



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

**Email: [sankarthavasi5@gmail.com](mailto:sankarthavasi5@gmail.com)**

**Team leader NM ID: 30AFB4D5A7AAD7D9ADE426341F285406**

**Team members:** Thavasi S, Silambarasan E, Thamizhmani M, Charles A

**Arunai Engineering college (An autonomous institution)**

## **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 12 : 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 settings to control access to customer records based on approval status.

## **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.

Ensure that approved users are granted access to curated property listings tailored to their preferences.

**1) 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.

**2) 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.

**3) Scalability and Performance Requirements:**

Ensure that the integration and Salesforce configuration are scalable to accommodate future growth in customer volume and data. Optimize system performance to ensure responsiveness and reliability, especially during peak usage periods.

**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.

The screenshot shows a Google search results page with a dark theme. The search query 'jotform' is entered in the search bar. The results page includes a snippet from the official Jotform website, links to 'Login', 'My Forms', 'Pricing', 'Form Templates', and 'Jotform Apps'. To the right, there is a detailed card for Jotform, providing information about the company's history, location, and social media links.

**Google** **jotform**

All Images Videos Shopping News Forums Web More Tools

**Jotform** <https://www.jotform.com> **Jotform: Free Online Form Builder & Form Creator** Create forms and surveys for free with Jotform's drag-and-drop form builder. Start collecting registrations, applications, orders, and payments today.

**Login** Jotform's form builder helps you create & publish online forms ...

**My Forms** Access and manage your forms and submissions on Jotform's ...

**Pricing** Whether you need a robust online form solution, or just the basics ...

**Form Templates** Jotform offers the largest selection of free form templates available ...

**Jotform Apps** Jotform Apps is a no-code app maker enabling you to build ...

**People also ask** :

**Jotform** Company [jotform.com](https://www.jotform.com)

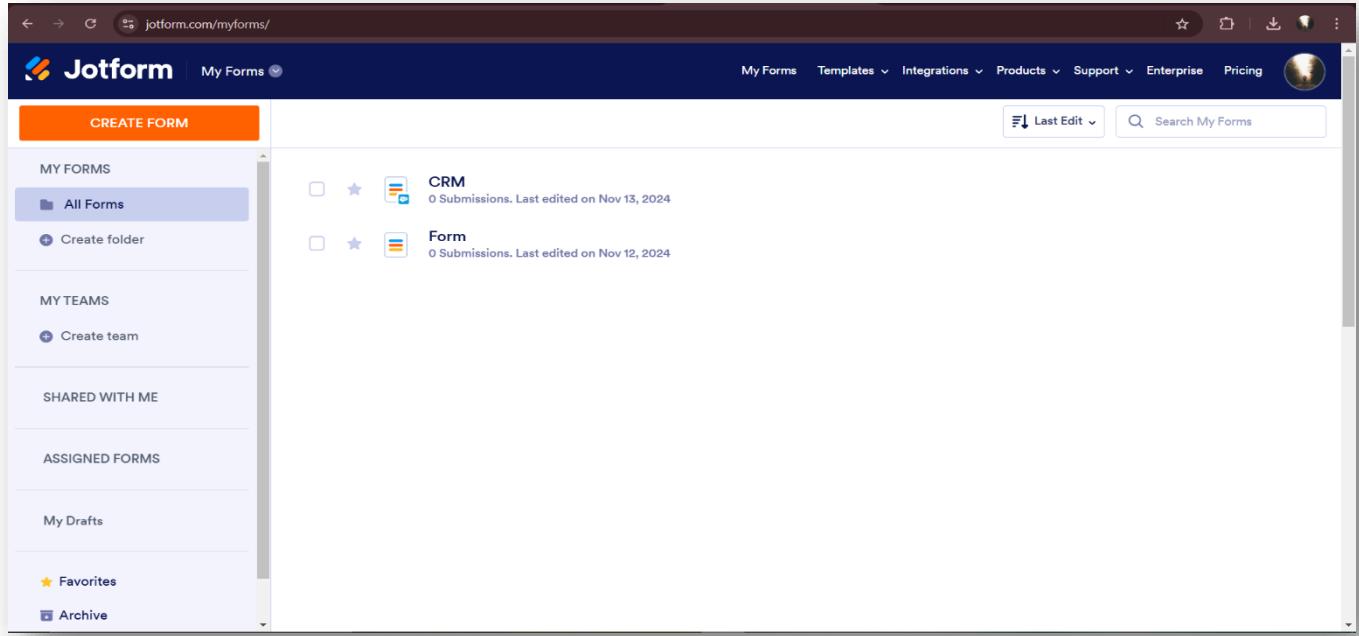
Jotform is a San Francisco-based company for building online forms. Founded in 2006 by Aytekin Tank, Jotform also offers a mobile app builder, a PDF editor, and electronic signature collection services. As of September 2023, it has more than 20 million users. [Wikipedia](#)

**Founded:** 2006  
**Available in:** 18 languages  
**Employees:** 550  
**Founder(s):** Aytekin Tank  
**Headquarters:** San Francisco, California, U.S

[Disclaimer](#)

Profiles: X (Twitter), Instagram, YouTube, Pinterest

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.

← → ⌛ jotform.com/build/243122874503048#preview

https://form.jotform.com/243122874503048  

Phone  Tablet  Desktop  Preview Form 

ORIENTATION  



## CRM

Name \*

First Name

Last Name

← → ⌛ jotform.com/build/243122874503048#preview

https://form.jotform.com/243122874503048  

Phone  Tablet  Desktop  Preview Form 

ORIENTATION  

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 \*

The screenshot shows a JotForm form builder interface with the following details:

- Budget Amount \***: An input field with placeholder "e.g., 23".
- Address \***: A multi-line input field divided into three sections:
  - Street Address
  - Street Address Line 2
  - City and State / Province side-by-side inputs.
- Postal / Zip Code**: A single-line input field.
- Submit**: A green rectangular button at the bottom center.

The top of the screen has a navigation bar with the URL <https://form.jotform.com/243122874503048#preview>, a "Fill Form" button, and device preview buttons for Phone, Tablet, and Desktop. There is also a "Preview Form" toggle switch.

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

<https://form.jotform.com/243122874503048>

## 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.

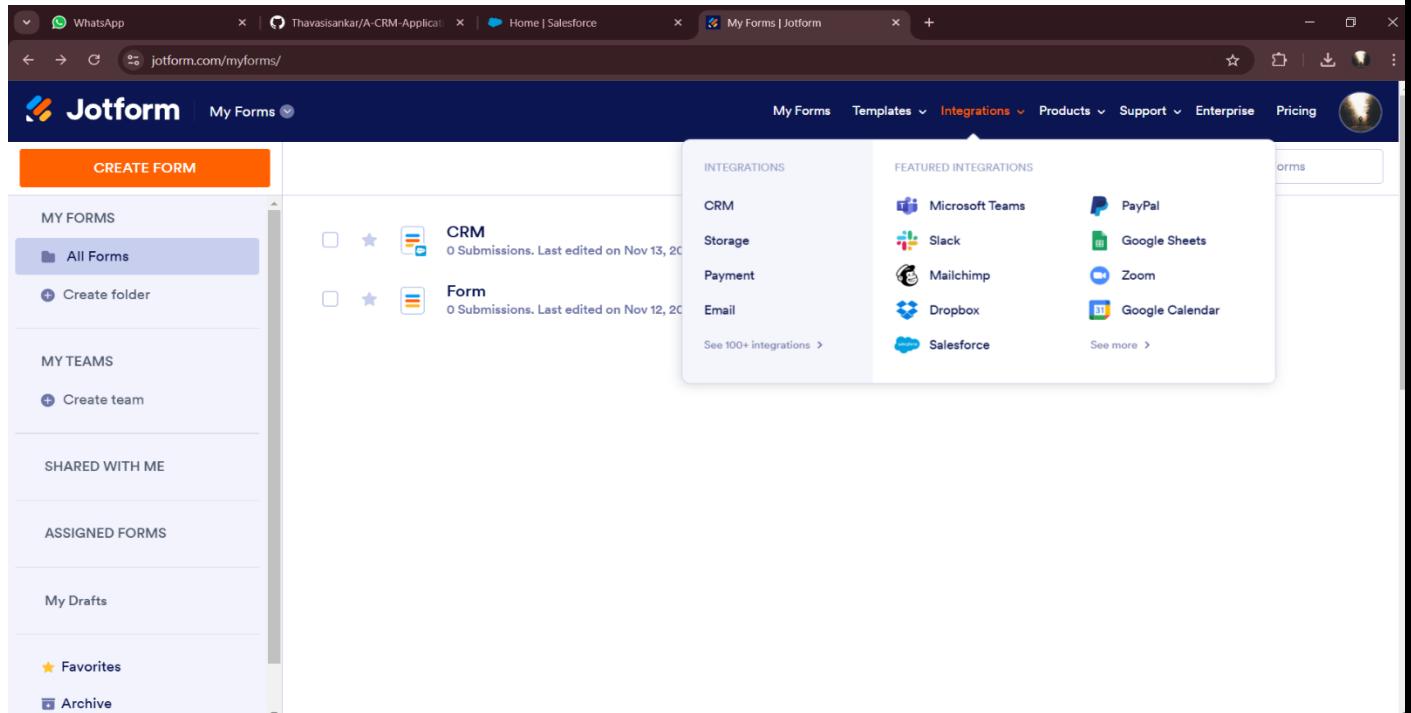
## ● Create Property object

- 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”.

