

CRM APPLICATION FOR JEWEL MANAGEMENT

SK

**COLLEGE NAME : GOVERNMENT ARTS COLLEGE
UDHAGAMANDALAM**

COLLEGE CODE : BRU11

TEAM ID :

TEAM MEMBERS : 4

TEAM LEADER NAME : SANJANA

E-MAIL :

TEAM MEMBER 1 : MANOJ

E-MAIL :

TEAM MEMBER : SAKTHIVEL

E-MAIL :

TEAM MEMBER : ARIF

E-MAIL :

1.INTRODUCTION

1.1 PROJECT OVERVIEW

- ❖ 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



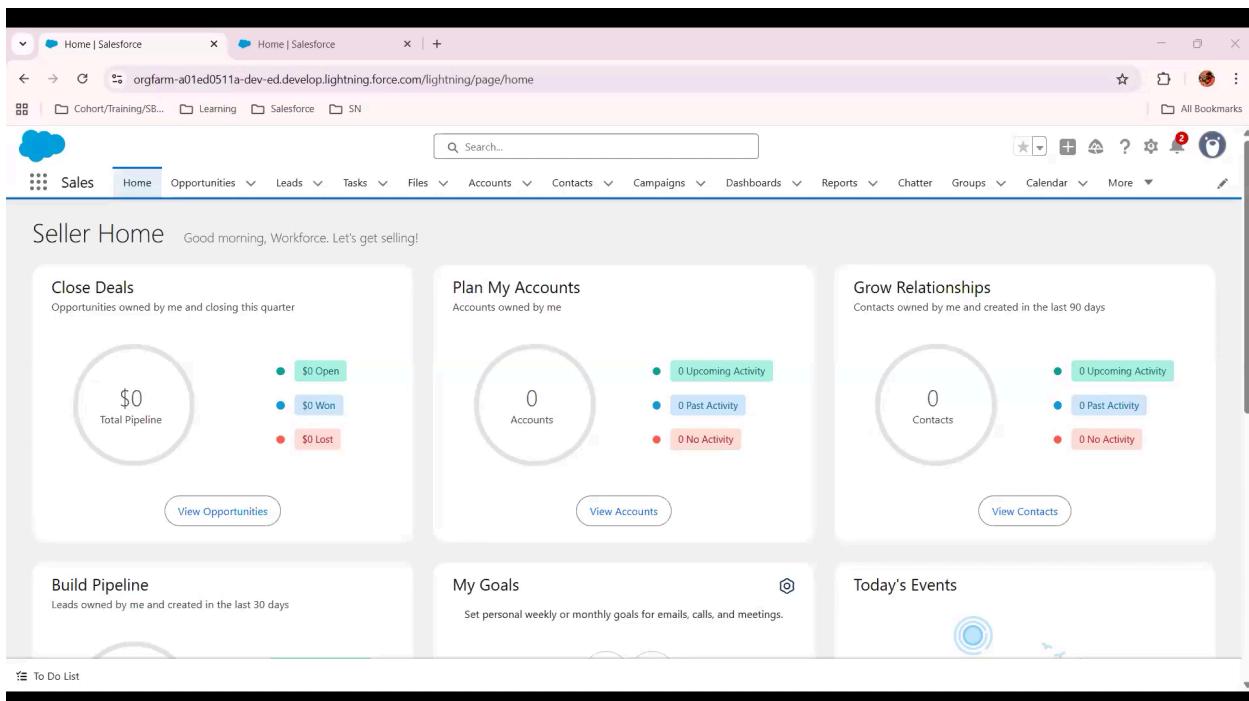
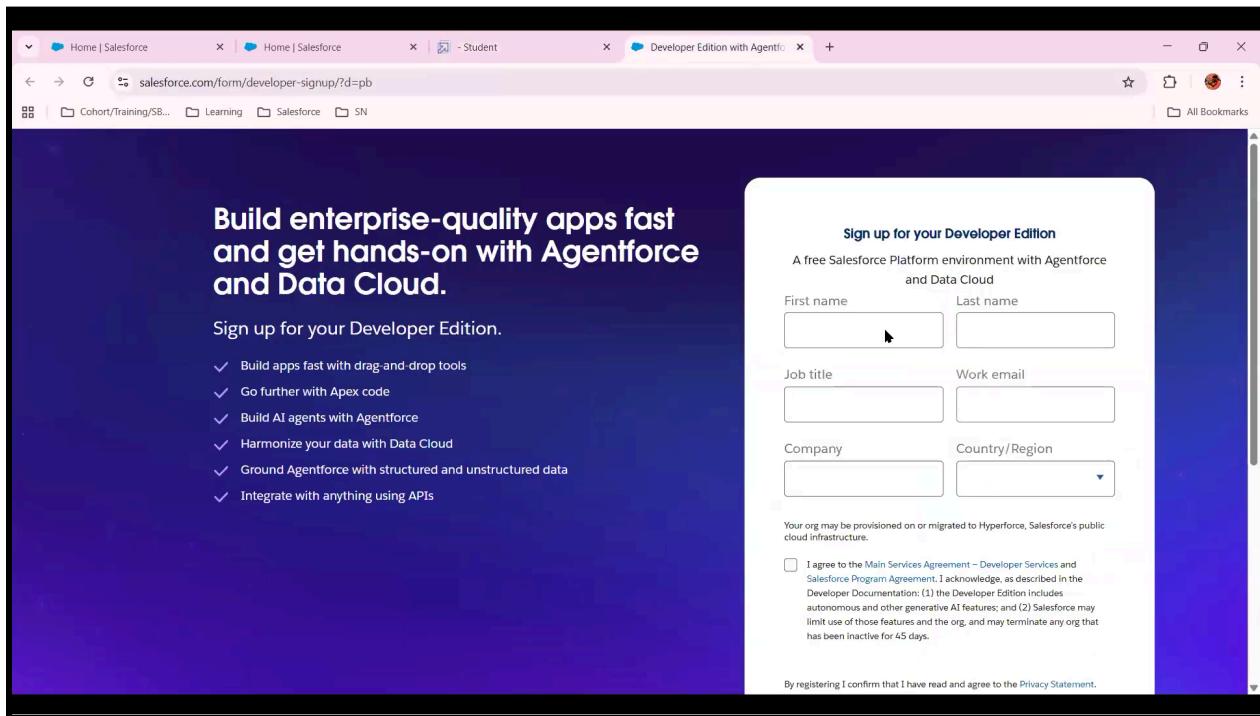
1.2 PURPOSE

