PROJECT DEVELOPMENT PHASE

Date	18th June 2025
Team ID	LTVIP2025TMID29987
Project Name	CRM Application for Jewelry Management – (Developer)
Maximum Marks	*(To be filled by evaluator)*

1. OVERVIEW

This document captures the technical deliverables implemented as part of the **CRM for Jewel Management** project in Salesforce. It includes the details of custom configurations, automation logic, sample datasets, and screenshots of functional outputs. The goal is to demonstrate the executable components of the solution and provide visual evidence that the core functionality has been correctly built and deployed.

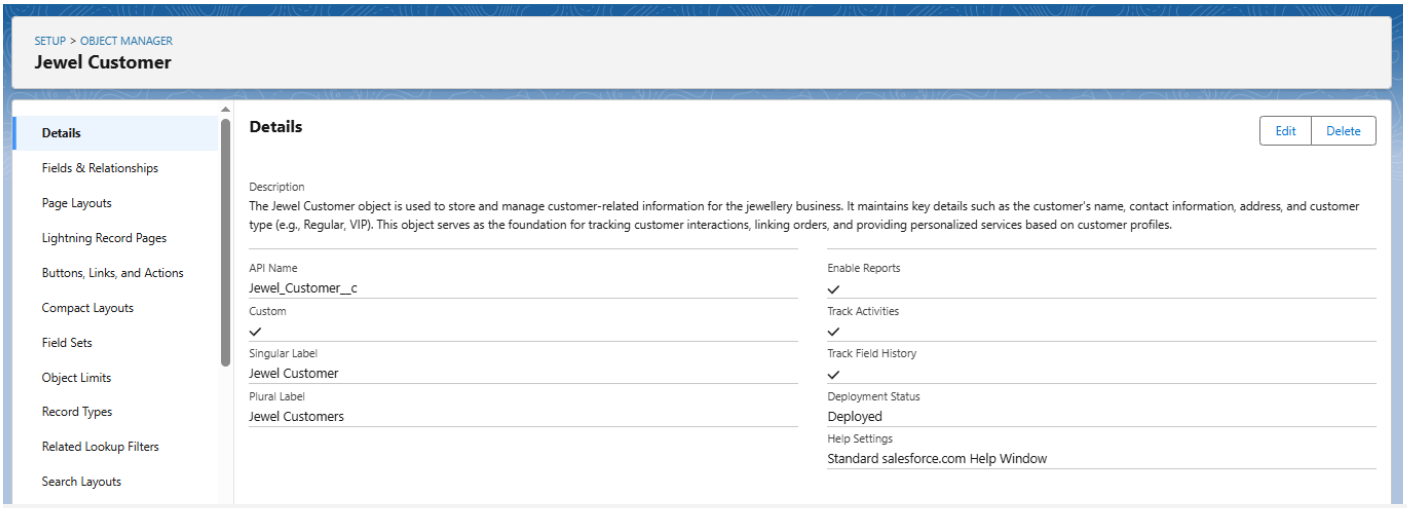
This section serves as a **proof of execution**, showcasing that the development aligns with the problem statement, solution design, and project objectives defined in the earlier phases.

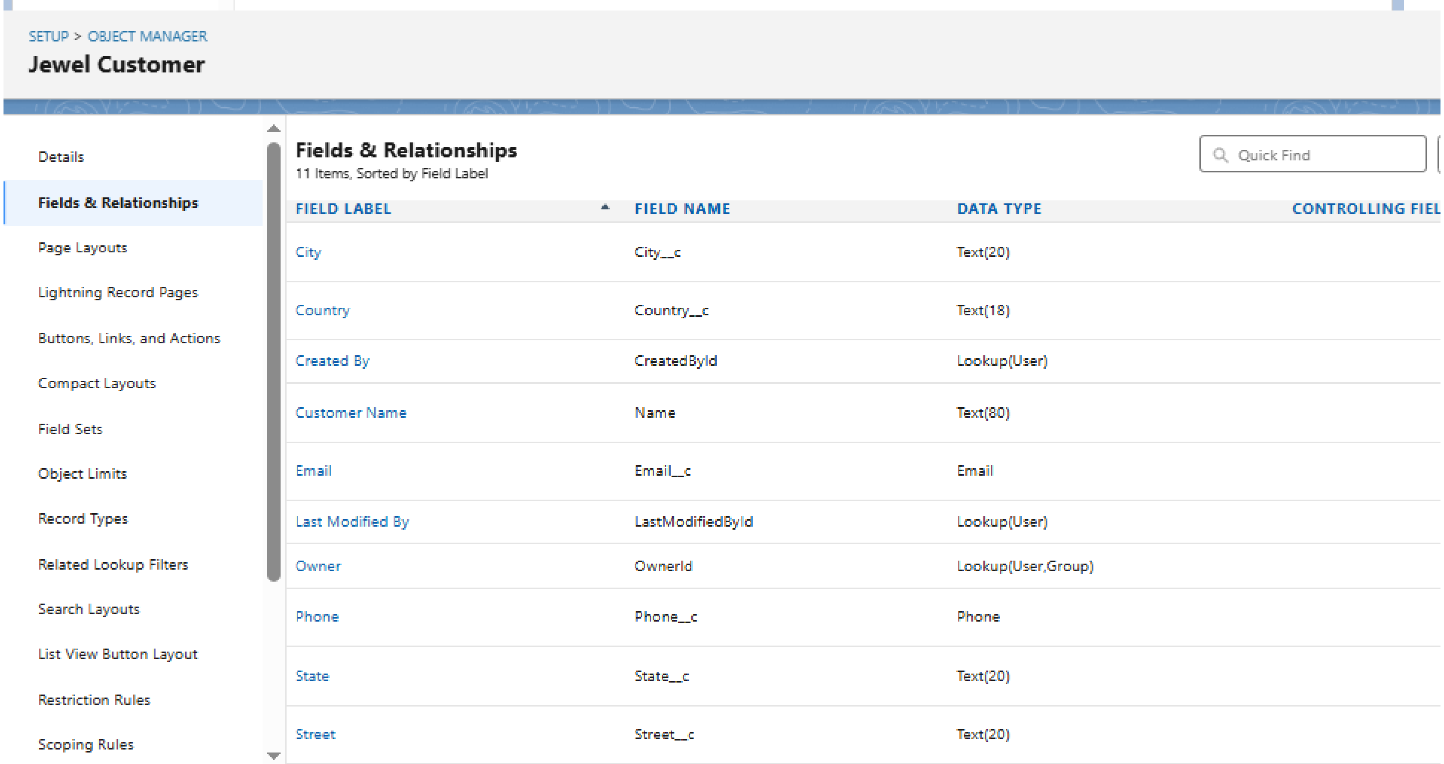
2. REQUIRED PROJECT FILES

A. Custom Objects and Their Roles

1. **Jewel_Customer__c**
    - **Purpose**: Stores customer information including name, contact number, email address, residential address, and gender.
    - **Key Fields :**
        - **Name**
        - **Contact_No__c**
        - **Email__c**
        - **Address__c**
        - **Gender__c**

Usage: This object is referenced in the Billing and Order modules for identifying customer-related transactions.



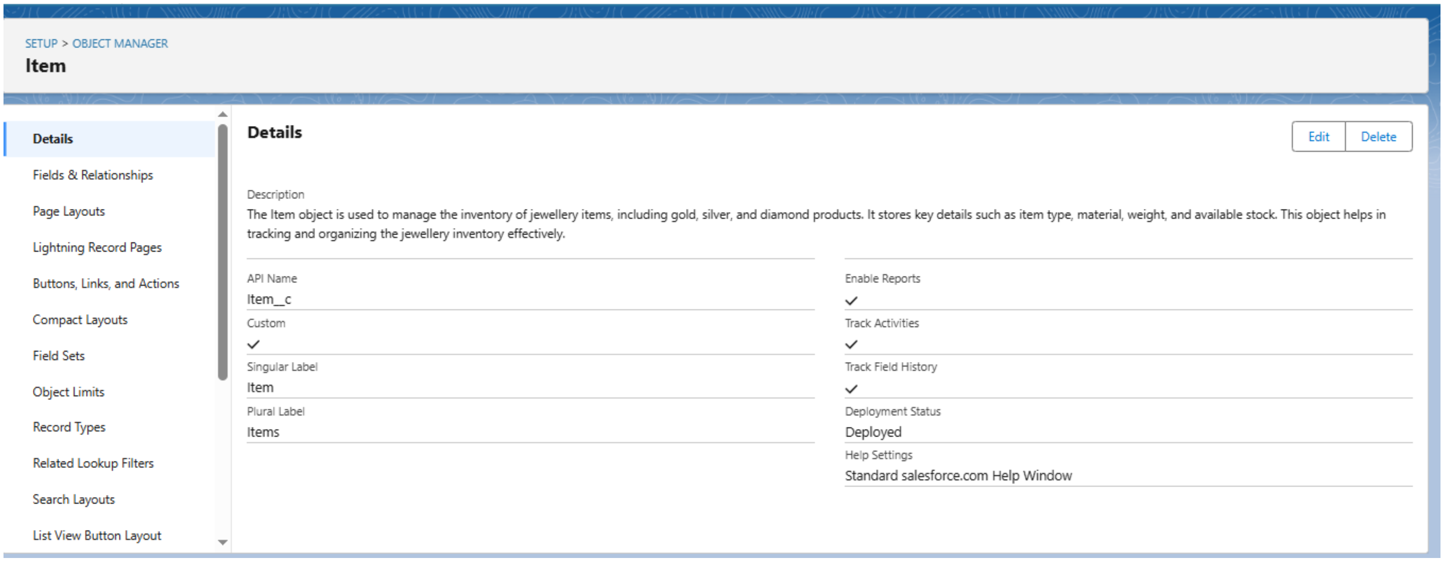


FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIEL
City	City__c	Text(20)	
Country	Country__c	Text(18)	
Created By	CreatedById	Lookup(User)	
Customer Name	Name	Text(80)	
Email	Email__c	Email	
Last Modified By	LastModifiedById	Lookup(User)	
Owner	OwnerId	Lookup(User,Group)	
Phone	Phone__c	Phone	
State	State__c	Text(20)	
Street	Street__c	Text(20)	

2. **Item__c**

   - **Purpose**: Maintains details about jewelry inventory (gold/silver).
   - Key Fields:
     - Item_Name__c
     - Ornament_Type__c
     - Weight__c
     - Price__c

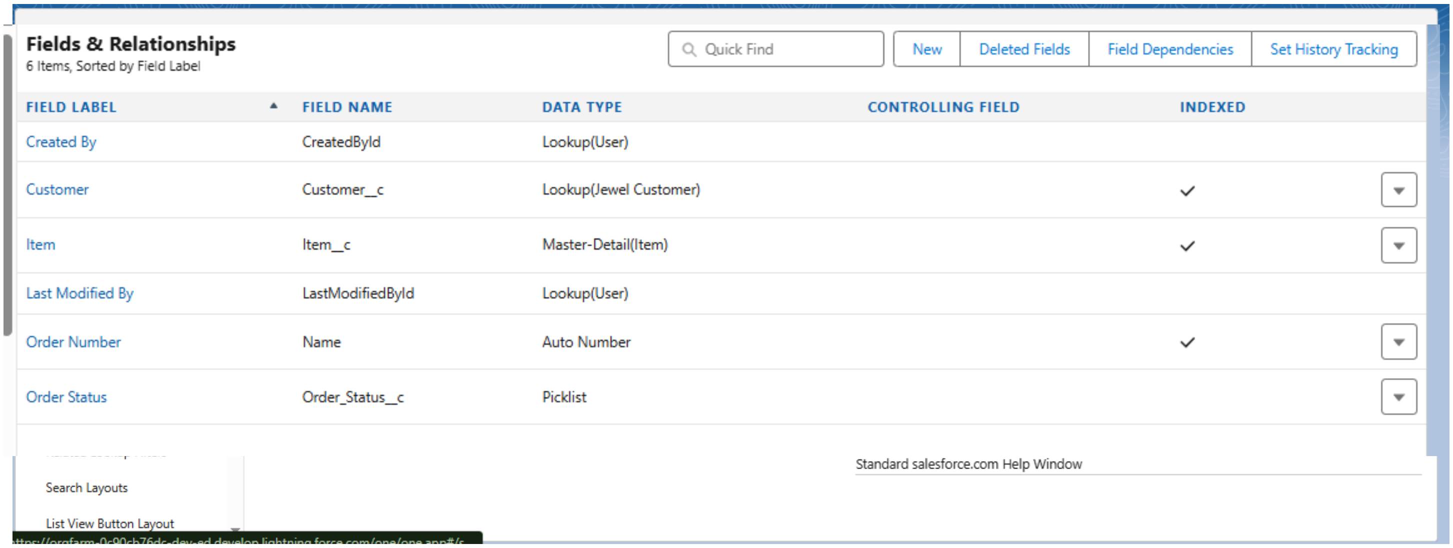
- KDM_Charges__c
- Stone_Price__c
- Making_Charges__c



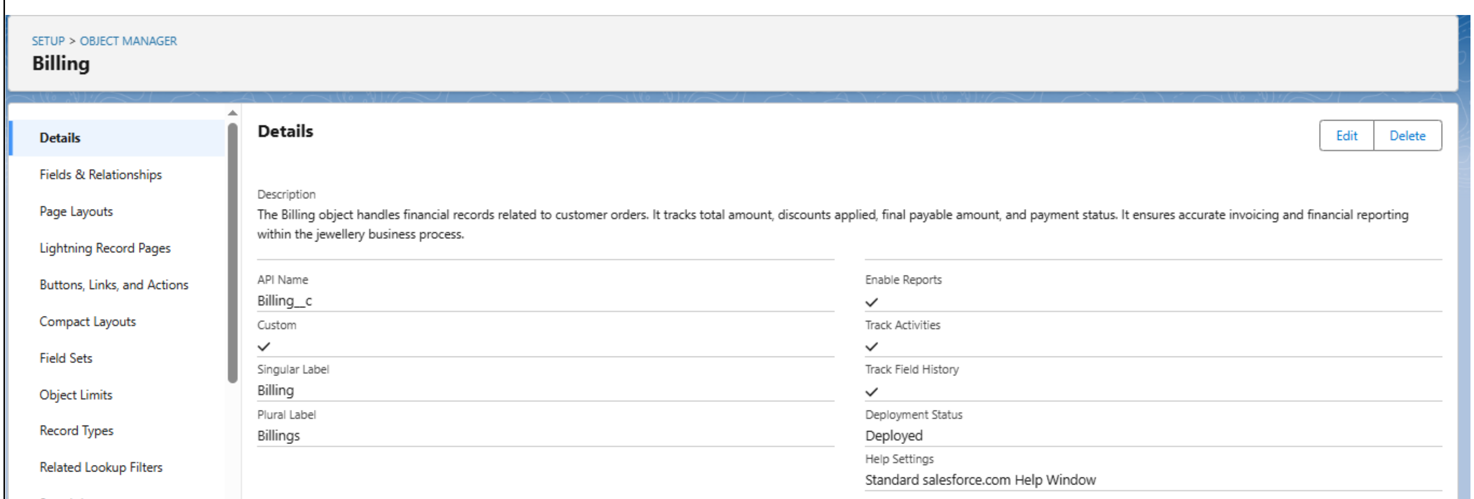
- Usage: Directly linked to Orders and Billing modules to calculate costs and manage inventory.

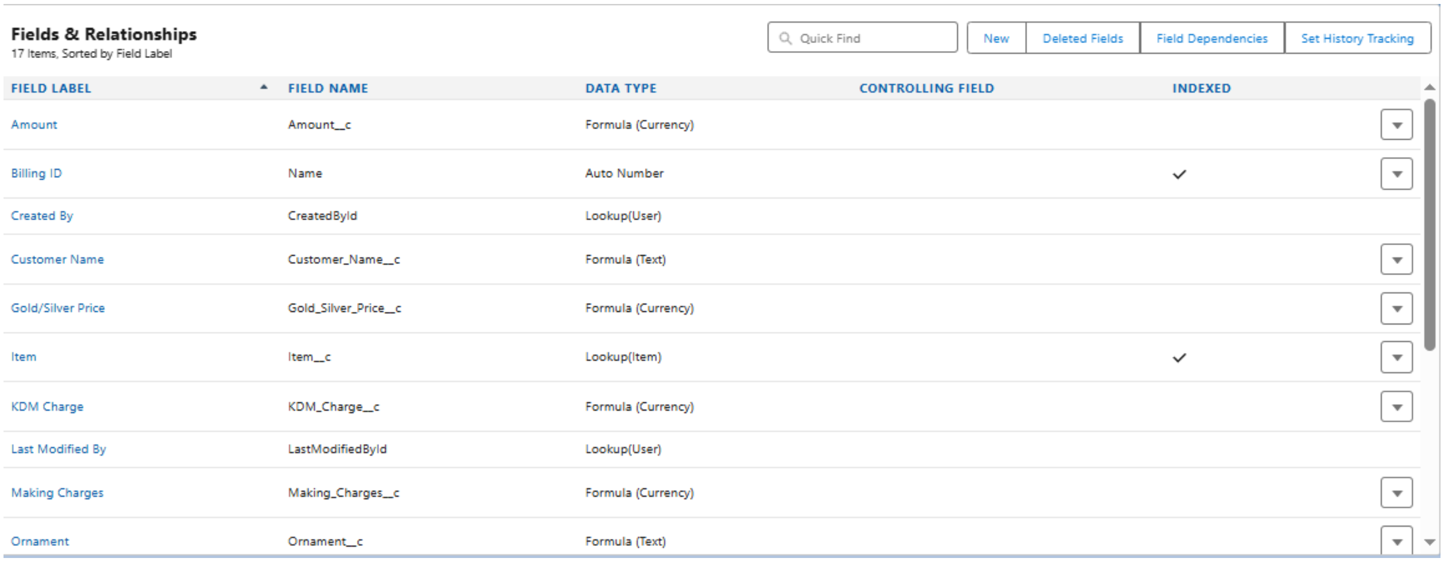
3. **Customer_Order__c**
   - Purpose: Captures item orders placed by customers.
   - Key Fields:
     - Ordered_Item__c (Lookup to Item__c)
     - Order_Date__c
     - Quantity__c
     - Linked_Customer__c (Lookup to Jewel_Customer__c)

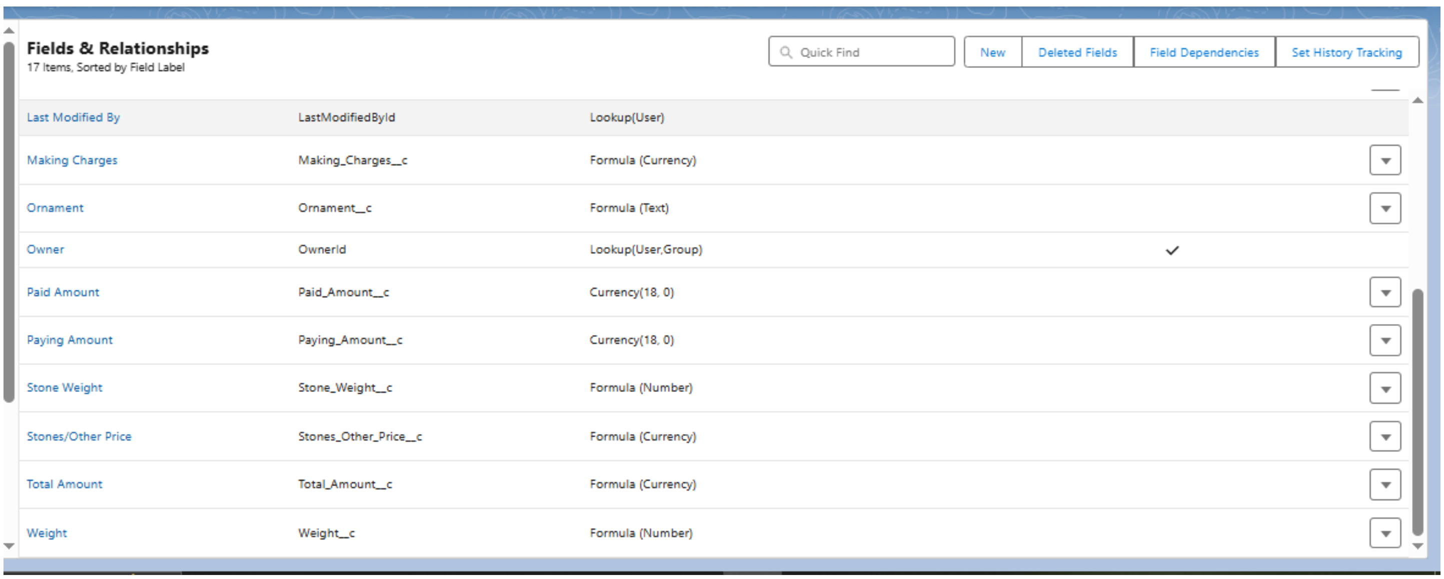
- Usage: Helps track what each customer has ordered, when, and in what quantity.



