

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

1. Project Overview

This project implements a comprehensive Salesforce solution for Dreams World Properties, focusing on automating and streamlining customer relationship management in the real estate sector. The primary objective is to create an integrated system that manages customer interactions, property listings, and approval workflows through Salesforce, enhancing operational efficiency and customer experience and support long-term goals of organization.

Key Features

- Automated customer record creation through JotForm integration
- Property management system with approval workflows
- Role-based access control
- Custom Lightning Web Component for property search
- Verified customer management system

2. Objectives

List the specific, measurable goals the project intends to achieve. Examples:

Business Goals:

- Automate customer data collection and management
- Streamline property listing and approval processes
- Implement role-based access control for different user types
- Enhance customer experience through personalized property recommendations
- Improve operational efficiency in property management

Specific Outcomes:

- Integrated customer registration system
- Automated approval workflows for properties

- Differentiated access for verified and non-verified customers
- Custom property search interface
- Secure role-based data access

3. Salesforce Key Features and Concepts Utilized

This highlights the main functionalities and concepts applied within the Salesforce project.

Platform Features

1. Custom Objects
 - Customer Object
 - Property Object
2. Security Model
 - Custom Profiles
 - Role Hierarchy
 - Field-Level Security
 - Record-Level Access
3. Automation
 - Record-Triggered Flows
 - Approval Processes
4. Integration
 - JotForm Integration
 - Data Import Tools
5. Custom Development
 - Lightning Web Components
 - Apex Classes

4. Detailed Steps to Solution Design

Milestone 1: Customer Management System

JotForm Integration

- Created custom JotForm for customer data collection
- Established automated integration with Salesforce
- Configured field mappings for customer record creation



Dreams World

Name *

First Name

Last Name

Email

example@example.com

Phone Number

Please enter a valid phone number.

Which type of Property are you looking for ?

- ☐ RESIDENTIAL
☐ COMMERCIAL
☐ RENTAL

Budget Amount *

s.g., 23

Address

Street Address

Street Address Line 2

City

State / Province

Postal / Zip Code

Submit

Create a custom object from a spreadsheet

Upload

or Drop File Here

Upload .xlsx or .csv



Google Sheet



Office 365 or Drive

Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

Worksheet Details

Field Label Source

- ☐ Enter manually
☒ Detect from row

* Field Labels Row

1

Import 3 rows of Data?

- ☐ No, skip import
☒ Yes, import data

Record Name Field

Let Salesforce Create a Default F

Fields 11 of 16 to import

☐ Hide mapped fields

| IMPORT FILE FIELD NAME | SALESFORCE FIELD NAME | SALESFORCE FIELD TYPE | ADD TO LAYOUTS | FIELD PREVIEW |
|------------------------|-----------------------|-----------------------|-------------------------------------|------------------|
| ✓ Customer | ✕ Customer | Text | <input checked="" type="checkbox"/> | Rakesh |
| ✓ Phone Number | ✕ Phone Number | Integer | <input checked="" type="checkbox"/> | 788797 |
| ✓ Email | ✕ Email | Email | <input checked="" type="checkbox"/> | rakesh@gmail.com |
| ✓ State | ✕ State | Text | <input checked="" type="checkbox"/> | Telangana |
| ✓ Property Type | ✕ Property Type | Text | <input checked="" type="checkbox"/> | Residential |

Create a custom object from a spreadsheet

Define object and fields

Choose the data source, map fields and their types, and import field data.

Worksheet Details

Field Label Source

- ☐ Enter manually
☒ Detect from row

* Field Labels Row

1

Import 3 rows of Data?

- ☐ No, skip import
☒ Yes, import data

Record Name Field

Let Salesforce Create a Default F

Fields 4 of 4 to import

☐ Hide mapped fields

| IMPORT FILE FIELD NAME | SALESFORCE FIELD NAME | SALESFORCE FIELD TYPE | ADD TO LAYOUTS | FIELD PREVIEW |
|------------------------|-----------------------|-----------------------|-------------------------------------|------------------|
| ✓ Property Name | ✕ Property Name | Text | <input checked="" type="checkbox"/> | Lotus Apartments |
| ✓ Type | ✕ Type | Text | <input checked="" type="checkbox"/> | Residential |
| ✓ Location | ✕ Location | Text | <input checked="" type="checkbox"/> | hydeerabad |
| ✓ Verified | ✕ Verified | Text | <input checked="" type="checkbox"/> | checked |

Object Creation

- Implemented Customer object with necessary fields
- Created Property object with required fields
- Configured relationships between objects

Jotform Form Builder

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

WORKFLOWS
Turn your form into a workflow

JOTFORM SIGN
Power your forms with Jotform Sign

MOBILE NOTIFICATIONS
Customize mobile app notifications

SALESFORCE
Send new leads, contacts, or accounts to your sales CRM

Select a Salesforce Object

Customer

Create a record
Send data from form fields to matched Salesforce fields

| Object Fields | Dreams World |
|-----------------------|--|
| Customer | Name - First Name |
| City | Address - City |
| Budget Amount | Budget Amount |
| Property Type | Which type of Property are you lookin... |
| Phone Number | Phone Number |
| Street Address | Address - Street Address |
| Email | Email |
| Customer Name | Name - Last Name |
| State | Address - State |
| Street Address line 2 | Address - Street Address 2 |

+ Add Field

Update an existing record
Update Salesforce data when the selected field values match an existing record

