

LEASE MANAGEMENT

COLLEGE NAME : SREE NARAYANA GURU COLLEGE

COLLEGE CODE : Bru36

TEAM ID: NM2025TMID26405

TEAM MEMBERS:

Team LeaderName : Arya GirishKumar

Email : aryagirish101@gmail.com

Team Member 1 : Athul Krishna

Email : athuloffix@gmail.com

Team Member 2 : Aysha Ruba

Email : aysharuba31@gmail.com

Team Member 3 : Aswin Ganga

Email : aswinganga16@gmail.com

Team Member 4 : Basil Biju

Email : Basilbiju2244@gmail.com

1. INTRODUCTION

1.1 Project Overview

The Lease Management System is a Salesforce-based application that automates and streamlines real estate leasing processes. It manages tenant records, lease contracts, rent payments, and communication within a centralized platform, reducing manual effort and errors. Using Salesforce features like Flows, Approval Processes, and Email Alerts, the system ensures timely reminders, accurate payment tracking, and proper contract approvals. With dashboards and reports, stakeholders gain insights into occupancy, payments, and renewals, enabling better decision-making. Overall, it enhances efficiency, improves tenant satisfaction, and secures data with role-based access.



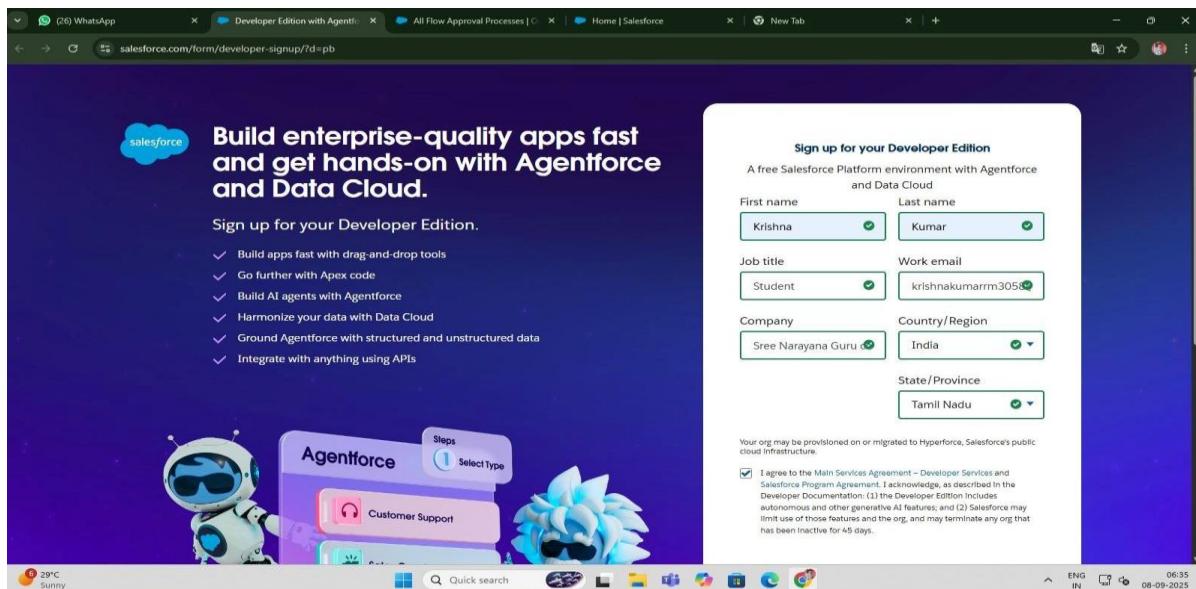
1.2 Purpose

The purpose of the Lease Management System is to provide a centralized, automated solution for managing properties, tenants, and lease activities. By using Salesforce features such as Flows, Approval Processes, and Email Alerts, the system reduces manual effort, improves accuracy, and ensures timely communication. It enhances compliance through secure role-based access and strengthens tenant–landlord relationships with transparent, efficient processes, ultimately increasing productivity and supporting better decision-making.

