

Government Arts & Science College

Anthiyur – 638 501

Department of Computer Science

**Sales Force Developer with Agent
Blazer Champion
To Lease management**

Team Leader : Manju A

Team Members :

**Sudhakumari R
Swetha M
Mega M**

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
2. Allow search >> Save.

The screenshot shows the Salesforce Setup interface for creating a new object. The URL is https://orgfarm-c9faa6fc6e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001vM6T/Details/view. The page title is 'SETUP > OBJECT MANAGER property'. On the left, there's a sidebar with various tabs like Fields & Relationships, Page Layouts, Lightning Record Pages, etc. The main area has two sections: 'Details' and 'Description'. In the 'Details' section, fields include API Name (property_c), Custom (✓), Singular Label (property), and Plural Label (property). In the 'Description' section, checkboxes for 'Enable Reports' (✓), 'Track Activities' (✓), and 'Track Field History' (✓) are checked. Other settings shown are Deployment Status (Deployed) and Help Settings (Standard salesforce.com Help Window). At the bottom right 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
2. Allow search >> Save.

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various setup categories like Fields & Relationships, Page Layouts, Lightning Record Pages, and Buttons, Links, and Actions. The 'Buttons, Links, and Actions' category is currently selected. The main content area displays the 'Details' tab for the 'tenant' object. It includes fields for API Name (tenant_name__c), Singular Label (tenant), and Plural Label (tenants). On the right, there are checkboxes for enabling Reports, Activities, Field History, and Deployment Status, all of which are checked. Buttons for 'Edit' and 'Delete' are at the top right. The URL in the browser bar is https://orgfarm-c9faa6fc6e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01gL000001vMHI/Details/view.

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 tenant
 2. Plural label name>> Payments
 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.

The screenshot shows the Salesforce setup interface under 'Object Manager'. A sidebar on the left lists various object configuration options like Fields & Relationships, Page Layouts, and Buttons, Links, and Actions. The main 'Details' tab is selected, showing the object's API name as 'payment_for_tenant_c' and its singular label as 'payment for tenant'. On the right, there are sections for 'Enable Reports' (with 'Track Activities' checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings'.

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
2. Allow search >> Save.

The screenshot shows the Salesforce setup interface under 'Object Manager'. A sidebar on the left lists various object configuration options like Fields & Relationships, Page Layouts, and Buttons, Links, and Actions. The main 'Details' tab is selected, showing the object's API name as 'lease_c' and its singular label as 'lease'. On the right, there are sections for 'Enable Reports' (with 'Track Activities' checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings'.

The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected. The top navigation bar includes tabs for 'Setup', 'Home', and 'Deploy'. A red arrow points to the 'Home' tab. Below it, a search bar contains the text 'tab'. To the right of the search bar is a 'SETUP' icon with a gear and the word 'Tabs'. A red arrow points to this icon. On the left, a sidebar titled 'User Interface' has a 'Tabs' section with a 'Rename Tabs and Labels' link. Another red arrow points to this link. The main content area is titled 'Custom Tabs' and contains a table for 'Custom Object Tabs'. The table has columns for 'Action', 'Label', 'Tab Style', and 'Description'. A red arrow points to the 'New' button at the top right of the table. The table lists various tabs like 'Leads', 'Accounts', 'Opportunities', etc., each with a description indicating they are related to specific applications.

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 screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected. The left sidebar is expanded to show 'Feature Settings' (Analytics, Tableau) and 'User Interface' (Console Settings, Tabs). The 'Tabs' section under 'User Interface' is selected and highlighted in blue. The main content area shows the 'Custom Object Tabs' section. It includes a table for 'Custom Object Tabs' with columns for Action, Label, Tab Style, and Description. The table lists tabs for 'lease' (Bank), 'payment' (Books), 'property' (Airplane), and 'tenants' (Bell). Below this are sections for 'Web Tabs' and 'Visualforce Tabs', both of which currently have no tabs defined.

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.

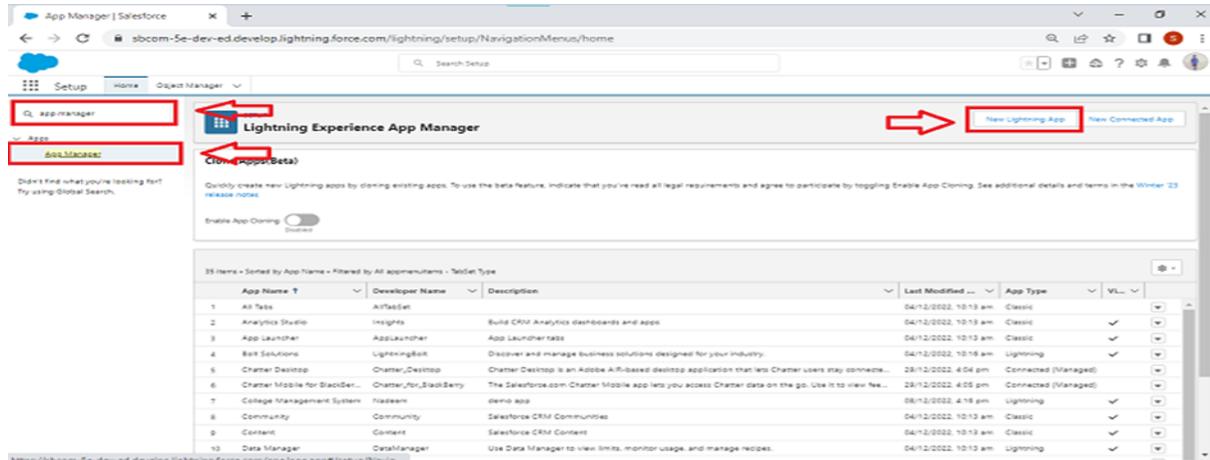
Use Case:

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.

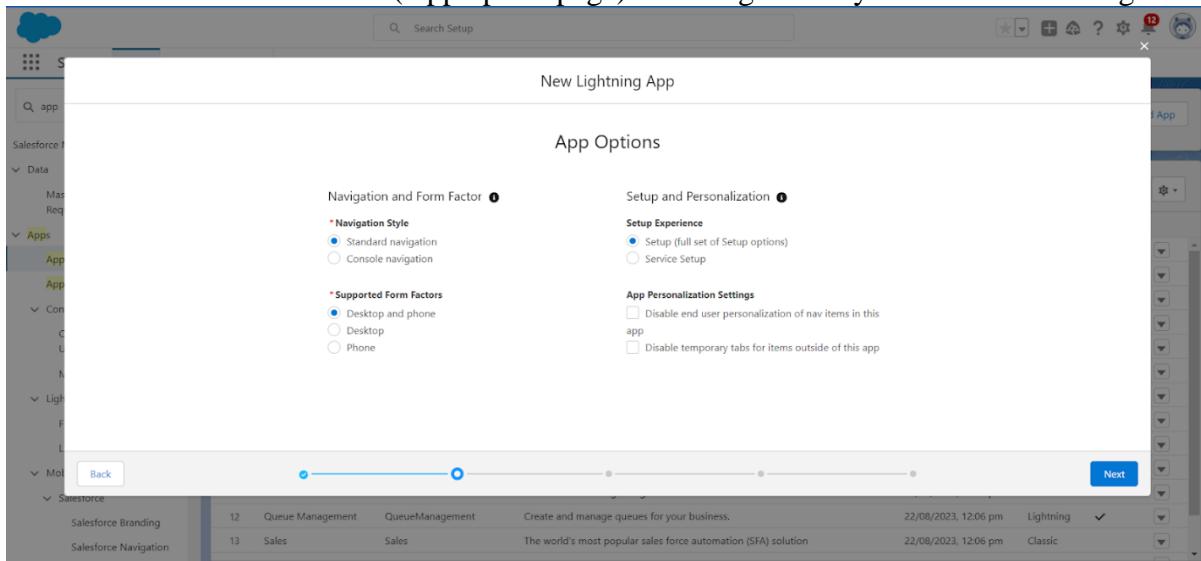


The screenshot shows the Salesforce App Manager interface. At the top, there are three search bars: 'Q app manager', 'Q Apps', and 'Q App Manager'. Below these, a red box highlights the 'Lightning Experience App Manager' section. A red arrow points to the 'New Lightning App' button on the right side of the screen. The main area displays a list of existing apps with columns for 'App Name', 'Developer Name', 'Description', 'Last Modified', 'App Type', and 'VLR'. One item in the list is 'demo app'.