Attachments
Send submission PDF and file uploads to this record

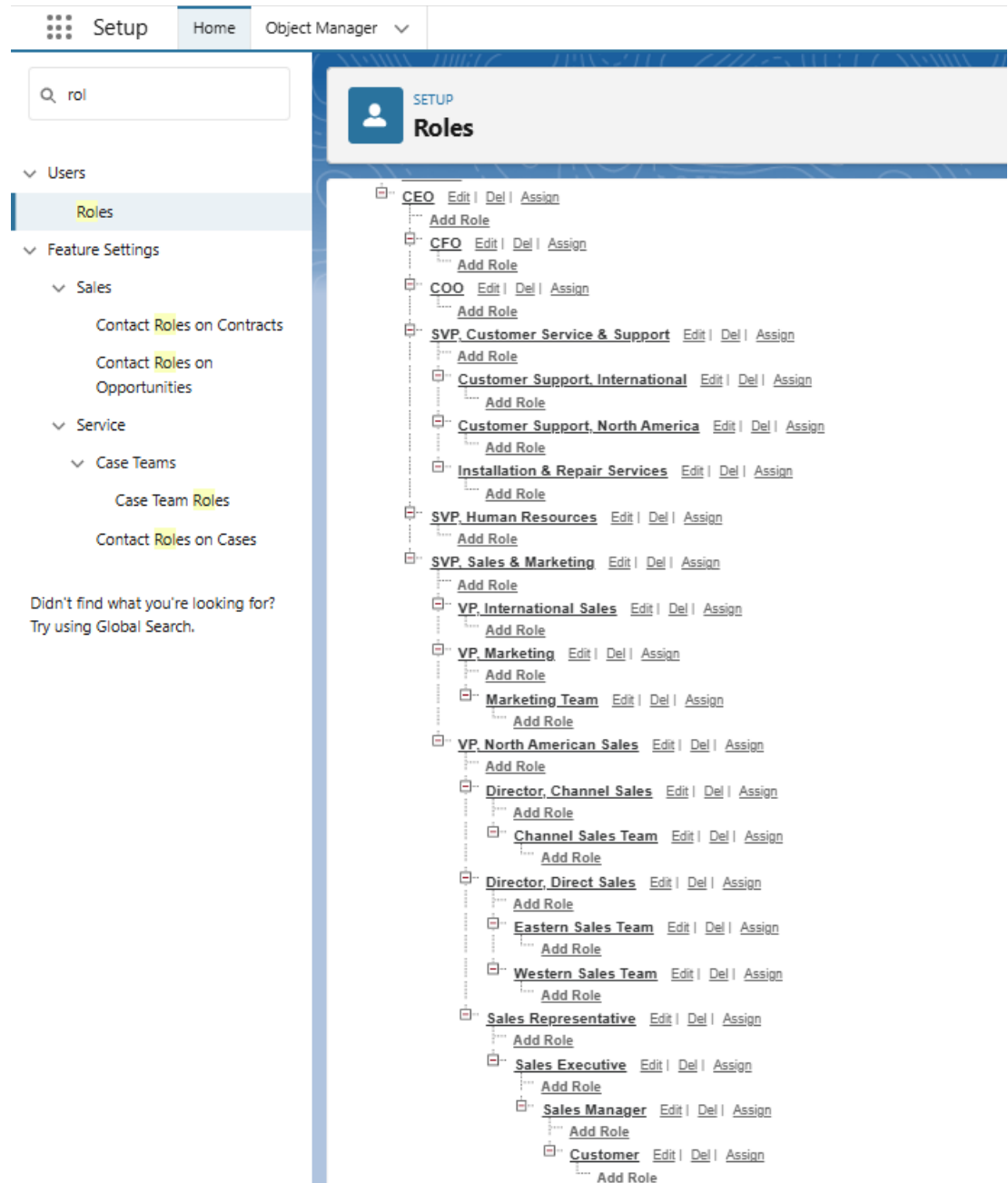
Cancel Save

Milestone 2: Security Implementation

Role Hierarchy

- Created three-tier role hierarchy:
 - Manager

- Customer Service
- Customer



The screenshot shows the Salesforce Setup interface. The left sidebar contains a navigation menu with 'Users' and 'Feature Settings' expanded. Under 'Feature Settings', 'Sales' and 'Service' are listed with sub-items like 'Contact Roles on Contracts' and 'Case Team Roles'. The main content area is titled 'Roles' and displays a hierarchical list of roles. The roles are organized into a tree structure, starting with 'CEO' at the top, followed by 'CFO', 'COO', and then various support and sales roles like 'SVP, Customer Service & Support', 'Customer Support, International', 'Customer Support, North America', 'Installation & Repair Services', 'SVP, Human Resources', 'SVP, Sales & Marketing', 'VP, International Sales', 'VP, Marketing', 'Marketing Team', 'VP, North American Sales', 'Director, Channel Sales', 'Channel Sales Team', 'Director, Direct Sales', 'Eastern Sales Team', 'Western Sales Team', 'Sales Representative', 'Sales Executive', 'Sales Manager', and 'Customer'. Each role has links for 'Edit', 'Del', and 'Assign'.

Profiles

- Developed custom profiles:
 - Customer Profile

Setup Home Object Manager

Q profile

Users

Profiles

Didn't find what you're looking for? Try using Global Search.

Profiles

Customer

Set the permissions and page layouts for this profile.

Profile Edit

Name: Customer

User License: Salesforce Platform

Description:

Custom Profile: ☒

Custom App Settings

Visible Default

Analytics Studio (standard__ Insights) ☐ ☐

App Launcher (standard__ AppLauncher) ☐ ☐

Platform (standard__ Platform) ☐ ☒

Property Details (Property_Details) ☒ ☐

WDC (standard__ Work) ☐ ☐

Service Provider Access

Tab Settings

☐ Override user's personal tab customizations

Standard Tab Settings

Home: Default On

Accounts: Default On

All Sites: Tab Hidden

Alternative Payment Methods: Tab Hidden

Analytics: Tab Hidden

Analytics: Tab Hidden

App Launcher: Default On

Appointment Categories: Default Off

Appointment Invitations: Default Off

Approval Requests: Default On

Assets: Tab Hidden

Assign Operation Logs: Default On

Inventory Item Reservations: Tab Hidden

Inventory Reservations: Tab Hidden

Investments: Tab Hidden

Labels: Default On

Leads: Tab Hidden

Licenses: Tab Hidden

Lightning Bolt Solutions: Default On

Lightning Usage: Default On

List Emails: Tab Hidden

Location Groups: Default On

Locations: Default On

Location Shipping Carrier Methods: Default On

Setup Home Object Manager

Q profile

Users

Profiles

Didn't find what you're looking for? Try using Global Search.

Profiles

Standard Object Permissions

The permissions defined here control access at the object level. Access to individual records within that object type is controlled by the sharing model. Set access levels based on the functional requirements for the profile. For example, create different groups of permissions for individual contributors, managers, and administrators. [View All Object Permissions](#)

Lightning Login Code: ☐

Manage Articles: ☐

View My Item's Connections: ☐

Manage Multi-Factor Authentication for Exempt Users: ☐

| | Basic Rights | | | | Data Administration | |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | Read | Create | Edit | Delete | View All | Modify All |
| Accounts | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Addresses | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Assets | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Authorization Forms | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Authorization Form Consents | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Authorization Form Data Uses | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Authorization Form Tests | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Background Operations | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Business Brands | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Communication Subscriptions | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Communication Subscription Channel Types | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Communication Subscription Consents | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Communication Subscription Settings | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Contacts | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Contact Point Addresses | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Contact Point Consents | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Contact Point Emails | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Contact Point Phones | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Contact Point Type Consents | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Customers | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| ESB Companies | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Data Use Legal Bases | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Data Use Purposes | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Documents | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Engagement Channel Types | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Files | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Individuals | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Labels | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Locations | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Party Consents | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Push Topics | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Sellers | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Streaming Channels | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| User External Credentials | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Custom Object Permissions

| | Basic Rights | | | | Data Administration | |
|------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| | Read | Create | Edit | Delete | View All | Modify All |
| Customer | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Entertainment Requests | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input 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"/> |

- Manager Profile

javascript:srcUp(%27%2F00edM000008e9pZ%2Fe%3FretURL%3D%252F00e%253Fcf%253D00BdM00000Ke5H0%2526rolodexindex%253D12%2526page%253D1%2526isdt%253Dp1%26isdt%3Dp1%27)

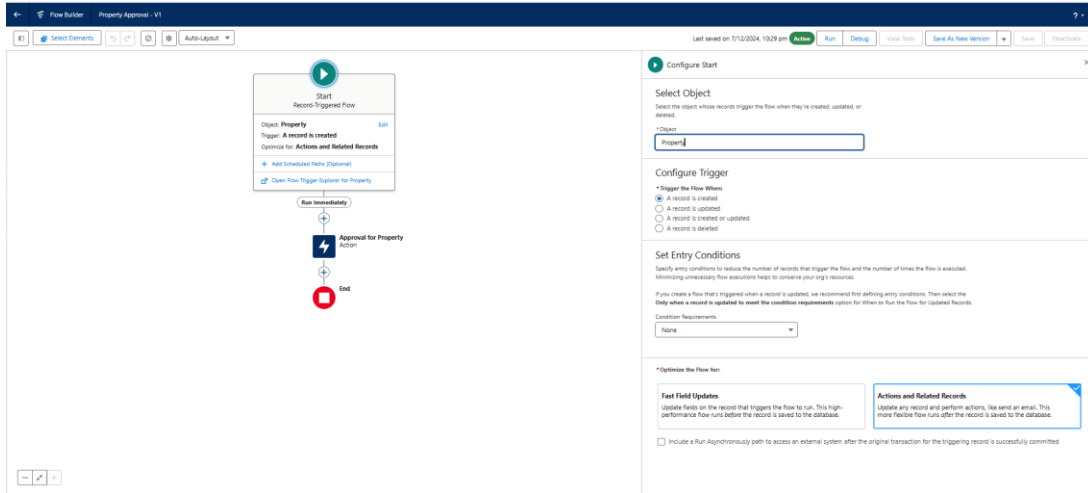
Session settings

Custom Fields

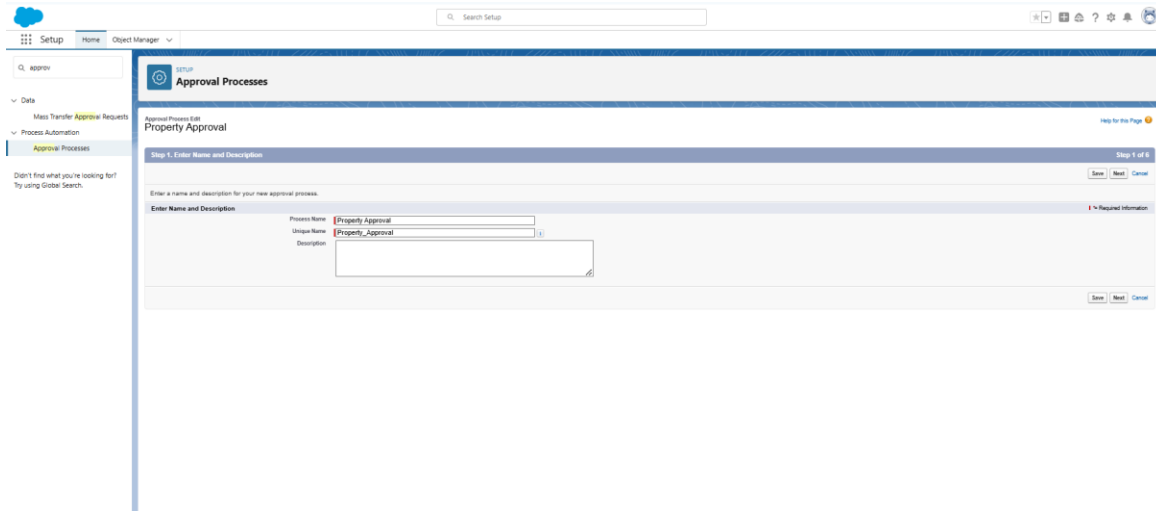
- Added verification checkbox on User object
- Configured field-level security

[illegible]

- Implemented property approval workflow
- Created automated submission process



- Configured approval notifications



Setup: Approval Processes

Approval Process Edit: **Property Approval**

Step 1: Enter Name and Description

Enter a name and description for your new approval process.

