

**carSearch**

HTML

|  |
| --- |
| <template>  <div class="slds-m-bottom\_small">  <lightning-card title="Search Your Car">  <c-car-search-form oncartypeselect={carTypeSelectHandler}></c-car-search-form>  </lightning-card>  </div>  <div class="slds-m-vertical\_small">  <lightning-card title="Matching Cars">  <c-car-search-result car-type-id={carTypeId}></c-car-search-result>  </lightning-card>  </div>  </template> |

JS

|  |
| --- |
| import { LightningElement,track } from 'lwc';  export default class CarSearch extends LightningElement {  @track carTypeId;  carTypeSelectHandler(event)  {  this.carTypeId = event.detail;  }  } |

META

|  |
| --- |
| <?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\_\_HomePage</target>  <target>lightning\_\_AppPage</target>  <target>lightning\_\_RecordPage</target>  </targets>  </LightningComponentBundle> |

carSearchForm

HTML

|  |
| --- |
| <template>  <lightning-layout horizontal-align="center">  <lightning-layout-item padding="horizontal-small" >  <lightning-combobox name="CarTypes" label="Car Types" placeholder="ALL Types" value={selectedValue} onchange={handleCarTypeChange} variant="label-hidden" options={carTypes}>  </lightning-combobox>  </lightning-layout-item>  <lightning-layout-item padding="horizontal-small" >  <lightning-button label="New" onclick={createNewCar}>  </lightning-button>  </lightning-layout-item>  </lightning-layout>  </template> |

JS

|  |
| --- |
| import { LightningElement, track, wire } from 'lwc';  import getCarTypes from '@salesforce/apex/CarSearchFormController.getCarTypes';  import { ShowToastEvent } from 'lightning/platformShowToastEvent';  import { NavigationMixin } from 'lightning/navigation';  export default class CarSearchForm extends NavigationMixin(LightningElement) {  @track carTypes;  @wire(getCarTypes)  wiredCarType({data, error}){  if(data){  this.carTypes = [{value:'', label:'All Types'}];  data.forEach(element => {  const carType = {};  carType.label = element.Name;  carType.value = element.Id;  this.carTypes.push(carType);  });  } else if(error){  this.showToast('ERROR', error.body.message, 'error');  }  }  handleCarTypeChange(event){  const carTypeId = event.detail.value;  const carTypeSelectionChangeEvent = new CustomEvent('cartypeselect', {detail : carTypeId});  this.dispatchEvent(carTypeSelectionChangeEvent);  }  createNewCarType(){  this[NavigationMixin.Navigate]({  type:'standard\_\_objectPage',  attributes:{  objectApiName : 'Car\_Type\_\_c',  actionName : 'new'  }  });  }  showToast(title, message, variant) {  const evt = new ShowToastEvent({  title: title,  message: message,  variant: variant,  });  this.dispatchEvent(evt);  }  } |

carSearchResult

HTML

|  |
| --- |
| <template>  <lightning-layout multiple-rows horizontal-align="center">  <template if:true={carsFound}>  <template for:each={cars} for:item="car">  <lightning-layout-item key={car.Id} padding="around-small" size="12" small-device-size="6"  large-device-size="3" medium-device-size="4">  <c-car-tile car={car} oncarselect={carSelectHandler} car-selected-id={selectedCarId}></c-car-tile>  </lightning-layout-item>  </template>  </template>  <template if:false={carsFound}>  <div class="slds-align\_absolute-center">  No Cars Found  </div>  </template>  </lightning-layout>  </template> |

JS

|  |
| --- |
| import { LightningElement, api, wire, track } from 'lwc';  import getCars from '@salesforce/apex/CarSearchResultController.getCars';  import { ShowToastEvent } from 'lightning/platformShowToastEvent';  export default class CarSearchResult extends LightningElement {  @api carTypeId;  @track cars;  @track selectedCarId;  @wire(getCars, {carTypeId : '$carTypeId'})  wiredCars({data, error}){  if(data){  this.cars = data;  } else if(error){  this.showToast('ERROR', error.body.message, 'error');  }  }  showToast(title, message, variant) {  const evt = new ShowToastEvent({  title: title,  message: message,  variant: variant,  });  this.dispatchEvent(evt);  }  carSelectHandler(event){  const carId = event.detail;  this.selectedCarId = carId;  }  get carsFound(){  if(this.cars){  return true;  }  return false;  }  } |

carTile

HTML

|  |
| --- |
| <template>  <a onclick={handleCarSelect} >  <div class={isCarSelected}>  <img src={car.Picture\_\_c} class="car slds-align\_absolute-center" alt="Car Picture" >  <div>  <p class="title slds-align\_absolute-center">  {car.Contact\_\_r.Name}  </p>  </div>  </div>  </a>  </template> |

