

CRM APPLICATION FOR JEWEL MANAGEMENT

SK

COLLEGE NAME : SREE NARAYANA GURU COLLEGE COIMBATORE

COLLEGE CODE : BRU36

TEAM ID : NM2025TMID26369

TEAM MEMBERS : 4

TEAM LEADER NAME : Swarag M

E-MAIL : Swaragmuralidharan@gmail.Com

TEAMMEMBER 1 : Adithyan K

E-MAIL : Adhithyanadhian2002@gmail.Com

TEAM MEMBER 2: Anirudh R

E-MAIL : manjuraju954@gmail.com

TEAM MEMBER 3 : Nikitha P S

E-MAIL : psnikitha01@gmail.com

TEAM MEMBER 4 : Rohini A

E-MAIL : rohiniarohinia03@gmail.com

1.INTRODUCTION

1.1 PROJECTOVERVIEW

- ❖ Jewellery management in a CRM (Customer Relationship Management) system is specialized software designed for jewelry businesses to centralize customer data, manage interactions, and optimize sales and inventory.
- ❖ It helps jewelers understand customer preferences and purchase history to provide personalized service, automate follow-ups, and track sales and orders efficiently. By analyzing data, a jewelry CRM can also improve inventory management, suggest targeted marketing campaigns, and ultimately foster customer loyalty and drive business growth.

SK



1.2 PURPOSE

❖ The primary purpose of jewellery management in a CRM is to enhance customer satisfaction and loyalty by providing personalized service, tracking purchase history and preferences, and facilitating targeted marketing and engagement.

DEVELOPMENT PHASE

CREATING DEVELOPER ACCOUNT :

BY USING THIS URL : <https://www.salesforce.com/form/developer-signup/>

The image shows two screenshots from a web browser. The top screenshot is the Salesforce Developer Edition signup page. It features a dark blue background with white text. On the left, it says "Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud." and "Sign up for your Developer Edition." followed by a list of benefits: "Build apps fast with drag-and-drop tools", "Go further with Apex code", "Build AI agents with Agentforce", "Harmonize your data with Data Cloud", "Ground Agentforce with structured and unstructured data", and "Integrate with anything using APIs". On the right, there is a white form titled "Sign up for your Developer Edition" with fields for "First name", "Last name", "Job title", "Work email", "Company", and "Country/Region". Below the form, there is a checkbox for "I agree to the Main Services Agreement" and a note about Hyperforce. The bottom screenshot shows the "Seller Home" dashboard. It has a navigation bar with "Sales" and "Home" tabs. The dashboard includes several widgets: "Close Deals" (Total Pipeline \$0, \$0 Open, \$0 Won, \$0 Lost), "Plan My Accounts" (0 Accounts, 0 Upcoming Activity, 0 Past Activity, 0 No Activity), "Grow Relationships" (0 Contacts, 0 Upcoming Activity, 0 Past Activity, 0 No Activity), "Build Pipeline" (Leads owned by me and created in the last 30 days), "My Goals" (Set personal weekly or monthly goals for emails, calls, and meetings), and "Today's Events".

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name Last name

Job title Work email

Company Country/Region

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☐ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

By registering I confirm that I have read and agree to the Privacy Statement.

orgfarm-a01ed0511a-dev-ed.develop.lightning.force.com/lightning/page/home

Search...

Sales Home Opportunities Leads Tasks Files Accounts Contacts Campaigns Dashboards Reports Chatter Groups Calendar More

Seller Home Good morning, Workforce. Let's get selling!

Close Deals Opportunities owned by me and closing this quarter

\$0 Total Pipeline

- \$0 Open
- \$0 Won
- \$0 Lost

View Opportunities

Plan My Accounts Accounts owned by me

0 Accounts

- 0 Upcoming Activity
- 0 Past Activity
- 0 No Activity

View Accounts

Grow Relationships Contacts owned by me and created in the last 90 days

0 Contacts

- 0 Upcoming Activity
- 0 Past Activity
- 0 No Activity

View Contacts

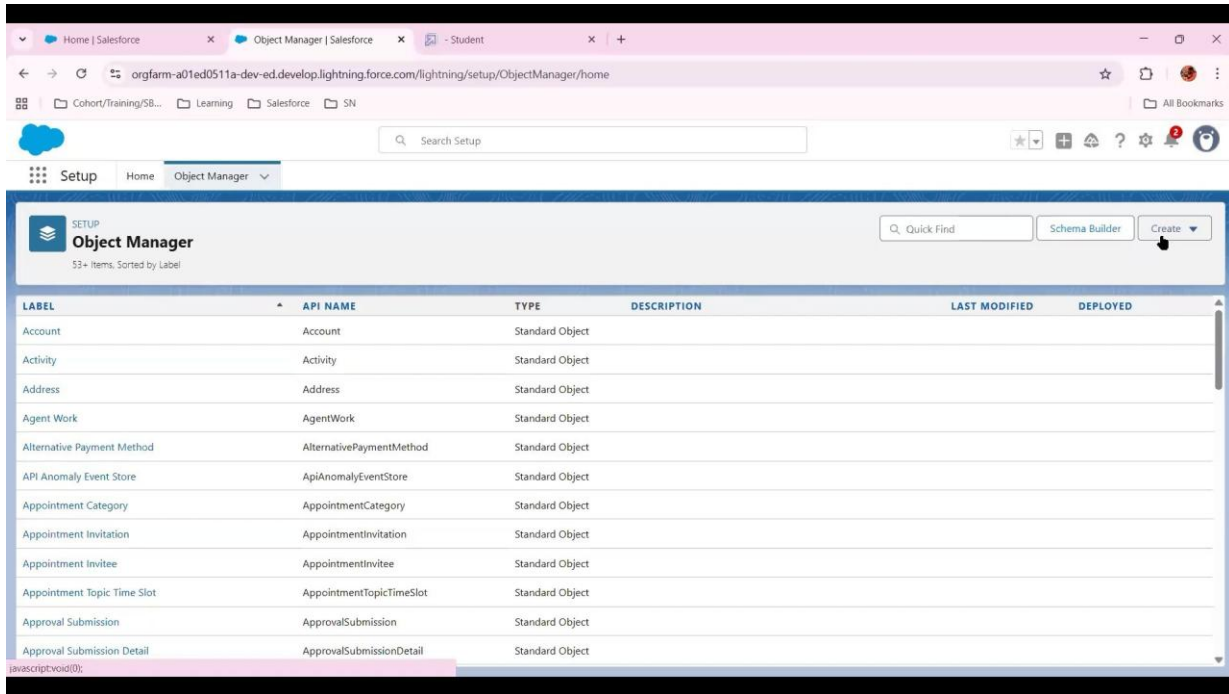
Build Pipeline Leads owned by me and created in the last 30 days

My Goals Set personal weekly or monthly goals for emails, calls, and meetings.

