

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

1. Project Overview:

The project aims to design and develop a CRM (Customer Relationship Management) application using Salesforce to manage clients and their property-related information. The application will enable the organization to streamline client interactions, property management, and related services.

2. Objectives:

- 1. Client Management:** Design a client management system to store and manage client information, including contact details, property ownership, and interaction history.
- 2. Property Management:** Develop a property management system to store and manage property-related information, including property type, location, and ownership details.
- 3. Service Management:** Create a service management system to manage various services offered to clients, such as property maintenance, repairs, and renovations.
- 4. Reporting and Analytics:** Develop reports and analytics to provide insights into client interactions, property management, and service delivery.
- 5. User Adoption:** Ensure user adoption by providing an intuitive and user-friendly interface.

3. Salesforce Key Features and Concepts Utilized:

- 1. Accounts and Contacts:** Utilize Accounts and Contacts to manage client information and relationships.
- 2. Custom Objects:** Create custom objects to manage property-related information, such as Property, Property Type, and Property Ownership.
- 3. Relationships:** Establish relationships between Accounts, Contacts, and custom objects to manage complex data relationships.
- 4. Page Layouts and Views:** Design page layouts and views to provide an intuitive and user-friendly interface for users.
- 5. Workflows and Approvals:** Implement workflows and approvals to automate business processes and ensure data accuracy.

6. Reports and Dashboards: Develop reports and dashboards to provide insights into client interactions, property management, and service delivery.

7. Security and Access: Implement security and access controls to ensure data integrity and confidentiality.

8. Integration: Integrate with external systems, such as property management software, to provide a seamless user experience.

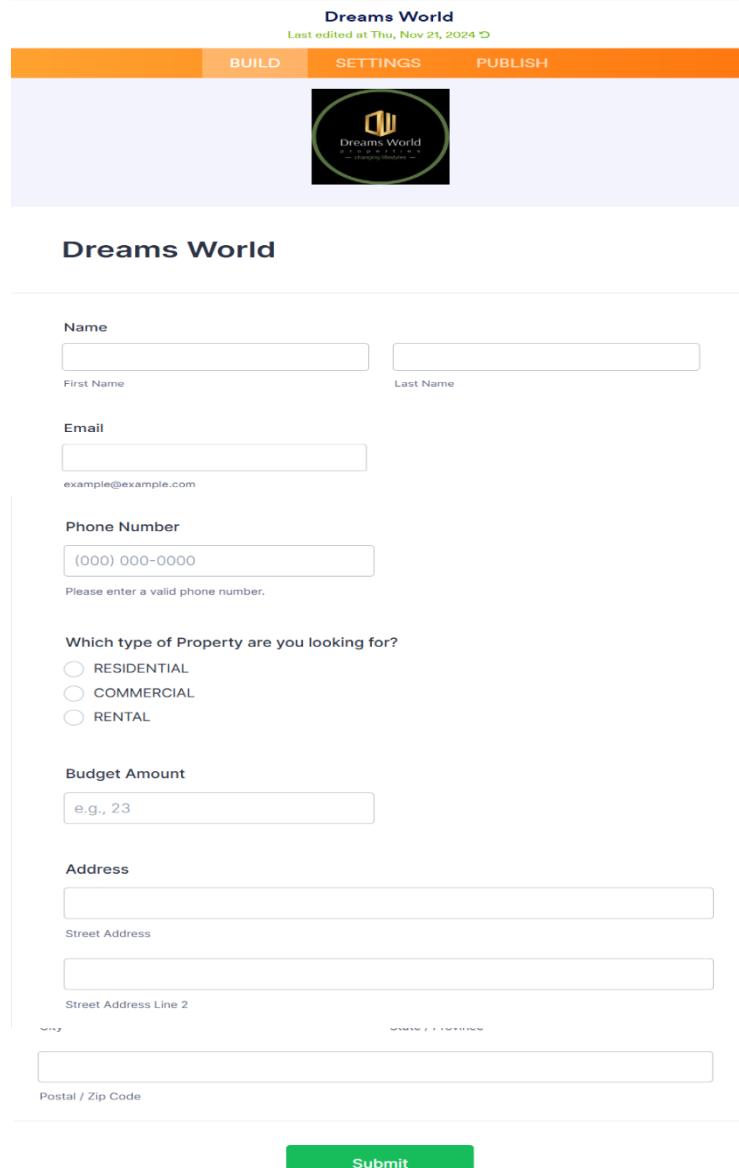
4. Detailed Steps to Solution Design:

STEP 1:

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

Activity: To create a form to get the customer details like Name, Phone, Email, Address and type of property the customer is interested in.

My form: <https://form.jotform.com/jaiharini03/dreams-world>



The screenshot shows a Jotform creation interface for a form titled "Dreams World". The top navigation bar has tabs for "BUILD", "SETTINGS", and "PUBLISH". Below the title, there is a logo for "Dreams World". The main form area contains the following fields:

- Name**: Two input fields for "First Name" and "Last Name".
- Email**: An input field with the placeholder "example@example.com".
- Phone Number**: An input field containing "(000) 000-0000" with a validation message below it: "Please enter a valid phone number."
- Which type of Property are you looking for?**: Three radio buttons for "RESIDENTIAL", "COMMERCIAL", and "RENTAL".
- Budget Amount**: An input field with the placeholder "e.g., 23".
- Address**: Two input fields for "Street Address" and "Street Address Line 2".
- Postal / Zip Code**: An input field.

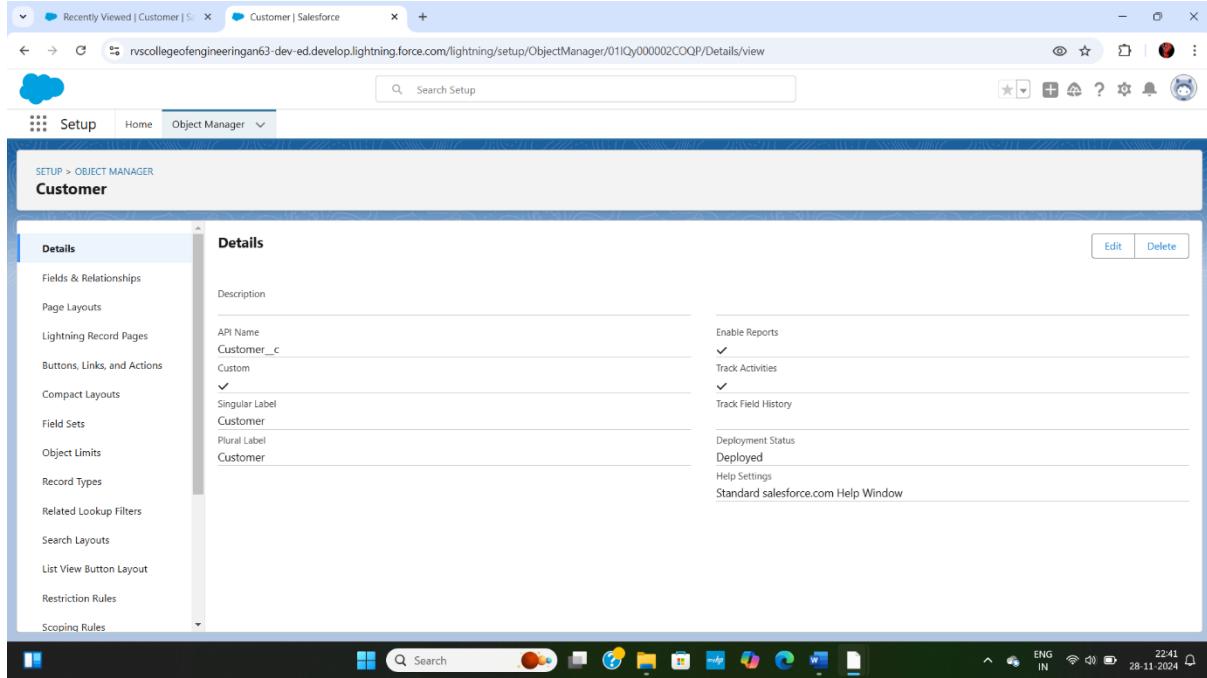
A large green "Submit" button is located at the bottom of the form.

STEP 2:

Create Objects from Spreadsheet.

Activity 1: Customer Object.

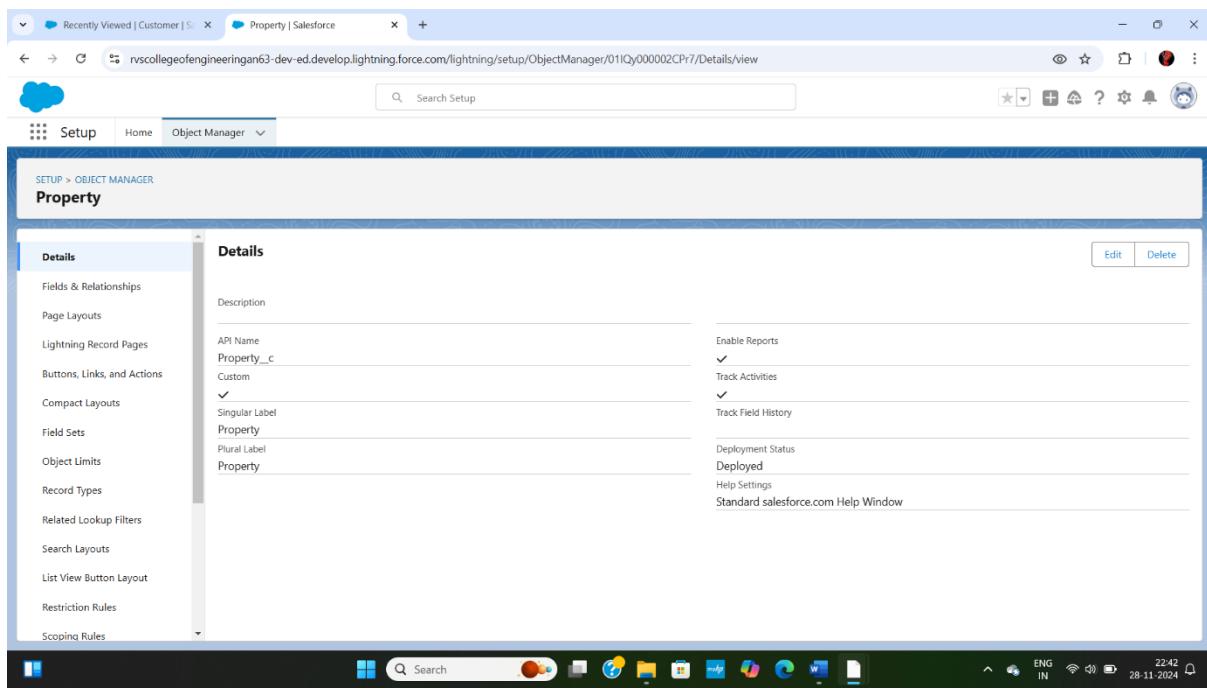
Go to object manager and create object from spreadsheet; map the fields and upload to create **Customer** and **Property** object.



The screenshot shows the Salesforce Object Manager interface for the 'Customer' object. The left sidebar lists various setup options like Fields & Relationships, Page Layouts, and Lightning Record Pages. The main 'Details' tab is selected, showing the following configuration:

- Description:** (empty)
- API Name:** Customer_c
- Custom:** ✓
- Singular Label:** Customer
- Plural Label:** Customer
- Enable Reports:** ✓
- Track Activities:** ✓
- Track Field History:** (empty)
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

Activity 2: Property Object.