JS

|  |
| --- |
| import { LightningElement, api, wire } from 'lwc';  import { CurrentPageReference } from 'lightning/navigation';  import { fireEvent } from 'c/pubsub';  export default class CarTile extends LightningElement {  @api car;  @api carSelectedId;  @wire(CurrentPageReference) pageRef;  handleCarSelect(event){  event.preventDefault();  const carId = this.car.Id;  const carSelect = new CustomEvent('carselect', {detail:carId});  console.info('Before dispatchEvent'+this.car.Id);  this.dispatchEvent(carSelect);  console.info('Before fireEvent'+this.car.Id+'-');  fireEvent(this.pageRef, 'carselect', this.car.Id);  }  get isCarSelected(){  if(this.car.Id === this.carSelectedId){  return "tile selected";  }  return "tile";  }  } |

CSS

|  |
| --- |
| .tile {  padding: 8px;  background-color: #FFFFFF;  border-radius: 0.25rem;  }  a {  color: inherit;  }  img.car {  height: 120px;  max-width: initial;  pointer-events: none;  }  .title {  font-weight: bold;  text-transform: uppercase;  }  .selected{  border-width: medium;  border-radius: 4px;  border-color: royalblue;  border-style: solid;  } |

carDetails

HTML

|  |
| --- |
| <template>  <template if:true={carFound}>  <lightning-layout multiple-rows>  <lightning-layout-item size="12">  <lightning-tabset active-tab-value={selectedTabValue}>  <lightning-tab label="Car Details" value="cardetailstab" onactive={tabChangeHandler}>  <c-car-detail car={car} ></c-car-detail>  </lightning-tab>  <lightning-tab label="Add Car Experience" value="addexperience" onactive={tabChangeHandler}>  <c-add-car-experience car-id={carId} ></c-add-car-experience>  </lightning-tab>  <lightning-tab label="View Experiences" value="viewexperience" onactive={tabChangeHandler}>  <c-car-experiences car-id={carId} ></c-car-experiences>  </lightning-tab>  </lightning-tabset>  </lightning-layout-item>  </lightning-layout>  </template>  </template> |

JS

|  |
| --- |
| import { LightningElement, wire, track } from 'lwc';  import { getRecord } from 'lightning/uiRecordApi';  import { registerListener, unregisterAllListeners } from 'c/pubsub';  import { CurrentPageReference } from 'lightning/navigation';  import CAR\_ID from '@salesforce/schema/Car\_\_c.Id';  import CAR\_NAME from '@salesforce/schema/Car\_\_c.Name';  import CAR\_MILEAGE from '@salesforce/schema/Car\_\_c.Mileage\_\_c';  import CAR\_PER\_DAY\_RENT from '@salesforce/schema/Car\_\_c.Per\_Day\_Rent\_\_c';  import CAR\_BUILD\_YEAR from '@salesforce/schema/Car\_\_c.Build\_Year\_\_c';  import CAR\_PICTURE from '@salesforce/schema/Car\_\_c.Picture\_\_c';  import CAR\_CONTACT\_NAME from '@salesforce/schema/Car\_\_c.Contact\_\_r.Name';  import CAR\_CONTACT\_EMAIL from '@salesforce/schema/Car\_\_c.Contact\_\_r.Email';  import CAR\_CONTACT\_PHONE from '@salesforce/schema/Car\_\_c.Contact\_\_r.HomePhone';  import CAR\_CARTYPE\_NAME from '@salesforce/schema/Car\_\_c.Car\_Type\_\_r.Name';  const fields = [  CAR\_ID,  CAR\_NAME,  CAR\_PER\_DAY\_RENT,  CAR\_BUILD\_YEAR,  CAR\_PICTURE,  CAR\_CONTACT\_NAME,  CAR\_CONTACT\_EMAIL,  CAR\_CONTACT\_PHONE,  CAR\_MILEAGE,  CAR\_CARTYPE\_NAME  ]  export default class CarDetails extends LightningElement {  carId;  @track selectedTabValue;    @wire(CurrentPageReference) pageRef;  @wire(getRecord, { recordId : '$carId', fields})  car;  connectedCallback(){  registerListener('carselect', this.callBackMethod, this);  }  callBackMethod(payload){  this.carId = payload;  }  disconnectedCallback(){  unregisterAllListeners(this);  }  tabChangeHandler(event){  this.selectedTabValue = event.target.value;  }  get carFound(){  if(this.car.data){  return true;  }  return false;  }  } |

