SALESFORCE VIRTUAL INTERSHIP

A CRM Application To Handle The Clients And Their Property Related Requirements

Introduction

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.

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

Client wants a form for the customers to get the details directly into the salesforce so that the admins can

Step 1: Create the Form in JotForm

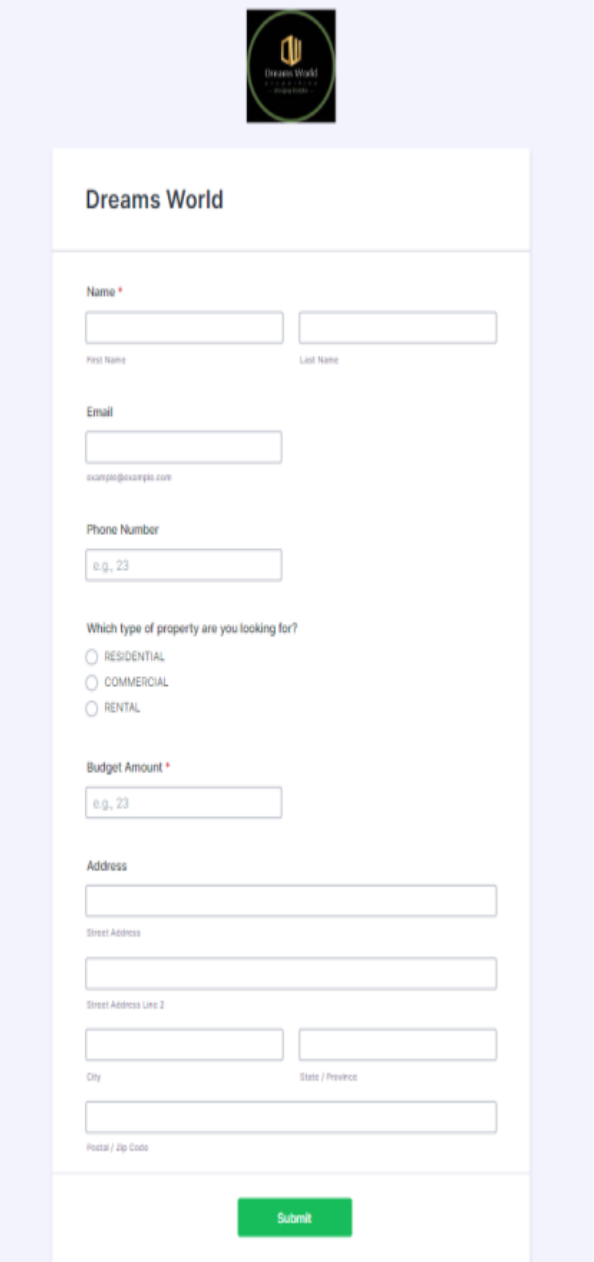
1. Sign up or log in to JotForm: Go to JotForm and sign in to your account.

2. Create a new form: Click on the "Create Form" button and choose a blank form or a template that suits your needs.

3. Add fields to the form: Add the necessary fields to collect customer information, such as Name, Email, Type of Property, Budget Amount, Address and Phone Number.

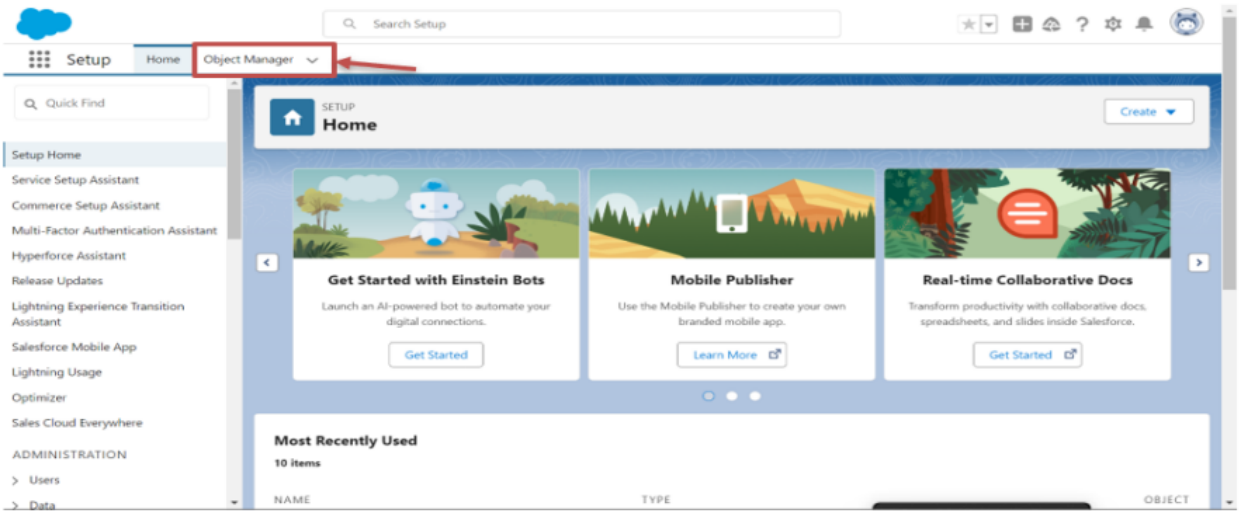
4. Once the form is created, publish it by clicking on publish.

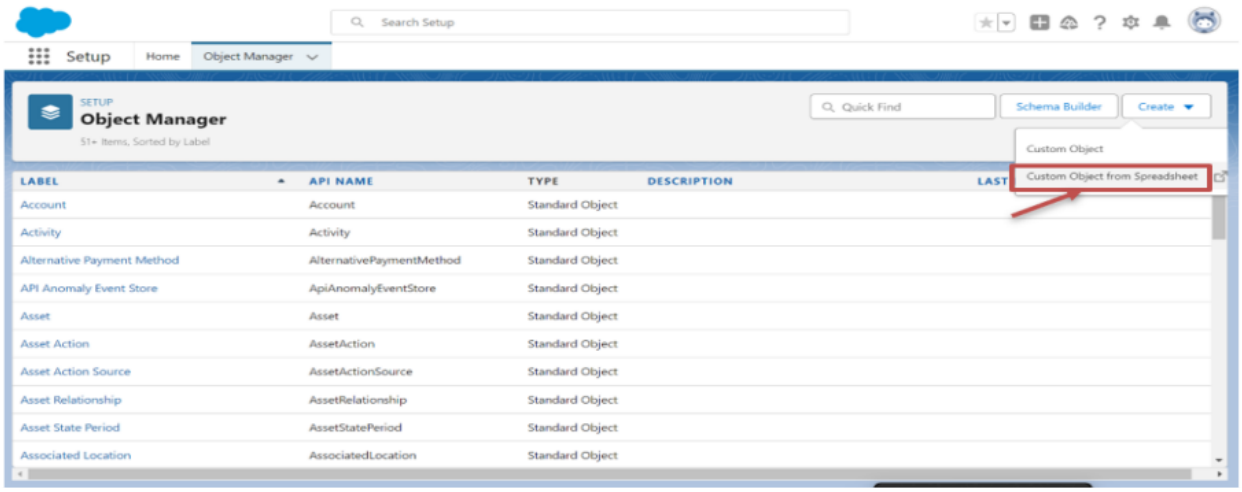
5. The JotForm created link: https://form.jotform.com/242124450785051

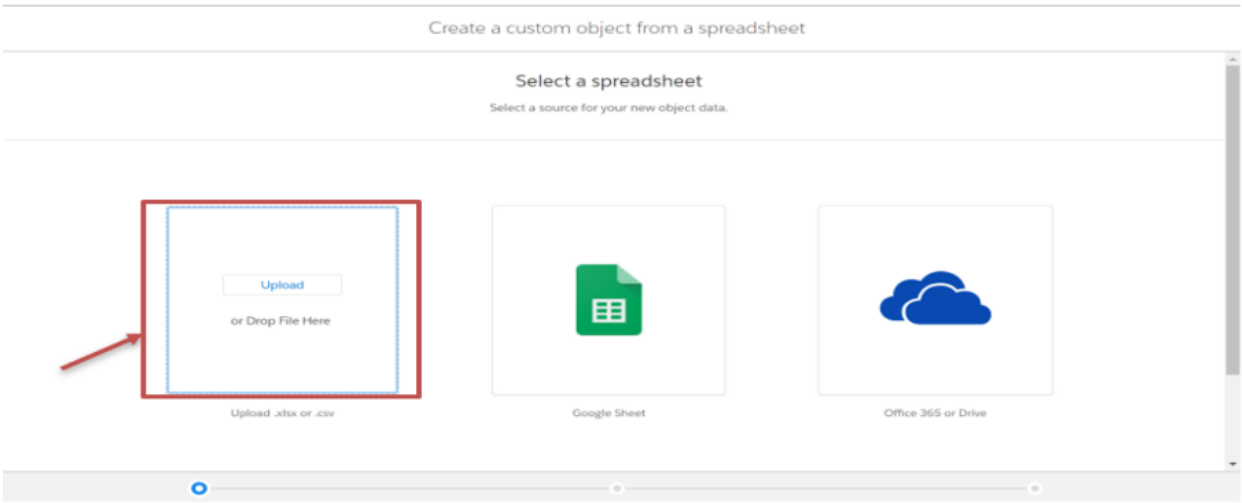


Activity 1: Create Objects from Spreadsheet.