The screenshot shows the Salesforce Object Manager interface for the 'Property' object. The left sidebar lists various setup options. The main 'Details' tab is selected, showing the following configuration:

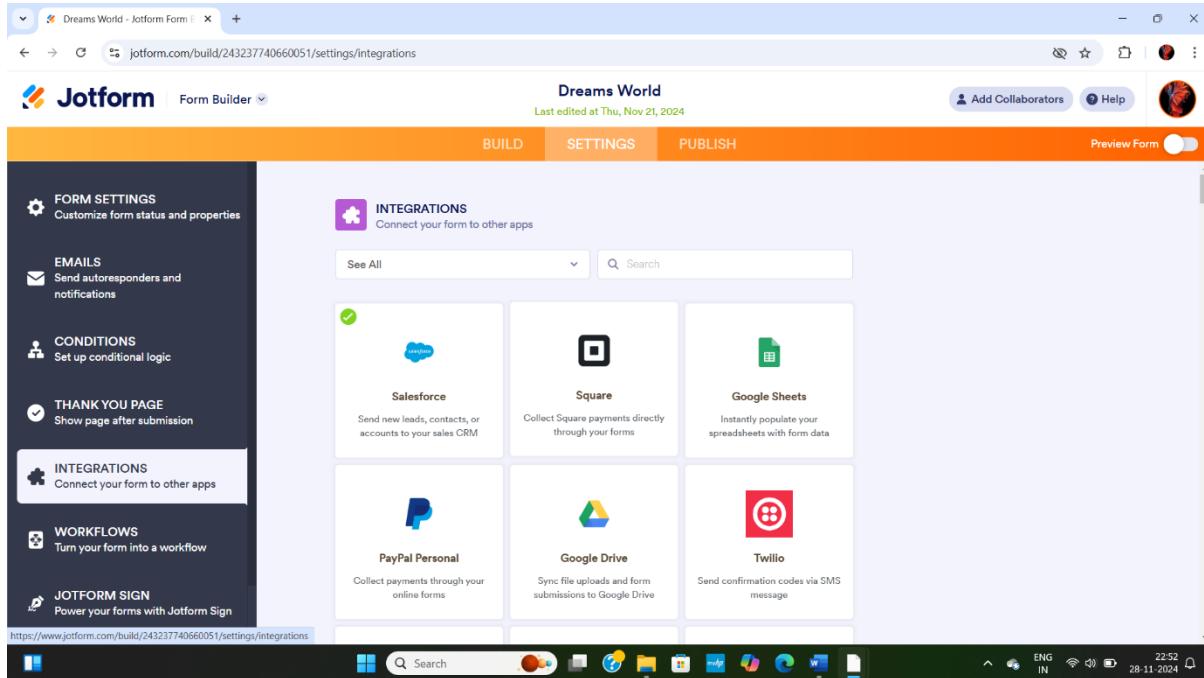
- Description:** (empty)
- API Name:** Property_c
- Custom:** ✓
- Singular Label:** Property
- Plural Label:** Property
- Enable Reports:** ✓
- Track Activities:** ✓
- Track Field History:** (empty)
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

STEP 3:

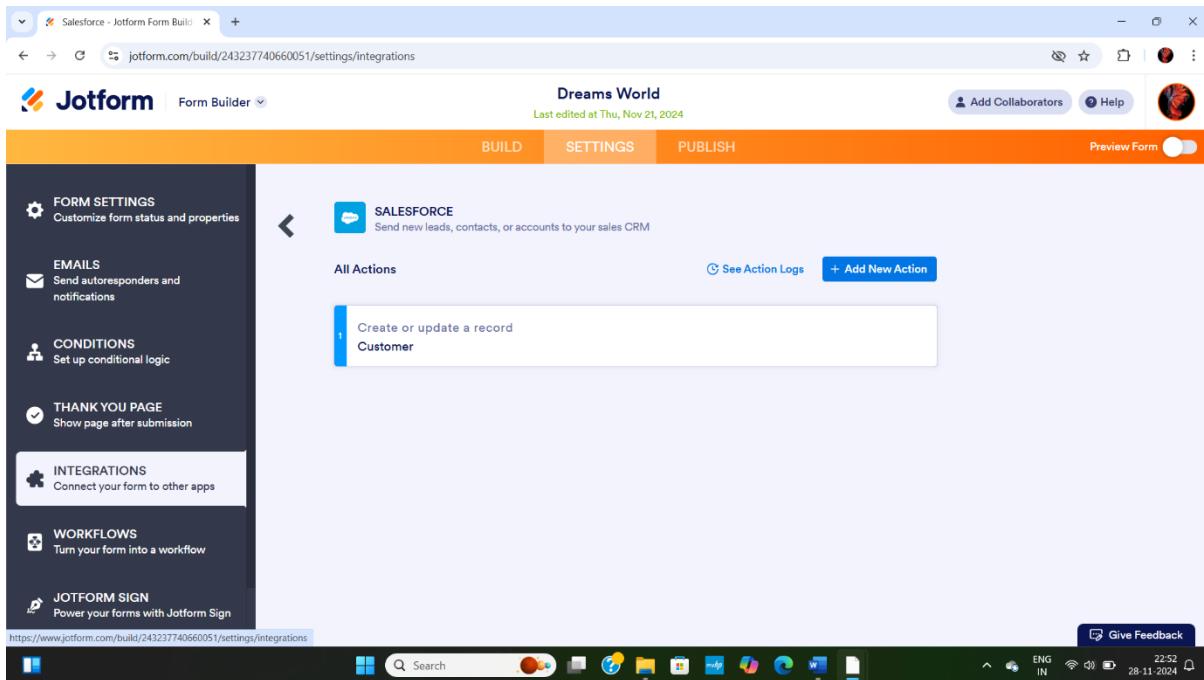
Integrate Jotform with Salesforce Platform

Activity 1:

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



The screenshot shows the Jotform Form Builder interface. The left sidebar has a 'SETTINGS' section with 'INTEGRATIONS' selected. The main area is titled 'INTEGRATIONS' and shows a grid of integration options. The 'Salesforce' option is highlighted with a green checkmark icon. Other options include Square, Google Sheets, PayPal Personal, Google Drive, and Twilio. The top navigation bar shows 'DREAMS WORLD' and the 'SETTINGS' tab is active. The bottom status bar shows the date and time as 28-11-2024 at 22:52.



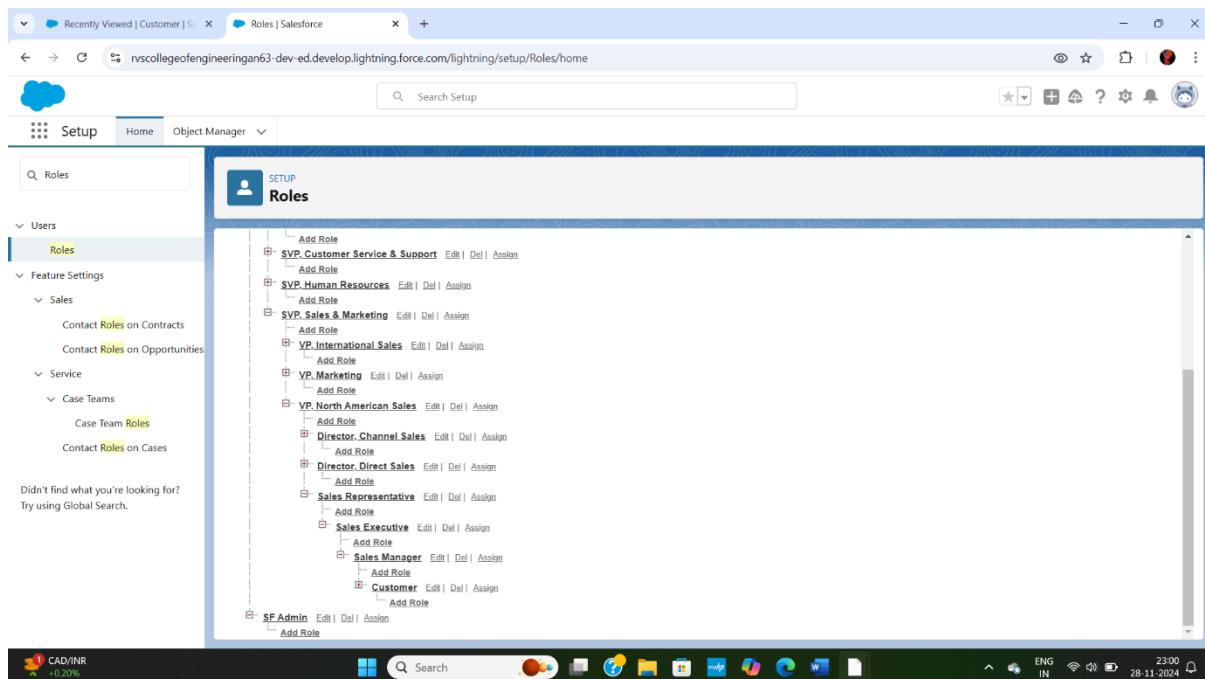
This screenshot shows the same Jotform interface after selecting the 'Salesforce' integration. The main area now displays the 'SALESFORCE' integration details. It shows 'All Actions' and a single action: 'Create or update a record' under the 'Customer' category. The 'See Action Logs' and 'Add New Action' buttons are visible. The rest of the interface and status bar are identical to the previous screenshot.

STEP 4:

Create Roles.

Activity:

Create **Sales Executive Role** below the Sales Representative and **Sales Manager** below Sales Executive which reports to Sales Executive, and add a role “**Customer**” which reports to Sales Manager.



The screenshot shows the Salesforce Setup Roles page. The left sidebar navigation includes 'Users' (selected), 'Feature Settings', 'Sales' (with 'Contact Roles on Contracts' and 'Contact Roles on Opportunities'), 'Service' (with 'Case Teams' and 'Case Team Roles'), and 'Service' (with 'Contact Roles on Cases'). The main content area displays a tree view of roles under 'SVP.Sales & Marketing'. The hierarchy is as follows:

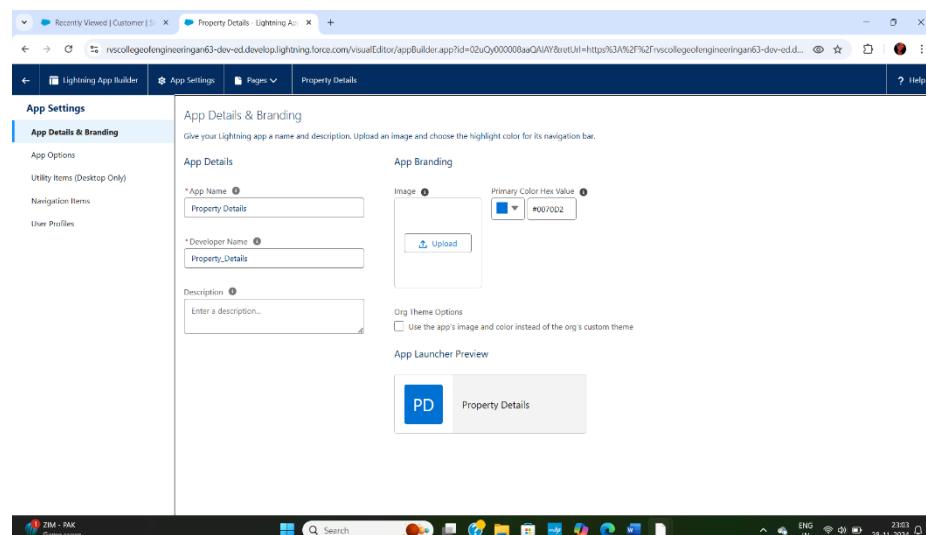
- SVP.Customer Service & Support
- SVP.Human Resources
- SVP.Sales & Marketing
 - VP.International Sales
 - VP.Marketing
 - VP.North American Sales
 - Director.Channel Sales
 - Director.Direct Sales
 - Sales Representative
 - Sales Executive
 - Sales Manager
 - Customer
- SF Admin

STEP 5:

Create a Property Details App.

