

Government Arts & Science College Anthiyur-638311

Department of Computer Science

Salseforce Agent Blazer Champion

To Lease management

Team Leader : Kavin K

Team Meambers :

Ramana R
Sasi S
Saravanakumar S

Object

To create an object:

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

The screenshot shows the Salesforce Setup interface for creating a new object. The URL in the browser is <https://orgfarm-c9faa6fc6e-dev.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001vM6T/Details/view>. The page title is "SETUP > OBJECT MANAGER". The object being created is named "property".
The left sidebar lists various object settings: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, and Search Layouts.
The main "Details" section contains the following fields:

- Description: (empty)
- API Name: property_c
- Custom: ✓
- Singular Label: property
- Plural Label: property
- Enable Reports: ✓
- Track Activities: ✓
- Track Field History: ✓
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

At the bottom right of the main area are "Edit" and "Delete" buttons. The system status bar at the bottom shows the date and time as 12:33 PM 9/2/2025.

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.

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 shows the API name as 'tenant_name__c' and the singular label as 'tenant'. On the right, there are checkboxes for enabling reports, tracking activities, and tracking field history, all of which are checked. Deployment status is listed as 'Deployed'.

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>> 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.

SETUP > OBJECT MANAGER
payment for tenant

Details

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

Edit | Delete

2.

Create Lease Object

To create an object:

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

SETUP > OBJECT MANAGER
lease

Details

Description	
API Name	lease_c
Custom	✓
Singular Label	lease
Plural Label	lease

Enable Reports

- ✓ Track Activities
- ✓ Track Field History
- ✓ Deployment Status
- Deployed
- Help Settings
- Standard salesforce.com Help Window

Edit | Delete

Snip copied to clipboard
Select here to mark up and share the image

3. Tabs

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

Types of Tabs:

1. Custom Tabs

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

2. Web Tabs

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

1. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2. Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

3. Lightning Page Tabs

Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu. Lightning Page tabs don't work like other custom tabs. Once created, they don't show up on the All Tabs page when you click the Plus icon that appears to the right of your current tabs. Lightning Page tabs also don't show up in the Available Tabs list when you customize the tabs for your apps.

•

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

4. Creating a Custom Tab

To create a Tab:(Property)

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

The screenshot shows the Salesforce Setup interface with the URL sbcom-5e-dev-ed.lightning.force.com/lightning/setup/CustomTabs/home. The top navigation bar includes tabs for 'Setup', 'Home', and 'Deploy'. A red arrow points to the 'Home' tab. Below the navigation is a search bar with the text 'tab' and a red arrow pointing to it. 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, there's a sidebar with 'User Interface' and 'Rename Tabs and Labels' sections, with a 'Tabs' link highlighted by a red arrow. The main content area is titled 'Custom Tabs' and contains a section for 'Custom Object Tabs'. It lists various tabs with columns for Action, Label, Tab Style, and Description. A red arrow points to the 'New' button at the top right of this table.

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 URL <https://orgfarm-c9faa6fc6e-dev-ed.lightning.force.com/lightning/setup/CustomTabs/home>. The top navigation bar includes tabs for 'Setup', 'Home', and 'Object Manager'. A red arrow points to the 'Setup' tab. Below the navigation is a search bar with the text 'tab' and a red arrow pointing to it. 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, there's a sidebar with 'Feature Settings' and 'User Interface' sections, with a 'Tabs' link highlighted by a red arrow. The main content area is titled 'Custom Tabs' and contains a section for 'Custom Object Tabs'. It lists four tabs with columns for Action, Label, Tab Style, and Description. The tabs are: 'Edit | Del lease' (Bank tab), 'Edit | Del payment' (Books tab), 'Edit | Del property' (Airplane tab), and 'Edit | Del tenants' (Bell tab).

The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

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

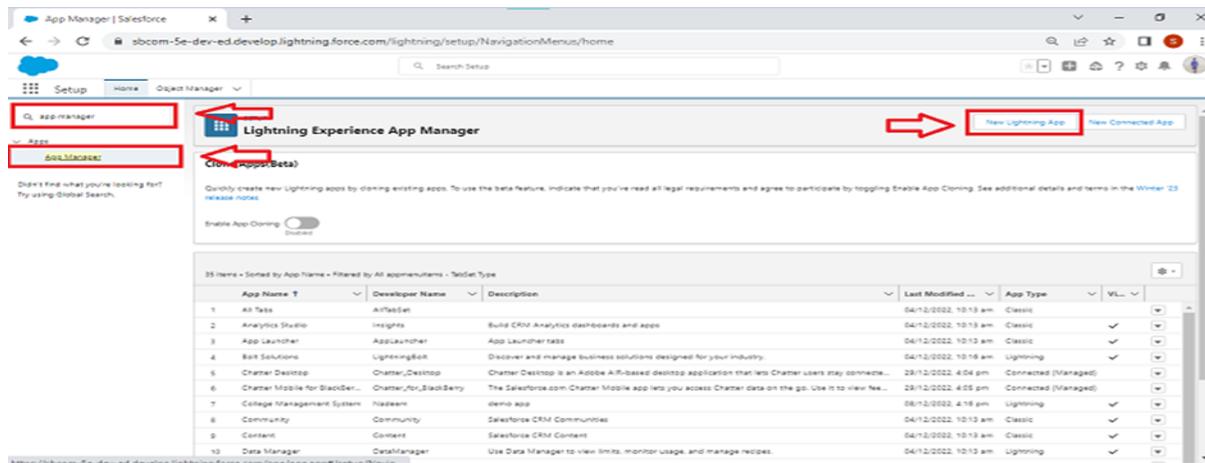
Use Case:

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

6. Create a Lightning App

To create a lightning app page:

