

# **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/243219157995064>

The screenshot shows the Jotform 'Form Builder' interface for a form titled 'Dreams World'. The left sidebar contains a 'Form Elements' menu with various widgets like Text, Text Area, Select, etc. The main workspace displays the form layout. The form fields are:

- Name \***: Two text input fields.
- Phone Number**: A text input field with a placeholder '(e.g., 123)'. Below it is a note: 'please enter a valid phone number'.
- Email**: A text input field with a placeholder 'example@example.com'.
- What type of property are you looking for?**: A dropdown menu with three options: Residential, Commercial, and Rental.
- Budget Amount \***: A text input field with a placeholder '(e.g., 123)'.
- Address**: A text input field.
- Other Contact**: A text input field.
- Home Address (optional)**: A text input field.
- City**: A text input field.
- State / Province**: A text input field.
- Postcode / Zip Code**: A text input field.

A green 'Submit' button is located at the bottom of the form. At the bottom of the Jotform interface, there is a banner for 'Jotform' with a 'Get Jotform' button.

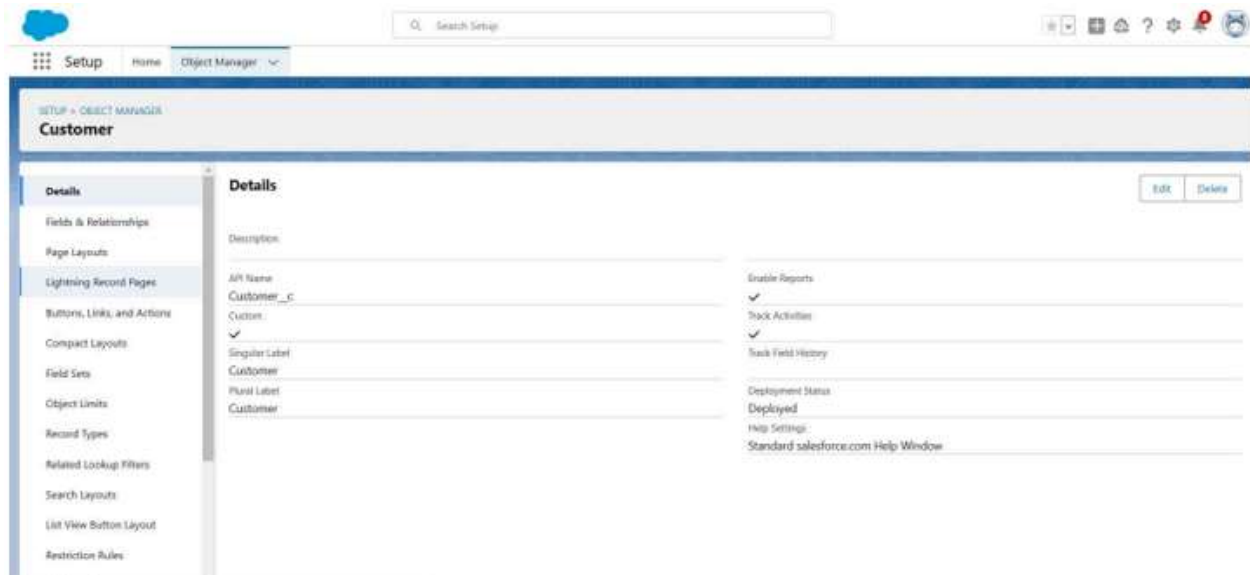
# Create Objects from Spreadsheet

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](#)
4. After downloading, upload the file, map the fields and upload to create an object.

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

## Creating Property Object :

1. Follow the same from the customer object to create the Property Object
2. [Property](#)
3. After downloading, upload the file, map the fields and upload to create an object. the fields as follows

A	B	C	D
Property Name	Type	Location	Verified
Lotus Appartme	Residential	hydeerabad	checked
500000 sq.ft pl	Commercial	Amravati	unchecked
3 Bhk fkat at st	rental	Jubliee hill Hyd	Checked

Setup

Home

Object Manager

Q Search Setup

Setup > OBJECT MANAGER

Property

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Details

Description

API Name  
Property\_\_c

Custom  
✓

Singular Label  
Property

Plural Label  
Property

Enable Reports  
✓

Track Activities  
✓

Track Field History

Deployment Status  
Deployed

Help Settings  
Standard salesforce.com Help Window

EditDelete

# 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 displays the Jotform Form Builder interface for integrating with Salesforce. The left sidebar lists various settings, with 'INTEGRATIONS' selected. The main workspace shows the 'SALESFORCE' integration setup. Under 'Select a Salesforce Object', 'Customer' is chosen. The 'Create a record' section maps Jotform fields to Salesforce fields: Customer ID to Name - First Name, City to Address - City, Budget Amount to Budget Amount, Property Type to Which type of property are you looking..., Phone Number to Phone Number, Street Address to Address - Street Address, Email to Email, Customer Name to Name - Last Name, State to Address - State, and Street Address line 2 to Address - Street Address 2. The top right shows the form name 'Dreamhome' and a timestamp 'Last edited at 12:04 AM'.

