

# **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 useable for sale, rent, or lease.
- c. Upload photos and documents for properties.

### **3. Requirement Matching**

- a. Match client demand with useable properties using filters.
- b. Notify clients about new properties that fit their measure.

### **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.

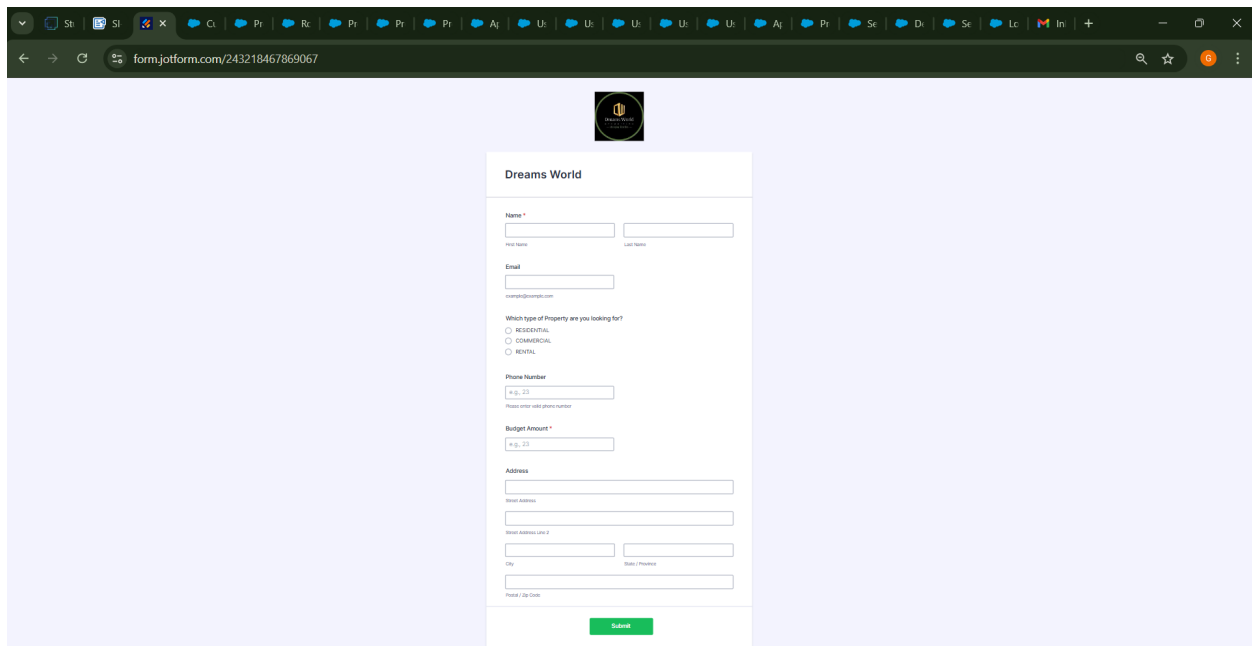
## Milestone 1: Create a Jot form and integrate it with the org to create a record of customers automatically.

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 organization. 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 organization.

### Activity1

Open your browser and search for jot form 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 the URL [form.jotform.com/21501a0534/SweetHome](https://form.jotform.com/21501a0534/SweetHome). The form is titled "Dreams World" and contains the following fields:

- Name \***: Two input fields for First Name and Last Name.
- Email**: One input field.
- Which type of Property are you looking for?**: Three radio button options: RESIDENTIAL, COMMERCIAL, and RENTAL.
- Phone Number**: One input field with a placeholder "e.g., 23" and a note "Please enter valid phone number".
- Budget Amount \***: One input field with a placeholder "e.g., 23".
- Address**: Four input fields for Street Address, Apartment/Suite #, City, and State / Province.
- Postcode / Zip Code**: One input field.
- Submit**: A green button at the bottom.

# Create Objects from Spreadsheet

Directly Creating Objects from a 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.

creative-badger-57g1y9-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager/01IWU000001qOGz/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER

**Customer**

Details

**Fields & Relationships**  
15 Items, Sorted by Field Label

Q Quick Find New Deleted Fields Field Dependencies Set History Tracking

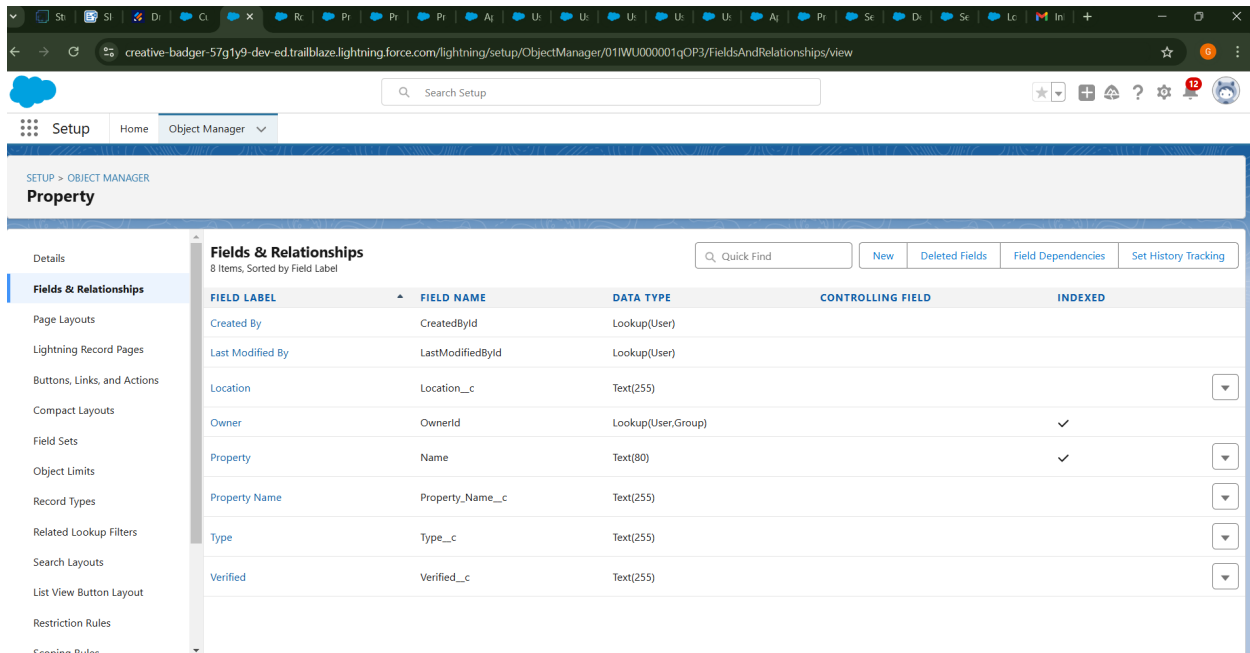
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Budget Amount	Budget_Amount__c	Number(18, 0)		
City	City__c	Text(255)		
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Text(255)		
Customer	Name	Text(80)		✓
Emial	Emial__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	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 pl	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 are as follows



