

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

1. Add, update, and delete client details.
2. Track client preferences, budget, and location interests.
3. Maintain contact details and communication history.

2. Property Management

1. Manage property listings with details like type, price, location, and features.
2. Track properties available for sale, rent, or lease.
3. Upload photos and documents for properties.

3. Requirement Matching

1. Match client requirements with available properties using filters.
2. Notify clients about new properties that fit their criteria.

4. Lead Tracking

1. Manage inquiries and follow up with potential clients.
2. Schedule meetings and site visits.
3. Assign leads to specific team members.

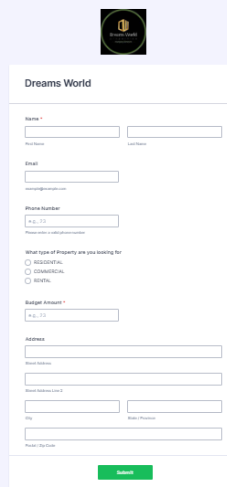
Milestone 1: Create a Jotform and integrate it with the org to create a record of customers automatically.

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/243242292940050>



The screenshot shows a Jotform titled "Dreams World" with the following fields:

- Name ***: First Name and Last Name input fields.
- Email**: Email address input field.
- Phone Number**: Phone number input field with a country code dropdown (+91, US).
- What type of Property are you looking for**: Radio button options for RESIDENTIAL, COMMERCIAL, and RENTAL.
- Budget Amount ***: Budget input field with a country code dropdown (+91, US).
- Address**: Address input field.
- City**: City input field.
- State/Province**: State/Province input field.
- Country**: Country input field.
- Submit**: Green submit button.

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	Emial	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.

The screenshot shows the Salesforce Setup interface. At the top, there's a search bar and navigation tabs for 'Setup', 'Home', and 'Object Manager'. The 'Object Manager' tab is selected, and the 'Customer' object is chosen. The left sidebar lists various setup options like 'Details', 'Fields & Relationships', 'Page Layouts', etc. The 'Fields & Relationships' section is active, displaying a table of 15 items. The table has columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The fields listed include 'Budget Amount', 'City', 'Created By', 'Customer', 'Name', 'Emial', 'Last Modified By', 'Owner', and 'Phone Number'.

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)		✓
Emial	Emial__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	Jubilee hill Hyd	Checked

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

SETUP > OBJECT MANAGER
Property

Details

Fields & Relationships
8 Items, Sorted by Field Label

Quick Find

New Deleted Fields Field Dependencies Set History Tracking

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 'SALESFORCE' integration selected. The 'Create a record' section is expanded, showing a list of 'Object Fields' on the left and 'Form' fields on the right. The 'Object Fields' list includes Customer, City, Budget Amount, Phone Number, Property Type, Street Address, Name, State, and Street Address line 2. The 'Form' list includes Name - First Name, Address - City, Budget Amount, Phone Number, which type of Property are you looking..., Address - Street Address, Name - Last Name, Address - Street Address 2, and Address - Street Address 2. The 'Update an existing record' toggle is currently OFF.

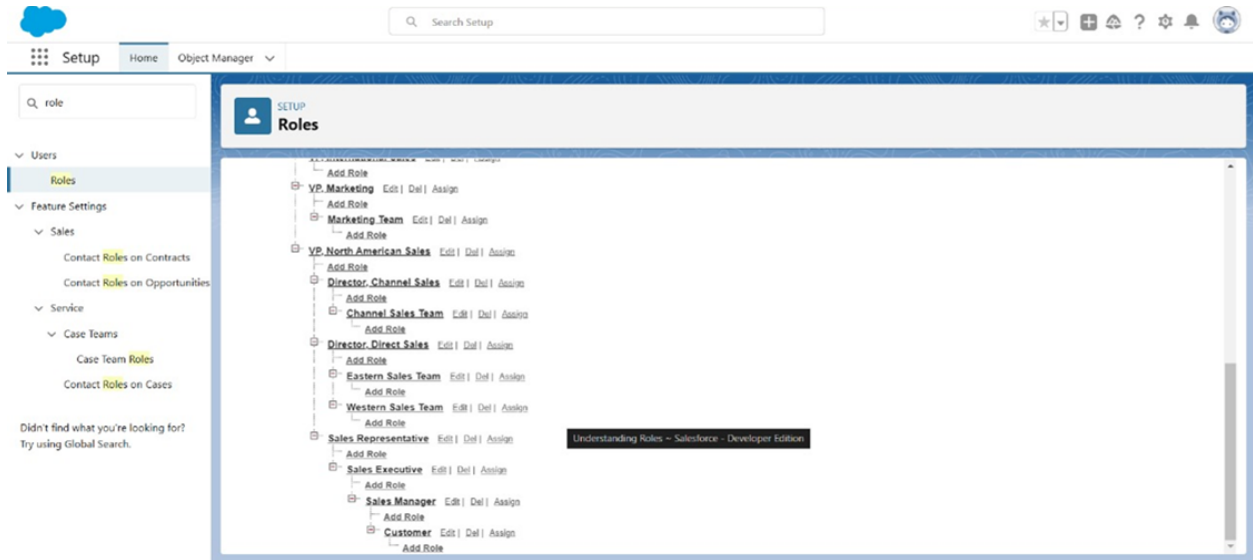
Then “Save the Integration” and “Finish”.

