

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



Introduction

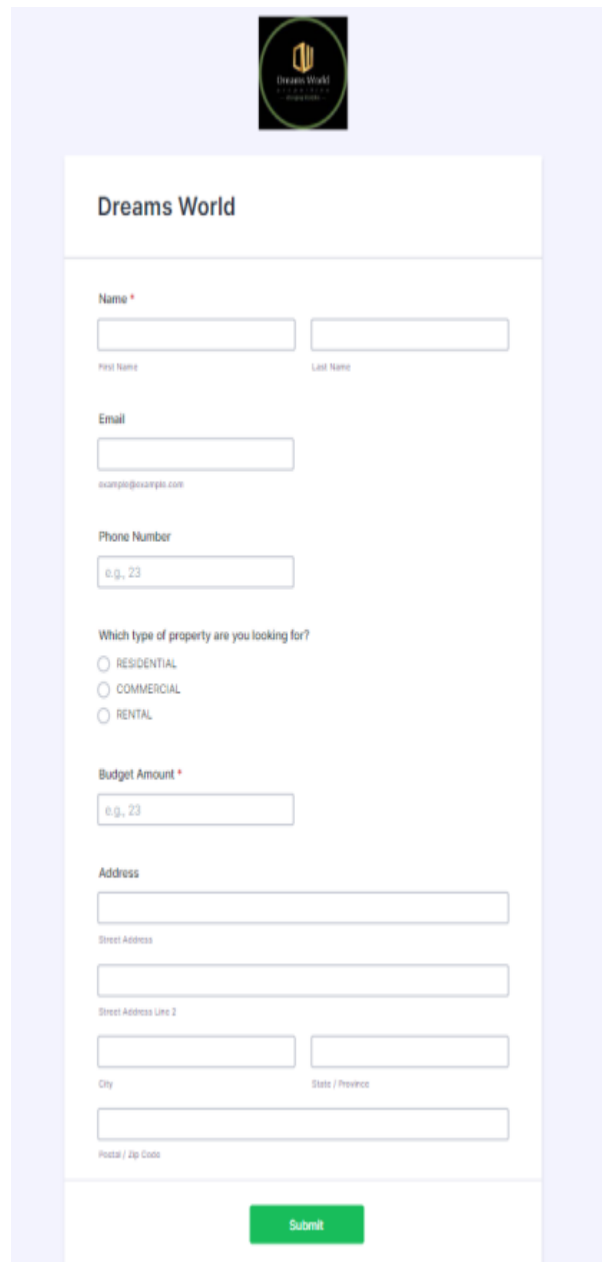
Dreams World Properties integrates Salesforce to streamline customer interactions. Website engagement triggers automated record creation in Salesforce, capturing customer details and preferences. Salesforce categorizes users as approved or non-approved, offering tailored property selections to approved 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>

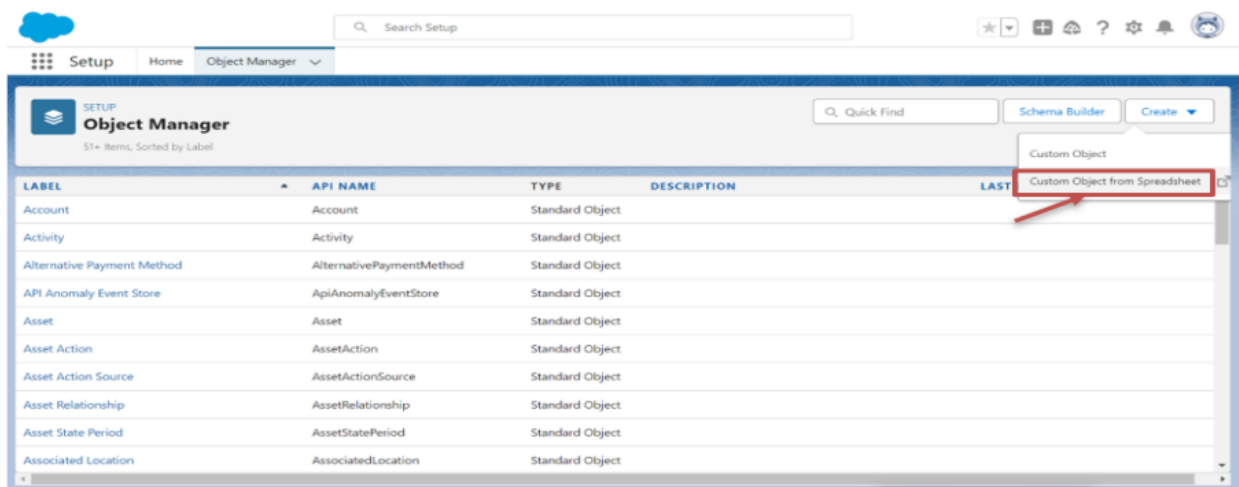
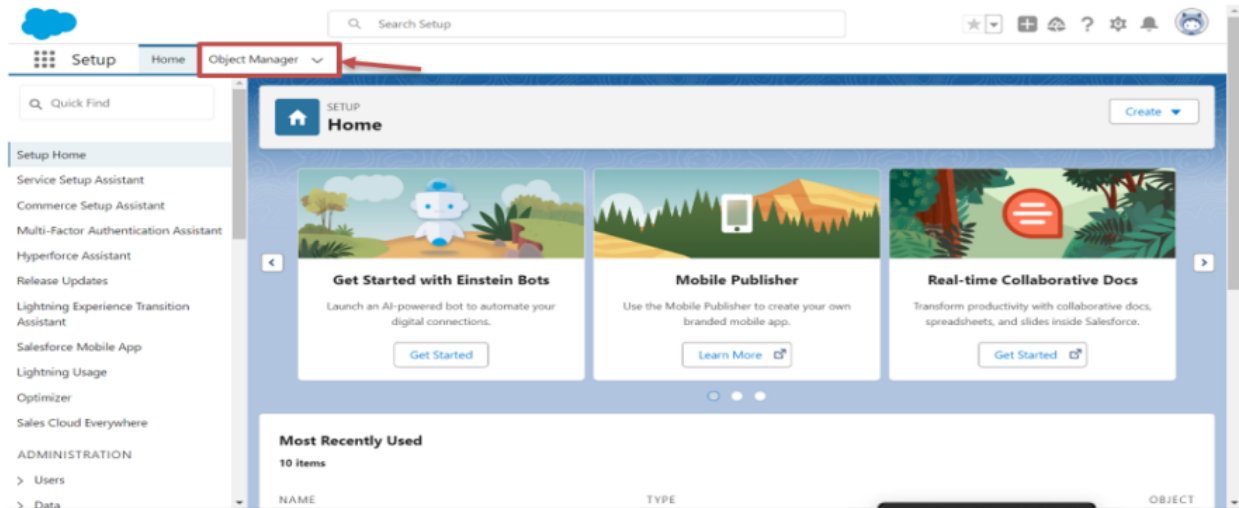