The screenshot shows the Jotform integration page for Salesforce. At the top, there's a header with the Jotform logo, a search bar, and links for 'Enterprise' and 'Pricing'. Below the header, a large central image illustrates the integration between Jotform and Salesforce. It shows a mobile phone displaying a 'Sales Contact Form' with fields for Name, E-mail, and Phone. Arrows point from the form to a laptop screen showing 'Contact Details' for 'Bernard Swords'. A 'Salesforce' logo is visible at the bottom of the mobile phone screen. To the right of the main image, there's a call-to-action button 'Send new leads, contacts, or accounts to your sales CRM'. Below this, a note says 'Salesforce is a powerful CRM tool used to help salespeople manage accounts, improve client'. On the left side of the main content area, there's a sidebar with sections for 'PRODUCT' (Form, E-Sign, Workflows), 'CATEGORIES' (Analytics & Reporting, Automation, Blogging, CMS, CRM, Communication), and a 'Form Integrations' section. On the right side, there are cards for other integrations like ActiveCampaign, Pipedrive, Highrise, and Moxie.

3) Select the Org with which you want to Integrate your jotform with.

The screenshot shows the Jotform Form Builder interface for a form titled 'Dreams World'. The top navigation bar includes 'Form Builder', 'Add Collaborators', 'Help', and a 'Preview Form' button. The main content area has tabs for 'BUILD', 'SETTINGS', and 'PUBLISH'. Under the 'BUILD' tab, there's a sidebar with various settings: 'FORM SETTINGS' (Customize form status and properties), 'EMAILS' (Send autoresponders and notifications), 'CONDITIONS' (Set up conditional logic), 'THANK YOU PAGE' (Show page after submission), 'INTEGRATIONS' (Connect your form to other apps), 'APPROVAL FLOWS' (Create an approval flow), 'JOTFORM SIGN' (Power your forms with Jotform Sign), and 'MOBILE NOTIFICATIONS' (Customize mobile app notifications). The main panel displays information about the Salesforce integration, including a note that it's available via AppExchange and a list of actions it can perform: 'Add new company leads', 'Add new contacts', 'Add new accounts', and 'Connect any custom object'. It also provides a link to learn how to integrate the form with Salesforce. The 'Authentication' section allows selecting a Salesforce account, with a dropdown menu showing 'dada rao - prajwal@thesmartbridge.com'. A 'Send Feedback' button is located at the bottom right.

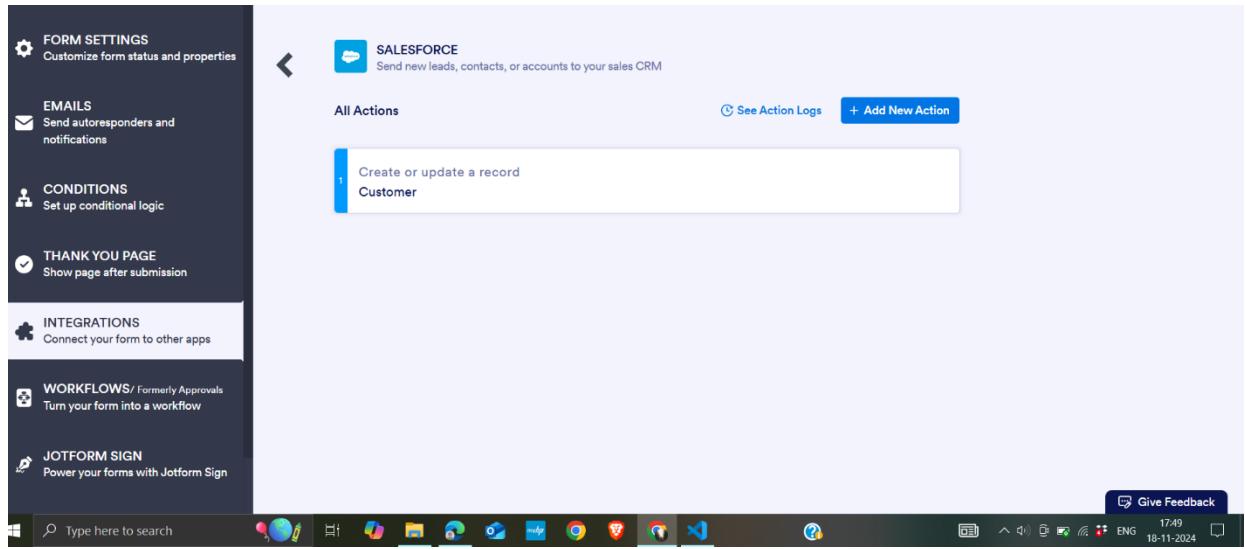
- 4) Select an Action - Create a record.  
 Select a Salesforce Object : - Customer

The screenshot shows the Jotform Form Builder interface. The top navigation bar has tabs for BUILD, SETTINGS, and PUBLISH, with 'BUILD' currently selected. A 'Preview Form' button is in the top right. On the left, a sidebar lists various settings: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE (checked), INTEGRATIONS, APPROVAL FLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The main workspace shows the 'Create a record' action selected. Below it is a 'Find existing record' option. A 'Select a Salesforce Object' dropdown shows 'Customer' selected. The 'Object Fields' section shows 'Customer' mapped to 'Customer'. The 'CRM' tab is active. At the bottom, there's a note about updating an existing record and a 'Send Feedback' button.

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

The screenshot shows the Jotform Form Builder interface with the 'Create a record' action selected. The 'Object Fields' section maps form fields to CRM fields: Name to Name, Email to Email, Phone Number to Phone Number, Property Type to Which Type of Property are you lookin..., Budget Amount to Budget Amount, Street Address to Address, Street Address line 2 to Address - Street Address 2, City to Address - City, State to Address - State, and postal code to Address - Postal/Zip Code. The 'CRM' tab is active. The top navigation bar includes 'Add Collaborators', 'Help', and a user profile icon. The bottom right has a 'Give Feedback' button.

