

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 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.

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/243228658565063>

The screenshot shows a web browser window with the address bar displaying form.jotform.com/243228658565063. The form itself is titled "Dreams World" and features a logo at the top. The form fields are as follows:

- Name ***: Two text input fields for "First Name" and "Last Name".
- Email ***: A text input field with the placeholder "example@gmail.com".
- Phone Number ***: A text input field with a placeholder "e.g., 73" and a note "Please enter a valid phone number".
- Which type of Property are you looking for?**: Three radio button options: "RESIDENTIAL", "COMMERCIAL", and "INDIA".
- Budget Amount ***: A text input field with a placeholder "e.g., 73".
- Address ***: A series of text input fields for "Street Address", "Apt. Suite, Bldg. No.", "City", "State/Province", and "Postal / Zip Code".

A green "Submit" button is located at the bottom right of the form.

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

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 navigation bar with 'Setup', 'Home', and 'Object Manager'. Below this, the 'Customer' object is selected. The left sidebar contains a list of 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, Restriction Rules, and Scoping Rules. The main content area is titled 'Details' and contains several sections: 'Description' (empty), 'API Name' (Customer__c), 'Custom' (checked), 'Singular Label' (Customer), 'Plural Label' (Customer), 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), 'Deployment Status' (Deployed), 'Help Settings' (Standard salesforce.com Help Window), and '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	uchecked
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 files as follows

The screenshot shows the Salesforce Setup interface, specifically the Object Manager for the 'Property' object. The left sidebar lists various setup options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts. The main content area is titled 'Details' and includes a description field, API Name (Property__c), Custom checkbox (checked), Singular Label (Property), Plural Label (Property), and a list of settings: Enable Reports (checked), Track Activities (checked), Track Field History, 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. 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”.

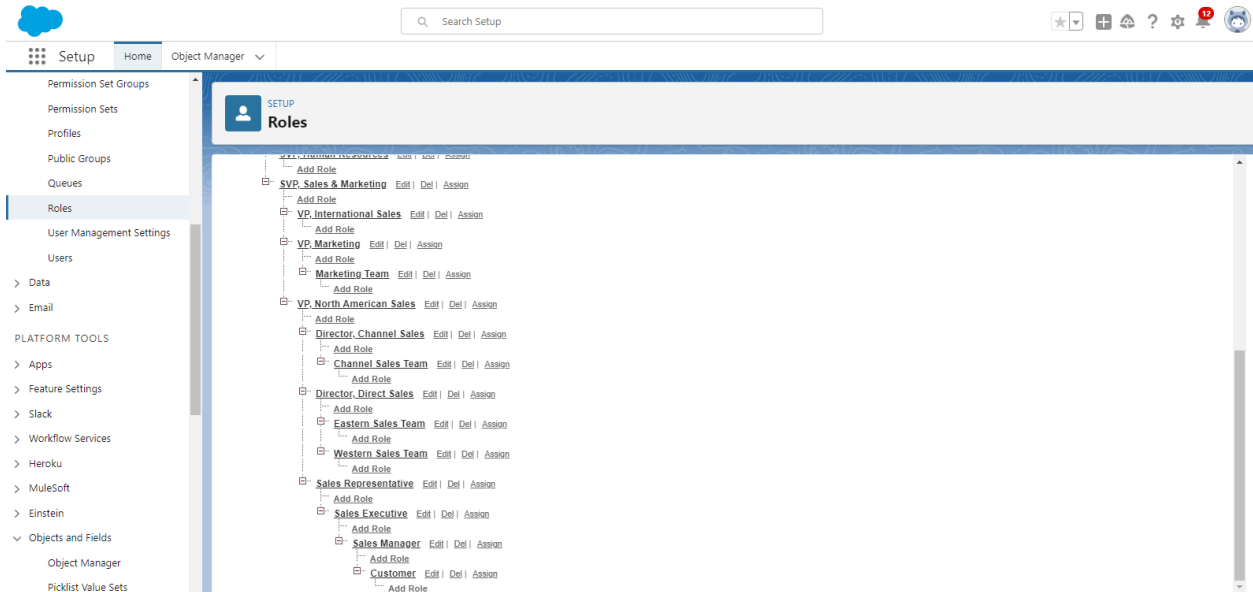
The screenshot shows the Jotform Form Builder interface for a form named "Dreams World". The left sidebar contains navigation options: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (highlighted), WORKFLOWS/ Formerly Approvals, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The top navigation bar includes BUILD, SETTINGS, and PUBLISH tabs. The main workspace is titled "Create a record" and shows a mapping of form fields to Salesforce fields. The "Object Fields" column lists form fields: Customer__c, City, Budget Amount, Property Type, Phone Number, Street Address, Email, Name, and State. The "Dreams World" column lists corresponding Salesforce fields: Name - First Name, Address - City, Budget Amount, Which type of Property are you lookin..., Phone Number, Address - Street Address, Email, Name - Last Name, and Address - Street Address 2. A "+ Add Field" link is at the bottom of the Object Fields column.

