

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

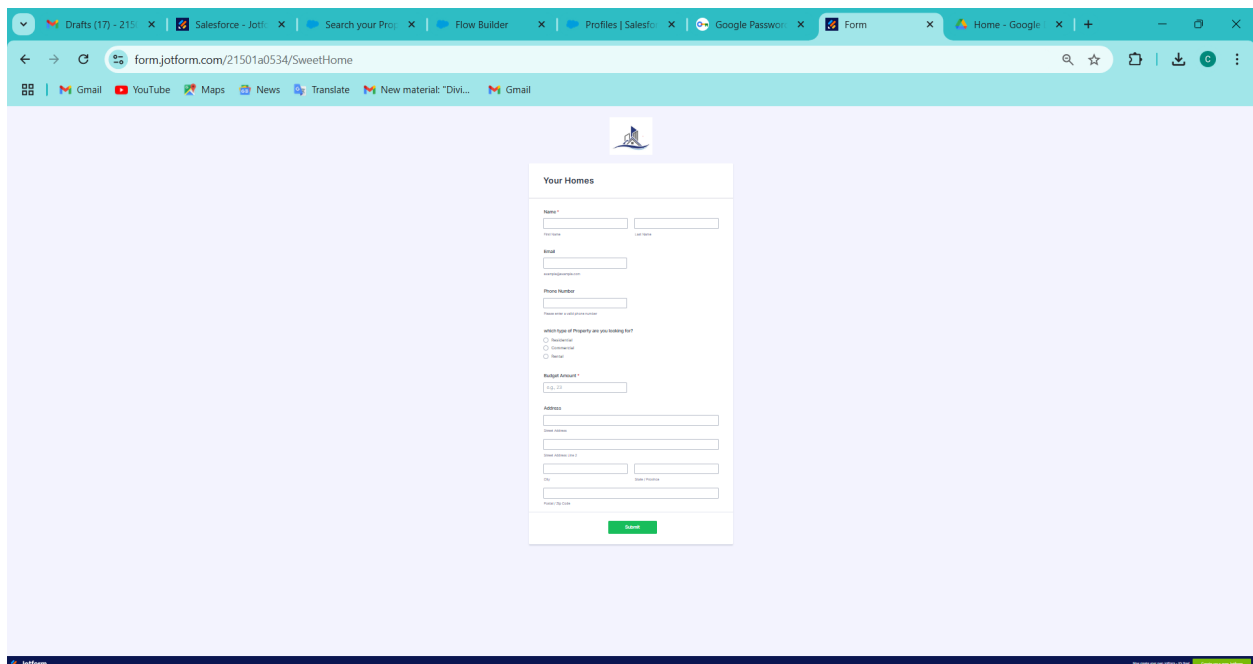
theorg to create a record of customers automatically.

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/21501a0534/SweetHome>



The screenshot shows a web browser window with multiple tabs. The active tab is titled 'form.jotform.com/21501a0534/SweetHome'. The browser's address bar shows the URL. Below the browser window, a Jotform titled 'Your Homes' is displayed. The form contains the following fields and options:

- Name ***: A text input field with a 'Last Name' label.
- Email**: A text input field with a placeholder 'example@gmail.com'.
- Phone Number**: A text input field with a placeholder '1234567890'.
- What type of Property are you looking for?**: A group of three radio button options: 'Residential', 'Commercial', and 'Retail'.
- Budget Amount ***: A text input field with a placeholder '\$10,000'.
- Address**: A text input field with a placeholder '123 Main St'.
- City**: A text input field with a placeholder 'New York'.
- State**: A text input field with a placeholder 'NY'.
- Zip**: A text input field with a placeholder '10001'.
- Save**: A green button at the bottom of the form.

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.	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.	Maharashtra	Rental	25000	kamdli	kathora	Amravati	444805	checked

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

SETUP > OBJECT MANAGER
Customer

Details

Fields & Relationships
15 Items, Sorted by Field Label

Q, Quick Find

New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Budget Amount	Budget_Amount__c	Number(18, 0)		
City	City__c	Text(255)		
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Text(255)		
Customer	Name	Text(80)		✓
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User, Group)		✓
Phone Number	Phone_Number__c	Number(18, 0)		

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 pk	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 fields as follows

The screenshot shows the Salesforce Setup interface, specifically the Object Manager for the 'Property' object. The 'Fields & Relationships' tab is selected, displaying a list of 8 fields. The fields are sorted by Field Label. The table below represents the data shown in the screenshot.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Location	Location__c	Text(255)		
Owner	OwnerId	Lookup(User,Group)		✓
Property	Name	Text(80)		✓
Property Name	Property_Name__c	Text(255)		
Type	Type__c	Text(255)		
Verified	Verified__c	Text(255)		

Integrate Jotform with Salesforce Platform

In this Milestone we are going to integrate jotform 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”.

The screenshot shows the Jotform 'Form Builder' interface with the 'SETTINGS' tab selected. On the left sidebar, 'INTEGRATIONS' is highlighted. The main area is titled 'SALESFORCE' with the subtitle 'Send new leads, contacts, or accounts to your sales CRM'. Below this, there's a 'Select a Salesforce Object' dropdown set to 'Customer'. Under 'Create a record', there's a table mapping 'Object Fields' to 'Form' fields. The mappings are: Customer__c to Name - First Name, City to Address - City, Budget Amount to Budget Amount, Phone Number to Phone Number, Property Type to which type of Property are you looking..., Street Address to Address - Street Address, Name to Name - Last Name, State to Address - Street Address 2, and Street Address line 2 to Address - Street Address 2. At the bottom, there's an 'Update an existing record' section with a toggle switch set to 'OFF'.

