**CRM APPLICATION FOR JEWELRY MANAGEMENT DEVELOPER**

COLLEGE NAME: United College of arts and science

College code-bubrl

TEAM ID:

TEAM MEMBERS:

Team Leader Name: AMRITHA S

Email Id: amrithasaravanan1929@gmail.com

Team memeber1: ABINESH M

Email id: abineshabi90258@gmail.com

Team member 2: AKASH M

Email id: akashakash78082@gmail.com

Team member3: BHUVANESHWARI S

Email id: buvanesanu85@gmail.com

1. **INTRODUCTION**

In today’s highly competitive jewelry industry, customer satisfaction and relationship building play a crucial role in business success. Developing a Customer Relationship Management (CRM) application for jewelry management provides an effective way to streamline operations, enhance customer interactions, and improve overall business efficiency. A CRM system tailored for jewelry management helps in tracking customer preferences, managing sales and purchases, handling inventory, offering personalized services, and maintaining after-sales support. By integrating modern technologies, a jewelry CRM application not only strengthens customer loyalty but also supports data-driven decision-making, ensuring that businesses can meet the unique demands of the jewelry market effectively.

1.**1PROJECT OVERVIEW**

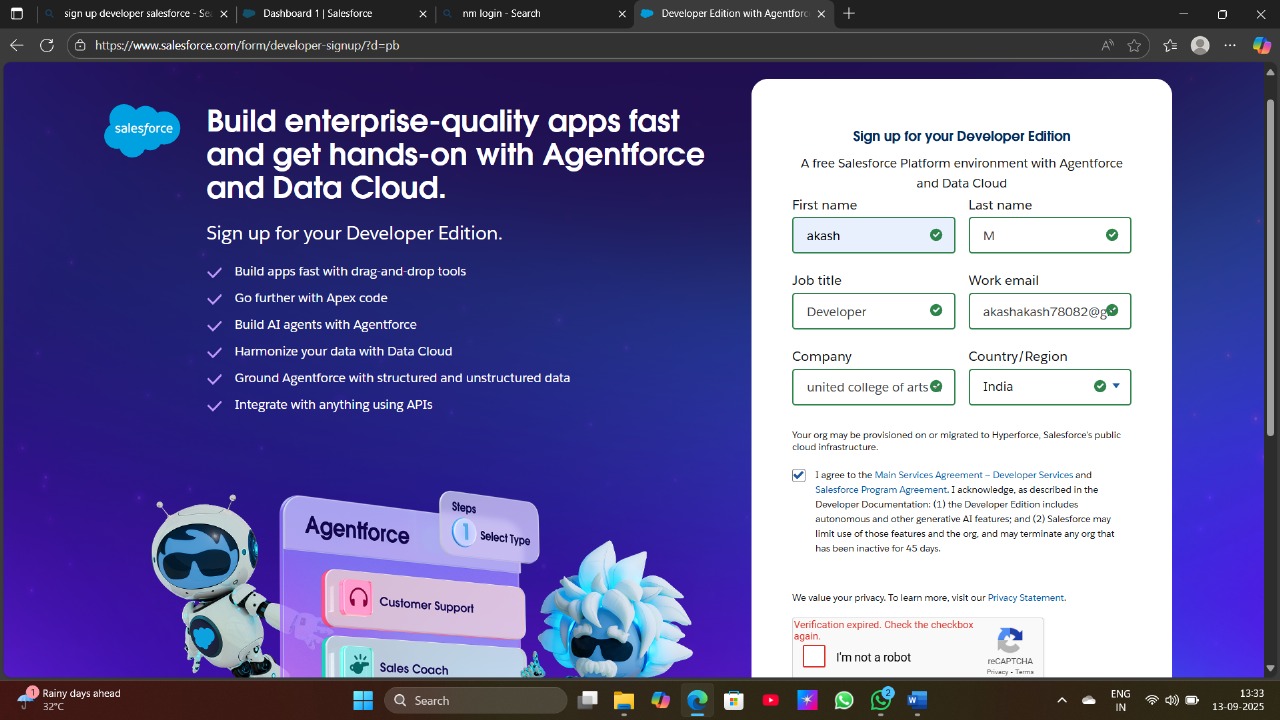
The Jewel Inventory System is a comprehensive software Solution designed to streamline and manage the inventory and sales processes of a jewellery store or a jewellery manufacturer. The system aims to provide an efficient and user-friendly solution to track and control the inventory of various jewellery items, maintain accurate records, and facilitate seamless sales transactions.