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

App Name : Lease Management
Developer Name : This will auto populated
Image : optional (if you want to give any image you can otherwise not mandatory)
Primary colour hex value : keep this default.

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



The screenshot shows the 'App Options' page. Under 'Navigation and Form Factor', 'Navigation Style' is set to 'Standard navigation'. Under 'Supported Form Factors', 'Desktop and phone' is selected. In the 'Setup and Personalization' section, 'Setup Experience' is set to 'Setup (full set of Setup options)'. Under 'App Personalization Settings', there are two checkboxes: 'Disable end user personalization of nav items in this app' and 'Disable temporary tabs for items outside of this app'. At the bottom, there is a progress bar with steps 12 and 13 completed, and a 'Next' button highlighted.

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

5. To Add Navigation Items:

The screenshot shows the 'Object Manager' interface. On the left, under 'Available Items', there is a search bar with placeholder text 'Type to filter list...' and a list of objects including Accounts, Alert Settings, All Sites, Alternative Payment Methods, Analytics, App Launcher, Appointment Categories, Appointment Invitations, Approval Requests, and Asset Action Sources. On the right, under 'Selected Items', there is a list with three items: 'Payment for tenant' (with a hand icon), 'Tenants' (with an upward arrow icon), and 'property' (with a house icon). Between the two lists are two small arrows pointing right and left, indicating the ability to move items between them.

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 'User Profiles' configuration screen. At the top, it says 'New Lightning App'. Below that, it says 'User Profiles' and 'Choose the user profiles that can access this app.' Under 'Available Profiles', there is a search bar containing 'System administrator' with a red box around it and a red arrow pointing to the right. To the right of the search bar is a 'Selected Profiles' list containing 'System Administrator'. At the bottom, there is a horizontal progress bar with several blue dots and a red arrow pointing to the right, followed by a 'Save & Finish' button with a red box around it.

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.

The screenshot shows the Salesforce Object Manager interface. At the top, there are several browser tabs: 'Object Manager | Salesforce', 'Garage Management system - Google Doc', 'Lease Management - Google Doc', and 'ChatGPT'. Below the tabs, the address bar shows the URL 'thesmarbridge197-dev-ed.lightning.force.com/lightning/setup/ObjectManager/home'. The main content area has a blue header bar with 'SETUP' and 'Object Manager' buttons. The main table lists one item: 'property' (Label), 'property__c' (API Name), 'Custom Object' (Type), and '1 Items. Sorted by Label' (Description). The last modified date is '21/11/2023' and the status is 'Deployed'. A search bar at the top right contains the word 'property'.

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

The screenshot shows the 'Fields & Relationships' list for the 'property' object. On the left, there's a sidebar with various setup options like Page Layouts, Lightning Record Pages, etc. The main table has columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. There are eight items listed. A red arrow points to the 'New' button at the top right of the table, and another red arrow points to the 'Fields & Relationships' link in the sidebar.

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.
5. Now click on “Fields & Relationships” >>New
5. Select Data type as a “picklist” and Click on Next
5. 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

The screenshot shows the Salesforce Object Manager interface. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The 'Fields & Relationships' section is active. Under 'Text', there is a tooltip: 'Allows users to enter any combination of letters and numbers.'

4. Click on next

The screenshot shows the 'Step 2. Enter the details' configuration screen. It includes fields for Field Label (set to 'Name'), Length (set to 25), Field Name (set to 'Name'), Description, Help Text, Required (checkbox checked), Unique (checkbox checked), External ID (checkbox unchecked), and Auto add to custom report type (checkbox checked). A red arrow points to the 'Next' button 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.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
address	address_c	Text Area(255)		
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)	✓	
property Name	Name	Text(80)	✓	
sfqt	sfqt_c	Text(2)		
type	type_c	Picklist		

- 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.
9. Now click on “Fields & Relationships” >> New
9. Select Data type as a “Text” and Click on Next
9. 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.

5. Now click on “Fields & Relationships” >> New
5. Select Data type as a “picklist” and Click on Next
5. Fill the Above as following:
 - Field Label : Type
 - Field Name : gets auto generated
 - Enter values, with each value separated by a new line

The screenshot shows the Salesforce Object Manager interface. The left sidebar has a 'Fields & Relationships' section selected. The main area displays a table of fields for the 'tenant' object. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
email	email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
phone	phone__c	Phone		
property	property__c	Master-Detail(property)	✓	
status	status__c	Picklist		
tenant Name	Name	Text(80)	✓	

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.

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. The left sidebar has a 'Fields & Relationships' section selected. The main area displays a table of fields with columns for Field Label, Name, and Type. Fields listed include 'Created By' (CreatedById, Lookup(User)), 'End date' (End_date__c, Date), 'Last Modified By' (LastModifiedById, Lookup(User)), 'lease name' (lease_name__c, Text(80)), 'lease Name' (Name, Text(80)), 'Owner' (OwnerId, Lookup(User,Group)), 'payment date' (payment_date__c, Date), and 'property' (property__c, Lookup(property)).

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.

Setup > OBJECT MANAGER
payment for tenant

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
check for payment	check_for_payment_c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
payment Name	Name	Text(80)		✓
property	property_c	Master-Detail(property)		✓
tenant	tenant_c	Lookup(tenant)		✓

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.

Setup > OBJECT MANAGER
lease

New Custom Field

Step 1. Choose the field type

Data Type

None Selected

Auto Number

Formula

Roll-Up Summary

Lookup Relationship

Master-Detail Relationship

Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:
 • The relationship field requires all detail records.
 • The membership and value of a detail record are determined by the master record.
 • When a user deletes the master record, all detail records are deleted.
 • You can create rollup summary fields on the master record to summarize the detail records.
 The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.

