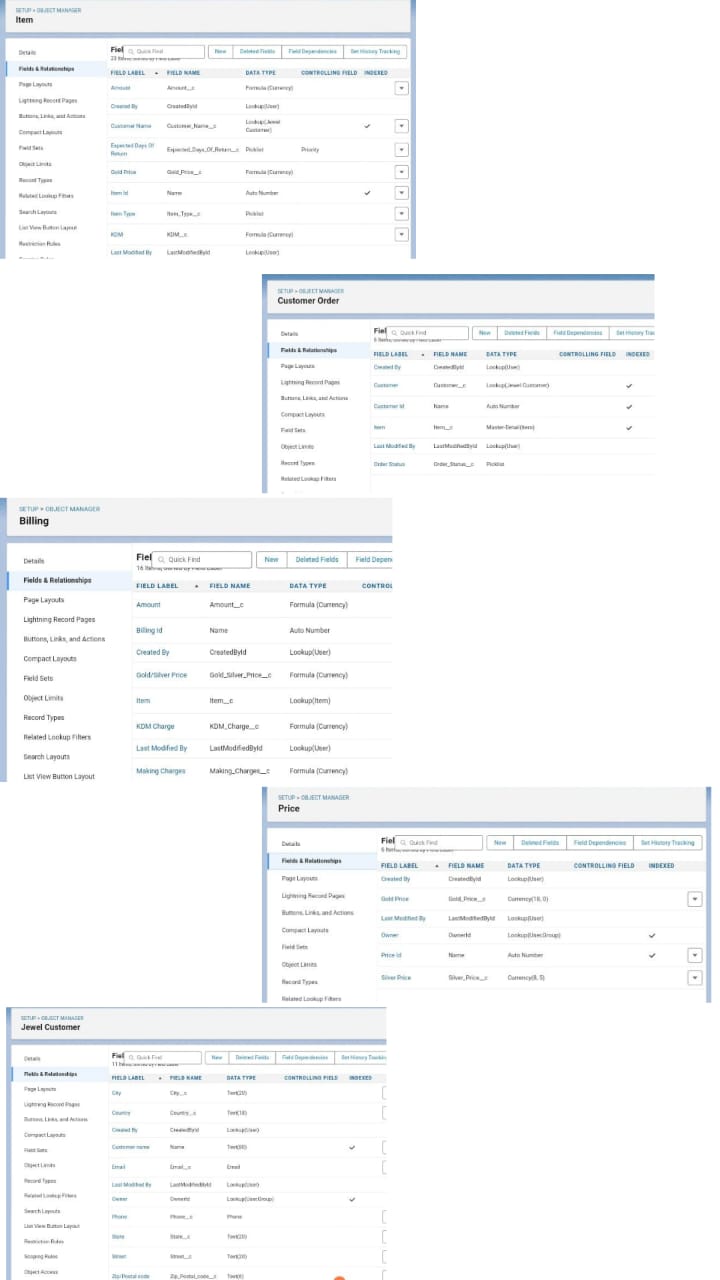






**FIELDS AND RELATIONSHIPS**

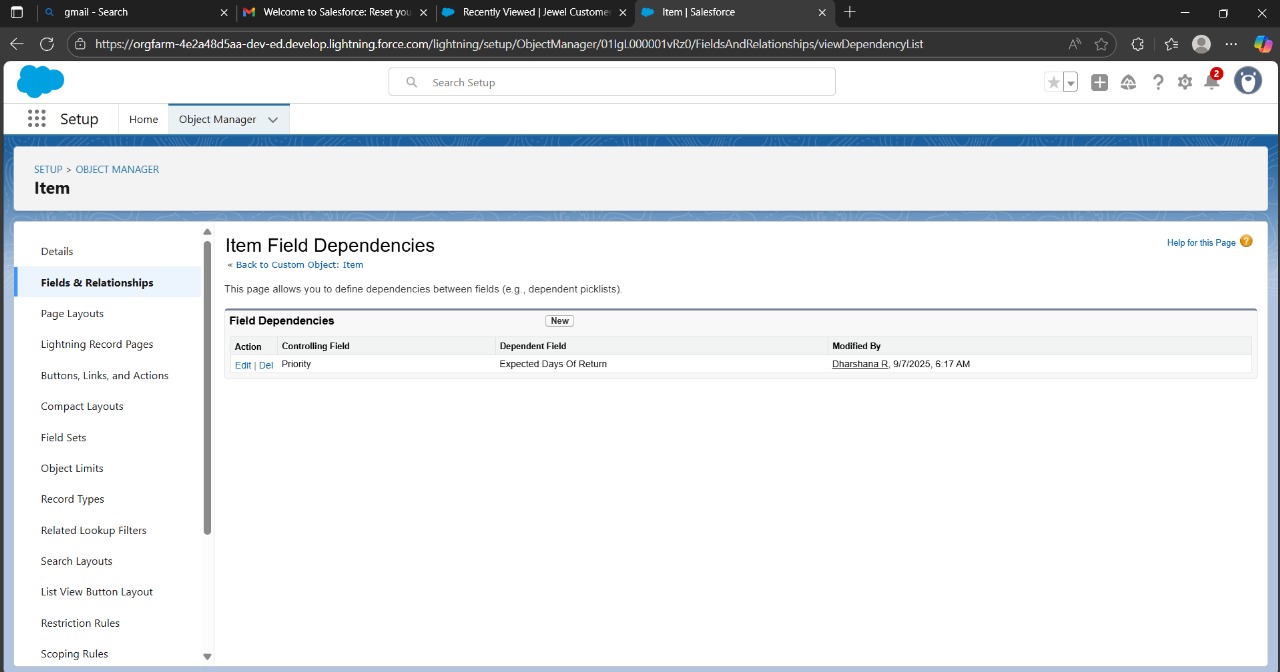
* Configured fields and relationships



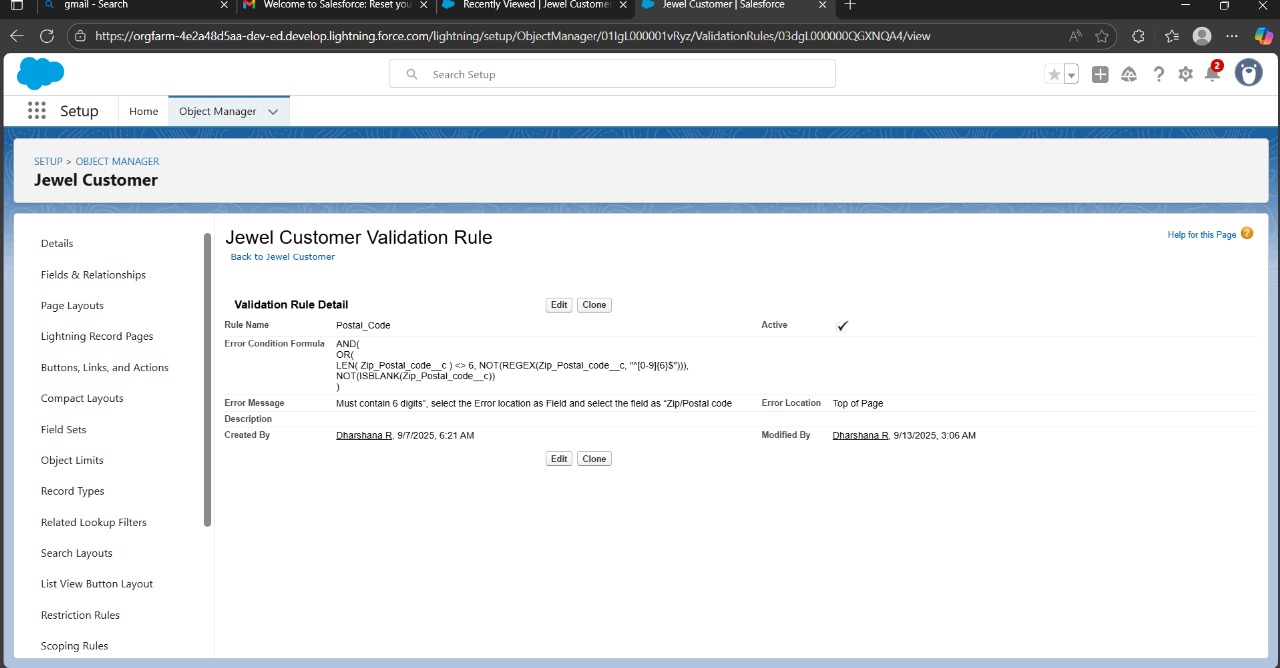
**FIELDS AND DEPENDENCIES**

* Created fields and dependencies for item



