



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

## 1. Aim:

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.

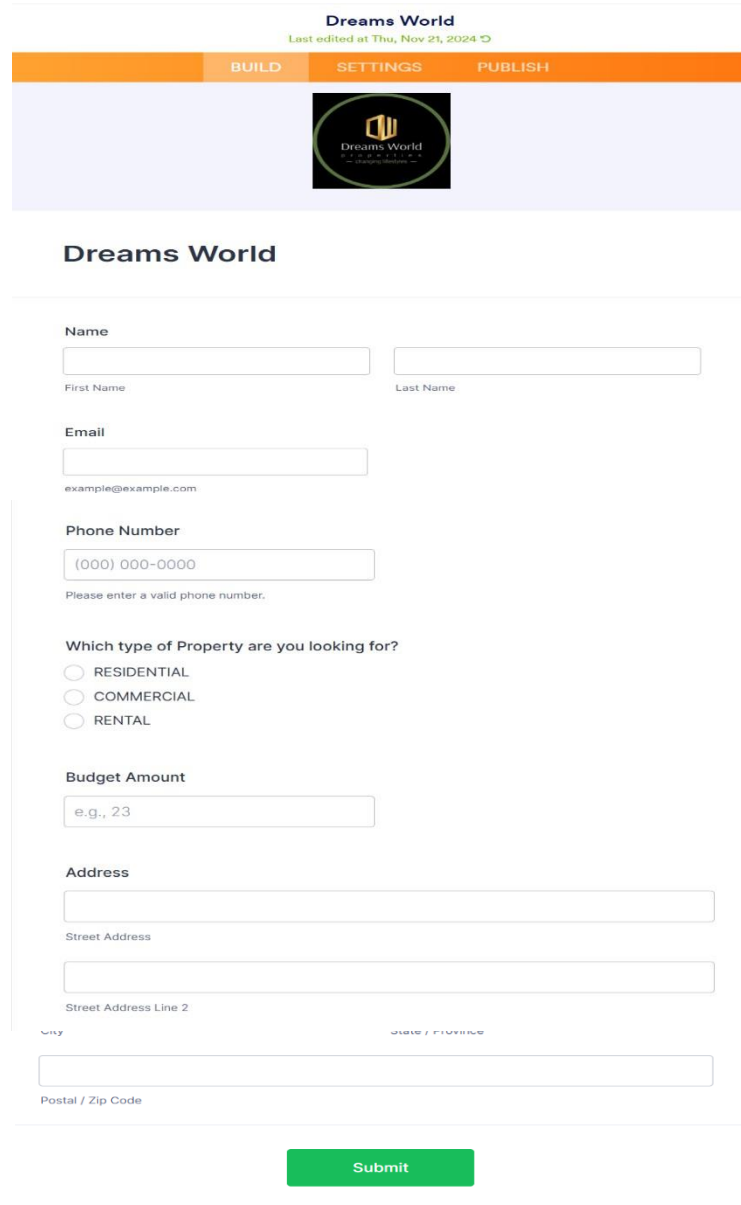
## 4. Detailed Steps to Solution Design:

### 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 titled "Dreams World" with a subtitle "Last edited at Thu, Nov 21, 2024". The form has three tabs: "BUILD", "SETTINGS", and "PUBLISH". The "BUILD" tab is active, showing the form layout. The form includes a logo for "Dreams World" and the following fields:

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

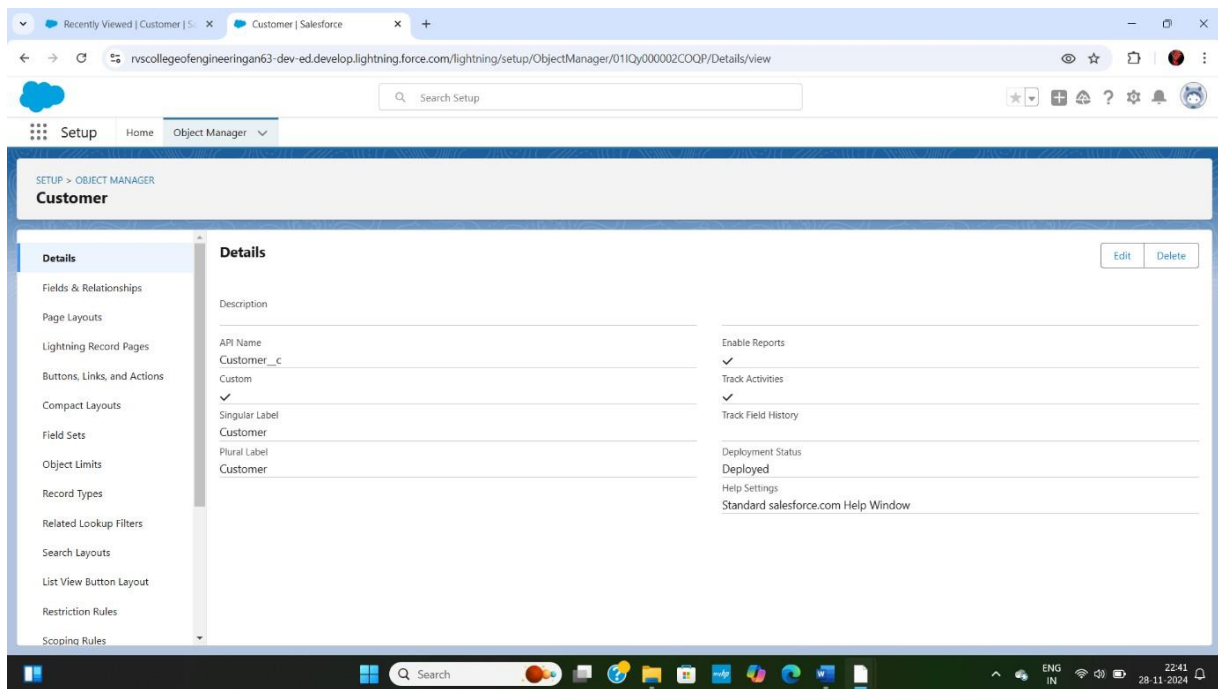
A green "Submit" button is located at the bottom of the form.

## 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.

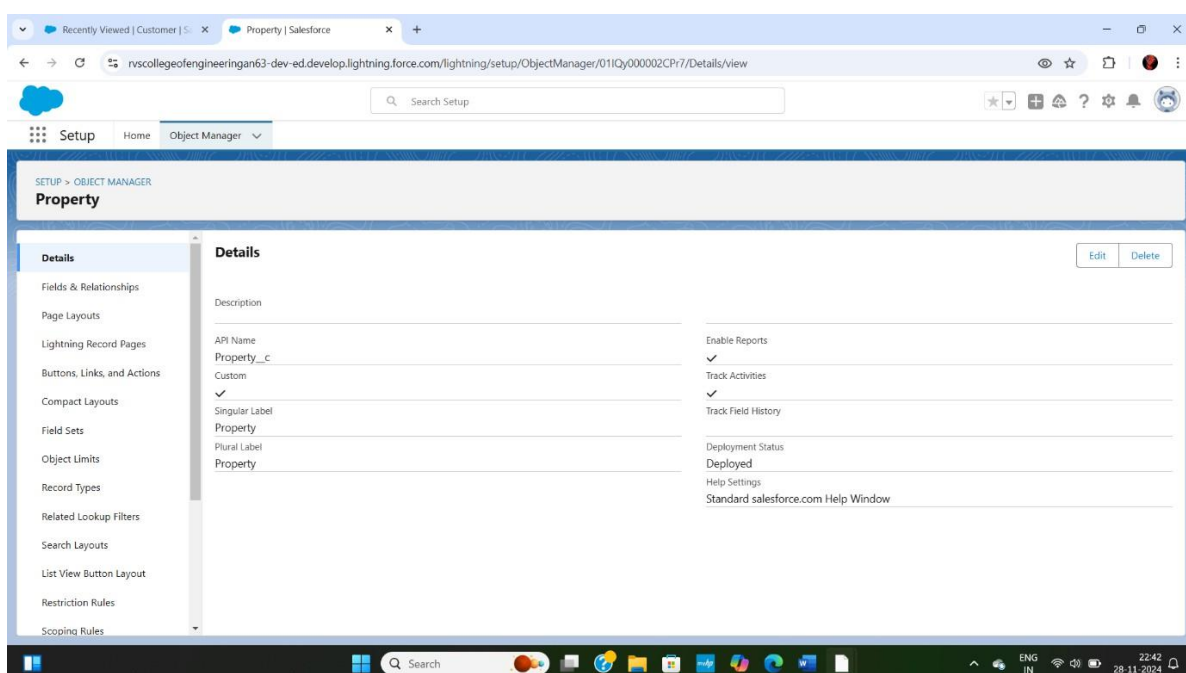


The screenshot shows the Salesforce Object Manager interface for the 'Customer' object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main content area is titled 'Customer' and includes a 'Details' section with the following fields:

- Description: (empty text field)
- API Name: Customer\_\_c
- Custom: ☒
- Singular Label: Customer
- Plural Label: Customer
- Enable Reports: ☒
- Track Activities: ☒
- Track Field History: ☒
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

Buttons for 'Edit' and 'Delete' are located in the top right corner of the details section.

### Activity 2: Property Object.



The screenshot shows the Salesforce Object Manager interface for the 'Property' object. The left sidebar contains a navigation menu with options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main content area is titled 'Property' and includes a 'Details' section with the following fields:

- Description: (empty text field)
- API Name: Property\_\_c
- Custom: ☒
- Singular Label: Property
- Plural Label: Property
- Enable Reports: ☒
- Track Activities: ☒
- Track Field History: ☒
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

