

ClientVista

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

Project Overview

Client Vista Properties leverages Salesforce to enhance customer relationship management and optimize real estate operations. By integrating Salesforce with their website, Dreams World Properties automates the process of capturing customer interactions, preferences, and details in real-time. The system categorizes clients into approved and non-approved users, allowing personalized property recommendations for approved clients, while offering broader selections to others. This seamless integration not only streamlines operations but also elevates customer engagement, delivering a tailored experience that fosters growth in the competitive real estate market. With Salesforce's robust features, Client Vista Properties aims to improve efficiency, client satisfaction, and market reach.

Objectives of the CRM Application

The primary objective of the CRM (Customer Relationship Management) Application is to create a centralized platform that streamlines the management of client relationships and property-related requirements. This application aims to enhance customer service, improve communication, and optimize operational efficiency for real estate businesses. Specifically, the objectives include:

1. **Client Management:** To provide a comprehensive database for storing client information, including contact details, preferences, and communication history, enabling personalized and timely interactions.
2. **Property Management:** To facilitate the management of property listings, including detailed descriptions, pricing, availability, and related documentation, allowing for efficient tracking and updates.
3. **Lead Tracking:** To implement tools for tracking and managing leads, ensuring that potential clients are effectively followed up on and nurtured through the sales pipeline.
4. **Task Automation:** To automate routine tasks such as scheduling appointments, sending reminders, and generating reports, thereby allowing staff to focus on more strategic activities.
5. **Reporting and Analytics:** To provide analytical tools for generating reports on sales performance, client interactions, and property trends, aiding in data-driven decision-making.

6. **User-Friendly Interface:** To ensure the application is intuitive and accessible, allowing users to quickly adapt and maximize its features without extensive training.
7. **Integration Capabilities:** To enable seamless integration with existing tools and platforms (e.g., email, calendars, and marketing tools) to create a cohesive workflow.

Salesforce Tools Used:

Lightning Web Component (LWC): is a modern framework for building reusable, dynamic web interfaces in Salesforce. It's built on native web standards, allowing developers to create components with better performance, maintainability, and scalability.

Apex: It is a strongly typed, object-oriented programming language used by developers to execute flow and transaction control statements on the Salesforce platform. It is similar to Java and enables developers to add complex business logic to system events like button clicks, record updates, and Visualforce pages.

Approval Process in Salesforce is an automated process that allows records (such as leads, opportunities, contracts, or any custom object) to be submitted for approval based on specific criteria. It streamlines workflows by routing records to designated approvers and ensures that business-critical decisions are properly reviewed and approved by authorized users.

Flows Salesforce Flows are part of the Salesforce Flow Builder, which allows you to build applications and automate tasks by visually designing a process. Flows are highly flexible and can be used to update records, create records, delete records, send emails, and more based on user inputs, data, or events.

Client Feedback Mechanism via Email Alerts An automated system for sending feedback forms to clients post-transaction. This mechanism enhances customer engagement and allows for continuous improvement based on client responses.

Custom Dashboards and Reports Tools for visualizing key metrics and insights. Custom dashboards aggregate data from various sources, allowing stakeholders to monitor performance and make informed decisions. Reports provide detailed analyses of customer interactions and sales performance.

Other Tools:

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft, designed for developers and programmers. It supports a wide variety of programming languages, including JavaScript, Python, C++, and many others.

- **Salesforce CLI (Command Line Interface):** Used for managing and deploying Salesforce applications from the command line, making it easier to perform tasks like creating and managing scratch orgs, running tests, and deploying changes.
- **Salesforce Extensions for Visual Studio Code:** This extension pack provides powerful features for Salesforce development within VS Code, including syntax highlighting, code completion, and debugging capabilities for Apex, Lightning Web Components (LWC), and Visualforce.
- **Git:** A version control system that helps in managing code changes, facilitating collaboration among developers, and maintaining a history of code modifications.

