

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



**Project by ,**  
Ambati Madhavi,  
[ambatimadhavi213@gmail.com](mailto:ambatimadhavi213@gmail.com)

## About CRM Application :

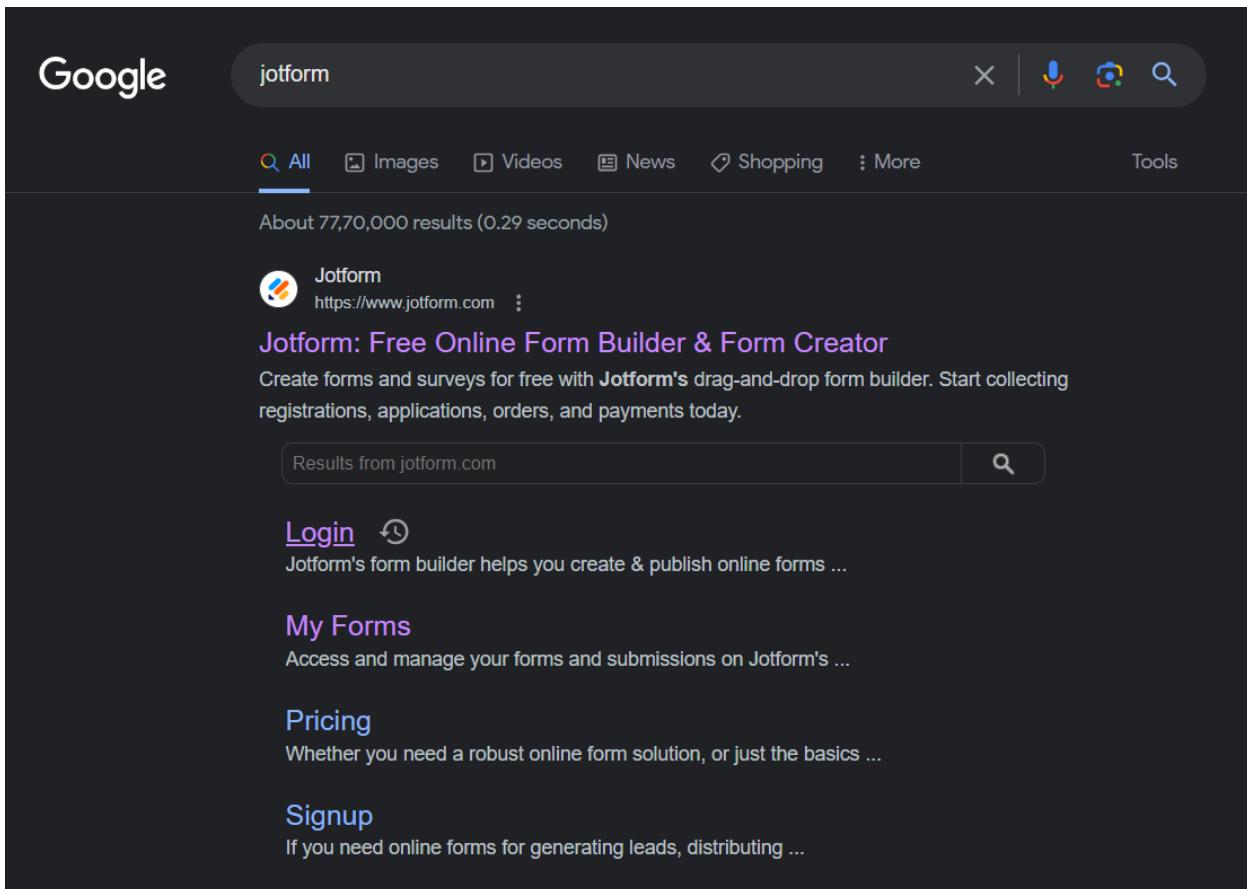
Dreams World Properties integrates Salesforce to streamline customer interactions. Website engagement triggers automated record creation in Salesforce, capturing customer details and preferences. Salesforce categorizes users as approved or non-approved, offering tailored property selections to approved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

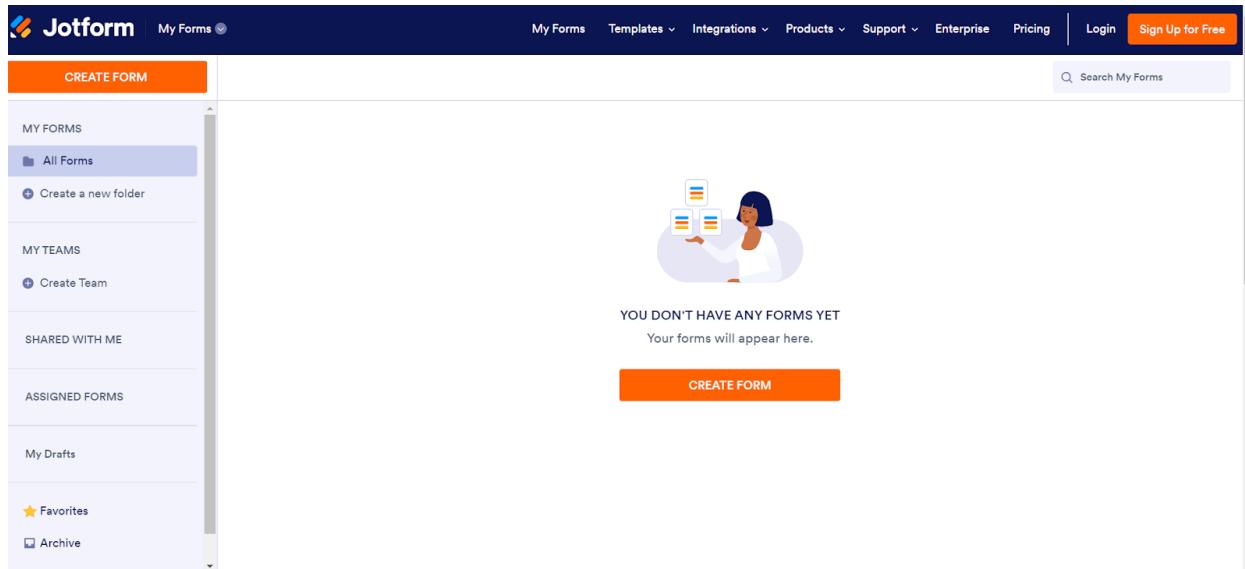
Client wants a form for the customers to get the details directly into the salesforce so that the admins can create a user in the org.

## Activity 1

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.

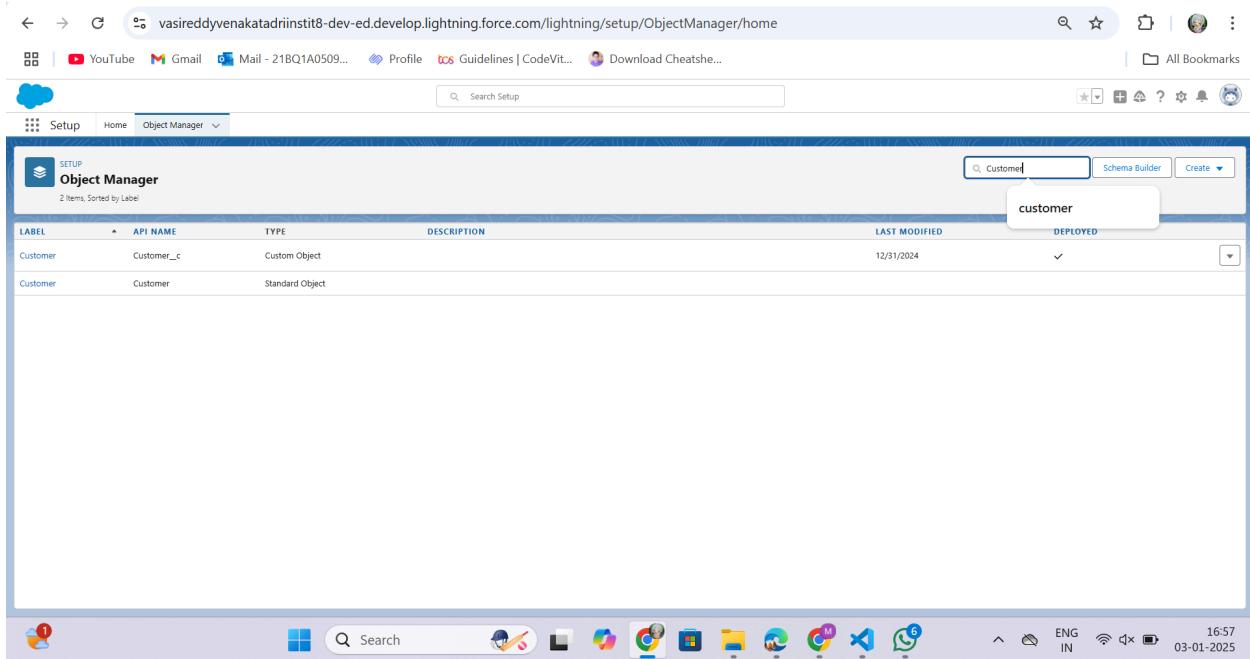
The screenshot shows a Jotform form titled 'Dreams World'. The form includes fields for 'Name' (First Name and Last Name), 'Email' (example: wwww@sample.com), 'Phone Number' (0000 0000), 'Which type of Property are you looking for?' (Residential, Commercial, Rental), 'Budget Amount' (1000-1000), 'Address' (Street Address, Street Address Line 2, City, State / Province, Postal / Zip Code), and a 'Submit' button. The form is currently in 'BUILD' mode. The browser address bar shows 'jotform.com/build/243653345461053'. The taskbar at the bottom shows various application icons.

