

## **NAAN MUDHALVAN- SALESFORCE REPORT**

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

PROJECT CREATED BY:

Priyabhuvani S - (422422104019)

Vinitha M - (422422104020)

Jagan S - (422422104021)

Sathya R - (422422104022)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UNIVERSITY COLLEGE OF ENGINEERING TINDIVANAM**



**COLLEGE CODE-4224**

**ANNA UNIVERSITY,CHENNAI-600025**

**TABLE OF CONTENTS:**

<b>1.Project overview -----</b>	<b>3</b>
<b>2.Objectives-----</b>	<b>3</b>
<b>3.Key features and concept-----</b>	<b>4</b>
<b>4.Detailed steps to solution design-----</b>	<b>5</b>
<b>5.Testing and validation -----</b>	<b>5</b>
<b>6.Key scenarios addressed by salesforce in this Implementation process-----</b>	<b>6</b>
<b>7.Step by Step Process Explanation-----</b>	<b>6</b>
<b>8.Conclusion-----</b>	<b>27</b>

---

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

---

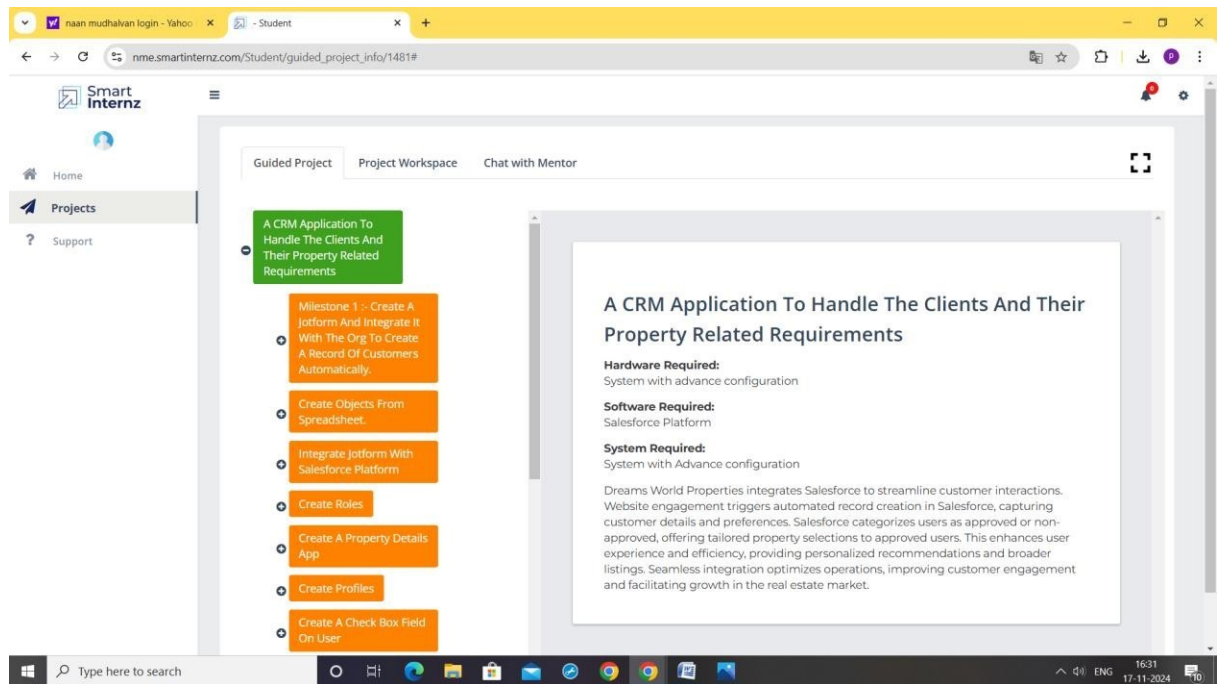
### **1. PROJECT OVERVIEW:**

This project focuses on developing a CRM application using Salesforce, enhanced with JotForm integration, to manage clients and their property-related requirements. The solution streamlines client interactions, centralizes property data, and automates workflows. JotForm is utilized for collecting client data through customizable forms, which integrate seamlessly into Salesforce to ensure accurate and efficient data processing.

### **2. PROJECT OBJECTIVES:**

- Centralized Data Management.
- Workflow Automation.
- Seamless Data Collection via JotForm.
- Enhanced Client Interaction.
- Scalability and Security.
- Enhance customer satisfaction through efficient data collection and communication.

**FIGURE1:**



### **3.KEY FEATURES AND CONCEPTS:**

- **Apex Triggers:** Automated logic to synchronize account data, ensuring accuracy in client details and property preferences.
- **Custom Objects and Fields:** Designed for client and property data storage and management.
- **JotForm Integration:** Enabled efficient data collection through user-friendly forms, directly populating Salesforce records.
- **Process Builder and Workflows:** Automated property matching and email notifications based on data received from JotForm.
- **Process Builder:** Designed workflows to match properties to client requirements automatically.
- **Reports and Dashboards:** Provided actionable insights on property trends and client activities.
- **Role-Based Access Controls:** Ensured data security and privacy.

## 4.DETAILED STEPS TO SOLUTION DESIGN:

- **Custom Object Creation:** Created objects for clients, properties, and transactions with appropriate fields.
- **Integration with JotForm:** Linked JotForm to Salesforce for automatic client data import.
- **Workflow Automation:** Developed workflows using Process Builder to match properties with client requirements and send alerts.
- **UI Customization:** Designed user-friendly layouts and Lightning components for managing data efficiently.
- **Security Implementation:** Applied profiles, roles, and sharing rules to protect sensitive client and property data.
- **Testing:** Conducted unit, system, and user acceptance testing to validate workflows and integrations.

## 5.TESTING AND VALIDATION:

- **Unit Testing:** Verified individual components like Apex triggers and workflows.
- **System Testing:** Ensured seamless integration of JotForm and Salesforce workflows.
- **User Acceptance Testing (UAT):** Gathered feedback from test users to refine the system.
- **Performance Testing:** Ensured scalability under increased client and property data loads.

## **6.KEY SCENARIOS ADDRESSED BY JOTFORM AND SALESFORCE INTEGRATION:**

- **Efficient Client Management:** Centralized client profiles with detailed property preferences and interaction.
- **Automated Property Matching:** Leveraged automation to match client requirements with suitable properties instantly.
- **Real-Time Updates:** Enabled clients to receive updates on property availability and transaction status.
- **Streamlined Data Collection:** Used JotForm for error-free and efficient data input directly into Salesforce.
- **Insights and Analytics:** Dashboards offered key metrics like property availability trends and client activity.

## **7.STEP BY STEP PROCESS EXPLANATION:**

### **STEP 1: CREATE A JOTFORM AND INTEGRATE IT WITH A ORG TO CREATE A RECORD OF CUSTOMERS AUTOMATICALLY**

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

FIGURE2:

The screenshot displays the Jotform Form Builder interface. The browser address bar shows the URL [jotform.com/build/243153828082053](https://jotform.com/build/243153828082053). The form titled "Form" is centered on a light purple background. It contains the following fields and sections:

- Name:** Two input fields for "First Name" and "Last Name".
- Email:** A single input field.
- Phone Number:** An input field with a note: "Please enter a valid phone number".
- Which type of Property are you looking for?:** Three radio button options: "RESIDENTIAL", "COMMERCIAL", and "RENTAL".
- Budget Amount:** An input field with a note: "e.g., 25".
- Address:** A series of input fields for "Street Address", "Apt, Suite, etc.", "City", "State (Province)", and "Postal / Zip Code".
- Submit:** A green button at the bottom of the form.