Then “Save the Integration” and “Finish”.

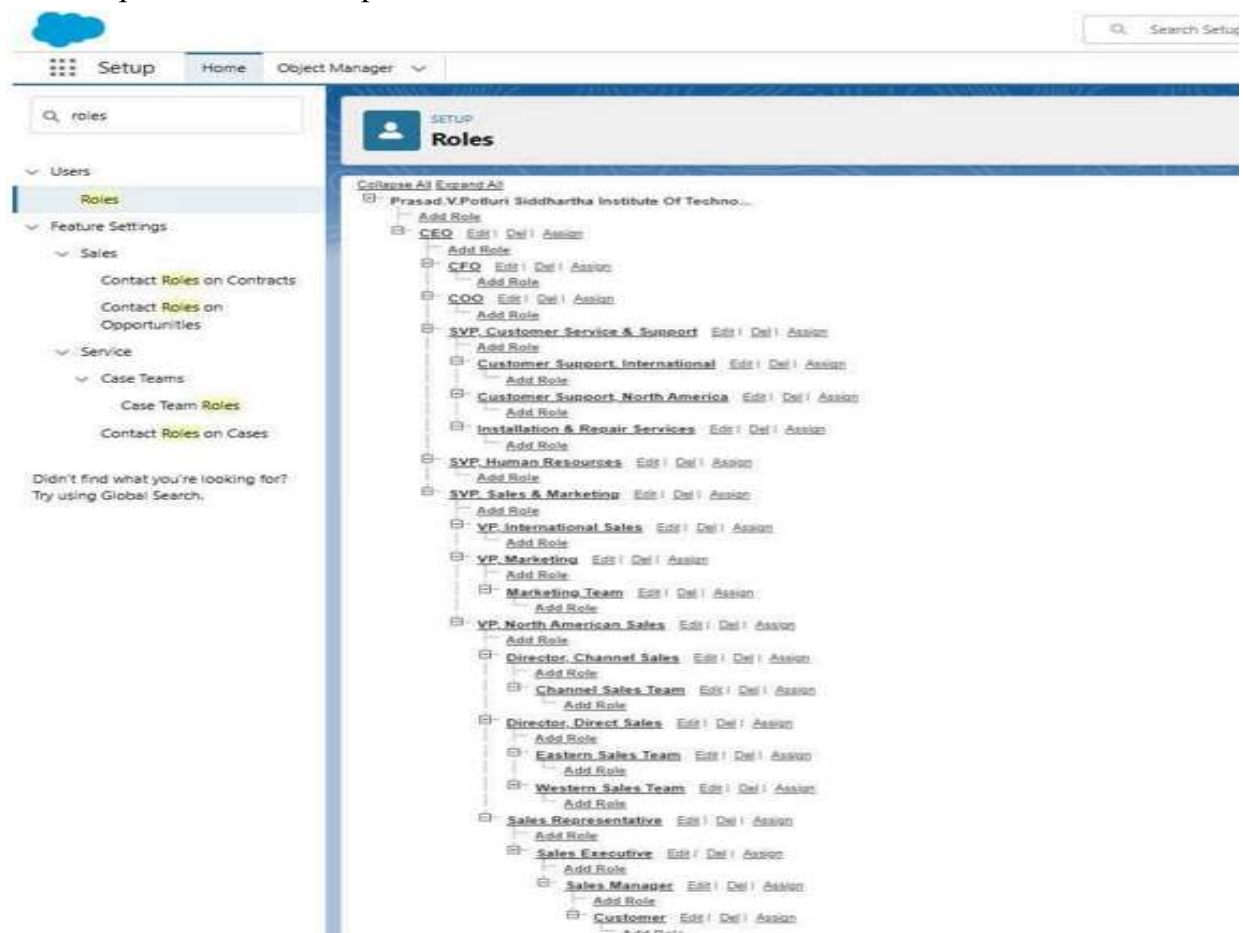
The screenshot displays the 'All Actions' section of the Jotform Form Builder interface for the Salesforce integration. The 'Create or update a record' action is selected for the 'Customer' object. The top right shows the form name 'Dreamhome' and a timestamp 'Last edited at 12:04 AM'.