The screenshot shows the Salesforce Setup interface for the 'Property' object. The 'Fields & Relationships' section is active, 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)		

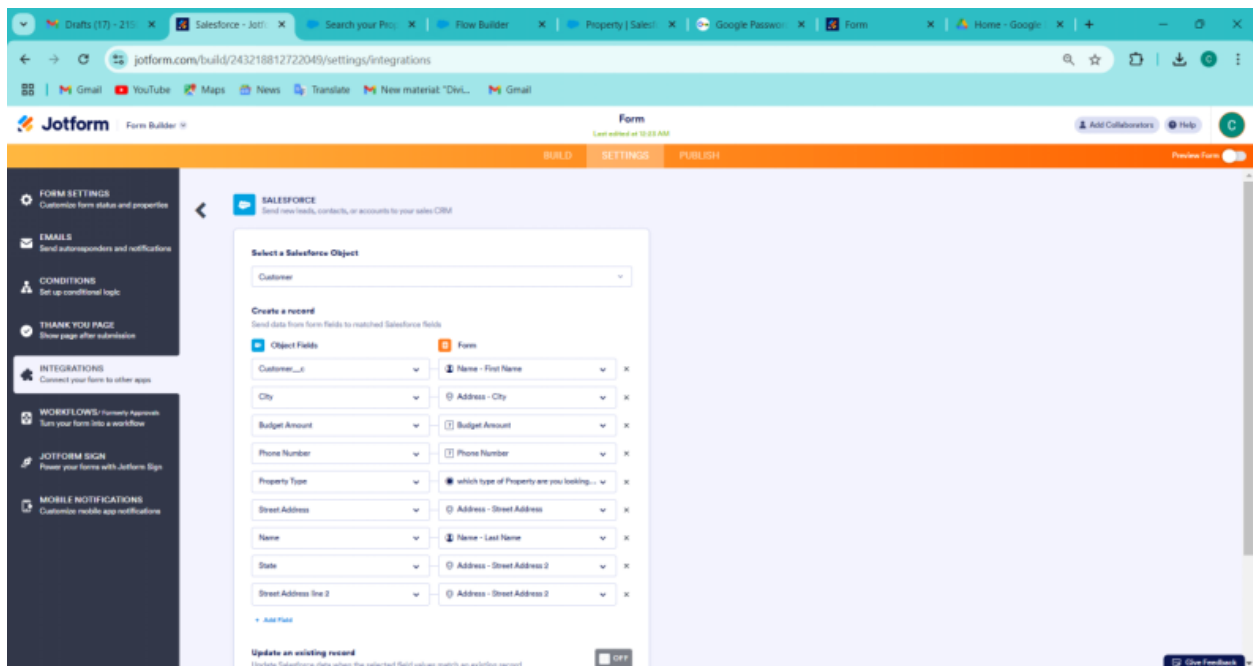
# Integrate Jot form with Salesforce Platform

In this Milestone, we are going to integrate jot form with Salesforce

## Activity

1. On the Jot form Platform, Click on Integration and choose Salesforce
2. Click on User Integration and choose “Add to From”
3. Select the Organization with which you want to Integrate your jot form with and select your account
4. Select an Action - Create a record.
5. Select a Salesforce Object : - Customer

Map Every field on the Object with the fields on the form and “Save Action”.




The screenshot shows the Jotform 'Form Builder' interface for a Salesforce integration. The left sidebar contains navigation options: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (selected), WORKFLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The main area is titled 'SALESFORCE' and shows the 'Select a Salesforce Object' dropdown set to 'Customer'. Below this, the 'Create a record' section is active, displaying a table to map form fields to Salesforce fields. The table has two columns: 'Object Fields' and 'Form'. The mappings are as follows:

Object Fields	Form
Customer__c	Name - First Name
City	Address - City
Budget Amount	Budget Amount
Phone Number	Phone Number
Property Type	which type of Property are you looking...
Street Address	Address - Street Address
Name	Name - Last Name
State	Address - Street Address 2
Street Address line 2	Address - Street Address 2

At the bottom, there is an 'Update an existing record' section with a toggle switch set to 'OFF'. The top of the interface shows the Jotform logo, 'Form Builder', and navigation tabs for BUILD, SETTINGS, and PUBLISH.

