

A CRM APPLICATION TO HANDLE THE CLIENTS AND PROPERTY RELATED REQUIREMENTS

Project Description:

Harmony Homes 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 and houses Management
 - a. Manage property listings with details like type, price, location, and features.
 - b. Track the 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

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/243219231536048>

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.

Harmony spaces

Name

First NameLast Name

Email

example@example.com

Phone Number

(000) 000-0000

Please enter a valid phone number.

What typed house do you prefer?

☐ Apartment

☐ Villa

☐ Bungalow

Address

Street Address

Street Address Line 2

CityState / Province

Postal / Zip Code

Submit

Create Objects

Directly Creating Objects

Creating

- Go to your object from spreadsheet.
- Click on the link to
- [customer.xlsx](#)

from Spreadsheet

from Spreadsheet in Salesforce

Customer Object :

manager and and click oncreate object

get the spreadsheet

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	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.

Setup
 Home
 Object Manager

SETUP > OBJECT MANAGER
Customer

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
 15 Items, Sorted by Field Label

 New Deleted Fields Field Dependencies Set History Tracking

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)		✓
Emial	Emial__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone Number	Phone_Number__c	Number(18, 0)		

Creating Property Object :

- Follow the same from the customer object to create the Property Object
- 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	Jubilee hill Hyd	Checked

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

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

 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 displays the Jotform Form Builder interface. At the top, the Jotform logo is on the left, and the user's name 'Harmony spaces' is on the right, with a status message 'All changes saved at 3:23 PM' and a button to 'Add Collaborators'. Below the header is an orange navigation bar with 'BUILD', 'SETTINGS', and 'PUBLISH' tabs. On the left is a dark sidebar with menu items: 'EMAILS' (Send autoresponders and notifications), 'CONDITIONS' (Set up conditional logic), 'THANK YOU PAGE' (Show page after submission), and 'INTEGRATIONS' (Connect your form to other apps). The main workspace shows the 'SALESFORCE' integration. A back arrow is on the left. The text 'SALESFORCE' is followed by 'Send new leads, contacts, or accounts to your sales CRM'. Below this, it says 'All Actions' with a clock icon and 'See Action Logs' button, and a '+ Add New Action' button. A list of actions is shown, with the first one being 'Create or update a record' for the 'Customer' object.

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

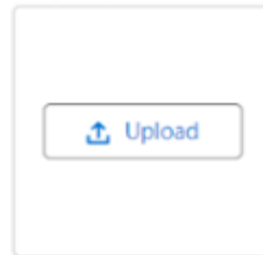
* App Name ⓘ

* Developer Name ⓘ

Description ⓘ

App Branding

Image ⓘ



Primary Color Hex Value ⓘ



Org Theme Options

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


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.


Profiles

The permissions defined here control access at the object level. Access to individual records within that object type is controlled by the sharing model. Set access levels based on the functional requirements for the profile. For example, create different groups of permissions for individual contributors, managers, and administrators. [How do I choose?](#)

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Accounts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Active Scratch Orgs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Authorization Forms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Data Uses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Texts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Background Operations	<input type="checkbox"/>					
Business Brands	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscriptions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Timings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contacts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Addresses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Type Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D&B Companies	<input type="checkbox"/>					
Data Use Legal Bases	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Use Purposes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Documents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Engagement Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ideas	<input type="checkbox"/>	<input type="checkbox"/>				
Individuals	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Labels	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Locations	<input type="checkbox"/>				<input type="checkbox"/>	
Namespace Registries	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Party Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Push Topics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Scratch Org Infos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Sellers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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”.

Profile Edit

Save

Save & New

Cancel

Name

Manager

User License

Salesforce Platform

Description

Custom Profile

✓

Custom App Settings

Visible

Default

Analytics Studio (standard__Insights)

☐

☐

App Launcher (standard__AppLauncher)

☐

☐

Dreamhouse (Dreamhouse)

☐

☐

Platform (standard__Platform)

☒

☒

Visible

Default

Playground Starter (trihdtps__Playground_Starter)

☐

☐

Property Details (Property_Details)

☒

☐

WDC (standard__Work)

☐

☐

Create Field on the User as per the business requirement.