4. **Billing__c**
   - Purpose: Generates invoices and calculates total payable amounts.
   - Key Formula Fields:
     - Total_Amount__c = Price + KDM + Stone + Making
     - Paid_Amount__c (Auto-updated via Trigger)
   - Usage: The Billing object is central to the payment and notification automation.







5. Price__c (Optional – Pricing History Tracker)

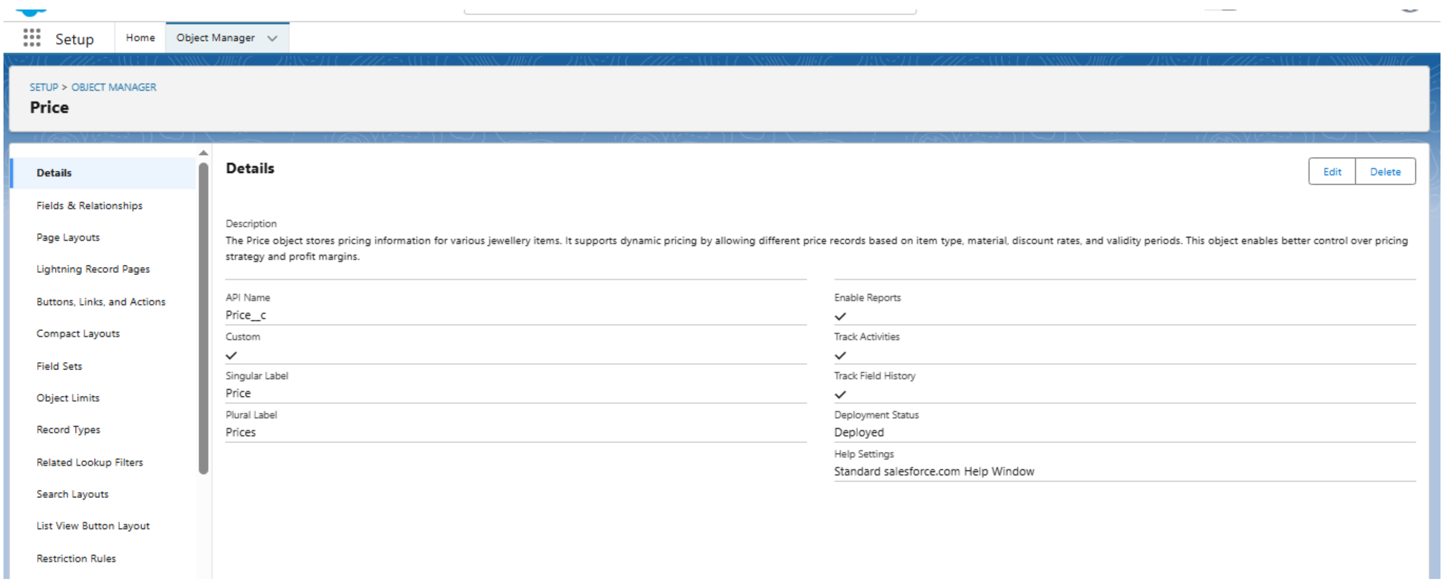
Purpose:
Enables dynamic and date-based pricing for gold and silver. Used for historical pricing records, supporting flexible pricing strategy for different order periods.

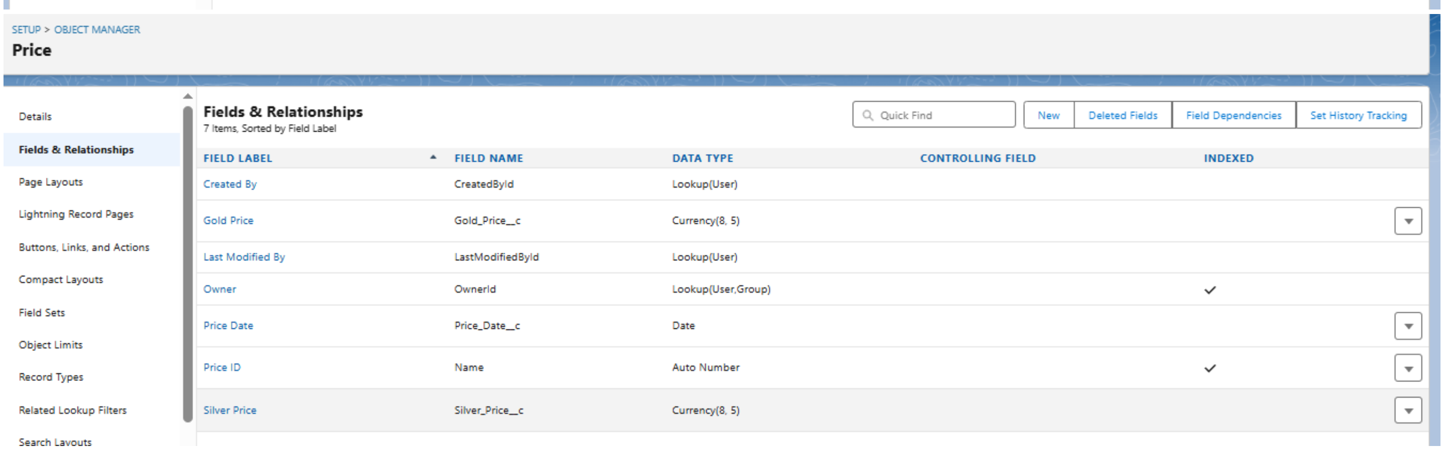
Key Fields:

Field Label	API Name	Data Type	Description
Gold Price	Gold_Price__c	Currency	Price per gram of gold for a specific date.
Silver Price	Silver_Price__c	Currency	Price per gram of silver for a specific date.
Price Date	Price_Date__c	Date	Date on which this price is valid.

Usage:

- Integrated with Item__c or Billing__c to reference latest or valid rates.
- Helps finance teams update market prices without affecting old records.
- Supports future automation and real-time rate integrations.





B. AUTOMATION ELEMENTS

The **CRM for Jewel Management** system leverages Salesforce's automation capabilities to streamline business processes, reduce manual interventions, and enhance operational accuracy. The automation comprises **Apex Triggers**, **Record-Triggered Flows**, and **Validation Rules**, all tailored to meet the critical needs of a jewelry sales and billing system.

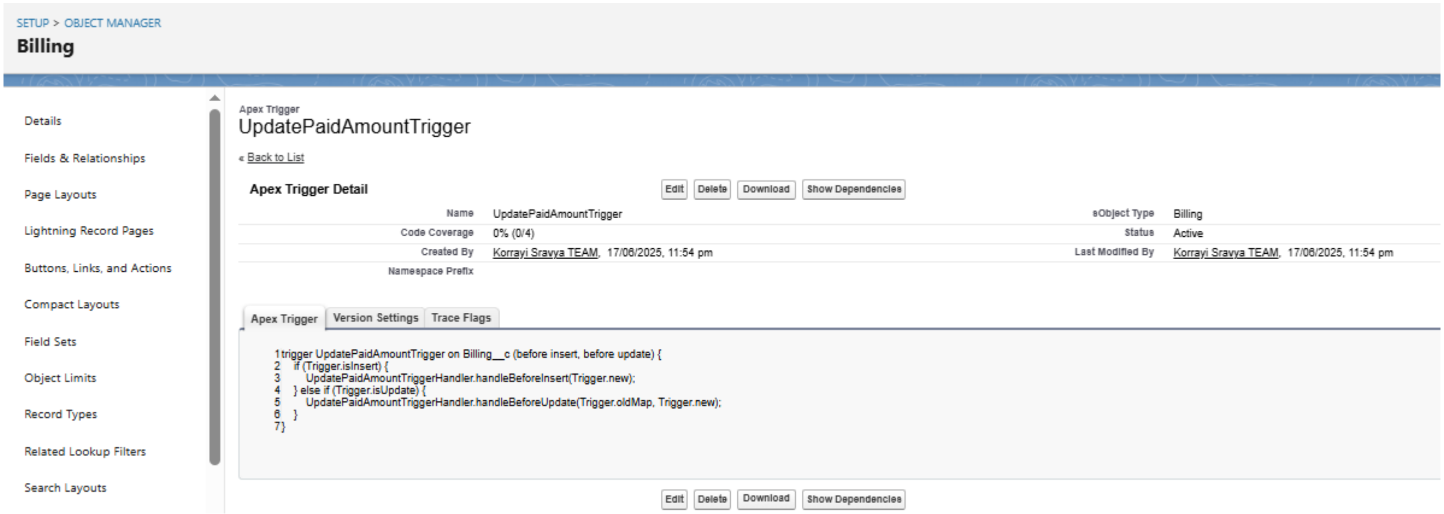
1. Apex Trigger – Auto Update of Paid Amount

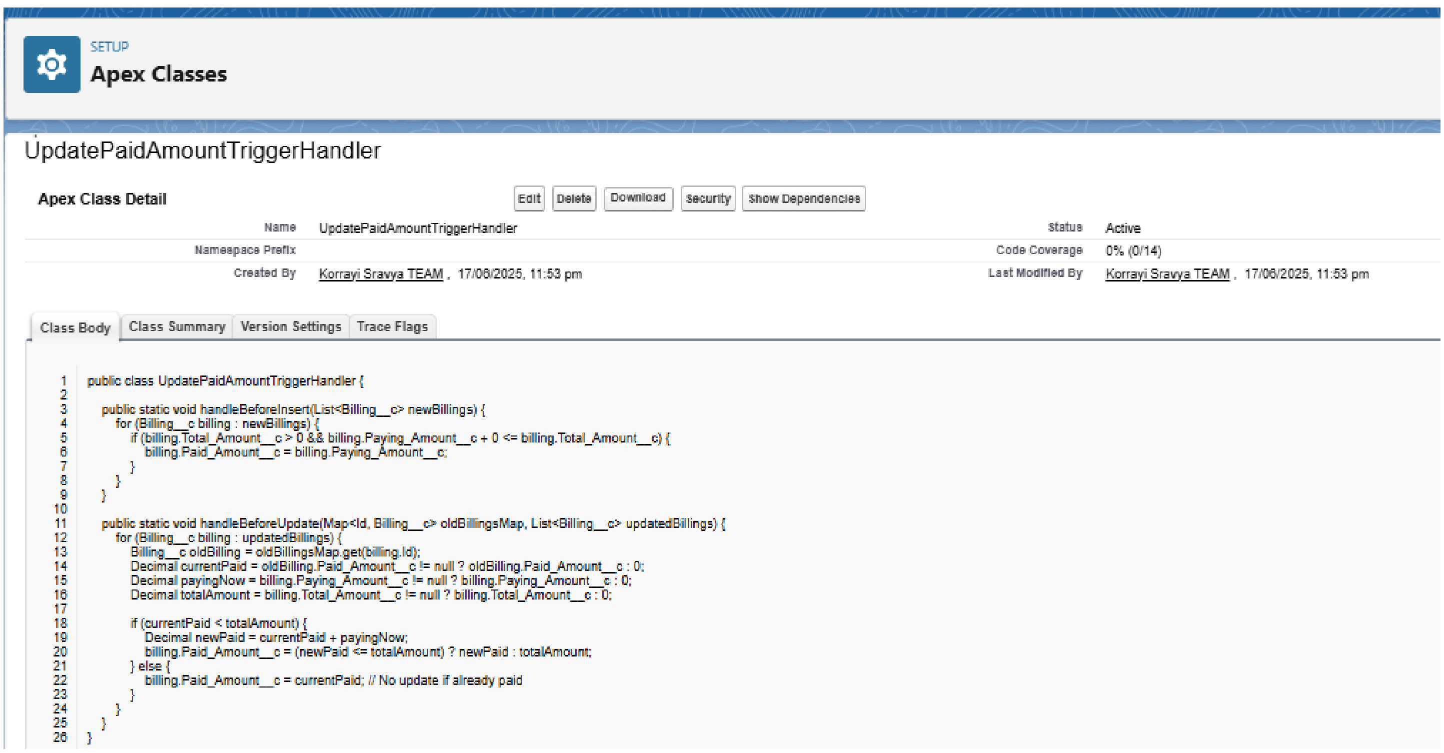
- **Name:** UpdatePaidAmountTrigger
- **Trigger Type:** After Insert, After Update (on a custom object such as Payment__c or Billing__c)
- **Description:**
  This Apex trigger ensures that the Paid_Amount__c field in the **Billing__c** object reflects the correct payment received by the customer. Once a payment record is inserted or updated, this trigger automatically recalculates and updates the amount paid without requiring manual data entry by the staff.

Core Logic:

```
trigger UpdatePaidAmountTrigger on Billing__c (after insert, after update) {
   for (Billing__c bill : Trigger.new) {
      if (bill.Payment_Received__c != null) {
         bill.Paid_Amount__c = bill.Payment_Received__c;
         update bill;
      }
   }
}
```

- **Impact:**
  - Eliminates human errors in manual entry of payment details.
  - Reduces workload on data-entry staff.
  - Keeps the financial records up-to-date and synchronized in real time.
  - Prepares accurate data for reporting and analytics (e.g., Outstanding Payments).
- **Testing Done:**
  - Trigger tested with multiple values (partial/full payments).
  - Negative cases validated where Payment_Received__c is null.





This trigger ensures the Paid_Amount__c field is auto-updated upon payment entry in the Billing__c object. It reduces manual effort and improves data accuracy.

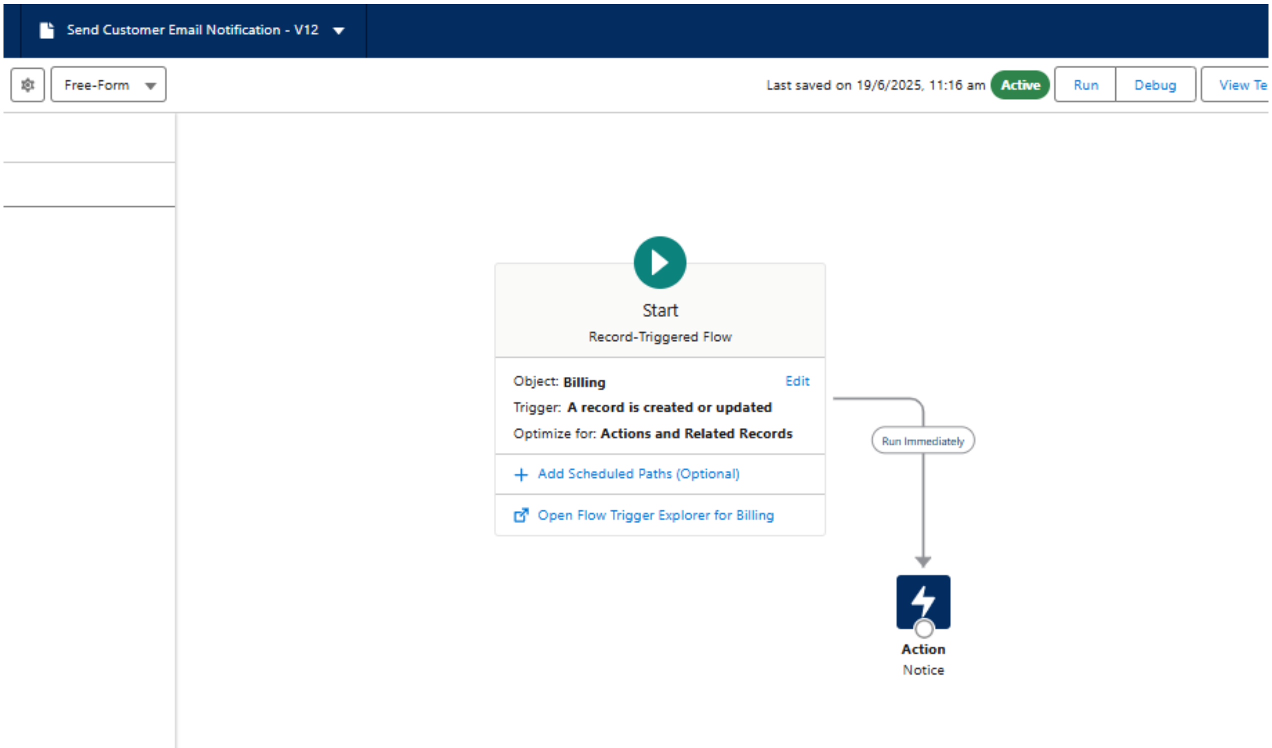
2. Record-Triggered Flow – Automated Email Notification

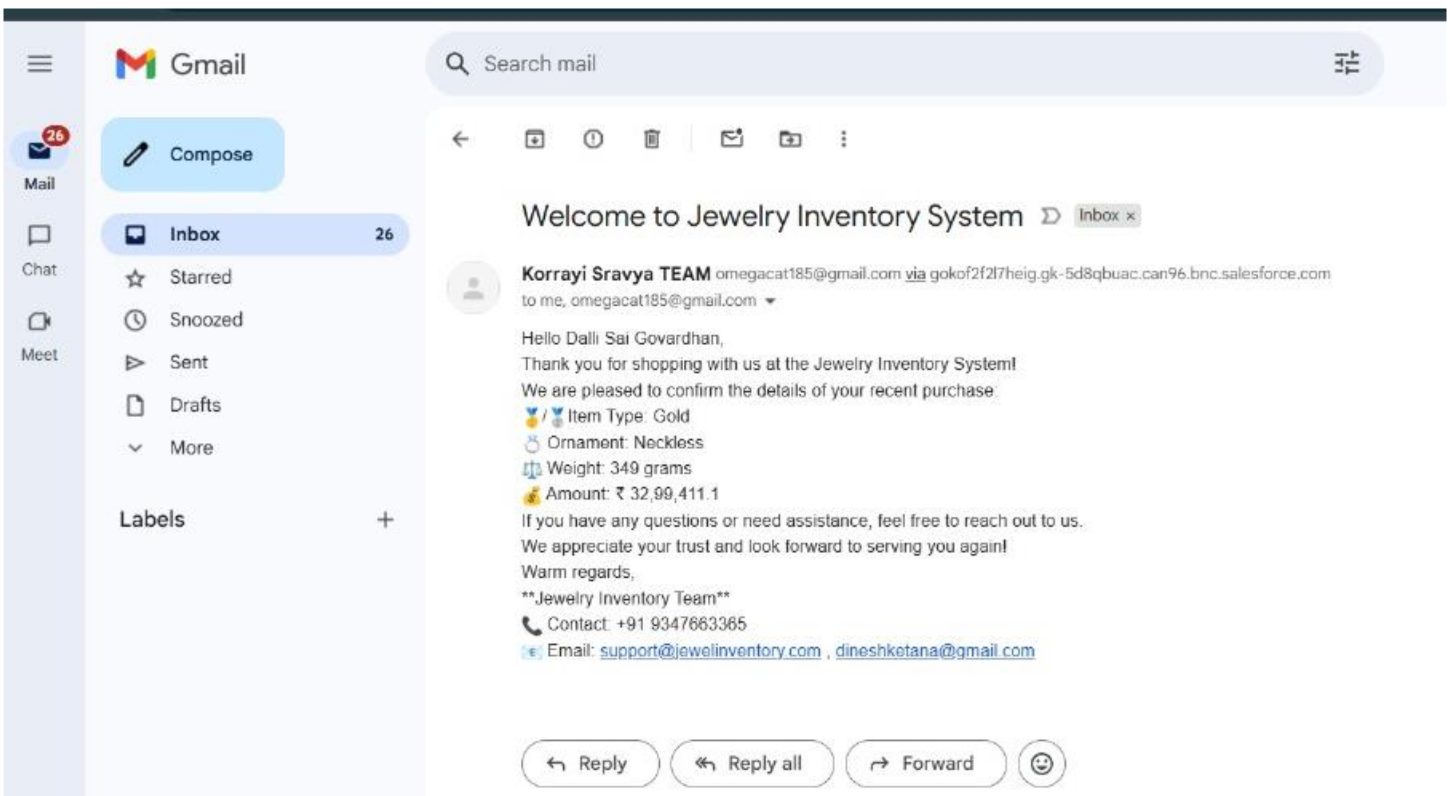
- **Flow Name:** Billing_Confirmation_Email_Flow
- **Trigger Event:** On Create or Update of a **Billing__c** record.
- **Flow Description:**
  A **Record-Triggered Flow** is designed to automatically send a billing confirmation email to the customer upon successful creation or update of a billing record. It fetches customer data using **lookup relationships** and uses **Send Email** action with a dynamic message template.