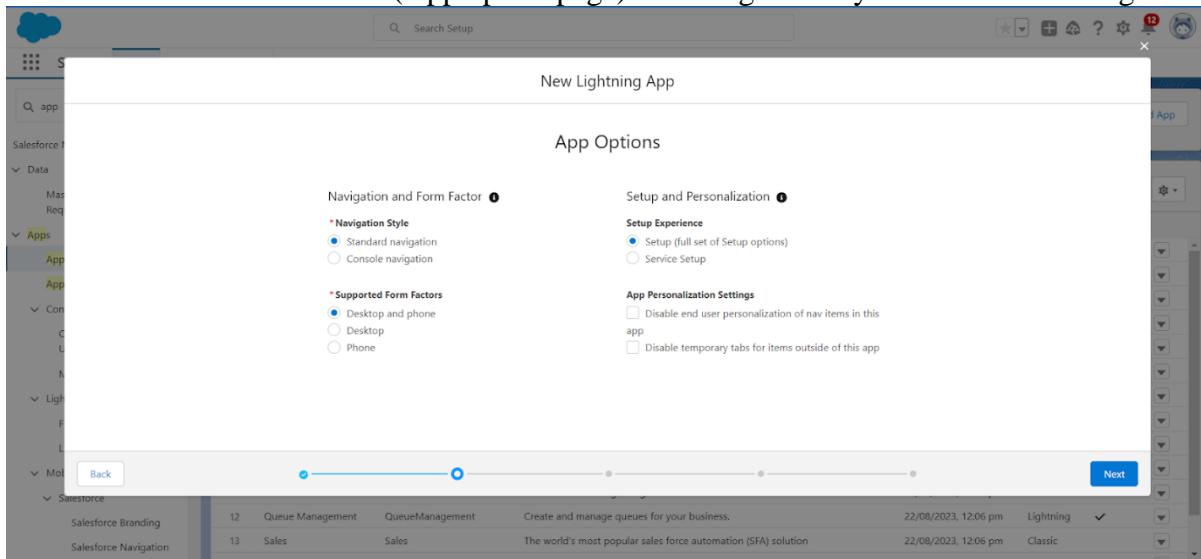
1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.



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

App Name : Lease 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.



(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' (indicated by a hand icon), 'Tenants' (indicated by an upward arrow icon), and 'property' (indicated by a house icon). Between the two lists are two small arrows pointing right and left, used for navigating between 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 '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' with a red box around it and a red arrow pointing to the right. At the bottom right of the screen is a 'Save & Finish' button with a red box around it and a red arrow pointing to 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 browser tabs for Object Manager, Garage Management system, Lease Management, and ChatGPT. Below the tabs, the address bar shows the URL for the Object Manager setup page. The main header includes a cloud icon, a search bar labeled 'Search Setup', and various navigation links like Setup, Home, and Object Manager. The 'Object Manager' section displays a table with one item: 'property' (API Name: property__c, Type: Custom Object). The table has columns for Label, API Name, Type, Description, Last Modified, and Deployed.

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

This screenshot shows the 'Fields & Relationships' page for the 'property' object. On the left, there's a sidebar with various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area is titled 'Fields & Relationships' and lists several fields: Address, Created By, Last Modified By, Name, Owner, property Name, sqft, and Type. Each field has its label, name, data type, controlling field, and indexed status. A red arrow points to the 'Fields & Relationships' link in the sidebar, and another red arrow points to the 'New' button at the top right of the list table.

3. Select Data Type as a “Text”

8. Creation of fields for the Tenant object

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

To create another fields in an object:

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

To create another fields in an object:

5. Go to setup >> click on Object Manager >> type object name(Tenant) in search bar >> click on the object.
6. Now click on “Fields & Relationships” >>New
7. Select Data type as a “picklist” and Click on Next
8. Fill the Above as following:
 - Field Label : status
 - Field Name : gets auto generated
 - Enter values, with each value separated by a new line
 - Enter these values
 - Stay
 - Leaving
 - Click on Next >> Next >> Save

Search Setup

SETUP > OBJECT MANAGER

property

Details

Fields & Relationships

- Date
- Date/Time
- Email
- Geolocation
- Number
- Percent
- Phone
- Picklist
- Picklist (Multi-Select)
- Text** (highlighted with a red box)
- Text Area
- Text Area (Long)
- Text Area (Rich)
- Text (Encrypted)
- Time
- URL

Allows users to enter a date and time, or pick a date from a pop-up calendar. When users click a date in the pop-up, that date and the current time are entered into the Date/Time field.

Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.

Allows users to define locations. Includes latitude and longitude components, and can be used to calculate distance.

Allows users to enter any number. Leading zeros are removed.

Allows users to enter a percentage number, for example, "10" and automatically adds the percent sign to the number.

Allows users to enter any phone number. Automatically formats it as a phone number.

Allows users to select a value from a list you define.

Allows users to select multiple values from a list you define.

Allows users to enter any combination of letters and numbers.

Allows users to enter up to 255 characters on separate lines.

Allows users to enter up to 131,072 characters on separate lines.

Allows users to enter formatted text, add images and links. Up to 131,072 characters on separate lines.

Allows users to enter any combination of letters and numbers and store them in encrypted form.

Allows users to enter a local time. For example, "2:40 PM", "14:40", "14:40:00", and "14:40:59:600" are all valid times for this field.

Allows users to enter any valid website address. When users click on the field, the URL will open in a separate browser window.

4. Click on next

SETUP > OBJECT MANAGER

property

Details

Fields & Relationships

Step 2. Enter the details

Step 2 of 4

Field Label (highlighted with a red box)

Length (highlighted with a red box)

Field Name (highlighted with a red box)

Description

Help Text

Required Always require a value in this field in order to save a record

Unique Do not allow duplicate values

Treat "ABC" and "abc" as duplicate values (case insensitive)

Treat "ABC" and "abc" as different values (case sensitive)

External ID Set this field as the unique record identifier from an external system

Auto add to custom report type Add this field to existing custom report types that contain this entity

Previous Next Cancel

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	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	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.
10. Now click on “Fields & Relationships” >> New
11. Select Data type as a “Text” and Click on Next
12. Fill the Above as following:
 - Field Label : sfqt
 - Field Name : gets auto generated
 - Length : 18
 - Click on Next >> Next >> Save.

2. To create another fields in an object:

1. Go to setup >> click on Object Manager >> type object name(property) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Long Text” and Click on Next
4. Fill the Above as following:
 - Field Label : Address
 - Field Name : gets auto generated
 - Click on Next >> Next >> Save and new.
3. To create another fields in an object:
 5. Go to setup >> click on Object Manager >> type object name(property) in search bar >> click on the object.

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

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 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

Specify the type of information that the custom field will contain.

Data Type

None Selected

Select one of the data types below.

Auto Number

A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.

Formula

A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.

Roll-Up Summary

A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.

Lookup Relationship

Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.

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 to have a master record.
- 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.