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

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 verifiedcheckboxis"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 Home Object Manager

Search Setup

users

Users

Permission Set Groups
Permission Sets
Profiles
Public Groups
Queues
Roles
User Management Settings
Users
Feature Settings
Data.com
Prospector Users

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

SETUP 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

Action	Full Name	Alias	Username	Role	Active	Profile
Edit	ALLA BHUVANA NAGA SAI_21501A0505	2ALLA	21501a0505@cunning-narwhal-qppdu.com		✓	System Administrator
Edit	Chatter Expert	Chatter	chatty.00dns00000973z2aa.2cpwim3ty6z9@chatter.salesforce.com		✓	Chatter Free User
Edit	Customer	cust	user3@pvp.com	Customer	✓	Customer
Edit	Customer2	cust	user41@pvp.com	Customer	✓	Customer
Edit	Executive	exec	user1@pvp.com	Sales Executive	✓	System Administrator
Edit	Manager	mana	user2@pvp.com	Sales Manager	✓	Manager
Edit	User Integration	integ	integration@00dns00000973z2aa.com		✓	Analytics Cloud Integration User
Edit	User Security	sec	insightssecurity@00dns00000973z2aa.com		✓	Analytics Cloud Security User

New User Reset Password(s) Add Multiple Users

Create an Approval Process 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- i 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 fieldUpdate 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.



SETUP

Approval Processes

Approval Processes

Property

Help for this Page

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	1	Property Approval	

Create a Record Trigger flow to submit the Approval Process

Automatically

A flow that can submit the records directly for approval

Activity 1

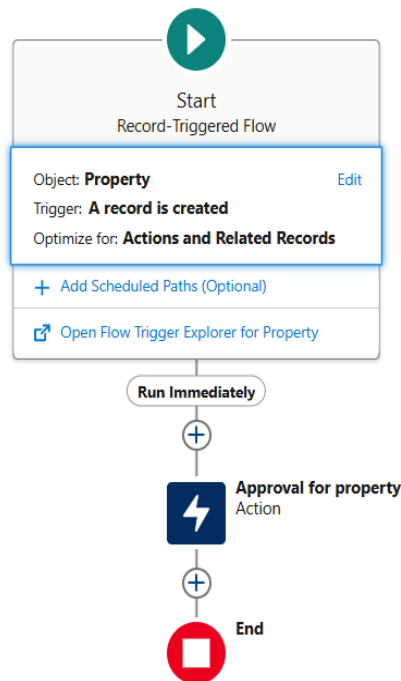
- From Setup >> Search for Flows >> Click On New and Select “Record Trigger Flow”.
- Select Object >> Property
- Select “Trigger the flow when” >> “A record is created”
- Set Entry Conditions >> “None”
- Add a “Action” >> “Submit for Approval”
- Give Label >> Approval for property
- Record Id >> (!SRecord.Id)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

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

Property Approval - V1

     Auto-Layout ▼

Last saved on 18/11/2024, 02:29 pm Active Run Debug View



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 Labelas“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 Settingsselectpageactivationas “Activate for all Users”.
7. From Lightning ExperienceClick on “Property Details” and click on Add Page“.
8. Then Clickon“Save”

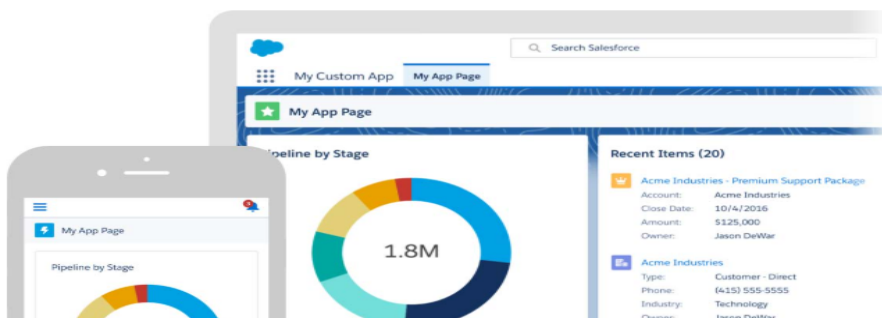
Create a new Lightning page

App Page

Home Page

Record Page

Build a one-page app for Lightning Experience and the mobile app.



