Gas Station CRM: A Salesforce-Based Solution for Efficient Gas Filling Operations

Introduction:

This project, "**Application to make the Gas Filling Station Easy using CRM (Developer)**", is built on Salesforce to simplify and automate gas station operations. It manages customer data, booking schedules, inventory, and billing through a single CRM platform. By automating manual processes and providing real-time updates, the application improves efficiency, enhances customer satisfaction, and reduces operational delays.

Salesforce:

Introduction:

Salesforce is a **cloud-based Customer Relationship Management (CRM) platform** that helps businesses manage their relationships and interactions with customers, clients, and prospects. It provides tools and applications to streamline sales, marketing, customer service, and other business processes.

Key Features of Salesforce:

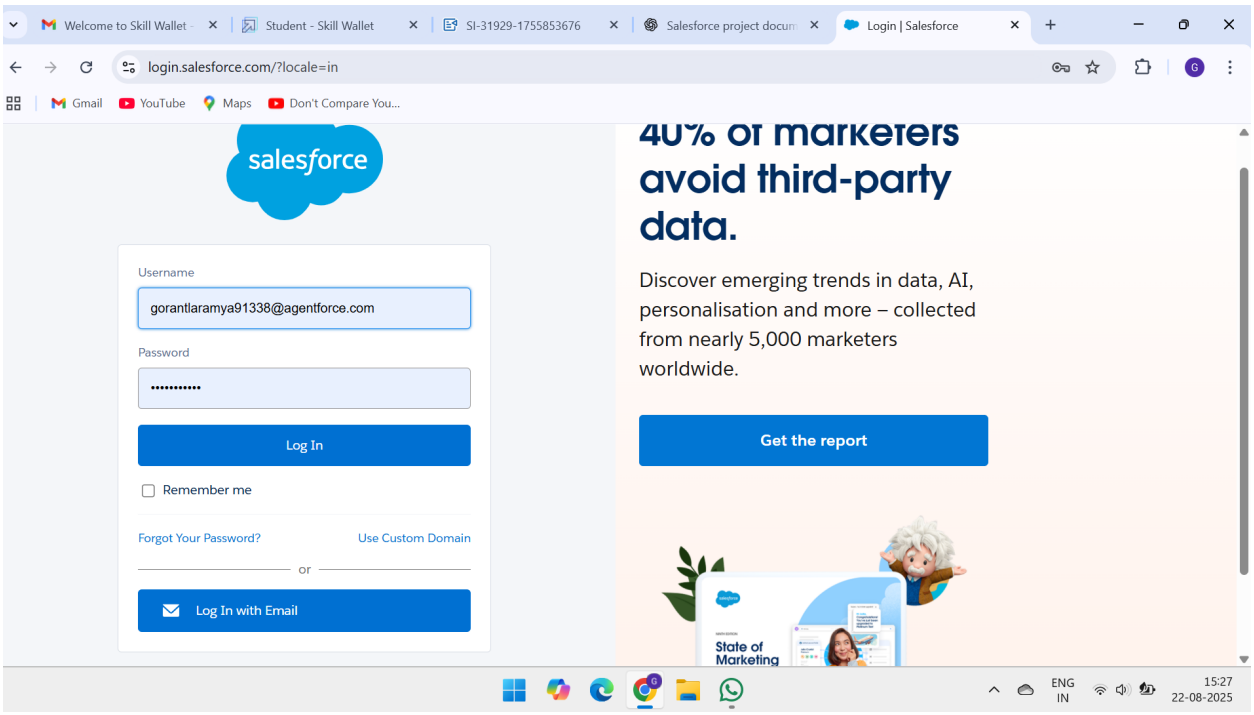
1. **Cloud-Based:** No need for physical servers or installations; it works online.
2. **Customer Management:** Stores and manages customer details, history, and communication.
3. **Automation:** Automates workflows, approvals, and repetitive tasks.

Creating a Salesforce Developer Account:

To develop and customize the "**Application to make the Gas Filling Station Easy using CRM**", a Salesforce Developer Account is required. This free account provides access to the full set of Salesforce tools needed for building and testing applications.

Steps to Create a Developer Account:

1. **Visit the Salesforce Developer Website:**
   Go to https://developer.salesforce.com/signup.
2. **Fill Out the Registration Form:**
   - Enter your **first name, last name, email, and company name** (use your project name if needed).
   - Choose your **Country/Region** and select a **Username** (in email format, e.g., ramya.gascrm@example.com).
3. **Verify Your Email:**
   - Check your email inbox for a **verification link from Salesforce**.
   - Click the link to activate your account.
4. **Set Your Password and Security Question:**
   - After verification, create a strong password and select a security question for account recovery.
5. **Login to Developer Edition:**
   - Access your developer account at https://login.salesforce.com.
   - Once logged in, you can start creating custom objects, workflows, and automation for your gas station CRM project.



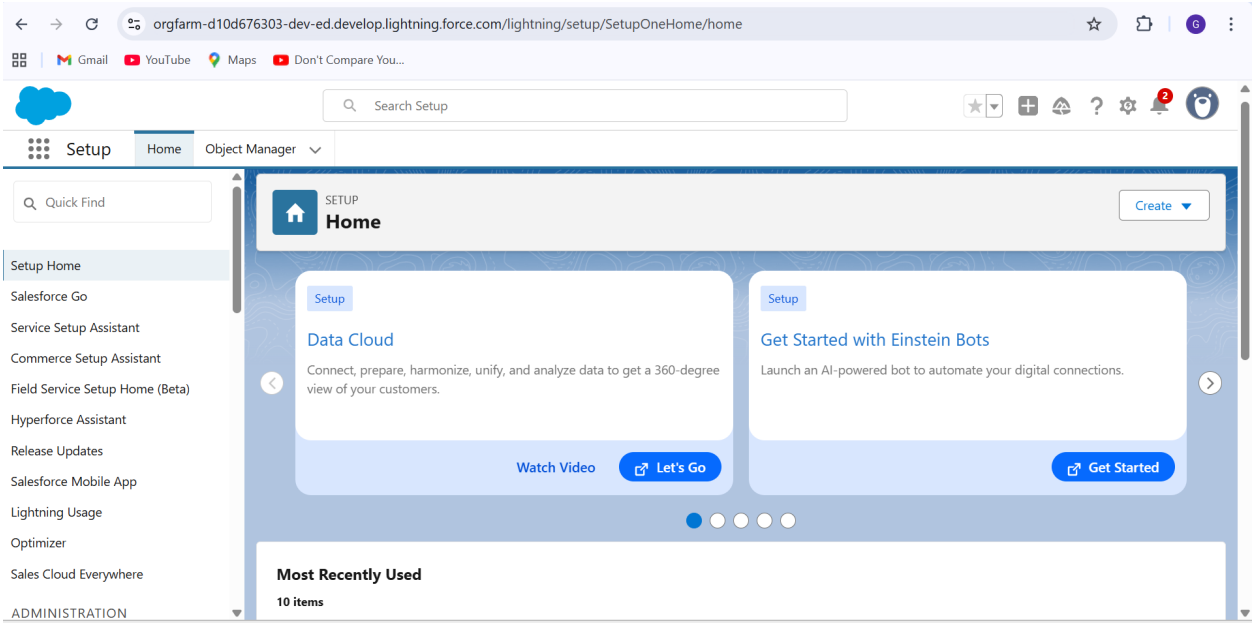
Account Activation

After registering for the Salesforce Developer Account, you need to activate it to access the platform and begin development.

Steps to Activate Your Account:

1. **Check Your Email:**
   - Open the email inbox you used during registration.
   - Look for an activation email from **Salesforce**.
2. **Click the Activation Link:**
   - Open the email and click the provided activation link to confirm your account.
3. **Set Your Password and Security Question:**
   - Create a strong password (include uppercase, lowercase, numbers, and special characters).
   - Choose a security question for future account recovery.
4. **Login to Your Developer Account:**
   - Go to https://login.salesforce.com.
   - Enter your registered username and the password you set.
5. **Access Developer Edition:**

- Once logged in, you will be redirected to your Salesforce Developer Edition, where you can start creating your CRM application for the gas filling station.



Object:

Creating the Supplier Object:

The **Supplier Object** is designed to store and manage details of gas suppliers who provide stock to the filling station. This custom object helps track supplier information, maintain inventory flow, and streamline communication.

Steps to Create Supplier Object in Salesforce:

1. **Login to Salesforce Developer Account**
   - Go to https://login.salesforce.com and sign in with your developer credentials.
2. **Navigate to Object Manager**
   - Click on the **Setup (gear icon)** in the top-right corner.
   - Select **Object Manager** from the Setup menu.
3. **Create a New Custom Object**
   - Click **Create → Custom Object**.
   - Enter the following details:
     - **Label:** Supplier
     - **Plural Label:** Suppliers

        - **Object Name:** Supplier
        - **Record Name:** Supplier Name (Text)
    - Enable the options:
        - Allow Reports
        - Allow Activities
        - Track Field History (optional)
4. **Define Custom Fields for Supplier Object**
   Add fields to store supplier-related data, such as:
    - **Supplier ID** (Auto Number)
    - **Company Name** (Text)
    - **Contact Person** (Text)
    - **Phone Number** (Phone)
    - **Email** (Email)
    - **Address** (Text Area)
    - **Gas Type** (Picklist – LPG, CNG, PNG, etc.)
5. **Set Permissions & Access**
    - Assign required permissions to **Admin and Staff profiles**.
    - Make the object available in the **App Navigation** for relevant users.
6. **Add Tabs and Layouts**
    - Create a new tab for the Supplier object to make it easily accessible.
    - Customize the page layout to include the fields mentioned above.
7. **Save and Test**
    - Save the object and create a sample supplier record to ensure it works correctly.

Create Gas Station Object:

The **Gas Station Object** is used to store and manage details of each gas filling station within the CRM system. This object helps track station locations, capacities,

and operational details, enabling efficient management across multiple branches.

Steps to Create Gas Station Object in Salesforce:

1. **Login to Salesforce Developer Account**
    - Visit https://login.salesforce.com and log in with your developer credentials.
2. **Open Object Manager**
    - Click the **Setup (gear icon)** in the top-right corner.
    - Select **Object Manager** from the Setup menu.

3. **Create a New Custom Object**
    - Click **Create → Custom Object**.
    - Fill in the following details:
        - **Label:** Gas Station
        - **Plural Label:** Gas Stations
        - **Object Name:** Gas_Station
        - **Record Name:** Station Name (Text)
    - Enable:
        - Allow Reports
        - Allow Activities
        - Track Field History (if needed)
4. **Add Custom Fields for Gas Station Object**
   Include fields that define each station, such as:
    - **Station ID** (Auto Number)
    - **Location** (Text)
    - **Manager Name** (Text)
    - **Contact Number** (Phone)
    - **Email** (Email)
    - **Capacity** (Number – e.g., Maximum cylinders or liters stored)
    - **Operating Hours** (Text or Time)
    - **Status** (Picklist – Active, Inactive, Under Maintenance)
5. **Set Access Permissions**
    - Grant permissions for Admin and Staff profiles.
    - Restrict customer-level access if not required.
6. **Create Tabs and Page Layouts**
    - Add a tab for the Gas Station object in the CRM navigation bar.
    - Customize the layout to include the defined fields.
7. **Save and Test**
    - Save the object and create a sample gas station record to verify its setup.

Creating the Buyer Object

The **Buyer Object** stores customer details for individuals or organizations purchasing gas from the station. It enables effective customer relationship management, tracking their orders, and communication.

Steps to Create Buyer Object:

1. **Login to Salesforce Developer Account**
   - Go to https://login.salesforce.com and log in.
2. **Open Object Manager**
   - Click the **Setup (gear icon) → Object Manager**.
3. **Create a New Custom Object**
   - Click **Create → Custom Object**.
   - Details:
     - **Label:** Buyer
     - **Plural Label:** Buyers
     - **Object Name:** Buyer
     - **Record Name:** Buyer Name (Text)
   - Enable:
     - Allow Reports
     - Allow Activities
     - Track Field History (optional)
4. **Add Custom Fields for Buyer Object**
   - **Buyer ID** (Auto Number)
   - **Full Name / Company Name** (Text)
   - **Contact Number** (Phone)
   - **Email Address** (Email)
   - **Billing Address** (Text Area)
   - **Preferred Fuel Type** (Picklist – LPG, CNG, PNG, etc.)
5. **Set Access & Permissions**
   - Assign permissions for Admin and Staff profiles.
6. **Create Tabs and Layouts**
   - Add a tab for easy access.
   - Customize page layout with the above fields.
7. **Save and Test**
   - Create a sample Buyer record to confirm functionality.

