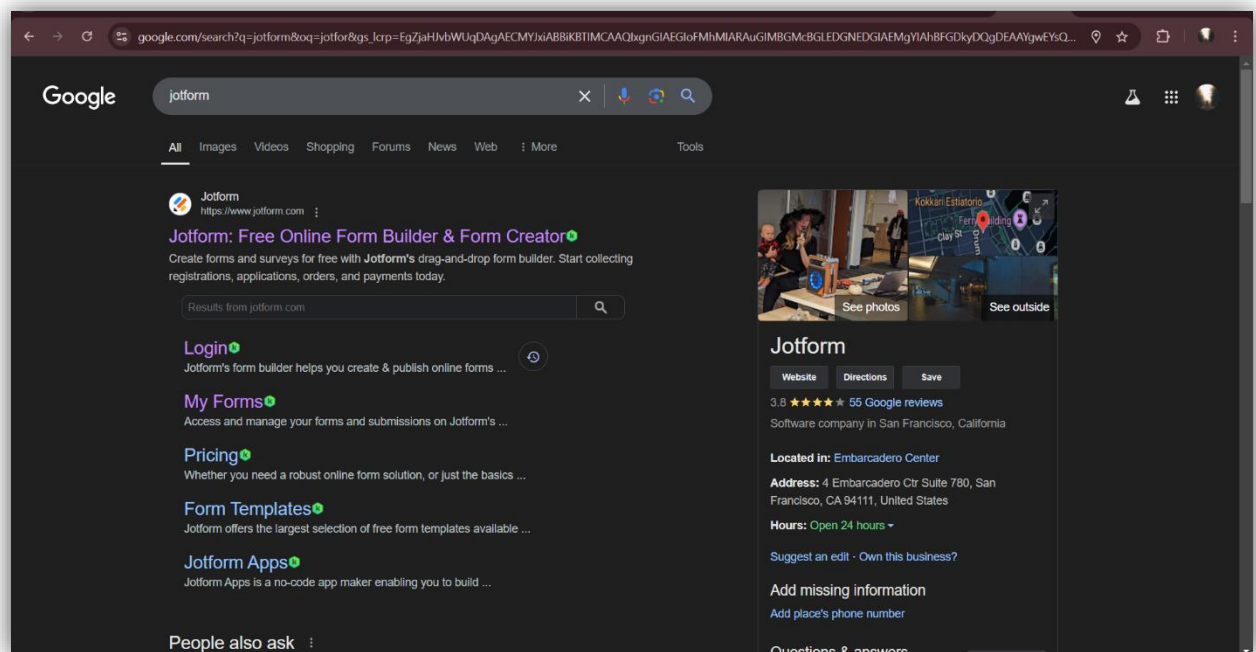


A CRM Application to Handle the Clients and their property Related Requirements

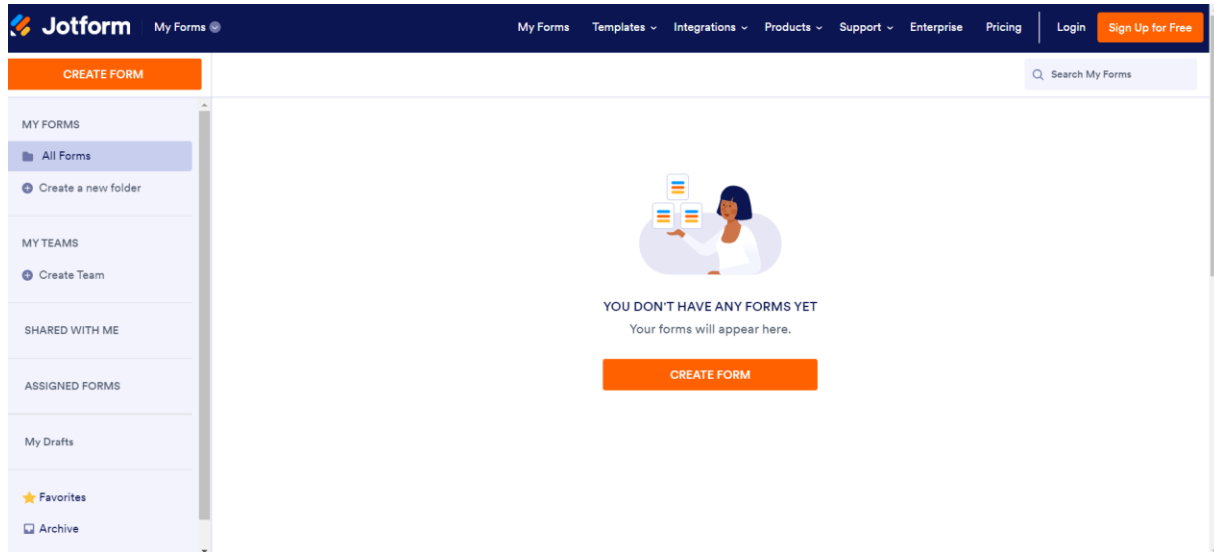
Milestone 1 :- Create a jotform and integrate it with the org to create a record of customers automatically.

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

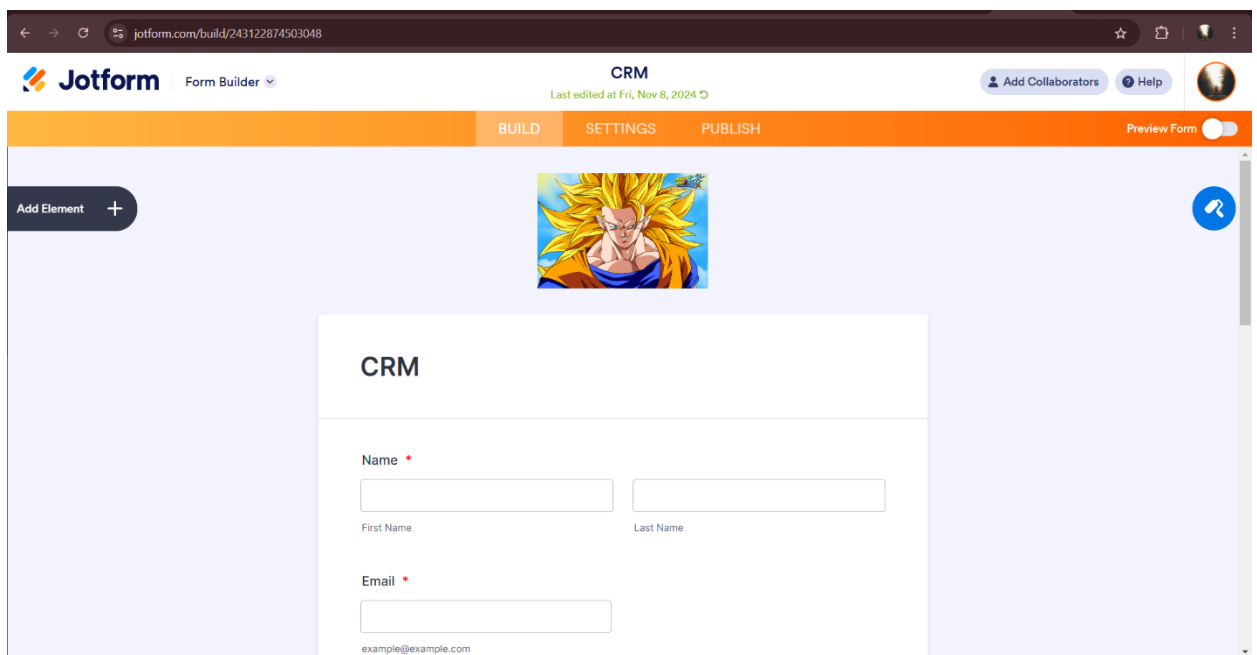


3) Now create a form to get the customer details like Name, Phone, Email, Address and type of property the customer is interested in.