Scope of the CRM Application

The scope of the CRM Application encompasses the following key areas:

1. **Target Users:**
 - Real estate agents and brokers
 - Property managers
 - Marketing teams
 - Administrative staff
2. **Core Features:**
 - **Client Database:** A robust database to store and manage client profiles, interaction history, and preferences.
 - **Property Listings:** A module to create, manage, and display property listings, including features for images, descriptions, and availability status.
 - **Lead Management:** Tools to capture, track, and nurture leads, including automated follow-up reminders and task assignments.
 - **Communication Tools:** Integrated messaging and email capabilities to facilitate communication between agents and clients.
 - **Reporting Dashboard:** A dashboard for visualizing key metrics, including lead conversion rates, sales performance, and client engagement levels.
3. **Integration and Compatibility:**
 - The application will be designed to integrate with third-party tools such as email services (e.g., Gmail, Outlook), calendar applications, and marketing platforms (e.g., Mailchimp).
 - The system will be compatible with both desktop and mobile devices to accommodate users' varying preferences for accessing the application.
4. **Data Security and Privacy:**
 - Implementation of robust security measures to protect sensitive client information, including data encryption, secure user authentication, and compliance with data protection regulations (e.g., GDPR).
 - Features to allow clients to manage their own data privacy settings and consent for communication.

5. Customization and Scalability:

- Scalability to support the growth of user bases and increase in data volume over time without compromising performance.

Detailed Steps to Solution Design:

Prerequisite Tools:

1. **Salesforce Org Access**
2. **Salesforce API Access**
3. **Development Tools**
 - Visual Studio Code (VS Code)
 - Salesforce CLI

1.Create Jotform and Integration with Salesforce

- **Created a Jotform** to capture customer details, including necessary fields such as name, email, and phone number.
- **Integrated with Salesforce** by navigating to the Jotform integrations section, selecting Salesforce, and mapping the form fields to corresponding Salesforce fields.

The image shows a Jotform titled "ClientVista" displayed on a light blue surface. The form is designed to capture customer details and is structured as follows:

- Name:** Two input fields for "First Name" and "Last Name".
- Email:** A single input field with a placeholder email address "jw@jotform.com".
- Phone Number:** A single input field with a placeholder "1000 000-0000" and a note "Please enter a valid phone number".
- Which type of Property are you looking for?:** Three checkboxes labeled "Residential", "Commercial", and "Rental".
- Budget Amount:** A single input field with a placeholder "e.g., 500".
- Address:** Multiple input fields for "Street Address", "Street Address (line 2)", "City", "State / Province", and "Postal / Zip Code".
- Submit:** A green button at the bottom of the form.

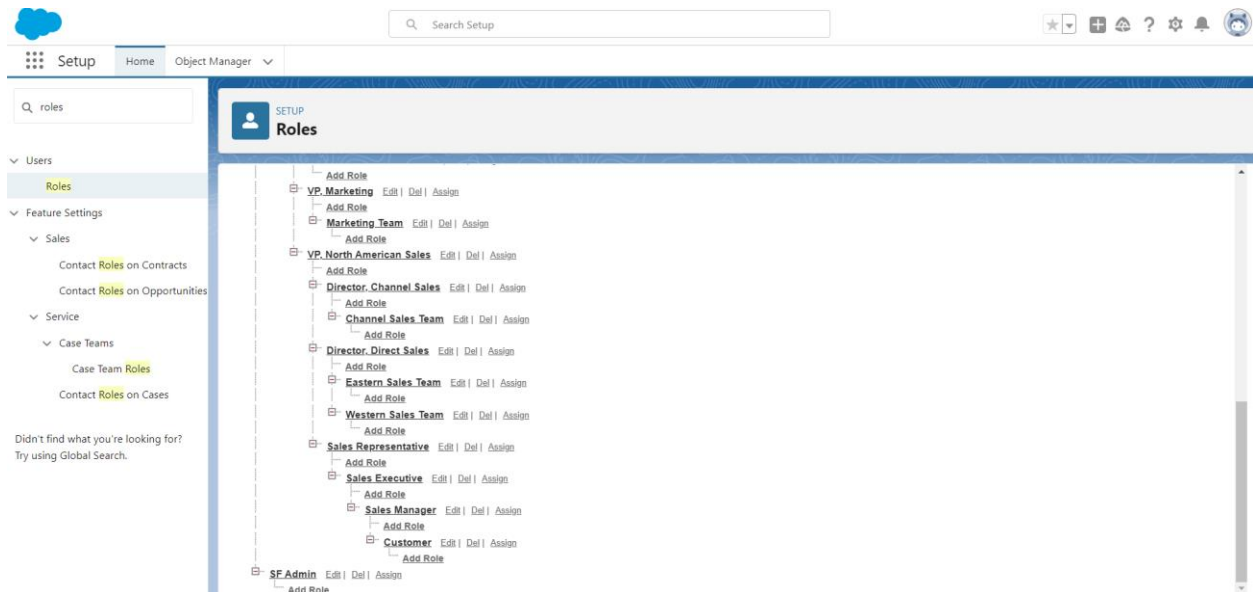
The form is set against a background featuring a small potted plant, a blue pencil, and a white pushpin on a light blue surface.

