

Application to make the Gas filling Station easy using CRM (admin)

By

SANDU SRIHARI

Student At Vaireddy Venkatadri Institute Of Technology

21BQ1A04B6

sriharisandu9864@gmail.com

Btech In Electronics & Communication Engineering

INDEX PAGE

S.NO	TASK	Page No
1	Abstract	1
2	Salesforce	2-3
3	Object	4-10
4	Tabs	10-12
5	The Lighting App & Fields	13-16
6	Page Layouts	17-19
7	Profiles	20-24
8	Role & Role Hierarchy	25-29
9	Users	30-32
10	Permission Sets	33-35
11	Set Up For OWD & User Adoption	36-37
12	Reports ,Flows&Dashboards	38-42
13	Result	43-44
14	Conclusion	45
15	Apex	45-50

ABSTRACT

The Gas Filling Store CRM Application is a comprehensive solution designed to streamline and simplify the gas filling process for both customers and store owners. It leverages the power of customer relationship management (CRM) to enhance customer experiences, optimize store operations, and improve overall efficiency in the gas filling industry. This project aims to develop a user-friendly and feature-rich application that addresses the specific needs of gas filling stores.

This project aims to address the specific needs of gas filling stores by leveraging cutting-edge technologies for both front-end and back-end development, database management, and cloud infrastructure.

The CRM functionalities are enhanced by features like automated notifications, loyalty program management, and integrations with platforms like Salesforce for optimizing customer relations. Additionally, advanced data analytics tools such as Power BI provide store owners with insights into customer behavior and inventory management.

Security protocols including SSL/TLS encryption and data privacy compliance ensure the protection of sensitive customer information. For enhanced customer convenience, payment gateways like Stripe and PayPal are integrated for seamless transactions. By utilizing a combination of cutting-edge technologies, this CRM application aims to optimize gas filling operations, improve customer engagement, and drive business growth for gas filling store owners.

CRM Features:

- **Salesforce API / HubSpot CRM Integration:** To manage customer relationships, order histories, and personalized offers for returning customers.
- **Automated Customer Notifications:** SMS or email notifications via **Twilio** or **SendGrid**, notifying customers of gas delivery schedules, refills, and special offers.
- **Loyalty Program Management:** Integration of features to support loyalty points, discounts, or membership perks for recurring customers.

SALESFORCE

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

Salesforce is a cloud-based customer relationship management (CRM) platform that helps businesses manage and improve their relationships with customers, clients, and prospects. It offers a suite of tools and services designed to streamline sales, marketing, customer service, and various business processes. Here are key features of Salesforce:

1. Core CRM Capabilities:

Sales Cloud: Manages the sales process, from lead generation to closing deals. It helps sales teams track customer interactions, automate follow-ups, and manage their sales pipeline.

Service Cloud: Manages customer service and support operations. It provides tools for handling customer inquiries, complaints, and case resolution across multiple channels (email, phone, social media).

Marketing Cloud: Manages marketing campaigns and automation, enabling businesses to engage customers through personalized marketing communications, email campaigns, and social media engagement.

Commerce Cloud: Provides eCommerce solutions for managing online stores and customer purchasing experiences.

Analytics Cloud (Tableau): Provides advanced data analytics and visualization tools to gain insights into customer behavior, sales trends, and business performance.

Community Cloud: Helps build and manage online communities for customers, employees, or partners to foster engagement and collaboration.

2. Customization and Integration:

AppExchange: Salesforce's marketplace for third-party applications, where users can find and integrate tools to extend Salesforce's functionality.

Custom Objects and Apps: Salesforce allows businesses to create custom objects (data structures) and custom applications tailored to their specific needs using the Salesforce platform.

APIs and Integration: Salesforce offers robust APIs for integrating with other business systems, such as ERP, accounting software, and external databases.

3. Automation and AI:

Workflow Automation: Salesforce helps automate routine tasks such as sending follow-up emails, creating reports, and updating customer records based on predefined triggers.

Salesforce Einstein: This is Salesforce's AI feature that provides predictive analytics and recommendations. It helps businesses make data-driven decisions by analyzing customer data and predicting outcomes such as sales forecasts and customer behavior.

4. Cloud-Based and Scalable:

Salesforce operates entirely on the cloud, meaning users can access it from anywhere with an internet connection. It is scalable, so businesses of all sizes—from small startups to large enterprises—can use Salesforce according to their needs.

5. Security:

Salesforce provides enterprise-grade security, offering encryption, user authentication, and strict access controls to protect sensitive customer data.

6. Collaboration Tools:

Chatter: A collaboration tool within Salesforce that allows team members to communicate, share files, and collaborate on projects in real-time.

7. Mobile Access:

Salesforce offers mobile applications, allowing sales, marketing, and support teams to manage tasks and customer interactions on-the-go.

Tasks In Application to make the Gas filling Station easy using CRM

Creating a developer org in salesforce.

1. Go to <https://developer.salesforce.com/signup>
2. On the sign up form, enter the following details :

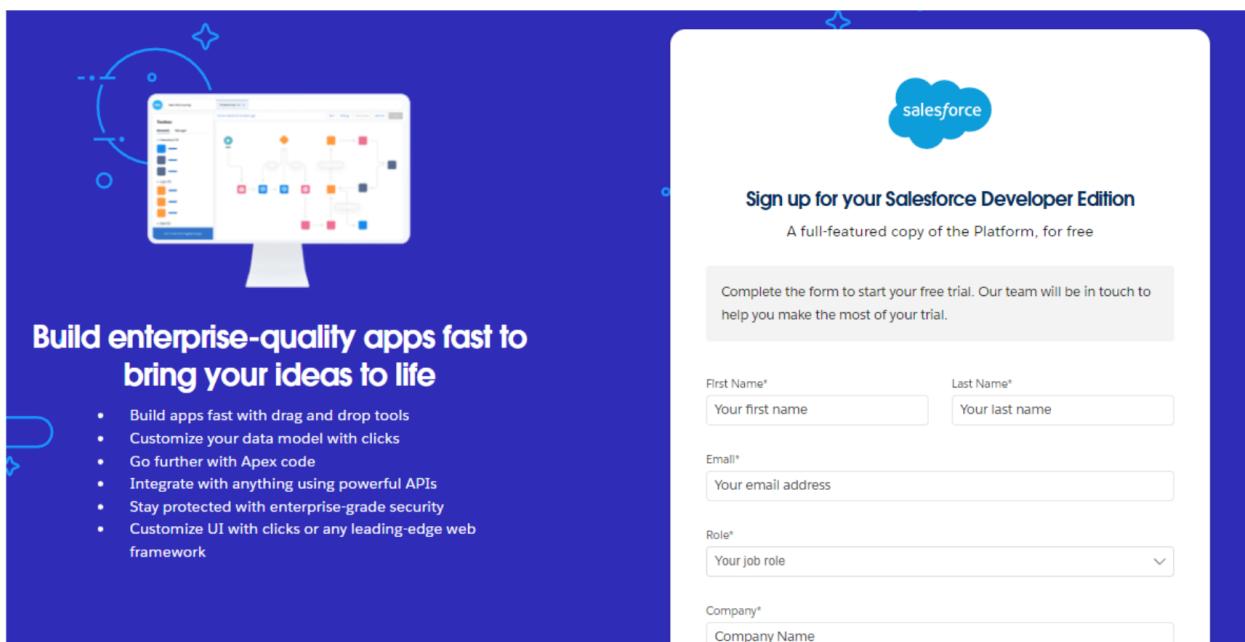
1. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code

Username : should be a combination of your name and company

This need not be an actual email id, you can give anything in the format :

username@organization.com

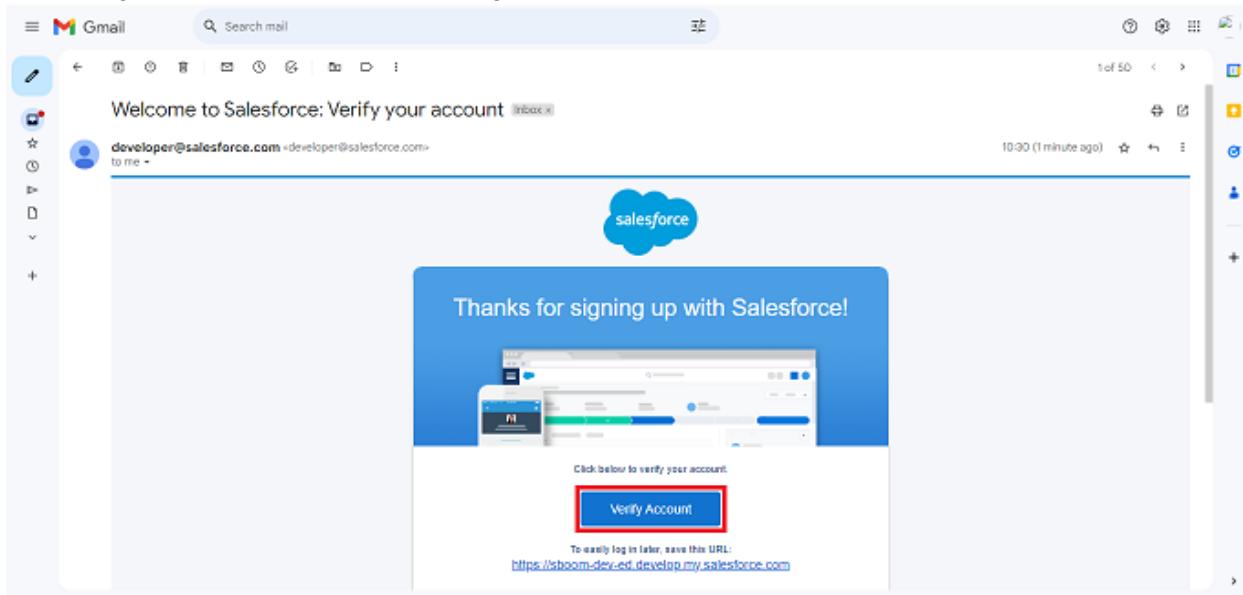
Click on sign me up after filling these.



Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to

activate your account. The email may take 5-10mins.

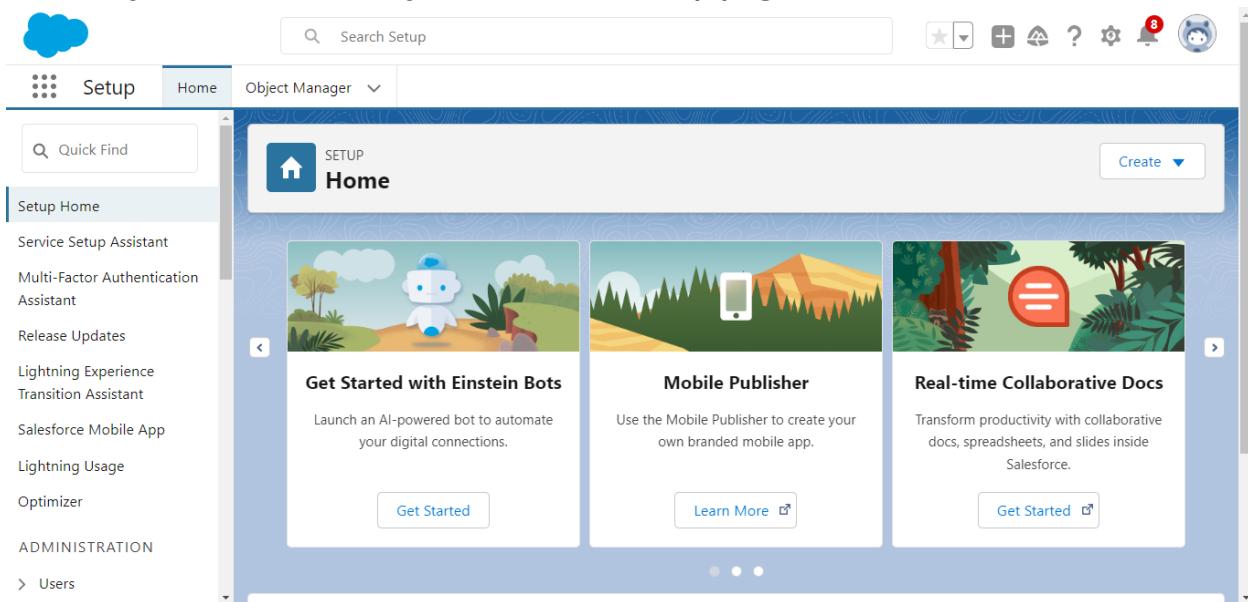


2.click on verify account

3. Give a password and answer a security question and click on change password.

A screenshot of a "Change Your Password" page. The title is "Change Your Password". It instructs the user to "Enter a new password for lead@sb.com. Make sure to include at least:" followed by three requirements: "8 characters", "1 letter", and "1 number". A red box highlights the "New Password" field, which contains "....." and is labeled "Good". Another red box highlights the "Confirm New Password" field, which contains "....." and is labeled "Match". Below these fields is a "Security Question" section with a dropdown menu showing "In what city were you born?". Under "Answer", there is a text input field containing "asdfghjkl". At the bottom is a large red box highlighting the "Change Password" button.

4. when you will redirect to your salesforce setup page.



OBJECT

Create Supplier Object

To create an object:

1. From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.
1. Enter the label name? Supplier
2. Plural label name? Suppliers
3. Enter Record Name Label and Format
 - Record Name ? Supplier Name
 - Data Type ? Name
2. Click on Allow reports and Track Field History,
3. Allow search ? Save.

Setup > OBJECT MANAGER
Supplier

Fields & Relationships
5 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
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)		✓

Create Gas Station Object

To create an object:

1. From the setup page ? Click on Object Manager ? Click on Create ? Click on Custom Object.
 Enter the label name? Gas Station
 Plural label name? Gas Stations
 Enter Record Name Label and Format
 - Record Name ? Gas Station
 - Data Type ? Auto Number
 - Display Format ? Gas-{000}
 - Starting number ? 1
2. Click on Allow reports and Track Field History,
3. Allow search ? Save.

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/litre	Fuel_Price_litre__c	Number(5, 0)		
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)		✓

Create Buyer and Fuel details Objects

Note: Follow the same steps as mentioned in Activity 2 for the Buyer and Receipt objects.

1. Use these display format for the Buyer
 - label name ? Buyer
 - Plural label name ? Buyers
 - Display Format ? Buyer-{000}
 - Starting number ? 1

2. Use these display format for the Fuel details
 - label name ? Fuel details
 - Plural label name ? Fuel details
 - Display Format ? fuel-{000}
 - Starting number ? 1

Buyer Details

Buyer Name

Field Label: Buyer Name
Data Type: Auto Number
Description:
Data Owner:
Field Usage:
Data Sensitivity Level:
Compliance Categorization:
Display Format: Buyer-{000}

Fuel Details

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	Supplier__c	Master-Detail(Supplier)	✓	▼

Tabs

What is Tab : A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

Custom Tabs :

Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard

salesforce.com tabs such as accounts, contacts, and opportunities.

Web Tabs :

Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application.

Visualforce Tabs :

Visualforce Tabs are custom tabs that display a Visualforce page.

Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

Lightning Component Tabs :

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

Lightning Page Tabs :

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

Creating a Custom Tab

To create a Tab:(supplier)

1. Go to setup page ? type Tabs in Quick Find bar ? click on tabs ? New (under custom object tab)
2. Select Object(Supplier) ? Select the tab style ? Next (Add to profiles page) keep it as default ? Next (Add to Custom App) uncheck the include tab
3. Make sure that Append tab to users' existing personal customizations is checked.

4.Click save.

Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are “ Gas station, Buyer, Fuel details”.
2. Follow the same steps as mentioned in Activity -1 .

The screenshot shows the Salesforce Setup interface with the 'Tabs' tab selected in the left sidebar under 'User Interface'. The main content area displays the 'Custom Tabs' section, which allows users to create new custom tabs to extend Salesforce functionality or build new application functionality. The 'Custom Object Tabs' section lists four tabs: 'Buyers' (Books style), 'Fuel details' (Flag style), 'Gas Stations' (Laptop style), and 'Suppliers' (Phone style). Below this are sections for 'Web Tabs' and 'Visualforce Tabs', both of which currently have no tabs defined.

Action	Label	Tab Style	Description
Edit Del	Buyers	Books	
Edit Del	Fuel details	Flag	
Edit Del	Gas Stations	Laptop	
Edit Del	Suppliers	Phone	

The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

To create a lightning app page:

1. Go to setup page ? search “app manager” in quick find ? select “app manager” ? click on New lightning App.
2. Fill the app name in app details as GAS STATION ?Next ? (App option page) keep it as default ? Next ? (Utility Items) keep it as default ? Next.
3. To Add Navigation Items:
4. Select the items (Supplier, Gas Station, Buyer, Receipt) from the search bar and move it using the arrow button ? Next.
5. To Add User Profiles:

Search profiles (System administrator) in the search bar ? click on the arrow button ? save & finish.

The screenshot shows the Salesforce Setup interface with the 'Lightning Experience App Manager' selected in the sidebar. The main area displays a table of 23 apps, sorted by App Name. The table includes columns for App Name, Developer Name, Description, Last Modified Date, App Type, and Visibility. The 'GAS STATION' app is listed as item 10, developed by 'GAS_STATION'. The URL in the browser is https://vnt257-dev-ed.lightning.force.com/lightning/setup/NavigationMenus/home

App Name ↑	Developer Name	Description	Last Modified Date	App Type	Visibil...
All Tabs	AllTabSet	Build CRM Analytics dashboards and apps	01/10/2024, 8:12 pm	Classic	✓
Analytics Studio	Insights	Build CRM Analytics dashboards and apps	01/10/2024, 8:12 pm	Classic	✓
App Launcher	AppLauncher	App Launcher tabs	01/10/2024, 8:12 pm	Classic	✓
Automation	FlowsApp	Automate business processes and repetitive tasks.	01/10/2024, 8:15 pm	Lightning	✓
Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	01/10/2024, 8:14 pm	Lightning	✓
Community	Community	Salesforce CRM Communities	01/10/2024, 8:12 pm	Classic	✓
Content	Content	Salesforce CRM Content	01/10/2024, 8:12 pm	Classic	✓
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	01/10/2024, 8:12 pm	Lightning	✓
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	01/10/2024, 8:12 pm	Lightning	✓
GAS STATION	GAS_STATION	Manage content and media for all of your sites.	01/10/2024, 10:26 pm	Lightning	✓
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	01/10/2024, 8:12 pm	Lightning	✓
Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	01/10/2024, 8:12 pm	Classic	✓
Platform	Platform	The fundamental Lightning Platform	01/10/2024, 8:12 pm	Classic	✓
Queue Management	QueueManagement	Create and manage queues for your business.	01/10/2024, 8:12 pm	Lightning	✓

Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields

1. Standard Fields
2. Custom Fields

3. Standard Fields:

As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

- ? Created By
- ? Owner
- ? Last Modified
- ? Field Made During object Creation

Custom Fields:

On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organizer or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.

Creating Junction Object

Junction object is a custom object that serves as a bridge between two related objects in a many-to-many relationship. It allows you to create a relationship between records of two different objects by creating a many-to-many relationship model.

Creating junction object as Fuel details with Supplier & Gas station

To create junction object

1. Go to the setup page ? click on object manager ? From drop down click edit for Fuel details object.
2. Click on fields & relationship ? click on New.
3. Select “Master-Detail relationship” as data type and click Next.
4. Select the related object “Supplier” and click next.
5. Give Field Label as “Supplier Name” and click Next.
6. Next ? Next ? Save & New.

