Property Object

SELECT Additional\_Room\_\_c,Area\_\_c,Available\_From\_\_c,Carpet\_Area\_Sqft\_\_c,

Country\_\_c,Covered\_Area\_Sqft\_\_c,CreatedById,CreatedDate,Electricity\_Status\_\_c,

Facing\_\_c,Floor\_\_c,For\_Bachellor\_\_c,For\_Family\_\_c,Furnished\_Type\_\_c,

Furnishing\_Available\_\_c,Geo\_Map\_\_c,Geo\_Map\_\_Latitude\_\_s,Geo\_Map\_\_Longitude\_\_s,

Id,IsDeleted,Landmark\_\_c,LastActivityDate,LastModifiedById,LastModifiedDate,

LastReferencedDate,LastViewedDate,Location\_\_c,Maintainance\_Charge\_\_c,

Name,No\_of\_Balcony\_\_c,No\_of\_Bath\_Room\_\_c,No\_Of\_Bed\_Room\_\_c,OwnerId,

Property\_Description\_\_c,Property\_Main\_Image\_\_c,Property\_Owner\_\_c,

Property\_Video\_URL\_\_c,Rent\_\_c,Security\_Deposit\_\_c,Status\_\_c,street\_\_c,

SystemModstamp,Verified\_\_c,Water\_Availability\_\_c FROM Property\_\_c

Property Owner object

SELECT CreatedById,CreatedDate,Email\_\_c,Id,IsDeleted,LastActivityDate,LastModifiedById,LastModifiedDate,LastReferencedDate,LastViewedDate,Name,OwnerId,Owner\_Address\_\_c,Owner\_Name\_\_c,Owner\_Phone\_\_c,Owner\_Photo\_\_c,SystemModstamp FROM Property\_Owner\_\_c

Property Enquiry object

SELECT CreatedById,CreatedDate,Email\_\_c,Id,IsDeleted,LastActivityDate,LastModifiedById,LastModifiedDate,LastReferencedDate,LastViewedDate,Message\_\_c,Name,Name\_\_c,OwnerId,Phone\_\_c,Property\_\_c,SystemModstamp FROM Property\_enquiry\_\_c

[**PropertyDetails**](https://ono-c-dev-ed.my.salesforce.com/01p5g000000qgF2) **Apex utility class**

public with sharing class PropertyDetails {  
    public PropertyDetails() {  
  
    }  
    @AuraEnabled(cacheable=true)  
    public static list<Property\_\_c> getLatestProperty(){  
        list<Property\_\_c> allPropertyList = new list<Property\_\_c>();  
        allPropertyList = [SELECT   Id,  
                            Additional\_Room\_\_c,  
                            Available\_From\_\_c,            
                            Carpet\_Area\_Sqft\_\_c,  
                            Covered\_Area\_Sqft\_\_c,  
                            Electricity\_Status\_\_c,  
                            Facing\_\_c,  
                            Floor\_\_c,  
                            For\_Bachellor\_\_c,  
                            For\_Family\_\_c,  
                            Furnishing\_Available\_\_c,  
                            Furnished\_Type\_\_c,  
                            Geo\_Map\_\_c,  
                            Geo\_Map\_\_Latitude\_\_s,  
                            Geo\_Map\_\_Longitude\_\_s,  
                            Landmark\_\_c,  
                            Location\_\_c,  
                            Area\_\_c,  
                            street\_\_c,  
                            Country\_\_c,  
                            Maintainance\_Charge\_\_c,  
                            No\_of\_Balcony\_\_c,  
                            No\_of\_Bath\_Room\_\_c,  
                            No\_Of\_Bed\_Room\_\_c,  
                            Property\_Description\_\_c,  
                            Property\_Main\_Image\_\_c,  
                            Property\_Owner\_\_c,  
                            Property\_Video\_URL\_\_c,  
                            Name,  
                            Rent\_\_c,  
                            Security\_Deposit\_\_c,  
                            Status\_\_c,  
                            Verified\_\_c,  
                            Water\_Availability\_\_c  
                        FROM Property\_\_c  
                        LIMIT 50  
                        ];  
            return allPropertyList;              
    } @AuraEnabled(cacheable=true)  
    public static list<Property\_\_c> getSearchedProperty(string location,string bedroom,string bathroom,string maxbudget){  
        list<Property\_\_c> allPropertyList = new list<Property\_\_c>();  
        string propertyQuery = 'SELECT Id,Additional\_Room\_\_c,Available\_From\_\_c,Carpet\_Area\_Sqft\_\_c,Covered\_Area\_Sqft\_\_c,Electricity\_Status\_\_c,';  
        propertyQuery = propertyQuery+'Facing\_\_c,Floor\_\_c,For\_Bachellor\_\_c,For\_Family\_\_c,Furnished\_Type\_\_c,Furnishing\_Available\_\_c,Geo\_Map\_\_c,';  
        propertyQuery = propertyQuery +'Landmark\_\_c, Area\_\_c,street\_\_c,country\_\_c,Location\_\_c,Maintainance\_Charge\_\_c,No\_of\_Balcony\_\_c,No\_Of\_Bath\_Room\_\_c,No\_Of\_Bed\_Room\_\_c,';  
        propertyQuery = propertyQuery +' Property\_Description\_\_c,Property\_Main\_Image\_\_c,Property\_Video\_URL\_\_c,Name,Rent\_\_c,Security\_Deposit\_\_c,Status\_\_c,Verified\_\_c,Water\_Availability\_\_c';                              
        propertyQuery = propertyQuery +' FROM Property\_\_c ';  
        string whereClause = 'WHERE Rent\_\_c!=NULL';  
        system.debug('\*\*\*\*location'+location);  
        if(string.isNotBlank(location)){  
            //string locationSearch = +location+'%';  
            if(location !='ALL'){  
                whereClause = whereClause+' AND Area\_\_c = :location';  
            }  
              