- ❖ 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/>



❖ **Created objects: Property, Tenant, Lease, Payment**

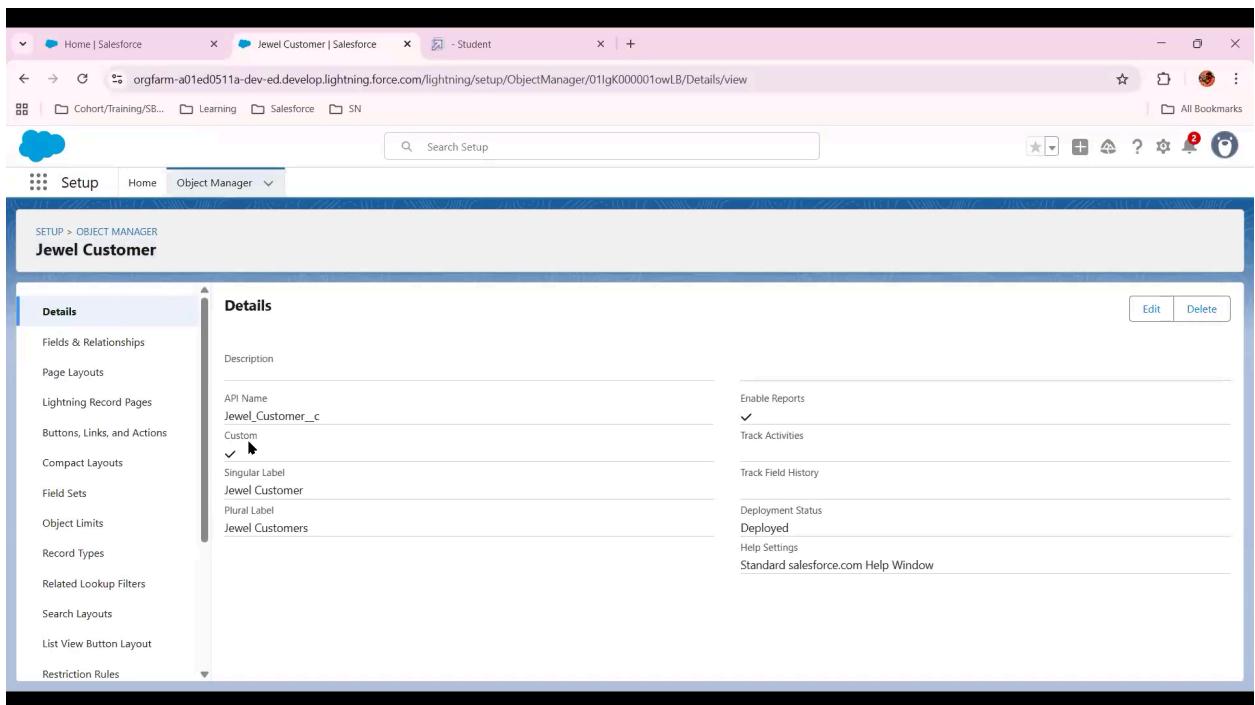
The screenshot shows the Salesforce Object Manager interface. The title bar includes tabs for Home, Object Manager, and Student. The main content area is titled "Object Manager" under "SETUP". A search bar at the top right contains "Search Setup". Below it is a "Create" button. The main table lists various objects with columns for Label, API Name, Type, Description, Last Modified, and Deployed. The table shows 53+ items, sorted by Label. The "Approval Submission Detail" object is highlighted with a pink background.