Then “Save the Integration” and “Finish”.

The screenshot shows the Jotform Form Builder interface for the "Dreams World" form, now in the "All Actions" configuration step. The left sidebar is the same as the previous screenshot. The top navigation bar shows BUILD, SETTINGS, and PUBLISH tabs. The main workspace is titled "SALESFORCE" and shows a list of actions. The first action is "Create or update a record" with the target "Customer". There are links for "See Action Logs" and "+ Add New Action".

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

Activity1

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.

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

Setup Home Object Manager

Quick Find

Profiles

Profile: Customer

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges | Enabled Apex Class Access | Enabled Visualforce Page Access | Enabled External Data Source Access | Enabled Named Credential Access | Enabled External Credential Principal Access | Enabled Custom Metadata Type Access | Enabled Custom Settings Definitions Access | Enabled Flow Access | Enabled Service Presence Status Access | Enabled Custom Permissions

Profile Detail

Name	Customer	
User License	Salesforce Platform	Custom Profile <input checked="" type="checkbox"/>
Description		
Created By	21501A0560 GUGULOTHU VARSHA, 19/11/2024, 3:48 pm	Modified By 21501A0560 GUGULOTHU VARSHA, 19/11/2024, 3:54 pm

Page Layouts

Standard Object Layouts	Global	Data Use Legal Basis
	Global Layout [View Assignment]	Data Use Legal Basis Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Data Use Purpose Layout [View Assignment]
Home Page Layout	Home Page Default [View Assignment]	Email Message Layout [View Assignment]
Access	Access Layout [View Assignment]	Event Layout [View Assignment]
Account	Account Layout [View Assignment]	Feed Item Layout [View Assignment]
Asset	Asset Layout [View Assignment]	Idea Varies by Record Type [View Assignment]
Asset Relationship	Asset Relationship Layout [View Assignment]	Individual Layout [View Assignment]

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”.

Setup Home Object Manager

Quick Find

Setup Home
Service Setup Assistant
Commerce Setup Assistant
Hyperforce Assistant
Release Updates
Lightning Experience Transition Assistant
Salesforce Mobile App
Lightning Usage
Optimizer
Sales Cloud Everywhere
ADMINISTRATION
Users
Permission Set Groups
Permission Sets
Profiles
Public Groups
Queues

SETUP Profiles

Profile Manager

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

[Login IP Ranges](#) | [Enabled Apex Class Access](#) | [Enabled Visualforce Page Access](#) | [Enabled External Data Source Access](#) | [Enabled Named Credential Access](#) | [Enabled External Credential Principal Access](#) | [Enabled Custom Metadata Type Access](#) | [Enabled Custom Setting Definitions Access](#) | [Enabled Flow Access](#) | [Enabled Service Presence Status Access](#) | [Enabled Custom Permissions](#)

Profile Detail [Edit](#) [Clone](#) [Delete](#) [View Users](#)

Name	Manager	Custom Profile
User License	Salesforce Platform	<input checked="" type="checkbox"/>
Description		
Created By	21501A0560 GUGULOTHU VARSHA, 19/11/2024, 3:50 pm	Modified By 21501A0560 GUGULOTHU VARSHA, 19/11/2024, 3:54 pm

Page Layouts

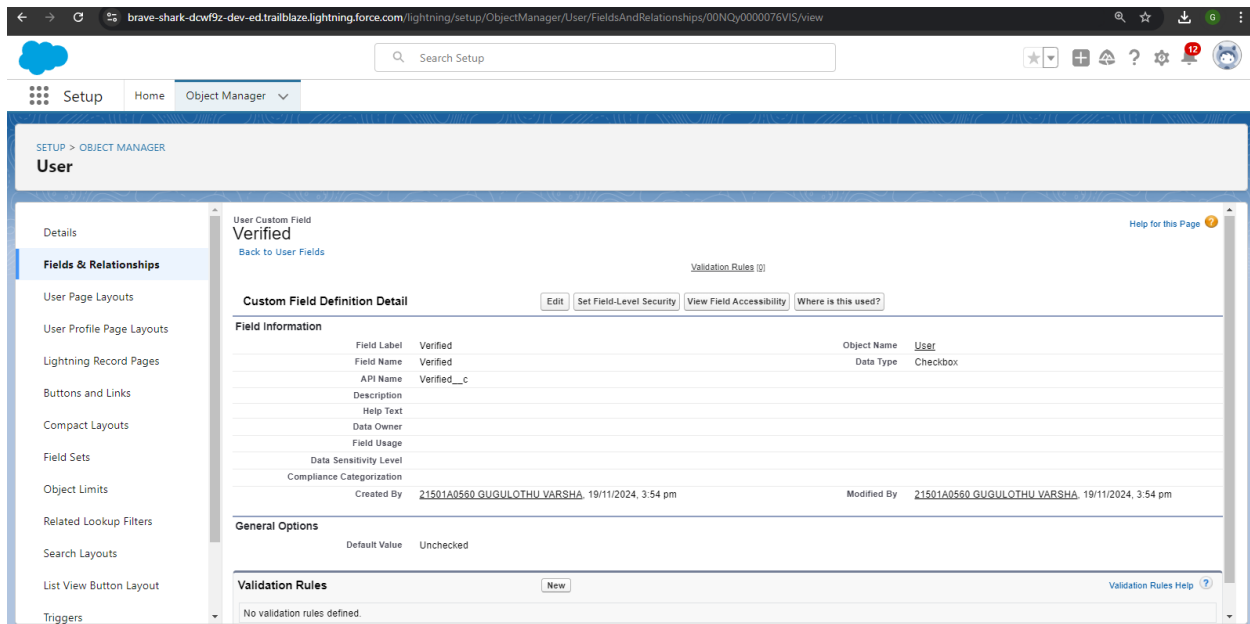
Standard Object Layouts	Global	Data Use Legal Basis	Data Use Legal Basis Layout
Email Application	Not Assigned View Assignment	Data Use Purpose	Data Use Purpose Layout (View Assignment)
Home Page Layout	Home Page Default View Assignment	Email Message	Email Message Layout (View Assignment)
Access	Access Layout (View Assignment)	Event	Event Layout (View Assignment)
Account	Account Layout (View Assignment)	Feed Item	Feed Item Layout (View Assignment)
Asset	Asset Layout (View Assignment)	Idea	Varies by Record Type (View Assignment)
Asset Relationship	Asset Relationship Layout (View Assignment)	Individual	Individual Layout (View Assignment)

