

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

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

2. Property Management

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

3. Requirement Matching

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

4. Lead Tracking

- Manage inquiries and follow up with potential clients.
- Schedule meetings and site visits.
- 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/243219167906057>

The screenshot shows the Jotform 'Dreams World' form builder interface. The form is titled 'Dreams World' and includes the following fields:

- Name**: First Name and Last Name
- Phone Number**: (e.g., 773)
- Email**: (e.g., example@example.com)
- Phone Number**: (e.g., 773)
- Which type of Property are you looking for?**:
 - ☐ RESIDENTIAL
 - ☐ COMMERCIAL
 - ☐ RENTAL
- Budget Amount**: (e.g., 773)
- Address**:
 - Street Address
 - House Address Line 2
 - City
 - State/Province
 - Postal/Zip Code

A green 'Submit' button is located at the bottom of the form. The Jotform logo and 'Form Builder' text are visible in the top left. The Jotform logo and 'Create your own Jotform' text are visible in the bottom right.

Create Objects from Spreadsheet

Directly Creating Objects from Spreadsheet in Salesforce

Creating Customer Object:

1. Go to your object manager and 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	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	Maharashtra	Rental	25000	kamdli	kathora	Amravati	444805	checked

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

The screenshot displays the Salesforce Setup interface for the 'Customer' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'Customer' and features a left sidebar with a 'Details' section. The 'Details' section includes a 'Description' field and a 'API Name' field set to 'Customer__c'. The 'Custom' checkbox is checked. The 'Singular Label' is 'Customer', and the 'Plural Label' is 'Customer'. The 'Enable Reports' checkbox is checked, and the 'Track Activities' checkbox is also checked. The 'Track Field History' checkbox is unchecked. The 'Deployment Status' is 'Deployed'. The 'Help Settings' section includes a link to 'Standard salesforce.com Help Window'. The right sidebar contains '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

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

The screenshot displays the Salesforce Setup interface for the 'Property' object. The left sidebar shows a navigation menu with options like 'Details', 'Fields & Relationships', 'Page Layouts', 'Lightning Record Pages', 'Buttons, Links, and Actions', 'Compact Layouts', 'Field Sets', 'Object Limits', 'Record Types', 'Related Lookup Filters', 'Search Layouts', 'List View Button Layout', and 'Restriction Rules'. The 'Details' section is currently selected. The main content area shows the 'Details' configuration for the 'Property' object, including fields for 'Description', 'API Name' (Property_c), 'Custom' (checked), 'Singular Label' (Property), 'Plural Label' (Properties), and 'Property'. On the right side, there are checkboxes for 'Enable Reports' (checked), 'Track Activities' (checked), and 'Track Field History' (checked). Below these, the 'Deployment Status' is set to 'Deployed', and there are links for 'Help Settings' and 'Standard salesforce.com Help Window'.

Field	Value
Description	
API Name	Property_c
Custom	✓
Singular Label	Property
Plural Label	Properties
Property	
Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	
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".

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".

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. The left sidebar contains a navigation menu with the following items: Users, Roles (highlighted), Feature Settings, Sales, Service, and Case Teams. The main content area is titled 'Roles' and displays a hierarchical tree of roles. The roles are organized into a tree structure, with 'Prasad.V.Potturi Siddhartha Institute Of Techno...' at the top. The roles listed include CEO, CFO, COO, SVP, Customer Support, Installation & Repair Services, SVP, Human Resources, SVP, Sales & Marketing, VP, International Sales, VP, Marketing, Marketing Team, VP, North American Sales, Director, Channel Sales, Channel Sales Team, Director, Direct Sales, Eastern Sales Team, Western Sales Team, Sales Representative, Sales Executive, Sales Manager, and Customer. Each role has an 'Add Role' button next to it. The 'Sales Representative' role is expanded, showing its sub-roles: Sales Executive, Sales Manager, and Customer. The 'Add Role' button is visible next to each role in the hierarchy.