4)

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

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



The screenshot displays the Jotform Form Builder interface for a CRM form. The browser address bar shows the URL `jotform.com/build/243122874503048`. The Jotform logo and 'Form Builder' text are in the top left. The title 'CRM' is centered at the top, with a note 'Last edited at Fri, Nov 8, 2024'. On the top right, there are buttons for 'Add Collaborators' and 'Help', and a user profile icon. Below the title bar, there are three tabs: 'BUILD' (active), 'SETTINGS', and 'PUBLISH'. A 'Preview Form' toggle switch is on the far right. On the left side, there is a dark button labeled 'Add Element' with a plus icon. The main workspace has a light blue background. At the top center of the workspace is a small image of Super Saiyan Goku. In the center, there is a white form box titled 'CRM'. The form contains three fields: 'Name' (required, marked with a red asterisk) which is split into 'First Name' and 'Last Name' input boxes, and 'Email' (required, marked with a red asterisk) which is a single input box. At the bottom of the form box, the email address 'example@example.com' is displayed.

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.

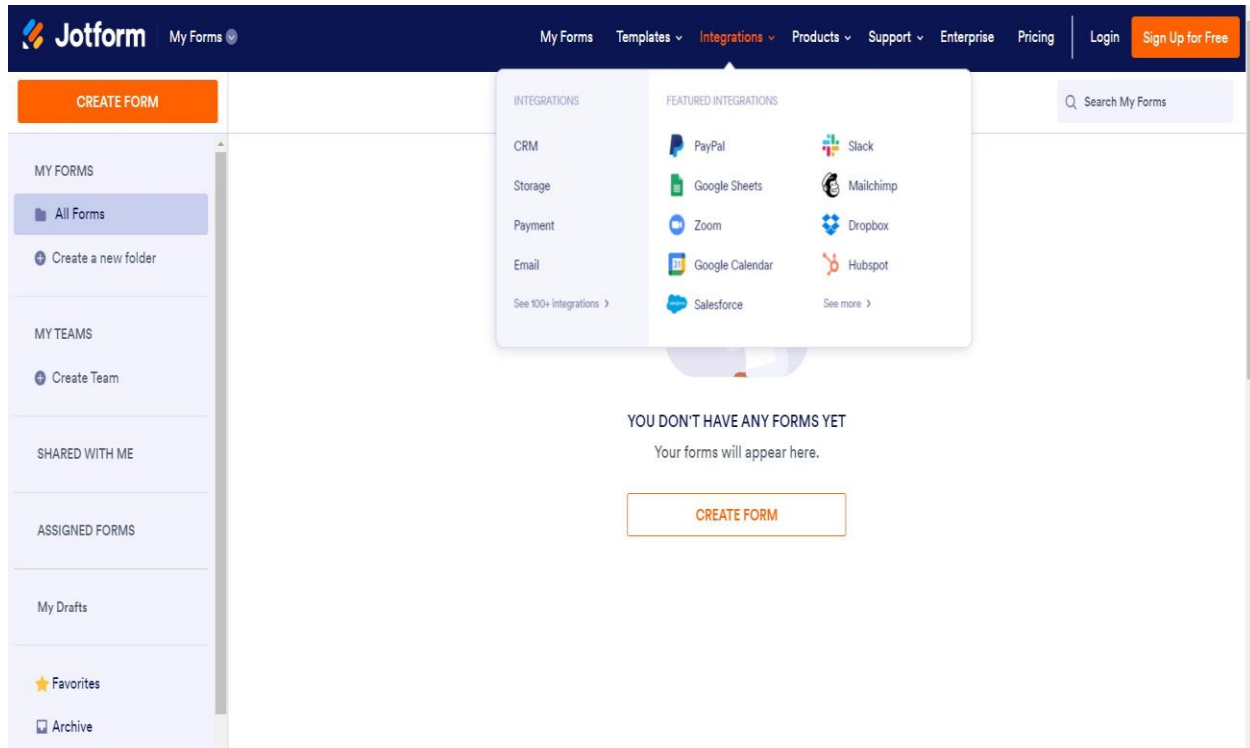
- **Create Property object**

- 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 top navigation bar includes "Jotform", "Form Builder", and "Dreams World" with a status "All changes saved at 2:43 PM". On the right, there are buttons for "Add Collaborators", "Help", and a user profile icon. Below the navigation bar is a tabbed interface with "BUILD", "SETTINGS", and "PUBLISH". The "SETTINGS" tab is active, showing a sidebar with various settings categories: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (selected), APPROVAL FLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The main content area is titled "Salesforce via this handy integration." and includes a list of actions: "Add new company leads", "Add new contacts", "Add new accounts", and "Connect any custom object". Below this, there is a section for "Authentication" 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 at the bottom right.

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

The screenshot shows the Jotform Form Builder interface for a form named "Dreams World". The top navigation bar includes "Jotform", "Form Builder", and "Dreams World" with a status "All changes saved at 2:43 PM". On the right, there are buttons for "Add Collaborators", "Help", and a user profile icon. Below the navigation bar is a tabbed interface with "BUILD", "SETTINGS", and "PUBLISH". The "SETTINGS" tab is active, showing a sidebar with various settings categories: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (selected), APPROVAL FLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The main content area is titled "Select a Salesforce Object" and includes a dropdown menu labeled "Select a Salesforce Object" with "Customer" selected. Below this, there is a section for "Object Fields" with a dropdown menu labeled "Select field" and a text input field containing "Customer". A "Send Feedback" button is located at the bottom right.

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

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

The screenshot shows the Jotform CRM integration settings page. The browser address bar displays `jotform.com/build/243122874503048/settings/integrations`. The page header includes the Jotform logo, "Form Builder", the title "CRM", and a timestamp "Last edited at Fri, Nov 8, 2024". Navigation tabs for "BUILD", "SETTINGS", and "PUBLISH" are present, along with a "Preview Form" toggle. A left sidebar lists various settings: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (highlighted), WORKFLOWS, and JOTFORM SIGN. The main content area is titled "SALESFORCE" and includes a description: "Send new leads, contacts, or accounts to your sales CRM". It features a "Select a Salesforce Object" dropdown set to "Customer". Below this is a "Create a record" section with the instruction "Send data from form fields to matched Salesforce fields". A table maps form fields to Salesforce CRM fields:

Object Fields	CRM
Name	Name
Email	Email
Phone Number	Phone Number
Property Type	Which Type of Property are you lookin...
Budget Amount	Budget Amount