Activity:

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



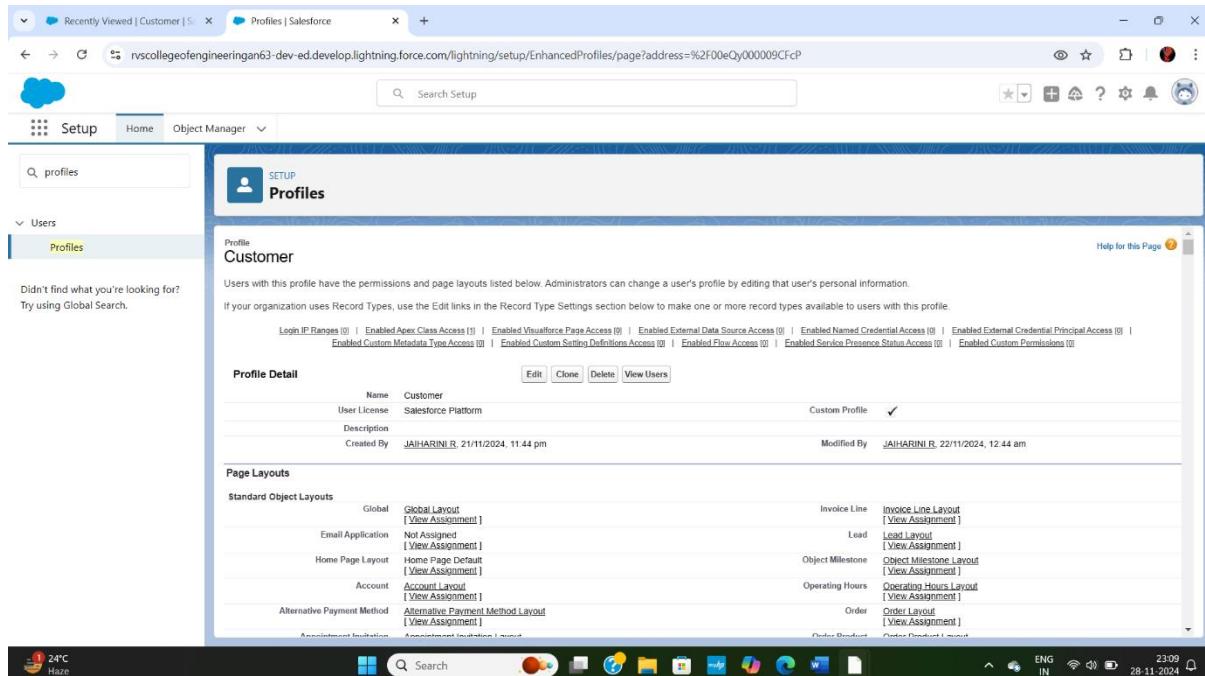
The screenshot shows the App Details & Branding section of the App Builder. The 'App Details' tab is selected. The 'App Name' field contains 'Property Details'. The 'Developer Name' field contains 'Property_Details'. The 'Image' field has a placeholder image and a 'Upload' button. The 'Primary Color Hex Value' is set to '#007022'. The 'Description' field contains 'Enter a description...'. Under 'Org Theme Options', there is a checkbox for 'Use the app's image and color instead of the org's custom theme'. The 'App Launcher Preview' shows a blue button labeled 'PD' and the text 'Property Details'.

STEP 6:

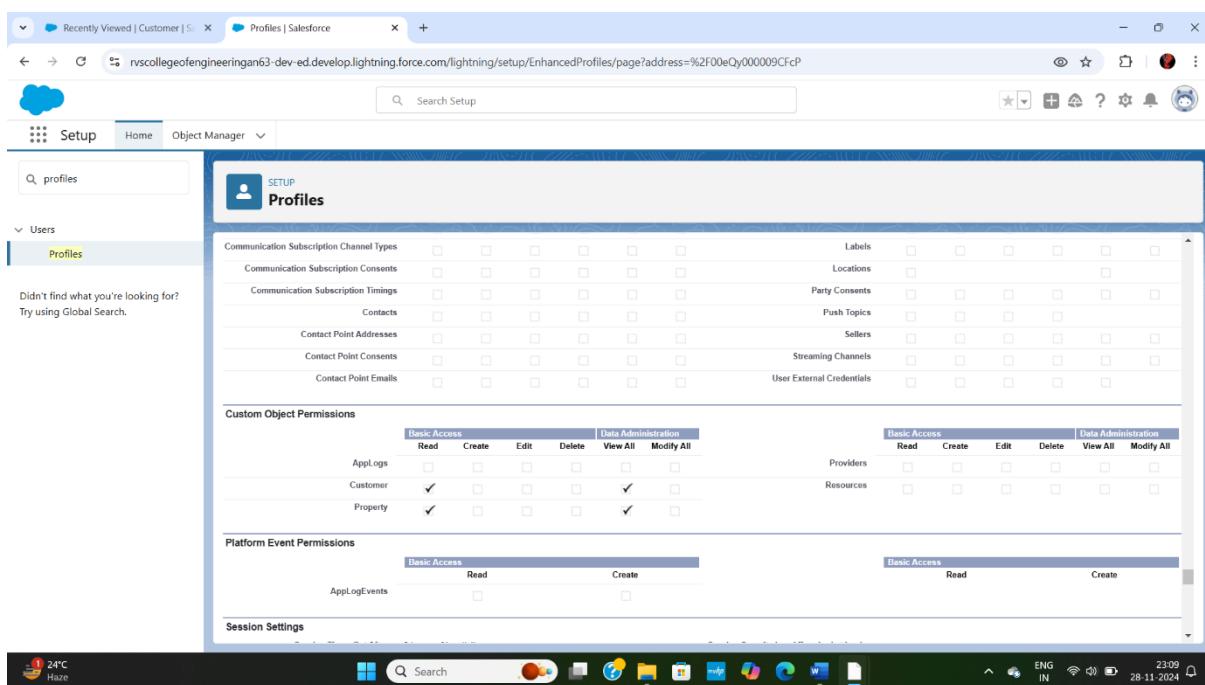
Create Profiles.

Activity 1: Customer

Go to Profiles and Clone Salesforce Platform User and Name it “Customer”.



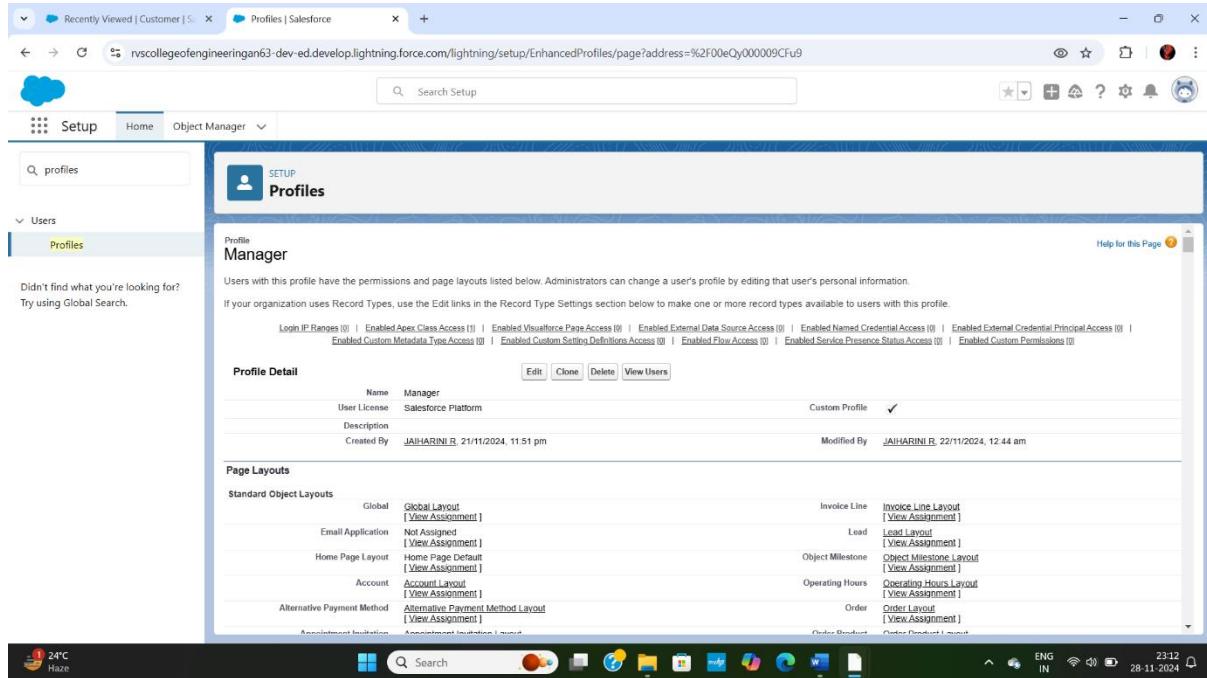
The screenshot shows the Salesforce 'Profiles' page. The 'Customer' profile is selected. The 'Profile Detail' section shows the profile name is 'Customer', user license is 'Salesforce Platform', and it was created by 'JAIHARINI R.' on 21/11/2024 at 11:44 pm. The 'Page Layouts' section lists various standard object layouts like Global, Lead, Object Milestone, and Order. The status bar at the bottom indicates it's 23:09 on 28-11-2024.



The screenshot shows the 'Customer' profile's permission settings. It includes sections for 'Communication Subscription Channel Types', 'Custom Object Permissions' (with rows for AppLogs, Customer, and Property), and 'Platform Event Permissions'. The status bar at the bottom indicates it's 23:09 on 28-11-2024.

Activity 2: Manager

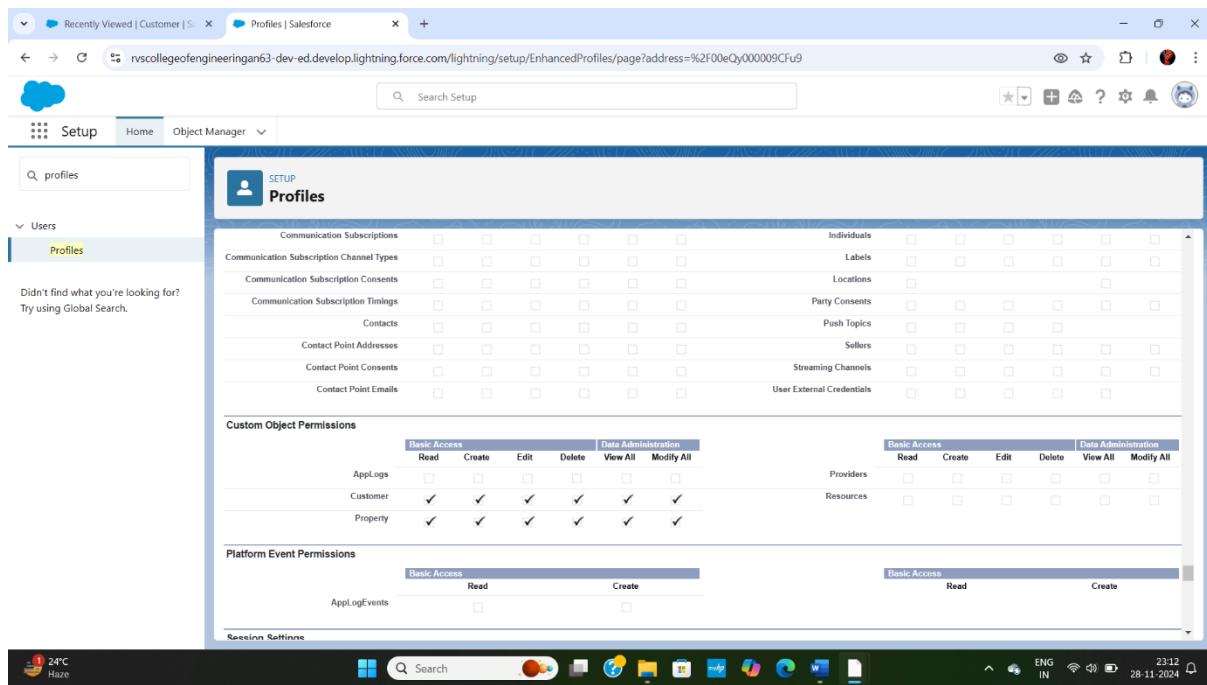
Go to Profiles and Clone Salesforce Platform User and Name it “Manager”.



The screenshot shows the Salesforce Setup interface with the 'Profiles' page selected. A specific profile named 'Manager' is displayed. The profile details include:

- Name:** Manager
- User License:** Salesforce Platform
- Description:** (empty)
- Created By:** JAIHARINI_R, 21/11/2024, 11:51 pm
- Modified By:** JAIHARINI_R, 22/11/2024, 12:44 am

The 'Page Layouts' section shows various standard object layouts assigned to the 'Manager' profile, such as Global Layout, Lead Layout, Object Milestone Layout, and Order Layout.



