

# **Project Title:A CRM Application To Handle The Clients And Their Property Related Requirements**

**College Name:** Arunai Engineering College

**College code:** 5104

**Department:** Computer Science And Engineering

## **Team Leader:**

**Name:** Rasiga Priya A G **Register No:** 510422104082

## **Team Members:**

**Name:** Monica G **Register No:** 510422104064

**Name:** Nivetha B **Register No:** 510422104071

**Name:** Salma Bee A **Register No:** 510422104088

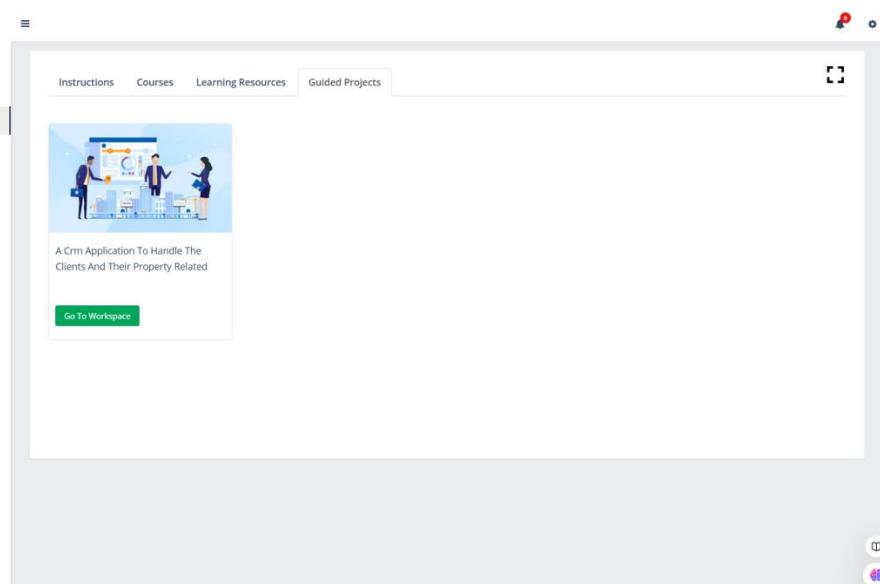
**GitHub link:** <https://github.com/Rasiga-priya/NM.git>

**Registered Email id:** rasigapriya771@gmail.com

# A CRM APPLICATION TO HANDLE THE CLIENTS AND THEIR PROPERTY RELATED REQUIREMENTS

## 1. PROJECT OVERVIEW

This project is focused on a CRM application to handle the clients and their property related requirements, designed to address the challenge of managing customer interactions efficiently and delivering personalized property recommendations based on user preferences. The goal is to deliver a comprehensive solution by leveraging Salesforce to automatically create and categorize customer records based on website engagement, capturing essential details and preferences. Users are categorized as either approved or non-approved, allowing approved users access to tailored property recommendations and broader listings. Through this project, we aim to enhance user experience and operational efficiency, providing personalized property recommendations and optimizing customer engagement and support the long-term goals of Dreams World Properties by facilitating growth in the real estate market and enhancing customer satisfaction.



## 2. OBJECTIVES

The specific, measurable goals the project intends to achieve.

### Business Goals:

- Streamline customer relationship management by automating record creation and categorization.
- Enhance customer engagement and satisfaction through tailored property recommendations.
- Support business growth in the real estate market by optimizing operational efficiency.

### Specific Outcomes:

- Automated creation and categorization of customer records in Salesforce based on website interactions.
- Delivery of personalized property selections to approved users, enhancing user experience.
- Improved data management, resulting in better insights into customer preferences and behaviours.

## 3. SALESFORCE KEY FEATURES AND CONCEPTS UTILIZED

The main functionalities and concepts applied within the Salesforce project:

1. **Automated Record Creation:** Integrated with JotForm to capture customer details from website interactions and automatically create records in Salesforce.
2. **User Categorization:** Salesforce categorizes users as approved or non-approved, enabling targeted property recommendations based on user status.
3. **Data Management and Segmentation:** Organizes and segments customer data efficiently within Salesforce, allowing easy access to user information for personalized interactions.
4. **Customized Recommendations:** Uses customer data to provide tailored property suggestions to approved users, enhancing engagement and relevance.

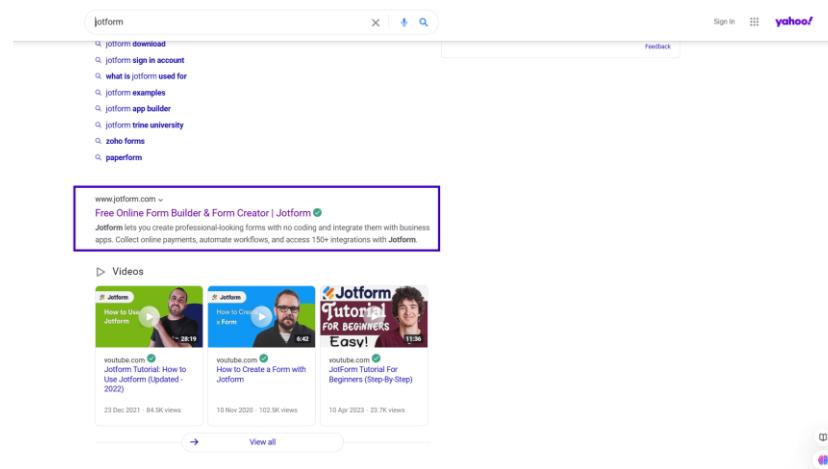
5. **Integration and Automation:** Seamlessly integrates JotForm with Salesforce, creating an automated workflow that reduces manual data entry and optimizes operations.
6. **Real-Time Customer Insights:** Delivers insights into customer preferences and behaviors, supporting strategic decision-making.
7. **Deployment via Visual Studio Code (VS Code):** The project was deployed using VS Code, enabling streamlined development, version control, and deployment processes.
8. **Apex Programming:** Utilized Apex to implement custom business logic, enhancing automation, data handling, and tailored functionalities within Salesforce.
9. **Lightning Web Components (LWC):** Built interactive and responsive UI components using LWC, improving user experience and enabling a more dynamic interface.

## 4. DETAILED STEPS TO SOLUTION DESIGN

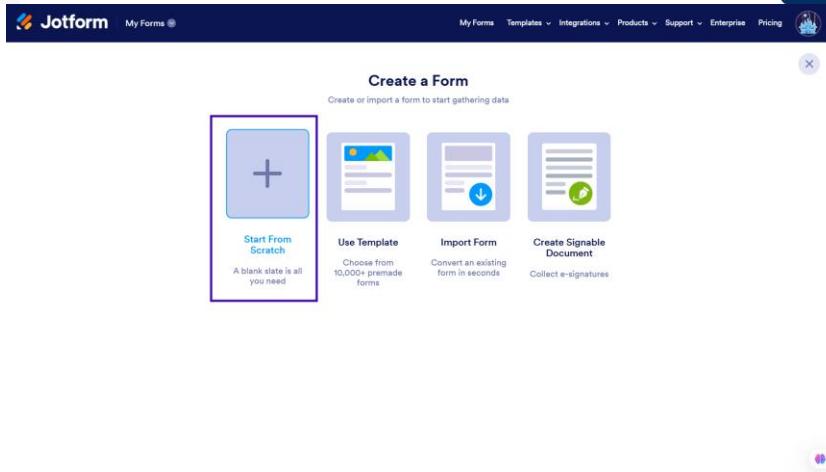
**Milestone 1 :- Create a Jotform and integrate it with the org to create a record of customers automatically.**

Activity 1:

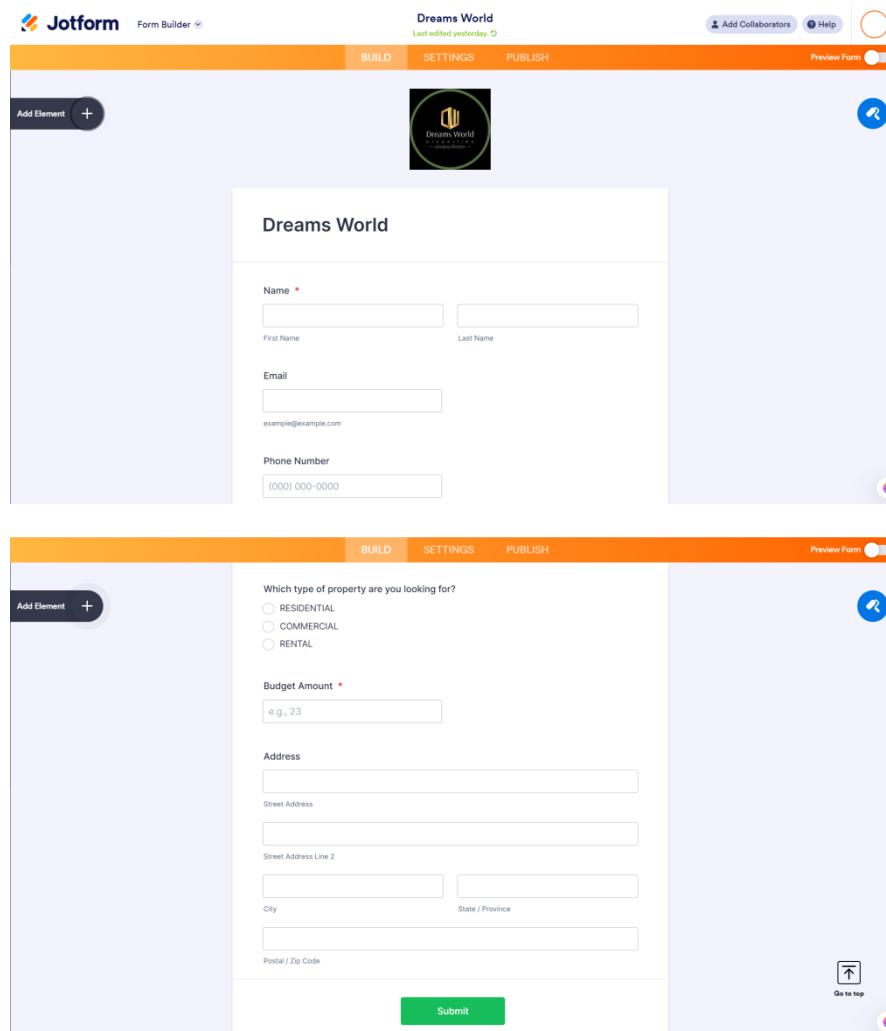
1. Open your browser, search for JotForm, go to the official website, and log in.



2. After login click on create form and click on start from scratch



- 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 the Jotform Form Builder interface with two forms displayed side-by-side:

- Top Form (Dreams World):** This form is for basic customer information. It includes fields for 'Name' (split into 'First Name' and 'Last Name'), 'Email' (with example@example.com), and 'Phone Number' (with placeholder (000) 000-0000). The 'BUILD' tab is selected at the top.
- Bottom Form (Dreams World):** This form is for property search criteria. It includes a section for 'Which type of property are you looking for?' with radio buttons for 'RESIDENTIAL', 'COMMERCIAL', and 'RENTAL'. Below that is a 'Budget Amount' field (placeholder e.g., 23). The 'Address' section contains fields for 'Street Address', 'Street Address Line 2', 'City', 'State / Province', and 'Postal / Zip Code'. A large green 'Submit' button is at the bottom. The 'PUBLISH' tab is selected at the top.

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

The screenshot shows the Jotform interface with a sidebar on the left containing links for Quick Share, Embed, Assign Form, Email, Prefill, PDF, and Platforms. The main area displays publishing options like Direct Link of Your Form, Share with Link (Public Form), Invite by Email, Share Form (with social sharing icons), and Create App. A preview form is visible on the right.