Below the form, there is a yellow banner with the text "Jotform is the leading online form builder platform" and a blue banner with the Jotform logo and the text "Get a free trial for 14 days".

FIGURE3:

The screenshot displays the Jotform Form Builder interface, specifically the "Integrations" settings page for a form titled "Form". The browser address bar shows the URL [jotform.com/build/243153828082053/settings/integrations](https://jotform.com/build/243153828082053/settings/integrations). The page has a dark sidebar on the left with the following menu items:

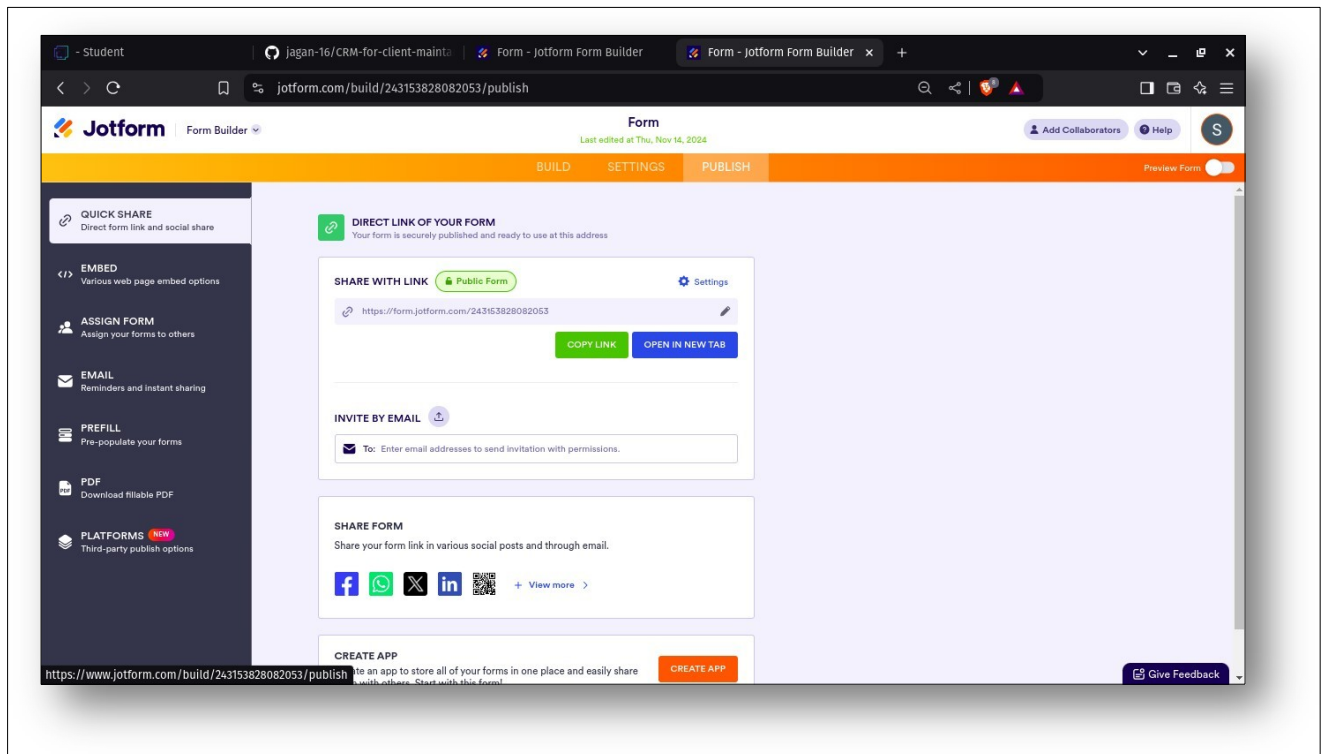
- FORM SETTINGS: Customize form status and properties
- EMAILS: Send autoresponders and notifications
- CONDITIONS: Set up conditional logic
- THANK YOU PAGE: Show page after submission
- INTEGRATIONS: Connect your form to other apps (highlighted)
- WORKFLOWS/ Formerly Approvals: Turn your form into a workflow
- JOTFORM SIGN: Power your forms with Jotform Sign
- MOBILE NOTIFICATIONS: Customize mobile app notifications

The main content area is titled "SALESFORCE" and includes the instruction "Send new leads, contacts, or accounts to your sales CRM". It features a "Select a Salesforce Object" dropdown menu with "Customer" selected. Below this, the "Create a record" section is titled "Send data from form fields to matched Salesforce Fields". It shows a mapping of form fields to Salesforce fields:

Object Fields	Form
Name	Name
City	Address - City
Budget Amount	Budget Amount
Property Type	Which type of Property are you lookin...
Street Address	Address - Street Address
Verified	Submission ID
Email	Email
State	Address - State

At the bottom right, there is a "Give Feedback" button.

**FIGURE4:**



## **STEP2:CREATE OBJECTS FROM SPREADSHEET.**

Directly Creating Objects from Spreadsheet in Salesforce.