The screenshot shows the 'Manager' profile's permissions and consents. It includes sections for:

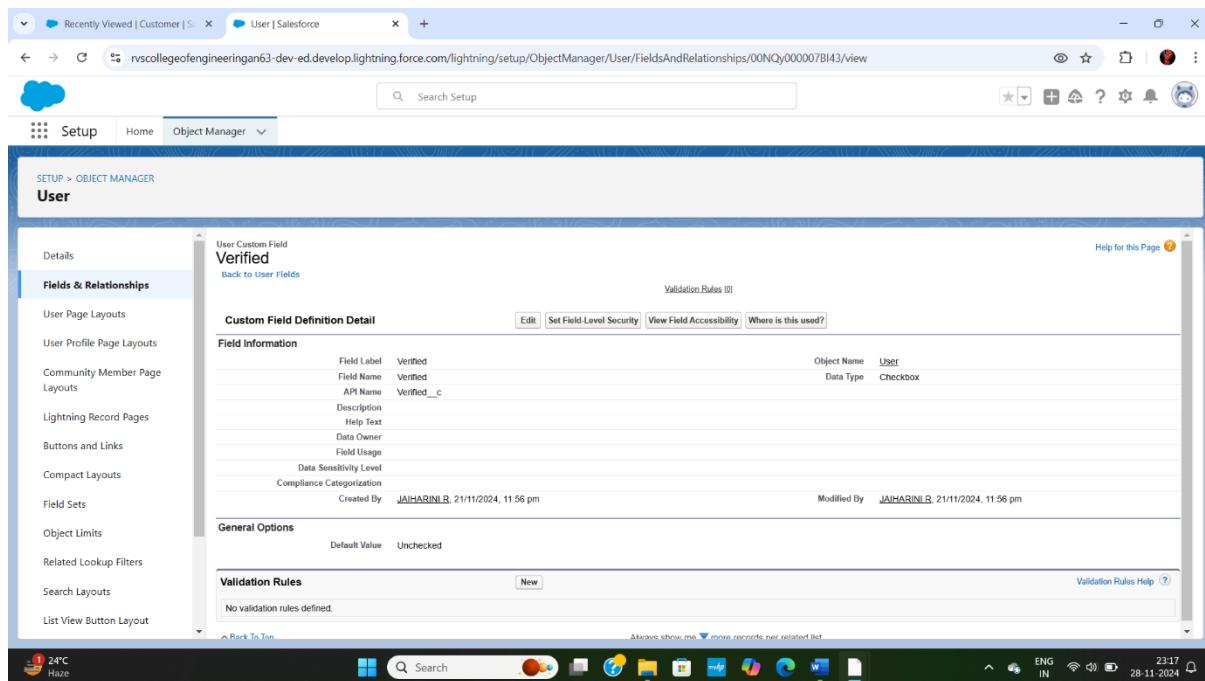
- Communication Subscriptions:** A grid of checkboxes for various communication types.
- Custom Object Permissions:** A detailed table showing permissions for objects like Appl.Logs, Customer, and Property across various access levels (Read, Create, Edit, Delete, View All, Modify All).
- Platform Event Permissions:** A table showing permissions for platform events like AppLogEvents.

STEP 7:

Create a Check Box field on user.

Activity:

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



STEP 8:

Create Users.

Activity 1: User 1

Go to Setup ->Administration -> Users -> New User;
 Last Name - Executive; Role - Sales Executive; License -
 Salesforce; Profile - System Administrator and Save.

Activity 2: User 2

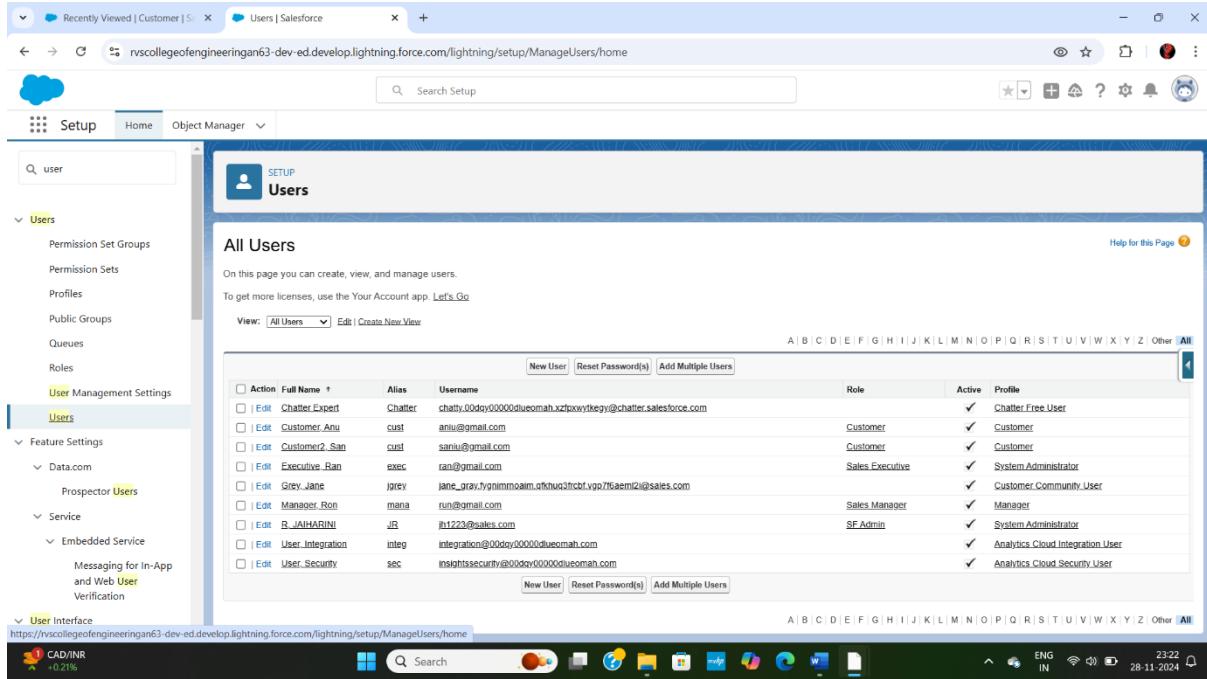
Go to Setup ->Administration ->Users ->New User;
 Last Name - Manager; Role - Sales Manager; License -
 Salesforce Platform; Profile - Manager and Save.

Activity 3: User 3

Go to Setup ->Administration ->Users ->New User;
 Last Name - Customer; Role - Customer; License -
 Salesforce Platform; Profile - Customer; check the
 verified check box is “Unchecked” and save.

Activity 4: User 4

Go to Setup ->Administration ->Users ->New User; Last Name - Customer2; Role - Customer; License - Salesforce Platform; Profile - Customer; check the verified check box is “Unchecked” and save.



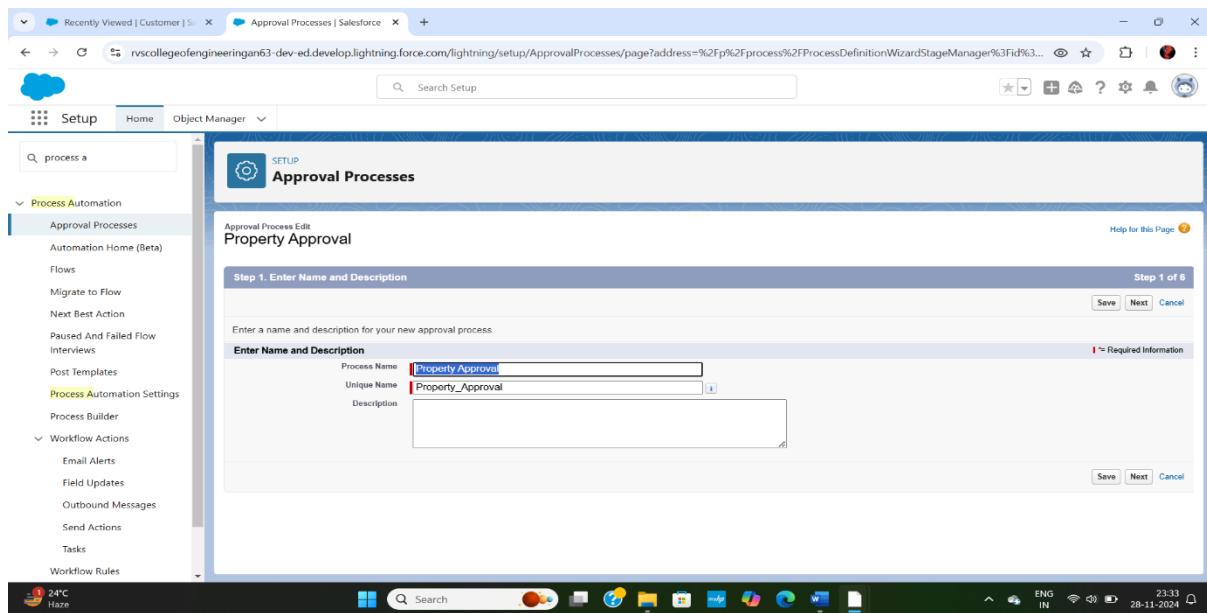
Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter Expert	Chatter	chatty@00000000dueomah.xzfxwytkgy@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	Customer_Anu	cust	anu@gmail.com	Customer	<input checked="" type="checkbox"/>	Customer
<input type="checkbox"/>	Customer2_San	cust	sania@gmail.com	Customer	<input checked="" type="checkbox"/>	Customer
<input type="checkbox"/>	Executive_Ran	exec	ran@gmail.com	Sales Executive	<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Grey_Jane	jgrey	jane_gravy@ynimmoalm.0knudzrbct.vg@7faemzii@sales.com		<input checked="" type="checkbox"/>	Customer Community User
<input type="checkbox"/>	Manager_Ron	mana	runt@gmail.com	Sales Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	R_JAIHARINI	JR	JR123@sales.com	SF Admin	<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	User_Integration	integ	integration@00dy@000000dueomah.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User_Security	sec	insightssecurity@00dy@000000dueomah.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

STEP 9:

Create an Approval Process for Property Object.

Activity:

- From Setup to Process Automation ->Approval Process and Process Name - Property Approval.

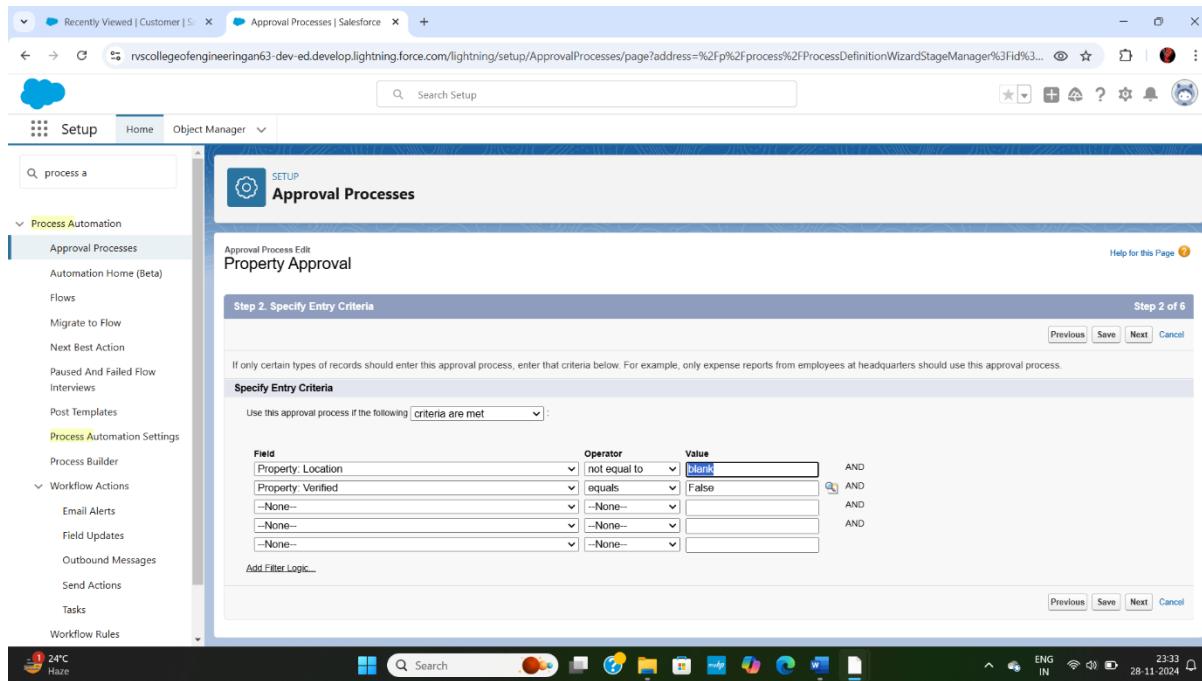


Step 1. Enter Name and Description

Enter a name and description for your new approval process.