META

|  |
| --- |
| <?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\_\_AppPage</target>  <target>lightning\_\_HomePage</target>  <target>lightning\_\_RecordPage</target>  </targets>  </LightningComponentBundle> |

carDetail

HTML

|  |
| --- |
| <template>  <lightning-card title={carName} icon-name="utility:setup\_assistant\_guide">  <div slot="actions">  <lightning-button label="Full Details" onclick={fullDetails}></lightning-button>  </div>  <lightning-layout multiple-rows>  <lightning-layout-item size="12" small-device-size="6" medium-device-size="6" large-device-size="6">  <div class="slds-p-horizontal\_small">  <div class="carproperty">  <span class="label">Owner Name:</span>  <lightning-formatted-text value={ownerName}></lightning-formatted-text>  </div>  <div class="carproperty">  <span class="label">Type:</span>  <lightning-formatted-text value={type}> </lightning-formatted-text>  </div>  <div class="carproperty">  <span class="label">Build Year:</span>  <lightning-formatted-number value={buildYear}></lightning-formatted-number>  </div>  <div class="carproperty">  <span class="label">Per Day Rent:</span>  <lightning-formatted-number value={perDayRent} currency-code="USD"></lightning-formatted-number>  </div>  <div class="carproperty">  <span class="label">Mileage:</span>  <lightning-formatted-number value={mileage}></lightning-formatted-number>  </div>  </div>  </lightning-layout-item>  <lightning-layout-item size="12" small-device-size="6" medium-device-size="6" large-device-size="6">  <img src={pictureUrl} class="car slds-align\_absolute-center" alt="Car Picture" />  </lightning-layout-item>  </lightning-layout>  </lightning-card>  </template> |

JS

|  |
| --- |
| import { LightningElement, api } from 'lwc';  import { NavigationMixin } from 'lightning/navigation';  export default class CarDetail extends NavigationMixin(LightningElement) {  @api car;  fullDetails(){  this[NavigationMixin.Navigate]({  type: "standard\_\_recordPage",  attributes :{  recordId : this.car.data.fields.Id.value,  objectApiName : "Car\_\_c",  actionName : "view",  }  });  }  get carName(){  try{  return this.car.data.fields.Name.value;  }catch(error){  return 'NA';  }  }  get ownerName(){  try{  return this.car.data.fields.Contact\_\_r.value.fields.Name.value;  } catch(error){  return 'NA';  }  }  get type(){  try{  return this.car.data.fields.Car\_Type\_\_r.value.fields.Name.value;  } catch(error){  return 'NA';  }  }  get buildYear(){  try{  return this.car.data.fields.Build\_Year\_\_c.value;  } catch(error){  return 'NA';  }  }  get perDayRent(){  try{  return this.car.data.fields.Per\_Day\_Rent\_\_c.value;  } catch(error){  return 'NA';  }  }  get mileage(){  try{  return this.car.data.fields.Mileage\_\_c.value;  } catch(error){  return 'NA';  }  }  get pictureUrl(){  try{  return this.car.data.fields.Picture\_\_c.value;  } catch(error){  return 'NA';  }  }  } |

CSS

|  |
| --- |
| .label {  font-weight: bold;  display: block;  }    .carproperty {  margin-bottom: 3px;  }    .imageview {  background-repeat: no-repeat;  background-size: contain;  height: 200px;  margin: 2px;  }    .car{  border-radius: 5px;  } |

addCarExperience

HTML

|  |
| --- |
| <template>  <lightning-layout multiple-rows>  <lightning-layout-item size="12" padding="around-small">  <lightning-input label="Title" onchange={handleTitleChange}></lightning-input>  </lightning-layout-item>  <lightning-layout-item size="12" padding="around-small">  <lightning-input-rich-text label="Description" onchange={handleDescriptionChange} ></lightning-input-rich-text>  </lightning-layout-item>  <lightning-layout-item size="12" padding="around-small">  <lightning-button class="slds-m-top\_small" variant="brand" label="Add Experience" onclick={addExperience} ></lightning-button>  </lightning-layout-item>  </lightning-layout>  </template> |

