LEASE MANAGEMENT

SALES FORCE

COLLEGE: AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY [AVEV]

TEAM ID : LTVIP2025TMID29184

TEAM MEMBERS: GOKADA SRAVYA [22Q71A0554]

Mail: (22q71a0554@aietta.ac.in)

TEAM Leader: ALLU MOHANA RAO [22Q71A0503]

DAMARASINGI TEJA [22Q71A0534]

INDEX:

Salesforce

Introduction:

Salesforce is a cloud-based customer relationship management (CRM) platform that enables businesses to manage sales, marketing, and customer service activities in a single platform. It provides tools for sales teams to track leads, manage accounts, and forecast sales, as well as marketing automation and customer service capabilities. Salesforce is highly customizable and scalable, making it a popular choice for businesses of all sizes. Its features include:

- Contact and account management
- Sales forecasting and pipeline management
- Marketing automation
- Customer service and support
- Analytics and reporting
- Integration with other business applications

Salesforce is widely used across various industries, and its platform is constantly evolving with new features and innovations.

What is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this: https://youtu.be/r9EX3IGde5k

Overview

Here's a project overview of Lease Management in Salesforce:

Project Title: Lease Management System

Objective: Develop a customized Salesforce application to manage leases, track lease agreements, and automate lease-related processes for a real estate company.

Key Features:

1. Lease Agreement Management: Store and manage lease agreements, including lease terms, rent, and security deposits.
2. Lease Tracking: Track lease start and end dates, renewal options, and notice periods.
3. Rent Management: Manage rent payments, payment schedules, and late payment fees.
4. Lease Renewal and Termination: Automate lease renewal and termination processes, including notifications and reminders.
5. Reporting and Analytics: Generate reports on lease performance, rent payments, and lease expirations.

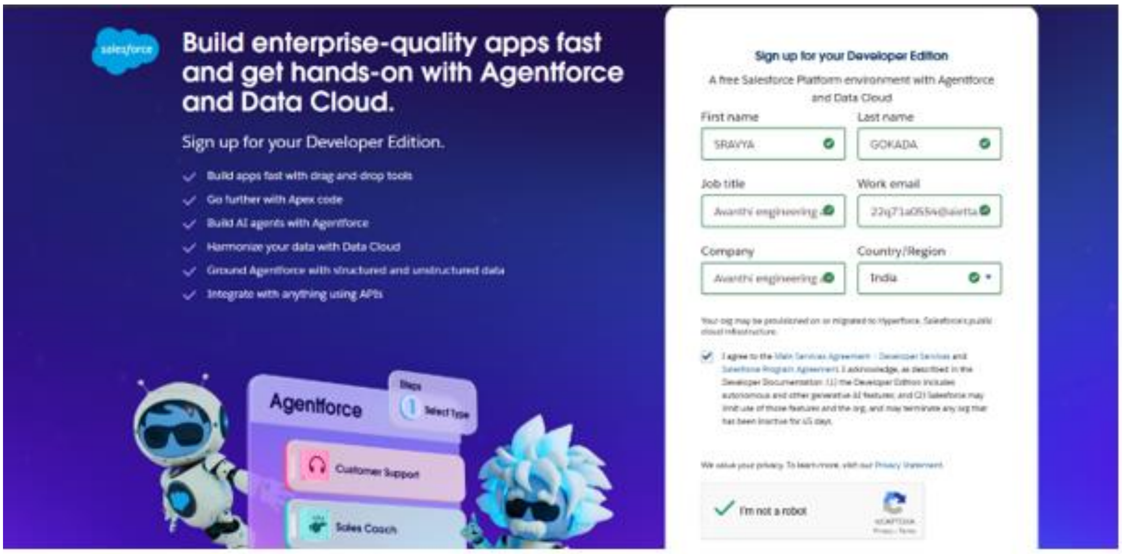
Salesforce Features Used:

1. Custom Objects: Create custom objects for Lease Agreements, Rent Payments, and Lease Terms.
2. Workflows and Approvals: Automate lease approval and renewal processes using workflows and approvals.
3. Triggers and Validation Rules: Use triggers and validation rules to enforce data consistency and accuracy.
4. Dashboards and Reports: Create custom dashboards and reports to provide insights into lease performance and rent payments.

Creating Developer Account:

Creating a developer org in salesforce.

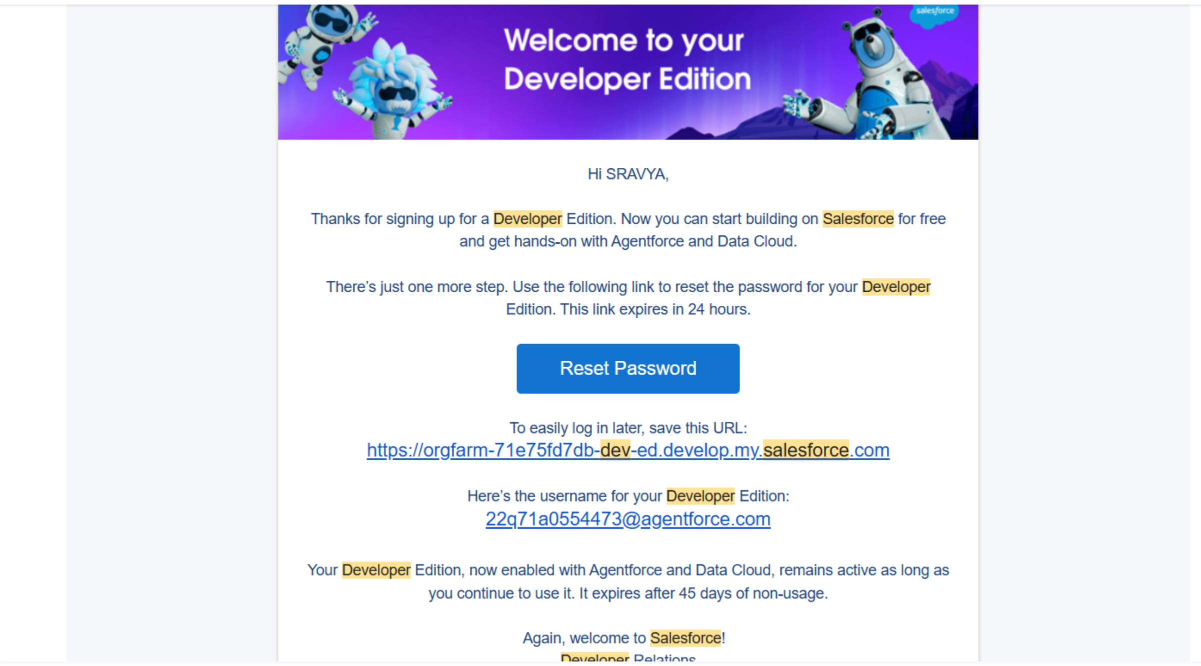
1. Go to https://developer.salesforce.com/signup
2. On the sign up form, enter the following details:



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

Account Activation:

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5 10mins

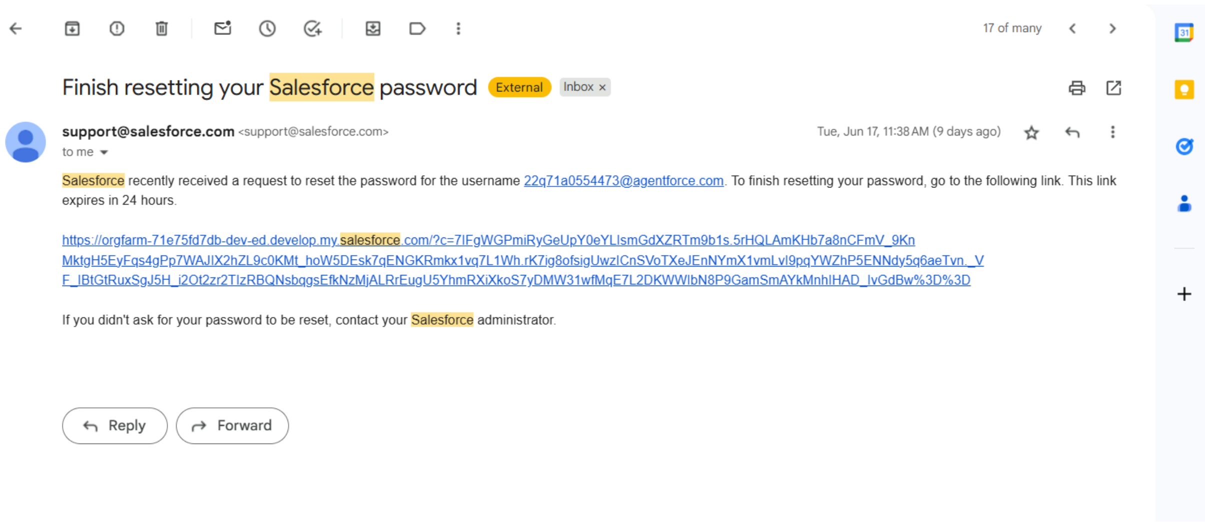


1.Click on Verify Account

2. Give a password and answer a security question and click on change password.

3. Give a password and answer a security question and click on change password.

4. Then you will redirect to your salesforce setup page.



DATA MODEL

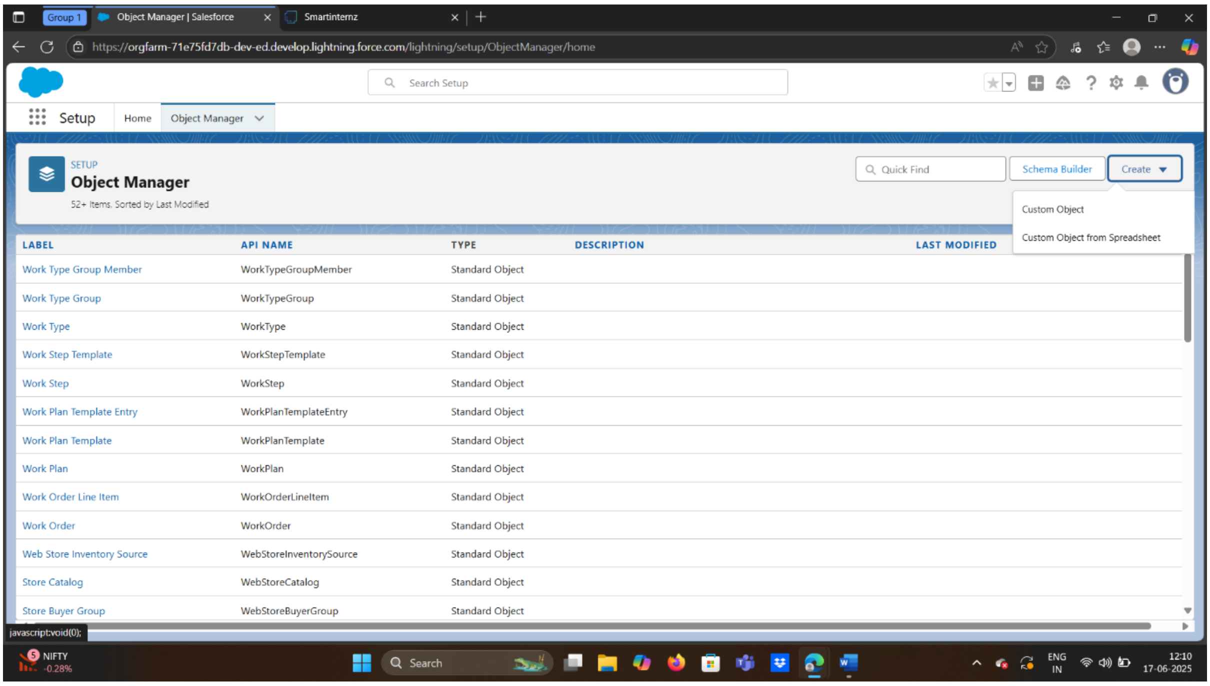
OBJECTS:

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects

Salesforce objects are of two types:

1. Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.

2. Custom Objects: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.



Create Property Object:

To create an object

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

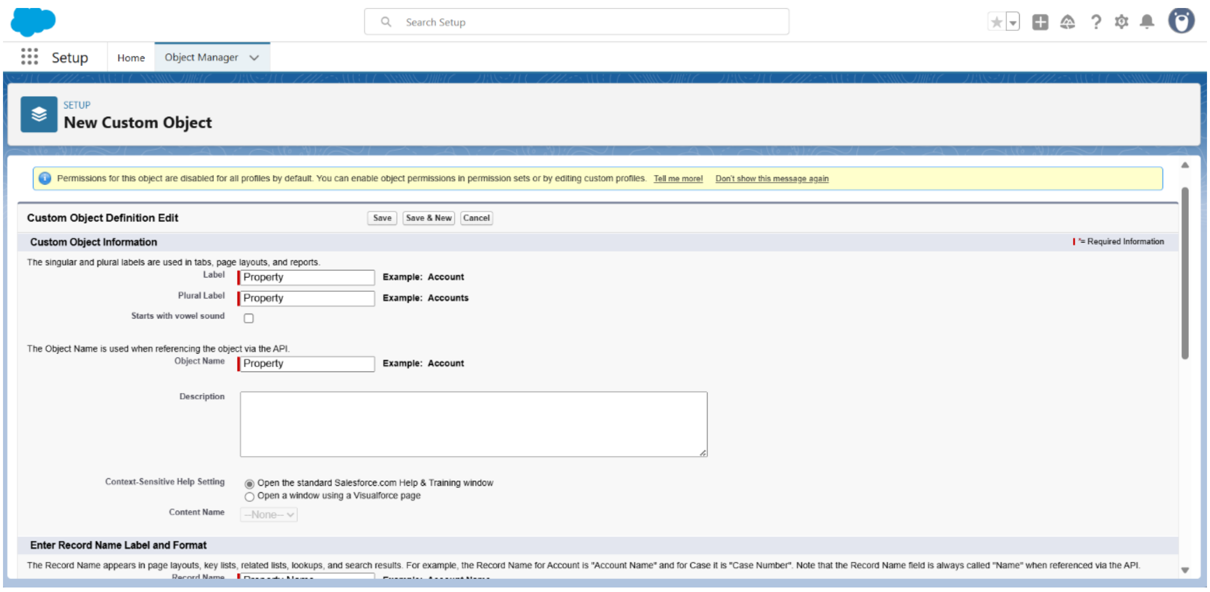
1. Enter the label name>> property
2. Plural label name>> property
3. Enter Record Name Label and Format

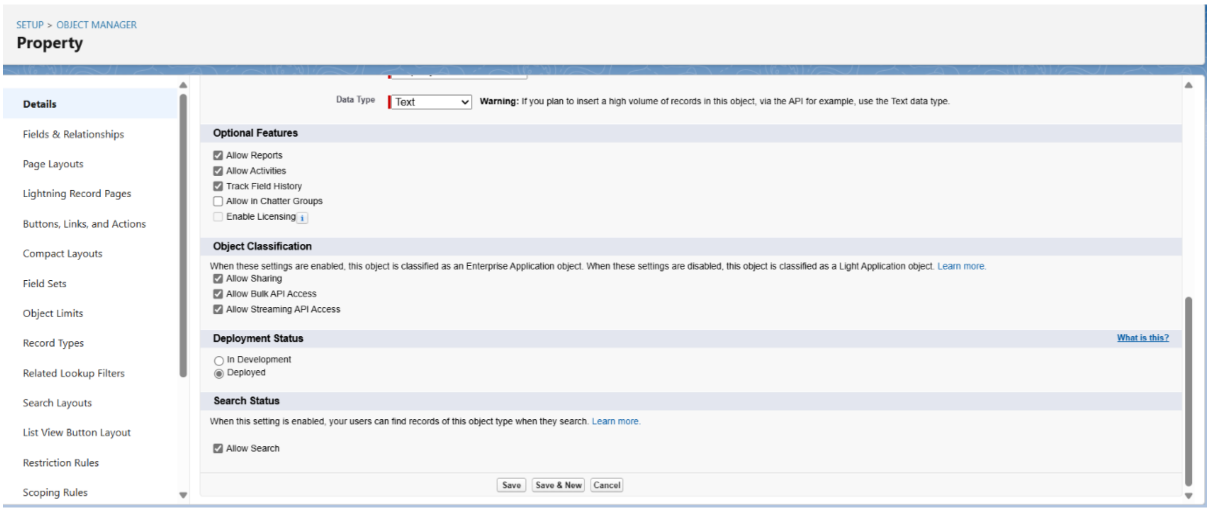
   .Record Name>>property Name

   .Data Type Text

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

3. Allow search >>Save



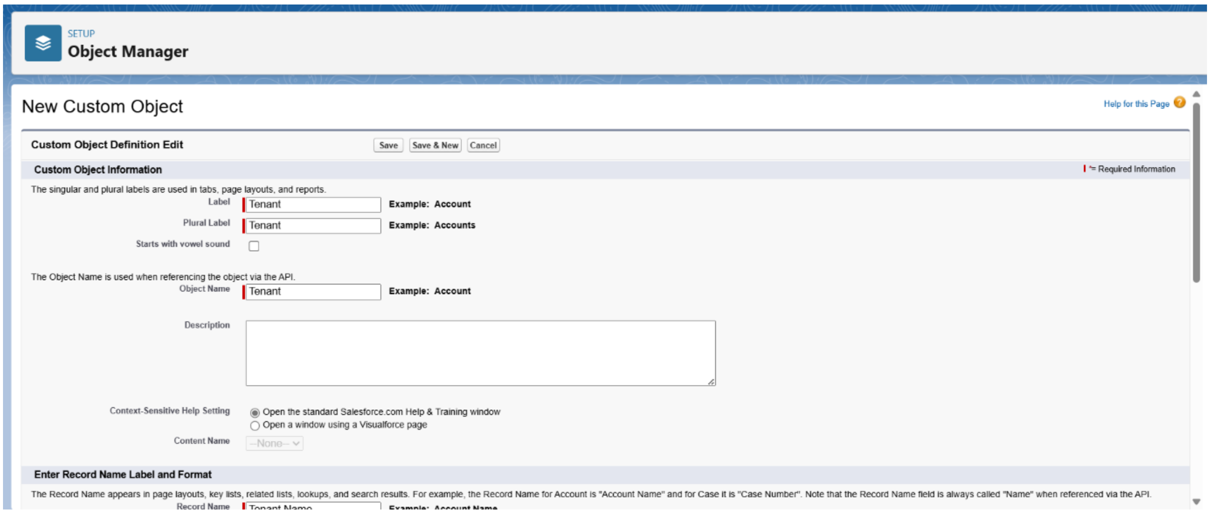


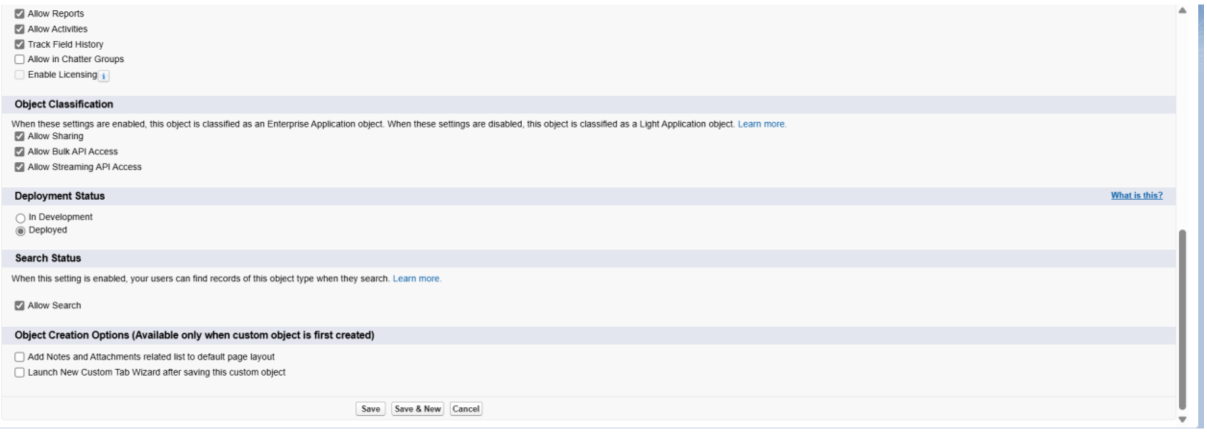
Create Tenant Object:

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

       To
       - Record Name >> Tenant Name
       - Data Type >> Text
2. Click on Allow reports and Track Field History Allow Activities
3. Allow search >> Save.

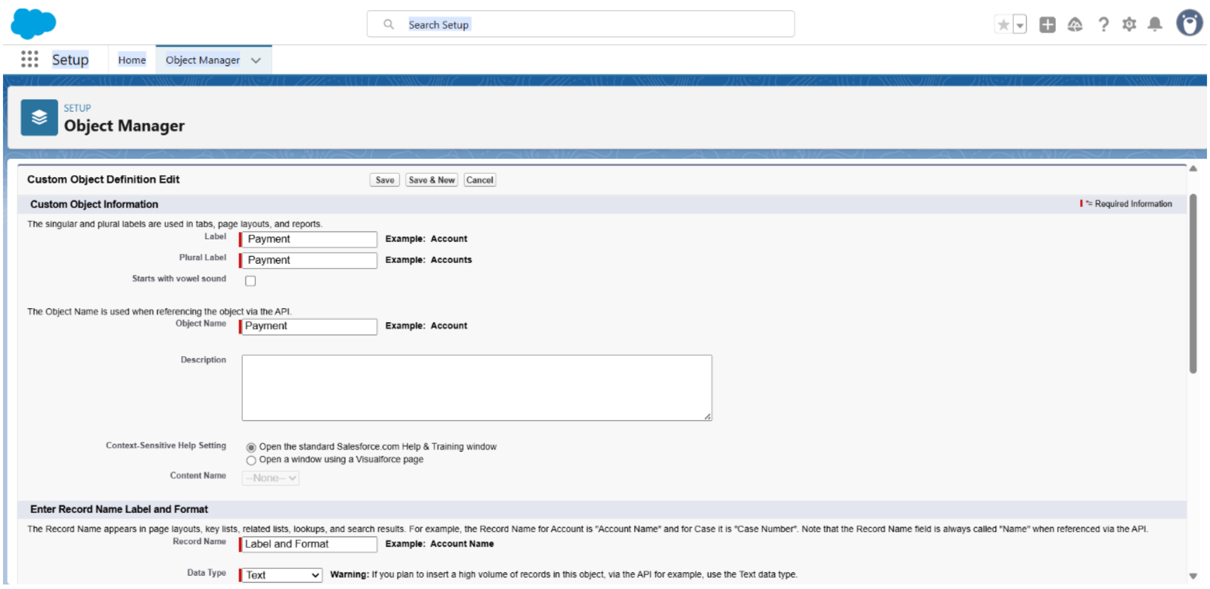


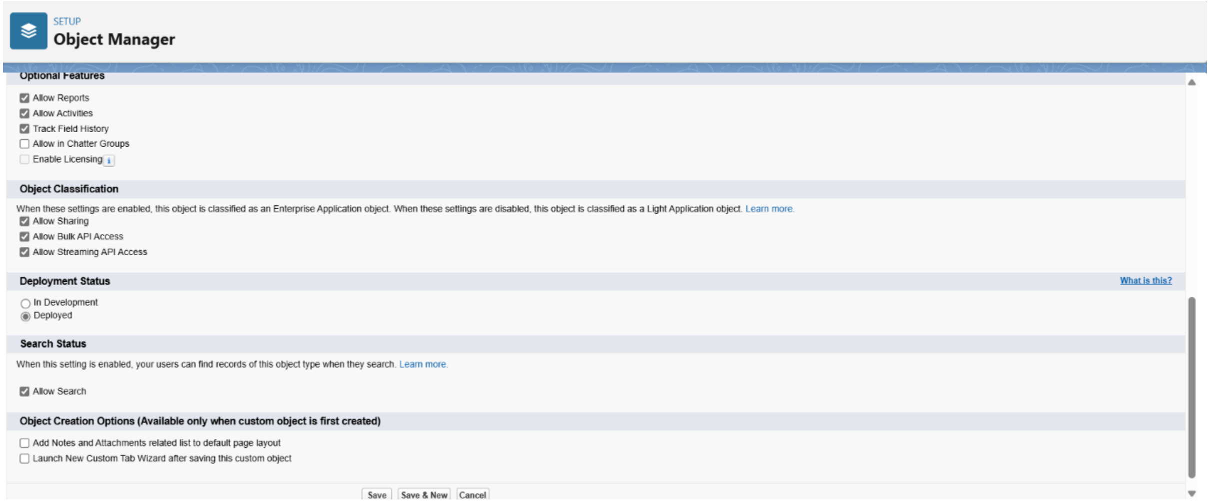


Create Payment Object

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>> Payment for tenanat
    2. Plural label name>> Payment
    3. Enter Record Name Label and Format
        - Record Name >> Payment Name
        - Data Type >> Text
2. Click on Allow reports and Track Field History, Allow Activities Allow search >> Save.

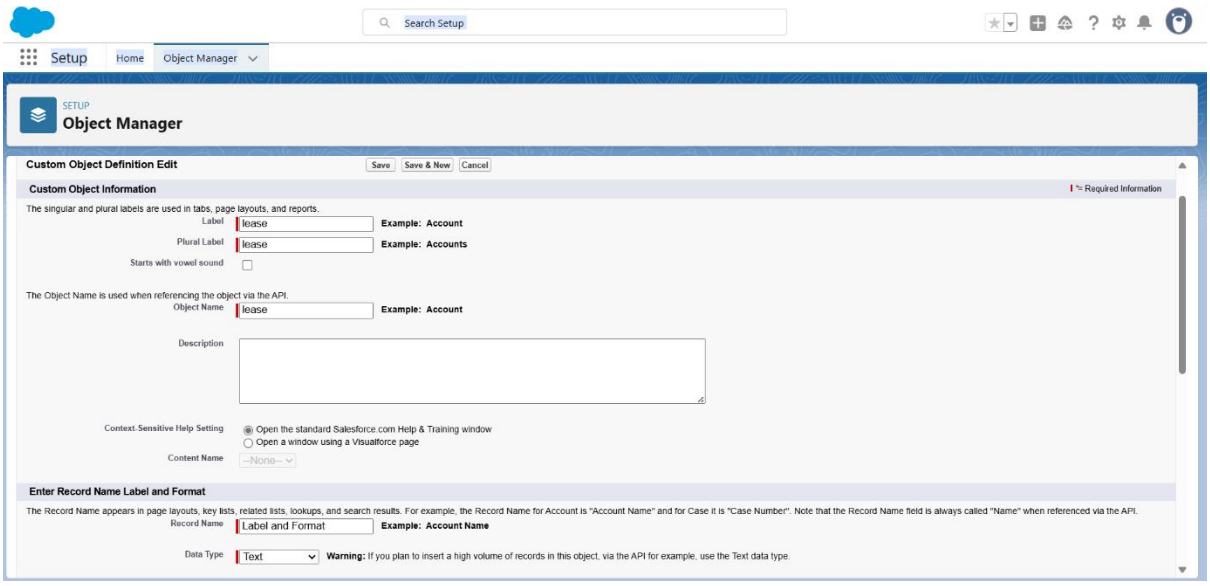




Create Lease Object

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 and Save.

Tabs

What is Tab: A tab is like a user interface that is used to build records for objects and to view the records in the objects.

Types of Tabs:

1. Custom Tabs Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as

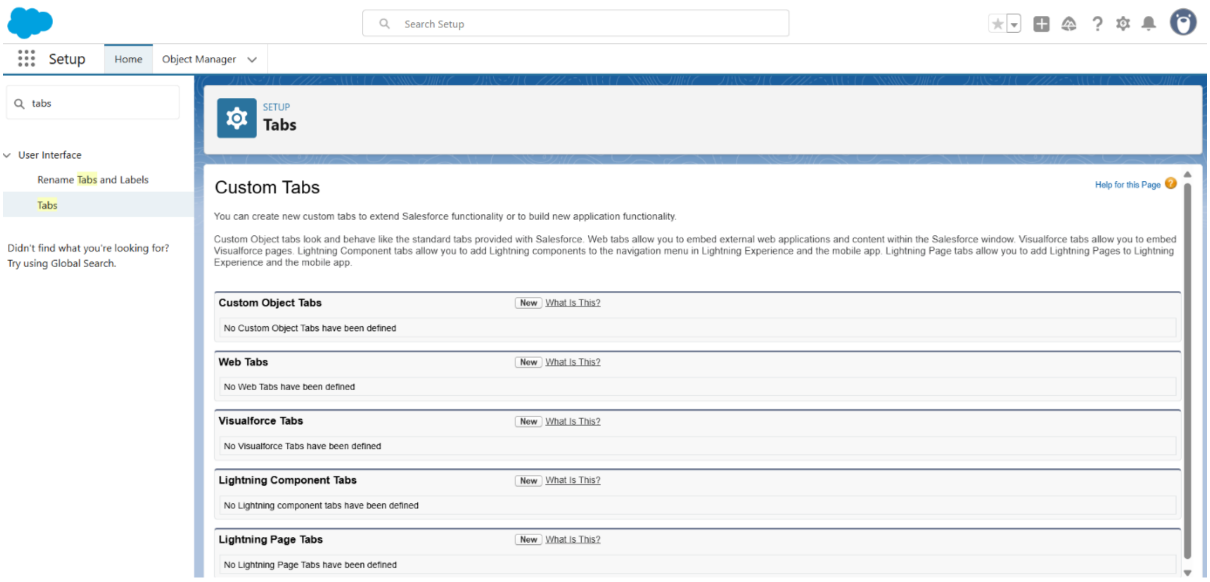
2. Web Tabs Web Tabs are custom tabs that display web content or applications embedded in the salesforce.com window. Web tabs make it easier for your users to quickly access content and applications they frequently use without leaving the salesforce.com application accounts, contacts, and opportunities.

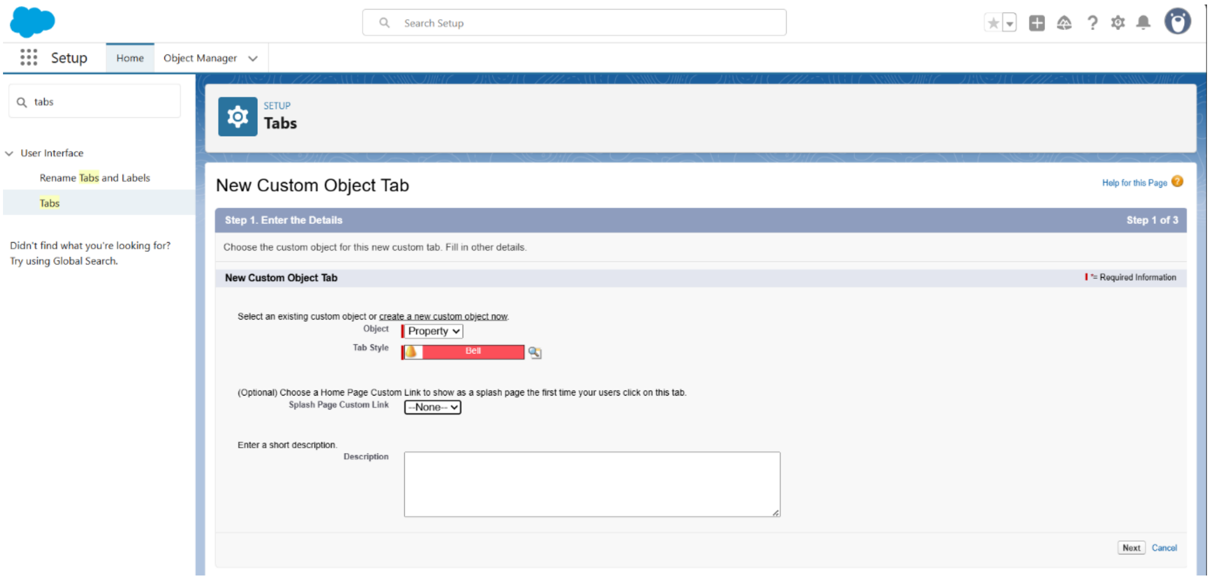
1. Visualforce Tabs
Visualforce Tabs are custom tabs that display a Visualforce page. Visualforco tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.
2 Lightning Component Tabs
Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app
1. Lightning Page-Tabe
Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu

Creating A Custom Tab

To create a Tab:(Property)

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

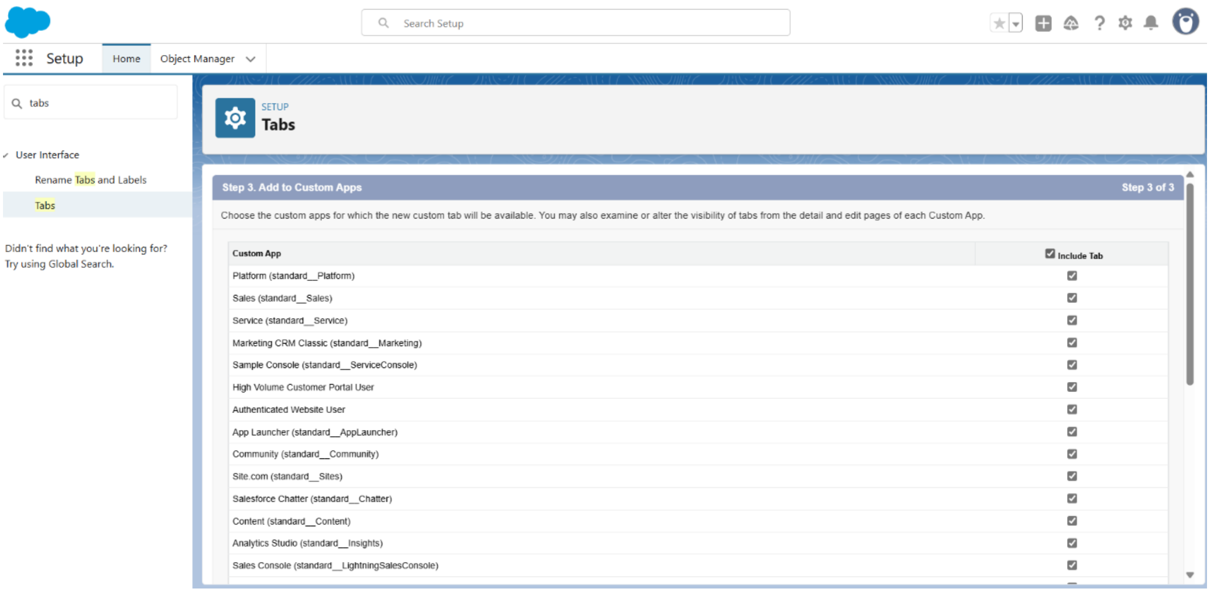


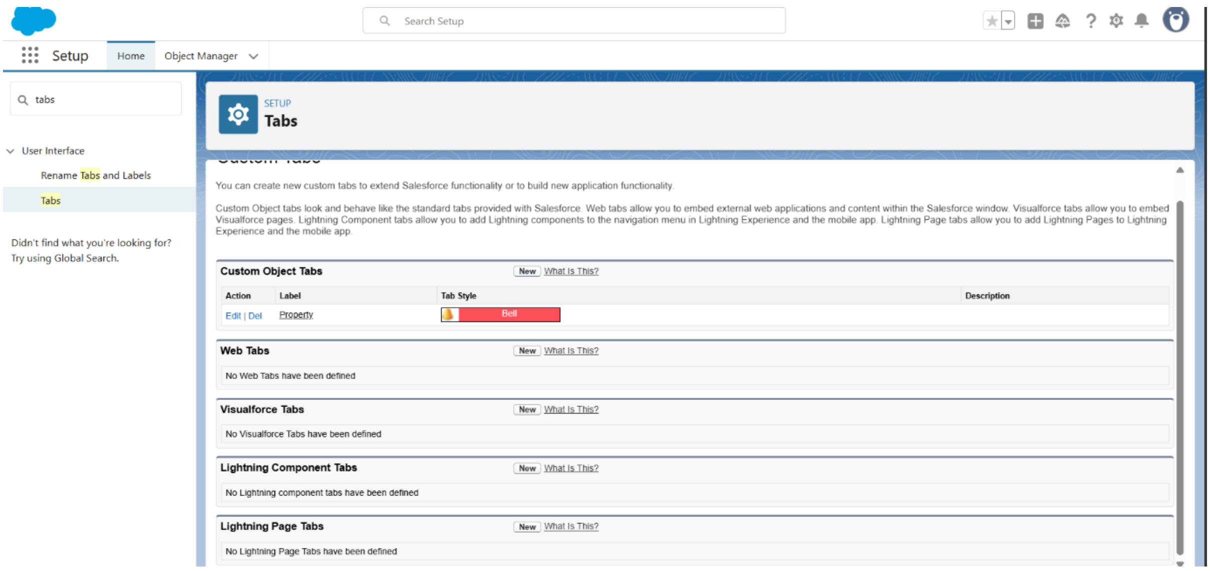


1.Select Object property>> Select the tab style >>> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab

2. Make sure that the Append tab to users' existing personal customizations is checked.

3.Click save



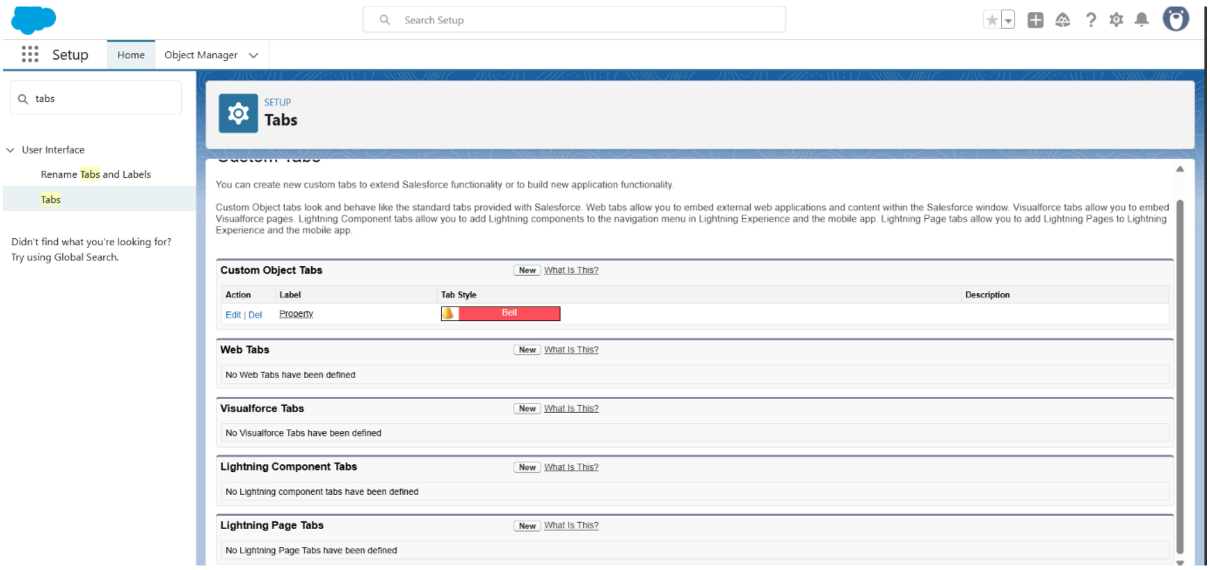


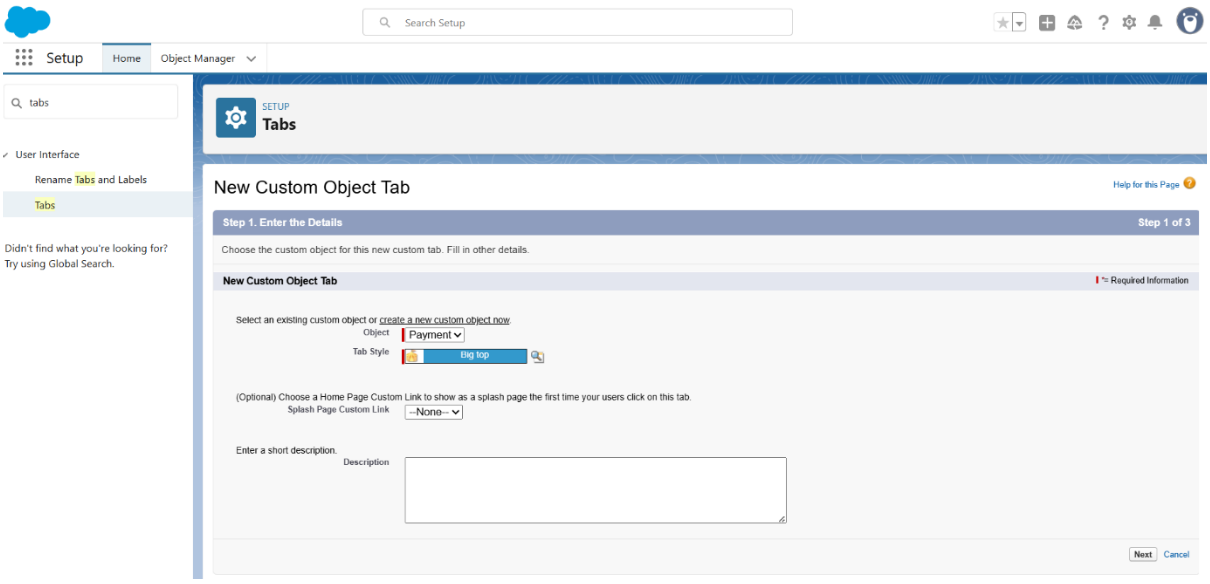
Creating payment for Tenant Tab

1.Now create the tabs for the remaining objects,they are “ payment for tenant.