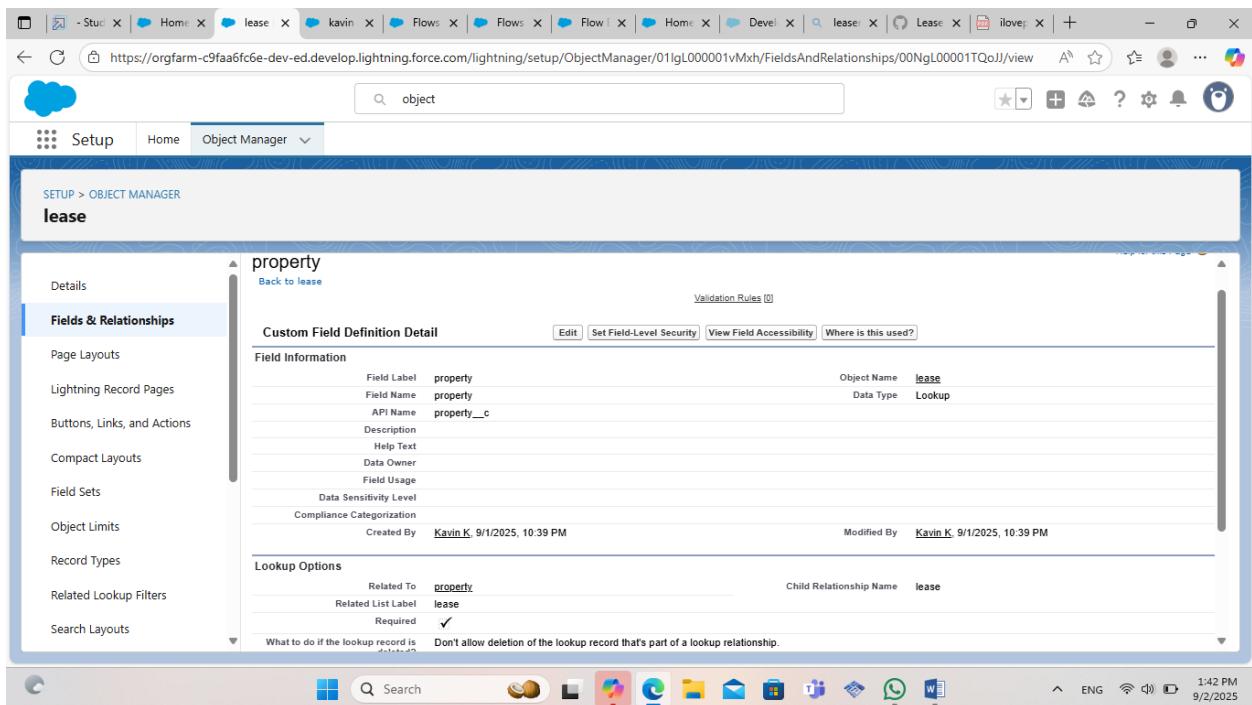
2. Now click on “Fields & Relationships” >> New
2. Select lookup relationship
2. Select the related object “ property” and click next.
2. Field Name : property
2. Field label : Auto generated
2. Next >> Next >> Save.

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.
8. Now click on “Fields & Relationships” >> New
8. Select lookup relationship
8. Select the related object “ Tenant” and click next.
8. Field Name : Tenant
8. Field label : Auto generated
8. 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.
15. Now click on “Fields & Relationships” >> New
15. Select masterdetail relationship
15. Select the related object “ property” and click next.
15. Field Name : property
15. Field label : Auto generated
15. 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.

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. On the left, a sidebar lists various configuration options like Buttons, Links, and Actions, Fields Sets, and Record Types. The 'Validation Rules' option is highlighted with a red box. The main area displays a table titled 'Validation Rules' with one column: 'RULE NAME'. A red box highlights the 'New' button at the top right of the table header. The status bar at the bottom indicates 'No items to display.'

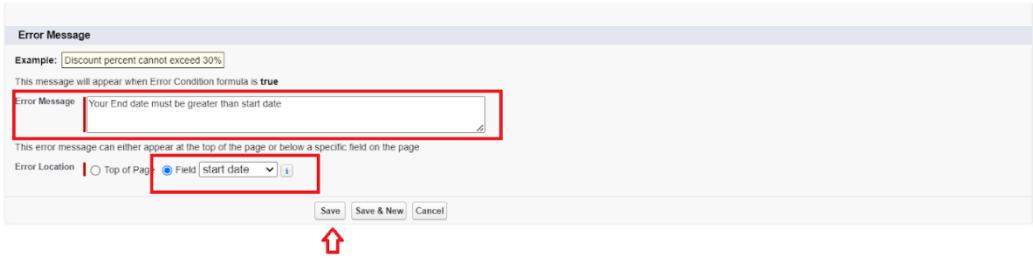
3. Enter the Rule name as “lease_end_date”.

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 'Rule Name' field contains 'lease_end_date' and has a red box around it. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains 'End_date__c > start_date__c' and has a red box around it. A dropdown menu for functions is open, showing options like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. The status bar at the bottom says 'Check Syntax | No errors found'.

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.



Rule Name	Error Location	Error Message	Active	Modified By
lease_end_date	start date	Your End date must be greater than start date	✓	Kavin K, 9/1/2025, 10:59 PM

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

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-c9faa6fc6e-dev-ed.lightning.force.com/lightning/setup/CommunicationTemplatesEmail/home>. The page title is "SETUP" and the sub-page title is "Classic Email Templates". A search bar at the top left contains "email t". On the left, there's a sidebar with "Email" categories: "Classic Email Templates" (which is selected), "Email to Salesforce", "Filter Email Tracking", and "Lightning Email Templates". Below the sidebar, a message says "Didn't find what you're looking for? Try using Global Search." The main content area displays a table of email templates. The columns are: "Edit | Del", "Email", "Custom", "Email template to confirm rescheduling of a service appointment", and "Created Date". The table lists various templates, such as "Sales: New Customer Email" (Custom, checked), "Scheduled Service Appointment Confirmation Email" (Custom, checked), "Scheduler Payments: Payment Reminder for Service Appointment Email" (Custom, checked), etc. The last row is a sample template: "SUPPORT: Self-Service New Comment Notification (SAMPLE)" (Text, checked).

Edit Del	Email	Custom	Email template to confirm rescheduling of a service appointment	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

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"

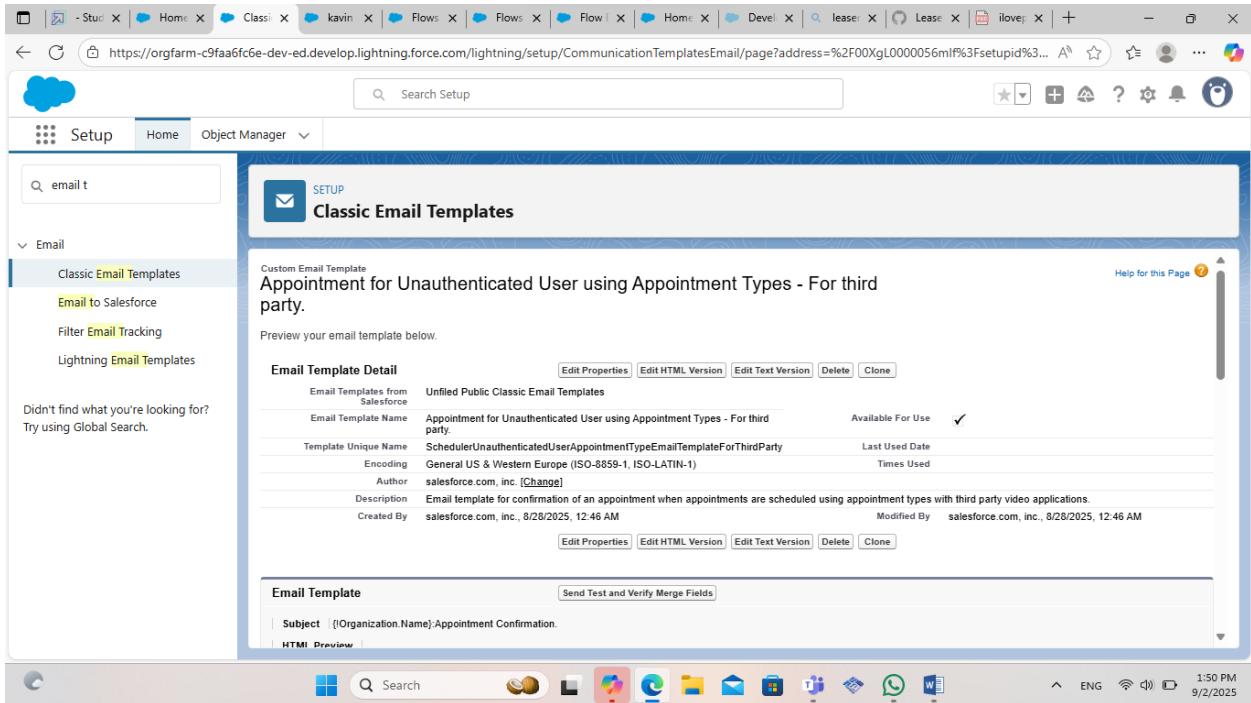
6. Email body :