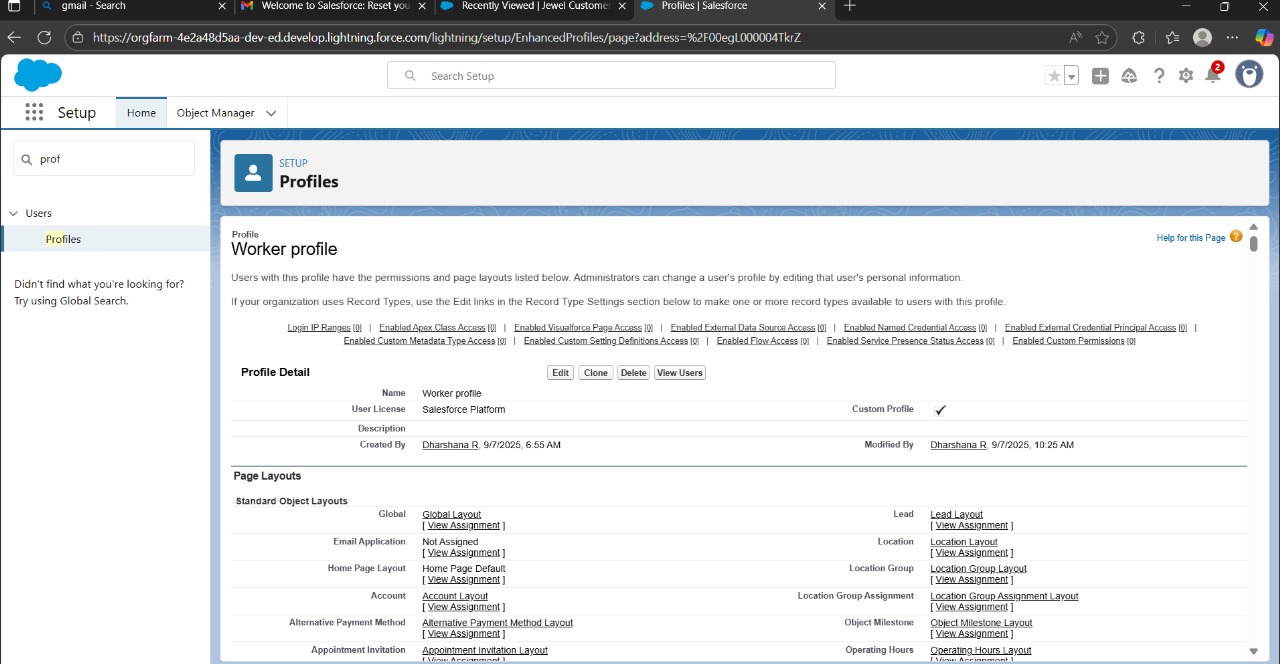


* Created validation rule for jewel Customer



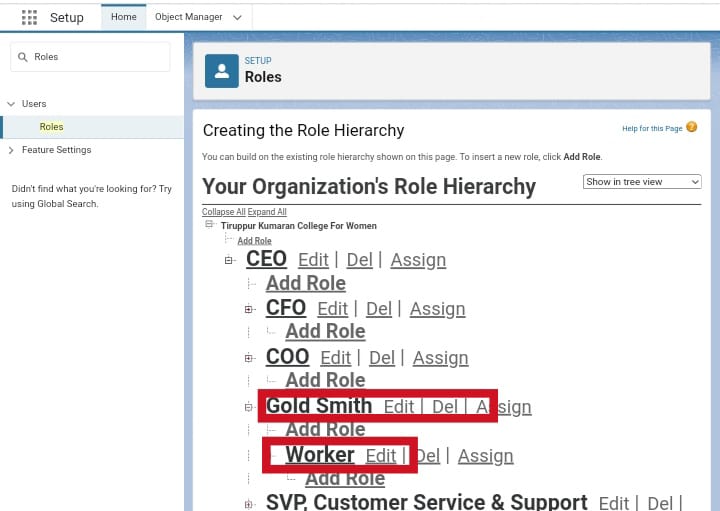
**PROFILES**

* Created profiles : Gold smith and worker profiles

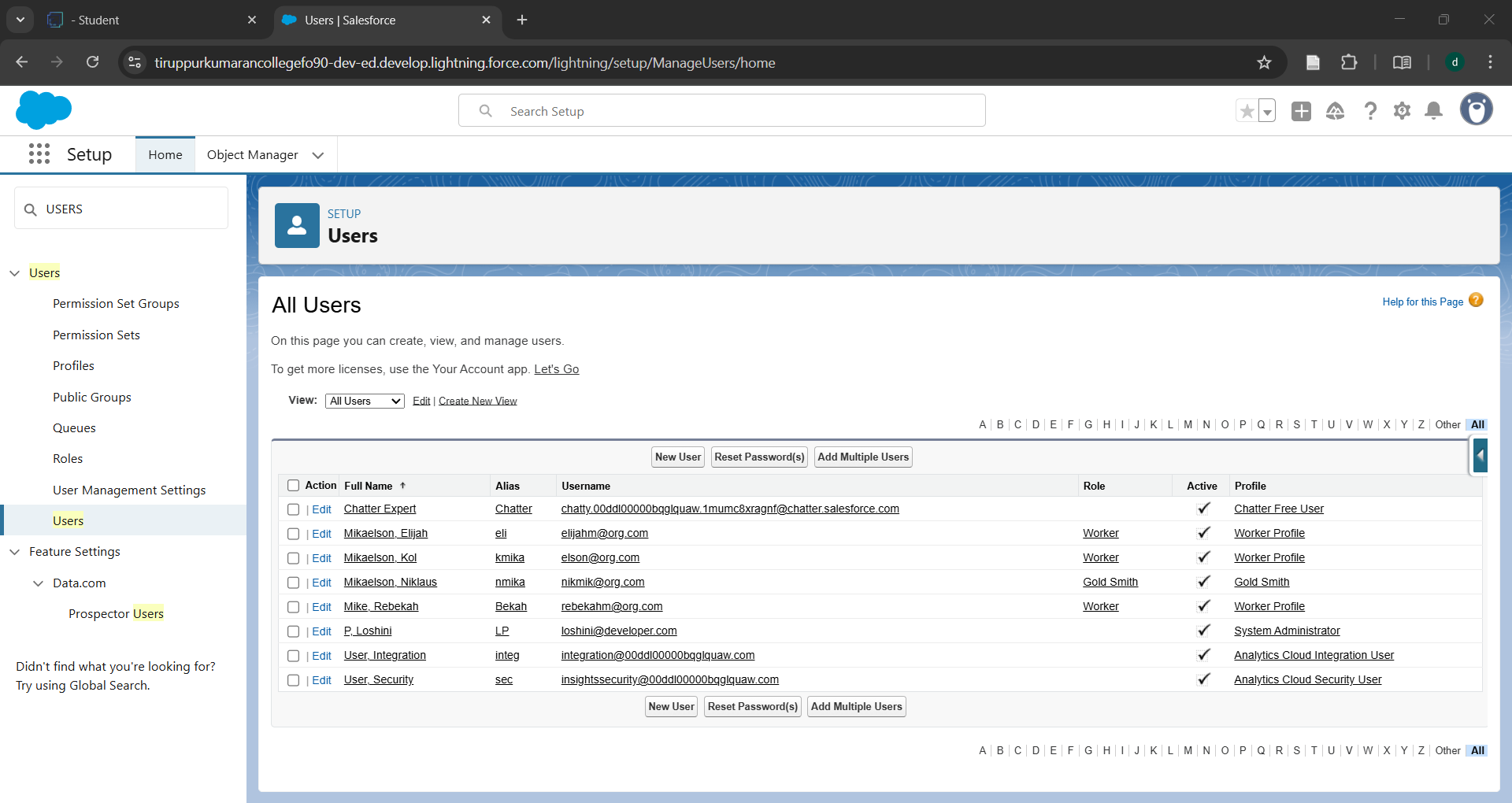


