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

**Department of Computer Science and Engineering**

|  |  |  |
| --- | --- | --- |
| NM Id | (Team): | 4224582A19853839BBB4409FBE0BD706 |

**Team Members**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **NM-ID** |
| 1 | Nandhini T | 4224582A19853839BBB4409FBE0BD706 |
| 2 | Carmal princy A | 4801F1749A0B590BA5942A3FC1B90254 |
| 3 | Ashiritha S | 4A99E958001A914BDC34EC698F983844 |
| 4 | Kirthana M | D5F1604B6269EE7BC174DAD602EB5FDA |

**Github link :** <https://github.com/Nandhini-nan/CRM-Project-Naan-Mudhalvan>

**Demo video link :** <https://youtu.be/WFnaaix2ju0?si=NkSIcOpVJd3aKGFm>

**TrailHead link :**

[https://www.salesforce.com/trailblazer/xj2khy4jl659txp2y8](https://www.salesforce.com/trailblazer/xj2khy4jl659txp2y8%20)

**A CRM Application to Handle the Clients and their Property Related Requirements**

# 1.Project Overview

This project focuses on creating a Customer Relationship Management (CRM) application to automate and streamline client interactions and property management processes. Using Salesforce, the project integrates features like Jotform-based data collection, role-based access management, property approval workflows, and dynamic property displays using Lightning Web Components (LWC).

The goal is to enhance operational efficiency, improve the customer experience, and provide tailored solutions for property-related requirements.

# 2.Objectives

**Business Goals:**

* Automate the process of capturing customer information.
* Categorize clients and provide personalized property recommendations.
* Improve the accuracy and efficiency of property approvals.

**Specific Outcomes:**

* Integrated a Jotform with Salesforce for automatic record creation.
* Developed roles and profiles for access control.
* Created a property approval process.
* Built a dynamic LWC to display verified and non-verified properties.

**3.Salesforce key features and concepts utilized**

1. Real Time Salesforce Project

2. Object & Fields

3. Integration Through Jotform

4. Roles

5. Application Management

6. Profiles

7. User Management

8. Approval Process

9. Flows

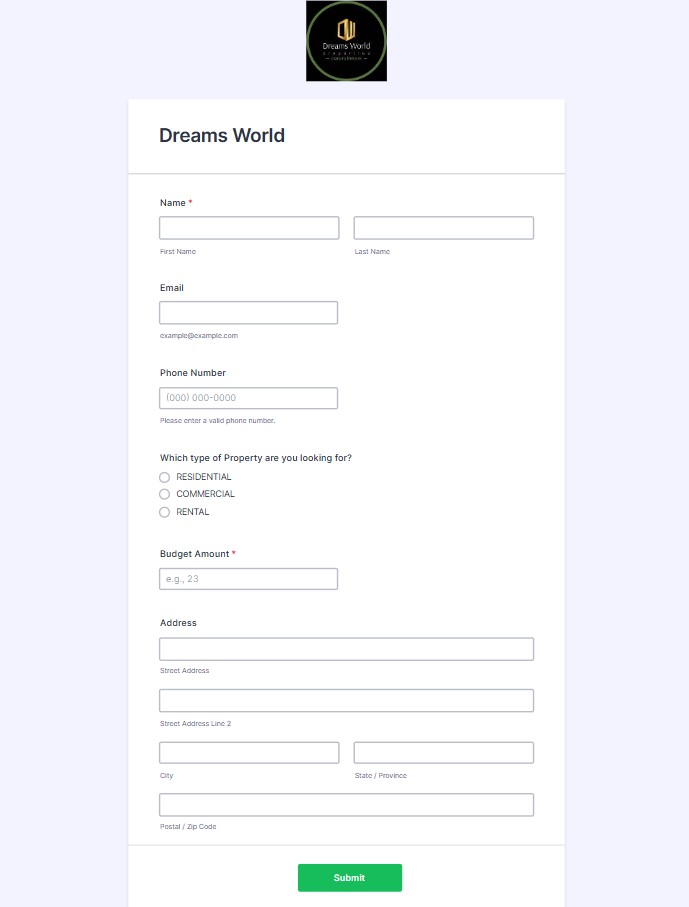
10. LWC Components

**4. Detailed Steps to Solution Design**

# Project Milestones

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

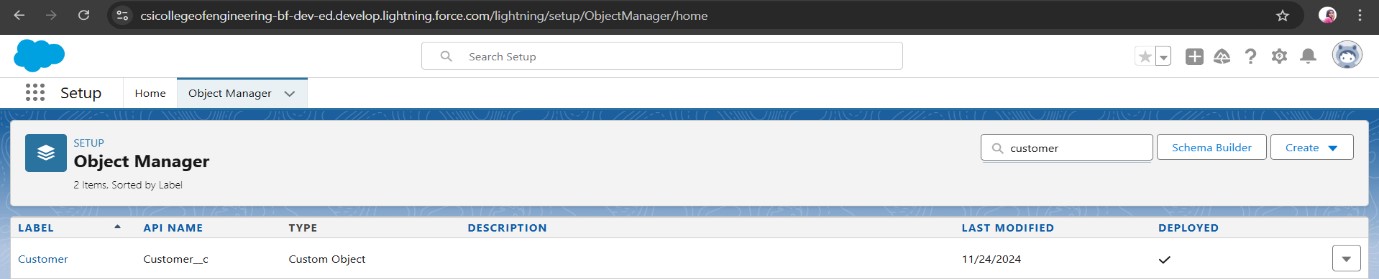
* Create a form to collect customer data and link it to Salesforce.
* Automatically generate customer records.
* Go to [https://www.jotform.com/.](https://www.jotform.com/)
* Create an account using your credentials.
* Click on the “Start from Scratch” button to create a form as shown in the figure below.
* Click on the “Start from Scratch” button to create a form
* Click on “Classic form” button as shown in the figure below.
* Click on the “Add Form Element” button as shown in the below figure.

 S

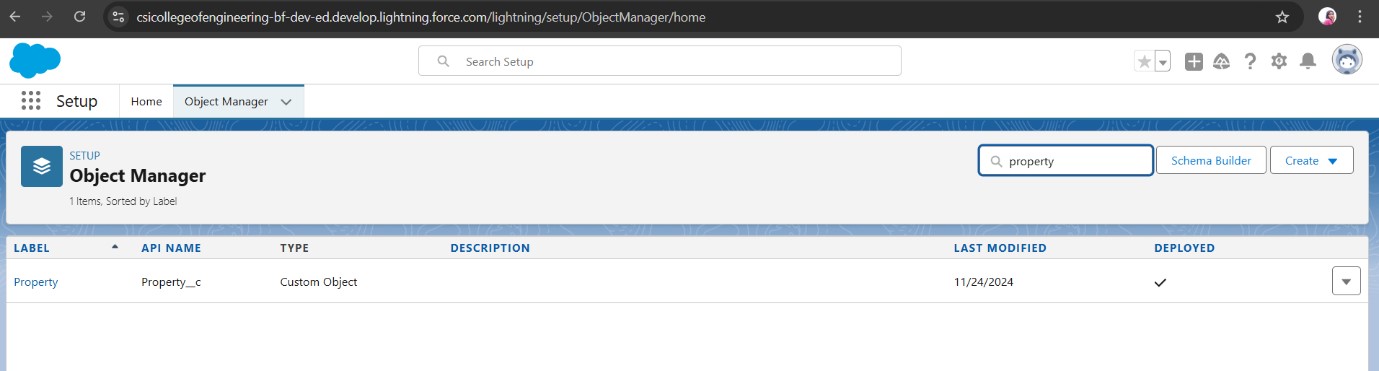
**Milestone 2: Create Objects from Spreadsheet.**