### **Create Customer object**

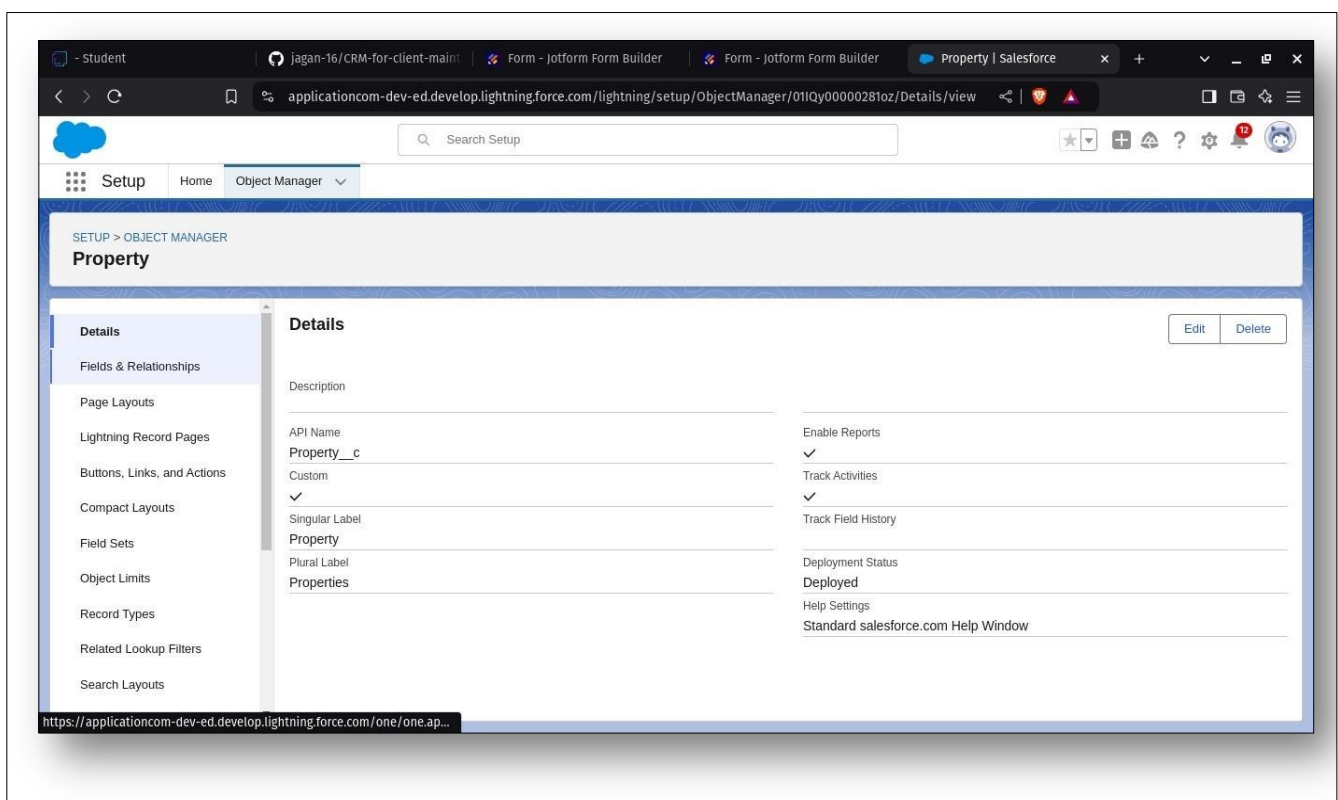
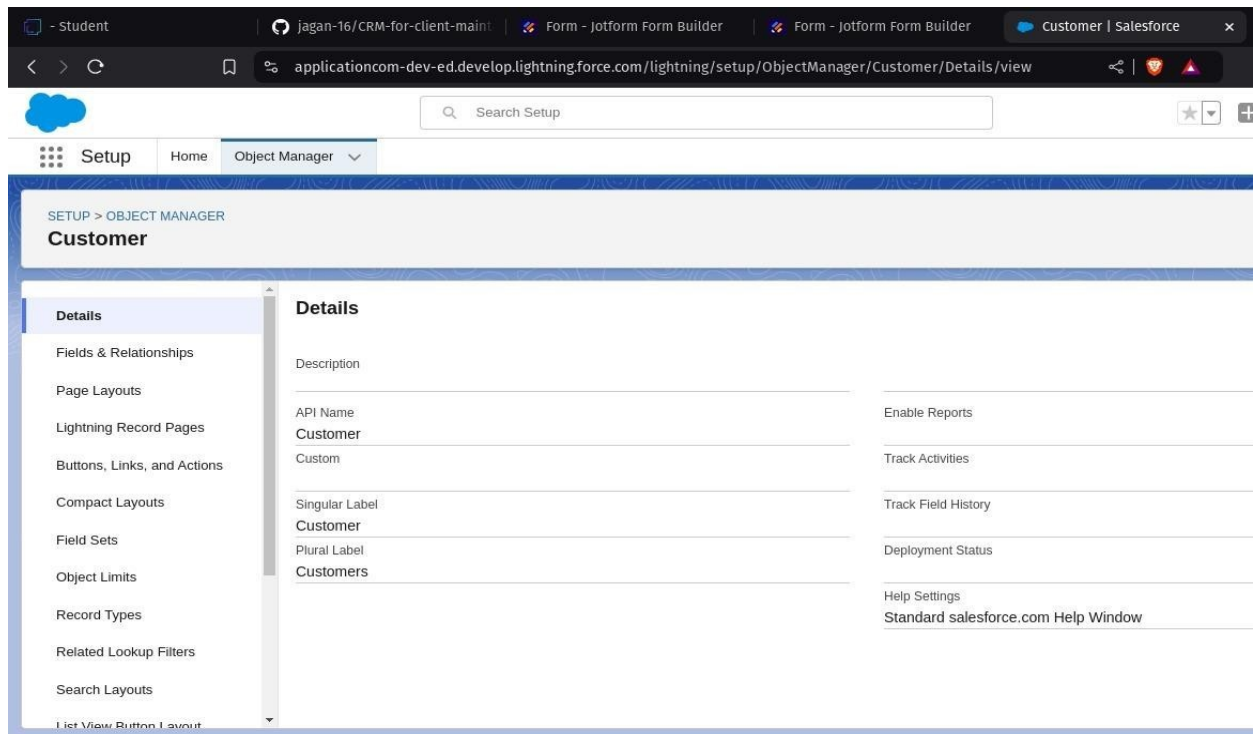
- Go to your object manager and and click on create object from spreadsheet.
- Click on the link to get the spreadsheet
- customer
- After downloading, upload the file, map the fields and upload to create an object.

### **Create Property object**

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



FIGURE5:



### STEP 3: INTEGRATE JOTFORM WITH SALESFORCE PLATFORM

In this Milestone we are going to integrate jotform with Salesforce

FIGURE6:

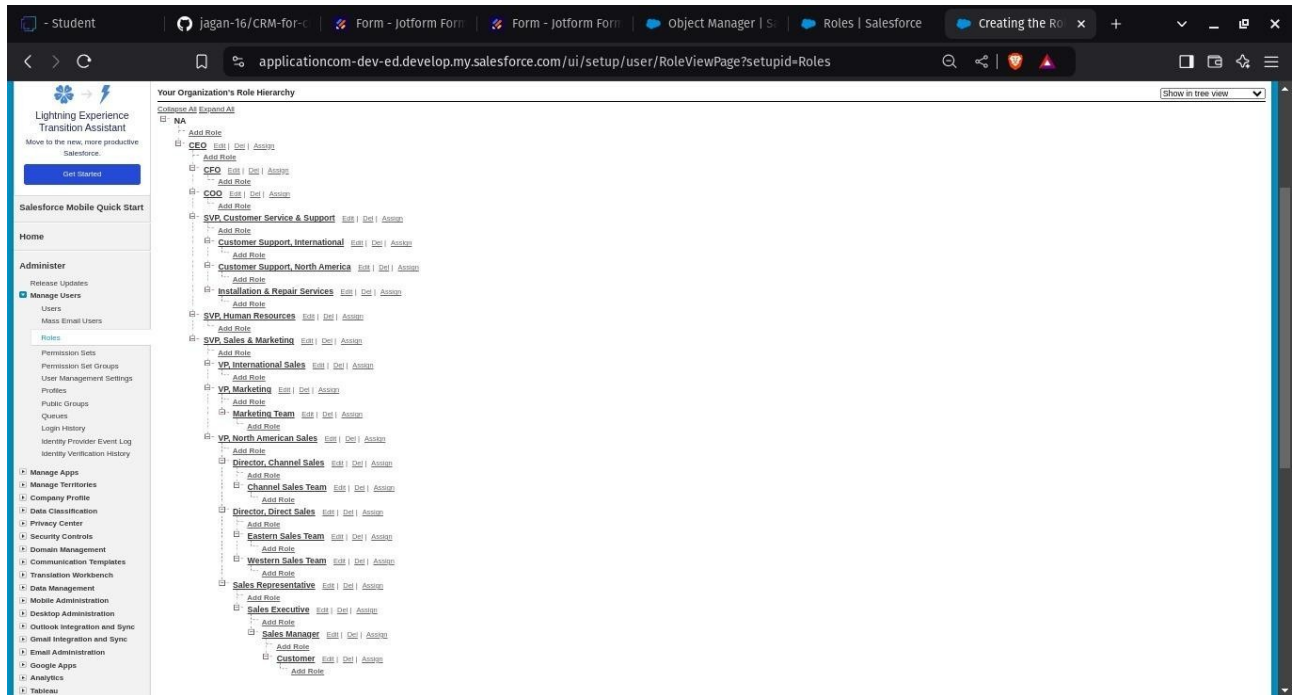
The screenshot displays the Jotform Form Builder interface in a web browser. The browser's address bar shows the URL: [jotform.com/build/243153828082053/settings/integrations](https://jotform.com/build/243153828082053/settings/integrations). The interface has a dark sidebar on the left with navigation options: FORM SETTINGS, EMAILS, CONDITIONS, THANK YOU PAGE, INTEGRATIONS (highlighted), WORKFLOWS, JOTFORM SIGN, and MOBILE NOTIFICATIONS. The main area is titled 'SALESFORCE' and 'Select a Salesforce Object', with 'Customer' selected in a dropdown. Below this, a 'Create a record' section shows a mapping between 'Object Fields' and 'Form' fields. The mapping includes: Name, City, Budget Amount, Property Type, Street Address, Verified, Email, and State, each paired with a corresponding form field. A 'Give Feedback' button is visible in the bottom right corner.

