LEASE MANAGEMENT

College Name: Adithya College of Arts and Science

College Code: BRUBH

TEAM ID: NM2025TMID26112

TEAM MEMBERS:

Team Leader Name: SAJITH. M

Email: sajithfiregaming@gmail.com

Team Member: SANJAY. S

Email: sanjayizana@gmail.com

Team Member: SHESHADRI. P

Email: sheshu1895.pssn@gmail.com

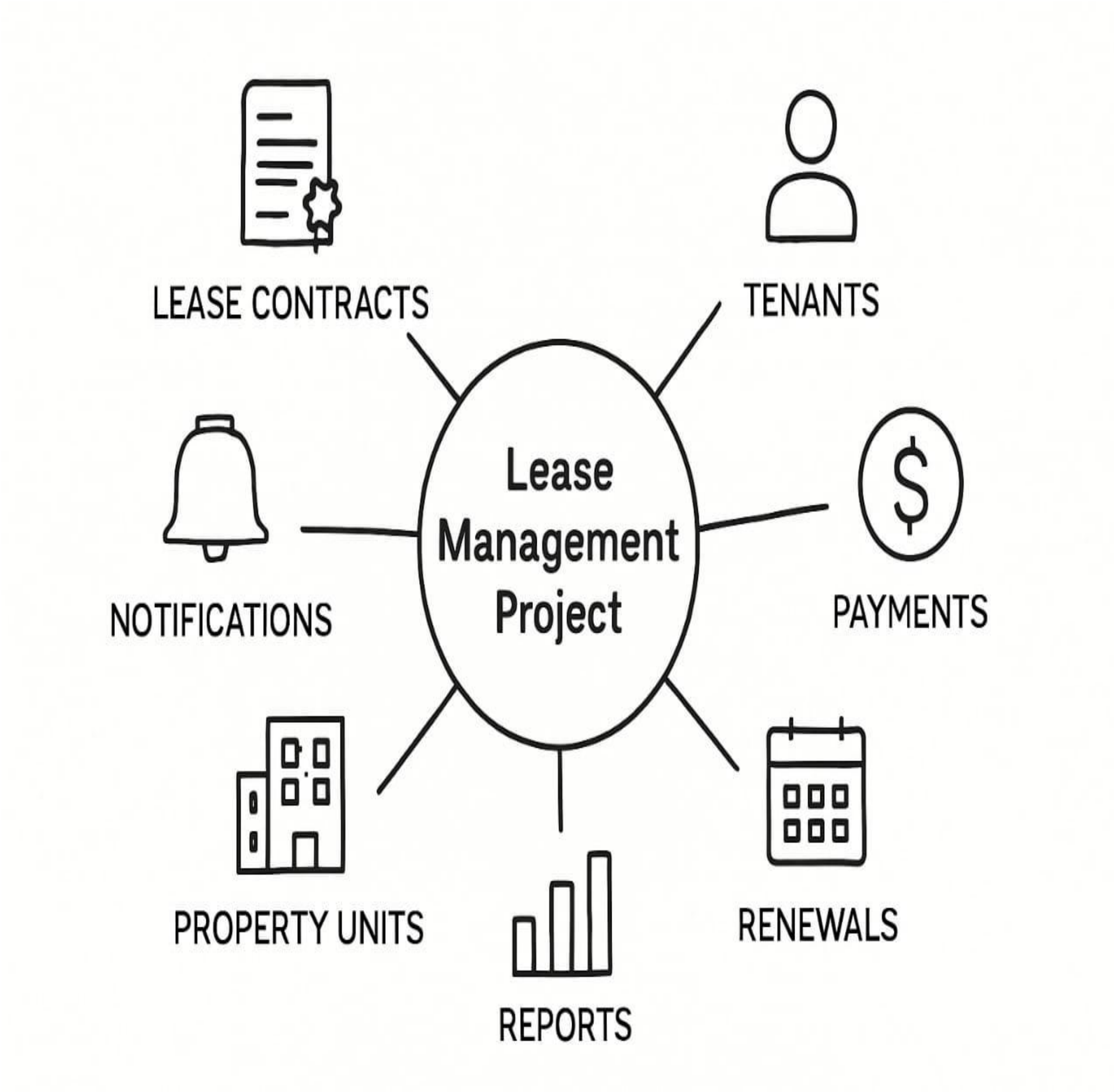
Team Member: SUMAN KARTHIKEYAN. S

Email: sumankarthikeyan131@gmail.com

1.INTRODUCTION

1.1 Project Overview

Salesforce lease management systems aim to simplify handling lease agreements, tenant information, lease renewals, payments, and vacancy tracking. These systems typically allow property owners or managements. Generate dashboards and repot showing metrics like total vacancies, occupied units, and revenue generated



1.2 Purpose

The lease management is to effectively organize, monitor, and administer leases for properties or assets to optimize their usage, ensure timely payments, and maintain compliance. It aims to automate lease related workflows, reduce manual error, and provide better visibility and control over lease contracts and tenant interactions.

DEVELOPMEMT PHASE

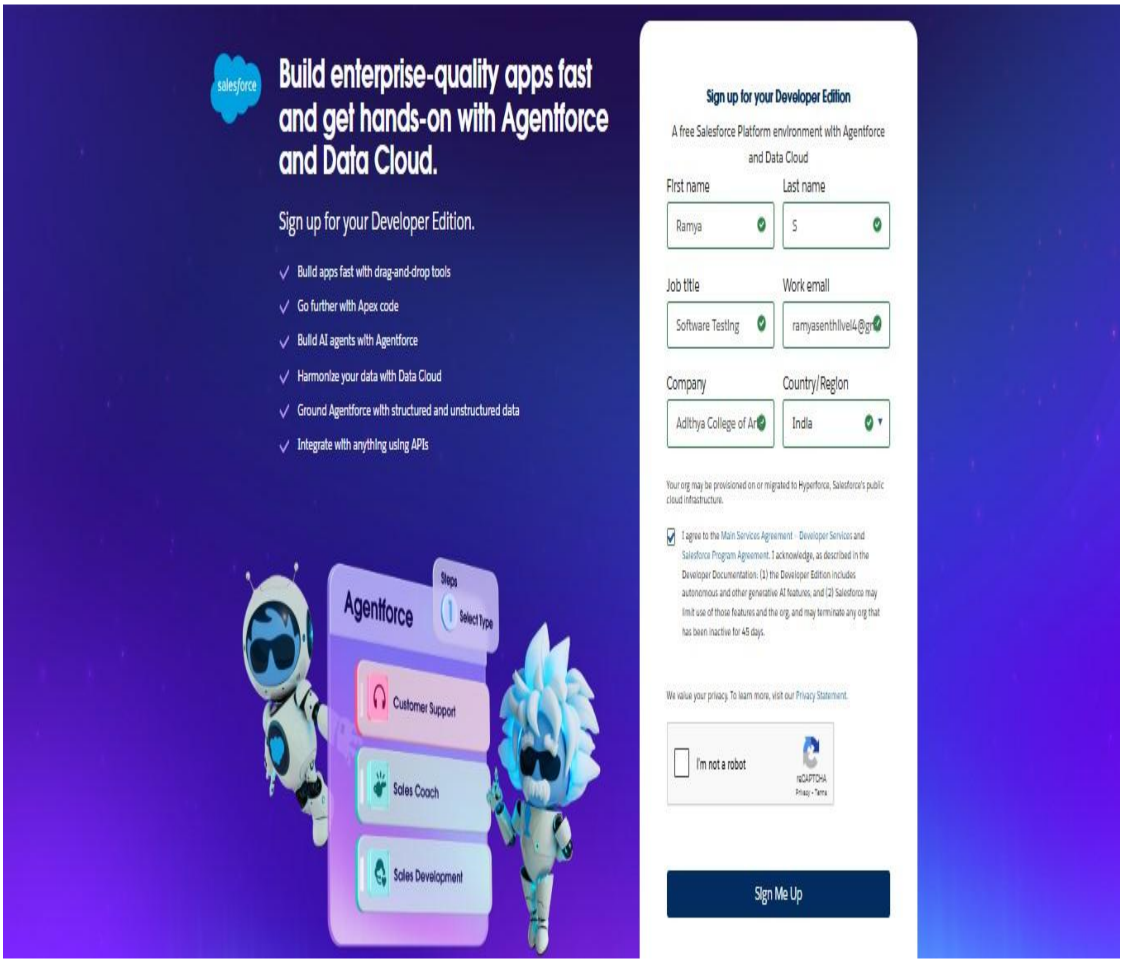
Creating Developer Account:

By using this URL - https://www.salesforce.com/form/developer-signup/?d=pb

1. First name & Last name
2. Email
3. Role: Developer
4. Company: College Name
5. County: India
6. Postal Code: pin code
7. Username: should be a combination of your name and company

This need not be an actual email id you can give anything in the format: username@organization.com

Click on sign me up after filling these.



salesforce

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Ramya ✓

Last name: S ✓

Job title: Software Testing ✓

Work email: ramyasenthilvel4@gr ✓

Company: Adithya College of Ar ✓

Country/Region: India ✓ ▾

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

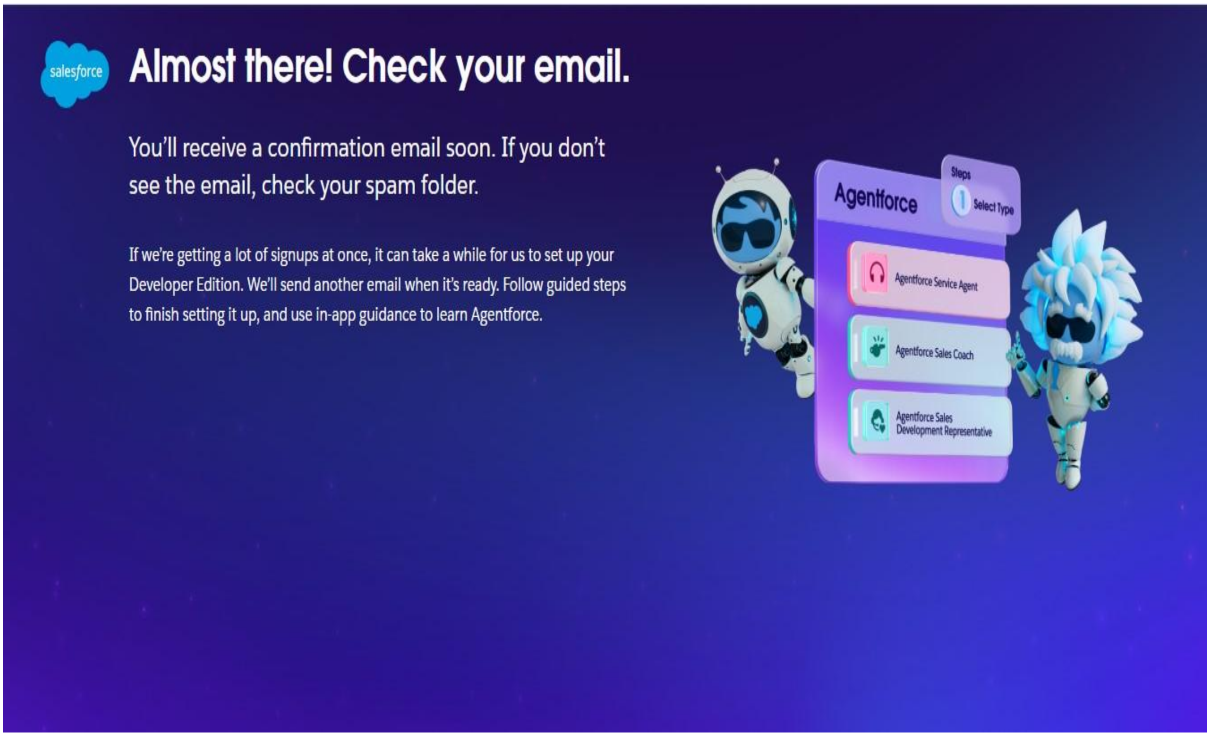
☑ I agree to the Main Services Agreement - Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features, and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our Privacy Statement.

□ I'm not a robot

reCAPTCHA
Privacy - Terms

Sign Me Up

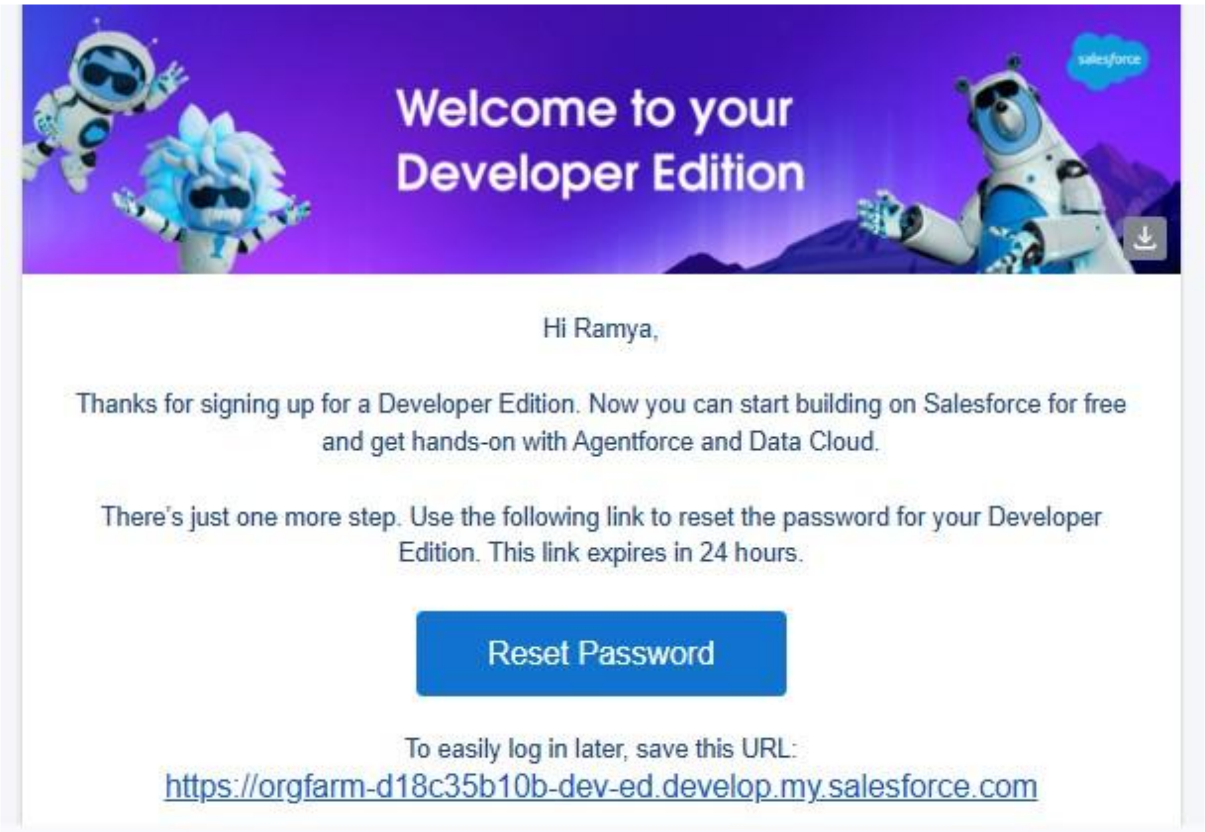


salesforce

Almost there! Check your email.

You'll receive a confirmation email soon. If you don't see the email, check your spam folder.

If we're getting a lot of signups at once, it can take a while for us to set up your Developer Edition. We'll send another email when it's ready. Follow guided steps to finish setting it up, and use in-app guidance to learn Agentforce.