2. Make sure that the Append tab to users' existing personal customizations is checked.

3. Click save



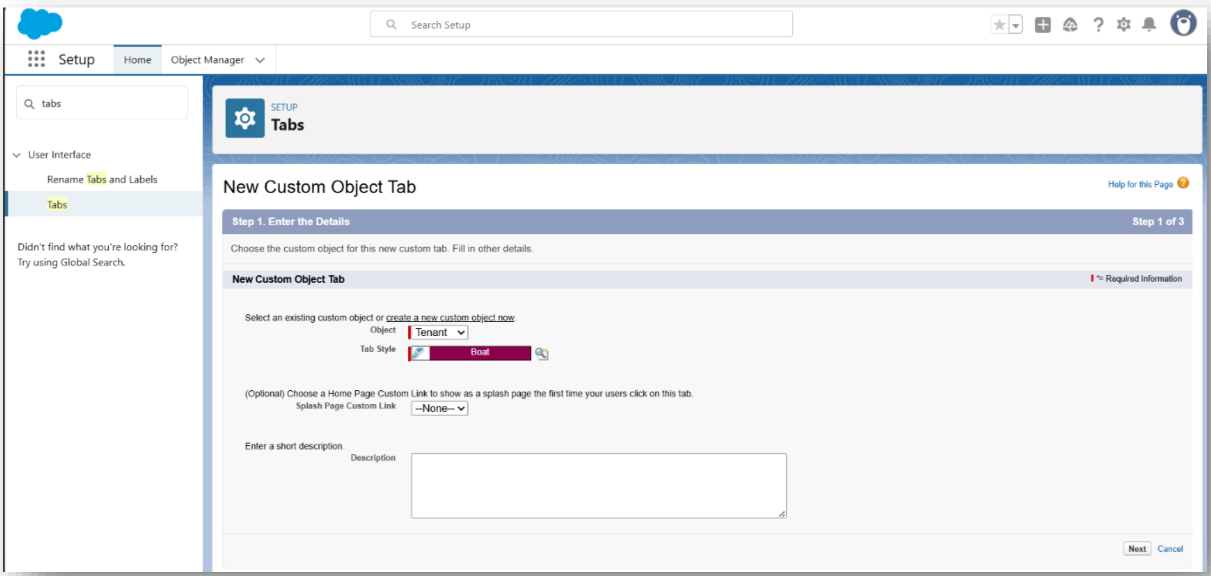


Creating Tenant for Tenant Tab

1.Now create the tabs for the remaining object is Lease .

2. Make sure that the Append tab to users' existing personal customizations is checked.

3. Click save

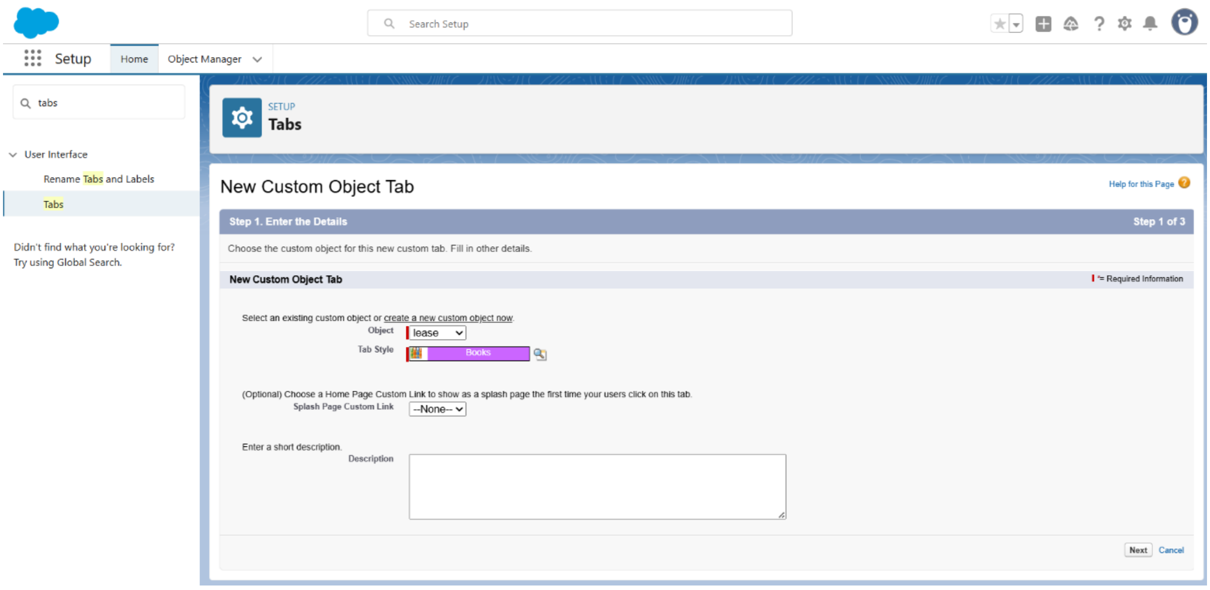


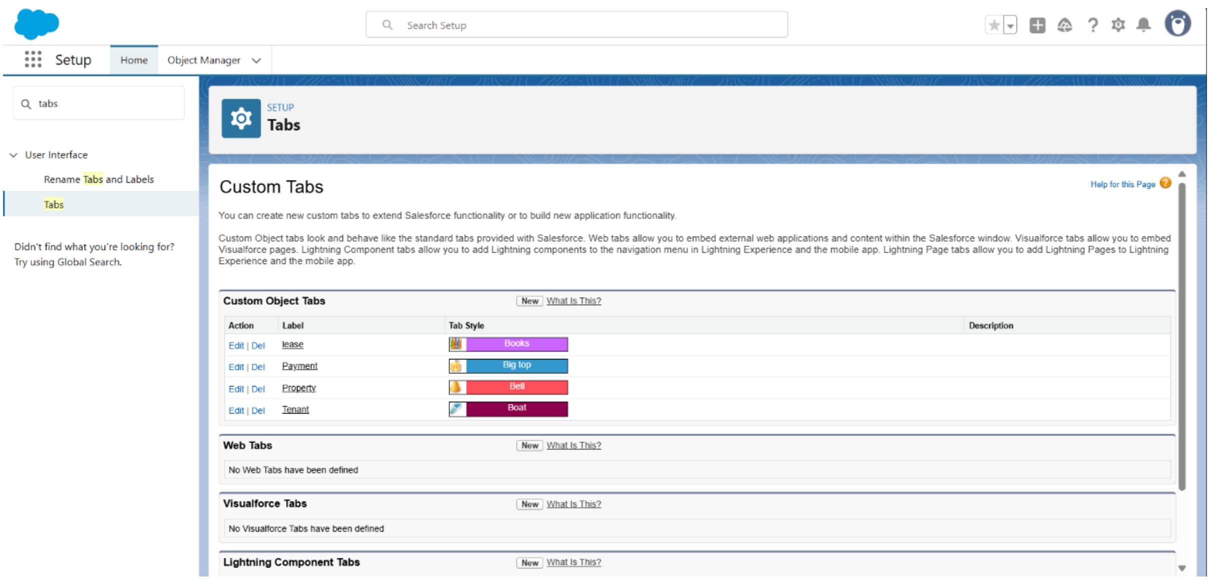
Creating Lease for Tenant Tab

1.Now create the tabs for the remaining object is Lease.

2. Make sure that the Append tab to users' existing personal customizations is checked.

3. Click save





BUSINEES LOGIC

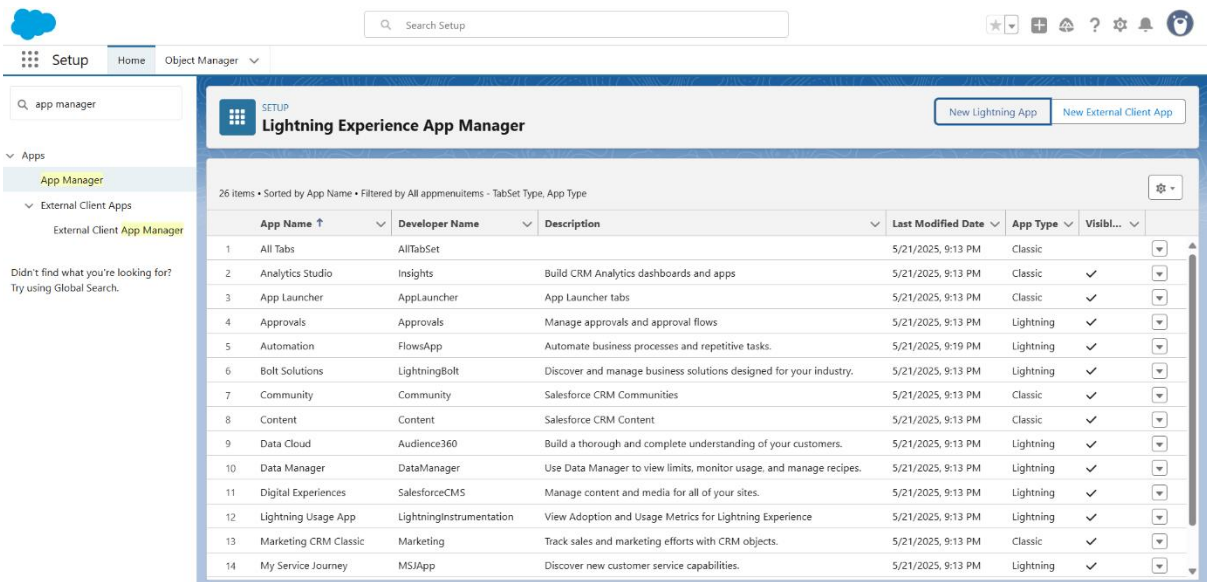
The Lightning App

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 color 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.

Use Case:

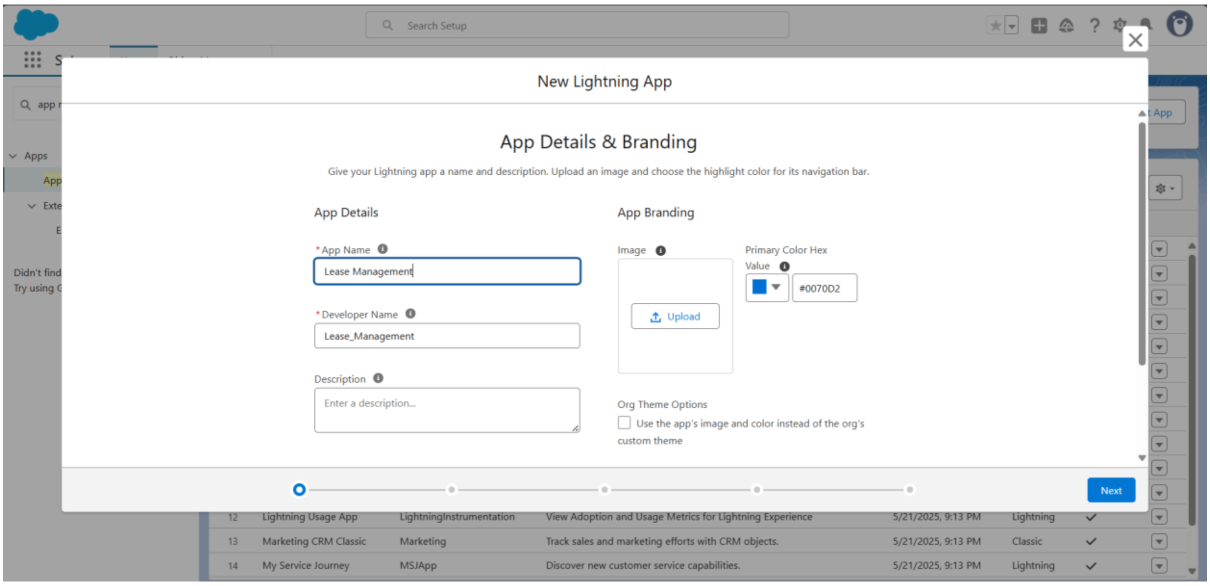
Well done you have reached close to your requirement by creating the objects to store the organisation's data. Making a database for an organisation is just not enough to reach out the requirements, the task is how the users at the organisation can access the objects you have created for them. As an Admin for the organisation it's your duty to make sure every user of the organisation is able to access the data modelling structure.



Create A Lightning App

To create a lightning app page

1. Go to setup page >> search "app manager in quick find >> select app manager >> click on New lighting App.



2. Fill the app name in app details and branding as follow

App Name: Lease Management

Developer Namee: This will auto populated

Image: optional (if you want to give any image you can otherwise not mandatory)
Primary colour hex value: keep this default

3. Then click Next >> (App option page Set Navigation Style as Standard Navigation >> Ne

(Utility Items) keep it as default >> Next.

5. To Add Navigation Items:

Search for the item in the (Payment for tenant, Tenants, property, lease) from the search bar and move it using the arrow button? Next? Next.

6. To Add User Profiles.

UI CONFIGURATION

Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

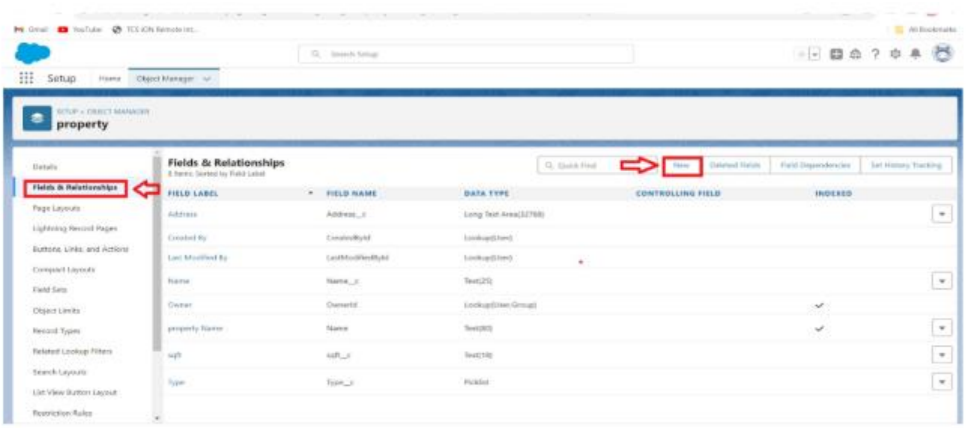
Types of Fields

1. Standard Fields
2. Custom Fields

Standard Fields:
As the riame suggests, the Standard Fields are the predefined fields in Salesforce that perform a standarn task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salestorce application. They are,
>>Created By
>>Owner
>>Last Modified
>>Ficid Made During object Creation

Custom Fields:

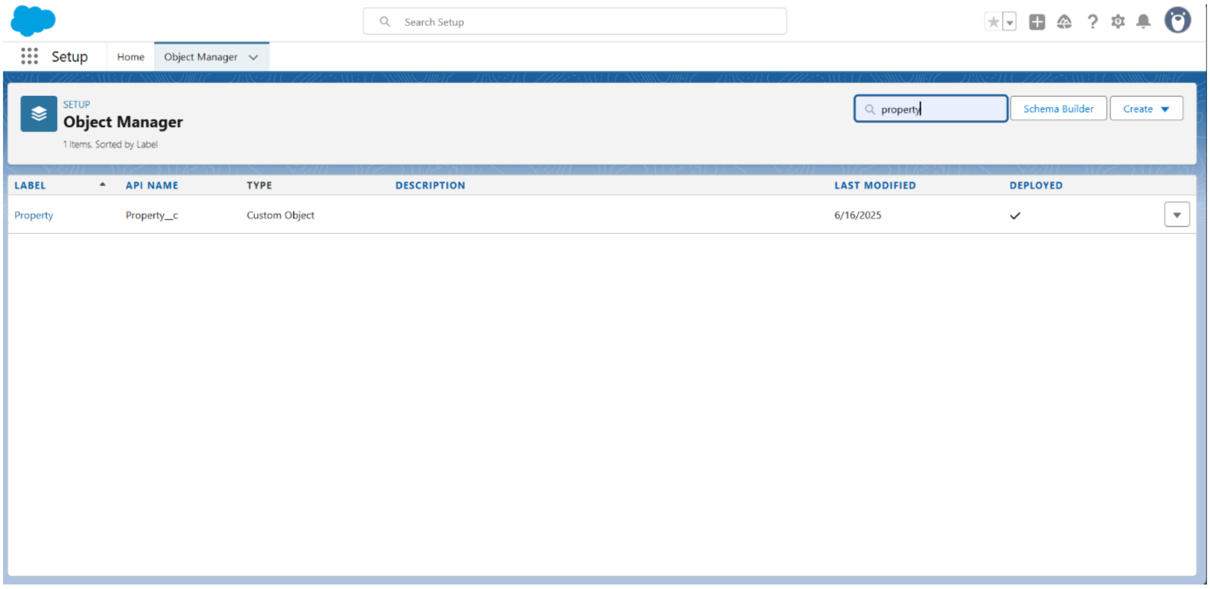
On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, cach organizer or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.


Setup
property
Fields & Relationships
FIELD LABEL
FIELD NAME
DATA TYPE
CONTROLLING FIELD
INDEXED

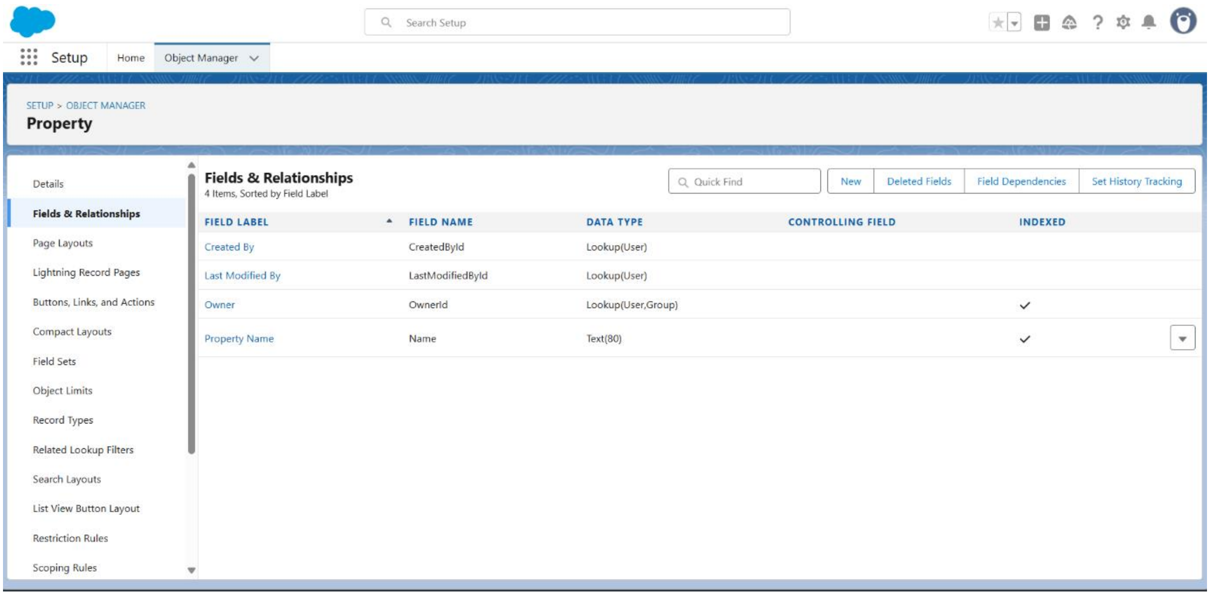
Creation Of Fields For The Property Object

To create fields in an object:

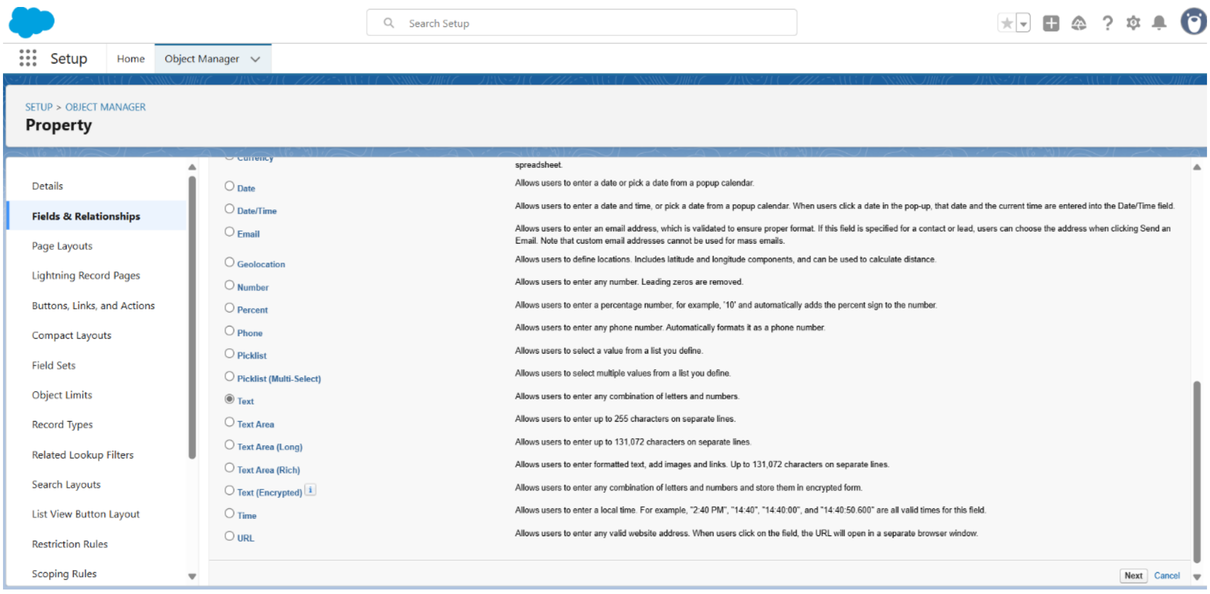
1. Go to setup >>click on Object Manager>>type object name(property) in search bar >>click on the object



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

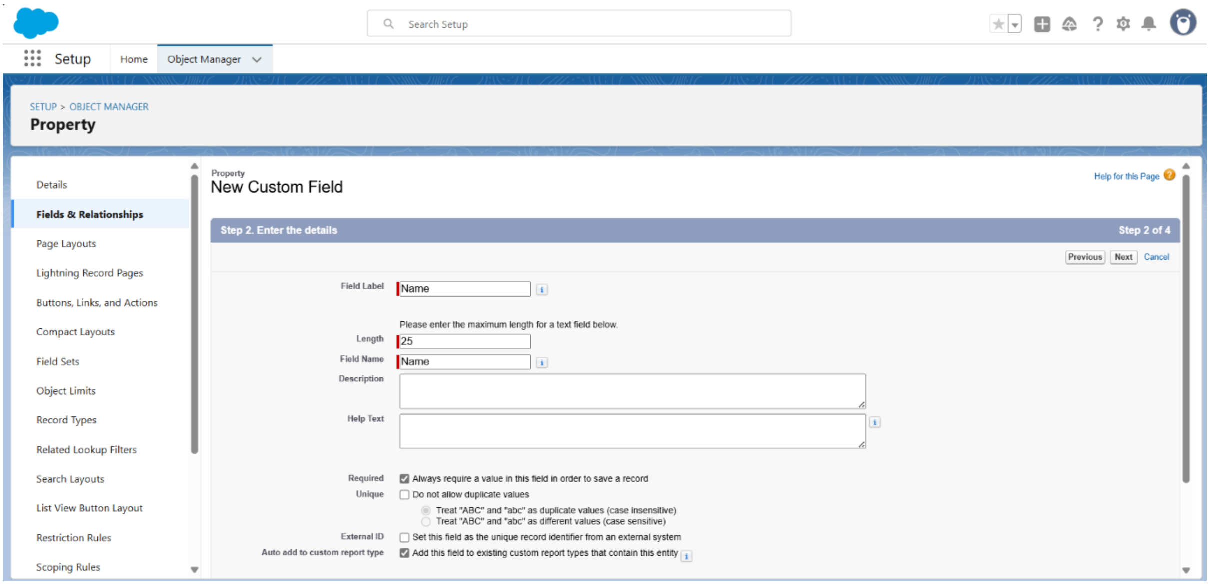


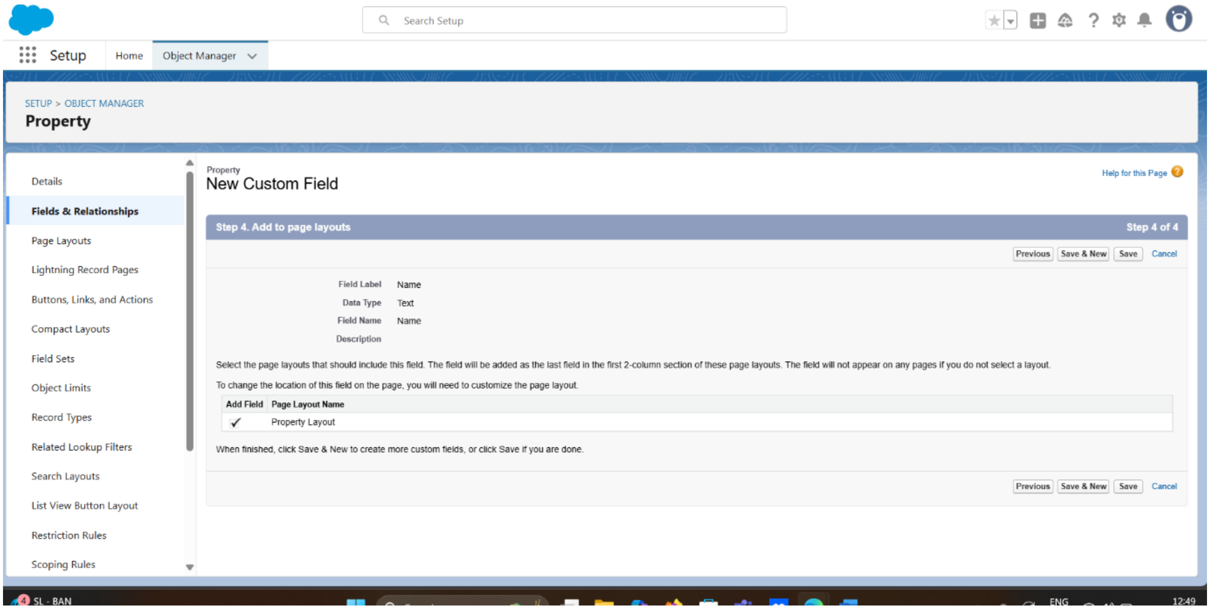
3.Select Data Type as “Text”



4.Click on next

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

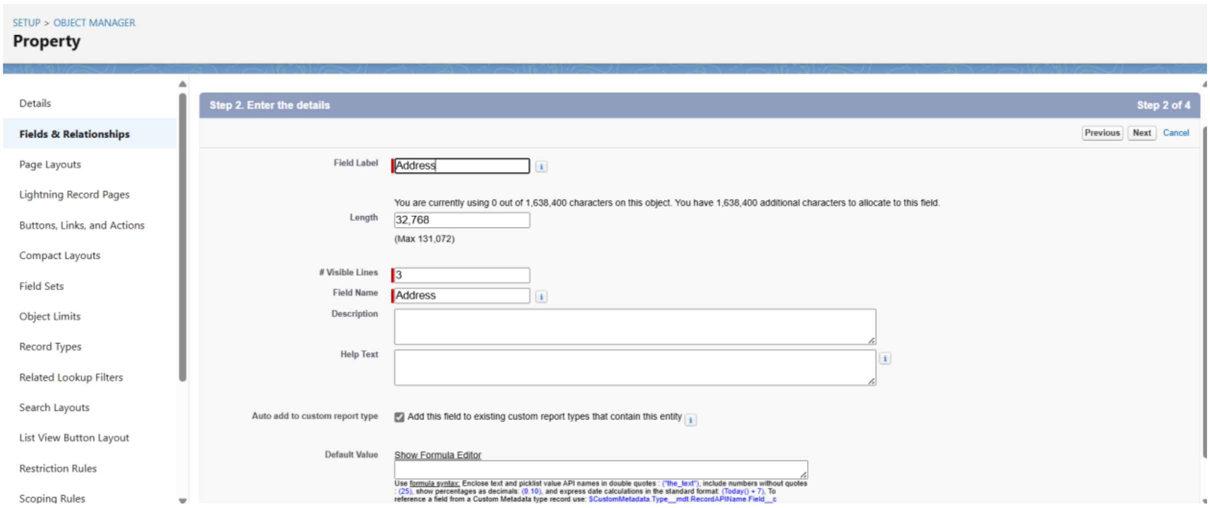


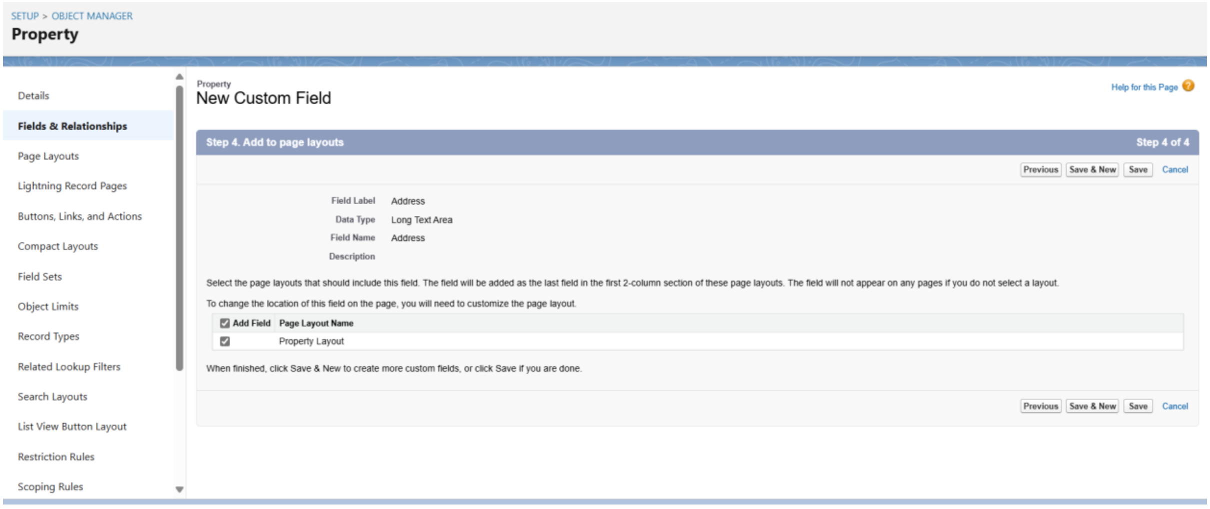


2. To create another fields in an object:

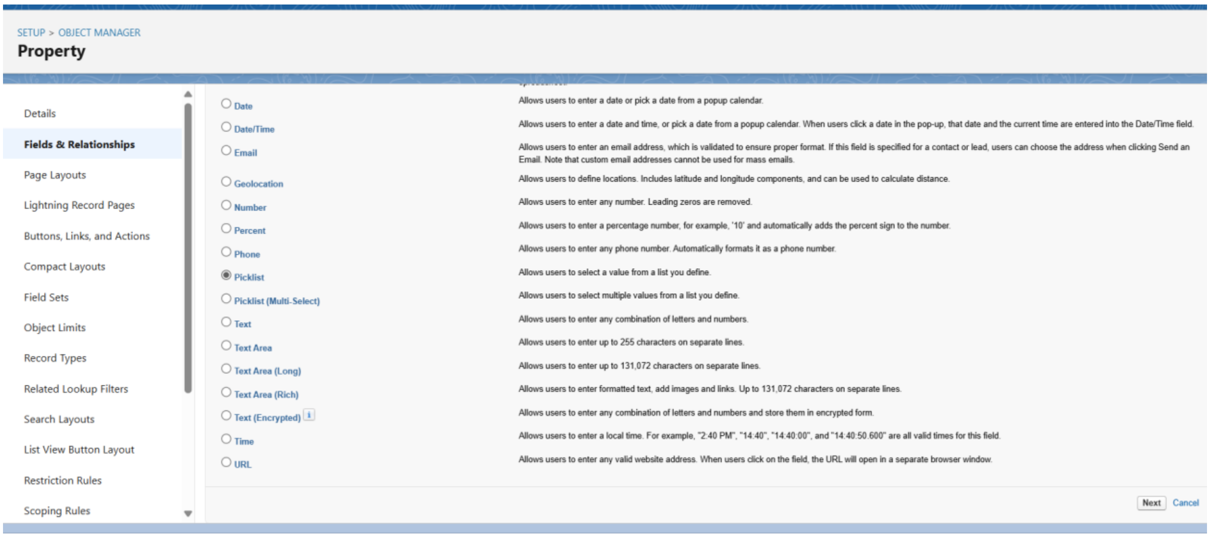
1. Go to setup >> click on Object Manager >>type object name (property) in search bar click on the object
2. Now click on "Fields & Relationships">>New
3. Select Data type as a "Long Text" and Click on Next
4. Fill the Above as following:

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





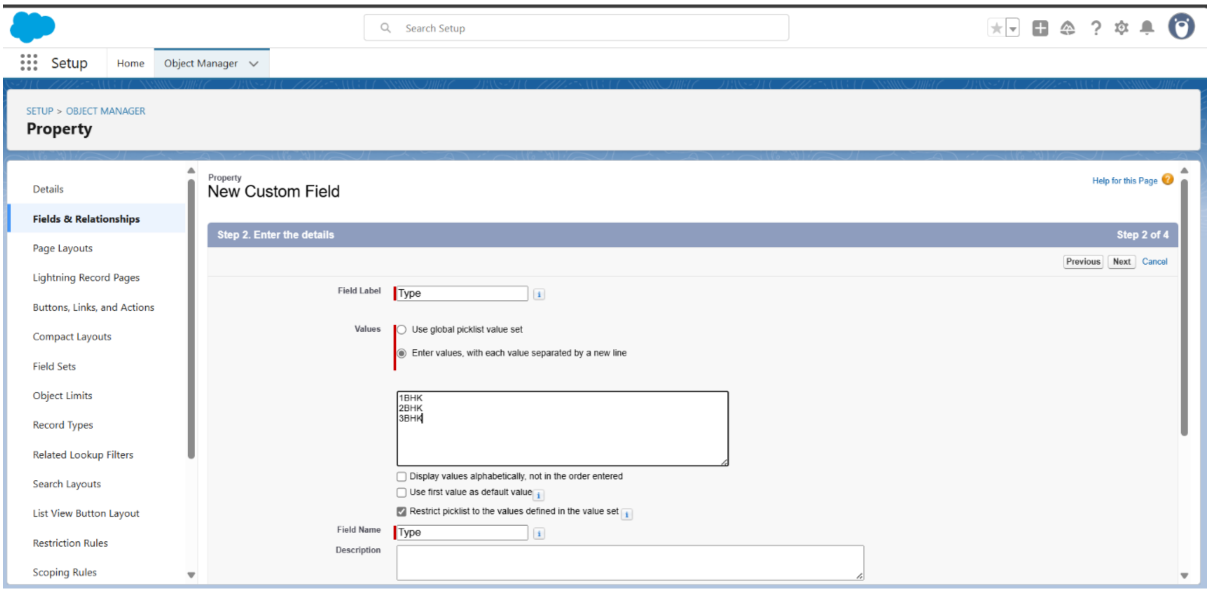
- To create another fields in an object:
  5. Go to setup >> click on Object Manager type object name(property) in search bar click on the object. 6.
  Now click on "Fields & Relationships New

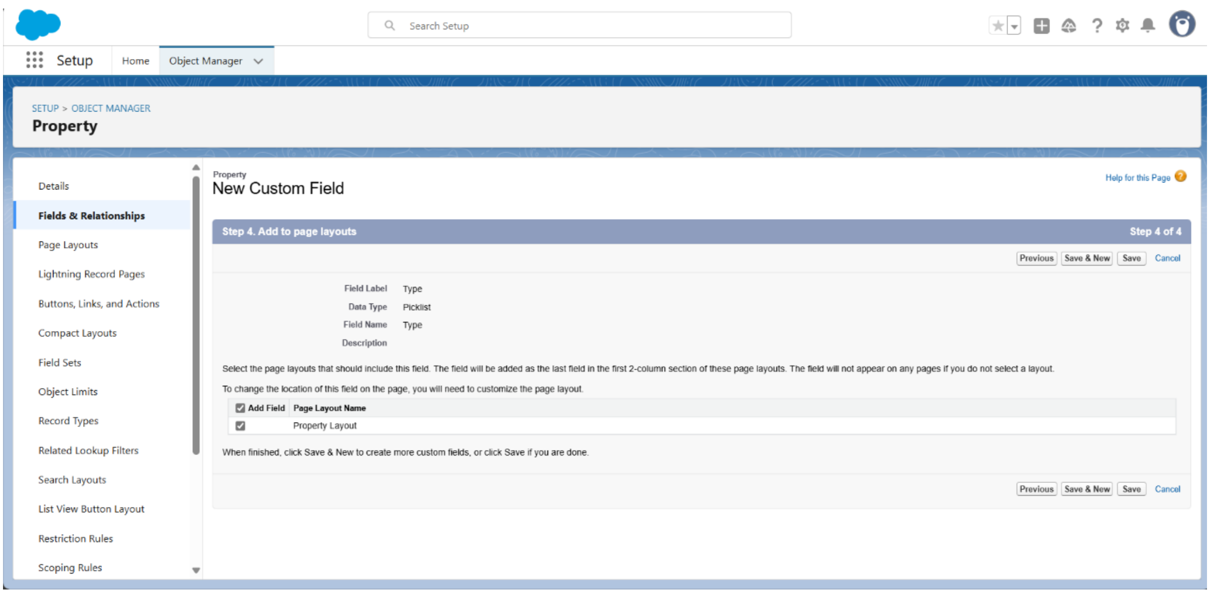


7. Select Data type as a "picklist and Click on Next

8. Fill the Above as following

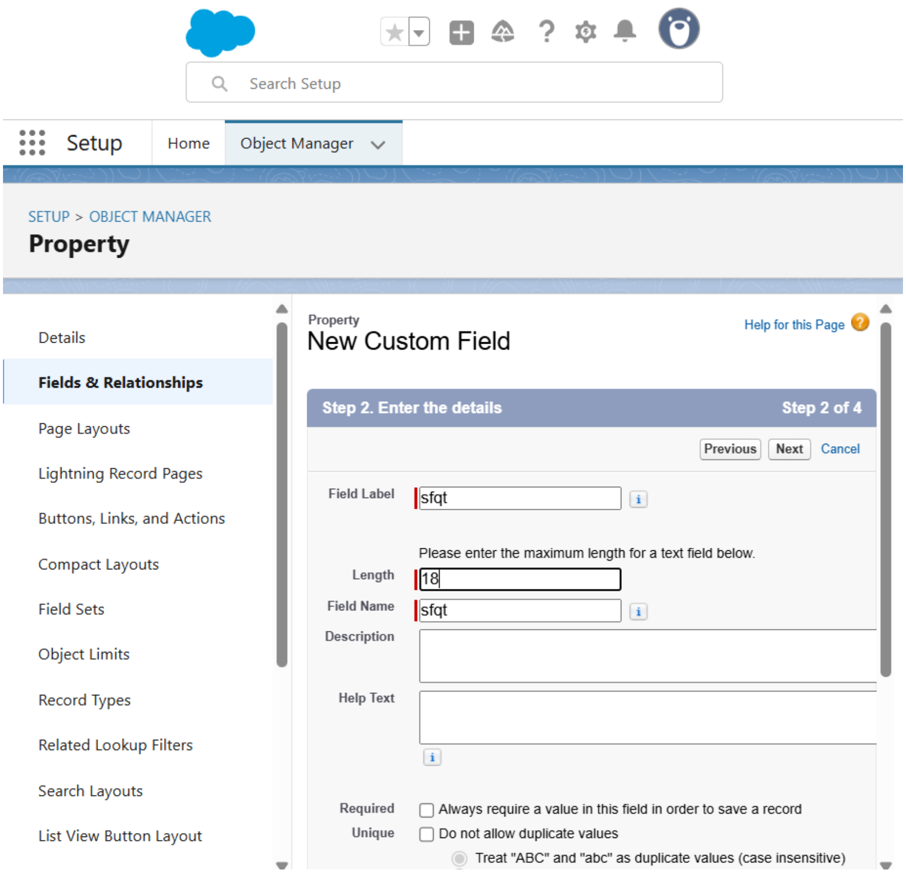
- Field Label: Type
- Field Name gets auto generated
- Enter values, with each value separated by a new line
- Enter these values
  1BHK
  2BHK
  3BHK
- Click on Next >> Next >> Save and new.

- 

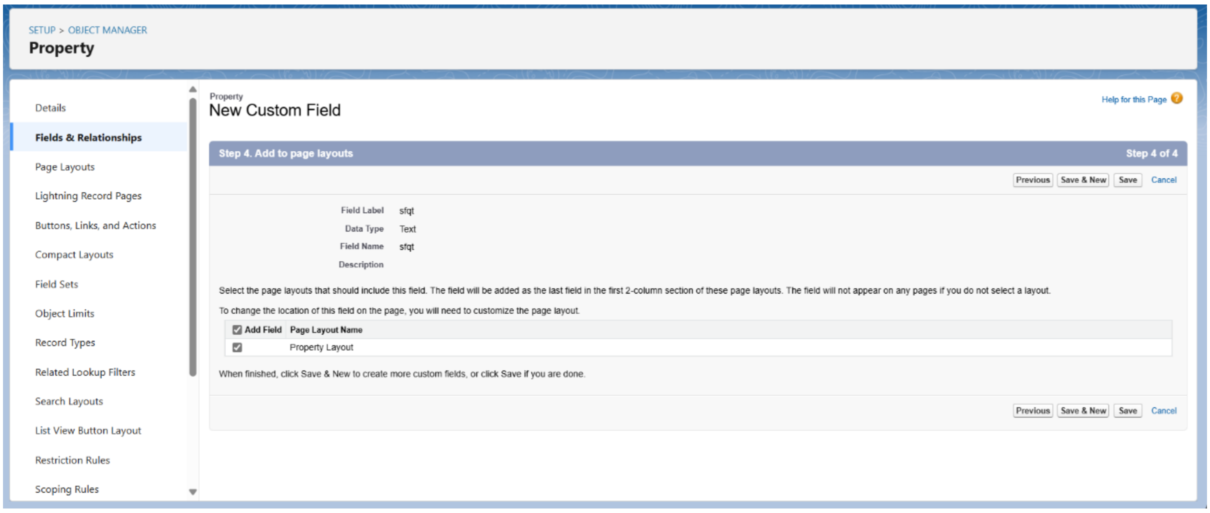
- 

To create another fields in an object:

9. Co 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

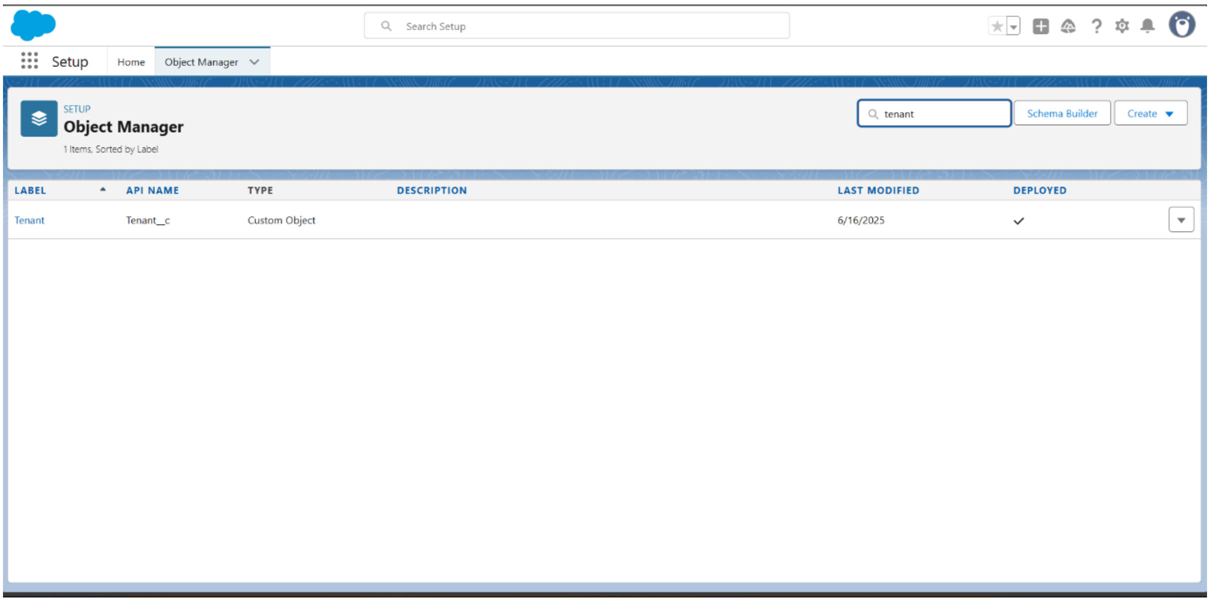
- 

12. Fill the Above as following:
Field Name: sqft
Field Name: gets auto generate

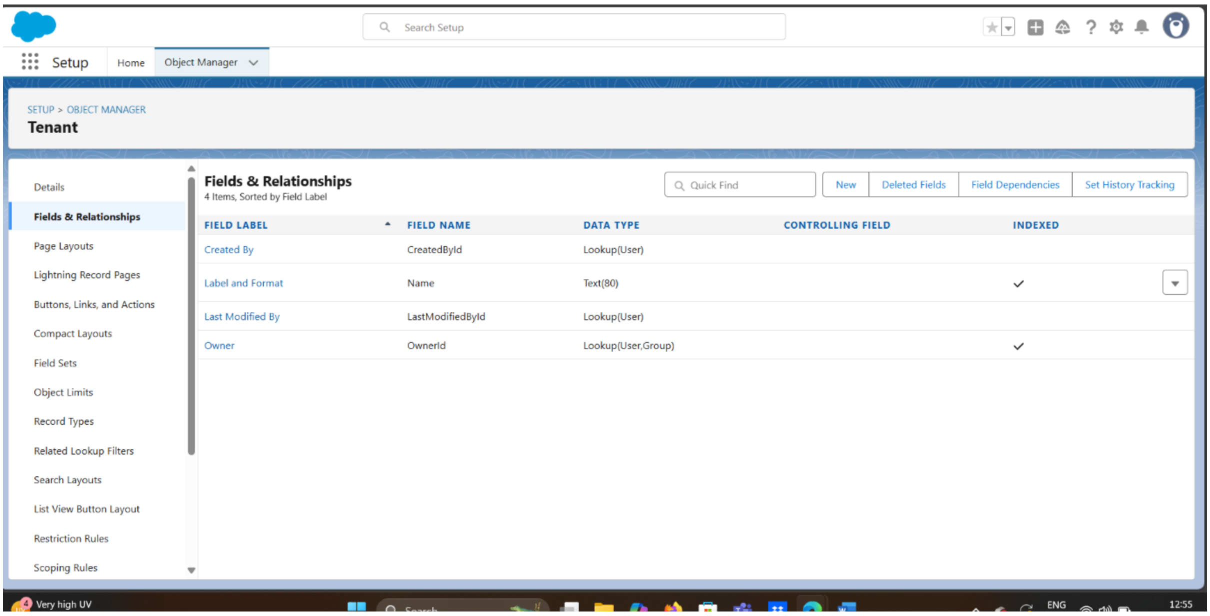
- Length: 18
- Click on Next >> Next >> Save
- 

Creation Of Fields For The Tenant Object

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

- 

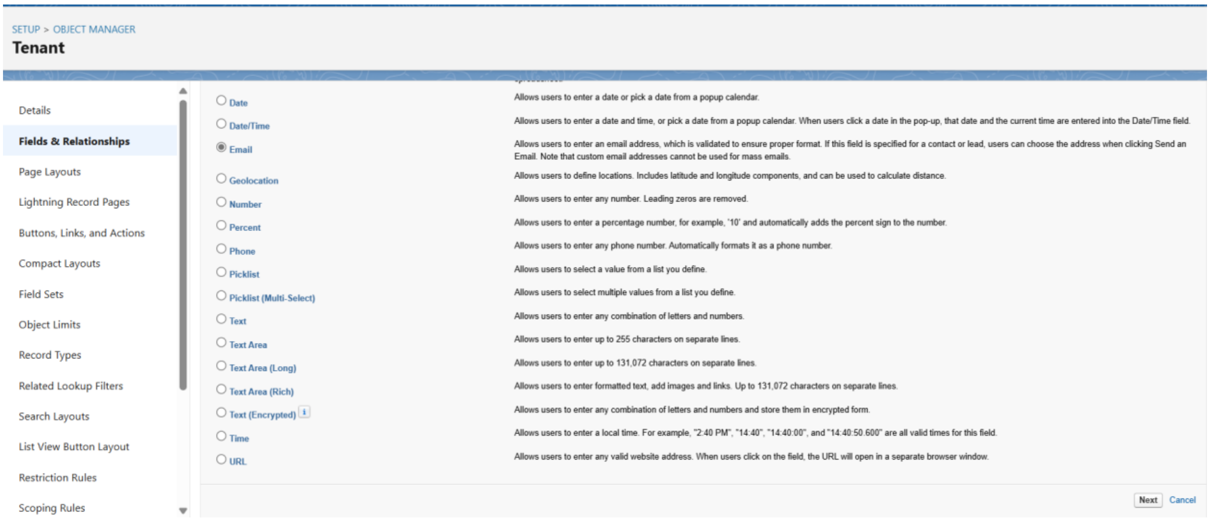
2. Now click on "Fields & Relationships" >>> New

- 

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

4. Fill the Above as following:

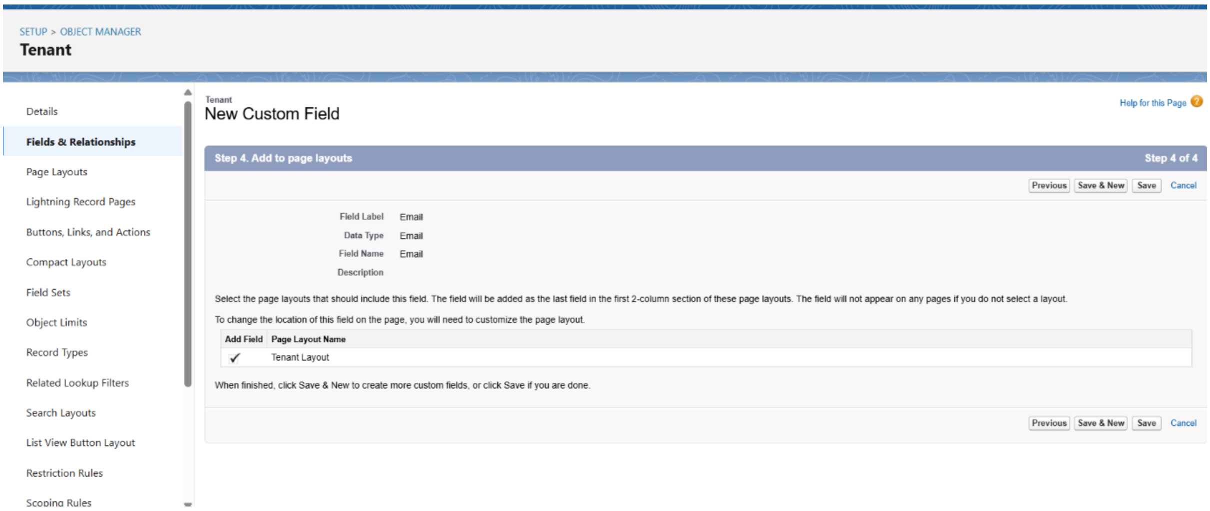
Field Label: Email



Field Name: gets auto generated

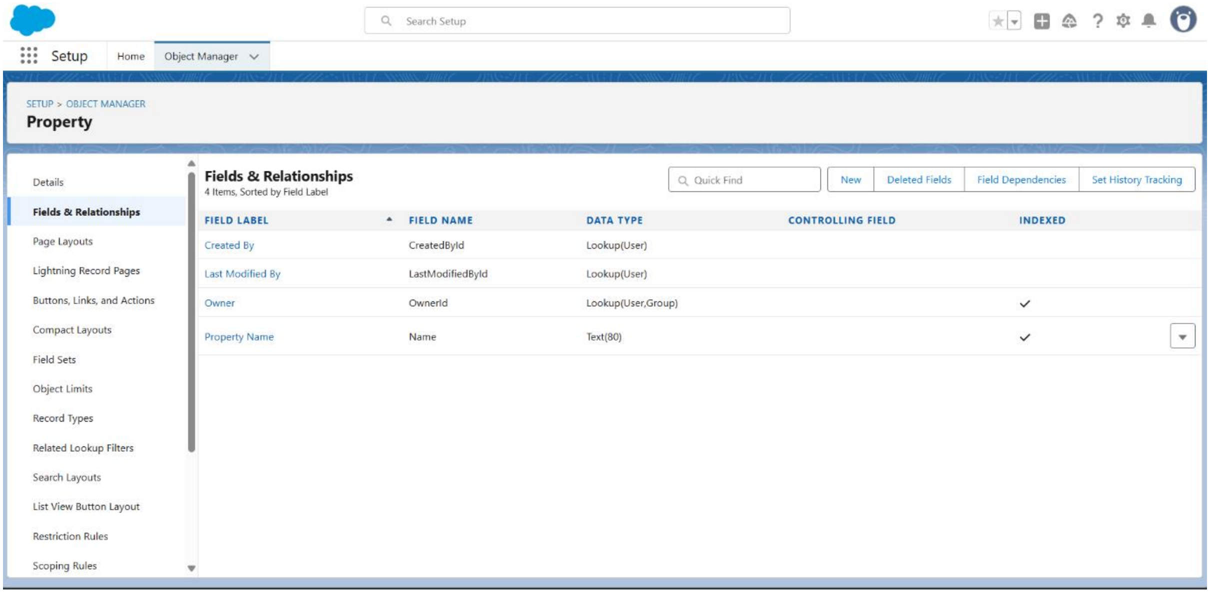
Click on required check box

Click on Next Next>> Save and new.

- 

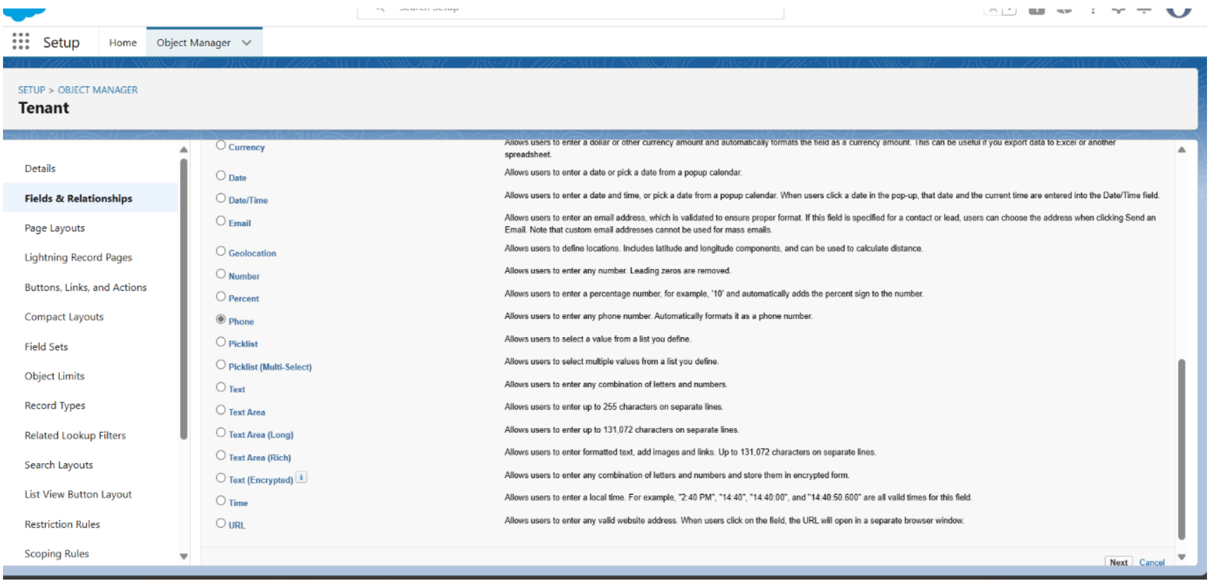
To create another fields in an object:

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

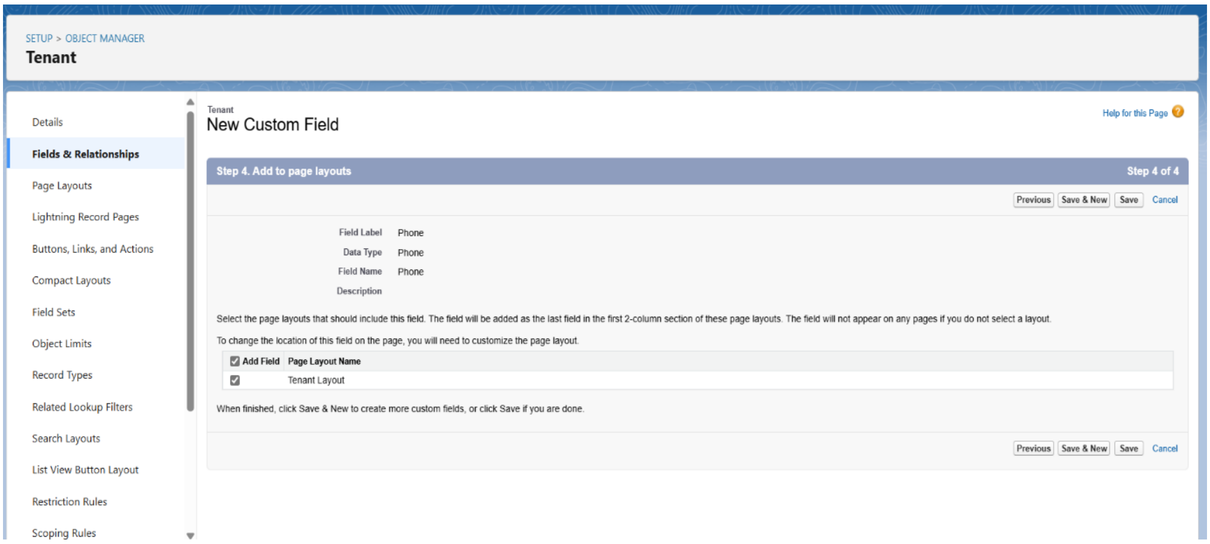


2.Now click an "Fields & Relationships"> New

3. Select Data type as a "phone" and Click on New

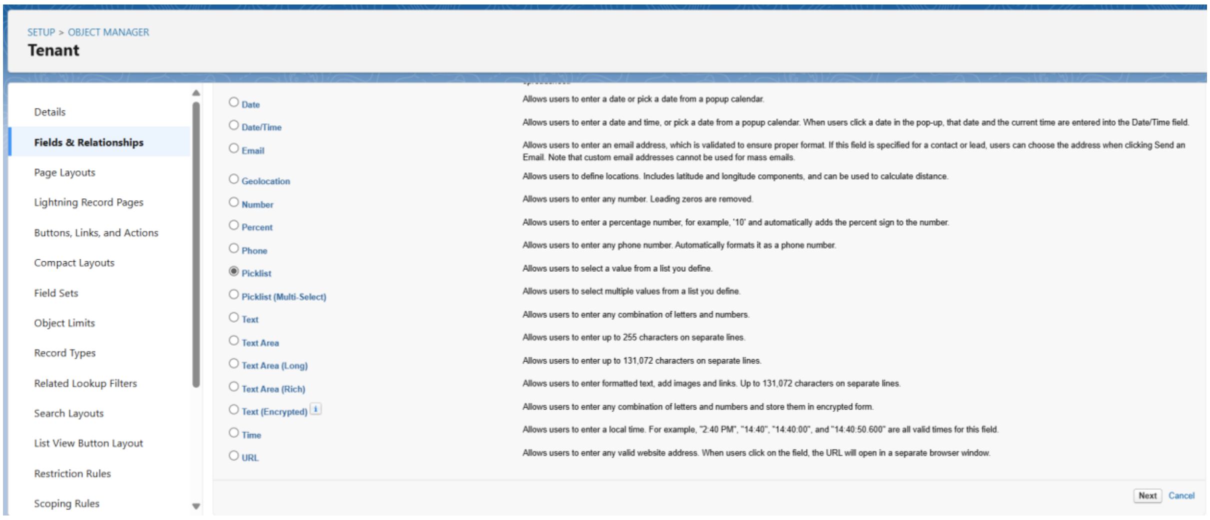


4. Fill the Above as following:
Field Label: Phone
Field Name: gets auto generated
Click on Next >> Next >> Save and new



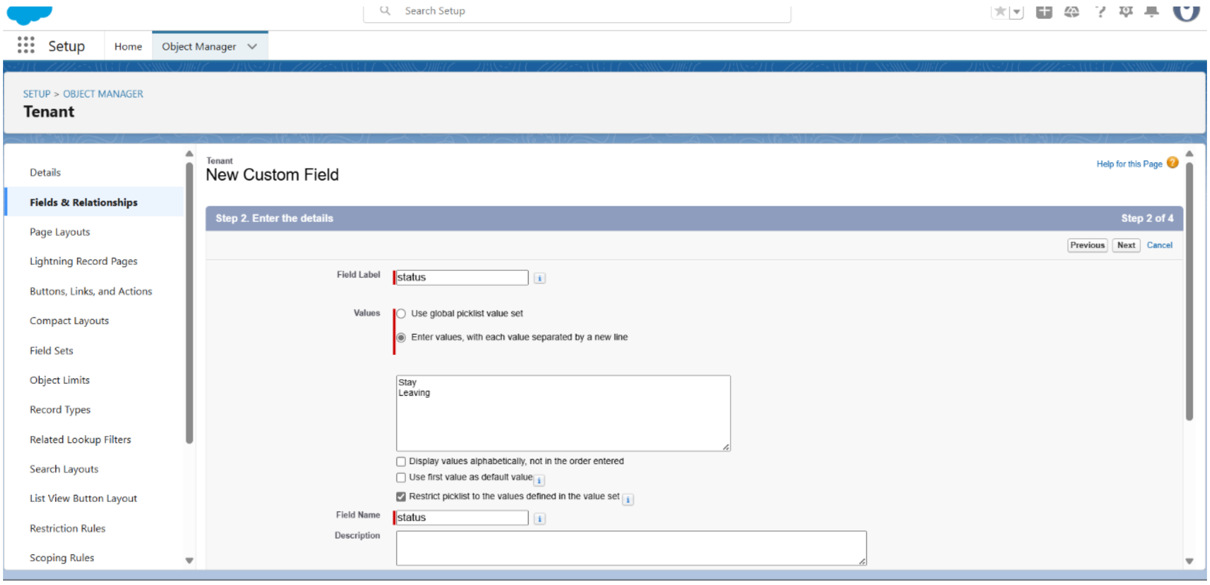
To create another fields in an object:

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

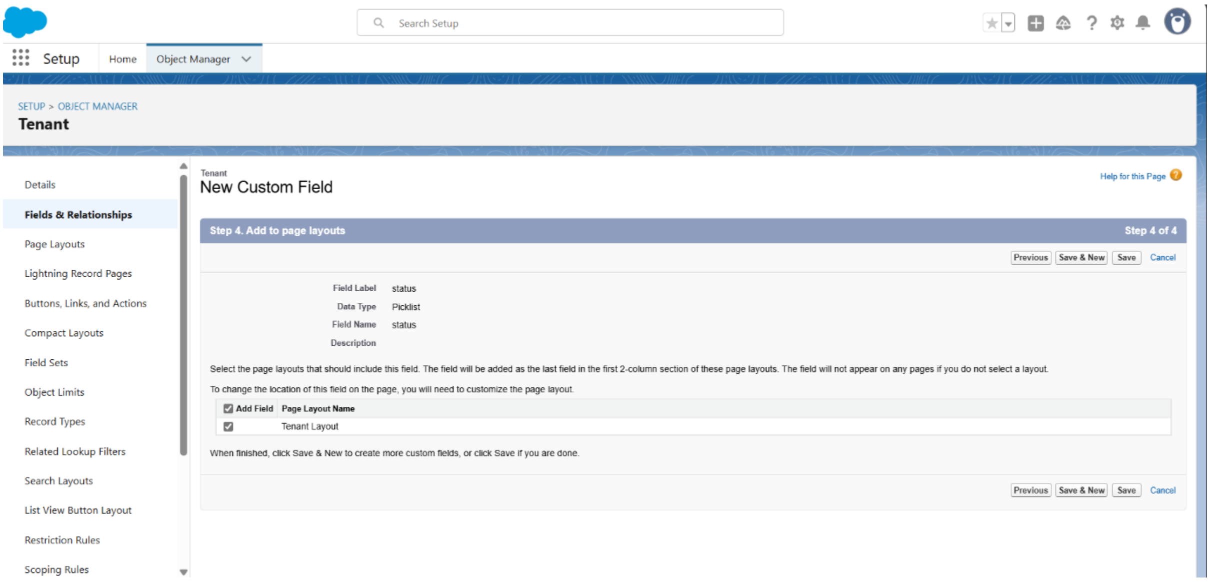


8. Fill the Above as following:

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

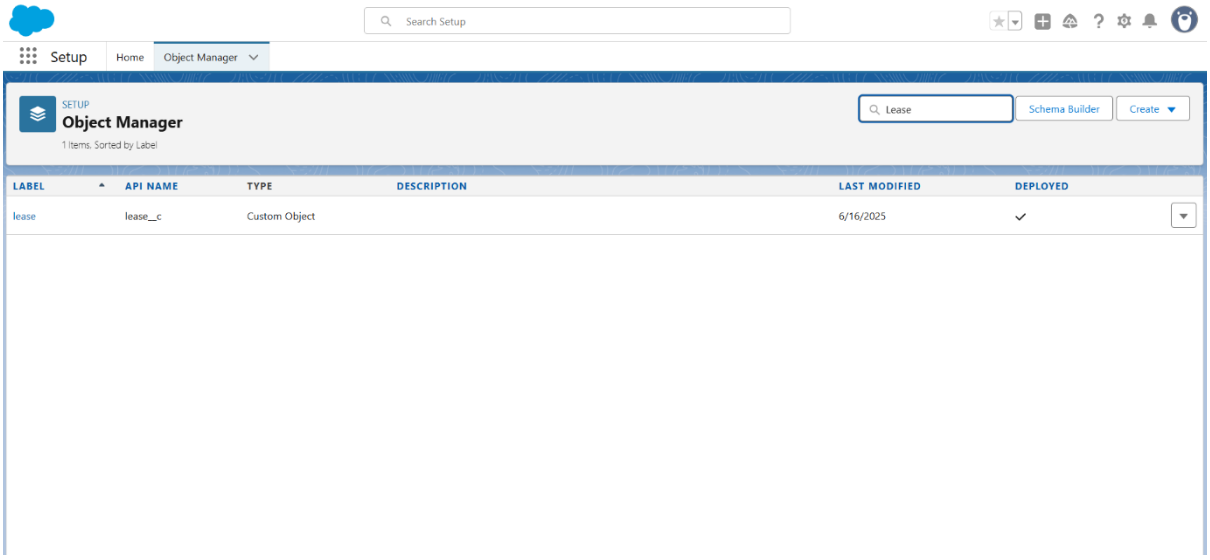


- Click on Next >> Next >> Save

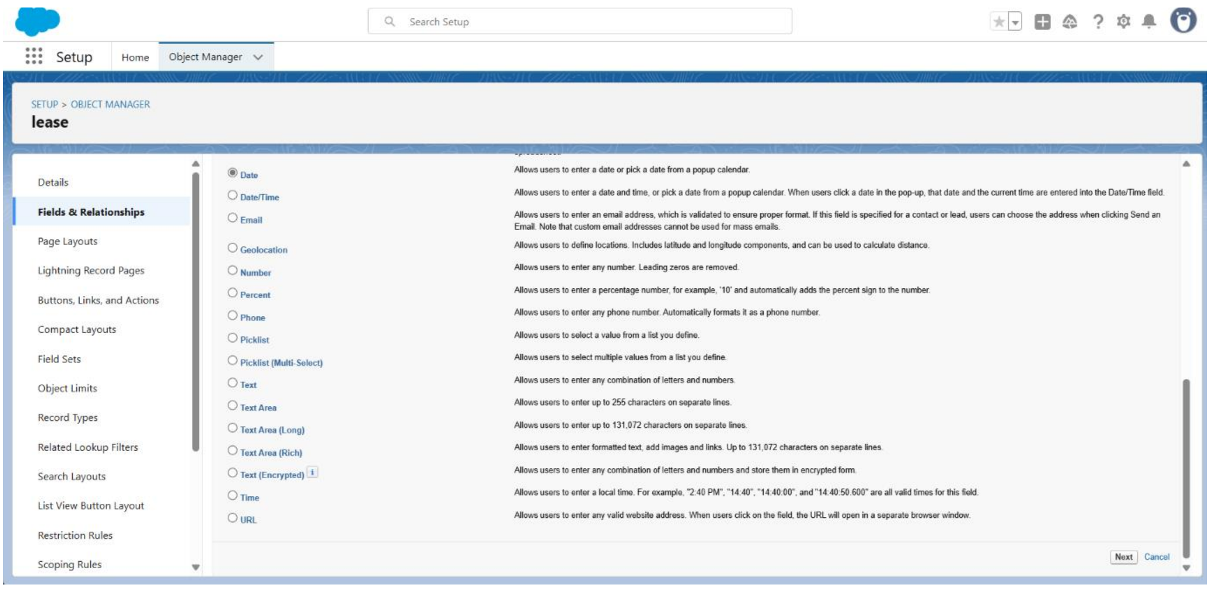


Creation Of Fields For The Lease Object

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

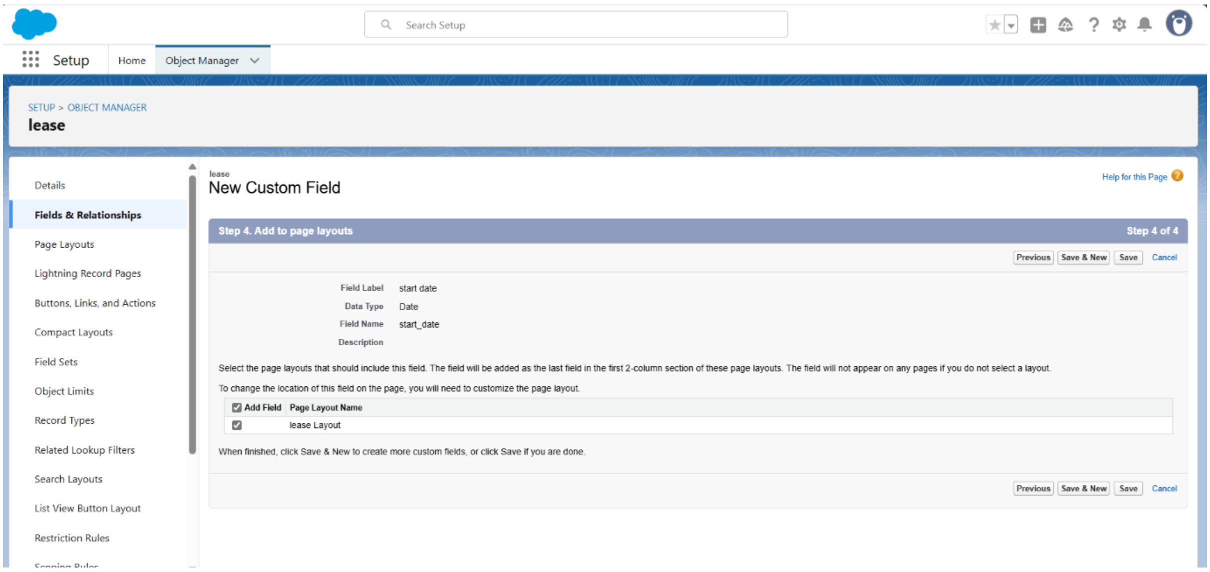
- 

2. Now click on "Fields & Relationships"> New
3. Select Data type as a "Date" and Click on Next

- 

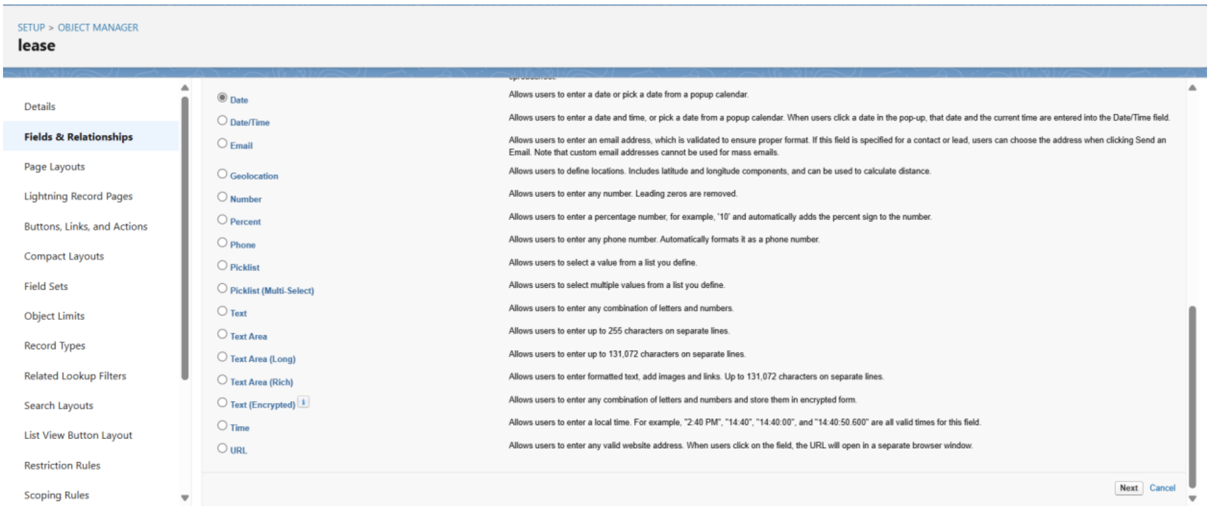
4. Fill the Above as following

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

•


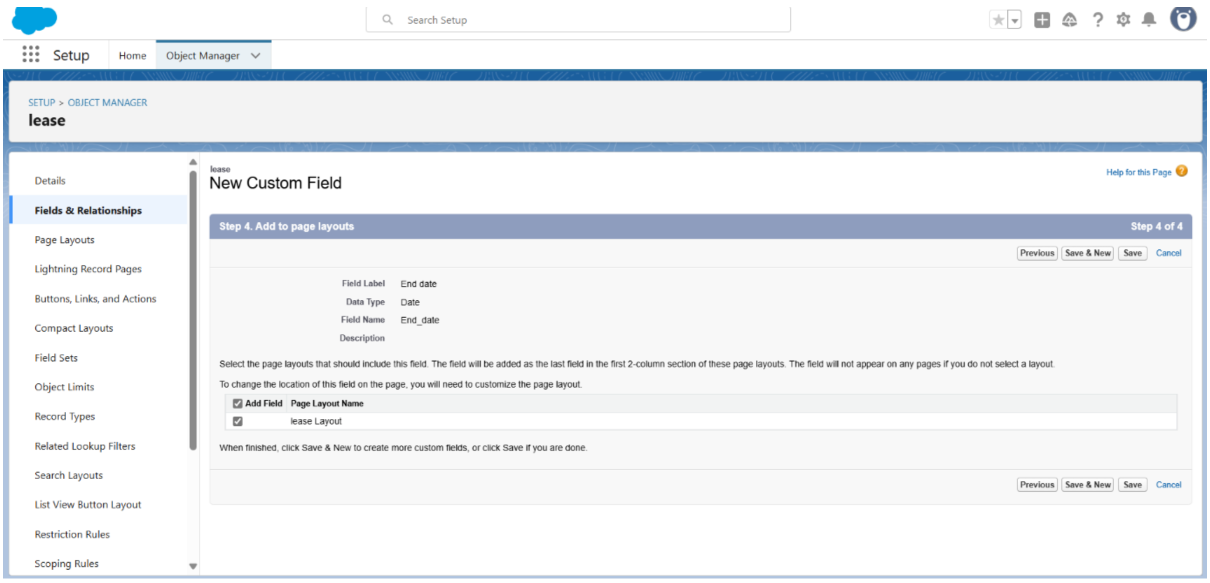
To create another fields in an object:

1.Go to setup click on Object Manager type object name (Lease) in search bar click on the object.
2. how click on 'Fields & Relationships New
3. Select Data type as a "Date" and Click on Next

•


4.Fill in the Above as following:

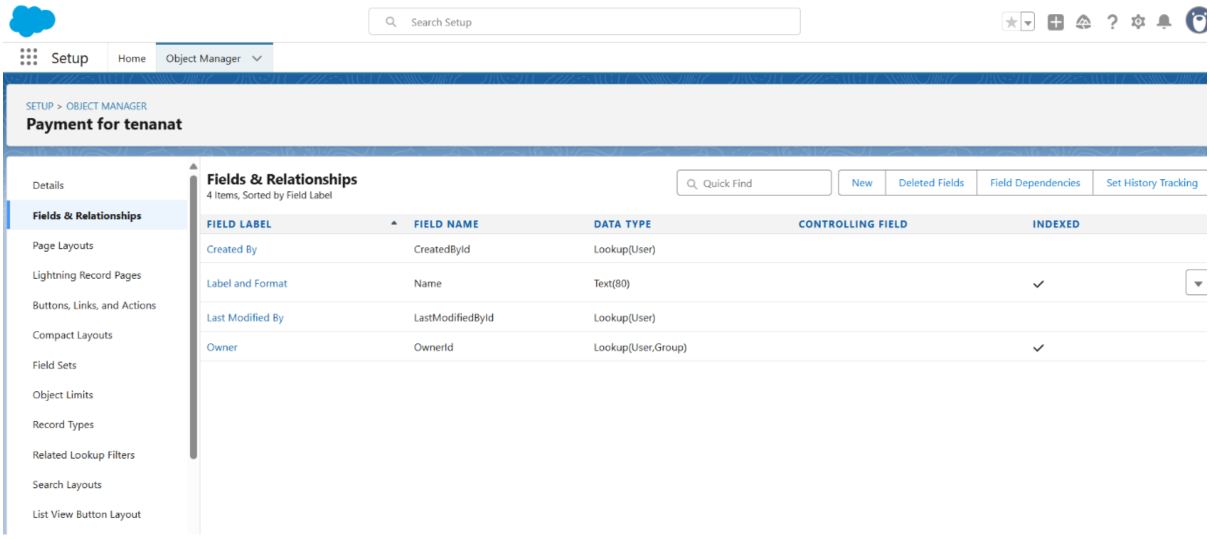
- Field Label: End date
- Field Name: gets auto generated
- Click on Next >> Next Save >> and new

- 

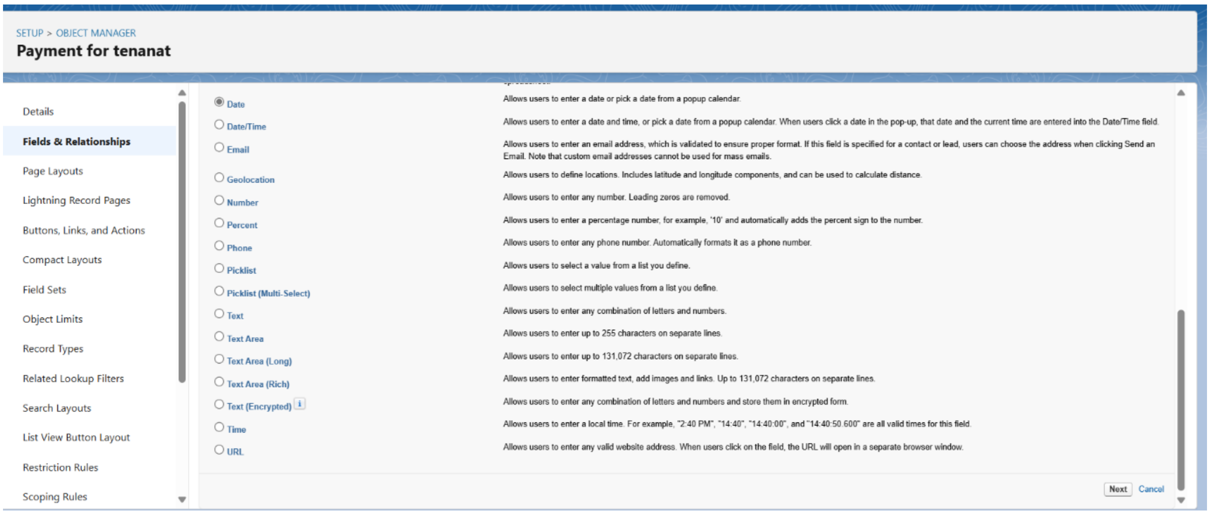
Creation Of Fields For The Payment For Tenant Object

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

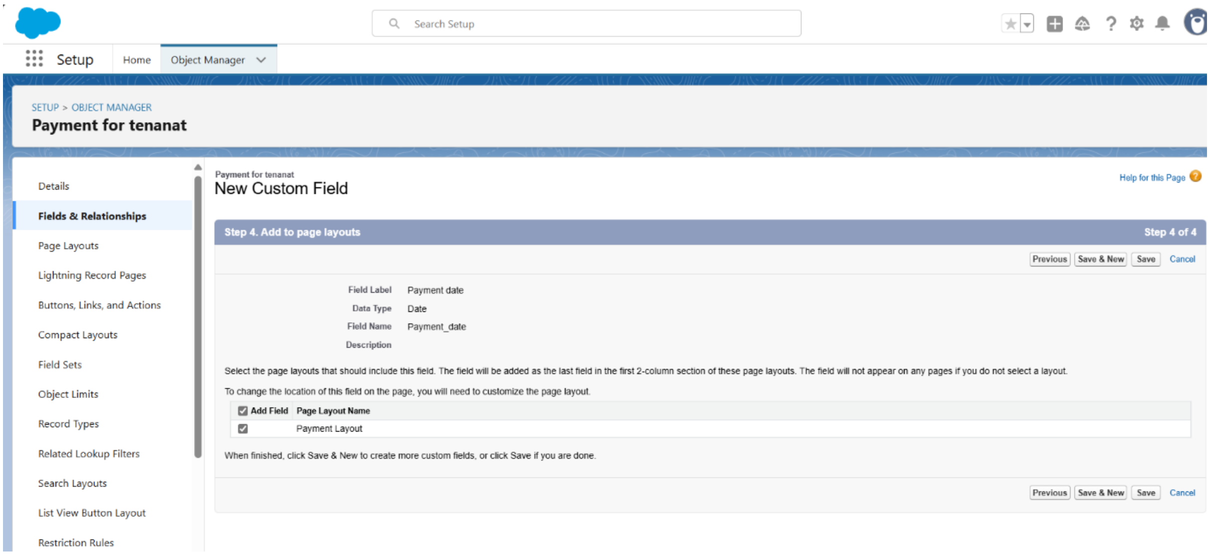
2. Now click on "Fields & Relationships" >>> New

- 

2. Select Data type as a "Date" and Click on Next

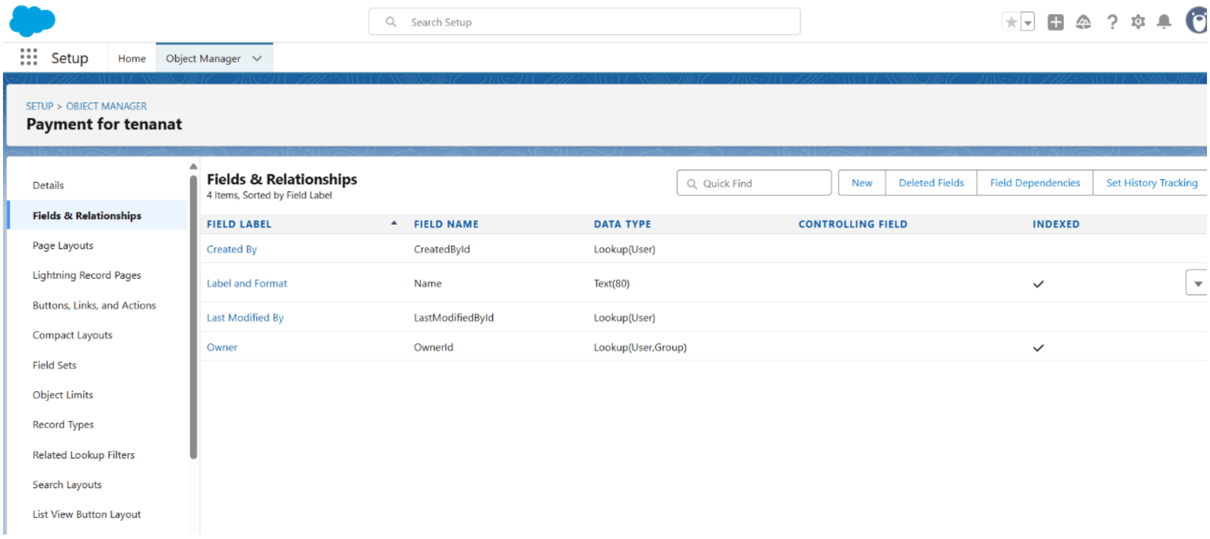


4. Fill the Above as following

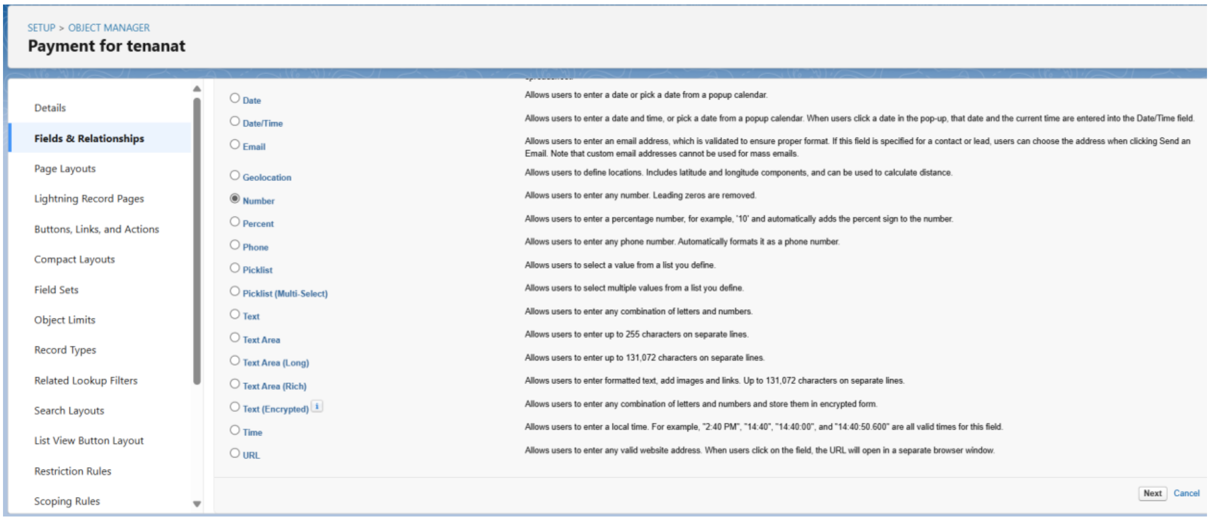
- Field Label: Payment date
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new
- 

To create another fields in an object:

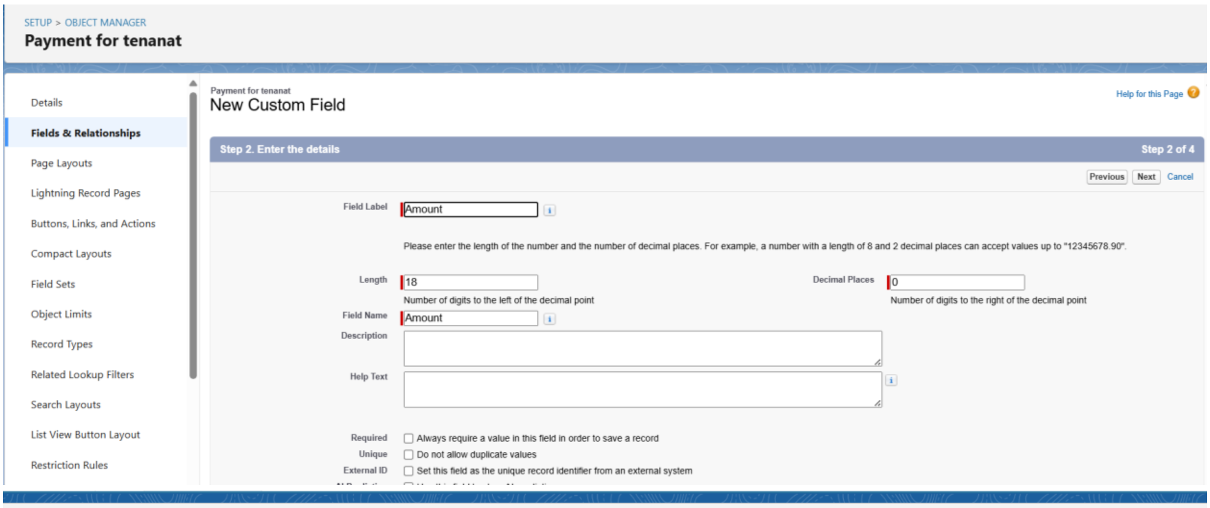
1.Go to setup click on Object Manager >> type object name (Payment for tenant!) in search bar click on the object
2. Now click on "Fields & Relationships>> New

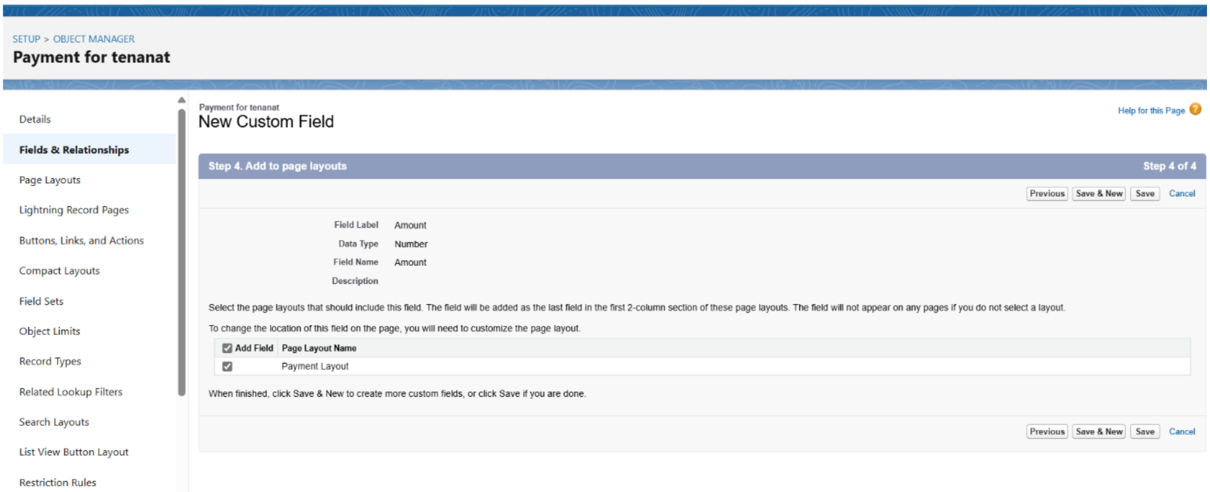


3. Select Data type as a "Number and Click on Next
4. Fill the Above as following

- 

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

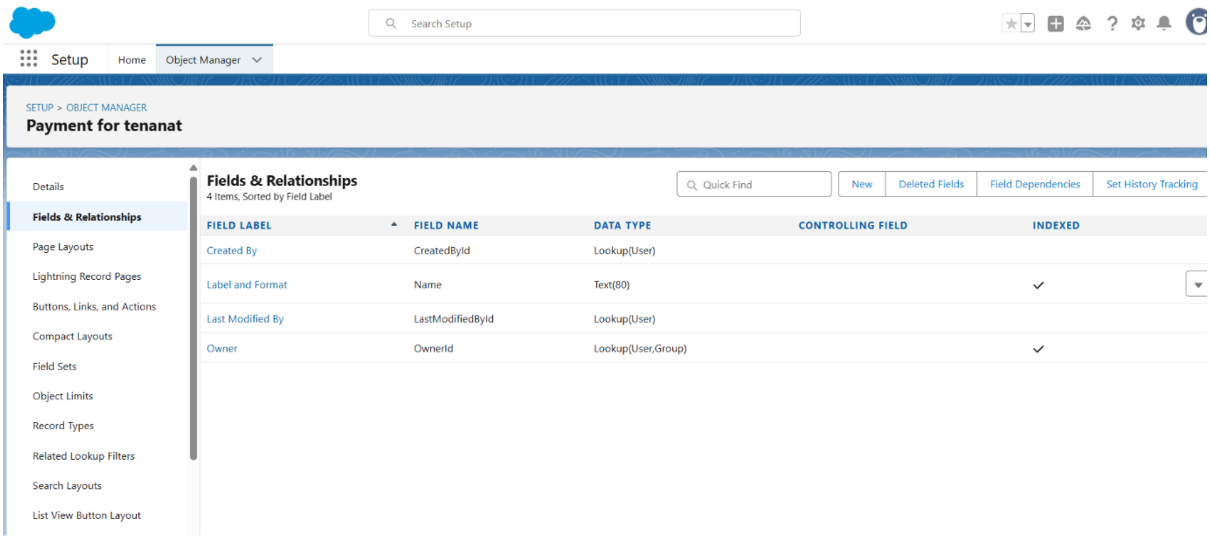
- 

- 

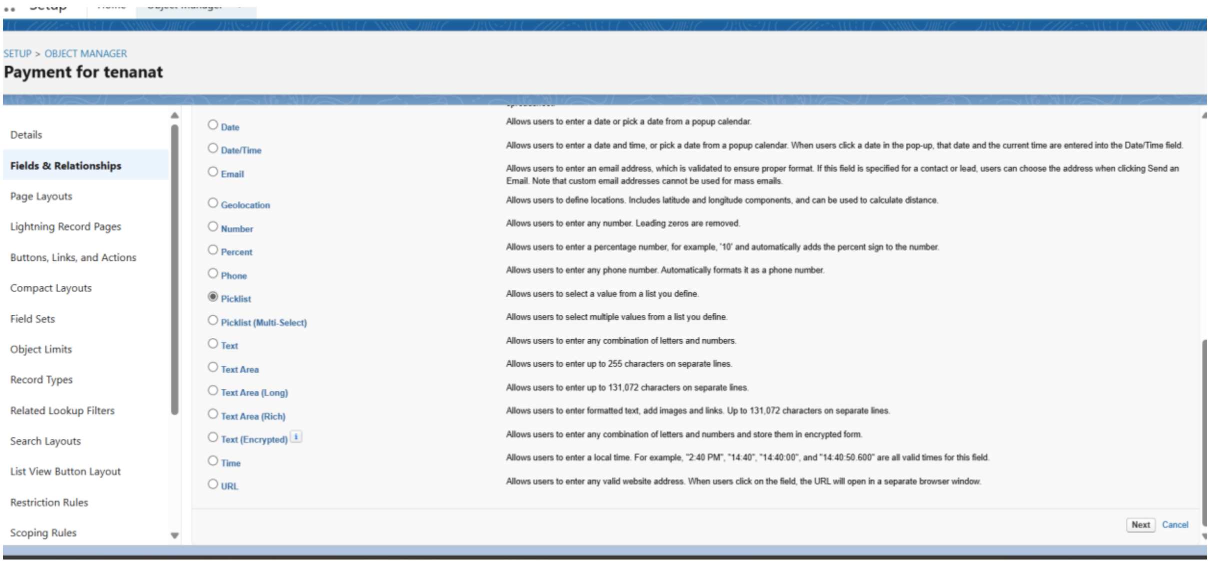
To create another fields in an object.

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

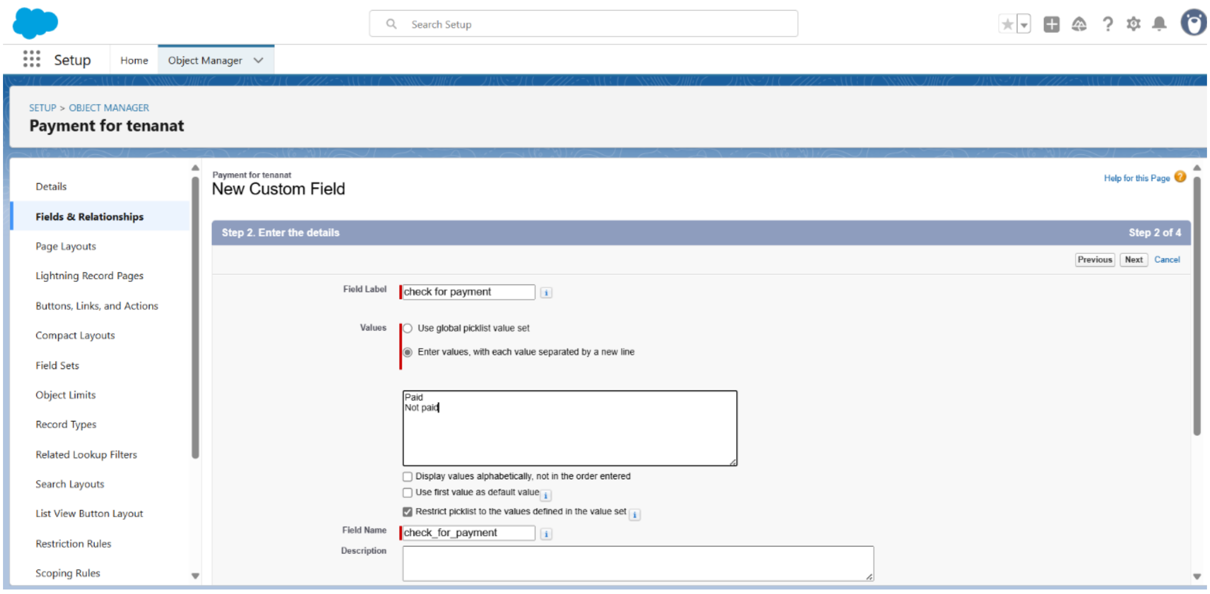


4. Fill the Above as following:

- 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.

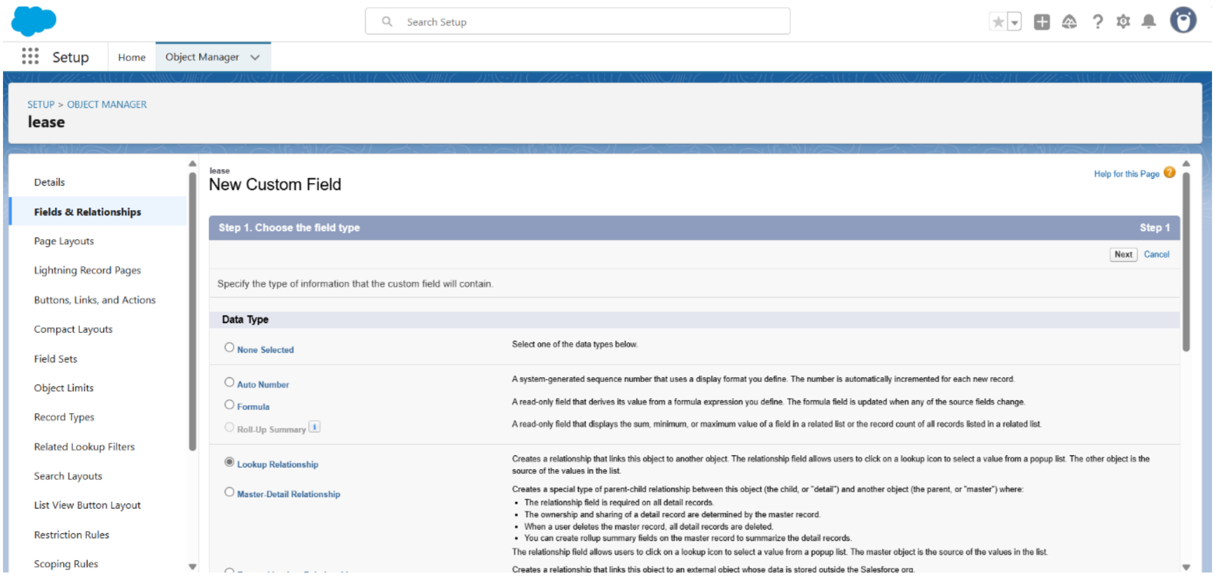


-
-

Creation Of Lookup Fields

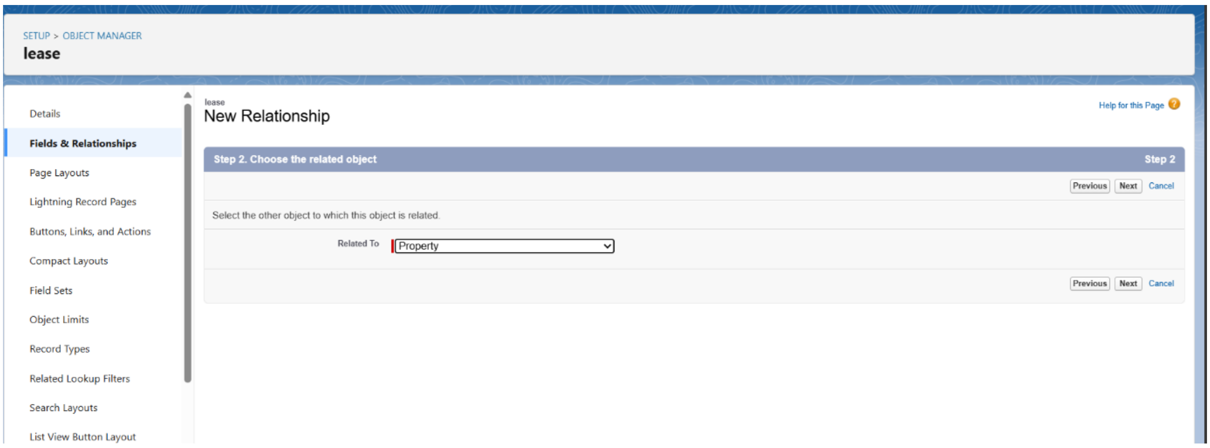
Creation of Lookup Field on Lease Object:

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



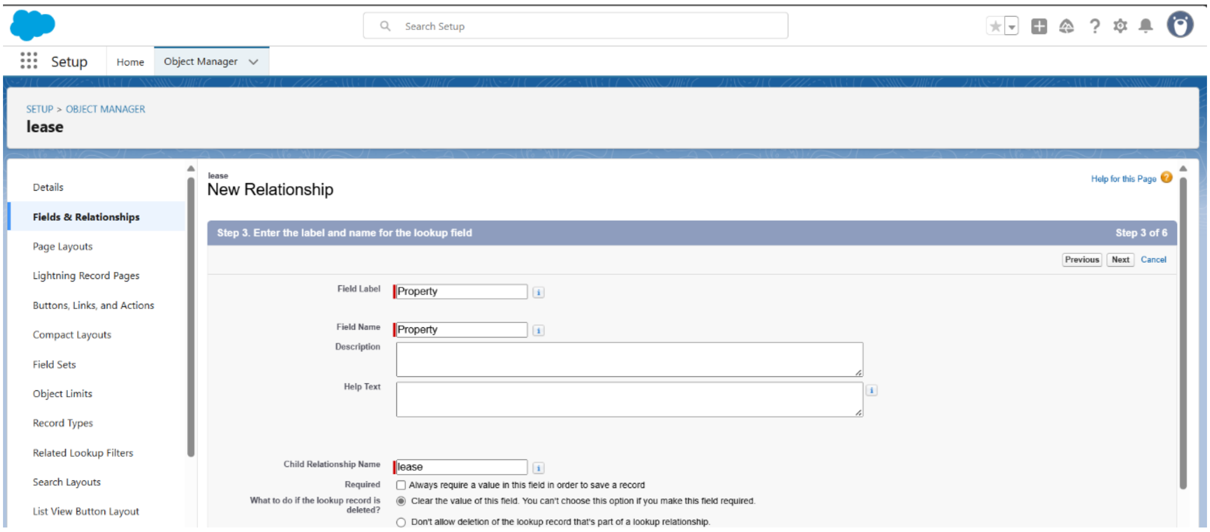
2. Now click on "Fields & Relationships" >> New
3. Select lookup relationship

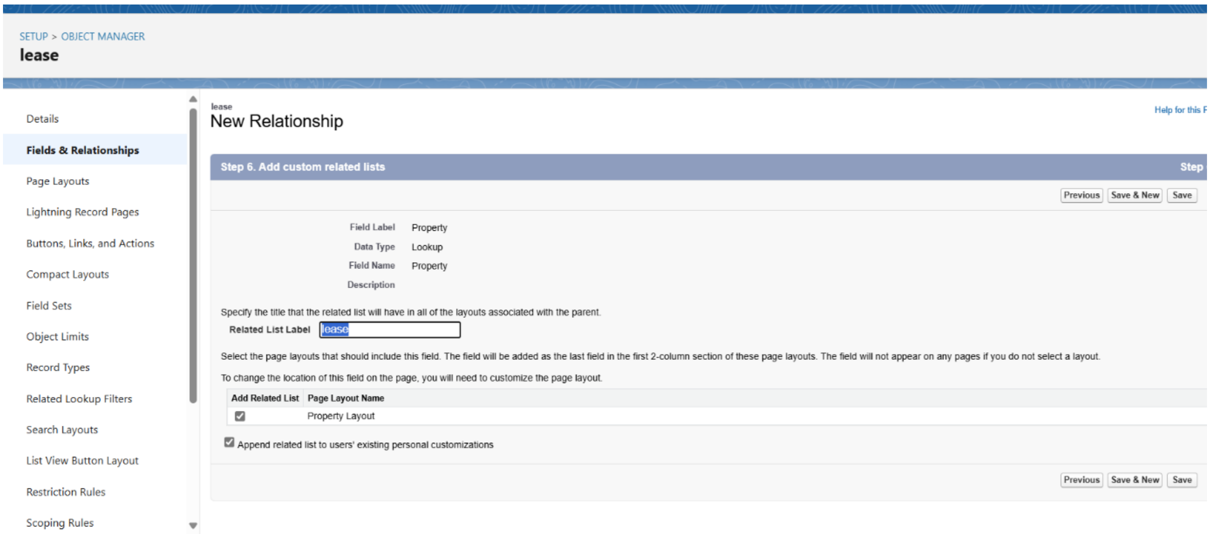
- 4. Select the related object" property" and click next.



