

# APPLICATION FOR SIMPLIFYING GAS STATION MANAGEMENT USING SALESFORCE CRM

## Abstract

Running a gas station involves handling customers, employees, and suppliers – all of which can be complicated without a proper system. This project introduces a customized Salesforce CRM application designed to make gas station operations more efficient. The solution integrates modules like **Customer Details, Fuel Records, Stations, and Supplier Data**, so everything is managed in a single platform.

With structured layouts that capture **personal details, vehicle information, and billing receipts**, the system improves accuracy and transparency. It also defines **user roles and permissions** to ensure smooth coordination between managers, sales executives, and sales staff.

Key features such as **secure login policies, access control, and session management** make the system more reliable. Overall, the project demonstrates how Salesforce CRM can be tailored to manage day-to-day processes at fuel stations, enhancing both employee productivity and customer satisfaction.

## Objectives

The main goals of the Gas Station CRM project are to simplify management, improve customer service, and ensure secure operations:

- **Efficient Operations:** Centralize records of customers, suppliers, fuels, and stations in one system.
- **Better Customer Experience:** Maintain accurate information for vehicles, receipts, and customer interactions.
- **Role-Based Access:** Define permissions for managers, executives, and

salespersons.

- **Automation:** Reduce manual effort by automating entries for buyers, fuel usage, and payments.
- **Security:** Safeguard sensitive data with policies and controlled access.
- **Scalability:** Provide room for future enhancements like dashboards and reports.

## Technology Overview

### Salesforce CRM

A cloud-based platform used for customer relationship management, offering tools for automation, security, reporting, and data handling.

### Core Components

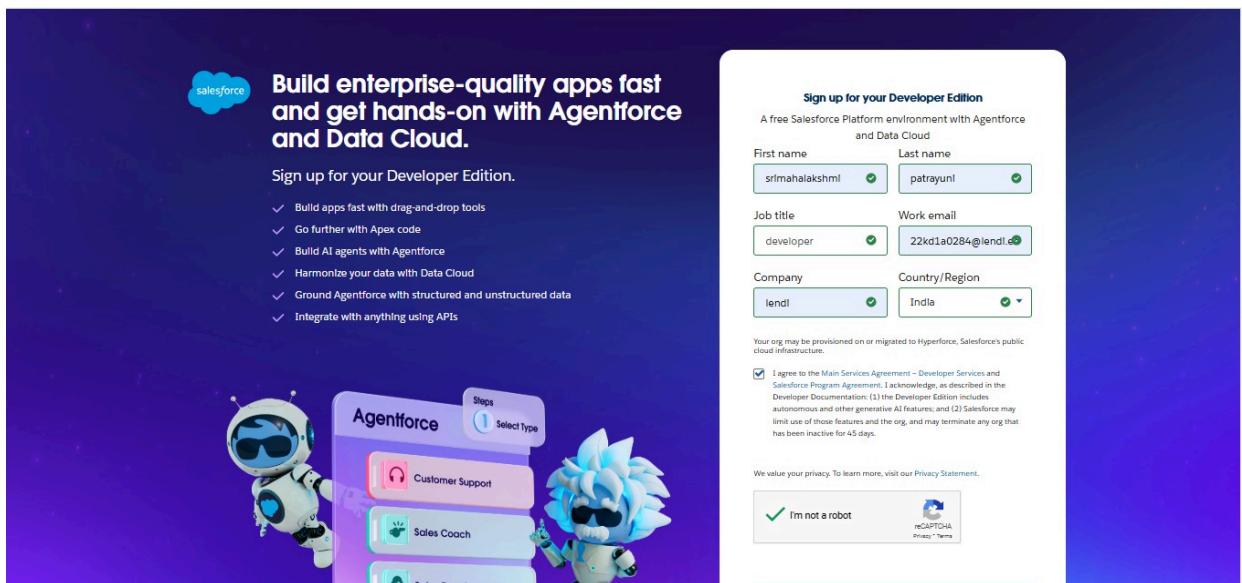
- **Custom Objects:** User-defined tables to store business-specific details.
- **Tabs:** Quick navigation elements to access objects and apps.
- **Lightning App:** A branded interface bundling objects, utilities, and tools for smooth workflows.
- **Page Layouts:** Structured record pages for easy and clear data entry.
- **Profiles:** Define what users can see or edit within the CRM.
- **Roles & Hierarchy:** Manage visibility and authority levels across the organization.
- **Users:** Individual accounts with specific permissions.
- **Permission Sets:** Extra access rights beyond standard profiles.
- **OWD (Organization-Wide Defaults):** Baseline sharing settings for data security.
- **Reports & Dashboards:** Graphical summaries for tracking progress and analyzing trends.
- **Flows:** Automations for routine processes triggered by actions or changes.
- **Apex Triggers:** Custom code to handle advanced logic, validations, and

automated updates.