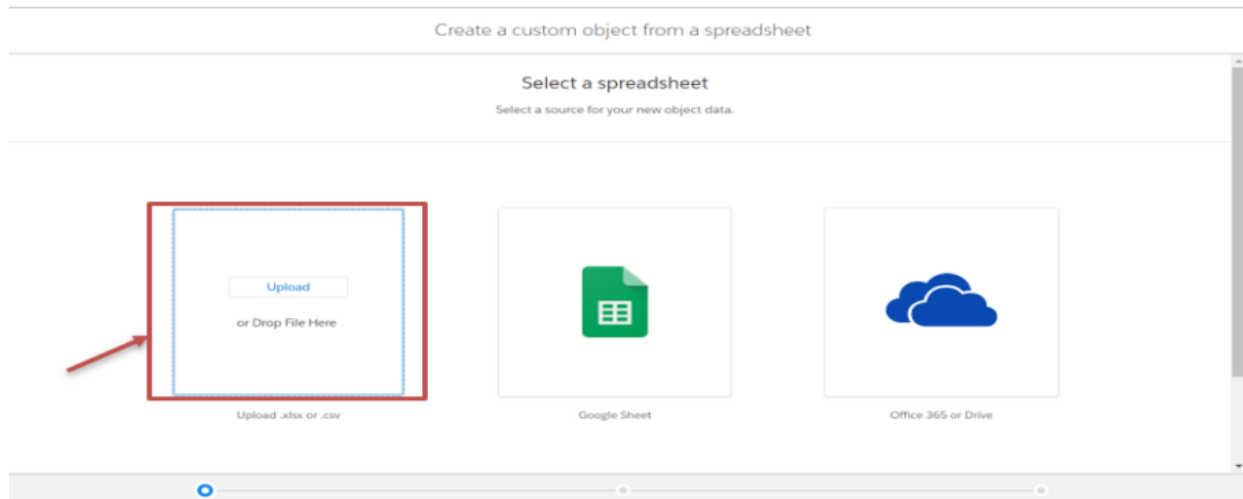
The image shows a JotForm titled "Dreams World" with a logo in the top right corner. The form contains the following fields and sections:

- Name ***: Two input fields for "First Name" and "Last Name".
- Email**: One input field with a placeholder "example@example.com".
- Phone Number**: One input field with a placeholder "e.g., 23".
- Which type of property are you looking for?**: Three radio button options: "RESIDENTIAL", "COMMERCIAL", and "RENTAL".
- Budget Amount ***: One input field with a placeholder "e.g., 23".
- Address**: Four input fields for "Street Address", "Street Address Line 2", "City", and "State / Province".
- Postal / Zip Code**: One input field.
- Submit**: A green button at the bottom right.

Activity 1: Create Objects from Spreadsheet.

Directly Creating Objects from Spreadsheet in Salesforce

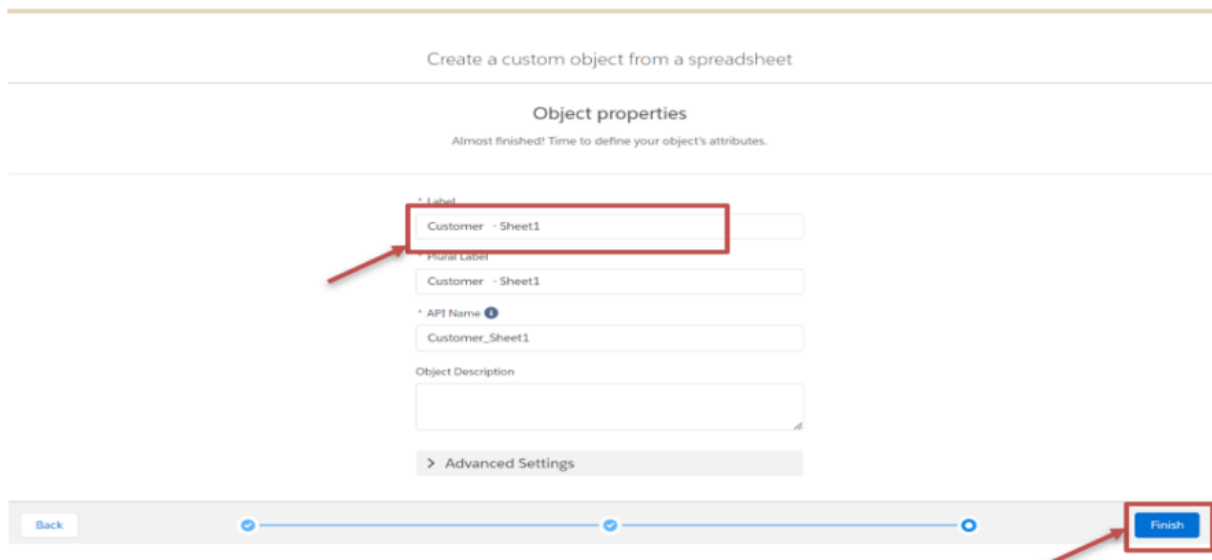




Create Customer object

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 ¹ Values Separated By Field Label Source ☐ Enter manually ☒ Detect from row * Field Labels Row Import **3** rows of Data? ² ☐ No, skip import ☒ Yes, import data Record Name Field ³

Fields 4 of 4 to import ☐ Hide mapped fields

IMPORT FILE FIELD NAME	SALESFORCE FIELD NAME	SALESFORCE FIELD TYPE	ADD TO LAYOUTS ⁴	FIELD PREVIEW
✓ Property Name	Property Name	Text	<input checked="" type="checkbox"/>	Lotus Appartments
✓ Type	Type	Text	<input checked="" type="checkbox"/>	Residential
✓ Location	Location	Text	<input checked="" type="checkbox"/>	hydeerabad
✓ Verified	Verified	Text	<input checked="" type="checkbox"/>	checked

[Back](#) [Next](#)

Create a custom object from a spreadsheet

Object properties

Almost finished! Time to define your object's attributes.