5. Field Name: property
6. Field label: Auto generated

- 7. Next >> Next >> Save

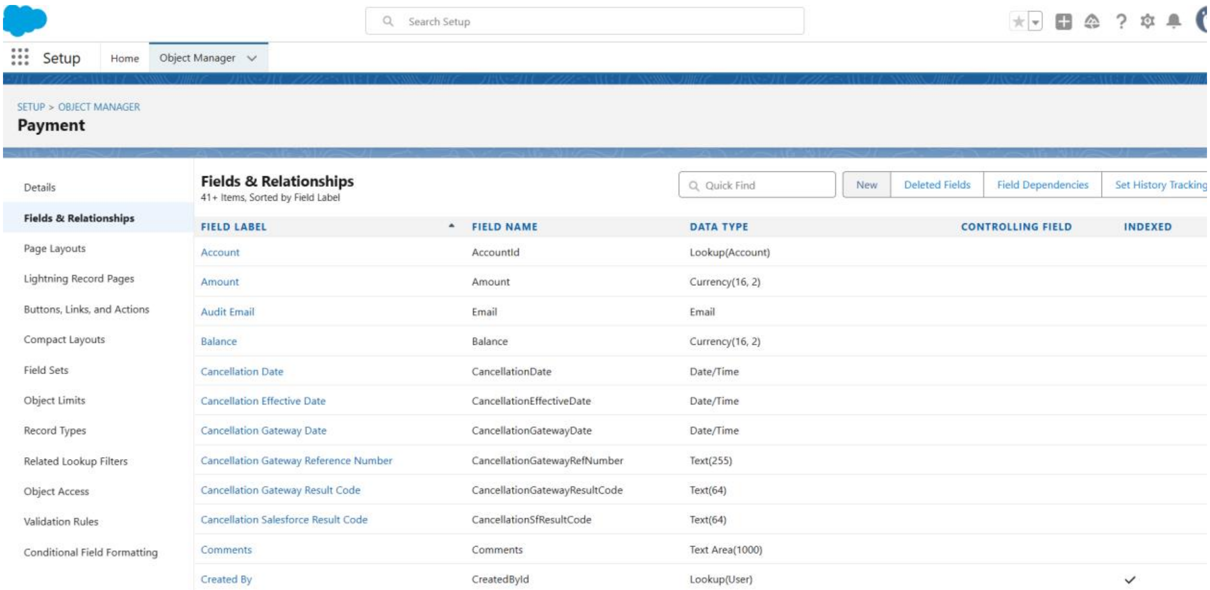


- 

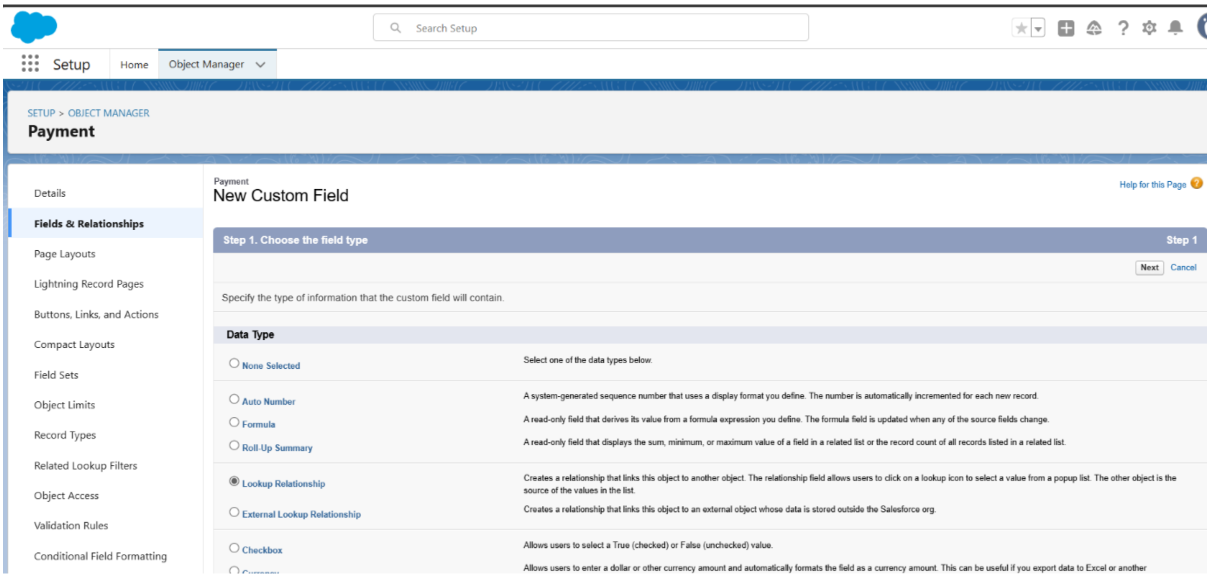
Creation of Lookup Field on Payment Object:

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

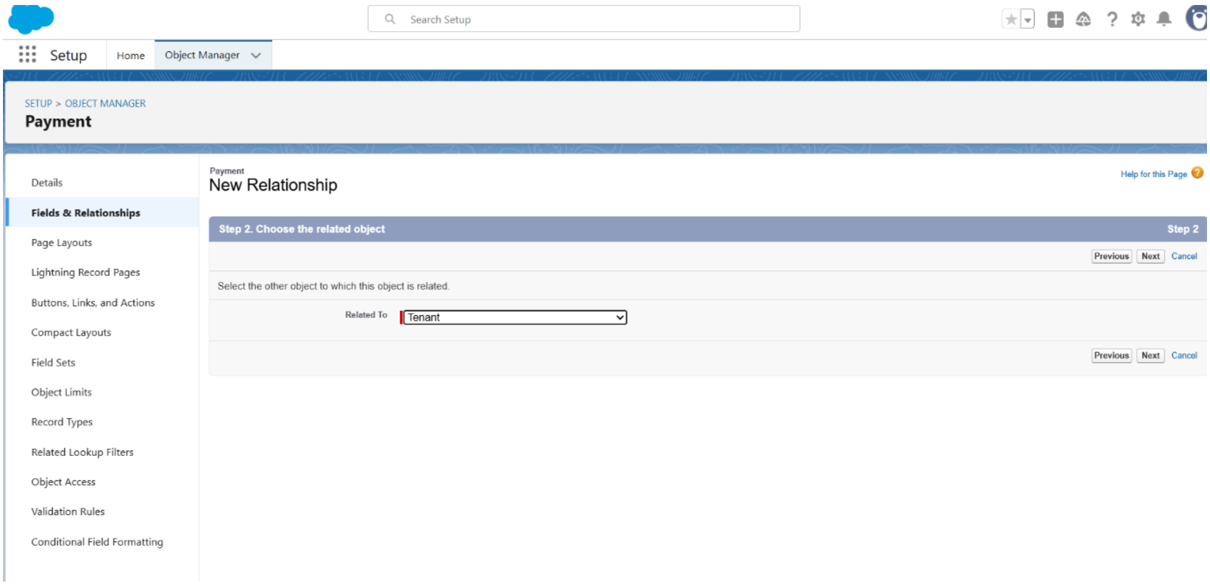
- 9. Now click on "Fields & Relationships >> New



- 10. Select lookup relationship

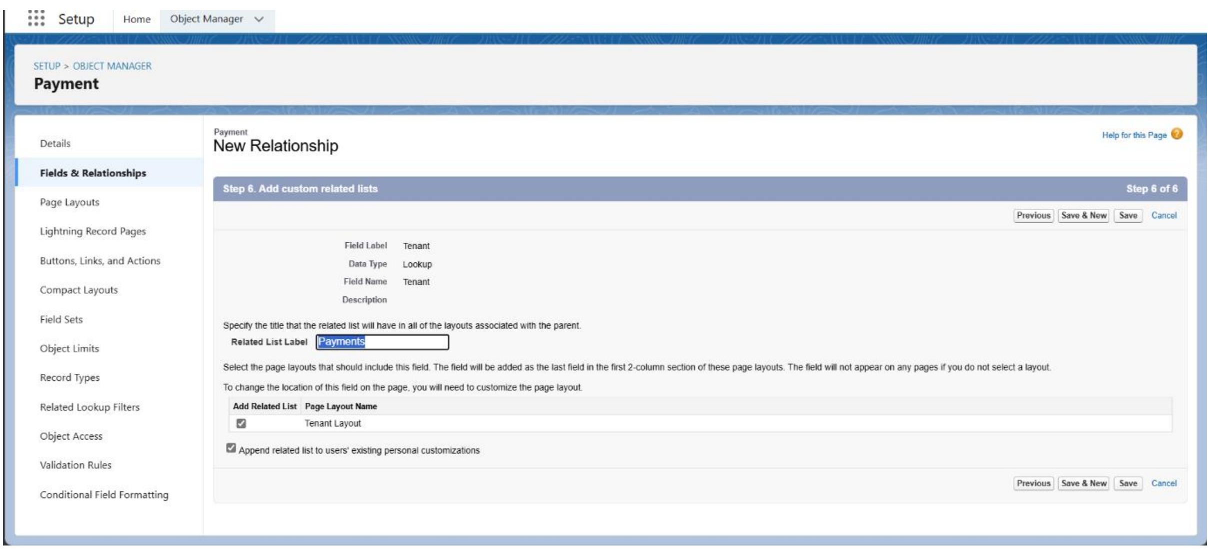


- 11. Select the related object" Tenant" and click next.



12. Field Name: Tenant
13. Field label: Auto generated

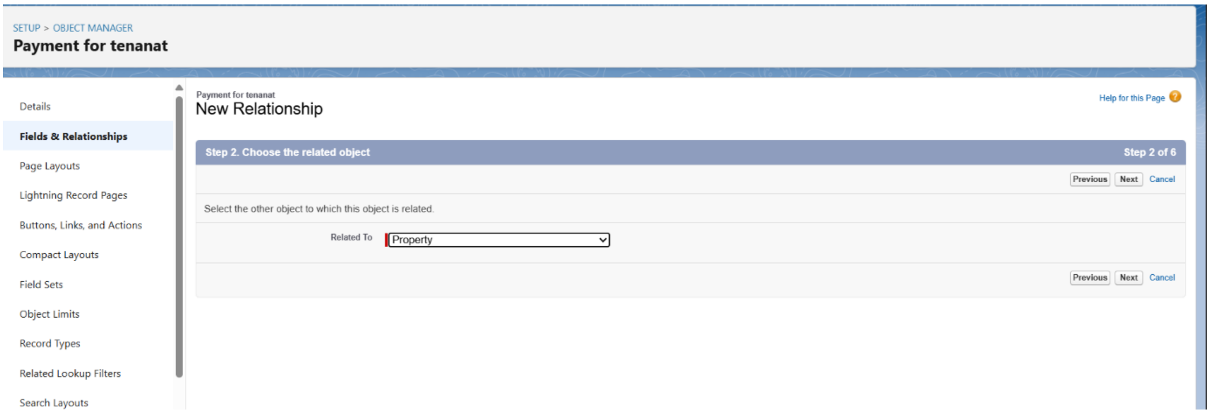
- 14. Next >> Next >> Save.

- 

Creation of Lookup Field on Payment for tenant Object:

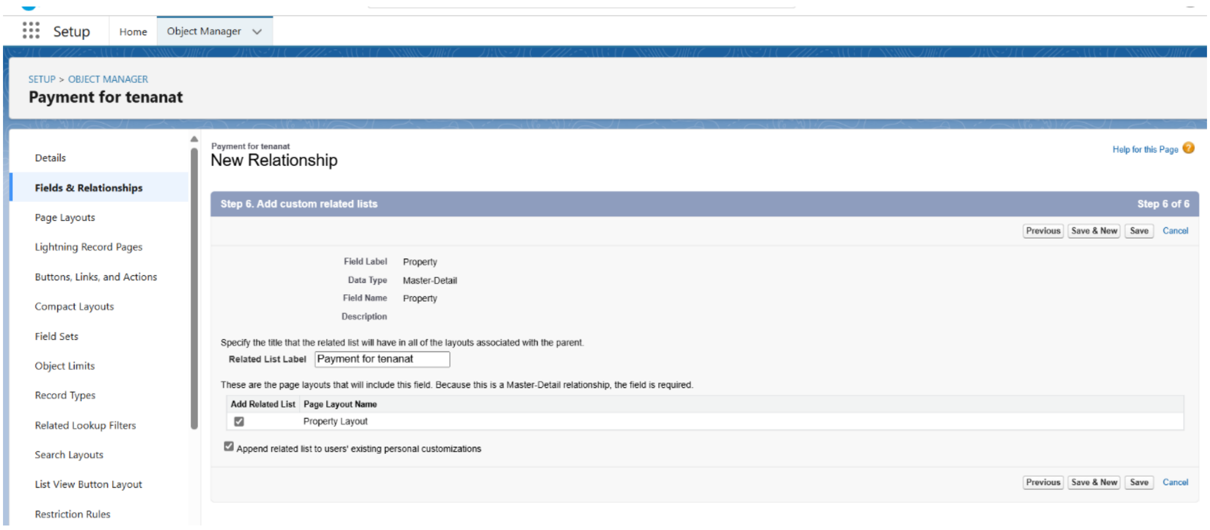
15. Go to setup>> click on Object Manager >> type object name (property) in the search bar >> click on the object.
16. Now click on "Fields & Relationships" >> New
17. Select masterdetail relationship

- 18. Select the related object" property and click next



19. Field Name: property
20. Field label: Auto generated

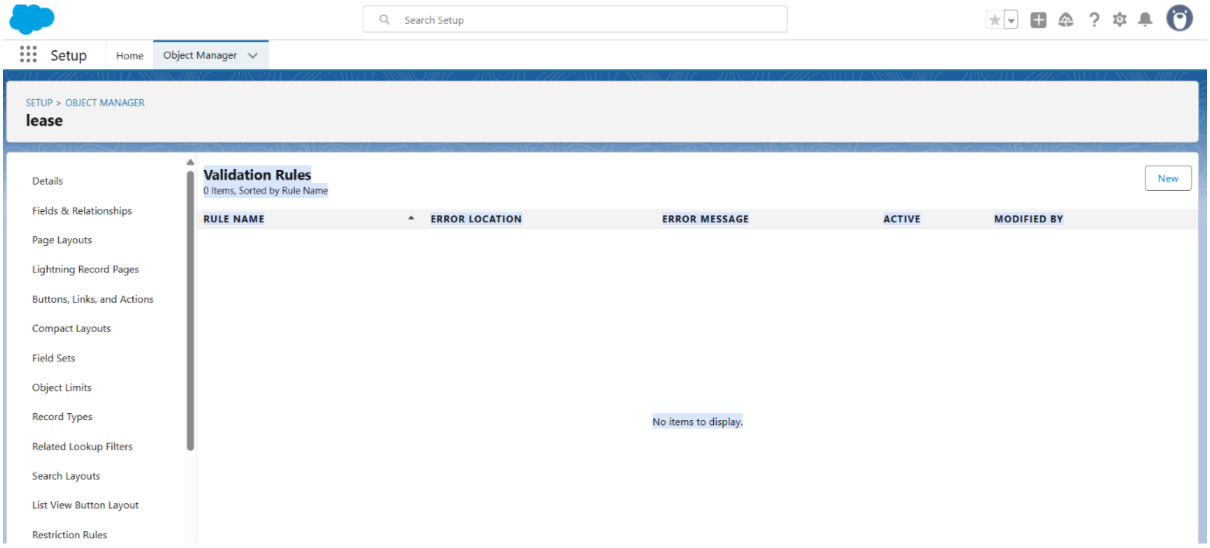
- 21. Next Next>> Save



REPORTS AND DASHBOARDS

Validation Rule

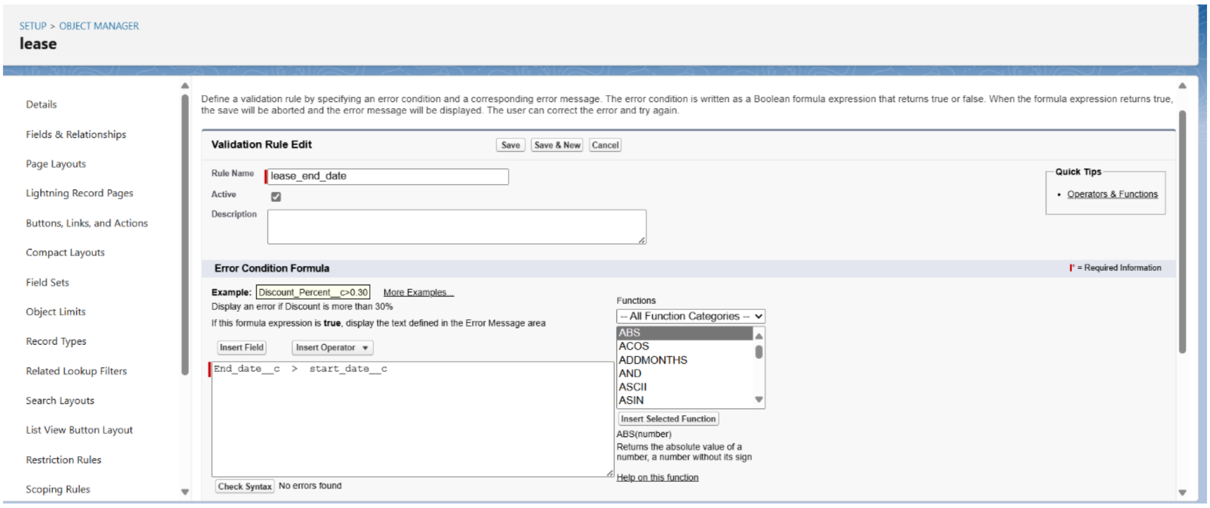
- Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.



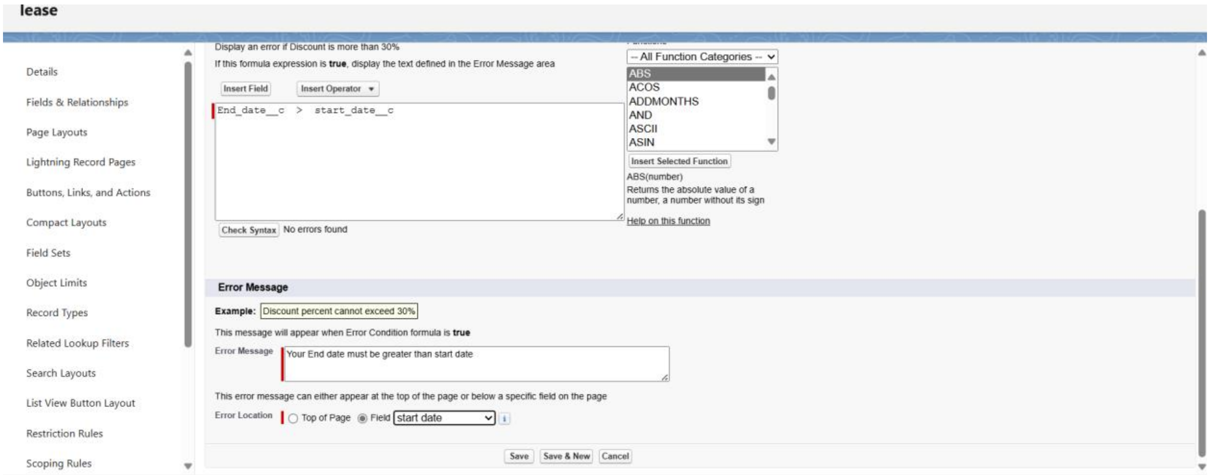
- ## To Create A Validation Rule To An Lease Object

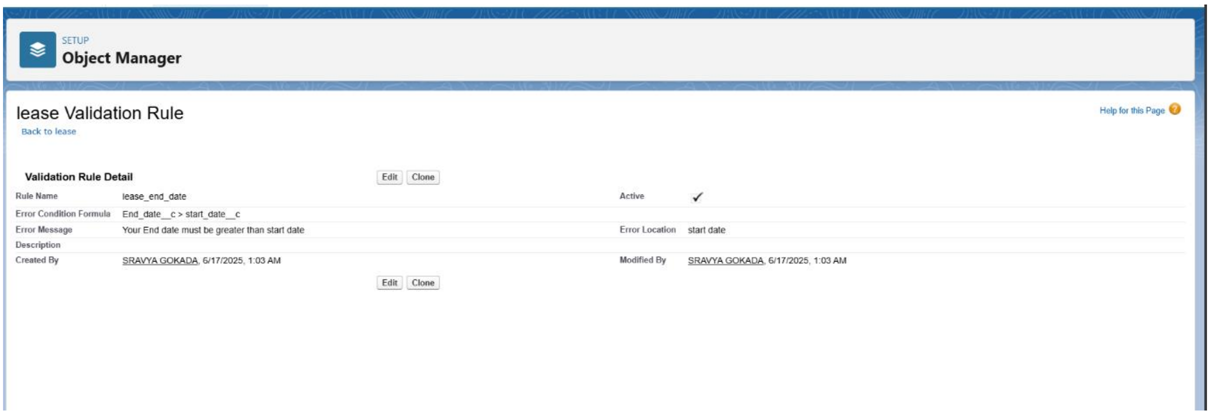
1. Go to the setup page >> click on object manager From drop down click edit for Lease object.
2. Click on the validation rule >> click New.
3.Enter the Rule name as lease_end_date".
4. Insert the Error Condition Formula as:

- 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.





DOCUMENT GENERATION

Email Templates

We use email templates to increase productivity and ensure consistent messaging. Email templates with merge fields let you quickly send emails that include field data from Salesforce records like contacts, leads, or opportunities. You can use email templates when emailing groups of people with list email or mass email-or just one person. Salesforce email templates are the easiest way to get your emails done. They help you create and send quick emails that include merge fields from Salesforce records like Contacts, Leads, Opportunities, or Custom Objects

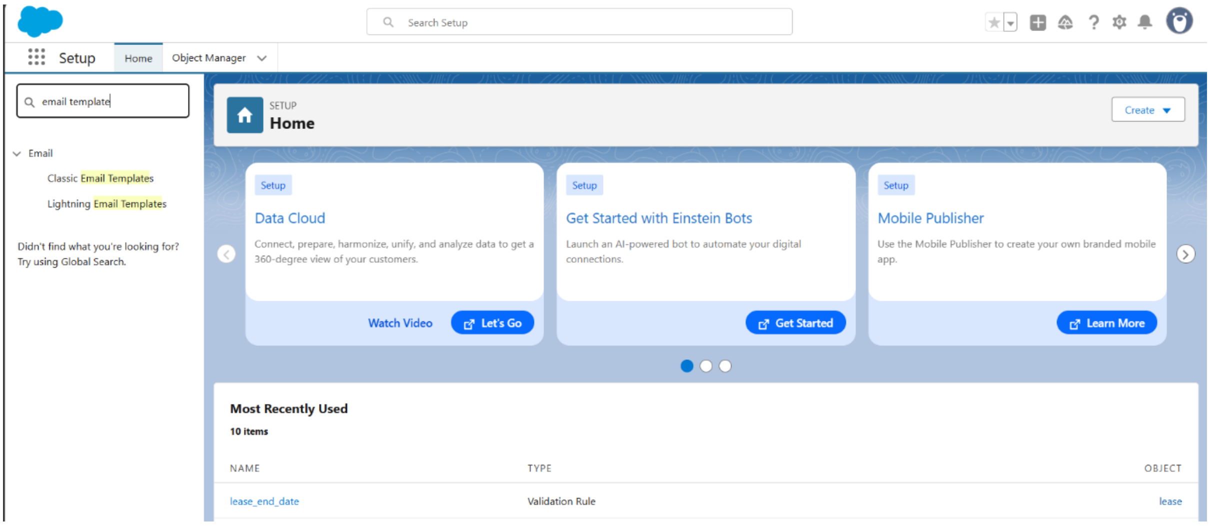
When you have a large number of contacts or leads in Salesforce, it can be difficult to keep track of who needs to be notified about new information. Salesforce email templates allow you to combine all these contacts or leads into one email and then send it out simultaneously.

Create Email Template For Tenant Leaving

To create Email Template:

1. Go to setup in quick find box enter email template

- Email Template >> click on classic



2. Click on New Email Template> Choose text

Folder: Untiled public Classic Email templates

Click on available for use

- 3. Email Template Name is "tenant leaving"

4. Template Unique Name: Auto populated

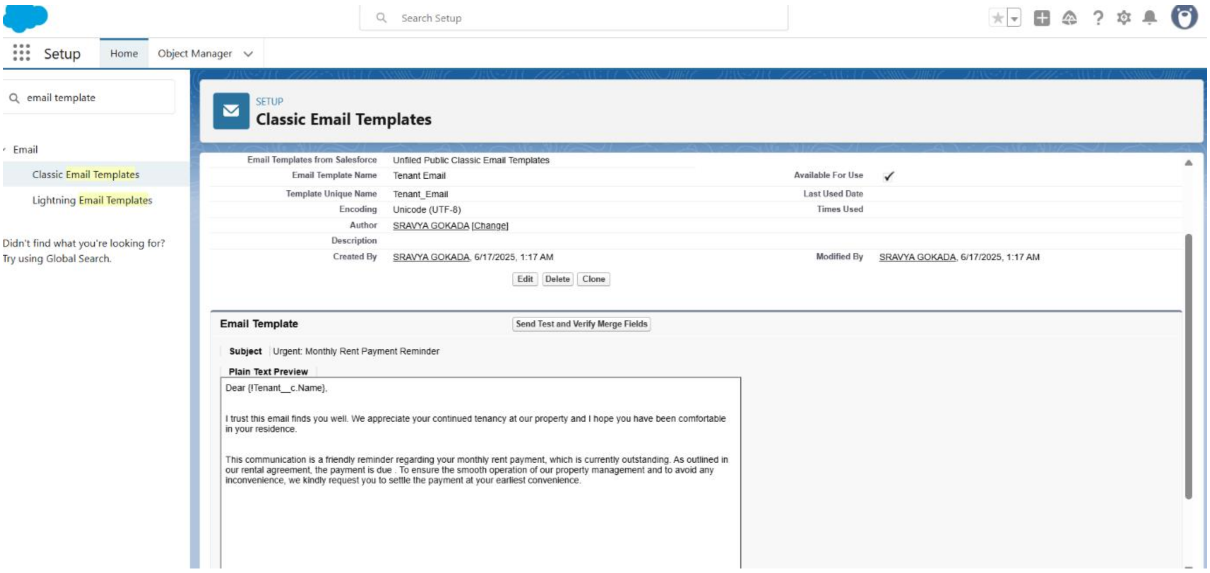
5. Subject:"request for approve the leave"

6. Email body:

Doar {!Tenant_c.CreatedBy}.

Please approve my leave.

- 7.Save


Search Setup
Setup
Home
Object Manager
email template
Email
Classic Email Templates
Lightning Email Templates
Didn't find what you're looking for?
Try using Global Search.
SETUP
Classic Email Templates
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
SRAVYA GOKADA [Change]
Description
Created By
SRAVYA GOKADA, 6/17/2025, 1:17 AM
Modified By
SRAVYA GOKADA, 6/17/2025, 1:17 AM
Edit
Delete
Clone
Email Template
Send Test and Verify Merge Fields
Subject
Urgent: Monthly Rent Payment Reminder
Plain Text Preview
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.

Create Email Template For Leave Approved

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 "Leave approved"

4. Template Unique Name: Auto populated

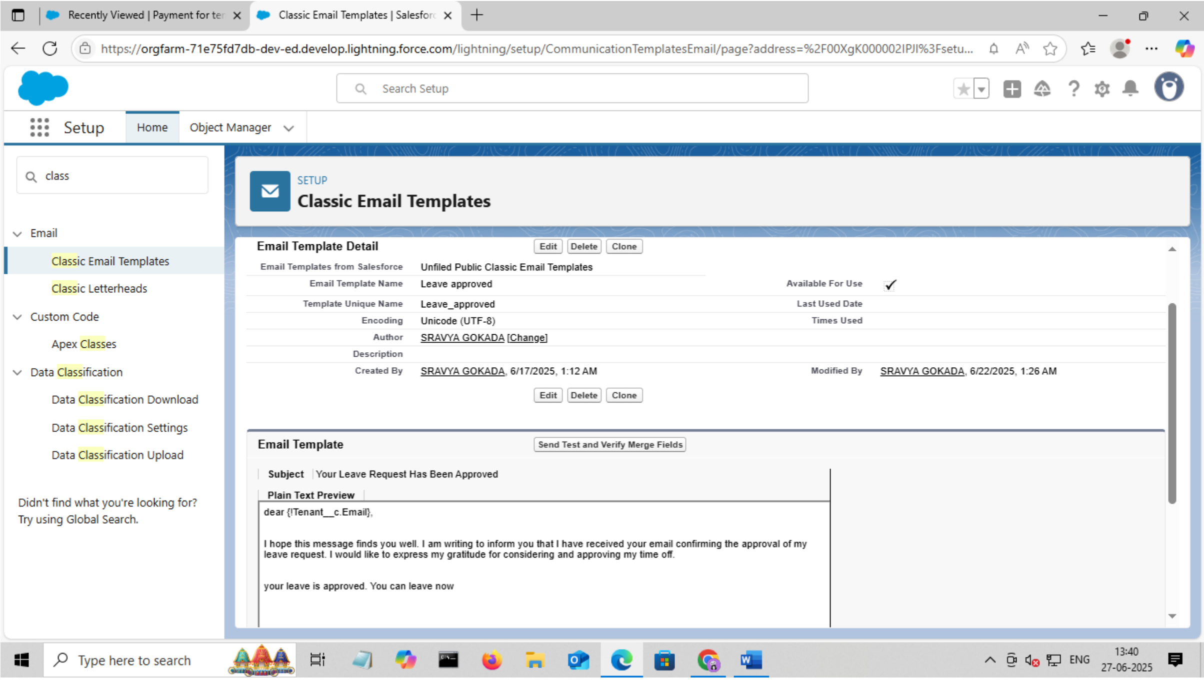
5. Subject: Leave approved?

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



7. Save

Create Email Template For Rejection For Leave

To create Email Template:

1. Go to setup in quick find box enter ernail template >> click on classic

Email Template.