**1.2 PURPOSE**

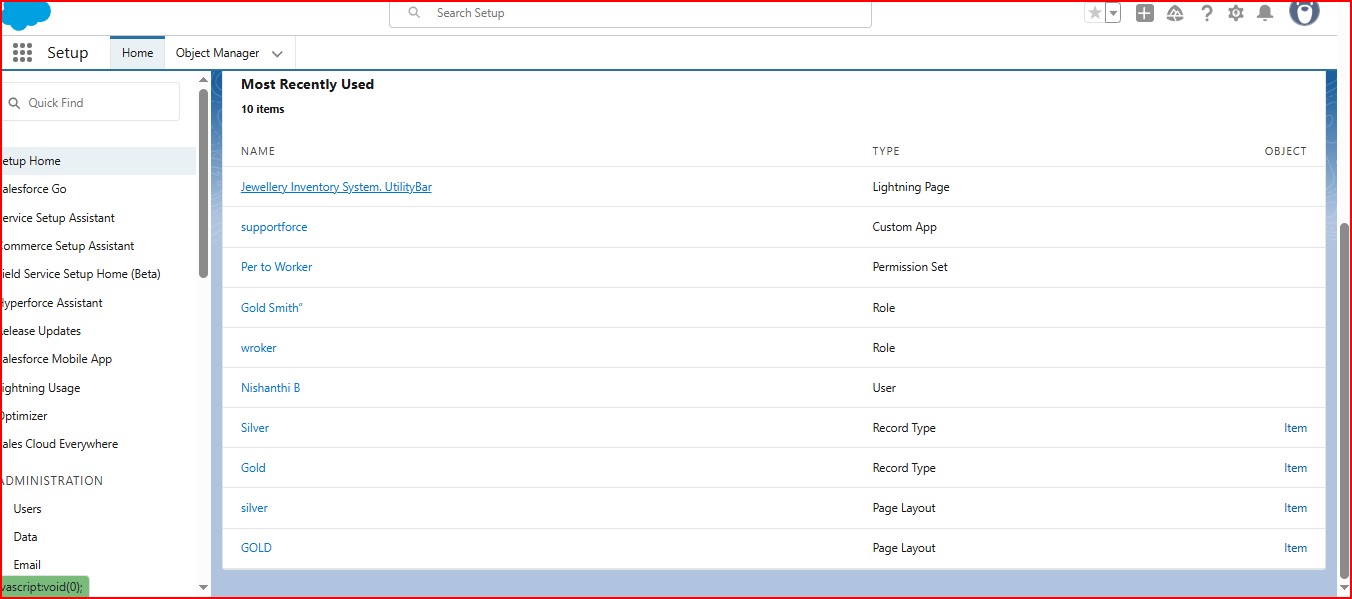
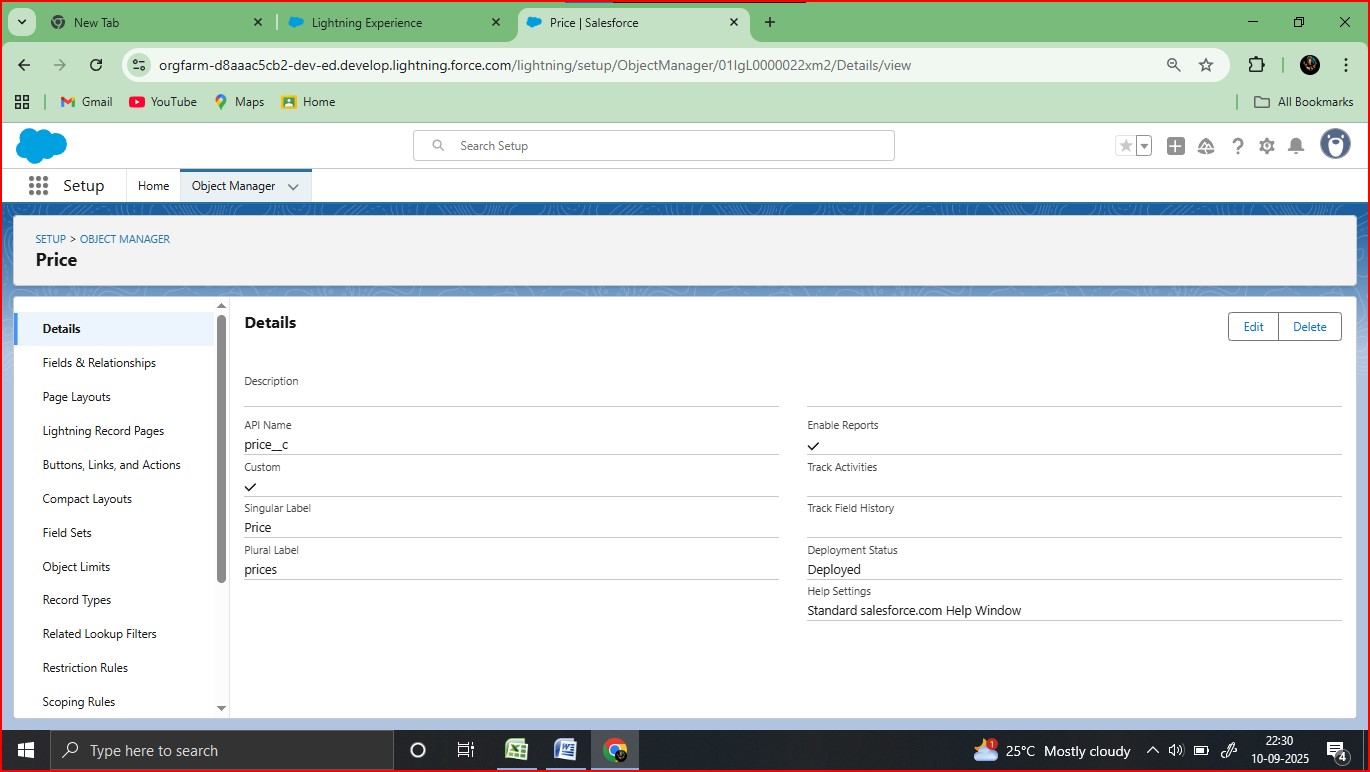
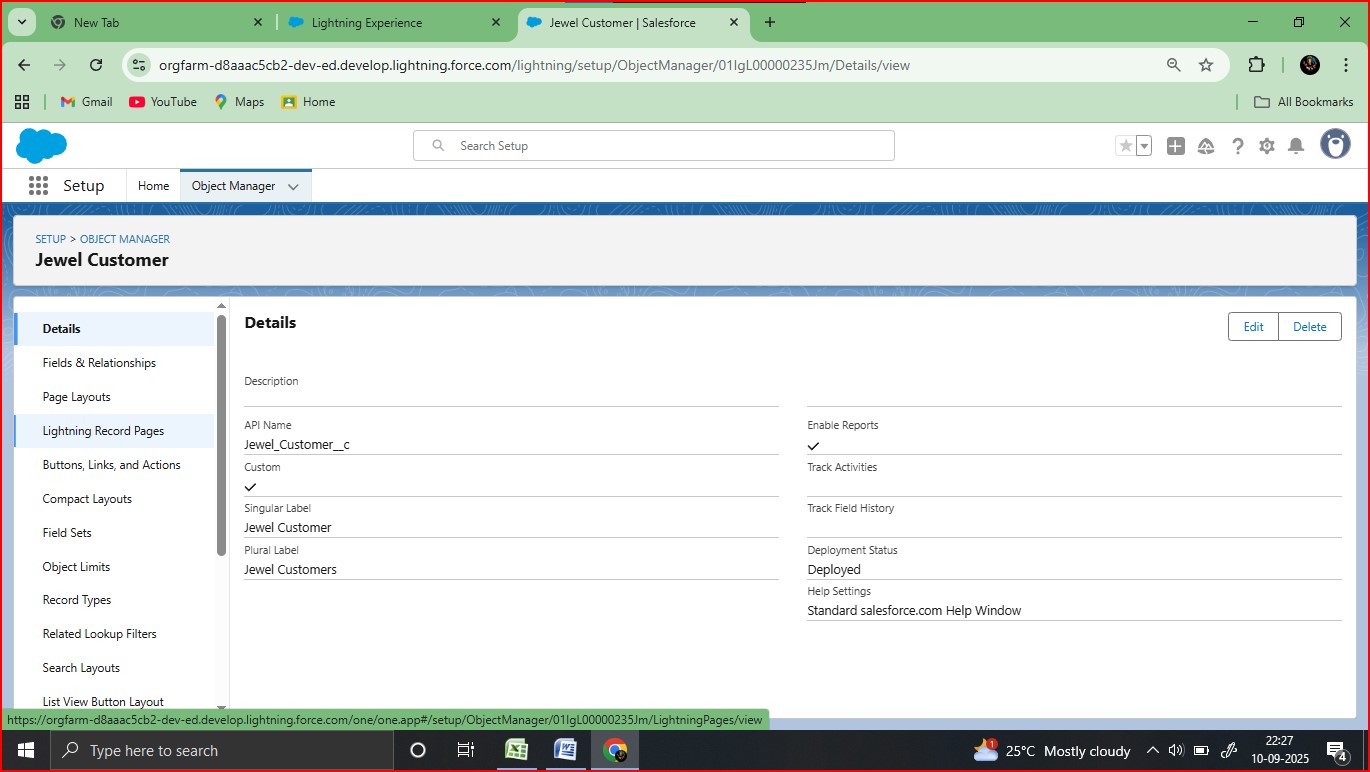
The purpose of jewelry management is to provide an organized system for handling all aspects of jewelry business operations, from inventory tracking to customer relationship management. It ensures that every piece of jewelry is properly cataloged, priced, and monitored, reducing errors and losses. Jewelry management helps streamline sales, purchases, and repairs while maintaining accurate records of stock levels, suppliers, and customer preferences. It also enhances security by monitoring valuable assets, improves customer satisfaction through personalized service, and supports business growth by providing insights into sales trends, demand patterns, and profitability