Directly Creating Objects from Spreadsheet in Salesforce

* Create Customer Object

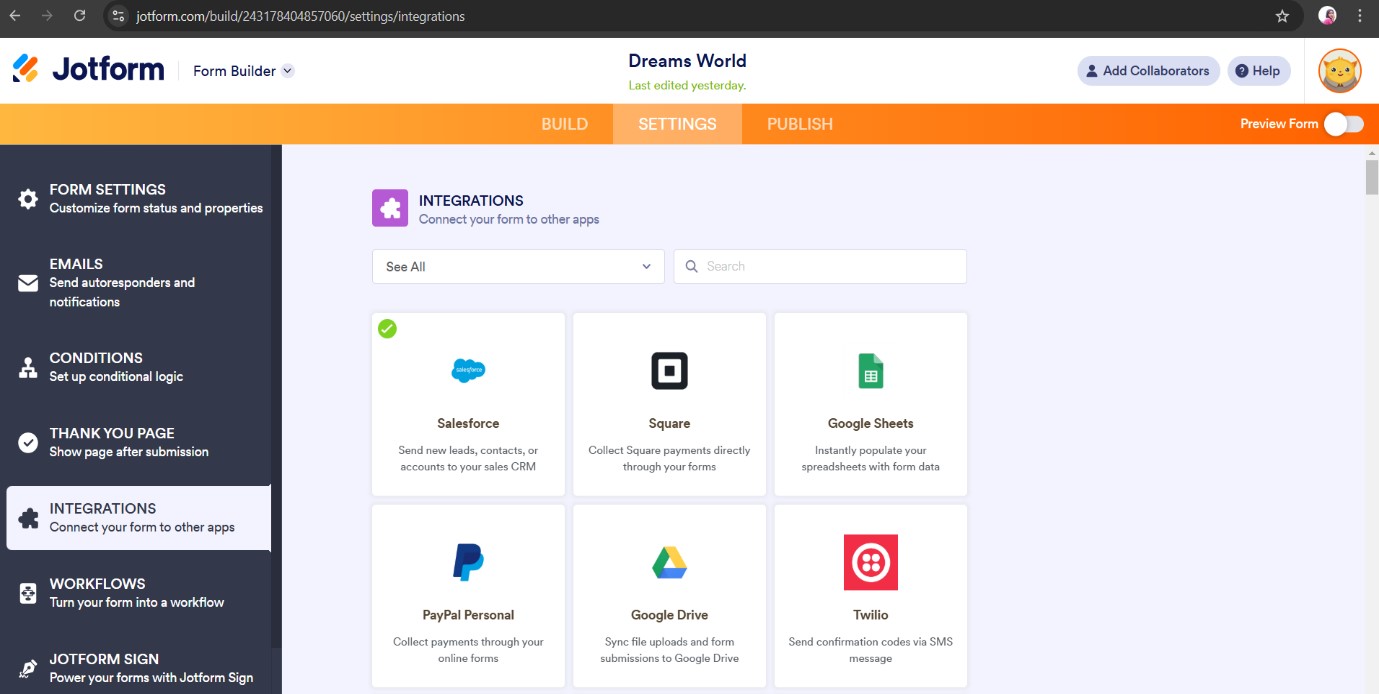


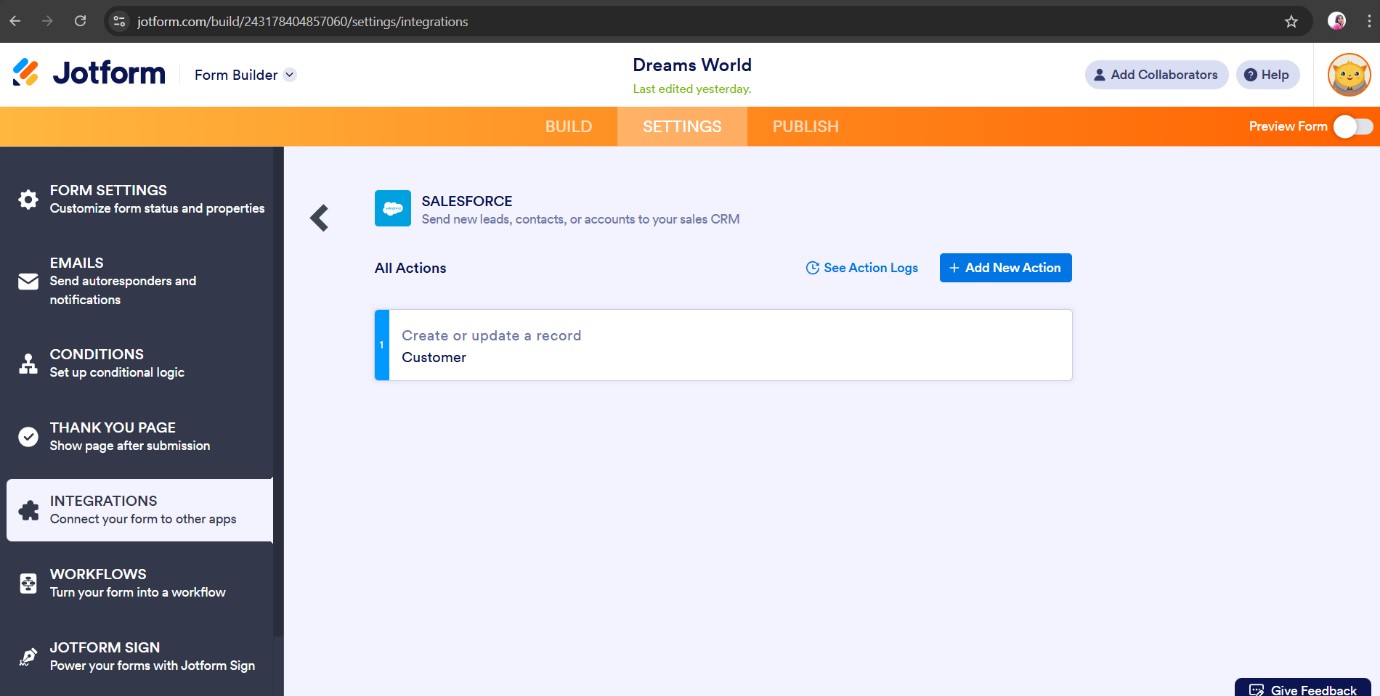
* Create Property object



# Milestone 3: Integrate Jotform with Salesforce Platform

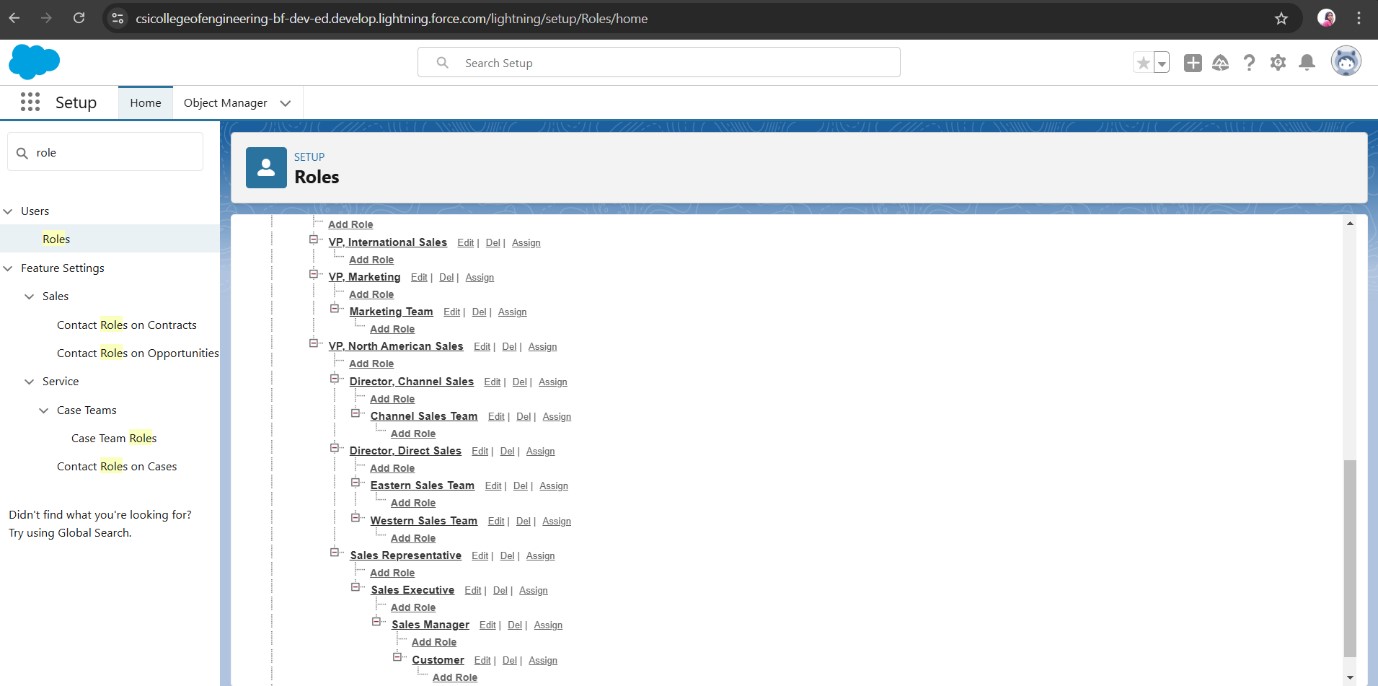
Integrate jotform with salesforce to create customer records





# Milestone 4: Create Roles

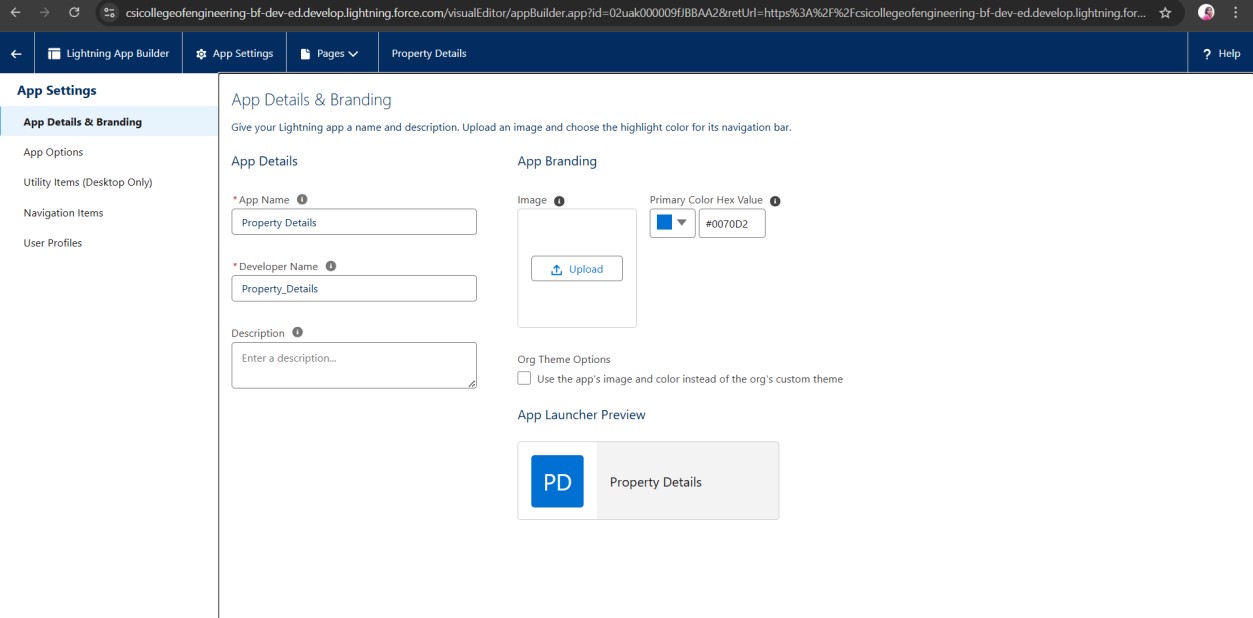
Create and assign roles for different user levels Sales Executive, Manager, Customer



# Milestone 5: Create a Property Details App

Create a Lightning App named *Property Details* in Salesforce, integrating the

*Customer* and *Property* objects. Assign the System Administrator profile for app access and management**.**