**ROLES**

* Created roles : Gold smith and Worker

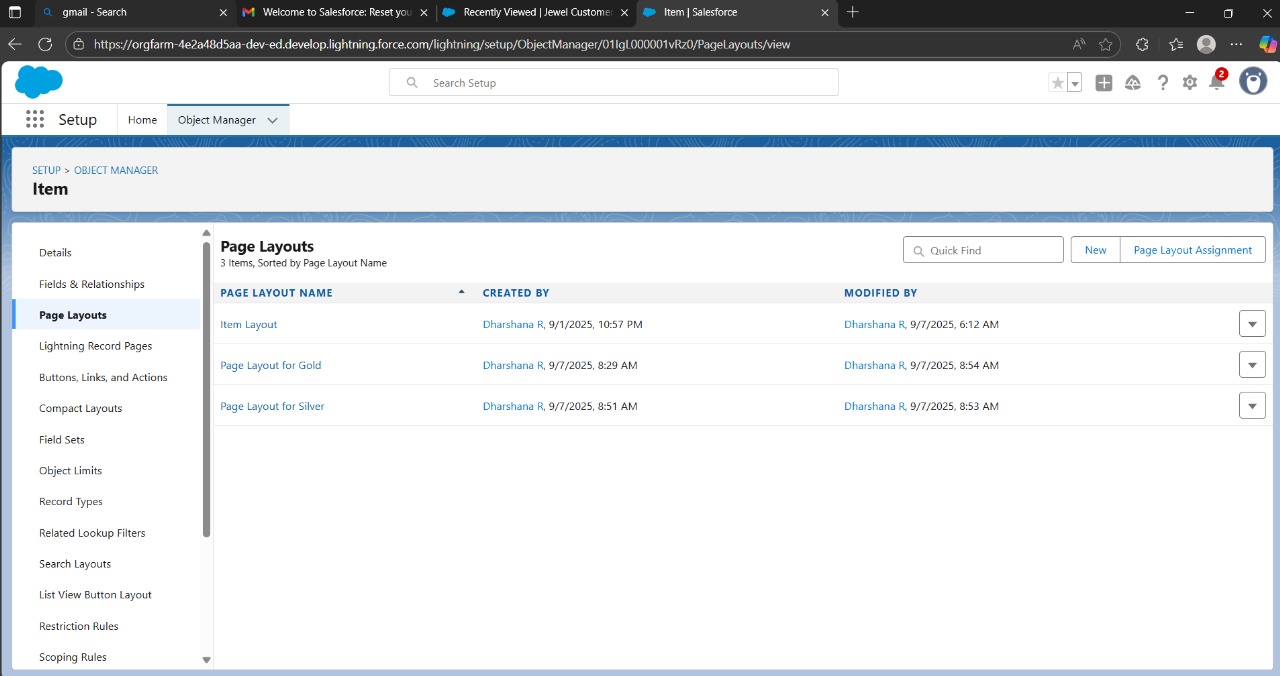
**USERS**

* Created users



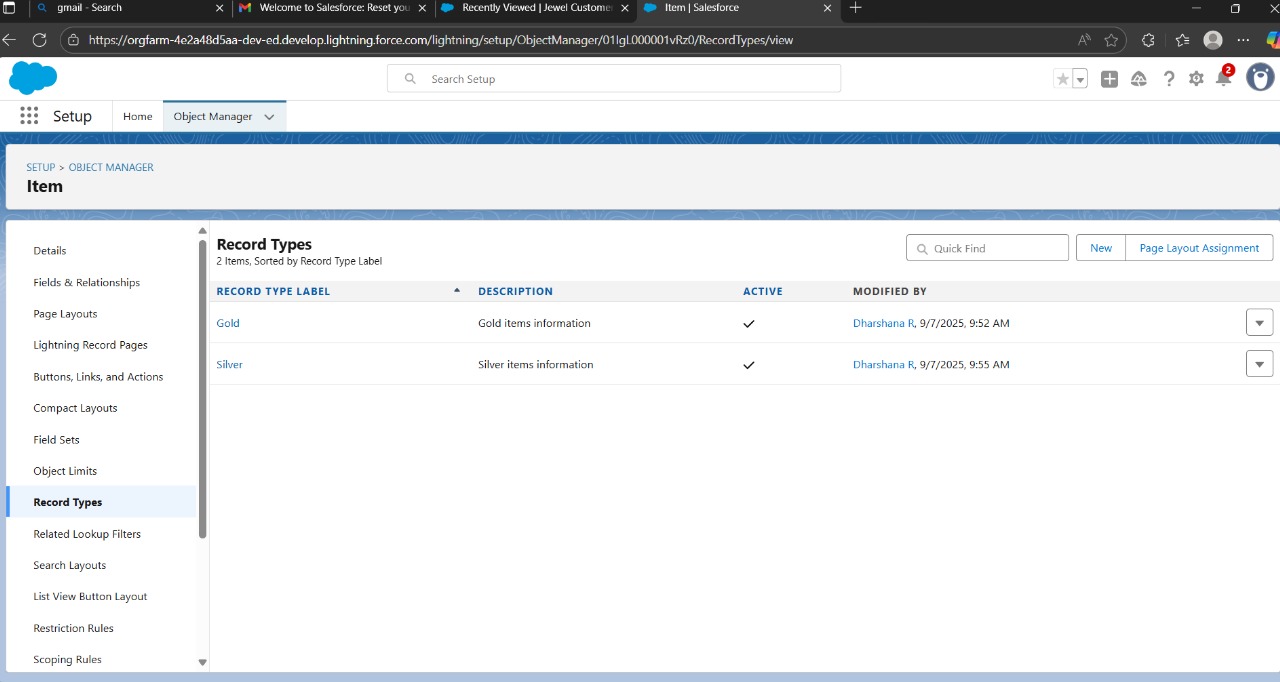
**PAGE LAYOUTS**

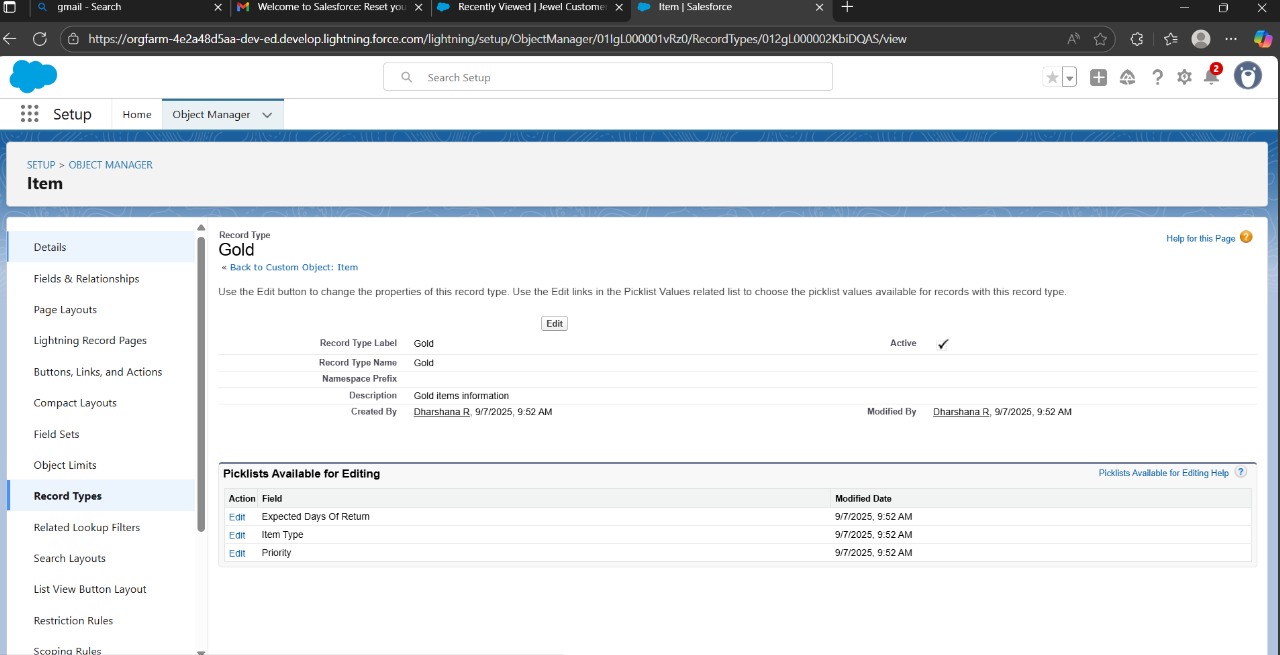
* Created page layouts:Gold page layout and silver page layout for item