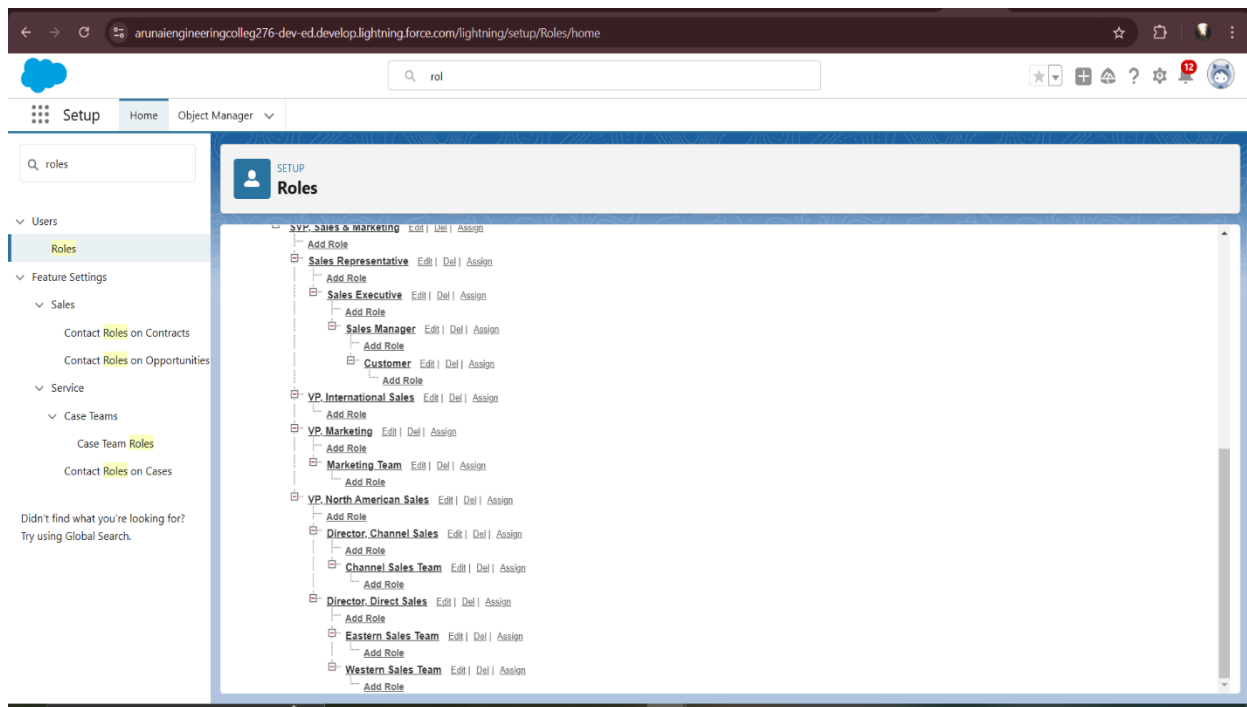
A "Give Feedback" button is located at the bottom right of the settings area.

Milestone 4 : Create Roles

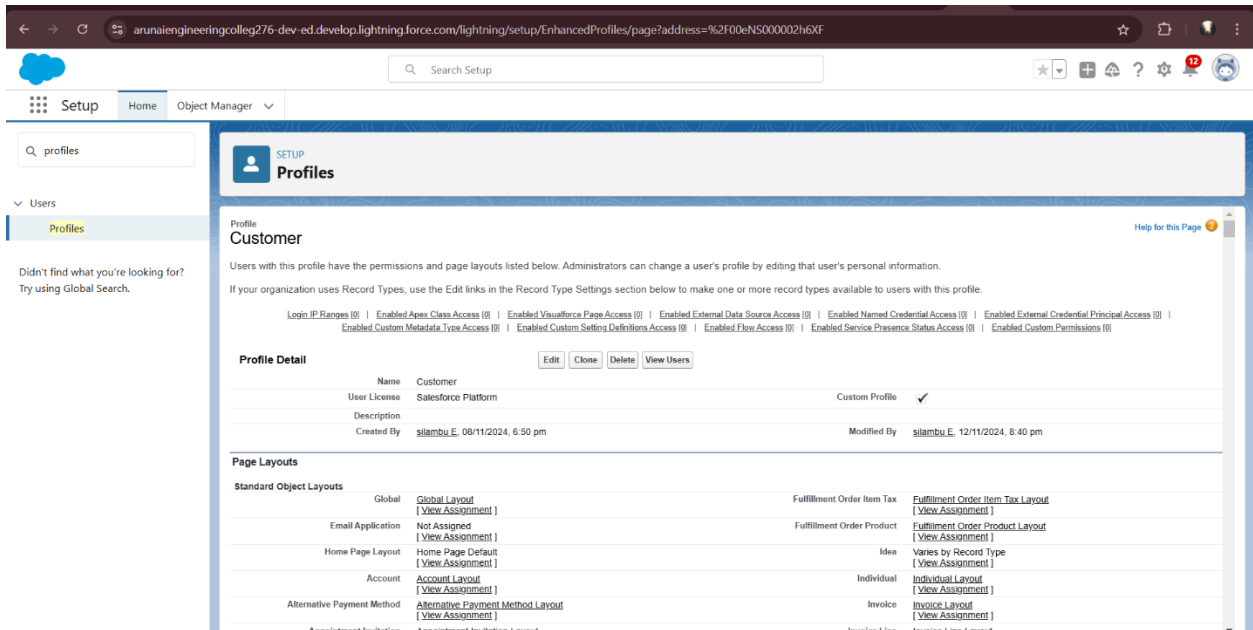
- Sales Executive Role

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

* It will use the “System Administrator Profile”.



2) Label - Sales Executive Reports to - Sales Representative



The screenshot shows the Salesforce Setup interface for the 'Customer' profile. The left sidebar contains a search bar and a list of setup areas, with 'Profiles' selected. The main content area displays the 'Customer' profile details, including its name, user license, and a table of page layouts.

Profile Detail

Name	Customer
User License	Salesforce Platform
Description	
Created By	silambu.E, 08/11/2024, 6:50 pm
Modified By	silambu.E, 12/11/2024, 8:40 pm

Page Layouts

Standard Object Layouts	Global	Fulfillment Order Item Tax	Fulfillment Order Item Tax Layout
Email Application	Not Assigned	Fulfillment Order Product	Fulfillment Order Product Layout
Home Page Layout	Home Page Default	Idea	Varies by Record Type
Account	Account Layout	Individual	Individual Layout
Alternative Payment Method	Alternative Payment Method Layout	Invoice	Invoice Layout
Announcement Invitation	Announcement Invitation Layout	Invoice Line	Invoice Line Layout

- 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.