JS

|  |
| --- |
| import { LightningElement, api } from 'lwc';  import { createRecord } from 'lightning/uiRecordApi';  import { ShowToastEvent } from 'lightning/platformShowToastEvent';  import NAME\_FIELD from '@salesforce/schema/Car\_Experience\_\_c.Name';  import EXPERIENCE\_FIELD from '@salesforce/schema/Car\_Experience\_\_c.Experience\_\_c';  import CAR\_FIELD from '@salesforce/schema/Car\_Experience\_\_c.Car\_\_c';  import EXPERIENCE\_OBJECT from '@salesforce/schema/Car\_Experience\_\_c'  export default class AddCarExperience extends LightningElement {  @api carId;  expTitle ='';  expDescription ='';  handleTitleChange(event){  this.expTitle = event.target.value;  }  handleDescriptionChange(event){  this.expDescription = event.target.value;  }  addExperience(){  console.info('CarId in add Car Experience ----->'+this.carId);  const fields = {};  fields[NAME\_FIELD.fieldApiName] = this.expTitle;  fields[EXPERIENCE\_FIELD.fieldApiName] = this.expDescription;  fields[CAR\_FIELD.fieldApiName] = this.carId;  const recordInput = {apiName : EXPERIENCE\_OBJECT.objectApiName, fields};  createRecord(recordInput).then( carExperience => {  this.showToast('SUCCESS', 'Experience Record Updated', 'success');  }).catch( error => {  this.showToast('ERROR', error.body.message, 'error');  });  }  showToast(title, message, variant) {  const evt = new ShowToastEvent({  title: title,  message: message,  variant: variant,  });  this.dispatchEvent(evt);  }  } |

carExperiences

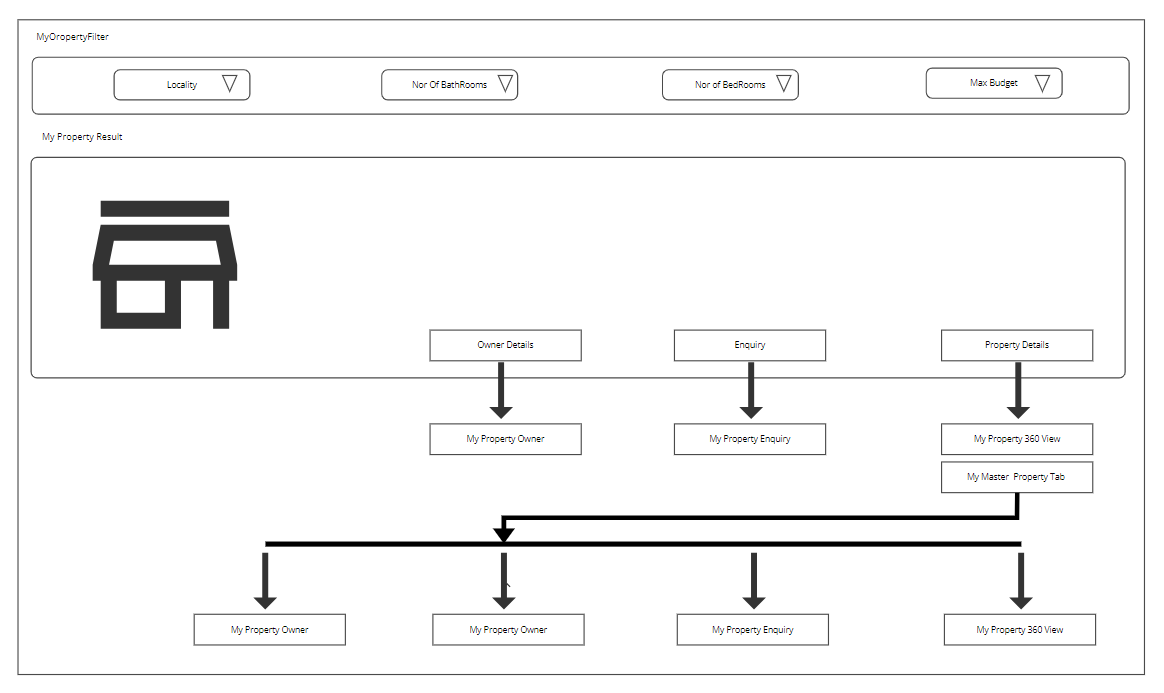
HTML

|  |
| --- |
| <template>  <template if:false={hasExperiences}>  <p>No Experience Posted For This Car</p>  </template>  <template if:true={hasExperiences}>  <div style="height: 300px; overflow: overlay;">  <template for:each={carExperiences} for:item="carExp">  <div class="slds-feed" key={carExp.Id}>  <ul class="slds-feed\_\_list">  <li class="slds-feed\_\_item">  <article class="slds-post">  <header class="slds-post\_\_header slds-media">  <div class="slds-media\_\_figure">  <a href="javascript:void(0);"  class="slds-avatar slds-avatar\_circle slds-avatar\_large">  <!-- Displaying user photo -->  <img alt={carExp.CreatedBy.Name} src={carExp.CreatedBy.SmallPhotoUrl}  title={carExp.CreatedBy.Name} />  </a>  </div>  <div class="slds-media\_\_body">  <div class="slds-grid slds-grid\_align-spread slds-has-flexi-truncate">  <p>  <!-- Displaying user name and company name -->  <a href="javascript:void(0);" data-userid={carExp.CreatedBy.Id}  onclick={userClickHandler}>{carExp.CreatedBy.Name}</a> —  {carExp.CreatedBy.CompanyName}</p>  </div>  <p class="slds-text-body\_small">  <lightning-formatted-date-time value={carExp.CreatedDate}>  </lightning-formatted-date-time>  </p>  </div>  </header>  <div class="slds-post\_\_content slds-text-longform">  <!-- Displaying the title -->  <p class="slds-text-title\_caps">{carExp.Name}</p>  <p class="slds-text-body\_small">  <!-- Displaying experince inside a formattedRichText -->  <lightning-formatted-rich-text value={carExp.Experience\_\_c}>  </lightning-formatted-rich-text>  </p>  </div>  </article>  </li>  </ul>  </div>  </template>  </div>  </template>  </template> |