Then “Save the Integration” and “Finish.”

**SALESFORCE**  
Send new leads, contacts, or accounts to your sales CRM

All Actions

🕒 See Action Logs

+ Add New Action

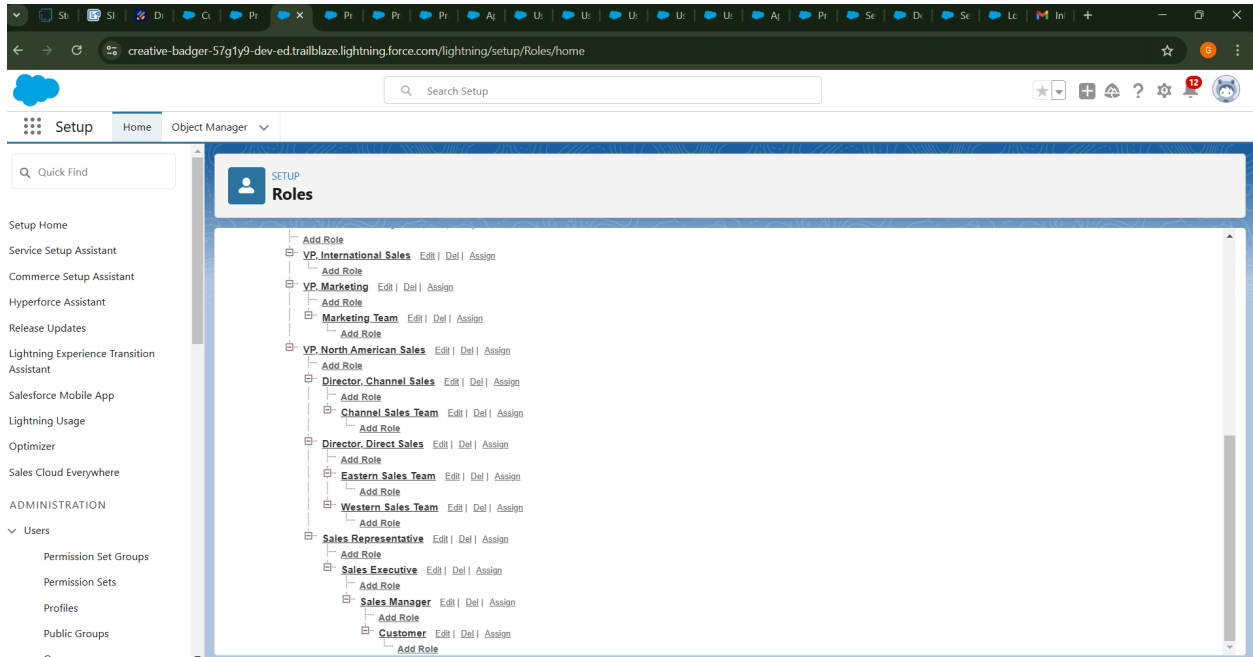
1

Create or update a record  
Customer

# Create Roles

here we need to Create Roles as per business essential

## Activity:- 1



1. if we don't find a 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" Objects.
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.

The screenshot displays the Salesforce Lightning App Builder interface. The top navigation bar includes 'Lightning App Builder', 'App Settings', 'Pages', and 'Property Details'. The left sidebar shows 'App Settings' with a sub-menu 'App Details & Branding'. The main content area is titled 'App Details & Branding' and contains the following sections:

- App Details:** Includes fields for 'App Name' (Property Details), 'Developer Name' (Property\_Details), and 'Description' (Enter a description...).
- App Branding:** Includes an 'Image' upload button, a 'Primary Color Hex Value' dropdown set to '#0070D2', and 'Org Theme Options' with a checkbox 'Use the app's image and color instead of the org's custom theme'.
- App Launcher Preview:** Shows a preview of the app launcher with a blue square icon containing 'PD' and the text 'Property Details'.



# 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

The screenshot shows the Salesforce Setup interface for creating a profile. The 'Profiles' page is displayed, showing two tables: 'Standard Object Permissions' and 'Custom Object Permissions'.

**Standard Object Permissions**

Object	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"/>
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Forms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Data Uses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Texts	<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 Brands	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscriptions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Timings	<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 Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Emails	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Phones	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Type Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D&B Companies	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Use Legal Bases	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Use Purposes	<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"/>
Ideas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Individuals	<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"/>
Party Consents	<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"/>
Sellers	<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	Read	Create	Edit	Delete	View All	Modify All
Customer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Property	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

# 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.”

The screenshot shows the Salesforce Setup interface for creating a profile. The main content area is titled "Profiles" and contains two tables for defining permissions.

**Standard Object Permissions**

	Basic Access				Data Administration	
	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"/>
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Forms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Data Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Texts	<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 Brands	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscriptions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communication Subscription Timings	<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 Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Emails	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Phones	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Custom Object Permissions**

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Customer	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Property	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

# Create a Check Box field on user

Create a 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 a new Field Named as “Verified”

The screenshot displays the Salesforce Setup interface for defining a custom field. The browser address bar shows the URL: `creative-badger-57g1y9-dev-ed.trailblaze.lightning.force.com/lightning/setup/ObjectManager/User/FieldsAndRelationships/00NWU000005rMeE/view`. The left sidebar contains a navigation menu with options like 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, and Flow Triggers. The main content area is titled 'User Custom Field' and 'Verified'. It includes a 'Back to User Fields' link and a 'Validation Rules' link. The 'Custom Field Definition Detail' section has tabs for 'Edit', 'Set Field-Level Security', 'View Field Accessibility', and 'Where is this used?'. The 'Field Information' table lists the following details:

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	GHANTASALA SAI SRAVANI	Modified By	GHANTASALA SAI SRAVANI
	18/11/2024, 1:42 pm		18/11/2024, 1:42 pm

The 'General Options' section shows the 'Default Value' as 'Unchecked'. The 'Validation Rules' section indicates 'No validation rules defined.' and includes a 'New' button and a 'Validation Rules Help' link. At the bottom, there is a 'Back To Top' link and a note to 'Always show me more records per related list'.