Directly Creating Objects from Spreadsheet in Salesforce



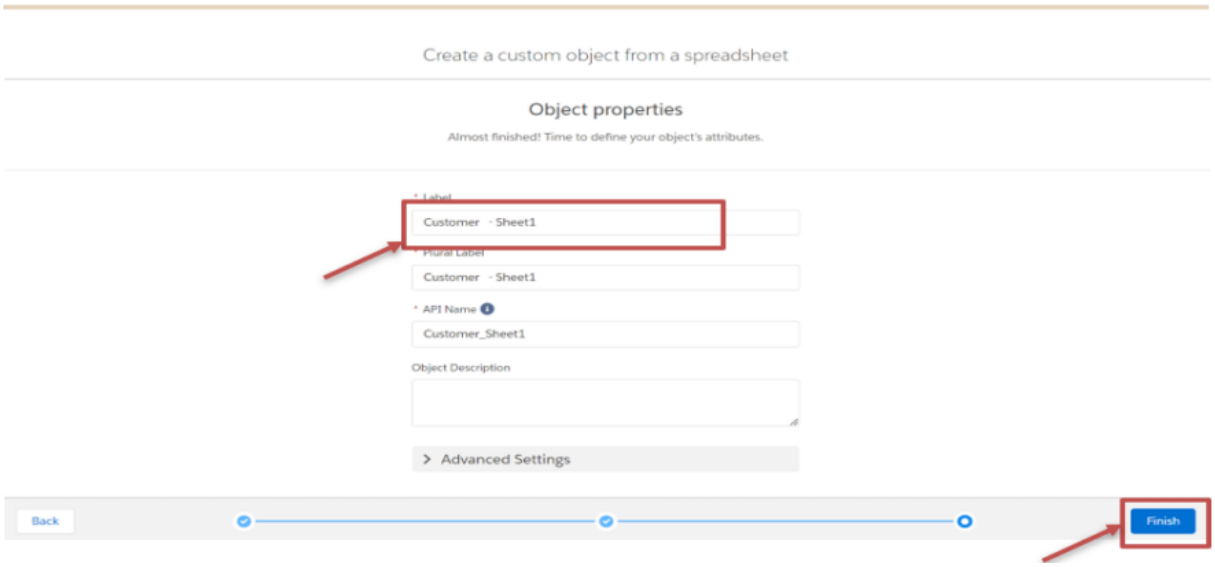


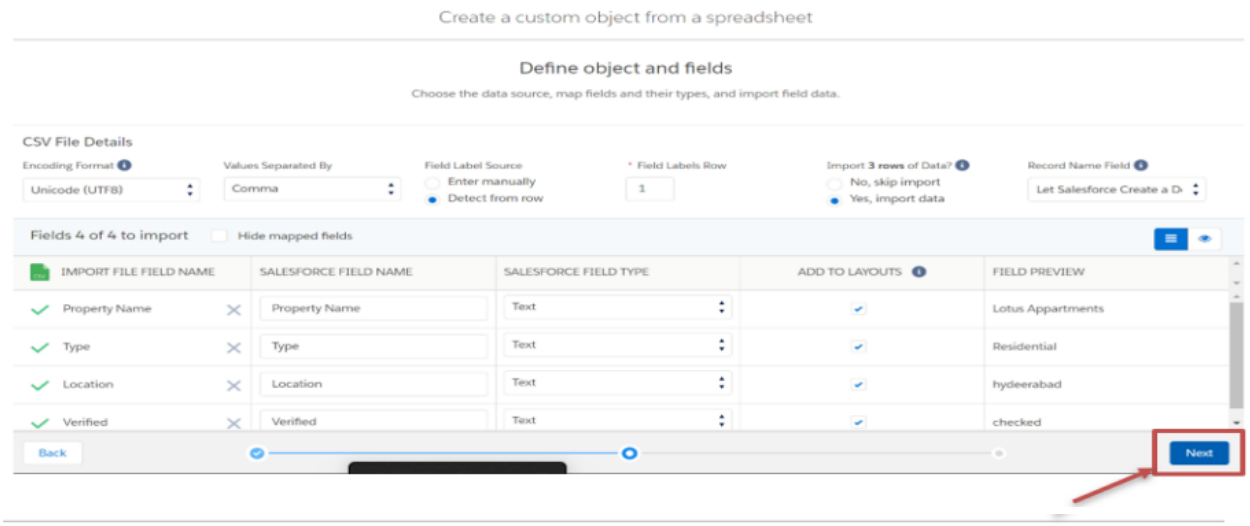


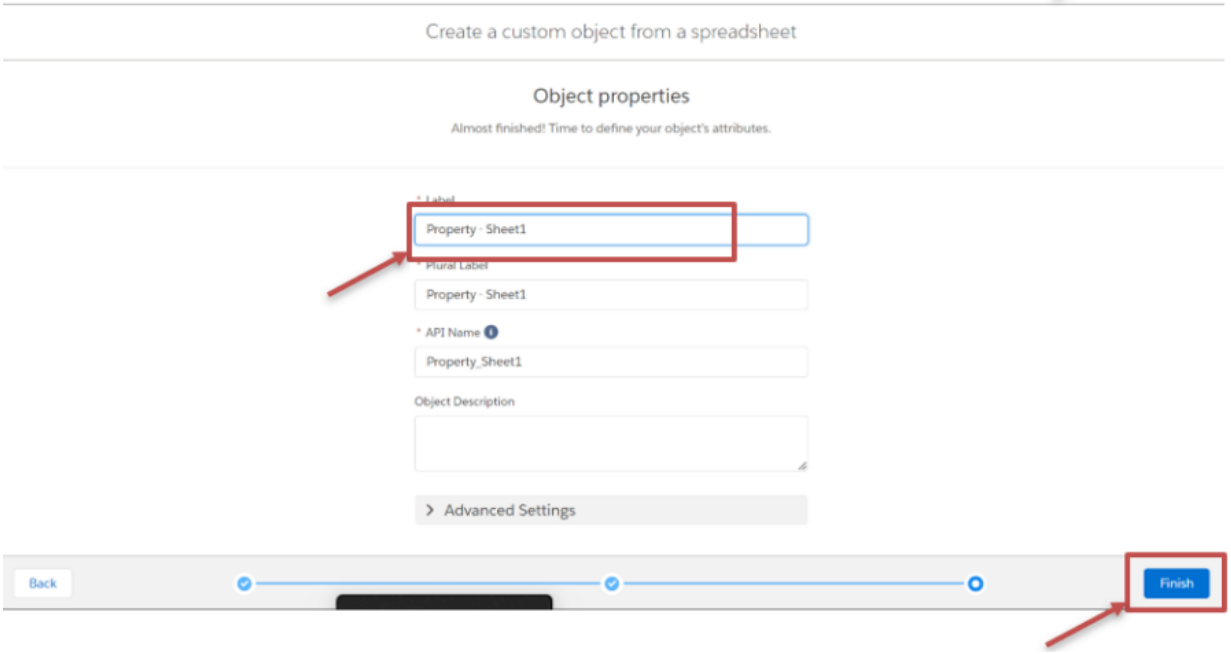
Create Customer object

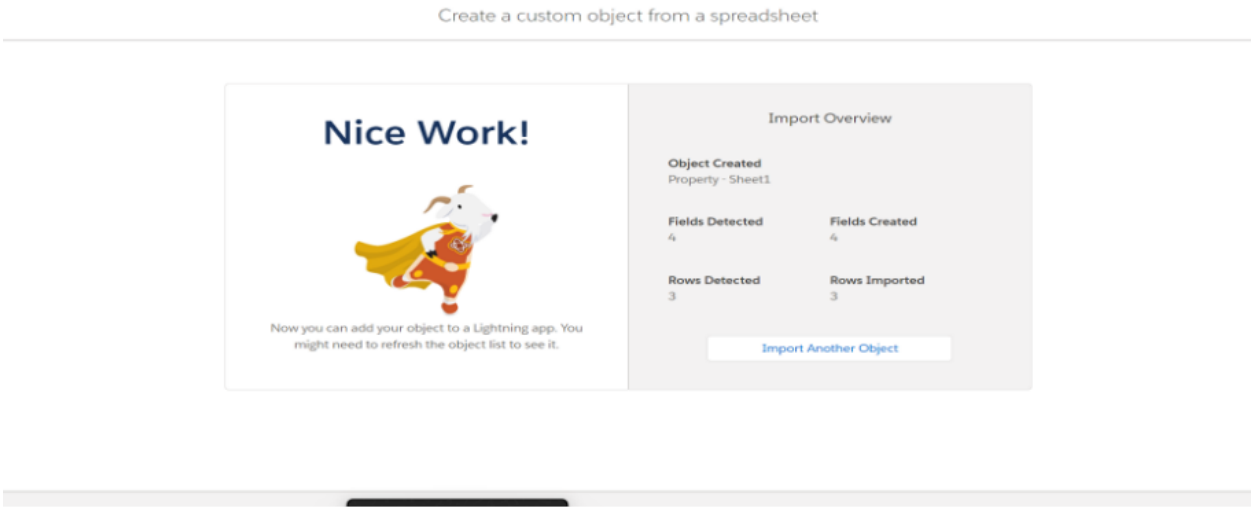
For creating the customer object follow the steps:

1. Go to the object manager and click on create object from spreadsheet.
2. Download the customer spreadsheet provided Customer.
3. After downloading, upload the file and map the fields and upload to create an object.




Create a custom object from a spreadsheet
Define object and fields
Choose the data source, map fields and their types, and import field data.
CSV File Details
Encoding Format
Unicode (UTF8)
Values Separated By
Comma
Field Label Source
Enter manually
Detect from row
* Field Labels Row
1
Import 3 rows of Data?
No, skip import
Yes, import data
Record Name Field
Let Salesforce Create a D
Fields 4 of 4 to import
Hide mapped fields
IMPORT FILE FIELD NAME
SALESFORCE FIELD NAME
SALESFORCE FIELD TYPE
ADD TO LAYOUTS
FIELD PREVIEW
Property Name
Property Name
Text
Lotus Appartments
Type
Type
Text
Residential
Location
Location
Text
hydeerabad
Verified
Verified
Text
checked
Back
Next

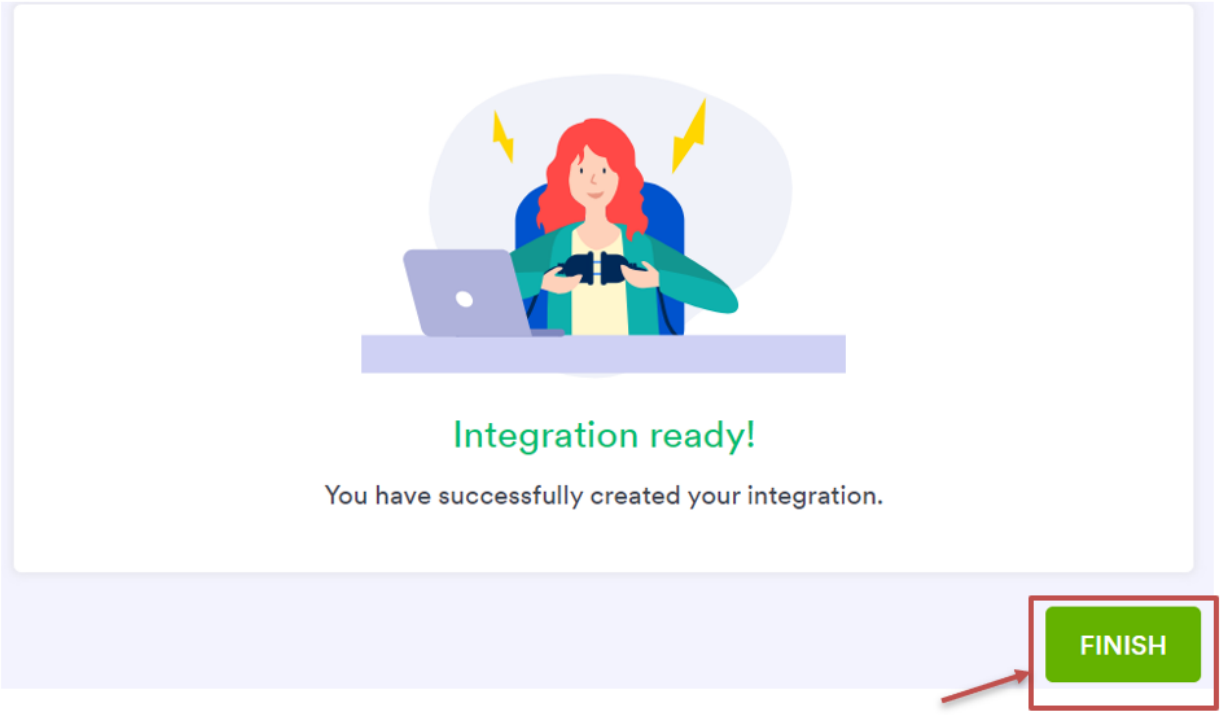

Create a custom object from a spreadsheet
Object properties
Almost finished! Time to define your object's attributes.
* Label
Property - Sheet1
* Plural Label
Property - Sheet1
* API Name
Property_Sheet1
Object Description
Advanced Settings
Back
Finish


Create a custom object from a spreadsheet
Nice Work!
Now you can add your object to a Lightning app. You might need to refresh the object list to see it.
Import Overview
Object Created
Property - Sheet1
Fields Detected
4
Fields Created
4
Rows Detected
3
Rows Imported
3
Import Another Object

Integrate Jotform with Salesforce Platform

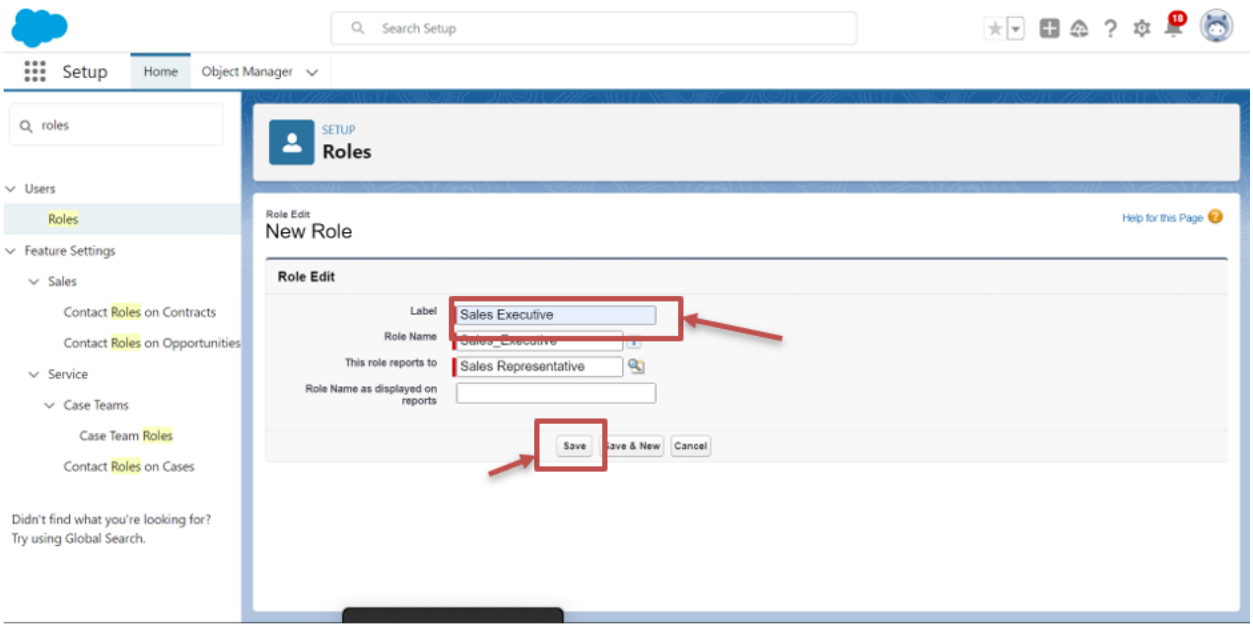
In this Milestone we are going to integrate jotform with Salesforce

1. On the Jotform Platform, Click on Integration and choose Salesforce.
2. Click on User Integration and choose “Add to Form”.
3. Select the Org with which you want to Integrate your JotForm with.
4. Select an Action.
5. Select a Salesforce Object: - Customer
6. Map Each and every field on the Object with the fields on the form and “Save Action”.
7. Then “Save the Integration” and “Finish”.

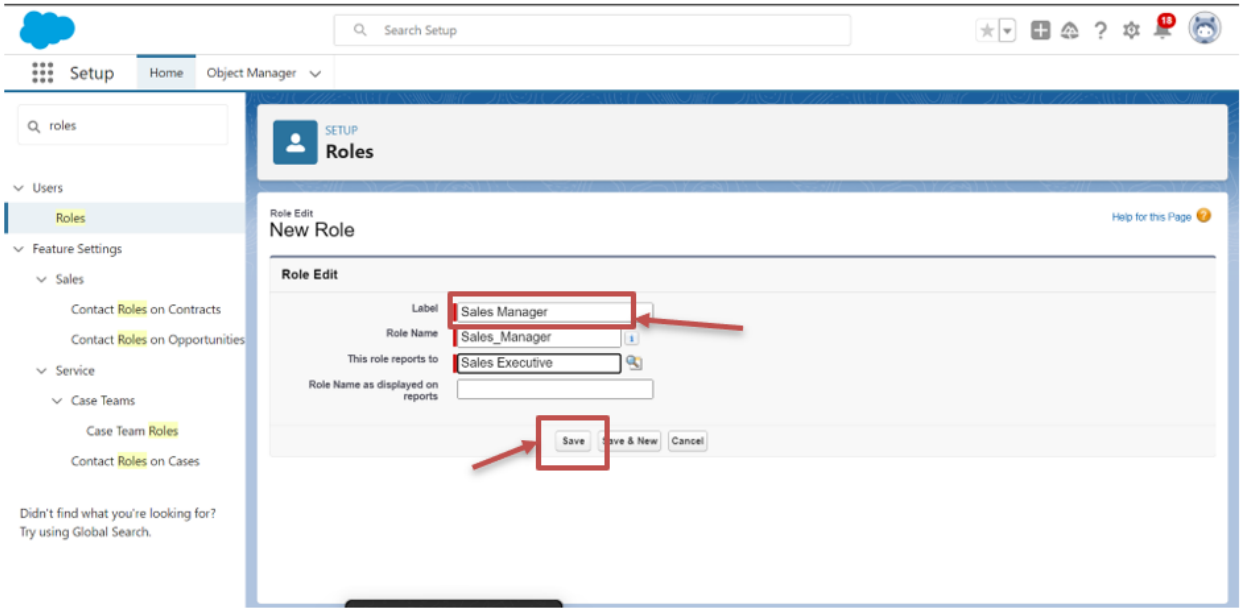


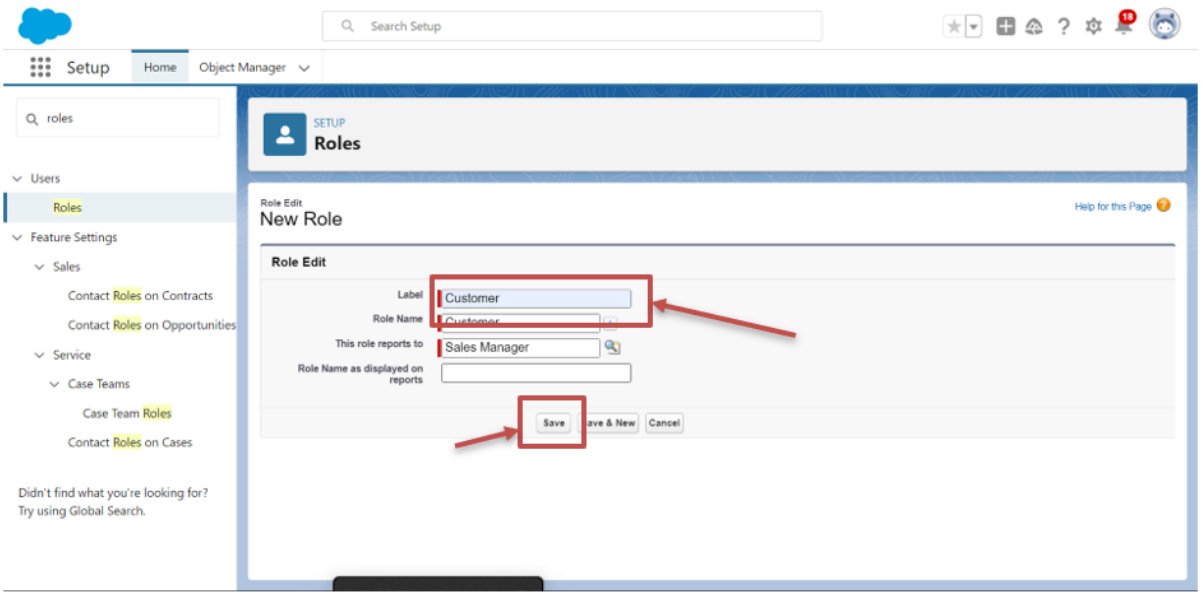
Activity 3: Create Roles Creating roles as per the business requirement.