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			

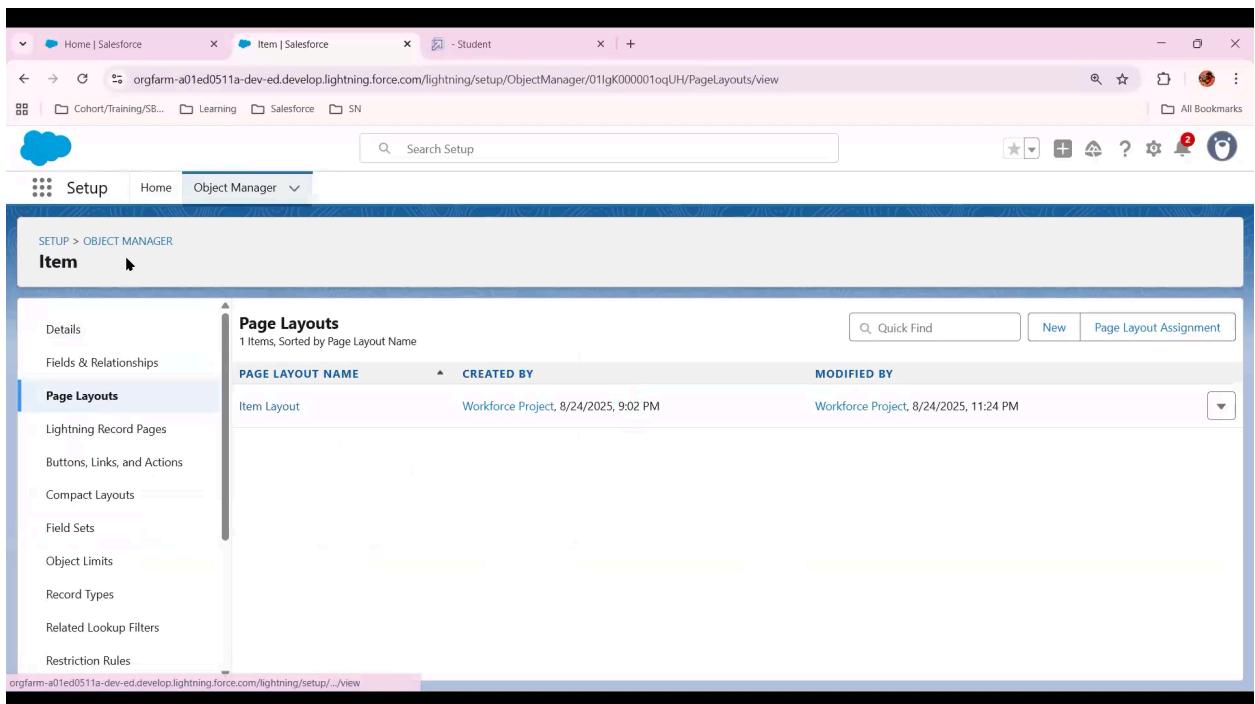
The screenshot shows the same Salesforce Object Manager interface, but with a search bar containing "item" and a results count of 19 items. The table lists objects related to items, such as Approval Work Item, Cart Item, and Incident Related Item. The "Cart Item Price Adjustment" object is highlighted with a pink background.

