

A CRM Application to Handle the Clients and their property Related Requirements

ProjectDescription:

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 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 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 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.
form link :- <https://form.jotform.com/243259013982055>

The screenshot displays the JotForm 'BUILD' interface for a form titled 'Dreams World'. The form is designed to collect customer information and is structured as follows:

- Name:** A group box containing two text input fields for 'First Name' and 'Last Name'.
- Email:** A single text input field with a placeholder 'example@example.com'.
- Phone Number:** A text input field with a placeholder 'e.g., 23' and a red error message 'Please enter a valid phone number'.
- Property Type:** A section titled 'Which type of Property are you looking for?' with three radio button options: 'RESIDENTIAL', 'COMMERCIAL', and 'RENTAL'.
- Budget Amount:** A text input field with a placeholder 'e.g., 23'.
- Address:** A section containing four text input fields: 'Street Address', 'Street Address Line 2', 'City', and 'State / Province'.
- Postcode:** A text input field with a placeholder 'Postcode / Zip Code'.

A green 'Submit' button is located at the bottom of the form. The interface includes a top navigation bar with 'BUILD', 'SETTINGS', and 'PUBLISH' tabs, and a left sidebar with an 'Add Elements' button. The browser's address bar shows the URL 'jotform.com/build/243259013982055'.

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](#)

| A | B | C | D | E | F | G | H | I | J | K |
|----------|--------------|---------------|-------------|---------------|--------------|----------------|----------------|-----------|-------------|-----------|
| Customer | Phone Number | Emial | State | Property Type | Budget Amoun | 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 displays 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, showing a list of objects. The 'Customer' object is highlighted. Below this, the 'Fields & Relationships' section is expanded, showing a table of fields for the 'Customer' object. The table has columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The fields listed are: Budget Amount (Number(18, 0)), City (Text(255)), Created By (Lookup(User)), Customer (Text(255)), Customer (Name, indexed), Email (Email), Last Modified By (Lookup(User)), Owner (Lookup(User, Group), indexed), and Phone Number (Number(18, 0)).

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

Integrate Forthwith Salesforce Platform

In this Milestone we are going to integrations 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 Org 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 Each and every field on the Object with the fields on the form and “Save Action”.

The screenshot shows the Jotform 'Form Builder' interface for a form named 'Dreams World'. 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 'Create a record' and shows a mapping of form fields to Salesforce fields. The 'Object Fields' are listed on the left, and the 'Dreams World' fields are on the right. The mapping is as follows:

| Object Fields | Dreams World |
|-----------------------|--|
| Customer__c | Name - First Name |
| City | Address - City |
| Budget Amount | Budget Amount |
| Property Type | Which type of Property are you lookin... |
| Phone Number | Phone Number |
| Street Address | Address - Street Address |
| Email | Email |
| Name | Name - Last Name |
| State | Address - State |
| Street Address line 2 | Address - Street Address 2 |

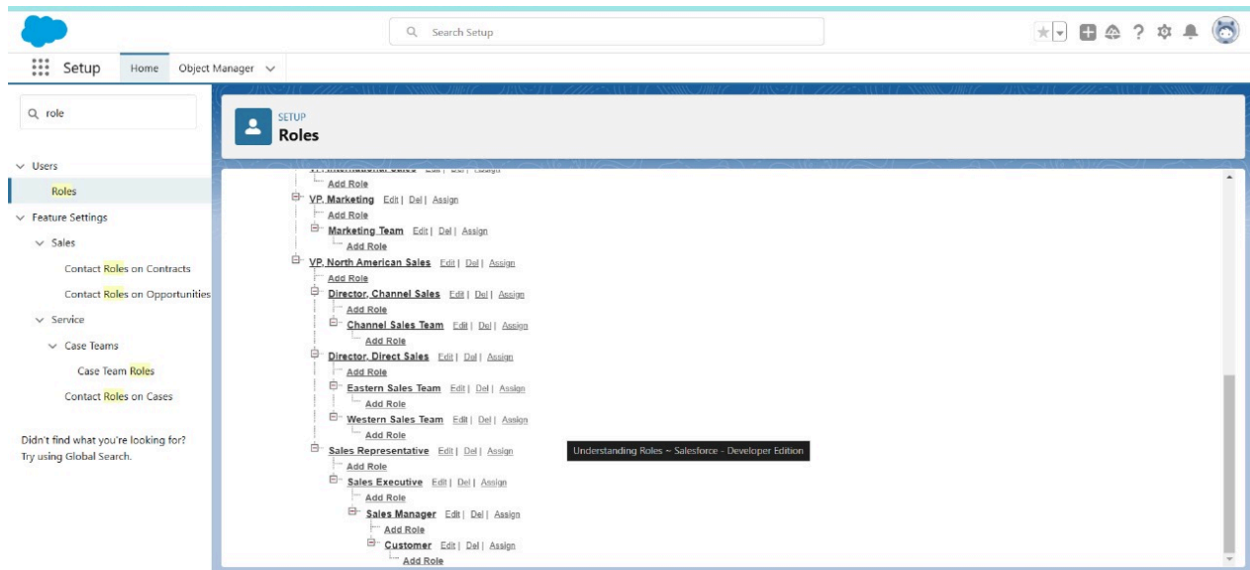
Below the mapping table, there is an 'Add Field' button and an 'Update an existing record' section. The top right of the interface shows 'All changes saved at 10:50 AM' and a 'Preview Form' button.

