

# **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 easing growth in the real estate market.

1. Client Management
  - a. Add, update, and delete client details.
  - b. Track client taste, 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 useable for sale, rent, or lease.
  - c. Upload photos and documents for properties.
3. Requirement Matching
  - a. Match client essentials with useable properties using filters.
  - b. Notify clients about new properties that fit their standard.
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.

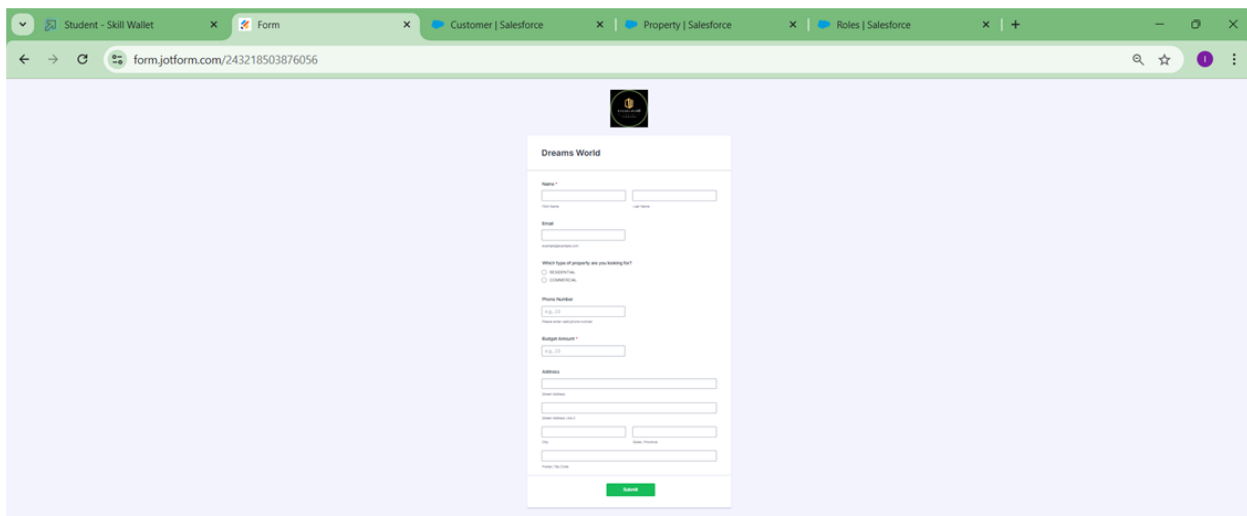
**Milestone 1: Create a Jot form and integrate it with the governance to create a record of customers automatically.**

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 governance. 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 governance.

## Activity1

Open your browser and search for jot form and log in.

1. After login, click on create form and click on start
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/21501a0534/SweetHome>

A screenshot of a web browser displaying a JotForm titled "Dreams World". The browser's address bar shows the URL "form.jotform.com/243218503876056". The form itself is white and centered on a light purple background. It contains several input fields: "Name" (split into first and last name), "Email", "Phone Number" (with a country code dropdown), "Budget Amount" (with a currency dropdown), and "Address" (split into street, city, state, and zip). There are also radio buttons for "Which type of property are you looking for?" with options "apartment/flat" and "condominium". A green "Submit" button is at the bottom of the form.

## Create Objects from Spreadsheet

Directly Creating Objects from a Spreadsheet in Salesforce

### Creating Customer Object :

1. Go to your object manager and click on create object from spreadsheet.
2. Click on the link to get the spreadsheet
3. [customer](#)

Customer	Phone Number	Email	State	Property Type	Budget Amount	Street Address	Street Address	City	postal code	Verified
Rakesh	788797	rakesh@gmail	Telangana	Residential	4000000	gb road	street no 45	Hyderabad	555001	checked
prakash	55448855	p@gmail.com	Maharashtra	Commercial	8000000	gachibowli	indira road	mumbai	6600014	unchecked
Prajwal	454545	prajwal@gmail	Maharashtra	Rental	25000	kamdli	kathora	Amravati	444805	checked