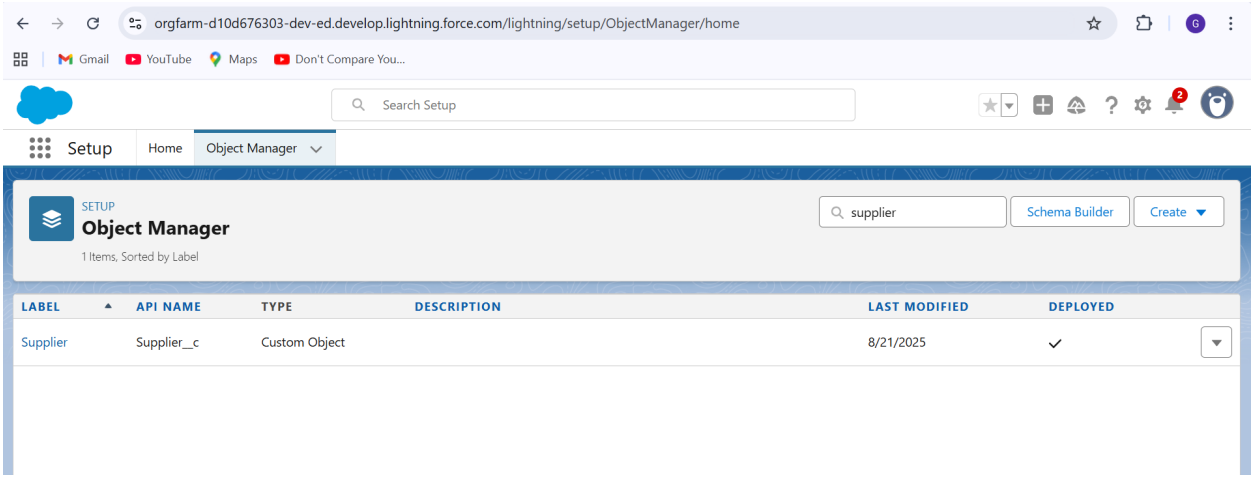
Creating the Fuel Details Object

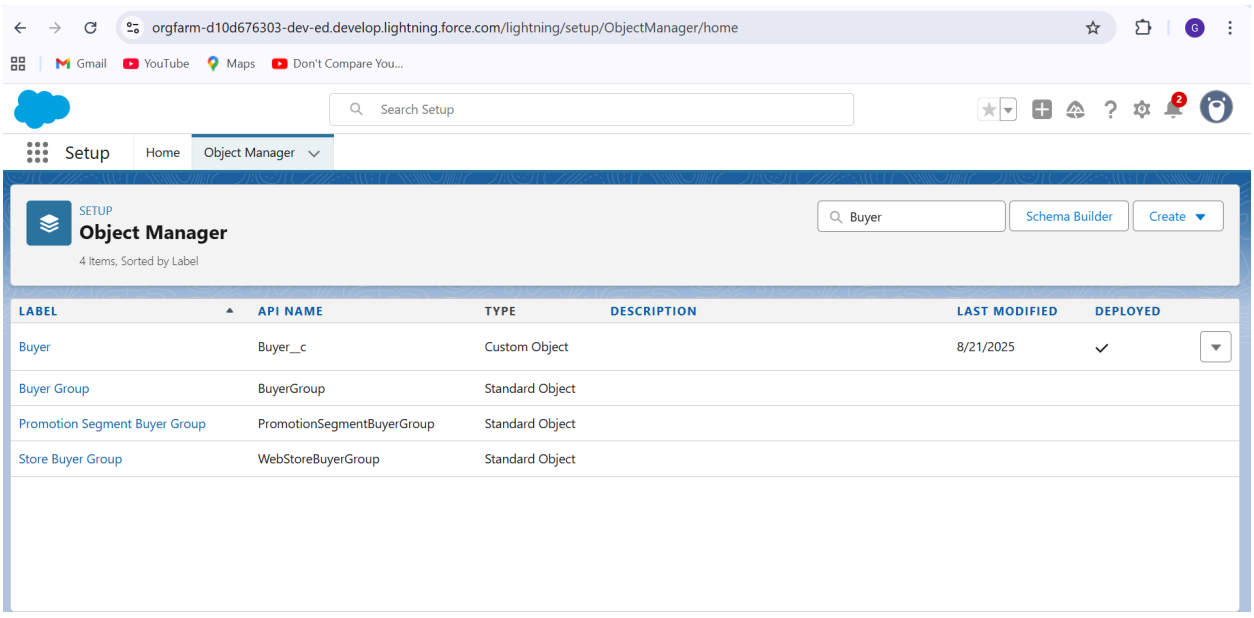
The **Fuel Details Object** manages information about different fuel types, their pricing, and

stock availability. It connects with the **Gas Station** and **Buyer Objects** to manage transactions and inventory.

Steps to Create Fuel Details Object:

1. **Login to Salesforce Developer Account**
    - Sign in at https://login.salesforce.com.
2. **Navigate to Object Manager**
    - Go to **Setup** → **Object Manager**.
3. **Create a New Custom Object**
    - Click **Create** → **Custom Object**.
    - Details:
        - **Label:** Fuel Details
        - **Plural Label:** Fuel Details
        - **Object Name:** Fuel_Details
        - **Record Name:** Fuel Type (Text or Picklist)
    - Enable:
        - Allow Reports
        - Allow Activities
4. **Add Custom Fields for Fuel Details Object**
    - **Fuel ID** (Auto Number)
    - **Fuel Type** (Picklist – LPG, CNG, Diesel, Petrol, etc.)
    - **Price per Unit** (Currency)
    - **Available Quantity** (Number)
    - **Supplier Name** (Lookup to Supplier Object)
    - **Station Name** (Lookup to Gas Station Object)
5. **Set Access & Permissions**
    - Assign Admin and Staff permissions.
6. **Create Tabs and Layouts**
    - Add a tab for Fuel Details.
    - Organize page layout for quick data entry and viewing.
7. **Save and Test**
    - Enter sample fuel data to ensure proper integration with other objects.





Tabs:

What is Tab?

A **Tab in Salesforce** is a navigation element that allows users to access different parts of the Salesforce application. It acts like a shortcut or menu item that opens specific data or functionality.

Types of Tabs in Salesforce:

1. **Custom Object Tabs**
    - Used for custom objects you create (like Supplier, Gas Station, Buyer, Fuel Details).
    - Displays data for each custom object in your application.
2. **Standard Object Tabs**
    - Comes with Salesforce by default (e.g., Accounts, Contacts, Opportunities, Leads).
    - Used for core CRM functionalities.
3. **Web Tabs**
    - Used to display external web content or applications within Salesforce.
    - Example: Linking to an external supplier portal or fuel pricing website.
4. **Visualforce Tabs**
    - Displays a Visualforce page inside Salesforce.
    - Useful for custom pages or dashboards in your gas station CRM.
5. **Lightning Page Tabs**
    - Used in the Lightning Experience to show Lightning App Builder pages.
    - Can include custom layouts, components, and workflows.

Creating a Custom Tab:

A **Custom Tab** in Salesforce allows users to easily access a **custom object** (e.g., Supplier, Gas Station, Buyer, Fuel Details) from the application's navigation bar. After creating custom objects, you must create tabs to make them visible and usable.

Steps to Create a Custom Tab in Salesforce:

1. **Login to Salesforce Developer Account**
    - Go to https://login.salesforce.com and sign in with your credentials.
2. **Navigate to Tabs Setup**
    - Click the **Setup (gear icon)** in the top-right corner.
    - In the **Quick Find box**, type **Tabs** and select **Tabs** under the User Interface section.
3. **Create a New Custom Object Tab**
    - Under the **Custom Object Tabs** section, click **New**.
4. **Select the Object**
    - From the drop-down list, choose the object you want to create a tab for (e.g., **Supplier, Gas Station, Buyer, or Fuel Details**).
5. **Choose a Tab Style**

    - Pick an icon and color that represents the object.
    - Example:
        - Supplier → Briefcase Icon
        - Gas Station → Fuel Pump Icon
        - Buyer → User Icon
        - Fuel Details → Barrel Icon
6. **Set Tab Visibility**
    - Choose the profiles that should have access (e.g., **System Administrator, Staff**).
7. **Add Tab to Apps**
    - Select the apps where the tab should appear (e.g., your **Gas Filling Station CRM Application**).
8. **Save and Verify**
    - Click **Save**, and the tab will now be visible in your application's navigation bar.

Tabs Created for This Project

- **Supplier Tab** – For supplier management.
- **Gas Station Tab** – For station operations.
- **Buyer Tab** – For customer records.
- **Fuel Details Tab** – For fuel pricing and availability

Creating Remaining Tabs:

Once the main tabs (**Supplier, Gas Station, Buyer, and Fuel Details**) have been created, the remaining modules (such as **Inventory, Booking, Invoice**) also require custom tabs to make them easily accessible within the application. Tabs serve as quick navigation points for users to manage all related records efficiently.

Steps to Create the Remaining Tabs

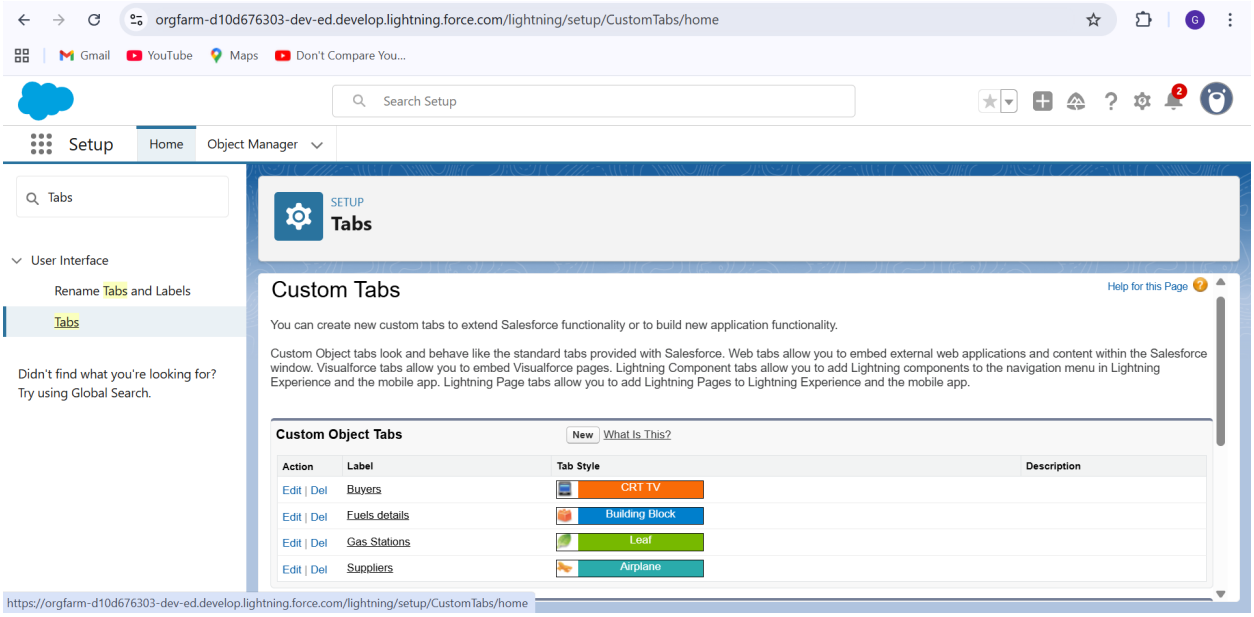
1. **Login to Salesforce Developer Account**
    - Go to https://login.salesforce.com and sign in with your credentials.
2. **Navigate to Tabs Setup**
    - Click the **Setup (gear icon)** in the top-right corner.
    - Type **Tabs** in the Quick Find box and select **Tabs**.
3. **Create Tabs for Remaining Objects**
    - Under **Custom Object Tabs**, click **New**.
    - Choose each remaining object (e.g., **Inventory, Booking, Invoice**) one by one.
4. **Select a Tab Style**
    - Pick a unique icon and color for each tab.

- Example:
    - Inventory → Box Icon
    - Booking → Calendar Icon
    - Invoice → Document Icon

5. **Set Tab Visibility**
    - Define which user profiles can see these tabs (e.g., System Administrator, Staff).
6. **Add Tabs to Application**
    - Choose your application (Gas Filling Station CRM) to include the newly created tabs.
7. **Save and Verify**
    - Save each tab.
    - Go to the navigation bar and confirm their presence.

Example of Remaining Tabs for Your Project

- **Inventory Tab** – Tracks available gas stock.
- **Booking Tab** – Manages refill appointments.
- **Invoice Tab** – Handles billing and payments.



The Lightning App:

The **Lightning App** in Salesforce is a custom application that brings together all your objects, tabs, and components in a single place. It is designed to improve user navigation and provide an efficient workspace.

