### PROJECT TITLE : CRM Application for Jewel Management - (Developer)

**College Name: Kamalam college of arts and science**

**College Code:bru5s**

**TEAM ID:  NM2025TMID26790**

**TEAM MEMBERS:4**

**Team LeaderName: S. BOOMIKA**

**Email: boomika1146@gmail.com**

**Team Member:  AJAY G**

**Email:** [**ajaysha12345sha@gmail.com**](mailto:ajaysha12345sha@gmail.com)

**Team Member:  DAVID T**

**Email: tdavidsam57@gmail.com**

**Team Member:  RANVINA P**

**Email: ranvinaranvina5@gmail.com**

**1.INTRODUCTION**

**1.1 Project Overview**

The goal of this CRM system is to unify customer interactions, streamline inventory and sales operations, and enhance marketing efforts for jewelry businesses. It’s built on Salesforce, leveraging its robust automation and customization capabilities to meet the unique needs of jewelers.

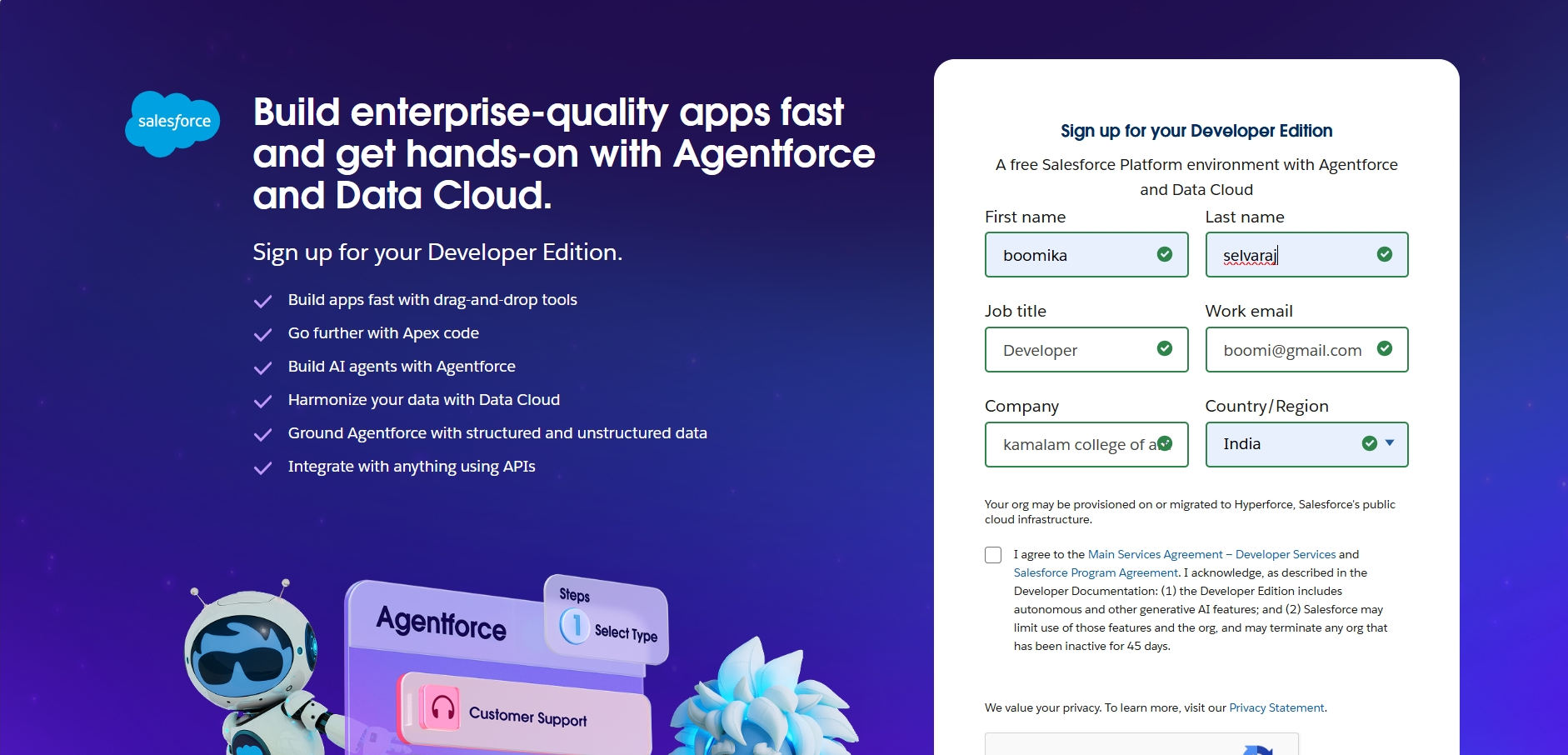
**1.2 Purpose**

To help jewelry businesses manage their customers, sales, and inventory more efficiently using Salesforce. It keeps track of jewel items, customer interactions, and automates marketing and follow-ups—all in one place.

