

**ANNA UNIVERSITY REGIONAL CAMPUS COIMBATORE**

**Project Title: CRM Application to Handle Client and their Property-Related Documents**

**Team members:**

**M.Ayishwaryaa - 710021104004**

**S.Radhika - 710021104308**

**K.Sandhya - 710021104309**

**P.S.Niranjaini - 710021104318**

**T.Swarna - 710021104706**

**1. Project Overview**

* This project, CRM Application for Document Management in Real Estate, is designed to address the challenge of efficiently managing client and property-related documentation. In the real estate industry, documentation is critical, and efficient handling and tracking of these documents can significantly impact client relationships and operational workflows. The aim of this project is to deliver a robust, scalable solution that leverages Salesforce’s CRM and document management capabilities to enhance data accuracy, streamline document processing, and improve user experience. By implementing this solution, we support the long-term goals of improving operational efficiency and maximizing client satisfaction within the organization’s sales and property management functions.

**2. Objectives**

**Business Goals:**

* **Enhanced Document Management:**: Implement a centralised document management system to securely store, track, and retrieve property-related and client-related documents.
* **Increased Efficiency in Client Management**: Simplify client interactions by reducing time spent locating and processing documents, improving response times and engagement.
* **Compliance and Security:** Ensure that all sensitive information is securely managed, adhering to regulatory and organizational standards for data handling and privacy.

**Specific Outcomes:**

* **Document Storage and Retrieval**: Store all property and client documents in a single, easily accessible location within Salesforce.
* **Automated Processes:** Use Salesforce automation to trigger notifications, reminders, and updates for document status changes, contract renewals, or missing information.
* **Improved Data Accuracy:** Minimize data entry errors and ensure accurate records through validation rules and automated data updates.
* **User Experience:** Enhance user interface design to make document and client data easy to view and manage for all team members.

**3. Salesforce Key Features and Concepts Utilized**

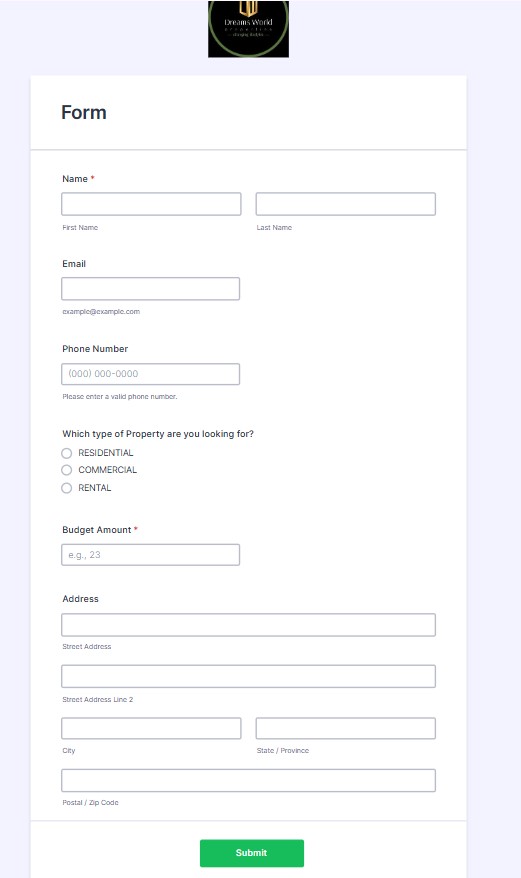
* **Salesforce CRM Capabilities:** Utilize Salesforce’s CRM functionality to organize and manage client interactions, document history, and property-related communications.
* **Salesforce Files & Attachments**: Leverage Salesforce Files to manage and securely store various types of documents, including contracts, legal documents, and property photos.
* **Custom Objects and Fields:** Create custom objects for “Properties” and “Documents” linked to client data to tailor Salesforce to real estate-specific requirements.
* **Workflow Rules and Process Builder**: Automate document-related processes, including sending notifications for document expiration or missing files.
* **Apex and Triggers:** Use Apex classes and triggers to handle advanced logic and real-time data updates where necessary.
* **Validation Rules and Data Security:** Ensure data integrity and security, implementing validation rules to maintain high data quality and field-level security to restrict access to sensitive information.

**4. Detailed Steps to Solution Design**

Project: A CRM Application to Handle the Clients and Their Property Related Requirements

**Snapshots of Outputs:**

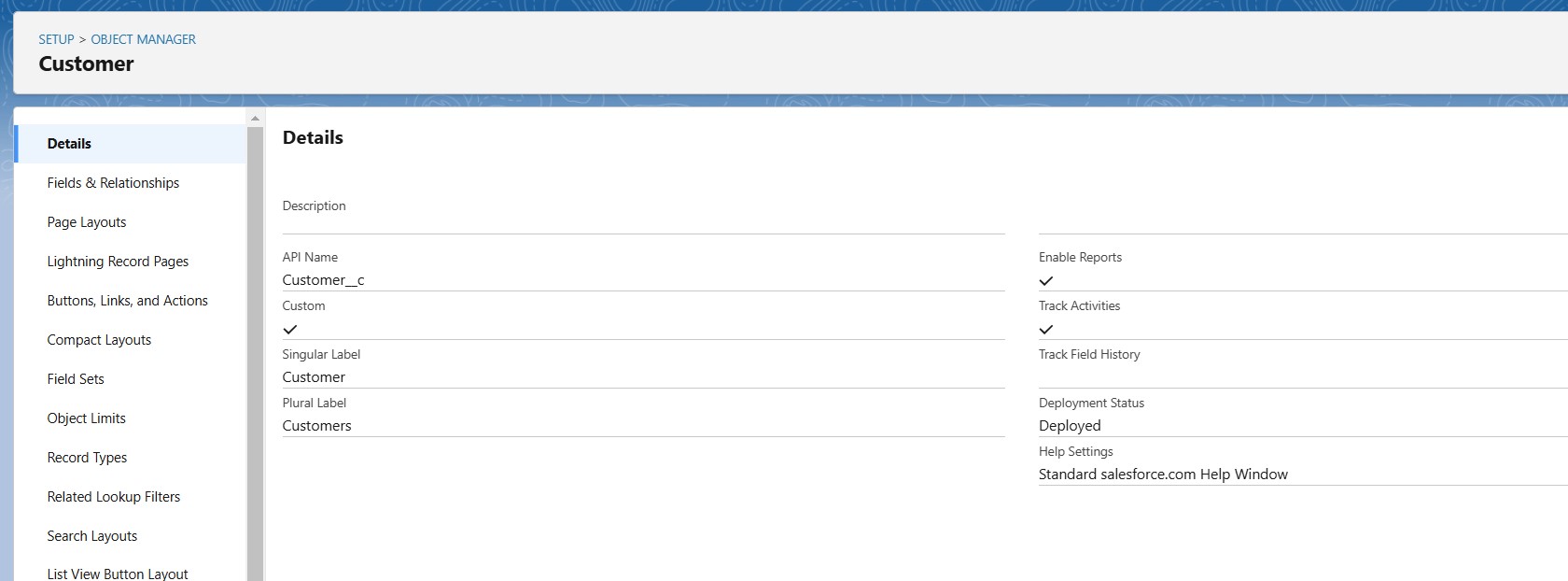
Step 1. Create A Jot form And Integrate It With The Org To Create A Record Of Customers Automatically.

Step 2.

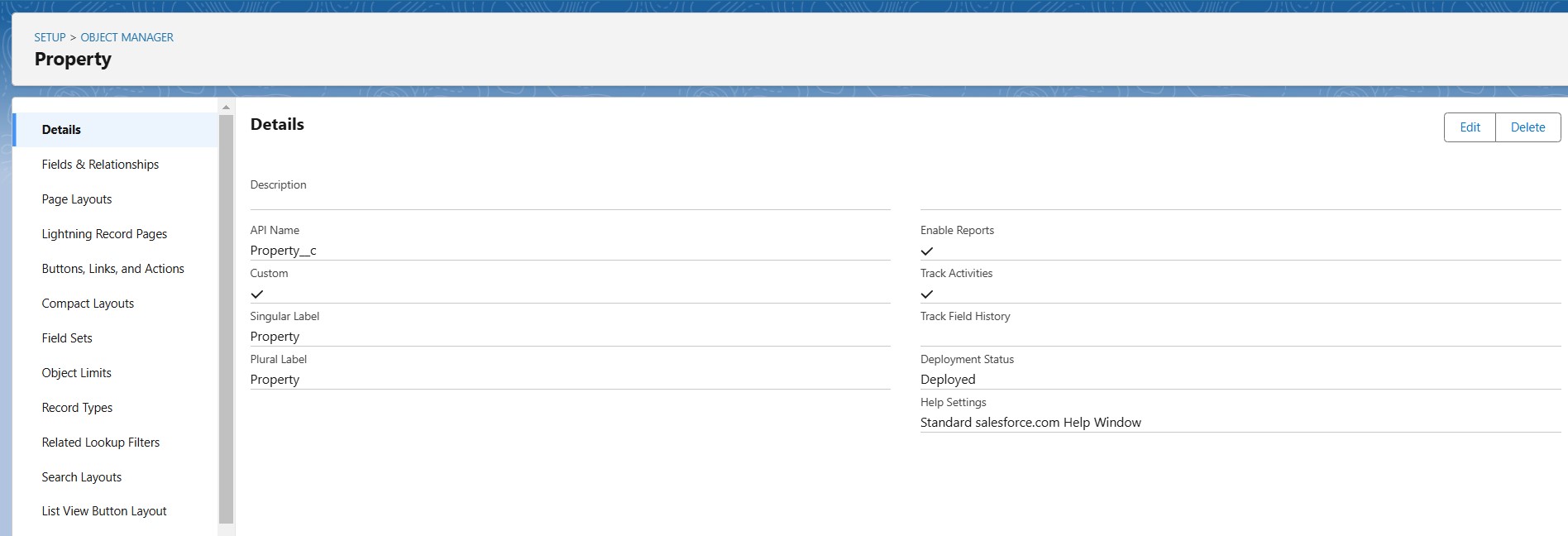
[Click here to view Jotform](https://form.jotform.com/243124538134047)

Step 3. Create Objects from Spreadsheet.

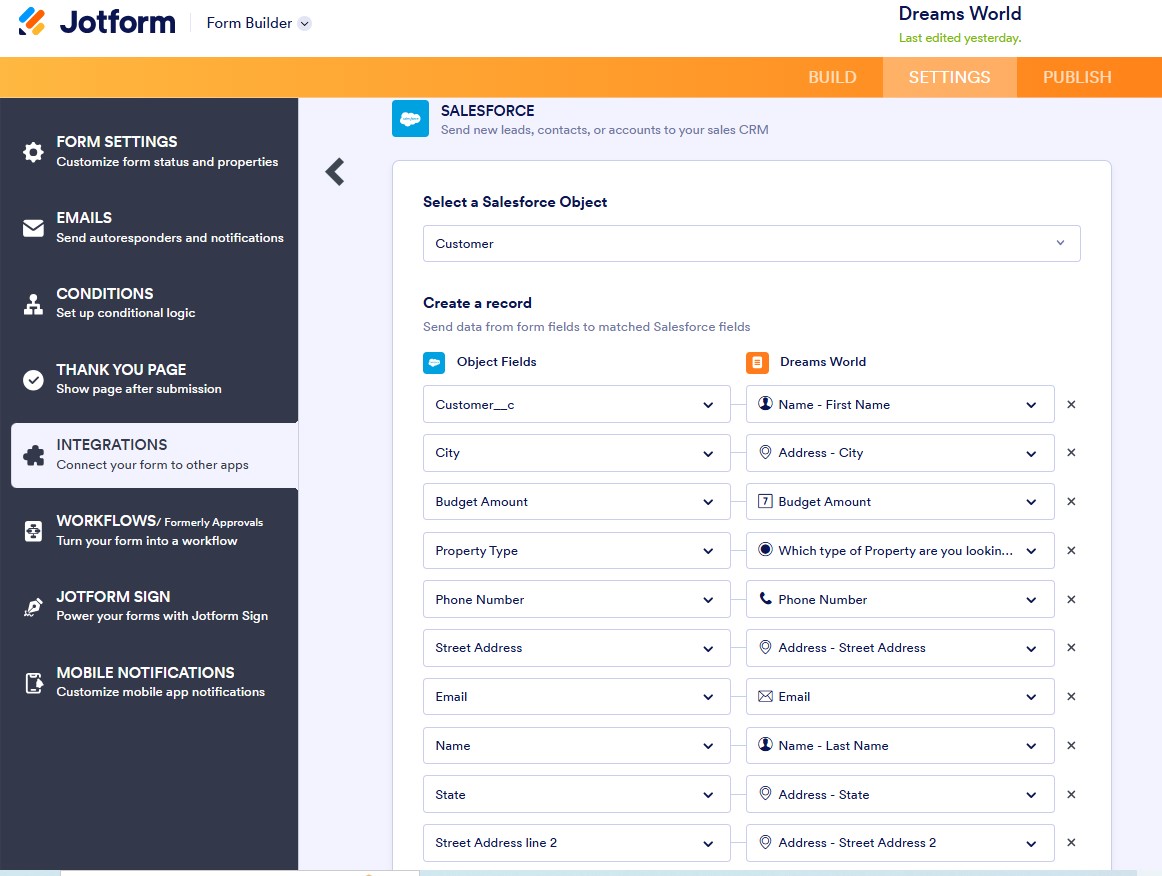
i) Create Customer Object.



