**Custom Page Template Design**

**aura.cmp:**

<aura:component implements="lightning:appHomeTemplate"

description='Custom Page Template Design'

access='global'>

<aura:attribute name="left" type="Aura.Component[]" access="global"/>

<aura:attribute name="center" type="Aura.Component[]" access="global"/>

<aura:attribute name="right" type="Aura.Component[]" access="global"/>

<div aura:id="container">

<lightning:layout>

<lightning:layoutItem size="2" aura:id="leftColumn" >

{!v.left}

</lightning:layoutItem>

<lightning:layoutItem size="7" aura:id="centerColumn" class="center">

{!v.center}

</lightning:layoutItem>

<lightning:layoutItem size="3" aura:id="rightColumn">

{!v.right}

</lightning:layoutItem>

</lightning:layout>

</div>

</aura:component>

**aura.css:**

.THIS.center {

padding:0px 20px;

}

**aura.design**:

<design:component label="Custom Page Layout 2 7 3">

<flexipage:template>

<flexipage:region name="left" defaultWidth="SMALL"></flexipage:region>

<flexipage:region name="center" defaultWidth="LARGE"></flexipage:region>

<flexipage:region name="right" defaultWidth="SMALL"></flexipage:region>

</flexipage:template>

</design:component>

**Car Controller**

**Apex:**

public with sharing class carController {

public class Filters {

@AuraEnabled

public String searchKey {get;set;}

@AuraEnabled

public Decimal maxPrice {get;set;}

@AuraEnabled

public String[] category {get;set;}

@AuraEnabled

public String[] makeType {get;set;}

}

@AuraEnabled(Cacheable=true)

public static List<Car\_\_c> getCars(Filters filters){

String key, whereClause ='';

Decimal maxPrice;

String[] category, makeType, criteria = new List<String>{};

if(filters !=null){

maxPrice = filters.maxPrice;

category = filters.category;

makeType = filters.makeType;

if(!String.isEmpty(filters.searchKey)){

key = '%'+filters.searchKey + '%';

criteria.add('Name LIKE: key');

}

if(filters.maxPrice >= 0){

criteria.add('MSRP\_\_c <= :maxPrice');

}

if(filters.category != null){

criteria.add('Category\_\_c IN :category');

}

if(filters.makeType != null){

criteria.add('Make\_\_c IN :makeType');

}

if(criteria.size()>0){

whereClause = 'WHERE ' + String.join(criteria, ' AND ');

}

}

String query = 'SELECT Id, Name, MSRP\_\_c,Make\_\_c, Description\_\_c,Category\_\_c, Picture\_URL\_\_c FROM Car\_\_c ' + whereClause + ' WITH SECURITY\_ENFORCED' + ' ORDER BY Name';

return Database.query(query);

}

@AuraEnabled(cacheable=true)

public static List<Car\_\_c> getCar(Id carId,String[] makeType){

return [SELECT Id, Name, MSRP\_\_c,Make\_\_c, Description\_\_c,Category\_\_c, Picture\_URL\_\_c FROM Car\_\_c WHERE Make\_\_c IN :makeType AND Id != :carId];

}

}

**Car Filter**

**HTML:**

<template>

<lightning-card title="Car Filter" icon-name="standard:calibration">

<div class="slds-p-horizontal\_medium">

<lightning-input label="Search key" type="search" value={filter.searchKey} onchange={handleChange}></lightning-input>

<section class="slider-section">

<lightning-slider label="Max Price" step="100" min="0" max="999999" value={filter.maxPrice} onchange={handleSlider}></lightning-slider>

</section>

<section>

<h2>Category</h2>

<template if:true={category.data}>

<template for:each={category.data.values} for:item="c">

<lightning-input key={c.value} label={c.label} checked="" type="checkbox" data-name="category" data-value={c.value} onchange={handleCheckbox}></lightning-input>

</template>

</template>

<template if:true={category.error}>

{categoryError}

</template>

</section>

<section>

<h2>Make</h2>

<template if:true={makeType.data}>

<template for:each={makeType.data.values} for:item="m">

<lightning-input key={m.value} label={m.label} data-value={m.value} checked type="checkbox" data-name="makeType" onchange={handleCheckbox}></lightning-input>

