Government Arts & Science Collage Anthiyur-638311

Department of Computer Science

Salseforce Agent Blazer Champion

To Lease management

Team Leader : Kavin K

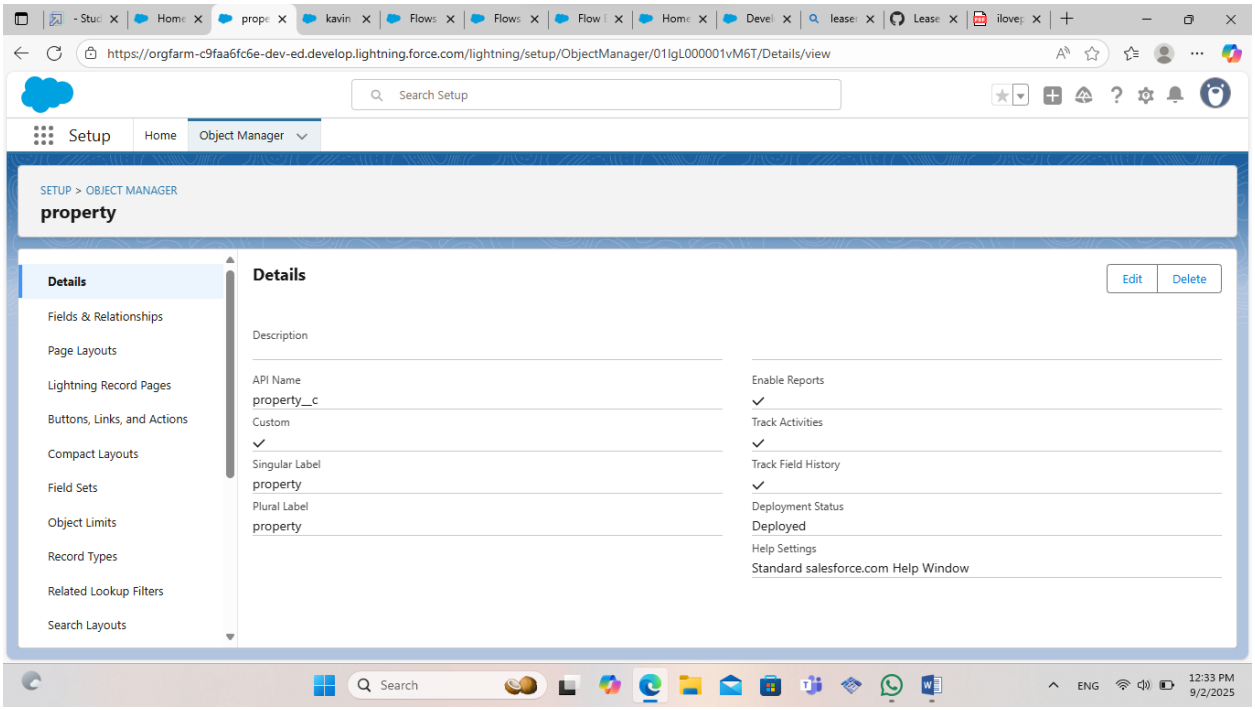
Team Meambers :

Ramana R
Sasi S
Saravanakumar S

Object

To create an object:

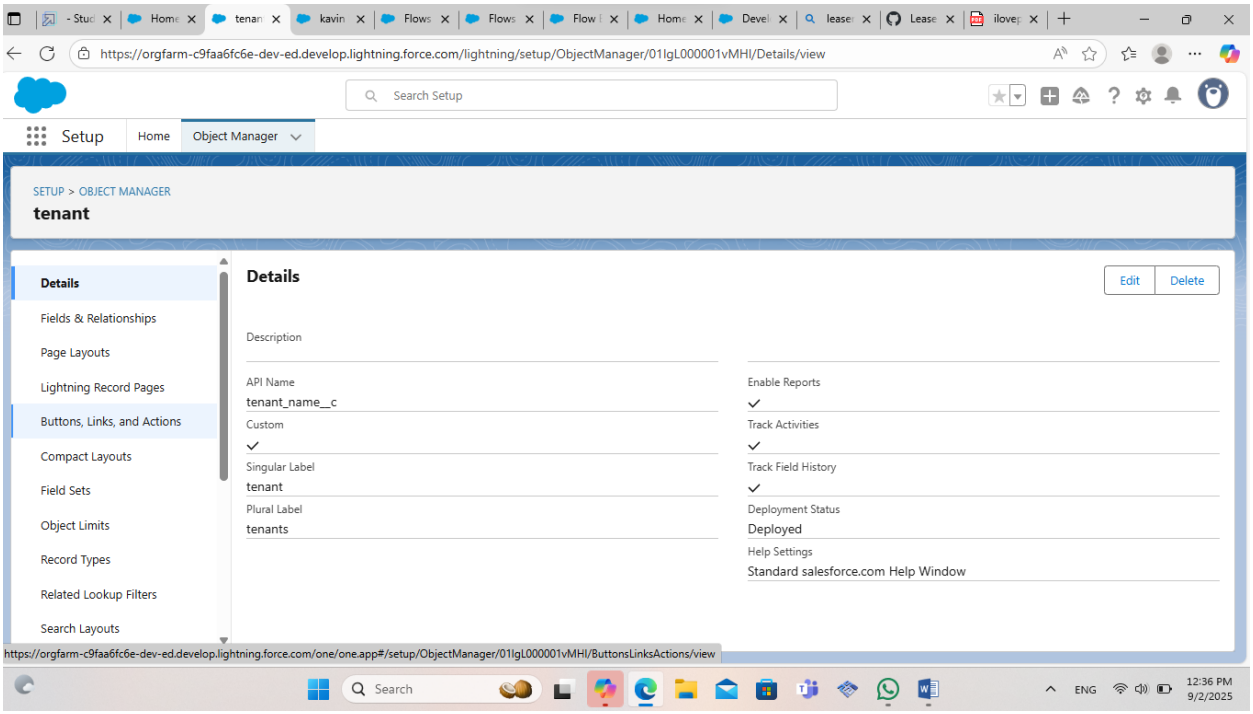
1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
    1. Enter the label name>> property
    2. Plural label name>> property
    3. Enter Record Name Label and Format
        - Record Name >>property Name
        - Data Type >> Text
2. Click on Allow reports and Track Field History,Allow Activities
3. Allow search >> Save.



Create Tenant 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>> 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.



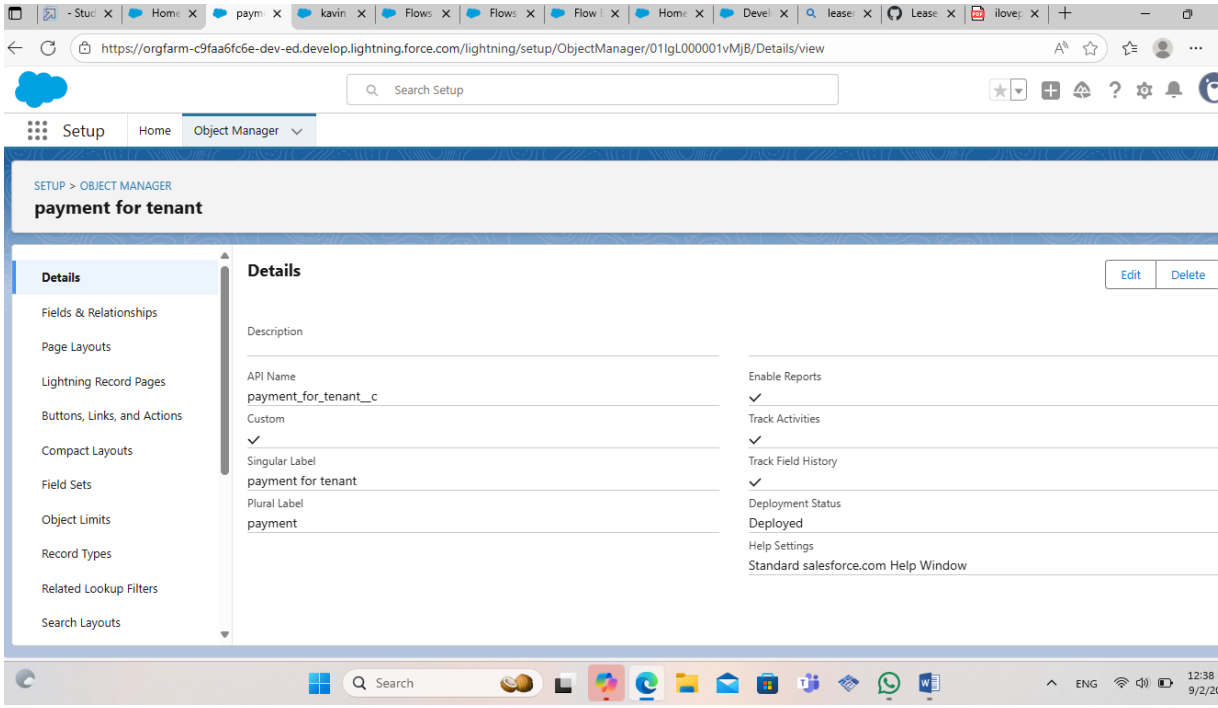
1. Create Payment Object

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

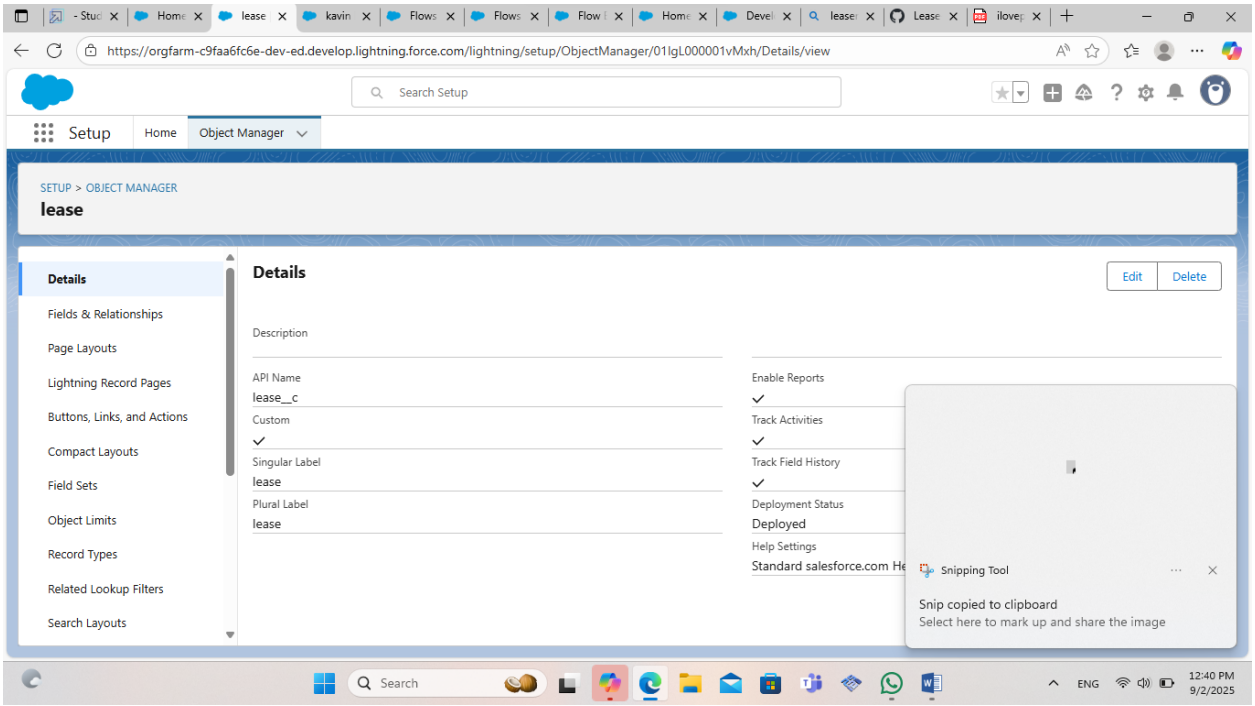
Allow search >> Save.



2. 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 Activities
3. Allow search >> Save.



3. 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:

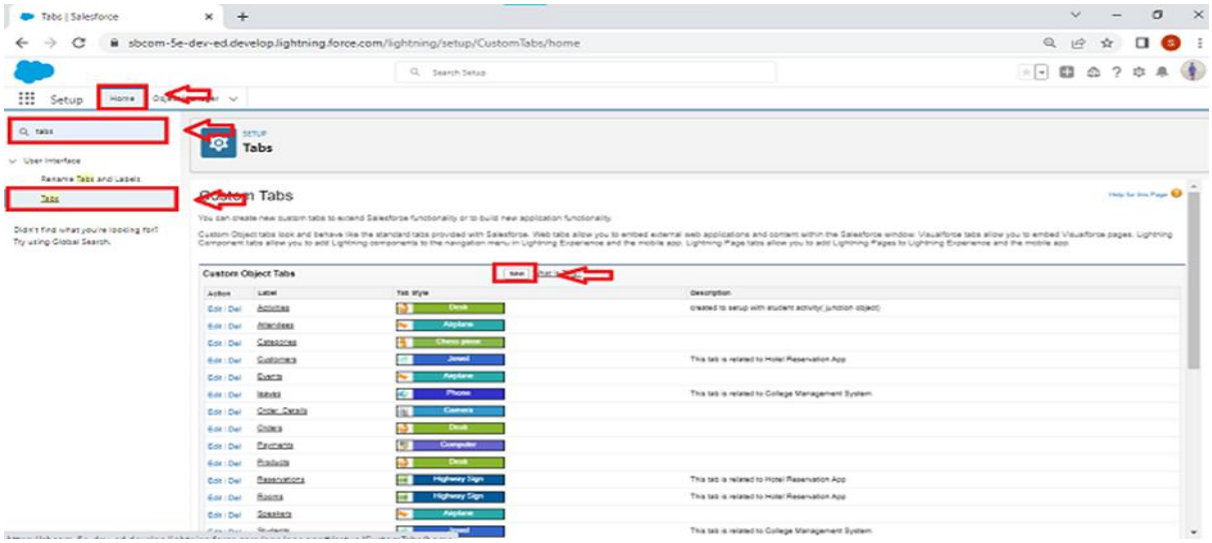
1. Custom Tabs
   Custom object tabs are the user interface for custom applications that you build in salesforce.com. They look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.
   2. Web TabsWeb Tabs are custom tabs that display web content or applications embedded 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.
1. Visualforce Tabs
   Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce 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.
3. Lightning Page Tabs
   Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.
   Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

-
  -
  -
    -
    -
    -
    -
  -
    -
    -
  -
  -
  -
  -
  -
  -
  -
  -
    -
    -

4. Creating a Custom Tab

To create a Tab:(Property)

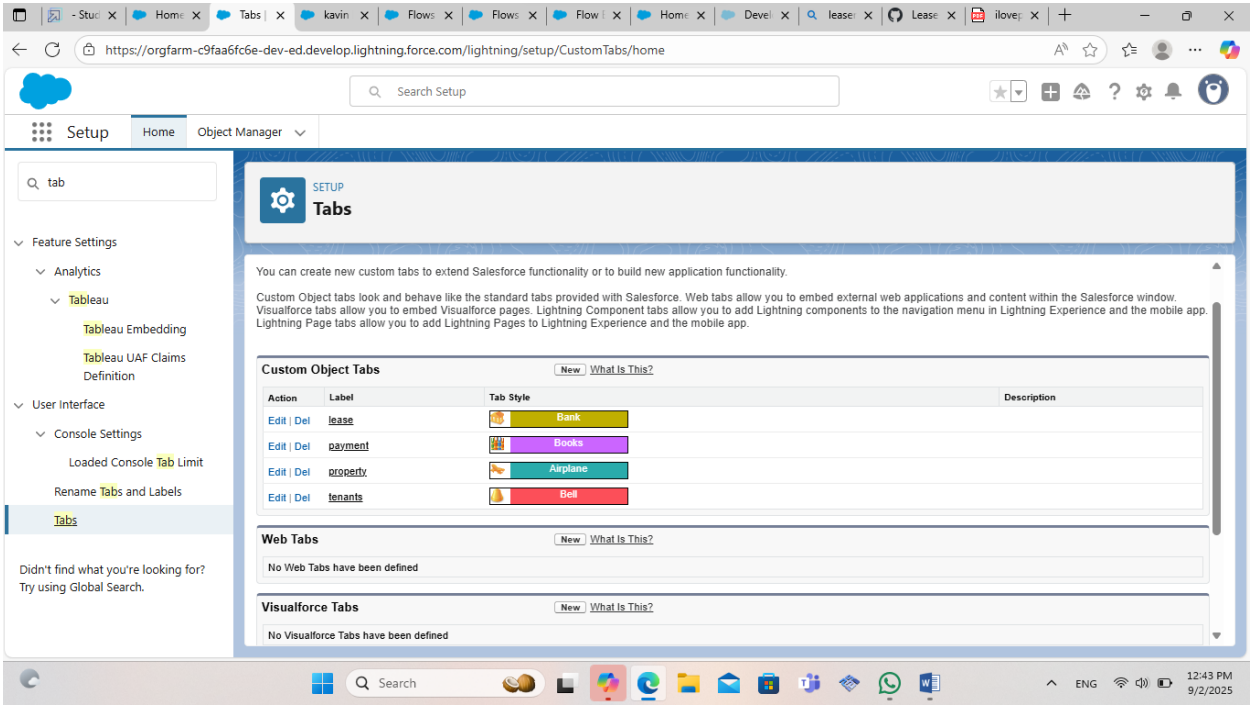
1. Go to setup page >>type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)



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

5. Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are “Payment for tenant,lease,tenant”.
2. Follow the same steps as mentioned in Activity -1 .