JS

|  |
| --- |
| import { LightningElement, api, track } from 'lwc';  import getExperiences from '@salesforce/apex/CarExperience.getExperiences';  import { ShowToastEvent } from 'lightning/platformShowToastEvent';  import { NavigationMixin } from 'lightning/navigation';  export default class CarExperiences extends NavigationMixin(LightningElement) {  @api carId;  @track carExperiences;  connectedCallback(){  this.getCarExperiences();  }  getCarExperiences(){  console.info('CarId -->'+this.carId);  getExperiences({carId : this.carId}).then( (experiences) =>{  this.carExperiences = experiences;  }).catch((error) =>{  this.showToast('ERROR', error.body.message, 'error');  })  }  userClickHandler(event){  const userId = event.target.getAttribute('data-userid');  this[NavigationMixin.Navigate]({  type: "standard\_\_recordPage",  attributes: {  recordId: userId,  objectApiName: "User",  actionName: "view",  }  });  }    showToast(title, message, variant) {  const evt = new ShowToastEvent({  title: title,  message: message,  variant: variant,  });  this.dispatchEvent(evt);  }  get hasExperiences(){  if(this.carExperiences){  return true;  }  return false;  }    } |

Home Search Project



My Property Filter

myPropertyFilter.html

|  |
| --- |
| <template>  <lightning-layout multiple-rows style="background-color: darkturquoise;">  <lightning-layout-item padding="around-small">  <div class="header-column">  <lightning-combobox label="Locality" variant="label-stacked" placeholder="Select Locality"  field-level-help="Search Property based on Locality" value={location} options={locationOptions} onchange={handleLocationChange}>    </lightning-combobox>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <lightning-combobox label="No of BedRoom" variant="label-stacked" placeholder="Select"  field-level-help="Search Property based on BedRoom" value={noOfBedRoom} options={noOfBedRoomOptions} onchange={handleBedRoomChange}>    </lightning-combobox>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <lightning-combobox label="No of BathRoom" variant="label-stacked" placeholder="Select"  field-level-help="Search Property based on BathRoom" value={noOfBathRoom} options={noOfBathRoomOptions} onchange={handleBathRoomChange}>    </lightning-combobox>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <lightning-input type="number" value={maxBudget} label="Maximum Budget"  field-level-help="Search Property based on Budget" name="maxBudget" placeholder="Enter maximum budget" onchange={handleBudgetChange}>  </lightning-input>  </div>  </lightning-layout-item>  </lightning-layout>  </template> |

myPropertyFilter.js

|  |
| --- |
| import { LightningElement, track, wire } from 'lwc';  import { fireEvent } from 'c/pubsub';  import { CurrentPageReference } from 'lightning/navigation';  export default class MyPropertyFilter extends LightningElement {  @track location;  @track noOfBedRoom;  @track noOfBathRoom;  @track maxBudget;  @wire(CurrentPageReference) pageRef;  get locationOptions(){  return [  {label:'ALL',value:'ALL'},  { label: 'Banglore', value: 'Banglore' },  { label: 'Mumbai', value: 'Mumbai' },  { label: 'Hyderabad', value: 'Hyderabad' },  { label: 'Pune', value: 'Pune' },  { label: 'Bhubaneswar', value: 'Bhubaneswar' }  ];  }  get noOfBedRoomOptions(){  return [  {label:'ALL',value:'ALL'},  { label: '1', value: '1' },  { label: '2', value: '2' },  { label: '3', value: '3' },  { label: '4', value: '4' },  { label: '5', value: '5' }  ];  }  get noOfBathRoomOptions(){  return [  {label:'ALL',value:'ALL'},  { label: '1', value: '1' },  { label: '2', value: '2' },  { label: '3', value: '3' },  { label: '4', value: '4' },  { label: '5', value: '5' }  ];  }  handleLocationChange(event){  this.location=event.target.value;  console.log('Location selected as '+this.location);  fireEvent(this.pageRef,'handleLocChangeEvent',this.location);  console.log('After Event is fired... '+this.location);  }  handleBedRoomChange(event){  this.noOfBedRoom=event.target.value;  console.log('BedRoom selected as '+this.noOfBedRoom);  fireEvent(this.pageRef,"handleBedChangeEvent",this.noOfBedRoom);  console.log('BedRoom selected Fired '+this.noOfBedRoom);  }  handleBathRoomChange(event){  this.noOfBathRoom=event.target.value;  console.log('BathRoom selected as '+this.noOfBathRoom);  fireEvent(this.pageRef,"handleBathChangeEvent",this.noOfBathRoom);  console.log('BathRoom selected Fired '+this.noOfBathRoom);  }  handleBudgetChange(event){  this.maxBudget=event.target.value;  console.log('Budget selected as '+this.maxBudget);  fireEvent(this.pageRef,"handleBudgetChangeEvent",this.maxBudget);  console.log('Budget selected Fired '+this.maxBudget);  }    } |