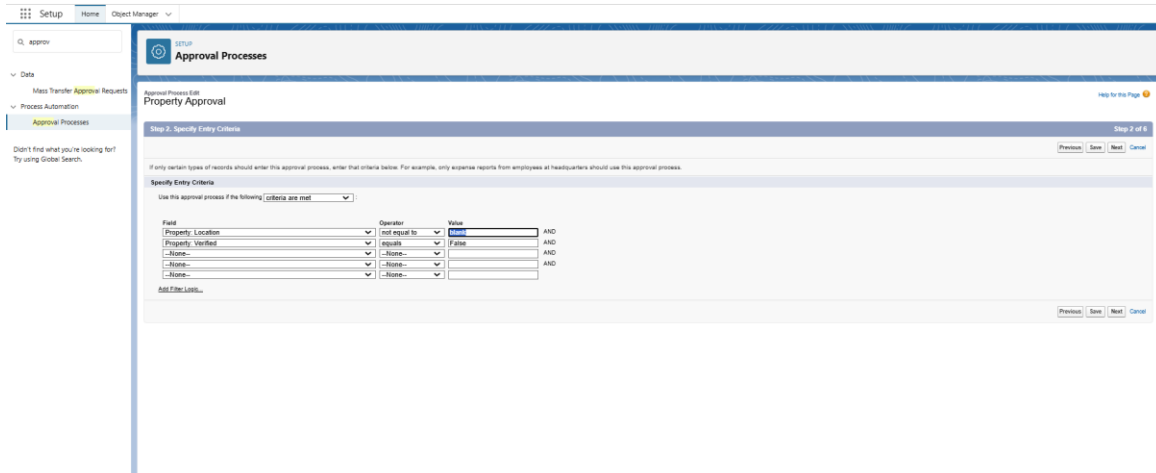
Enter Name and Description:

Process Name: **Property Approval**

Unique Name: **Property_Approval**

Description: **Property Approval**

Save **Next** **Cancel**



Setup: Approval Processes

Approval Process Edit: **Property Approval**

Step 2: Specify Entry Criteria

If only certain types of records should enter the approval process, enter that criteria below. For example, only expense reports from employees at headquarters should use this approval process.

Specify Entry Criteria

Use this approval process if the following criteria are met:

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

Save **Next** **Cancel**

Setup

Home

Object Manager

Q

approval

Data

Mass Transfer

Approval Requests

Process Automation

Approval Processes

Didn't find what you're looking for?

Try using Global Search.

Setup

Approval Processes

Approval Process Edit

Property Approval

Help for this Page

Step 6. Select Fields to Display on Approval Page Layout

Step 6 of 6

The approval page is where an approver will actually approve or reject a request. Using the options below, choose the fields to display on this page.

Available Fields

Created By

Last Modified By

Verified

Add

Remove

Selected Fields

Property

Owner

Property Name

Location

Type

Up

Down

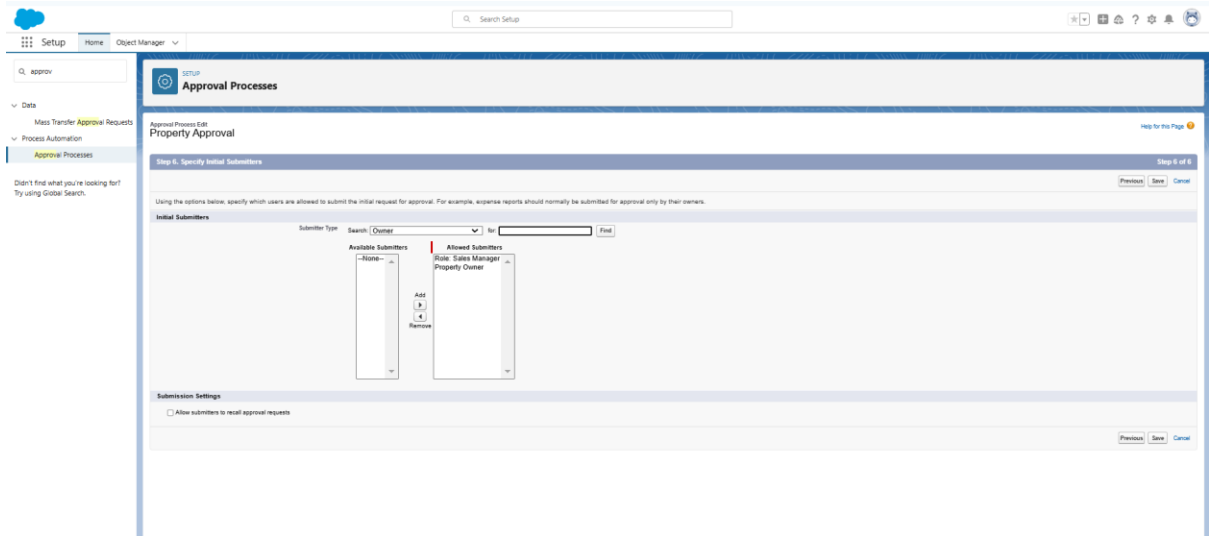
Click here to learn about approvals

Previous

Save

Next

Cancel



Approval Processes

Approval Process Edit: Property Approval

Step 6: Specify Initial Submitters

Using the options below, specify which users are allowed to submit the initial request for approval. For example, expense reports should normally be submitted for approval only by their owners.

Initial Submitters

Submitter Type: Search: Owner for Find

Available Submitters

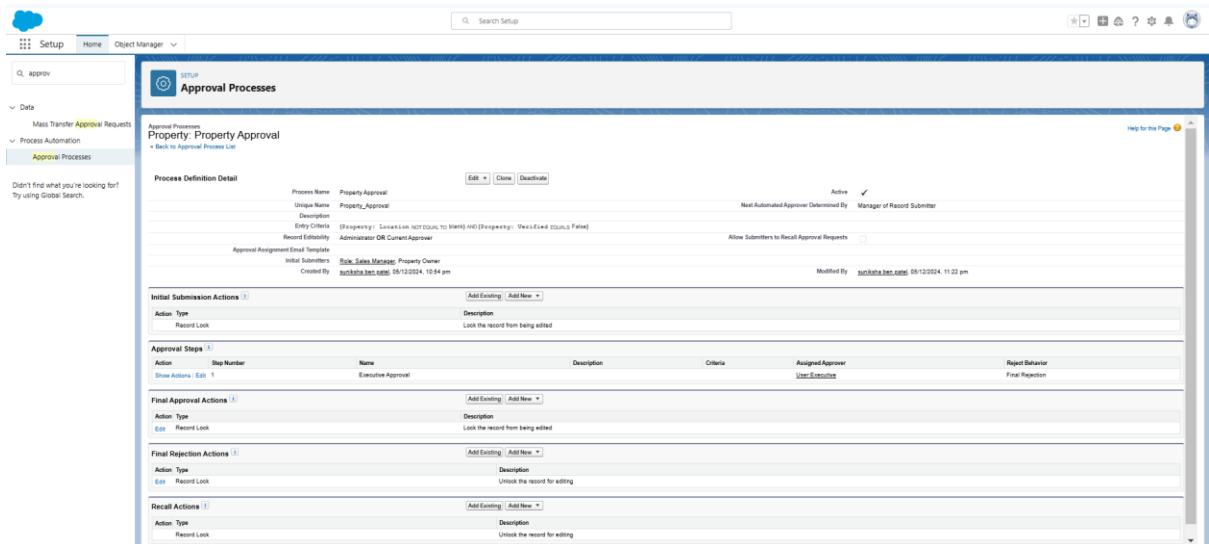
- None

Allowed Submitters

- Role: Sales Manager
- Property Owner

Submission Settings

☐ Allow submitters to recall approval requests



Approval Processes

Approval Process Edit: Property Approval

Back to Approval Process List

Process Definition Detail

Process Name: Property Approval

Unique Name: Property_Approval

Description: (Property: Location NOT EQUAL to Item) AND (Property: Division EQUAL False)