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.

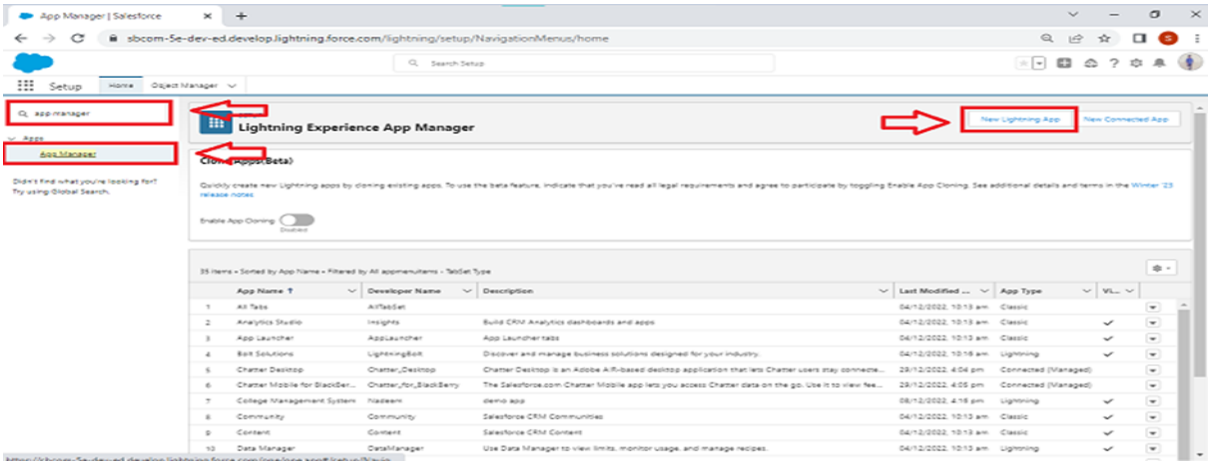
Use Case:

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.

6. Create a Lightning App

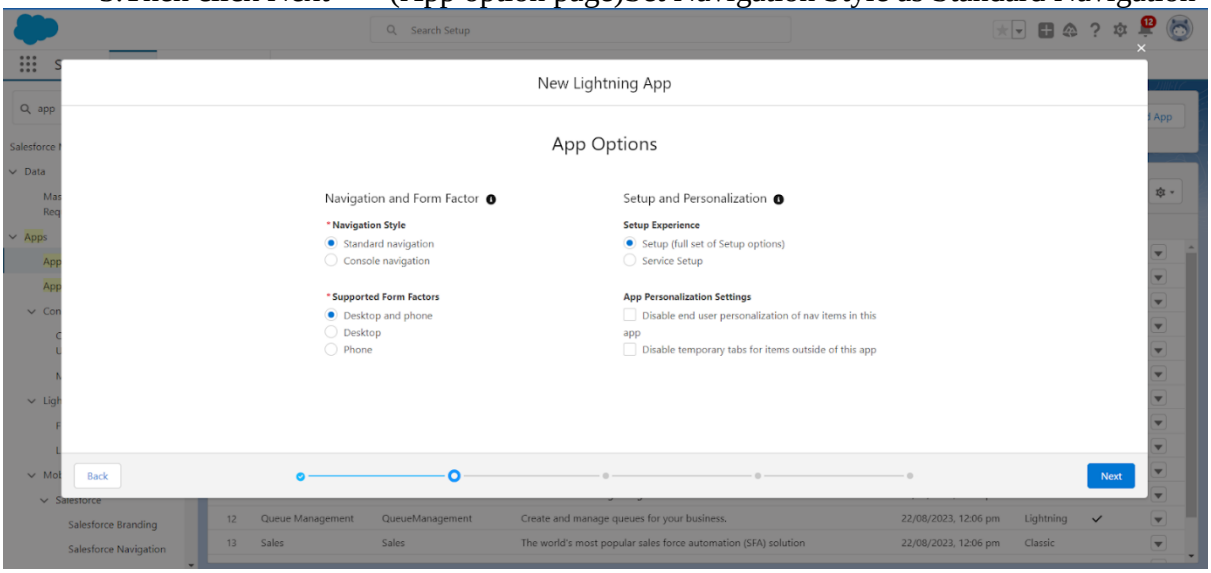
To create a lightning app page:

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

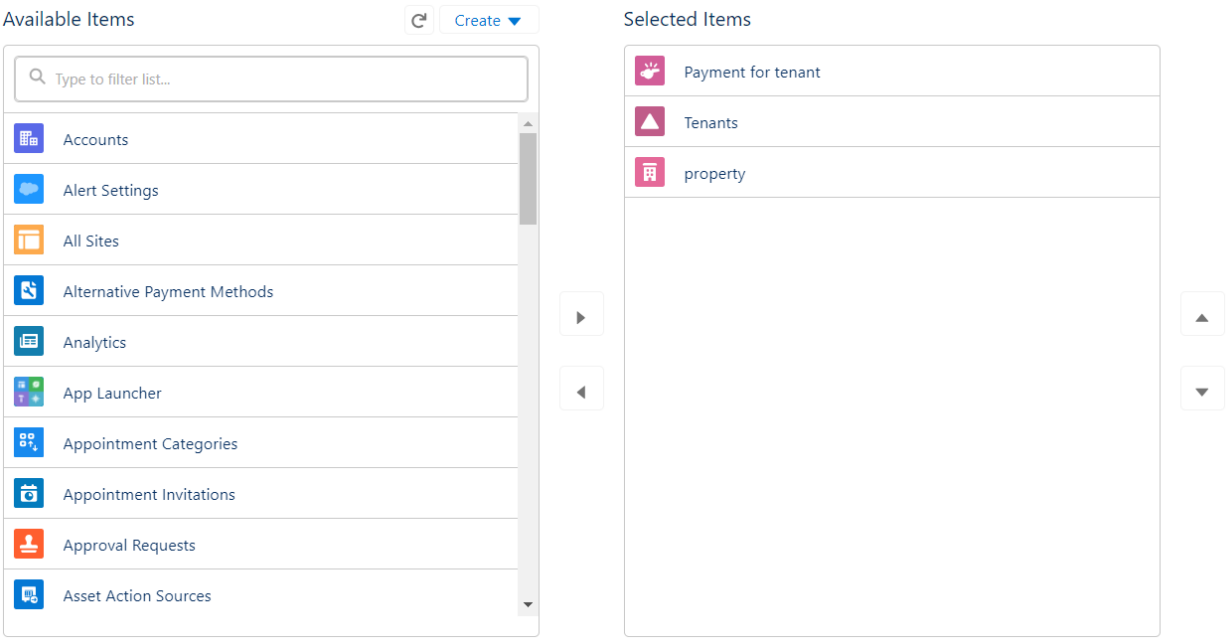


2. Fill the app name in app details and branding as follow
App Name : Lease ManagementDeveloper Name : This will auto populatedImage : optional (if you want to give any image you can otherwise not mandatory) Primary colour hex value : keep this default.

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

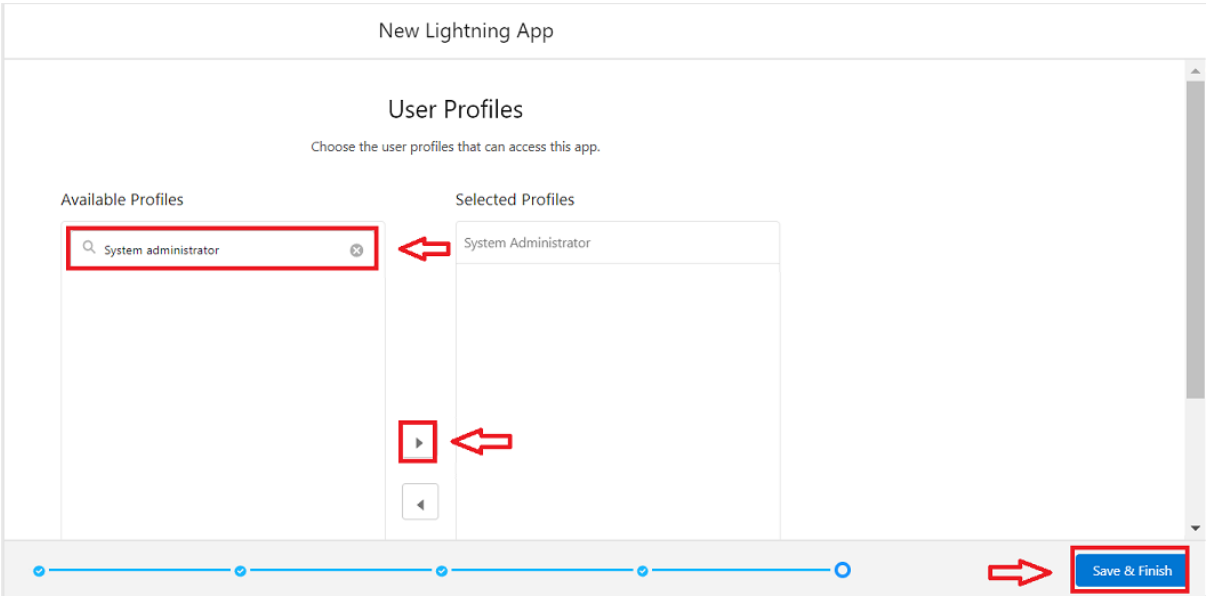


(Utility Items) keep it as default >> Next.
5. To Add Navigation Items:



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:



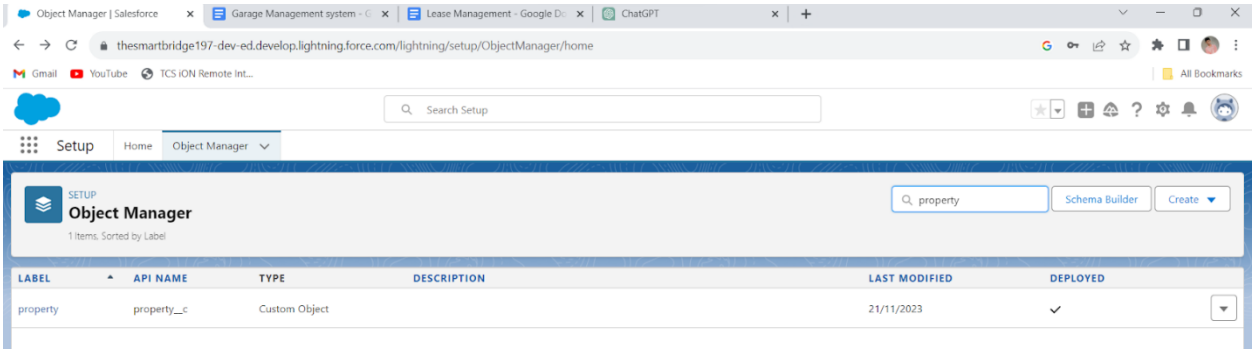
Search profiles (System administrator) in the search bar >>click on the arrow button >> save & finish.

FIELDS

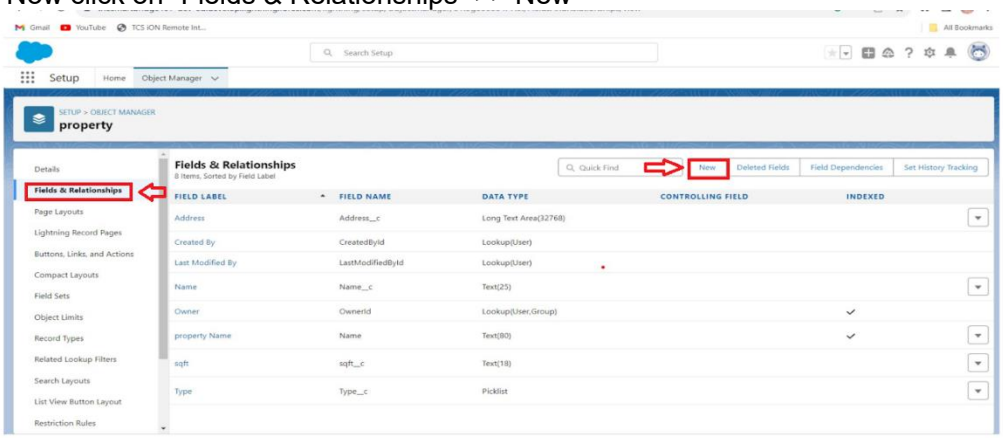
7. Creation of fields for the property object

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 a “Text”

8. 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.

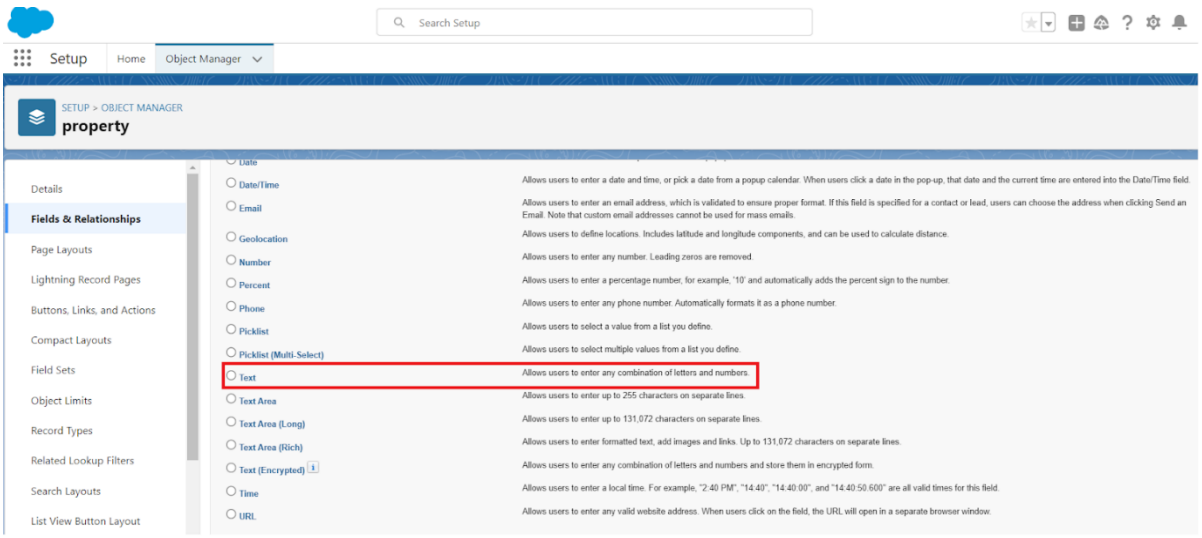
To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Tenant) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “phone” and Click on Next
4. Fill the Above as following:
   - Field Label : Phone
   - Field Name : gets auto generated
   - Click on Next >> Next >> Save and new.

