

LEASE MANAGEMENT

1. Project Overview

This project aims to develop a Salesforce-based application designed to manage and streamline the lease management process for real estate properties. The primary goal is to enhance the efficiency of lease tracking, coordinate lease renewals, manage tenant relationships, and generate insightful reports on occupancy, lease expirations, and rental income.

2. Objectives

a) Business Goals:

- **Optimize Lease Tracking:** Ensure all lease agreements, renewals, and expirations are easily accessible and trackable within a single Salesforce-based platform, reducing administrative overhead.
- **Enhance Tenant Relationship Management:** Streamline communication with tenants, automate reminders for lease renewals, and resolve tenant inquiries faster, thereby improving tenant satisfaction.
- **Increase Operational Efficiency:** Automate routine processes such as rent invoicing, lease renewals, and maintenance requests, reducing manual input and minimizing errors.
- **Improve Reporting and Insights:** Provide accurate, real-time reports on occupancy rates, lease expirations, rental income, and other key metrics to support strategic decision-making.
- **Ensure Compliance and Reduce Risk:** Track lease obligations and ensure compliance with regulatory standards, minimizing the risk of missed renewals, late fees, or legal issues.

b) Specific Outcomes:

- **Centralized Lease Management System:** A comprehensive dashboard that consolidates all lease-related information, including lease terms, tenant contact information, renewal dates, and payment schedules.
- **Automated Notifications and Reminders:** Scheduled reminders and alerts for upcoming lease expirations, rent payment due dates, and maintenance requirements, enabling proactive management.
- **Tenant Portal:** A self-service portal where tenants can view lease details, make payments, submit maintenance requests, and access important documents.
- **Reports and Analytics Dashboard:** Customizable reports and analytics on key metrics, such as occupancy rates, tenant turnover, lease renewal rates, and rental income forecasts.
- **Document Management and Compliance Tracking:** A secure repository for all lease documents, contracts, and compliance checklists, with version control and audit trails to ensure all records are up to date.
- **Mobile Accessibility:** A mobile-responsive interface to allow property managers and tenants to access lease information, communicate, and complete tasks from any device.

3. Salesforce Key Features and Concepts Utilized in Lease Management:

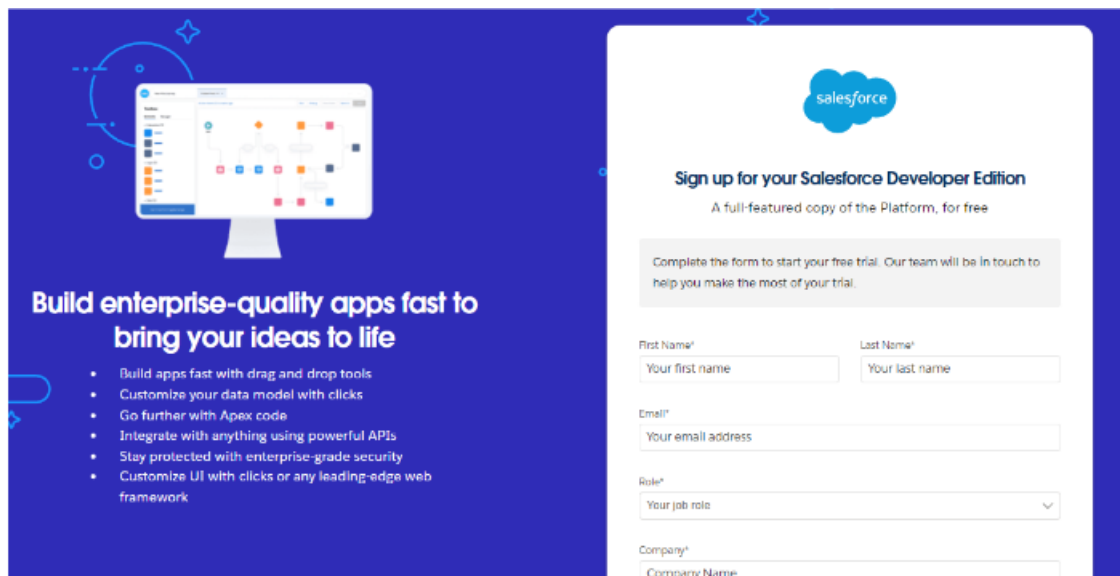
- i. **Custom Objects:** Structures lease-specific data (e.g., "Leases," "Properties," "Tenants") for organized management.
- ii. **Process Builder & Automation:** Automates lease renewals, rent notifications, and maintenance alerts, reducing manual work.

- iii. **Email Alerts:** Sends notifications for lease renewals and payment due dates to keep stakeholders informed.
- iv. **Reports & Dashboards:** Provides real-time insights into occupancy, income, and tenant trends for better decision-making.
- v. **Lightning App Builder:** Customizes user interfaces to display relevant lease and tenant details for easy access.
- vi. **Salesforce Mobile App:** Offers access to lease information and tasks on-the-go for property managers and tenants.
- vii. **Security & Permissions:** Controls data access through role-based permissions to protect sensitive lease data.
- viii. **Document Management:** Stores lease documents securely with version control and easy retrieval.
- ix. **Experience Cloud (Tenant Portal):** Provides tenants with a portal to access lease details, payments, and requests.
- x. **Einstein Analytics:** Delivers advanced insights on tenant retention, property performance, and lease optimization.
- xi. **AppExchange Integrations:** Expands functionality with tools for payments, accounting, and tenant screening.

4. Detailed Steps to Solution Design

a) Creating Developer Account

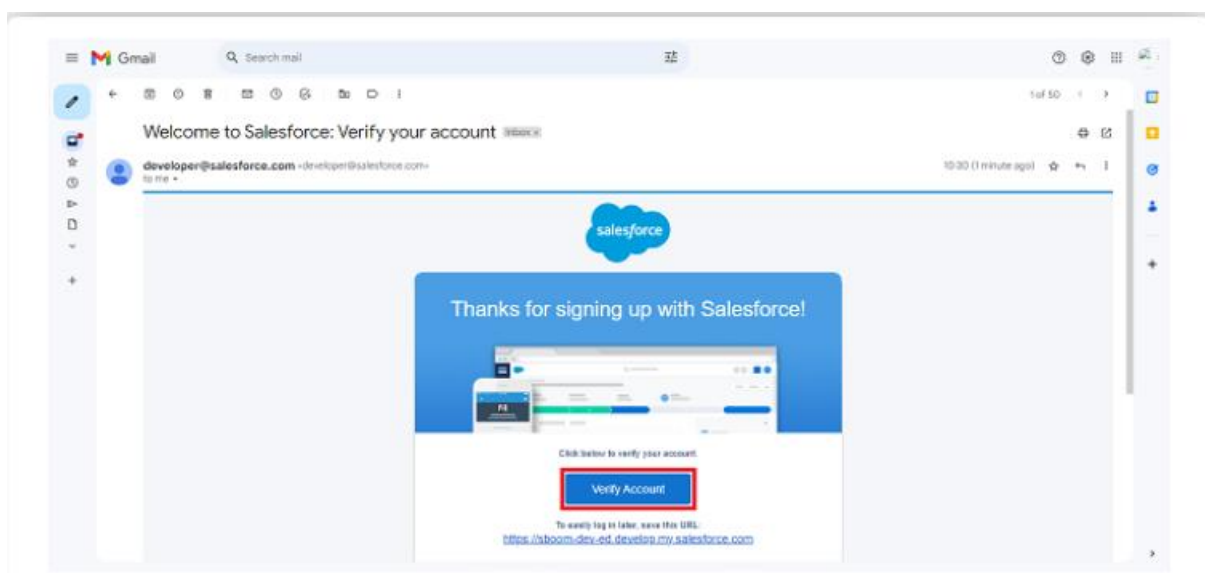
- Creating a developer org in salesforce.
 1. Go to <https://developer.salesforce.com/signup>
 2. Click Sign Up, provide your details (name, email, company, username), and register.