- **Flow Design Highlights:**
    - **Start Element:** Triggered when a new billing record is created or an existing one is updated.
    - **Get Records:** Retrieve the related **Jewel_Customer__c** record using the lookup field.
    - **Decision Element:** Checks if a valid email exists and if the billing status is 'Paid' or 'Generated'.
    - **Action Element:** Sends a formatted HTML email summarizing:
        - Customer Name
        - Item Details
        - Amount Paid and Balance (if any)
        - Billing Date
- **Importance:**
    - Improves customer experience by providing immediate confirmation.
    - Builds trust and transparency.
    - Ensures the customer has a digital proof of the transaction.
    - Supports future customer queries and dispute resolution.
- **Email Template Sample Content:**
- Dear {!Jewel_Customer__c.Name},
- Thank you for your recent purchase. Your billing for the item {!Billing__c.Related_Item__c} has been successfully generated.
- Amount Paid: {!Billing__c.Paid_Amount__c}
- Total Amount: {!Billing__c.Total_Amount__c}
- Balance Due: {!Billing__c.Total_Amount__c - Billing__c.Paid_Amount__c}

  Regards,

CRM Jewel Team.





This Record-Triggered Flow sends an automated email to the customer upon creation or update of a billing record. It fetches customer email through a lookup and dynamically composes the message.

3. Validation Rules – Ensuring Financial Integrity

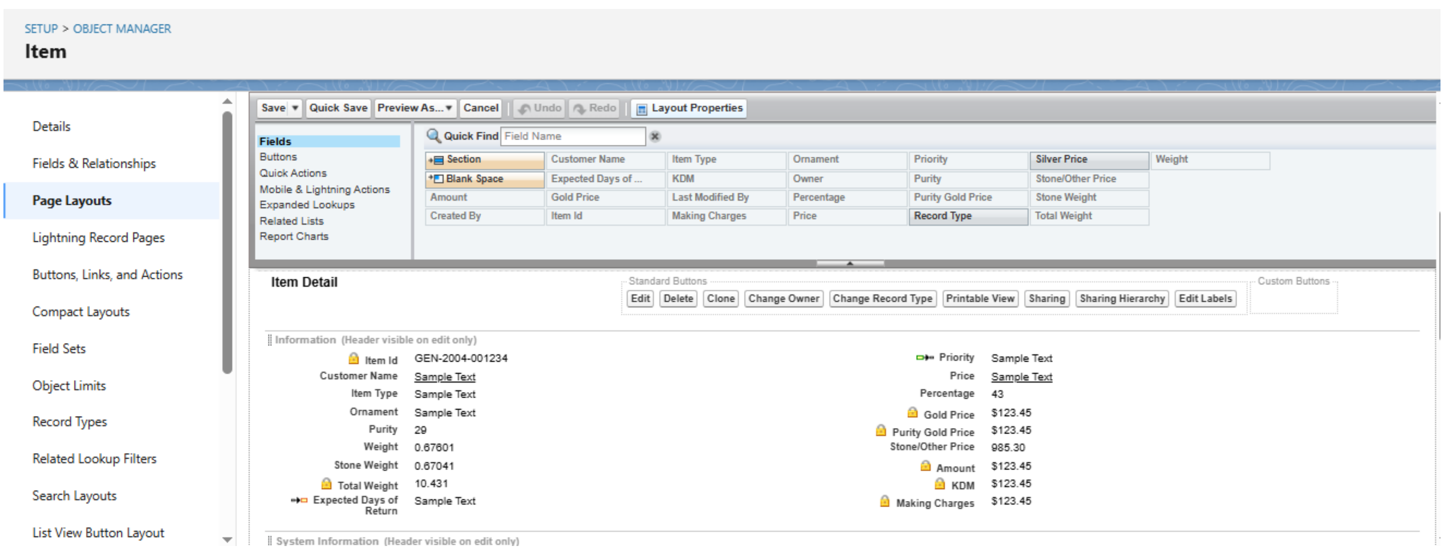
- **Validation Rule Name:** Validate_Paid_Amount
- **Associated Object:** Billing__c
- **Formula Logic:**
- Paid_Amount__c > Total_Amount__c
- **Error Message:**
  "Paid Amount cannot be greater than the Total Amount. Please verify the values."
- **Purpose:**
  - Prevents users from entering a paid amount greater than the actual invoice total.
  - Maintains **financial accuracy** within the billing records.
  - Ensures billing reports, outstanding balance calculations, and customer receipts are error-free.
  - Acts as a safeguard against data-entry mistakes and fraudulent inputs.
- **Usage Scenario:**
  - During billing record creation or updates, if a staff member mistakenly inputs a higher paid amount, the system displays an error and blocks the operation.
  - Ensures integrity of data especially when used in integrations or reports.

C. UI COMPONENTS

A. Custom Page Layouts

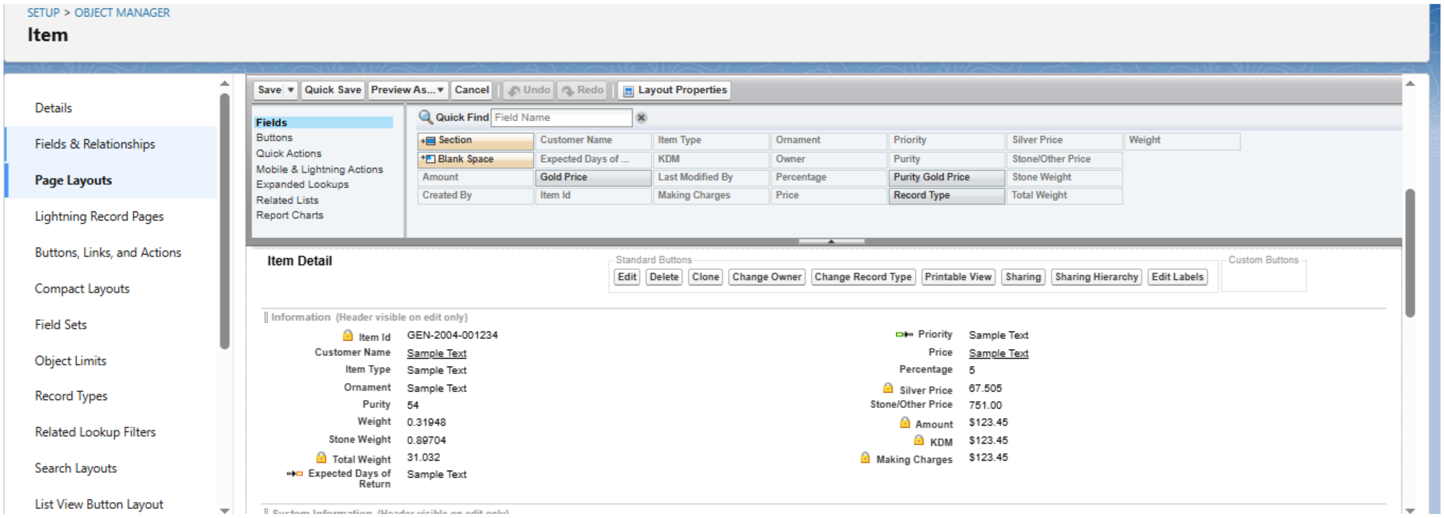
Activity 1: Page Layout for Gold Items

- Navigated to **Item Object → Page Layouts**
- Created a new layout titled: Page Layout for Gold
- Removed fields related to Silver items
- Organized fields relevant to gold ornament information such as:
  - Ornament Type
  - Weight
  - Price
  - Making Charges
  - KDM Charges
- Saved and assigned layout accordingly



Activity 2: Page Layout for Silver Items

- Created a separate layout titled: Page Layout for Silver
- Removed gold-specific fields
- Displayed only silver-related attributes for accurate record entry and display



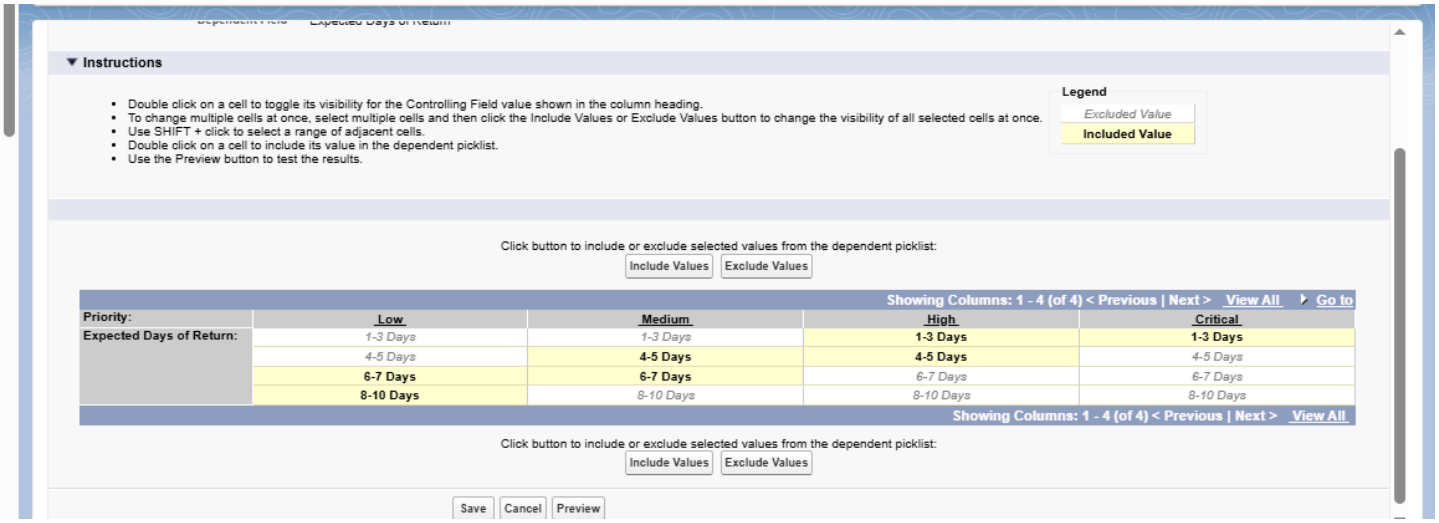
B. Field Dependencies

Use Case:

Enable conditional visibility of fields in the **Item** object using field dependencies.

Configuration Steps:

1. Navigated to **Item Object → Fields & Relationships**
2. Selected controlling field: Priority
3. Selected dependent field: Expected Days of Return
4. Mapped:
    - High Priority → 1-2 Days
    - Medium Priority → 3-5 Days
    - Low Priority → 5+ Days
5. Saved field dependency mapping



C. Reports Configuration

Use Case:

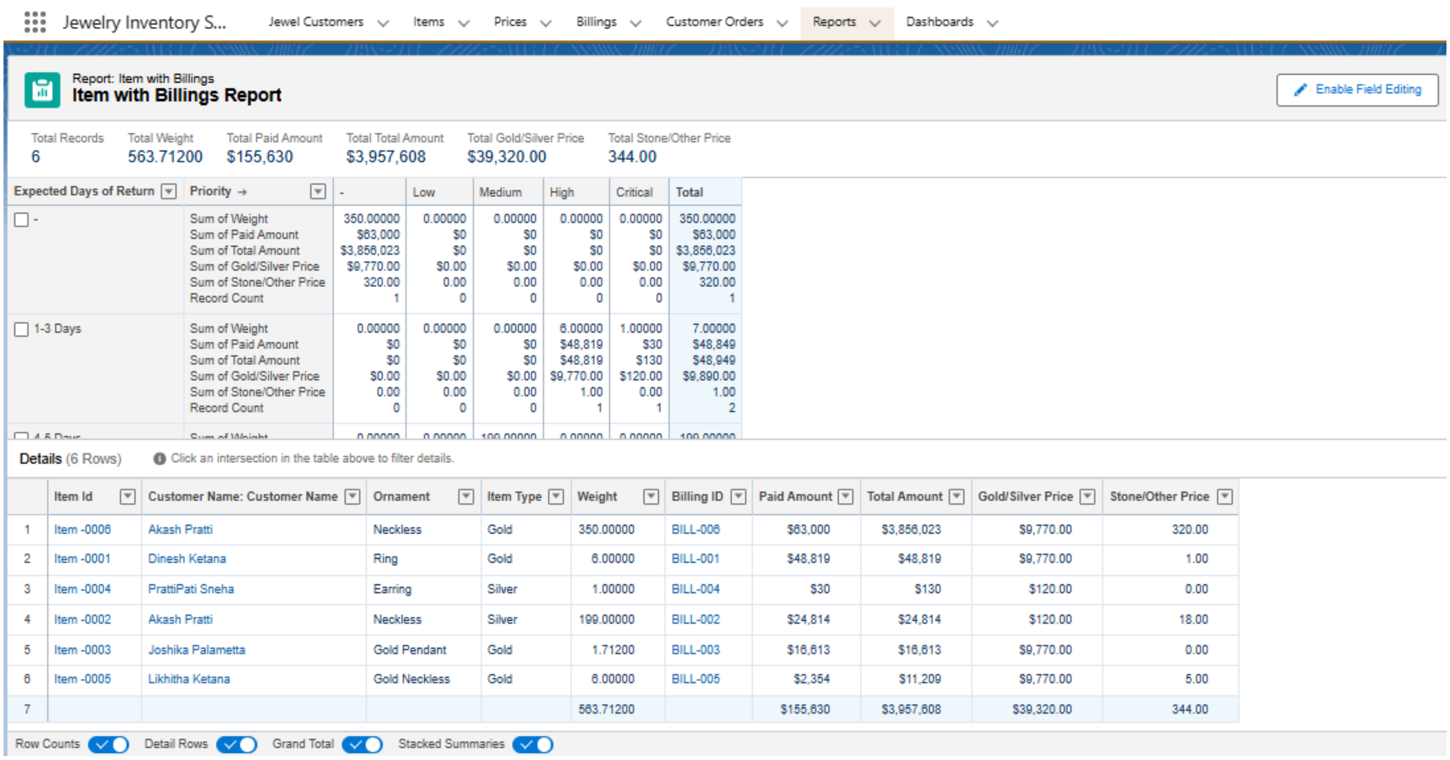
Goldsmith requires summarized reports for inventory, customer orders, and billing for analysis and decision-making.

Activity 1: Custom Reports

- Created reports using Report Builder:
    - **Tabular Report**: Jewel Customers by Gender
    - **Summary Report**: Item Price Distribution by Ornament Type
- Added fields: Customer Name, Item Type, Price, Order Date, Total Amount

Activity 2:

- Created:
    - Report 1: **Item with Billings**
    - Report 2: **Billings with Item and Customer Order**



Jewelry Inventory S... Jewel Customers Items Prices Billings Customer Orders Reports Dashboards

Report: Item with Billings

Item with Billings Report

Enable Field Editing

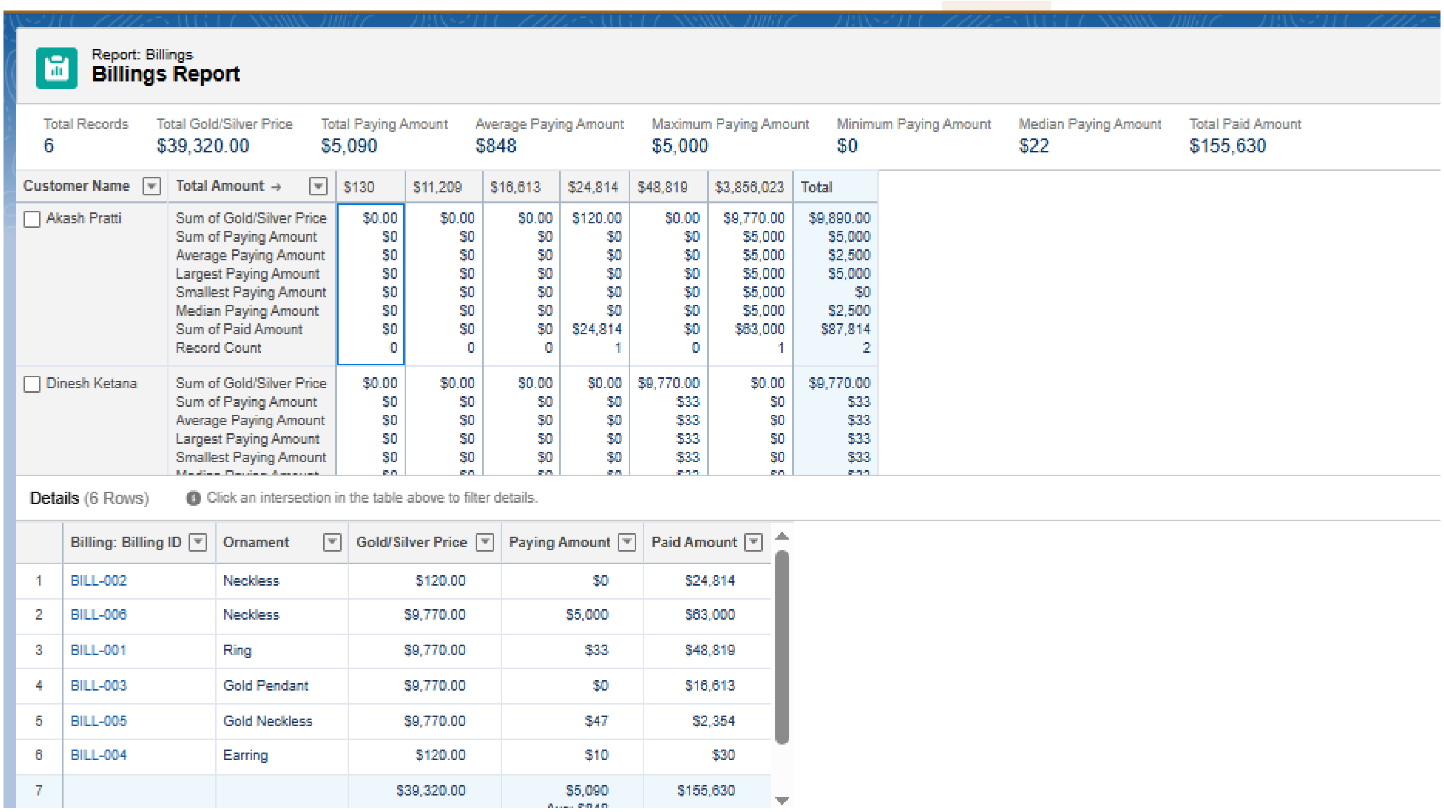
Total Records	Total Weight	Total Paid Amount	Total Total Amount	Total Gold/Silver Price	Total Stone/Other Price
6	563.71200	$155,630	$3,957,608	$39,320.00	344.00

Expected Days of Return	Priority →	-	Low	Medium	High	Critical	Total
-	Sum of Weight	350.00000	0.00000	0.00000	0.00000	0.00000	350.00000
	Sum of Paid Amount	$63,000	$0	$0	$0	$0	$63,000
	Sum of Total Amount	$3,856,023	$0	$0	$0	$0	$3,856,023
	Sum of Gold/Silver Price	$9,770.00	$0.00	$0.00	$0.00	$0.00	$9,770.00
	Sum of Stone/Other Price	320.00	0.00	0.00	0.00	0.00	320.00
	Record Count	1	0	0	0	0	1
1-3 Days	Sum of Weight	0.00000	0.00000	0.00000	6.00000	1.00000	7.00000
	Sum of Paid Amount	$0	$0	$0	$48,819	$30	$48,849
	Sum of Total Amount	$0	$0	$0	$48,819	$130	$48,949
	Sum of Gold/Silver Price	$0.00	$0.00	$0.00	$9,770.00	$120.00	$9,890.00
	Sum of Stone/Other Price	0.00	0.00	0.00	1.00	0.00	1.00
	Record Count	0	0	0	1	1	2

Details (6 Rows) Click an intersection in the table above to filter details.

