APPLICATION TO MAKE THE GAS FILLING STATION EASY USING CRM (DEVELOPER)

ABSTRACT :

The Gas Filling Store CRM Application is a specialized software solution engineered to revolutionize the gas filling industry. It functions as a complete customer and operational management system, moving beyond simple point-of-sale functionality. The application's core mission is twofold: to significantly enhance the customer experience through personalized service and efficient transaction processing, and to optimize store operations by digitizing tasks like inventory tracking, customer history logging, and staff management. This development effort aims to create a robust, user-friendly platform that integrates essential CRM features to boost efficiency, foster customer loyalty, and ultimately make the gas filling process easier for both the store and the customer.

PROJECT OBJECTIVES :

The primary objectives of this Salesforce project were centered on transforming the gas filling industry through a powerful CRM solution. The key goals include :

- **Centralize Customer Data**: Establish a unified platform to store and manage all customer information, including contact details, purchase history, and service interactions, creating a single source of truth.
- **Improve Customer Service**: Enable quick access to comprehensive customer data for service personnel, allowing them to provide more personalized and efficient support.
- **Streamline Operational Workflows**: Automate key business processes such as service requests, complaint handling, and loyalty program management to reduce manual effort and boost efficiency.
- **Enhance Data Analytics and Reporting**: Provide a robust framework for generating insights into customer behavior, service performance, and operational efficiency, empowering informed, data-driven decisions.
- **Increase Customer Loyalty**: Implement features for managing loyalty programs and delivering personalized offers and communications to retain customers and build long-term relationships.

SALESFORCE :

Salesforce is a **customer success platform** that helps businesses sell, service, market, and connect with customers. It offers **productivity-boosting features** to manage relationships and collaborate effectively. With **secure cloud storage**, it allows you to run your business from anywhere.

Tools and Technologies Used :

The core of this project revolved around the Salesforce platform, utilizing various clouds and features:

- **Salesforce Sales Cloud:** Used for managing customer accounts, contacts, and sales opportunities related to bulk fuel purchases or B2B services.
- **Salesforce Service Cloud:** Implemented for managing customer service cases, inquiries, and support requests. This included setting up case management workflows, escalation rules, and knowledge bases.
- **Apex and Visualforce:** Used for custom development of specific functionalities and user interfaces not available out-of-the-box.
- **Lightning Web Components (LWC):** Employed for building modern, responsive, and reusable user interface components.
- **Salesforce Flow:** Leveraged for automating complex business processes and workflows without extensive coding.
- **Salesforce Reports and Dashboards:** Used for creating real-time operational reports and analytical dashboards.
- **Integration Tools:** Various APIs and integration platforms were used to connect Salesforce with existing Point-of-Sale (POS) systems and inventory management solutions.

PROJECT IMPLEMENTATION STEPS :

The project followed a structured implementation methodology, typically involving the following phases:

1. Creating a Salesforce Developer Account

- Go to Salesforce Developer Signup and fill the form (Name, Email, Role: Developer, Company, Country, Postal Code, Username).
- Click Sign Me Up.
- Check your email and click Verify Account.
- Set a password and answer the security question.
- You're redirected to your Salesforce Setup page.

2. Creating Custom Objects

Salesforce objects are database tables for storing organization-specific data:

- **Standard Objects:** Predefined by Salesforce (e.g., Users, Reports)
- **Custom Objects:** Created by users for unique organizational needs

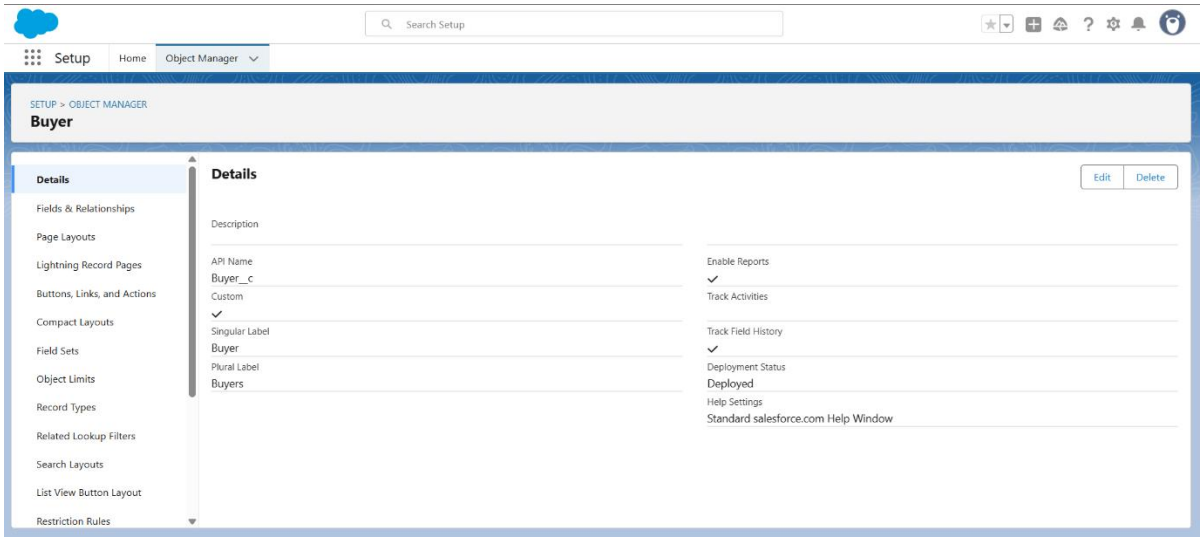
Setup & Navigation:
Access **Setup → Object Manager → Create → Custom Object**

Custom Objects Created:

- **Supplier__c:** Tracks suppliers and total fuel supplied
- **Gas_Station__c:** Maintains gas station info, fuel availability, and usage
- **Buyer__c:** Stores customer details and manages fuel purchases
- **Fuel_details__c:** Records fuel transactions linking buyers, suppliers, and gas stations

Record & Format Configuration:

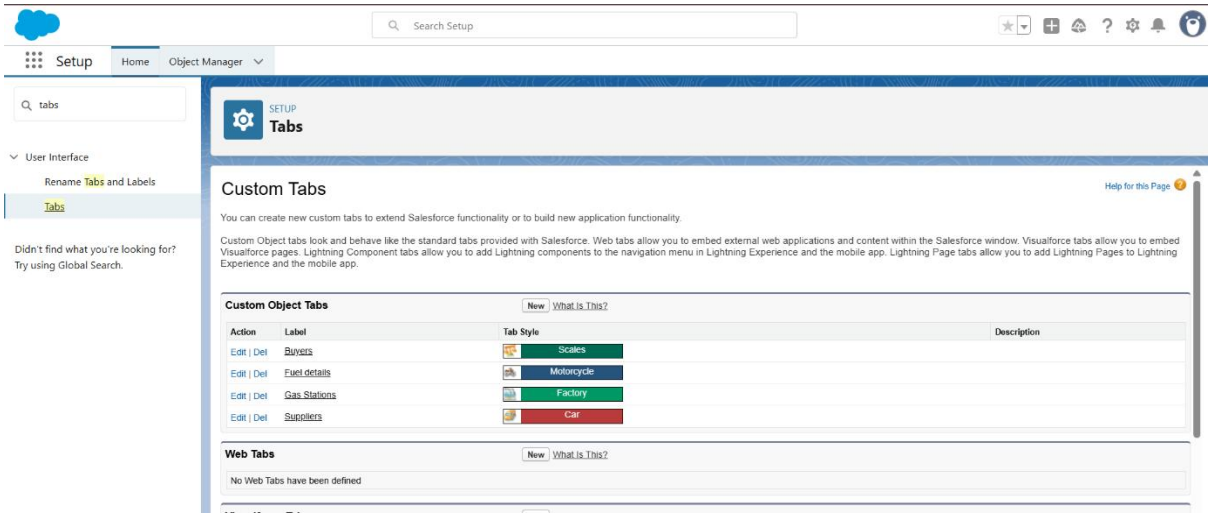
- Supplier: Name, Reports & Search enabled
- Gas Station: Auto Number, Format Gas-{000}, Reports & Search enabled
- Buyer: Auto Number, Format Buyer-{000}, Reports & Search enabled
- Fuel Details: Auto Number, Format Fuel-{000}, Reports & Search enabled



3. Creating a Custom Tabs (Supplier, Gas Station, Buyer, and Fuel Details)

A tab is a user interface element used to **build and view records** for objects in Salesforce. Creating a Custom Tab (Supplier):

- Go to Setup → Quick Find → Tabs → New (Custom Object Tab)
- Select Object (Supplier) and tab style → Next
- Add to profiles (default) → Next
- Add to Custom App (uncheck “Include Tab”)
- Ensure “Append tab to users’ existing personal customizations” is checked → Save



4. Creating a Lightning App: GAS STATION

A Lightning App in Salesforce is a collection of objects, tabs, and utilities that work together to serve a specific purpose. It allows users to:

- Access multiple objects and tabs in one convenient navigation bar
- Brand the app with a custom color and logo
- Include a utility bar and Lightning page tabs
- Switch between apps efficiently, improving productivity