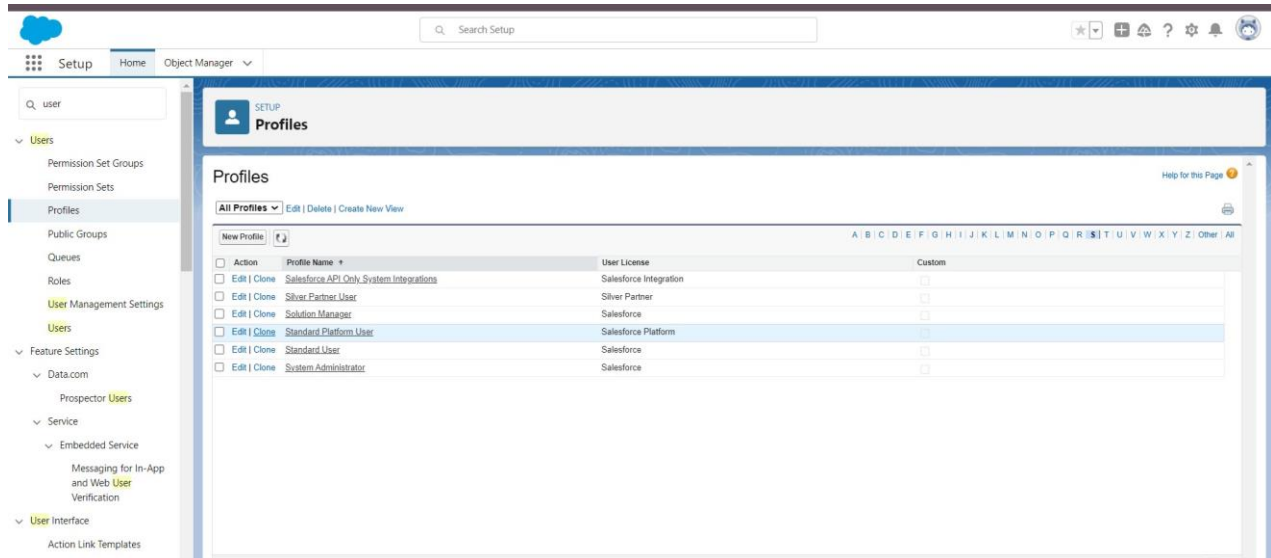
The screenshot shows the Lightning App Builder interface in a web browser. The browser's address bar displays a URL from 'anunaiengineering.colleg276-dev-ed.develop.lightning.force.com'. The top navigation bar includes tabs for 'Lightning App Builder', 'App Settings', 'Pages', and 'Property Details'. On the left, the 'App Settings' sidebar is expanded, showing options like 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', 'Navigation Items', and 'User Profiles'. The main content area is titled 'App Details & Branding' and includes instructions: 'Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.' The 'App Details' section contains input fields for 'App Name' (filled with 'Property Details'), 'Developer Name' (filled with 'Property_Details'), and a 'Description' field. The 'App Branding' section features an 'Image' upload area with an 'Upload' button and a 'Primary Color Hex Value' dropdown set to '#0070D2'. Below this, the 'Org Theme Options' section has a checkbox labeled 'Use the app's image and color instead of the org's custom theme'. At the bottom, the 'App Launcher Preview' shows a blue button with 'PD' and a grey button with 'Property Details'.

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

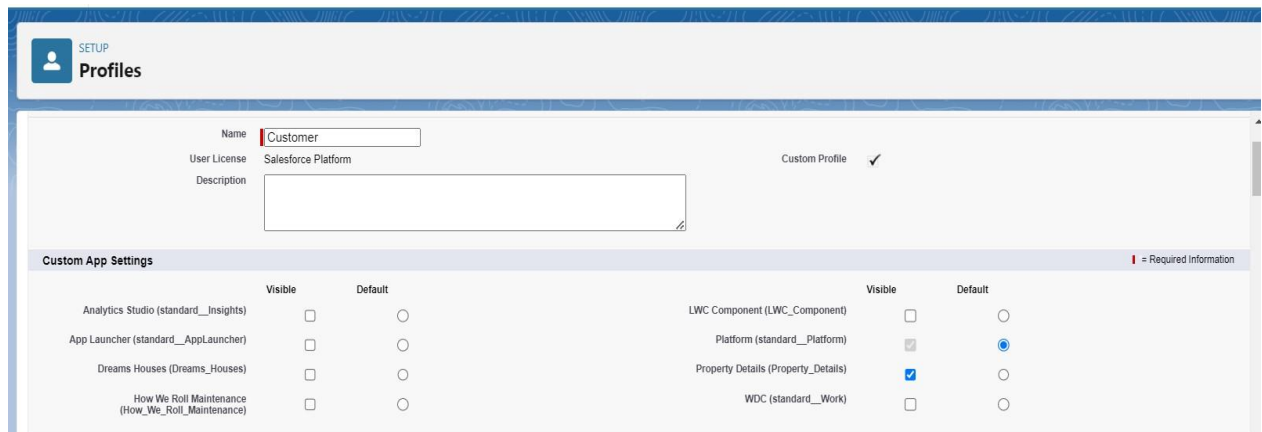
Milestone 6 : - Create Profiles

- **Customer : -**

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



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



3) Also Remove all the Standard Object Permissions.



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”

[illegible]

- **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 I = 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

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_lsg	31/01/2024, 11:22 am	Modified By
			data_lsg, 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

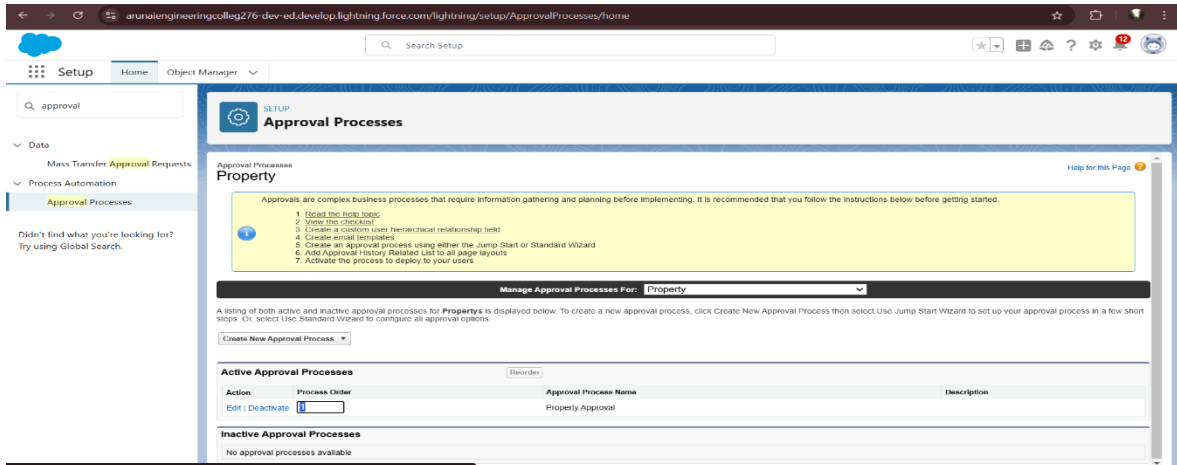
User 4 : -

- 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

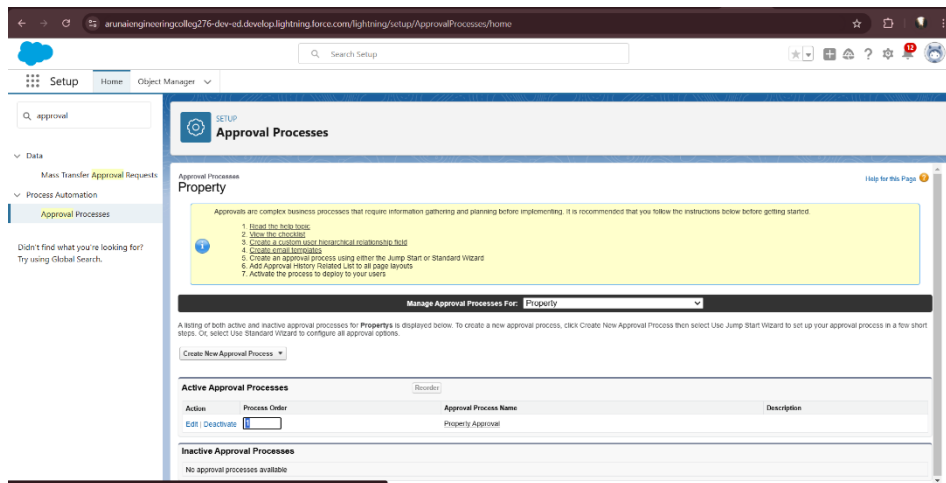
- 1) From Setup → Process Automation → Approval Process