7. Follow the same steps from 1 to 3.
8. Select the related object “Gas station” and click Next.
9. Give Field Label as “Gas Station” and click Next.
10. Next ? Next ? Save.
11. Below their is an overview of junction object for better understanding.

The image displays two screenshots of the Salesforce Setup interface, specifically the Object Manager section. Both screenshots show the creation of custom fields for the 'Fuel details' object.

Screenshot 1: Gas Station Custom Field

- Object Name:** Fuel details
- Field Label:** Gas Station
- Field Name:** Gas_Station
- API Name:** Gas_Station__c
- Description:** Help Text
- Data Owner:** Data Owner
- Field Usage:** Field Usage
- Data Sensitivity Level:** Data Sensitivity Level
- Compliance Categorization:** Compliance Categorization
- Created By:** Srihari Sandu, 01/10/2024, 10:37 pm
- Modified By:** Srihari Sandu, 01/10/2024, 10:37 pm
- Object Name:** Fuel details
- Data Type:** Master-Detail

Screenshot 2: Supplier Custom Field

- Object Name:** Fuel details
- Field Label:** Supplier
- Field Name:** Supplier
- API Name:** Supplier__c
- Description:** Help Text
- Data Owner:** Data Owner
- Field Usage:** Field Usage
- Data Sensitivity Level:** Data Sensitivity Level
- Compliance Categorization:** Compliance Categorization
- Created By:** Srihari Sandu, 01/10/2024, 10:35 pm
- Modified By:** Srihari Sandu, 01/10/2024, 10:35 pm
- Object Name:** Fuel details
- Data Type:** Master-Detail

Creating a Master-Detail Relationship

Master-detail relationship is a type of relationship between two objects where the master object controls certain behaviors and settings of the detail object. Here are a few use cases that demonstrate the use of master-detail relationships

Creating Master-Detail Relationship between Buyer & Gas Station Object

To Create a Master-Detail relationship

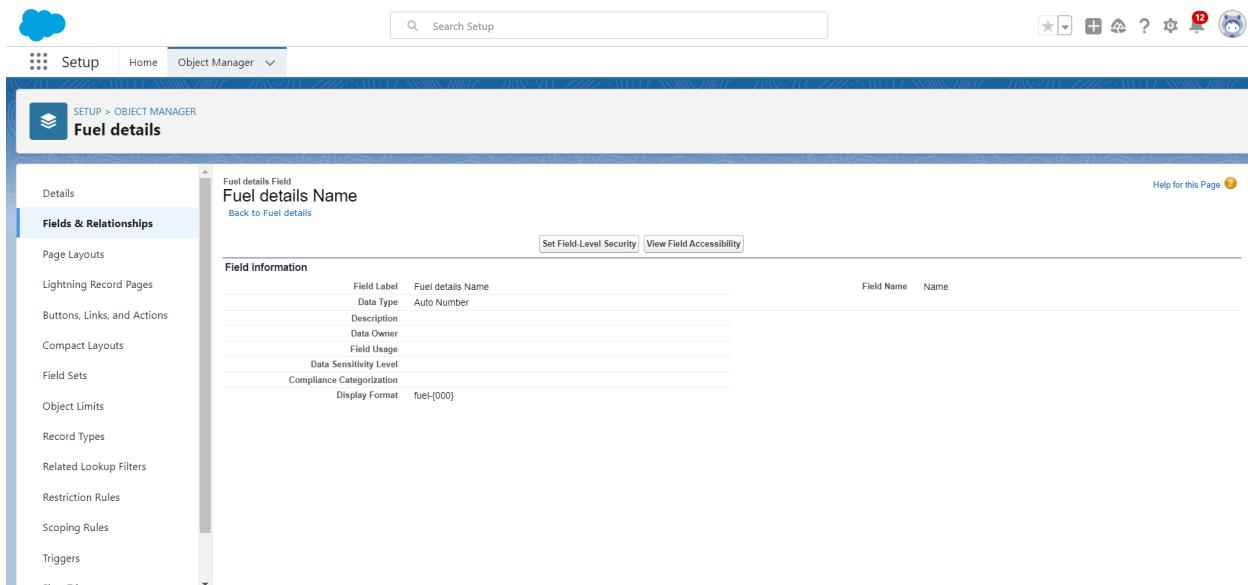
1. Go to the setup page ? click on object manager ? From drop down click edit for Buyer object.

2. Click on fields & relationship ? click on New.
3. Select “Master-Detail relationship” as data type and click Next.
4. Select the related object “ Gas station ”.
5. Give Field Label as “Gas Station name” and click Next.
6. Next ? Next ? Save.

Creating the number field in Fuel details object

Creating the number field in Fuel details object

1. Repeat step 1 and 2 mentioned in activity 1
2. Select Data type as “Number” and click Next.
3. Given the Field Label as “ Fuel Supplied ” and length as “ 5 ”.
4. Field Name will be auto populated, and click on Next? Next ? Save.



Creating the Roll-up Summary

A rollup summary field is a field that summarizes data from a child object to a parent object that share a master-detail relationship. Rollup summary fields can use the COUNT, SUM, MIN, and MAX functions. For example, you could use a rollup summary field to display the total value (amount of fuel supplied) from Fuel details on a related Supplier.

Creating the Roll-up summary field on Supplier & Gas Station Objects.

1. Go to setup ? click on Object Manager ? type object name(Supplier) in search bar

- ? click on the object.
2. Now click on “Fields & Relationships” ? New
 3. Select the data type as “Rollup summary ”,and click Next.
 4. Give the Field label as “ sum of Fuel supplied ”,Field Name will be Auto generated, and click Next.
 5. Select the summarized object as “ Fuel details ”.
 6. Select the Rollup type as “sum”.
 7. Select the field to aggregate as “ Fuel supplied ”, and click Next ? Next ? Save.
 8. Follow the same steps for the Gas station Object from 1 to 3
 9. Give the Field label as “ Fuel supplied to bunk ”,Field Name will be Auto generated, and click Next.
 10. Select the summarized object as “ Fuel details ”.
 11. Select the Rollup type as “sum”.
 12. Select the field to aggregate as “ Fuel supplied ”, and click Next ? Next ? Save.

Note : create the field as “ Fuel filled in vehicle ” using number datatype in Buyer object.

13. Follow the same steps for the Gas station Object from 1 to 3
14. Give the Field label as “ Fuel used ”,Field Name will be Auto generated, and click Next.
15. Select the summarized object as “ Buyer”.
16. Select the Rollup type as “sum”.
17. Select the field to aggregate as “ Fuel filled in vehicle ”, and click Next ? Next ? Save.

The image displays two screenshots of the Salesforce Setup interface, illustrating the creation of custom fields for the Supplier and Gas Station objects.

Supplier Custom Field: sum of Fuel supplied

Custom Field Definition Detail:

- Field Information:**
 - Field Label: sum of Fuel supplied
 - Field Name: sum_of_Fuel_supplied
 - API Name: sum_of_Fuel_supplied_c
 - Description: Help Text
 - Data Owner: Data Owner
 - Field Usage: Data Sensitivity Level
 - Compliance Categorization: Created By: Srihari Sandu 01/10/2024, 11:00 pm
 - Modified By: Srihari Sandu 01/10/2024, 11:00 pm
- Object Name:** Supplier
- Roll-Up Summary Options:**
 - Data Type: Roll-Up Summary
 - Summarized Object: Fuel details
 - Field to Aggregate: Fuel details: Fuel Supplied
 - Filter Criteria: Filter Criteria
 - Summary Type: SUM

Gas Station Custom Field: Fuel supplied to bunk

Custom Field Definition Detail:

- Field Information:**
 - Field Label: Fuel supplied to bunk
 - Field Name: Fuel_supplied_to_bunk
 - API Name: Fuel_supplied_to_bunk_c
 - Description: Help Text
 - Data Owner: Data Owner
 - Field Usage: Data Sensitivity Level
 - Compliance Categorization: Created By: Srihari Sandu 02/10/2024, 7:49 am
 - Modified By: Srihari Sandu 02/10/2024, 7:49 am
- Object Name:** Gas_Station
- Roll-Up Summary Options:**
 - Data Type: Roll-Up Summary
 - Summarized Object: Fuel details
 - Field to Aggregate: Fuel details: Fuel Supplied
 - Filter Criteria: Filter Criteria
 - Summary Type: SUM

Creating Formula Field in Gas Station Object

A formula field is a custom field that can be used to calculate or display data on a Salesforce record.

Formula fields can be used to perform a variety of tasks, such as:

- Calculating totals or averages
- Creating custom fields that display data from other fields
- Validating data entry
- Automating processes

1. Go to setup ? click on Object Manager ? type object name(Gas station) in search bar ? click on the object.
2. Click on fields & relationship ? click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “Fuel Available in bunk” and select formula return type as “Number” and click next.
5. Under Advanced Formula write down the formula and click “Check Syntax” and Save.
6. Insert field formula should be : Fuel_supplied_to_bunk__c - Fuel_Used__c
7. **Creating the Formula field in Buyer Object**

Note : check wheather that the fields that mentioned in the formula field are created are not , if not go to activity 9 and create that fields mentioned in Buyer object

8. Go to setup ? click on Object Manager ? type object name(Buyer) in search bar ? click on the object.
9. Click on fields & relationship ? click on New.
10. Select Data type as “Formula” and click Next.
11. Give Field Label and Field Name as “Customer Name” and select formula return type as “TEXT” and click next.
12. Insert field formula should be : First_Name__c + ' ' + Last_Name__c
13. click “Check Syntax” and Save.