* Label

Plural Label

* API Name ¹

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

Rows Detected
3

Fields Created
4

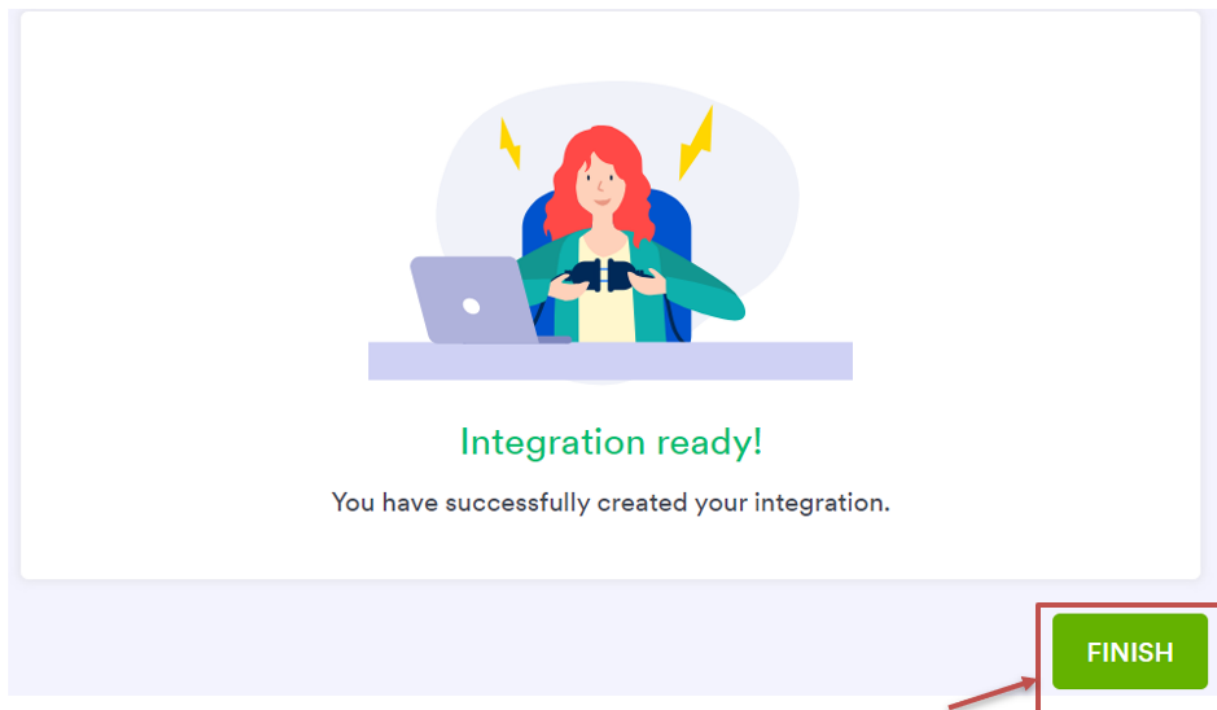
Rows Imported
3

[Import Another Object](#)

Integrate Jotform with Salesforce Platform

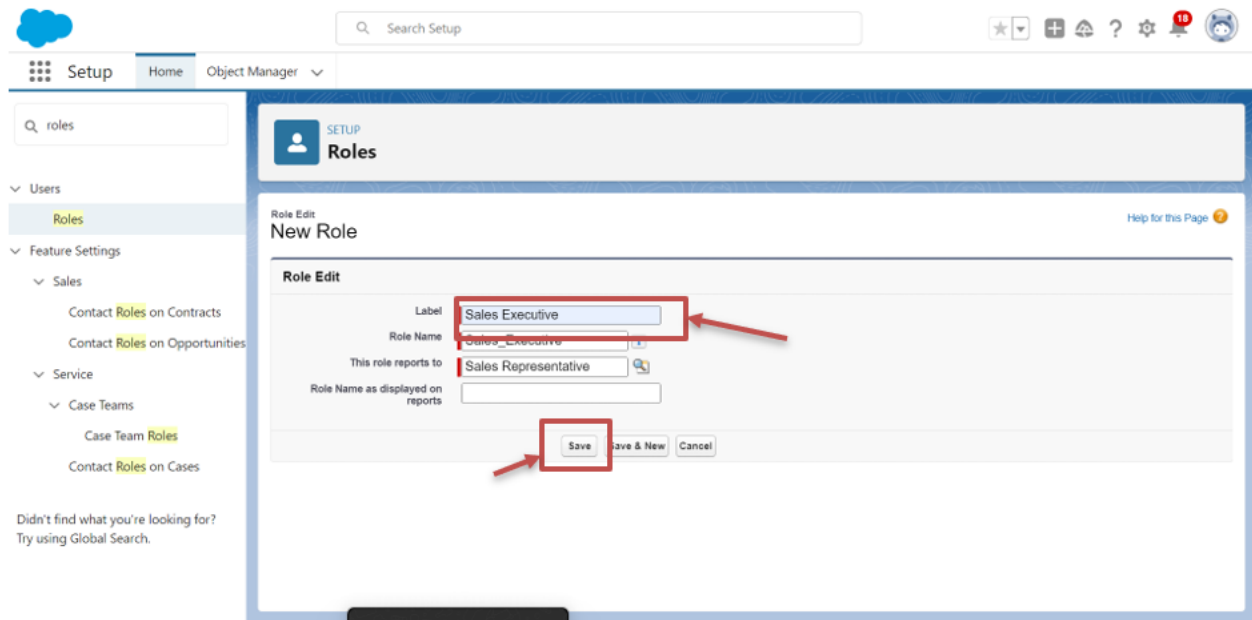
In this Milestone we are going to integrate jotform with Salesforce

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”.