Process Name	Property Approval	! = Required Information
Unique Name	Property_Approval	
Description	<div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div>	

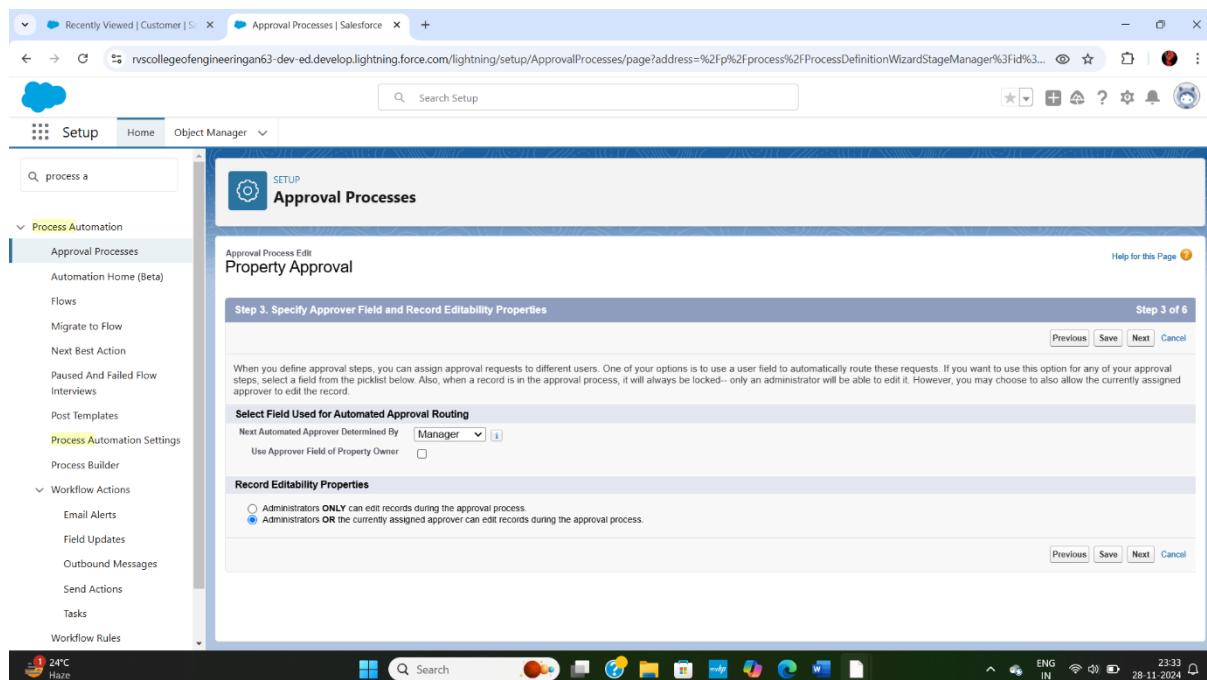
2. Give 2 criteria → Location is not equal to blank, Verified Equals false.



The screenshot shows the 'Approval Processes' setup page for 'Property Approval'. It is on 'Step 2. Specify Entry Criteria'. There are two criteria defined:

- Field: Property: Location, Operator: not equal to, Value: Blank
- Field: Property: Verified, Operator: equals, Value: False

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



The screenshot shows the 'Approval Processes' setup page for 'Property Approval'. It is on 'Step 3. Specify Approver Field and Record Editability Properties'.

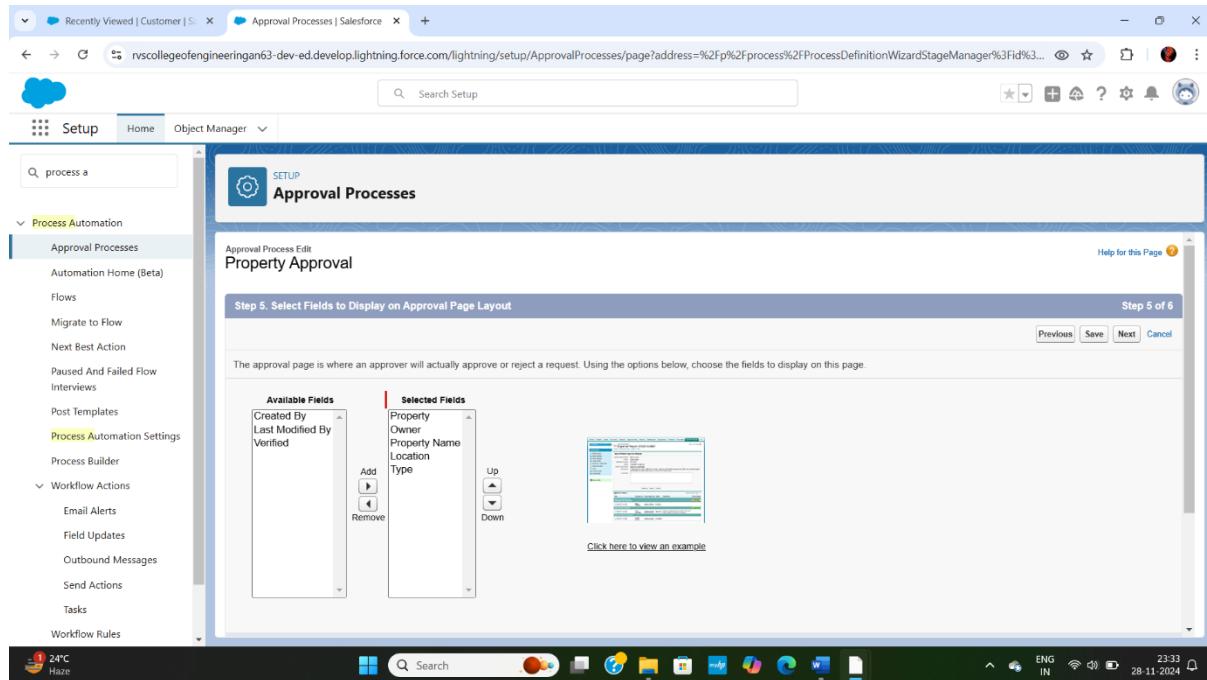
Select Field Used for Automated Approval Routing:

- Next Automated Approver Determined By: Manager
- Use Approver Field of Property Owner:

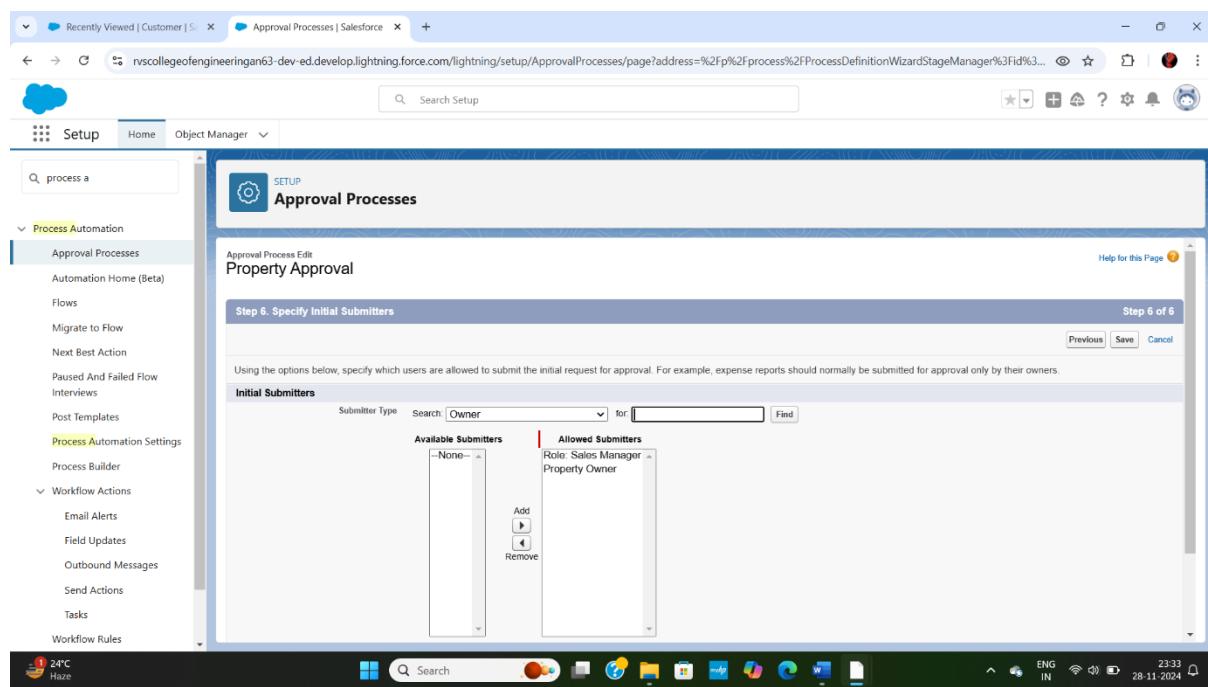
Record Editability Properties:

- Administrators ONLY can edit records during the approval process.
- Administrators OR the currently assigned approver can edit records during the approval process.

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



5. Click Next and Select the initial Submitters >>
- Owner >> Property Owner
 - Roles >> Sales Manager



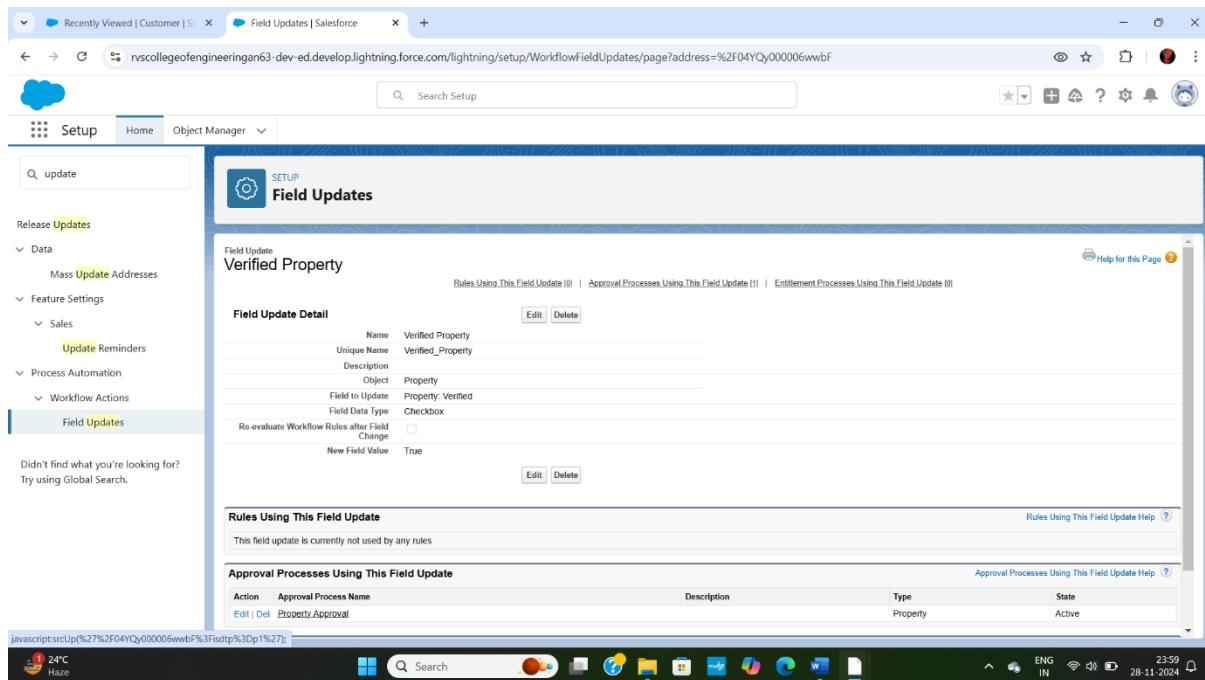
6. Add an approval step name “Executive Approval”; specify the criteria **all record should enter**, click next and select the Approver as “Sales Executive” and “Save”.

