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

COLLEGE NAME: VET institute of arts and sciences (co-education) college

COLLEGE CODE:636301

TEAM ID : nm2025TmiD25791

TEAM MEMBERS:

Team Leadername : D.HemaLatha

Email : [Hemalathad23csa@vetias.ac.in](mailto:Hemalathad23csa@vetias.ac.in)

Team Members : G.Surya

Email : [Suryag23csa@vetias.ac.in](mailto:Suryag23csa@vetias.ac.in)

Team Members : S.Thenmozhi

Email : [Thenmozhis23csa@vetias.ac.in](mailto:Thenmozhis23csa@vetias.ac.in)

Team Members : S.Deepak

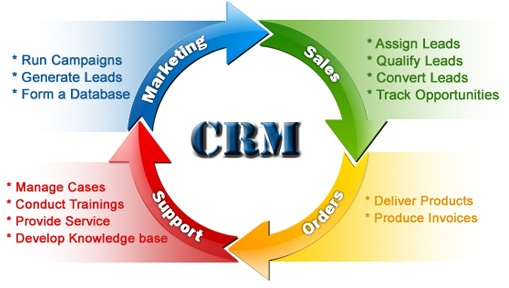
Email : [Deepaks23csa@vetias.ac.in](mailto:Deepaks23csa@vetias.ac.in)

1. **INTRODUCTION**
   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. what we learn**

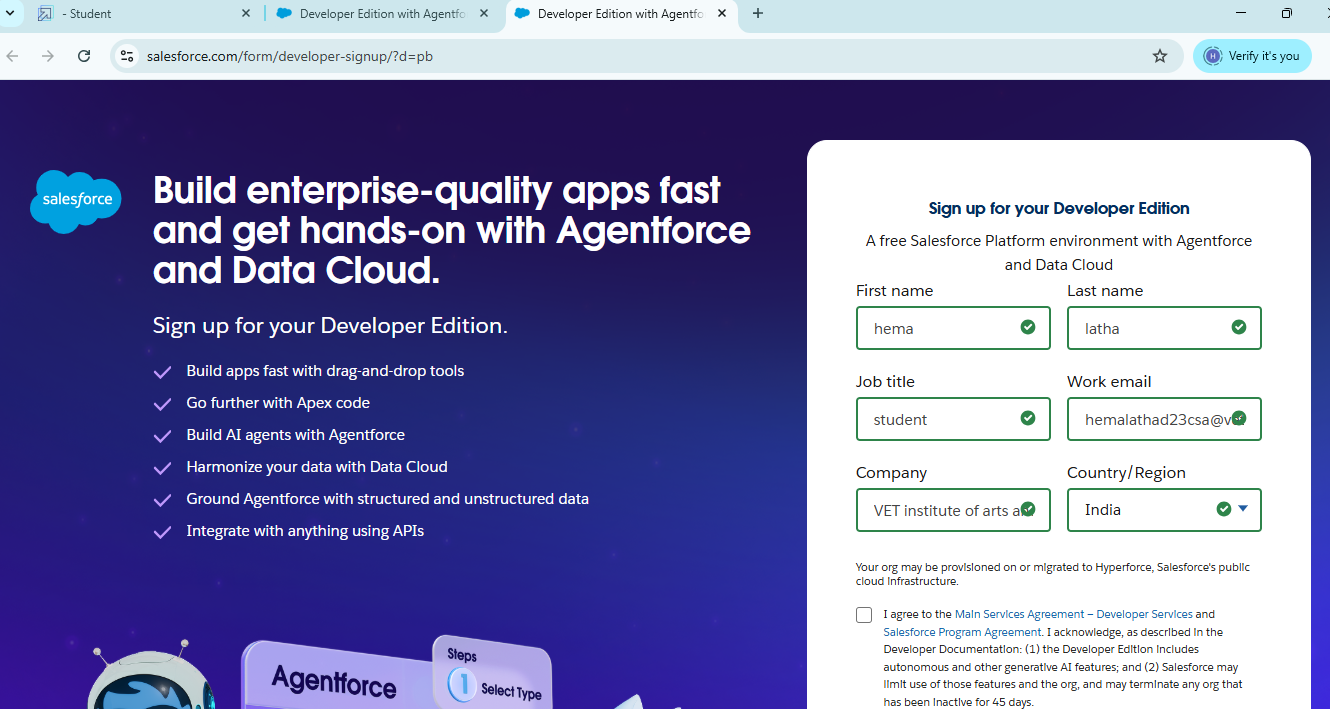
1. Real Time Salesforce Project
2. Data Modelling
3. Creating an Application
4. User Interface Customization
5. Object & Relationship in Salesforce
6. Formula fields and Validation rules.
7. Field Dependencies
8. Record Types
9. Cross object formula fields.
10. Conditional formatting.
11. Flows
12. Email alerts and email templates
13. Reports & Dashboards