The screenshot shows the Salesforce integration setup screen in Jotform. It features the Salesforce logo and the text "Send new leads, contacts, or accounts to your sales CRM". Below this, there is a section titled "All Actions" with a "See Action Logs" link and an "Add New Action" button. A single action is listed: "1 Create or update a record 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

Activity1

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

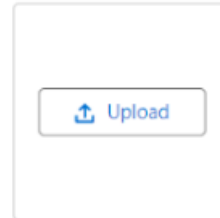
* App Name ⓘ

* Developer Name ⓘ

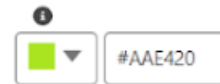
Description ⓘ

App Branding

Image ⓘ



Primary Color Hex Value ⓘ



Org Theme Options

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

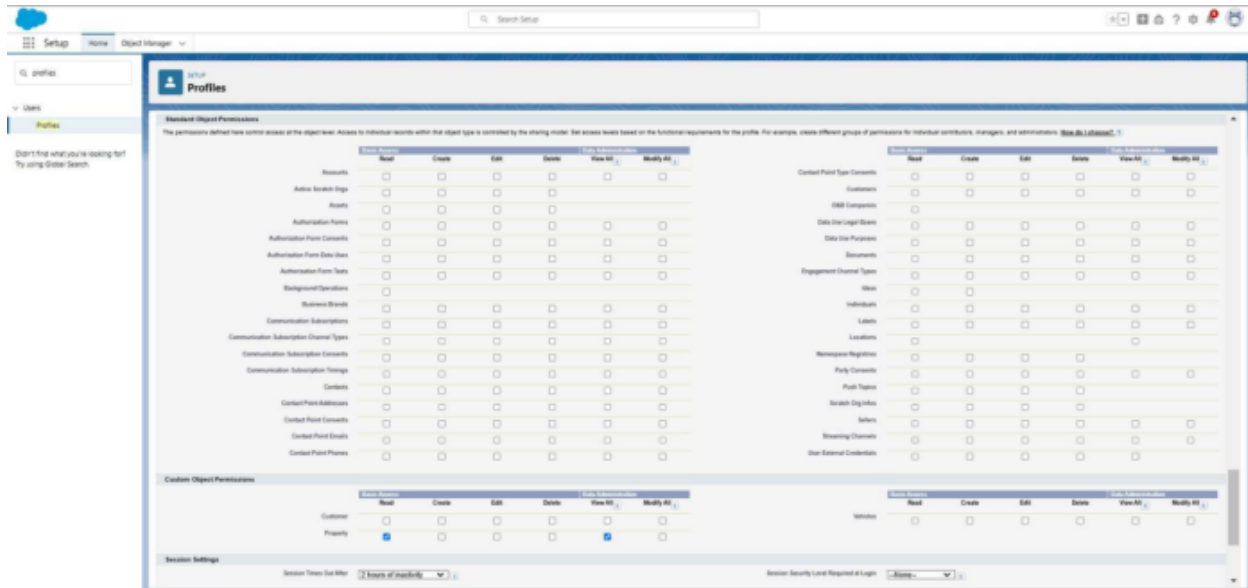
App Launcher Preview

Create Profiles

Create profiles as per business requirement

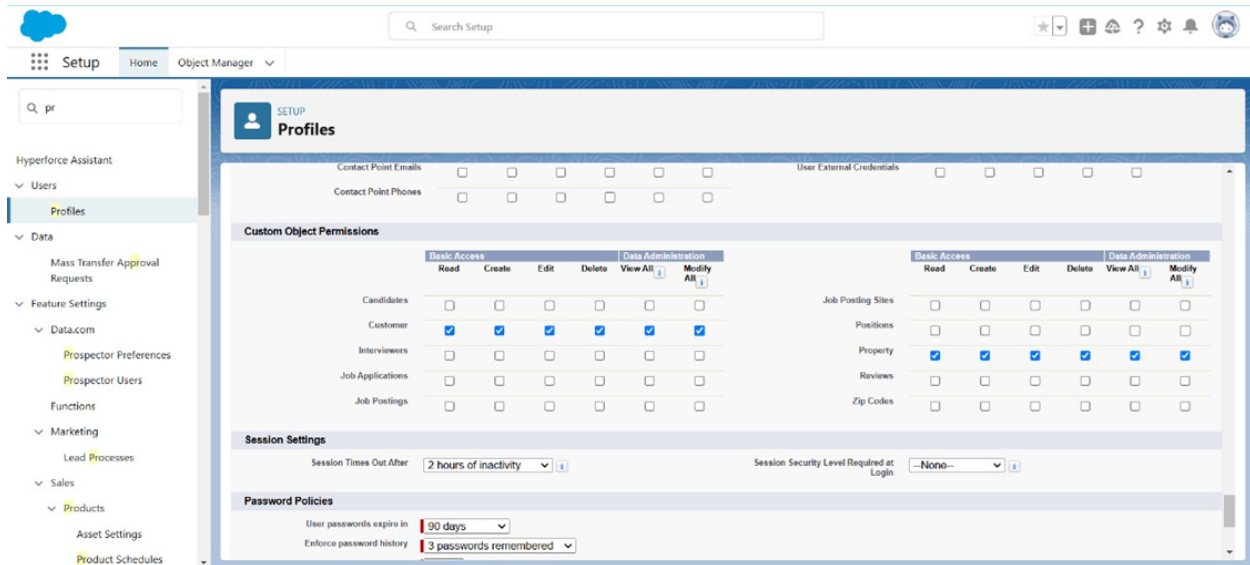
Creating Customer Profiles