</template>

</template>

<template if:true={makeType.error}>

{makeError}

</template>

</section>

</div>

</lightning-card>

</template>

**JS:**

import { LightningElement,wire } from 'lwc';

import { getObjectInfo,getPicklistValues } from 'lightning/uiObjectInfoApi';

import CAR\_OBJECT from '@salesforce/schema/Car\_\_c'

import CATOGERY from '@salesforce/schema/Car\_\_c.Category\_\_c'

import MAKE from '@salesforce/schema/Car\_\_c.Make\_\_c'

import SAMPLEC from '@salesforce/messageChannel/carFiltered\_\_c'

import { publish,MessageContext } from 'lightning/messageService';

const CATEGORY\_ERROR='Error loading at Category'

const MAKE\_ERROR='Error loading at Make Type'

export default class CarFilter extends LightningElement {

categoryError=CATEGORY\_ERROR

makeError=MAKE\_ERROR

filter={

searchKey:'',

maxPrice:999999

}

@wire(MessageContext)

msgContext

@wire(getObjectInfo,{objectApiName:CAR\_OBJECT})

objectInfo

@wire(getPicklistValues,{recordTypeId:'$objectInfo.data.defaultRecordTypeId',fieldApiName:CATOGERY})

category

@wire(getPicklistValues,{recordTypeId:'$objectInfo.data.defaultRecordTypeId',fieldApiName:MAKE})

makeType

handleChange(event)

{

this.filter={...this.filter,"searchKey":event.target.value}

this.filterDataPasstoList()

}

handleSlider(event)

{

this.filter={...this.filter,"maxPrice":event.target.value}

console.log(this.filter.maxPrice)

console.log(event.target.value)

this.filterDataPasstoList()

}

handleCheckbox(event)

{

const {name,value}=event.target.dataset

console.log("name",name)

console.log("value",value)

console.log("Checkbox",event.target.checked)

if(!this.filter.category){

const category=this.category.data.values.map(item=>item.value)

const makeType=this.makeType.data.values.map(item=>item.value)

this.filter={...this.filter,category,makeType}

console.log(this.filter)

}

if(event.target.checked)

{

if(!this.filter[name].includes(value))

{

this.filter[name]=[...this.filter[name],value]

}

}

else{

this.filter[name]=this.filter[name].filter(item=>item !==value)

console.log(this.filter)

}

this.filterDataPasstoList()

}

filterDataPasstoList()

{

publish(this.msgContext,SAMPLEC,{lmcfield:this.filter})

}

}

**Meta.xml:**

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

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

<apiVersion>52.0</apiVersion>

<isExposed>true</isExposed>

<masterLabel>Car Filter</masterLabel>

<targets>

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

</targets>

</LightningComponentBundle>

**CarTile**

**HTML:**

<template>

<div onclick={handleclick}>

<div class="content" >

<img src={cars.Picture\_URL\_\_c} class="slds-align\_absolute-center carImg" alt="Car Picture"/>

<div>

<p class="title slds-align\_absolute-center">{cars.Name}</p>

<p class="currency slds-align\_absolute-center">

MSRP:

<lightning-formatted-number format-style="currency" currency-code="USD" class="price" maximum-fraction-digits="0" value={cars.MSRP\_\_c}></lightning-formatted-number>

</p>

</div>

</div>

</div>

</template>

**JS:**

import { LightningElement,api} from 'lwc';

export default class CarTile extends LightningElement {

@api cars={}

handleclick()

{

this.dispatchEvent(new CustomEvent('selected',{

detail:this.cars.Id

}))

}

}

**CSS:**

.carImg{

height:120px;

max-width:initial;

pointer-events: none;

}

.title{

font-weight: bold;

text-transform: uppercase;

}

.content{

padding:8px;

background-color: white;

border-radius: 0.25rem;

}

**meta.xml:**

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

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

<apiVersion>52.0</apiVersion>

<isExposed>false</isExposed>

</LightningComponentBundle>

**CarTileList**

**HTML:**

<template>

<lightning-card title="Cars Available" icon-name="standard:account">

<div class="slds-p-around\_medium wrapper">

<template if:true={cars.length}>

<div class="content">

<template for:each={cars} for:item="car">

<c-car-tile cars={car} key={car.Id} onselected={handleslectedCar}></c-car-tile>

</template>

</div>

</template>

<template if:false={cars.length}>

<c-placeholder message="There is no car matching your select current car"></c-placeholder>

</template>

</div>

</lightning-card>

</template>

**JS:**

import { LightningElement, wire } from 'lwc';

import getCars from '@salesforce/apex/carController.getCars';

import SAMPLEC from '@salesforce/messageChannel/carFiltered\_\_c'

import CAR\_FILTERED from '@salesforce/messageChannel/selectedCar\_\_c'

import { publish,subscribe,MessageContext,APPLICATION\_SCOPE } from 'lightning/messageService';

export default class CarTileList extends LightningElement {

cars=[]

errors

filters={}

subscription

@wire(MessageContext)

msgContext

@wire(getCars,{filters:'$filters'})

carsList({data,error})