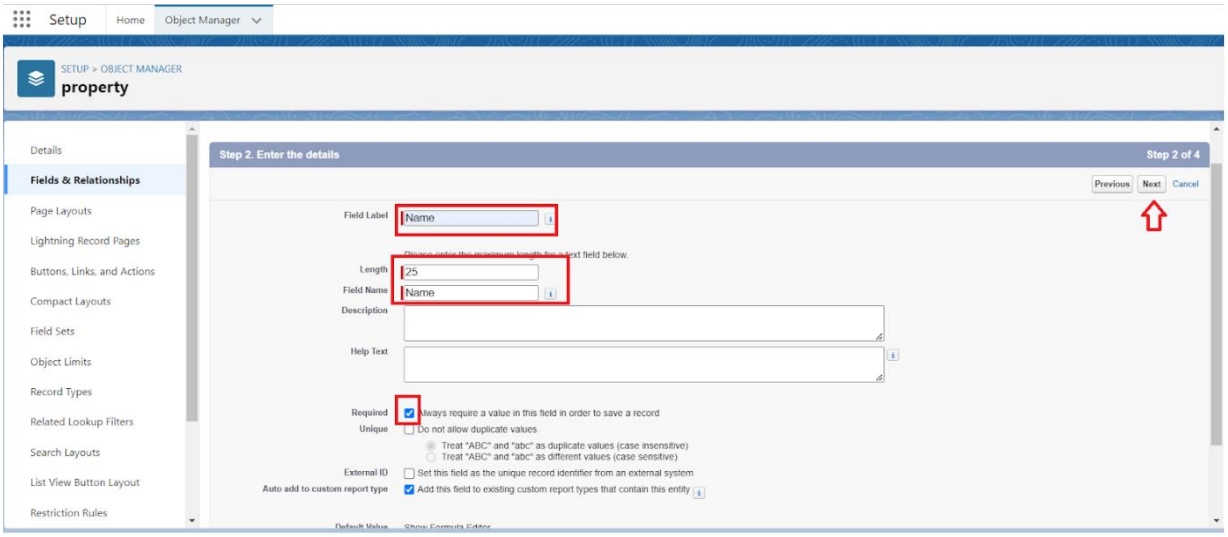
To create another fields in an object:

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

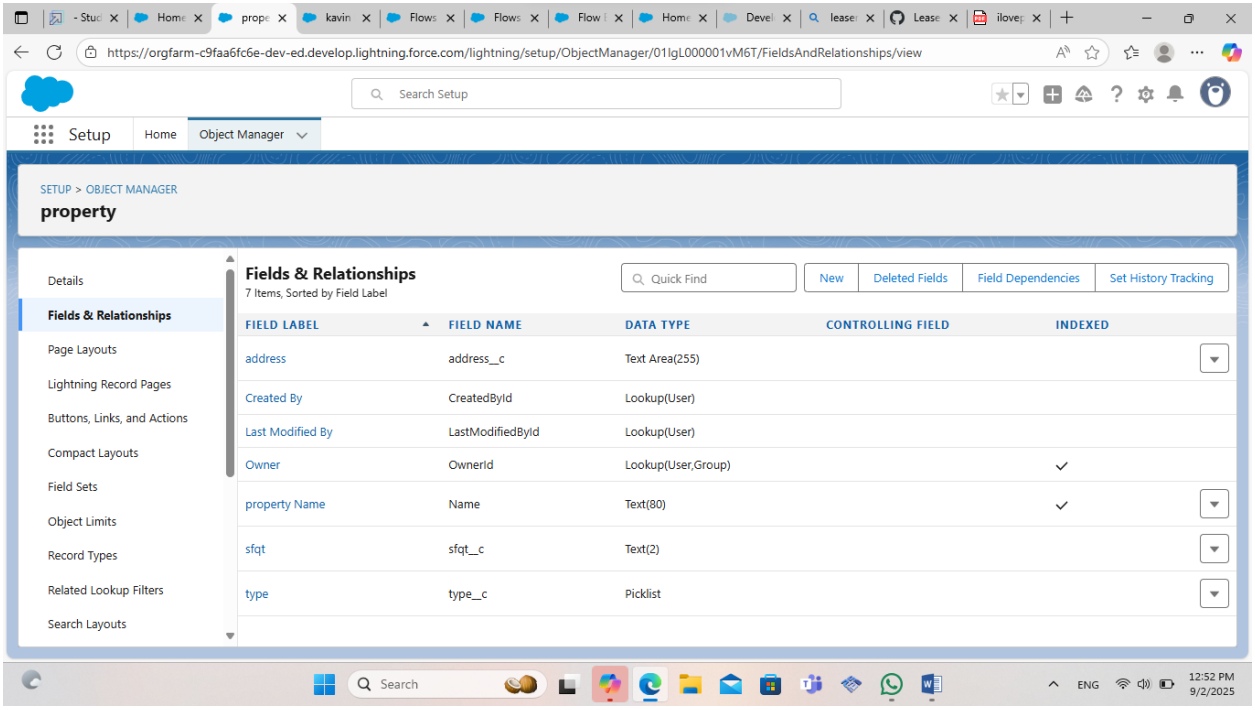


4. Click on next



5. 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.



- 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.

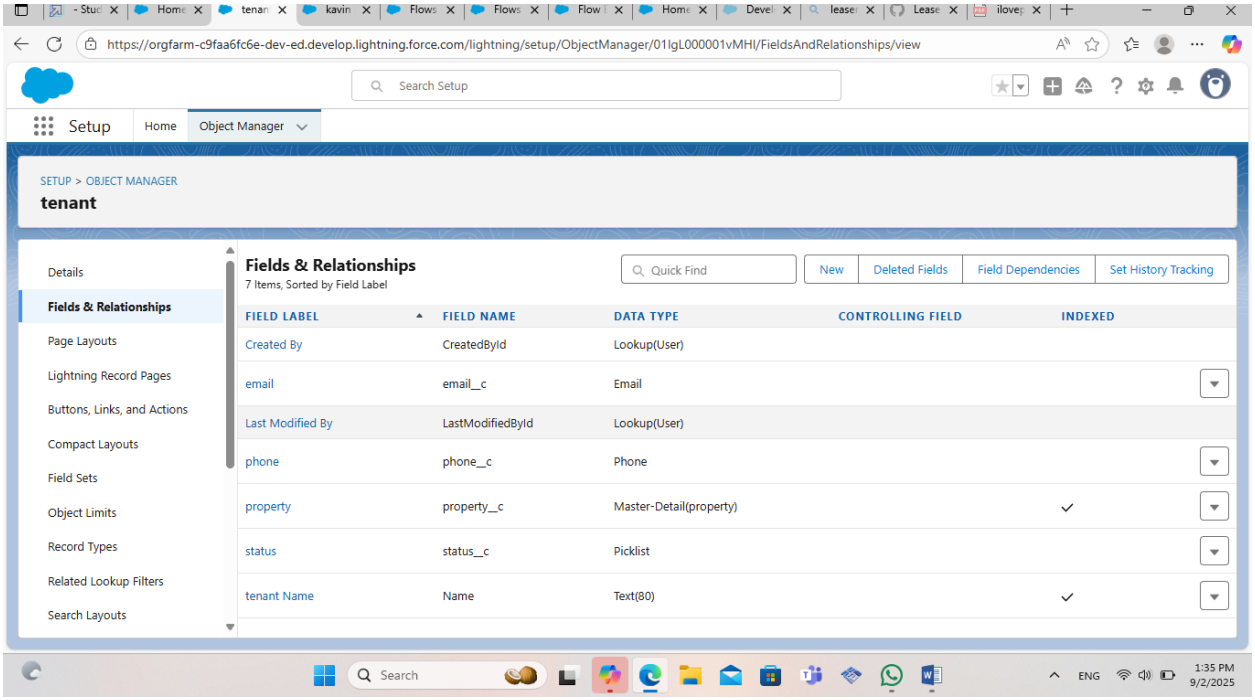
2. To create another fields in an object:

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

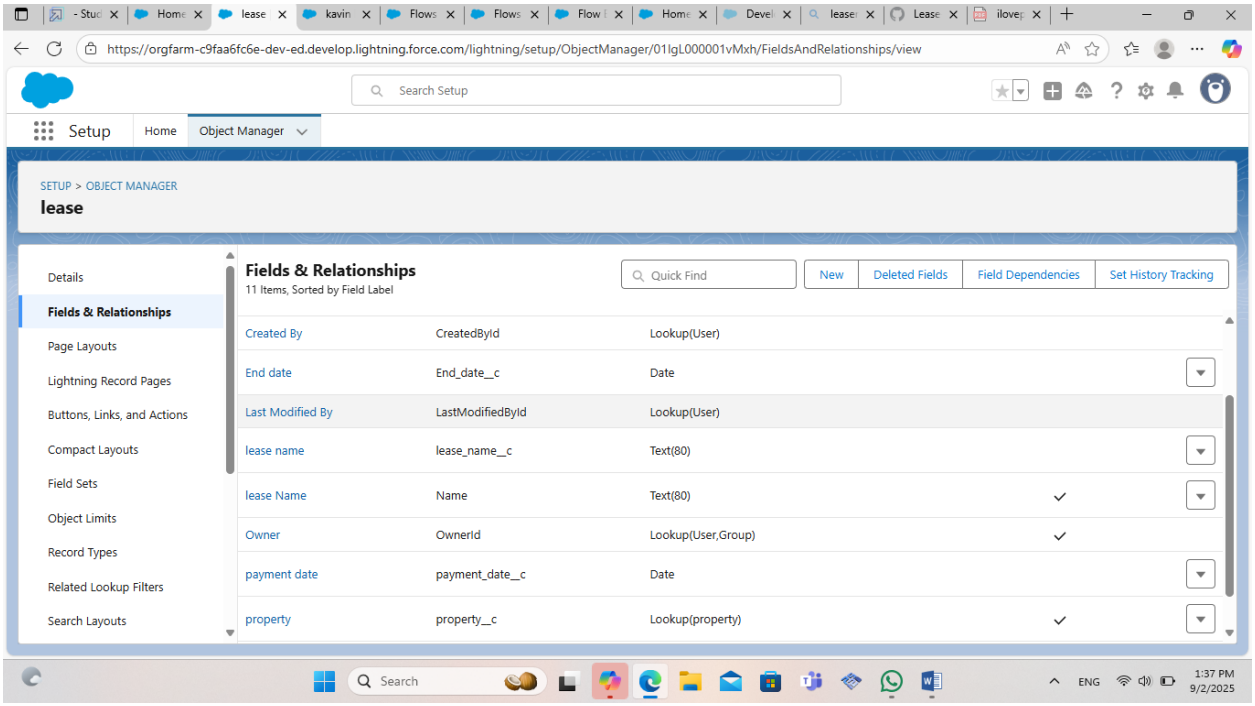


9. Creation of fields for the Lease object

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. 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.

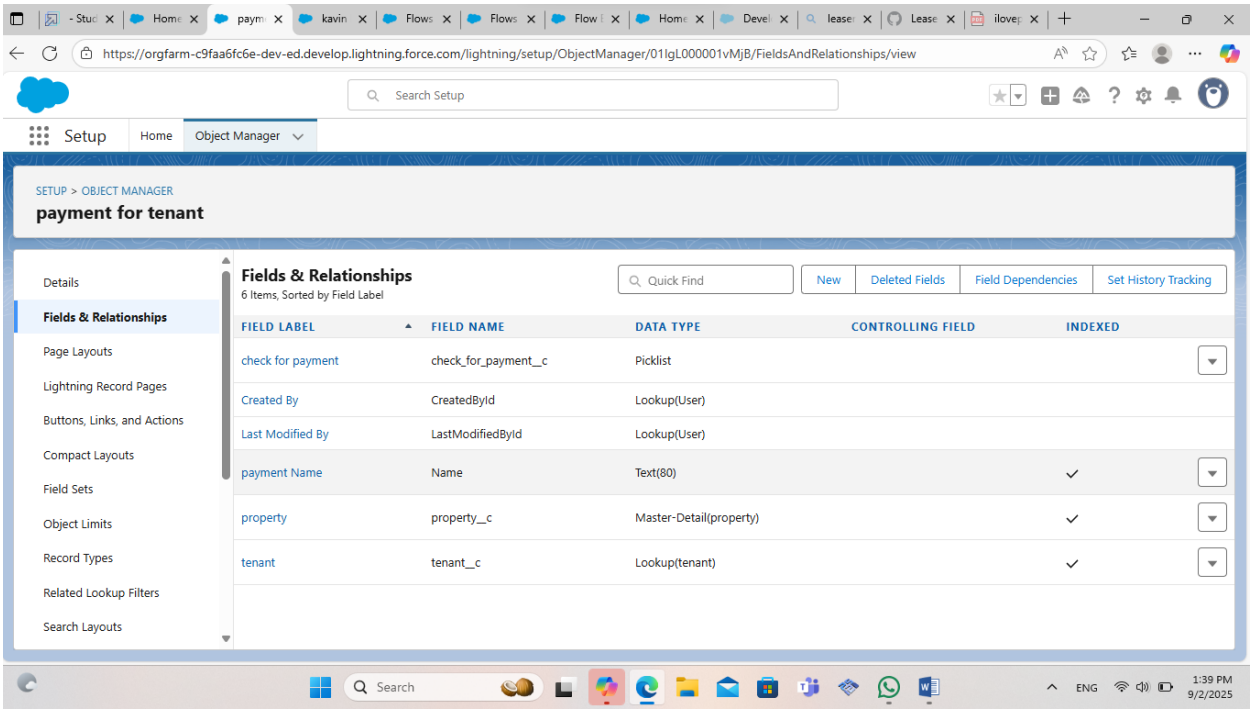


10. Creation of fields for the Payment for tenant object

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
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.

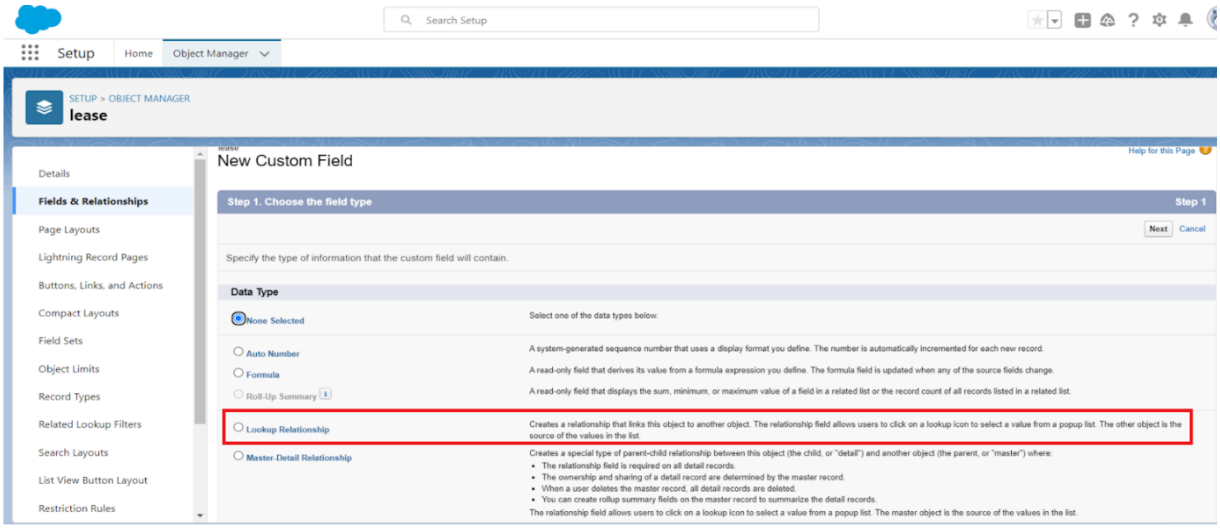
To create another fields in an object:
1.Go to setup >> click on Object Manager >> type object name(Payment for tenant) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “picklist” and Click on Next
4. Fill the Above as following:
   - Field 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.



11. 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.



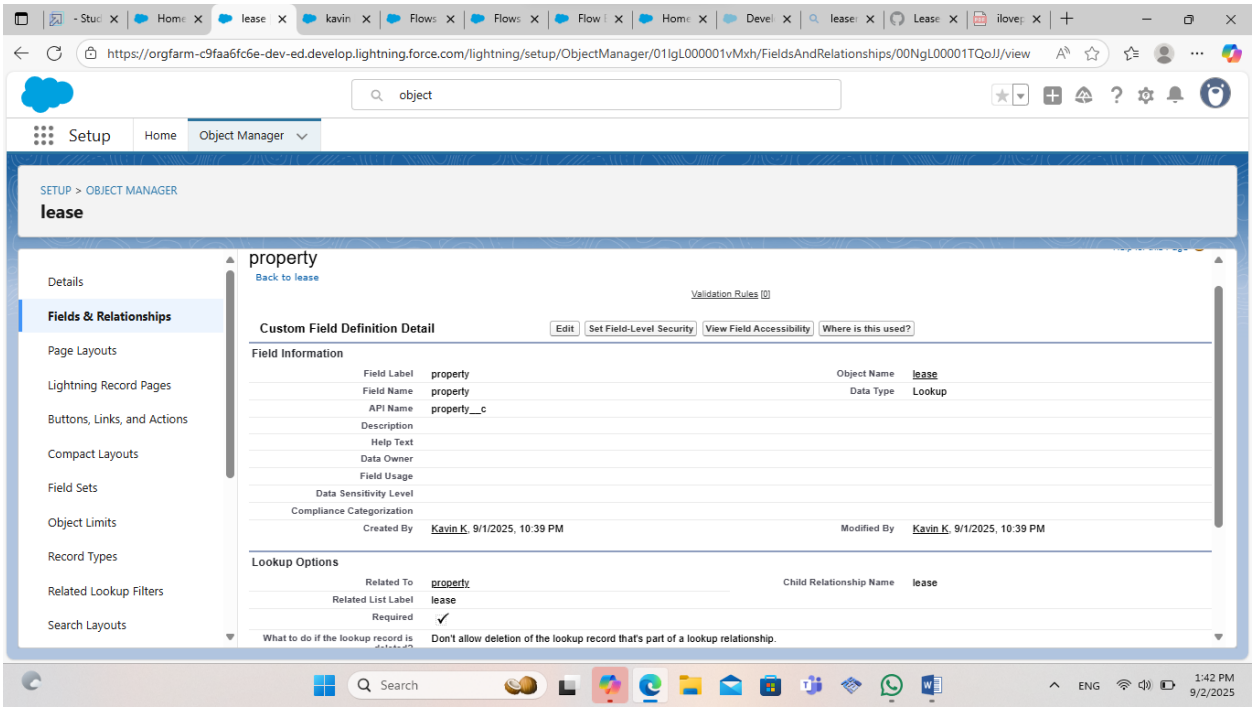
2. Now click on “Fields & Relationships” >> New
3. Select lookup relationship
4. Select the related object “ property” and click next.
5. Field Name : property
6. Field label : Auto generated
7. Next >> Next >> Save.