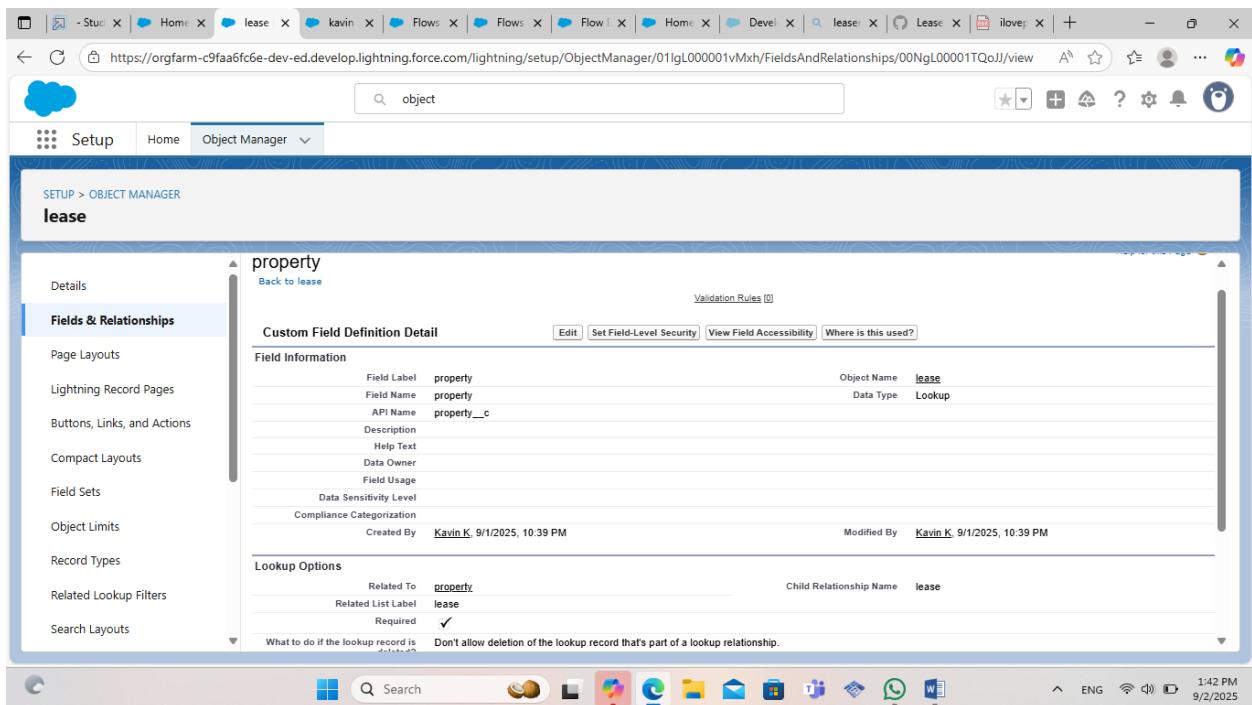
2. Now click on “Fields & Relationships” >> New
3. Select lookup relationship
4. Select the related object “ property” and click next.
5. Field Name : property
6. Field label : Auto generated
7. Next >> Next >> Save.

Creation of Lookup Field on Payment Object :

8. Go to setup >> click on Object Manager >> type object name(payment) in the search bar >> click on the object.
9. Now click on “Fields & Relationships” >> New
10. Select lookup relationship
11. Select the related object “ Tenant” and click next.
12. Field Name : Tenant
13. Field label : Auto generated
14. Next >> Next >> Save.

Creation of Lookup Field on Payment for tenant Object :

15. Go to setup>> click on Object Manager >> type object name(property) in the search bar >> click on the object.
16. Now click on “Fields & Relationships” >> New
17. Select masterdetail relationship
18. Select the related object “ property” and click next.
19. Field Name : property
20. Field label : Auto generated
21. Next >> Next >> Save.



12. Validation rule

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

13. To create a validation rule to an Lease Object

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

The screenshot shows the 'Validation Rules' list view for the 'lease' object in the Salesforce Setup. The left sidebar has a 'Validation Rules' item highlighted with a red box. The main area shows a table with columns: RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. A red box highlights the 'New' button in the top right corner of the table header.

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_end_date' rule. The 'Rule Name' field contains 'lease_end_date' and is highlighted with a red box. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains 'End_date__c > start_date__c' and is also highlighted with a red box. The formula editor interface is visible, showing function categories like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, and a dropdown menu for inserting selected functions.

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.

Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is true

Error Message: Your End date must be greater than start date

This error message can either appear at the top of the page or below a specific field on the page

Error Location: Top of Page Field **start date**



https://orgfarm-c9faa6fc6e-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001vMxh/ValidationRules/view

object

Setup Home Object Manager

SETUP > OBJECT MANAGER
lease

Validation Rules				
1 Items, Sorted by Rule Name				
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

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts

1:45 PM 9/2/2025

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 has "email t" typed into it. 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 contains 18 rows of data, each with a checkmark in the "Custom" column and a date in the last column. The rows include: "Sales: New Customer Email" (Custom, 8/28/2025), "Scheduled Service Appointment Confirmation Email" (Custom, 8/28/2025), "Scheduler Payments: Payment Reminder for Service Appointment Email" (Custom, 8/28/2025), "Scheduler Payments: Service Appointment Cancellation Email" (Custom, 8/28/2025), "Scheduler Payments: Service Appointment Confirmation Email" (Custom, 8/28/2025), "Scheduler Payments: Service Appointment Confirmation Email for Guest Users" (Custom, 8/28/2025), "Scheduler Payments: Service Appointment Rescheduled Email - Authenticated Users" (Custom, 8/28/2025), "Scheduler Payments: Service Appointment Rescheduled Email - Guest Users" (Custom, 8/28/2025), "SUPPORT: Self-Service New Comment Notification (SAMPLE)" (Text, 8/28/2025), "SUPPORT: Self-Service New User Login Information (SAMPLE)" (Text, 8/28/2025), "SUPPORT: Self-Service Reset Password (SAMPLE)" (Text, 8/28/2025), "Support: Case Assignment Notification" (Text, 8/28/2025), "Support: Case Created (Phone Inquiries)" (Text, 8/28/2025), and "Support: Case Created (Web Inquiries)" (Text, 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"

