

Sales Force Developer with Agent Blazer ChampionTo Lease Management

**Vet Institute of Arts and Science (Co-education) College
College Code : 65651**

Department of Information Technology

Team ID : NM2025TMID25680

Team Size : 5

Team Leader : PREM RAHUL R

Email Id : premrahulr23sit@vetias.ac.in

Team Member : ALOYSIUS A

Email Id : aloysiysa23sit@vetias.ac.in

Team Member : RANJITH M

Email Id : ranjithm23sit@vetias.ac.in

Team Member : PRABU P

Email Id : prabup23sit@vetias.ac.in

Team Member : DHAMODHARAN P

Email Id : dhamodharanp23sit@vetias.ac.in

LEASE MANAGEMENT



The Least Management Project in Salesforce is a simplified approach to implementing CRM with only the most essential features required for managing sales and customer data. Instead of focusing on complex customizations and advanced automation, this project emphasizes minimal configuration and core functionalities such as lead capture, opportunity tracking, contact management, and basic reporting. By keeping management tasks to a minimum, the project ensures faster deployment, reduced maintenance, and an easy-to-use system for teams. This approach is particularly suited for small organizations or businesses that want to adopt Salesforce with low complexity while still improving efficiency and productivity.

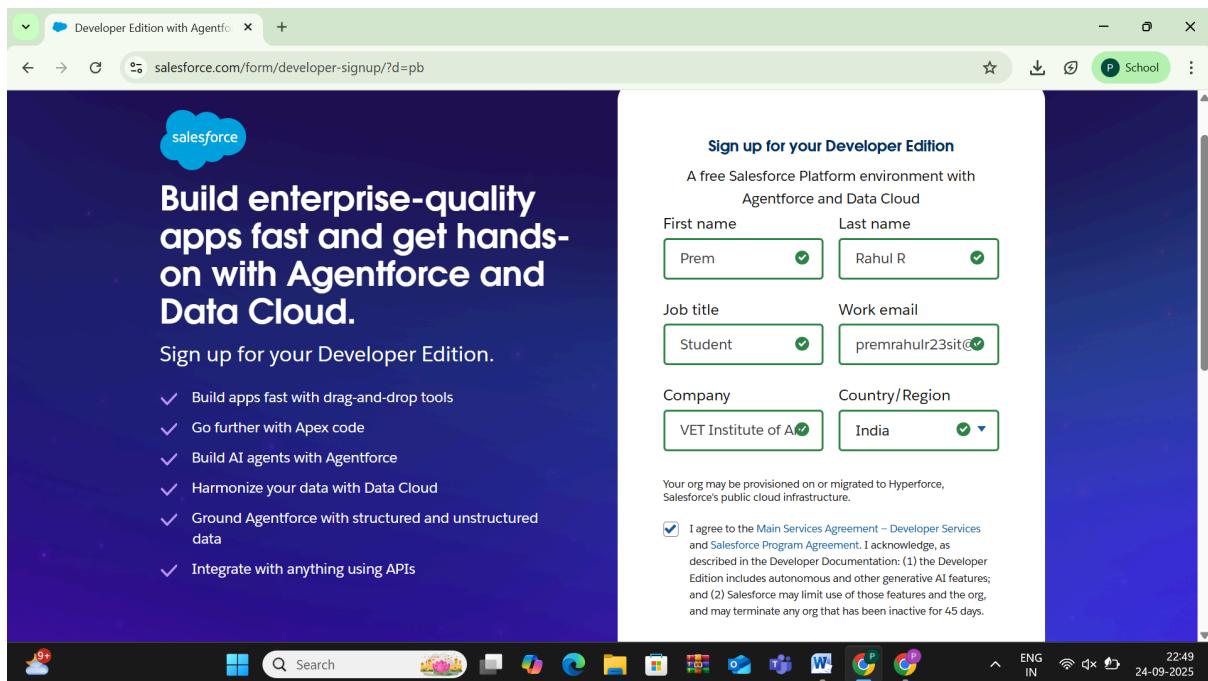
Salesforce

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 :

- First name & Last name
 - Email
 - Role : Developer
 - Company : College Name
 - County : India
 - Postal Code : pin code
 - 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:

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.



Hi Prem,

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

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

[Reset Password](#)

To easily log in later, save this URL:

<https://orgfarm-fd5ecb6d3d-dev-ed.develop.my.salesforce.com>

Here's the username for your Developer Edition:
premrahulr23sit985@agentforce.com

Your Developer Edition, now enabled with Agentforce and Data Cloud, remains active as long as you continue to use it. It expires after 45 days of non-usage.

Again, welcome to Salesforce!
Developer Relations

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

A screenshot of the Salesforce Setup Home page. The page has a blue header with the word "SETUP" and a "Home" button. On the left, there's a sidebar with links like "Setup Home", "Salesforce Go", "Service Setup Assistant", etc. The main content area shows three cards: "Data Cloud" (with "Watch Video" and "Let's Go" buttons), "Get Started with Einstein Bots" (with "Get Started" button), and "Mobile Publisher" (with "Learn More" button). Below these cards is a section titled "Most Recently Used" with a table showing 10 items. The table has columns for "NAME", "TYPE", and "OBJECT". The last item in the table is "MonthlyEmailScheduler" with type "Apex Class". A small calendar icon in the bottom right corner shows "25 September 2025" and "Thu 13:48 (Local time)".

Object:

Create Property Object:

To create an object:

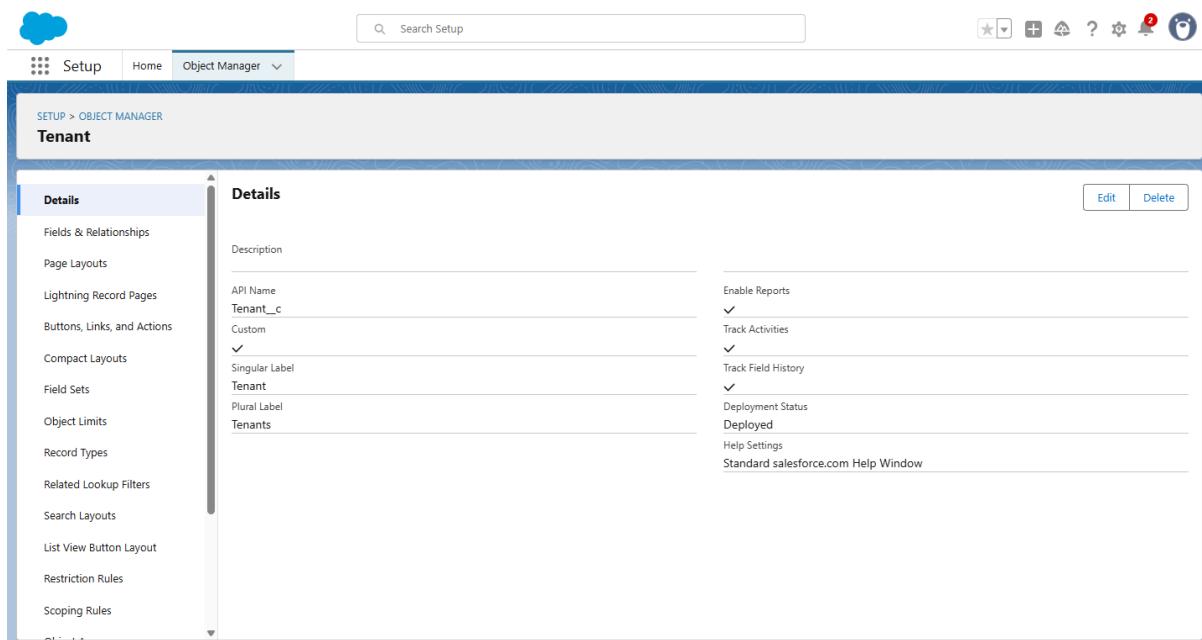
1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
 1. Enter the label name >> property
 2. Plural label name >> property
 3. Enter Record Name Label and Format
→ Record Name >> property Name
→ Data Type >> Text
2. Click on Allow reports and Track Field History,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

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Property'. On the left, a sidebar lists various object configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The right side displays the 'Details' section for the 'Property' object. It contains fields for 'Description' (empty), 'API Name' (set to 'property__c'), 'Custom' (checkbox checked), 'Singular Label' (set to 'Property'), 'Plural Label' (set to 'property'), and 'Enable Reports' (checkbox checked). Other settings like 'Track Activities', 'Track Field History', 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window') are also shown. At the bottom right of the details section are 'Edit' and 'Delete' buttons.

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.

