

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
 - a. Add, update, and delete client details.
 - b. Track client preferences, budget, and location interests.
 - c. Maintain contact details and communication history.
2. Property Management
 - a. Manage property listings with details like type, price, location, and features.
 - b. Track properties available for sale, rent, or lease.
 - c. Upload photos and documents for properties.
3. Requirement Matching
 - a. Match client requirements with available properties using filters.
 - b. Notify clients about new properties that fit their criteria.
4. Lead Tracking
 - a. Manage inquiries and follow up with potential clients.
 - b. Schedule meetings and site visits.
 - c. Assign leads to specific team members.

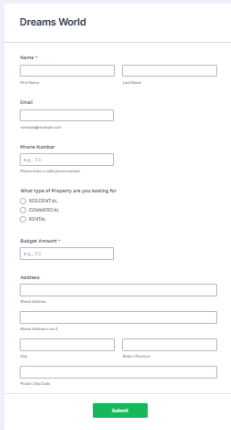
Milestone 1: Create a Jotform and integrate it with theorg to create a record of customers automatically.

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.

Activity1

Open your browser and search for jotform and log in.

1. After login click on create form and click on start from scratch
2. Now create a form to get the customer details like Name, Phone, Email, Address and type of property the customer is interested in.
3. Once the form is created, publish it by clicking on publish.
4. form link :- <https://form.jotform.com/243223483079054>



The screenshot shows a Jotform titled "Dreams World" with the following fields and options:

- Name ***: Two text input fields for First Name and Last Name.
- Email ***: A text input field with a placeholder "example@gmail.com".
- Phone Number ***: A text input field with a placeholder "e.g., 123 456 7890123".
- What type of property are you looking for?**: Three radio button options: ☐ RESIDENTIAL, ☐ COMMERCIAL, and ☐ MIXED.
- Budget (Amount) ***: A text input field with a placeholder "e.g., 123".
- Address ***: A text input field.
- City, State, Zip ***: Three text input fields for City, State, and Zip.
- Submit**: A green button at the bottom.

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	Emial	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	6800014	unchecked
Prajwal	454545	prajwal@gmail	Maharashtra	Rental	25000	kamdli	kathora	Amravati	444805	checked

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 and navigation tabs for Setup, Home, and Object Manager. The 'Object Manager' tab is selected, and the 'Customer' object is chosen. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The 'Fields & Relationships' section is active, displaying a table of 15 fields for the 'Customer' object. The table has columns for Field Label, Field Name, Data Type, Controlling Field, and Indexed status. Fields include Budget Amount, City, Created By, Customer, Name, Email, Last Modified By, Owner, and Phone Number.


FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Budget Amount	Budget_Amount__c	Number(18, 0)		
City	City__c	Text(255)		
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Text(255)		
Customer	Name	Text(80)		✓
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone Number	Phone_Number__c	Number(18, 0)		

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	Jubliee hill Hyd	Checked

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



Search Setup



Setup

Home

Object Manager

SETUP > OBJECT MANAGER

Property

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

Restriction Rules

Fields & Relationships

8 Items, Sorted by Field Label





Q, Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Location	Location__c	Text(255)		
Owner	OwnerId	Lookup(User,Group)		✓
Property	Name	Text(80)		✓
Property Name	Property_Name__c	Text(255)		
Type	Type__c	Text(255)		
Verified	Verified__c	Text(255)		

Integrate Jotform with Salesforce Platform

In this Milestone we are going to integrate jotform with Salesforce

Activity

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 'Form Builder' interface with the 'SALESFORCE' integration selected. The left sidebar contains navigation options: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (selected), WORKFLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The main area is titled 'SALESFORCE' and shows the 'Select a Salesforce Object' dropdown set to 'Customer'. Below this, the 'Create a record' section is active, displaying a table to map form fields to Salesforce fields. The table has two columns: 'Object Fields' and 'Form'. The 'Object Fields' column lists fields from the 'Customer' object, and the 'Form' column lists fields from the current Jotform. The mapping is as follows:

Object Fields	Form
Customer__c	Name - First Name
City	Address - City
Budget Amount	Budget Amount
Phone Number	Phone Number
Property Type	which type of Property are you looking...
Street Address	Address - Street Address
Name	Name - Last Name
State	Address - Street Address 2
Street Address Line 2	Address - Street Address 2

At the bottom, there is an 'Update an existing record' section with a toggle switch set to 'OFF'. The top of the interface shows the Jotform logo, 'Form Builder', and a 'Form' tab with a timestamp 'Last edited at 10:29 AM'. The top right has links for 'Add Collaborators', 'Help', and a user profile icon. The bottom right has a 'Give Feedback' button.