Then “Save the Integration” and “Finish”.

The screenshot shows the 'SALESFORCE' integration page. It features a header with 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 a '+ Add New Action' button. A list of actions is shown, with the first action being 'Create or update a record' for the 'Customer' object.

here we need to Create Roles as per businessrequirement

Activity:- 1



1. if we don't find sales representative we need to create it according Tothentoughened
2. louse 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>> Got 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.

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 ⓘ



⬆️ Upload

Primary Color Hex Value ⓘ

⌵

#AAE420

Org Theme Options

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

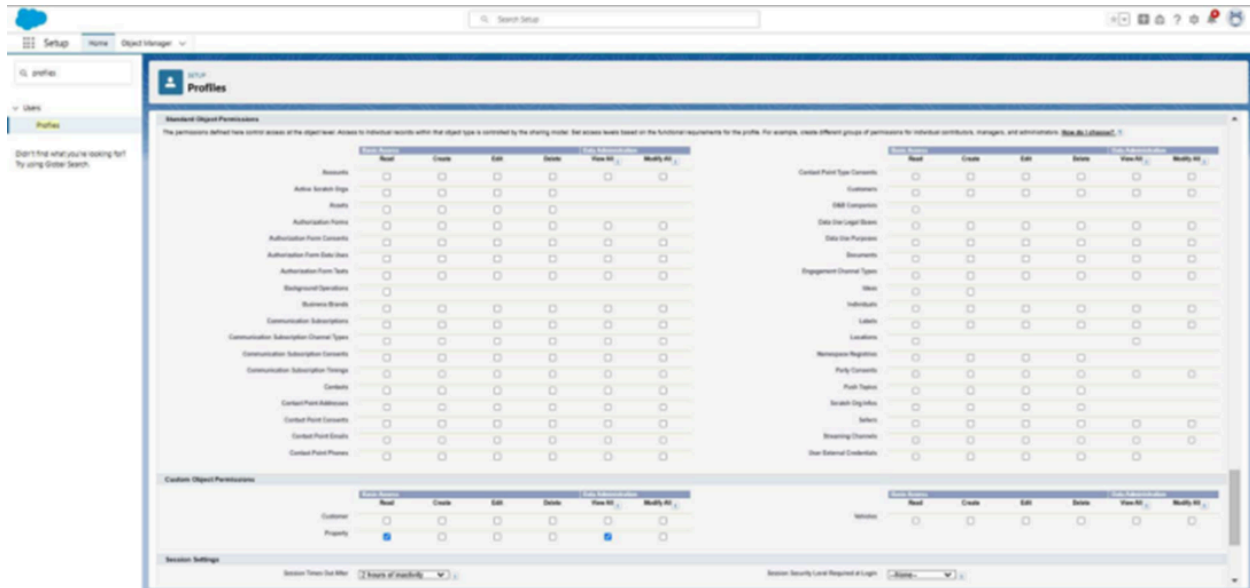
App Launcher Preview

Create Profiles

Create profiles as per business requirement

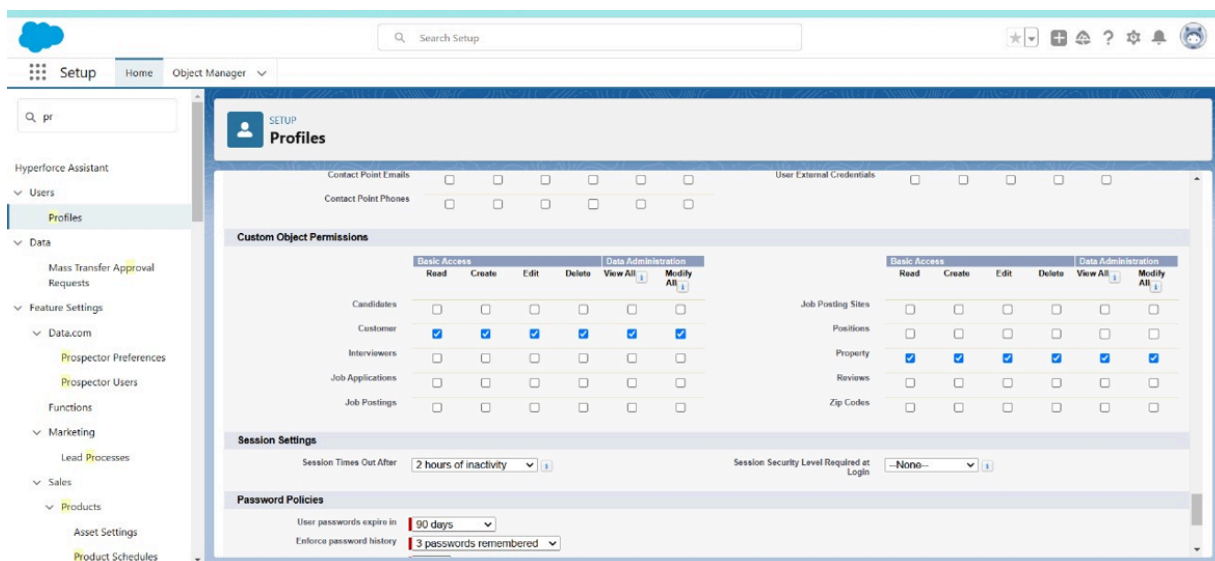
Creating Customer Profiles

1. From Setup? Go to Profiles and Clone (standard platform)Salesforce Platform User and Name it “Customer”..
2. Uncheck all the Custom Objects andCheck onlyProperty Details From Custom App Settings.
3. so RemovealltheStandard Object Permissions
4. Uncheck all the Custom Object Permissions and check read and view all in “Property”
5. make sure every 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”.



Create a CheckBoxfield 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”