## 2.Create Objects from Spreadsheet.

Directly Creating Objects from Spreadsheet in Salesforce

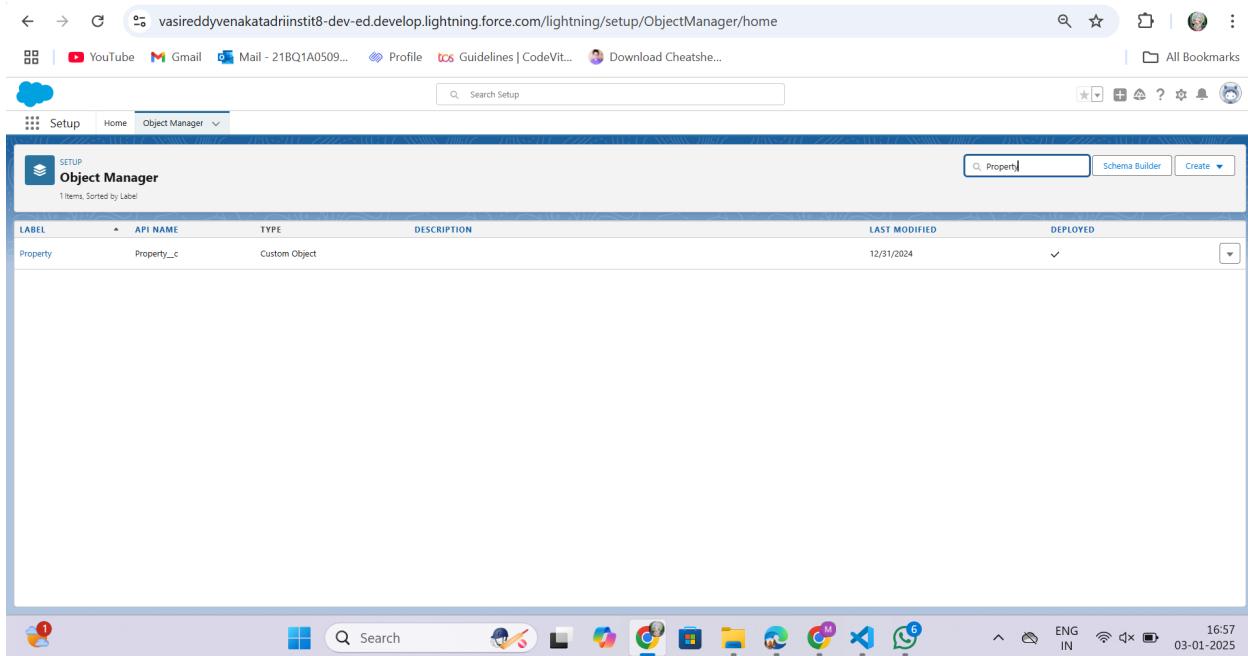
## Create Customer object

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 steps from the customer object to create the Property Object
2. Property



### 3. 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.

A screenshot of the Jotform platform. The top navigation bar includes "My Forms", "Templates", "Integrations", "Products", "Support", "Enterprise", "Pricing", "Login", and "Sign Up for Free". The left sidebar has sections for "MY FORMS" (All Forms, Create a new folder), "MY TEAMS" (Create Team), "SHARED WITH ME", "ASSIGNED FORMS" (My Drafts), and "Favorites" (Archive). The main area features a "CREATE FORM" button. To the right, there's a "FEATURED INTEGRATIONS" section with icons for PayPal, Slack, Google Sheets, Mailchimp, Zoom, Dropbox, Google Calendar, and Hubspot, along with a "See more" link. A message at the bottom says "YOU DON'T HAVE ANY FORMS YET" and "Your forms will appear here." with a "CREATE FORM" button.

2. Click on User Integration and choose "Add to Form".

**Salesforce**

Watch Video    Use Integration ▾

Go to AppExchange    Add to Form

Add to Form

**Jotform** + **salesforce**

Send new leads, contacts, or accounts to your sales CRM

Salesforce is a powerful CRM tool used to help salespeople manage accounts, improve client relations, and close big deals. To automatically sync Jotform submissions to your Salesforce account, click the 'Add to Form' button above.

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

Form Settings    Form Builder

Dreams World    All changes saved at 2:43 PM

Add Collaborators    Help    Logout

BUILD    SETTINGS    PUBLISH    Preview Form

Salesforce via this handy integration.

Use this integration to:

- Add new company leads
- Add new contacts
- Add new accounts
- Connect any custom object

By gathering leads through Jotform and automatically relaying them to your sales pipeline — using either standard or custom objects — you can save time on manual processes and spend more time connecting with customers and closing important deals with your team. Get started with a [Salesforce form](#) now!

Learn how to integrate your form with [Salesforce](#).

Authentication

Select your Salesforce account to create an integration.

Select a Salesforce account

dada rao - prajwal@thesmartbridge.com

Send Feedback

### 4. Select an Action - Create a record.

Select a Salesforce Object : - Customer

Dreams World  
All changes saved at 2:43 PM

**BUILD** **SETTINGS** **PUBLISH**

**Create a record**  
Generate a new record or update an existing record

**Find existing record**  
Find an existing record or create a new one if it doesn't already exist

Select a Salesforce Object

Customer

**Object Fields** **Dreams World**

Customer → Name - First Name

City → Address - City

Budget Amount → Budget Amount

Property Type → Which type of Property ar...

Phone Number → Phone Number

Street Address → Address - Street Address

Email → Email

Customer Name → Name - Last Name

State → Address - State

Street Address line 2 → Address - Street Address 2

**Update an existing record**  
Update Salesforce data when the selected field values match an existing record

**Send Feedback**

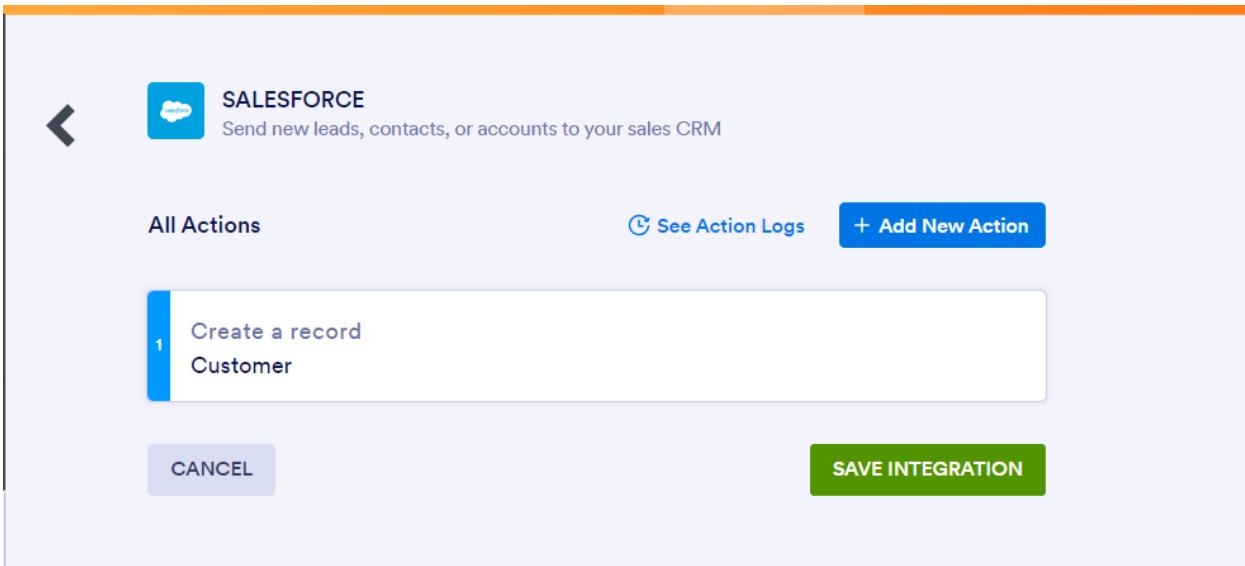
## 5. Map Each and every field on the Object with the fields on the form and “Save Action”.