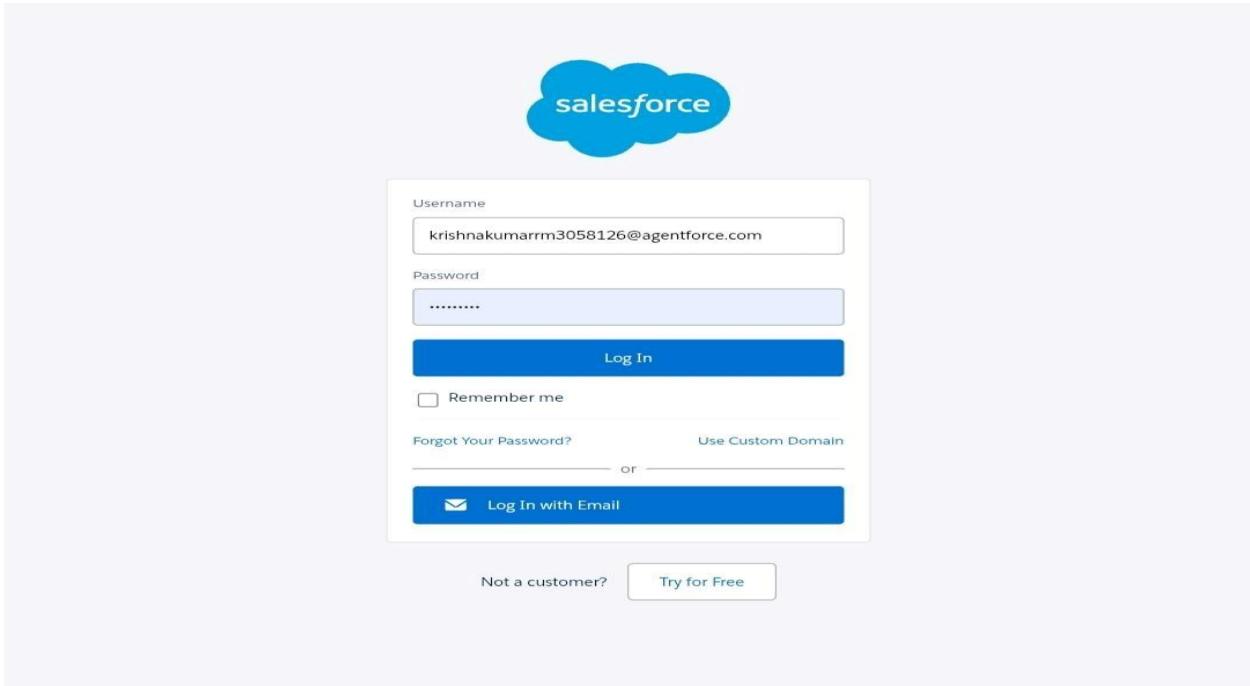
DEVELOPMENT PHASE

- **Creating Developer Account:**

By using this URL - <https://www.salesforce.com/form/developer-signup/?d=pb>



Account activation then next login this link <https://login.salesforce.com/>



- Created objects: Property, Tenant, Lease, Payment

A screenshot of the Salesforce Setup interface, specifically the Object Manager. The title bar shows multiple tabs including "Developer Edition with Agents", "Home | Salesforce", "Tenant | Salesforce", and "New Tab". The main area is titled "SETUP > OBJECT MANAGER" and shows a list of objects. The "Tenant" object is selected and displayed in detail. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Buttons, Links, and Actions. The right panel shows the "Details" for the Tenant object, including its API name "Tenant__c", custom singular label "Tenant", and plural label "Tenants". It also shows deployment status "Deployed" and help settings pointing to "Standard salesforce.com Help Window". There are "Edit" and "Delete" buttons at the top right of the detail panel.

The screenshot shows the Salesforce Setup interface for the 'Object Manager'. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main content area displays the 'Details' tab for the 'lease' object. The 'API Name' is set to 'lease__c'. The 'Custom' checkbox is checked. The 'Singular Label' is 'lease' and the 'Plural Label' is also 'lease'. On the right, there are sections for 'Description', 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window'). At the bottom right of the page, there is a status bar showing '29°C Sunny', system icons, and the date/time '08-09-2025 06:46'.

The screenshot shows the Salesforce Setup interface for the 'Object Manager'. The left sidebar lists various configuration options: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main content area displays the 'Details' tab for the 'property' object. The 'API Name' is set to 'property__c'. The 'Custom' checkbox is checked. The 'Singular Label' is 'property' and the 'Plural Label' is 'properties'. On the right, there are sections for 'Description', 'Enable Reports' (checked), 'Track Activities' (checked), 'Track Field History' (checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window'). At the bottom right of the page, there is a status bar showing '29°C Sunny', system icons, and the date/time '08-09-2025 06:45'.