# 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 AppManager 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.

#### 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

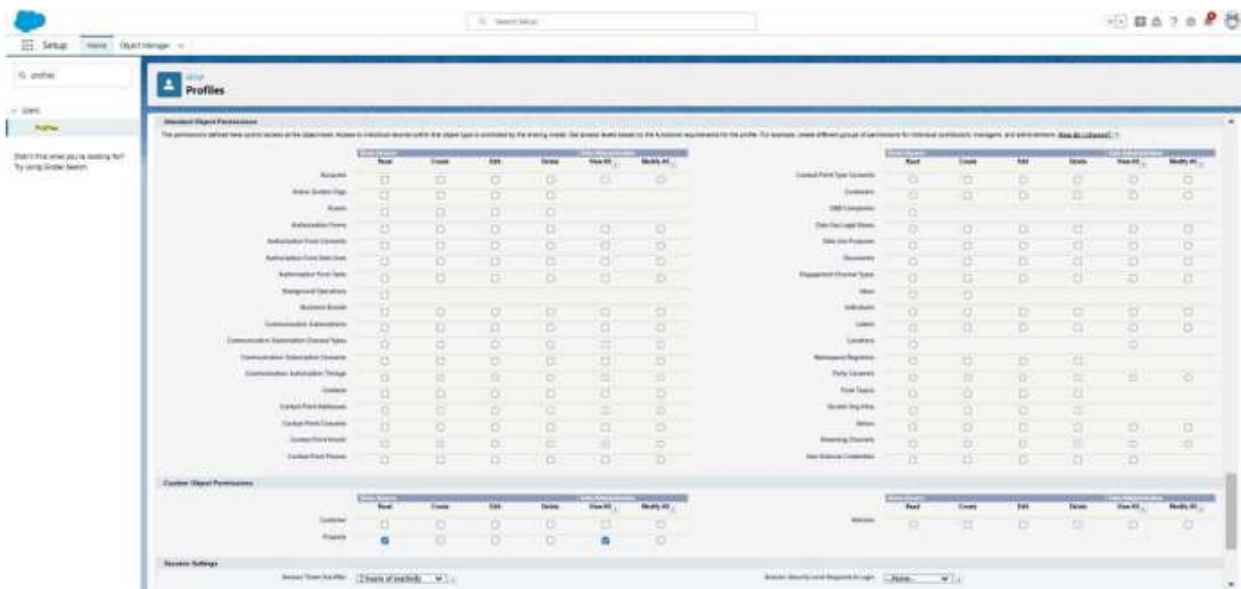


## Create Profiles

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 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”.

**Standard Object Permissions**

The permissions defined here control access to the standard objects. Access to individual records within that object type is controlled by the sharing model. Set access levels based on the functional requirements for the profile. For example, create different groups of permissions for individual contributors, managers, and administrators. [View All Settings](#)

Object Name	Read	Create	Edit	Delete	View All	Modify All
Accounts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Address Customizable Page	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Profiles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Profile Contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Profile Data Sets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorized Roles Tests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Background Operations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Business Events	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Plans	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Plan Content Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Settings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contacts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Addresses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Details	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Profiles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Custom Object Permissions**

Object Name	Read	Create	Edit	Delete	View All	Modify All
Customer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Property	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Standard Object Permissions**

Object Name	Read	Create	Edit	Delete	View All	Modify All
Contact Point Type Contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CRM Composite	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customizable Page	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Use Purpose	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Documents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Engagement Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subscribable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Labels	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Locations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Message Registration	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Party Contents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Push Topics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Search Log Index	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Streaming Channels	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User External Credentials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Custom Object Permissions**

Object Name	Read	Create	Edit	Delete	View All	Modify All
Website	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Session Settings**

Session Timeout (in minutes):  [View All Settings](#)

Session Security Level (Required at Login):  [View All Settings](#)

# 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”

User

Details

Fields & Relationships

User Page Layouts

User Profile Page Layouts

Lightning Record Pages

Buttons and Links

Compact Layouts

Field Sets

Object Limits

Related Lookup Filters

Search Layouts

List View Button Layout

Triggers

User Custom Field

Verified

Back to User Fields

Validation Rules (0)

Custom Field Definition Detail

Edit

Set Field-Level Security

View Field Accessibility

Where is this used?

