Team Details

S.NO	NAME	ROLL NO	MAIL ID
1	CharuLatha.R	2322KB10	2322kb10@kgcas.com
2	Deepak.M	2322KB11	2322kb11@kgcas.com
3	Dhanushree.K	2322KB12	2322kb12@kgcas.com
4	Dharmesh.P	2322KB13	2322kb13@kgcas.com

Salesforce

Lease Management –(Developer)

A lease management project involves creating a system or application to efficiently handle the processes related to leasing real estate properties, equipment, or other assets. The goal is to streamline and automate various tasks associated with lease agreements, ensuring accurate record-keeping, compliance with regulations, and effective communication between parties involved.

What you'll learn

1. Creating Objects

2. Designing Tabs

3. Building Lightning Apps

4. Adding Custom Fields

5. Setting Validation Rules

6. Creating Email Templates

7. Configuring Approval Processes

8. Writing Apex Triggers

9. Working with Flows

10. Scheduling Apex Classes

Salesforce

Introduction

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you. Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?".

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 organised something like this:

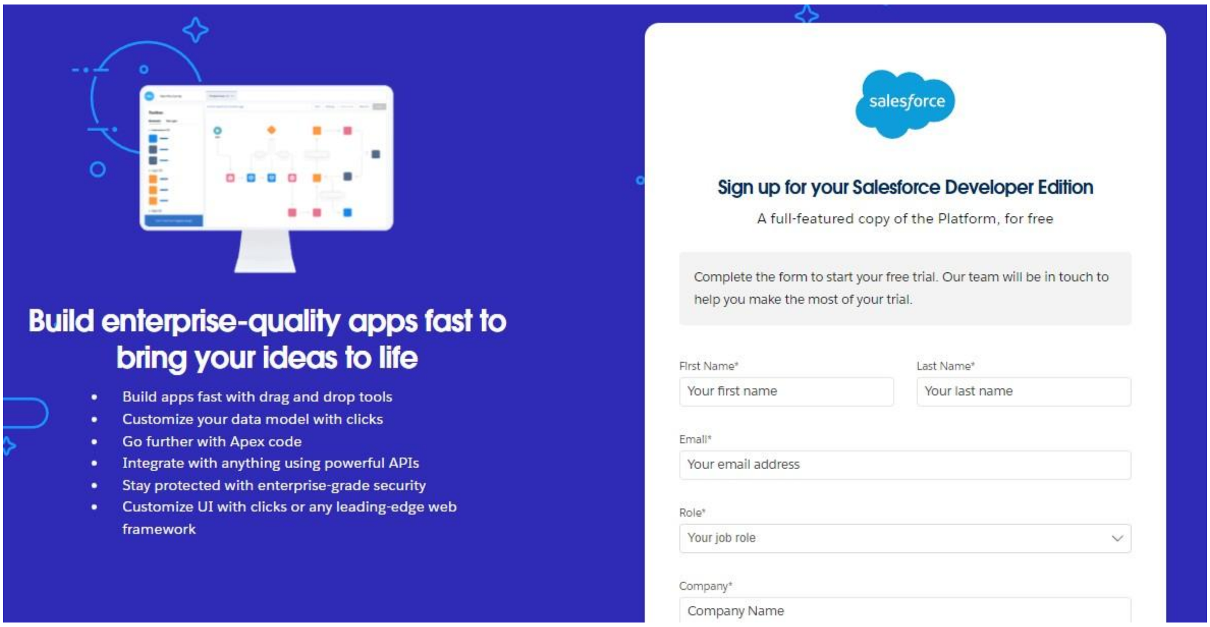
Lease Management

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. First name & Last name
2. Email
3. Role : Developer
4. Company : College Name
5. County : India
6. Postal Code : pin code
7. Username : should be a combination of your name and company This need not be an actual email id, you can give anything in the format

: username@organization.com

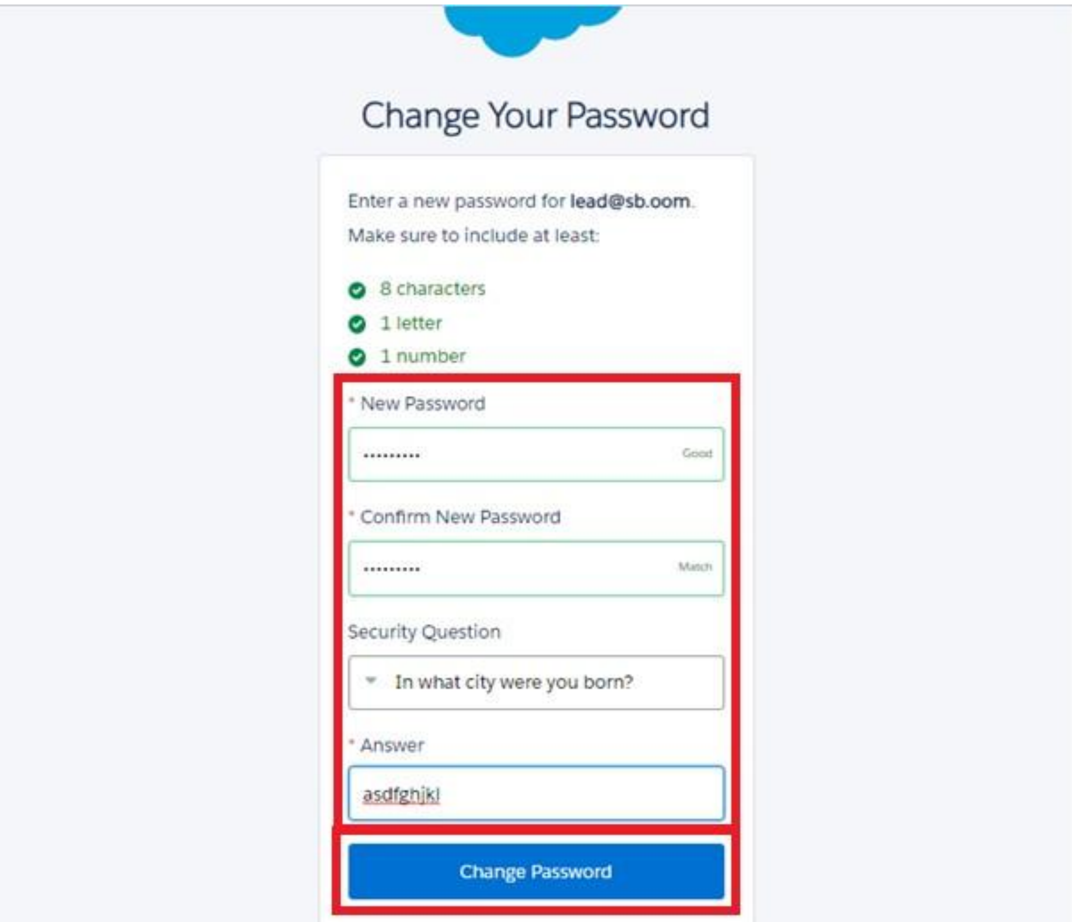
Click on sign me up after filling these.

Account Activation

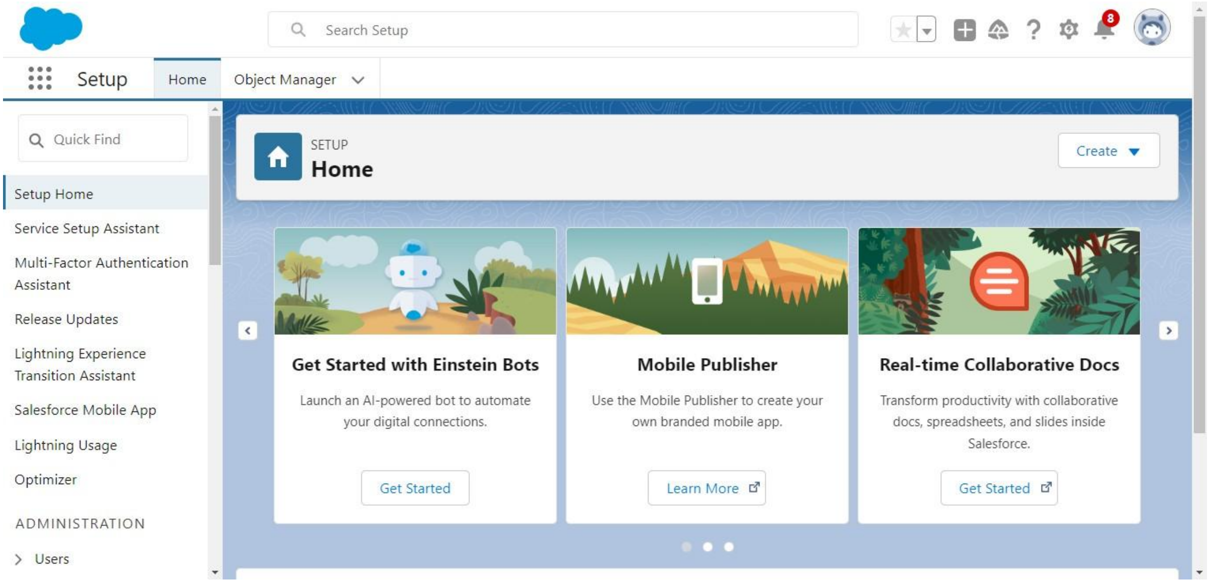
1. Go to the inbox of the email that you used while signing up. Click on the Reset Password to activate your account. The email may take 5-10mins.

0. Click on Reset Password

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



Then you will redirect to your salesforce setup page.



Object

What Is an Object?

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:

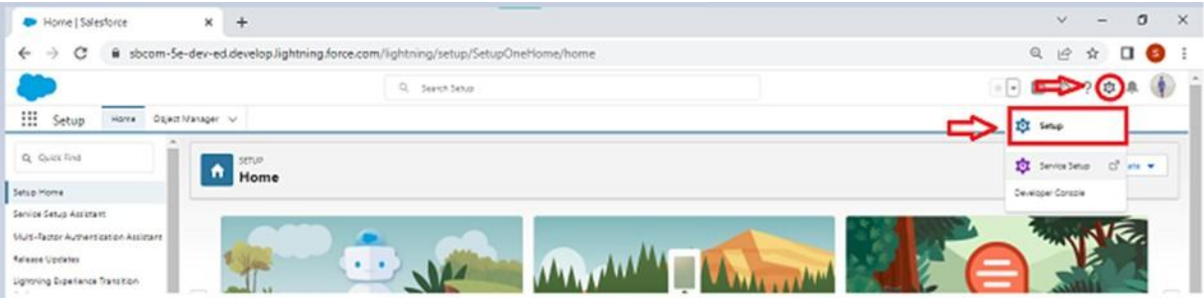
1. Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.

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. Use Case:

Creating an object in Salesforce organisation is essential for efficient data management and process automation. By defining custom objects, businesses can structure and store data specific to their needs, enabling streamlined workflows, personalized reporting, and enhanced user experiences. Objects serve as the foundation for organizing and leveraging critical information within Salesforce.

To Navigate to Setup page:

Click on gear icon >> click setup.