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"



Create Users

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

User : 1

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

User : 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 : 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 : 4

1. Go to Setup »> Administration >> Users >> New User

2. Last Name >> Customer2

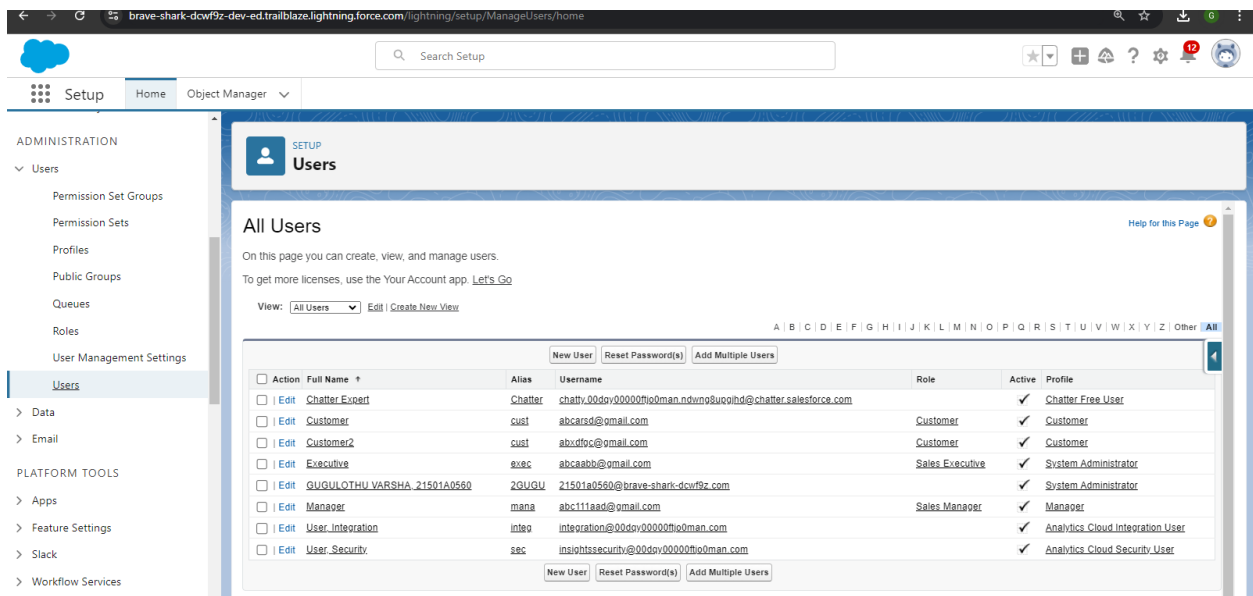
3. Role >> Customer

4. License >> Salesforce Platform

5. Profile »> Customer

6. Make Sure the verified check box is "checked"

7. Save



Create an Approval Process for Property Object

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
1. Process Name - Property Approval
2. select 2 criteria -
3. Location- i not equal to- blank,
4. Verified- Equals- false
5. Click next and "Next Automated Approver Determined By" Select Manager
6. From Record Editability Properties >> Click on AdministratorsoRthe currently assigned approver can edit records during the approval process.
7. FromStep 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.