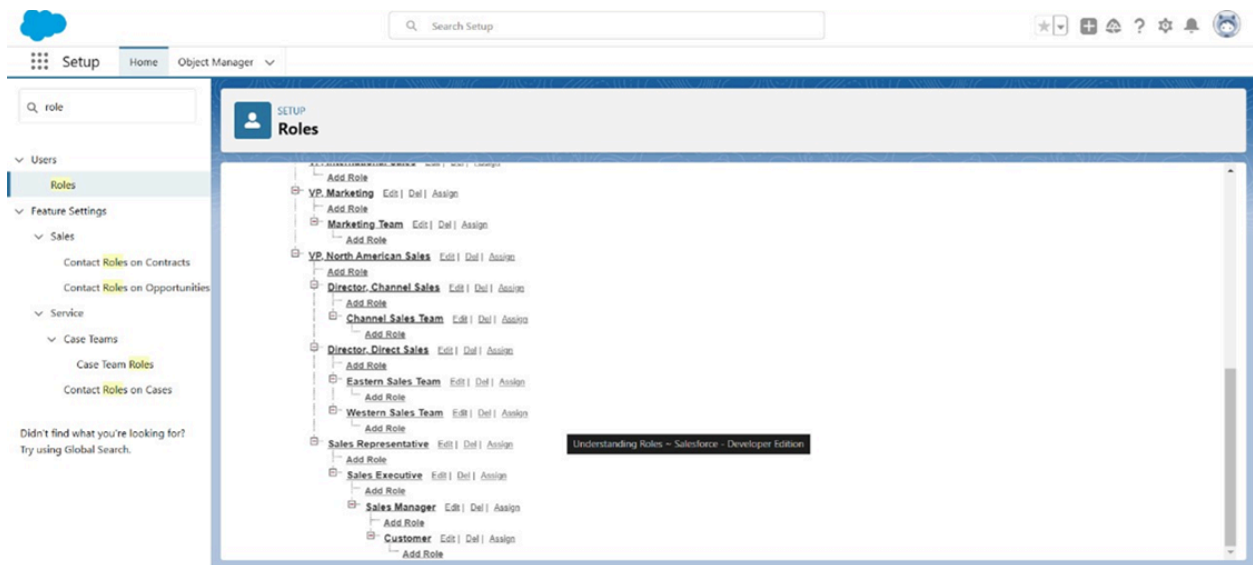
Then “Save the Integration” and “Finish”.

The screenshot shows the 'All Actions' screen for the Salesforce integration. At the top, there is a 'SALESFORCE' header with the text 'Send new leads, contacts, or accounts to your sales CRM'. Below this, there are two buttons: 'See Action Logs' and '+ Add New Action'. The main area displays a list of actions, with the first action being '1 Create or update a record Customer'. The action is shown in a light blue box with a white border.

Create Roles

here we need to Create Roles as per business requirement

Activity:- 1



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

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 AppManager 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.

New Lightning App

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 ⓘ

 #AAE420

Org Theme Options

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

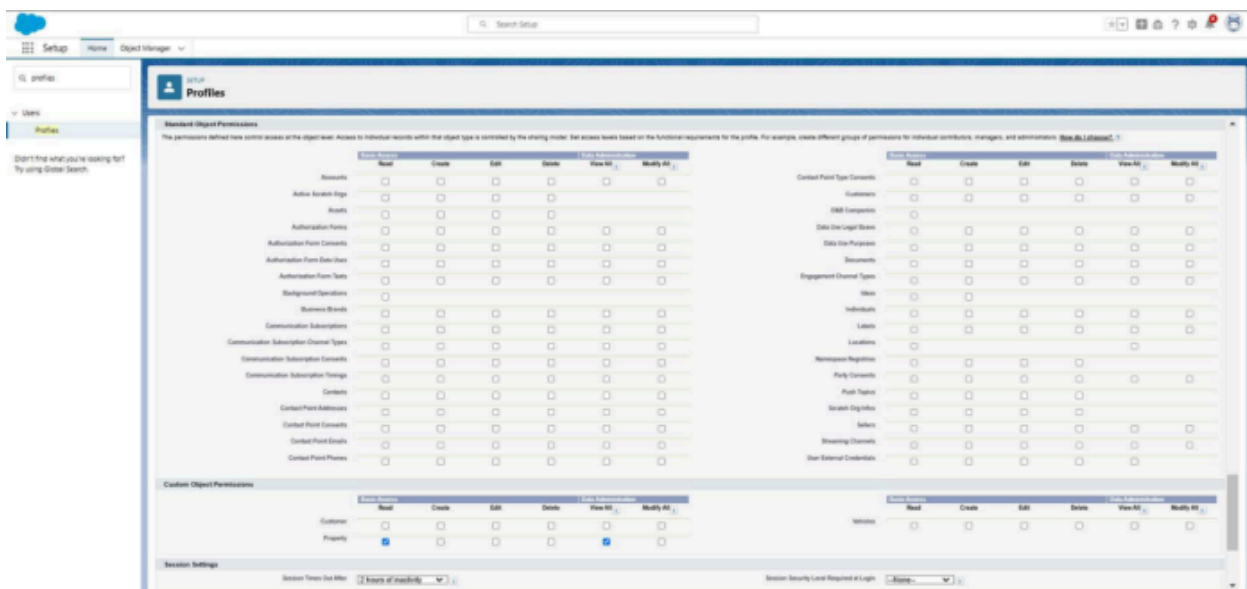
App Launcher Preview

Create Profiles

Create profiles as per business requirement

Creating Customer Profiles

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. so 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