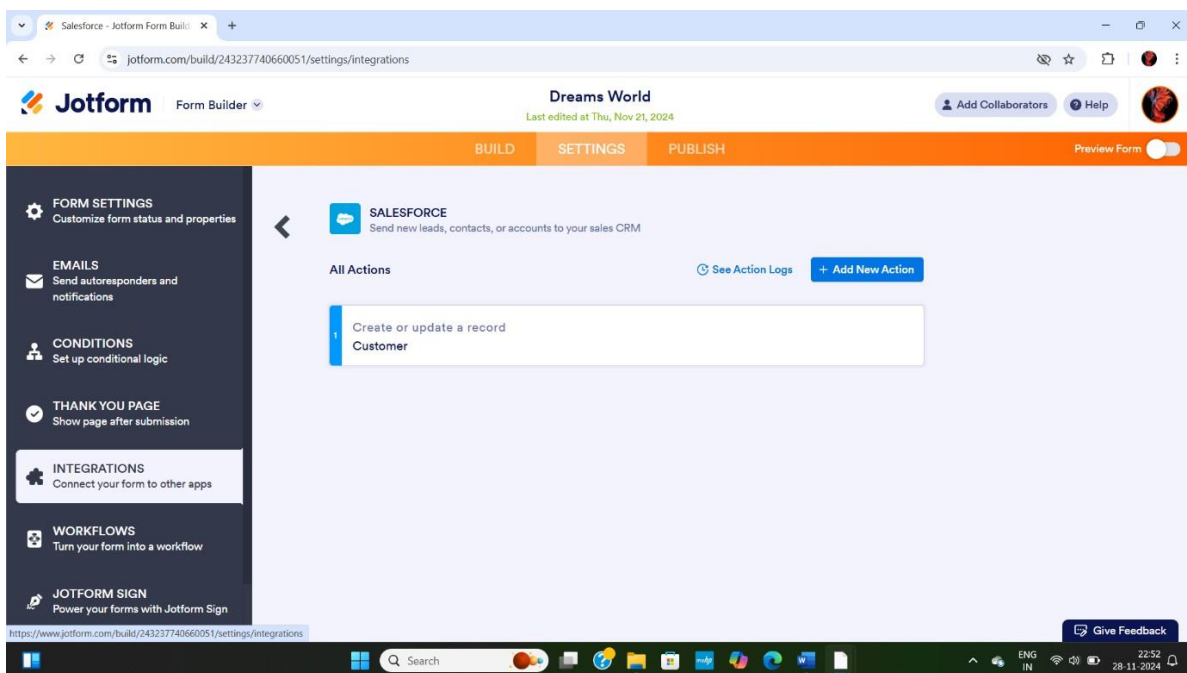
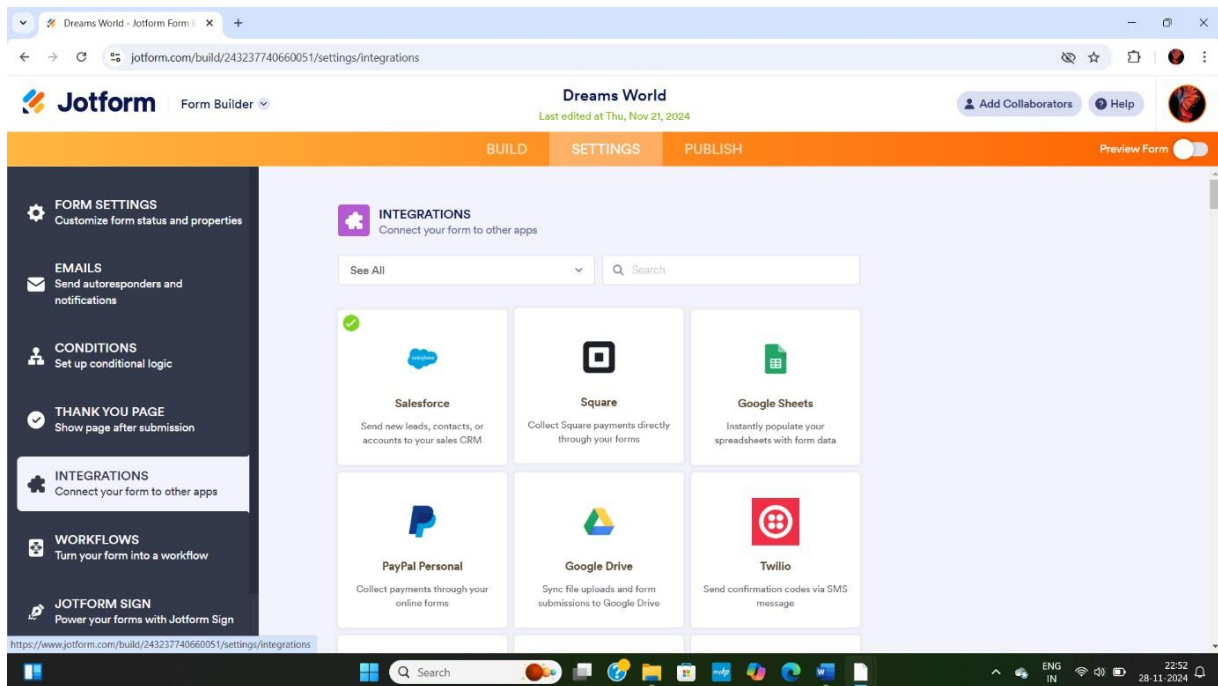
Buttons for 'Edit' and 'Delete' are located in the top right corner of the details section.

## STEP 3:

### Integrate Jotform with Salesforce Platform

#### Activity 1:

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

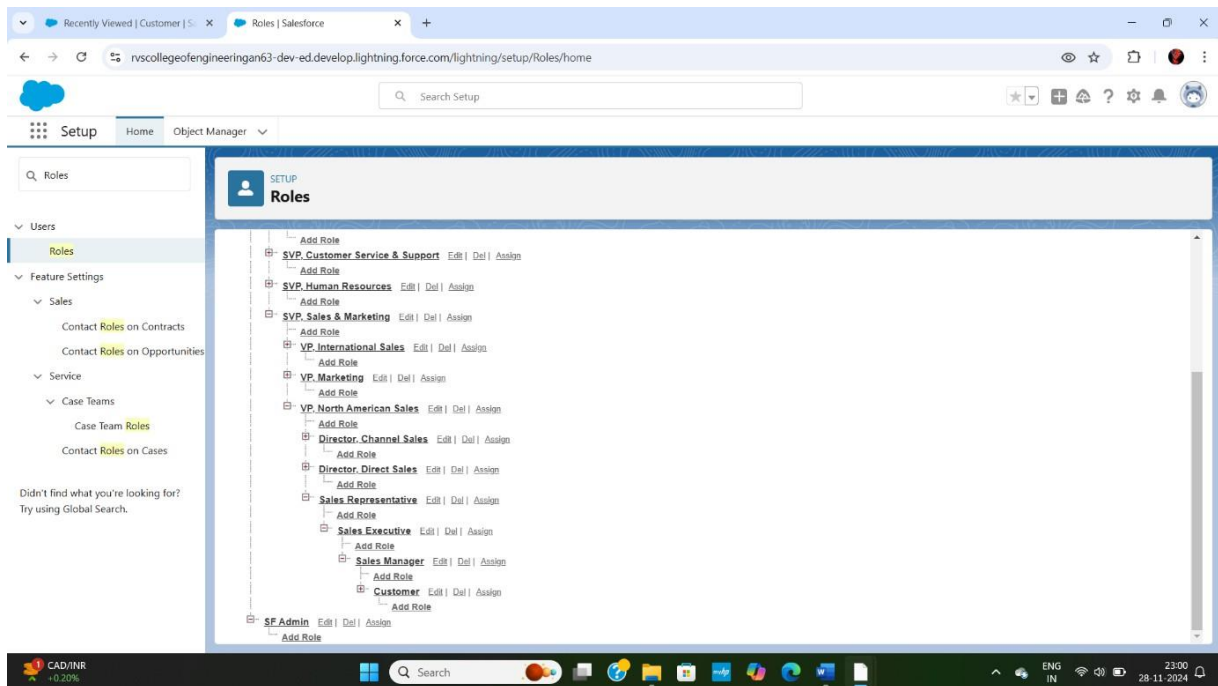


## STEP 4:

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.

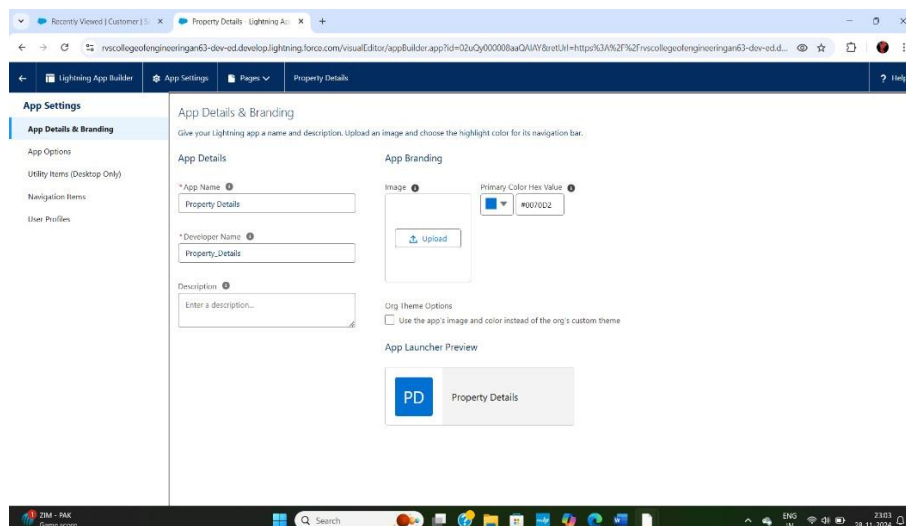


## STEP 5:

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.

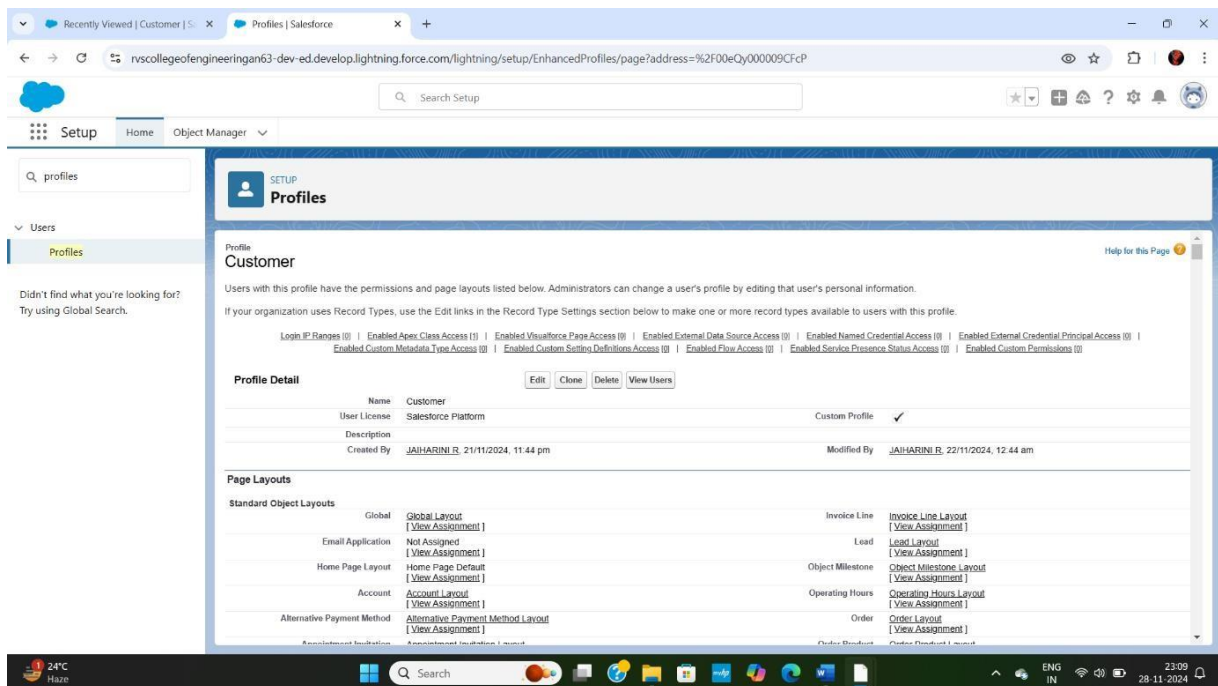


## STEP 6:

Create Profiles.