**Create a record**  
Send data from form fields to matched Salesforce fields

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

**+ Add Field**

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



## 1.Create Roles

Create Roles as per business requirement.

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

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

2) Label - Sales Executive

## Reports to - Sales Representative

The screenshot shows the Salesforce Setup Roles page. At the top, there's a header with a user icon, 'SETUP', and 'Roles'. Below it, the title 'Role Edit' and 'New Role' are displayed. The main form has the following fields:

- Label:** Sales Executive
- Role Name:** Sales\_Executive
- This role reports to:** Sales Representative
- Role Name as displayed on reports:** (empty field)

At the bottom right of the form are three buttons: 'Save', 'Save & New', and 'Cancel'.

Similarly Create a Role Name “Sales Manager” below Sales Executive which reports to Sales Executive, Also Add a Role below Sales Manager labeled as “Customer” which reports to Sales Manager.

The screenshot shows the Salesforce Setup Roles page. On the left, a sidebar navigation includes 'Users', 'Roles' (which is selected), 'Feature Settings', 'Sales' (with sub-options 'Contact Roles on Contracts', 'Contact Roles on Opportunities'), 'Service' (with sub-options 'Case Teams', 'Case Team Roles', 'Contact Roles on Cases'), and a search bar. The main area displays a hierarchical list of roles:

- SVP\_Human Resources
- SVP\_Sales & Marketing
  - VP\_International\_Sales
  - VP\_Marketing
    - Marketing\_Team
  - VP\_North\_American\_Sales
    - Director\_Channel\_Sales
      - Channel\_Sales\_Team
    - Director\_Direct\_Sales
      - Eastern\_Sales\_Team
      - Western\_Sales\_Team
  - Sales\_Representative
    - Sales\_Executive
      - Sales\_Manager
        - Customer
- SF Admin

## 2.Create a Property Details App

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 Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

\*App Name

\*Developer Name

Description

App Branding

Image

Primary Color Hex Value

Org Theme Options  Use the app's image and color instead of the org's custom theme

App Launcher Preview

Next

- 2) Click Next >> Next >> Save and Add "System Admin "Profile.

### 3.Create Profiles

Create profiles as per business requirement

### Customer

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

The screenshot shows the Salesforce Setup interface with the 'Profiles' page selected. The left sidebar is expanded, showing categories like 'Users', 'Feature Settings', and 'User Interface'. The main content area displays a list of profiles. The columns are: Action, Profile Name, User License, and Custom. The profiles listed are:

Action	Profile Name	User License	Custom
<input type="checkbox"/> Edit   Clone	Salesforce API Only System Integrations	Salesforce Integration	<input type="checkbox"/>
<input type="checkbox"/> Edit   Clone	Silver Partner User	Silver Partner	<input type="checkbox"/>
<input type="checkbox"/> Edit   Clone	Solution Manager	Salesforce	<input type="checkbox"/>
<input type="checkbox"/> Edit   Clone	Standard Platform User	Salesforce Platform	<input type="checkbox"/>
<input type="checkbox"/> Edit   Clone	Standard User	Salesforce	<input type="checkbox"/>
<input type="checkbox"/> Edit   Clone	System Administrator	Salesforce	<input type="checkbox"/>

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

The screenshot shows the 'Profiles' page in the Salesforce Setup interface. A new profile named 'Customer' is being created. The 'Custom Profile' checkbox is checked. In the 'Custom App Settings' section, there are several objects listed with 'Visible' and 'Default' checkboxes. The objects and their settings are:

Object	Visible	Default
Analytics Studio (standard_insights)	<input type="checkbox"/>	<input type="radio"/>
App Launcher (standard_AppLauncher)	<input type="checkbox"/>	<input type="radio"/>
Dreams Houses (Dreams_Houses)	<input type="checkbox"/>	<input type="radio"/>
How We Roll Maintenance (How_We_Roll_Maintenance)	<input type="checkbox"/>	<input type="radio"/>
LWC Component (LWC_Component)	<input type="checkbox"/>	<input type="radio"/>
Platform (standard_Platform)	<input type="checkbox"/>	<input checked="" type="radio"/>
Property Details (Property_Details)	<input checked="" type="checkbox"/>	<input type="radio"/>
WDC (standard_Work)	<input type="checkbox"/>	<input type="radio"/>

## 3. Also Remove all the Standard Object Permissions.

**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 <a href="#">?</a>	Modify All <a href="#">?</a>	
Accounts	<input type="checkbox"/>	<input type="checkbox"/>					
Addresses	<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"/>					
Authorization Form Consents	<input type="checkbox"/>	<input type="checkbox"/>					
Authorization Form Data Uses	<input type="checkbox"/>	<input type="checkbox"/>					
Authorization Form Texts	<input type="checkbox"/>	<input type="checkbox"/>					
Background Operations	<input type="checkbox"/>						
Business Brands	<input type="checkbox"/>	<input type="checkbox"/>					
Communication Subscriptions	<input type="checkbox"/>	<input type="checkbox"/>					
Communication Subscription Channel Types	<input type="checkbox"/>	<input type="checkbox"/>					
Communication Subscription Consents	<input type="checkbox"/>	<input type="checkbox"/>					
Communication Subscription Timings	<input type="checkbox"/>	<input type="checkbox"/>					
Contacts	<input type="checkbox"/>	<input type="checkbox"/>					
Contact Point Phones	<input type="checkbox"/>	<input type="checkbox"/>					
Contact Point Type Consents	<input type="checkbox"/>	<input type="checkbox"/>					
Customers	<input type="checkbox"/>	<input type="checkbox"/>					
D&B Companies	<input type="checkbox"/>						
Data Use Legal Bases	<input type="checkbox"/>	<input type="checkbox"/>					
Data Use Purposes	<input type="checkbox"/>	<input type="checkbox"/>					
Documents	<input type="checkbox"/>	<input type="checkbox"/>					
Engagement Channel Types	<input type="checkbox"/>	<input type="checkbox"/>					
Ideas	<input type="checkbox"/>	<input type="checkbox"/>					
Individuals	<input type="checkbox"/>	<input type="checkbox"/>					
Locations	<input type="checkbox"/>						
Party Consents	<input type="checkbox"/>	<input type="checkbox"/>					
Push Topics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Sellers	<input type="checkbox"/>	<input type="checkbox"/>					

- Uncheck all the Custom Object Permissions and check read and view all in "Property"

	Basic Access					Data Administration	
	Read	Create	Edit	Delete	View All <a href="#">?</a>	Modify All <a href="#">?</a>	
Communication Subscriptions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Communication Subscription Channel Types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Communication Subscription Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Communication Subscription Timings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Contacts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Contact Point Addresses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Contact Point Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Contact Point Emails	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Individuals	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Locations	<input type="checkbox"/>						
Party Consents	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Push Topics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Sellers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Streaming Channels	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
User External Credentials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Basic Access					Data Administration	
	Read	Create	Edit	Delete	View All <a href="#">?</a>	Modify All <a href="#">?</a>	
Customer	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Customer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
error logs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Property	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Property	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Sales orders	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Vehicles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

## Manager

- 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.