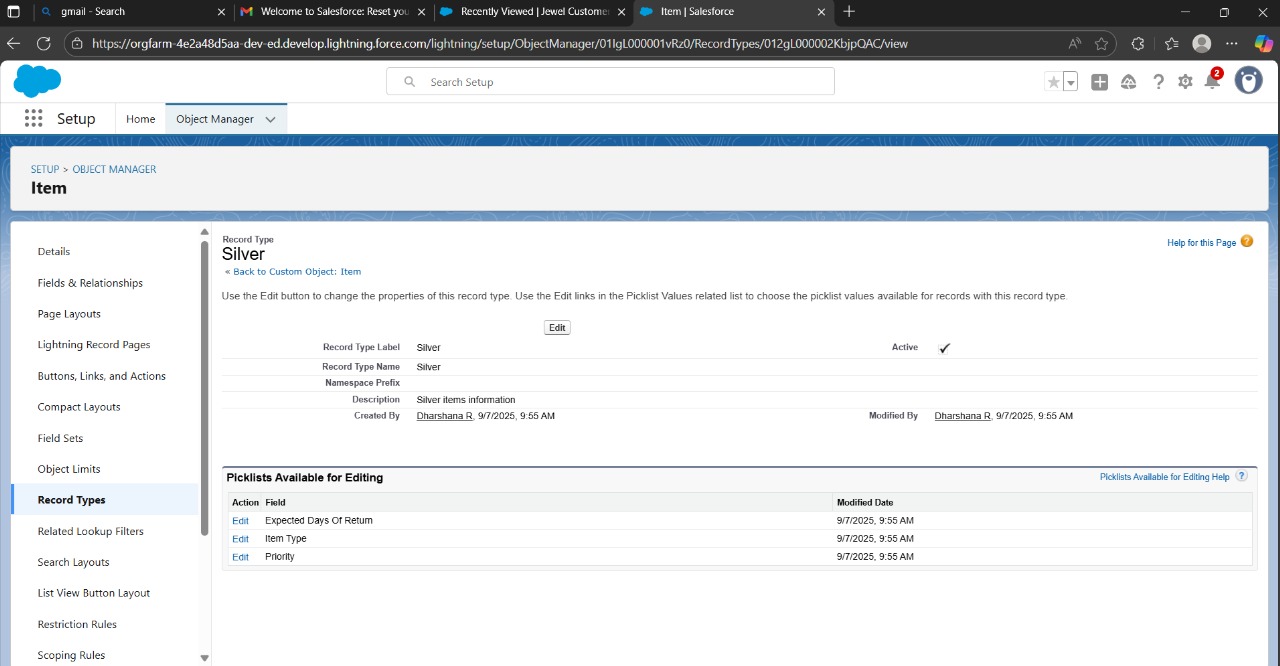
****

**RECORD TYPES**

* Created record types : Gold and Silver for item

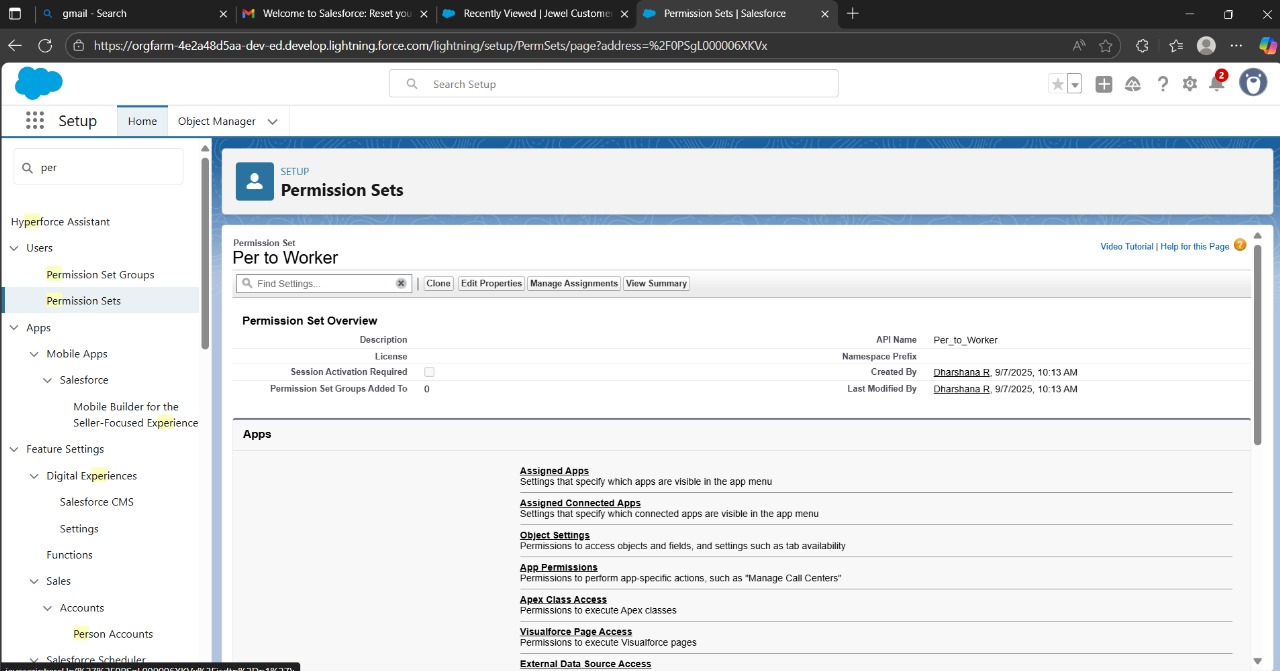






**PERMISSION SET**

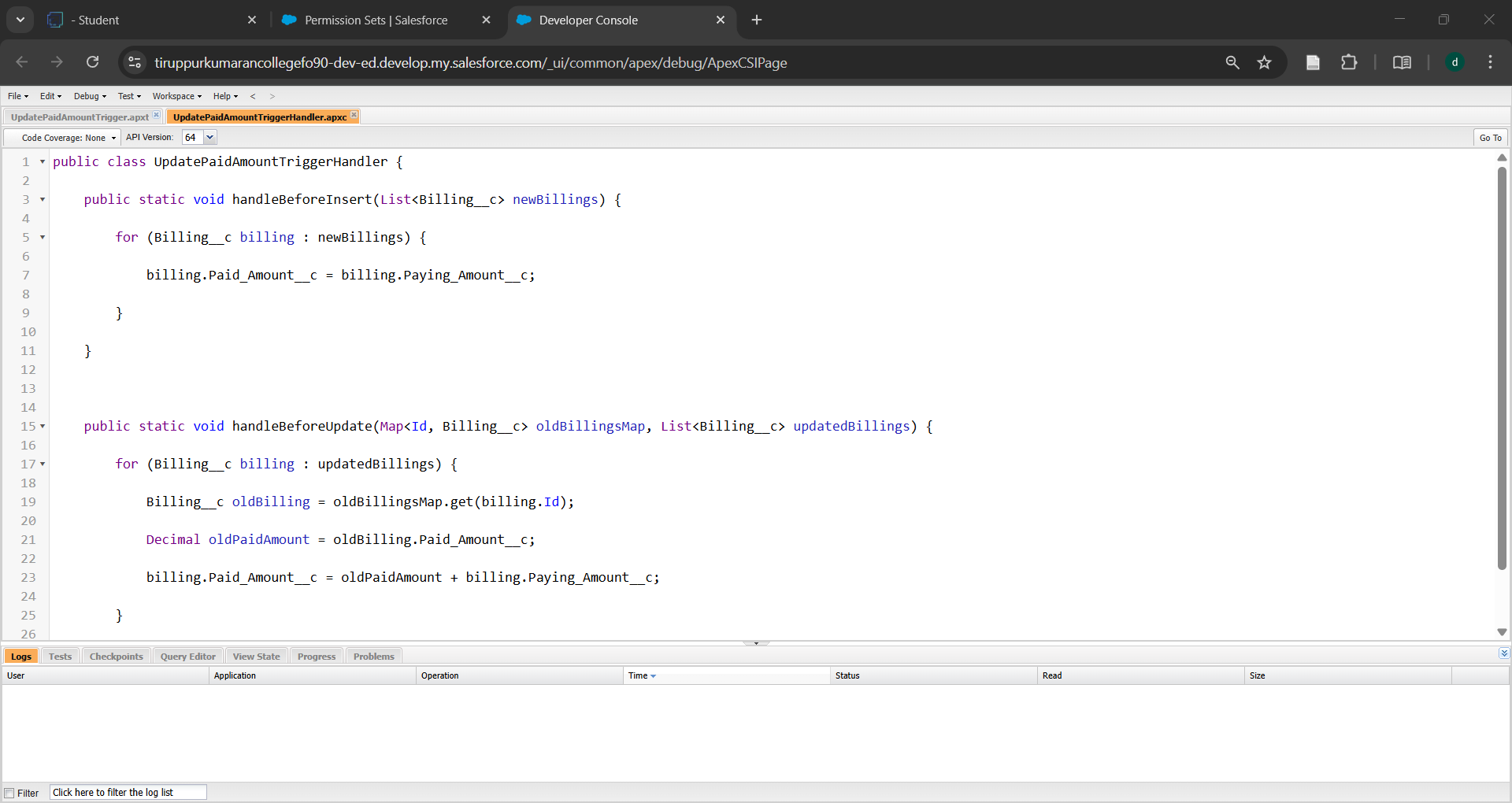
* Created permission set : Per to Worker