### STEP4:CREATE ROLES

Create Roles as per business requirement Steps:

- Go to Setup and Click on Roles, then click on Expand all and Add a Role just below the Sales Representative. \* It will use the “System Administrator Profile”.
- Label-Sales Executive
- Reports to -Sales Representative
- Similarly Create a Role Name “Sales Manager” below Sales Executive which reports to Sales Executive, Also Add a Role below SalesManager labeled as “Customer” which reports to SalesManager.

**FIGURE7:**

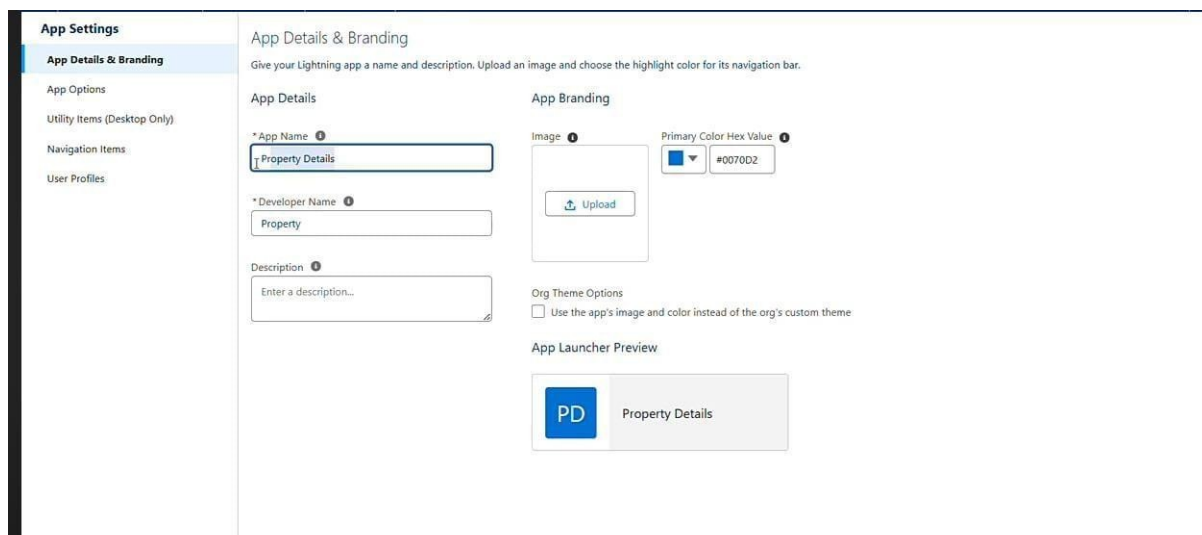


## STEP5:CREATE A PROPERTY DETAILS APP

An App where the objects will be displayed Steps:

- From Setup>> Go to App Manager and click on New Lightning App and Name it as “Property Details” and add “Customer” and “Property”Object.
- Click Next >>Next>> Save and Add “System Admin”Profile.

**FIGURE8:**



## STEP6:CREATE PROFILES

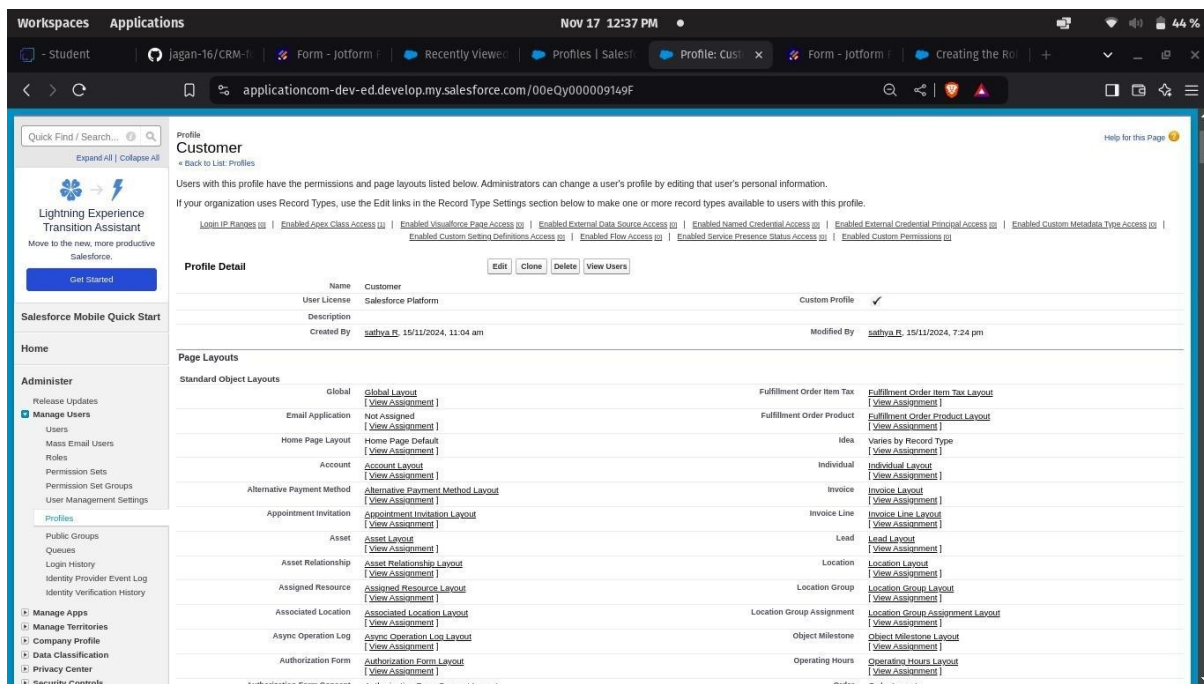
Create profiles as per business requirement

### Customer:

#### Steps:

- 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.
- Also Remove all the Standard Object Permissions.

### FIGURE9:



### Manager:

#### Steps:

1. From Setup>> Go to Profiles and Clone Salesforce Platform User and Name it “Manager”..
2. Uncheck all the Custom Objects and Check only Property Details

From Custom App Settings.