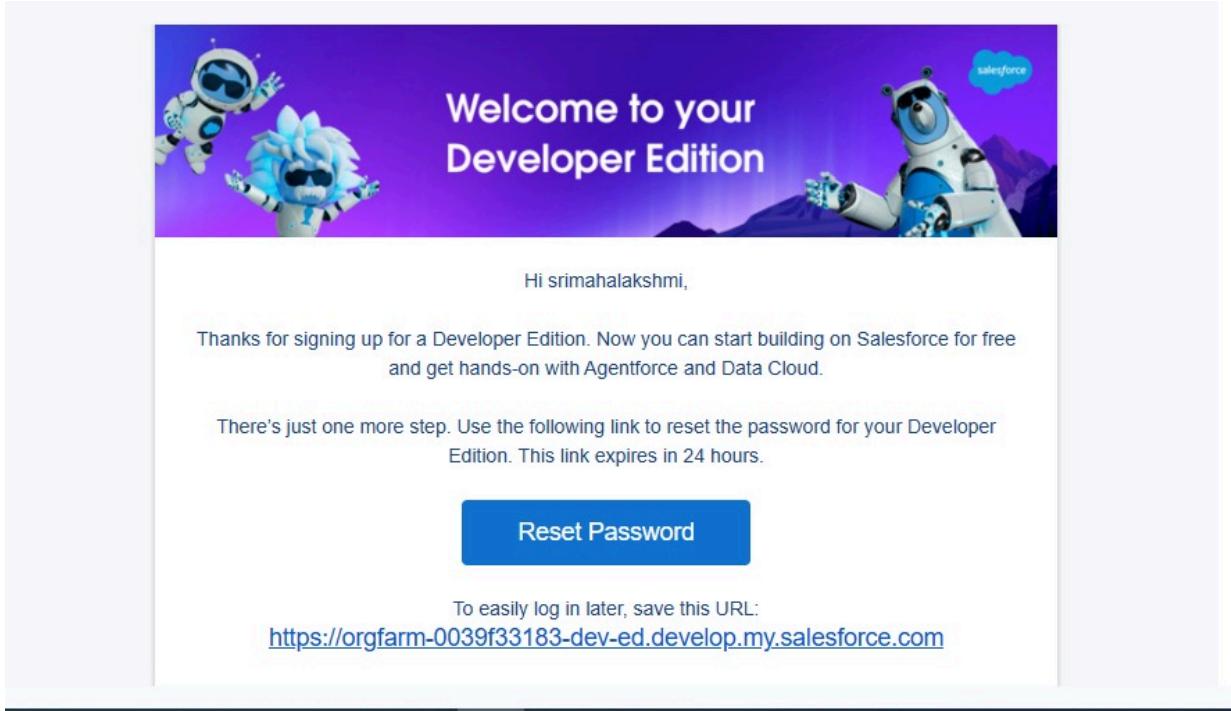
## Project Implementation Steps

### 1. Setting Up a Salesforce Developer Org:

- Sign up for a Developer Org with your details using  
<https://developer.salesforce.com/signup>
- Verify your email, set a password, and access the setup page.



Activation through the salesforce developer mail.



## 2. Custom Objects Creation:

Procedure for Object Creation:

Defines the method of setting up custom objects in Salesforce, including naming, data type selection, and enabling reporting and field tracking.

Four custom objects:

- Buyer\_\_c: Stores customer details and manages their fuel purchases.
- Fuel\_Details\_\_c: Records fuel transactions linking buyers, suppliers, and gas stations.
- Gas\_Station\_\_c: Maintains gas station information, fuel availability, and usage.
- Supplier\_\_c: Tracks suppliers and the total fuel supplied to stations.

A screenshot of the Salesforce Object Manager interface. The browser address bar shows the URL: orgfarm-0039f33183-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gL000001yqRN/Details/view. The page title is "SETUP &gt; OBJECT MANAGER" and the object name is "Buyer". On the left, a sidebar lists various object settings like Fields &amp; Relationships, Page Layouts, and Record Types. The main content area shows the "Buyer" object details. The "Details" section includes fields for API Name (Buyer\_\_c), Singular Label (Buyer), and Plural Label (Buyers). Other settings include checkboxes for Enable Reports, Track Activities, and Track Field History, all of which are checked. Deployment Status is set to Deployed. Help Settings point to the Standard salesforce.com Help Window.

### 3. Custom Tabs:

Represents a way to access Salesforce objects easily.

- Tabs are created for each custom object - Buyer, Fuel Details, Gas Station, and Supplies users to navigate and manage records efficiently within the Lightning App.

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected. The left sidebar has 'Tabs' highlighted under 'User Interface'. The main content area displays a table of custom object tabs:

Action	Label	Tab Style	Description
Edit   Del	Buyers	Alarm clock	
Edit   Del	Fuel details	Apple	
Edit   Del	Gas Stations	Balls	
Edit   Del	Receipts	Bank	
Edit   Del	Suppliers	Airplane	

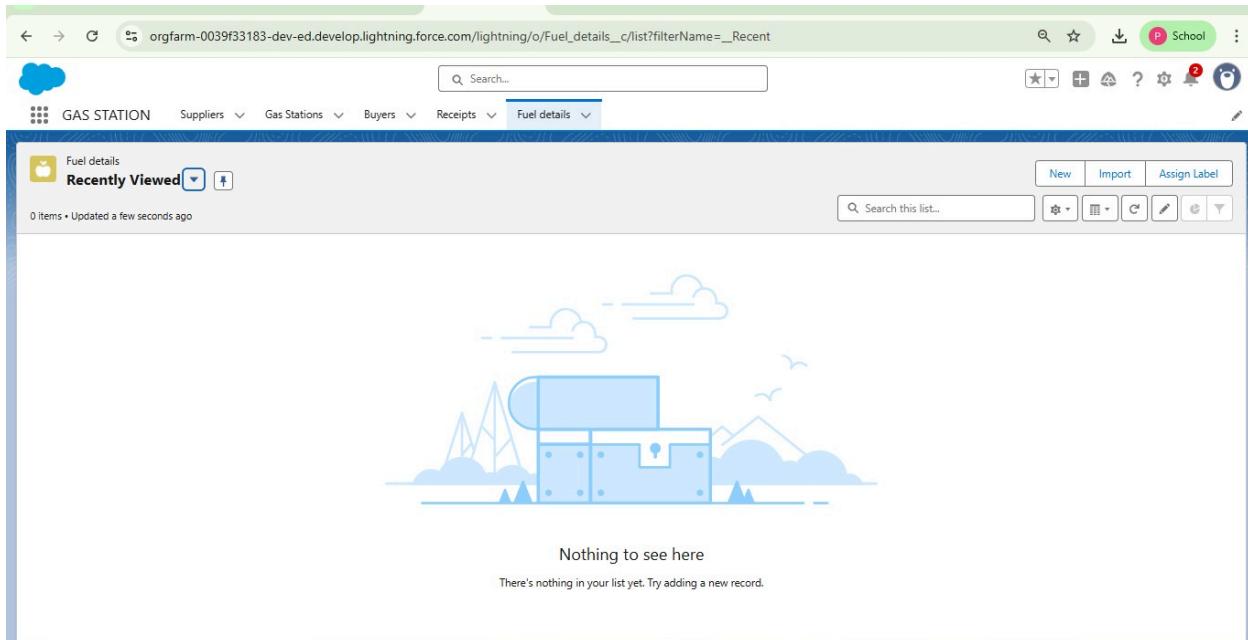
### 3. Create Lightning App:

Created Gas Station App for managing CRM operations. Configured with required custom objects for smooth usage.

- Created Lightning App “GAS STATION”.
- Assigned to System Administrator profile.
- Created objects:

1. Buyer
2. Fuel Details
3. Gas Station
4. Supplier

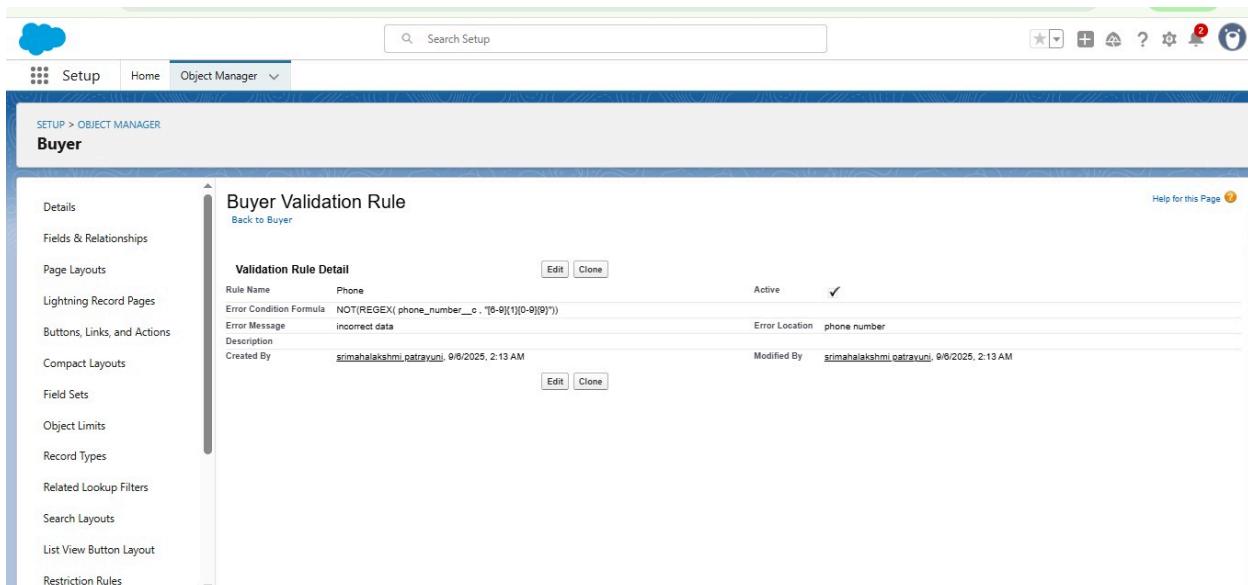
and added necessary fields to each for managing the Gas Station CRM.



#### 4. Fields to be created:

- 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) .

#### Validation in buyer object (phone number)



- 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)

Setup | Home | Object Manager

SETUP > OBJECT MANAGER  
Buyer

Fields & Relationships					
FIELD LABEL		FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount paid	Amount_paid__c	Formula (Number)			
Buyer Name	Name	Auto Number		✓	
Created By	CreatedById	Lookup(User)			
Customer Name	Customer_Name__c	Formula (Text)			
email	email__c	Email			
First Name	First_Name__c	Text(90)			
Fuel filled in vehicle	Fuel_filled_in_vehicle__c	Number(18, 0)			
Fuel Supplied	Fuel_Supplied__c	Number(5, 0)			
Gas Station name	Gas_Station_name__c	Master-Detail(Gas Station)		✓	

Setup | Home | Object Manager

SETUP > OBJECT MANAGER  
Fuel details

Fields & Relationships					
FIELD LABEL		FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)			
Fuel details Name	Name	Auto Number		✓	
Fuel Supplied	Fuel_Supplied__c	Number(5, 0)			
Gas Station	Gas_Station__c	Master-Detail(Gas Station)		✓	
Last Modified By	LastModifiedById	Lookup(User)			
Supplier Name	Supplier__c	Master-Detail(Supplier)		✓	

Fields & Relationships					
8 Items, Sorted by Field Label					
FIELD LABEL		FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By		CreatedById	Lookup(User)		
Fuel Available in bunk		Fuel_Available_in_bunk_c	Formula (Number)		
Fuel Price per Liter		Fuel_Price_per_Liter_c	Number(6, 2)		
Fuel supplied to bunk		Fuel_supplied_to_bunk_c	Roll-Up Summary (SUM Fuel details)		
Fuel used		Fuel_used_c	Roll-Up Summary (SUM Buyer)		
Gas Station Name		Name	Auto Number		✓
Last Modified By		LastModifiedById	Lookup(User)		
Owner		OwnerId	Lookup(User,Group)		✓

Fields & Relationships					
6 Items, Sorted by Field Label					
FIELD LABEL		FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By		CreatedById	Lookup(User)		
Fuel supplied to bunk		Fuel_supplied_to_bunk_c	Roll-Up Summary (SUM Fuel details)		
Last Modified By		LastModifiedById	Lookup(User)		
Owner		OwnerId	Lookup(User,Group)		✓
sum of Fuel supplied		sum_of_Fuel_supplied_c	Roll-Up Summary (SUM Fuel details)		
Supplier Name		Name	Text(80)		✓

## 5. Create Page Layout for Objects:

Created page layouts for objects, organized sections, and arranged fields for better usability.