The jotform link: <https://form.jotform.com/243143317779058>

## Create Objects from Spreadsheet.

### Create Customer object

1. Go to your object manager and click on create object from spreadsheet.

The screenshot shows the Salesforce Object Manager page. It lists various objects with columns for Label, API Name, Type, Description, and Last Modified. A search bar at the top and a toolbar with various icons are also visible.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED
Account	Account	Standard Object		
Activity	Activity	Standard Object		
Address	Address	Standard Object		
Alternative Payment Method	AlternativePaymentMethod	Standard Object		
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object		
Appointment Category	AppointmentCategory	Standard Object		
Appointment Invitation	AppointmentInvitation	Standard Object		
Appointment Invitedee	AppointmentInvitedee	Standard Object		
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object		
Asset	Asset	Standard Object		
Asset Action	AssetAction	Standard Object		
Asset Action Source	AssetActionSource	Standard Object		
Asset Relationship	AssetRelationship	Standard Object		
Asset State Period	AssetStatePeriod	Standard Object		
Assigned Resource	AssignedResource	Standard Object		
Associated Location	AssociatedLocation	Standard Object		

2. Click on the link to get the spreadsheet

Customer

File Edit View Insert Format Data Tools Extensions Help

View only

A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1
1	Customer	Phone Number	Email	State	Property Type	Budget Amount	Street Address	City	Postal Code	Verified				
2	rahul	708797	rahul@gmail.com	Telangana	Residential	4000000	jp noida	saket ext 45	555001	checked				
3	prakash	55448855	p@gmail.com	Maharashtra	Commercial	8000000	gachibowli	indira road	600014	unchecked				
4	Prawal	454545	prawal@gmail.com	Maharashtra	Rental	25000	kemps	kothrud	444805	checked				
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														

Sheet1

### 3. Downloading of customer spreadsheet

### 4. After downloading, upload the file, map the fields and upload to create an object.

Create a custom object from a spreadsheet

Select a spreadsheet

Select a source for your new object data.

Upload .xlsx or .ods

or Drop File Here

Google Sheet

Office 365 or Drive

Setup Home Object Manager

Object Manager

customer

customer\_c

Customer

Customer

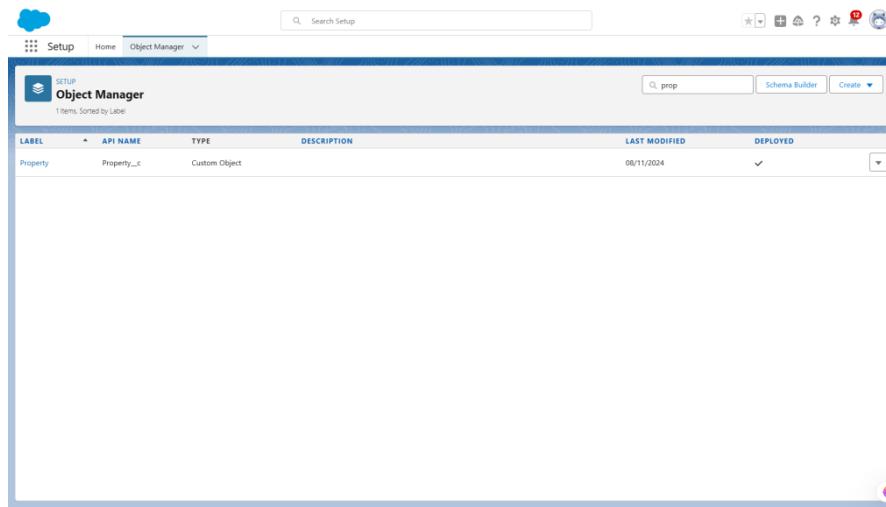
Search Setup

Last Modified

Deployed

## Create Property object

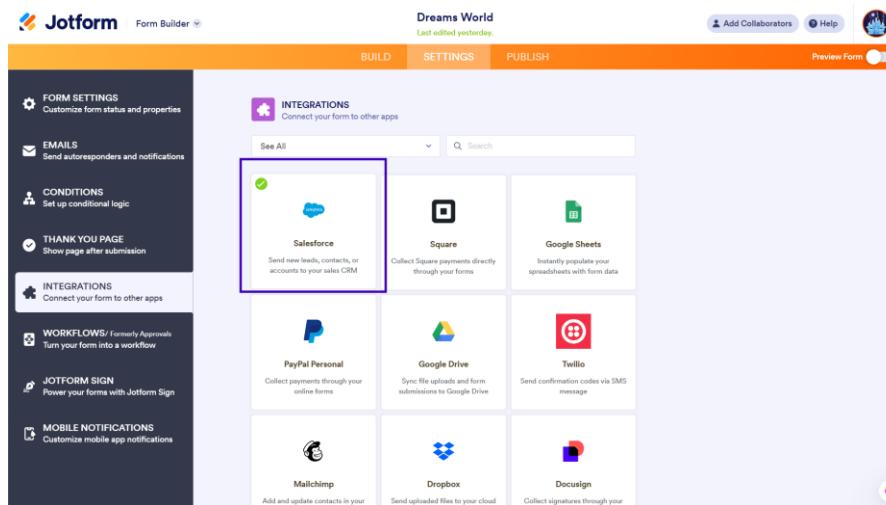
- Follow the same from the customer object to create the Property Object



## Integrate Jotform with Salesforce Platform

### Activity 1

- On the Jotform Platform, Click on Integration and choose Salesforce

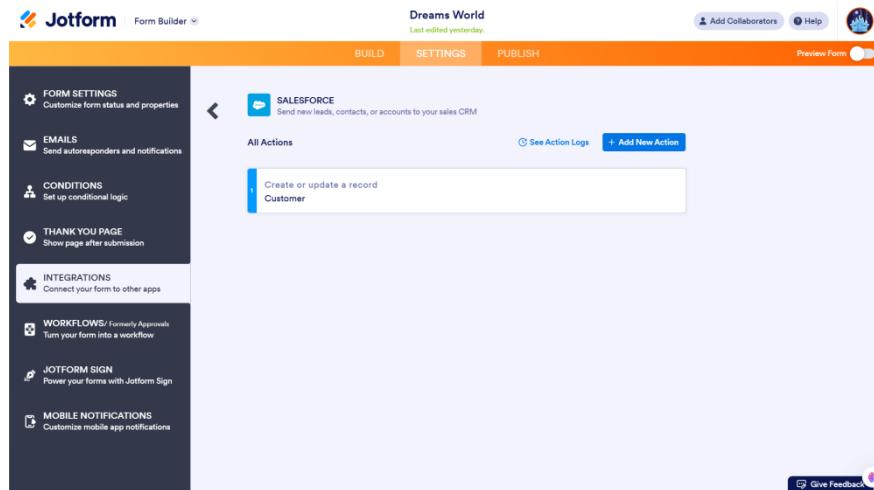


- Click on User Integration and choose “Add to From”

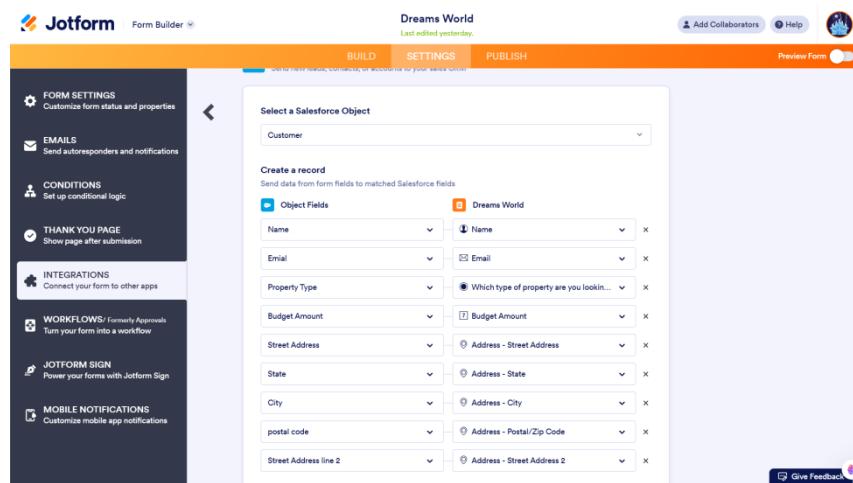
- Select the Org with which you want to Integrate your jotform with.

- Select an Action - Create a record.

## Select a Salesforce Object : - Customer



5. Map Each and every field on the Object with the fields on the form and “Save Action”
6. Then “Save the Integration” and “Finish”.



## Create Roles

### Sales Executive Role

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

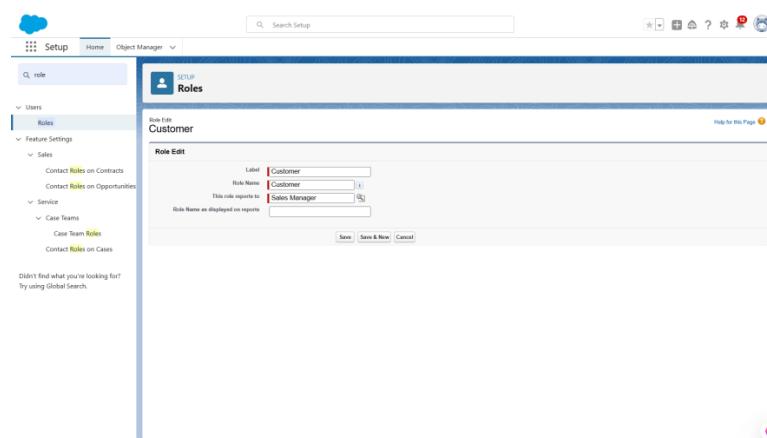
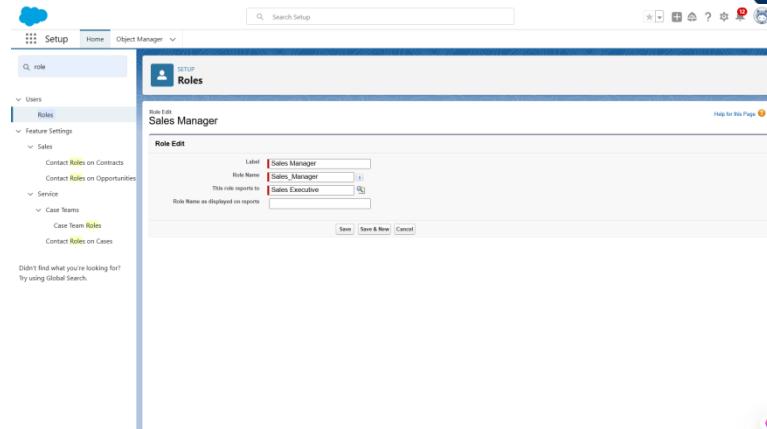


The screenshot shows the Salesforce Setup Roles page. On the left, a sidebar lists categories like Users, Feature Settings, Sales, Service, Case Teams, and Case Team Roles. The 'Roles' section is selected. The main area displays a tree view of the 'Your Organization's Role Hierarchy'. The hierarchy includes roles such as CEO, CFO, Sales Representative, Sales Executive, Sales Manager, Customer Support, International Sales, and VP Marketing. Each role has options to Edit, Delete, or Assign.

## 2. Label - Sales Executive Reports to - Sales Representative

The screenshot shows the 'Role Edit' screen for creating a new role named 'Sales Executive'. The 'Label' field is set to 'Sales Executive' and the 'Role Name' field is set to 'Sales\_Executive'. A dropdown menu under 'This role reports to' is set to 'Sales Representative'. At the bottom, there are 'Save', 'Save & New', and 'Cancel' buttons.

- 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.



## Create a Property Details App

### Activity 1

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.