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



## Milestone 4 : Create Roles

- **Sales Executive Role**

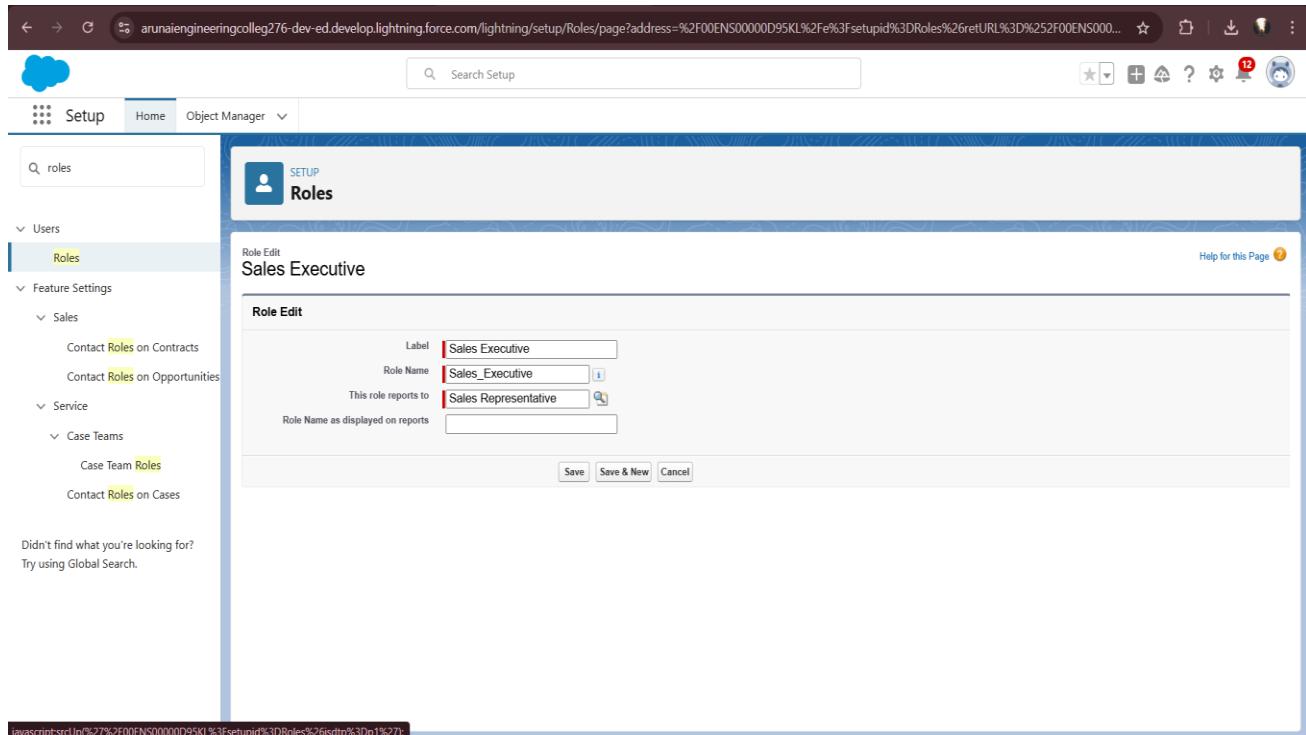
- 1) Go to Setup and Click on Roles, then click on Expand all and Add a Role just below the Sales Representative

The screenshot shows the Salesforce Setup Roles page. The left sidebar has a search bar and navigation links for Users, Feature Settings, Sales, Service, and Case Teams. The main content area is titled "Roles" and displays a hierarchical tree of roles:

- Add Role
- SVP, Human Resources (Edit | Del | Assign)
- SVP, Sales & Marketing (Edit | Del | Assign)
  - Add Role
  - Sales Representative (Edit | Del | Assign)
    - Add Role
    - Sales Executive (Edit | Del | Assign)
      - Add Role
      - Sales Manager (Edit | Del | Assign)
        - Add Role
        - Customer (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

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

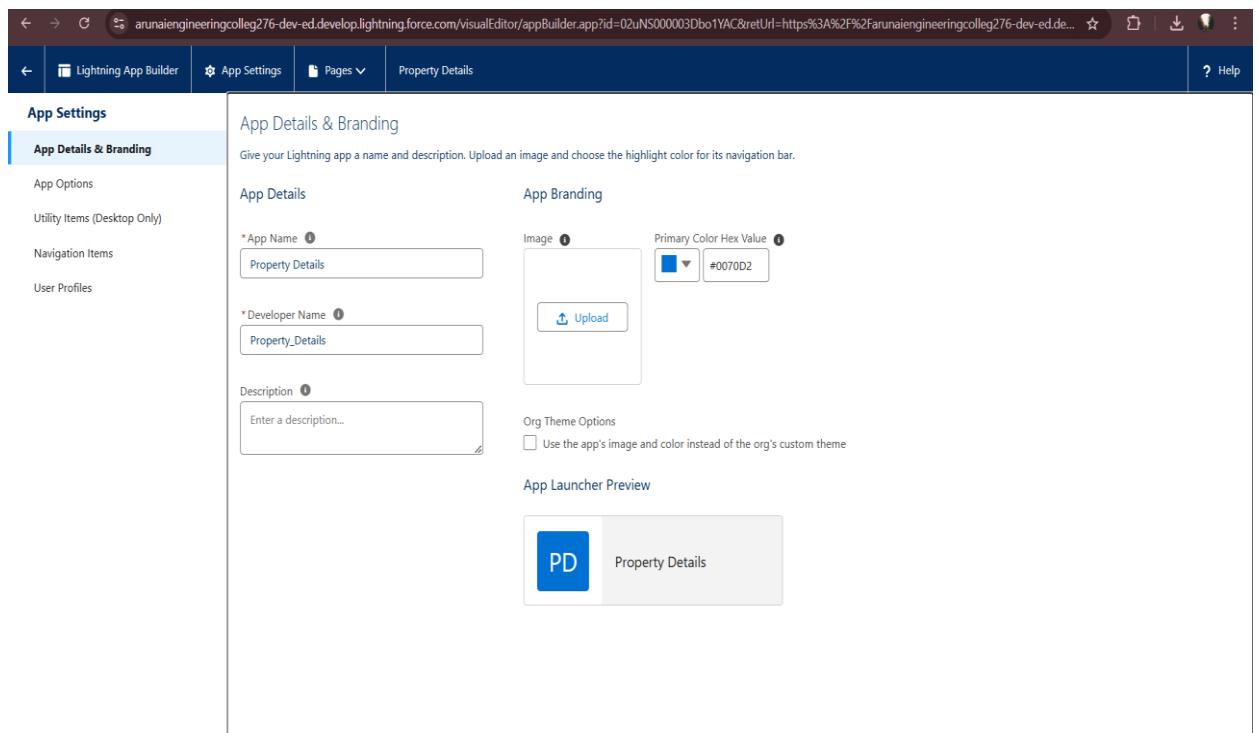
- 2) Label - Sales Executive
- Reports to - Sales Representative



- 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.

## 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.



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" ..

The screenshot shows the Salesforce Setup interface with the 'Profiles' page open. The 'Profiles' tab is selected in the sidebar. The main area displays a list of profiles with columns for Action, Profile Name, User License, and Custom. The 'Standard Platform User' profile is selected and highlighted in blue. The 'User License' column for this profile shows 'Salesforce Platform'. Other profiles listed include 'Salesforce API Only System Integrations', 'Silver Partner User', 'Solution Manager', 'Standard User', and 'System Administrator'. The 'User License' column for these profiles shows 'Salesforce'.

- 2) Uncheck all the Custom Objects and Check only Property Details From CustomApp Settings
  
- 1) Uncheck all the Custom Objects and Check only Property Details From Custom App Settings.

The screenshot shows the Salesforce Setup interface with the 'Customer' profile edit screen open. The 'Customer' profile is selected in the sidebar. The main area shows the 'Profile Edit' screen for the 'Customer' profile. Under the 'Custom App Settings' section, the 'CRM (CRM)' object is checked under both 'Visible' and 'Default'. Under the 'Platform (standard\_Platform)' section, 'Property Details (Property\_Details)' is checked under 'Default'. There is a note at the bottom right indicating 'I = Required Information'.

## **2) Also Remove all the Standard Object Permissions.**

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

The screenshot shows the Salesforce Setup interface for managing Profiles. The left sidebar has a search bar and navigation links for Users and Profiles. A message says "Didn't find what you're looking for? Try using Global Search." The main content area is titled "Profiles". It displays two tables of permissions:

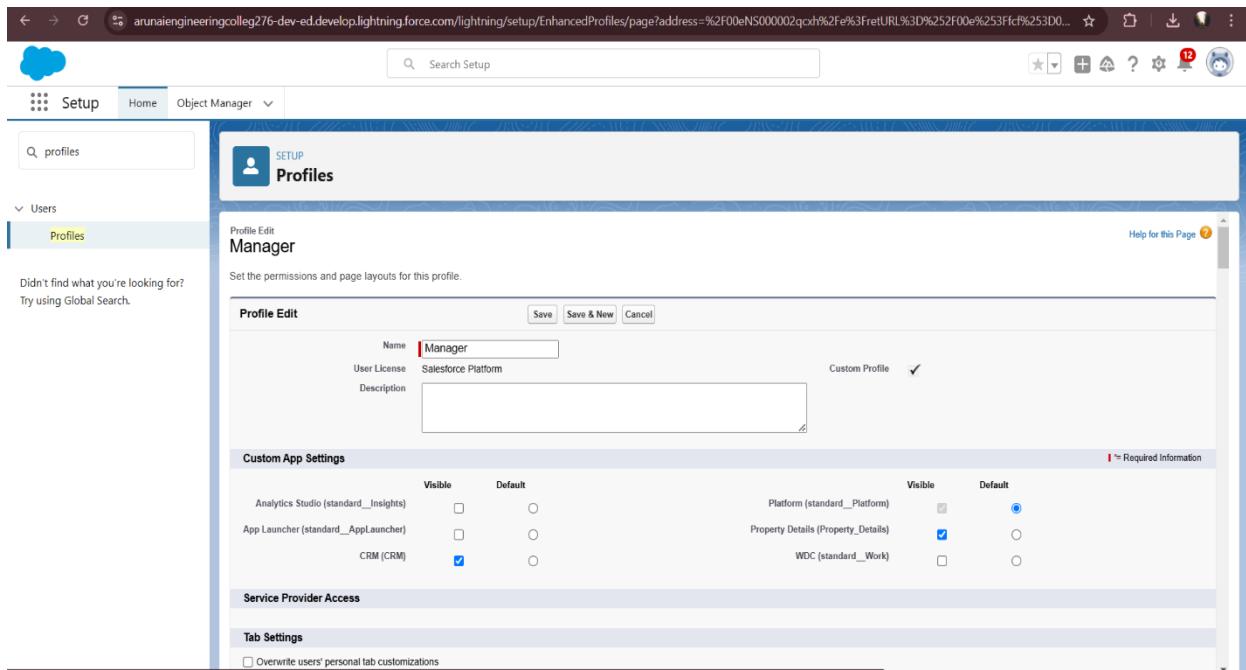
Object	Background Operations	Engagement Channel Types
Business Brands	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Communication Subscriptions	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<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"/>	<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"/>	<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"/>	<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"/>	<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"/>	<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"/>	<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"/>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Object	Basic Access	Data Administration
	Read Create Edit Delete View All <small>i</small> Modify All <small>i</small>	Read Create Edit Delete View All <small>i</small> Modify All <small>i</small>
Customers	<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"/>
Properties	<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"/>

