

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://www.jotform.com/build/243218794664063>

Create

Directly Creating Objects

Creating

Dreams World

Name *

First Name _____ Last Name _____

Email _____
example@example.com

Phone Number _____
(000) 000-0000
Please enter a valid phone number.

Which type of Property are you looking for?

RESIDENTIAL
 COMMERCIAL
 RENTAL

Budget Amount *

e.g., 23

Address

Street Address _____
Street Address Line 2 _____

City _____ State / Province _____
Postal / Zip Code _____

Submit

Objects from Spreadsheet

from Spreadsheet in Salesforce

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.com	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.com	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 for creating a new object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Customer'. On the left, a sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages (which is selected), Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Restriction Rules. The main 'Details' tab displays the following information:

Description	API Name: Customer_c
Custom	✓
Singular Label	Customer
Plural Label	Customer

On the right, there are sections for 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History', 'Deployment Status' (Deployed), and 'Help Settings' (Standard salesforce.com Help Window). At the bottom right are '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	unchecked
3 Bhk fk at st	rental	Jubilee hill Hyd	Checked

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

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. A specific object named 'Property' is being viewed. The left sidebar lists various configuration options under 'Record Types'. The main 'Details' tab displays the following information:

Field	Value
Description	
API Name	Property_c
Custom	✓
Singular Label	Property
Plural Label	Property
Enable Reports	✓
Track Activities	✓
Track Field History	
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Buttons for 'Edit' and 'Delete' are located at the top right of the detail pane.

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. On the left, there is a sidebar with various settings and integration options. The main area is titled "SALESFORCE" and shows a configuration for "Customer" objects. It includes a "Create a record" section where form fields are mapped to Salesforce fields. The mappings are as follows:

Object Fields	Dreamhome
Customer__c	Name - First Name
City	Address - City
Budget Amount	Budget Amount
Property Type	Which type of property are you lookin...
Phone Number	Phone Number
Street Address	Address - Street Address
Email	Email
Customer Name	Name - Last Name
State	Address - State
Street Address line 2	Address - Street Address 2

Then “Save the Integration” and “Finish”.

The screenshot shows the completed integration setup in Jotform. The "SALESFORCE" section now displays a single action: "Create or update a record Customer". There are buttons for "See Action Logs" and "+ Add New Action".

Create Roles

here we need to Create Roles as per business requirement

Activity:- 1

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

The screenshot shows the Salesforce Setup interface for managing roles. On the left, there's a sidebar with navigation links: Opportunities, Service (expanded), Case Teams (expanded), Case Team Roles, Contact Roles on Cases, and a search bar. The main area displays a hierarchical list of roles under 'Add Role'. The roles listed are:

- SVP Customer Service & Support** Edit | Del | Assign
- Customer Support, International** Edit | Del | Assign
 - Add Role
- Customer Support, North America** Edit | Del | Assign
 - Add Role
- Installation & Repair Services** Edit | Del | Assign
 - Add Role
- SVP, Human Resources** Edit | Del | Assign
 - Add Role
- SVP, Sales & Marketing** Edit | Del | Assign
 - Add Role
 - VP, International Sales** Edit | Del | Assign
 - Add Role
 - VP, Marketing** Edit | Del | Assign
 - Add Role
 - Marketing Team** Edit | Del | Assign
 - Add Role
 - VP, North American Sales** Edit | Del | Assign
 - Add Role
 - Director, Channel Sales** Edit | Del | Assign
 - Add Role
 - Channel Sales Team** Edit | Del | Assign
 - Add Role
 - Director, Direct Sales** Edit | Del | Assign
 - Add Role
 - Eastern Sales Team** Edit | Del | Assign
 - Add Role
 - Western Sales Team** Edit | Del | Assign
 - Add Role
 - Sales Representative** Edit | Del | Assign
 - Add Role
 - Sales Executive** Edit | Del | Assign
 - Add Role
 - Sales Manager** Edit | Del | Assign
 - Add Role
 - Customer** Edit | Del | Assign
 - Add Role

Save **Save & New** **Cancel**

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 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.

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 i

*Developer Name i

Description i

App Branding

Image i

Primary Color Hex Value i

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 allthe Custom Objects and Check onlyProperty Details From Custom App Settings.
3. so RemovealltheStandard Object Permissions
4. Uncheck all the Custom Object Permissions and check read and view all in “Property”
5. make sure every submissionobject permissions are unselected and then save

The screenshot shows the Salesforce Setup interface under the Profiles section. It displays two main sections: Standard Object Permissions and Custom Object Permissions, both for the Customer profile.