Step 4. Create Property Object.

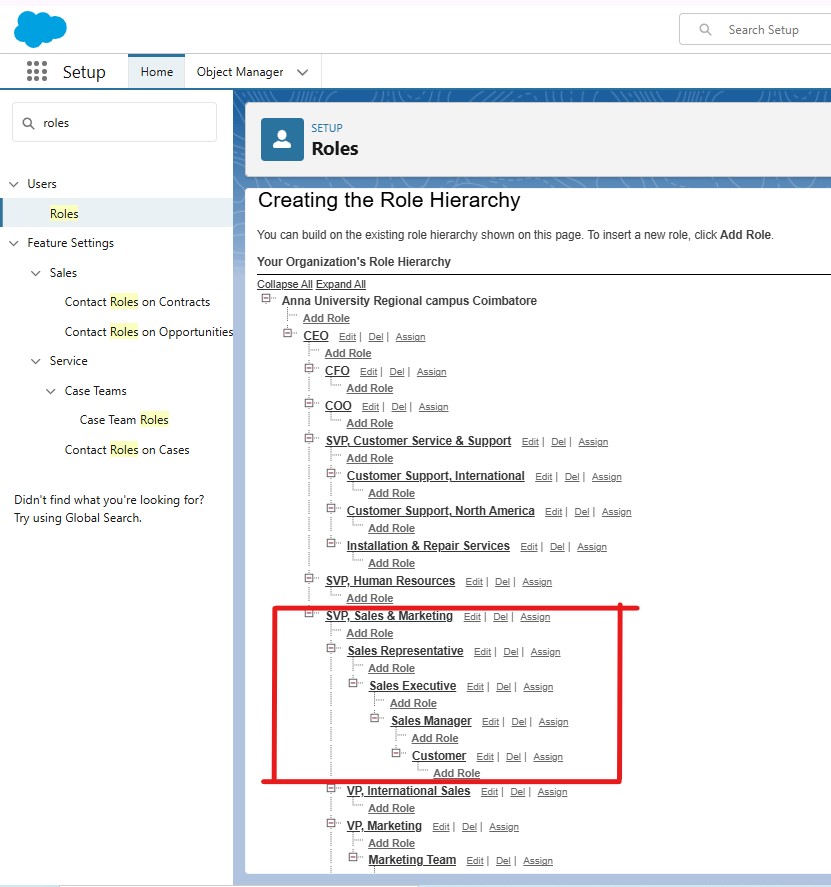


Step 5. Integrate Jotform With Salesforce Platform.

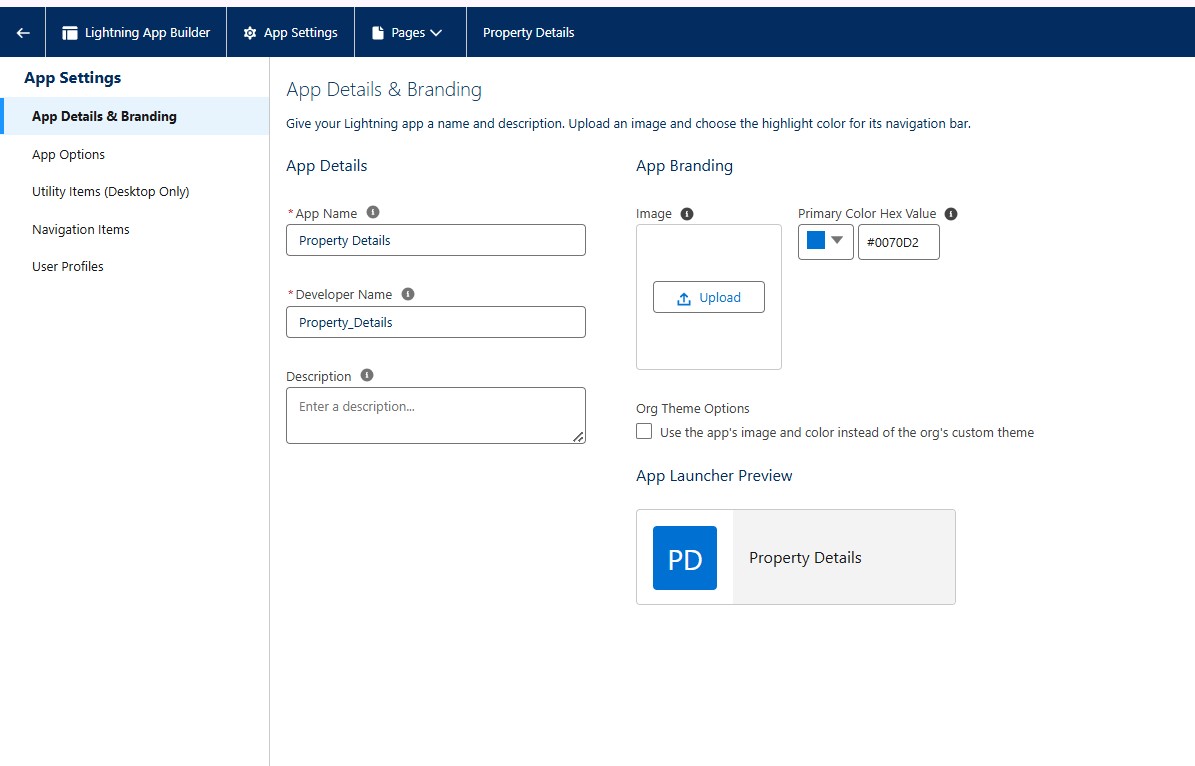


Step 6. Create Roles.

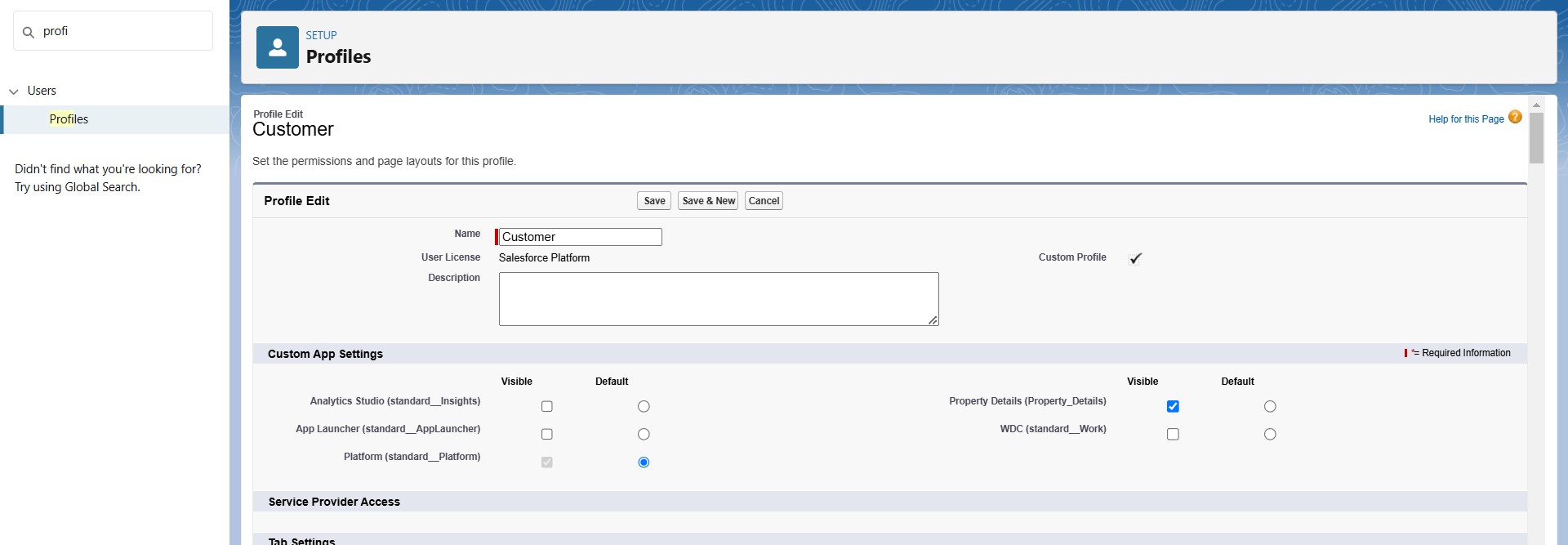
i) Sales Executive Role.



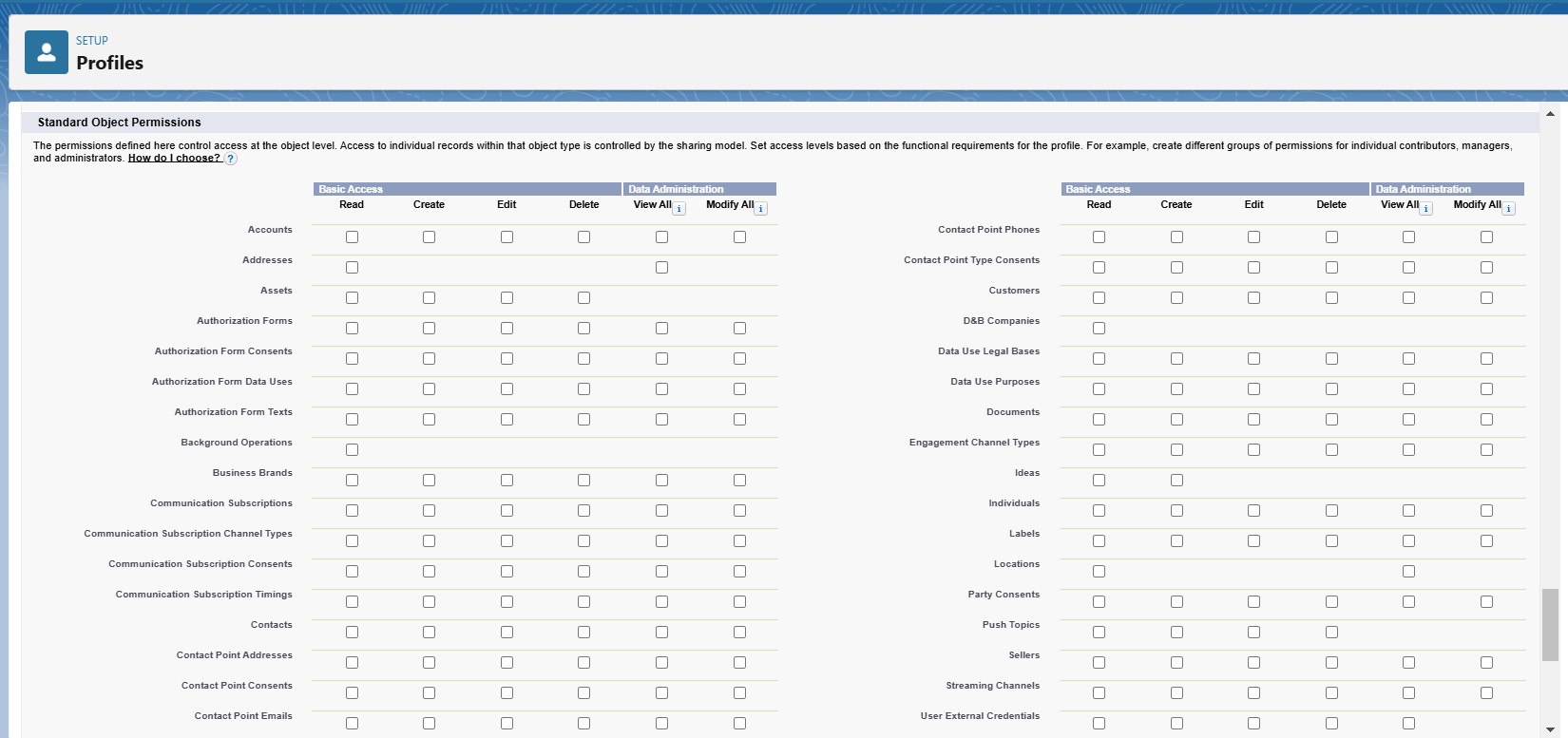
Step 7.Create A Property Details App.