Setup Home Object Manager

Q app

Salesforce Mobile App

App Manager

Connected Apps OAuth Usage

Manage Connected Apps

External Client Apps

Lightning Bolt

Flow Category

Lightning Bolt Solutions

Mobile Apps

Salesforce

Salesforce Branding

Mass Transfer Approval Requests

All Tabs

Analytics Studio

App Launcher

Automation

Batch Solutions

Business Rules Engine

Community

Content

Data Manager

Digital Experiences

Form

Lightning Usage App

Marketing CRM Classic

Platform

Property Details

Queue Management

Sales

Salesforce Branding

LightningSales

Search Setup

Lightning Experience App Manager

New Lightning App New Connected App

25 items • Sorted by App Name • Filtered by All appmenurelms - TabSet Type, App Type

App Name ↑	Developer Name	Description	Last Modified ...	App Type	Vl...
All Tabs	AllTabSet		07/11/2024, 7:03 pm	Classic	✓
Analytics Studio	Insights	Build CRM Analytics dashboards and apps	07/11/2024, 7:03 pm	Classic	✓
App Launcher	AppLauncher	App Launcher tabs	07/11/2024, 7:03 pm	Classic	✓
Automation	FlowsApp	Automate business processes and repetitive tasks.	07/11/2024, 7:08 pm	Lightning	✓
Batch Solutions	LightningBatch	Discover and manage business solutions designed for your industry.	07/11/2024, 7:03 pm	Lightning	✓
Business Rules Engine	ExpressionSetConsole	Create and maintain business rules that perform complex lookups and calcul...	07/11/2024, 7:08 pm	Lightning	✓
Community	Community	Salesforce CRM Communities	07/11/2024, 7:03 pm	Classic	✓
Content	Content	Salesforce CRM Content	07/11/2024, 7:03 pm	Classic	✓
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	07/11/2024, 7:03 pm	Lightning	✓
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	07/11/2024, 7:03 pm	Lightning	✓
Form	Form	Powerful online forms and workflow automation for Salesforce	10/11/2024, 9:26 am	Lightning (Managed)	✓
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	07/11/2024, 7:03 pm	Lightning	✓
Marketing CRM Classic	Marketing	Track sales and marketing efforts with CRM objects.	07/11/2024, 7:03 pm	Classic	✓
Platform	Platform	The fundamental Lightning Platform	07/11/2024, 7:03 pm	Classic	✓
Property Details	Property_Details		08/11/2024, 9:06 pm	Lightning	✓
Queue Management	QueueManagement	Create and manage queues for your business.	07/11/2024, 7:03 pm	Lightning	✓
Sales	Sales	The world's most popular sales force automation (SFA) solution	07/11/2024, 7:03 pm	Classic	✓
Salesforce Branding	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	07/11/2024, 7:03 pm	Lightning	✓

https://arauiaengineeringcollege-24s-dev-ed.lightning.force.com/lightning/setup/NavigationMenus/home

Lightning App Builder

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

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 Branding

\*App Name: Property Details

\*Developer Name: Property\_Details

Description: Enter a description...

Image:  Primary Color Hex Value: #007002

Upload

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

App Launcher Preview

PD Property Details

Lightning App Builder

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

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

Type to filter list...

Accounts

All Sites

Alternative Payment Methods

Analytics

App Launcher

Appointment Categories

Appointment Invitations

Approval Requests

Asset Action Sources

Asset Actions

Asset State Periods

Assets

Async Operation Logs

Authorization Form

Selected Items

Customer

Property

Search your Property

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

#### User Profiles

Choose the user profiles that can access this app.

##### Available Profiles

<input type="text"/> Type to filter list...
Analytics Cloud Integration User
Analytics Cloud Security User
Authenticated Website
Authenticated Website
B2B Reordering Portal Buyer Profile
Contract Manager
Cross Org Data Proxy User
Custom: Marketing Profile
Custom: Sales Profile
Custom: Support Profile
Customer Community Login User
Customer Community Plus Login User
Customer Community Plus User
Customer Community User

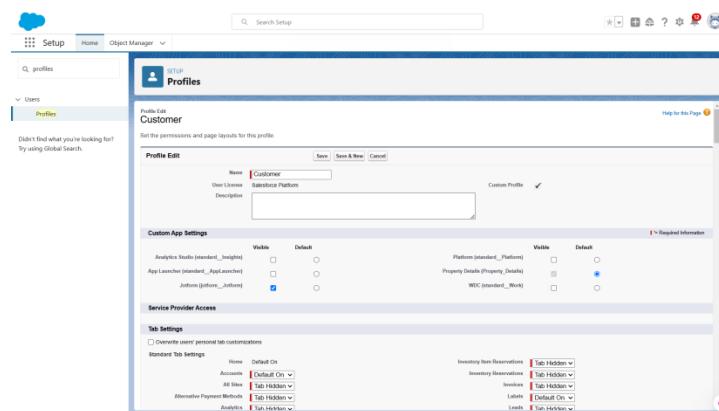
##### Selected Profiles

System Administrator
Customer
Manager

## Create Profiles

### Customer

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



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

Custom App Settings		Service Provider Access		Required Information		
		Visible	Default	Visible	Default	
Analytics Studio (standard_Insights)		<input type="checkbox"/>	<input checked="" type="radio"/>	Platform (standard_Platform)	<input type="checkbox"/>	<input checked="" type="radio"/>
App Launcher (standard_AppLauncher)		<input type="checkbox"/>	<input checked="" type="radio"/>	Property Details (Property_Details)	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
Jotform (jotform_Jotform)		<input checked="" type="checkbox"/>	<input type="radio"/>	WDC (standard_Work)	<input type="checkbox"/>	<input type="radio"/>

- Also Remove all the Standard Object Permissions.



The screenshot shows the 'Profiles' setup page. At the top, there is a search bar and a 'Standard Object Permissions' section with a note about controlling access at the object level. Below this is a large grid titled 'Basic Access' and 'Data Administration' for various standard objects like Accounts, Contacts, and Opportunities.

	Read	Create	Edit	Delete	View All	Modify All
Accounts	<input type="checkbox"/>					
Addresses	<input type="checkbox"/>					
Assets	<input type="checkbox"/>					
Authorization Forms	<input type="checkbox"/>					
Authorization Form Consents	<input type="checkbox"/>					
Authorization Form Data Uses	<input type="checkbox"/>					
Authorization Form Tests	<input type="checkbox"/>					
Background Operations	<input type="checkbox"/>					
Business Brands	<input type="checkbox"/>					
Communication Subscriptions	<input type="checkbox"/>					
Communication Subscription Channel Types	<input type="checkbox"/>					
Communication Subscription Consents	<input type="checkbox"/>					
Communication Subscription Timings	<input type="checkbox"/>					
Contacts	<input type="checkbox"/>					
Contact Point Addresses	<input type="checkbox"/>					
Contact Point Consents	<input type="checkbox"/>					
Contact Point Emails	<input type="checkbox"/>					

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

The screenshot shows the 'Custom Object Permissions' setup page. It displays two profiles: 'Customer' and 'JotformUsers'. For each profile, it lists the permissions for the 'Property' object. The 'Customer' profile has 'Read' and 'View All' checked, while 'JotformUsers' has 'Create', 'Edit', and 'Modify All' checked.

	Read	Create	Edit	Delete	View All	Modify All
Customer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
JotformUsers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" 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 checked="" type="checkbox"/>

## Create a Check Box field on user

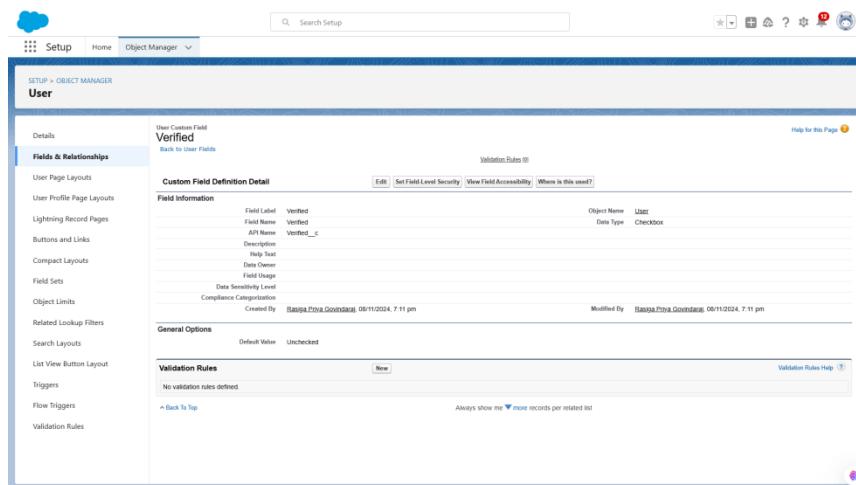
### Activity 1

- Setup > Object Manager > Search for User > Fields and Relationships

The screenshot shows the 'Fields & Relationships' section of the 'User' object manager. It lists various fields like 'About Me', 'Active', 'Address', etc., along with their field labels, field names, data types, controlling fields, and indexing status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
About Me	AboutMe	Text Area<10000		
Active	IsActive	Checkbox		
Address	Address	Address		
Admin Info Emails	ReceivesAdminInfoEmails	Checkbox		
Alias	Alias	Text(8)		
Allow Forecasting	ForecastEnabled	Checkbox		
Banner Photo	BannerPhotoId	Lookup(Photo)		
Call Center	CallCenterId	Lookup(Call Center)		
Chatter Email Highlights Frequency	DigestFrequency	Picklist		
Company Name	CompanyName	Text(80)		
Contact	ContactId	Lookup(Contact)		
Data.com Monthly Addition Limit	JigaveImportLimitOverride	Number(9, 0)		
Default Notification Frequency when Joining Groups	DefaultGroupNotificationFrequency	Picklist		
Delegated Approver	DelegatedApproverId	Lookup(User/Group)		
Department	Department	Text(80)		
Phone	Phone	Text(40)		

- Create new Field Named as “Verified” as Data type “Check Box”

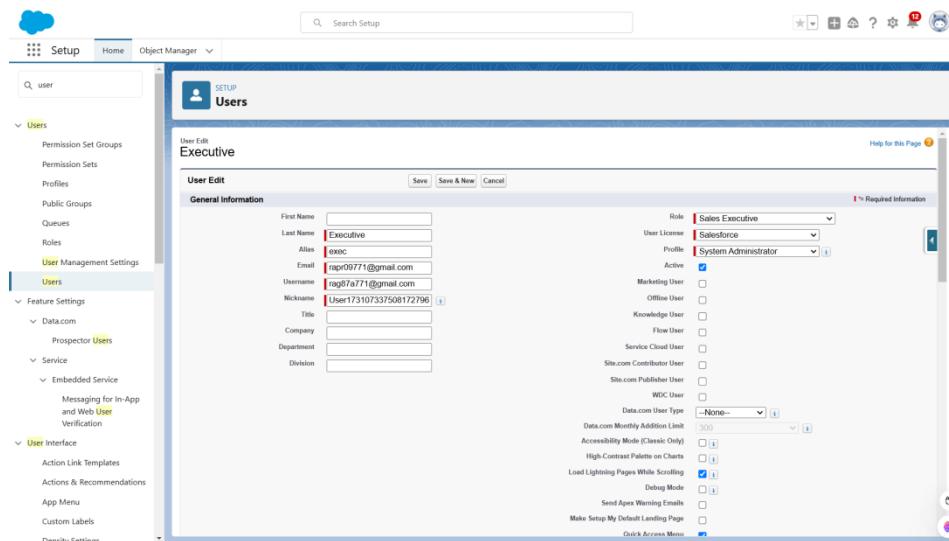


The screenshot shows the Salesforce Setup interface for the 'User' object. In the 'Field & Relationships' section, a new custom field named 'Verified' has been created. The field is of type 'checkbox' and is labeled 'Verified'. It has a field-level security rule applied.

## Create Users

### 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

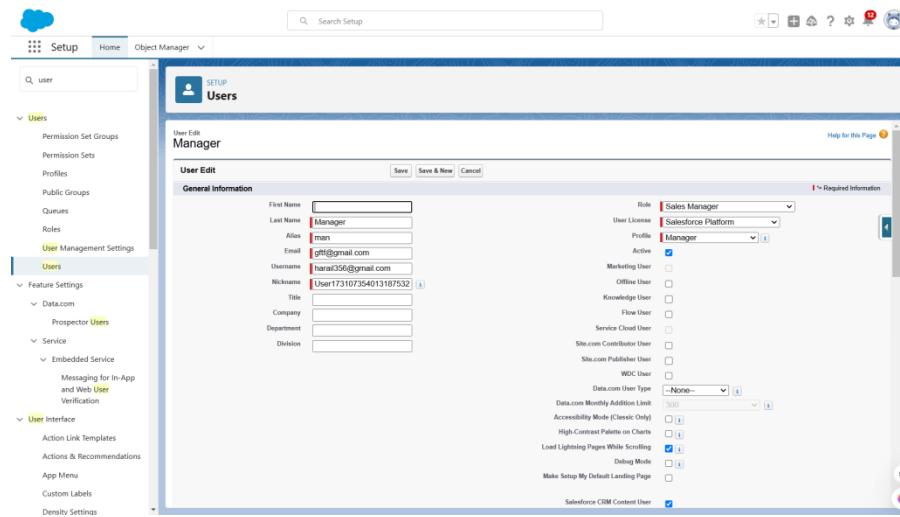


The screenshot shows the 'User Edit' page for a user named 'Executive'. The user's first name is 'Executive', last name is 'Executive', and email is 'rapi@rpiya.com'. The role is set to 'Sales Executive', license to 'Salesforce', and profile to 'System Administrator'. The 'Active' checkbox is checked.