Create a Lightning App:

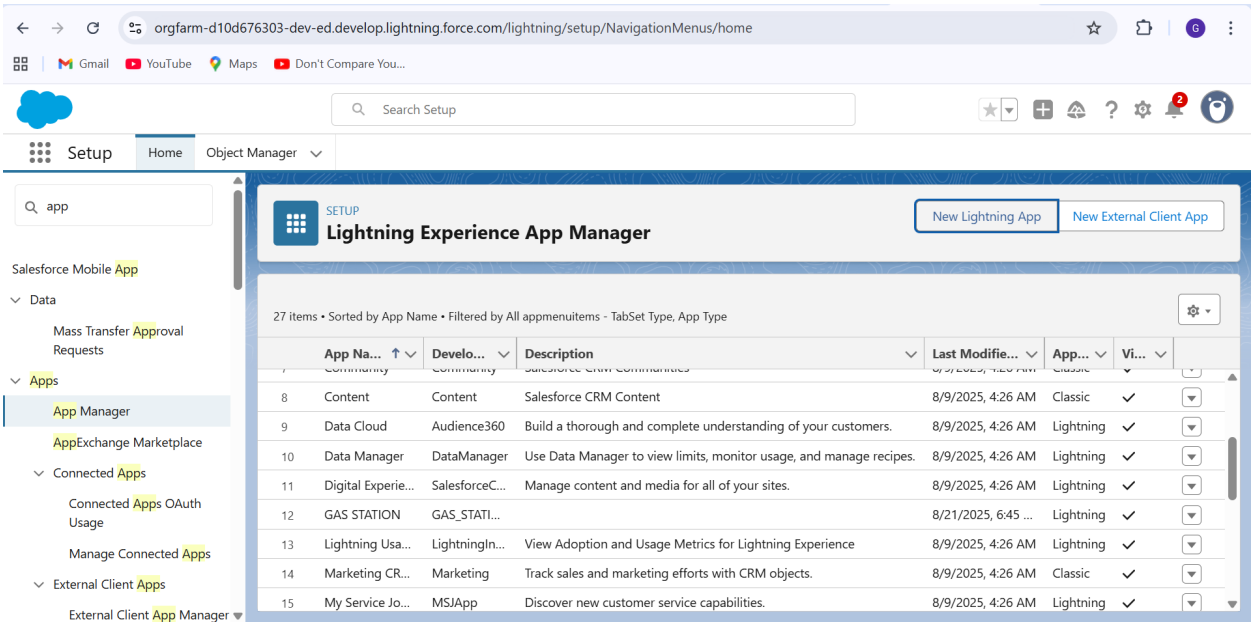
A **Lightning App** in Salesforce is a customizable application that groups all the essential objects, tabs, and components into a unified interface. For the **Gas Filling Station CRM**, this app will help users manage suppliers, gas stations, buyers, fuel details, inventory, bookings, and invoices effectively.

Steps to Create a Lightning App:

1. **Login to Salesforce Developer Account**
   - Visit https://login.salesforce.com and sign in using your developer credentials.
2. **Navigate to App Manager**
   - Click the **Setup (gear icon)** in the top-right corner.
   - In the **Quick Find box**, type **App Manager** and select it.
3. **Click New Lightning App**
   - In the App Manager, click the **New Lightning App** button.
4. **Enter App Details**
   - **App Name**: Gas Filling Station CRM
   - **Developer Name**: Gas_Filling_Station_CRM
   - **Description**: This app streamlines gas filling station operations using Salesforce CRM.
5. **Set Branding (Optional)**
   - Choose a logo and color theme to make your app visually distinct.
6. **Assign User Profiles**
   - Select the profiles that should have access (e.g., System Administrator, Staff).
7. **Add Navigation Items (Tabs)**
   - Select the tabs you created earlier:
     - Supplier
     - Gas Station
     - Buyer
     - Fuel Details
     - Inventory
     - Booking
     - Invoice
8. **Select Navigation Style**
   - Choose **Standard Navigation** for a simple layout or **Console Navigation** for managing multiple records simultaneously.

9. **Review and Finish**
    - Check all configurations and click **Finish**.
10. **Verify the App**

- Go to the **App Launcher (grid icon)** in Salesforce.
- Search for **Gas Filling Station CRM** and open it.



Fields:

Fields in Salesforce are the data points that store specific information within an object. Each object (like Supplier, Gas Station, Buyer, or Fuel Details) contains multiple fields to capture and manage data effectively.

Purpose of Fields:

- To store detailed information about each record.
- To organize and manage data for reporting and analysis.
- To ensure users can input, update, and retrieve information easily.

Types of Fields in Salesforce:

1. **Standard Fields**
    - Predefined by Salesforce (e.g., Name, Created Date, Owner).
    - Cannot be deleted but can be customized.

2. **Custom Fields**
    - Created by the developer or admin as per the business requirements.
    - Examples for your project:
        - Supplier Contact Number
        - Gas Station Location
        - Fuel Type
        - Buyer Payment Mode

Creating a Junction Object:

A **Junction Object** in Salesforce is a custom object used to create a **many-to-many relationship** between two objects. It helps in connecting related data across multiple objects, allowing flexible data management.

In your **Gas Filling Station CRM**, a junction object can be used to link:

- **Buyer ↔ Fuel Details** (A buyer can purchase multiple fuel types, and each fuel type can be sold to multiple buyers.)
- **Gas Station ↔ Supplier** (A gas station may have multiple suppliers, and each supplier may serve multiple gas stations.)

Creating a Master-Detail Relationship:

A **Master-Detail Relationship** in Salesforce is a type of relationship that tightly links two objects, where:

- **Master object** controls the behavior of the **Detail object**.
- When the master record is deleted, its related detail records are also deleted (cascade delete).
- The detail object inherits sharing, security, and ownership settings from the master.

Creating a Number Field in Fuel Details Object:

Purpose of a Number Field:

A **Number field** is used to store numeric data such as quantities, prices, or counts.
In your project, this field can represent:

- **Fuel Quantity (Liters)**

- **Fuel Price per Liter**
- **Stock Available**

Example:

Field Name: **Fuel Quantity**
Data Type: **Number**
Description: Stores the quantity of fuel in liters available in the gas station.

Steps to Create a Number Field:

1. **Login to Salesforce Developer Account**
2. Navigate to **Setup → Object Manager**
3. Select **Fuel Details Object**
4. Go to **Fields & Relationships → New**
5. Select **Number → Next**
6. Fill in:
    - **Field Label:** Fuel Quantity
    - **Field Name:** Fuel_Quantity
    - **Length:** (e.g., 6) – Maximum digits
    - **Decimal Places:** (e.g., 2) – For fractional quantities like 125.50 liters
7. Click **Next** → Set field-level security → Add to Layout → **Save**

Creating the Roll-Up Summary:

A **Roll-Up Summary Field** in Salesforce allows you to calculate values (SUM, COUNT, MIN, MAX) from related detail records in a **Master-Detail Relationship**. It helps in summarizing key data from child records onto the parent object.

Purpose in Gas Filling Station CRM:

In this project, the Roll-Up Summary field is used to:

- Calculate the **total fuel quantity** available at each gas station.
- Count the number of fuel types linked to a gas station.
- Summarize total stock or price based on the **Fuel Details object**.

Example

- **Master Object:** Gas Station
- **Detail Object:** Fuel Details
- **Roll-Up Summary Field:** Total Fuel Quantity (SUM of all fuel quantities)

Steps to Create a Roll-Up Summary

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Object Manager**
3. Select **Gas Station (Master Object)**
4. Navigate to **Fields & Relationships → New**
5. Choose **Roll-Up Summary → Next**
6. Fill the details:
    - **Field Label:** Total Fuel Quantity
    - **Field Name:** Total_Fuel_Quantity
    - **Summarized Object:** Fuel Details
    - **Roll-Up Type:** SUM
    - **Field to Aggregate:** Fuel Quantity
7. Click **Next → Set Field-Level Security → Add to Page Layout → Save**

Supported Functions

- **COUNT:** Count number of fuel records linked.
- **SUM:** Calculate total fuel stock.
- **MIN:** Find the minimum fuel price.
- **MAX:** Find the maximum fuel price.

Creating a Formula Field in Gas Station Object:

A **Formula Field** in Salesforce is a read-only field that automatically calculates and displays values based on other fields or related objects. It updates dynamically whenever the referenced fields change.

Purpose in Gas Filling Station CRM

In this project, a formula field helps in **automatically calculating values for each gas station**. For

example:

- **Total Value of Fuel Stock** = Fuel Quantity × Price per Liter
- **Fuel Availability Status** (Available/Low/Out of Stock)
- **Remaining Capacity** based on total capacity – used capacity

Example

- **Object:** Gas Station
- **Formula Field Name:** Total Fuel Value
- **Return Type:** Currency
- **Formula:**`SUM(Fuel_Quantity__c * Price_per_Liter__c)` (if using Roll-Up Summary fields or child object reference)

Steps to Create a Formula Field

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Object Manager**
3. Select **Gas Station Object**
4. Navigate to **Fields & Relationships → New**
5. Choose **Formula → Next**
6. Enter details:
    - **Field Label:** Total Fuel Value
    - **Field Name:** Total_Fuel_Value
    - **Return Type:** Currency (or Number, depending on requirement)
7. Click **Next → Insert Fields → Select related Roll-Up fields or Fuel Details fields**
8. Write your formula (e.g., `Total_Fuel_Quantity__c * Price_per_Liter__c`)
9. Click **Check Syntax → Next → Set Field-Level Security → Add to Page Layout → Save.**

Creating Cross-Object Formula Field in Buyer Object:

A **Cross-Object Formula Field** allows you to display data from a related object (parent object) in the current object. This is especially useful in **Master-Detail** or **Lookup Relationships** where you want to bring information from one object into another.

Purpose in Gas Filling Station CRM

In this project, a Cross-Object Formula Field helps to:

- Display **Gas Station Name** for each Buyer.

- Show **Total Fuel Quantity or Total Price** from the Fuel Details object.
- Fetch **Supplier Name** directly into Buyer records without duplicating data.

Example

- **Object:** Buyer (Child)
- **Related Object:** Gas Station (Parent)
- **Cross-Object Formula Field Name:** Station Name
- **Return Type:** Text
- **Formula:**`Gas_Station__r.Name`

Steps to Create a Cross-Object Formula Field

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Object Manager**
3. Select **Buyer Object**
4. Navigate to **Fields & Relationships → New**
5. Choose **Formula → Next**
6. Enter details:
    - **Field Label:** Gas Station Name
    - **Field Name:** Gas_Station_Name
    - **Return Type:** Text (or Number/Currency depending on data type)
7. Click **Next → Insert Field → Navigate to Related Object (Gas Station)**
8. Select the field you want to display (e.g., `Gas Station → Name`)
9. Click **Insert → Check Syntax → Next → Set Field-Level Security → Add to Page Layout → Save**

Creating Picklist Field in Buyer Object:

A **Picklist Field** in Salesforce allows users to select a value from a predefined list. It helps maintain data consistency and standardizes the options available to users.

Purpose in Gas Filling Station CRM

In this project, a Picklist field in the **Buyer Object** can be used for:

- **Fuel Type Selection** (e.g., Petrol, Diesel, CNG)
- **Payment Method** (e.g., Cash, Card, UPI, Credit)
- **Buyer Type** (e.g., Regular, Bulk, Corporate)

Example

- **Object:** Buyer
- **Field Label:** Fuel Type
- **Field Name:** Fuel_Type
- **Data Type:** Picklist
- **Values:** Petrol, Diesel, CNG

Steps to Create a Picklist Field

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Object Manager**
3. Select **Buyer Object**
4. Navigate to **Fields & Relationships → New**
5. Choose **Picklist → Next**
6. Enter details:
    - **Field Label:** Fuel Type
    - **Field Name:** Fuel_Type
    - **Values:** Enter each value on a new line (e.g., Petrol, Diesel, CNG)
7. Select **Restrict picklist to the values defined** (optional)
8. Click **Next → Set Field-Level Security → Add to Page Layout → Save**

Creating the Validation Rule:

A **Validation Rule** in Salesforce ensures data accuracy and integrity by preventing users from saving records that don't meet specified criteria. It uses a formula to validate the data entered in one or more fields.

Purpose in Gas Filling Station CRM:

Validation rules are used to:

- Ensure buyers do not enter negative fuel quantities.
- Restrict transactions when fuel quantity is zero.
- Validate that a payment method is selected before purchase.

Example:

- **Object:** Fuel Details
- **Validation Rule Name:** Prevent_Negative_Quantity
- **Error Condition Formula:**
- `Fuel_Quantity__c < 0`
- **Error Message:***"Fuel Quantity cannot be negative."*

Steps to Create a Validation Rule:

1. **Login to Salesforce Developer Account**
2. Go to **Setup** → **Object Manager**
3. Select the object (e.g., **Fuel Details** or **Buyer**)
4. Navigate to **Validation Rules** → **New**
5. Enter:
    - **Rule Name:** Prevent_Negative_Quantity
    - **Error Condition Formula:**`Fuel_Quantity__c < 0`
    - **Error Message:** "Fuel Quantity cannot be negative."
6. Choose where to display the error (Top of Page or Field Level)
7. Click **Save**

Creating Remaining Fields in Objects:

After creating essential fields like Picklist, Formula, and Roll-up Summary, additional fields are required to store specific data for each object in the CRM system. These **remaining fields** ensure all business-related information is captured for smooth operation of the Gas Filling Station.