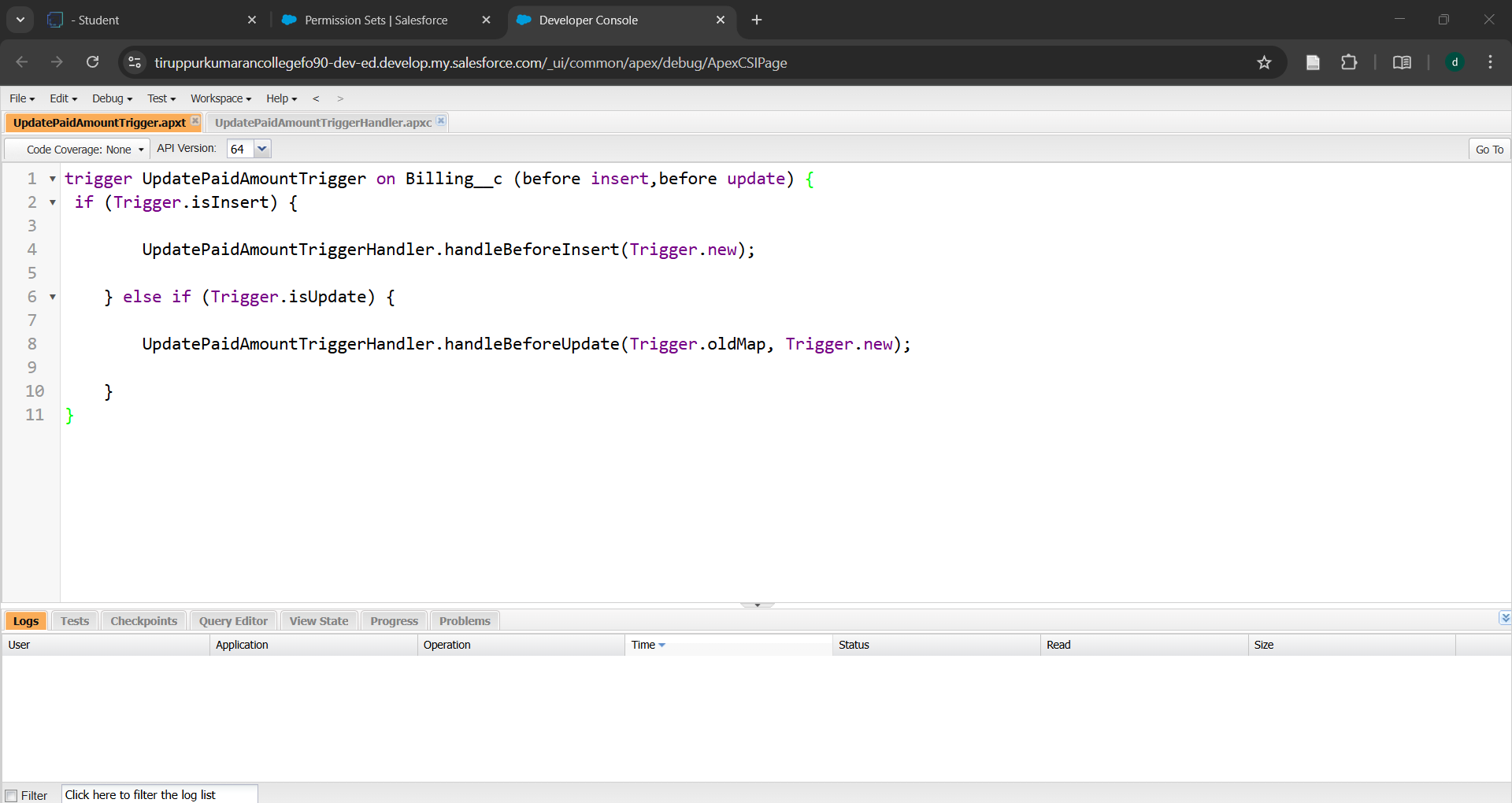


**TRIGGERS**

* Created a trigger handler class: UpdatePaidAmountTriggerHandler

UpdatePaidAmountTrigger





**USER ADOPTION**

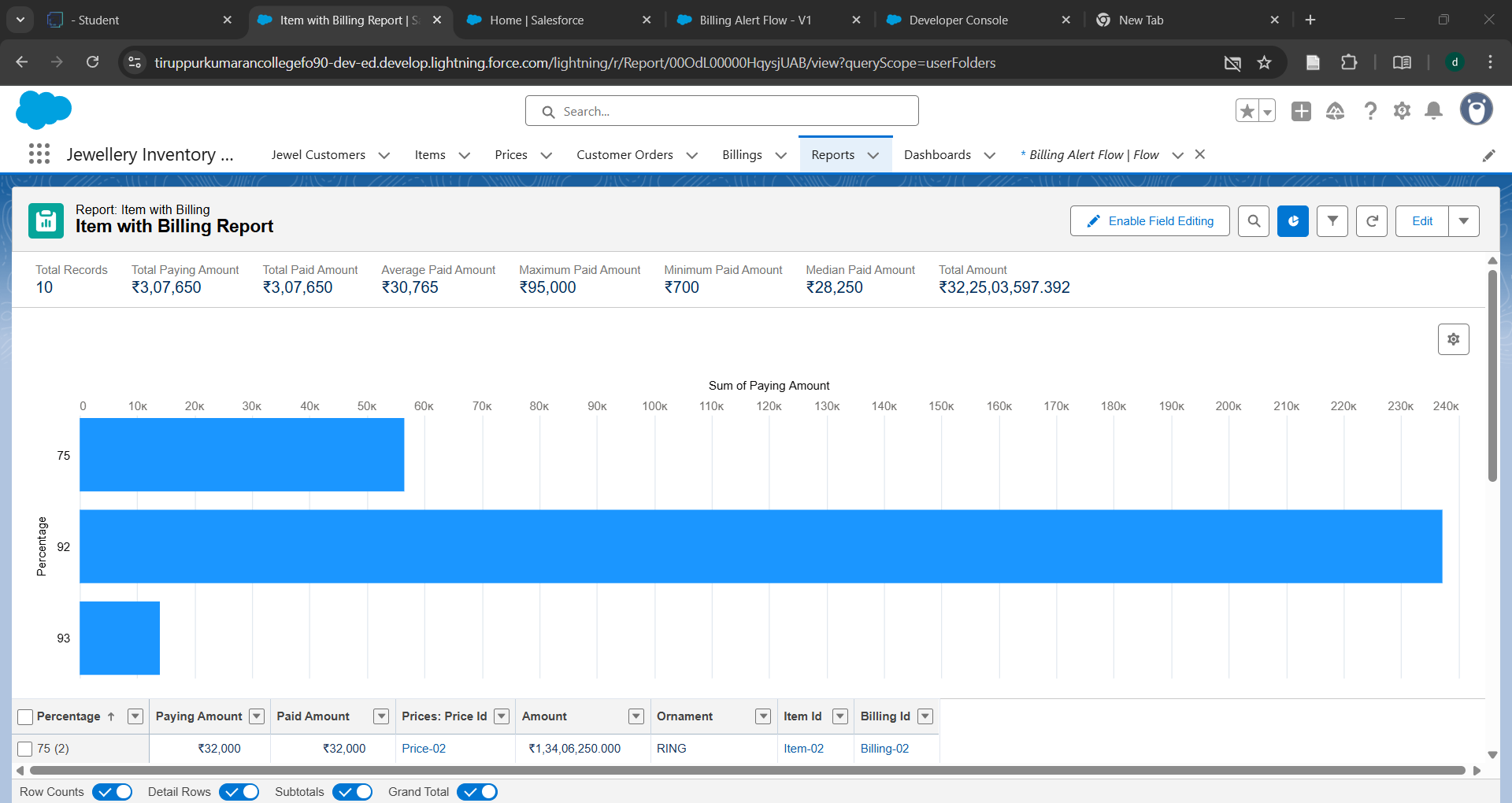