### User 2

1. Go to Setup >> Administration >> Users >> New User
2. Last Name >> Manager
3. Role >> Sales Manager

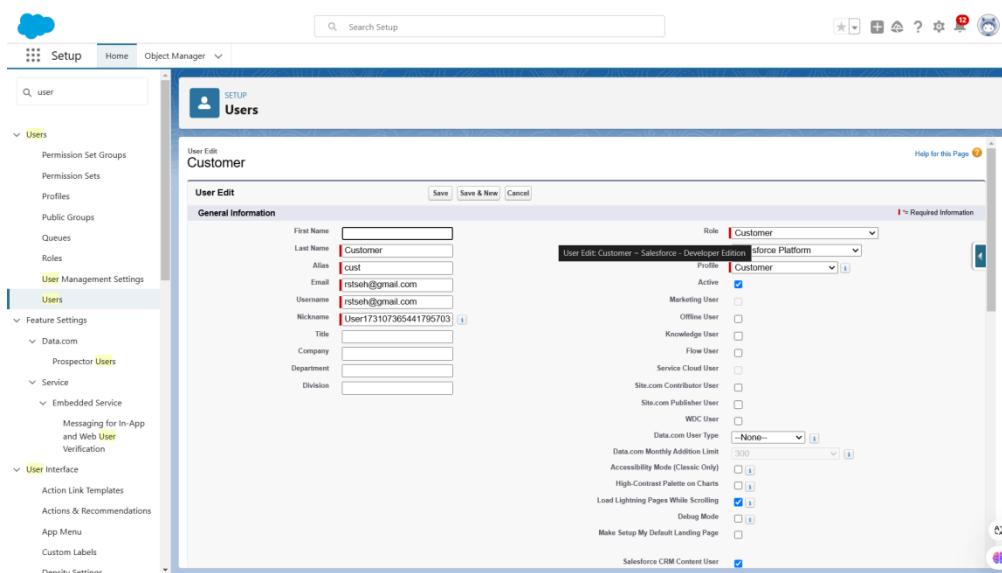
4. License >> Salesforce Platform
5. Profile >> Manager
6. Save



The screenshot shows the 'User Edit' screen for a user named 'Manager'. The 'Role' dropdown is set to 'Sales Manager'. The 'User License' dropdown is set to 'Salesforce Platform'. The 'Profile' dropdown is set to 'Manager'. Other fields like First Name, Last Name, Alias, Email, Username, Nickname, Title, Company, Department, and Division are populated with sample data. The 'Active' checkbox is checked. The 'Data.com Monthly Addition Limit' dropdown is set to 300. The 'Accessibility Mode (Classic Only)' checkbox is checked. The 'Load Lightning Pages While Scrolling' checkbox is checked. The 'Debug Mode' checkbox is checked. The 'Make Setup My Default Landing Page' checkbox is checked.

### User 3

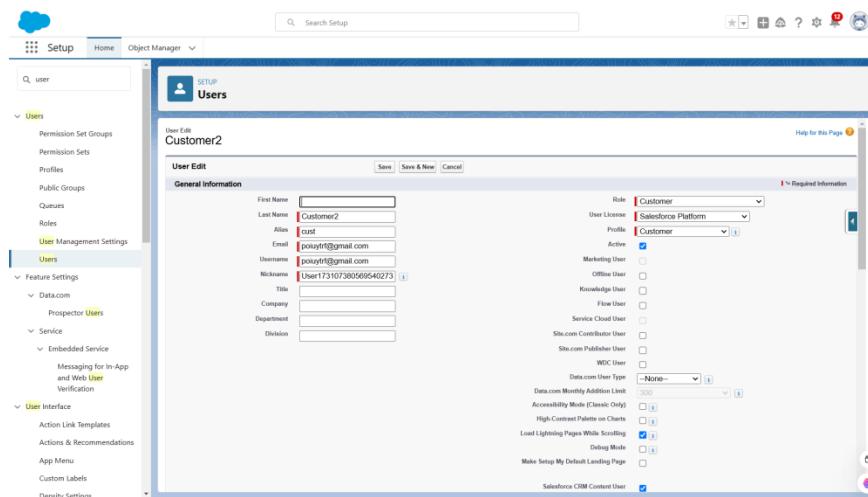
1. Go to Setup >> Administration >> Users >> New User
2. Last Name >> Customer
3. Role >> Customer
4. License >> Salesforce Platform
5. Profile >> Customer
6. Make Sure the verified check box is “Unchecked”
7. Save



The screenshot shows the 'User Edit' screen for a user named 'Customer'. The 'Role' dropdown is set to 'Customer'. The 'User License' dropdown is set to 'Salesforce Platform'. The 'Profile' dropdown is set to 'Customer'. Other fields like First Name, Last Name, Alias, Email, Username, Nickname, Title, Company, Department, and Division are populated with sample data. The 'Active' checkbox is checked. The 'Data.com Monthly Addition Limit' dropdown is set to 300. The 'Accessibility Mode (Classic Only)' checkbox is checked. The 'Load Lightning Pages While Scrolling' checkbox is checked. The 'Debug Mode' checkbox is checked. The 'Make Setup My Default Landing Page' checkbox is checked.

### User 4

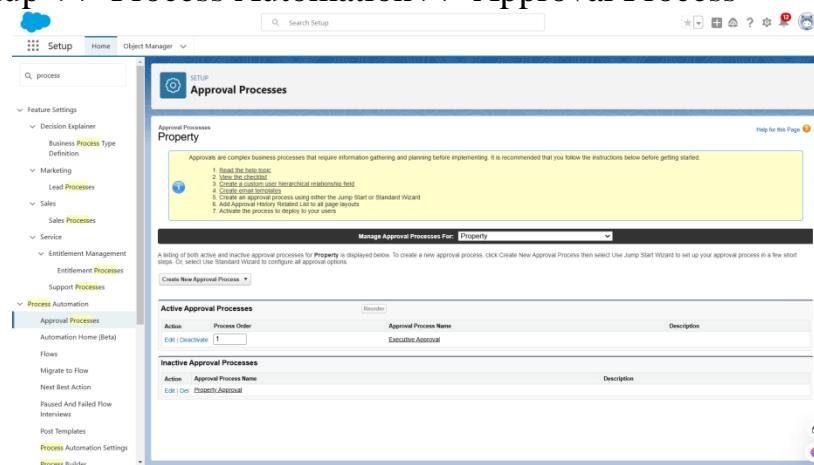
1. Go to Setup >> Administration >> Users >> New User
2. Last Name >> Customer2
3. Role >> Customer
4. License >> Salesforce Platform
5. Profile >> Customer
6. Make Sure the verified check box is “checked”
7. Save



## Create an Approval Process for Property Object

### Activity 1

1. From Setup >> Process Automation >> Approval Process



2. Process Name - Property Approval



Setup Home Object Manager

Feature Settings

- Decision Explorer
- Business Process Type Definition
- Marketing
- Sales
- Service
- Entitlement Management
- Entitlement Processes
- Support Processes
- Process Automation
  - Approval Processes
  - Automation Home (Beta)
  - Flows
  - Migrate to Flow
  - Next Best Action
  - Paused And Failed Flow Interviews
  - Post Templates
  - Process Automation Settings
  - Process Builder
  - Workflow Actions
  - Email Alerts

Approval Process Edit  
Property Approval

Step 1. Enter Name and Description

Enter a name and description for your new approval process.

Enter Name and Description

Process Name: Property Approval  
Unique Name: Property\_Approval

Description:

Save Next Cancel

### 3. Give 2 criteria -Location is not equal to blank, Verified Equals false

Setup Home Object Manager

Feature Settings

- Decision Explorer
- Business Process Type Definition
- Marketing
- Sales
- Service
- Entitlement Management
- Entitlement Processes
- Support Processes
- Process Automation
  - Approval Processes
  - Automation Home (Beta)
  - Flows
  - Migrate to Flow
  - Next Best Action
  - Paused And Failed Flow Interviews
  - Post Templates
  - Process Automation Settings
  - Process Builder
  - Workflow Actions
  - Email Alerts

Approval Process Edit  
Property Approval

Step 2. Specify Entry Criteria

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
Property_Verified	equals	false
-None-	-None-	-None-
-None-	-None-	-None-
-None-	-None-	-None-

AND  
AND  
AND  
AND

Add Filter Logic... Previous Save Next Cancel

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

Setup Home Object Manager

Feature Settings

- Decision Explorer
- Business Process Type Definition
- Marketing
- Sales
- Service
- Entitlement Management
- Entitlement Processes
- Support Processes
- Process Automation
  - Approval Processes
  - Automation Home (Beta)
  - Flows
  - Migrate to Flow
  - Next Best Action
  - Paused And Failed Flow Interviews
  - Post Templates
  - Process Automation Settings
  - Process Builder
  - Workflow Actions
  - Email Alerts

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. 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.

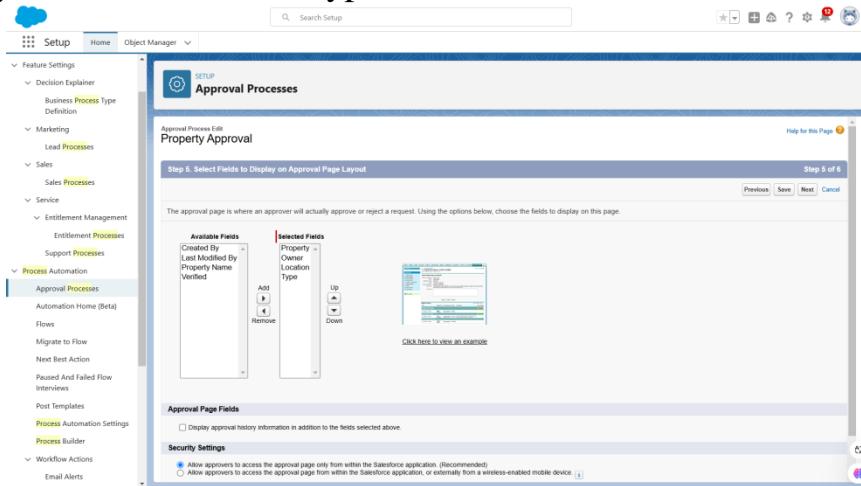
Previous Save Next Cancel

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

### 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.

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

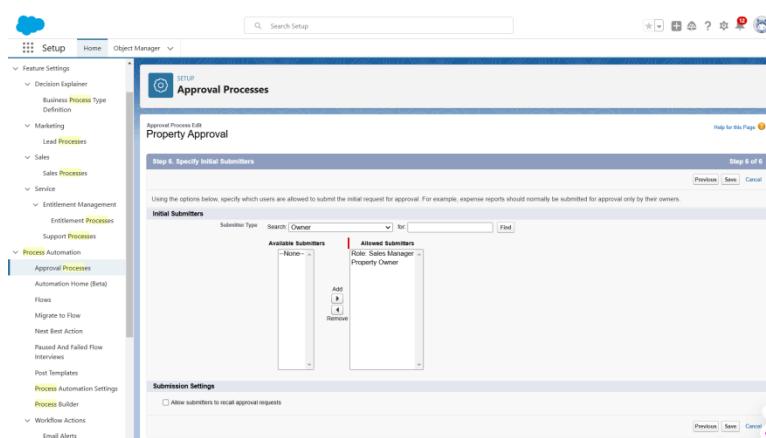


The screenshot shows the Salesforce Setup interface for creating an Approval Process. The left sidebar navigation includes 'Feature Settings', 'Decision Explorer', 'Business Process Type Definition', 'Marketing', 'Lead Processes', 'Sales', 'Sales Processor', 'Service', 'Entitlement Management', 'Entitlement Processes', 'Support Processes', 'Process Automation', 'Approval Processes', 'Automation Home (Beta)', 'Flows', 'Migrate to Flow', 'Next Best Action', 'Paused And Failed Flow', 'Interviews', 'Post Templates', 'Process Automation Settings', 'Process Builder', and 'Workflow Actions'. The 'Email Alerts' option is under 'Process Automation'. The main window title is 'Approval Processes' and the sub-section is 'Property Approval'. The step being worked on is 'Step 6. Select Fields to Display on Approval Page Layout'. The interface shows a list of 'Available Fields' on the left and 'Selected Fields' on the right. The 'Selected Fields' list contains 'Property', 'Owner', 'Location', and 'Type'. There are 'Add' and 'Remove' buttons between the two lists. To the right of the lists is a preview of the approval page layout. Below the preview are sections for 'Approval Page Fields' (checkbox for 'Display approval history information in addition to the fields selected above') and 'Security Settings' (radio buttons for 'Allow approvers to access the approval page only from within the Salesforce application. (Recommended)' and 'Allow approvers to access the approval page from within the Salesforce application, or externally from a wireless-enabled mobile device').