Purpose

- To store additional data such as **contact information, pricing details, location, and fuel type**.
- To make each object (Supplier, Gas Station, Buyer, Fuel Details) fully functional.

Example of Remaining Fields

1. Supplier Object

- **Phone (Text)** – Contact number of supplier.
- **Email (Email)** – Supplier's official email ID.

- **Address (Text Area)** – Supplier's address.

2. Gas Station Object

- **Location (Text)** – Gas station location.
- **Capacity (Number)** – Total storage capacity of the station.
- **Operating Hours (Text)** – Timings of operation.

3. Buyer Object

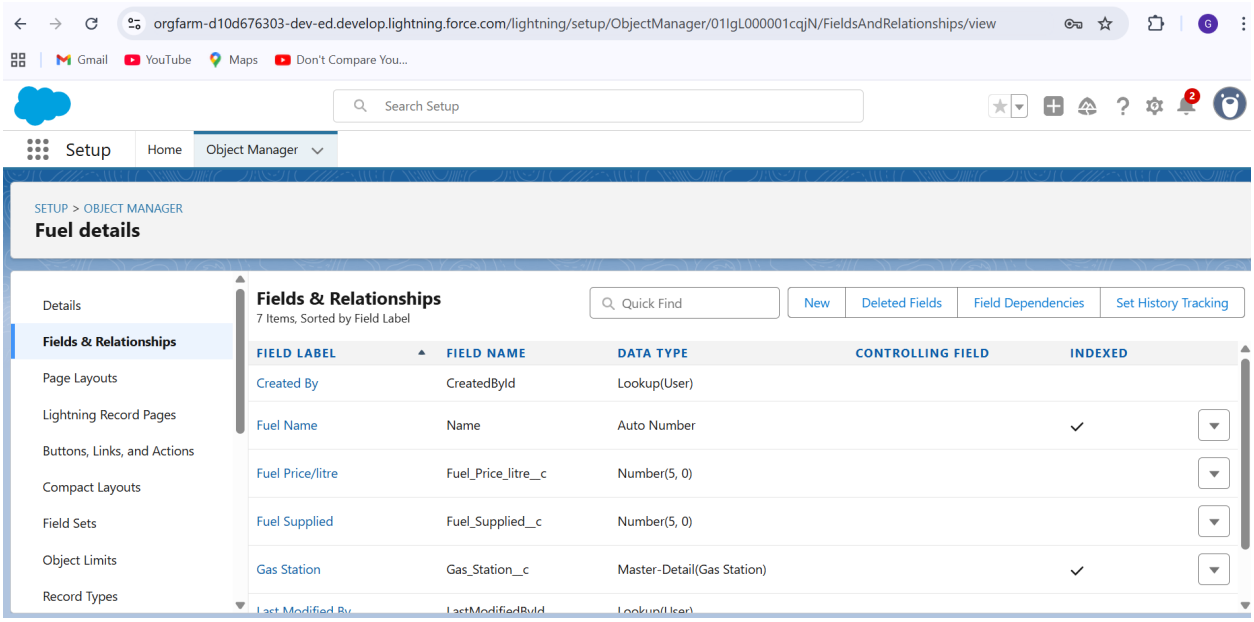
- **Contact Number (Phone)** – Buyer's mobile number.
- **Buyer Type (Picklist)** – Regular, Corporate, Bulk.
- **Payment Method (Picklist)** – Cash, Card, UPI, Credit.

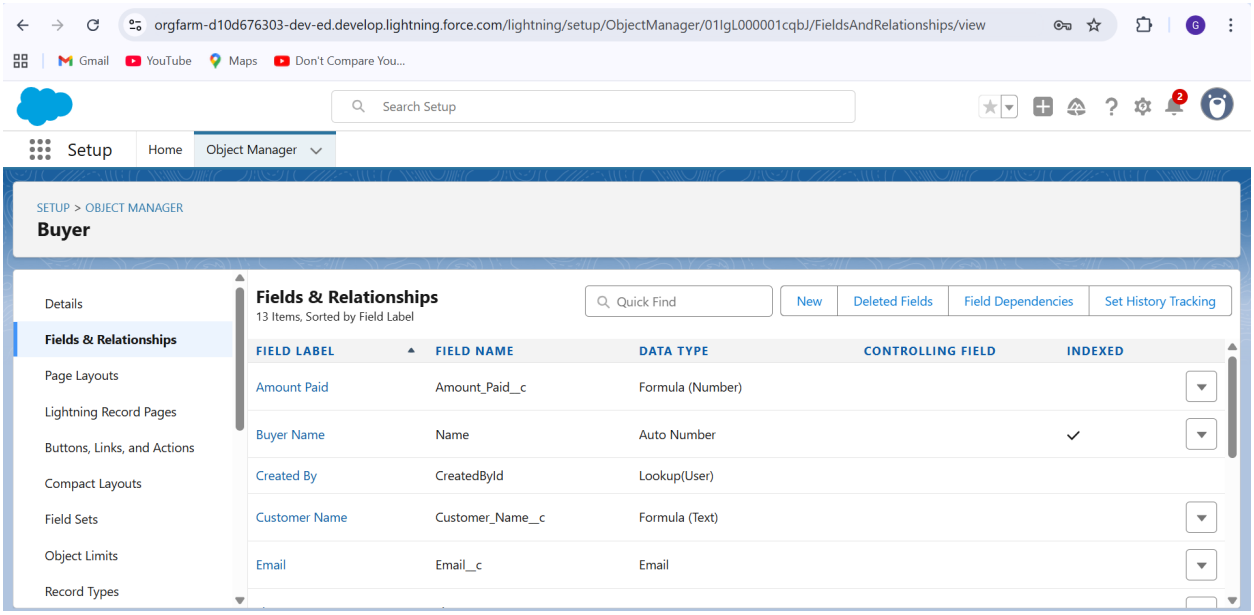
4. Fuel Details Object

- **Fuel Type (Picklist)** – Petrol, Diesel, CNG.
- **Fuel Price (Currency)** – Price per liter.
- **Quantity Purchased (Number)** – Liters purchased by the buyer.

Steps to Create Remaining Fields

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Object Manager**
3. Select the desired object (e.g., Buyer, Fuel Details, Gas Station, Supplier)
4. Navigate to **Fields & Relationships → New**
5. Select the appropriate data type (Text, Number, Picklist, Currency, Email, Phone, etc.)
6. Enter details:
    - **Field Label**
    - **Field Name**
    - **Field Type**
    - **Help Text (if needed)**
7. Set field-level security and add to page layout.
8. Click **Save**.





Page Layouts:

A **Page Layout** in Salesforce controls the organization and visibility of fields, sections, buttons, links, and related lists on an object's record page. It determines **what users can see and interact with** when they view or edit a record.

Creating the Page Layout:

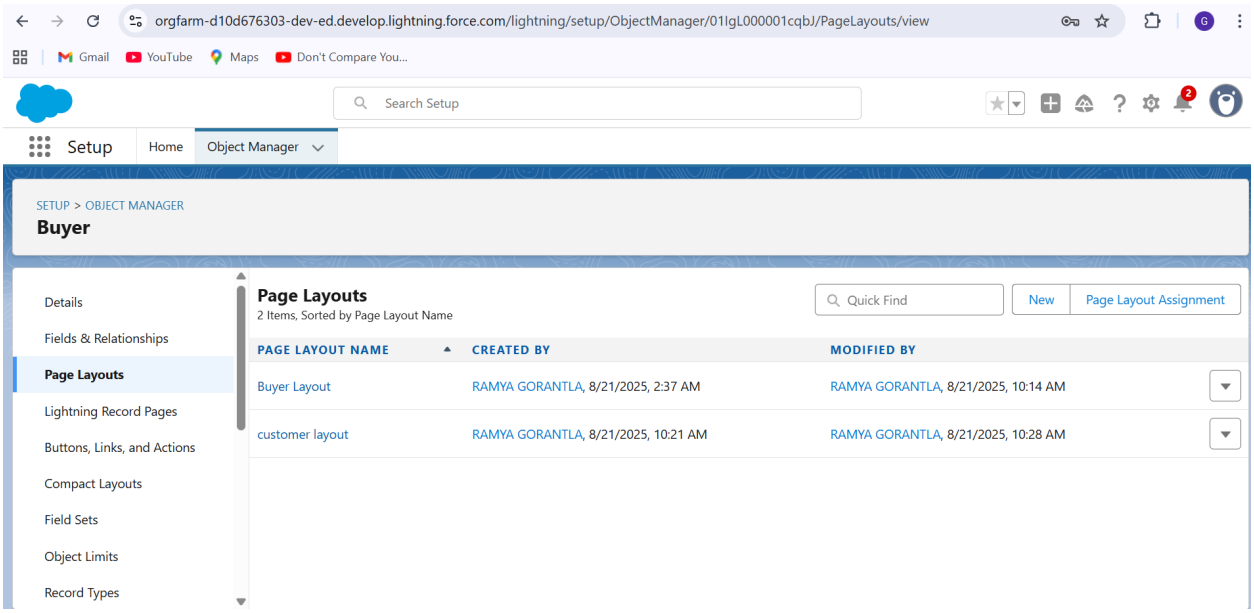
Page Layouts in Salesforce define how fields, sections, related lists, and buttons are organized and displayed on an object's record page. They help tailor the user interface based on your project's needs and user roles.

Purpose

- To organize fields logically for **better user experience**.
- To display only **relevant information** for each object (Supplier, Gas Station, Buyer, Fuel Details).
- To simplify **data entry and record viewing**.

Steps to Create a Page Layout

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Object Manager**
3. Select the required object (e.g., Gas Station, Supplier, Buyer, Fuel Details)
4. Click **Page Layouts → New** (or clone an existing layout)
5. Provide a **Layout Name** (e.g., "Gas Station Layout")
6. Drag and drop:
    - **Fields** (e.g., Station Name, Location, Capacity)
    - **Sections** (e.g., General Info, Contact Details, Transaction Details)
    - **Buttons & Quick Actions**
7. Add **Related Lists** (e.g., Fuel Purchases for Buyer Object)
8. Assign the layout to specific profiles if needed.
9. Click **Save.**



Profiles:

In Salesforce, a **Profile** is a set of settings and permissions that defines what a user can do within the Salesforce organization. Profiles determine **access to objects, fields, tabs, applications, and features**.

Types of Profiles:

1. **Standard Profiles** (provided by Salesforce)
    - Examples: Standard User, System Administrator, Read Only, Marketing User.
    - Cannot be deleted but can be cloned and customized.
2. **Custom Profiles** (created by developers/admins)
    - Tailored for specific business needs.
    - Can define object-level, field-level, and tab-level permissions.

Manager Profile:

The **Manager Profile** is a custom profile designed for managers overseeing gas station operations. It provides them with the required permissions to manage **Suppliers, Buyers, Gas Stations, and Fuel Details** efficiently while restricting access to administrative functions.

Key Permissions:

- **Objects Access**:

    - **Gas Station**: Full access (create, read, edit, delete).
    - **Supplier**: Read and edit access.
    - **Buyer**: Full access.
    - **Fuel Details**: Full access.
- **Tabs Access**:
    - Visible: Gas Station, Suppliers, Buyers, Fuel Details, Reports.
    - Hidden: Setup/Developer tools.
- **Record Types & Page Layouts**:
    - Assigned to all relevant layouts for operational data management.
- **Reports & Dashboards**:
    - Can create and view analytics reports for supply chain and sales tracking.

Steps to Create Manager Profile:

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Profiles**
3. Click **New Profile → Clone Standard User** (or any suitable base profile)
4. Enter the Profile Name: **Manager Profile**
5. Configure:
    - **Object Permissions** (Gas Station, Suppliers, Buyers, Fuel Details)
    - **Tab Settings** (set to Default On where needed)
    - **Field-Level Security** (restrict sensitive fields if required)
6. Save the profile and assign it to the **Manager role users**.

Sales Executive Profile:

The **Sales Executive Profile** is a custom profile designed for sales representatives responsible for handling **buyer interactions, sales tracking, and fuel orders**. This profile ensures they have access only to the necessary data and features required to perform their duties effectively.

Key Permissions:

- **Objects Access**:
    - **Buyer**: Full access (create, read, edit, delete).
    - **Fuel Details**: Read and create access (to track sales transactions).
    - **Gas Station**: Read-only access (to check availability).
    - **Supplier**: No access (or read-only, if needed).
- **Tabs Access**:
    - Visible: Buyers, Fuel Details, Sales Reports.
    - Hidden: Suppliers, System Setup.
- **Record Types & Page Layouts**:

    - Assigned layouts optimized for sales activities.
- **Reports & Dashboards**:
    - Can view and create sales performance reports.

Steps to Create Sales Executive Profile:

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Profiles**
3. Click **New Profile → Clone Standard User** (or any suitable profile).
4. Enter the Profile Name: **Sales Executive Profile**
5. Configure:
    - **Object Permissions** (Buyers – Full, Fuel Details – Create/Read, Gas Station – Read Only).
    - **Tab Settings** (set only necessary ones to Default On).
    - **Field-Level Security** (restrict supplier-related fields).
6. Save and assign it to the **Sales Executive users**.