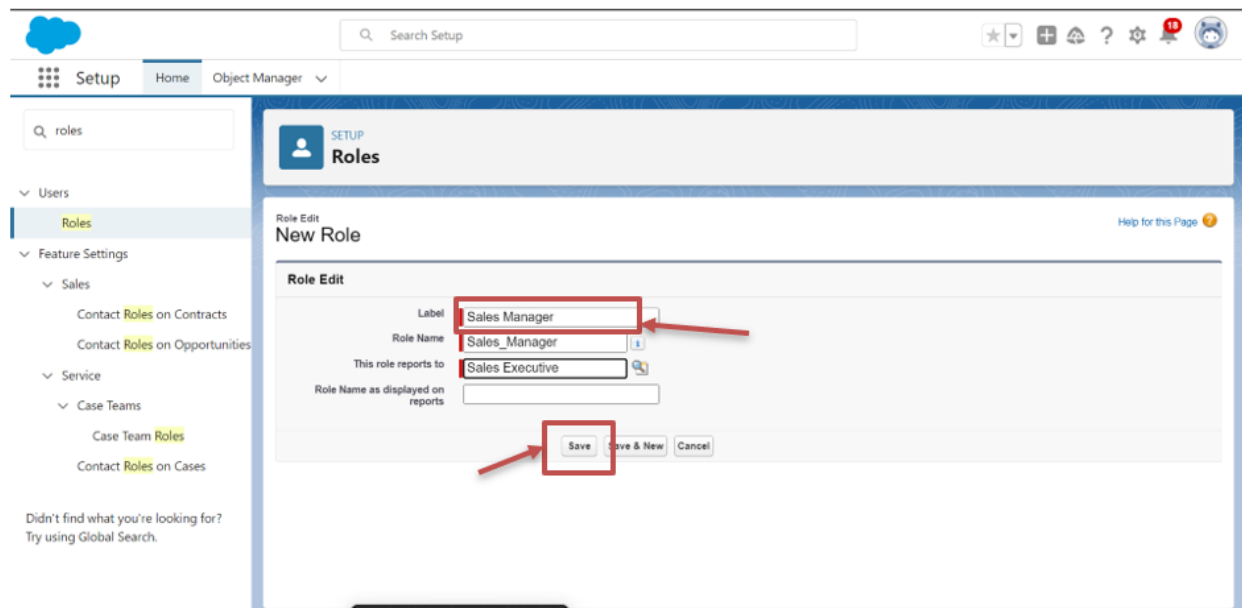


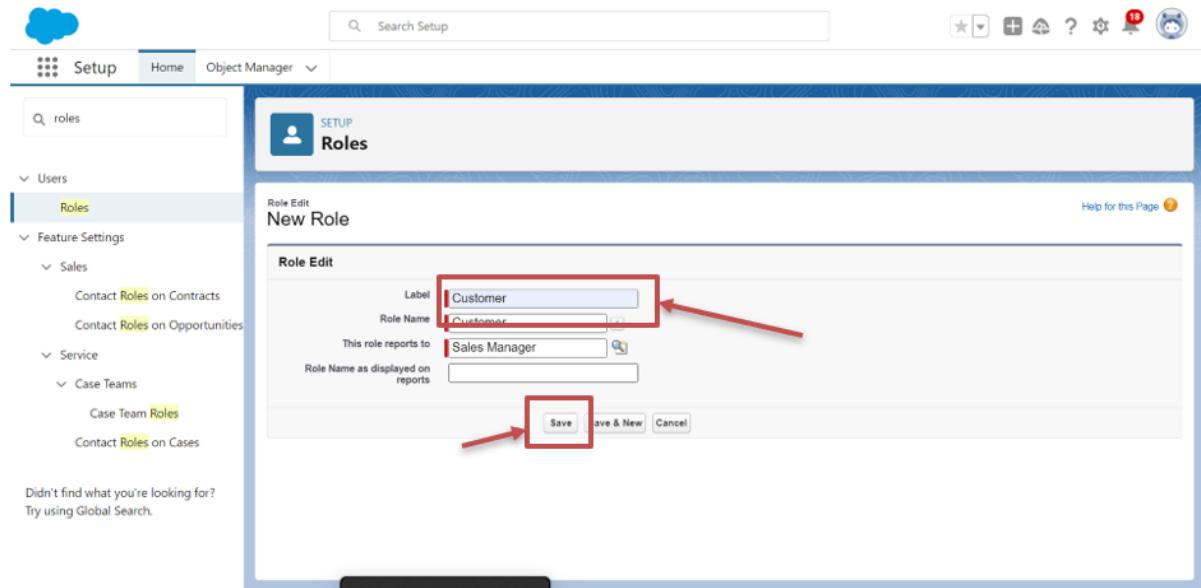
Activity 3: Create Roles Creating roles as per the business requirement.

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.

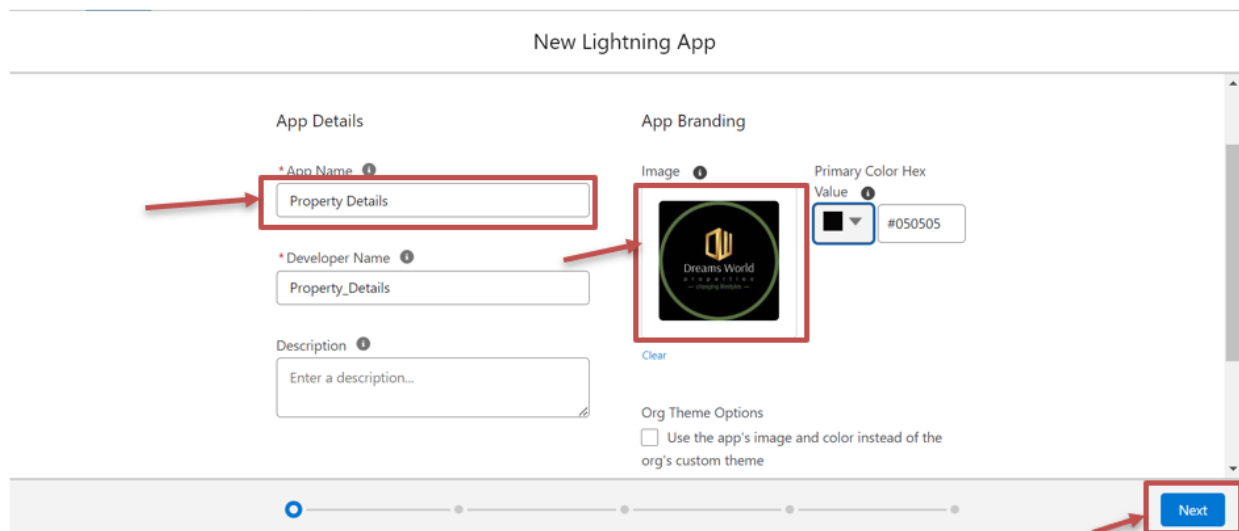




Activity 4: Create A Property Details App

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.



New Lightning App

App Options

Navigation and Form Factor ⓘ

* Navigation Style

- ☒ Standard navigation
- ☐ Console navigation

* Supported Form Factors

- ☒ Desktop and phone
- ☐ Desktop
- ☐ Phone

Setup and Personalization ⓘ

Setup Experience

- ☒ Setup (full set of Setup options)
- ☐ Service Setup

App Personalization Settings

- ☐ Disable end user personalization of nav items in this app
- ☐ Disable temporary tabs for items outside of this app
- ☐ Use Omni-Channel sidebar

Back

Next

New Lightning App

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Create ▼

groups

Property - Sheet1

Selected Items

Customer - Sheet1

Add

Back

Next

- Click Next and Next then add "System Admin" Profile and Save.

New Lightning App

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Create

prope

Selected Items

Customer - Sheet1

Property - Sheet1

Back

Next

New Lightning App

User Profiles

Choose the user profiles that can access this app.

Available Profiles

syst

Selected Profiles

System Administrator


Back

Save & Finish

Activity 5: Create Profiles

Customer

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


New Profile 			A B C D E F G H I J									
<input type="checkbox"/>	Action	Profile Name ↑	User License									
<input type="checkbox"/>	Edit Clone	Silver Partner User	Silver Partner									
<input type="checkbox"/>	Edit Clone	Solution Manager	Salesforce									
<input type="checkbox"/>	Edit Clone	Standard Platform User	Salesforce Platform									
<input type="checkbox"/>	Edit Clone	Standard User	Salesforce									
<input type="checkbox"/>	Edit Clone	System Administrator	Salesforce									


Clone Profile

Enter the name of the new profile.

You must select an existing profile to clone from.

Existing Profile	Standard Platform User
User License	Salesforce Platform
Profile Name	<input type="text" value="Customer"/>



 **Save** **Cancel**

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

Custom App Settings ! * Required Information

	Visible	Default		Visible	Default
Analytics Studio (standard__Insights)	<input type="checkbox"/>	<input type="radio"/>	Playground Starter (trihdtps__Playground_Starter)	<input type="checkbox"/>	<input type="radio"/>
App Launcher (standard__AppLauncher)	<input type="checkbox"/>	<input type="radio"/>	Property Details (Property_Details)	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
Platform (standard__Platform)	<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 read and view all in “Property”

SETUP Profiles

Contact Point Consents: ☐ ☐ ☐ ☐ ☐ ☐ ☐

Contact Point Emails: ☐ ☐ ☐ ☐ ☐ ☐ ☐

Streaming Channels: ☐ ☐ ☐ ☐ ☐ ☐ ☐

User External Credentials: ☐ ☐ ☐ ☐ ☐ ☐ ☐

Custom Object Permissions

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Customer - Sheet1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Property - Sheet1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Manager

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

SETUP Profiles

Clone Profile [Help for this Page](#)

Enter the name of the new profile.

You must select an existing profile to clone from. ⓘ Required Information

Existing Profile	Standard Platform User
User License	Salesforce Platform
Profile Name	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”.