Account Activation:

Check your email for the activation link.

Click the link, set up your password and security question, and log in to the Salesforce Developer account.



Enter a new password for
screenfflowdistribution@superbadgeunit.com. Make
sure to include at least:

- ✓ 8 characters
- ✓ 1 letter
- ✓ 1 number

* New Password

..... Good

* Confirm New Password

..... Match

Security Question

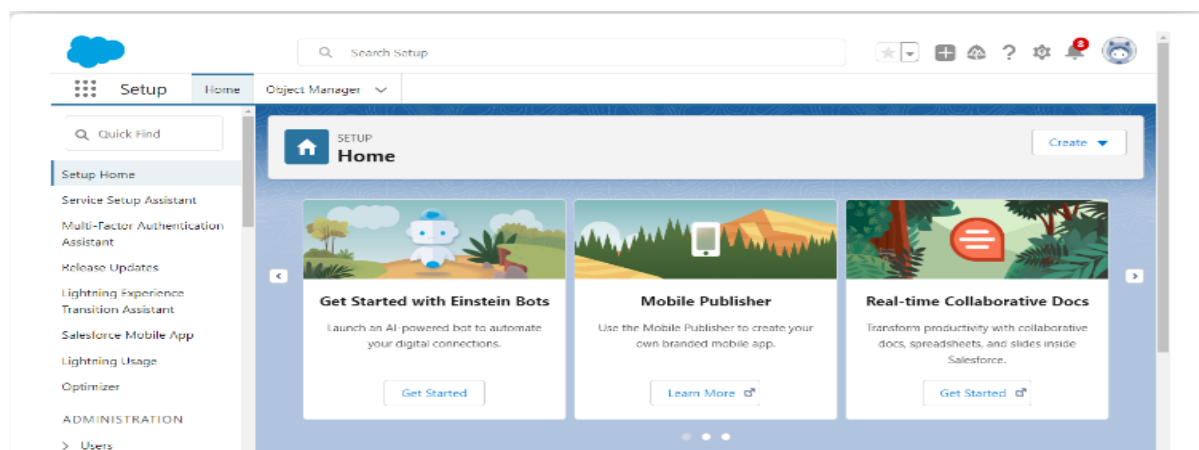
▼ In what city were you born?

* Answer

.....

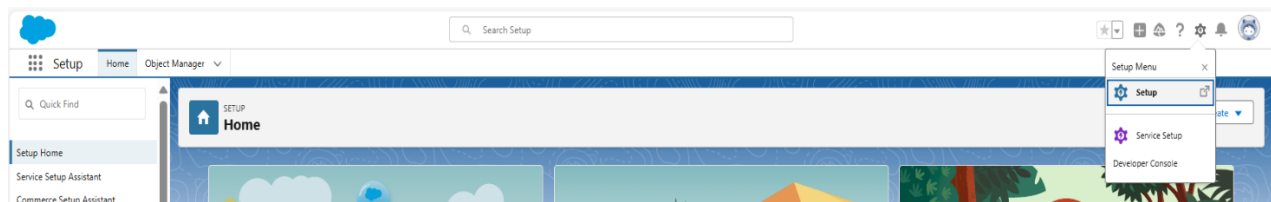
Change Password

- Click on Verify Account
- Give a password and answer a security question and click on change password.
- Give a password and answer a security question and click on change password.
- Then you will redirect to your salesforce setup page.



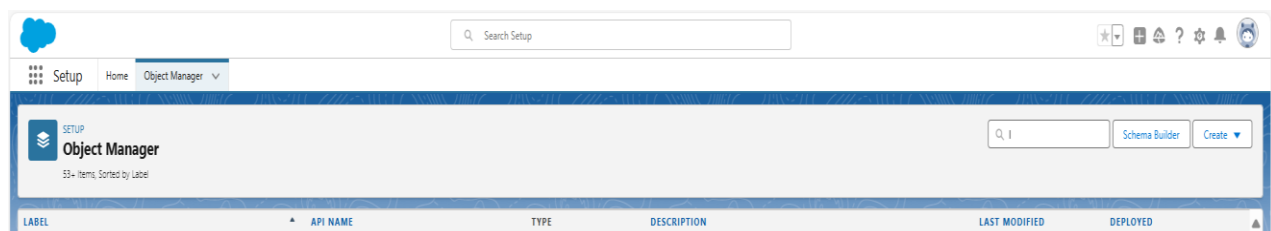
b) Object Creation

Custom objects are used to manage lease-related data.



- **Create Property Object**

- Navigate to **Setup** → **Object Manager** → **Create Object** → Select **Custom Object**.
- Name it **Property** to store information about rental properties.
- Add standard fields like **Property Name**, **Location**, **Type**, and **Rental Cost**.