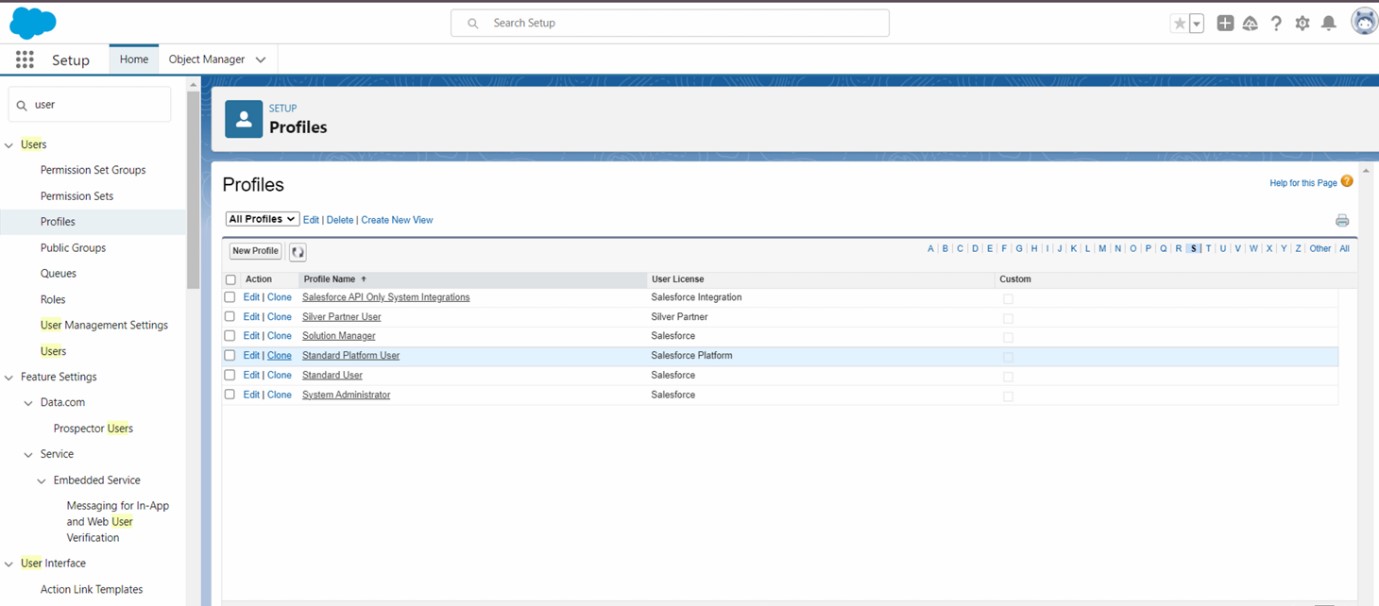


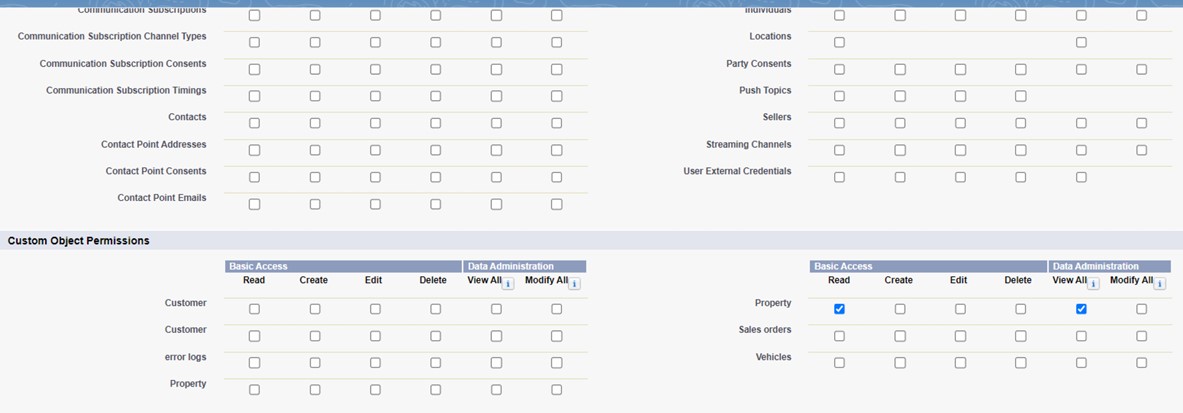
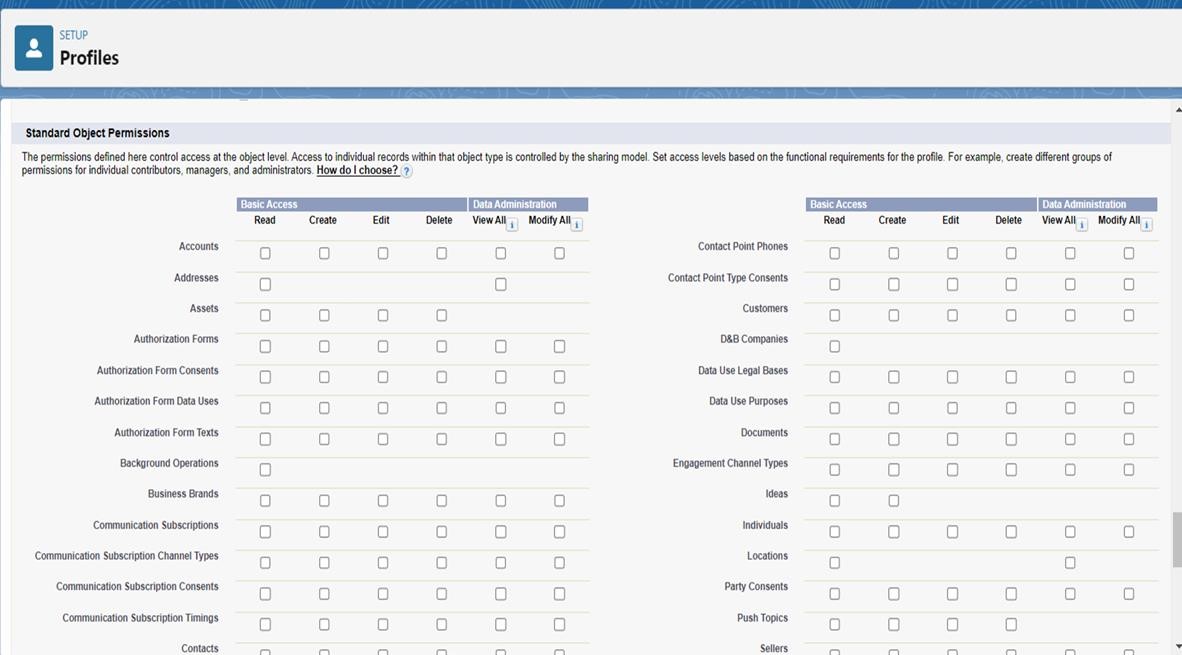
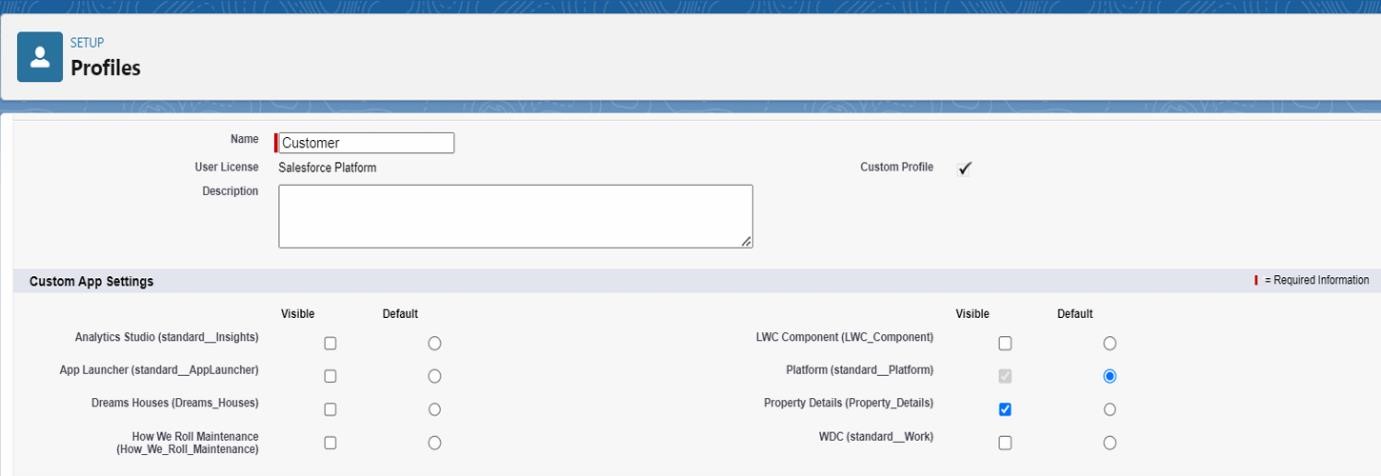
# Milestone 6: Create Profiles

Clone the Salesforce Platform User profile to create Customer and Manager profiles.