The screenshot shows the Salesforce Setup interface. The breadcrumb trail is 'SETUP > OBJECT MANAGER > User'. The left sidebar shows the 'Fields & Relationships' section selected. The main content area displays the 'User Custom Field' configuration for the 'Verified' field. The field is a 'Checkbox' type. The 'Field Information' section shows the field label as 'Verified', field name as 'Verified', API name as 'Verified__c', and data type as 'Checkbox'. The 'General Options' section shows the default value as 'Unchecked'. The 'Validation Rules' section shows 'No validation rules defined'.

| Field Information | |
|---------------------------|---|
| Field Label | Verified |
| Field Name | Verified |
| API Name | Verified__c |
| Description | |
| Help Text | |
| Data Owner | |
| Field Usage | |
| Data Sensitivity Level | |
| Compliance Categorization | |
| Created By | DUDIPALLA RASHMITHA, 22/11/2024, 11:29 am |
| Modified By | DUDIPALLA RASHMITHA, 22/11/2024, 11:29 am |

| General Options | |
|-----------------|-----------|
| Default Value | Unchecked |

| Validation Rules | |
|------------------------------|--|
| 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. Got Setup --> Administration --> Users --> New User
2. Last Name - Executive
3. Role - Sales Executive
4. License - Salesforce
5. Profile - System Administrator
6. Save

User : 2

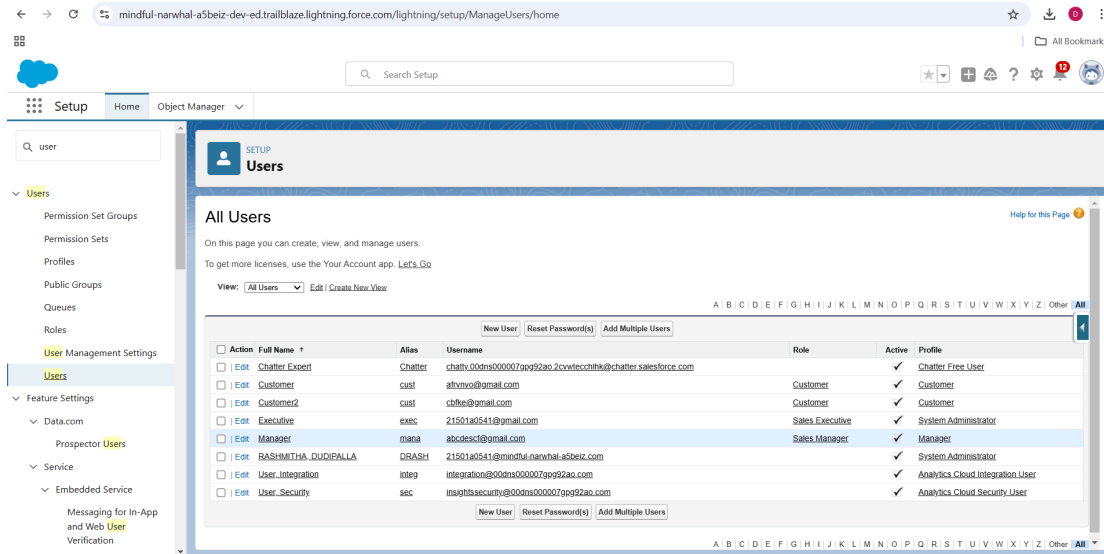
1. Got Setup > Administration>> Users >>New User
2. Last Name>> Manager
3. Role >> Sales Manager
4. License >> Salesforce Platform