- Go to Setup → Object Manager → [Object] → Page Layouts → New.
- Create sections as needed and drag relevant fields into each section.
- Arrange fields for clear organization and improved usability.
- Click Save to apply the layout across the object.

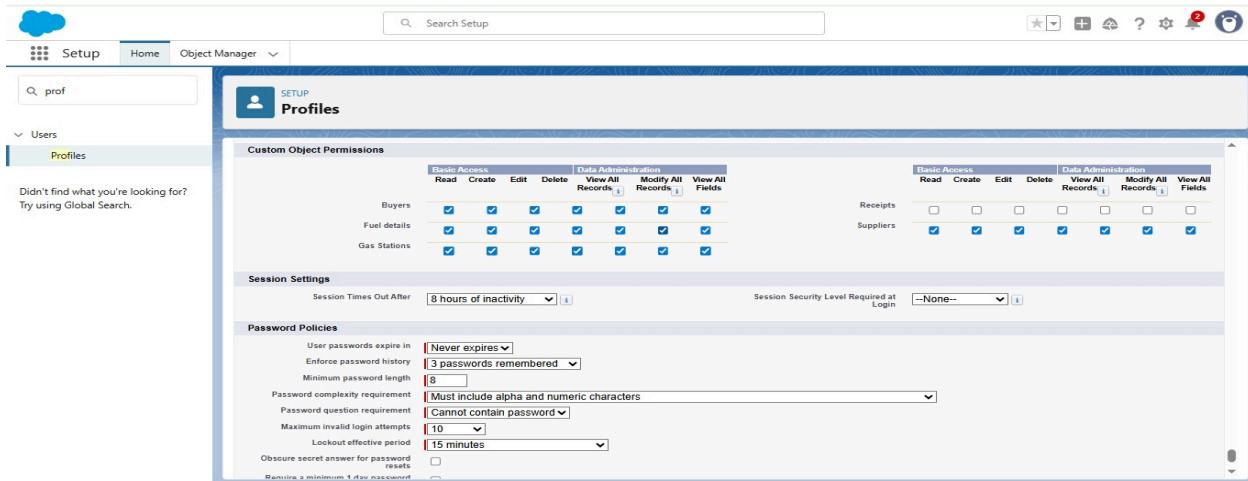
## 6. Create Profiles:

Profiles ensure proper access control and define what data/actions each role can perform. They also help in maintaining security and restricting unnecessary access.

- Cloned an existing profile Standard User for manager and Salesforce Platform User for remaining.
- Give a Profile Name (Manager, Sales Executive, Sales Person) and click Save.

Action	Profile Name	User License	Custom
<input type="checkbox"/>	Manager	Salesforce	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Marketing User	Salesforce	<input type="checkbox"/>
<input type="checkbox"/>	Minimum Access - API Only Integrations	Salesforce Integration	<input type="checkbox"/>
<input type="checkbox"/>	Minimum Access - Salesforce	Salesforce	<input type="checkbox"/>

- Click Edit to configure Custom App Settings and Object Permissions for relevant objects.
- Adjust Session Settings and Password Policies as needed.



## 7. Creating Roles & Hierarchy:

- Go to Setup → Roles → Set Up Roles → Expand All and click Add Role under the relevant parent.
- Enter the Role Label (Manager); the Role Name auto-populates, then click Save.
- Create Sales Executive and Sales Person roles under the Manager with the assigned profiles.
- Ensure the hierarchy reflects reporting lines: Manager → Sales Executive → Sales Person.

The screenshot shows the Salesforce Setup interface with the 'Roles' page selected. The left sidebar has 'Roles' highlighted under 'Users'. The main content area shows a tree view of the 'Role Hierarchy' with nodes like 'Land', 'Sales', 'Service', and 'Marketing'. Each node has 'Edit | Delete | Assign' options. A message at the bottom says ' Didn't find what you're looking for? Try using Global Search.'.

## 8. Creating users:

Users are Salesforce accounts that allow access based on role and profile. In this project, users were created for Manager, Sales Executive, and Sales Person.

- Exec Sales – Sales Executive
- Michelson, Nicklaus – Manager
- Person, Sales – Sales Person

The screenshot shows the Salesforce Setup interface with the 'User Edit' page for 'Niklaus Mikaelson'. The left sidebar has 'Users' selected under 'User Management Settings'. The main form shows user details: First Name (Niklaus), Last Name (Mikaelson), Alias (niklaus), Email (pgfhenriksen950@gmail.com), Username (nik@nik.mn), Nickname (nikm), Title (Title), Company (Company), Department (Department), and Delete (Delete). On the right, the 'Role' is set to 'Manager', 'User License' to 'Salesforce', 'Profile' to 'Manager', and 'Active' is checked. Other checkboxes for Marketing User, Office User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, Data.com User Type (None), Data.com Monthly Addition Limit (500), and Accessibility Mode (Classic Only) are shown.

Created users are displayed.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter_Expert	Chatter	chatter.00dg0000000000000@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	KFC_OrgFarm	KFC	kfc.c1094649914@orgfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	KFC_sales	KFCsales	esther.kfc@gmail.com	Sales Executive	<input checked="" type="checkbox"/>	Sales Executive
<input type="checkbox"/>	Mikaelson_Niklaus	Niklaus	niklaus.mikaelson@gmail.com	Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	Person_Sales	Person	esthersef@gmail.com	Supervisor	<input checked="" type="checkbox"/>	Supervisor
<input type="checkbox"/>	Self_Protect_Protecta	Protecta	22hr14056572@protecta.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	User_Integration	Integ	integration.0200e0000000000@Integ.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User_Security	Security	Insightssecurity@00q0t0000000000@share.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

## 9. Create Permission Sets:

Permission sets extend users' access without changing their profiles. In this project, a permission set was created to give additional access to relevant objects.