The screenshot shows the Salesforce Setup interface for the 'Object Manager'. The top navigation bar includes tabs for 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Payment for tenantat'. On the left, a sidebar lists various configuration options under 'Details'. The main content area is titled 'Details' and contains sections for 'Description', 'API Name' (set to 'Payment_for_tenantat__c'), 'Custom' (checked), 'Singular Label' (set to 'Payment for tenantat'), 'Plural Label' (set to 'Payment'), and 'Enable Reports' (checked). Buttons for 'Edit' and 'Delete' are located at the top right of the main content area. The bottom of the screen shows the Windows taskbar with icons for weather (29°C, sunny), quick search, and various application icons.

Configured fields and relationships

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'property' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, etc. The main content area displays a table titled 'Fields & Relationships' with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(25)		
Owner	OwnerId	Lookup(User,Group)	✓	
property	property__c	Lookup(property)	✓	
property Name	Name	Text(80)	✓	
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

The screenshot shows the Salesforce Setup interface under the Object Manager for the 'Payment for tenant' object. The left sidebar lists various setup categories. The main content area displays a table titled 'Fields & Relationships' with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Number(18, 0)		
check for payment	check_for_payment__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)	✓	
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(80)	✓	
Tenant	Tenant__c	Lookup(Tenant)	✓	

Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
End date	End_date_c	Date		
Last Modified By	LastModifiedBy	Lookup(User)		
lease Name	Name	Text(80)		
Owner	OwnerId	Lookup(User,Group)		
property	property_c	Lookup(property)		
start date	start_date_c	Date		

Fields & Relationships
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone	Phone_c	Phone		
property	property_c	Lookup(property)		
status	status_c	Picklist		
Tenant Name	Name	Text(80)		

Developed Lightning App with relevant tabs

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name

*Developer Name

Description

App Branding

Image Primary Color Hex Value

Org Theme Options Use the app's image and color instead of the org's custom theme

App Launcher Preview

LM Lease Management

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

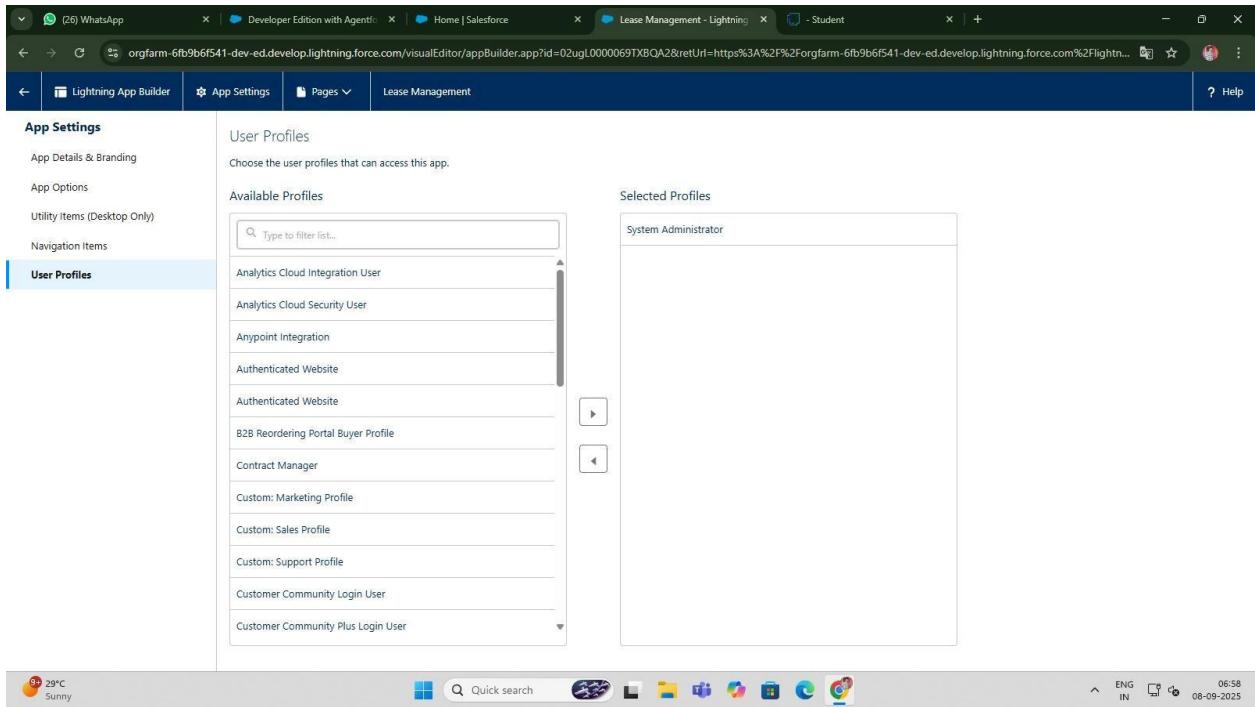
Available Items

Type to filter list...