Standard Object Permissions: This section lists various standard objects like Accounts, Assets, and Contact Point Types, each with five permission levels: Read, Create, Edit, Delete, and Modify. For most objects, the "Modify All" checkbox is checked. The Customer object has its "Modify All" checkboxes unchecked for Read, Create, Edit, and Delete, but checked for View All.

Custom Object Permissions: This section lists custom objects such as Customer, Property, and Vehicles. For the Customer object, the "Modify All" checkboxes are checked for Read, Create, Edit, and Delete, but unchecked for View All. The Property object has its "Modify All" checkboxes checked for all four levels (Read, Create, Edit, Delete). The Vehicles object has its "Modify All" checkboxes unchecked for all four levels.

Session Settings: At the bottom, it shows session times out after 2 hours of inactivity and a session security level required at login set to None.

Creating Manager Profiles :-

1. From Setup » Go to Profiles and Clone Salesforce Platform User and Name it “Manager”.
2. Uncheck allthe Custom Objects and Check onlyProperty Details From Custom App Settings.
3. Also Remove all the Standard Object Permissions.
4. Uncheck allthe Custom ObjectPermissions and check only“modify all” from “Property” and“Customer”.

The screenshot shows the Salesforce Setup interface under the Profiles section, similar to the previous one but for the Manager profile. It displays Standard Object Permissions and Custom Object Permissions for the Manager profile.

Standard Object Permissions: This section lists various standard objects like Accounts, Assets, and Contact Point Types, each with five permission levels: Read, Create, Edit, Delete, and Modify. For most objects, the "Modify All" checkbox is checked. The Customer object has its "Modify All" checkboxes unchecked for Read, Create, Edit, and Delete, but checked for View All.

Custom Object Permissions: This section lists custom objects such as Customer, Property, and Vehicles. For the Customer object, the "Modify All" checkboxes are checked for all four levels (Read, Create, Edit, Delete). The Property object has its "Modify All" checkboxes checked for all four levels. The Vehicles object has its "Modify All" checkboxes unchecked for all four levels.

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 shows the 'Custom Field Definition Detail' page for a custom field named 'Verified'. The field is defined on the 'User' object and has a data type of 'Checkbox'. The 'Field Information' section includes details like Field Label ('Verified'), Field Name ('Verified'), API Name ('Verified_c'), and Data Type ('Checkbox'). The 'General Options' section shows the field was created by 'Alturi Poojitha' on 18/11/2024 at 12:05 pm and modified by the same user at the same time.

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

User Detail

		Edit	Sharing	Reset Password	Freeze	View Summary
Name	Executive					Role Sales Executive
Alias	eexec					User License Salesforce
Email	atuirpoojitha.002@gmail.com [Verify]					Profile System Administrator
Username	atuirpoojitha2004@gmail.com					Active <input checked="" type="checkbox"/>
Nickname	User17319152908567166227					Marketing User <input type="checkbox"/>
Title						Offline User <input type="checkbox"/>
Company						Knowledge User <input type="checkbox"/>
Department						Flow User <input type="checkbox"/>
Division						Service Cloud User <input type="checkbox"/>

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

SETUP **Users**

User Manager

[User Profile Help for this Page](#) [?](#)

Edit		Sharing	Reset Password	Freeze	View Summary	
Name	Manager					Role Sales Manager
Alias	mana					User License Salesforce Platform
Email	atuirpoojitha.002@gmail.com [Verify]					Profile Manager
Username	poojitha2004@gmail.com					Active <input checked="" type="checkbox"/>
Nickname	User17319156117326079874					Marketing User <input type="checkbox"/>
Title						Offline User <input type="checkbox"/>
Company						Knowledge User <input type="checkbox"/>
Department						Flow User <input type="checkbox"/>
Division						Service Cloud User <input type="checkbox"/>
Address						Site.com Contributor User <input type="checkbox"/>
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)					Site.com Publisher User <input type="checkbox"/>
Locale	English (India)					WDC User <input type="checkbox"/>
Language	English					Mobile Push Registrations View
Delegated Approver						Data.com User Type View
Manager						Accessibility Mode (Classic Only) <input type="checkbox"/> View
Receive Approval Request Emails	Only if I am an approver					Debug Mode <input type="checkbox"/> View
Federation ID						High-Contrast Palette on Charts <input type="checkbox"/> View
Ann Registration: One-Time Password						Load Lightning Pages While Scrolling <input checked="" type="checkbox"/> View