Create a new Lightning page

STANDARD (8)

Header and Left Sidebar

Header and Right Sidebar

Header and Three Regions

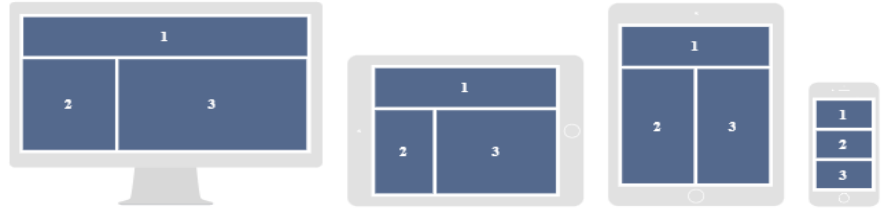
Header and Two Regions

Main Region and Right Sidebar

One Region

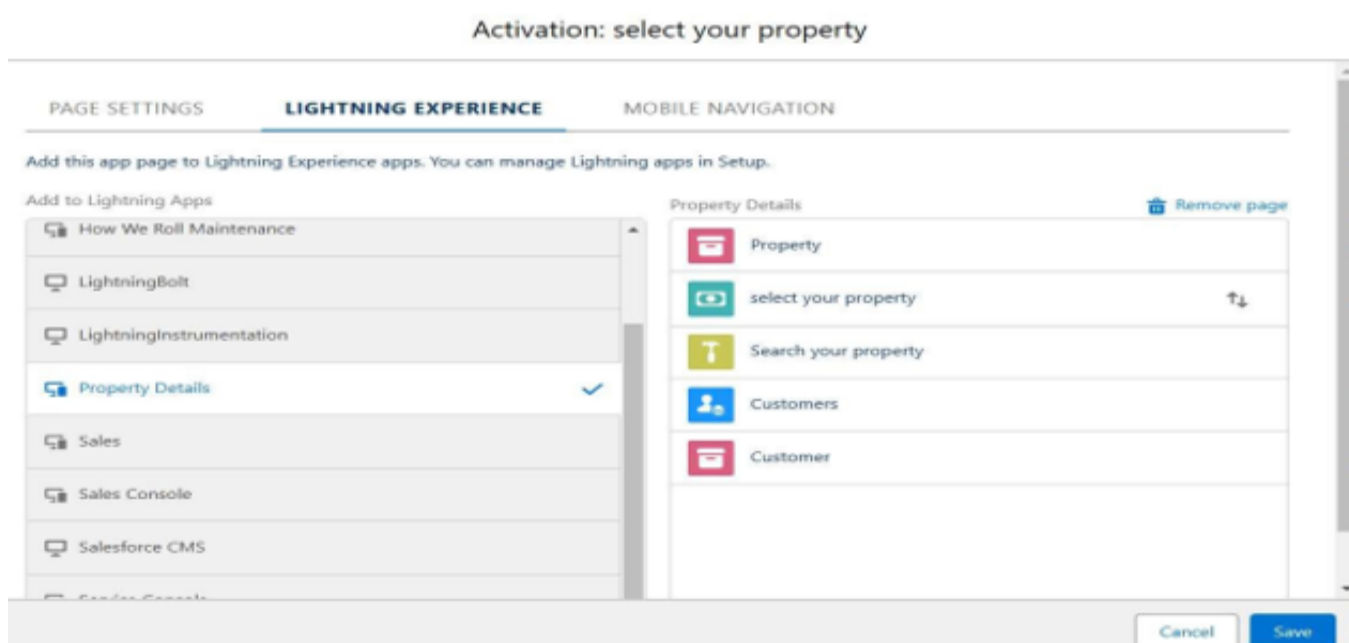
Three Regions

Two Regions



Full-width header above a left sidebar region and a wide main region. On a tablet in portrait orientation, the regions below the header are equal width. On a phone, the regions stack vertically.

Supported form factors: desktop, tablet, and phone.



Create a LWC Component

1. 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"

```
public class PropertHandler_LWC {

    @AuraEnabled(cacheable=true)

    public static List<Property1__c> getProperty(String type, String verified) {

        // Ensure the verified parameter matches the field name in Salesforce

        return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c

                FROM Property1__c

                WHERE Type__c =: type AND Verified__c =: verified];

    }

}
```

```

cunning-narwhal-qpptu-dev-ed.trailblaze.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help < >
PropertyHandler_LWC.apxc
Code Coverage: None API Version: 62
1 public class PropertyHandler_LWC {
2
3     @AuraEnabled(cacheable=true)
4     public static List<Property1__c> getProperty(String type, String verified) {
5         // Ensure the verified parameter matches the field name in Salesforce
6         return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c
7                 FROM Property1__c
8                 WHERE Type__c =: type AND Verified__c =: verified];
9     }
10 }
11

```

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={propertyoptions} onchange={handleChange}></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>

```