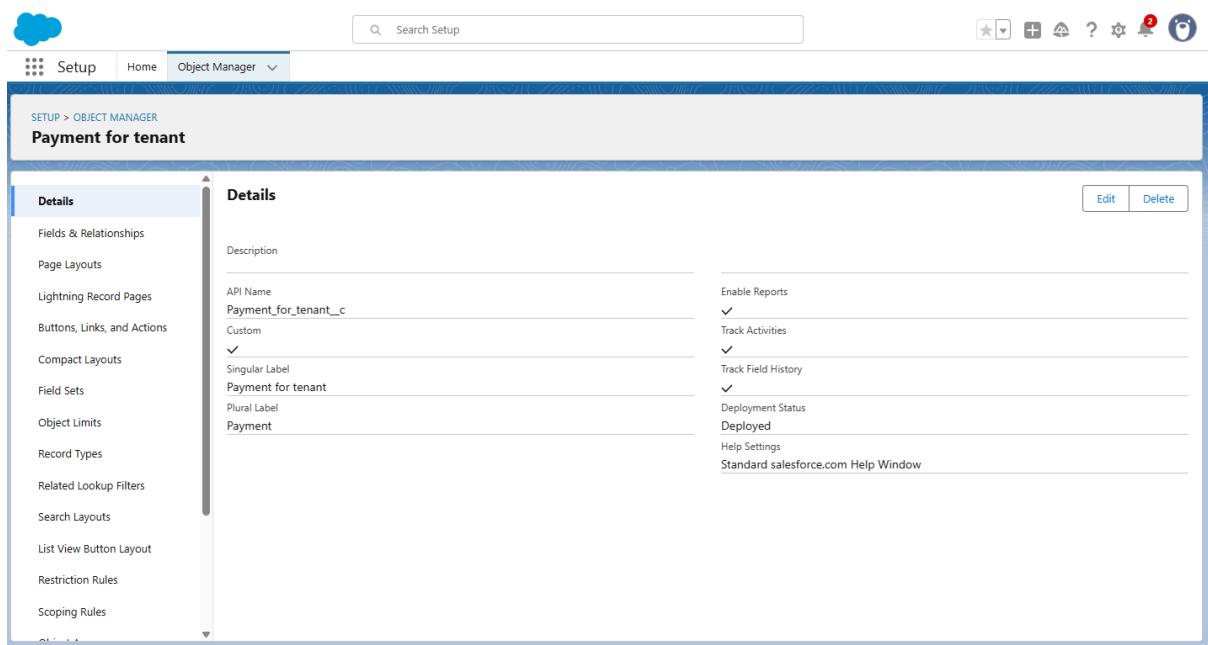


Create Payment Object:

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 tenant

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:

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)

Action	Label	Tab Style	Description
Edit Del	lease		Handsaw
Edit Del	Payment		Compass
Edit Del	property		Bottle
Edit Del	Tenants		Diamond

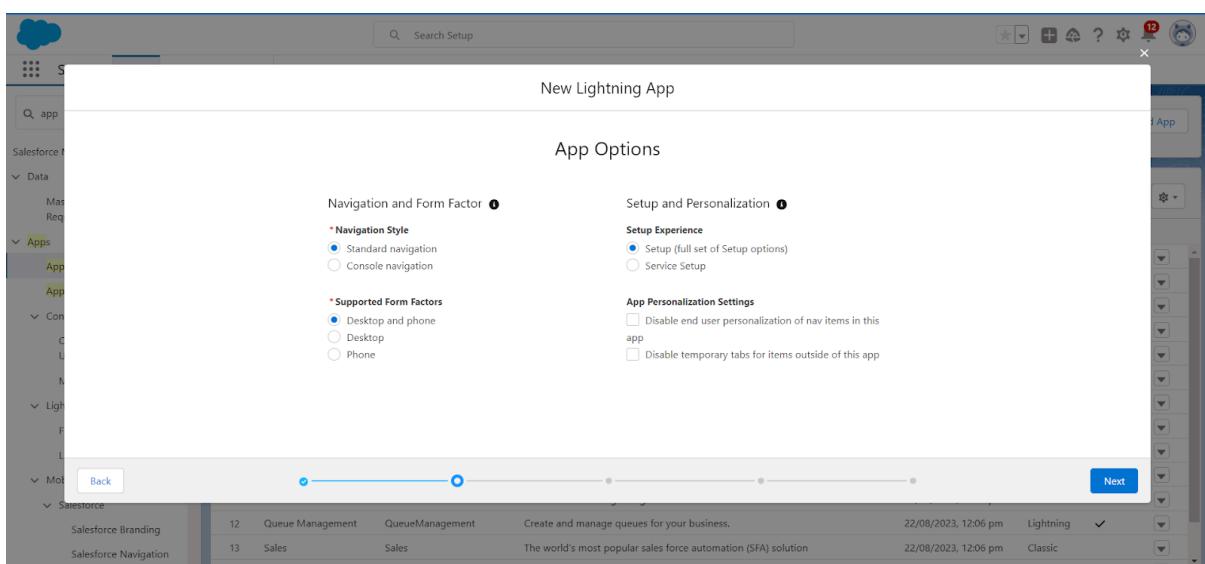
1. Select Object(property) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
2. Make sure that the Append tab to users' existing personal customizations is checked.
3. Click save.

The Lightning App:

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:

The screenshot shows the 'Available Items' and 'Selected Items' sections of the Navigation Item configuration screen. The 'Available Items' section contains a search bar and a list of items including Accounts, Alert Settings, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests, and Asset Action Sources. The 'Selected Items' section contains three items: 'Payment for tenant', 'Tenants', and 'property'. Arrows on the right side of the 'Selected Items' list indicate the ability to move items up and down.

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

Fields:

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.

The screenshot shows the 'Object Manager' screen in Salesforce Setup. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main area displays a table with one item: 'Property' (Label), 'property_c' (API Name), 'Custom Object' (Type), and '9/8/2025' (Last Modified). A 'Create' button is visible at the top right of the table area.

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

The screenshot shows the Salesforce Setup interface with the Object Manager selected. In the left sidebar, 'Fields & Relationships' is highlighted. The main area displays a table titled 'Fields & Relationships' with 8 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name_c	Text(25)		
Owner	OwnerId	Lookup(User,Group)		✓
property Name	Name	Text(80)		✓
sft	sft_c	Text(18)		

3. Select Data Type as a “Text”

The screenshot shows the Salesforce Setup interface with the Object Manager selected. In the left sidebar, 'Fields & Relationships' is highlighted. The main area displays a list of data types with their descriptions:

- Geolocation
- Number
- Percent
- Phone
- Picklist
- Picklist (Multi-Select)
- Text
- Text Area
- Text Area (Long)
- Text Area (Rich)
- Text (Encrypted) (i)
- Time
- URL

Descriptions for each type are provided on the right side of the list.

4. Click on next

The screenshot shows the Salesforce Setup interface with the Object Manager selected. In the left sidebar, 'Fields & Relationships' is highlighted. The main area is titled 'New Custom Field' and shows 'Step 2. Enter the details'. The form fields are:

- Field Label:
- Length: Please enter the maximum length for a text field below.
- Field Name:
- Description:
- Help Text:

At the top right of the form, there are buttons for 'Previous', 'Next', and 'Cancel'. A 'Help for this page' link is also visible at the top right.

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

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

The screenshot shows the Salesforce Setup interface. At the top, there's a navigation bar with icons for Home, Object Manager, and other setup options. Below it, a search bar says "Search Setup". The main area is titled "SETUP > OBJECT MANAGER" and shows "Tenant". On the left, a sidebar lists various configuration tabs: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The right side shows the "Details" tab for the Tenant object. It includes fields for Description, API Name (set to "Tenant__c"), Singular Label (set to "Tenant"), Plural Label (set to "Tenants"), and several checkboxes for features like Enable Reports, Track Activities, Track Field History, Deployment Status (set to "Deployed"), and Help Settings (linking to "Standard salesforce.com Help Window"). There are "Edit" and "Delete" buttons at the bottom right of the details panel.

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:

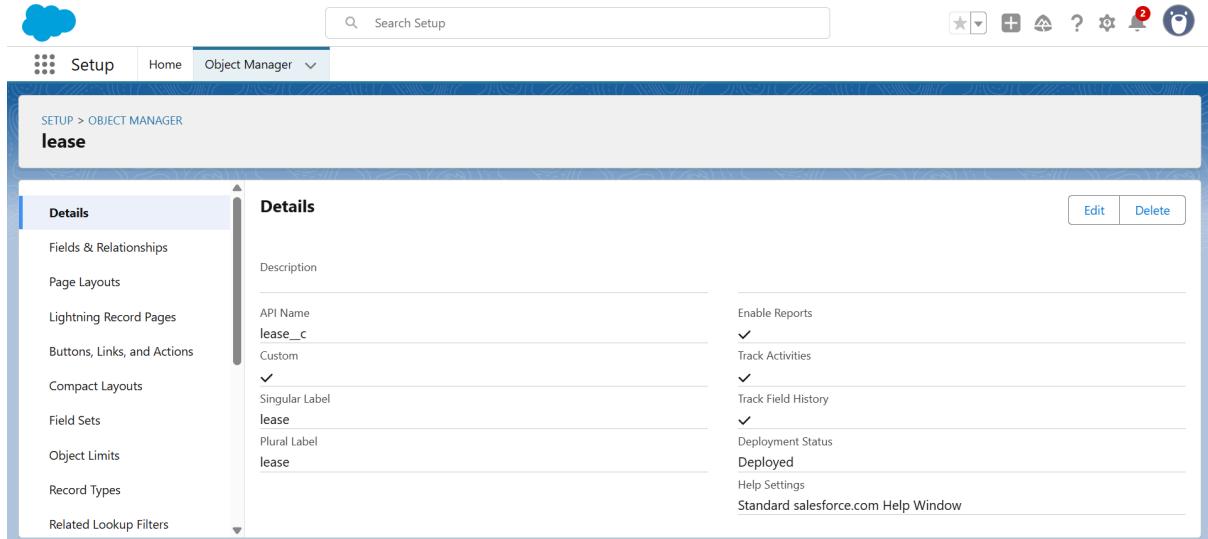
1. Go to setup >> click on Object Manager >> type object name(Tenant) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “phone” and Click on Next
4. **Fill the Above as following:**
 - Field Label : Phone
 - Field Name : gets auto generated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

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

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

The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with a cloud icon, 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Setup'. To the right are various icons for configuration. Below the bar, the title 'SETUP > OBJECT MANAGER' and the object name 'Payment for tenant' are displayed. On the left, a sidebar lists options like 'Fields & Relationships', 'Page Layouts', 'Lightning Record Pages', etc. The main area is titled 'Details' and shows the following fields:

- Description
- API Name: Payment_for_tenant_c
- Custom: ✓
- Singular Label: Payment for tenant
- Plural Label: Payment
- Enable Reports: ✓
- Track Activities: ✓
- Track Field History: ✓
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

At the bottom right are 'Edit' and 'Delete' buttons.