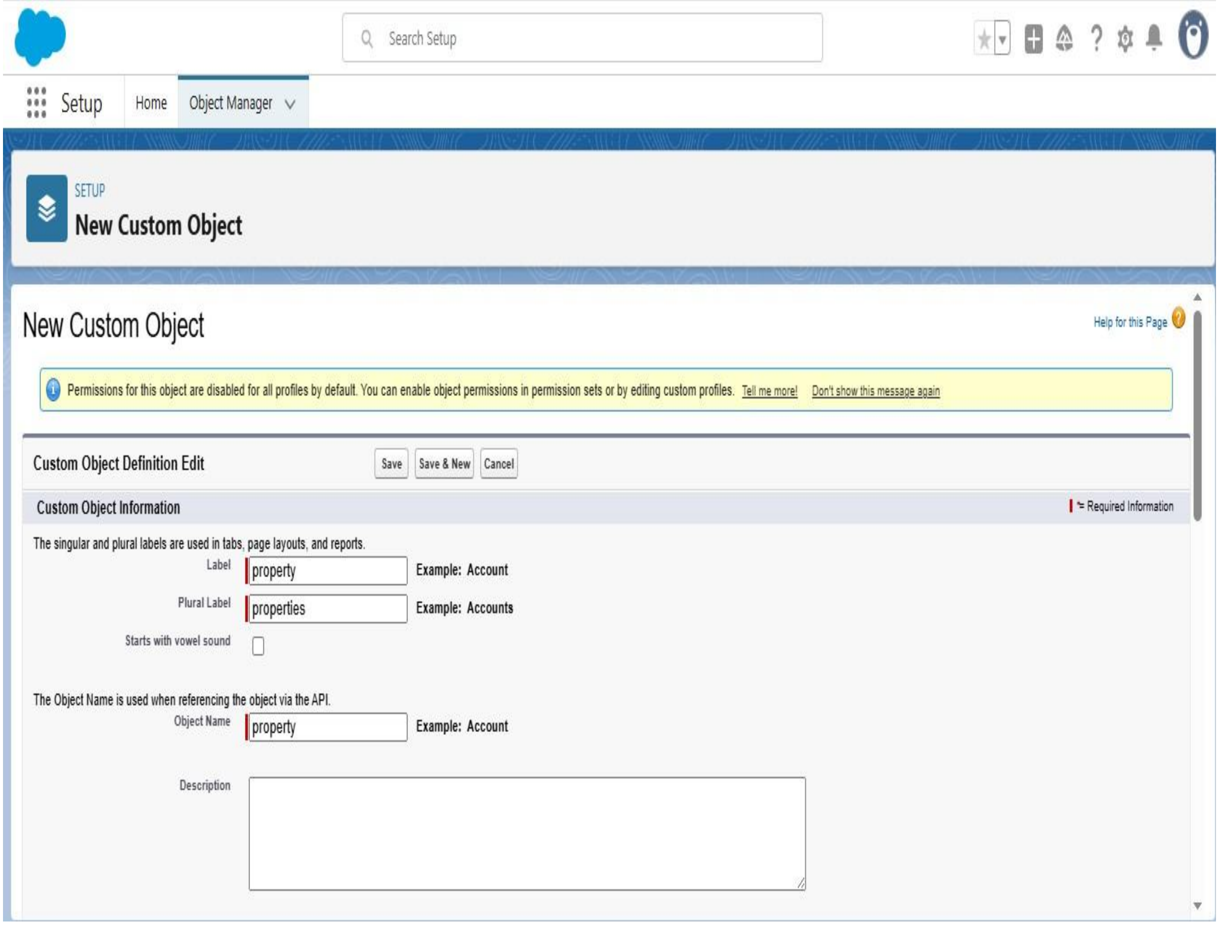


Hi Ramya,

Thanks for signing up for a Developer Edition. Now you can start building on Salesforce for free and get hands-on with Agentforce and Data Cloud.

There's just one more step. Use the following link to reset the password for your Developer Edition. This link expires in 24 hours.

To easily log in later, save this URL:
https://orgfarm-d18c35b10b-dev-ed.develop.my.salesforce.com



New Custom Object

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! Don't show this message again

Custom Object Definition Edit

Custom Object Information

| = Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label: property — Example: Account

Plural Label: properties — Example: Accounts

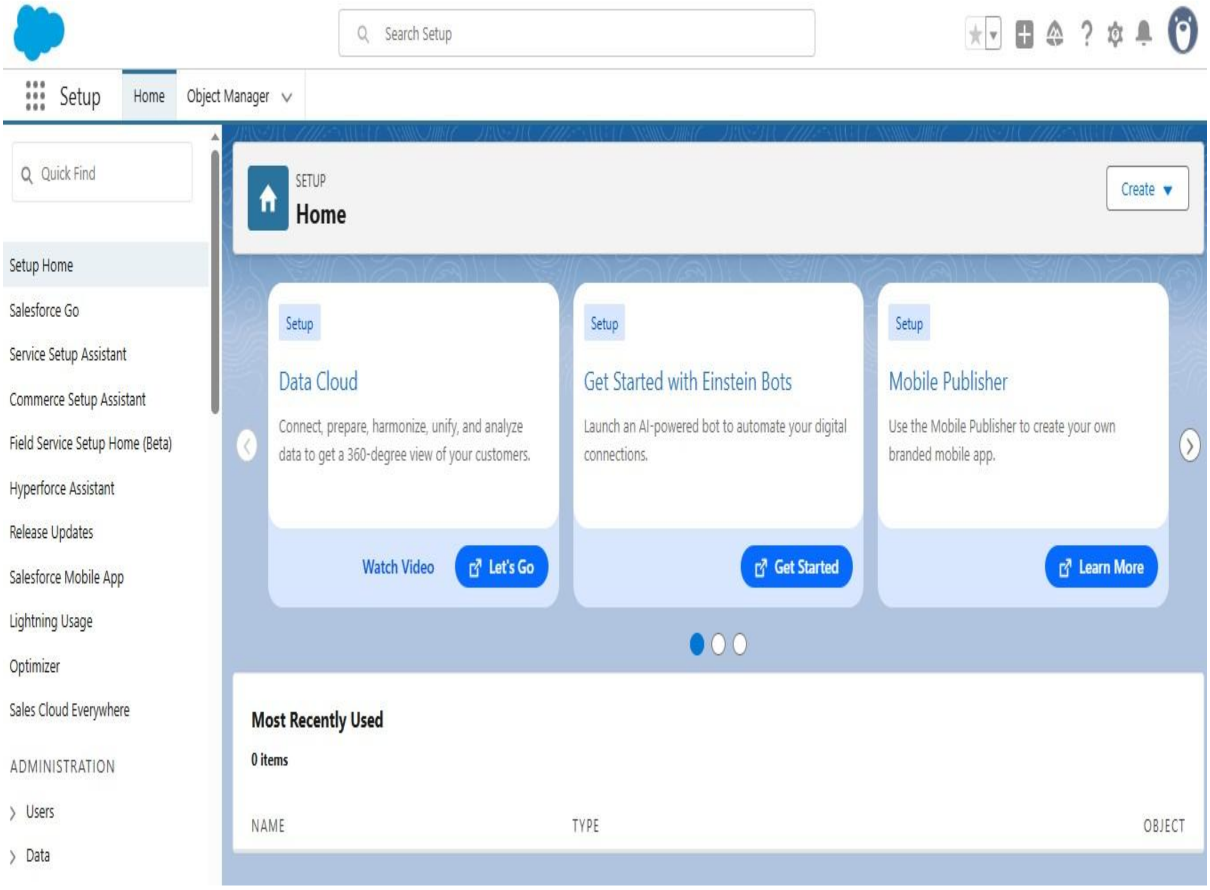
Starts with vowel sound: □

The Object Name is used when referencing the object via the API.

Object Name: property — Example: Account

Description:

- Create an object: Property, Tenant, Payment, Lease.



Connect, prepare, harmonize, unify, and analyze data to get a 360-degree view of your customers.

Launch an AI-powered bot to automate your digital connections.

Use the Mobile Publisher to create your own branded mobile app.

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

1. Enter the label name>> property

2. Plural label name>> property

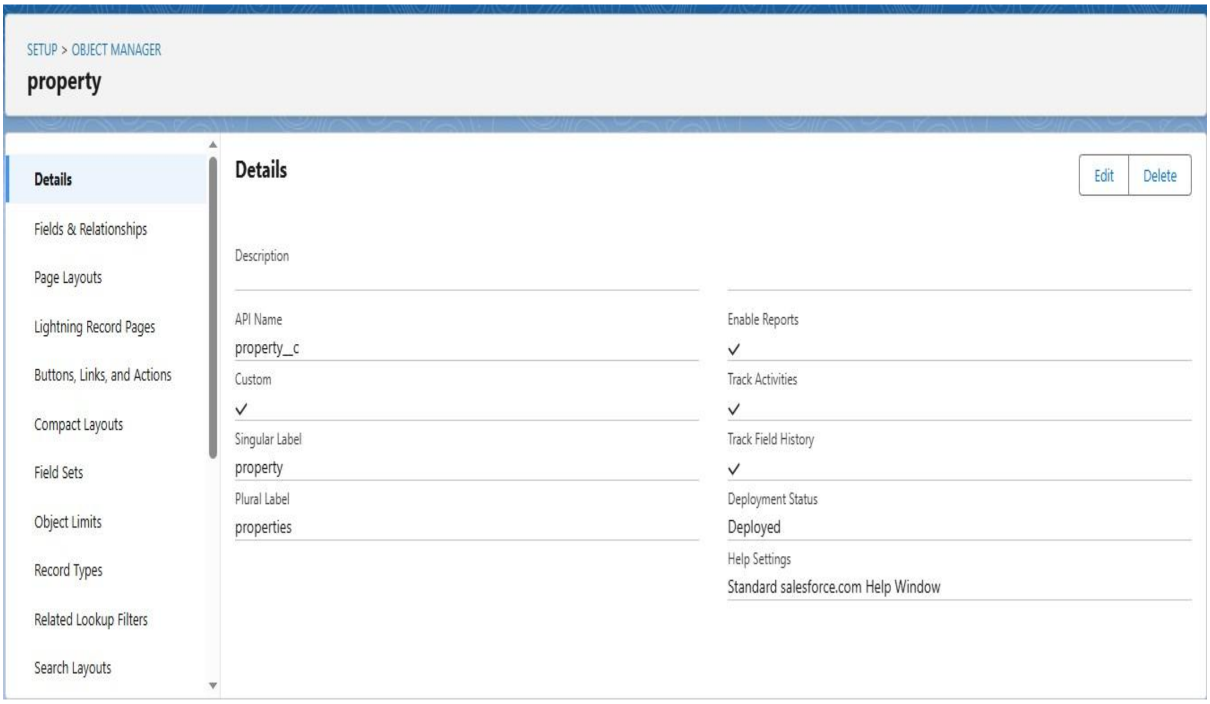
3. Enter Record Name Label and Format

- Record Name >>property Name

- Data Type >> Text

2. Click on Allow reports and Track Field History, Allow Activities

3. Allow search >> Save.



SETUP > OBJECT MANAGER

property

Details

Description

API Name: property_c

Custom: ✓

Singular Label: property

Plural Label: properties

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

To create an object:

1. From the setup page >> Click on Object Manager >>Click on Create >> Click on Custom Object.

1. Enter the label name>> Tenant

2. Plural label name>> Tenants

3. Enter Record Name Label and Format

- Record Name >> Tenant Name

- Data Type >> Text

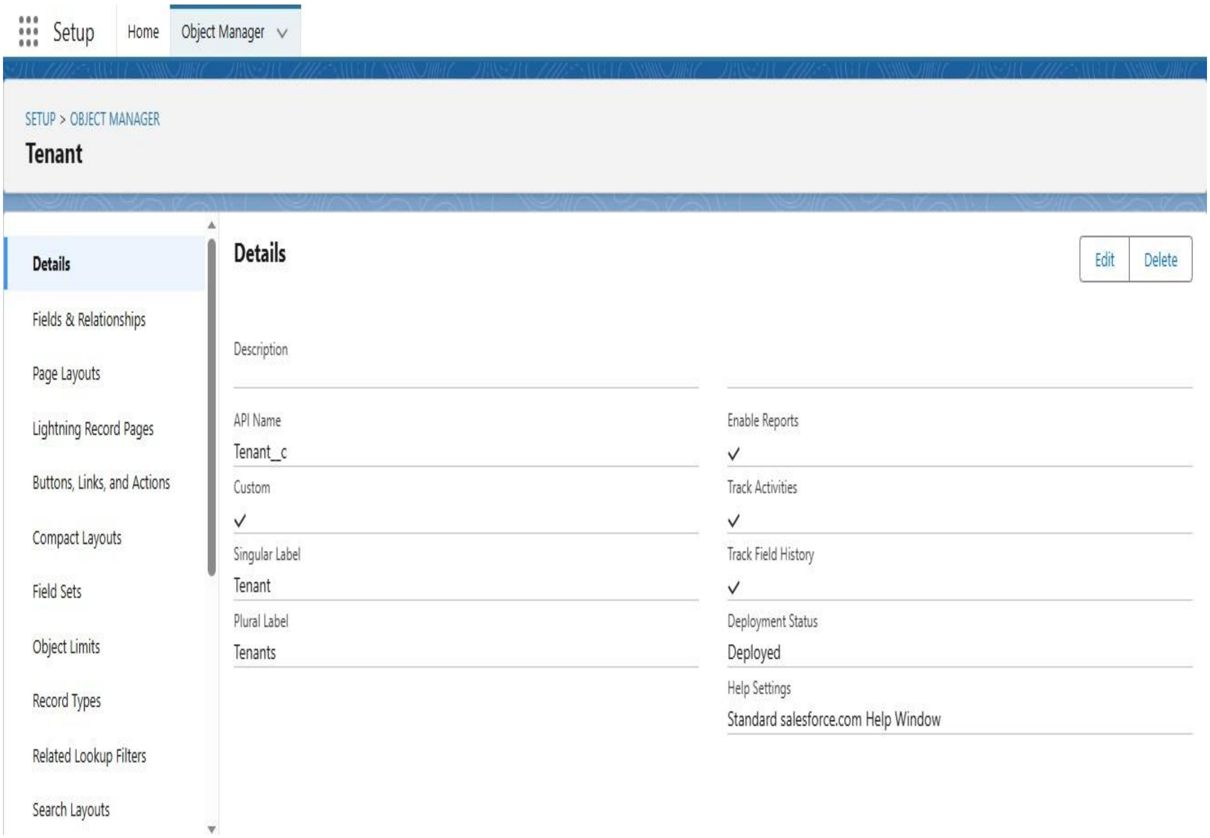
2. Click on Allow reports and Track Field History, Allow Activities

3. Allow search >> Save.

- Data Type >> Text

4. Click on Allow reports and Track Field History, Allow Activities

5. Allow search >> Save.



Description

API Name
Tenant_c

Custom
✓

Singular Label
Tenant

Plural Label
Tenants

Enable Reports
✓

Track Activities
✓

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

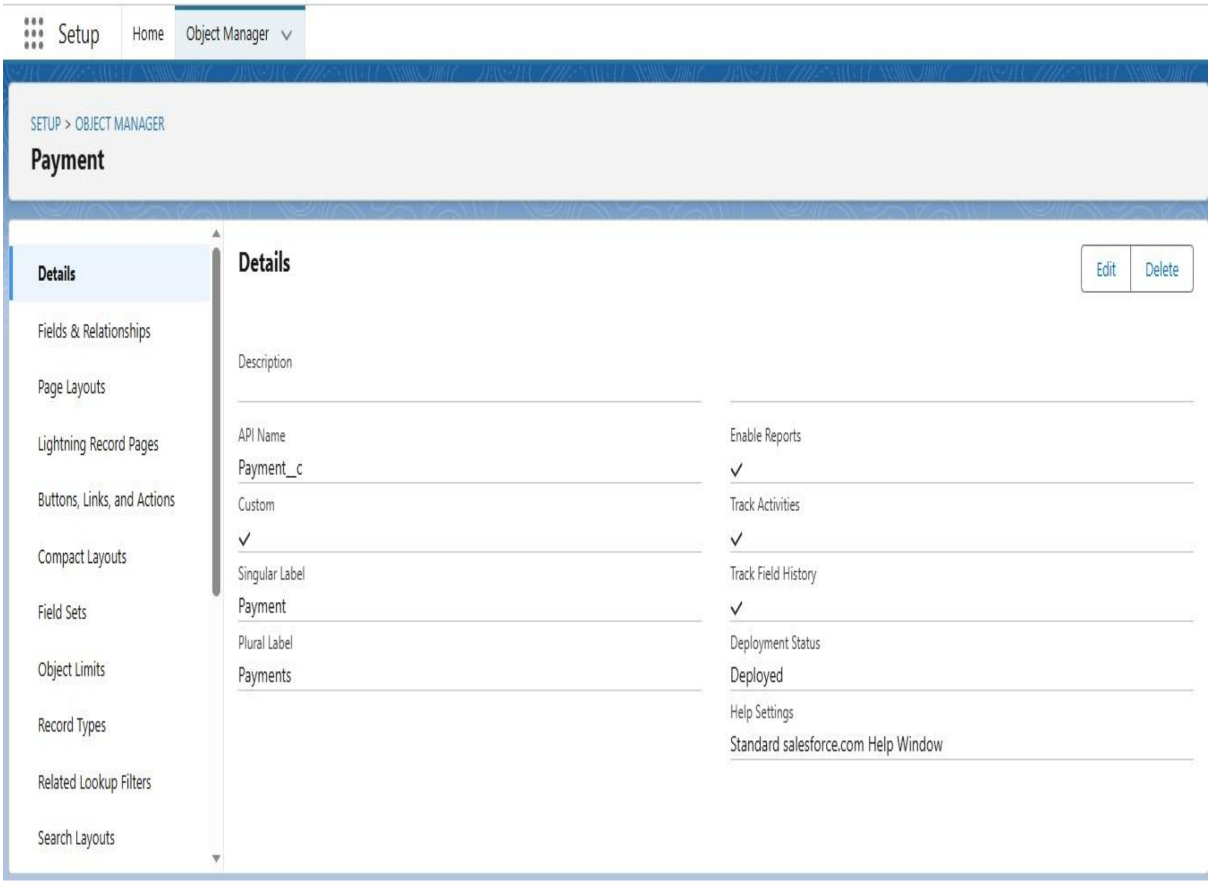
To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name>> Payment
3. Plural label name>> Payment
4. Enter Record Name Label and Format

- Record Name >> Payment Name
- Data Type >> Text

5. Click on Allow reports and Track Field History, Allow Activities

Allow search >> Save.