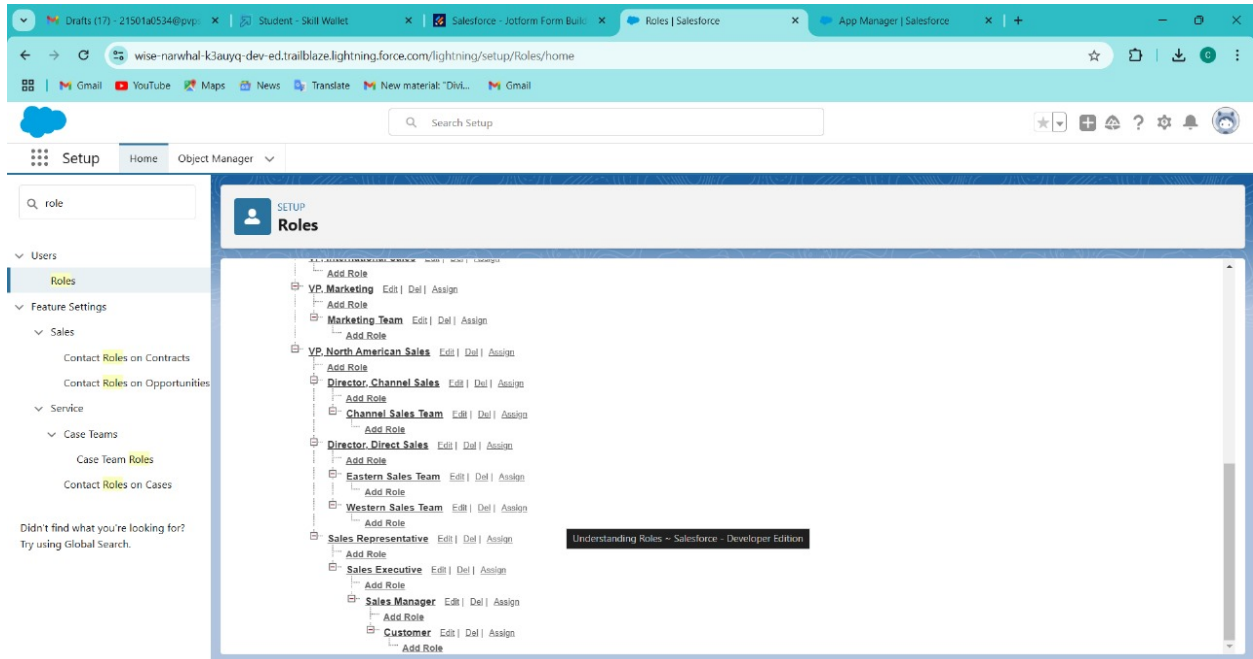
Then “Save the Integration” and “Finish”.

The screenshot shows the 'All Actions' page for the Salesforce integration. It features the Salesforce logo and the text 'Send new leads, contacts, or accounts to your sales CRM'. There are two buttons: 'See Action Logs' and '+ Add New Action'. Below these, there's a list of actions. The first action is numbered '1' and is titled 'Create or update a record' with the object 'Customer'.

Create Roles

here we need to Create Roles as per business requirement

Activity:- 1



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

Activity 1

1. From Setup >> Go to App Manager 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.

New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details


* App Name ⓘ

* Developer Name ⓘ


Description ⓘ

App Branding

Image ⓘ



Primary Color Hex Value ⓘ



Org Theme Options

☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview

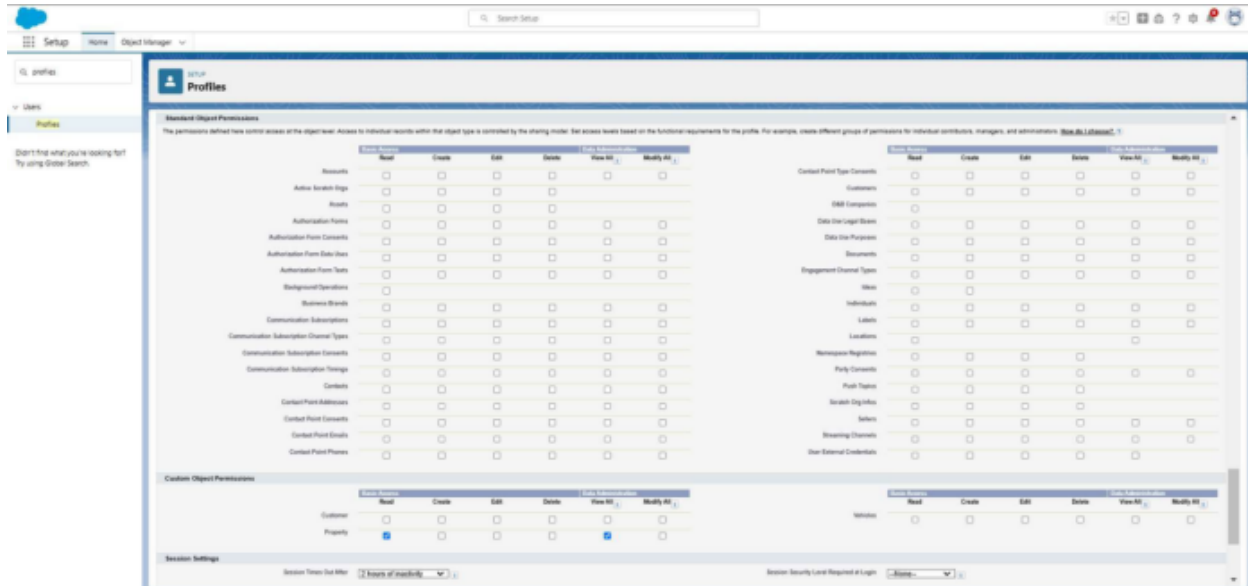
Create Profiles

Create profiles as per business requirement

Creating Customer Profiles

1. From Setup? Go to Profiles and Clone (standard platform)Salesforce Platform User and Name it "Customer"..
2. Uncheck allthe Custom Objects and Check onlyProperty Details From Custom App Settings.
3. so RemovealltheStandard Object Permissions