Set the permissions and page layouts for this profile.

**Profile Edit**

Name: Manager  
User License: Salesforce Platform  
Description:

Custom Profile: ✓

**Custom App Settings**

	Visible	Default		Visible	Default
Analytics Studio (standard_Insights)	<input type="checkbox"/>	<input checked="" type="radio"/>	LWC Component (LWC_Component)	<input type="checkbox"/>	<input checked="" type="radio"/>
App Launcher (standard_AppLauncher)	<input type="checkbox"/>	<input checked="" type="radio"/>	Platform (standard_Platform)	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
Dreams Houses (Dreams_Houses)	<input type="checkbox"/>	<input checked="" 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 checked="" type="radio"/>	WDC (standard_Work)	<input type="checkbox"/>	<input type="radio"/>

I = Required Information

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

**Custom Object Permissions**

	Customer	Sales orders	Vehicles
Basic Access	Read: <input checked="" type="checkbox"/> Create: <input checked="" type="checkbox"/> Edit: <input checked="" type="checkbox"/> Delete: <input checked="" type="checkbox"/> View All: <input checked="" type="checkbox"/> Modify All: <input checked="" type="checkbox"/>	Read: <input type="checkbox"/> Create: <input type="checkbox"/> Edit: <input type="checkbox"/> Delete: <input type="checkbox"/> View All: <input type="checkbox"/> Modify All: <input type="checkbox"/>	Read: <input type="checkbox"/> Create: <input type="checkbox"/> Edit: <input type="checkbox"/> Delete: <input type="checkbox"/> View All: <input type="checkbox"/> Modify All: <input type="checkbox"/>
error logs	Read: <input type="checkbox"/> Create: <input type="checkbox"/> Edit: <input type="checkbox"/> Delete: <input type="checkbox"/> View All: <input type="checkbox"/> Modify All: <input type="checkbox"/>	Read: <input type="checkbox"/> Create: <input type="checkbox"/> Edit: <input type="checkbox"/> Delete: <input type="checkbox"/> View All: <input type="checkbox"/> Modify All: <input type="checkbox"/>	Read: <input type="checkbox"/> Create: <input type="checkbox"/> Edit: <input type="checkbox"/> Delete: <input type="checkbox"/> View All: <input type="checkbox"/> Modify All: <input type="checkbox"/>
Property	Read: <input checked="" type="checkbox"/> Create: <input checked="" type="checkbox"/> Edit: <input checked="" type="checkbox"/> Delete: <input checked="" type="checkbox"/> View All: <input checked="" type="checkbox"/> Modify All: <input checked="" type="checkbox"/>	Read: <input type="checkbox"/> Create: <input type="checkbox"/> Edit: <input type="checkbox"/> Delete: <input type="checkbox"/> View All: <input type="checkbox"/> Modify All: <input type="checkbox"/>	Read: <input type="checkbox"/> Create: <input type="checkbox"/> Edit: <input type="checkbox"/> Delete: <input type="checkbox"/> View All: <input type="checkbox"/> Modify All: <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
- Obfuscate secret answer for password resets
- Require a minimum 1 day password lifetime
- Don't immediately expire links in forgot

- Create a Check Box field on user

Create Field on the User as per the business requirement.

# Activity 1

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

The screenshot shows the Salesforce Object Manager interface. A custom field named 'Verified' has been created under the 'User' object. The field is of type 'Checkbox'. The details page for this field includes sections for 'Field Information', 'General Options', and 'Validation Rules'. The 'Field Information' section shows the field label 'Verified', field name 'Verified', and API name 'Verified\_\_c'. The 'General Options' section shows the default value as 'Unchecked'. There are no validation rules defined.

# 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. Last Name - 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. Last Name >> 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. Last Name >> 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. Last Name >> Customer2

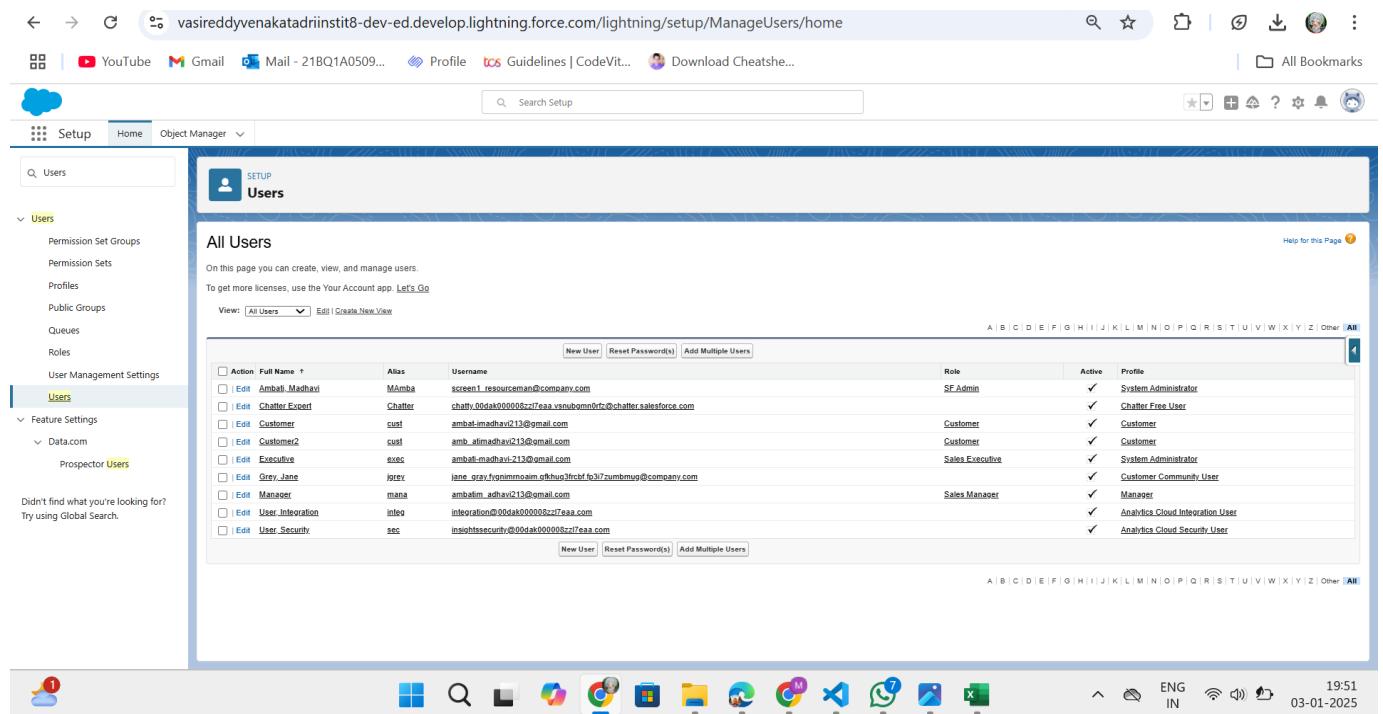
3. Role >> Customer

4. License >> Salesforce Platform

5. Profile >> Customer

6. Make Sure the verified check box is “checked”

## 7. Save



The screenshot shows the Salesforce Setup - Users page. The left sidebar is collapsed, and the main area displays a table of users. The table columns include Action, Full Name, Alias, Username, Role, Active, and Profile. The 'Active' column contains checked boxes for most users, except for 'Customer' and 'Sales Executive'. The 'Profile' column shows 'Customer' for most users, except for 'System Administrator' and 'Manager'. The 'Role' column includes 'SF Admin', 'Chatter Free User', 'Customer', 'Customer', 'Sales Executive', 'Sales Manager', and 'Analytics Cloud Integration User'. The 'Profile' column also lists 'System Administrator', 'Chatter Free User', 'Customer', 'Customer', 'System Administrator', 'Customer Community User', 'Manager', and 'Analytics Cloud Security User'. The 'Last Name' column header is visible at the top of the table.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Ambati_Madhavi	MAmba	screen_1_resourceman@company.com	SF Admin	<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Chatter Expert	Chatter	chatty.00da0000000000000000000000000000@chatter.salesforce.com	Chatter Free User	<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	Customer	cust	ambat-imadhev213@gmail.com	Customer	<input checked="" type="checkbox"/>	Customer
<input type="checkbox"/>	Customer2	cust	ambit-madhavi213@gmail.com	Customer	<input checked="" type="checkbox"/>	Customer
<input type="checkbox"/>	Executive	exec	ambat-madhavi213@gmail.com	Sales Executive	<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Grey_Jane	jorev	jane_gray.yogimmaan.dhruv1ref.fel3j7cumbug@company.com	Sales Manager	<input checked="" type="checkbox"/>	Customer Community User
<input type="checkbox"/>	Manager	mana	ambati_adhav213@gmail.com	Sales Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	User_Integration	inteo	integration@00da0000000000000000000000000000@feaa.com			Analytics Cloud Integration User
<input type="checkbox"/>	User_Security	sec	insightssecurity@00da000000000000@feaa.com			Analytics Cloud Security User