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.

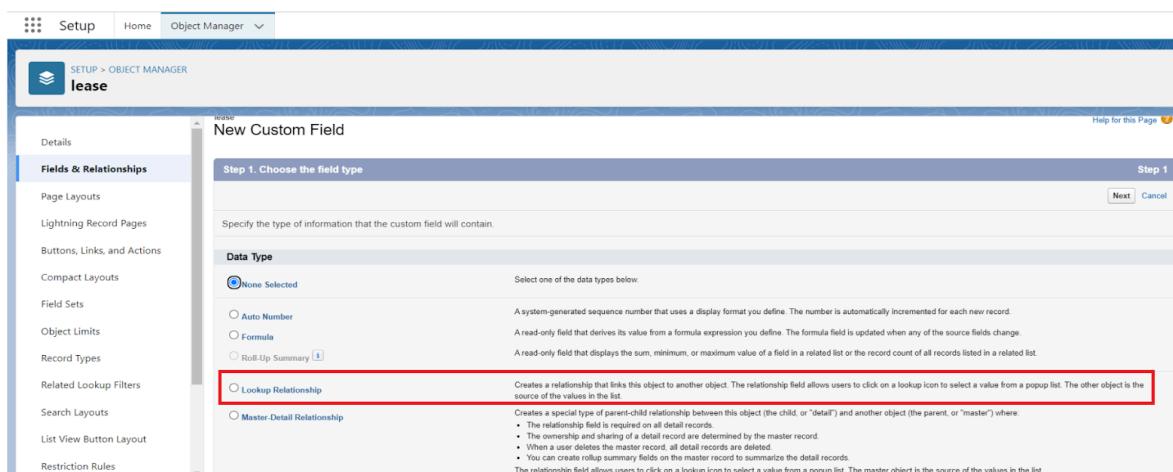
To create another fields in an object:

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

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.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes a cloud icon, 'Setup', 'Home', and 'Object Manager'. The 'Object Manager' tab is selected, showing the 'lease' object. On the left, a sidebar lists various configuration options: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Validation Rules' and displays one item: 'lease_end_date' (Rule Name), 'start date' (Error Location), 'Your End date must be greater than start date' (Error Message), 'Active' (Status), and 'Prem Rahul R, 9/9/2025, 1:31 AM' (Modified By). A 'New' button is visible in the top right corner of the list.

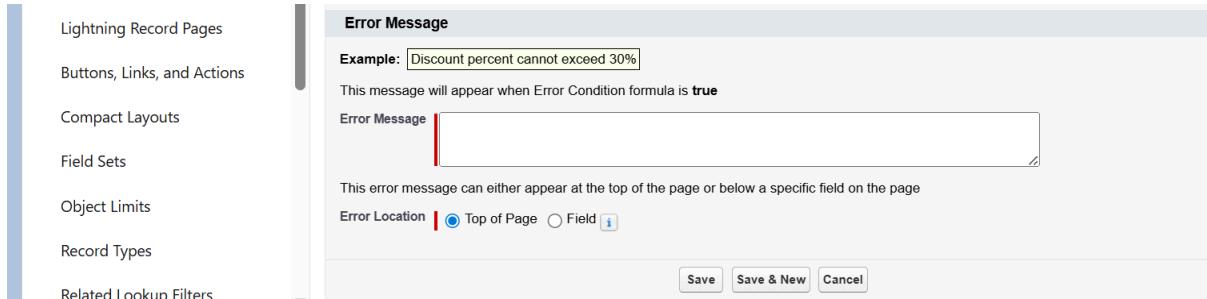
3. Enter the Rule name as “ lease_end_date ”.

4. Insert the Error Condition Formula as :

End_date__c > start_date__c

The screenshot shows the 'Validation Rule Edit' screen for the 'lease' object. The top navigation bar is identical to the previous screenshot. The main content area is titled 'lease Validation Rule'. It contains a help link 'Help for this Page' and a note: 'Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.' Below this is a 'Validation Rule Edit' form with fields for 'Rule Name' (containing 'lease_end_date'), 'Active' (checked), and 'Description'. To the right is a 'Quick Tips' box with 'Operators & Functions'. At the bottom is an 'Error Condition Formula' section with an example 'Discount_Percent__c>0.30', a 'More Examples...' link, and a note: 'Display an error if Discount is more than 30%. If this formula expression is true, display the text defined in the Error Message area'. A 'Functions' dropdown shows 'All Function Categories' with 'ABS' and 'ACOS' listed. There are also 'Insert Field' and 'Insert Operator' buttons.

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

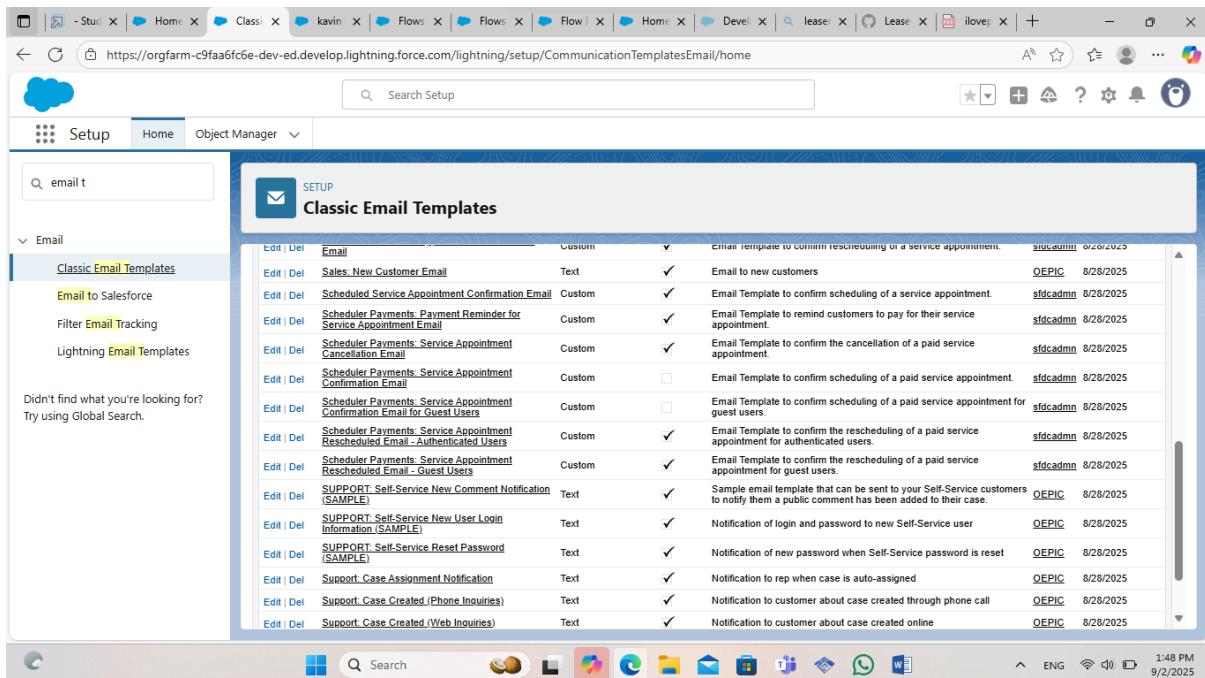
To create Email Template:

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.



