

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

**NAAN MUDHALVAN PROJECT REPORT**

**Submitted by**

<b>A.VANASOFY</b>	<b>(912422104054)</b>
<b>S.SWETHA</b>	<b>(912422104049)</b>
<b>SM.SWETHA</b>	<b>(912422104050)</b>
<b>T.SWETHA</b>	<b>(912422104051)</b>

**DEPARTMENT OF**  
**COMPUTER SCIENCE AND ENGINEERING**



**SHANMUGANATHAN ENGINEERING COLLEGE**  
**ARASAMPATTI – 622 507**

## **BONAFIED CERTIFICATE**

Certified that this project report “**CRM APPLICATION FOR JEWEL MANAGEMENT-(DEVELOPER)**” is the bonafide work of “**A.VANASOFY(912422104054), S. SWETHA(912422104049),SM. SWETHA(912422104050),T. SWETHA(912422104051)**” who carried out the mini project Work under my supervision.

### **COURSE COORDINATOR,**

Mrs.S.Vinodha,M.E,  
Assistant Professor,  
Department of CSE,  
Shanmuganathan Engineering College,  
Arasampatti.

### **HEAD OF THE DEPARTMENT,**

Mr.S.Saravanakumar ,M.E,  
Assistant Professor,  
Department of CSE,  
Shanmuganathan Engineering College,  
Arasampatti.

### **STAFF IN-CHARGE**

### **HEAD OF THE DEPARTMENT**

**Submitted for Viva-Voce Examination held on \_\_\_\_\_**

### **INTERNAL EXAMINER**

### **EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

With warm hearts and immense pleasure, we thank the almighty for his grace and blessing bestowed on us, which drove us to the successful completion of this project. We take this opportunity to express our sincere thanks to the significant contribution given by the management of our College Chairman "**KALVI VALLAL Mrs.Pichappa Valliammai**", who is guiding light for all activities in our college.

We express our sincere and humble tone of thanks to our Correspondent **Dr. P.Manikandan** we would like to thank the management of the college for providing us with all facilities needed for the successful completion of our work.

We also extend our heartfelt gratitude to our respected Secretary **Mr.Vishwanathan** who kindly permitted us to go ahead with our project.

We express our sincere and humble tone of thanks to our energetic Principal **Dr.KL.MuthuramuM.E.,(W.R),M.E.,(S.E),M.I.S.T.E.,FIE.,ph.d.,** we would like to thank the management of the college for providing us with all facilities needed for the successful completion of our work.

We would like to thank all the staff members of CSE department who had provided their contribution to steer our project towards the successful completion. Thank our Head of the Department **Mr.S.Saravanakumar,M.E,** Head of the Department, Department of Computer Science and Engineering, for her cooperation, advice and suggestions at every stage of our project work.

We also wish to acknowledge with thanks the cooperation, assistance, and encouragement extended by **Mrs.S.Vinodha,M.E,A/P** of the CSE department who contributed their time and expertise during the development of our project.

We also convey our humble Finally , we thank our family members who have spent their wealth for our studies and have motivated us to complete this project successfully.

<b>S.NO</b>	<b>TITLE</b>	<b>PG.NO</b>
1.	Abstract	1
2.	Introduction	2
3.	Objectives	3
4.	System Requirements	4
5.	Modules of the system	4
6.	Technology used	9
7.	System Design	9
8.	Workflow Description	10
9.	Implementation Steps	13
10.	Expected Outcomes	28
11.	Advantages	28
12.	Future Enhancements	28
13.	Conclusion	29
14.	References	29

## 1. ABSTRACT

The **Jewel Management System** is a comprehensive **Salesforce-based cloud application** designed to streamline and automate the operations of a jewelry business. This system eliminates the limitations of manual record maintenance by providing an integrated platform to manage customers, items, prices, and billing details in a structured and efficient manner. Built entirely using Salesforce's **declarative development features**, such as *Object Manager*, *App Builder*, and *Flow Builder*, the project highlights the power of low-code tools to deliver enterprise-level solutions without writing complex code.

The system architecture is based on multiple interconnected custom objects—**Jewel Customer, Item, Price, and Billing**—each serving a unique purpose within the overall business process. These objects are linked through **lookup relationships**, enabling seamless data flow and ensuring that all records remain consistent and accurate. For example, when a new item is created or its price changes, the related billing record automatically reflects the updates, maintaining data integrity across the platform.

To enhance automation and user experience, the project incorporates **Record-Triggered Flows** that execute predefined actions when specific conditions are met. This eliminates the need for manual updates, reduces human error, and improves operational efficiency. Additionally, the system leverages Salesforce's **cloud infrastructure**, providing accessibility from anywhere, automatic data backups, and real-time synchronization across users and devices.

Overall, the Jewel Management System demonstrates how **cloud-based CRM solutions** can digitally transform small and medium enterprises by reducing manual work, increasing accuracy, and ensuring scalability. The project reflects the principles of modern software engineering—automation, integration, and usability—implemented through Salesforce's powerful yet user-friendly platform. This application serves as a foundation for future enhancements like inventory tracking, sales analytics, and customer relationship optimization, showing the real-world impact of adopting **low-code technology** for business process automation.

## 2. INTRODUCTION

In today's digital era, businesses are rapidly shifting from traditional management systems to cloud-based platforms to improve efficiency, accuracy, and customer satisfaction. The jewelry industry, in particular, deals with complex operations such as maintaining customer information, tracking product details, managing pricing updates, and generating invoices. Managing these operations manually often leads to errors, delays, and poor data organization. To overcome these challenges, the **Jewel Management System** was developed using **Salesforce**, a leading cloud-based Customer Relationship Management (CRM) platform.

The Jewel Management System aims to simplify and automate the day-to-day activities of a jewelry business by utilizing Salesforce's powerful declarative tools. It enables seamless management of **customers, items, prices, and billing records** within a single integrated system. The project demonstrates how Salesforce's low-code environment allows users to design, implement, and automate business processes without extensive programming knowledge.

By using tools such as **Object Manager** and **Flow Builder**, the system provides a structured approach to data organization and automation. The Object Manager facilitates the creation of custom objects and relationships, while Flow Builder automates tasks like updating billing records when a new item or customer is added. These features together ensure **data consistency, accuracy, and real-time synchronization**, which are critical for maintaining business reliability.

The project also showcases Salesforce's adaptability for small and medium enterprises (SMEs) that seek affordable and efficient solutions for digital transformation. With its cloud infrastructure, the system offers benefits such as **remote accessibility, secure data storage, scalability, and reduced maintenance effort**.

Overall, this project serves as an example of how technology can transform traditional business models into modern, automated systems. The **Jewel Management System** not only enhances business efficiency but also sets the foundation for future developments like sales analytics, inventory management, and customer loyalty programs. It illustrates the potential

of CRM-based applications in improving decision-making and fostering growth in the jewelry sector.

### 3. OBJECTIVES

The primary objective of the **Jewel Management System** is to design and implement a **Salesforce-based cloud solution** that simplifies and automates the operations of a jewelry business. The project focuses on replacing manual processes with an efficient, accurate, and easily accessible digital system using Salesforce's declarative tools.

The following are the key objectives of the project:

**1. To Automate Jewelry Business Operations:**

The system aims to reduce manual work by automating core processes such as maintaining customer details, updating item prices, and generating billing information through record-triggered flows.

**2. To Ensure Data Accuracy and Consistency:**

By establishing relationships between custom objects like Jewel Customer, Item, Price, and Billing, the system ensures that all related data remains synchronized and error-free across all records.

**3. To Utilize Salesforce Declarative Tools:**

The project demonstrates the use of Salesforce tools such as **Object Manager**, **App Builder**, and **Flow Builder** to design a fully functional application without traditional programming.

**4. To Improve Accessibility and Efficiency:**

Being cloud-based, the Jewel Management System allows users to access business information from anywhere at any time, ensuring better data availability and faster decision-making.

**5. To Provide a Scalable and Secure Platform:**

The system leverages Salesforce's secure and scalable infrastructure to handle growing data volumes and ensure data protection, reliability, and backup.

**6. To Facilitate Easy Customization and Maintenance:**

Salesforce's low-code environment enables easy updates, additions, or modifications to business logic without requiring extensive technical knowledge.