## Create Users

Create three different users with three different Roles and profiles, as we have mentioned above. Here, we are going to create four 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. Ensure the verified check box is "Unchecked"

## 7. Save

# 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. Ensure the verified check box is "checked"
7. Save

The screenshot shows the Salesforce Setup page for managing users. The browser address bar indicates the URL: `creative-badger-57g1y9-dev-ed.trailblaze.lightning.force.com/lightning/setup/ManageUsers/home`. The page title is "Users" under the "SETUP" tab. The "All Users" section provides instructions on creating, viewing, and managing users, with a link to "Let's Go" for more licenses. Below this, there are buttons for "New User", "Reset Password(s)", and "Add Multiple Users". A table lists the current users with columns for Action, Full Name, Alias, Username, Role, Active status, and Profile. The table contains 8 rows of user data, including Chatter Expert, Customer, Customer2, Executive, Manager, SAI SRAVANI GHANTASALA, User Integration, and User Security. Each row has an "Edit" link and a checked "Active" checkbox. At the bottom of the table, there are again buttons for "New User", "Reset Password(s)", and "Add Multiple Users".

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit	Chatter Expert	Chatter	chatter_00duv00000cga2az.jydukkisqet@chatter.salesforce.com		✓	Chatter Free User
<input type="checkbox"/> Edit	Customer	cust	gsa1123@gmail.com	Customer	✓	Customer
<input type="checkbox"/> Edit	Customer2	cust	gsa1123@gmail.com	Customer	✓	Customer
<input type="checkbox"/> Edit	Executive	exec	gsa1123@gmail.com	Sales Executive	✓	System Administrator
<input type="checkbox"/> Edit	Manager	mana	gsa1123@gmail.com	Sales Manager	✓	Manager
<input type="checkbox"/> Edit	SAI SRAVANI GHANTASALA	GSAI	21501a0550@creative-badger-57g1y9.com		✓	System Administrator
<input type="checkbox"/> Edit	User Integration	integ	integration@00duv00000cga2az.com		✓	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User Security	sec	insightsecurity@00duv00000cga2az.com		✓	Analytics Cloud Security User

# 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 Ten Manage Approval process
  3. Process Name - Property Approval
  4. select 2 standard -
  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 Edit ability 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.

The screenshot displays the Salesforce Setup interface. The left sidebar contains navigation links such as Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, ADMINISTRATION (Users, Data, Email), and PLATFORM TOOLS (Apps). The main content area is titled "Approval Processes" and is specific to the "Property" object. It includes a "Manage Approval Processes For:" dropdown set to "Property". Below this, a "Create New Approval Process" button is visible. The "Active Approval Processes" section shows a table with one entry: "Property Approval" with a process order of 1. The "Inactive Approval Processes" section indicates that no approval processes are available.