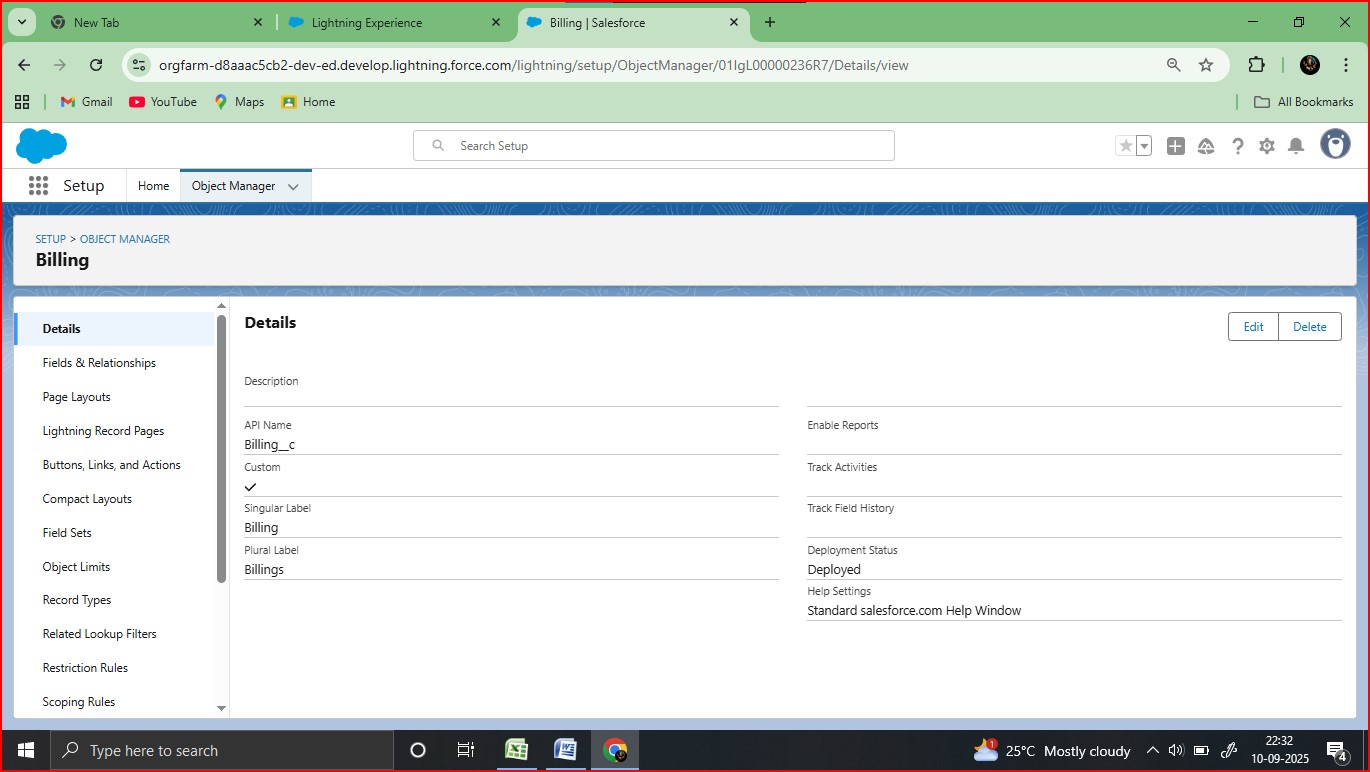
**DEVELOPMENT PHASE.**

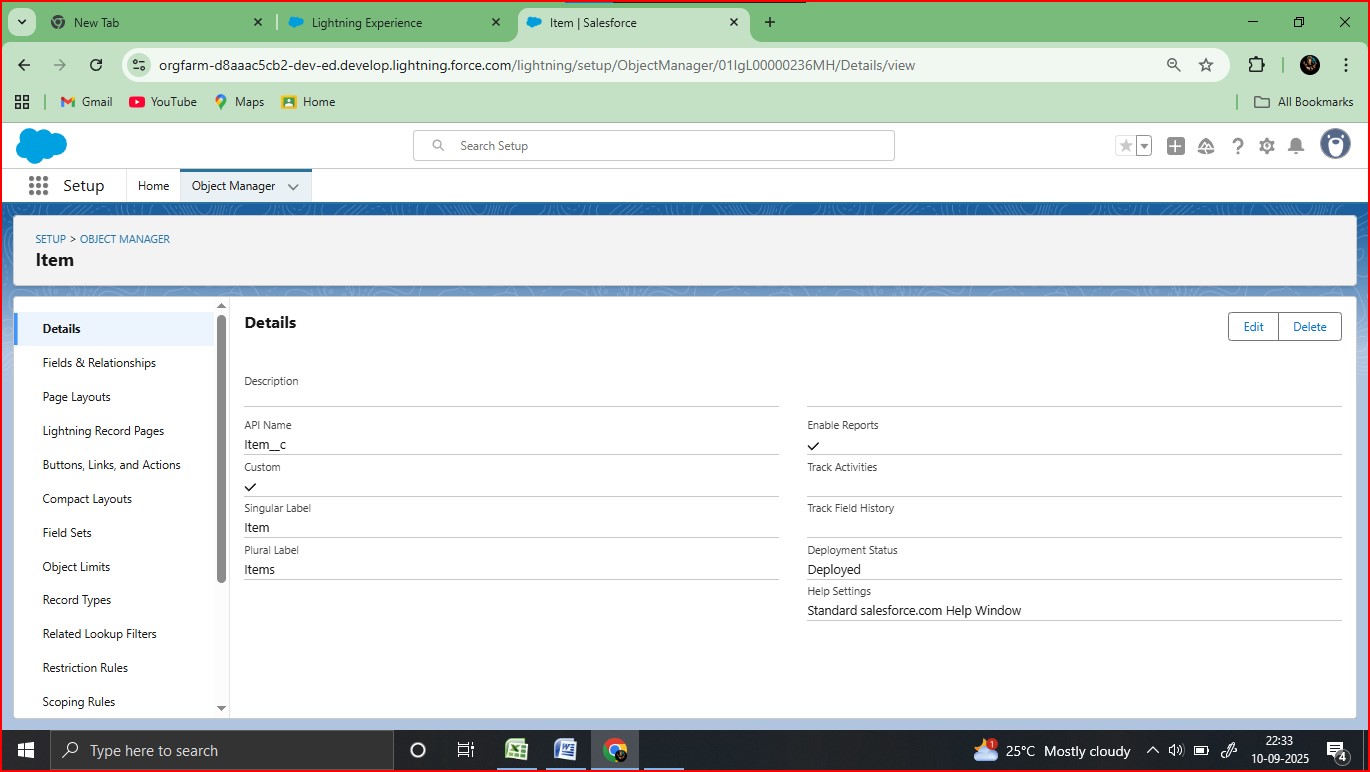
**Creating Developer Account:** <https://developer.salesforce.com/signup>



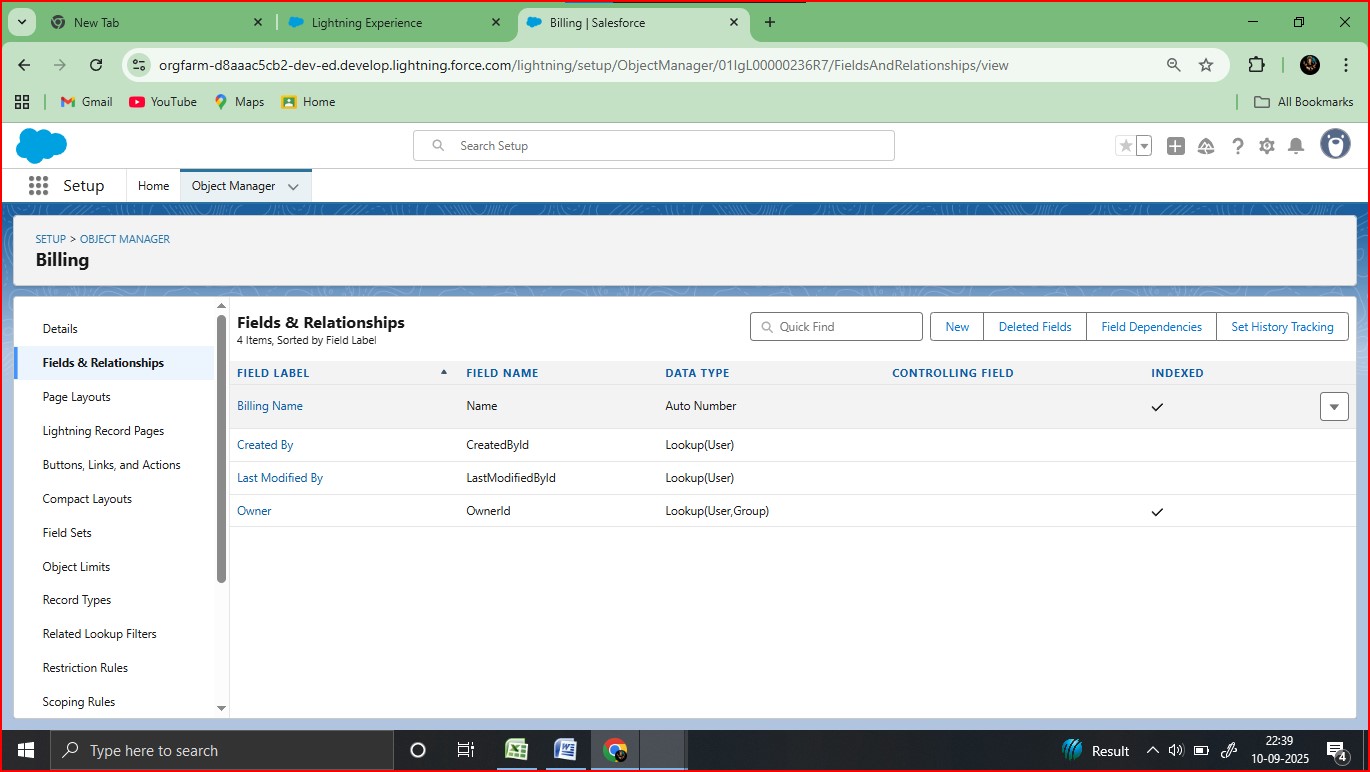
Created object: jewel customer, price, order, billing, item