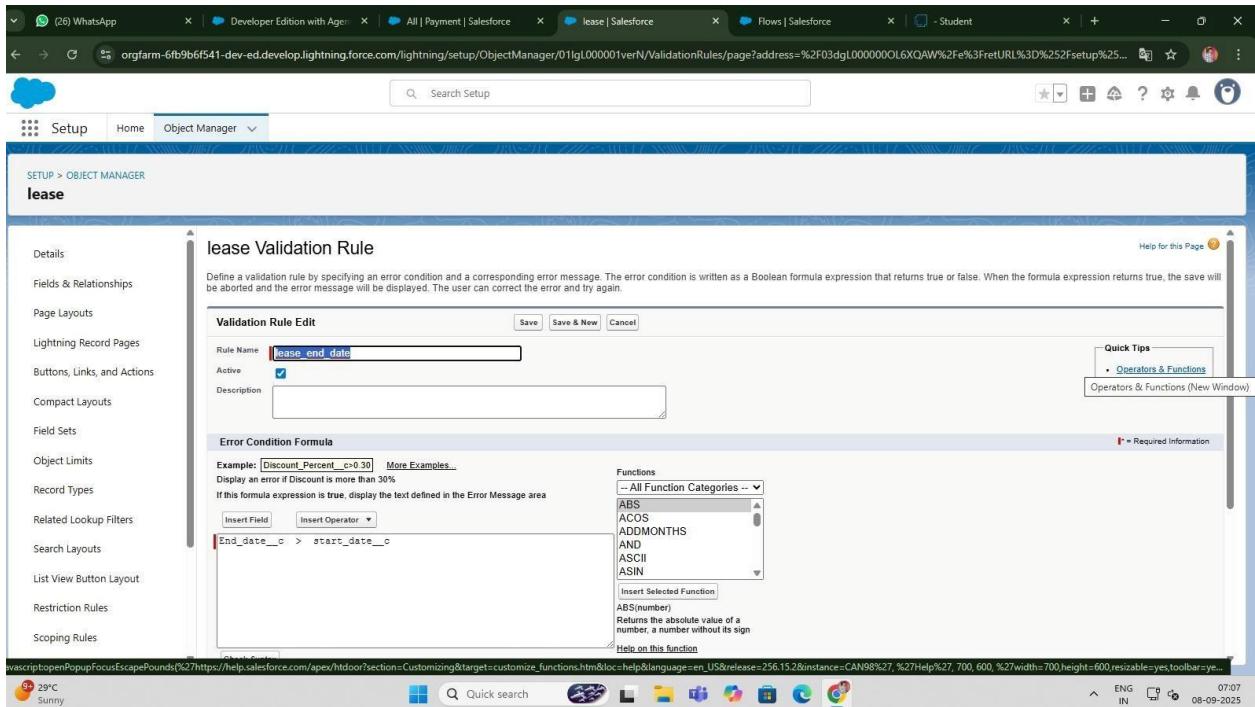
- Accounts
- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests
- Approval Submission Details

