

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

## **Project Overview : -**

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.

## **Project Flow:**

- Milestone 1 : Create a jotform and integrate it with the org to create records automatically
- Milestone 2 : Create Objects from Spreadsheet
- Milestone 3 : Integrate Jotform with Salesforce Platform
- Milestone 4 : Create Roles
- Milestone 5 : Create a Property Details App
- Milestone 6 : Create Profiles
- Milestone 7 : Create a Checkbox Field on User
- Milestone 8 : Create Users
- Milestone 9 : Create an Approval Process for Property Object
- Milestone 10 : Create a Record trigger flow to submit the Approval Process Automatically.
- Milestone 11 : Create an App Page
- Milestone 11 : Create an LWC Component
- Milestone 11 : Drag this Component To Your App Page

## **Requirements : -**

### **1) Website Integration Requirements :**

Implement a form on the website for users to express interest in property listings.

Ensure the form captures essential details such as name, contact information, preferred property type, location, budget, etc. Set up validation rules to ensure data accuracy and completeness.

Integrate the form submission process with Salesforce.

### **2) Salesforce Configuration Requirements :**

Set up Salesforce objects and fields to store customer data. This includes fields for name, contact information, preferences, approval status, etc. Define workflows or processes to automate the creation of records when a user submits the form on the website.

Implement validation rules and data integrity checks to maintain data quality.

Configure Salesforce security

### **3) Approval Process Requirements :**

Define criteria for categorizing users as approved or non-approved based on specific parameters such as budget, property preferences, etc. Implement an approval process in Salesforce to review and approve users. Set up email notifications or alerts to notify relevant stakeholders when a user is approved or rejected.

### **4) User Experience Requirements:**

Design user interfaces in Salesforce for managing customer records, approval processes, and property listings. Ensure a seamless user experience for both customers and internal users interacting with Salesforce. Provide training and documentation for internal staff on how to use the Salesforce system effectively.

## **5) Integration Testing and Quality Assurance Requirements:**

Conduct thorough testing of the integration between the website and Salesforce to ensure data is accurately captured and transferred. Perform end-to-end testing of the approval process to verify that users are categorized correctly and granted appropriate access. Identify and resolve any issues or bugs encountered during testing.

## **6) Documentation and Maintenance Requirements:**

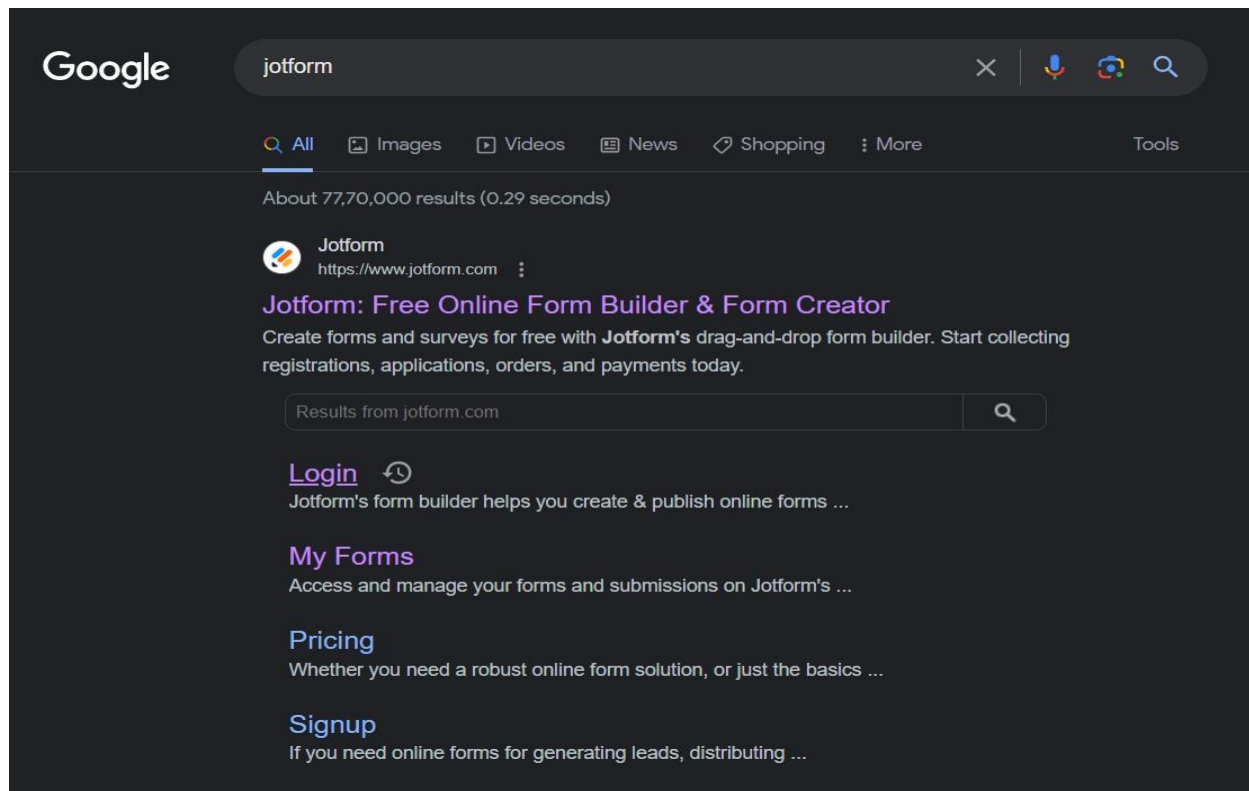
Document the integration architecture, data flows, and configuration details for future reference. Establish procedures for ongoing maintenance and updates to the integration and Salesforce configuration. Provide ongoing support and training for users to address any issues or questions that may arise.

### **\*After Creating the Salesforce org Start with the First Milestone\***

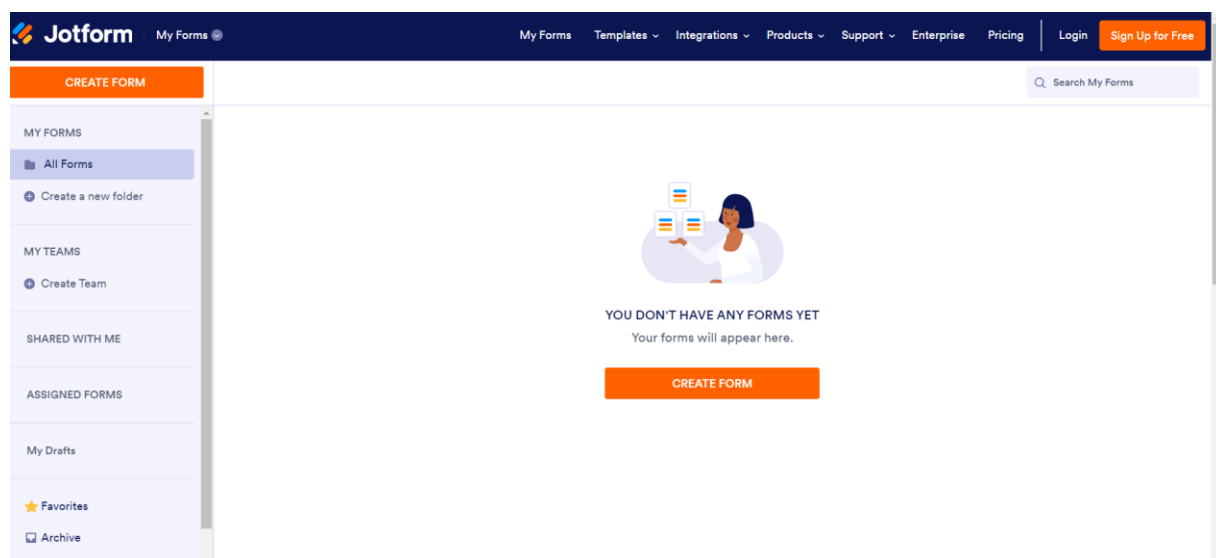
**Milestone 1 :- Create a jotform and integrate it with the org to create a record of customers automatically.**

**USE CASE :** - 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.

- 1) Open your browser and search for jotform and log in.




2) After login click on create form and click on start from scratch



- 3) Now create a form to get the customer details like Name, Phone, Email, Address and type of property the customer is interested in.

Dreams World  
Last edited yesterday 12:00

BUILD SETTINGS PUBLISH



**Dreams World**

Name \*

First Name Last Name

Email

example@example.com

Phone Number

(000) 000-0000

Please enter a valid phone number.