Customer:

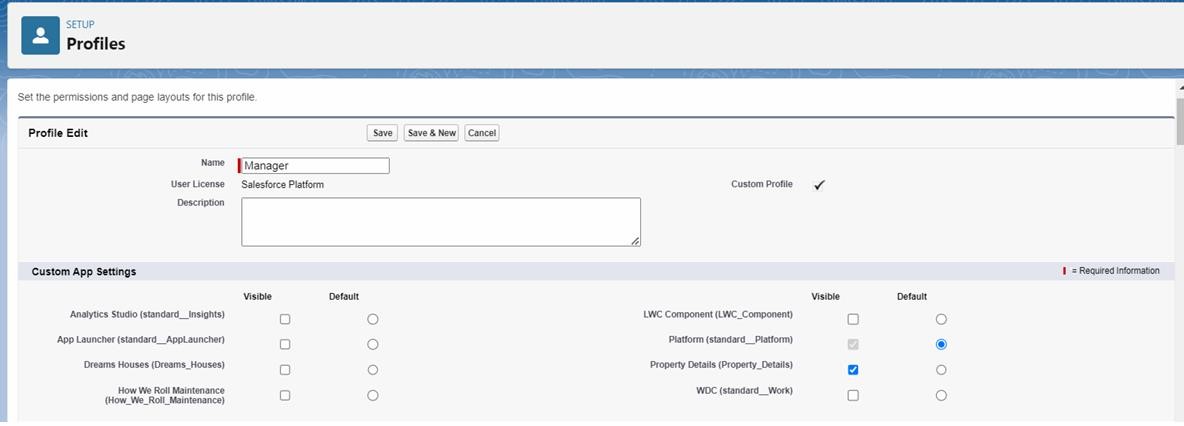
For Customer, restricted access to only the *Property Details* app, removed standard object permissions, and allowed read-only access to the *Property* object.

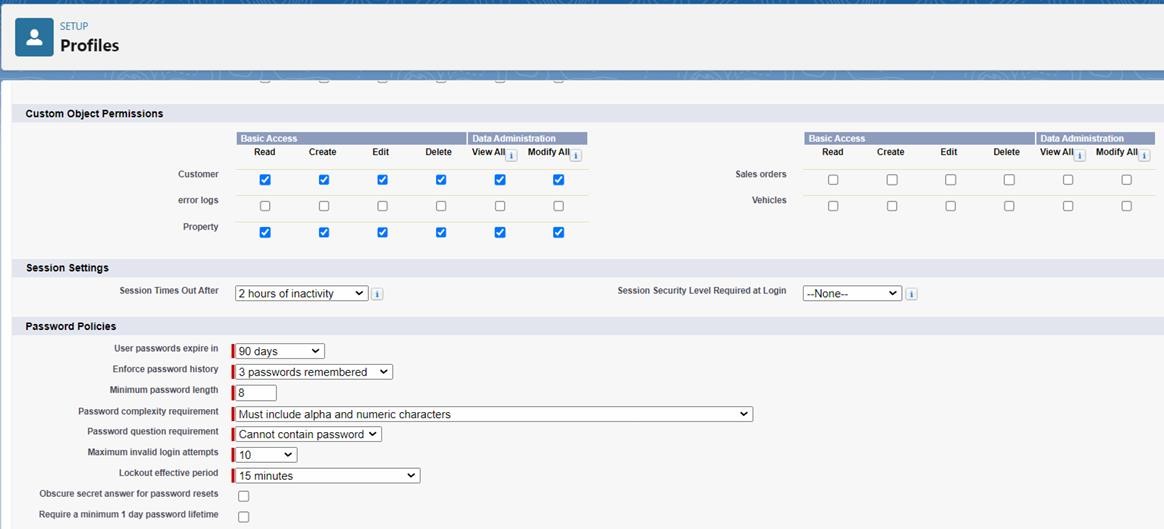




Manager:

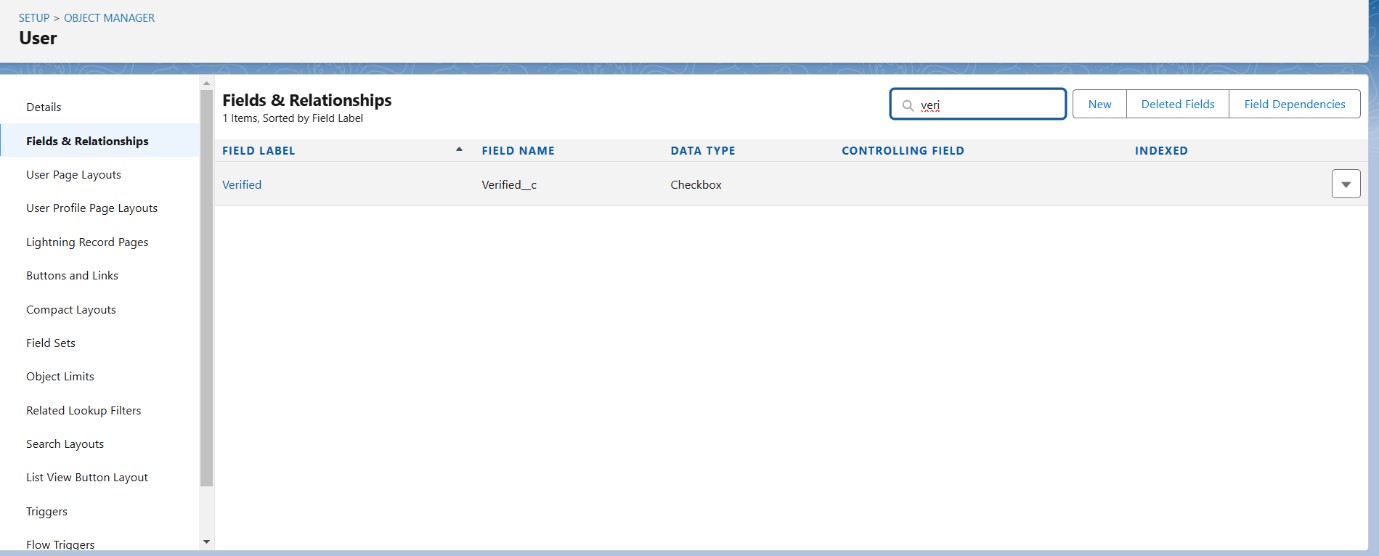
For Manager, restricted access to the *Property Details* app, removed standard object permissions, and granted full access (modify all) to both *Property* and *Customer* objects.

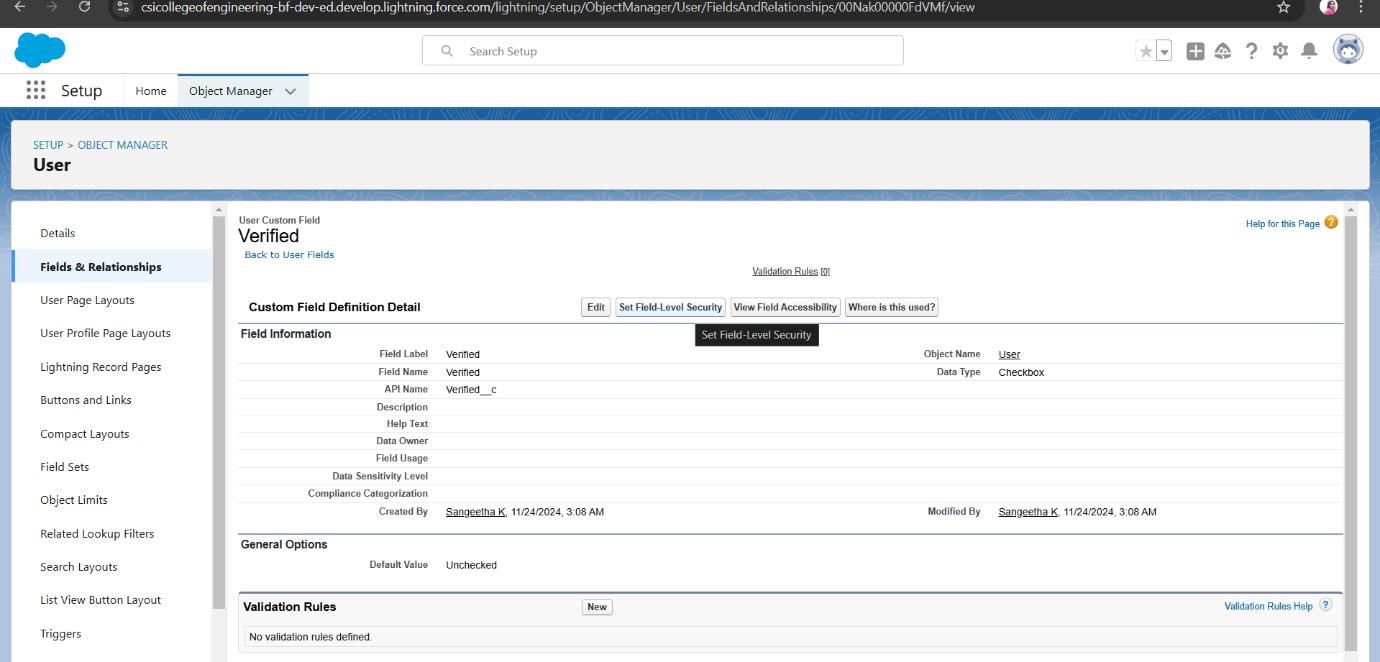




# Milestone 7: Create a CheckBox field on user

Create a new custom field "Verified" in the User object under Fields and Relationships with the data type set to Checkbox.