Field information

Field Label	Verified	Object Name	User
Field Name	Verified	Data Type	Checkbox
API Name	Verified__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	21501A0552.GODAWARTHI KRISHNASRI	Modified By	21501A0552.GODAWARTHI KRISHNASRI
	19/11/2024, 12:04 pm		19/11/2024, 12:04 pm

General Options

Default Value	Unchecked
---------------	-----------

Validation Rules

New

Validation Rules Help

No validation rules defined.

# 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

Executive

[Permission Set Assignments \(0\)](#) | [Permission Set Assignments: Activation Required \(0\)](#) | [Permission Set Group Assignments \(0\)](#) | [Permission Set License Assignments \(0\)](#) | [Personal Groups \(0\)](#) | [Public Group Membership \(0\)](#) | [Qu](#)  
[Managers in the Role Hierarchy \(0\)](#) | [OAuth Apps \(0\)](#) | [Third-Party Account Links \(0\)](#) | [Installed Mobile Apps \(0\)](#) | [Authentication Settings for External Systems \(0\)](#) | [Login History \(0+\)](#) | [User Provisioning Ac](#)

### User Detail

[Edit](#) [Sharing](#) [Reset Password](#) [Freeze](#) [View Summary](#)

Name	Executive	Role	<a href="#">Sales Executive</a>
Alias	exec	User License	Salesforce
Email	<a href="mailto:krishna1@gmail.com">krishna1@gmail.com</a> <a href="#">[Verify]</a> <a href="#">i</a>	Profile	<a href="#">System Administrator</a>
Username	<a href="mailto:vedasree@gmail.com">vedasree@gmail.com</a>	Active	<input checked="" type="checkbox"/>
Nickname	<a href="#">User17319155391242007709</a> <a href="#">i</a>	Marketing User	<input type="checkbox"/>

## 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

Manager

[Permission Set Assignments \(0\)](#) | [Permission Set Assignments: Activation Required \(0\)](#) | [Permission Set Group Assignments \(0\)](#) | [Permission Set License Assignments \(0\)](#) | [Personal Groups \(0\)](#) | [Public Group M](#)  
[Managers in the Role Hierarchy \(1\)](#) | [OAuth Apps \(0\)](#) | [Third-Party Account Links \(0\)](#) | [Installed Mobile Apps \(0\)](#) | [Authentication Settings for External Systems \(0\)](#) | [Login History \(0+\)](#) | [!](#)

### User Detail

[Edit](#) [Sharing](#) [Reset Password](#) [Freeze](#) [View Summary](#)

Name	Manager	Role	<a href="#">Sales Manager</a>
Alias	mana	User License	Salesforce Platform
Email	<a href="mailto:krish@gmail.com">krish@gmail.com</a> <a href="#">[Verify]</a> <a href="#">i</a>	Profile	<a href="#">Manager</a>
Username	<a href="mailto:vedasree11@gmail.com">vedasree11@gmail.com</a>	Active	<input checked="" type="checkbox"/>
Nickname	<a href="#">User17319156325169574707</a> <a href="#">i</a>	Marketing User	<input type="checkbox"/>

## 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 verifiedcheckboxois"Unchecked"
7. Save

User

Customer

[Permission Set Assignments \(0\)](#) | [Permission Set Assignments: Activation Required \(0\)](#) | [Permission Set Group Assignments \(0\)](#) | [Permission Set License Assignments \(0\)](#) | [Personal Groups \(0\)](#) | [Public Group Membership \(0\)](#) | [Managers in the Role Hierarchy \(0\)](#) | [OAuth Apps \(0\)](#) | [Third-Party Account Links \(0\)](#) | [Installed Mobile Apps \(0\)](#) | [Authentication Settings for External Systems \(0\)](#) | [Login History \(0+\)](#) | [User Provisioning](#)

### User Detail

[Edit](#) [Sharing](#) [Reset Password](#) [Freeze](#) [View Summary](#)

Name	Customer	Role	Customer
Alias	cust	User License	Salesforce Platform
Email	krishnasri@gmail.com <a href="#">[Verify]</a>	Profile	Customer
Username	vedasri@gmail.com	Active	<input checked="" type="checkbox"/>
Nickname	User17319157178427587980 <a href="#">[i]</a>	Marketing User	<input type="checkbox"/>