Creating Manager Profiles :-

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 shows the Salesforce Setup interface for the 'Profiles' page. The left sidebar contains a navigation menu with the following items: Hyperforce Assistant, Users, Profiles (selected), Data, Mass Transfer Approval Requests, Feature Settings, Data.com, Prospect Preferences, Prospect Users, Functions, Marketing, Lead Processes, Sales, Products, Asset Settings, and Product Schedules. The main content area is titled 'SETUP Profiles' and contains the following sections:

- Custom Object Permissions:** A table with columns for 'Basic Access' (Read, Create, Edit, Delete) and 'Data Administration' (View All, Modify All). The table is currently empty.
- Session Settings:** A section with two settings: 'Session Times Out After' set to '2 hours of inactivity' and 'Session Security Level Required at Login' set to 'None'.
- Password Policies:** A section with two settings: 'User passwords expire in' set to '90 days' and 'Enforce password history' set to '3 passwords remembered'.

Create a CheckBox 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 shows the 'Verified' custom field configuration. The field is a checkbox type, and its API name is 'Verified__c'. The 'Field Information' section lists various details, and the 'General Options' section shows the default value as 'Unchecked'. The 'Validation Rules' section indicates that no validation rules are defined.

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](#)

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	21501A05F5 SANGAM HEMANTH KUMAR, 19/11/2024, 9:22 pm	Modified By	21501A05F6 SANGAM HEMANTH KUMAR, 19/11/2024, 9:22 pm

General Options

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

Validation Rules [New](#) [Validation Rules Help](#)

No validation rules defined.

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

User : 2


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

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 checkbox is "Unchecked"
7. Save

User : 4

1. Goto 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

 **SETUP**
Users

All Users

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. [Let's Go](#)

View: All Users [Edit](#) [Create New View](#)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | **All**

<input type="checkbox"/> Action	Full Name ↑	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatty.00dns000006vksh2aq_dun3pvwsjk9@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	Customer	cust	cost123@gmail.com	Customer	✓	Customer
<input type="checkbox"/> Edit	Customer2	cust	cust1234@gmail.com	Customer	✓	Customer
<input type="checkbox"/> Edit	Executive	exec	hemu123@gmail.com	Sales Executive	✓	System Administrator
<input type="checkbox"/> Edit	Manager	mana	somu123@gmail.com	Sales Manager	✓	Manager
<input type="checkbox"/> Edit	SANGAM,HEMANTH KUMAR, 21501A05F6	2SANG	21501a05f6@creative-raccoon-mxskbc.com		✓	System Administrator
<input type="checkbox"/> Edit	User_Integration	integ	integration@00dns000006vksh2aq.com		✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User_Security	sec	insightssecurity@00dns000006vksh2aq.com		✓	Analytics Cloud Security User

[New User](#) [Reset Password\(s\)](#) [Add Multiple Users](#)

Create an ApprovalProcess for PropertyObject

An Approval process to approve or reject the records as according

Activity1

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 -
5. Location- is not equal to- blank,
6. Verified- Equals- false
7. Click next and "Next Automated Approver Determined By" Select Manager
8. From Record Editability Properties >> Click on Administrator so the currently assigned approver can edit records during the approval process.
9. From Step 5. Select Fields to Display on Approval Page Layout select Property, Owner, Location, Type.

1. Click Next and Select the initial Submitters »
2. Owner >> Property Owner
3. Roles >> Sales Manager
4. 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**

Property

Approvals are complex business processes that require information gathering and planning before implementing. It is recommended that you follow the instructions below before getting started.

1. [Read the help topic](#)
2. [View the checklist](#)
3. [Create a custom user hierarchical relationship field](#)
4. [Create email templates](#)
5. Create an approval process using either the Jump Start or Standard Wizard
6. Add Approval History Related List to all page layouts
7. Activate the process to deploy to your users

Manage Approval Processes For: Property

A listing of both active and inactive approval processes for **Property** is displayed below. To create a new approval process, click Create New Approval Process then select Use Jump Start Wizard to set up your approval process in a few short steps. Or, select Use Standard Wizard to configure all approval options.