Session Settings at the bottom include "Session Times Out After" set to "2 hours of inactivity" and "Session Security Level Required at Login" set to "-None-".

- **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.



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

## 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”

## **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 Proces

## 2) Process Name - Property Approval

The screenshot shows the Salesforce Setup interface. On the left, the navigation sidebar is open, showing various categories like Feature Settings, Marketing, Sales, Service, and Process Automation. Under Process Automation, 'Approval Processes' is selected. The main content area is titled 'Approval Processes' and shows a sub-page for 'Property Approval'. The sub-page is titled 'Step 1. Enter Name and Description'. It has fields for 'Process Name' (set to 'Property Approval') and 'Unique Name' (set to 'Property\_Approval'). There is also a 'Description' field which is currently empty. At the bottom right of the sub-page, there are 'Save', 'Next', and 'Cancel' buttons.

## 3) Give 2 criteria →

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

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

The screenshot shows the continuation of the 'Approval Processes' setup. The sidebar remains the same, with 'Approval Processes' still selected. The main content area is now titled 'Step 2. Specify Entry Criteria'. It contains a section titled 'Specify Entry Criteria' with the instruction: '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.' Below this, there is a table where users can define criteria. The table has columns for 'Field', 'Operator', and 'Value'. The first row defines a criterion: 'Property: Location' with 'operator' 'not equal to' and 'value' 'blank'. The second row defines another criterion: 'Property: Verified' with 'operator' 'equals' and 'value' 'False'. There are three additional rows labeled '-None-' which are currently empty. At the bottom of the table, there is a link 'Add Filter Logic...'. At the bottom right of the page, there are 'Previous', 'Save', 'Next', and 'Cancel' buttons.

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

The screenshot shows the Salesforce Setup interface for creating an approval process. The left sidebar navigation includes Feature Settings, Marketing, Sales, Service, and Process Automation categories, with Approval Processes selected. The main content area is titled "Approval Process Edit: Property Approval" and is on "Step 3 of 6". It contains sections for "Select Field Used for Automated Approval Routing" (set to Manager) and "Record Editability Properties" (set to Administrators OR the currently assigned approver). Buttons for Previous, Save, Next, and Cancel are at the bottom.

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

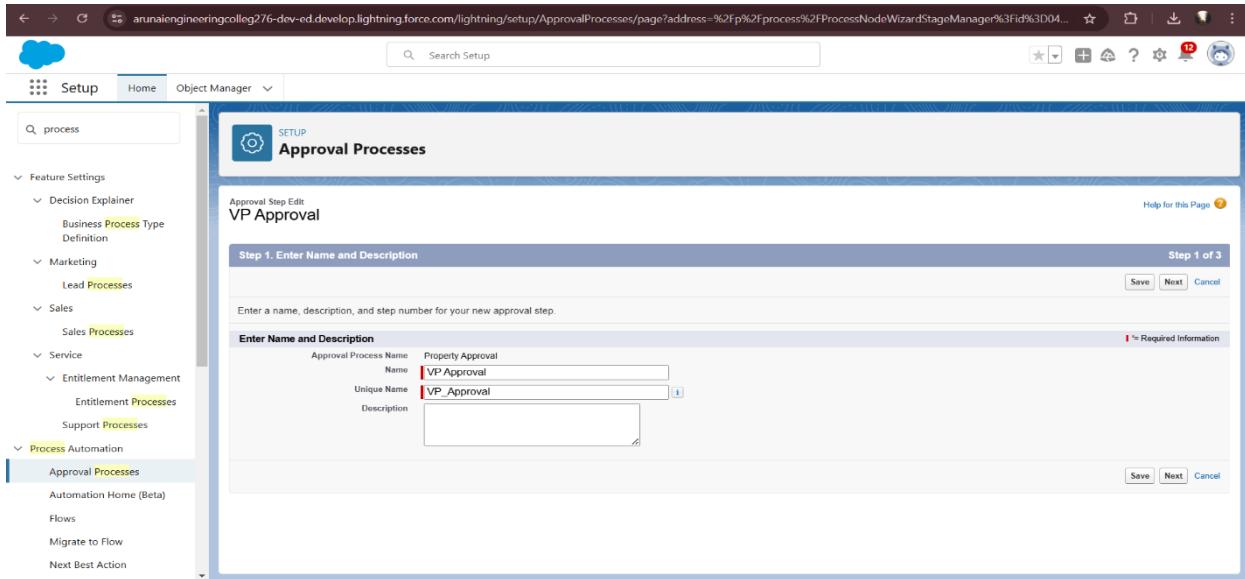
The screenshot shows the Salesforce Setup interface for creating an approval process, specifically Step 5 of 6. The left sidebar navigation includes Feature Settings, Marketing, Sales, Service, and Process Automation categories, with Approval Processes selected. The main content area is titled "Approval Process Edit: Property Approval" and is on "Step 5 of 6". It displays a list of "Available Fields" (Created By, Last Modified By, Verified) and a "Selected Fields" list containing (Property, Owner, Location, Property Name, Type). A preview of the approval page layout is shown on the right, and a link "Click here to view an example" is provided. Buttons for Previous, Save, Next, and Cancel are at the bottom.

7) Click Next and Select the initial Submitters →

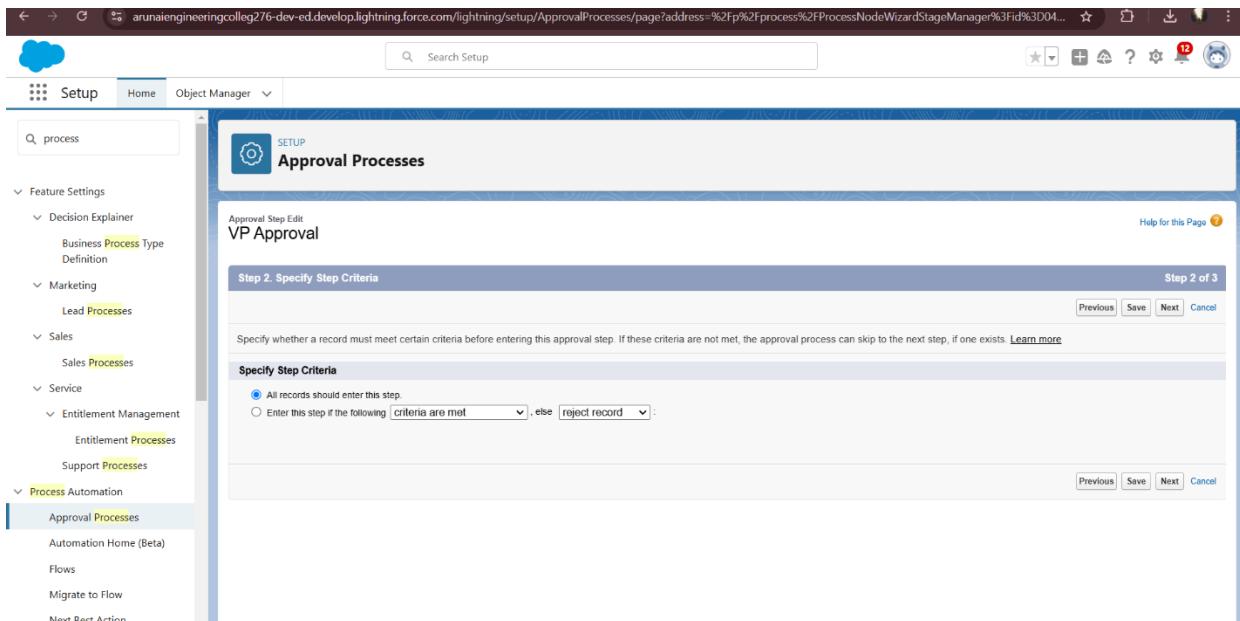
- a ) Owner → Property Owner
- b) Roles → Sales Manager

8) Save.