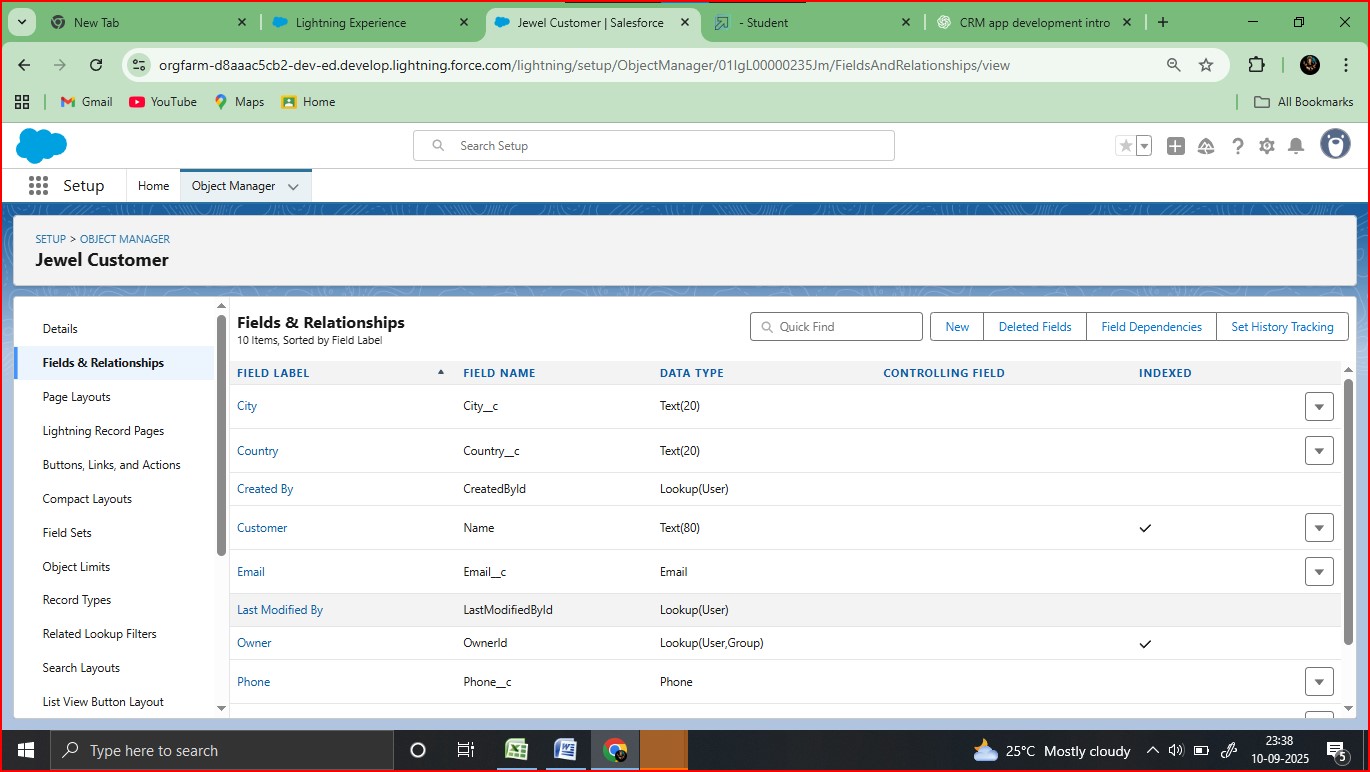
**** ****

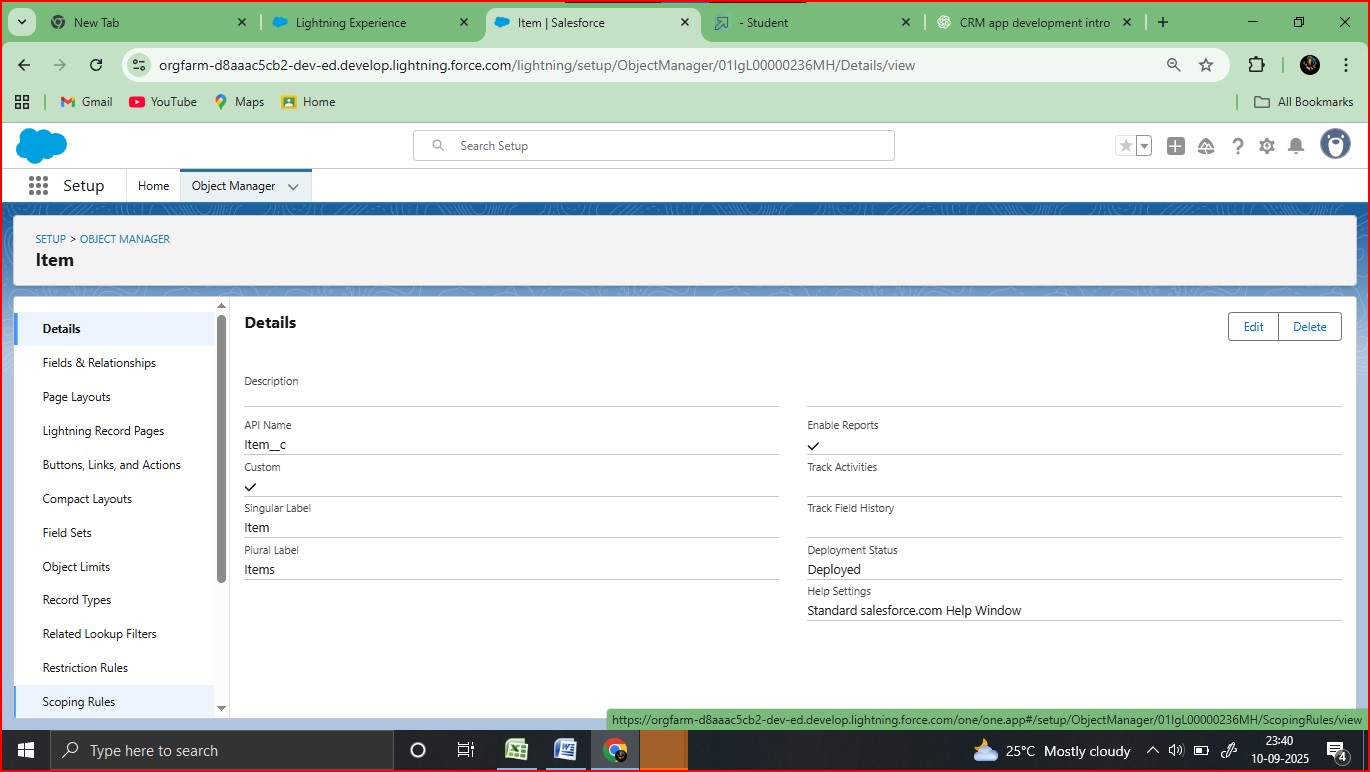


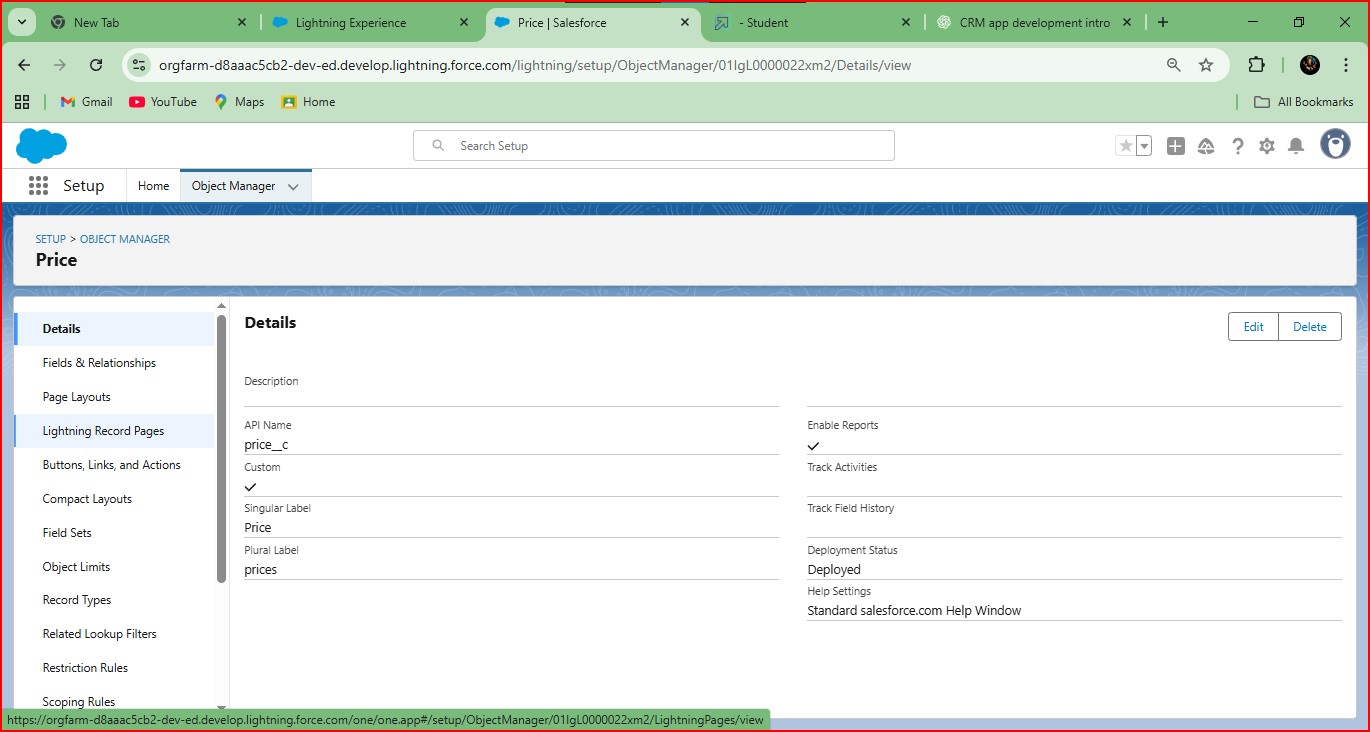