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

FIGURE10:

The screenshot displays the Salesforce Profile Manager interface. The top navigation bar includes the Salesforce logo, a search bar, and links for 'Switch to Lightning Experience', 'sathya R', 'Setup', 'Help', and 'Content'. The main header area features a banner for 'It's Better in Lightning' with a 'Check Readiness' button. The left sidebar contains navigation links for 'Quick Find / Search...', 'Lightning Experience Transition Assistant', 'Salesforce Mobile Quick Start', 'Home', and 'Administer' (with sub-links for Release Updates, Manage Users, Users, Mass Email Users, Roles, Permission Sets, User Management Settings, and Profiles).

The main content area is titled 'Profile Manager' and includes a 'Back to List Profiles' link. It provides instructions on how to manage user profiles and lists various settings for the 'Manager' profile. The settings are organized into sections: 'Profile Detail' and 'Page Layouts'.

**Profile Detail**

Name	Manager
User License	Salesforce Platform
Description	
Created By	sathya R. 15/11/2024, 11:14 am
Modified By	sathya R. 15/11/2024, 7:24 pm

**Page Layouts**

Standard Object Layouts	Global	Global Layout	Fulfillment Order Item Tax	Fulfillment Order Item Tax Layout
Email Application	Not Assigned	[View Assignment]	Fulfillment Order Product	Fulfillment Order Product Layout
Home Page Layout	Home Page Default	[View Assignment]	Invoice	Invoice Layout
Account	Account Layout	[View Assignment]	Invoice Line	Invoice Line Layout
Alternative Payment Method	Alternative Payment Method Layout	[View Assignment]		
Appointment Invitation	Appointment Invitation Layout	[View Assignment]		

## STEP7:CREATE A CHECKBOX FIELD ON USER

Create Field on the User as per the business requirement.

FIGURE11:

The screenshot shows the Salesforce Custom Field Definition page for a field named 'Verified' on the 'User' object. The field is of type 'Checkbox'. The page includes a sidebar with navigation links, a top navigation bar, and a main content area with tabs for 'Field Information', 'Set Field-Level Security', 'View Field Accessibility', and 'Where is this used?'. The 'Field Information' tab is active, showing details like Field Label, Field Name, API Name, Description, Help Text, Data Owner, Field Usage, Data Sensitivity Level, and Compliance Categorization. The 'General Options' section shows the Default Value as 'Unchecked'. The 'Validation Rules' section indicates that no validation rules are defined.

Field Information	
Field Label	Verified
Field Name	Verified
API Name	Verified__c
Description	
Help Text	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	
Created By	sathya R. 15/11/2024, 11:29 am
Modified By	sathya R. 15/11/2024, 11:29 am

General Options	
Default Value	Unchecked

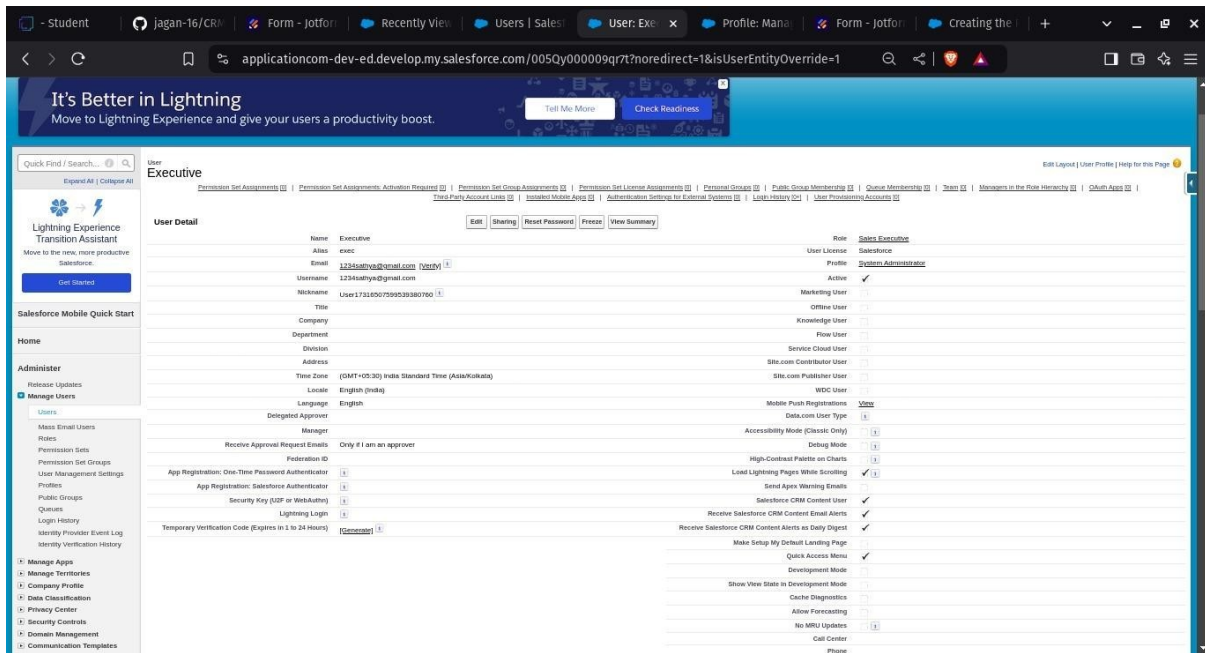
Validation Rules: No validation rules defined.

## STEP8:CREATE USERS

### USER1:

- GotoSetup-->Administration--> Users-->NewUser
- LastName- Executive
- Role- SalesExecutive
- License-Salesforce
- Profile - System Administrator
- 6.Save

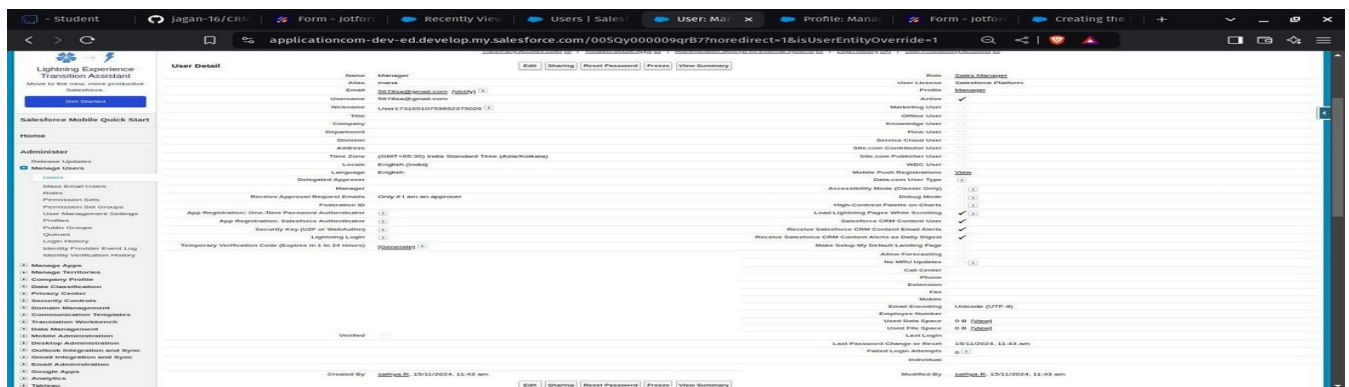
**FIGURE12:**



**USER2:**

- GotoSetup>>Administration>>Users>>NewUser
- LastName>>Manager
- Role>>SalesManager
- License>>Salesforce Platform
- Profile>>Manager
- Save