Gas Station Custom Field
Fuel supplied to bunk
[Back to Gas Station](#)

Custom Field Definition Detail	
Field Information	Field Label: Fuel supplied to bunk Field Name: Fuel_supplied_to_bunk API Name: Fuel_supplied_to_bunk__c Description: Help Text: Data Owner: Field Usage: Data Sensitivity Level: Compliance Categorization: Created By: Srihari Sandu 02/10/2024, 7:49 am Modified By: Srihari Sandu 02/10/2024, 7:49 am
Roll-Up Summary Options	Data Type: Roll-Up Summary Summarized Object: Fuel details Field to Aggregate: Fuel_details_Fuel_Supplied Filter Criteria: Summary Type: SUM

Creating Cross Object Formula Field in Buyer Object

A cross-object formula field is a formula field that references fields from another object in Salesforce. This type of formula allows users to calculate and display data from multiple objects on a single record.

Note : check wheather that the fields that mentioned in the formula field are created are not , if not go to activity 9 and create that fields mentioned in Buyer object.

1. Go to setup ? click on Object Manager ? type object name(Buyer) in search bar ? click on the object.
2. Click on fields & relationship ? click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “Amount Paid ” and select formula return type as “Number” and click next.
5. Insert fields formula should be :
`Fuel_filled_in_vehicle__c * Gas_Station_name__r.Fuel_price_liter__c`
6. Under Advanced Formula write down the formula and click “Check Syntax” and Save.

Creating Picklist Field in Buyer Object

1. Go to setup ? click on Object Manager ? type object name(Buyer) in search bar ? click on the object.
2. Click on fields & relationship ? click on New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Vehicle type”, under values select “Enter values, with each value separated by a new line” and enter values as shown below.
5. The values are: two wheeler, three wheeler, four wheeler, six wheeler, eight wheeler and Others.
6. Click Next.
7. Next ? Next ? Save & New.
8. Repeat the process 1 and 2 steps .
9. Enter Field Label as “Mode of payment”, under values select “Enter values, with each value separated by a new line” and enter values as shown below.
10. The values are : credit card, debit card, net banking, upi, cash.
11. Click Next.
12. Next ? Next ? Save & New.

Creating the validation rule

Creating the validation rule for phone number field in Buyer object

Note : check whether that the fields that mentioned in the formula field are created are not , if not go to activity 9 and create that fields mentioned in Buyer object.

1. Go to the setup page ? click on object manager ? From drop down click edit for Buyer object.
2. Click on the validation rule ? click New.
3. Enter the Rule name as “Phone ”.
4. Insert the Error Condition Formula as :-
NOT(REGEX(Phone_Number__c , "[6-9]{1}[0-9]{9}")).
5. Enter the Error Message as “ incorrect data”, select the Error location as Field and select the field as “phone number”, and click Save.

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various object settings like Details, Fields & Relationships, Page Layouts, etc. The main area displays the 'Buyer Validation Rule' details. The validation rule is named 'Phone' and has the formula NOT(REGEX(Phone_Number__c, "[6-9]{1}[0-9]{9}")). It is active and points to the 'Phone Number' field as the error location. The rule was created by Srilhari Sandu on 02/10/2024, 11:35 am, and modified by the same user on the same date and time.

Validation Rule Detail	
Rule Name	Phone
Error Condition Formula	NOT(REGEX(Phone_Number__c, "[6-9]{1}[0-9]{9}"))
Error Message	incorrect data
Description	
Created By	Srilhari Sandu 02/10/2024, 11:35 am
Modified By	Srilhari Sandu 02/10/2024, 11:35 am

Creating Remaining Fields in Objects

s.no	Object name	Fields	
1	Fuel details	Field Name	Data type
		Fuel supplied	number
		Supplier name	Master details
		Gas station	Master details
2	Supplier	Sum of fuel supplied	Rollup summary (Fuel detail object)

3	Gas station	<table border="1"> <tr> <td data-bbox="649 249 1008 435">Fuel supplied to bunk</td><td data-bbox="1008 249 1357 435">Rollup summary (Fuel detail object)</td></tr> <tr> <td data-bbox="649 435 1008 494">Fuel Price/litre</td><td data-bbox="1008 435 1357 494">Number</td></tr> <tr> <td data-bbox="649 494 1008 629">Fuel used</td><td data-bbox="1008 494 1357 629">Rollup summary (Buyer object)</td></tr> <tr> <td data-bbox="649 629 1008 772">Fuel available in bunk</td><td data-bbox="1008 629 1357 772">Formula</td></tr> </table>	Fuel supplied to bunk	Rollup summary (Fuel detail object)	Fuel Price/litre	Number	Fuel used	Rollup summary (Buyer object)	Fuel available in bunk	Formula										
Fuel supplied to bunk	Rollup summary (Fuel detail object)																			
Fuel Price/litre	Number																			
Fuel used	Rollup summary (Buyer object)																			
Fuel available in bunk	Formula																			
4	Buyer	<table border="1"> <tr> <td data-bbox="649 891 1008 1009">First name</td><td data-bbox="1008 891 1357 1009">Text</td></tr> <tr> <td data-bbox="649 1009 1008 1144">Last name</td><td data-bbox="1008 1009 1357 1144">Text</td></tr> <tr> <td data-bbox="649 1144 1008 1262">Customer name</td><td data-bbox="1008 1144 1357 1262">Formula</td></tr> <tr> <td data-bbox="649 1262 1008 1343">Phone number</td><td data-bbox="1008 1262 1357 1343">phone</td></tr> <tr> <td data-bbox="649 1343 1008 1461">email</td><td data-bbox="1008 1343 1357 1461">email</td></tr> <tr> <td data-bbox="649 1461 1008 1575">Fuel filled in vehicle</td><td data-bbox="1008 1461 1357 1575">Number (length = 5)</td></tr> <tr> <td data-bbox="649 1575 1008 1710">Vehicle type</td><td data-bbox="1008 1575 1357 1710">picklist values • Two wheeler • Three wheeler</td></tr> <tr> <td data-bbox="649 1710 1008 1824">Mode of payment</td><td data-bbox="1008 1710 1357 1824">Picklist values • Credit card</td></tr> <tr> <td data-bbox="649 1824 1008 1900">Amount paid</td><td data-bbox="1008 1824 1357 1900">Formula</td></tr> </table>	First name	Text	Last name	Text	Customer name	Formula	Phone number	phone	email	email	Fuel filled in vehicle	Number (length = 5)	Vehicle type	picklist values • Two wheeler • Three wheeler	Mode of payment	Picklist values • Credit card	Amount paid	Formula
First name	Text																			
Last name	Text																			
Customer name	Formula																			
Phone number	phone																			
email	email																			
Fuel filled in vehicle	Number (length = 5)																			
Vehicle type	picklist values • Two wheeler • Three wheeler																			
Mode of payment	Picklist values • Credit card																			
Amount paid	Formula																			

Page layouts

Page Layout in Salesforce allows us to customize the design and organize detail and edit pages of records in Salesforce. Page layouts can be used to control the appearance of fields, related lists, and custom links on standard and custom objects' detail and edit pages.

creating the page layout

To Create a Page layout:

1. Go to Setup ? Click on Object Manager ? Search for the object (Buyer) ? From drop down select the object and click on it.
2. Click on Page layout ? Click on New.
3. Select the existing page layout, and give the page layout name as “customer layout”, and click save.
4. Drag and drop the section field to Buyer details and create the section.
5. Enter the section name as “Persoanl details”, ? click Ok.
6. Now drag the fields to this section that mentioned , they are
 - First name , last name , customer name , phone number, email, Gas station name.
7. Follow the same process for another two sections as shown above , they are
8. One section is “ vehicle info ” , drag the fields that are
 - Fuel filled in vehicle, vehicle type.
9. Another section is “Recepit details ”, and drag the fields that are
 - Mode of payment , Amount paid.
10. Then , Click save.

Profiles

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

Types of profiles in salesforce

1. Standard profiles:

By default salesforce provides below standard profiles.

- Contract Manager
- Read Only
- Marketing User
- Solutions Manager
- Standard User
- System Administrator.

We cannot deleted standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

2. Custom Profiles:

Custom ones defined by us.

They can be deleted if there are no users assigned with that particular one.

Manager Profile

To create a new profile:

1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (Standard User) ? enter profile name (Manager) ? Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Gas station.
4. Scroll down to Custom Object Permissions and Give access permissions for Buyers, Fuel details , gas station and suppliers objects as mentioned in the below diagram.
5. Change the session times out after should be “ 8 hours of inactivity”.
6. Change the password policies as mentioned :
7. User passwords expire in should be “ never expires ”.
8. Minimum password length should be “ 8 ”, and click save.

The screenshot shows the Salesforce Setup interface with the 'Profiles' tab selected. The left sidebar has a search bar and navigation links for Home, Object Manager, and various system settings. The main content area is titled 'Profiles' and shows a list of profiles with columns for Action, Profile Name, User License, and Status. The 'User License' column includes options like Analytics Cloud Integration User, Authenticated Website, External Apps Login, Chatter External, Chatter Free, Chatter Moderator, Contract Manager, XOrg Proxy User, Salesforce, and Customer Community Login. The 'Status' column indicates which profiles have been customized, with several checked off. A navigation bar at the bottom shows page 1 of 2.

Action	Profile Name	User License	Status
<input type="checkbox"/> Edit Clone	Analytics_Cloud_Integration_User	Analytics Cloud Integration User	
<input type="checkbox"/> Edit Clone	Analytics_Cloud_Security_User	Analytics Cloud Integration User	
<input type="checkbox"/> Edit Clone	Authenticated_Website	Authenticated Website	
<input type="checkbox"/> Edit Clone	Authenticated_Website	Authenticated Website	
<input type="checkbox"/> Edit Del ...	R2B_Recruiting_Portal_Buyer_Profile	External Apps Login	✓
<input type="checkbox"/> Edit Clone	Chatter_External_User	Chatter External	
<input type="checkbox"/> Edit Clone	Chatter_Free_User	Chatter Free	
<input type="checkbox"/> Edit Clone	Chatter_Moderator_User	Chatter Free	
<input type="checkbox"/> Edit Clone	Contract_Manager	Salesforce	
<input type="checkbox"/> Edit Clone	Cross_Org_Data_Proxy_User	XOrg Proxy User	
<input type="checkbox"/> Edit Del ...	Custom_Marketing_Profile	Salesforce	✓
<input type="checkbox"/> Edit Del ...	Custom_Sales_Profile	Salesforce	✓
<input type="checkbox"/> Edit Del ...	Custom_Support_Profile	Salesforce	✓
<input type="checkbox"/> Edit Clone	Customer_Community_Login_User	Customer Community Login	

sales executive Profile

1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (Salesforce Platform User) ? enter profile name (sales executive) ? Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Gas station.
4. Scroll down to Custom Object Permissions and Give access permissions for Buyers, Fuel details , gas station and suppliers objects as mentioned in the below diagram.
5. And click save.