1. From Setup? Go to Profiles and Clone (standard platform)Salesforce Platform User and Name it "Customer"..
2. Uncheck all the Custom Objects and Check only Property Details From Custom App Settings.
3. so Remove all the Standard Object Permissions
4. Uncheck all the Custom Object Permissions and check read and view all in "Property"
5. make sure every submission object permissions are unselected and then save



Creating Manager Profiles :-

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

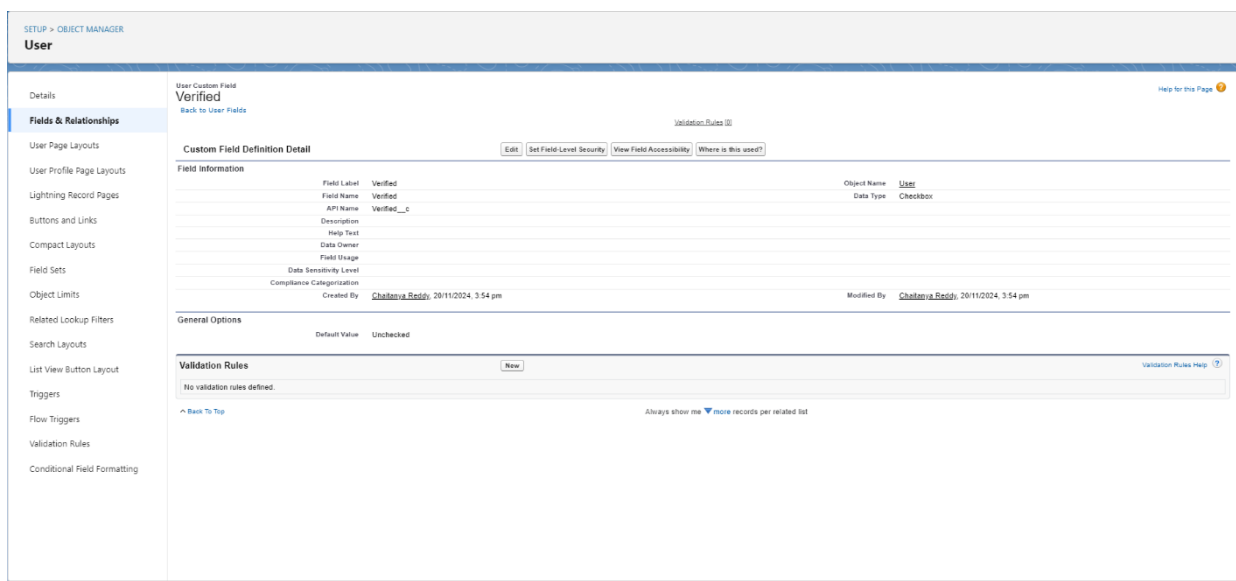


Create a Check Box field on user

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. Go to 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. Go to 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 verified check box is "Unchecked"

User : 4

1. Go to 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

SETUP
Users

All Users

On this page you can create, view, and manage users.