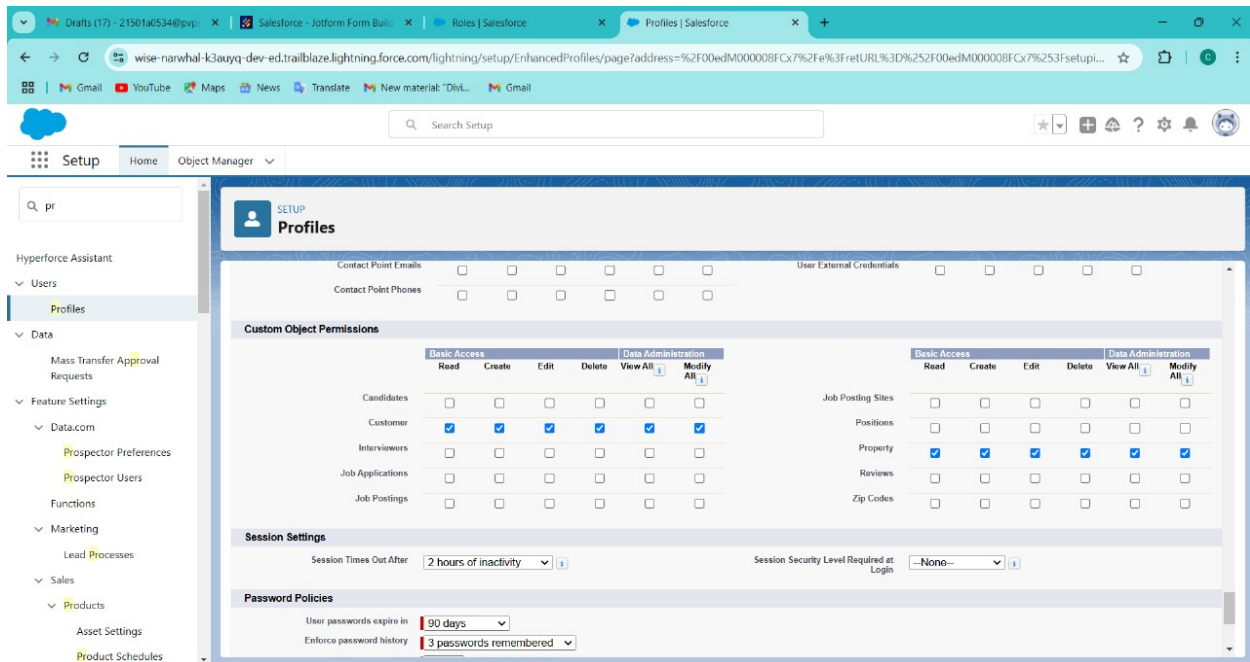
4. Uncheck all the Custom Object Permissions and check read and view all in “Property”
5. make sure every submissionobject permissions are unselected and then save



Creating ManagerProfiles :-

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".

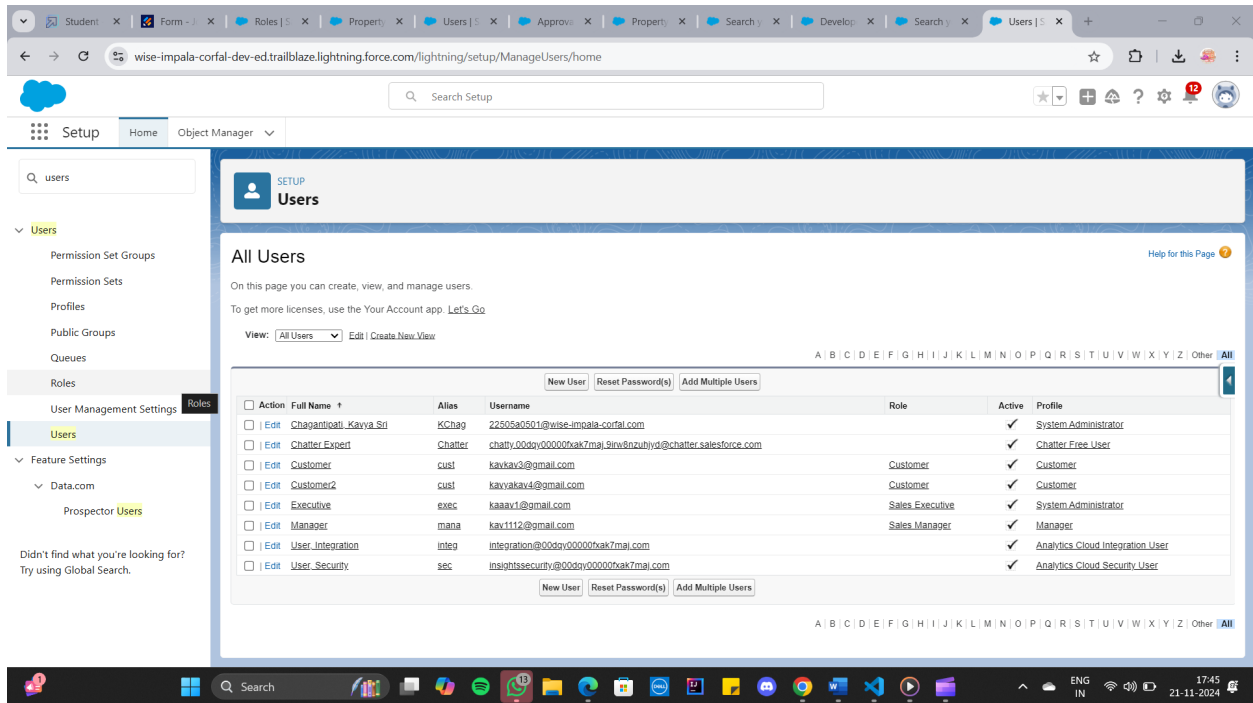


Create a CheckBox field on user

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"