The screenshot shows the Salesforce Setup interface with the 'Profiles' section selected. A specific profile named 'sales executive' is displayed. The profile details include:

- Name:** sales executive
- User License:** Salesforce Platform
- Description:** (empty)
- Created By:** Srujan Sandu, 03/10/2024, 8:50 pm
- Modified By:** Srujan Sandu, 06/10/2024, 12:01 am

Page Layouts section:

Standard Object Layouts	Global	Lead
Email Application	Not Assigned [View Assignment]	Location [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Location Group [View Assignment]
Account	Account Layout [View Assignment]	Location Group Assignment [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Object Milestone [View Assignment]

sales person Profile

1. Go to setup ? type profiles in quick find box ? click on profiles ? clone the desired profile (Salesforce Platform User) ? enter profile name (sales person) ? Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Gas station.
4. Scroll down to Custom Object Permissions and Give access permissions for Buyers, Fuel details , gas station and suppliers objects as mentioned in the below diagram.
5. And click save.

The screenshot shows the Salesforce Setup interface with the 'Profiles' section selected. A specific profile named 'sales person' is displayed. The profile details are identical to the 'sales executive' profile:

- Name:** sales person
- User License:** Salesforce Platform
- Description:** (empty)
- Created By:** Srujan Sandu, 03/10/2024, 9:04 pm
- Modified By:** Srujan Sandu, 06/10/2024, 12:01 am

Page Layouts section:

Standard Object Layouts	Global	Lead
Email Application	Not Assigned [View Assignment]	Location [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Location Group [View Assignment]
Account	Account Layout [View Assignment]	Location Group Assignment [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]	Object Milestone [View Assignment]

Role & Role Hierarchy

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Creating Manager Role

Creating Manager Role:

1. Go to quick find ? Search for Roles ? click on set up roles.
2. Click on Expand All and click on add role under whom this role works.
3. Give Label as “Manager” and Role name gets auto populated. Then click on Save.

Creating another roles

Creating another two roles under manager

1. Go to quick find ? Search for Roles ? click on set up roles.
2. Click plus on CEO role, and click add role under manager.
3. Give Label as “sales executive” and Role name gets auto populated. Then click on Save.
4. Repeat the same steps,another role.
5. Click plus on CEO role, and click plus on manager, and click add role under sales executive .

give Label as “sales person” and Role name gets auto populated. Then click on Save.

The screenshot shows the Salesforce Setup interface with the 'Roles' tab selected. On the left, there's a sidebar with 'Users' expanded, showing 'Sales' and 'Service' sections. The main area is titled 'Creating the Role Hierarchy' and displays a tree view of roles. At the top of the hierarchy is 'VINIT'. Below it are 'CEO', 'CFO', 'COO', and 'Manager'. Under 'Manager', there are three roles: 'sales executive', 'sales person', and 'SVP.Customer Service & Support'. Each role has 'Edit | Del | Assign' options. A search bar at the top says 'Search Setup' and a help link 'Help for this Page' is in the top right.

Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Create User

1. Go to setup ? type users in quick find box ? select users ? click New user.
2. Fill in the fields
1. First Name : Niklaus
2. Last Name : Mikaelson

3. Alias : Give a Alias Name
4. Email id : Give your Personal Email id
5. Username : Username should be in this form: text@text.text
6. Nick Name : Give a Nickname
7. Role : Manager
8. User licence : Salesforce
9. Profiles : Manager

Action	Full Name	Alias	Username	Role	Active	Profile
Edit	Chatter Expert	Chatter	chatty.00dd00000d9fz8uai.iucp3okvvgill@chatter.salesforce.com	Manager	<input checked="" type="checkbox"/>	Chatter Free User
Edit	Mikaelson_Niklaus	nmika	sriharisandu@gmail.sandu	Manager	<input checked="" type="checkbox"/>	Manager
Edit	Sandu_Srihari	SSand	sandusrihari@salesforce.com	System Administrator	<input checked="" type="checkbox"/>	System Administrator
Edit	sri_srihari	ssri	srihari@gmail.naidu	sales executive	<input checked="" type="checkbox"/>	sales executive
Edit	sri_sandu	ssru	sriharisandu@gmail.naidu	sales person	<input checked="" type="checkbox"/>	sales person
Edit	User_Integration	integ	integration@00dd00000d9fz8uai.com	Analytics Cloud Integration User	<input checked="" type="checkbox"/>	Analytics Cloud Integration User
Edit	User_Security	sec	insightssecurity@00dd00000d9fz8uai.com	Analytics Cloud Security User	<input checked="" type="checkbox"/>	Analytics Cloud Security User

creating another users

1. Follow the same steps from above activity and create another user using
 1. Role : sales executive
 2. User licence : Salesforce Platform
 3. Profile : sales executive
2. Repeat the steps and create another user using
 1. Role : sales person
 2. User licence : Salesforce Platform
 3. Profile : sales person

User Detail

Name: sandu sun
Alias: ssun
Email: scharisandu8664@gmail.com [Verify]
Username: scharisandu@gmail.nadu
Nickname: User17279707849055692870
Title:
Company:
Department:
Division:
Address:
Time Zone: (GMT+05:30) India Standard Time (Asia/Kolkata)
Locale: English (India)
Language: English
Delegated Approver:
Manager:
Receive Approval Request Emails: Only if I am an approver
Federation ID:
Role: sales_person
User License: Salesforce Platform
Profile: sales_person
Active: ✓
Marketing User:
Offline User:
Knowledge User:
Flow User:
Service Cloud User:
Site.com Contributor User:
Site.com Publisher User:
WDC User:
Mobile Push Registrations: View
Data.com User Type:
Accessibility Mode (Classic Only):
Debug Mode:
High-Contrast Palette on Charts:

Permission sets

A standard permission set consists of a group of common permissions for a particular feature associated with a permission set license. Using a standard permission set saves you time and facilitates administration because you don't need to create the custom permission set.

Creating permission set

A permission set is a collection of settings and permissions that give users access to various tools and functions. Permission sets extend users' functional access without changing their profiles. Users can have only one profile but, depending on the Salesforce edition, they can have multiple permission sets.

1. Go to setup ? type “permission sets” in quick search ? select permission sets ? New
2. Enter the label name as “P1”, API will be auto populated ? save.
3. Under Apps Select object settings.
4. Click on Fuel details object ? click on Edit ? under object permission check for read and create.
5. Click on Save.
6. After saving the permission click on the Manage assignment
7. Now click on the Add Assignment.
8. Now select the users which you have created in user milestone, using sales executive profile and click on Next ? Assign? Done.

Setup For OWD

Organization-Wide Defaults, or OWDs, are the pattern security rules that you can follow for your Salesforce instance. Organization Wide Defaults are utilized to confine who can access what information in your CRM. You can award access through different methods that we will discuss later (sharing principles, Role Hierarchy, Sales Teams, and Account groups, manual sharing, and so forth).

Primarily, there are four levels of access that can be set in Salesforce OWD and they are-

- ? Public Read/Write/Transfer (only available of Enquiry and Cases)
- ? Public Read/Write
- ? Public Read/Only
- ? Private

Create OWD Setting

1. Go to setup ? type “sharing settings ” in quick search ? Click edit
- Scroll down, change the default internal access to “ public read-only” for Gas station and Supplier object.Click save.
- Extra information, By these every profile has their own access, according to their profile.
- But in our case we created a roles and given the roles in such a way that manager can see sales executive and sales person records , sales executive can see the sales person records.

Sharing Settings

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data. Go to [Background Jobs](#) to monitor the progress of a change to an organization-wide default or a parallel sharing recalculations.

Manage sharing settings for: All Objects

Object	Default Internal Access	Default External Access	Grant Access Using Hierarchies
Lead	Public Read/Write/Transfer	Private	✓
Account and Contract	Public Read/Write	Private	✓
Contact	Controlled by Parent	Controlled by Parent	✓
Order	Controlled by Parent	Controlled by Parent	✓
Asset	Controlled by Parent	Controlled by Parent	✓
Opportunity	Public Read/Write	Private	✓
Case	Public Read/Write/Transfer	Private	✓
Campaign	Public Full Access	Private	✓
Campaign Member	Controlled by Campaign	Controlled by Campaign	✓
User	Public Read Only	Private	✓

User Adoption

create a record

To create a record in junction object follow these steps

1. Click on the app launcher locate at left side of the screen.
2. Search for “ Gas station” and click on it.
3. Click on “ fuel details tab”.
4. Click on new and fill the details as shown below figs, and click save.
- 6.
- Creating the supplier record in fuel detail record, by clicking the “ new supplier ”.Fill the details in supplier record and click on save.
7. Creating the Gas station record in fuel details record, by clicking on new gas station.
8. Fill the details in gas station record, Click save.
- 10.
- Fill the remaining details in fuel detail record , and click save.Followed by these create 10 more records in Buyer object.

The screenshots illustrate the 'Suppliers' section of a Microsoft Dynamics 365 application. The top screenshot shows a list of suppliers with two entries: 'HP' and 'Hp'. The bottom screenshot shows the detailed view for the 'HP' supplier, including fields for Supplier Name, sum of Fuel supplied, Owner, and activity history.