**DEVELOPMENT PHASE**

**Creating Developer Account:**

**By using this URL -** [**https://www.salesforce.com/form/developer-signup/?d=pb**](https://www.salesforce.com/form/developer-signup/?d=pb)

****

### Account Activation

### https://lh7-rt.googleusercontent.com/docsz/AD_4nXfcmLxkJh0lEhKVPYYRVfVYbUWlCqg5Z8-KZexgqwqJZGm7EzMpo6WpdGSlwcV4tDeVUrZftM8Gx-_7jLcPFMhZBomdoxJJI7Qzh_lkddYTkDx0yoHipXv-tmGTsQo79T0y8KdAQg?key=dfLmaqUsnVoHOKcSZi4NZA

### Create Jewel Customer, Item, Customer Order,Price,Billing Object

### 

### 

### 

### 

### 

### Creating a Custom Tab

### https://lh7-rt.googleusercontent.com/docsz/AD_4nXdKSTxRvfrdilRY_6vbkWhaKHvH9Qu4xd4cQilOvW4miuTWOL9qUaANDgK9QqBju3mvire3v2N52W9ITjGXprrYHZ9nkv_FMNLa5pHpMhY_EOFerJwyVogCXt-9ouSTXFJVEgKuMw?key=dfLmaqUsnVoHOKcSZi4NZA