User : 3

1. Go to Setup»>Administration »> Users »> New User
2. Last Name » Customer
3. Role >> Customer
4. License»>Salesforce Platform
5. Profile»>Customer
6. Make Sure the verifiedcheckboxis"Unchecked"
7. Save

User : 4

1. Got Setup »> Administration >> Users >> New User
2. Last Name >> Customer2
3. Role >» Customer
4. License >> Salesforce Platform
5. Profile »> Customer
6. Make Sure the verified check box is "checked"
7. Save



Create an 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 ten 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 Edit ability Properties >> Click on Administrator so the currently assigned approver can edit records during the approval process.
9. From Step 5. Select Fields to Display on Approval Page Layout select Property, Owner, Location, Type.

1. Click Next and Select the initial Submitters »
2. Owner >> Property Owner
3. Roles >> Sales Manager
4. Save.

after saving we are directed to approval steps and we need to do as follows Add an approval step name

"ExecutiveApproval "

click next and select the Approver as "Sales Executive" and "Save" AddOne field Update as

"VerifiedProperty"

1. Select Object »Property
2. Field to Update >> Verified
3. Field DataType >»CheckBox
4. Select CheckBox Option as"True"
5. Save.

Add One fieldUpdate as "UnVerified Property"

1. Select Object » Property
2. Field toUpdate >>>Verified
3. Field DataType >»CheckBox
4. Select CheckBox Option as"False"
5. Save.

Activate the Approval Process.

SETUP
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 now to submit the Approval Process Automatically

A flow that can submit the records directly for approval

Activity1

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

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

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 by 'A record is created'. The flow steps are: 'Start' (Record-Triggered Flow), 'Run Immediately', 'Approval for property' (Action), and 'End'. The right-hand pane shows the 'Configure Start' configuration for the flow trigger. Under 'Select Object', 'Property' is selected. Under 'Configure Trigger', the trigger is set to 'A record is created'. Under 'Set Entry Conditions', the condition requirements are set to 'None'.

Flow Builder: Property Approval - V1

Last saved on 18/11/2024, 12:51 pm **Active** Run Debug View Tests Save As New Version Save Deactivate

Select Elements Auto-Layout

Start
Record-Triggered Flow

Object: **Property** Edit

Trigger: **A record is created**

Optimize for: **Actions and Related Records**

+ Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Property

Run Immediately

Approval for property
Action

End

Configure Start

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

* Object
Property

Configure Trigger

* Trigger the Flow When:

☒ A record is created
☐ A record is updated
☐ A record is created or updated
☐ A record is deleted

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements
None

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 Labelas“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 Settingsselectpageactivationas “Activate for all Users”.
7. From Lightning ExperienceClick on “Property Details” and click on Add Page“.
8. Then Clickon“Save”

The image shows two screenshots of the Salesforce Lightning App Builder interface.

The top screenshot is titled "Create a new Lightning page". It features a list of standard page layouts on the left, including "Header and Left Sidebar", "Header and Right Sidebar", "Header and Three Regions", "Header and Two Regions", "Main Region and Right Sidebar", "One Region", "Three Regions", and "Two Regions". The "Header and Left Sidebar" layout is selected. To the right, there are visual representations of how the layout appears on desktop, tablet, and phone. Below these, a description states: "Full-width header above a left sidebar region and a wide main region. On a tablet in portrait orientation, the regions below the header are equal width. On a phone, the regions stack vertically." At the bottom, it says "Supported form factors: desktop, tablet, and phone." There are "Back" and "Done" buttons at the bottom.

The bottom screenshot is titled "Activation: select your property". It has three tabs: "PAGE SETTINGS", "LIGHTNING EXPERIENCE" (which is active), and "MOBILE NAVIGATION". Under the "LIGHTNING EXPERIENCE" tab, there is a section "Add this app page to Lightning Experience apps. You can manage Lightning apps in Setup." Below this, there is a list of "Add to Lightning Apps" with items like "How We Roll Maintenance", "LightningBolt", "LightningInstrumentation", "Property Details" (which is checked with a blue checkmark), "Sales", "Sales Console", and "Salesforce CMS". To the right, there is a "Property Details" section with a "Remove page" button. Below this, there is a list of "Property" items: "Property", "select your property" (with a dropdown arrow), "Search your property", "Customers", and "Customer". At the bottom, there are "Cancel" and "Save" buttons.