## 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

User

Customer2

[Permission Set Assignments \(0\)](#) | [Permission Set Assignments: Activation Required \(0\)](#) | [Permission Set Group Assignments \(0\)](#) | [Permission Set License Assignments \(0\)](#) | [Personal Groups \(0\)](#) | [Public Group Membership \(0\)](#) | [Managers in the Role Hierarchy \(0\)](#) | [OAuth Apps \(0\)](#) | [Third-Party Account Links \(0\)](#) | [Installed Mobile Apps \(0\)](#) | [Authentication Settings for External Systems \(0\)](#) | [Login History \(0+\)](#) | [User Provisioning](#)

### User Detail

[Edit](#) [Sharing](#) [Reset Password](#) [Freeze](#) [View Summary](#)

Name	Customer2	Role	Customer
Alias	cust	User License	Salesforce Platform
Email	krishnasree@gmail.com <a href="#">[Verify]</a>	Profile	Customer
Username	vedasri11@gmail.com	Active	<input checked="" type="checkbox"/>
Nickname	User1731915845860701539 <a href="#">[i]</a>	Marketing User	<input type="checkbox"/>

# Create an Approval Process for Property Object

An Approval process to approve or reject the records as according

## Activity 1

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.


### Approval Process Edit Property Approval

**Step 5. Select Fields to Display on Approval Page Layout**

The approval page is where an approver will actually approve or reject a request. Using the options below, choose the fields to display on this page.

Available Fields		Selected Fields
Created By	Add Remove	Property
Last Modified By		Owner
Verified		Location
		Property Name
		Type

Up  
Down



[Click here to view an example](#)

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.

Approval Processes  
Property: Property Approval  
[Back to Approval Process List](#)

Help for this Page

### Process Definition Detail

[Edit](#) [Clone](#) [Deactivate](#)

Process Name	Property Approval	Active	✓
Unique Name	Property_Approval	Next Automated Approver Determined By	Manager of Record Submitter
Description	(Property: Location NOT EQUAL TO blank) AND (Property: Verified EQUALS False)		
Entry Criteria	Administrator OR Current Approver		
Record Editability	Allow Submitters to Recall Approval Requests <input type="checkbox"/>		
Approval Assignment Email Template	Bde_Sales_Manager_Property_Owner		
Initial Submitters	Bde_Sales_Manager_Property_Owner		
Created By	2150140552 GODAVARTHI KRISHNABR	Modified By	2150140552 GODAVARTHI KRISHNABR
	18/11/2024, 1:23 pm		18/11/2024, 3:40 pm

### Initial Submission Actions

[Add Existing](#) [Add New](#)

Action	Type	Description
<a href="#">Edit</a>	Record Lock	Lock the record from being edited

### Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
<a href="#">Show Actions</a> <a href="#">Edit</a>	1	Executive Approval			User_Executive	Final Rejection

### Final Approval Actions

[Add Existing](#) [Add New](#)

Action	Type	Description
<a href="#">Edit</a>	Record Lock	Lock the record from being edited
<a href="#">Edit</a> <a href="#">Remove</a>	Field Update	Verified Property

### Final Rejection Actions

[Add Existing](#) [Add New](#)

Action	Type	Description
<a href="#">Edit</a>	Record Lock	Unlock the record for editing
<a href="#">Edit</a> <a href="#">Remove</a>	Field Update	UnVerified Property

# Create a Recordtriggerer 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”

Last saved on 17/11/2024, 11:49 am **Active** Run Debug View Tests Save As New Version Save Deactivate



**Submit for Approval**

\* Label  
Approval for property

\* API Name ⓘ  
Approval\_for\_property

Description

**Submit for Approval** ⓘ  
submit-submit

Use values from earlier in the flow to set the inputs for the “Submit for Approval” core action. To use its outputs later in the flow, store them in variables.

Set Input Values for the Selected Action

A<sub>a</sub> \* Record ID ⓘ  
A<sub>a</sub> Triggering Property\_\_c > Record ID X