1. Go to Setup and Click on Roles, then click on Expand all and Add a Role just below the Sales Representative.
2. Label - Sales Executive Reports to - Sales Representative



3. 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.

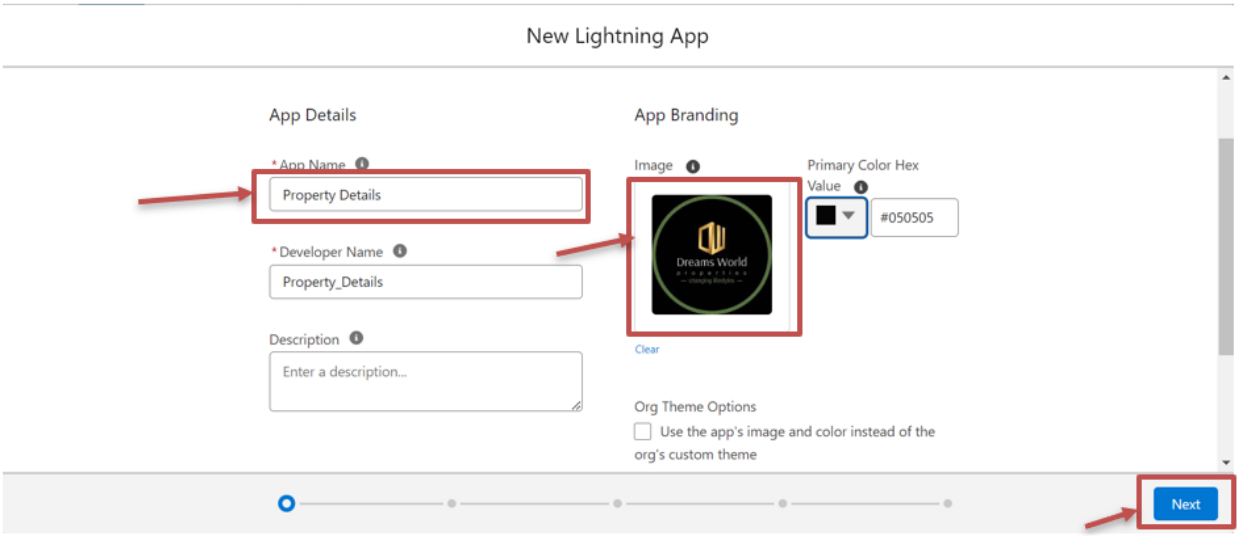


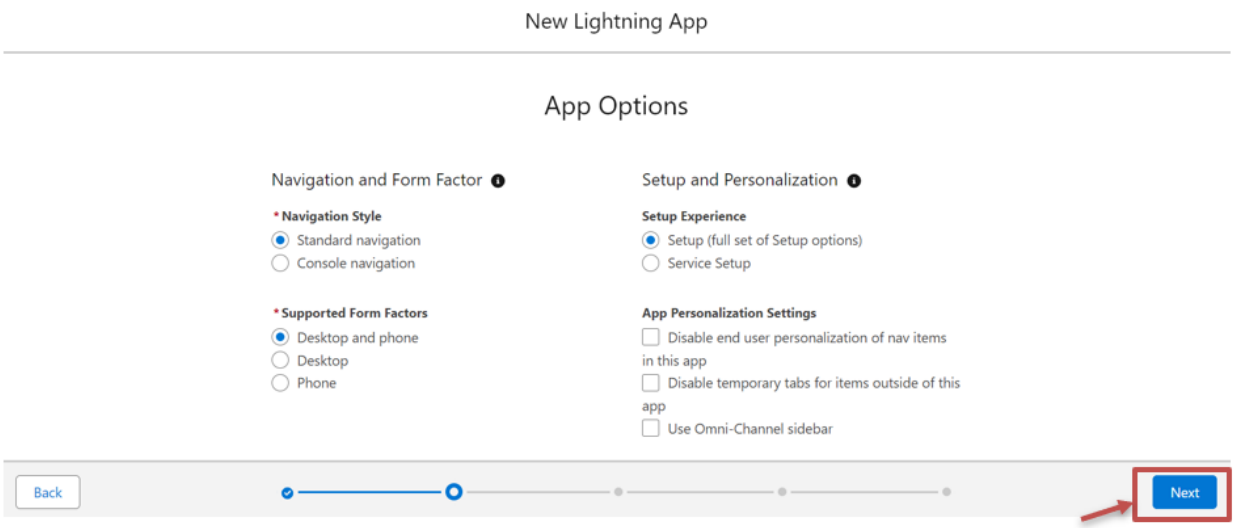


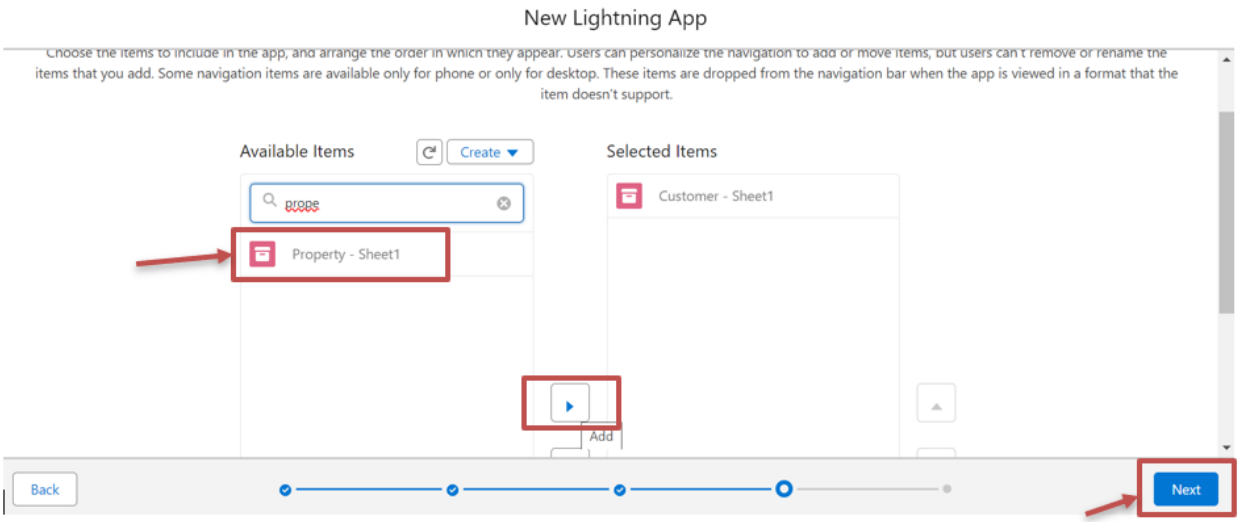
Activity 4: Create A Property Details App

Creating an app where the objects will be displayed.

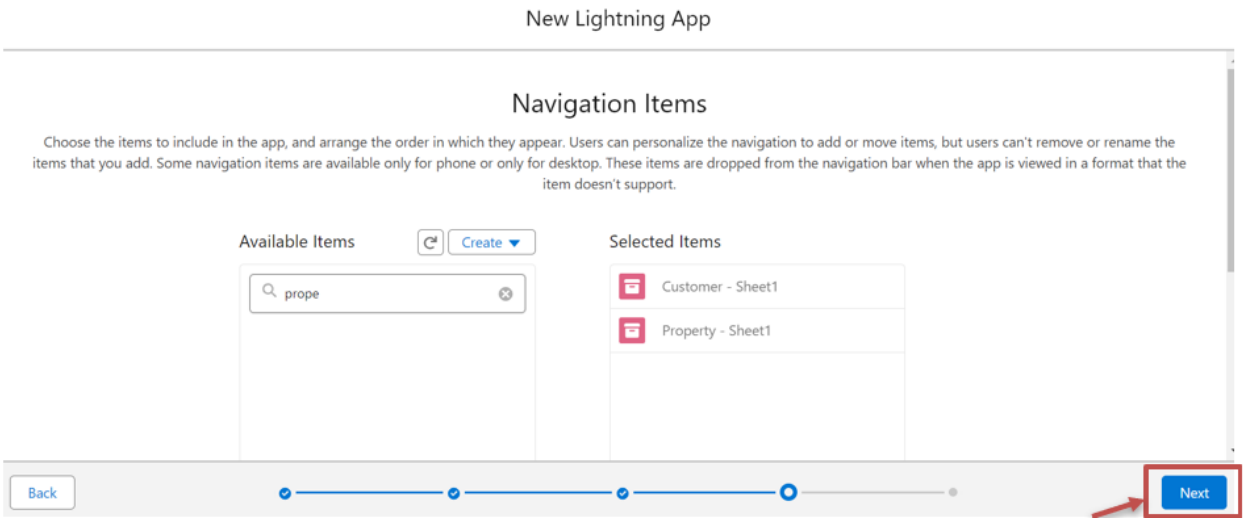
1. From Setup, Go to App Manager and click on New Lightning App and Name it as “Property Details” and add “Customer” and “Property” Object.