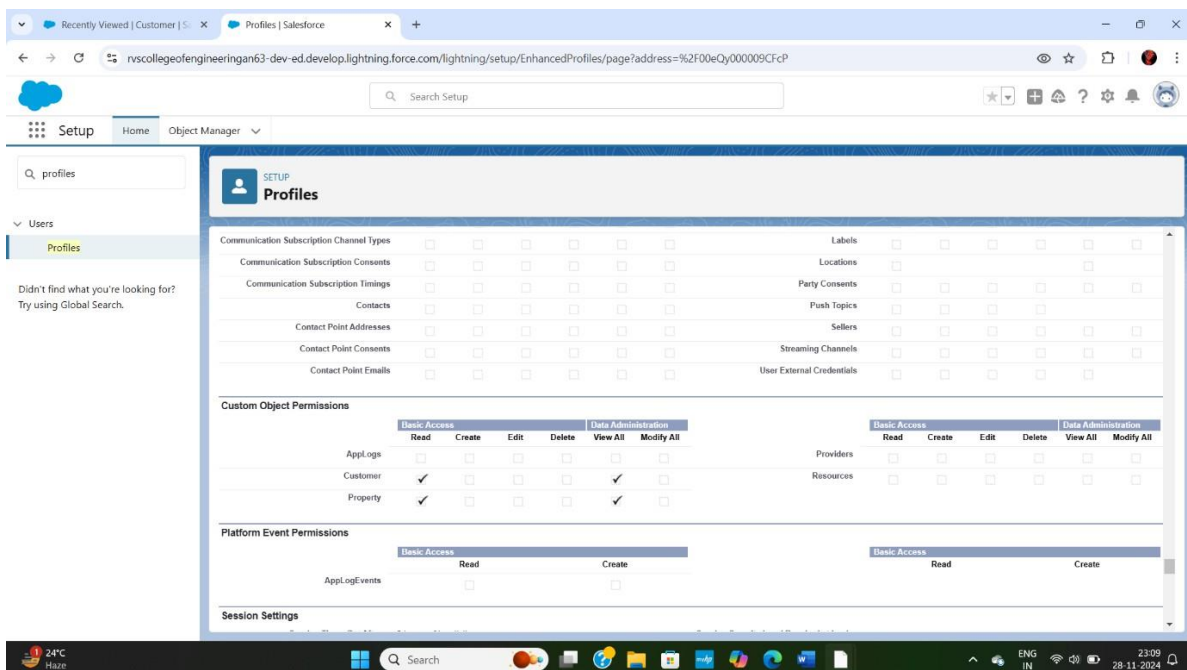
### Activity 1: Customer

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



The screenshot shows the Salesforce Setup interface for the 'Customer' profile. The left sidebar includes 'Setup', 'Home', and 'Object Manager'. The main content area displays the 'Profile: Customer' details. Below the profile name, there is a section for 'Page Layouts' with a table of standard object layouts. The table lists various objects and their assigned layouts, with links to view assignments.

Standard Object Layouts	Global	Global Layout [View Assignment]	Invoice Line	Invoice Line Layout [View Assignment]
Email Application	Not Assigned [View Assignment]		Lead	Lead Layout [View Assignment]
Home Page Layout	Home Page Default [View Assignment]		Object Milestone	Object Milestone Layout [View Assignment]
Account	Account Layout [View Assignment]		Operating Hours	Operating Hours Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout [View Assignment]		Order	Order Layout [View Assignment]



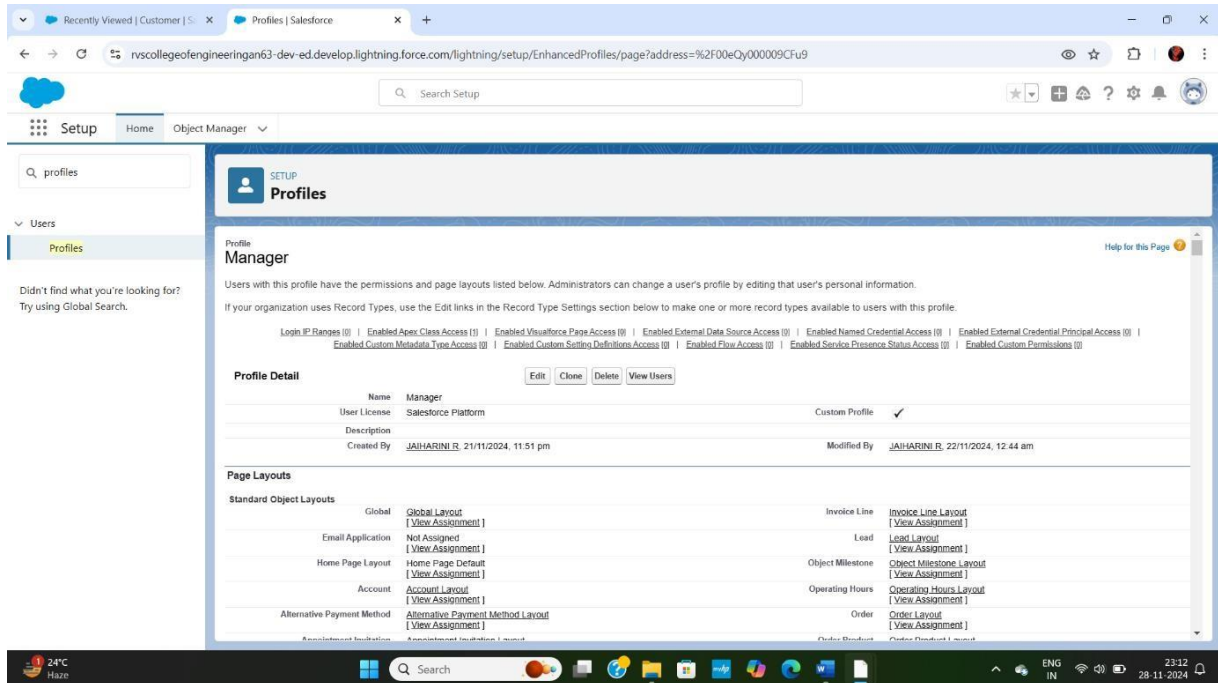
The screenshot shows the Salesforce Setup interface for the 'Customer' profile, specifically the permissions section. It includes a table for 'Communication Subscription Channel Types' and 'Labels'. Below this is a section for 'Custom Object Permissions' with a table for 'AppLog' and 'Property'. The table lists permissions for 'Read', 'Create', 'Edit', 'Delete', 'View All', and 'Modify All'. The 'Customer' profile has 'Read' and 'View All' permissions checked for both 'AppLog' and 'Property'.

Custom Object Permissions	Read	Create	Edit	Delete	View All	Modify All
AppLog	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customer	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Property	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

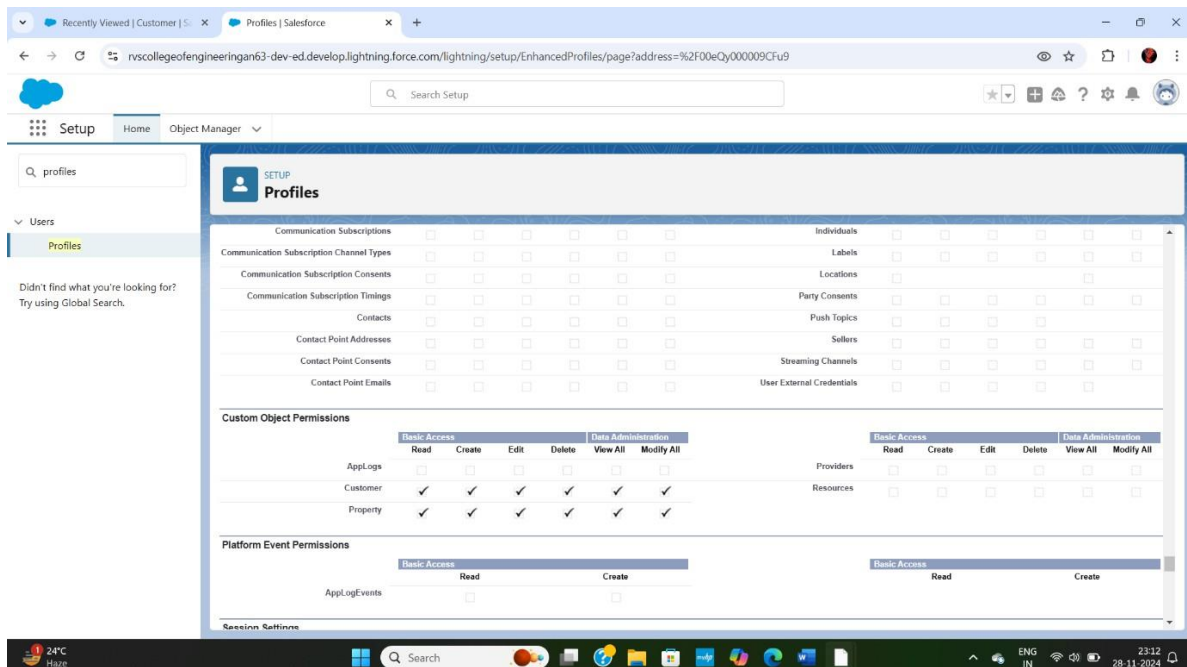


## Activity 2: Manager

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



The screenshot shows the Salesforce Setup interface for the 'Profiles' section. The profile 'Manager' is selected, and its details are displayed. The profile is a Custom Profile, created by JAIHARINI R. on 21/11/2024 at 11:51 pm, and modified by the same user on 22/11/2024 at 12:44 am. The profile is associated with the 'Salesforce Platform' user license. The 'Page Layouts' section shows the standard object layouts assigned to the profile, including Global, Email Application, Home Page Layout, Account, and Alternative Payment Method. The 'Profile Detail' section includes buttons for Edit, Clone, Delete, and View Users.



The screenshot shows the 'Manager' profile permissions in the Salesforce Setup interface. The permissions are organized into several sections: Communication Subscriptions, Custom Object Permissions, Platform Event Permissions, and Session Settings. The 'Custom Object Permissions' section includes a table for Basic Access and Data Administration permissions for various objects. The 'Platform Event Permissions' section includes a table for Basic Access permissions for AppLogEvents. The 'Session Settings' section includes a table for Session Settings.

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

Object	Read	Create
AppLogEvents	<input type="checkbox"/>	<input type="checkbox"/>

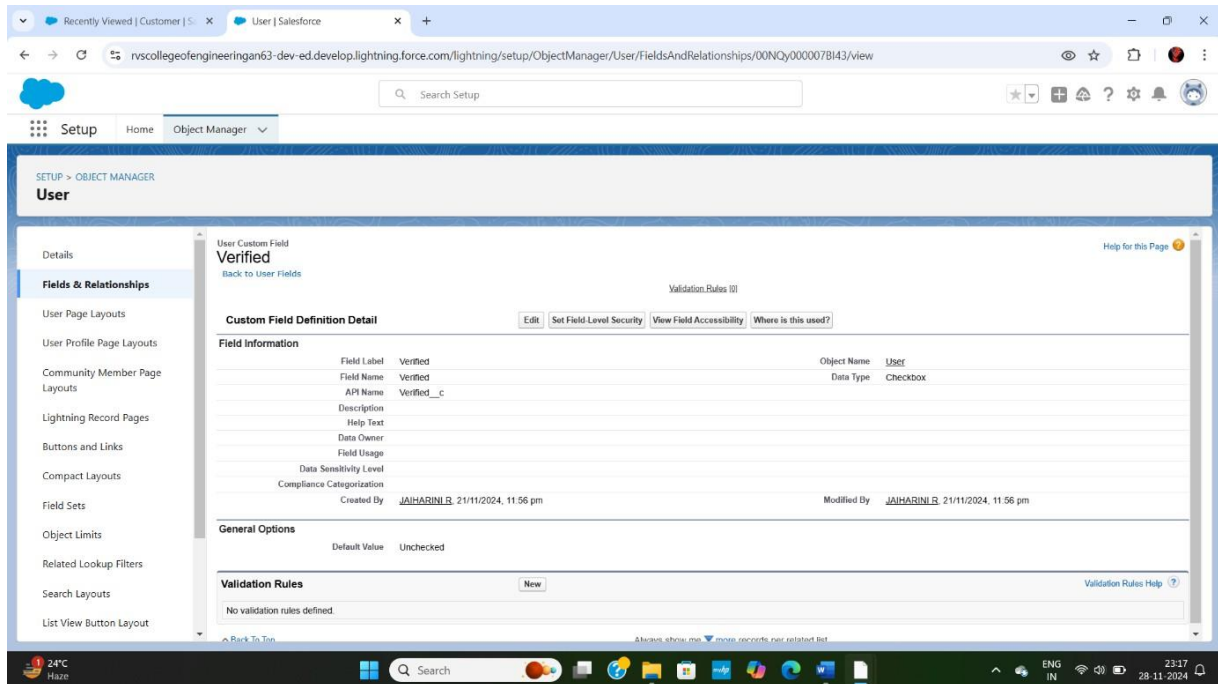


## STEP 7:

Create a Check Box field on user.