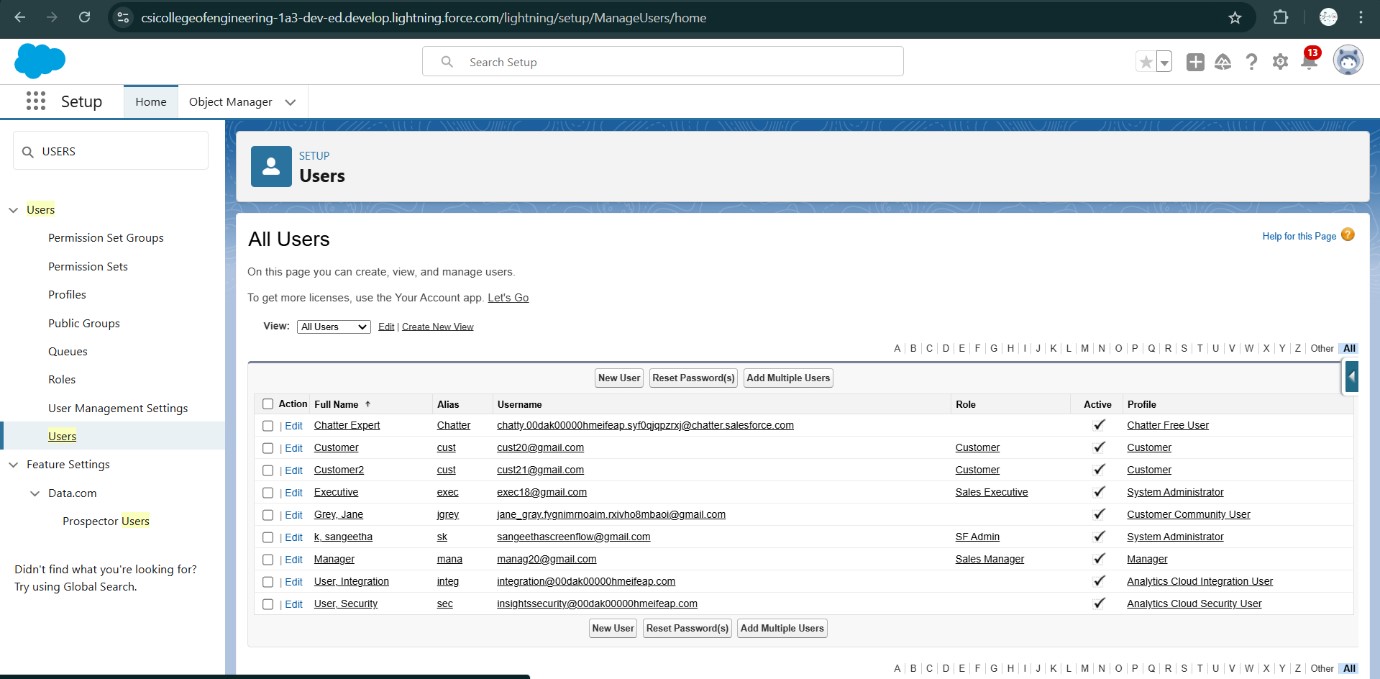




# Milestone 8: Create Users

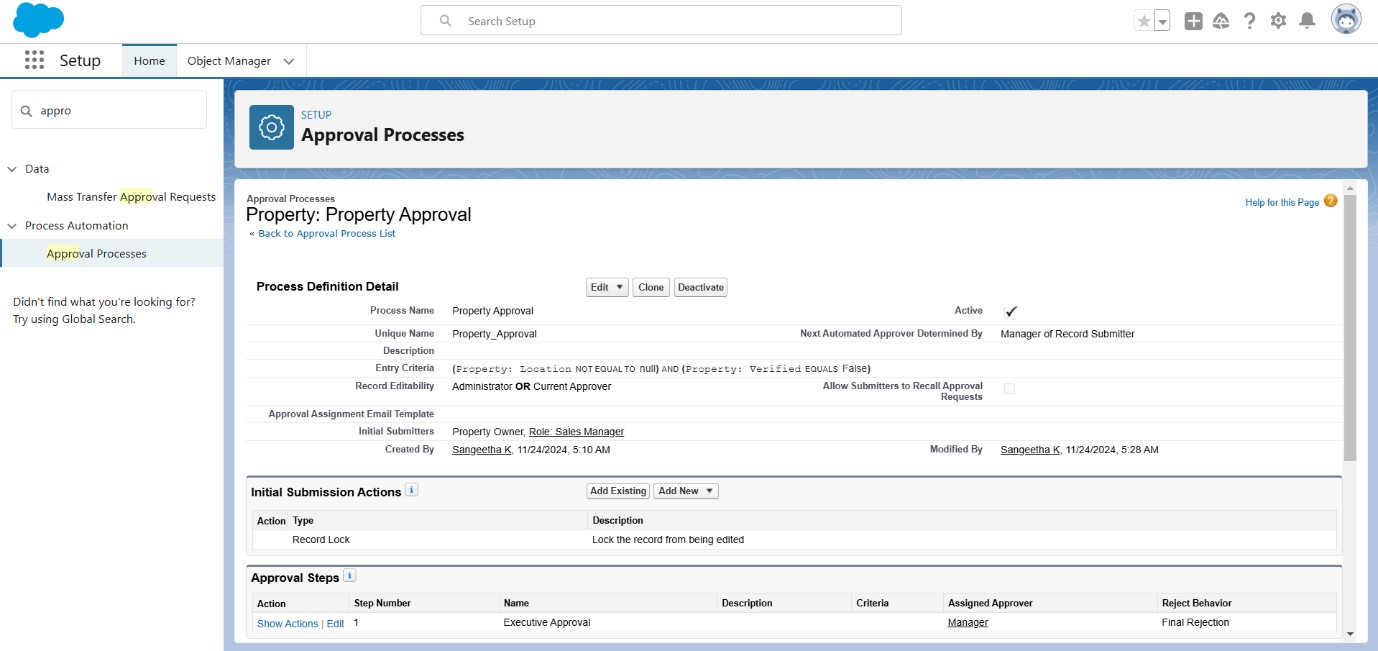
Create four users with different roles and profiles in Salesforce:

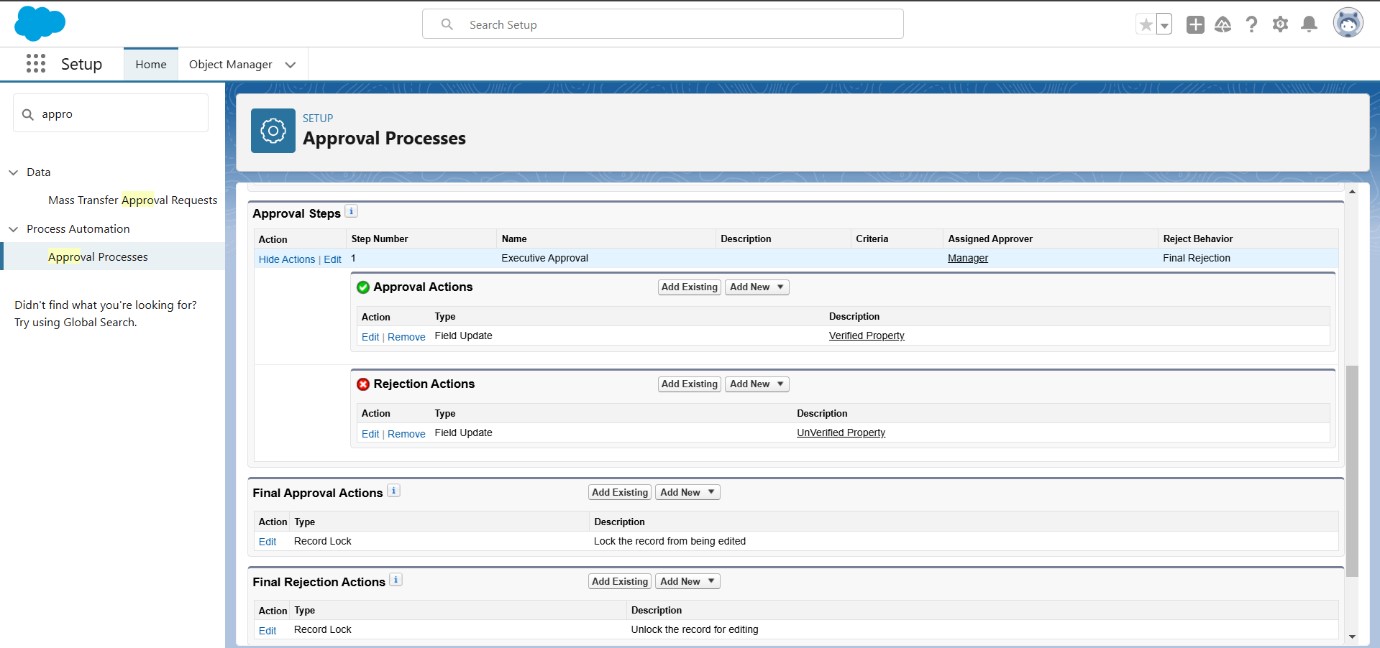
1. User 1 (Executive): Assigned Sales Executive role, System Administrator profile, and Salesforce license.
2. User 2 (Manager): Assigned Sales Manager role, Manager profile, and Salesforce Platform license.
3. User 3 (Customer): Assigned Customer role, Customer profile, and Salesforce Platform license, with the Verified checkbox unchecked.
4. User 4 (Customer2): Assigned Customer role, Customer profile, and Salesforce Platform license, with the Verified checkbox checked.



# Milestone 9: Create an Approval Process for Property Object

An Approval Process named *Property Approval* was created for the *Property* object in Salesforce. The process was configured with two key criteria: Location is not blank and Verified is false. The Sales Manager was designated as the automated approver, with permission for Administrators or the approver to edit records during the process. The approval page layout was customized to display essential fields like Property, Owner, Location, and Type. Sales Manager was set as the initial submitter, and Sales Executive as the final approver. Field updates were added to mark properties as Verified (True) or Unverified (False), and the approval process was activated for use.