2.Create salesforce objects from spreadsheet

- **Download Customer Spreadsheet** containing the necessary customer data.
- **Create Customer Object** by navigating to Salesforce Object Manager, selecting "Create Object from Spreadsheet," and uploading the file.
- **Map the fields** in the spreadsheet to the corresponding Salesforce fields and save the object.
- **Create Property Object** by repeating the process using the customer object as a reference for the property data.

3.Creating Roles and User Profiles

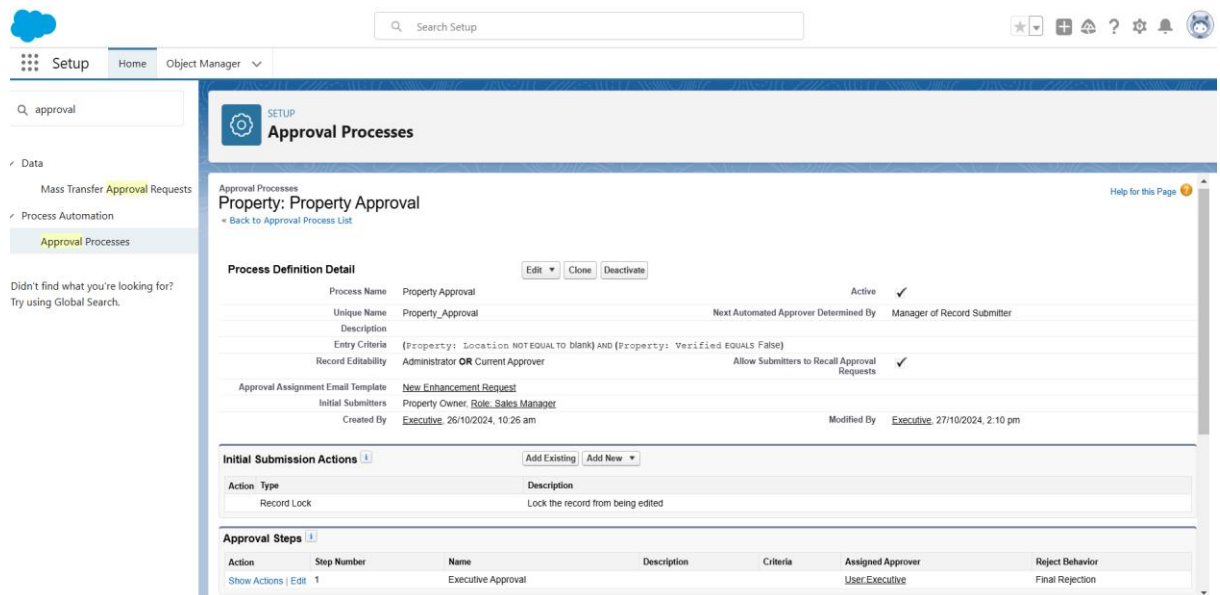
- Created a role hierarchy in Setup → Roles by adding Sales Executive under Sales Representative, Sales Manager under Sales Executive, and Customer under Sales Manager.
- Cloned the Salesforce Platform User profile, renamed it to "Customer," and adjusted permissions as necessary.
- **Created user accounts for each role:**
 - **Sales Manager**
 - **Sales Executive**
 - **Customer1**
 - **Customer2**