Create New Approval Process

Active Approval Processes Reorder

Action	Process Order	Approval Process Name	Description
Edit Deactivate	<input type="text" value="1"/>	Property Approval	

Inactive Approval Processes

No approval processes available

Create a Recordtriggerer ñow to submit the Approval Process Automatically

A flow that can submit the recordsdirectly for approval

Activity1

- FromSetup>>Search forFlows>>Click OnNew andSelect “Record Trigger Flow”.
- SelectObject>>Property
- Select“Trigger the flowwhen”>> “A record is created”
- SetEntry Conditions>> “None”
- Add a“Action”>> “Submit for Approval”
- Give Label>> Approval forproperty
- Record Id>> (!SRecord.Id)
- Done

Save the Flow and Give label as “Property Approval” and “Activate”

Create an App Page

The screenshot displays the Salesforce Flow Builder interface for a 'Record-Triggered Flow' named 'Property Approval - V1'. The flow is configured with the following steps:

- Start** (Record-Triggered Flow)
- Object: Property** (Trigger: A record is created)
- Optimize for: Actions and Related Records**
- Run Immediately** (Action)
- Approval for property** (Action)
- End** (Action)

The right-hand pane shows the 'Configure Start' section, which includes the 'Select Object' dropdown set to 'Property' and the 'Configure Trigger' section. The 'Trigger the Flow When' options are:

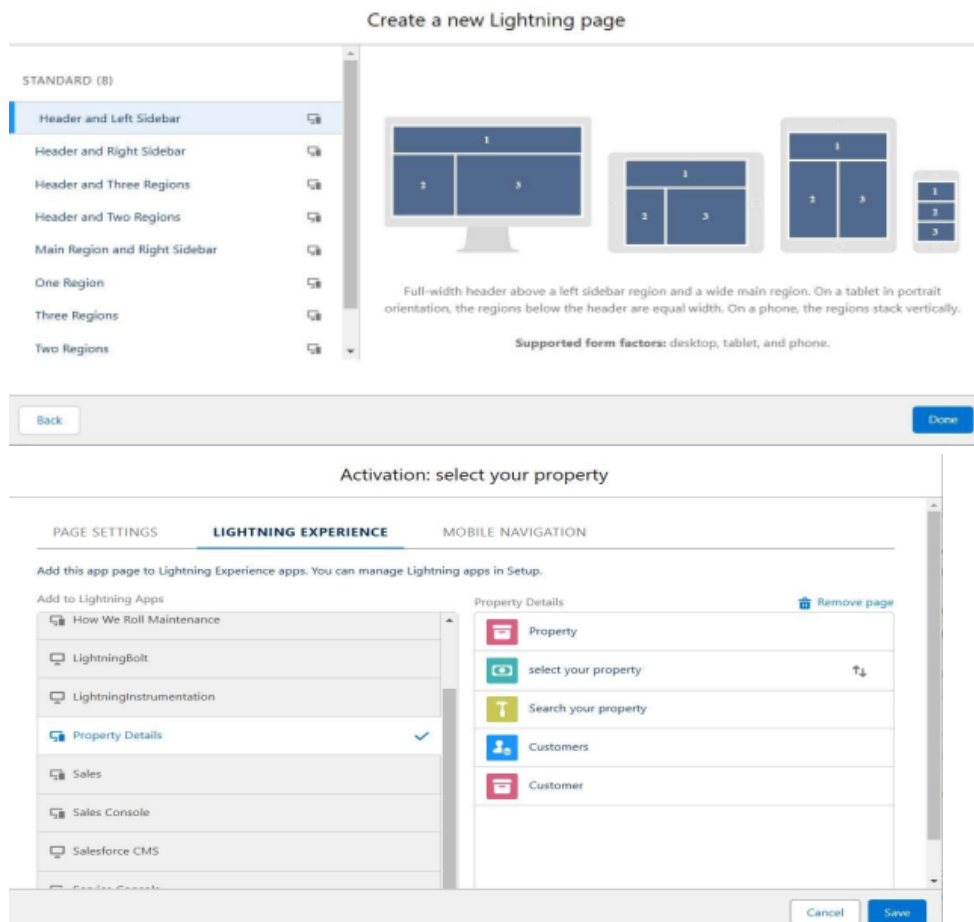
- ☒ A record is created
- ☐ A record is updated
- ☐ A record is created or updated
- ☐ A record is deleted

The 'Set Entry Conditions' section is also visible, with a note: 'Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.' The 'Condition Requirements' dropdown is set to 'None'.

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

Activity1

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 Settings 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