Which type of Property are you looking for?

☐ RESIDENTIAL

☐ COMMERCIAL

☐ RENTAL

Budget Amount \*

e.g., 23

Address

Street Address


Street Address Line 2

City State / Province

Postal / Zip Code

+ ADD NEW PAGE HERE

If you want to remove Jotform Branding, [please upgrade your account](#)

 Jotform

Now create your own Jotforms - It's Free [Create your own Jotform](#)

- 4) Once the form is created, publish it by clicking on publish.

<https://www.jotform.com/form/240031134484041>

## **Milestone 2 :- Create Objects from Spreadsheet.**

- **Create 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.

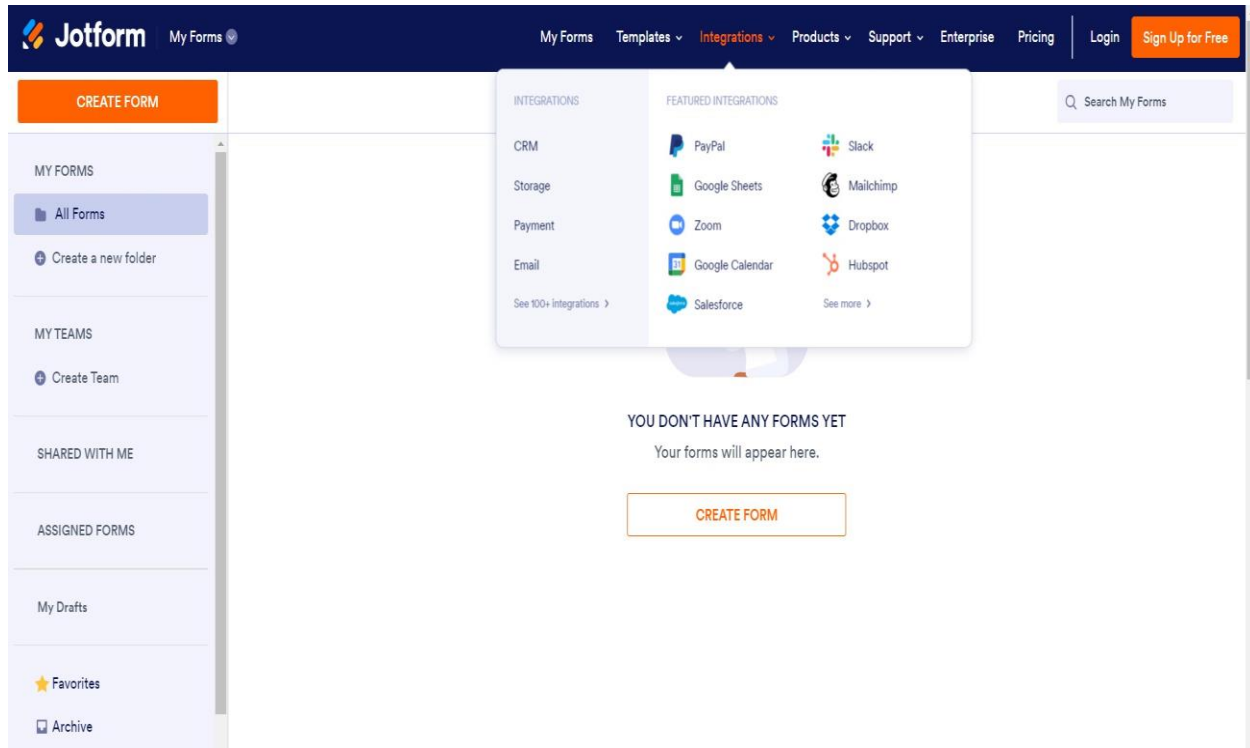
- **Create Property object**

- 1) Follow the same from the customer object to create the Property Object
- 2) [Property](#)

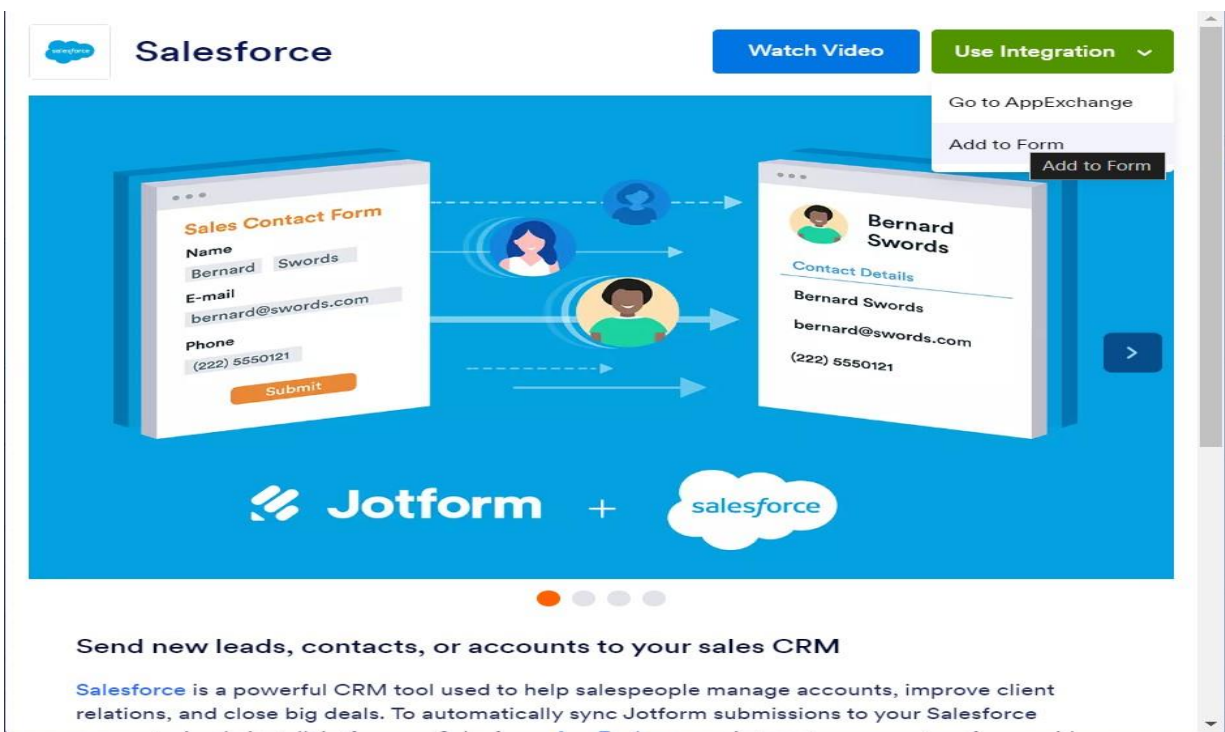
## **Milestone 3 : - Integrate Jotform with Salesforce Platform**

]

- 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.

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), APPROVAL FLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The top navigation bar includes BUILD, SETTINGS, and PUBLISH tabs. The main content area displays the "Salesforce via this handy integration" section. It includes a list of actions: Add new company leads, Add new contacts, Add new accounts, and Connect any custom object. Below this, there is an "Authentication" section with a dropdown menu labeled "Select a Salesforce account" and a text input field containing "dada rao - prajwal@thesmartbridge.com". A "Send Feedback" button is located in the bottom right corner.

4) Select an Action - Create a record.  
Select a Salesforce Object : - Customer

The screenshot shows the Jotform Form Builder interface for the "Create a record" action. The left sidebar is the same as in the previous screenshot. The top navigation bar includes BUILD, SETTINGS, and PUBLISH tabs. The main content area displays the "Create a record" section with two radio buttons: "Create a record" (selected) and "Find existing record". Below this, there is a "Select a Salesforce Object" section with a dropdown menu showing "Customer" and a search bar. Underneath, there is an "Object Fields" section with two dropdown menus labeled "Select field" and a button labeled "+ Add Field". At the bottom, there is an "Update an existing record" section with a checkbox labeled "OFF". A "Send Feedback" button is located in the bottom right corner.



5) Map Each and every field on the Object with the fields on the form and “Save Action”.

The screenshot shows a mapping interface titled "Create a record" with the subtitle "Send data from form fields to matched Salesforce fields". It features two columns: "Object Fields" (Salesforce) and "Dreams World" (form fields). The mappings are as follows:

Object Fields	Dreams World
Customer	Name - First Name
City	Address - City
Budget Amount	Budget Amount
Property Type	Which type of Property ar...
Phone Number	Phone Number
Street Address	Address - Street Address
Email	Email
Customer Name	Name - Last Name
State	Address - State
Street Address line 2	Address - Street Address 2

At the bottom, there is a "+ Add Field" button.