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"

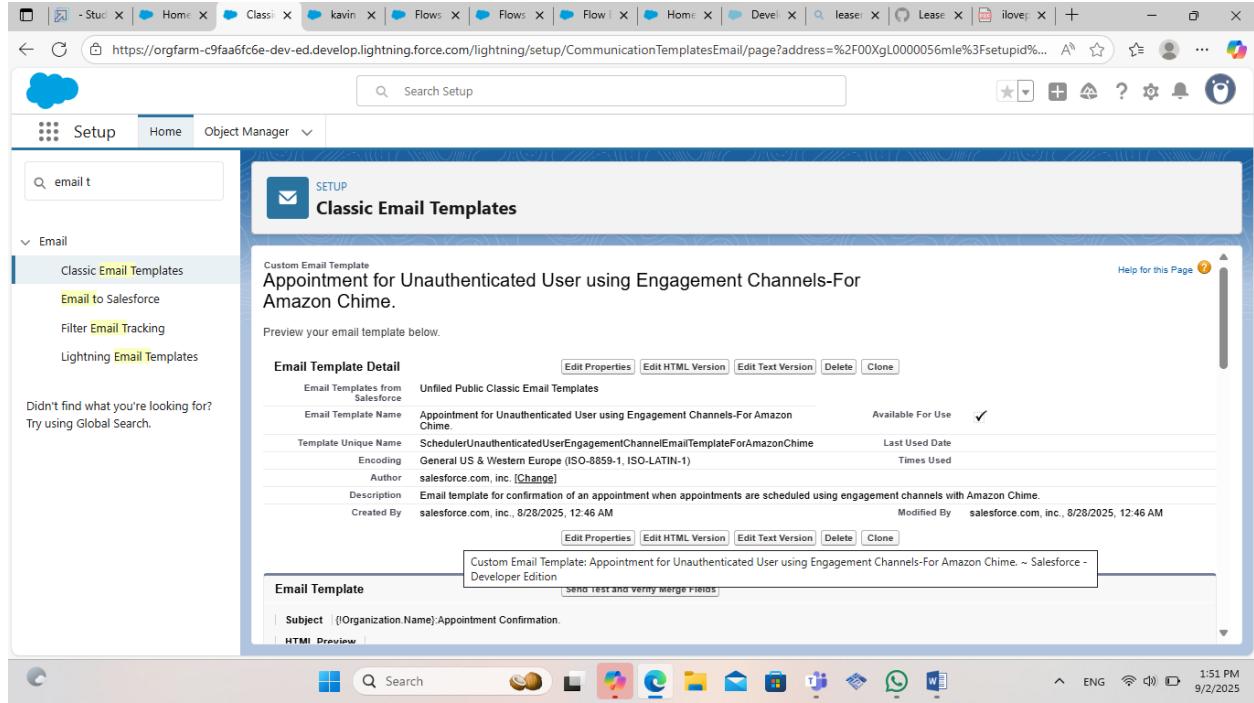
6. Email body :

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

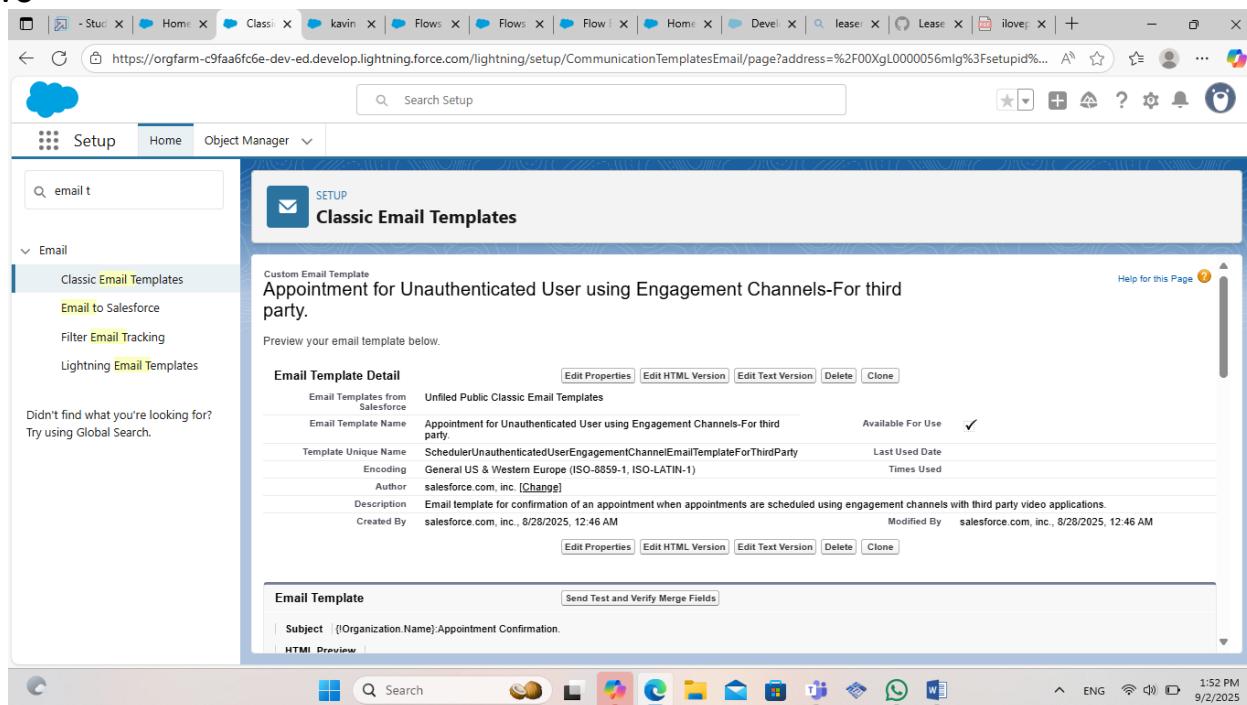
17. 3. Email Template Name is "Tenant Email"

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

The screenshot shows the Salesforce Setup interface with the 'CommunicationTemplatesEmail' section selected. The 'Classic Email Templates' tab is active. The page lists several email templates under the heading 'Unfiled Public Classic Email Templates'. Each template entry includes the name, template type (Custom), availability status (checked), description, author (sfdcadmin), and last modified date (8/28/2025). The descriptions provide details about the purpose of each template, such as confirming appointments or inviting contacts to a commerce reordering portal.