Today's Events

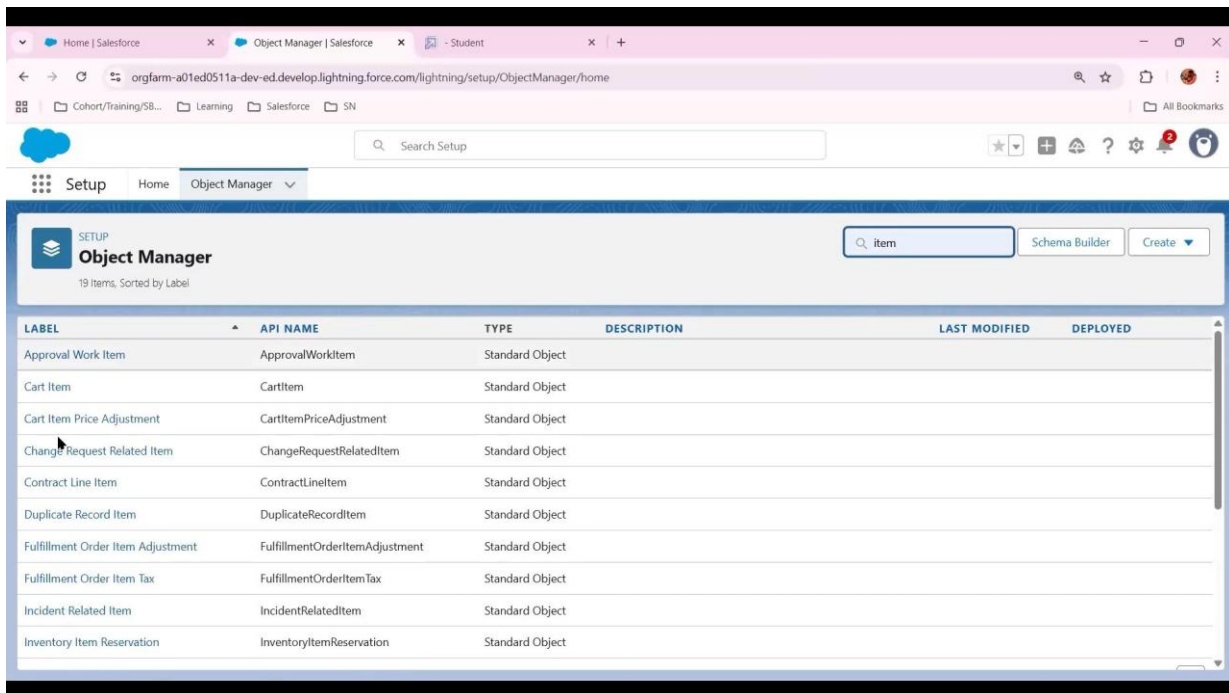
To Do List

❖ Created objects: Property, Tenant, Lease, Payment



The screenshot shows the Salesforce Object Manager interface. The browser address bar displays the URL: `orgfarm-a01ed0511a-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/home`. The page header includes a search bar with the text "Search Setup" and a "Create" button. The main content area is titled "Object Manager" and shows a list of 53 items, sorted by Label. The table below lists the objects:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Account	Account	Standard Object			
Activity	Activity	Standard Object			
Address	Address	Standard Object			
Agent Work	AgentWork	Standard Object			
Alternative Payment Method	AlternativePaymentMethod	Standard Object			
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object			
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object			
Approval Submission	ApprovalSubmission	Standard Object			
Approval Submission Detail	ApprovalSubmissionDetail	Standard Object			



The screenshot shows the Salesforce Object Manager interface with a search filter applied. The browser address bar displays the URL: `orgfarm-a01ed0511a-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/home`. The page header includes a search bar with the text "Search Setup" and a "Create" button. The main content area is titled "Object Manager" and shows a list of 19 items, sorted by Label. The table below lists the objects:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Approval Work Item	ApprovalWorkItem	Standard Object			
Cart Item	CartItem	Standard Object			
Cart Item Price Adjustment	CartItemPriceAdjustment	Standard Object			
Change Request Related Item	ChangeRequestRelatedItem	Standard Object			
Contract Line Item	ContractLineItem	Standard Object			
Duplicate Record Item	DuplicateRecordItem	Standard Object			
Fulfillment Order Item Adjustment	FulfillmentOrderItemAdjustment	Standard Object			
Fulfillment Order Item Tax	FulfillmentOrderItemTax	Standard Object			
Incident Related Item	IncidentRelatedItem	Standard Object			
Inventory Item Reservation	InventoryItemReservation	Standard Object			

❖ Configured fields and relationships

The screenshot shows the Salesforce Setup interface for the 'Jewel Customer' object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The 'Details' section is active, showing the following information:

- Description:** (Empty field)
- API Name:** Jewel_Customer__c
- Custom:** ☒
- Singular Label:** Jewel Customer
- Plural Label:** Jewel Customers
- Enable Reports:** ☒
- Track Activities:** ☒
- Track Field History:** ☐
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

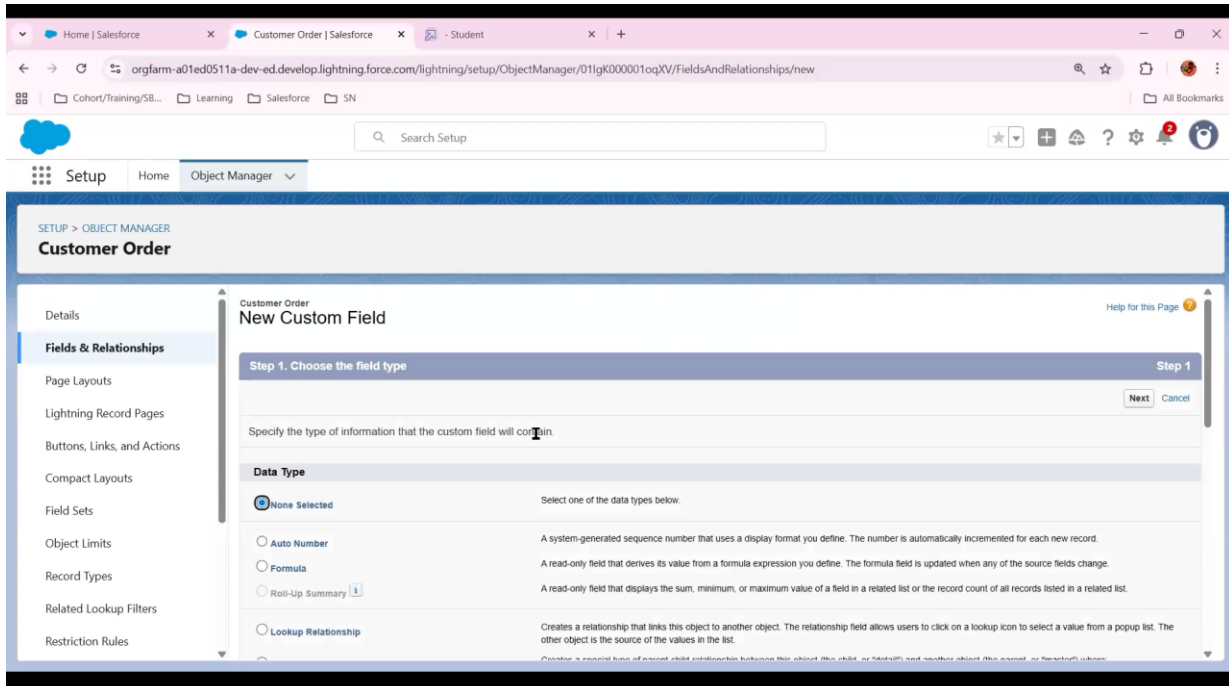
Buttons for 'Edit' and 'Delete' are located in the top right corner of the details section.