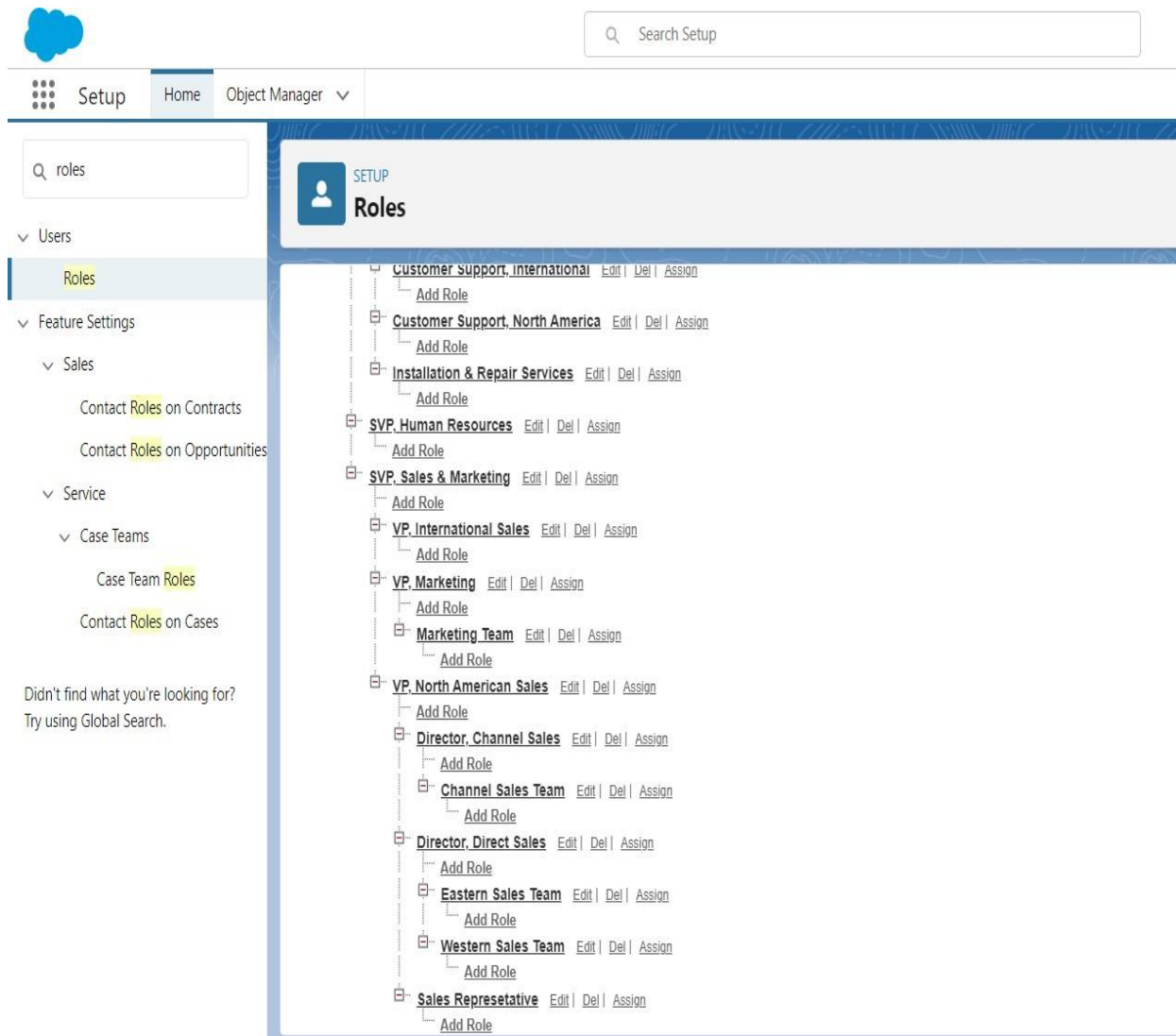
6) Then “Save the Integration” and “Finish”.

The screenshot shows the "SALESFORCE" integration summary screen. It includes a back arrow, 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 a record" for the "Customer" object. At the bottom, there are two buttons: "CANCEL" and "SAVE INTEGRATION".

## Milestone 4 : Create Roles

- **Sales Executive Role**

1) Go to Setup and Click on Roles, then click on Expand all and Add a Role just below the Sales Representative



The screenshot shows the Salesforce Setup interface. At the top, there is a search bar labeled "Search Setup". Below the navigation bar, the "Roles" section is selected. The left sidebar shows a tree view with "Roles" highlighted under "Feature Settings". The main content area displays a list of roles with expand/collapse icons, role names, and action links (Edit, Del, Assign). The roles are organized hierarchically, with "Sales Representative" at the bottom. Below the list, there is a message: "Didn't find what you're looking for? Try using Global Search."

Role Name	Actions
Customer Support, International	Edit   Del   Assign
Add Role	
Customer Support, North America	Edit   Del   Assign
Add Role	
Installation & Repair Services	Edit   Del   Assign
Add Role	
SVP, Human Resources	Edit   Del   Assign
Add Role	
SVP, Sales & Marketing	Edit   Del   Assign
Add Role	
VP, International Sales	Edit   Del   Assign
Add Role	
VP, Marketing	Edit   Del   Assign
Add Role	
Marketing Team	Edit   Del   Assign
Add Role	
VP, North American Sales	Edit   Del   Assign
Add Role	
Director, Channel Sales	Edit   Del   Assign
Add Role	
Channel Sales Team	Edit   Del   Assign
Add Role	
Director, Direct Sales	Edit   Del   Assign
Add Role	
Eastern Sales Team	Edit   Del   Assign
Add Role	
Western Sales Team	Edit   Del   Assign
Add Role	
Sales Representative	Edit   Del   Assign
Add Role	

\* It will use the “System Administrator Profile”.

- 2) Label - Sales Executive  
Reports to - Sales Representative

The screenshot shows a web-based interface for setting up roles. At the top, there is a header bar with a blue icon of a person and the text 'SETUP Roles'. Below this, the main content area is titled 'Role Edit' and 'New Role'. The form itself is titled 'Role Edit' and contains four input fields with labels to their left: 'Label' with the value 'Sales Executive', 'Role Name' with the value 'Sales Executive' and an information icon, 'This role reports to' with the value 'Sales Representative' and a search icon, and 'Role Name as displayed on reports' which is currently empty. At the bottom right of the form, there are three buttons: 'Save', 'Save & New', and 'Cancel'.

Role Edit	
Label	<input type="text" value="Sales Executive"/>
Role Name	<input type="text" value="Sales Executive"/> ⓘ
This role reports to	<input type="text" value="Sales Representative"/> 🔍
Role Name as displayed on reports	<input type="text"/>

Save Save & New Cancel

- *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.*

## Milestone 5 : - Create a Property Details App

- 1) From Setup —> Go to App Manager and click on New Lightning App and Name it as “Property Details” and add “Customer” and “Property” Object.

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 ⓘ  
Search Your Property

\* Developer Name ⓘ  
Search\_Your\_Property

Description ⓘ  
Enter a 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

Next

- 2) Click Next → Next → Save and Add “System Admin ”Profile.

## Milestone 6 : - Create Profiles

- **Customer : -**

- 1) From Setup → Go to Profiles and Clone Salesforce Platform User and Name it “Customer”..



### Standard Object Permissions

The permissions defined here control access at the object level. 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. [How do I choose?](#)

[illegible]

- 4) Uncheck all the Custom Object Permissions and check read and view all in “Property”

Communication Subscriptions							Individuals						
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			<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"/>		Locations					
	<input type="checkbox"/>	<input type="checkbox"/>	<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"/>		Party Consents					
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<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"/>		Push Topics					
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<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"/>		Sellers					
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<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"/>		Streaming Channels					
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<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"/>		User External Credentials					
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<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"/>							

Custom Object Permissions

	Basic Access				Data Administration			Basic Access				Data Administration		
	Read	Create	Edit	Delete	View All	Modify All		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"/>
Customer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Sales orders	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
error logs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Vehicles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Property	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>								

- **Manager : -**

- 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.

SETUP  
Profiles

Set the permissions and page layouts for this profile.