4. Property Approval Process Setup

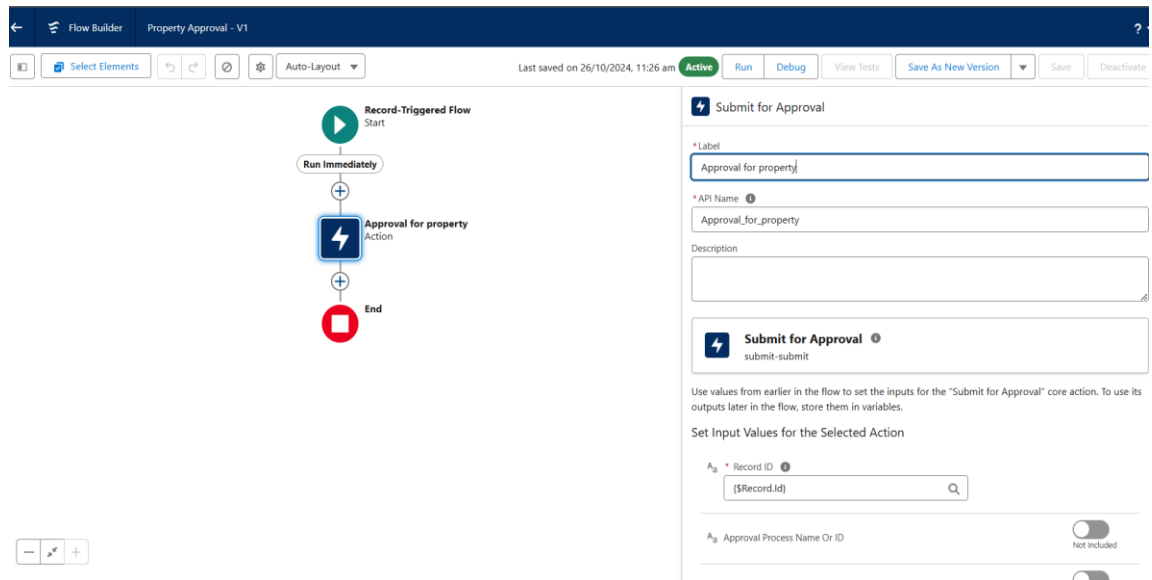
Purpose: Automate the approval flow for property records based on role hierarchy.

- Created an approval process called **Property Approval** in Process Automation.
- Configured submitters to include the Property Owner and Sales Manager, and added an Executive Approval step with the Sales Executive as the designated approver.



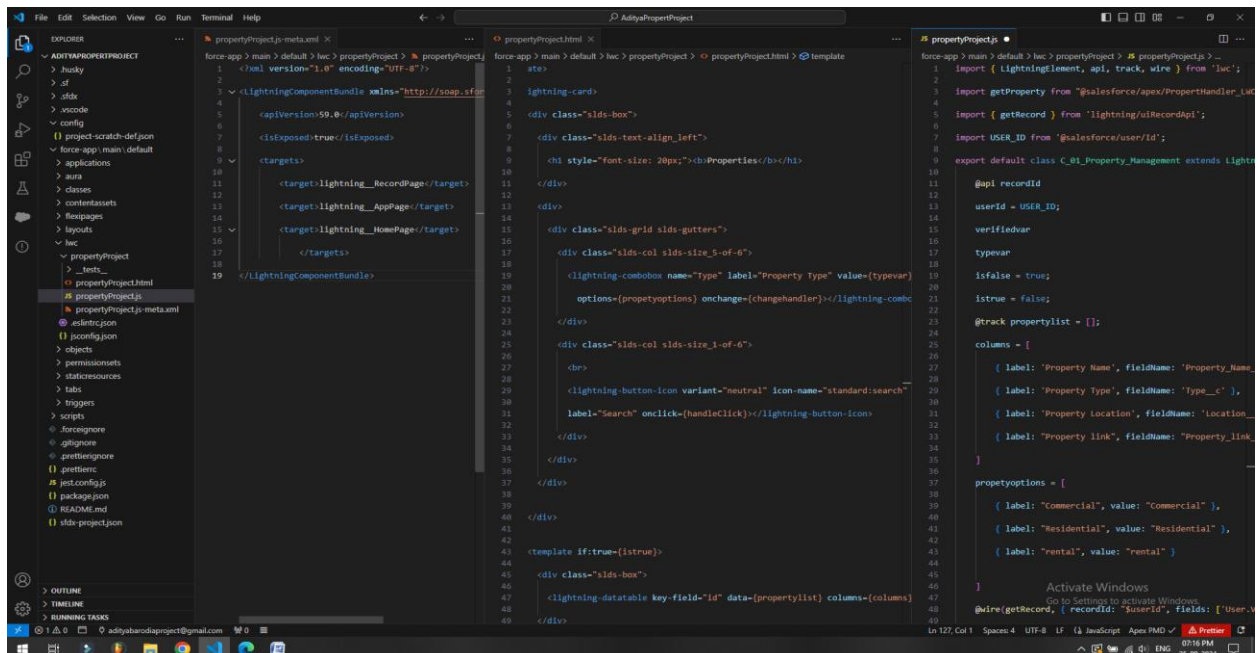
5. Record Trigger Flow for Automatic Approval Submission

