

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/243229403451046>

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

- Name:** Two input fields for 'First Name' and 'Last Name'.
- Phone Number:** Two input fields, each with a placeholder 'P.N. / 10'.
- Email:** One input field with a placeholder 'www@gmail.com'.
- Phone Number:** One input field with a placeholder 'P.N. / 10'.
- Which type of property are you looking for?:** Three radio button options: RESIDENTIAL, COMMERCIAL, and RENTAL.
- Budget Amount:** One input field with a placeholder 'P.N. / 10'.
- Address:** Four input fields for 'Street Address', 'Street Address', 'Street Address Line 2', and 'City'.
- State / Province:** One input field.
- Postal / Zip Code:** One input field.
- Submit:** A green button at the bottom of the form.

The Jotform logo and 'Form Builder' text are visible in the top left. The top right shows 'Dreams World' and 'All changes saved at 6:55 PM'. The bottom of the form has a Jotform logo and a 'Create your own Jotform' button.

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.	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

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

The screenshot shows the Salesforce Setup interface. At the top, there's a search bar labeled 'Search Setup'. Below it, the navigation menu includes 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'Customer' and shows the 'Details' tab selected. The left 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, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The 'Details' section on the right includes fields for Description, API Name (Customer__c), Singular Label (Customer), and Plural Label (Customer). It also has checkboxes for 'Enable Reports', 'Track Activities', and 'Track Field History', all of which are checked. At the bottom, there's a section for 'Deployment Status' showing 'Deployed' and a link to 'Help Settings'.

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 pk	Commercial	Amravati	unchecked
3 Bhk fkat at st	rental	Jubliee hill Hyd	Checked

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

The screenshot shows the Salesforce Setup interface. At the top, there's a search bar labeled "Search Setup". Below it, the navigation menu includes "Setup", "Home", and "Object Manager". The main content area is titled "SETUP > OBJECT MANAGER" and "Property". On the left, a sidebar lists various configuration options under "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 "Record Types" option is selected. The main area displays the "Details" for the "Property" object. It includes fields for "API Name" (Property_c), "Custom" (checked), "Singular Label" (Property), "Plural Label" (Property), "Enable Reports" (checked), "Track Activities" (checked), "Track Field History" (checked), "Deployment Status" (Deployed), and "Help Settings" (Standard salesforce.com Help Window). There are "Edit" and "Delete" buttons in the top right corner.

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".