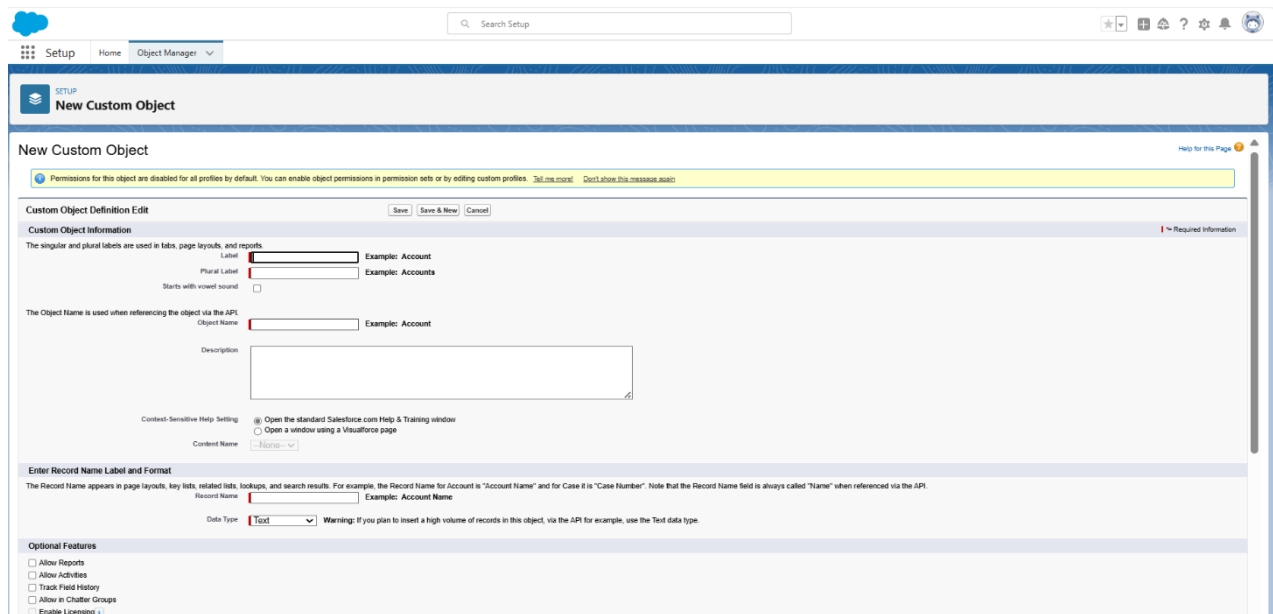
- **Create Tenant Object**

- Create a **Tenant** object to manage tenant details.
- Fields include **Tenant Name**, **Email**, **Phone**, and **Current Lease** (lookup to Lease object).

- **Create Payment Object**

- Create a **Payment** object to track tenant payments.

- Fields include **Payment Date**, **Amount**, **Status**, and **Tenant** (lookup to Tenant object).
- **Create Lease Object**
 - Create a **Lease** object to store lease agreement details.
 - Fields include **Lease Start Date**, **End Date**, **Property** (lookup to Property object), and **Tenant** (lookup to Tenant object).



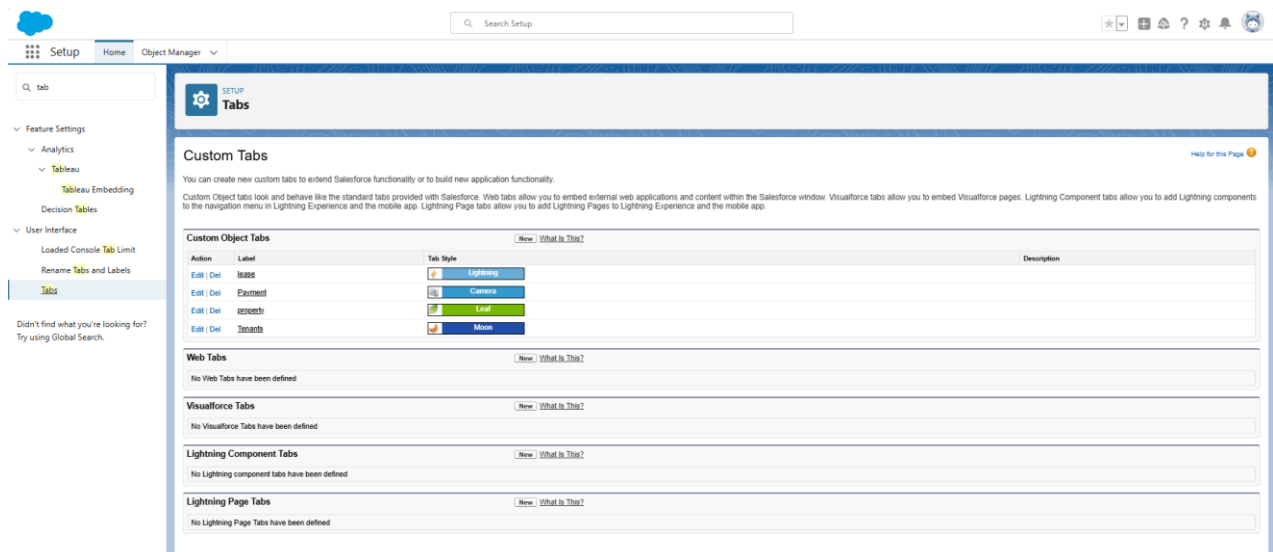
The screenshot displays the 'New Custom Object' configuration page in Salesforce. At the top, there's a navigation bar with 'Setup', 'Home', and 'Object Manager'. Below this, the 'New Custom Object' section is active. A yellow banner at the top of the main content area states: 'Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more](#) [Click to learn this message again](#)'. The main form is titled 'Custom Object Definition Edit' and includes a 'Save' button. The 'Custom Object Information' section contains fields for 'Label' (with an example 'Account'), 'Plural Label' (with an example 'Accounts'), 'Object Name' (with an example 'Account'), and a 'Description' text area. There are also checkboxes for 'Starts with vowel sound' and 'Context-Sensitive Help Setting'. Below this, the 'Enter Record Name Label and Format' section has a 'Record Name' field (example 'Account Name') and a 'Data Type' dropdown set to 'Text'. A warning note states: 'Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.' The 'Optional Features' section at the bottom has checkboxes for 'Allow Reports', 'Allow Activities', 'Track Field History', 'Allow in Chatter Groups', and 'Enable Licensing'.

c) Tabs

- **Creating a Custom Tab**
 - Navigate to **Setup** → **Tabs** → **New Custom Tab**.
 - Select the object (e.g., Property) to create a tab for easy access.
 - Add tabs for other objects like Tenant, Payment, and Lease.
- **Creating Remaining Tabs**
 - Repeat the process for remaining custom objects.
 - Customize tab icons and order for better navigation.

- **Create a Lightning App**

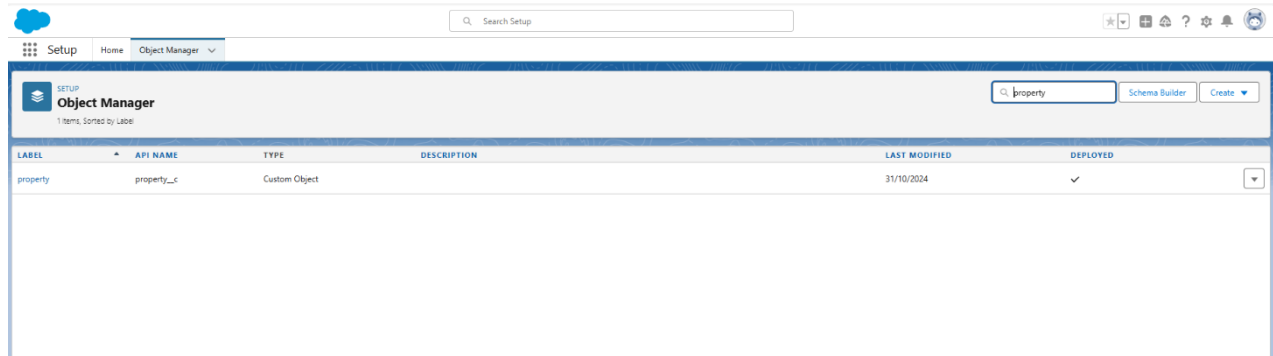
- Go to **Setup** → **App Manager** → **New Lightning App**.
- Name the app **Lease Management**.
- Add created tabs (Property, Tenant, Payment, Lease) and customize navigation for users.