# Create an Approval Process for Property Object

An Approval process to approve or reject the records as according

## Activity 1

- From Setup >> Process Automation >> Approval Process

- Process Name - Property Approval

The screenshot shows the 'Approval Process Edit' screen for 'Property Approval'. It's 'Step 1 of 6'. The 'Enter Name and Description' section contains fields for 'Process Name' (Property Approval), 'Unique Name' (Property\_Approval), and 'Description' (left empty). A note says 'Enter a name and description for your new approval process.' Buttons at the bottom include 'Save', 'Next', and 'Cancel'.

3.Give 2 criteria -

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

The screenshot shows the 'Step 2. Specify Entry Criteria' screen. It lists two criteria: 'Property: Location' set to 'not equal to blank' and 'Property: Verified' set to 'equals False'. These are connected by 'AND' logic. Buttons at the bottom include 'Previous', 'Save', 'Next', and 'Cancel'.

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

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

Approval Process Edit  
Property Approval

Step 3. Specify Approver Field and Record Editability Properties

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.

Step 3 of 6

Previous Save Next Cancel

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

Step 5. Select Fields to Display on Approval Page Layout

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 Last Modified By Property link Verified	Property Owner Location Property Name Type

Add Remove Up Down

Click here to view an example

Step 5 of 6

Previous Save Next Cancel

- 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

Step 1. Enter Name and Description Step 1 of 3

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

Name: **VP Approval** Unique Name: **VP\_Approval** Description:

Save Next Cancel

10) specify the Criteria >> All record should enter

Approval Step Edit  
VP Approval

Step 2. Specify Step Criteria Step 2 of 3

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 [criteria are met] , else [reject record]:

Previous Save Next Cancel

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

Approval Step Edit  
VP Approval

Step 3. Select Assigned Approver Step 3 of 3

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).  
User: **dada rao**

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”**

1. Select Object >> Property
2. Field to Update >> Verified
3. Field Data Type >> CheckBox
4. Select CheckBox Option as “True”
5. Save.

The screenshot shows the 'Step 3. Select Assigned Approver' page of the Approval Step Edit process for 'VP Approval'. The page title is 'Approval Step Edit' and the sub-section is 'VP Approval'. At the top right, there are 'Help for this Page' and 'Step 3 of 3' buttons. Below the title, a note says '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.' There are four radio button options under '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):

Below this, there is a section for 'When multiple approvers are selected':

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

At the bottom, there is a checkbox:  The approver's delegate may also approve this request. On the far right, there are 'Previous', 'Save', and 'Cancel' buttons.

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

1. Select Object >> Property
2. Field to Update >> Verified
3. Field Data Type >> CheckBox
4. Select CheckBox Option as “False”
5. Save.

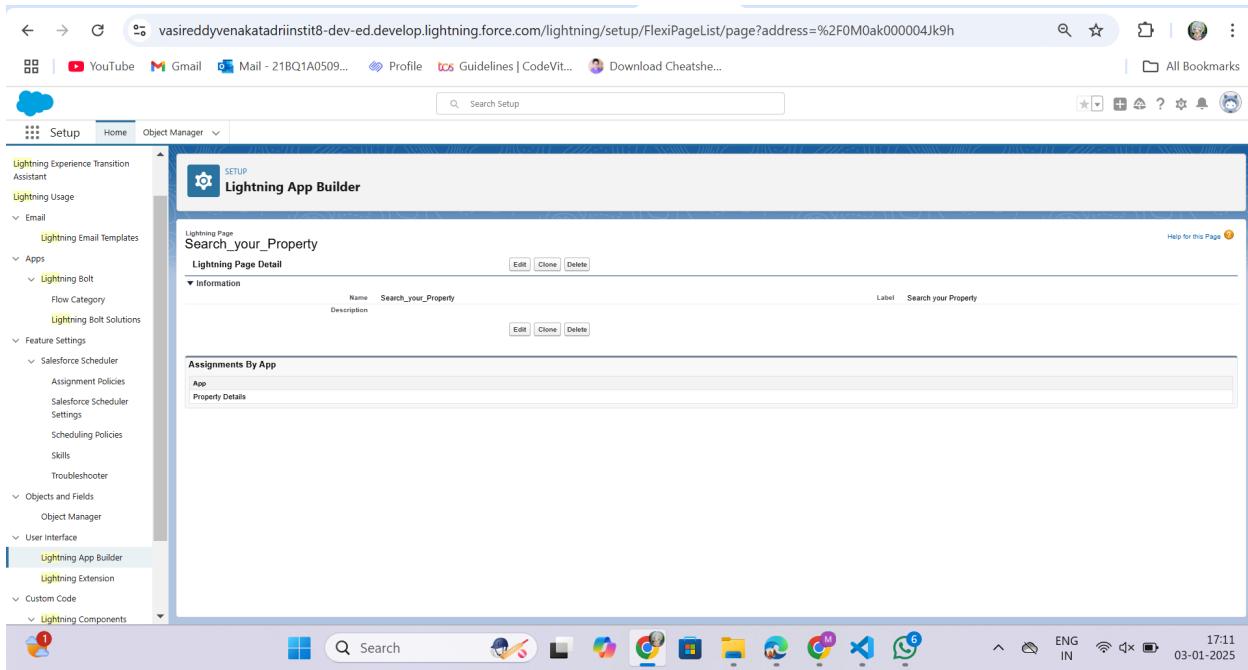
The screenshot shows the 'Field Update Edit' page in the Salesforce Setup. The 'Identification' section contains fields for Name (Unverified Property), Unique Name (Unverified\_Property), and Description. The 'Object' field is set to Property. The 'Field to Update' field is set to Property.Verified. The 'Field Data Type' is set to Checkbox. A checkbox labeled 'Re-evaluate Workflow Rules after Field Change' is checked. The 'Specify New Field Value' section shows 'Checkbox Options' with 'False' selected. At the bottom, there are 'Save', 'Save & New', and 'Cancel' buttons.

14) Activate the Approval Process.