- Assigning P1 set to sales executive

Action	Full Name	Active	Role	Profile	User License	Expires On
<input type="checkbox"/>	sales exec	<input checked="" type="checkbox"/>	Sales Executive	Sales Executive	Salesforce Platform	

## 10. OWD (Organization-Wide Defaults)

OWD settings define the baseline level of access users have to records they don't own.

- In this project, Gas Station and Supplier objects were set to Public Read-Only, while roles and profiles controlled additional access.

## 11. Create, View, and Delete Records

User adoption measures how effectively users engage with Salesforce features.

- In this project, creating roles, profiles, permission sets, and intuitive layouts ensured smooth usage and higher adoption of the Gas Station CRM application.

•View record

The screenshot shows a Salesforce page for a 'Fuel details' record named 'fuel-002'. The page has a header with navigation links for GAS STATION, Suppliers, Gas Stations, Buyers, Receipts, and Fuel details. A search bar is at the top right. The main content area displays the following fields:

- Fuel details Name: fuel-002
- Supplier Name: HP
- Gas Station: Gas-001
- Fuel Supplied: 9,000
- Created By: srimahalakshmi patrayuni, 9/23/2025, 2:43 AM
- Last Modified By: srimahalakshmi patrayuni, 9/23/2025, 2:43 AM

At the bottom right of the page, there is a small context menu with options: New Contact, Edit, New Opportunity, and Delete.

### • Delete record

This screenshot is identical to the one above, but a context menu is open on the right side of the page. The 'Delete' option is highlighted in blue, indicating it is selected.

## 12. Creating Reports

Reports are used to analyze 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.

**Report: Gas Stations with Buyers**  
**Amount range**

Total Records	Total Fuel filled in vehicle	Total Amount Paid	
10	733	96,148.00	
<input type="checkbox"/> Fuel Available in bunk	<input type="checkbox"/> Customer name	<input type="checkbox"/> Fuel filled in vehicle	<input type="checkbox"/> Amount Paid
<input type="checkbox"/> -298.00 (3)	kala navya (1)	78	11,700.00
	Subtotal	78	11,700.00
	lal sri (1)	100	15,000.00
	Subtotal	100	15,000.00
	vel kumar (1)	120	18,000.00
	Subtotal	120	18,000.00
	Subtotal	298	44,700.00
<input type="checkbox"/> -247.00 (3)	ham bob (1)	120	14,400.00
	Subtotal	120	14,400.00
	lap lace (1)	7	840.00
	Subtotal	7	840.00
	monn tha (1)	120	14,400.00
	Subtotal	120	14,400.00
	Subtotal	247	29,640.00
<input type="checkbox"/> -188.00 (4)	bonn saru (1)	78	9,048.00

Row Counts  Detail Rows  Subtotals  Grand Total

**Report: Gas Stations with Buyers**  
**Amount range**

Total Records	Total Fuel filled in vehicle	Total Amount Paid
10	733	96,148.00

Sum of Fuel filled in vehicle

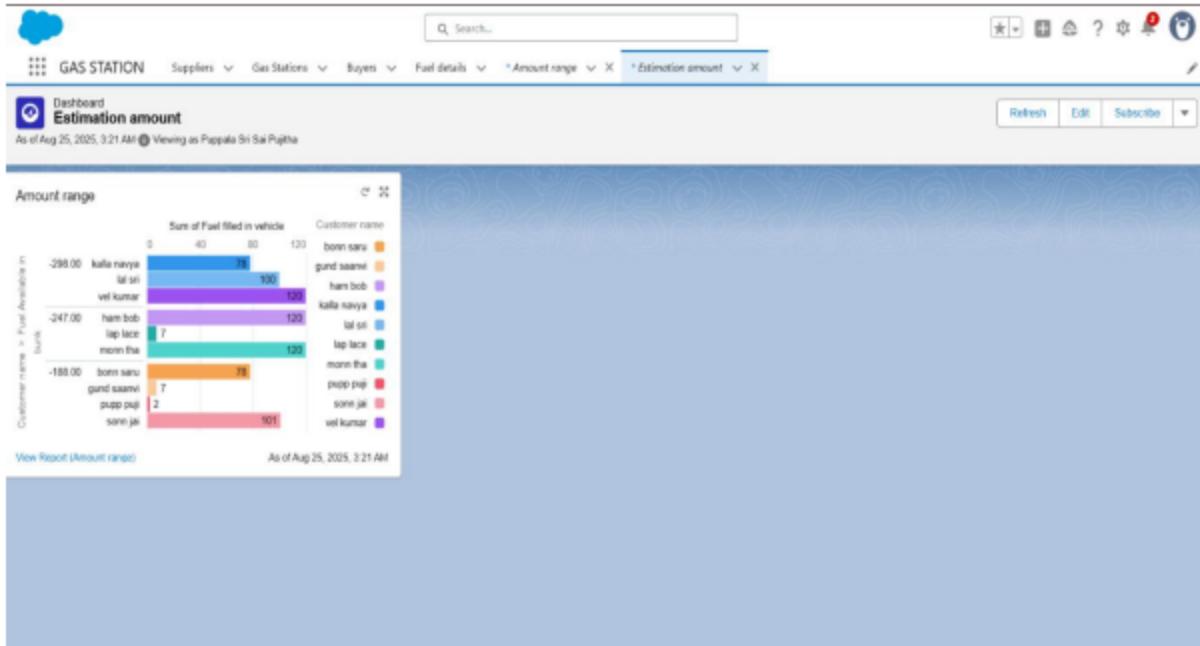
Fuel Available in bunk	Customer name	Fuel filled in vehicle	Amount Paid
<input type="checkbox"/> -298.00 (3)	kala navya (1)	78	11,700.00

Row Counts  Detail Rows  Subtotals  Grand Total

## 13. Creating Dashboards

Dashboards visually display key metrics and report data in charts and graphs.