d) Fields

- **Creation of Fields**

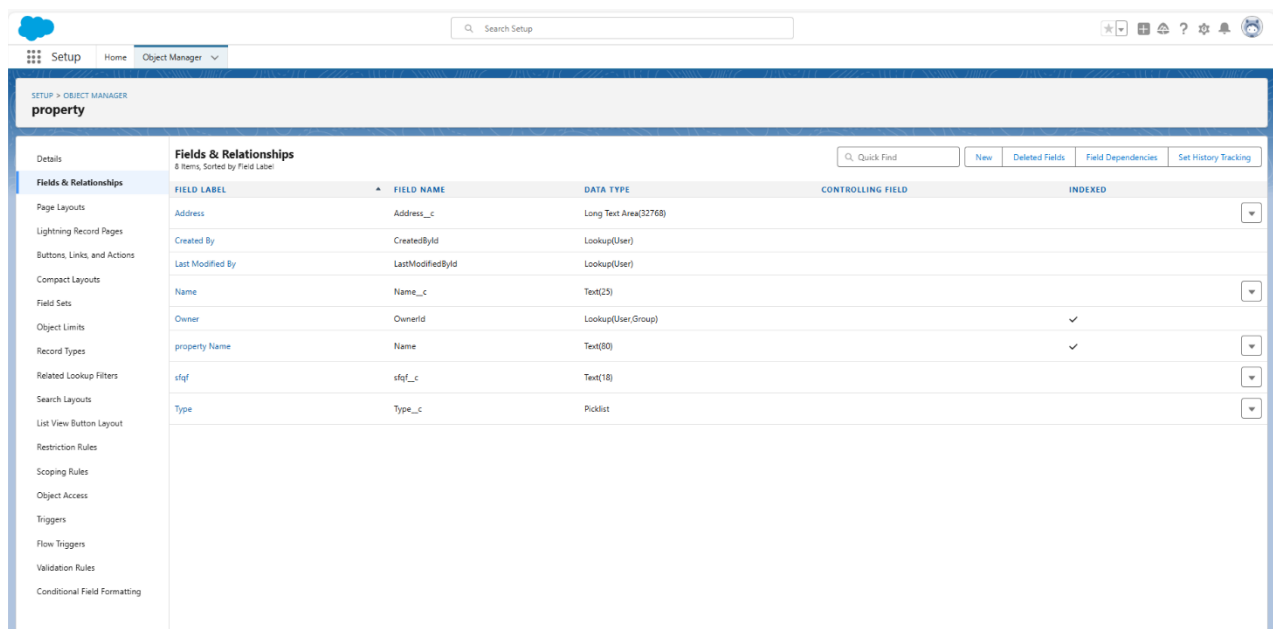
- **Property Object:** Add fields for **Property Size**, **Number of Units**, and **Owner Contact**.
- **Tenant Object:** Add fields like **Date of Move-In**, **Preferred Payment Method**, and **Lease Duration**.
- **Lease Object:** Include **Monthly Rent**, **Deposit Amount**, and **Payment Status**.
- **Payment Object:** Add fields such as **Transaction ID** and **Payment Confirmation**



LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
property	property__c	Custom Object		31/10/2024	✓

• Creation of Lookup Fields

- Create lookup relationships between objects (e.g., Tenant to Lease, Lease to Property, Payment to Tenant).

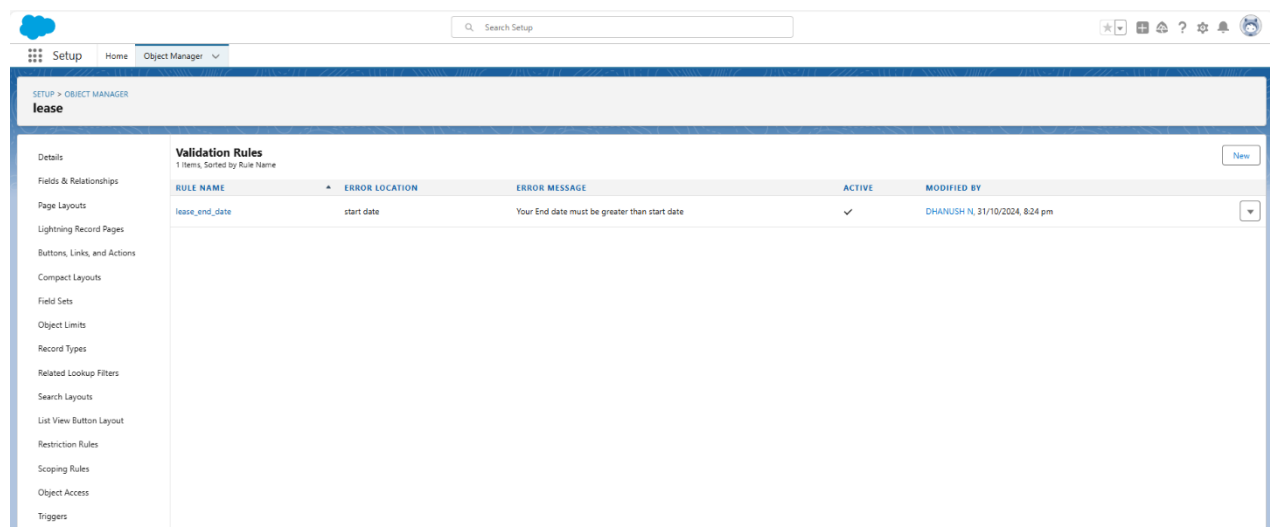


FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(25)		
Owner	OwnerId	Lookup(User/Group)		✓
property Name	Name	Text(80)		✓
staf	staf__c	Text(18)		
Type	Type__c	Picklist		

e) Validation Rules

- **Validation for Lease Object**

- Navigate to **Validation Rules** under the Lease object.
- Example Rule: Ensure lease end date is after the start date.
 - **Condition:** Lease_End_Date__c < Lease_Start_Date__c
 - Add an error message: "End date must be after start date."