Steps to Create the GAS STATION Lightning App

Step 1: Open App Manager

Go to Setup → App Manager → New Lightning App

Step 2: Enter App Details

App Name: GAS STATION

Click Next, keep default options for App Options and Utility Items

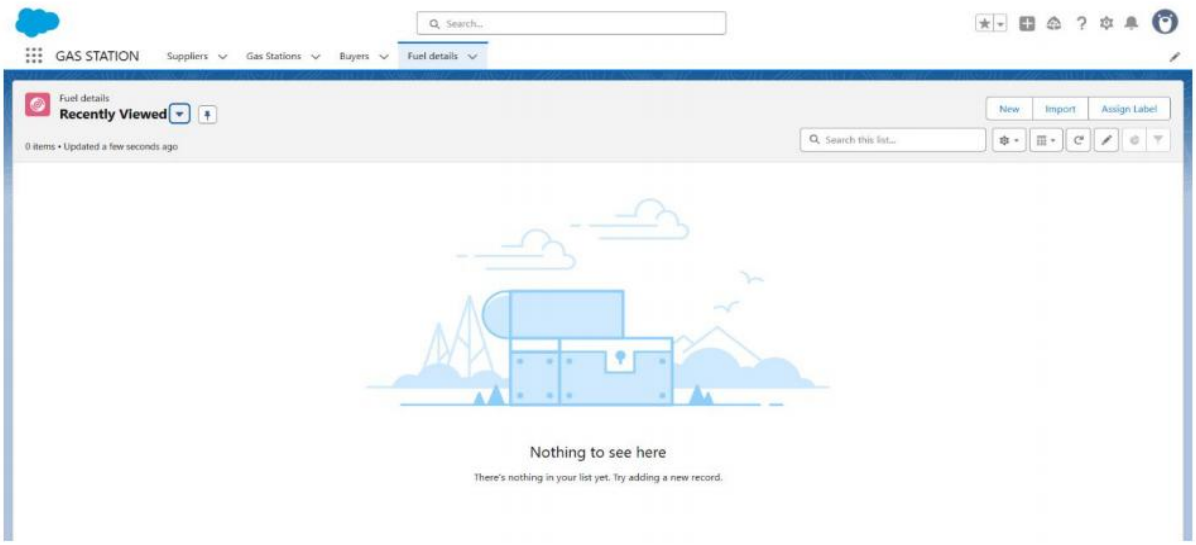
Step 3: Add Navigation Items

Select Supplier, Gas Station, Buyer, Fuel Details

Move them to selected items using the arrow → Click Next

Step 4: Assign User Profiles

Select System Administrator profile → Move to selected → Click Save & Finish

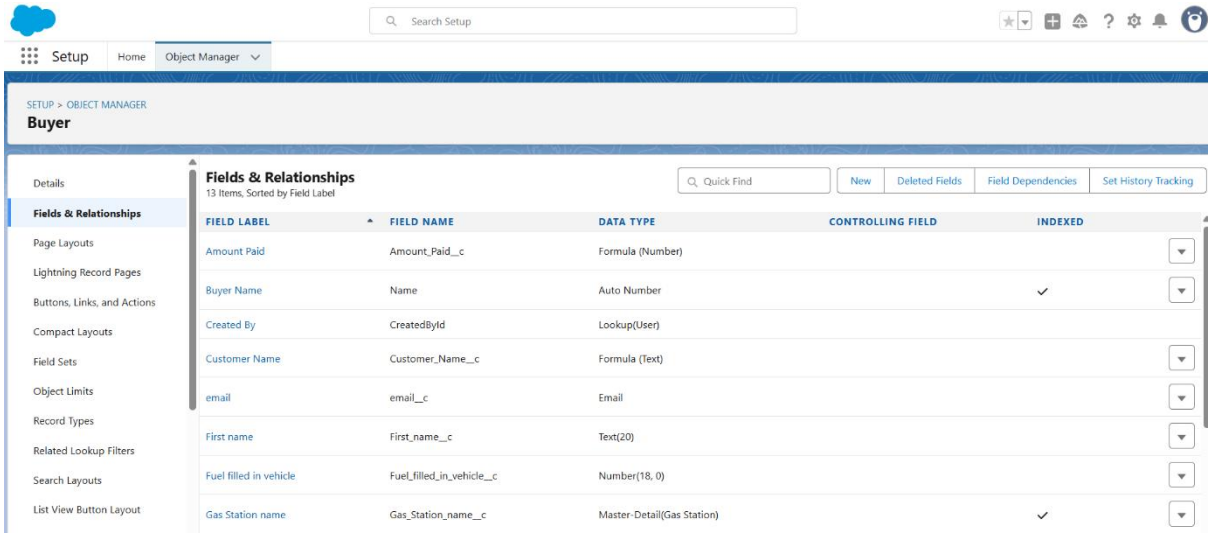


5. Field Creation

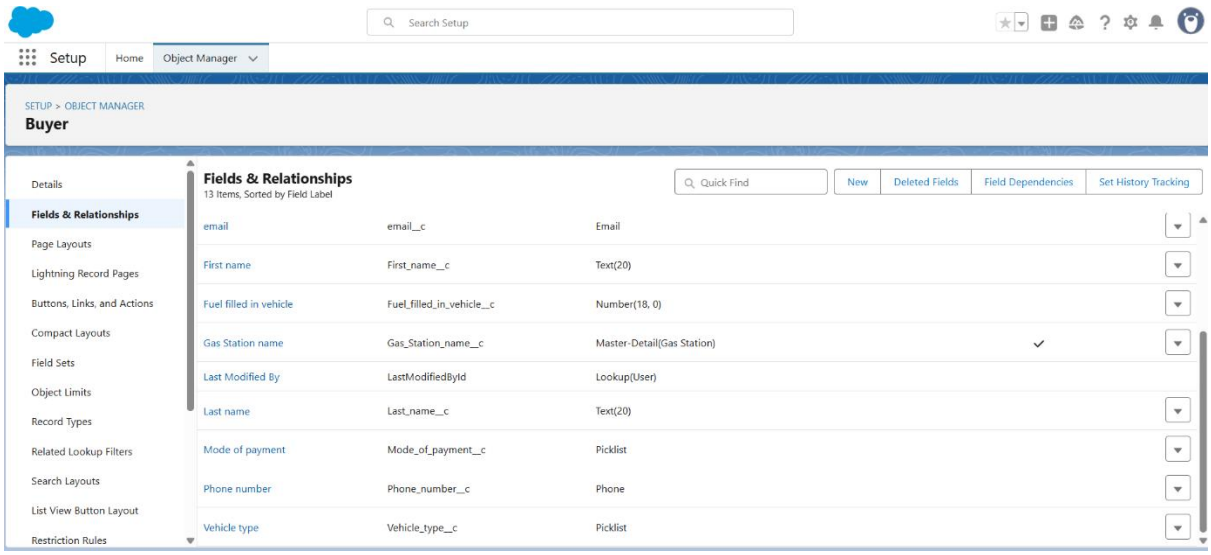
Fields in Salesforce represent data stored in object columns, making it easier to search, edit, and manage records.

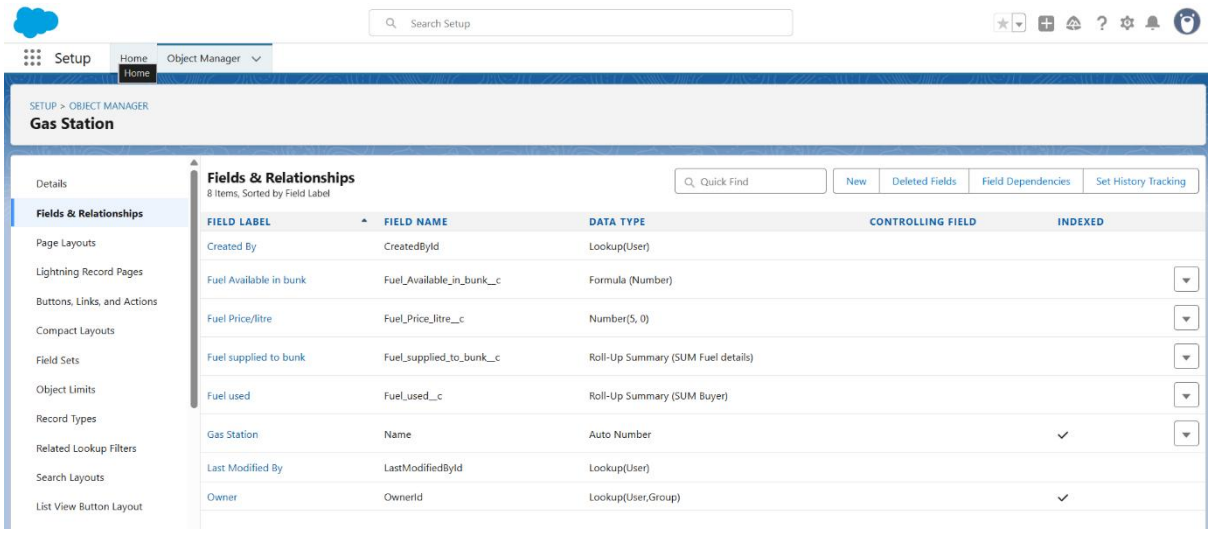
Steps to Create Fields

1. Navigate to Object: Setup → Object Manager → Select Object → Fields & Relationships → New
2. Choose Field Type: Number, Text, Formula, Master-Detail, Picklist, Rollup Summary.
3. Configure Field: Enter Field Label, Field Name, set length/format, pick related objects if required.
4. Set Additional Options: Enable Reports, Search, Track Field History as needed.
5. Save Field: Click Next → Next → Save