The screenshot shows the Salesforce Setup interface for the 'Item' object, specifically the 'Page Layouts' section. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Restriction Rules. The 'Page Layouts' section is active, showing a table of page layouts:

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Item Layout	Workforce Project, 8/24/2025, 9:02 PM	Workforce Project, 8/24/2025, 11:24 PM

Buttons for 'New' and 'Page Layout Assignment' are located in the top right corner of the page layout section.

❖ Create New Custom Object



This screenshot shows the 'New Custom Field' page in Salesforce Setup. The breadcrumb trail is 'SETUP > OBJECT MANAGER'. The left sidebar lists navigation options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Restriction Rules. The main content area is titled 'Customer Order' and 'New Custom Field'. It features a 'Step 1. Choose the field type' section with a 'Next' button. Below this, a 'Data Type' section allows selection from 'None Selected', 'Auto Number', 'Formula', 'Roll-Up Summary', and 'Lookup Relationship'. Each option includes a brief description of its function.

Home | Salesforce | Customer Order | Salesforce | Student | +

orgfarm-a01ed0511a-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gK000001oqXV/FieldsAndRelationships/new

Setup | Home | Object Manager

SETUP > OBJECT MANAGER
Customer Order

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules

Customer Order
New Custom Field

Step 1. Choose the field type

Specify the type of information that the custom field will contain.

Data Type

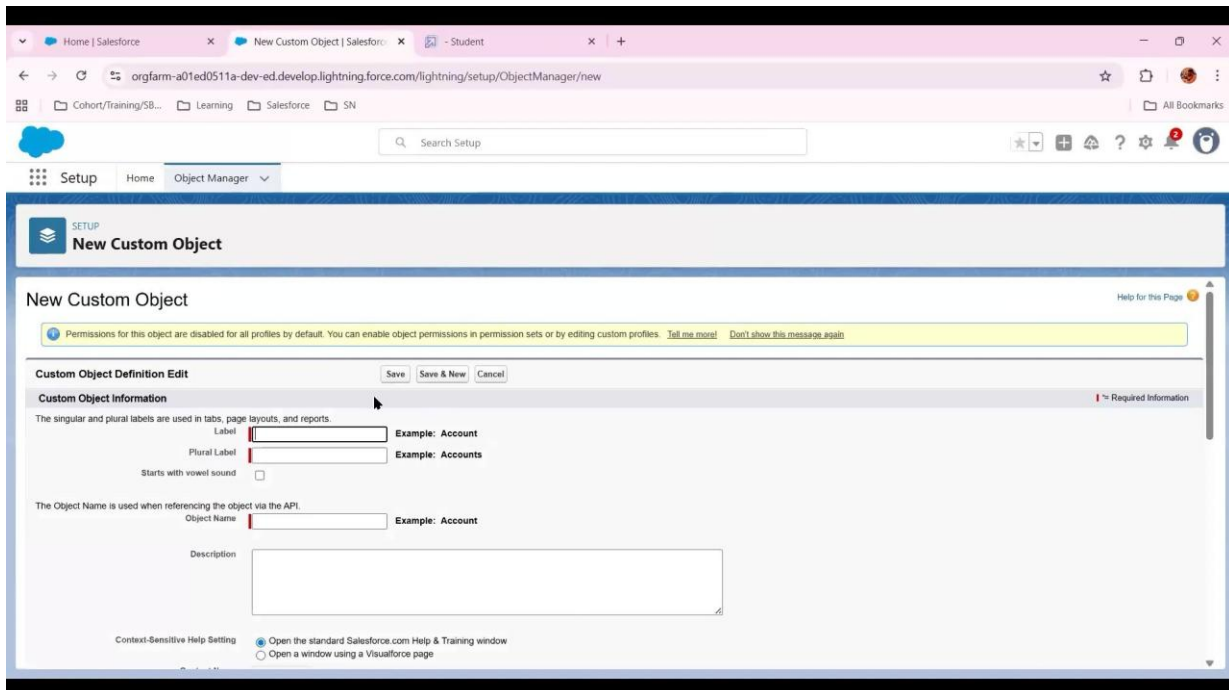
☒ None Selected Select one of the data types below.

☐ Auto Number A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

☐ Formula A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

☐ Roll-Up Summary A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

☐ Lookup Relationship Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.



This screenshot shows the 'New Custom Object' page in Salesforce Setup. The breadcrumb trail is 'SETUP > OBJECT MANAGER'. The left sidebar is the same as the previous screenshot. The main content area is titled 'New Custom Object'. It features a 'Custom Object Definition Edit' section with 'Save', 'Save & New', and 'Cancel' buttons. Below this, a 'Custom Object Information' section contains fields for 'Label', 'Plural Label', 'Object Name', and 'Description'. Each field has an example value (e.g., 'Account', 'Accounts'). There is also a 'Context-Sensitive Help Setting' section with two radio button options.

Home | Salesforce | New Custom Object | Salesforce | Student | +

orgfarm-a01ed0511a-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new

Setup | Home | Object Manager

SETUP
New Custom Object

New Custom Object

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more!](#) [Don't show this message again](#)

Custom Object Definition Edit Save Save & New Cancel

Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label Example: Account

Plural Label Example: Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name Example: Account

Description

Context-Sensitive Help Setting

☒ Open the standard Salesforce.com Help & Training window

☐ Open a window using a Visualforce page

❖ Create New User and Roles

The screenshot shows the Salesforce Setup page for Users. The left sidebar contains a search bar and a list of navigation items: Users, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, Feature Settings, Data.com, and Prospector Users. The main content area is titled "All Users" and includes a description: "On this page you can create, view, and manage users. To get more licenses, use the Your Account app. Let's Go". Below this is a "View" dropdown set to "All Users" and a "Create New User" link. A table lists existing users with columns for Action, Full Name, Alias, Username, Role, Active, and Profile. The table includes users like Chatter Expert, EPIC_OrgFarm, Mikaelson_Kol, Mikaelson_Niklaus, Mikaelson_Paul, Mikaelson_Van, Project_Workforce, and User_Integration. At the bottom of the table are buttons for "New User", "Reset Password(s)", and "Add Multiple Users".