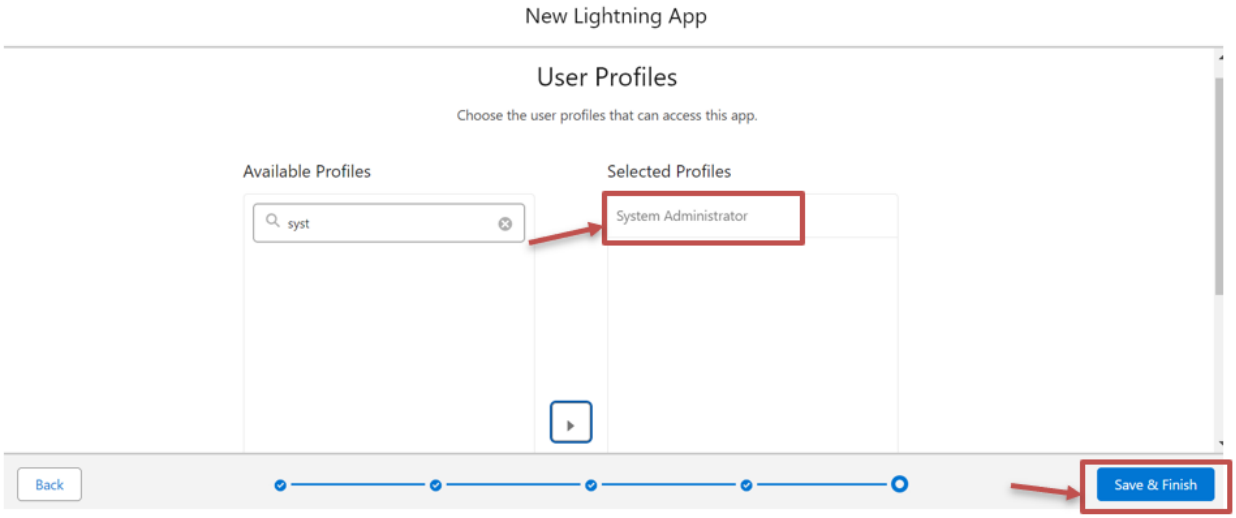






2. Click Next and Next then add “System Admin” Profile and Save.

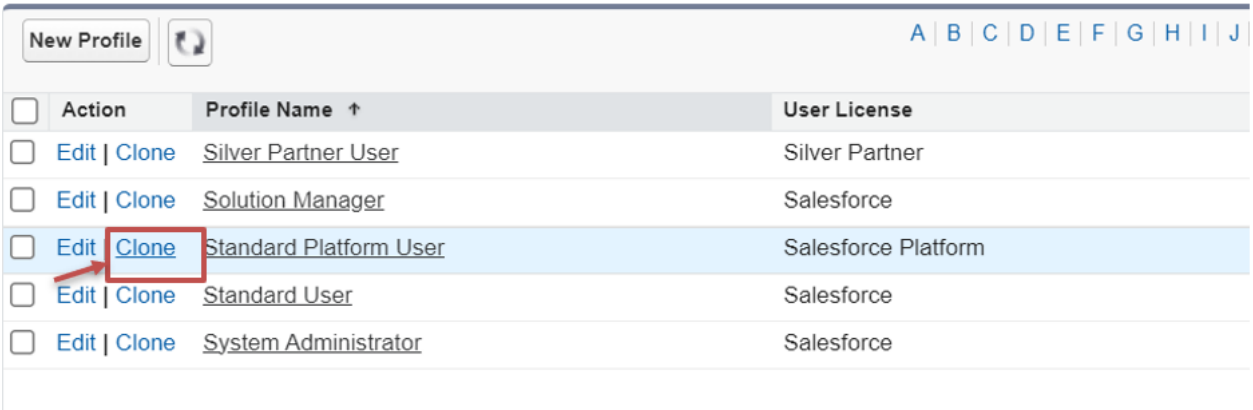


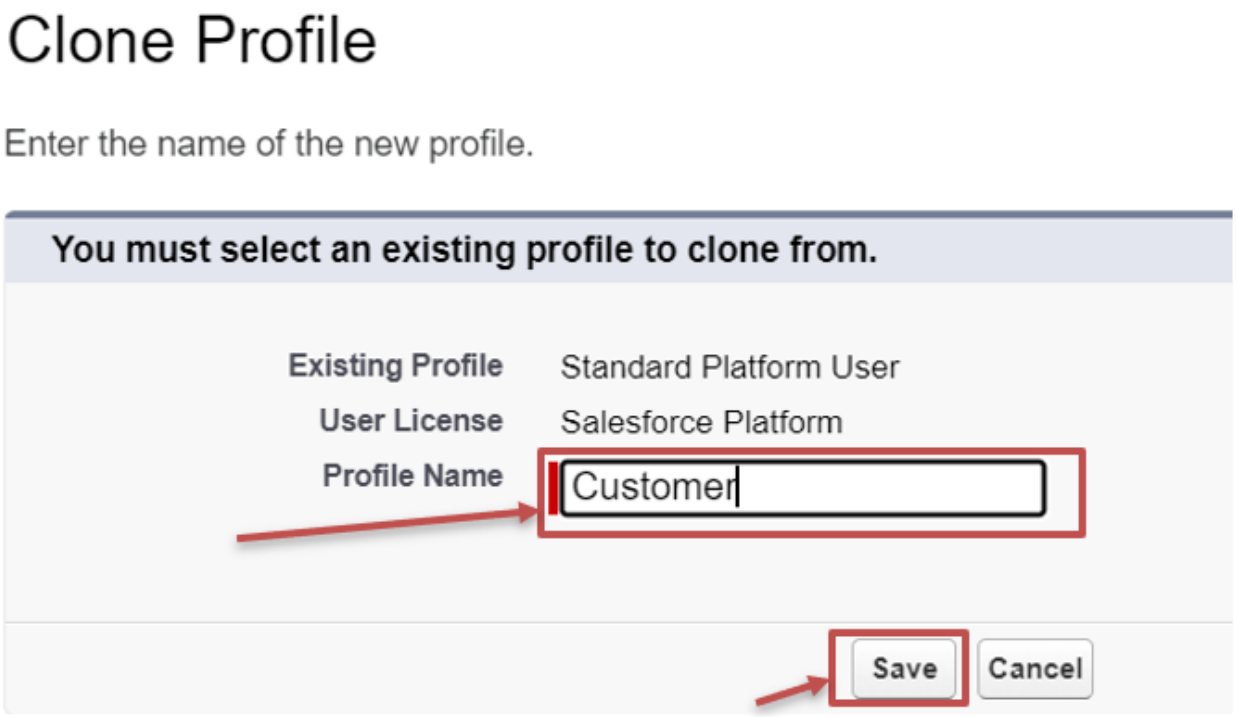


Activity 5: Create Profiles

Customer

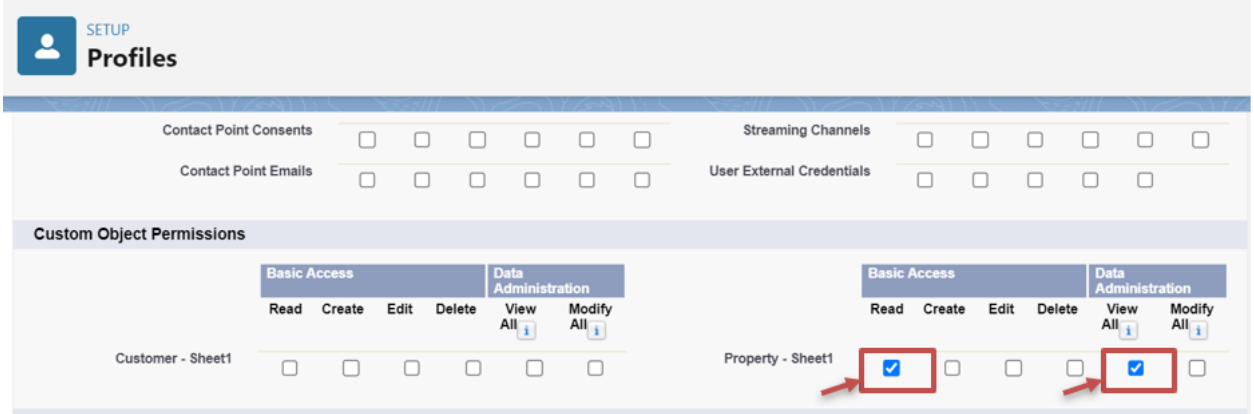
1. From Setup, Go to Profiles and Clone Salesforce Platform User and Name it “Customer”.





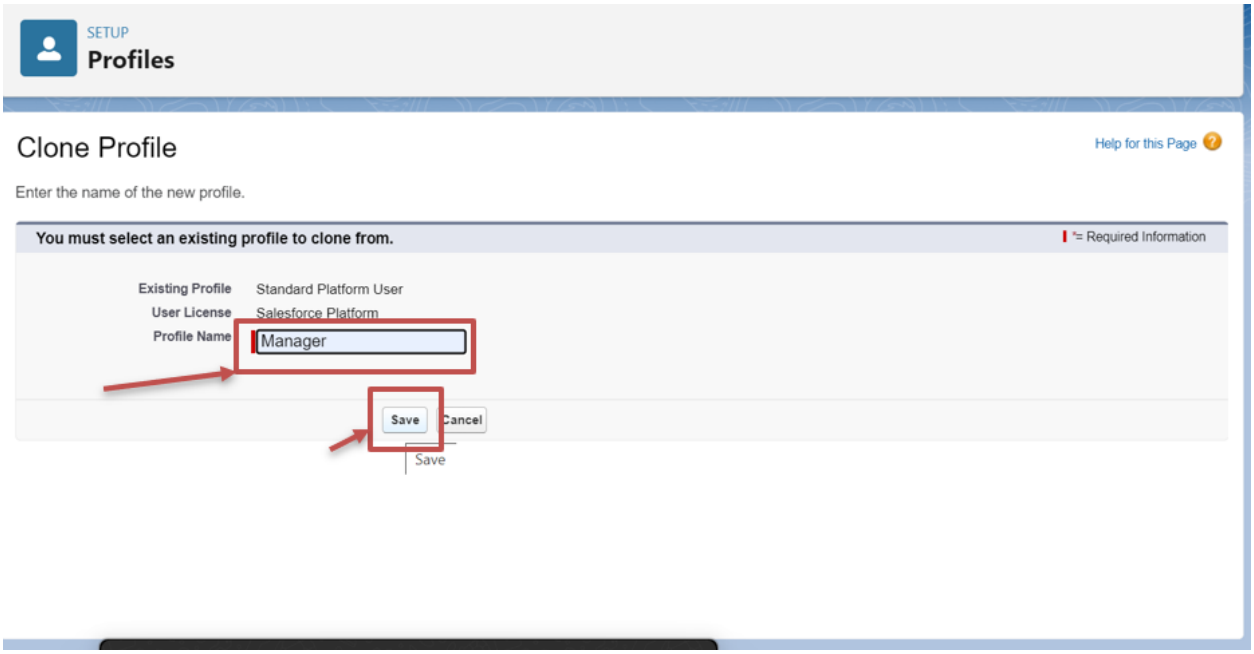
2. Uncheck all the Custom Objects and Check only Property Details from Custom App Settings.

3. Also Remove all the Standard Object Permissions.
4. Uncheck all the Custom Object Permissions and check read and view all in “Property”

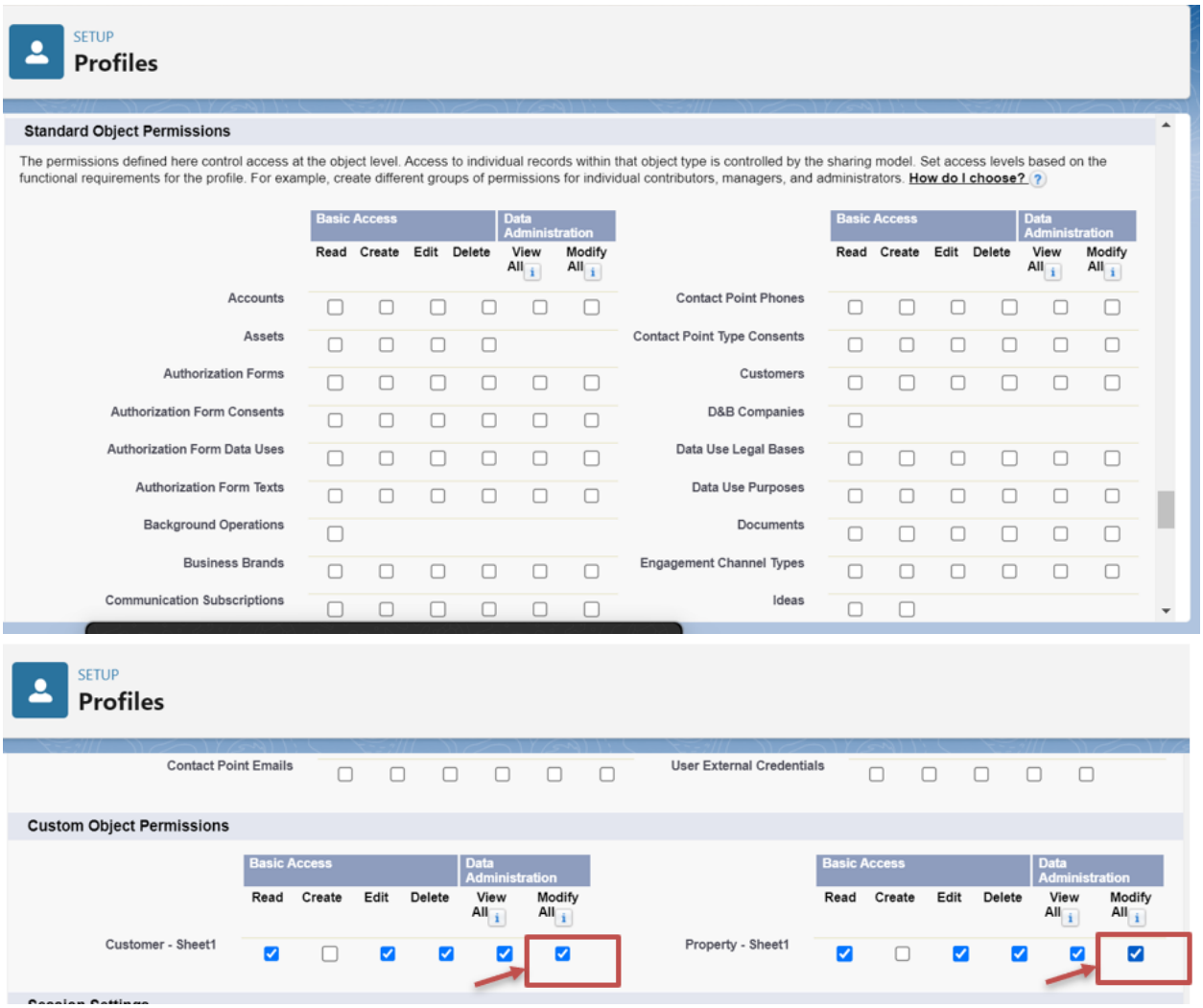


Manager

1. From Setup, Go to Profiles and Clone Salesforce Platform User and Name it “Manager”.

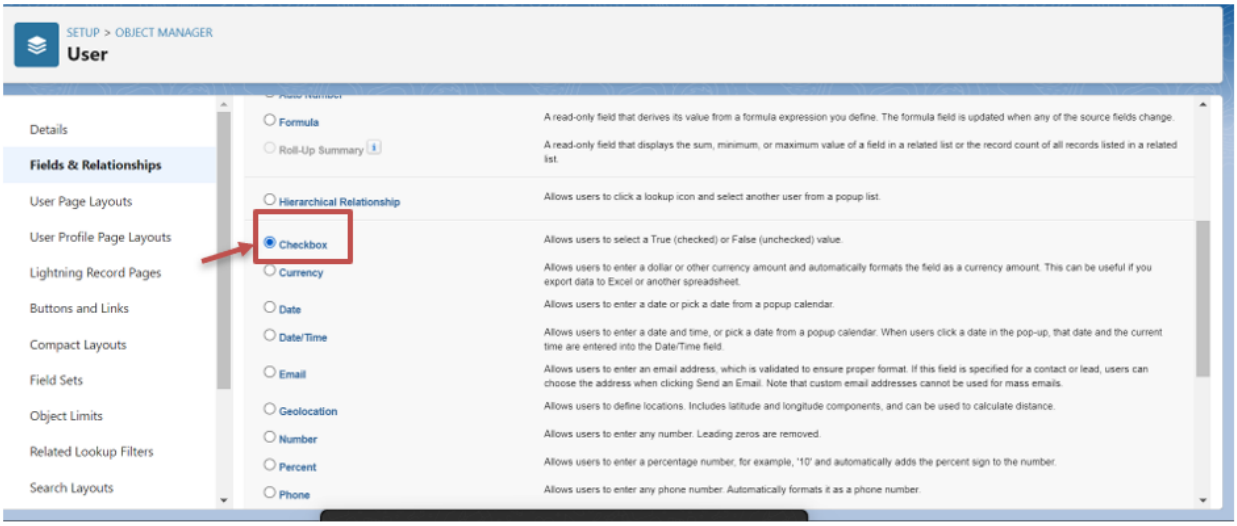


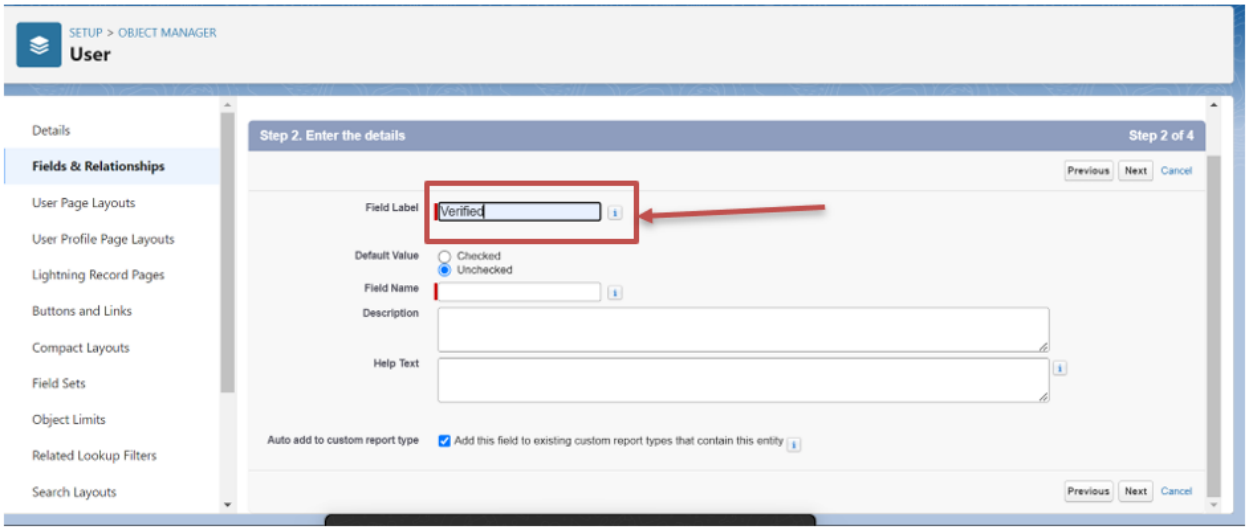
2. Uncheck all the Custom Objects and Check only Property Details from Custom App Settings.
3. Also Remove all the Standard Object Permissions
4. Uncheck all the Custom Object Permissions and check only “modify all” from “Property” and “Customer”.