The screenshot shows the Salesforce Setup interface with the search bar set to "email t". The left sidebar has "Email" expanded, with "Classic Email Templates" selected. The main area displays a table titled "Classic Email Templates" with the following data:

Action	Name	Type	Custom	Description	Created By	Created Date
Edit Del	Sales_New_Customer_Email	Text	✓	Email to new customers	OEPIC	8/28/2025
Edit Del	Scheduled_Service_Appointment_Confirmation_Email	Custom	✓	Email Template to confirm scheduling of a service appointment.	sfdcadmin	8/28/2025
Edit Del	Scheduler_Payments_Payment_Reminder_for_Service_Appointment_Email	Custom	✓	Email Template to remind customers to pay for their service appointment.	sfdcadmin	8/28/2025
Edit Del	Scheduler_Payments_Service_Appointment_Cancellation_Email	Custom	✓	Email Template to confirm the cancellation of a paid service appointment.	sfdcadmin	8/28/2025
Edit Del	Scheduler_Payments_Service_Appointment_Confirmation_Email	Custom	□	Email Template to confirm scheduling of a paid service appointment.	sfdcadmin	8/28/2025
Edit Del	Scheduler_Payments_Service_Appointment_Confirmation_Email_for_Guest_Users	Custom	□	Email Template to confirm scheduling of a paid service appointment for guest users.	sfdcadmin	8/28/2025
Edit Del	Scheduler_Payments_Service_Appointment_Rescheduled_Email - Authenticated_Users	Custom	✓	Email Template to confirm the rescheduling of a paid service appointment for authenticated users.	sfdcadmin	8/28/2025
Edit Del	Scheduler_Payments_Service_Appointment_Rescheduled_Email - Guest_Users	Custom	✓	Email Template to confirm the rescheduling of a paid service appointment for guest users.	sfdcadmin	8/28/2025
Edit Del	SUPPORT_Self-Service_New_Comment_Notification_(SAMPLE)	Text	✓	Sample email template that can be sent to your Self-Service customers to notify them a public comment has been added to their case.	OEPIC	8/28/2025
Edit Del	SUPPORT_Self-Service_New_User_Login_Information_(SAMPLE)	Text	✓	Notification of login and password to new Self-Service user	OEPIC	8/28/2025
Edit Del	SUPPORT_Self-Service_Reset_Password_(SAMPLE)	Text	✓	Notification of new password when Self-Service password is reset	OEPIC	8/28/2025
Edit Del	Support_Case_Assignment_Notification	Text	✓	Notification to rep when case is auto-assigned	OEPIC	8/28/2025
Edit Del	Support_Case_Created_(Phone_Inquiries)	Text	✓	Notification to customer about case created through phone call	OEPIC	8/28/2025
Edit Del	Support_Case_Created_(Web_Inquiries)	Text	✓	Notification to customer about case created online	OEPIC	8/28/2025

Create Email Template For Leave Approved

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”

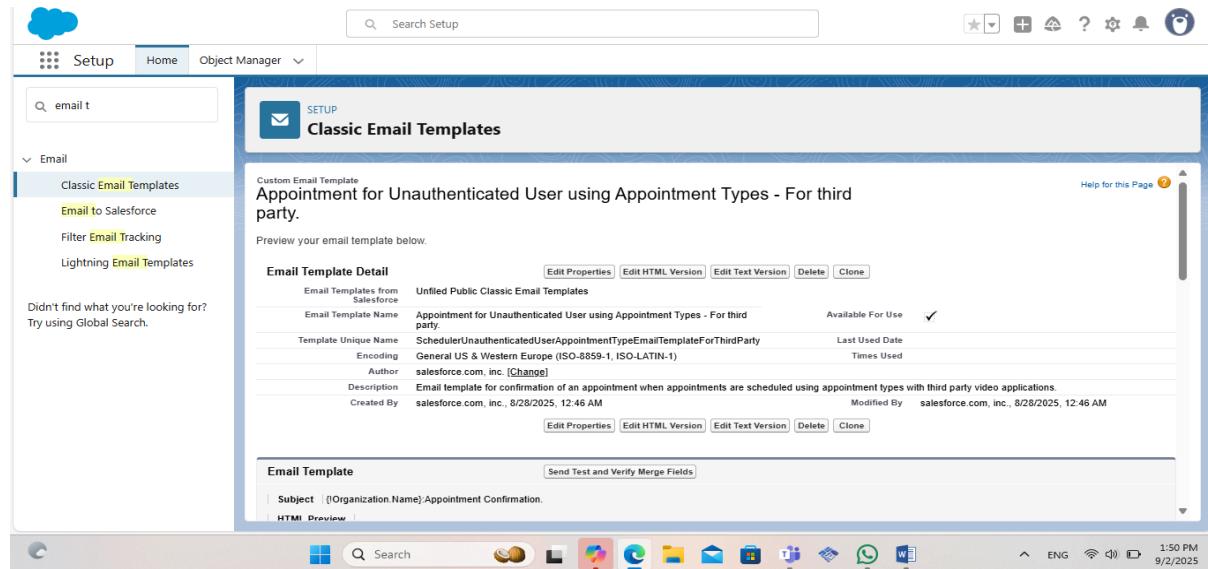
6. Email body :

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

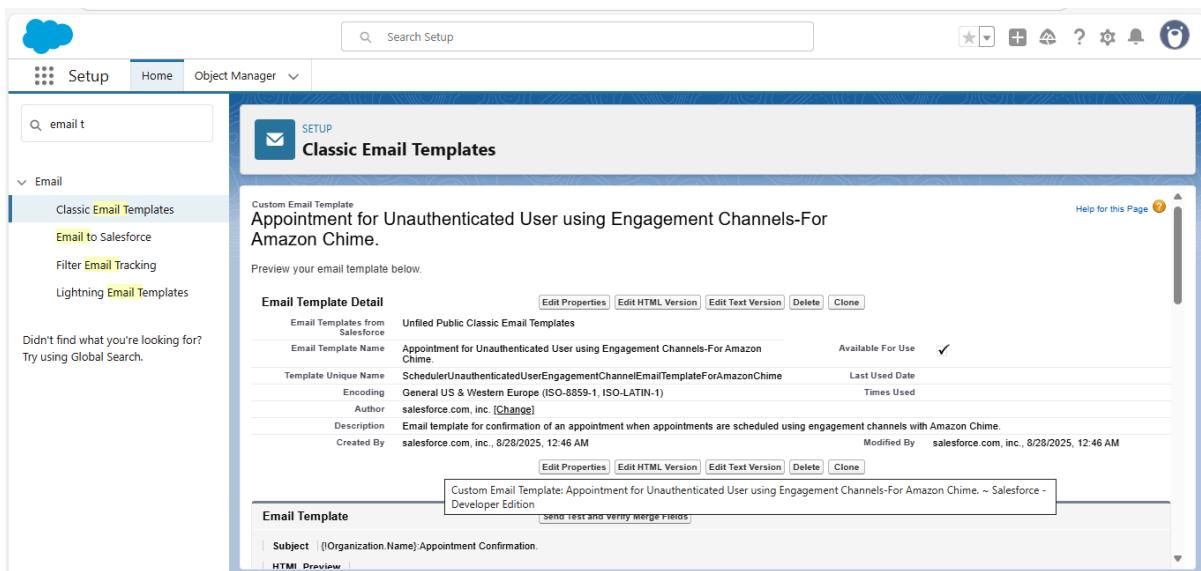
5. Subject : "Leave rejected"

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:

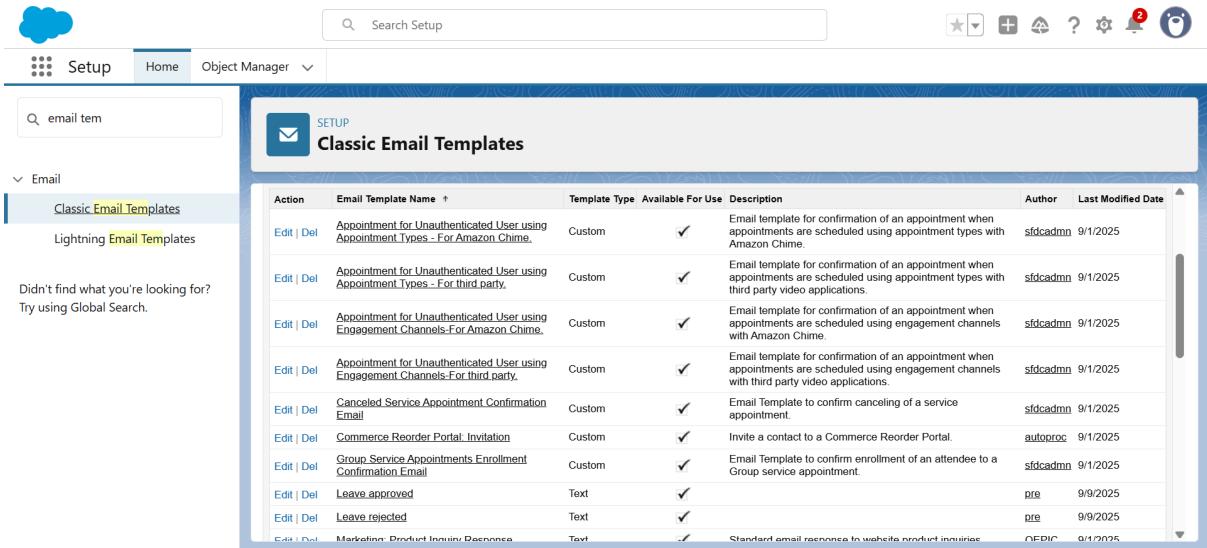
1. Go to setup in quick find box enter email template >> click on classic Email Template.
2. Click on >> New Email Template==>Choose text

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



The screenshot shows the Salesforce Setup interface with the 'Classic Email Templates' page selected. The left sidebar shows 'Email' categories: 'Classic Email Templates' (selected) and 'Lightning Email Templates'. A search bar at the top says 'email tem'. The main area displays a table of email templates:

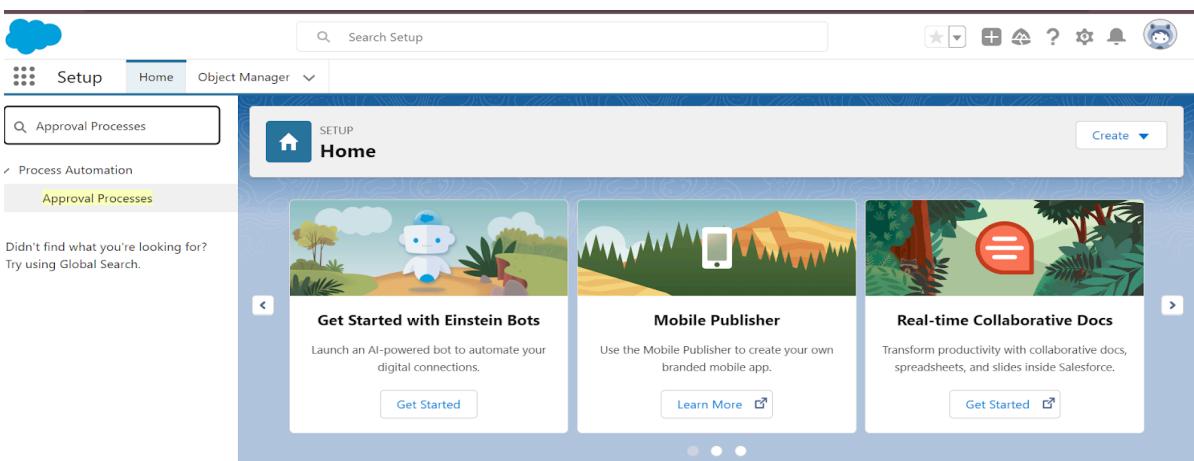
Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit Del	Appointment for Unauthenticated User using Appointment Types - For Amazon Chime.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with Amazon Chime.	sfdcadmin	9/1/2025
Edit Del	Appointment for Unauthenticated User using Appointment Types - For third-party.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with third party video applications.	sfdcadmin	9/1/2025
Edit Del	Appointment for Unauthenticated User using Engagement Channels-For Amazon Chime.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using engagement channels with Amazon Chime.	sfdcadmin	9/1/2025
Edit Del	Appointment for Unauthenticated User using Engagement Channels-For third-party.	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using engagement channels with third party video applications.	sfdcadmin	9/1/2025
Edit Del	Canceled Service Appointment Confirmation Email	Custom	✓	Email Template to confirm canceling of a service appointment.	sfdcadmin	9/1/2025
Edit Del	Commerce Reorder Portal_ Invitation	Custom	✓	Invite a contact to a Commerce Reorder Portal.	autoproc	9/1/2025
Edit Del	Group Service Appointments Enrollment Confirmation Email	Custom	✓	Email Template to confirm enrollment of an attendee to a Group service appointment.	sfdcadmin	9/1/2025
Edit Del	Leave approved	Text	✓		pre	9/9/2025
Edit Del	Leave rejected	Text	✓		pre	9/9/2025
Edit Del	Marketing Product Inquiry Response	Text	✓	Standard email response to website product inquiries.	OEPLIC	9/1/2025

Approval Process

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.



The screenshot shows the Salesforce Setup interface with the 'Home' page selected. The left sidebar shows 'Process Automation' categories: 'Approval Processes' (selected) and 'Quick Actions'. A search bar at the top says 'Approval Processes'. The main area displays a section titled 'Home' with three cards:

- Get Started with Einstein Bots**: Launch an AI-powered bot to automate your digital connections. [Get Started](#)
- Mobile Publisher**: Use the Mobile Publisher to create your own branded mobile app. [Learn More](#)
- Real-time Collaborative Docs**: Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce. [Get Started](#)

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.

The screenshot shows the Salesforce Setup interface for creating a new approval process. The process name is "check for vacant". In Step 2, the approval criteria is defined: "Tenant:status" is checked for "not equal to" "Leaving". A lookup window is open, showing the value "Leaving" selected. The main page also shows other criteria being defined.

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.

The screenshot shows Step 3 of 6 for specifying approver fields and record editability properties. It includes sections for selecting the approver field and defining record editability properties. The logic is set to AND.

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.

The screenshot shows the 'Approval Processes' setup page. At the top, there's a header with a gear icon labeled 'SETUP' and the page title 'Approval Processes'. Below the header, a section titled 'What Would You Like To Do Now?' contains a message about creating an approval process. It includes three radio button options:

- Yes, I'd like to create an approval step now.
- I'll do this later. Take me to the approval detail page to review what I've just created.
- I'll do this later. Take me back to the listing of all approval processes for this object.

A 'Go!' button is located at the bottom of this section. In the top right corner of the main content area, there's a 'Help for this Page' link with a question mark icon.

- 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.

The screenshot shows the 'Initial Submission Actions' configuration screen. At the top, there's a header with a gear icon labeled 'SETUP' and the page title 'Initial Submission Actions'. Below the header, there's a table with columns: Action, Type, Description, and Task. A red box highlights the 'Add New' button in the 'Task' column. Another red box highlights the 'Email Alert' entry in the 'Task' column of the first row, which corresponds to the 'Record Lock' action.

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. 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

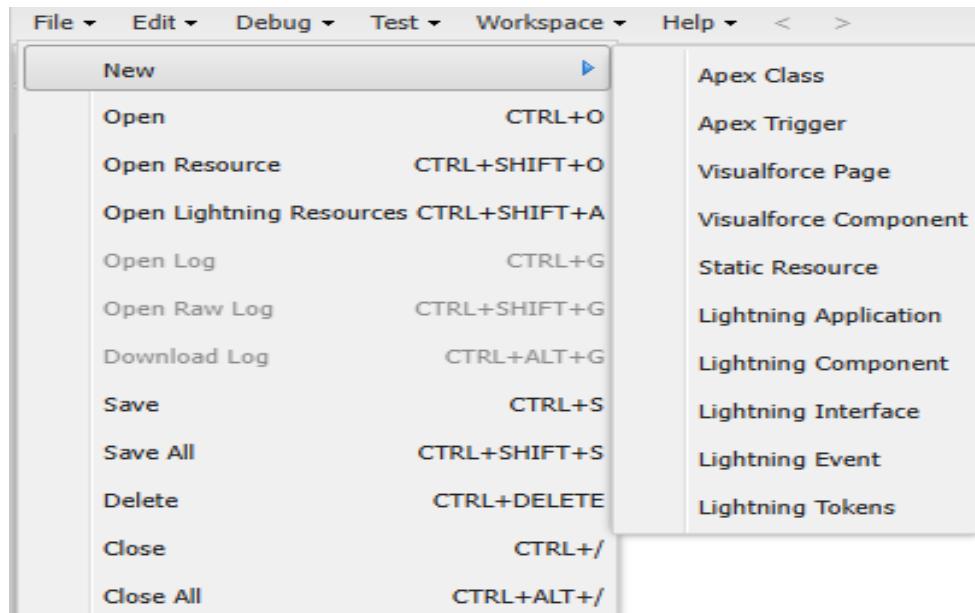
1. Under final rejection action click on add new and then select email alert.
2. Description: “your request for leave is rejected”.
3. unique name : auto populated
4. Email template : leave rejected
5. Recipient type : Email field
6. Available Recipients : Email field : Email
7. From Email address : Current user’s email
8. Click save

Apex Trigger

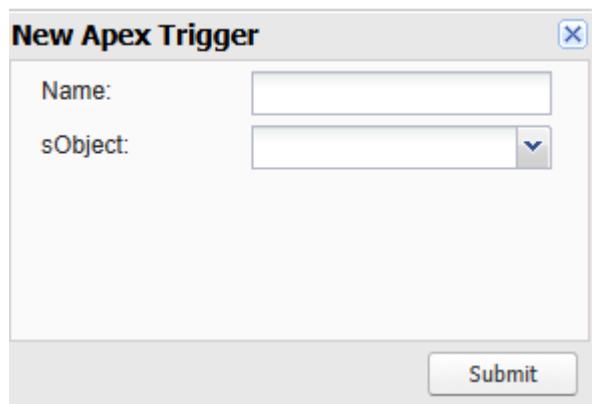
Create an Apex Trigger

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

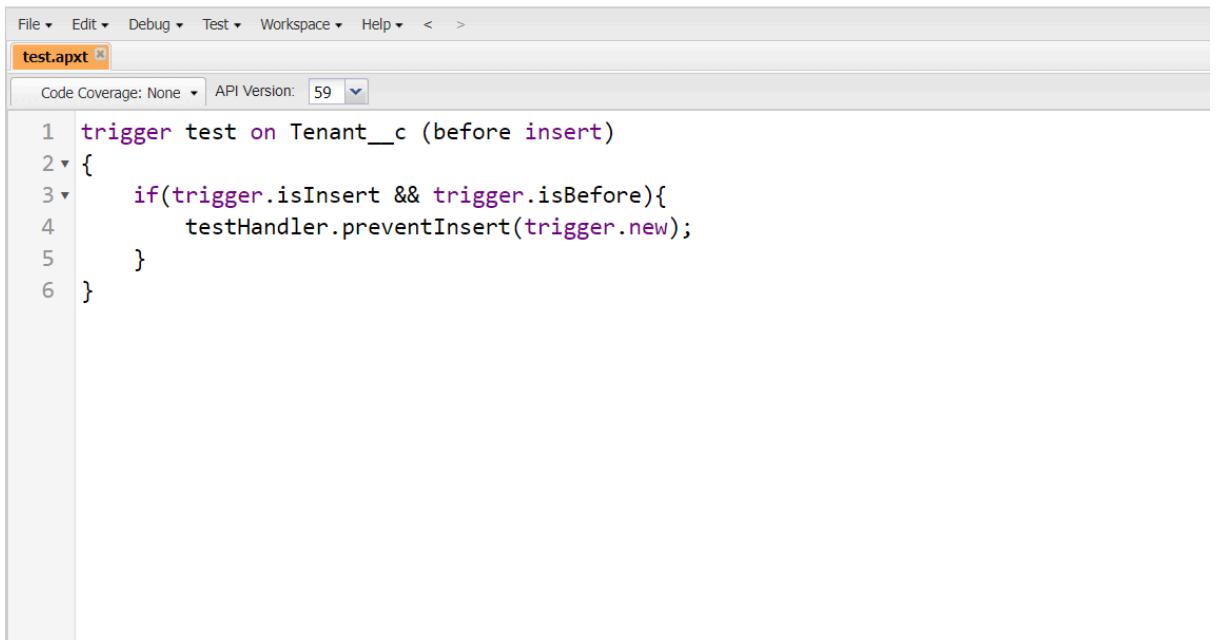
Click on the file >> New ? Apex Class.