myPropertyFilter.css

|  |
| --- |
| .c-container {  border: 1px solid #d8dde6;  margin: 10px 0 20px 0;  }  .header-column {  padding: 0 2rem;  color:brown;  }  .field-title {  font-size: .75rem;  line-height: 1.25;  color: #54698d;  margin-bottom: .25rem;  } |

myPropertyFilter.js-meta.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">  <apiVersion>50.0</apiVersion>  <isExposed>true</isExposed>  <targets>  <target>lightning\_\_RecordPage</target>  <target>lightning\_\_AppPage</target>  </targets>  </LightningComponentBundle> |

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

|  |
| --- |
| 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);  alert("Property OwnerId :"+this.propOwnerId);  }  feedbackClicked(event){  alert("Jai JVH ....you are in Enquiry form");  this.feedbackPropertId=event.target.value;  this.enquiryDetails=true;  }  closeFeedbackModal(event){  this.enquiryDetails=false;  }  closeOwnerModel(event){  this.openOwnerDetail=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){  this.propid=event.target.value;  console.log('In Navigation .....'+this.propid);  this[NavigationMixin.Navigate]({  type: 'standard\_\_component',  attributes: {  componentName:'c\_\_MyProperty360View'  },  state: {  c\_\_propertyId:this.propid  }  });  console.log('In Navigation .....'+this.propid);  }  } |

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> |

myOwnerDetaisl.html

|  |
| --- |
| <template>  <lightning-record-view-form record-id={propertyOwnerId} object-api-name="Property\_Owner\_\_c">  <lightning-layout multiple-rows>  <div class="slds-box slds-grid slds-wrap slds-p-around\_small">  <div class="slds-col" style="width:100%">  <lightning-output-field variant="label-hidden" style="color:brown;font-weight:bold;" field-name=Owner\_Photo\_\_c>  </lightning-output-field>  </div>  <div class="slds-col" style="margin-left:4%">  <lightning-output-field variant="label-hidden" style="color:brown;font-weight:bold;" field-name=Owner\_Name\_\_c>  </lightning-output-field>  <lightning-output-field variant="label-hidden" style="color:brown;font-weight:bold;" field-name=Email\_\_c>  </lightning-output-field>  <lightning-output-field variant="label-hidden" style="color:brown;font-weight:bold;" field-name=Owner\_Phone\_\_c>  </lightning-output-field>  <lightning-output-field variant="label-hidden" style="color:brown;font-weight:bold;" field-name=Owner\_Address\_\_c>  </lightning-output-field>  </div>  </div>  </lightning-layout>  </lightning-record-view-form>  </template> |

myOwnerDetaisl.js

|  |
| --- |
| import { LightningElement, api ,track} from 'lwc';    export default class myPropertyOwner extends LightningElement {  @api propertyOwnerId;  } |

myOwnerDetaisl.js-meta.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata" fqn="myPropertyOwner">  <apiVersion>50.0</apiVersion>  <isExposed>true</isExposed>  <targets>  <target>lightning\_\_RecordPage</target>  <target>lightning\_\_AppPage</target>  </targets>  </LightningComponentBundle> |