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

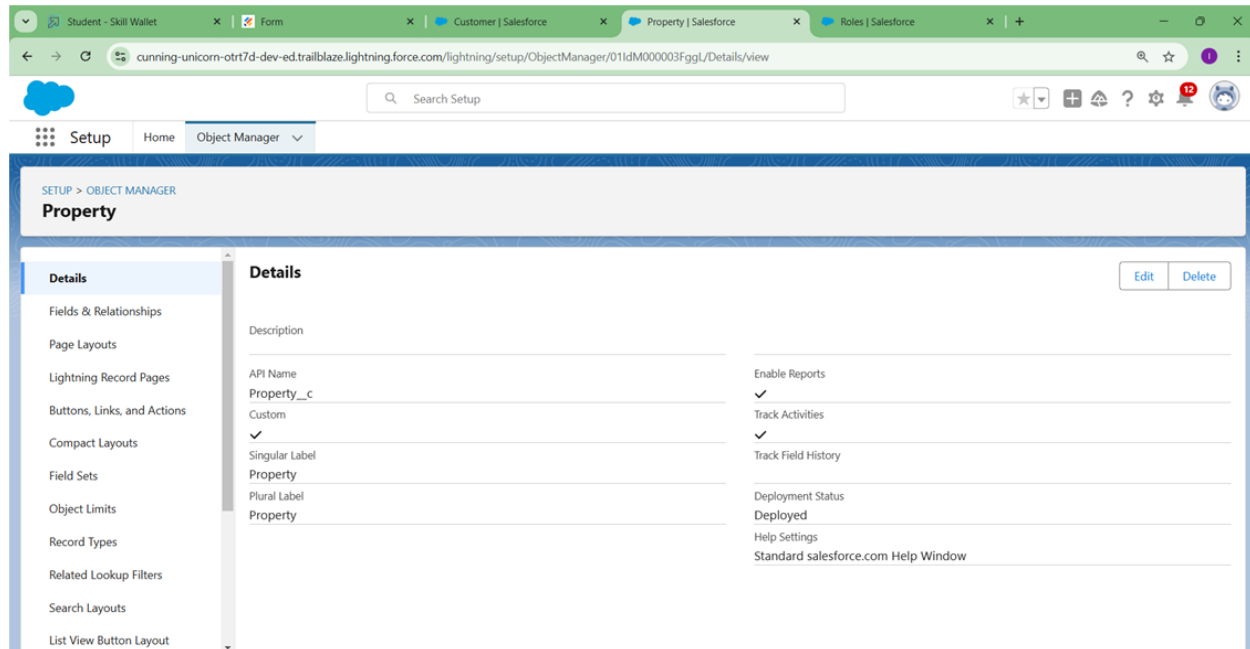
The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. The 'Customer' object is chosen, and the 'Details' section is expanded. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Object Limits. The main area displays the object's details, including its API Name (Customer\_\_c), Singular Label (Customer), and Plural Label (Customers). It also shows settings for Enable Reports, Track Activities, and Track Field History, all of which are checked. The Deployment Status is 'Deployed', and the Help Settings point to the standard Salesforce help window.

## Creating Property Object :

1. Follow the same from the customer object to create theProperty 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 file is as follows



## Integrate Jot form with Salesforce Platform

In this Milestone, we are going to integrate jot form with Salesforce

### action

1. On the Jot form Platform, Click on Integration and choose Salesforce
2. Click on User Integration and choose “Add to From”
3. Select the Organization with which you want to Integrate your jot form with and select your account
4. Select an Action -Create a record.
5. Select a Salesforce Object : - Customer