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



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



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

Trigger Code:

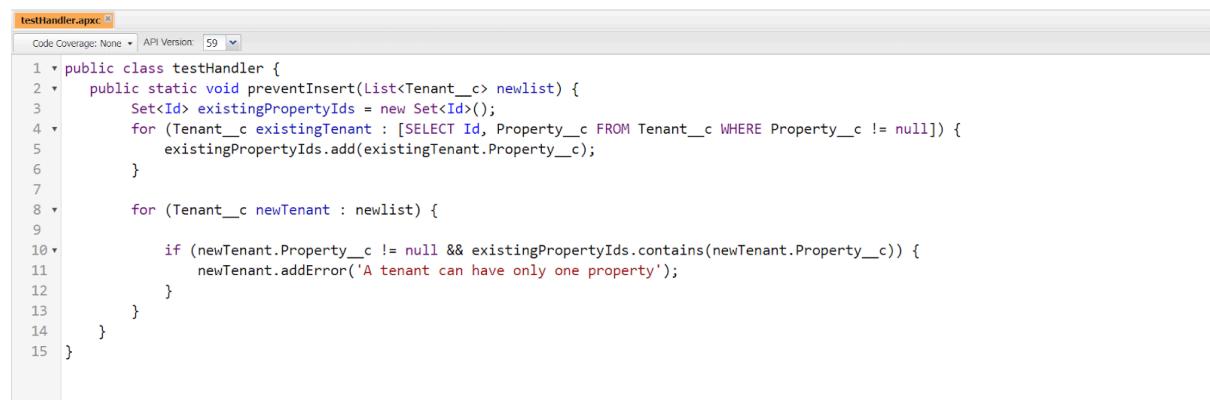
```
trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}
```

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.