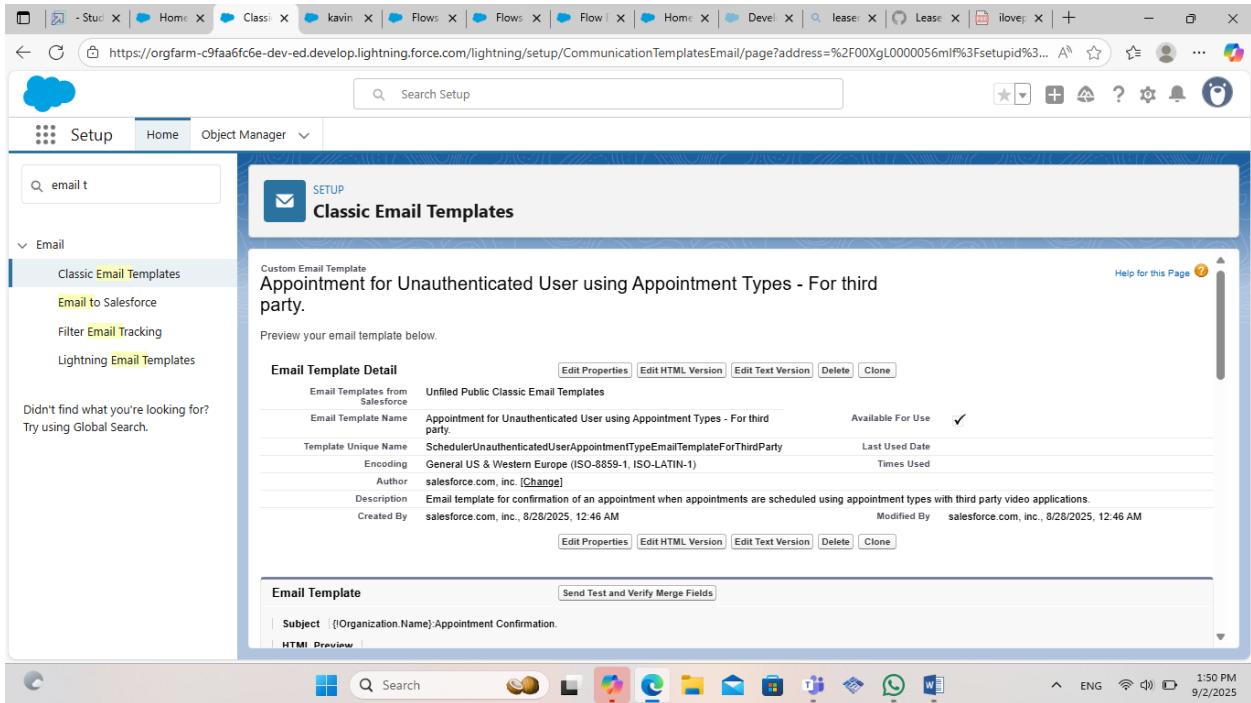
6. Email body :

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"

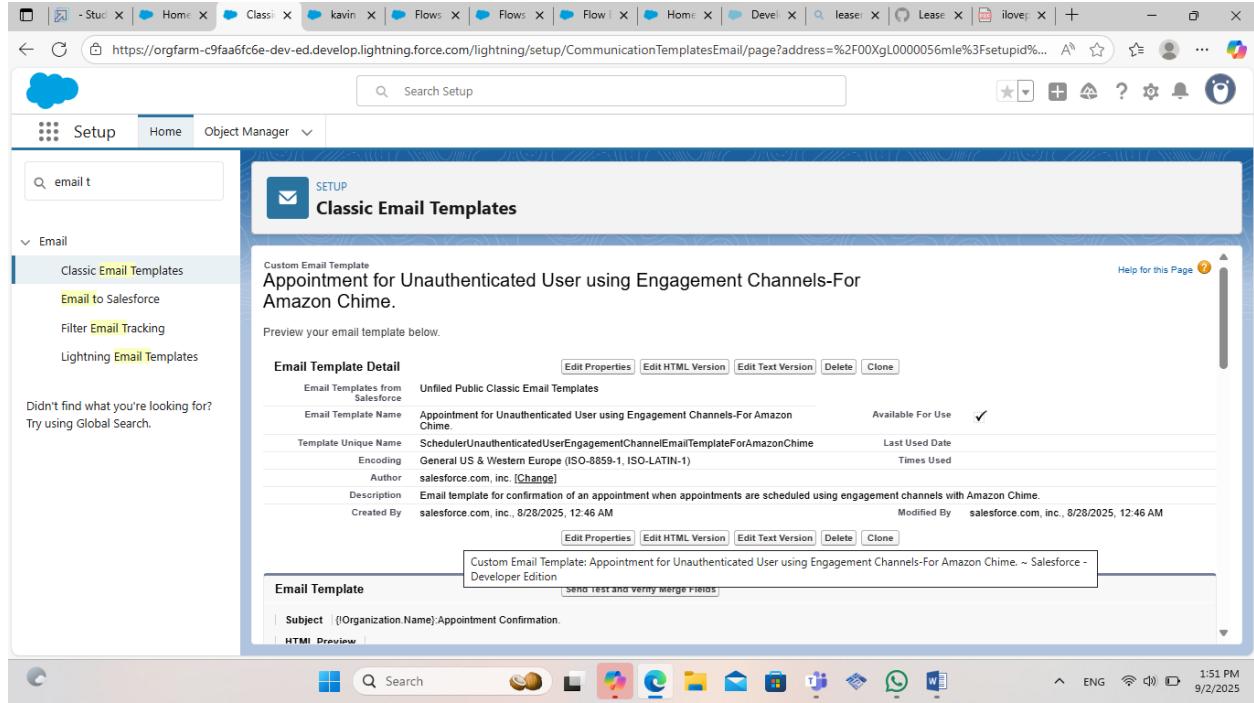
6. Email body :