To get more licenses, use the Your Account app. [Let's Go](#)

View: All Users | [Edit](#) | [Create New User](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#) Other [\[All\]](#)

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Award_Youmes	yawad	yawad.80e0bf9baed5c0e1eb9zou.cady7Zpppfk.bafm-dodcy@comcanv.com		✓	Cross Qrs Data Proxy User
<input type="checkbox"/> Edit	Chatter Expert	chatter	chatt.0d6d0000a8@buuf.i0gedrsvdy@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	Customer	cust1	chalanyo403@gmail.com	Customer	✓	Customer
<input type="checkbox"/> Edit	Customer2	cust2	redy0304@gmail.com	Customer	✓	Customer
<input type="checkbox"/> Edit	Executive	esbs	redy03042004@gmail.com	Sales Executive	✓	System Administrator
<input type="checkbox"/> Edit	Gutten_duv99k	aout	aout.rbd8ey7ck.6rkrmuscvg.icdecedms.vrhumiakf@comcanv.com		✓	Cross Qrs Data Proxy User
<input type="checkbox"/> Edit	Manager	mana	chalanyo305@gmail.com	Sales Manager	✓	Manager
<input type="checkbox"/> Edit	Reddy_Challanya	CRedd	salsaleschalny@comcanv.com		✓	System Administrator
<input type="checkbox"/> Edit	User_Integration	inte0	integration00d6d0000a8@buuf.com		✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User_Security	sec	insghasecurty@0d6d0000a8@buuf.com		✓	Analytics Cloud Security User

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#) Other [\[All\]](#)

Create an Approval Process for Property Object

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 AdministratorsoRthe currently assigned approver can edit records during the approval process.
9. FromStep 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 field Update 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.


Approval Processes

Property

Approvals are complex business processes that require information gathering and planning before implementing. It is recommended that you follow the instructions below before getting started.

1. [Read the help topic](#)
2. [View the checklist](#)
3. [Create a custom user hierarchical relationship field](#)
4. [Create email templates](#)
5. Create an approval process using either the Jump Start or Standard Wizard
6. Add Approval History Related List to all page layouts
7. Activate the process to deploy to your users

Manage Approval Processes For: **Property**

A listing of both active and inactive approval processes for **Property** is displayed below. To create a new approval process, click Create New Approval Process then select Use Jump Start Wizard to set up your approval process in a few short steps. Or, select Use Standard Wizard to configure all approval options.

Create New Approval Process

Active Approval Processes

Reorder

Action	Process Order	Approval Process Name	Description
Edit Deactivate	1	Property Approval	

Inactive Approval Processes

No approval processes available

Create a Record trigger ñow to submit the Approval Process Automatically

A flow that can submit the records directly for approval

Activity1

1. From Setup >> Search for Flows >> Click On New and Select "Record Trigger Flow".
2. Select Object >> Property
3. Select "Trigger the flow when" >> "A record is created"
4. Set Entry Conditions >> "None"
5. Add a "Action" >> "Submit for Approval"
6. Give Label >> Approval for property
7. Record Id >> (!SRecord.Id)
8. Done

Save the Flow and Give label as "Property Approval" and "Activate"