```

</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  <template>
2
3  <lightning-card>
4
5    <div class="slds-box">
6
7      <div class="slds-text-align_left">
8
9        <h1 style="font-size: 20px;"><b>Properties</b></h1>
10
11      </div>
12
13      <div>
14
15        <div class="slds-grid slds-gutters">
16
17          <div class="slds-col slds-size_5-of-6">
18
19            <lightning-combobox name="Type" label="Property Type" value={typevar} placeholder="Select Property type"
20
21              options={propetyoptions} onchange={changehandler}></lightning-combobox>
22
23          </div>
24
25          <div class="slds-col slds-size_1-of-6">
26
27            <br>
28
29            <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search"
30
31              label="Search" onclick={handleClick}></lightning-button-icon>
32
33          </div>

```

```

import ( LightningElement, api, track, wire ) from 'Twc';

import getProperty from "@salesforce/apex/PropertHandler_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)

```

```

webcomp.html JS webcomp.js X webcomp.js-meta.xml
force-app > main > default > lwc > webcomp > JS webcomp.js > Webcomp > columns
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 Webcomp extends LightningElement {
10
11      @api recordId
12
13      userId = USER_ID;
14
15      verifiedvar
16
17      typevar
18
19      isfalse = true;
20
21      istrue = false;
22
23      @track propertylist = [];
24
25      columns = [
26
27          { label: 'Property Name', fieldName: 'Property_Name__c' },
28
29          { label: 'Property Type', fieldName: 'Type__c' },
30
31          { label: 'Property Location', fieldName: 'Location__c' },
32
33          { label: "Property link", fieldName: "Property_link__c" }

```