f) Email Templates

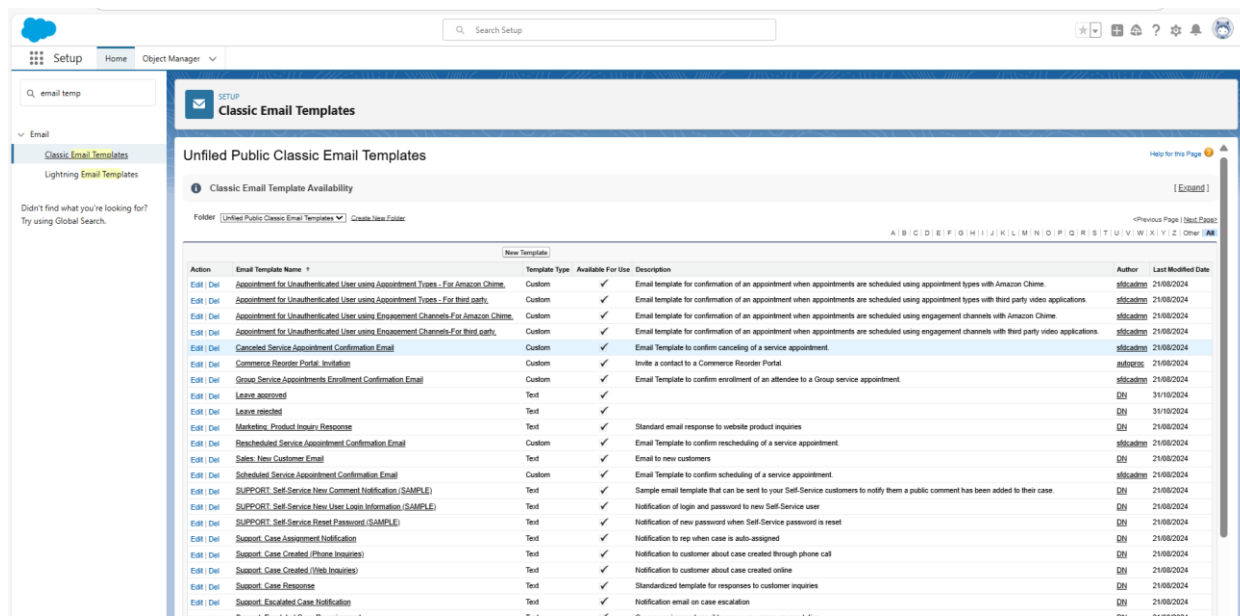
- **Templates for Communication**

- I. **Tenant Leaving Notification**

- **Purpose:** To inform the tenant about the end of their lease term and the actions required.
- **Use Case:** When a tenant is preparing to leave or has given notice that they will vacate the property.

II. Lease Approval Notification

- **Purpose:** To notify tenants and property managers about the approval of a lease agreement.
- **Use Case:** After a tenant applies for a lease and the application is approved



The screenshot shows the Salesforce Classic Email Templates interface. The left sidebar contains navigation links for Setup, Home, and Object Manager. The main content area is titled 'Classic Email Templates' and displays a table of 'Unfiled Public Classic Email Templates'. The table has columns for Action, Email Template Name, Template Type, Available For Use, Description, Author, and Last Modified Date. The table lists various email templates, including appointment confirmations, service appointment confirmations, and support case notifications.

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit Del	Appointment for Unauthenticated User using Appointment Types - For Amazon Chime	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with Amazon Chime	sfscadm	21/08/2024
Edit Del	Appointment for Unauthenticated User using Appointment Types - For third party	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with third party video applications.	sfscadm	21/08/2024
Edit Del	Appointment for Unauthenticated User using Engagement Channels For Amazon Chime	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using engagement channels with Amazon Chime.	sfscadm	21/08/2024
Edit Del	Appointment for Unauthenticated User using Engagement Channels For third party	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using engagement channels with third party video applications.	sfscadm	21/08/2024
Edit Del	Canceled Service Appointment Confirmation Email	Custom	✓	Email Template to confirm canceling of a service appointment.	sfscadm	21/08/2024
Edit Del	Commerce Recorder Portal: Invitation	Custom	✓	Invite a contact to a Commerce Recorder Portal.	sfscadm	21/08/2024
Edit Del	Group Service Appointments Enrollment Confirmation Email	Custom	✓	Email Template to confirm enrollment of an attendee to a Group service appointment.	sfscadm	21/08/2024
Edit Del	Leave accepted	Text	✓		DN	31/10/2024
Edit Del	Leave rejected	Text	✓		DN	31/10/2024
Edit Del	Marketing: Product Inquiry Response	Text	✓	Standard email response to website product inquiries	DN	21/08/2024
Edit Del	Rescheduled Service Appointment Confirmation Email	Custom	✓	Email Template to confirm rescheduling of a service appointment.	sfscadm	21/08/2024
Edit Del	Sales: New Customer Email	Text	✓	Email to new customers	DN	21/08/2024
Edit Del	Scheduled Service Appointment Confirmation Email	Custom	✓	Email Template to confirm scheduling of a service appointment.	sfscadm	21/08/2024
Edit Del	SUPPORT: Self-Service New Comment Notification (SAMPLE)	Text	✓	Sample email template that can be sent to your Self-Service customers to notify them a public comment has been added to their case.	DN	21/08/2024
Edit Del	SUPPORT: Self-Service New User Login Information (SAMPLE)	Text	✓	Notification of login and password to new Self-Service user	DN	21/08/2024
Edit Del	SUPPORT: Self-Service Reset Password (SAMPLE)	Text	✓	Notification of new password when Self-Service password is reset	DN	21/08/2024
Edit Del	Support Case Assignment Notification	Text	✓	Notification to rep when case is auto-assigned	DN	21/08/2024
Edit Del	Support Case Created (Phone Inquiries)	Text	✓	Notification to customer about case created through phone call	DN	21/08/2024
Edit Del	Support Case Created (Web Inquiries)	Text	✓	Notification to customer about case created online	DN	21/08/2024
Edit Del	Support Case Response	Text	✓	Standardized template for responses to customer inquiries	DN	21/08/2024
Edit Del	Support Escalated Case Notification	Text	✓	Notification email on case escalation	DN	21/08/2024
Edit Del	Support Escalated Case Notification	Text	✓	Case escalation email to new case owner on escalation	DN	21/08/2024

III. Lease Rejection Notification

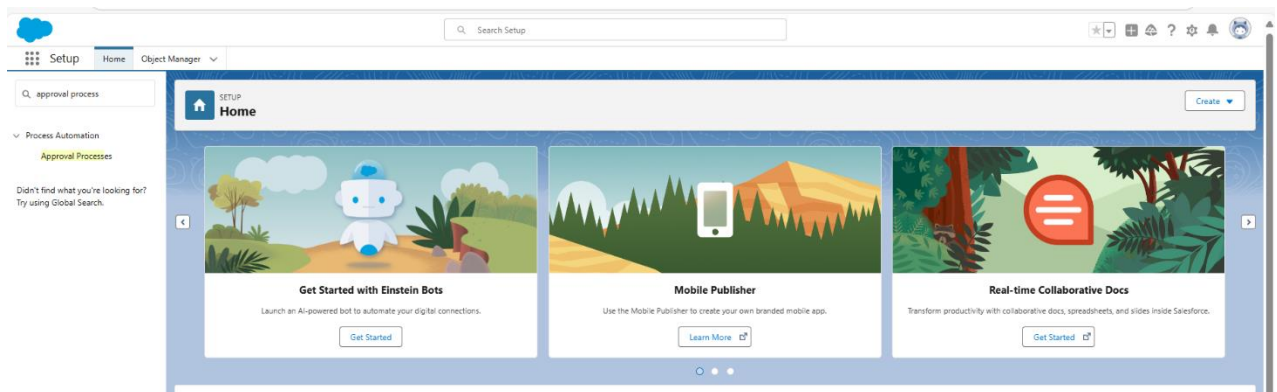
- **Purpose:** To inform a tenant that their lease application has been rejected.
- **Use Case:** When a tenant's lease application does not meet the necessary criteria or is denied for any reason.