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.

The screenshot shows a web interface for editing a role. At the top, it says 'Role Edit' and 'Sales Executive'. Below this is a section titled 'Role Edit' containing four input fields: 'Label' with the value 'Sales Executive', 'Role Name' with the value 'Sales_Executive', 'This role reports to' with the value 'Sales Representative', and 'Role Name as displayed on reports' which is empty. At the bottom right of the form are three buttons: 'Save', 'Save & New', and 'Cancel'.

Label	Sales Executive
Role Name	Sales_Executive
This role reports to	Sales Representative
Role Name as displayed on reports	

Save Save & New Cancel

- 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 ⓘ

 Upload


Primary Color Hex Value ⓘ



 #0070D2

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

App Launcher Preview



Property Details

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 displays the Salesforce 'Profiles' setup page. The left sidebar shows the navigation menu with 'Setup' and 'Profiles' selected. The main content area is titled 'Profiles' and contains the 'Standard Object Permissions' section. This section includes a table with columns for 'Read', 'Create', 'Edit', 'Delete', 'View All', and 'Modify All' for various objects. The 'Custom Object Permissions' section is also visible, showing permissions for 'Customer' and 'Property' objects. The 'Session Settings' section at the bottom shows 'Session Times Out After' set to '2 hours of inactivity' and 'Session Security Level Required at Login' set to 'None'.

Object	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 Tools	<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 Settings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Points	<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"/>

Object	Read	Create	Edit	Delete	View All	Modify All
Customer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Property	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Session Settings

Session Times Out After: 2 hours of inactivity

Session Security Level Required at Login: 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"/>		
Assets	<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 Texts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Background Operations	<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"/>
Contact Point Type Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D&B Companies	<input type="checkbox"/>					
Data Use Legal Bases	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Use Purposes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Documents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Engagement Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ideas	<input type="checkbox"/>	<input type="checkbox"/>				
Individuals	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Labels	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Locations	<input type="checkbox"/>				<input type="checkbox"/>	
Namespace Registries	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Party Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Push Topics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Scratch Org Infos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Sellers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Streaming Channels	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User External Credentials	<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"/>

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Vehicles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Session Settings

Session Times Out After: 2 hours of inactivity

Session Security Level Required at Login: --None--

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 displays the Salesforce Object Manager interface for the 'User' object. The left sidebar shows the navigation menu with 'Fields & Relationships' selected. The main content area is titled 'User Custom Field Verified' and includes a 'Back to User Fields' link. Below this, the 'Custom Field Definition Detail' section contains tabs for 'Edit', 'Set Field-Level Security', 'View Field Accessibility', and 'Where is this used?'. The 'Field Information' table lists the field details:

Field Label	Verified	Object Name	User
Field Name	Verified	Data Type	Checkbox
API Name	Verified__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	ALAPATI.DEVI	Modified By	ALAPATI.DEVI
	18/11/2024, 1:40 pm		18/11/2024, 1:40 pm

Below the field information, the 'General Options' section shows the 'Default Value' as 'Unchecked'. The 'Validation Rules' section indicates 'No validation rules defined.' and includes a 'New' button. At the bottom, there is a 'Back To Top' link and a note to 'Always show me more records per related list'.

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

The screenshot shows the 'User Edit' page in Salesforce. The page title is 'User Edit' and the user's name is 'Executive'. Below the title are three buttons: 'Save', 'Save & New', and 'Cancel'. The page is divided into two main sections: 'General Information' and 'User Settings'. The 'General Information' section contains fields for First Name, Last Name (Executive), Alias (exec), Email (hellouser1@gmail.com), Username (hellouser12@gmail.com), Nickname (User1731917440072859977), Title, Company, Department, and Division. The 'User Settings' section contains fields for Role (Sales Executive), User License (Salesforce), Profile (System Administrator), Active (checked), Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, Site.com Publisher User, WDC User, and 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 Save Save & New Cancel