SETUP > OBJECT MANAGER

Payment

Details

Description

API Name
Payment_c

Custom
✓

Singular Label
Payment

Plural Label
Payments

Enable Reports
✓

Track Activities
✓

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

1. Enter the label name>> lease

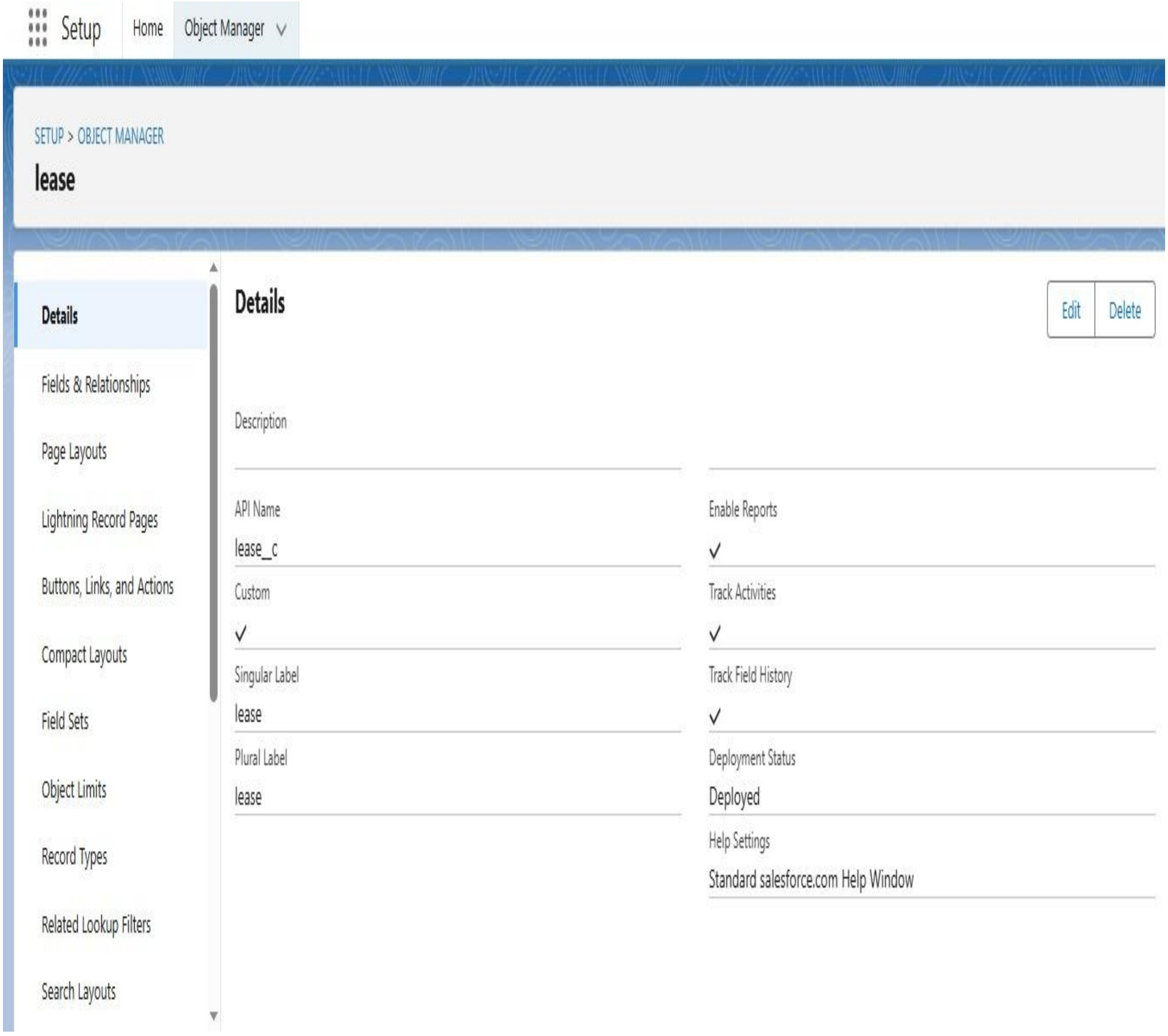
2. Plural label name>> lease

3. Enter Record Name Label and Format

- Record Name >> lease Name
- Data Type >> Text

2. Click on Allow reports and Track Field History, Allow Activities

3. Allow search >> Save.



lease

Details

Description

API Name: lease_c

Custom: ✓

Singular Label: lease

Plural Label: lease

Enable Reports: ✓

Track Activities: ✓

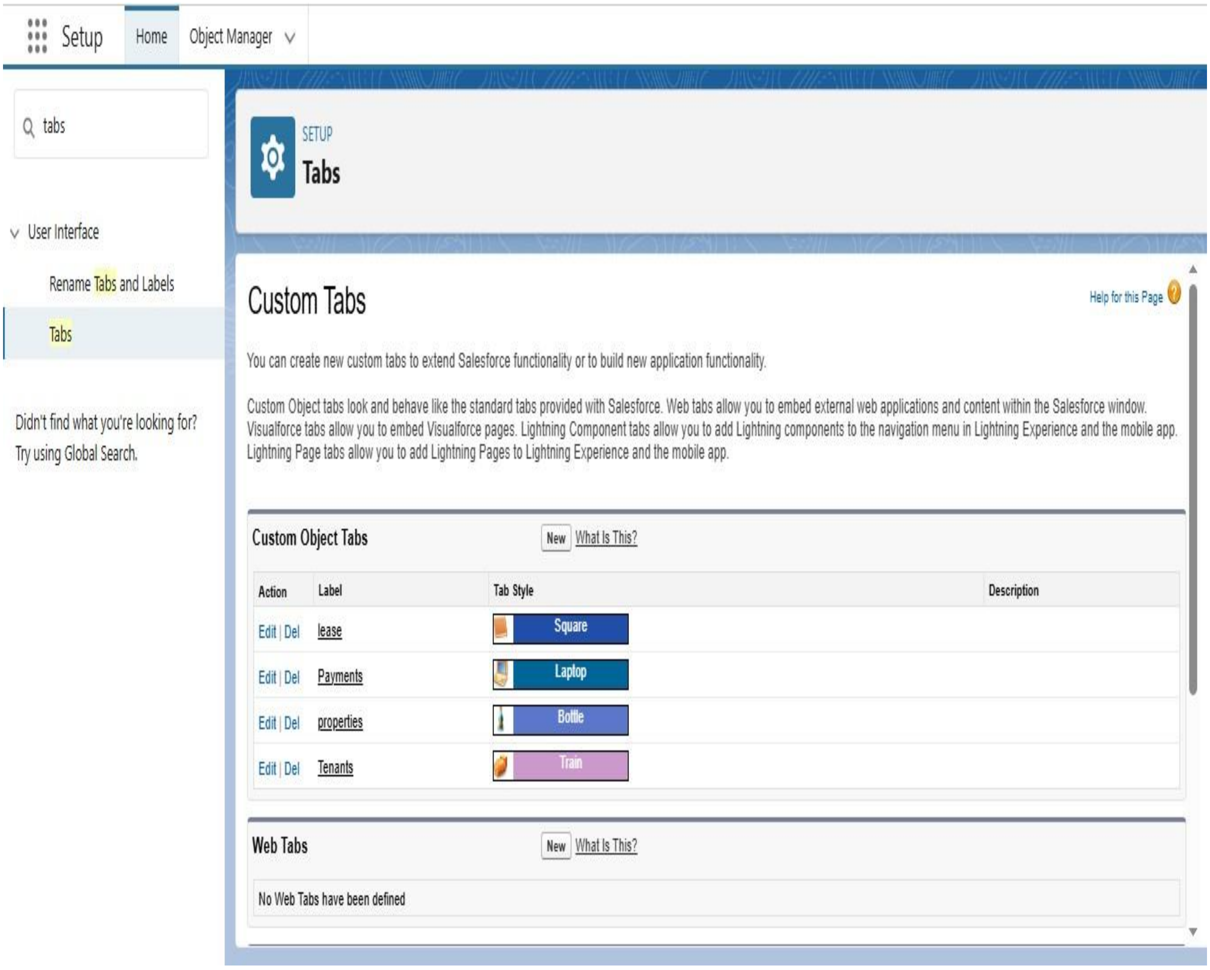
Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

- Go to setup page >>type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)

1. Select Object (property) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab.
2. Make sure that the Append tab to users' existing personal customizations is checked.
3. Click save



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

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

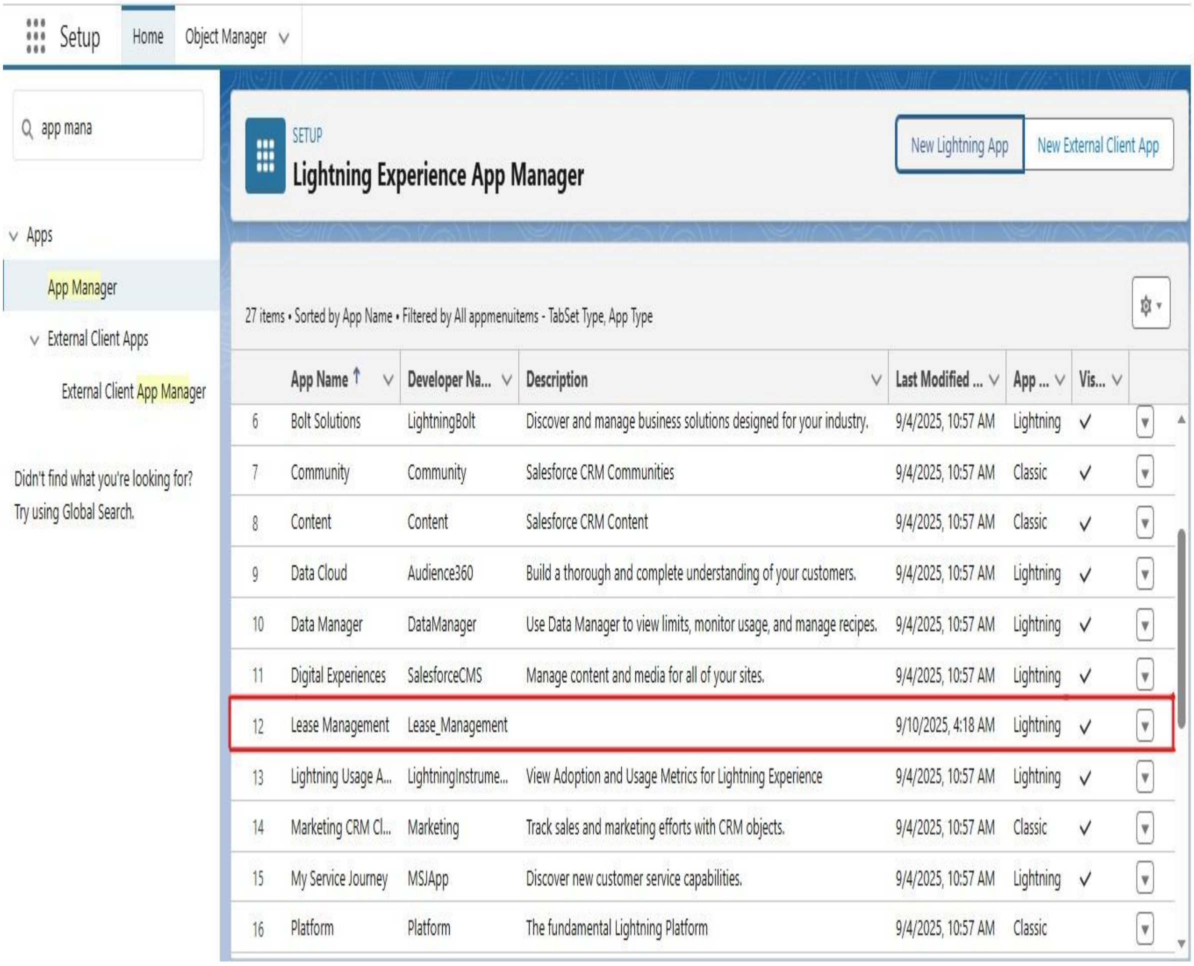
Action	Label	Tab Style	Description
Edit | Del	lease	Square	
Edit | Del	Payments	Laptop	
Edit | Del	properties	Bottle	
Edit | Del	Tenants	Train	

Web Tabs

No Web Tabs have been defined

- Create a Lightning App.

- An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.
- Lightning apps let you brand your apps with a custom colour and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.



Lightning Experience App Manager

27 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type, App Type

	App Name	Developer Na...	Description	Last Modified ...	App ...	Vis...
6	Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	9/4/2025, 10:57 AM	Lightning	✓
7	Community	Community	Salesforce CRM Communities	9/4/2025, 10:57 AM	Classic	✓
8	Content	Content	Salesforce CRM Content	9/4/2025, 10:57 AM	Classic	✓
9	Data Cloud	Audience360	Build a thorough and complete understanding of your customers.	9/4/2025, 10:57 AM	Lightning	✓
10	Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	9/4/2025, 10:57 AM	Lightning	✓
11	Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	9/4/2025, 10:57 AM	Lightning	✓
12	Lease Management	Lease_Management		9/10/2025, 4:18 AM	Lightning	✓
13	Lightning Usage A...	LightningInstrume...	View Adoption and Usage Metrics for Lightning Experience	9/4/2025, 10:57 AM	Lightning	✓
14	Marketing CRM Cl...	Marketing	Track sales and marketing efforts with CRM objects.	9/4/2025, 10:57 AM	Classic	✓
15	My Service Journey	MSJApp	Discover new customer service capabilities.	9/4/2025, 10:57 AM	Lightning	✓
16	Platform	Platform	The fundamental Lightning Platform	9/4/2025, 10:57 AM	Classic	

- Configured Fields and Relationship.

Fill the Above as following:

- Field Label: Name
- Field Name: gets auto generated
- Length: 25
- Required: check box
- Click on Next >> Next >> Save and new.

2. To create another fields in an object:

1. Go to setup >> click on Object Manager >>type object name(property) in search bar >>click on the object.
2. Now click on “Fields & Relationships” >>New
3. Select Data type as a “Long Text” and Click on Next

4. Fill the Above as following:

- Field Label: Address
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

3. To create another fields in an object:

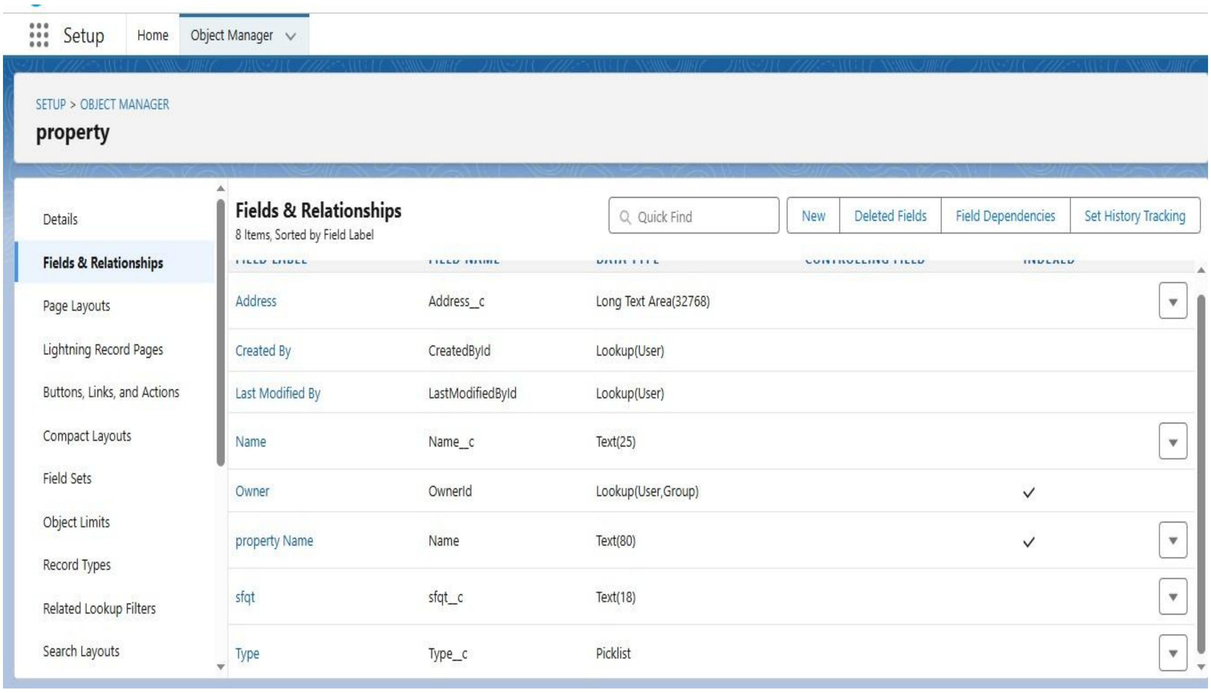
5. Go to setup >> click on Object Manager >>type object name(property) in search bar >> click on the object.
6. Now click on “Fields & Relationships” >> New
7. Select Data type as a “picklist” and Click on Next
8. Fill the Above as following:

- Field Label: Type
- Field Name: gets auto generated
- Enter values, with each value separated by a new line
- Enter these values
  1BHK
  2BHK
  3BHK
- Click on Next >> Next >> Save and new.