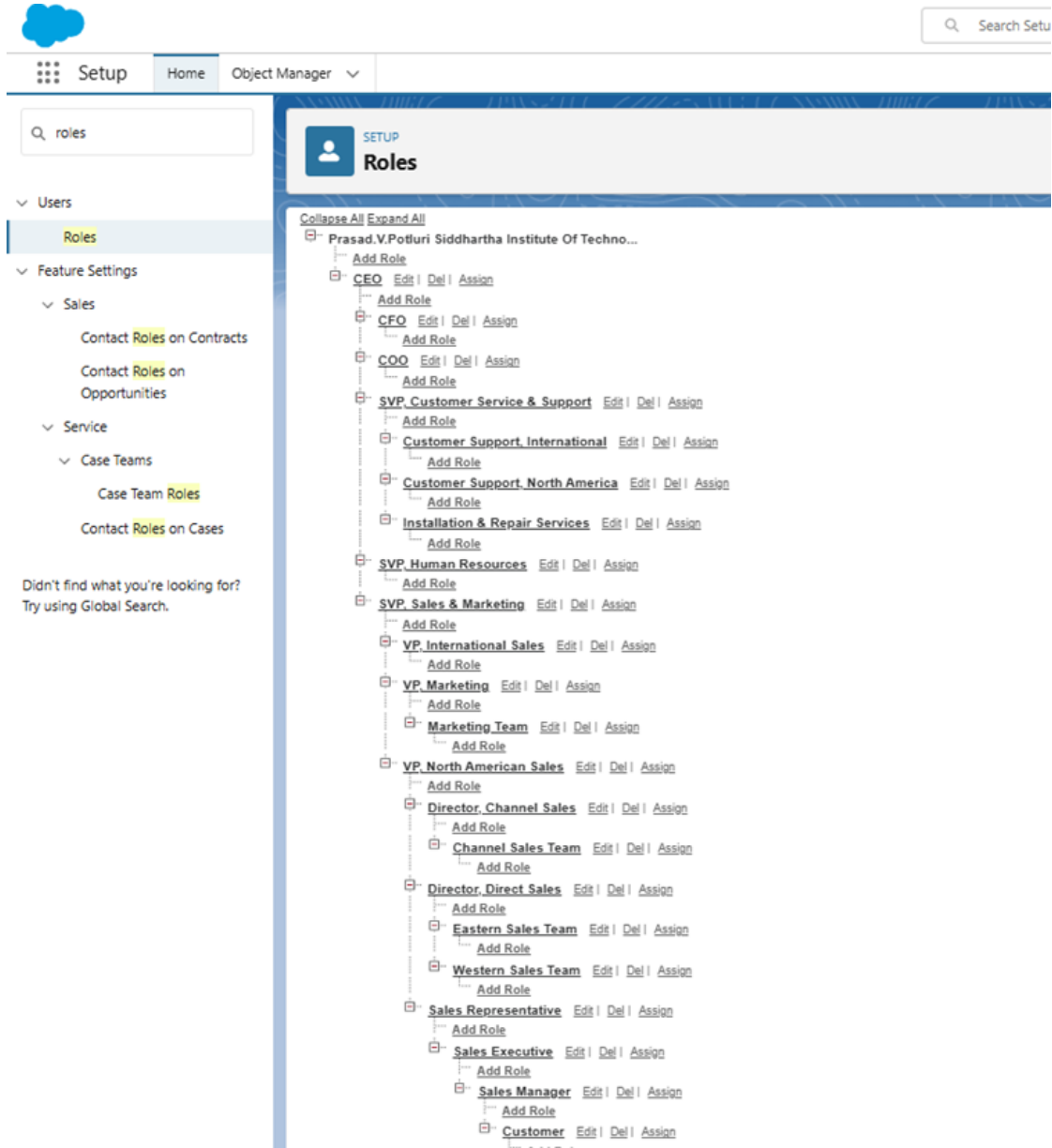
Then "Save the Integration" and "Finish".

Create Roles

Here we need to Create Roles as per business requirementment

Activity :- 1

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



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

Role Name



This role reports to



Role Name as displayed on reports

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 ⓘ
Property Details

* Developer Name ⓘ
Property_Details

Description ⓘ
Enter a description...

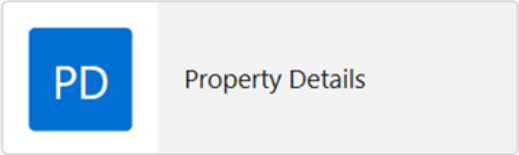
App Branding

Image ⓘ


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



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.

Setup Profiles

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. [Click to Learn More](#)

Object Name	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 Search Page	<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 Rows	<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 Testings	<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"/>

Object Name	Read	Create	Edit	Delete	View All	Modify All
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"/>
DSB Companies	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Use Legal Rows	<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"/>
Identities	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<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"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nonresponse Registrations	<input type="checkbox"/>	<input type="checkbox"/>	<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"/>	<input type="checkbox"/>	<input type="checkbox"/>
Scratch Org Infos	<input type="checkbox"/>	<input type="checkbox"/>	<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

Object Name	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".

The screenshot displays the Salesforce Profile Setup interface for a profile named "Manager". It is divided into three main sections: Standard Object Permissions, Custom Object Permissions, and Session Settings.

Standard Object Permissions: This section contains two tables. The left table lists standard objects with checkboxes for Read, Create, Edit, Delete, View All, and Modify All. The right table lists standard objects with checkboxes for Read, Create, Edit, Delete, View All, and Modify All. In the left table, "Accounts" has "View All" checked. In the right table, "Contact Point Type Consents" has "View All" checked.