General Information Required Information

First Name	<input type="text"/>	Role	<input type="text" value="Sales Manager"/>
Last Name	<input type="text" value="Manager"/>	User License	<input type="text" value="Salesforce Platform"/>
Alias	<input type="text" value="mana"/>	Profile	<input type="text" value="Manager"/>
Email	<input type="text" value="hellouser2@gmail.com"/>	Active	<input checked="" type="checkbox"/>
Username	<input type="text" value="hellouser21@gmail.com"/>	Marketing User	<input type="checkbox"/>
Nickname	<input type="text" value="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	<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"/>
		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 Save Save & New Cancel

General Information Required Information

First Name		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		Knowledge User	<input type="checkbox"/>
Company		Flow User	<input type="checkbox"/>
Department		Service Cloud User	<input type="checkbox"/>
Division		Site.com Contributor User	<input type="checkbox"/>
		Site.com Publisher User	<input type="checkbox"/>
		WDC User	<input type="checkbox"/>
		Data.com User Type	--None--
		Data.com Monthly Addition Limit	300
		Accessibility Mode (Classic Only)	<input type="checkbox"/>
		High-Contrast Palette on Charts	<input 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		Role	Customer
Last Name	Customer2	User License	Salesforce Platform
Alias	cust2	Profile	Customer
Email	hellouser4@gmail.com	Active	<input checked="" type="checkbox"/>
Username	hellouser41@gmail.com	Marketing User	<input type="checkbox"/>
Nickname	User1731917730413946117	Offline User	<input type="checkbox"/>
Title		Knowledge User	<input type="checkbox"/>
Company		Flow User	<input type="checkbox"/>
Department		Service Cloud User	<input type="checkbox"/>
Division		Site.com Contributor User	<input type="checkbox"/>
		Site.com Publisher User	<input type="checkbox"/>
		WDC User	<input type="checkbox"/>
		Data.com User Type	--None--
		Data.com Monthly Addition Limit	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 -
 - a. Location - not equal to - blank,
 - b. 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


- Created By
- Last Modified By
- Verified

Add
Remove

Selected Fields

- Property
- Owner
- Location
- Property Name
- Type

Up
Down



[Click here to view an example](#)

Click Next and Select the initial Submitters >>

- Owner >> Property Owner
- 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"

- Select Object >> Property
- Field to Update >> Verified
- Field Data Type >> CheckBox
- Select CheckBox Option as "True"
- Save.

Add One field Update as "UnVerified Property"

- Select Object >> Property
- Field to Update >> Verified
- Field Data Type >> CheckBox
- Select CheckBox Option as "False"
- Save.

Activate the Approval Process.



SETUP

Approval Processes

Approval Processes

Property: Property Approval

[Back to Approval Process List](#)

Process Definition Detail

Edit

Clone

Delete

Activate

Process Name	Property Approval	Active	<input type="checkbox"/>
Unique Name	Property_Approval	Next Automated Approver Determined By	Manager of Record Submitter
Description			
Entry Criteria	(Property: Location NOTEQUALTO 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	Role: Sales Manager, Property Owner		
Created By	ALAPATI DEVI 18/11/2024, 1:56 pm	Modified By	ALAPATI DEVI 18/11/2024, 2:07 pm

Initial Submission Actions

Add Existing

Add New

Action	Type	Description
Record Lock		Lock the record from being edited

Approval Steps

New Approval Step

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

Final Approval Actions

Add Existing

Add New

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"

Last saved on 17/11/2024, 11:49 am **Active** Run Debug View Tests Save As New Version Save Deactivate

Submit for Approval

*Label
Approval for property

*API Name
Approval_for_property

Description

Submit for Approval
submit-submit

Use values from earlier in the flow to set the inputs for the "Submit for Approval" core action. To use its outputs later in the flow, store them in variables.