{

if(data)

{

this.cars=data

console.log(data)

}

if(error)

{

this.errors=error

console.error(error)

}

}

connectedCallback()

{

this.subscribeMethod()

}

subscribeMethod()

{

this.subscription= subscribe(this.msgContext,SAMPLEC,(message)=>{this.handleMessage(message)},{scope:APPLICATION\_SCOPE})

}

handleMessage(message)

{

console.log(message.lmcfield)

this.filters={...message.lmcfield}

console.log('this.filters',this.filters)

}

handleslectedCar(event)

{

console.log(event.detail)

publish(this.msgContext,CAR\_FILTERED,{carid:event.detail})

}

}

**CSS:**

c-car-tile{

min-width:200px;

flex:1;

}

.content{

display:flex;

flex-wrap: wrap;

}

.wrapper{

background-color: lightgray;

}

**meta.xml:**

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

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

<apiVersion>52.0</apiVersion>

<isExposed>true</isExposed>

<masterLabel>Car Tile List</masterLabel>

<targets>

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

</targets>

</LightningComponentBundle>

**LightningMessageChannel**

**carFiltered.messageChannel-meta.xml:**

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

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

<isExposed>true</isExposed>

<masterLabel>carFilterd</masterLabel>

<description>Passing data from LMC</description>

<lightningMessageFields>

<fieldName>lmcfield</fieldName>

<description>Passing data using LMC field</description>

</lightningMessageFields>

</LightningComponentBundle>

**Car Card**

**HTML:**

<template>

<lightning-card title={carName} icon-name="standard:apex\_plugin">

<template if:true={recordId}>

<lightning-button-icon icon-name="utility:expand\_alt" slot="actions" onclick={handleIcon}></lightning-button-icon>

<div class="slds-m-horizantal\_medium">

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

<img src={carImage} alt="Car Picture"/>

<lightning-record-view-form onload={handleLoad} object-api-name={objectApiName} record-id={recordId} density="compact">

<lightning-output-field field-name={category}></lightning-output-field>

<lightning-output-field field-name={make}></lightning-output-field>

<lightning-output-field field-name={msrp}></lightning-output-field>

<section>Other Features</section>

<lightning-output-field field-name={fueltype}></lightning-output-field>

<lightning-output-field field-name={control}></lightning-output-field>

<lightning-output-field field-name={seats}></lightning-output-field>

</lightning-record-view-form>

</div>

</div>

</template>

<template if:false={recordId}>

<c-placeholder message="Select car to see details"></c-placeholder>

</template>

</lightning-card>

</template>

**JS:**

import { LightningElement, wire } from 'lwc';

import CAR\_OBJECT from '@salesforce/schema/Car\_\_c'

import NAME\_FIELD from '@salesforce/schema/Car\_\_c.Name'

import IMAGE\_FIELD from '@salesforce/schema/Car\_\_c.Picture\_URL\_\_c'

import CATOGERY\_FIELD from '@salesforce/schema/Car\_\_c.Category\_\_c'

import MAKE\_FIELD from '@salesforce/schema/Car\_\_c.Make\_\_c'

import MSRP\_FIELD from '@salesforce/schema/Car\_\_c.MSRP\_\_c'

import FUEAL\_FIELD from '@salesforce/schema/Car\_\_c.Fuel\_Type\_\_c'

import CONTROL\_FIELD from '@salesforce/schema/Car\_\_c.Control\_\_c'

import SEATS\_FIELD from '@salesforce/schema/Car\_\_c.Seats\_\_c'

import { getFieldValue } from 'lightning/uiRecordApi';

import CAR\_FILTERED from '@salesforce/messageChannel/selectedCar\_\_c'

import {subscribe, MessageContext ,APPLICATION\_SCOPE} from 'lightning/messageService';

import { NavigationMixin } from "lightning/navigation"

export default class CarCard extends NavigationMixin(LightningElement) {

objectApiName=CAR\_OBJECT

category=CATOGERY\_FIELD

make=MAKE\_FIELD

msrp=MSRP\_FIELD

fueltype=FUEAL\_FIELD

control=CONTROL\_FIELD

seats=SEATS\_FIELD

recordId

carImage

carName

@wire(MessageContext)

msgContext

handleLoad(event)