SETUP
Profiles

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?](#)

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Accounts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assets	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Authorization Forms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Data Uses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authorization Form Texts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Background Operations	<input type="checkbox"/>					
Business Brands	<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"/>
Contact Point Phones	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Contact Point Type Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D&B Companies	<input type="checkbox"/>					
Data Use Legal Bases	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Data Use Purposes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Documents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Engagement Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ideas	<input type="checkbox"/>	<input type="checkbox"/>				

SETUP
Profiles

Contact Point Emails
☐
☐
☐
☐
☐
☐

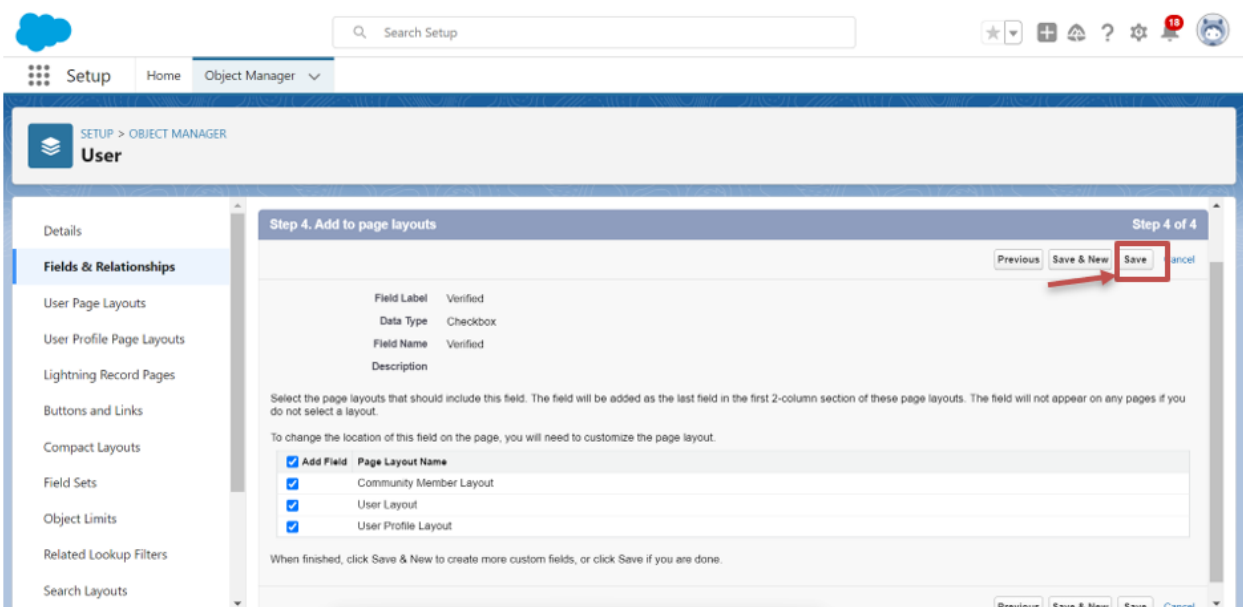
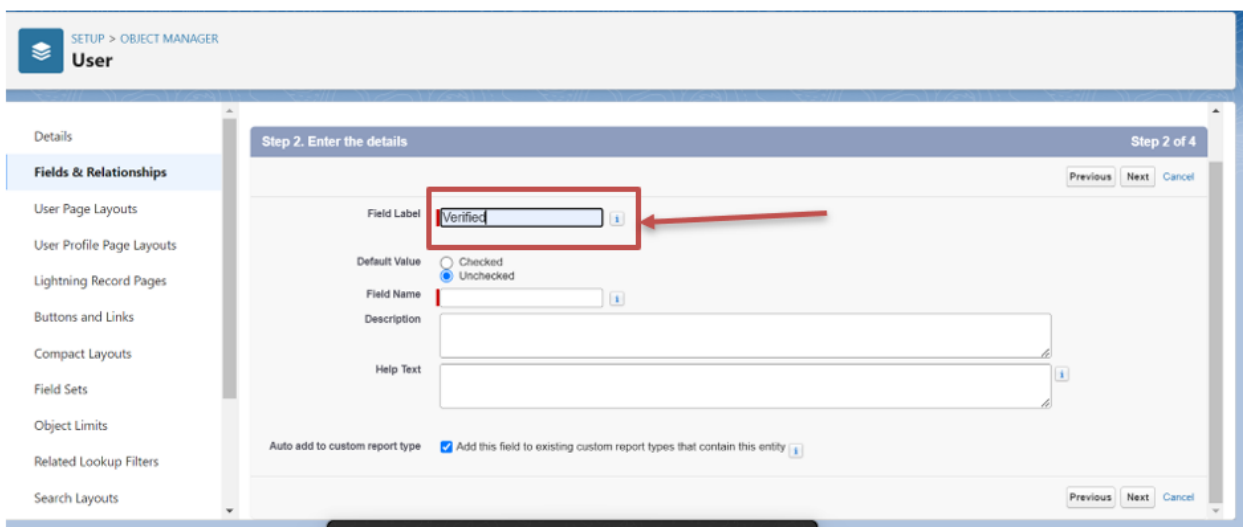
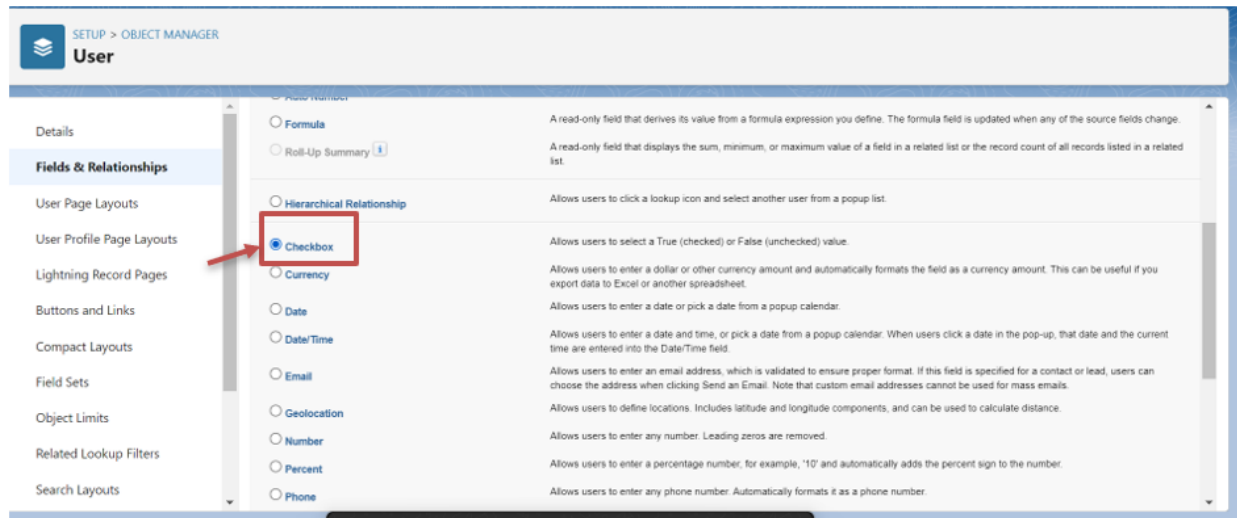
User External Credentials
☐
☐
☐
☐
☐

Custom Object Permissions

	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Customer - Sheet1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Property - Sheet1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

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”



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

The screenshot shows the Salesforce Setup interface. The left sidebar contains a search bar with 'user' and a list of navigation items: Users, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, Feature Settings, Data.com, Prospector Users, Service, and Embedded Service. The main content area is titled 'Users' and shows the 'User Edit' form for a user named 'Executive'. The form has tabs for 'General Information', 'Permissions', and 'Advanced Settings'. The 'General Information' tab is active, showing fields for First Name, Last Name (Executive), Alias (exec), Email (21052678@kiit.ac.in), Username (salesexecutive1@gamil.cor), Nickname (seu1), Title, Company, Department, and Division. On the right, there are dropdown menus for Role (Sales Executive), User License (Salesforce), and Profile (System Administrator). Below these are checkboxes for Active (checked), Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, and Site.com Publisher User. A 'Help for this Page' link is visible in the top right corner.