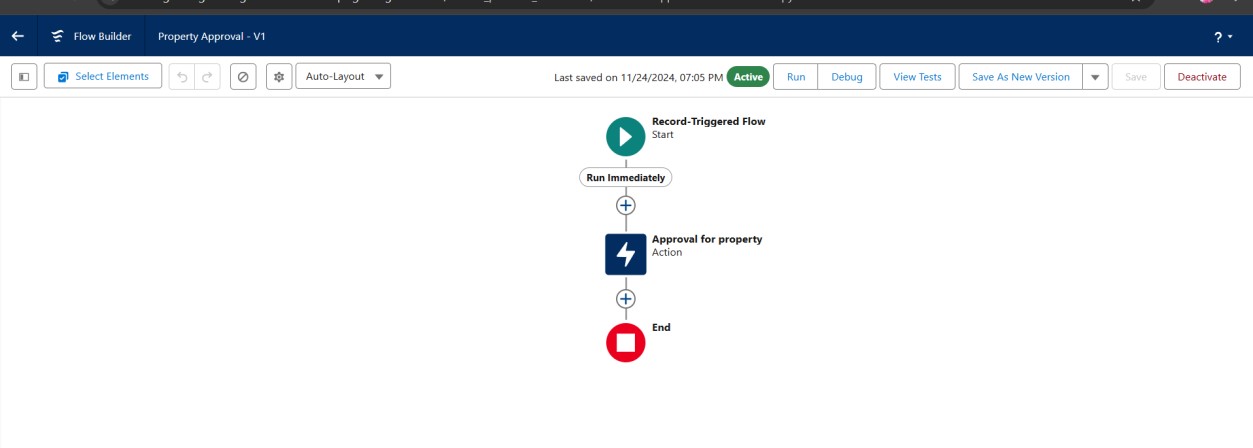




# Milestone 10: Create a Record trigger flow to submit the Approval Process Automatically

Created a Record Trigger Flow for the *Property* object in Salesforce to automate the submission of properties for approval. The flow was triggered when a record is created, with no entry conditions specified. The action added was "Submit for Approval", with the label "Approval for Property" and the Record

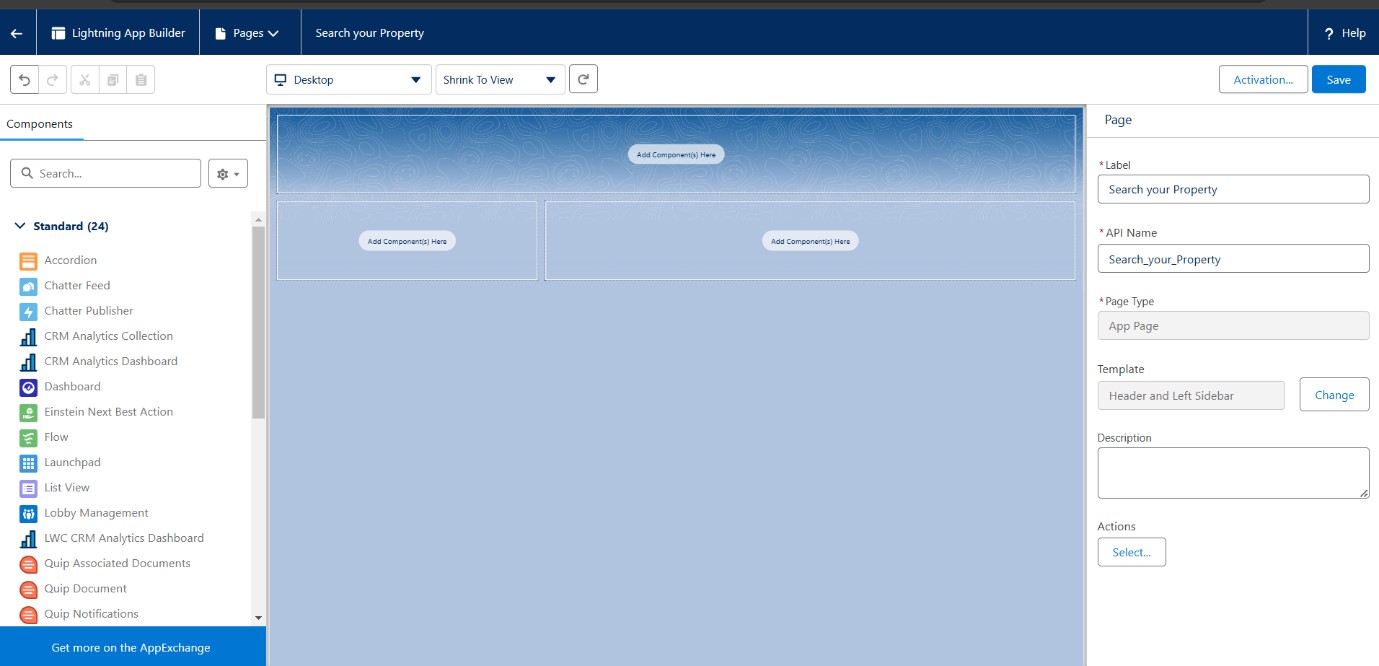
ID set to {!$Record.Id}. After configuring the flow, it was saved with the label "Property Approval" and activated for use. This automation ensures properties are automatically submitted for approval when they are created.

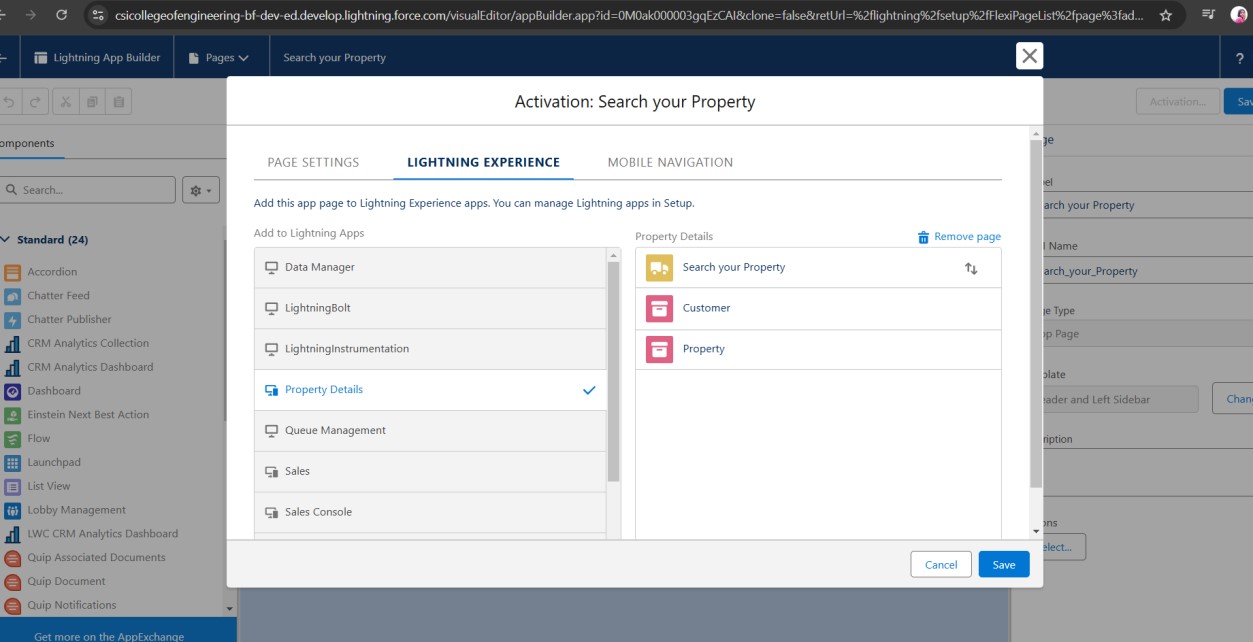


# Milestone 11: Create an App Page

Created an App Page for the *Property Details* object, titled "Search Your

Property", using Lightning App Builder. The page was configured with the Header and Left Sidebar components. After saving, it was activated and set to be available for all users. The page was added to the Property Details app in Lightning Experience. The page setup allows users to search properties directly within the app. All configurations were saved and the app was made accessible to users, streamlining property searches. This integration enhances user navigation and property management.