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
Customer

Permission Set Assignments [0] | Permission Set Assignments: Activation Required [0] | Permission Set Group Assignments [0] | Permission Set License Assignments [0] | Personal Groups [0] | Public Group Membership [0] | Queue Membership [0] | Team [0] | Managers in the Role Hierarchy [2] | OAuth Apps [0] | Third-Party Account Links [0] | Installed Mobile Apps [0] | Authentication Settings for External Systems [0] | Login History [0+] | User Provisioning Accounts [0]

User Detail

User Detail		Edit	Sharing	Reset Password	Freeze	View Summary
Name	Customer	Role	Customer			
Alias	cust	User License	Salesforce Platform			
Email	aturipoonitha.002@gmail.com [Verify]	Profile	Customer			
Username	aturi2004@gmail.com	Active	<input checked="" type="checkbox"/>			
Nickname	User17319157277164315141	Marketing User	<input type="checkbox"/>			
Title		Offline User	<input type="checkbox"/>			
Company		Knowledge User	<input type="checkbox"/>			
Department		Flow User	<input type="checkbox"/>			
Division		Service Cloud User	<input type="checkbox"/>			

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

User
Customer2

Permission Set Assignments [0] | Permission Set Assignments: Activation Required [0] | Permission Set Group Assignments [0] | Permission Set License Assignments [0] | Personal Groups [0] | Public Group Membership [0] | Queue Membership [0] | Team [0] | Managers in the Role Hierarchy [2] | OAuth Apps [0] | Third-Party Account Links [0] | Installed Mobile Apps [0] | Authentication Settings for External Systems [0] | Login History [0+] | User Provisioning Accounts [0]

User Detail

User Detail		Edit	Sharing	Reset Password	Freeze	View Summary
Name	Customer2	Role	Customer			
Alias	cust	User License	Salesforce Platform			
Email	aturipoonitha.002@gmail.com [Verify]	Profile	Customer			
Username	aturi@gmail.com	Active	<input checked="" type="checkbox"/>			
Nickname	User17319158513339399989	Marketing User	<input type="checkbox"/>			
Title		Offline User	<input type="checkbox"/>			
Company		Knowledge User	<input type="checkbox"/>			
Department		Flow User	<input type="checkbox"/>			
Division		Service Cloud User	<input type="checkbox"/>			
Address		Site.com Contributor User	<input type="checkbox"/>			
Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)	Site.com Publisher User	<input type="checkbox"/>			
Locale	English (India)	WDC User	<input type="checkbox"/>			
Language	English	Mobile Push Registrations	View			
Delegated Approver		Data.com User Type				
Manager		Accessibility Mode (Classic Only)	<input type="checkbox"/>			

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- 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.

Approval Process Edit Property Approval

Step 5. Select Fields to Display on Approval Page Layout

The approval page is where an approver will actually approve or reject a request. Using the options below, choose the fields to display on this page.

Available Fields

- Created By
- Last Modified By
- Verified

Add Remove

Selected Fields

- Property
- Owner
- Location
- Property Name
- Type

Up Down



[Click here to view an example](#)

1. Click Next and Select the initial Submitters »

2. Owner >> Property Owner
1. Roles>> Sales Manager
2. 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.

The screenshot shows the SAP Fiori Approval Processes interface for the 'Property Approval' process. The process definition details include:

- Process Name:** Property Approval
- Unique Name:** Property_Approval
- Description:** (Property: Location NOT EQUAL TO blank) AND (Property: Verified equals false)
- Entry Criteria:** (Property: Location NOT EQUAL TO blank) AND (Property: Verified equals false)
- Record Editability:** Administrator OR Current Approver
- Next Automated Approver Determined By:** ✓
- Allow Submitters to Recall Approval Requests:**
- Approval Assignment Email Template:** (No template assigned)
- Initial Submitters:** Sales Manager, Property Owner
- Created By:** 21551A0556 JALADI PRAVEEN KUMAR, 17/11/2024, 11:32 am
- Modified By:** 21551A0556 JALADI PRAVEEN KUMAR, 17/11/2024, 9:30 pm

The process includes the following steps:

- Initial Submission Actions:**
 - Action: Record Lock, Description: Lock the record from being edited.
- Approval Steps:**

Action	Type	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions	Edit	1	Executive Approval	(Property: Location NOT EQUAL TO blank) AND (Property: Verified equals false)		User/Executive	Final Rejection
- Final Approval Actions:**
 - Action: Record Lock, Description: Lock the record from being edited.
 - Action: Remove, Type: Field Update, Description: Verified_Property
- Final Rejection Actions:**
 - Action: Record Lock, Description: Unlock the record for editing.
 - Action: Remove, Type: Field Update, Description: Unverified_Property
- Recall Actions:**
 - Action: Record Lock, Description: Unlock the record for editing.