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 'Object Manager' section. The object being configured is 'Jewel Customer'. The left sidebar lists various configuration 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 main panel displays the 'Details' tab for 'Jewel Customer'. It includes fields for Description, API Name (set to 'Jewel_Customer__c'), Singular Label ('Jewel Customer'), and Plural Label ('Jewel Customers'). Other settings shown include Enable Reports (checked), Track Activities (checked), Track Field History, Deployment Status (set to 'Deployed'), and Help Settings. There are 'Edit' and 'Delete' buttons at the top right of the details panel.



The screenshot shows the Salesforce Setup interface for the 'Object Manager' section. The object being configured is 'Item'. The left sidebar lists various configuration 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 main panel displays the 'Page Layouts' tab for 'Item'. It shows a table with one item: 'Item Layout', created by 'Workforce Project' on 8/24/2025, 9:02 PM, and modified by 'Workforce Project' on 8/24/2025, 11:24 PM. The table has columns for 'PAGE LAYOUT NAME', 'CREATED BY', and 'MODIFIED BY'. There are 'Quick Find', 'New', and 'Page Layout Assignment' buttons at the top right of the table area.

❖ Create New Custom Object

The screenshot shows the Salesforce Setup interface. The top navigation bar includes tabs for Home, Object Manager, and a search bar. The main content area is titled "Customer Order" under "SETUP > OBJECT MANAGER". A sidebar on the left lists various setup categories like Details, Fields & Relationships, Page Layouts, and Record Types. The current step is "Step 1. Choose the field type" for a "New Custom Field" on the "Customer Order" object. The "Data Type" section is open, showing options: "None Selected" (selected), "Auto Number", "Formula", "Roll-Up Summary", and "Lookup Relationship". Each option has a brief description below it.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes tabs for Home, Object Manager, and a search bar. The main content area is titled "New Custom Object" under "SETUP". A message at the top states: "Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles." Below this is the "Custom Object Definition Edit" form. The "Custom Object Information" section requires filling in the "Label" (set to "Account") and "Plural Label" (set to "Accounts"). There is also a checkbox for "Starts with vowel sound". The "Object Name" field is set to "Account". The "Description" field is empty. At the bottom, there are "Context-Sensitive Help Setting" options: "Open the standard Salesforce.com Help & Training window" (selected) and "Open a window using a Visualforce page".

❖ Create New User and Roles

All Users

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. [Let's Go](#)

View: All Users | Edit | Create New View

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter Expert	Chatter	chatty@00dgk00000v4sluas.orgfarm.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	EPIC_OrgFarm	OEPI	spic.a61286bf7b47@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Mikaelson_Kol	kmika	kol@org.com	Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	Mikaelson_Niklaus	nmika	niklaus@org.com	HR	<input checked="" type="checkbox"/>	HR
<input type="checkbox"/>	Mikaelson_Paul	pmika	pmaul@org.com	On Site Employee	<input checked="" type="checkbox"/>	On Site Employee
<input type="checkbox"/>	Mikaelson_Van	vmika	van@org.com	Remote Employee	<input type="checkbox"/>	Remote Employee
<input type="checkbox"/>	Project_Workforce	nad	nadeem816@agentforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	User_Integration	integ	integration@00dgk00000v4sluas.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User

Understanding Roles

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](#)

```

graph TD
    CEO[CEO President] --> CFO[CFO VP, Sales]
    CEO --> SalesDir[Sales Director]
    CFO --> WSD[Western Sales Director]
    CFO --> ESD[Eastern Sales Director]
    CFO --> ISD[International Sales Director]
    WSD --> WSR[Western Sales Rep]
    ESD --> ESR[Eastern Sales Rep]
    ISD --> ISR[International Sales Rep]
  