## Create a Record trigger flow to submit the Approval Process Automatically.

A flow that can submit the records directly for approval

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



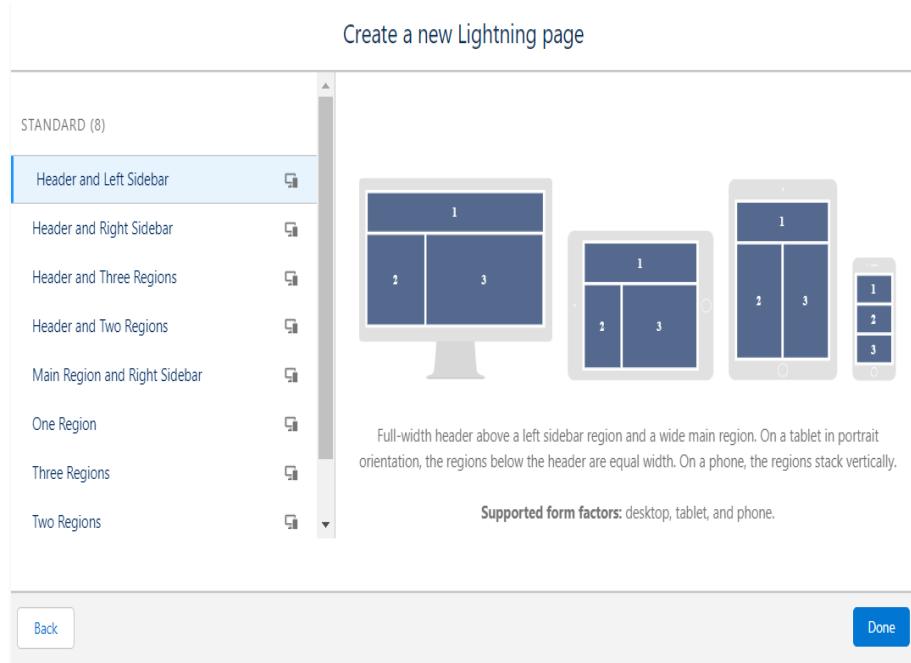
## Create an App Page

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

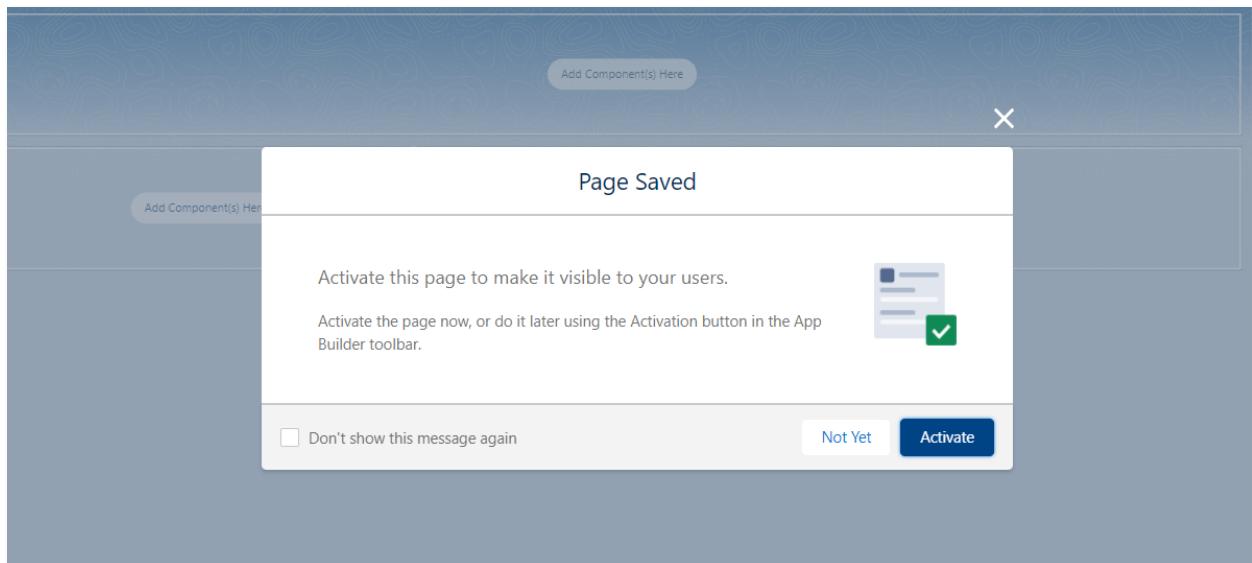
1. From Setup >> Go to Lightning App Builder >> Click on New >> Select App Page and Click on Next.

The screenshot shows the first step of the 'Create a new Lightning page' wizard. The left sidebar lists options: App Page (selected), Home Page, Record Page, Embedded Service Page, and Voice Extension. The main area displays a preview of the app page with a header 'My Custom App' and 'My App Page'. Below the header are two cards: 'Pipeline by Stage' (showing a donut chart with '1.8M') and 'Recent Items (20)' (listing items from Acme Industries). At the bottom right is a blue 'Next' button.

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”.

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

- Lightning Bolt
- LightningInstrumentation
- LWC Component
- Property Details**
- Queue Management
- Sales
- Sales Console
- Salesforce CMS

Property Details

Remove page

Search Your Property

Search your property

Cancel Save

This screenshot shows the 'Activation' screen in the Salesforce setup. It's titled 'Activation: Search your property'. At the top, there are tabs for 'PAGE SETTINGS', 'LIGHTNING EXPERIENCE' (which is selected), and 'MOBILE NAVIGATION'. A message says 'Add this app page to Lightning Experience apps. You can manage Lightning apps in Setup.' Below this, a section titled 'Add to Lightning Apps' lists various apps. The 'Property Details' app is highlighted with a blue checkmark. To the right, under 'Mobile Navigation', the 'Property Details' app is listed with its icon and name. There are buttons for 'Remove page' and 'Save'. At the bottom right are 'Cancel' and 'Save' buttons.

## 6. Then Click on "Save"

This screenshot shows a web browser window with the URL 'vasireddyvenakatadriinstit8-dev-ed.develop.lightning.force.com/lightning/n/Search\_your\_Property'. The page title is 'Search your Property'. The main content area is titled 'Properties' and has a 'Property Type' dropdown menu labeled 'Select Property type'. The browser's address bar and taskbar are visible at the bottom.

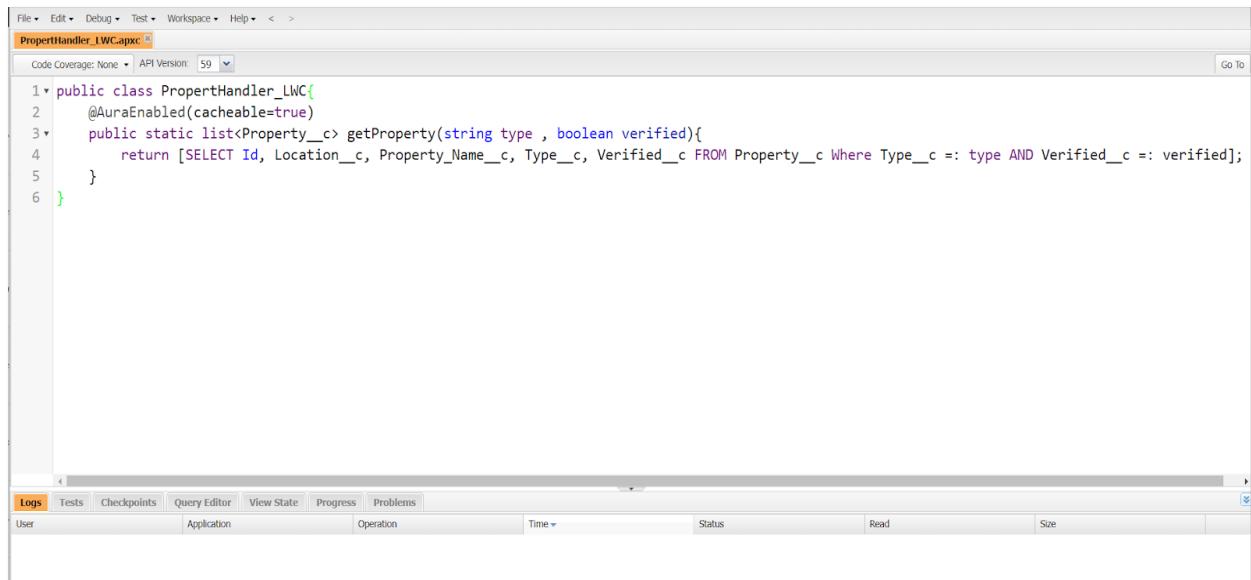
# Create a LWC Component

- 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) and Create a lightning Web Component and Name it

Anything you want to. (Example - )

5. In your Html File Write this code :-

Code :-

```
<template>
  <lightning-card>
    <div class="slds-box">
      <div class="slds-text-align_left">
        <h1 style="font-size: 20px;"><b>Properties</b></h1>
      </div>
      <div>
        <div class="slds-grid slds-gutters">
          <div class="slds-col slds-size_5-of-6">
            <lightning-combobox name="Type" label="Property Type" value={typevar}
placeholder="Select Property type"
options={propertyoptions} onchange={changehandler}></lightning-combobox>
          </div>
          <div class="slds-col slds-size_1-of-6">
            <br>
            <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-
text="Search"
label="Search" onclick={handleClick}></lightning-button-icon>
          </div>
        </div>
      </div>
      <template if:true={istrue}>
        <div class="slds-box">
          <lightning-datable key-field="id" data={propertylist}
columns={columns}></lightning-datable>
        </div>
      </template>
      <template if:false={isfalse}>
        <div class="slds-box">
          <div style="font-size: 15px;"><b>No properties Are Found !!</b></div>
        </div>
      </template>
    </div>
  </lightning-card>
</template>
```