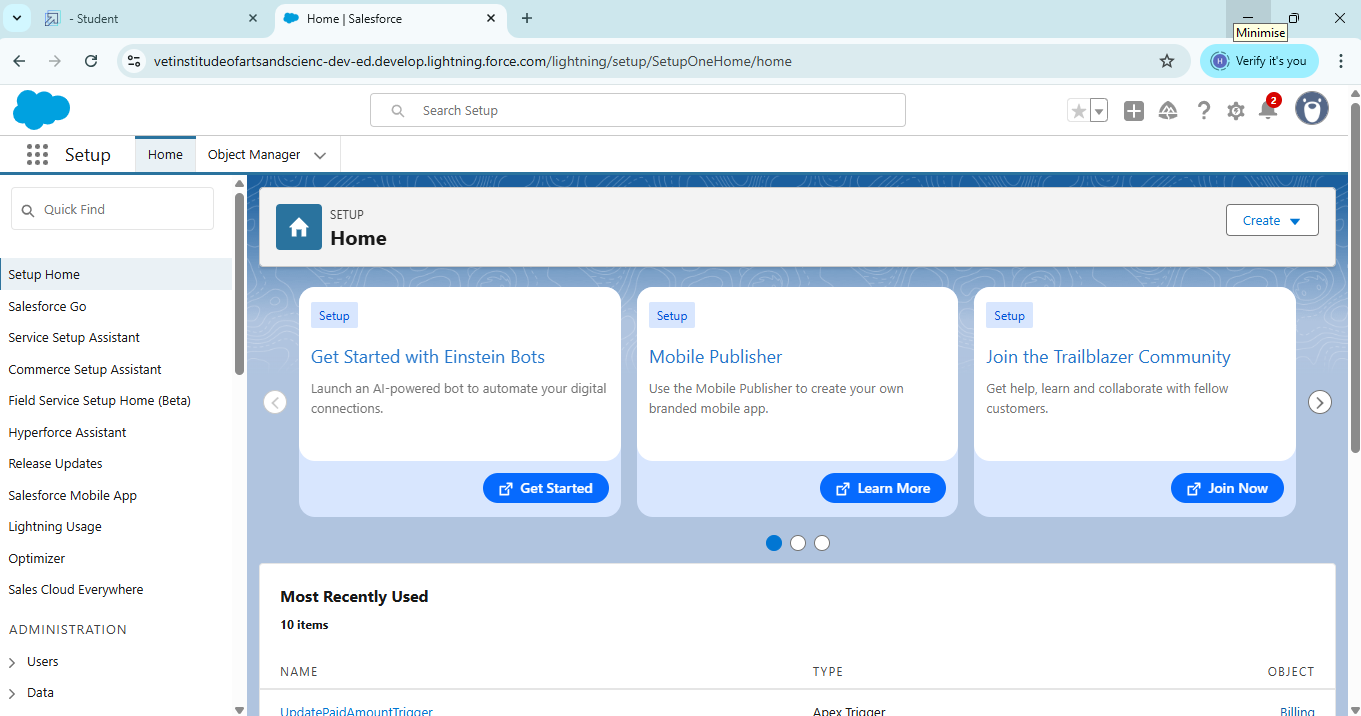
**DEVELOPER PHASE**

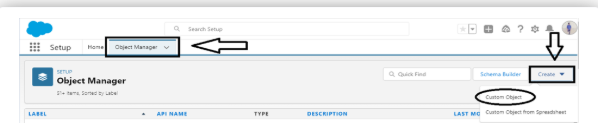
Creating developer account:

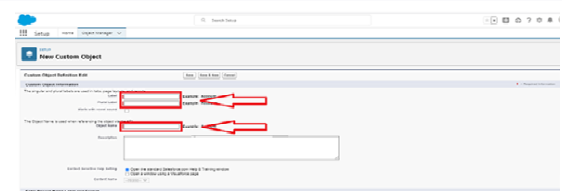
By using this URL: https://www.salesforce.com/form/developer-signup/?d=pb