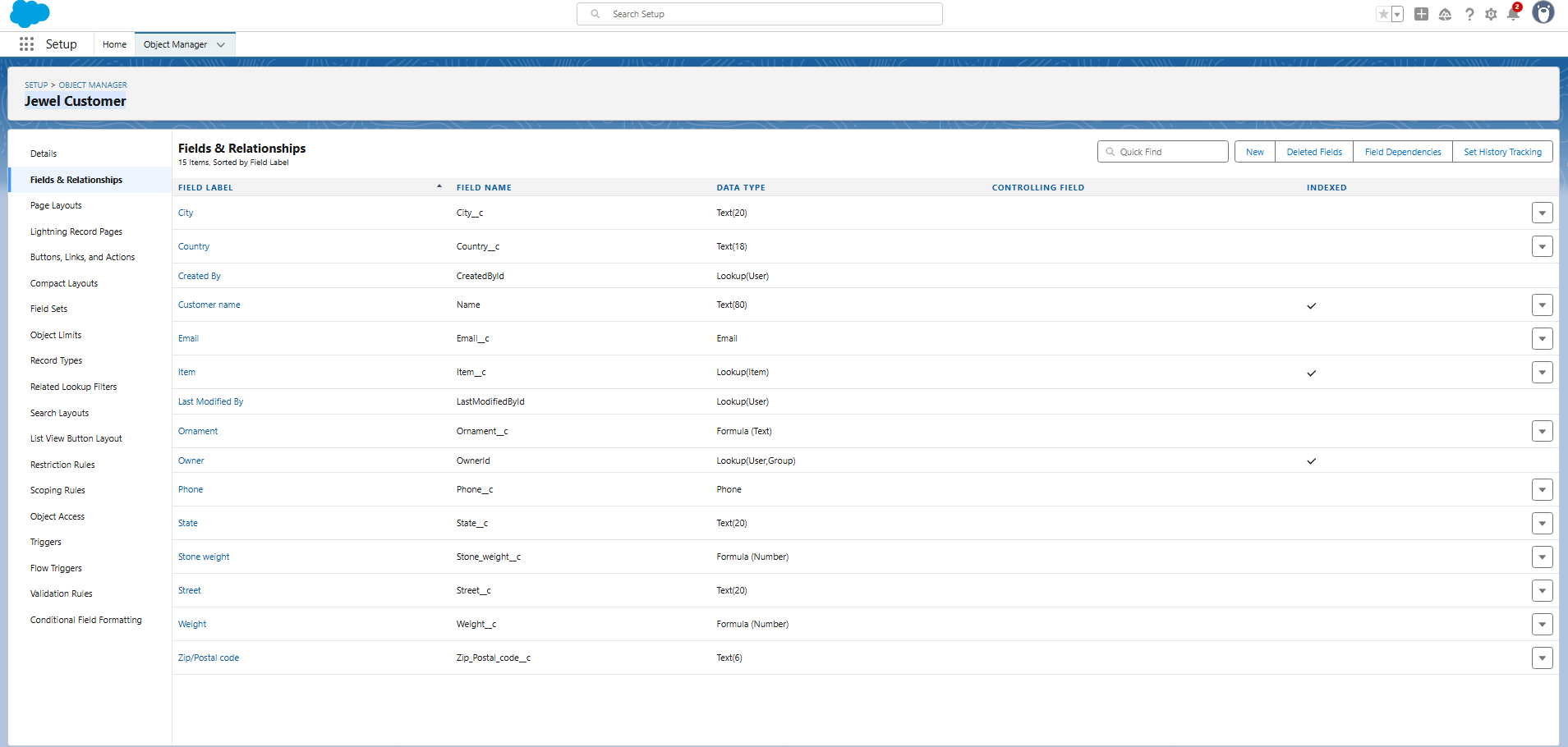
### 

### Create a Lightning App

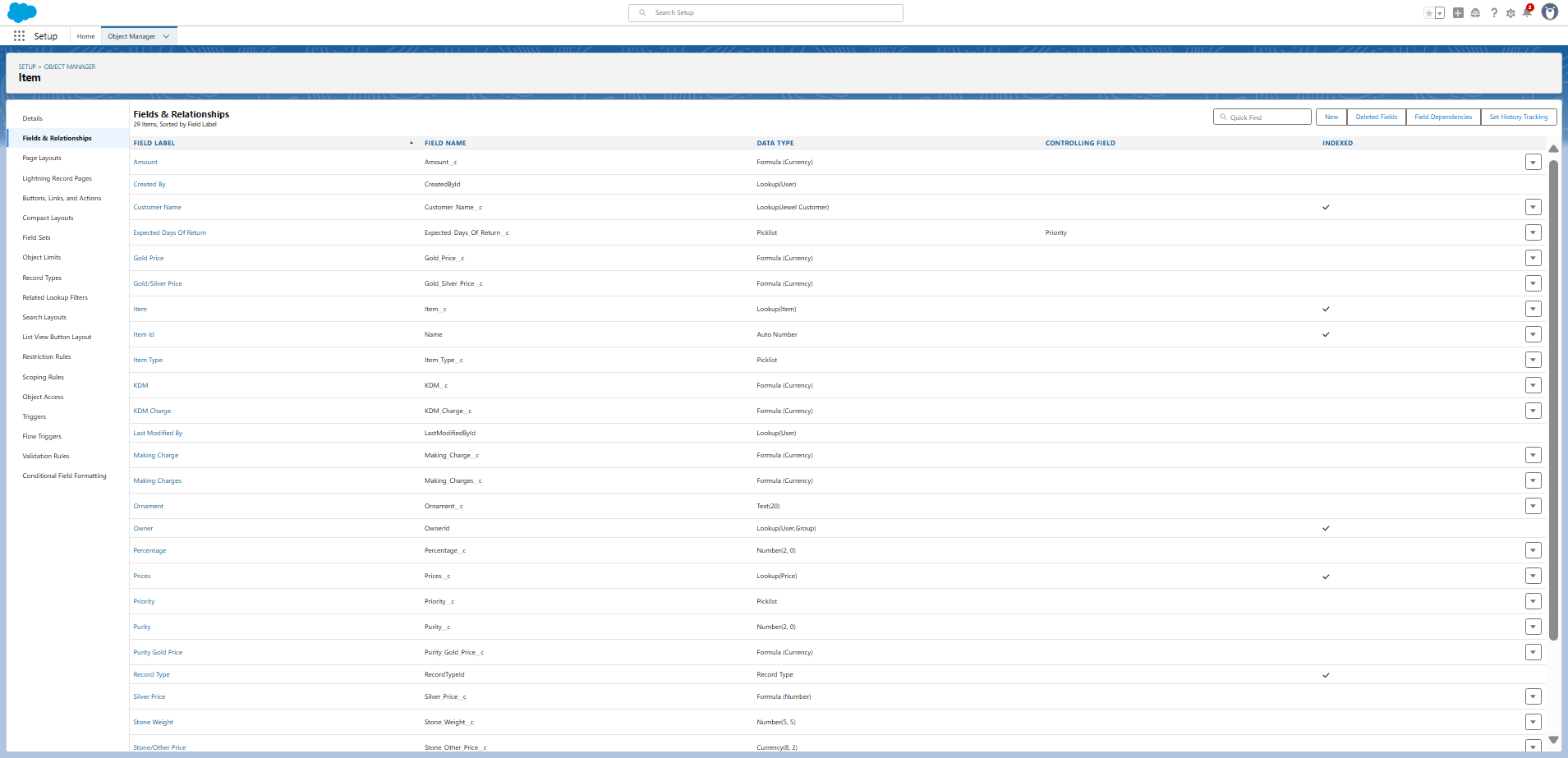
****

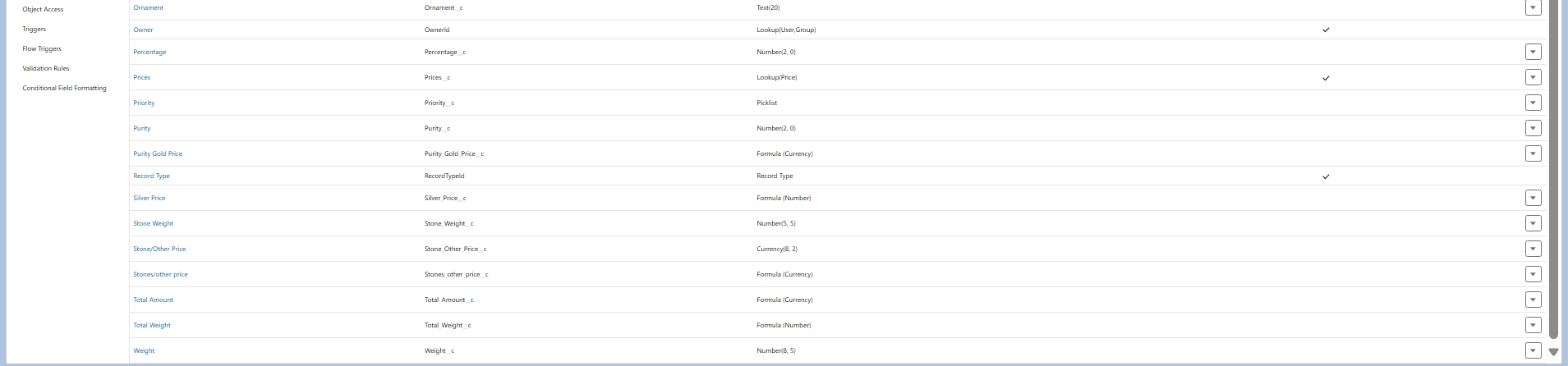
### Creating Fields and Relationships

# Jewel Customer

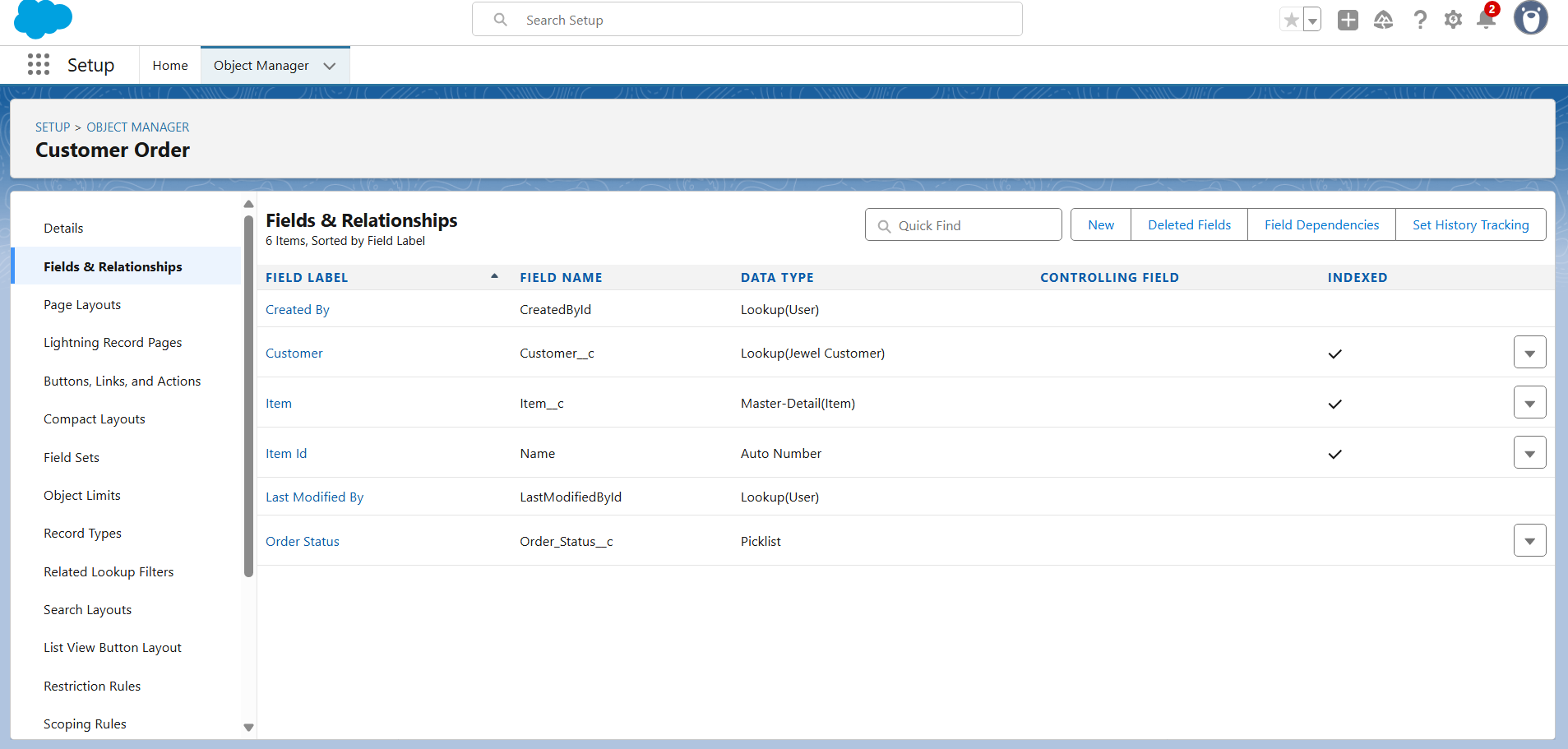


# Item

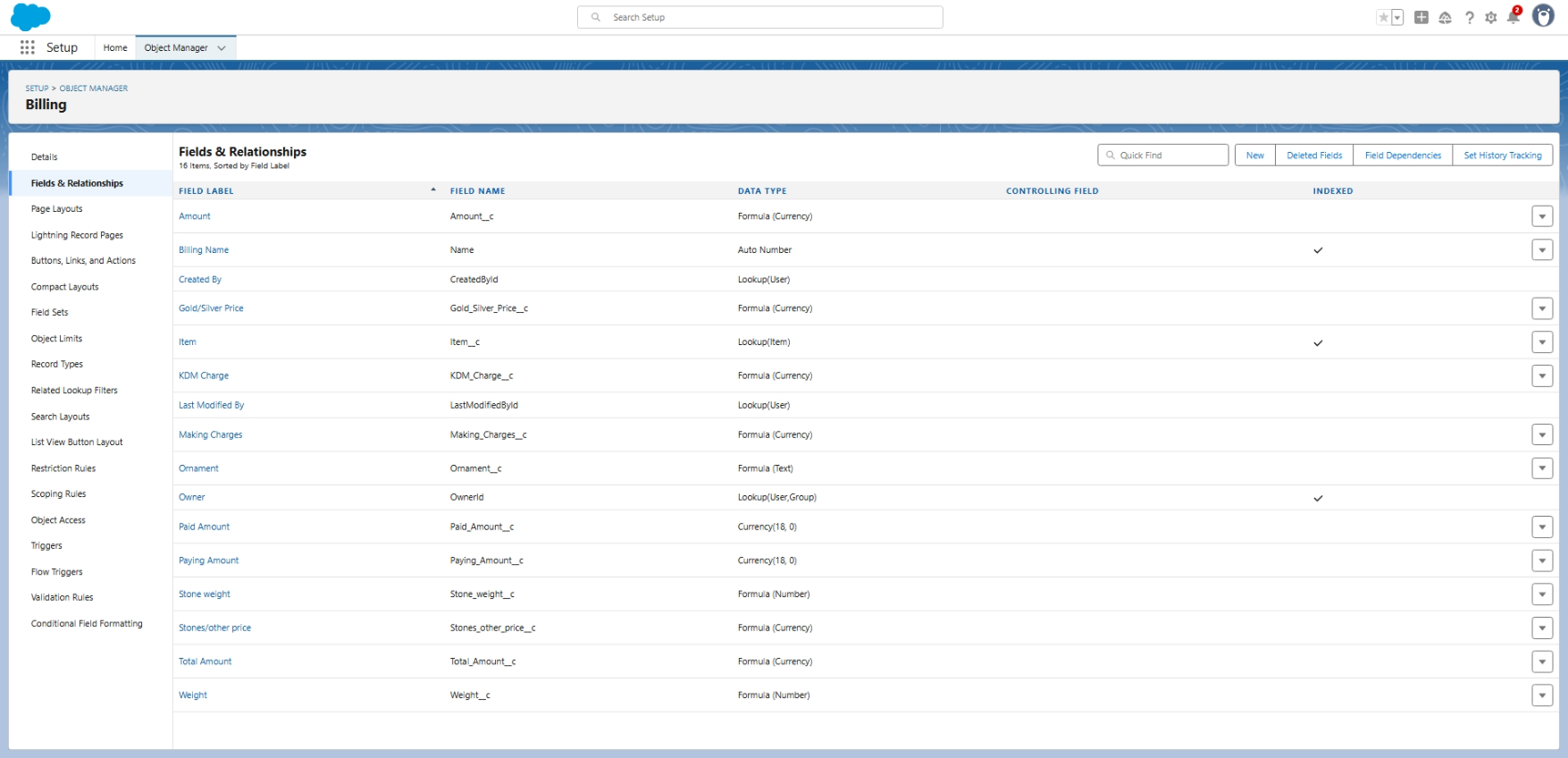


****

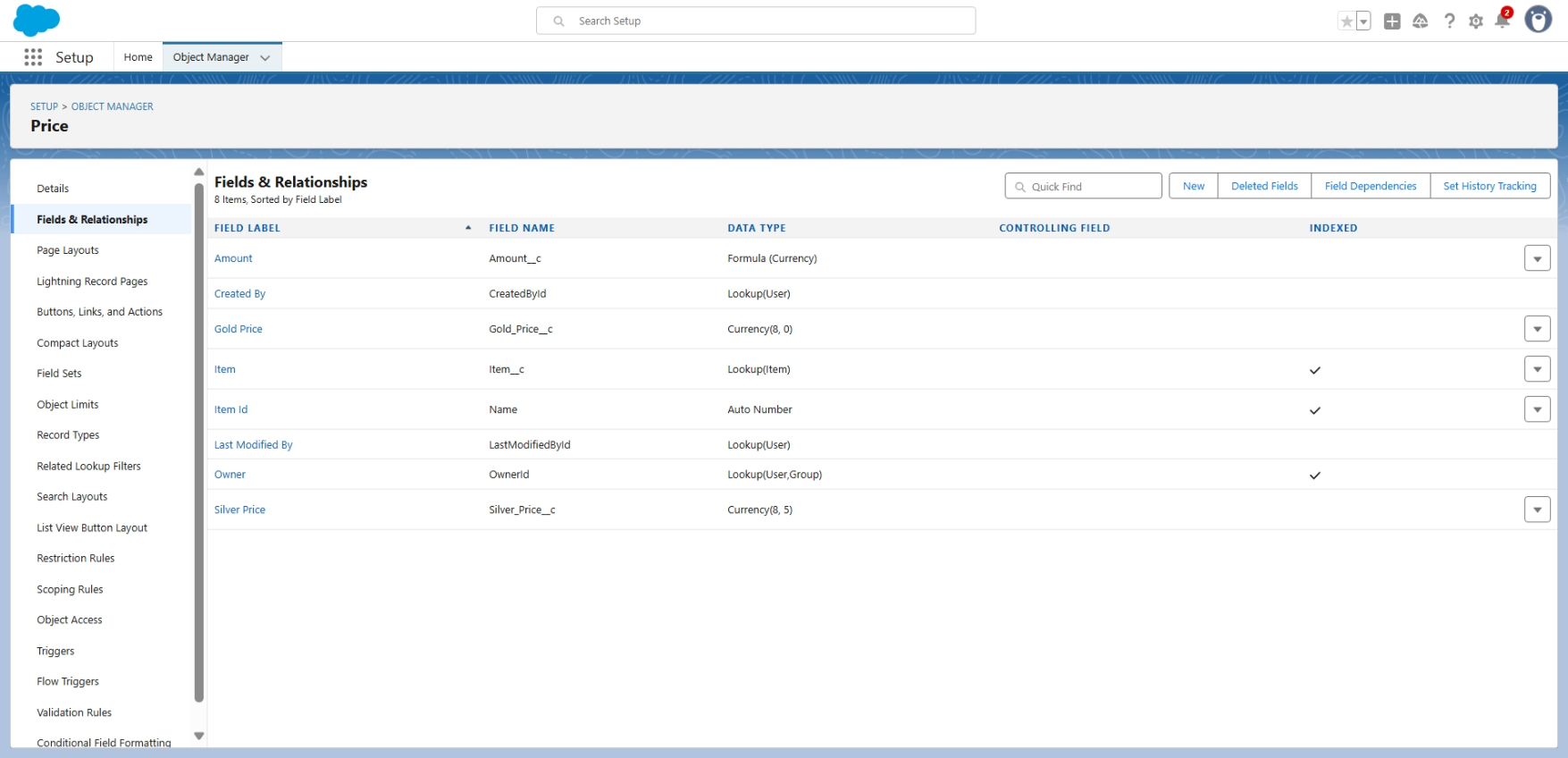
# Customer Order



# Billing



# Price



### Profiles

### Gold Smith Profile

### https://lh7-rt.googleusercontent.com/docsz/AD_4nXc4eb58asMjIPJy5ObneGqPbJGsoIqlCdXErpW43vMJg4WHiHm6gIUB0UJeW-Zy3YsRUf6AuogCRU5ZJlEYlc4K_jgqxaRVYffA74zGdh4SoSQPN1zjFQRASObVeDr6gVgsY6M-0Q?key=dfLmaqUsnVoHOKcSZi4NZA

### 

### Worker Profile

### 

### 

### Roles

### Creating Gold Smith Role

### 

### 

### https://lh7-rt.googleusercontent.com/docsz/AD_4nXf8S6Y9zf4cQOlHekFSFvlGZ_oRTeqdd7eLHmt2bUGY75ur4Px9zI7lrKrzHf8lQiaZEkQBCXgST8RFaszuEz8flWOLoOBhJTICtAJ0zt9STJeCHYCWVQt7tPPnzRBMWBnzfrl-?key=dfLmaqUsnVoHOKcSZi4NZA

### Create one more role as Worker which reports to Gold Smith

### 

### 

### Users

### Creating two users

### 

### 

### Page layouts

### To Create a Gold Page layout

### 

### To Create a Silver Page layout

### 

### Record Types

### To create a Record Type for gold and silver

### 

### 

### Permission sets

### Creating permission set

### 

### 

### 

### 

### 

### 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 the trigger

**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);

    }

}

### 

### User Adoption

### Create a Record (Jewel Customer)

### 

### 

### View a Record(Jewel Customer)

### 

### Delete a Record(Jewel Customer)

### 

### Reports

### Create Report

### 

### 

### Dashboards

### Create Dashboard

### 

### Flows

### Create a Flow

### 

### RESULTS

### Output Screenshots

## Objects for Jewel Customer,Item,Customer Order,Price,Billing

### 

## Tabs for Jewel Customer,Item,Customer Order,Price,Billing

### 

### Reports and dashboard

### 

### 

### 

### Flow runs

### 

### CONCLUSION

### The implementation of a Salesforce-based CRM system for jewelry management has successfully transformed traditional operations into a streamlined, data-driven ecosystem. By integrating inventory tracking, customer relationship management, and automated marketing workflows, the project has delivered measurable improvements in efficiency, customer engagement, and sales performance.

### APPENDIX

### ● Source Code: Provided in Apex Classes and Triggers

### UpdatePaidAmountTriggerHandler,apxc

**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;

        }

    }

}

### UpdatePaidAmountTrigger.apxt

**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);

    }

}