Custom Object Permissions: This section contains two tables. The left table lists custom objects with checkboxes for Read, Create, Edit, Delete, View All, and Modify All. The right table lists custom objects with checkboxes for Read, Create, Edit, Delete, View All, and Modify All. In the left table, "Customer" and "Property" have "Modify All" checked. In the right table, "Vehicles" has "View All" checked.

Session Settings: This section contains a table with checkboxes for Session Timeout, Password Policy, and Session Termination. The "Session Timeout" checkbox is checked.

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"

SETUP > OBJECT MANAGER

User

Details

Fields & Relationships

User Page Layouts

User Profile Page Layouts

Lightning Record Pages

Buttons and Links

Compact Layouts

Field Sets

Object Limits

Related Lookup Filters

Search Layouts

List View Button Layout

Triggers

User Custom Field

Verified

Back to User Fields

Validation Rules (0)

Custom Field Definition Detail

Edit

Set Field Level Security

View Field Accessibility

Where is this used?

Field Information

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	ANITHA BONDU, 19/11/2024, 11:18 am	Modified By	ANITHA BONDU, 19/11/2024, 11:18 am

General Options

Default Value	Unchecked
---------------	-----------

Validation Rules

New

No validation rules defined.

Validation Rules Help

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 for a user named 'Executive'. The page has a header with 'SETUP Users' and a 'Help for this Page' link. Below the header, there are buttons for 'Save', 'Save & New', and 'Cancel'. The main section is titled 'General Information' and contains two columns of fields. The left column includes 'First Name', 'Last Name' (filled with 'Executive'), 'Alias' (filled with 'exec'), 'Email' (filled with 'kol@abc.com'), 'Username' (filled with 'kol@abc.com'), 'Nickname' (filled with 'User173199534395650668'), 'Title', 'Company', 'Department', and 'Division'. The right column includes 'Role' (filled with 'Sales Executive'), 'User License' (filled with 'Salesforce'), 'Profile' (filled with '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', 'Data.com User Type' (filled with '--None--'), 'Data.com Monthly Addition Limit' (filled with '300'), and 'Accessibility Mode (Classic Only)'. A red exclamation mark icon indicates required information.

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

Save

Save & New

Cancel

General Information

First Name

Last Name

Manager

Alias

mana

Email

sed@wer.com

Username

sed@wer.com

Nickname

User173199548161790588

Title

Company

Department

Division

Role

Sales Manager

User License

Salesforce Platform

Profile

Manager

Active

☒

Marketing User

☐

Offline User

☐

Knowledge User

☐

Flow User

☐

Service Cloud User

☐

Site.com Contributor User

☐

Site.com Publisher User

☐

WDC User

☐

Data.com User Type

--None--

Data.com Monthly Addition Limit

300

Accessibility Mode (Classic Only)

☐

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

Save

Save & New

Cancel

General Information

First Name

Last Name

Customer

Alias

cust

Email

cus@kuy.com

Username

cus@kuy.com

Nickname

User173199556587343834

Title

Company

Department

Division

Role

Customer

User License

Salesforce Platform

Profile

Customer

Active

☒

Marketing User

☐

Offline User

☐

Knowledge User

☐

Flow User

☐

Service Cloud User

☐

Site.com Contributor User

☐

Site.com Publisher User

☐

WDC User

☐

Data.com User Type

--None--

Data.com Monthly Addition Limit

300

Accessibility Mode (Classic Only)