Sales Person Profile:

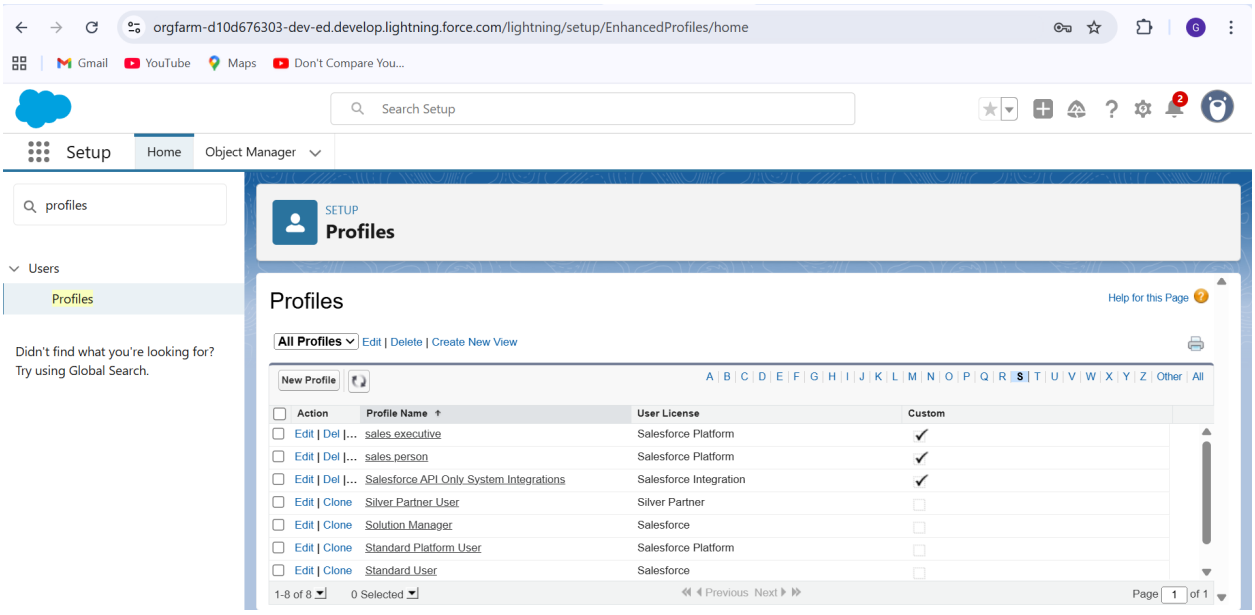
The **Sales Person Profile** is designed for employees who are primarily responsible for **interacting with buyers, processing orders, and ensuring smooth sales operations**. This profile limits access to only essential sales-related data, ensuring security and efficient workflow management.

Key Permission:

- **Objects Access:**
    - **Buyer**: Full access (create, read, edit, delete).
    - **Fuel Details**: Read and create access.
    - **Gas Station**: Read-only access (to check stock/availability).
    - **Supplier**: No access.
- **Tabs Access**:
    - Visible: Buyers, Fuel Details.
    - Hidden: Suppliers, Admin Setup, Reports (optional).
- **Record Types & Page Layouts**:
    - Configured for quick buyer and fuel sales operations.
- **Reports & Dashboards**:
    - View only (if allowed by admin).

Steps to Create Sales Person Profile:

1. **Login to Salesforce Developer Account**
2. Navigate to **Setup → Profiles**
3. Click **New Profile → Clone Standard User** (or a similar profile).
4. Enter Profile Name: **Sales Person Profile**
5. Configure:
    - **Object Permissions** (Buyers – Full, Fuel Details – Read/Create, Gas Station – Read Only).
    - **Tab Settings** (enable Buyers & Fuel Details as Default On).
    - **Field-Level Security** (restrict supplier fields).
6. Save and assign to **Sales Person users**.



Role & Role Hierarchy:

What is a Role?

A **Role** in Salesforce defines a user's position within the organization's hierarchy. It controls the **level of visibility and access to records** a user has, based on their job responsibilities.

Unlike profiles (which control object and field-level permissions), **roles determine record-level access**.

What is Role Hierarchy?

Role Hierarchy is a structure that reflects the organization's chain of command.

- Higher roles in the hierarchy inherit **all records owned by the roles below them**.
- It ensures that managers and supervisors have visibility into their subordinates' data.

Example Role Hierarchy Structure:

```
Gas Station Admin (Top Level)
│
├── Station Manager
│     ├── Sales Executive
│     │     └── Sales Person
│
└── Supplier Manager
```

Key Features:

- Managers can view and manage data of their **subordinates**.
- Sales Executives can view **Sales Person data**, but not vice versa.
- Each user has access to **their own records** and records shared by hierarchy.

Steps to Create Role Hierarchy in Salesforce:

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Roles → Set Up Roles**
3. Click **Add Role** under the desired parent role.
4. Enter **Role Name (e.g., Sales Person, Manager)**
5. Define the role hierarchy by selecting a **Parent Role**.
6. Save and assign users to roles accordingly.

Creating Manager Role:

The **Manager Role** is used to assign record-level visibility and access to users who manage gas station operations and oversee sales teams. This role allows managers to view, edit, and report on records of their subordinates while maintaining proper data security.

Steps to Create Manager Role:

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Users → Roles → Set Up Roles**

3. Click **Add Role** under the appropriate parent role (e.g., Gas Station Admin)
4. Enter details:
    - **Role Name:** Manager
    - **Role Label:** Manager
    - **Parent Role:** Gas Station Admin (or top-level as required)
    - **Description:** Oversees station operations, sales team, and fuel management
5. Click **Save**

Assign Users to Manager Role:

1. Go to **Setup** → **Users** → **Users**
2. Select the user who will be a manager
3. Click **Edit** → **Role** → Select **Manager**
4. Save the changes

Creating Other Roles:

After creating the **Manager Role**, you can create additional roles to reflect the operational hierarchy of the Gas Filling Station CRM. These roles define **record-level access** and maintain the visibility structure among users.

Other Roles in Gas Filling Station CRM:

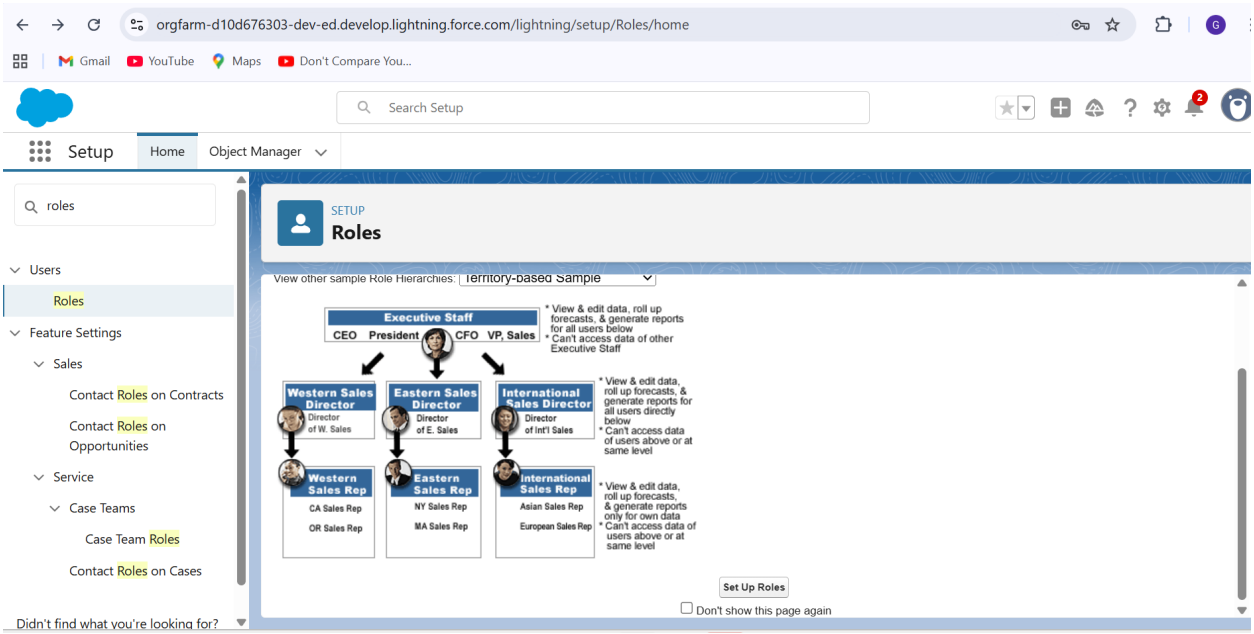
1. **Sales Executive Role**
    - **Purpose:** Oversees the sales team and manages sales transactions.
    - **Parent Role:** Manager
    - **Permissions:** Can view and manage records of assigned Sales Persons.
2. **Sales Person Role**
    - **Purpose:** Handles day-to-day sales activities and buyer interactions.
    - **Parent Role:** Sales Executive
    - **Permissions:** Can access and manage only their own records.
3. **Supplier Manager Role**
    - **Purpose:** Manages supplier-related data and deliveries.
    - **Parent Role:** Gas Station Admin (or Manager)
    - **Permissions:** Can view and edit supplier details and related transactions.
4. **Gas Station Admin Role** (Top-level)
    - **Purpose:** Has access to all records and can oversee the entire operation.
    - **Parent Role:** None (Top of the hierarchy)

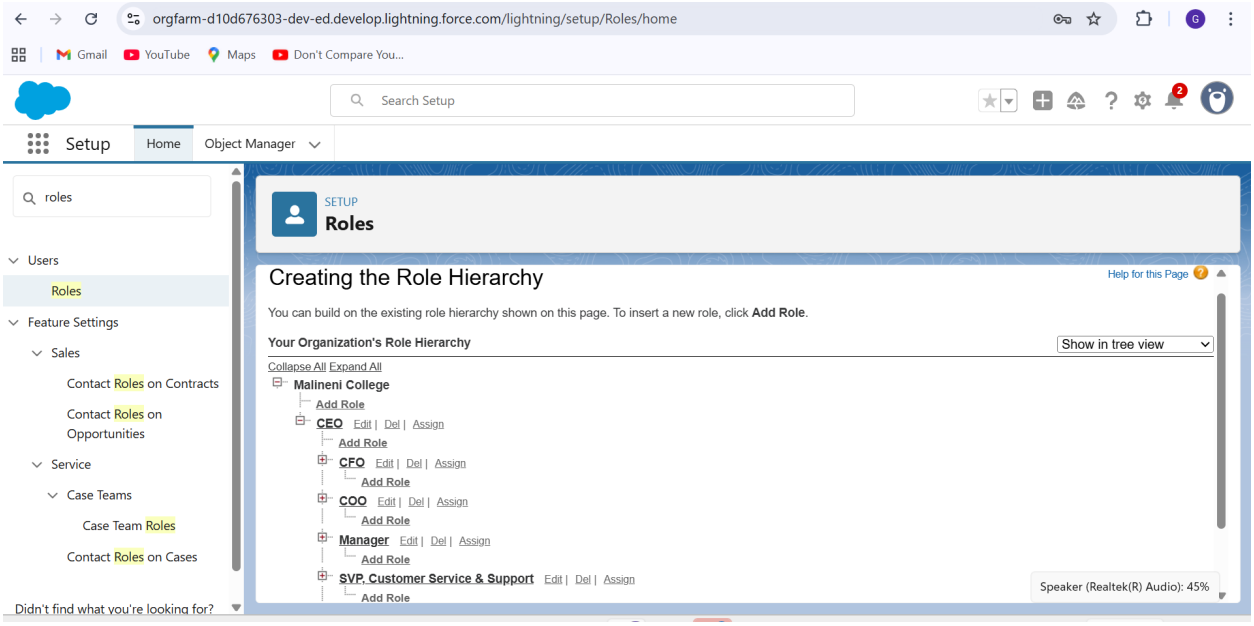
Steps to Create Additional Roles:

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Users → Roles → Set Up Roles**
3. Click **Add Role** under the desired parent role (e.g., Manager)
4. Enter details:
    - **Role Name**
    - **Role Label**
    - **Parent Role** (to define hierarchy)
    - **Description** (optional, to clarify role responsibility)
5. Click **Save**

Assign Users to Roles:

1. Go to **Setup → Users → Users**
2. Select the user → **Edit**
3. Choose the appropriate **Role** (Sales Executive, Sales Person, Supplier Manager)
4. Save the changes,





Users:

In Salesforce, a **User** represents an individual who can log in and access the system based on their assigned **Profile** and **Role**. Users are the core entities that interact with objects, records, and processes in the CRM.

Creating a User:

A **User** in Salesforce is an individual who can log in to the system and perform actions according to their assigned **Profile** and **Role**. Creating users allows staff to access the CRM and manage gas station operations efficiently.