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

The screenshot shows the Salesforce Setup interface with the 'Users' section selected. The 'User Edit' form for a user named 'Manager' is displayed. The form includes fields for General Information and a list of roles with checkboxes.

General Information		Role
First Name		Sales Manager
Last Name	Manager	User License
Alias	mana	Salesforce Platform
Email	21052678@kiit.ac.in	Profile
Username	salesmanageru2@gmail.co	Manager
Nickname	smu2	Active
Title		<input checked="" type="checkbox"/>
Company		Marketing User
Department		<input type="checkbox"/>
Division		Offline User
		<input type="checkbox"/>
		Knowledge User
		<input type="checkbox"/>
		Flow User
		<input type="checkbox"/>
		Service Cloud User
		<input type="checkbox"/>
		Site.com Contributor User
		<input type="checkbox"/>
		Site.com Publisher User
		<input type="checkbox"/>

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

The screenshot shows the Salesforce Setup interface with the 'Users' section selected. The 'User Edit' form for a user named 'Customer' is displayed. The form includes fields for General Information and a list of roles with checkboxes.

General Information		Role
First Name		Customer
Last Name	Customer	User License
Alias	cust	Salesforce Platform
Email	21052678@kiit.ac.in	Profile
Username	customer3@gmail.com	Customer
Nickname	cu3	Active
Title		<input checked="" type="checkbox"/>
Company		Marketing User
Department		<input type="checkbox"/>
Division		Offline User
		<input type="checkbox"/>
		Knowledge User
		<input type="checkbox"/>
		Flow User
		<input type="checkbox"/>
		Service Cloud User
		<input type="checkbox"/>
		Site.com Contributor User
		<input type="checkbox"/>
		Site.com Publisher User
		<input type="checkbox"/>

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

The screenshot shows the Salesforce 'User Edit' interface for a user named 'Customer2'. The left sidebar contains navigation links: Setup, Home, Object Manager, Users, Permission Set Groups, Permission Sets, Profiles, Public Groups, Queues, Roles, User Management Settings, Users (highlighted), Feature Settings, Data.com, Prospector Users, Service, and Embedded Service. The main content area is titled 'User Edit Customer2' and includes a 'Help for this Page' link. Below the title are 'Save', 'Save & New', and 'Cancel' buttons. The 'General Information' section contains the following fields: First Name (empty), Last Name (Customer2), Alias (cust2), Email (21052678@kiit.ac.in), Username (customer2u4@gmail.com), Nickname (c2u4), Title (empty), Company (empty), Department (empty), and Division (empty). On the right side, there are dropdown menus for Role (Customer), User License (Salesforce Platform), and Profile (Customer). Below these are checkboxes for Active (checked), Marketing User, Offline User, Knowledge User, Flow User, Service Cloud User, Site.com Contributor User, and Site.com Publisher User. A red bar at the bottom of the form indicates required information.

Activity 8: Create An Approval Process for Property Object

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

Setup Home Object Manager

Search Setup

Q proc

Feature Settings

- Marketing
 - Lead Processes
- Sales
 - Sales Processes
- Service
 - Entitlement Management
 - Entitlement Processes
 - Support Processes
- Process Automation
 - Approval Processes
 - Automation Home (Beta)
 - Flows
 - Migrate to Flow

Approval Processes

Approval Process Edit

Property Approval

Step 1 of 6

Save Next Cancel

Enter a name and description for your new approval process.

Enter Name and Description

Process Name Property Approval

Unique Name Property_Approval

Description

Required Information

Save Next Cancel

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

Setup Approval Processes

Step 2 of 6

Previous Save Next Cancel

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

Specify Entry Criteria

Use this approval process if the following criteria are met

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

Add Filter Logic...

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.

Property Approval

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.
7. Click Next and Select the Initial Submitters
 - a. Owner - Property Owner
 - b. Roles - Sales Manager

Initial Submitters

Submitter Type Search: **Owner** for: Find

Available Submitters

--None--

Allowed Submitters

Role: Sales Manager
Property Owner

Add
Remove

8. Save.
9. Add an approval step name "Executive Approval"

Executive Approval

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: Executive Approval

Unique Name: Executive_Appr

Description: Executive Approval

Save Next Cancel

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.

SETUP

Field Updates

Identification

! = Required Information

Name: Verified Property

Unique Name: Verified_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

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.

SETUP
Field Updates

Identification

Name: UnVerified Property

Unique Name: UnVerified_Property

Description:

Object: Property

Field to Update: Property: Verified

Field Data Type: Checkbox

Re-evaluate Workflow Rules after Field Change: ☐

Specify New Field Value

Checkbox Options:

☐ True

☒ False

Save Save & New Cancel

14. Activate the Approval Process.

SETUP
Approval Processes

Approval Processes
Property: Property Approval
Back to Approval Process List

Process Definition Detail

Process Name: Property Approval

Unique Name: Property_Approval

Description: (Property: Location NOT EQUAL TO Blank) AND (Property: Verified EQUALS False)

Entry Criteria: (Property: Location NOT EQUAL TO Blank) AND (Property: Verified EQUALS False)