```

Executive Staff
CEO President, CFO VP, Sales, Sales Director

Western Sales Director
Director of W. Sales

Eastern Sales Director
Director of E. Sales

International Sales Director
Director of Int'l Sales

Western Sales Rep
CA Sales Rep, OR Sales Rep

Eastern Sales Rep
NY Sales Rep, MA Sales Rep

International Sales Rep
Asian Sales Rep, European Sales Rep

* View & edit data, roll up forecasts, & generate reports for all users below.
* Can't access data of users above or at same level

* View & edit data, roll up forecasts, & generate reports for all users directly below.
* Can't access data of users above or at same level

* View & edit data, roll up forecasts, & generate reports only for own data or for data of users above or at same level

[Set Up Roles](#)

Don't show this page again

SAKTHIVEL

❖ Create New Lighting App

The screenshot shows the 'App Manager | Salesforce' interface with the title 'New Lightning App'. The page is divided into sections: 'App Details' and 'App Branding'. In 'App Details', fields include 'App Name' (placeholder 'Name your app...'), 'Developer Name' (placeholder 'Enter a developer name...'), and 'Description' (placeholder 'Enter a description...'). In 'App Branding', there's an 'Image' section with an 'Upload' button and a color picker set to '#0070D2'. A checkbox 'Org Theme Options' is present. Below these sections is a preview area labeled 'App Launcher Preview'.

The screenshot shows the 'New Lightning App' configuration page. It features two main panels: 'Available Items' on the left and 'Selected Items' on the right. The 'Available Items' panel lists various items like 'Approval Work Items', 'Contract Line Items', etc. The 'Selected Items' panel contains one item, 'Jewel Customers'. At the bottom, there are 'Back' and 'Next' buttons, and a progress bar indicating the configuration process.