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

```

} else {
    console.error(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 :-

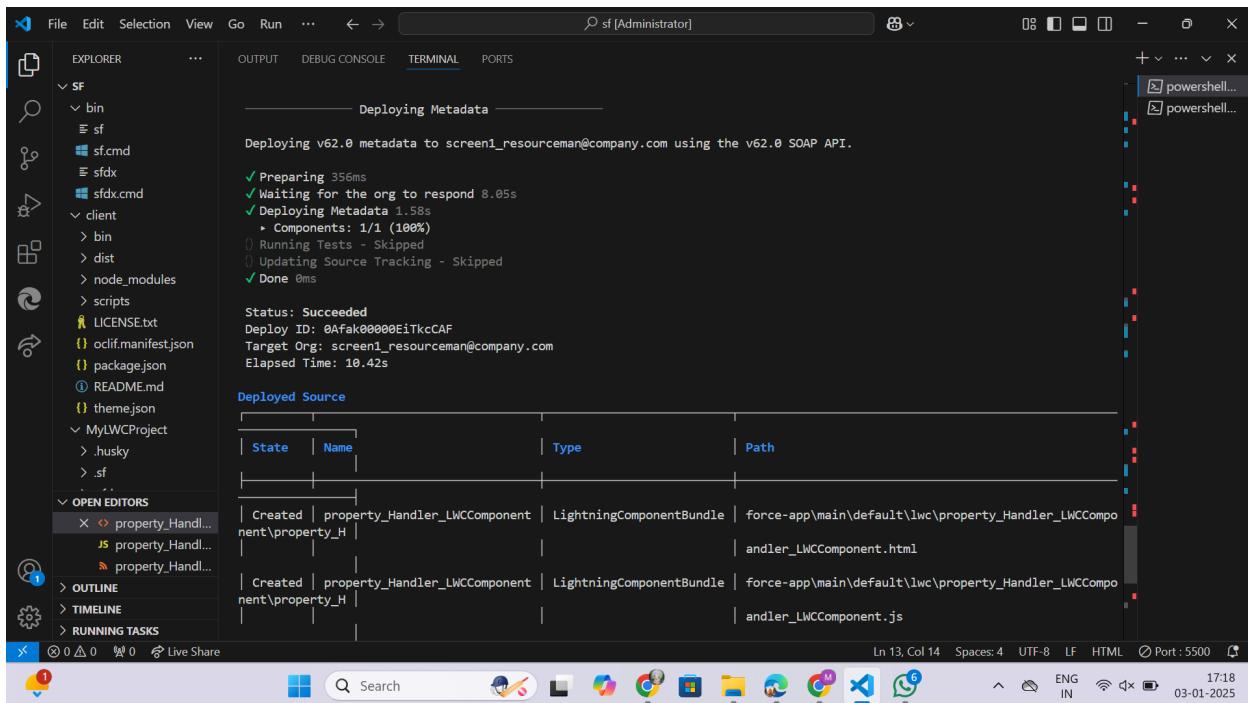
```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<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.



The screenshot shows the Salesforce CLI interface with the following details:

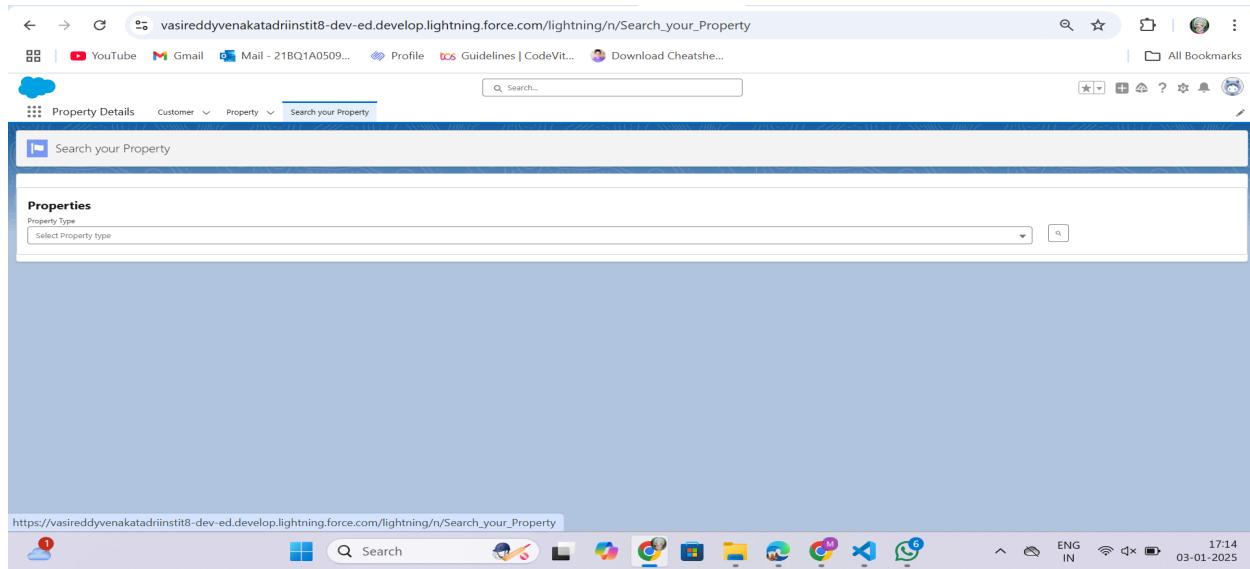
- EXPLORER** pane: Shows the project structure with files like sf, sf.cmd, sf.sfdx, sf.sfdx.cmd, client, bin, dist, node\_modules, scripts, LICENSE.txt, oclif.manifest.json, package.json, README.md, theme.json, and MyLWCProject.
- OUTPUT** pane: Displays the deployment logs:
  - Deploying v62.0 metadata to screen1\_resourceman@company.com using the v62.0 SOAP API.
  - Preparing: 356ms
  - Waiting for the org to respond 8.05s
  - Deploying Metadata 1.58s
    - Components: 1/1 (100%)
    - Running Tests - Skipped
    - Updating Source Tracking - Skipped
  - Done 0ms
 Status: Succeeded  
 Deploy ID: 0Afak00000EiTkcCAF  
 Target Org: screen1\_resourceman@company.com  
 Elapsed Time: 10.42s
- DEPLOYED SOURCE** pane: Shows a table of deployed components:
 

State	Name	Type	Path
Created	property_Handler_LWCComponent	LightningComponentBundle	force-app\main\default\lwc\property_Handler_LWCComponent\property_Handler_LWCComponent.html
Created	property_Handler_LWCComponent	LightningComponentBundle	force-app\main\default\lwc\property_Handler_LWCComponent\property_Handler_LWCComponent.js

**Drag this Component to your App Page**

### Adding the Component to your Page

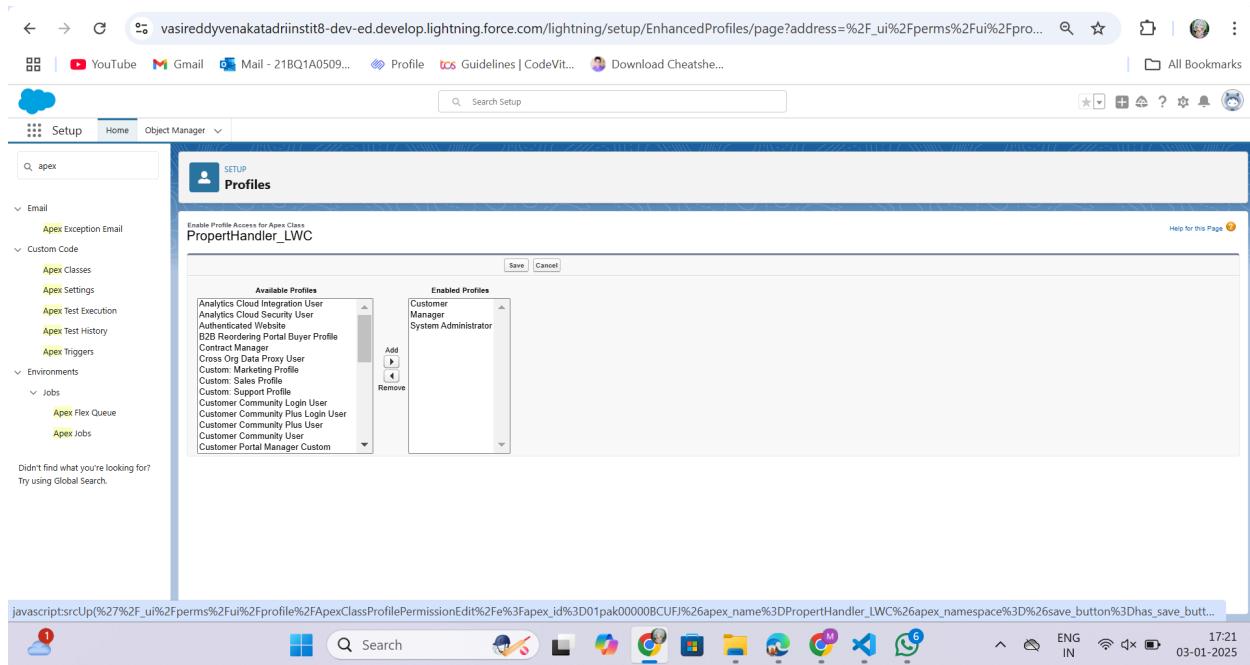
- From Setup >> Go to App Launcher >> Search for Property Details
- On this Page click on gear icon and click on Edit Page
- Drag the Component to your App Page and Save the Page.