☐

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

General Information ! = Required Information

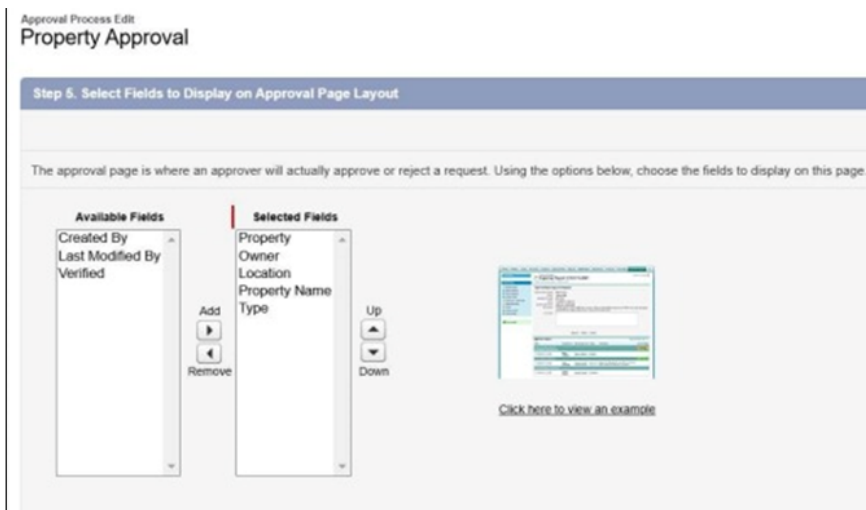
First Name	<input type="text"/>	Role	<input type="text" value="Customer"/>
Last Name	<input type="text" value="Customer2"/>	User License	<input type="text" value="Salesforce Platform"/>
Alias	<input type="text" value="cust"/>	Profile	<input type="text" value="Customer"/>
Email	<input type="text" value="dfg@wer.com"/>	Active	<input checked="" type="checkbox"/>
Username	<input type="text" value="dfg@wer.com"/>	Marketing User	<input type="checkbox"/>
Nickname	<input type="text" value="User173199569236094129"/>	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"/>

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.



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.


Approval Processes

Approval Processes

Property: Property Approval

[Back to Approval Process List](#)
[Help for this Page](#)

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	(Property: Location NOT EQUAL TO blank) AND (Property: Verified EQUALS False)		
Entry Criteria	(Property: Location NOT EQUAL 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	ANITHA BONDU, 19/11/2024, 11:30 am	Modified By	ANITHA BONDU, 19/11/2024, 11:34 am

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

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

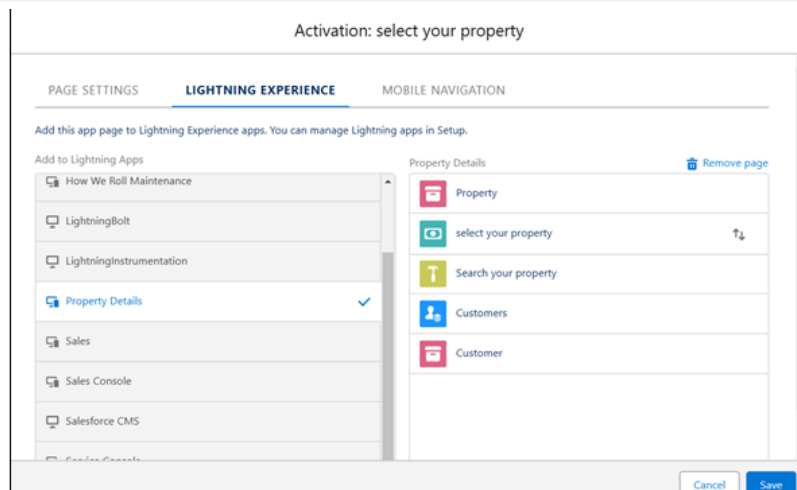
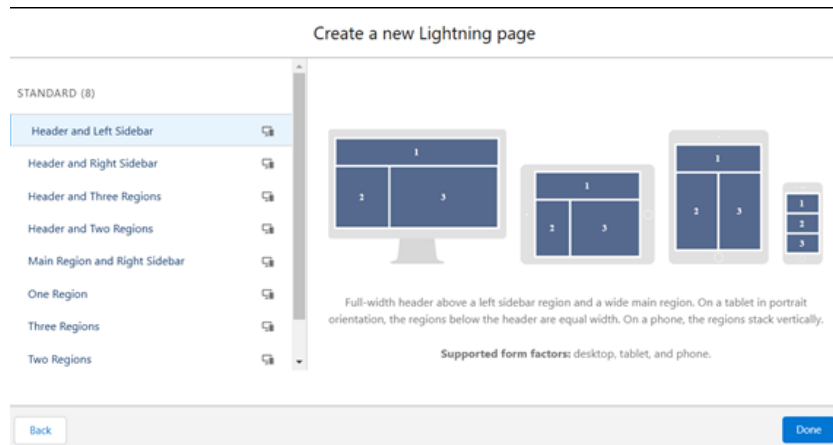
A_a * Record ID ⓘ
A_a Triggering Property__c > Record ID ×