IV. Monthly Payment Reminder

- **Purpose:** To remind tenants of upcoming rent payments or overdue payments.
- **Use Case:** Sends reminders about rent due dates, outstanding payments, or upcoming payment deadlines.

V. **Successful Payment Confirmation**

- **Purpose:** To confirm the receipt of a tenant's monthly rent payment.
- **Use Case:** Sent after a payment is successfully processed.



g) **Approval Process: Lease Vacancy Check**

- **Purpose:** The approval process ensures that properties are flagged for vacancy status and appropriately managed within the system.
 - **Initial Submission Action:**
 - When a property is identified as vacant, a property manager submits a request for approval. The request includes property details, reason for vacancy, and suggested next steps.
 - **Final Approval Action:**
 - The designated approver (e.g., senior property manager) reviews the vacancy request and, if valid, approves it. The system updates the vacancy status and triggers further actions like property re-listing or tenant search.

- **Final Rejection Action:**

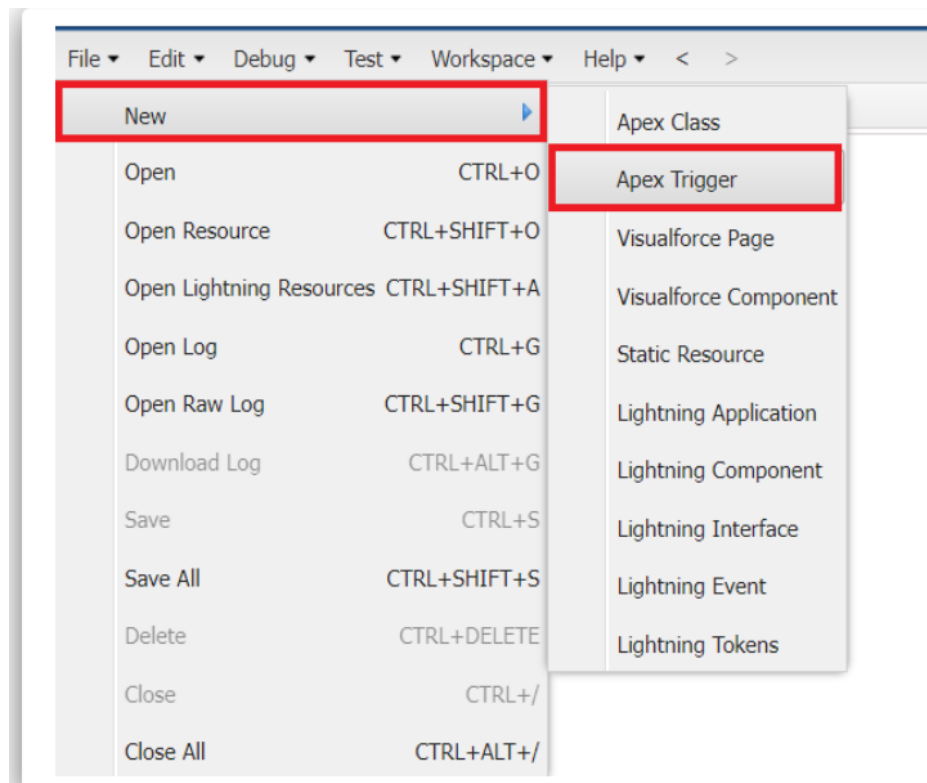
- If the request is not approved (e.g., tenant is still in the process of moving out), the status is updated as "Pending" or "Rejected," and the appropriate steps are taken to follow up with the tenant or property manager.

h) Apex Trigger: Lease Expiration Notification

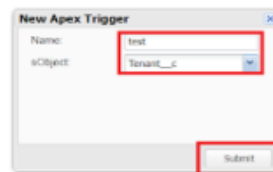
- **Purpose:** To automatically send notifications to tenants and property managers when a lease is nearing expiration or requires renewal.

a) Create an Apex Trigger:

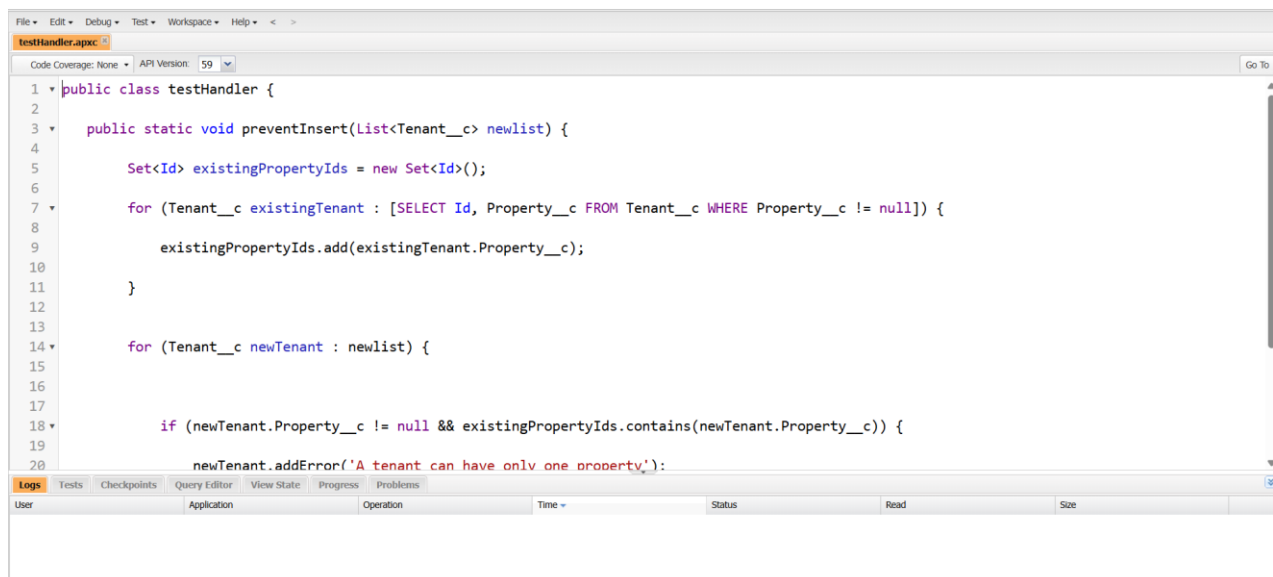
- The trigger is set to fire when a lease record is created or updated. It checks for lease expiration dates and compares them to the current date. If the expiration is within a predefined threshold (e.g., 30 days), it triggers an action to notify relevant parties.