### Activity:

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



The screenshot shows the Salesforce Setup interface for configuring a custom field named "Verified" on the User object. The field is of type "Checkbox". The configuration page includes sections for Field Information, General Options, and Validation Rules.

Field Information			
Field Label	Verified	Object Name	User
Field Name	Verified	Data Type	Checkbox
API Name	Verified__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	JAIHARINI R.	Modified By	JAIHARINI R.
	21/11/2024, 11:56 pm		21/11/2024, 11:56 pm

**General Options**

Default Value	Unchecked
---------------	-----------

**Validation Rules**

No validation rules defined.

## 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

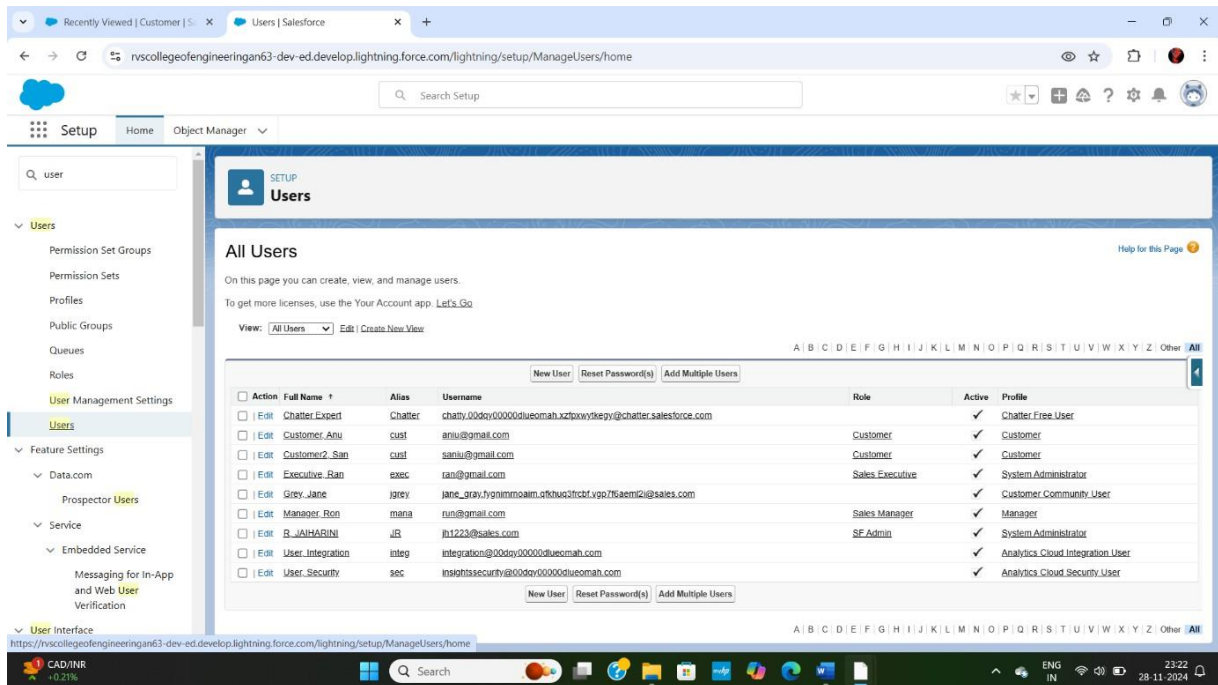
Go to Setup ->Administration ->Users ->New User;  
Last Name - Manager; Role - Sales Manager; License - Salesforce Platform; Profile - Manager and Save.

### 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.



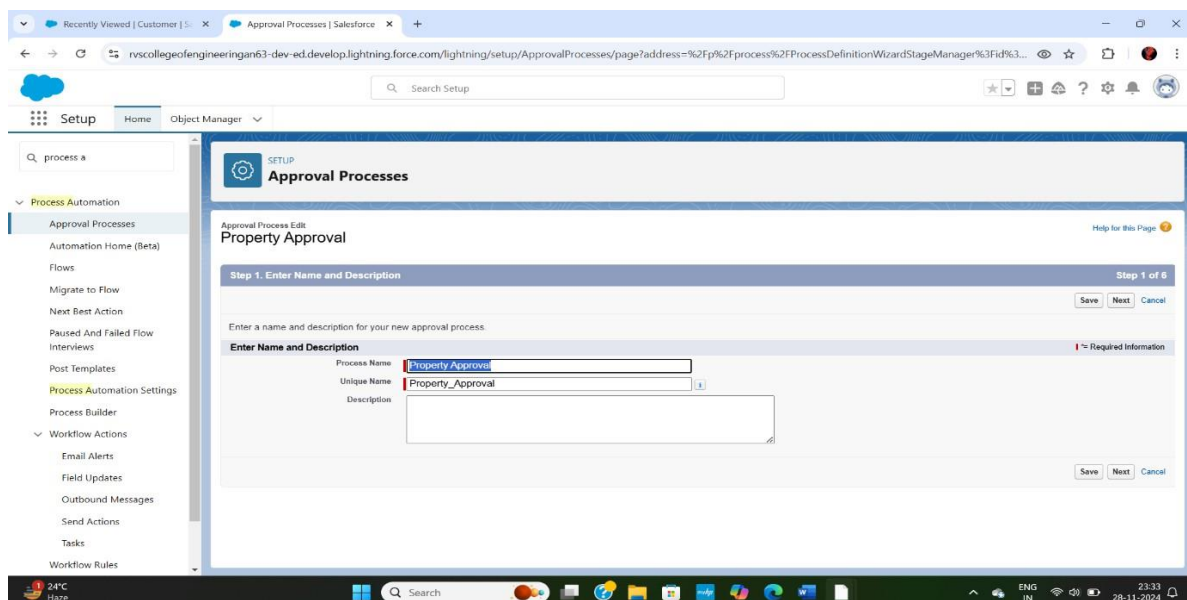
The screenshot shows the Salesforce Setup interface for the 'Users' section. The left sidebar contains navigation options like 'Users', 'Permission Set Groups', 'Permission Sets', 'Profiles', 'Public Groups', 'Queues', 'Roles', 'User Management Settings', 'Feature Settings', 'Data.com', 'Prospector Users', 'Service', 'Embedded Service', and 'User Interface'. The main content area is titled 'All Users' and includes a table of existing users. The table has columns for Action, Full Name, Alias, Username, Role, Active, and Profile. The table lists several users, including 'Chatter Expert', 'Customer\_Anu', 'Customer2\_San', 'Executive\_Ran', 'Grey\_Jane', 'Manager\_Ron', 'B\_JAHARINI', 'User\_Integration', and 'User\_Security'. The 'Active' column shows checkboxes for each user, and the 'Profile' column lists the assigned profiles. The bottom of the screen shows a Windows taskbar with the date and time as 23:22 on 28-11-2024.

## STEP 9:

Create an Approval Process for Property Object.

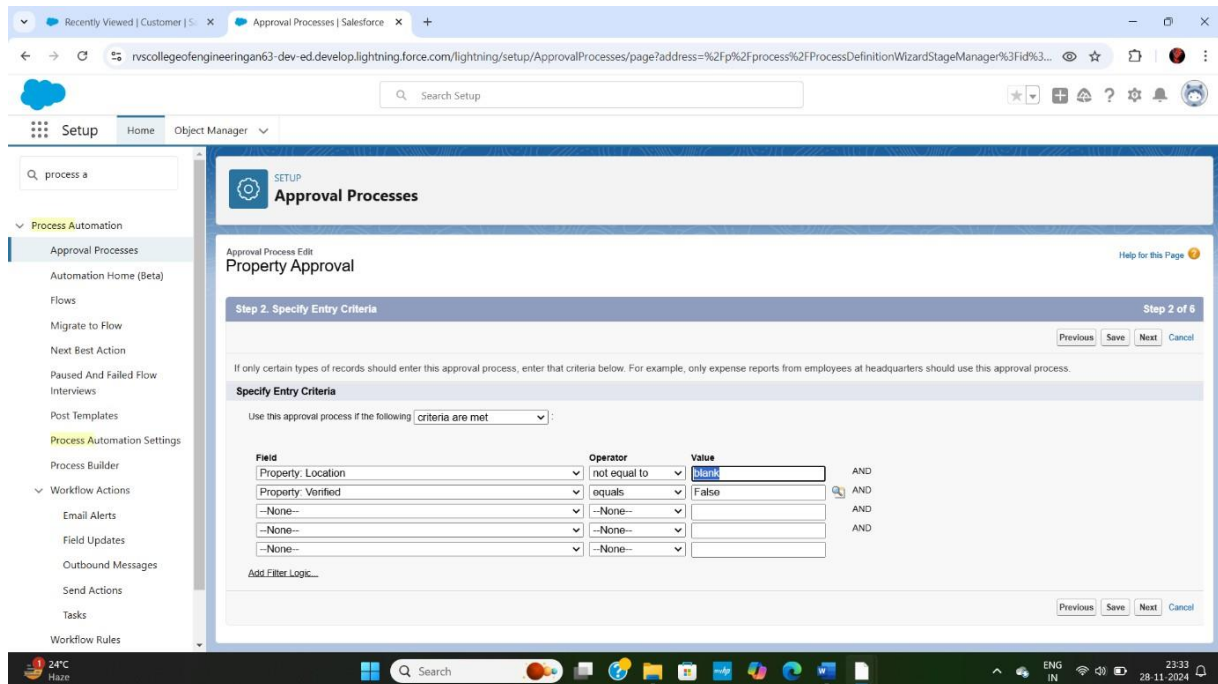
### Activity:

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



The screenshot shows the Salesforce Setup interface for the 'Approval Processes' section. The left sidebar contains navigation options like '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', 'Field Updates', 'Outbound Messages', 'Send Actions', 'Tasks', and 'Workflow Rules'. The main content area is titled 'Approval Processes' and shows the 'Property Approval' process. The 'Enter Name and Description' section includes fields for 'Process Name' (Property Approval), 'Unique Name' (Property\_Approval), and 'Description'. The 'Save' button is visible at the bottom right of the form. The bottom of the screen shows a Windows taskbar with the date and time as 23:33 on 28-11-2024.