* + 1. **Create object** : jewel customer , customer order , price , billings

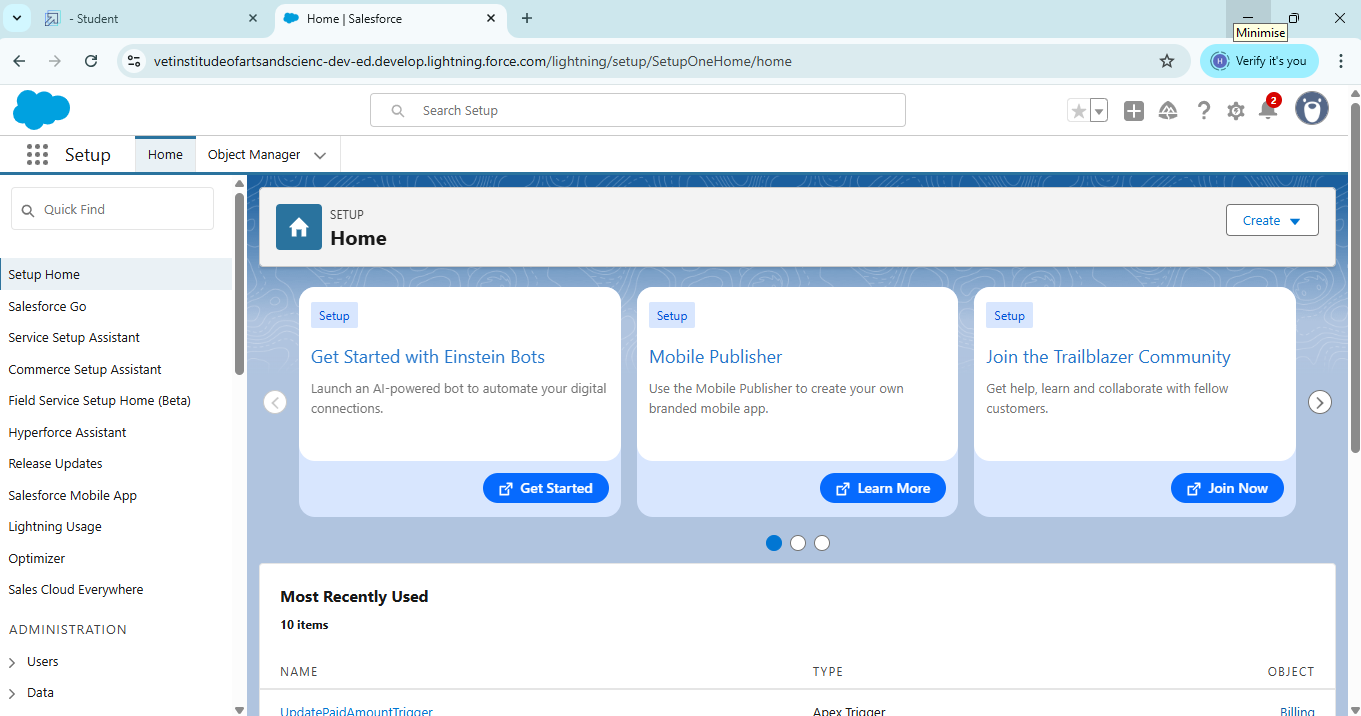


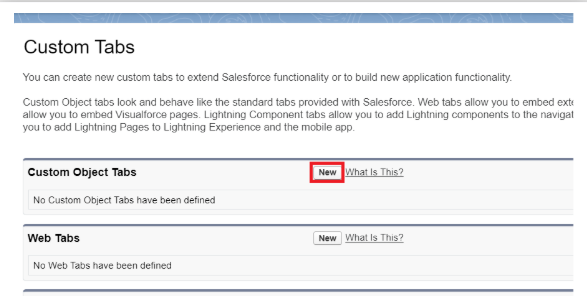


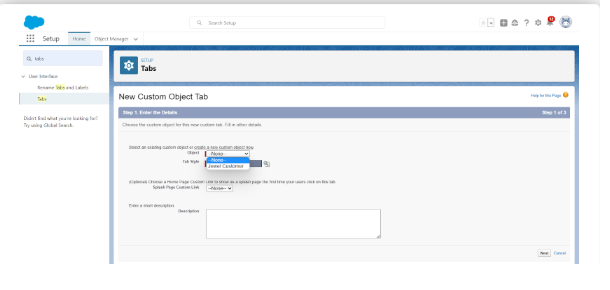




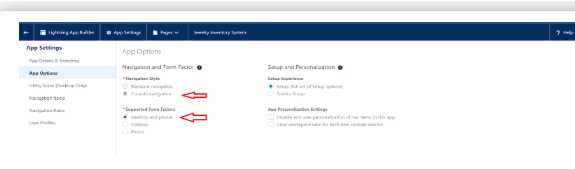
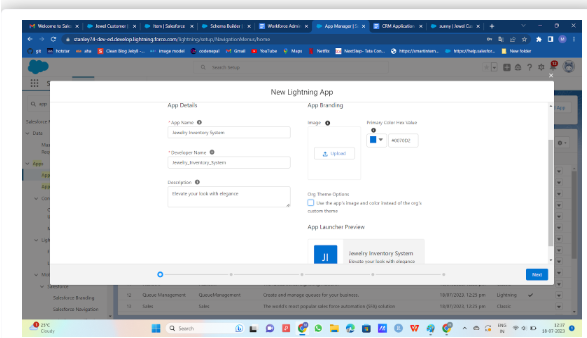
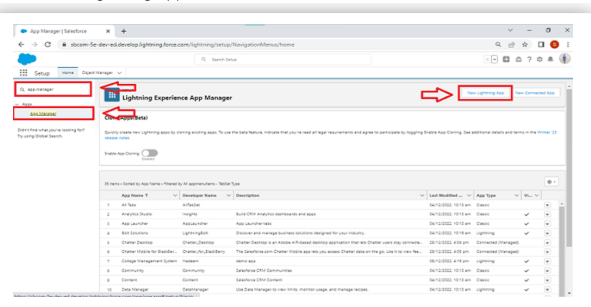
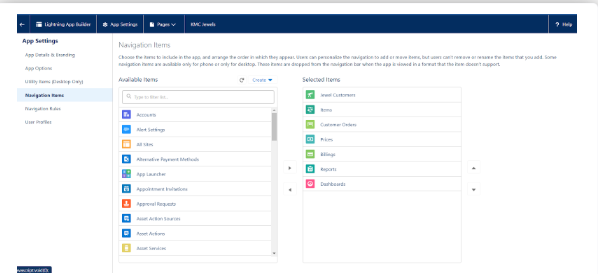
* Click save.
  + 1. **CREATE A CUSTOM TAB OBJECT :** jewel customer , customer order , price , billings

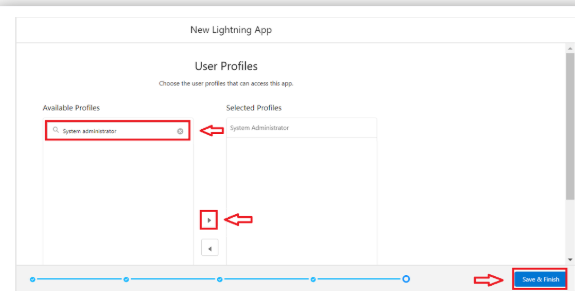






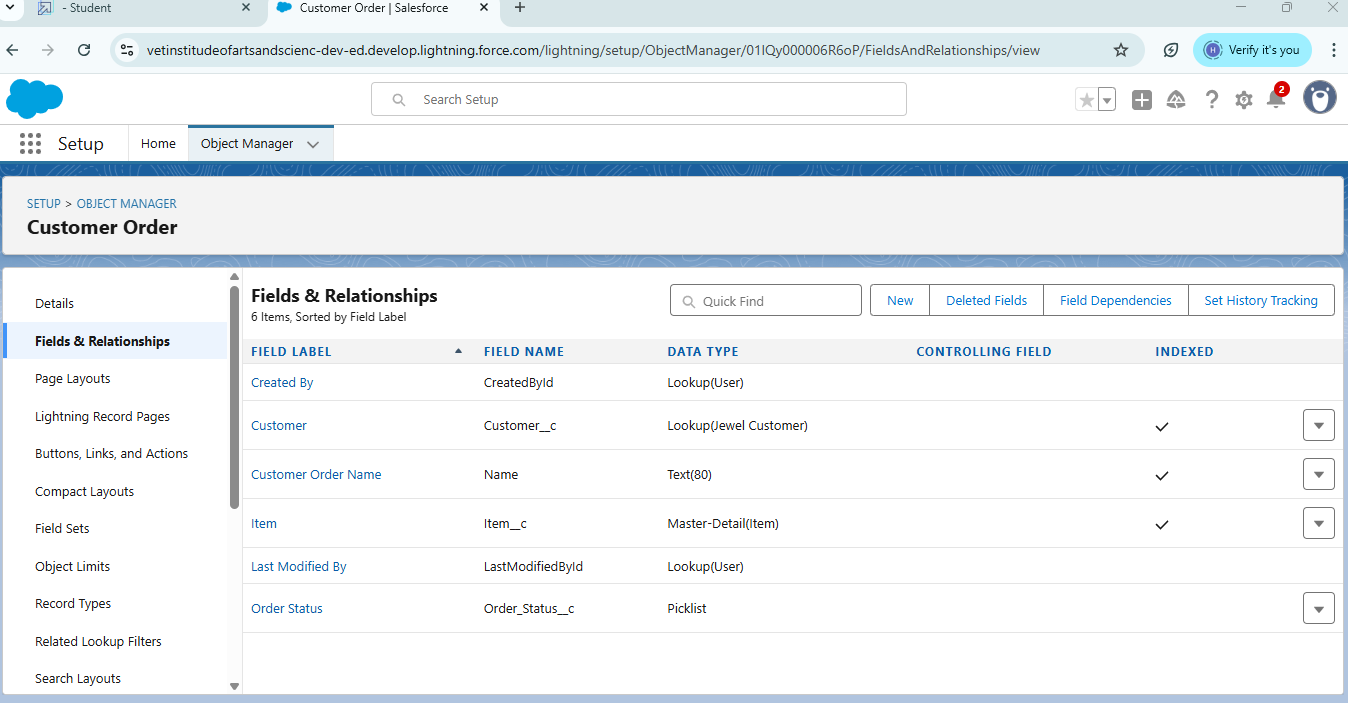
* Click save.