Suppliers List View:

- Supplier Name: HP
- Supplier Name: Hp

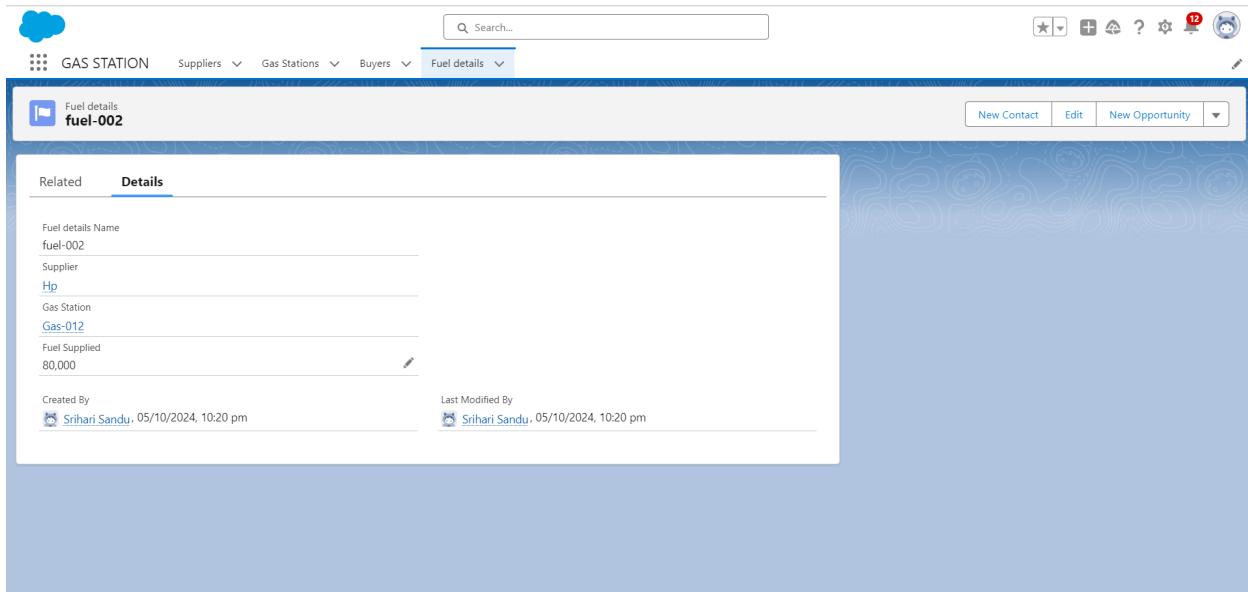
Supplier Details View:

Related	Details
Supplier Name HP	Owner Srihari Sandu
sum of Fuel supplied 0	
Created By Srihari Sandu, 04/10/2024, 10:41 pm	Last Modified By Srihari Sandu, 04/10/2024, 11:01 pm

View a record

To create a record in junction object follow these steps

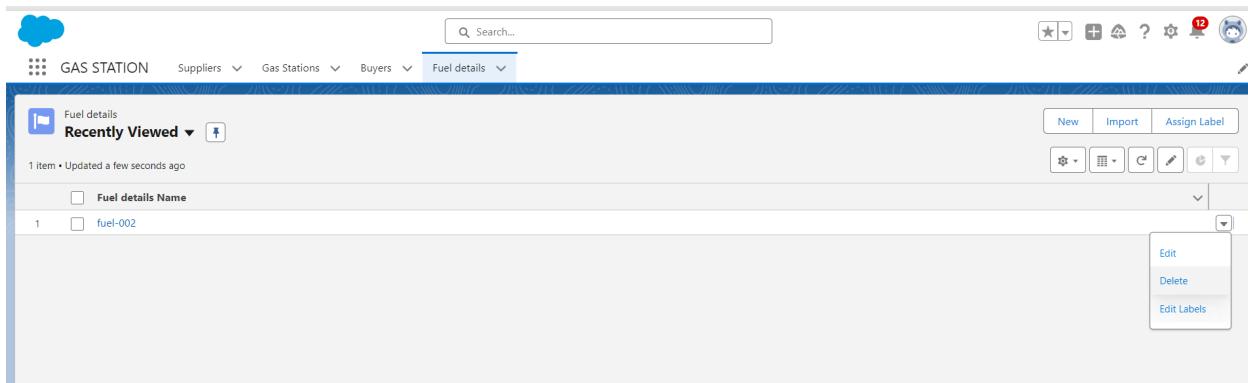
1. Click on the app launcher locate at left side of the screen.
2. Search for “ Gas station” and click on it.
3. Click on “ fuel details tab”.
4. Click on the records that are already created.



Delete a record

To create a record in junction object follow these steps

1. Click on the app launcher locate at left side of the screen.
2. Search for “ Gas station” and click on it.
3. Click on “ fuel details tab”.
4. Click on Arrow at right hand side on that Particular record.
5. Click delete and delete again.



Reports

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

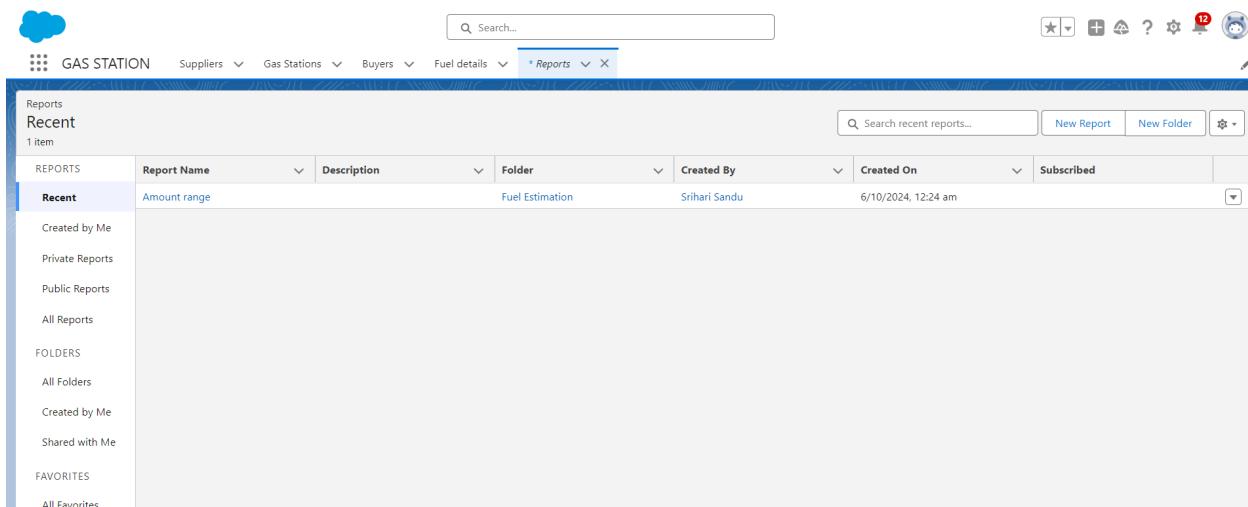
Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports

create a report folder

1. Click on the app launcher and search for reports.
2. Double click on the report, “reports tab” will be autopopulated in navigation bar.
3. Click on the report tab, click on new folder.

4. Give the Folder label as “Fuel Estimation”, Folder unique name will be auto populated.
5. Click save.



The screenshot shows the Salesforce Reports page. At the top, there's a navigation bar with links for GAS STATION, Suppliers, Gas Stations, Buyers, Fuel details, Reports, and a search bar. On the right side of the header are various icons for filtering, saving, and sharing. The main content area has a title 'Recent' with '1 item'. Below this is a table with columns: Report Name, Description, Folder, Created By, Created On, and Subscribed. One row is visible, showing 'Amount range' as the Report Name, 'Fuel Estimation' as the Folder, 'Srihari Sandu' as the Created By, and '6/10/2024, 12:24 am' as the Created On. To the left of the table is a sidebar with sections for Reports (Recent, Created by Me, Private Reports, Public Reports, All Reports), Folders (All Folders, Created by Me, Shared with Me), and Favorites (All Favorites). The 'Recent' section is currently selected.

Report: Gas station with buyers

Amount range

Total Records	Total Fuel filled in vehicle	Total Amount Paid			
11	1,578	1,82,768.00			
Fuel Available in bunk	Customer Name	Buyer Name	Fuel filled in vehicle	Amount Paid	
(11)	(9)	Buyer-001	54	6,264.00	
		Buyer-002	654	75,864.00	
		Buyer-003	555	64,380.00	
		Buyer-004	34	3,944.00	
		Buyer-005	66	7,656.00	
		Buyer-007	7	812.00	
		Buyer-009	43	4,988.00	
		Buyer-010	50	5,800.00	
		Buyer-011	70	7,840.00	
		Subtotal	1,533	1,77,548.00	
		Sa sr (1)	Buyer-008	45	5,220.00
		Subtotal	45	5,220.00	
		SR SA (1)	Buyer-006	-	0.00
		Subtotal	0	0.00	
		Subtotal	1,578	1,82,768.00	

<https://vii257-dev-ed.develop.lightning.force.com/lightning/r/s02d000005hb/QAQ/view>

Sharing a report folder

1. Go to the app ? click on the reports tab.
2. Click on the All folder , click on the arrow for Fuel estimation folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.