Create a LWC Component

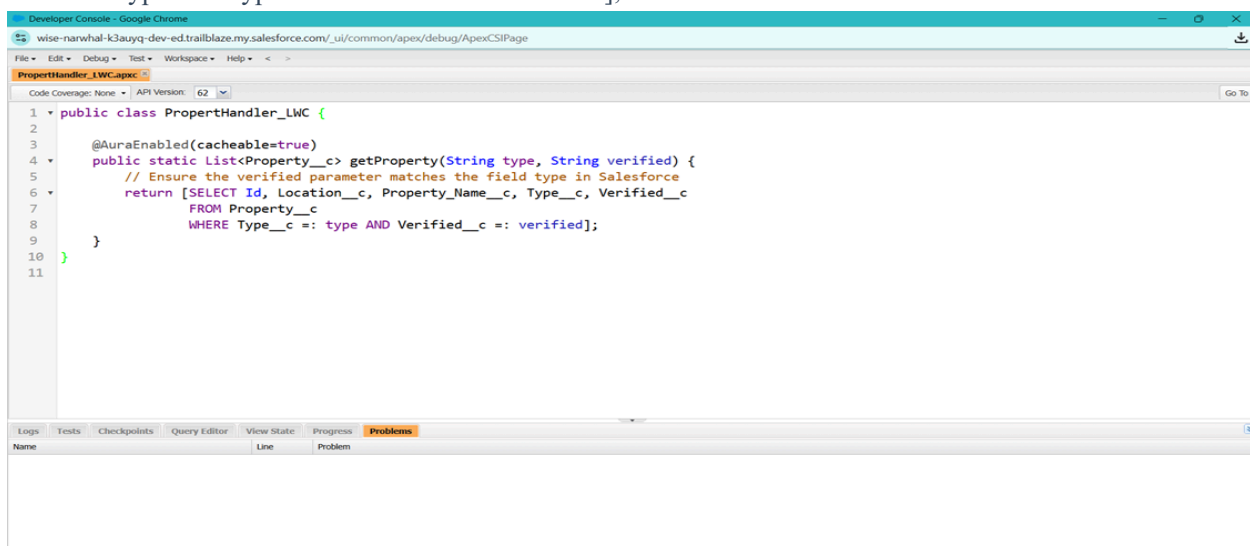
- a. Create an LWCComponent for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on “Search your Property Page”

Activity1

1. Create an ApexClass mandrake aura enabled and name it “PropertHandler_LWC”

Code: -

```
public class PropertHandler_LWC {  
    @AuraEnabled(cacheable=true)  
  
    public static List<Property> getProperty(String type, Boolean verified) {  
        String verifiedStr = verified ? 'true' : 'false'; // Convert boolean to string  
        return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c  
                FROM Property__c  
                WHERE Type__c = :type AND Verified__c = :verifiedStr];  
    }  
}
```



1. Create a Lightning Web Component in your VsCode, and (ctrl+shift +P) and click on authorize an org.
2. Enter your login id and password to authorize your org.
3. Now (ctrl+shift +P) and Create a lightning Web Component and Name it Anything you want to.
(Example -

4. In yourHtml File Write this code :-

Code :-

```
<template>

<lightning-card>

  <div class="slds-box">

    <div class="slds-text-align_left">

      <h1 style="font-size: 20px;"><b>Properties</b></h1>

    </div>

    <div>

      <div class="slds-grid slds-gutters">

        <div class="slds-col slds-size_5-of-6">

          <lightning-combobox name="Type" label="Property Type" value={typevar}
placeholder="Select Property type"
          options={propetyoptions} onchange={changeHandler}></lightning-combobox>

        </div>

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

          <bar>

            <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={isttrue}>

    <div class="slds-box">

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

    </div>

  </template>

<template if:false={isfalse}>
```

```

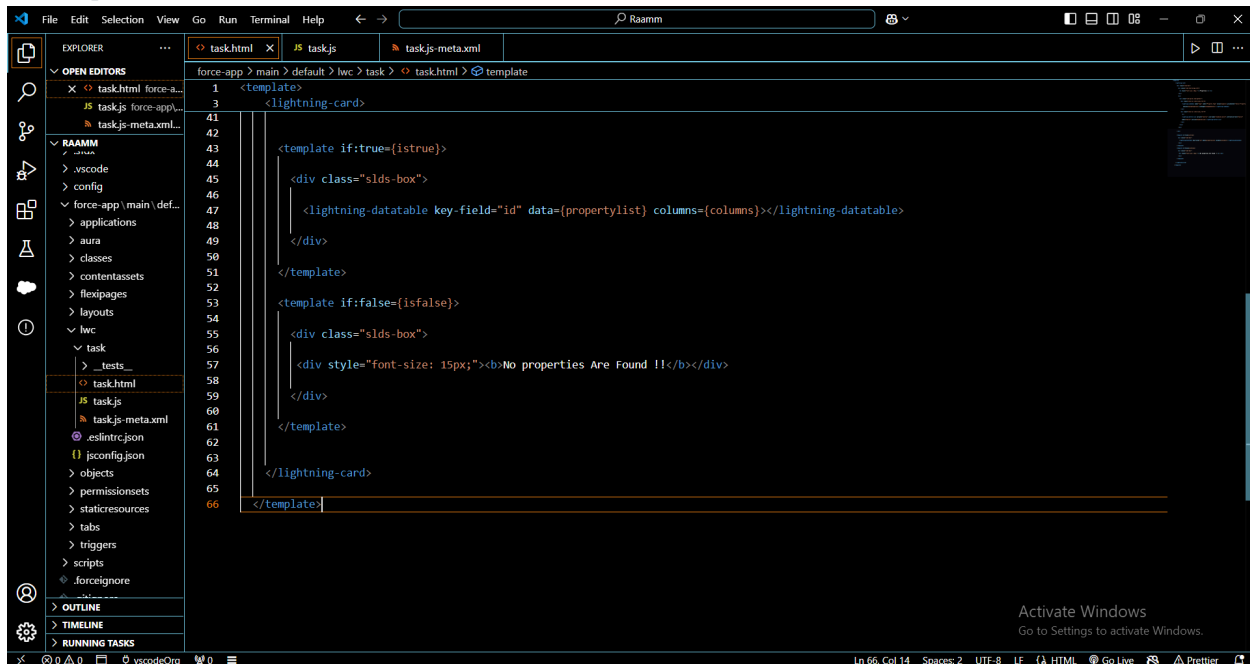
<div class="slds-box">

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

</div>
</template>
</lightning-card>

</templates>