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty.00d0k00000b4sluas.grbzarrhwa@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	EPIC_OrgFarm	OEPIG	epic.a61286b7b47@orgfarm.salesforce.com		✓	System Administrator
<input type="checkbox"/> Edit	Mikaelson_Kol	kmika	kol@org.com	Manager	✓	Manager
<input type="checkbox"/> Edit	Mikaelson_Niklaus	nmika	niklaus@org.com	HR	✓	HR
<input type="checkbox"/> Edit	Mikaelson_Paul	pmika	paulm@org.com	On Site Employee	✓	On Site Employee
<input type="checkbox"/> Edit	Mikaelson_Van	vmika	van@org.com	Remote Employee	✗	Remote Employee
<input type="checkbox"/> Edit	Project_Workforce	nad	nadeem816@agentforce.com		✓	System Administrator
<input type="checkbox"/> Edit	User_Integration	integ	integration@00d0k00000b4sluas.com		✓	Analytics Cloud Integration User

SAKTHI

The screenshot shows the Salesforce Setup page for Roles. The left sidebar contains a search bar and a list of navigation items: Roles, Users, Feature Settings, Sales, Contact Roles on Contracts, Contact Roles on Opportunities, Service, Case Teams, Case Team Roles, and Contact Roles on Cases. The main content area is titled "Roles" and includes a description: "Set up your Role Hierarchy to control how your organization reports on and accesses data. Sample Role Hierarchy View other sample Role Hierarchies: Territory-based Sample". Below this is a diagram showing a role hierarchy. The hierarchy starts with "Executive Staff" (CEO, President, CFO, VP, Sales) at the top. Below this are three boxes: "Western Sales Director", "Eastern Sales Director", and "International Sales Director". Each of these boxes has a "Director of W. Sales", "Director of E. Sales", and "Director of Int'l Sales" respectively. Below these are three boxes: "Western Sales Rep", "Eastern Sales Rep", and "International Sales Rep". Each of these boxes has a "CA Sales Rep", "NY Sales Rep", and "Asian Sales Rep" respectively. The diagram also includes a "Set Up Roles" button and a checkbox for "Don't show this page again".

Set Up Roles

☐ Don't show this page again

SAKTHIVEL

❖ Create New Lightning App

New Lightning App

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

* App Name ¹
Name your app...

* Developer Name ¹
Enter a developer name...

Description ¹
Enter a description...

App Branding

Image ¹
Upload

Primary Color Hex Value ¹
#0070D2

Org Theme Options
☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview

New Lightning App

Available Items ¹ Create

Item

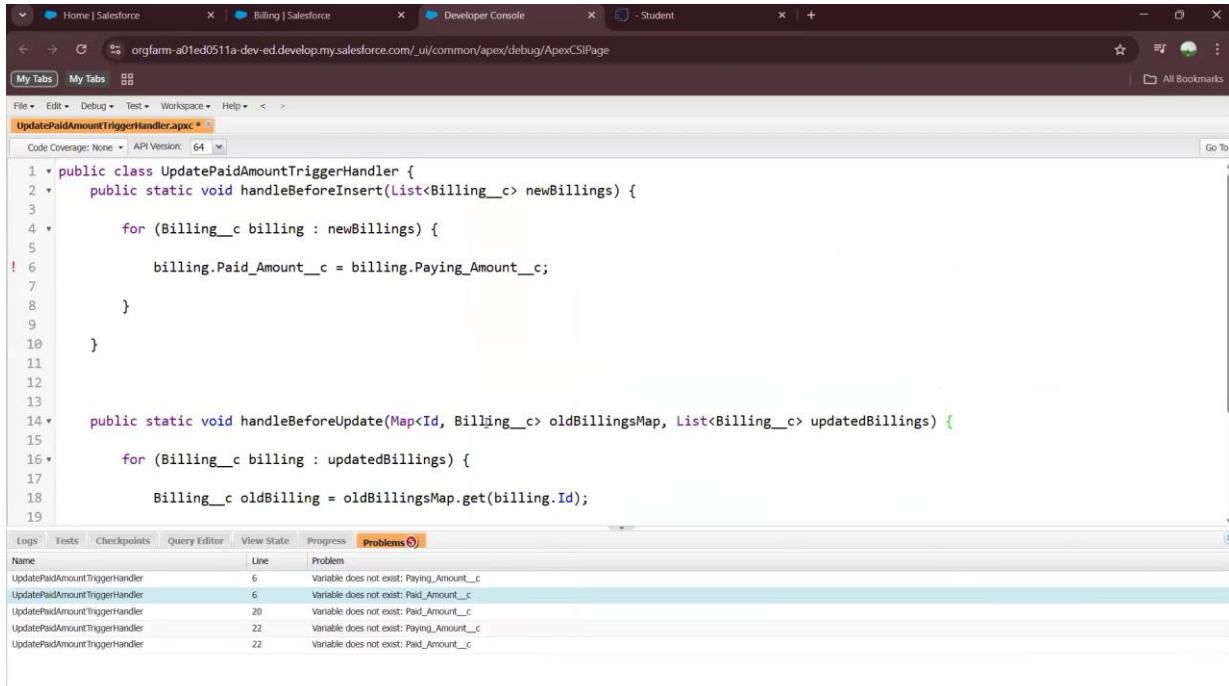
- Approval Work Items
- Contract Line Items
- Inventory Count Plan Items
- Inventory Count Product Batch Items
- Inventory Item Reservations
- IT Services Configured Items
- Items
- Orchestration Work Items
- Price Protection Execution Line Items
- Product Batch Items