Create Property 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

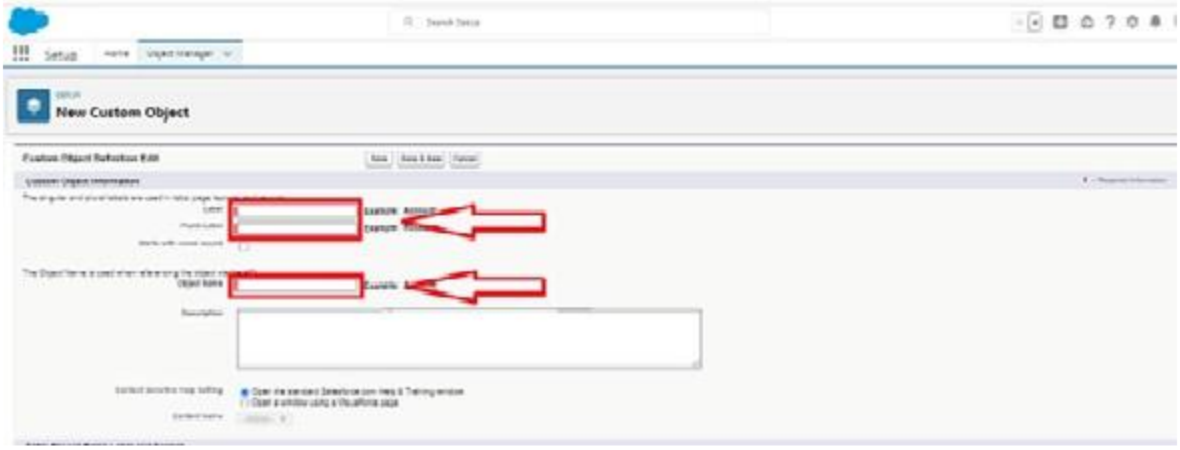
2. Plural label name>> property

3. Enter Record Name Label and Format

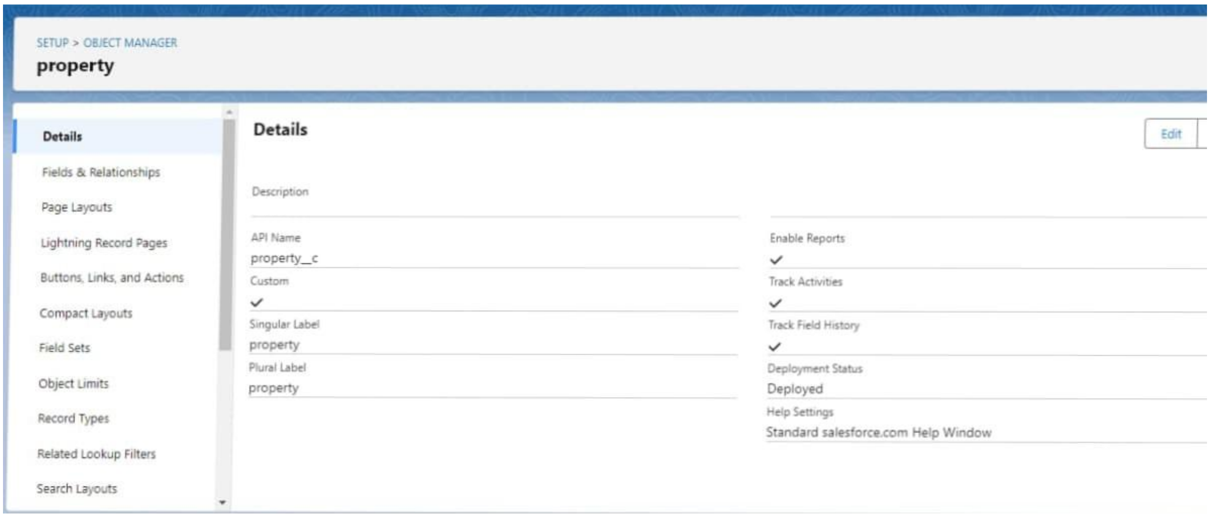
- Record Name >>property Name

- Data Type >> Text

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



3. Allow search >> Save.



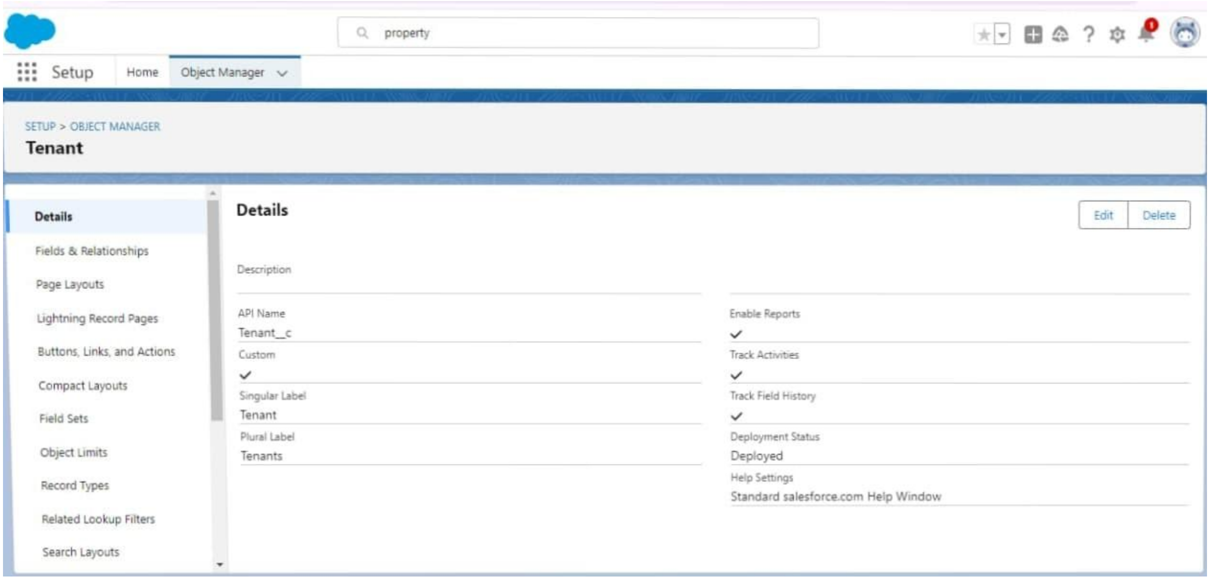
Create Tenant Object:

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.



Create Payment Object

To create an object:

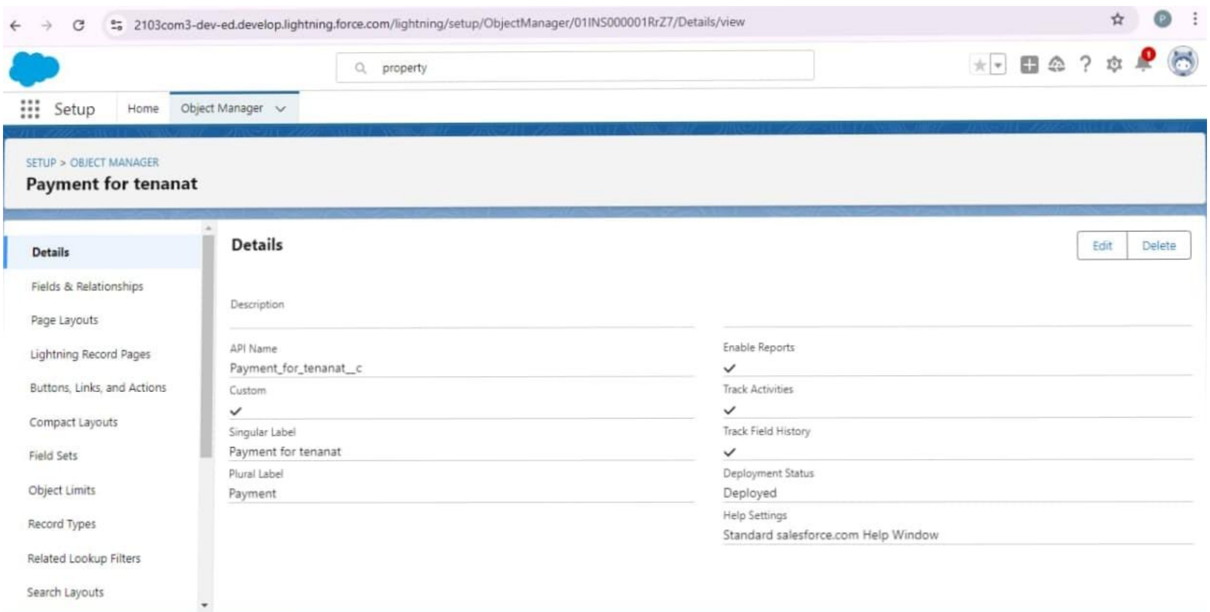
1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

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:

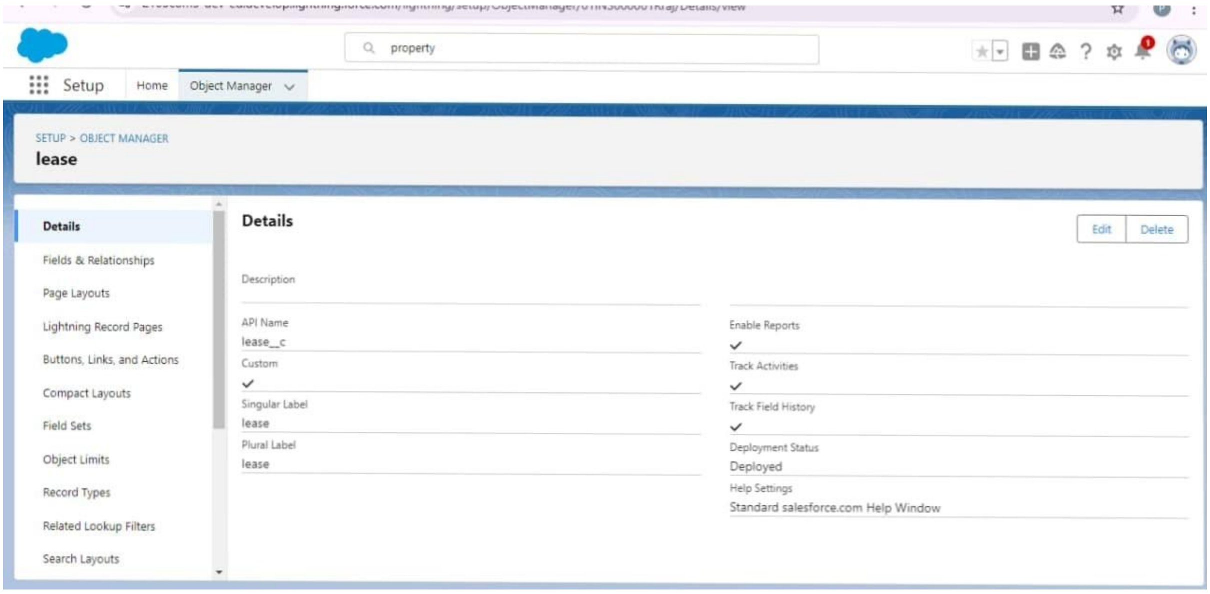
1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.