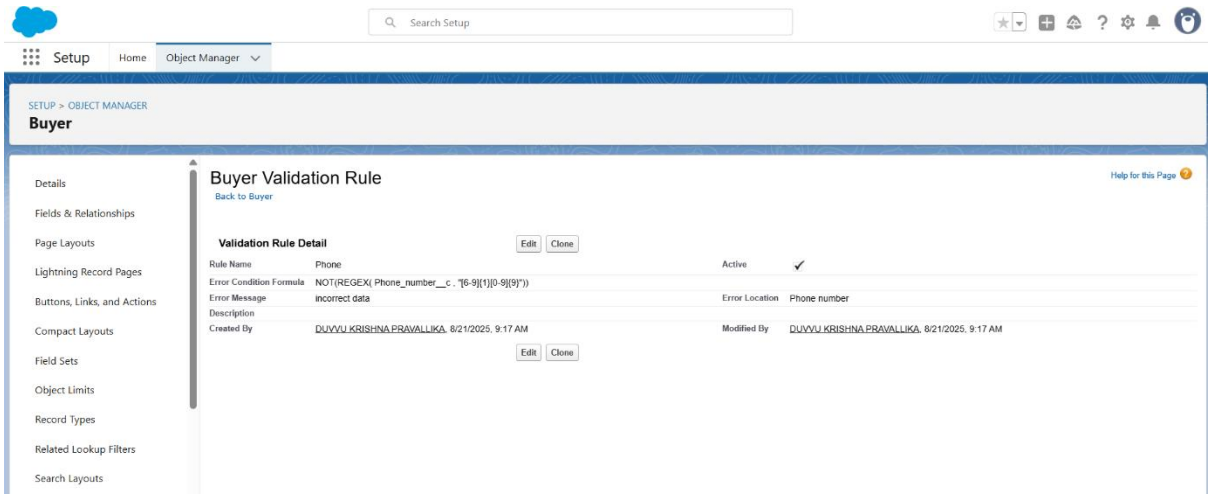
- Buyer__c: First Name, Last Name, Customer Name (Formula), Phone Number, Email, Vehicle Type (Picklist), Fuel Filled in Vehicle, Mode of Payment (Picklist), Amount Paid (Formula)
- Fuel_details__c: Fuel Supplied (Number), Supplier Name (Master-Detail), Gas Station (Master-Detail)
- Gas_Station__c: Fuel Price per Liter (Number), Fuel Supplied to Bunk (Roll-up Summary), Fuel Used (Roll-up Summary), Fuel Available in Bunk (Formula)
- Supplier__c: Sum of Fuel Supplied (Roll-up Summary)





Validation Rule in Buyer Object

A validation rule was created to ensure that the **Phone Number** field in the Buyer object accepts only valid 10-digit numbers starting with digits **6–9**.

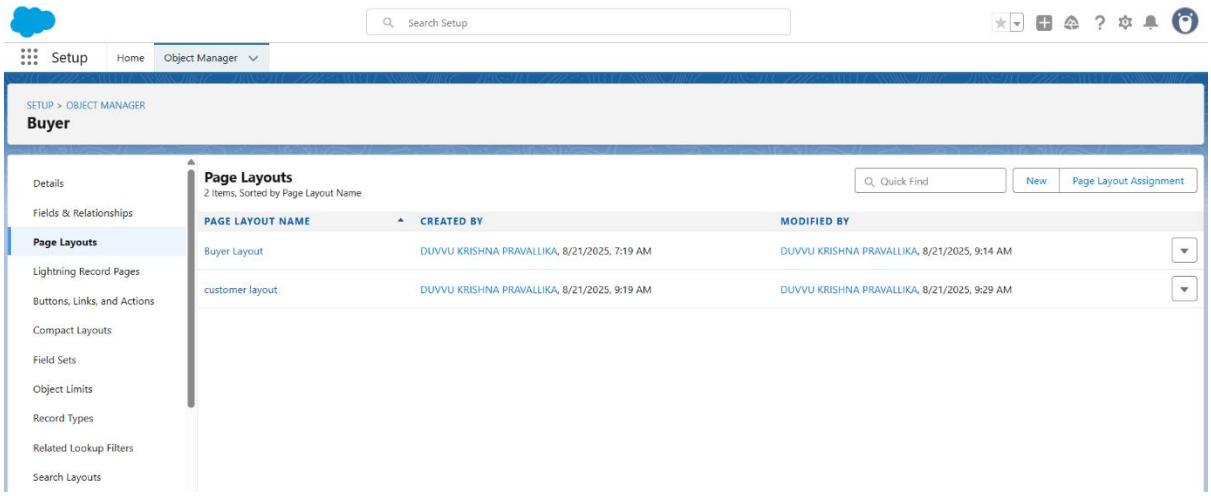


6. Page Layouts

A Page Layout in Salesforce defines how records are displayed and organized on detail and edit pages. It allows us to control the visibility, order, and grouping of fields, related lists, and custom links for both standard and custom objects.

Creating a Page Layout (Buyer Object)

1. Go to **Setup → Object Manager → Buyer → Page Layouts**.
2. Click on **New**, select the existing layout, name it **Customer Layout**, and click **Save**.
3. Drag and drop a **Section** field into the layout and create three sections:
   - **Personal Details**: Add fields → *First Name, Last Name, Customer Name, Phone Number, Email, Gas Station Name*.
   - **Vehicle Info**: Add fields → *Fuel Filled in Vehicle, Vehicle Type*.
   - **Receipt Details**: Add fields → *Mode of Payment, Amount Paid*.
4. Click **Save** to apply the changes.



7.Profiles in Project

Purpose: Profiles control user access to objects, fields, and apps in Salesforce. Different roles (Manager, Sales Executive, Sales Person) were given custom profiles for controlled access.

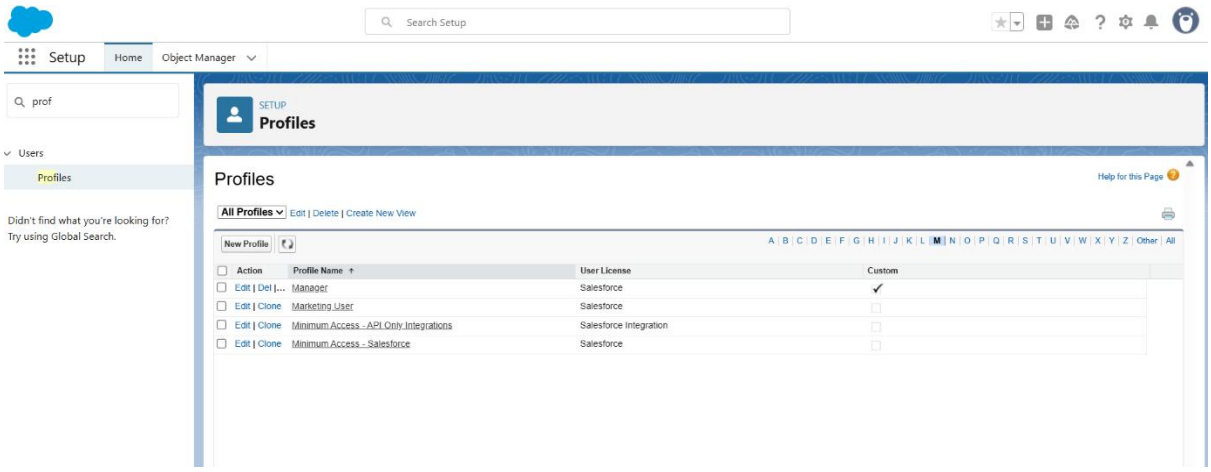
Types of Profiles:

- **Standard Profiles**: Predefined (e.g., System Administrator, Standard User, Read Only). Cannot be deleted.
- **Custom Profiles**: Created by users. Can be deleted if no user is assigned.