Setup

Home Object Manager

Quick Find

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

Create New Approval Process

Active Approval Processes

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

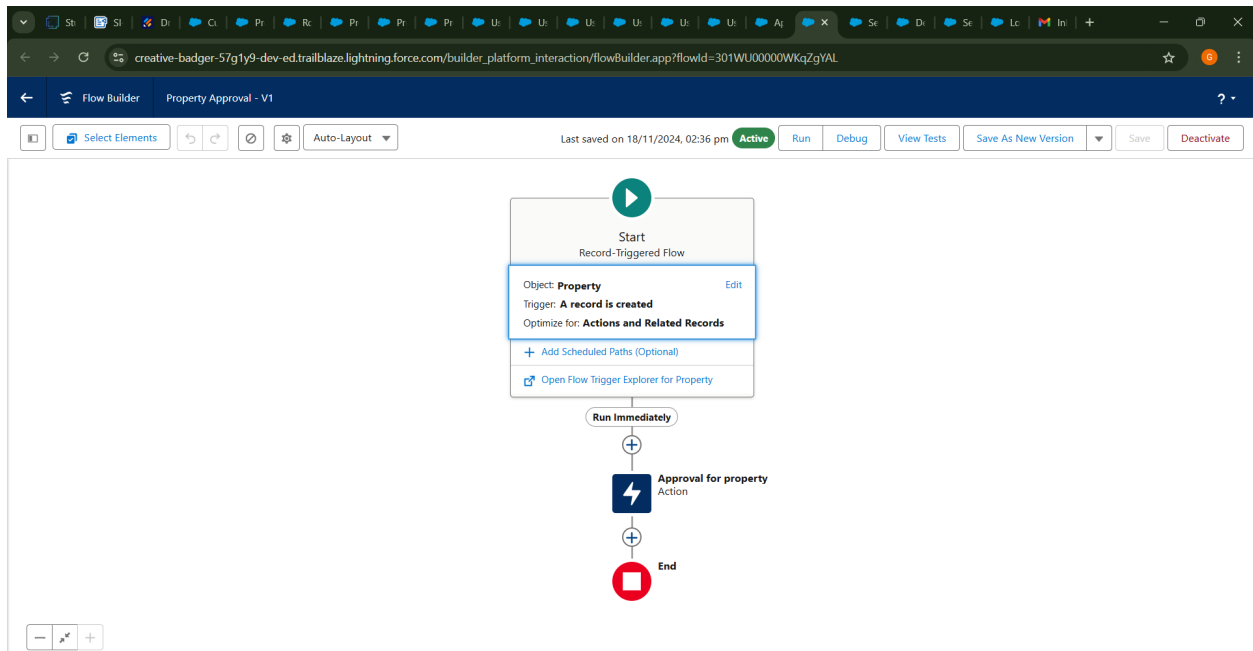
## Process Automatically

A flow that can submit the records directly for approval

### Activity1

- i. From Setup >> Search for Flows >> Click On New and Select “Record Trigger Flow”.
- ii. Select Object >> Property
- iii. Select “Trigger the flow when” >> “A record is created”
- iv. Set Entry Conditions >> “None”
- v. Add an “Action” >> “Submit for Approval”
- vi. Give Label >> Approval for property
- vii. Record Id >> (!SRecord.Id)
- viii. Done

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



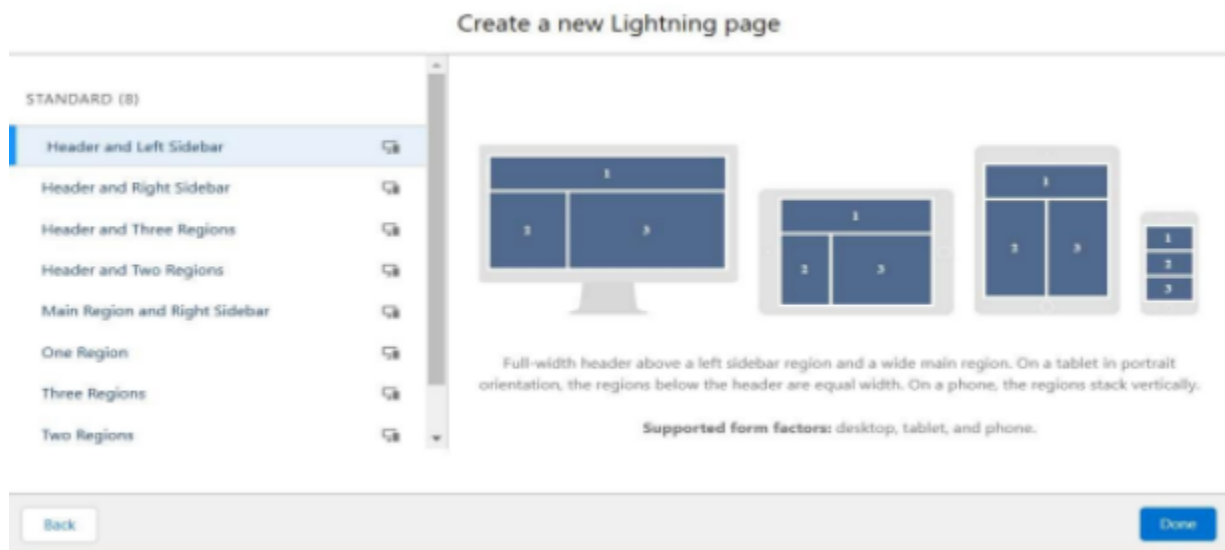


# Create an App Page

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 on “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”



## Activation: select your property

PAGE SETTINGS

**LIGHTNING EXPERIENCE**

MOBILE NAVIGATION

Add this app page to Lightning Experience apps. You can manage Lightning apps in Setup.

Add to Lightning Apps

How We Roll Maintenance

LightningBolt

LightningInstrumentation

Property Details



Sales

Sales Console

Salesforce CMS

Salesforce CMS

Property Details

Remove page

Property

select your property



Search your property

Customers

Customer

Cancel

Save

# Create a LWC Component

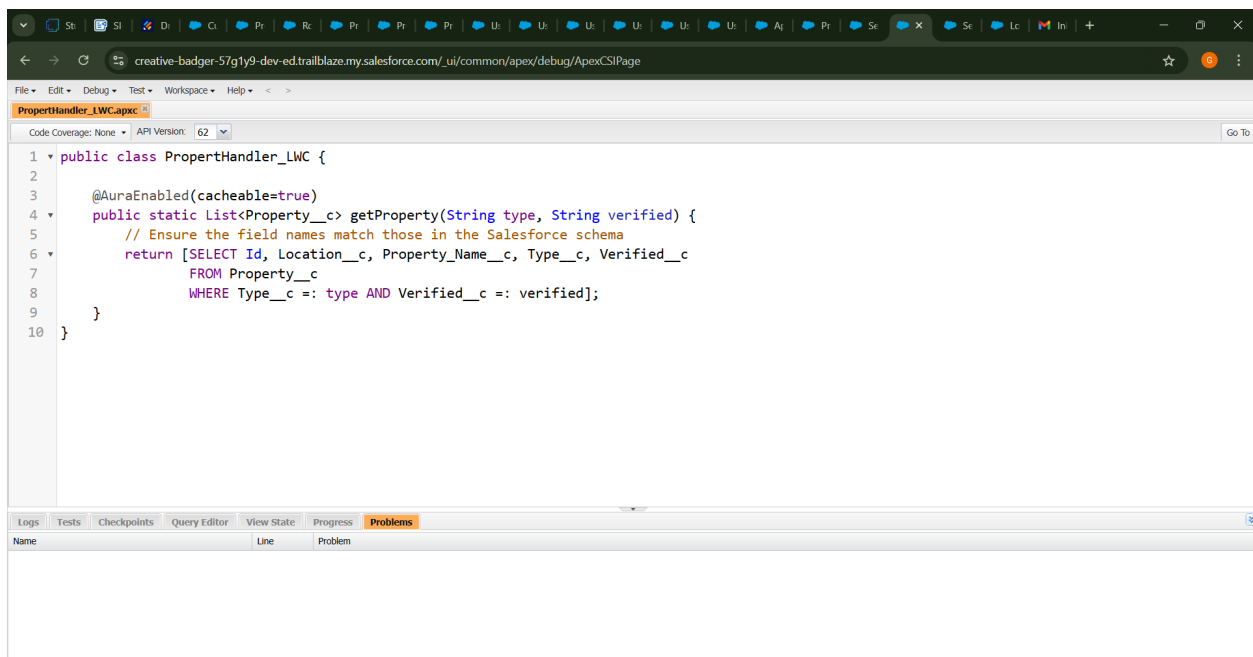
- i. Create an LWC Component for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on “Search your Property Page”