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



Recently Viewed | Customer | Approval Processes | Salesforce

Search Setup

Setup Home Object Manager

process a

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
- Field Updates
- Outbound Messages
- Send Actions
- Tasks
- Workflow Rules

Approval Processes

Approval Process Edit

Property Approval

Step 2 of 6

Specify Entry Criteria

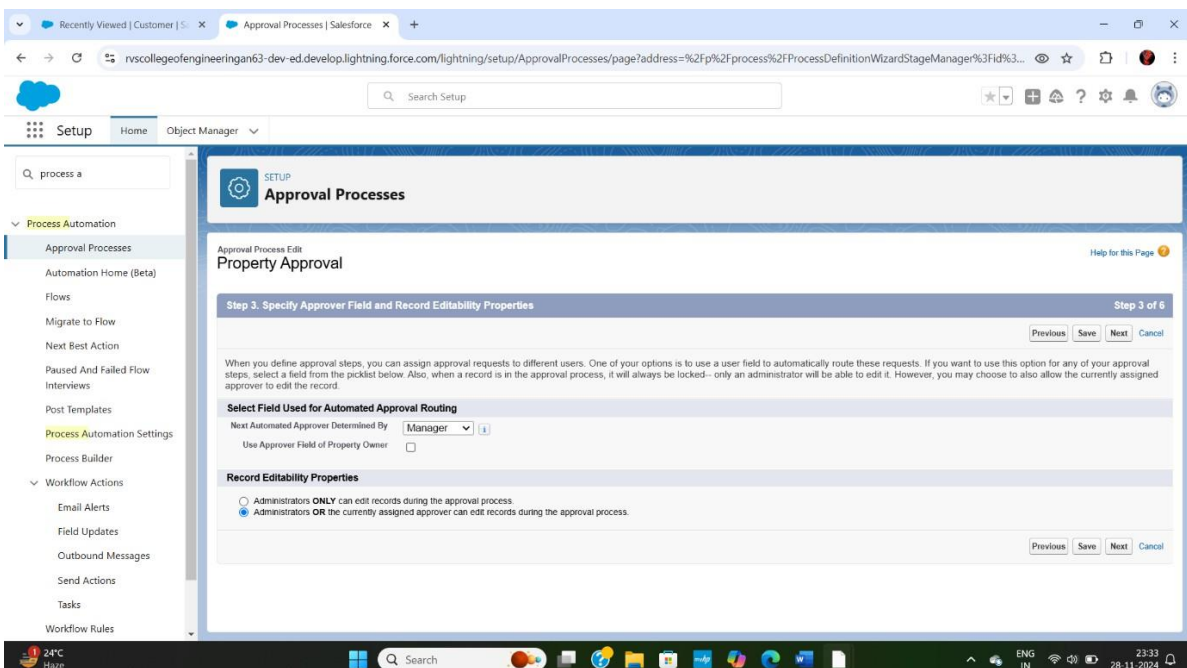
Use this approval process if the following criteria are met

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

Add Filter Logic...

Previous Save Next Cancel

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.



Recently Viewed | Customer | Approval Processes | Salesforce

Search Setup

Setup Home Object Manager

process a

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
- Field Updates
- Outbound Messages
- Send Actions
- Tasks
- Workflow Rules

Approval Processes

Approval Process Edit

Property Approval

Step 3 of 6

Specify Approver Field and Record Editability Properties

When you define approval steps, you can assign approval requests to different users. One of your options is to use a user field to automatically route these requests. If you want to use this option for any of your approval steps, select a field from the picklist below. Also, when a record is in the approval process, it will always be locked-- only an administrator will be able to edit it. However, you may choose to also allow the currently assigned approver to edit the record.

Select Field Used for Automated Approval Routing

Next Automated Approver Determined By: Manager

Use Approver Field of Property Owner: ☐

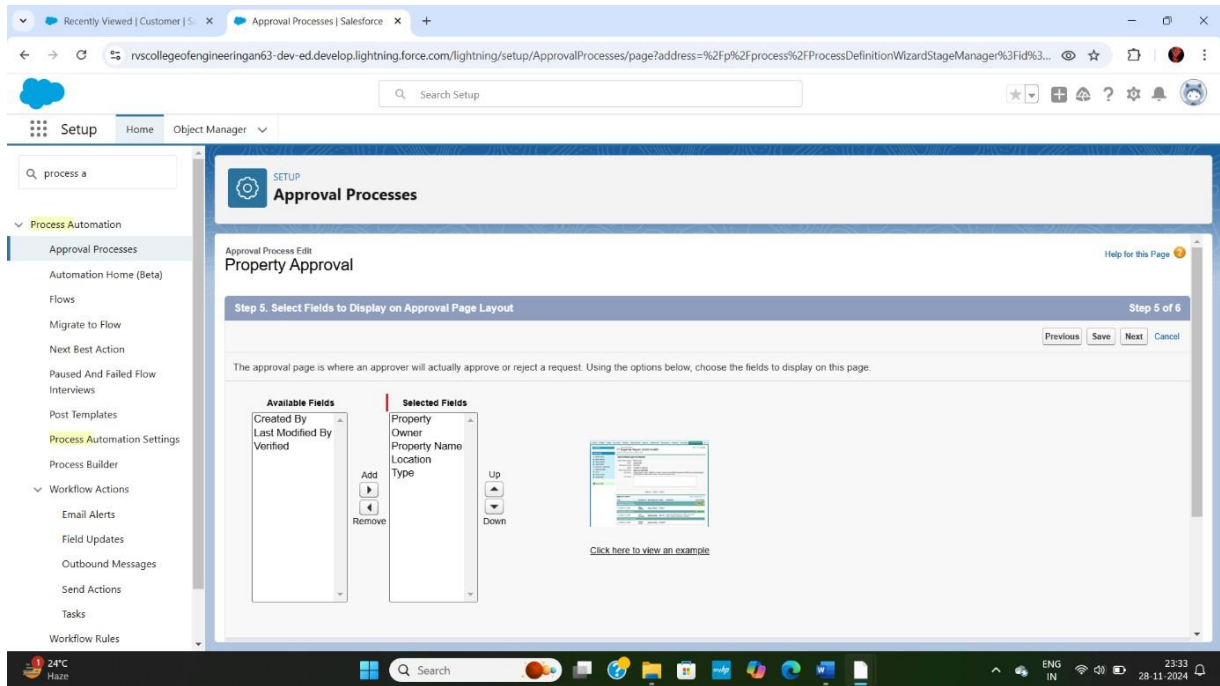
Record Editability Properties

☐ Administrators ONLY can edit records during the approval process.

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

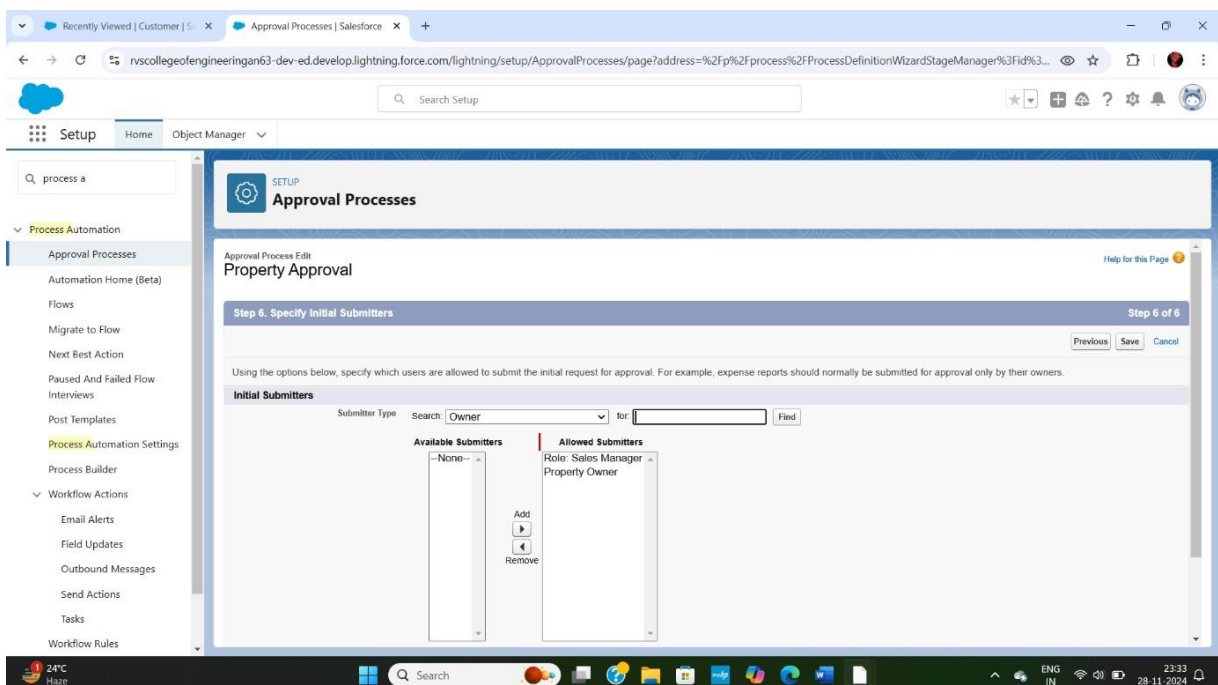
Previous Save Next Cancel

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



#### 5. Click Next and Select the initial Submitters >>

- a) Owner >> Property Owner
- b) 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".





Sir

Recently Viewed | Customer | Salesforce | Approval Processes | Setup

Search Setup

Setup Home Object Manager

process a

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

Field Updates

Outbound Messages

Send Actions

Tasks

Workflow Rules

## Approval Processes

Approval Step Edit: Executive Approval

Step 1. Enter Name and Description

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

Enter Name and Description

Approval Process Name: Property Approval

Name: Executive Approval

Unique Name: Executive\_Approval

Description:

Save Next Cancel

Recently Viewed | Customer | Salesforce | Approval Processes | Setup

Search Setup

Setup Home Object Manager

process a

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

Field Updates

Outbound Messages

Send Actions

Tasks

Workflow Rules

## Approval Processes

Approval Step Edit: Executive Approval

Step 2. Specify Step Criteria

Specify whether a record must meet certain criteria before entering this approval step. If these criteria are not met, the approval process can skip to the next step, if one exists. [Learn more](#)

Specify Step Criteria

☒ All records should enter this step.

☐ Enter this step if the following criteria are met:  else reject record:

Previous Save Next Cancel

Recently Viewed | Customer | Salesforce | Approval Processes | Setup

Search Setup

Setup Home Object Manager

process a

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

Field Updates

Outbound Messages

Send Actions

Tasks

Workflow Rules

## Approval Processes

Approval Step Edit: Executive Approval

Step 3. Select Assigned Approver

Specify the user who should approve records that enter this step. Optionally, choose whether the approver's delegate is also allowed to approve these requests.

Select Approver

☐ Let the submitter choose the approver manually.