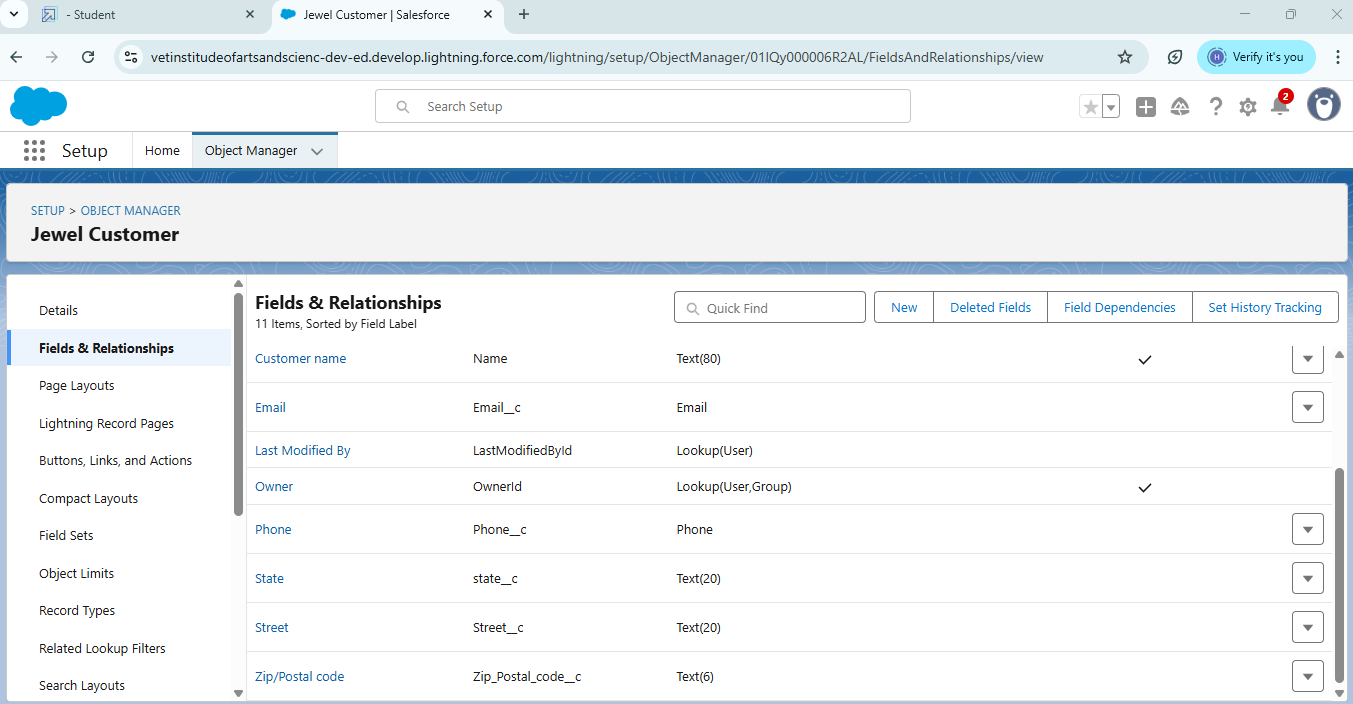
**Creating a lightning app: **

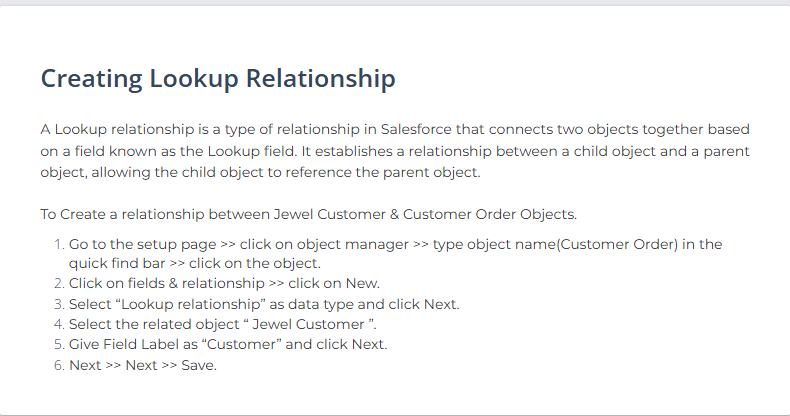
****

* click save & finish

**TO C REATE A FIELDS IN A CUSTOMER ORDER AND JEWEL CUSTOMER:**

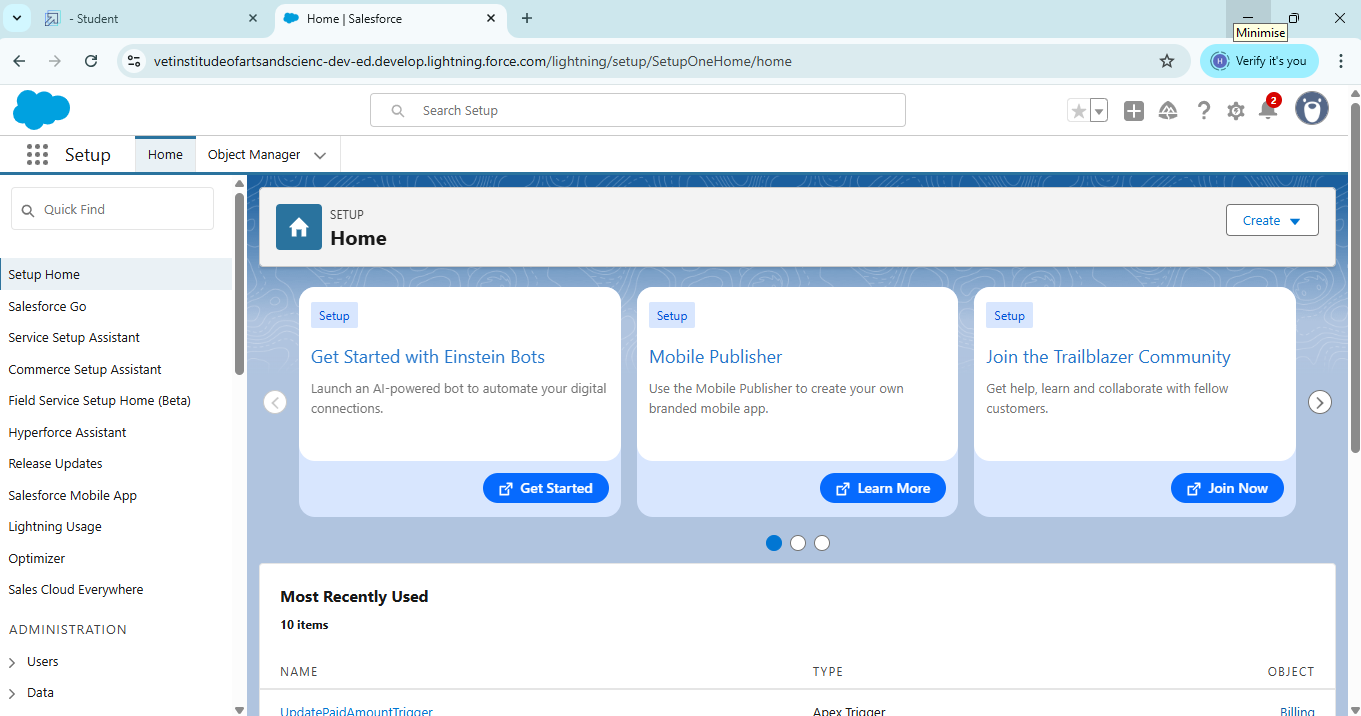


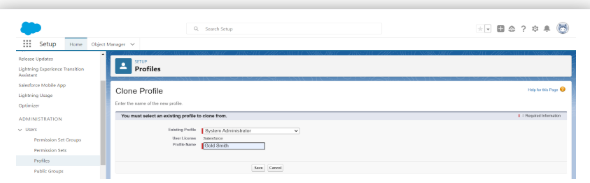


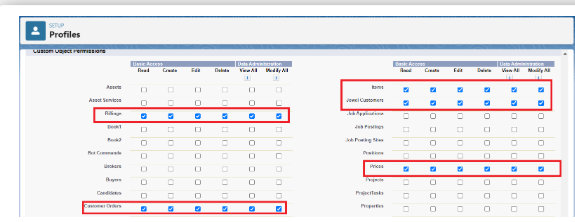


* Follow this above steps to create fields on jewel customer and customer order

**Create a profiles:**gold smith,worker

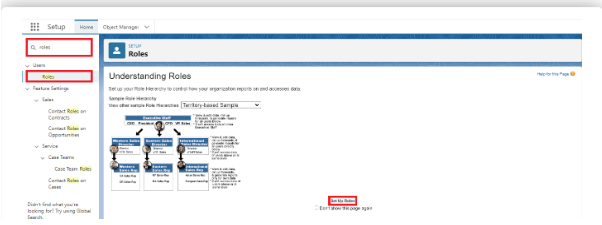