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

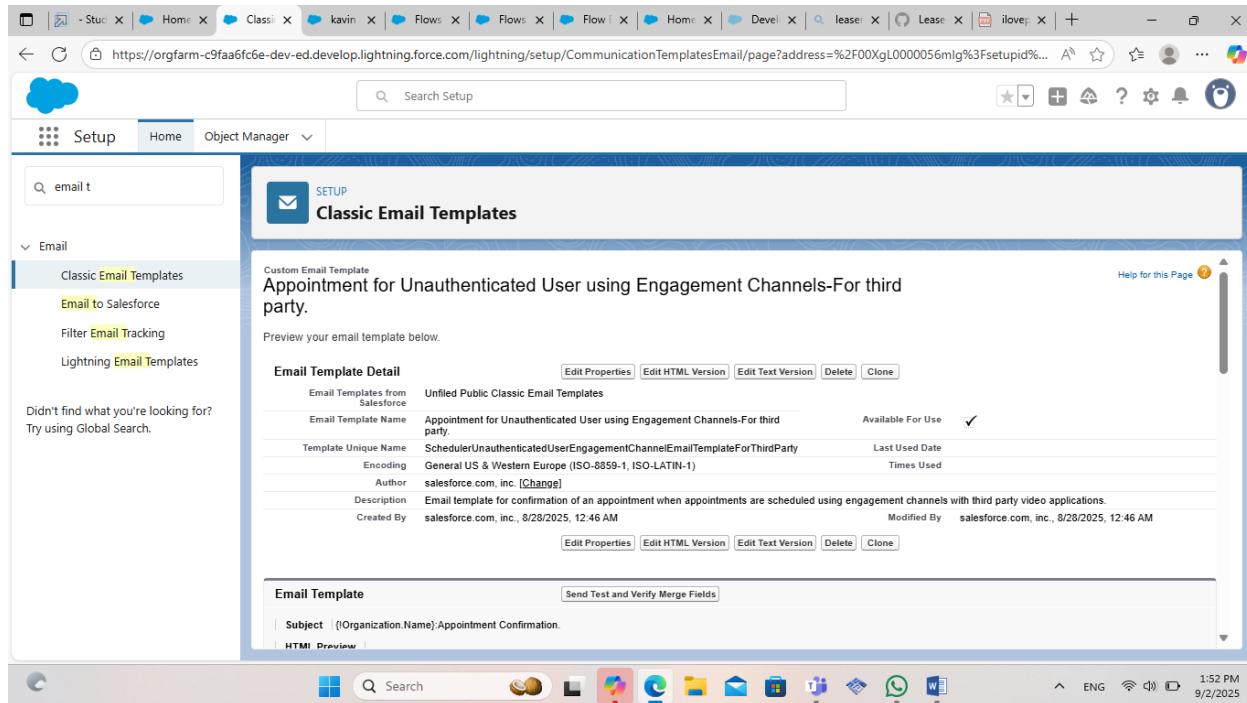
17. 3. Email Template Name is "Tenant Email"

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 for managing email templates. The user is in the 'Classic Email Templates' section under the 'Email' category. The page title is 'Unfiled Public Classic Email Templates'. A table lists several templates, each with a description, author, and last modified date. The descriptions mention various appointment types and confirmation emails. The interface includes standard Salesforce navigation elements like a search bar, a breadcrumb trail, and a toolbar at the top.

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit Del	Appointments - 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.

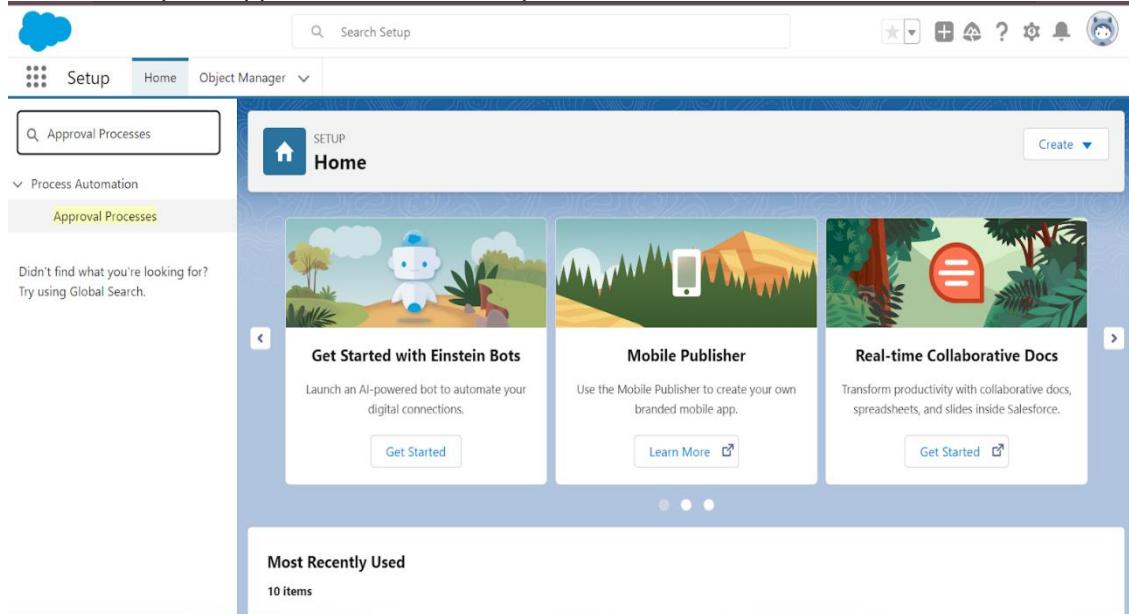
An approval process consists of finalizing the basic properties of the approval process (as shown in the below image), approval steps, and actions to be executed.

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.

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

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

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

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

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

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

- Then click save.

What Would You Like To Do Now?

You have just created an approval process. However, you cannot activate this process until you define at least one approval step. Would you like to do that now?

- 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.

Go!

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

Search Setup

Approval Processes

Process Automation

Approval Processes

Didn't find what you're looking for?
Try using Global Search.

Manage Approval Processes For: tenant

A listing of both active and inactive approval processes for tenants is displayed below. To create a new approval process, click Create New Approval Process then select Use Jump Start Wizard to set up your approval process in a few short steps. Or, select Use Standard Wizard to configure all approval options.