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 Activities

3. Allow search >> 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 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

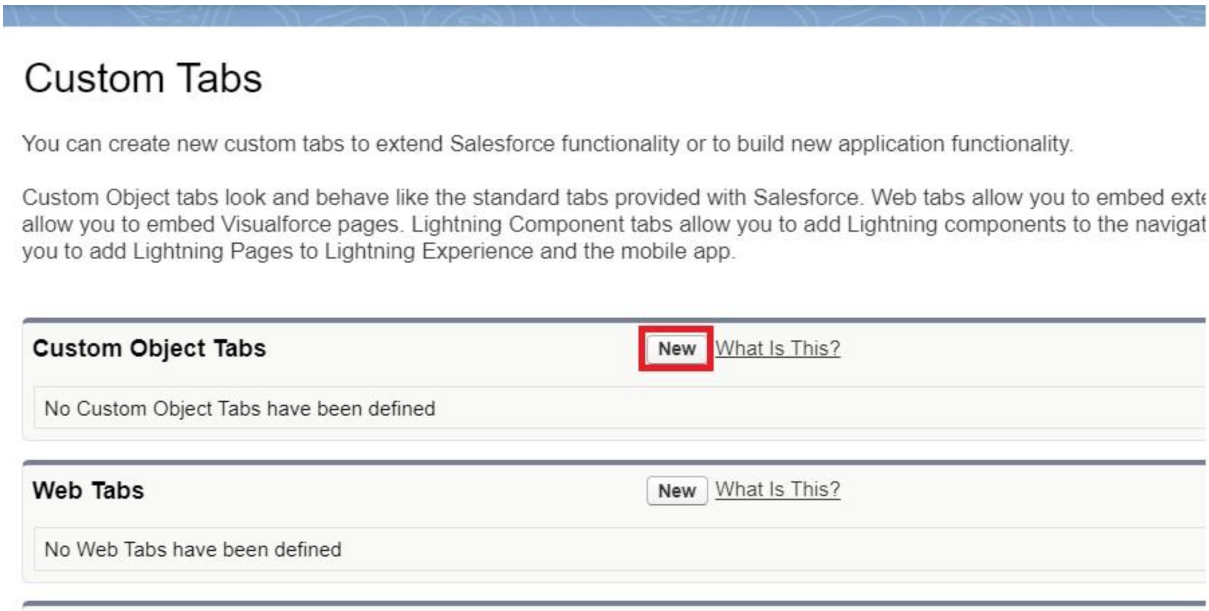
Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

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.

Creating a Custom Tab

To create a Tab:(Customer)

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

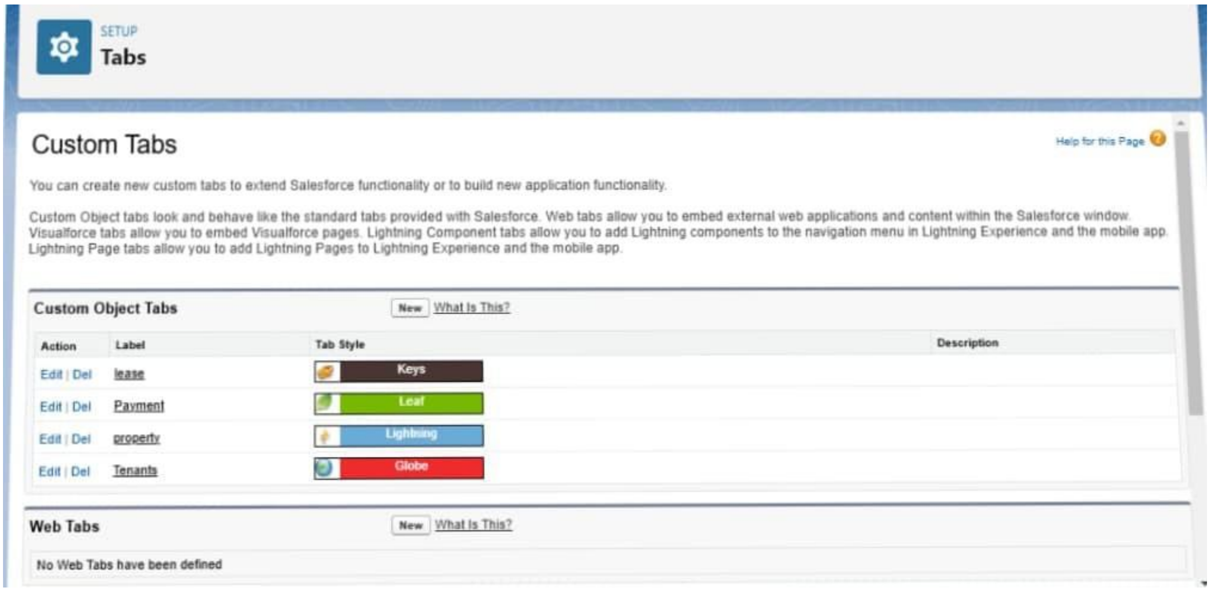


To create a Tab:(Property)

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 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.

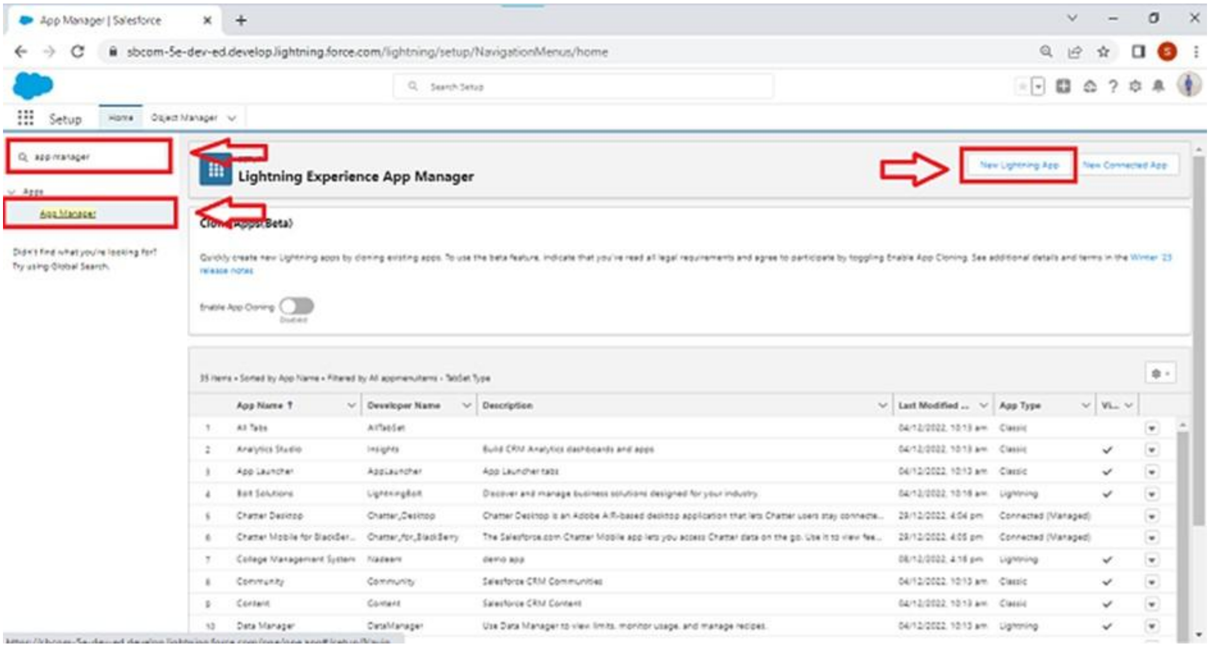
Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

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.

To create a lightning app page:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> 2. click on New lightning App.

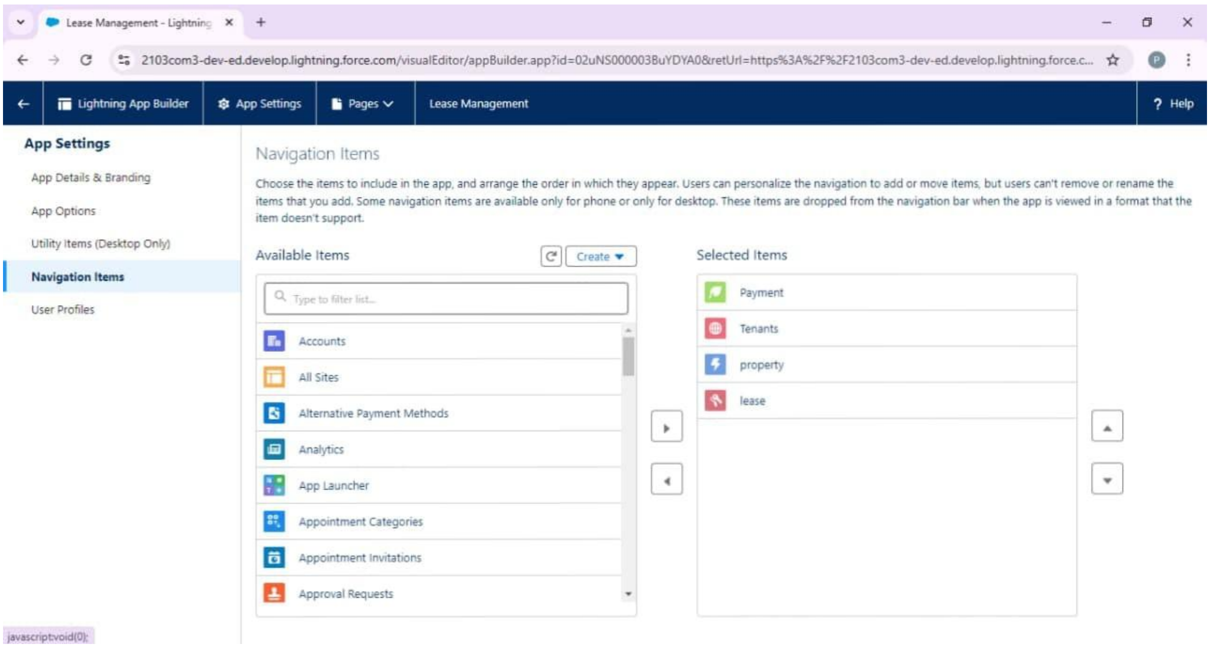


2. Fill the app name in app details and branding as follow

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.

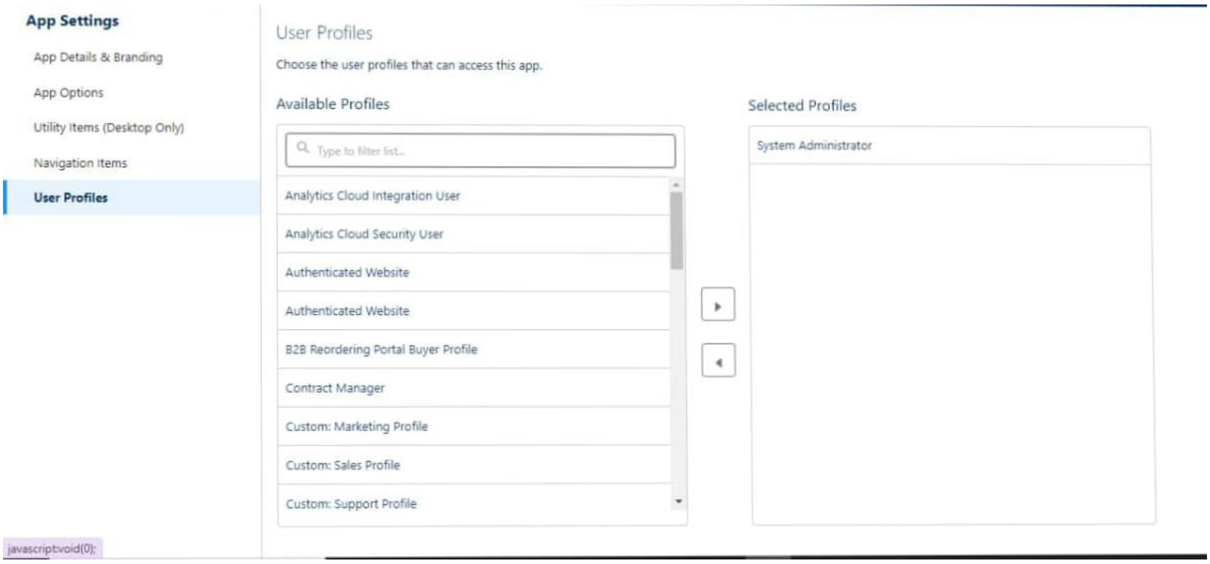
(Utility Items) keep it as default >> Next.