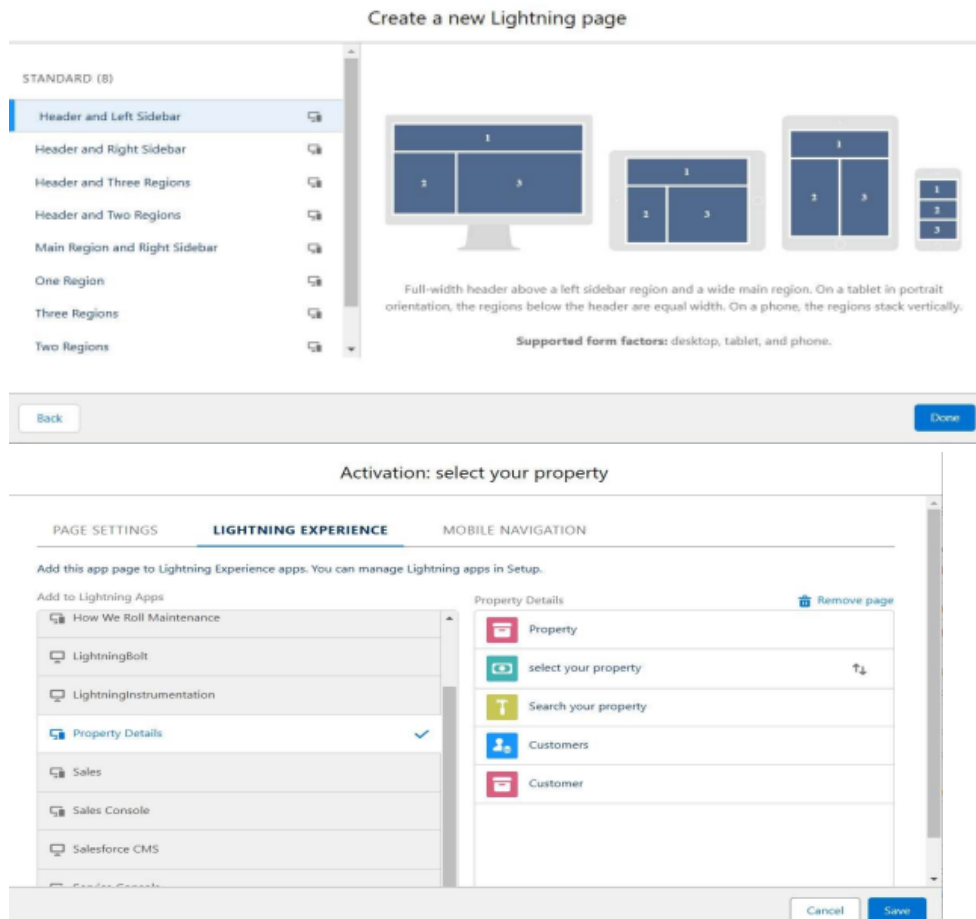
Create an App Page

The screenshot displays the Salesforce Flow Builder interface for a flow named "Property Approval - V1". The flow is a record-triggered flow for the "Property" object, triggered when "A record is created". The flow steps are: Start (Record-Triggered Flow), Run Immediately, Approval for property (Action), and End. 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 on the Property details Object named as “Search Your Property”

Activity1

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



Create a LWC Component

1. 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 Apex Class and make it aura enabled and name it "PropertHandler_LWC"

Code: -

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

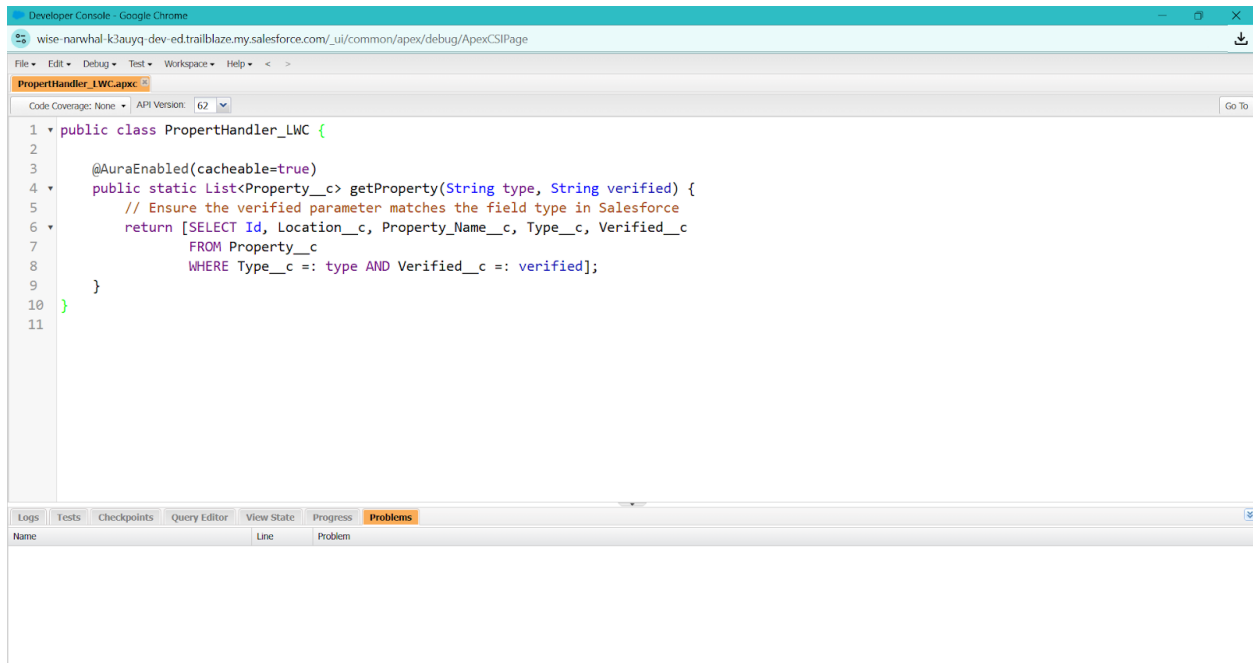
    public static List<Property> getProperty(String type, Boolean verified) (
```