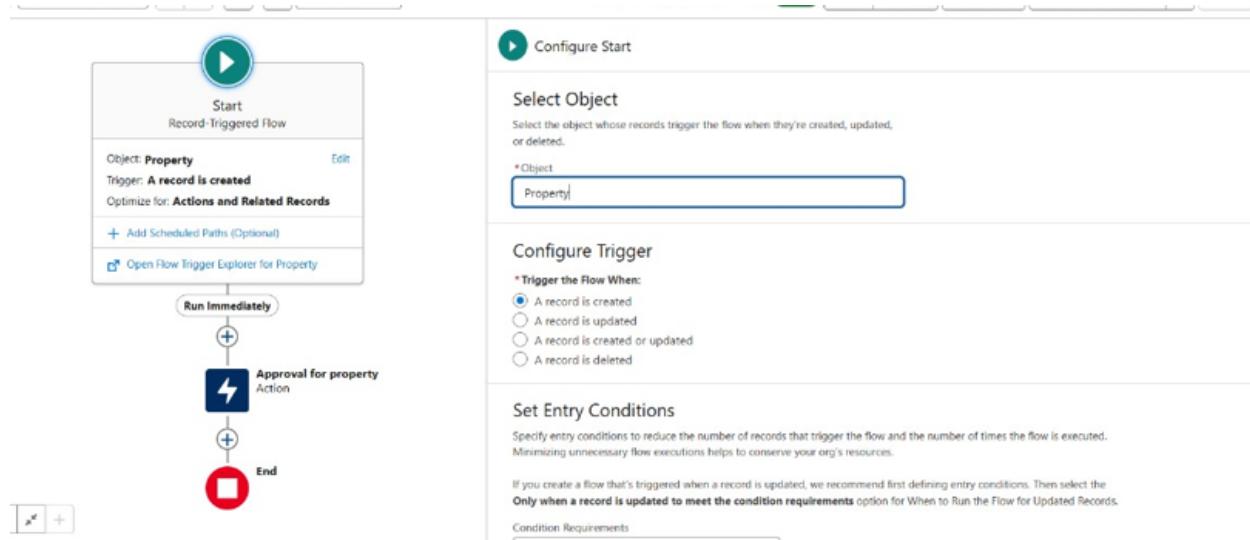
Create a Recordtrigger now to submit the Approval Process Automatically