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

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 name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard 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 Salesforce application. They are,

>>Created By

>>Owner

>>Last Modified

>> Field Made During object Creation

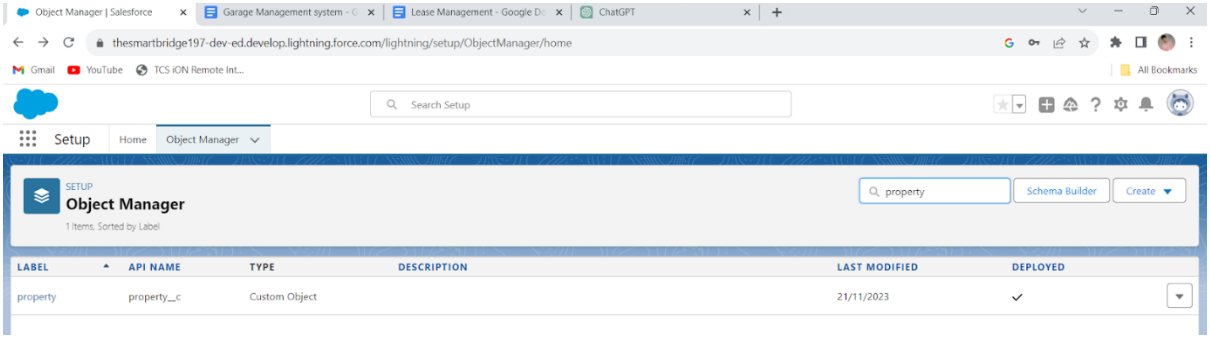
Custom Fields:

On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each 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.

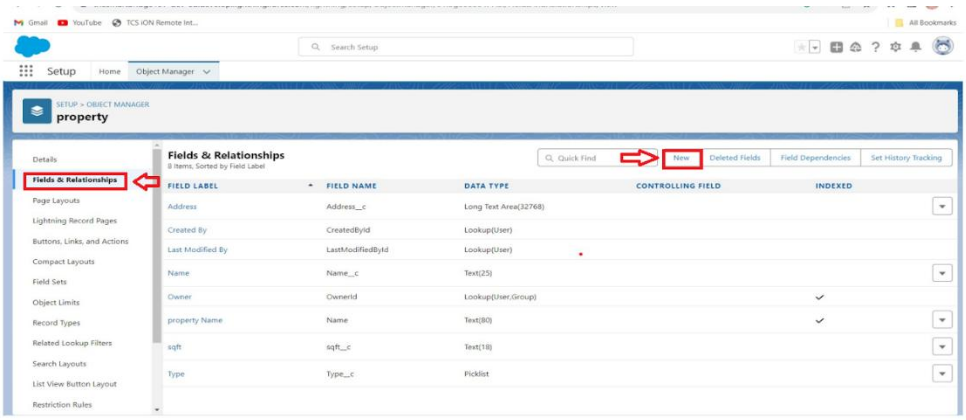
Creation of fields for the property object

To create fields in an object:

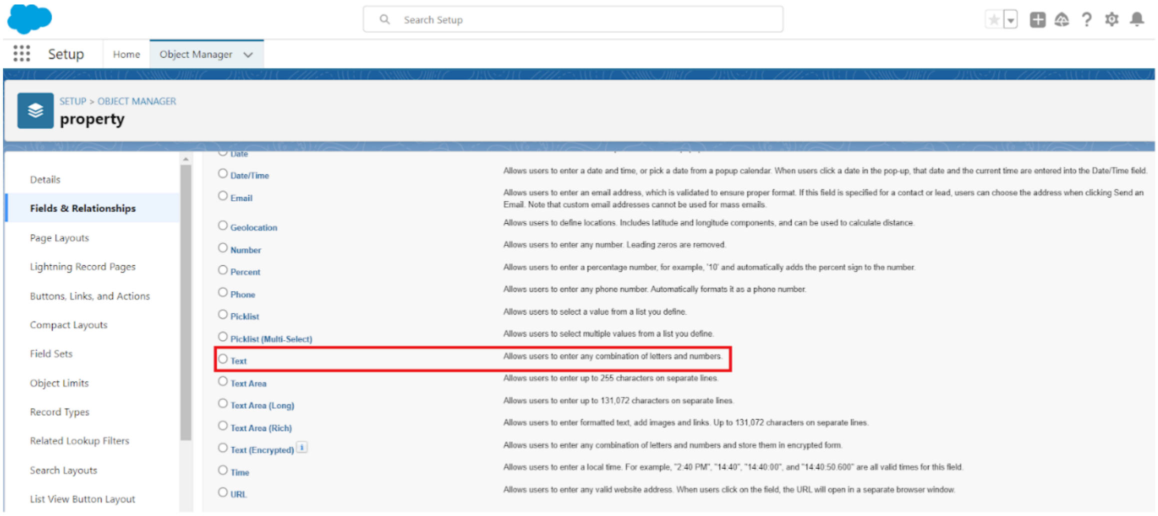
1. Go to setup >> click on Object Manager >> type object name(property) in search bar >>click on the object.



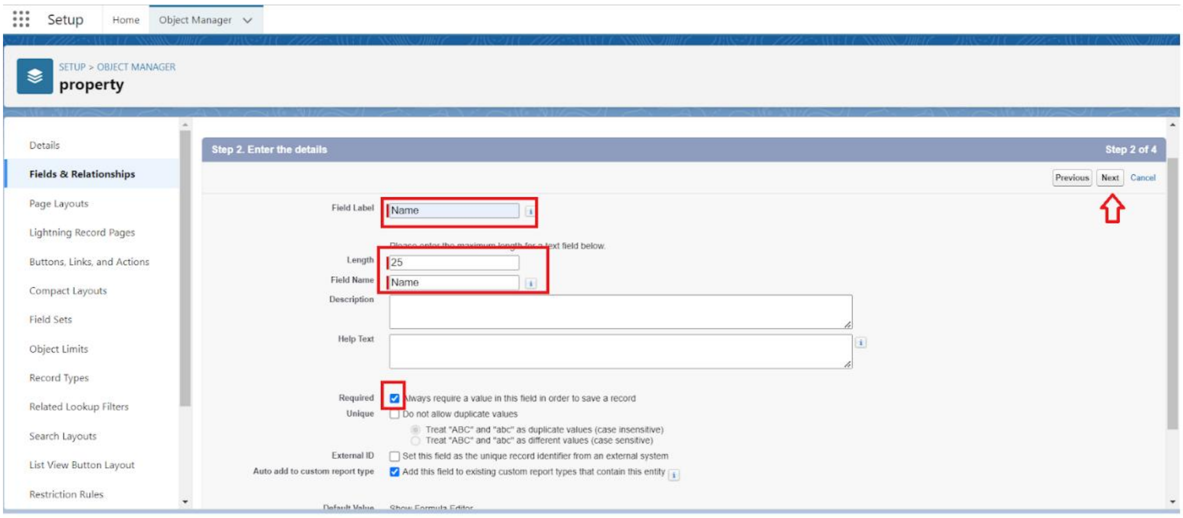
2. Now click on "Fields & Relationships" >> New



3. Select Data Type as a "Text"



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.

2. To create another fields in an object:

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

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

3. To create another fields in an object:

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

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

To create another fields in an object:

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

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

Creation of fields for the Tenant 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

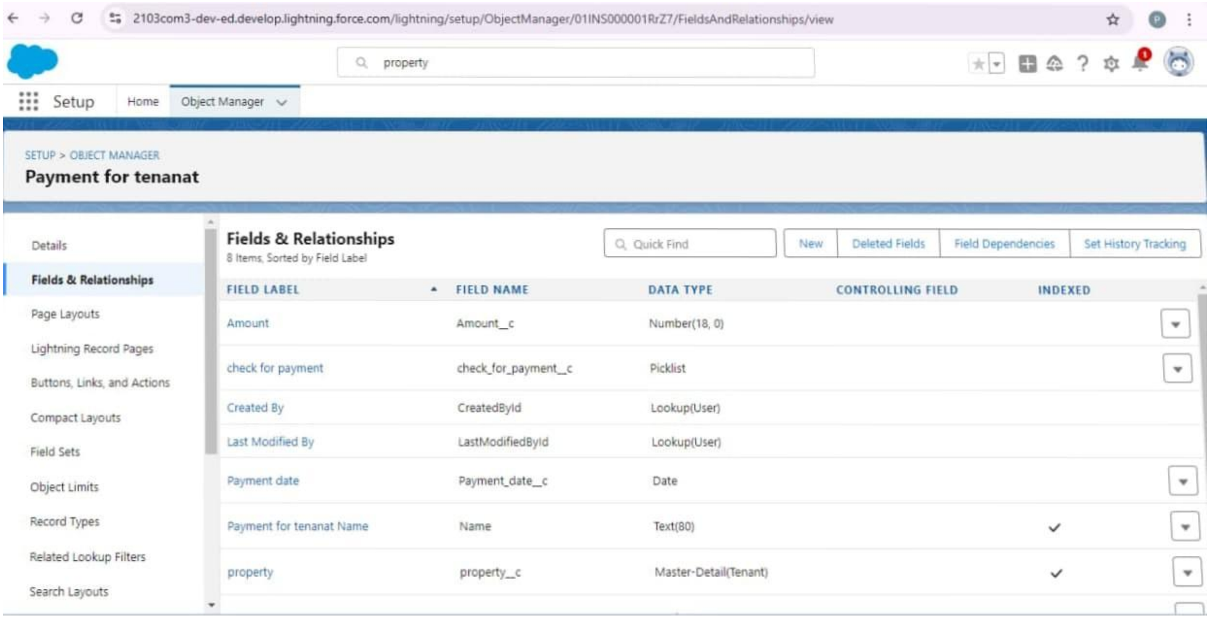
7. Select Data type as a “picklist” and Click on Next

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



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.

Creation of fields for the Payment for tenant object

.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

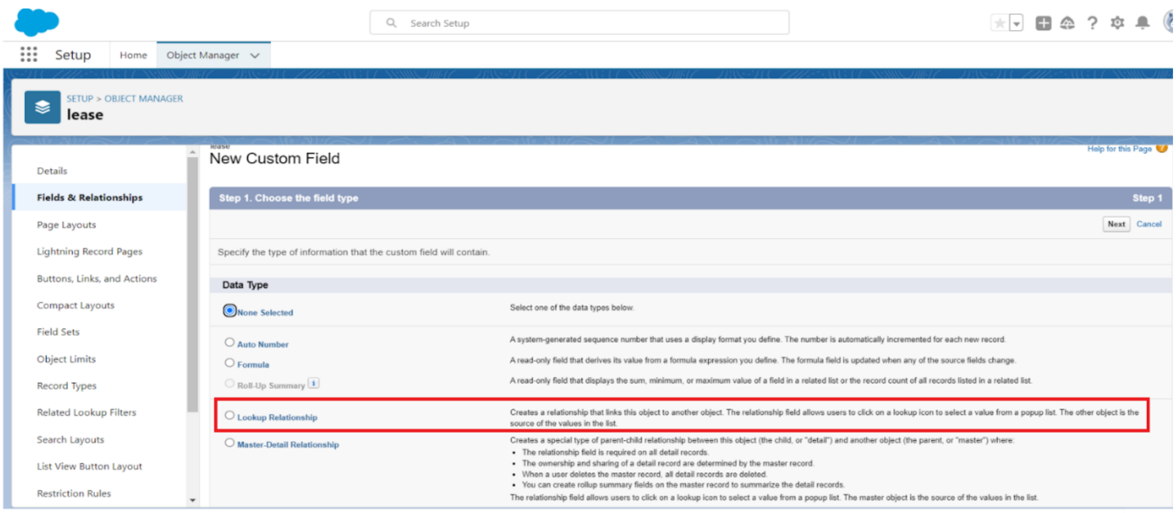
4. Fill the Above as following:

- 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.