```



1. In YourJs FileWrite 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_M mismanagement LightningElement ( @api recordId

user=recorded

verifiedvartypevar

is false = true; istru

= false;

@track property list = [];

columns = [
  ( label: 'Property Name', fieldName: 'Property_Name c' ), ( label:
    'Property Type', fieldName: 'Type c' ),
  ( label: 'Property Location', fieldName: 'Locationc' ), ( label:

```

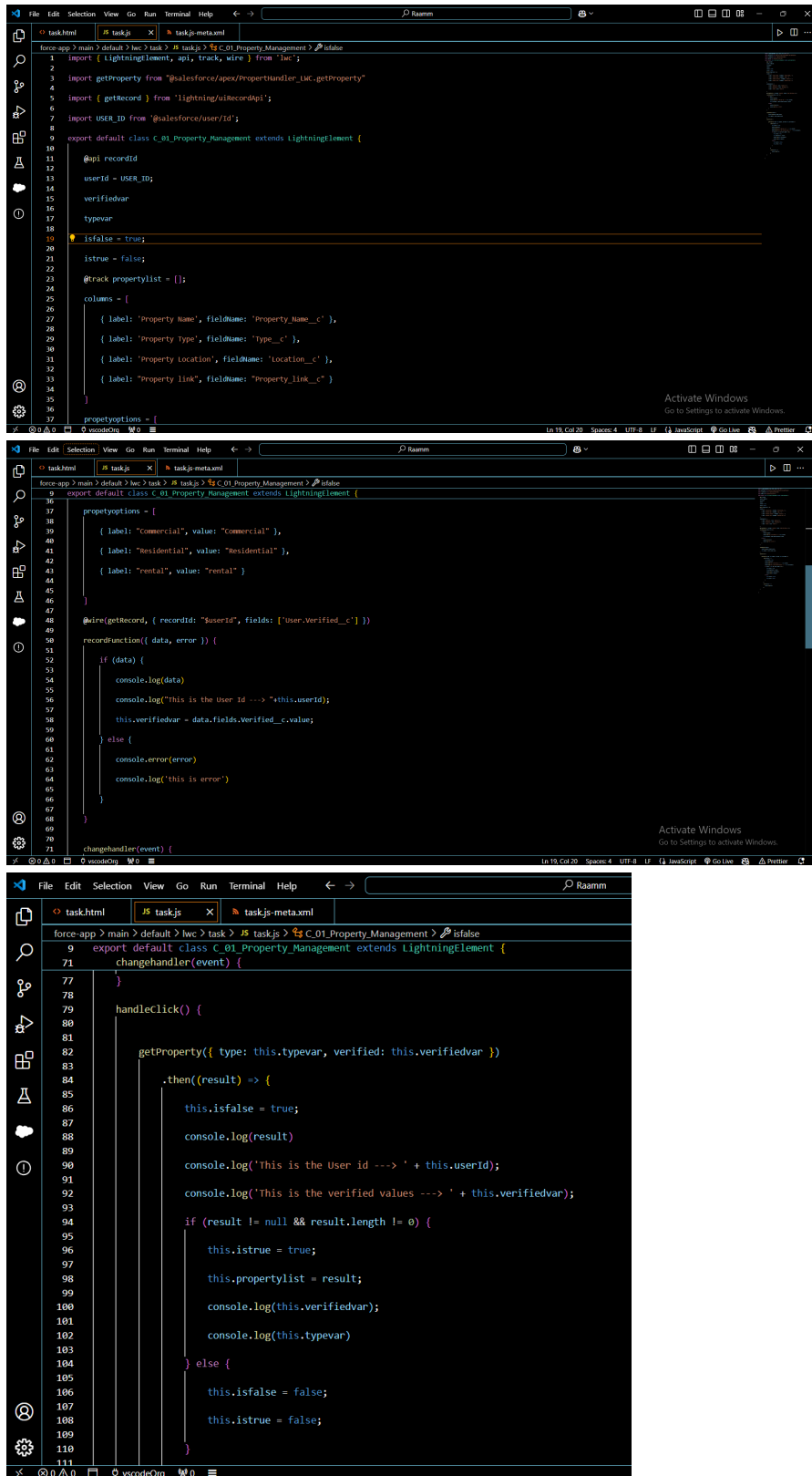