**7. To Demonstrate Digital Transformation Using CRM Tools:**

The project showcases how small and medium businesses can adopt CRM technology to improve efficiency, automate repetitive tasks, and enhance overall customer experience.

## **4. SYSTEM REQUIREMENTS**

### **4.1. Software Requirements**

- Salesforce Developer Edition
- Trailhead Playground
- Google Chrome / Microsoft Edge
- Windows or macOS Operating System
- Internet Connectivity

### **4.2. Hardware Requirements**

- Processor: Intel i3 or higher
- RAM: Minimum 4 GB (recommended 8 GB)
- Storage: Minimum 10 GB
- Stable Internet Connection

## **5. MODULES OF THE SYSTEM**

### **5.1. Setup & Configuration Module**

#### **5.1.1. Creation of Object**

- In Salesforce, **Objects** are like **database tables** that store specific business data.

- They hold information related to entities such as *Customers, Items, or Orders*.
- Salesforce includes **Standard Objects** (like Account, Contact) and allows the creation of **Custom Objects** as per project requirements.
- **Example:** A custom object named **Jewelry\_Item** was created to store jewelry details such as **Material Type, Weight, and Price** for each product.

#### **5.1.2. Creation of Fields**

- **Fields** represent individual pieces of data within an object, similar to columns in a table.
- Salesforce provides both **Standard Fields** (ID, Owner, Created Date) and **Custom Fields** that can be added manually.
- Common field types include **Text, Number, Picklist, Currency, Date, Formula, and Lookup**.
- **Example:** Custom fields like **Gold Purity, Weight, and Price** were added to the *Jewelry\_Item* object to record specific product information.

#### **5.1.3. Creation of Tabs**

- **Tabs** provide an easy way to access object records directly from the app navigation bar.
- When a tab is created for an object, users can open, view, and manage its records conveniently.
- **Example:** A tab was created for *Jewelry\_Item* so users can quickly access jewelry details in the app.

#### **5.1.4. Creation of Lightning App**

- **A Lightning App** is a customized workspace in Salesforce that combines multiple objects, tabs, and dashboards into one application.
- It helps users focus only on the tools relevant to their business tasks.

- **Example:** A “Jewelry Management App” was created containing:
  - Objects: *Jewelry Items, Customers, Orders*
  - Dashboards: *Sales Insights*
  - Tabs: *Reports, Settings*

#### **5.1.5. Creation of Page Layouts**

- **Page Layouts** control the design and visibility of fields, sections, buttons, and related lists on record pages.
- They determine what information users see when they create or view a record.
- **Example:** Different layouts were designed for **Retail Jewelry** and **Wholesale Jewelry** to make forms more user-friendly.

#### **5.1.6. Creation of Record Types**

- **Record Types** help manage multiple business processes within the same object.
- They allow different page layouts, picklist values, and workflows to be assigned based on record type.
- **Example:** For the *Jewelry\_Item* object:
  - Record Type 1: **Gold Jewelry**
  - Record Type 2: **Silver Jewelry**

### **5.2. Security & Access Control Module**

#### **5.2.1. Creation of Profiles**

- A **Profile** defines what a user can do in Salesforce (e.g., Create, View, Edit, Delete).
- Every user must have one profile that controls their access and permissions.

- **Example:**
  - *Admin Profile* – Full access to all objects and settings.
  - *Sales Profile* – Limited access to only customer and item records.

### 5.2.2. Creation of Roles

- A **Role** defines **data visibility** — what information a user can view within the organization.
- Roles help structure data sharing according to hierarchy.
- **Example:**
  - *Manager* – Can view all team records.
  - *Salesperson* – Can only view their own records.

### 5.2.3. Creation of Permission Sets

- **Permission Sets** give additional access to users without changing their profile.
- They are optional but useful when temporary or extra permissions are needed.
- **Example:** A *Permission Set* was created to give specific users access to **Reports** and **Dashboards**.

### 5.2.4. Creation of User

- A **User** represents a person who logs in to Salesforce.
- Each user is assigned a **Profile**, **Role**, and optional **Permission Sets**.
- **Example:** Creating a new user account for a sales employee to manage customer records in the system.

## **5.3. Reporting & Analytics Module**

### **5.3.1. Creation of Reports**

- A **Report** provides a detailed and filtered view of stored data.
- It helps analyze trends, sales, and business performance using various filters and summaries.
- **Example:** A **Sales Report** showing total jewelry sales per month was created for business insights.

### **5.3.2. Creation of Dashboards**

- A **Dashboard** is a visual representation of data collected from reports.
- It uses **charts, graphs, and metrics** for quick decision-making and tracking performance.
- **Example:** A dashboard displaying **Top-Selling Jewelry Items** and **Monthly Revenue** was designed for management review.

## **5.4. Automation & Process Module**

### **5.4.1. Creation of Flow**

- A **Flow** is Salesforce's automation tool used to perform business processes without coding.
- It can create, update, or delete records automatically and perform logical operations.
- **Example:** When a new jewelry order is created, a **Flow** automatically updates the related billing record and sends a confirmation email to the customer.

## **5.5. User Management & Adoption Module**

### **5.5.1. Creation of User Adoption**

- **User Adoption** measures how effectively users are engaging with the Salesforce system.

- It tracks login frequency, feature usage, and data entry activity.
- Tools like **User Adoption Reports**, **Dashboards**, and **In-App Guidance** help improve engagement.
- **Example:** A dashboard was built to monitor which employees actively log in and enter sales data regularly.

## 6. TECHNOLOGIES USED

- Salesforce Lightning Platform
- Metadata-Driven Architecture
- Low-Code Tools: Flow Builder, App Builder, Page Layout Editor
- Security Framework: Profiles, Roles, Permission Sets
- Analytics Tools: Report Builder, Dashboard Builder
- Schema Tools: Object Manager, Schema Builder

## 7. SYSTEM DESIGN

### 7.1 Design Approach:

The project follows a **modular and relational design** using Salesforce custom objects. Each object (Customer, Item, Price, Billing) handles a specific part of the jewelry business process and is connected using lookup relationships.

### 7.2 Architecture Used:

The system is based on the **Salesforce cloud architecture**, which includes:

- **User Interface Layer:** Salesforce Lightning App and Tabs for users to interact with data.

- **Application Layer:** Declarative tools like Object Manager and Flow Builder to handle logic.
- **Database Layer:** Salesforce database that stores all object records securely in the cloud.

### **7.3 Data Flow:**

- The user enters customer and item details.
- Item prices are stored in the Price object.
- When a new billing record is created or updated, the **Flow** automatically calculates and updates the billing amount.
- All related data (Customer → Item → Billing) stays connected and updated.

### **7.4 Key Design Tools Used:**

- **Object Manager** – for creating and managing objects and fields.
- **Flow Builder** – for automation and record-triggered actions.
- **App Builder** – for creating the custom app and page layouts.
- **Schema Builder** – for visualizing relationships among objects.

## **8. WORKFLOW DESCRIPTION**

The **workflow** of the Jewel Management System explains how data moves and actions take place inside the Salesforce application — from entering customer details to generating the final billing record. It shows the step-by-step process that keeps all modules connected and automated.

### **Step 1: Customer Information Entry**

- The process begins when a new **customer** is added using the *Jewel Customer* object.
- Details like customer name, address, state, country, and postal code are entered and saved in Salesforce.

## **Step 2: Adding Item Details**

- After creating a customer, the next step is to add **item details** in the *Item* object.
- Each item record is linked to a specific customer using a **lookup relationship**.
- Information like item name, type, and category is stored here.

## **Step 3: Setting the Price**

- The **Price** object is used to define the cost of jewelry items such as silver or gold rates.
- These prices are stored using the currency data type and can be updated whenever rates change.

## **Step 4: Generating Billing Record**

- Once the customer, item, and price data are available, the **Billing** record is created.
- It includes details like customer name, item name, and total amount.
- This record connects all three objects to generate accurate billing information.

## **Step 5: Automation using Flow**

- A **Record-Triggered Flow** is created for the Billing object.
- Whenever a billing record is created or updated, the flow automatically fetches related item and price data.
- It calculates the total billing amount and updates the record without manual effort.

## **Step 6: Data Update and Storage**

- All records (Customer, Item, Price, Billing) are stored securely in the Salesforce cloud.
- Any update made in one object (like price change) automatically reflects in the related billing record through the flow.