☐ Automatically assign using the user field selected earlier. (Manager)

☐ Automatically assign to queue:

☒ Automatically assign to approver(s):

User:  Run Executive

Add Row Remove Row

When multiple approvers are selected:

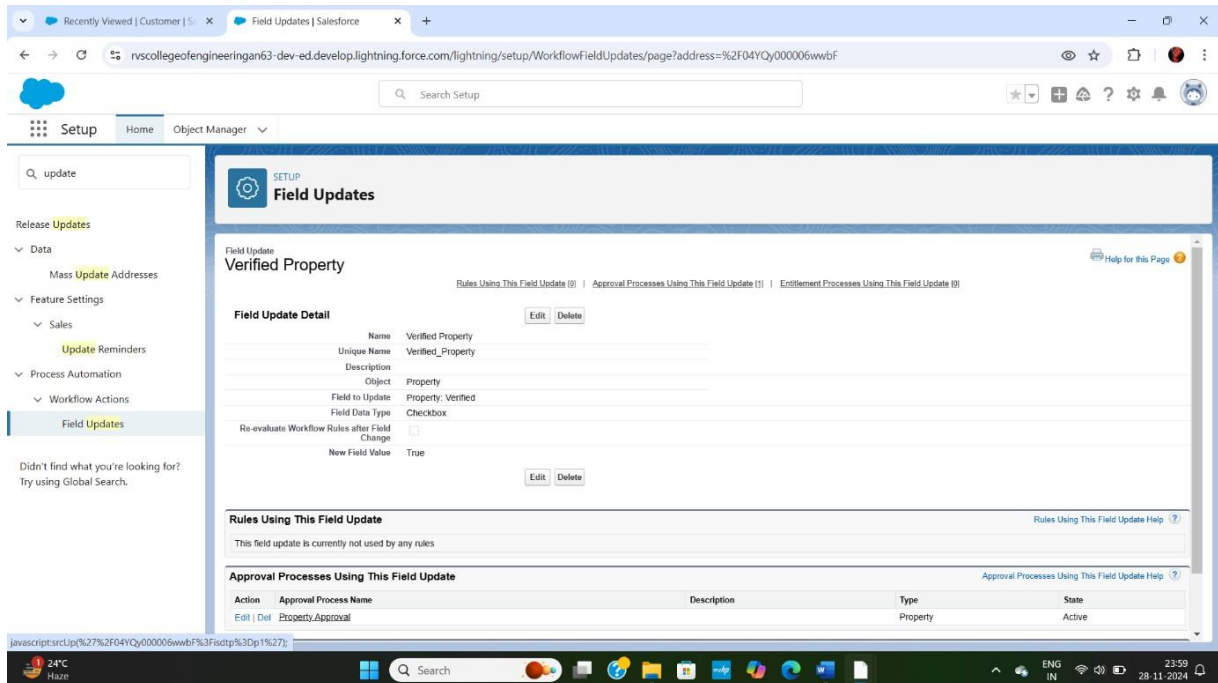
☒ Approve or reject based on the FIRST response.

☐ Require UNANIMOUS approval from all selected approvers.

☐ The approver's delegate may also approve this request.

Previous Save Cancel

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.

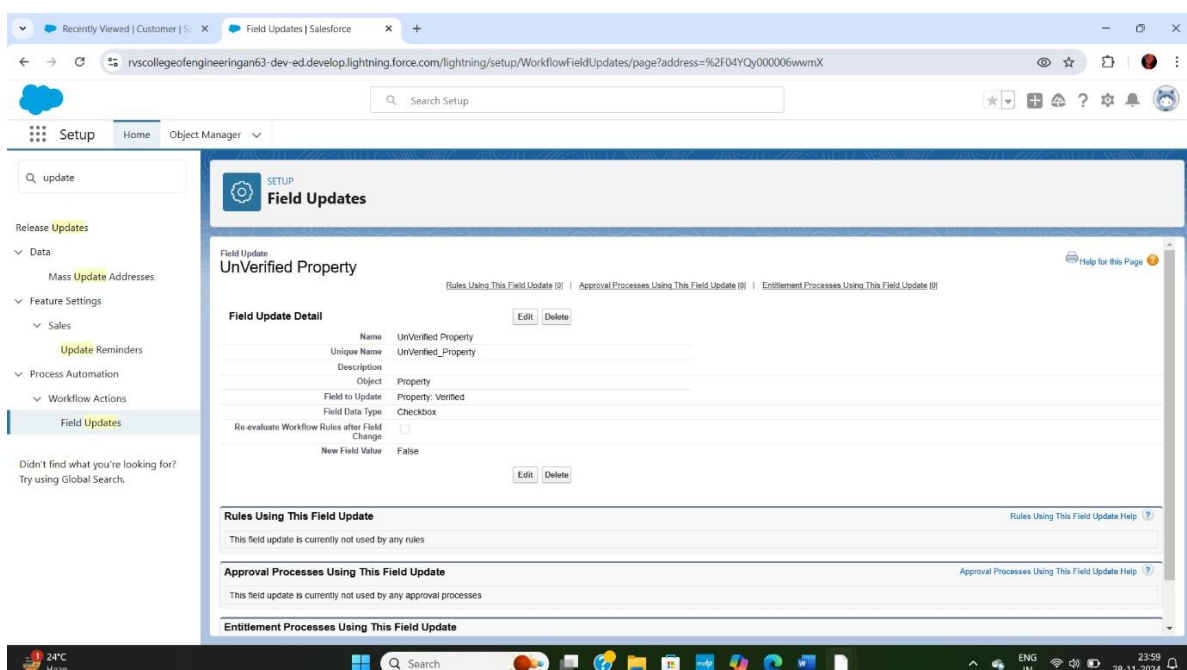


The screenshot shows the Salesforce Setup page for Field Updates. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled 'Field Updates' and shows a 'Verified Property' field update configuration. The 'Field Update Detail' section includes fields for Name, Unique Name, Description, Object, Field to Update, Field Data Type, Re-evaluate Workflow Rules after Field Change, and New Field Value. The 'Rules Using This Field Update' section shows a table with one rule. The 'Approval Processes Using This Field Update' section shows a table with one approval process.

Name	Unique Name	Description	Object	Field to Update	Field Data Type	Re-evaluate Workflow Rules after Field Change	New Field Value
Verified Property	Verified_Property		Property	Property: Verified	Checkbox	<input type="checkbox"/>	True

Action	Approval Process Name	Description	Type	State
Edit   Del	Property Approval		Property	Active

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 page for Field Updates. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled 'Field Updates' and shows an 'UnVerified Property' field update configuration. The 'Field Update Detail' section includes fields for Name, Unique Name, Description, Object, Field to Update, Field Data Type, Re-evaluate Workflow Rules after Field Change, and New Field Value. The 'Rules Using This Field Update' section shows a table with one rule. The 'Approval Processes Using This Field Update' section shows a table with one approval process.

Name	Unique Name	Description	Object	Field to Update	Field Data Type	Re-evaluate Workflow Rules after Field Change	New Field Value
UnVerified Property	UnVerified_Property		Property	Property: Verified	Checkbox	<input type="checkbox"/>	False

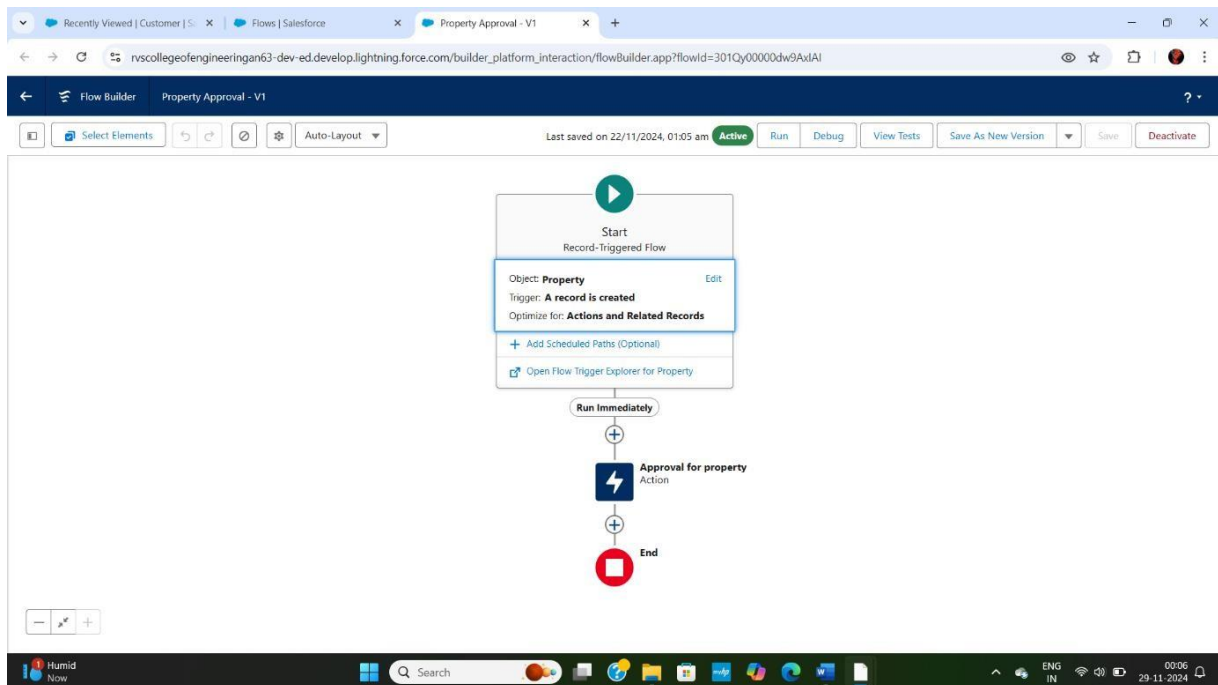
Action	Approval Process Name	Description	Type	State
Edit   Del	Property Approval		Property	Active

## 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:

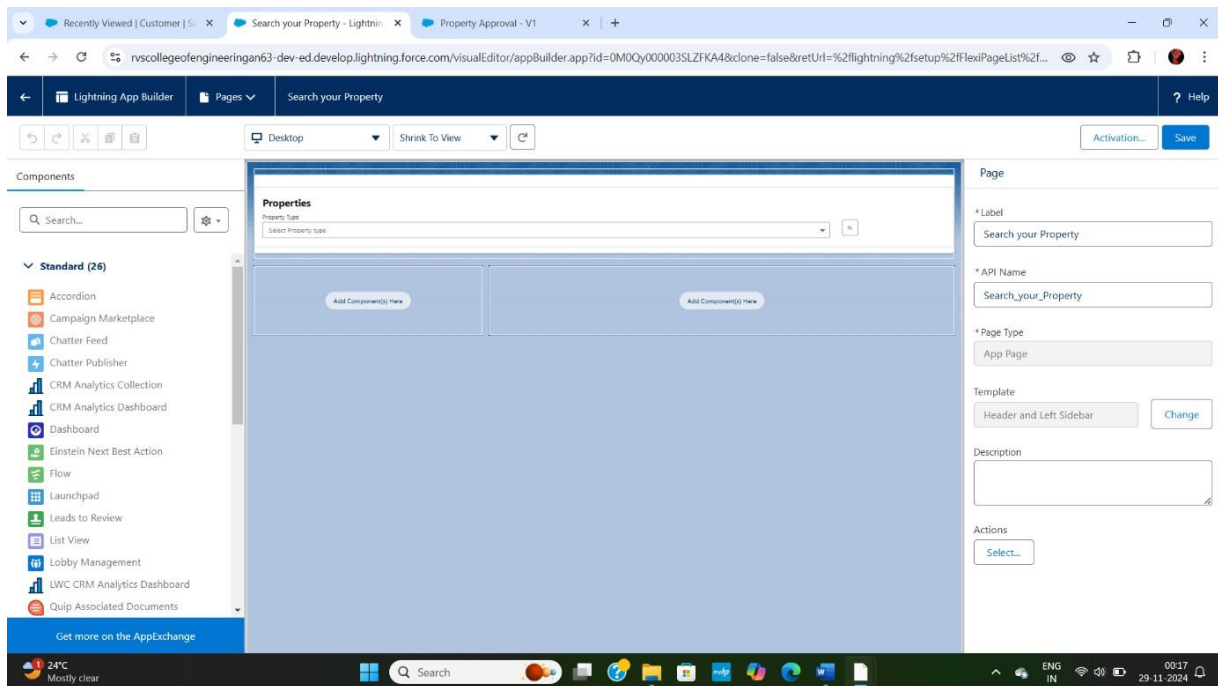
Create an App Page.