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

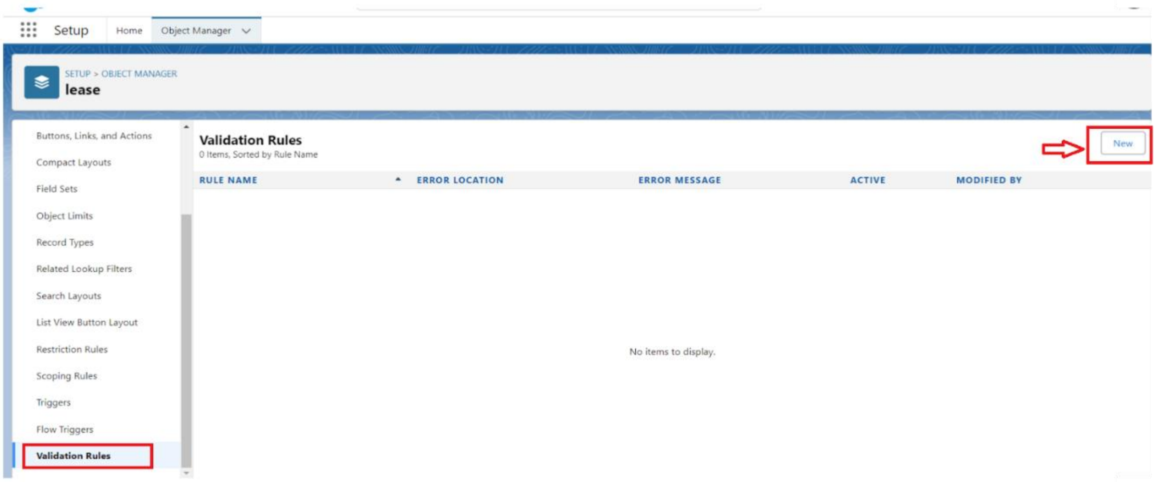
20. Field label : Auto generated

21. Next >> Next >> Save.

Validation rule

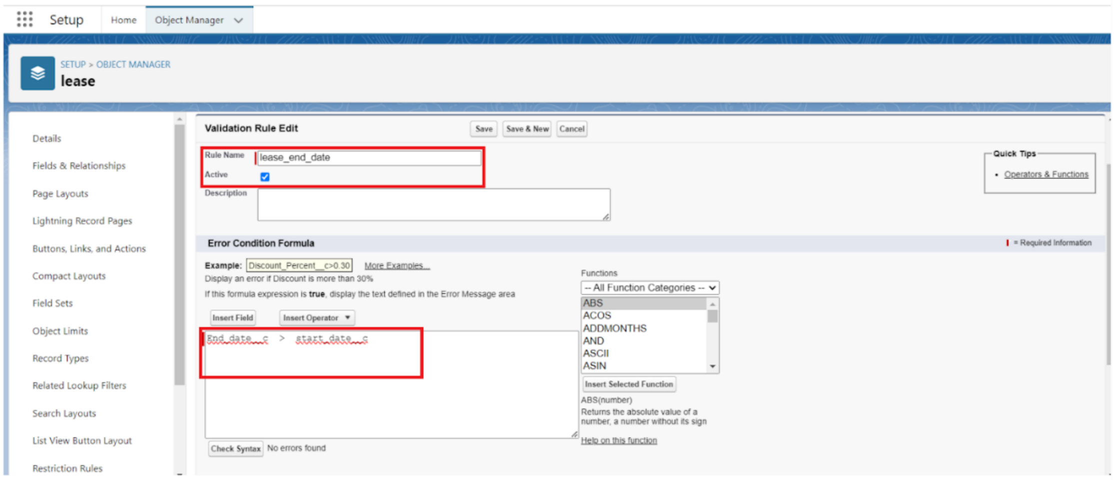
Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

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.

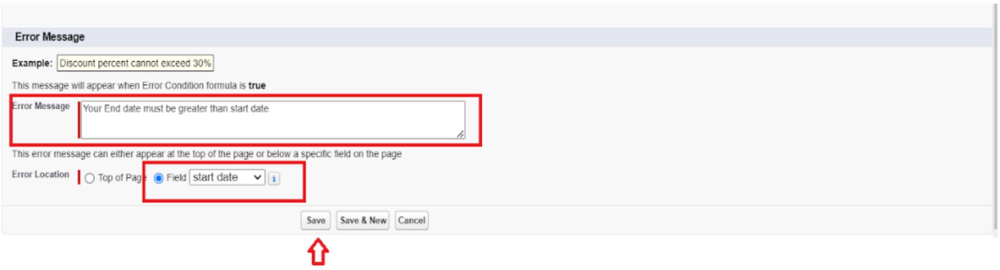


3. Enter the Rule name as “ lease_end_date”.

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.



Email Templates

Create Email Template For Tenant Leaving

To create Email Template:

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

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

   Folder : Unfiled public Classic Email templates

Click on available for use

3. Email Template Name is “tenant leaving”

4. Template Unique Name : Auto populated

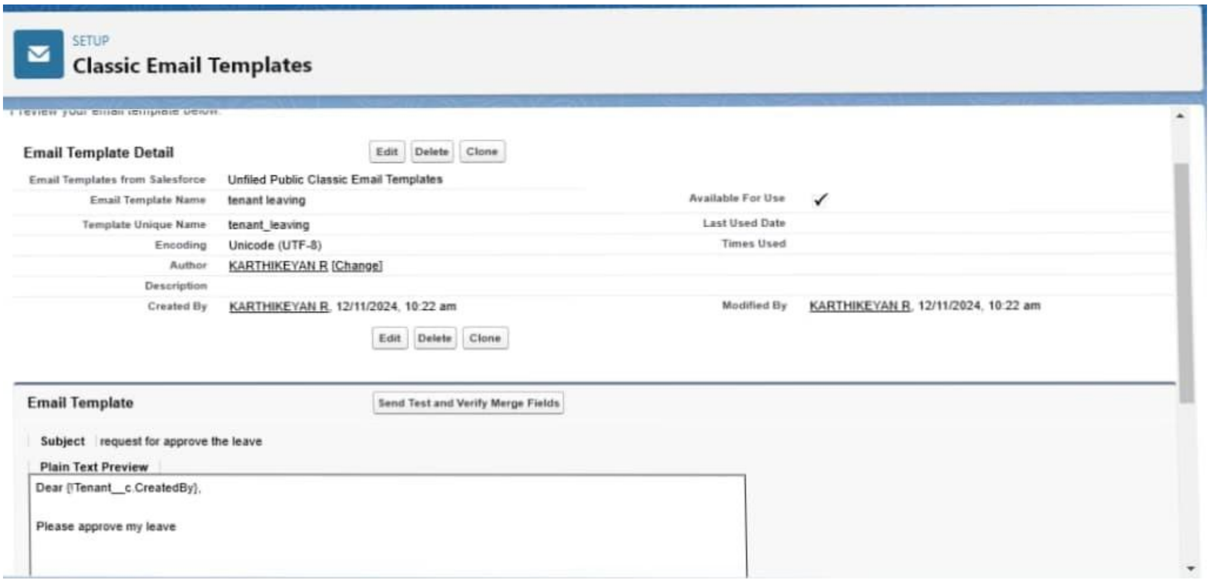
5. Subject : ” request for approve the leave”

6. Email body :

Dear {!Tenant__c.CreatedBy},

Please approve my leave

7. Save



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

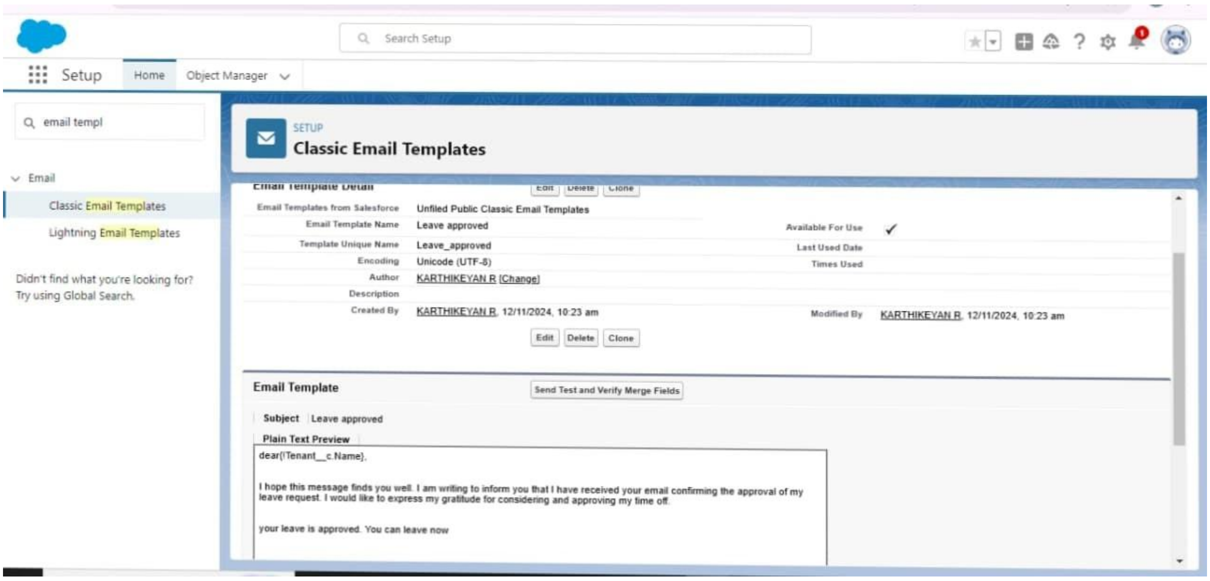
Click on available for use

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 email template >> click on classic Email Template.
2. Click on >>New Email Template===>Choose text

   Folder : Unfiled public Classic Email templates

Click on available for use

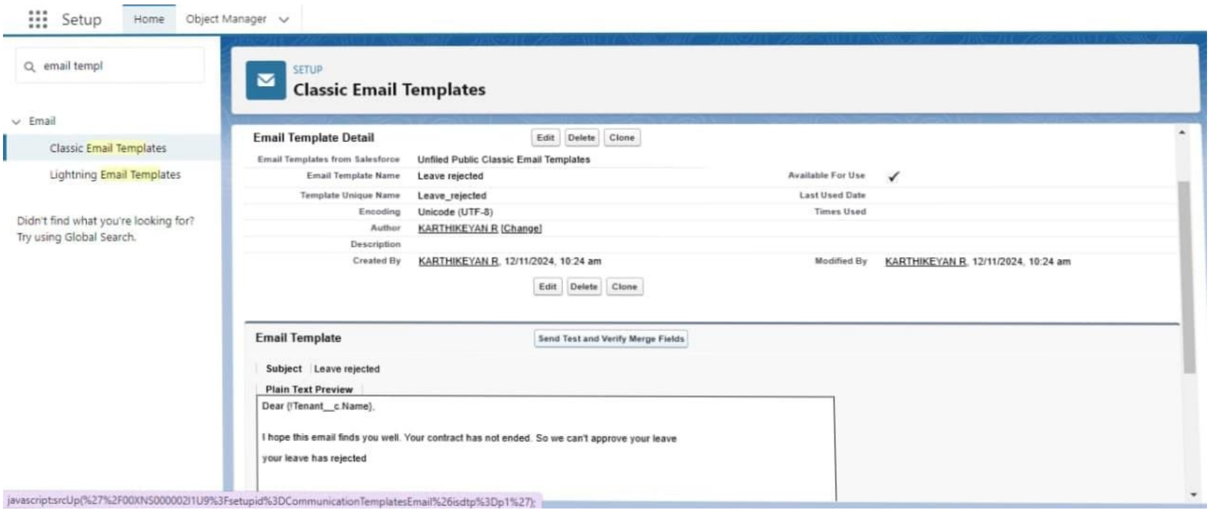
3. Email Template Name is "Leave rejected"
4. Template Unique Name : Auto populated
5. Subject : " Leave rejected"
6. Email body :

Dear {!Tenant__c.Name},

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

your leave has rejected

7. Save



Create Email Template For Monthly 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 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

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 {!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

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 alerts, field updates, tasks, or outbound messages.