2. Click on New Email Template ChooSO TEXE

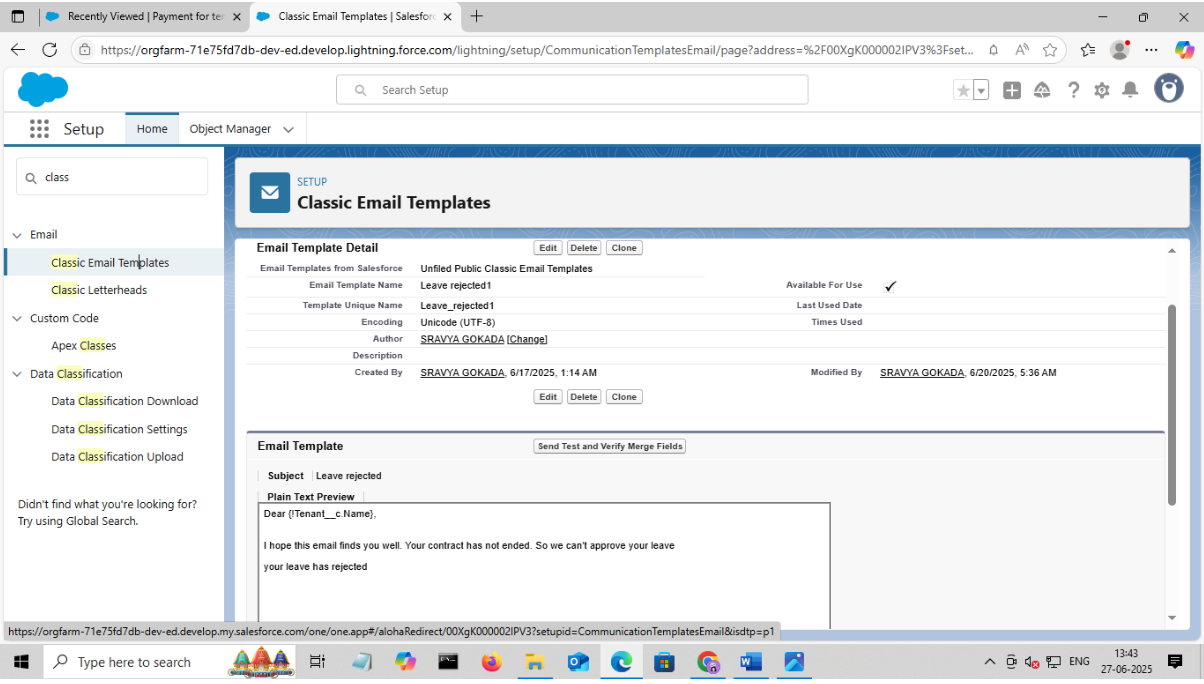
Folder: Unfiled public Classic Email templates

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



Create Email Template For Monthly Payment

To create Email Template:

1. Go to setup in quick find bax enter email template >> click on classic

   Email Template.
2. Click an >> 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

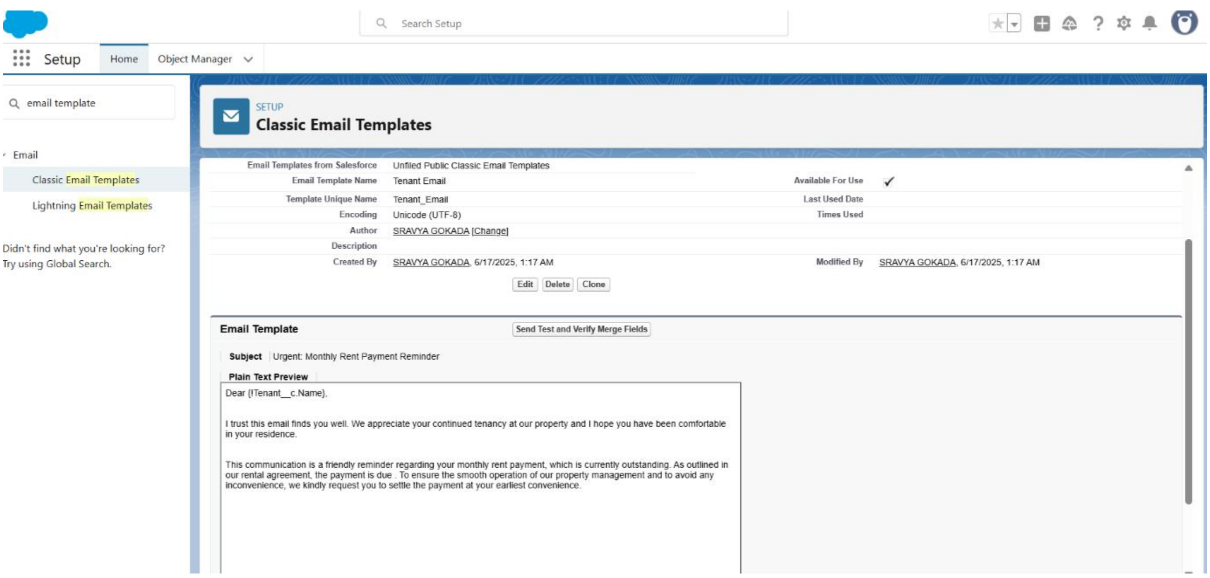
6. Emal body:

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 remainder 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

7. Save



Create Email Template For Successful Payment

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

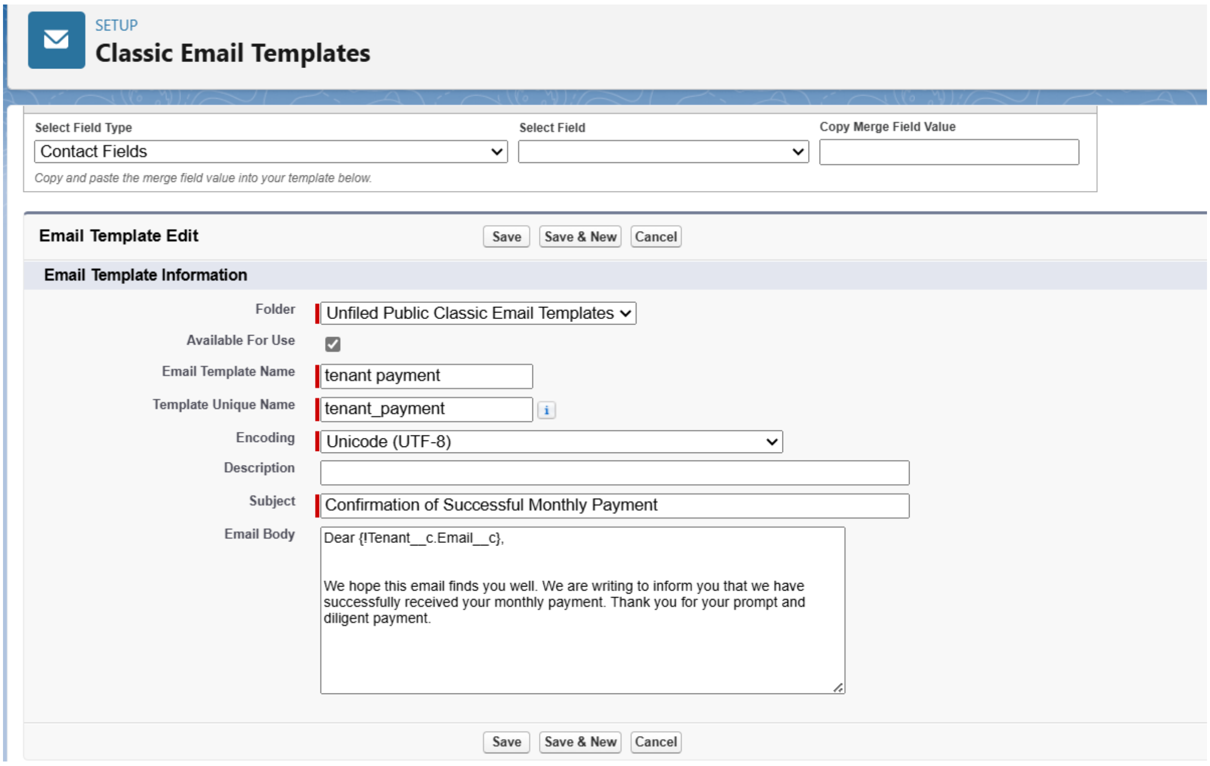
 Click on available for use

3. Email Template Name is "tenant payment”
4. Template Unique Name: Auto populated

5. Subject: Confirmation of Successful Monthly Payment
6. Email body :

Dear(Teriant_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 and Save



SECURITY AND PERMISSIONS

Approval Process

What Is Approval Process In Salesforce?

The Approval Process is an automated process that an org uses to approve records in Salesforce. For example, When In the organization, someone is not able to decide a particular thing then he can ask someone else for approval. So, for such frequent cases or situations, one can define the approval process. So, Users can take benefit of such an approval process whenever needed.

Records submitted for approval are approved by the user(s) in the organization. These users are called Approvers. A single Approval process is bound to a single object because when a rule is defined, this object influences the fields that will be available to set the criteria.

An approval process consists of finalizing the basic properties of the approval process (as shown in the below image), approval steps, and actions to be executed.

Actions in Salesforce Approval Process

There are 4 actions present except the approval steps which complete an approval process, following are:

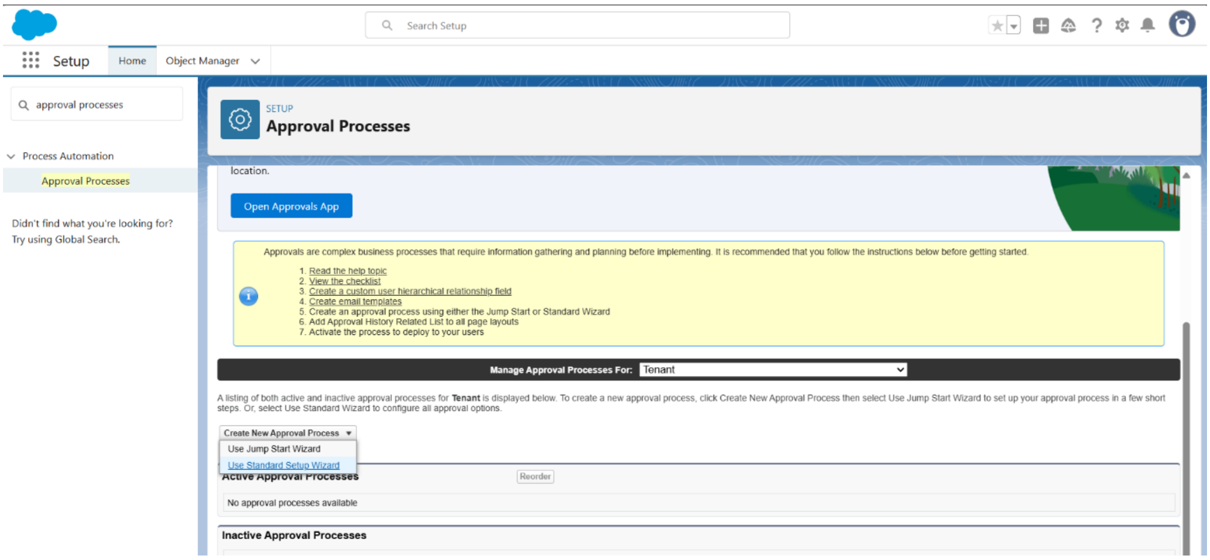
1. Initial Submission Actions

Initial submission actions are the actions that occur when a user first submits a record for approval. By default, an action to lock the record runs automatically on initial submission. Initial submission actions can include any approval actions such as email , field updates , takes , or outboard messages

Create Approval Process For Check For Vacant

To create fields in an object:

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



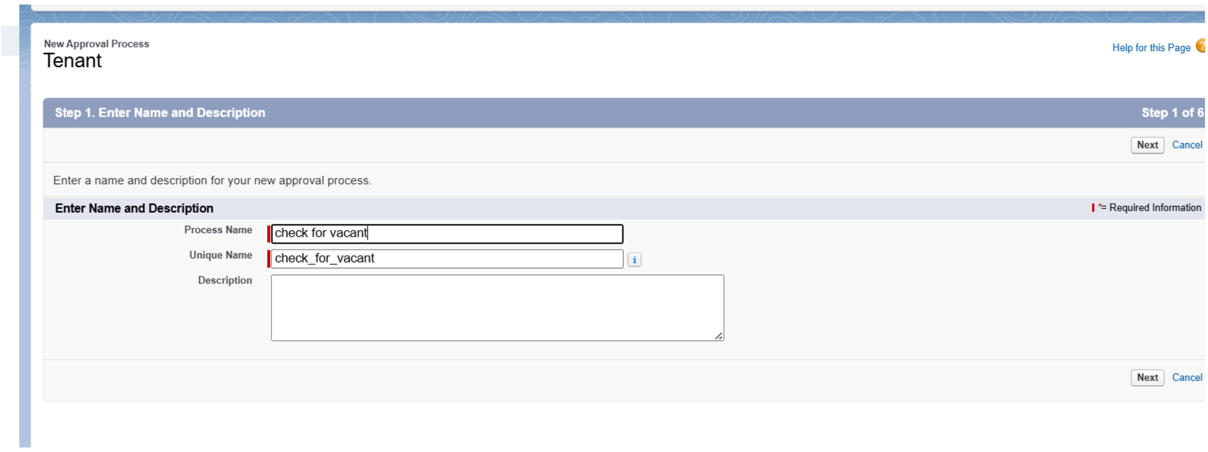
2.Manage Approval Process For >> "Tenant" from the drop down.

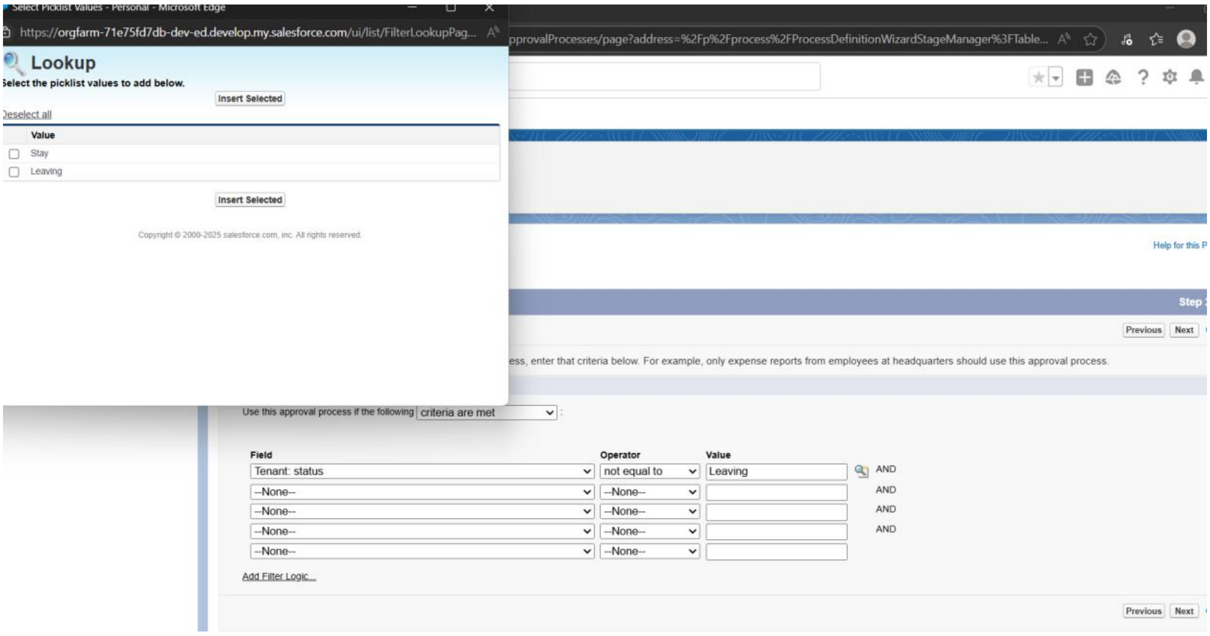
3.Click on "Create New Approval Process" >>> Use standard setup wizard.

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".

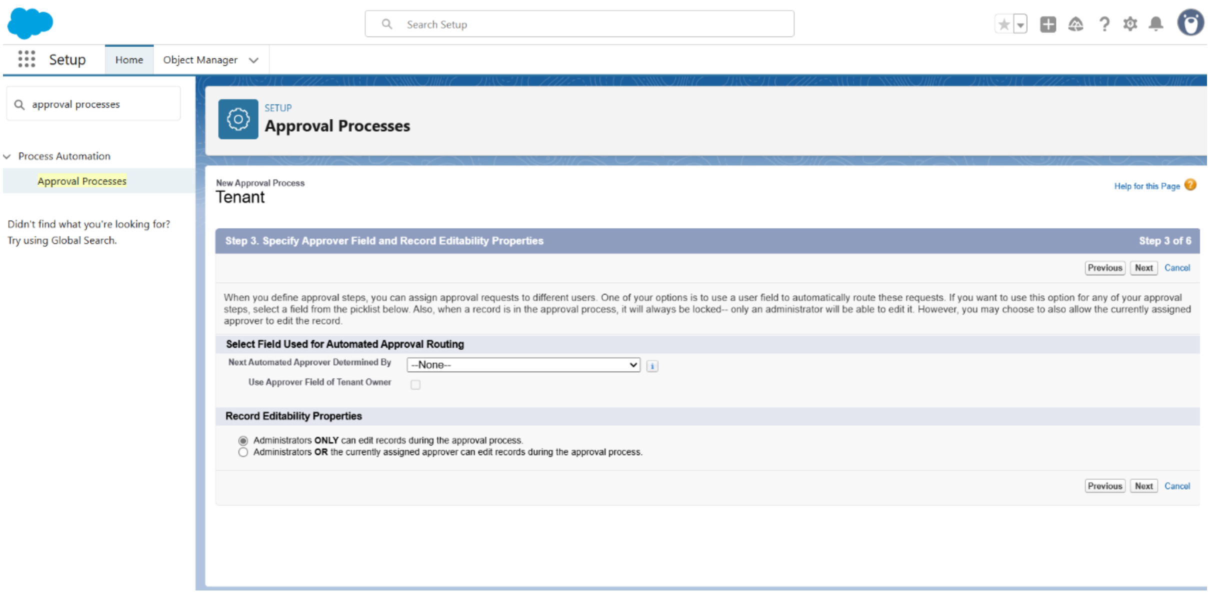
6.Click insert field, then click Next



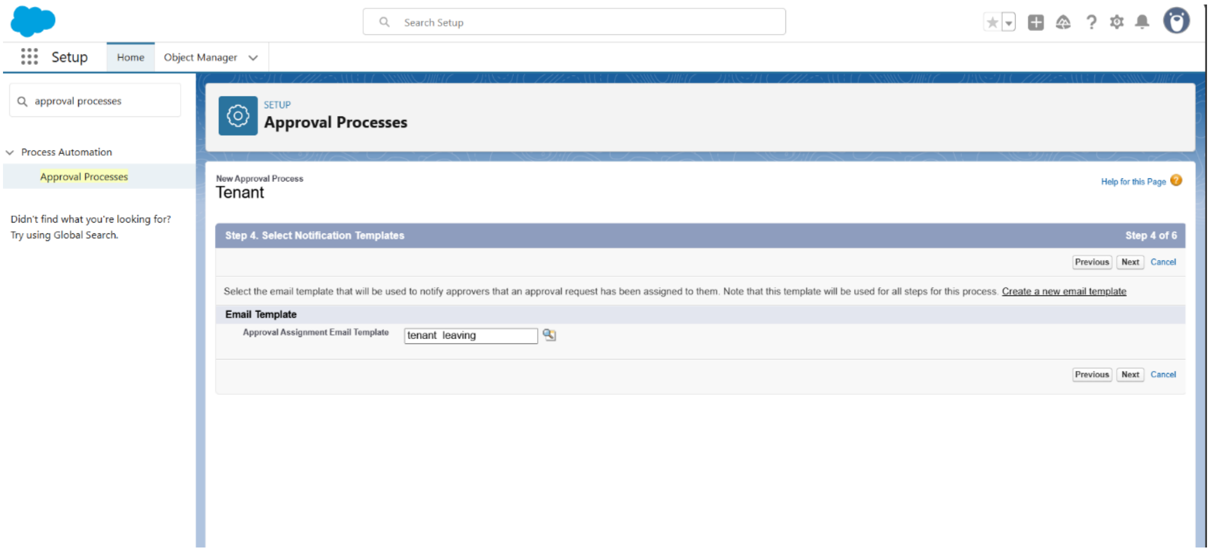


7. Next Autornated Approver determined by "None" from the drop down.

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



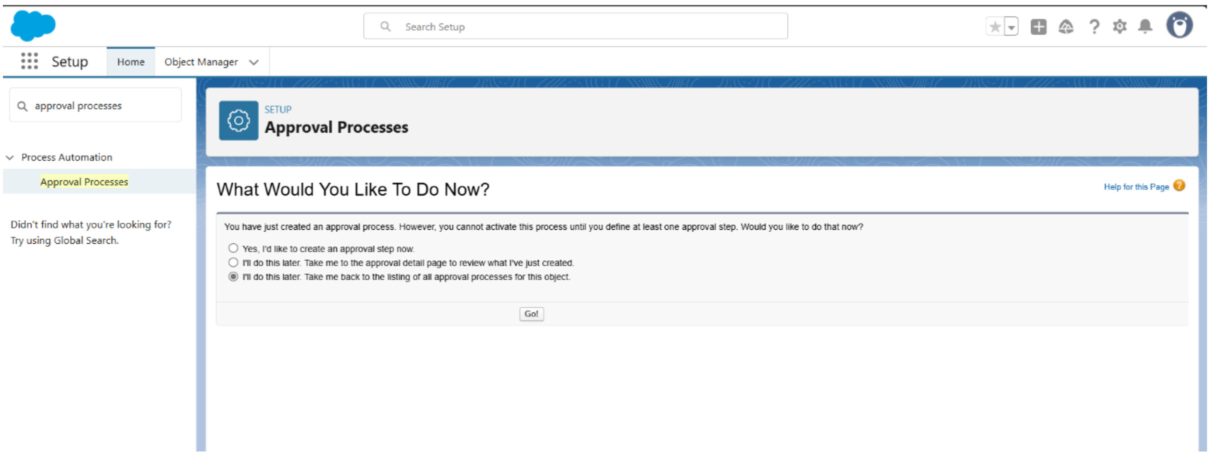
10.From the available fields select >> Tenant Name, and then add >>Add it to the selected. Then Next. Make sure Display approver history is checked.



- 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.

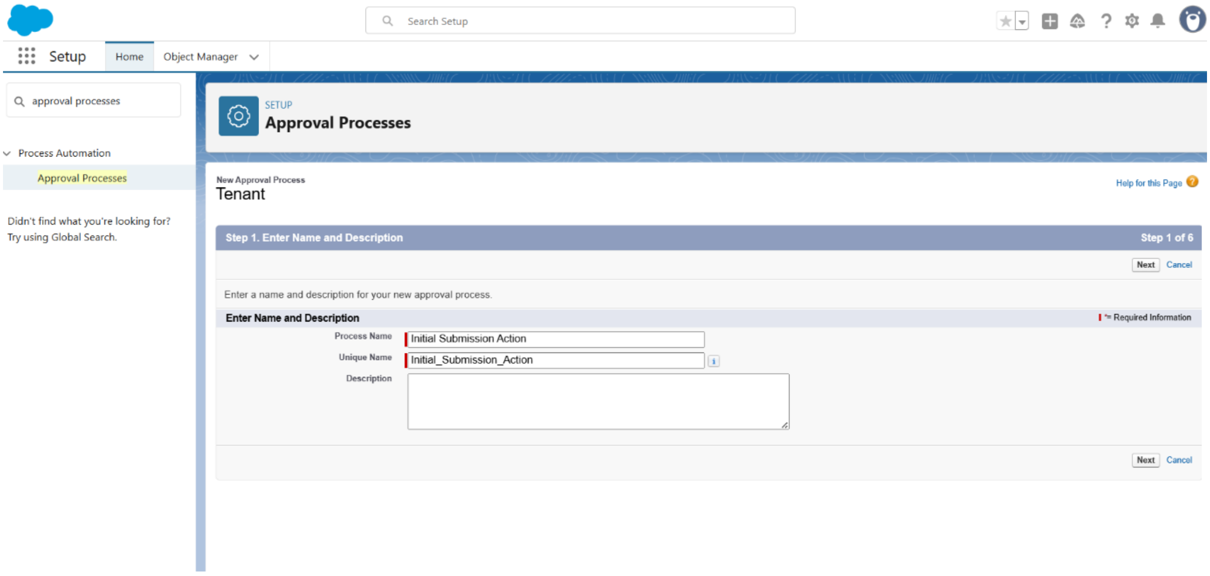
Then click save.



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

Initial Submission Action:

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



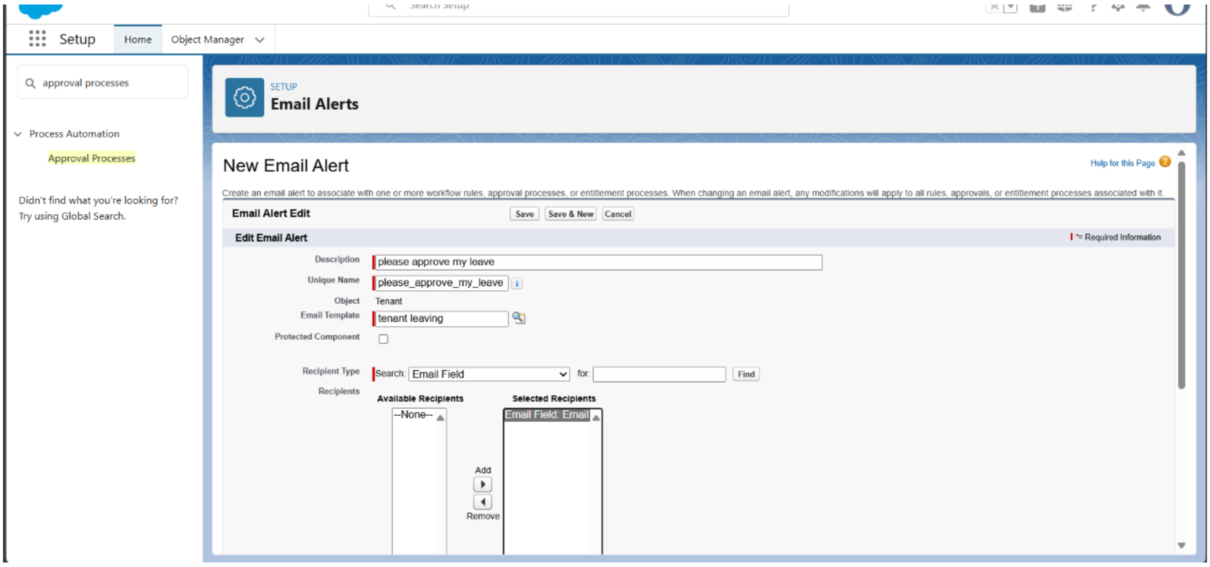
2.Description: "please approve my leave".

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

Final Approval Action

1. Under Final approval action click on new and then select email alert.

2. Description: "Tenant leaving".

3. unique name: auto populated

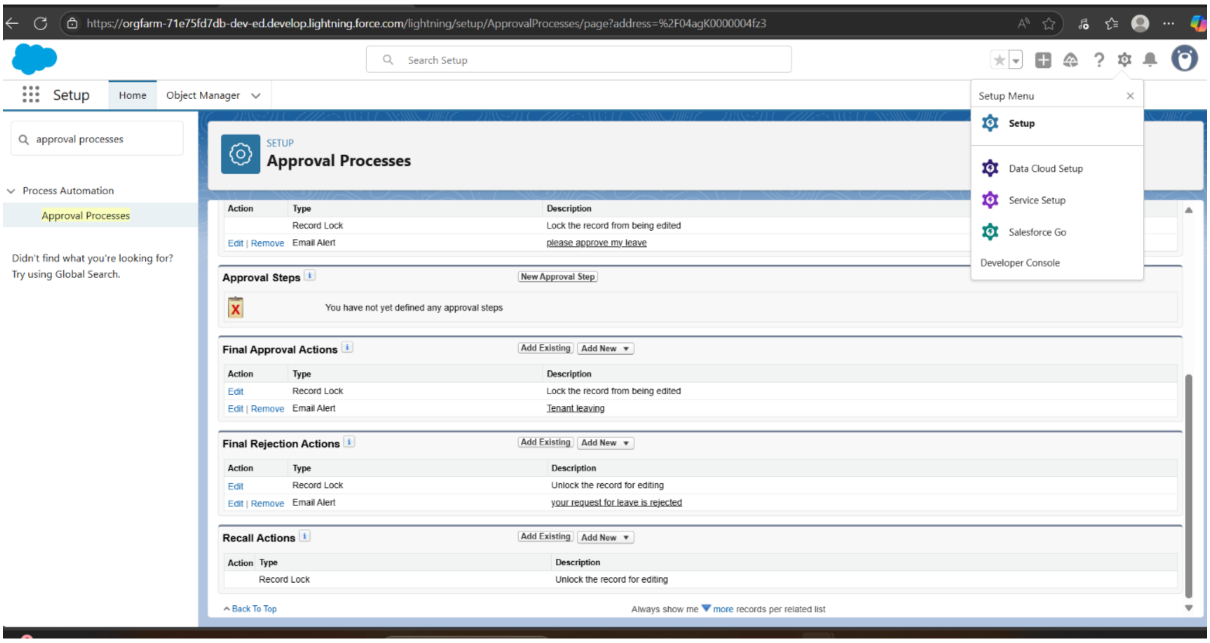
4. Ernail template: Leave approved

5. Recipient type: Email field

6. Available Recipients: Email field: Email

7. From Email address: Current user's email

8. Click save



Final Rejection Action

1. Under final rejection action click on add new and then select email alert.

2.Description: "your request for leave is rejected".

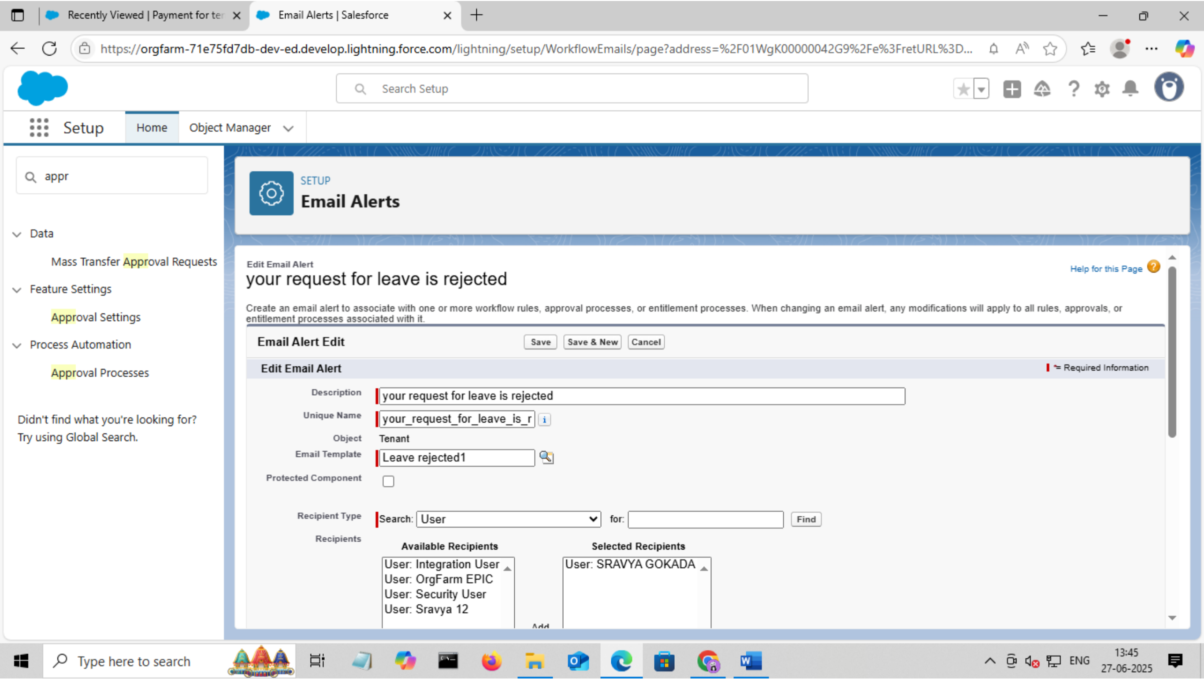
3.unique name: auto populated

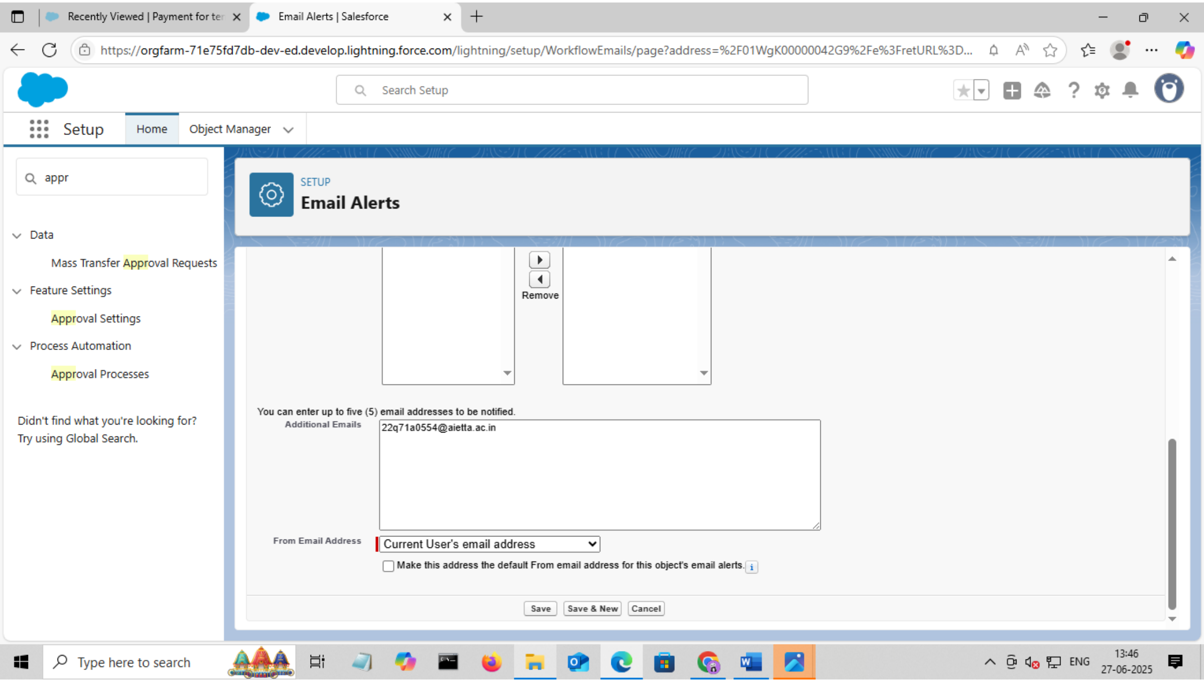
4.Email template: leave rejected

5. Recipient type: Email field

6. Available Recipients: Email field: Email

7. From Email address: Current user's email an save.



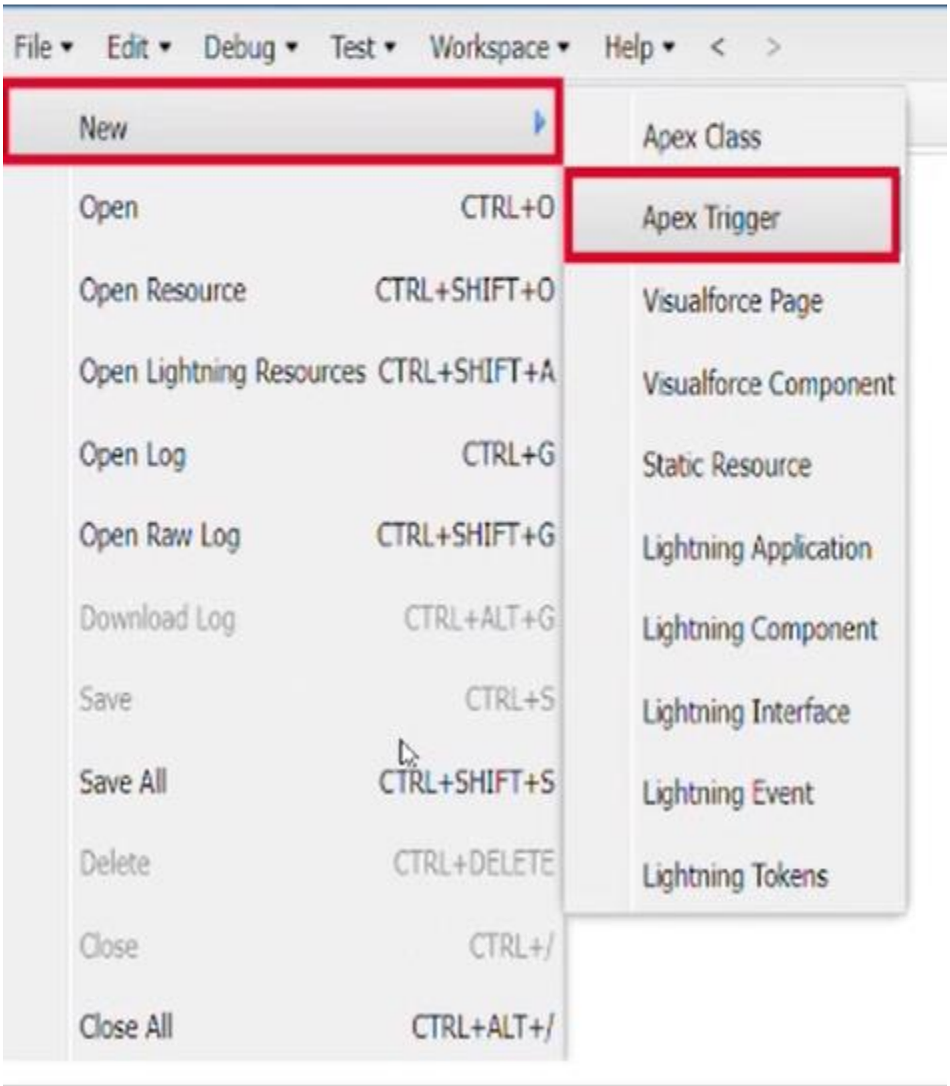


Apex Trigger

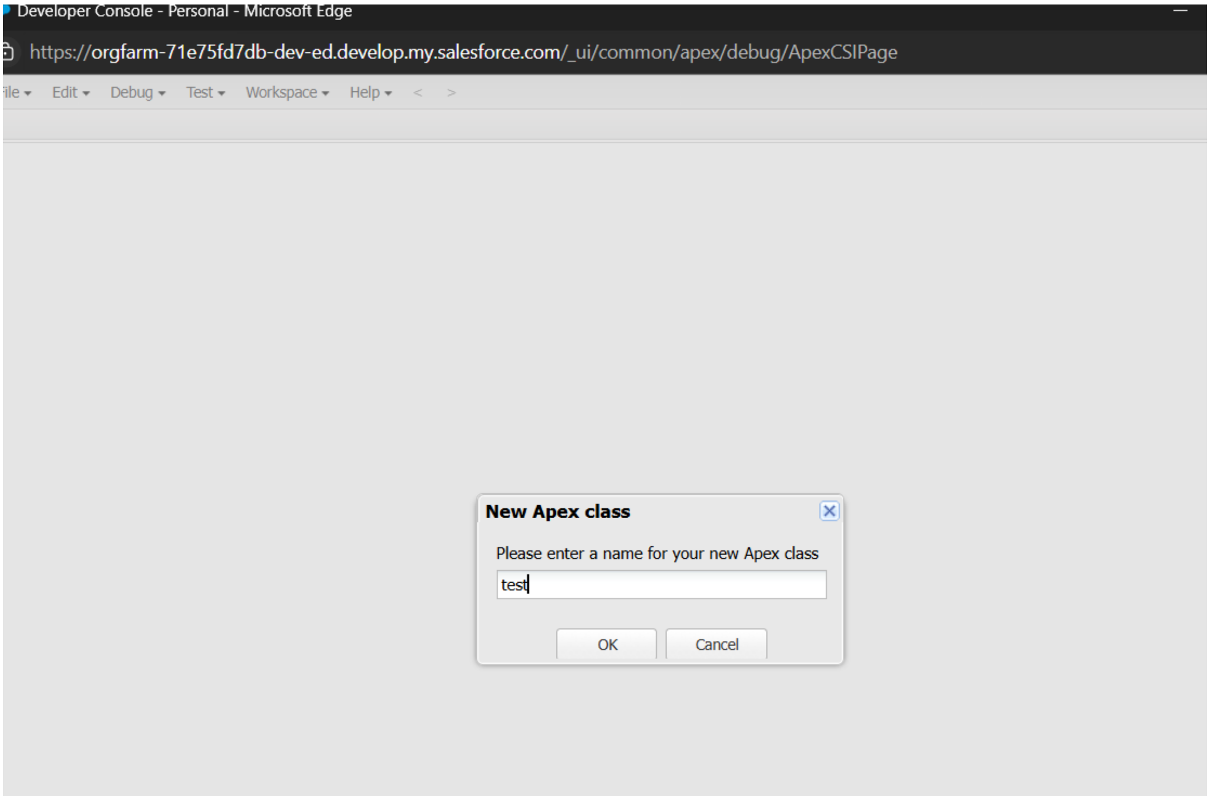
Use case:

The tenant and property are in a master-detail relationship, wherein each tenant is associated with only one property. When a tenant attempts to create a new record with an existing property, an error should be displayed, indicating that a tenant can have only one property.