Set Input Values for the Selected Action

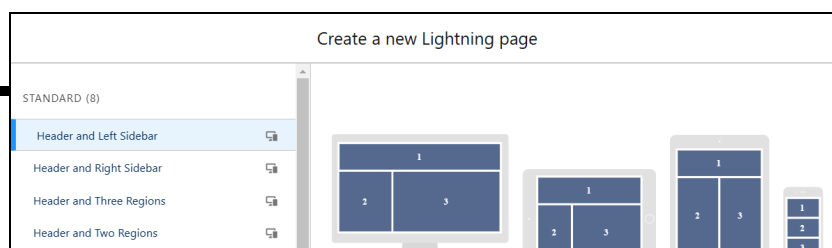
A_a * Record ID
A_a Triggering Property_c > Record ID

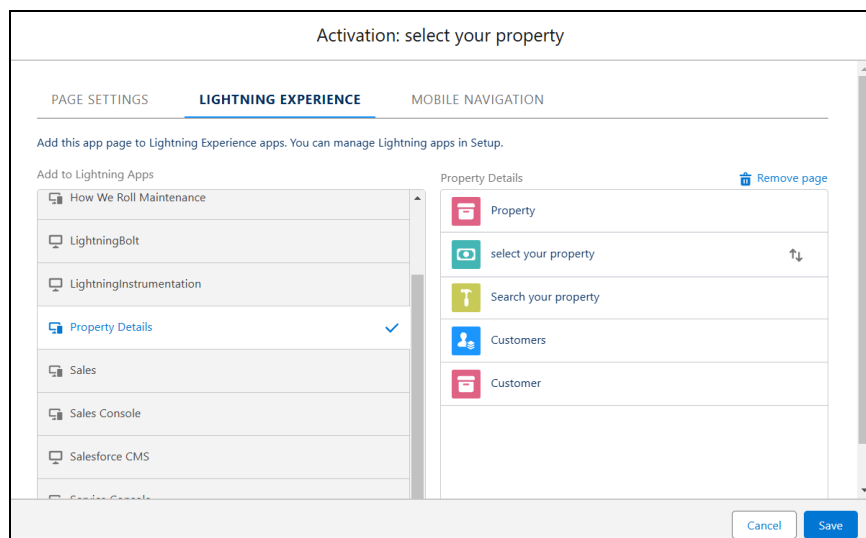
Create an App Page

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 "PropertHandler_LWC".

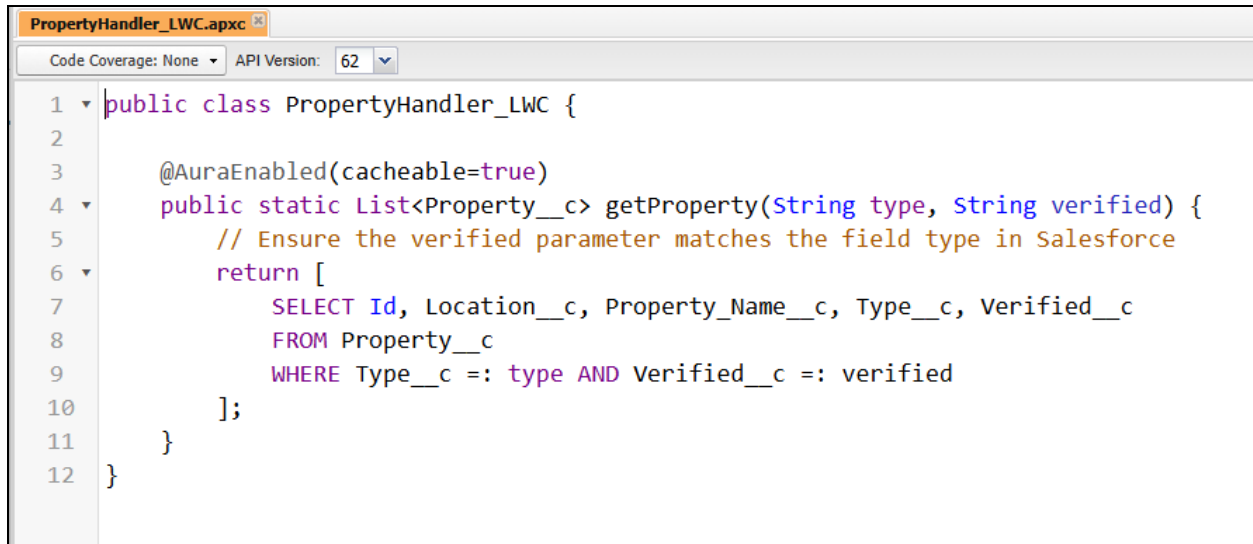
Code:

```
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
];
}
}