Create New Approval Process

Action	Approval Process Name	Description
Edit Activate Del	check for vacant	

2. Initial Submission Action:

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

Initial Submission Actions		Add Existing	Add New
Action	Type	Description	Task
Record Lock		Lock this record	
		Email Alert	
		Field Update	

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:

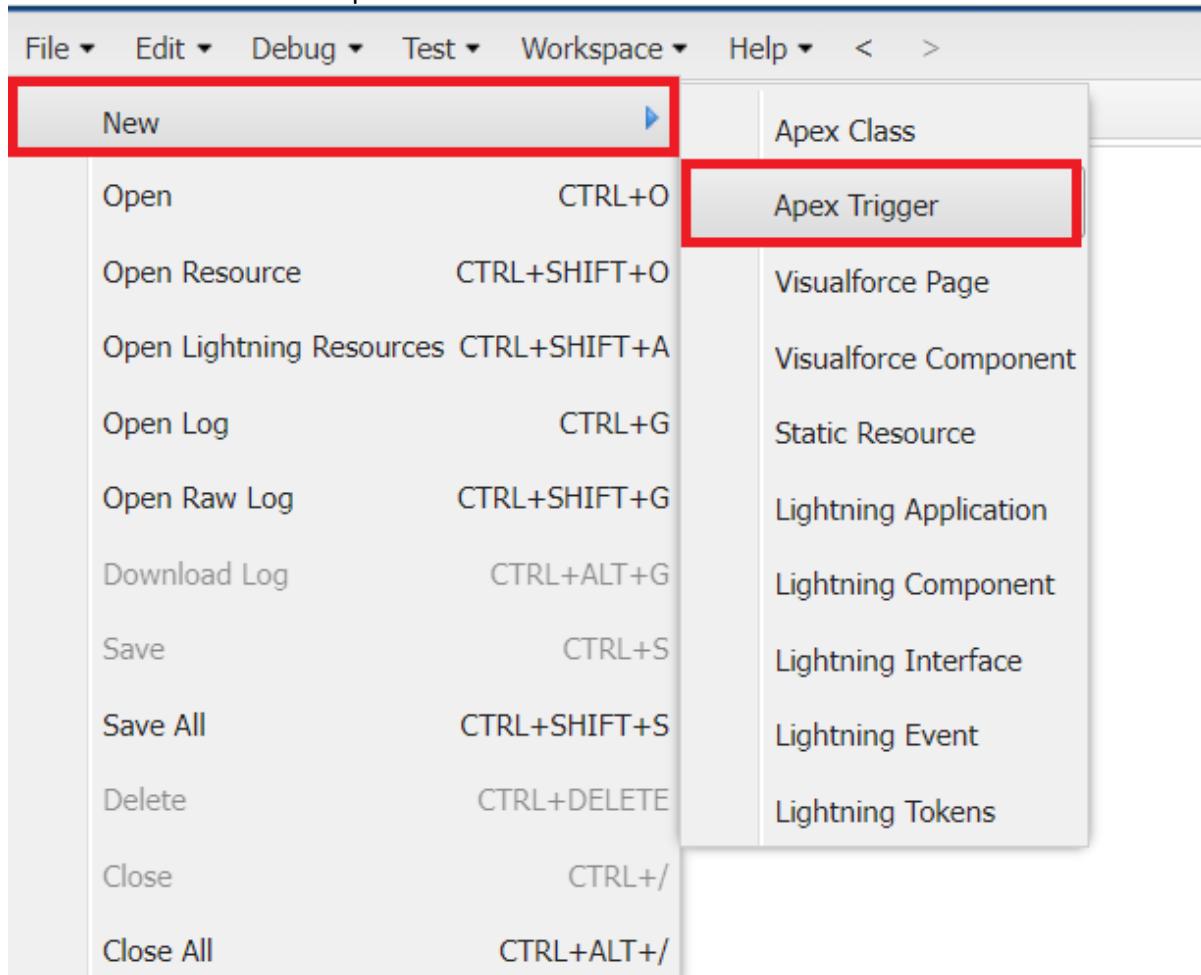
The tenant and property are in a master-detail relationship, wherein each tenant is associated with only one property. When a tenant attempts to create a new record with an existing property, an error should be displayed, indicating that a tenant can have only one property.