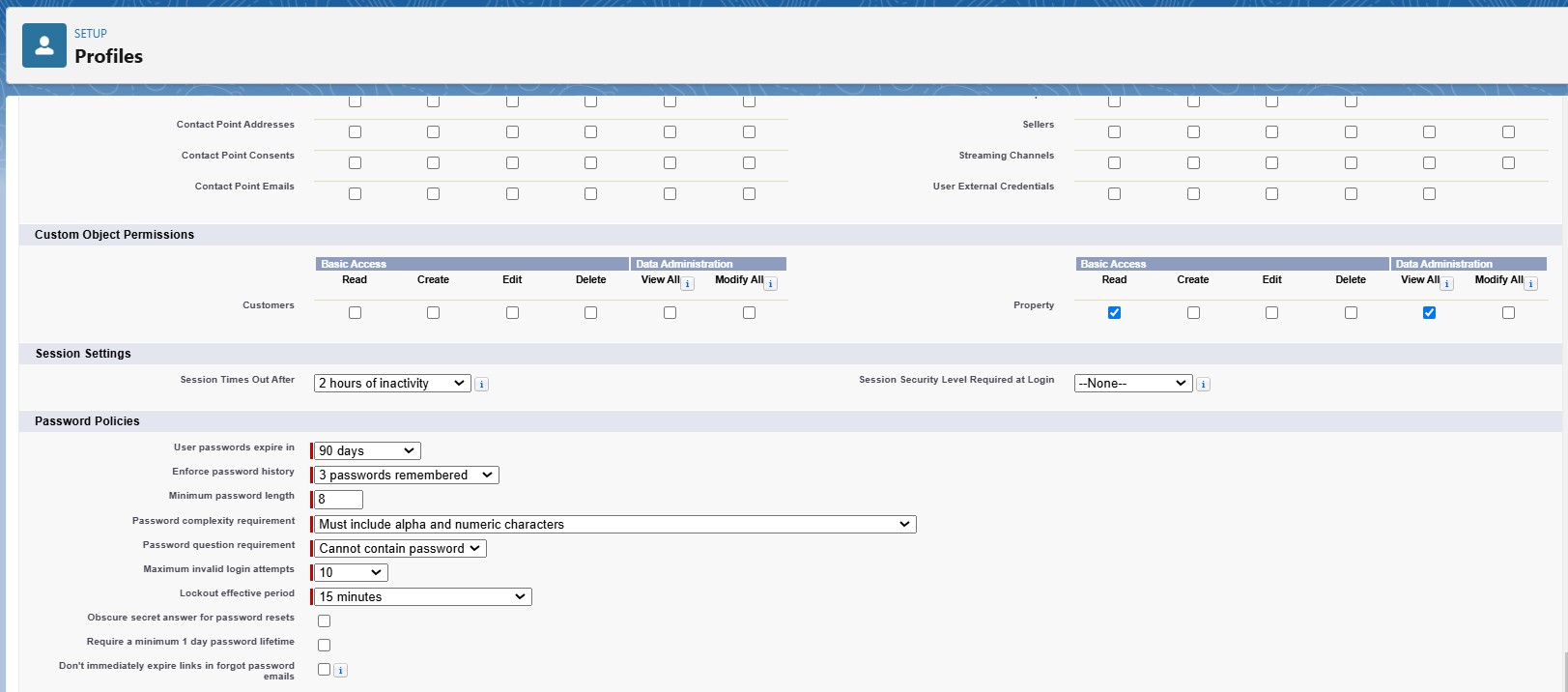


**Create Profiles for customers.**

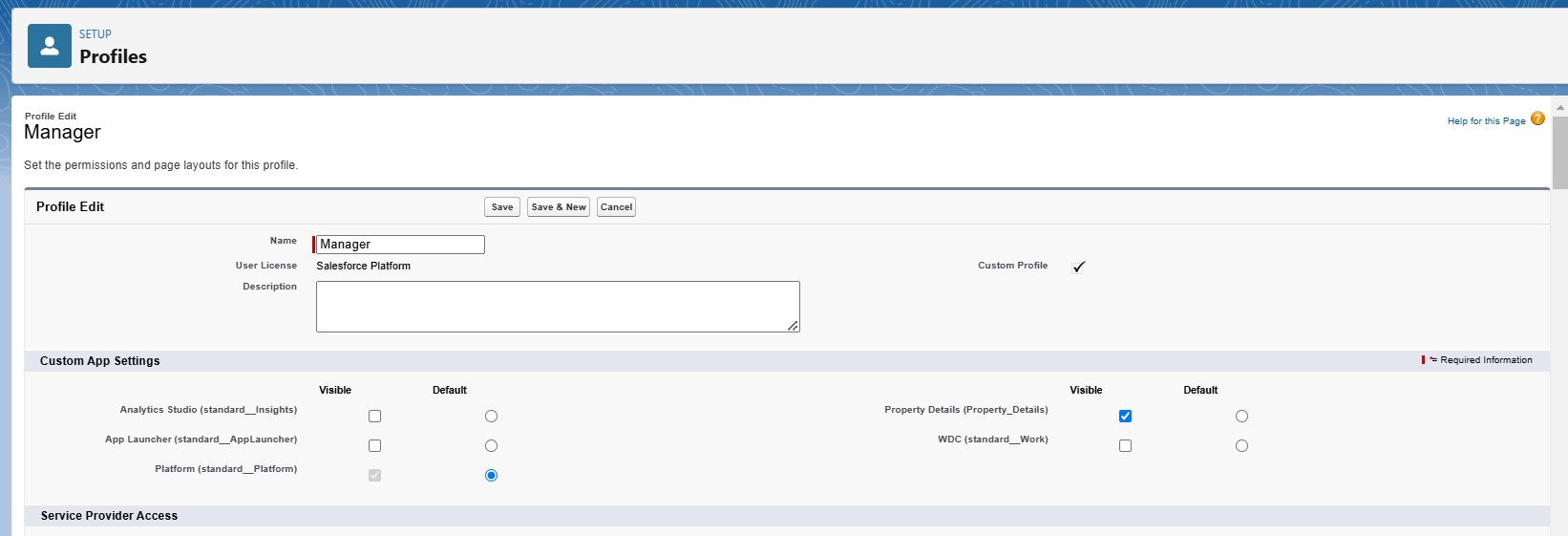
7.1 Customer profile 1

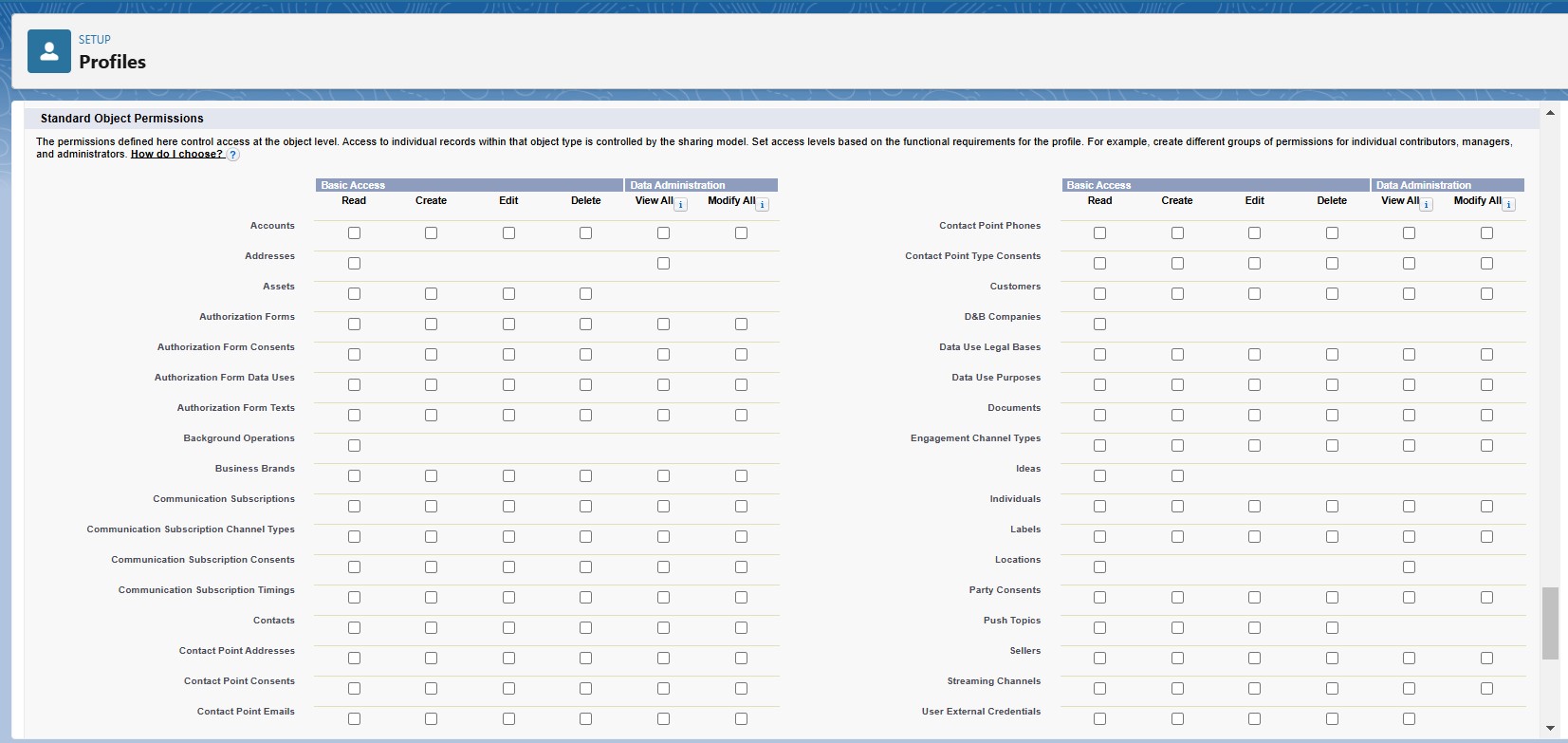
7.2 Customer profile 2



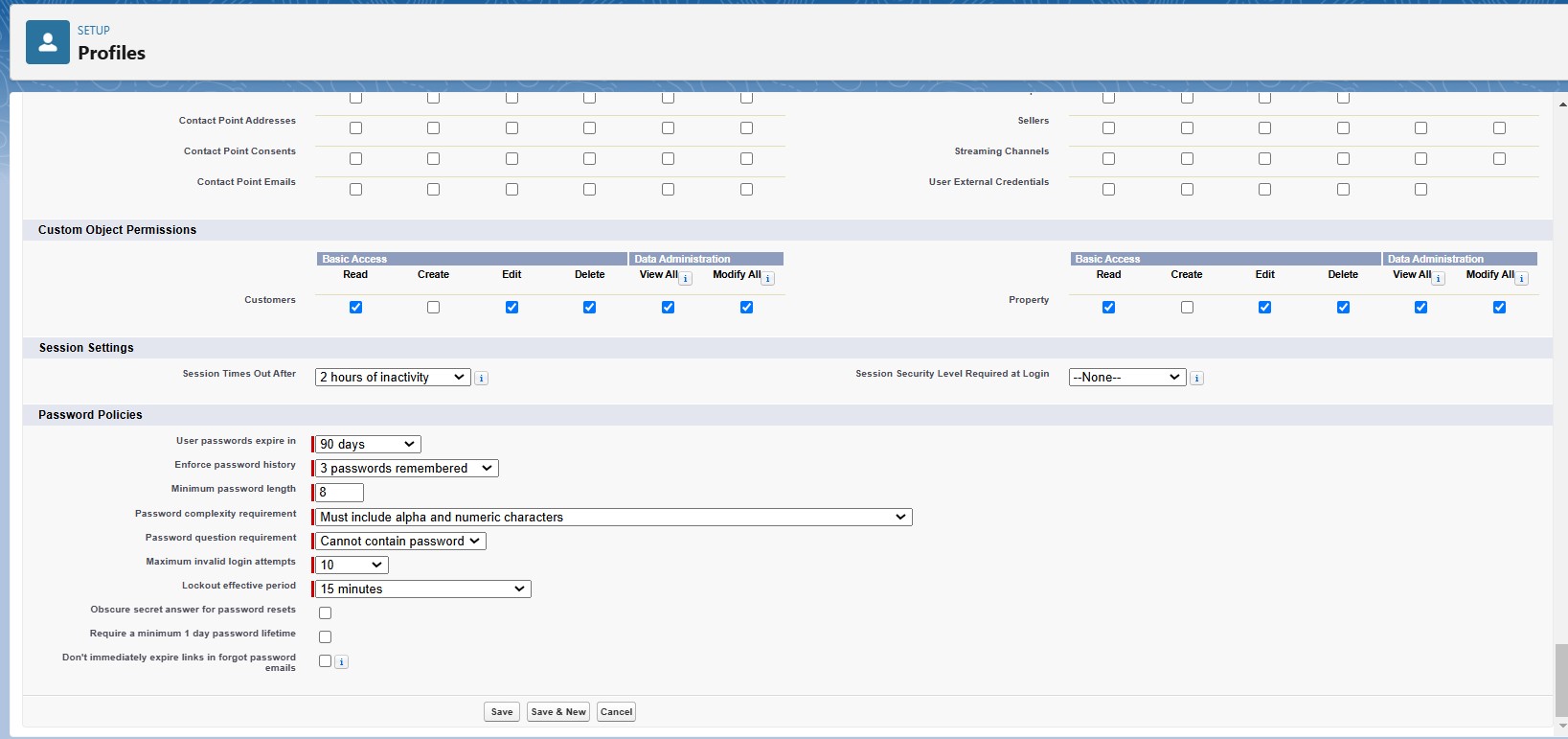
7.3 Customer profile 3

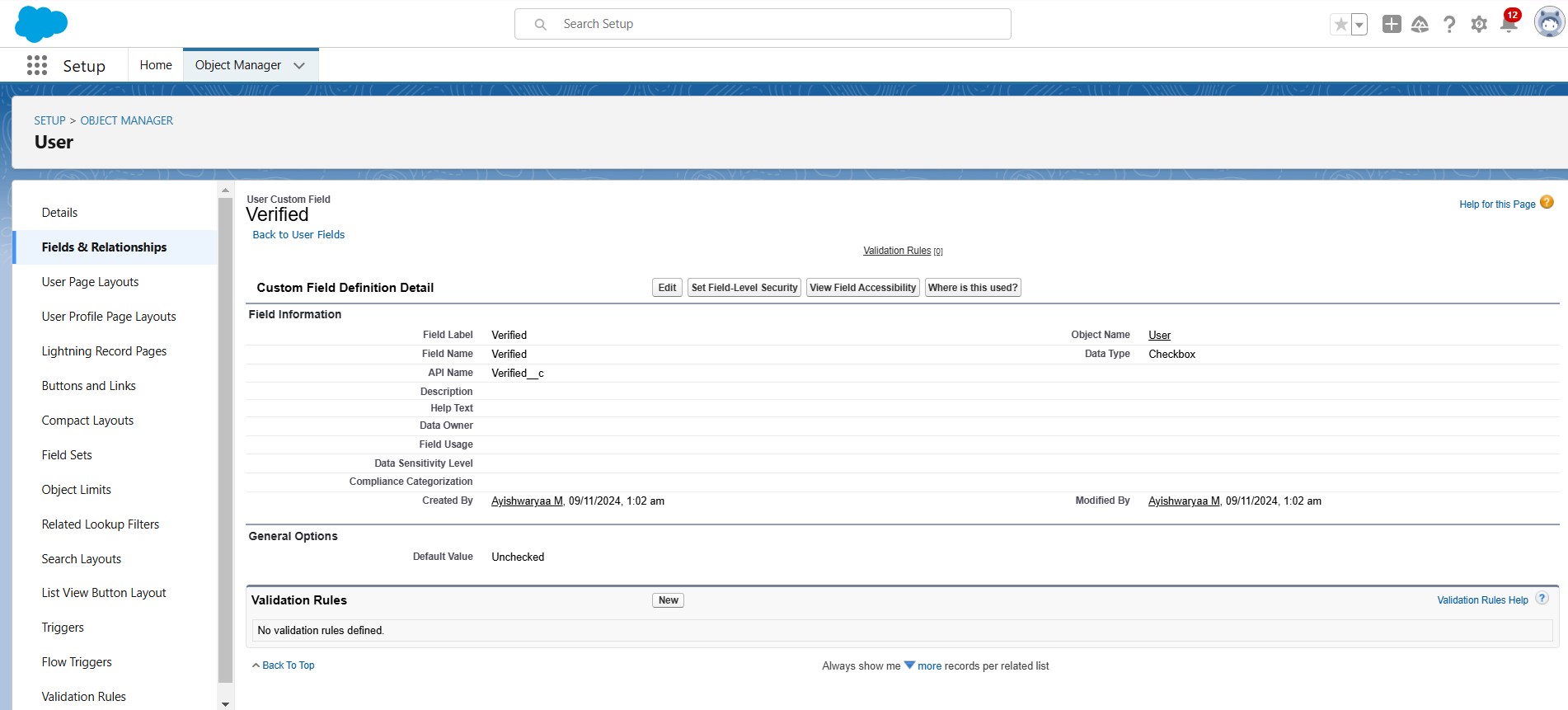
**Step 8. Create Profiles for Managers**