Selected Items

Jewel Customers

Back Next

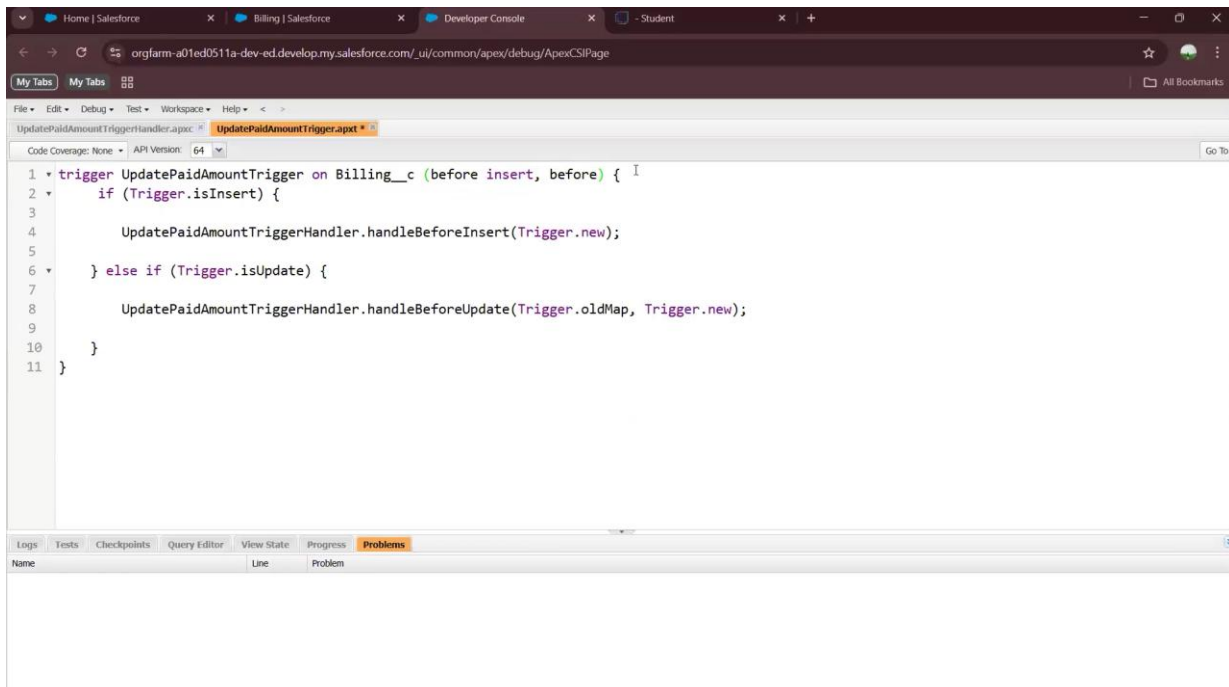
❖ Create An Apex Class



The screenshot shows the Salesforce Developer Console with the file `UpdatePaidAmountTriggerHandler.apex` open. The code defines a class `UpdatePaidAmountTriggerHandler` with two static methods: `handleBeforeInsert` and `handleBeforeUpdate`. The `handleBeforeInsert` method iterates over a list of `Billing__c` objects and attempts to assign `billing.Paid_Amount__c = billing.Paying_Amount__c;`. The `handleBeforeUpdate` method iterates over a list of `Billing__c` objects and attempts to retrieve an old billing record using `oldBillingsMap.get(billing.Id)`. The console shows four compilation errors, all stating "Variable does not exist: Paid_Amount__c".

```
1 public class UpdatePaidAmountTriggerHandler {
2     public static void handleBeforeInsert(List<Billing__c> newBillings) {
3
4         for (Billing__c billing : newBillings) {
5
6             billing.Paid_Amount__c = billing.Paying_Amount__c;
7
8         }
9     }
10 }
11
12
13
14 public static void handleBeforeUpdate(Map<Id, Billing__c> oldBillingsMap, List<Billing__c> updatedBillings) {
15
16     for (Billing__c billing : updatedBillings) {
17
18         Billing__c oldBilling = oldBillingsMap.get(billing.Id);
19     }
20 }
```

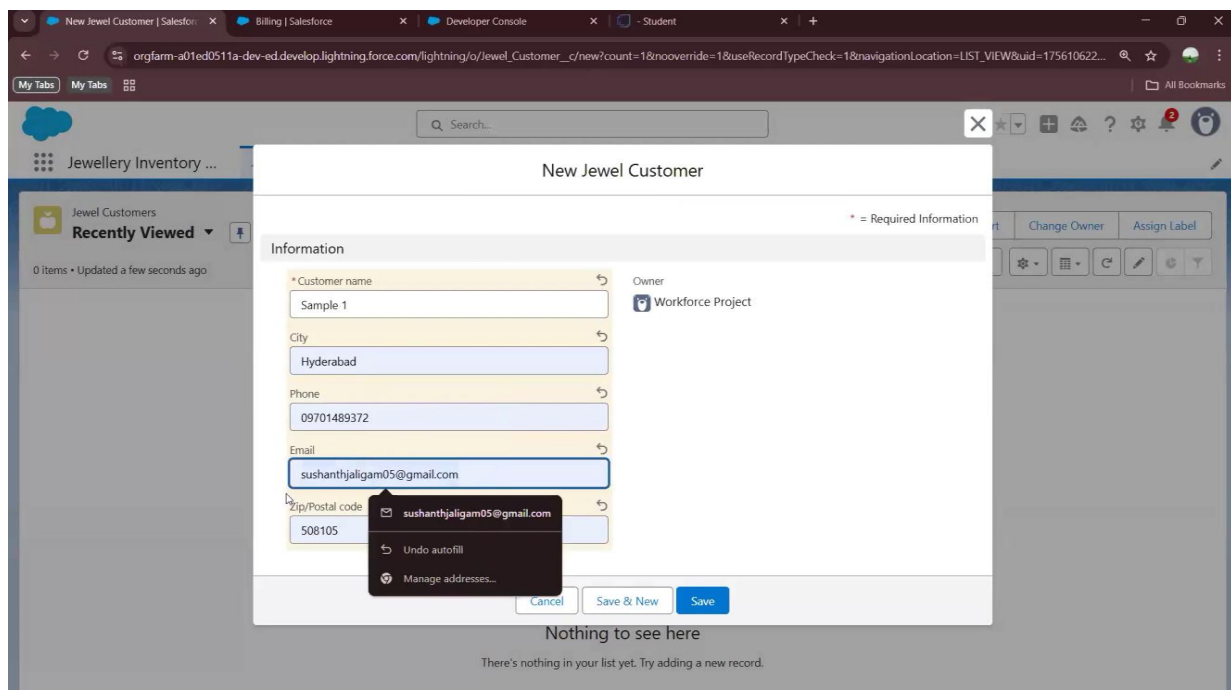
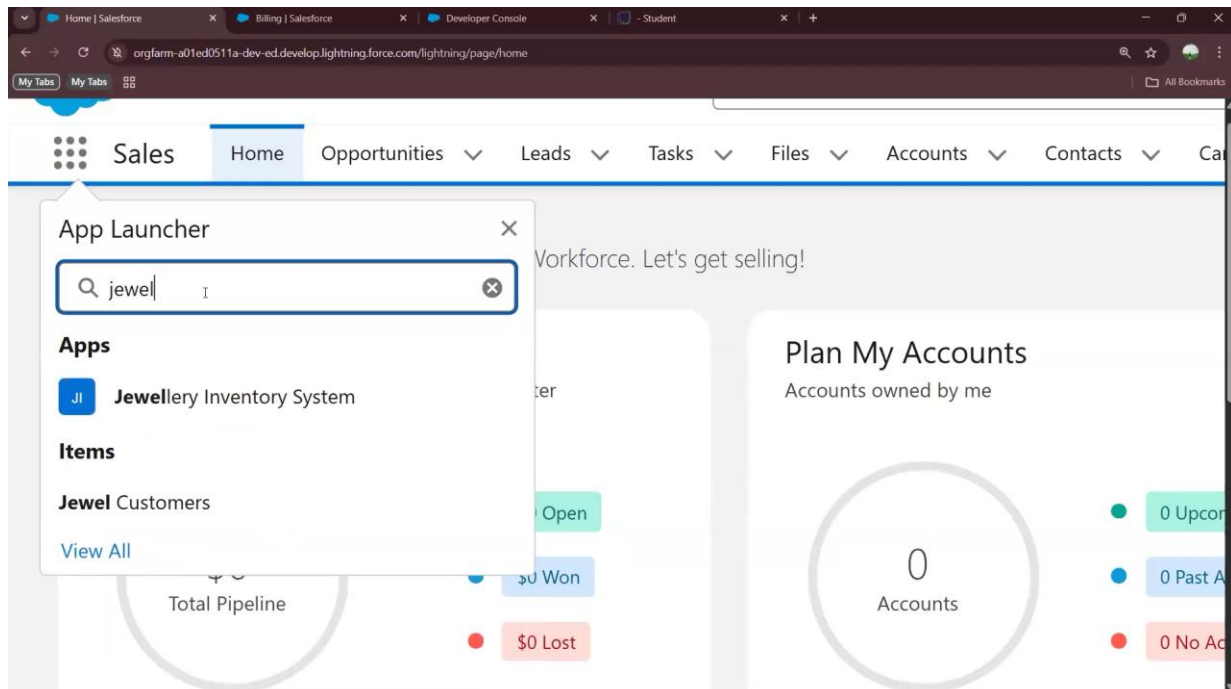
Name	Line	Problem
UpdatePaidAmountTriggerHandler	6	Variable does not exist: Paying_Amount__c
UpdatePaidAmountTriggerHandler	6	Variable does not exist: Paid_Amount__c
UpdatePaidAmountTriggerHandler	20	Variable does not exist: Paid_Amount__c
UpdatePaidAmountTriggerHandler	22	Variable does not exist: Paying_Amount__c
UpdatePaidAmountTriggerHandler	22	Variable does not exist: Paid_Amount__c