## Give Access of Apex Classes to Profiles

The Apex Class has a Security, Enable the security for the profiles that needs to access this class.

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



# Project Overview

The **Dreams World Properties CRM Application** is a comprehensive solution designed to streamline customer and property management processes for a real estate business. The project leverages Salesforce as the core platform and integrates tools like Jotform to automate workflows and enhance operational efficiency.

The application includes hierarchical role management, dynamic property views based on user verification status, and an automated approval system for property verification. Through the use of Lightning Web Components (LWC) and Apex, it provides a tailored experience for verified and unverified customers while ensuring secure and centralized data management.

The CRM application includes the following key pages to deliver a seamless and functional user experience:

## **Pages of the Dreams World Properties CRM Application**

The CRM application includes the following key pages to deliver a seamless and functional user experience:

### **1. Home Page**

- **Purpose:** Acts as the central dashboard for users to view key insights and navigate to other modules of the application.
- **Features:**
  - Overview of recently added properties and customers.
  - Notifications for pending approvals and updates.
  - Quick links to access key features like property search, approvals, and customer management.

Dreams World

Name \*

First Name \_\_\_\_\_ Last Name \_\_\_\_\_

Email \_\_\_\_\_  
example@example.com

Phone Number \_\_\_\_\_  
0000 0000 0000

Please enter a valid phone number.

Which type of Property are you looking for?

Residential  
 Commercial  
 Serviced

Budget Amount \*

e.g. 22

Address

Street Address \_\_\_\_\_

Other Address Line 1 \_\_\_\_\_

City \_\_\_\_\_ State / Province \_\_\_\_\_

Postal / Zip Code \_\_\_\_\_

Submit

## 2. Customer Management Page

- **Purpose:** To manage customer records efficiently.
- **Features:**
  - View, edit, and delete customer details.
  - Add new customer records.
  - Status indicators for verified and unverified customers.
  - Search and filter functionalities for quick access.

vasireddyvenakatadriinst8-dev-ed.develop.lightning.force.com/lightning/o/Customer\_\_c/list?filterName=\_Recent

Property Details Customer Property Search your Property

Customer Recently Viewed

2 items • Updated a few seconds ago

	Customer	Customer	Phone Number	Email	State	Property Type	Budget Am...	Street Address	Street Address line 2	City	
1	Ambati	Rakesh	(788) 797-7864	rakeshaedf@gmail.com	Andhra Pradesh	Commercial	700,000	Ikkuru behind peerala savidi	Ikkuru behind peerala savidi	Palnadu	
2	Ambati	Madhavi	(899) 207-8836	ambatimadhavi213@gmail.com	Andhra Pradesh	Residential	50,000	Ikkuru behind peerala savidi	Ikkuru behind peerala savidi	Palnadu	

## 3. Property Management Page

- **Purpose:** Provides a comprehensive interface for managing property details.
- **Features:**

- List of all properties with key details like location, type, and price.
- Options to add, edit, or delete property records.
- Verification

status displayed for each property.

- Filters to sort properties by type, location, or verification status.

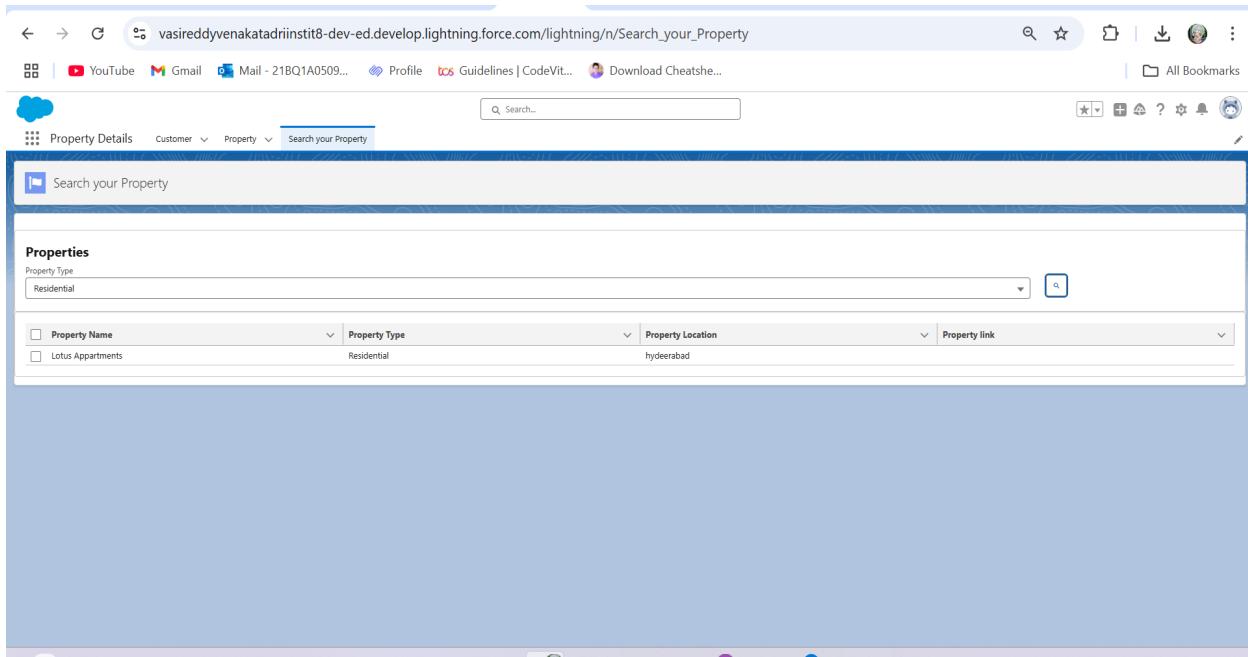
#### 4. Search Your Property Page

- **Purpose:**

Enables users to search and view properties dynamically based on their preferences and verification status.

- **Features:**

- Combobox for selecting property types (e.g., Residential, Commercial).
- Data table displaying properties tailored to the user's verification status.
- Conditional rendering:
  - Verified customers see all property details.
  - Unverified customers see limited property details.



## 5. Property Approval Page

- **Purpose:**

Streamlines the approval process for verifying property records.

- **Features:**

- Displays a list of properties pending approval.
- Step-by-step approval workflow for managers and executives.
- Status updates (Approved/Rejected) with automated field updates.

## 6. User Profile Management Page

- **Purpose:**

Allows users to view and manage their profile information.

- **Features:**

- Display of user roles and permissions.
- Option to update personal information.
- Status of user verification (verified or unverified).

## 7. Settings Page

- **Purpose:**

Enables administrators to configure the application settings.

- **Features:**

- Role and profile management.
- Customization options for property types, approval criteria, and notification preferences.
- Integration settings for external tools like Jotform.

## Custom Pages for Specific Roles

- **Sales Manager:**
  - A tailored dashboard highlighting pending approvals and property insights.
- **Sales Executive:**
  - Quick access to new leads and tasks assigned for follow-up.
- **Customer:**
  - Simplified interface to browse properties and track approval status.