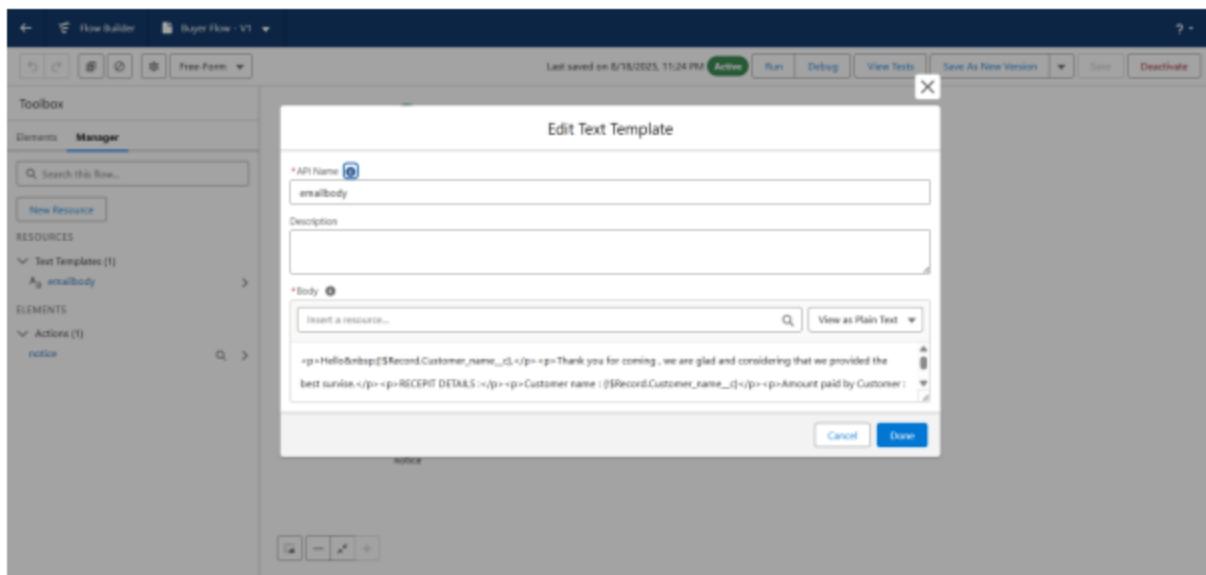
- In this project, dashboards were created to provide insights on fuel estimation, sales, and customer activity

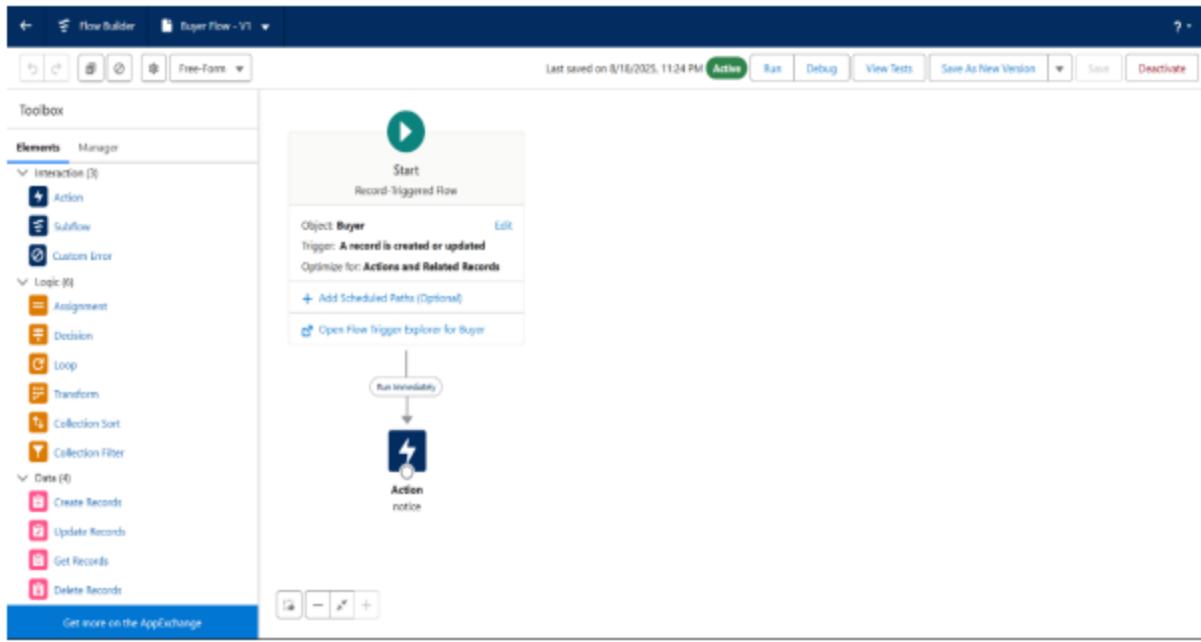


## 14. Creating Flow

Helps in reducing manual effort and ensures quick communication with customers.

- A record-triggered flow was created on the Buyer object.
- The flow triggers when a record is created or updated.
- It automatically sends an email receipt to the customer.
- The email includes Customer Name, Amount Paid, Vehicle Type, and Fuel Intake.