Steps to Create a User:

1. **Login to Salesforce Developer Account**
2. Navigate to **Setup → Users → Users**
3. Click **New User**
4. Fill in the required details:
    - **First Name & Last Name**
    - **Email Address**
    - **Username** (must be unique)

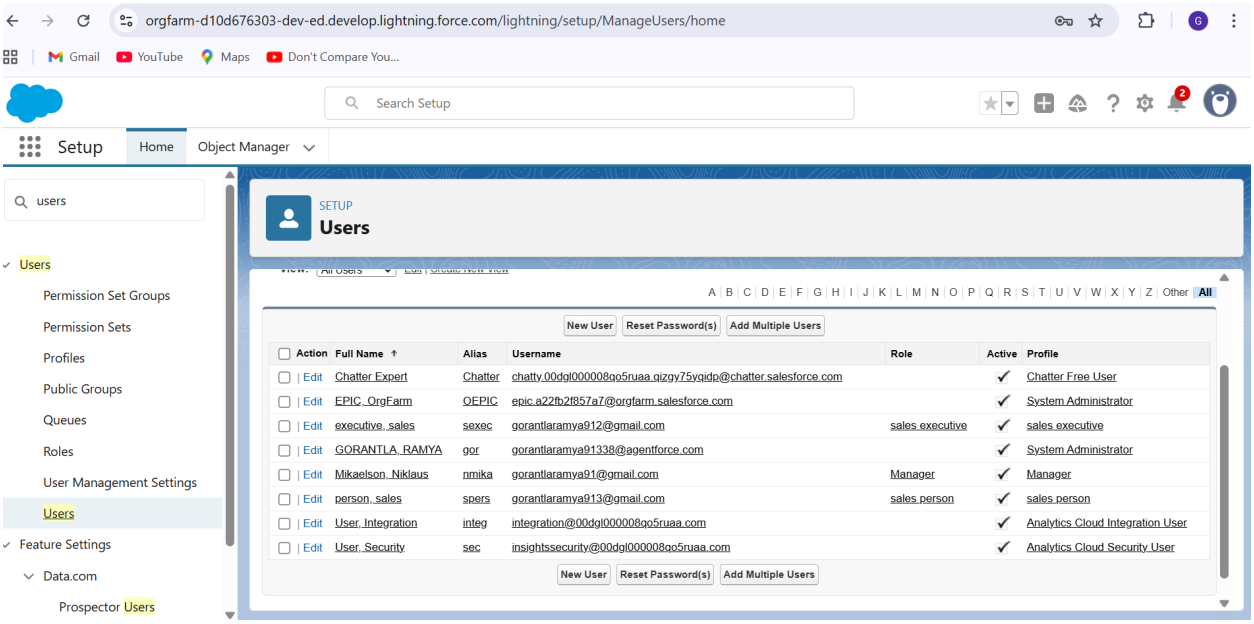
    - **Alias**
    - **Role** (e.g., Manager, Sales Executive)
    - **Profile** (e.g., Manager Profile, Sales Person Profile)
    - **Locale, Time Zone, Language** (optional)
5. Click **Save**
6. Salesforce sends a **welcome email** to the user with login credentials.

Creating Another Users:

After creating the first user, you can add more users to represent different roles in your Gas Filling Station CRM. Each user is assigned a **Profile** and **Role** to control access and permissions within Salesforce.

Steps to Create Additional Users:

1. **Login to Salesforce Developer Account**
2. Navigate to **Setup → Users → Users**
3. Click **New User**
4. Enter user details:
    - **First Name & Last Name**
    - **Email Address**
    - **Username** (must be unique)
    - **Alias**
    - **Role** (Manager, Sales Executive, Sales Person, Supplier Manager)
    - **Profile** (e.g., Manager Profile, Sales Executive Profile, Sales Person Profile)
    - **Locale, Time Zone, Language** (optional).
5. Click **Save**
6. Salesforce sends a **welcome email** with login credentials to the user.



Permission Sets:

A **Permission Set** in Salesforce is a collection of settings and permissions that **grant additional access** to users without changing their Profile. Unlike Profiles, Permission Sets can be **assigned to multiple users** and **stacked** to provide flexible access control.

Creating a Permission Set:

A **Permission Set** in Salesforce allows you to **grant additional permissions** to users without modifying their Profile. This is useful when a user needs **extra access for specific tasks** such as managing fuel inventory or viewing reports.

Steps to Create a Permission Set:

1. **Login to Salesforce Developer Account**
2. Go to **Setup** → **Permission Sets** → **New**
3. Enter details:
    - **Label:** Fuel Manager
    - **API Name:** Fuel_Manager
    - **License:** (Optional, can leave as “--None--”)
4. Click **Save**
5. Configure permissions:

    - **Object Settings** → Select object → Edit permissions (Read, Create, Edit, Delete)
    - **App Settings** → Assign apps/tabs visibility
    - **System Permissions** → Enable required system-level access
6. Click **Save**
7. Assign the Permission Set to users:
    - **Manage Assignments → Add Assignments → Select Users → Assign.**

Configuring the Permission Set:

1. Click the created permission set (e.g., **Fuel Manager**)
2. Configure **Object Settings**:
    - Select an object (e.g., Fuel Details)
    - Click **Edit** → Grant **Read, Create, Edit, Delete** as needed
3. Configure **App Settings**:
    - Enable necessary **apps and tabs** visibility
4. Configure **System Permissions**:
    - Enable system-level permissions (optional, e.g., “Run Reports”)
5. Click **Save**

Assigning Users to the Permission Set:

1. In the Permission Set, click **Manage Assignments**
2. Click **Add Assignments**
3. Select the users to assign the permission set
4. Click **Assign → Done.**

Setup for OWD (Organization-Wide Defaults):

Organization-Wide Defaults (OWD) in Salesforce define the **baseline level of access** to records for all users in the organization. OWD controls **record-level security**, ensuring that users can only see or edit records according to the default sharing settings.

Steps to Set Up OWD

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Security → Sharing Settings**
3. Scroll to **Organization-Wide Defaults**
4. Click **Edit**
5. Set **Default Internal Access** for each object (Public Read/Write, Public Read Only, or Private)

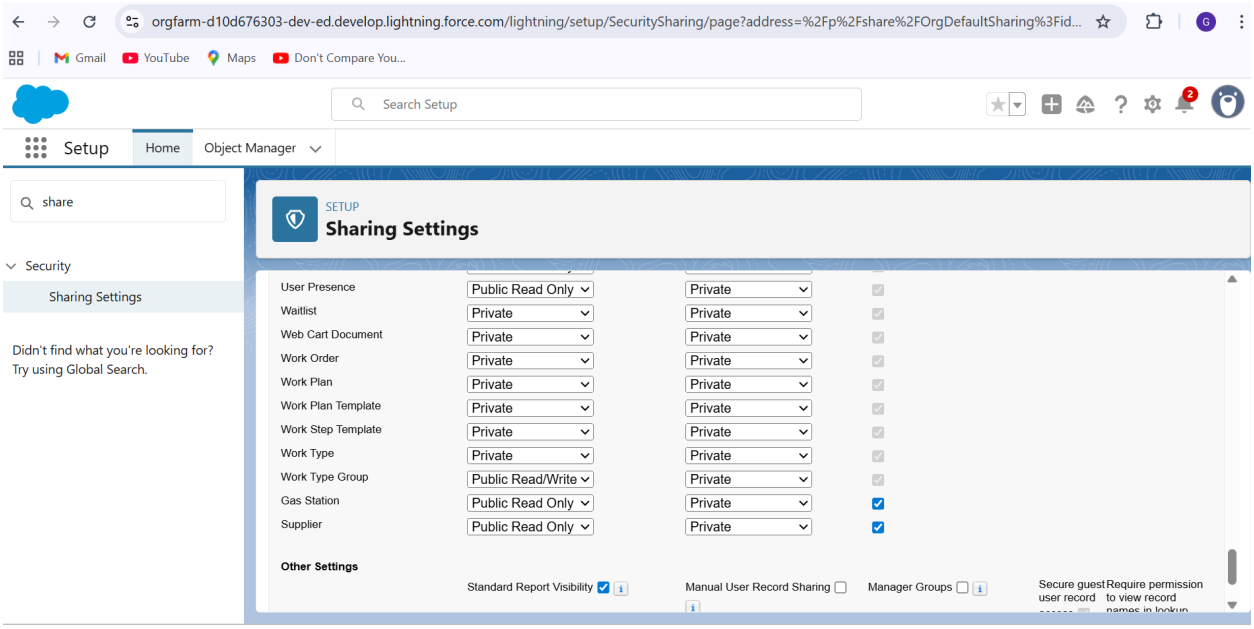
6. Save the settings

Creating OWD Setting:

Organization-Wide Defaults (OWD) in Salesforce define the **default level of access** that all users have to records in each object. OWD settings are the **baseline security** for controlling record visibility across the organization.

Steps to Create OWD Setting

1. **Login to Salesforce Developer Account**
2. Go to **Setup → Security → Sharing Settings**
3. Scroll down to **Organization-Wide Defaults**
4. Click **Edit**
5. Set default access levels for each object:
    - **Gas Station**: Public Read/Write (or Private if restricted)
    - **Buyer**: Private
    - **Supplier**: Private
    - **Fuel Details**: Private
6. Set **Default External Access** if needed (for external users/communities)
7. Click **Save.**



User Adoption:

User Adoption refers to the process of encouraging and enabling users to **effectively use Salesforce CRM** to perform their daily tasks. High user adoption ensures that the CRM system delivers its intended benefits, such as better data management, streamlined operations, and informed decision-making.

Create A Record:

A **record** in Salesforce represents a single instance of an object (e.g., a Buyer, Supplier, Gas Station, or Fuel Details). Creating records allows users to **store and manage data** within the CRM.

Steps to Create a Record:

1. **Login to Salesforce Developer Account**
2. Navigate to the **App** containing your object (e.g., Gas Station App)
3. Click on the **tab** for the object you want to create a record for (e.g., Buyer, Supplier, Fuel Details, Gas Station)
4. Click **New**
5. Fill in the required fields:
    - Example for **Buyer**: Name, Contact Number, Buyer Type, Payment Method
    - Example for **Fuel Details**: Fuel Type, Quantity, Price, Linked Buyer, Date of Purchase
    - Example for **Supplier**: Name, Contact Number, Email, Address
    - Example for **Gas Station**: Name, Location, Capacity, Operating Hours
6. Click **Save**

Example Record Creation:

Creating a Buyer Record

- **Name:** Rajesh Kumar
- **Contact Number:** 9876543210
- **Buyer Type:** Regular
- **Payment Method:** UPI

Creating a Fuel Details Record

- **Fuel Type:** Petrol

- **Quantity Purchased:** 50 Liters
- **Fuel Price:** ₹100 per Liter
- **Linked Buyer:** Rajesh Kumar

View A Record:

In Salesforce, viewing a record allows users to **see the detailed information** stored for a specific object instance, such as a Buyer, Supplier, Gas Station, or Fuel Details.

Steps to View a Record:

1. **Login to Salesforce Developer Account**
2. Navigate to the **App** containing your object (e.g., Gas Station App)
3. Click on the **tab** for the object you want to view (e.g., Buyer, Supplier, Fuel Details, Gas Station)
4. Browse the **list view** to find the record
5. Click on the **record name** to open its **record detail page**
6. Review all the information including:
    - **Fields** (e.g., Name, Contact, Quantity, Fuel Type)
    - **Related Lists** (e.g., Linked Buyers, Fuel Purchases)
    - **Chatter / Activities** (if enabled)**Example: Viewing a Buyer Record:**
- Navigate to **Buyer Tab**
- Click on **Rajesh Kumar**
- View details:
    - Contact Number: 9876543210
    - Buyer Type: Regular
    - Payment Method: UPI
    - Related Fuel Details: 50 Liters Petrol purchased on 20-Aug-2025

Delete A Record:

In Salesforce, deleting a record permanently removes it from the system (unless it is restored from the Recycle Bin). Deleting records is usually restricted to users with **appropriate permissions**, such as Managers or Admins.

