

A CRM Application to Handle the Clients and their property Related Requirements

Project Description:

Dreams World Properties integrates Salesforce to streamline customer interactions. Website engagement triggers automated record creation in Salesforce, capturing customer details and preferences. Salesforce categorizes users as approved or non-approved, offering tailored property selections to approved users. This enhances user experience and efficiency, providing personalized recommendations and broader listings. Seamless integration optimizes operations, improving customer engagement and facilitating growth in the real estate market.

1. Client Management

- a. Add, update, and delete client details.
- b. Track client preferences, budget, and location interests.
- c. Maintain contact details and communication history.

2. Property Management

- a. Manage property listings with details like type, price, location, and features.
- b. Track properties available for sale, rent, or lease.
- c. Upload photos and documents for properties.

3. Requirement Matching

- a. Match client requirements with available properties using filters.
- b. Notify clients about new properties that fit their criteria.

4. Lead Tracking

- a. Manage inquiries and follow up with potential clients.
- b. Schedule meetings and site visits.
- c. Assign leads to specific team members.

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

Client wants a form for the customers to get the details directly into the salesforce so that the admins can create a user in the org. Client wants a form for the customers to get the details directly into the salesforce so that the admins can create a user in the org.

Activity1

Open your browser and search for jotform and log in.

1. After login click on create form and click on start from scratch
2. Now create a form to get the customer details like Name, Phone, Email, Address and type of property the customer is interested in.
3. Once the form is created, publish it by clicking on publish.
4. form link :-<https://form.jotform.com/243218458761058>

The screenshot shows a Jotform survey titled "Dreams World". The survey has the following fields:

- Name ***: Two input fields for First Name and Last Name.
- Email**: An input field with placeholder "example@example.com".
- Which type of Property are you looking for?**: Radio buttons for RESIDENTIAL, COMMERCIAL, and RENTAL.
- Phone Number**: An input field with placeholder "e.g., 23". A validation message "Please enter valid phone number" is displayed below it.
- Budget Amount ***: An input field with placeholder "e.g., 23".
- Address**: A multi-line input field for Street Address, Street Address Line 2, City, State / Province, and Postal / Zip Code.
- Submit**: A green button at the bottom right.

Create Objects from Spreadsheet

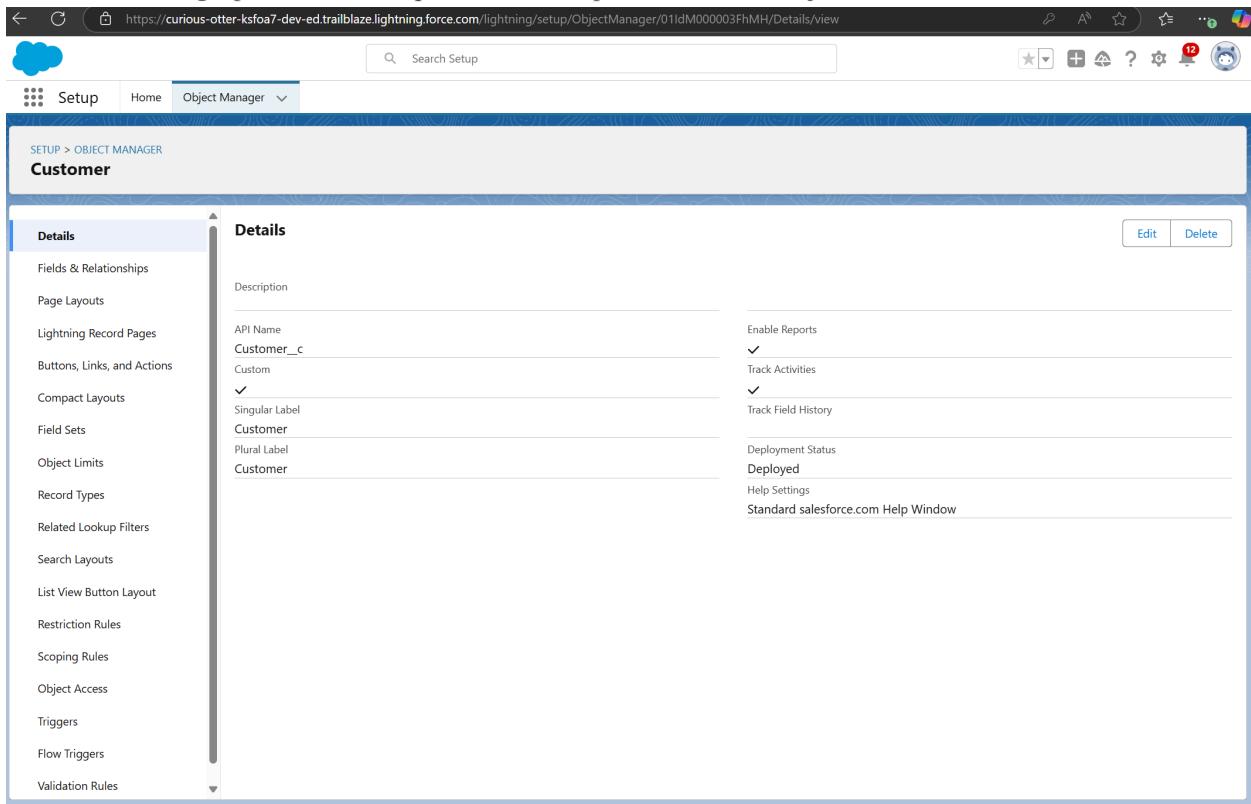
Directly Creating Objects from Spreadsheet in Salesforce