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

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)) {  
                newTenantaddError('A tenant can have only one property');  
            }  
        }  
    }  
}
```

Testing the Trigger

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

New Tenant

* = Required Information

Information

* Tenant Name: luharmerp

Phone: (empty)

Email: (empty)

status: stay

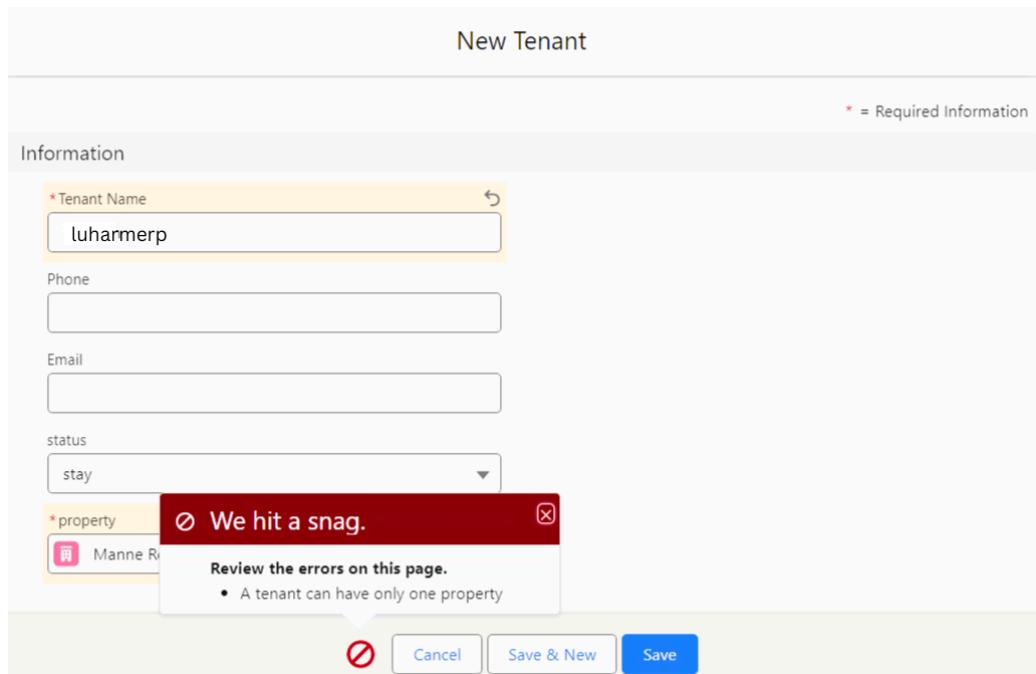
* property: Manne R

We hit a snag.

Review the errors on this page.

A tenant can have only one property

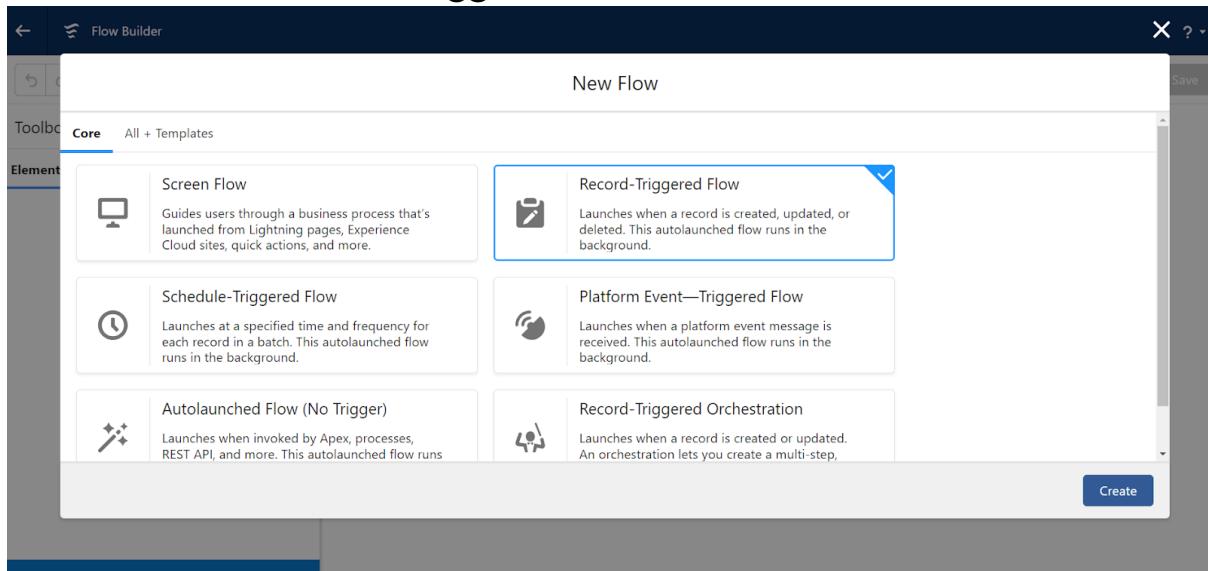
Cancel Save & New Save



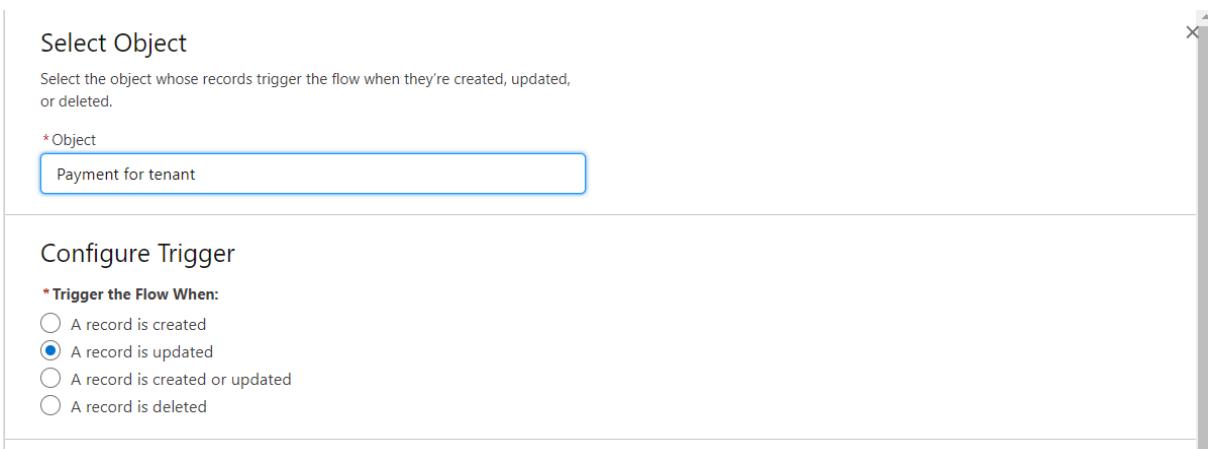
FLOWs

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.



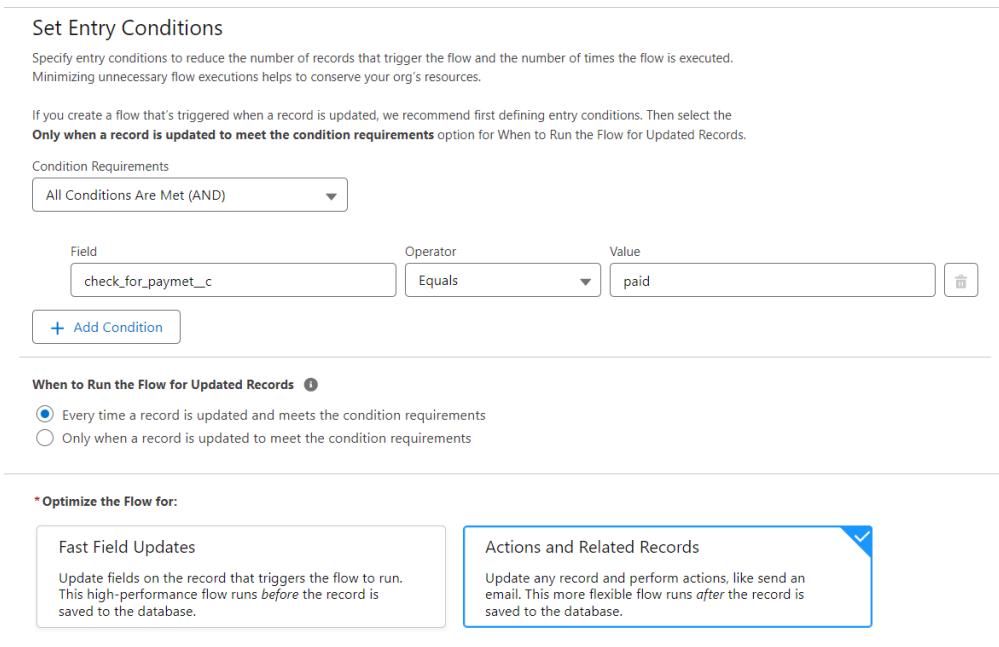
4. Set Entry Conditions

Under Condition Requirements

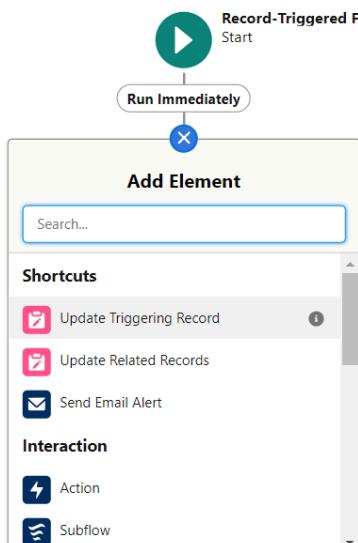
All Conditions are met

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



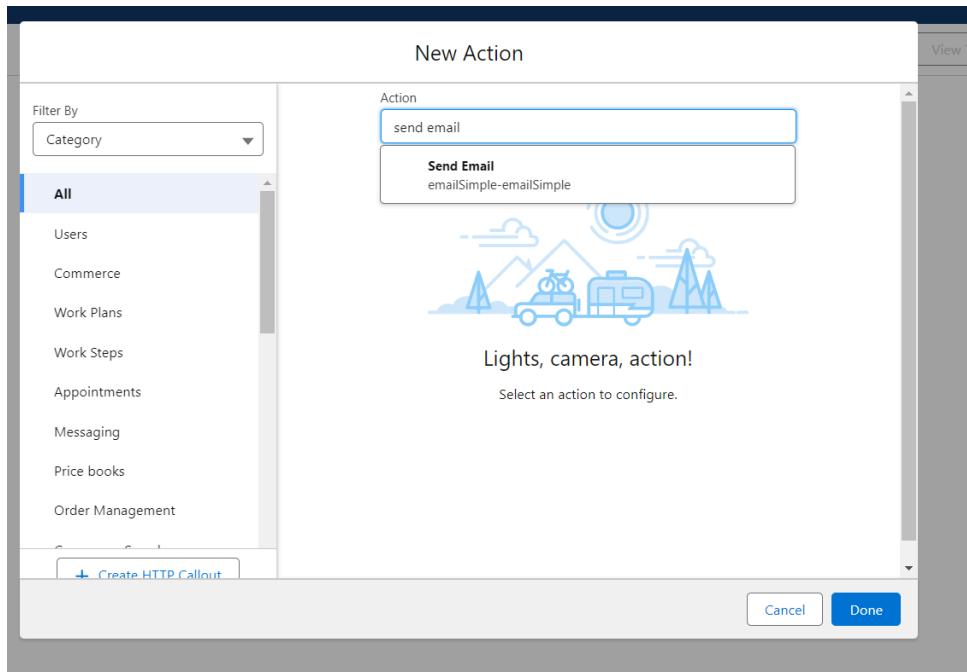
7. Under record trigger flow click on “+” icon and select action



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

8. Label : send email

API Name : send_email

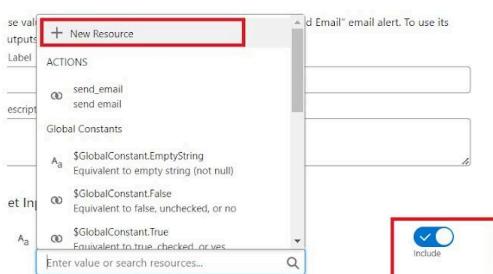


9. Label : send email

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

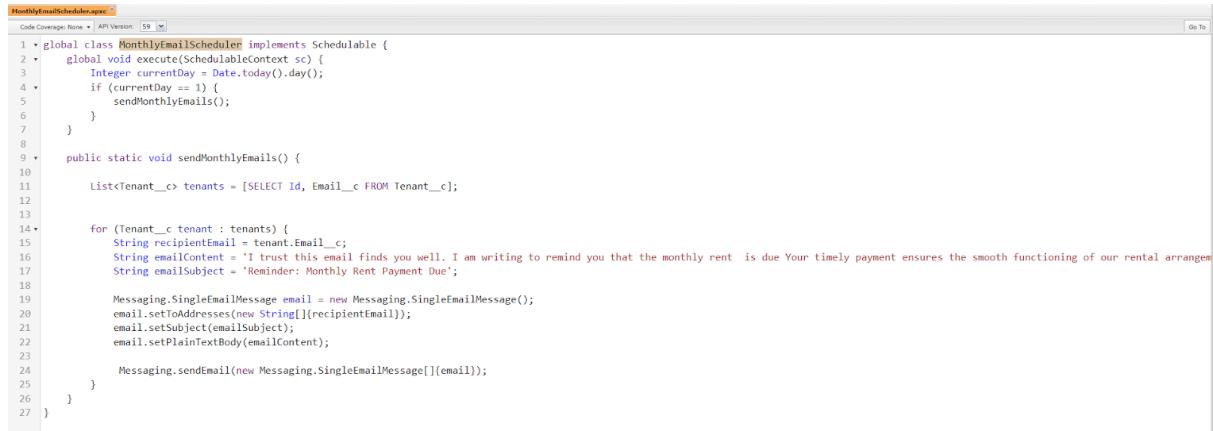
Flow API Name : monthly_payment

Click on activate

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.



```

1 • global class MonthlyEmailScheduler implements Schedulable {
2 •     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendMonthlyEmails();
6         }
7     }
8
9     public static void sendMonthlyEmails() {
10
11     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
12
13
14     for (Tenant__c tenant : tenants) {
15         String recipientEmail = tenant.Email__c;
16         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.';
17         String emailSubject = 'Reminder: Monthly Rent Payment Due';
18
19         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
20         email.setToAddresses(new String[]{recipientEmail});
21         email.setSubject(emailSubject);
22         email.setPlainTextBody(emailContent);
23
24         Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
25     }
26 }
27 }
```

Apex logic:

```

global class MonthlyEmailScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        Integer currentDay = Date.today().day();
        if (currentDay == 1) {
            sendMonthlyEmails();
        }
    }
    public static void sendMonthlyEmails() {
        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
        for (Tenant__c tenant : tenants) {
            String recipientEmail = tenant.Email__c;
            String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
            String emailSubject = 'Reminder: Monthly Rent Payment Due';