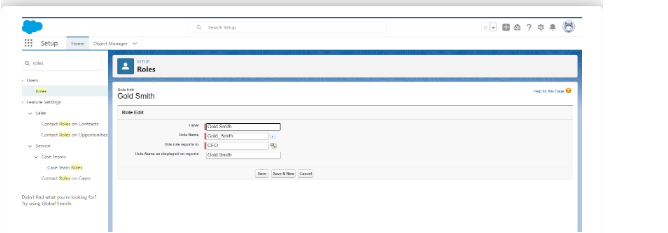


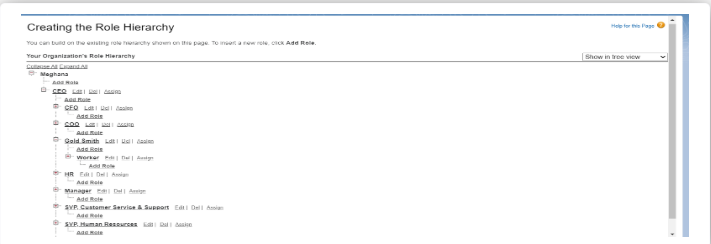
* Click save

**CREATE A ROLES :** GOLD SMITH,WORKER

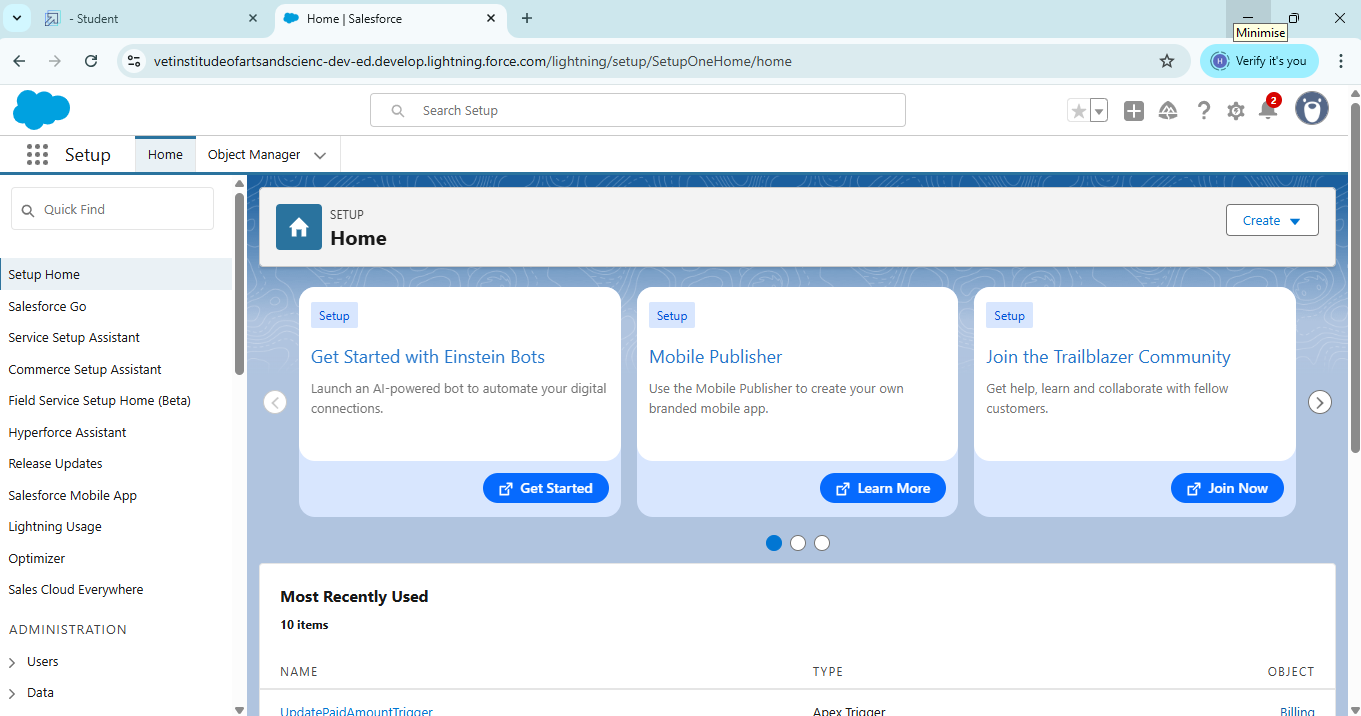


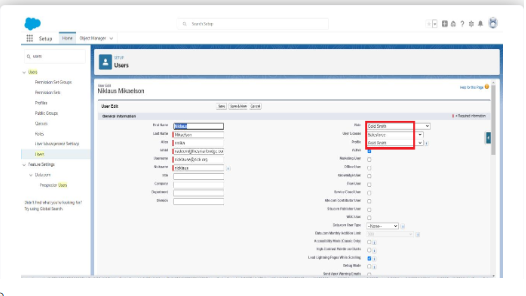






**CREATE A USERS : GOLD SMITH , WORKER**

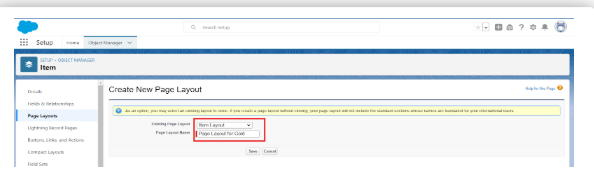
****

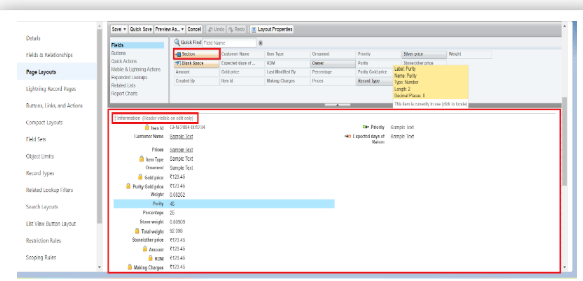
****

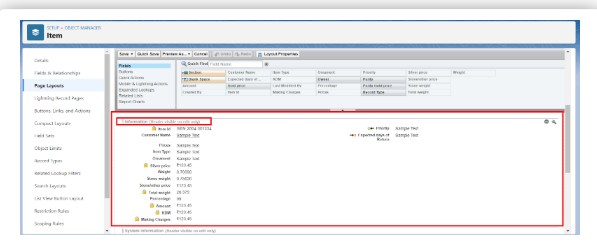
* Click save