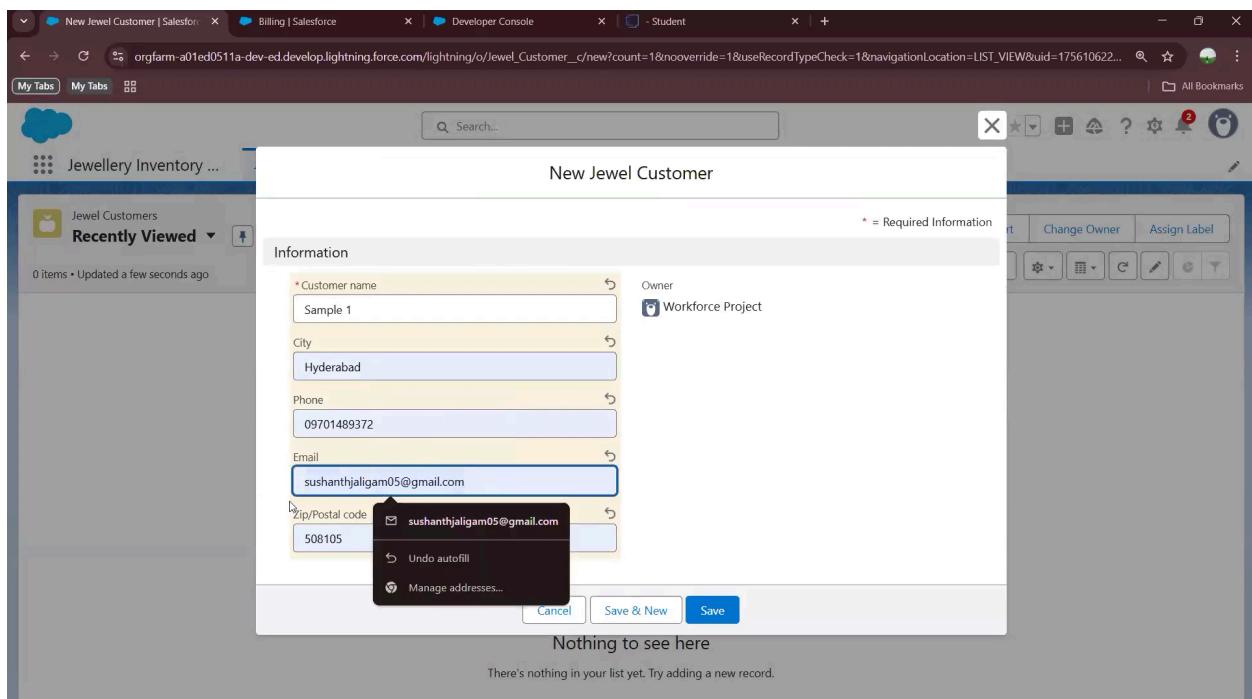
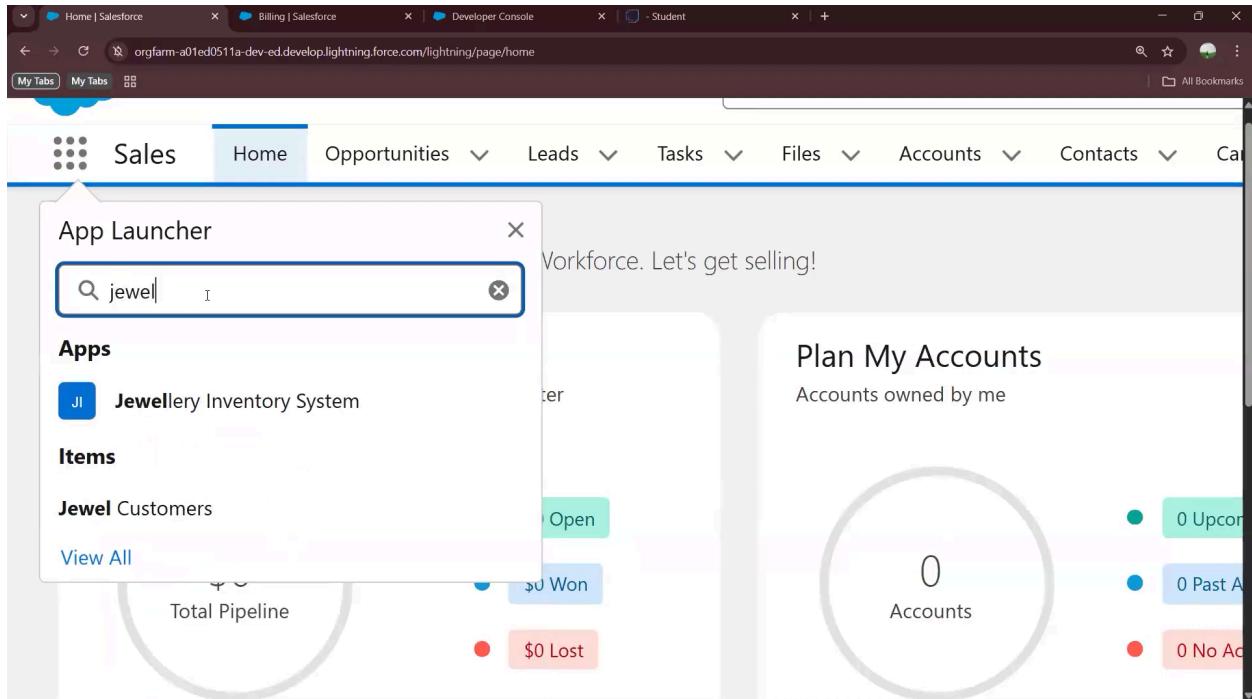
❖ Create An Apex Class

The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Home | Salesforce, Billing | Salesforce, and Developer Console. Below the tabs, the main area displays the code for `UpdatePaidAmountTriggerHandler.apxc`. The code defines a public class `UpdatePaidAmountTriggerHandler` with two static void methods: `handleBeforeInsert` and `handleBeforeUpdate`. The `handleBeforeInsert` method iterates through a list of `Billing__c` records and copies the `Paying_Amount__c` value to `Paid_Amount__c`. The `handleBeforeUpdate` method iterates through a map of `Billing__c` records and updates their `Paid_Amount__c` value based on the old value from the map. A warning symbol (!) is present on line 6. The bottom section of the interface shows the `Problems` tab, which lists five errors related to variable names.