```
String verifiedstr = verified ? 'true' : 'false' // Convert boolean to string

return [SELECT Id, Location c, Property_Namec, Type c, Verified c

FROM Property c

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



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 your Html 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={istrue}>
  <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 <template>
2
3 <lightning-card>
4
5 <div class="slds-box">
6
7 <div class="slds-text-align_left">
8
9 <h1 style="font-size: 20px;"><b>Properties</b></h1>
10
11 </div>
12
13 <div>
14
15 <div class="slds-grid slds-gutters">
16
17 <div class="slds-col slds-size_5-of-6">
18
19 <lightning-combobox name="Type" label="Property Type" value={typevar} placeholder="Select Property type"
20 options={propertyoptions} onchange={changehandler}</lightning-combobox>
21
22 </div>
23
24 <div class="slds-col slds-size_1-of-6">
25
26 <br>
27
28 <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search"
29 label="Search" onclick={handleClick}</lightning-button-icon>
30
31 </div>
32
33 </div>
34
35 </div>
36
37 </div>
38
39 </div>
40
41
42 <template if:true={istru}>
43
44 <div class="slds-box">
45

```

1. In Your Js 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 anagement extends LightningElement ( @api
```

```
recordId
```

```
userId = USER_ID;
```

```
verifiedvar typevar
```

```
isfalse = true; istru
```

```
= false;
```

```
@track propertylist = [];
```

```
columns = [
```

```
( label: 'Property Name', fieldName: 'Property_Name c' ), (
```

```
label: 'Property Type', fieldName: 'Type c' ),
```

```
( label: 'Property Location', fieldName: 'Locationc' ), (
```

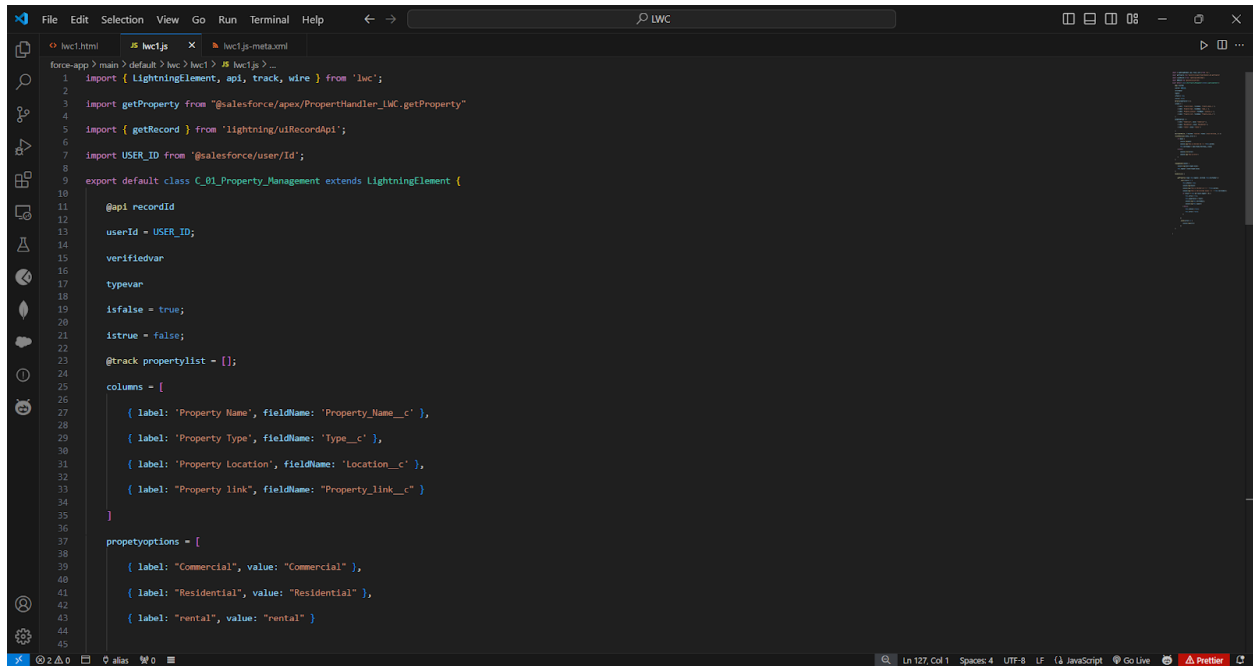
```
label: "Property link", fieldName: "Property link c" }
```

```
propetyoptions = [
```

```

    ( label: "Commercial", value: "Commercial" }, (
    label: "Residential", value: "Residential" ),
    ( label: "rental", value: "rental" }
@wire(get Record, ( 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 import { LightningElement, api, track, wire } from 'lwc';
2
3 import getProperty from '@salesforce/apex/PropertyHandler_LWC.getProperty'
4
5 import { getRecord } from 'lightning/uiRecordApi';
6
7 import USER_ID from '@salesforce/user/Id';
8
9 export default class C_01_Property_Management extends LightningElement {
10
11     @api recordId
12
13     userId = USER_ID;
14
15     verifiedvar
16
17     typevar
18
19     isfalse = true;
20
21     istrue = false;
22
23     @track propertylist = [];
24
25     columns = [
26
27         { label: 'Property Name', fieldName: 'Property_Name__c' },
28
29         { label: 'Property Type', fieldName: 'Type__c' },
30
31         { label: 'Property Location', fieldName: 'Location__c' },
32
33         { label: 'Property link', fieldName: 'Property_link__c' }
34
35     ]
36
37     propertyoptions = [
38
39         { label: 'Commercial', value: 'Commercial' },
40
41         { label: 'Residential', value: 'Residential' },
42
43         { label: 'rental', value: 'rental' }
44
45     ]
46 }
```