## 15. Creating Apex Triggers

Apex triggers are custom code that execute before or after events on records. Handler (FuelRecordHandler.apxc)

- beforeDeleteInfo → Stops deletion of Fuel Details if supplier qty > 500.
- beforeDeleteGas → Ensures Gas Station fuel price > 50 before insert.

```

1 * public class FuelRecordHandler {
2 *
3 *     public static void beforeDeleteInfo(list<Fuel_details__c> fuelList){
4 *
5 *         //fuelList = [select Id from Fuel_details__c];
6 *
7 *         for(Fuel_details__c ful : fuelList){
8 *
9 *             if(ful.Fuel_supplied__c > 500){
10 *
11 *                 ful.addError('you cannot delete the fuel details record because it is associated with supplier and Gas station records');
12 *
13 *             }
14 *
15 *         }
16 *
17 *     }
18 *
19 *     public static void beforeDeleteGas(list<Gas_Station__c> gasList){
20 *
}

```

The developer console shows the FuelRecordHandler.apxc code. It contains two static void methods: beforeDeleteInfo and beforeDeleteGas. The beforeDeleteInfo method iterates through a list of Fuel\_details\_\_c records and adds an error to any record where the Fuel\_supplied\_\_c value is greater than 500, stating that the record is associated with supplier and Gas station records. The beforeDeleteGas method is currently empty.

### Trigger: Fuel Details (before delete)

Prevents deleting Fuel Details if linked records exist.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [https://orgfarm-aec4fe1cf1-dev-ed-develop.my.salesforce.com/\\_ui/common/apex/debug?openCSPage](https://orgfarm-aec4fe1cf1-dev-ed-develop.my.salesforce.com/_ui/common/apex/debug?openCSPage). The trigger code is:

```
trigger beforeDelete on Fuel_details__c (before Delete) {  
    if(trigger.isbefore && trigger.isDelete){  
        FuelRecordHandler.beforeDeleteInfo(trigger.old);  
    }  
}
```

The console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. A log list at the bottom shows one entry: "User Application Operation Status Read Size".

## Trigger: Gas Station (before insert)

Validates fuel price before record creation.

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [https://orgfarm-aec4fe1cf1-dev-ed-develop.my.salesforce.com/\\_ui/common/apex/debug?openCSPage](https://orgfarm-aec4fe1cf1-dev-ed-develop.my.salesforce.com/_ui/common/apex/debug?openCSPage). The trigger code is:

```
trigger beforeInsert on Gas_Station__c (before insert ) {  
    if(trigger.isbefore && trigger.isinsert){  
        FuelRecordHandler.beforeDeleteGas(trigger.new);  
    }  
}
```

The console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. A log list at the bottom shows one entry: "User Application Operation Status Read Size".

**Feature Enhancements for Gas Station App:** Salesforce Einstein features like AI chatbots, predictive insights, and analytics can be integrated into the Gas Station app to improve customer experience and business decision making.

1. Einstein Bots Integration – Enable AI-powered chatbots to answer customer queries, share fuel prices, and generate receipts automatically.
2. Einstein Next Best Action – Provide personalized offers or discounts to customers based on their fuel purchase history.
3. Einstein Activity Capture – Automatically log customer communications such as emails for better tracking and engagement.
4. Einstein Analytics Dashboard – Visualize sales trends, supplier performance, and customer demand using advanced analytics.
5. Einstein Prediction Builder – Predict customer return behavior to support targeted marketing strategies

### Additional Screenshots:

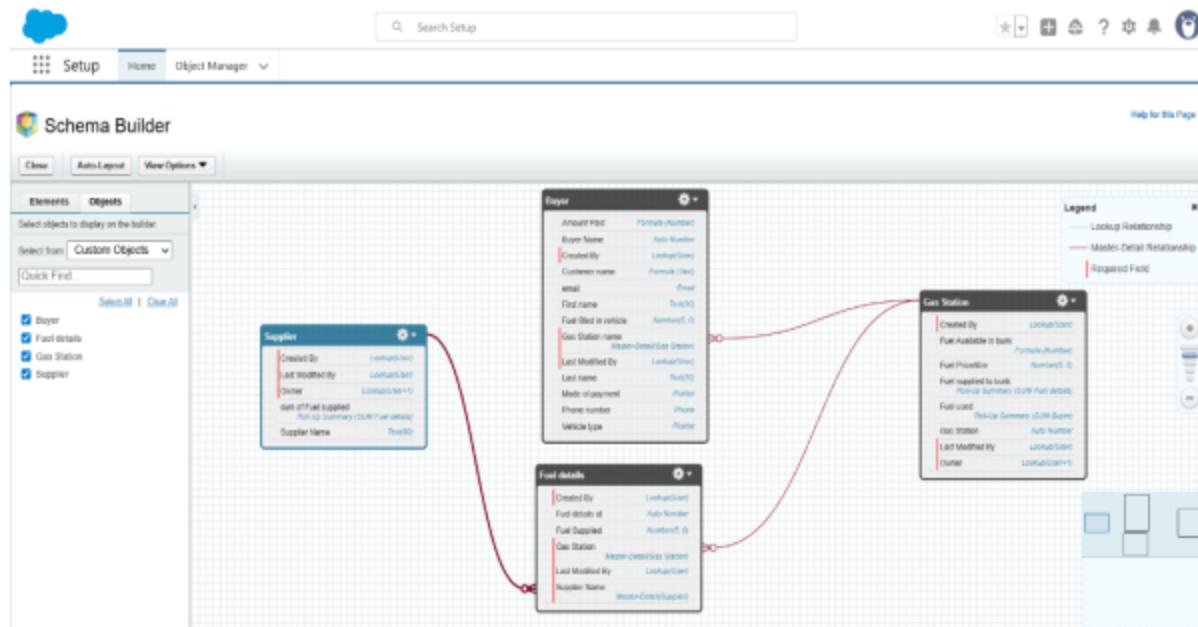


Fig: custom objects relationship

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Search bar with placeholder "Search Setup" and a magnifying glass icon.
- Top Navigation:** Home, Object Manager, and a user icon.
- Left Sidebar:** "User Interface" section with "Rename Tabs and Labels" and "Tabs" selected.
- Page Content:**
  - Custom Tabs Section:** Header "Custom Tabs". Subtext: "You can create new custom tabs to extend Salesforce functionality or to build new application functionality." Description: "Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app." Help link: "Help for this Page".
  - Custom Object Tabs Table:** A table listing custom object tabs. Columns: Action, Label, Tab Style, and Description.