A screenshot of a computer

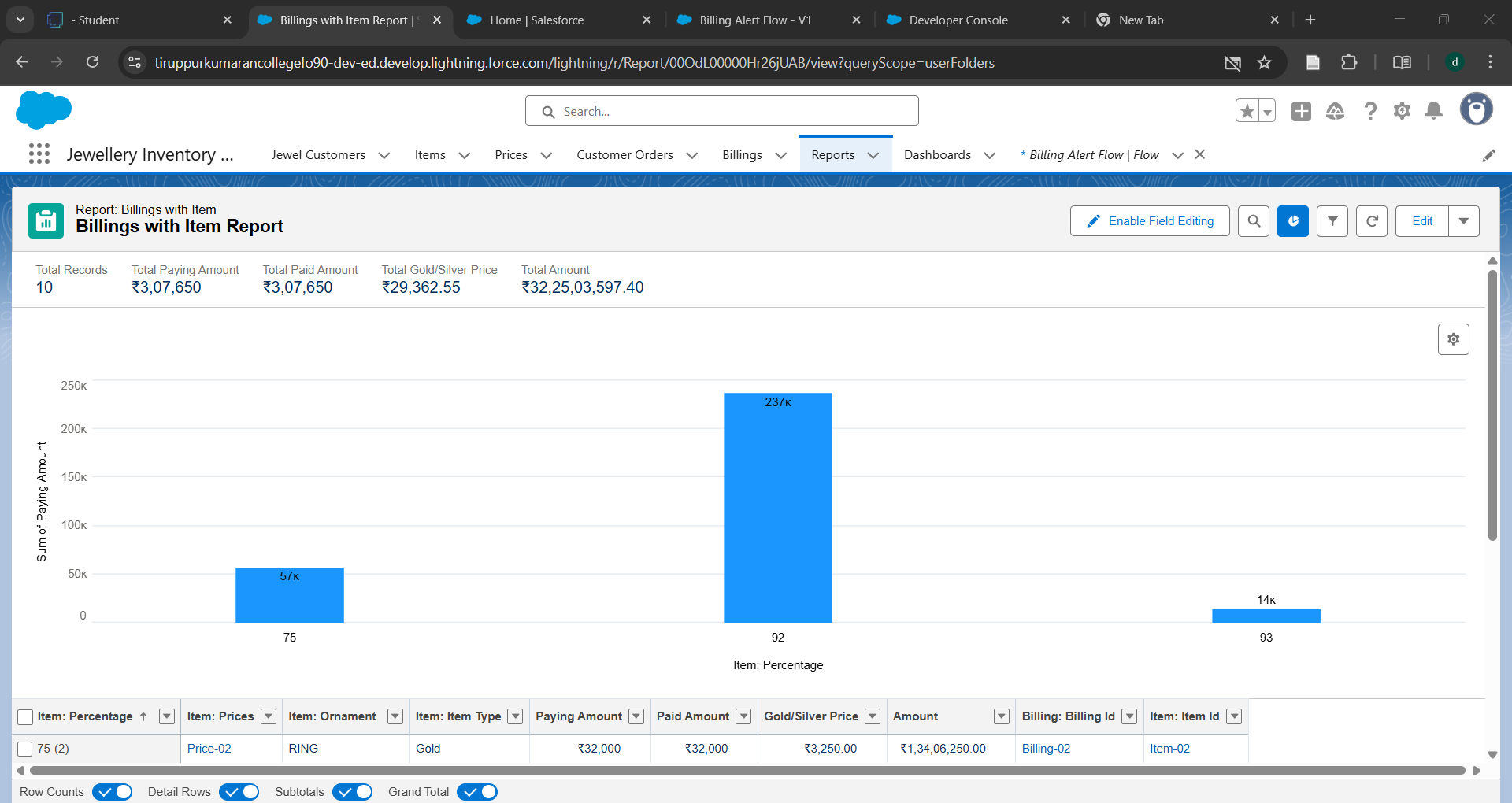
AI-generated content may be incorrect.

**REPORTS**

A screenshot of a computer

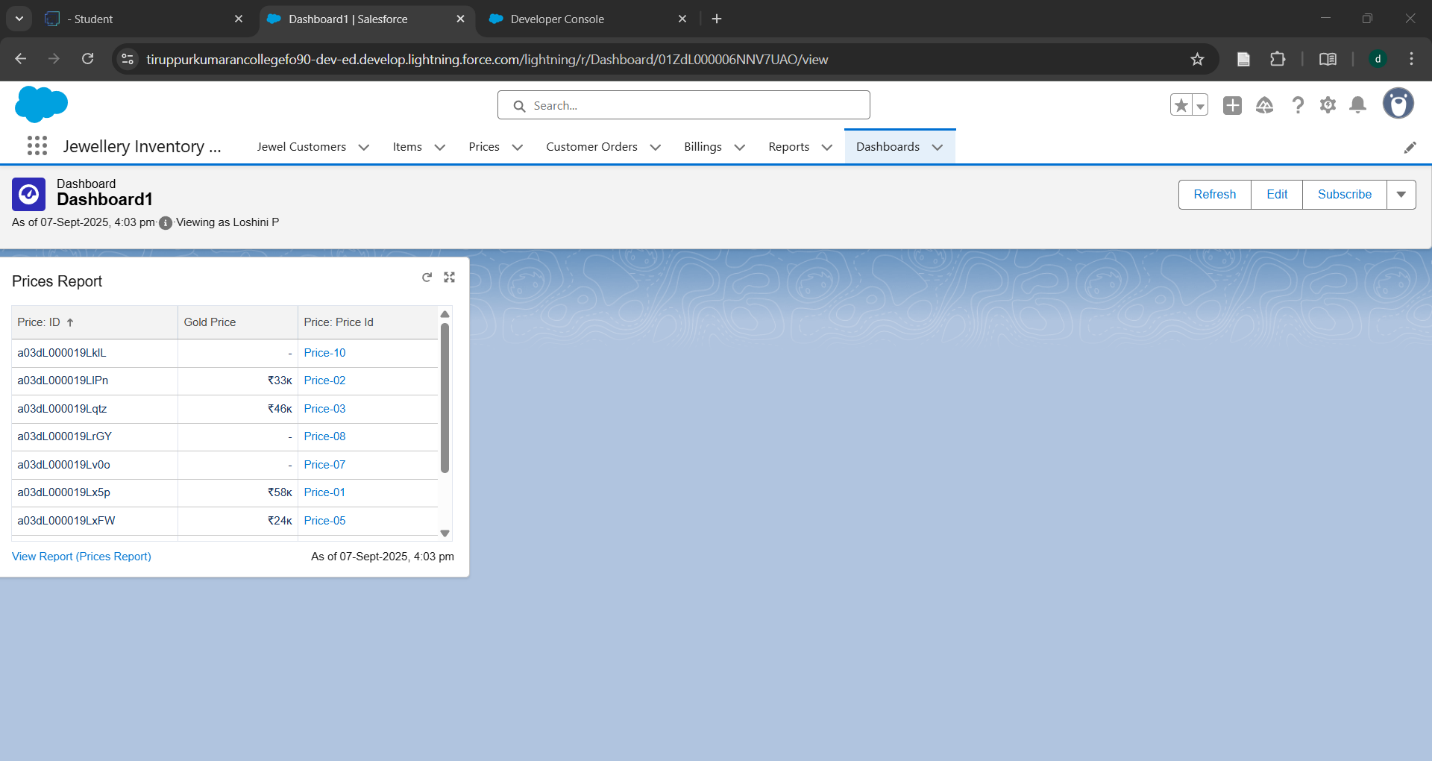
AI-generated content may be incorrect.