9) Add an approval step name “Executive Approval ”



10) specify the Criteria → All record should enter



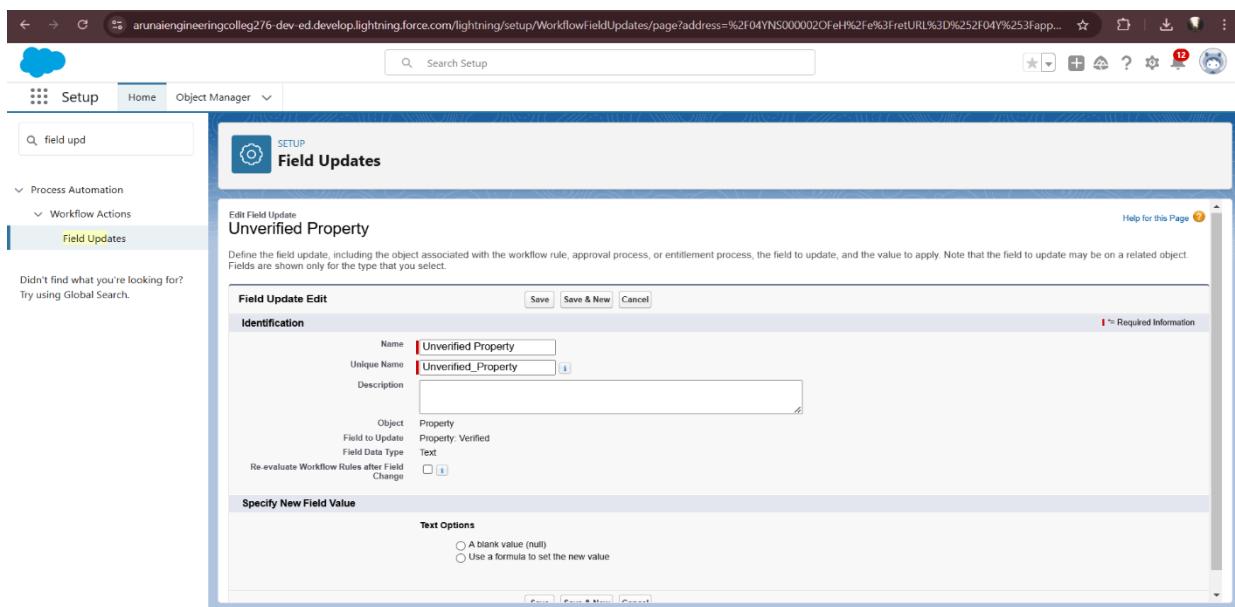
11) click next and select the Approver as “ Sales Executive “ and “Save”

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

- Select Object → Property
- Field to Update → Verified
- Field Data Type → CheckBox
- Select CheckBox Option as “True”
- Save.

13) Add One field Update as “UnVerified Property”

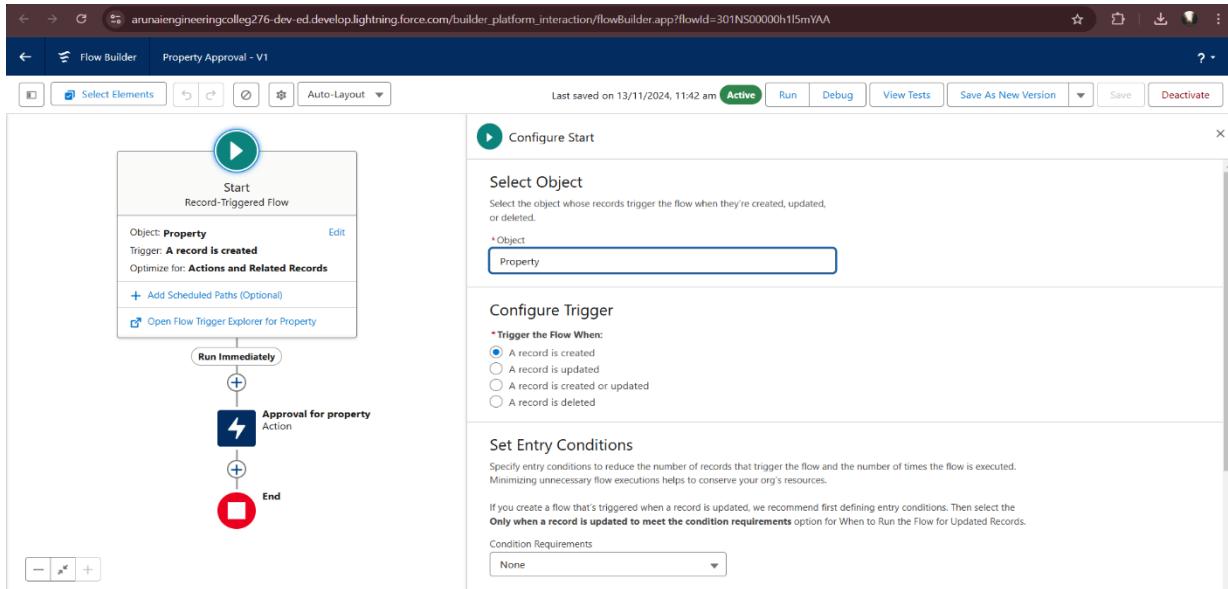
- a) Select Object → Property
- b) Field to Update → Verified
- c) Field Data Type → CheckBox
- d) Select CheckBox Option as “False”
- e) Save.



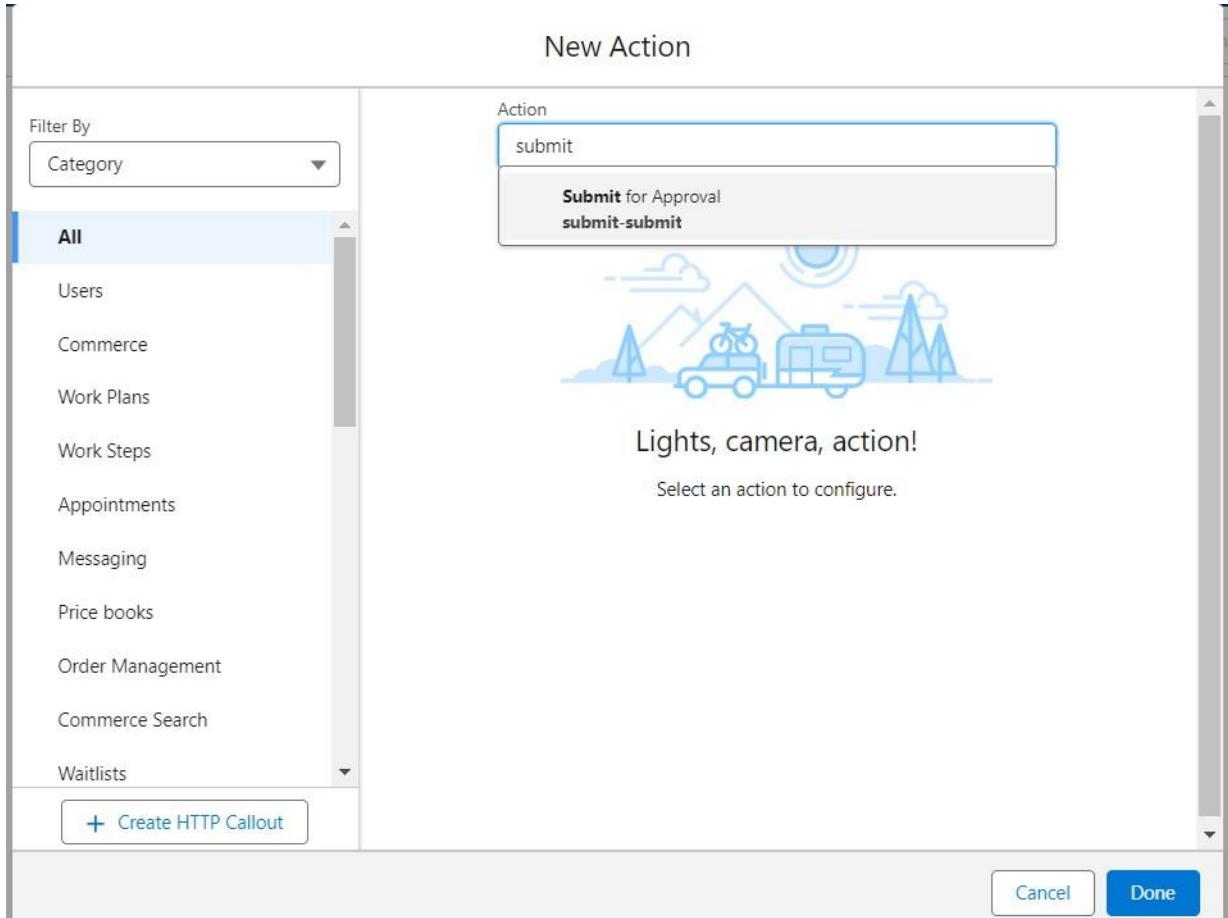
14) Activate the Approval Process.