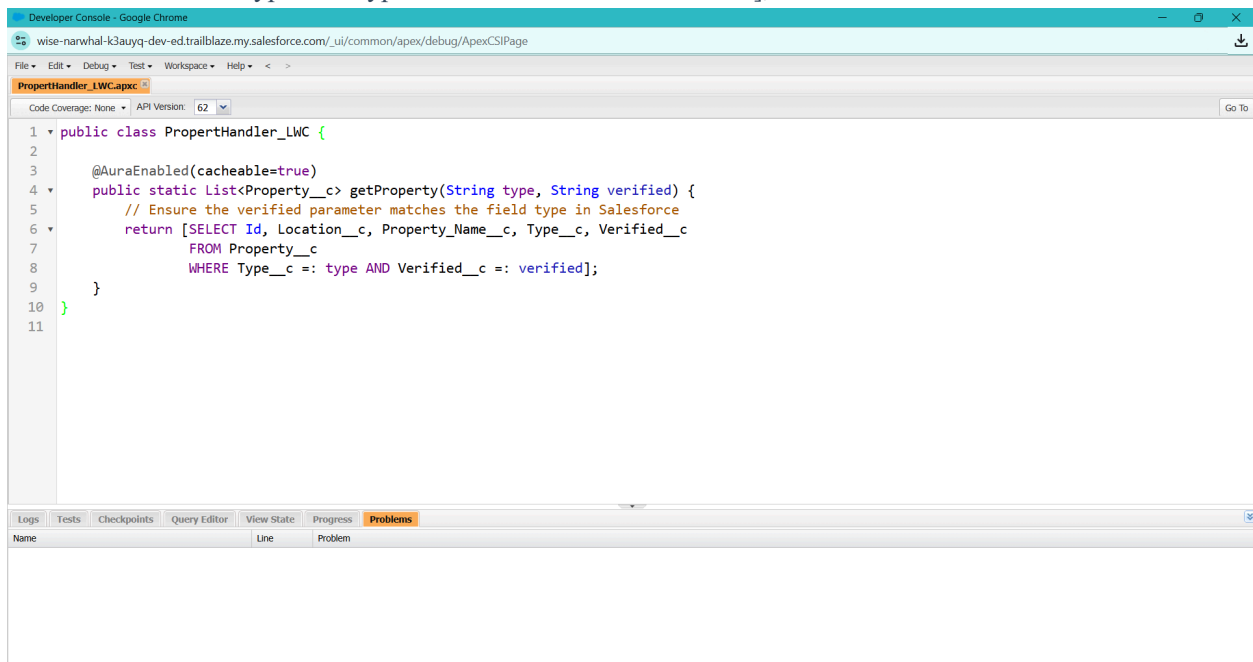
- a. Create an LWCComponent for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on “Search your Property Page”

Activity1

1. Create an ApexClass and make it aura enabled and name it “PropertHandler_LWC”

Code: -

```
public class PropertHandler_LWC (  
    @AuraEnabled(cacheable=true)  
  
    public static List<Property__c> getProperty(String type, Boolean verified) (  
        String  
        verifiedstr = verified ? 'true' : 'false' // Convert boolean to string return [SELECT Id,  
        Location__c, Property_Name__c, Type__c, Verified__c  
  
        FROM Property__c  
  
        WHERE Type__c = :type AND Verified__c = :verifiedStr];
```



1. Create a Lightning Web Component in your VsCode, and (ctrl+shift +P) and click on authorize an org.
2. Enter your login id and password to authorize your org.

3. Now (ctrl+shift +P) and Create a lightning Web Component and Name it Anything you want to.
(Example -
4. In yourHtml 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>