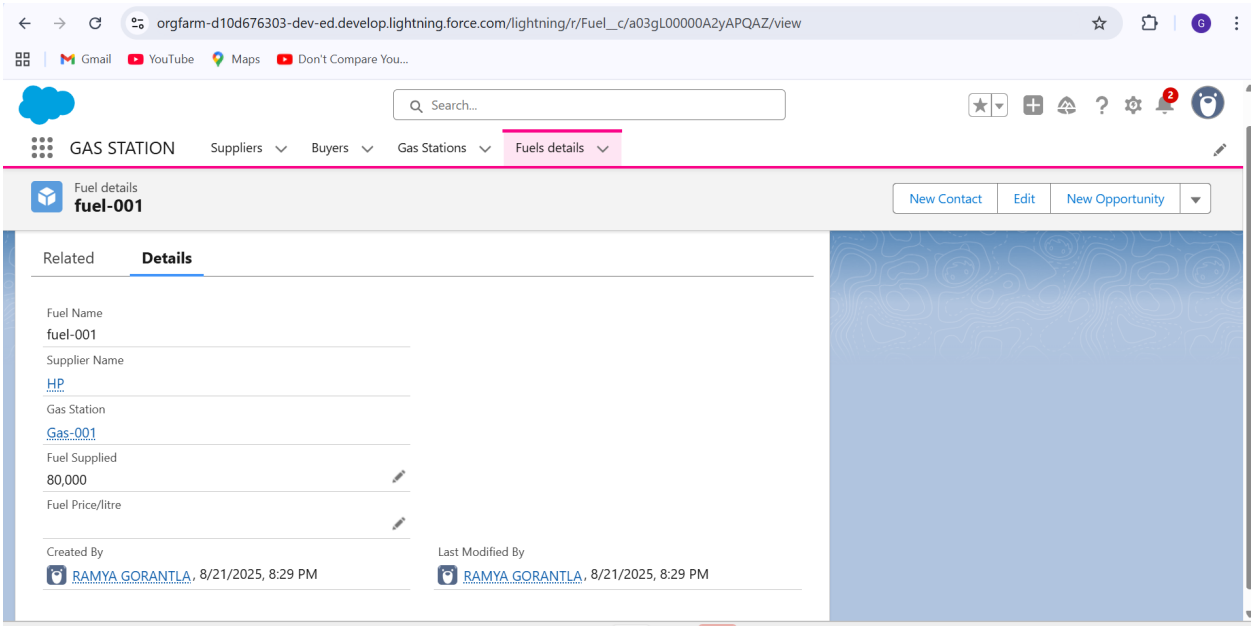
Steps to Delete a Record:

1. **Login to Salesforce Developer Account**

2. Navigate to the **App** containing your object (e.g., Gas Station App)
3. Click on the **tab** of the object (e.g., Buyer, Supplier, Fuel Details, Gas Station)
4. Use the **list view** or **search** to locate the record
5. Click on the **record name** to open the **record detail page**
6. Click **Delete** (usually available in the record detail page toolbar or dropdown)
7. Confirm the deletion when prompted

Example: Deleting a Buyer Record

- Navigate to **Buyer Tab**
- Select **Rajesh Kumar**
- Click **Delete** → Confirm
- The record is moved to **Recycle Bin**, where it can be restored if needed.



Reports:

In Salesforce, a Report is a way to summarize, analyze, and present data stored in objects like Gas Stations, Buyers, Suppliers, and Fuel Details. Reports help users make data-driven decisions and track key business .

Types of Reports:

1. **Tabular Reports**
    - Simple list of records with selected fields.
    - Example: List of all buyers or suppliers.
2. **Summary Reports**
    - Groups records by specific fields and shows subtotals.
    - Example: Fuel sold by Fuel Type or Buyer.
3. **Matrix Reports**
    - Groups records by rows and columns for cross-tab analysis.
    - Example: Fuel sales by Gas Station vs Fuel Type.
4. **Joined Reports**
    - Combines multiple report types into one view.
    - Example: Comparing fuel sales and supplier deliveries in one report.

metrics.

Creating a Report Folder:

A **Report Folder** in Salesforce is used to **organize reports** so that users can easily access, share, and manage them. Folder-level permissions control **who can view or edit reports** stored in that folder.

Steps to Create a Report Folder:

1. **Login to Salesforce Developer Account**
2. Go to **Reports** → **All Folders** → **New Folder**
3. Enter details:
    - **Folder Name:** e.g., Gas Station Reports
    - **Description:** Optional, e.g., “Reports for Gas Station operations”
4. Set **Access Permissions**:
    - **Viewer Access:** Users who can only view reports
    - **Editor Access:** Users who can create, edit, and delete reports
    - **Manager Access:** Users who can share or change folder access
5. Click **Save**

Sharing a Report Folder:

Sharing a **Report Folder** in Salesforce allows you to **control who can view or edit reports** stored in that folder. This ensures that sensitive data is only accessible to the right users while enabling collaboration for teams.

Steps to Share a Report Folder:

1. **Login to Salesforce Developer Account**
2. Navigate to **Reports → All Folders**
3. Find the folder you want to share (e.g., **Gas Station Reports**)
4. Click the **Share** button next to the folder
5. In the **Sharing Settings** window:
    - Select **Users, Roles, or Public Groups** you want to share with
    - Choose **Access Level**:
        - **Viewer** → Can view reports only
        - **Editor** → Can view, create, and edit reports
        - **Manager** → Can view, edit, and manage sharing
6. Click **Share** and then **Done.**

Creating a Report:

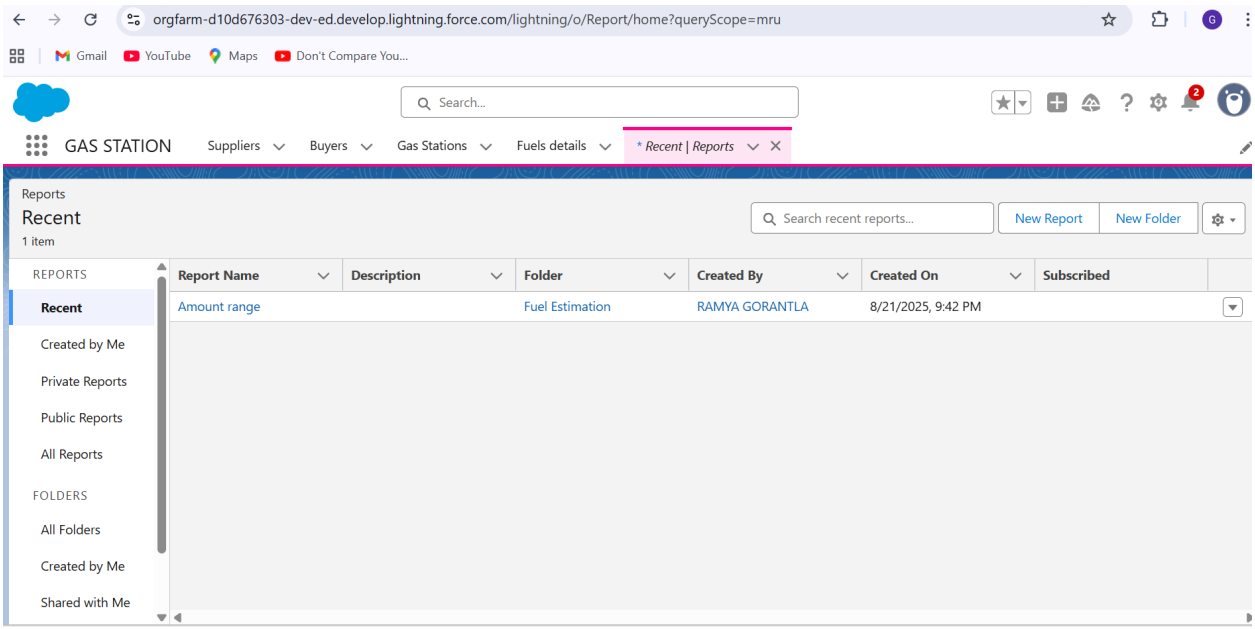
A **Report** in Salesforce is a way to **analyze, summarize, and present data** from objects like Gas Stations, Buyers, Suppliers, or Fuel Details. Reports allow users to make **data-driven decisions** and monitor operations effectively.

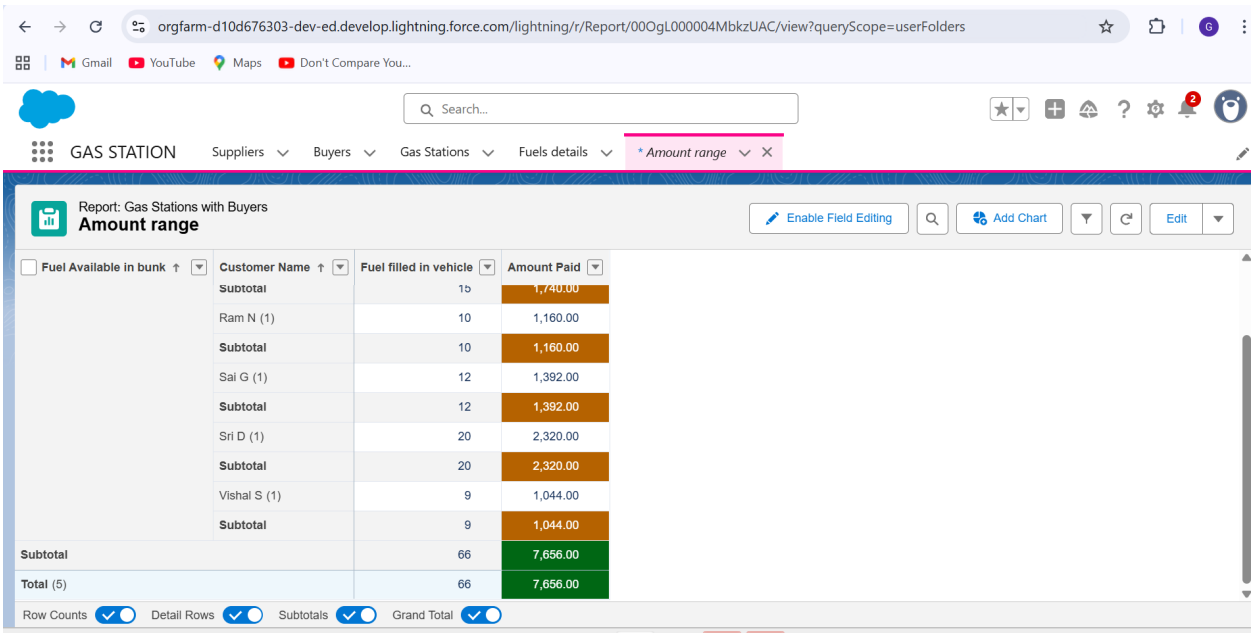
Steps to Create a Report:

1. **Login to Salesforce Developer Account**
2. Navigate to **Reports → New Report**
3. Select the **Report Type** corresponding to the object:
    - Examples: Buyer, Fuel Details, Supplier, Gas Station
4. Click **Continue**
5. Apply **Filters** to narrow down the data:
    - Example: Fuel Type = Petrol, Date = Last Month
6. Select **Columns** to display the relevant fields:
    - Example: Buyer Name, Fuel Quantity, Fuel Price
7. Apply **Grouping** (if needed for Summary or Matrix Reports)
8. Click **Save & Run**
9. Provide a **Report Name**, **Description**, and select the **Report Folder.**

Example: Creating a Fuel Sales Report:

- **Report Type:** Fuel Details
- **Filters:** Date = Last Month, Fuel Type = Diesel
- **Columns:** Buyer Name, Quantity Purchased, Total Price
- **Group By:** Gas Station
- **Folder:** Fuel Details Reports
- **Report Name:** “Diesel Sales by Station – Last Month”.





Dashboards:

A **Dashboard** in Salesforce is a **visual representation of data** from multiple reports. Dashboards help users **quickly understand key metrics** and make informed decisions by displaying charts, graphs, and tables in one place.

Components of a Dashboard:

1. **Charts** – Pie, Bar, Line, or Donut charts to represent data visually.
2. **Tables** – Display detailed record information in tabular form.
3. **Metrics** – Show key performance indicators (KPIs) as single values.
4. **Gauges** – Track progress against a target value.

Create Dashboard Folder:

A **Dashboard Folder** in Salesforce is used to **organize dashboards** so that users can easily find, access, and manage them. Folder-level permissions control **who can view or edit dashboards** stored in that folder.

Steps to Create a Dashboard Folder:

1. **Login to Salesforce Developer Account**
2. Go to **Dashboards → All Folders → New Folder**

3. Enter details:
    - **Folder Name:** e.g., Gas Station Dashboards
    - **Description:** Optional, e.g., “Dashboards for monitoring gas station operations”
4. Set **Access Permissions**:
    - **Viewer Access:** Users who can only view dashboards
    - **Editor Access:** Users who can create, edit, and delete dashboards
    - **Manager Access:** Users who can share or change folder access
5. Click **Save.**

Create Dashboard:

A Dashboard in Salesforce is a visual representation of data from one or more reports. Dashboards help users quickly analyze performance metrics and make informed operational decisions.