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.

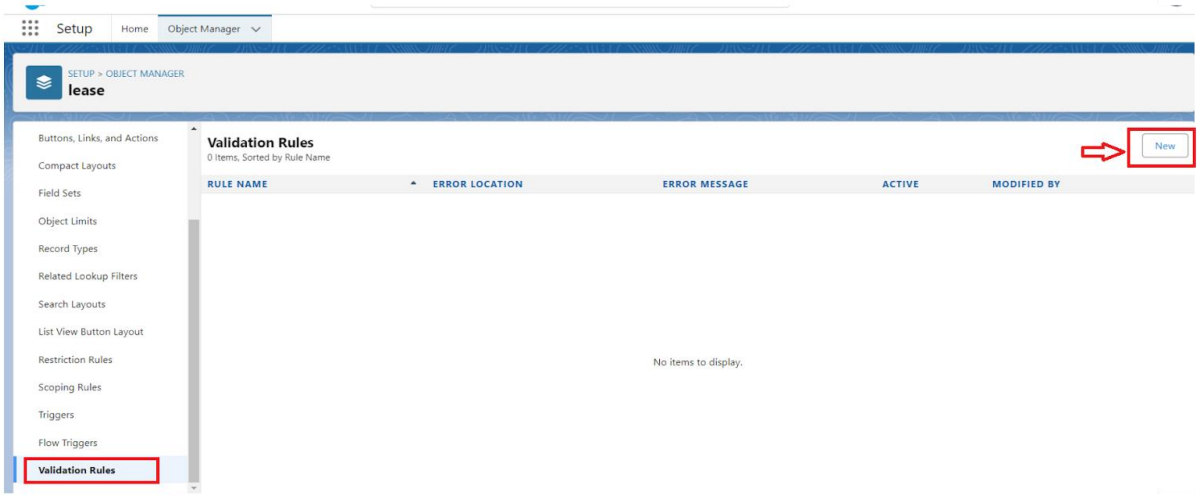


12. Validation rule

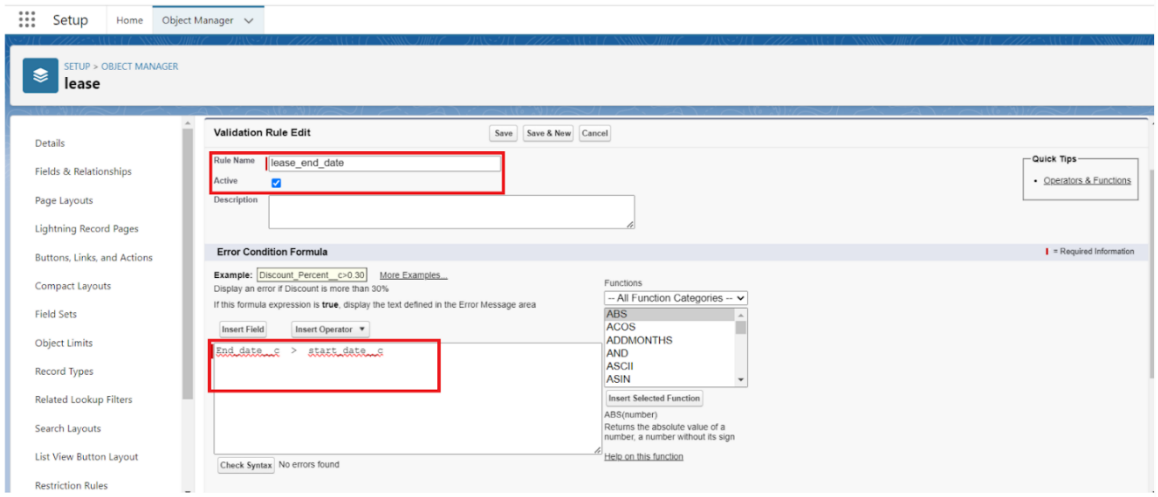
Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

13. To create a validation rule to an Lease Object

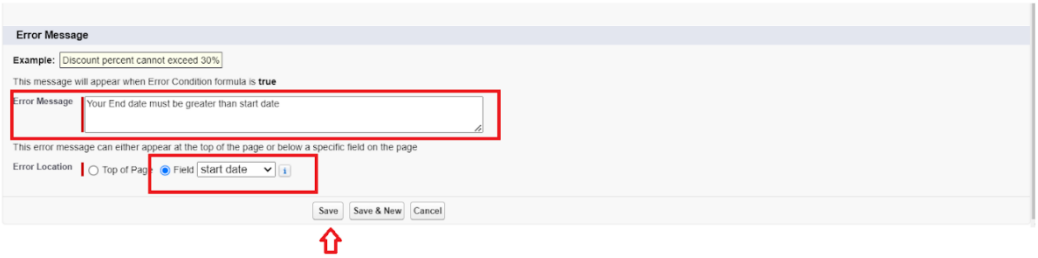
1. Go to the setup page >> click on object manager >> From drop down click edit for Lease object.
2. Click on the validation rule >> click New.

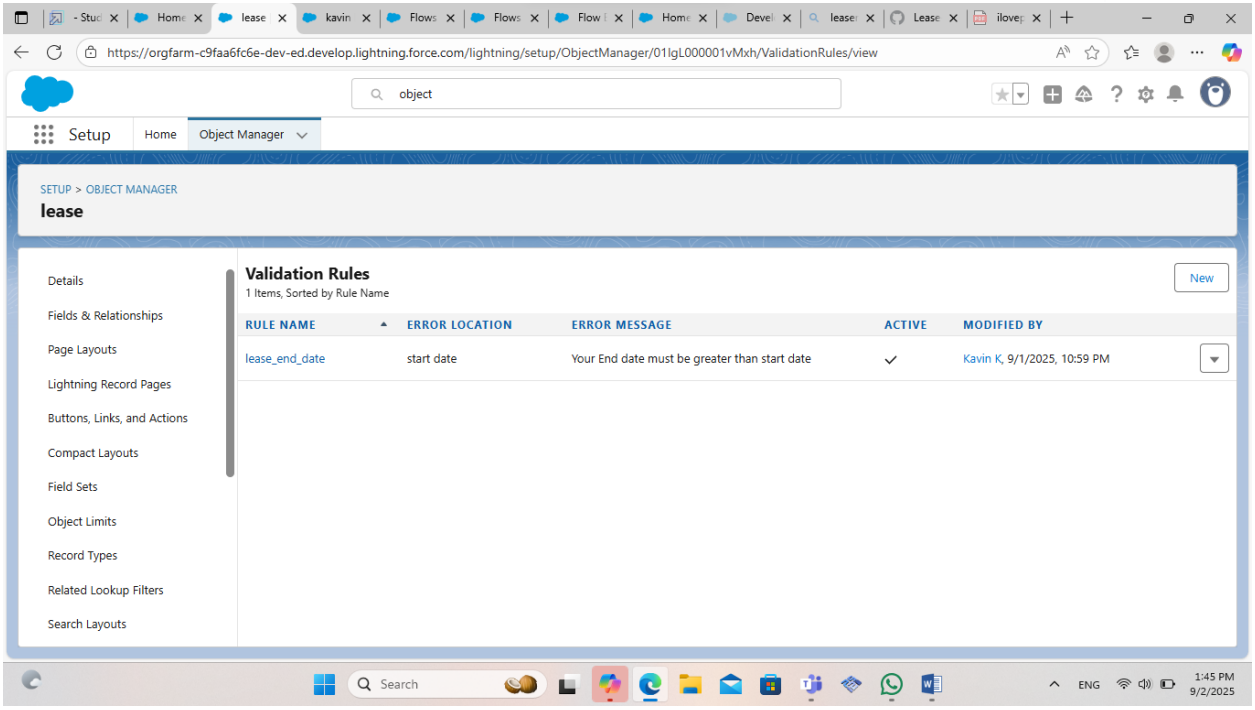


3. Enter the Rule name as " lease_end_date".
4. Insert the Error Condition Formula as :
End_date__c > start_date__c



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.





14. 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.

15. Create Email Template For Tenant Leaving

2. To create Email Template:

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

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

Folder : Unfiled public Classic Email templates

Click on available for use

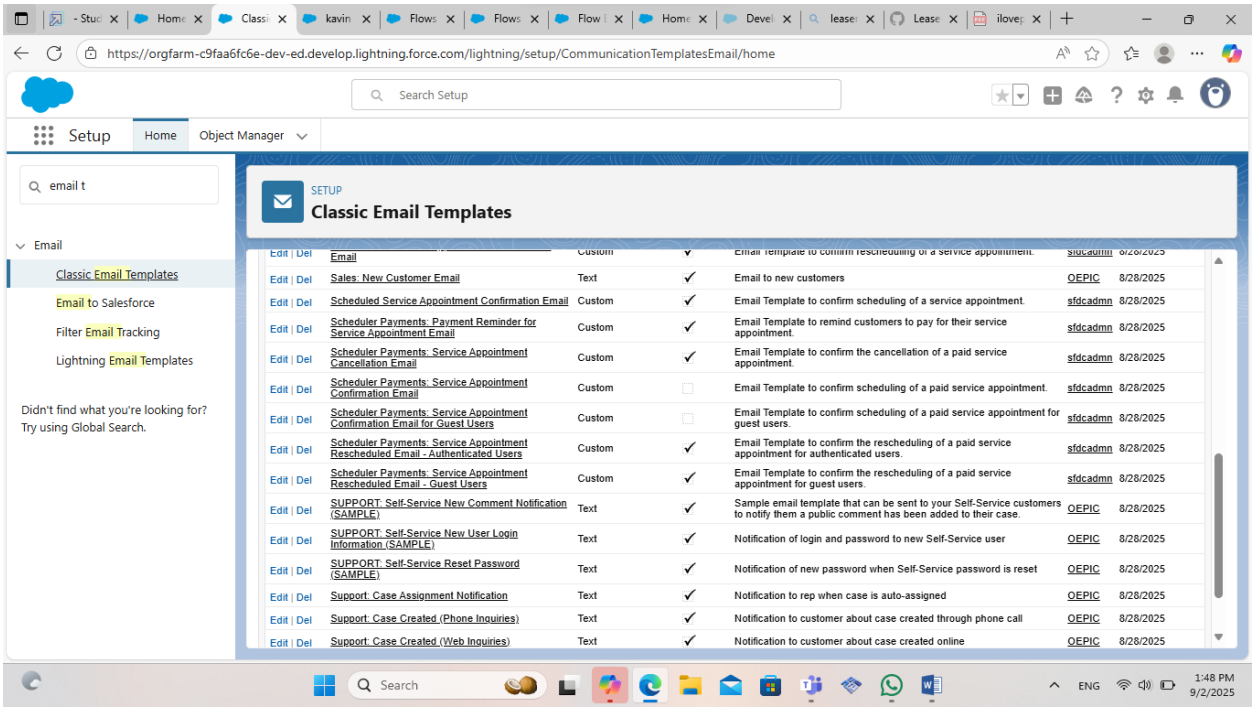
5. 3. Email Template Name is “tenant leaving”

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



1. Create Email Template For Leave Approved

6. To create Email Template:

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

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

Folder : Unfiled public Classic Email templates

Click on available for use

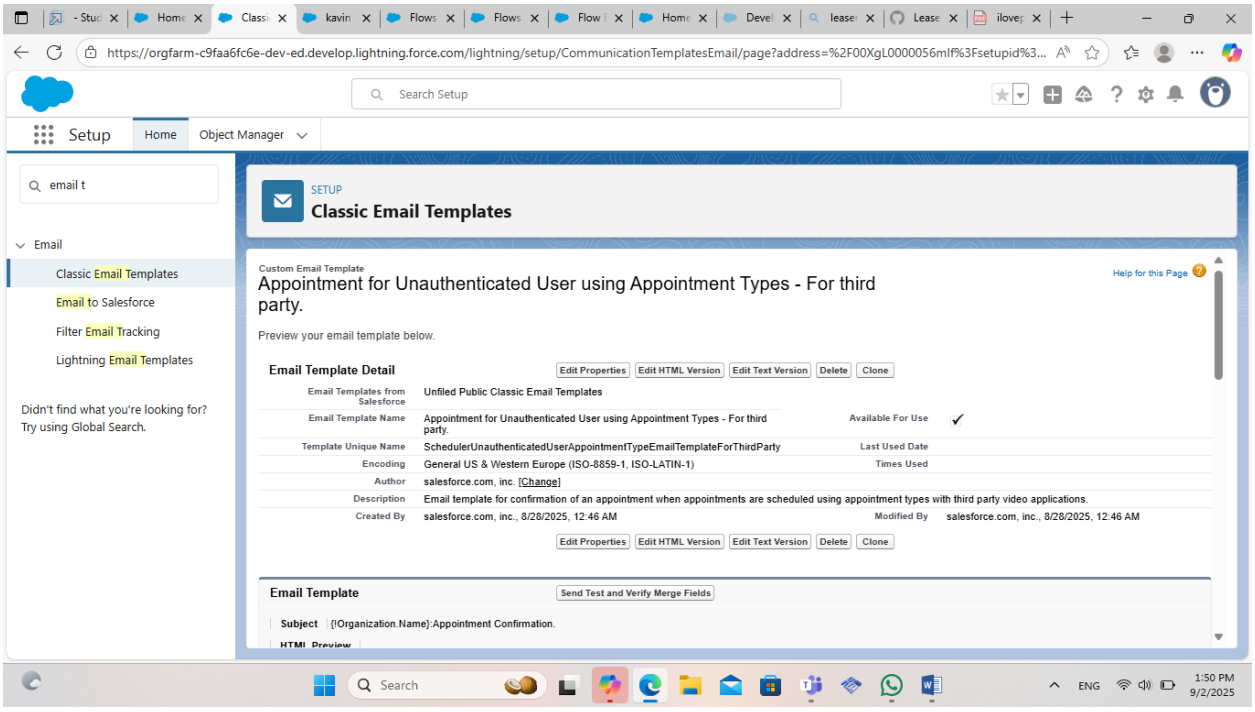
9. 3. Email Template Name is "Leave approved"

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



1. Create Email Template For rejection for leave

10. To create Email Template:

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

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

Folder : Unfiled public Classic Email templates

Click on available for use

13. 3. Email Template Name is "Leave rejected"