1. In Your metafile give your targets to deploy the component.

Code

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

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

```
  <apiVersion>59.0</apiVersion>
```

```
  <isExposed>true</isExposed>
```

```
  <targets>
```

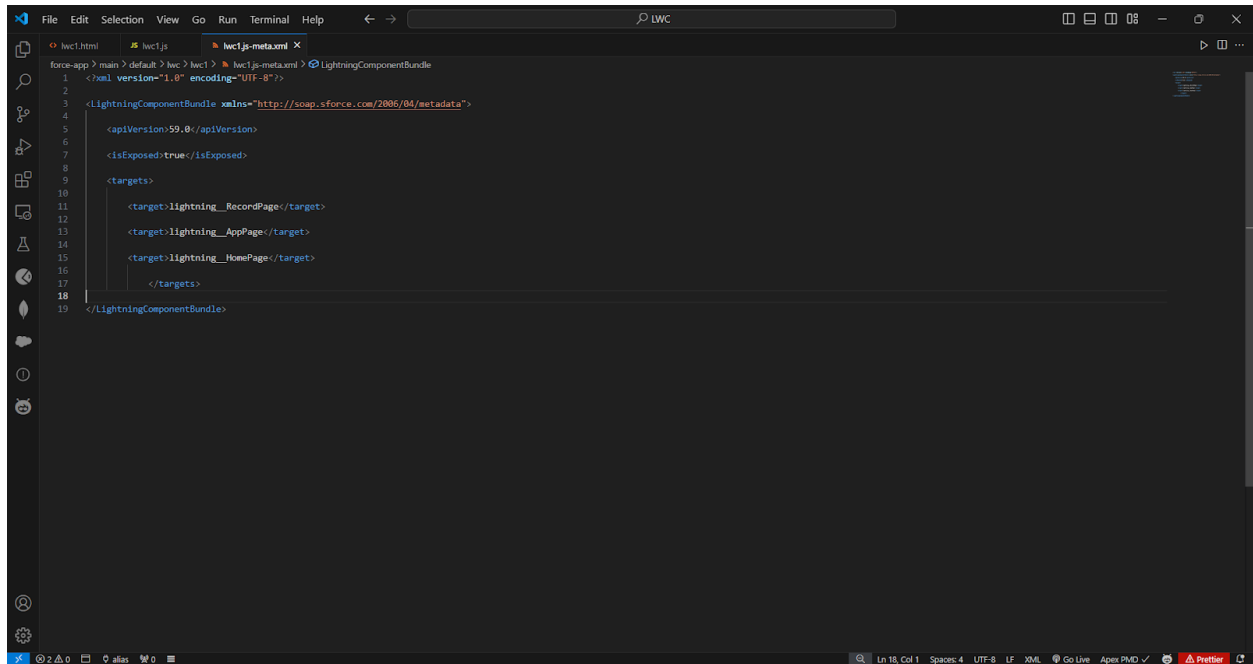
```
    <target>lightning_RecordPage</targets>
```

```
    <target>lightning_AppPage</targets>
```

```
    <target>lightning_HomePage</targets>
```