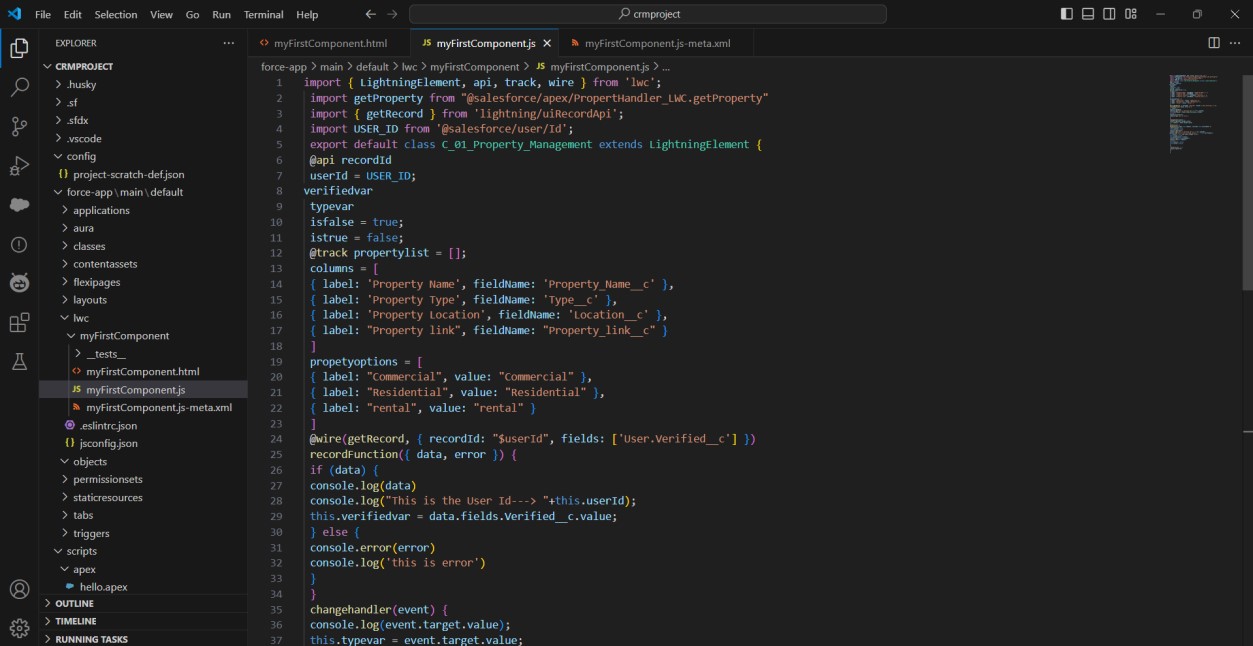
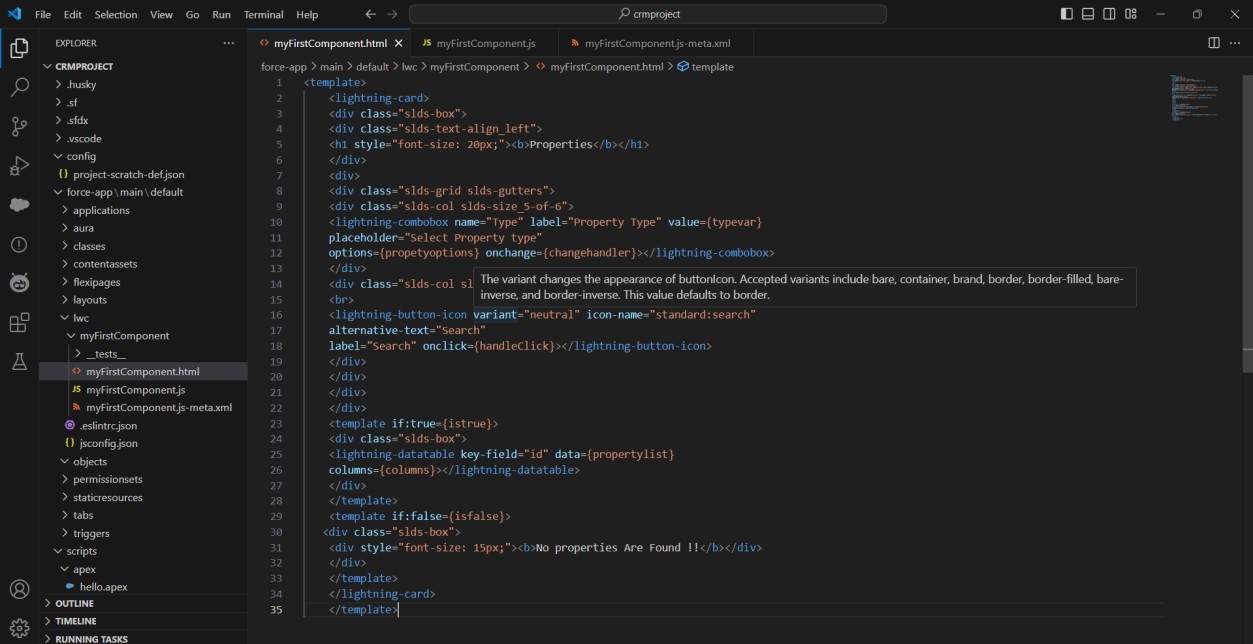
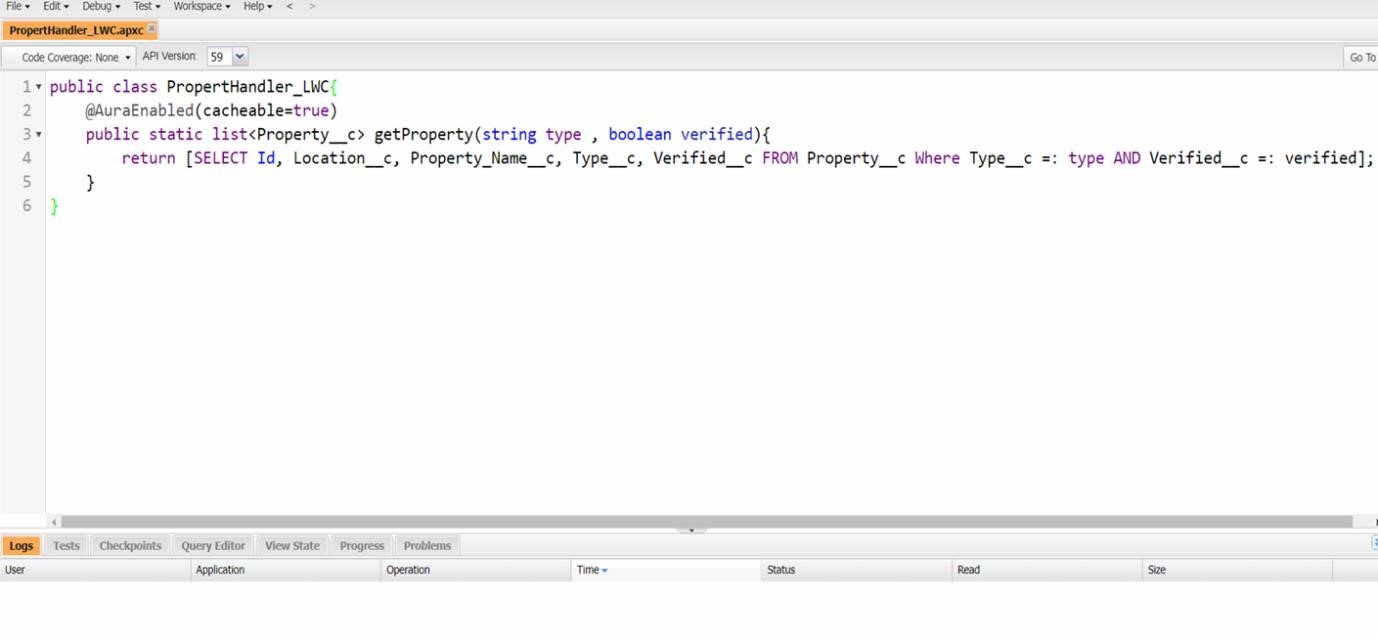


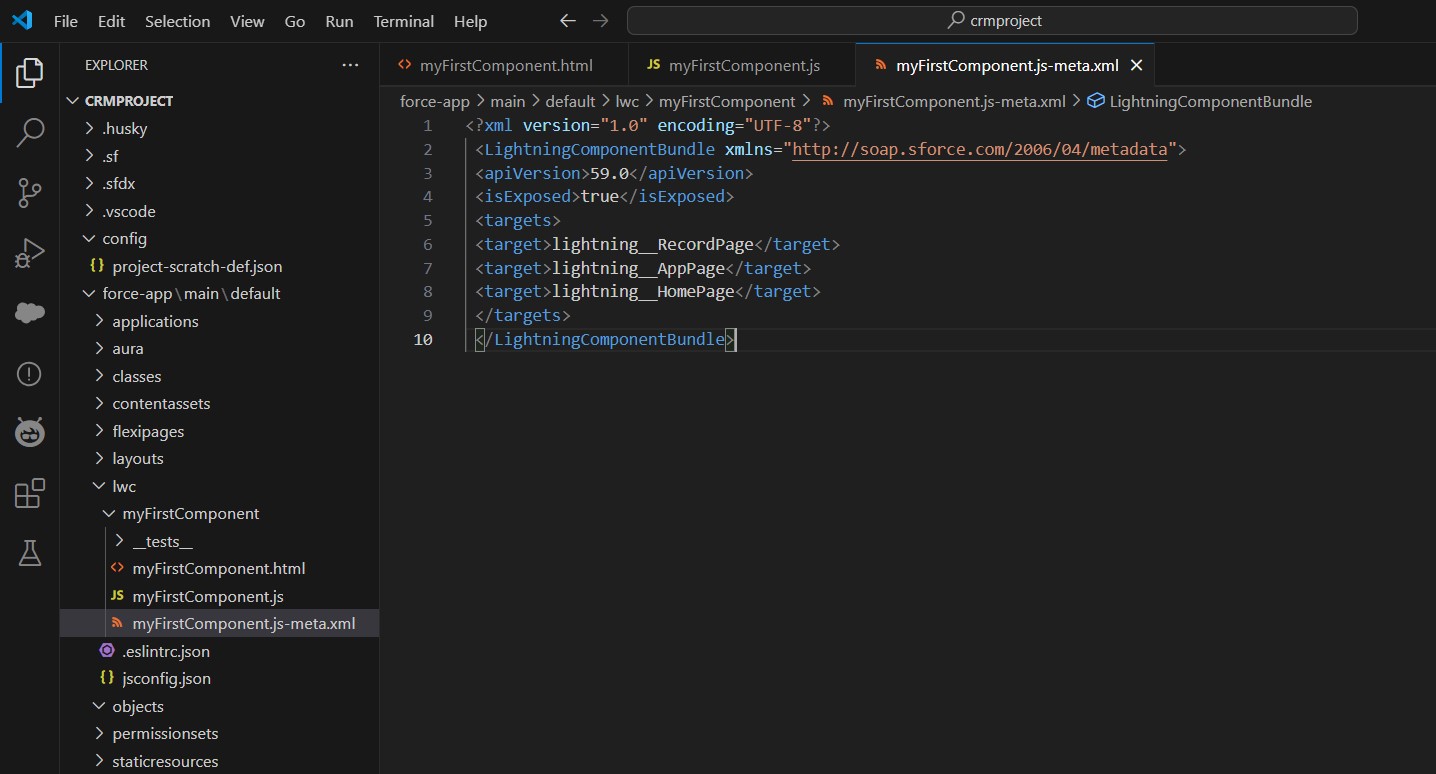
# Milestone 12: Create a LWC Component

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”

1.Create an Apex Class and make it aura enabled and name it

“PropertHandler\_LWC” and make it aura enabled,then in VSCode,authorize your org,create a Lighting Web Component and write the provided code.