**Profile Edit** [Save] [Save & New] [Cancel]

Name: Manager  
User License: Salesforce Platform  
Description: [Text Area]  
Custom Profile: ☒

**Custom App Settings** ⓘ = Required Information

	Visible	Default		Visible	Default
Analytics Studio (standard__Insights)	<input type="checkbox"/>	<input type="radio"/>	LWC Component (LWC_Component)	<input type="checkbox"/>	<input type="radio"/>
App Launcher (standard__AppLauncher)	<input type="checkbox"/>	<input type="radio"/>	Platform (standard__Platform)	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
Dreams Houses (Dreams_Houses)	<input type="checkbox"/>	<input type="radio"/>	Property Details (Property_Details)	<input checked="" type="checkbox"/>	<input type="radio"/>
How We Roll Maintenance (How_We_Roll_Maintenance)	<input type="checkbox"/>	<input type="radio"/>	WDC (standard__Work)	<input type="checkbox"/>	<input type="radio"/>

- 3) Also Remove all the Standard Object Permissions.
- 4) Uncheck all the Custom Object Permissions and check only “modify all” from “Property” and “Customer”

SETUP

Profiles

Custom Object Permissions

	Basic Access				Data Administration	
	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"/>
error logs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input 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"/>

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Sales orders	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vehicles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Session Settings

Session Times Out After

2 hours of inactivity

Session Security Level Required at Login

--None--

Password Policies

User passwords expire in

90 days

Enforce password history

3 passwords remembered

Minimum password length

8

Password complexity requirement

Must include alpha and numeric characters

Password question requirement

Cannot contain password

Maximum invalid login attempts

10

Lockout effective period

15 minutes

Obscure secret answer for password resets

☐

Require a minimum 1 day password lifetime

☐

Don't immediately expire links in forgot

☐

## Milestone 7 : - Create a CheckBox field on user

- 1) Setup → Object Manager → Search for User → Fields and Relationships
- 2) Create new Field Named as “Verified” as Data type “CheckBox”

Search Setup

Setup

Home

Object Manager

SETUP > OBJECT MANAGER

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

Flow Triggers

Validation Rules

User Custom Field

Verified

Back to User Fields

Validation Rules

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	data_rag	Modified By	data_rag
	31/01/2024, 11:22 am		31/01/2024, 11:22 am

General Options

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

Validation Rules

No validation rules defined

Back To Top

Always show me more records per related list



## **Milestone 8 : - Create Users**

Create three different users with three different Roles and profiles as we have mentioned above.

### **User 1 : -**

- 1) Go to Setup → Administration → Users → New User
- 2) LastName - 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) LastName - 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) LastName - Customer
- 3) Role - Customer
- 4) License - Salesforce Platform
- 5) Profile - Customer
- 6) Make Sure the verified check box is “Unchecked”
- 7) Save

#### **User 4 : -**

- 1) Go to Setup → Administration → Users → New User
- 2) LastName - Customer2
- 3) Role - Customer
- 4) License - Salesforce Platform
- 5) Profile - Customer
- 6) Make Sure the verified check box is “checked”
- 7) Save

#### **Milestone 9 :- Create an Approval Process for Property Object**

- 1) From Setup → Process Automation → Approval Process

## 2) Process Name - Property Approval

SETUP Approval Processes

Approval Process Edit  
Property Approval

Step 1 of 6

Enter a name and description for your new approval process.

Enter Name and Description

Process Name: Property Approval

Unique Name: Property\_Approval

Description:

Save Next Cancel

## 3) Give 2 criteria →

- a) Location is not equal to blank,
- b) Verified Equals false.

SETUP Approval Processes

Approval Process Edit  
Property Approval

Step 2 of 6

If only certain types of records should enter this approval process, enter that criteria below. For example, only expense reports from employees at headquarters should use this approval process.

Specify Entry Criteria

Use this approval process if the following criteria are met:

Field	Operator	Value	
Property: Location	not equal to	blank	AND
Property: Verified	equals	False	AND
--None--	--None--		AND
--None--	--None--		AND
--None--	--None--		

Add Filter Logic...

Previous Save Next Cancel

## 4) Click next and “Next Automated Approver Determined By” → Select Manager

5) From Record Editability Properties → Click on Administrators **OR** the currently assigned approver can edit records during the approval process.

**SETUP Approval Processes**

Approval Process Edit  
Property Approval Help for this Page

**Step 3. Specify Approver Field and Record Editability Properties** Step 3 of 6

Previous Save Next Cancel

When you define approval steps, you can assign approval requests to different users. One of your options is to use a user field to automatically route these requests. If you want to use this option for any of your approval steps, select a field from the picklist below. Also, when a record is in the approval process, it will always be locked-- only an administrator will be able to edit it. However, you may choose to also allow the currently assigned approver to edit the record.

**Select Field Used for Automated Approval Routing**

Next Automated Approver Determined By: Manager

Use Approver Field of Property Owner: ☐

**Record Editability Properties**

☐ Administrators **ONLY** can edit records during the approval process.

☒ Administrators **OR** the currently assigned approver can edit records during the approval process.

Previous Save Next Cancel

6) From Step 5. Select Fields to Display on Approval Page Layout select Property, Owner, Location, Type.

**SETUP Approval Processes**

Property Approval

**Step 5. Select Fields to Display on Approval Page Layout** Step 5 of 6

Previous Save Next Cancel

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 <input type="button" value="Add"/> <input type="button" value="Remove"/>	Property
Last Modified By		Owner
Property link		Location
Verified		Property Name
		Type

[Click here to view an example](#)

7) Click Next and Select the initial Submitters →

a) Owner → Property Owner

b) Roles → Sales Manager

8) Save.

## 9) Add an approval step name “Executive Approval ”

Approval Step Edit  
VP Approval [Help for this Page](#)

Step 1. Enter Name and Description Step 1 of 3

Save Next Cancel

Enter a name, description, and step number for your new approval step.

**Enter Name and Description** ! = Required Information

Approval Process Name	Property Approval
Name	VP Approval
Unique Name	VP_Approval
Description	

Save Next Cancel

## 10) specify the Criteria → All record should enter

Approval Step Edit  
VP Approval [Help for this Page](#)

Step 2. Specify Step Criteria Step 2 of 3

Previous Save Next Cancel

Specify whether a record must meet certain criteria before entering this approval step. If these criteria are not met, the approval process can skip to the next step, if one exists. [Learn more](#)

**Specify Step Criteria**

☒ All records should enter this step.

☐ Enter this step if the following  , else .

Previous Save Next Cancel

## 11) click next and select the Approver as “ Sales Executive “ and “Save”

Approval Step Edit  
VP Approval Help for this Page ?

---

**Step 3. Select Assigned Approver** Step 3 of 3

[Previous](#) [Save](#) [Cancel](#)

Specify the user who should approve records that enter this step. Optionally, choose whether the approver's delegate is also allowed to approve these requests.

**Select Approver**

☐ Let the submitter choose the approver manually.  
☐ Automatically assign using the user field selected earlier. **(Manager)**  
☐ Automatically assign to queue.    
☒ Automatically assign to approver(s).

[Add Row](#) [Remove Row](#)

**When multiple approvers are selected:**

☒ Approve or reject based on the **FIRST** response.  
☐ Require **UNANIMOUS** approval from all selected approvers.

☐ The approver's delegate may also approve this request.

[Previous](#) [Save](#) [Cancel](#)

## 12) Add One field Update as “Verified Property”

- a) Select Object → Property
- b) Field to Update → Verified
- c) Field Data Type → CheckBox
- d) Select CheckBox Option as “True”
- e) Save.

**SETUP**  
**Field Updates**

---

Edit Field Update  
**Verified Property** Help for this Page ?

Define the field update, including the object associated with the workflow rule, approval process, or entitlement process, the field to update, and the value to apply. Note that the field to update may be on a related object. Fields are shown only for the type that you select.

**Field Update Edit** [Save](#) [Save & New](#) [Cancel](#)

**Identification** ! = Required information

Name   
 Unique Name    
 Description   
 Object   
 Field to Update   
 Field Data Type   
 Re-evaluate Workflow Rules after Field Change ☐

**Specify New Field Value**

**Checkbox Options**

☒ True  
☐ False

[Save](#) [Save & New](#) [Cancel](#)

### 13) Add One field Update as “UnVerified Property”

- a) Select Object → Property
- b) Field to Update → Verified
- c) Field Data Type → CheckBox
- d) Select CheckBox Option as “False”
- e) Save.