{

const {records}=event.detail

const recordData=records[this.recordId]

this.carImage=getFieldValue(recordData,IMAGE\_FIELD)

this.carName=getFieldValue(recordData,NAME\_FIELD)

}

connectedCallback()

{

this.subscribeMethod()

}

subscribeMethod()

{

this.subscription= subscribe(this.msgContext,CAR\_FILTERED,(message)=>{this.handleMessage(message)},{scope:APPLICATION\_SCOPE})

}

handleMessage(message)

{

console.log('message',message)

this.recordId=message.carid

}

handleIcon()

{

this[NavigationMixin.Navigate]({

type:'standard\_\_recordPage',

attributes:

{

// recordId:{recordId}, or

recordId:this.recordId,

objectApiName:'Account',

actionName:'view'

}

})

}

}

**CSS:**

section{

font-weight: bold;

font-size: 0.85rem;

margin-top:0.75rem;

padding: 0.25rem;

text-transform: uppercase;

background:lightgray;

border-radius:5px;

}

**meta.xml:**

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

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

<apiVersion>52.0</apiVersion>

<isExposed>true</isExposed>

<masterLabel>Car Card</masterLabel>

<targets>

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

</targets>

</LightningComponentBundle>

**LightningMessageChannel**

**selectedCar.messageChannel-meta.xml:**

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

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

<isExposed>true</isExposed>

<masterLabel>selectedCar</masterLabel>

<description>This LMS called when car selected</description>

<lightningMessageFields>

<fieldName>carid</fieldName>

<description>Pass CardId</description>

</lightningMessageFields>

</LightningComponentBundle>

**Place Holder**

**HTML:**

<template>

<div class="slds-m-around\_large">

<img src={carImg}/>

<div class="slds-m-around\_large">{message}</div>

</div>

</template>

**JS:**

import { LightningElement,api } from 'lwc';

import CAR\_HUB from '@salesforce/resourceUrl/placeholder'

export default class Placeholder extends LightningElement {

@api message

carImg=CAR\_HUB

}

**CSS:**

div{

text-align: center;

color:orange;

}

img{

height: 70px;;

}

**Fetch Similar Car**

**HTML:**

<template>

<lightning-card title="Fetch Similar Cars">

<div class="slds-m-around\_medium">

<div class="slds-text-align\_center">

<template if:false={similarCars.length}>

<lightning-button label="Fetch Similar Cars" variant="brand" onclick={handleSimilarCars}></lightning-button>

</template>

</div>

<template if:true={similarCars.length}>

<template for:each={similarCars} for:item="car">

<lightning-layout vertical-align="center" key={car.Id}>

<lightning-layout-item >

<img src={car.Picture\_URL\_\_c} alt="Car Picture" style="height: 80px; max-width: initial;"/>

</lightning-layout-item>

<lightning-layout-item flexibility="grow" class="slds-p-horizontal\_small">

<p>{car.Name}</p>

<lightning-formatted-number format-style="currency" currency-code="USD" maximum-fraction-digits="0" value={car.MSRP\_\_c}></lightning-formatted-number>

</lightning-layout-item>

<lightning-button label="View Details" value="brand" data-id={car.Id} onclick={handleViewDetails}></lightning-button>

</lightning-layout>

</template>

</template>

</div>

</lightning-card>

</template>

**JS:**

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

import getCar from '@salesforce/apex/carController.getCar';

import { getRecord } from 'lightning/uiRecordApi';

import MAKE\_TYPE from '@salesforce/schema/Car\_\_c.Make\_\_c'

import { NavigationMixin } from "lightning/navigation"

export default class FetchSimilarCar extends NavigationMixin(LightningElement) {

@api objectApiName

@api recordId

similarCars=[]

@wire(getRecord,{recordId:'$recordId',fields:[MAKE\_TYPE]})

car

handleSimilarCars()

{

getCar({carId:this.recordId,makeType:this.car.data.fields.Make\_\_c.value}).then(result=>{

this.similarCars=result

console.log(result)

}).catch(error=>{

console.error(error)

})

}

handleViewDetails(event)

{

this[NavigationMixin.Navigate]({

type:'standard\_\_recordPage',

attributes:

{

// recordId:{recordId}, or

recordId:event.target.dataset.id,

objectApiName:this.objectApiName,

actionName:'view'

}

})

}

}

**meta.xml:**

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

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

<apiVersion>52.0</apiVersion>

<isExposed>true</isExposed>

<targets>

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

</targets>

</LightningComponentBundle>