# Create a Recordtrigger ñow to submit the Approval

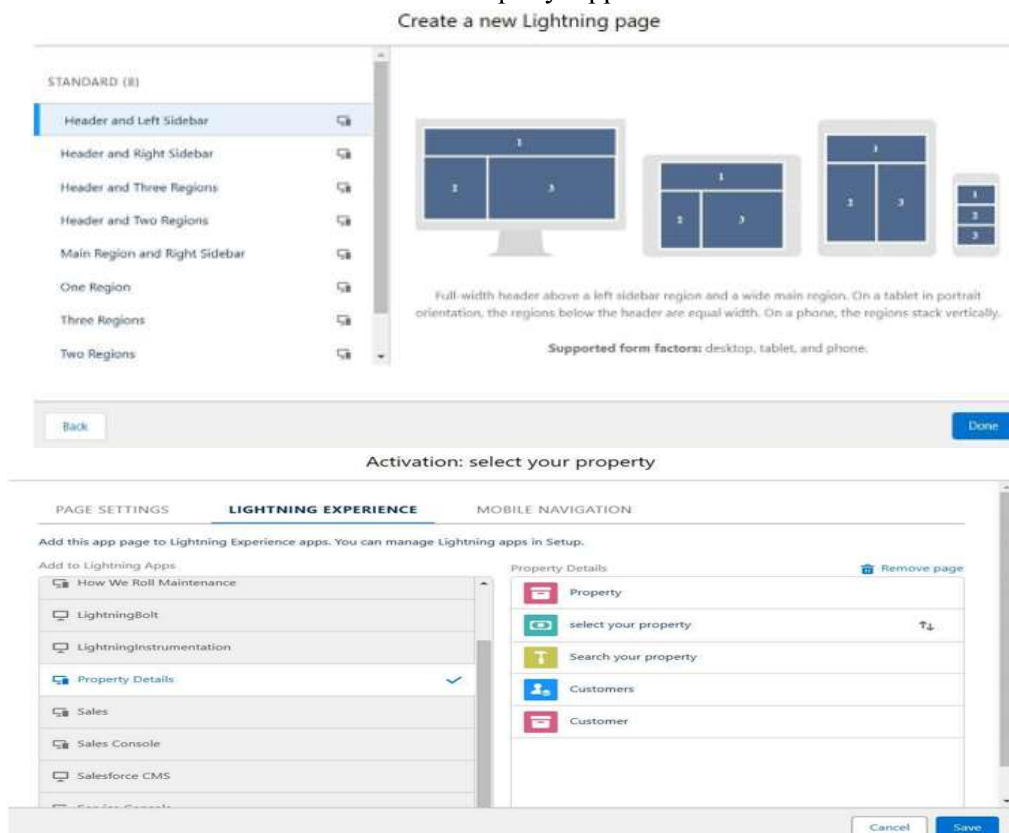
## Process Automatically

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”





# Create a LWC Component

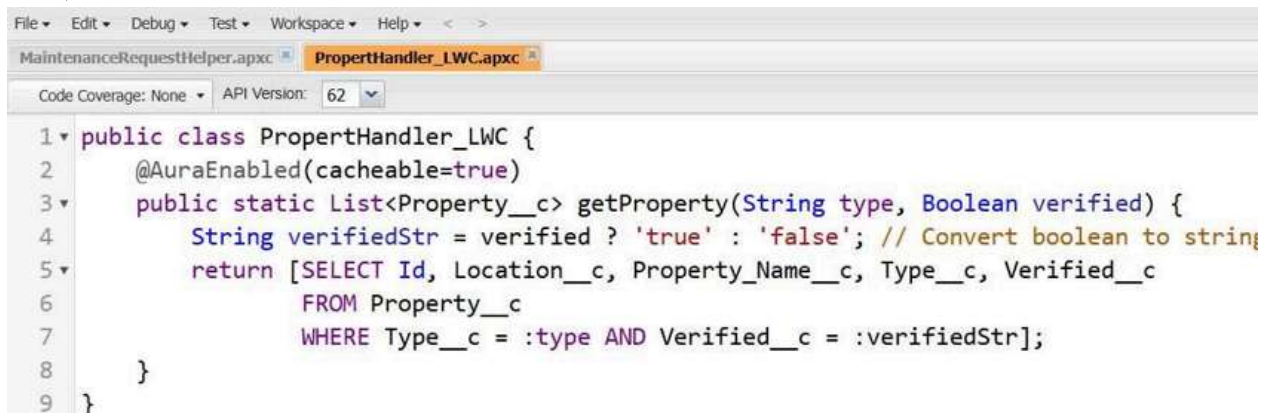
- a. Create an LWCComponent for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on “Search your Property Page”

## 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__c> 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. 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={propertyoptions} 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>