Record Editability: Administrator OR Current Approver

Approval Assignment Email Template: Initial Submitters: Role: Sales Manager; Property Owner

Created By: sushanta.jen.sage 05/12/2024, 10:34 pm

Modified By: sushanta.jen.sage 05/12/2024, 11:22 pm

Initial Submission Actions

Action: Record Lock

Approval Steps

| Action | Step Number | Name | Description | Criteria | Assigned Approver | Reject Behavior |
|--------------------|-------------|------|-------------|----------|-------------------|-----------------|
| Executive Approval | | | | | User: Executive | Final Rejection |

Final Approval Actions

Action: Record Lock

Final Rejection Actions

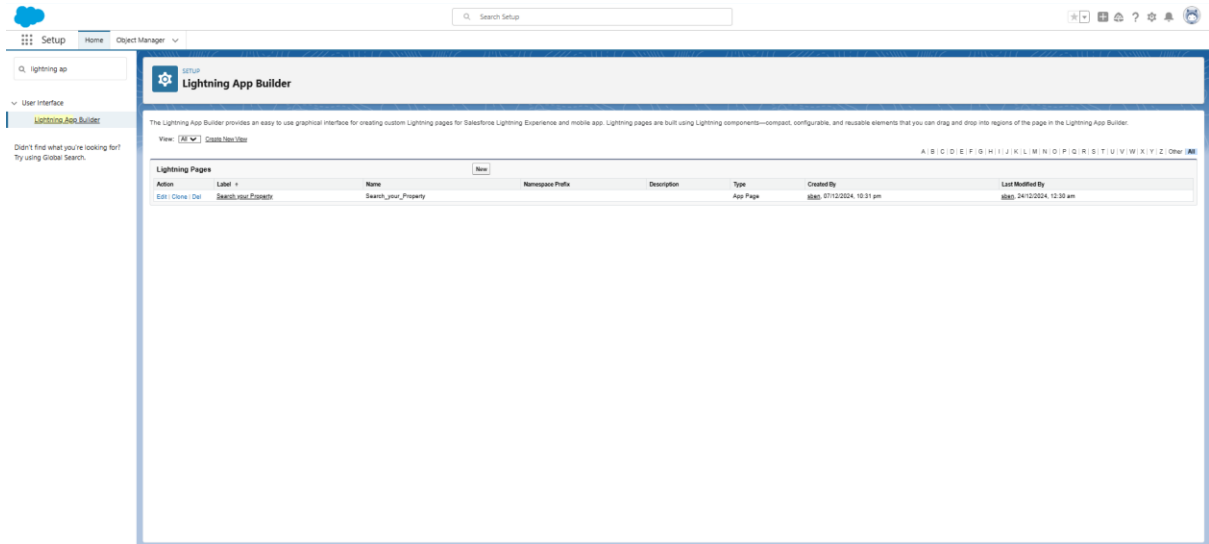
Action: Record Lock

Recall Actions

Action: Record Lock

Custom Interface

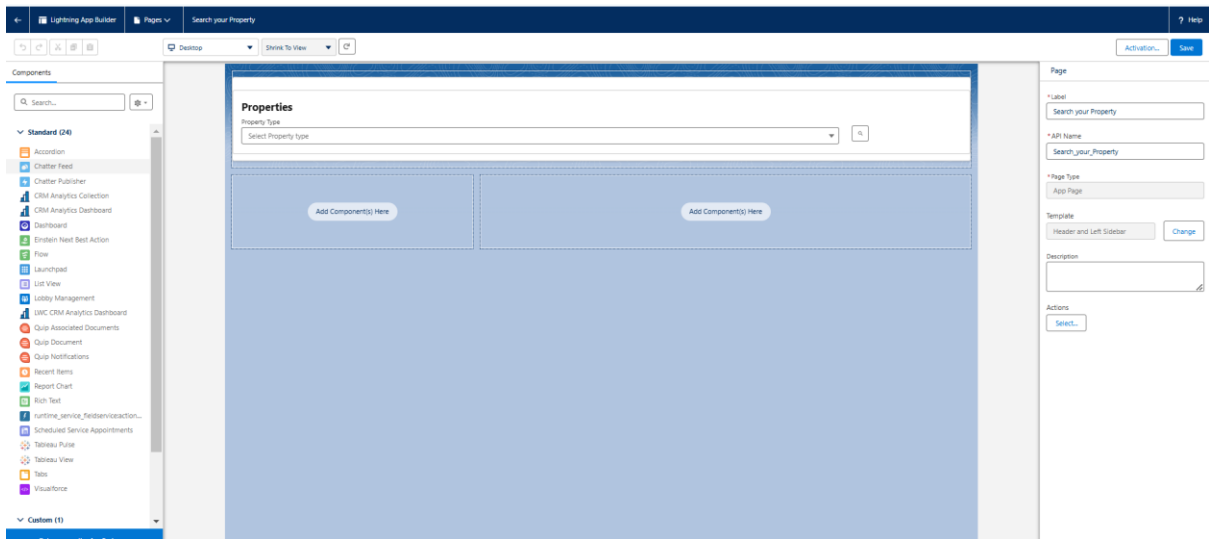
- Developed "Search Your Property" app page



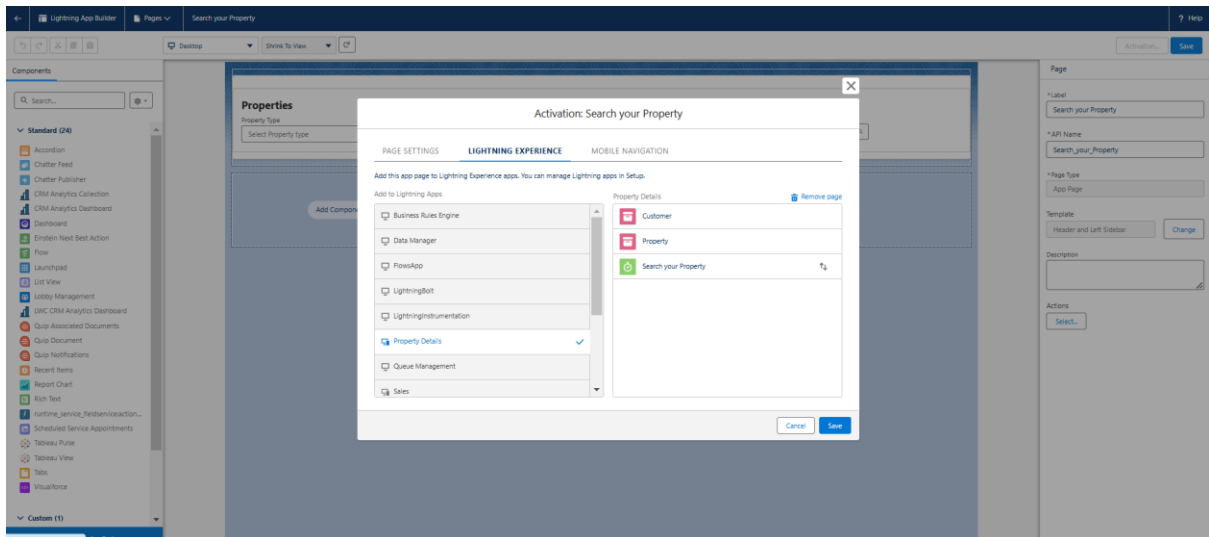
The Lightning App Builder provides an easy-to-use graphical interface for creating custom Lightning pages for Salesforce Lightning Experience and mobile app. Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder.