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

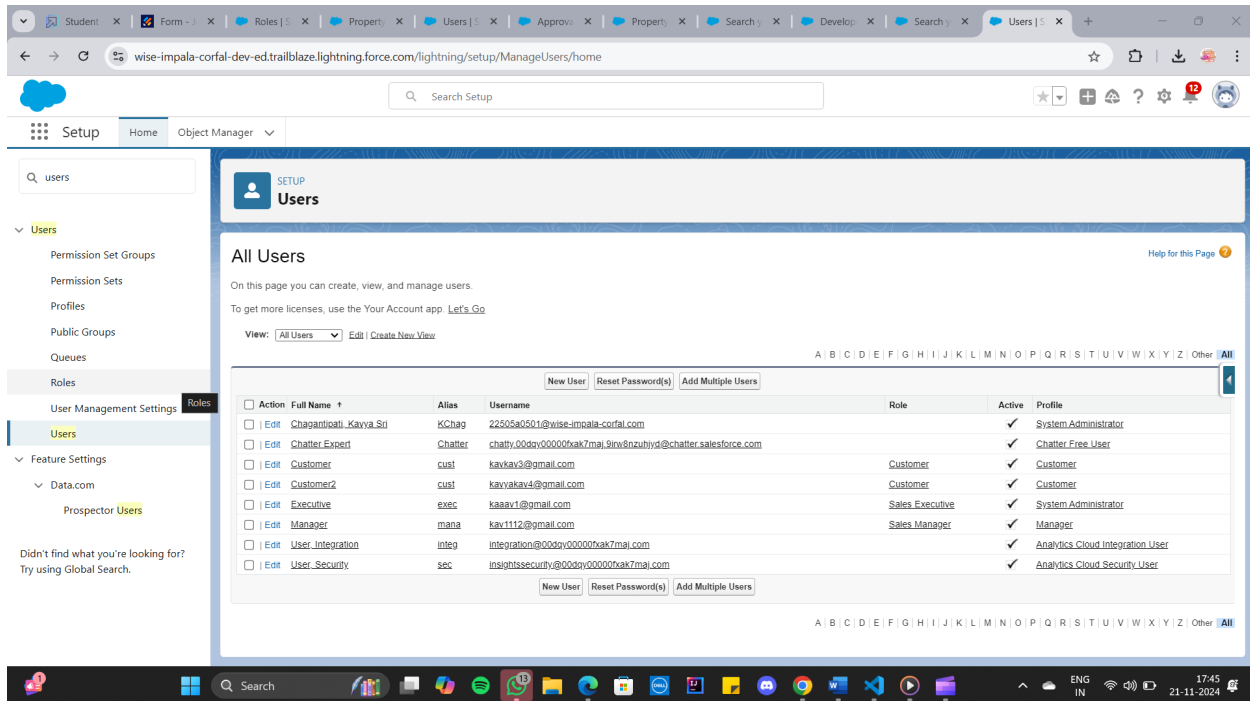
2. Last Name >> Manager
3. Role >> Sales Manager
4. License >> Salesforce Platform
5. Profile >> Manager
6. Save

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



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

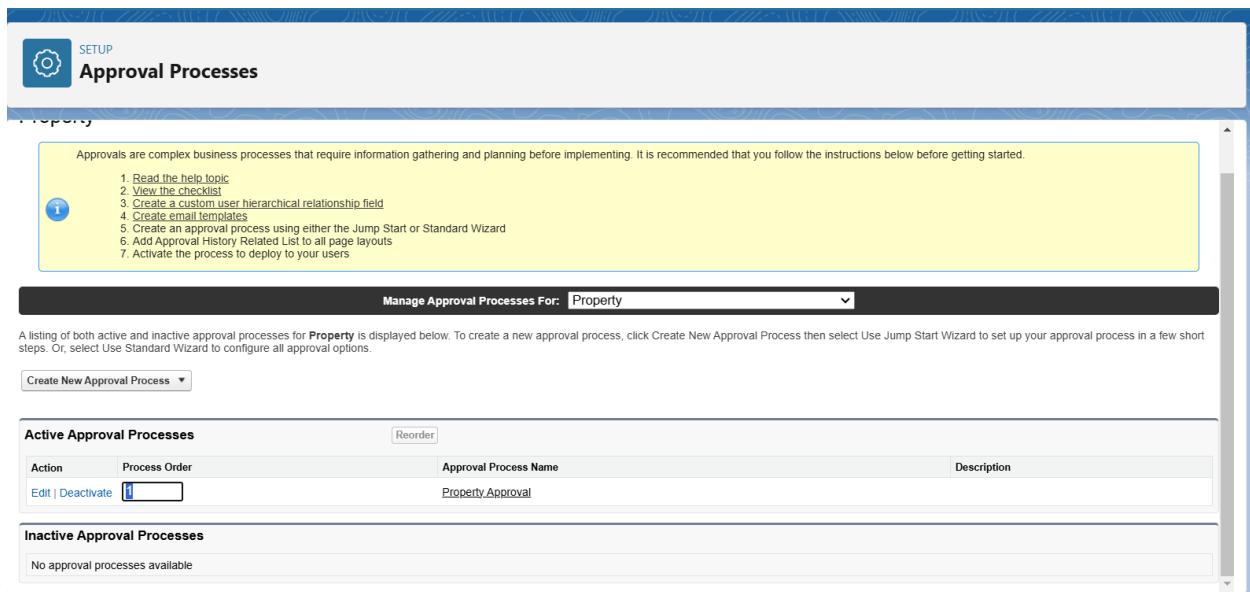
Update as "Verified Property"

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.