Q. Search Setup

Setup Home Object Manager

Q. process

Feature Settings

- Marketing
 - Lead Processes
- Sales
 - Sales Processes
- Service
- Entitlement Management
 - Entitlement Processes
- Support Processes
- Process Automation
 - Approval Processes
 - Automation Home (Beta)
 - Flows
 - Migrate to Flow
 - Next Best Action
 - Paused And Failed Flow Interviews
 - Post Templates

Approval Processes

Approval Assignment Email template

Initial Submitters: Property Owner, Role: Sales Manager

Created By: Z1501A0569 GUGULOTHU VARSHA, 19/11/2024, 4:02 pm

Modified By: Z1501A0569 GUGULOTHU VARSHA, 19/11/2024, 4:09 pm

Initial Submission Actions

Action	Type	Description
Record Lock		Lock the record from being edited

Approval Steps

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

Final Rejection Actions

Action	Type	Description
Record Lock		Unlock the record from being editing
Field Update		UnVerified Property

Recall Actions

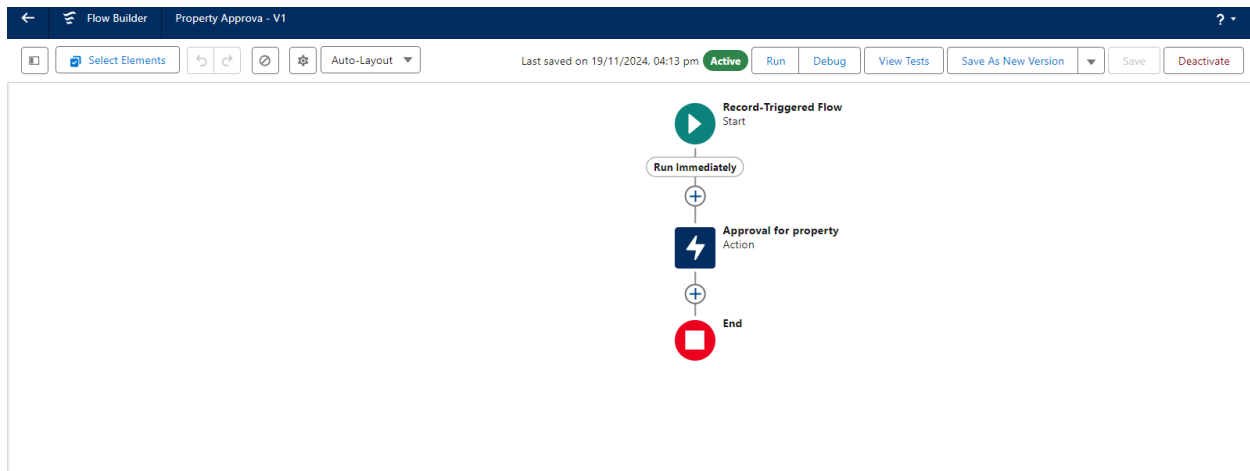
Create a Record trigger now to submit the Approval Process Automatically

A flow that can submit the records directly for approval

Activity1

- i. From Setup >> Search for Flows >> Click On New and Select “Record Trigger Flow”.
- ii. Select Object >> Property
- iii. Select “Trigger the flow when” >> “A record is created”
- iv. Set Entry Conditions >> “None”
- v. Add a “Action” >> “Submit for Approval”
- vi. Give Label >> Approval for property
- vii. Record Id >> (!SRecord.Id)
- viii. Done