            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();

```

```

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

```

→ Save the code.

Schedule Apex class

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

The screenshot shows the Salesforce Setup Apex Classes page. The URL is [/apex/class](#). The page title is "Apex Classes". It displays a table of Apex classes with columns: Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. There are three entries:

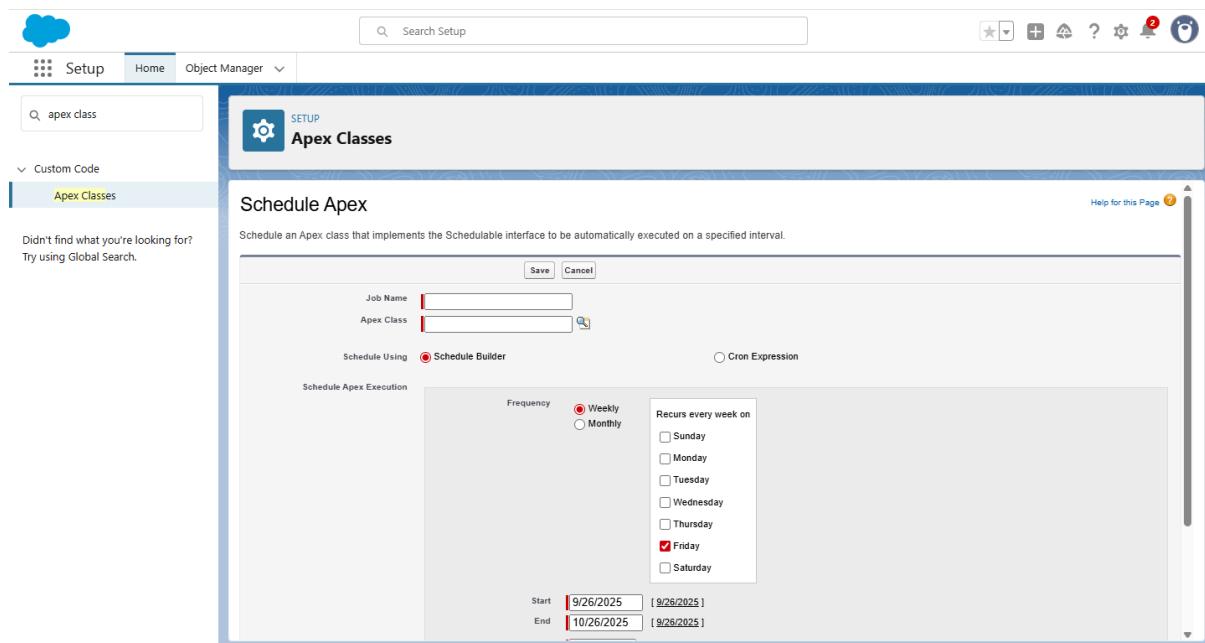
Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	MonthlyEmailScheduler		64.0	Active	1,125	Prem Rahul R. 9/10/2025, 2:12 AM	<input type="checkbox"/>
Edit Del Security	testHandler		64.0	Active	29	Prem Rahul R. 9/9/2025, 8:56 PM	<input type="checkbox"/>
Edit Del Security	testHandler1		64.0	Active	585	Prem Rahul R. 9/9/2025, 9:55 PM	<input type="checkbox"/>

Below the table, there is a section titled "Dynamic Apex Classes" with the sub-instruction: "Dynamic Apex extends your programming reach by interacting with Lightning Platform components." A table for "Dynamic Apex Classes" is shown with columns: Class Name, Namespace Prefix, Api Version, Created By, and Last Modified By. The message "No records to display." is displayed.

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

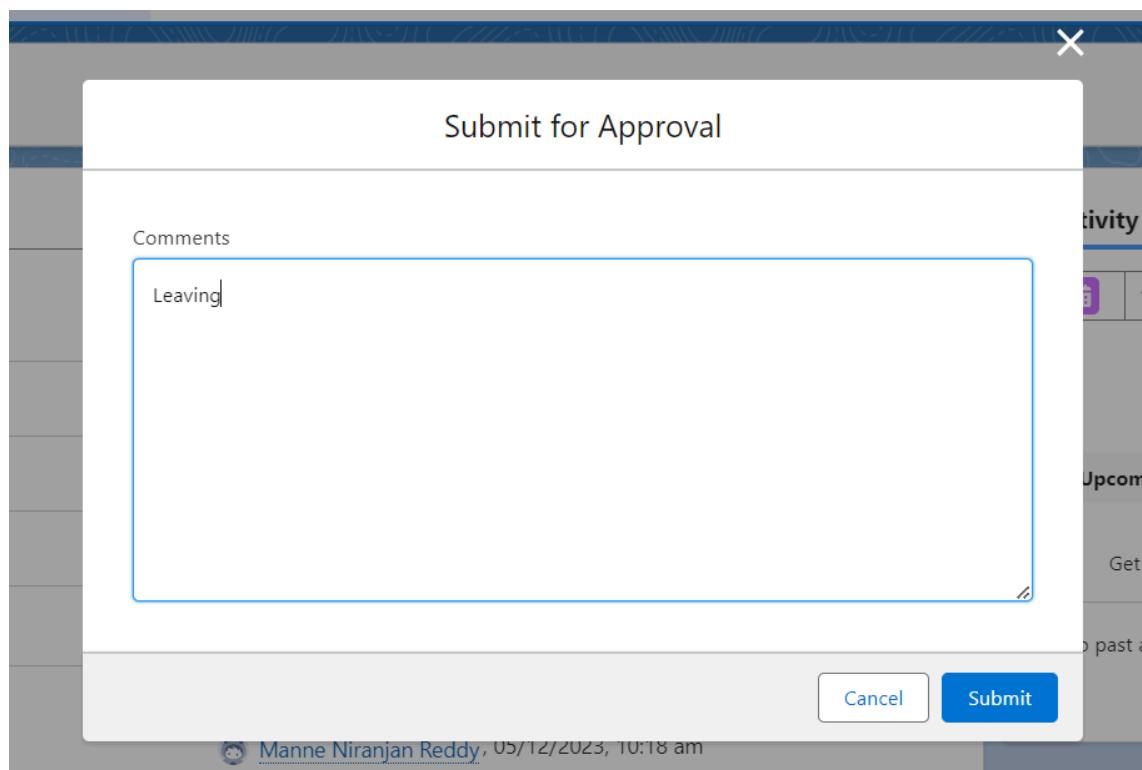
8. Preferred start time : 09:00 am

9. save



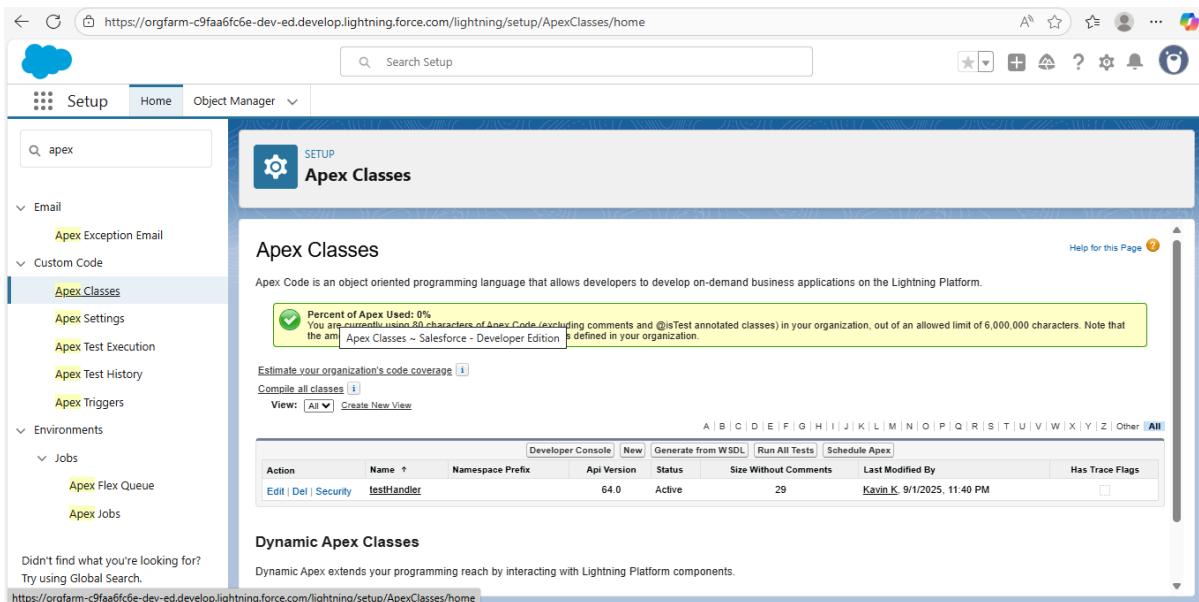
Testing the approval process

Enter any comment and 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 reject also you will get mail and notification.



The screenshot shows the Salesforce Setup Apex Classes page. The URL is <https://orgfarm-c9faa6fc6e-dev-ed.lightning.force.com/lightning/setup/ApexClasses/home>. The page title is "Apex Classes".

The sidebar on the left has sections for Email, Custom Code (with "Apex Classes" selected), and Environments. The main content area displays the "Apex Classes" section with the following details:

- Percent of Apex Used:** 0%
- Message:** You are currently using 80 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount of Apex Classes ~ Salesforce - Developer Edition is defined in your organization.
- Buttons:** Estimate your organization's code coverage, Compile all classes, View: All, Create New View.
- Table Headers:** Action, Name, Namespace Prefix, Api Version, Status, Size Without Comments, Last Modified By, Has Trace Flags.
- Table Data:** A single row for "testHandler" with the following values: Action (Edit | Del | Security), Name (testHandler), Namespace Prefix (None), Api Version (64.0), Status (Active), Size Without Comments (29), Last Modified By (Kavin K, 9/1/2025, 11:40 PM), Has Trace Flags (checkbox).