To create another fields in an object:

9. Go to setup >> click on Object Manager >> type object name(property) in search bar >> click on the object.
10. Now click on “Fields & Relationships” >> New
11. Select Data type as a “ Text” and Click on Next
12. Fill the Above as following:

- Field Label: sfqt
- Field Name: gets auto generated
- Length: 18
- Click on Next >> Next >> Save.



SETUP > OBJECT MANAGER

property

Fields & Relationships

8 Items, Sorted by Field Label

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)		✓
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

- Go to setup >> click on Object Manager >> type object name(Tenant) in search bar >> click on the object.

2. Now click on “Fields & Relationships” >> New

3. Select Data type as a “Email” and Click on Next

4. Fill the Above as following:

- Field Label: Email
- Field Name: gets auto generated
- Click on required check box
- Click on Next >> Next >> Save and new.

To create another fields in an object:

- Go to setup >> click on Object Manager >> type object name (Tenant) in search bar >> click on the object.

1. Now click on “Fields & Relationships” >> New
2. Select Data type as a “phone” and Click on Next
3. Fill the Above as following:

- Field Label: Phone
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

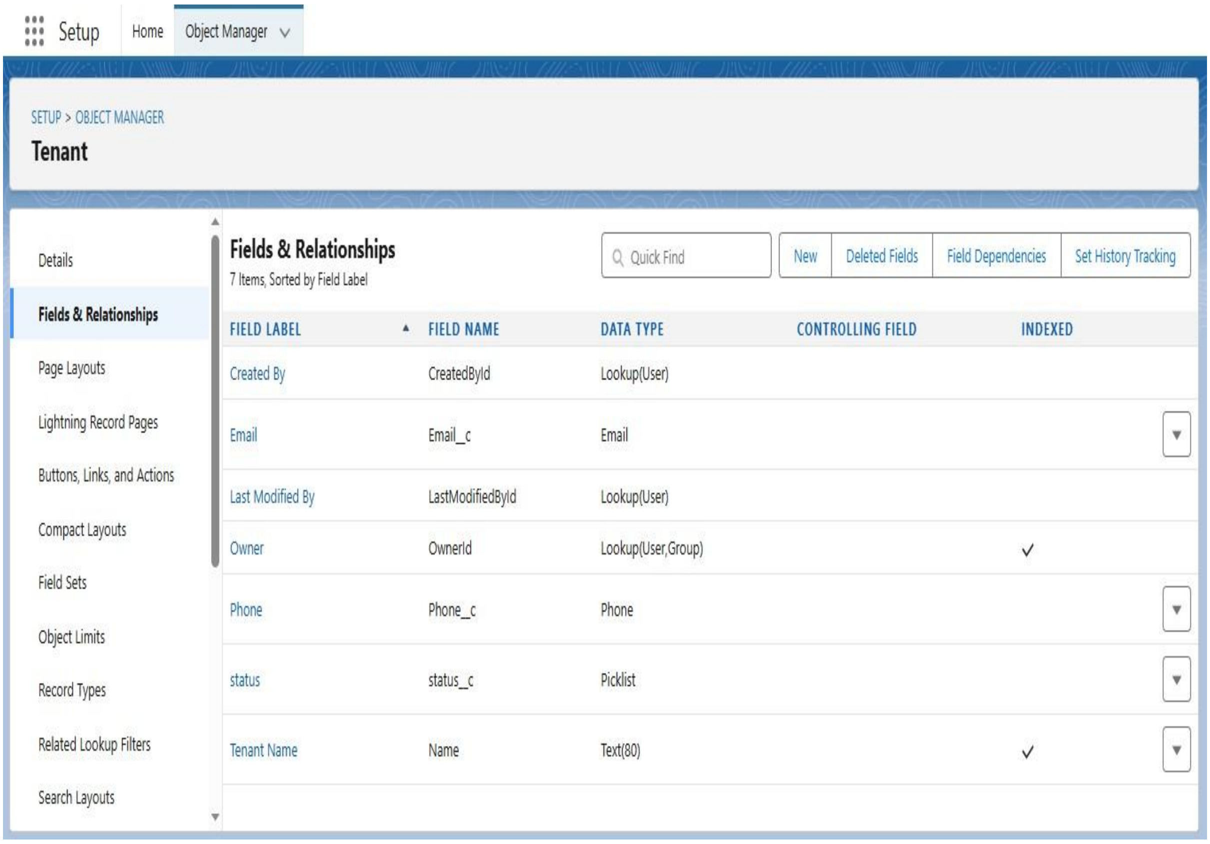
To create another fields in an object:

5. Go to setup >> click on Object Manager >> type object name (Tenant) in search bar >> click on the object.
6. Now click on “Fields & Relationships” >>New
7. Select Data type as a “picklist” and Click on Next
8. Fill the Above as following:

- Field Label: status
- Field Name: gets auto generated
- Enter values, with each value separated by a new line
- Enter these values

  Stay

  Leaving
- Click on Next >> Next >> Save



Tenant

Fields & Relationships

7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
status	status__c	Picklist		
Tenant Name	Name	Text(80)		✓

- Go to setup >> click on Object Manager >> type object name(Lease) in search bar >> click on the object.

2. Now click on “Fields & Relationships” >> New

3. Select Data type as a “Date” and Click on Next

4. Fill the Above as following:

- Field Label: start date
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

To create another fields in an object:

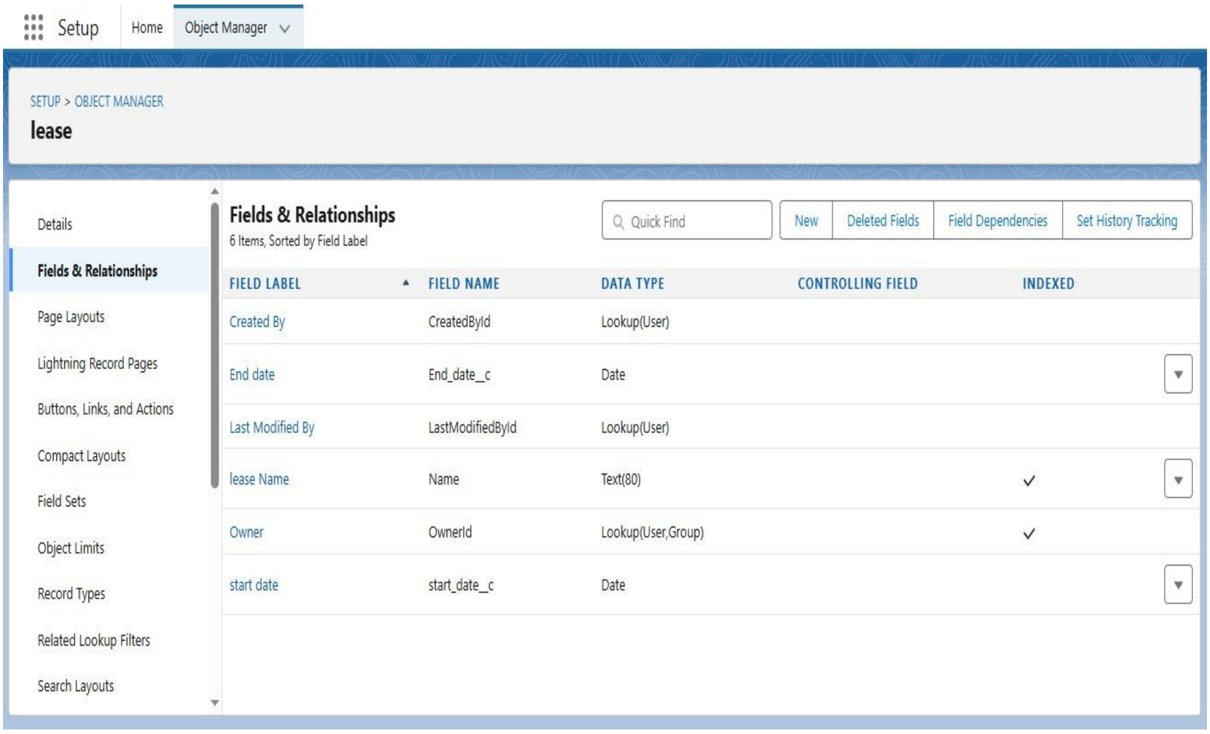
1.Go to setup >> click on Object Manager >> type object name(Lease) in search bar >> click on the object.

2. Now click on “Fields & Relationships” >> New

3. Select Data type as a “Date” and Click on Next

4.Fill the Above as following:

- Field Label: End date
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.



Fields & Relationships

6 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End date	End_date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓
start date	start_date__c	Date		

1.Go to setup >> click on Object Manager >> type object name (Payment for tenant) in search bar >> click on the object.

2. Now click on “Fields & Relationships” >> New

3. Select Data type as a “Date” and Click on Next

4. Fill the Above as following:

- Field Label: Payment date
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

To create another fields in an object:

1.Go to setup >> click on Object Manager >> type object name (Payment for tenant) in search bar >> click on the object.

2. Now click on “Fields & Relationships” >> New

3. Select Data type as a “Number” and Click on Next

4. Fill the Above as following:

- Field Label: Amount
- Length: 18
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

To create another fields in an object:

1.Go to setup >> click on Object Manager >> type object name (Payment for tenant) in search bar >> click on the object.

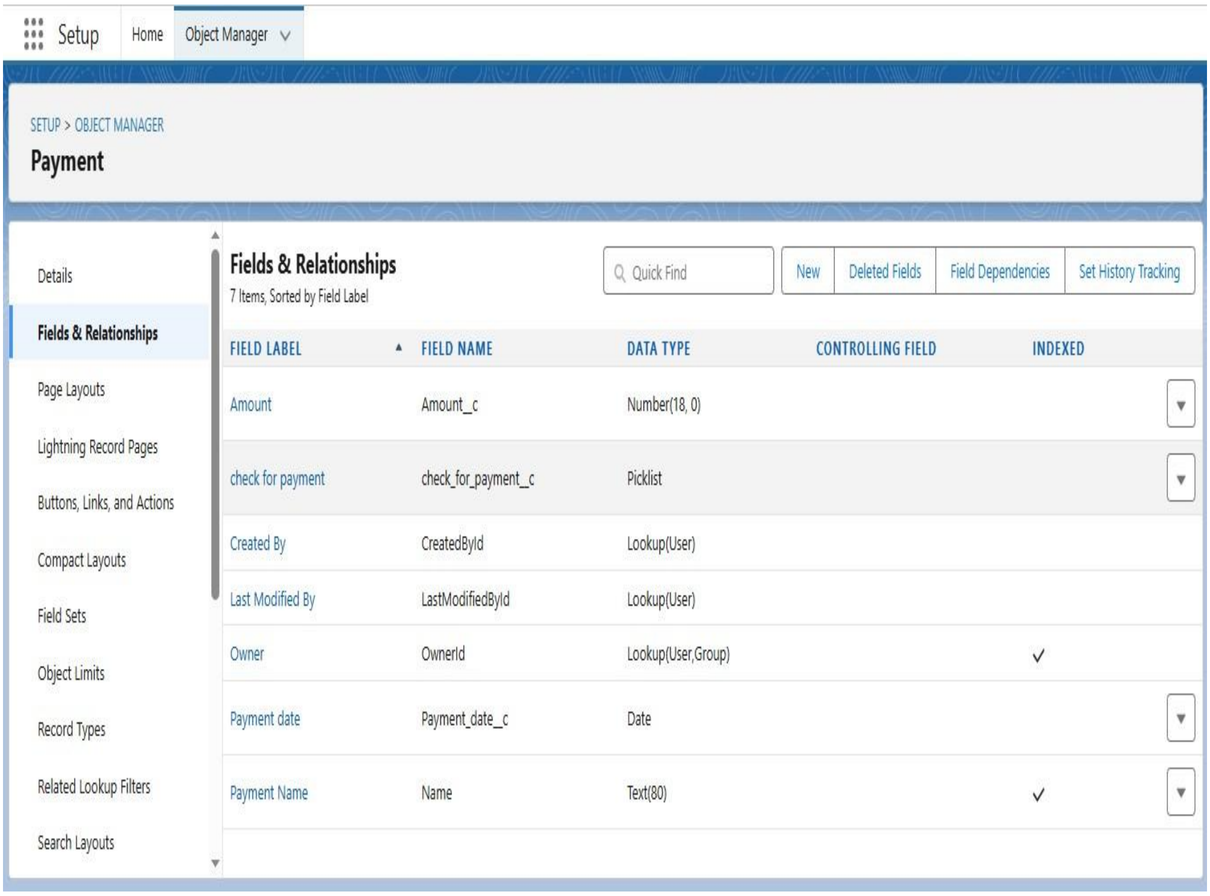
2. Now click on “Fields & Relationships” >> New

3. Select Data type as a “picklist” and Click on Next

4. Fill the Above as following:

- Field Label: check for payment
- Field Name: gets auto generated
- Enter values, with each value separated by a new line

- Enter these values
  Paid
  Not paid
- Click on Next >> Next >> Save and new.



SETUP > OBJECT MANAGER

Payment

Fields & Relationships

7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Number(18, 0)		
check for payment	check_for_payment_c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment date	Payment_date_c	Date		
Payment Name	Name	Text(80)		✓

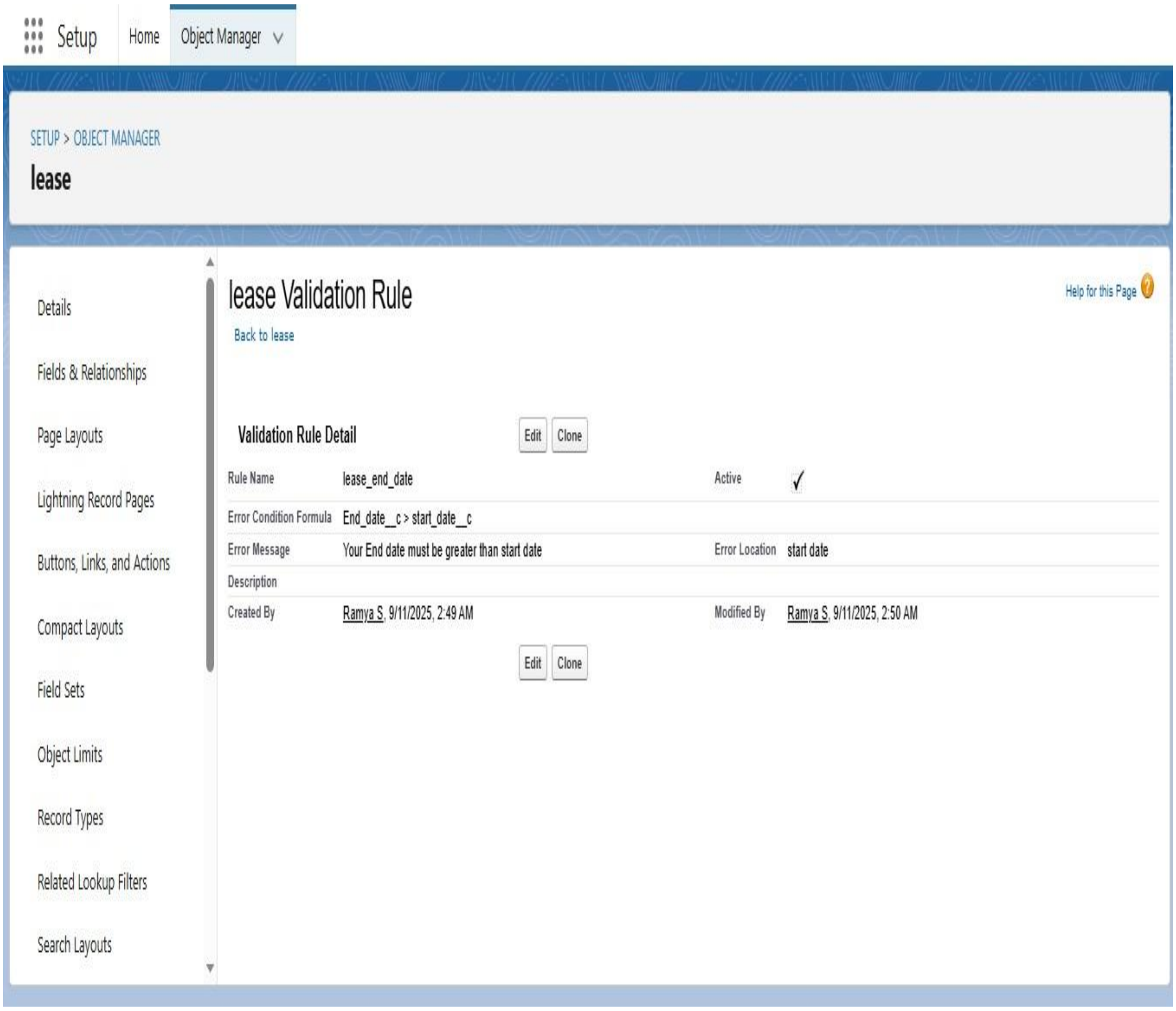
- Create a validation Rule to an lease object.