All Folders

REPORTS	Name	Created By	Created On	Last Modified By	Last Modified Date
Recent	Einstein Bot Reports	Automated Process	1/10/2024, 8:12 pm	Automated Process	1/10/2024, 8:12 pm
Created by Me	Einstein Bot Reports Spring '23	Automated Process	1/10/2024, 8:12 pm	Automated Process	1/10/2024, 8:12 pm
Private Reports	Einstein Bot Reports Summer '23	Automated Process	1/10/2024, 8:12 pm	Automated Process	1/10/2024, 8:12 pm
Public Reports	Einstein Bot Reports Summer '22	Automated Process	1/10/2024, 8:12 pm	Automated Process	1/10/2024, 8:12 pm
All Reports	Einstein Bot Reports Winter '23	Automated Process	1/10/2024, 8:12 pm	Automated Process	1/10/2024, 8:12 pm
FOLDERS	Enablement Dashboard Reports Spring '24	Automated Process	1/10/2024, 8:12 pm	Automated Process	1/10/2024, 8:12 pm
All Folders	Enablement Dashboard Reports Summer '24	Automated Process	1/10/2024, 8:12 pm	Automated Process	1/10/2024, 8:12 pm
Created by Me	Fuel Estimation	Srihari Sandu	4/10/2024, 11:04 pm	Srihari Sandu	4/10/2024, 11:04 pm
Shared with Me	Gas station with buyers	Srihari Sandu	5/10/2024, 10:25 pm	Srihari Sandu	5/10/2024, 10:25 pm

FAVORITES

All Favorites

Create Report

Note : Before creating report, create latest “10” records in buyer object.

Try to fill every field in each record for better experience.

1. Go to the app ? click on the reports tab
2. Click New Report.
3. select for report type, search for “Gas station with buyers” click on it. And click on start report.
4. Their outline pane is opened already, select the fields that mentioned below in column section.
 1. Fuel filled in vehicle
 2. Amount paid
5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.
 1. Fuel Available in bunk
 2. Customer name
7. Click on conditional formatting located at the bottom of the preview pane.
8. Click on add conditional formatting rule.
9. Change the apply conditional formatting to “ sum of Amount paid ”.
10. Mention the range form “ 1000 to 5000 ”.
11. Dont change the colours, and click on Done.
12. Click apply.
13. Click save, give the report name as “Amount range”, report unique name will be auto populated.
14. Click on select folder, select “ Fuel estimation ” , click select folder
15. Click save.
16. Click save & run , then the preview will be shown below.

The screenshot shows a report titled "Report: Gas station with buyers" for the "Amount range" report. The report details 11 records with a total amount paid of 1,82,768.00. The table includes columns for Buyer Name, Fuel filled in vehicle, and Amount Paid. Conditional formatting is applied, with the last three rows highlighted in green. The bottom of the screen shows report settings for Row Counts, Detail Rows, Subtotals, and Grand Total.

Total Records	Total Fuel filled in vehicle	Total Amount Paid
11	1,578	1,82,768.00
<input type="checkbox"/> Fuel Available in bunk ↑ ↴		
Customer Name ↑ ↴		
Buyer Name ↓		
Fuel filled in vehicle ↓		
Amount Paid ↓		
Subtotal		
Sa sr (1)		
Subtotal		
SR SA (1)		
Subtotal		
Subtotal		
Row Counts	Detail Rows	Subtotals
Grand Total		

Buyer Name	Fuel filled in vehicle	Amount Paid
Buyer-001	54	6,264.00
Buyer-002	654	75,864.00
Buyer-003	555	64,380.00
Buyer-004	34	3,944.00
Buyer-005	66	7,656.00
Buyer-007	7	812.00
Buyer-009	43	4,988.00
Buyer-010	50	5,800.00
Buyer-011	70	7,840.00
Subtotal	1,533	1,77,548.00
Sa sr (1)	Buyer-008	45
Subtotal		5,220.00
SR SA (1)	Buyer-006	-
Subtotal	0	0.00
Subtotal	1,578	1,82,768.00

Dashboards

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

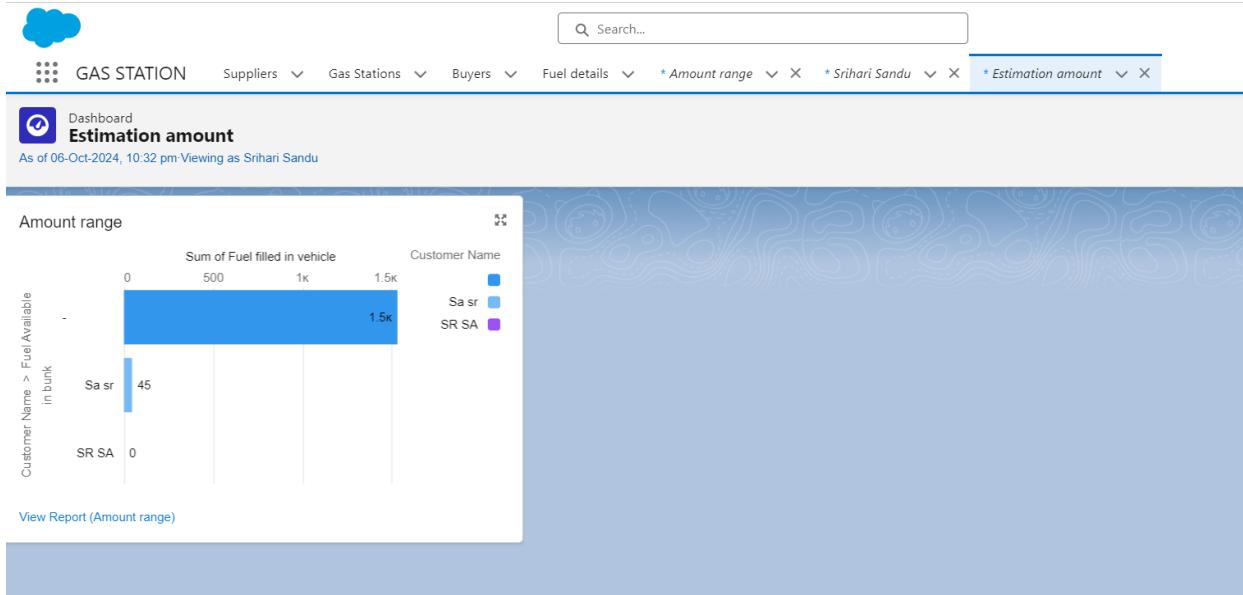
Create Dashboard Folder

1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “Amount estimation dashboard”.
4. Folder unique name will be auto populated.
5. Click save.
6. Follow the same steps, from milestone 12, and activity 2, and provide the sharing settings for the folder that just created.

The screenshot shows a dashboard application interface. At the top, there is a navigation bar with icons for search, favorite, add, help, settings, notifications (with 12), and a user profile. Below the navigation bar, the header includes the application logo (GAS STATION), user information (Suppliers, Gas Stations, Buyers, Fuel details, Srihari Sandu), and a Dashboards tab. The main area displays a table titled "Recent" with one item. The table columns are: DASHBOARDS, Dashboard Name, Description, Folder, Created By, Created On, and Subscribed. The single item listed is "Estimation amount" under the "Recent" category, which is part of the "Amount estimation dashboard" folder, created by Srihari Sandu on 6/10/2024, 12:33 am. On the left side, there is a sidebar with categories: Dashboards, Recent (1 item), Recent, Created by Me, Private Dashboards, and All Dashboards.

Create Dashboard

1. Go to the app ? click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.
3. Select add component.
4. Select a Report and click on select.
5. Click Add then click on Save and then click on Done.
6. Preview is shown below.



Flows

In Salesforce, a flow is a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created without any coding knowledge.

Create a Flow

1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.
2. Select the Record-triggered flow and Click on Create.
3. Select the Object as a “buyer” in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimize the flow for: “Actions and Related Records” and Click on Done
6. Now change the mode form Auto-layout to free-form.
7. Now select the manger option in toolbox, click New resource.
8. Select the resource type as text template.
9. Enter the API name as “emailbody”.
10. In body field paste the syntax that given below.

```
Hello {!$Record.Customer_name_c},
Thank you for coming , we are glad and considering that we provided the best survise.
RECEPIT DETAILS :
Customer name : {!$Record.Customer_name_c}
Amount paid by Customer : {!$Record.Amount_Paid_c}
Vehicle type : {!$Record.Vehicle_type_c}
Fuel intake in vehicle : {!$Record.Fuel_filled_in_vehicle_c}
```

11. Change the view as Rich Text ? View to Plain Text.
12. Click done.
9. Now click on elements, and drag the action element into the preview pane.
10. Their action bar will be opened in that search for “ send email ” and click on it.
11. Give the label name as “ notice ”
12. API name will be auto populated.
13. Enable the body in set input values for the selected action.
14. Select the text template that created.
15. Include recipient address list select the email form the record.
16. Include subject as “ welcome to gas station ”.
17. Click done.
18. Now drag the path form the start to action element.
19. Click on save. Give the Flow label , Flow Api name will be autopopulated.
20. And click save, and click on activate.

The screenshot shows the Salesforce Flow Builder interface. On the left, there's a sidebar with navigation links like Setup, Home, Object Manager, and a search bar. The main area displays a list of 'Flow Definitions' under the heading 'All Flows'. The list includes various flows such as 'Enroll or Unenroll Service Appointment Attendees', 'Even Exchange Flow', 'Exchanges with RMA Returns', etc. Below this is a detailed view of a specific flow named 'flow'. The flow starts with a 'Start' step (Record-Triggered Flow) triggered by 'A record is created or updated' on the 'Buyer' object. This triggers an 'Action' step labeled 'notice'. At the bottom of the flow editor, there are buttons for Save, Run, Debug, View Tests, Activate, and Save As.