## Milestone 10 : - Create a Record trigger flow to submit the ApprovalProcess 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

### New Action

Filter By Category ▾

Action Submit for Approval

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.

\* Label Approval for property \* API Name Approval\_for\_property

Description

Set Input Values for the Selected Action

A\_a \* Record ID ⓘ {!\$Record.Id}

A\_a Approval Process Name Or ID  Don't Include

A\_a Next Approver IDs  Don't Include

Cancel Done

+ Create HTTP Callout

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

### Save the flow

\* Flow Label Property Approval \* Flow API Name Property Approval

Description

Show Advanced

Cancel Save

\* Flow Label  
Property Approval

\* Flow API Name  
Property Approval

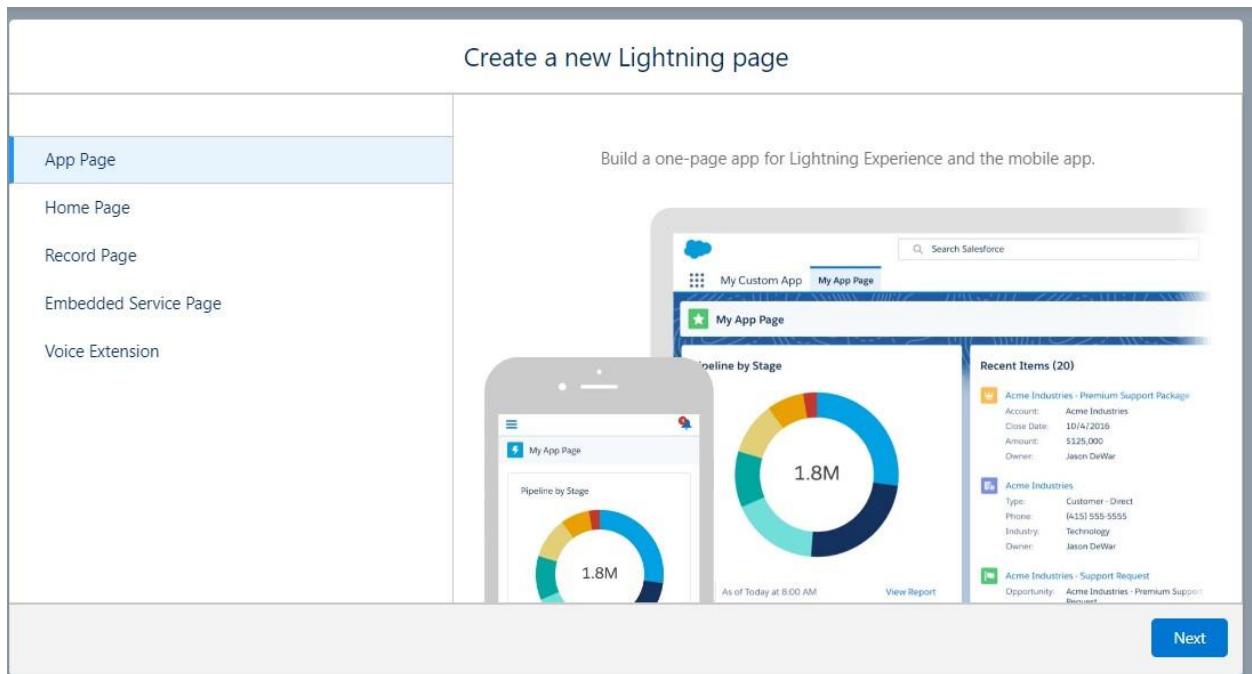
Description

Show Advanced

Cancel Save

# 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”

## Create a new Lightning page

STANDARD (8)

Header and Left Sidebar



Header and Right Sidebar



Header and Three Regions



Header and Two Regions



Main Region and Right Sidebar



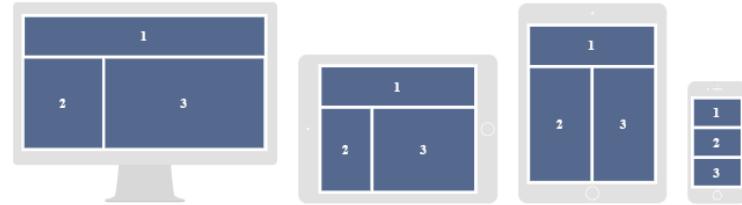
One Region



Three Regions



Two Regions



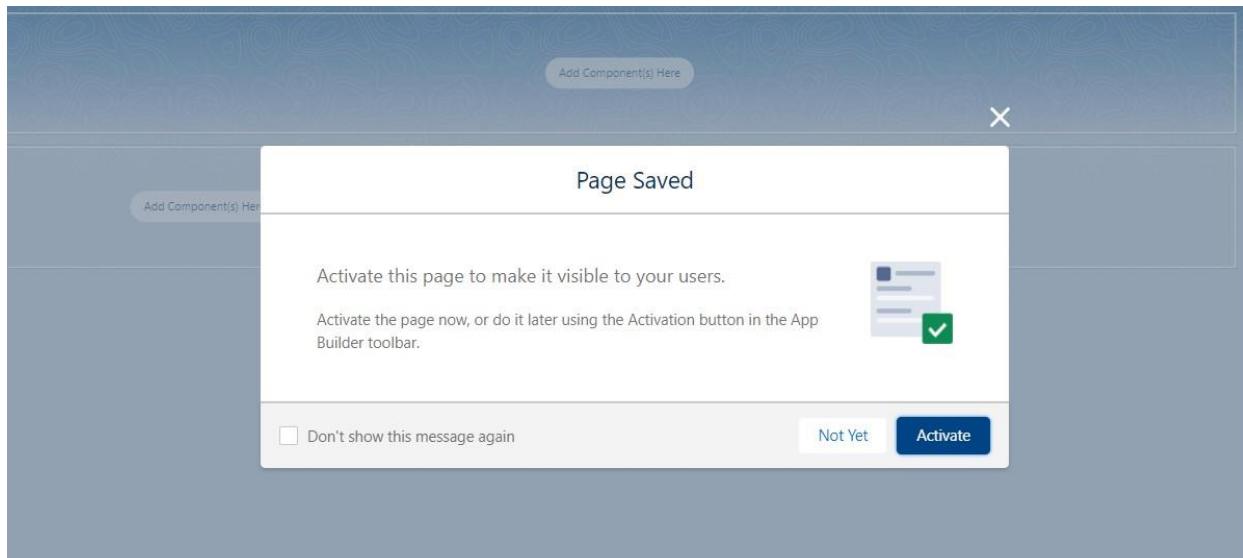
Full-width header above a left sidebar region and a wide main region. On a tablet in portrait orientation, the regions below the header are equal width. On a phone, the regions stack vertically.