- Give the Apex Trigger name as “test”, and select “Tenant__c” from the dropdown for sObject.



- Now the code logic can be written here



```

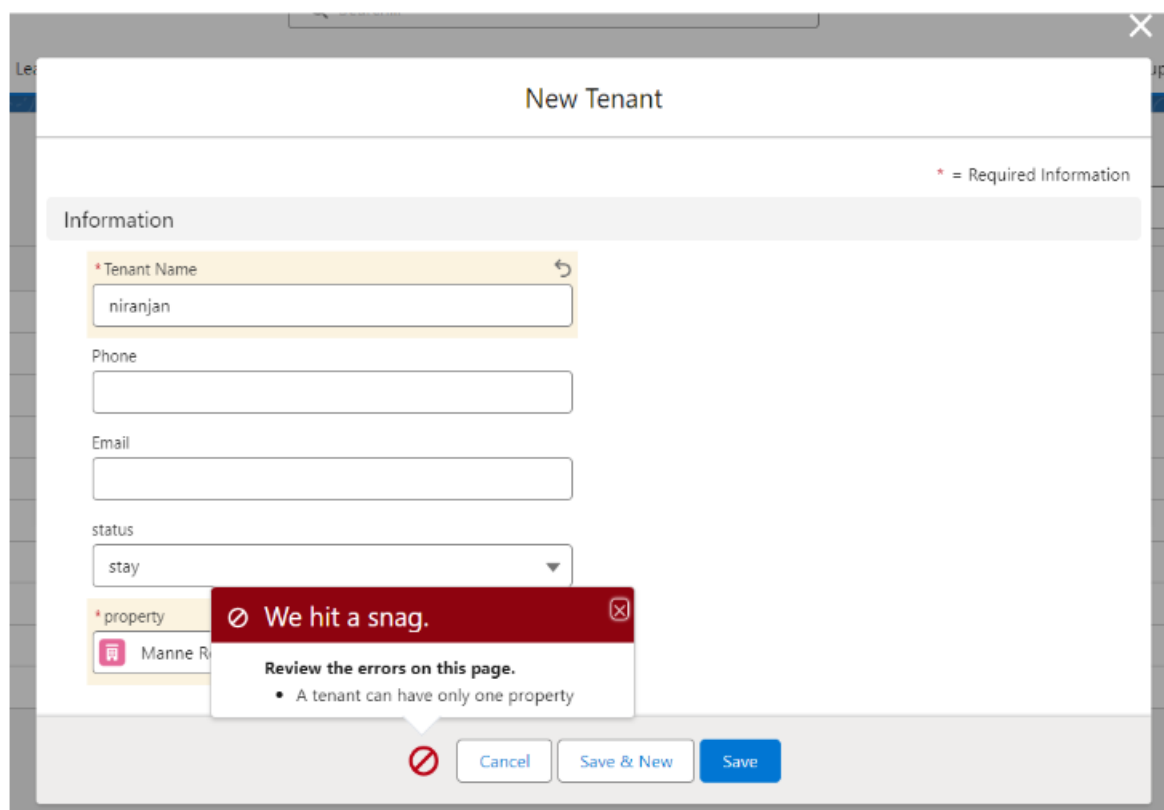
1 public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11         }
12
13         for (Tenant__c newTenant : newList) {
14
15             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
16
17                 newTenant.addError('A tenant can have only one property');
18
19             }
20
21         }
22     }
23 }
  
```

b) .Create an Apex Handler Class:

- The handler class encapsulates the logic for sending notifications and tasks based on the data received from the trigger.

c) Testing the Trigger:

- To ensure the trigger and handler class work as expected, a test class should be created that simulates the lease expiration scenarios and validates that notifications and tasks are triggered properly.



The screenshot shows a 'New Tenant' form with the following fields:

- Tenant Name** (required): niranjan
- Phone**
- Email**
- status**: stay
- property** (required): Manne R

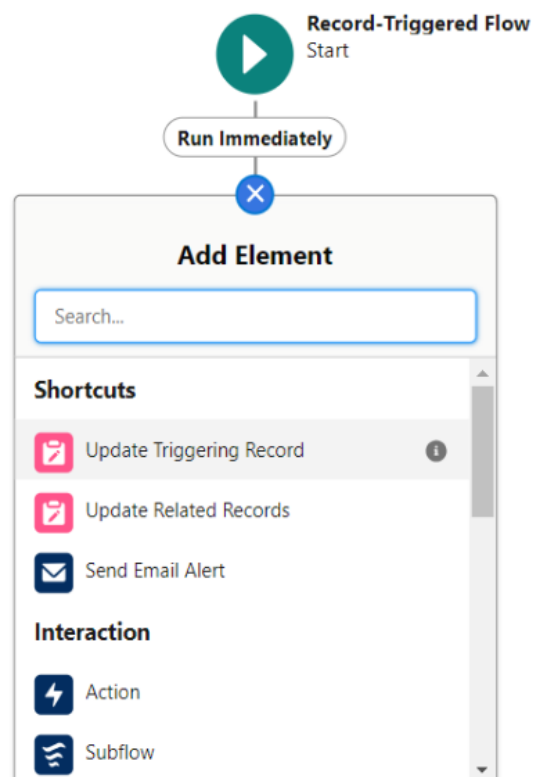
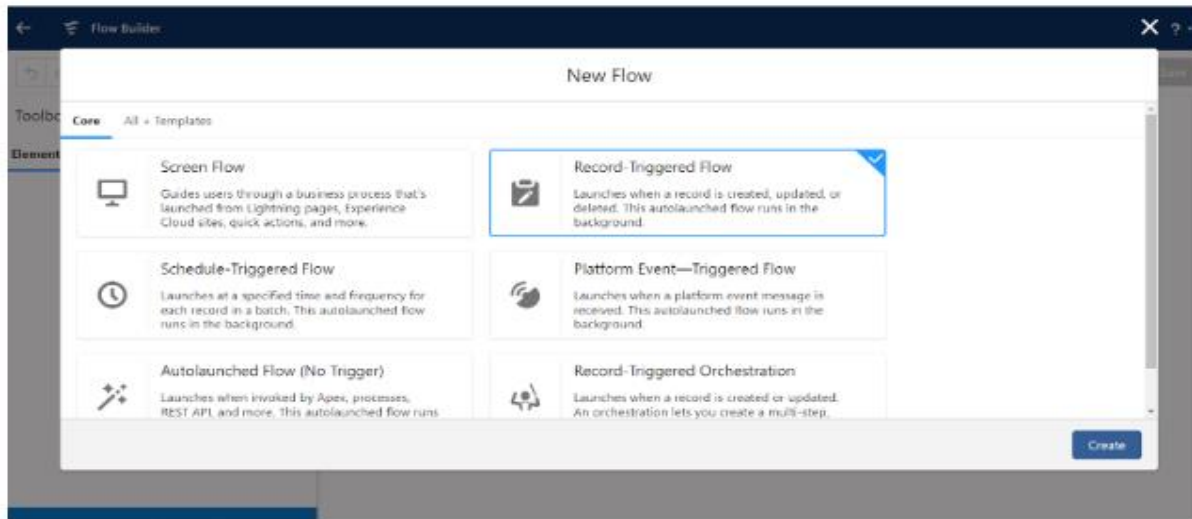
A red error message box is displayed over the form, stating: "We hit a snag. Review the errors on this page. A tenant can have only one property". The form also includes a 'Cancel' button, a 'Save & New' button, and a 'Save' button.