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 }
```

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 'New Tenant' form. The 'Information' section contains fields for 'Tenant Name' (niranjan), 'Phone' (empty), 'Email' (empty), and 'status' (stay). A dropdown menu for 'property' is open, showing 'Manne R' as the selected item. 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.

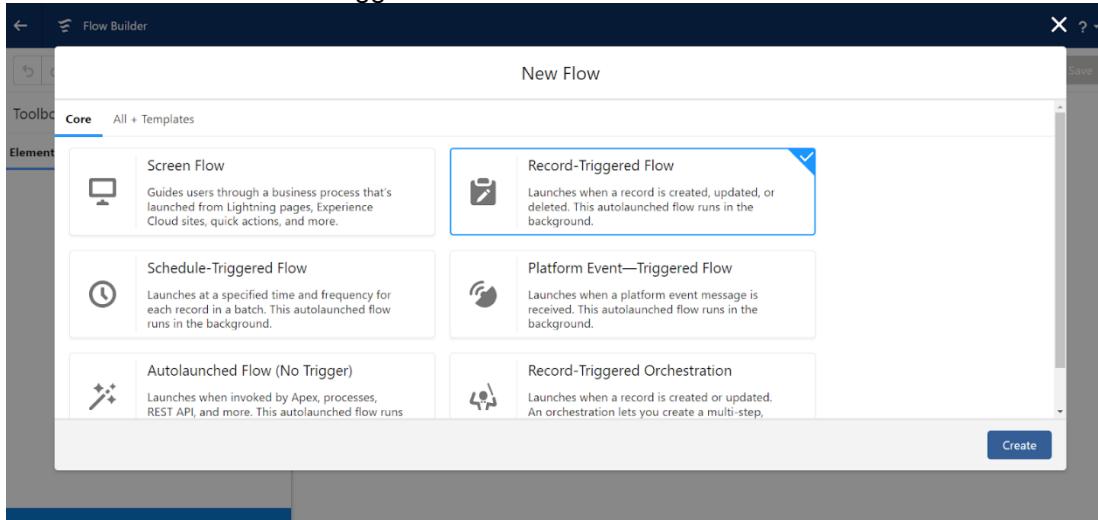
When and why should we use a flow

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. The object 'Payment for tenant' is selected in the dropdown. Below it, the 'Configure Trigger' section shows the 'Trigger the Flow When:' section with the option 'A record is updated' selected (indicated by a blue border).

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:

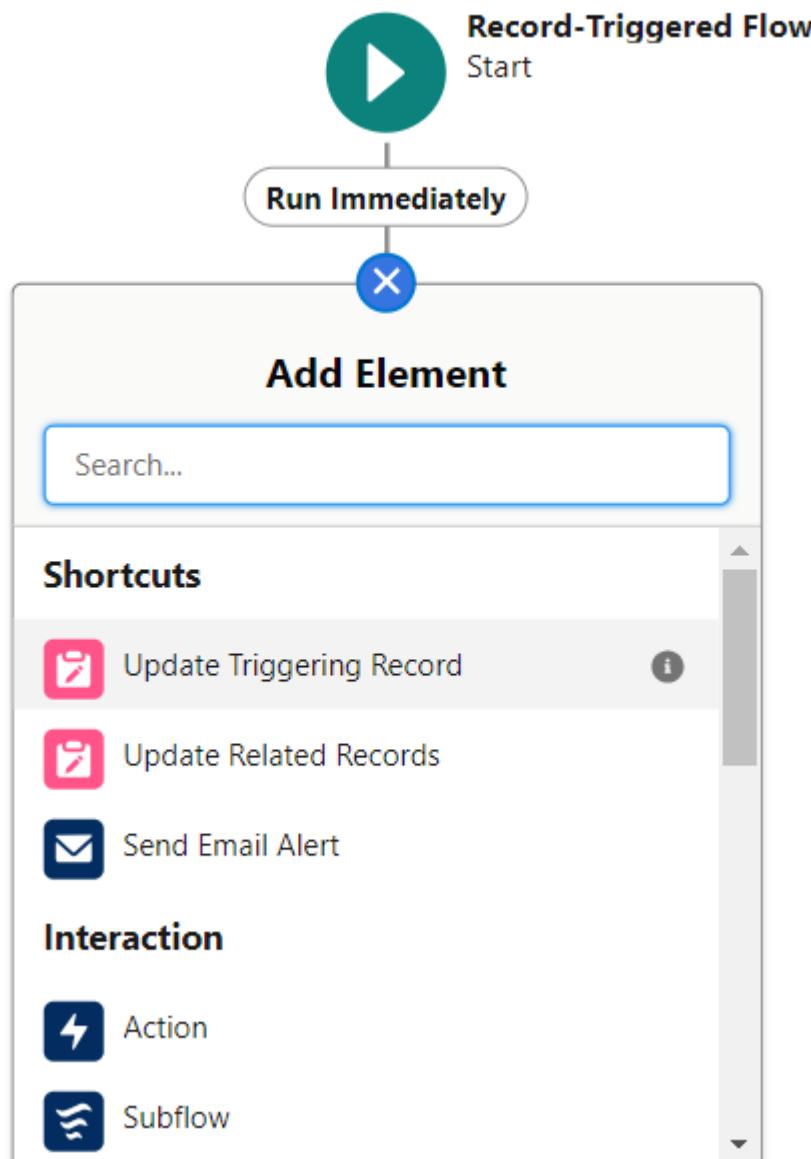
Fast Field Updates

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.

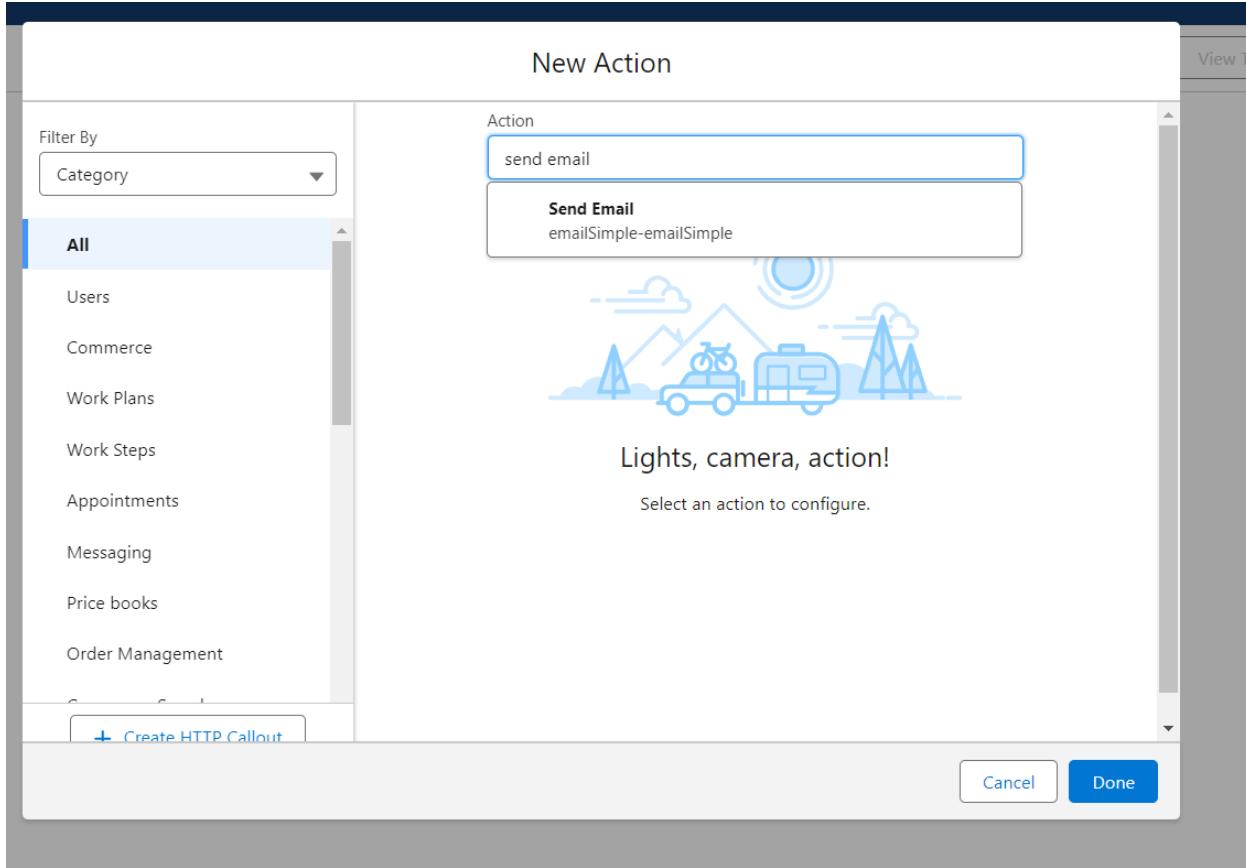
7. Under record trigger flow click on “+” icon and select action



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 'Email'. The results list includes 'send_email' under 'ACTIONS' and several global constants like '\$GlobalConstant.EmptyString' and '\$GlobalConstant.True'. At the bottom right is a toggle switch labeled 'Include' with its checkbox checked, also highlighted by a red box.

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

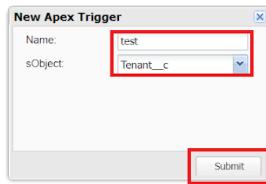
Flow label : monthly payment

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.
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);
    }
}
```

