

A CRM Application To Handle The Clients And Their Property Related Requirements

Project Description:

Dreams World Properties integrates Salesforce to streamline customer interactions. Website engagement triggers automated record creation in Salesforce, capturing customer details and preferences. Salesforce categorizes users as approved or non-approved, offering tailored property selections to approved users. This enhances user experience and efficiency, providing personalized recommendations and broader listings. Seamless integration optimizes operations, improving customer engagement and facilitating growth in the real estate market.

1. Client Management

1. Add, update, and delete client details.
2. Track client preferences, budget, and location interests.
3. Maintain contact details and communication history.

2. Property Management

1. Manage property listings with details like type, price, location, and features.
2. Track properties available for sale, rent, or lease.
3. Upload photos and documents for properties.

3. Requirement Matching

1. Match client requirements with available properties using filters.
2. Notify clients about new properties that fit their criteria.

4. Lead Tracking

1. Manage inquiries and follow up with potential clients.
2. Schedule meetings and site visits.
3. Assign leads to specific team members.

Milestone 1:- Create a Jotform and integrate it with the org to create a record of customers automatically.

Client wants a form for the customers to get the details directly into the salesforce so that the admins can create a user in the org. Client wants a form for the customers to get the details directly into the salesforce so that the admins can create a user in the org.

Activity: 1

1. Open your browser and search for jotform and log in.
2. After login click on create form and click on start from scratch
3. Now create a form to get the customer details like Name, Phone, Email, Address and type of property the customer is interested in.
4. Once the form is created, publish it by clicking on publish,
5. form link: <https://form.jotform.com/243224871950054>

The screenshot shows the Jotform Form Builder interface. On the left, there is a sidebar titled 'Form Elements' with various input types listed: AND, PAYMENTS, WIDGETS, Heading, Full Name, Email, Address, Phone, Date Picker, Appointment, Signature, Fill in the Blank, Product List, and more. The main workspace is titled 'Dreams World' and contains the following fields:

- Name (with two separate input fields for First Name and Last Name)
- Phone Number
- Email
- Phone Number
- Which type of Property are you looking for?
 - RESIDENTIAL
 - COMMERCIAL
 - RENTAL
- Budget Amount
- Address
- Block Number
- Street Address Line 2
- City
- State / Province
- Postal / Zip Code

At the bottom right of the form, there is a green 'Submit' button. Below the form, there is a footer bar with the Jotform logo and links for 'Create Field', 'Create Form', and 'Create your own software'.

Create Objects from Spreadsheet

Directly Creating Objects from Spreadsheet in Salesforce

Creating Customer Object:

1. Go to your object manager and click on create object from spreadsheet.
2. Click on the link to get the spreadsheet
3. [customer](#)

Customer	Phone Number	Email	State	Property Type	Budget Amount	Street Address	Street Address	City	postal code	Verified
Rakesh	788797	rakesh@gmail.com	Telangana	Residential	4000000	gb road	street no 45	Hyderabad	555001	checked
prakash	55448855	p@gmail.com	Maharashtra	Commercial	8000000	gachibowli	indira road	mumbai	6600014	unchecked
Prajwal	454545	prajwal@gmail.com	Maharashtra	Rental	25000	kamldi	kathora	Amravati	444805	checked

4. After downloading, upload the file, map the fields and upload to create an object.

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. On the left, a sidebar lists various configuration options like Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main panel displays the 'Details' section for the 'Customer' object. It includes fields for API Name (Customer__c), Singular Label (Customer), Plural Label (Customer), and several checkboxes for enabling Reports, Track Activities, and Track Field History. At the bottom right of the main panel are 'Edit' and 'Delete' buttons.

Creating Property Object:

1. Follow the same from the customer object to create the Property Object
2. [Property](#)

A	B	C	D
Property Name	Type	Location	Verified
Lotus Appartme	Residential	hydeerabad	checked
500000 sq.ft pl	Commercial	Amravati	unchecked
3 Bhk fkat at st	rental	Jubilee hill Hyd	Checked

1. After downloading, upload the file, map the fields and upload to create an object.
2. the fileds as follows.

The screenshot shows the Salesforce Setup interface under the Object Manager. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Property'. On the left, a sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types (which is selected), Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main content area displays the 'Details' tab for the 'Property' object. It shows the API Name as 'Property__c', a custom singular label 'Property', and a plural label 'Property'. Under the 'Edit' tab, there are sections for 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History', 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (link to 'Standard salesforce.com Help Window').

Integrate Jotform with Salesforce Platform

In this Milestone we are going to integrate jotform with Salesforce

Activity :-1

1. On the Jotform Platform, Click on Integration and choose Salesforce
2. Click on User Integration and choose "Add to From"
3. Select the Org with which you want to Integrate your jotform with and select your account
4. Select an Action - Create a record.
5. Select a Salesforce Object: - Customer

Map Each and every field on the Object with the fields on the form and "Save Action".

The screenshot shows the Jotform integration setup interface. At the top, there are three tabs: BUILD (highlighted in orange), SETTINGS, and PUBLISH. Below the tabs, on the left, is a sidebar with various settings options: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (selected), WORKFLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The main area is titled 'SALESFORCE' with the sub-instruction 'Send new leads, contacts, or accounts to your sales CRM'. A sub-section titled 'Select a Salesforce Object' has a dropdown menu set to 'Customer'. The 'Create a record' section contains a table mapping form fields to Salesforce objects:

Object Fields	Dreamhome
Customer__c	Name - First Name
City	Address - City
Budget Amount	Budget Amount
Property Type	Which type of property are you lookin...
Phone Number	Phone Number
Street Address	Address - Street Address
Email	Email
Customer Name	Name - Last Name
State	Address - State
Street Address line 2	Address - Street Address 2

Then "Save the Integration" and "Finish".