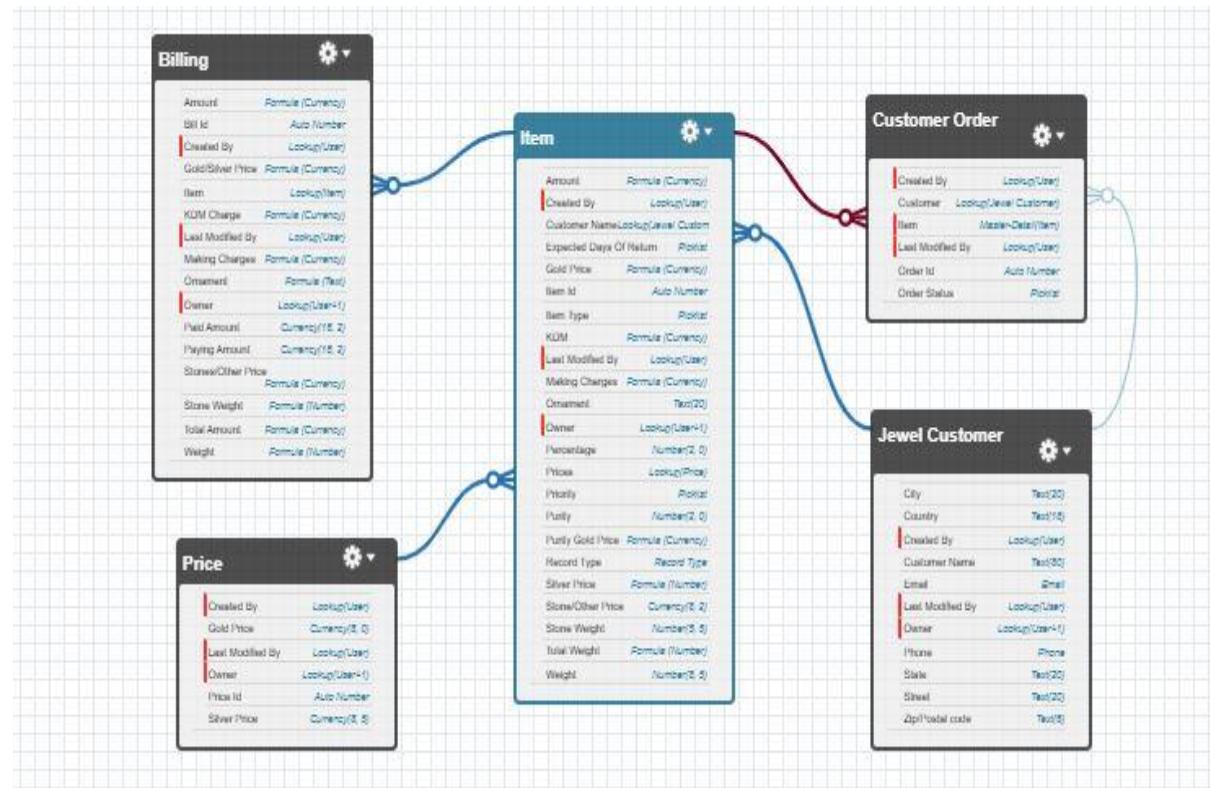
## Step 7: Viewing and Managing Data

- Users can view all related records through tabs and page layouts.
- The **App Builder** ensures that all modules are easily accessible within a single application view.

## Step 8: Reporting & Dashboards

- Management can view real-time sales reports, stock availability, and customer trends.
- Dashboards visually show KPIs and performance metrics to support business decisions.