Steps to Create a Profile:

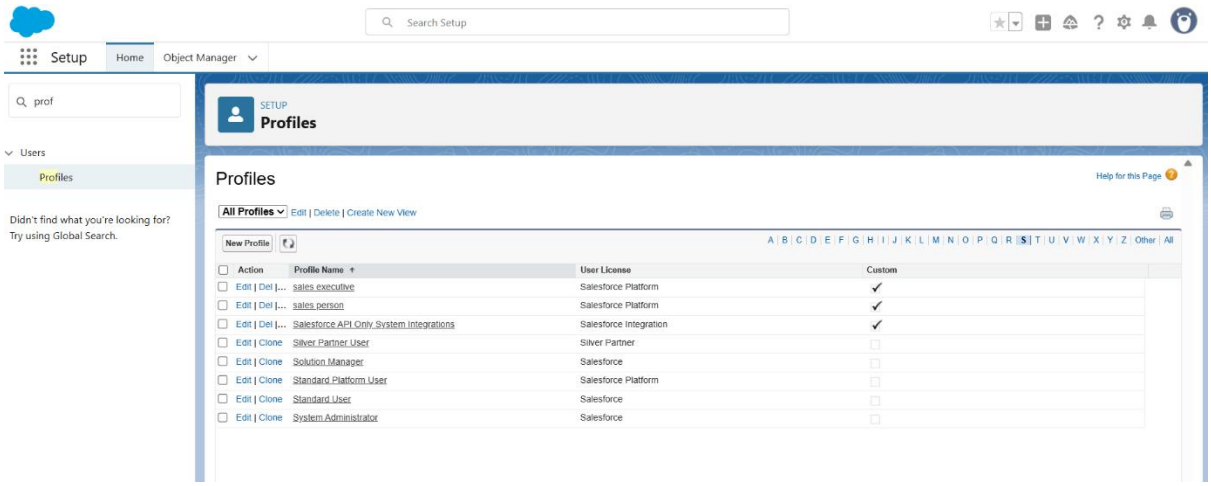
1. Go to **Setup → Profiles**.

2. **Clone** an existing profile (e.g., Standard User / Salesforce Platform User).
3. Enter **Profile Name** and **Save**.
4. Click **Edit** → set **Custom App = Gas Station**.
5. Give **object permissions** (Buyers, Fuel Details, Gas Station, Supplier).
6. Adjust session timeout & password policies if needed.



Profiles Created:

- **Manager Profile** – full access to all objects, session timeout 8 hrs, password never expires.
- **Sales Executive Profile** – object permissions given, default app set to Gas Station.
- **Sales Person Profile** – access to Buyers, Fuel Details, Gas Station & Supplier.

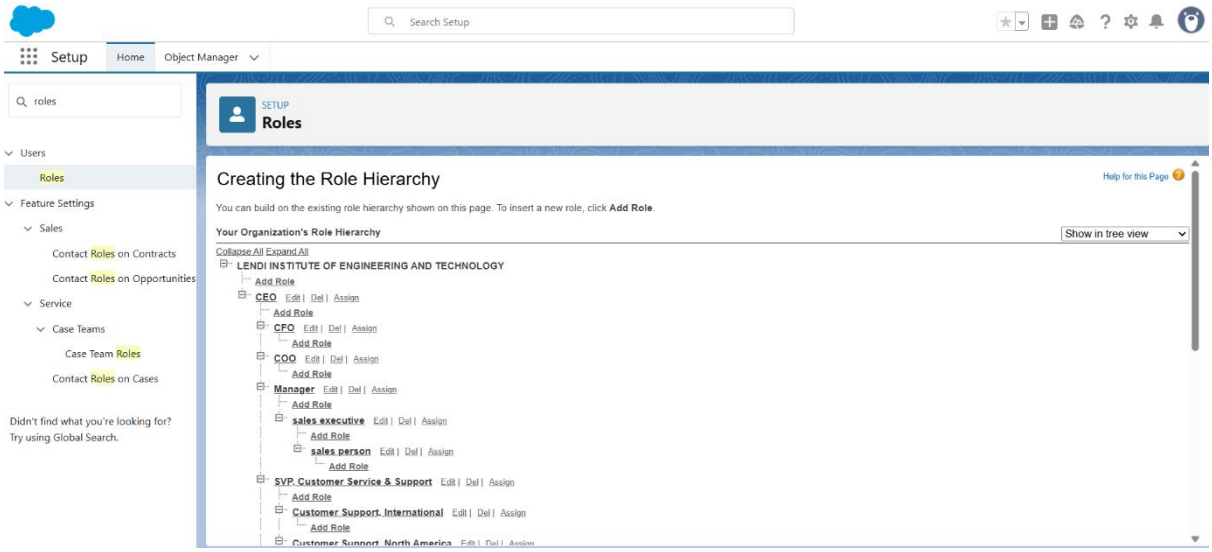


8. Role & Role Hierarchy

A role in Salesforce defines record-level visibility for users. Roles ensure that data access is aligned with responsibilities. A role hierarchy allows managers to see and access records owned by their subordinates.

Steps to Create Roles:

1. Go to Setup → Quick Find → Roles → Set Up Roles.
2. Click Expand All.
3. Choose where the role should be placed and click Add Role.
4. Enter the Label (role name auto-populates) and click Save.



Roles Created in Project:

- Manager Role – created directly under CEO.
- Sales Executive Role – created under Manager.
- Sales Person Role – created under Sales Executive.

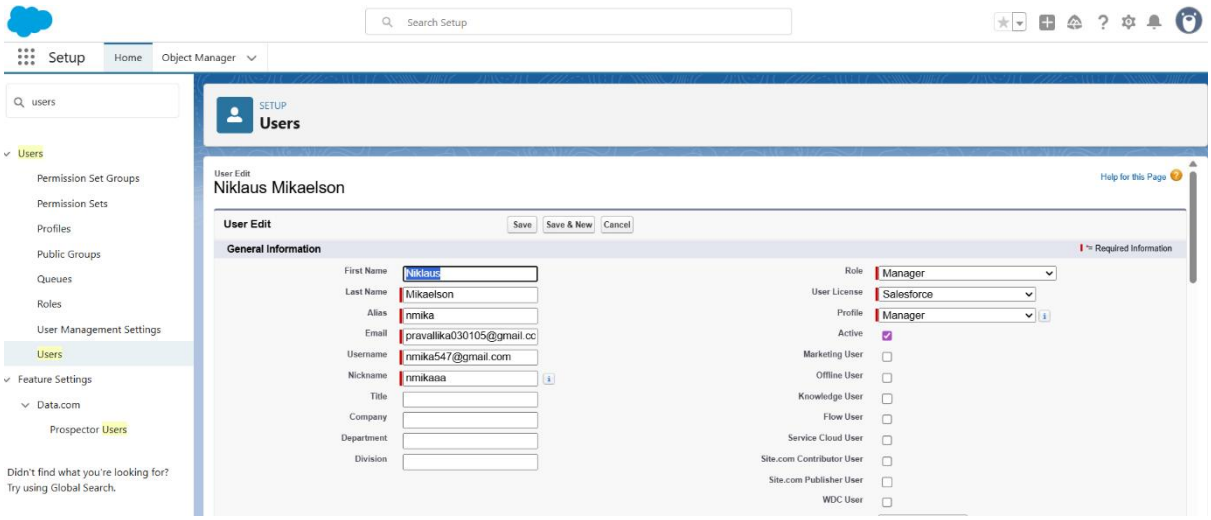
9. Creating Users

A **user** is anyone who logs in to Salesforce and needs access to company records. Each user has a **role, profile, and license** that define their permissions.

Steps to Create Users:

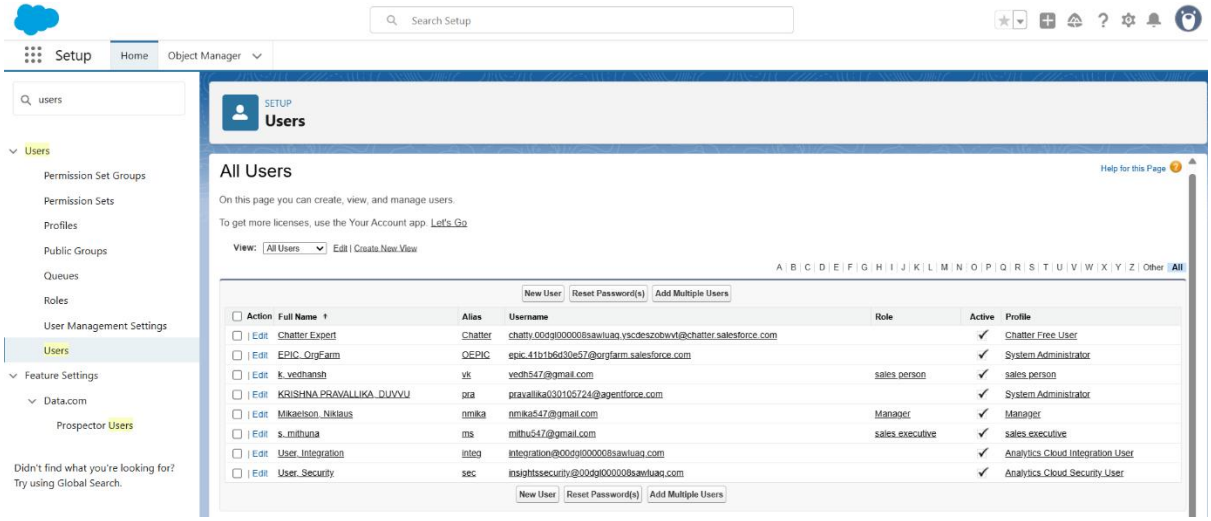
1. Setup → Quick Find → **Users → New User**
2. Fill in required details: Name, Alias, Email, Username, Nickname.