1. Go to the setup page >> click on object manager >> From drop down click edit for Lease object.
2. Click on the validation rule >> click New.

3. Enter the Rule name as “ lease_end_date”.

4. Insert the Error Condition Formula as:
   End_date__c > start_date__c

5. Enter the Error Message as “Your End date must be greater than start date”, select the Error location as Field and select the field as “start date”, and click Save



lease Validation Rule

Back to lease

Validation Rule Detail

Rule Name	lease_end_date	Active	✓
Error Condition Formula	End_date__c > start_date__c		
Error Message	Your End date must be greater than start date	Error Location	start date
Description			
Created By	Ramya S, 9/11/2025, 2:49 AM	Modified By	Ramya S, 9/11/2025, 2:50 AM

To create Email Template:

1. Go to setup in quick find box enter email template >> click on classic Email Template.
2. Click on >> New Email Template===>Choose text

    Folder: Unfiled public Classic Email templates

Click on available for use

3. Email Template Name is “tenant leaving”

4. Template Unique Name: Auto populated

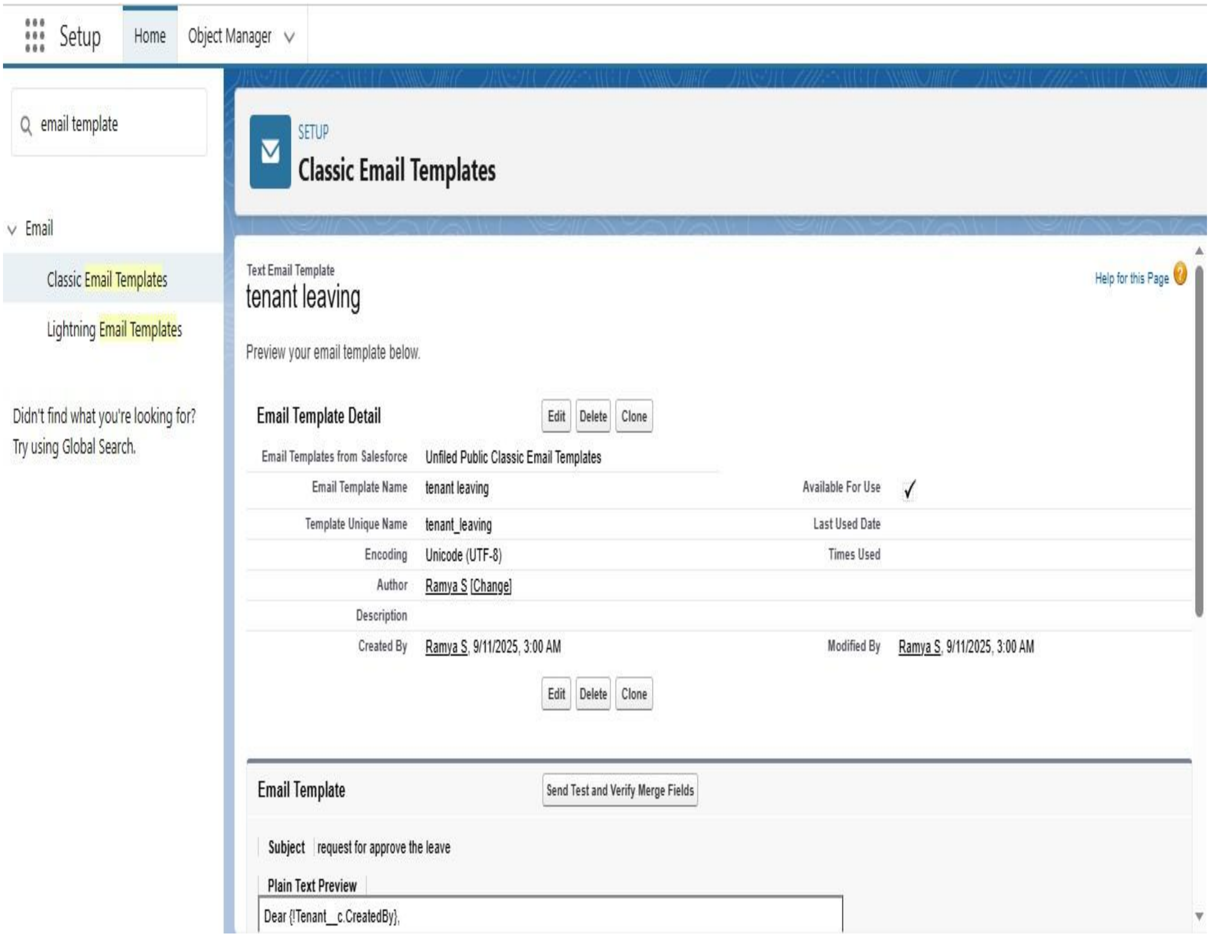
5. Subject: ” request for approve the leave”

6. Email body:

Dear {!Tenant__c.CreatedBy},

Please approve my leave

7. Save



Text Email Template

tenant leaving

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates		
Email Template Name	tenant leaving	Available For Use	✓
Template Unique Name	tenant_leaving	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Ramya S [Change]		
Description			
Created By	Ramya S, 9/11/2025, 3:00 AM	Modified By	Ramya S, 9/11/2025, 3:00 AM

Email Template

Subject request for approve the leave

Plain Text Preview

Dear {!Tenant__c.CreatedBy},

- Go to setup in quick find box enter email template >> click on classic Email Template.

2. Click on >> New Email Template===>Choose text

   Folder: Unfiled public Classic Email templates

Click on available for use

3. Email Template Name is “Leave approved”

4. Template Unique Name: Auto populated

5. Subject: “ Leave approved”

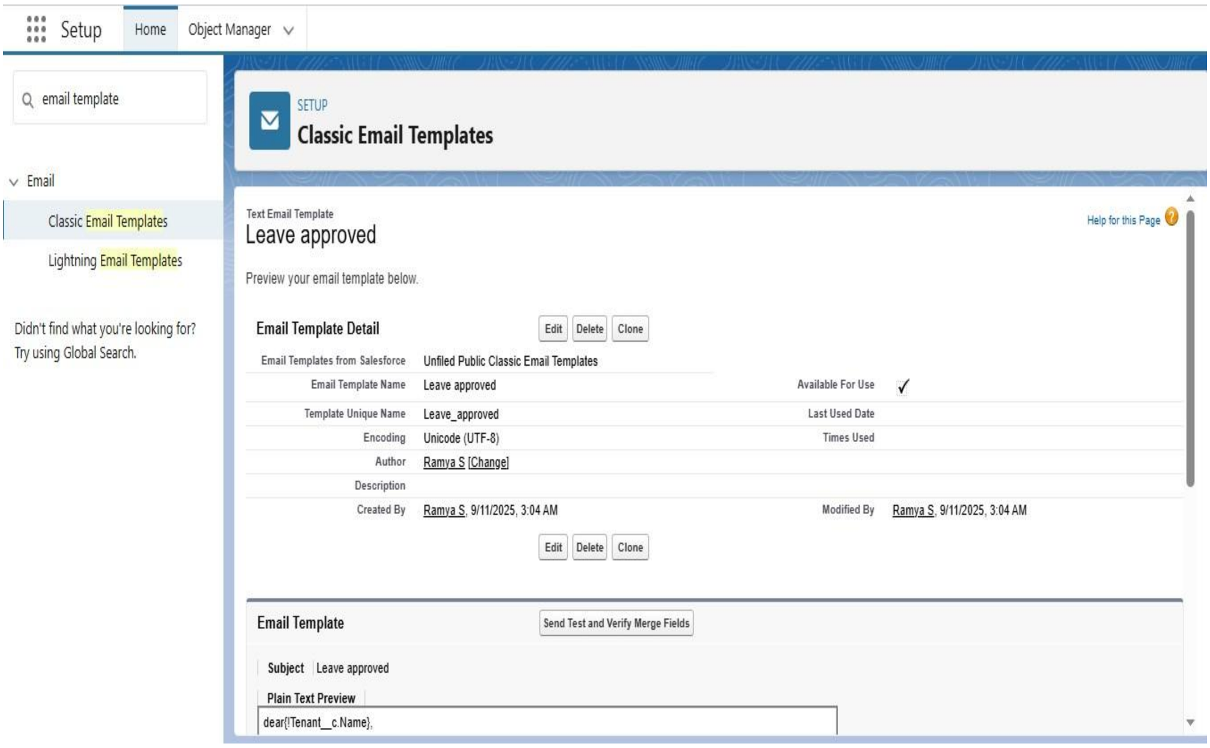
6. Email body:

dear{!Tenant__c.Name},

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

your leave is approved. You can leave now

8. Save



Text Email Template

Leave approved

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates		
Email Template Name	Leave approved	Available For Use	✓
Template Unique Name	Leave_approved	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Ramya S [Change]		
Description			
Created By	Ramya S, 9/11/2025, 3:04 AM	Modified By	Ramya S, 9/11/2025, 3:04 AM

Email Template

Subject Leave approved

Plain Text Preview

dear{!Tenant__c.Name},

Go to setup in quick find box enter email template >> click on classic Email Template.

2. Click on >>New Email Template===>Choose text

Folder: Unfiled public Classic Email templates

Click on available for use

3. Email Template Name is “Leave rejected”

4. Template Unique Name: Auto populated

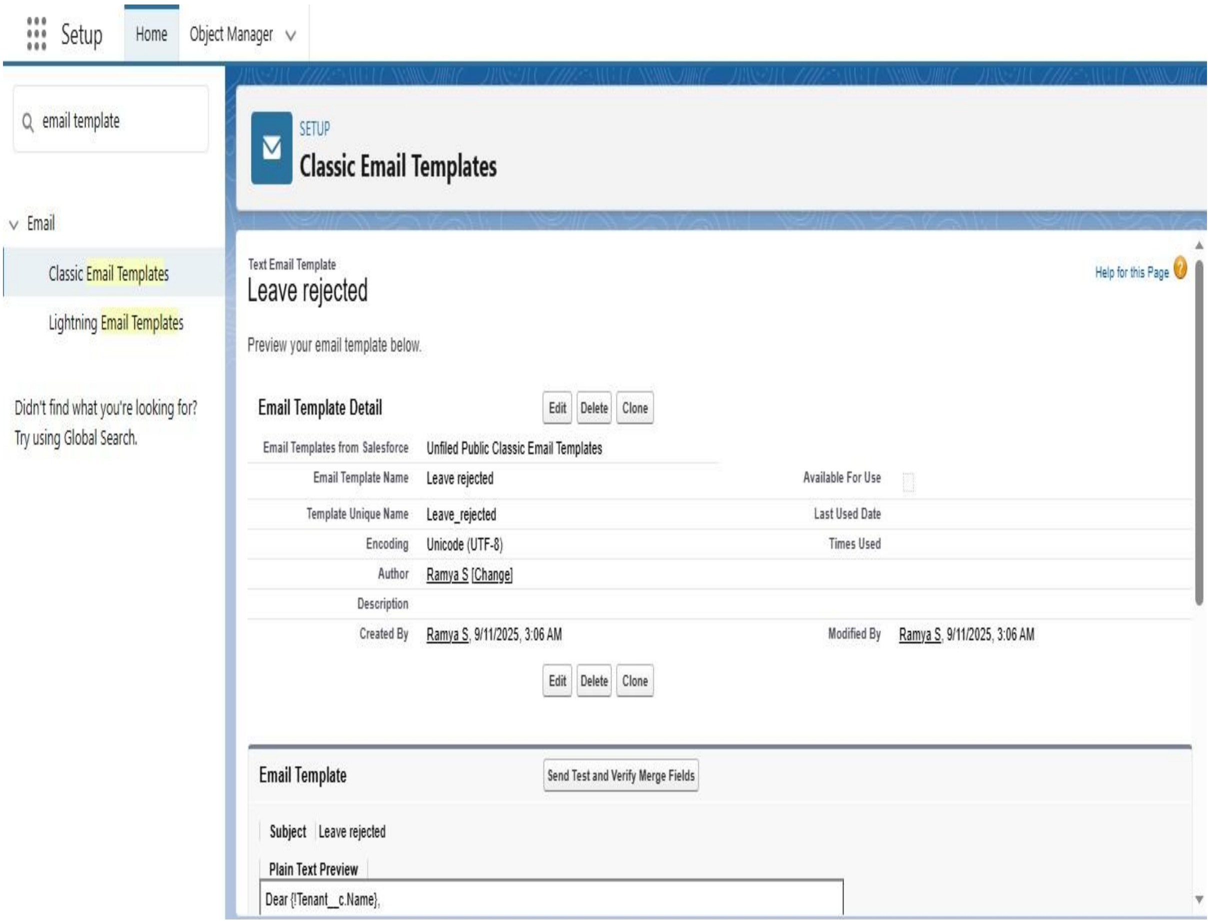
5. Subject: ” Leave rejected”

6. Email body: Dear {!Tenant__c.Name},

I hope this email finds you well. Your contract has not ended. So we can't approve your leave

your leave has rejected

7. Save



Text Email Template

Leave rejected

Preview your email template below.

Email Template Detail

Email Templates from Salesforce: Unfiled Public Classic Email Templates

Email Template Name: Leave rejected

Available For Use: □

Template Unique Name: Leave_rejected

Last Used Date:

Encoding: Unicode (UTF-8)

Times Used:

Author: Ramya S [Change]

Description:

Created By: Ramya S, 9/11/2025, 3:06 AM

Modified By: Ramya S, 9/11/2025, 3:06 AM

Email Template

Subject: Leave rejected

Plain Text Preview

Dear {!Tenant__c.Name},

- Go to setup in quick find box enter email template >> click on classic Email Template.

2. Click on >> New Email Template===>Choose text

   Folder: Unfiled public Classic Email templates

Click on available for use

3. Email Template Name is “Tenant Email”

4. Template Unique Name: Auto populated

5. Subject: ” Urgent: Monthly Rent Payment Reminder”

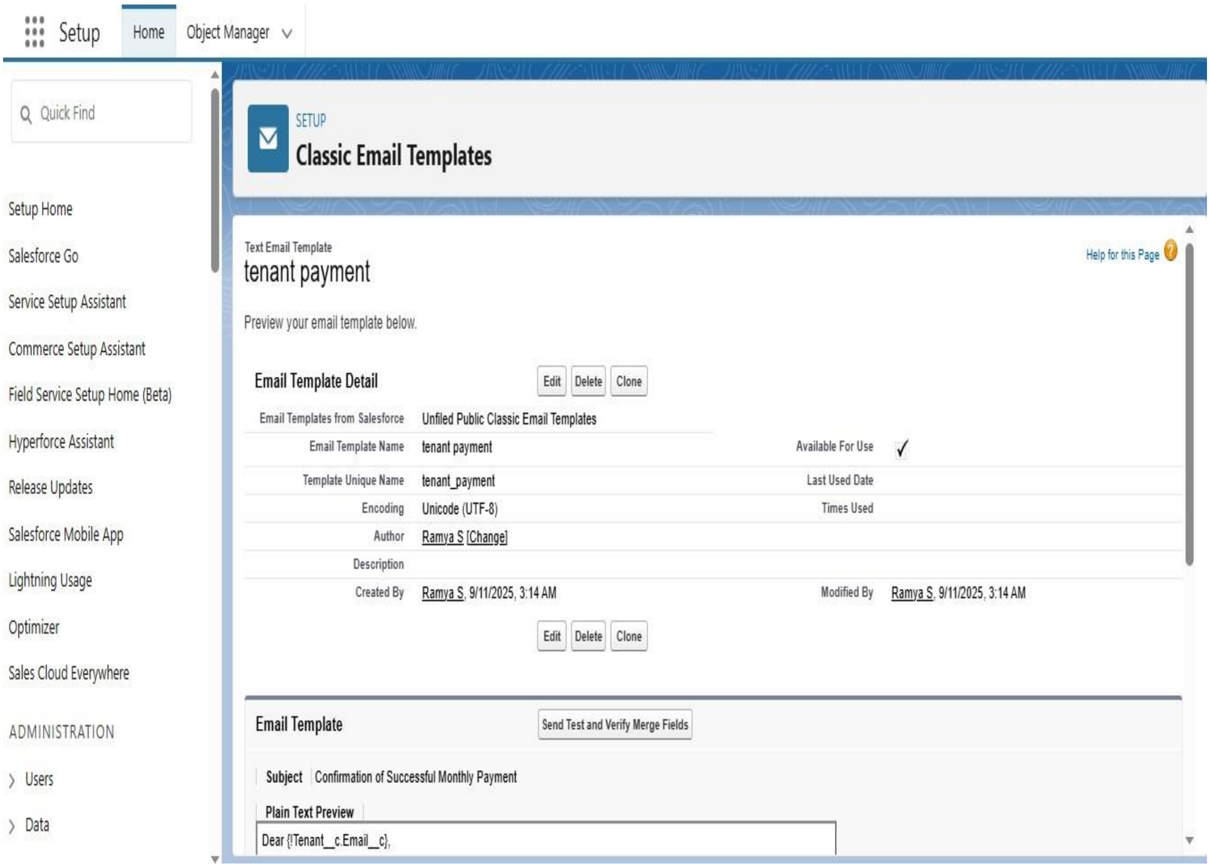
6. Email body:

Dear {!Tenant__c.Name},

I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding. As outlined in our rental agreement, the payment is due . To ensure the smooth operation of our property management and to avoid any inconvenience, we kindly request you to settle the payment at your earliest convenience.