**Field Updates**

**Edit Field Update**  
Unverified Property

Define the field update, including the object associated with the workflow rule, approval process, or entitlement process, the field to update, and the value to apply. Note that the field to update may be on a related object. Fields are shown only for the type that you select.

**Field Update Edit**

**Identification**

Name: Unverified Property  
Unique Name: Unverified\_Property  
Description:   
Object: Property  
Field to Update: Property: Verified  
Field Data Type: Checkbox  
Re-evaluate Workflow Rules after Field Change: ☐

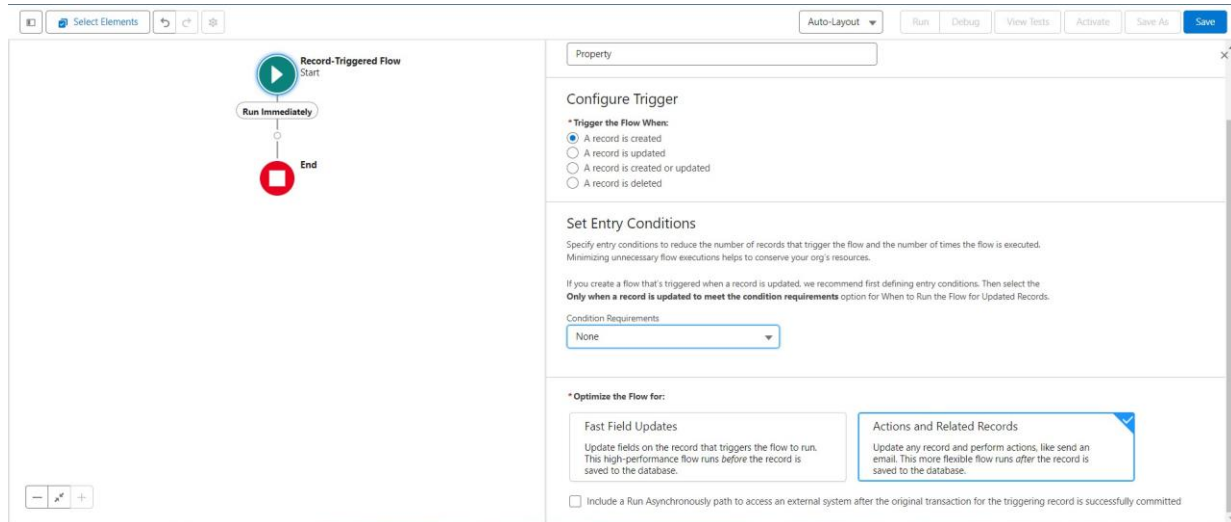
**Specify New Field Value**

Checkbox Options  
☐ True  
☒ False

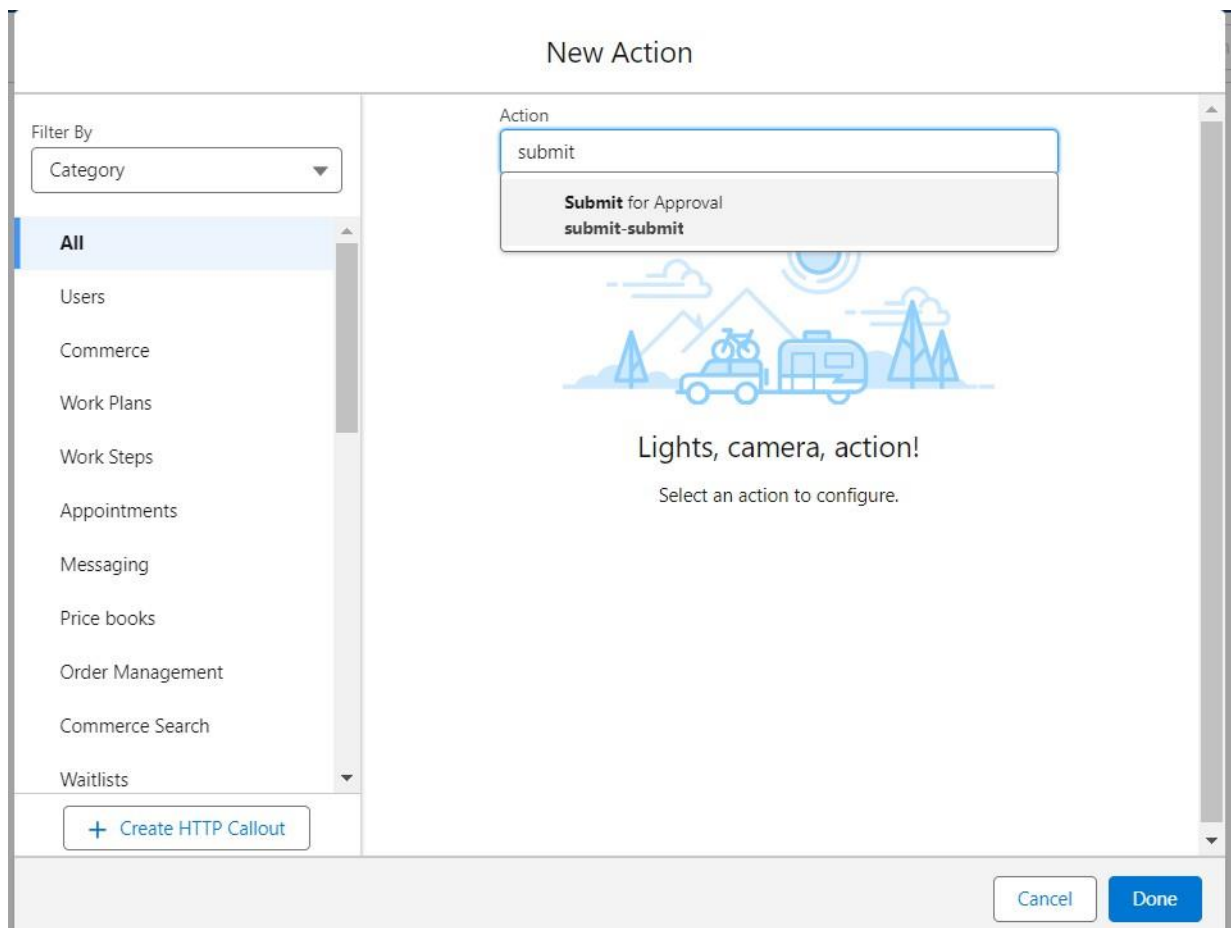
### 14) Activate the Approval Process.

## Milestone 10 : - Create a Record trigger flow to submit the Approval Process Automatically.

- 1) From Setup → Search for Flows → Click On New and Select “Record Trigger Flow”.
- 2) Select Object → Property
- 3) Select “Trigger the flow when” → “A record is created”
- 4) Set Entry Conditions → “None”



5) Add a “Action” → “Submit for Approval”