7. Click Next and Select the initial Submitters >>

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

8. Save.



The screenshot shows the 'Approval Processes' setup page continuing from the previous step. The main window title is 'Approval Processes' and the sub-section is 'Property Approval'. The step is 'Step 6. Specify Initial Submitters'. The interface shows a list of 'Available Submitters' on the left and 'Selected Submitters' on the right. The 'Selected Submitters' list contains 'Role: Sales Manager - Property Owner'. There are 'Add' and 'Remove' buttons between the two lists. At the top of the list area is a search bar with placeholder text 'Search / Owner' and a 'Find' button. Below the lists is a section titled 'Submission Settings' with a checkbox for 'Allow submitters to recall approval requests'. The left sidebar navigation is identical to the previous screenshot.

9. Add an approval step name “Executive Approval ”

The screenshot shows the 'SETUP Approval Processes' screen. On the left, a sidebar lists various setup categories like Feature Settings, Marketing, Sales, Service, and Process Automation. Under Process Automation, 'Approval Processes' is selected. The main area shows 'Step 1. Enter Name and Description' for a process named 'Executive Approval'. Fields include 'Process Name' (Executive Approval), 'Unique Name' (Executive\_Approval), and 'Description'. Buttons at the bottom right include 'Save', 'Next', and 'Cancel'.

10. specify the Criteria >> All record should enter

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

The screenshot shows 'Step 5. Select Fields to Display on Approval Page Layout'. It displays two columns: 'Available Fields' (Created By, Last Modified By, Location, Property Name, Type) and 'Selected Fields' (Property - Owner Verified). A 'Click here to view an example' link is present. Below this, there's an 'Approval Page Fields' section with a checkbox for 'Display approval history information in addition to the fields selected above' and a 'Security Settings' section with two options: 'Allow approvers to access the approval page only from within the Salesforce application. (Recommended)' and 'Allow approvers to access the approval page from within the Salesforce application, or externally from a wireless-enabled mobile device'.

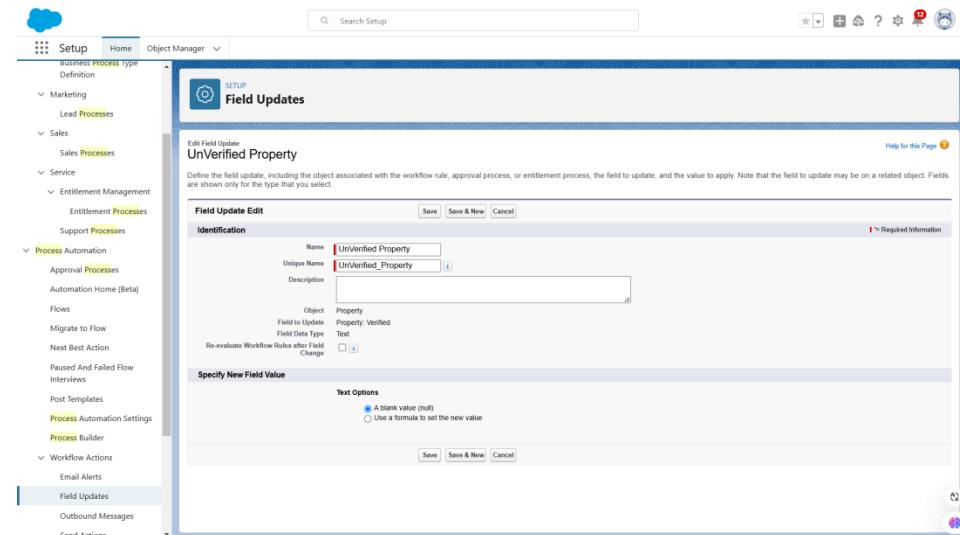
12. Add One field Update as “Verified Property”

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

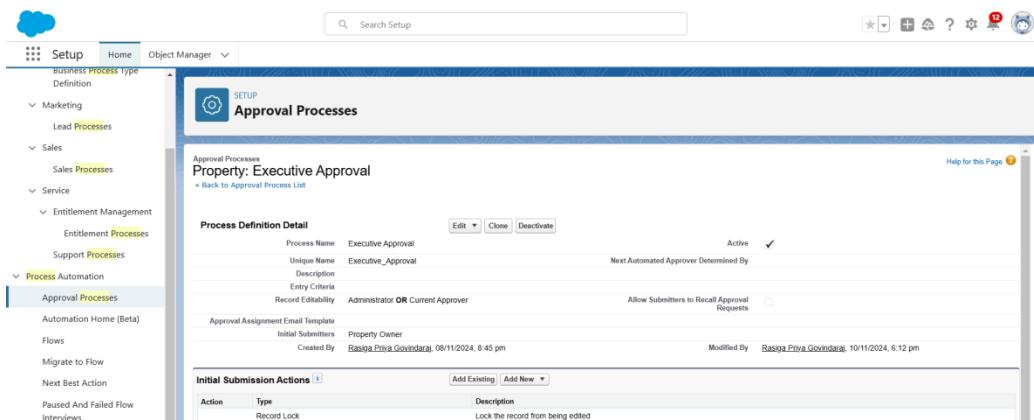
The screenshot shows the 'SETUP Field Updates' screen. The sidebar includes 'Feature Settings', 'Decision Explorer', 'Business Process Type Definition', 'Marketing', 'Sales', 'Service', 'Entitlement Management', 'Support Processes', and 'Process Automation' (selected). The main area shows 'Field Update Edit' for 'Verified Property'. Identification fields include 'Name' (Verified Property), 'Unique Name' (Verified\_Property), and 'Object' (Property). Under 'Specify New Field Value', there are 'Text Options' with radio buttons for 'A blank value (null)' and 'Use a formula to set the new value'. Buttons at the bottom right include 'Save', 'Save & New', and 'Cancel'.

### 13. Add One field Update as “UnVerified Property”

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



### 14. Activate the Approval Process.



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

#### Activity 1

- From Setup >> Search for Flows >> Click On New and Select “Record Trigger Flow”.



Select Type

Recommended

- Screen Flow
- Schedule-Triggered Flow
- Autolaunched Flow (No Trigger)
- Record-Triggered Flow
- Platform Event—Triggered Flow
- Record-Triggered Orchestration

All Flow Types

- Autolaunched Flow (No Trigger)
- Autolaunched Orchestration (No Trigger)
- Contact Request Flow
- Evaluation Flow

Get more on the AppExchange

Create

2. Select Object >> Property

3. Select “Trigger the flow when” >> “A record is created”

Flow Builder

Start Record-triggered Flow

Object: Property

Trigger: A record is created

Optimize for: Actions and Related Records

+ Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Property

End

Configure Start

Select Object

Object: Property

Trigger the Flow When:

- A record is created (selected)
- A record is updated
- A record is created or updated
- A record is deleted

Set Entry Conditions

Condition Requirements: None

4. Set Entry Conditions >> “None”

5. Add a “Action” >> “Submit for Approval”

Flow Builder

**Start**  
Record-Triggered Flow  
Object: **Property** Edit  
Trigger: **A record is created**  
Optimize for: **Actions and Related Records**  
+ Add Scheduled Paths (Optional)  
Open Flow Trigger Explorer for Property

**End**

**Configure Start**  
A record is updated  
A record is created or updated  
A record is deleted

**Set Entry Conditions**  
Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed.  
Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

**Condition Requirements**  
None

**\* Optimize the Flow for:**

- Fast Field Updates**  
Update fields on the record that triggers the flow to run. This high-performance flow runs *before* the record is saved to the database.
- Actions and Related Records**  
Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.

Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed

6. Give Label >> Approval for property

7. Record Id >> { !\$Record.Id }

Flow Builder

**Start**  
Record-Triggered Flow  
Object: **Property** Edit  
Trigger: **A record is created**  
Run Immediately  
+  
**Action**  
Approval for property  
End

**Submit for Approval**

**Label**: Approval for property

**API Name**: Approval\_for\_property

**Description**

**Submit for Approval** submit-submit

Use values from earlier in the flow to set the inputs for the "Submit for Approval" core action. To use its outputs later in the flow, store them in variables.

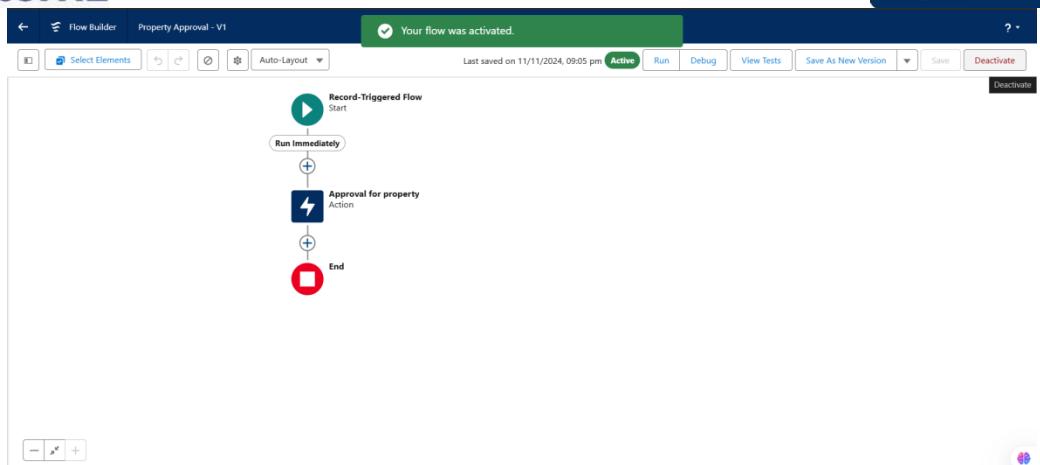
Set Input Values for the Selected Action

**A<sub>1</sub>** Record ID: A<sub>1</sub> Triggering Property...c > Record ID

**A<sub>2</sub>** Approval Process Name Or ID: Not Included

8. Done

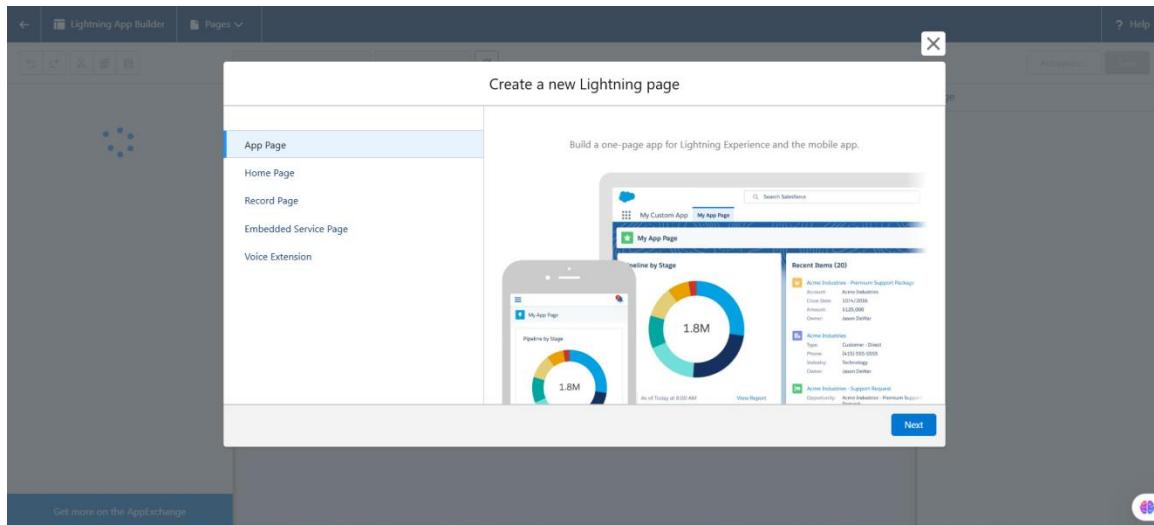
9. Save the Flow and Give label as “Property Approval” and “Activate”