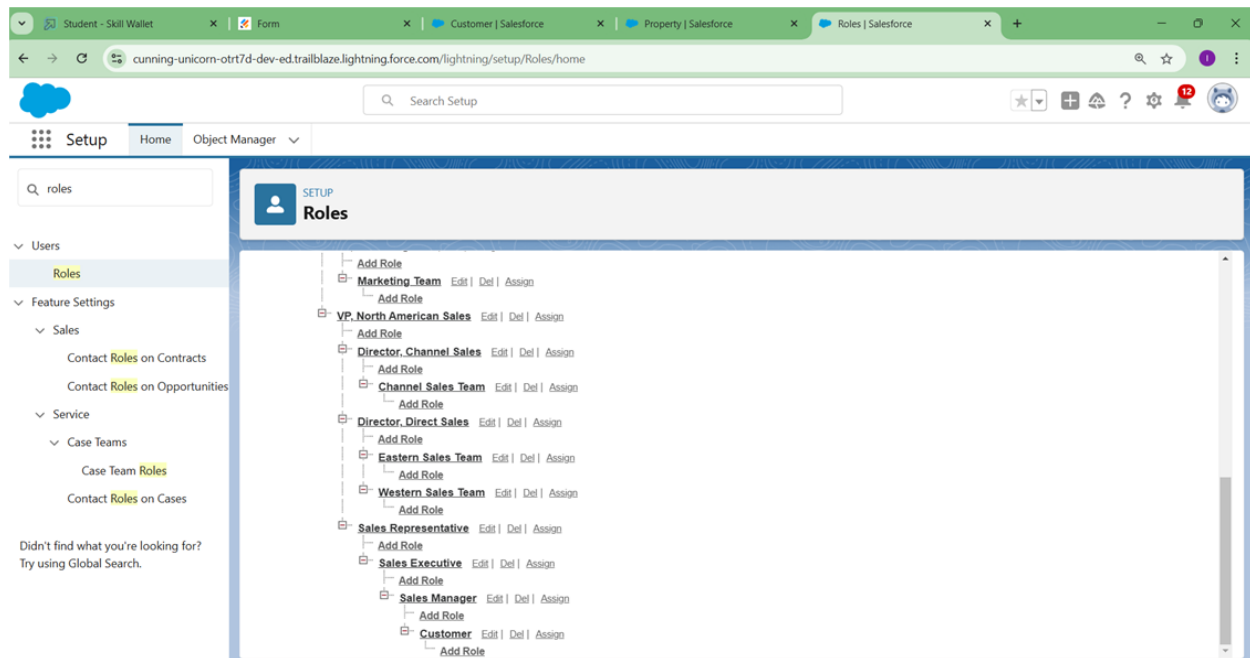
Map Every field on the Object with the fields on the form and “Save Action”.

Then “Save the Integration” and “Finish.”

## Create Roles

here we need to Create Roles as per business demand

### Activity:- 1



1. if we don't find a sales interpreter, 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 DetailsApp

An App where the objects will be displayed

### Activity1

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 piloting 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