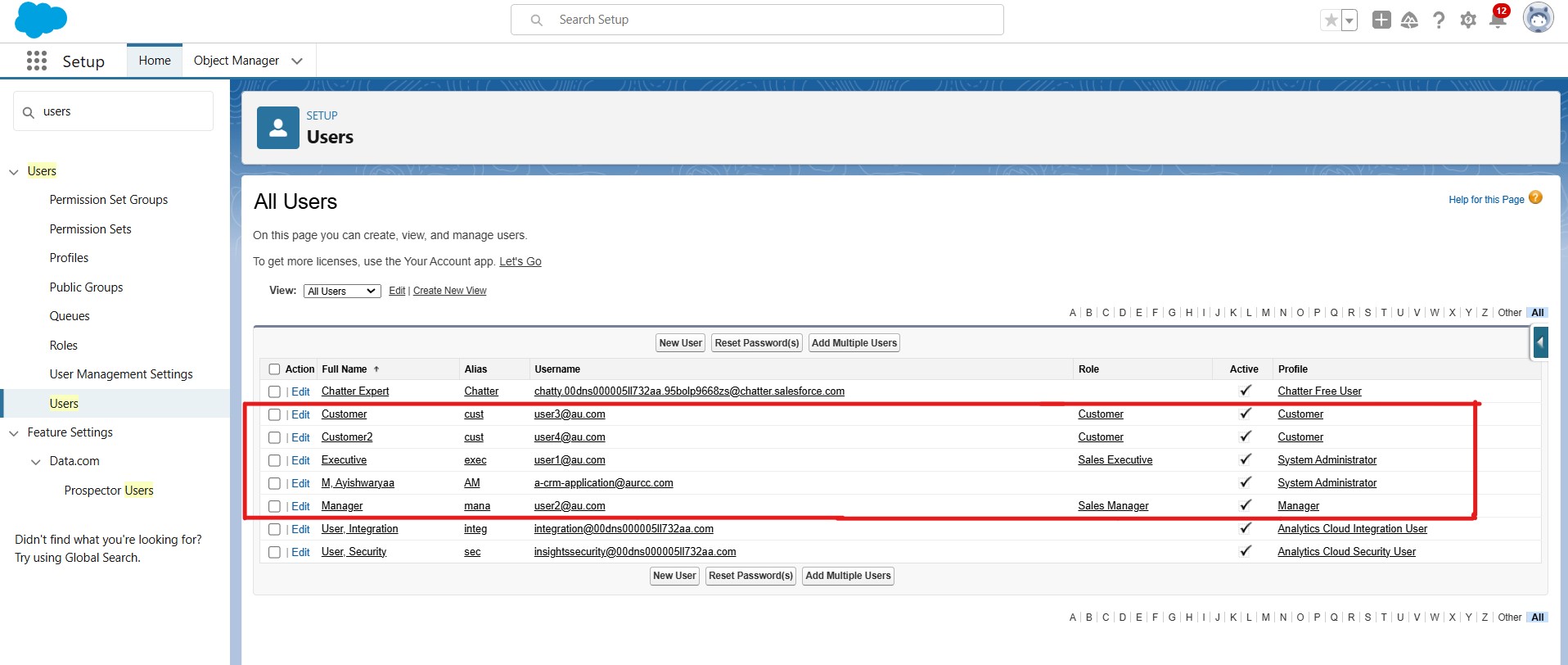
8.1 Manager Profile 1

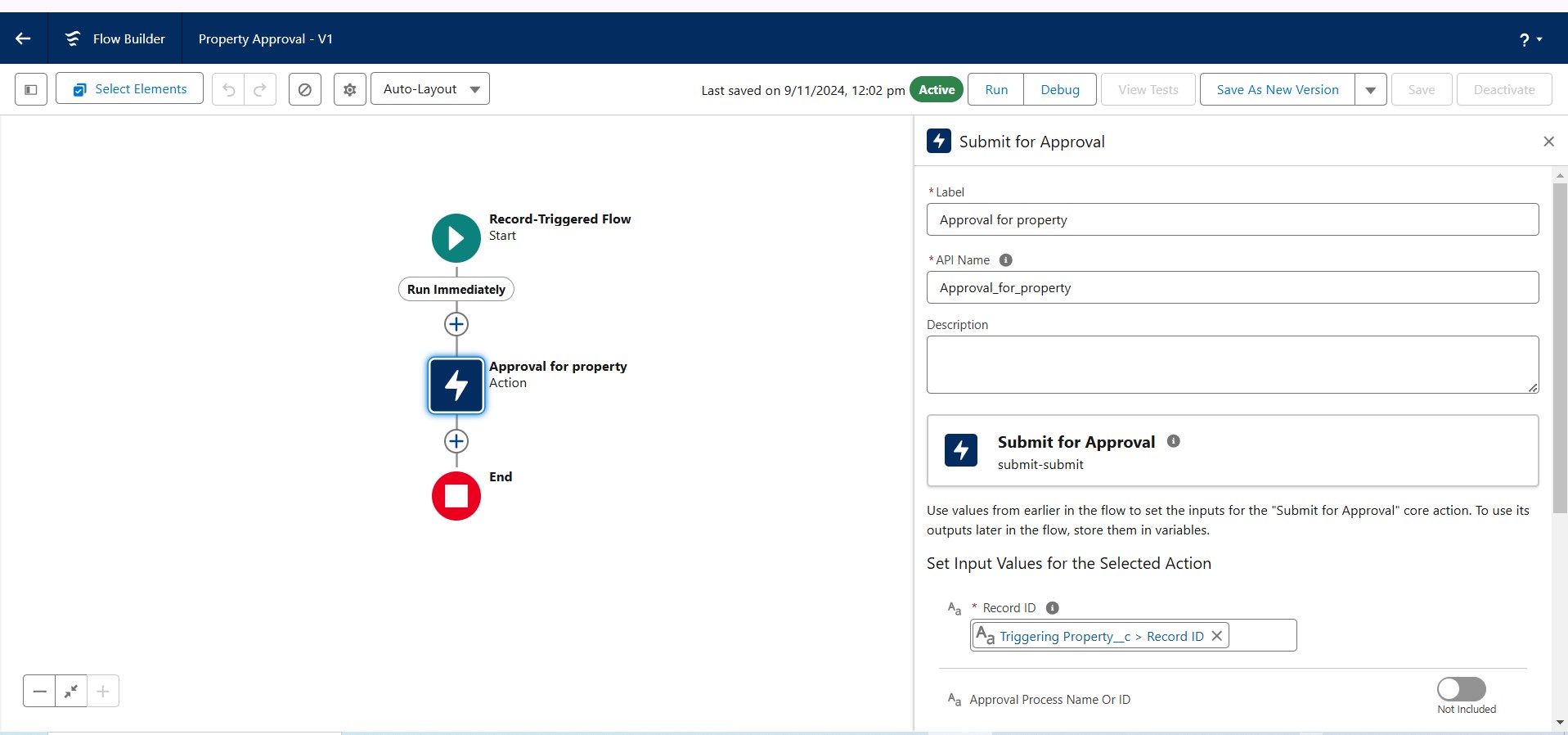
8.2 Manager profile 2

8.3 Manager profile 3

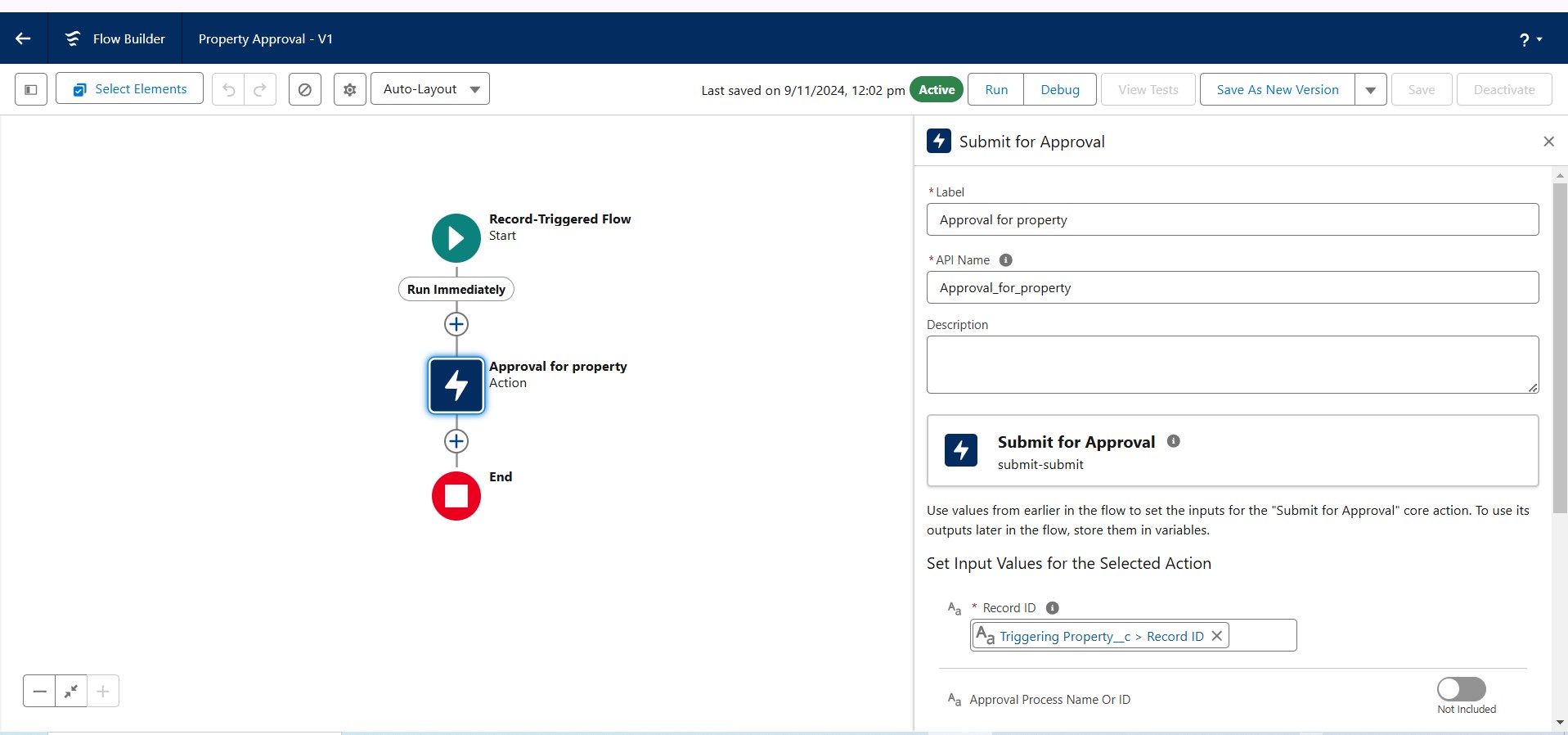


**Step 9. Create A Check Box Field On User.**

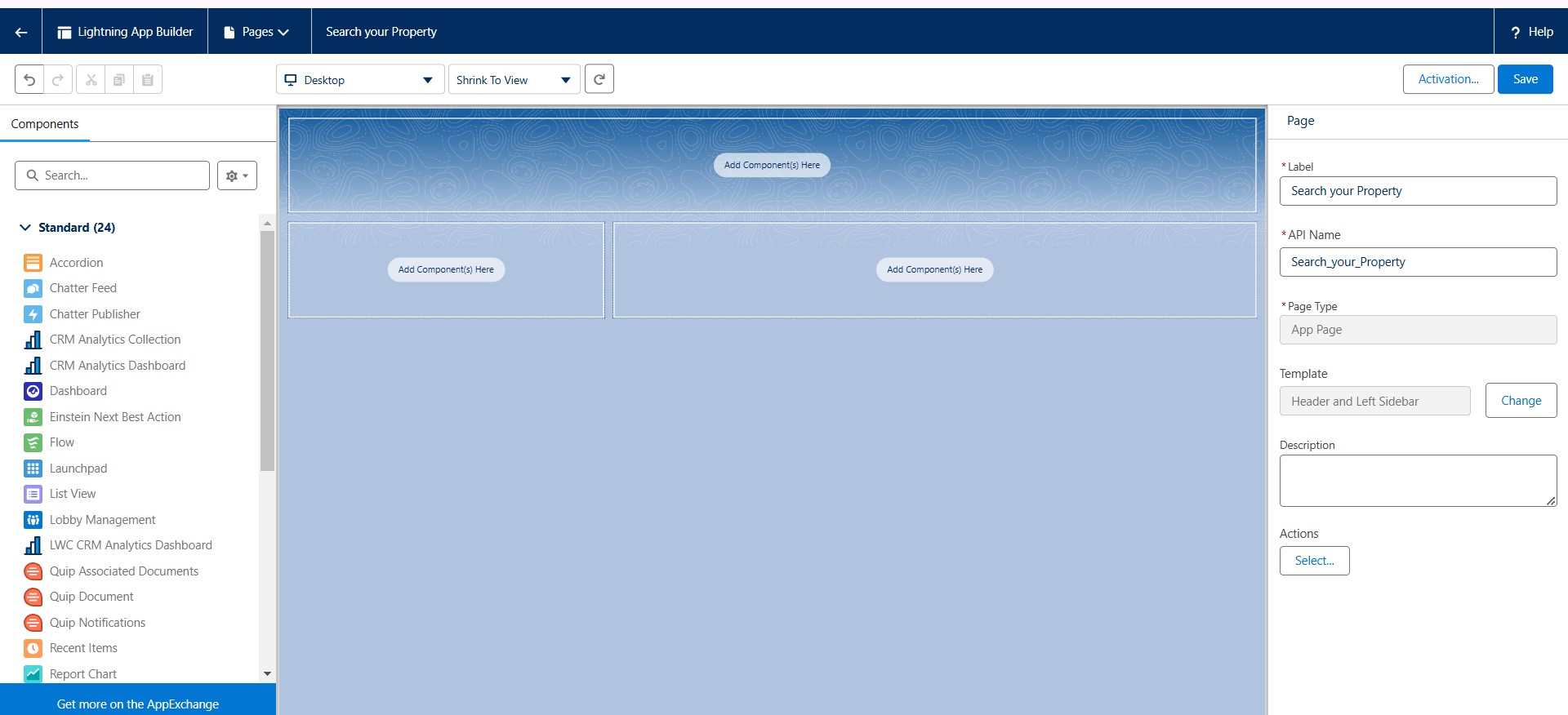