Record Editability: Administrator OR Current Approver

Approval Assignment Email Template: Role: Sales Manager, Property Owner

Initial Submitters: Jyoti Gupta, 24/06/2024, 2:27 pm

Created By: Jyoti Gupta, 28/06/2024, 3:18 pm

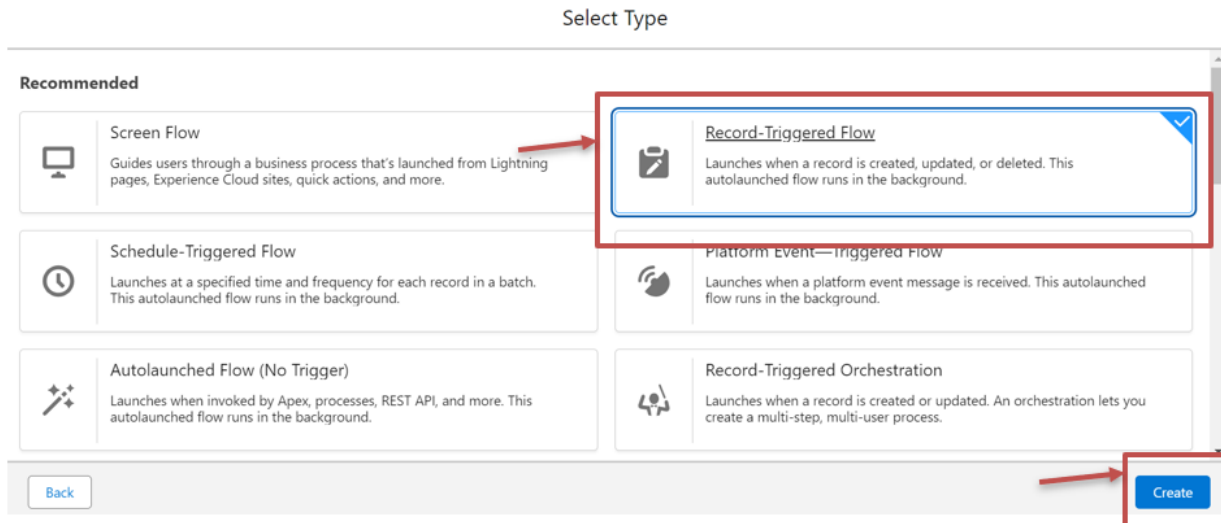
Activate

Initial Submission Actions

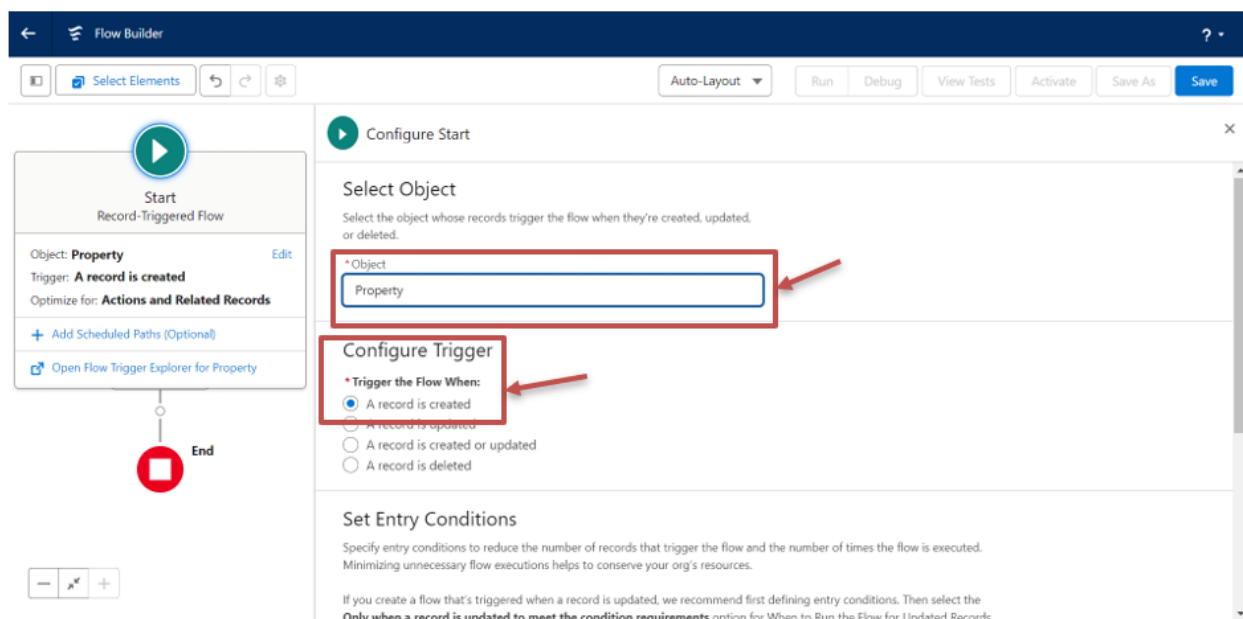
Action Type	Description
-------------	-------------

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”

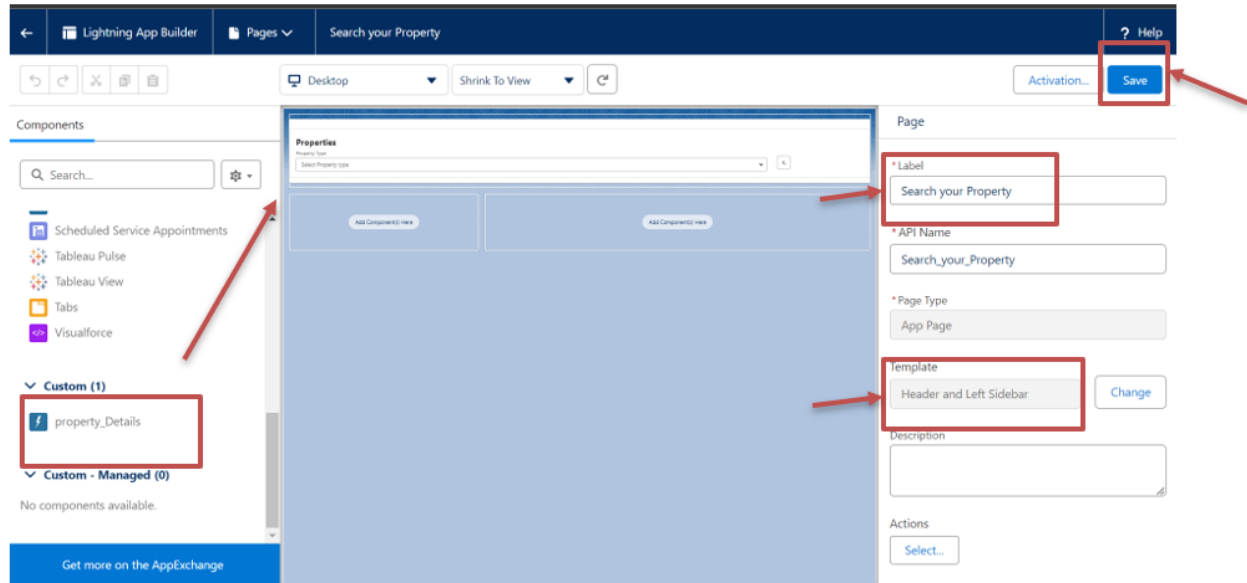


4. Set Entry Conditions - “None”
5. Add a “Action” - “Submit for Approval”
6. Give Label - Approval for property
7. Record Id - {\$Record.Id}
8. Done.