The screenshot shows the Salesforce Setup interface for creating a new approval process. The left sidebar is titled 'Process Automation' and includes 'Approval Processes'. The main window is titled 'Approval Step Edit' for 'Executive Approval' and is on 'Step 1 of 3'. The form has fields for 'Approval Process Name' (Property Approval) and 'Unique Name' (Executive_Approval). A 'Description' field is also present. Buttons at the bottom include 'Save', 'Next', and 'Cancel'. The status bar at the bottom right shows '23:49 IN 28-11-2024'.

The screenshot shows the 'Step 2 of 3' screen for 'Executive Approval'. It's titled 'Specify Step Criteria'. The form includes a section for specifying criteria: 'All records should enter this step.' or 'Enter this step if the following criteria are met [dropdown] else [dropdown]'. Buttons at the bottom include 'Previous', 'Save', 'Next', and 'Cancel'. The status bar at the bottom right shows '23:49 IN 28-11-2024'.

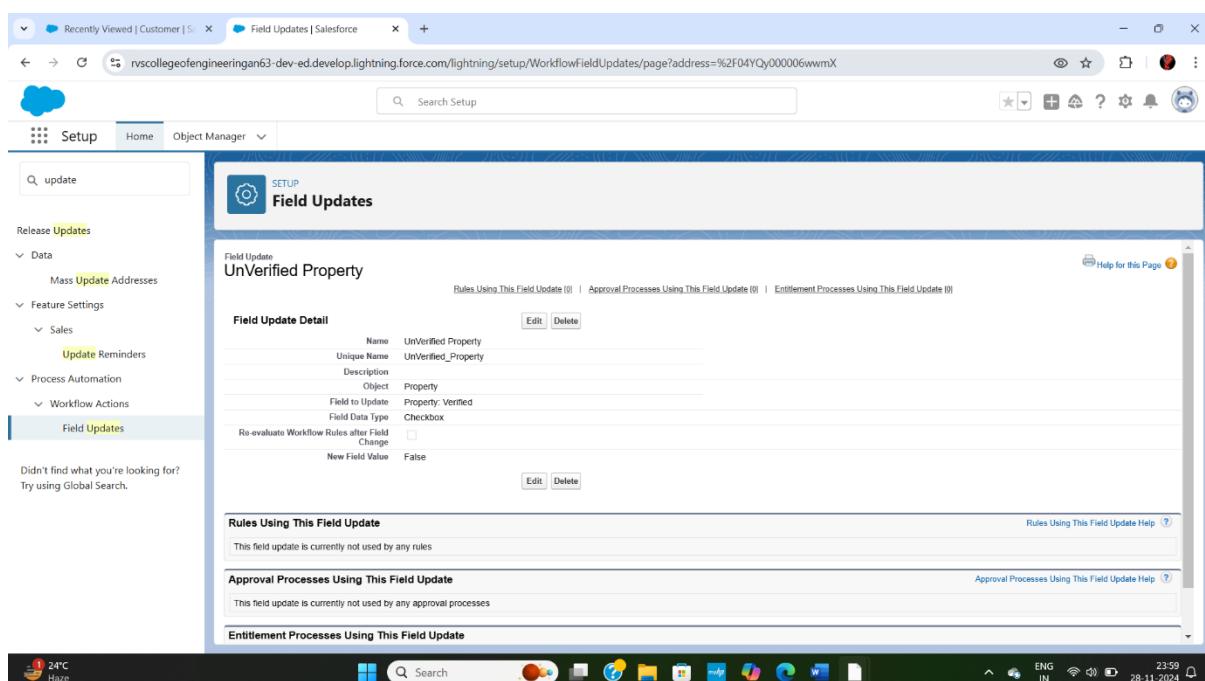
The screenshot shows the 'Step 3 of 3' screen for 'Executive Approval'. It's titled 'Select Assigned Approver'. The form includes a section for selecting approvers: 'Let the submitter choose the approver manually', 'Automatically assign using the user field selected earlier. (Manager)', 'Automatically assign to queue', or 'Automatically assign to approver(s)' (which is selected). A dropdown menu shows 'User' and 'Ran Executive'. A note below says 'When multiple approvers are selected: Approve or reject based on the FIRST response.' or 'Require UNANIMOUS approval from all selected approvers.' A checkbox at the bottom left says 'The approver's delegate may also approve this request.' Buttons at the bottom include 'Previous', 'Save', and 'Cancel'. The status bar at the bottom right shows '23:49 IN 28-11-2024'.

7. Add One field Update as “Verified Property”, Select Object >> Property Field to Update >> Verified Field Data Type >> CheckBox, Select CheckBox Option as “True” and save.



The screenshot shows the Salesforce Setup interface under the 'Field Updates' section. A new field update named 'Verified Property' is being created. The 'Field to Update' is set to 'Property' with 'Property: Verified' selected. The 'Field Data Type' is 'Checkbox'. Under 'New Field Value', the checkbox is checked ('True'). The 'Approval Processes Using This Field Update' section shows a single entry for 'Property Approval'.

8. Add One field Update as “UnVerified Property”, Select Object >> Property Field to Update >> Verified Field Data Type >> CheckBox, Select CheckBox Option as “False” and save.
9. Activate the Approval Process.



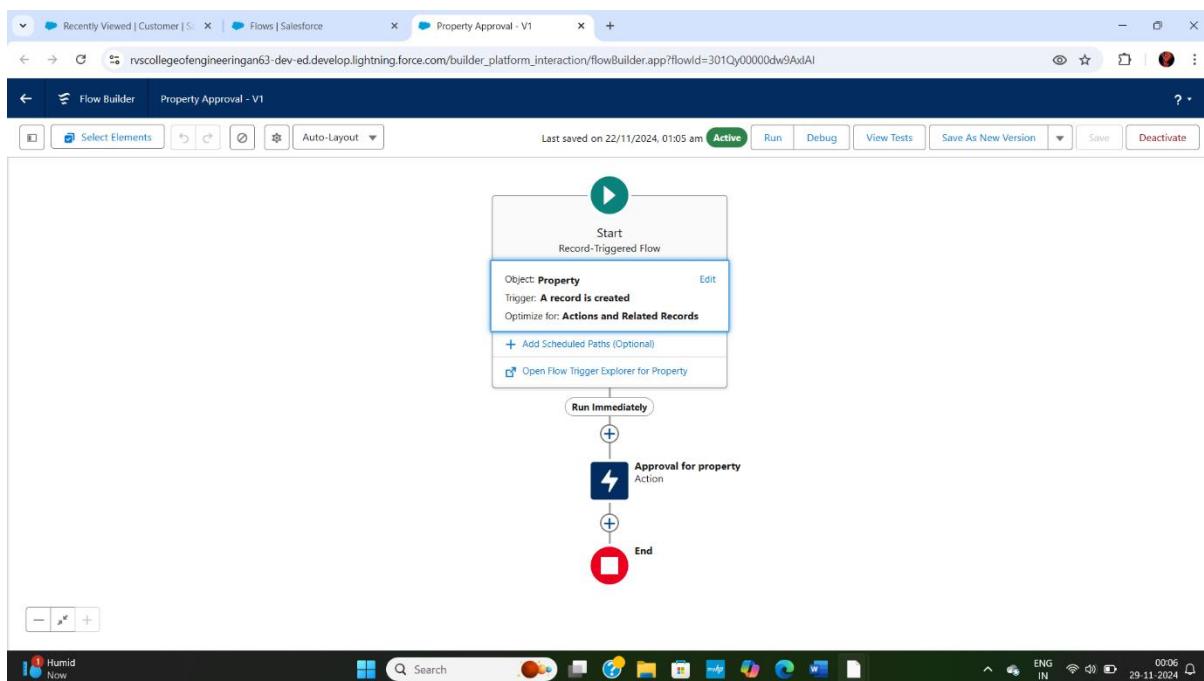
The screenshot shows the Salesforce Setup interface under the 'Field Updates' section. A new field update named 'UnVerified Property' is being created. The 'Field to Update' is set to 'Property' with 'Property: Verified' selected. The 'Field Data Type' is 'Checkbox'. Under 'New Field Value', the checkbox is unchecked ('False'). The 'Approval Processes Using This Field Update' section shows a single entry for 'Property Approval'.

STEP 10:

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

Activity:

From Setup >> Search for Flows >> Click On New and Select “Record Trigger Flow”; Select Object >> Property; Select “Trigger the flow when” >> “A record is created”; Set Entry Conditions >> “None”; Add a “Action” >> “Submit for Approval”; Give Label >> Approval for property; Record Id >> { !\$Record.Id }; Save the Flow and Give label as “Property Approval” and “Activate”.



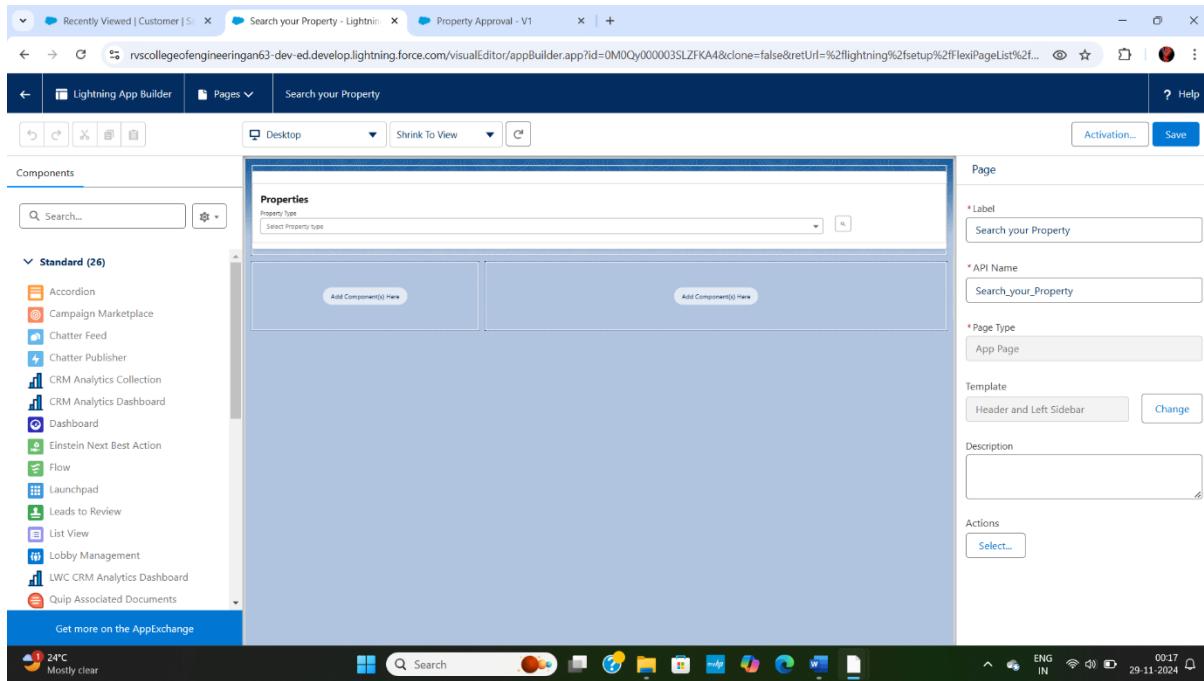
STEP 11:

Create an App Page.