The screenshot shows the Salesforce Developer Console with the file `UpdatePaidAmountTrigger.apxt` open. The code defines a trigger `UpdatePaidAmountTrigger` on the `Billing__c` object. The trigger has two conditions: `before insert, before` and `before update, before`. The `before insert, before` condition calls `UpdatePaidAmountTriggerHandler.handleBeforeInsert`. The `before update, before` condition calls `UpdatePaidAmountTriggerHandler.handleBeforeUpdate`. The console shows no compilation errors.

```
1 trigger UpdatePaidAmountTrigger on Billing__c (before insert, before) {
2     if (Trigger.isInsert) {
3
4         UpdatePaidAmountTriggerHandler.handleBeforeInsert(Trigger.new);
5     }
6 } else if (Trigger.isUpdate) {
7
8     UpdatePaidAmountTriggerHandler.handleBeforeUpdate(Trigger.oldMap, Trigger.new);
9 }
10 }
11 }
```

❖ Go to Home and Search For Jewellery Inventory System



❖ CreateNew Customer and Give Price

SAKTHIVEL

New Jewel Customer

* = Required Information

Information

* Customer name
Sample 1

City
Hyderabad

Phone
09701489372

Email
sushanthjaligam05@gmail.com

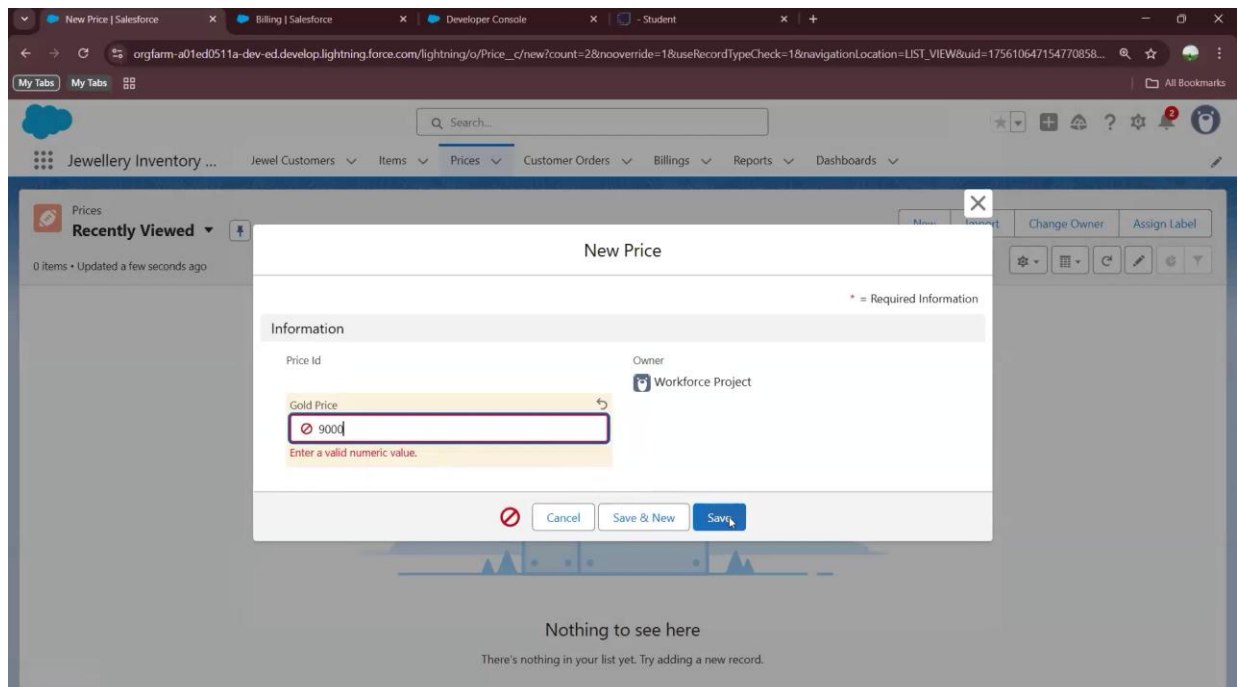
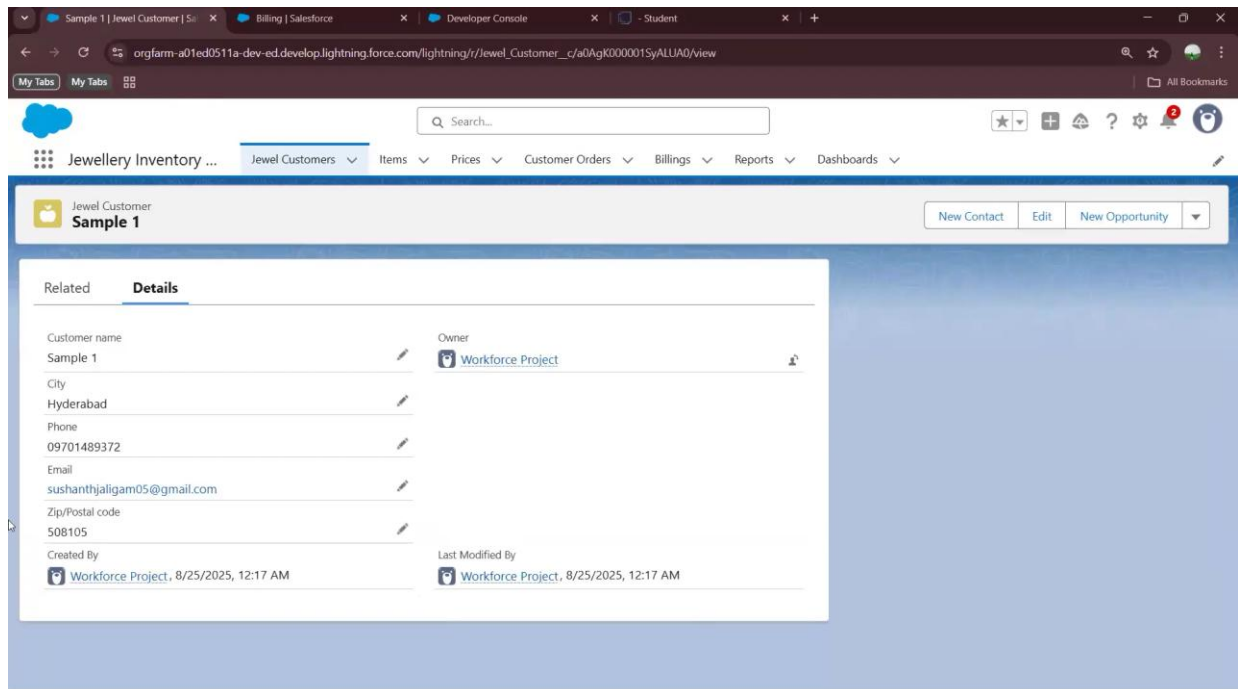
Zip/Postal code
508105