3. Assign **Role, User License, and Profile**.
4. Click **Save**.



Users Created:

- **Niklaus Mikaelson** - Role: Manager, Profile: Manager, License: Salesforce
- **Sales Executive User** - Role: Sales Executive, Profile: Sales Executive, License: Salesforce Platform
- **Sales Person User** - Role: Sales Person, Profile: Sales Person, License: Salesforce Platform

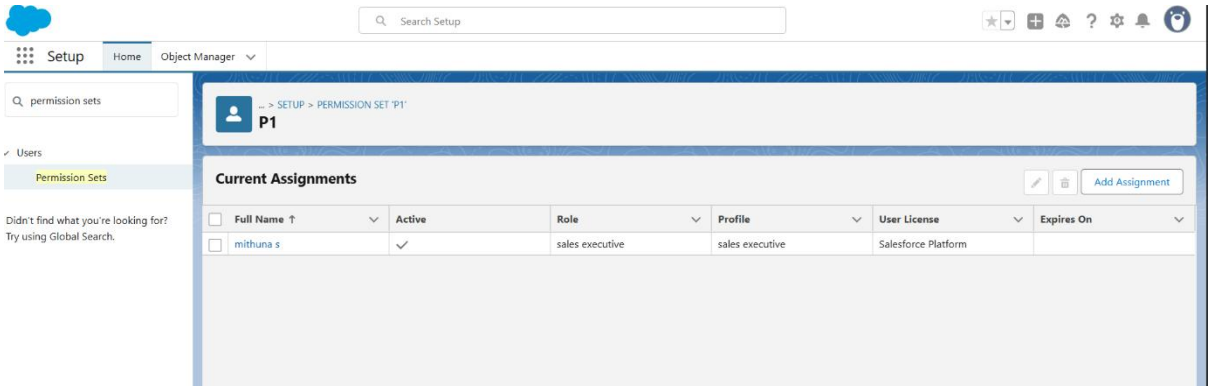


10. Create Permission Sets:

A permission set is a collection of settings and permissions that extends a user's functional access without changing their profile. Users can have multiple permission sets but only one profile.

Steps to Create Permission Set:

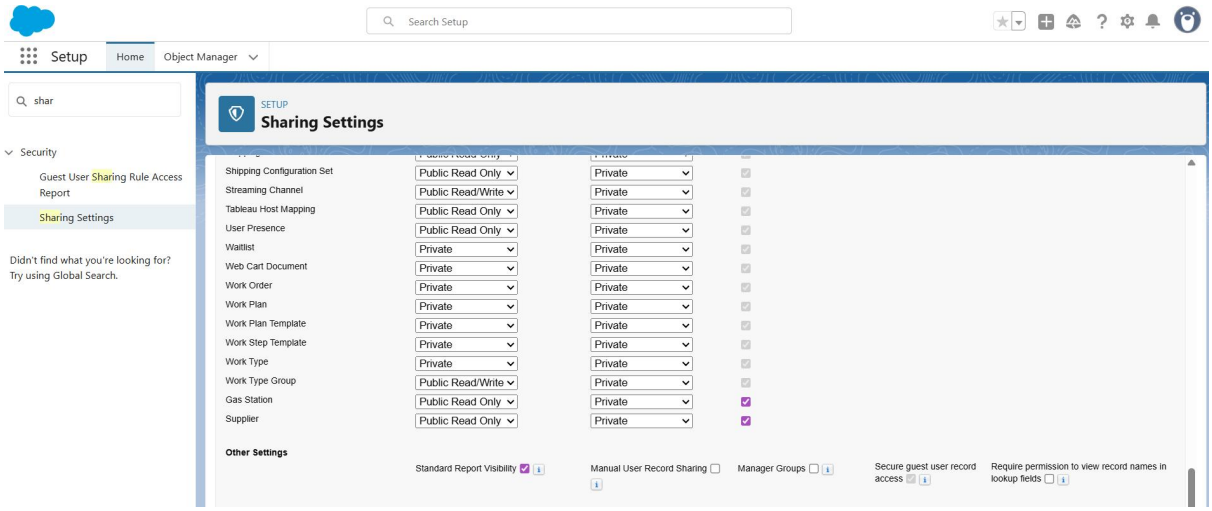
1. Setup → Quick Find → Permission Sets → New
2. Enter Label = P1 (API auto-populated) → Save
3. Under Apps → Object Settings, select Fuel Details → Edit → enable Read & Create → Save
4. Click Manage Assignments → Add Assignment → select users (e.g., Sales Executive) → Next → Assign → Done



- Permission Set: P1
- Assigned To: Users with Sales Executive Profile
- Access Granted: Fuel Details Object → Read & Create permissions

11. Organization-Wide Defaults (OWD) Setup

OWD defines the baseline record-level access in Salesforce, controlling who can view or edit records. In this project, Gas Station and Supplier objects were set to Public Read-Only, while roles and profiles controlled additional access.

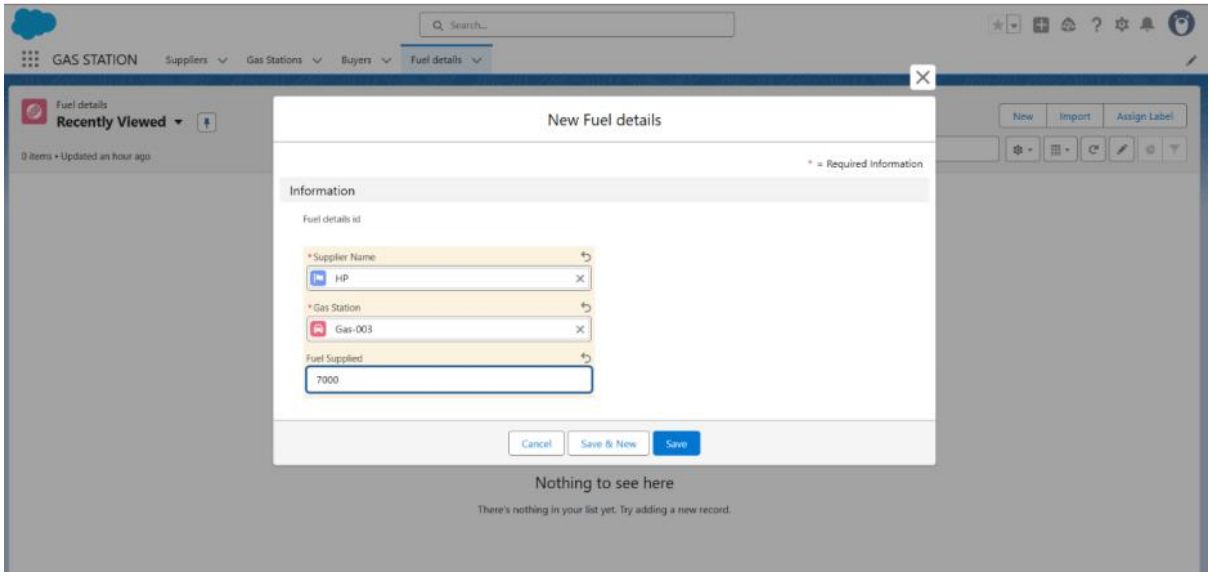


12. User Adoption: Managing Records

User adoption measures how effectively users engage with Salesforce features.

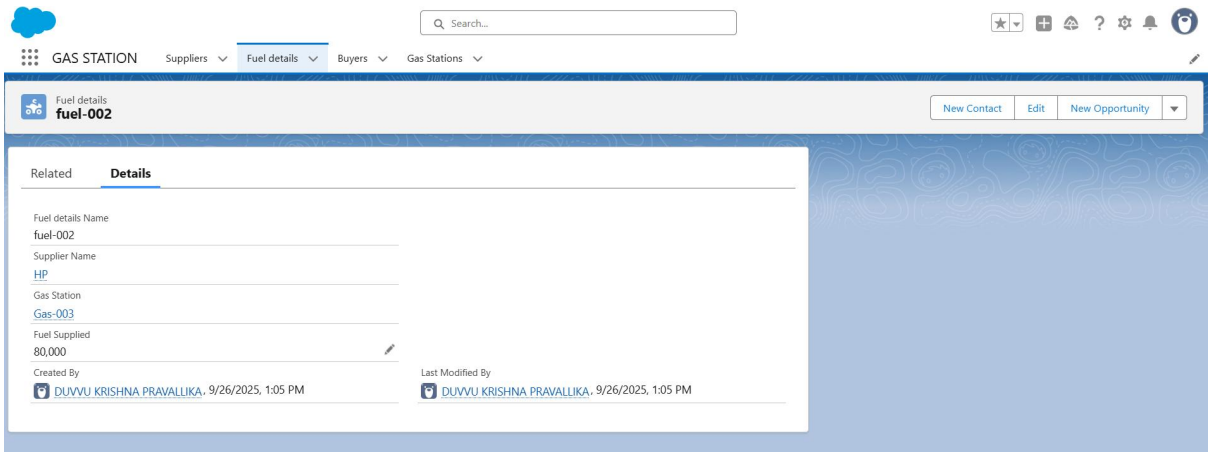
Creating Records:

1. Open App Launcher → Gas Station → Fuel Details tab → New
2. Fill in Fuel Details, then create related Supplier and Gas Station records directly from the junction object.
3. Complete remaining fields in Fuel Details → Save
4. Repeat to create 10 Buyer records



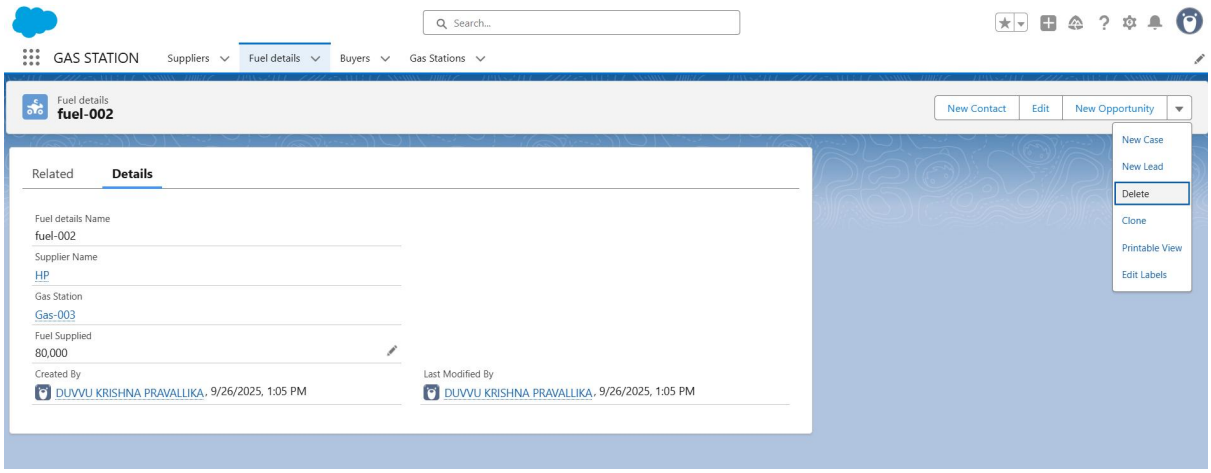
Viewing Records:

1. Open App Launcher → Gas Station → Fuel Details tab
2. Click on any created record to view details



Deleting Records:

1. Open App Launcher → Gas Station → Fuel Details tab
2. Click the arrow on the record → Delete → Confirm

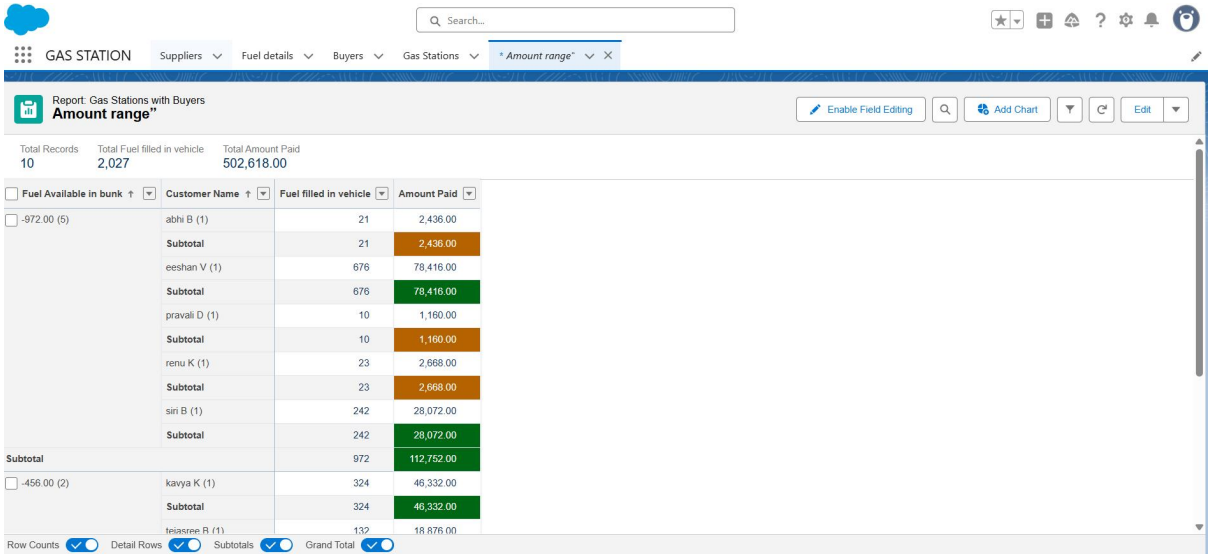


13. Creating Reports

Reports are used to analyse and display Salesforce data in a structured format. In this project, custom reports were created to track fuel usage, customer activity, and sales for better decision-making.

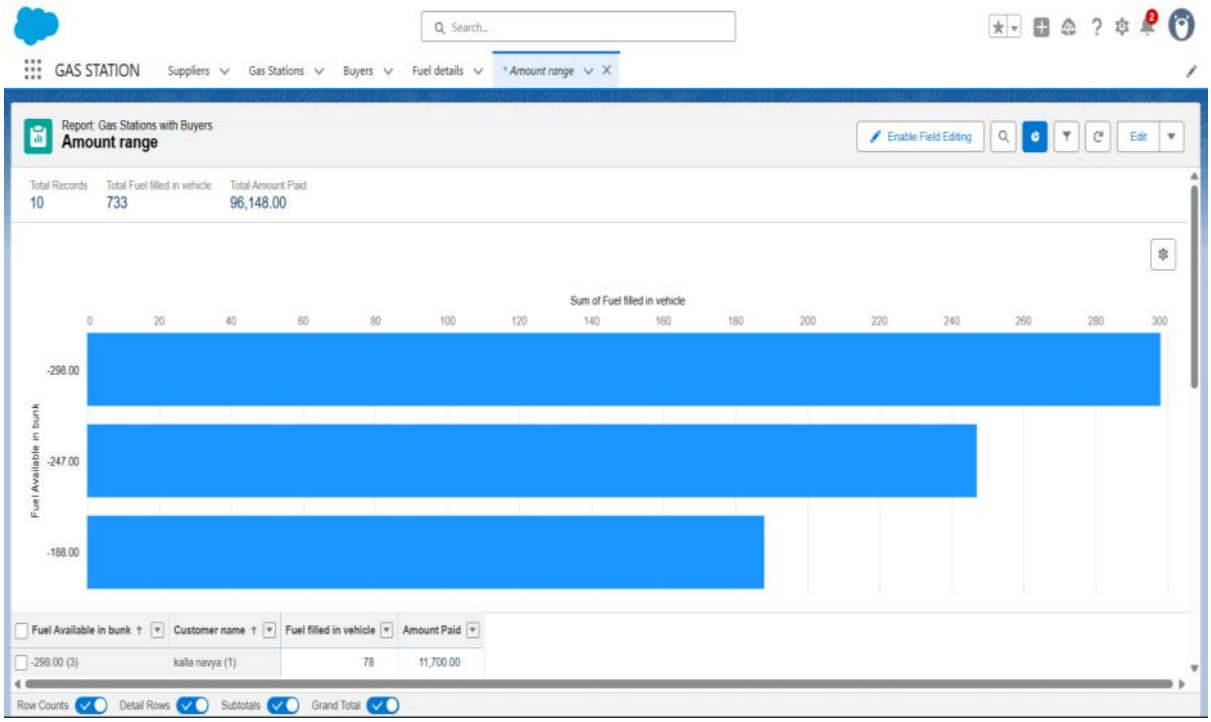
Create Report Folder: App Launcher → Reports → New Folder → **Fuel Estimation** → Save

Share Report Folder: Reports → All Folders → Fuel Estimation → Share → Role: **Manager** → Access: View → Done



Create Report:

- Ensure **10 Buyer records** are created with complete data
- Reports tab → **New Report** → Select type: **Gas Station with Buyers** → Start Report
- Columns: Fuel Filled in Vehicle, Amount Paid
- Group Rows: Fuel Available in Bunk, Customer Name
- **Conditional Formatting:** Sum of Amount Paid, range 1000–5000 → Apply
- Save Report: Name → **Amount Range**, Folder → Fuel Estimation → Save & Run



14. Creating Dashboards

Dashboards provide a **visual summary of Salesforce data**, helping users identify trends, track metrics, and make decisions based on real-time reports.