### 8.1 WORKFLOW DIAGRAM



## 9. IMPLEMENTATION PROCESS

### 1. SALESFORCE ACCOUNT CREATION

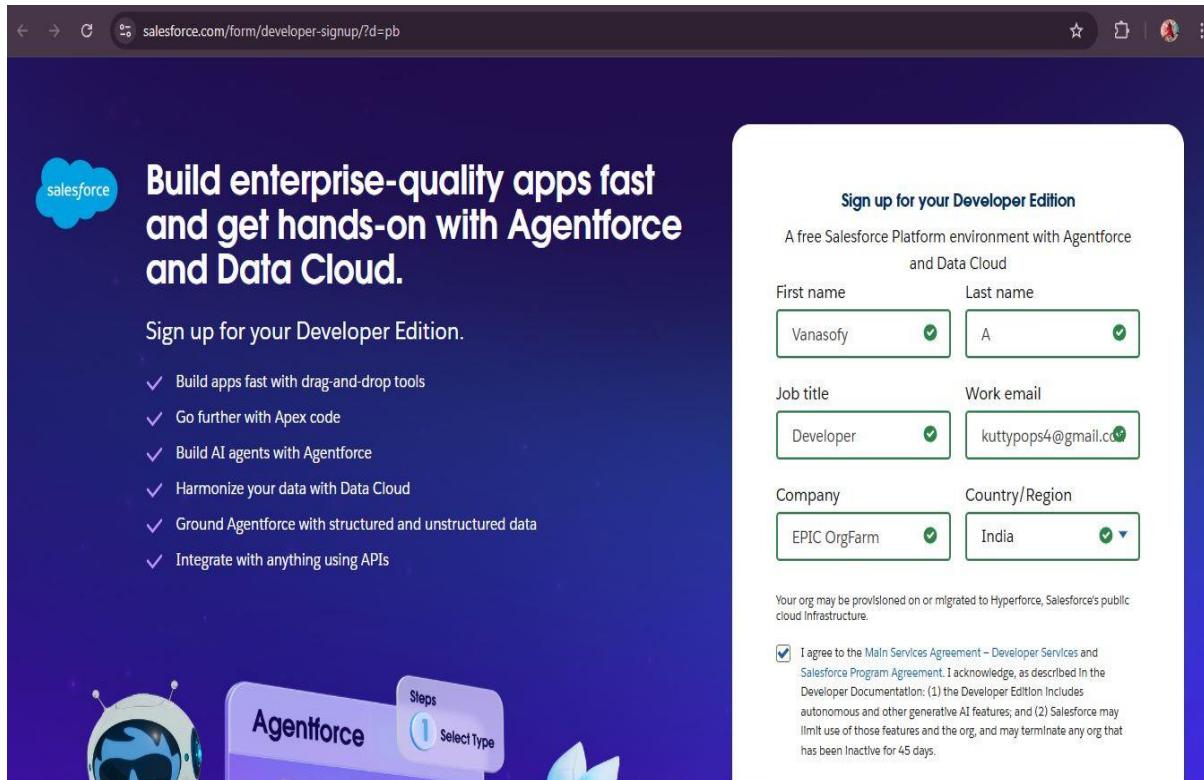


Fig 1.1 Sign up for salesforce developer

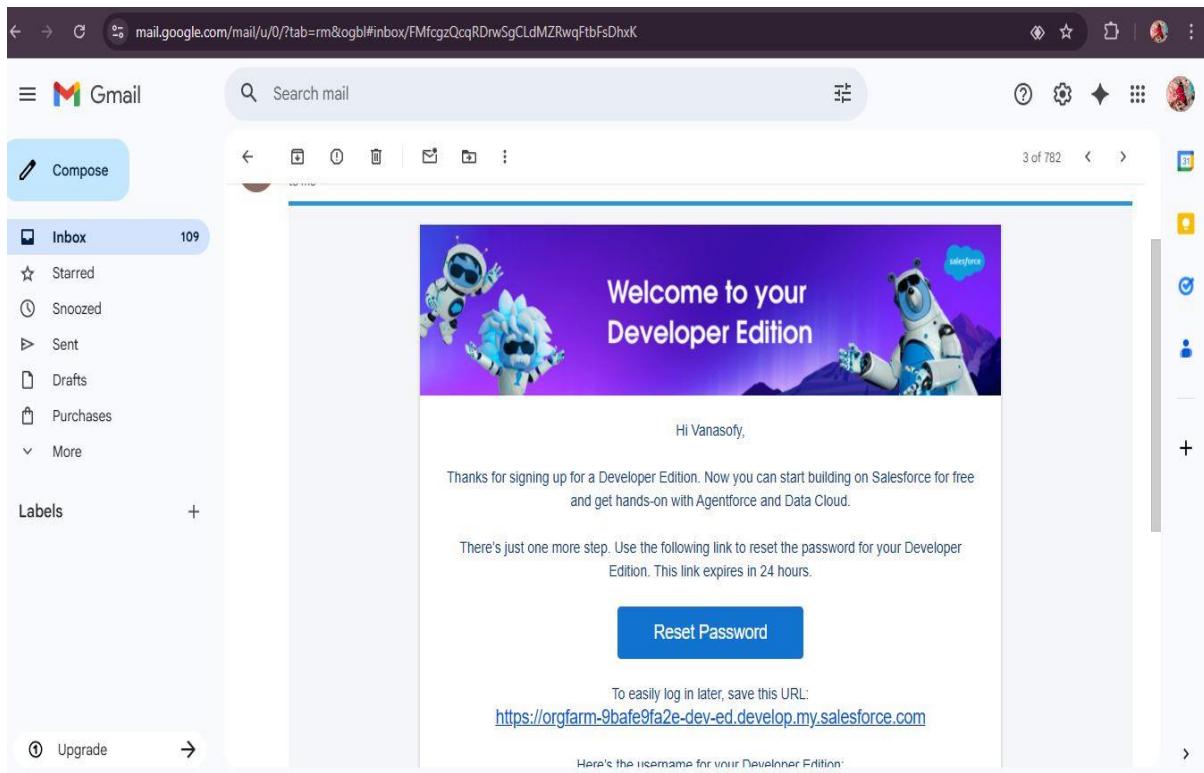


Fig 1.2 Activation of salesforce account

## 2. OBJECT CREATION

The screenshot displays two screenshots of the Salesforce Setup interface, illustrating the creation and configuration of custom objects.

**Screenshot 1: Custom Object Creation - Jewel Customer**

This screenshot shows the "Edit Custom Object" page for "Jewel Customer".

- Custom Object Definition Edit:**
  - Custom Object Information:** Label: "Jewel Customer", Plural Label: "Jewel Customers". A note states: "The singular and plural labels are used in tabs, page layouts, and reports. Be careful when changing the name or label as it may affect existing integrations and merge templates."
  - Object Name:** "Jewel\_Customer" (API Name).
  - Description:** An empty text area.
  - Context-Sensitive Help Setting:** Radio buttons for "Open the standard Salesforce.com Help & Training window" (selected) and "Open a window using a Visualforce page".
- Details Sidebar:** Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts.

**Screenshot 2: Object Manager - Item**

This screenshot shows the "Details" page for the "Item" object.

- Details Sidebar:** Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts.
- Details Panel:**
  - Description:** An empty text area.
  - API Name:** "Item\_c" (Custom).
  - Singular Label:** "Item".
  - Plural Label:** "Items".
  - Enable Reports:** Checked.
  - Track Activities:** Checked.
  - Track Field History:** Checked.
  - Deployment Status:** "Deployed".
  - Help Settings:** "Standard salesforce.com Help Window".

Fig 2.1 Customer object for item & some of the objects

### 3. CREATING CUSTOM OBJECT TABS

The screenshot shows the Salesforce Setup page for creating Custom Tabs. The left sidebar has a search bar and navigation links like Setup, Home, and Object Manager. Under 'Tableau', there's a link to 'Tabs'. The main area is titled 'Custom Tabs' with a sub-section 'Custom Object Tabs'. It shows a table with five rows:

Action	Label	Tab Style	Description
Edit   Del	Billing	Jewel	
Edit   Del	Customer Orders	Jewel	
Edit   Del	Items	Jewel	
Edit   Del	Jewel Customers	Jewel	
Edit   Del	Prices	Jewel	

Below this is a section for 'Web Tabs' which says 'No Web Tabs have been defined'.

Fig 3.1 Creating custom tabs for billing, customer order, items, jewel customers, prices.

### 4. CREATING A LIGHTNING APP

The screenshot shows the Lightning App Builder interface for creating a 'Jewellery Inventory System'. The left sidebar has 'App Settings' and 'App Details & Branding' selected. In the main area, under 'App Details & Branding', there are sections for 'App Details' and 'App Branding'. The 'App Details' section includes fields for 'App Name' (Jewellery Inventory System) and 'Developer Name' (Jewellery.Inventory\_System). The 'App Branding' section includes an 'Image' field with a diamond icon and a 'Primary Color Hex Value' field set to #0070D2. There's also an 'Org Theme Options' checkbox and an 'App Launcher Preview' section showing the app icon and name.

Fig 4.1 Lightning app for CRM application for jewel management-(developer).

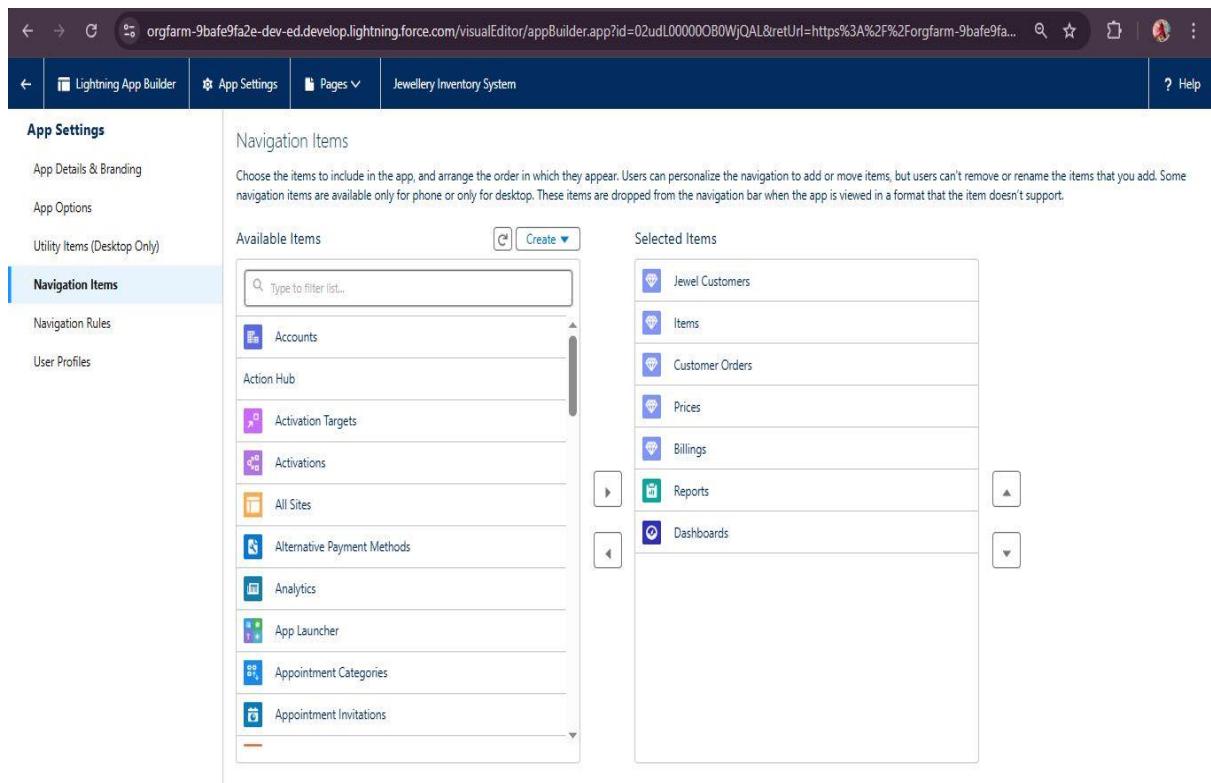


Fig 4.2 Navigation items for the application.

## 5. CREATION OF FIELDS IN OBJECTS

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(Jewel Customer)	✓	▼
Item	Item__c	Master-Detail(Item)	✓	▼
Last Modified By	LastModifiedById	Lookup(User)		
Order Id	Name	Auto Number	✓	▼
Order Status	Order_Status__c	Picklist		▼

Fig 5.1 Fields of customer, item, order, order status for customer order.