	Item Id	Customer Name: Customer Name	Ornament	Item Type	Weight	Billing ID	Paid Amount	Total Amount	Gold/Silver Price	Stone/Other Price
1	Item -0006	Akash Pratti	Neckless	Gold	350.00000	BILL-006	$63,000	$3,856,023	$9,770.00	320.00
2	Item -0001	Dinesh Ketana	Ring	Gold	6.00000	BILL-001	$48,819	$48,819	$9,770.00	1.00
3	Item -0004	PrattiPati Sneha	Earring	Silver	1.00000	BILL-004	$30	$130	$120.00	0.00
4	Item -0002	Akash Pratti	Neckless	Silver	199.00000	BILL-002	$24,814	$24,814	$120.00	18.00
5	Item -0003	Joshika Palametta	Gold Pendant	Gold	1.71200	BILL-003	$16,613	$16,613	$9,770.00	0.00
6	Item -0005	Likhitha Ketana	Gold Neckless	Gold	6.00000	BILL-005	$2,354	$11,209	$9,770.00	5.00
7					563.71200		$155,630	$3,957,608	$39,320.00	344.00

Row Counts Detail Rows Grand Total Stacked Summaries



Report: Billings

Billings Report

Total Records	Total Gold/Silver Price	Total Paying Amount	Average Paying Amount	Maximum Paying Amount	Minimum Paying Amount	Median Paying Amount	Total Paid Amount
6	$39,320.00	$5,090	$848	$5,000	$0	$22	$155,630

Customer Name	Total Amount →	$130	$11,209	$16,613	$24,814	$48,819	$3,856,023	Total
Akash Pratti	Sum of Gold/Silver Price	$0.00	$0.00	$0.00	$120.00	$0.00	$9,770.00	$9,890.00
	Sum of Paying Amount	$0	$0	$0	$0	$0	$5,000	$5,000
	Average Paying Amount	$0	$0	$0	$0	$0	$5,000	$2,500
	Largest Paying Amount	$0	$0	$0	$0	$0	$5,000	$5,000
	Smallest Paying Amount	$0	$0	$0	$0	$0	$5,000	$0
	Median Paying Amount	$0	$0	$0	$0	$0	$5,000	$2,500
	Sum of Paid Amount	$0	$0	$0	$24,814	$0	$63,000	$87,814
	Record Count	0	0	0	1	0	1	2
Dinesh Ketana	Sum of Gold/Silver Price	$0.00	$0.00	$0.00	$0.00	$9,770.00	$0.00	$9,770.00
	Sum of Paying Amount	$0	$0	$0	$0	$33	$0	$33
	Average Paying Amount	$0	$0	$0	$0	$33	$0	$33
	Largest Paying Amount	$0	$0	$0	$0	$33	$0	$33
	Smallest Paying Amount	$0	$0	$0	$0	$33	$0	$33

Details (6 Rows) Click an intersection in the table above to filter details.

	Billing: Billing ID	Ornament	Gold/Silver Price	Paying Amount	Paid Amount
1	BILL-002	Neckless	$120.00	$0	$24,814
2	BILL-006	Neckless	$9,770.00	$5,000	$63,000
3	BILL-001	Ring	$9,770.00	$33	$48,819
4	BILL-003	Gold Pendant	$9,770.00	$0	$16,613
5	BILL-005	Gold Neckless	$9,770.00	$47	$2,354
6	BILL-004	Earring	$120.00	$10	$30
7			$39,320.00	$5,090	$155,630

D. Dashboards

Use Case:

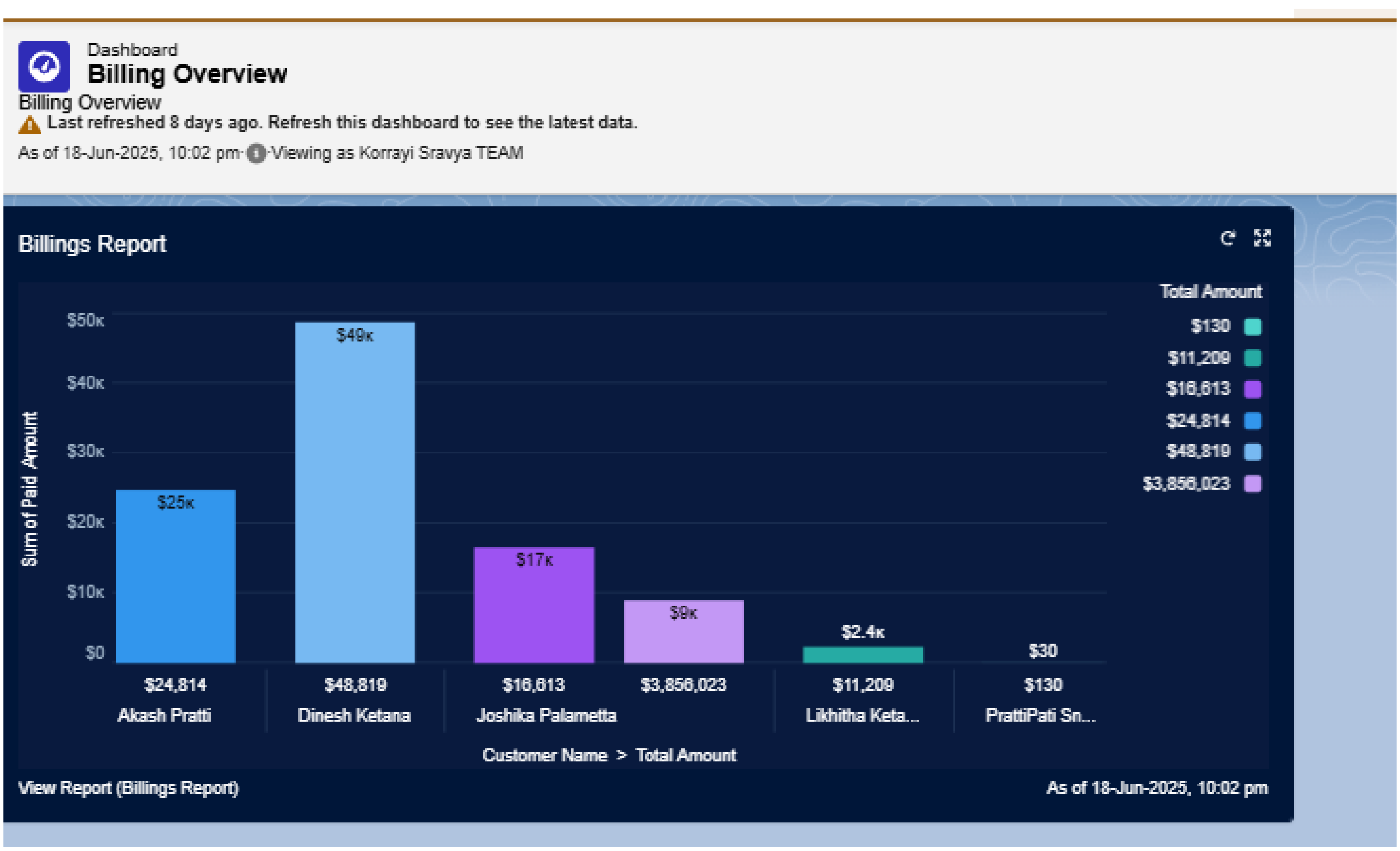
To visualize business KPIs and trends for gold and silver items, customer activity, and financial insights.

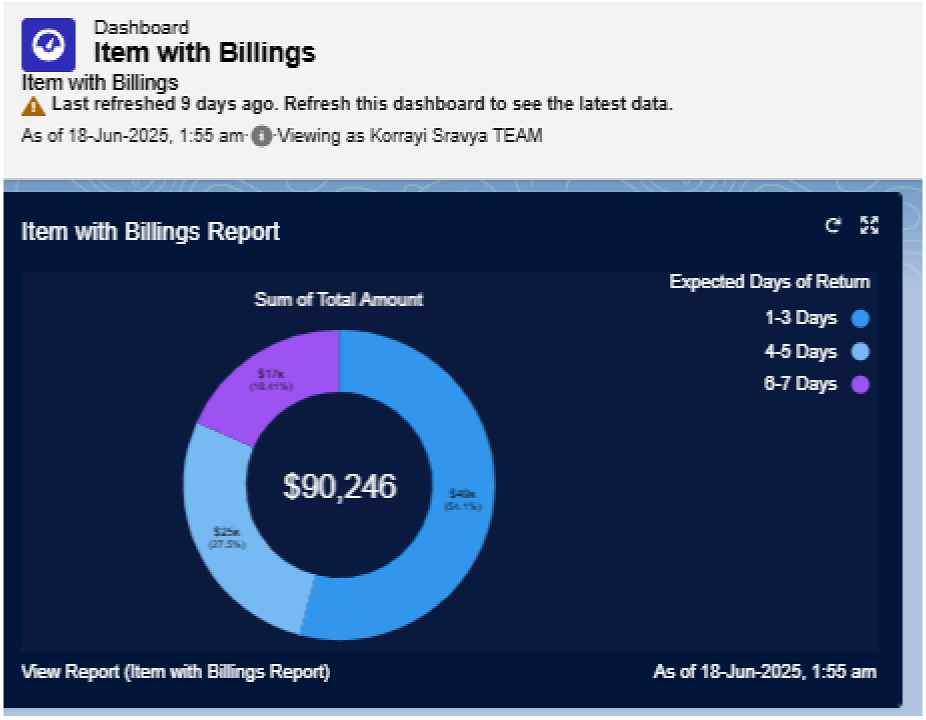
Activity 1: Create Dashboard – "Jewelry Business Overview"

- Added 3 components:
    - Total Billing by Item
    - Orders by Ornament Type
    - Customer Count by Gender

Activity 2: Additional Dashboard – "Customer and Order Insights"

- Included:
    - Orders by Date
    - Total Revenue Chart
    - Item Inventory Breakdown





E: Automation: Flows & Triggers

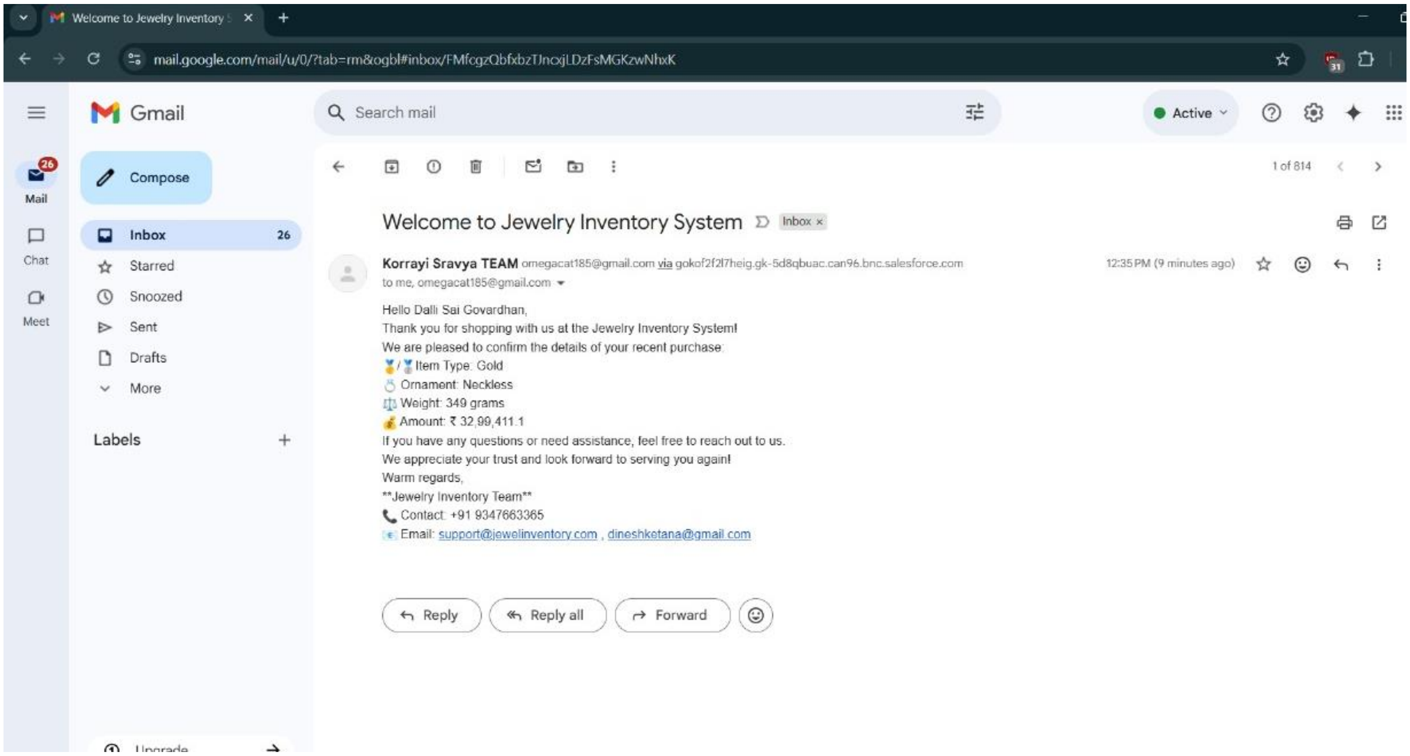
Flow: Record-Triggered Flow to Send Billing Confirmation Email

Use Case:

Automatically send an email upon Billing record creation/update.

Steps:

1. Created a **Record-Triggered Flow** on Billing__c (when created or updated)
2. Used Text Template to craft the email body with merge fields:
    - Customer Name
    - Item Type
    - Ornament
    - Weight
    - Amount
3. Configured email action to send notification using Send Email element
4. Activated flow



F. Automation: Apex Trigger

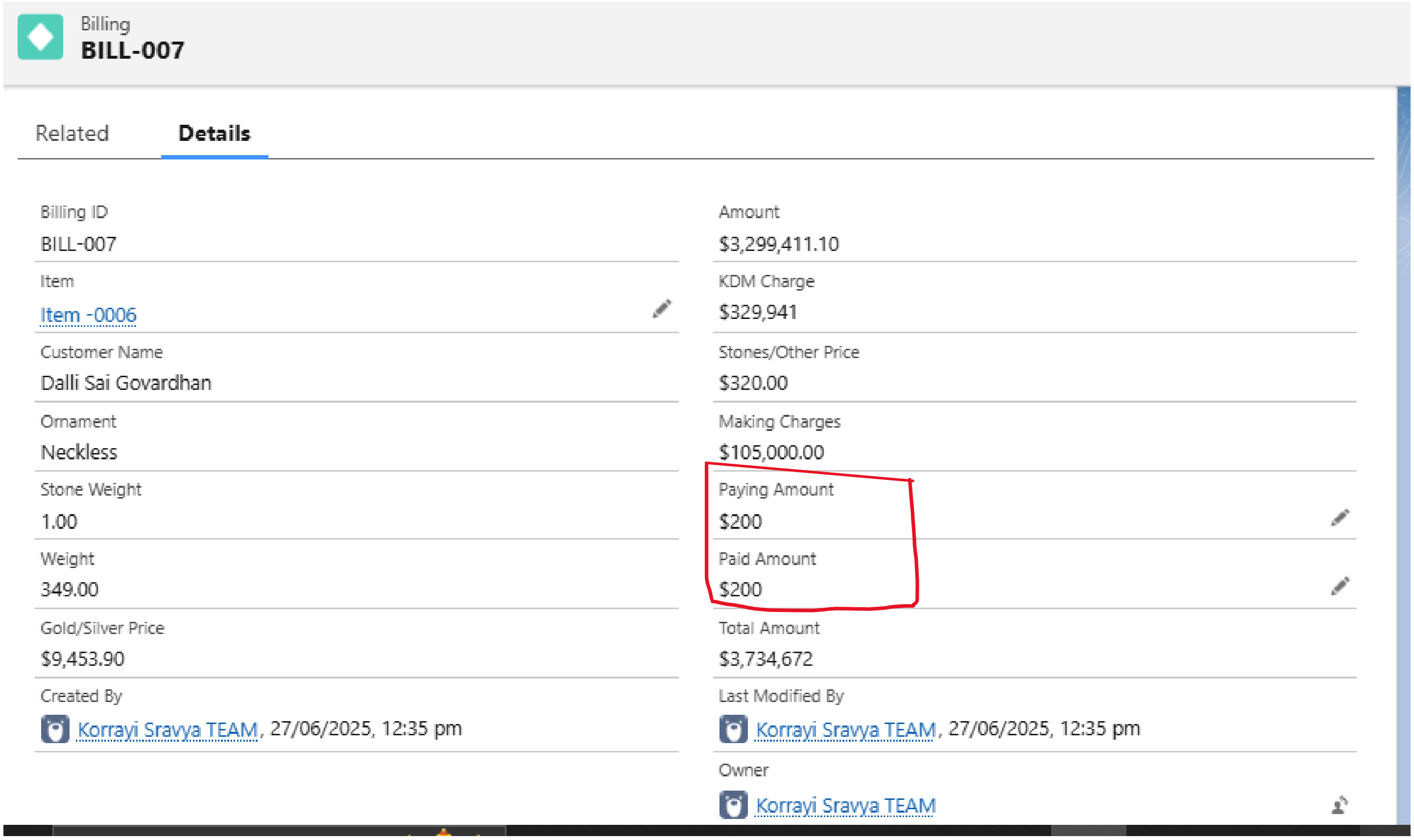
Trigger: Auto-Update Paid Amount

Objective:

Ensure Paid_Amount__c is updated automatically without manual input

Logic:

- Trigger runs on insert/update of Payment record
- Matches corresponding Billing__c record
- Updates Paid_Amount__c = sum of payments received



G. Validation Rules

Rule: Prevent Overpayment

Name: Validate_Paid_Amount

Formula:

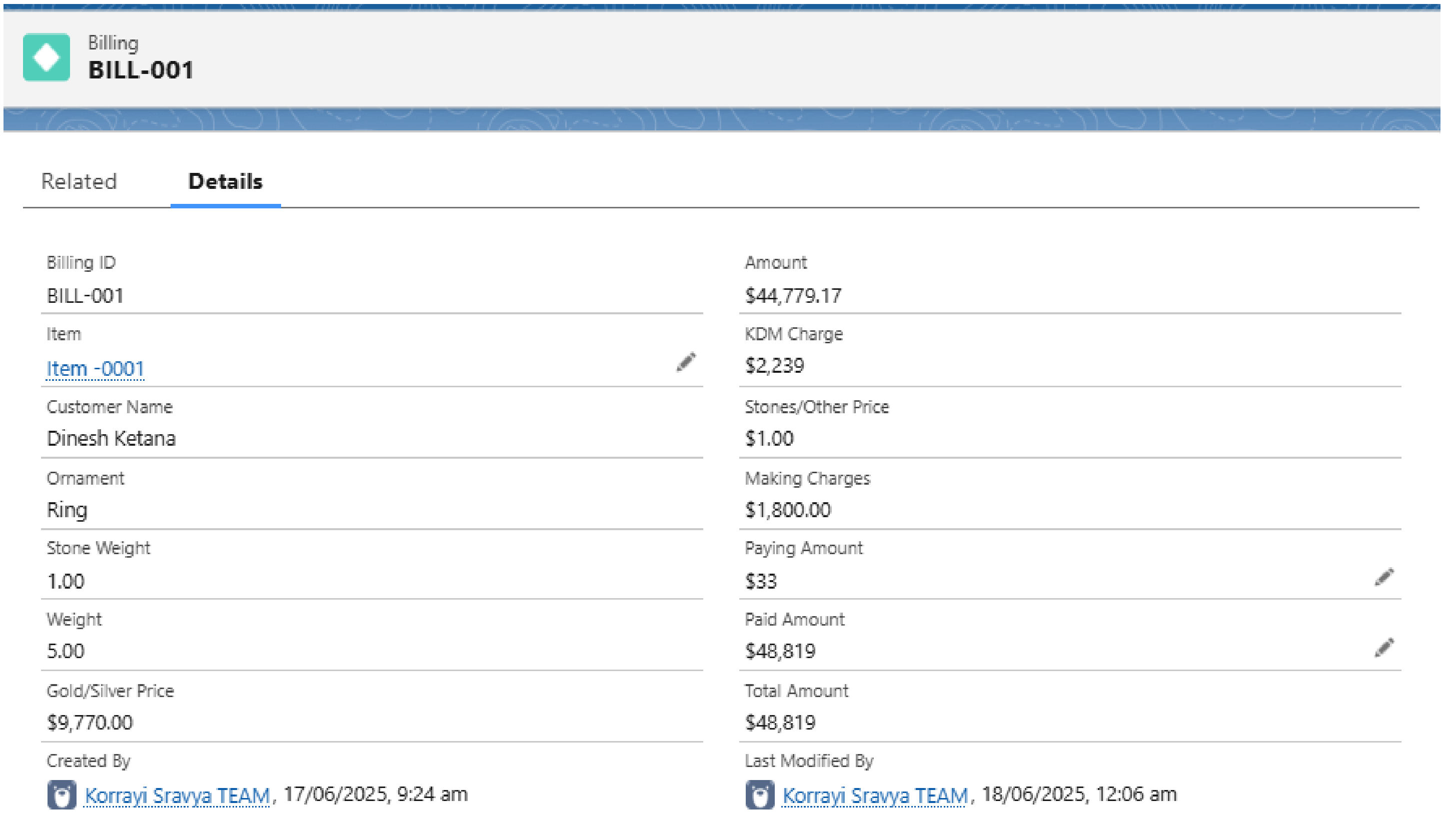
Paid_Amount__c > Total_Amount__c

Error Message:

"Paid amount cannot exceed the total billing amount."

Purpose:

Maintains financial integrity and prevents accidental overcharges



H. Object Relationship Diagram (ERD)

Purpose

The Entity Relationship Diagram (ERD) is a crucial architectural representation of how objects in the CRM for Jewelry Management system relate to each other. This diagram helps visualize the database schema, illustrating the lookup relationships between various custom objects. It supports better understanding of the data flow, record hierarchy, and how modules like Billing, Orders, Items, and Customers are interlinked.

ERD Overview of Custom Objects and Relationships

Parent Object	Child Object	Relationship Type	API Field Name	Description
Jewel_Customer__c	Customer_Order__c	Lookup	Linked_Customer__c	Connects customer data to each order placed.
Item__c	Customer_Order__c	Lookup	Ordered_Item__c	Links the item (gold/silver ornament) to the customer's order.
Jewel_Customer__c	Billing__c	Lookup	Related_Customer__c (example)	Tracks the customer

				associated with each bill.
Item__c	Billing__c	Lookup	Related_Item__c (example)	Maps the jewelry item to its billing record.
Price__c	(Referenced in) Item__c / Billing__c	Lookup or Formula Reference	Gold_Price__c / Silver_Price__c	Not directly related via lookup, but referenced in formula fields for price use.

Relationship Flow Summary

The relationships can be summarized in this logical order:

```
Jewel_Customer__c
        ↓
  Customer_Order__c ← Item__c
        ↓
     Billing__c ← Jewel_Customer__c, Item__c
```

This implies:

- A customer places an order for a specific item
- A billing record is generated for that order, pulling customer and item details
- Price__c is used as a reference for gold/silver pricing in calculations

Visual ERD Illustration (Sample)

While this cannot display an actual image here, you should generate a diagram that looks like this in Draw.io or Lucidchart (example text for your doc):

"The diagram below shows the lookup relationships between core custom objects. Arrows point from child to parent."

```
           +---------------------+
           |  Jewel_Customer__c  |
           +---------------------+
                    ↑         ↑
                    |         |
+-----------------------+  +---------------------+
| Customer_Order__c     |  |  Billing__c         |
| - Ordered_Item__c     |  | - Related_Item__c   |
| - Linked_Customer__c  |  | - Related_Customer__c|
+-----------------------+  +---------------------+
              ↑
              |
        +------------+
        |  Item__c   |
        +------------+
```

Use Case Explanation

Example Scenario:

Dinesh (Jewel_Customer__c) orders a Gold Necklace (Item__c). The system creates a Customer Order (Customer_Order__c) linking both the customer and item. Once the order is processed, a Billing record (Billing__c) is generated with references to the same customer and item. Pricing is calculated using fields and formula referencing values from Price__c (not directly related via lookup)._

How You Can Create It Practically

Option 1: Schema Builder (Salesforce Native Tool)

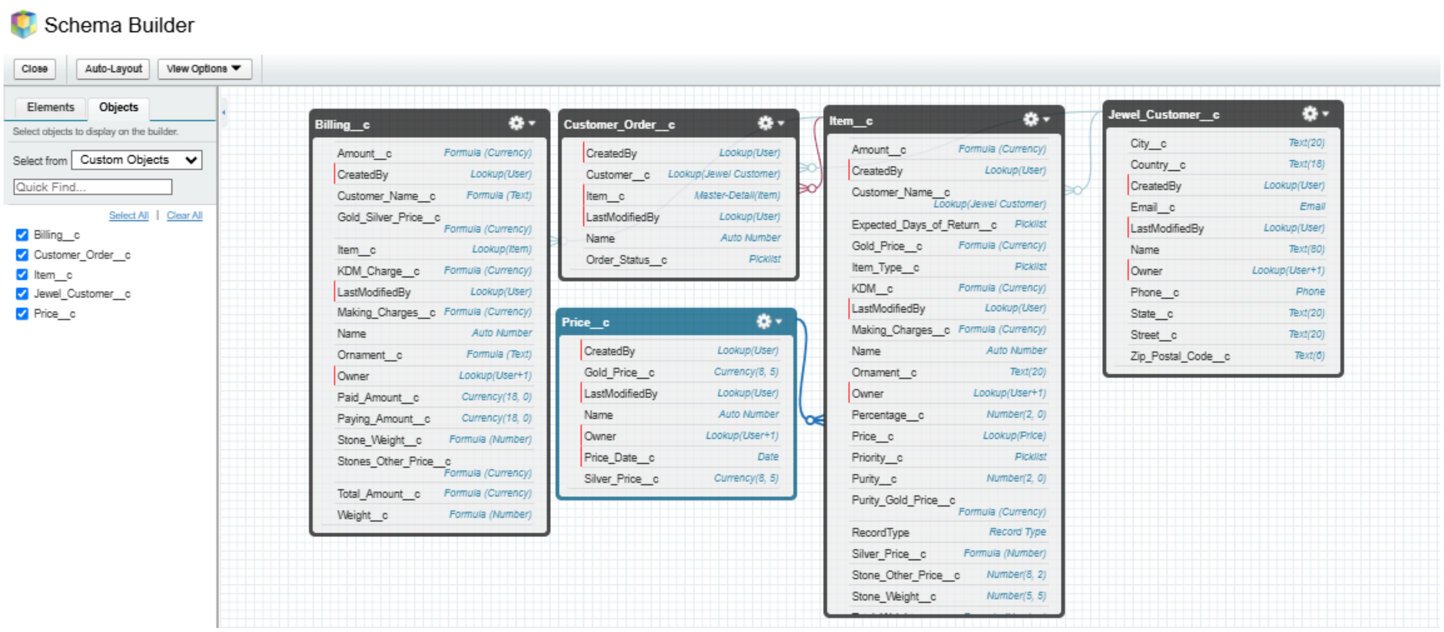
1. Go to Setup → Search Schema Builder
2. Drag your objects: Jewel_Customer__c, Customer_Order__c, Item__c, Billing__c, Price__c
3. View the lookup relationships visually
4. Take a screenshot and paste it into your document

Option 2: Draw.io / Lucidchart

1. Go to https://draw.io or https://lucidchart.com
2. Use Entity shapes to create object boxes
3. Use arrows with labels (e.g., "Lookup") to show relationships

4. Export as PNG or JPEG

5. Insert into your Word/PDF documentation under the “ERD” section



I. User Profiles and Permission Management

Overview:

Profiles in Salesforce determine the level of access and control that a user has over various features and data within the organization. They define permissions related to:

- Object-level access
- Field-level access
- Tab and App visibility
- Apex class and Visualforce page access
- Record Types, Page Layouts
- Login hours and IP restrictions

To ensure data security and task-based functionality for different users in the CRM for Jewel Management system, custom profiles were created and configured.

A. Types of Profiles in Salesforce

1. Standard Profiles (Predefined, cannot be deleted):

- System Administrator
- Standard User

- Read Only
- Marketing User
- Contract Manager
- Solutions Manager

These come with default permissions and are typically used for basic or administrative users.

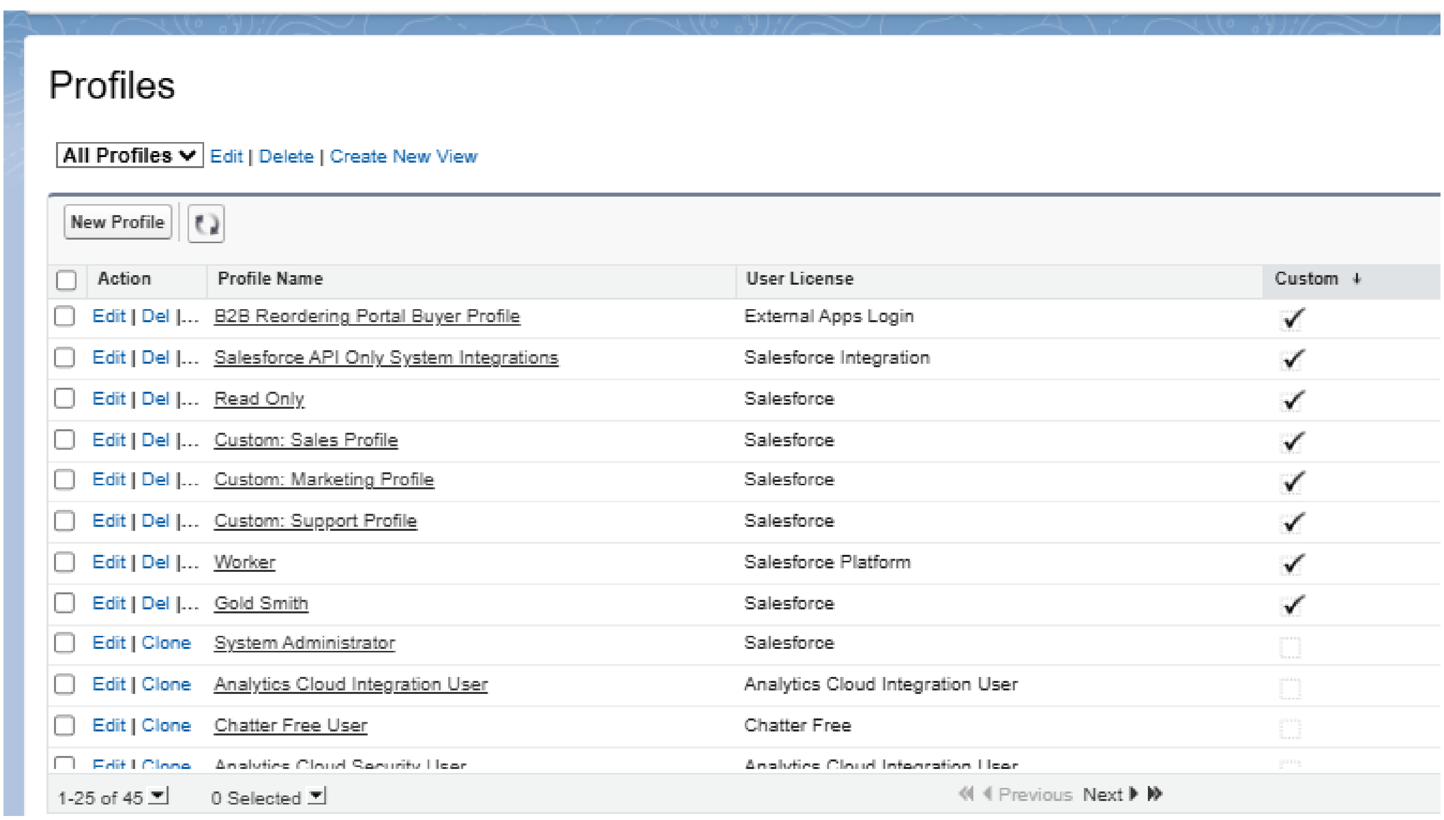
2. Custom Profiles (User-defined, can be deleted if not in use):

- Custom profiles were created to align with organizational roles such as Goldsmith and Worker to provide **role-based access control**.

B. Use Case Scenario: Role-Based Access

The Goldsmith (business owner) requested differentiated access for users based on their **role and responsibilities**:

- **Gold Smith**: Needs full access to manage customers, items, orders, pricing, and billing.
- **Worker**: Needs access to item management and order handling but **should not access sensitive customer or billing data**.


Profiles

All Profiles | Edit | Delete | Create New View

New Profile

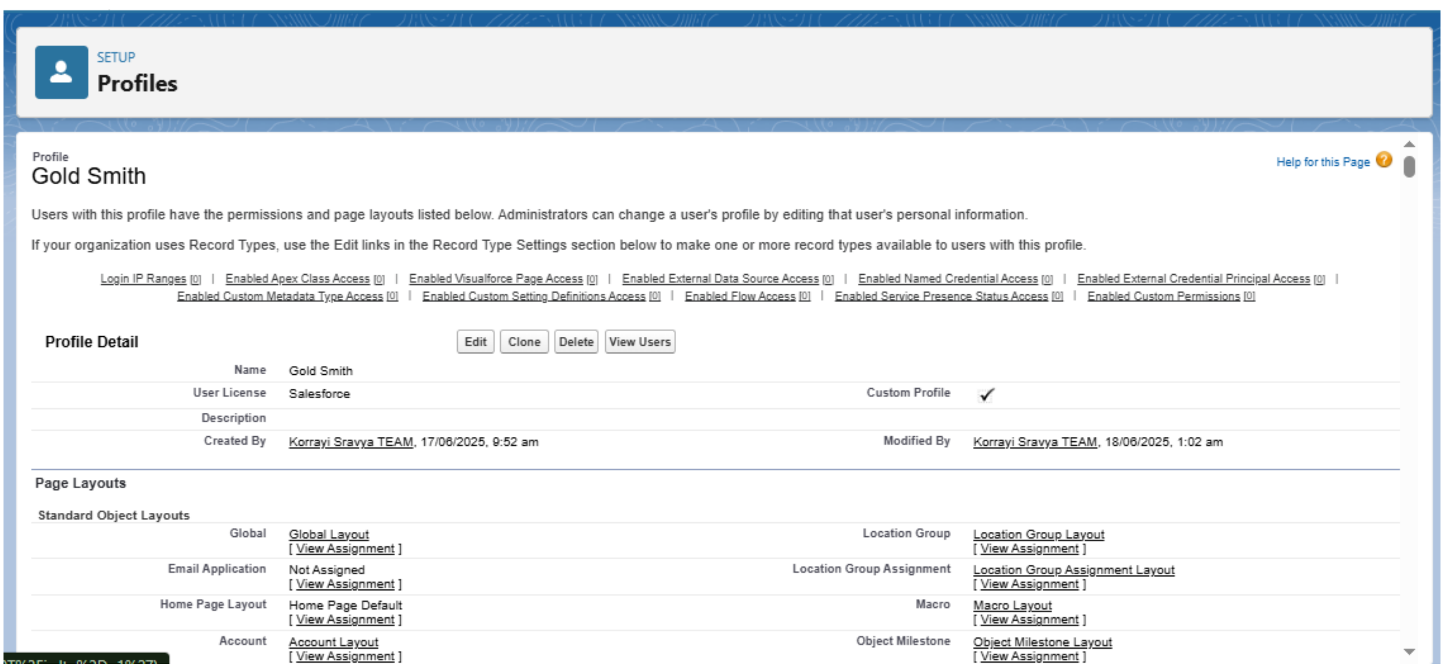
	Action	Profile Name	User License	Custom ↓
☐	Edit | Del |...	B2B Reordering Portal Buyer Profile	External Apps Login	✓
☐	Edit | Del |...	Salesforce API Only System Integrations	Salesforce Integration	✓
☐	Edit | Del |...	Read Only	Salesforce	✓
☐	Edit | Del |...	Custom: Sales Profile	Salesforce	✓
☐	Edit | Del |...	Custom: Marketing Profile	Salesforce	✓
☐	Edit | Del |...	Custom: Support Profile	Salesforce	✓
☐	Edit | Del |...	Worker	Salesforce Platform	✓
☐	Edit | Del |...	Gold Smith	Salesforce	✓
☐	Edit | Clone	System Administrator	Salesforce	☐
☐	Edit | Clone	Analytics Cloud Integration User	Analytics Cloud Integration User	☐
☐	Edit | Clone	Chatter Free User	Chatter Free	☐
☐	Edit | Clone	Analytics Cloud Security User	Analytics Cloud Integration User	☐

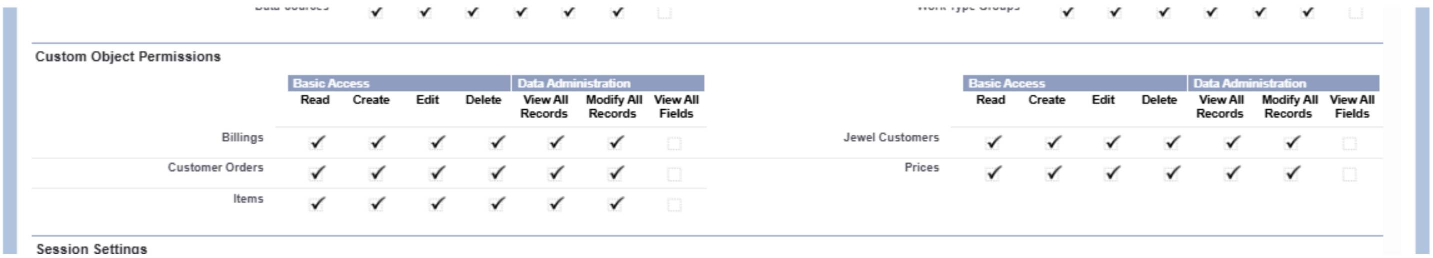
1-25 of 45 · 0 Selected · Previous Next

C. Implementation Activities

1. Gold Smith Profile Creation

- **Objective:** To allow full access to all business-critical objects.
- **Steps:**
    1. Navigate to **Setup → Profiles**
    2. Clone the **System Administrator** profile
    3. Name the new profile: Gold Smith
    4. Save and then click **Edit**
    5. Under **Custom Object Permissions**, provide full CRUD access to:
        - Jewel Customer
        - Item
        - Customer Order
        - Price
        - Billing
    6. Click **Save** to confirm permissions
- **Outcome:** The Gold Smith user can create, edit, view, and delete records across all core objects.