Steps Taken:

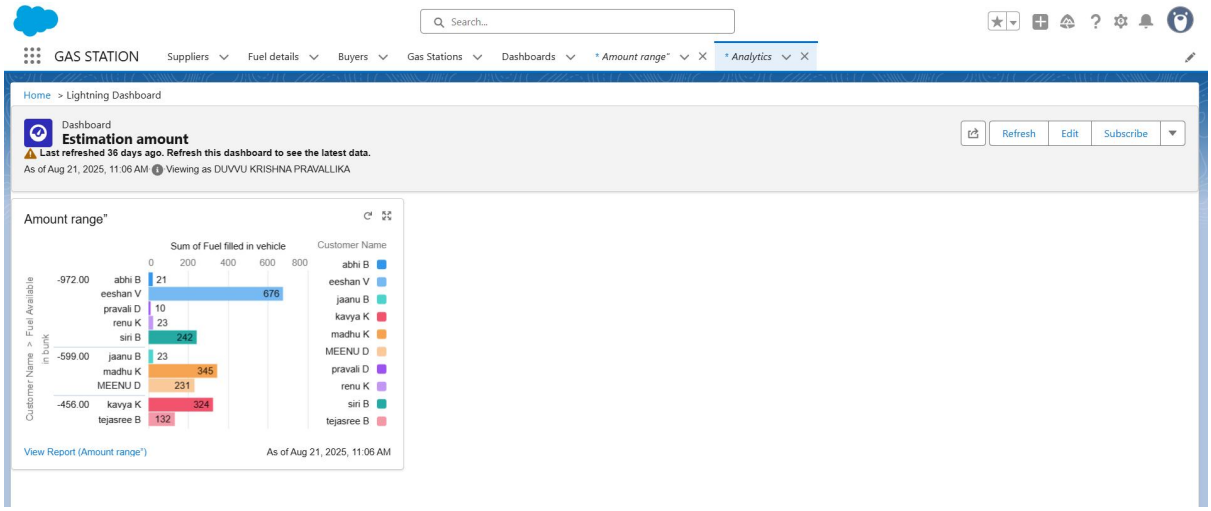
1. Create Dashboard Folder:

- App Launcher → Dashboards → New Folder → **Amount Estimation Dashboard** → Save
- Apply **sharing settings** as done for reports (e.g., Manager role with view access)

2. Create Dashboard:

- Dashboards tab → New → Enter **Name** (Estimation amount)→ Select Folder → Create

- Add Component → Select Report → Add → Save → Done

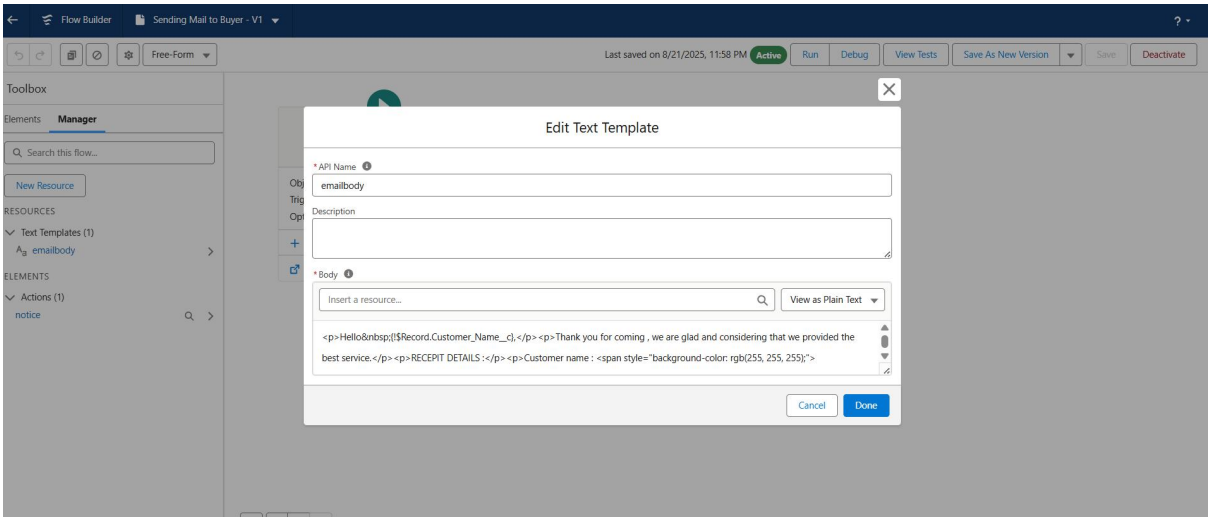


15. Creating Flow

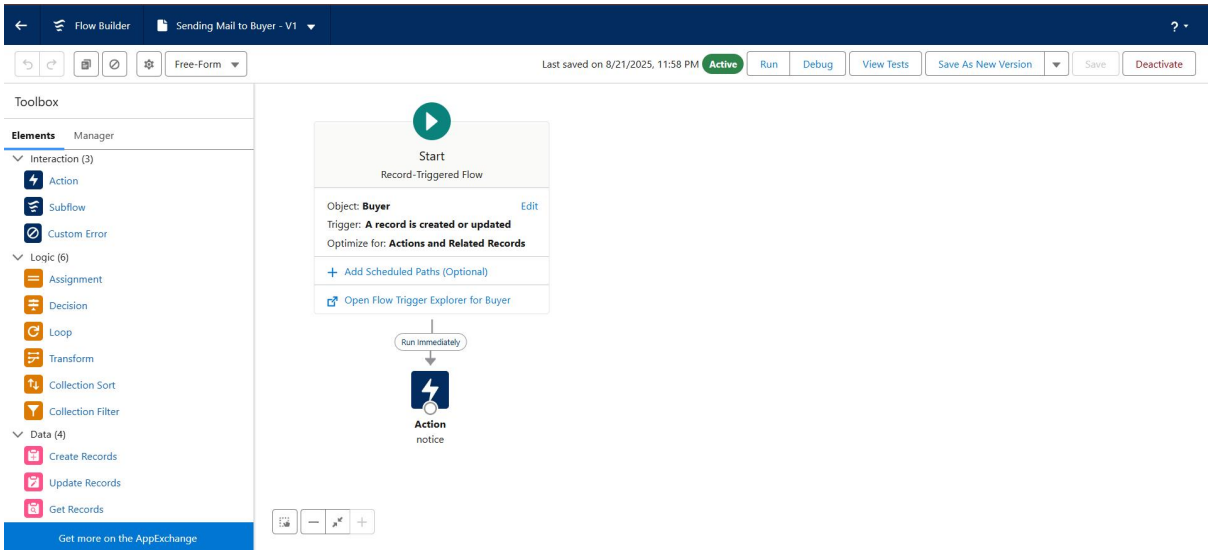
Flows in Salesforce **automate business processes**, guide users, and update data without coding. In this project, a flow was created to **send automated emails to buyers** after a record is created or updated.

Steps Taken:

1. **Create Flow:** Setup → Flows → New Flow → Record-Triggered Flow → Object: Buyer → Trigger: Created or Updated → Optimize: Actions and Related Records → Done
2. **Create Text Template:** Resource → Text Template → API Name: emailbody → Body includes customer name, amount paid, vehicle type, fuel intake → Done



3. **Add Action:** Drag Action → Search: Send Email → Label: Notice → Set Body: emailbody → Recipient: Buyer Email → Subject: “Welcome to Gas Station” → Done
4. **Connect Flow:** Drag path from Start → Action → Save → Activate



A record-triggered flow on the Buyer object sends an automatic email receipt whenever a record is created or updated.

16. Creating Apex Triggers

Apex triggers allow custom actions before or after changes to Salesforce records, such as insert, update, or delete. In this project:

Use Case:

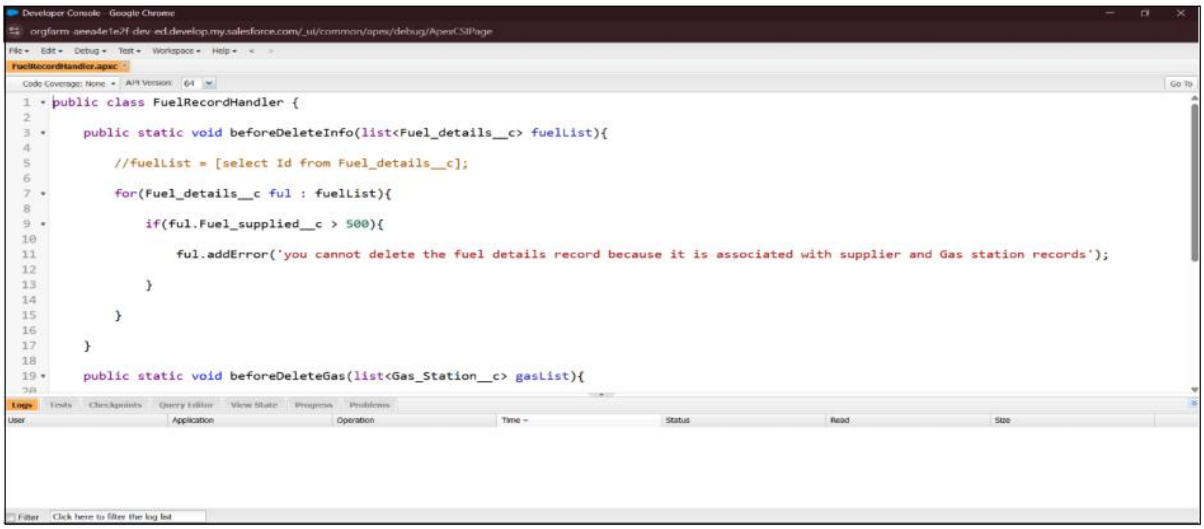
- **Fuel Details Object:** Prevent deletion of records if Fuel_supplied__c > 500.
- **Gas Station Object:** Ensure Fuel_price_liter__c > 50 before inserting a record.