Creating Customer Object :

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

Customer	Phone Number	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	kamdi	kathora	Amravati	444805	checked

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



The screenshot shows the Salesforce Object Manager interface. The URL in the browser is https://curious-otter-ksfoa7-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01dM000003FhMH/Details/view. The page title is "Customer". On the left, there's a sidebar with various setup options like Fields & Relationships, Page Layouts, Lightning Record Pages, etc. The main content area is titled "Details" and contains sections for "Description", "API Name" (Customer__c), "Custom", "Singular Label" (Customer), "Plural Label" (Customer), and "Enable Reports" (checkbox checked). At the bottom right of the main area 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 fkat at st	rental	Jubliee hill Hyd	Checked

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

The screenshot shows the Salesforce Setup interface under 'Object Manager'. On the left, a sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Buttons, Links, and Actions. The main panel displays the 'Details' section for the 'Property' object. It includes fields for API Name (Property__c), Singular Label (Property), and Plural Label (Property). Other settings shown include Enable Reports (checked), Track Activities (checked), and Track Field History. Deployment Status is set to 'Deployed'.

Integrate Jotformwith Salesforce Platform

In this Milestone we are going to integratejotform 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”.

Then “Save the Integration” and “Finish”.

Create Roles

here we need to Create Roles as per business requirement

Activity:- 1

The screenshot shows the Salesforce Setup Roles page. The left sidebar navigation includes: Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, Administration (with sub-options like Users, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, and Users), Data, and Email.

The main content area displays a hierarchical list of roles:

- Customer Support, North America (Edit | Del | Assign)
- 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

1. if we don't find sales representative we need to create it according to the need
2. It will use the "System Administrator Profile".
3. Label -Sales Executive
4. Reports to- Sales Representative

Similarly Create a Role Name "Sales Manager" below Sales Executive which reports to Sales Executive, Also Add a Role below Sales Manager labeled as "Customer" which reports to Sales Manager.

Create a Property Details App

An App where the objects will be displayed

Activity1

1. From Setup>> Go to 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.

The screenshot shows the 'App Details & Branding' section of the Lightning App Builder. On the left, a sidebar lists 'App Settings' with 'App Details & Branding' selected. The main area has two tabs: 'App Details' and 'App Branding'. In 'App Details', fields include 'App Name' (Property Details), 'Developer Name' (Property_Details), and 'Description' (Enter a description...). In 'App Branding', there's an 'Image' field with an 'Upload' button, a 'Primary Color Hex Value' field (#0070D2), and an 'Org Theme Options' checkbox (unchecked). Below these, an 'App Launcher Preview' shows a blue square with 'PD' and the text 'Property Details'.

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

The screenshot shows the Salesforce Setup Profiles page. On the left, the sidebar includes links like Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, Administration (Users, Permission Set Groups, Permission Sets, Profiles), Data, and Email. The main content area is titled "Profiles". It contains sections for "Communication Subscription Timings" (with rows for Contacts, Contact Point Addresses, Contact Point Consents, Contact Point Emails, Contact Point Phones, Party Consents, Push Topics, Sellers, Streaming Channels, User External Credentials), "Custom Object Permissions" (with tables for Customer and Property), "Session Settings" (Session Times Out After: 2 hours of inactivity, Session Security Level Required at Login: None), and "Password Policies" (User passwords expire in: 90 days, Enforce password history: 3 passwords remembered, Minimum password length: 8, Password complexity requirement: Must include alpha and numeric characters, Password question requirement: Cannot contain password, Maximum invalid login attempts: 10, Lockout effective period: 15 minutes, Obscure secret answer for password resets, Require a minimum 1 day password). The "Profiles" link in the sidebar is highlighted.