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



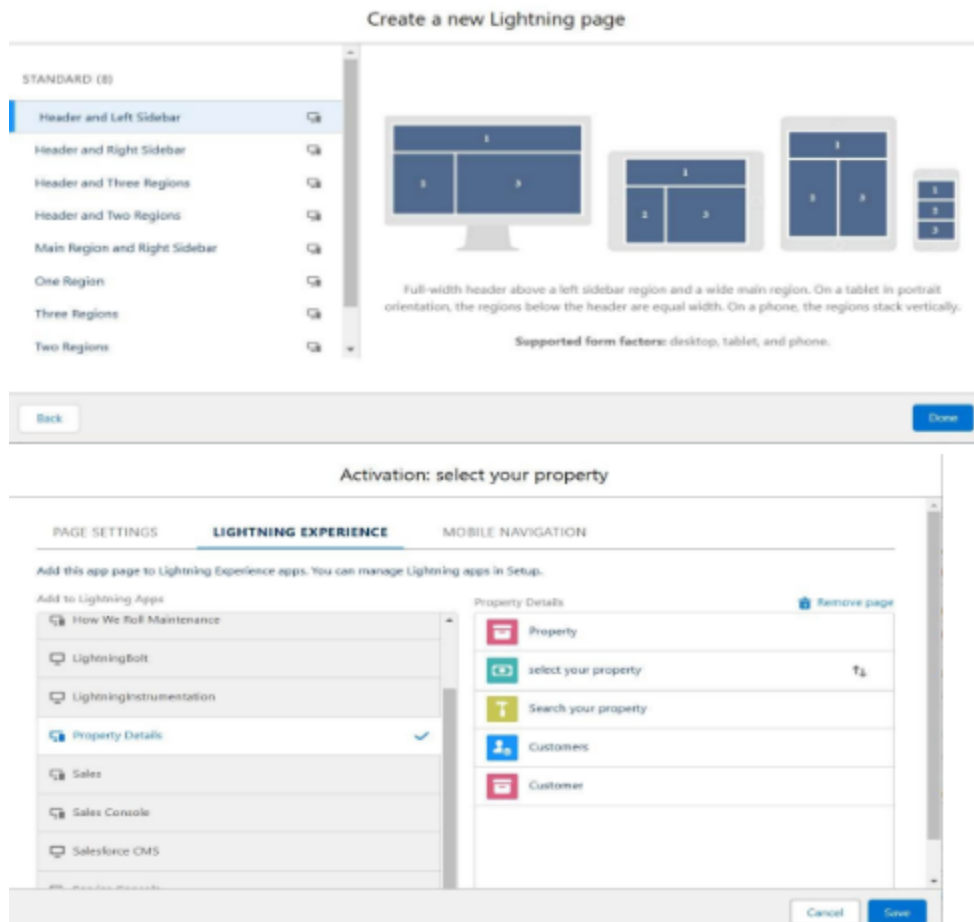
Create an App Page

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

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 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

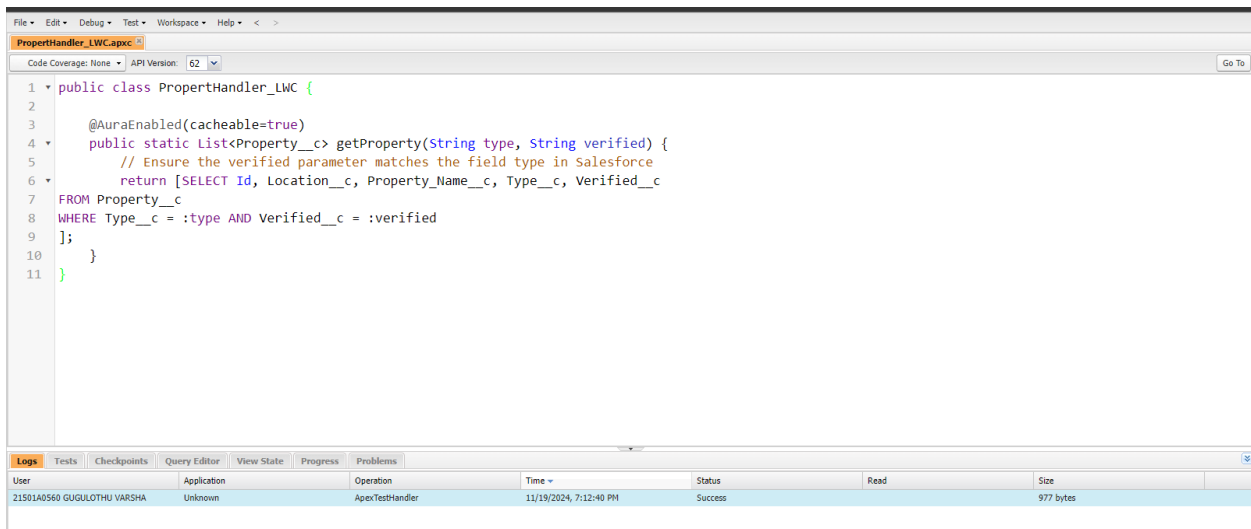
- i. Create an LWC Component for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on “Search your Property Page”

Activity1

1. Create an Apex Class 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 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={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>