</div>

<template if:true={istrue}>

  <div class="slds-box">

    <lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-
datatable>

  </div>

</template>

<template if:false={isfalse}>

  <div class="slds-box">

    <div style="font-size: 15px;"><b>No properties Are Found !!</b></div>

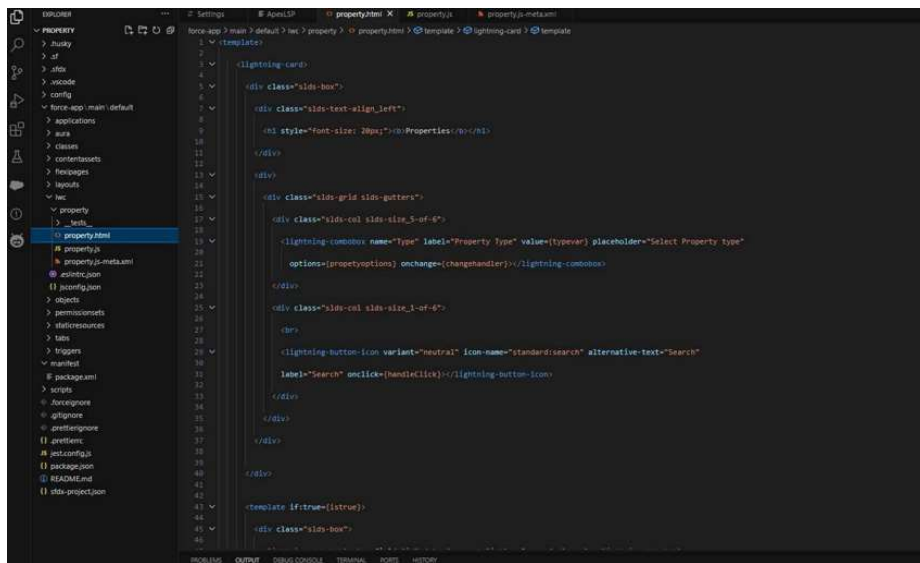
  </div>

</template>

</lightning-card>

</template>

```



1. In YourJs File Write this code :-

Code :-

```
import ( LightningElement, api, track, wire ) from 'lwc';
```

```
import getProperty from "@salesforce/apex/PropertyHandler_LWC.getProperty"
```

```
import ( getRecord } from 'lightning/uiRecordApi'; import USER
```

```
ID from '@salesforce/user/Id';
```

```
export default class C 01_Property_Management extends 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" }
```

```
propertyoptions= [
```

```
( 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 (

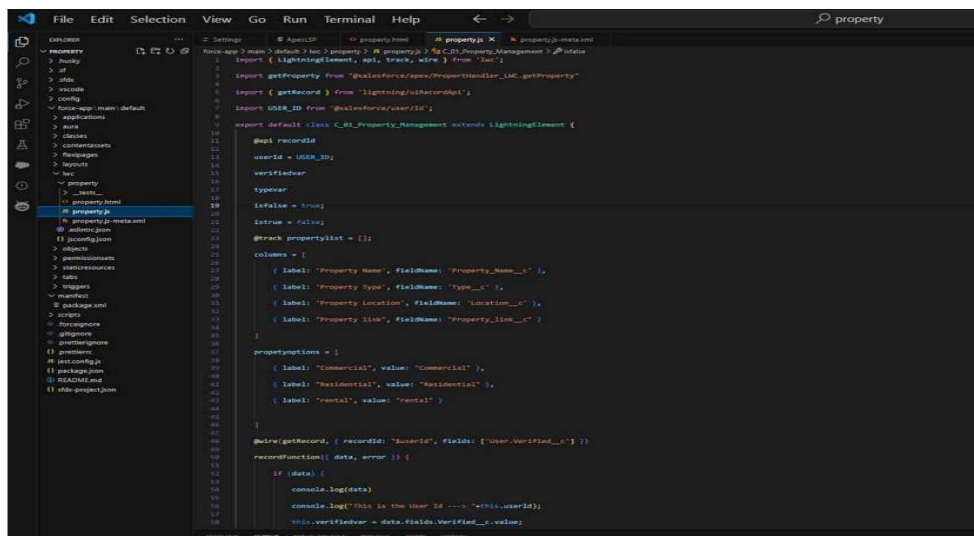
this.isfalse = false;

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>
```

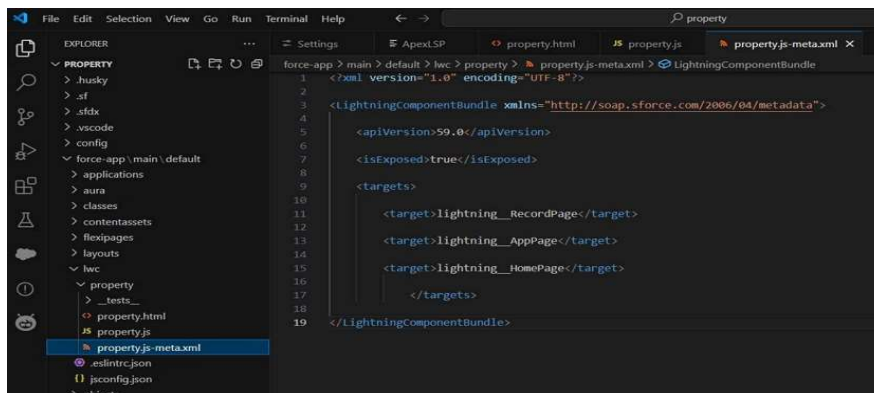
```
    <target>Lightning__RecordPage</target>
```