myPropertyEnquiry.html

|  |
| --- |
| <template>  <lightning-card title="Submit Your Enquiery" icon-name="standard:contact">  <div class="slds-p-around\_small">  <lightning-record-edit-form object-api-name="Property\_enquiry\_\_c" layout-type="compact" onsuccess={handleSuccess}>  <lightning-input-field field-name="Property\_\_c" value={propertyId} variant="label-stacked">  </lightning-input-field>  <lightning-input-field field-name="Name\_\_c"></lightning-input-field>  <lightning-input-field field-name="Email\_\_c"></lightning-input-field>  <lightning-input-field field-name="Phone\_\_c"></lightning-input-field>  <lightning-input-field field-name="Message\_\_c"></lightning-input-field>  <lightning-button type="submit" name="update" variant="brand" label="Submit Enquiery"  class="slds-m-top\_small"></lightning-button>  </lightning-record-edit-form>  </div>  </lightning-card>  </template> |

myPropertyEnquiry.js

|  |
| --- |
| import { LightningElement, api } from 'lwc';  import { ShowToastEvent } from 'lightning/platformShowToastEvent';  export default class MyPropertyEnquiery extends LightningElement {  @api objectApiName;  @api propertyId;    handleSuccess(event){  alert("Before Toast Event.....");  const evt = new ShowToastEvent({  title: "Feedback Submitted",  message: "Success",  variant: "success"  });  this.dispatchEvent(evt);  }  } |

myPropertyEnquiry.css

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">  <apiVersion>50.0</apiVersion>  <isExposed>true</isExposed>  <targets>  <target>lightning\_\_RecordPage</target>  <target>lightning\_\_AppPage</target>  </targets>  </LightningComponentBundle> |

Aura Component

myProperty360View.cmp

|  |
| --- |
| <aura:component implements="flexipage:availableForAllPageTypes,forceCommunity:availableForAllPageTypes,lightning:isUrlAddressable" access="global">  <aura:attribute name="PropertyId" type="String"></aura:attribute>  <c:myMasterPropertyTab propertyId="{!v.pageReference.state.c\_\_propertyId}"></c:myMasterPropertyTab>  </aura:component> |

myMasterPropertyTab.html

|  |
| --- |
| <template>  <lightning-tabset>  <lightning-tab label="Property Details">  <div class="slds-p-around\_medium">  <c-my-property-details property={property} property-found={propertyFound}></c-my-property-details>  </div>  </lightning-tab>  <lightning-tab label="Pricing Details">  <div class="slds-p-around\_medium">  <c-my-property-pricing property={property} property-found={propertyFound}></c-my-property-pricing>  </div>  </lightning-tab>  <lightning-tab label="Owner Details">  <div class="slds-p-around\_medium">  <c-my-property-owner property-owner-id={propertyOwnerId}></c-my-property-owner>  </div>  </lightning-tab>  <lightning-tab label="Property Video">  <div class="slds-p-around\_medium">  <c-my-property-video video={video}></c-my-property-video>  </div>  </lightning-tab>  </lightning-tabset>  </template> |

myMasterPropertyTab.js

|  |
| --- |
| import { LightningElement, api, track, wire } from 'lwc';  import getDetails from '@salesforce/apex/PropertyDetails.getDetails';  export default class MyMasterPropertyTab extends LightningElement {  @api property;  @api propertyOwnerId;  @api propertyFound;  @api propertyId;  @track error;  @track propertyOwnerId;  @track video;  @wire(getDetails, { propId: '$propertyId' })  wiredProperties({ data, error }) {  console.log('WiredProp propId :' + this.propertyId);  //console.log('WiredProp property.propId :' + this.property.propertyId);  if (data) {  console.log('Data======>' + data.Property\_Owner\_\_c.value);  this.property = data;  this.propertyFound = true;  this.propertyOwnerId = this.property.Property\_Owner\_\_c;  this.video=this.property.Property\_Video\_URL\_\_c;  //console.log('WiredProp propertyOwnerId' + this.propertyOwnerId);  console.log('WiredProp propertyOwnerId' + this.propertyOwnerId);  } else if (error) {  this.showToast('ERROR', error.body.message, 'error');  }  }  } |

myMasterPropertyTab.js-meta.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">  <apiVersion>50.0</apiVersion>  <isExposed>true</isExposed>  <targets>  <target>lightning\_\_RecordPage</target>  <target>lightning\_\_AppPage</target>  </targets>  </LightningComponentBundle> |