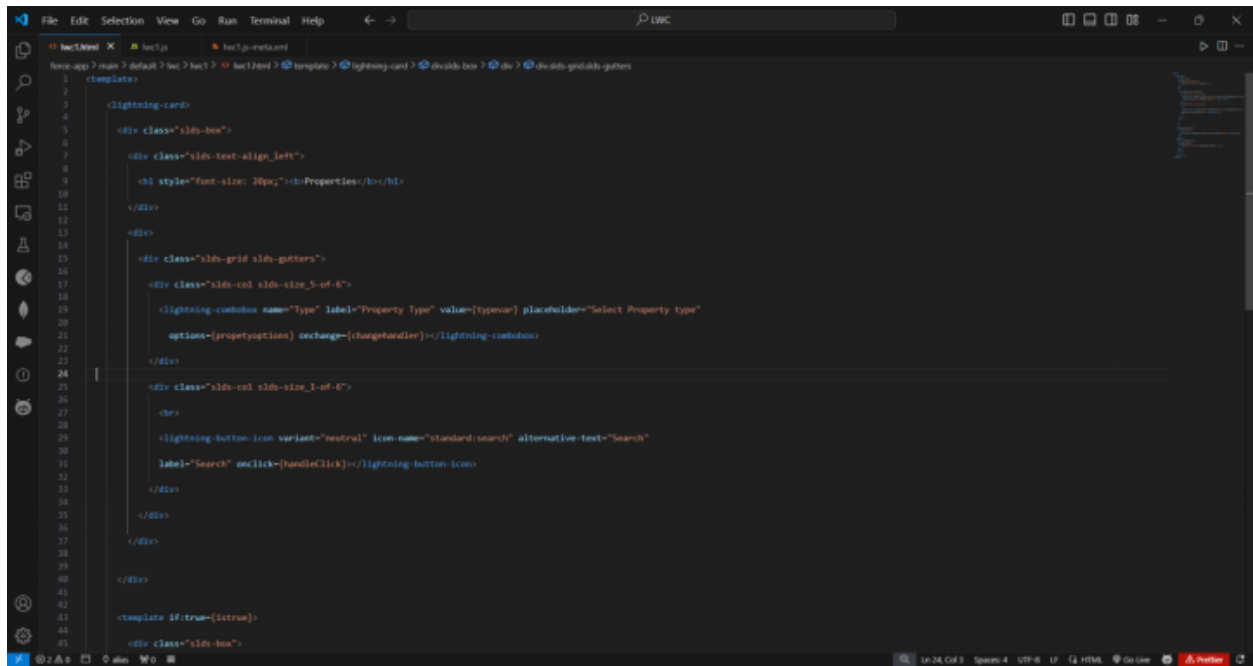
  </div>

</template>

</lightning-card>

```

</templates>



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; istruel =

    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:

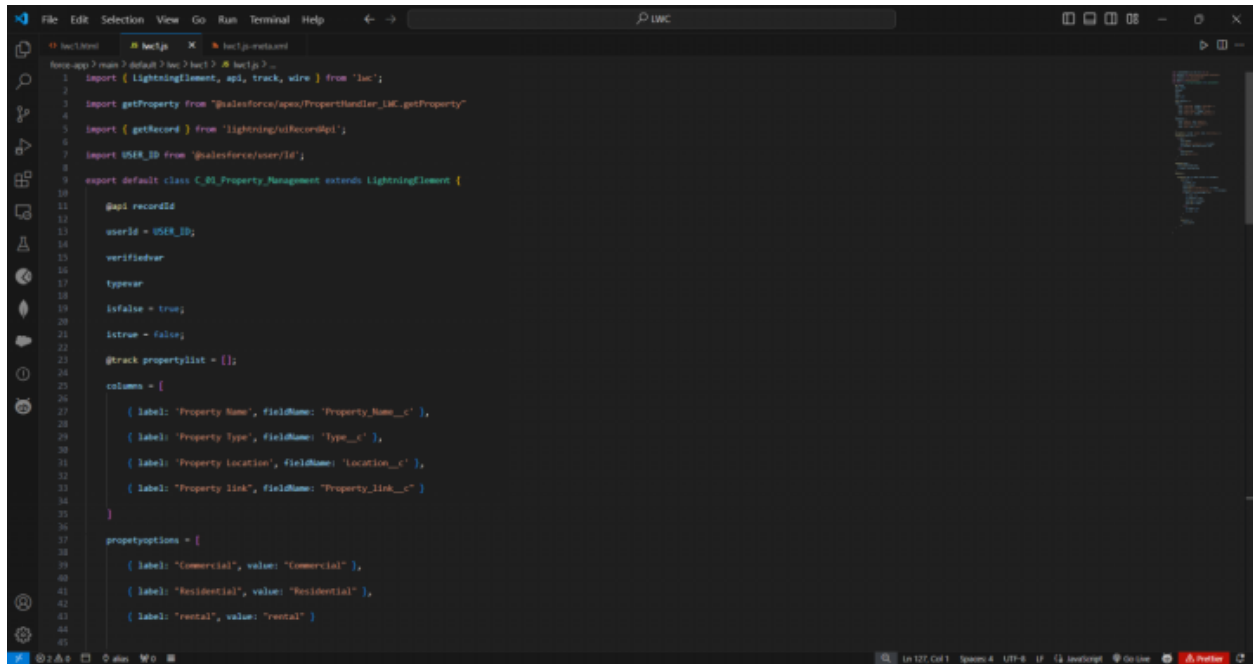
        "Residential", value: "Residential" ),
```

```

        ( 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)
            )
    )
}