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

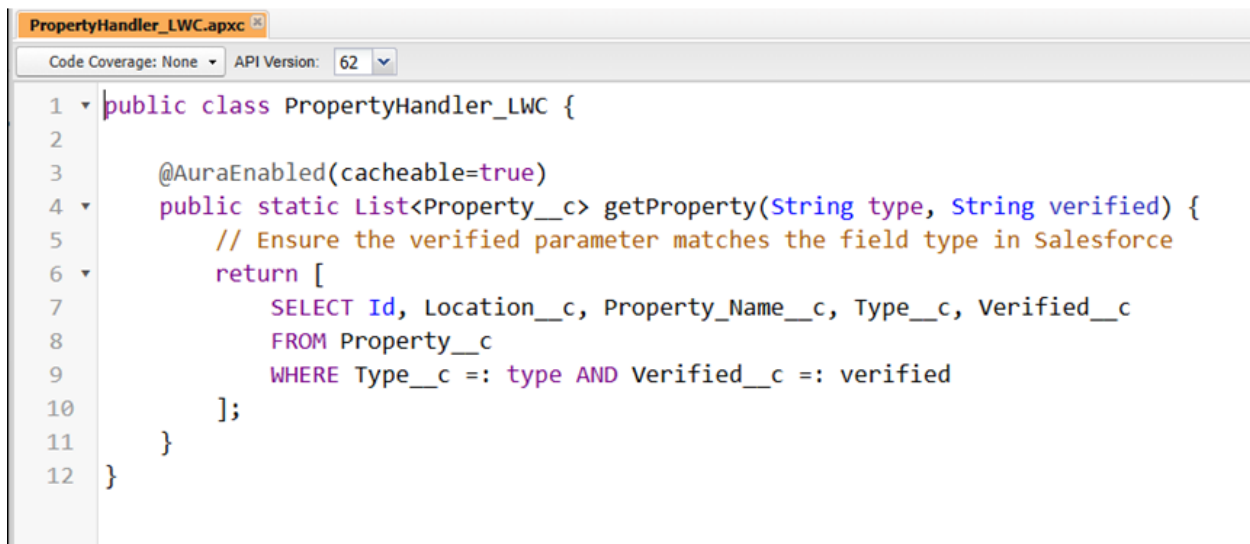
Create an LWC Component for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on "Search your Property Page"