6) Give Label → Approval for property

7) Record Id → {!!\$Record.Id}

8) Done

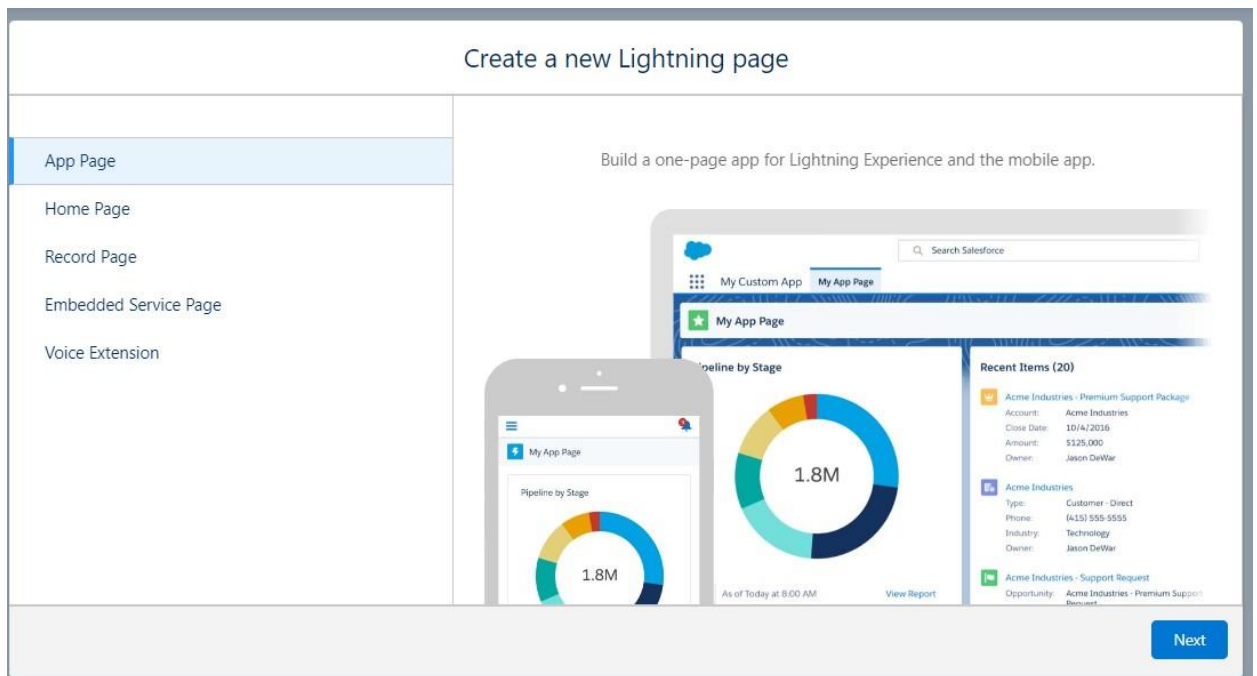
The screenshot shows a 'New Action' configuration window. On the left is a sidebar with a 'Filter By' dropdown set to 'Category'. Below it is a list of categories: All, Users, Commerce, Work Plans, Work Steps, Appointments, Messaging, Price books, Order Management, Commerce Search, and Waitlists. At the bottom of the sidebar is a button '+ Create HTTP Callout'. The main area of the window is titled 'Action' and contains a text input field with the value 'Submit for Approval'. Below this is a paragraph of instructional text: '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.' This is followed by two input fields: '\* Label' with the value 'Approval for property' and '\* API Name' with the value 'Approval\_for\_property'. Below these is a 'Description' text area. Further down is a section titled 'Set Input Values for the Selected Action' containing three items: 'Record ID' with a value of {!!\$Record.Id}, 'Approval Process Name Or ID' with a toggle switch set to 'Don't Include', and 'Next Approver IDs' with a toggle switch also set to 'Don't Include'. At the bottom right of the window are 'Cancel' and 'Done' buttons.

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

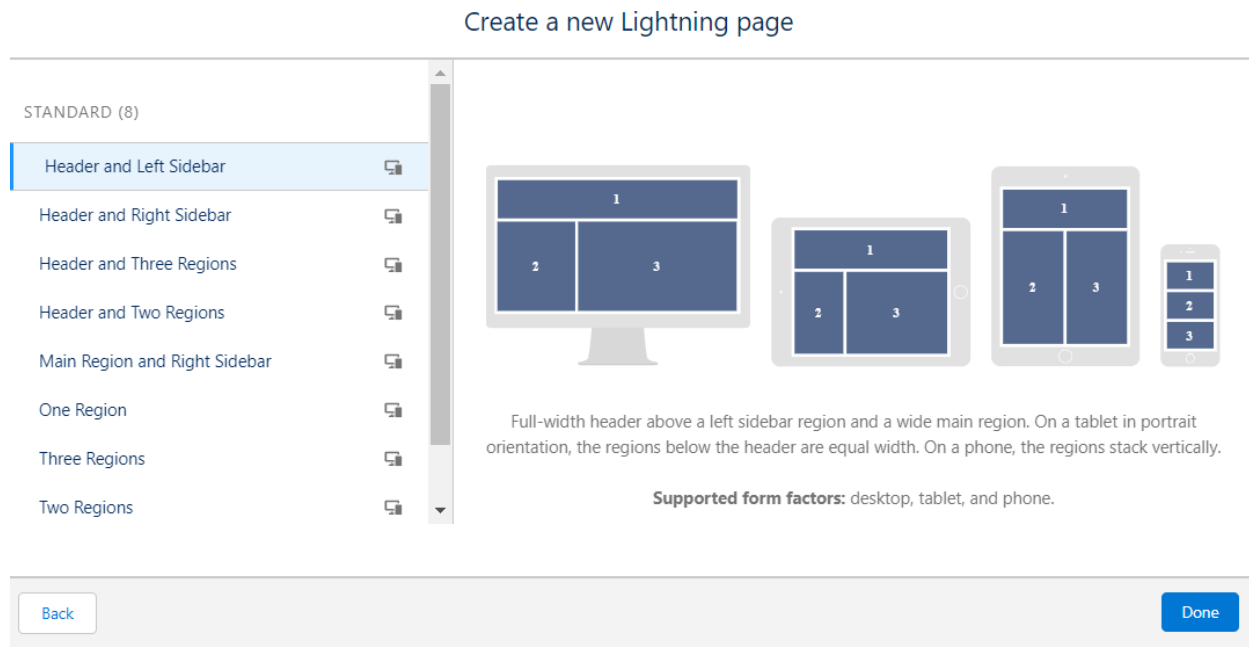
The screenshot shows a 'Save the flow' configuration window. It has two input fields at the top: '\* Flow Label' with the value 'Property Approval' and '\* Flow API Name' with the value 'Property\_Approval'. Below these is a 'Description' text area. At the bottom left is a link 'Show Advanced'. At the bottom right are 'Cancel' and 'Save' buttons.

## Milestone 11 :- Create an App Page

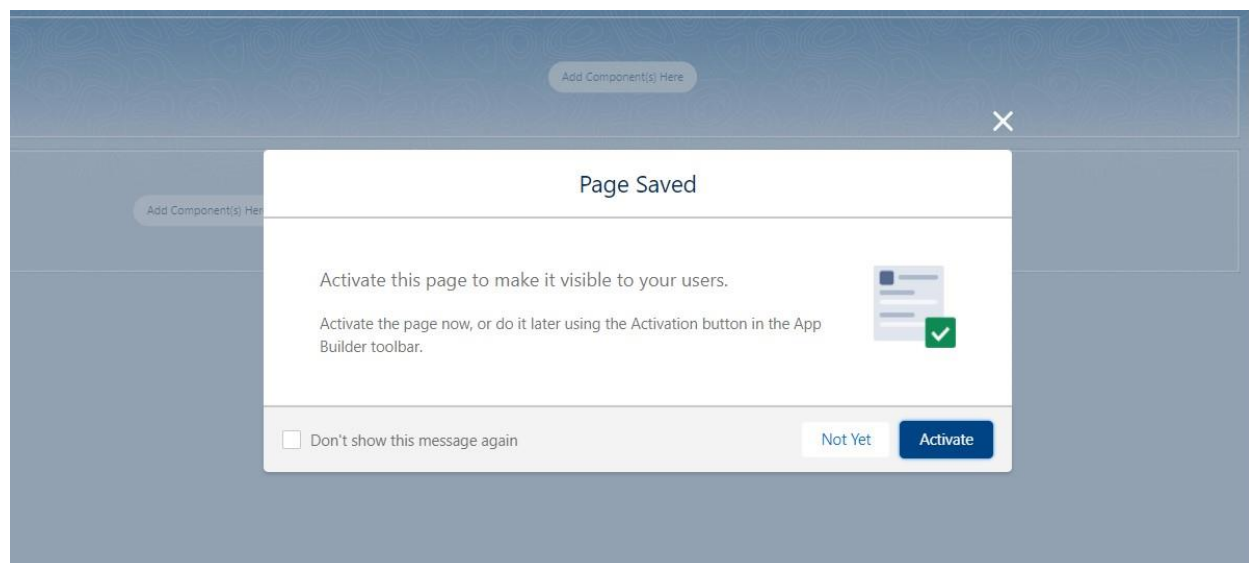
- Create an App Page on the Property details Object named as “Search Your Property”
- 1) From Setup → Go to Lightning App Builder → Click on New → Select App Page and Click on Next.



- 2) Give Label as “Search your Property” click “Next”.
- 3) Click “header and Left Sidebar” and Click on “Done”



4) Click on “Save ” and then click on “Activate”.



5) From Page Setting select page activation as “Activate for all Users”.

Activation: Search your property

PAGE SETTINGS

LIGHTNING EXPERIENCE

MOBILE NAVIGATION

Give this app page a name, set the page visibility, and choose an icon.

Name

Enter a name for your page.

Search your property

Icon

Choose an icon to represent your app in Lightning Experience and the mobile app.

Change...

Page Activation

When you activate this page, a custom tab is created for it. You can manage the tab's visibility in Setup.