Selected Items

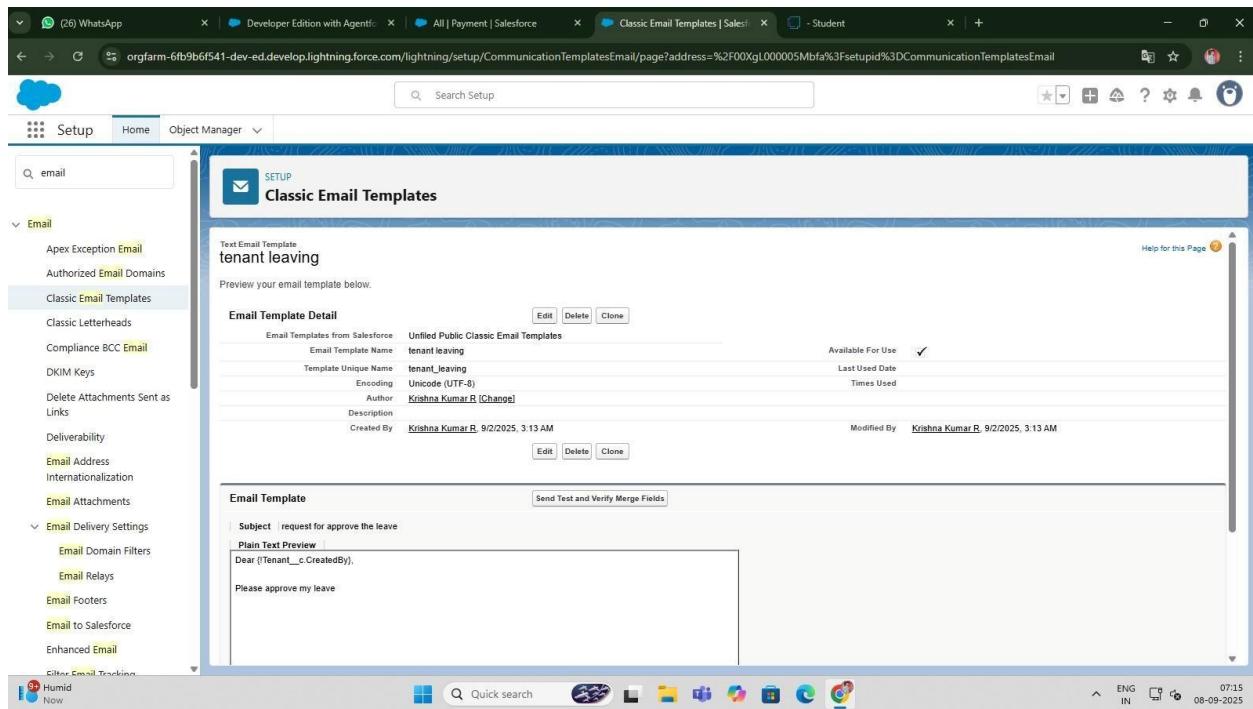
- Home
- properties
- Payment
- Tenants
- lease



- create a validation rule to an Lease Object



- Built and tested email templates for leave request, approval, rejection, payment, and reminders



Classic Email Templates

Leave approved

Preview your email template below.

Email Template Detail	
Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave approved
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	Krishna Kumar R [Change]
Description	
Created By	Krishna Kumar R 9/2/2025, 3:17 AM
Modified By	Krishna Kumar R 9/2/2025, 3:17 AM

Email Template

Subject | Leave approved

Plain Text Preview

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

Leave rejected

Preview your email template below.

Email Template Detail	
Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Krishna Kumar R [Change]
Description	
Created By	Krishna Kumar R 9/2/2025, 3:24 AM
Modified By	Krishna Kumar R 9/2/2025, 3:24 AM

Email Template

Subject | Leave rejected

Plain Text Preview

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

The screenshot shows the Salesforce Setup interface for managing email templates. The left sidebar is collapsed, and the main area displays the 'Classic Email Templates' page. A specific template named 'Tenant Email' is selected, shown in a detailed view. The template details include:

Field	Value
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Krishna Kumar R [Change]
Description	
Created By	Krishna Kumar R 9/2/2025, 3:28 AM
Modified By	Krishna Kumar R 9/2/2025, 3:28 AM

The 'Email Template' section contains the following content:

Subject: Urgent: Monthly Rent Payment Reminder

Plain Text Preview:

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.

The bottom of the page includes a weather widget showing '29°C Mostly sunny', a quick search bar, and system status indicators.

The screenshot shows the Salesforce Setup interface with the 'Email' category selected. Under 'Classic Email Templates', the 'tenant payment' template is selected. The page displays the 'Email Template Detail' and a preview of the email template content.

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	tenant payment
Template Unique Name	tenant_payment
Encoding	Unicode (UTF-8)
Author	Krishna Kumar R [Change]
Description	
Created By	Krishna Kumar R 9/2/2025, 3:30 AM
Modified By	Krishna Kumar R 9/2/2025, 3:30 AM

Email Template

Subject: Confirmation of Successful Monthly Payment

Plain Text Preview:

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.

● Approval Process creation For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Data' category selected. Under 'Process Automation', the 'Approval Processes' option is selected. A new approval process named 'check for vacant' is being created, currently at Step 1 of 6.