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 }
```

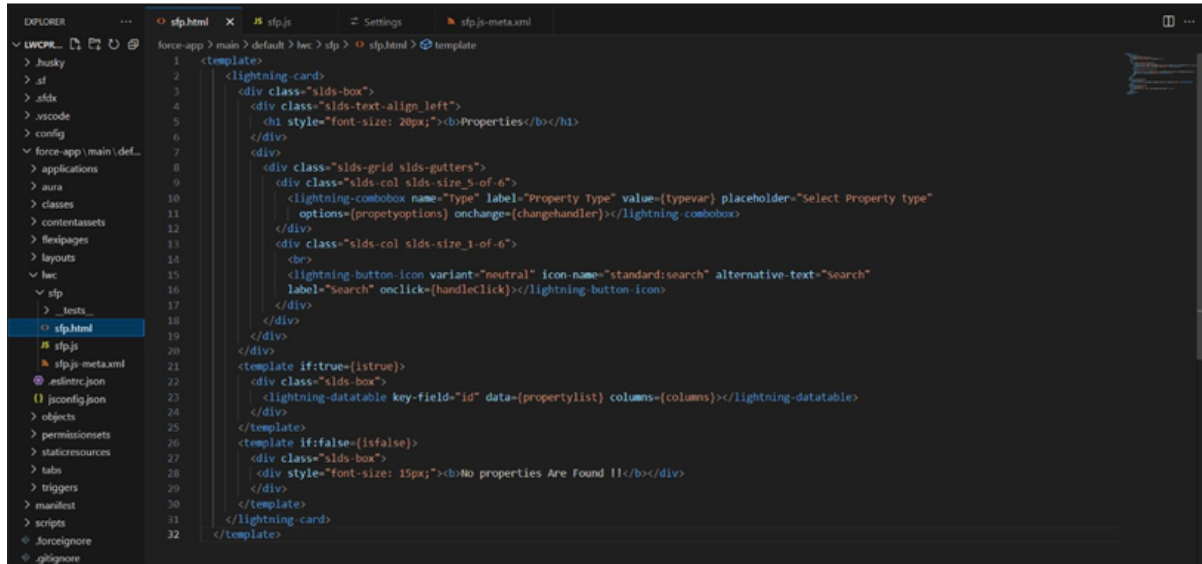
1. Create a Lightning Web Component in your VsCode, and (ctrl+shift +P) and click on authorize an org.
2. Enter your login ic and password to authorize your org.
3. Now (ctrl+shift +P) and Create a lightning Web Component and Name it Anything you want to.
4. In your Html File, write this code

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={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>
      <div>