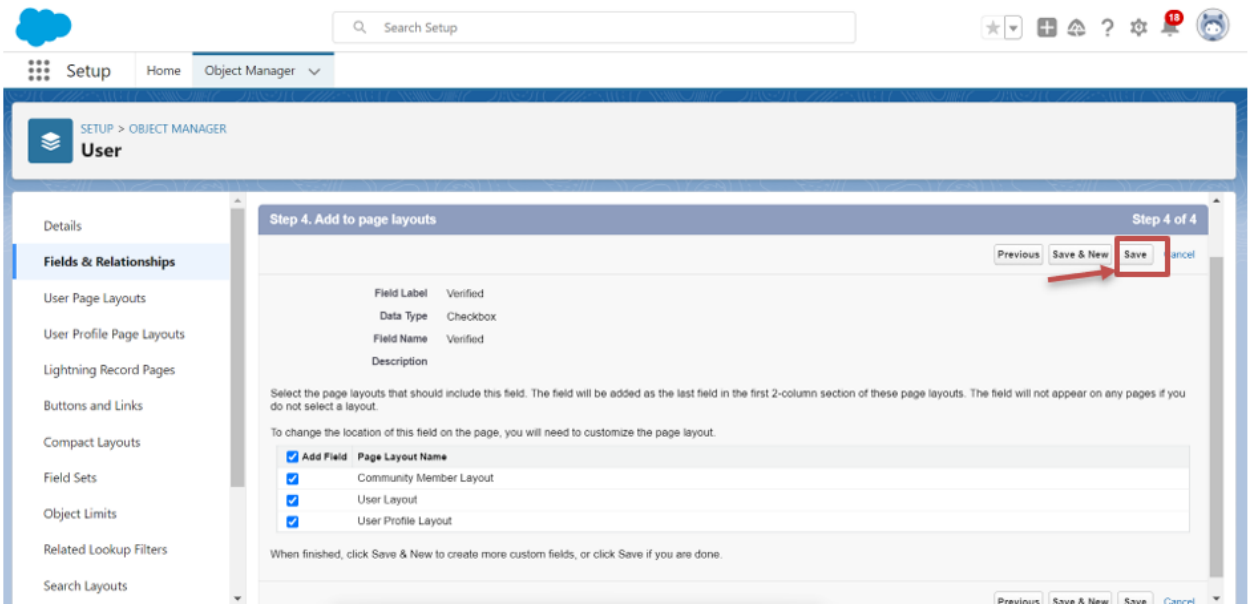


Activity 6: Create A Check Box Field on User

1. Setup go to Object Manager then go to Search for User select Fields and Relationships
2. Create new Field Named as “Verified” as Data type “Check Box”


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
Formula
A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.
Roll-Up Summary
A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.
Hierarchical Relationship
Allows users to click a lookup icon and select another user from a popup list.
Checkbox
Allows users to select a True (checked) or False (unchecked) value.
Currency
Allows users to enter a dollar or other currency amount and automatically formats the field as a currency amount. This can be useful if you export data to Excel or another spreadsheet.
Date
Allows users to enter a date or pick a date from a popup calendar.
Date/Time
Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.
Email
Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.
Geolocation
Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.
Number
Allows users to enter any number. Leading zeros are removed.
Percent
Allows users to enter a percentage number, for example, '10' and automatically adds the percent sign to the number.
Phone
Allows users to enter any phone number. Automatically formats it as a phone number.

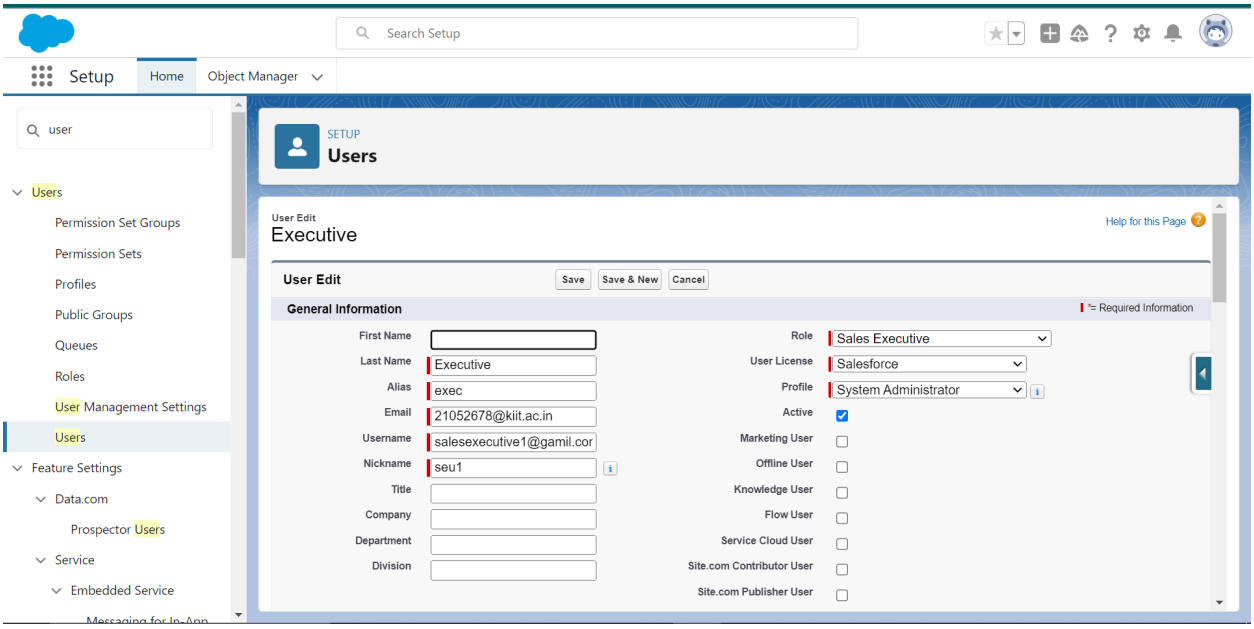

SETUP > OBJECT MANAGER
User
Details
Fields & Relationships
User Page Layouts
User Profile Page Layouts
Lightning Record Pages
Buttons and Links
Compact Layouts
Field Sets
Object Limits
Related Lookup Filters
Search Layouts
Step 2. Enter the details
Step 2 of 4
Previous
Next
Cancel
Field Label
Verified
Default Value
Checked
Unchecked
Field Name
Description
Help Text
Auto add to custom report type
Add this field to existing custom report types that contain this entity
Previous
Next
Cancel


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
Step 4. Add to page layouts
Step 4 of 4
Previous
Save & New
Save
Cancel
Field Label
Verified
Data Type
Checkbox
Field Name
Verified
Description
Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.
To change the location of this field on the page, you will need to customize the page layout.
Add Field
Page Layout Name
Community Member Layout
User Layout
User Profile Layout
When finished, click Save & New to create more custom fields, or click Save if you are done.

Activity7: Create Users

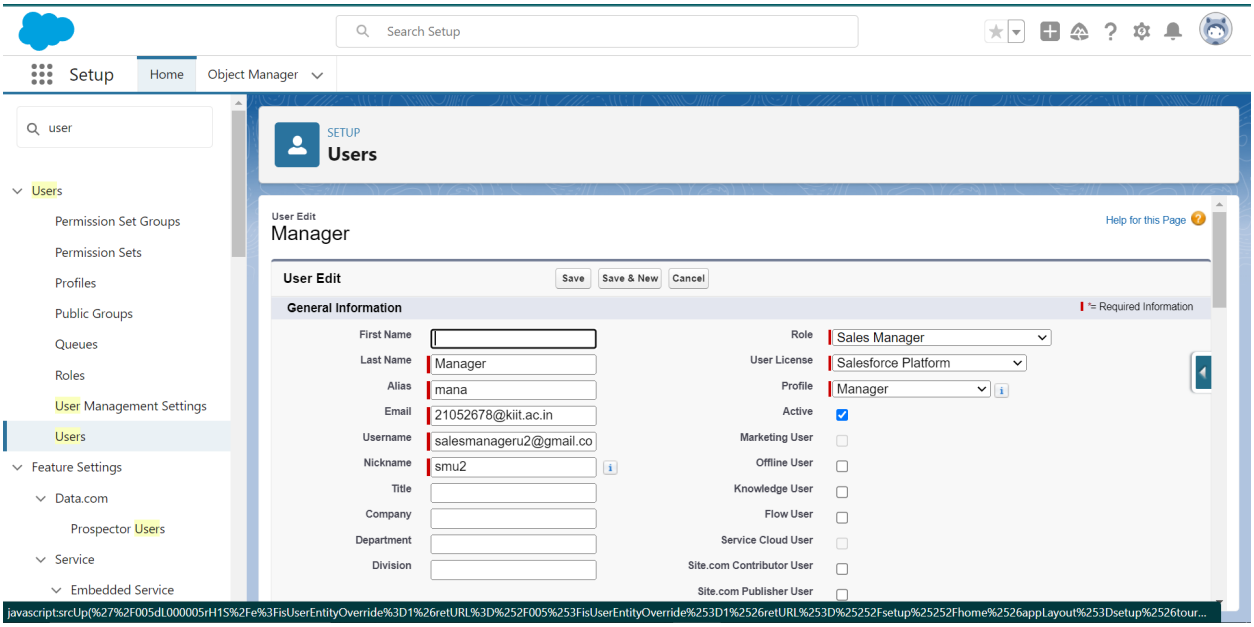
User1

1. Go to Setup then from Administration select Users then New User.
2. Last Name - Executive
3. Role - Sales Executive
4. License - Salesforce
5. Profile - System Administrator
6. Save



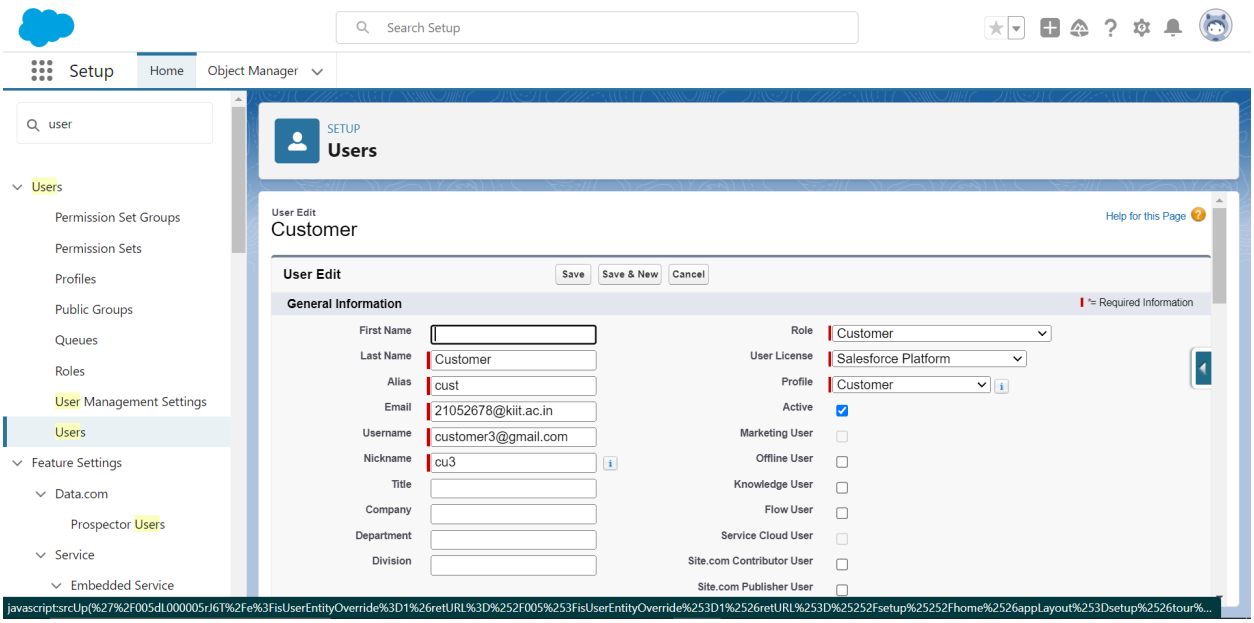
User2

1. Go to Setup then from Administration select Users then New User.
2. Last Name - Manager
3. Role - Sales Manager
4. License - Salesforce Platform
5. Profile - Manager
6. Save