myPropertyDetails.html

|  |
| --- |
| <template>    <template if:true={propertyFound}>  <lightning-layout multiple-rows style="background-color:azure">  <div style="margin-top: 2%;">  <lightning-formatted-rich-text style="color:brown;font-weight:bold" value={property.Property\_Description\_\_c}></lightning-formatted-rich-text>  </div>  <div class="slds-box slds-grid slds-wrap slds-p-around\_small" style="margin-top: 2%;width:100%;">  <div class="slds-col">  <lightning-formatted-rich-text value={property.Property\_Main\_Image\_\_c}></lightning-formatted-rich-text>  </div>  <div class="slds-col">  <div class="slds-col slds-border\_bottom">  <lightning-layout multiple-rows>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="BedRooms">BedRooms</p>  <p>{property.No\_Of\_Bed\_Room\_\_c}</p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="BathRooms">BathRooms</p>  <p>{property.No\_Of\_Bath\_Room\_\_c}</p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Balconies">Balconies</p>  <p>{property.No\_of\_Balcony\_\_c}</p>  </div>  </lightning-layout-item>  </lightning-layout>  </div>  <div class="slds-border\_bottom">  <lightning-layout>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Covered Area">Covered Area</p>  <p>{property.Coverd\_area\_sqft\_\_c} Sqft</p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Carpet Area">Carpet Area</p>  <p>{property.Carpet\_area\_sqft\_\_c} Sqft.</p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Furnished Type">Furnished Type</p>  <p>{property.Furnished\_Type\_\_c} </p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Furnishing Available">Furnishing Available</p>  <p>{property.Furnishning\_Available\_\_c} </p>  </div>  </lightning-layout-item>  </lightning-layout>  </div>    <div class="slds-border\_bottom">  <lightning-layout>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Status of Electricity">Status of Electricity</p>  <p>{property.Electricity\_Status\_\_c} </p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Water Availaibility">Water Availaibility</p>  <p>{property.Water\_Availability\_\_c} </p>  </div>  </lightning-layout-item>    </lightning-layout>  </div>  <div class="slds-border\_bottom">  <lightning-layout>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Floor">Floor</p>  <p>{property.Floor\_\_c} </p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Facing">Facing</p>  <p>{property.Facing\_\_c} </p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Status">Status</p>  <p>{property.Status\_\_c} </p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Avaialble From">Avaialble From</p>  <p>{property.Available\_From\_\_c} </p>  </div>  </lightning-layout-item>    </lightning-layout>  </div>  </div>  </div>  </lightning-layout>  </template>  </template> |

myPropertyDetails.js

|  |
| --- |
| import { LightningElement,api } from 'lwc';    export default class MyPropertyDetails extends LightningElement {  @api property;  @api propertyFound;  } |

myPropertyDetails.js-meta.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">  <apiVersion>50.0</apiVersion>  <isExposed>true</isExposed>  <targets>  <target>lightning\_\_RecordPage</target>  <target>lightning\_\_AppPage</target>  </targets>  </LightningComponentBundle> |

myPropertyDetails.css

|  |
| --- |
| .c-container{  border:1px solid #d8dde6;  margin:10px 0 20px 0;  }  .header-column{  padding: 0 2rem;  }  .field-title{  font-size: .75rem;  line-height: 1.25;  color: brown;  margin-bottom: .25rem;  } |

myPropertyPricing.html

|  |
| --- |
| <template>  <template if:true={propertyFound}>  <lightning-layout multiple-rows style="background-color: azure">  <div style="margin-top: 2%">  <lightning-formatted-rich-text style="font-weight: bold;color: brown;"  value={property.Property\_Description\_\_c}></lightning-formatted-rich-text>  </div>  <div class="slds-box slds-grid slds-wrap slds-p-around-small" style="margin-top: 2%;width:100%">  <div class="slds-col">  <lightning-layout multiple-rows>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Rental Value">Rental Value</p>  <p>{property.Rent\_\_c}</p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Security Depoist">Security Depoist</p>  <p>{property.Security\_Deposit\_\_c}</p>  </div>  </lightning-layout-item>  <lightning-layout-item padding="around-small">  <div class="header-column">  <p class="field-title" title="Maintaince Charge">Maintaince Charge</p>  <p>{property.Maintainance\_Charge\_\_c}</p>  </div>  </lightning-layout-item>  </lightning-layout>  </div>  </div>  </lightning-layout>  </template>  </template> |

myPropertyPricing.js

|  |
| --- |
| import { LightningElement, api} from 'lwc';  export default class MyPropertyPricing extends LightningElement {  @api property;  @api propertyFound;  } |

myPropertyPricing.js-meta.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">  <apiVersion>50.0</apiVersion>  <isExposed>true</isExposed>  <targets>  <target>lightning\_\_RecordPage</target>  <target>lightning\_\_AppPage</target>  </targets>  </LightningComponentBundle> |

myPropertyPricing.css

|  |
| --- |
| .c-container{  border:1px solid #d8dde6;  margin:10px 0 20px 0;  }  .header-column{  padding: 0 2rem;  }  .field-title{  font-size: .75rem;  line-height: 1.25;  color: brown;  margin-bottom: .25rem;  } |