**FIGURE13:**





### USER3:

- Goto Setup>>Administration>> Users>>New User
- Last Name>> Customer
- Role>> Customer
- License>>Salesforce Platform
- Profile>>Customer
- Make Sure the verified check box is “Unchecked”
- Save

FIGURE:14

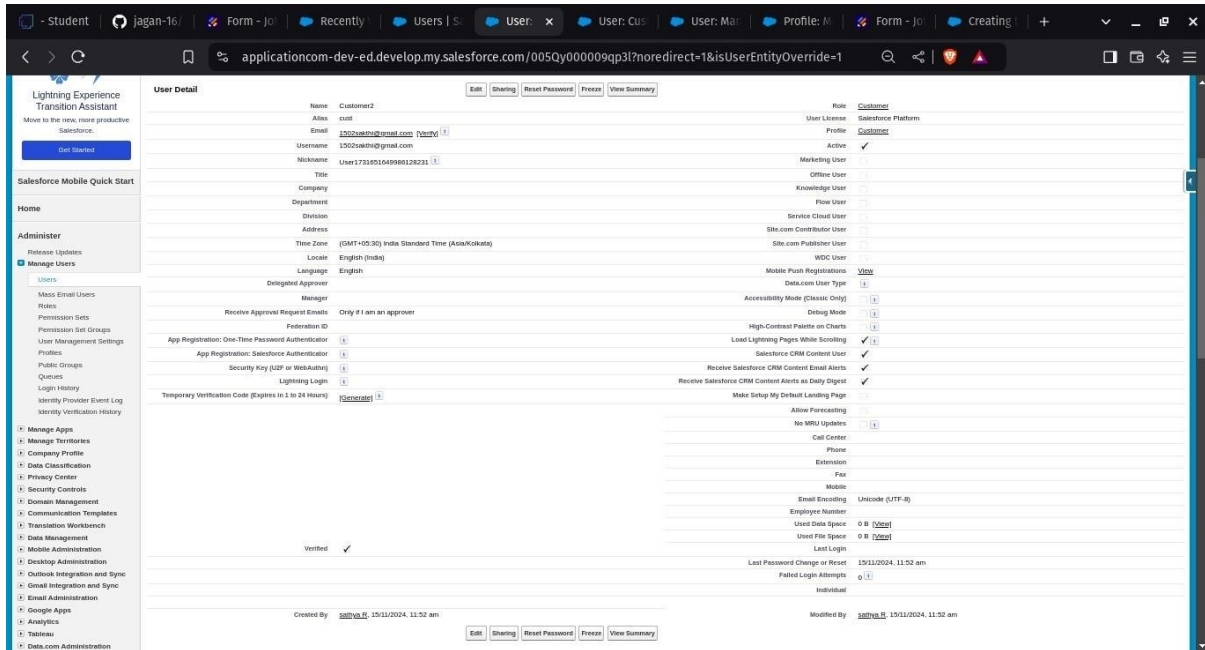
The screenshot shows the Salesforce 'User Detail' page for a new user. The user's name is 'Customer', email is '8912satmyaa@gmail.com', and role is 'Customer'. The 'Verified' checkbox is unchecked. The page includes a left sidebar with navigation options like 'Manage Users', 'Roles', and 'Permissions'. The main content area displays various user attributes and settings, including 'Marketing User', 'Offline User', 'Knowledge User', 'Flow User', 'Service Cloud User', 'Site.com Contributor User', 'Site.com Publisher User', 'WDC User', 'Mobile Push Registrations', 'Data.com User Type', 'Accessibility Mode (Classic Only)', 'Default Mode', 'High-Contrast Palette on Charts', 'Load Lightning Pages While Scrolling', 'Salesforce CRM Content User', 'Receive Salesforce CRM Content Email Alerts', 'Receive Salesforce CRM Content Alerts as Daily Digest', 'Make Setup My Default Landing Page', 'Allow Forecasting', 'No MRU Updates', 'Call Center', 'Phone', 'Extension', 'Fax', 'Mobile', 'Email Encoding', 'Unicode (UTF-8)', 'Employee Number', 'Used Date Space', 'Used File Space', 'Last Login', 'Last Password Change or Reset', 'Failed Login Attempts', and 'Individual'.

### USER4:

- Goto Setup>>Administration>>Users>>New User
- Last Name>>Customer2
- Role>>Customer
- License>>Salesforce Platform
- Profile>>Customer
- Make Sure the verified checkbox is “checked”



**FIGURE15:**



## STEP 9: CREATE AN APPROVAL PROCESS FOR PROPERTY OBJECT

### Steps:

- From Setup>>Process Automation>>Approval Process
- Process Name -PropertyA pproval
- Location is not equal to blank
- Verified Equals false.
- Click next and “Next Automated Approver Determined By Select Manager.
- From Record Edit ability Properties>>Click on
- Administrators OR the currently assigned approver can edit records during the approval process.
- From Step 5 Select Fields to Display on Approval Page Layout select Property, Owner, Location,Type.
- Click Next and Select the initial Submit
- Owner>>Property Owner
- Roles>>Sales Manager
- Save
- 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”.
- Add One field Update as “UnVerified Property” Select Object>>Property
- Field to Update >> Verified Field Data Type>>CheckBox
- Select CheckBox Option as “False” Save.
- Add One field Update as “UnVerified Property” Select Object>> Propert Field to Update>>Verified
- Field Data Type >> CheckBox
- Select CheckBox Option as “False”
- Activate the approval process

**FIGURE16:**

The screenshot displays the Salesforce Lightning Experience interface for defining an approval process. The browser address bar shows the URL: `applicationcom-dev-ed.develop.my.salesforce.com/04aQy000000IEPV`. The page title is "Property: Property Approval".

**Process Definition Detail:**

- Process Name:** Property Approval
- Unique Name:** Property\_Approval
- Description:** PROPERTY: LOCATION NOT EQUAL TO UNVERIFIED PROPERTY: VERIFIED EQUALS TRUE
- Key Fields:** @Property: LOCATION NOT EQUAL TO UNVERIFIED PROPERTY: VERIFIED EQUALS TRUE
- Record Eligibility:** Administrator OR Current Approver
- Approval Assignment Rule Template:** Allow Submitters to Recall Approval Requests
- Initial Submitters:** jagan@16
- Property Owner:** jagan@16
- Created By:** jagan@16
- Modified By:** jagan@16
- Created On:** 10/10/2024, 11:05 am

**Initial Submission Actions:**

- Action:** Record Lock
- Description:** Lock the record from being edited

**Approval Steps:**

Action	Type	Step Number	Name	Description	Criteria	Assigned Approver	Report Behavior
Approve <td>Final Update <td>1 <td>VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td></td></td></td>	Final Update <td>1 <td>VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td></td></td>	1 <td>VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td></td>	VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td>	<td> <td>VP Approval</td> <td>Final Report</td> </td>	<td>VP Approval</td> <td>Final Report</td>	VP Approval	Final Report
Reject <td>Final Update <td>2 <td>VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td></td></td></td>	Final Update <td>2 <td>VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td></td></td>	2 <td>VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td></td>	VP Approval <td> <td> <td>VP Approval</td> <td>Final Report</td> </td></td>	<td> <td>VP Approval</td> <td>Final Report</td> </td>	<td>VP Approval</td> <td>Final Report</td>	VP Approval	Final Report