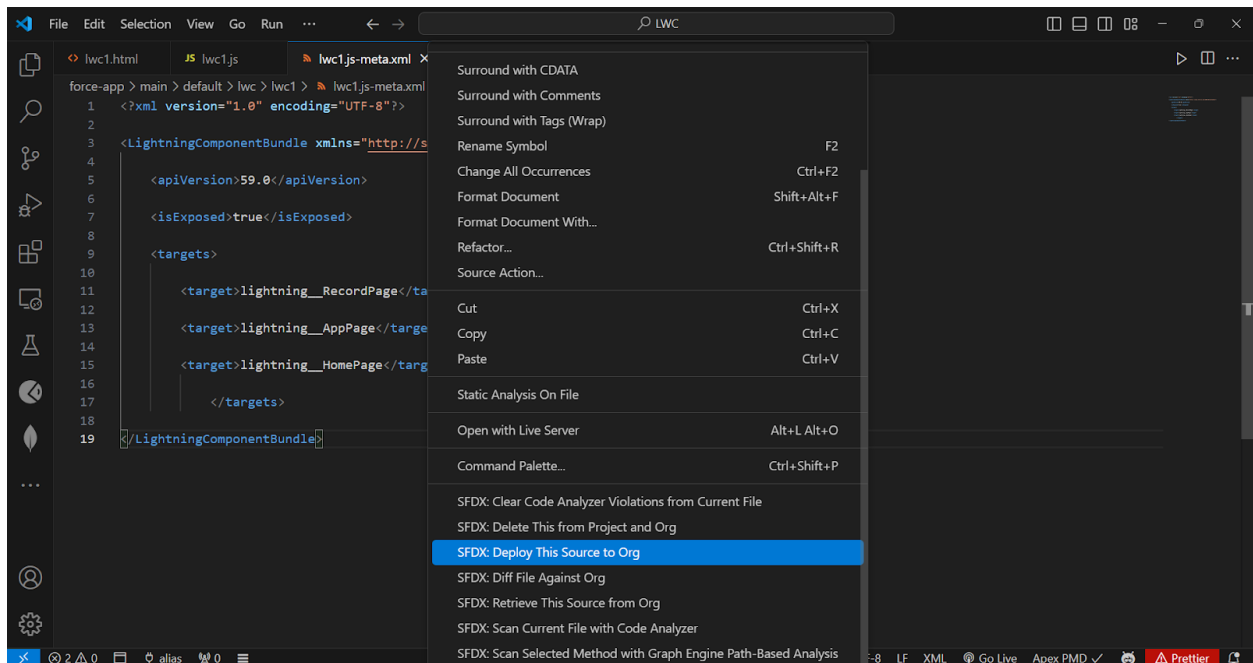
```
  </targets>
```

```
</LightningComponentBundle>
```



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
4   <apiVersion>59.0</apiVersion>
5   <isExposed>true</isExposed>
6   <targets>
7     <target>lightning__RecordPage</target>
8     <target>lightning__AppPage</target>
9     <target>lightning__HomePage</target>
10  </targets>
11 </LightningComponentBundle>
```

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



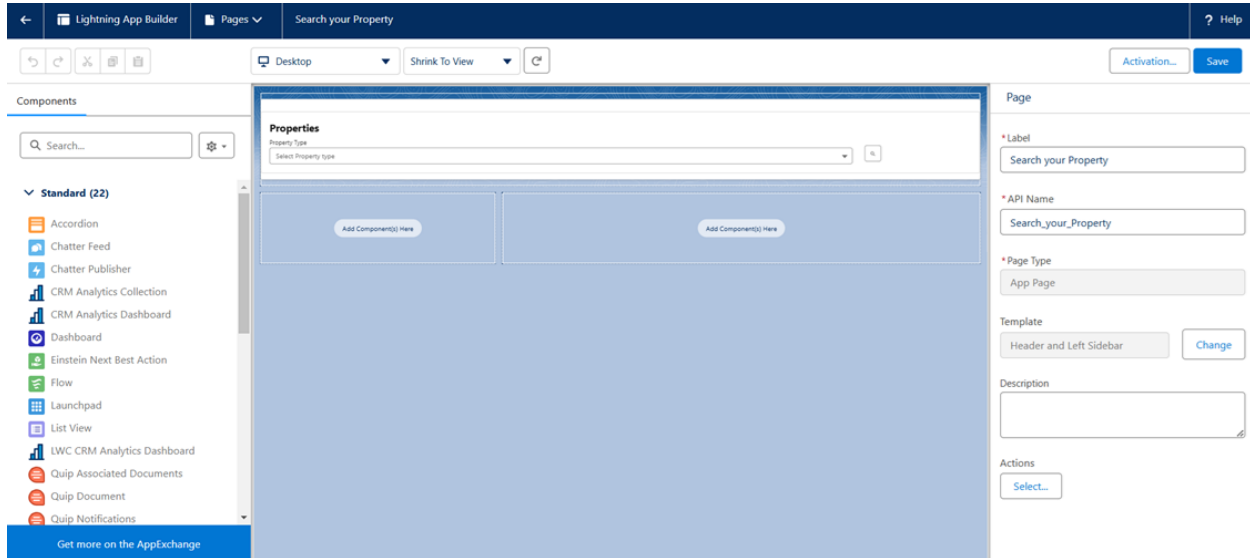
Drag this Component to your App Page

Adding the Component to your Page

Activity1

1. From Setup >> Go to App Launcher >> Search for Property Details

2. On this Page click on gear icon and click on Edit Page
3. after clicking on edit page it will be directed 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
PropertHandler_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