2. Final Approval Actions

Final Approval actions are the actions that occur when a record is approved from all the approval steps. It also locks or unlocks the record, as specified. It can include any approval actions such as email alerts, field updates, tasks, or outbound messages.

3. Final Rejection Actions Final Rejection actions are the actions that occur when a record is rejected from any of the approval steps. It also locks or unlocks the record, as specified. It can include any approval actions such as email alerts, field updates, tasks, or outbound messages.

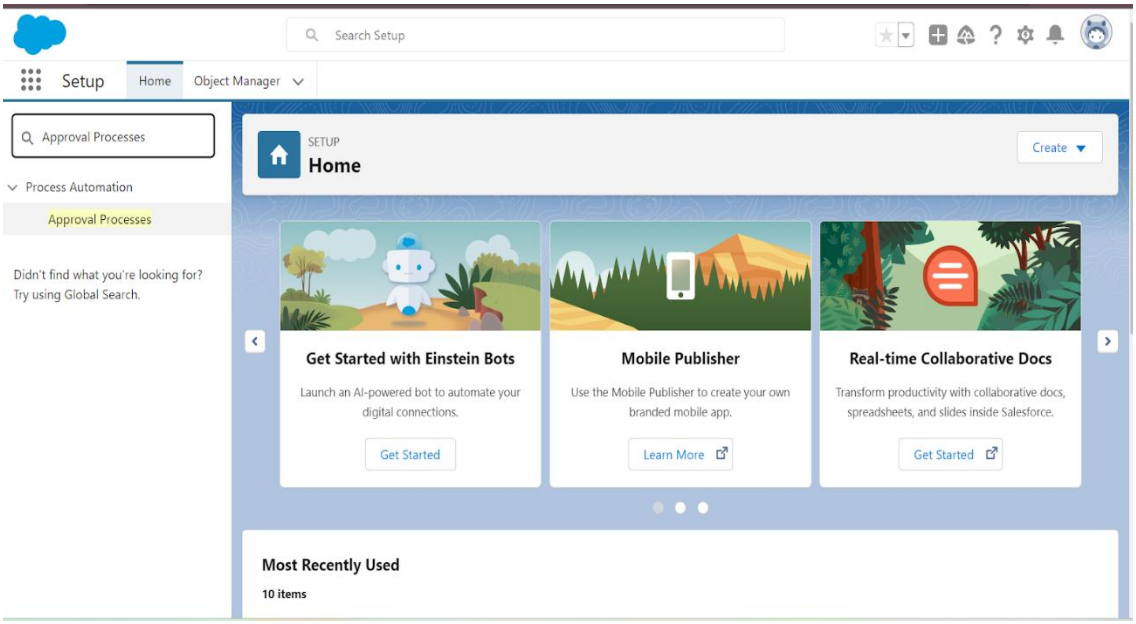
4. Recall Actions

Recall actions are the actions that occur when a record is recalled after submission for approval. It can include any approval actions such as email alerts, field updates, tasks, or outbound 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.



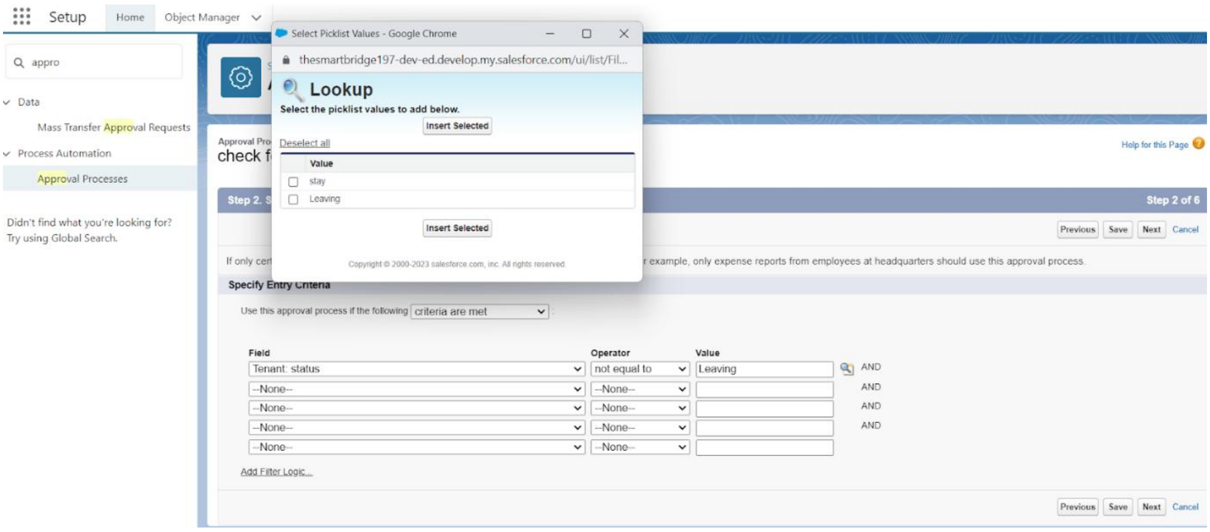
2.Manage Approval Process For >> "Tenant" from the drop down.

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

4. Process Name "check for vacant" >> Click Next.

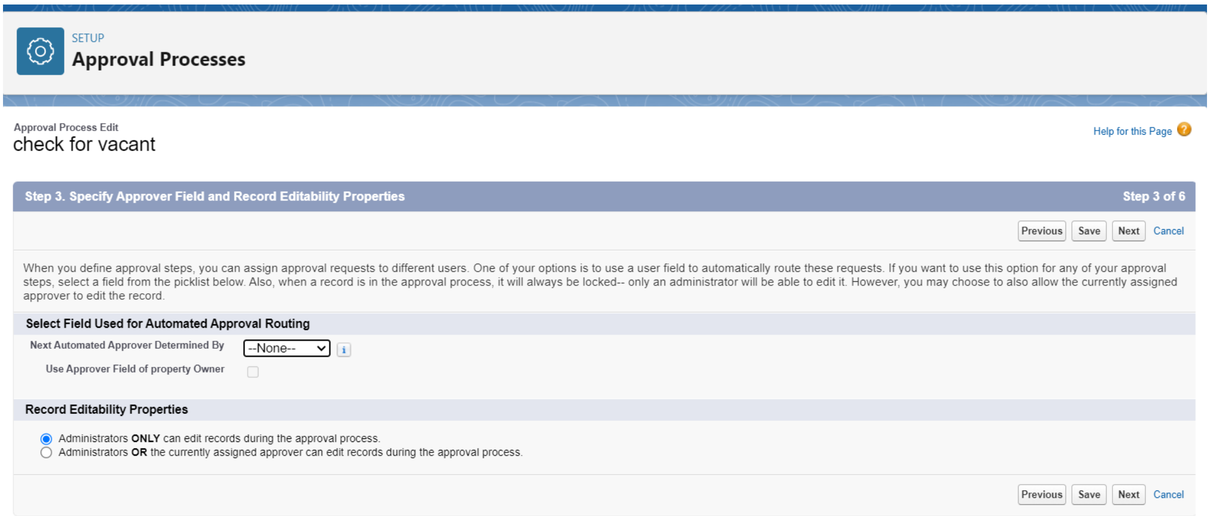
5. Field "Tenant:status" >> Operator : Not equals , Value >> Click on the lookup filter   icon and select "Leaving".

6.Click insert field,then click Next.



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

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



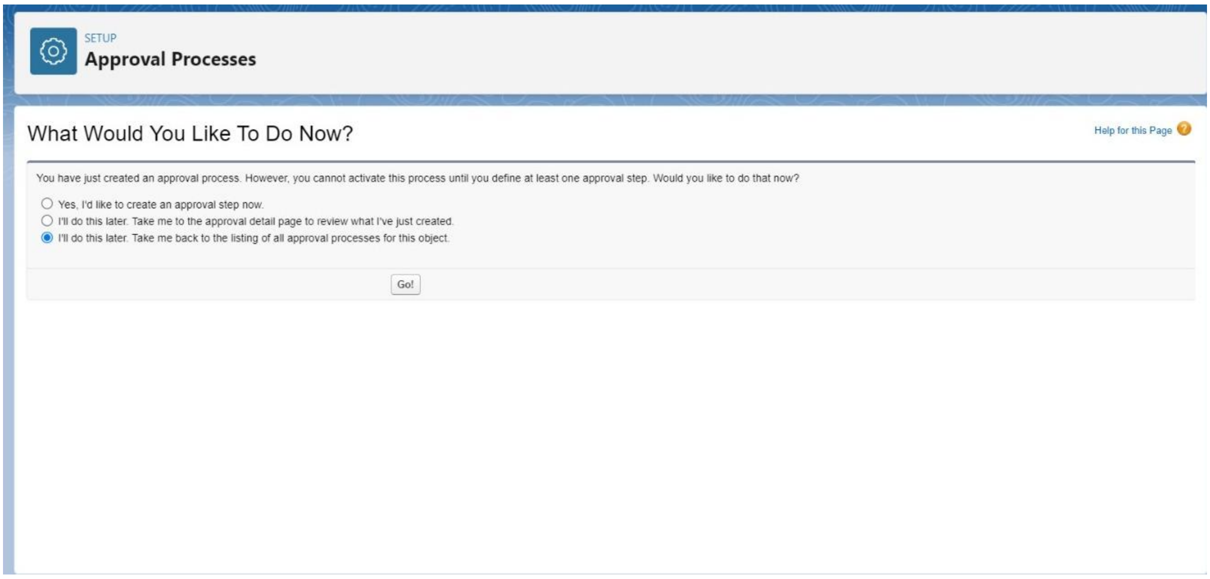
9. Click on next leave the email template click on next

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

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

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

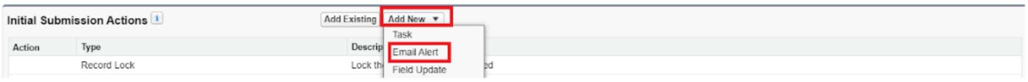
- Then click save.