**Step 10. Create Users. (User 1, User 2, User 3, User 4)**

**Step 11. Create An Approval Process For Property Object.**

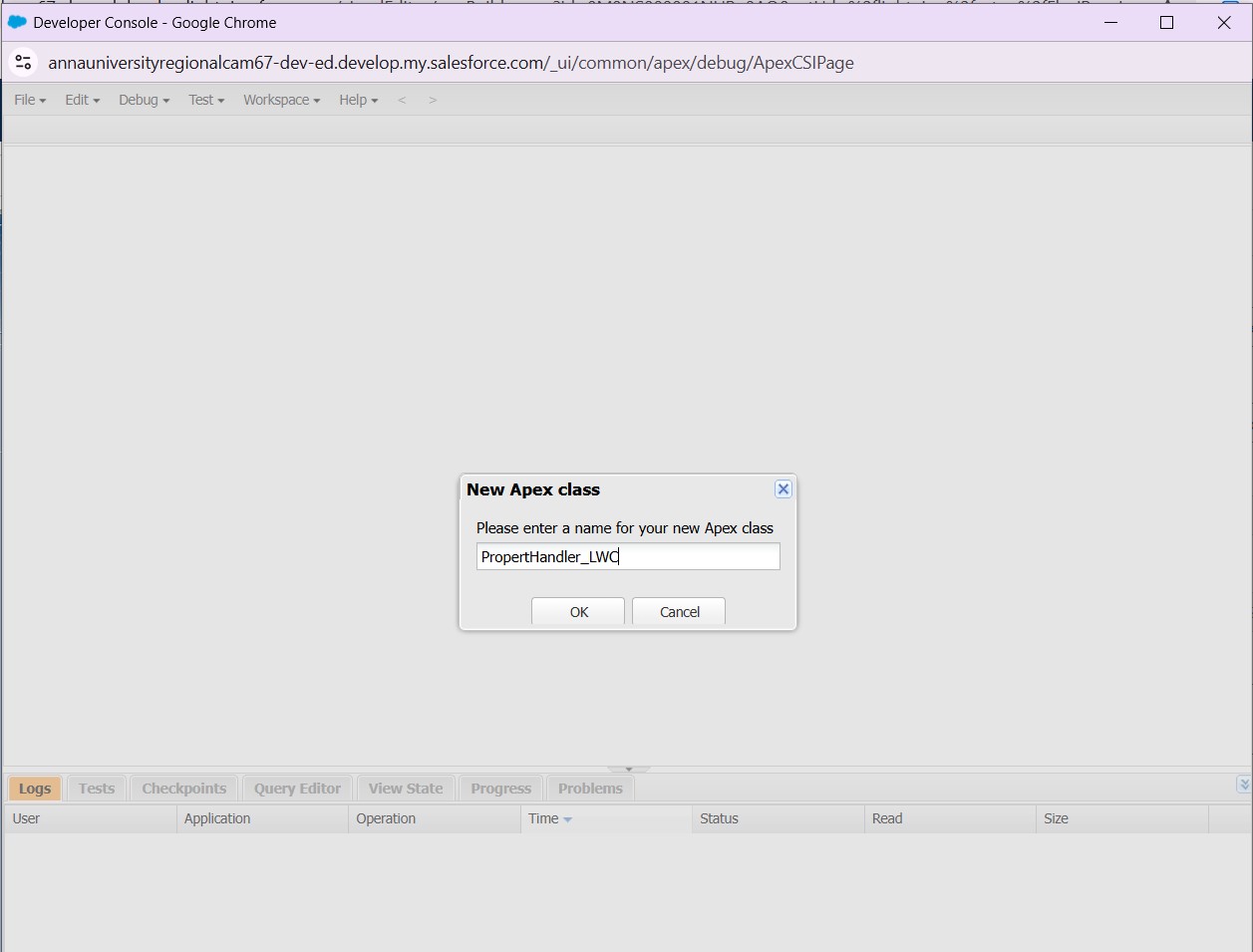
**Step 12. Create A Record Trigger Flow to Submit The Approval Process Automatically.**