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

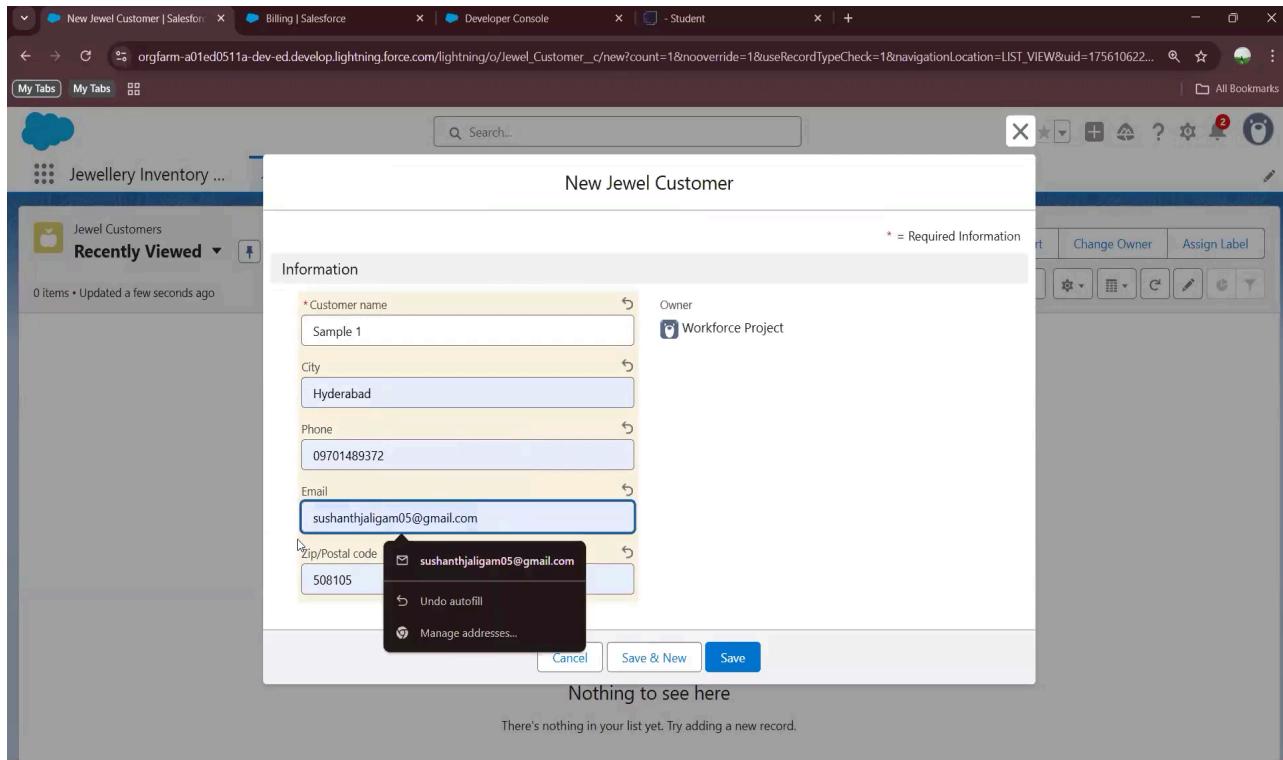
The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for Home | Salesforce, Billing | Salesforce, and Developer Console. Below the tabs, the main area displays the code for `UpdatePaidAmountTrigger.apxt`. The code defines a trigger `UpdatePaidAmountTrigger` on the `Billing__c` object. It uses the `before insert` and `before update` events. For insertions, it calls the `handleBeforeInsert` method of the `UpdatePaidAmountTriggerHandler`. For updates, it calls the `handleBeforeUpdate` method. The bottom section of the interface shows the `Problems` tab, which is currently empty.