## 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  
        verified = 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 system.
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">
          <bar>
            <lightning-button-icon variant="neutral" icon-name="standard:search" alternate-text="Search"
label="Search" onclick={handleClick}></lightning-button-icon>
          </div>
        </div>
      </div>
    </div>
  </div>
</template>
<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>

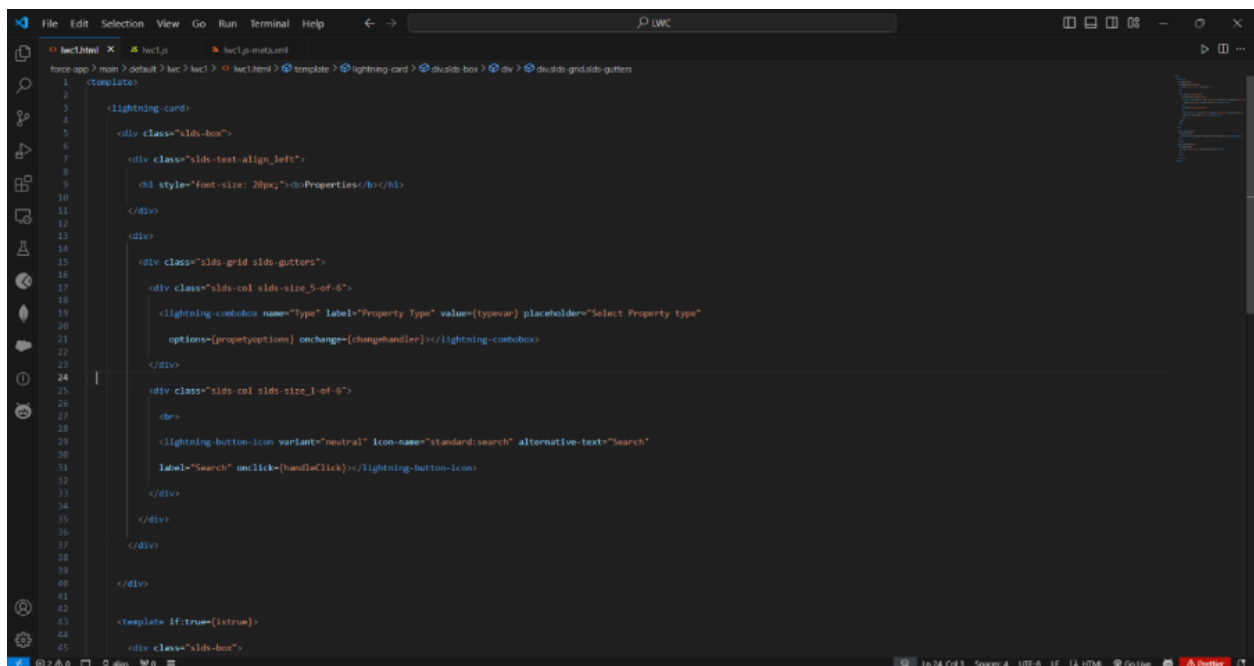
</div>

</template>

</lightning-card>

</templates>

```



1. In Your Js 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 recorded
user = USER_ID; verified

```

```

var type var
is false = true; is true
= false;
@track property list = [];
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" ) }
proportions = [
  ( label: "Commercial", value: "Commercial" }, ( label:
    "Residential", value: "Residential" ),
  ( label: "rental", value: "rental" }

@wire(getRecord, ( recorded: "$user", 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 && != 0) (
        this.isTrue = true; this.propertylist =
        result; console.log(this.verifiedvar);
        console.log(this.typevar)
      ) else (