```

webcomp.html JS webcomp.js X webcomp.js-meta.xml
force-app > main > default > lwc > webcomp > JS webcomp.js > Webcomp > columns
9  export default class Webcomp extends LightningElement {
37      propertyoptions = [
38
39          { label: "Commercial", value: "Commercial" },
40
41          { label: "Residential", value: "Residential" },
42
43          { label: "rental", value: "rental" }
44
45      ]
46
47
48      Tabnine | Edit | Test | Explain | Document | Ask
49      @wire(getRecord, { recordId: "$userId", fields: ['User.Verified__c'] })
50      recordFunction({ data, error }) {
51
52          if (data) {
53
54              console.log(data)
55
56              console.log("This is the User Id ---> "+this.userId);
57
58              this.verifiedvar = data.fields.Verified__c.value;
59
60          } else {
61
62              console.error(error)
63
64              console.log('this is error')
65
66          }
67
68      }

```

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>

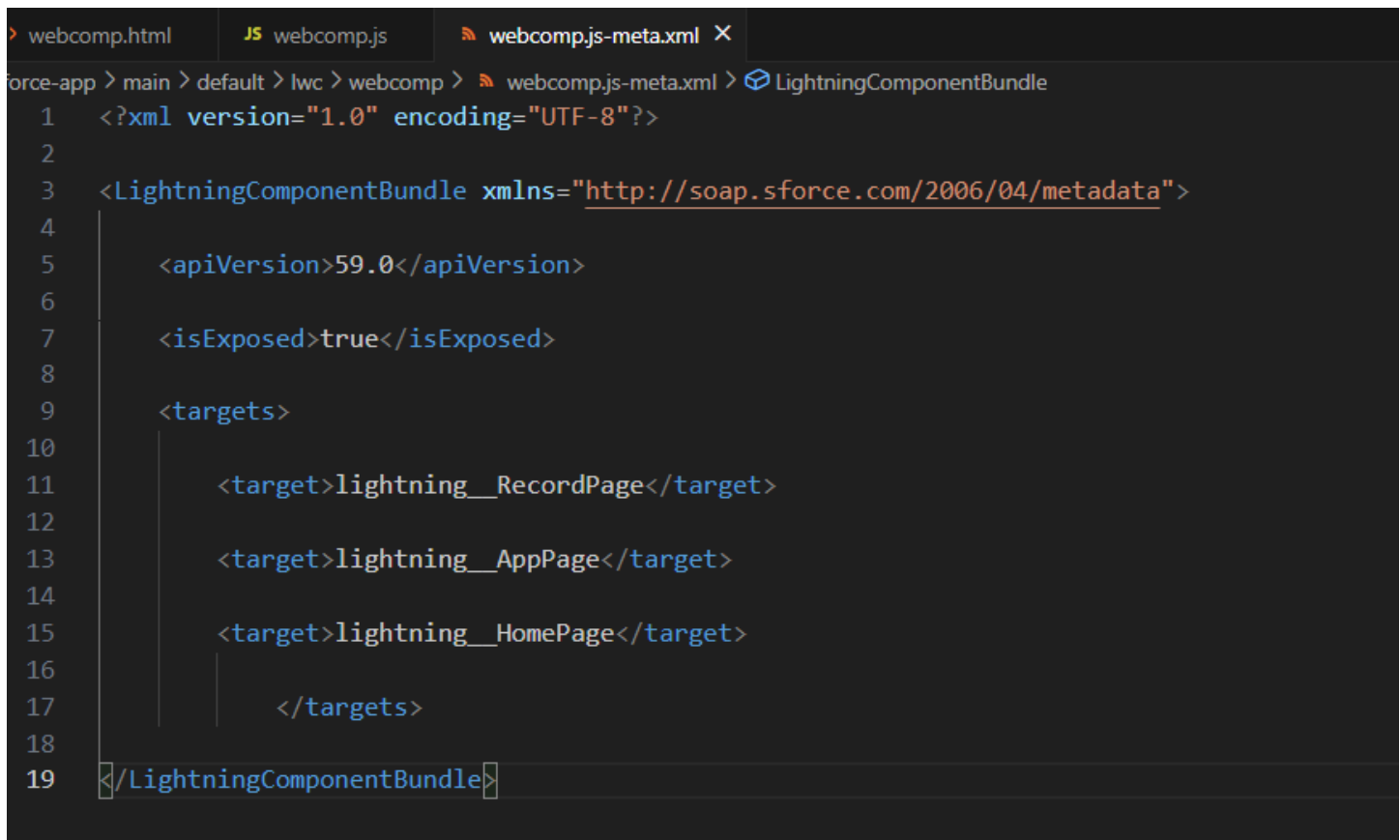
    <target>Lightning__RecordPage</target>

    <target>Lightning__AppPage</target>

    <target>Lightning__HomePage</target>

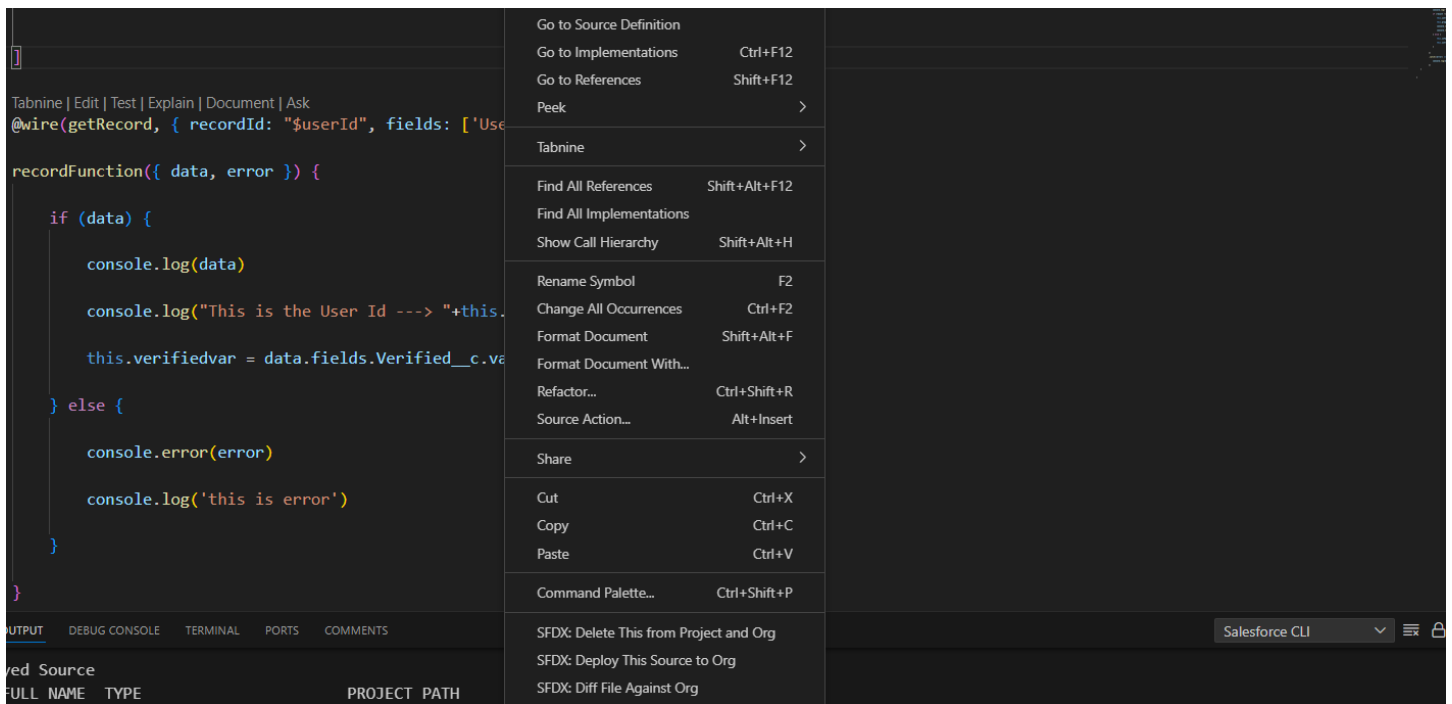
  </targets>

</LightningComponentBundle>
```