7. Save



Text Email Template

tenant payment

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates		
Email Template Name	tenant payment	Available For Use	✓
Template Unique Name	tenant_payment	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Ramya S [Change]		
Description			
Created By	Ramya S, 9/11/2025, 3:14 AM	Modified By	Ramya S, 9/11/2025, 3:14 AM

Email Template

Subject Confirmation of Successful Monthly Payment

Plain Text Preview

Dear {!Tenant__c.Email__c},

Go to setup in quick find box enter email template >> click on classic Email Template.

2. Click on >> New Email Template===>Choose text

Folder: Unfiled public Classic Email templates

Click on available for use

3. Email Template Name is “tenant payment”

4. Template Unique Name: Auto populated

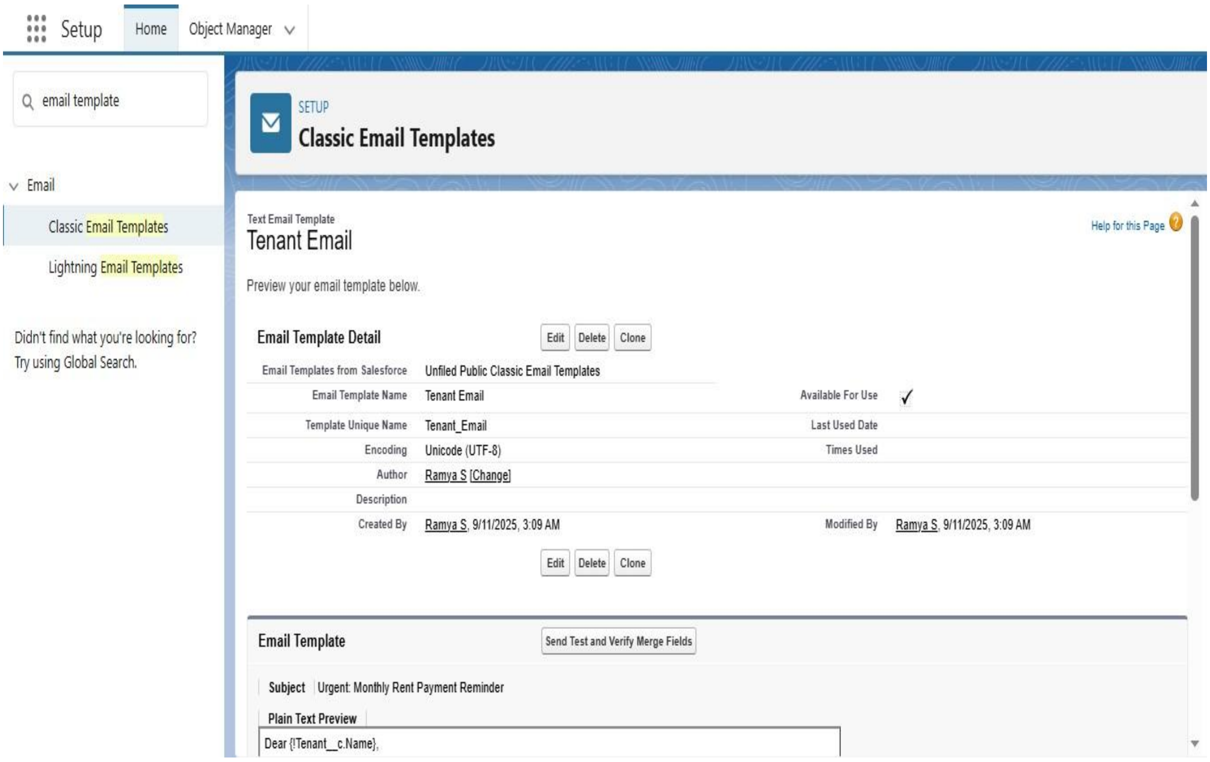
5. Subject: ” Confirmation of Successful Monthly Payment”

6. Email body:

Dear {!Tenant__c.Email__c},

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

7. Save



Text Email Template

Tenant Email

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates		
Email Template Name	Tenant Email	Available For Use	✓
Template Unique Name	Tenant_Email	Last Used Date	
Encoding	Unicode (UTF-8)	Times Used	
Author	Ramya S [Change]		
Description			
Created By	Ramya S, 9/11/2025, 3:09 AM	Modified By	Ramya S, 9/11/2025, 3:09 AM

Email Template

Subject Urgent: Monthly Rent Payment Reminder

Plain Text Preview

Dear {!Tenant__c.Name},

- Create Approval Process and initial submission.

To create fields in an object:

1.Go to setup >> Approval Processes in quick find bar>>click on it

Manage Approval Process For >> “Tenant” from the drop down.

3.Click on “Create New Approval Process” >> Use standard setup wizard.

4. Process Name “check for vacant” >> Click Next.

5. Field “Tenant:status” >> Operator: Not equals, Value >> Click on the lookup filter icon and select “Leaving”.

6.Click insert field,then click Next.

7. Next Automated Approver determined by “None” from the drop down.

8. Select the “Administrators ONLY can edit records during the approval process”. Then Next.

9. Click on next leave the email template click on next

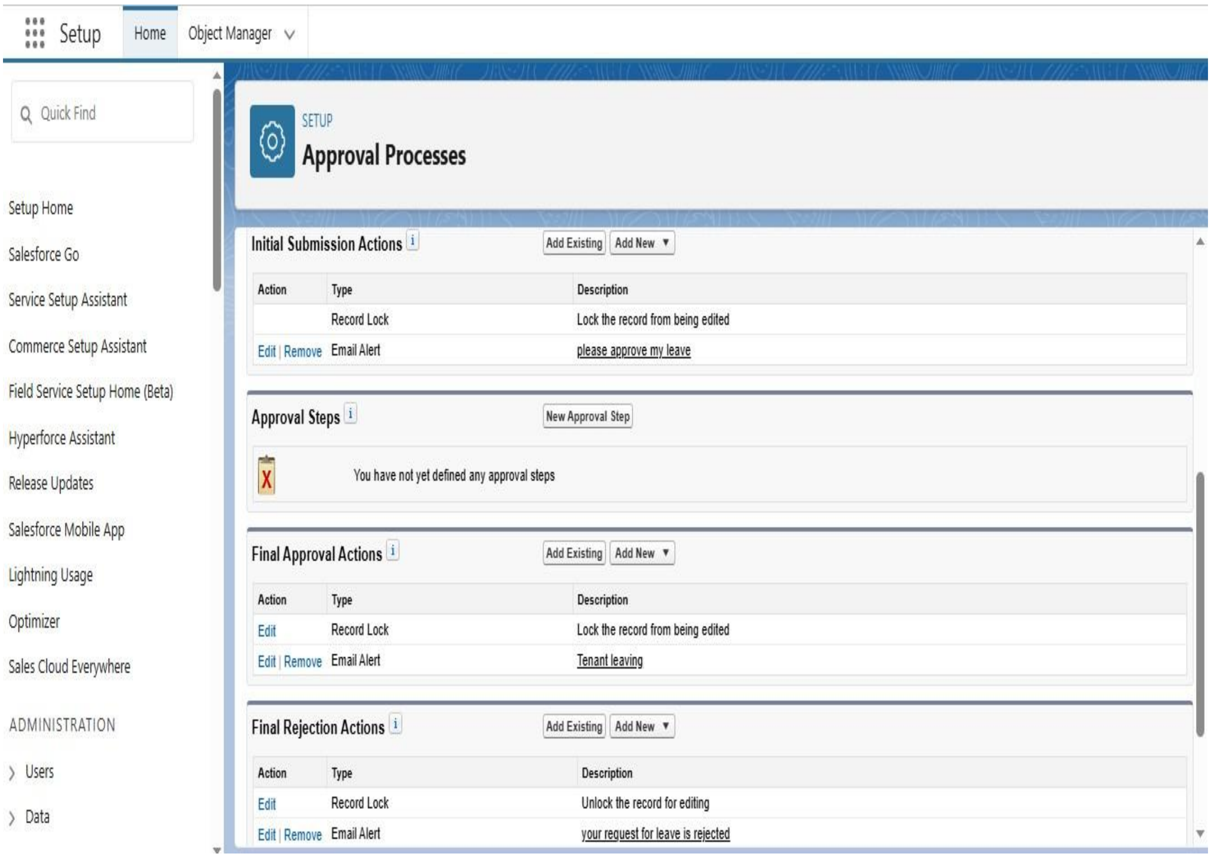
10.From the available fields select >> Tenant Name, and then add >>Add it to the selected.Then Next.

- Make sure Display approver history is checked.
- And under security settings check the “Allow approvers to access the approval page only from within the Salesforce application. (Recommended)” option.

11.Submitter type Search>>Owner, Allowed Submitters>>Property Owner.Then Next.

- Then click save.

- Click on “i’ll do this later. Take me back to the listing of all approval process for this object”
- Click go



Initial Submission Actions

Action	Type	Description
	Record Lock	Lock the record from being edited
Edit | Remove	Email Alert	please approve my leave

Approval Steps

You have not yet defined any approval steps

Final Approval Actions

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit | Remove	Email Alert	Tenant leaving

Final Rejection Actions

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit | Remove	Email Alert	your request for leave is rejected

1. Under initial submission action click on add new and then select email alert.

2.Description: “please approve my leave”.

3.unique name: auto populated

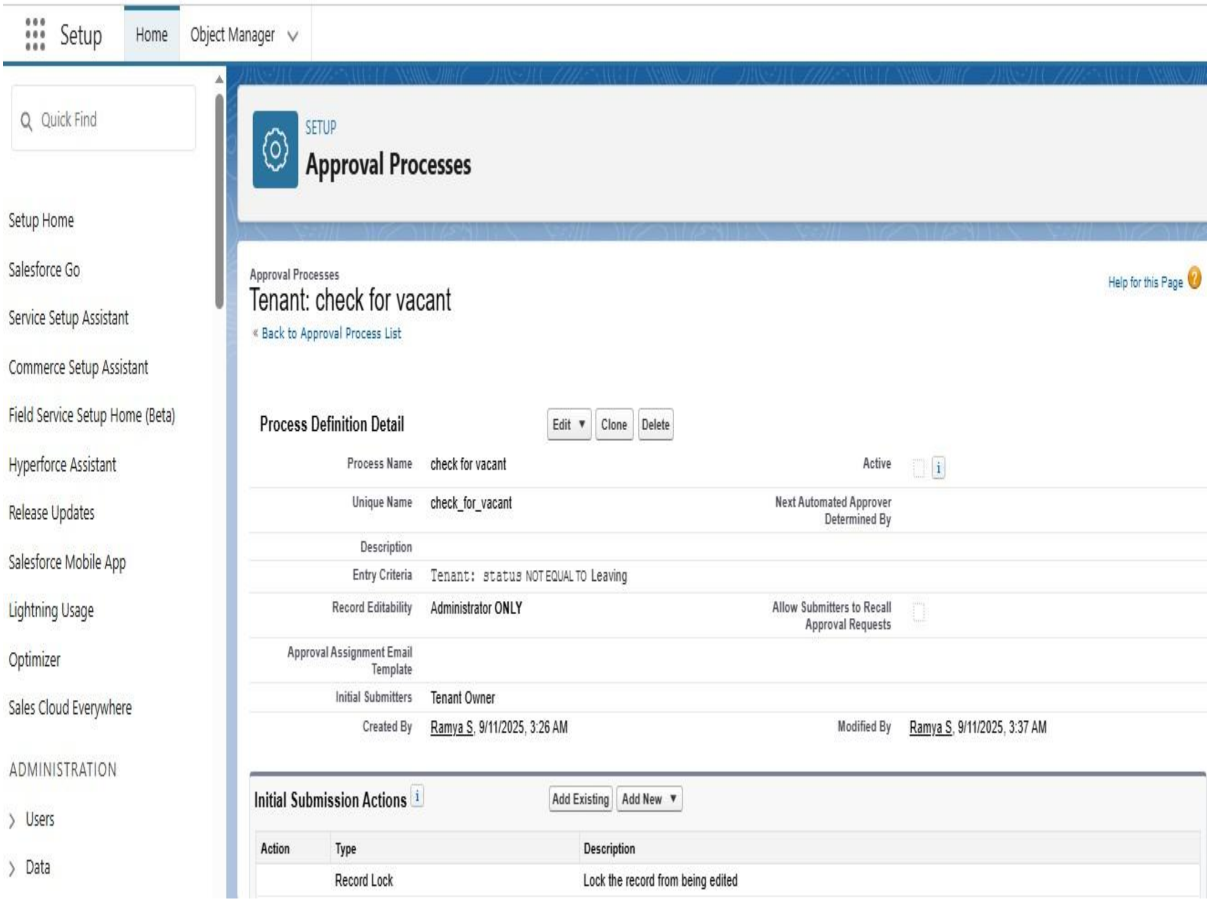
4.Email template: tenant leaving

5. Recipient type: Email field

6. Available Recipients: Email field: Email

7. From Email address: Current user’s email

8. Click save.



Approval Processes

Tenant: check for vacant

« Back to Approval Process List

Process Definition Detail

Process Name	check for vacant	Active	□
Unique Name	check_for_vacant	Next Automated Approver Determined By	
Description			
Entry Criteria	Tenant: status NOT EQUAL TO Leaving		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	□
Approval Assignment Email Template			
Initial Submitters	Tenant Owner		
Created By	Ramya S, 9/11/2025, 3:26 AM	Modified By	Ramya S, 9/11/2025, 3:37 AM

Initial Submission Actions

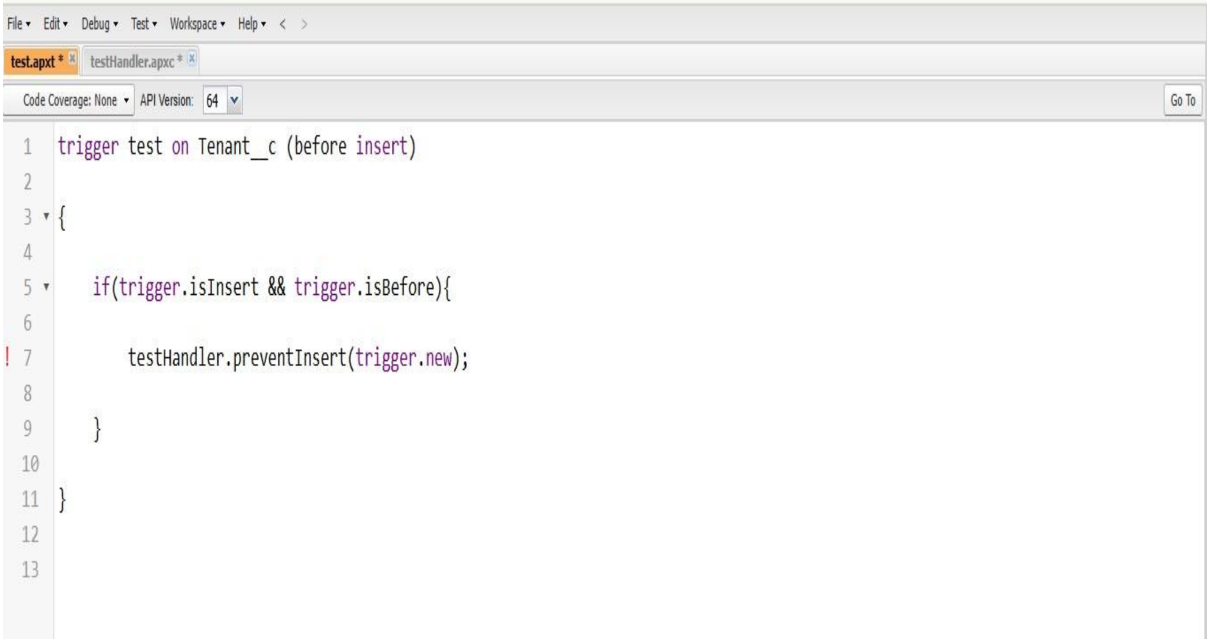
Action	Type	Description
	Record Lock	Lock the record from being edited

- Create an apex trigger and apex handler.