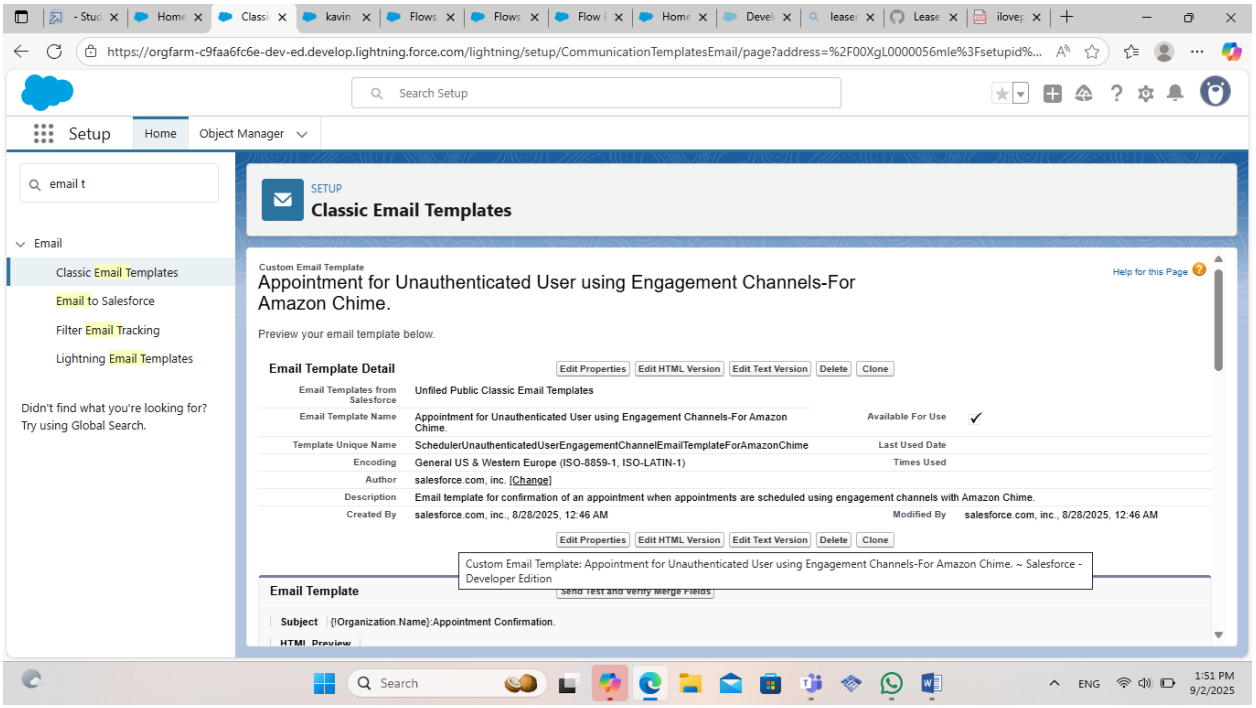
4. Template Unique Name : Auto populated
5. Subject : " Leave rejected"
6. Email body :

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



1. Create Email Template For Monthly payment

14. **To create Email Template:**

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

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

Folder : Unfiled public Classic Email templates
Click on available for use

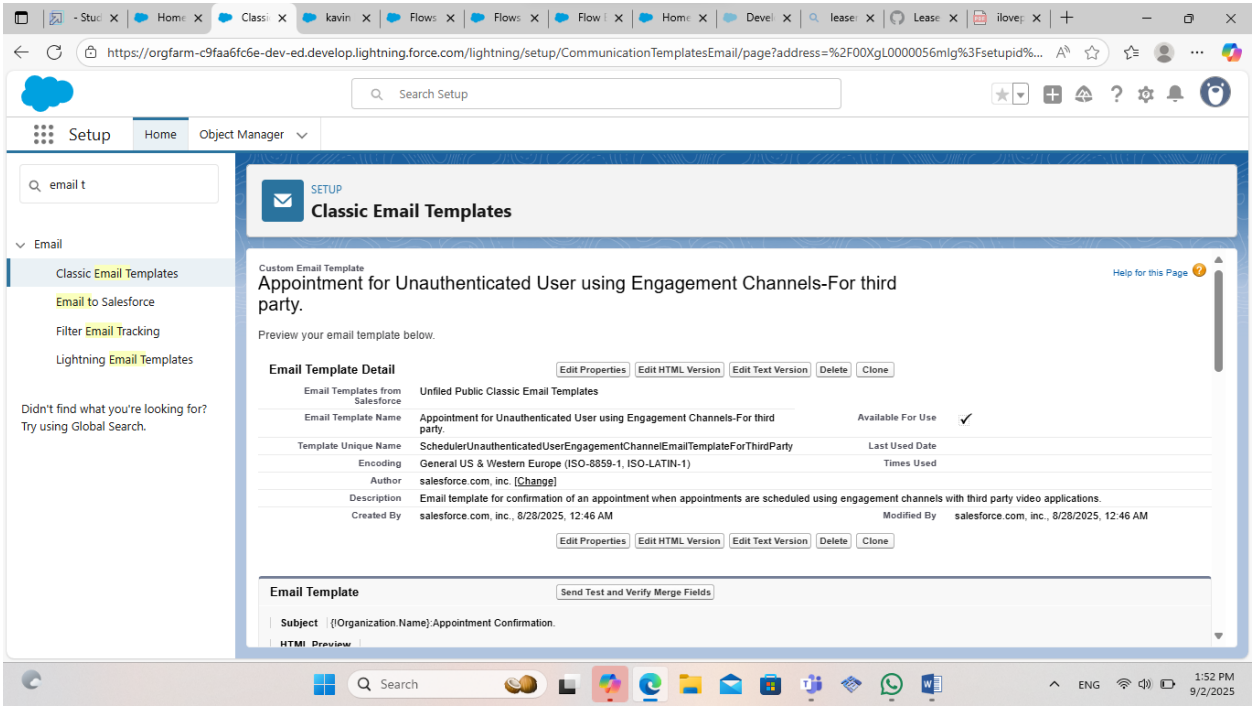
17. **3. Email Template Name is "Tenant Email"**

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



1. Create Email Template For successful payment

18. **To create Email Template:**

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

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

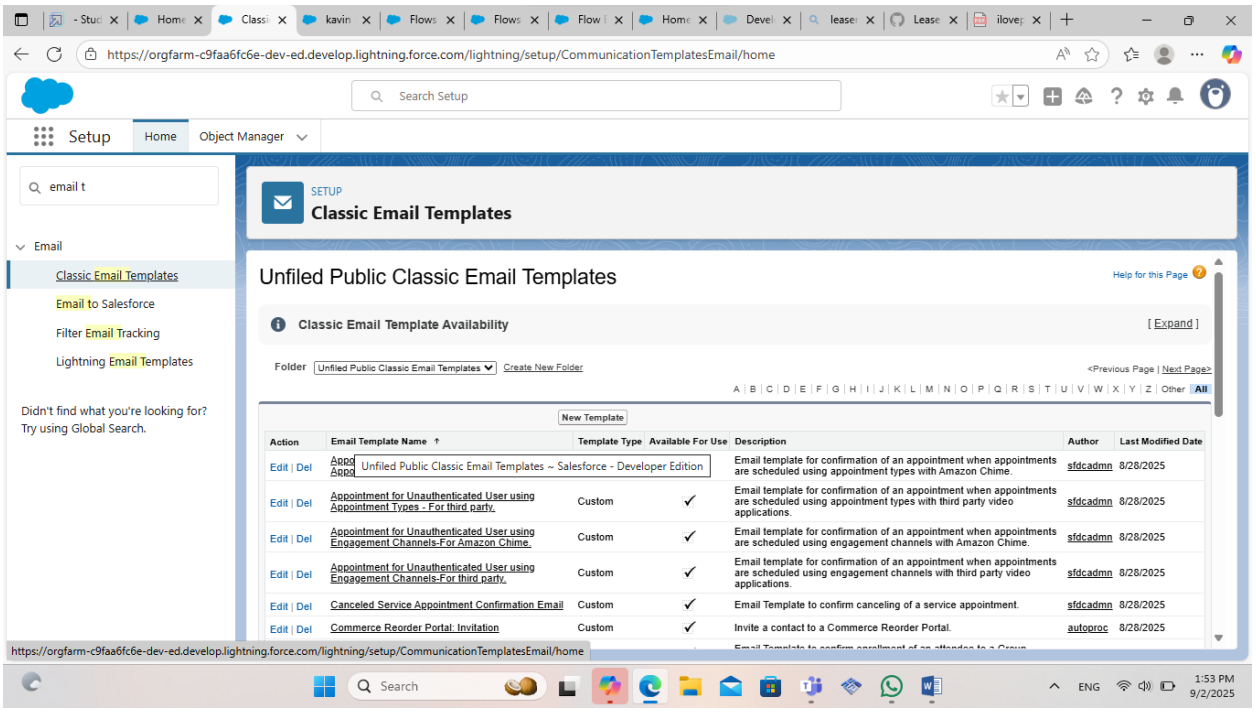
Folder : Unfiled public Classic Email templates
Click on available for use

21. 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



1. Approval Process

2. What Is Approval Process In Salesforce?

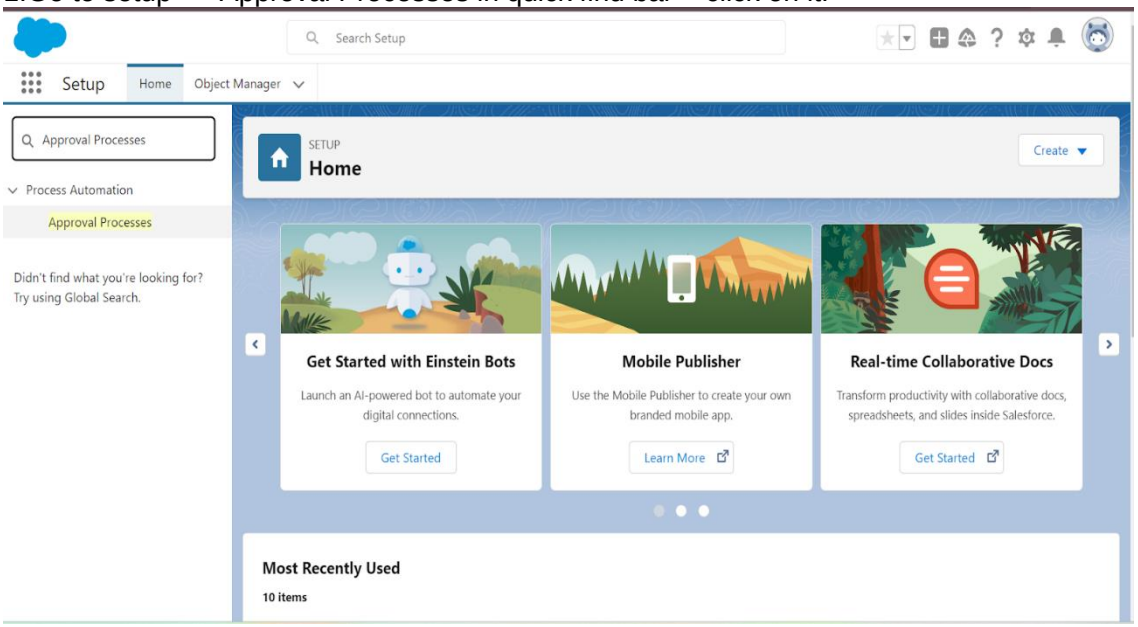
The Approval Process is an automated process that an org uses to approve records in Salesforce. For example, When In the organization, someone is not able to decide a particular thing then he can ask someone else for approval. So, for such frequent cases or situations, one can define the approval process.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

2. *Actions In Salesforce Approval Process*

1. 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.

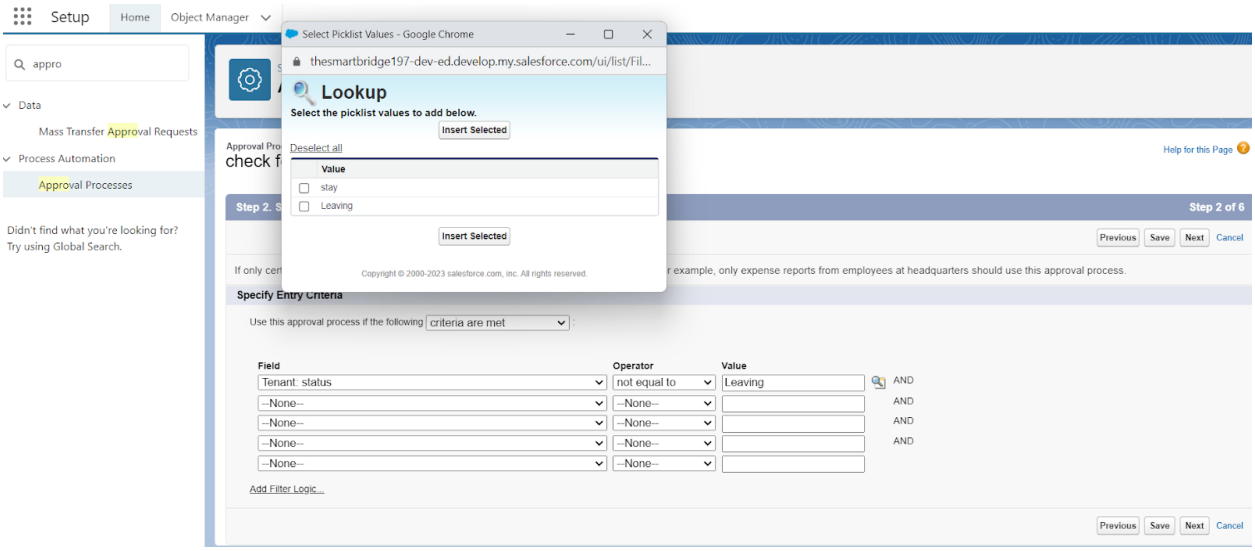


2.Manage Approval Process For >> “Tenant” from the drop down.

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

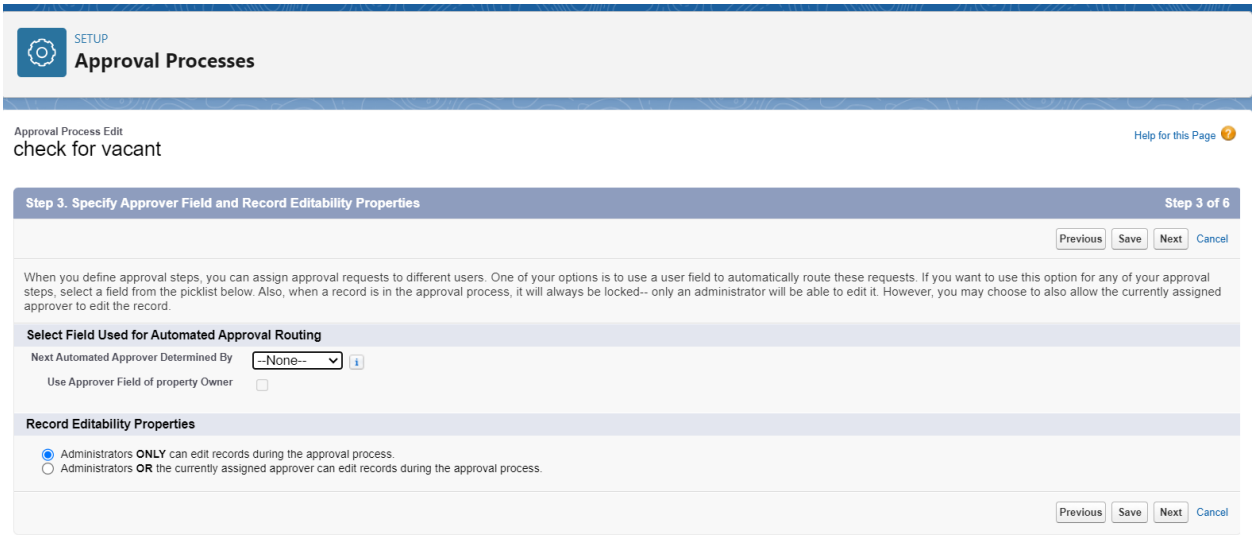
4. Process Name “check for vacant” >> Click Next.