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit Del	Appointm... Unfiled Public Classic Email Templates ~ Salesforce - Developer Edition	Custom	✓	Email template for confirmation of an appointment when appointments are scheduled using appointment types with Amazon Chime.	sfdcadmin	8/28/2025
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	8/28/2025
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	8/28/2025
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	8/28/2025
Edit Del	Canceled Service Appointment Confirmation Email	Custom	✓	Email Template to confirm canceling of a service appointment.	sfdcadmin	8/28/2025
Edit Del	Commerce Reorder Portal- Invitation	Custom	✓	Invite a contact to a Commerce Reorder Portal.	autoproc	8/28/2025

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.

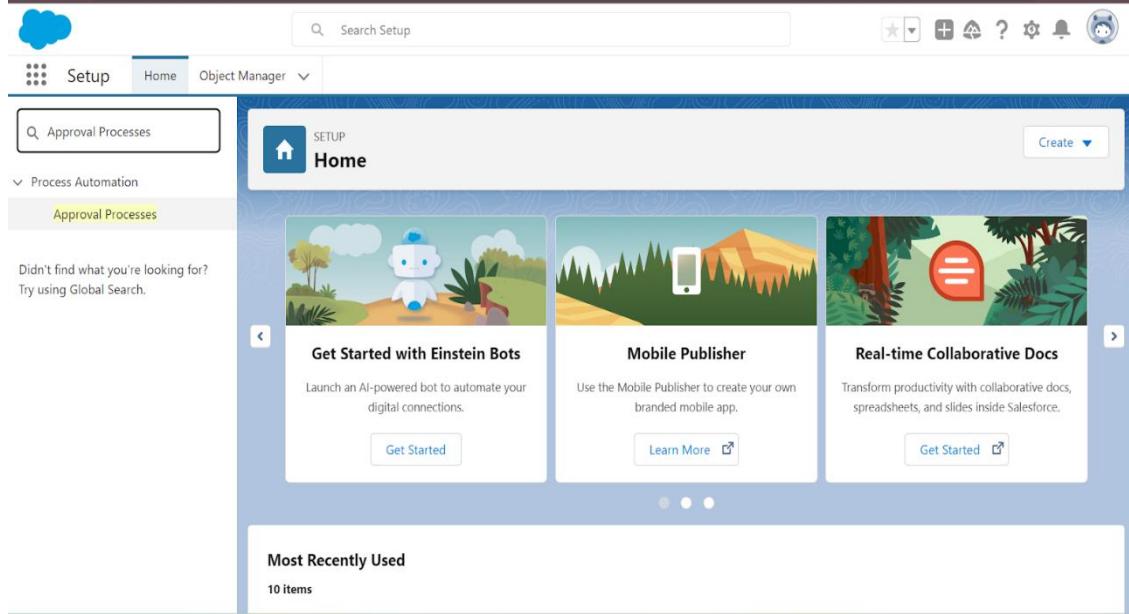
An approval process consists of finalizing the basic properties of the approval process (as shown in the below image), approval steps, and actions to be executed.

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.

Step 2 of 6

Specify Entry Criteria

Field	Operator	Value	Logic
Tenant status	not equal to	Leaving	AND
-None-	-None-		

Previous Save Next Cancel

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.

Step 3 of 6

Select Field Used for Automated Approval Routing

Next Automated Approver Determined: -None-

Use Approver Field of property Owner:

Record Editability Properties

Administrators ONLY can edit records during the approval process.
 Administrators OR the currently assigned approver can edit records during the approval process.

Previous Save Next Cancel

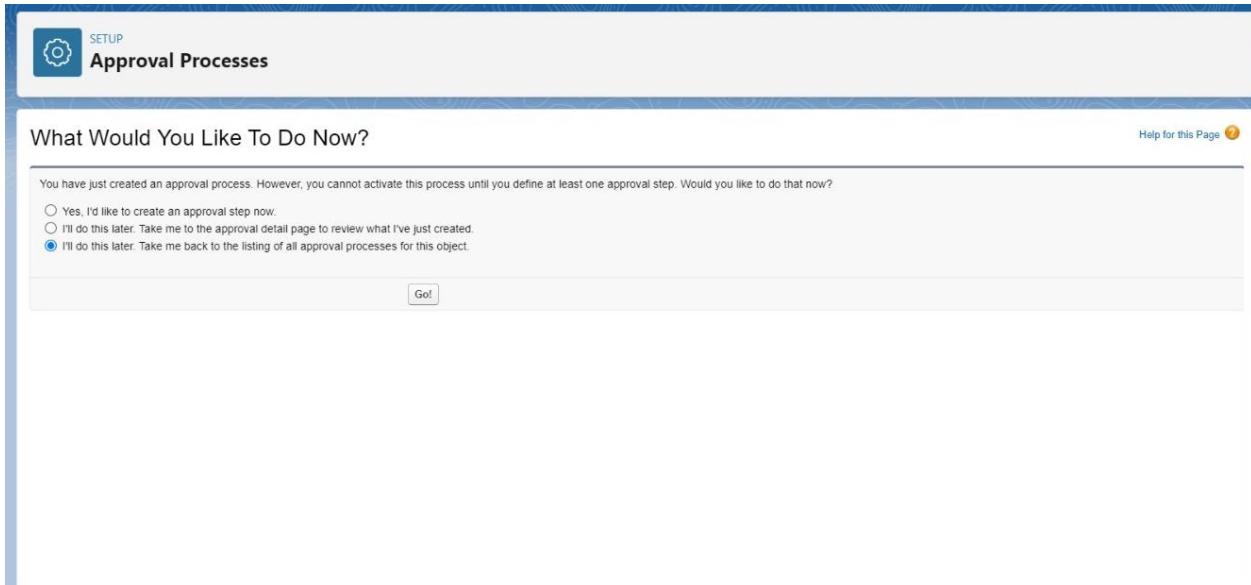
9. Click on next leave the email template click on next

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 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