## 6. CREATE SCHEMA BUILDER

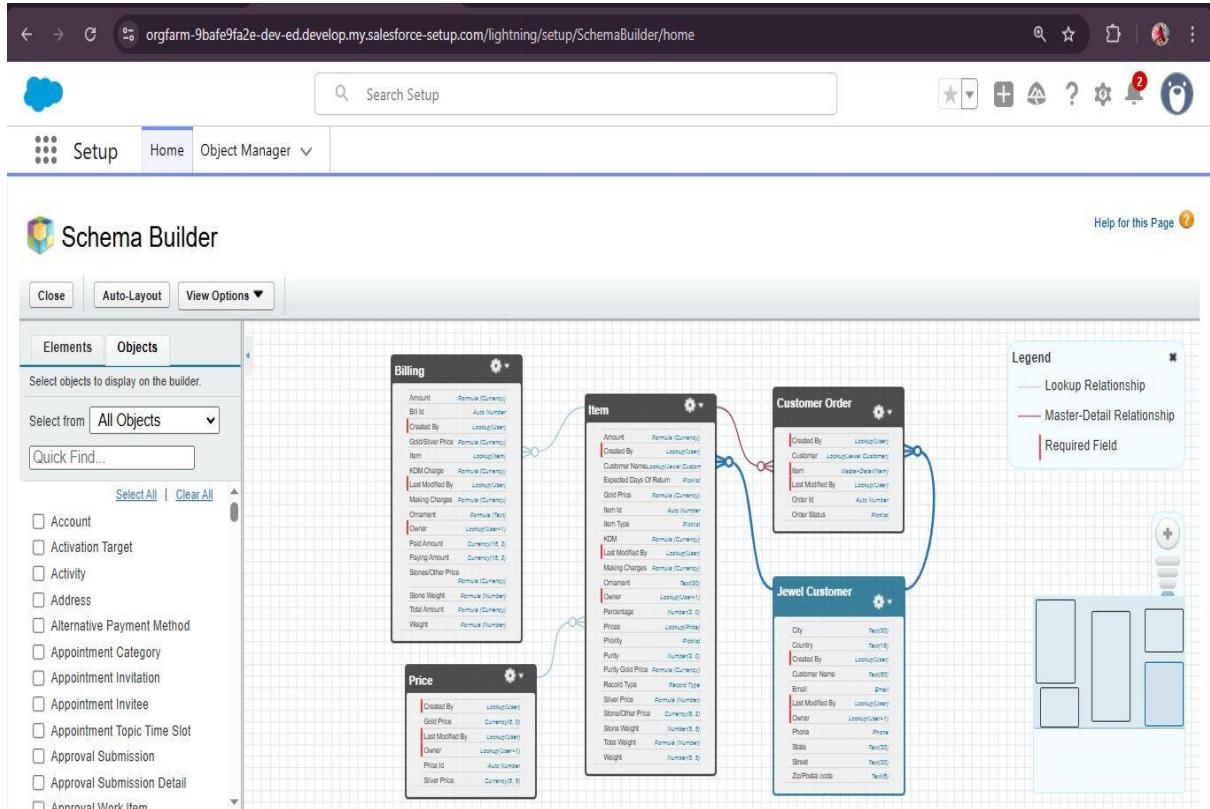


Fig 6.1 Schema builder for the application.

## 7. CREATING PROFILES

Standard Object Layouts	Location Group Assignment
Global Global Layout [View Assignment]	Location Group Assignment Layout [View Assignment]
Email Application Not Assigned [View Assignment]	Macro Macro Layout [View Assignment]
Home Page Layout Home Page Default	Object Milestone Object Milestone Layout

Fig 7.1 Setting up profile for gold smith and worker

The screenshot shows the Salesforce Setup interface. The top navigation bar includes links for Home and Object Manager. On the left, there's a sidebar with a search bar labeled "Search Setup" and sections for "Users" and "Profiles". The main content area is titled "SETUP Profiles" and displays information about the "Standard Platform User". It includes a brief description, a list of permissions (e.g., Login IP Ranges, Enabled Apex Class Access, etc.), and a "Profile Detail" section with fields for Name, User License, and Created By. Below this is a "Page Layouts" section showing standard object layouts like Global, Email Application, and Home Page Layout, each with their respective global and lead layouts.

Standard Platform User

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Log In IP Ranges | Enabled Apex Class Access | Enabled Visualforce Page Access | Enabled External Data Source Access | Enabled Named Credential Access | Enabled External Credential Principal Access | Enabled Custom Metadata Type Access | Enabled Custom Setting Definitions Access | Enabled Flow Access | Enabled Service Presence Status Access | Enabled Custom Permissions

**Profile Detail**

Name: Standard Platform User

User License: Salesforce Platform

Created By: salesforce.com, inc., 10/19/2025, 2:35 PM

Modified By: Yanasofy\_A, 10/25/2025, 10:15 PM

**Page Layouts**

Standard Object Layouts	Global	Lead
	Global Layout [View Assignment]	Lead Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Location [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Location Group [View Assignment]

Fig 7.2 Setting up profile for gold smith and worker in standard platform user

## 8. CREATING ROLES

Fig 8.1 Creating rule hierarchy for gold smith and worker

## 9. CREATING USERS

Action	Full Name	Alias	Username	Role	Active	Profile
<a href="#">Edit</a>	A_Vanasofy	kut	kuttyoops4869@agentforce.com		<input checked="" type="checkbox"/>	System Administrator
<a href="#">Edit</a>	Chatter Expert	Chatter	shatty00dd00000fmcevac.wjownu9wdyo@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<a href="#">Edit</a>	EPIC_OrgFarm	OEPIC	epic.9846c059656e@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<a href="#">Edit</a>	Mikaelson_Elijah	emika	emika@jewelapp.com	Worker	<input checked="" type="checkbox"/>	Worker
<a href="#">Edit</a>	Mikaelson_Kol	kmika	kol@jewelapp.com	Worker	<input checked="" type="checkbox"/>	Worker
<a href="#">Edit</a>	Mikaelson_Niklaus	nmika	niklaus@jewelapp.com	Gold Smith	<input checked="" type="checkbox"/>	Gold Smith
<a href="#">Edit</a>	User_Integration	integ	integration@00dd00000fmcevac.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<a href="#">Edit</a>	User_Security	sec	insightssecurity@00dd00000fmcevac.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

Fig 10.1 Creating users

## 10. CREATING PAGE LAYOUTS

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Item Layout	Vanasofy A, 10/24/2025, 10:37 PM	Vanasofy A, 10/25/2025, 10:15 PM
Page Layout for Gold	Vanasofy A, 10/25/2025, 7:49 AM	Vanasofy A, 10/25/2025, 10:15 PM
Page Layout for Silver	Vanasofy A, 10/25/2025, 8:13 AM	Vanasofy A, 10/25/2025, 10:15 PM

Fig 10.1 Creating page layouts for item, gold, silver

## 11. CREATING A RECORD TYPES

The screenshot shows the Salesforce Object Manager interface under the 'Record Types' section. On the left, a sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc., with 'Record Types' selected. The main area displays a table titled 'Record Types' with two items: 'Gold' and 'Silver'. The table columns are 'RECORD TYPE LABEL', 'DESCRIPTION', 'ACTIVE', and 'MODIFIED BY'. Both records have 'Gold items information' in the description, are marked as active, and were modified by 'Vanasofy A' on 10/25/2025 at 8:06 AM and 8:51 AM respectively.