View: [All] [Create New Page]

| Lightning Pages | Name | Namespace Prefix | Description | Type | Created By | Last Modified By |
|--|----------------------|----------------------|-------------|----------|--------------------------|--------------------------|
| Edit Clone Del | Search your Property | Search_your_Property | | App Page | 8888 4712/2024, 10:31 pm | 8888 2412/2024, 10:30 am |



Lightning App Builder Edit screen for "Search your Property". The interface shows a central canvas with two "Add Component(s) Here" buttons. On the left, a "Components" panel lists various standard and custom components. On the right, a "Page" panel shows configuration options for the page, including Label, API Name, Page Type, Template, and Description.



Lightning App Builder Edit screen for "Search your Property" with an "Activation" dialog box open. The dialog box shows the "LIGHTNING EXPERIENCE" tab, where the "Search your Property" page is listed as active. The "MOBILE NAVIGATION" tab is also visible, showing the "Property Details" section.

- Created custom LWC component for property search

```

Command Prompt - sfdx forc x + v

C:\Users\sunik\OneDrive\Desktop\salesforce>npm install

up to date, audited 1 package in 1s

found 0 vulnerabilities

C:\Users\sunik\OneDrive\Desktop\salesforce>npm install @salesforce/sfdx-lwc-jest --save-dev
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if
you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerfu
l.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm warn deprecated @babel/plugin-proposal-dynamic-import@7.18.6: This proposal has been merged to the ECMAScript standa
rd and thus this plugin is no longer maintained. Please use @babel/plugin-transform-dynamic-import instead.
npm warn deprecated domexception@4.0.0: Use your platform's native DOMException instead

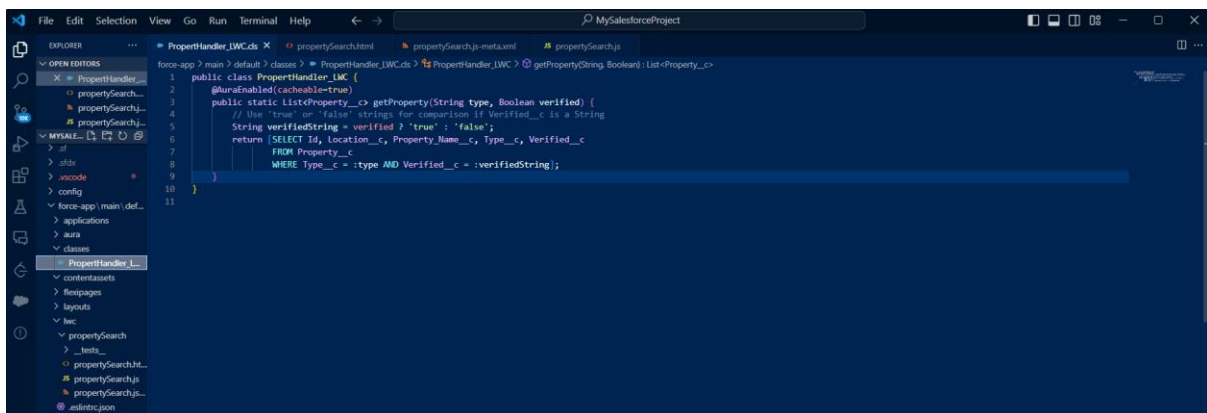
added 406 packages, and audited 407 packages in 1m

43 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\sunik\OneDrive\Desktop\salesforce>sfdx force:source:deploy -p force-app/main/default/classes/PropertHandler_LWC
.cls
» Warning: force source deploy is not a sf command.
Did you mean force source delete? (Y/n)
» Error: Run sf help force for a list of available commands.

```



```

force-app > main > default > classes > PropertyHandler_LWC > PropertyHandler_LWC.cls
1 public class PropertyHandler_LWC {
2     @AuraEnabled(cacheable=true)
3     public static List<Property__c> getProperty(String type, Boolean verified) {
4         // Use 'true' or 'false' strings for comparison if Verified__c is a String
5         String verifiedString = verified ? 'true' : 'false';
6         return [SELECT Id, Location__c, Property_Name__c, Type__c, Verified__c
7                 FROM Property__c
8                 WHERE Type__c = :type AND Verified__c = :verifiedString];
9     }
10 }
11

```

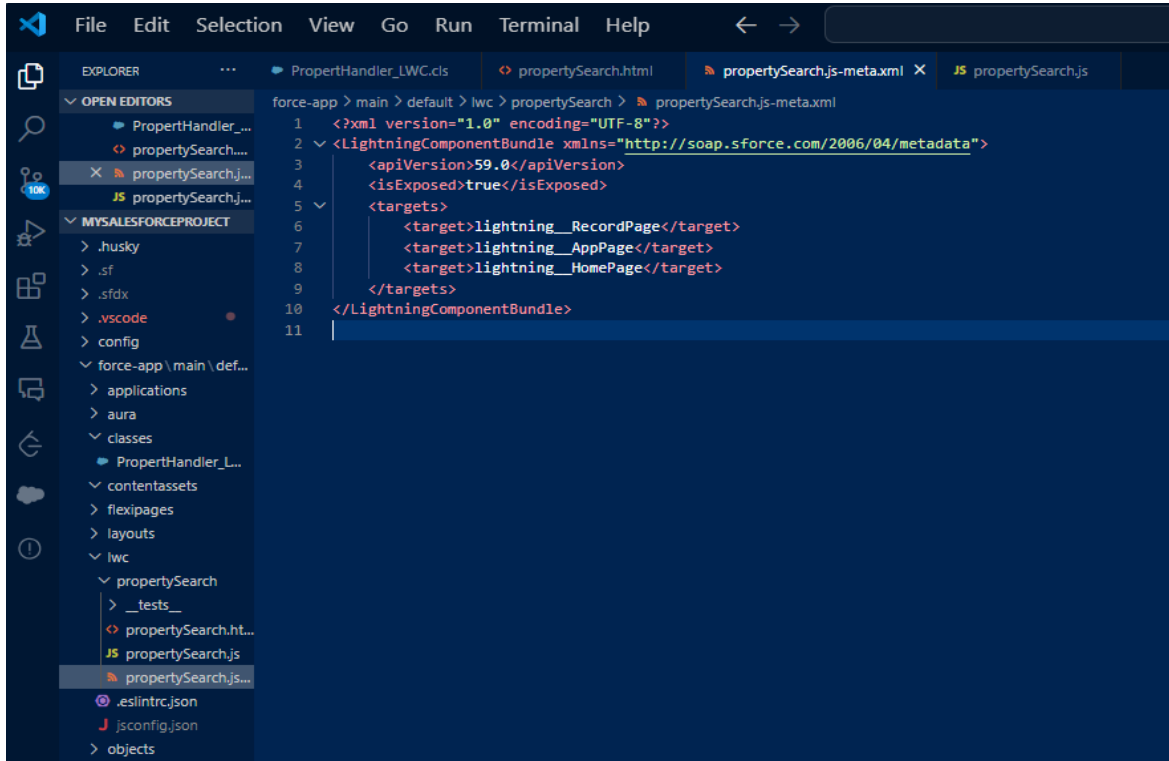
```

File Edit Selection View Go Run Terminal Help

force-app > main > default > lwc > propertySearch > propertySearch.html > template > lightning-card

<template>
  <lightning-card>
    <div class="slds-box">
      <div class="slds-text-align_left">
        <h1 style="font-size: 28px;"><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">
            <b>
              <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>

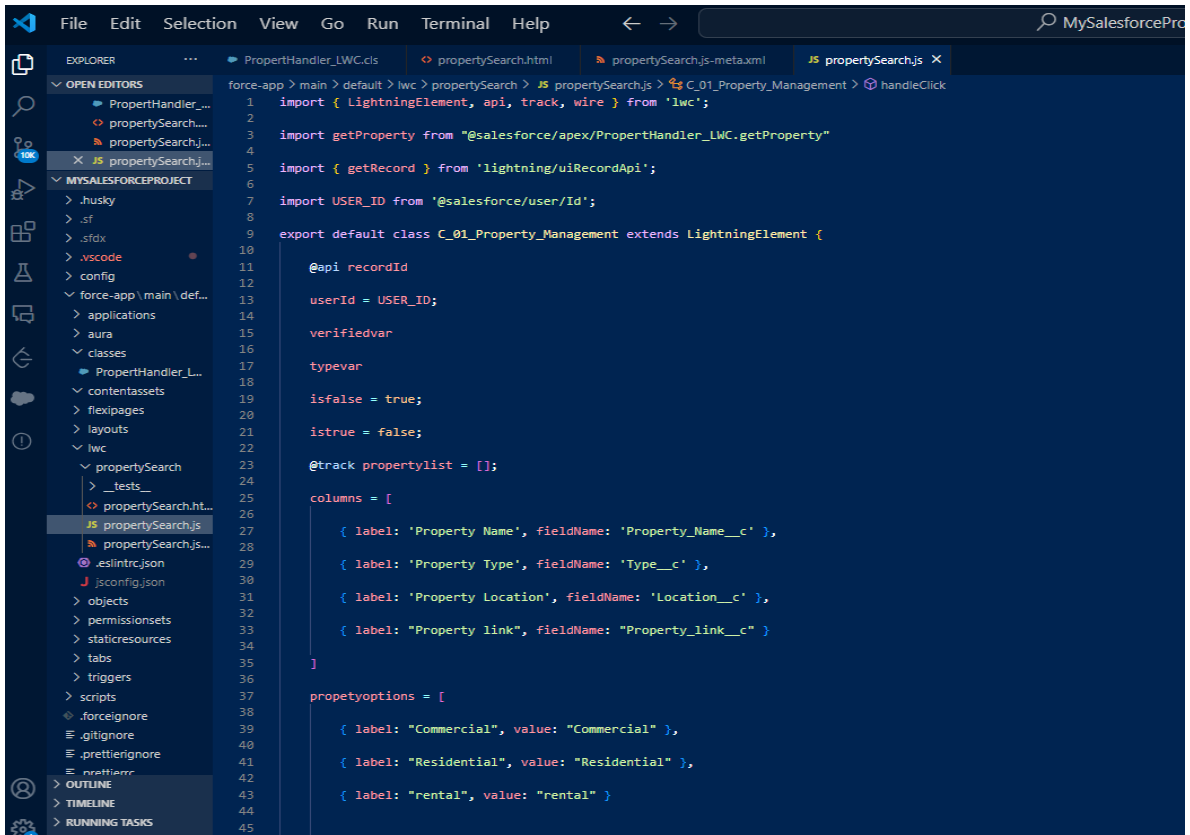
```