A screenshot of a code editor with a dark theme. The editor has three tabs at the top: 'webcomp.html', 'JS webcomp.js', and 'webcomp.js-meta.xml'. The active tab is 'webcomp.js-meta.xml'. The breadcrumb navigation shows the path: 'force-app > main > default > lwc > webcomp > webcomp.js-meta.xml > LightningComponentBundle'. The code is an XML snippet for a Lightning Component Bundle, with line numbers 1 through 19 on the left. The code is:

```
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

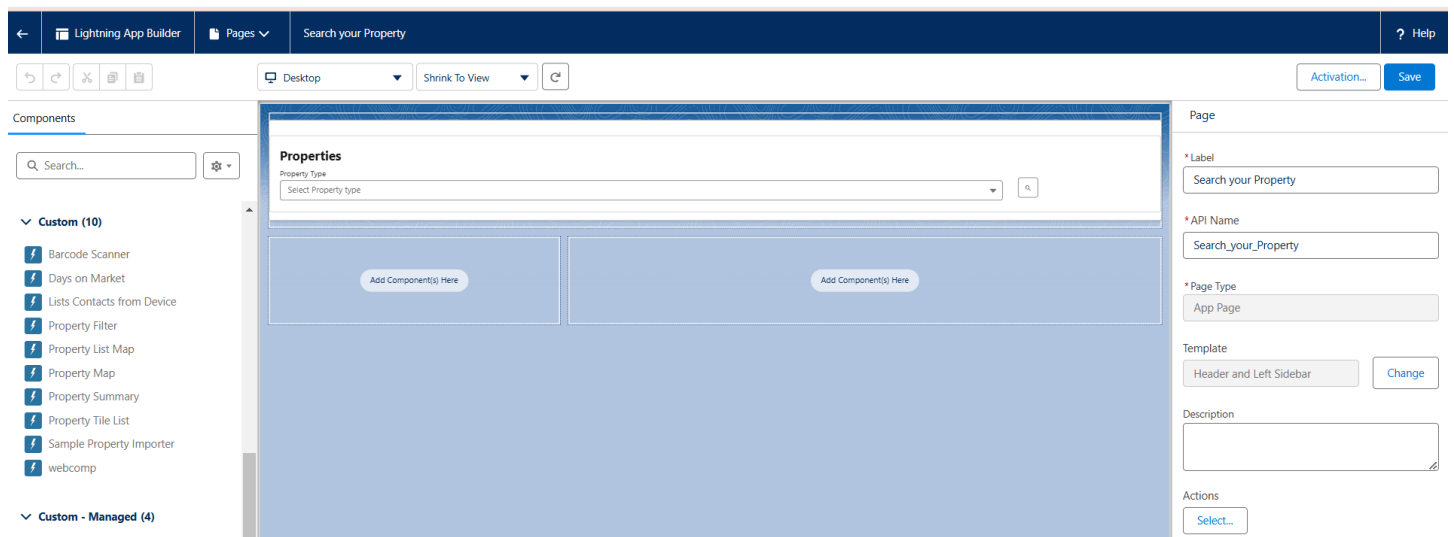


Drag this Component to your App Page

Adding the Component to your Page

Activity1

1. From Setup >> Go to App Launcher >> Search for PropertyDetails
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
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. Activity1

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

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

Q apex

✓ Email

- Apex Exception Email

✓ Custom Code

- Apex Classes
- Apex Settings
- Apex Test Execution
- Apex Test History
- Apex Triggers

✓ Environments

- Jobs
 - Apex Flex Queue
 - Apex Jobs

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		Enabled Profiles
Analytics Cloud Integration User	Add Remove	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		