**Supported form factors:** desktop, tablet, and phone.

Back

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

<input type="checkbox"/> LightningBot
<input type="checkbox"/> LightningInstrumentation
<input type="checkbox"/> LWC Component
<input checked="" type="checkbox"/> Property Details
<input type="checkbox"/> Queue Management
<input type="checkbox"/> Sales
<input type="checkbox"/> Sales Console
<input type="checkbox"/> Salesforce CMS

Property Details

Remove page

Search Your Property

Search your property

Cancel

Save

7) Then Click on “Save”

## Milestone 12 :- Create a LWC Component

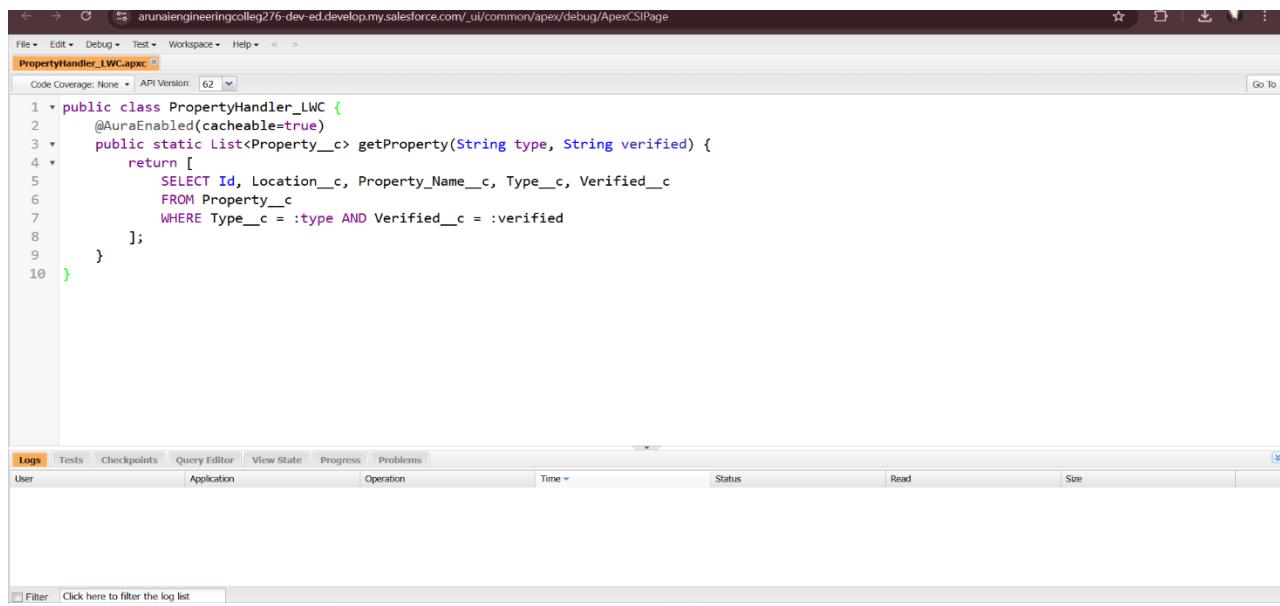
- Create an Lwc Component for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on “Search your Property Page”

- 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];
}

}
```



```
1 * public class PropertyHandler_LWC {
2     @AuraEnabled(cacheable=true)
3     public static List<Property__c> getProperty(String type, String verified) {
4         return [
5             SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c
6             FROM Property__c
7             WHERE Type__c = :type AND Verified__c = :verified
8         ];
9     }
10 }
```

The screenshot shows the Salesforce Dev Console interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area displays the code for the `PropertyHandler_LWC.apxc` class. The code defines a static method `getProperty` that queries the `Property__c` object using a WHERE clause to filter by `Type__c` and `Verified__c`. Below the code editor is a tabs bar with **Logs** selected, followed by Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The logs table has columns for User, Application, Operation, Time, Status, Read, and Size. A filter input field at the bottom allows users to search the log list.

- 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={propertyoptions} onchange={changehandler}></lightning-combobox>
          </div>
          <div class="slds-col slds-size_1-of-6">
            <br>
            <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search" label="Search" onclick={handleClick}></lightning-button-icon>
          </div>
        </div>
      </div>
    </div>
  </template if:true={istrue}>

```

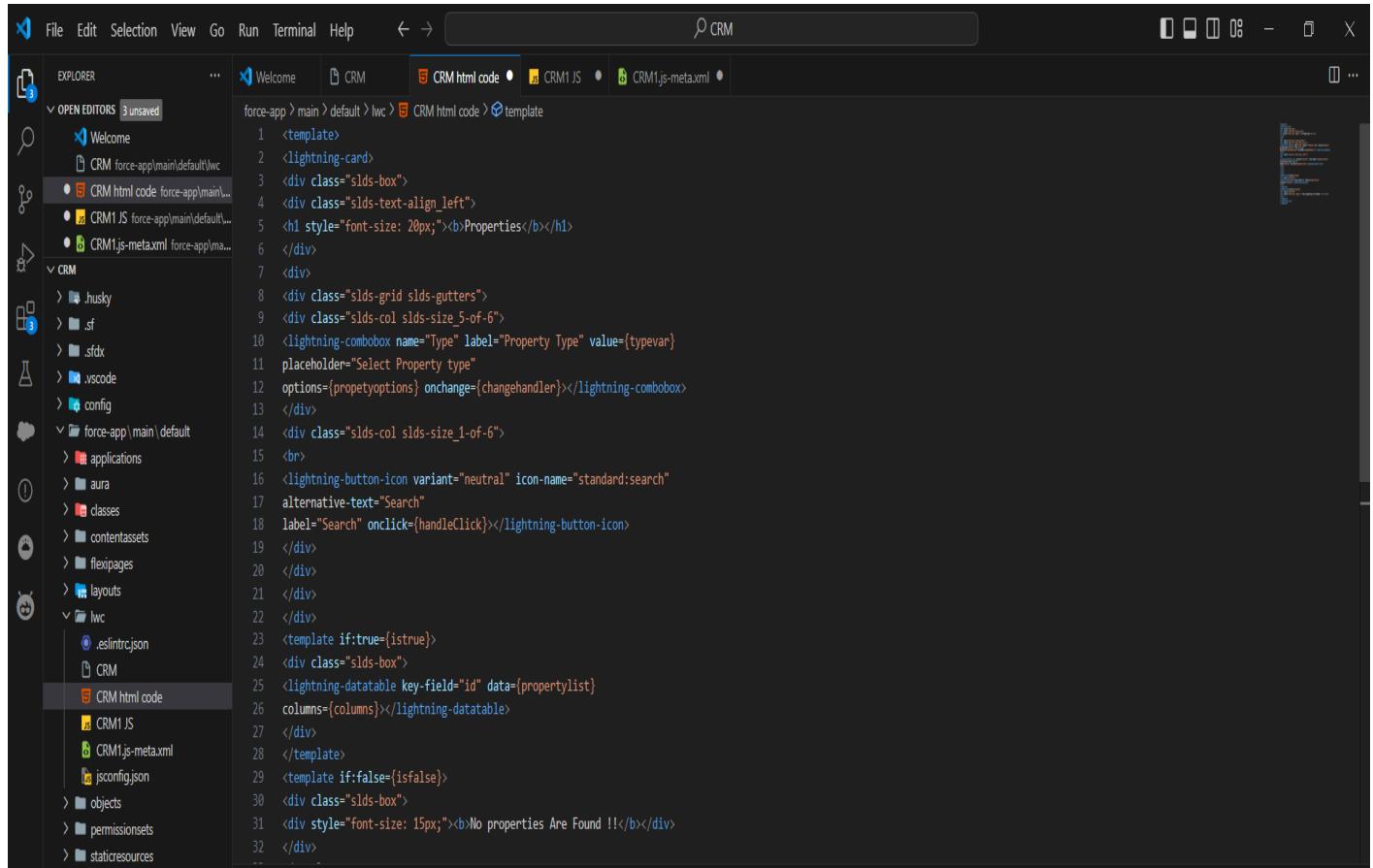
```
<div class="slds-box">
  <lightning-datable key-field="id"
data={propertylist}
columns={columns}></lightning-
datatable>
</div>
</template>
<template if:false={isfalse}>
```

```

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

</lightning-card>
</template>

```



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;
```

```
verified
var type
var 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" }
]
propertyoptions = [
  { label: "Commercial", value: "Commercial" },
  { label: "Residential", value: "Residential" },
  { label: "rental", value: "rental" }
]

]
@wire(getRecord, { recordId: "$userId", fields: ['User.Verified_c'] })recordFunction({ data, error })
{
  if (data) {
    console.log(data)
    console.log("This is the User Id ---> "+this.userId);this.verifiedvar =
    data.fields.Verified_c.value;
```