Create field relationship

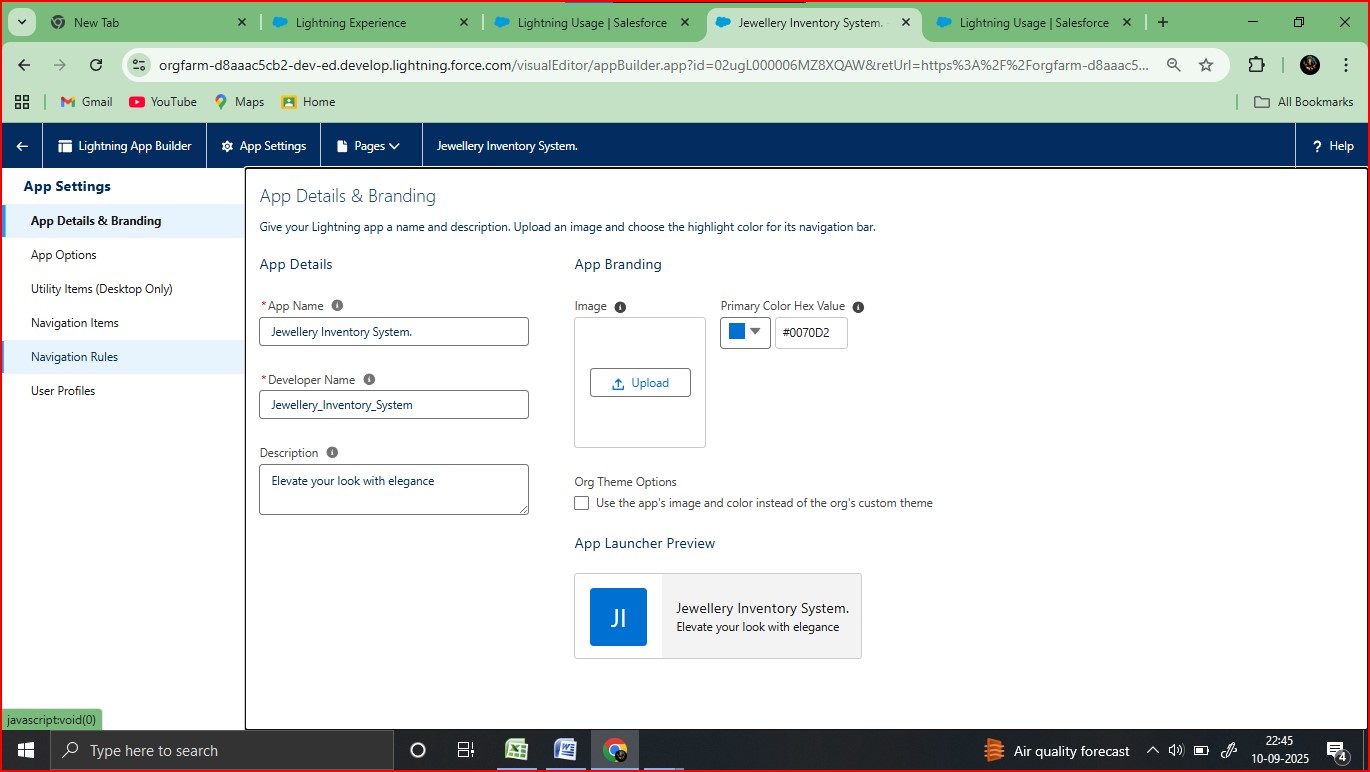


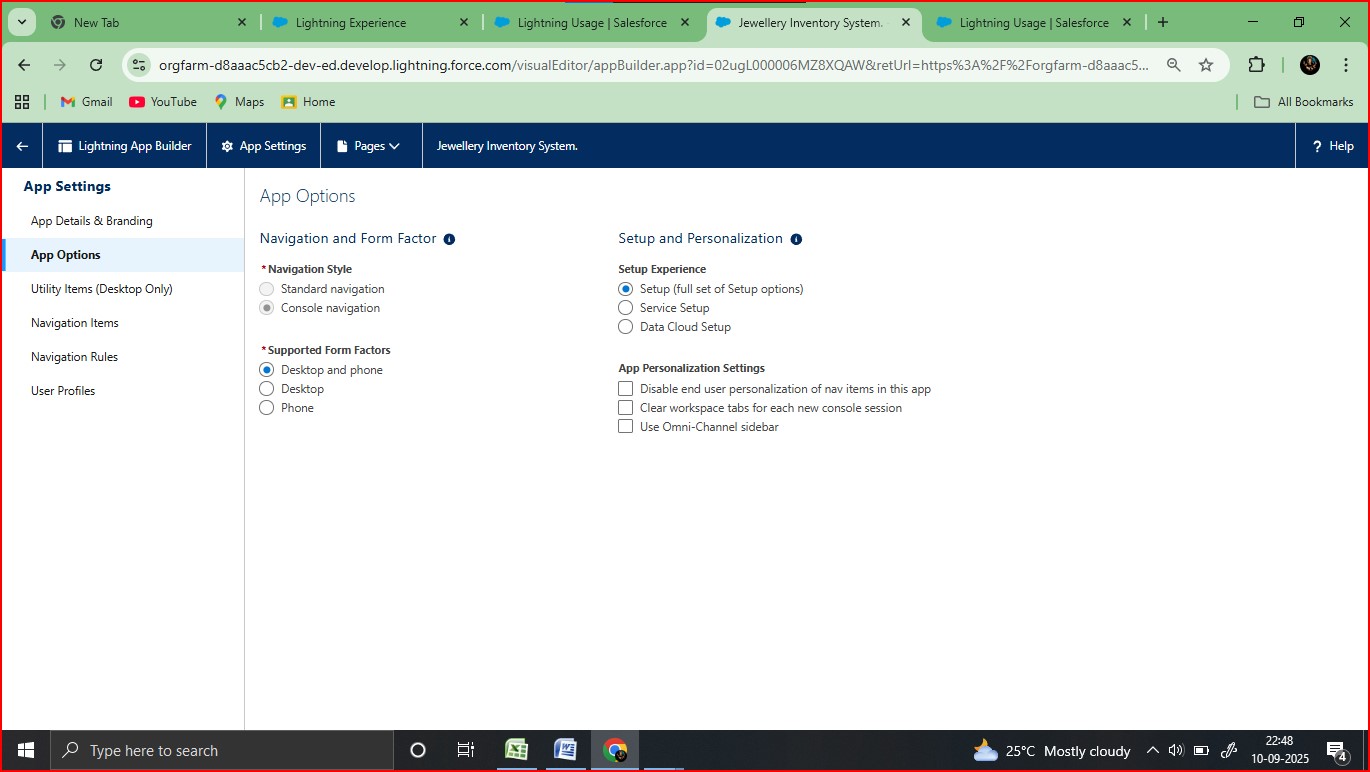


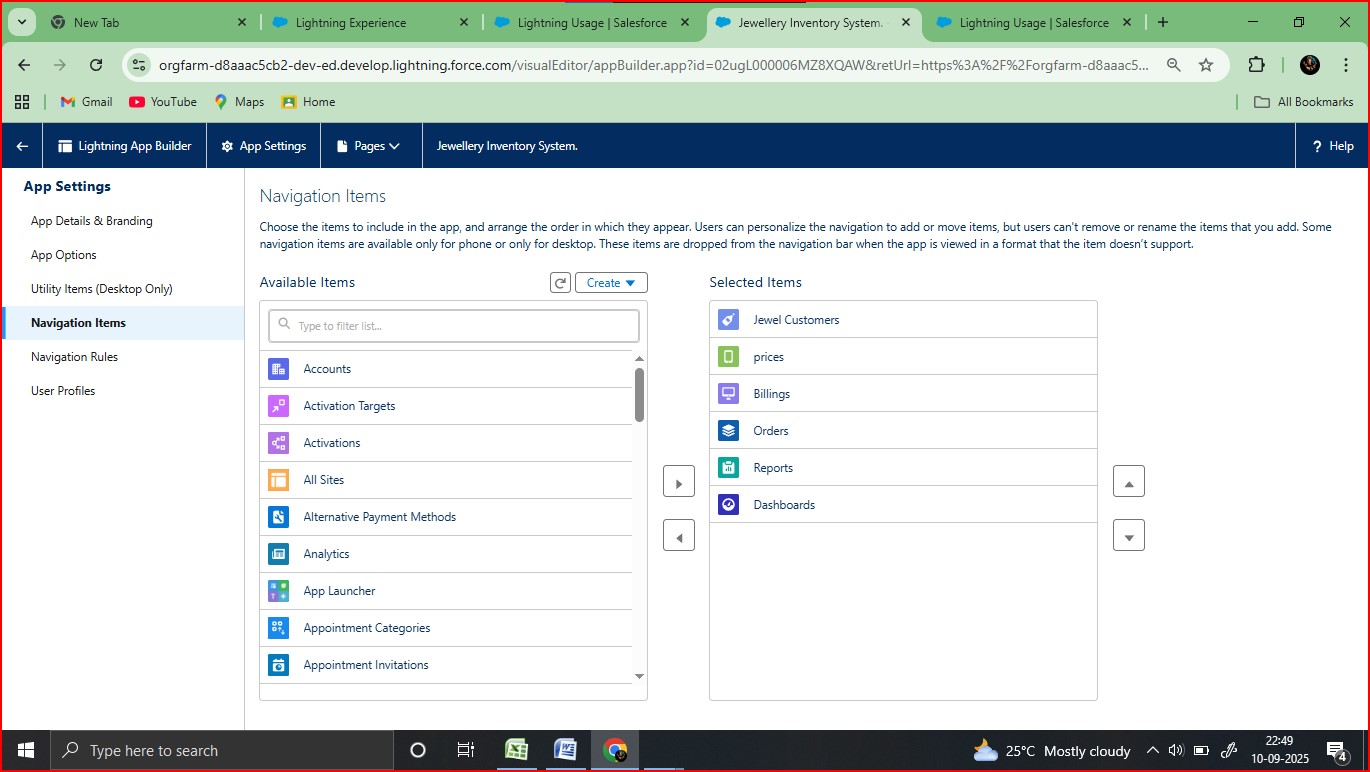