4. Apex Trigger

3. Use case:

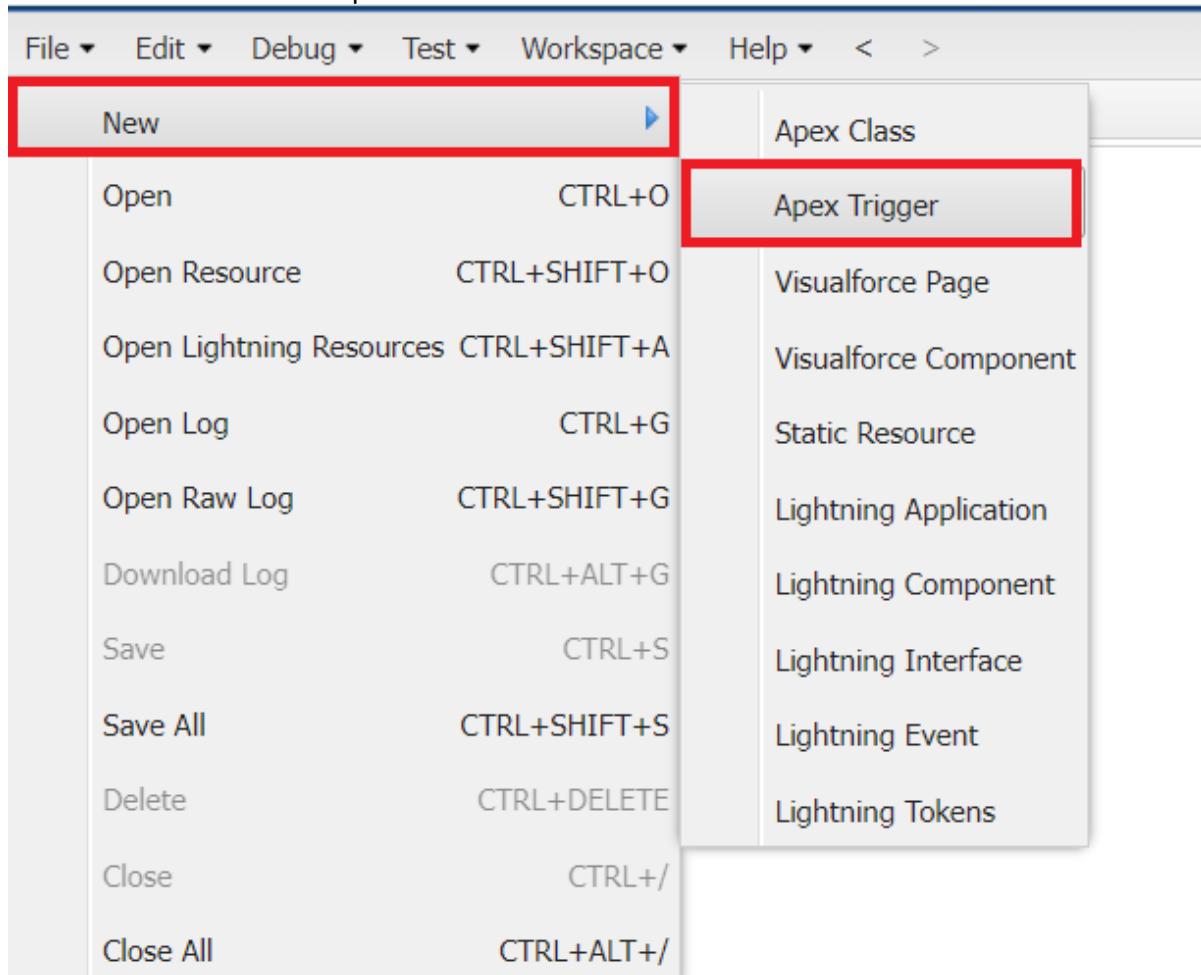
The tenant and property are in a master-detail relationship, wherein each tenant is associated with only one property. When a tenant attempts to create a new record with an existing property, an error should be displayed, indicating that a tenant can have only one property.

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.



```
testHandler.apxc
Code Coverage: None API Version: 59
1 public class testHandler {
2     public static void preventInsert(List<Tenant__c> newList) {
3         Set<Id> existingPropertyIds = new Set<Id>();
4         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
5             existingPropertyIds.add(existingTenant.Property__c);
6         }
7         for (Tenant__c newTenant : newList) {
8             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
9                 newTenant.addError('A tenant can have only one property');
10            }
11        }
12    }
13 }
14 }
15 }
```

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

The screenshot shows a Salesforce form titled "New Tenant". The "Information" section contains fields for "Tenant Name" (niranjan), "Phone", "Email", and "status" (stay). A dropdown menu for "property" is open, showing "Manne R". An error message box titled "We hit a snag." appears, stating "Review the errors on this page." with the note "• A tenant can have only one property". At the bottom are "Cancel", "Save & New", and "Save" buttons.

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.

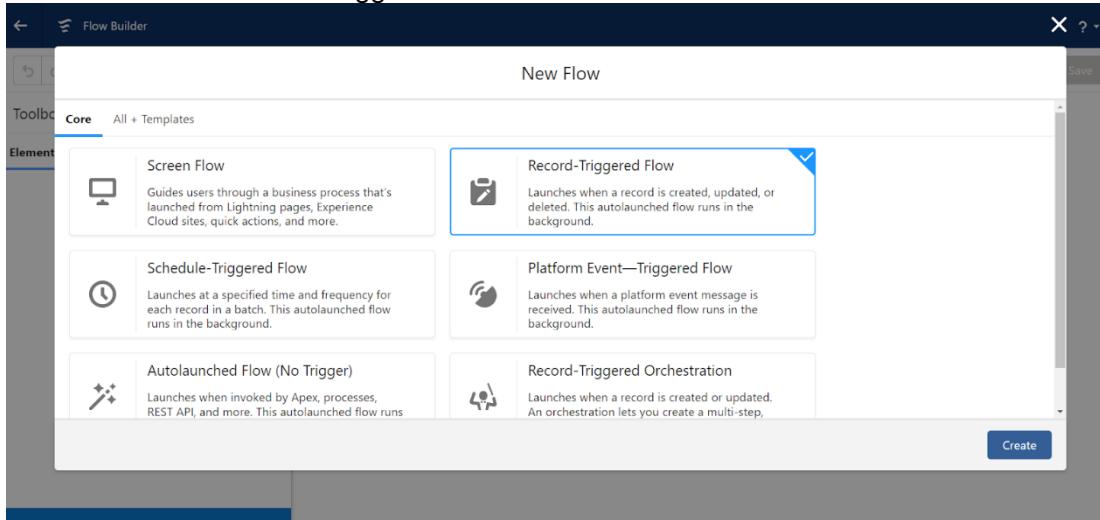
When and why should we use a flow

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.

The screenshot shows the 'Select Object' step. It has a 'Select Object' heading and a note: 'Select the object whose records trigger the flow when they're created, updated, or deleted.' Below is a field labeled '* Object' containing 'Payment for tenant'. Below this, the 'Configure Trigger' section has a heading '* Trigger the Flow When:' followed by four radio button options: 'A record is created', 'A record is updated' (which is selected and highlighted with a blue border), 'A record is created or updated', and 'A record is deleted'.

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

Set Entry Conditions

X

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

All Conditions Are Met (AND) ▾

Field

check_for_payment_c

Operator

Equals ▾

Value

paid



+ Add Condition

When to Run the Flow for Updated Records ⓘ

- Every time a record is updated and meets the condition requirements
 Only when a record is updated to meet the condition requirements

* Optimize the Flow for:

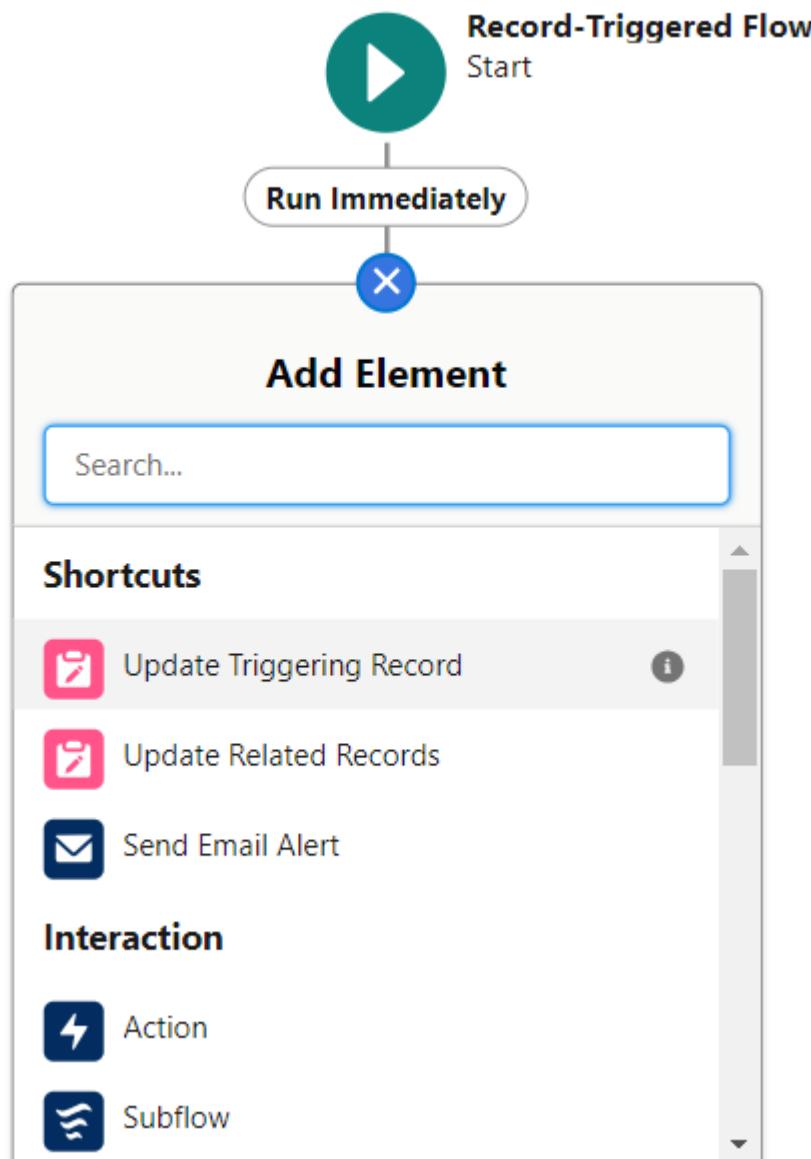
Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs *before* the record is saved to the database.

Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.

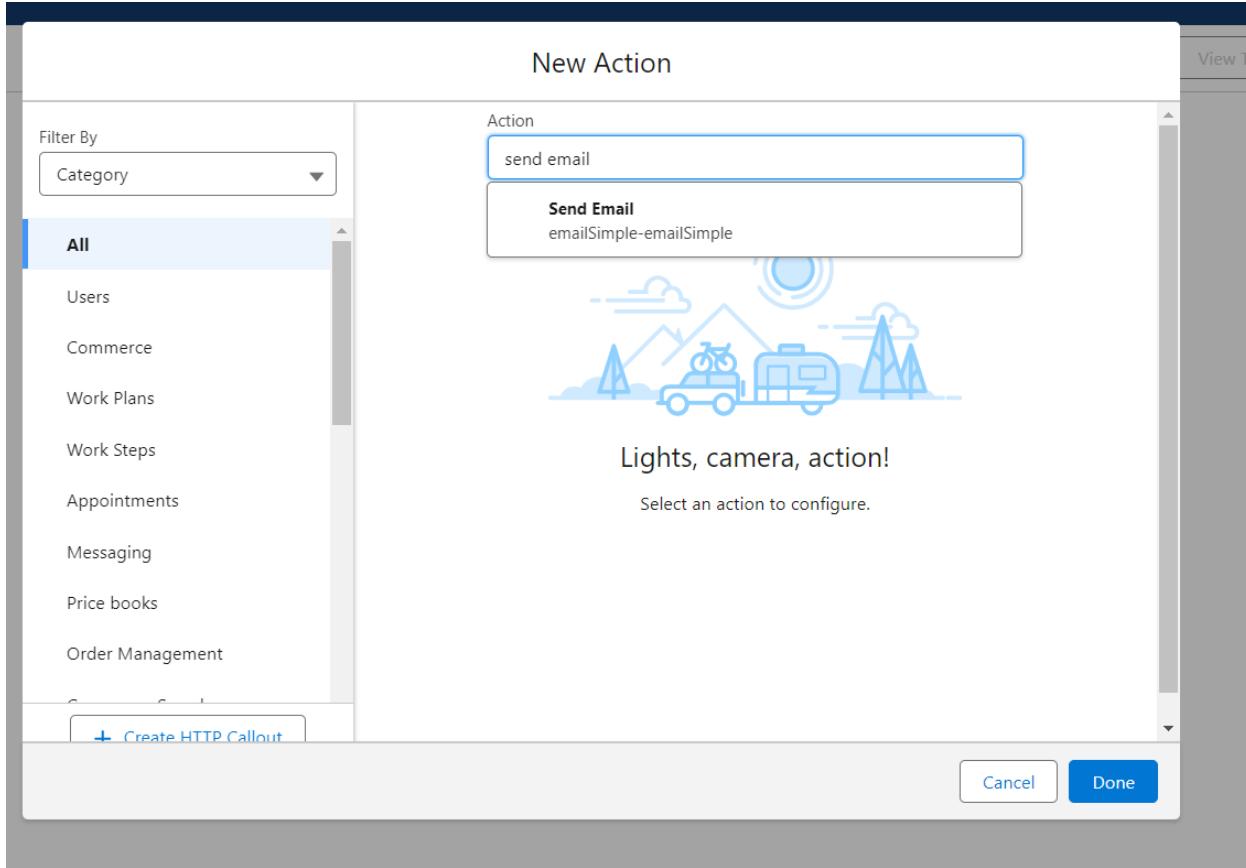
7. Under record trigger flow click on “+” icon and select action



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

This screenshot shows a search interface for selecting a resource. A red box highlights the '+ New Resource' button at the top left. The search bar contains the text 'send_email'. The results section is titled 'ACTIONS' and lists 'send_email' twice. Below the results are 'Global Constants' and 'Actions'. A red box highlights the 'Include' toggle switch at the bottom right of the results area. The placeholder text 'Enter value or search resources...' is visible at the bottom.

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

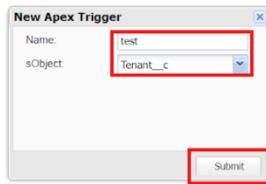
Flow label : monthly payment

Flow API Name : monthly_payment

Click on activate

The screenshot shows the Salesforce Setup interface with the 'Flows' tab selected. The main area displays a table of flow definitions. The columns include Flow Label, Process Type, Active, Triggered, Package State, and Last Run. The table lists various flows such as 'Add or Modify Service Appointment Attendees', 'Approvals Workflow: Evaluate Approval Requests', and 'Chats Routed to Agents and Queues'. The 'Process Type' column indicates the flow type (e.g., Salesforce Scheduler Flow, Screen Flow, Identity User Registration Flow). The 'Package State' column shows that most flows are 'Managed-installed'. The 'Last Run' column shows the last execution time for each flow.

Flow Label	Process Type	Active	Triggered	Package State
Add or Modify Service Appointment Attendees	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-installed
Approvals Workflow: Evaluate Approval Requests	Screen Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed
Approvals Workflow: Process Approval Submission	Screen Flow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Managed-installed
Authentication Provider User Registration	Identity User Registration Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed
Basic Approval Request	Flow Orchestration for CMS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed
Book Appointment from Invitation	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed
Cancel Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed
Change Case Owner to Incident Owner	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed
Chats Routed to Agents and Queues	Omni-Channel Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-installed



3. Click Submit.
3. 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);
    }
}
```

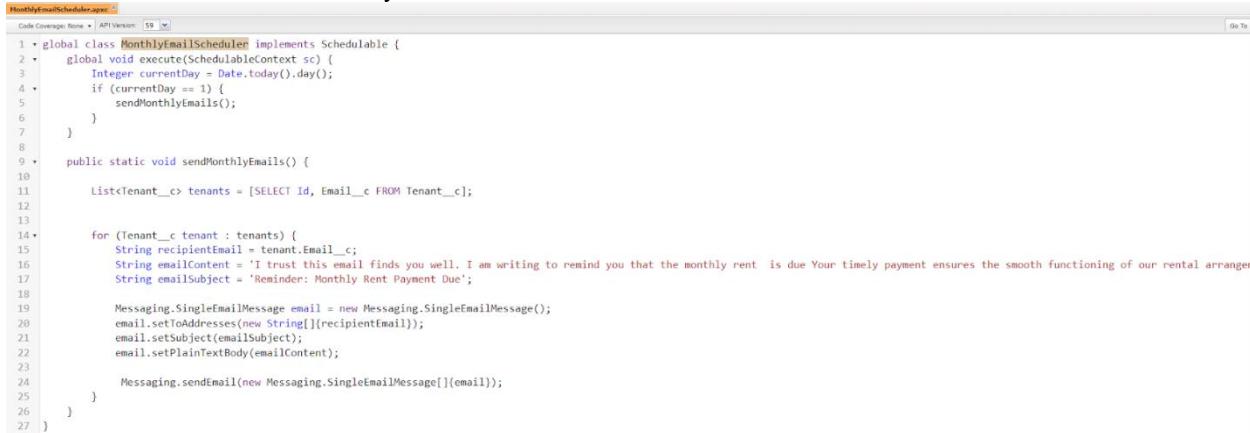
4. Schedule class :

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.