Creating Manager Profiles :-

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

The screenshot shows the Salesforce Setup interface with the URL <https://curious-otter-ksfoa7-dev-ed.trailblaze.lightning.force.com/lightning/setup/EnhancedProfiles/page?address=%2F00edM000008Fzor%2Fe%3FretURL%3D%252F00e...>. The page title is "SETUP Profiles". The left sidebar includes links for Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, Administration (Users, Permission Set Groups, Permission Sets, Profiles), Data, and Email. The main content area displays sections for Communication Subscription Timings, Contacts, Contact Point Addresses, Contact Point Consents, Contact Point Emails, Contact Point Phones, Party Consents, Push Topics, Sellers, Streaming Channels, and User External Credentials. Below these are sections for Custom Object Permissions (Customer and Property profiles) and Session Settings (Session Times Out After: 2 hours of inactivity, Session Security Level Required at Login: None). The Password Policies section includes fields for User passwords expire in (90 days), Enforce password history (3 passwords remembered), Minimum password length (8), Password complexity requirement (Must include alpha and numeric characters), Password question requirement (Cannot contain password), Maximum invalid login attempts (10), Lockout effective period (15 minutes), Obscure secret answer for password resets, and Require a minimum 1 day password.

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 Salesforce Object Manager interface. On the left, a sidebar lists various setup options like User Page Layouts, Lightning Record Pages, and Validation Rules. The main content area is titled 'User Custom Field Verified' and shows the 'Custom Field Definition Detail' for the 'Verified' field. The field is of type 'Checkbox' and has a default value of 'Unchecked'. There are no validation rules defined.

Create Users

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

User : 1

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

User : 2

1. Goto Setup > Administration >> Users >>New User
2. Last Name >> Manager
3. Role >> Sales Manager

4. License >> Salesforce Platform
5. Profile >> Manager
6. Save

User : 3

1. Go to Setup»>Administration »> Users »> New User
2. Last Name » Customer
3. Role >> Customer
4. License»>Salesforce Platform
5. Profile»>Customer
6. Make Sure the 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

Action	Full Name	Alias	Username	Role	Active	Profile
Edit	Chatter Expert	Chatter	chatty.00ddm00000fxztual.0a55tmicbc2w@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
Edit	Customer	cust	user31@nvp.com	Customer	<input checked="" type="checkbox"/>	Customer
Edit	Customer2	custi	user42@nvp.com	Customer	<input checked="" type="checkbox"/>	Customer
Edit	Executive	exec	user12@nvp.com	Sales Executive	<input checked="" type="checkbox"/>	System Administrator
Edit	GOPU AASRITHA	AGOPU	21501a0557@curious-otter-ksfoa7.com		<input checked="" type="checkbox"/>	System Administrator
Edit	Manager	mana	user23@nvp.com	Sales Manager	<input checked="" type="checkbox"/>	Manager
Edit	User_Integration	integ	integration@00ddm00000fxztual.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
Edit	User_Security	sec	insightssecurity@00ddm00000fxrxtual.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

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.

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

The screenshot shows the Salesforce Setup interface under the 'Approval Processes' section for the 'Property' object. At the top, there's a help section with steps 1 through 7. Below it, a message says '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.' A button labeled 'Create New Approval Process' is visible. The main area is divided into two sections: 'Active Approval Processes' and 'Inactive Approval Processes'. The 'Active Approval Processes' section contains one entry: 'Property Approval' with an 'Edit | Deactivate' link. The 'Inactive Approval Processes' section shows 'No approval processes available'.