        <template if:true={isttrue}>
          <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>
      </div>
    </lightning-card>
  </template>

```



1. 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" }
    ]
    propetyoptions = [
        { label: "Commercial", value: "Commercial" },
        { label: "Residential", value: "Residential" },
        { label: "rental", value: "rental" }
    ]
    @wire(getRecord, { recordId: "$userId", fields: ['User.Verified__c'] })
```

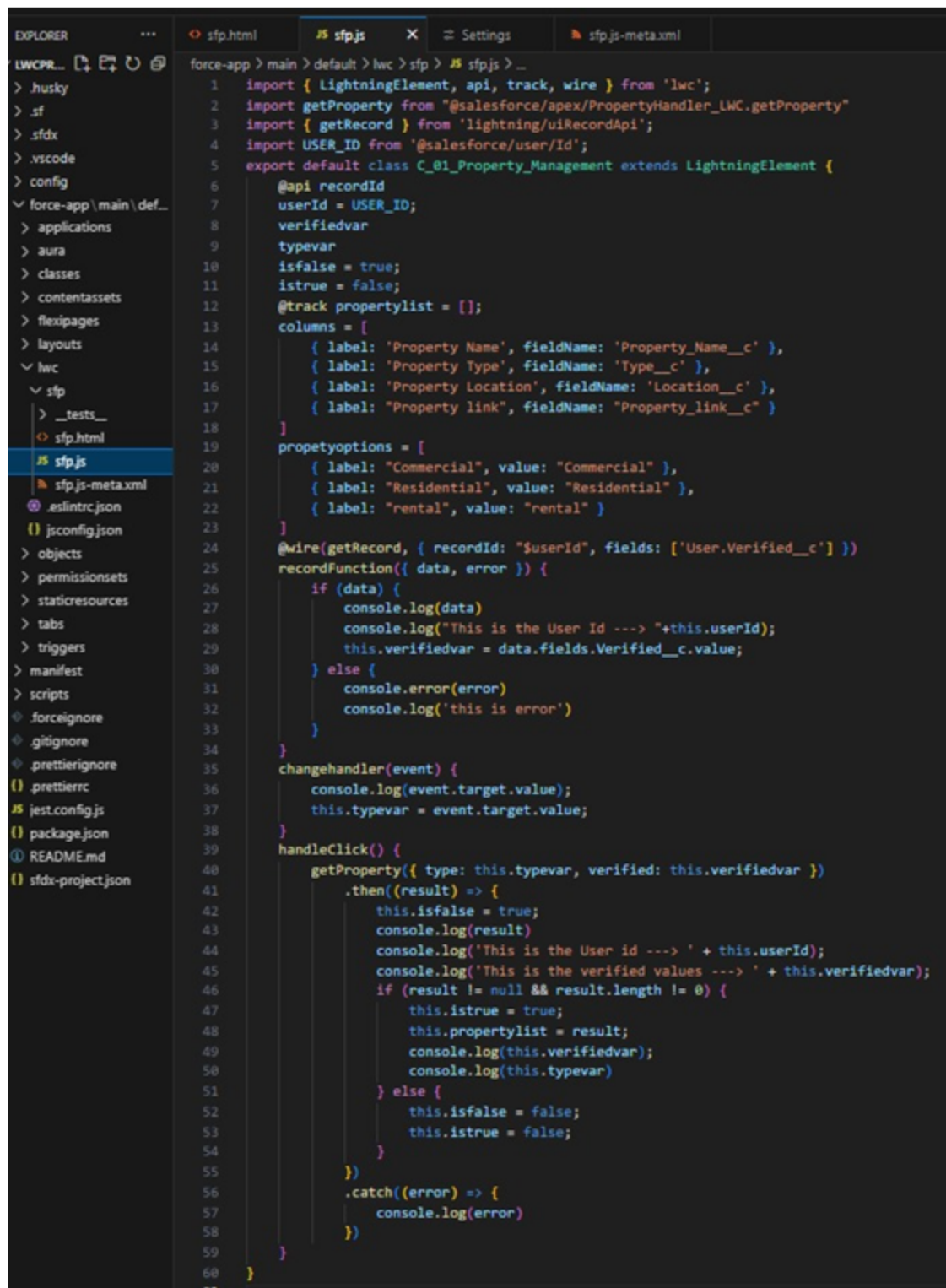
```