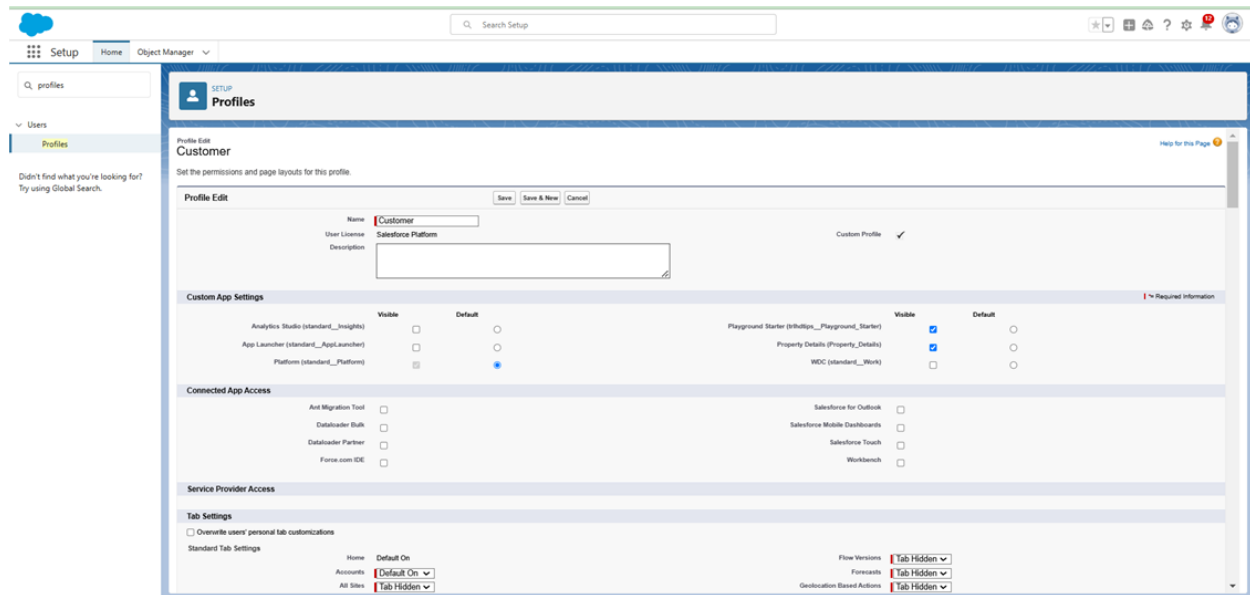
App Launcher Preview

## Create Profiles

Create profiles as per business demand

### 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 onlyProperty 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”.



**Profiles Manager**

Profile Edit: Name: Manager, User License: Salesforce Platform, Description: [Empty]

**Custom App Settings**

App Name	Visible	Default
Analytics Studio (standard__insights)	<input type="checkbox"/>	<input type="radio"/>
App Launcher (standard__AppLauncher)	<input type="checkbox"/>	<input type="radio"/>
Platform (standard__Platform)	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
Playground Starter (standard__PlaygroundStarter)	<input checked="" type="checkbox"/>	<input type="radio"/>
Property Details (Property_Details)	<input type="checkbox"/>	<input type="radio"/>
WDC (standard__Work)	<input type="checkbox"/>	<input type="radio"/>

**Connected App Access**

App Name	Visible
Acad Migration Tool	<input type="checkbox"/>
Dataloader Bulk	<input type="checkbox"/>
Dataloader Partner	<input type="checkbox"/>
Force.com IDE	<input type="checkbox"/>
Salesforce for Outlook	<input type="checkbox"/>
Salesforce Mobile Dashboards	<input type="checkbox"/>
Salesforce Touch	<input type="checkbox"/>
Workbench	<input type="checkbox"/>

**Service Provider Access**

**Tab Settings**

☐ Override users' personal tab customizations

Standard Tab Settings

Tab Name	Home	Default On	Tab Hidden
Accounts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
All Sites	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Flow Versions: ☐ Tab Hidden, ☐ Tab Hidden, ☐ Tab Hidden

Forecast: ☐ Tab Hidden, ☐ Tab Hidden, ☐ Tab Hidden

Geolocation Based Actions: ☐ Tab Hidden, ☐ Tab Hidden, ☐ Tab Hidden

## Create a CheckBox field on user

Create a Field on the User as per the business essential.

### Activity:- 1

1. Setup >> Object Manager >> Search for User >> Fields and Relationships
2. select the Data type "Check Box"
3. Create a new Field Named as "Verified"

**User**

**Fields & Relationships**

**Custom Field Definition Detail**

Field Information

Field Label	Verified	Field Name	Verified	API Name	Verified__c	Description	Help Text	Date Owner	Field Usage	Date Sensitivity Level	Compliance Categorization	Created By	Created On	Modified By	Modified On
Verified	Verified	Verified	Verified	Verified__c	Verified__c							BUMINI PRASADHANA REDURU	18/11/2024, 1:42 pm	BUMINI PRASADHANA REDURU	18/11/2024, 1:42 pm

General Options

Default Value: Unchecked

**Validation Rules**

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 four 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