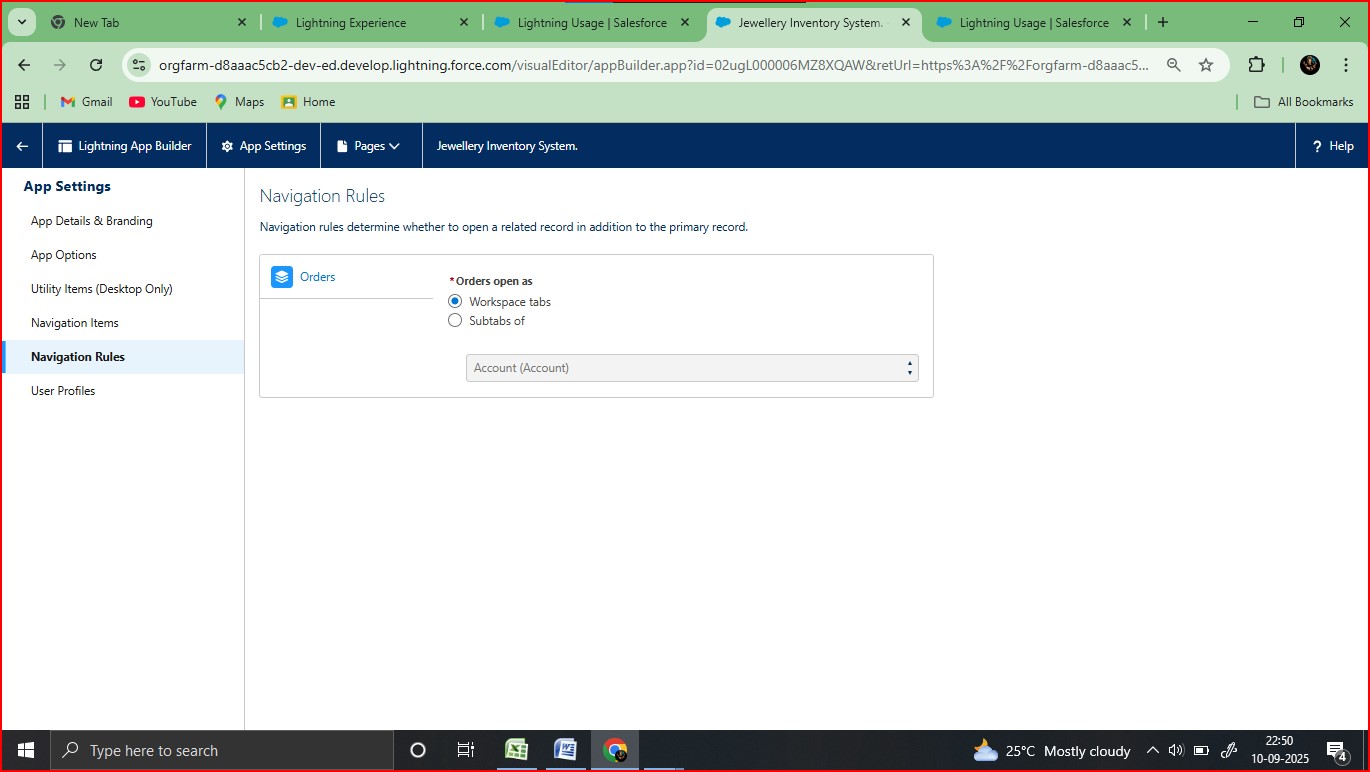


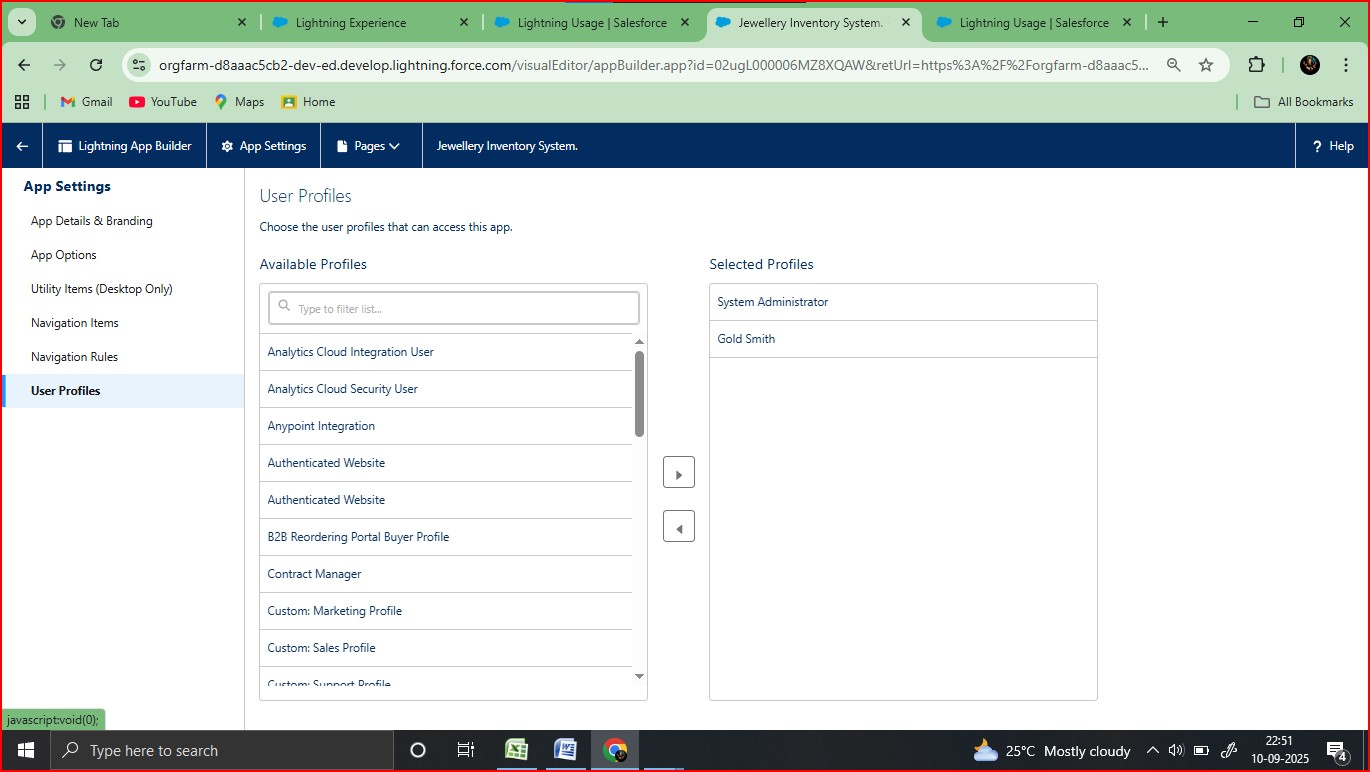
* Developed Lightning App with relevant tabs