**CREATE A PAGELAYOUTS:**SILVER PAGELAYOUT, GOLD PAGELAYOUTS



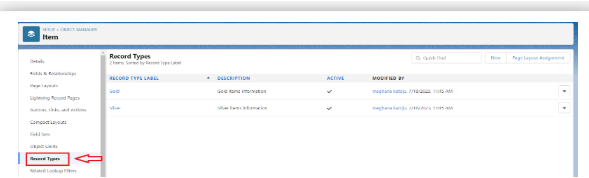


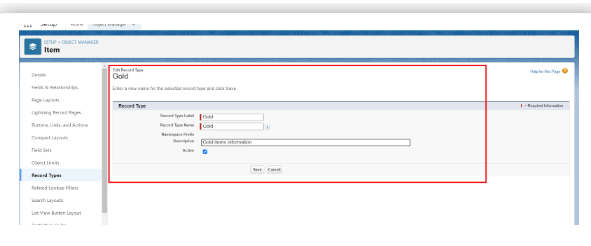




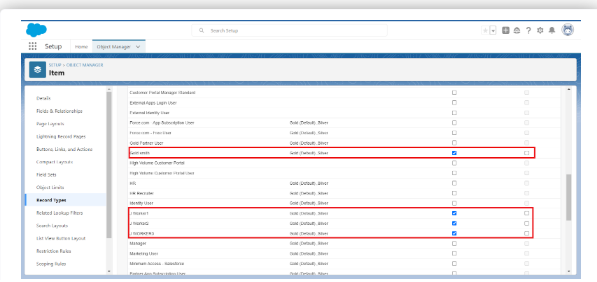
* Click save.

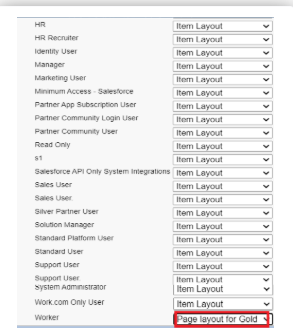
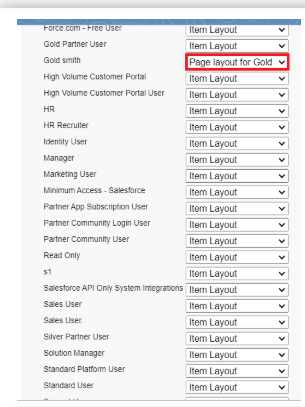
**CREATE A RECORD TYPE:**GOLD , SILVER



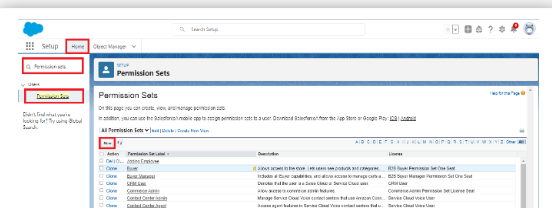


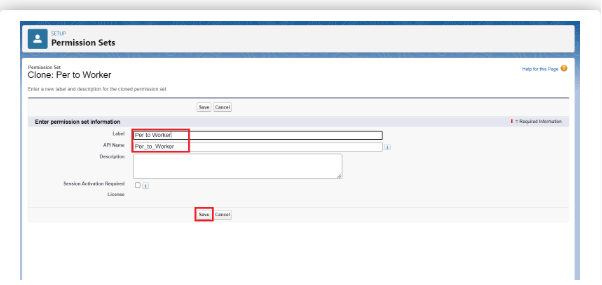


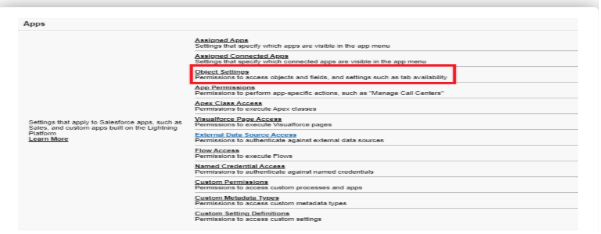


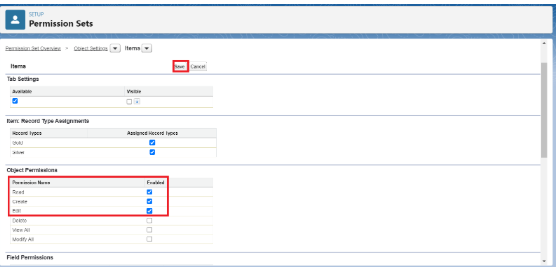


* Click save&new.