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

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



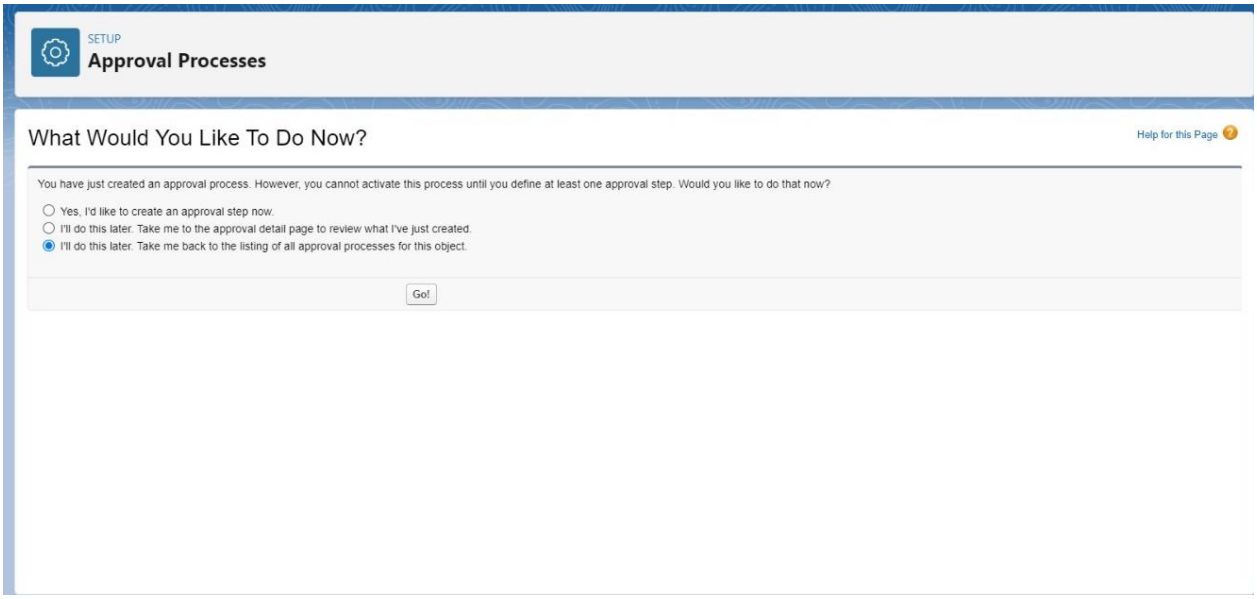
9. Click on next leave the email template click on next

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

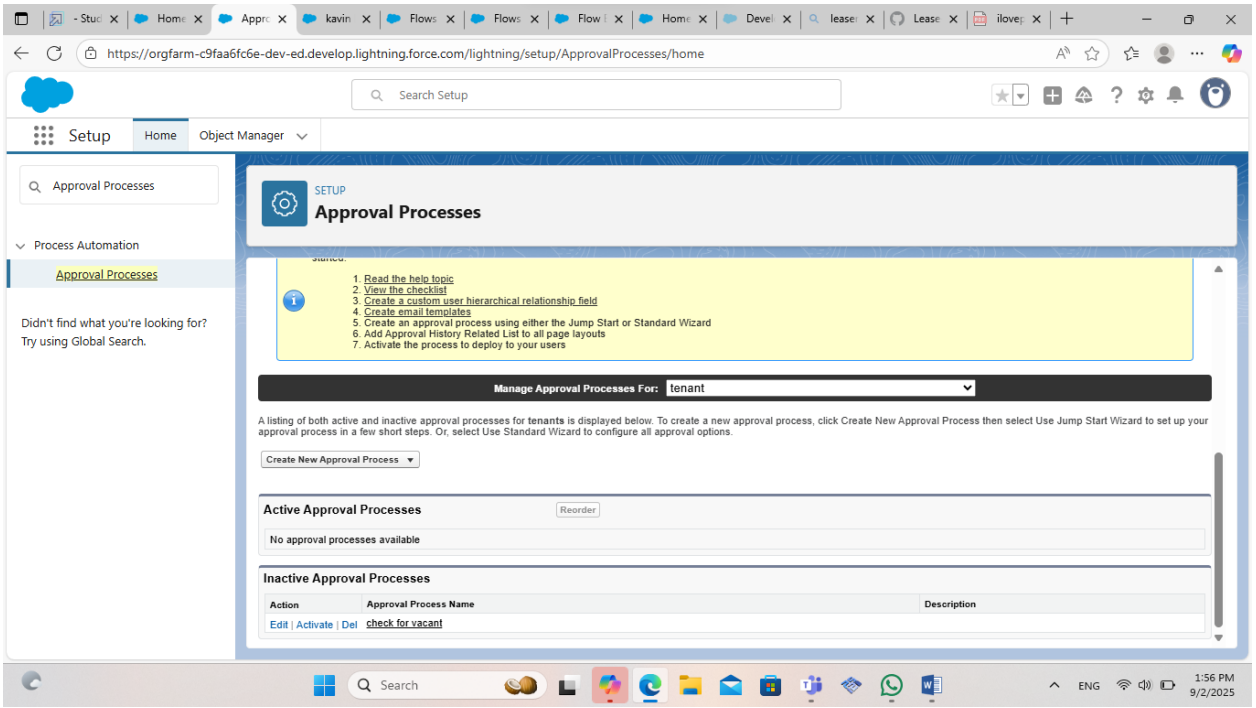
- Make sure Display approver history is checked.
- And under security settings check the “Allow approvers to access the approval page only from within the Salesforce application. (Recommended)” option.

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

- Then click save.

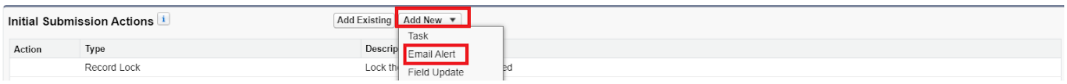


- Click on “i’ll do this later. Take me back to the listing of all approval process for this object”
- Click go



2. 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

3. Final Approval Action

1. Under Final approval action click o n new and then select email alert.
2. Description: “Tenant leaving”.
3. unique name : auto populated
4. Email 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
8. Click save

4. Apex Trigger

3. *Use case:*

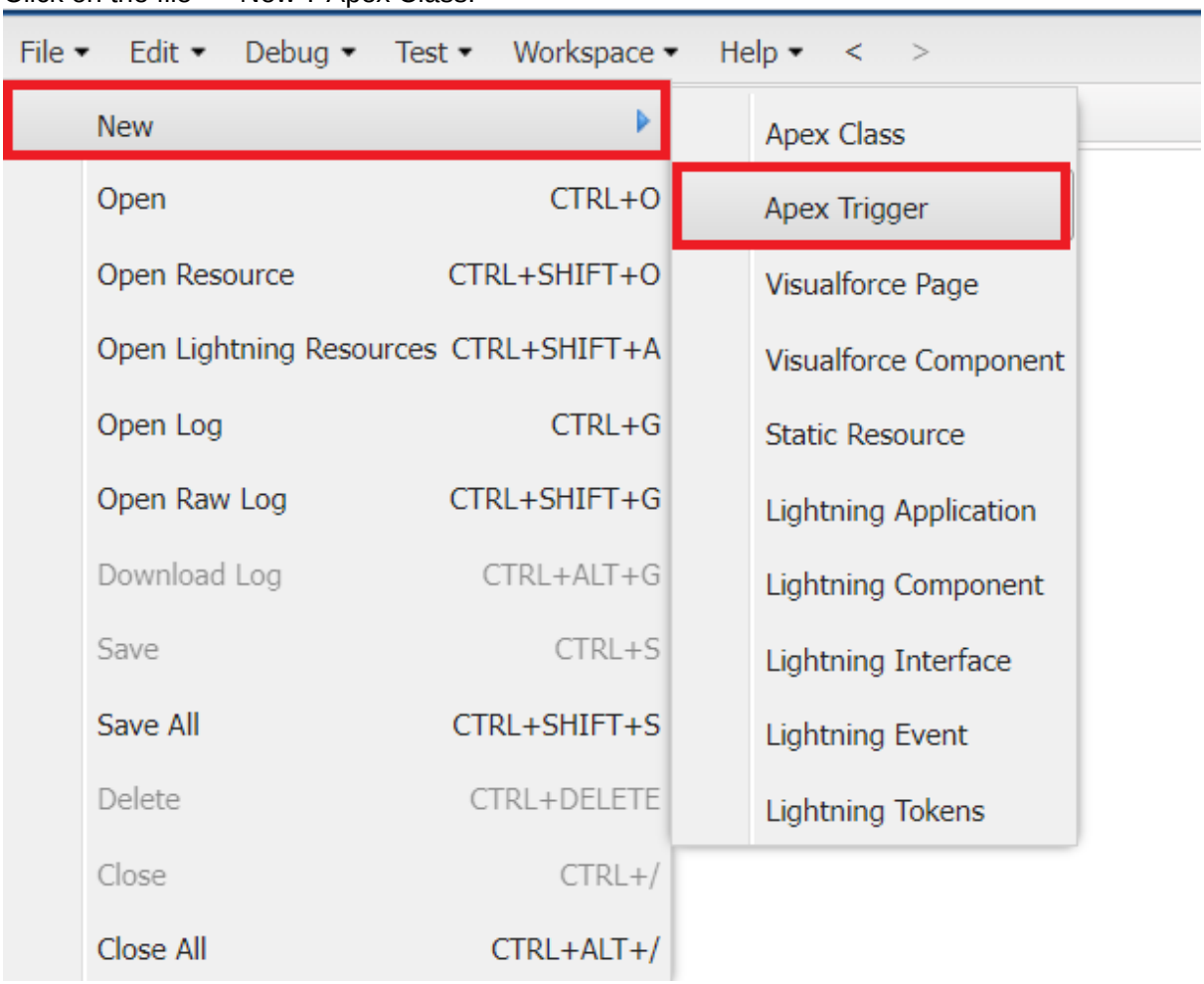
The tenant and property are in a master-detail relationship, wherein each tenant is associated with only one property. When a tenant attempts to create a new record with an existing property, an error should be displayed, indicating that a tenant can have only one property.

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

1. Create an Apex Trigger

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

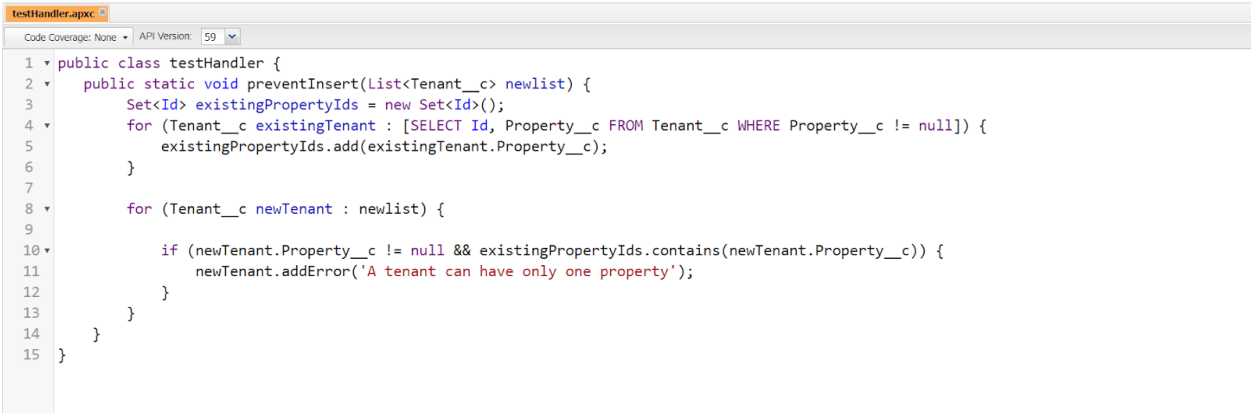
   Click on the file >> New ? Apex Class.



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

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.

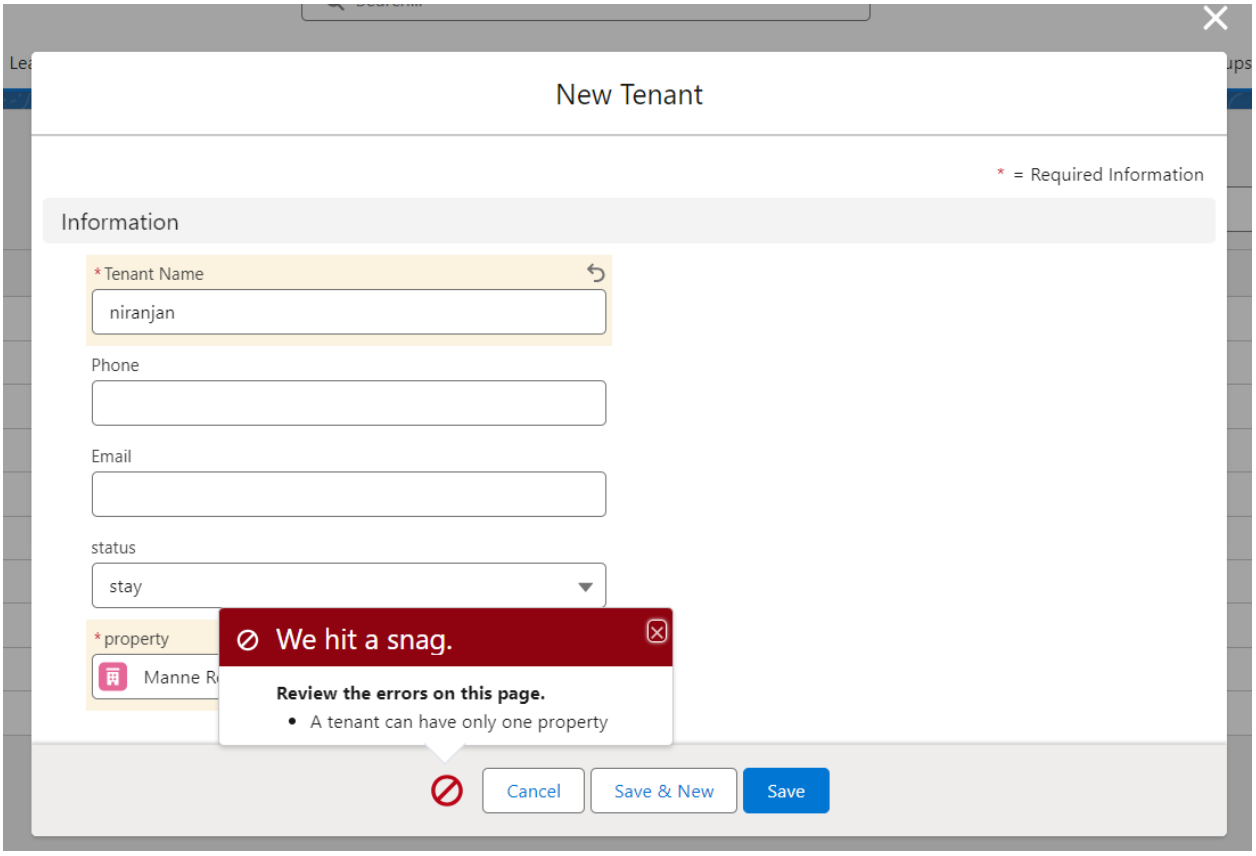


Apex logic:

```
public class testHandler {
  public static void preventInsert(List<Tenant__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');
         }
      }
   }
}
```

2. Testing the Trigger

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



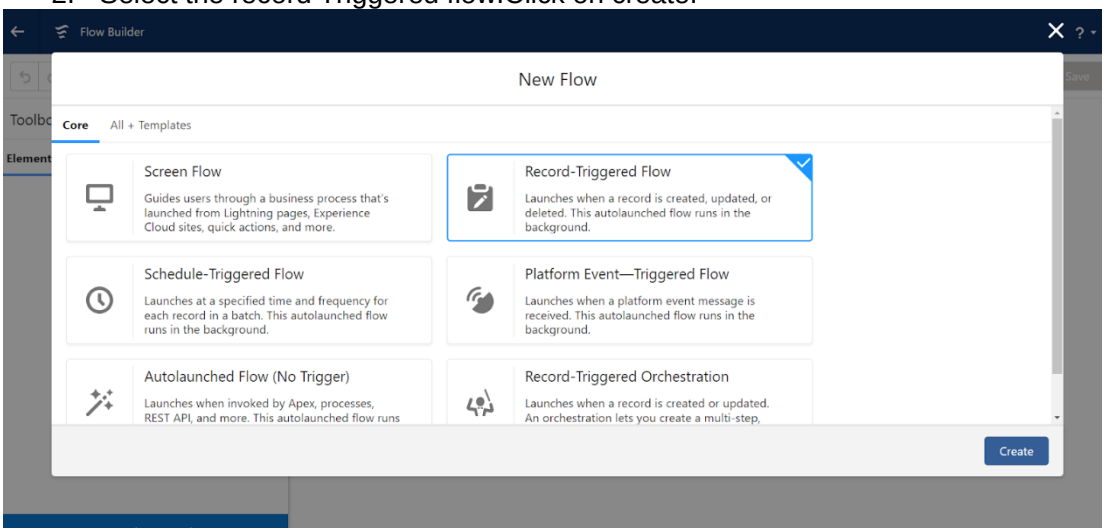
FLOWS

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 UI 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 process, or user guided experience that does not reach the complexity threshold for Apex Code, then flow is your go-to tool. If you are modifying an existing process that was built with Process Builder or workflow, then you should consider a number of factors when deciding whether to modify the existing process or migrate it to Flow. Flows are able to create, edit, and delete records in Salesforce, send emails, show relevant data and gather input from users, and generate outbound messages.