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3      <apiVersion>59.0</apiVersion>
4      <isExposed>true</isExposed>
5      <targets>
6          <target>lightning__RecordPage</target>
7          <target>lightning__AppPage</target>
8          <target>lightning__HomePage</target>
9      </targets>
10 </LightningComponentBundle>
11

```



```

1  import { LightningElement, api, track, wire } from 'lwc';
2
3  import getProperty from "@salesforce/apex/PropertyHandler_LWC.getProperty";
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
12
13      @api recordId;
14
15      userId = USER_ID;
16
17      verifiedvar;
18
19      typevar;
20
21      isfalse = true;
22
23      istrue = false;
24
25      @track propertylist = [];
26
27      columns = [
28          { label: 'Property Name', fieldName: 'Property_Name__c' },
29          { label: 'Property Type', fieldName: 'Type__c' },
30          { label: 'Property Location', fieldName: 'Location__c' },
31          { label: 'Property link', fieldName: 'Property_link__c' }
32      ]
33
34      propertyoptions = [
35          { label: "Commercial", value: "Commercial" },
36          { label: "Residential", value: "Residential" },
37          { label: "rental", value: "rental" }
38      ]
39
40
41
42
43
44
45

```



```

File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    force-app > main > default > lwc > propertySearch > JS propertySearch.js > C_01_Property_Management > handleClick
    propertySearch.js
  MYSALESFORCEPROJECT
    .husky
    .sf
    .sfdx
    .vscode
    config
    force-app\main\def...
    applications
    aura
    classes
      PropertyHandler_L...
    contentassets
    flexipages
    layouts
    lwc
      propertySearch
        _tests_
        propertySearch.ht...
        JS propertySearch.js
        propertySearch.js...
        .eslintrc.json
        .jsconfig.json
        objects
        permissionsets
        staticresources
        tabs
        triggers
        scripts
        .forceignore
        .gitignore
        .prettierrc
        .prettierrc
    OUTLINE
    TIMELINE
    RUNNING TASKS
    JAVA PROJECTS

force-app > main > default > lwc > propertySearch > JS propertySearch.js > C_01_Property_Management > handleClick
export default class C_01_Property_Management extends LightningElement {
  propertyoptions = [
  ]

  @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)
    })
  }
}

```

```

File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    force-app > main > default > lwc > propertySearch > JS propertySearch.js > C_01_Property_Management > handleClick
    propertySearch.js
  MYSALESFORCEPROJECT
    .husky
    .sf
    .sfdx
    .vscode
    config
    force-app\main\def...
    applications
    aura
    classes
      PropertyHandler_L...
    contentassets
    flexipages
    layouts
    lwc
      propertySearch
        _tests_
        propertySearch.ht...
        JS propertySearch.js
        propertySearch.js...
        .eslintrc.json
        .jsconfig.json
        objects
        permissionsets
        staticresources
        tabs
        triggers
        scripts
        .forceignore
        .gitignore
        .prettierrc
        .prettierrc
    OUTLINE
    TIMELINE
    RUNNING TASKS
    JAVA PROJECTS

force-app > main > default > lwc > propertySearch > JS propertySearch.js > C_01_Property_Management > handleClick
handleClick() {
  .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);

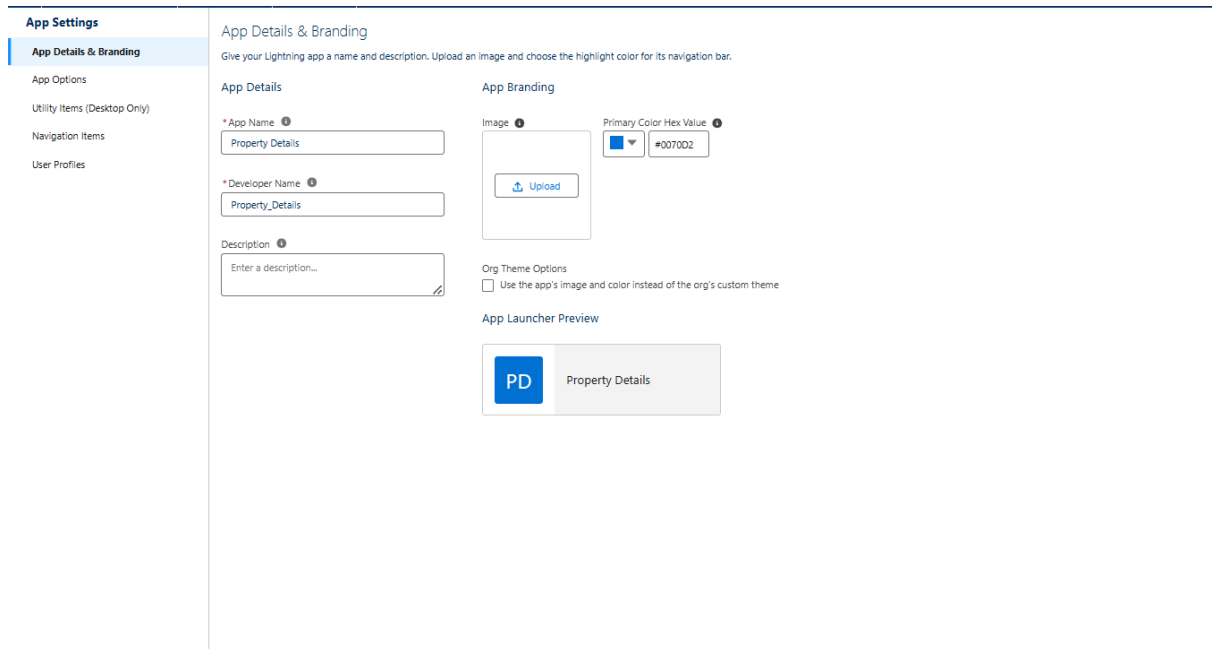
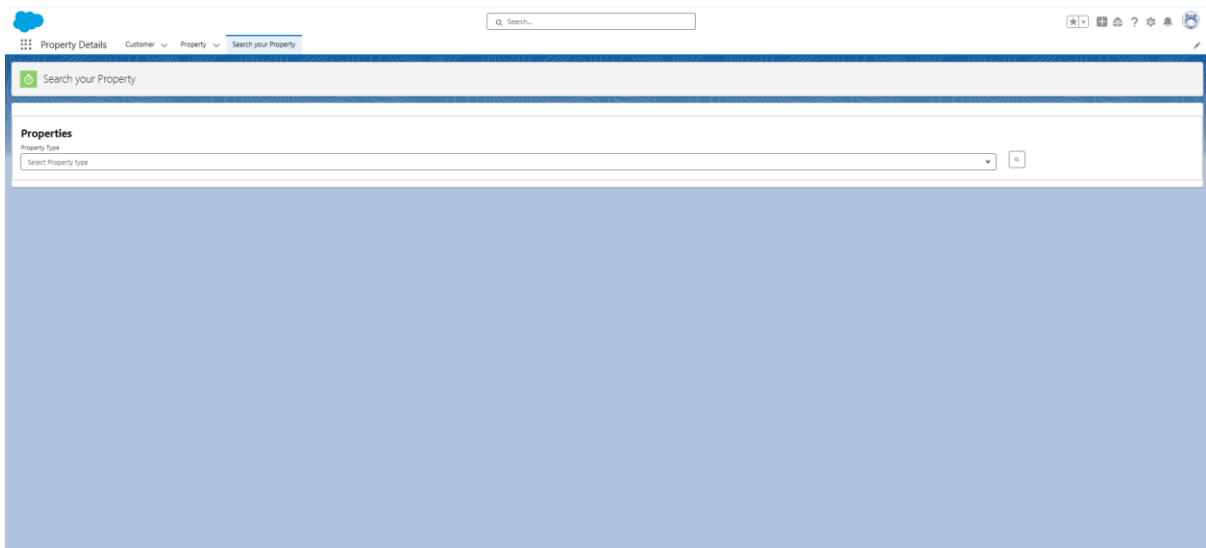
      console.log(this.typevar)
    } else {
      this.isfalse = false;