2) Process Name - Property Approval



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

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

Up

 Down

[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	<input type="text" value="VP Approval"/>
Unique Name	<input type="text" value="VP_Approval"/>
Description	<input type="text"/>

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

SETUP

Edit Field Update
Unverified 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 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

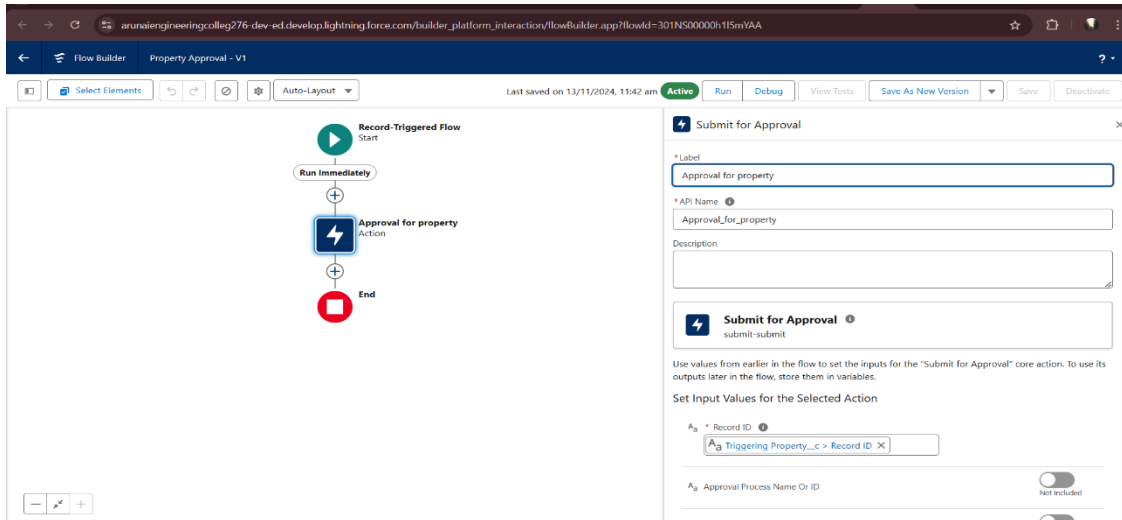
True False

Save Save & New Cancel

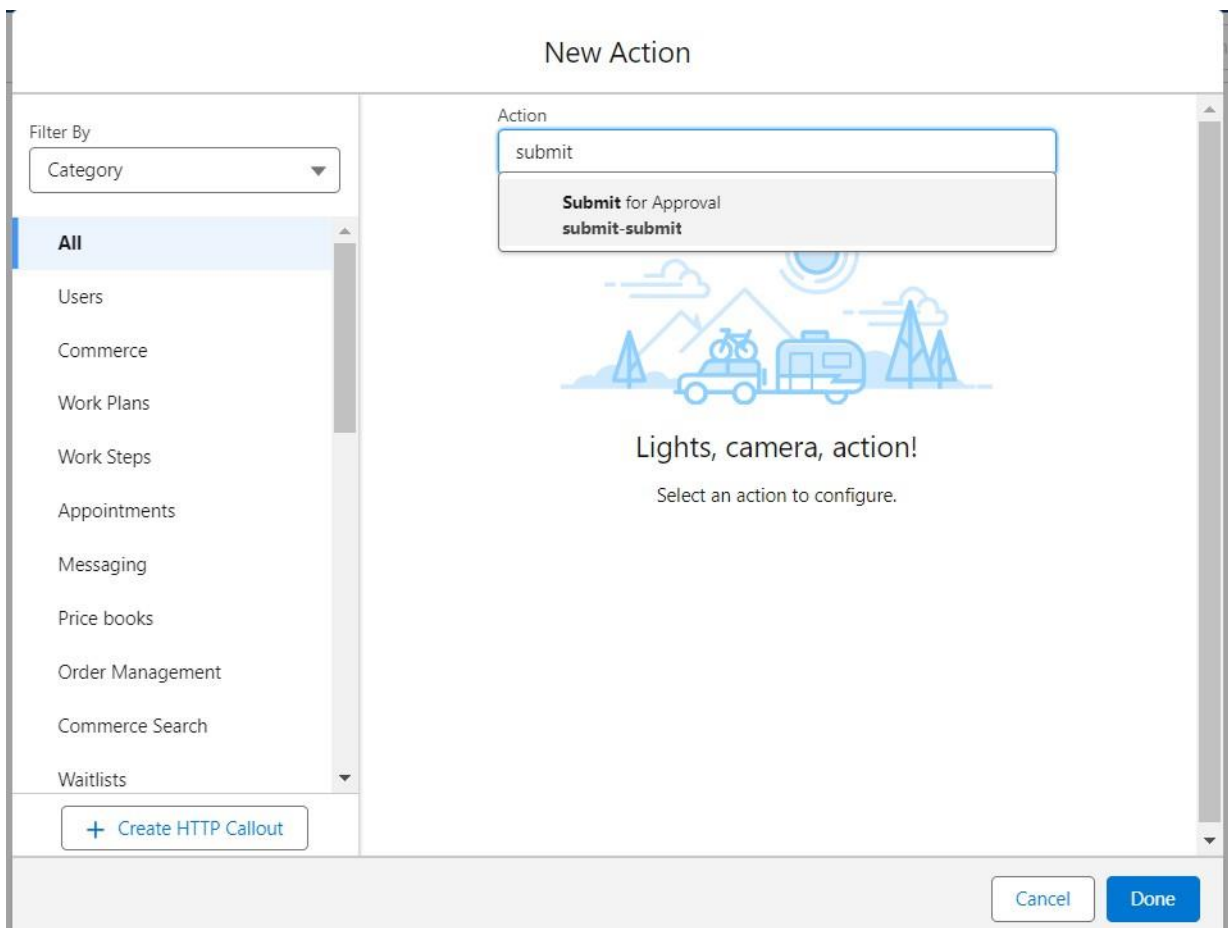
14) Activate the Approval Process.