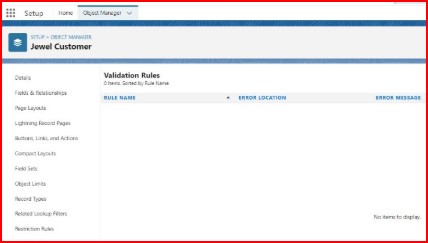


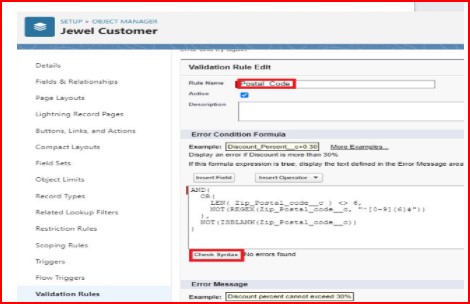




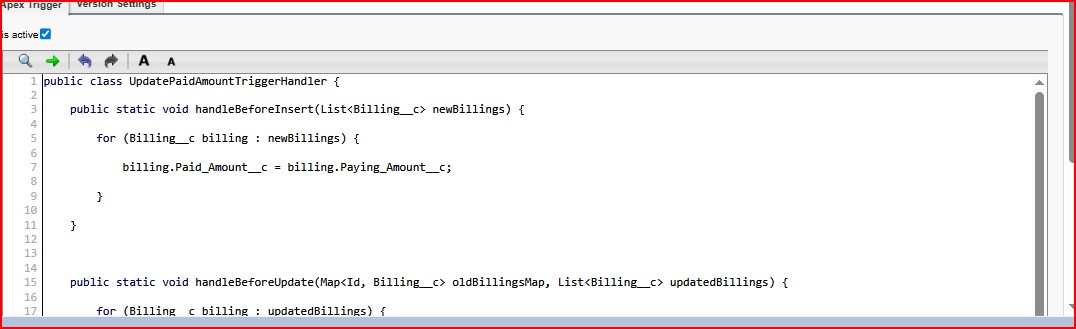


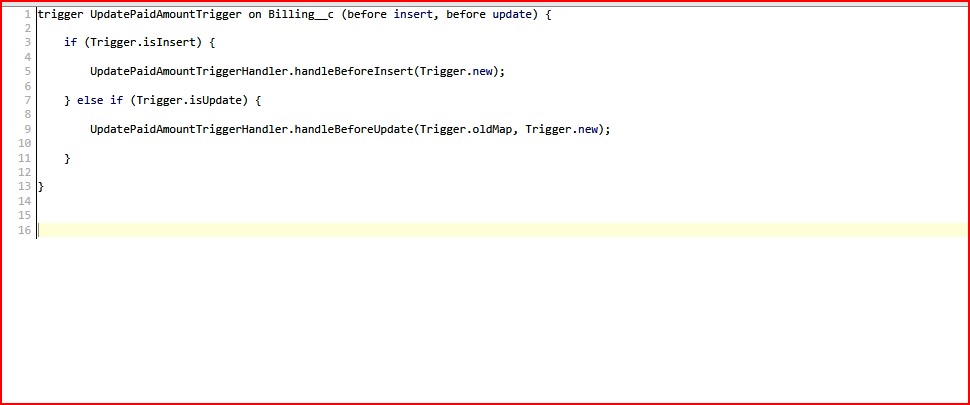
* To create a validation rules



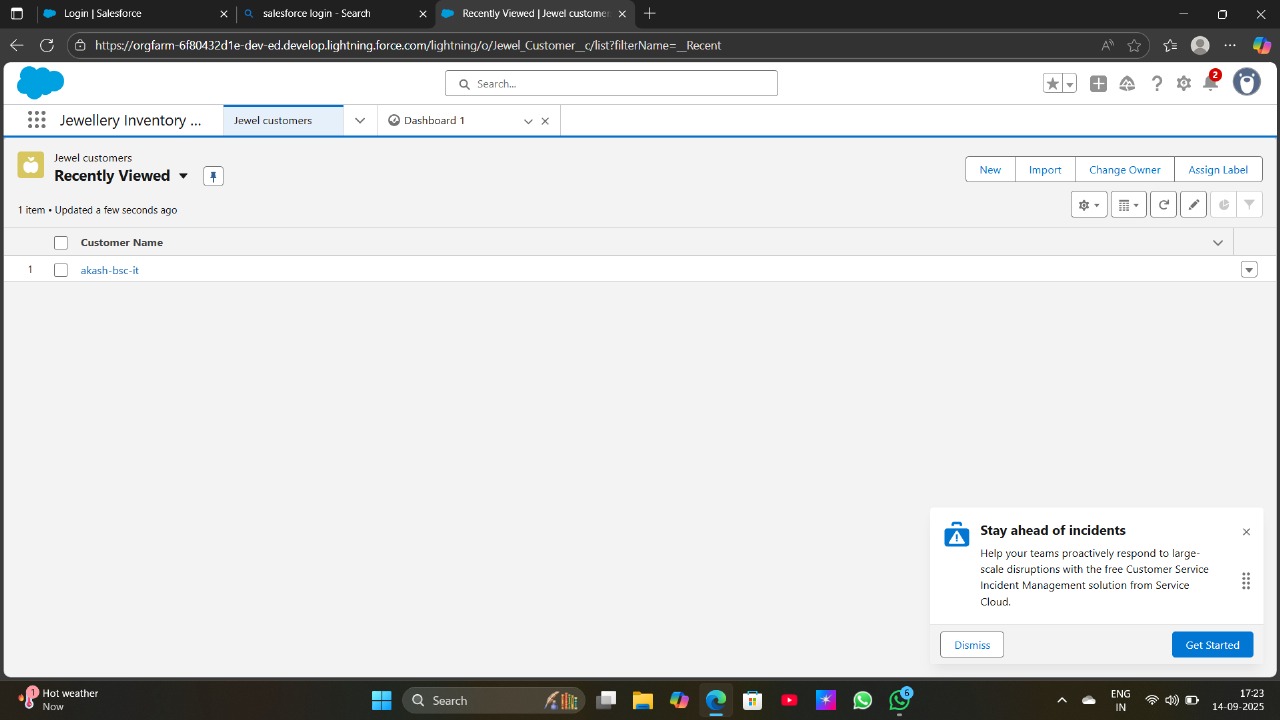


* Create Apex trigger handler



Create a Trigger 

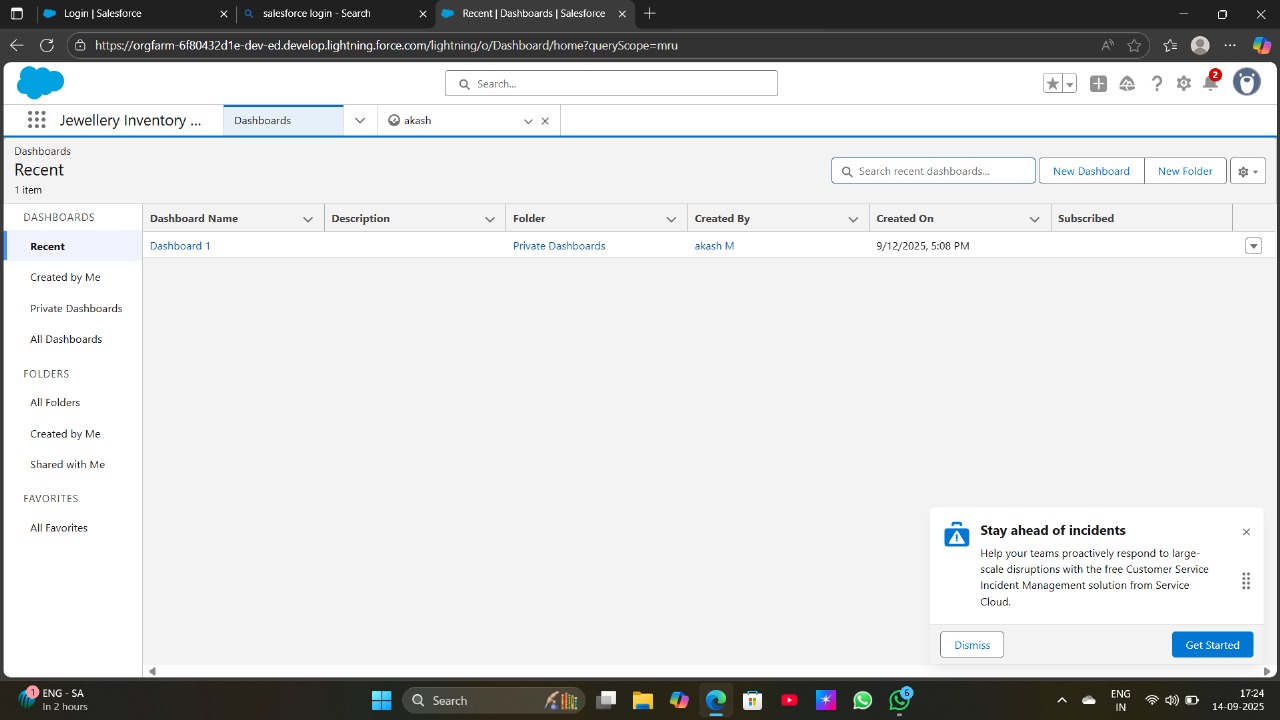
* Flows

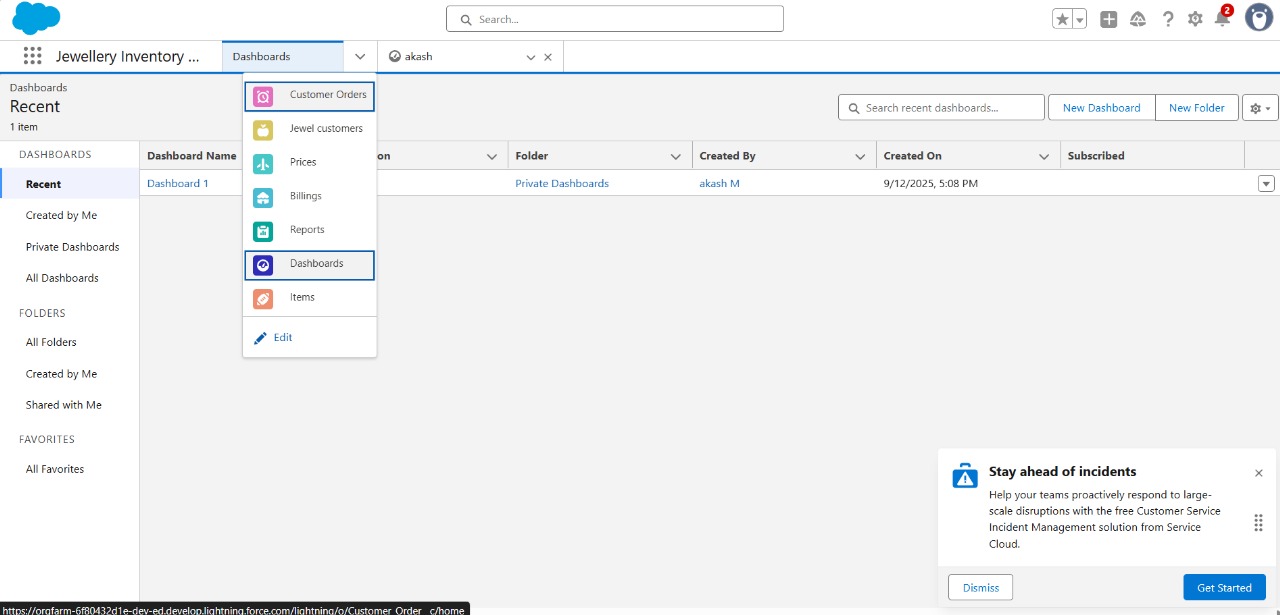


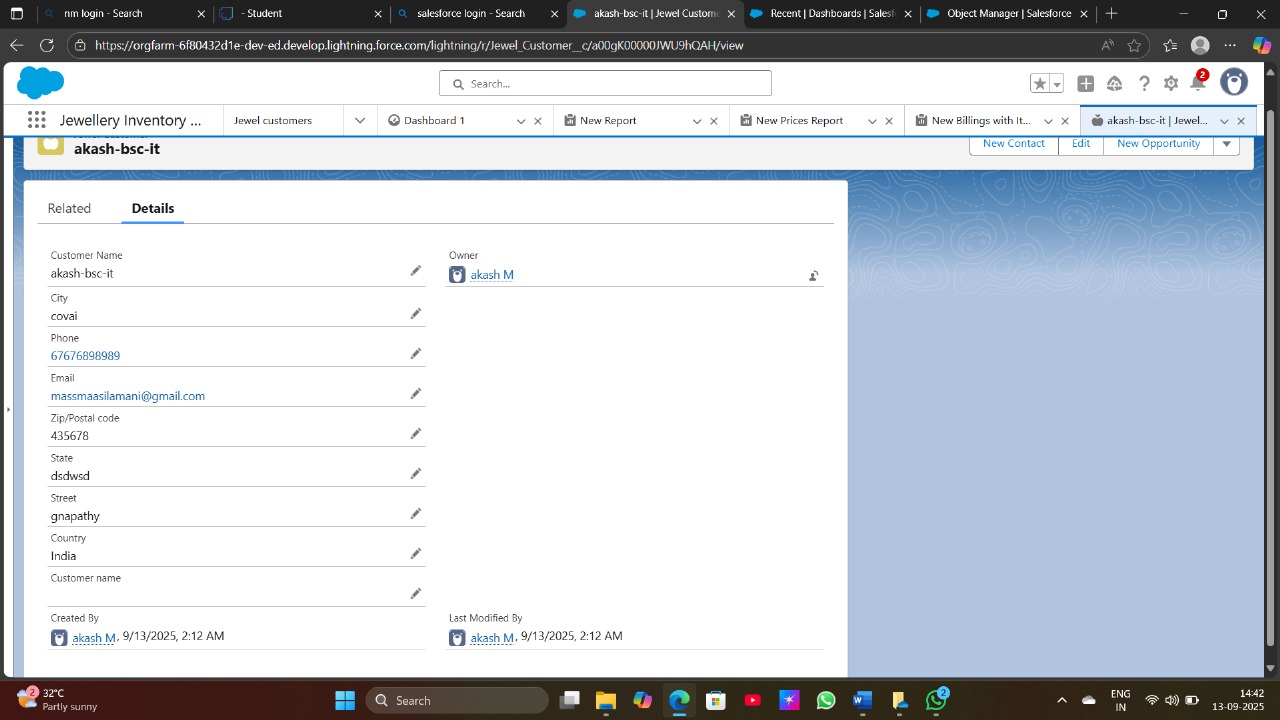
Result

OUTPUT SCREEN

Tab for customer order,Billing,item,price,jewel customer







**Advantage and Disadvantage**

**Advantages**

1. **Customer Relationship Building**
   * Helps track customer preferences (e.g., type of jewelry, design, budget range).
   * Enables personalized recommendations and targeted promotions.
2. **Sales & Marketing Efficiency**
   * Automates follow-ups, reminders for special occasions (weddings, birthdays, anniversaries).
   * Supports loyalty programs and discounts to retain customers.
3. **Data Management**
   * Centralized database for customer information, purchase history, and feedback.
   * Reduces errors compared to manual record-keeping.
4. **Improved Decision-Making**
   * Provides analytics on sales trends, popular designs, and seasonal demands.
   * Helps in inventory planning and marketing strategy.
5. **Enhanced Customer Service**
   * Quick access to purchase history and service requests.
   * Increases trust by offering personalized after-sales services (repairs, cleaning, upgrades).
6. **Competitive Advantage**
   * Builds brand loyalty through customized communication.
   * Differentiates the jewelry store from competitors who rely only on walk-in sales.

**Disadvantage**

1. **High Implementation Cost**