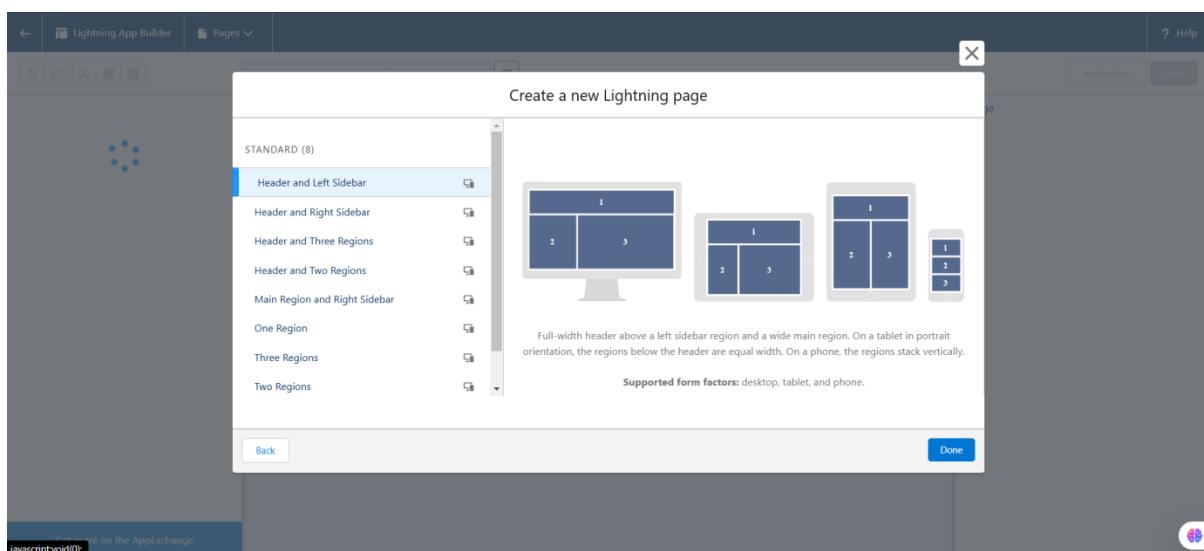
## Create an App Page

### Activity 1

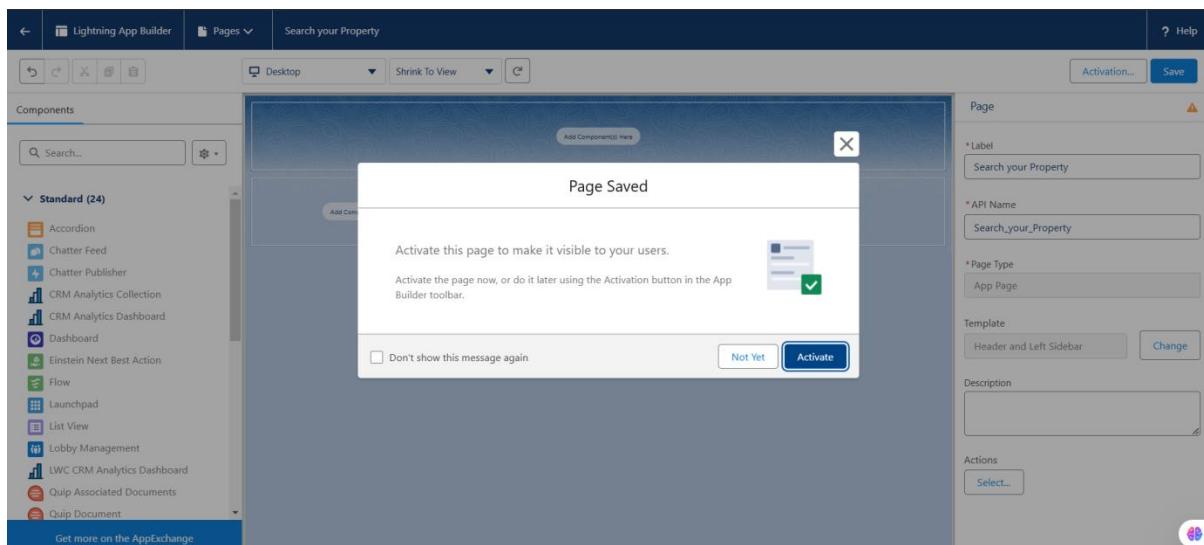
- From Setup >> Go to Lightning App Builder >> Click on New >> Select App Page and Click on Next.



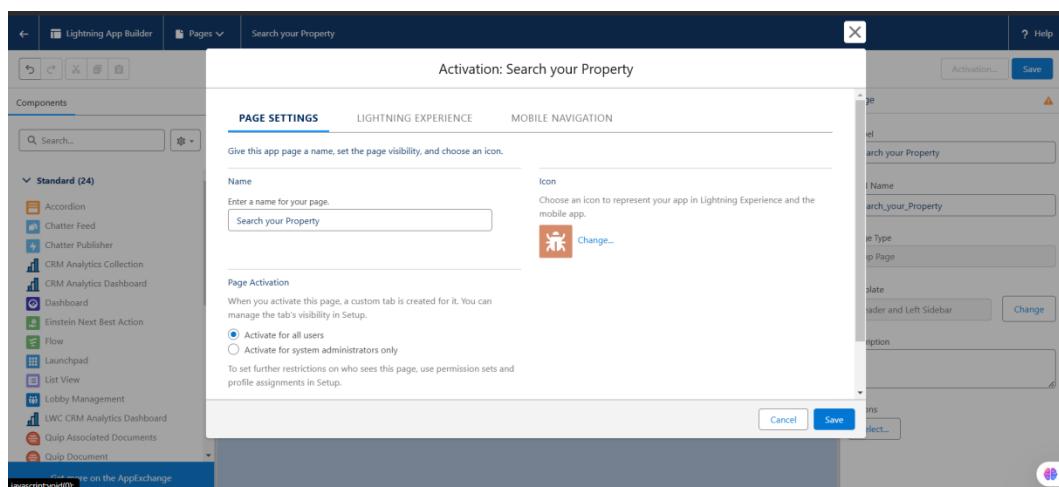
- Give Label as “Search your Property” click “Next”.
- Click “header and Left Sidebar” and Click on “Done”



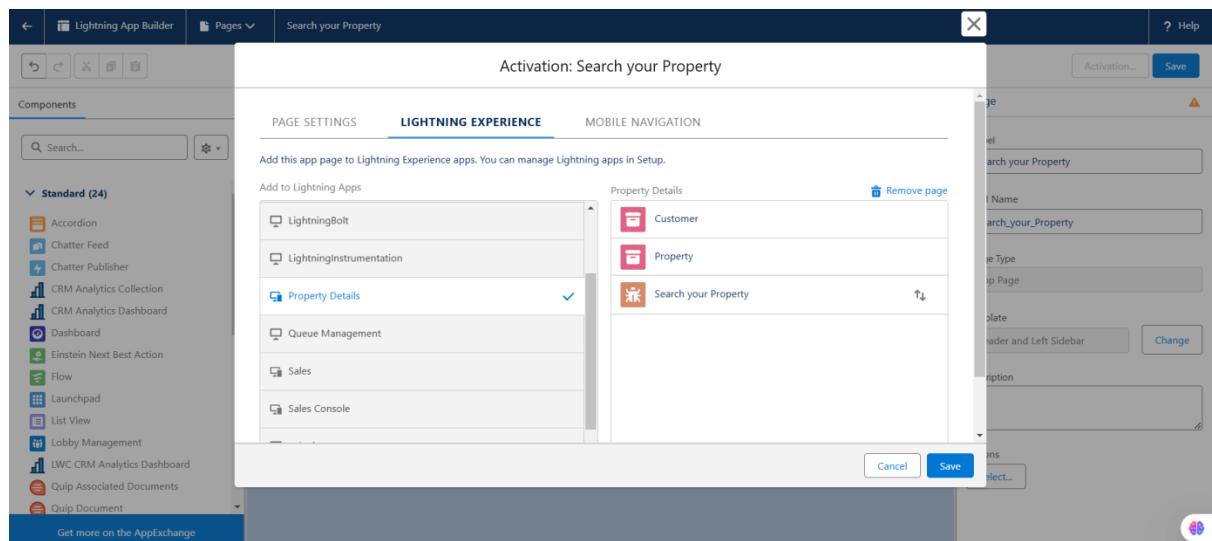
#### 4. Click on “Save ” and then click on “Activate”.