RECORD TYPE LABEL	DESCRIPTION	ACTIVE	MODIFIED BY
Gold	Gold items information	✓	Vanasofy A, 10/25/2025, 8:06 AM
Silver	Silver items information	✓	Vanasofy A, 10/25/2025, 8:51 AM

Fig 11.1 Creating report types for gold & silver

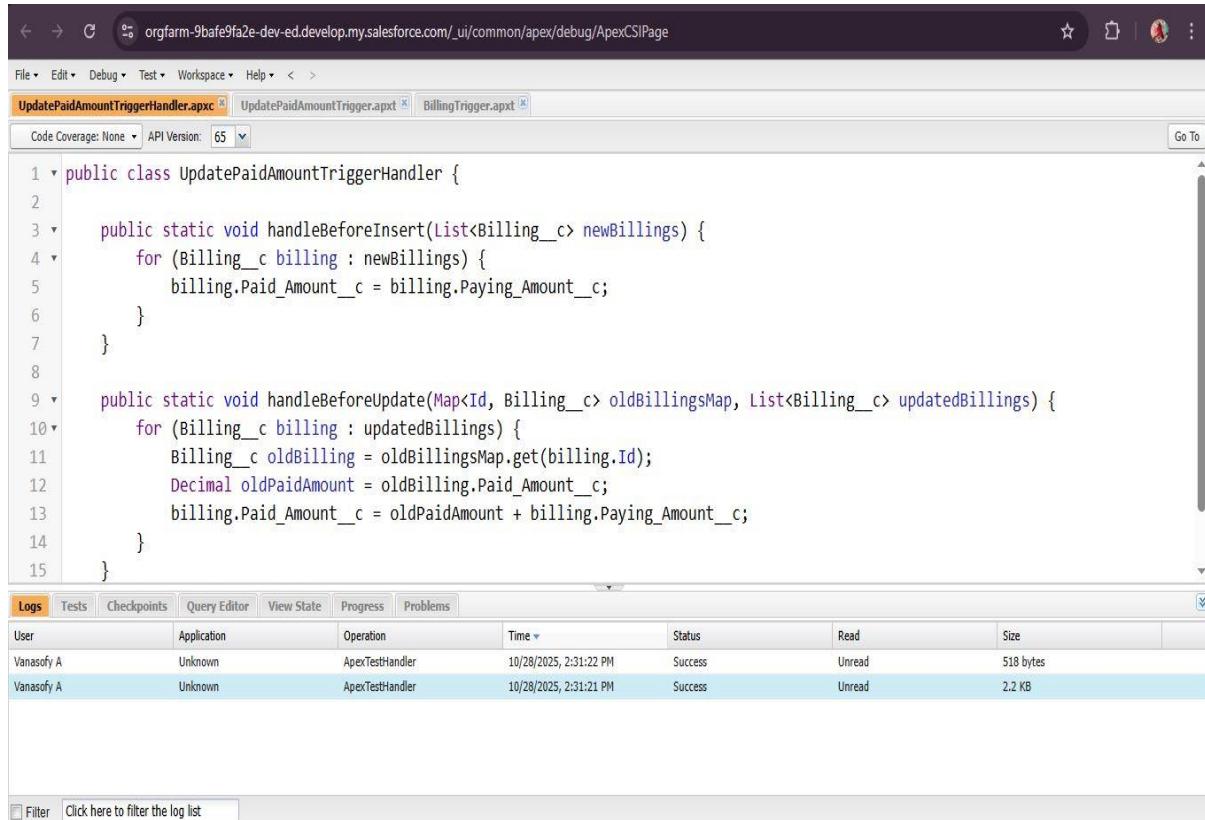
## 12. CREATING PERMISSION SETS

The screenshot shows the Salesforce Permission Sets page for a permission set named 'Per to Worker'. The left sidebar has 'Permission Sets' selected under the 'Users' category. The main area shows the 'Permission Set Overview' for 'Per to Worker', which includes fields for 'Description', 'License', and 'Namespace Prefix'. It also shows 'Manage Assignments' and 'View Summary' buttons. Below this, the 'Apps' section lists several settings: 'Assigned Apps', 'Assigned Connected Apps', 'Object Settings', 'App Permissions', and 'Apex Class Access'.

Description	API Name	Namespace Prefix
Session Activation Required	Per_to_Worker	Created By Vanasofy A, 10/25/2025, 8:20 AM
Permission Set Groups Added To	0	Last Modified By Vanasofy A, 10/25/2025, 8:22 AM

Fig 15.3 Assigning permission sets for per to worker

### 13. DEVELOP APEX CLASS & TRIGGERS



The screenshot shows the Salesforce Apex code editor interface. The top navigation bar includes links for File, Edit, Debug, Test, Workspace, Help, and tabs for UpdatePaidAmountTriggerHandler.apc, UpdatePaidAmountTrigger.apxt, and BillingTrigger.apxt. Below the tabs, there are dropdowns for Code Coverage (None) and API Version (65). The main code area contains the following Apex class:

```

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

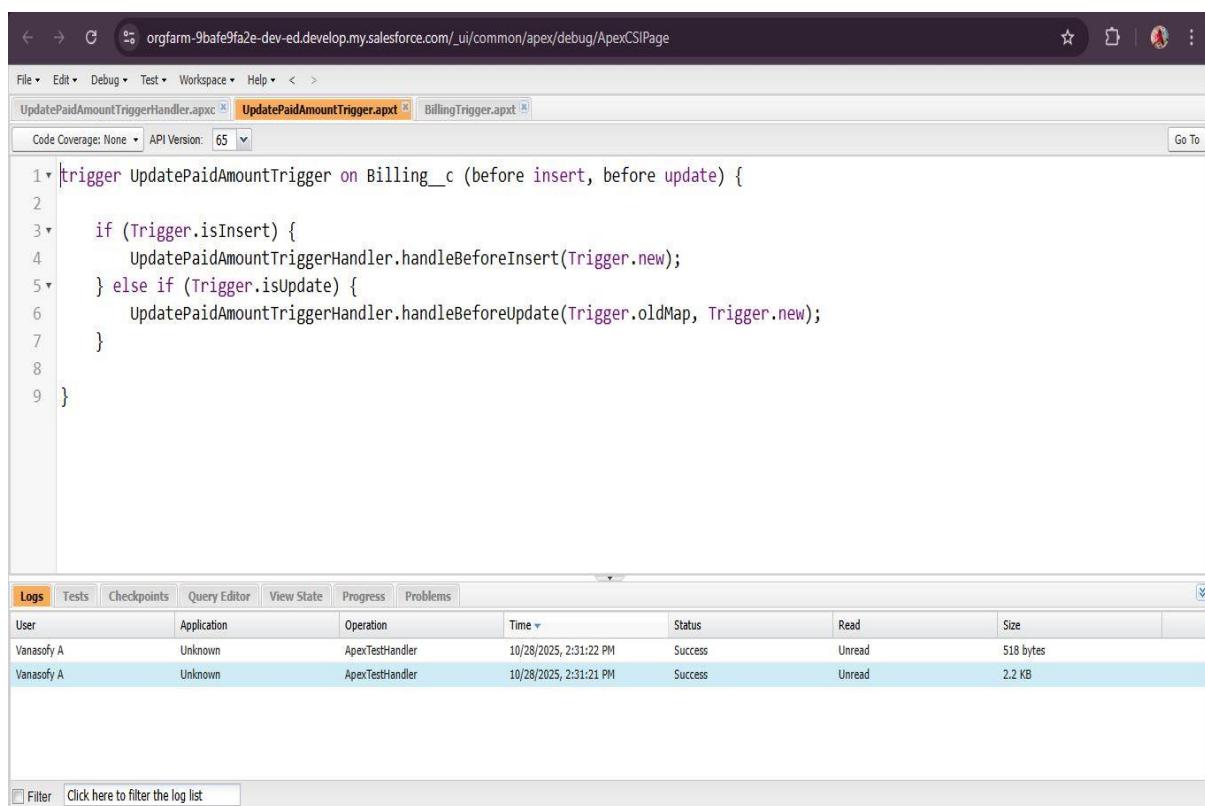
```

Below the code editor is a log viewer with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected, showing two entries from 'Vanasofy A' in the Application column:

User	Application	Operation	Time	Status	Read	Size
Vanasofy A	Unknown	ApexTestHandler	10/28/2025, 2:31:22 PM	Success	Unread	518 bytes
Vanasofy A	Unknown	ApexTestHandler	10/28/2025, 2:31:21 PM	Success	Unread	2.2 KB

At the bottom of the log viewer is a 'Filter' button and a placeholder 'Click here to filter the log list'.

Fig 13.1 Creating apex classes for update paid amount trigger handler



The screenshot shows the Salesforce Apex code editor interface. The top navigation bar includes links for File, Edit, Debug, Test, Workspace, Help, and tabs for UpdatePaidAmountTriggerHandler.apc, UpdatePaidAmountTrigger.apxt, and BillingTrigger.apxt. Below the tabs, there are dropdowns for Code Coverage (None) and API Version (65). The main code area contains the following trigger definition:

```

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

```

Below the code editor is a log viewer with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected, showing two entries from 'Vanasofy A' in the Application column:

User	Application	Operation	Time	Status	Read	Size
Vanasofy A	Unknown	ApexTestHandler	10/28/2025, 2:31:22 PM	Success	Unread	518 bytes
Vanasofy A	Unknown	ApexTestHandler	10/28/2025, 2:31:21 PM	Success	Unread	2.2 KB

At the bottom of the log viewer is a 'Filter' button and a placeholder 'Click here to filter the log list'.