Create a Recordtrigger ñow to submit the Approval Process Automatically

A flow that can submit the recordsdirectly for approval

Activity1

- FromSetup >>Search forFlows >>Click OnNew andSelect "Record Trigger Flow".
- SelectObject >>Property
- Select"Trigger the flowwhen" >> "A record is created"
- SetEntry Conditions>> "None"
- Add a"Action" >> "Submit for Approval"
- Give Label >> Approval forproperty
- Record Id>> (!SRecord.Id)
- Done

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

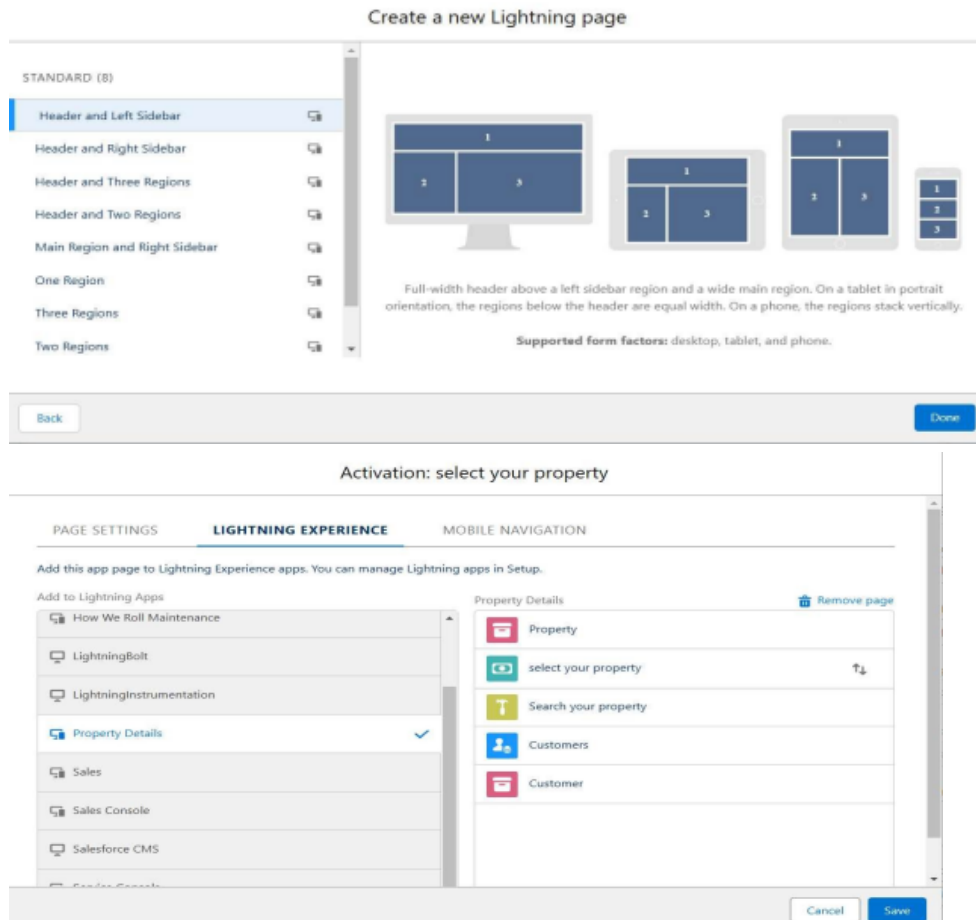
The screenshot displays the Salesforce Flow Builder interface. The main canvas shows a flow diagram starting with a 'Start' event (Record-Triggered Flow) for the 'Property' object, triggered by 'A record is created'. The flow then proceeds through a 'Run Immediately' step, followed by an 'Approval for property' action, and ends with an 'End' step. The right sidebar shows the 'Configure Start' configuration for the 'Property' object, with the trigger set to 'A record is created'.

Create an App Page

Create an App Page on the Property details Object named as “Search Your Property”

Activity1

1. From Setup »Go to Lightning App Builder >> Click on New >> Select App Page and
2. Click on Next.
3. Give Label as “Search your Property” click “Next”.
4. Click “header and Left Sidebar” and Click on “Done”
5. Click on “Save ”and then click on “Activate”.
6. From Page Settings select page activation as “Activate for all Users”.
7. From Lightning Experience Click on “Property Details” and click on Add Page”.
8. Then Click on “Save”



Create a LWC Component

- a. Create an LWC Component for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on "Search your Property Page"

Activity1

1. Create an ApexClass and make it aura enabled and name it "PropertHandler_LWC"

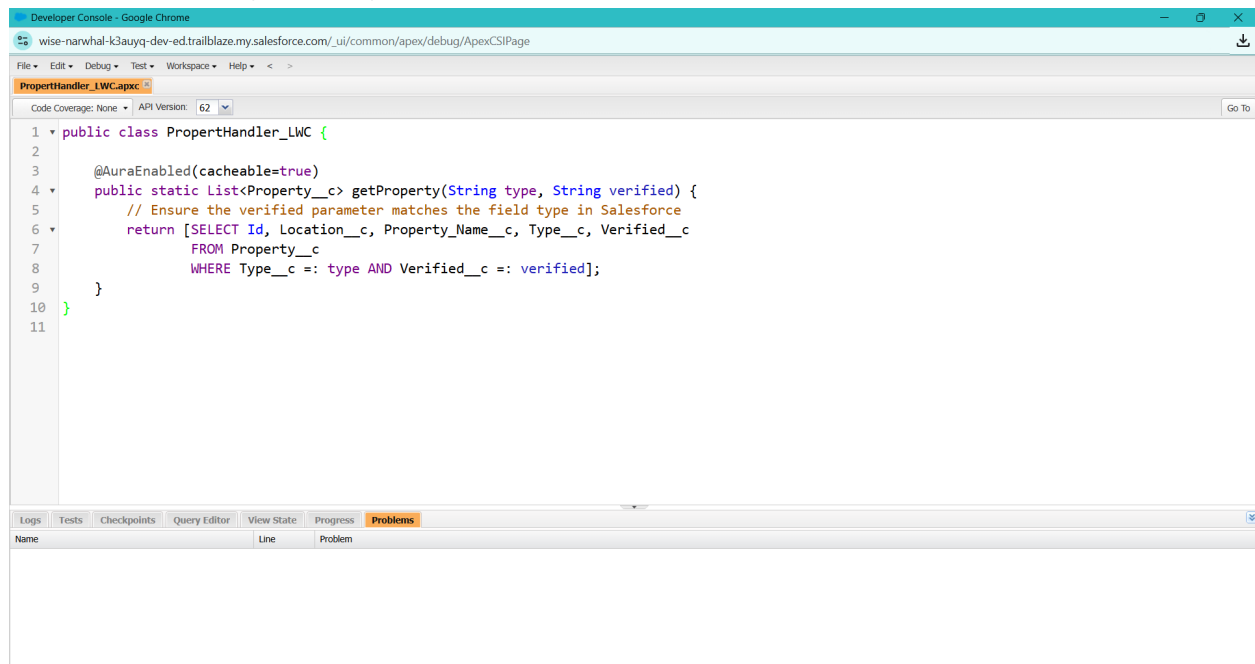
Code: -

```
public class PropertHandler_LWC (
    @AuraEnabled(cacheable=true)

    public static List<Property> getProperty(String type, Boolean verified) (
        String verifiedstr = verified ? 'true' : 'false' // Convert boolean to string return
        [SELECT Id, Location c, Property_Name c, Type c, Verified c
```