```



```

PropertyHandler_LWC.apxc
Code Coverage: None API Version: 62
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 }

```

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

Code:

```

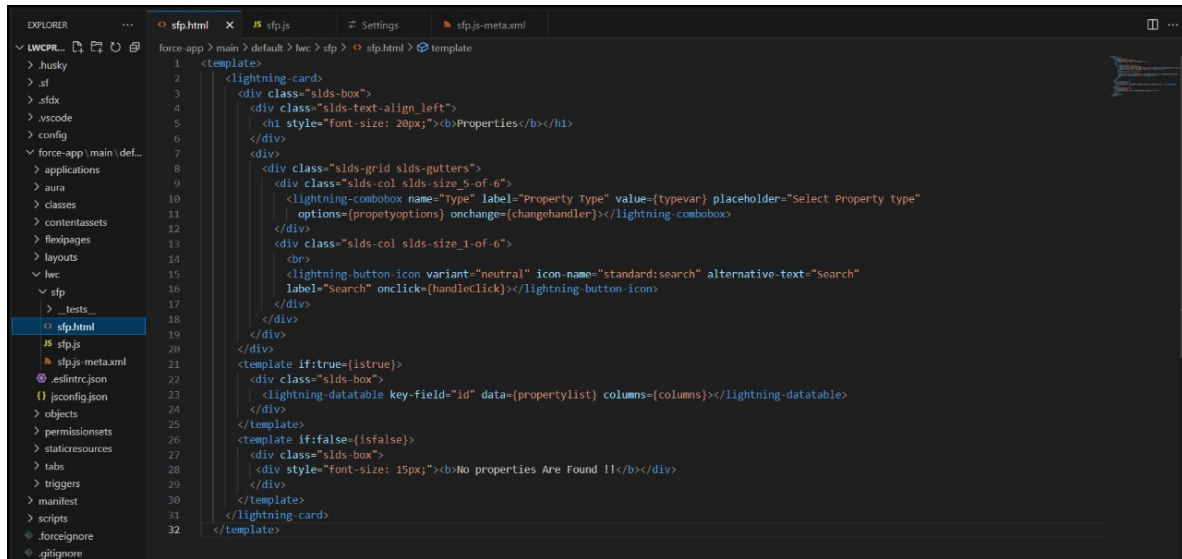
<template>
<lightning-card>
<div class="slds-box">
<div class="slds-text-align_left">
<h1 style="font-size: 20px;"><b>Properties</b></h1>
</div>
<div>
<div class="slds-grid slds-gutters">

```

```

<div class="slds-col slds-size_5-of-6">
  <lightning-combobox name="Type" label="Property Type" value={typevar}
placeholder="Select Property type"
  options={propetyoptions} 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>

```



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;
  istru = 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)
    })
}

```

}