Activity:

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

From Page Setting select page activation as Activate for all Users. From Lightning Experience Click on “Property Details” and click on “Add Page” and save.

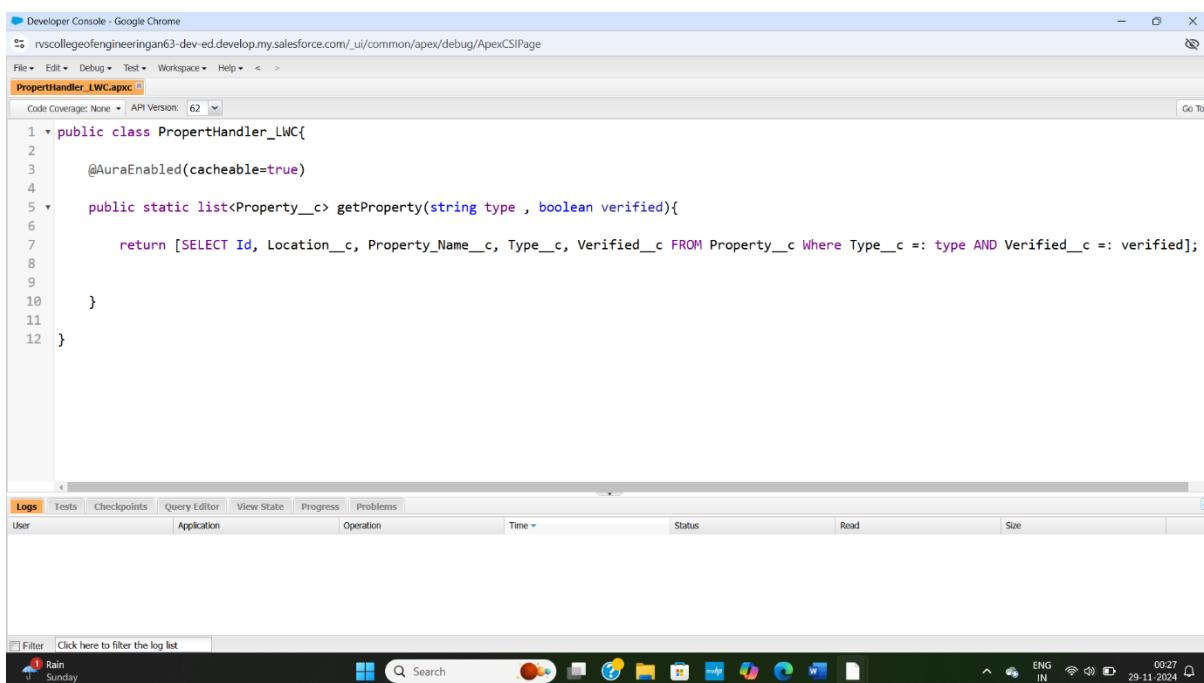


STEP 12:

Create a LWC Component

Activity:

Create an Apex Class and make it aura enabled and name it “PropertHandler_LWC”.



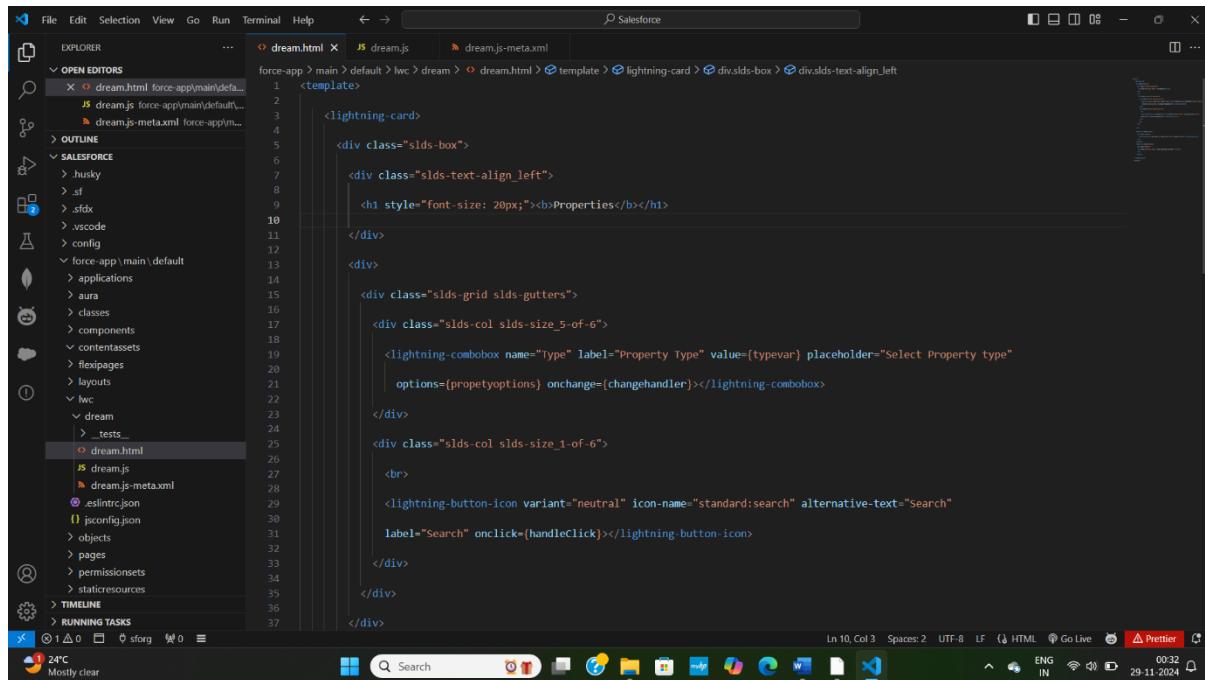
```

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

```

Create a Lightning Web Component in your VsCode, and (ctrl+shift +P) and click on authorize an org. Enter your login id and password to authorize your org. Now (ctrl+shift +P) and Create a lightning Web Component and Name it dream.

1. HTML file:

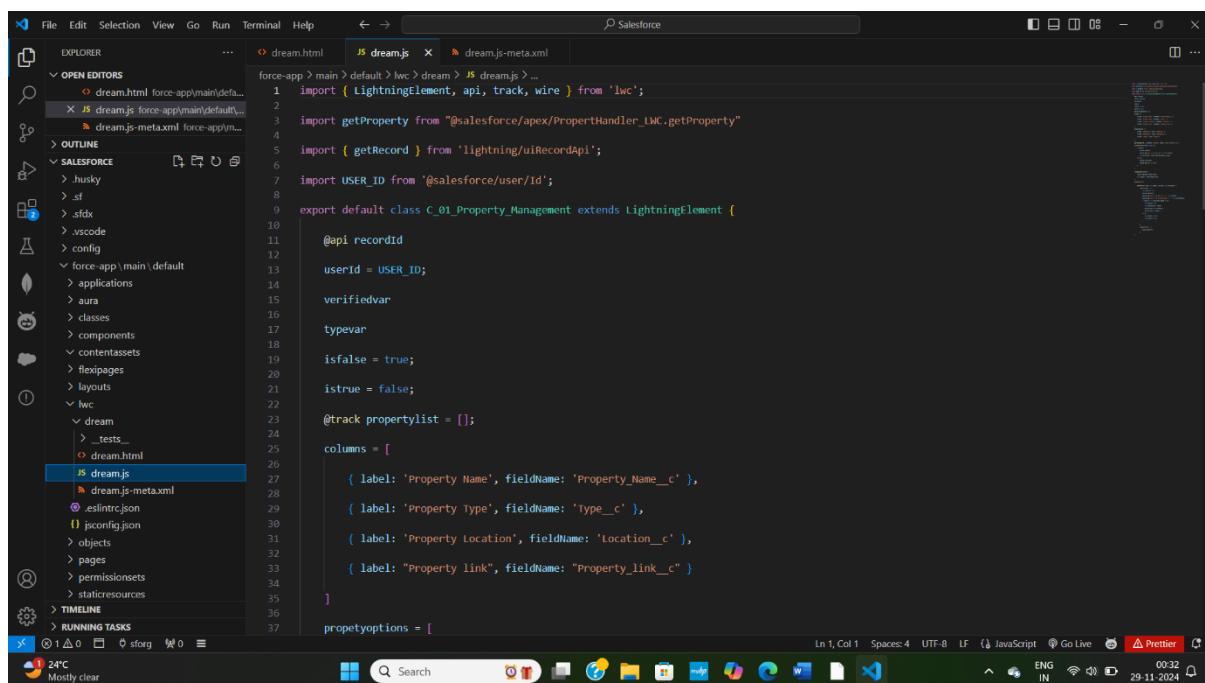


```

<template>
    <lightning_card>
        <div class="slds-text-align_left">
            <div 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>
    </lightning_card>
</template>

```

2. JS file:



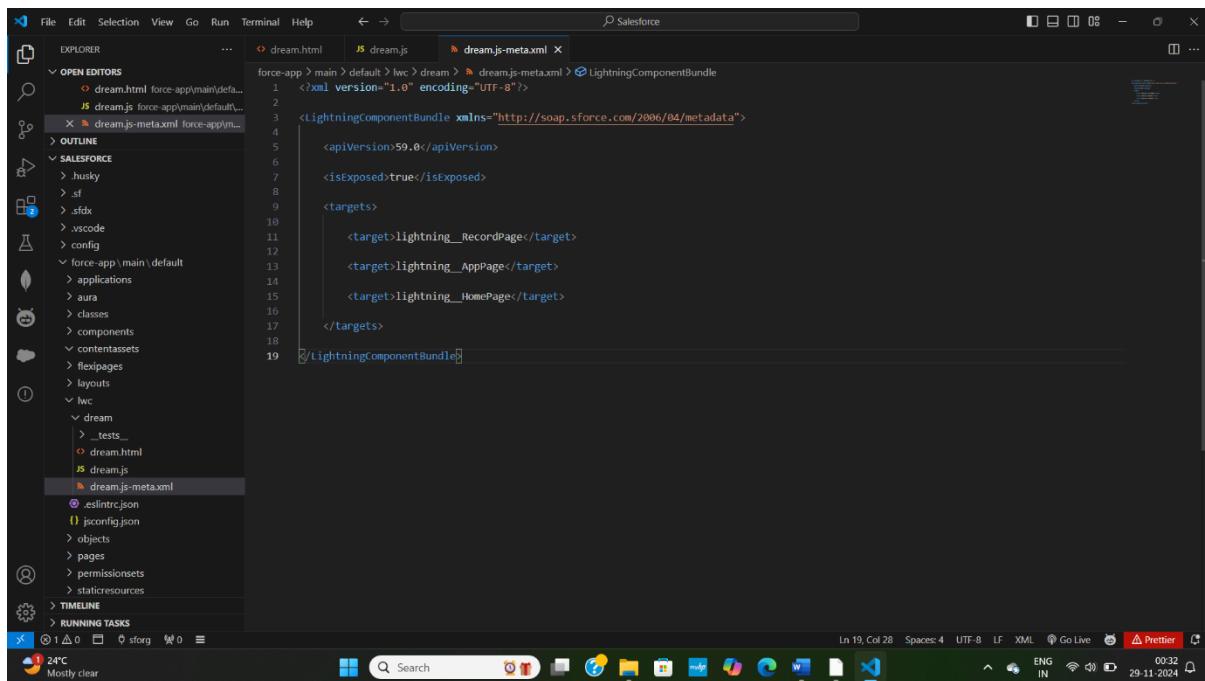
```

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 = [
        ...
    ];
}