### 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.

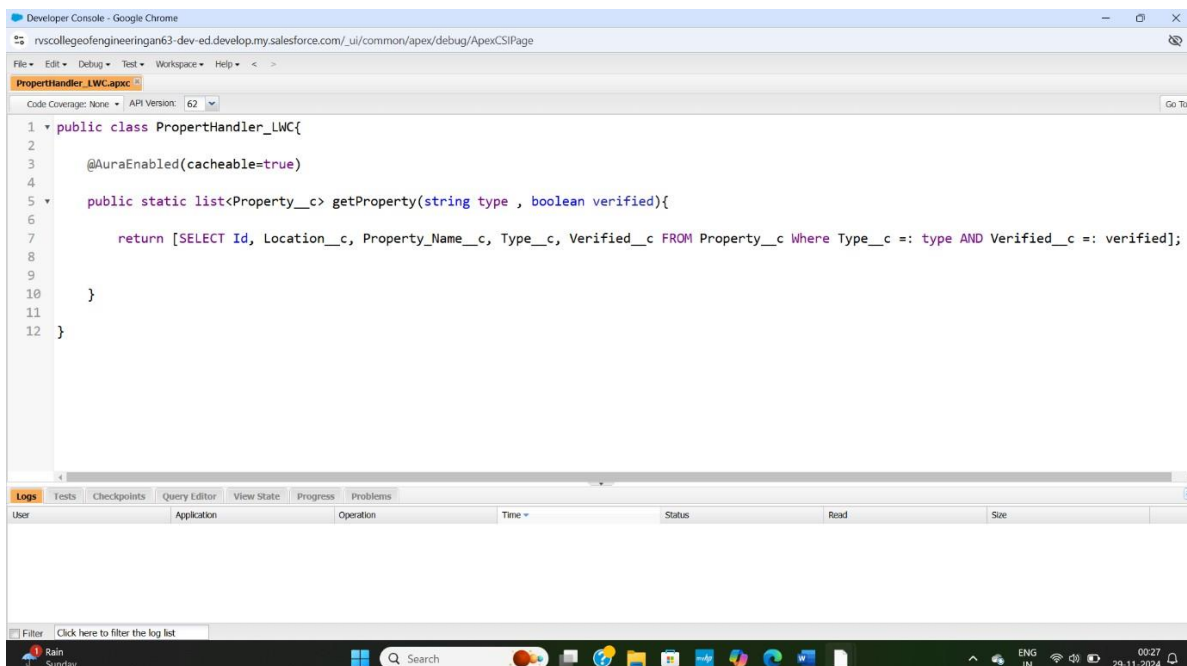


## STEP 12:

Create a LWC Component

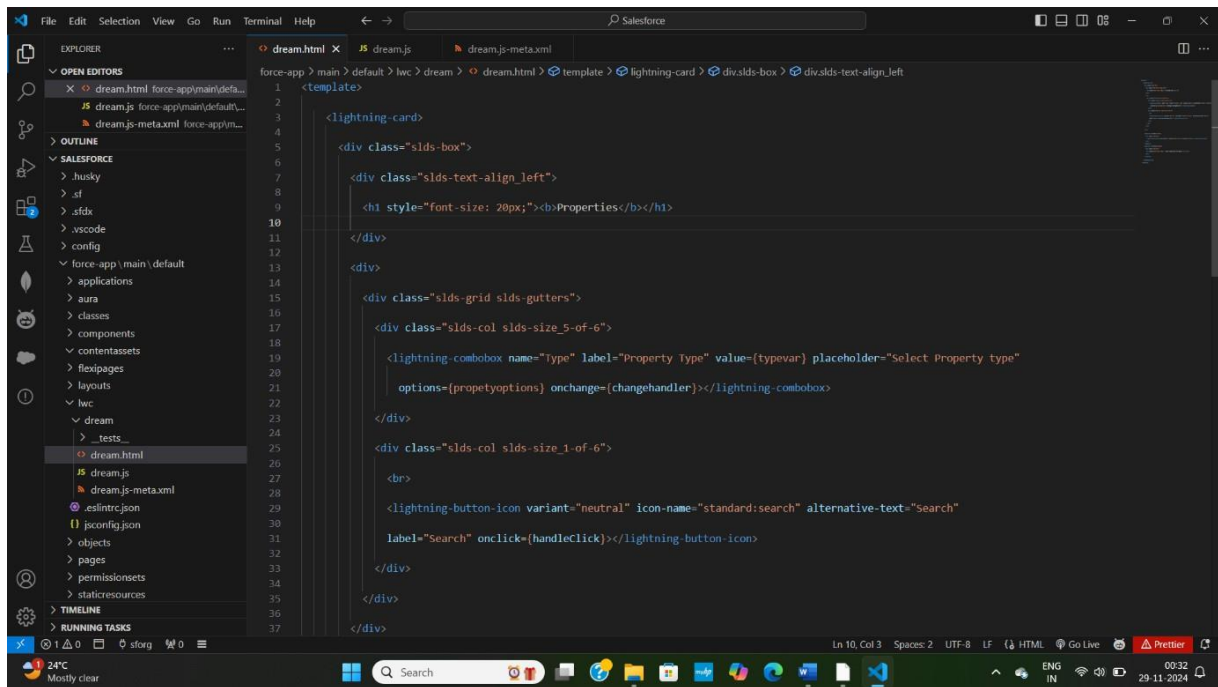
### Activity:

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



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:

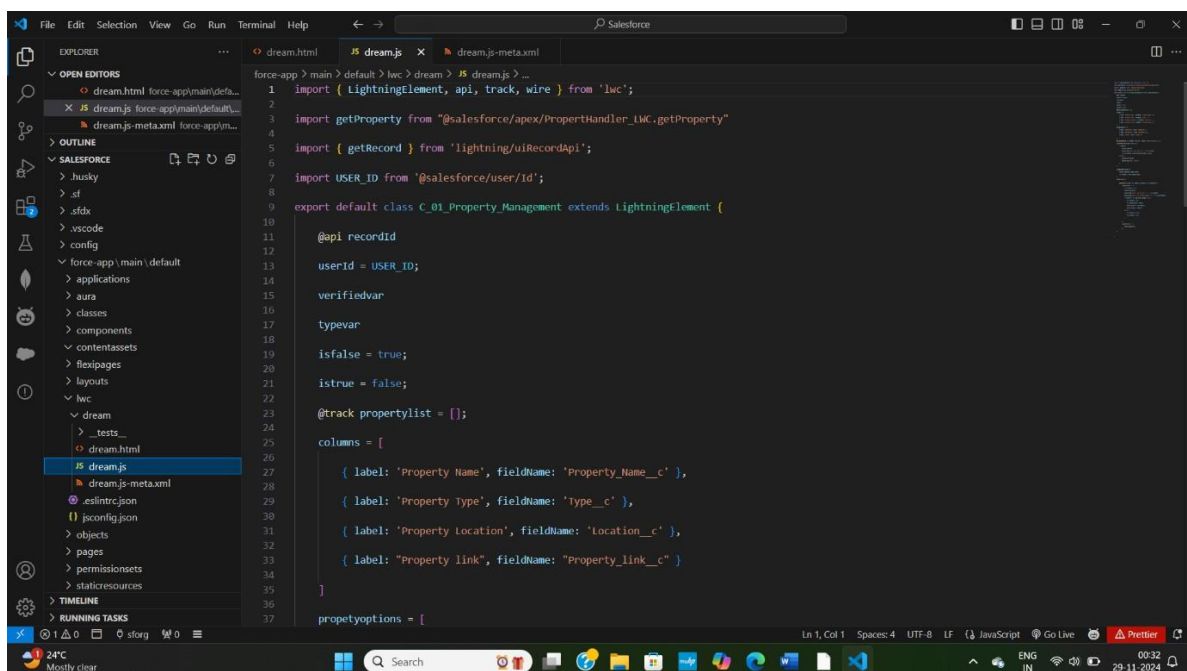


```

1 <template>
2
3 <lightning-card>
4
5 <div class="slds-box">
6
7 <div class="slds-text-align_left">
8
9 <h1 style="font-size: 20px;"><b>Properties</b></h1>
10
11 </div>
12
13 <div>
14
15 <div class="slds-grid slds-gutters">
16
17 <div class="slds-col slds-size_5-of-6">
18
19 <lightning-combobox name="Type" label="Property Type" value={typevar} placeholder="Select Property type"
20 options={propertyoptions} onchange={changehandler}></lightning-combobox>
21
22 </div>
23
24 <div class="slds-col slds-size_1-of-6">
25
26 <br>
27
28 <lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search"
29 label="Search" onclick={handleClick}></lightning-button-icon>
30
31 </div>
32
33 </div>
34
35 </div>
36
37 </div>

```

## 2. JS file:

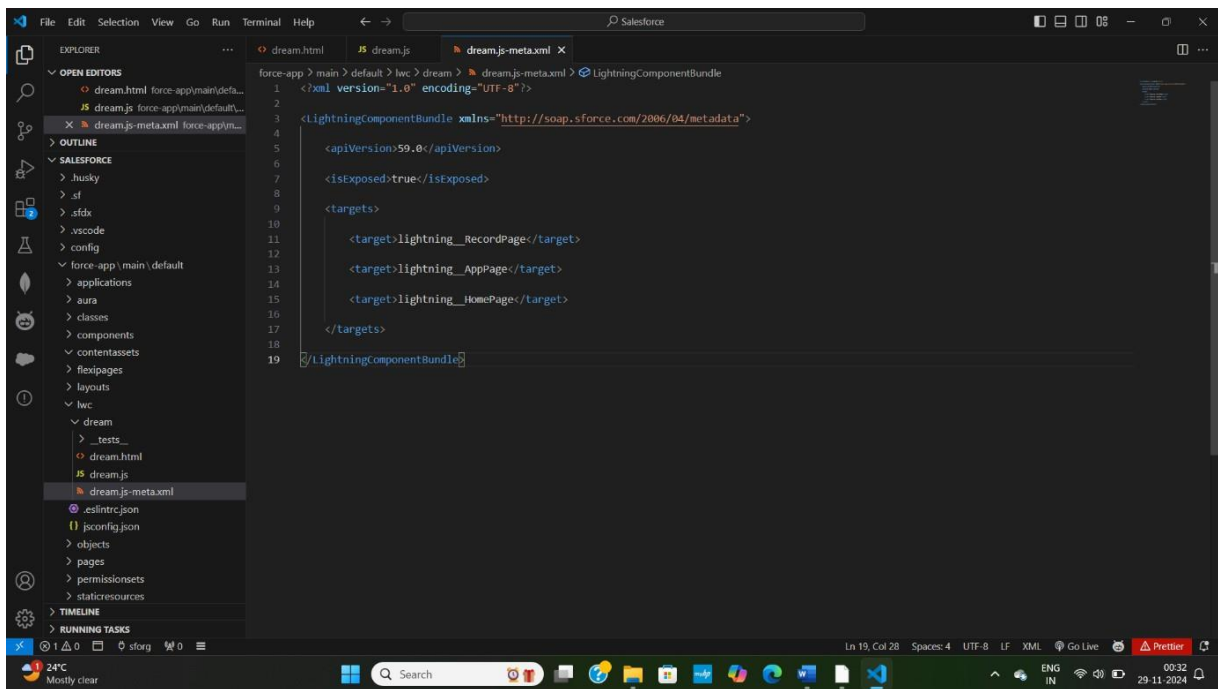


```

1 import { LightningElement, api, track, wire } from 'lwc';
2
3 import { getRecord } from '@salesforce/apex/PropertyHandler_LWC.getRecord';
4
5 import { getRecord } from 'lightning/uiRecordApi';
6
7 import USER_ID from '@salesforce/user/id';
8
9 export default class C_01_Property_Management extends LightningElement {
10
11 @api recordId;
12
13 userId = USER_ID;
14
15 verifiedvar;
16
17 typevar;
18
19 isfalse = true;
20
21 istrue = false;
22
23 @track propertylist = [];
24
25 columns = [
26
27 { label: 'Property Name', fieldName: 'Property_Name__c' },
28
29 { label: 'Property Type', fieldName: 'Type__c' },
30
31 { label: 'Property Location', fieldName: 'Location__c' },
32
33 { label: 'Property link', fieldName: 'Property_Link__c' }
34
35 ]
36
37 propertyoptions = [