**Final Approval Actions:**

- Action:** Record Lock
- Description:** Lock the record from being edited

**Final Rejection Actions:**

- Action:** Record Lock
- Description:** Lock the record from being edited

**Recall Actions:**

- Action:** Record Lock
- Description:** Lock the record from being edited

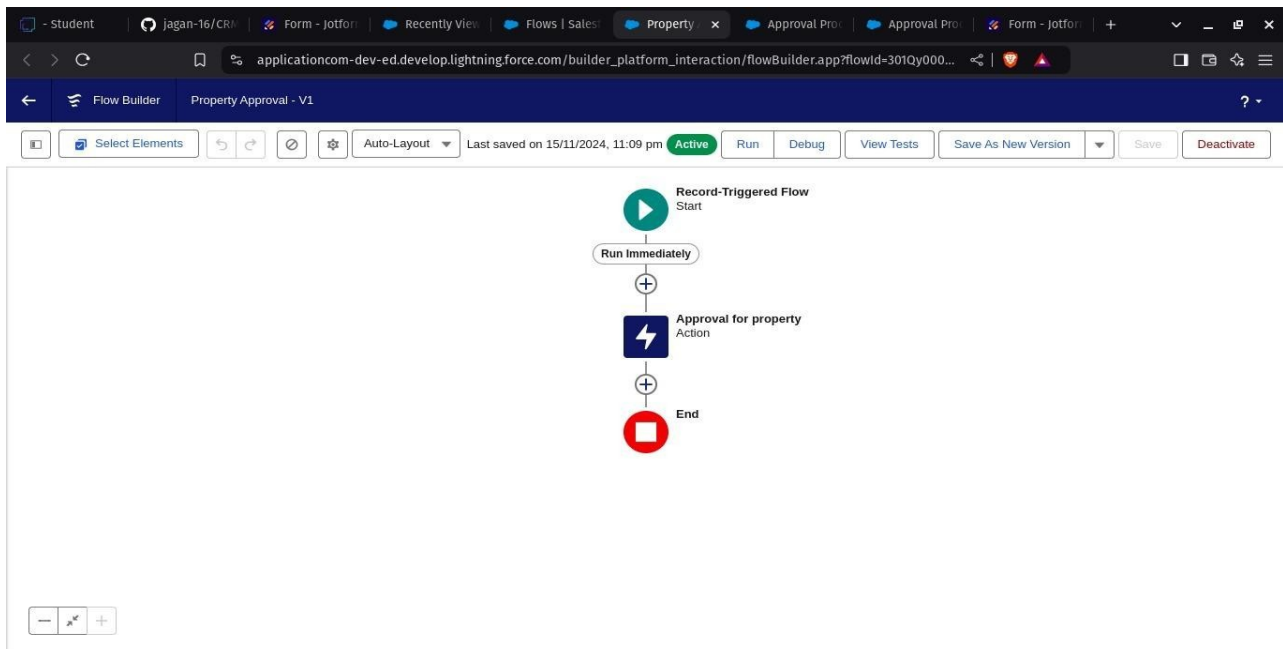
## STEP10:CREATE A RECORD TRIGGER FLOW TO SUMBIT THE APPROVAL PROCESS AUTOMATICALLY

A flow that can submit the records directly for approval

### Steps:

- From Setup >> Search for Flows >> Click On New and Select “RecordTriggerFlow”.
- 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}
- Done
- Save the Flow and Give label as “Property Approval” and “Activate”.

FIGURE17:



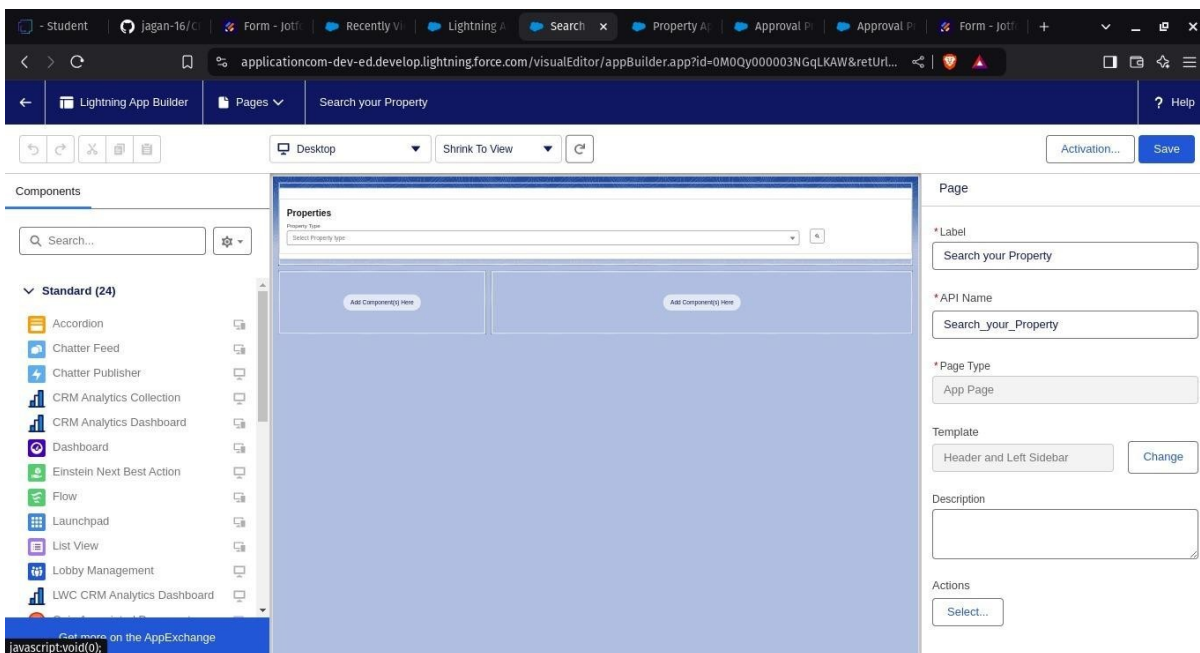
## STEP11:CREATE AN APP PAGE

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

### STEPS:

- 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.
- Then click on Save.