```
    } else {
      console.error(err
      or)
      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.veri
fiedvar);
            console.log(this.type
var)
        } else {
            this.isfalse =
            false;
            this.isTrue =
            false;
        }
    })
    .catch((error) => {
        console.log(error)
    })
}
}
```

The screenshot shows a dark-themed instance of the Visual Studio Code (VS Code) code editor. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar containing 'CRM'. The title bar displays 'CRM' with standard window control icons.

The left sidebar, titled 'EXPLORER', lists the project structure:

- OPEN EDITORS (3 unsaved): Welcome, CRM html code, CRM1 JS
- CRM (selected):
  - .husky
  - .sf
  - .sfdx
  - .vscode
  - config
- force-app\main\default:
  - applications
  - aura
  - classes
  - contentassets
  - flexipages
  - layouts
  - lwc
    - .eslintrc.json
    - CRM
    - CRM html code
    - CRM1 JS (selected)
    - CRM1js-meta.xml
    - jsconfig.json
    - objects
    - permissionsets
    - staticresources

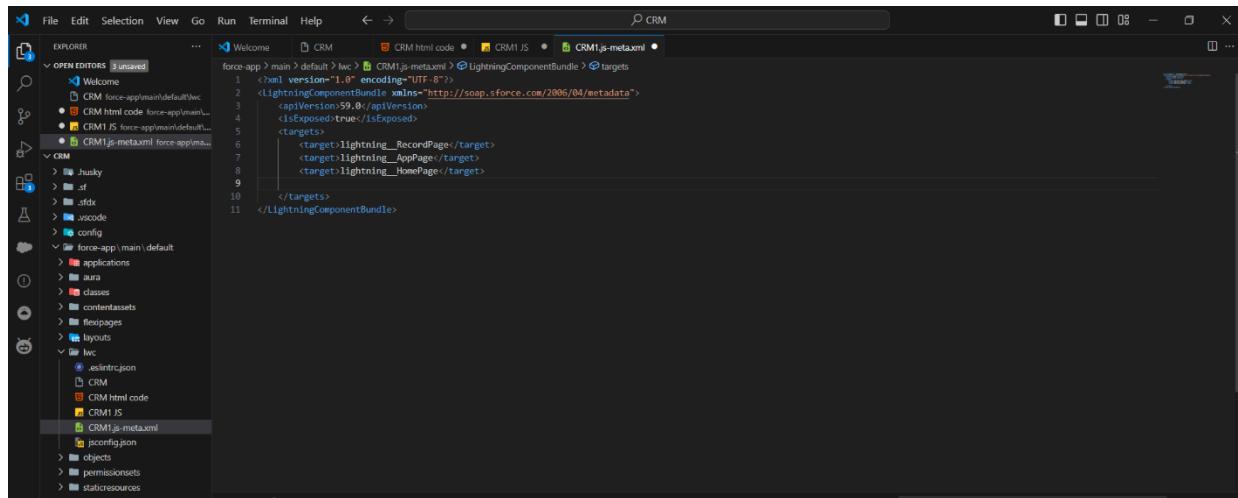
The main editor area displays the 'CRM1 JS' file content:

```
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    @wire(getRecord, { recordId: "$userId", fields: ['User.Verified_c'] })
25    recordFunction({ data, error }) {
26        if (data) {
27            console.log(data)
28            console.log("This is the User Id ---> "+this.userId);
29            this.verifiedvar = data.fields.Verified_c.value;
30        } else {
31            console.error(error)
32            console.log('this is error')
33        }
34    }
35}
```

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

## Code :-

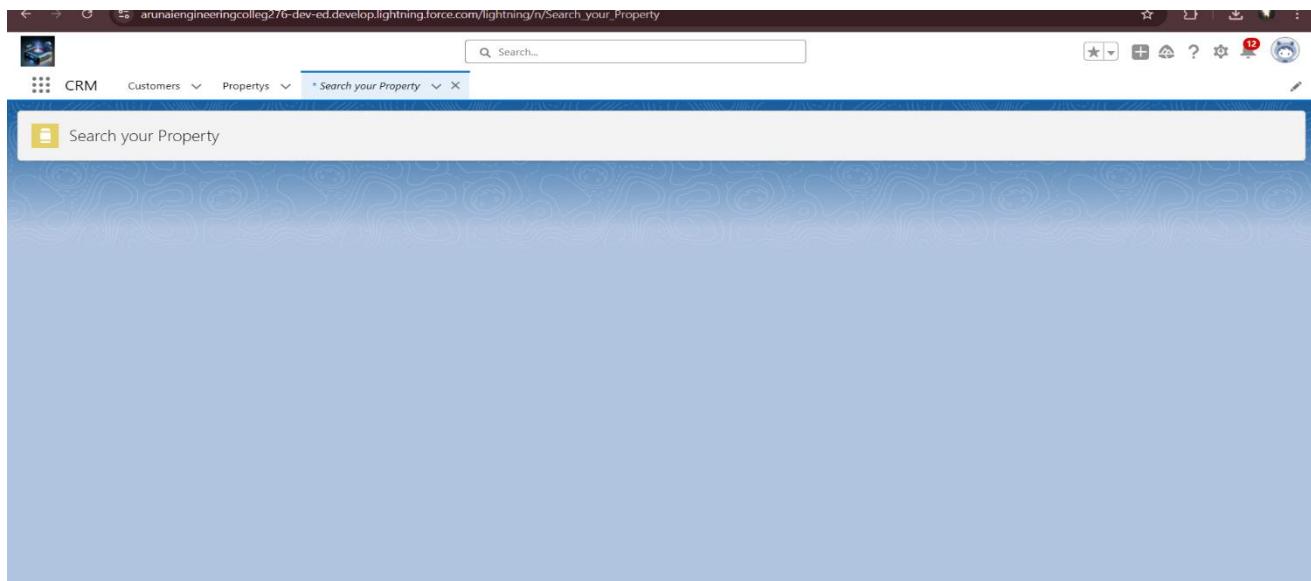
```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle
    xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>59.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning_RecordPage</target>
        <target>lightning_AppPage</target>
        <target>lightning_HomePage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



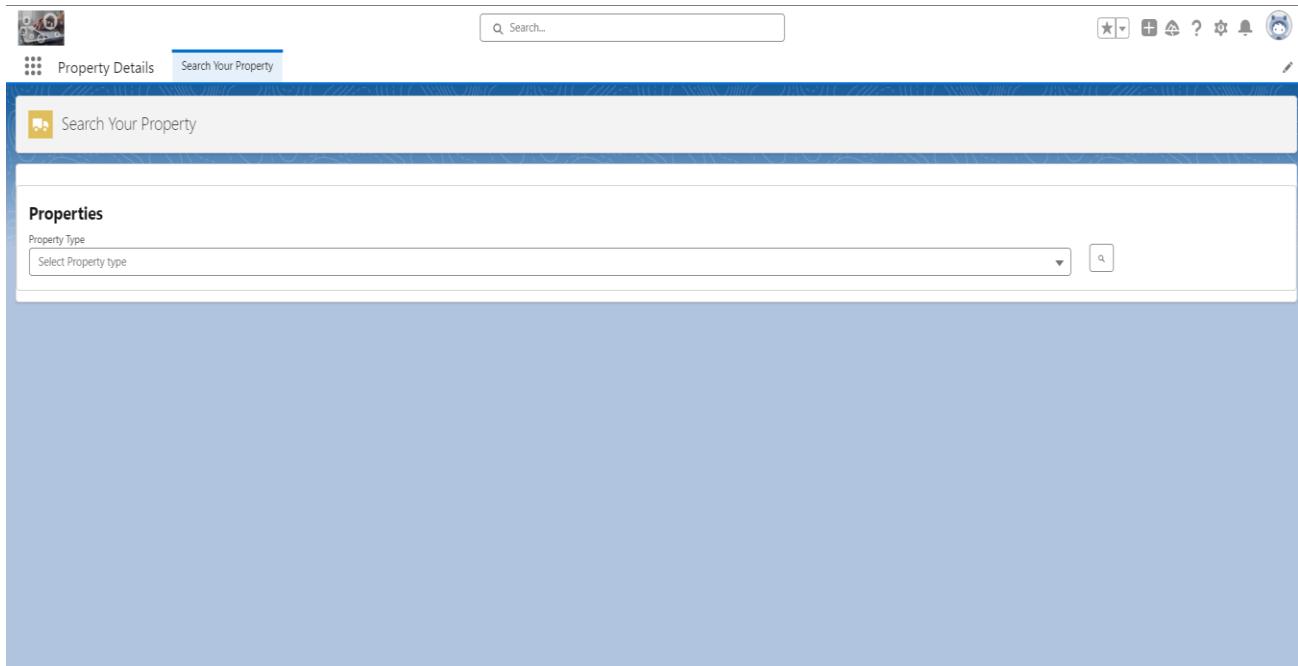
- 7) After Saving all the three Codes , Right Click and deploy this component to the org.

# 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 the Component 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”.
-