#### 5. From Page Setting select page activation as “Activate for all Users”.

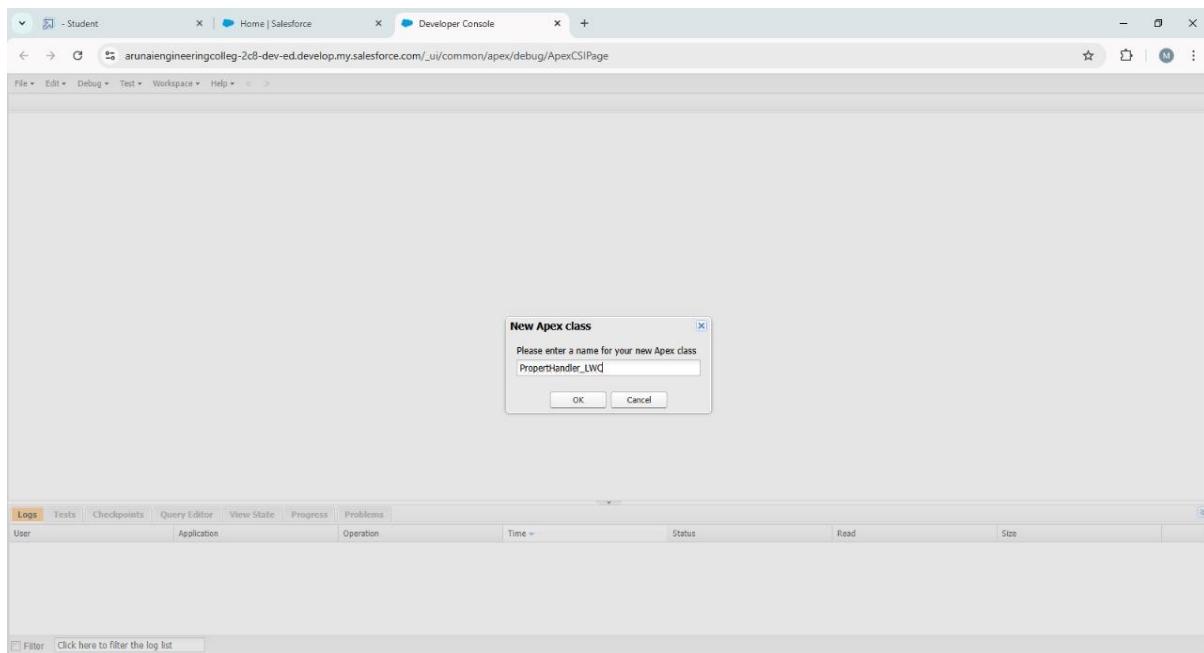


6. From Lightning Experience Click on “Property Details” and click on Add Page“.
7. Then Click on “Save”



## Create a LWC Component

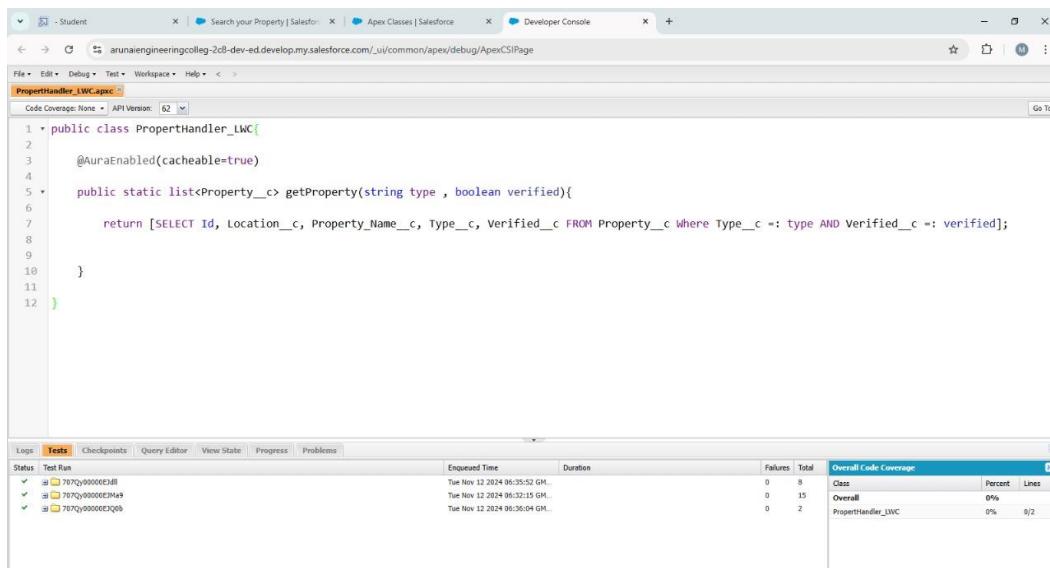
1. Create an Apex Class and make it aura enabled and name it “PropertHandler\_LWC”



2. After creating the apex class paste the code.

CODE:

```
public class PropertyHandler_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];
    }
}
```



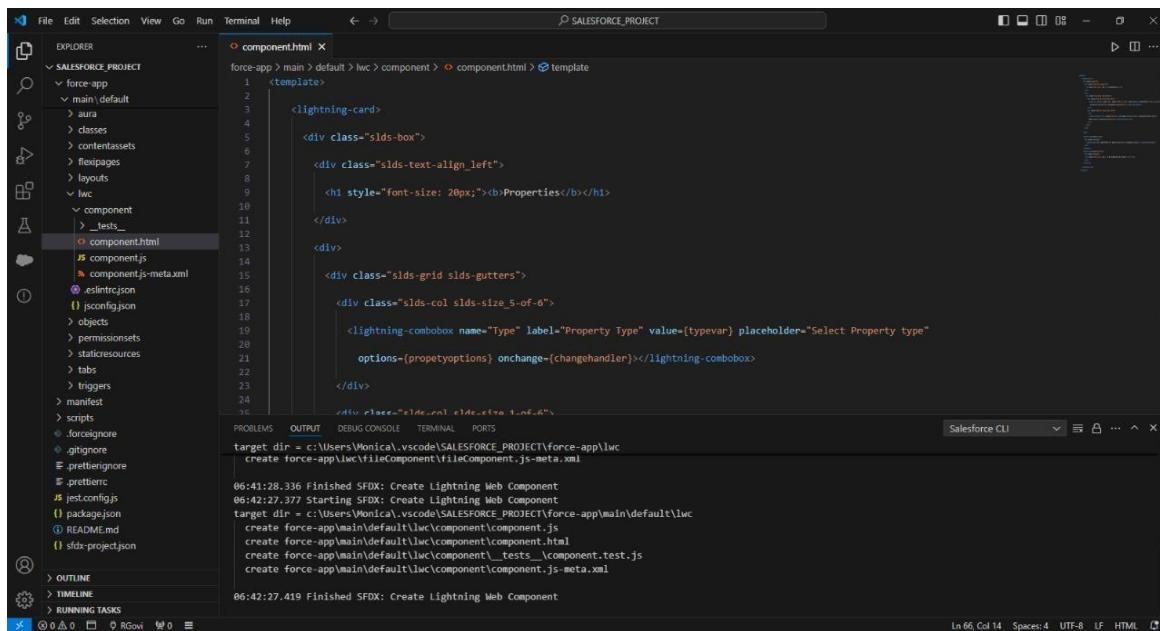
2. Create a Lightning Web Component in your VsCode, and (ctrl+shift +P) and click on authorize an org.
3. Enter your login id and password to authorize your org.
4. Now (ctrl+shift +P) and Create a lightning Web Component and Name it Anything you want to. (Example - )
5. In your Html File Write this 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>
</div>

<template if:true={istru}>
  <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>
```

```
</lightning-card>
</template>
```



The screenshot shows the VS Code interface with a Salesforce LWC project open. The Explorer sidebar shows the project structure: force-app > main > default > lwc > component > component.html > template. The component.html file is selected and displayed in the main editor area. The code in component.html is:

```

<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>
        </div>
      </div>
    </div>
  </lightning-card>
</template>

```

The terminal tab shows the command-line output of the SFDX command to create the component:

```

target dir = c:\Users\Monica\vscode\SALESFORCE_PROJECT\force-app\lwc
create force-app\lwc\fileComponent\fileComponent.js-meta.xml
06:41:28.336 Finished SFDX: Create Lightning Web Component
06:42:27.377 Starting SFDX: Create Lightning Web Component
target dir = c:\Users\Monica\vscode\SALESFORCE_PROJECT\force-app\main\default\lwc
create force-app\main\default\lwc\component\component.js
create force-app\main\default\lwc\component\component.html
create force-app\main\default\lwc\component\_tests\component.test.js
create force-app\main\default\lwc\component\component.js-meta.xml
06:42:27.419 Finished SFDX: Create Lightning Web Component

```

## 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" }
  ]
  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)
})
}
}
  
```



```
force-app > main > components > JS component > component.js ...  
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  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1
```

```

component.html
component.js
component.js-meta.xml

force-app > main > default > lwc > component > component.html > component.js > component.js-meta.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
4    <apiVersion>59.0</apiVersion>
5
6    <isExposed>true</isExposed>
7
8    <targets>
9      <target>lightning__RecordPage</target>
10     <target>lightning__AppPage</target>
11     <target>lightning__HomePage</target>
12   </targets>
13
14 </LightningComponentBundle>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
target dir = c:\Users\Monica\.vscode\SALESFORCE_PROJECT\force-app\lwc
create force-app\lwc\fileComponent\fileComponent.js-meta.xml

06:41:28.336 Finished SFDX: Create Lightning Web Component
06:42:27.377 Starting SFDX: Create Lightning Web Component
target dir = c:\Users\Monica\.vscode\SALESFORCE_PROJECT\force-app\main\default\lwc
create force-app\main\default\lwc\component\component.js
create force-app\main\default\lwc\component\component.html
create force-app\main\default\lwc\component\_tests\component.test.js
create force-app\main\default\lwc\component\component.js-meta.xml

06:42:27.419 Finished SFDX: Create Lightning Web Component

```

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

```

component.html
component.js
component.js-meta.xml

force-app > main > default > lwc > component > component.html > component.js > component.js-meta.xml
1  <template>
2
3    <lightning-card>
4
5      <div class="slds-box">
6        <div class="slds-text-align_left">
7          <h1 style="font-size: 20px;"><b>Properties</b></h1>
8        </div>
9
10       <div class="slds-grid slds-gutters">
11         <div class="slds-col slds-size_5-of-6">
12           <lightning-combobox name="Type" label="Property Type" value={typevar} placeholder="Select Property type"
13             options={propetyoptions} onchange={changehandler}></lightning-combobox>
14         </div>
15       </div>
16
17       <button class="slds-button slds-button--neutral slds-size_1-of-6">
18         <lightning-icon icon="utility:refresh" /> Refresh
19       </button>
20     </div>
21
22   </lightning-card>
23
24 </div>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
target dir = c:\Users\Monica\.vscode\SALESFORCE_PROJECT\force-app\main\default\lwc
06:42:27.419 Finished SFDX: Create Lightning Web Component
06:44:35.49 Starting SFDX: Deploy This Source to Org

--- Deployed Source ---
STATE  FULL NAME      TYPE      PROJECT PATH
Created component  LightningComponentBundle  force-app\main\default\lwc\component\component.html
Created component  LightningComponentBundle  force-app\main\default\lwc\component\component.js
Created component  LightningComponentBundle  force-app\main\default\lwc\component\component.js-meta.xml

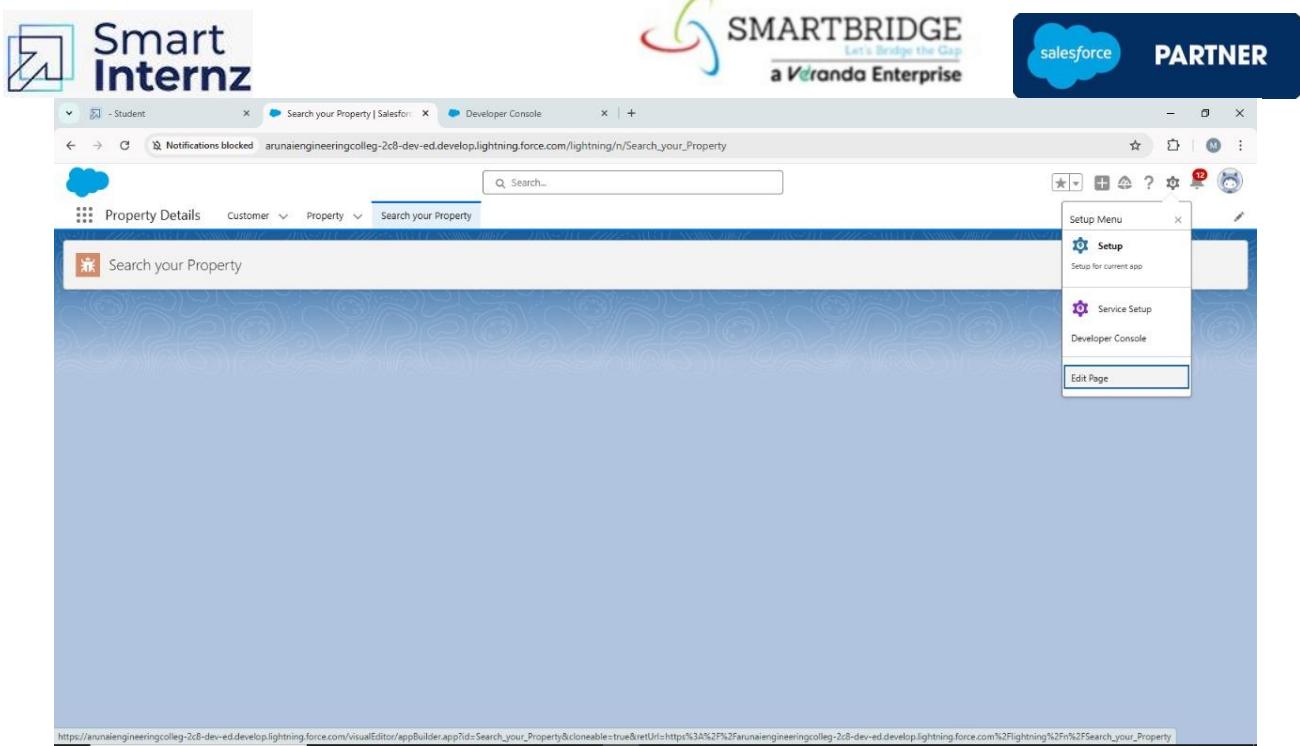
06:44:37.786 ended SFDX: Deploy This Source to Org