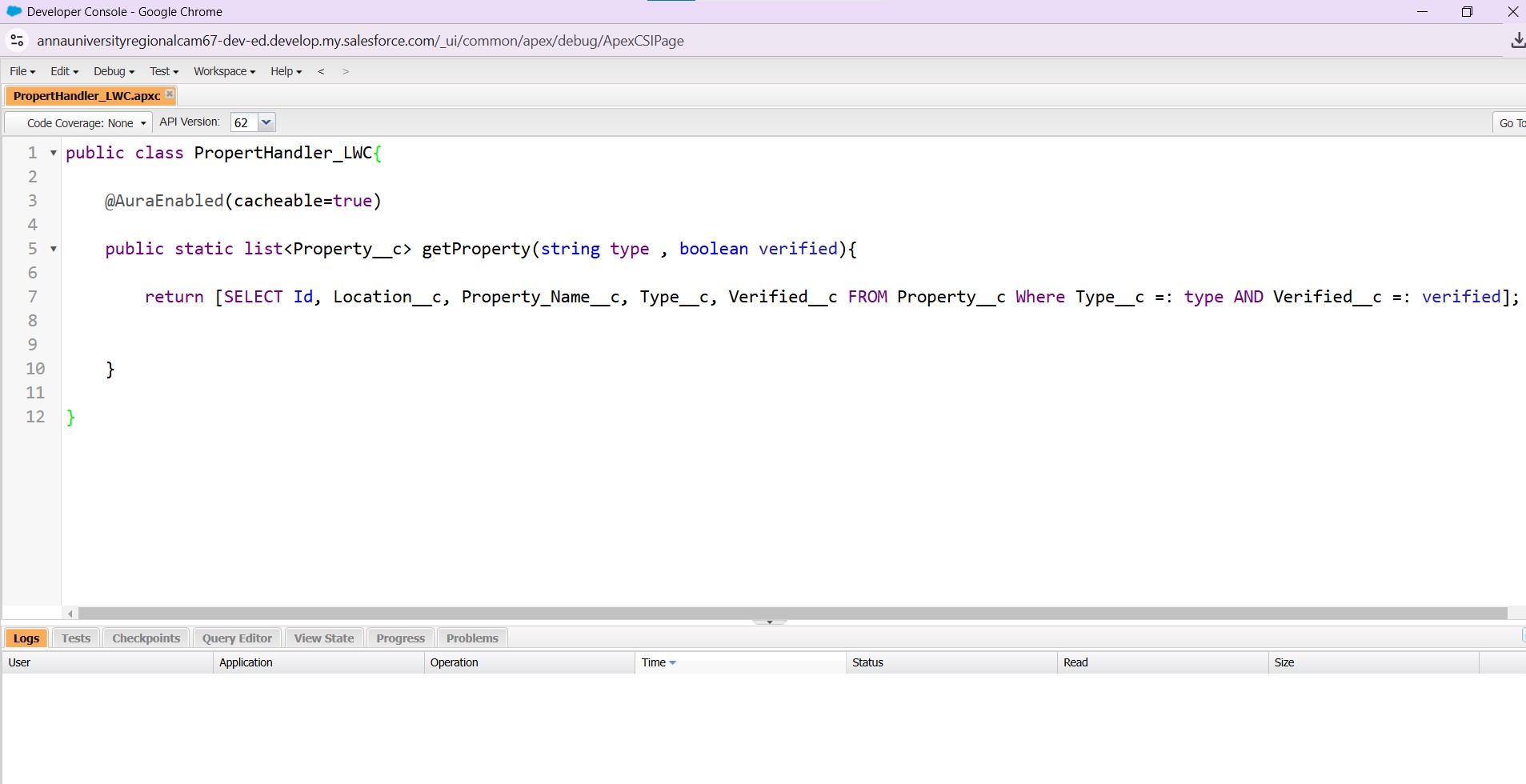


**Step 13. Create An App Page.**

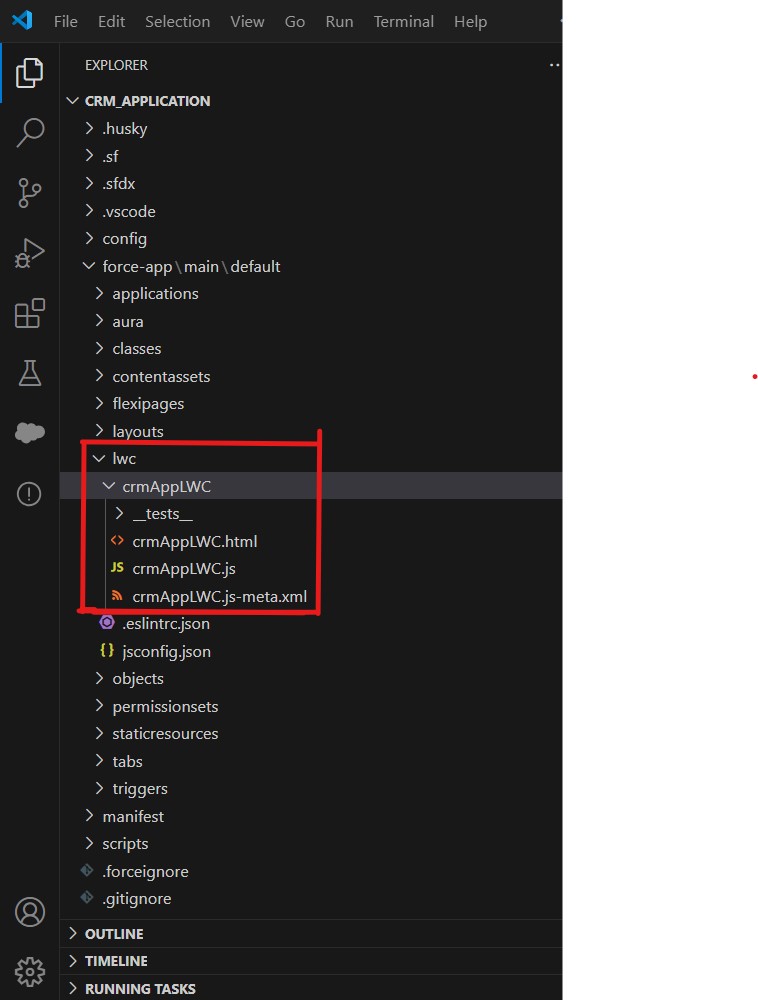


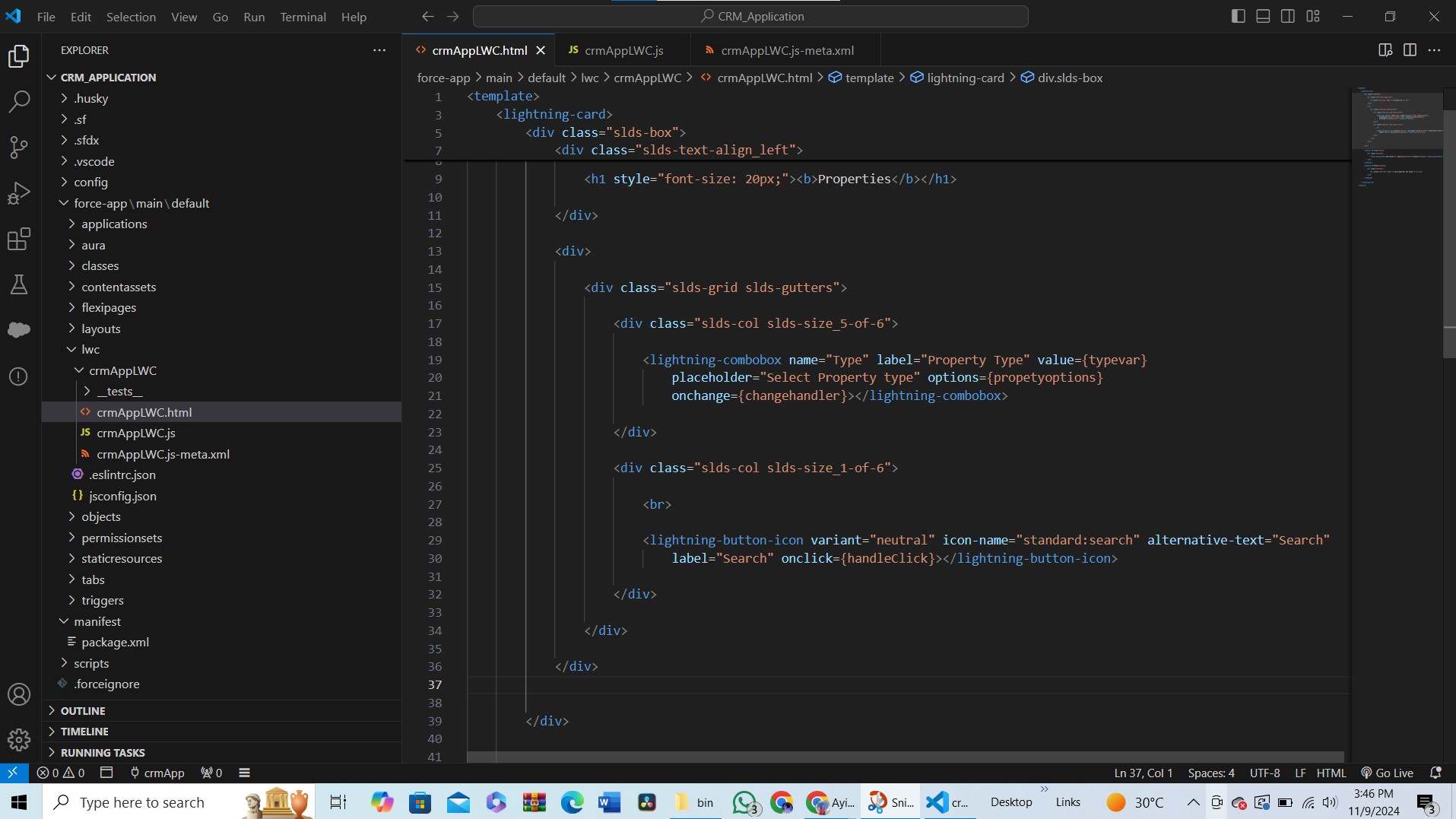
**Step 14. Create A LWC Component.**

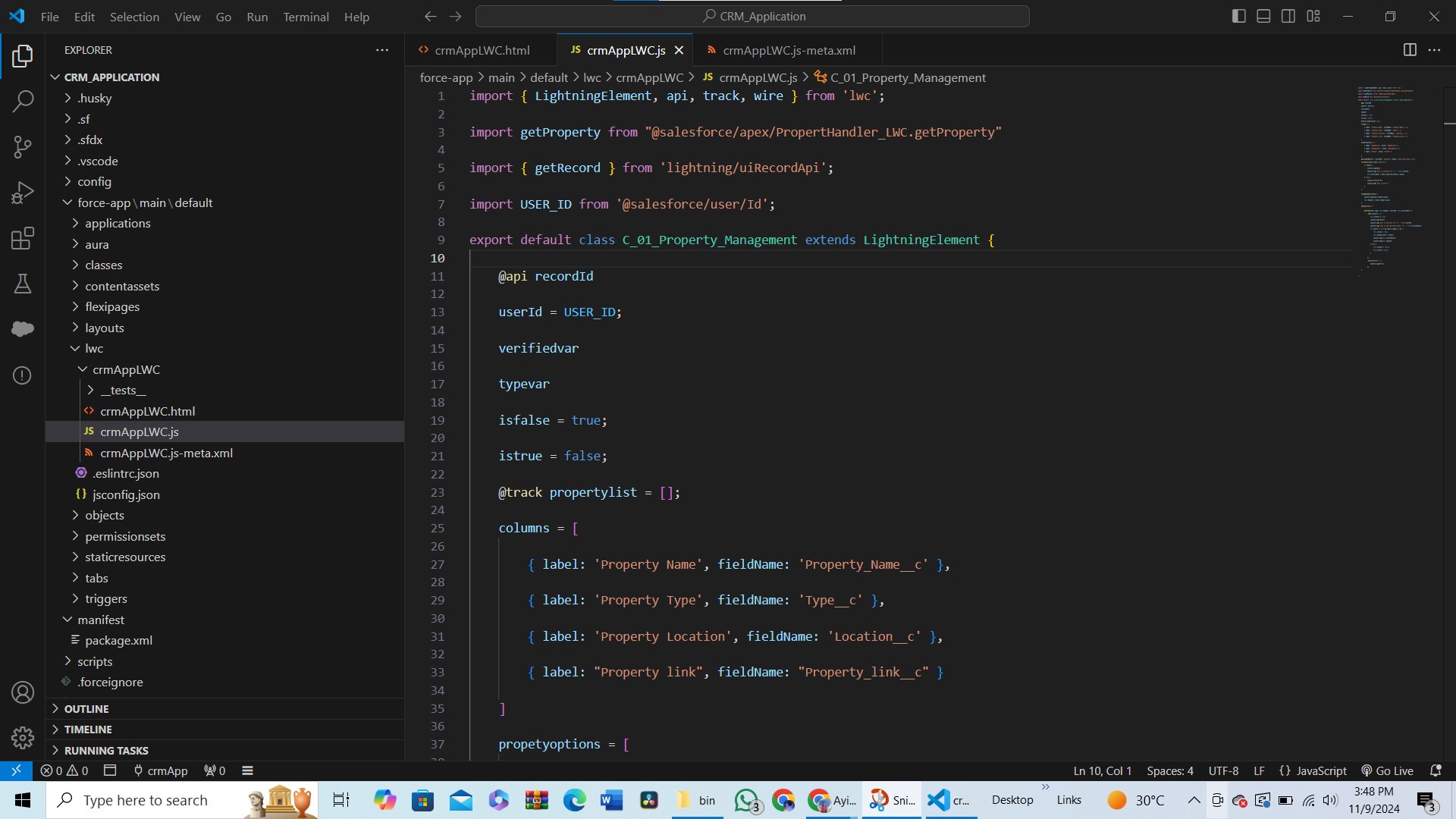
14.1 Create a Apex class namedPropertHandler\_LWC