```
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.

5. Schedule Apex class

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

Apex Classes

Didnt find what you're looking for? Try using Global Search.

Percent of Apex Used: 0.07% You are currently using 4,618 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage. [View](#)

View: All [Create New View](#)

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	ContactCreator		59.0	Active	618	Manne Nirjan Reddy, 29/11/2023, 3:02 pm	<input type="checkbox"/>
Edit Del Security	createaccount		59.0	Active	447	Manne Nirjan Reddy, 29/11/2023, 1:17 pm	<input type="checkbox"/>
Edit Del Security	MonthlyEmailScheduler		59.0	Active	1,125	Manne Nirjan Reddy, 02/12/2023, 9:53 am	<input type="checkbox"/>
Edit Del Security	testHandler		59.0	Active	584	Manne Nirjan Reddy, 27/11/2023, 11:20 am	<input type="checkbox"/>

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

Schedule Apex

Schedule an Apex class that implements the 'Schedulable' interface to be automatically executed on a weekly or monthly interval.

Save Cancel

Job Name

Apex Class

Schedule Apex Execution

Frequency Weekly Monthly On day of every month On Sunday

Start [04/12/2023]

End [04/12/2023]

Preferred Start Time

Exact start time will depend on job queue activity.

Save Cancel

Testing the approval process

Tenant
Niranjan

New Contact New Case New Lead

Related Details

Tenant Name
Niranjan

Phone

Email
niranjareddymanne2507@gmail.com

status
stay

property
Manne Residency

Created By
Manne Niranjan Reddy, 29/11/2023, 10:07 am

Last Modified By
Manne Niranjan Reddy, 05/12/2023, 10:18 am

Activity

New Opportunity

Submit for Approval

Delete

Printable View

Edit

Clone

Filters: All time • All activities

Upcoming & Overdue

No activities to show.

Get started by sending an email, scheduling a task, and more.

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

Enter any comment and click on submit

X

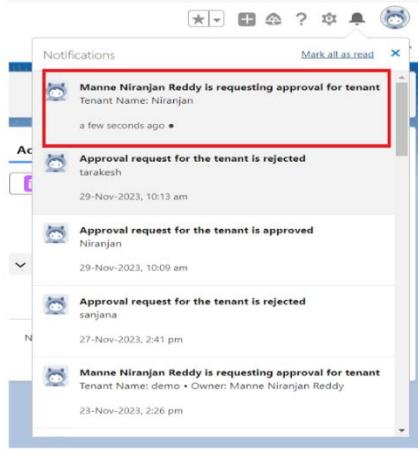
Submit for Approval

Comments

Leaving

Cancel Submit

Manne Niranjan Reddy, 05/12/2023, 10:18 am



Click on that notification

Approval Request
Tenant Approval Pending

Submitter: Manne Nirajan Reddy Date Submitted: 05-Dec-2023 Actual Approver: Manne Nirajan Reddy Assigned To: Manne Nirajan Reddy

Details

Approval Details
Tenant Name: Niranjan

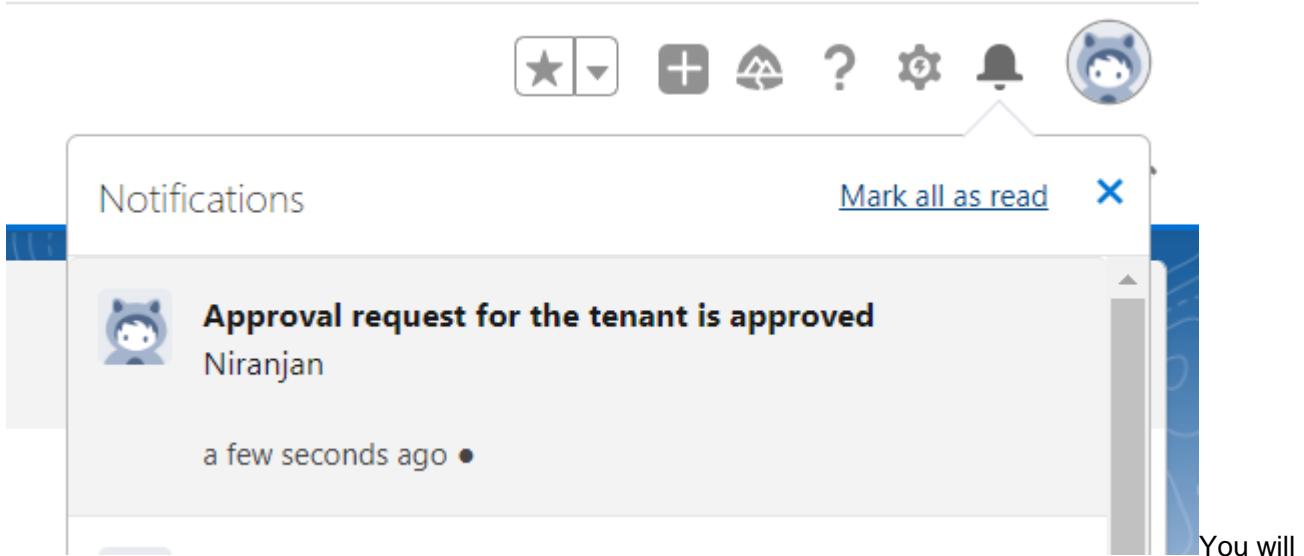
Submitter Comments

Manne Nirajan Reddy
Leaving
05-Dec-2023, 10:22:01 am

click

on approve

Give any comment and submit



find notification like this and you will get an email check
Note: similarly do for reject also you will get mail and notification

A screenshot of the Salesforce Setup Apex Classes page. The URL in the browser is https://orgfarm-c9faa6fc6e-dev-ed.develop.lightning.force.com/lightning/setup/ApexClasses/home. The page title is "Apex Classes". The left sidebar shows navigation categories: Email, Custom Code (with "Apex Classes" selected), Environments, and Jobs. A search bar at the top left contains the text "apex". The main content area includes a status message: "Percent of Apex Used: 0% You are currently using 0 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 limit applies to Apex Classes ~ Salesforce - Developer Edition is defined in your organization." Below this is a section for "Estimate your organization's code coverage" with a link to "Compile all classes". A table lists one Apex class: "testHandler" (Name, Namespace Prefix: testHandler, Api Version: 64.0, Status: Active, Size Without Comments: 29, Last Modified By: Kavin K, Last Modified Date: 9/1/2025, 11:40 PM). At the bottom, there is a section titled "Dynamic Apex Classes" with the subtext "Dynamic Apex extends your programming reach by interacting with Lightning Platform components." The bottom of the screen shows the Windows taskbar with various pinned icons.