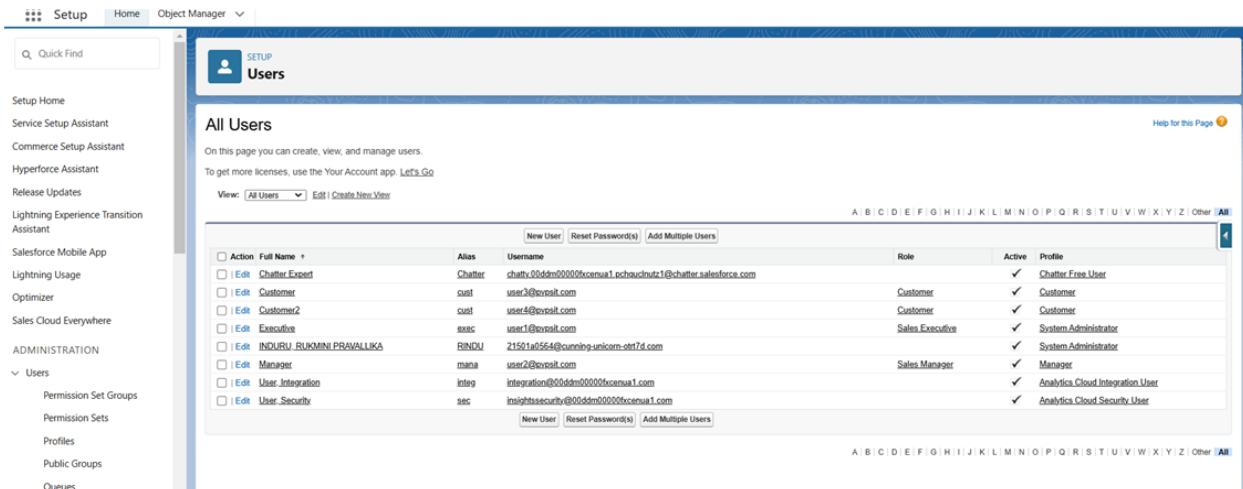
1. Go to Setup > Administration >> Users >> New User
2. Last Name >> Manager
3. Role >> Sales Manager
4. License >> Salesforce Platform
5. Profile >> Manager
6. Save

## 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 verifiedcheckboxis"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. Ensure 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
3. Process Name - Property Approval

4. select 2 measures -
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 AdministratorsoRthe assigned approver can edit records during the approval process.
9. 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 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

Property

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
<a href="#">Edit</a>   <a href="#">Deactivate</a>	1	Property Approval	

Inactive Approval Processes

No approval processes available

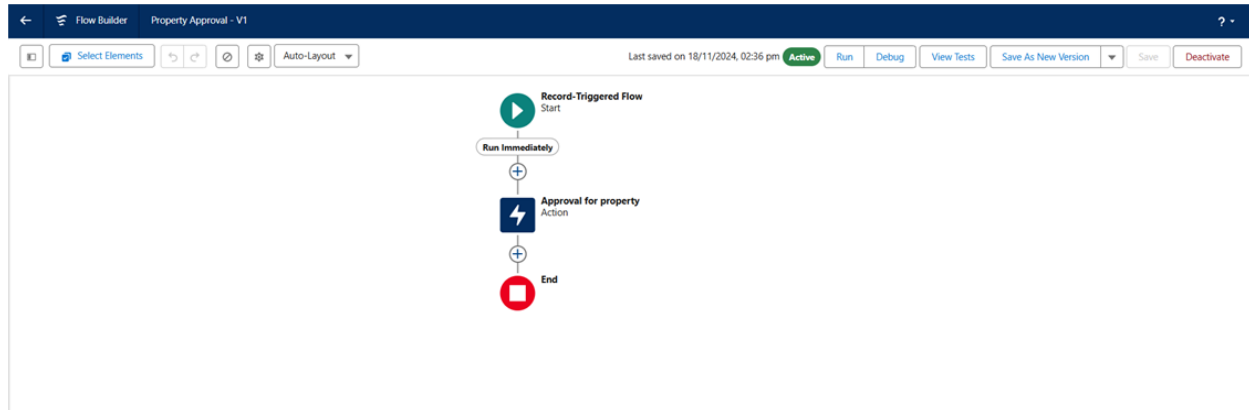
## Create a Record triggerñow to submit the Approval Process Automatically

A flow that can submit the records directly for approval

### Activity1

- FromSetup >>Search forFlows >>Click OnNew andSelect “Record Trigger Flow”.
- SelectObject >>Property
- Select“Trigger the flow when” >> “A record is created”
- SetEntry 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”