i) Flow:

Purpose:

To automatically send a confirmation email to tenants once their monthly payment has been marked as "paid" in the system. This flow triggers when the **Payment for Tenant** record is updated with the payment status.

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.
2. Select the record Triggered flow. Click on create.



Components:

1. Text Component:

- **Purpose:** Enter the **Tenant Name** dynamically, which will be used in the email body.
- **Usage:** {!\$ Record. Tenant __ r. Name}

2. Email Component:

- **Purpose:** Set the tenant's email address dynamically so the flow sends the confirmation email to the correct recipient.
- **Usage:** {!\$ Record .Tenant __r. Email __c}

3. Text Component:

- **Purpose:** Enter a predefined subject for the email, which is **Confirmation of Successful Monthly Payment**.
- **Usage:** "Confirmation of Successful Monthly Payment"

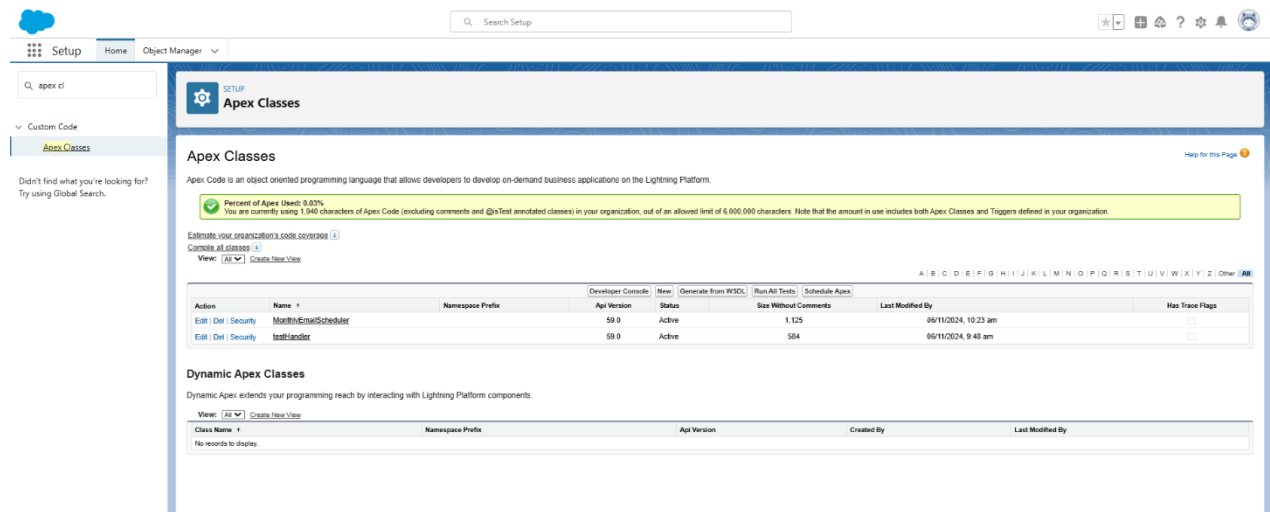
4. Text Template:

- **Purpose:** Create the body of the email with a personalized message to inform the tenant that their monthly payment has been successfully processed.

j) Scheduling an Apex Class in Salesforce

• Steps to Create and Schedule an Apex Class

- **Create the Apex Class:** The first step is to create an Apex class that contains the logic to be executed. Here's an example class that sends email reminders for unpaid tenant payments
- This class checks for any unpaid tenant payments and sends a reminder email.



Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 8.83%
You are currently using 1,946 characters of Apex Code (including comments and @xTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage [\[+\]](#)
Compile all classes [\[+\]](#)
View: [\[All\]](#) [\[New\]](#) [\[Create New View\]](#)

Action	Name	Namespace Prefix	Developer Console	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	MonthEmailScheduler			59.0	Active	1,125	06/11/2024, 10:23 am	<input type="checkbox"/>
Edit Del Security	testHandler			59.0	Active	504	06/11/2024, 9:48 am	<input type="checkbox"/>

Dynamic Apex Classes

Dynamic Apex extends your programming reach by interacting with Lightning Platform components.

View: [\[All\]](#) [\[Create New View\]](#)

Class Name	Namespace Prefix	Api Version	Created By	Last Modified By
No records to display.				

- **Create a Schedule Class:**Next, you'll create a **Schedulable** class that will trigger the execution of the above class at scheduled intervals.

6. Key Scenarios Addressed by Salesforce in Lease Management

- **Lease Tracking:** Salesforce centralizes lease details, automating reminders for renewals and expirations to prevent missed deadlines.
- **Tenant Communication:** Automates email notifications and reminders for rent payments, renewals, and maintenance, improving communication and tenant satisfaction.
- **Payment Tracking:** Automates rent invoicing and payment logging, reducing errors and ensuring smooth rent collection.
- **Maintenance Requests:** Enables tenants to submit requests via a portal, allowing property managers to track and resolve issues efficiently.

- **Reporting & Analytics:** Provides real-time insights on key metrics like occupancy, lease expirations, and rental income for better decision-making.
- **Compliance & Documentation:** Stores lease documents securely, ensuring compliance with legal standards and automating workflows to meet lease terms.
- **Workflow Automation:** Automates routine tasks such as payment reminders and lease status updates, reducing manual work and ensuring timely actions

7. Conclusion: Lease Management Project

The Salesforce-based lease management solution has streamlined lease tracking, tenant communication, payment processing, and maintenance management. By automating key processes and centralizing information, the project has reduced administrative work, improved efficiency, and ensured timely actions, such as lease renewals and rent reminders. Real-time reporting and compliance tracking have enhanced decision-making and reduced risks. Overall, this Salesforce solution has optimized property management, boosted tenant satisfaction, and increased operational effect