Action	Label	Tab Style	Description
Edit   Del	Dealers	Credit card	
Edit   Del	Fast-foods	CELENO	
Edit   Del	Gas Stations	Car	
Edit   Del	Satellites	Flag	
  - Web Tabs Section:** Subtext: "No Web Tabs have been defined".
  - Visualforce Tabs Section:** Subtext: "No Visualforce Tabs have been defined".

Fig: custom tabs for custom objects

SETUP > OBJECT MANAGER  
**Buyer**

**Details**

**Fields & Relationships**

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

**Field Name:** **Amount\_Fuel**

**Description:**

**Help Text:**

**Data Owner:**

**Field Usage:**

**Data Sensitivity Level:**

**Compliance Categorization:**

Available	Chosen
<input type="button" value="IP"/>	
<input type="button" value="HRPA"/>	
<input type="button" value="GDPR"/>	
<input type="button" value="PCI"/>	

**Formula Options**

**Formula Return Type:** **Number**

**Decimal Places:**

Enter your formula and click CHECK Syntax to check the syntax. Click the Advanced formula button to use additional fields, operators, and functions.  
**Example:** `1.0 * Column_A + 10` [More Examples...](#)

**Simple Formula**

**Insert Field**

**Amount\_Fuel (Required)**

**Amount\_Fuel\_in\_vehicle\_c**

**@es\_Station\_name\_r**

**Fuel\_Price\_litre\_c**

**Functions:**

**Add**

**ACOS**

Fig: formula field of amount paid

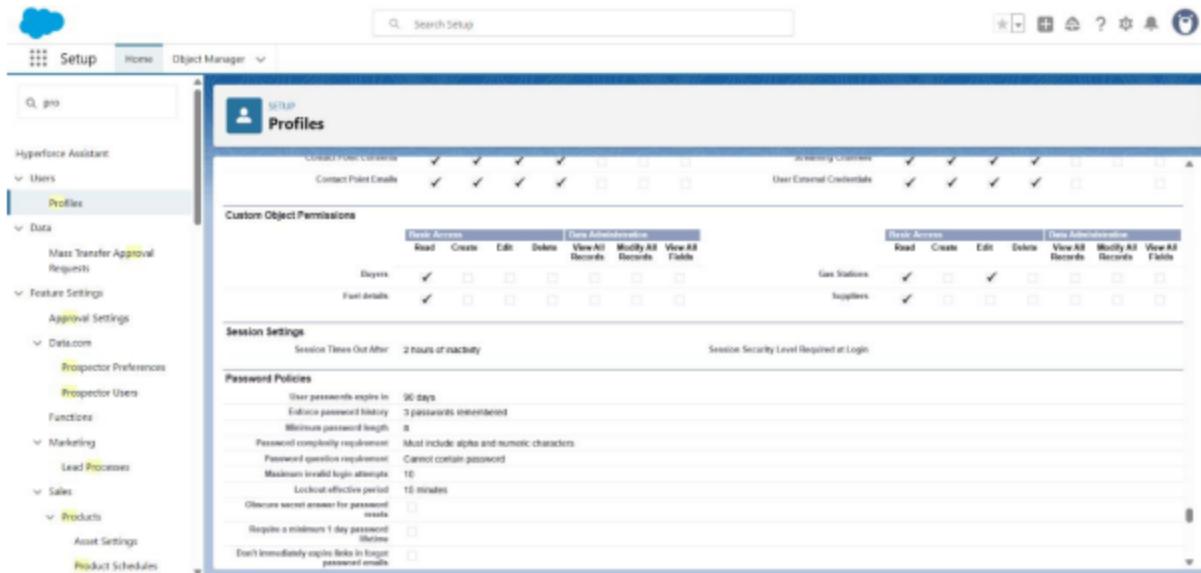


Fig: sales executive profile

Contact Point Emails						User External Credentials											
Custom Object Permissions																	
Session Settings																	
Buyers						Fuel details											
Gas Stations						Suppliers											
Password Policies																	
User passwords expire in	90 days					Enforce password history	3 passwords remembered										
Minimum password length	8					Password complexity requirement	Must include alpha and numeric characters										
Password question replacement	Cannot contain password					Maximum invalid login attempts	10										
Lockout effective period	10 minutes					Obligate secret answer for password resets	Never										
Require a minimum 1 day password lifetime	Lifetime					Don't immediately expire Beta in forget password emails	Never										

Fig: salesperson profile

## Conclusion:

The **Gas Station CRM** project in Salesforce was developed to optimize key business operations, including managing customers, suppliers, and fuel details, while automating essential tasks using Flows and Apex triggers. Custom objects, profiles, permission sets, and page layouts were created to ensure effective data organization and secure access for various users such as Managers, Sales Executives, and Sales Personnel. Automated Flows were designed to send customer receipts, enhancing service delivery, while Apex triggers were implemented to maintain data accuracy during record

insertions and deletions. Reports and dashboards offered valuable insights for managers, enabling them to track performance and improve decision-making. In summary, this project showcases how Salesforce can be effectively utilized to streamline operations, reduce manual efforts, and enhance the overall customer experience in gas station management.