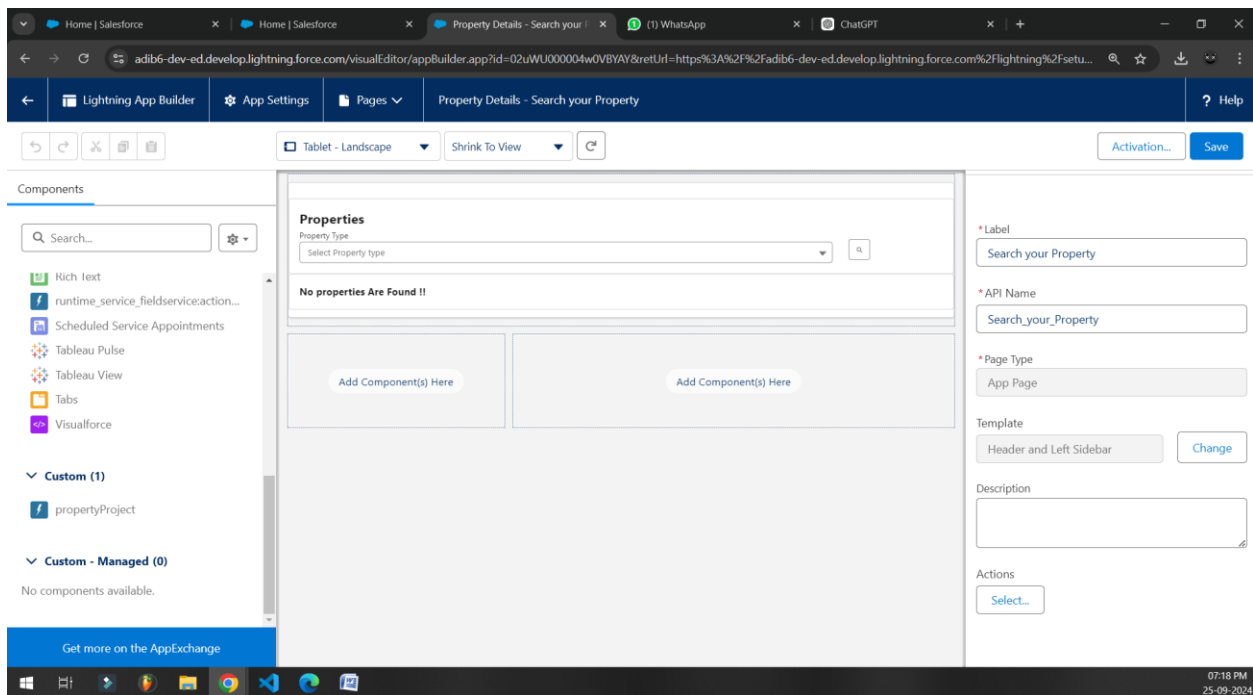
- Created a Record-Triggered Flow to automatically submit new Property records for approval upon creation.



6. Lightning Web App (with VS Code project integration)

Built an LWC component for property access based on customer verification status, along with an Apex class for secure data retrieval. Deployed the component to the "Search Your Property" page using the Lightning App Builder for user-friendly navigation.





7. Agent Object

Created an **Agent** object in Salesforce to manage relationships between properties and clients. The object includes the following key fields

- **Agent ID:** Unique identifier for each agent in the system.
- **Property ID:** Reference to the associated property.
- **Client ID:** Reference to the client linked to the property.
- **Date Assigned:** The date when the property was assigned to the client.
- **Transaction Status:** Indicates the current status of the transaction (e.g., Active, Completed, In Progress).

8. Client Feedback System Using Email Alert

- **Setup Email Alert:** Created an email alert in Salesforce to trigger automatically after a transaction is completed, ensuring clients receive immediate communication.
- **Create Email Template:** Designed a customizable email template that includes a personalized message and a link to the client feedback form for easy access.
- **Define Recipient:** Configured the email alert to automatically send to the client's email address associated with the transaction, ensuring feedback requests reach the correct recipient.