Custom Object Permissions

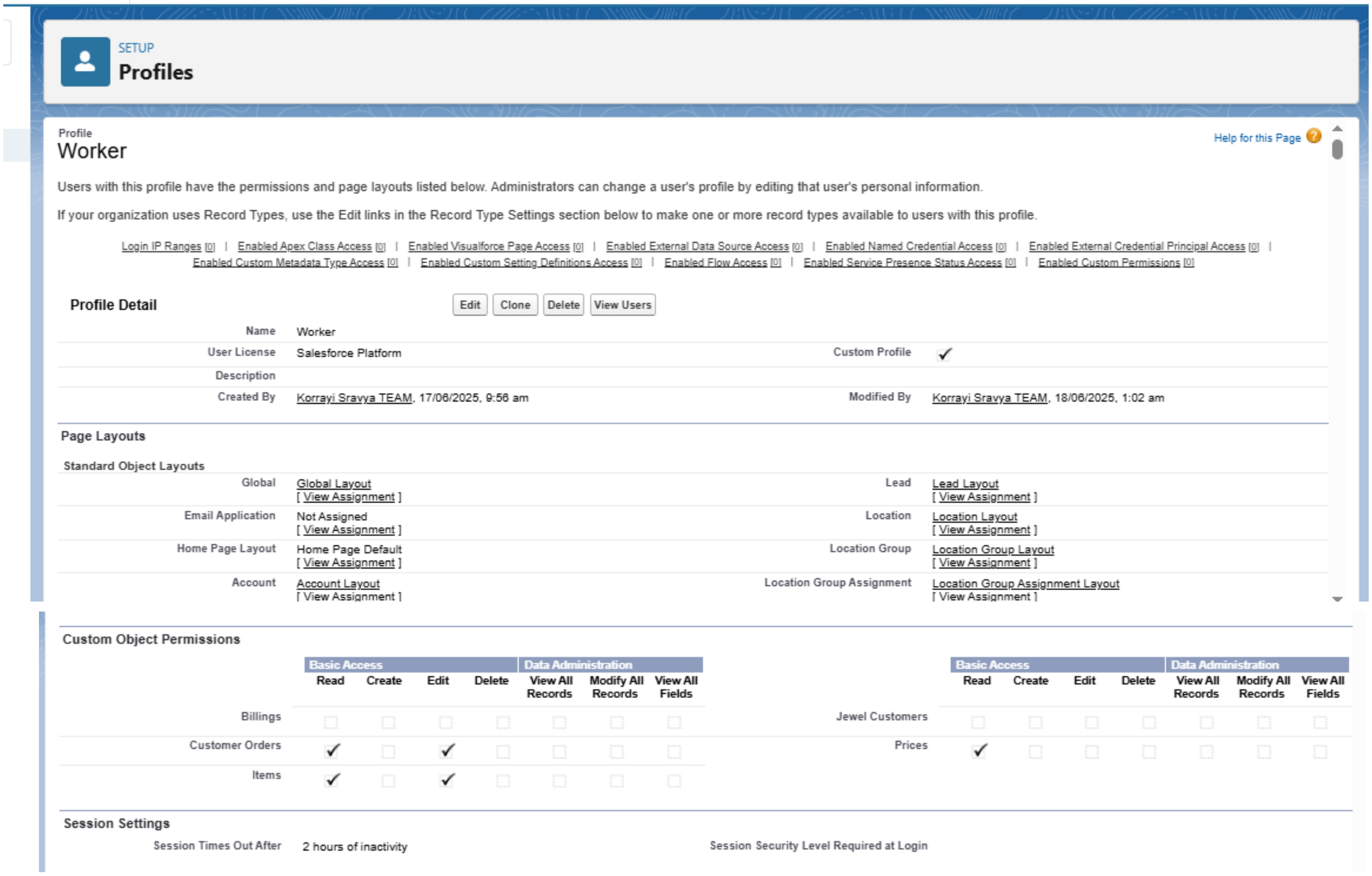
	Basic Access: Read	Create	Edit	Delete	Data Administration: View All Records	Modify All Records	View All Fields
Billings	✓	✓	✓	✓	✓	✓	☐
Customer Orders	✓	✓	✓	✓	✓	✓	☐
Items	✓	✓	✓	✓	✓	✓	☐
Jewel Customers	✓	✓	✓	✓	✓	✓	☐
Prices	✓	✓	✓	✓	✓	✓	☐

Session Settings

2. Worker Profile Creation

- **Objective:** To limit access only to objects necessary for daily operations.
- **Steps:**
  1. Navigate to **Setup → Profiles**
  2. Clone the **Salesforce Platform User** profile
  3. Name the new profile: Worker
  4. Save and then click **Edit**
  5. Under **Custom Object Permissions**, grant access only to:
     - Item
     - Price
     - Customer Order
  6. Deny access to:
     - Jewel Customer
     - Billing
  7. Click **Save**

- **Outcome:** The Worker user can manage inventory and order processing without accessing sensitive financial or customer data.



D. Benefits and Security Measures

- **Data Protection:** Ensures sensitive data like customer information and payment amounts are not accessible by junior staff.
- **Operational Efficiency:** Enables each role to access only the data and tools required for their job function.
- **Scalability:** Profiles can be reused and extended for new roles in the future.

Summary Table – Custom Profiles

Profile Name	Cloned From	Object Access Granted	Notes
Gold Smith	System Administrator	Jewel Customer, Item, Order, Billing, Price	Full Access
Worker	Salesforce Platform User	Item, Price, Customer Order	Restricted – no billing/customer

J. Role Hierarchy and Record-Level Access Control

Overview

In Salesforce, **Roles** are used to control **record-level access** based on a user's position in the organization. Unlike profiles (which control object-level permissions), roles determine **what records a user can *see* and *share*** based on the hierarchy. By assigning users to different roles, administrators can implement **data visibility control** that mimics the company's organizational structure.

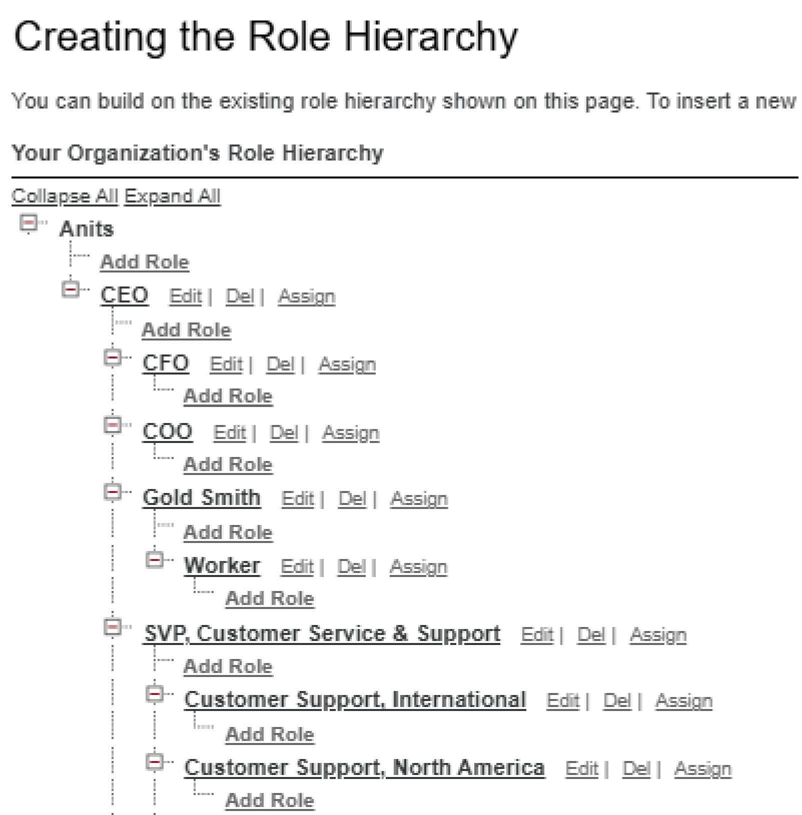
A. Use Case: Role-Based Record Visibility

After successfully setting up profiles to control *permissions and object-level access*, the next step is to configure roles that determine *record visibility* within the organization.

Requirement:
The organization wants to implement a structure where:

- The **Gold Smith (Manager)** has visibility over all records under him.
- The **Worker (Subordinate)** only sees records they own or are explicitly shared with them.

This ensures **secure and hierarchical data access**, improving data control and clarity.



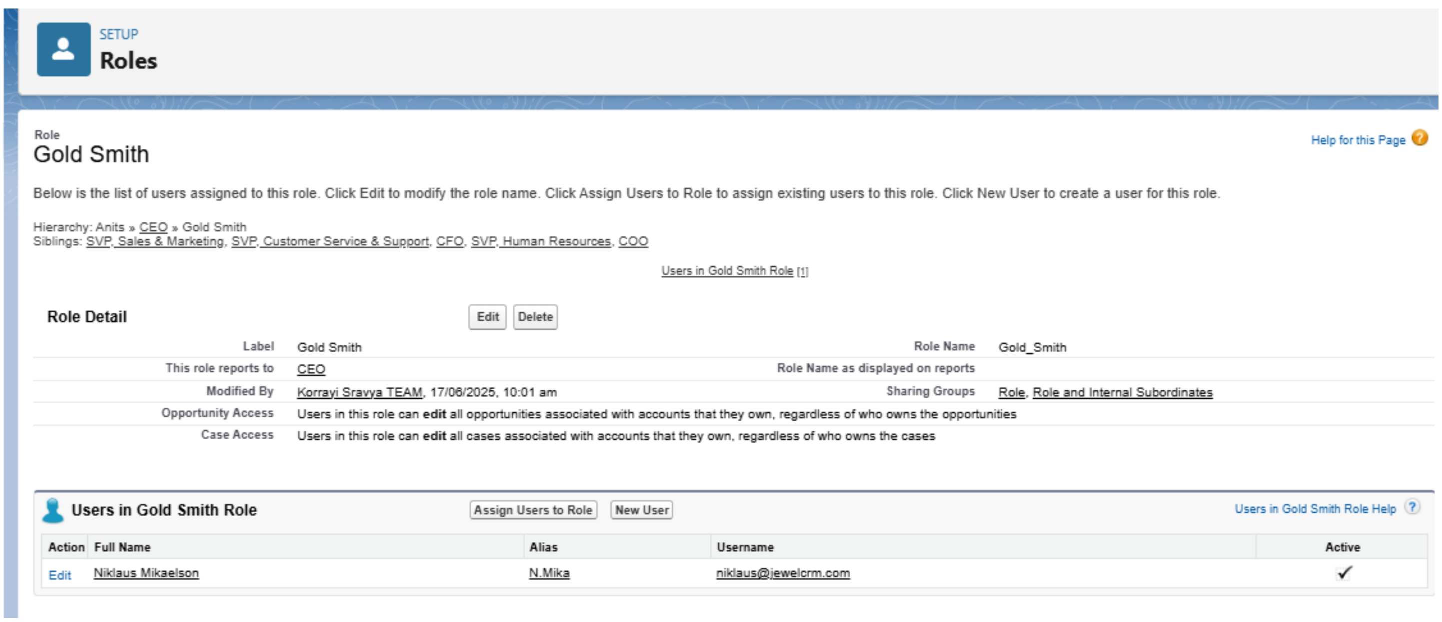
B. Implementation Steps

1. Create Role: Gold Smith

- **Navigation**:
  Go to **Setup → Roles → Set Up Roles**
  Alternatively, use Quick Find and type "Roles".

- **Steps**:

  1. Click on **Expand All** to view the default hierarchy.
  2. Choose the appropriate parent (e.g., CEO or any top-level role).
  3. Click **Add Role** under the selected parent.
  4. Fill the following:
     - **Label**: Gold Smith
     - **Role Name**: (Auto-populated)
     - **This Role Reports To**: Select as per organization (e.g., CEO or another higher role)
  5. Click **Save**



- **Outcome**:
  The Gold Smith role is now part of the hierarchy and can be assigned to users responsible for managing inventory, billing, and customers.

2. Create Role: Worker

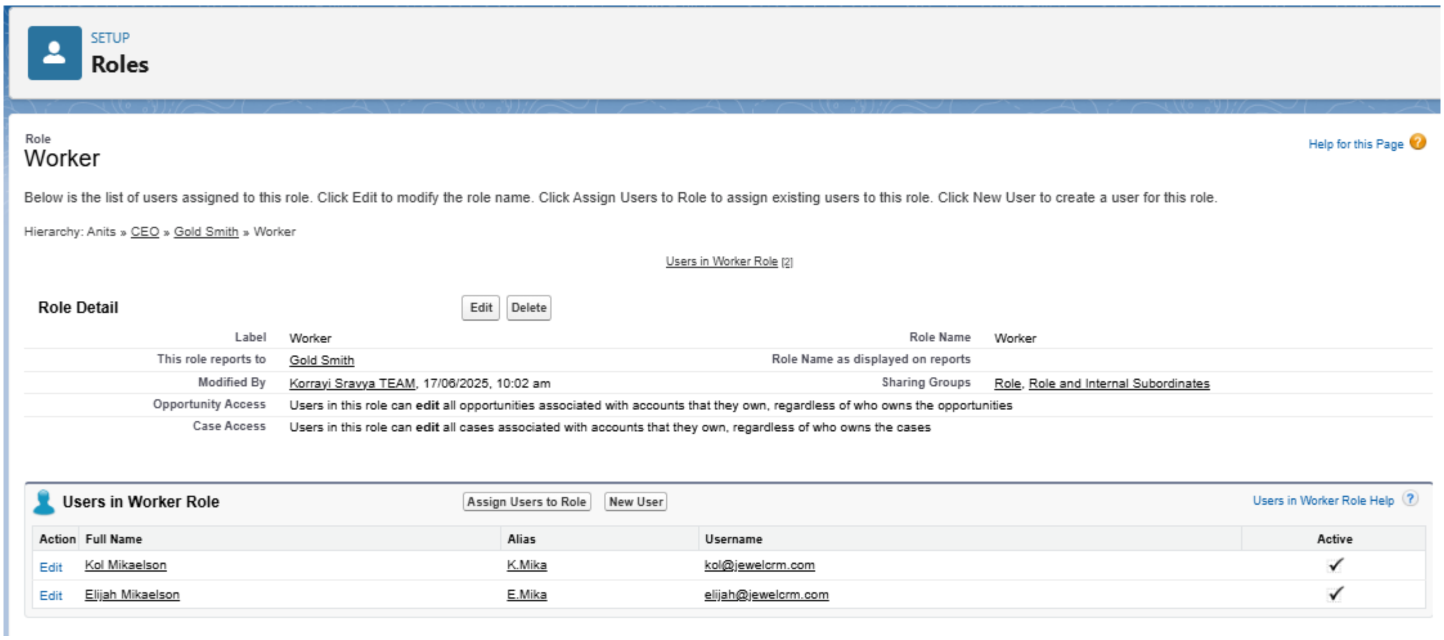
- **Steps**:

  1. Under the **Gold Smith** role, click on **Add Role**.

2. Enter:

   - **Label**: Worker
   - **Role Name**: (Auto-populated)
   - **This Role Reports To**: Gold Smith

3. Click **Save**



- **Outcome**:
  Workers now fall directly under the Gold Smith in the role hierarchy and will only have access to records they own unless shared.

C. Visual Example of Role Hierarchy

```
 ┌─────────────┐
 │    CEO      │
 └──────┬──────┘
        │
 ┌──────▼──────────┐
 │   Gold Smith    │  ← Has visibility over all Worker data
 └──────┬──────────┘
        │
 ┌──────▼──────────┐
 │     Worker      │  ← Sees only own data unless shared
 └─────────────────┘
```

D. Role Assignment to Users

Once roles are created, assign them to users based on their job role in the org:

1. Go to **Setup → Users → Users**
2. Edit the user record
3. Under **Role**, select either:
    - Gold Smith (for managers/owners)
    - Worker (for general staff)
4. Click **Save**

E. Benefits of Role Hierarchy

Feature	Benefit
Record-level visibility	Users can see only the data they need
Managerial oversight	Higher roles can access subordinate records
Secure sharing	Prevents unauthorized access to sensitive records
Collaboration	Enables efficient data flow between team levels

Role	Reports To	Visibility Scope	Assigned To
Gold Smith	CEO/Admin	Full access to all records under hierarchy	Owner, Admin, Supervisor
Worker	Gold Smith	Access to own records	Staff, Entry-level workers

K. User Creation and Access Configuration

Overview

Users in Salesforce are individuals who have a **login ID and profile** and are granted specific access to objects, records, and features. Each user must be assigned a **profile**, and optionally a **role**, to define **what they can see and do** in the system. In this CRM for Jewel Management project, users are created and configured to reflect the actual organizational hierarchy and job responsibilities.

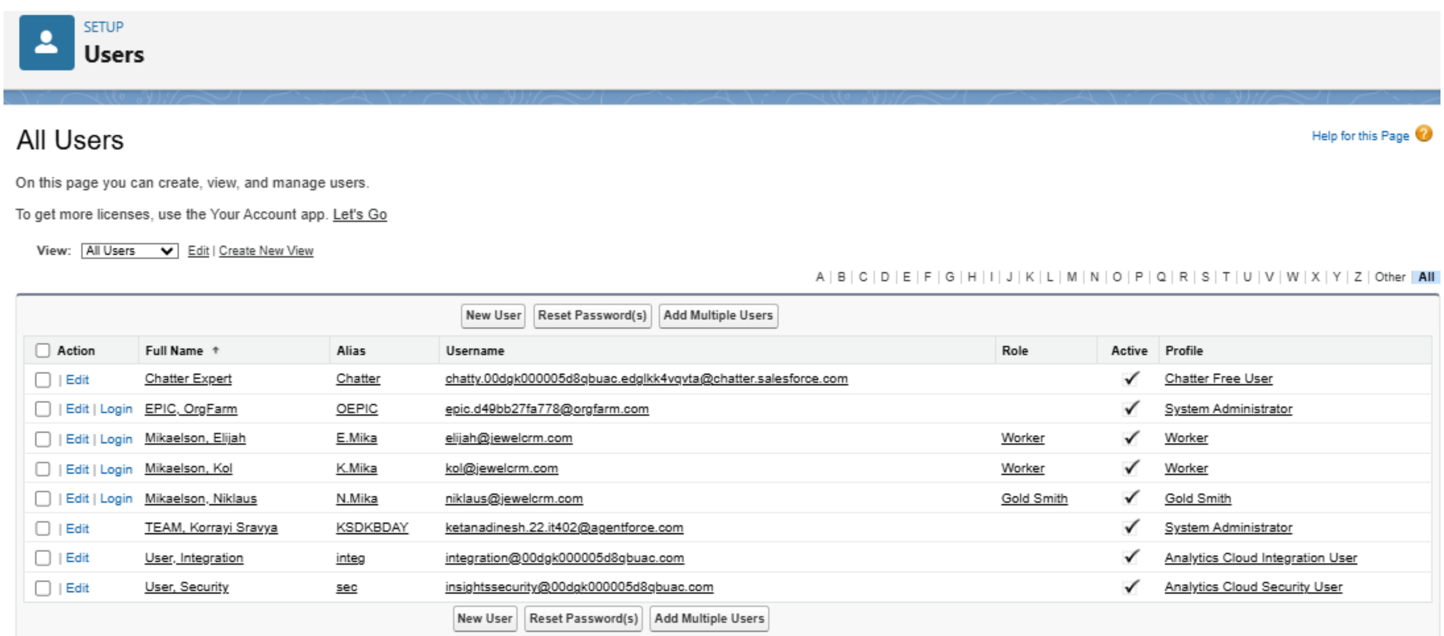
A. Use Case

The Goldsmith requested that users be created for different positions such as managers and workers. These users must be provided with **appropriate access** (via profiles and roles) to perform their job functions, such as managing orders, inventory, and billing.