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     @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    propertyoptions = [
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 }
```

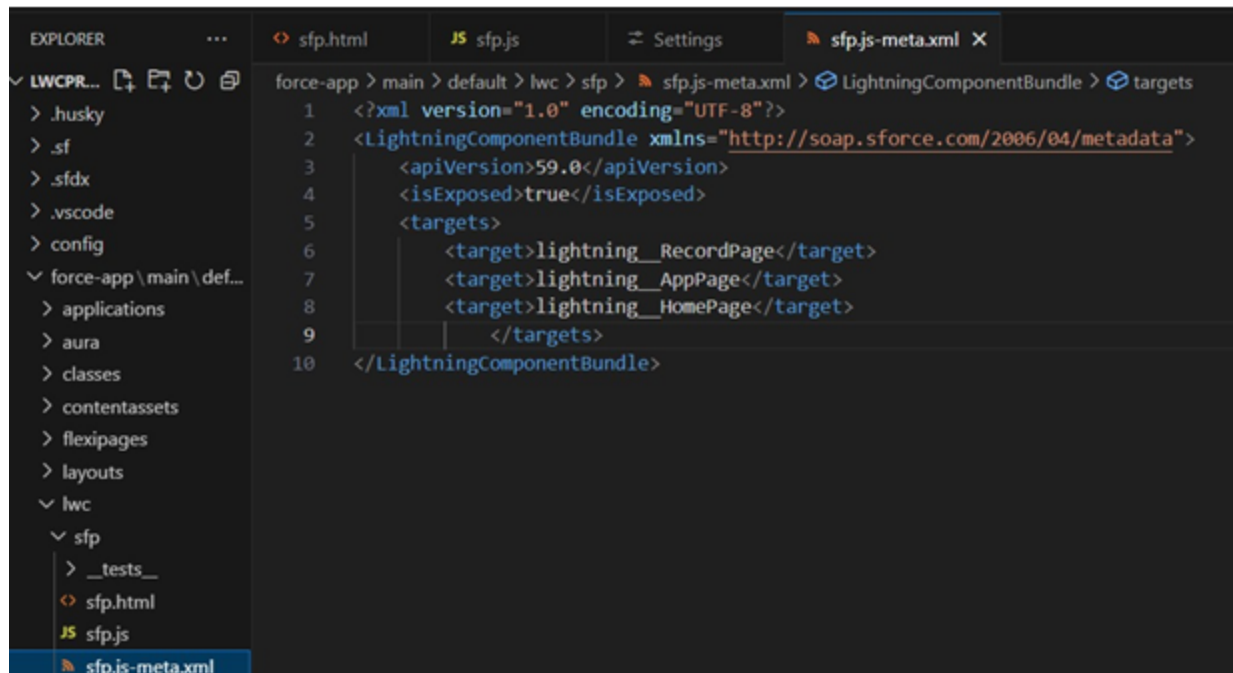
7. In your Metafile, give your targets to deploy the component.

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.

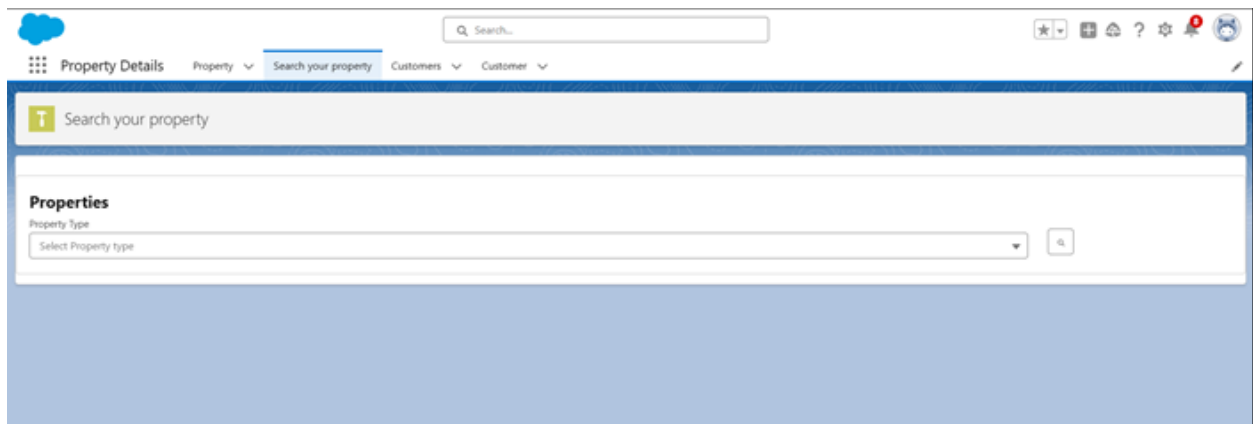
> flexipages	Find in Folder...	Shift+Alt+F
> layouts		
✓ lwc	Cut	Ctrl+X
> property	Copy	Ctrl+C
🔗 .eslintrc.json	Paste	Ctrl+V
{ } jsconfig.json		
> objects	Copy Path	Shift+Alt+C
> permissionsets	Copy Relative Path	Ctrl+K Ctrl+Shift+C
> staticresources		
> tabs	Rename...	F2
> triggers	Delete	Delete
✓ manifest		
≡ package.xml	SFDX: Clear Code Analyzer Violations from Selected Files or Folders	
✓ scripts	SFDX: Delete from Project and Org	
• .forceignore	SFDX: Deploy This Source to Org	
• .gitignore	SFDX: Diff Folder Against Org	
• .prettierignore	SFDX: Generate Manifest File	
• .prettierrc		

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".

SETUP
Profiles

Enable Profile Access for Apex Class
PropertyHandler_LWC

Available Profiles

- 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
- Customer Portal Manager Custom

Enabled Profiles

- Customer
- Manager
- System Administrator

Add
Remove

Save Cancel