```

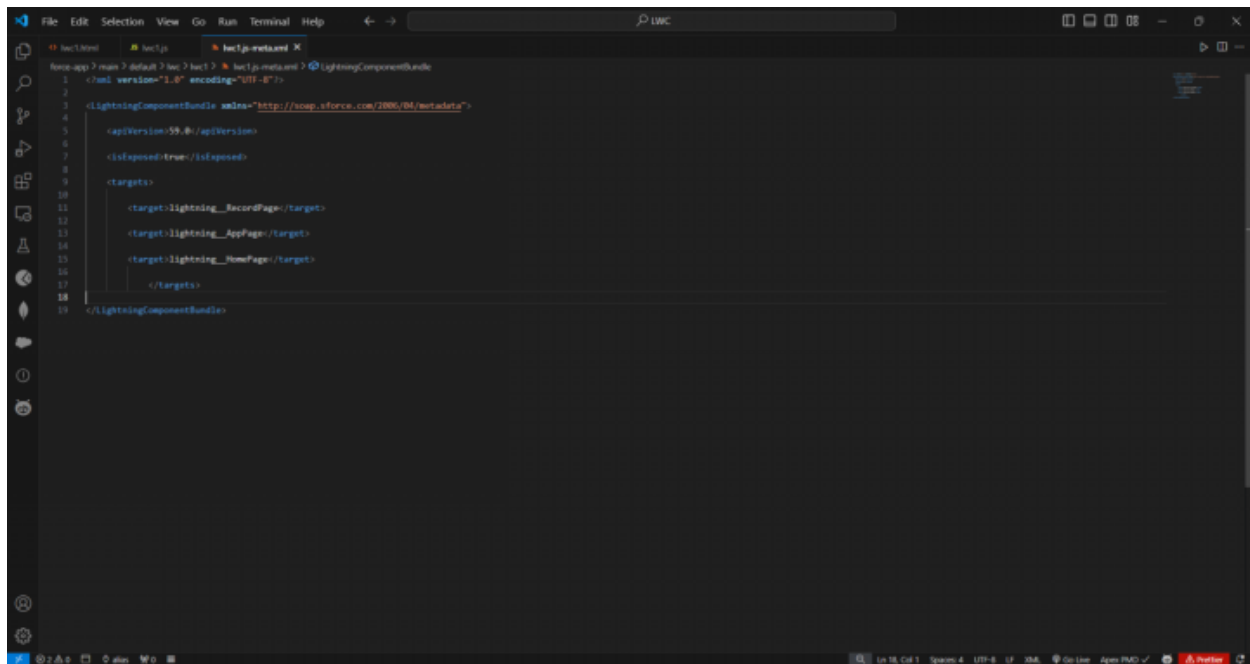
A screenshot of an IDE window showing Apex code. The code defines a Lightning component class that extends LightningComponent. It includes imports for LightningElement, api, track, wire, and various Salesforce APIs. The class has a recordId attribute, a user ID, and a list of property options. The @api recordId is used to fetch property details from a database. The code is as follows:

```
1 import { LightningElement, api, track, wire } from 'lwc';
2
3 import { getRecord } from 'lightning/uiRecordApi';
4 import { getRecord } from 'lightning/uiRecordApi';
5
6 import USER_ID from '@salesforce/user/id';
7
8 export default class C_01_Property_Management extends LightningComponent {
9
10     @api recordId;
11
12     userId = USER_ID;
13
14     verifyRecord;
15
16     typevar;
17
18     isfalse = true;
19
20     istrue = false;
21
22     @track propertylist = [];
23
24     columns = [
25
26         { label: 'Property Name', fieldName: 'Property_Name__c' },
27         { label: 'Property Type', fieldName: 'Type__c' },
28         { label: 'Property location', fieldName: 'location__c' },
29         { label: 'Property link', fieldName: 'Property_link__c' }
30     ]
31
32     propertyoptions = [
33         { label: 'Commercial', value: 'Commercial' },
34         { label: 'Residential', value: 'Residential' },
35         { label: 'rental', value: 'rental' }
36     ]
37 }
```

1. In Your metafile give your targets to deploy the component.

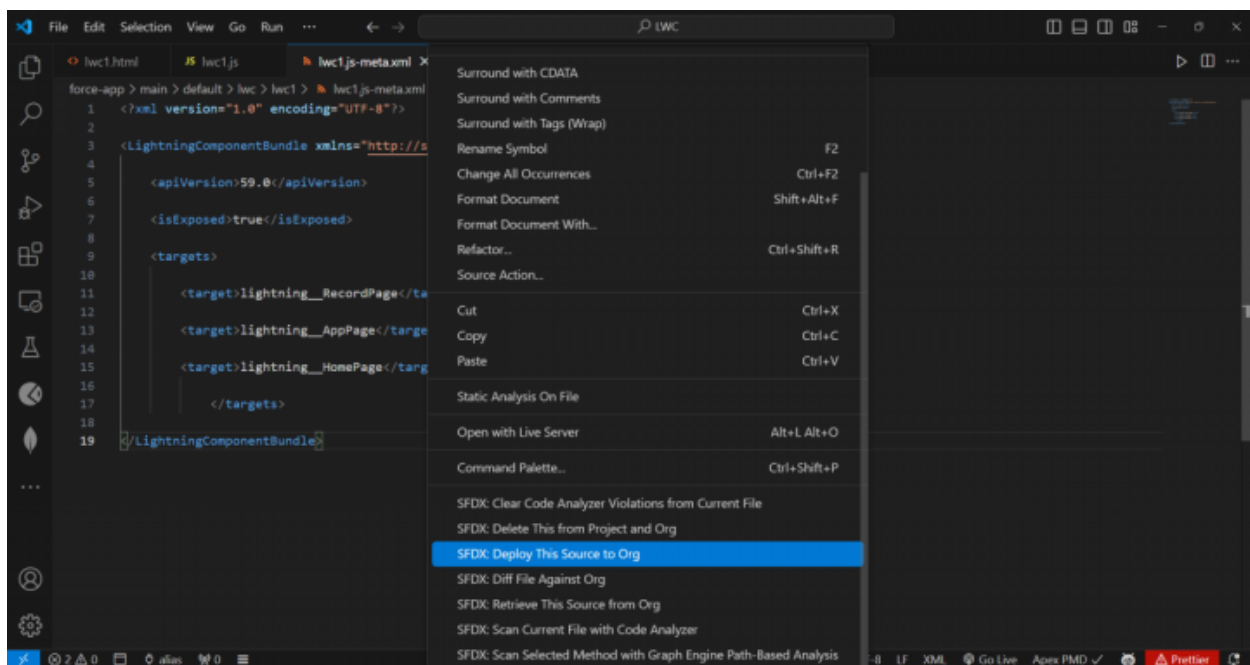
Code :-

```
<?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>
```