- **Trigger on Transaction Completion:** Implemented a trigger to automatically activate the email alert when a transaction is marked as complete, ensuring timely requests for client feedback.

The screenshot shows a web-based survey titled "Client Feedback" with a version dropdown set to "Version 1". The interface includes a top navigation bar with "View", "Send", and "Analyze" buttons. On the left, a sidebar lists "Pages" with options for "Welcome Page", "Page 1", and "Thank You Page". The main content area displays a "Standard Survey" with a background image of pebbles. The survey consists of two star-rating sections: "Rate your experience with the agent (1-5 stars)" and "Rate your experience with the property (1-5 stars)". Each section has five stars and "Bad" and "Good" labels. Below these is a text field for "Comments/Feedback on your experience" with a placeholder "Enter answer...". At the bottom of the browser window, a taskbar shows the temperature as 80°F and the time as 11:10 PM.

Email Alert:

The screenshot displays the Salesforce "Email Alert" configuration page for an alert named "clint feedback email alert". The page header includes the Salesforce logo, a search bar, and navigation links for "Switch to Lightning Experience", "Executive", "Setup", "Help", and "Content". The left sidebar contains navigation options like "Home", "Chatter", "Libraries", "Content", "Subscriptions", and "Enhancement Requests", along with a "Lightning Experience Transition Assistant" and "Salesforce Mobile Quick Start" sections. The main content area shows the "Email Alert Detail" for the selected alert, including fields for Description, Unique Name, From Email Address, Recipients, Additional Emails, Created By, and Modified By. Below this, there are sections for "Rules Using This Email Alert" and "Approval Processes Using This Email Alert".

Email Alert Detail	
Description	clint feedback email alert
Unique Name	Client_feedback
From Email Address	Current User's email address
Recipients	User: Customer2 User: Customer
Additional Emails	21pa1a0518@vishnu.edu.in anushabhuchupalli@gmail.com
Created By	Executive, 27/10/2024, 12:21 pm
Modified By	Executive, 27/10/2024, 12:30 pm

Recorded Trigger:

Flow Builder

Email_alert - V1

?

Select Elements

↶ ↷

⌂

⚙️

Auto-Layout

Last saved on 27/10/2024, 12:36 pm

Active

Run

Debug

View Tests

Save As New Version

Save

Deactivate

Record-Triggered Flow

Start

Run Immediately

+

Feedback_email

Email Alert

+

End

✉️ Clint feedback email alert

✕

* Label

Feedback_email

* API Name ⓘ

Feedback_email

Description

✉️ clint feedback email alert ⓘ

emailAlert-agents_data_c.Client_feedback

Use values from earlier in the flow to set the inputs for the "clint feedback email alert" email alert. To use its outputs later in the flow, store them in variables.