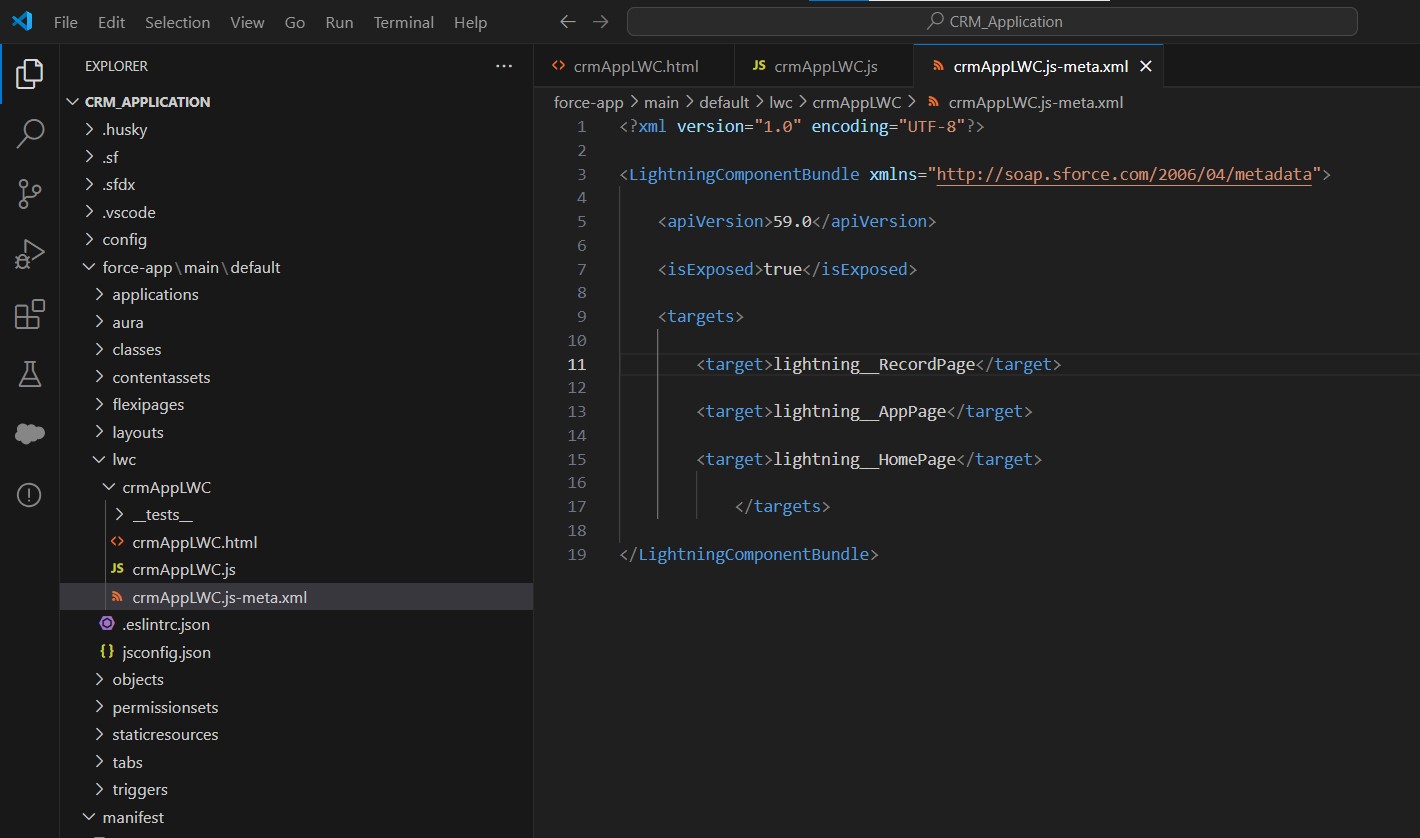
14.2. Apex Code

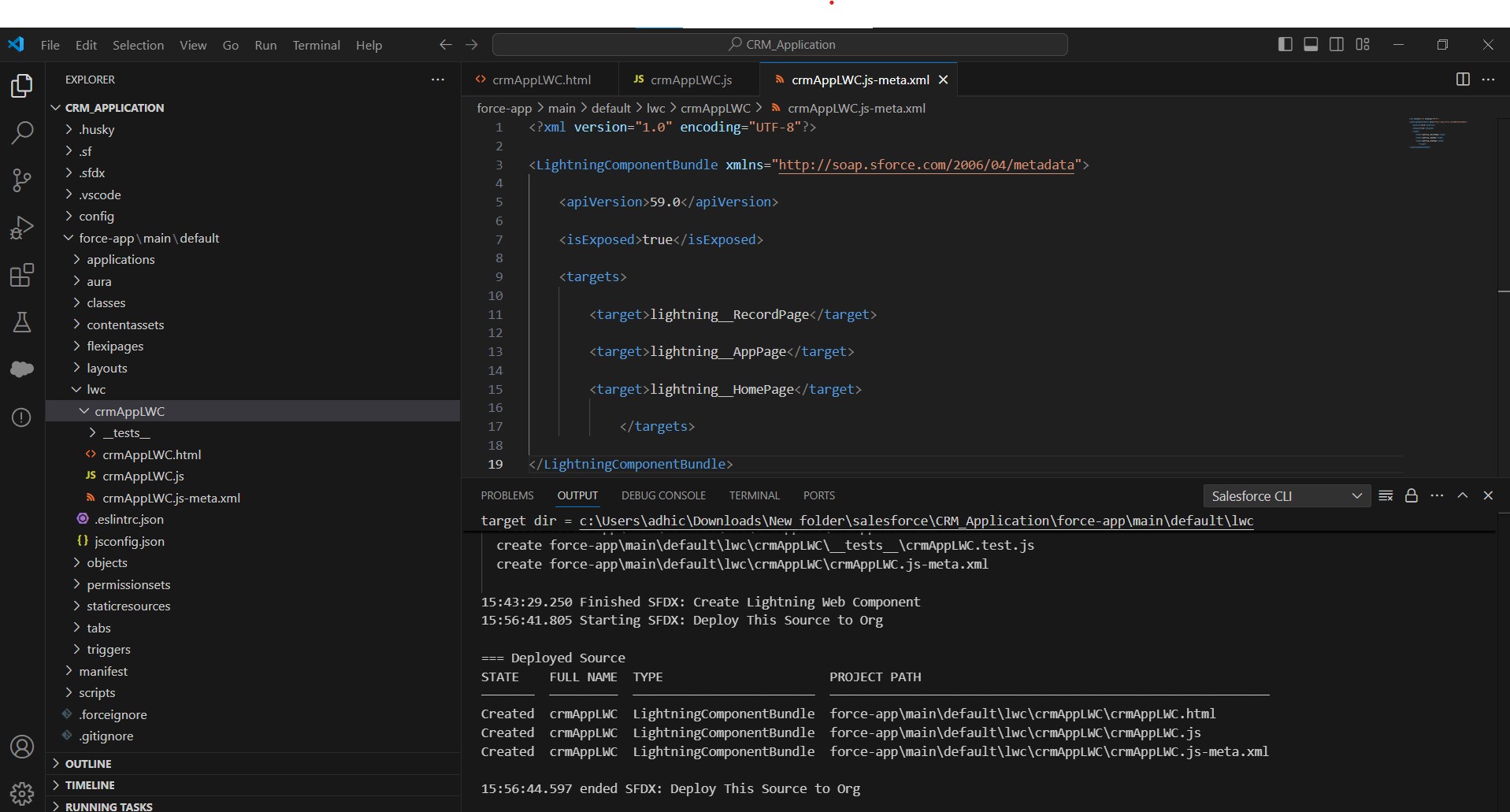
14.3 Folder structure:



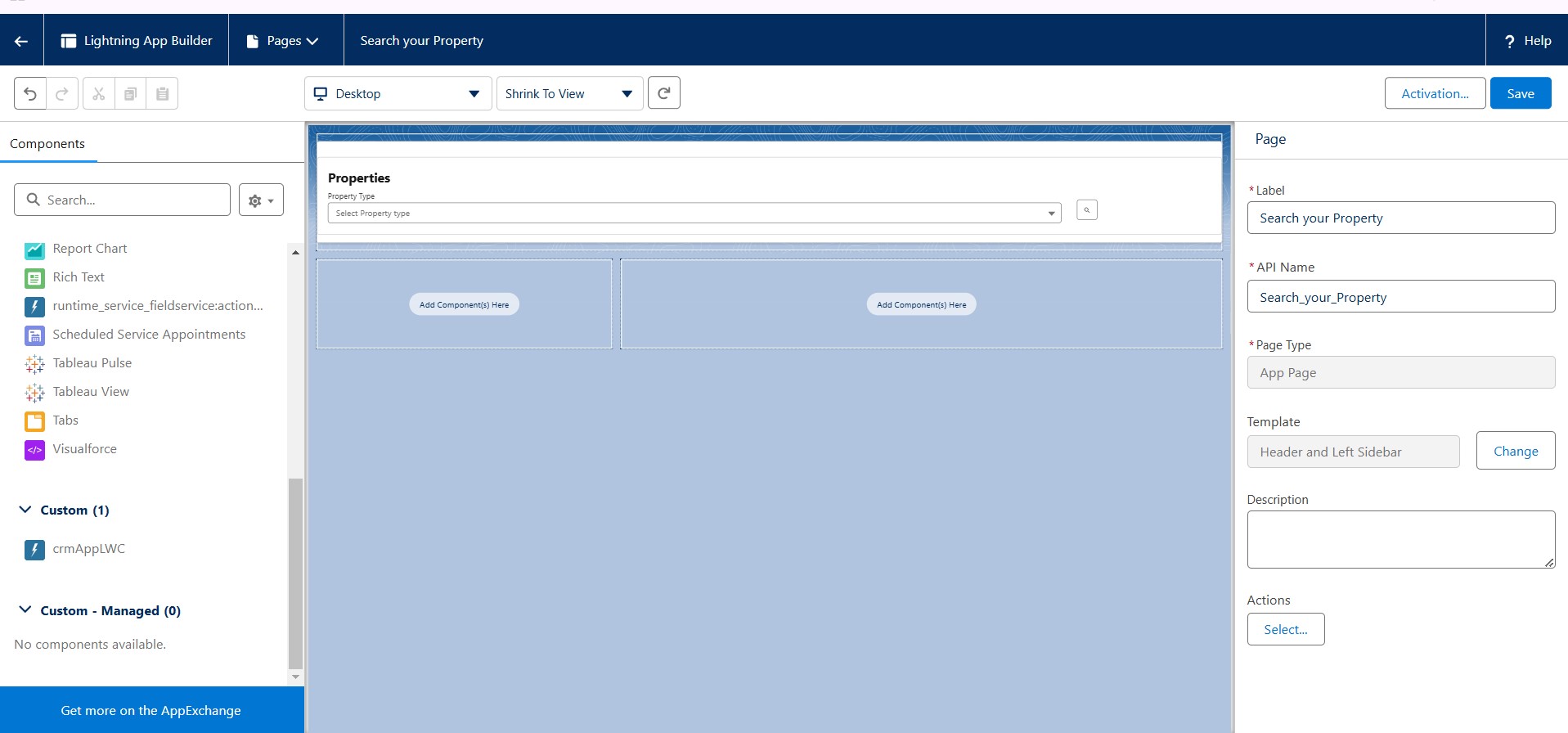
14.4 LWC-Html

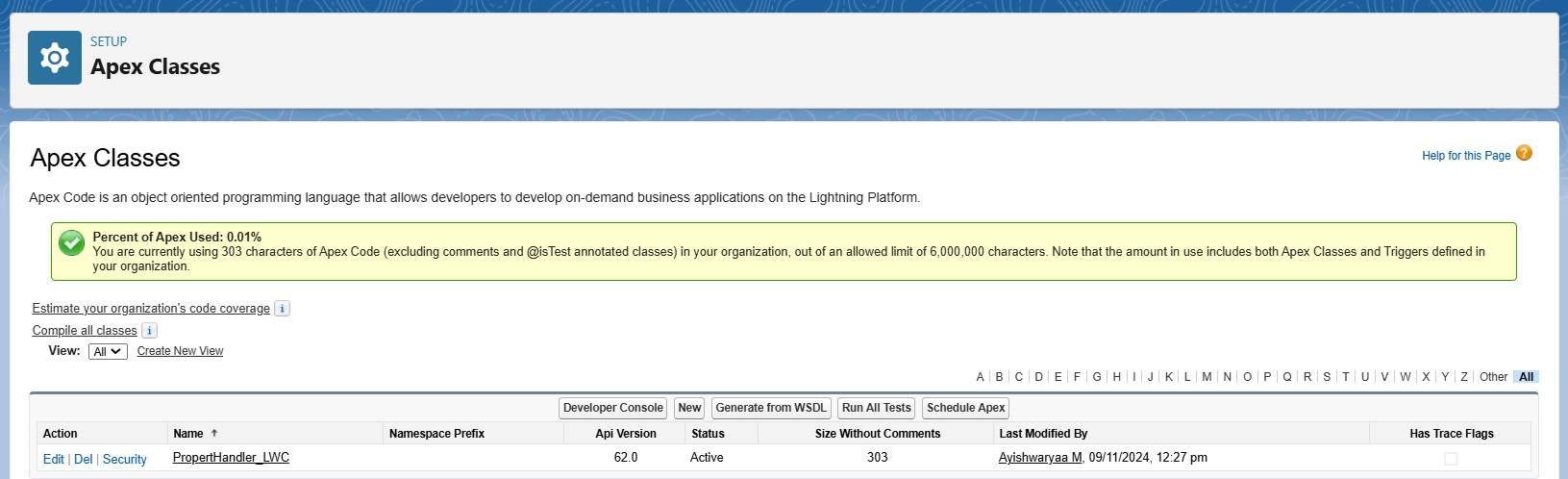
14.5 LWC-js

14.6 LWC- meta

14.7 LWC- Depolyment

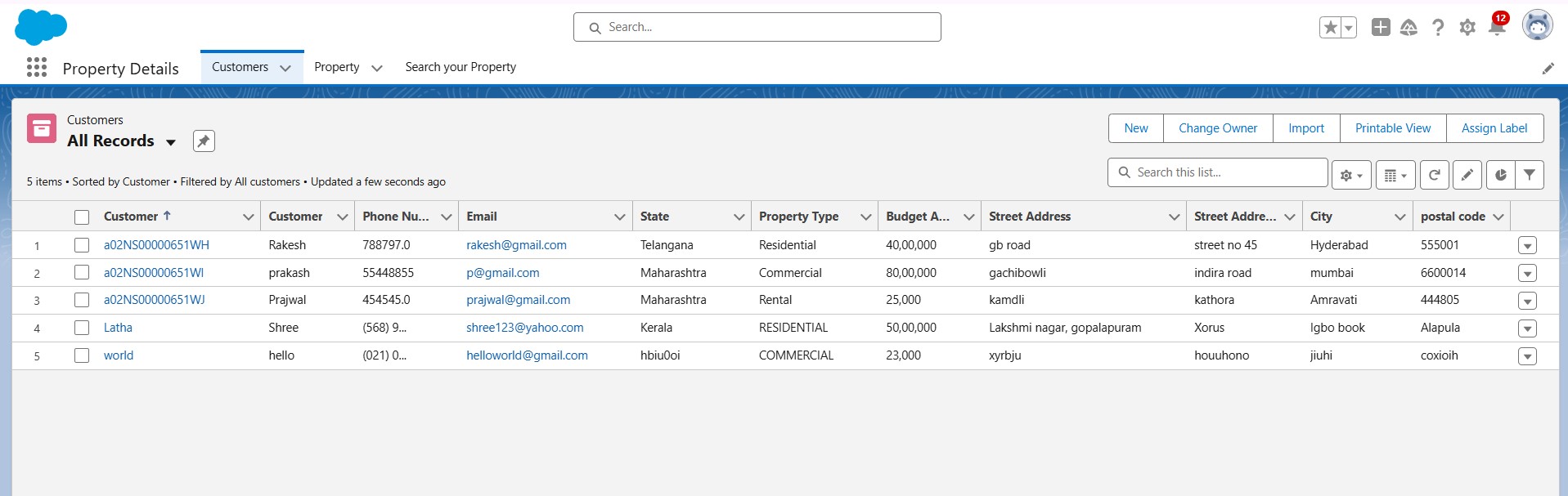
**Step 15 Drag This Component To Your App Page**.