SALESFORCE

Send new leads, contacts, or accounts to your sales CRM

All Actions

See Action Logs

+ Add New Action

1

Create or update a record
Customer

Create Roles

Here we need to Create Roles as per business requirement

Activity :- 1

Go to Setup and Click on Roles, then click on Expand all and Add a Role just below the Sales Representative

The screenshot shows the Salesforce Setup interface with the 'Roles' page selected. The left sidebar includes links for 'Setup', 'Home', and 'Object Manager'. A search bar at the top right says 'Search Setup'. The main content area has a title 'SETUP Roles' with a user icon. Below it, there's a 'Collapse All' and 'Expand All' button. The page lists various roles under a user named 'Prasad.V.Poturi Siddhartha Institute Of Techno...'. The roles are categorized by department: Sales, Service, and Feature Settings. Under Sales, there are several sub-roles like 'CEO', 'CFO', 'COO', and 'SVP Customer Service & Support'. Under Service, there are 'SVP Human Resources' and 'SVP Sales & Marketing'. Under Feature Settings, there are 'VP International Sales', 'VP Marketing', and 'Marketing Team'. At the bottom of the list, there are 'Sales Representative', 'Sales Executive', 'Sales Manager', and 'Customer'. Each role entry includes 'Edit', 'Del', and 'Assign' buttons, along with an 'Add Role' link.

1. If we don't find sales representative we need to create it according to the need.
2. It will use the "System Administrator Profile".
3. Label Sales Executive.
4. Reports to - Sales Representative.

Role Edit
Sales Executive

Role Edit

Label	<input type="text" value="Sales Executive"/>
Role Name	<input type="text" value="Sales_Executive"/> i
This role reports to	<input type="text" value="Sales Representative"/> 🔍
Role Name as displayed on reports	<input type="text"/>

[Save](#) [Save & New](#) [Cancel](#)

1. Similarly Create a Role Name "Sales Manager" below Sales Executive which reports to Sales Executive, Also Add a Role below Sales Manager labeled as "Customer" which reports to Sales Manager.

Create a Property Details App

An App where the objects will be displayed

Activity :- 1

1. From Setup>> Go to App Manager and click on New Lightning App and Name it as "Property Details" and add "Customer" and "Property" Object.
2. Click Next >> Next >> Save and Add "System Admin "Profile.

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name

*Developer Name

Description

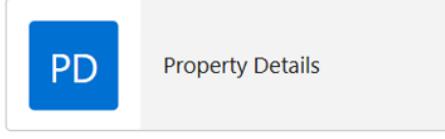
App Branding

Image

Primary Color Hex Value

Org Theme Options Use the app's image and color instead of the org's custom theme

App Launcher Preview



Create Profiles

Create profiles as per business requirement

Creating Customer Profile :-

1. From Setup? Go to Profiles and Clone (standard platform) Salesforce Platform User and Name it "Customer".
2. Uncheck all the Custom Objects and Check only Property Details From Custom App Settings.
3. Also Remove all the Standard Object Permissions
4. Uncheck all the Custom Object Permissions and check read and view all in "Property"
5. Make sure every submission object permissions are unselected and then save.

The screenshot shows the Salesforce Setup interface under the Profiles section. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar. The main content area displays the 'Standard Object Permissions' and 'Custom Object Permissions' sections for a new profile being created. In the 'Standard Object Permissions' section, most checkboxes are unchecked, except for 'Read' and 'View All' under 'Basic Access' for the 'Property' object. In the 'Custom Object Permissions' section, the 'Customer' object has its 'Read' and 'View All' checkboxes checked. At the bottom, session settings are set to '2 hours of inactivity' and 'Session Security Level Required at Login' is set to 'None'.

Creating Manager Profile :-

1. From Setup >> Go to Profiles and Clone Salesforce Platform User and Name it "Manager".
2. Uncheck all the Custom Objects and Check only Property Details From Custom App Settings.
3. Also Remove all the Standard Object Permissions.
4. Uncheck all the Custom Object Permissions and check only "modify all" from "Property" and "Customer".

Standard Object Permissions						
The permissions defined here control access at the object level. Access to individual records within that object type is controlled by the sharing model. Set access levels based on the functional requirements for the profile. For example, create different groups of permissions for individual contributors, managers, and administrators. How do I choose?						
	Basic Access			Data Administration		
	Read	Create	Edit	Delete	View All	Modify All
Accounts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Active Scratch Orgs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Forms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Data Uses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Tests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Background Operations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Business Brands	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscriptions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Timings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contacts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Addresses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Emails	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Phones	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Custom Object Permissions						
	Basic Access			Data Administration		
	Read	Create	Edit	Delete	View All	Modify All
Customer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Property	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Session Settings						
Session Times Out After		<input type="checkbox"/> 2 hours of inactivity				
Session Security Level Required at Login						
<input type="checkbox"/> None		<input type="checkbox"/> Standard				

Create a Check Box field on user

Create Field on the User as per the business requirement,

Activity :- 1

1. Setup >> Object Manager >> Search for User >> Fields and Relationships
2. Select the Data type "Check Box"
3. Create new Field Named as "Verified"

The screenshot shows the Salesforce Object Manager interface for creating a custom field. The left sidebar lists various setup options like Details, Fields & Relationships, and Page Layouts. The main area is titled 'User Custom Field Verified' under 'Fields & Relationships'. The 'Field Information' section shows the field is named 'Verified' with API name 'Verified_c'. It's a checkbox type ('checkbox') and is associated with the 'User' object. The 'Validation Rules' section indicates 'No validation rules defined'. The 'General Options' section shows the default value is 'Unchecked'. The 'Created By' and 'Modified By' fields both show '21501A05A8 GAYATRI MANGALAGIRI' with the timestamp '18/11/2024, 11:49 pm'.