B. Key User Account Attributes

Each user account in Salesforce includes the following mandatory and configurable details:

Field	Description
First Name	User's first name (optional)
Last Name	User's last name (required)
Alias	Shortened identifier for the user
Email	Email address used for notifications and login
Username	Must be unique across all Salesforce orgs
Nickname	Displayed in the app UI (for communities, etc.)
User License	Determines available features and permissions
Profile	Determines object-level and field-level access
Role	Controls record visibility within the org



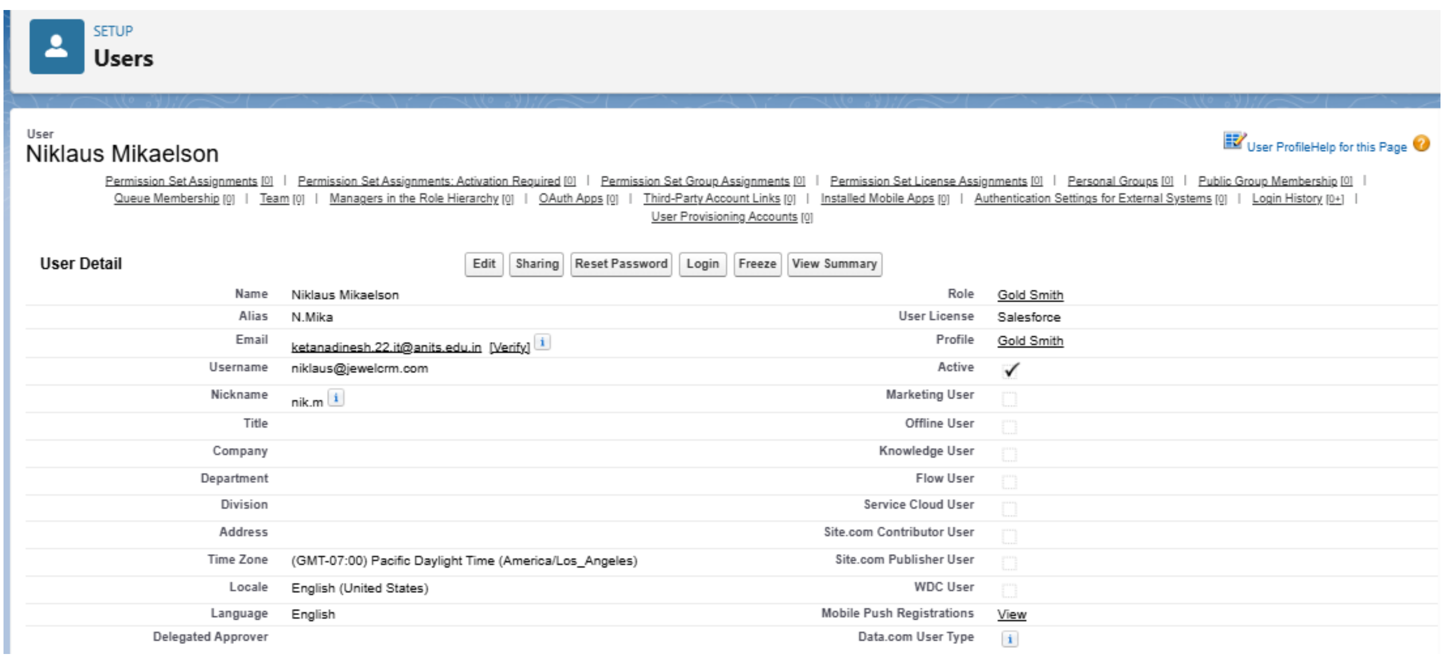
C. User Creation Activities

Activity 1: Create User – Gold Smith

- **Navigation**:
  Setup → Users → Click **New User**
- **Input Details**:

Field	Value
First Name	Niklaus
Last Name	Mikaelson
Alias	n.mikaelson
Email	niklaus@email.com
Username	niklaus@email.com
Nickname	nick
User License	Salesforce
Profile	Gold Smith
Role	Gold Smith

- **Click**: Save



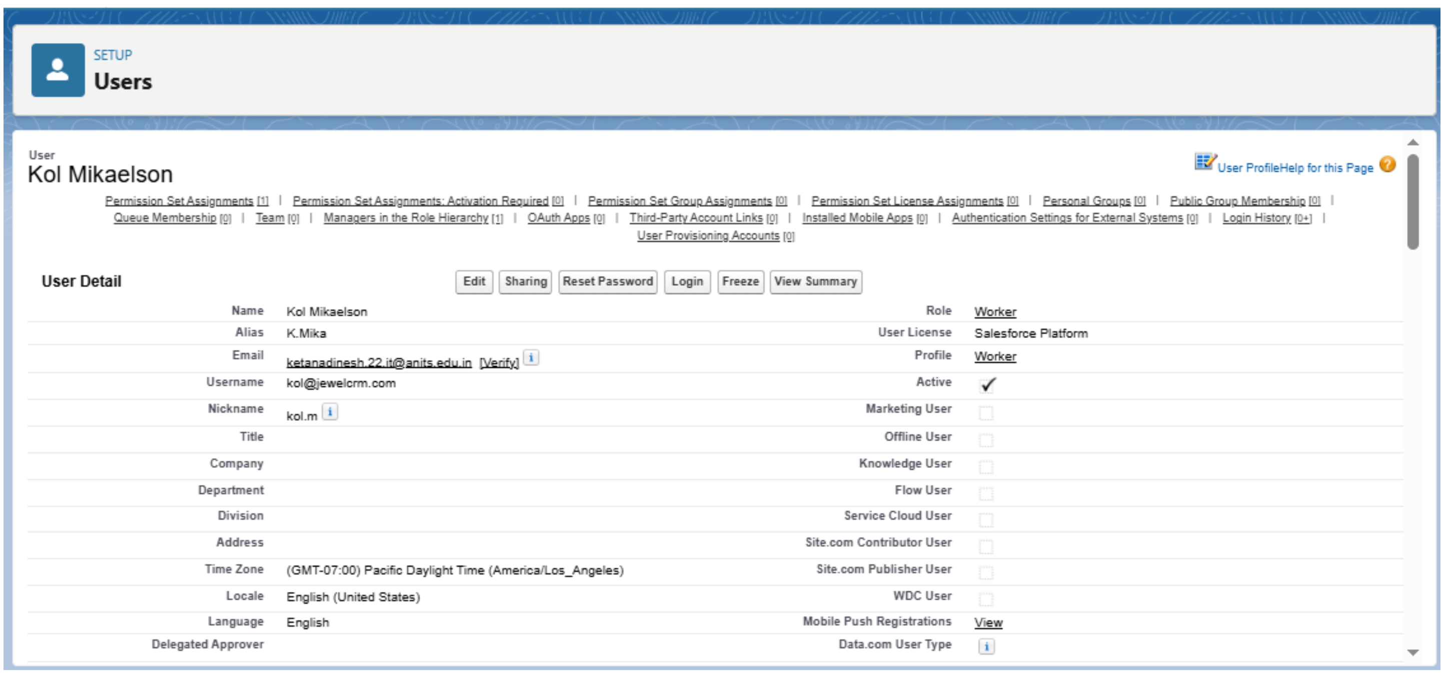
Activity 2: Create User – Worker

- **Navigation**:
  Setup → Users → Click **New User**

- **Input Details**:

Field	Value
First Name	Kol
Last Name	Mikaelson
Alias	k.mikaelson
Email	kol@email.com
Username	kol@email.com
Nickname	kolm
User License	Salesforce Platform
Profile	Worker
Role	Worker

- **Click**: Save



D. Activity 3: Create Additional Users

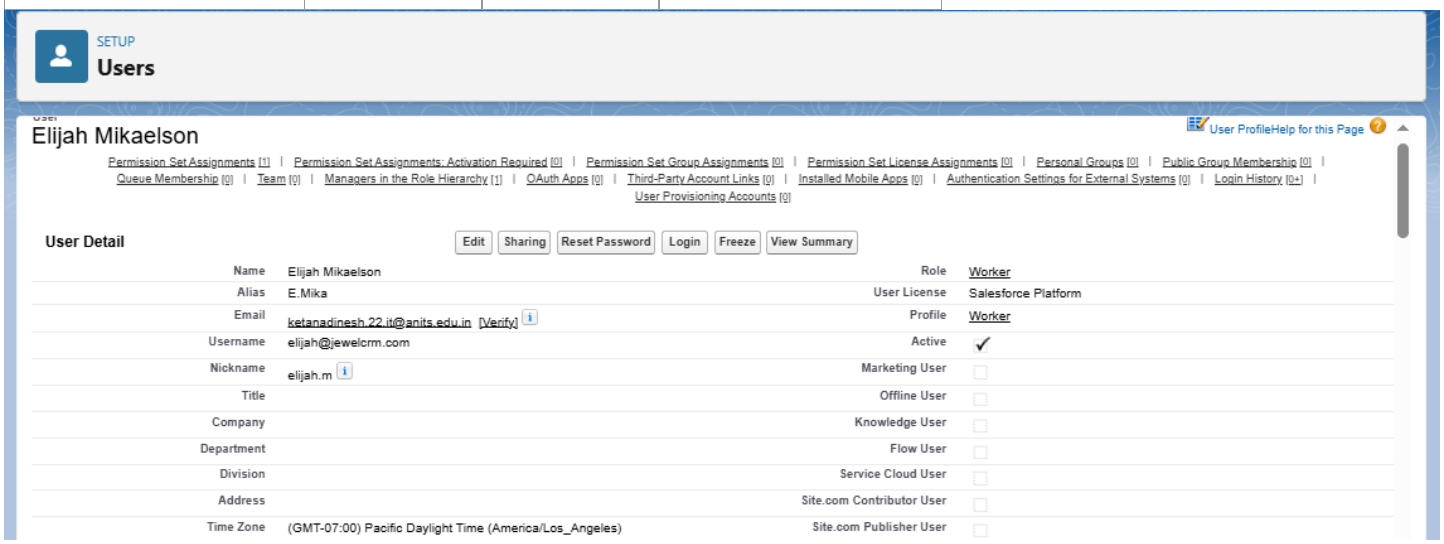
Following the same steps as above, create **two more users** using the **Worker** profile. Ensure:

- Different unique usernames (e.g., rebekah@email.com, elijah@email.com)
- Email IDs may be placeholders or real for demo purposes

- Assign both the **Worker profile** and **Worker role**

E. Summary of Users Created

User Name	Role	Profile	License
Niklaus Mikaelson	Gold Smith	Gold Smith	Salesforce
Kol Mikaelson	Worker	Worker	Salesforce Platform
Rebekah Mikaelson	Worker	Worker	Salesforce Platform
Elijah Mikaelson	Worker	Worker	Salesforce Platform



Benefits of User Management in Salesforce

- Enables **secure access control** tailored to user responsibilities.
- Ensures **data segregation** between managerial and operational roles.
- Supports **auditing, tracking, and accountability** for transactions.
- Simplifies **user-based automation**, email alerts, and workflow targeting.

L. Record Types Implementation

Overview

Record Types in Salesforce allow administrators to display **different page layouts**, **picklist values**, and **field-level requirements** for the same object. This feature helps organizations **customize data entry forms** based on user roles, record category (e.g., Gold vs Silver), or business process variations—all while using the same object model.

In the CRM for Jewel Management project, **Record Types** were created for **Gold** and **Silver** jewelry items to simplify the user experience, ensuring that users only see and input information relevant to the specific ornament type.

Use Case

The organization required **customized forms** for different types of items (Gold vs Silver). The GoldSmith requested that data entry should be simplified by showing only the relevant fields when creating or editing records. To solve this:

- A **Gold Record Type** was created using a layout optimized for gold-specific fields.
- A **Silver Record Type** was created using a layout optimized for silver-specific fields.
  This helps streamline data entry and avoid confusion.

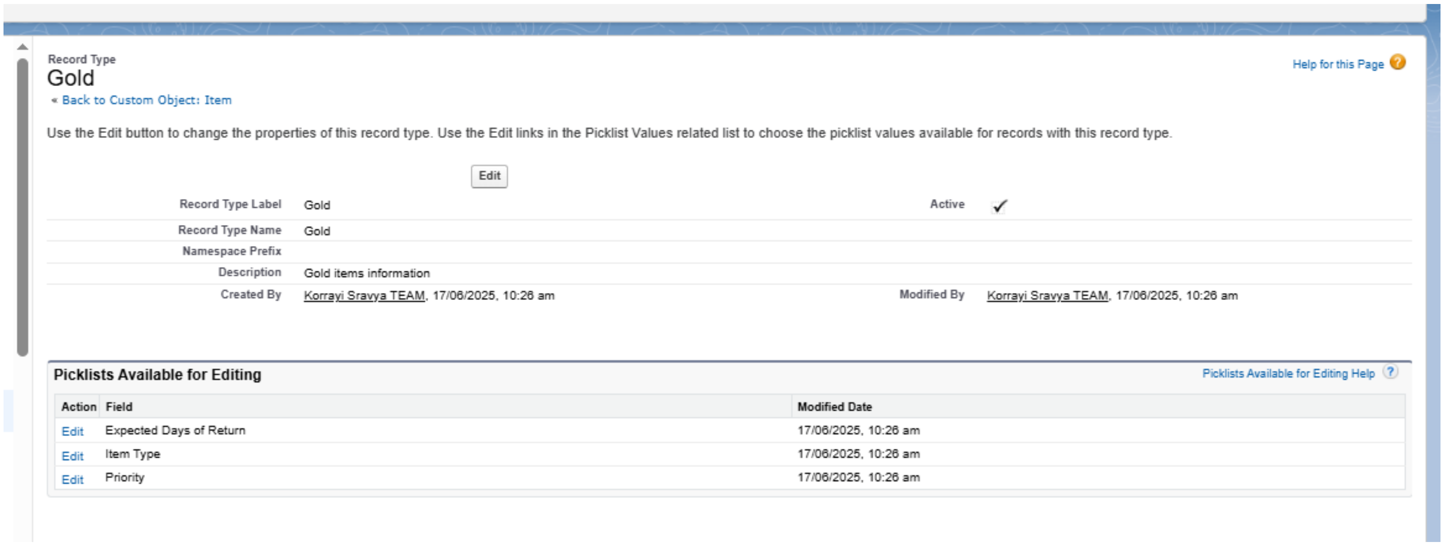
Activity 1: Create Record Type – Gold

Steps:

1. **Navigate to Setup** → Object Manager → Search for **Item**
2. Click on **Record Types** under the Item object.
3. Click **New** to create a new Record Type.
4. Fill the following details:
    - **Existing Record Type**: Master
    - **Record Type Label**: Gold
    - **Record Type Name**: Auto-populated as Gold
    - **Description**: Gold items information
5. **Uncheck** "Make Available" for all profiles (to customize visibility).
6. **Check Access** for the following profiles only:
    - Gold Smith
    - Worker
    - System Administrator
7. Click **Next**
8. Choose: "**Apply a different layout for each profile**"
9. Assign the following layouts:
    - **Gold Smith** → Page Layout for Gold

- **Worker** → Page Layout for Gold
- **System Administrator** → Page Layout for Gold

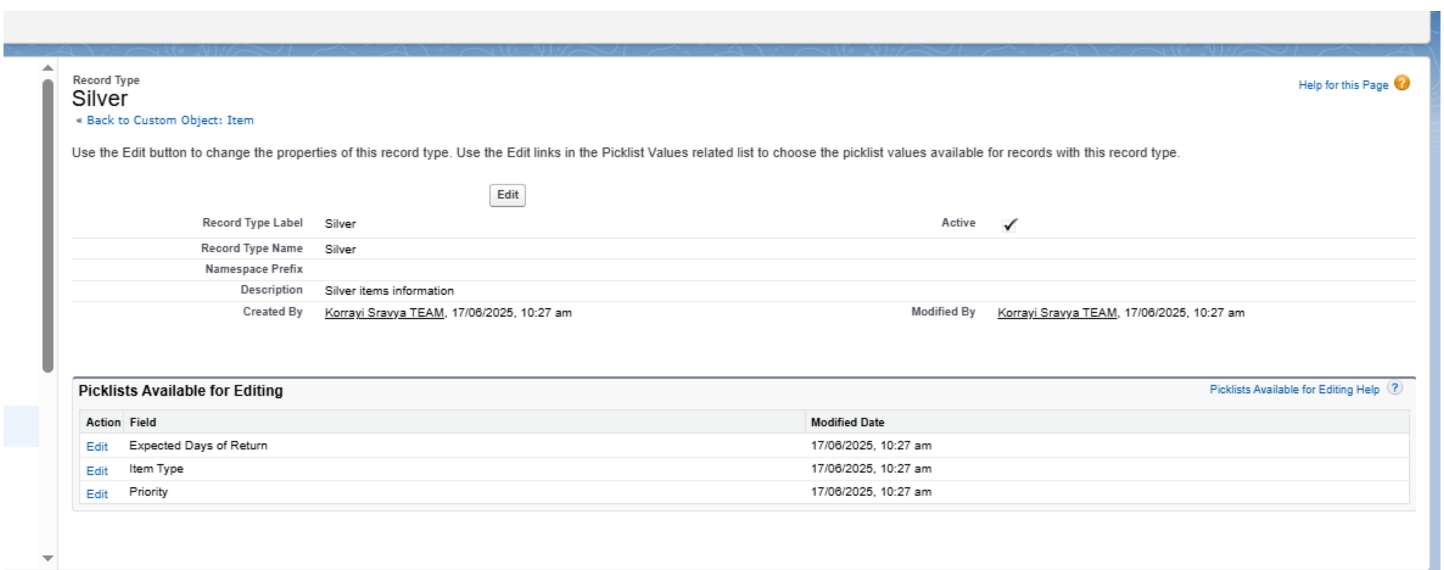
10. Click **Save & New**

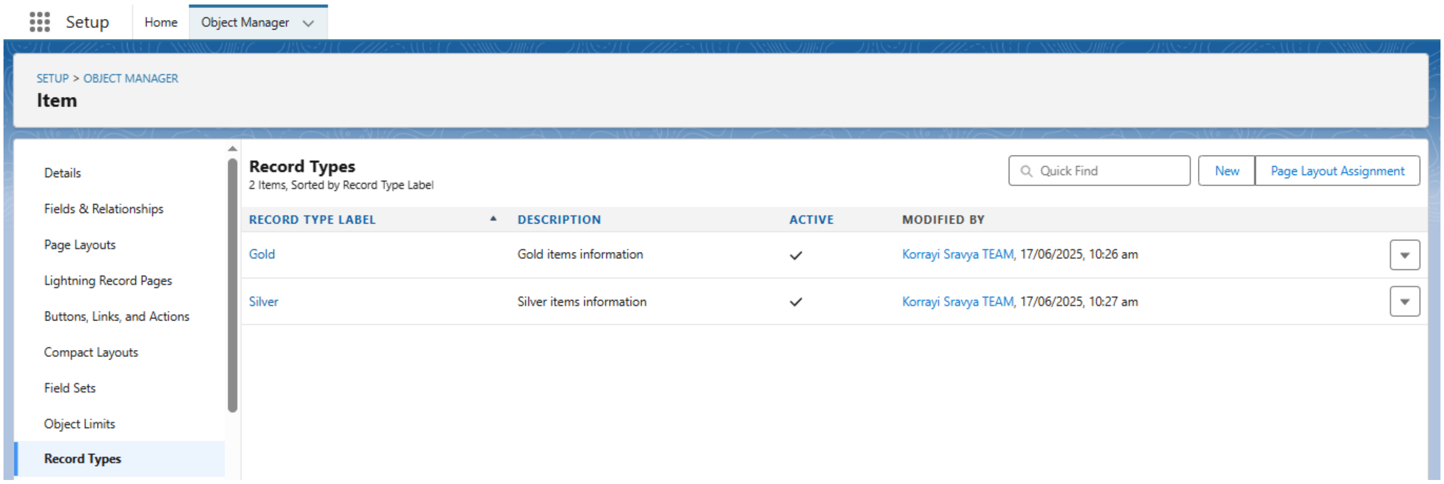


Activity 2: Create Record Type – Silver

Repeat the steps from Activity 1 with the following changes:

- **Record Type Label**: Silver
- **Description**: Silver items information
- Assign layout as **Page Layout for Silver** for all selected profiles (Gold Smith, Worker, System Administrator).