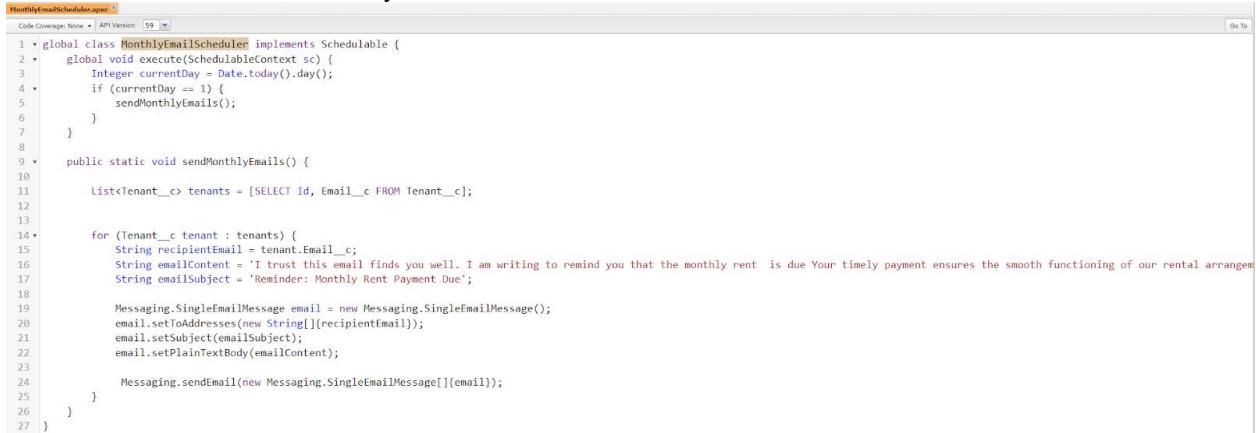
4. Schedule class :

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 all classes](#)

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 []

End []

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

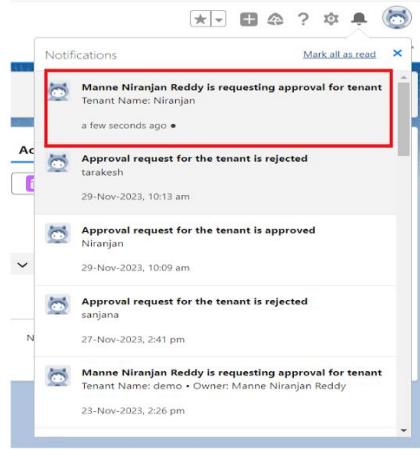
Submit for Approval

Comments

Leaving

Cancel Submit

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

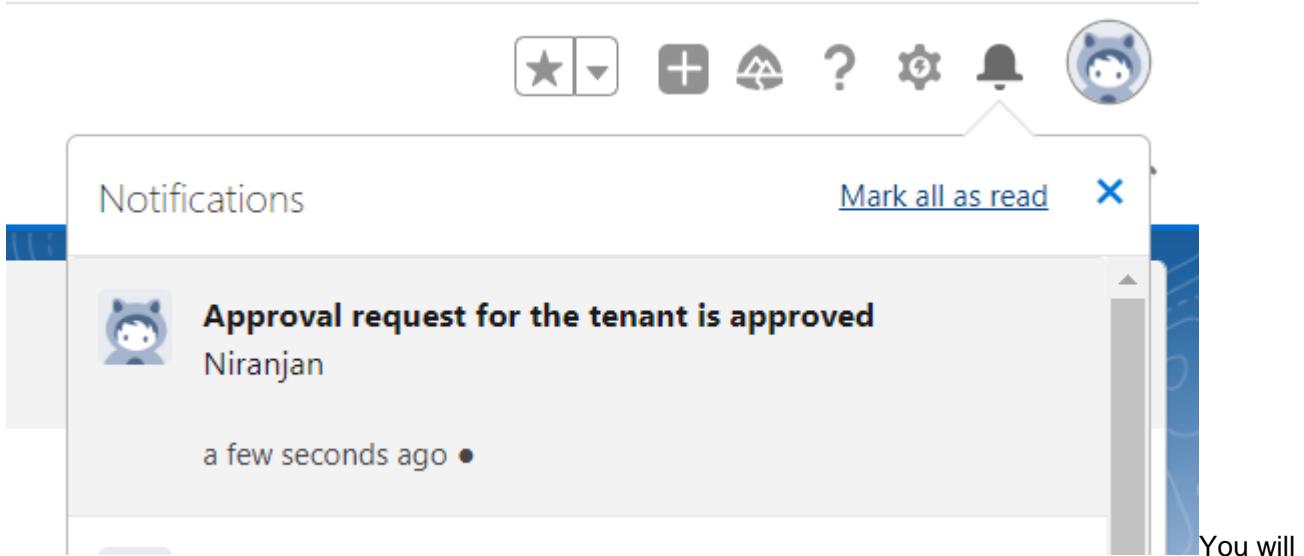


Click on that notification

A screenshot of a web application interface for a tenant approval request. At the top, there is a header with a user icon, the text "Approval Request", "Tenant Approval", and a status indicator "Pending". Below the header, there are four data fields: "Submitter" (Manne Niranjana Reddy), "Date Submitted" (05-Dec-2023), "Actual Approver" (Manne Niranjana Reddy), and "Assigned To" (Manne Niranjana Reddy). On the right side, there are three buttons: "Approve" (highlighted with a red box), "Reject", and "Reassign". The main body of the page is divided into two sections: "Details" (containing "Approval Details" and "Submitter Name: Niranjan") and "Submitter Comments" (containing a comment from Manne Niranjana Reddy about leaving).

on approve
Give any comment and submit

click



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.lightning.force.com/lightning/setup/ApexClasses/home. The page title is "Apex Classes". On the left, there is a sidebar with a search bar and sections for Email, Custom Code (with "Apex Classes" selected), Environments, and Jobs. A message at the bottom of the sidebar says "Didn't find what you're looking for? Try using Global Search." The main content area shows a summary of Apex usage: "Percent of Apex Used: 0%" and "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 "Compile all classes" button. A table lists one Apex class: "testHandler" (Name, Namespace Prefix, Api Version 64.0, Status Active, Size Without Comments 29, Last Modified By Kavin K, 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 status bar at the bottom right shows the time as 2:37 PM and the date as 9/2/2025.