```

### 3. Metafile



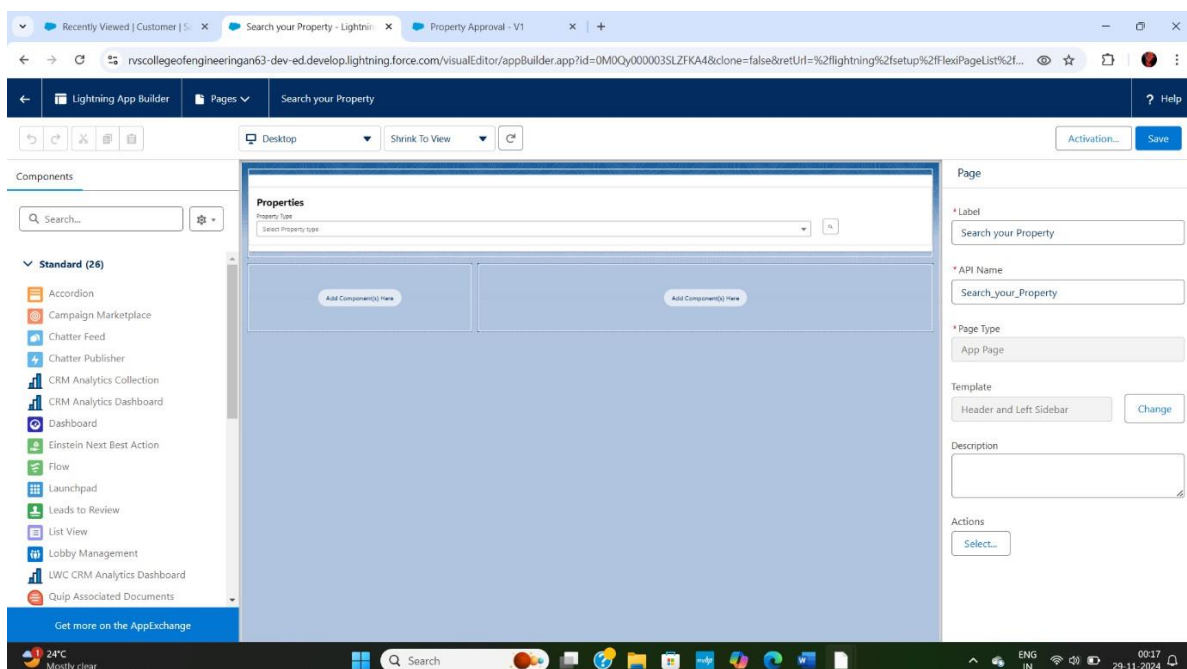
After Saving all the three Codes, Right Click and deploy this component to the org.

#### 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.

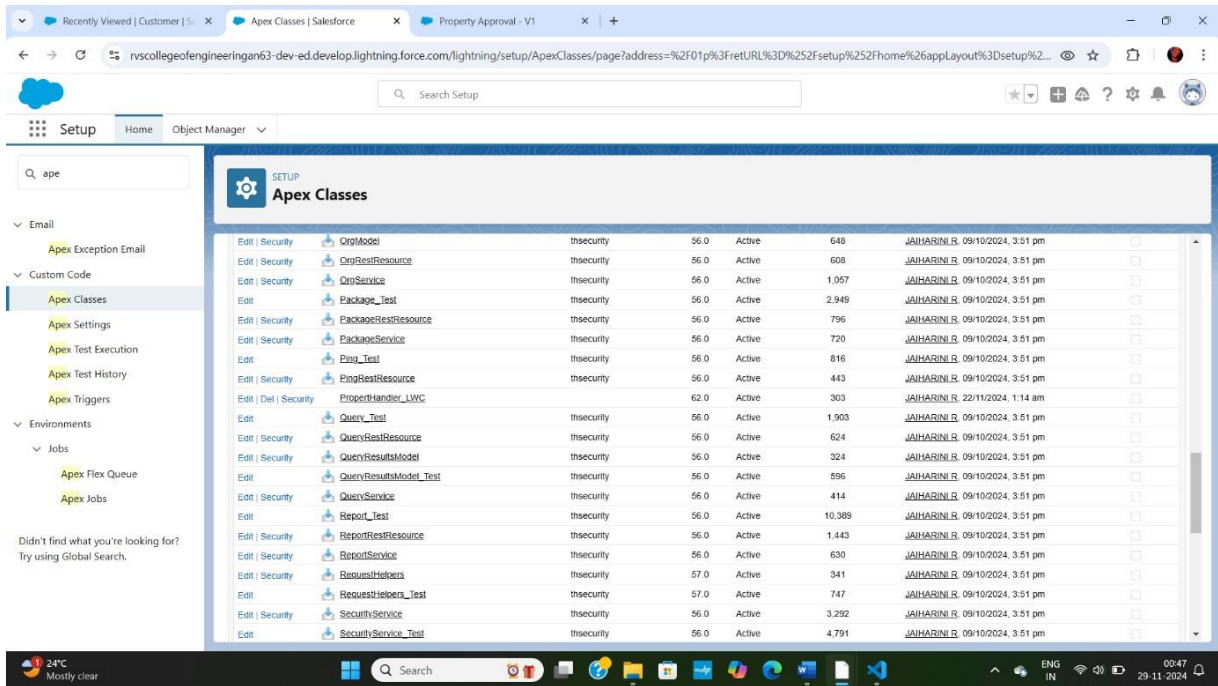


## STEP 14:

Give Access of Apex Classes to Profiles.

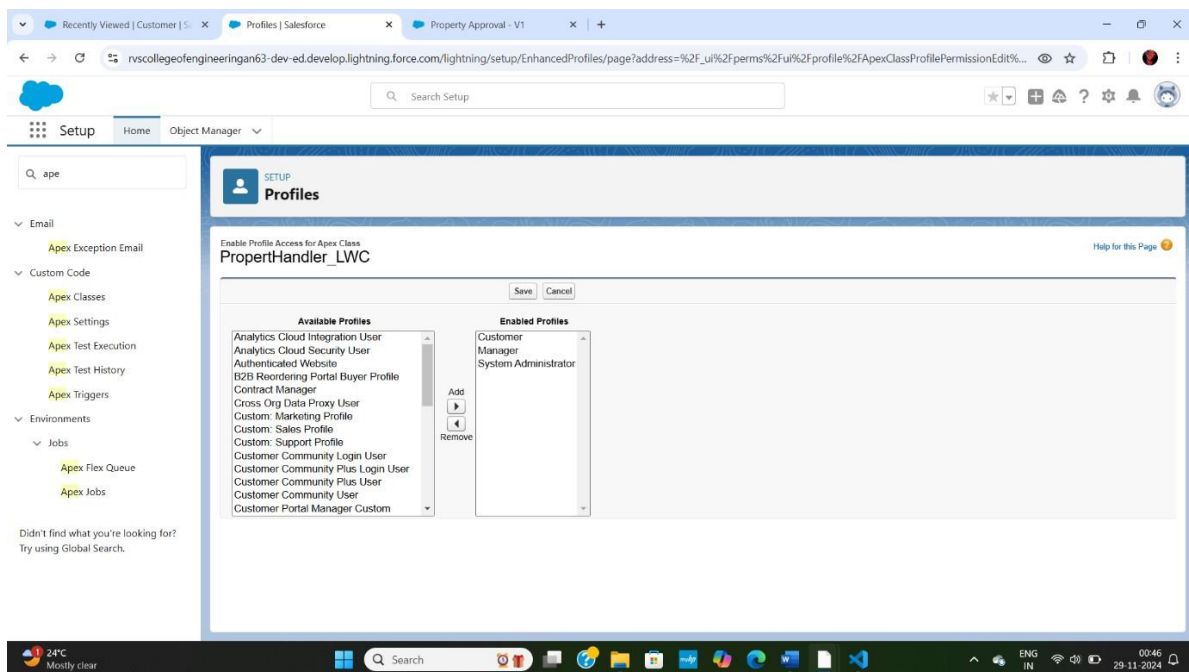
### Activity:

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 page for Apex Classes. The left sidebar contains a search bar and a navigation menu with categories like Email, Custom Code, Environments, and Jobs. The main content area displays a table of Apex Classes. The table has columns for Edit, Security, Name, Namespace, Version, Status, Size, and Last Modified By/Date. The class 'PropertyHandler\_LWC' is highlighted in the table.

Edit	Security	Name	Namespace	Version	Status	Size	Last Modified By/Date
Edit	Security	OrgModel	thsecurity	56.0	Active	648	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	OrgRestResource	thsecurity	56.0	Active	608	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	OrgService	thsecurity	56.0	Active	1,057	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	Package_Test	thsecurity	56.0	Active	2,949	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	PackageRestResource	thsecurity	56.0	Active	796	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	PackageService	thsecurity	56.0	Active	720	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	Play_Test	thsecurity	56.0	Active	816	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	PlayRestResource	thsecurity	56.0	Active	443	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Del   Security	PropertyHandler_LWC	thsecurity	62.0	Active	303	JAIHARINI_R 22/11/2024, 1:14 am
Edit	Security	Query_Test	thsecurity	56.0	Active	1,903	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	QueryRestResource	thsecurity	56.0	Active	624	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	QueryResultsModel	thsecurity	56.0	Active	324	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	QueryResultsModel_Test	thsecurity	56.0	Active	596	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	QueryService	thsecurity	56.0	Active	414	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	Report_Test	thsecurity	56.0	Active	10,389	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	ReportRestResource	thsecurity	56.0	Active	1,443	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	ReportService	thsecurity	56.0	Active	630	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	RequestHelpers	thsecurity	57.0	Active	341	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	RequestHelpers_Test	thsecurity	57.0	Active	747	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	SecurityService	thsecurity	56.0	Active	3,292	JAIHARINI_R 09/10/2024, 3:51 pm
Edit	Security	SecurityService_Test	thsecurity	56.0	Active	4,791	JAIHARINI_R 09/10/2024, 3:51 pm



The screenshot shows the Salesforce Setup page for Profiles. The left sidebar contains a search bar and a navigation menu. The main content area displays the 'Enable Profile Access for Apex Class' page for 'PropertyHandler\_LWC'. The page has a 'Save' button and a 'Cancel' button. Below the buttons, there are two lists: 'Available Profiles' and 'Enabled Profiles'. The 'Available Profiles' list includes various roles like 'Analytics Cloud Integration User', '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', and 'Customer Portal Manager Custom'. The 'Enabled Profiles' list currently contains 'Customer Manager' and 'System Administrator'.

Available Profiles	Enabled Profiles
Analytics Cloud Integration User	Customer Manager
Analytics Cloud Security User	System Administrator
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	
Customer Portal Manager Custom	

## 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 behavior.

### 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 behavior.
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.

## **8.Result:**

Thus the CRM (Customer Relationship Management) application using Salesforce to manage clients and their property-related information is designed and developed. The application will enable the organization to streamline client interactions, property management, and related services.