Owner
Workforce Project

Undo autofill
Manage addresses...

Cancel Save & New Save

Nothing to see here
There's nothing in your list yet. Try adding a new record.



❖ Check For Price Report

The screenshot shows the Salesforce interface with the 'Prices' tab selected. A green notification banner at the top states 'Price "PD-0001" was created.' The main content area displays the details for Price ID PD-0001. The 'Details' tab is active, showing the following information:

- Price Id: PD-0001
- Gold Price: \$9,000
- Created By: Workforce Project, 8/25/2025, 12:21 AM
- Last Modified By: Workforce Project, 8/25/2025, 12:21 AM

The 'Related' tab is also visible, showing a list of related records.

SAKTHI

The screenshot shows the Salesforce interface with the 'Reports' tab selected. The 'Price Report' is displayed, showing a summary of the report data. The report includes the following information:

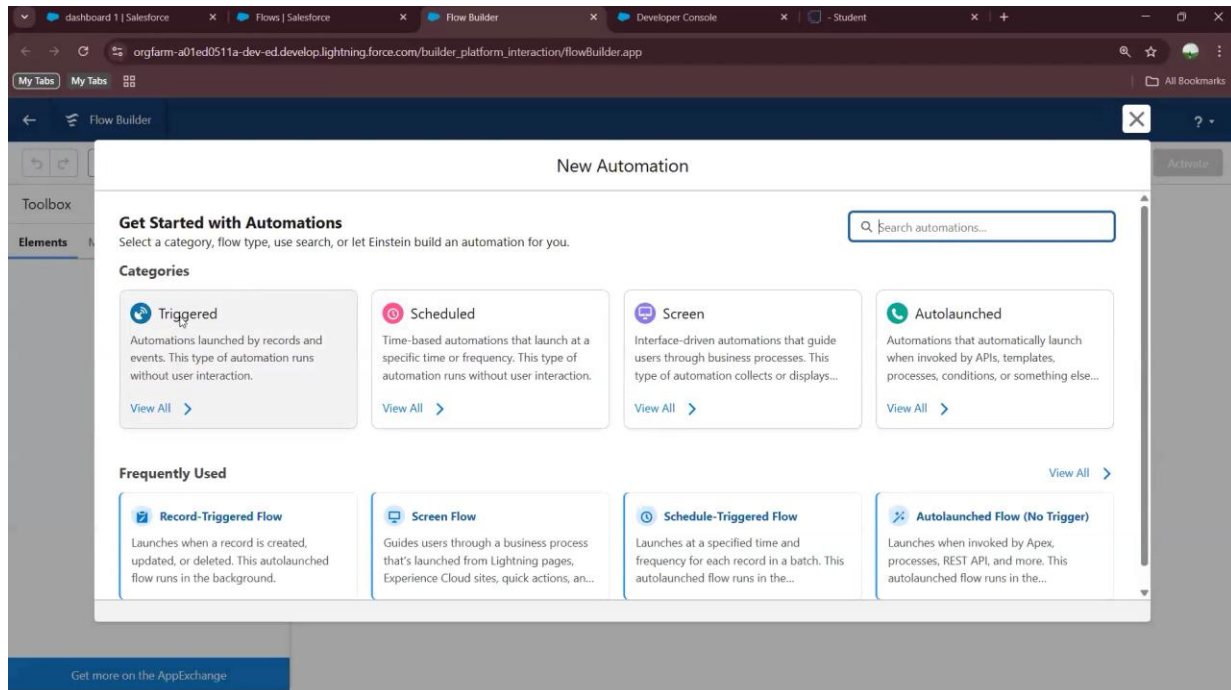
- Report: Prices
- Price Report
- Enable Field Editing
- Search
- Add Chart
- Edit

The report data is summarized as follows:

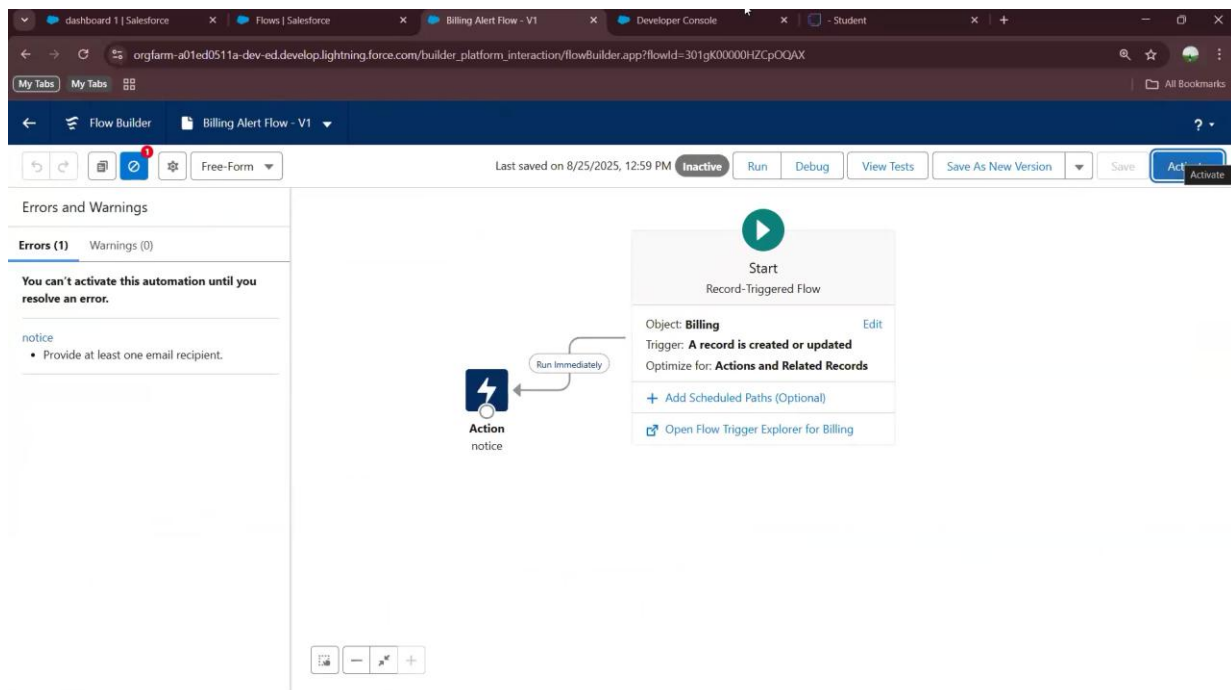
Price: ID	Gold Price	Price: Price Id
1	\$9,000	PD-0001
2	\$9,000	