        }  
        system.debug('\*\*\*\*bedroom'+bedroom);  
        integer bedroomcount;  
        if(string.isNotBlank(bedroom)){  
            if(bedroom !='ALL'){  
                  bedroomcount= integer.valueOf(bedroom);  
                 system.debug('\*\*\*\*bedroomcount'+bedroomcount);  
            whereClause  = whereClause+' AND No\_Of\_Bed\_Room\_\_c=:bedroomcount';    
            }    
        }  
         system.debug('\*\*\*\*bathroom'+bathroom);  
          integer bathroomcount;  
        if(string.isNotBlank(bathroom)){  
             if(bathroom !='ALL'){  
                 bathroomcount= integer.valueOf(bathroom);  
                whereClause  = whereClause+' AND No\_Of\_Bath\_Room\_\_c=:bathroomcount';    
             }    
        }  
       system.debug('\*\*\*\*maxbudget'+bathroom);  
       double maxBudgetValue;  
        if(string.isNotBlank(maxbudget)){  
              
            maxBudgetValue = double.valueOf(maxbudget) ;      
            whereClause  = whereClause+' AND Rent\_\_c<=:maxBudgetValue';   
                  
        }  
        propertyQuery = propertyQuery +' '+whereClause;  
        system.debug('propertyQuery'+propertyQuery);  
        allPropertyList = Database.query(propertyQuery);  
        system.debug('allPropertyList'+allPropertyList);  
        return allPropertyList;  
    }  
    @AuraEnabled(cacheable=true)  
    public static Property\_\_c getDetails(string propId){  
        Property\_\_c prop = [SELECT Id,  
                                    Additional\_Room\_\_c,  
                                    Available\_From\_\_c,  
                                    Carpet\_Area\_Sqft\_\_c,  
                                    Covered\_Area\_Sqft\_\_c,  
                                    Electricity\_Status\_\_c,  
                                    Facing\_\_c,  
                                    Floor\_\_c,  
                                    For\_Bachellor\_\_c,  
                                    For\_Family\_\_c,  
                                    Furnishing\_Available\_\_c,  
                                    Furnished\_Type\_\_c,  
                                    Geo\_Map\_\_c,  
                                    Landmark\_\_c,  
                                    Location\_\_c,  
                                    country\_\_c,  
                                    street\_\_c,  
                                    Area\_\_c,  
                                    Maintainance\_Charge\_\_c,  
                                    No\_of\_Balcony\_\_c,  
                                    No\_Of\_Bath\_Room\_\_c,  
                                    No\_Of\_Bed\_Room\_\_c,  
                                    Property\_Description\_\_c,  
                                    Property\_Main\_Image\_\_c,  
                                    Property\_Owner\_\_c,    
                                    Property\_Video\_URL\_\_c,                                    
                                    Name,  
                                    Rent\_\_c,  
                                    Security\_Deposit\_\_c,  
                                    Status\_\_c,  
                                    Verified\_\_c,  
                                    Water\_Availability\_\_c  
                            FROM Property\_\_c  
                            WHERE Id=:propId  
                            LIMIT 50  
                            ];  
        return prop;  
    }  
}

**myPropertyResult component**

**myPropertyResult.js**

import { LightningElement, track, wire,api } from 'lwc';

import getLatestProperty from '@salesforce/apex/PropertyDetails.getLatestProperty';

import {ShowToastEvent} from 'lightning/platformShowToastEvent';

import { NavigationMixin} from 'lightning/navigation';

import getSearchedProperty from '@salesforce/apex/PropertyDetails.getSearchedProperty';

import { registerListener, unregisterAllListeners } from 'c/pubsub';