☒ Activate for all users

☐ Activate for system administrators only

To set further restrictions on who sees this page, use permission sets and profile assignments in Setup.

Cancel

Save

6) From Lightning Experience Click on “Property Details” and click on Add Page“.

Activation: Search 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

LightningBolt

LightningInstrumentation

LWC Component

**Property Details**

Queue Management

Sales

Sales Console

Salesforce CMS

Property Details

Search Your Property

Search your property

↕

Remove page

Cancel

Save

7) Then Click on “Save”

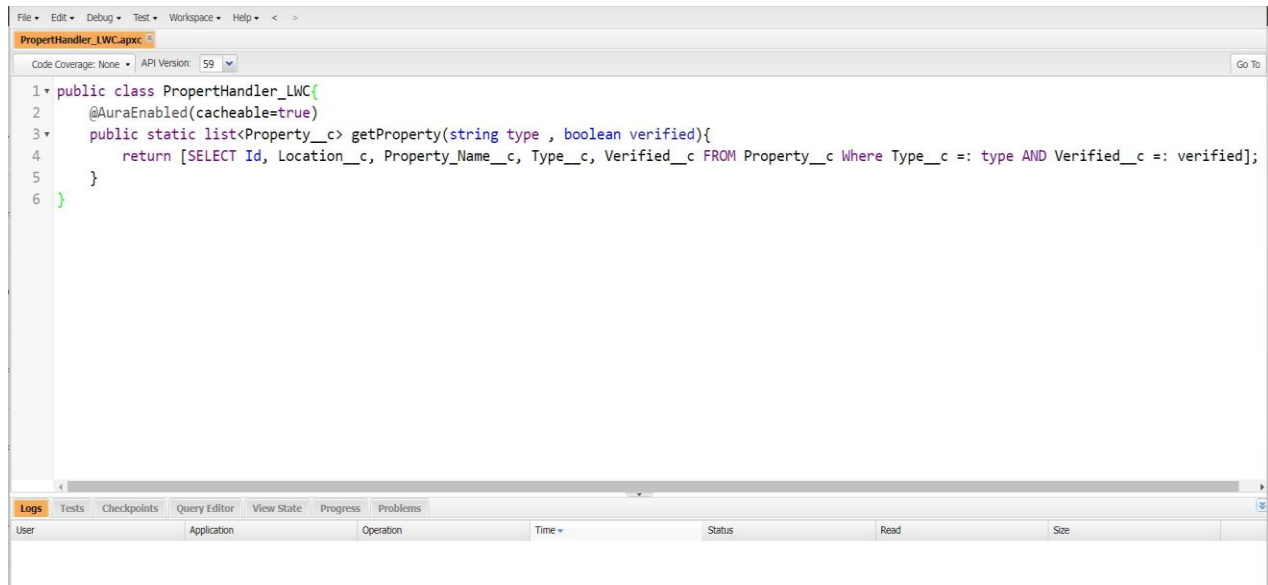
## Milestone 12 :- Create a LWC Component

- 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”

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__c> getProperty(string type , boolean verified){
        return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c FROM
Property__c Where Type__c =: type AND Verified__c =: verified];
    }
}
```



2) Create a Lightning Web Component in your VsCode, and (ctrl+shift +P) and click on authorize an org.

- 3) Enter your login id and password to authorize your org.
- 4) Now (ctrl+shift +P) → Create a lightning Web Component and Name it Anything you want to. (Example - )
- 5) In your Html File Write this code : -

**Code :-**

```

<template>
  <lightning-card>
    <div class="slds-box">
      <div class="slds-text-align_left">
        <h1 style="font-size: 20px;"><b>Properties</b></h1>
      </div>
      <div>
        <div class="slds-grid slds-gutters">
          <div class="slds-col slds-size_5-of-6">
            <lightning-combobox name="Type" label="Property Type" value={typevar}
placeholder="Select Property type"
options={propetyoptions} onchange={changehandler}></lightning-combobox>
          </div>
          <div class="slds-col slds-size_1-of-6">
            <br>
            <lightning-button-icon variant="neutral" icon-name="standard:search"
alternative-text="Search"
label="Search" onclick={handleClick}></lightning-button-icon>
          </div>
        </div>
      </div>
    </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>
</template>

```

```

1  <template>
2  <lightning-card>
3  <div class="slds-box">
4  <div class="slds-text-align_left">
5  <h1 style="font-size: 20px;"><b>Properties</b></h1>
6  </div>
7  <div>
8  <div class="slds-grid slds-gutters">
9  <div class="slds-col slds-size_5-of-6">
10 <lightning-combobox name="Type" label="Property Type" value={typevar} placeholder="Select Property type"
11 options={propetyoptions} onchange={changehandler}></lightning-combobox>
12 </div>
13 <div class="slds-col slds-size_1-of-6">
14 <br>
15 <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search"
16 label="Search" onclick={handleClick}></lightning-button-icon>
17 </div>
18 </div>
19 </div>
20
21 </div>
22
23 <template if:true={istrue}>
24 <div class="slds-box">
25 <lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-datatable>
26 </div>
27 </template>
28 <template if:false={isfalse}>
29 <div class="slds-box">
30 <div style="font-size: 15px;"><b>No properties Are Found !!</b></div>
31 </div>
32 </template>
33
34 </lightning-card>
35 </template>

```

6) In Your Js File Write this code :-

Code :-

```

import { LightningElement, api, track, wire } from 'lwc';
import getProperty from "@salesforce/apex/PropertHandler_LWC.getProperty";
import { getRecord } from 'lightning/uiRecordApi';
import USER_ID from '@salesforce/user/Id';
export default class C_01_Property_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: 'Location_c' },
    { label: "Property link", fieldName: "Property_link_c" }
]
propetyoptions = [
    { label: "Commercial", value: "Commercial" },
    { label: "Residential", value: "Residential" },
    { label: "rental", value: "rental" }
]
@wire(getRecord, { recordId: "$userId", fields: ['User.Verified_c'] })
recordFunction({ data, error }) {
    if (data) {
        console.log(data)
        console.log("This is the User Id ---> "+this.userId);
        this.verifiedvar = data.fields.Verified_c.value;
    } else {
        console.error(error)
        console.log('this is error')
    }
}

changehandler(event) {
    console.log(event.target.value);
    this.typevar = event.target.value;
}

handleClick() {

    getProperty({ type: this.typevar, verified: this.verifiedvar })
        .then((result) => {
            this.isfalse = true;
            console.log(result)
            console.log('This is the User id ---> ' + this.userId);
        })
}

```



```
    console.log('This is the verified values ---> ' + this.verifiedvar);
    if (result != null && result.length != 0) {
        this.istrue = true;
        this.propertylist = result;
        console.log(this.verifiedvar);
        console.log(this.typevar)
    } else {
        this.isfalse = false;
        this.istrue = false;
    }

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

}
```

◀ c\_01\_Property\_Management.html    **c\_01\_Property\_Management.js** X    ▶ c\_01\_Property\_Management.js-meta.xml

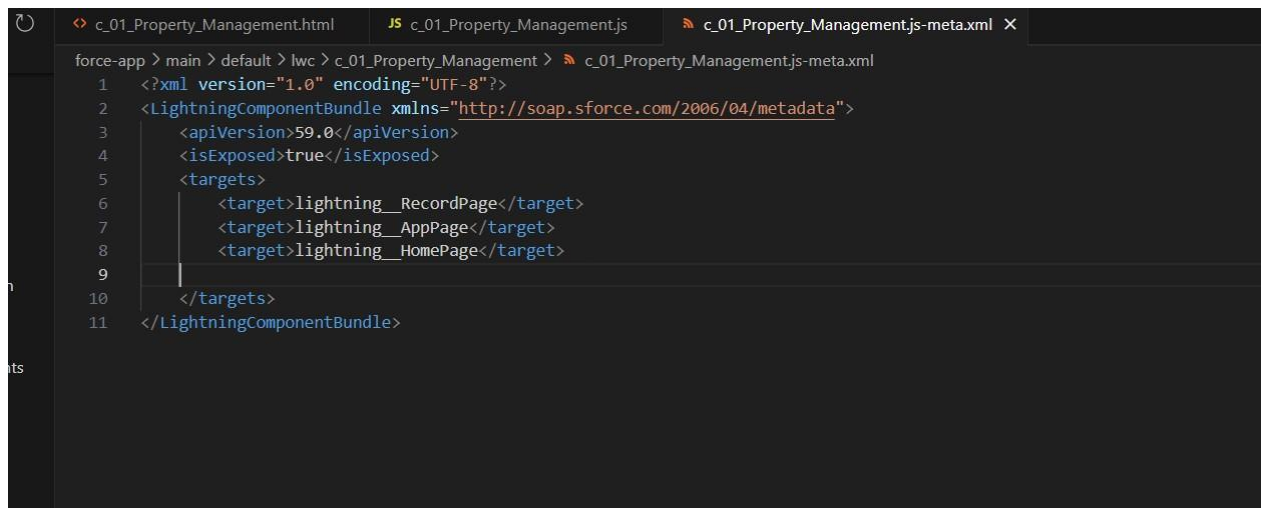
force-app > main > default > lwc > c\_01\_Property\_Management > **c\_01\_Property\_Management.js** > C\_01\_Property\_Management > propertyoptions