Write a code to achieve this requirement using Salesforce developer skills to fulfill the Managers requirement.



2.Give the Apex Trigger name as "test", and select "Tenant_c" from the dropdown for sObject.

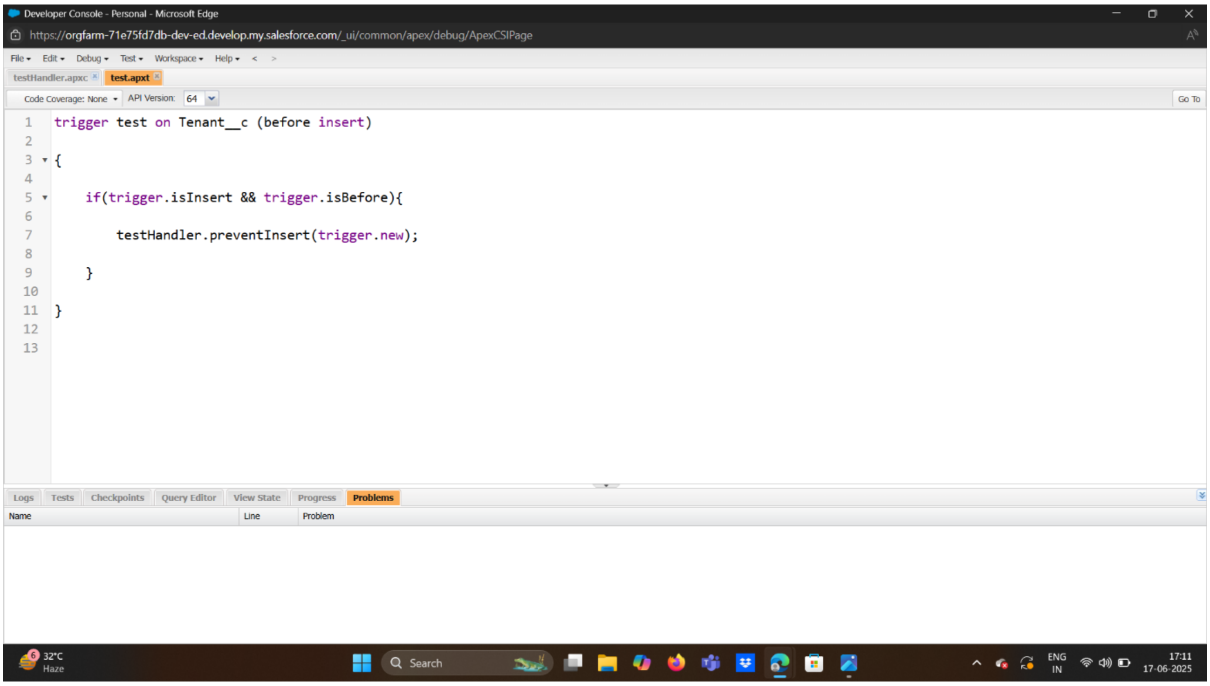


3.click submit

4.Now write the code logic here

Trigger Code:

```
trigger test on Tenant_c (before insert)
{
   if(trigger.isInsert && trigger.isBefore]{
          testi landler.preventInsert(trigger.new);
   }
}
```



Apex logic:

```
public class testHandler {

  public static void preventInsert(List<Tenant_c> newlist) (

            Set<Id> existing Propertylds = new Set<Id>();
               for (Tenant c existing Tenant: [SELECT ID, Property_c FROM Tenant c WHERE
      Property_c != null]) {

                  existing Propertylds.add(existingTenant.Property_c);

  for (Tenant c newTenant: newlist) {

          if (newTenant.Property_c != null && existing
      Propertylds.contains(newTenant.Property_c)) {

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


              }        }
```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

test.apxt * testHandler.apxc *

Code Coverage: None ▾ API Version: 64

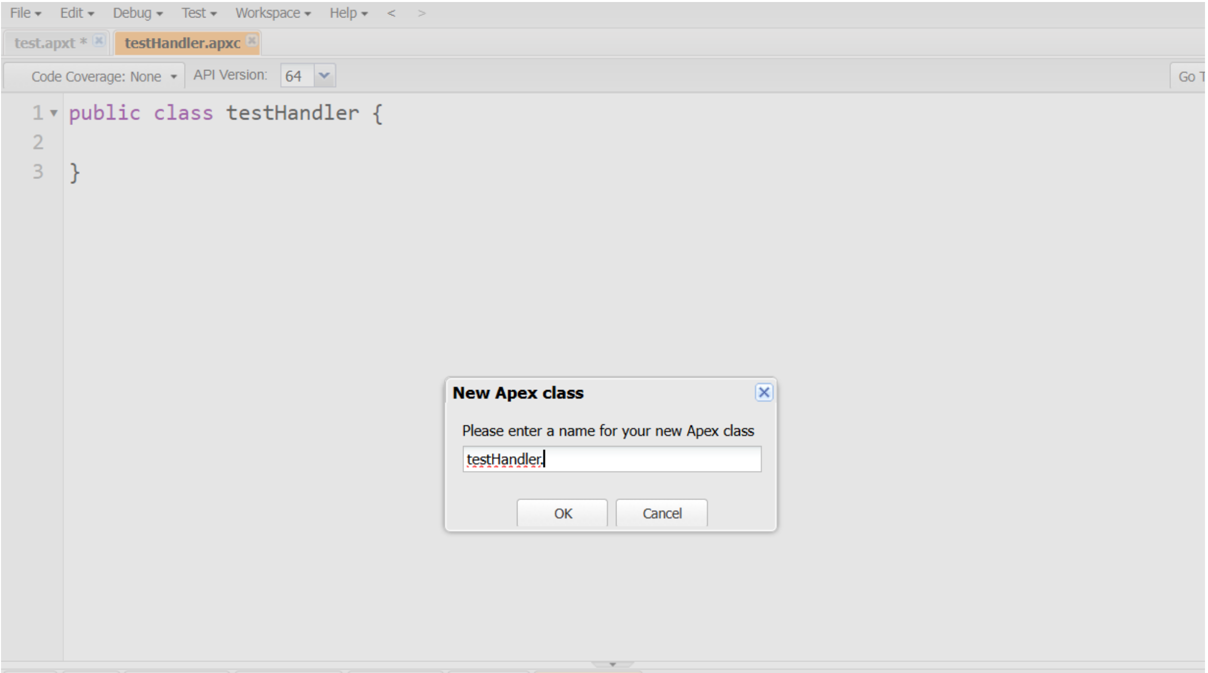
```
public static void preventInsert(List<Tenant__c> newlist) {

    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');
```

Create An Apex Handler Class

To create a new Apex Class follow the below steps:

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

2. Enter class name as testHandler.



Apex logic:

public class testHandler

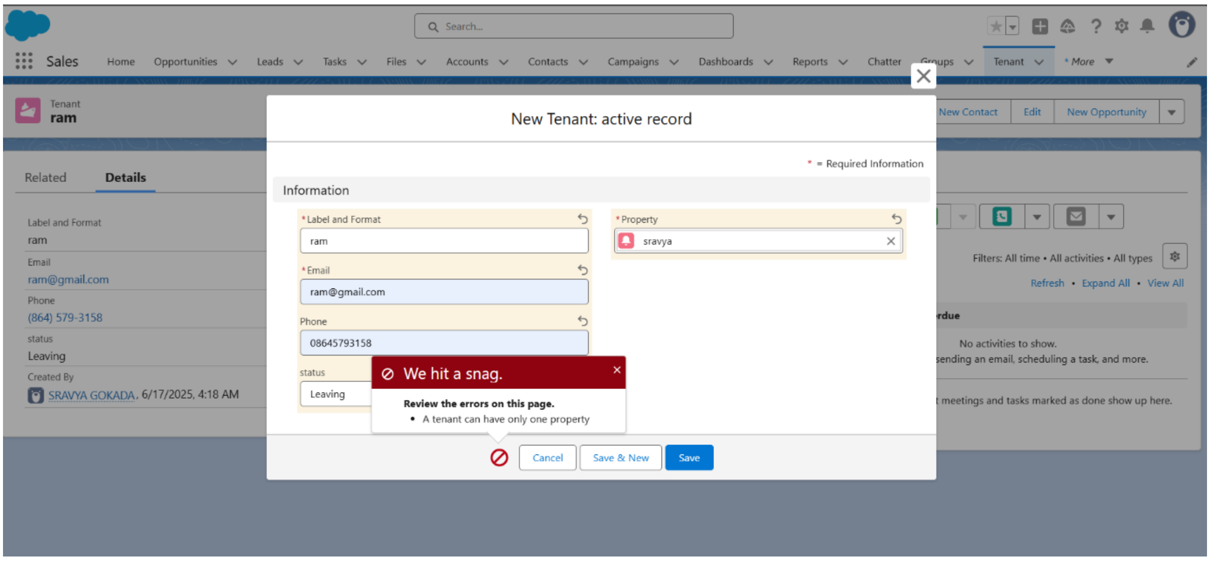
public static void preventinsert List<Tenant ce newlist)

Set<id> existing Propertylds = new Set<id>: for (Tenant c existing Tenant:
(SELECT id, Property FROM

Tonant WHERE Property != null) existing Propertylds.addiexisting Tenant. Property).

Testing the tigger

Try to creat new tenant with the exiting property then it shows the error



FLOWS

What is a flow?

In Salesforce, a flow is a tool that automates complex business processes. Simply put, it collects data and then does something with that data. Flow Builder is the declarative interface used to build individual flows.

Flows fall into five categories:

Screen Flows. These are flows that have a Ul element and require input from users. These types of flows are either launched as an action or embedded as an element on a Lightning page.

Schedule-Triggered Flows: These autolaunched flows launch at a

specified time and frequency for each record in a batch, and they run in the background.

Autolaunched Flows: Run automated tasks with this flow type.
Autolaunched flows can be invoked from other flows (subflow), process builder, from within an Apex class, from a set schedule, from record changes, or from platform events.

Record-Triggered Flows: These autolaunched flows run in the

Background either before a record save or after the record is saved when a record is created, updated, or deleted.

Platform Event-Triggered Flows: When a platform event message is received, these autolaunched flows run in the background

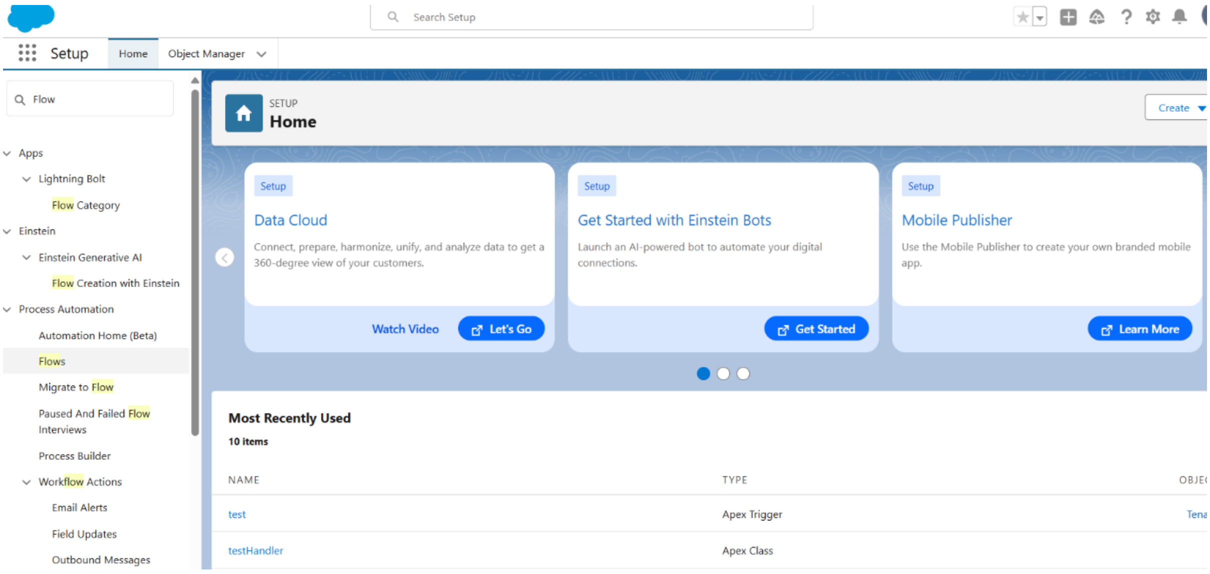
When and why should we use a flow

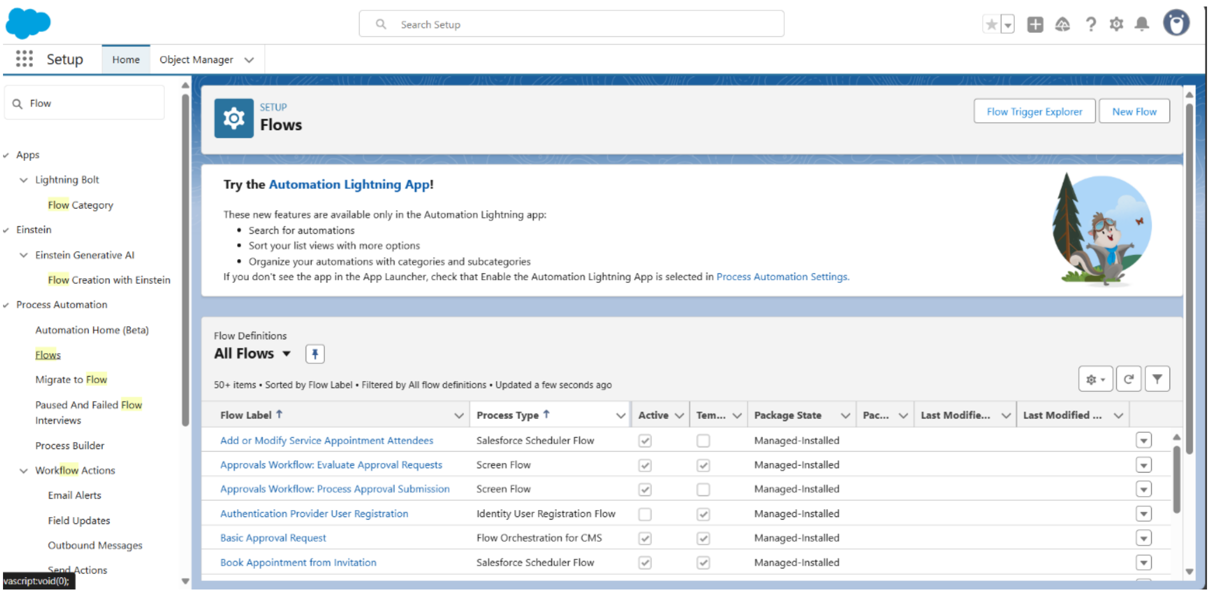
If you need to generate a new automated business processzor user guided experience that does not reach the complexity threshold for Apex

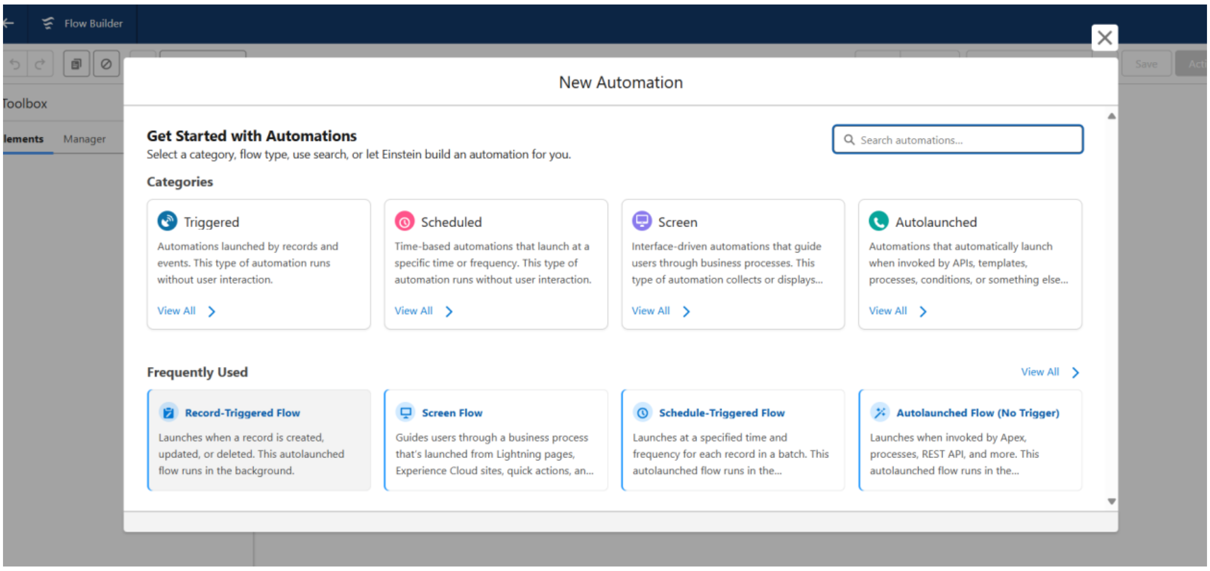
Create Flow For Monthly Payment

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.

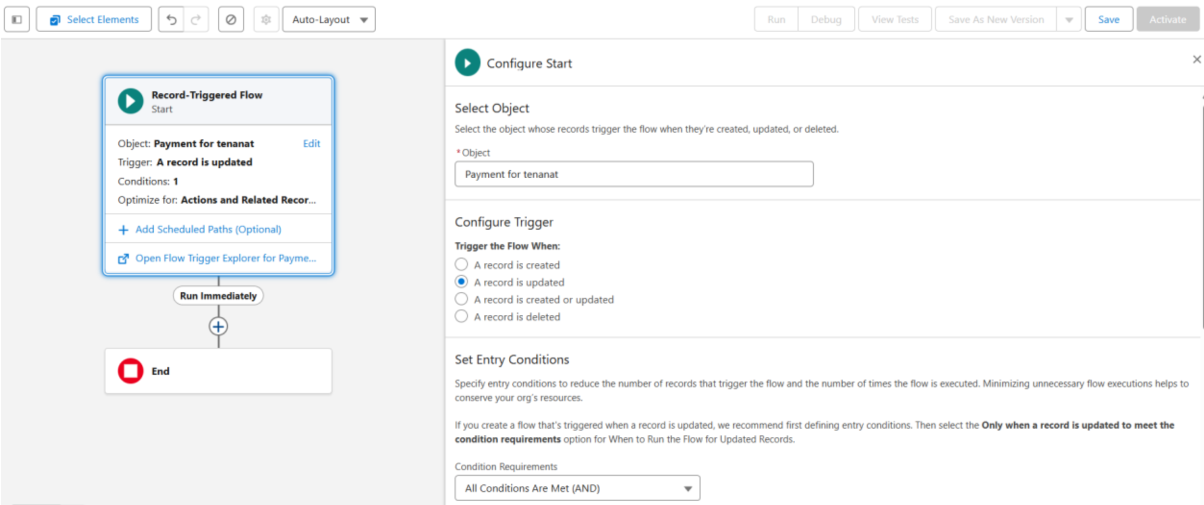
2. Select the record Triggered flow.Click on create.







3. Under Object select "Payment for tenant. Click on A record is updated



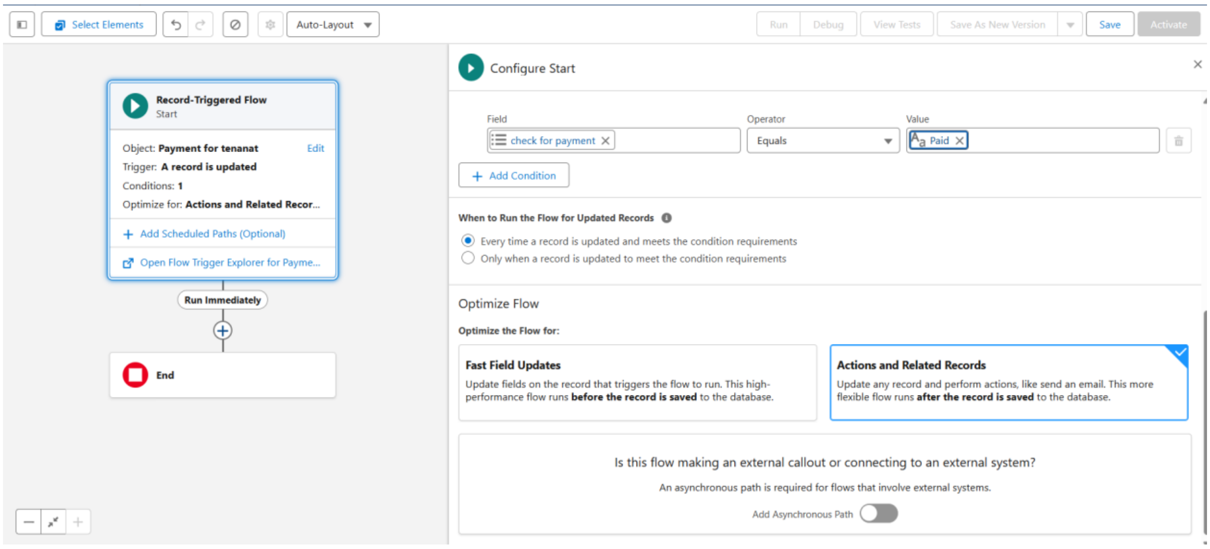
4. Set Entry Conditions

Under Condition Requirements

All Conditions are met

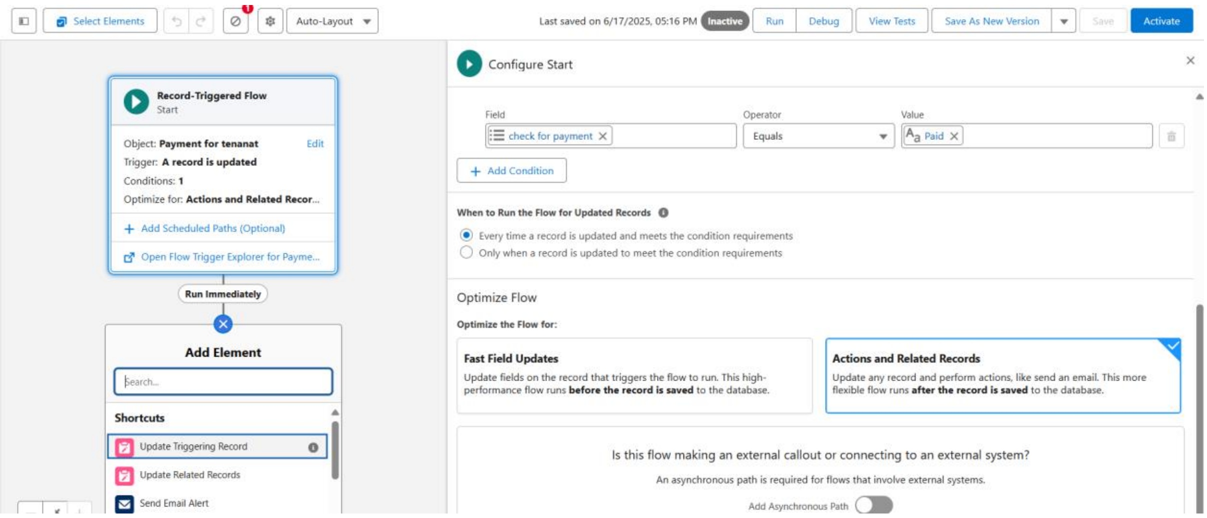
Field: Operator: Equals Value: poid

check for payment_c

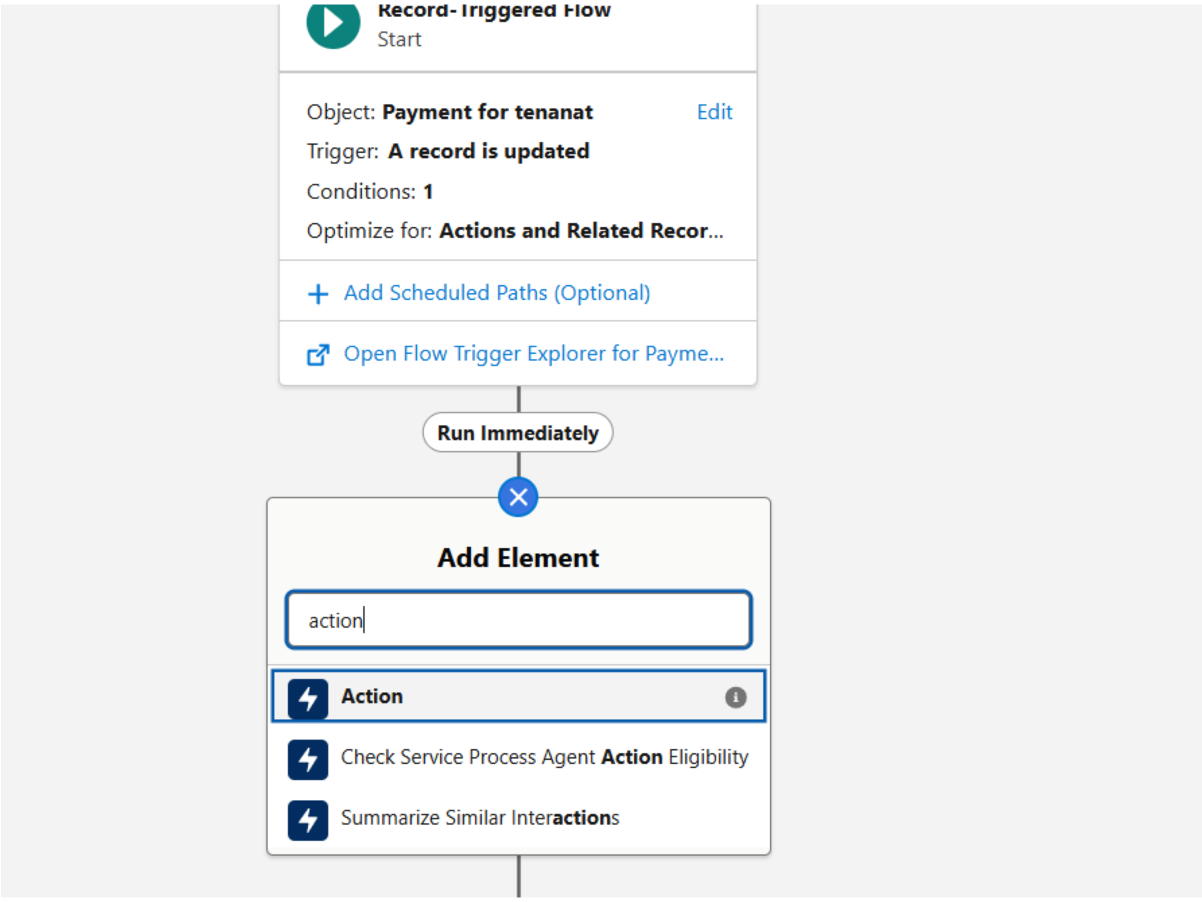


5. Click on: Every time a record is updated and meets the condition requirements

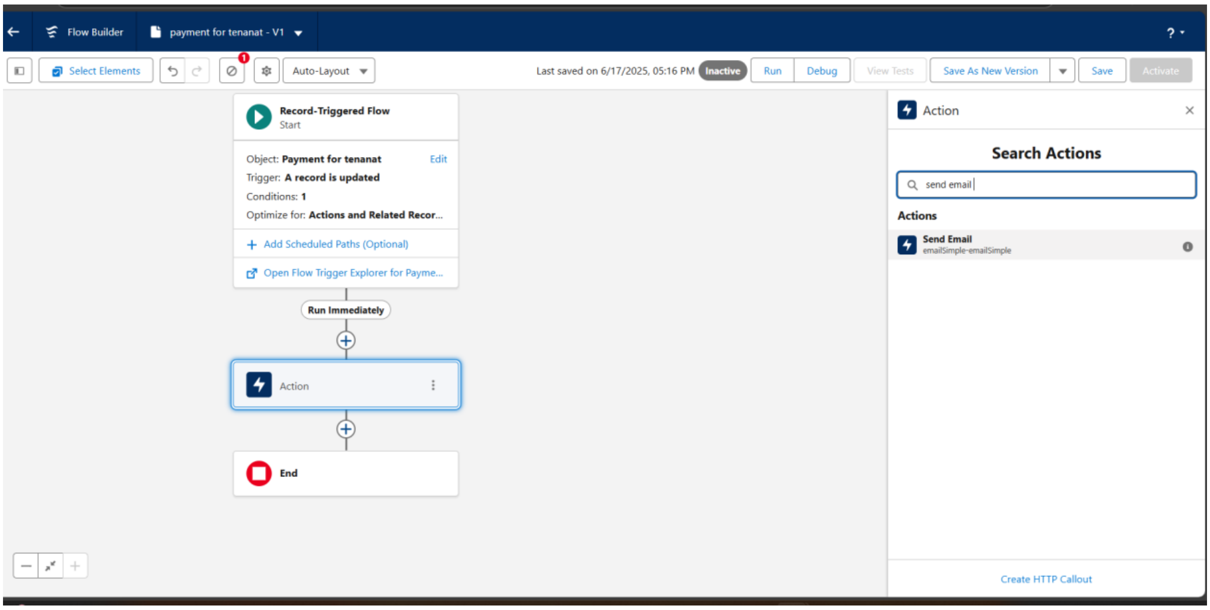
6. Click on: Actions and related records, done



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

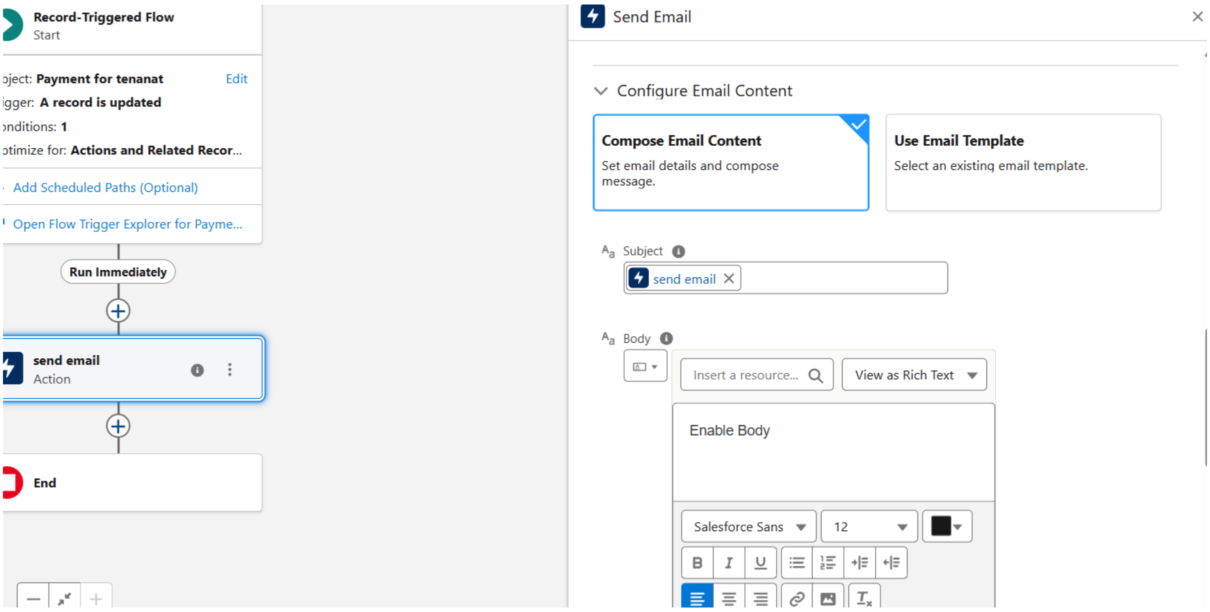


In action search for send email then click on send email (check below image)



8. Label: send email

API Name: send email



9. Label: send email

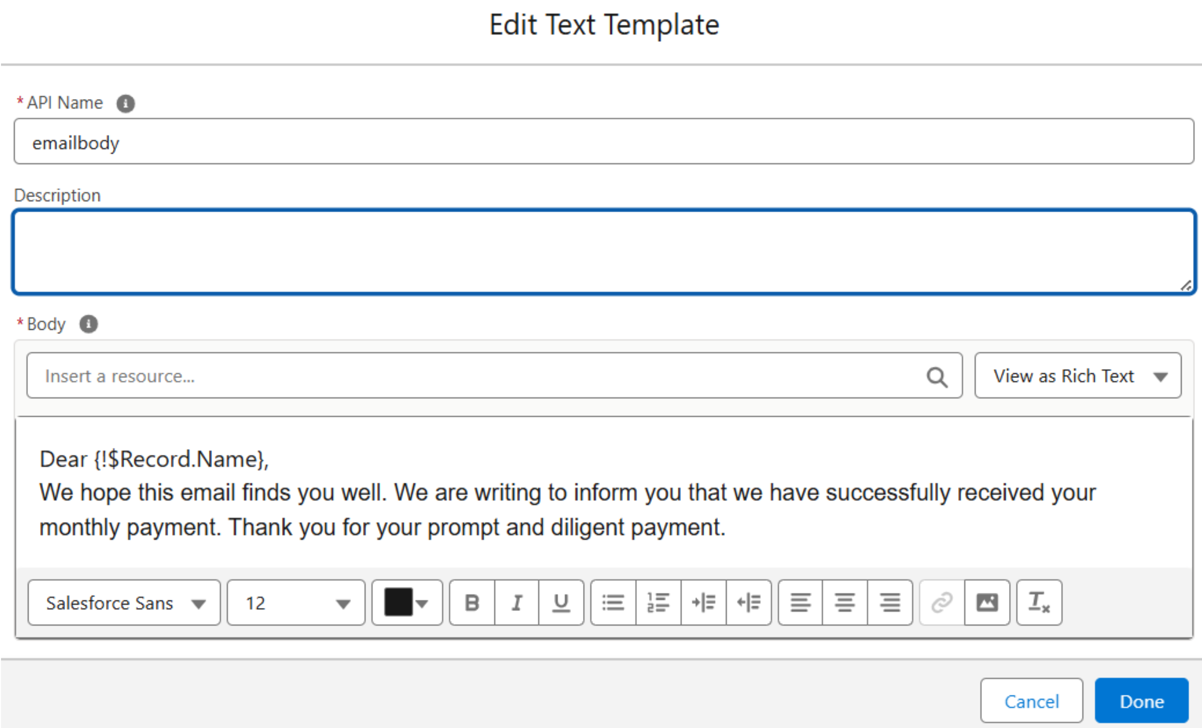
10. API Name: send email

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.Tenantr.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.