FIGURE18:



## STEP12:CREATE A LWC COMPONENT

Create an LWC Component for the customers so that only verified customers can access the verified properties and non Verified customers can access non verified properties, and deploy it on“Search your Property Page”

### STEPS:

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

### CODE:

```
public class PropertHandler_LWC{

    @AuraEnabled(cacheable=true)

    public static list<Property__c> getProperty(string type , boolean verified){

        return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c FROM Property__c Where
        Type__c =: type AND Verified__c =: verified];

    }

}
```

### CODE:HTML

```
<template>

    <lightning-card>
        <div class="slds-box">
            <div class="slds-text-align_left">
                <h1style="font-size:20px;"><b>Properties</b></h1>
            </div>
            <div>
                <div class="slds-gridslds-gutters">
                    <div class="slds-colslds-size_5-of-6">
                        <lightning-combobox name="Type" label="Property Type" value={type
var}placeholder="Select Property type"
                        options={property options} on change={changehandler}></lightning-combobox>
                    </div>
                    <div class="slds-colslds-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={is true}>
    <div class="slds-box">
        <lightning-datatable key-field="id" data={property
list} columns={columns}></lightning-datatable>
    </div>
</template>
<template if:false={is false}>
    <div class="slds-box">
        <div style="font-size:15px;"><b>No properties Are Found!!</b></div>
    </div>
</template>
</lightning-card>
</template>

```

In Your Js File Write this code :

## CODE:JS

```

import{LightningElement,api,track,wire}from'lw';
import getProperty from
"@salesforce/apex/PropertHandler_LWC.getProperty"import{ getRecord}
from'lightning/uiRecordApi';
importUSER_IDfrom'@salesforce/user/Id';
export default class C_01_Property_Management extends LightningElement
{
    @apirecordId
    userId =
    USER_ID;verified
    vartypevar
    isfalse =
    true;istrue=fal
    se;
    @track propertylist =

```

```

[];columns=[
    {label:'PropertyName',fieldName:'Property_Namec'},
    {label:'PropertyType',fieldName:'Typec'},
    {label:'PropertyLocation',fieldName:'Locationc'},
    {label:"Propertylink",fieldName:"Property_linkc"}
]
propetyoptions=[
    {label:"Commercial",value:"Commercial"},
    {label:"Residential",value:"Residential"},
    {label:"rental",value:"rental"}

]
@wire(getRecord, { recordId: "$userId", fields:
['User.Verifiedc'] })recordFunction({data,error}) {
    if (data)
        {console.log(data
        )
        console.log("This is the User Id --->
"+this.userId);this.verifiedvar=data.fields.Verifiedc.
value;
        } else {
            console.error(error)console.log('th
isiserror')
        }
    }

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(res
            ult)
            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.verified
                var);console.log(this.typevar)
            } else {
                this.is false =
                false;
                this.is true=false;
            }
        }
    }

```

```

    })
    .catch((error) =>
        { console.log(error)
        })
    }
}

```

In Your meta file give your targets to deploy the component.

### CODE:js-meta.xml

```

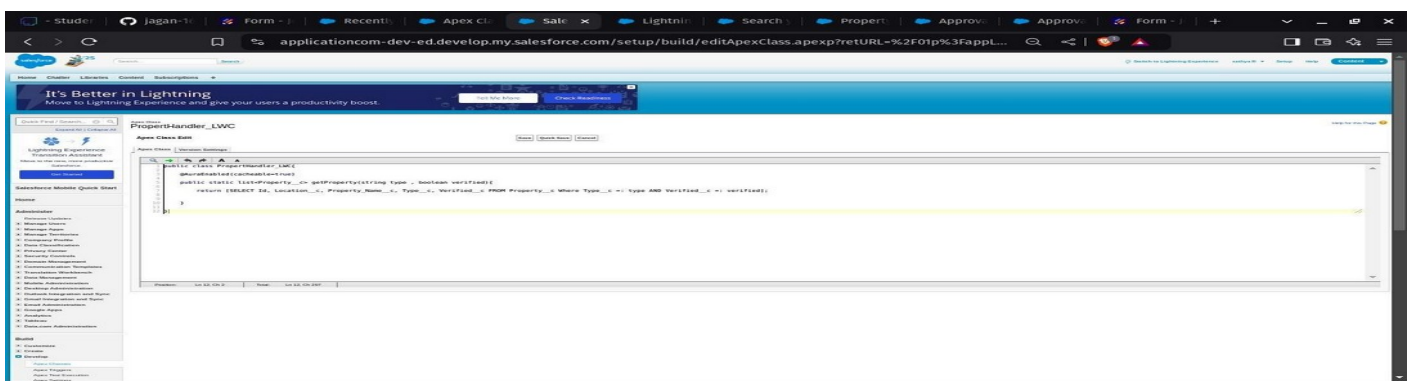
<?xmlversion="1.0"encoding="UTF-8"?>
<LightningComponentBundlexmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>59.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightningRecordPage</target>
        <target>lightningAppPage</target>
        <target>lightningHomePage</target>

    </targets>
</LightningComponentBundle>

```

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

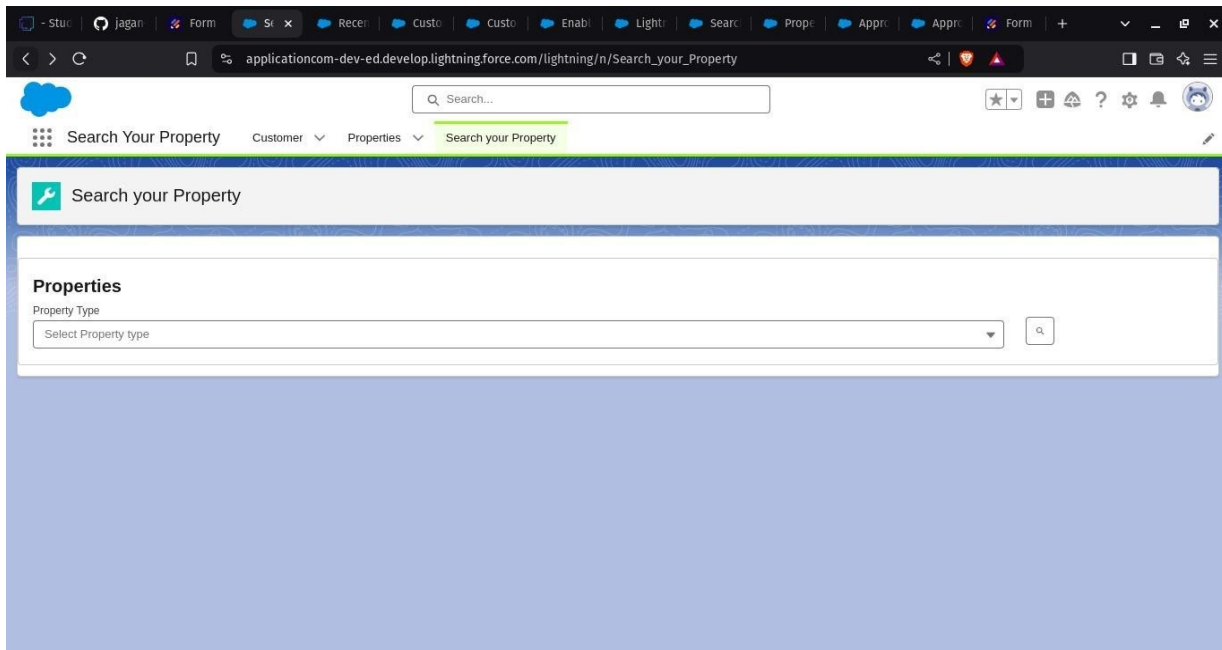
**FIGURE19:**











#### **STEP14:GIVE ACCESS OF APEX CLASSES TO PROFILES**

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

##### **STEPS:**

- From Setup >> Search For Apex Classes >> Click on “Security” behind “PropertyHandlerLWC”.
- From Profiles Add “Manager” and “Customer” and “Save”.

## **8. Conclusion**

The CRM Property Management System successfully combines Salesforce and JotForm to address the complexities of managing client and property-related workflows. By centralizing data, automating processes, and enhancing communication, this solution has improved operational efficiency and client satisfaction. It is scalable, secure, and tailored for future business growth, making it an ideal tool for property management enterprises.