```

        "Property link", fieldName: "Property link c"}
propertyoptions= [
    ( 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 && 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 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>

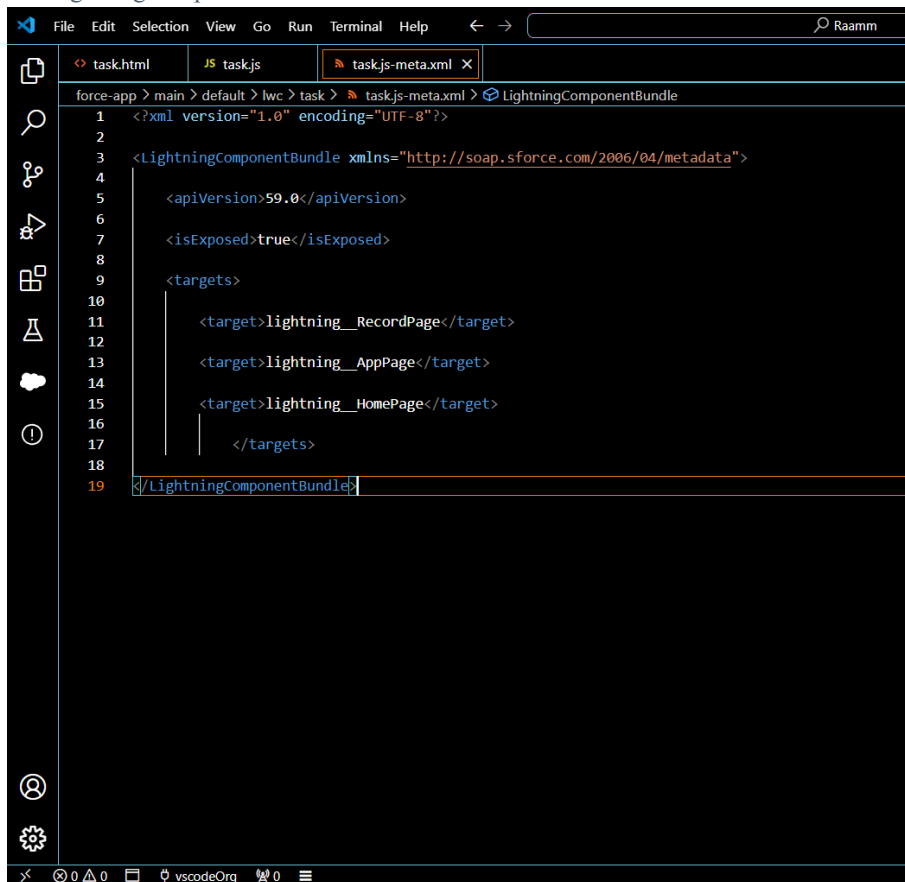
    <target>Lightning__RecordPage</target>