```

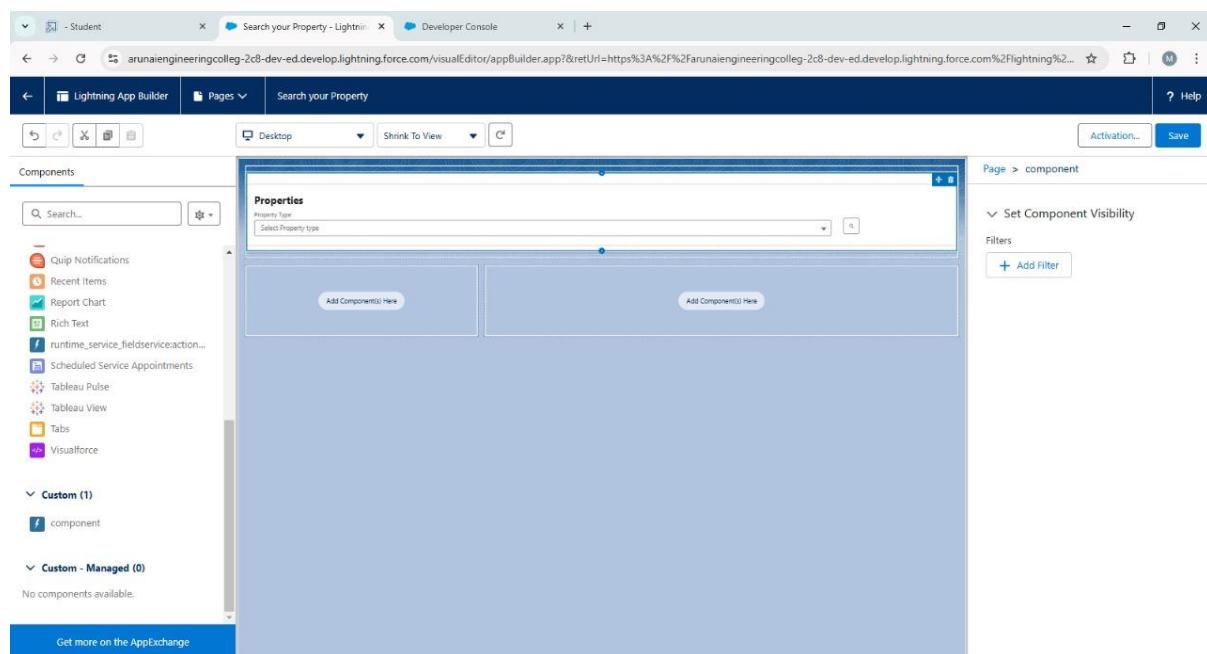
**Drag this Component to your App Page**

### Activity 1

- From Setup >> Go to App Launcher >> Search for Property Details
- On this Page click on gear icon and click on Edit Page



### 3. Drag the Component to your App Page and Save the Page.



## Give Access of Apex Classes to Profiles

### Activity 1

- From Setup >> Search For Apex Classes >> Click on “Security” behind “PropertyHandler\_\_LWC”.

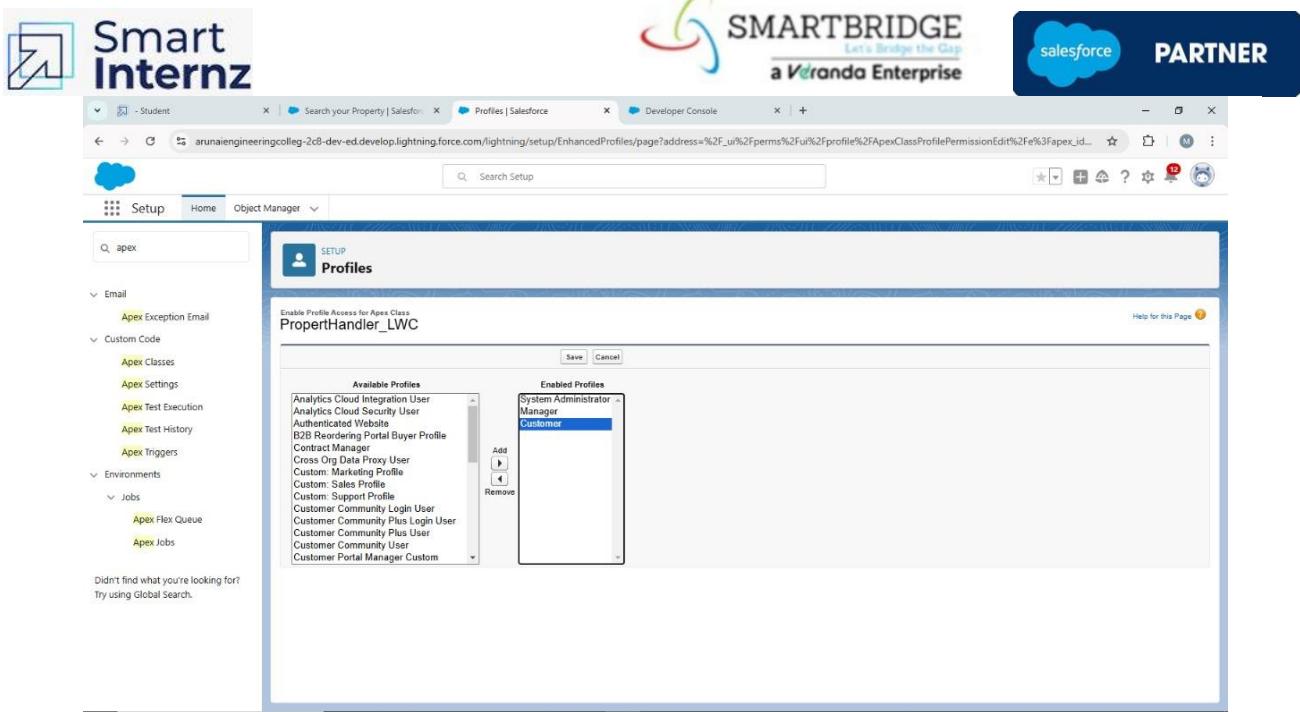
Action	Name	Namespace Prefix	API Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit : Security	CurrentUserInfo	jotform	60.0	Active	886	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit : Security	JotFormSetupController	jotform	60.0	Active	6,532	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit : Security	JotFormSetupException	jotform	60.0	Active	57	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit : Security	JotFormUserInfo	jotform	60.0	Active	801	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit	MockHttpResponse	jotform	60.0	Active	629	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit : Del : Security	PropertyHandler__LWC	jotform	62.0	Active	303	Rasika Priya Govindaraj	12/11/2024, 6:35 am
Edit : Security	ProxyRequest	jotform	60.0	Active	496	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit : Security	RequestUtil	jotform	60.0	Active	1,898	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit	TestCurrentUserInfo	jotform	60.0	Active	2,056	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit	TestJotFormSetupController	jotform	60.0	Active	7,645	Rasika Priya Govindaraj	11/11/2024, 8:17 pm
Edit	TestRequestUtil	jotform	60.0	Active	2,895	Rasika Priya Govindaraj	11/11/2024, 8:17 pm

Show me fewer ▲ records per list page

**Dynamic Apex Classes**  
Dynamic Apex extends your programming reach by interacting with Lightning Platform components.

View:	All	Create New View		
Class Name	Namespace Prefix	API Version	Created By	Last Modified By
No records to display.				

- From Profiles Add “Manager” and “Customer” and “Save”



## 5. TESTING AND VALIDATION

### 1. Unit testing(apex classes, Triggers)

Unit testing in Salesforce with Aura involves verifying the functionality of individual Lightning components and their JavaScript controllers, helpers, and handlers. Using the **Lightning Testing Service (LTS)** and **Jest**, developers can create test cases to simulate user actions, test component logic, and verify data flows between the client and server-side controllers. Testing in Aura also helps catch errors early, especially when components interact with Apex or other Lightning components.

```

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 WHERE Type__c =: type AND Verified__c =: verified];
8
9     }
10
11 }
12

```

User	Operation	Time	Status	Read	Size
Rasiga Priya Govindaraj	Apex Test Handler	11/13/2024, 12:22:18 AM	Success	Unread	18.96 KB
Rasiga Priya Govindaraj	Apex Test Handler	11/13/2024, 12:22:18 AM	Success	Unread	2.63 KB
Rasiga Priya Govindaraj	Apex Test Handler	11/13/2024, 12:22:18 AM	Success	Unread	6.63 KB





File • Edit • Debug • Test • Workspace • Help • < >

Code Coverage: None • API Version: 62 • Go To

```
1 * public class PropertHandler_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 Where Type__c =: type AND Verified__c =: verified];
8
9
10    }
11
12 }
```

**Logs** Tests Checkpoints Query Editor View State Progress Problems

User	Application	Operation	Time	Status	Read	Size
Rasga Priya Govindaraj	Unknown	ApexTestHandler	11/13/2024, 12:22:38 AM	Success	Unread	2.66 KB
Rasga Priya Govindaraj	Unknown	ApexTestHandler	11/13/2024, 12:22:37 AM	Success	Unread	18.78 KB
Rasga Priya Govindaraj	Unknown	ApexTestHandler	11/13/2024, 12:22:37 AM	Success	Unread	6.46 KB
Rasga Priya Govindaraj	Unknown	ApexTestHandler	11/13/2024, 12:22:18 AM	Success	Unread	6.63 KB
Rasga Priya Govindaraj	Unknown	ApexTestHandler	11/13/2024, 12:22:18 AM	Success	Unread	18.96 KB
Rasga Priya Govindaraj	Unknown	ApexTestHandler	11/13/2024, 12:22:18 AM	Success	Unread	

Filter Click here to filter the log list

## 2. User Interface Testing

UI testing for a JotForm integrated with Salesforce CRM ensures the form is visually consistent across devices, validates input fields for correctness, and verifies that submitted data maps accurately to Salesforce fields. It also checks error handling for invalid inputs and tests end-to-end functionality to confirm smooth data flow from the form to Salesforce, ensuring proper synchronization and triggering of workflows. This process ensures a user-friendly experience and reliable data integration.

Here are the steps for testing the process where a JotForm is submitted, data is visible in the Property Detail tab in Salesforce, and also available on the JotForm website:

### 1. Form Submission:

- A user submits the JotForm, filling in the relevant information.
- Ensure that all form fields (such as property details) are correctly filled out and validated before submission.


**Dreams World**

Name \*

First Name

Last Name

Email

example@example.com

Phone Number

Please enter a valid phone number.

Which type of Property are you looking for?

- RESIDENTIAL  
 COMMERCIAL  
 RENTAL

Budget Amount \*

Address

Street Address

Street Address Line 2

<input type="text" value="Minnesota"/>	<input type="text" value="Saint Paul"/>
--	---

City

State / Province

Postal / Zip Code

## Data Sync to Salesforce:

- Verify that upon submission, the data is transferred from JotForm to Salesforce.
- Check that the information appears in the correct Property Detail tab within Salesforce, ensuring that each field maps to the appropriate Salesforce CRM fields.
- Confirm that any workflows, such as notifications or updates, are triggered based on the form submission



PARTNER

Property Details

Customer Property Search your Property

Search...

Recently Viewed

2 items • Updated a few seconds ago

	Customer	Custo...	Phone Number	Email	State	Property Type	Budget...	Street Address	Street Addres...	City	
1	Charlotte Jones	Jones	(123) 456-7891	charlottee340jones@gmail.com	Saint Paul	RESIDENTIAL	45,897	1234 Elm st	Apt 56B	Minnesota	
2	james walter	walter	(458) 754-1254	jameswalter23@gmail.com	texas	COMMERCIAL	45,623	834, white building	golden street	Austin	

New Change Owner Import Assign Label

Search this list... Filter














































































































































































































































































































## 6. CONCLUSION

In conclusion, our project for Dreams World Properties successfully integrates Salesforce to enhance and streamline customer interactions. By linking website engagement with Salesforce's CRM capabilities, customer details and preferences are automatically captured, creating a comprehensive record of each client. This integration categorizes users into approved and non-approved groups, allowing approved users to receive tailored property recommendations, enhancing their experience with relevant and expanded listings.

The seamless data flow between systems optimizes operations, improves customer engagement, and provides a more efficient and personalized user journey. This solution ultimately positions Dreams World Properties for growth, making them more competitive and responsive in the real estate market.