Create Users

Create three different users with three different Roles and profiles as we have mentioned above.
Here we are going to create 4 users.

User: 1

1. Go to Setup -> Administration -> Users -> New User
2. Last Name - Executive
3. Role - Sales Executive
4. License - Salesforce
5. Profile - System Administrator
6. Save

User Edit
Executive

User Edit

Save Save & New Cancel

General Information			
First Name	<input type="text"/>	Role	Sales Executive
Last Name	Executive	User License	Salesforce
Alias	exec	Profile	System Administrator
Email	hellouser1@gmail.com	Active	<input checked="" type="checkbox"/>
Username	hellouser12@gmail.com	Marketing User	<input type="checkbox"/>
Nickname	User1731917440072859977	Offline User	<input type="checkbox"/>
Title	<input type="text"/>	Knowledge User	<input type="checkbox"/>
Company	<input type="text"/>	Flow User	<input type="checkbox"/>
Department	<input type="text"/>	Service Cloud User	<input type="checkbox"/>
Division	<input type="text"/>	Site.com Contributor User	<input type="checkbox"/>
		Site.com Publisher User	<input type="checkbox"/>
		WDC User	<input type="checkbox"/>
		Data.com User Type	--None--

User: 2

1. Go to Setup >> Administration >> Users >> New User
2. Last Name >> Manager
3. Role >> Sales Manager
4. License >> Salesforce Platform
5. Profile >> Manager
6. Save

**User Edit
Manager**Help for this Page 

User Edit Manager

General Information

First Name	<input type="text"/>	Role	Sales Manager
Last Name	Manager	User License	Salesforce Platform
Alias	mana	Profile	Manager
Email	hellouser2@gmail.com	Active	<input checked="" type="checkbox"/>
Username	hellouser21@gmail.com	Marketing User	<input type="checkbox"/>
Nickname	User1731917567751583154	Offline User	<input type="checkbox"/>
Title	<input type="text"/>	Knowledge User	<input type="checkbox"/>
Company	<input type="text"/>	Flow User	<input type="checkbox"/>
Department	<input type="text"/>	Service Cloud User	<input type="checkbox"/>
Division	<input type="text"/>	Site.com Contributor User	<input type="checkbox"/>
Site.com Publisher User			
WDC User			
Data.com User Type <input type="text"/> -None--			
Data.com Monthly Addition Limit <input type="text"/> 300			
Accessibility Mode (Classic Only) <input type="checkbox"/>			
High-Contrast Palette on Charts <input type="checkbox"/>			
Load Lightning Pages While Scrolling <input checked="" type="checkbox"/>			

User: 3

1. Go to Setup >> Administration >> Users >> New User
2. Last Name >> Customer
3. Role >> Customer
4. License >> Salesforce Platform
5. Profile >> Customer
6. Make Sure the verified check box is "Unchecked"
7. Save

**User Edit
Customer**Help for this Page 

User Edit Customer

General Information

First Name	<input type="text"/>	Role	Customer
Last Name	Customer	User License	Salesforce Platform
Alias	cust	Profile	Customer
Email	hellouser3@gmail.com	Active	<input checked="" type="checkbox"/>
Username	hellouser31@gmail.com	Marketing User	<input type="checkbox"/>
Nickname	User1731917644996634866	Offline User	<input type="checkbox"/>
Title	<input type="text"/>	Knowledge User	<input type="checkbox"/>
Company	<input type="text"/>	Flow User	<input type="checkbox"/>
Department	<input type="text"/>	Service Cloud User	<input type="checkbox"/>
Division	<input type="text"/>	Site.com Contributor User	<input type="checkbox"/>
Site.com Publisher User			
WDC User			
Data.com User Type <input type="text"/> -None--			
Data.com Monthly Addition Limit <input type="text"/> 300			
Accessibility Mode (Classic Only) <input type="checkbox"/>			
High-Contrast Palette on Charts <input type="checkbox"/>			
Load Lightning Pages While Scrolling <input checked="" type="checkbox"/>			

User: 4

1. Go to Setup >> Administration >> Users >> New User
2. Last Name >> Customer2
3. Role >> Customer
4. License >> Salesforce Platform
5. Profile >> Customer
6. Make Sure the verified check box is "checked"
7. Save

User Edit
Customer2

Help for this Page

User Edit

Save Save & New Cancel

General Information

Required Information

First Name	<input type="text"/>	Role	<input type="text" value="Customer"/>
Last Name	<input type="text" value="Customer2"/>	User License	<input type="text" value="Salesforce Platform"/>
Alias	<input type="text" value="cust2"/>	Profile	<input type="text" value="Customer"/>
Email	<input type="text" value="hellouser4@gmail.com"/>	Active	<input checked="" type="checkbox"/>
Username	<input type="text" value="hellouser41@gmail.com"/>	Marketing User	<input type="checkbox"/>
Nickname	<input type="text" value="User1731917730413946117"/>	Offline User	<input type="checkbox"/>
Title	<input type="text"/>	Knowledge User	<input type="checkbox"/>
Company	<input type="text"/>	Flow User	<input type="checkbox"/>
Department	<input type="text"/>	Service Cloud User	<input type="checkbox"/>
Division	<input type="text"/>	Site.com Contributor User	<input type="checkbox"/>
		Site.com Publisher User	<input type="checkbox"/>
		WDC User	<input type="checkbox"/>
		Data.com User Type	<input type="text" value="--None--"/>
		Data.com Monthly Addition Limit	<input type="text" value="300"/>
		Accessibility Mode (Classic Only)	<input type="checkbox"/>
		High-Contrast Palette on Charts	<input type="checkbox"/>

Create an Approval Process for Property Object

An Approval process to approve or reject the records as according

Activity :- 1

1. From Setup >> Process Automation >> Approval Process
2. before proceeding we need to select property in the manage approval process
3. Process Name - Property Approval
4. Select 2 criteria -
 - i. Location - not equal to - blank,
 - ii. Verified-Equals- false
5. Click next and "Next Automated Approver Determined By" Select Manager
6. From Record Editability Properties >> Click on Administrators OR the currently assigned approver can edit records during the approval process.
7. From Step 5: Select Fields to Display on Approval Page Layout select Property, Owner, Location, Type.