//import pubsub from 'c/pubsub';

import { CurrentPageReference } from 'lightning/navigation';

export default class MyPropertyResult extends NavigationMixin (LightningElement) {

    @track properties;

    @track propertiesFound;

    @track propid;

    @track enquiryDetails=false;

    @track feedbackPropertId;

    @track locFilter;

    @track bedroomFilter;

    @track bathroomFilter;

    @track budgetFilter;

    openOwnerDetail=false;

    propOwnerId;

    @wire(getLatestProperty)

    wiredProperties({data,error}){

        if(data){

            this.properties = data;

            this.propertiesFound = true;

        }

        else if(error){

            this.showToast('Error',error.body.message,'error');

            this.propertiesFound = false;

        }

    }

    get propertiesFound() {

        if (this.properties) {

            return true;

        }

        return false;

    }

    showToast(title,message,variant){

        const evt = new ShowToastEvent({

            title: title,

            message:message,

            variant:variant,

        });

        this.dispatchEvent(evt);

    }

    ownerDetailsClick(event){

        console.log("====================================> JVH");

        alert("Jai JVH ....JAI");

        alert(event.target.value);

        alert("Jai JVH ....JAI2");

        this.propOwnerId=event.target.value;

        this.openOwnerDetail=true;

        alert(this.openOwnerDetail);

    }

    feedbackClicked(event){

        alert("Jai JVH ....you are in Enquiry form");

        this.feedbackPropertId=event.target.value;

        this.enquiryDetails=true;

    }

    closeFeedbackModal(event){

        this.enquiryDetails=false;

    }

    @wire(CurrentPageReference) pageRef;

    connectedCallback(){

        console.log('Connected Call back ');

        registerListener('handleLocChangeEvent', this.handleLocChangeEvent,this) ;

        registerListener('handleBedChangeEvent', this.handleBedChangeEvent,this) ;

        registerListener('handleBathChangeEvent', this.handleBathChangeEvent,this) ;

        registerListener('handleBudgetChangeEvent', this.handleBudgetChangeEvent,this) ;

    }

    disconnectedCallback(){

        unregisterAllListeners(this);

    }

    handleLocChangeEvent(locchange){

        console.log('handleLocChangeEvent invoked ');

        this.locFilter = locchange;

        getSearchedProperty({

            location: this.locFilter,

            bedroom : this.bedroomFilter,

            bathroom :this.bathroomFilter,

            maxbudget: this.budgetFilter

        })

        .then(result=>{

            this.properties = result;

        })

        .catch(error =>{

            this.showToast('ERROR',error.body.message,'error');

        });

    }

    handleBedChangeEvent(bedchange){

        console.log('handleBedChangeEvent invoked ');

        this.bedroomFilter = bedchange;

        getSearchedProperty({

            location: this.locFilter,

            bedroom : this.bedroomFilter,

            bathroom :this.bathroomFilter,

            maxbudget: this.budgetFilter

        })

        .then(result=>{

            this.properties = result;

        })

        .catch(error =>{

            this.showToast('ERROR',error.body.message,'error');

        });

    }

    handleBathChangeEvent(bathchange){

        console.log('handleBathChangeEvent invoked ');

        this.bathroomFilter = bathchange;

        getSearchedProperty({

            location: this.locFilter,

            bedroom : this.bedroomFilter,

            bathroom :this.bathroomFilter,

            maxbudget: this.budgetFilter

        })

        .then(result=>{

            this.properties = result;

        })

        .catch(error =>{

            this.showToast('ERROR',error.body.message,'error');

        });

    }

    handleBudgetChangeEvent(budgetchange){

        console.log('handleBudgetChangeEvent invoked ');

        this.budgetFilter = budgetchange;

        getSearchedProperty({

            location: this.locFilter,

            bedroom : this.bedroomFilter,

            bathroom :this.bathroomFilter,

            maxbudget: this.budgetFilter

        })

        .then(result=>{

            this.properties = result;

        })

        .catch(error =>{

            this.showToast('ERROR',error.body.message,'error');

        });

    }

    NavigateToPropDetails(event){

        console.log('In Navigation .....');

        this.propid=event.target.value;

        this[NavigationMixin.Navigate]({

            type: 'standard\_\_component',

            attributes: {

                componentName:'c\_\_MyProperty360View'

            },

            state: {

                c\_\_propertyId:this.propid

            }

        });

        console.log('In Navigation .....'+this.propid);

    }

}

**myPropertyResult.html**

<template>

    <lightning-layout multiple-rows>

        <template for:each={properties} for:item="prop">

            <lightning-layout-item key={prop.Id} size="6" padding="around-small">

                <lightning-layout key={prop.Id} multiple-rows>

                    <div key={prop.Id} class="slds-box" style="width: 100%;background-color:white;">

                        <lightning-layout>

                            <lightning-layout-item size="4" padding="around-small">

                                <lightning-formatted-rich-text class="propImage" value={prop.Property\_Main\_Image\_\_c}>