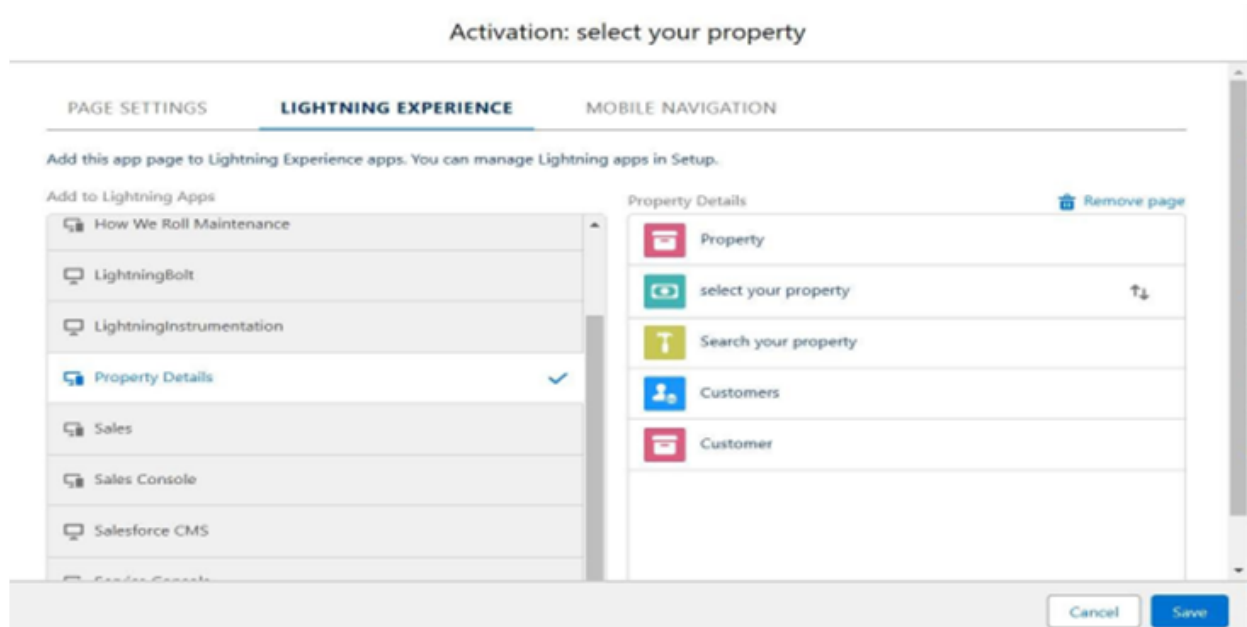
## Create an App Page

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 on “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> getProperty(String type, Boolean verified) ( String
        verified = verified ? 'true' : 'false' // Convert Boolean to string return [SELECT Id,
        Location c, Property_Namec, Type c, Verified c
        FROM Property c
        WHERE Type c = :type AND Verified c = :verifiedStr];
```

```

PropertyHandler_LWC.apex
Code Coverage: None API Version: 62 Go To
1 public class PropertyHandler_LWC {
2
3     @AuraEnabled(cacheable=true)
4     public static List<Property__c> getProperty(String type, String verified) {
5         // Ensure the field names match those in the Salesforce schema
6         return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c
7                 FROM Property__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 placement.
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" option-text="Search"
label="Search" onclick={handleClick}></lightning-button-icon>

                </div>

            </div>

        </div>

    </div>

</div>

```



```

    </div>

</div>

</div>

<template if:true={isTrue}>

    <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={propertyOptions} onchange={changeHandler}></lightning-combobox>
22
23                  </div>
24
25                  <div class="slds-col slds-size_1-of-6">
26
27                      <div>
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>
34
35                  </div>
36
37              </div>
38
39          </div>
40
41      </div>
42
43      <template if:true={isTrue}>
44
45          <div class="slds-box">

```

1. In YourJs File Write this code :-

Code :-

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