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

Initial Submission 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 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

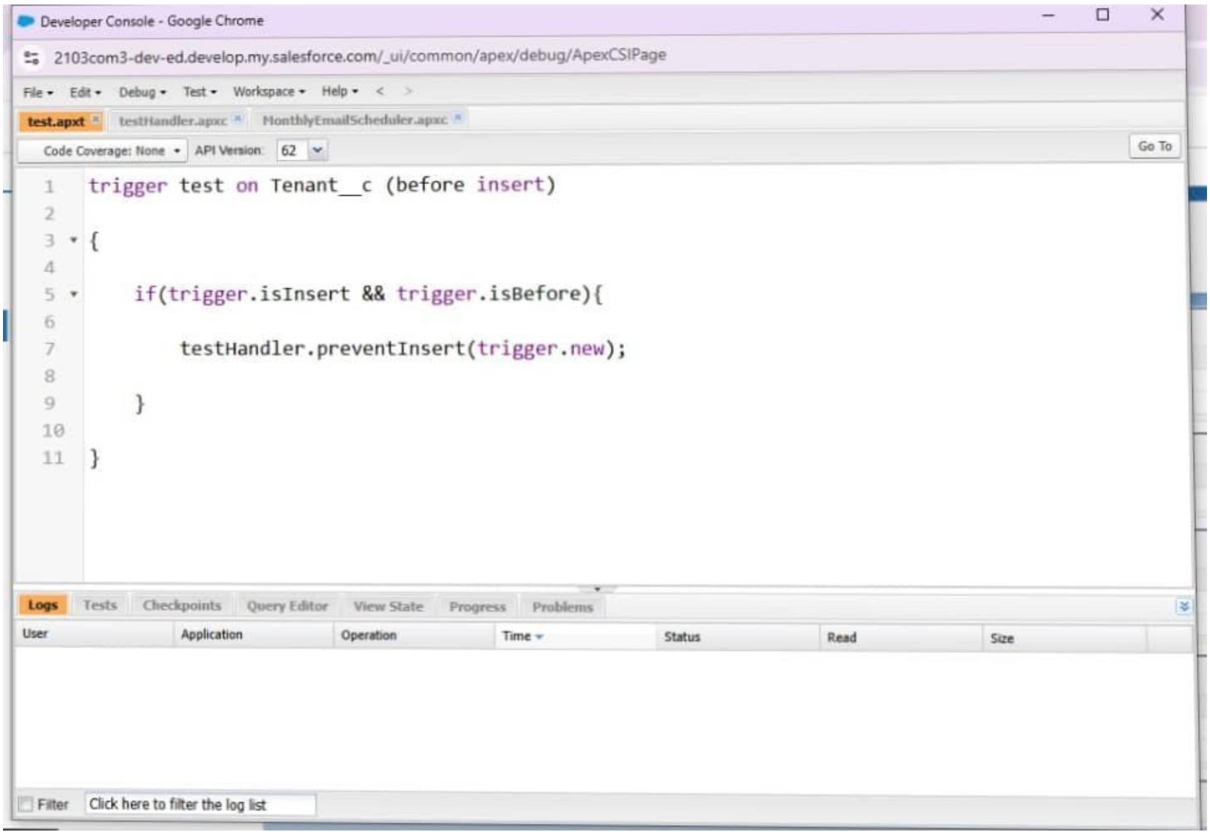
8. Click save

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.

Create an Apex Trigger

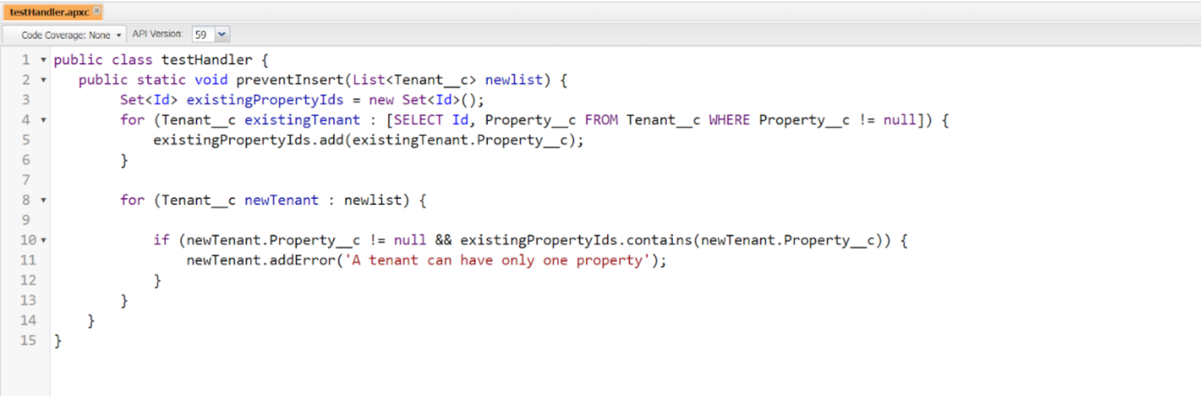


Create an Apex Handler class

To create a new Apex Class follow the below steps:

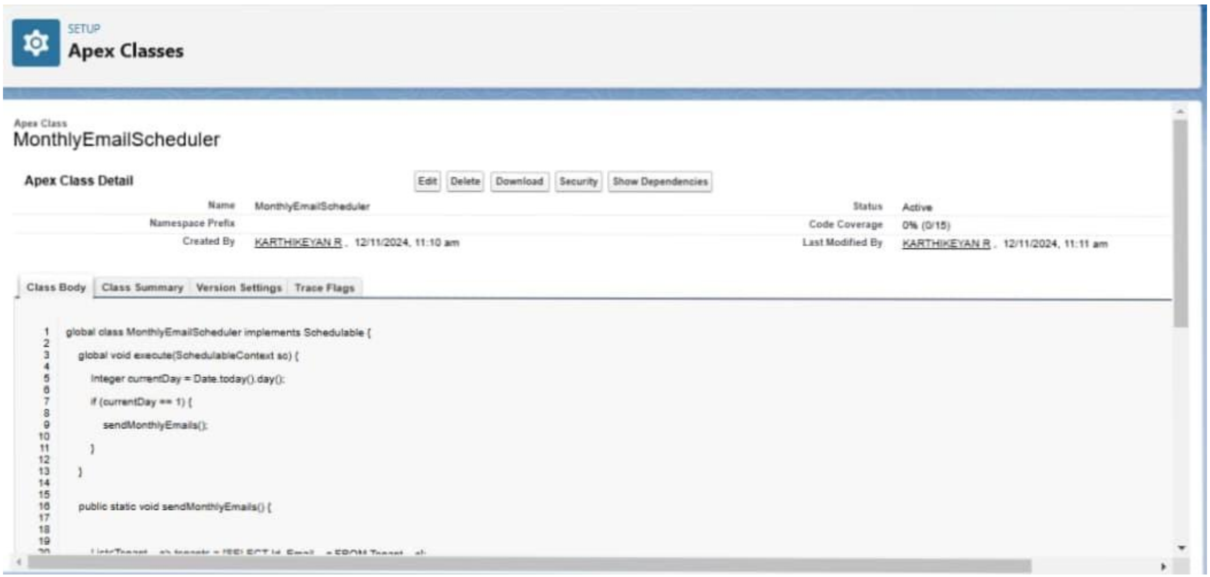
Click on the file >> New >>Apex Class.

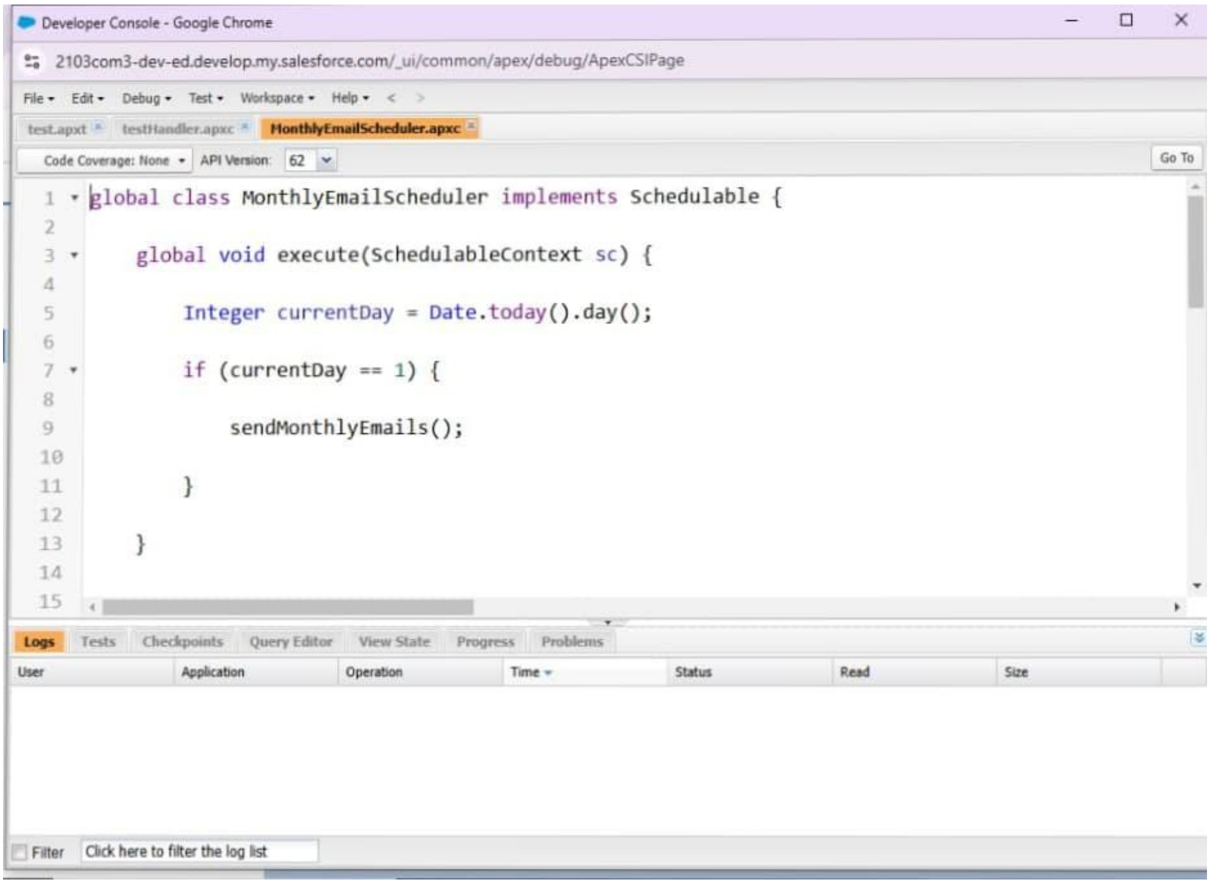
2. Enter class name as testHandler.