SAKTHIVEL

❖ Create New Automation and Start



SK



ADVANTAGES :

1. DOMAIN-SPECIFIC SOLUTION :

- ❖ Focused on jewelry business needs like inventory tracking (gold, diamonds, stones), customized billing, and customer preferences.
- ❖ Easier to target a niche market compared to generic CRMs.

2. SCALABILITY & FLEXIBILITY :

- ❖ Developers can design the system to handle both small jewelry shops and larger chains.
- ❖ Can integrate features like loyalty programs, gemstone certifications, and repair tracking.

3. INTEGRATION OPPORTUNITIES :

- ❖ Can connect with POS systems, e-commerce websites, and accounting tools.
- ❖ Increases product value and gives developers scope to expand modules.

4. DATA-DRIVEN INSIGHTS :

- ❖ Offers jewelers customer insights (purchase history, trends).
- ❖ Developers can implement analytics dashboards, improving usability and business decision-making.

5. USER-FRIENDLY UI FOCUS :

- ❖ Jewelry staff may not be very tech-savvy, so creating an intuitive interface is a strong developer achievement.
- ❖ Opportunity to showcase skill in UI/UX design.

6. SECURITY-ORIENTED DEVELOPMENT :

- ❖ Sensitive customer and pricing data require strong encryption.
- ❖ Developers gain experience in secure coding practices (GDPR, data privacy compliance).

DISADVANTAGES :

1. COMPLEX CUSTOMIZATION :

- ❖ Jewelry items vary in weight, purity, stone count, certification, etc.
- ❖ Building flexible yet simple modules can be technically challenging.

2. HIGH DEVELOPMENT COST & TIME :

- ❖ Requires multiple features (CRM + inventory + invoicing + reporting).
- ❖ Can be overwhelming for a student project or small developer team.

3. SECURITY RISKS :

- ❖ Handling customer identity data, purchase values, and payment details.
- ❖ Vulnerable to cyber-attacks if not built with strong security layers.

4. MAINTENANCE CHALLENGES :

- ❖ Jewelry businesses may request frequent changes (new schemes, pricing structures, GST/tax updates).
- ❖ Developers need to ensure easy maintainability and version upgrades.

5. INTEGRATION DIFFICULTIES :

- ❖ Not all jewelers use modern systems; integrating with old accounting software or manual processes may be hard.

6. MARKET COMPETITION :

- ❖ Existing CRM tools (Zoho, Salesforce, etc.) already dominate.
- ❖ Niche customization may limit scalability outside the jewelry domain.

CONCLUSION :

- ❖ The development of the CRM Application for Jewel Management has successfully integrated customer relationship management, sales tracking, and inventory control into a single streamlined system. By centralizing customer data, purchase history, and stock information, this application not only enhances business efficiency but also improves customer satisfaction through personalized services and timely responses.
- ❖ From a developer's perspective, the project demonstrated the importance of database design, system integration, and user-friendly interface development in building a reliable solution. This application can be further expanded with advanced features such as data analytics, AI-driven recommendations, and mobile compatibility, ensuring long-term adaptability in the jewelry business.
- ❖ In conclusion, the project proves that a well-designed CRM system can bridge the gap between technology and traditional jewelry business operations, resulting in better decision-making, improved sales performance, and stronger customer relationships.

APPENDIX :

```
public with sharing class JewelCRMManager {

    // Add Customer
    public static Id addCustomer(String name, String phone, String email) {
        Customer__c cust = new Customer__c(
            Name = name,
            Phone__c = phone,
            Email__c = email
        );
        insert cust;
        return cust.Id;
    }

    // Get All Customers
    public static List<Customer__c> getCustomers() {
        return [SELECT Id, Name, Phone__c, Email__c FROM Customer__c];
    }

    // Add Jewel Item
```

```

publicstatic Id addJewel(String name, String type, Decimal price, Integer stock) {
    Jewel__c jewel = new Jewel__c(
        Name = name,
        Type__c = type,
        Price__c = price,
        Stock__c = stock
    );
    insert jewel;
    return jewel.Id;
}

//GetAll Jewelry Inventory
publicstatic List<Jewel__c> getInventory() {
    return [SELECT Id, Name, Type__c, Price__c, Stock__c FROM Jewel__c];
}

//Record Sale
publicstatic Id addSale(Id customerId, Id jewelId, Integer qty, Date sDate) {
    Jewel__c jewel = [SELECT Id, Stock__c FROM Jewel__c WHERE Id = :jewelId LIMIT 1];
    if(jewel.Stock__c < qty) {
        throw new AuraHandledException("Not enough stock available!");
    }

    //Reduce stock
    jewel.Stock__c -= qty;
    update jewel;

    Sale__c sale = new Sale__c(
        Customer__c = customerId,
        Jewel__c = jewelId,
        Quantity__c = qty,
        Date__c = sDate
    );
    insert sale;
    return sale.Id;
}

//GetAll Sales
publicstatic List<Sale__c> getSales() {
    return [
        SELECT Id, Customer__r.Name, Jewel__r.Name, Quantity__c, Date__c
        FROM Sale__c
    ];
}
}

```

APEX TEST CLASS :

```
@isTest
publicclass JewelCRMMManagerTest {
    @isTest
    staticvoid testCRMFlow() {
        //Add customer
        IdcustId = JewelCRMMManager.addCustomer('Sakthivel', '7845770375', 'alice@test.com');

        //Add jewel
        IdjewelId = JewelCRMMManager.addJewel('Gold Ring', 'Ring', 15000, 10);

        //Record sale
        IdsaleId = JewelCRMMManager.addSale(custId, jewelId, 2, Date.today());

        //Query results
        List<Customer__c> customers = JewelCRMMManager.getCustomers();
        List<Jewel__c> jewels = JewelCRMMManager.getInventory();
        List<Sale__c> sales = JewelCRMMManager.getSales();

        System.assertEquals(1, customers.size());
        System.assertEquals(1, jewels.size());
        System.assertEquals(1, sales.size());
    }
}
```