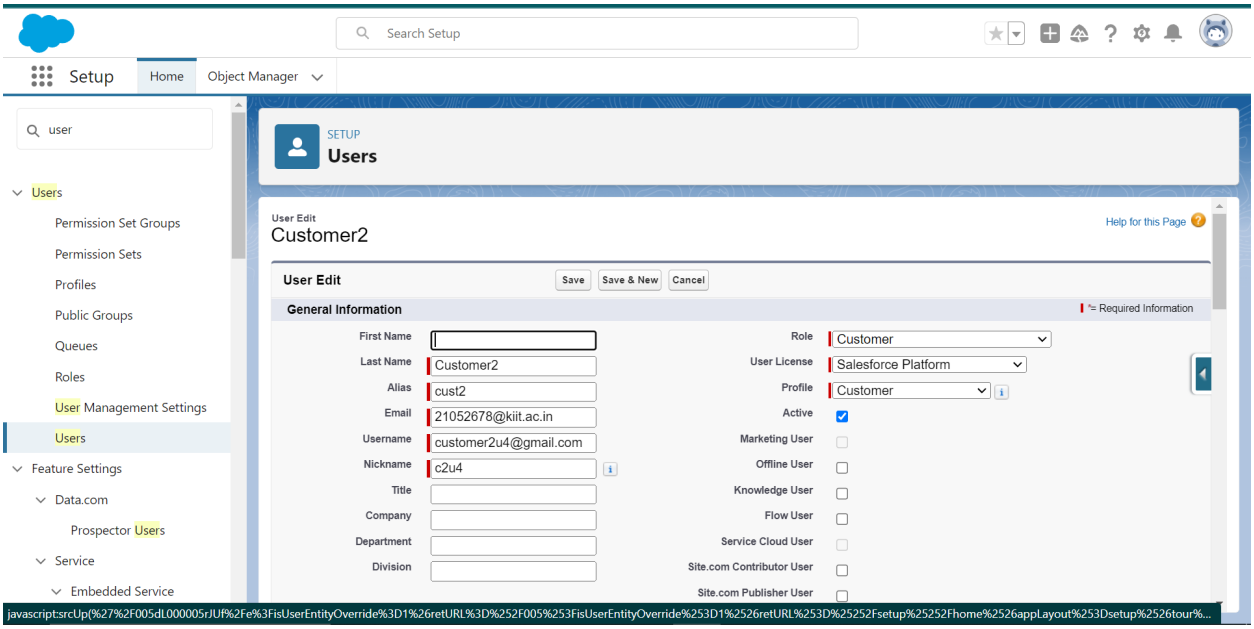
User3

1. Go to Setup then from Administration select Users then New User.
2. Last Name - Customer
3. Role - Customer
4. License - Salesforce Platform Jyoti Gupta Institute of Engineering and Management
5. Profile - Customer
6. Make Sure the verified check box is “Unchecked”
7. Save



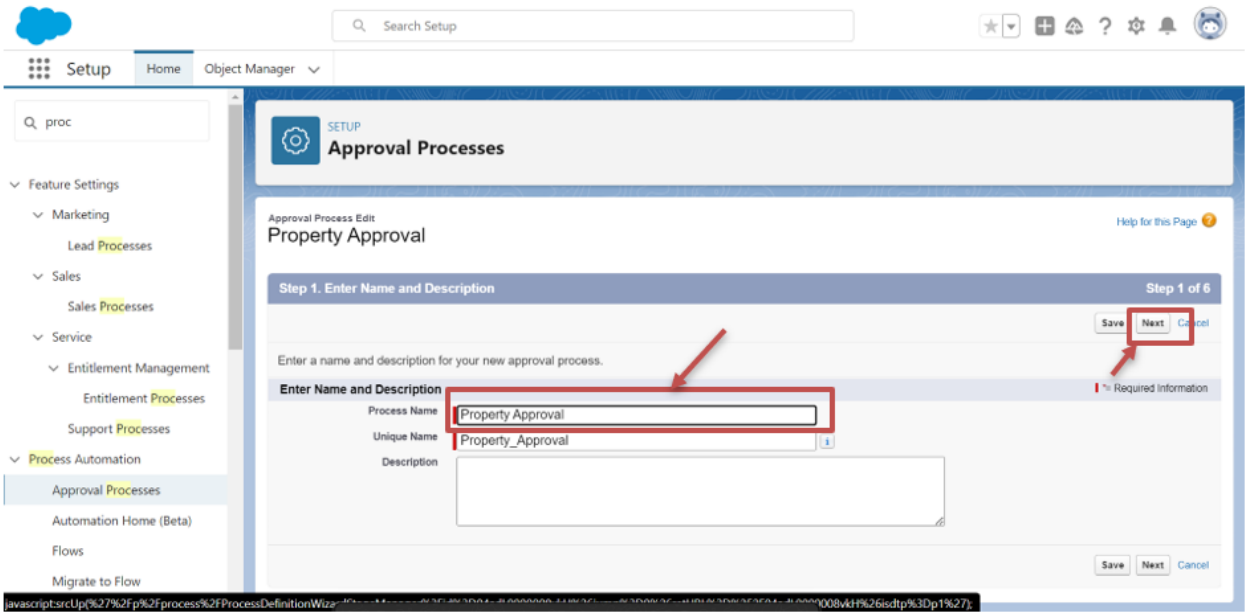
User4

1. Go to Setup then from Administration select Users then New User.
2. Last Name - Customer2
3. Role - Customer
4. License - Salesforce Platform
5. Profile - Customer
6. Make Sure the verified check box is “checked”.
7. Save

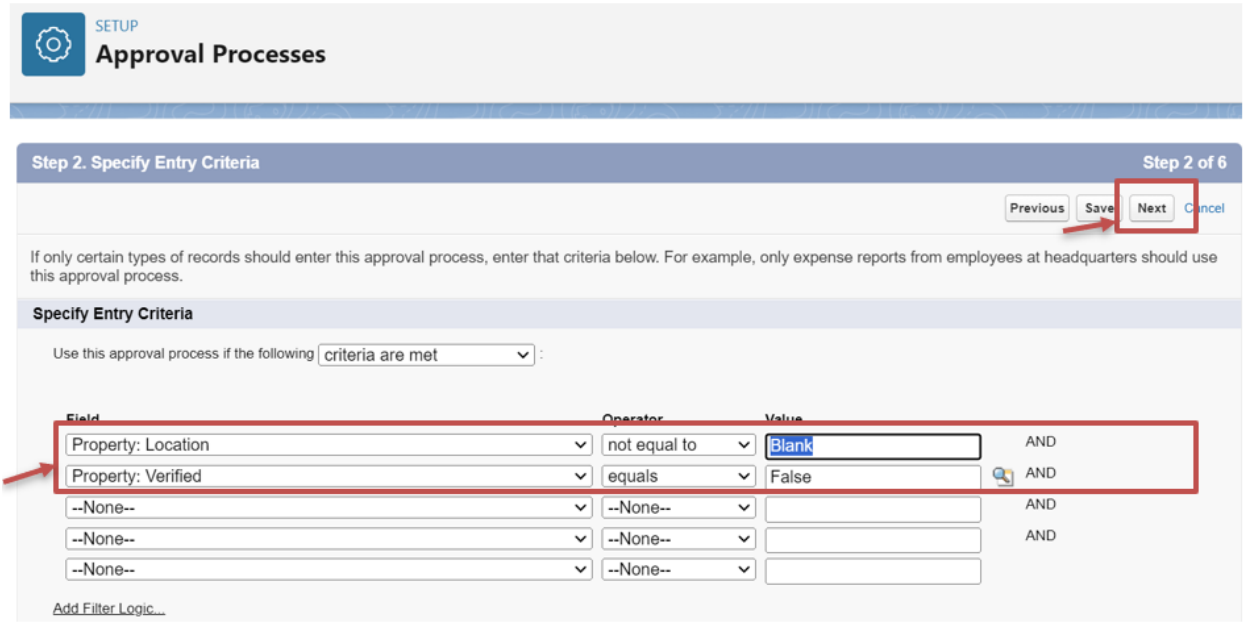


Activity 8: Create An Approval Process for Property Object

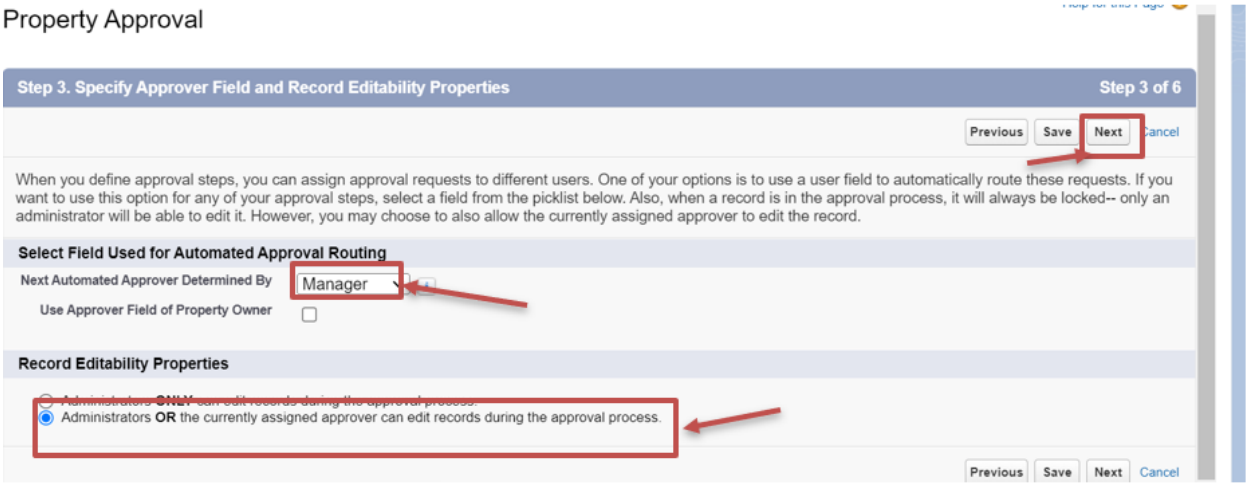
1. From Setup search Process Automation and select Approval Process
2. Process Name - Property Approval



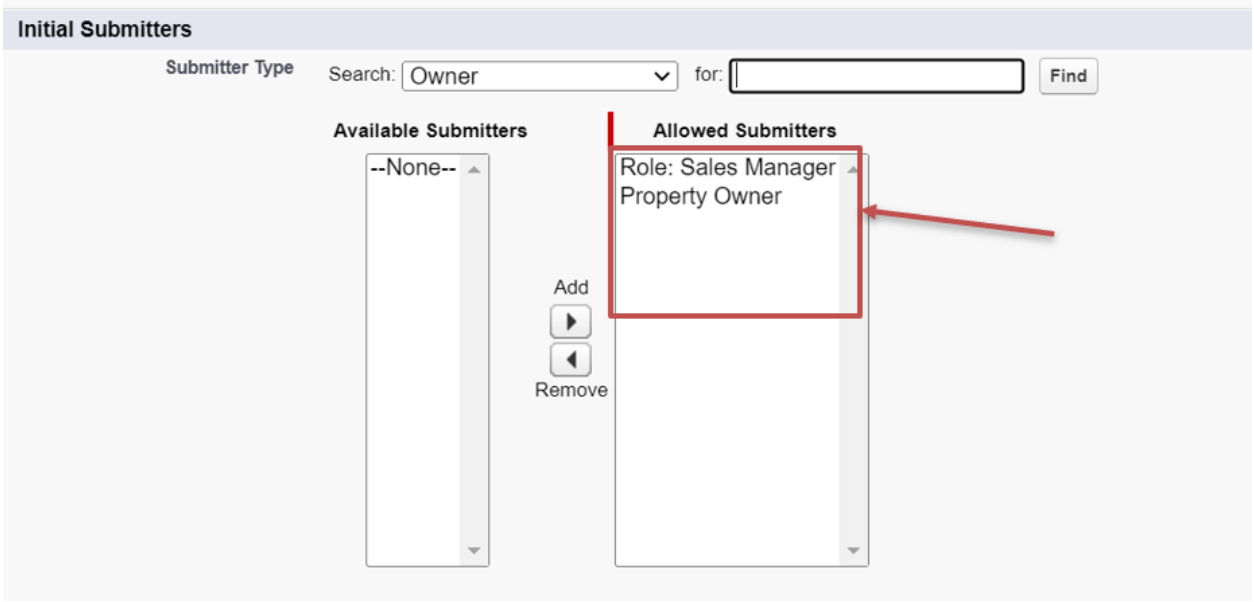
3. Give 2 criteria - Jyoti Gupta Institute of Engineering and Management
    a. Location is not equal to blank,
    b. Verified Equals false.



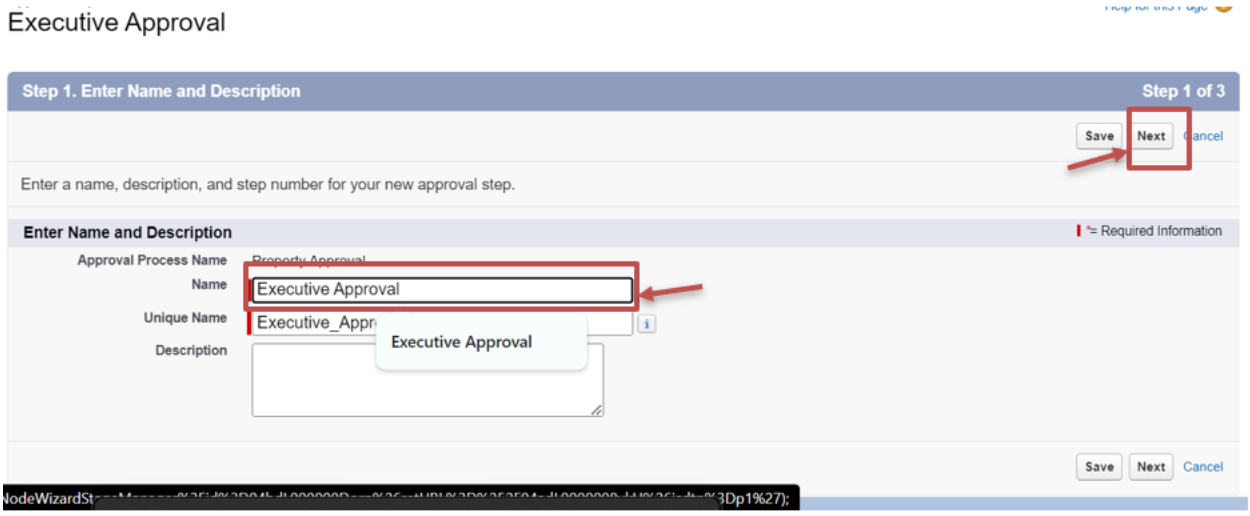
4. Click next and “Next Automated Approver Determined By” Select Manager
5. From Record Editability Properties click on Administrators OR the currently assigned approver can edit records during the approval process.