Approval Process Edit
Property Approval

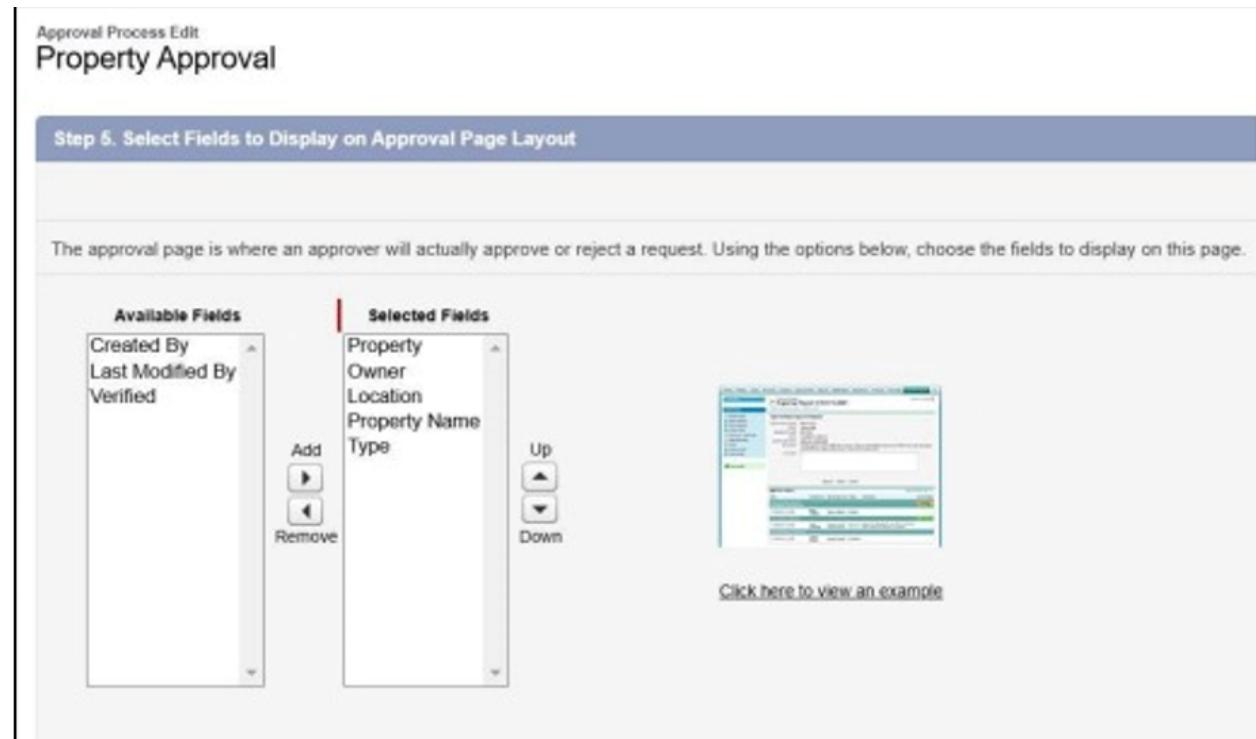
Step 5. Select Fields to Display on Approval Page Layout

The approval page is where an approver will actually approve or reject a request. Using the options below, choose the fields to display on this page.

Available Fields	Selected Fields
Created By Last Modified By Verified	Property Owner Location Property Name Type

Add Remove Up Down

[Click here to view an example](#)



Click Next and Select the initial Submitters >>

1. Owner >> Property Owner
2. Roles >> Sales Manager

Save

After saving we are directed to approval steps and we need to do as follows

Add an approval step name "Executive Approval "

Click next and select the Approver as " Sales Executive" and "Save"

Add One field Update as "Verified Property"

1. Select Object >> Property
2. Field to Update >> Verified
3. Field Data Type >> CheckBox
4. Select CheckBox Option as "True"
5. Save.

Add One field Update as "UnVerified Property"

1. Select Object >> Property
2. Field to Update >> Verified
3. Field Data Type >> CheckBox
4. Select CheckBox Option as "False"
5. Save.

Activate the Approval Process.

The screenshot shows the 'Approval Processes' page in Salesforce. The top navigation bar includes 'SETUP' and 'Approval Processes'. The main title is 'Property: Property Approval'. Below the title, there are links to 'Help for this Page' and 'Back to Approval Process List'.