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

New Action

Filter By
Category

All

- Users
- Commerce
- Work Plans
- Work Steps
- Appointments
- Messaging
- Price books
- Order Management
- Commerce Search
- Waitlists

+ Create HTTP Callout

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

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

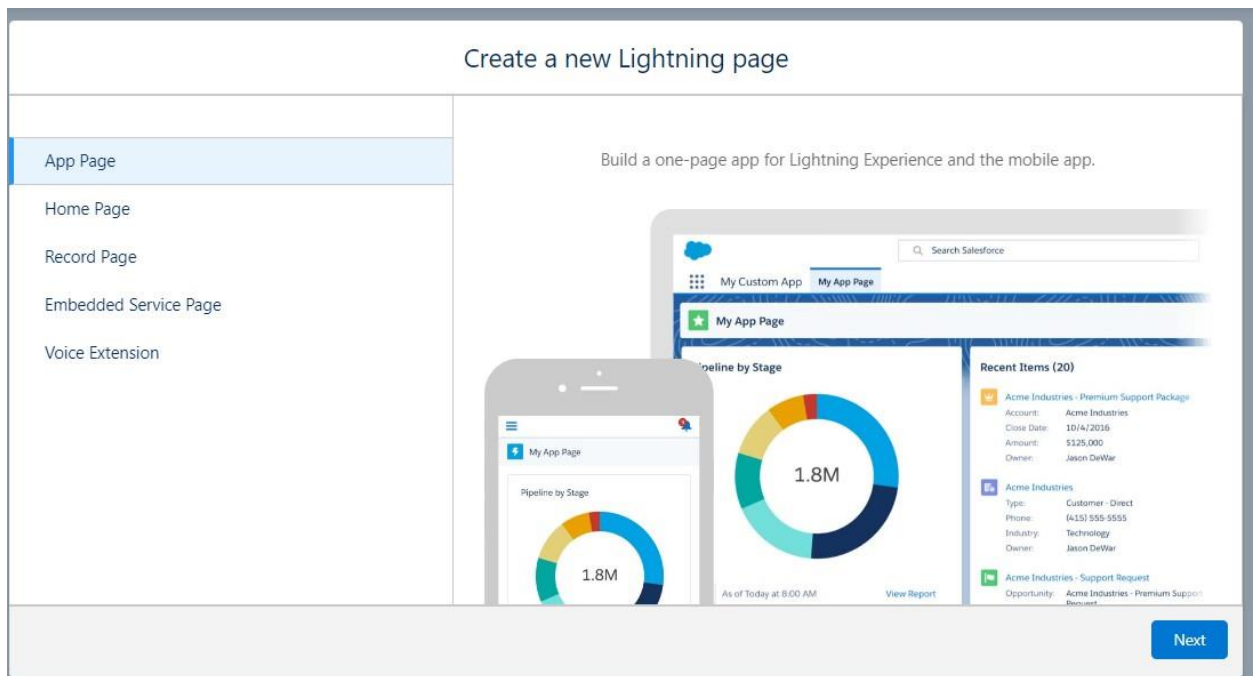
Show Advanced

Cancel Save

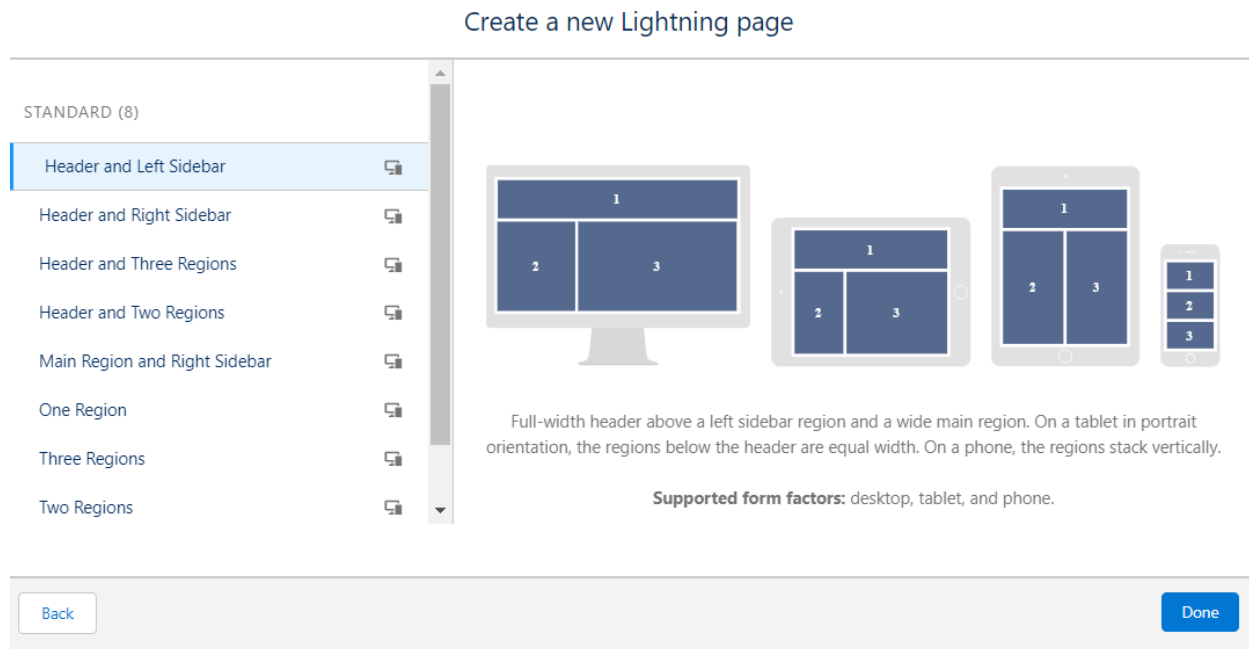
Milestone 11 :- Create an App Page

- Create an App Page on the Property details Object named as “Search Your Property”

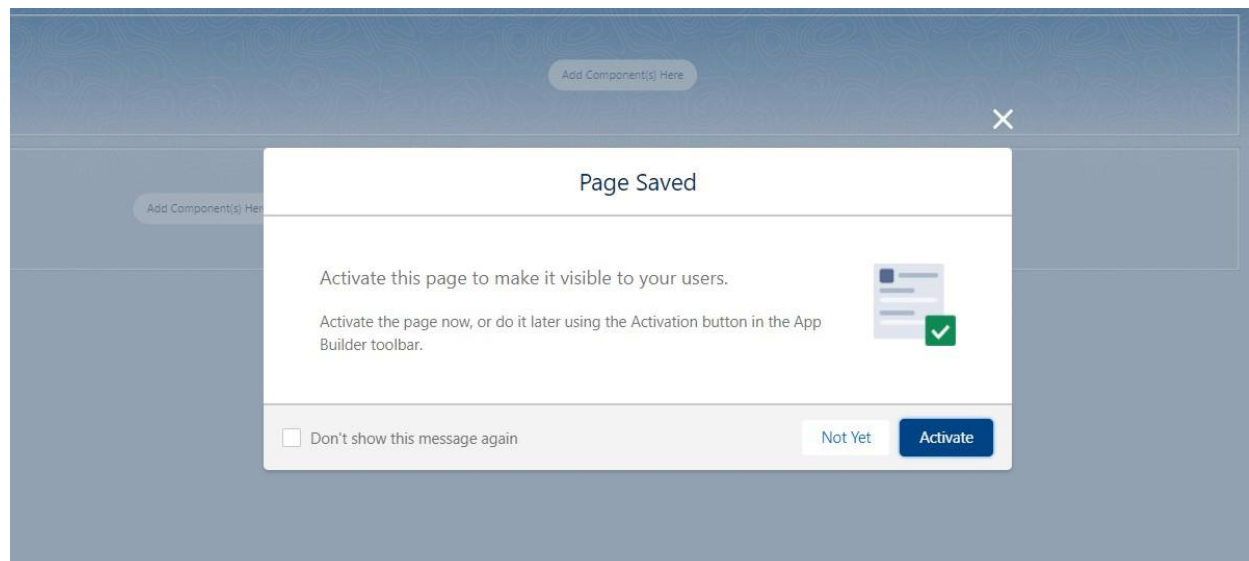
1) From Setup → Go to Lightning App Builder → Click on New → Select App Page and Click on Next.