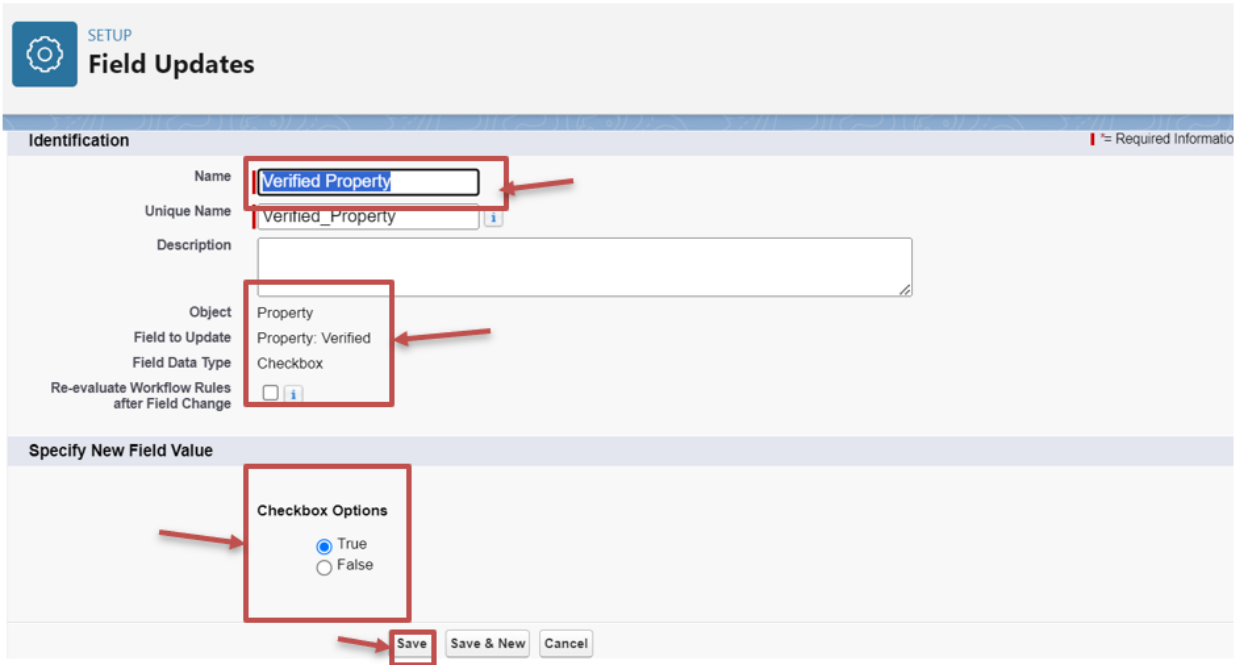
6. From Step 5. Select Fields to Display on Approval Page Layout select Property, Owner, Location, Type.
7. Click Next and Select the Initial Submitters
    a. Owner - Property Owner
    b. Roles - Sales Manager



8. Save.
9. Add an approval step name “Executive Approval”



10. Specify the Criteria as All record should enter
11. Click next and select the Approver as “Sales Executive “and “Save”
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.



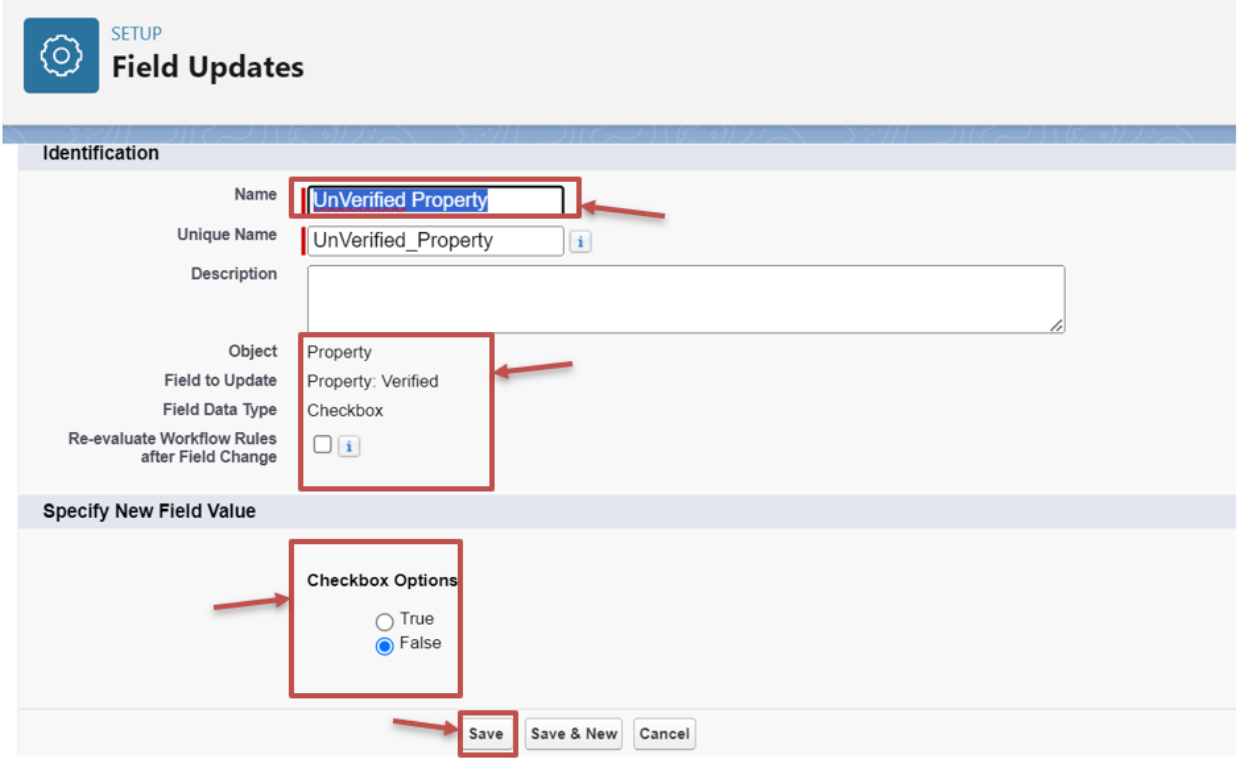
13. Add One field Update as “UnVerified Property”
    a. Select Object - Property Jyoti Gupta Institute of Engineering and Management

b. Field to Update - Verified

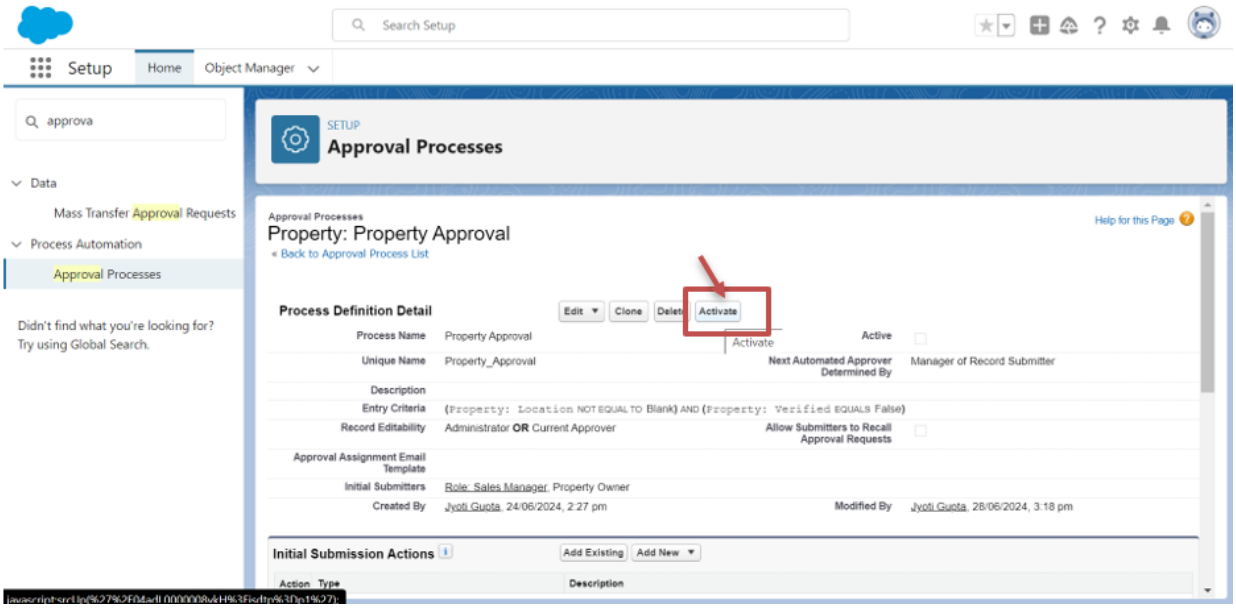
c. Field Data Type - Checkbox

d. Select Checkbox Option as “False”

e. Save.

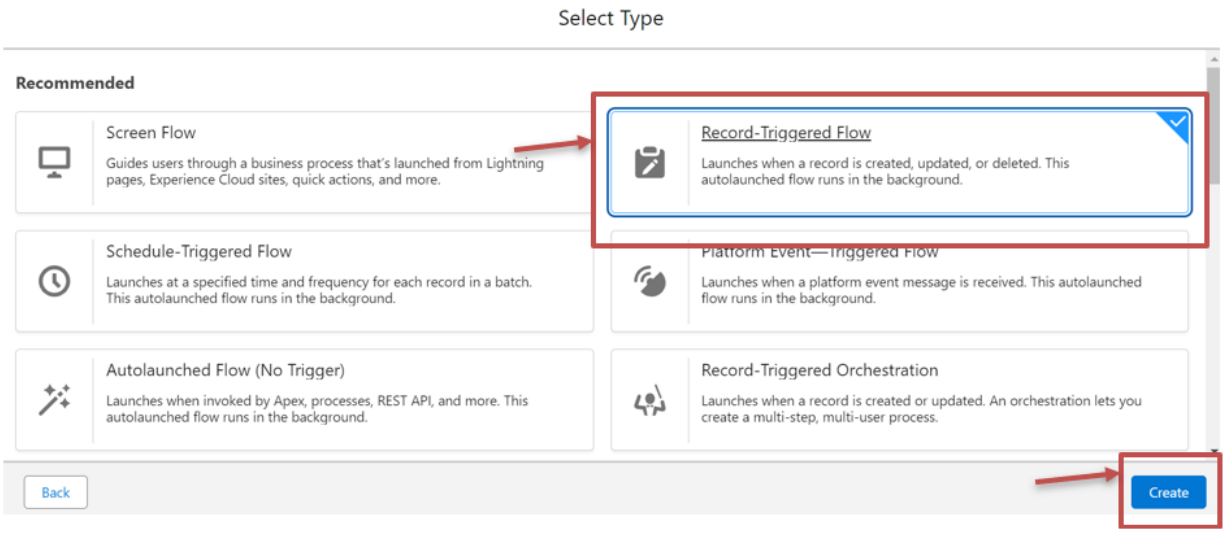


14. Activate the Approval Process.

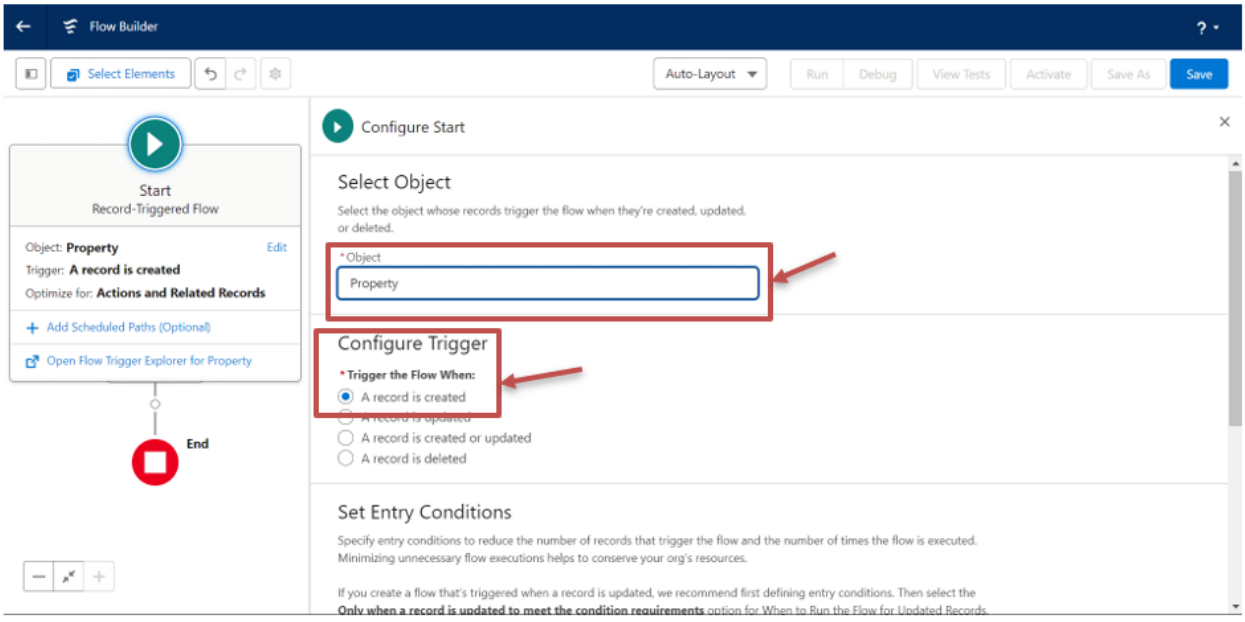


Activity 9: Create A Record Trigger Flow to Submit the Approval Process Automatically

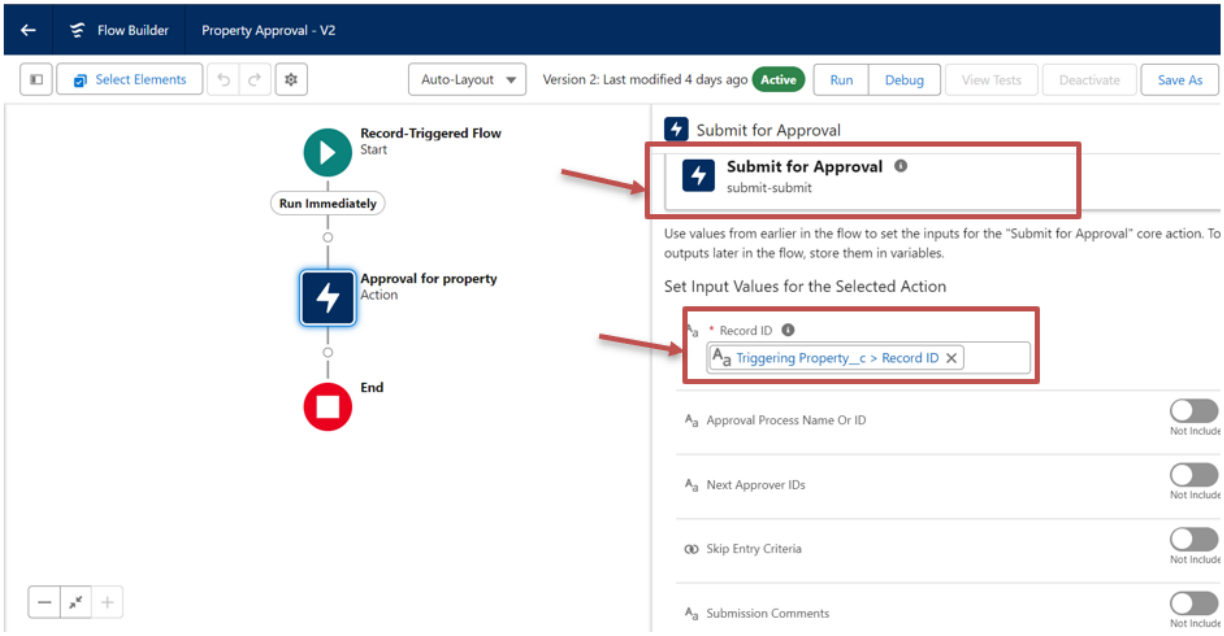
1. From Setup Search for Flows then Click on New and Select “Record Trigger Flow”.