Process Definition Detail

Process Name	Property Approval	Active	✓
Unique Name	Property_Approval	Next Automated Approver Determined By	Manager of Record Submitter
Description			
Entry Criteria	(Property: Location NOTEQUAL to blank) AND (Property: Verified EQUALS False)		
Record Editability	Administrator OR Current Approver	Allow Submitters to Recall Approval Requests	<input type="checkbox"/>
Approval Assignment Email Template			
Initial Submitters	Property Owner, Role_Sales Manager		
Created By	21501A05A8 GAYATRI MANGALAGIRI	Created On	19/11/2024, 12:04 am
		Modified By	21501A05A8 GAYATRI MANGALAGIRI
		Modified On	19/11/2024, 12:08 am

Initial Submission Actions

Action	Type	Description
Record Lock		Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Executive Approval			User Executive	Final Rejection

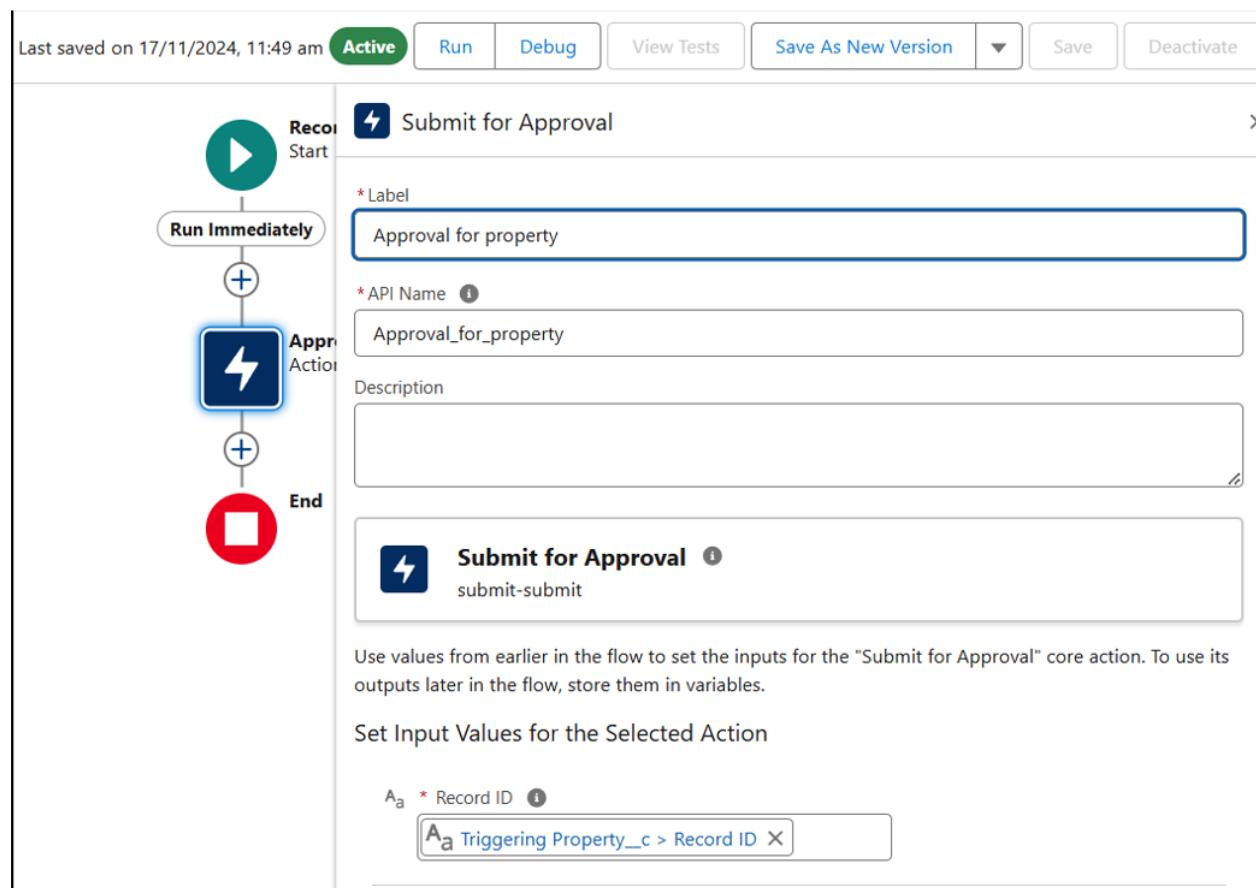
Create a Record trigger flow to submit the Approval Process Automatically

A flow that can submit the records directly for approval

Activity :- 1

1. From Setup >> Search for Flows >> Click On New and Select "Record Trigger Flow".
2. Select Object >> Property
3. Select "Trigger the flow when" >> "A record is created"
4. Set Entry Conditions >> "None"
5. Add a "Action" >> "Submit for Approval"
6. Give Label >> Approval for property
7. Record Id >> `(!$Record.Id)`
8. Done

Save the Flow and Give label as "Property Approval" and "Activate"

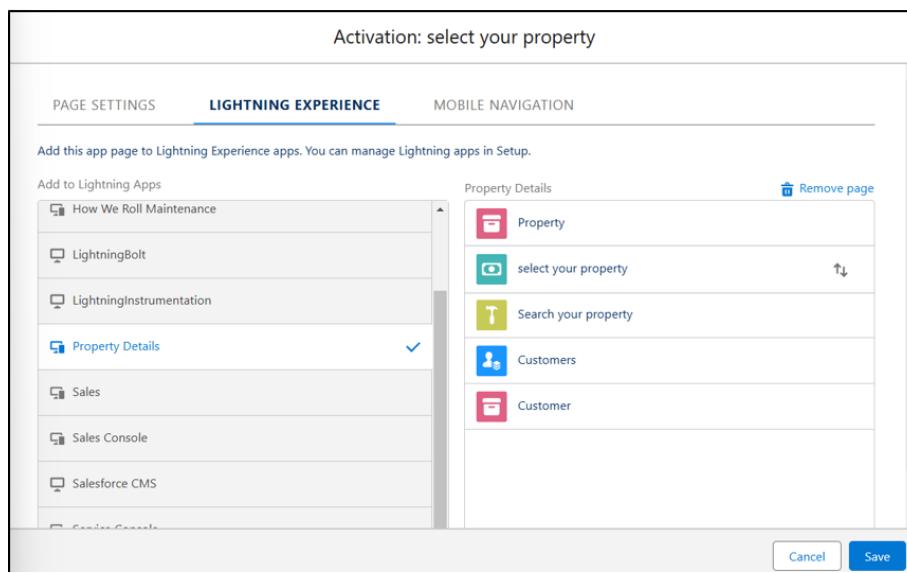
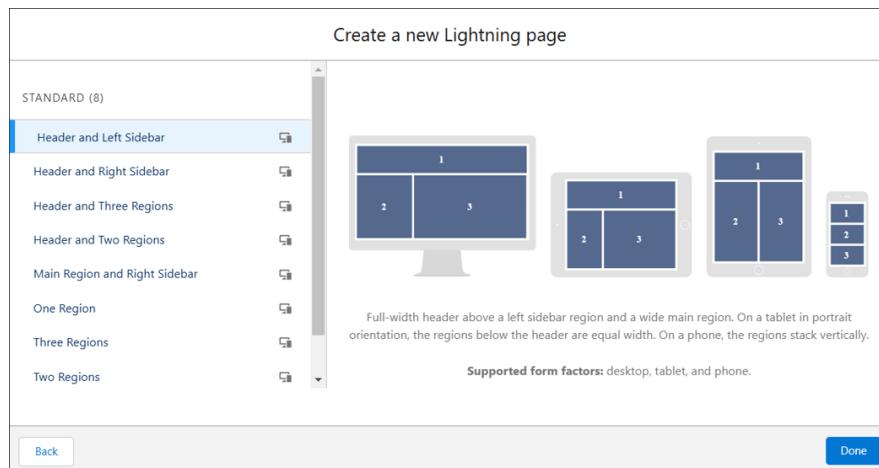