<template if:true={istru}>

  <div class="slds-box">

    <lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-
datatable>

  </div>

</template>
```

```

</templates>

<template if:false={isfalse}>

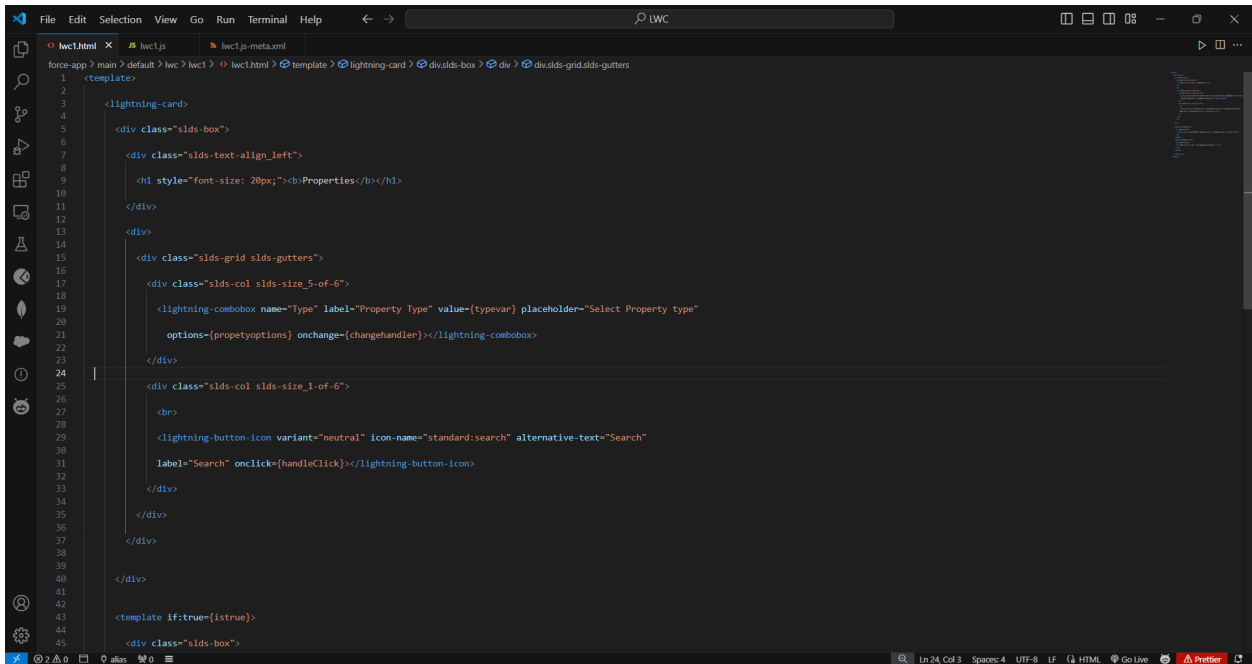
  <div class="slds-box">

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

  </div>
</template>
</lightning-card>

</templates>

```



1. In YourJs 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 = [

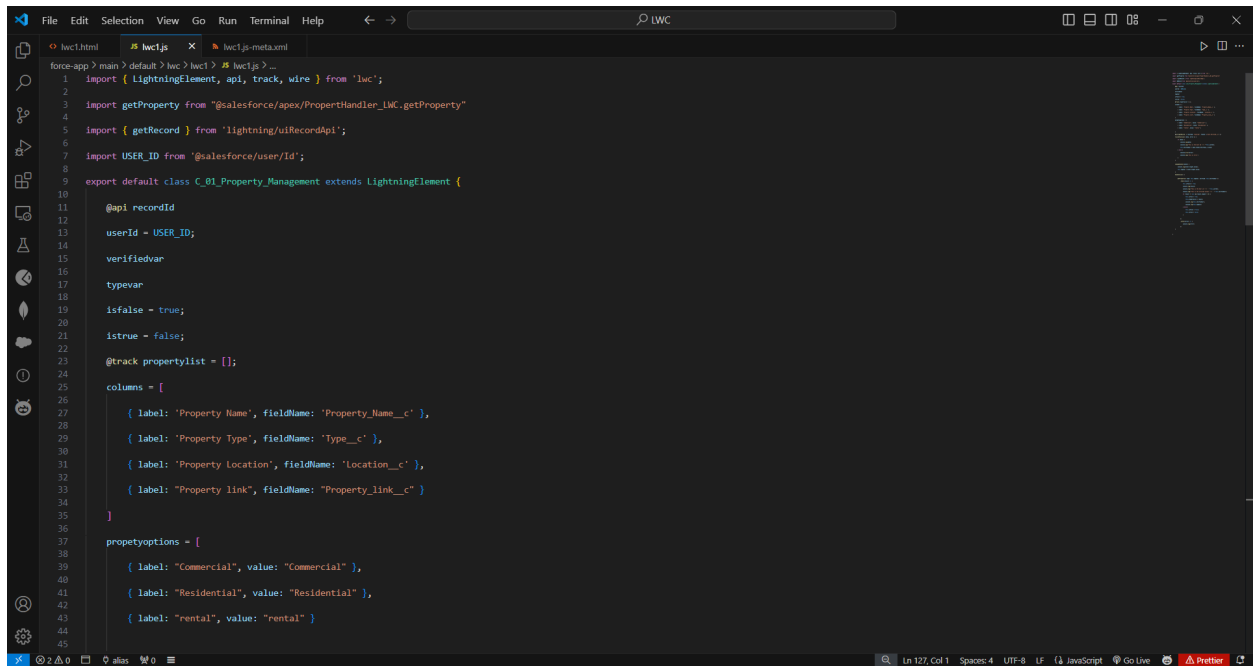
```

```

        ( label: 'Property Name', fieldName: 'Property_Name  c' ), ( label:
        'Property Type', fieldName: 'Type  c' ),
        ( label: 'Property Location', fieldName: 'Locationc' ), ( 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)