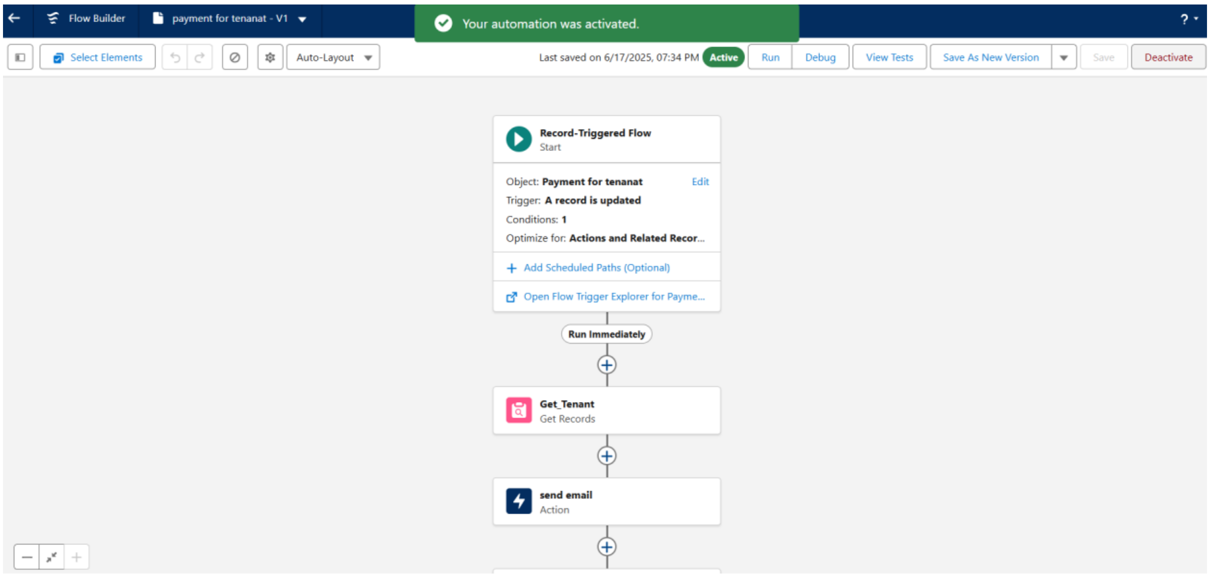
14. Click Done.
15. Enable recipient Address List Paste this?$Record Tenant_r.Email_c)
16. Click Done
17. Enable subject

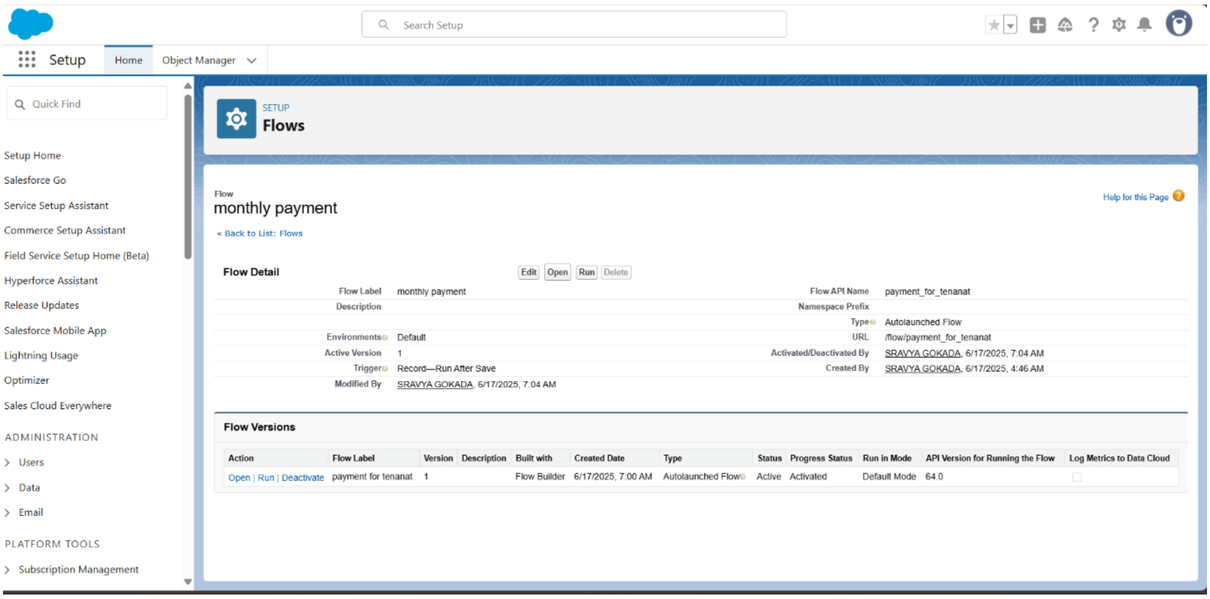
Paste this >> Confirmation of Successful Monthly Payment

18. Click on save

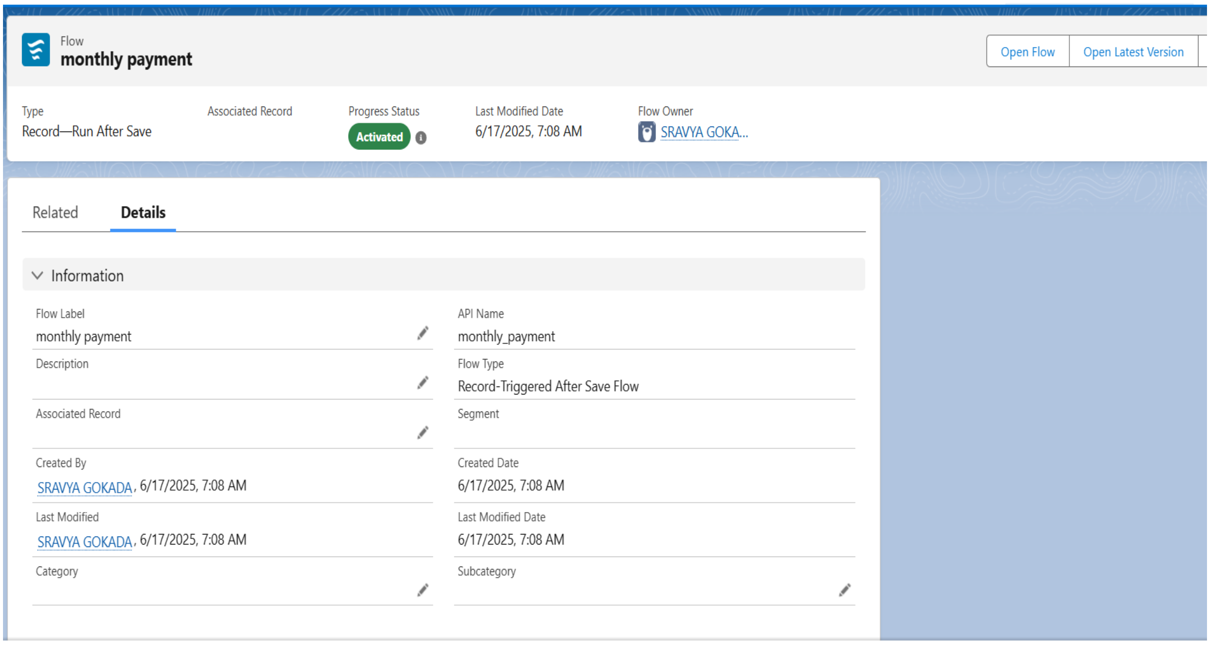
Flow label: monthly payment

Flow API Name monthly payment and Click on activate





Search Setup
Setup
Home
Object Manager
Quick Find
Setup Home
Salesforce Go
Service Setup Assistant
Commerce Setup Assistant
Field Service Setup Home (Beta)
Hyperforce Assistant
Release Updates
Salesforce Mobile App
Lightning Usage
Optimizer
Sales Cloud Everywhere
ADMINISTRATION
Users
Data
Email
PLATFORM TOOLS
Subscription Management
SETUP
Flows
Flow
monthly payment
Help for this Page
« Back to List: Flows
Flow Detail
Edit
Open
Run
Delete
Flow Label monthly payment
Flow API Name payment_for_tenanat
Description
Namespace Prefix
Type Autolaunched Flow
Environments Default
URL /flow/payment_for_tenanat
Active Version 1
Activated/Deactivated By SRAVYA GOKADA, 6/17/2025, 7:04 AM
Trigger Record—Run After Save
Created By SRAVYA GOKADA, 6/17/2025, 4:46 AM
Modified By SRAVYA GOKADA, 6/17/2025, 7:04 AM
Flow Versions
Action | Flow Label | Version | Description | Built with | Created Date | Type | Status | Progress Status | Run in Mode | API Version for Running the Flow | Log Metrics to Data Cloud
Open | Run | Deactivate | payment for tenanat | 1 | Flow Builder | 6/17/2025, 7:00 AM | Autolaunched Flow | Active | Activated | Default Mode | 64.0



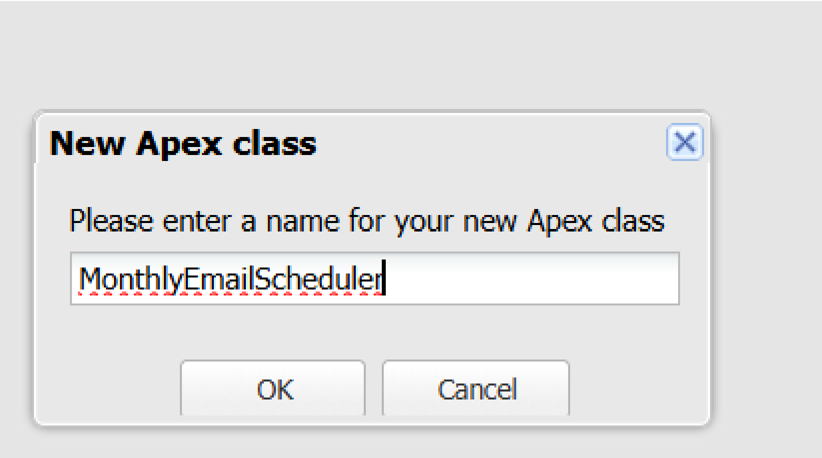
Flow
monthly payment
Open Flow
Open Latest Version
Type
Record—Run After Save
Associated Record
Progress Status
Activated
Last Modified Date
6/17/2025, 7:08 AM
Flow Owner
SRAVYA GOKA...
Related
Details
Information
Flow Label
monthly payment
API Name
monthly_payment
Description
Flow Type
Record-Triggered After Save Flow
Associated Record
Segment
Created By
SRAVYA GOKADA, 6/17/2025, 7:08 AM
Created Date
6/17/2025, 7:08 AM
Last Modified
SRAVYA GOKADA, 6/17/2025, 7:08 AM
Last Modified Date
6/17/2025, 7:08 AM
Category
Subcategory

Schedule class:

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.



Apex logic:

```
global class MonthlyEmailScheduler implements Schedulable {
        global void execute(SchedulableContext sc)  {
                    Integer currentDay-Date.today) day!);

                if (currentDay===1)  {

                        send Monthly Emalist[];

    }

}

public static void send Monthly Emails() {

     List Tenant c> tenants = [SELECT id, Email_c FROM Tenant_cl: for (Tenant_c tenant:
tenants) (

    String recipient Email = 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 email Subject = "Reminder: Monthly Rent Payment Due': Messaging
SingleEmailMessage email = new
```

Messaging Single Email Message();

email.setToAddresses(new String[][recipient Email]):

emailsetSubject(emailSubject);

email.setPlainTextBody(emailContent):

Messaging.sendEmailinow Messaging Single Email Messagell

(email):

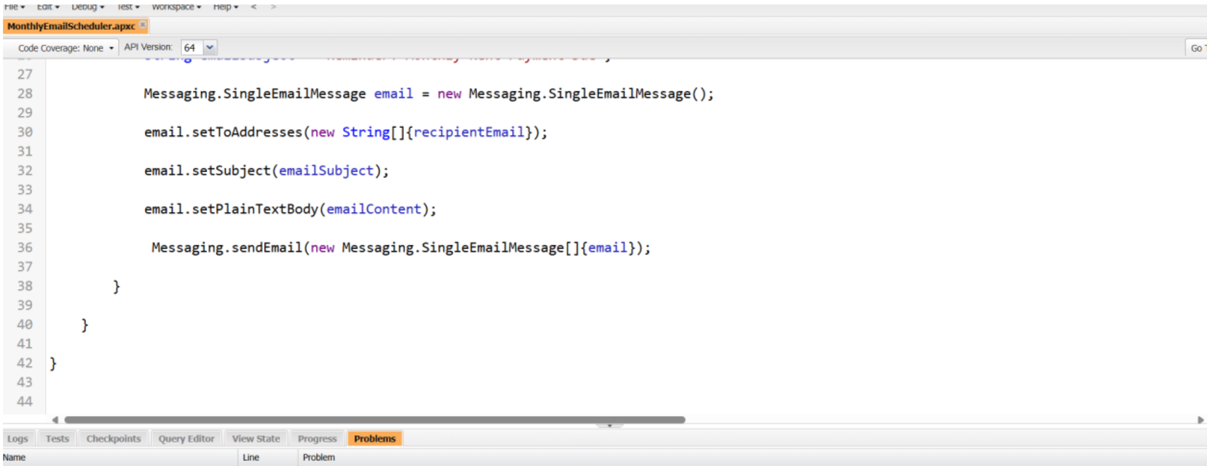
}

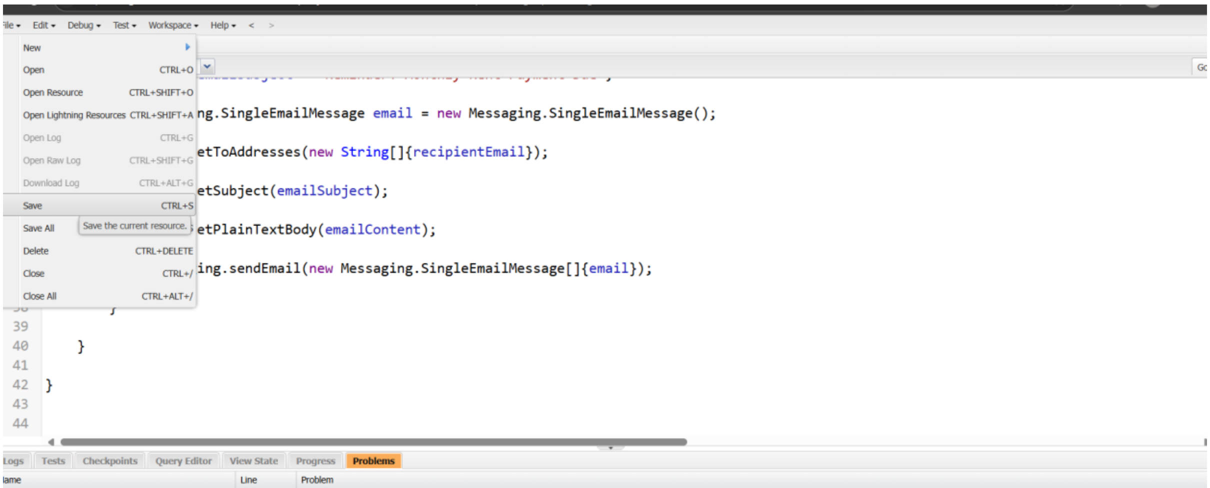
}

Save the code.

```
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 paymen

            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});
```

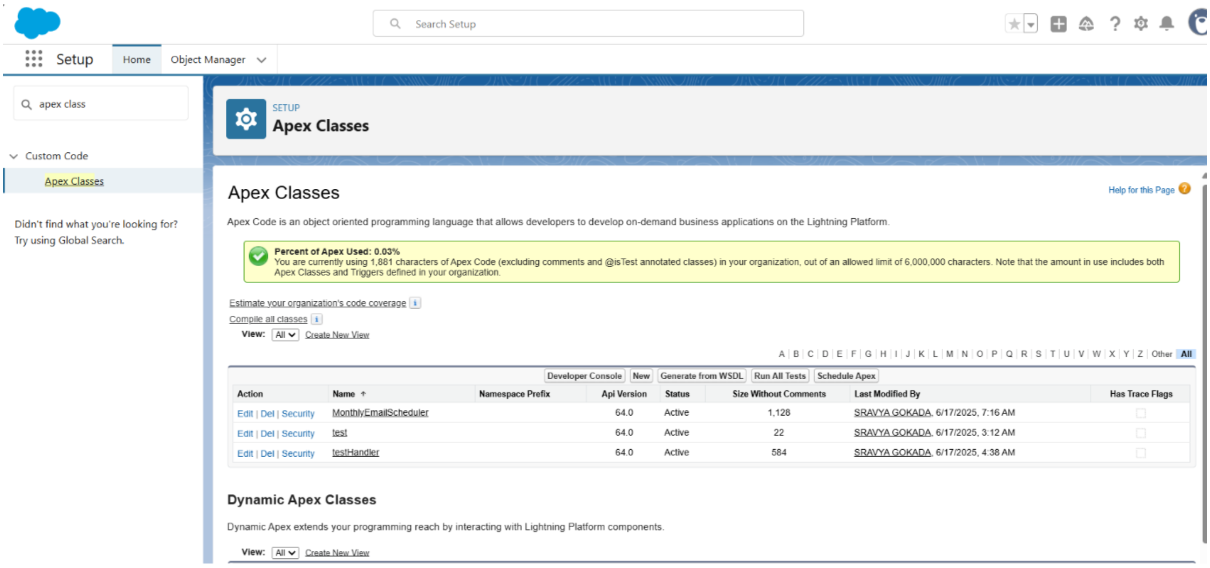




ttps://orgfarm-71e75fd7db-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage#

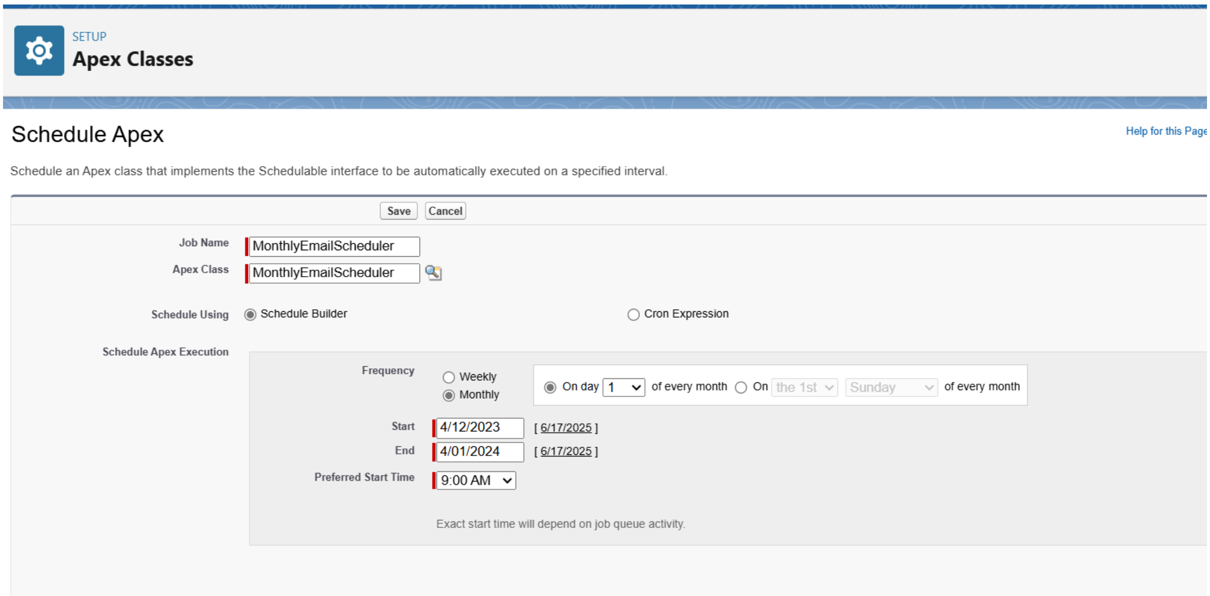
Schedule Apex Class

1. Enter Apex class in quick find box
2. Select schedule Apex



3. Enter job Name: Monthly EmailScheduler
 4. Apex class: Monthly Email Scheduler

- 5. Frequency: Monthly===>select on day 1



6. Start date:

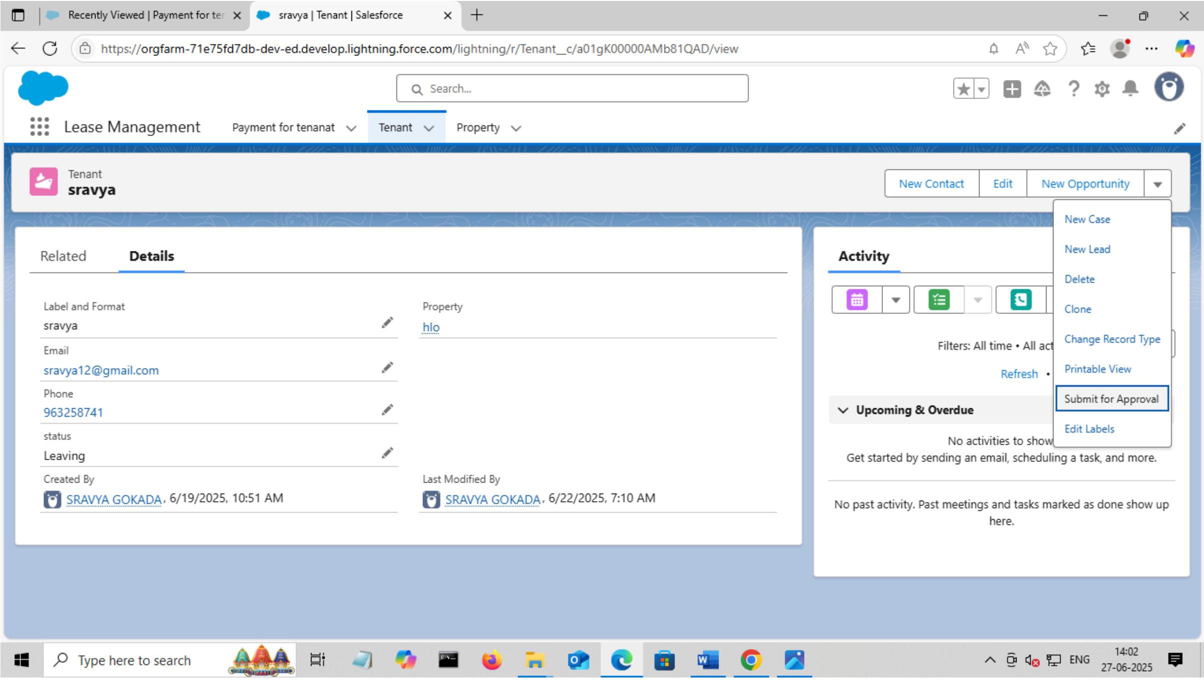
7 End date

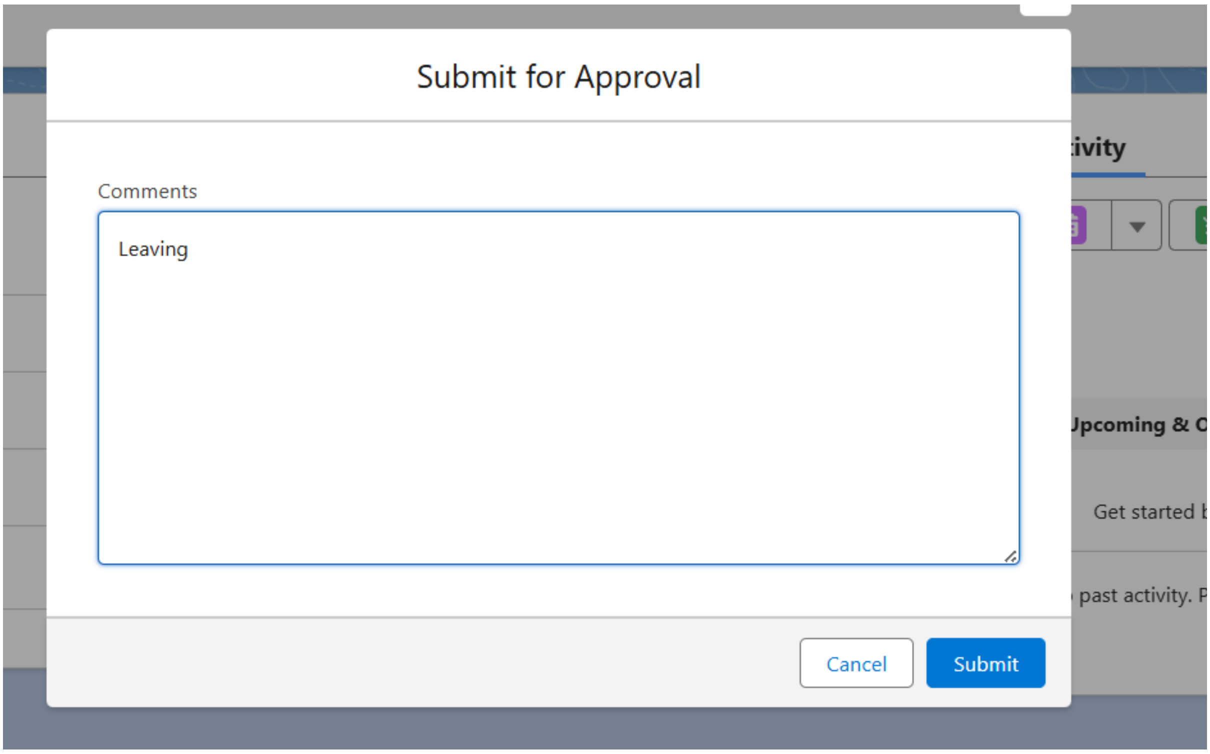
8. Preferred start time

- Testing the approval process

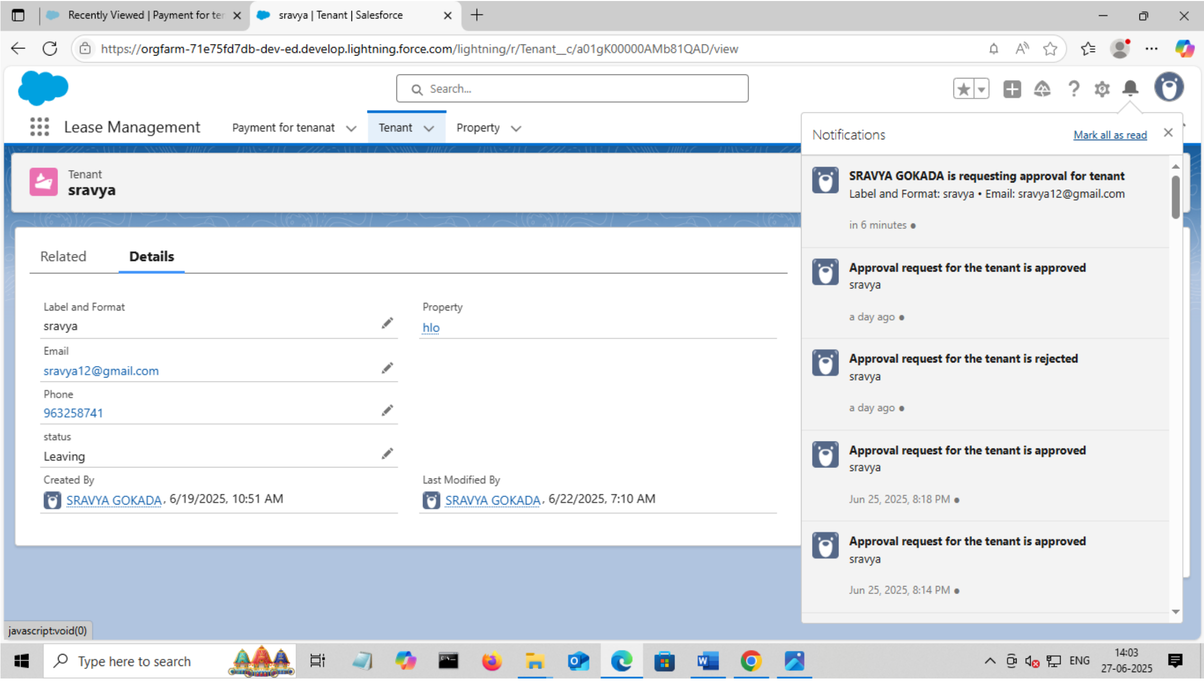
Enter any comment am\nd click on submit

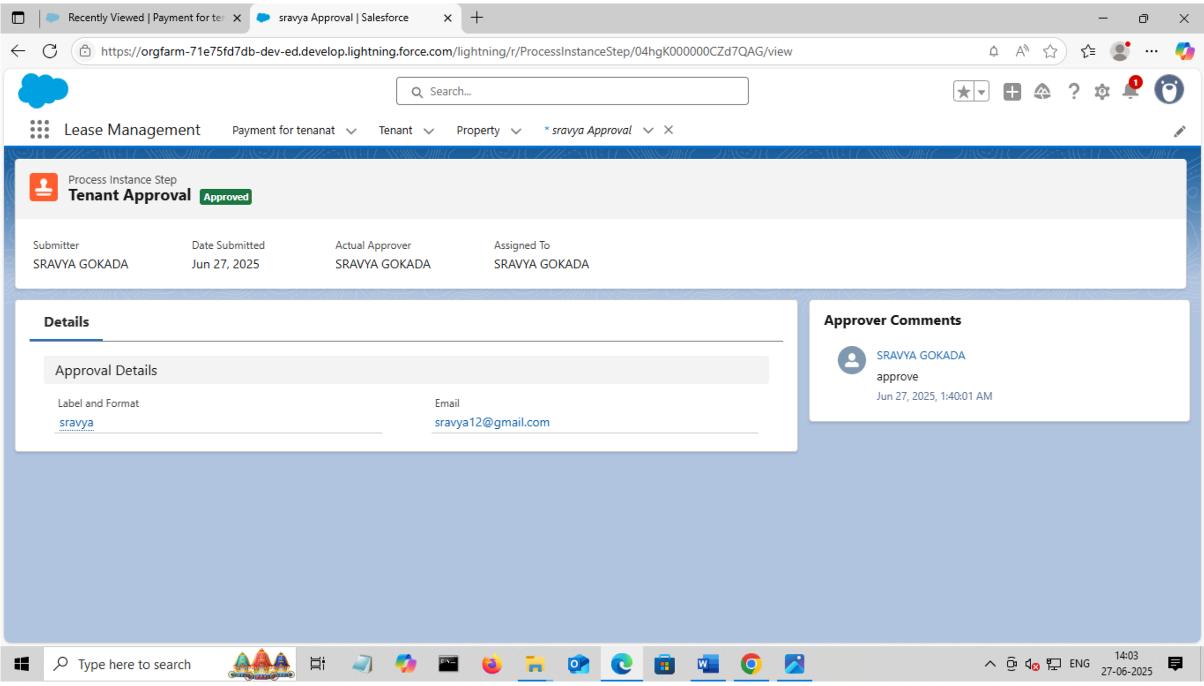
- Click on the notification

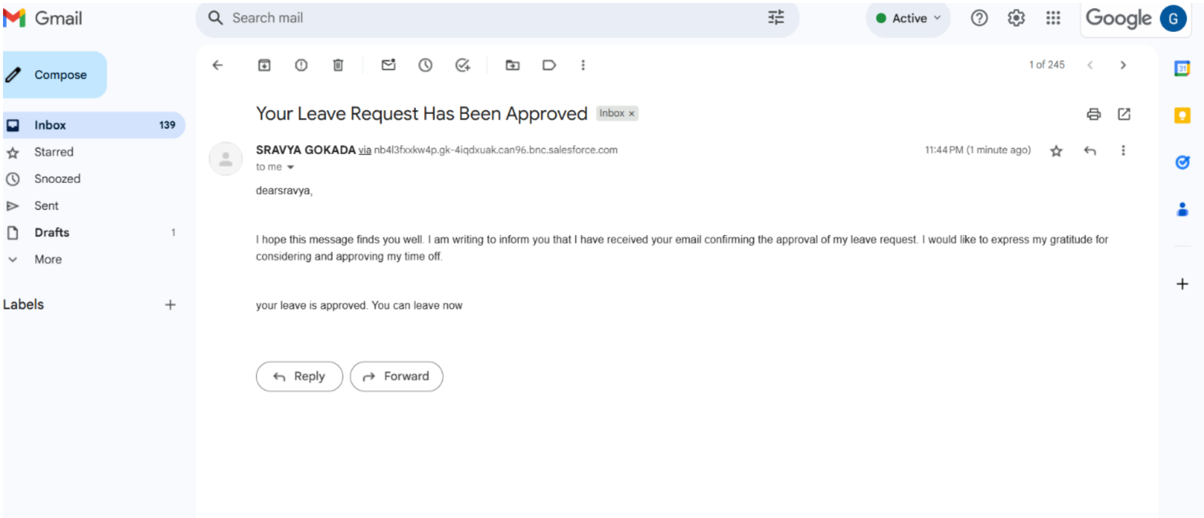




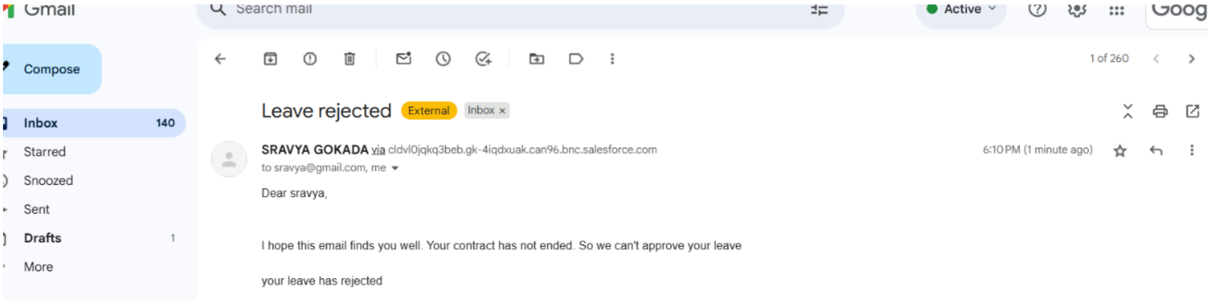
You will find notification like this and you will get an email check

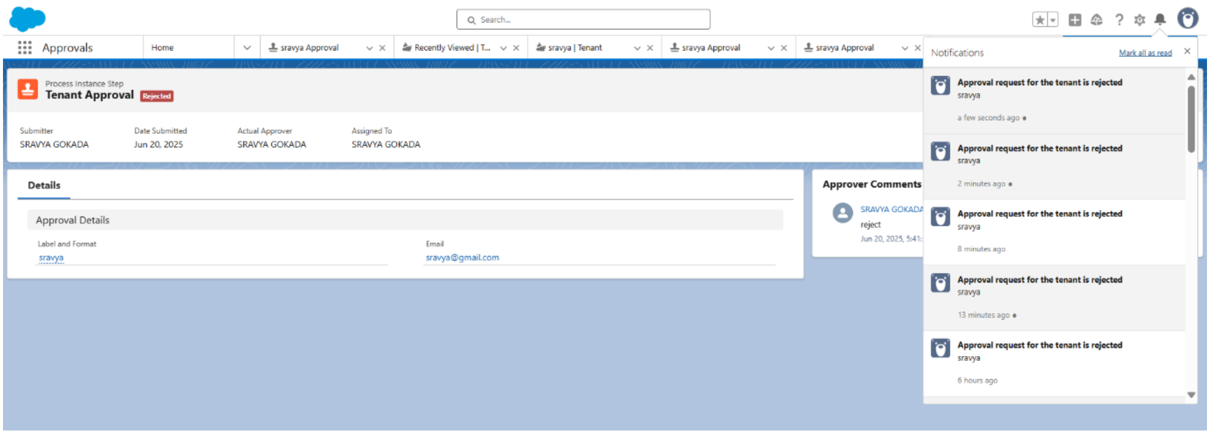
- 

- 



Note : Similarly do for reject also you will get mail and notification





CONCLUSION:

The Lease Management project in Salesforce is a comprehensive solution that streamlines lease-related processes, providing a centralized platform for managing lease agreements, rent payments, and lease terms. By leveraging Salesforce's capabilities, the project offers numerous benefits, including:

- Improved efficiency and productivity
- Enhanced visibility and reporting
- Better compliance and risk management
- Scalability and flexibility

While there are potential challenges and disadvantages, careful planning, user training, and ongoing support can mitigate these issues. Overall, the Lease Management project in Salesforce has the potential to transform lease management processes, enabling organizations to optimize their lease portfolios and improve business outcomes.

Key Takeaways:

- Lease Management in Salesforce offers a robust solution for managing lease agreements and related processes.
- The project requires careful planning, implementation, and ongoing support to ensure success.
- By leveraging Salesforce's capabilities, organizations can improve efficiency, visibility, and compliance in lease management.

Future Directions:

- Continuously monitor and evaluate the system's performance to identify areas for improvement.
- Explore opportunities to integrate the Lease Management system with other Salesforce applications and external systems.

- Stay up-to-date with changing regulations and industry best practices to ensure the system remains compliant and effective.