```

3. Metofile



```

<?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>

```

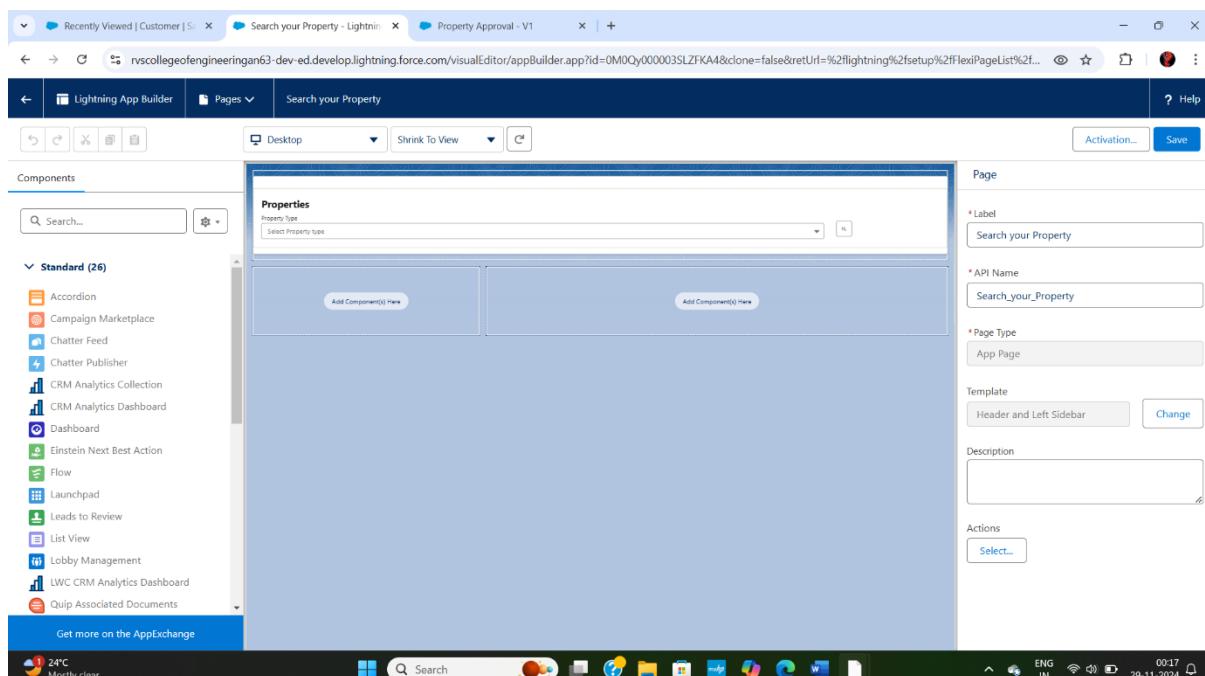
After Saving all the three Codes, Right Click and deploy this component to the org.

STEP 13:

Drag this Component to your App Page.

Activity:

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.

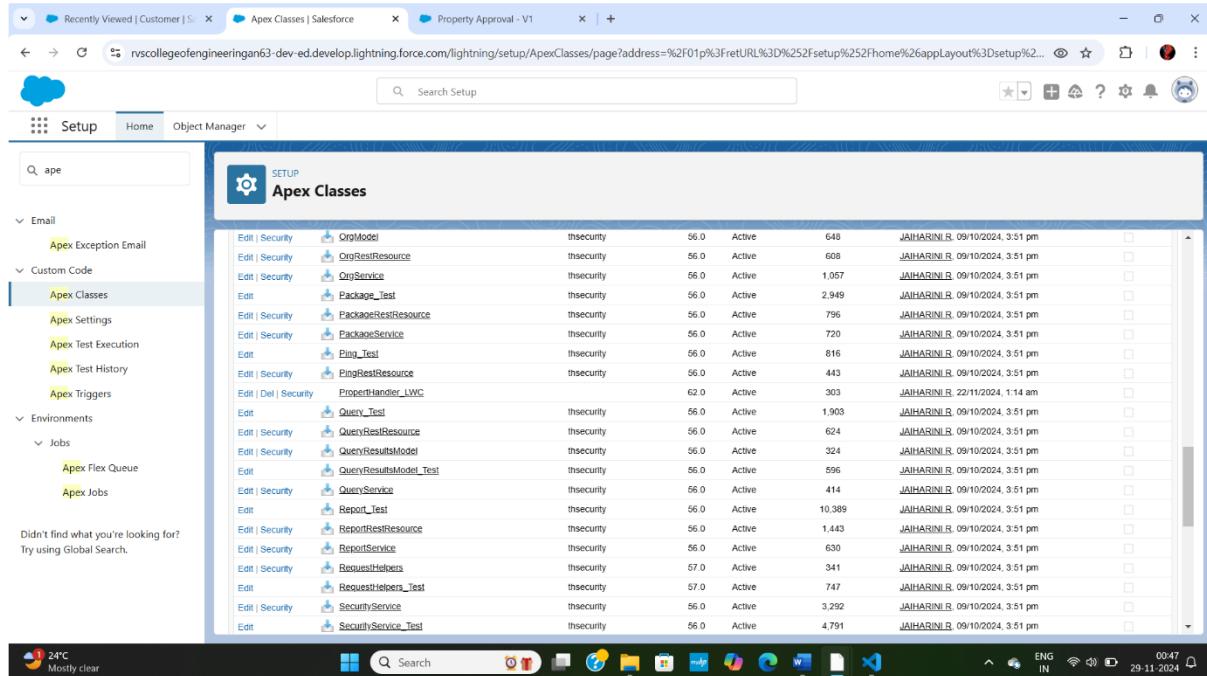


STEP 14:

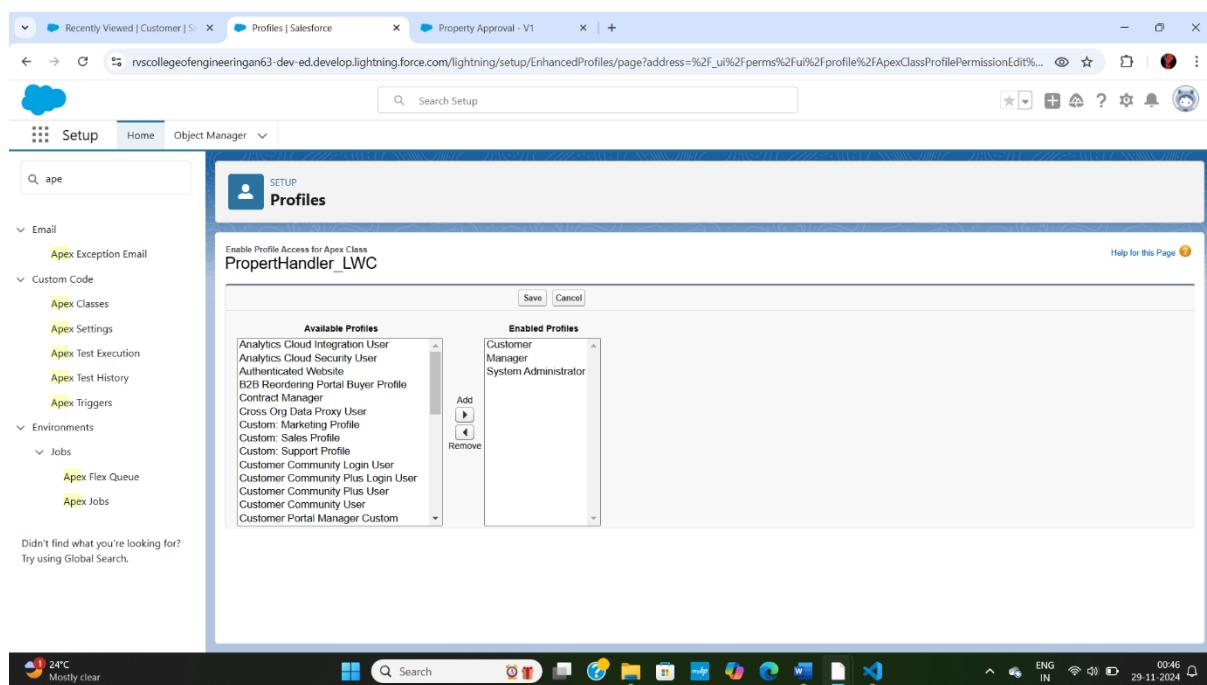
Give Access of Apex Classes to Profiles.

Activity:

From Setup >> Search for Apex Classes >> Click on “Security” behind “PropertyHandler_LWC”. From Profiles Add “Manager” and “Customer” and “Save”.



The screenshot shows the Salesforce Setup interface with the search bar set to "ape". Under the "Apex Classes" section, there is a table listing various Apex classes. Each row includes an "Edit | Security" link. The "Security" column lists the profile name (thsecurity), the number of active users (e.g., 56.0), and the last modified date (e.g., 09/10/2024, 3:51 pm). The "Enabled" column shows checkboxes for each row. The "Last Modified By" column shows the user "JAIHARINI_R". The status column indicates if the class is Active or Inactive.



The screenshot shows the Salesforce Setup interface with the search bar set to "ape". Under the "Profiles" section, a dialog box is open for the "PropertyHandler_LWC" Apex Class. It displays two sections: "Available Profiles" and "Enabled Profiles". The "Available Profiles" section lists numerous profiles such as Analytics Cloud Integration User, Analytics Cloud Security User, Authenticated Website, B2B Reordering Portal Buyer Profile, Contract Manager, Cross Org Data Proxy User, Customer: Marketing Profile, Customer: Sales Profile, Customer: Service Profile, Customer Community Login User, Customer Community Plus Login User, Customer Community Plus User, Customer Community User, and Customer Portal Manager Custom. The "Enabled Profiles" section shows three profiles: Customer, Manager, and System Administrator. There are "Add" and "Remove" buttons between the two sections. At the top right of the dialog box, there is a "Help for this Page" link.

5. Testing and Validation

To ensure the Salesforce CRM application meets the required standards, we will employ a comprehensive testing approach, including:

Unit Testing (Apex Classes, Triggers):

- 1. Apex Unit Tests:** Write unit tests for Apex classes and triggers to validate their functionality and ensure they meet the required standards.
- 2. Test Classes:** Create test classes to test Apex classes and triggers in isolation.
- 3. Test Methods:** Write test methods to validate specific scenarios and edge cases.
- 4. Code Coverage:** Ensure that the Apex code coverage is at least 75% to guarantee that the code is thoroughly tested.
- 5. Test Data:** Use test data to simulate real-world scenarios and validate the application's behaviour.

User Interface Testing:

- 1. User Interface (UI) Testing:** Perform UI testing to validate the application's user interface and ensure it meets the required standards.
- 2. User Acceptance Testing (UAT):** Conduct UAT to validate the application's functionality and ensure it meets the business requirements.
- 3. Test Scenarios:** Create test scenarios to validate specific user interactions and workflows.
- 4. Test Data:** Use test data to simulate real-world scenarios and validate the application's behaviour.
- 5. Browser and Device Testing:** Perform testing on different browsers and devices to ensure compatibility.

6. Key Scenarios Addressed by Salesforce in the Implementation Project

1. Client Onboarding:

- 1. Scenario:** Automate client onboarding process to reduce manual effort and improve client experience.

2. Salesforce Solution: Utilize Salesforce's out-of-the-box features, such as Account and Contact creation, to automate client onboarding. Leverage Process Builder and Workflow to assign tasks and send notifications to relevant stakeholders.

2. Property Management:

- 1. Scenario:** Manage property-related information, including property type, location, and ownership details.
- 2. Salesforce Solution:** Create a custom object, "Property," to store property-related information. Utilize relationships to link properties to clients and accounts.

3. Service Management:

- 1. Scenario:** Manage various services offered to clients, including property maintenance, repairs, and renovations.
- 2. Salesforce Solution:** Utilize Salesforce's Service Cloud features, such as Cases and Work Orders, to manage service requests and delivery.

4. Reporting and Analytics:

- 1. Scenario:** Provide insights into client interactions, property management, and service delivery.
- 2. Salesforce Solution:** Leverage Salesforce's reporting and analytics features, such as Reports and Dashboards, to provide real-time insights into key business metrics.

5. Security and Compliance:

- 1. Scenario:** Ensure the security and integrity of client data and comply with regulatory requirements.
- 2. Salesforce Solution:** Utilize Salesforce's built-in security features, such as encryption and access controls, to protect client data. Leverage Salesforce's compliance features, such as GDPR and HIPAA compliance, to meet regulatory requirements.

7. Conclusion:

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.