FROM Property c

WHERE Type c = :type AND Verified c = :verifiedStr];



```
1 public class PropertyHandler_LWC {
2
3     @AuraEnabled(cacheable=true)
4     public static List<Property__c> getProperty(String type, String verified) {
5         // Ensure the verified parameter matches the field type in Salesforce
6         return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c
7                 FROM Property__c
8                 WHERE Type__c =: type AND Verified__c =: verified];
9     }
10 }
11
```

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 :-

```
<template>
```

```
<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={propetyoptions} onchange={changehandler}></lightning-combobox>
    </div>

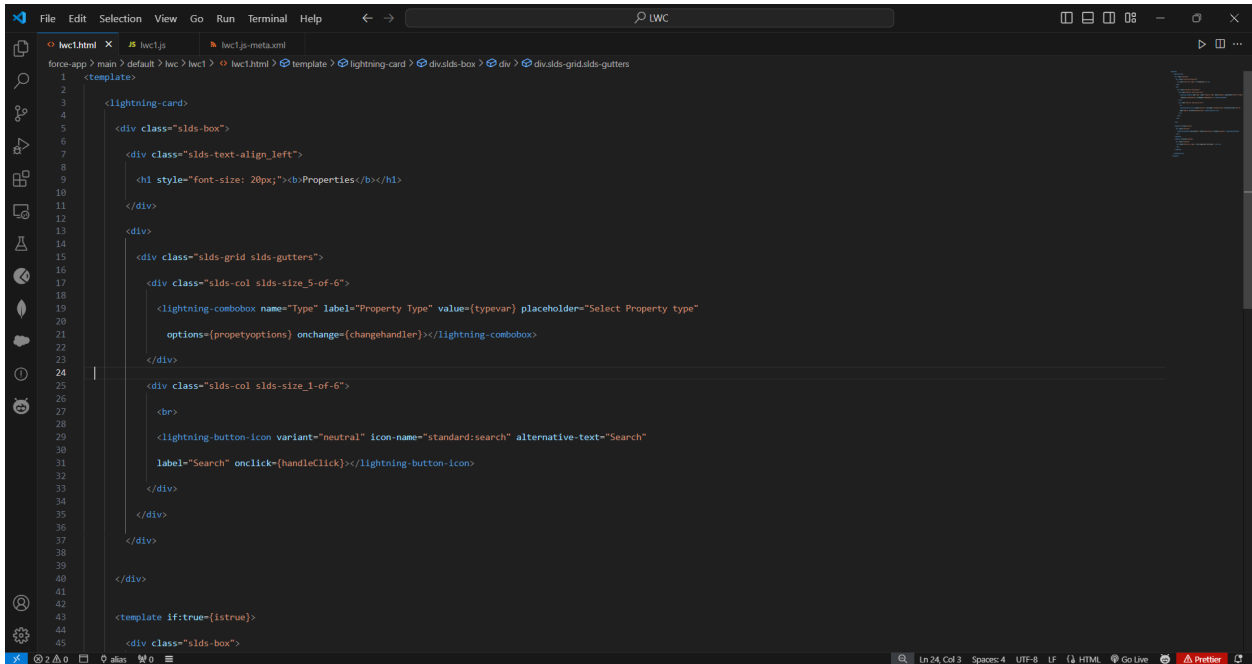
    <div class="slds-col slds-size_1-of-6">
        <br>

        <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}>
    <div class="slds-box">
        <lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-
datatable>
    </div>
</templates>
<template if:false={isfalse}>
    <div class="slds-box">
        <div style="font-size: 15px;"><b>No properties Are Found !!</b></div>
    </div>
</template>
</lightning-card>
</templates>

```



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 anagementextends LightningElement ( @api
recordId
userId = USER_ID;
verifiedvar typevar
isfalse = true; istrue
= false;
@track propertylist = [];
columns = [
( label: 'Property Name', fieldName: 'Property_Name c'), ( label:
'Property Type', fieldName: 'Type c'),
( label: 'Property Location', fieldName: 'Locationc'), ( label:
"Property link", fieldName: "Property link c"}
propetyoptions= [
( label: "Commercial", value: "Commercial" }, ( label:
"Residential", value: "Residential" ),
```