Steps to Create Apex Trigger:

1. Log in → Gear icon → Developer Console → File → New → Apex Class → Name: FuelRecordHandler.
2. Write methods beforeDeleteInfo for Fuel Details and beforeDeleteGas for Gas Station.
3. Save the class.
4. Create a Trigger: File → New → Trigger → Assign object (Fuel_details__c or Gas_Station__c) → Specify event (before Delete/Insert).
5. Call respective handler methods inside the trigger.

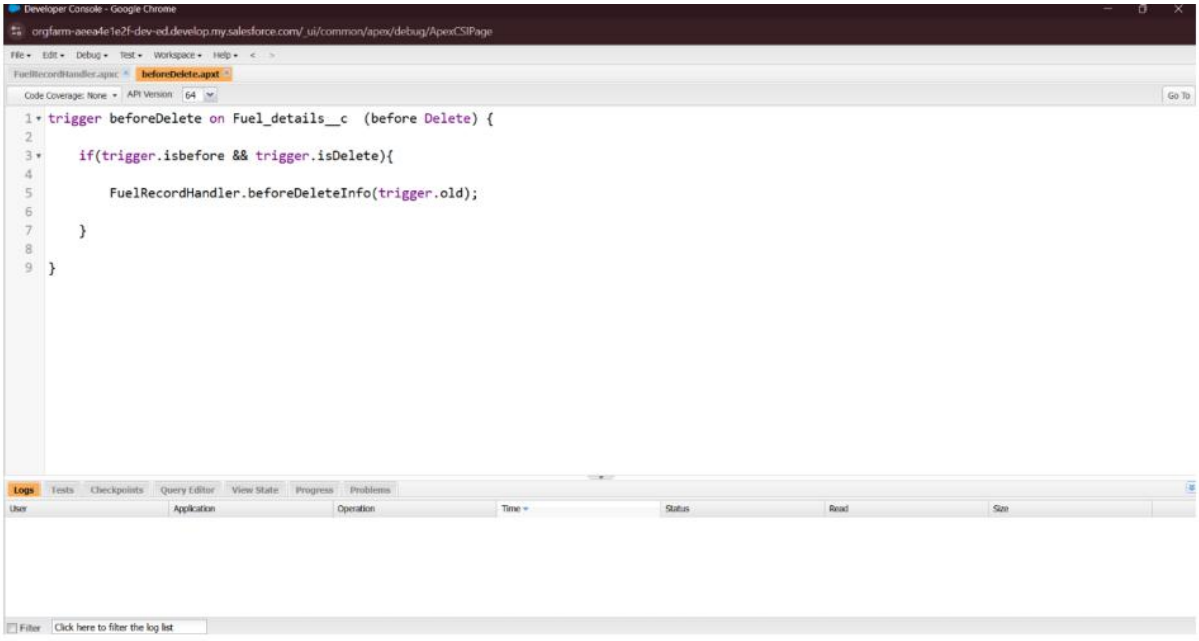
Trigger: Fuel Details (before delete):

- Executes **before a new record is saved** to the database.
- Used to **validate or modify field values** before insertion.
- In **Gas Station Object**, beforeInsert checks that Fuel_price_liter__c > 50 before allowing the record to be saved.



Trigger: Gas Station (before insert):

- Executes **before a record is deleted** from the database.
- Used to **prevent deletion** if certain business conditions are not met.
- In **Fuel Details Object**, beforeDelete prevents deletion if Fuel_supplied__c > 500.



Additional Screenshots:

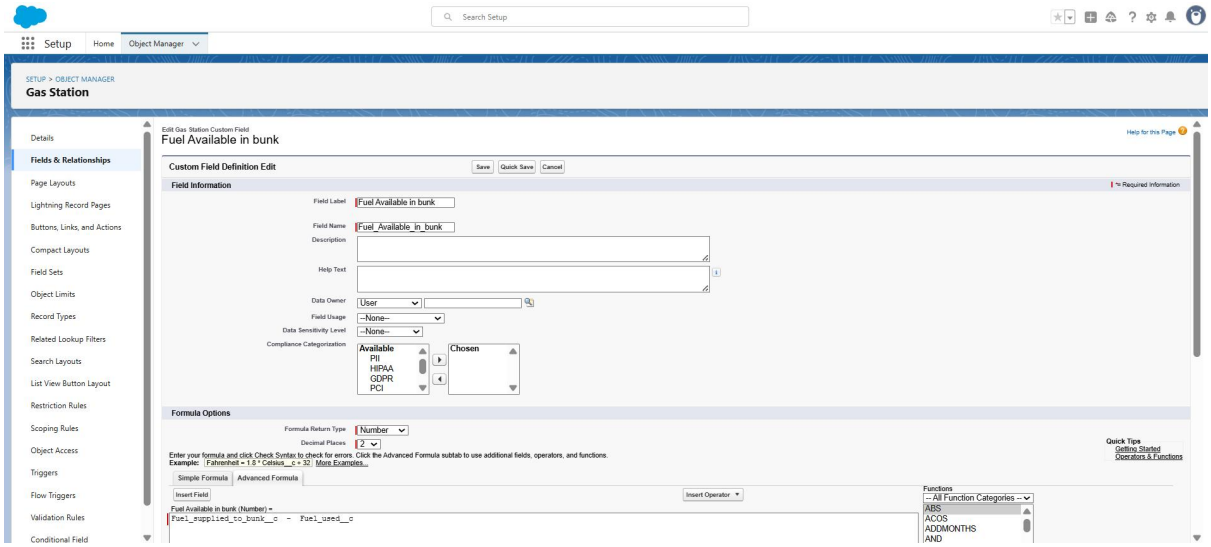


Fig: formula field of amount paid

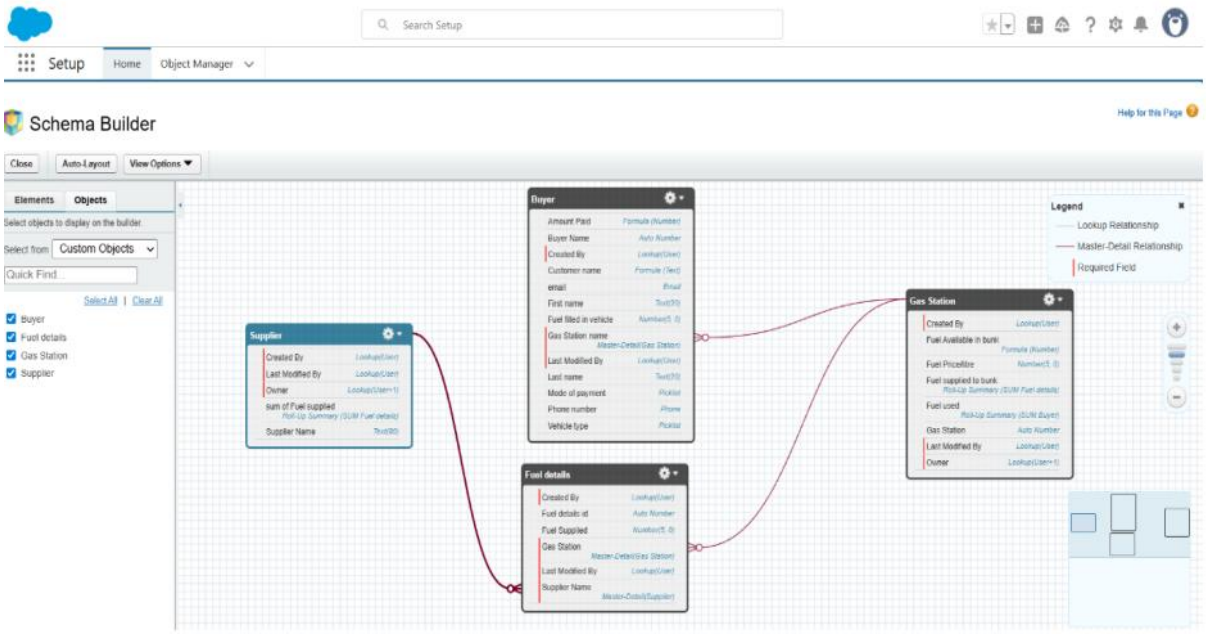


Fig: custom objects relationship

CONCLUSION

The “Application to Make Gas Filling Station Easy Using CRM” project successfully utilized Salesforce to streamline and optimize gas station operations. Custom objects, fields, page layouts, profiles, and permission sets ensured organized data management and secure access for roles like Manager, Sales Executive, and Sales Person. Automated flows and Apex triggers improved service efficiency by sending customer receipts and maintaining data accuracy during record changes. Reports and dashboards provided actionable insights for monitoring performance and decision-making. Overall, the project demonstrates how Salesforce can enhance operational efficiency, reduce manual effort, and deliver a superior customer experience in gas station management.