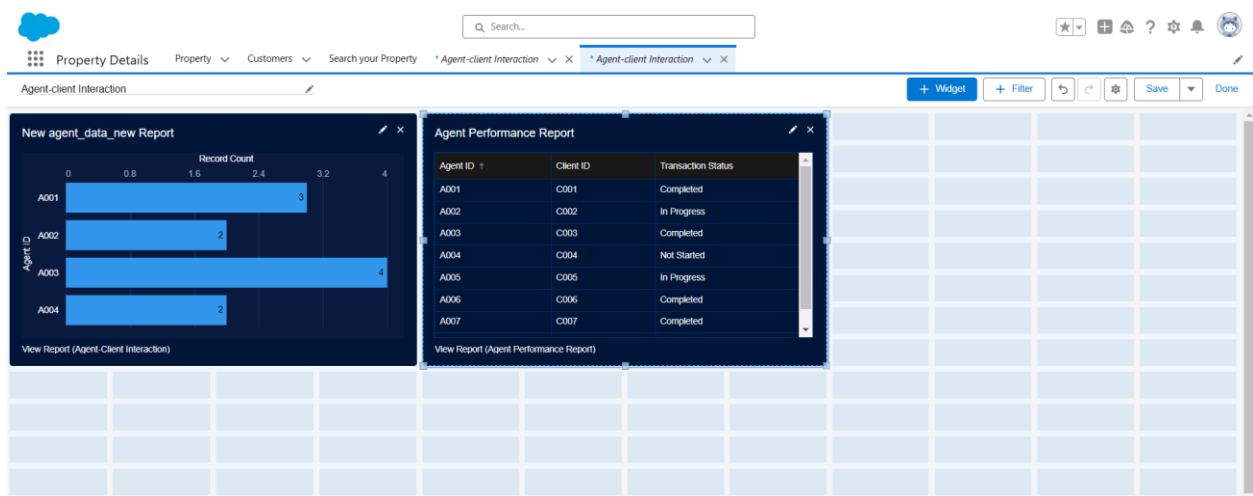
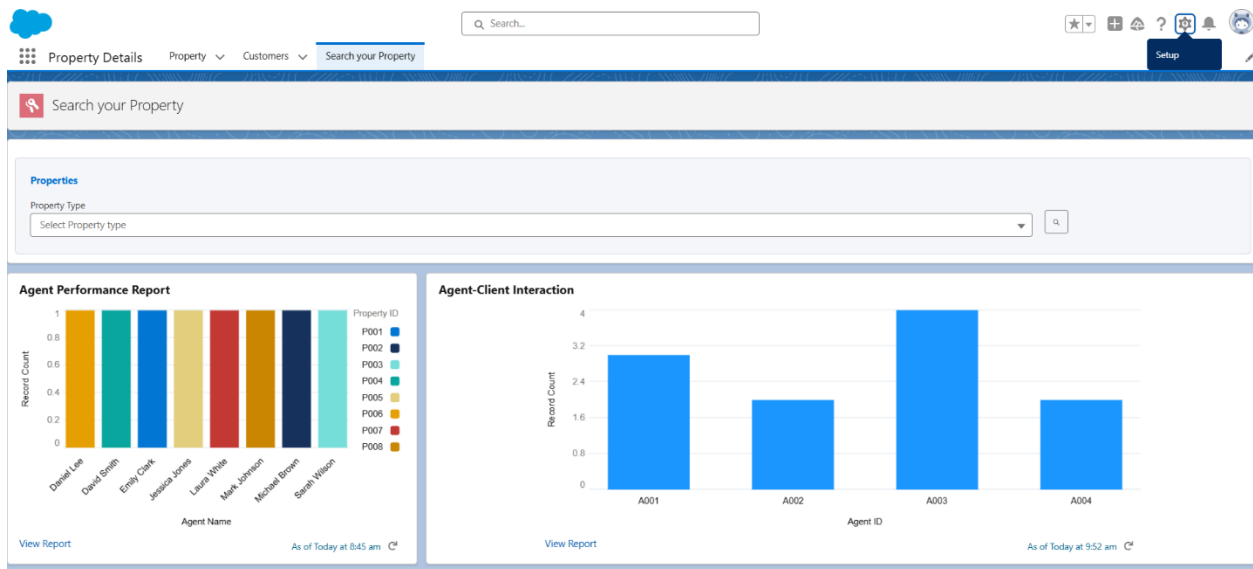
Set Input Values for the Selected Action

A_a * Record ID ⓘ

\$Record

Creating Reports and Dashboards

- Agent Property Sales Report:** Developed a report to track the number of properties sold by each agent, providing insights into individual performance and sales effectiveness.
- Agent-Client Communication Report:** Created a report to monitor interactions between agents and clients, tracking the frequency and types of communications to ensure effective engagement.