```
( label: "rental", value: "rental" }
```

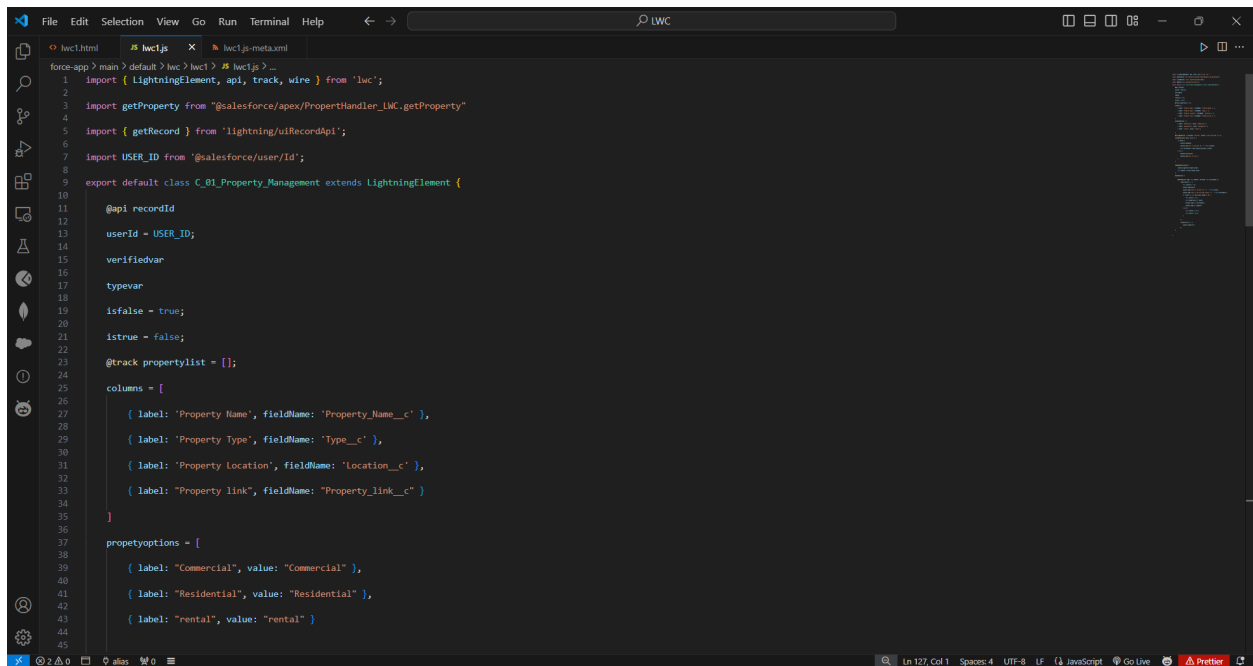
```
@wire(getRecord, ( recordId: "$userId", fields: ['User.Verified c'] )) recordFunction((  
data, error )) (
```

```
    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')
```

```
changeHandler(event) (  
    console.log(event.target.value); this.typevar =  
    event.target.value;
```

```
handleClick() {  
    getProperty(( type: this.typevar, verified: this.verifiedvar ))  
        .then((result) => ( this.isfalse  
            = true; console.log(result)  
            console.log("This is the User id --> ' + this.userId);  
            console.log("This is the verified values --> ' + this.verifiedvar); if  
            (result != null && result.length != 0) (  
                this.istrue = true; this.propertylist =  
                result; console.log(this.verifiedvar);  
                console.log(this.typevar)  
            ) else (  
                this.isfalse = false;  
                this.istrue = false;  
            .catch((error) => (  
                console.log(error)
```



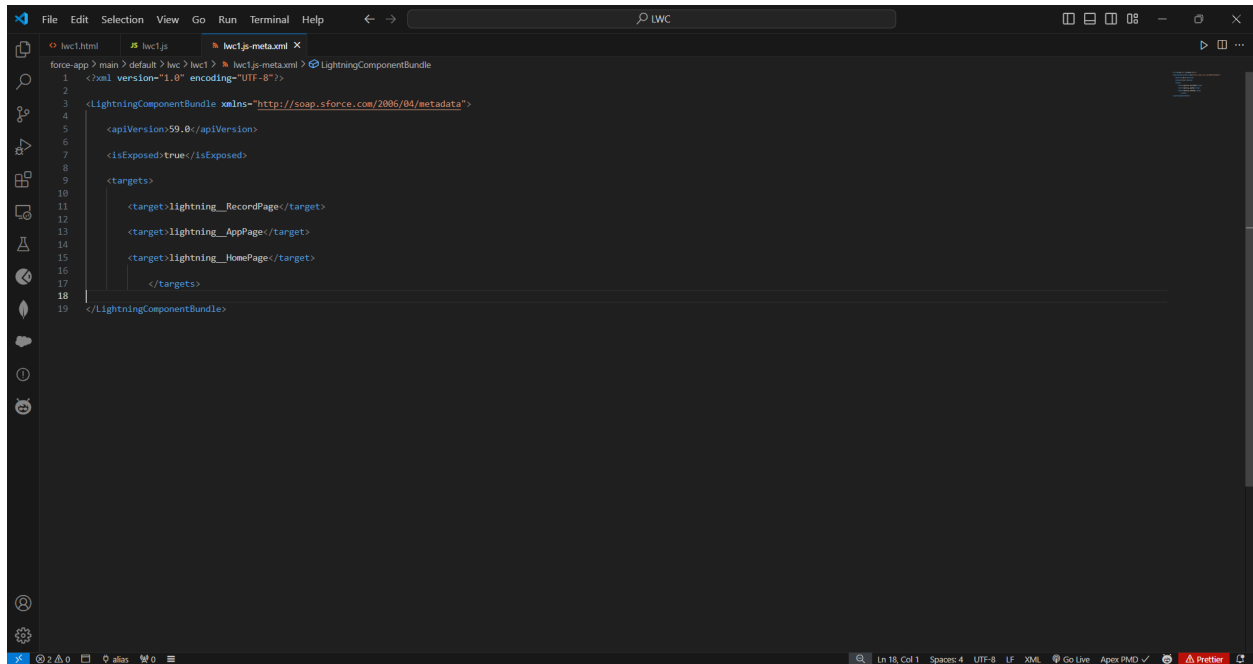
1. In Yourmetafile give your targets to deploy the component.

Code