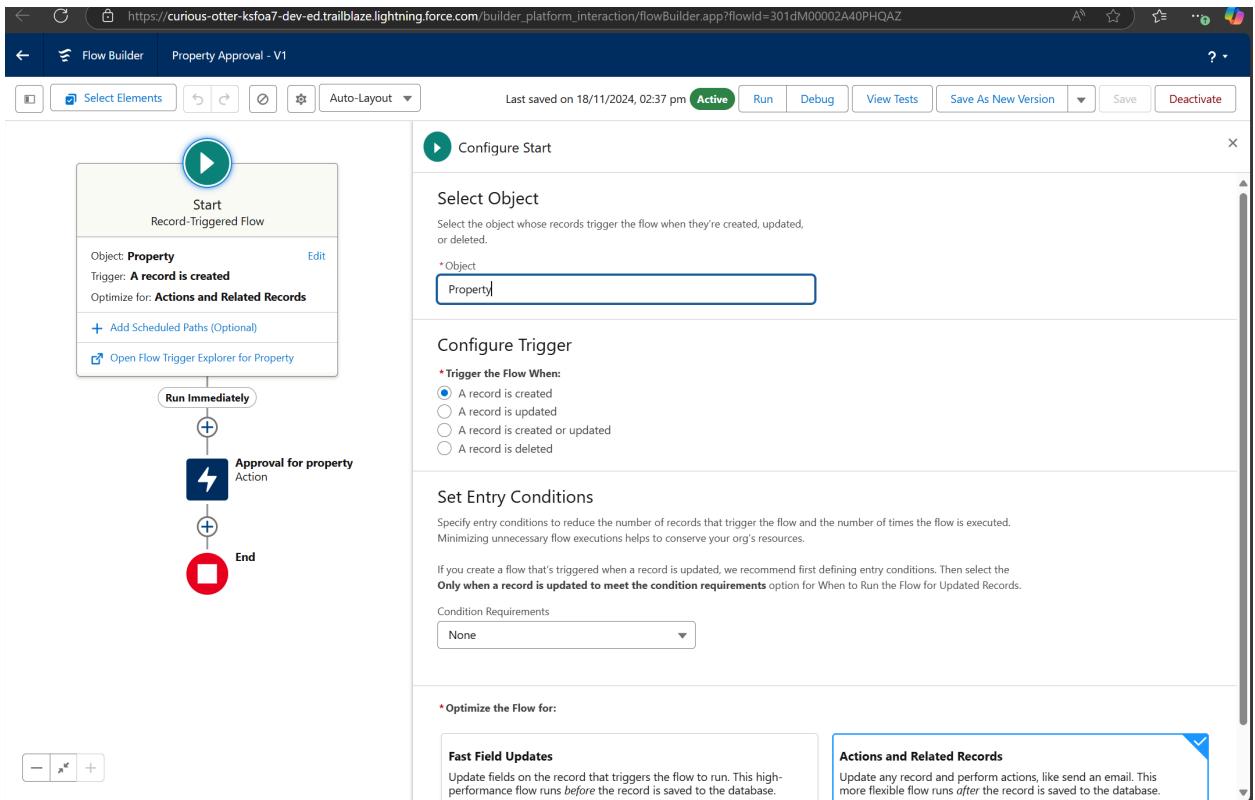
Create a Recordtrigger ñow to submit the Approval Process Automatically

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 Setting select page activation as “Activate for all Users”.
7. From Lightning Experience Click on “Property Details” and click on Add Page“.
8. Then Click on “Save”

Create a 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.

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
- Sales
- Sales Console
- Salesforce CMS

Property Details

Remove page

- Property
- select your property
- Search your property
- Customers
- Customer

Cancel Save

Create a LWC Component

- 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

- 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> getPropert(String type, Boolean verified) {
        String verifiedstr = verified ? 'true' : 'false' // Convert boolean to string return [SELECT Id,
        Location c, Propert_Namec, 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         // Convert the Boolean value to String
5         String verifiedString = verified ? 'true' : 'false';
6
7         // Update the query to use the String value
8         return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c
9                 FROM Property__c
10                WHERE Type__c = :type AND Verified__c = :verifiedString];
11    }
12 }

```

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 :-

```

<tempIate>
<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={changehandler}></lightning-combobox>
</div>

<div class="slds-col slds-size_1-of-6">
    <br>
    <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search"
    label="Search" onclick={handleClick}></lightning-button-icon>
</div>
</div>
</div>

<tempIate if:true={istrue}>
    <div class="slds-box">
        <lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-
datatable>
    </div>
</tempIates

<tempIate if:false={isfalse}>
    <div class="slds-box">
        <div style="font-size: 15px;"><b>No properties Are Found !!</b></div>
    </div>
</tempIate
</lightning-card>
</tempIates>

```

```

<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"
            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>
    </div>
    <template if:true={istru}>

```

1. In YourJs File Write this code :-

Code :-

```

import ( LightningElement, api, track, wire ) from 'lwc';
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 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')
  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.istru = true; this.propertylist =
          result; console.log(this.verifiedvar);
          console.log(this.typevar)
        ) else (
          this.isfaIse = false;
          this.istru = false;
        .catch((error)=> (
          console.log(error)

```