```

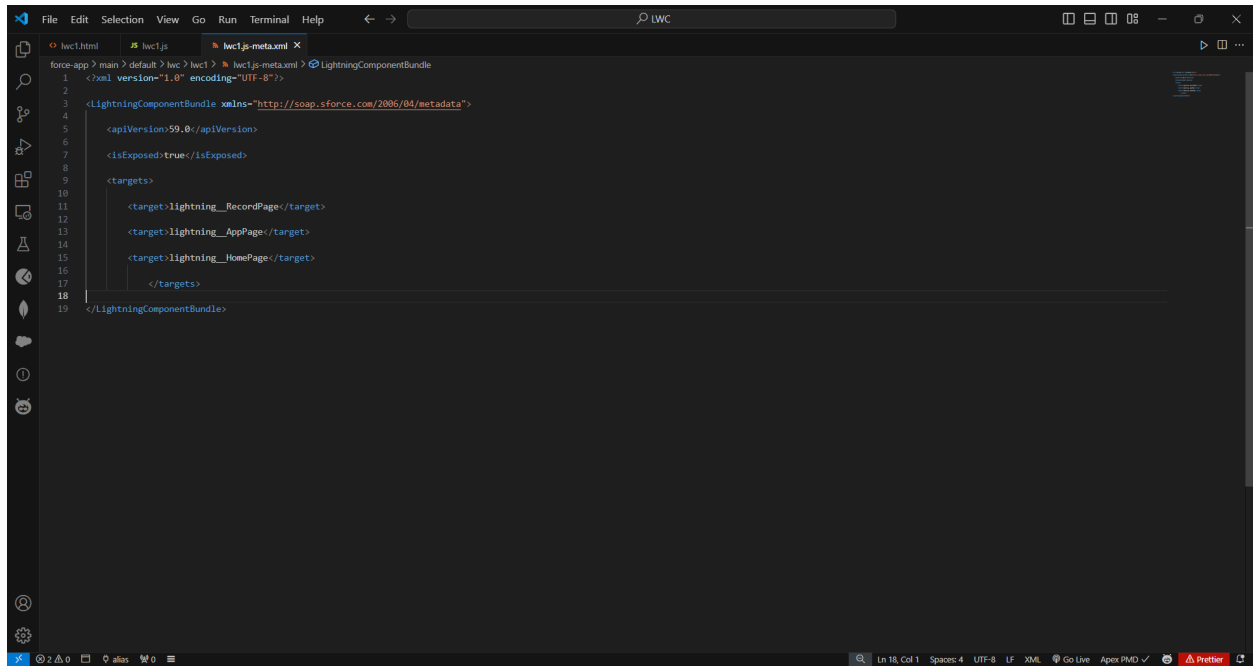


```
force-app > main > default > lwc > lwc1 > lwc1.js-meta.xml
1  import { LightningElement, api, track, wire } from 'lwc';
2
3  import getProperty from "@salesforce/apex/PropertyHandler_LWC.getProperty";
4
5  import { getRecord } from 'lightning/uiRecordApi';
6
7  import USER_ID from '@salesforce/user/Id';
8
9  export default class C_01_Property_Management extends LightningElement {
10
11      @api recordId;
12      userId = USER_ID;
13      verifiedvar;
14      typevar;
15      isfalse = true;
16      istrue = false;
17
18      @track propertylist = [];
19
20      columns = [
21          { label: 'Property Name', fieldName: 'Property_Name_c' },
22          { label: 'Property Type', fieldName: 'Type_c' },
23          { label: 'Property Location', fieldName: 'location_c' },
24          { label: 'Property link', fieldName: 'Property_link_c' }
25      ]
26
27      propertyoptions = [
28          { label: "Commercial", value: "Commercial" },
29          { label: "Residential", value: "Residential" },
30          { label: "rental", value: "rental" }
31      ]
32  }
```

1. In Yourmetafile give your targets to deploy the component.

Code

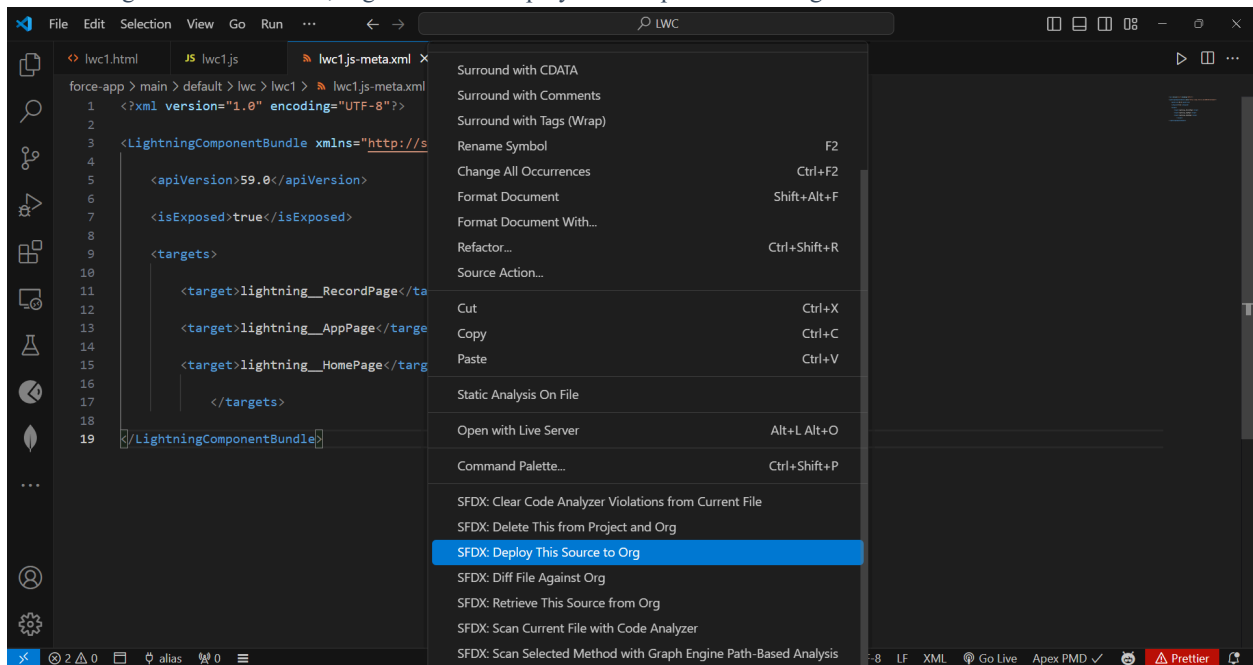
```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>59.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>Lightning__RecordPage</target>
    <target>Lightning__AppPage</target>
    <target>Lightning__HomePage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



The screenshot shows the Visual Studio Code editor with the file `lwc1js-meta.xml` open. The file contains an XML structure for a Lightning Component Bundle. The XML is as follows:

```
<?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>
```

After Saving all the three Codes , Right Click and deploy this component to the org

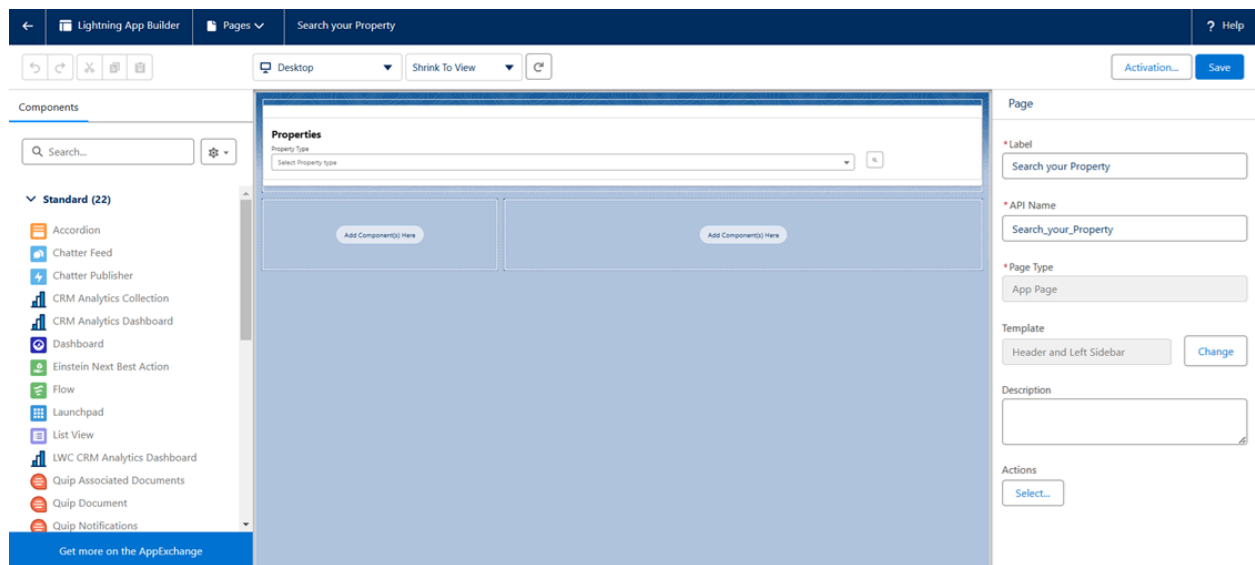


Drag this Component to your App Page

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 redirected to app pages then
3. 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.

1. Activity 1

From Setup>> Search For Apex Classes>> Click on "Security" behind "PropertyHandlerLWC".

2. From Profiles Add "Manager" and "Customer" and "Save".

The screenshot displays the Salesforce Setup interface. At the top, there is a search bar labeled 'Search Setup'. Below it, the navigation bar includes 'Setup', 'Home', and 'Object Manager'. A left sidebar contains a search bar with 'apex class' and a list of categories, including 'Custom Code' and 'Apex Classes'. The main content area is titled 'Profiles' and shows the configuration for 'PropertyHandler_LWC'. It includes a 'Save' button and a 'Cancel' button. Below these are two lists: 'Available Profiles' and 'Enabled Profiles'. The 'Available Profiles' list includes various system and custom profiles. The 'Enabled Profiles' list currently contains 'Customer', 'Manager', and 'System Administrator'. Between the two lists are 'Add' and 'Remove' buttons.

Search Setup

Setup Home Object Manager

apex class

Custom Code

Apex Classes

Didn't find what you're looking for?
Try using Global Search.

SETUP
Profiles

Enable Profile Access for Apex Class
PropertyHandler_LWC

Save Cancel

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

Add
Remove

Enabled Profiles

- Customer
- Manager
- System Administrator