```

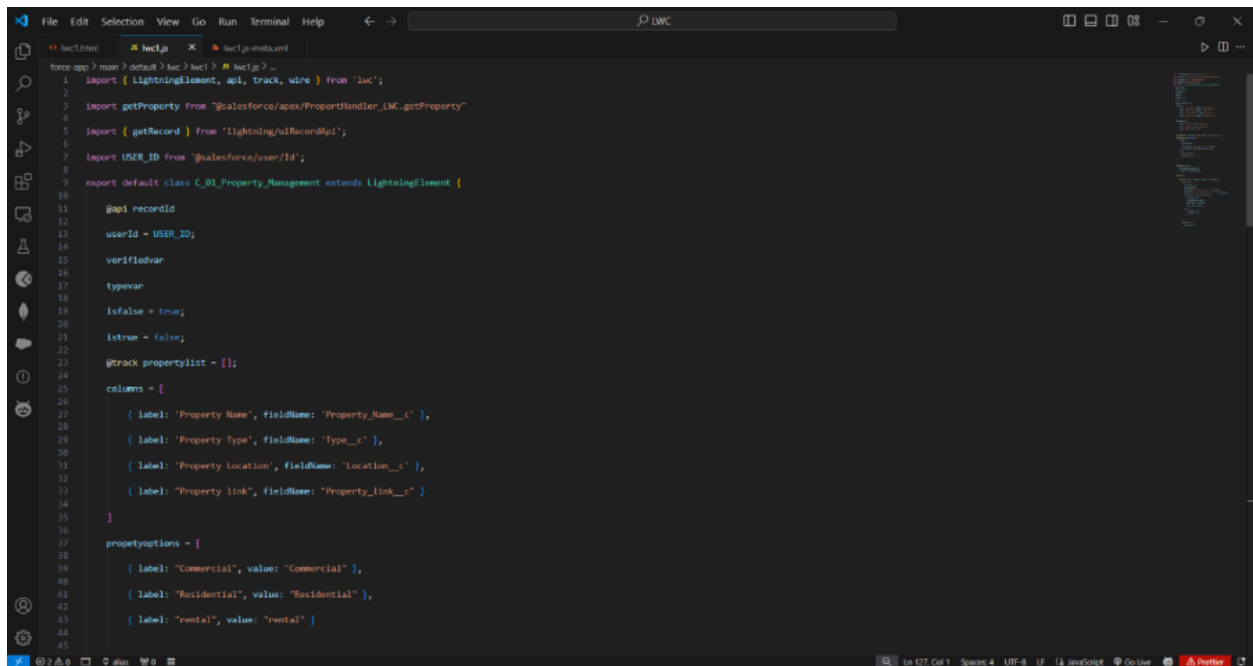
```

        this.isfalse = false;

        this.istrue = false;

    }.catch((error) => (
        console.log(error)
    )

```



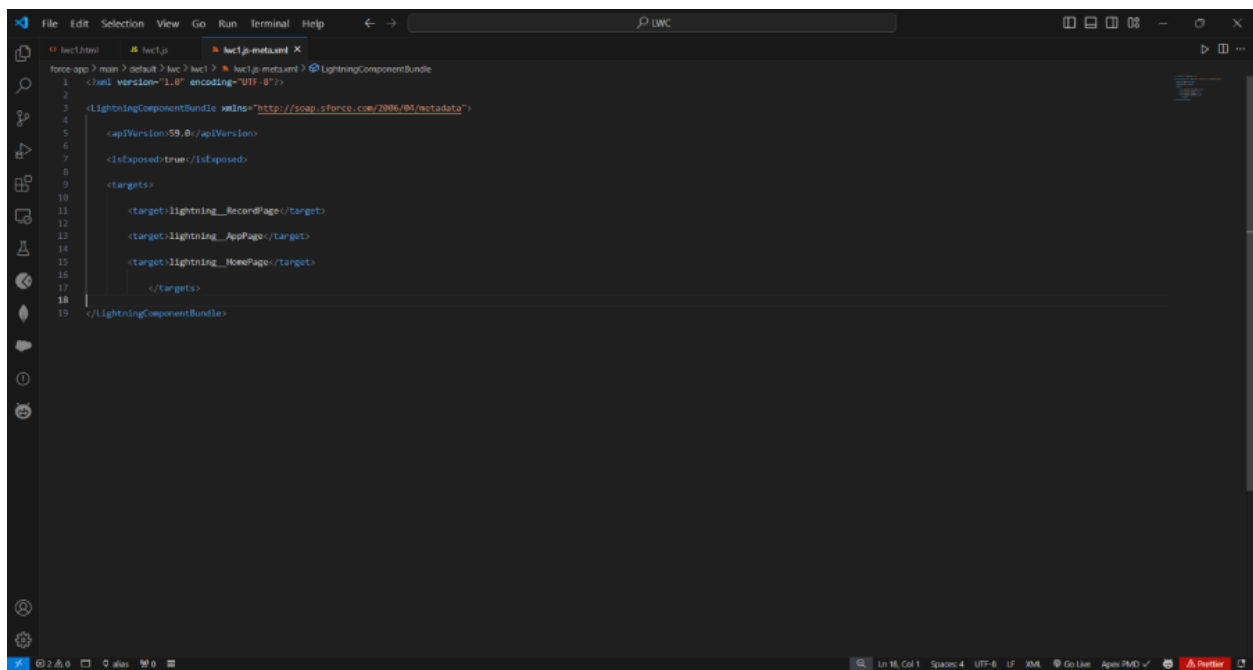
1. In Your metafile, give your targets to deploy the component.

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</target>
    <target>Lightning__AppPage</target>
    <target>Lightning__HomePage</target>
  </targets>
</LightningComponentBundle>