❖ Go to Home and Search For Jewellery Inventory System



❖ Create New Customer and Give Price

SAKTHIVEL



SK

SAKTHIVEL

The screenshot shows a Salesforce Lightning interface. At the top, there are several tabs: 'Sample 1 | Jewel Customer | Sa...', 'Billing | Salesforce', 'Developer Console', and '- Student'. The main URL is 'orgfarm-a01ed0511a-dev-ed.develop.lightning.force.com/lightning/r/jewel_Customer_c/a0AgK000001syALUA0/view'. Below the tabs is a navigation bar with links: 'Jewellery Inventory ...', 'Jewel Customers', 'Items', 'Prices', 'Customer Orders', 'Billings', 'Reports', and 'Dashboards'. A search bar is at the top right. The main content area displays a customer record for 'Sample 1'. The record includes fields: Customer name (Sample 1), Owner (Workforce Project), City (Hyderabad), Phone (09701489372), Email (sushanthjali@gmail.com), Zip/Postal code (508105). It also shows 'Created By' (Workforce Project) and 'Last Modified By' (Workforce Project). Buttons for 'New Contact', 'Edit', and 'New Opportunity' are at the top right of the record view.

The screenshot shows a Salesforce Lightning interface. The top tabs are 'New Price | Salesforce', 'Billing | Salesforce', 'Developer Console', and '- Student'. The main URL is 'orgfarm-a01ed0511a-dev-ed.develop.lightning.force.com/lightning/o/Price_c/new?count=28&nooverride=1&useRecordTypeCheck=1&navigationLocation=LIST_VIEW&uid=175610647154770858...'. The navigation bar below includes 'Jewellery Inventory ...', 'Jewel Customers', 'Items', 'Prices', 'Customer Orders', 'Billings', 'Reports', and 'Dashboards'. On the left, a sidebar titled 'Recently Viewed' shows '0 items • Updated a few seconds ago'. The main content is a 'New Price' form. It has an 'Information' section with a 'Price Id' field containing '900d' (which is highlighted with a red border and has an error message 'Enter a valid numeric value.' below it). The 'Owner' is listed as 'Workforce Project'. At the bottom of the form are buttons for 'Cancel', 'Save & New', and 'Save' (the 'Save' button is highlighted with a blue border). A status message 'Nothing to see here' and a note 'There's nothing in your list yet. Try adding a new record.' are displayed at the bottom of the page.

❖ Check For Price Report

The screenshot shows the Salesforce Lightning interface. A green success message at the top right says "Price 'PD-0001' was created." The main view displays a single Price record with the following details:

Field	Value
Price Id	PD-0001
Gold Price	\$9,000
Created By	Workforce Project
Last Modified By	Workforce Project

Below the main view, there is a large blue placeholder area.

SAKTHI

The screenshot shows the Salesforce Lightning interface displaying a report titled "Report: Prices Price Report". The report summary indicates 1 Total Record and \$9,000 Total Gold Price. The report table lists the following data:

	Price: ID	Gold Price	Price: Price Id
1	a08gK0000089nsv	\$9,000	PD-0001
2		\$9,000	

SAKTHIVEL

❖ Create New Automation and Start

The screenshot shows the 'New Automation' screen in the Flow Builder. At the top, there's a search bar with the placeholder 'search automations...'. Below it, there are four categories: 'Triggered', 'Scheduled', 'Screen', and 'Autolaunched'. Each category has a brief description and a 'View All' link. Under 'Frequently Used', there are four examples: 'Record-Triggered Flow', 'Screen Flow', 'Schedule-Triggered Flow', and 'Autolaunched Flow (No Trigger)'. A 'View All' link is also present here. At the bottom left, there's a link 'Get more on the AppExchange'.

The screenshot shows a specific automation flow named 'Billing Alert Flow - V1'. The status is 'Inactive'. The flow starts with a 'Record-Triggered Flow' trigger (Object: Billing, Trigger: A record is created or updated). This triggers an 'Action' step (notice), which then has a 'Run Immediately' option. The flow builder interface includes tabs for 'My Tabs' and 'Flow Builder', and various buttons like 'Run', 'Debug', 'View Tests', 'Save As New Version', 'Save', and 'Activate'.

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
}

```

```

public static 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;
}

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

// Record Sale
public static 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;
}

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

```

APEX TEST CLASS :

```
@isTest
public class JewelCRMManagerTest {
    @isTest
    static void testCRMFlow() {
        // Add customer
        Id custId = JewelCRMManager.addCustomer('Sakthivel', '7845770375', 'alice@test.com');

        // Add jewel
        Id jewelId = JewelCRMManager.addJewel('Gold Ring', 'Ring', 15000, 10);

        // Record sale
        Id saleId = JewelCRMManager.addSale(custId, jewelId, 2, Date.today());

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

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