```

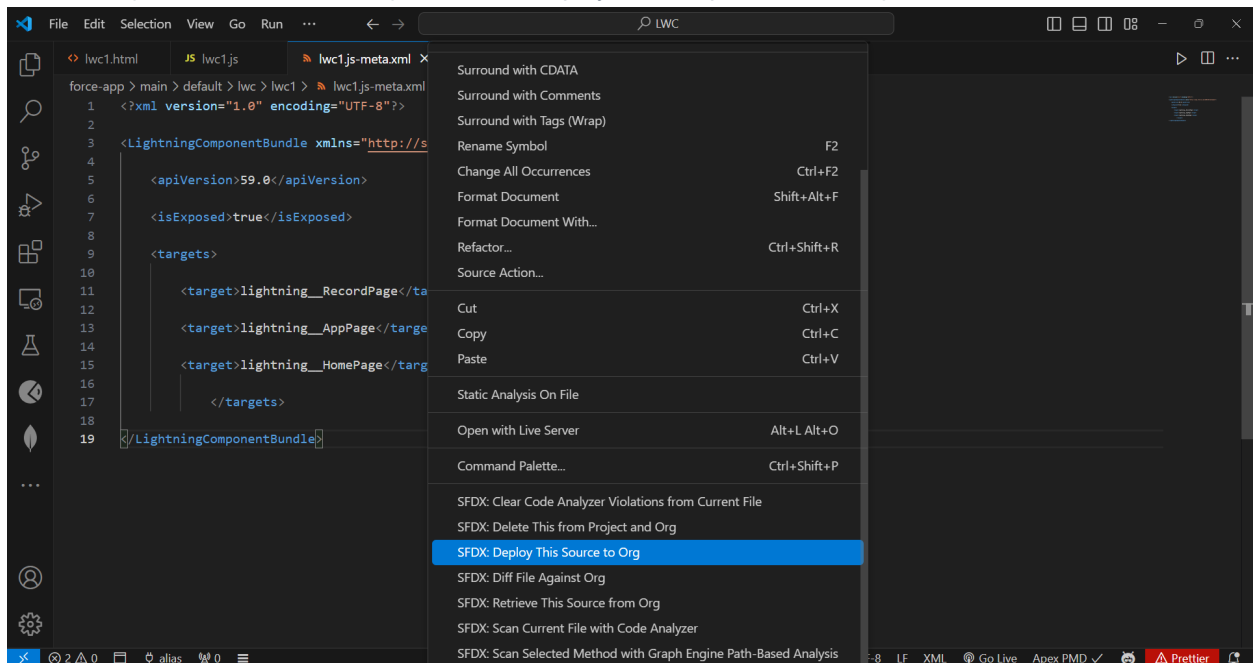
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>59.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordPage</target>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage</target>
  </targets>
</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
13     <target>lightning__AppPage</target>
14
15     <target>lightning__HomePage</target>
16
17   </targets>
18
19 </LightningComponentBundle>
```

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



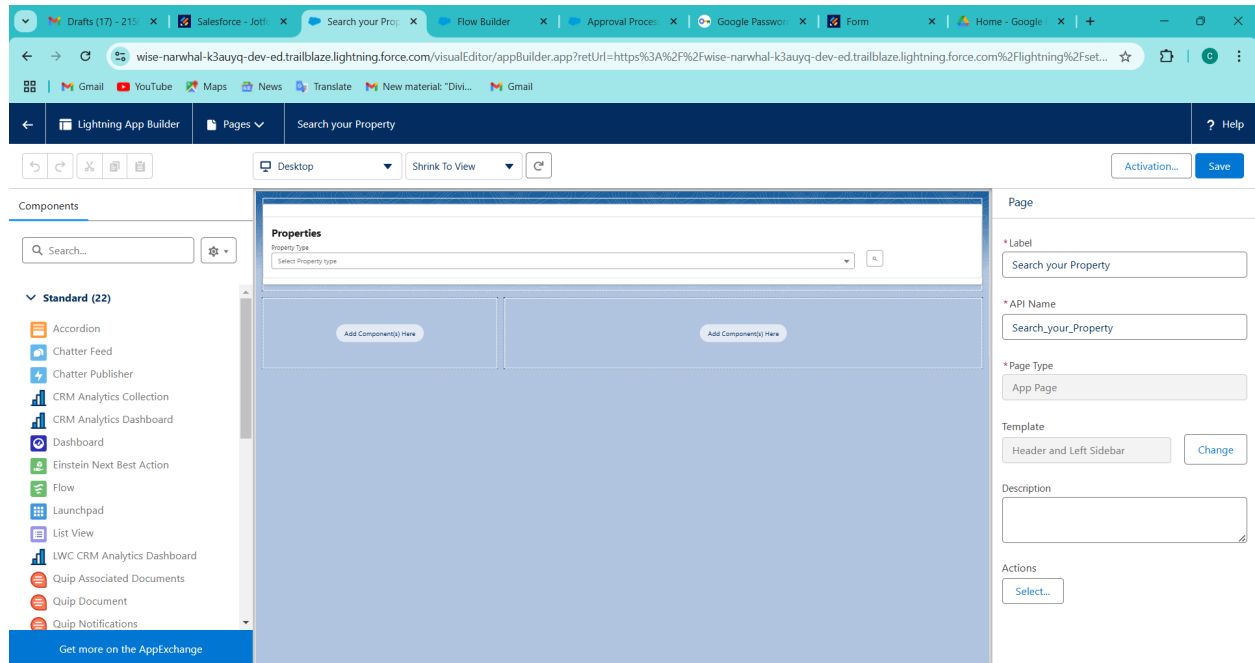
Drag this Component to your App Page

Adding the Component to your Page

Activity1

1. From Setup >> Go to App Launcher >> Search for PropertyDetails

2. On thisPage click on gear icon and click on EditPage
3. after clicking on edit page it will be redirected to app pages 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".



Search Setup



Setup

Home

Object Manager

apex class

Custom Code

Apex Classes

Didn't find what you're looking for?
Try using Global Search.



SETUP

Profiles

Enable Profile Access for Apex Class

ProperHandler_LWC

Save

Cancel

Available Profiles

Analytics Cloud Integration User
Analytics Cloud Security User
Authenticated Website
B2B Reordering Portal Buyer Profile
Contract Manager
Cross Org Data Proxy User
Custom: Marketing Profile
Custom: Sales Profile
Custom: Support Profile
Customer Community Login User
Customer Community Plus Login User
Customer Community Plus User
Customer Community User
Customer Portal Manager Custom

Add

Remove

Enabled Profiles

Customer
Manager
System Administrator