A flow that can submit the recordsdirectly for approval

Activity1

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

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

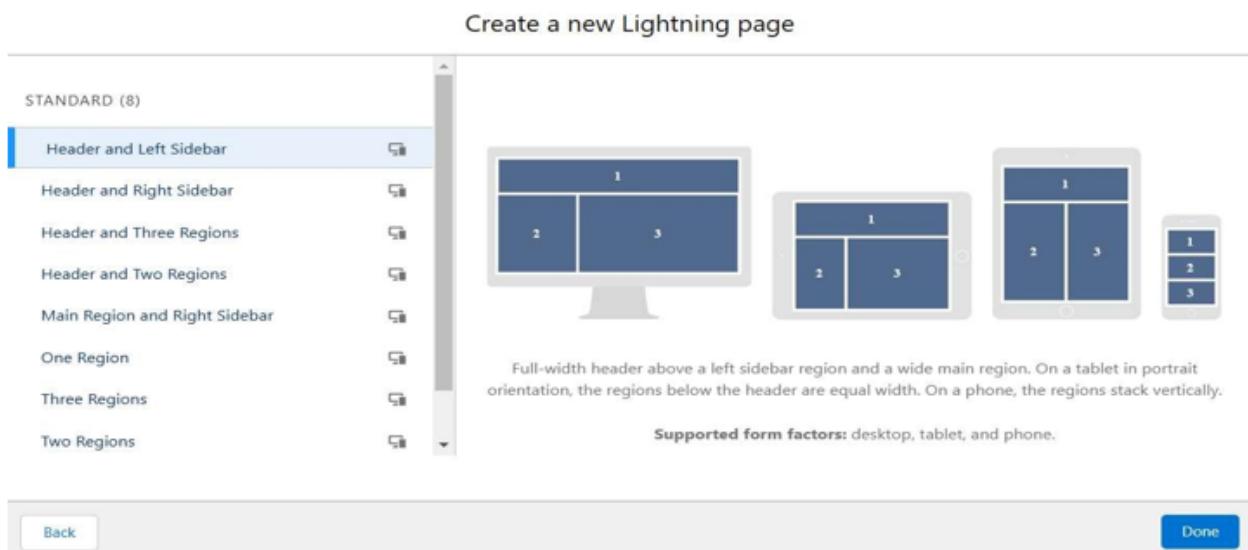


Create an App Page

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 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”



Activation: select your property

PAGE SETTINGS **LIGHTNING EXPERIENCE** MOBILE NAVIGATION

Add this app page to Lightning Experience apps. You can manage Lightning apps in Setup.

Add to Lightning Apps

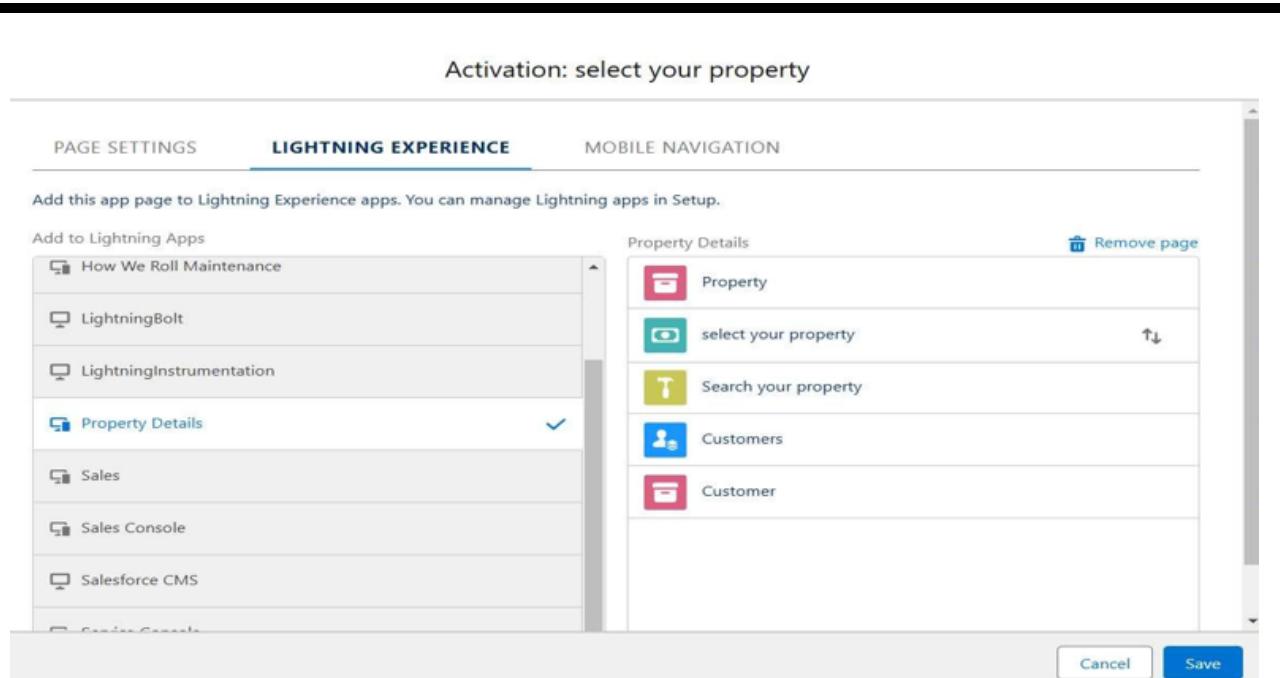
How We Roll Maintenance
LightningBolt
LightningInstrumentation
Property Details <input checked="" type="checkbox"/>
Sales
Sales Console
Salesforce CMS
Customer