  1.To create a new Apex Class follow the below steps:
  Click on the file >> New ? Apex Class.

2.Give the Apex Trigger name as “test”, and select “Tenant c” from the dropdown for subject.

3.Click Submit.
4.Now write the code logic here

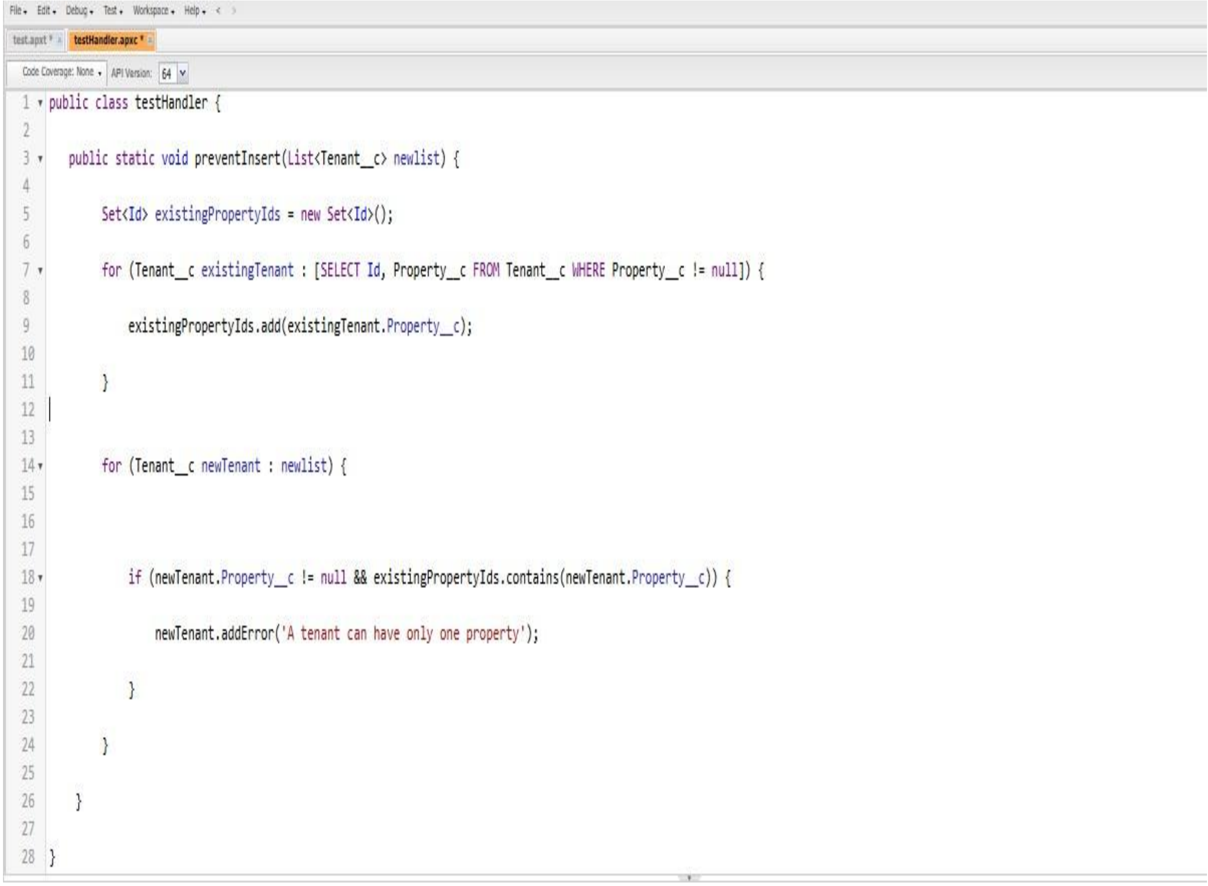


```
trigger test on Tenant__c (before insert)

{

    if(trigger.isInsert && trigger.isBefore){

        testHandler.preventInsert(trigger.new);

    }

}
```

- To create a new Apex Class follow the below steps:

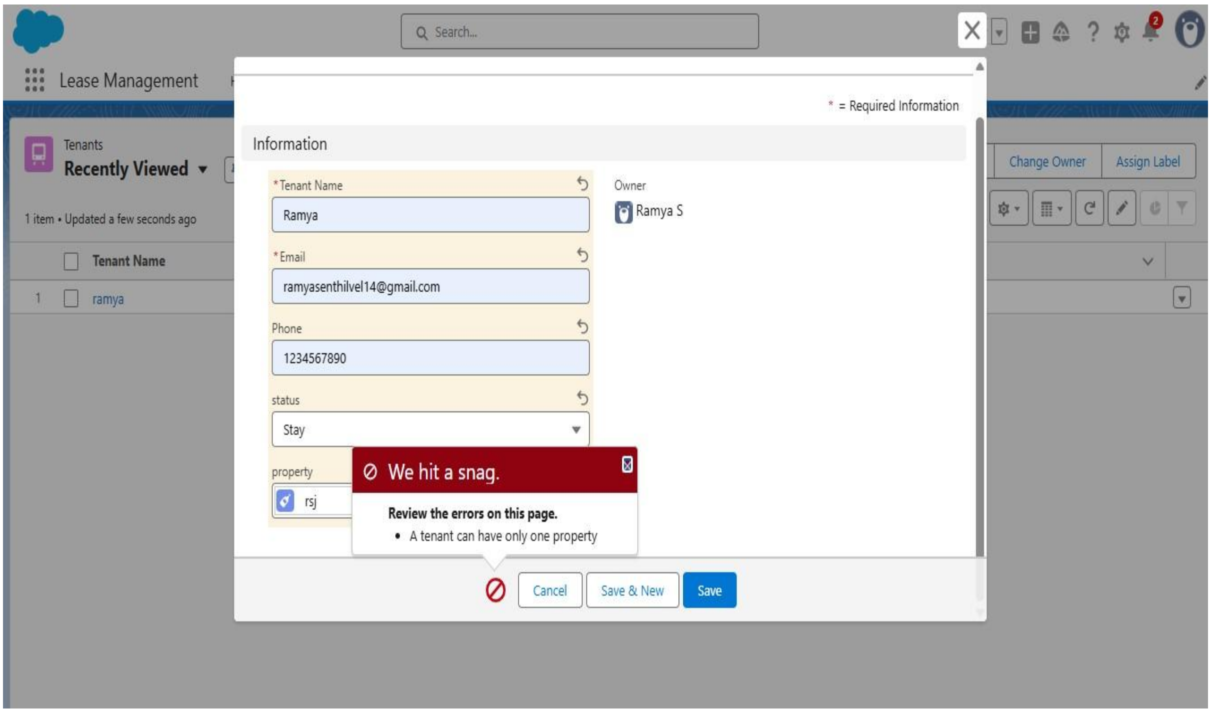
  Click on the file >> New >>Apex Class.

  2. Enter class name as testHandler.



```
public class testHandler {

    public static void preventInsert(List<Tenant__c> newlist) {

        Set<Id> existingPropertyIds = new Set<Id>();

        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {

            existingPropertyIds.add(existingTenant.Property__c);

        }


        for (Tenant__c newTenant : newlist) {



            if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {

                newTenant.addError('A tenant can have only one property');

            }

        }

    }

}
```

- Try to create new tenant with the existing property then it shows the error



1 item • Updated a few seconds ago

1 ramya

* = Required Information

Information

*Tenant Name: Ramya

Owner: Ramya S

*Email: ramyasenthilvel14@gmail.com

Phone: 1234567890

status: Stay

property: rsj

We hit a snag.

Review the errors on this page.

- A tenant can have only one property

Create Flow for monthly payment

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.
3. Under Object select ”Payment for tenant”. Click on A record is updated.
4. Set entry condition

Under Condition Requirements

All Conditions are met

Field: check_for_payment__c	Operator: Equals	Value : paid

5. Click on : Every time a record is updated and meets the condition requirements

6. Click on : Actions and related records,done

7. Under record trigger flow click on “+” icon and select action

In action search for send email then click on send email (check below image)

8. Label : send email

API Name : send_email

9. Label : send email

10. API Name : send_email

11. Enable Body

12. Click on new resource

Under resource type select “Text Template”

API Name : emailbody

Under body: (paste the below text)

Dear {!$Record.Tenant__r.Name},

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

14. Click Done.

15. Enable recipient Address List

Paste this ?{!$Record.Tenant__r.Email__c}

16. Click Done

17. Enable subject

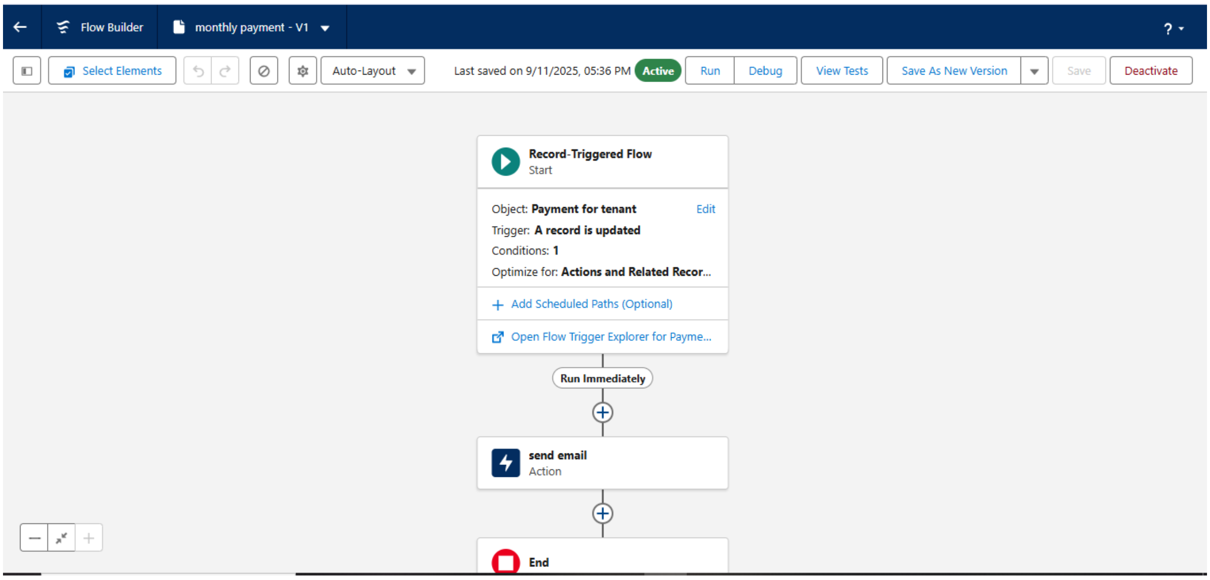
Pate this >> Confirmation of Successful Monthly Payment

18. Click on save

Flow label : monthly payment

Flow API Name : monthly_payment

Click on activate



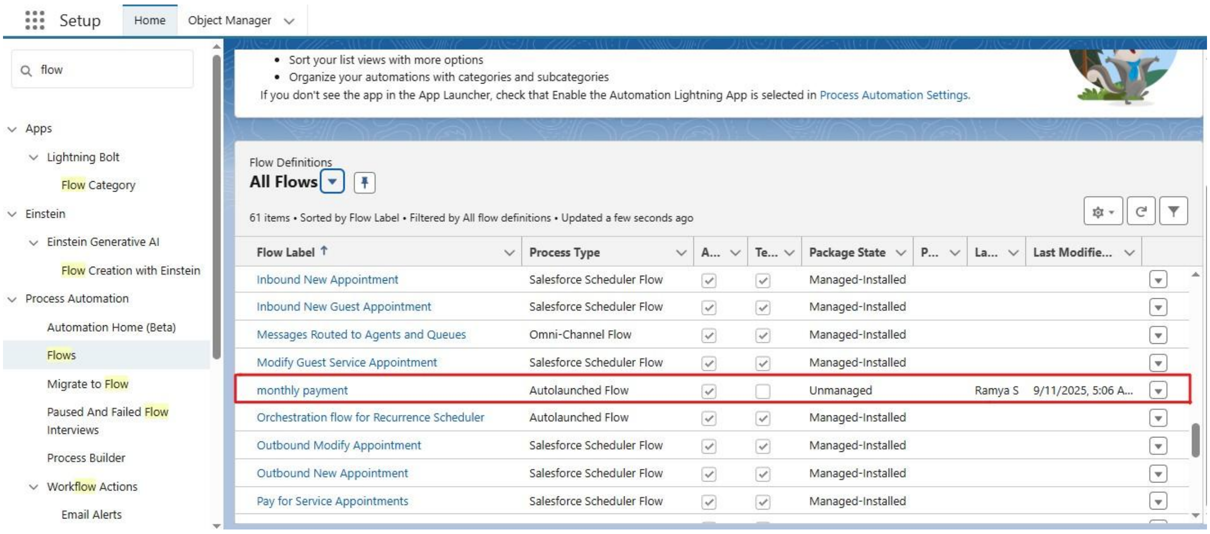
Last saved on 9/11/2025, 05:36 PM Active

Record-Triggered Flow
Start

Object: **Payment for tenant** Edit
Trigger: **A record is updated**
Conditions: **1**
Optimize for: **Actions and Related Recor...**

+ Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Payme...



- Sort your list views with more options
- Organize your automations with categories and subcategories

If you don't see the app in the App Launcher, check that Enable the Automation Lightning App is selected in Process Automation Settings.

Flow Definitions
All Flows

61 items • Sorted by Flow Label • Filtered by All flow definitions • Updated a few seconds ago

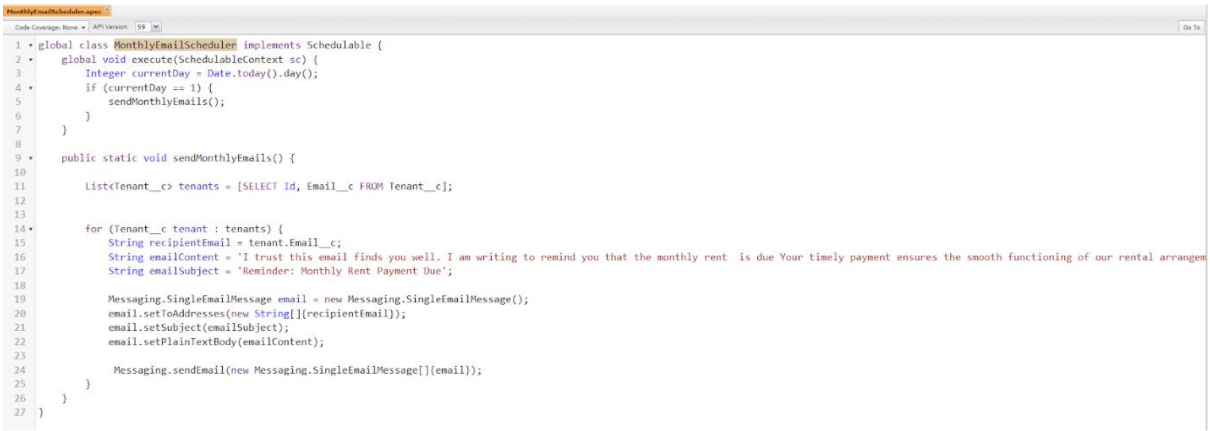
Flow Label	Process Type	A...	Te...	Package State	P...	La...	Last Modifie...
Inbound New Appointment	Salesforce Scheduler Flow	☑	☑	Managed-Installed			
Inbound New Guest Appointment	Salesforce Scheduler Flow	☑	☑	Managed-Installed			
Messages Routed to Agents and Queues	Omni-Channel Flow	☑	☑	Managed-Installed			
Modify Guest Service Appointment	Salesforce Scheduler Flow	☑	☑	Managed-Installed			
monthly payment	Autolaunched Flow	☑	□	Unmanaged		Ramya S	9/11/2025, 5:06 A...
Orchestration flow for Recurrence Scheduler	Autolaunched Flow	☑	☑	Managed-Installed			
Outbound Modify Appointment	Salesforce Scheduler Flow	☑	☑	Managed-Installed			
Outbound New Appointment	Salesforce Scheduler Flow	☑	☑	Managed-Installed			
Pay for Service Appointments	Salesforce Scheduler Flow	☑	☑	Managed-Installed			

Create an Apex Class

1. To create a new Apex Class follow the below steps:

Click on the file >> New >> Apex Class.