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');
          }
      }
  }
}
```





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.

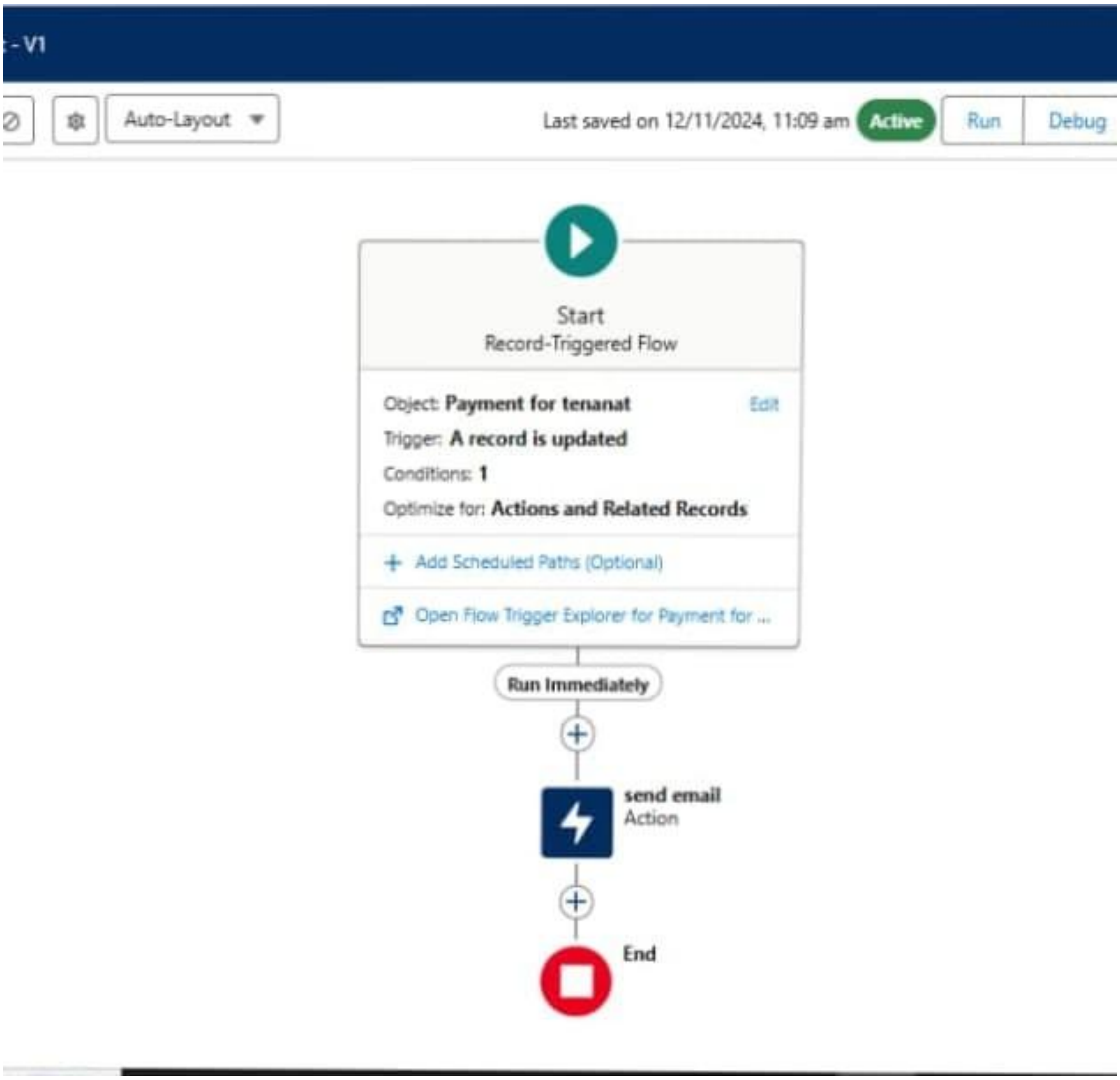
Autolaunched Flows: Run automated tasks with this flow type. Autolaunched flows can be invoked from other flows (subflow), process builder, from within an Apex class, from a set schedule, from record changes, or from platform events.

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.

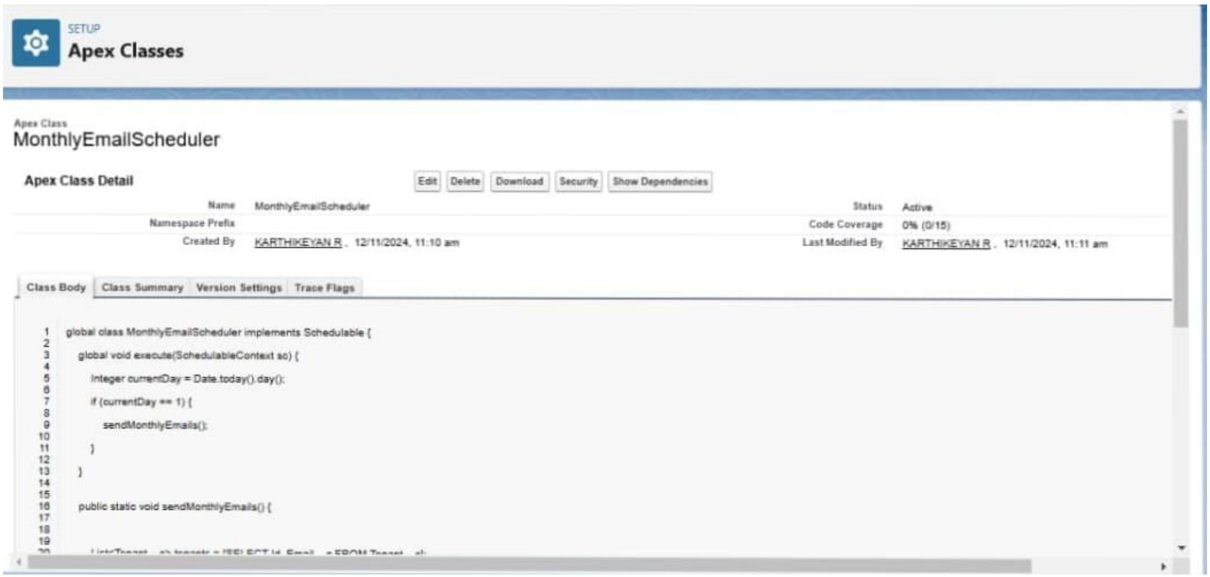
When and why should we use a flow

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.



Schedule class :

Create an Apex Class



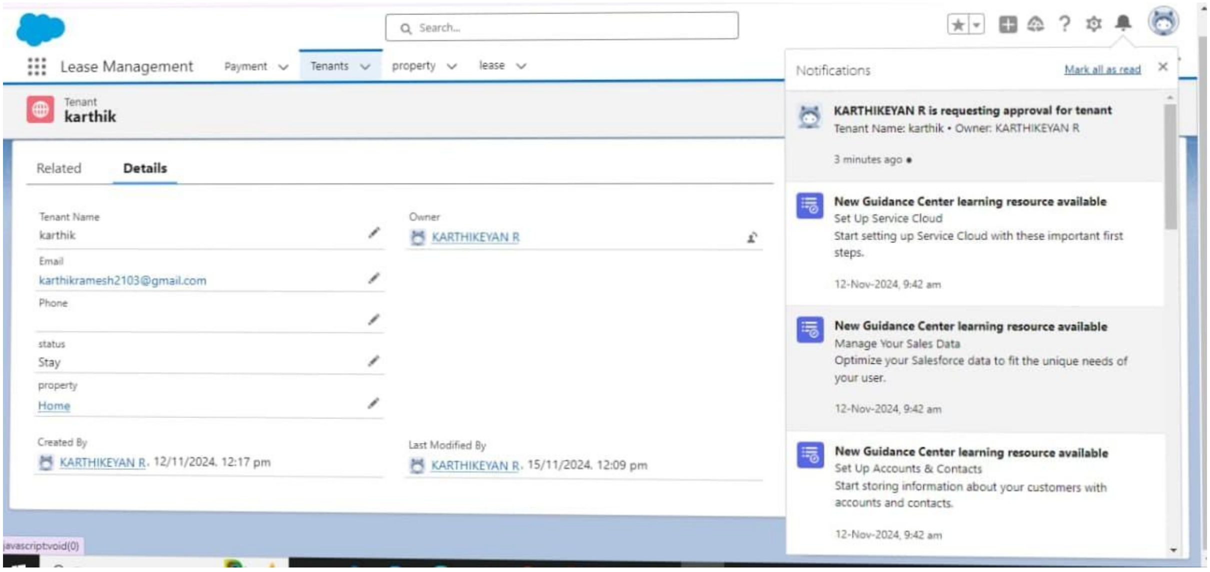
Apex logic:

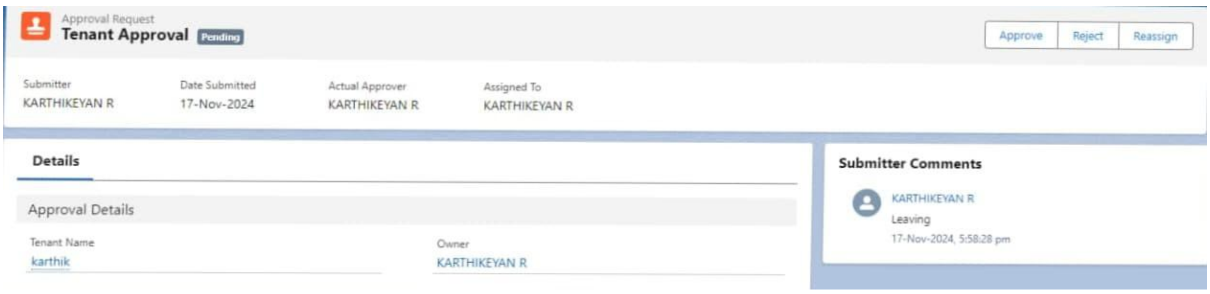
```
global class MonthlyEmailScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        Integer currentDay = Date.today().day();
        if (currentDay == 1) {
            sendMonthlyEmails();
        }
    }

    public static void sendMonthlyEmails() {

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



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

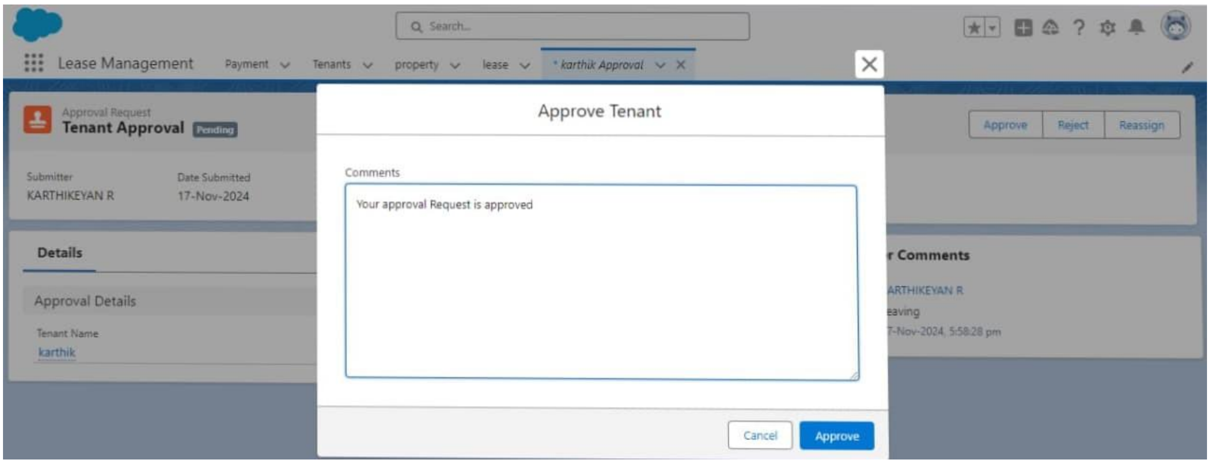
```
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
            email.setToAddresses(new String[]{recipientEmail});
            email.setSubject(emailSubject);
            email.setPlainTextBody(emailContent);

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

Save the code.

Schedule Apex class






Search...
Lease Management
Payment
Tenants
property
lease
* karthik Approval
Approval Request
Tenant Approval
Pending
Approve
Reject
Reassign
Submitter
KARTHIKEYAN R
Date Submitted
17-Nov-2024
Details
Approval Details
Tenant Name
karthik
Approve Tenant
Comments
Your approval Request is approved
Cancel
Approve