```

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

```

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

Code:

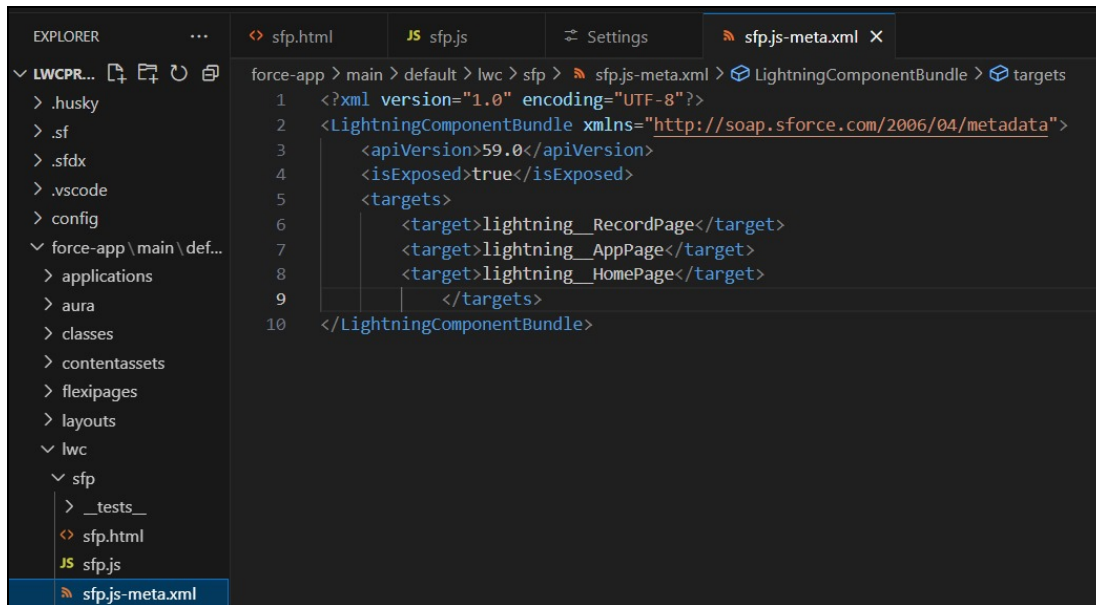
```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
```

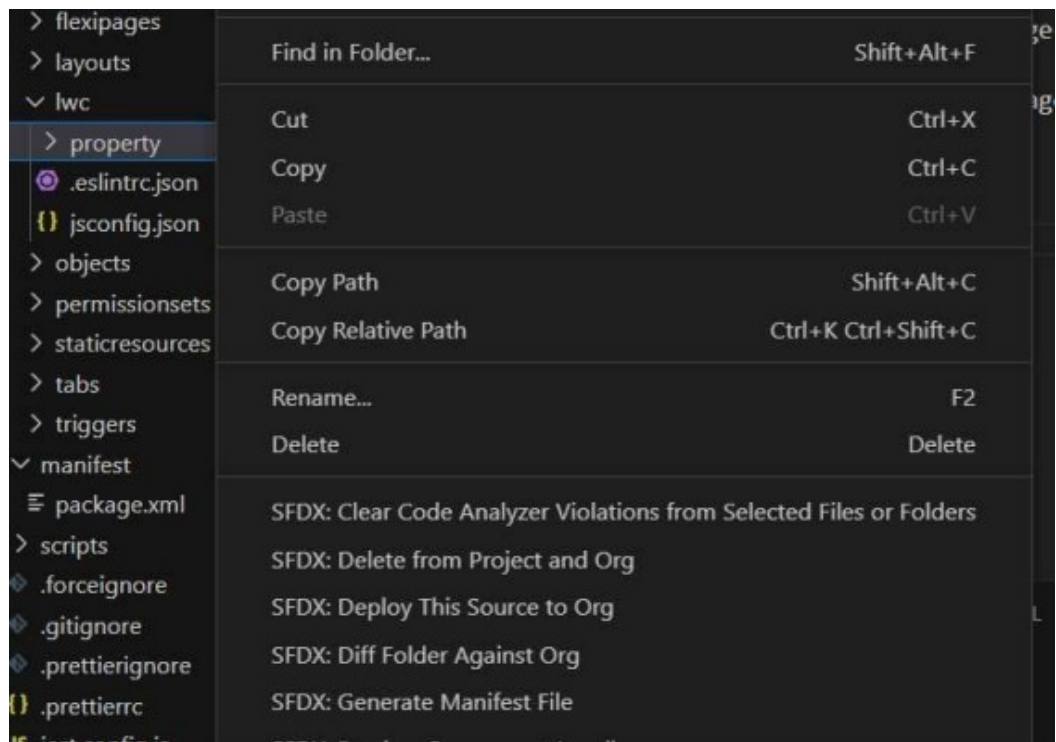
```
<apiVersion>59.0</apiVersion>
```