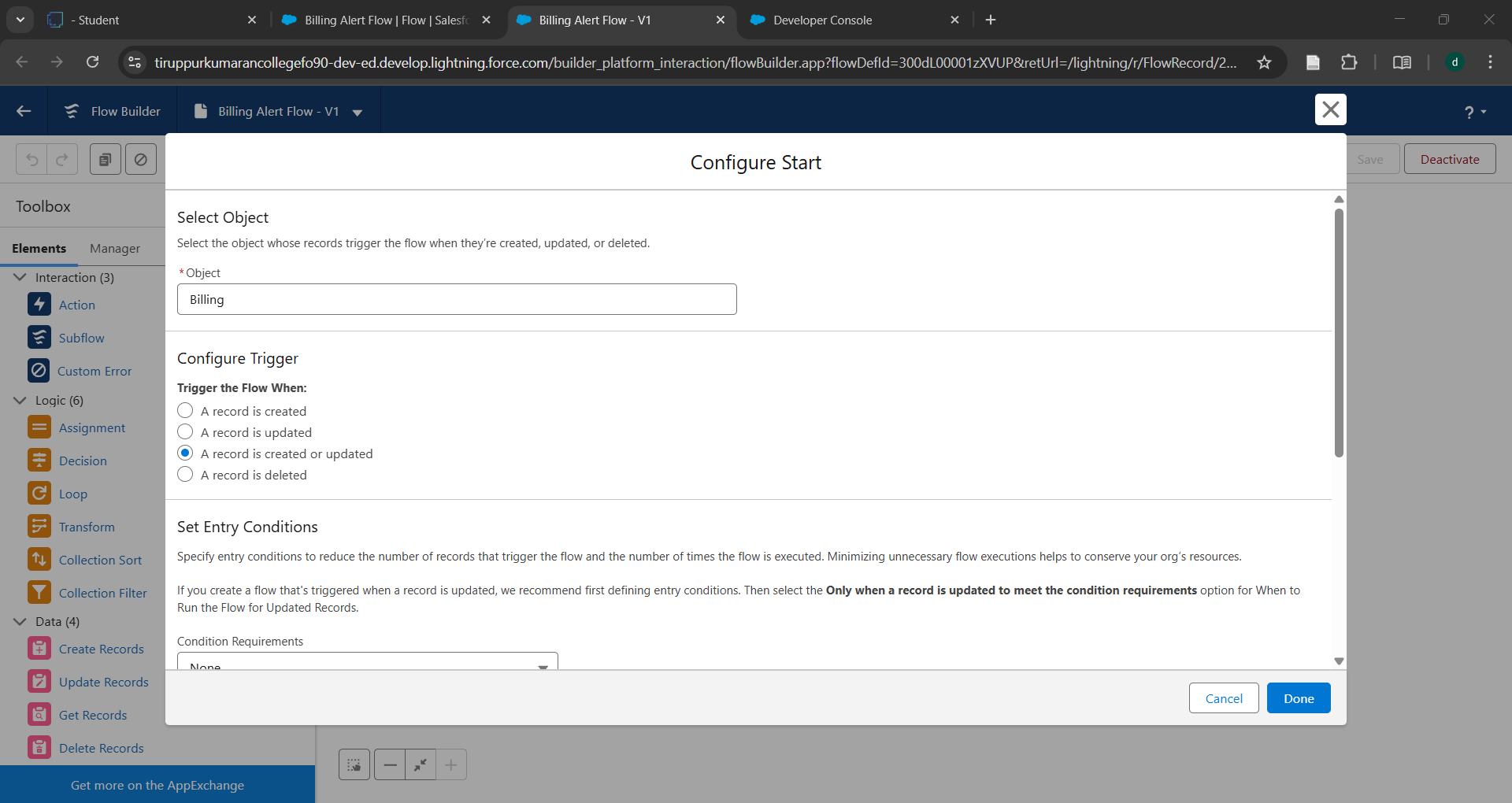
**DASHBOARDS**

* Created Dashboards for reports



**FLOWS**

* Created Flow : Billing Alert Flow



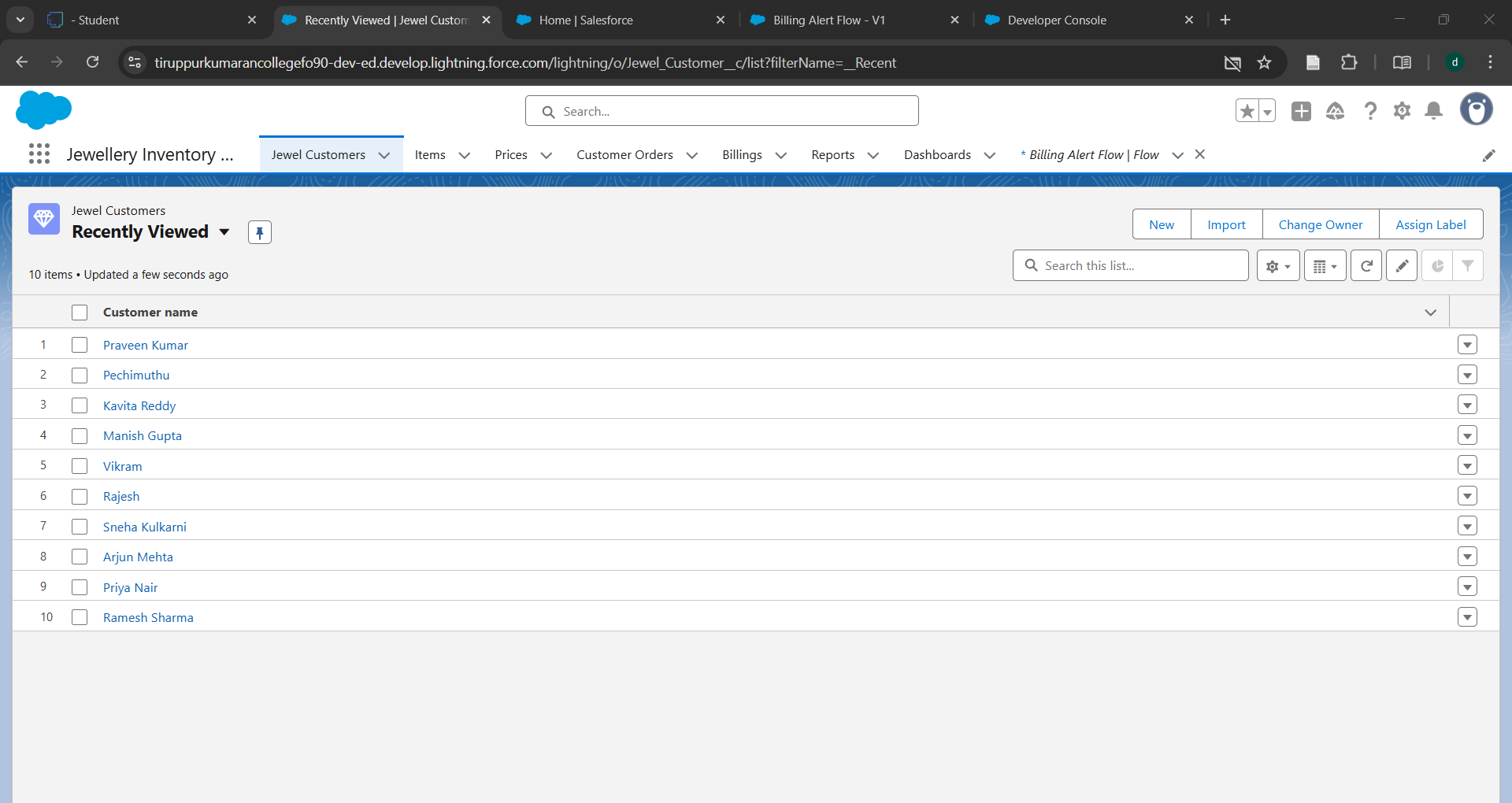
A screenshot of a computer

AI-generated content may be incorrect.