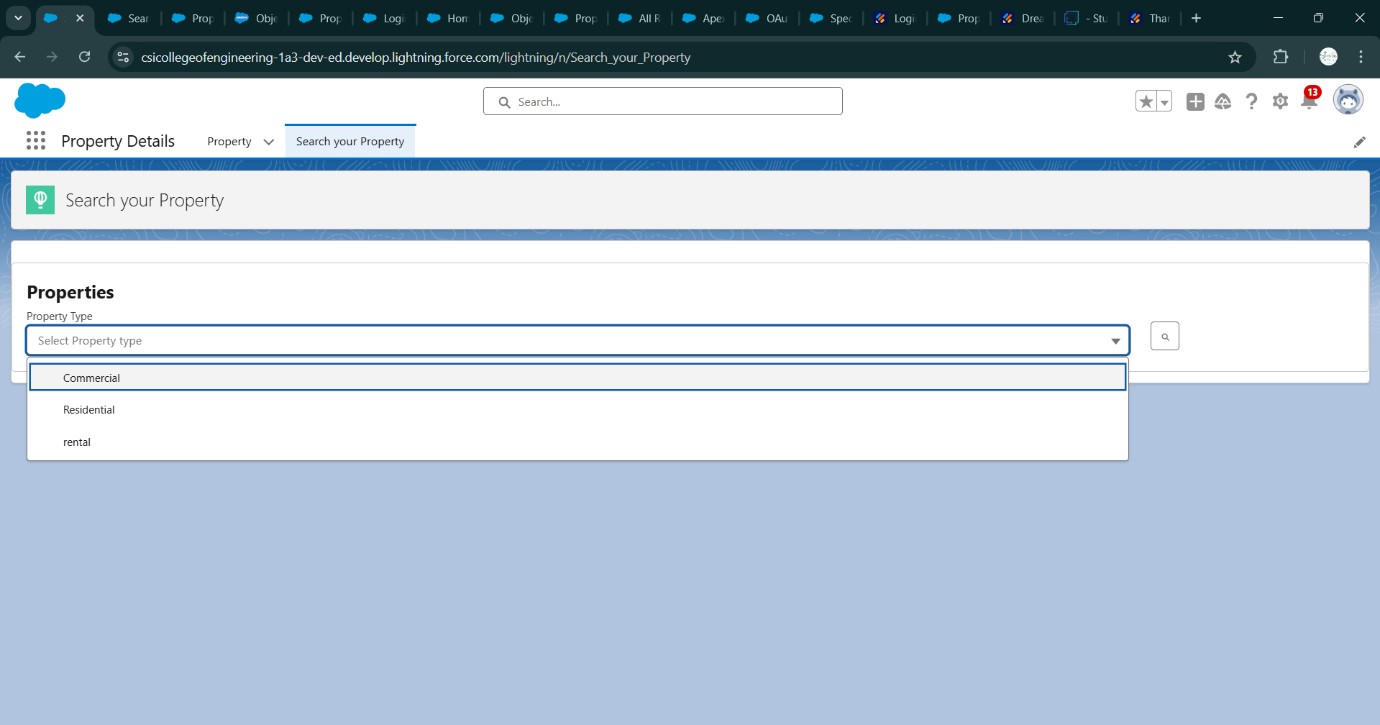


# Milestone 13: Drag this Component to your App Page

Navigate to App Launcher in Salesforce and searched for Property Details.

Click the gear icon and selected Edit Page.

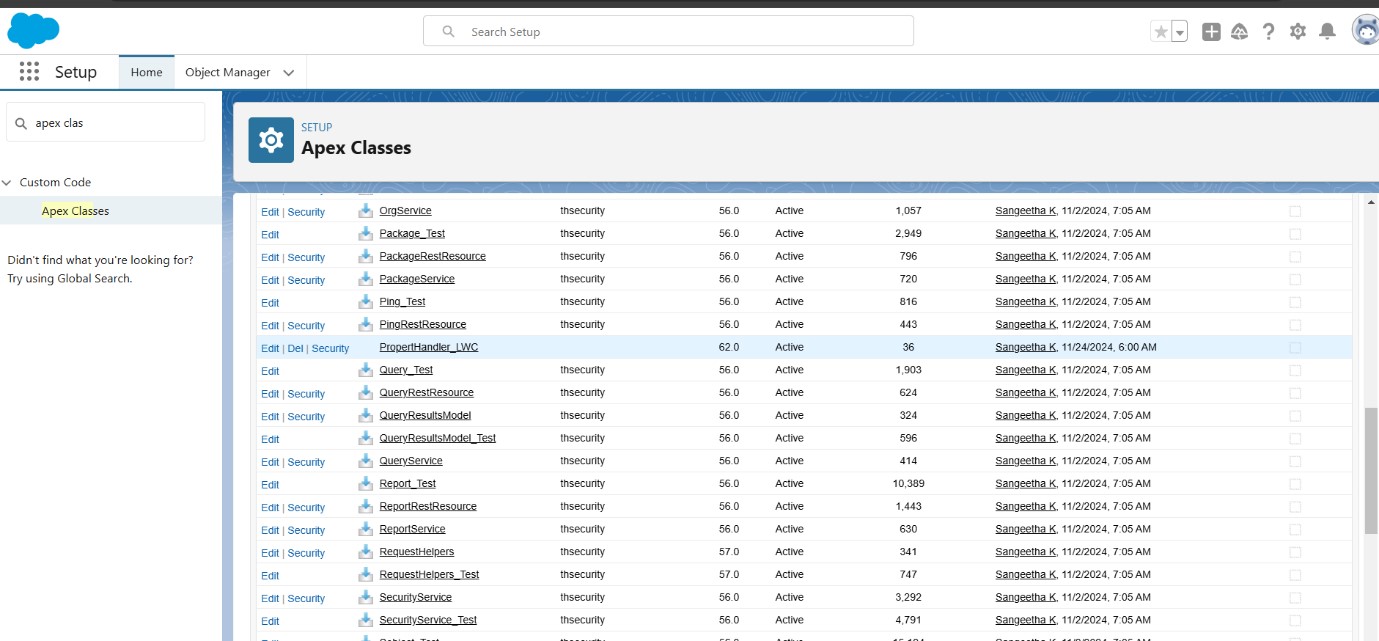
Drag the Lightning Web Component to the app page and save the changes.

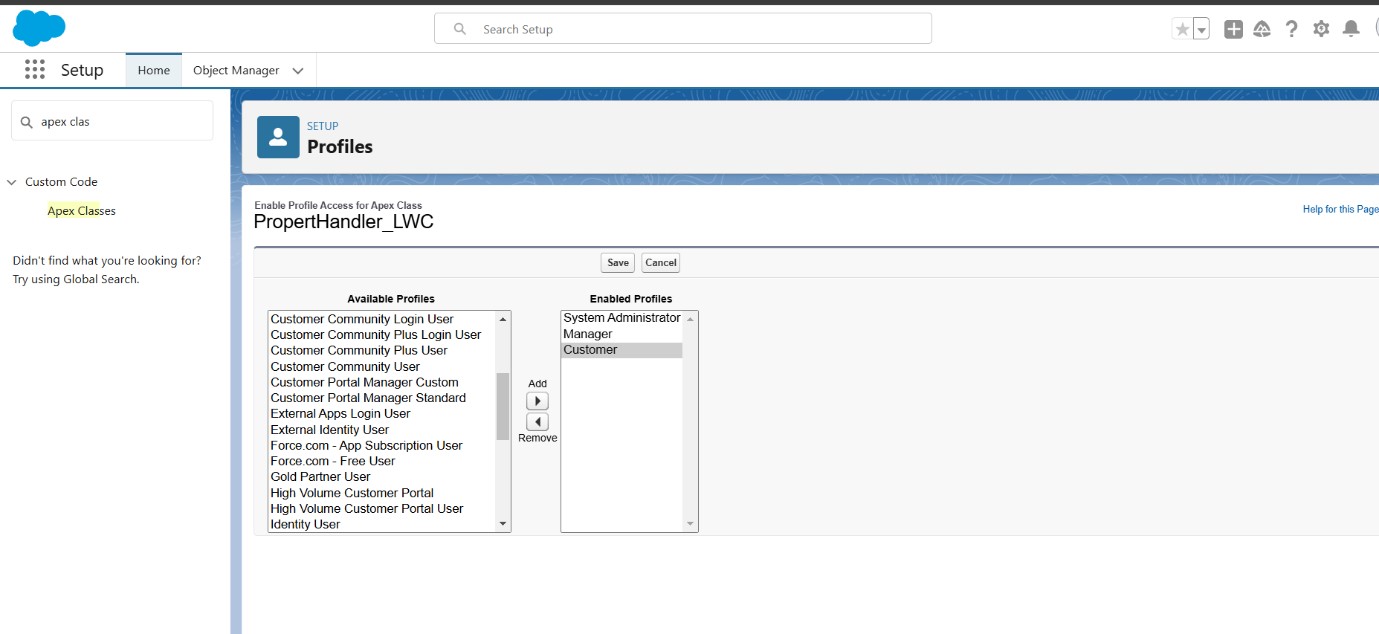


# Milestone 14: Give Access of Apex Classes to Profile

Navigate to Setup Apex Classes and click on "Security" next to

PropertyHandler\_\_LWC.In the Profiles section, added Manager and Customer profiles for access.Save the changes to grant the necessary profiles access to the PropertyHandler\_\_LWC Apex class.





# 5.Testing and Validation

**Approach to Testing:**

**Unit Testing**:

* Apex classes and triggers were tested to ensure that business logic, such as property approvals and record creation, executed correctly.
* Verified that Record Trigger Flows automated the approval process without errors.

**User Interface Testing:**

* + Tested the integration of Jotform with Salesforce to confirm seamless data flow.
  + Validated the functionality of the Property Details App, including the Lightning Web Component (LWC) for displaying properties.
  + Ensured proper role-based access to property records through different user profiles.

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

**Customer Data Management:**

* Automatically captured customer details from Jotform and created Salesforce records with accurate mapping.

**Property Management:**

* Allowed admins to manage property data, including adding, updating, and approving properties.

**Approval Workflow Automation:**

* Streamlined property verification using an automated approval process.
* Ensured only verified properties were visible to customers.

**Dynamic Property Display:**

* Displayed verified and non-verified properties dynamically using the LWC based on the user's status.

**Role-Based Access Control:**

* Enabled role and profile management to restrict or grant access to sensitive property and customer data.

# 7.Conclusion

This project successfully automated and streamlined client and property management processes using Salesforce. By integrating Jotform for seamless data collection, setting up roles and profiles for secure access, and implementing an approval workflow for property verification, the application enhanced operational efficiency and ensured data accuracy. The use of Lightning Web Components (LWC) provided a dynamic and user-friendly interface for property search and management, catering to both verified and non-verified customers. Overall, the project demonstrates how Salesforce's capabilities can be leveraged to address complex business requirements, improve user experience, and achieve operational goals effectively.