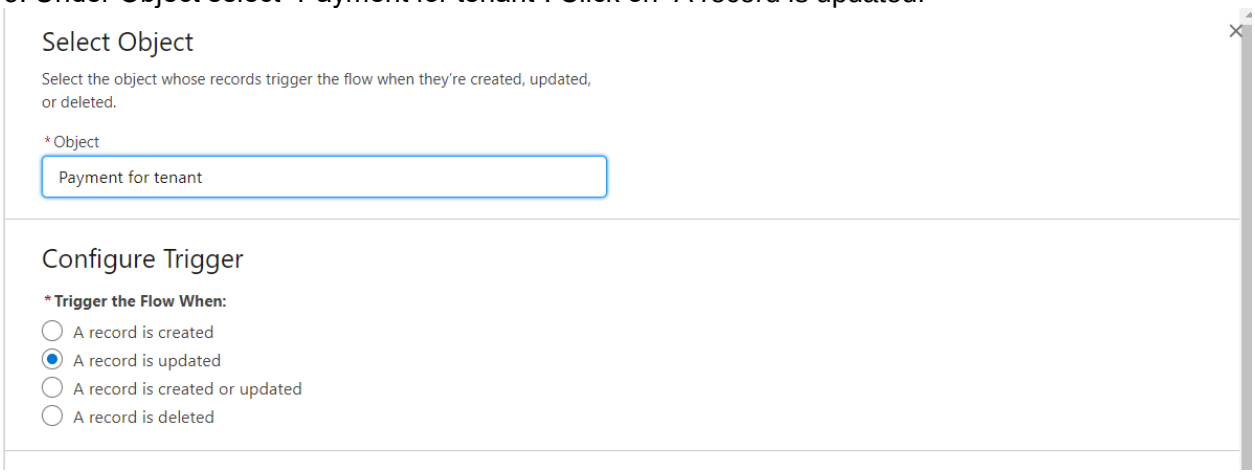
3. Create Flow for monthly payment

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.

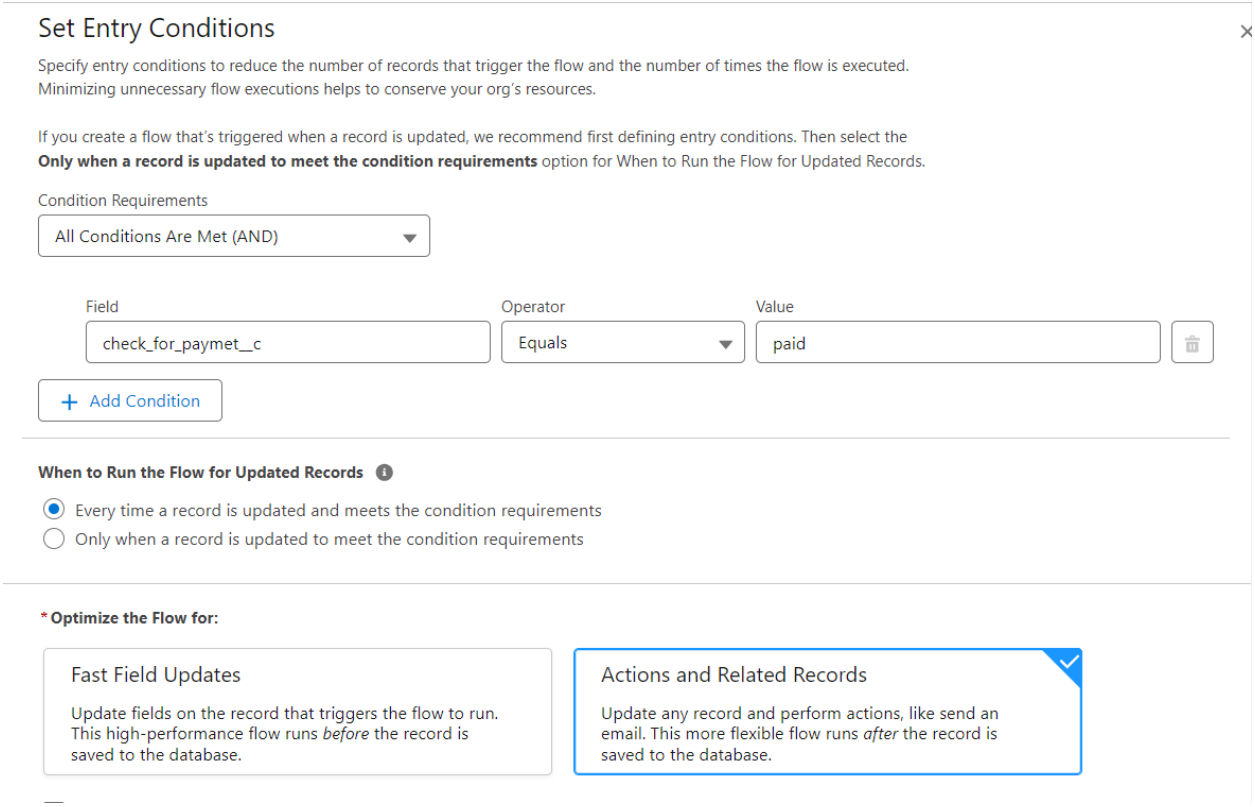


4. Set Entry Conditions

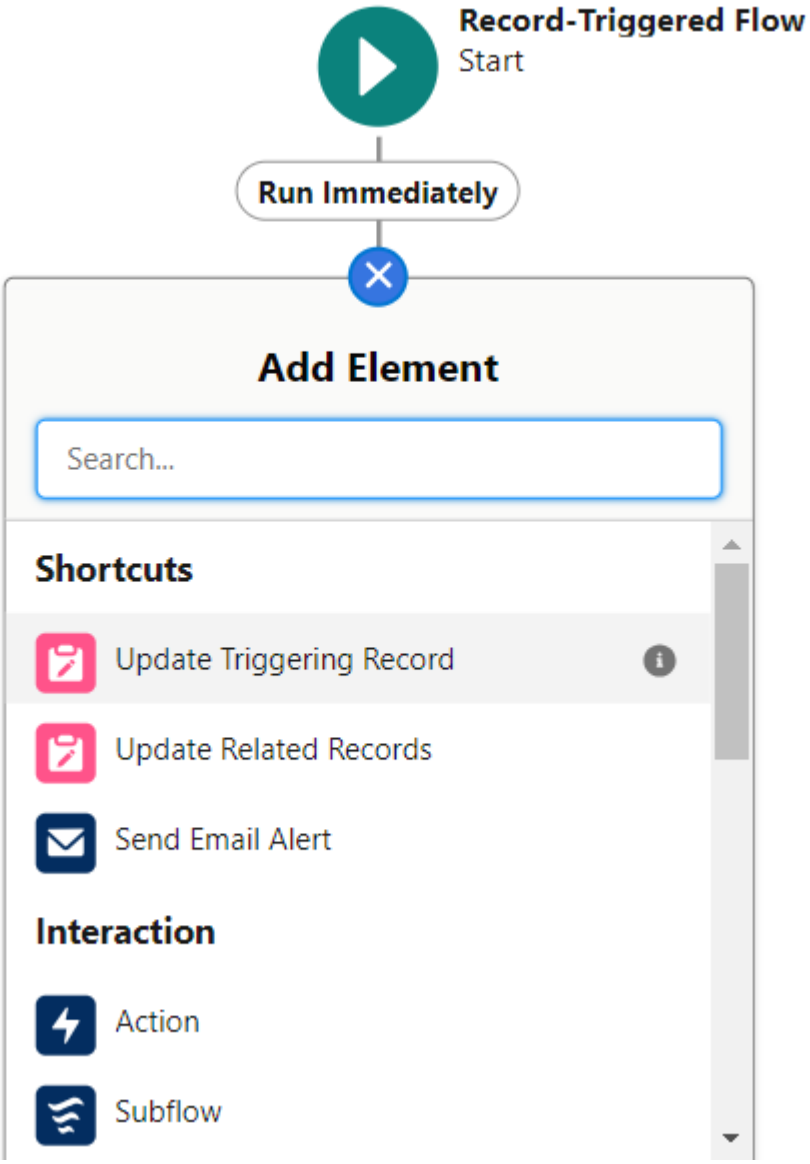
Under Condition Requirements
All Conditions are met

Field: 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



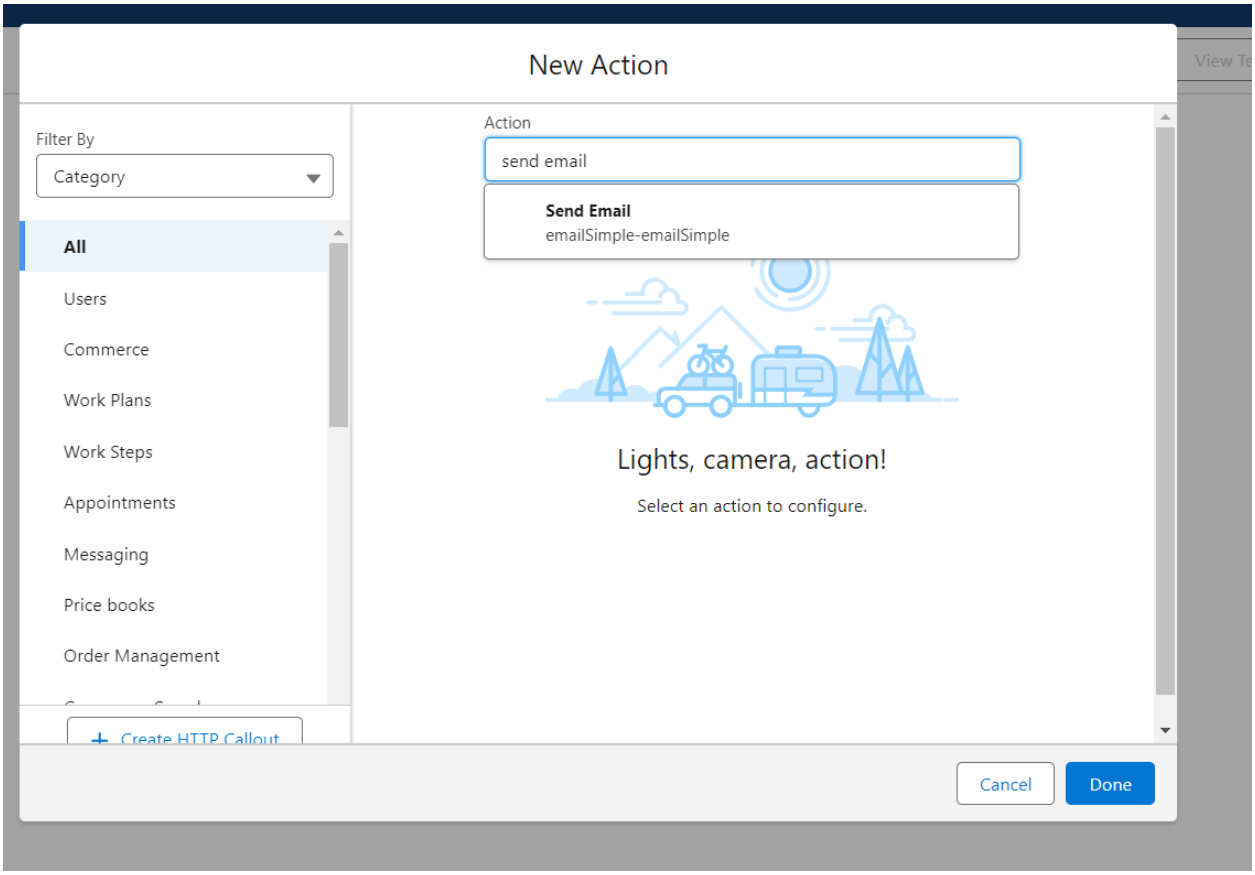
7. Under record trigger flow click on “+” icon and select action



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

8. Label : send email

API Name : send_email

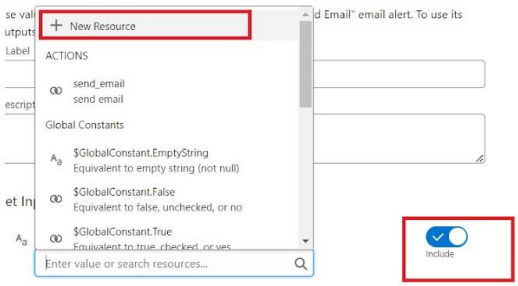


9. Label : send email

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

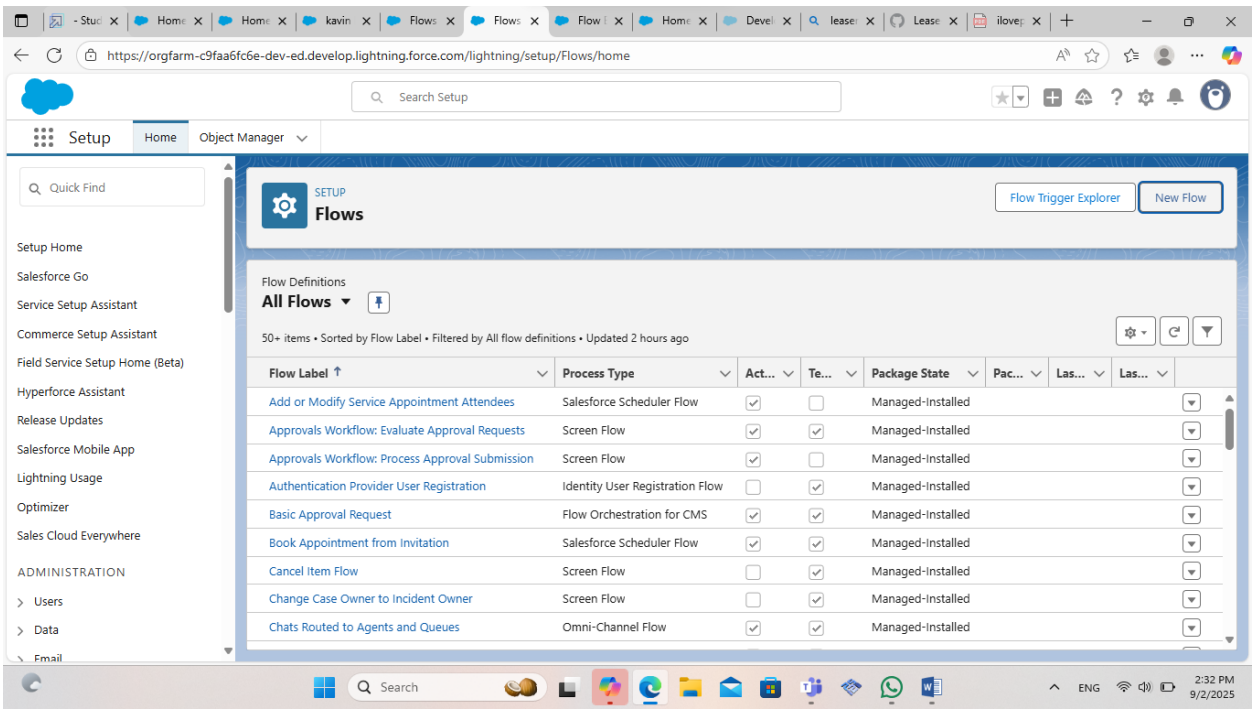
Pate this >> Confirmation of Successful Monthly Payment