```
<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.

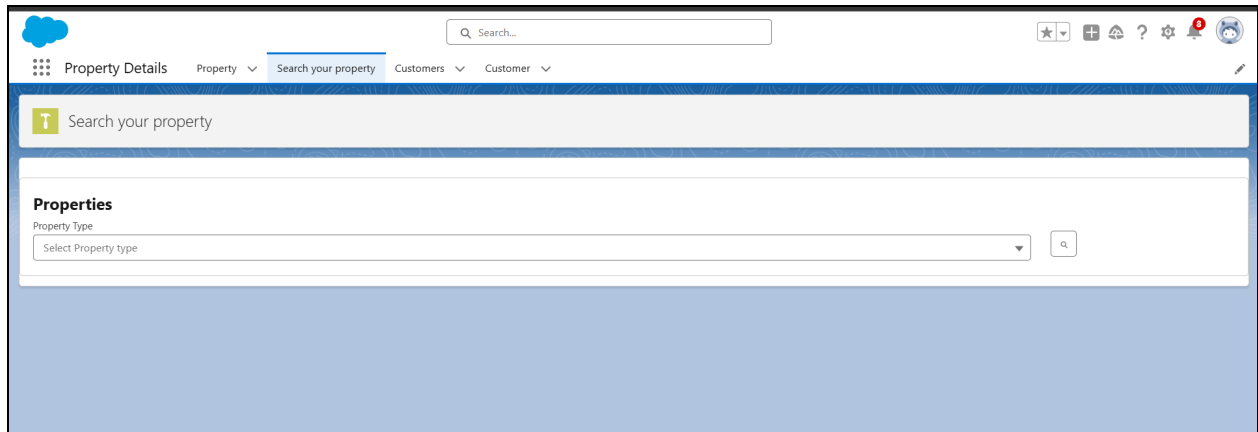


Drag this Component to your App Page

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".

The screenshot shows the Salesforce Setup interface for enabling profile access for the Apex Class "PropertyHandler_LWC". The page title is "Enable Profile Access for Apex Class PropertyHandler_LWC". At the top, there is a "SETUP" button and a "Profiles" header. Below the header, there are "Save" and "Cancel" buttons. The main content area is divided into two columns: "Available Profiles" and "Enabled Profiles". The "Available Profiles" list includes: Analytics Cloud Integration User, Analytics Cloud Security User, Authenticated Website, B2B Reordering Portal Buyer Profile, Contract Manager, Cross Org Data Proxy User, Custom: Marketing Profile, Custom: Sales Profile, Custom: Support Profile, Customer Community Login User, Customer Community Plus Login User, Customer Community Plus User, Customer Community User, and Customer Portal Manager Custom. The "Enabled Profiles" list includes: Customer, Manager, and System Administrator. Between the two lists are "Add" and "Remove" buttons.

Available Profiles	Enabled Profiles
Analytics Cloud Integration User	Customer
Analytics Cloud Security User	Manager
Authenticated Website	System Administrator
B2B Reordering Portal Buyer Profile	
Contract Manager	
Cross Org Data Proxy User	
Custom: Marketing Profile	
Custom: Sales Profile	
Custom: Support Profile	
Customer Community Login User	
Customer Community Plus Login User	
Customer Community Plus User	
Customer Community User	
Customer Portal Manager Custom	