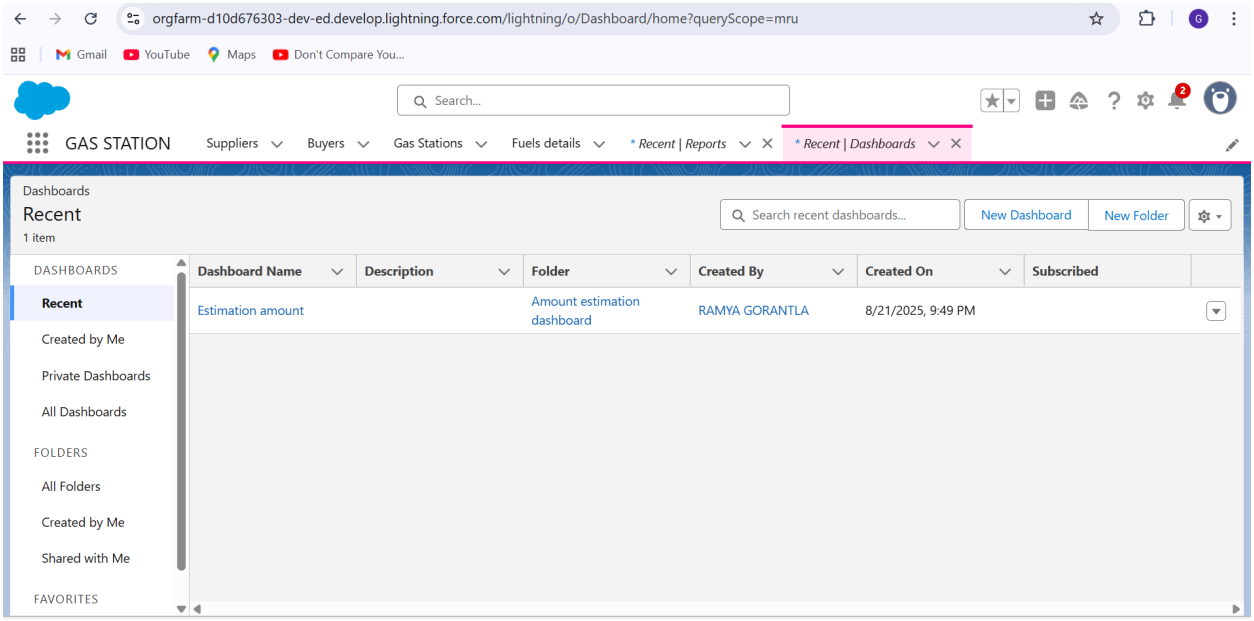
Steps to Create a Dashboard:

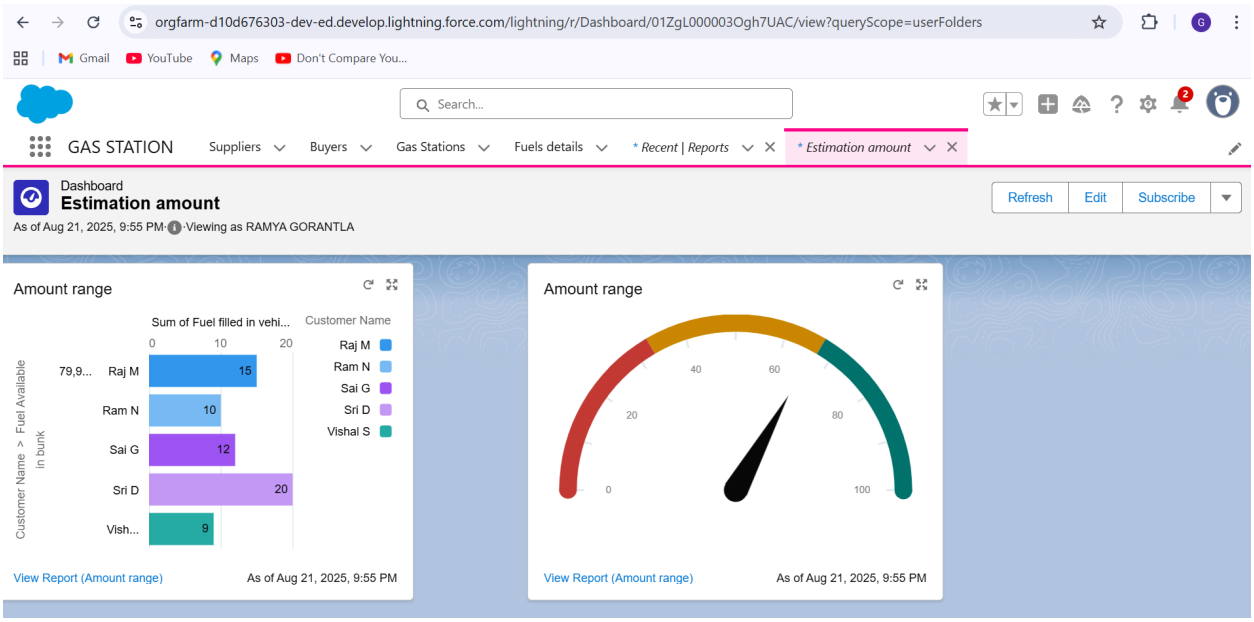
1. **Login to Salesforce Developer Account**
2. Navigate to **Dashboards** → **New Dashboard**
3. Fill in details:
    - **Dashboard Name:** e.g., Gas Station Performance Dashboard
    - **Folder:** e.g., Gas Station Dashboards.
    - **Description:** Optional
4. Click **Create**
5. Add components:
    - Click **+ Component**
    - Select the **source report** (e.g., Fuel Sales Report, Buyer Transactions Report)
    - Choose **component type** (Bar Chart, Pie Chart, Gauge, Metric, Table)
    - Configure **display options** (groupings, filters, labels)
6. Arrange components on the dashboard canvas
7. Click **Save & Run**

Example Dashboard in Gas Filling Station CRM:

- **Fuel Sales Dashboard**:
    - Bar chart: Fuel sold by Gas Station
    - Pie chart: Fuel sales by Fuel Type
    - Table: Top 10 buyers by purchase quantity
    - Metric: Total Fuel Sold This Month
- **Sales Performance Dashboard**:

- Line chart: Daily sales trend
- Gauge: Target vs Actual Sales
- Table: Sales by Sales Person.





Flows:

Flows in Salesforce are **automated processes** that guide users through a series of screens,

gather information, and perform actions like creating or updating records. Flows help **automate repetitive tasks** and improve efficiency in CRM operations.

Types of Flows:

1. **Screen Flow**
    - Interactive flow that guides users through screens to collect data.
    - Example: A flow for a Sales Person to log a fuel purchase.
2. **Record-Triggered Flow**
    - Automatically runs when a record is created, updated, or deleted.
    - Example: Automatically update fuel inventory when a fuel sale record is created.
3. **Schedule-Triggered Flow**
    - Runs at a specific time or interval.
    - Example: Send weekly sales summary to Managers.
4. **Autolaunched Flow**
    - Runs automatically without user interaction, often called by other flows or processes.
    - Example: Auto-assign new buyers to a Sales Person.

Creating a Flow:

A **Flow** in Salesforce automates business processes, such as logging fuel sales, updating inventory, or assigning buyers, without needing code. Creating flows improves efficiency and ensures **consistent and accurate data**.

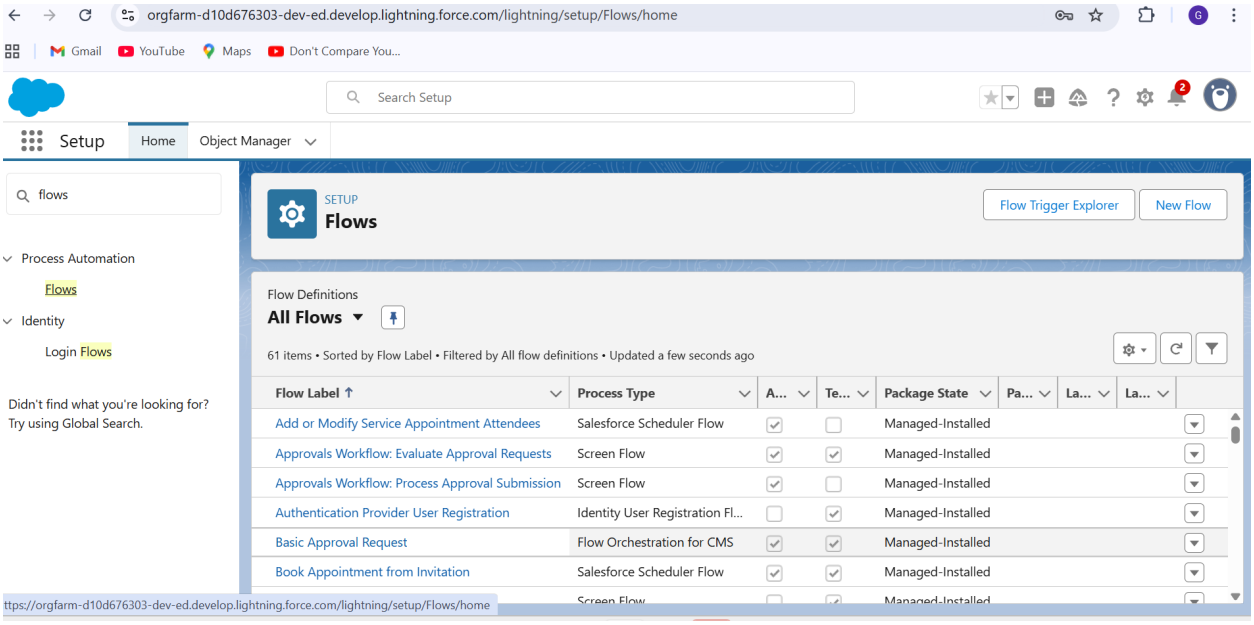
Steps to Create a Flow:

1. **Login to Salesforce Developer Account**
2. Navigate to **Setup → Process Automation → Flows → New Flow**
3. Select the **Flow Type**:
    - **Screen Flow:** Interactive, requires user input
    - **Record-Triggered Flow:** Runs automatically when a record is created/updated/deleted
    - **Autolaunched Flow:** Runs automatically without user input
    - **Scheduled-Triggered Flow:** Runs at scheduled intervals
4. Click **Create**
5. Use the **Flow Builder** to design the flow:
    - Drag **Elements** from the toolbox:
        - **Screen:** Collect input from users
        - **Get Records:** Retrieve existing records

        - **Create Records:** Create new records
        - **Update Records:** Update existing records
        - **Decision:** Add conditional logic
        - **Loop:** Process multiple records
    - Connect elements in the correct sequence to define the process
6. Click **Save** and provide a **Flow Name** and **Description**
7. Click **Activate** to make the flow available for use

Example: Fuel Sale Logging Flow

- **Flow Type:** Screen Flow
- **Steps:**
    a. **Screen 1:** Select Buyer
    b. **Screen 2:** Enter Fuel Type and Quantity
    c. **Screen 3:** Confirm Total Price
    d. **Action:** Create a **Fuel Details** record and update Buyer's purchase history.



Apex Trigger:

An **Apex Trigger** is a piece of Apex code that executes **before or after certain events** occur on Salesforce records, such as creating, updating, deleting, or undeleting. Triggers allow you to **automate complex processes** that cannot be handled with standard Salesforce automation

tools like Flows or Process Builder.

Types of Apex Triggers:

1. **Before Triggers**
    - Executes **before a record is saved** to the database.
    - Used for **validation or modifying record values**.
    - Example: Before creating a Fuel Details record, calculate the total price automatically.
2. **After Triggers**
    - Executes **after a record is saved** to the database.
    - Used for **updating related records or performing actions that require the record ID**.
    - Example: After creating a Buyer record, automatically assign them to a Sales Person.

Basic Syntax of an Apex Trigger:

```
trigger TriggerName on ObjectName (before insert, after insert, before
update, after update) {
    for (ObjectName obj : Trigger.New) {
        // Trigger logic here
    }
}
```

Example: Fuel Stock Update Trigger

Scenario: When a **Fuel Details** record is created, automatically reduce the fuel stock in the corresponding **Gas Station**.

```
trigger UpdateFuelStock on Fuel_Details__c (after insert) {
    List<Gas_Station__c> stationsToUpdate = new
List<Gas_Station__c>();

    for (Fuel_Details__c fuel : Trigger.New) {
        Gas_Station__c station = [SELECT Id, Fuel_Stock__c FROM
Gas_Station__c WHERE Id = :fuel.Gas_Station__c LIMIT 1];
        station.Fuel_Stock__c -= fuel.Quantity__c;
        stationsToUpdate.add(station);
```

```
    }

    update stationsToUpdate;
}
```

Apex Handler:

An **Apex Handler** (also called a **Trigger Handler**) is a **separate Apex class** used to manage the logic for an Apex Trigger. Using handlers promotes **clean, modular, and reusable code** and separates **business logic** from trigger execution.

Basic Structure of an Apex Handler:

```
public class FuelDetailsHandler {

    // Method to update Gas Station fuel stock
    public static void updateFuelStock(List<Fuel_Details__c> fuelList)
{
        List<Gas_Station__c> stationsToUpdate = new
List<Gas_Station__c>();
            for (Fuel_Details__c fuel : fuelList) {
            Gas_Station__c station = [SELECT Id, Fuel_Stock__c
                                      FROM Gas_Station__c
                                      WHERE Id = :fuel.Gas_Station__c
LIMIT 1];
            station.Fuel_Stock__c -= fuel.Quantity__c;
            stationsToUpdate.add(station);
        }

        update stationsToUpdate;
    }
}
```

Using Apex Handler in a Trigger:

```
trigger FuelDetailsTrigger on Fuel_Details__c (after insert) {
    // Call handler method
    FuelDetailsHandler.updateFuelStock(Trigger.New);
}
```