Benefits of Using Record Types

- Customizes the **user interface** based on item category (Gold/Silver)
- Reduces **data entry errors** by showing only relevant fields
- Simplifies **training and onboarding** for workers and staff
- Improves **data accuracy and reporting** by segmenting records properly

M. Permission Sets Configuration

Overview

Permission Sets in Salesforce provide a way to grant users access to specific objects, fields, or functionality **without altering their profile**. Unlike profiles (which are limited to one per user), **multiple permission sets can be assigned** to users, allowing more flexible and scalable access management.

In the **CRM for Jewel Management** system, we used permission sets to **grant additional access rights to users with the Worker profile**. This ensures they can handle Gold and Silver item records while maintaining overall profile-based access restrictions.

Use Case

While users with the **Worker profile** had basic access, they also needed permissions to:

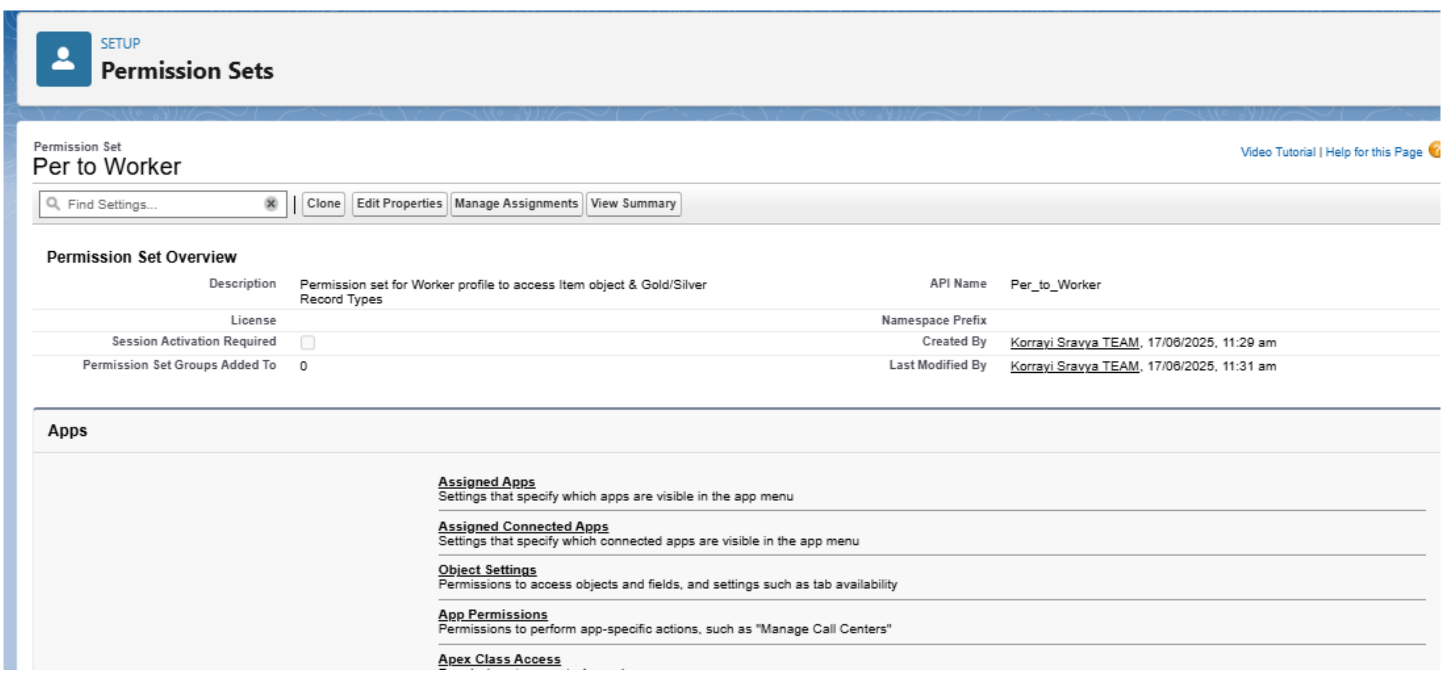
- View and work on **Gold** and **Silver** item records
- Create and update inventory items with appropriate record types

Instead of modifying the Worker profile, we created a **Permission Set** to grant this functionality without disrupting role-based access structure.

Activity: Create and Assign Permission Set to Worker Users

Step 1: Create Permission Set

1. Go to **Setup** → search **Permission Sets** in Quick Find box → click **New**.
2. Enter the following:
   - **Label**: Per to Worker
   - **API Name**: (auto-filled)
3. Click **Save**

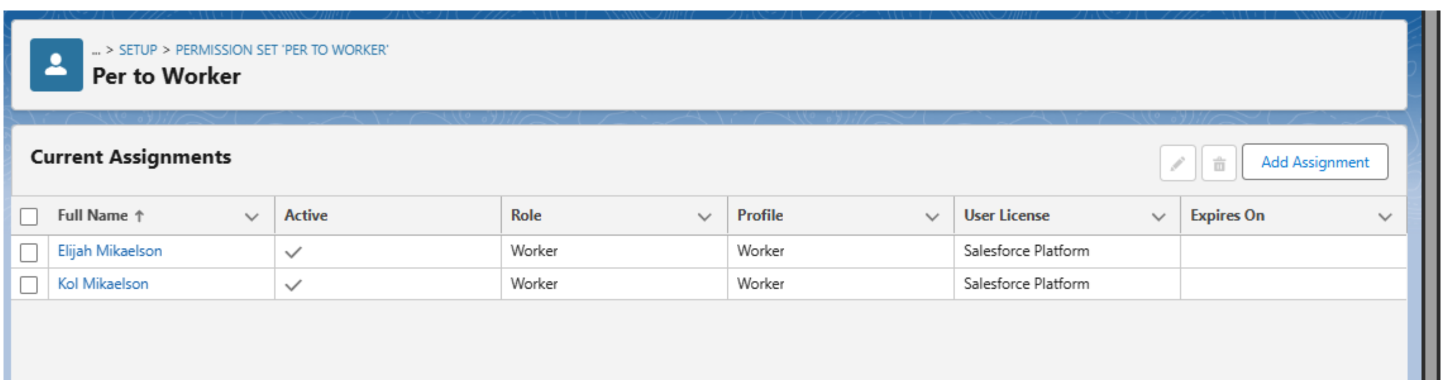


Step 2: Configure Object Permissions

1. On the Permission Set detail page, scroll down to **Apps** → click **Object Settings**.
2. Click on **Item** → then click **Edit**.
3. Under **Item: Record Type Assignments**, check both:
   - Gold
   - Silver
4. Under **Object Permissions**, enable:
   - **Read**
   - **Edit**
   - **Create**
5. Click **Save**

Step 3: Assign Permission Set to Users

1. Back on the Permission Set page, click **Manage Assignments**.
2. Click **Add Assignment**
3. From the user list, select users who are assigned the **Worker** profile (e.g., *Kol Mikaelson*, *any other Worker role users*)
4. Click **Next** → **Assign** → **Done**



Benefits of Using Permission Sets

- Extends access without compromising the **principle of least privilege**
- Ensures **Worker users** can manage both Gold and Silver items as needed
- Allows **granular control** over permissions without cloning or editing profiles

- **Future-proof**: Easily assign the same access to new users without editing the base profile

N. Lightning App Setup

Overview

A **Lightning App** in Salesforce provides a consolidated, branded workspace that allows users to access related tabs, objects, and features efficiently. For the *CRM for Jewel Management* project, a custom Lightning App was created to organize navigation and improve user productivity. It helps different users (e.g., Gold Smith, Worker) seamlessly interact with custom objects like Jewel Customer, Item, Billing, etc.

Use Case

After building a complete data model for jewelry management (objects, relationships, automation), it is essential that users can navigate the system easily. This Lightning App ensures that users can access only the relevant tabs they need to perform their roles effectively within the CRM system.

Steps to Create the Lightning App

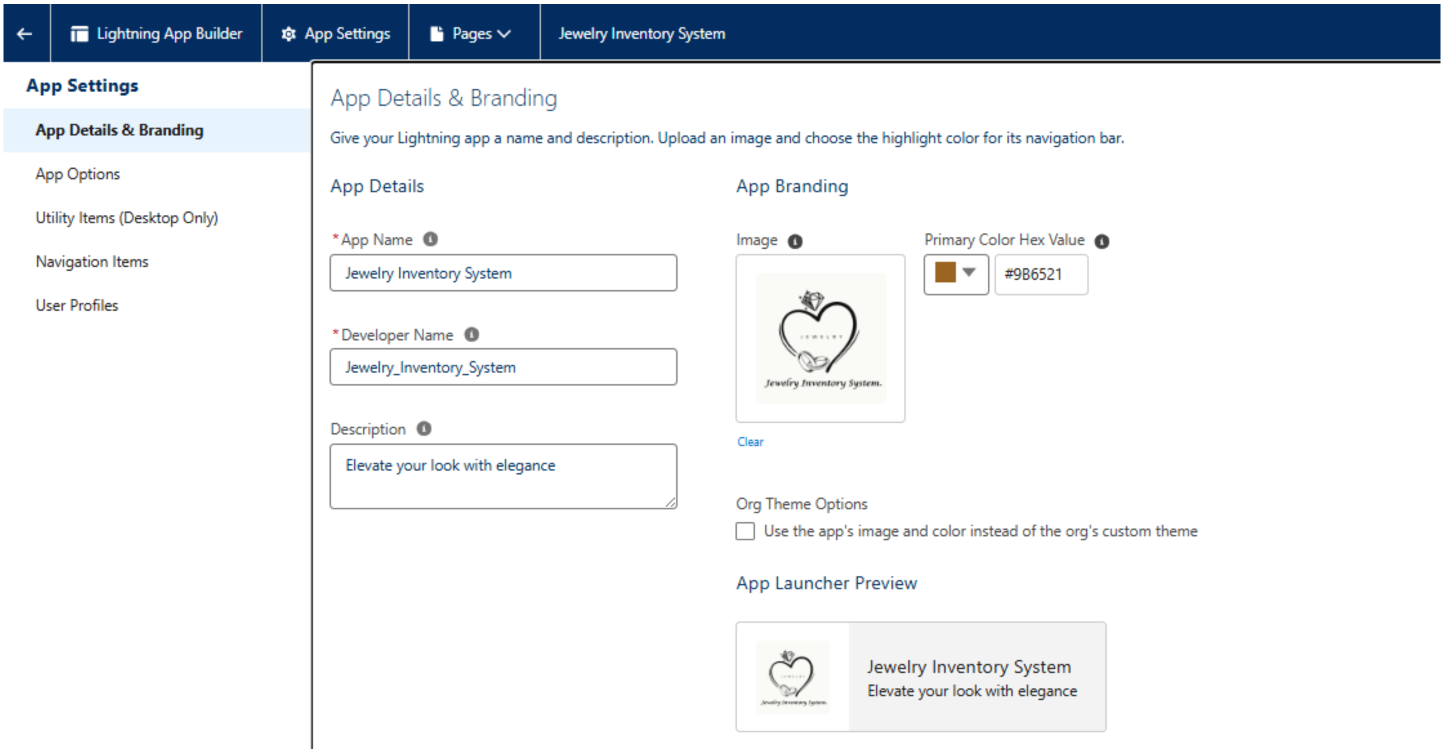
Step 1: Access the App Manager

- Navigate to:
  Setup → Quick Find → App Manager
- Click **New Lightning App**

Step 2: App Details & Branding

- **App Name:** Jewelry Inventory System
- **Developer Name:** Auto-populated

- **Description:** Elevate your look with elegance

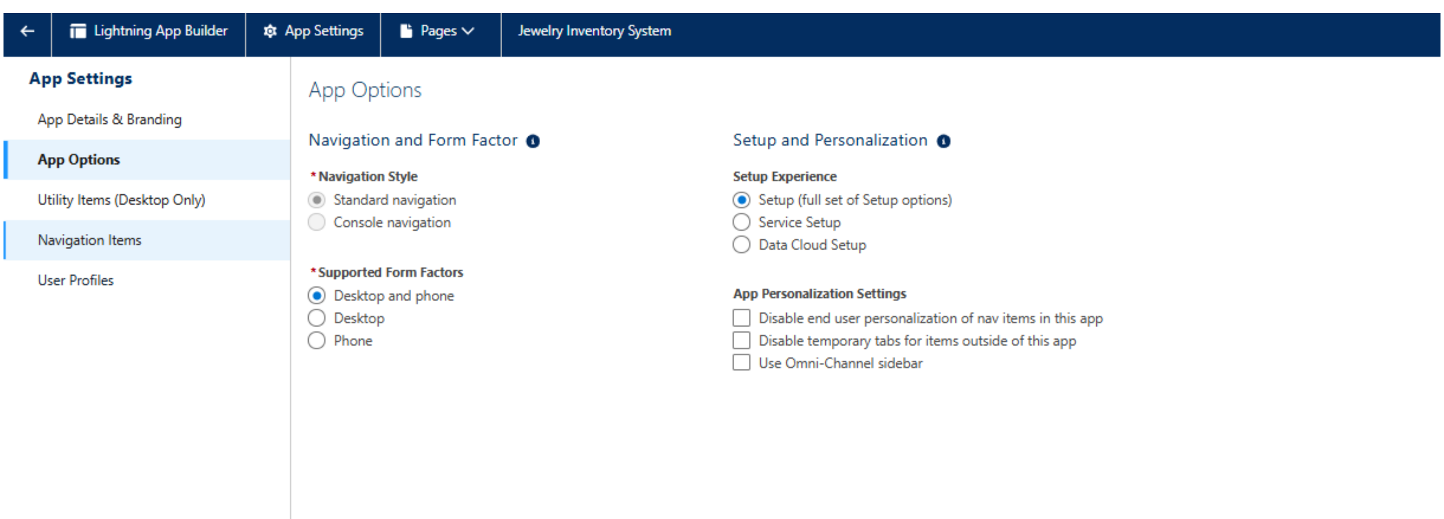


- **Logo:** *(Optional)* Upload if branding image is available
- **Primary Color Hex Value:** Keep default or choose a brand color (e.g., #006DCC)

Click **Next**

Step 3: App Options

- **Navigation Style:** Console Navigation
  - Rationale: Allows multi-tabbed workspace for efficient multitasking
  - Click **Next**



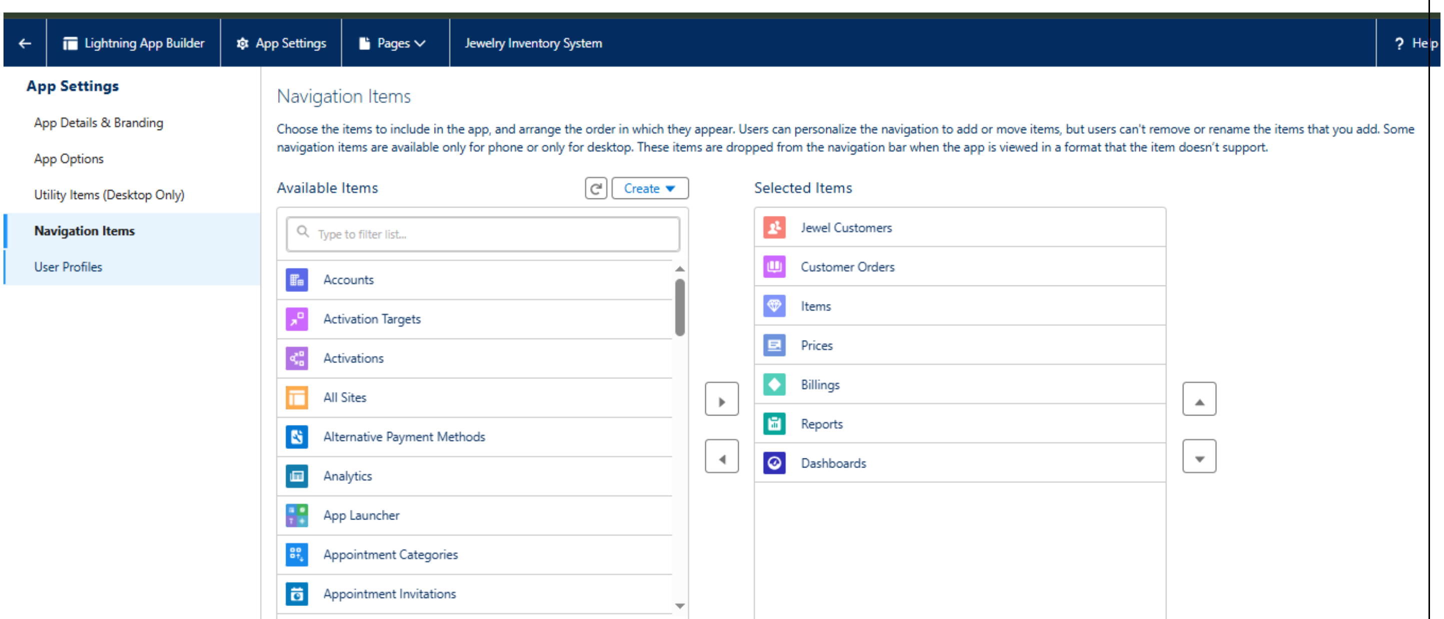
Step 4: Utility Items

- Leave utility items as default *(e.g., Recent Items, History)*

Click **Next**

Step 5: Navigation Items

- Use the left-hand **Available Items** list to search and add:
    - Jewel Customer
    - Item
    - Customer Order
    - Price
    - Billing
    - Reports
    - Dashboards
- Use the right arrow ➞ to move selected items into the **Selected Items** list *(This defines the tabs visible to users in the app)*
- Click **Next**



Step 6: Assign User Profiles

- In the **Profile Assignment** step:
    - Search and select profiles like:

        - System Administrator
        - Gold Smith
        - Worker
- Use the right arrow ➞ to add selected profiles
- Click **Save & Finish**

3. Deployment Process

All components were developed and tested in the Salesforce Sandbox environment and then deployed to the Production Org using the following steps:

- **Change Set Creation:**
  A Change Set was created that included custom objects, fields, validation rules, flows, and triggers.
- **Component Validation:**
  The components were validated in the target environment to ensure compatibility and no dependency conflicts.
- **Deployment Execution:**
  The Change Set was deployed successfully with all components reflected in the live system.
- **Post-Deployment Testing:**
  Manual testing and sample record creation were conducted to validate system behavior.

4. Dataset

Since no external datasets were used, the testing and demonstration were performed using sample data records entered into the Salesforce environment for each object.

Sample Data Records:

- **Customers:**
    - Dinesh Ketana, dinesh@gmail.com
    - Dalli Sai Govardhan, govardhan@gmail.com
- **Items:**
    - Gold Necklace – 20g – ₹1,20,000
    - Silver Ring – 10g – ₹8,000
- **Orders and Billing:**
    - Orders were placed for both items by both customers. , Bills were generated with taxes and other charges.