    <target>Lightning__AppPage</target>

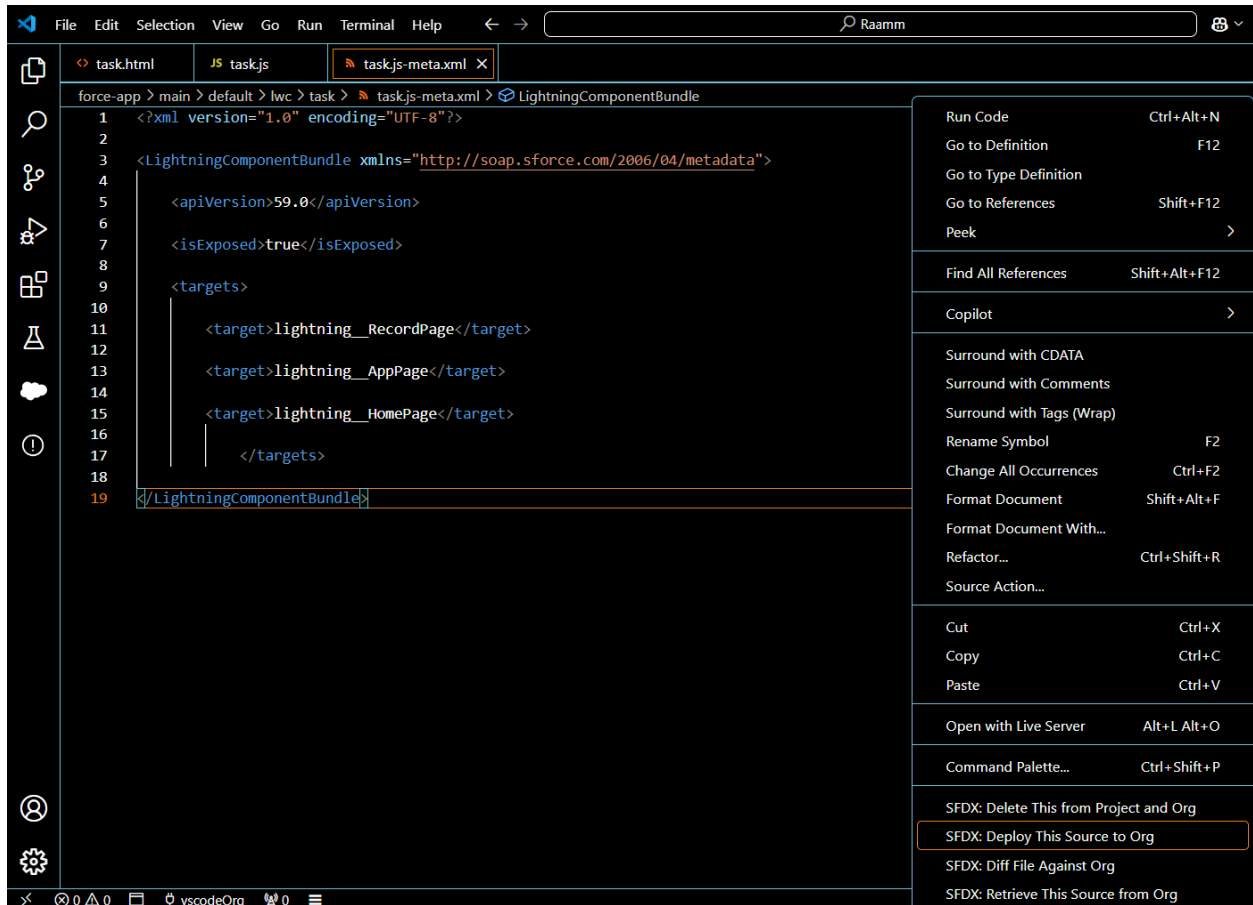
    <target>Lightning__HomePage</target>

  </targets>

</LightningComponentBundle>
```



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

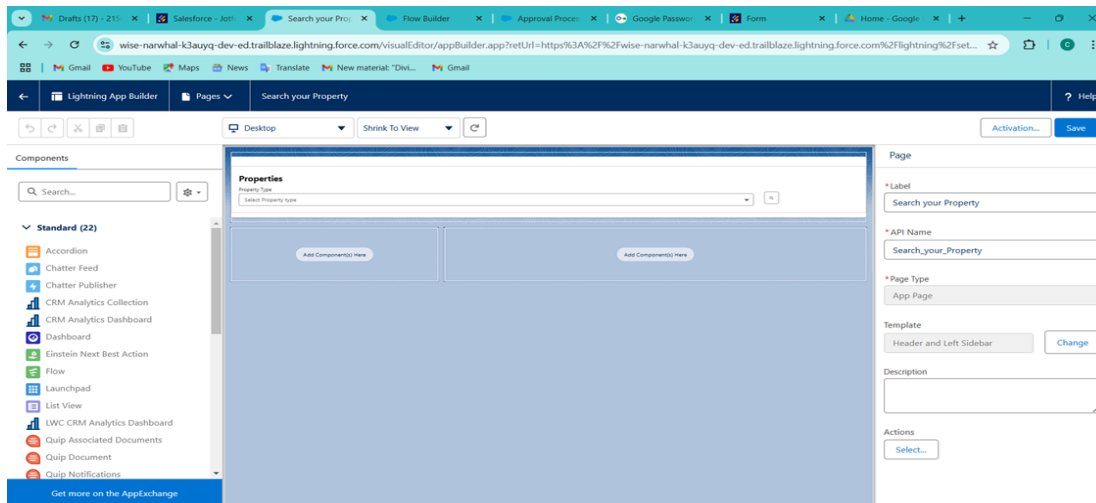


Drag this Component to your App Page

Adding the Component to your Page

Activity 1

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 redirections 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. Activity

From Setup >> Search For Apex Classes >> Click on "Security" behind "PropertyHandlerLWC".

2. From Profiles Add "Manager" and "Customer" and "Save".