      this.istrue = false;
    }
  })

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

```

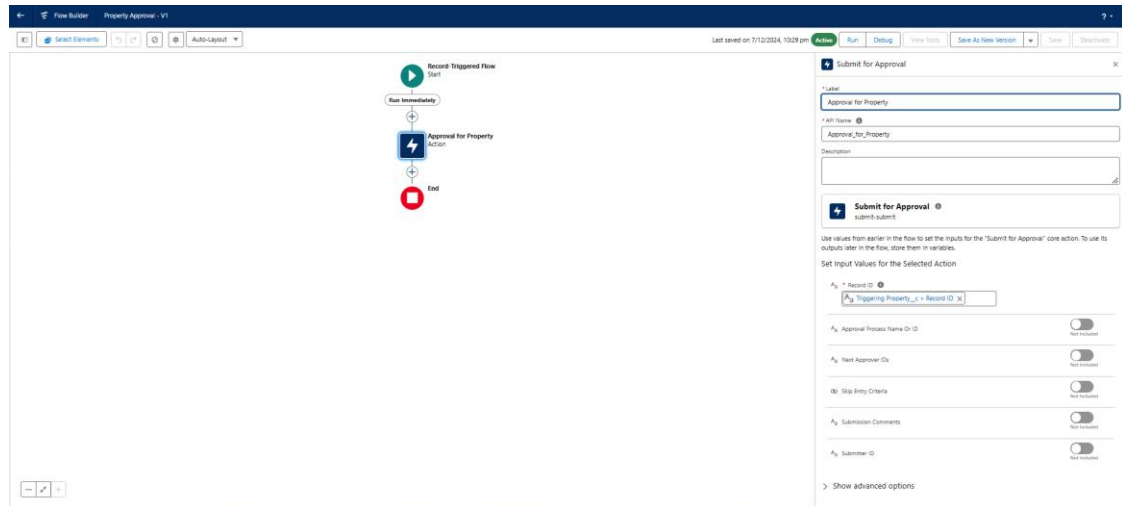
- Implemented verified/non-verified property filtering

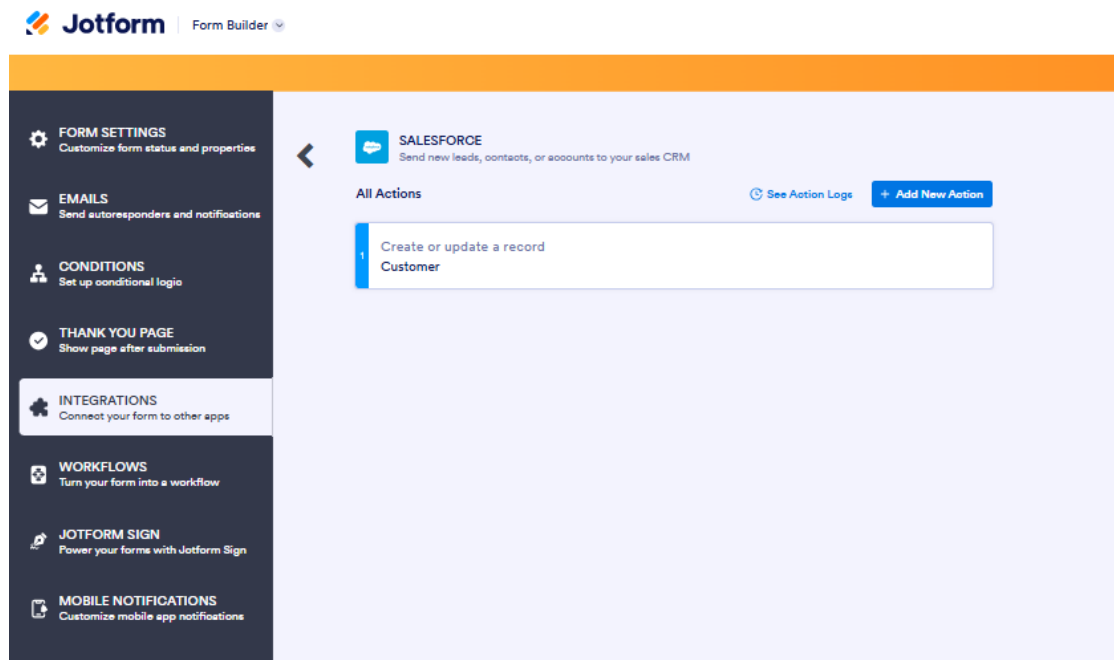
5. Testing and Validation

Describe the approach to testing:

- Unit Testing (Apex Classes, Triggers).
 1. Apex class testing for property filtering logic
 2. Trigger testing for approval process automation

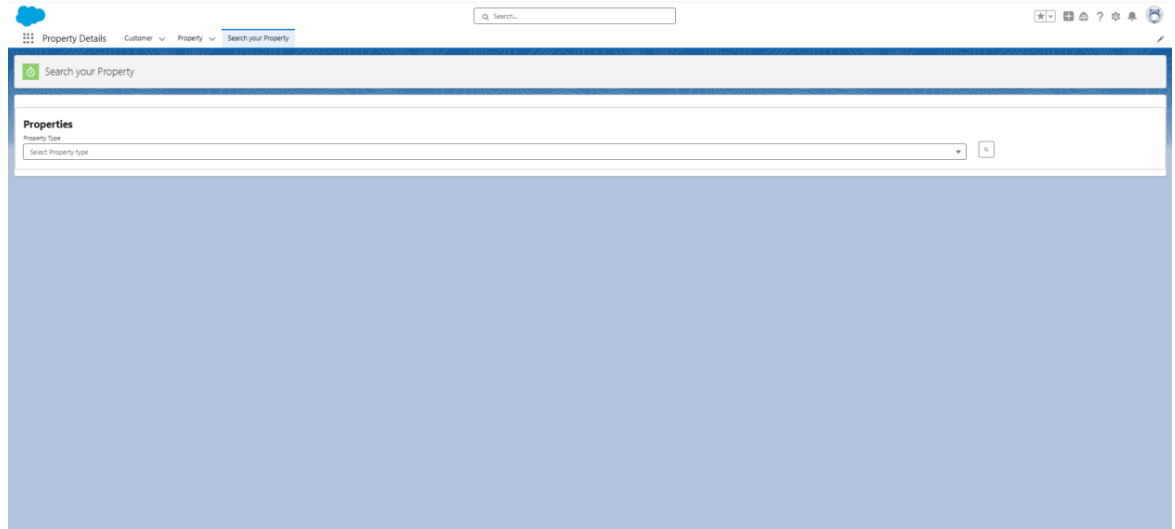


3. Integration testing for JotForm connectivity

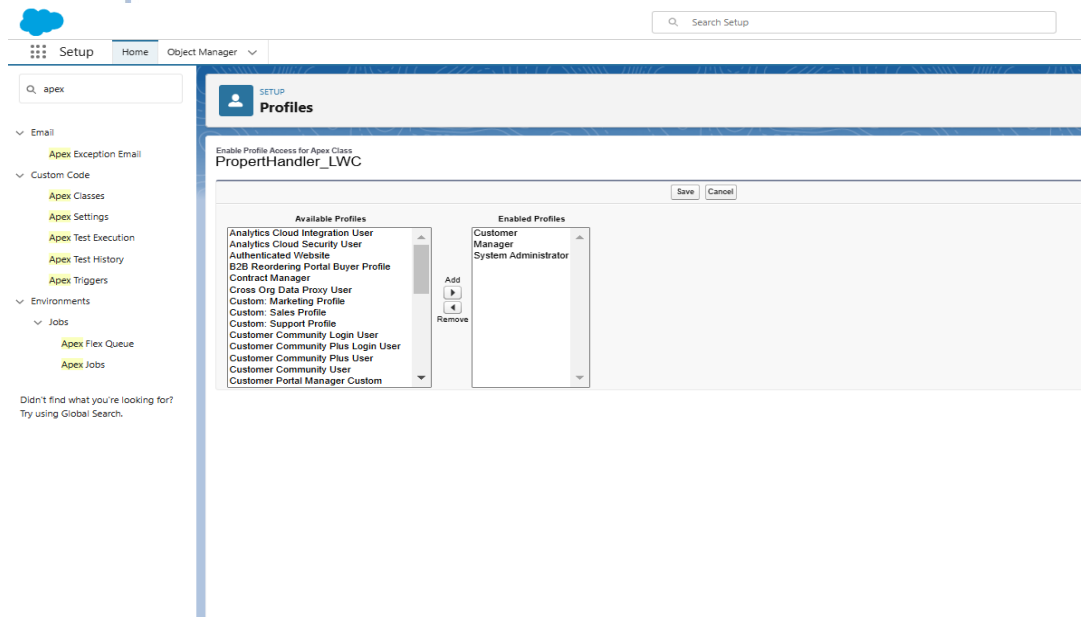
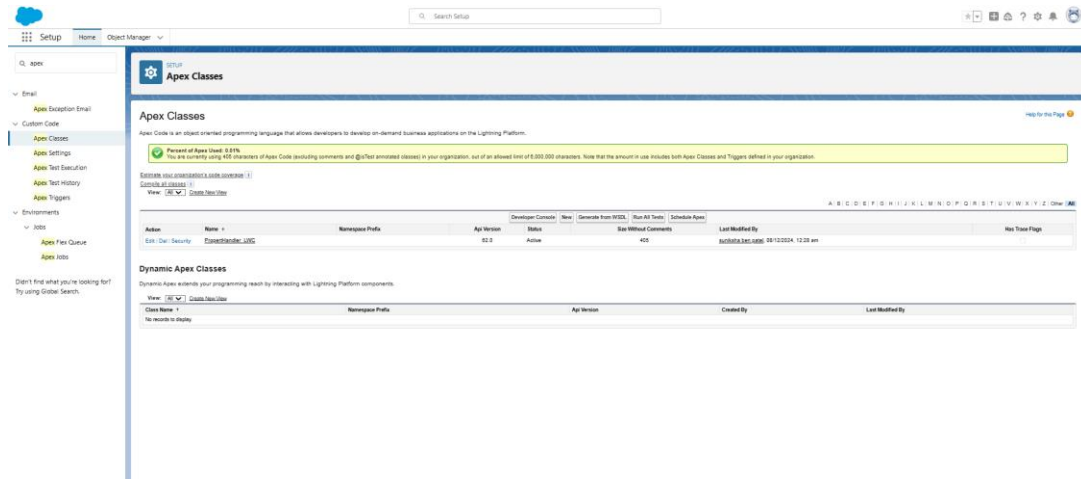


- User Interface Testing

1. LWC component functionality verification
2. App page layout and responsiveness



3. Role-based access control validation



6. Key Scenarios Addressed by Salesforce in the Implementation Project

Customer Registration Process

The customer registration process begins when a potential client visits the Dreams World Properties website and accesses the integrated JotForm. This form serves as the primary entry point for collecting essential customer information, including personal details, contact information, and property preferences. Upon form submission, the JotForm integration automatically triggers a record creation process in Salesforce, eliminating manual data entry and reducing potential errors.

The system processes the submitted information and creates a new customer record in the Salesforce database. During this creation, the system automatically assigns appropriate access levels and permissions based on predefined criteria. A custom profile is automatically associated with the new customer account, granting them basic access to the property search functionality. This automation ensures consistent handling of new customer registrations while maintaining security protocols.

After the initial setup, customers receive automated communications with their login credentials and basic system navigation instructions. The system also flags new accounts for review by customer service representatives, who can then verify the provided information and update the customer's verification status as needed.

Property Listing Process

The property listing workflow begins with the creation of a new property record in Salesforce. Property managers or authorized staff input comprehensive property details, including location, pricing, specifications, amenities, and availability status. Each property record contains both public-facing information and internal administrative data to facilitate efficient property management.

Once a property record is created, it automatically enters the approval workflow through a record-triggered flow. This automation eliminates the need for manual submission and ensures that all new properties undergo the required review

process. The system generates notifications to designated approvers (typically managers) who receive detailed information about the new property listing.

During the approval phase, managers review the property details for accuracy, compliance, and alignment with business standards. They can either approve the listing, making it visible to appropriate customer segments, or reject it with comments for revision. The approval process includes multiple checkpoints to ensure quality control and maintain listing standards. Upon approval, the system automatically updates the property's visibility status and makes it available for search based on predefined criteria.

Property Search Process

The property search functionality implements a sophisticated filtering system based on customer verification status. When customers log into the system, it automatically checks their verification status through a custom Lightning Web Component. This verification check is crucial as it determines which properties will be visible to the customer during their search.