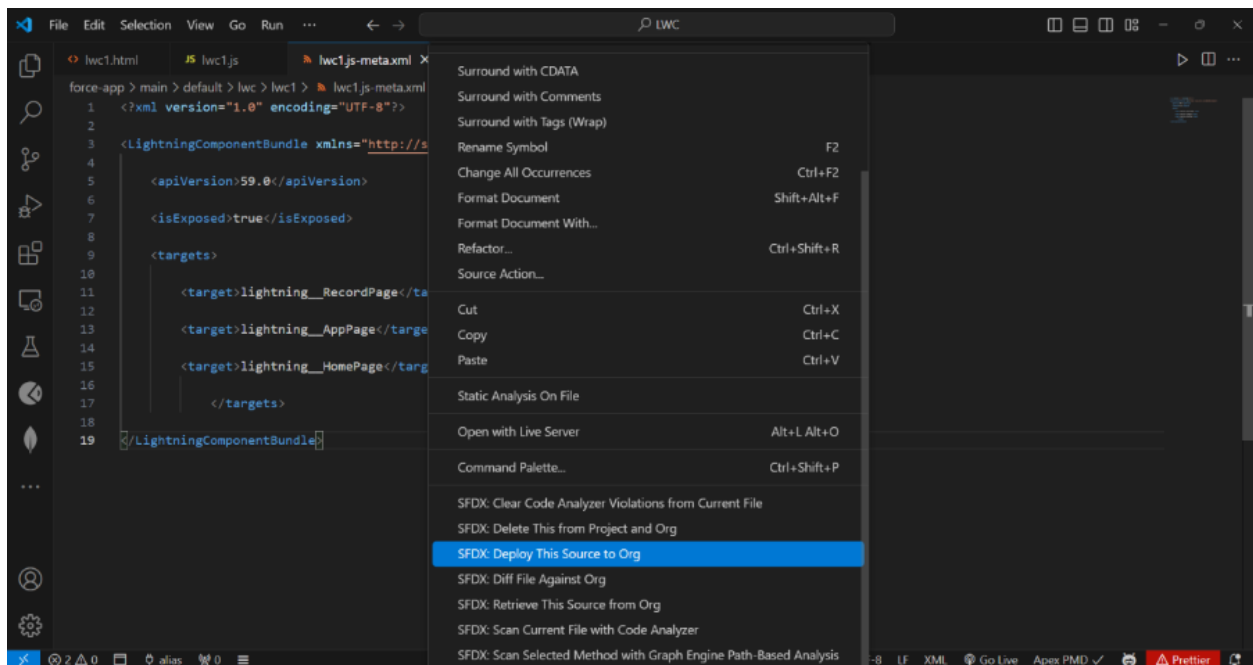
```



The screenshot shows the VS Code editor with the file `lwc1js-meta.xml` open. The file content is as follows:

```
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 organization



The screenshot shows the VS Code editor with the file `lwc1js-meta.xml` open. A right-click context menu is displayed over the file, showing various actions. The action `SFDX: Deploy This Source to Org` is highlighted in blue.

The context menu options are:

- Surround with CDATA
- Surround with Comments
- Surround with Tags (Wrap)
- Rename Symbol (F2)
- Change All Occurrences (Ctrl+F2)
- Format Document (Shift+Alt+F)
- Format Document With...
- Refactor... (Ctrl+Shift+R)
- Source Action...
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Static Analysis On File
- Open with Live Server (Alt+L Alt+O)
- Command Palette... (Ctrl+Shift+P)
- SFDX: Clear Code Analyzer Violations from Current File
- SFDX: Delete This from Project and Org
- SFDX: Deploy This Source to Org**
- SFDX: Diff File Against Org
- SFDX: Retrieve This Source from Org
- SFDX: Scan Current File with Code Analyzer
- SFDX: Scan Selected Method with Graph Engine Path-Based Analysis

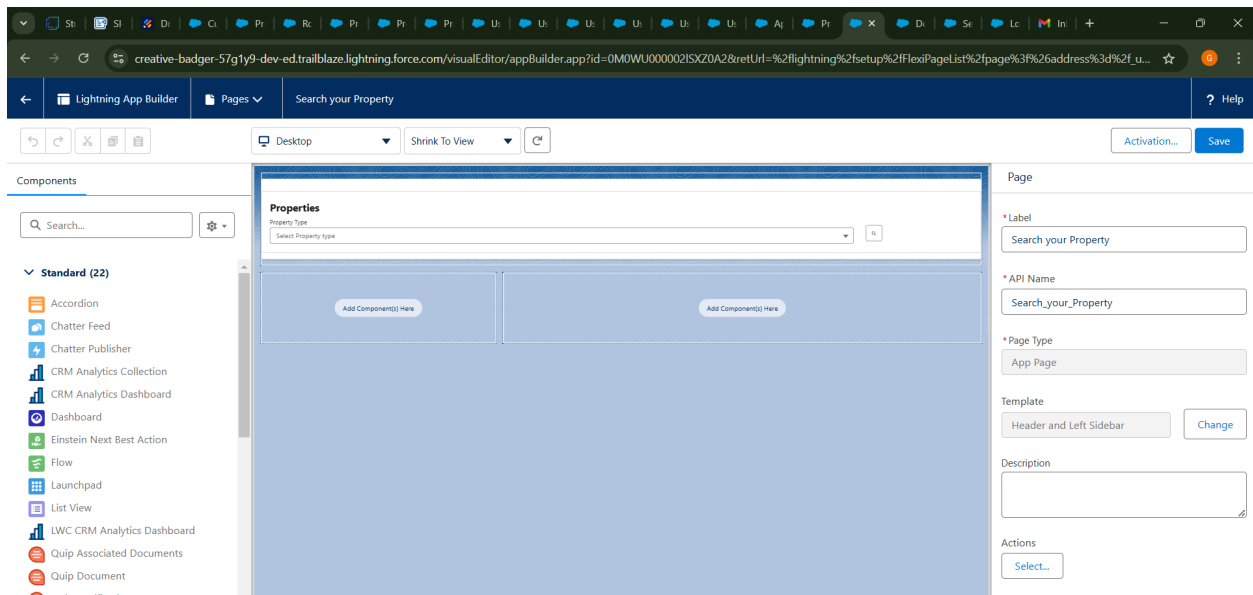


# Drag this Component to your App Page

Adding the Component to your Page

## Activity1

1. From Setup >> Go to App Launcher >> Search for Property Details
2. On this Page, click on the gear icon and click on Edit Page
3. after clicking on the edit page, it will one 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 protection for the profiles that need 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".