Flow Definitions - All Flows

Flow Label	Process Type	Active	Tem...	Package State	Pac...	Last Mod...	Last Modified D...
Enroll or Unenroll Service Appointment Attendees	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Even Exchange Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Exchanges with RMA Returns	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Find Contact Associated with Messaging Session	Individual-Object Linking Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
flow	Autolaunched Flow	<input type="checkbox"/>	<input type="checkbox"/>	Unmanaged	Srihari Sandu	06/10/2024, 1:03 am	
Generate Appointment Invitation	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Generate Payment Link	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Inbound Cancel Appointment	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Inbound Modify Appointment	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Inbound New Appointment	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Inbound New Guest Appointment	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Modify Guest Service Appointment	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			

Flow - V2

Toolbox:

- Elements Manager
- Interaction (3): Action, Subflow, Custom Error
- Logic (6): Assignment, Decision, Loop, Transform, Collection Sort, Collection Filter
- Data (4): Create Records, Update Records, Get Records, Delete Records

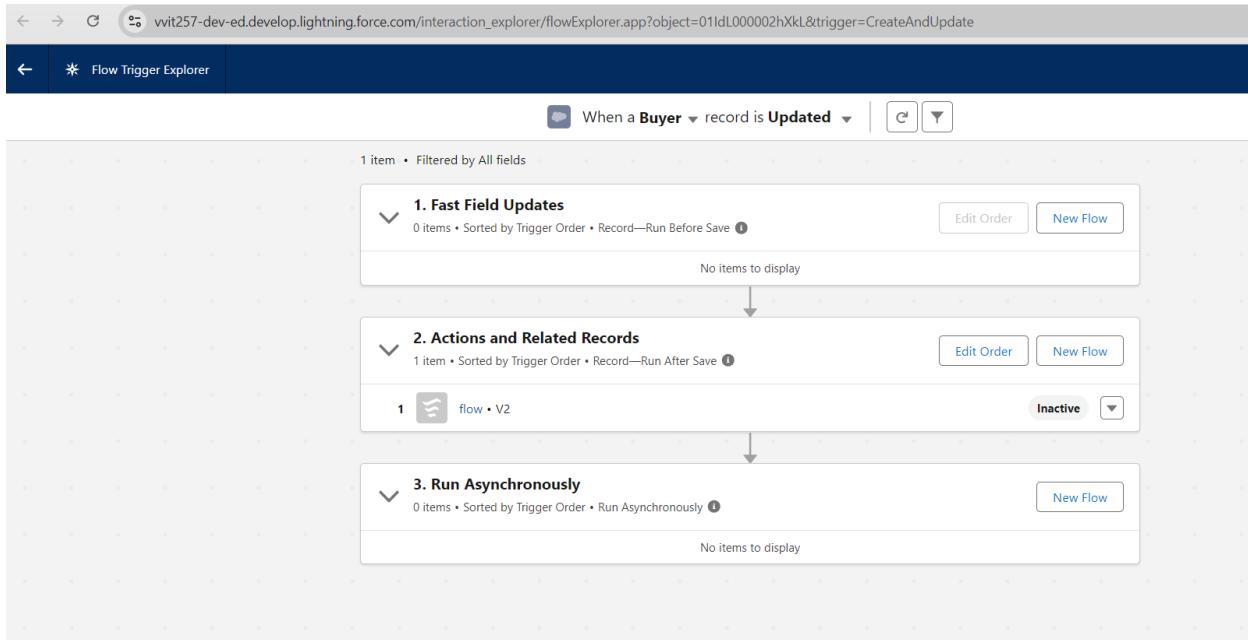
Flow Details:

```

graph TD
    Start((Start)) -- "Run immediately" --> Action[Action notice]
  
```

Start: Record-Triggered Flow
Object: Buyer
Trigger: A record is created or updated
Optimize for: Actions and Related Records
Action: notice

RESULT



The development and implementation of the Gas Filling Store CRM Application demonstrated significant improvements in store operations, customer engagement, and overall business efficiency. The key results include:

1. Enhanced Customer Relationship Management:

- The application provided store owners with a 360-degree view of customer profiles, including transaction history, preferences, and interactions, allowing for more personalized services.
- Automated notifications via SMS and email improved communication, leading to increased customer satisfaction and reduced missed gas refills or deliveries.

2. Optimized Store Operations:

- The integration of inventory management features allowed stores to track gas stock levels in real time, reducing the risk of shortages and optimizing refill schedules.
- Streamlined payment processing with integrated payment gateways like Stripe and PayPal led to faster transactions and a better customer checkout experience.

3. Increased Customer Retention:

- The implementation of loyalty programs and personalized offers boosted customer retention, with customers more likely to return due to the rewards and incentives offered through the app.
- The CRM's ability to track and reward loyal customers increased customer lifetime value, which led to higher sales growth over time.

4. Data-Driven Decision Making:

- Advanced analytics and reporting tools provided actionable insights into sales trends, inventory usage, and customer behavior, allowing store owners to make more informed decisions.

informed decisions regarding stock management and marketing strategies.

- Predictive analytics, powered by machine learning, allowed stores to anticipate demand fluctuations and better manage resources.

5. Scalability and Security:

- The use of cloud services (AWS/Google Cloud) ensured that the system could scale effortlessly as the number of users increased, maintaining high performance without service disruption.

- Implementing industry-standard security protocols like SSL/TLS encryption and GDPR compliance ensured that customer data remained secure, boosting trust in the system.

Conclusion:

The Gas Filling Store CRM Application successfully addressed the operational challenges and customer service needs of gas filling stores. By leveraging cutting-edge CRM technologies, the application improved both store efficiency and customer satisfaction. The platform's ability to provide real-time insights, automate processes, and offer personalized services created a more streamlined and customer-friendly experience. Additionally, the integration of payment gateways, loyalty programs, and predictive analytics further enhanced the value proposition for both store owners and customers.

In conclusion, this CRM solution proved to be a comprehensive tool for the gas filling industry, helping store owners optimize operations, retain customers, and grow their businesses. The scalable nature of the platform and its robust security measures ensured its long-term sustainability and trustworthiness, making it an ideal solution for businesses looking to modernize their gas filling services.

APEX

Example Apex Code for Transaction Record Creation:

```
public class GasTransactionService {

    // Method to create a new gas transaction
    public static void createGasTransaction(Id customerId, Decimal quantity, Decimal
pricePerUnit) {
        Gas_Transaction__c newTransaction = new Gas_Transaction__c(
            Customer__c = customerId,
            Quantity__c = quantity,
            Price_Per_Unit__c = pricePerUnit,
            Total_Amount__c = quantity * pricePerUnit,
            Transaction_Date__c = Date.today()
        );
        insert newTransaction;
    }

    // Method to update loyalty points for a customer after a transaction
    public static void updateCustomerLoyaltyPoints(Id customerId, Decimal amountSpent) {
        Customer__c customer = [SELECT Id, Loyalty_Points__c FROM Customer__c WHERE Id =
:customerId LIMIT 1];
        if (customer != null) {
            customer.Loyalty_Points__c += (amountSpent / 100); // e.g., 1 point per 100 spent
            update customer;
        }
    }
}
```

Example Apex Code for Sending an Email Notification:

```
public class NotificationService {

    public static void sendEmailNotification(Id customerId, String subject, String messageBody)
{
```

```

Customer__c customer = [SELECT Id, Email__c FROM Customer__c WHERE Id =
:customerId LIMIT 1];

if (customer.Email__c != null) {
    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    mail.setToAddresses(new String[] { customer.Email__c });
    mail.setSubject(subject);
    mail.setPlainTextBody(messageBody);
    Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });
}

// Trigger email after new transaction
public static void sendTransactionNotification(Id transactionId) {
    Gas_Transaction__c transaction = [SELECT Id, Customer__r.Email__c, Total_Amount__c
FROM Gas_Transaction__c WHERE Id = :transactionId LIMIT 1];

    if (transaction.Customer__r.Email__c != null) {
        String subject = 'Thank you for your purchase!';
        String body = 'Dear Customer, thank you for your recent purchase of gas. The total
amount is ' + transaction.Total_Amount__c + ':';
        sendEmailNotification(transaction.Customer__r.Id, subject, body);
    }
}

```

Example: Predicting Next Refill Date Based on Past Purchases

```

public class PredictiveAnalyticsService {

    // Method to predict the next gas refill date based on past purchase behavior
    public static Date predictNextRefill(Id customerId) {

```

```

List<Gas_Transaction__c> transactions = [SELECT Transaction_Date__c, Quantity__c
                                         FROM Gas_Transaction__c
                                         WHERE Customer__c = :customerId
                                         ORDER BY Transaction_Date__c DESC
                                         LIMIT 5];

if (transactions.size() > 1) {
    // Calculate average time between transactions
    Integer totalDaysBetweenPurchases = 0;
    for (Integer i = 0; i < transactions.size() - 1; i++) {
        totalDaysBetweenPurchases +=
            transactions[i].Transaction_Date__c.daysBetween(transactions[i+1].Transaction_Date__c);
    }
    Integer avgDaysBetweenPurchases = totalDaysBetweenPurchases / (transactions.size() - 1);
    return transactions[0].Transaction_Date__c.addDays(avgDaysBetweenPurchases);
}

return null; // Not enough data to make a prediction
}
}

```

Example: Automatic Low Stock Alert

```

public class InventoryService {

    public static void checkAndAlertLowStock() {
        List<Product__c> lowStockItems = [SELECT Id, Name, Stock_Level__c
                                         FROM Product__c
                                         WHERE Stock_Level__c < 10];

        if (lowStockItems.size() > 0) {

```

```

        for (Product__c product : lowStockItems) {
            sendLowStockAlert(product.Name);
        }
    }
}

// Sends an email to store owner when stock is low
private static void sendLowStockAlert(String productName) {
    String emailAddress = 'storeowner@example.com'; // Store owner's email

    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    mail.setToAddresses(new String[] { emailAddress });
    mail.setSubject('Low Stock Alert: ' + productName);
    mail.setPlainTextBody('The stock level of ' + productName + ' is below the threshold.
Please restock soon.');
    Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });
}
}

```

Example: Loyalty Points Tracking

```

trigger LoyaltyPointsTrigger on Gas_Transaction__c (after insert) {
    for (Gas_Transaction__c transaction : Trigger.New) {
        GasTransactionService.updateCustomerLoyaltyPoints(transaction.Customer__c,
transaction.Total_Amount__c);
    }
}

```

This simple **Apex trigger** updates customer loyalty points each time a new gas transaction is inserted into the database.

These examples showcase how Salesforce's Apex language can be used to implement key functionalities such as transaction management, notifications, predictive analytics, and inventory management, all within the Gas Filling Store CRM Application.