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

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];
    }
}
```



```

1 public class PropertyHandler_LWC{
2
3     @AuraEnabled(cacheable=true)
4
5     public static list<Property__c> getProperty(string type , boolean verified){
6
7         return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c FROM Property__c
8                 Where Type__c =: type AND Verified__c =: verified];
9
10    }
11
12 }
13 }

```

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;">Properties</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">

<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={istru}>
    <div class="slds-box">
        <lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-datatable>
    </div>
</template>
<template if:false={isfalse}>
    <div class="slds-box">
        <div style="font-size: 15px;"><b>No properties Are Found !!</b></div>
    </div>
</template>

</lightning-card>
</template>

```

```

1  <template>
2
3      <lightning-card>
4
5          <div class="slds-box">
6
7              <div class="slds-text-align_left">
8
9                  <h1 style="font-size: 20px;"><b>Properties</b></h1>
10
11              </div>
12
13              <div>
14
15                  <div class="slds-grid slds-gutters">
16
17                      <div class="slds-col slds-size_5-of-6">
18
19                          <lightning-combobox name="Type" label="Property Type" values={typevar} placeholder="Select Property type"
20
21                          options={propetyoptions} onchange={changehandler}></lightning-combobox>
22
23                      </div>
24
25                      <div class="slds-col slds-size_1-of-6">
26
27                          <br>
28
29                          <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search"
30
31                          label="Search" onclick={handleClick}></lightning-button-icon>
32

```

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)
            })
    }

}

```

```

1  import { LightningElement, api, track, wire } from 'lwc';
2
3  import getProperty from "@salesforce/apex/PropertyHandler_LWC.getProperty";
4
5  import { getRecord } from 'lightning/uiRecordApi';
6
7  import USER_ID from '@salesforce/user/Id';
8
9  export default class C_01_Property_Management extends LightningElement {
10
11      @api recordId
12
13      userId = USER_ID;
14
15      verifiedvar
16
17      typevar
18
19      isfalse = true;
20
21      istrue = false;
22
23      @track propertylist = [];
24
25      columns = [
26
27          { label: 'Property Name', fieldName: 'Property_Name__c' },
28
29          { label: 'Property Type', fieldName: 'Type__c' },
30
31          { label: 'Property Location', fieldName: 'Location__c' },
32

```

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>

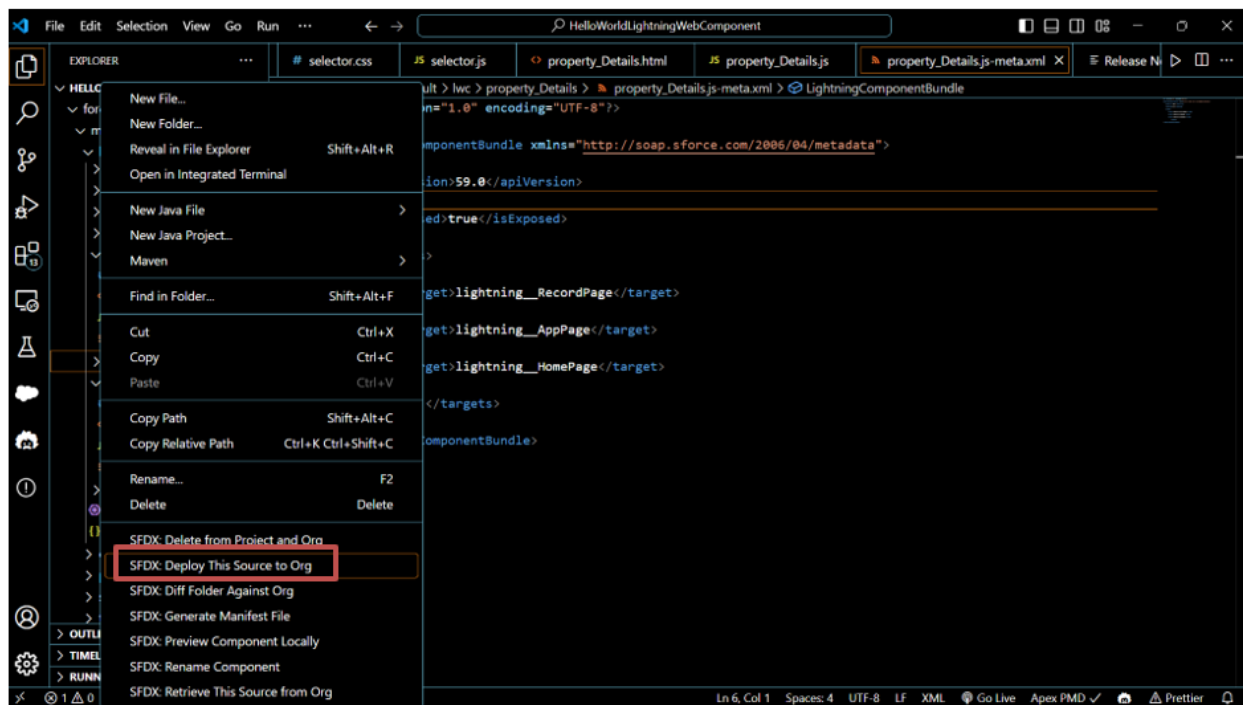
```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
4
5      <apiVersion>59.0</apiVersion>
6
7      <isExposed>true</isExposed>
8
9      <targets>
10
11          <target>lightning__RecordPage</target>
12
13          <target>lightning__AppPage</target>
14
15          <target>lightning__HomePage</target>
16
17      </targets>
18
19 </LightningComponentBundle>

```

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

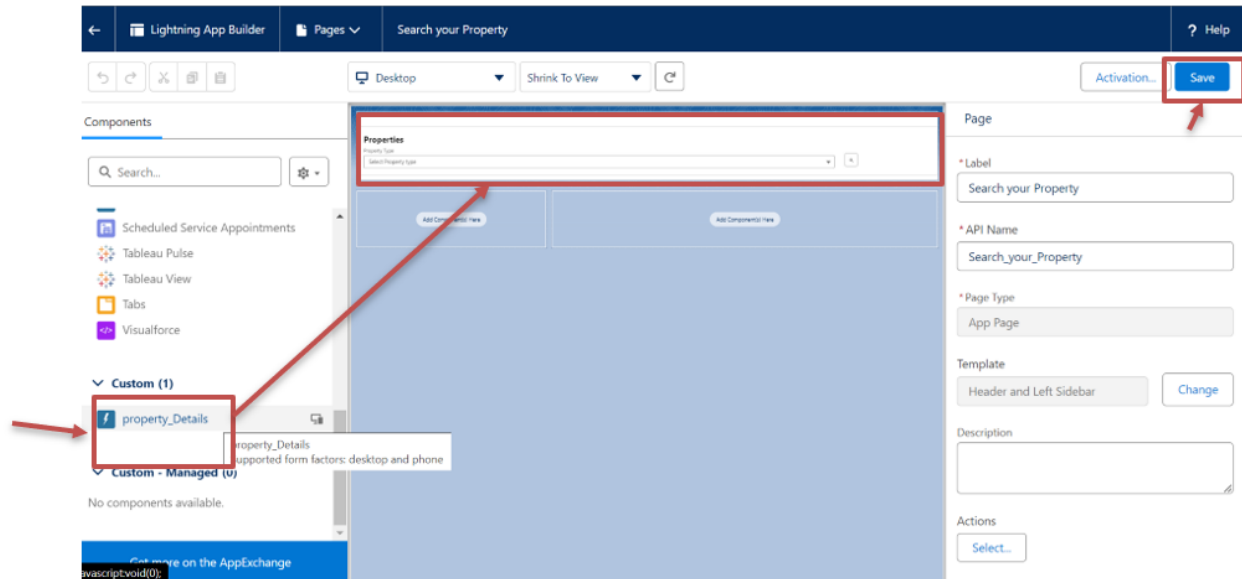


Activity 12: Drag This Component To Your App Page

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”