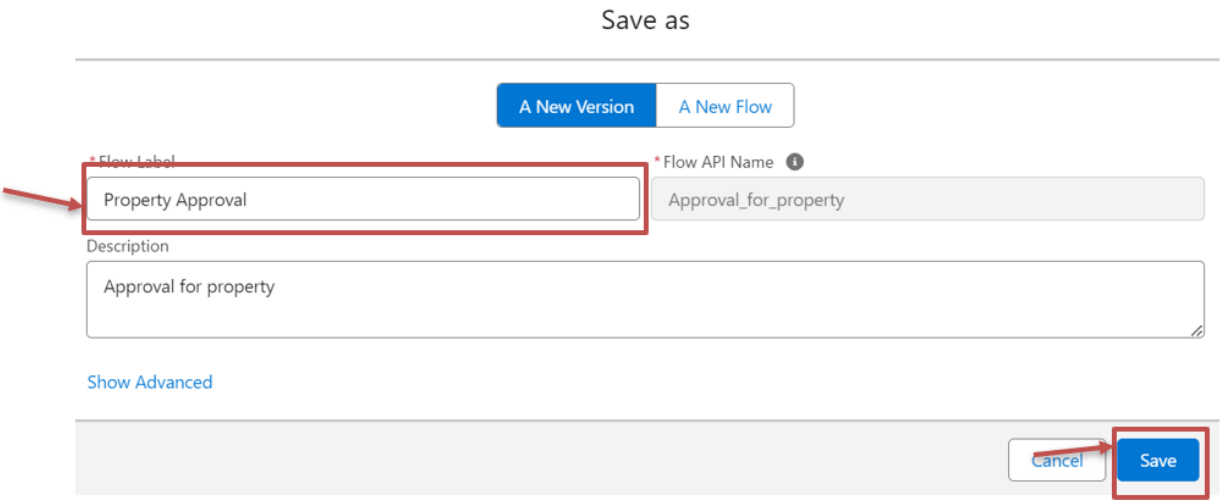
2. Select Object as Property
3. Select “Trigger the flow when” - “A record is created”



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.

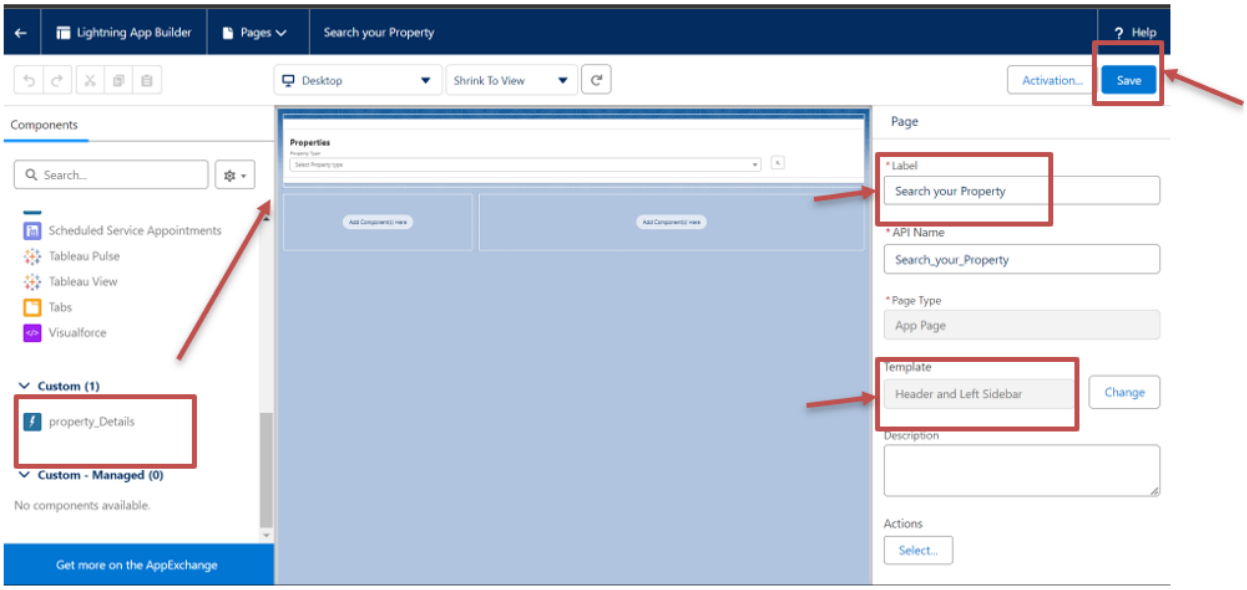


9. Save the Flow and Give label as “Property Approval” and “Activate



Activity 10: Create An App Page

1. From Setup go to Lightning App Builder Then Click on New and Select App Page then 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”.
6. From Lightning Experience Click on “Property Details” and click on Add Page “.



7. Then Click on “Save”

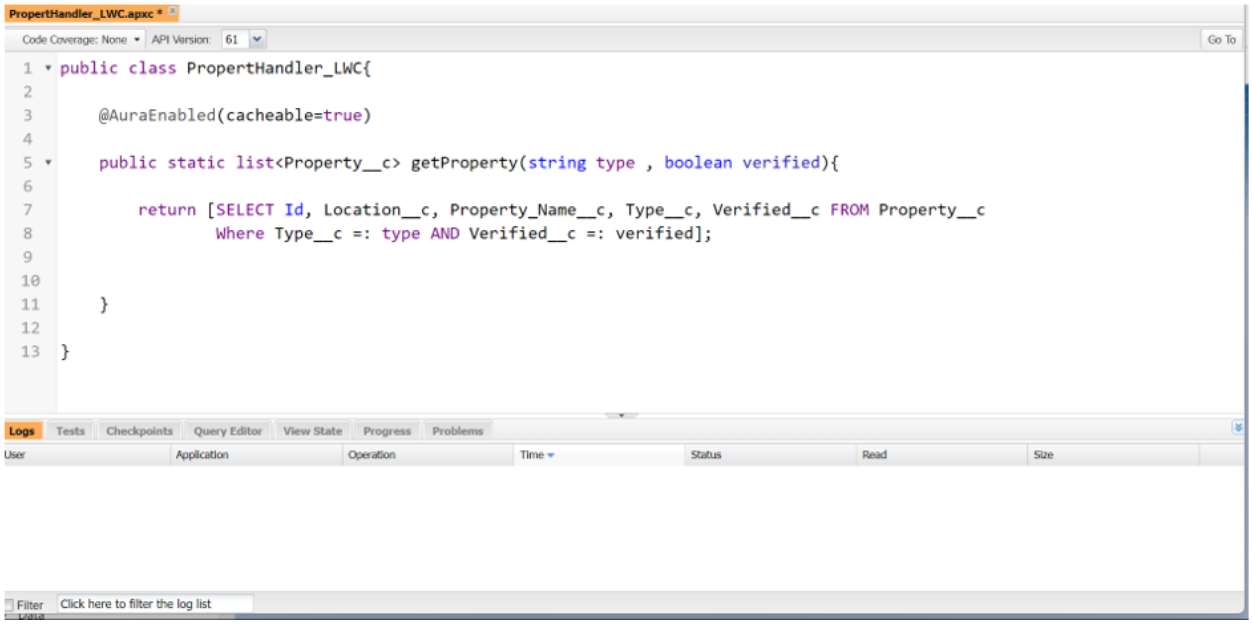
Activity 11: Create A LWC Component

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 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) and Create a lightning Web Component and Name it Anything you want to.
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={istrue}>
    <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/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)
      })
  }


}
```

```
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' },
```

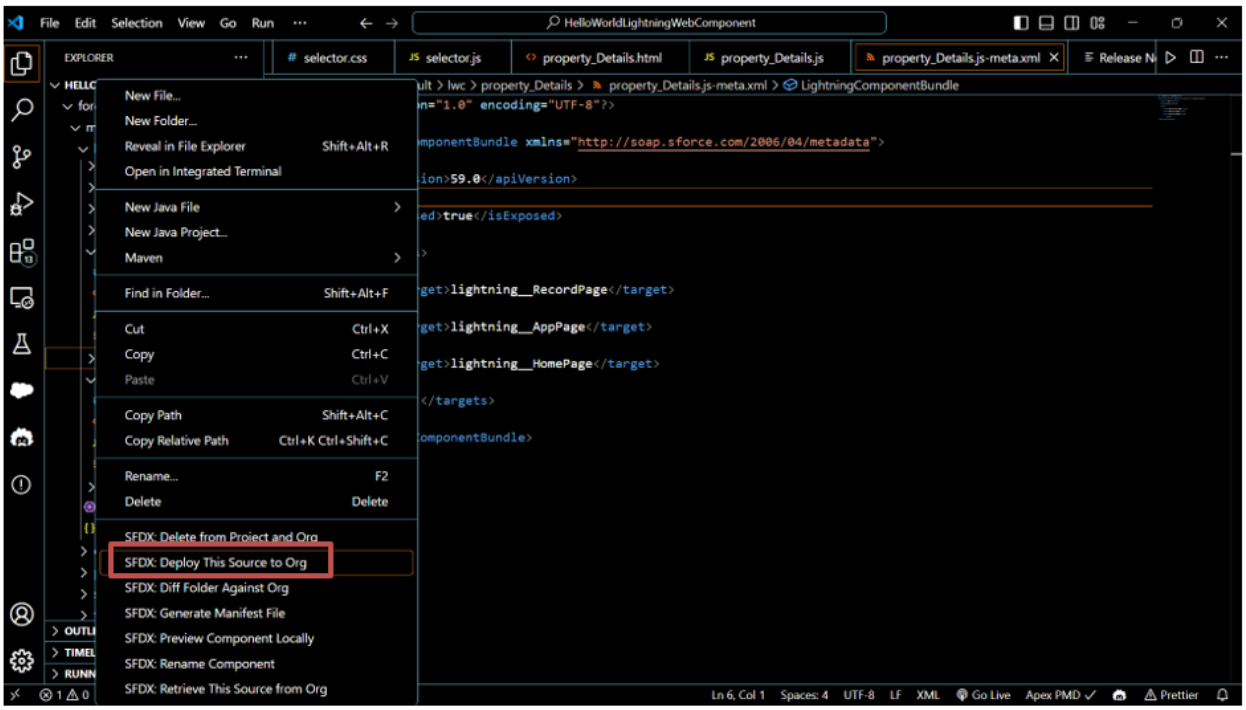
7. In Your metafile give your targets to deploy the component.

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>
```

```xml
<?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.

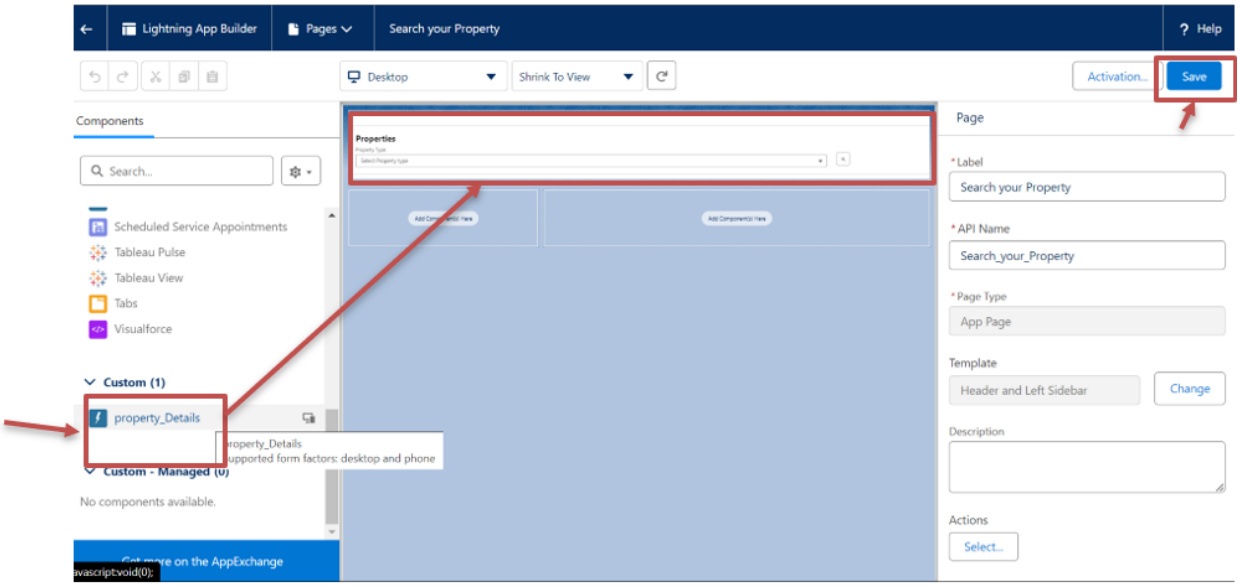


Activity 12: Drag This Component To Your App Page

Adding the Component to your Page

1. From Setup go to App Launcher and select 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.



Activity 13: Give Access of Apex Classes to Profiles

1. From Setup search for Apex Classes and click on “Security” behind “PropertyHandler__LWC”.
2. From Profiles Add “Manager” and “Customer” and “Save”