Create an App Page

Create an App Page on the Property details Object named as "Search Your Property"

Activity :- 1

1. From Setup >> Go to Lightning App Builder >> Click on New >> Select App Page and
2. Click on Next
3. Give Label as "Search your Property" click "Next".
4. Click "header and Left Sidebar" and Click on "Done"
5. Click on "Save" and then click on "Activate".
6. From Page Setting select page activation as "Activate for all Users".
7. From Lightning Experience Click on "Property Details" and click on Add Page".
8. Then Click on "Save"



Create a LWC Component

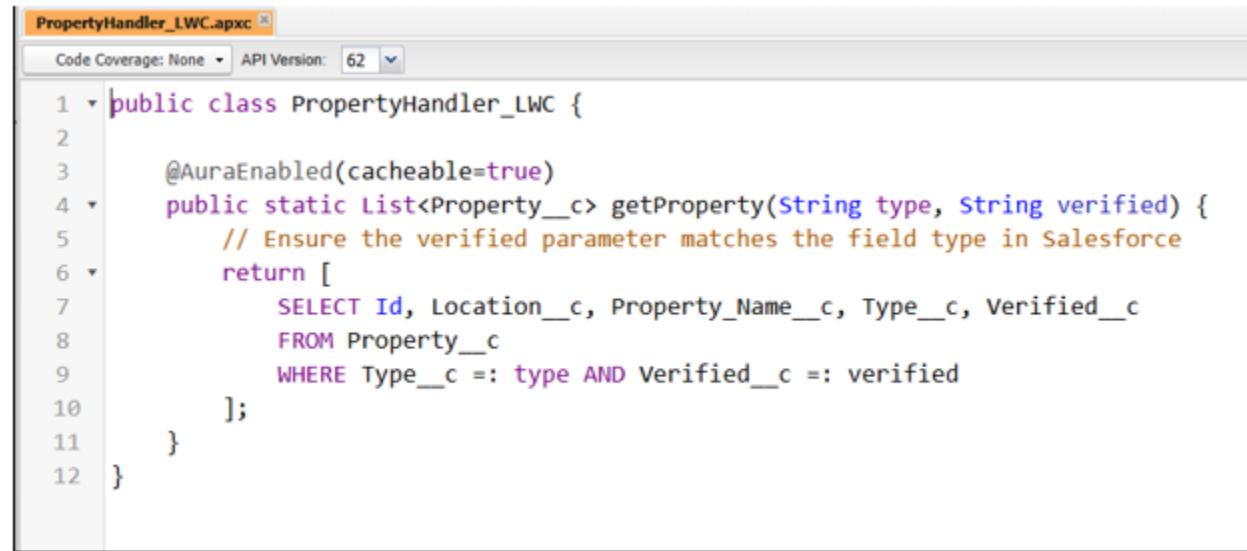
Create an LWC Component for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on "Search your Property Page"

Activity :- 1

1. Create an Apex Class and make it aura enabled and name it "PropertyHandler_LWC".

Code:

```
public class PropertyHandler_LWC {  
    @AuraEnabled(cacheable=true)  
    public static List<Property__c> getProperty(String type, String verified) {  
        // Ensure the verified parameter matches the field type in Salesforce  
        return [  
            SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c  
            FROM Property__c  
            WHERE Type__c =: type AND Verified__c =: verified  
        ];  
    }  
}
```



```
1  public class PropertyHandler_LWC {  
2  
3      @AuraEnabled(cacheable=true)  
4      public static List<Property__c> getProperty(String type, String verified) {  
5          // Ensure the verified parameter matches the field type in Salesforce  
6          return [  
7              SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c  
8              FROM Property__c  
9              WHERE Type__c =: type AND Verified__c =: verified  
10         ];  
11     }  
12 }
```

1. Create a Lightning Web Component in your VsCode, and (ctrl+shift +P) and click on authorize an org.
2. Enter your login ic and password to authorize your org.
3. Now (ctrl+shift +P) and Create a lightning Web Component and Name it Anything you want to.
4. In your Html File, write this code