Property Details

Property
select your property
Search your property
Customers
Customer

Remove page

Cancel 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<Propertyc> 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 public class PropertyHandler_LWC {
2     @AuraEnabled(cacheable=true)
3     public static List<Property__c> getProperty(String type, Boolean verified) {
4         String verifiedStr = verified ? 'true' : 'false'; // Convert boolean to string
5         return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c
6                 FROM Property__c
7                 WHERE Type__c = :type AND Verified__c = :verifiedStr];
8     }
9 }

```

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 :-

```

<tempIate>
<lightning-card>
<div class="sIds-box">
<div class="sIds-text-align_left">
<h1 style="font-size: 20px;"><b>Properties</b></h1>
</div>
<div>
<div class="sIds-grid sIds-gutters">
<div class="sIds-col sIds-size_5-of-6">
<lightning-combobox name="Type" label="Property Type" value={typevar}
placeholder="Select Property type"
options={propertyoptions} onchange={changehandler}></lightning-combobox>
</div>
<div class="sIds-col sIds-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={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>
```

```

<template>
  <lightning-card>
    <div class="slds-box">
      <div class="slds-text-align_left">
        <h1 style="font-size: 20px;"><b>Properties</b></h1>
      </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={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>
      <template if:true={istrue}>
        <div class="slds-box">

```

1. In YourJs File Write this code :-

Code :-

```

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

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_M anagementextends 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:
    "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')
    changehandIer(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.isfaIse = false;
                    this.istrue = false;
                .catch((error) => (
                    console.log(error)
```

```
force-app > main > default > lwc > property > property.js > C_01_Property_Management > property.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
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' }
34
35     ];
36
37     propertyoptions = [
38
39         { label: "Commercial", value: "Commercial" },
40
41         { label: "Residential", value: "Residential" },
42
43         { label: "rental", value: "rental" }
44
45     ];
46
47     @wire(getRecord, { recordId: "SuserId", Fields: ["User.Verified__c"] })
48     recordFunction({ data, error }) {
49
50         if (data) {
51
52             console.log(data);
53
54             console.log("This is the User Id ---> " + this.userId);
55
56             this.verifiedvar = data.fields.Verified__c.value;
57
58         }
59     }
60 }
```

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</targets>
        <target>lightning__AppPage</targets>
        <target>lightning__HomePage</targets>
    </targets>
</LightningComponentBundle>
```

```
<?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 componentto the org

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://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>
```

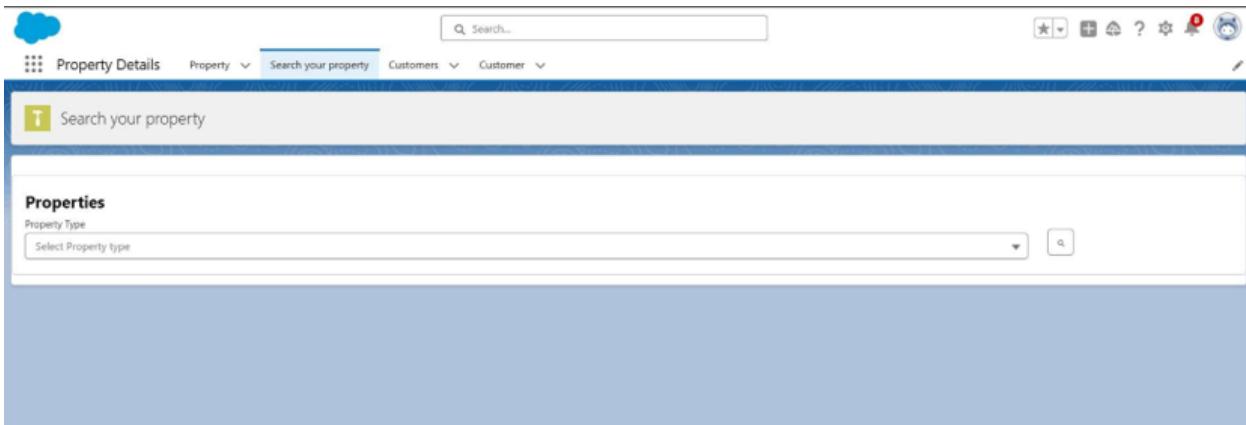
Drag this Component to your App Page

Adding the Componentto your Page

Activity1

1. From Setup >> Go to App Launcher >> Searchfor PropertyDetails
2. On thisPage click on gear icon andclick on EditPage
3. after clicking on edit pageit will bnedirected to apppages then

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".