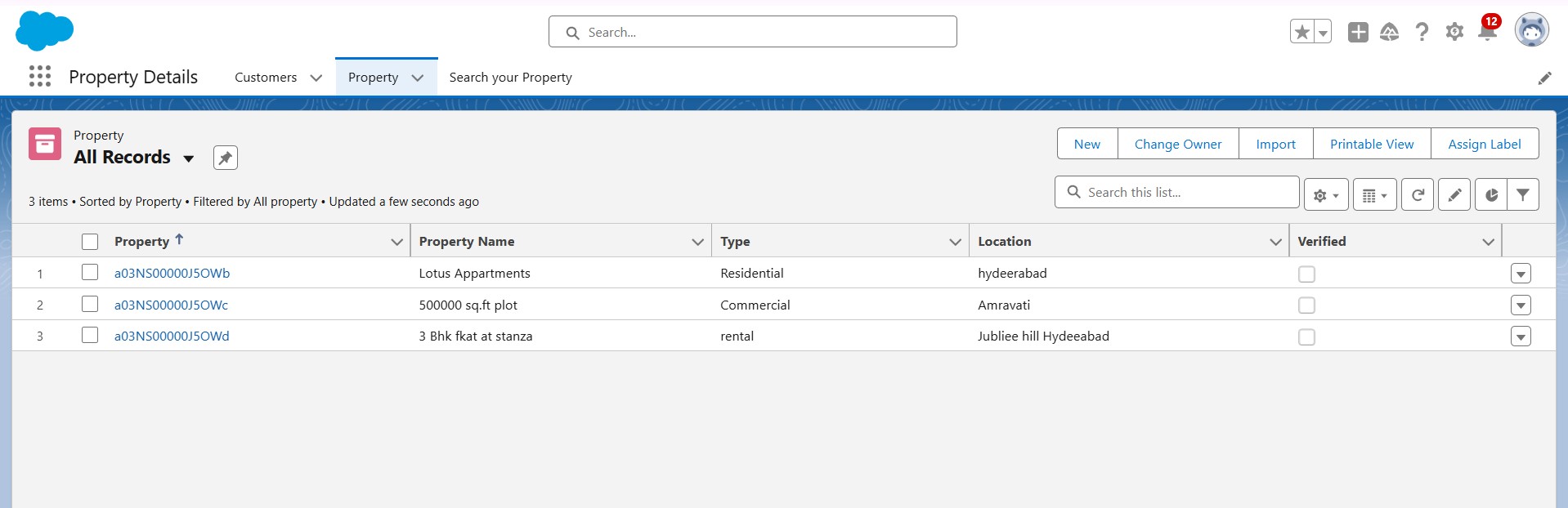


**Step 16 Give Access Of Apex Classes To Profiles**.

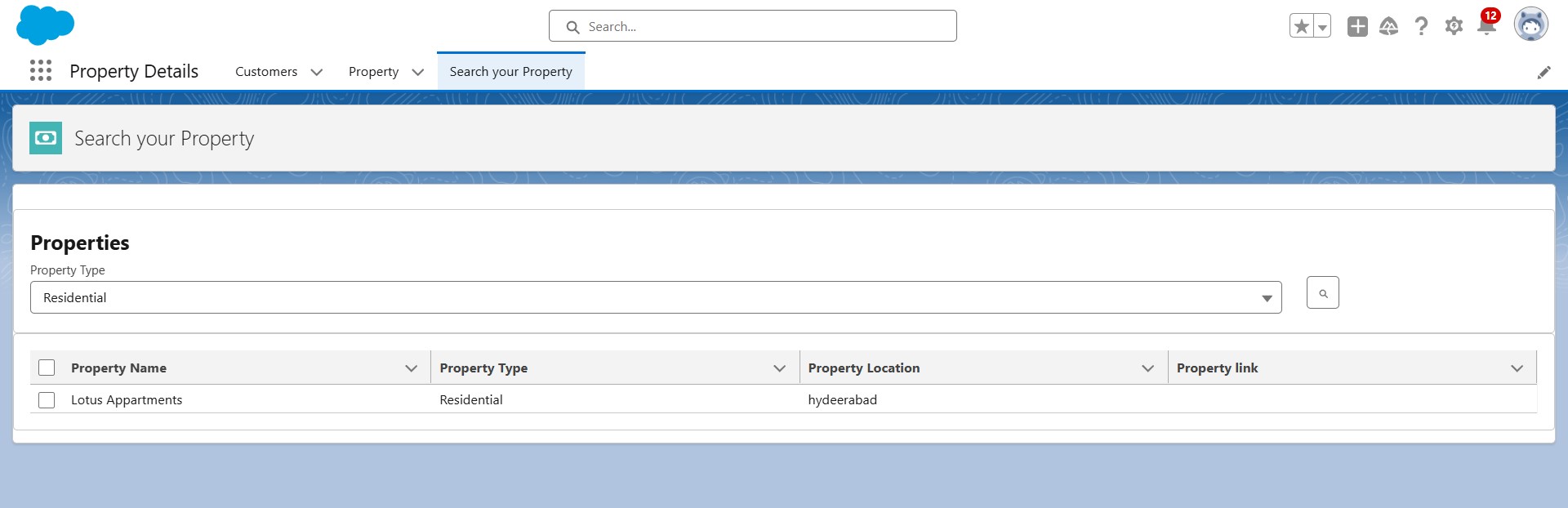
**Step 17 Final Pages**

17.1 Customer page

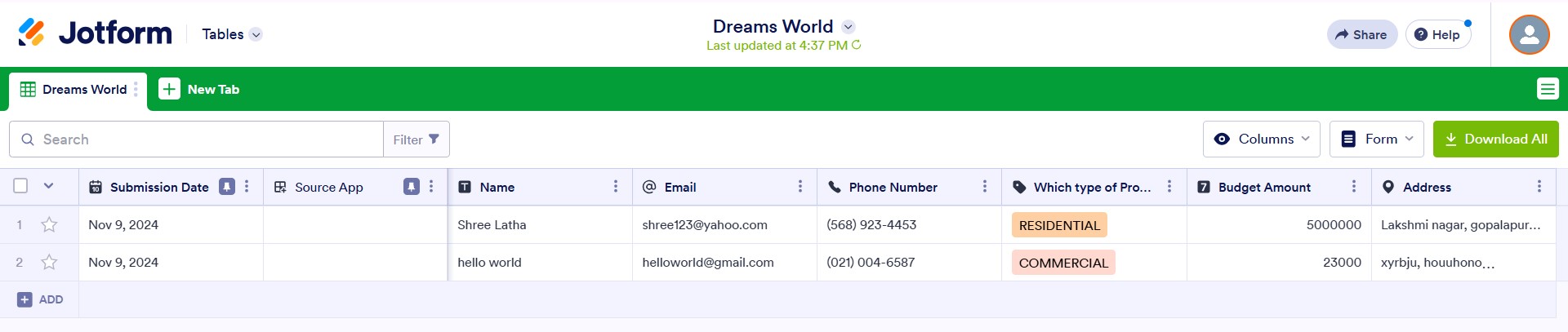


17.2 Property Page

17.3 Search PropertyPage



17.4 Final Jotform Table



**5. Testing and Validation**

**Apex class code:**

public class PropertHandler\_LWC{

@AuraEnabled(cacheable=true)

public static list<Property\_\_c> getProperty(string type , boolean verified){

return [SELECT Id, Location\_c, Property\_Namec, Typec, Verifiedc FROM Propertyc Where Typec =: type AND Verified\_c =: verified];

}

}

**HTML CODE:**

<template>

<lightning-card>

<div class="slds-box">

<div class="slds-text-align\_left">

<h1 style="font-size: 20px;"><b>Properties</b></h1>

</div>

<div>

<div class="slds-grid slds-gutters">

<div class="slds-col slds-size\_5-of-6">

<lightning-combobox name="Type" label="Property Type" value={typevar} placeholder="Select Property type"

options={propetyoptions} onchange={changehandler}></lightning-combobox>

</div>

<div class="slds-col slds-size\_1-of-6">

<br>

<lightning-button-icon variant="neutral" icon-name="standard:search" alternative-text="Search"

label="Search" onclick={handleClick}></lightning-button-icon>

</div>

</div>

</div>

</div>

<template if:true={istrue}>

<div class="slds-box">

<lightning-datatable key-field="id" data={propertylist} columns={columns}></lightning-datatable>

</div>

</template>

<template if:false={isfalse}>

<div class="slds-box">

<div style="font-size: 15px;"><b>No properties Are Found !!</b></div>

</div>

</template>

</lightning-card>

</template>

**JAVASCRIPT CODE:**

import { LightningElement, api, track, wire } from 'lwc';

import getProperty from "@salesforce/apex/PropertHandler\_LWC.getProperty"

import { getRecord } from 'lightning/uiRecordApi';

import USER\_ID from '@salesforce/user/Id';

export default class C\_01\_Property\_Management extends LightningElement {

@api recordId

userId = USER\_ID;

verifiedvar

typevar

isfalse = true;

istrue = false;

@track propertylist = [];

columns = [

{ label: 'Property Name', fieldName: 'Property\_Name\_\_c' },

{ label: 'Property Type', fieldName: 'Type\_\_c' },

{ label: 'Property Location', fieldName: 'Location\_\_c' },

{ label: "Property link", fieldName: "Property\_link\_\_c" }

]

propetyoptions = [

{ label: "Commercial", value: "Commercial" },

{ label: "Residential", value: "Residential" },

{ label: "rental", value: "rental" }

]

@wire(getRecord, { recordId: "$userId", fields: ['User.Verified\_\_c'] })

recordFunction({ data, error }) {

if (data) {

console.log(data)

console.log("This is the User Id ---> "+this.userId);

this.verifiedvar = data.fields.Verified\_\_c.value;

} else {

console.error(error)

console.log('this is error')

}

}

changehandler(event) {

console.log(event.target.value);

this.typevar = event.target.value;

}

handleClick() {

getProperty({ type: this.typevar, verified: this.verifiedvar })

.then((result) => {

this.isfalse = true;

console.log(result)

console.log('This is the User id ---> ' + this.userId);

console.log('This is the verified values ---> ' + this.verifiedvar);

if (result != null && result.length != 0) {

this.istrue = true;

this.propertylist = result;

console.log(this.verifiedvar);

console.log(this.typevar)

} else {

this.isfalse = false;

this.istrue = false;

}

})

.catch((error) => {

console.log(error)

})

}

}

**META FILE CODE:**

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

<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">

<apiVersion>59.0</apiVersion>

<isExposed>true</isExposed>

<targets>

<target>lightning\_\_RecordPage</target>

<target>lightning\_\_AppPage</target>

<target>lightning\_\_HomePage</target>

</targets>

</LightningComponentBundle>

**6.** **Key Scenarios Addressed by Salesforce in the Implementation Project**

* **Client Onboarding and Document Upload:** Use automated workflows to guide users through the document upload process during client onboarding, ensuring all necessary documents are available and correctly filed.
* **Property Document Tracking:** Track document statuses, expiration dates, and update requirements for all properties, enabling proactive management and client notifications.
* **Compliance Monitoring:** Ensure data is handled in compliance with legal standards through validation rules, encryption, and access control, particularly for sensitive documents.
* **Automated Reminders and Notifications**:Set up automated alerts and reminders for expiring contracts, missing documents, or required updates to improve proactive client management.

**7. Conclusion**

**Summary of Achievements:**

* Through the development of the **CRM Application for Document Management in Real Estate**, we have created a solution that centralizes document handling and enhances client management in Salesforce. By implementing a structured data model, automating workflows, and adhering to compliance requirements, the project has achieved its goals of streamlining document management, increasing operational efficiency, and improving the overall user experience for the real estate team. This project will enable the organization to manage client and property documentation more effectively, supporting long-term business objectives and fostering a more organized, client-centered approach to real estate management.