- 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

Remove page

Search Your Property

Search your property

Cancel

Save

7) Then Click on “Save”

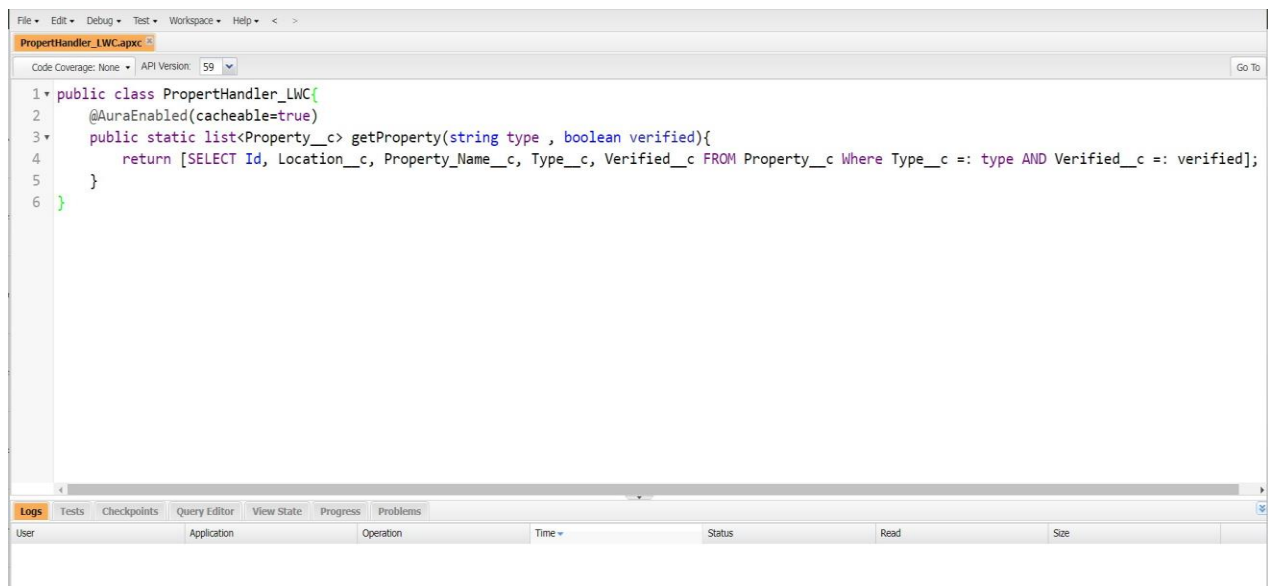
Milestone 12 :- Create a LWC Component

- 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];
    }
}
```



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>

  </div>

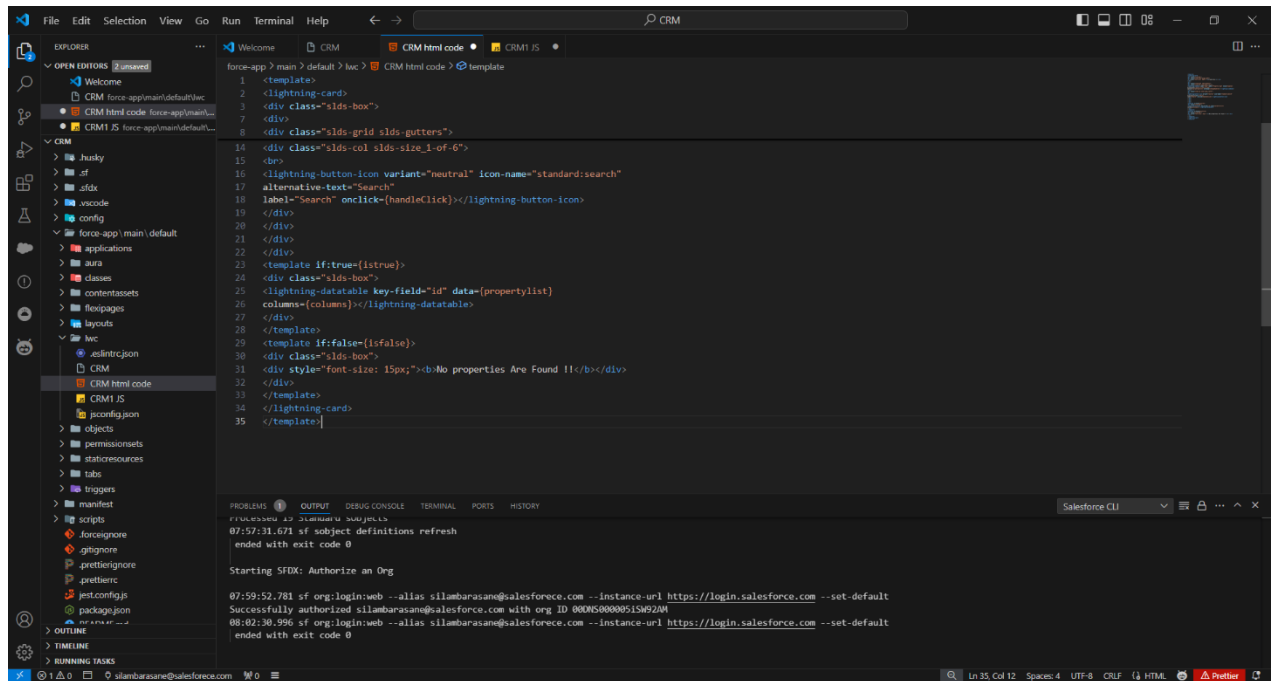
  <template if:true={isttrue}>
    <div class="slds-box">
      <lightning-datatable key-field="id" data={propertylist}
columns={columns}></lightning-datatable>
    </div>
  </template>
  <template if:false={isfalse}>
```

```

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

</lightning-card>
</template>