```

force-app > main > default > lwc > light > js light.js > ...
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 ]

```

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>

```

```

force-app > main > default > lwc > light > light.js-meta.xml > LightningComponentBundle
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
4
5      <apiVersion>59.0</apiVersion>
6
7      <isExposed>true</isExposed>
8
9      <targets>
10
11         <target>lightning__RecordPage</target>
12             <target>
13                 Enables a component to be used on a Home page in Lightning App Builder.
14                 Source: js-meta.xsd
15             <target>lightning__HomePage</target>
16
17         </targets>
18
19     </LightningComponentBundle>

```

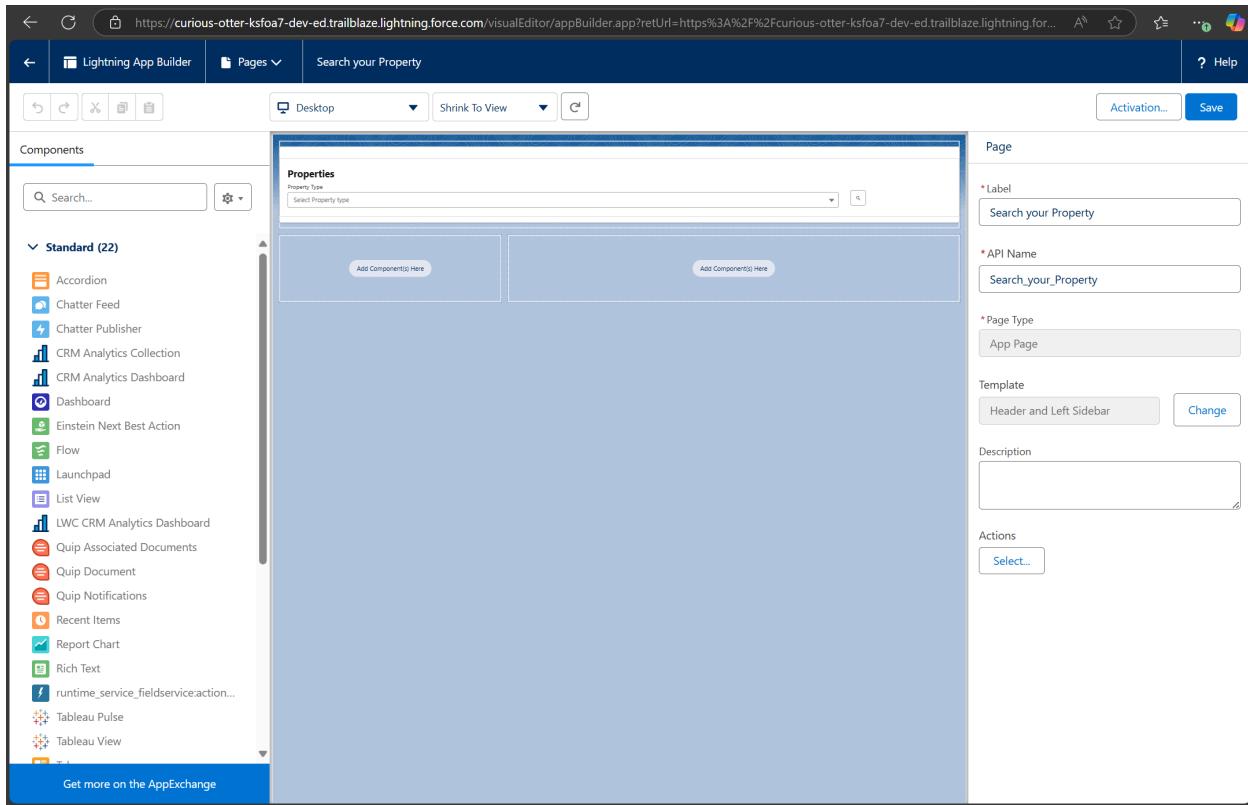
After Saving all the three Codes , Right Click and deploy this componentto the org

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
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".

The screenshot shows the Salesforce Setup interface. The left sidebar is titled "Setup" and includes sections for Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, Administration, and Users. Under "Users", the "Profiles" tab is selected. The main content area is titled "Profiles" and shows the "Enable Profile Access for Apex Class" page for "PropertyHandler_LWC". The page has two main sections: "Available Profiles" (listing various standard and custom profiles) and "Enabled Profiles" (listing "Customer Manager" and "System Administrator"). There are "Add" and "Remove" buttons between the two sections. A "Save" and "Cancel" button are at the top right of the form.