   * Requires investment in software, licensing, customization, and employee training.
2. **Complexity of Use**
   * Staff may need time and training to adapt.
   * Overloaded features can confuse smaller jewelry businesses.
3. **Data Security Risks**
   * Sensitive customer data (personal details, purchase amounts) may be vulnerable to cyberattacks if not well-secured.
4. **Maintenance & Upgrades**
   * Regular updates, technical support, and server maintenance can be costly and time-consuming.
5. **Dependency on Technology**
   * If the system crashes or faces downtime, sales and customer interaction may get disrupted.
6. **Resistance from Employees**
   * Traditional jewelers may resist shifting from manual/relationship-based sales to technology-driven processes.

**Conculsion**

The implementation of a CRM application in jewelry management plays a vital role in enhancing customer relationships, streamlining business operations, and driving sales growth. By maintaining detailed customer data and purchase history, jewelers can offer personalized services, targeted promotions, and improved after-sales support, which helps in building long-term customer loyalty. Although challenges such as high costs, data security risks, and staff adaptation exist, the overall benefits of improved efficiency, better decision-making, and competitive advantage outweigh the drawbacks. Therefore, a well-implemented CRM system can transform jewelry businesses into more customer-centric, data-driven, and profitable enterprises.

**Appendix**

Source code: Provided in Apex Classes And Trigger

Test.apxt

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;

        }

    }

}

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);

    }

}