2. Enter class name as MonthlyEmailScheduler.



```
global class MonthlyEmailScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        Integer currentDay = Date.today().day();
        if (currentDay == 1) {
            sendMonthlyEmails();
        }
    }

    public static void sendMonthlyEmails() {

        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];


        for (Tenant__c tenant : tenants) {
            String recipientEmail = tenant.Email__c;
            String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent  is due Your timely payment ensures the smooth functioning of our rental arrangem
            String emailSubject = 'Reminder: Monthly Rent Payment Due';

            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
            email.setToAddresses(new String[]{recipientEmail});
            email.setSubject(emailSubject);
            email.setPlainTextBody(emailContent);

             Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
        }
    }
}
```

Apex logic:

```
global class MonthlyEmailScheduler implements Schedulable {
   global void execute(SchedulableContext sc) {
      Integer currentDay = Date.today().day();
      if (currentDay == 1) {
         sendMonthlyEmails();
      }
   }

   public static void sendMonthlyEmails() {

      List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];


      for (Tenant__c tenant : tenants) {
         String recipientEmail = tenant.Email__c;
         String emailContent = 'I trust this email finds you well. I am writing to remind you that
the monthly rent  is due Your timely payment ensures the smooth functioning of our rental
arrangement and helps maintain a positive living environment for all.';
         String emailSubject = 'Reminder: Monthly Rent Payment Due';

         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
         email.setToAddresses(new String[]{recipientEmail});
         email.setSubject(emailSubject);
         email.setPlainTextBody(emailContent);

          Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
      }
   }
}
```

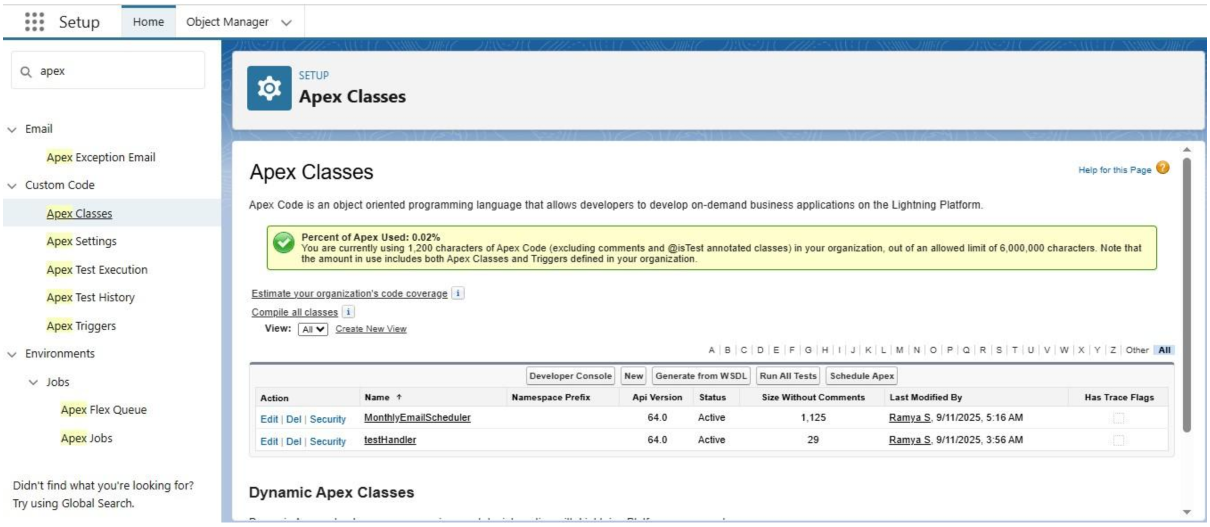
Save the code.

Schedule Apex class

1. Enter Apex class in quick find box
2. Select schedule Apex
3. Enter job Name : MonthlyEmailScheduler
4. Apex class : MonthlyEmailScheduler
5. Frequency : Monthly===>select on day 1
6. Start date : 09/11/2025
7. End date : 10/11/2025
8. Preferred start time : 03:00 am
9. save

Testing the approval process

Enter any comment am\nd click on submit



Apex Classes

Apex Classes

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

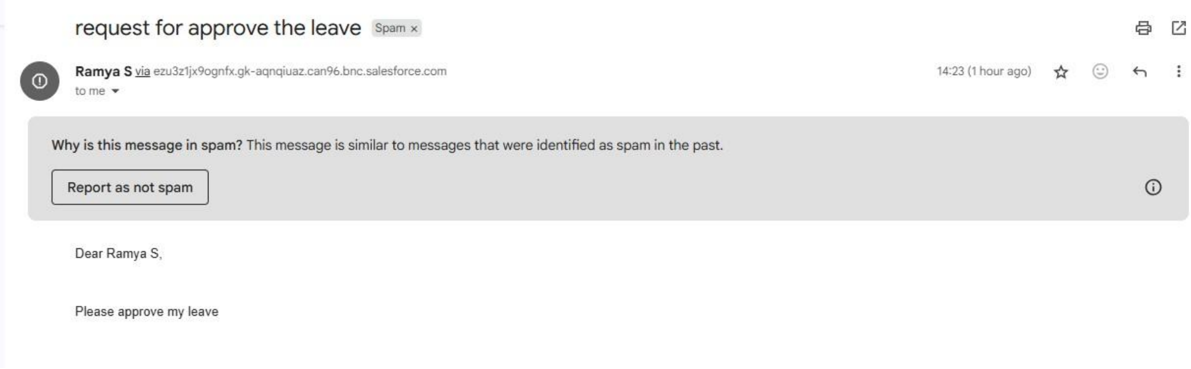
Percent of Apex Used: 0.02%
You are currently using 1,200 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Action	Name ↑	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit | Del | Security	MonthlyEmailScheduler		64.0	Active	1,125	Ramya S, 9/11/2025, 5:16 AM	□
Edit | Del | Security	testHandler		64.0	Active	29	Ramya S, 9/11/2025, 3:56 AM	□

Dynamic Apex Classes

RESULT

Request for Approve the leave



request for approve the leave

Ramya S via ezu3z1jx9ognfx.gk-aqnqiuaz.can96.bnc.salesforce.com
to me

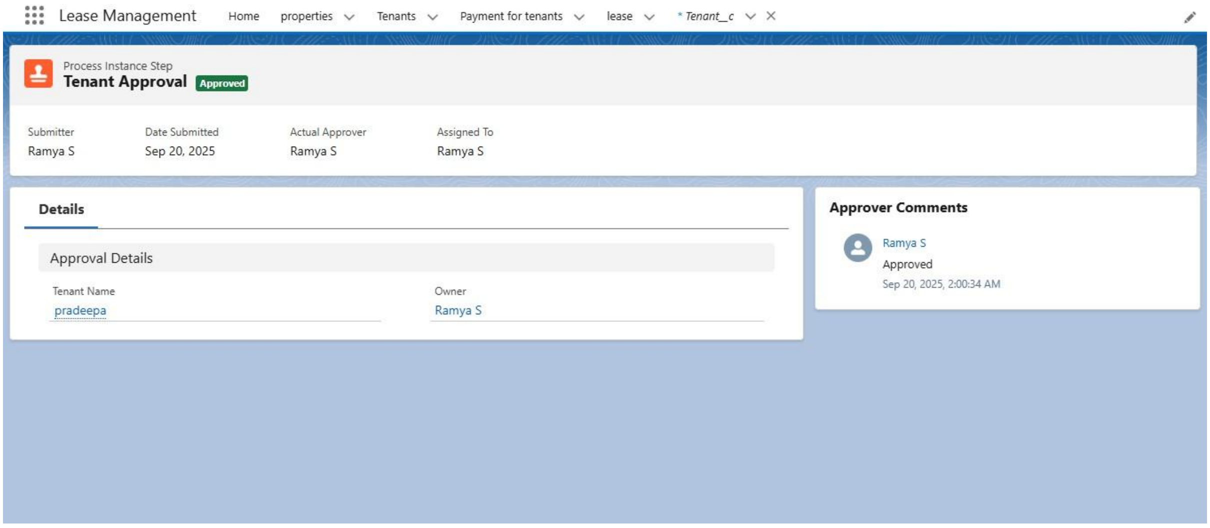
14:23 (1 hour ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear Ramya S,

Please approve my leave

Leave Approved



Process Instance Step
Tenant Approval Approved

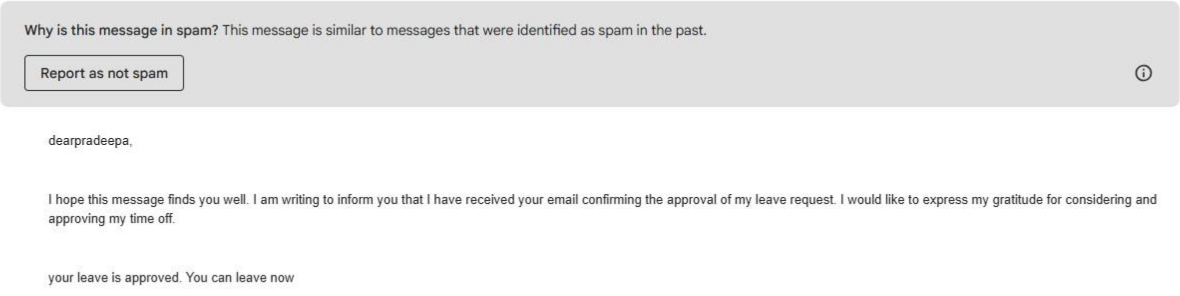
Submitter	Date Submitted	Actual Approver	Assigned To
Ramya S	Sep 20, 2025	Ramya S	Ramya S

Approval Details

Tenant Name	Owner
pradeepa	Ramya S

Approver Comments

Ramya S
Approved
Sep 20, 2025, 2:00:34 AM



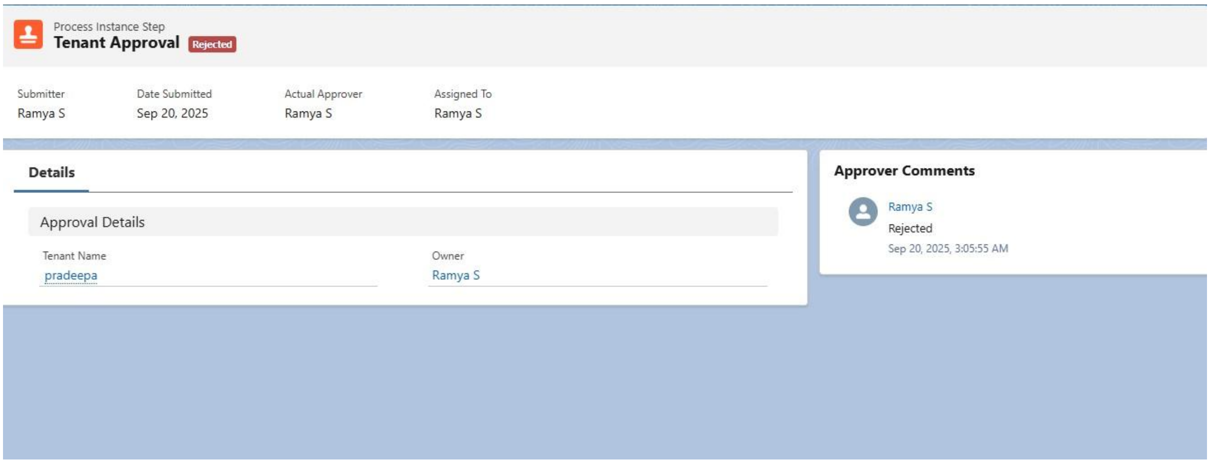
Why is this message in spam? This message is similar to messages that were identified as spam in the past.

dearpradeepa,

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

your leave is approved. You can leave now

Leave Rejected



Process Instance Step

Tenant Approval Rejected

Submitter Ramya S

Date Submitted Sep 20, 2025

Actual Approver Ramya S

Assigned To Ramya S

Details

Approval Details

Tenant Name pradeepa

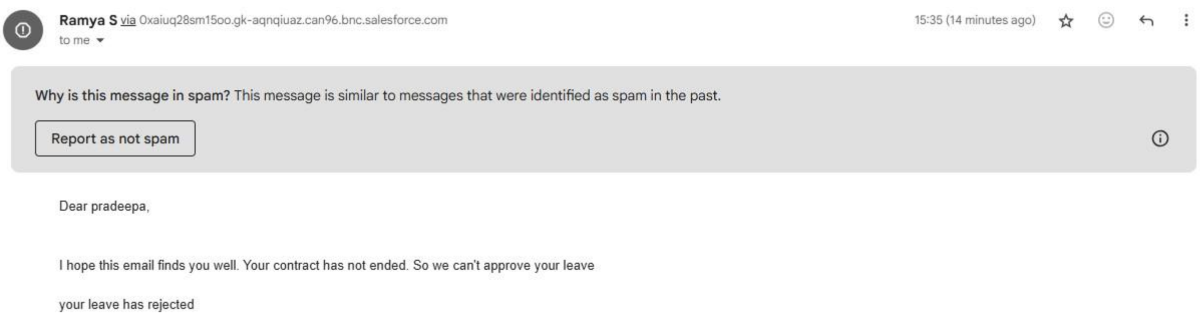
Owner Ramya S

Approver Comments

Ramya S

Rejected

Sep 20, 2025, 3:05:55 AM



Ramya S via 0xaiuq28sm15oo.gk-aqnqiuaz.can96.bnc.salesforce.com

to me

15:35 (14 minutes ago)

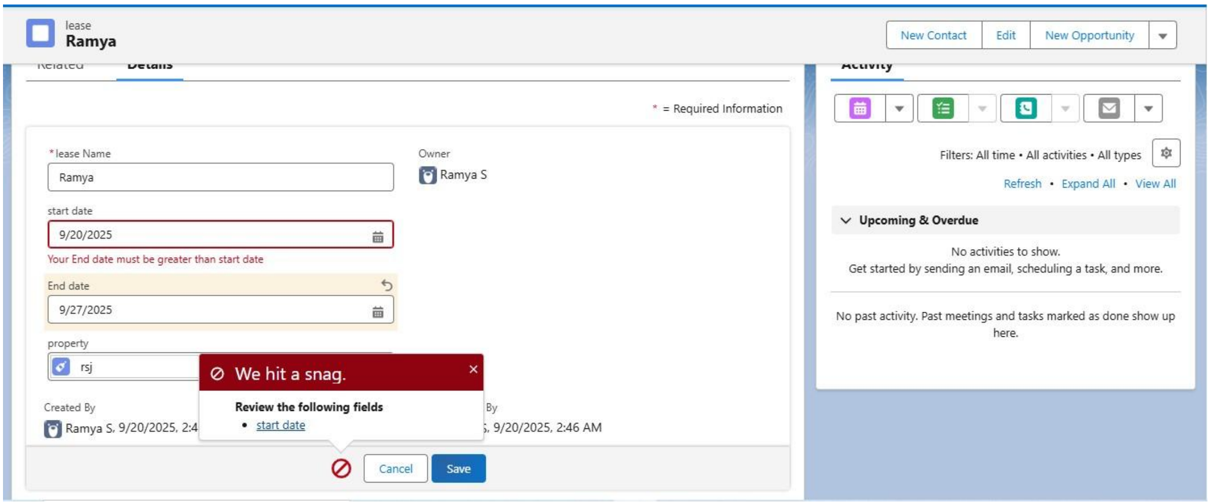
Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear pradeepa,

I hope this email finds you well. Your contract has not ended. So we can't approve your leave

your leave has rejected

Validation Rule



lease

Ramya

* = Required Information

*lease Name Ramya

Owner Ramya S

start date 9/20/2025

Your End date must be greater than start date

End date 9/27/2025

property rsj

Created By Ramya S, 9/20/2025, 2:4

By …, 9/20/2025, 2:46 AM

We hit a snag.

Review the following fields

- start date

Filters: All time • All activities • All types

Upcoming & Overdue

No activities to show.
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

Conclusion

The Lease Management System successfully streamlines the operations of leasing through a structured, automated Salesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.

Appendix

.source code : provide in apex class and triggers

Test.apxt:

```
trigger test on Tenant__c (before insert) { if
(trigger.isInsert && trigger.isBefore){
testHandler.preventInsert(trigger.new);
} }
```

testHandler.apxc:

```
public class testHandler { public static void preventInsert(List<
Tenant__c> newlist) { Set<Id> existingPropertyIds
= new Set<Id>()
for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c
WHERE Property__c != null]) { existingPropertyIds.add(existingTenant.Property__c;
} for (Tenant__c newTenant :
newlist) {
if (newTenant.Property__c != null &&
existingPropertyIds.contains(newTenant.Property__c)) { newTenant.addError('A tenant can have only one property');
}
}
}
}
```

MothlyEmailScheduler.apxc:

```
global class MonthlyEmailScheduler implements Schedulable { global void execute(SchedulableContext sc) { Integer currentDay = Date.today().day(); if (currentDay == 1) { sendMonthlyEmails();
}
} public static void sendMonthlyEmails() { List<Tenant__c> tenants = [SELECT Id, Email__c
FROM

Tenant__c]; for (Tenant__c tenant :

tenants) {

String recipientEmail = tenant.Email__c;

String emailContent = 'I trust this email finds you well. I am writing to remind you that the
monthly rent is due Your timely payment ensures the smooth functioning of our rental
arrangement and helps maintain a positive living environment for all.';

String emailSubject = 'Reminder: Monthly Rent Payment Due';

Messaging.SingleEmailMessage email = new

Messaging.SingleEmailMessage(); email.setToAddresses(new String[]{recipientEmail});
email.setSubject(emailSubject); email.setPlainTextBody(emailContent);

Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});

}

}

}
```