Fig 13.2 Creating apex trigger for update paid amount trigger

```

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

```

The screenshot shows the Salesforce Apex Editor interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and tabs for UpdatePaidAmountTriggerHandler.apxc, UpdatePaidAmountTrigger.apxt, and BillingTrigger.apxt. A status bar at the bottom indicates Code Coverage: None and API Version: 65. Below the editor is a log viewer with tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Logs tab is selected, displaying two entries from the user Vanasofy A:

User	Application	Operation	Time	Status	Read	Size
Vanasofy A	Unknown	ApexTestHandler	10/28/2025, 2:31:22 PM	Success	Unread	518 bytes
Vanasofy A	Unknown	ApexTestHandler	10/28/2025, 2:31:21 PM	Success	Unread	2.2 KB

At the bottom of the log viewer is a filter input field labeled "Click here to filter the log list".

Fig 13.1 Creating apex trigger for billing trigger

## 14. CREATING USER ADOPTION

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes Setup, Home, and Object Manager. The left sidebar features an App Launcher with a search bar containing "jew". The main content area displays a table of objects found, with a search bar "Quick Find" and buttons for "Schema Builder" and "Create". The table has columns for PI NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED.

PI NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Count	Standard Object			
Activity	Standard Object			
Address	Standard Object			
Agent Work	Standard Object			
Alternative Payment Method	Standard Object			
API Anomaly Event Store	Standard Object			
Appointment Category	Standard Object			
Appointment Invitation	Standard Object			
Appointment Invitee	Standard Object			
Appointment Topic Time Slot	Standard Object			
Approval Submission	Standard Object			

Fig 14.1 Opening the applications

**Jewelry Inventory System**

**Jewel Customers**

Recently Viewed

10 items • Updated a few seconds ago

	Customer Name
1	Marcel Gerard
2	Camille O'Connell
3	Davina Claire
4	Hope Mikaelson
5	Freya Mikaelson
6	Finn Mikaelson
7	Rebekah Mikaelson
8	Elijah Mikaelson
9	Niklaus Mikaelson
10	Kol Mikaelson

**Items**

Recently Viewed

10 items • Updated a few seconds ago

	Item Id
1	Item-04
2	Item-05
3	Item-06
4	Item-07
5	Item-10
6	Item-09
7	Item-08
8	Item-03
9	Item-02
10	Item-01

Fig 14.2 Checking the application's tabs like jewel customers & items where the details are entered

The image displays two screenshots of a Salesforce Lightning application interface for a 'Jewellery Inventory System'. Both screenshots show a list view with a header bar containing a cloud icon, a search bar, and various navigation and action buttons.

**Screenshot 1: Customer Orders**

- Header:** orgfarm-9bafef9fa2e-dev-ed.lightning.force.com/lightning/o/Customer\_Order\_c/list?filterName=\_Recent
- Section:** Customer Orders
- Section Header:** Recently Viewed
- List:**
  - 1 Order-10
  - 2 Order-09
  - 3 Order-08
  - 4 Order-07
  - 5 Order-06
  - 6 Order-05
  - 7 Order-04
  - 8 Order-03
  - 9 Order-02
  - 10 Order-01

**Screenshot 2: Prices**

- Header:** orgfarm-9bafef9fa2e-dev-ed.lightning.force.com/lightning/o/Price\_c/list?filterName=\_Recent
- Section:** Prices
- Section Header:** Recently Viewed
- List:**
  - 1 Price-10
  - 2 Price-09
  - 3 Price-08
  - 4 Price-07
  - 5 Price-06
  - 6 Price-05
  - 7 Price-04
  - 8 Price-03
  - 9 Price-02
  - 10 Price-01

Fig 14.3 Checking the application's tabs like customers order & prices where the details are entered

Jewellery Inventory System

Billings

Recently Viewed ▾

10 items • Updated a few seconds ago

	Bill Id
1	<input type="checkbox"/> Bill-13
2	<input type="checkbox"/> Bill-12
3	<input type="checkbox"/> Bill-11
4	<input type="checkbox"/> Bill-10
5	<input type="checkbox"/> Bill-09
6	<input type="checkbox"/> Bill-08
7	<input type="checkbox"/> Bill-07
8	<input type="checkbox"/> Bill-06
9	<input type="checkbox"/> Bill-05
10	<input type="checkbox"/> Bill-04

Fig 14.2 Checking the application's tabs like billing where the details are entered

## 15. CREATING A REPORT

Reports

Recent

3 items

REPORTS	Report Name	Description	Folder	Created By	Created On	Subscribed
Recent	Billings with item and Customer order		Public Reports	Vanasofy A	10/26/2025, 12:37 AM	<input type="checkbox"/>
Created by Me	Item with Billings		Public Reports	Vanasofy A	10/26/2025, 12:39 AM	<input type="checkbox"/>

Fig 15.1 Created & assigned the report types

Joined Report  
Billings with item and Customer order

Item with Billings										Item with Customer Order				
Item with Billings block 1										Item with Customer Order block 1				
Item Id	Item Type	Purity	Priority	Weight	Bill Id	Paying Amount	Making Charges	KDM Charge	Paid Amount	Total Amount	Item Type	Ornament	Weight	Customer Order ID
Subtotal	Count: 1	0		20.00000		\$2,500.00	\$200.00	\$0		\$402	Count: 1		20.00000	
Item-05	Gold	-	High	25.00000	Bill-08	\$18,000.00	\$7,500.00	\$0	\$18,000.00	\$8,100	Gold	Gold Bracelet	25.00000	a02dL00000Pjpm
Subtotal	Count: 1	0		25.00000		\$18,000.00	\$7,500.00	\$0		\$8,100	Count: 1		25.00000	
Item-06	Silver	-	Low	10.00000	Bill-09	\$3,000.00	\$100.00	\$0	\$3,000.00	\$251	Silver	Silver Ring	10.00000	a02dL00000Pjv3l
Subtotal	Count: 1	0		10.00000		\$3,000.00	\$100.00	\$0		\$251	Count: 1		10.00000	
Item-07	Gold	-	Low	18.00000	Bill-10	\$16,000.00	\$5,400.00	\$0	\$16,000.00	\$5,800	Gold	Gold Earrings	18.00000	a02dL00000Pgjc
Subtotal	Count: 1	0		18.00000		\$16,000.00	\$5,400.00	\$0		\$5,800	Count: 1		18.00000	
Item-08	Silver	-	High	22.00000	Bill-11	\$3,500.00	\$220.00	\$0	\$3,500.00	\$472	Silver	Silver Bangles	22.00000	a02dL00000Pjv6b
Subtotal	Count: 1	0		22.00000		\$3,500.00	\$220.00	\$0		\$472	Count: 1		22.00000	
Item-09	Gold	-	Low	40.00000	Bill-12	\$25,000.00	\$12,000.00	\$0	\$25,000.00	\$12,800	Gold	Gold Chain	40.00000	a02dL00000Pjv8c
Subtotal	Count: 1	0		40.00000		\$25,000.00	\$12,000.00	\$0		\$12,800	Count: 1		40.00000	
Item-10	Silver	-	High	12.00000	Bill-13	\$2,000.00	\$120.00	\$0	\$2,000.00	\$221	Silver	Silver Pendant	12.00000	a02dL00000PjyAD
Subtotal	Count: 1	0		12.00000		\$2,000.00	\$120.00	\$0		\$221	Count: 1		12.00000	
Total	Count: 10	44		277.00000		\$121,000.00	\$55,840.00	\$5,631		\$125,785	Count: 10		277.00000	

Row Counts  Detail Rows  Subtotals  Grand Total

Fig 15.1 Assigned details report

## 16. CREATING A DASHBOARDS

Dashboard Name Description Folder Created By Created On Subscribed

Dashboard Name	Description	Folder	Created By	Created On	Subscribed
Gold Dashboard	Private Dashboards	Vanasofy A	10/26/2025, 12:43 AM		