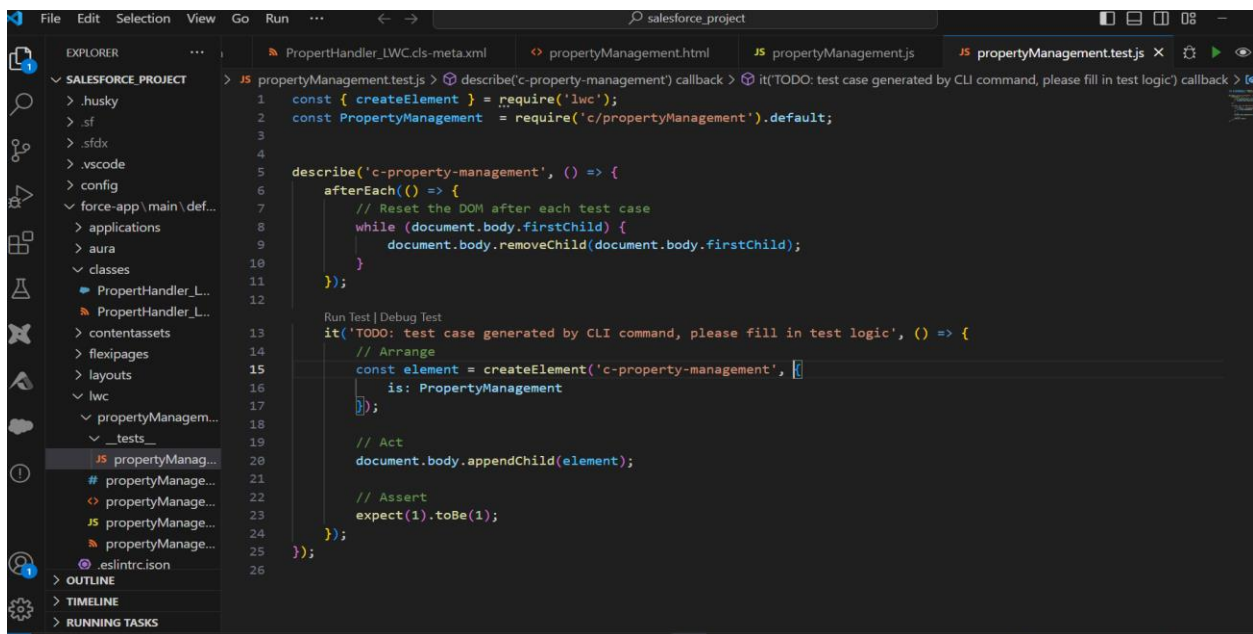
Testing and Validation

- **Unit Testing:** Developed unit tests for the PropertyHandler_LWC class, ensuring methods like getProperty returned expected results and achieved over 80% code coverage for deployment.

- **User Interface Testing:** Tested the "Search Your Property" Lightning App page with end-users, confirming verified customers accessed the correct properties and all functionalities were intuitive.

Jest and LWC Component Testing in VS Code

- **Set Up Jest:** Install Jest and related packages in your project using `npm install --save-dev jest @salesforce/sfdx-lwc-jest`.
- **Create Test Files:** Add a test file in the `__tests__` folder for your LWC component to write your test cases.
- **Run Tests:** Use `npm run test:unit` in the terminal to execute your tests and check for correct functionality.



The screenshot shows the Visual Studio Code editor with a project named 'salesforce_project'. The Explorer sidebar on the left shows the project structure, including a folder named 'lwc' and a sub-folder 'propertyManagement'. The file 'propertyManagement.test.js' is selected and open in the editor. The code in the file is as follows:

```

1  const { createElement } = require('lwc');
2  const PropertyManagement = require('c/propertyManagement').default;
3
4
5  describe('c-property-management', () => {
6
7      afterEach(() => {
8          // Reset the DOM after each test case
9          while (document.body.firstChild) {
10             document.body.removeChild(document.body.firstChild);
11         }
12     });
13
14     it('TODO: test case generated by CLI command, please fill in test logic', () => {
15         // Arrange
16         const element = createElement('c-property-management', {
17             is: PropertyManagement
18         });
19
20         // Act
21         document.body.appendChild(element);
22
23         // Assert
24         expect(1).toBe(1);
25     });
26

```

Key Scenarios Addressed by Salesforce in the Implementation Project

- **Streamlined Customer Onboarding:** Implemented Jotform integration to automate the collection of customer details, allowing for immediate record creation in Salesforce and reducing administrative overhead.
- **Dynamic Role and User Management:** Established a clear role hierarchy (Sales Executive, Sales Manager, Customer) to ensure appropriate access levels, enhancing security and operational efficiency.
- **Automated Property Approval Workflow:** Developed an approval process for property records, allowing for swift review and decision-making, which improved the overall transaction flow.
- **Automated Email Alerts for Client Feedback:** Created an email alert system that triggers automatically after a transaction is completed, sending clients a feedback form link to gather valuable insights about their experience.
- **Insightful Reporting and Dashboards:** Designed comprehensive reports and dashboards to track property metrics, agent performance, and client interactions, facilitating informed decision-making and strategy development.

Conclusion

In conclusion, the CRM Application is designed to address the specific needs of real estate professionals by providing a robust tool for managing client relationships and property-related tasks. By enhancing efficiency, improving communication, and enabling data-driven decision-making, the application aims to support the overall growth and success of real estate businesses.