For verified customers, the system provides access to the complete property database, including premium and exclusive listings. These customers can view detailed property information, high-resolution images, and specific location details. The search interface dynamically adjusts to show additional fields and features available only to verified users, providing them with a premium search experience.

Non-verified customers have access to a filtered version of the property database, showing only general listings and basic property information. While they can still search and view properties, certain premium features and detailed information are restricted. This tiered access system encourages customers to complete the verification process while still providing value to all users. The property search interface maintains consistency across both user types while dynamically adjusting the depth and breadth of information displayed based on verification status.

Each property listing in the search results includes basic information visible to all users, such as property type, general location, and price range. Additional details like exact address, contact information, and detailed specifications are displayed based on the user's verification status. This graduated access approach helps maintain property confidentiality while providing sufficient information for initial property evaluation.

7. Conclusion

Key Achievements

The successful implementation of the Salesforce platform for Dreams World Properties has resulted in significant operational improvements. The automated customer registration system has revolutionized the onboarding process, seamlessly capturing and processing customer data through JotForm integration. The implementation of role-based access control has established a secure, hierarchical system that appropriately restricts and grants access based on user roles and verification status.

The property approval workflow has transformed the property listing process, introducing automated submissions and structured review procedures. This is complemented by a customized property search functionality that delivers personalized experiences based on customer verification status, effectively balancing accessibility with security.

Realized Benefits

The automation of key processes has substantially reduced the time spent on manual data entry, allowing staff to focus on higher-value activities. Customer experience has seen marked improvement through personalized property access, with verified customers enjoying premium features while maintaining basic accessibility for all users. The implementation of role-based permissions has strengthened security protocols while streamlining access management.

The property approval process has become more efficient, with automated workflows reducing approval times and ensuring consistent review procedures. The customer verification system has created a clear distinction between verified and non-verified users, encouraging engagement while maintaining data security.

Strategic Recommendations

Looking ahead, several opportunities for enhancement have been identified. The implementation of additional automation for property updates would further reduce manual intervention and improve data accuracy. Enhanced reporting capabilities would provide deeper insights into customer behavior and property performance.

The development of mobile-responsive features would cater to the increasing mobile user base, improving accessibility and user experience. Integration with additional third-party services could expand functionality and create a more

comprehensive real estate management platform. These recommendations align with current market trends and would position Dreams World Properties for continued growth and improved customer service.

The successful implementation has laid a strong foundation for these future enhancements, with the modular design allowing for seamless integration of new features and capabilities.