**RESULTS**

**OUTPUT SCREENSHOTS**

* Created records for jewel customer, billing , customer order ,price and item

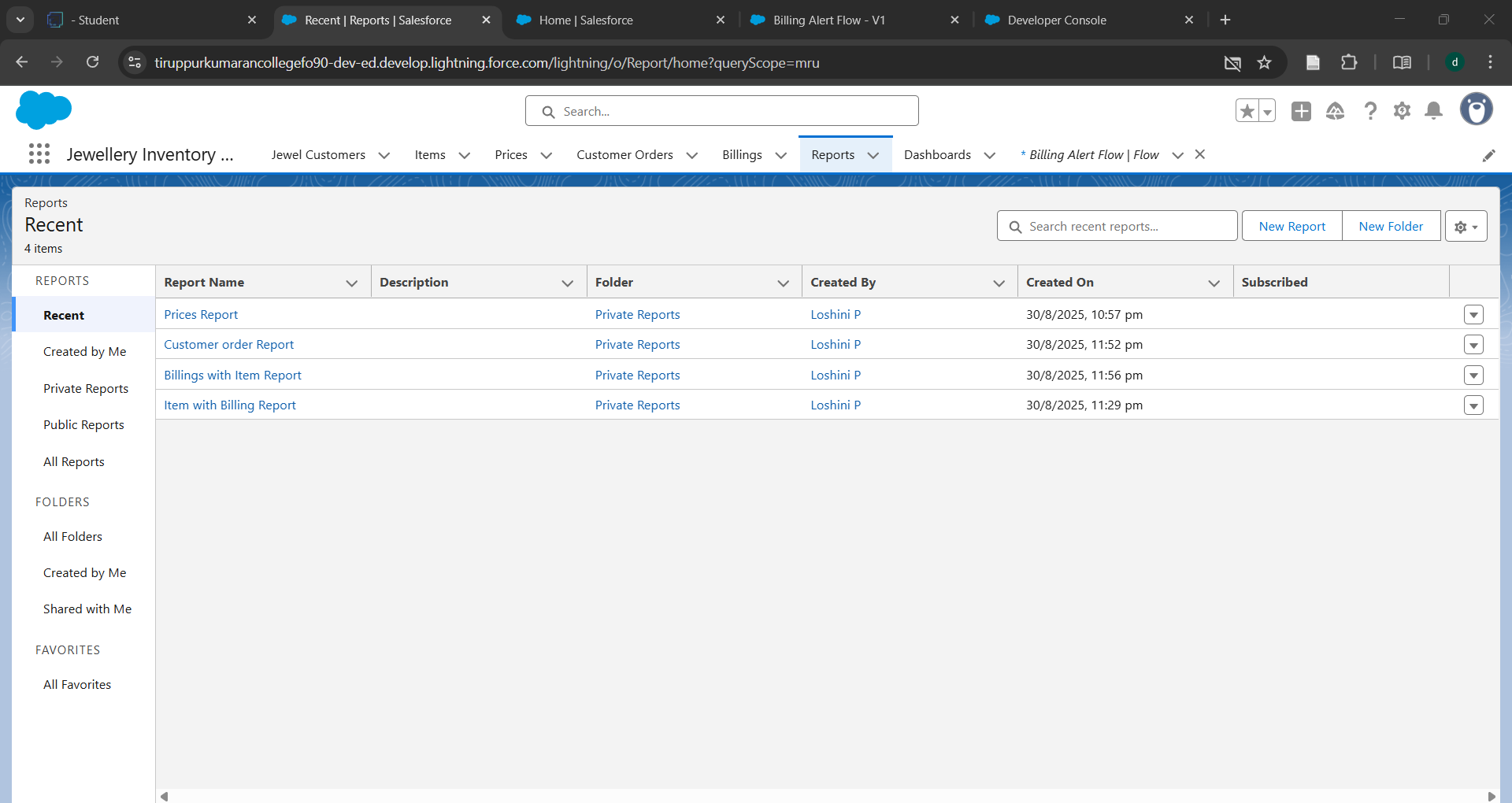
****

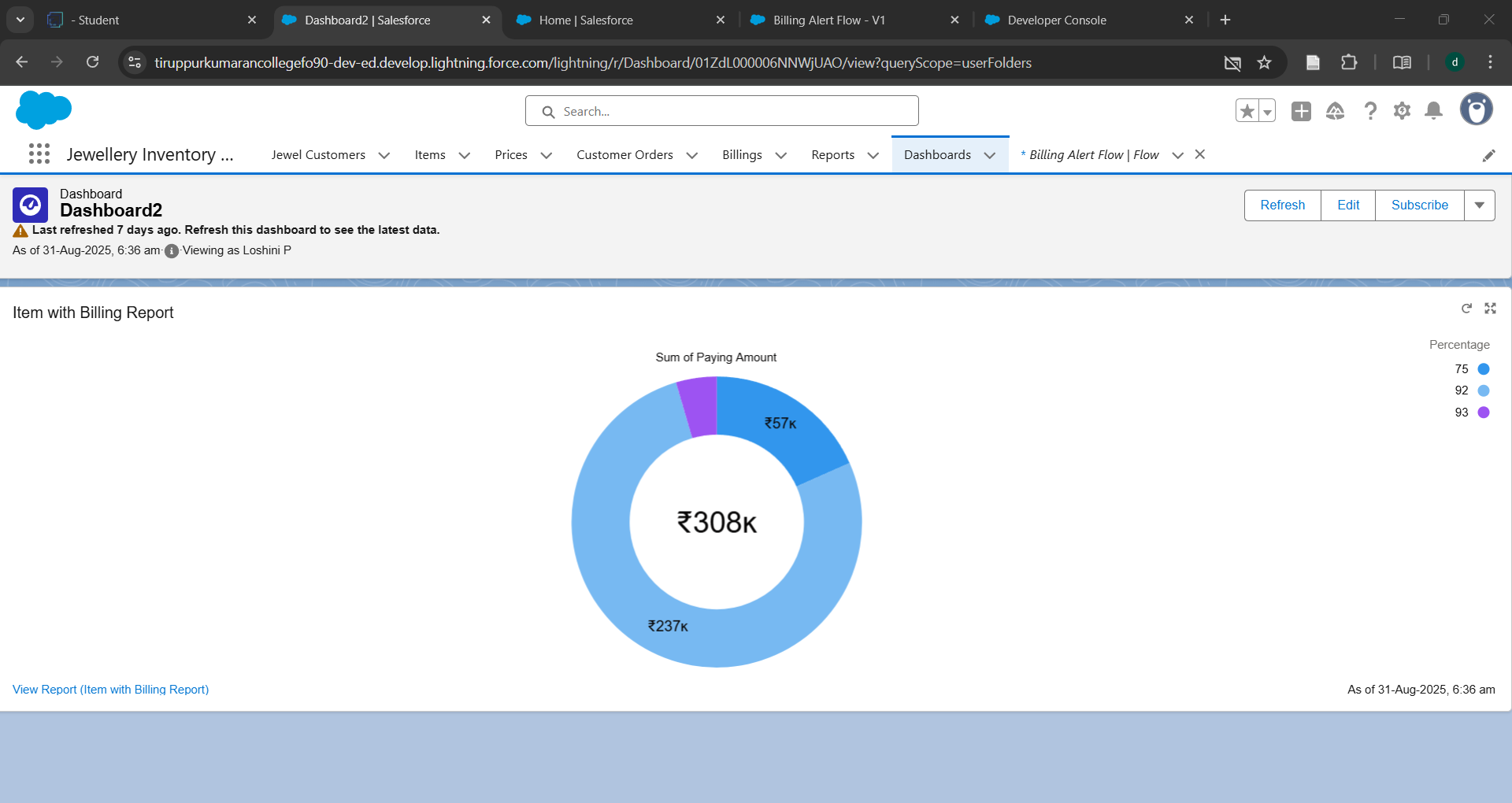
****

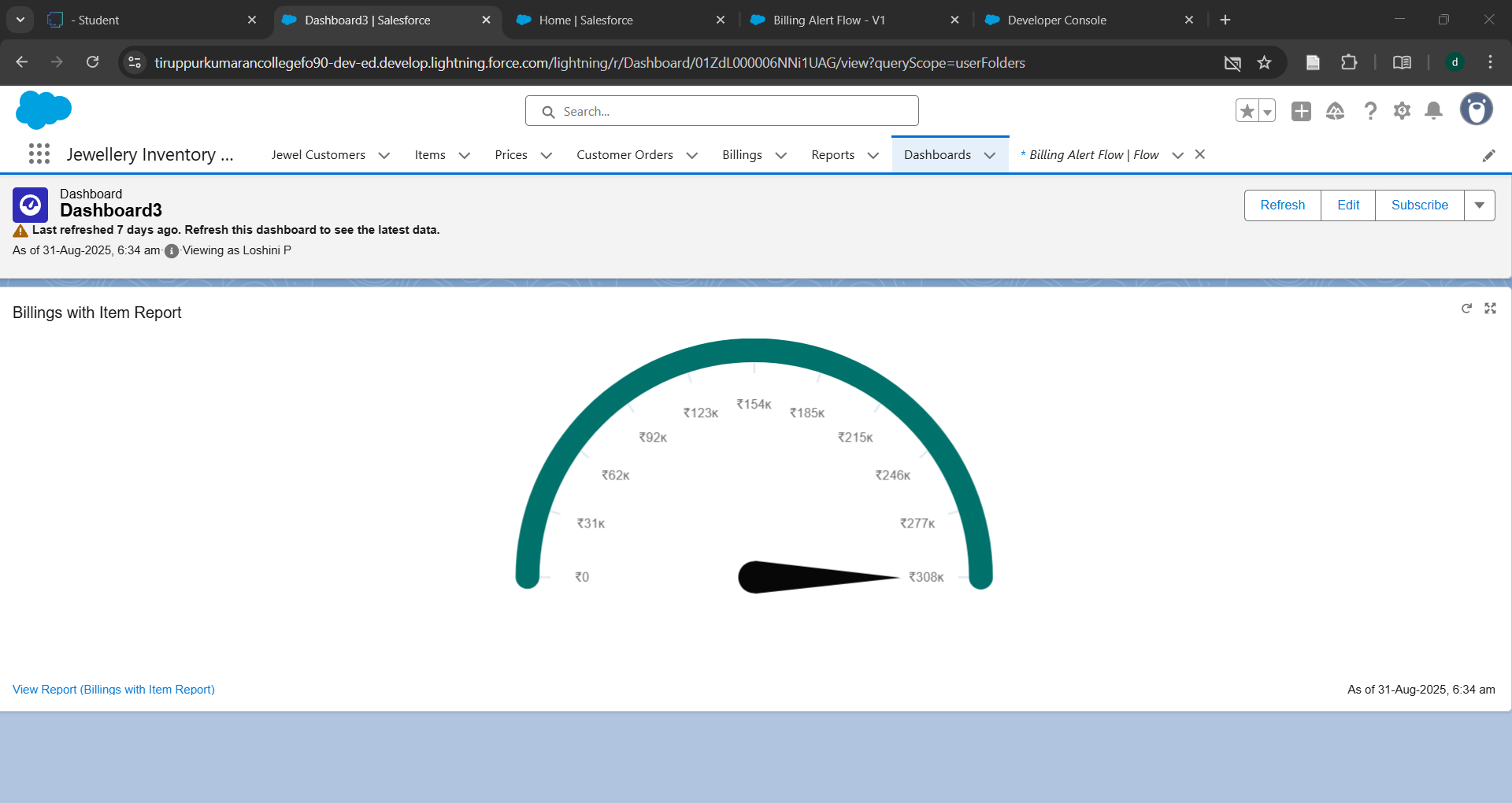
A screenshot of a computer

AI-generated content may be incorrect.

* Created reports and dashboards for jewel inventory system







* Created flow output



**CONCLUSION**

The CRM application for jewellery management effectively streamlines customer handling, sales, and inventory processes while enhancing customer relationships. It provides a reliable and scalable solution that supports business growth, improves efficiency, and strengthens customer loyalty in the jewellery industry.

**APPENDIX**

* **Source Code:** Provided in Apex Classes

Trigger handler:

 UpdatePaidAmountTriggerHandler

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

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);

    }

}