Code:

```
<template>
<lightning-card>
<div class="slds-box">
<div class="slds-text-align_left">
<h1 style="font-size: 20px;"><b>Properties</b></h1>
</div>
<div>
<div class="slds-grid slds-gutters">
<div class="slds-col slds-size_5-of-6">
<lightning-combobox name="Type" label="Property Type" value={typevar}
placeholder="Select Property type"
options={propertyoptions} onchange={changehandler}></lightning-combobox>
</div>
<div class="slds-col slds-size_1-of-6">
<br>
<lightning-button-icon variant="neutral" icon-name="standard:search" alternative-
text="Search"
label="Search" onclick={handleClick}></lightning-button-icon>
</div>
</div>
</div>
<template if:true={istru}>
<div class="slds-box">
<lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-
datatable>
</div>
</template>
<template if:false={isfalse}>
<div class="slds-box">
```

```

<div style="font-size: 15px;"><b>No properties Are Found !!</b></div>
</div>
</template>
</lightning-card>
</template>

```

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "LWCPR...". The "sfp.html" file is selected.
- File List:** Shows files like ".husky", ".sf", ".sfdx", ".vscode", "config", "force-app/main/def...", "applications", "aura", "classes", "contentsets", "flexipages", "layouts", "lwc", "sfp", "_tests_".
- Code Editor:** Displays the template for "sfp.html". The code includes a lightning-card component. Inside the card, there is a message: "<div style='font-size: 15px;'>No properties Are Found !!</div>".
- JavaScript File:** A JS file named "sfp.js" is also visible in the list.

6. In your Js File, write this code

Code:

```

import { LightningElement, api, track, wire } from 'lwc';
import getProperty from "@salesforce/apex/PropertyHandler_LWC.getProperty"
import { getRecord } from 'lightning/uiRecordApi';
import USER_ID from '@salesforce/user/Id';
export default class C_01_Property_Management extends LightningElement {
    @api recordId
    userId = USER_ID;
    verifiedvar
    typevar
    isfalse = true;
    istrue = false;
    @track propertylist = [];
    columns = [
        { label: 'Property Name', fieldName: 'Property_Name__c' },
        { label: 'Property Type', fieldName: 'Type__c' },
        { label: 'Property Location', fieldName: 'Location__c' },
    ]
}

```

```

        { label: "Property link", fieldName: "Property_link__c" }
    ]
    propertyoptions = [
        { label: "Commercial", value: "Commercial" },
        { label: "Residential", value: "Residential" },
        { label: "rental", value: "rental" }
    ]
    @wire(getRecord, { recordId: "$userId", fields: ['User.Verified__c'] })
    recordFunction({ data, error }) {
        if (data) {
            console.log(data)
            console.log("This is the User Id ---> "+this.userId);
            this.verifiedvar = data.fields.Verified__c.value;
        } else {
            console.error(error)
            console.log('this is error')
        }
    }
    changehandler(event) {
        console.log(event.target.value);
        this.typevar = event.target.value;
    }
    handleClick() {
        getProperty({ type: this.typevar, verified: this.verifiedvar })
        .then((result) => {
            this.isfalse = true;
            console.log(result)
            console.log('This is the User id ---> ' + this.userId);
            console.log('This is the verified values ---> ' + this.verifiedvar);
            if (result != null && result.length != 0) {
                this.isTrue = true;
                this.propertylist = result;
                console.log(this.verifiedvar);
                console.log(this.typevar)
            } else {
                this.isfalse = false;
                this.isTrue = false;
            }
        })
    }
}

```

```

        }
    })
    .catch((error) => {
        console.log(error)
    })
}

}

```

```

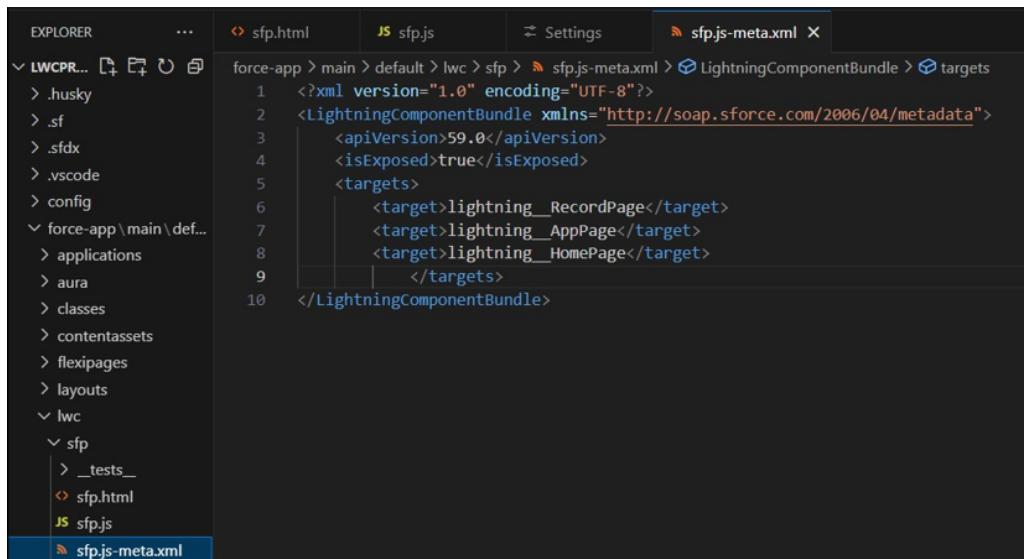
force-app > main > default > lwc > sfp > sfp.js > ...
1 import { LightningElement, api, track, wire } from 'lwc';
2 import getProperty from '@salesforce/apex/PropertyHandler_LWC.getProperty'
3 import { getRecord } from 'lightning/uiRecordApi';
4 import USER_ID from '@salesforce/user/Id';
5 export default class C_01_Property_Management extends LightningElement {
6     @api recordId
7     userId = USER_ID;
8     verifiedvar
9     typevar
10    isfalse = true;
11    istrue = false;
12    @track propertylist = [];
13    columns = [
14        { label: 'Property Name', fieldName: 'Property_Name__c' },
15        { label: 'Property Type', fieldName: 'Type__c' },
16        { label: 'Property Location', fieldName: 'Location__c' },
17        { label: "Property link", fieldName: "Property_link__c" }
18    ]
19    propetyoptions = [
20        { label: "Commercial", value: "Commercial" },
21        { label: "Residential", value: "Residential" },
22        { label: "rental", value: "rental" }
23    ]
24    @wire(getRecord, { recordId: "$userId", fields: ['User.Verified__c'] })
25    recordFunction({ data, error }) {
26        if (data) {
27            console.log(data)
28            console.log("This is the User Id ---> "+this.userId);
29            this.verifiedvar = data.fields.Verified__c.value;
30        } else {
31            console.error(error)
32            console.log('this is error')
33        }
34    }
35    changehandler(event) {
36        console.log(event.target.value);
37        this.typevar = event.target.value;
38    }
39    handleClick() {
40        getProperty({ type: this.typevar, verified: this.verifiedvar })
41        .then(result) => {
42            this.isfalse = true;
43            console.log(result)
44            console.log('This is the User id ---> ' + this.userId);
45            console.log('This is the verified values ---> ' + this.verifiedvar);
46            if (result != null && result.length != 0) {
47                this.istrue = true;
48                this.propertylist = result;
49                console.log(this.verifiedvar);
50                console.log(this.typevar)
51            } else {
52                this.isfalse = false;
53                this.istrue = false;
54            }
55        })
56        .catch(error) => {
57            console.log(error)
58        }
59    }
60 }
61