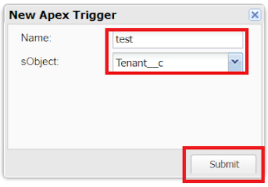
18. Click on save

Flow label : monthly payment

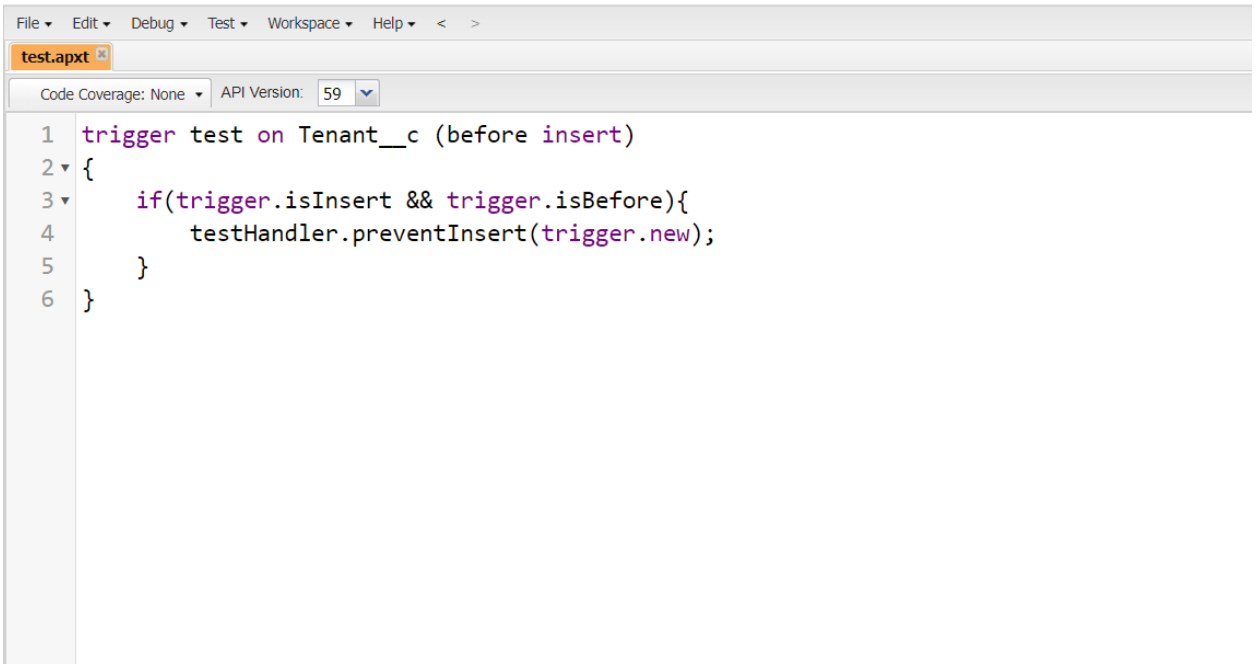
Flow API Name : monthly_payment

Click on activate





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



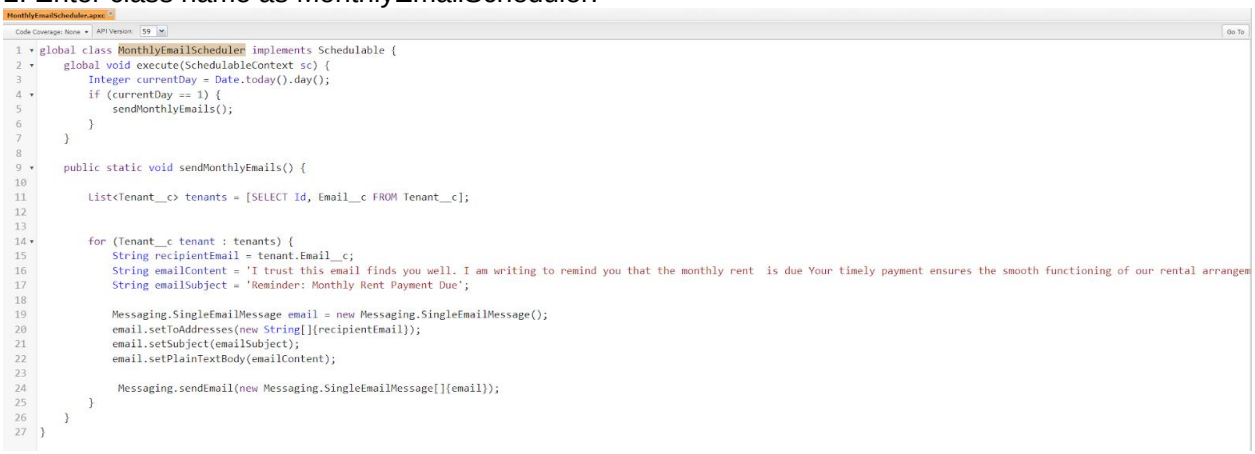
Trigger Code:

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

4. 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.



```
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});
        }
    }
}
```

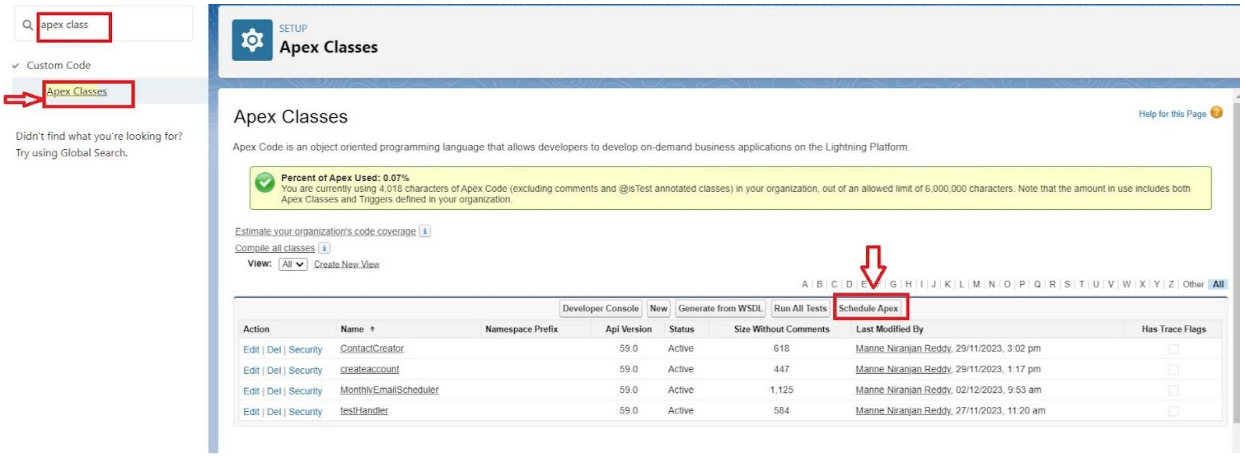
Apex logic:

```
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});
        }
    }
}
```

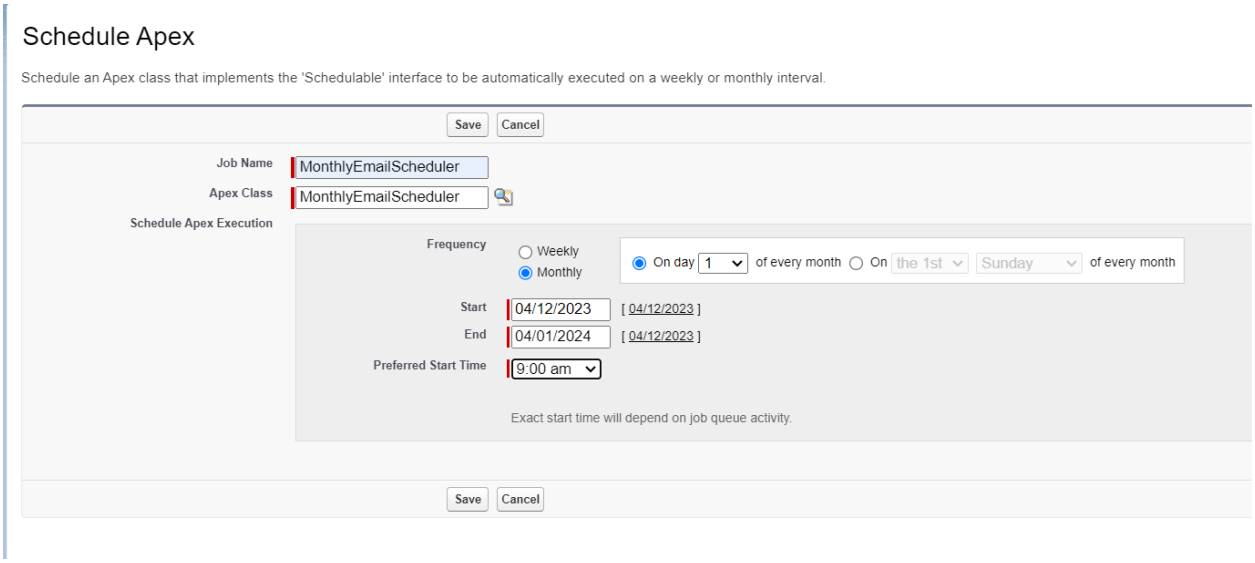
Save the code.

5. Schedule Apex class

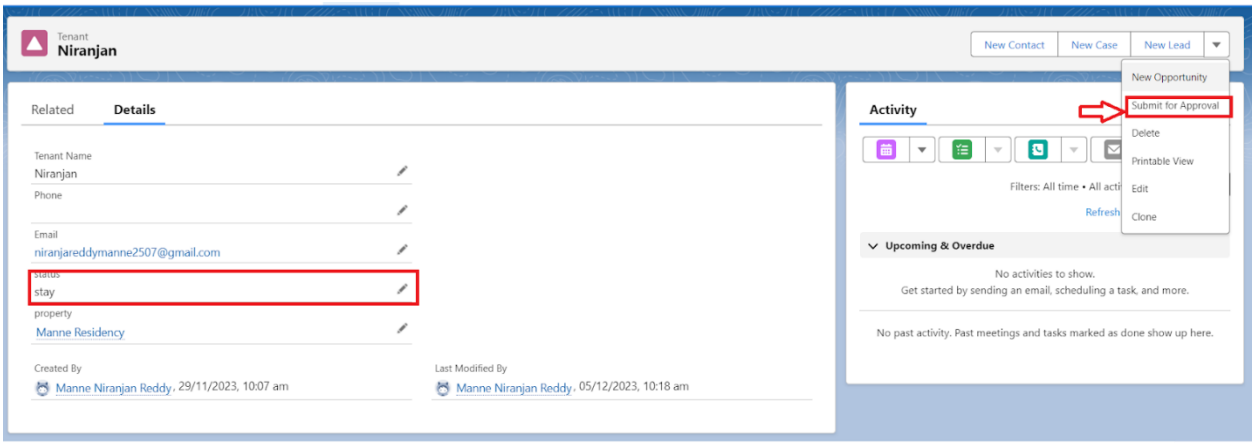
1. Enter Apex class in quick find box
2. Select schedule Apex



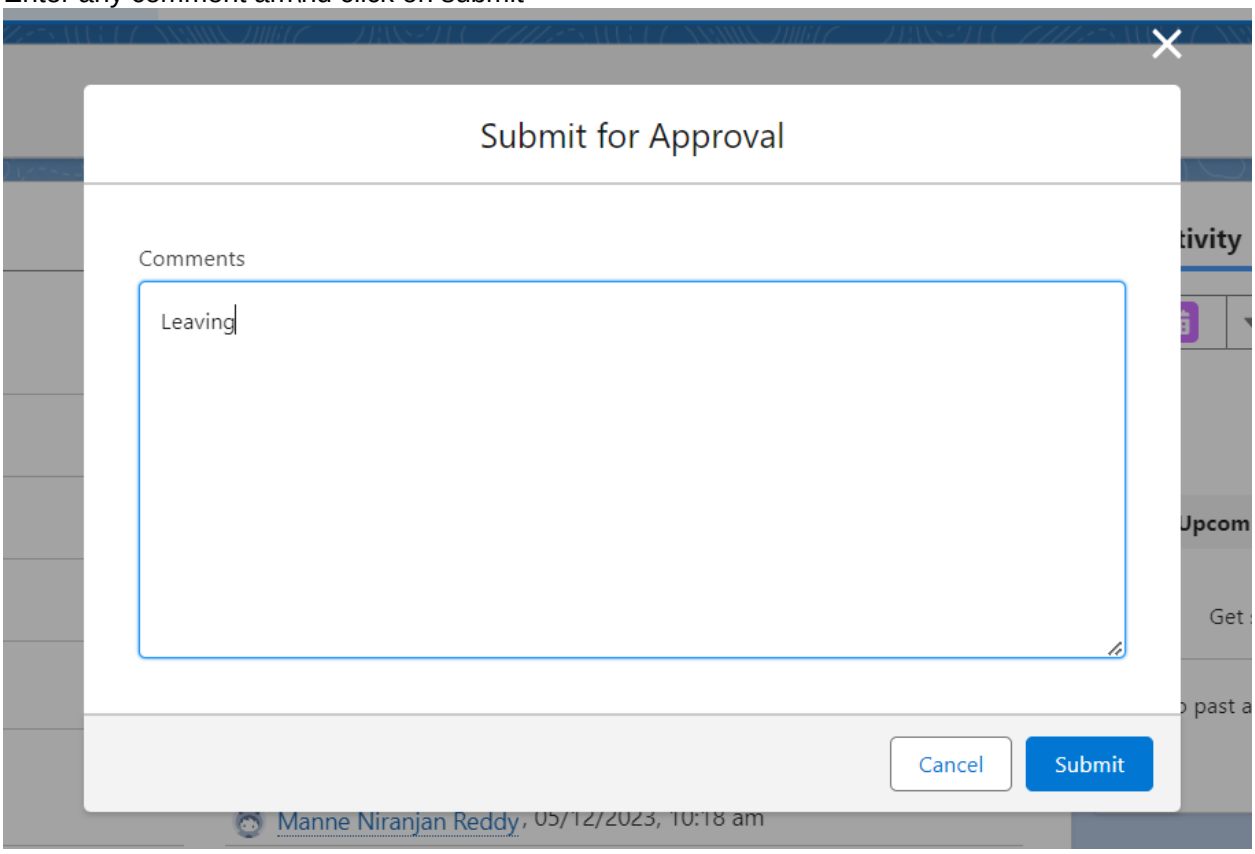
3. Enter job Name : MonthlyEmailScheduler
4. Apex class : MonthlyEmailScheduler
5. Frequency : Monthly===>select on day 1
6. Start date : 04/12/2023
7. End date : 04/01/2024
8. Preferred start time : 09:00 am
9. save

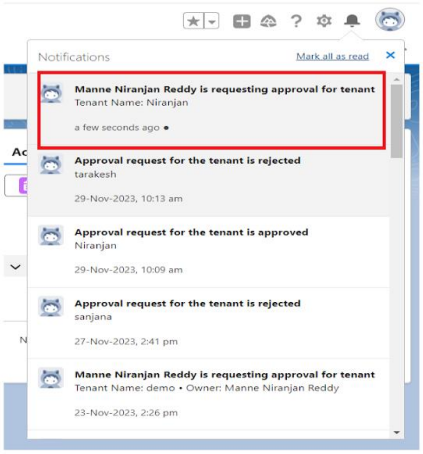


Testing the approval process

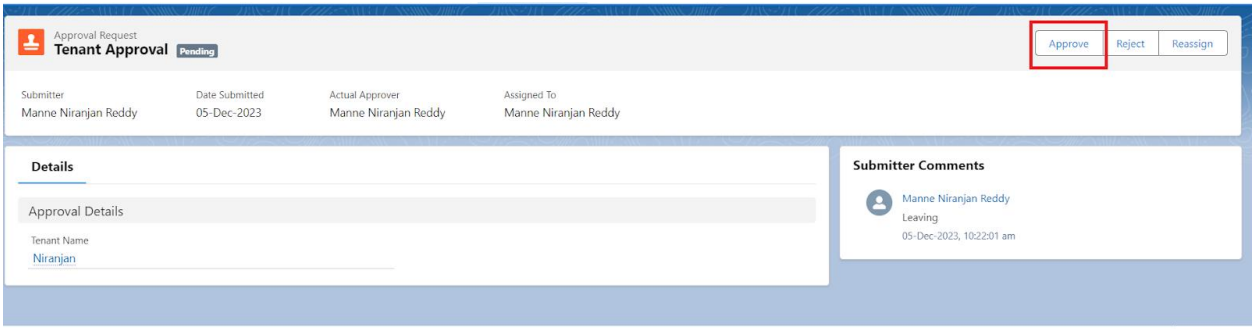


Enter any comment am\nd click on submit



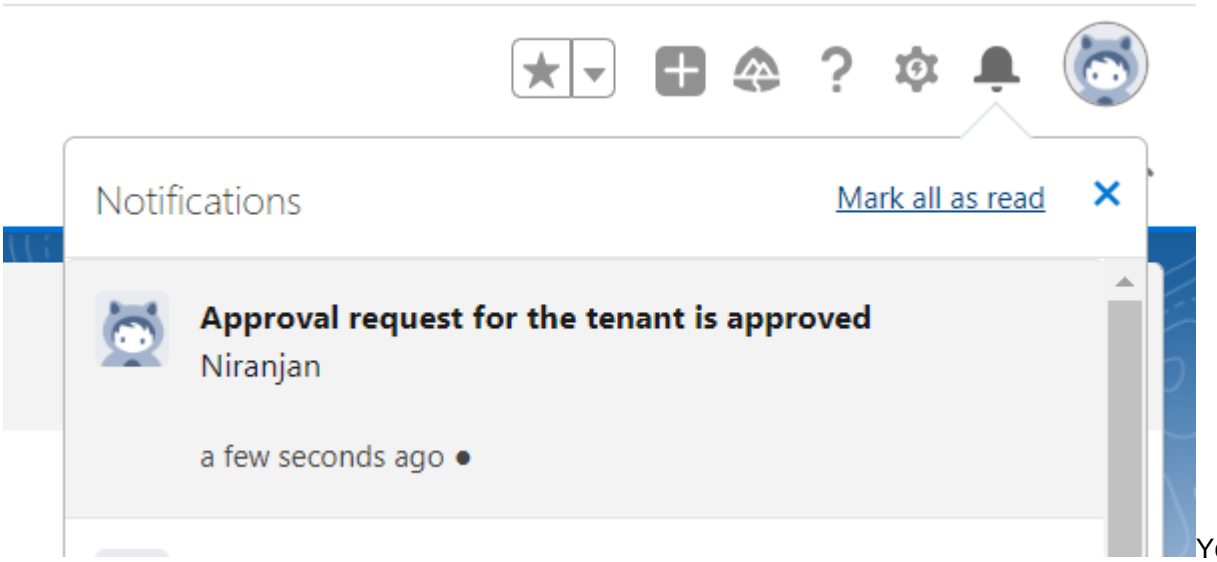


Click on that notification



click on approve

Give any comment and submit



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