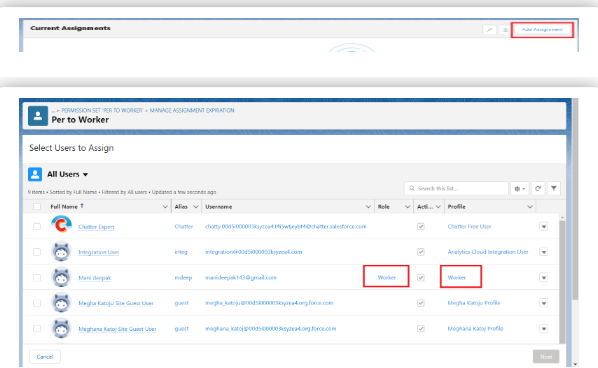


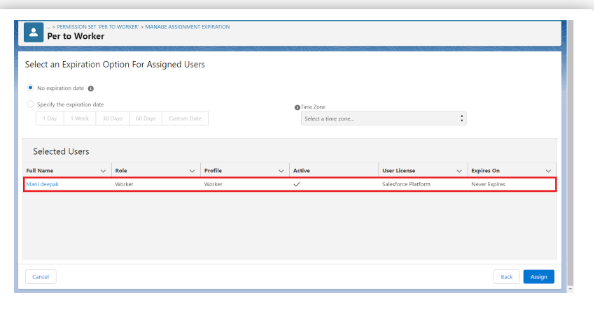




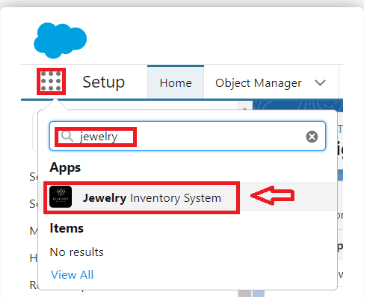


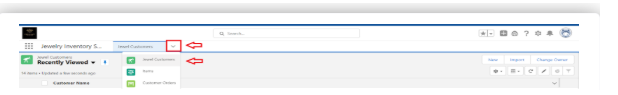
* Click save.



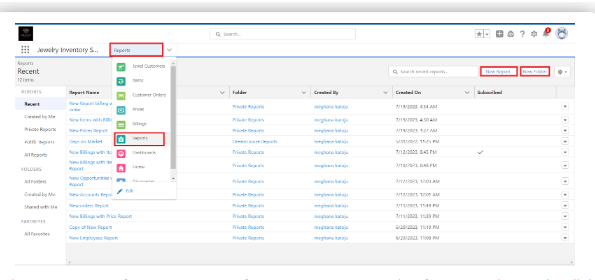


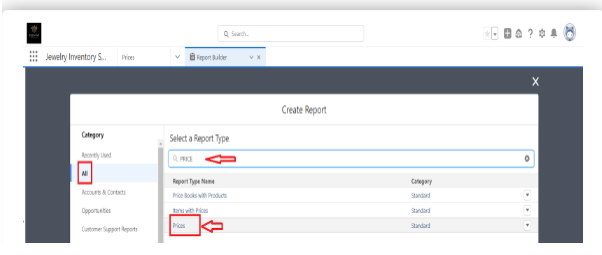
**USER ADOPTION:**CREATE,VIEW,DELETE A RECORD FOR JEWEL CUSTOMER

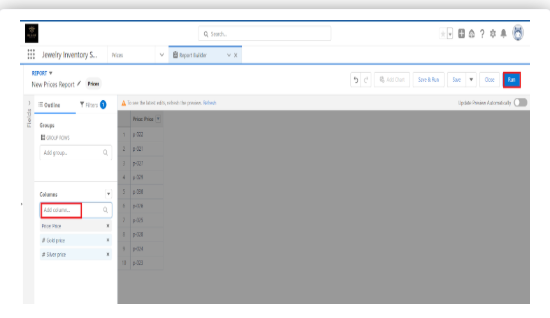


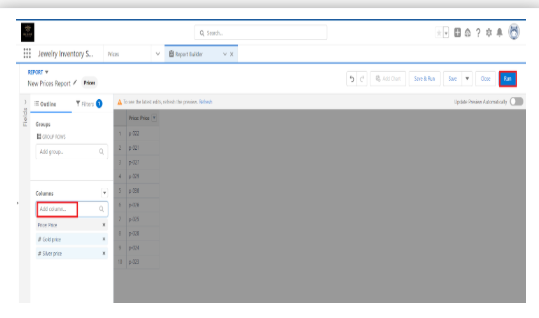


**CREATE A REPORTS:**

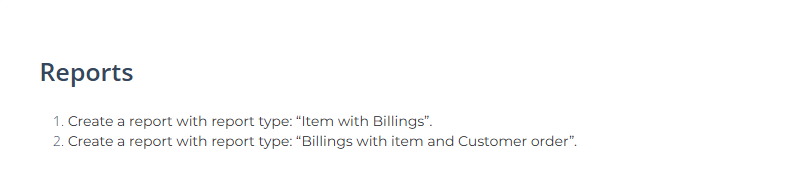
****

****

****

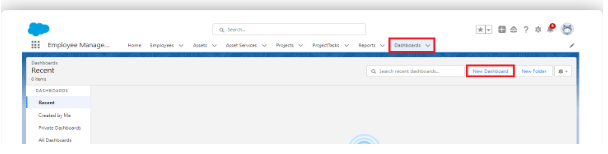
****

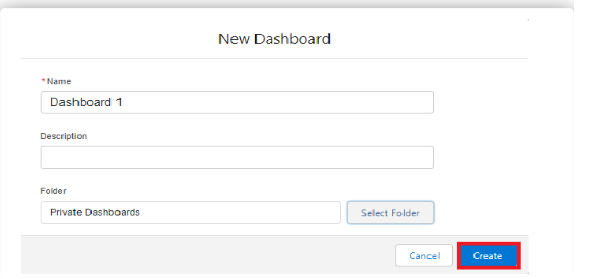
* Click save & run.

****

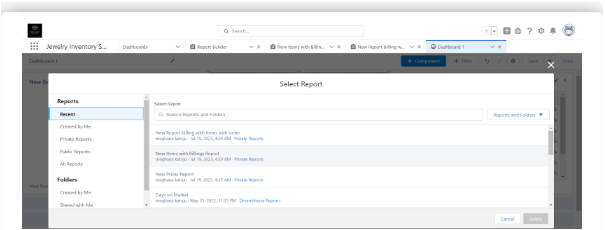
* Also create a report for those types.