```
1 import { LightningElement, api, track, wire } from 'lwc';
2 import getProperty from "@salesforce/apex/PropertyHandler_LWC.getProperty";
3 import { getRecord } from 'lightning/uiRecordApi';
4 import USER_ID from '@salesforce/user/Id';
5 export default class C_01_Property_Management extends LightningElement {
6     @api recordId
7     userId = USER_ID;
8     verifiedvar
9     typevar
10    isfalse = true;
11    istrue = false;
12    @track propertylist = [];
13    columns = [
14        { label: 'Property Name', fieldName: 'Property_Name__c' },
15        { label: 'Property Type', fieldName: 'Type__c' },
16        { label: 'Property Location', fieldName: 'Location__c' },
17        { label: 'Property link', fieldName: 'Property_link__c' }
18    ]
19    propertyoptions = [
20        { label: "Commercial", value: "Commercial" },
21        { label: "Residential", value: "Residential" },
22        { label: "rental", value: "rental" }
23    ]
24
25    @wire(getRecord, { recordId: "$userId", fields: ['User.Verified__c'] })
26    recordFunction({ data, error }) {
27        if (data) {
28            console.log(data)
29            console.log("This is the User Id ----> " + this.userId);
30            this.verifiedvar = data.fields.Verified__c.value;
31        } else {
32            console.error(error)
33            console.log('this is error')
34        }
35    }
36
37    changehandler(event) {
38        console.log(event.target.value);
39        this.typevar = event.target.value;
40    }
41    handleClick() {
42
43        getProperty({ type: this.typevar, verified: this.verifiedvar })
44            .then((result) => {
45                this.isfalse = true;
46                console.log(result)
47                console.log('This is the User id ----> ' + this.userId);
48                console.log('This is the verified values ----> ' + this.verifiedvar);
49                if (result != null && result.length != 0) {
50                    this.istrue = true;
51                    this.propertylist = result;
52                    console.log(this.verifiedvar);
53                    console.log(this.typevar)
54                } else {
55                    this.isfalse = false;
56                    this.istrue = false;
57                }
58            })
59            .catch((error) => {
60                console.log(error)
61            })
62    }
63 }
64
65
66 }
```

7) In Your metafile give your targets to deploy the component.

**Code :-**

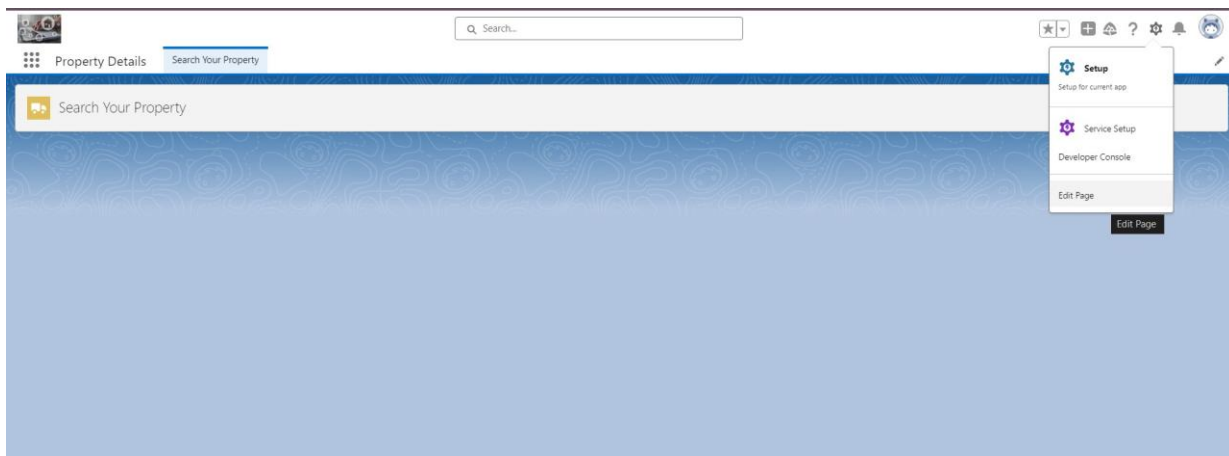
```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>59.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordPage</target>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage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



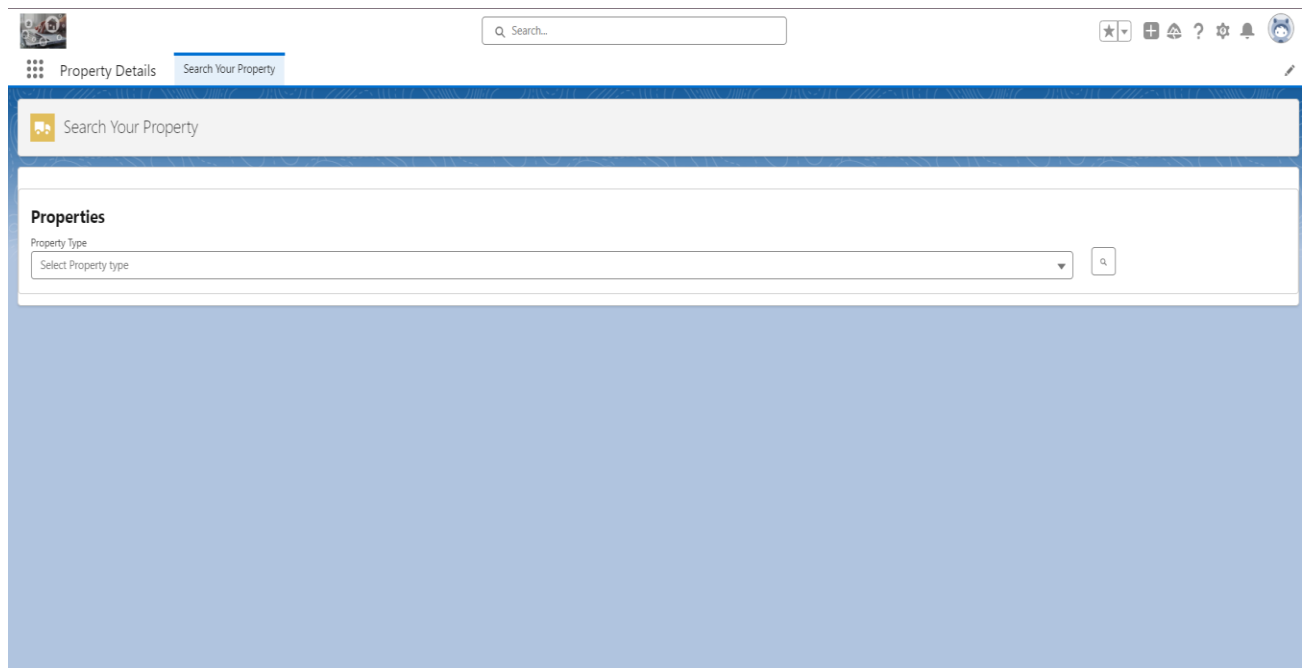
8) After Saving all the three Codes , Right Click and deploy this component to the org.

## Milestone 13 : - Drag this Component to your App Page

- 1) From Setup → Go to App Launcher → Search for Property Details
- 2) On this Page click on gear icon and click on Edit Page



- 3) Drag theComponent to your App Page and Save the Page.



## **Milestone 14 : - Give Access of Apex Classes to Profiles**

- 1) From Setup → Search For Apex Classes → Click on “Security” behind “PropertyHandler\_LWC”.
- 2) From Profiles Add “Manager” and “Customer” and “Save”.

---

\*

### **SUBMITTED BY**

au711221104056  
au711221104003  
au711221104052  
au711221104016