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
4
5   <apiVersion>59.0</apiVersion>
6
7   <isExposed>true</isExposed>
8
9   <targets>
10
11     <target>lightning__RecordPage</target>
12
13     <target>lightning__AppPage</target>
14
15     <target>lightning__HomePage</target>
16
17   </targets>
18
19 </LightningComponentBundle>
```

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

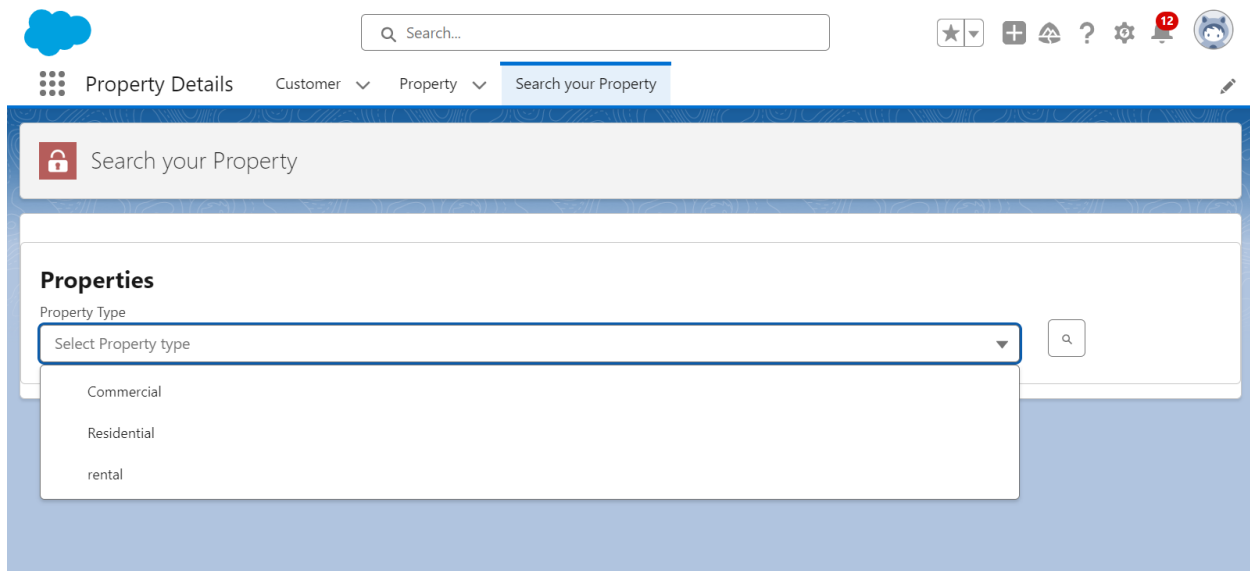


Drag this Component to your App Page

Adding the Component to your Page

Activity1

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.



The screenshot displays a web application interface. At the top, there is a blue header bar with a search bar on the left and a navigation menu on the right. The search bar contains the text "Search...". The navigation menu includes icons for a star, a plus sign, a house, a question mark, a gear, a bell with a red notification badge showing "12", and a user profile icon. Below the header, there is a sub-header with a grid icon, the text "Property Details", and two dropdown menus labeled "Customer" and "Property". A blue button labeled "Search your Property" is also present. The main content area features a search bar with a red padlock icon and the text "Search your Property". Below this, there is a section titled "Properties" with a "Property Type" label. A dropdown menu is open, showing three options: "Commercial", "Residential", and "rental". A search icon is visible to the right of the dropdown menu.

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. Activity1

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

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

The screenshot shows the Salesforce Setup interface. The left sidebar contains a search bar with 'apex' and a list of categories: Email, Custom Code, Environments, and Jobs. Under Custom Code, 'Apex Classes' is selected. The main content area is titled 'Profiles' and shows the 'Enable Profile Access for Apex Class' page for 'PropertyHandler_LWC'. It features 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'. There are 'Add' and 'Remove' buttons between the lists, and 'Save' and 'Cancel' buttons at the top right of the main content area.

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	