```
    <target>Lightning__AppPage</target>
```

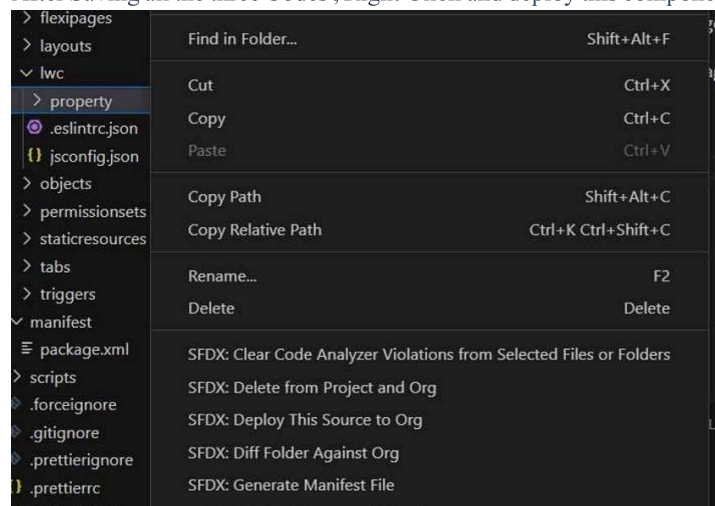
```
    <target>Lightning__HomePage</target>
```

```
  </targets>
```

```
</LightningComponentBundle>
```



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

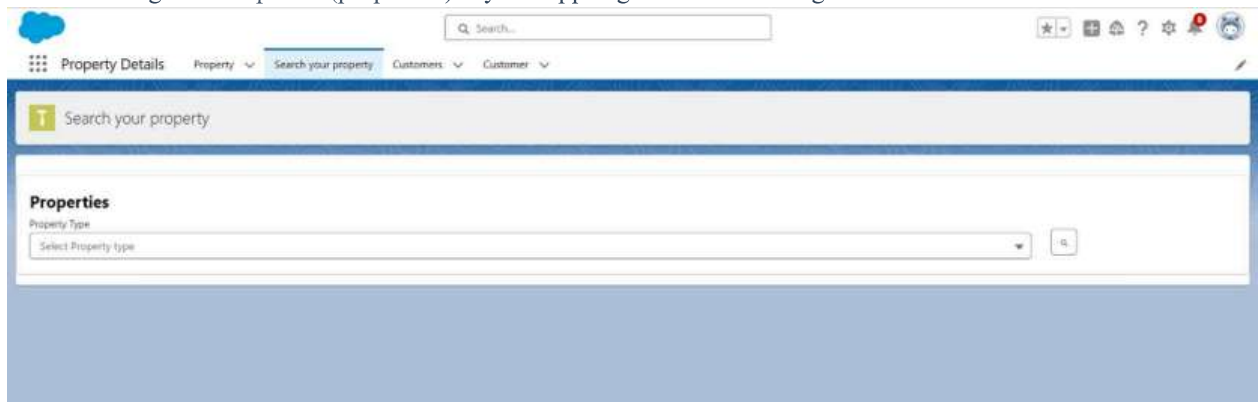


# Drag this Component to your App Page

Adding the Component to your Page

## Activity1

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



# 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 this Page click on gear icon and click on Edit Page
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.

The screenshot displays the Salesforce Setup interface. At the top, there is a search bar labeled 'Search Setup'. Below it, the navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar shows a search for 'apex class' under the 'Custom Code' section, with 'Apex Classes' listed below. The main content area is titled 'SETUP Profiles' and shows the configuration for 'PropertyHandler\_LWC'. It includes a 'Save' button and a 'Cancel' button. The 'Available Profiles' list on the left includes: 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, and Customer Portal Manager Custom. The 'Enabled Profiles' list on the right includes: Customer Manager and System Administrator. Arrows and 'Add'/'Remove' buttons are used to manage the profile assignments.