```

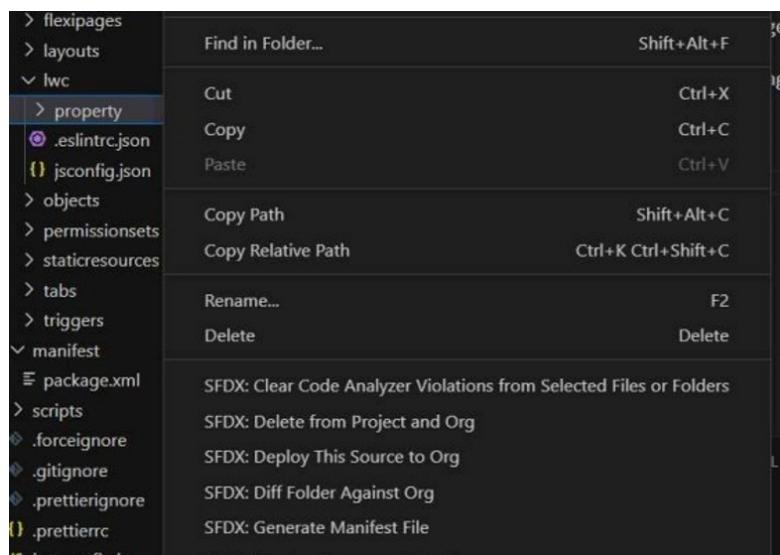
7. In your Metafile, give your targets to deploy the component.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>59.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__RecordPage</target>
        <target>lightning__AppPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```



8. After saving all the three codes, right click and deploy this component to the org.

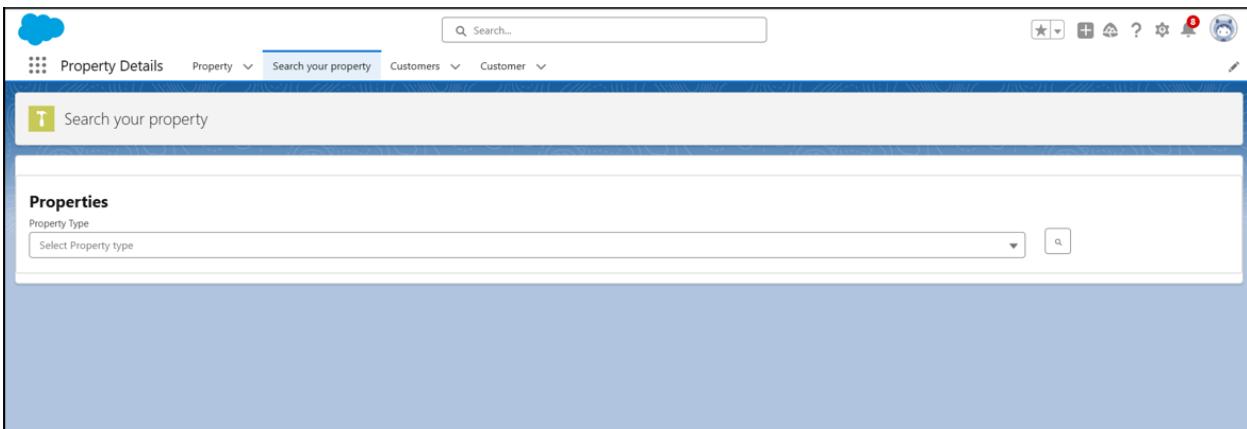


Drag this Component to your App Page

Adding the Component to your Page

Activity :- 1

1. From Setup >> Go to App Launcher >> Search for Property Details
2. On this Page click on gear icon and click on Edit Page
3. After clicking on edit page it will be directed to app pages then
4. Drag the Component(properties) to your App Page and Save the Page.



Give Access of Apex Classes to Profiles

The Apex Class has a Security, Enable the security for the profiles that needs to access this class.

Activity :- 1

1. From Setup >> Search For Apex Classes >> Click on "Security" behind "PropertyHandler_LWC".
2. From Profiles Add "Manager" and "Customer" and "Save".