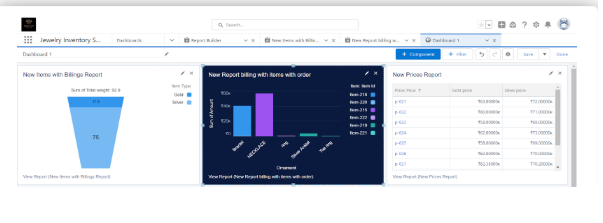
**CREATE A DASHBOARD:**

****

****

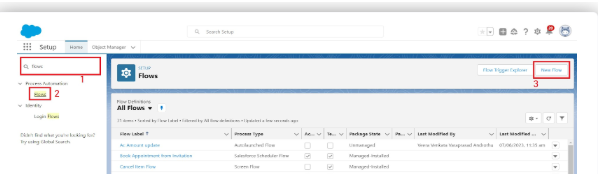
****

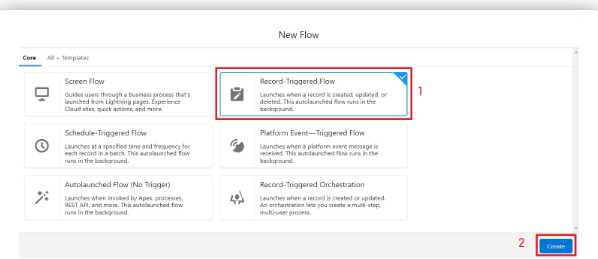
****

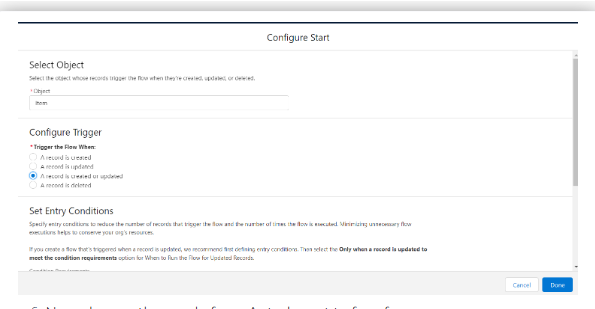
****

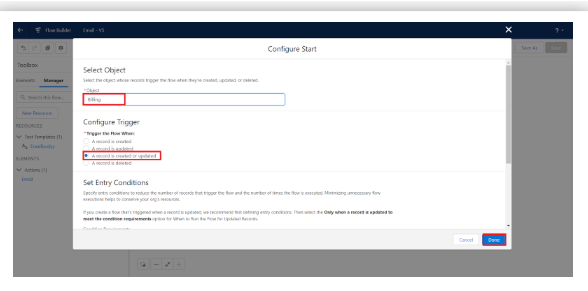
* Click save and done.

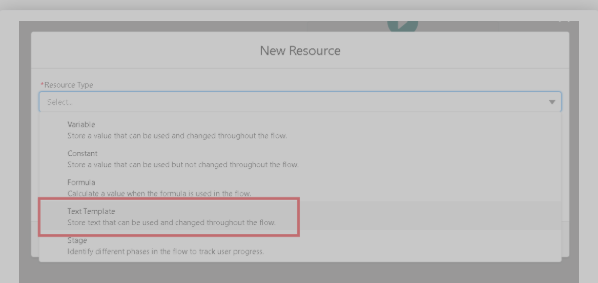
**CREATE A FLOW:**

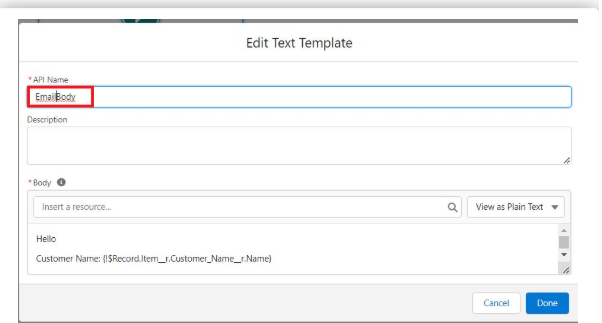
****

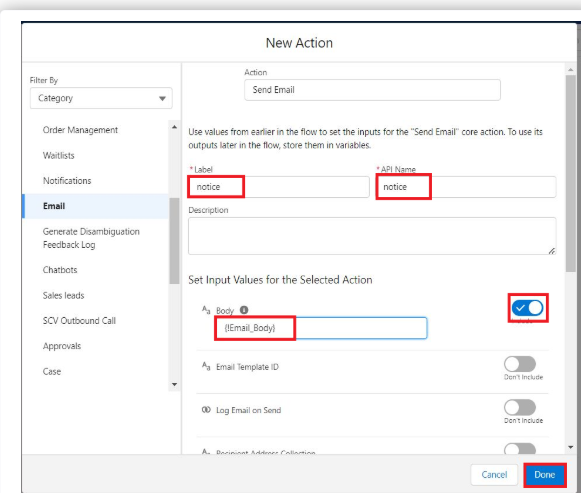
****

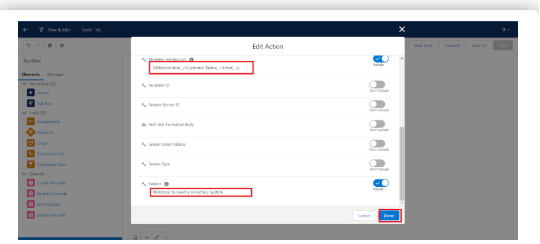
****

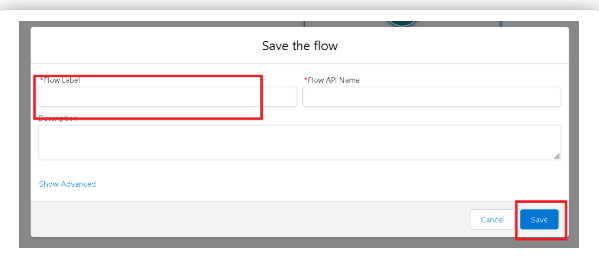
****

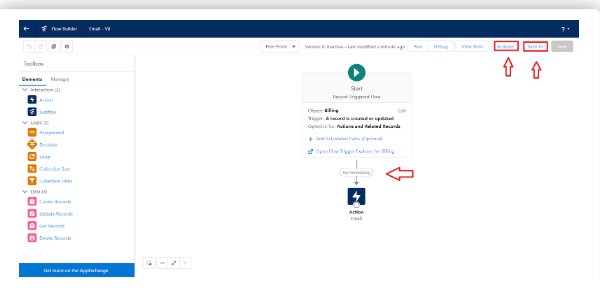
****

****

****

****

****

****

* Click save.

CONCLUSION

The CRM Application for Jewel Management helps jewelry businesses manage customer data, sales, and interactions efficiently.It improves customer satisfaction, streamlines operations, and supports better decision-making.The application is secure, scalable, and user-friendly, making it easy to adapt to business needs.Overall, it enhances growth, customer loyalty, and business performance in the jewelry sector.

APPENDIX

* SOURCE CODE:PROVIDED IN TRIGGER

**Create a Trigger Handler class:**

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;

        }

    }

}

**Create a trigger:**

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);

    }

}