```

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; istrue =
    false;
    @track propertylist = [];

    columns = [
        { label: 'Property Name', fieldName: 'Property_Name__c' }, { label:
        'Property Type', fieldName: 'Type__c' },
        { label: 'Property Location', fieldName: 'Location__c' }, { label:
        'Property link', fieldName: 'Property link__c' }
    ]
    propertyoptions = [
        { label: "Commercial", value: "Commercial" }, { label:
        "act", 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 && != 0) (

        this.isTrue = true; this.propertylist =
        result; console.log(this.verifiedvar);
        console.log(this.typevar)

    ) else (

        this.isfalse = false;
        this.isTrue = false;

    ).catch((error) => (
        console.log(error)
    )

```

```

1  import { LightningElement, api, track, wire } from 'lwc';
2
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
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          { label: 'Property Type', fieldName: 'Type_c' },
29          { label: 'Property Location', fieldName: 'Location_c' },
30          { label: 'Property link', fieldName: 'Property_link_c' }
31      ]
32
33      propertyoptions = [
34
35          { label: 'Commercial', value: 'Commercial' },
36          { label: 'Residential', value: 'Residential' },
37          { label: 'rental', value: 'rental' }
38      ]
39
40
41
42
43
44
45

```

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>
    </targets>

```

```
<target>lightning__AppPage</targets>

<target>lightning__HomePage</targets>

</targets>

</LightningComponentBundle>
```

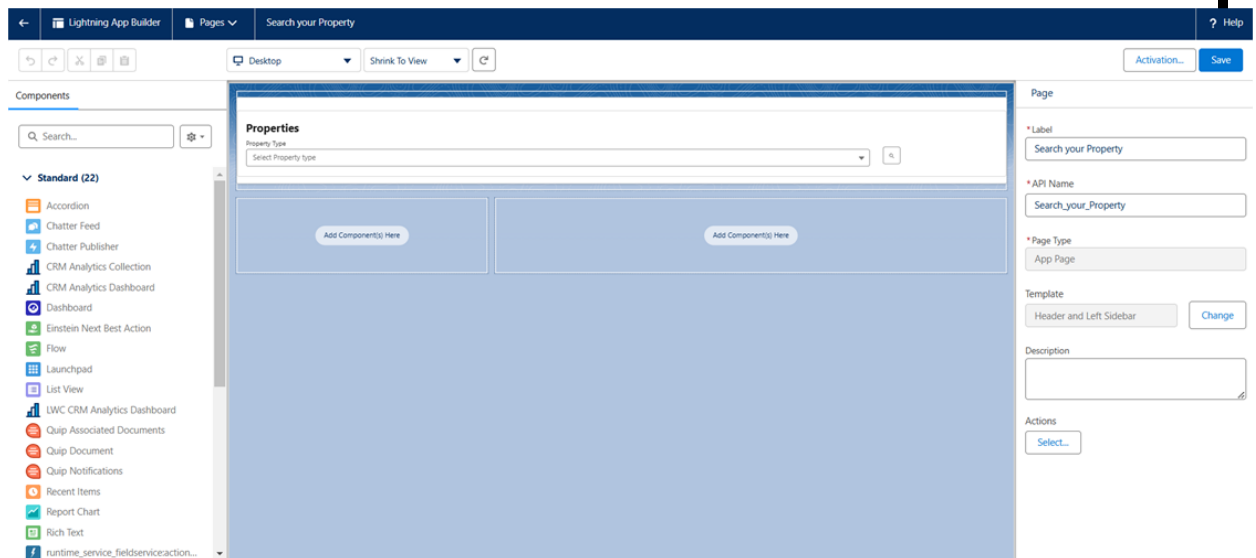
After Saving all the three Codes, Right Click and deploy this component to the governance

# 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 thisPage click on gear icon and click on EditPage
3. after clicking on the 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 protection; Enable the protection for the profiles that need to access this class.

1. Activity1  
From Setup>> Search For Apex Classes>> Click on "protection" behind "PropertyHandlerLWC".
2. From Profiles Add "Manager" and "Customer" and "Save".

Search Setup

Setup

Home

Object Manager

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

Help for this Page

Save

Cancel

Available Profiles

Enabled 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

Customer

Manager

System Administrator