Recent

1 item

Dashboards

Created by Me

Private Dashboards

All Dashboards

FOLDERS

All Folders

Created by Me

Shared with Me

FAVORITES

All Favorites

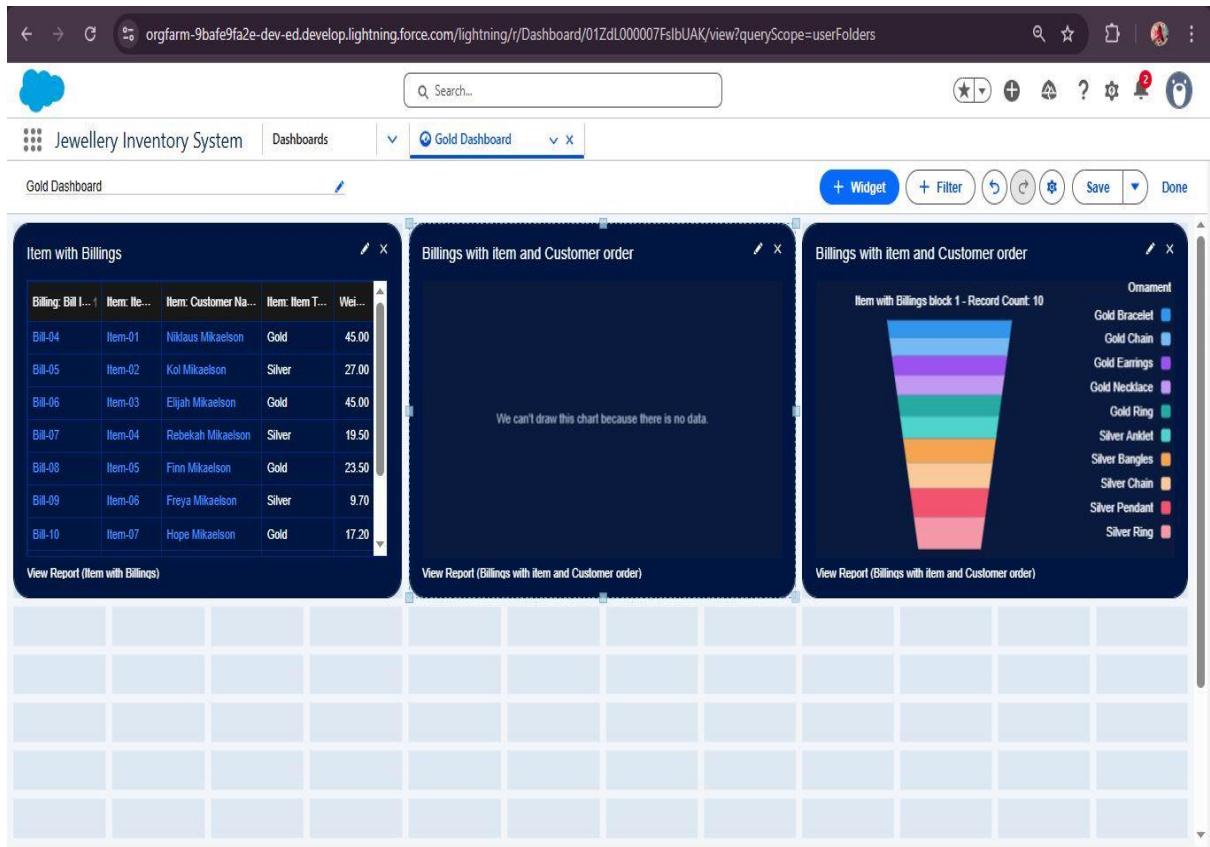


Fig 16.2 CRM application for jewel management-(developer) Dashboard

## 17. CREATING A FLOWS

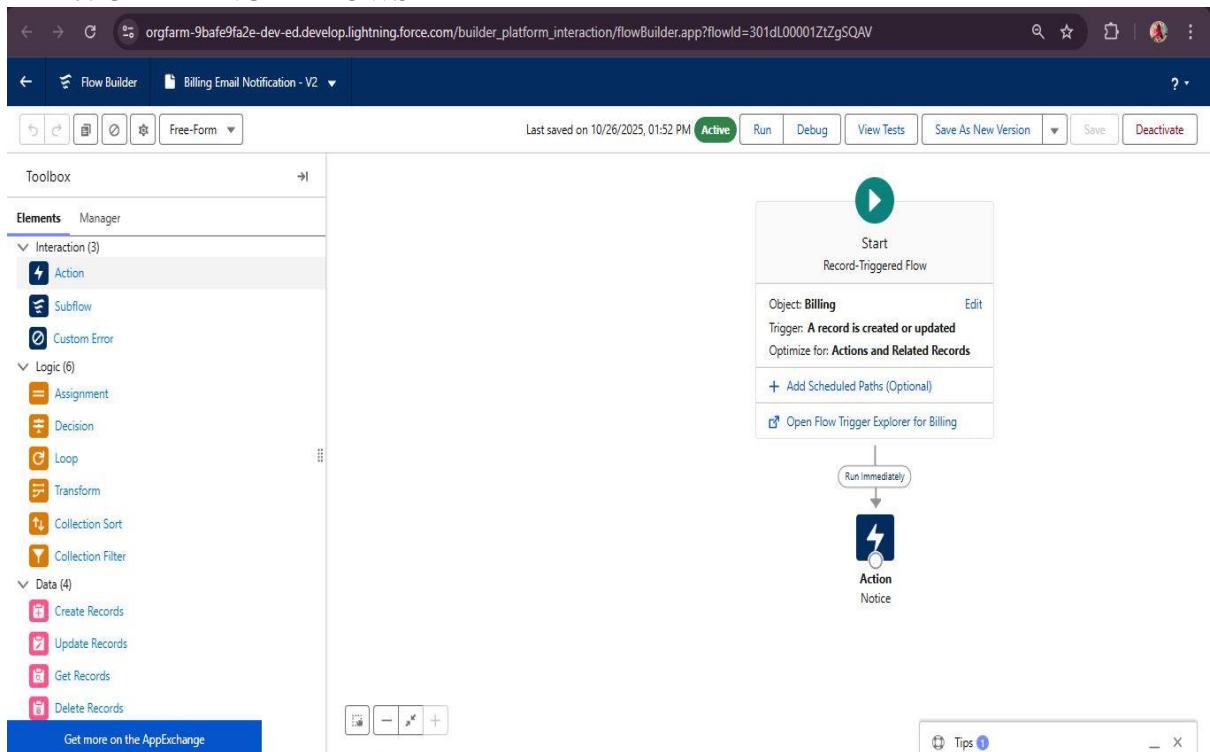


Fig 17.1 Creating flow for billing amount flow with email

## **10. EXPECTED OUTCOMES**

- Efficient management of customer and product data.
- Improved customer satisfaction through timely communication.
- Better business decisions using analytical reports.
- Reduced manual workload and paperwork.
- Enhanced brand loyalty via personalized service.

## **11. ADVANTAGES**

- Centralized management of jewelry and customer information.
- Automation reduces time and human errors.
- Real-time access to sales and customer data.
- Boosts marketing through personalized recommendations.
- Easy scalability and customization.

## **12. FUTURE ENHANCEMENTS**

- Integration with mobile app for customers and staff.
- Implementation of AI-based recommendations for jewelry suggestions.
- Addition of loyalty programs and reward points.
- Chatbot support for instant customer service.
- Integration with payment gateways and invoice automation.

## **13. CONCLUSION**

The CRM Application for Jewel Management provides an efficient and modern solution for managing jewelry business operations. It helps store owners build strong customer relationships, track inventory, and improve sales through data-driven strategies. With its modular design and scalability, this system not only simplifies daily tasks but also enhances customer experience. Future upgrades can further transform it into a complete digital platform for jewelry businesses.

## **14. REFERENCES**

❖ **Salesforce Developer Documentation:**

<http://salesforce.com/docs>

❖ **Trailhead Modules:**

<https://trailhead.salesforce.com/>

❖ **Salesforce Flow Guide:**

<http://salesforce.com/developer-centers/flow>

❖ **Apex Developer Guide:**

[http://salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_dev\\_guide.htm](http://salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_dev_guide.htm)

❖ **Lightning Web Components Documentation:**

<http://salesforce.com/docs/component-library/overview/components>