                                </lightning-formatted-rich-text>

                            </lightning-layout-item>

                            <lightning-layout-item size="8" padding="around-small">

                                <lightning-card key={prop.Id} title={prop.Furnished\_Type\_\_c}>

                                    <lightning-layout-item key={prop.Id} style="color:red" padding="around-small" size="12"

                                        small-device-size="6" large-device-size="6" medium-device-size="6">

                                        INR: {prop.Rent\_\_c}

                                    </lightning-layout-item><br>

                                    <lightning-layout-item key={prop.Id} padding="around-small" size="12"

                                        small-device-size="6" large-device-size="6" medium-device-size="6">

                                        {prop.No\_Of\_Bed\_Room\_\_c} BHK House for Rent {prop.Covered\_Area\_Sqft\_\_c} sqft.

                                    </lightning-layout-item><br>

                                    <lightning-layout-item key={prop.Id} padding="around-small" size="12"

                                        small-device-size="6" large-device-size="6" medium-device-size="6">

                                        {prop.Area\_\_c}

                                    </lightning-layout-item><br>

                                    <lightning-layout-item key={prop.Id} padding="around-small" size="12"

                                        small-device-size="6" large-device-size="6" medium-device-size="6">

                                        {prop.No\_Of\_Bed\_Room\_\_c} Room {prop.No\_of\_Bath\_Room\_\_c} BathRoom

                                        {prop.Furnished\_Type\_\_c}

                                    </lightning-layout-item>

                                </lightning-card>

                                <div class="slds-p-top\_small">

                                    <lightning-button-group>

                                        <lightning-button variant="success" label="Owner Details"

                                            class="slds-m-left\_x-small" value={prop.Property\_Owner\_\_c}

                                            onclick={ownerDetailsClick}></lightning-button>

                                        <lightning-button variant="success" label="Enquiery" class="slds-m-left\_x-small"

                                            value={prop.Id} onclick={feedbackClicked}></lightning-button>

                                        <lightning-button variant="success" label="Property Details"

                                            class="slds-m-left\_x-small" value={prop.Id} onclick={NavigateToPropDetails}>

                                        </lightning-button>

                                    </lightning-button-group>

                                </div>

                            </lightning-layout-item>

                        </lightning-layout>

                    </div>

                </lightning-layout>

            </lightning-layout-item>

        </template>

    </lightning-layout>

<template if:false={propertiesFound} >

    <div class="slds-align\_absolute-center" style="color:chartreuse">

        No Properties defined. Please stay tune for listing out the properties soon..

    </div>

</template>

<!--Owner Details Section Starts here-->

<template if:true={openOwnerDetail}>

    <div style="margin-top:5%">

        <section role="dialog" tabindex="-1" aria-modal="true" aria-describedby="modal-content-id-1" class="slds-modal slds-fade-in-open">

            <div class="slds-modal\_\_container">

                <header class="slds-modal\_\_header">

                    <lightning-card title="Property Owner Details" >

                    </lightning-card>

                </header>

                <div class="slds-modal\_\_content slds-p-around\_medium">

                    <c-my-property-owner property-owner-id={propOwnerId}></c-my-property-owner>

                </div>

            </div>

        </section>

    </div>

</template>

<!--Owner Details Section Ends here-->

<!--Enquiry Details Section Starts here-->

<template if:true={enquiryDetails}>

    <div style="margin-top:5%">

        <section role="dialog" tabindex="-1" aria-modal="true" aria-describedby="modal-content-id-1" class="slds-modal slds-fade-in-open">

            <div class="slds-modal\_\_container">

                <header class="slds-modal\_\_header">

                    <lightning-button-icon class="slds-modal\_\_close" icon-name="utility:close" onclick={closeFeedbackModal}></lightning-button-icon>

                </header>

                <div class="slds-modal\_\_content slds-p-around\_medium">

                    <c-my-property-enquiery property-id={feedbackPropertId} object-api-name="Property\_enquiry\_\_c"></c-my-property-enquiery>

                </div>

            </div>

        </section>

    </div>

</template>

<!--Enquiry Details Section Ends here-->

</template>

**myPropertyResult.js-meta.xml**

<?xml version="1.0" encoding="UTF-8"?>

<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata" fqn="MyPropertyResult">

    <apiVersion>46.0</apiVersion>

    <isExposed>true</isExposed>

     <targets>

        <target>lightning\_\_RecordPage</target>

         <target>lightning\_\_AppPage</target>

    </targets>

</LightningComponentBundle>