```



6) In Your Js File Write this code : -

Code :-

```

import { LightningElement, api, track, wire } from 'lwc';
import getProperty from '@salesforce/apex/PropertyHandler_LWC.getProperty';
import { getRecord } from 'lightning/uiRecordApi';
import USER_ID from '@salesforce/user/Id';
export default class C_01_Property_Management extends LightningElement {
  @api 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() {

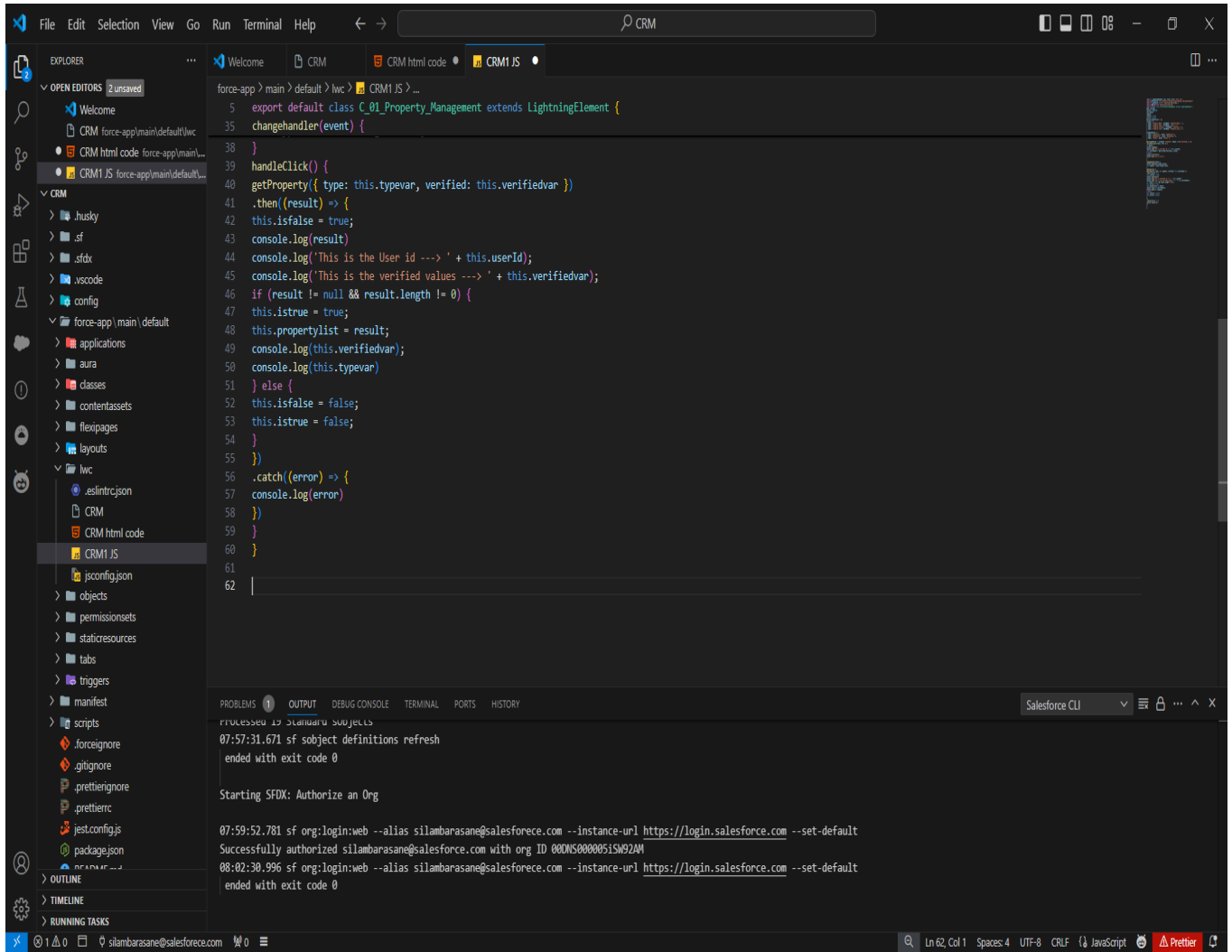
    getProperty({ type: this.typevar, verified: this.verifiedvar })
        .then((result) => {
            this.isfalse = true;
            console.log(result)
            console.log('This is the User id ---> ' + this.userId);
        })
}

```

```
    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)
})
}

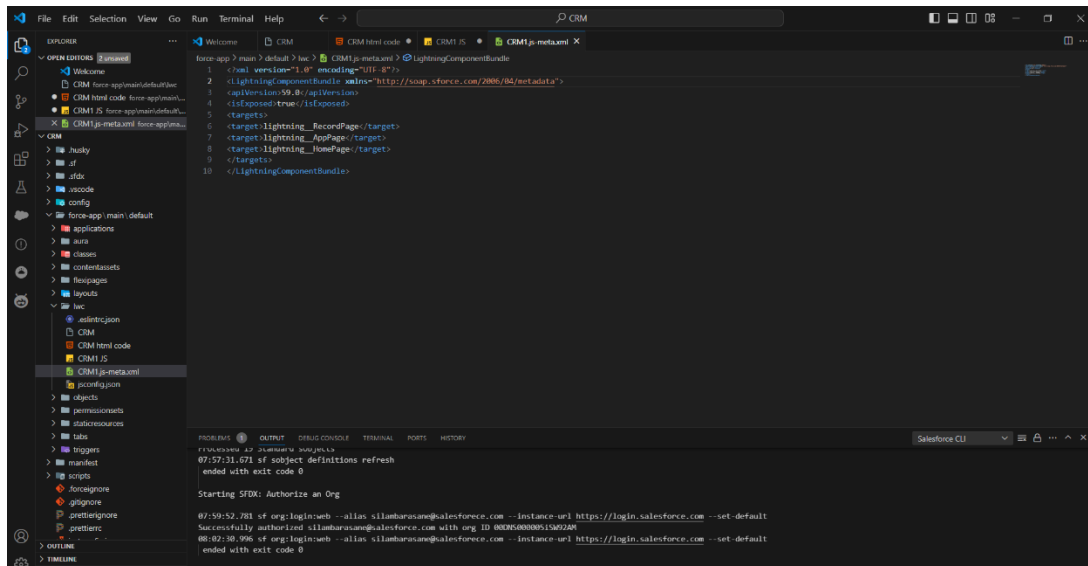
}
```



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

Code :-

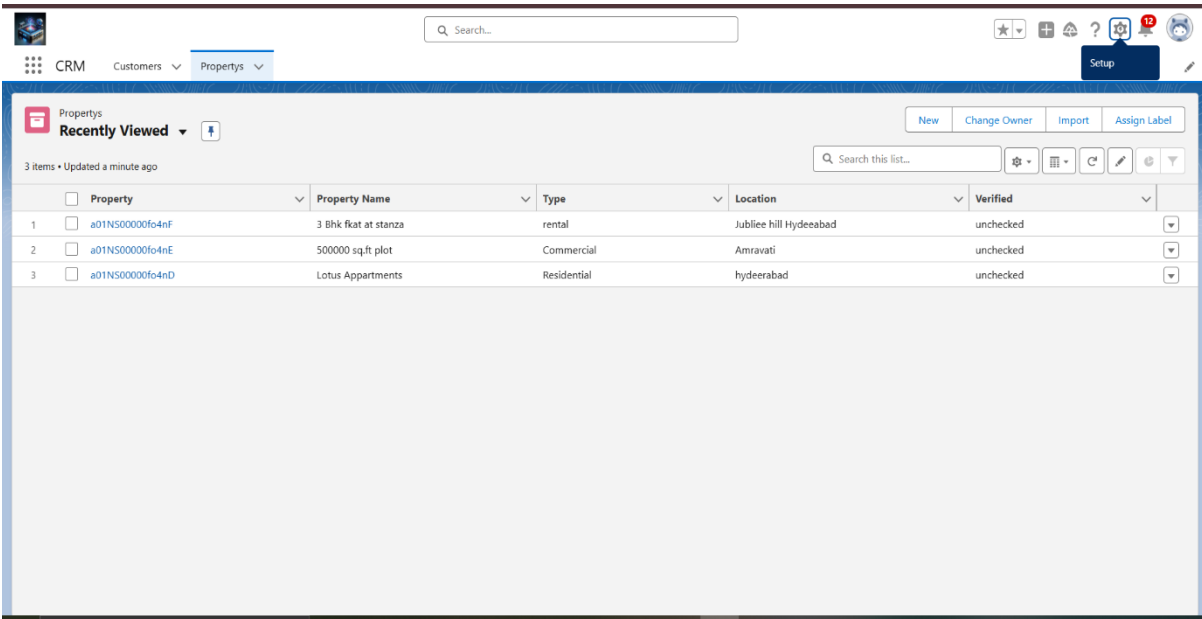
```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>59.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__RecordPage</target>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage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