Step 1. Enter Name and Description

Enter Name and Description

Process Name: check for vacant
Unique Name: check_for_vacant
Description:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The process name is 'check for vacant'. The 'Process Definition Detail' section includes fields for Process Name (check for vacant), Unique Name (check_for_vacant), Description (Tenant: status NOTEQUAL TO Leaving), and Record Editability (Administrator ONLY). The 'Initial Submission Actions' section contains a single entry: 'Record Lock' with the description 'Lock the record from being edited' and 'please approve my leave'. The 'Approval Steps' section is currently empty.

● Initial Submission Action

The screenshot shows the Salesforce Home page with the 'Approval Processes' page open. The process name is 'check for vacant'. The 'Initial Submission Actions' section has a context menu open, showing options: Task, Email Alert, Field Update, and Outbound Message. The 'Approval Steps' section is currently empty.

The screenshot shows the Salesforce Email Alert detail page for an alert named "please approve my leave".

Email Alert Detail:

- Description: please approve my leave
- Unique Name: please_approve_my_leave
- From Email Address: Current User's email address
- Recipients: Email Field: Email
- Additional Emails: (empty)
- Created By: Krishna Kumar R, 9/8/2025, 8:05 AM
- Modified By: Krishna Kumar R, 9/8/2025, 8:05 AM

Rules Using This Email Alert: This alert is currently not used by any rules.

Approval Processes Using This Email Alert:

Action	Approval Process Name	Description	Type	State
Edit Del	check_for_vacant	(empty)	Tenant	Active

Entitlement Processes Using This Email Alert: This alert is currently not used by any entitlement processes.

Flows Using This Email Alert: (empty)

Back To Top Always show me ▾ more records per related list

● Final Approval Action

The screenshot shows the Salesforce Email Alert detail page for an alert named "Tenant leaving".

Email Alert Detail:

- Description: Tenant leaving
- Unique Name: Tenant_leaving
- From Email Address: Current User's email address
- Recipients: Email Field: Email
- Additional Emails: (empty)
- Created By: Krishna Kumar R, 9/8/2025, 8:07 AM
- Modified By: Krishna Kumar R, 9/8/2025, 8:07 AM

Rules Using This Email Alert: This alert is currently not used by any rules.

Approval Processes Using This Email Alert:

Action	Approval Process Name	Description	Type	State
Edit Del	check_for_vacant	(empty)	Tenant	Active

Entitlement Processes Using This Email Alert: This alert is currently not used by any entitlement processes.

Flows Using This Email Alert: (empty)

Back To Top Always show me ▾ more records per related list

● Final Rejection Action

salesforce 125

Search... Search ▾ Switch to Lightning Experience Krishna Kumar R ▾ Setup Help Sales

Home Chatter Campaigns Leads Accounts Contacts Opportunities Forecasts Contracts Orders Cases Solutions + ▾

Quick Find / Search... Expand All | Collapse All

Salesforce Mobile Quick Start

Home

Administer

- Release Updates
- Manage Users
- Manage Apps
- Manage Territories
- Company Profile
- Data Classification
- Privacy Center
- Security Controls
- Domain Management
- Communication Templates
- Translation Workbench
- Data Management
- Mobile Administration
- Desktop Administration
- Outlook Integration and Sync
- Gmail Integration and Sync
- Email Administration
- Google Apps
- Analytics
- Tableau
- Data.com Administration

Build

Email Alert
your request for leave is rejected

Printable View | Help for this Page

Rules Using This Email Alert | Approval Processes Using This Email Alert | Entitlement Processes Using This Email Alert

Email Alert Detail

		Edit	Delete	Clone
Description	your request for leave is rejected	Email Template	Leave rejected	
Unique Name	your_request_for_leave_is_rejected	Object	Tenant	
From Email Address	Current User's email address			
Recipients	Email Field: Email			
Additional Emails				
Created By	Krishna Kumar R, 9/8/2025, 8:08 AM	Modified By	Krishna Kumar R, 9/8/2025, 8:08 AM	
		Edit	Delete	Clone

Rules Using This Email Alert

Rules Using This Email Alert Help

This alert is currently not used by any rules

Approval Processes Using This Email Alert

Approval Processes Using This Email Alert Help

Action	Approval Process Name	Description	Type	State
Edit Del	check_for_vacant		Tenant	Active

Entitlement Processes Using This Email Alert

This alert is currently not used by any entitlement processes

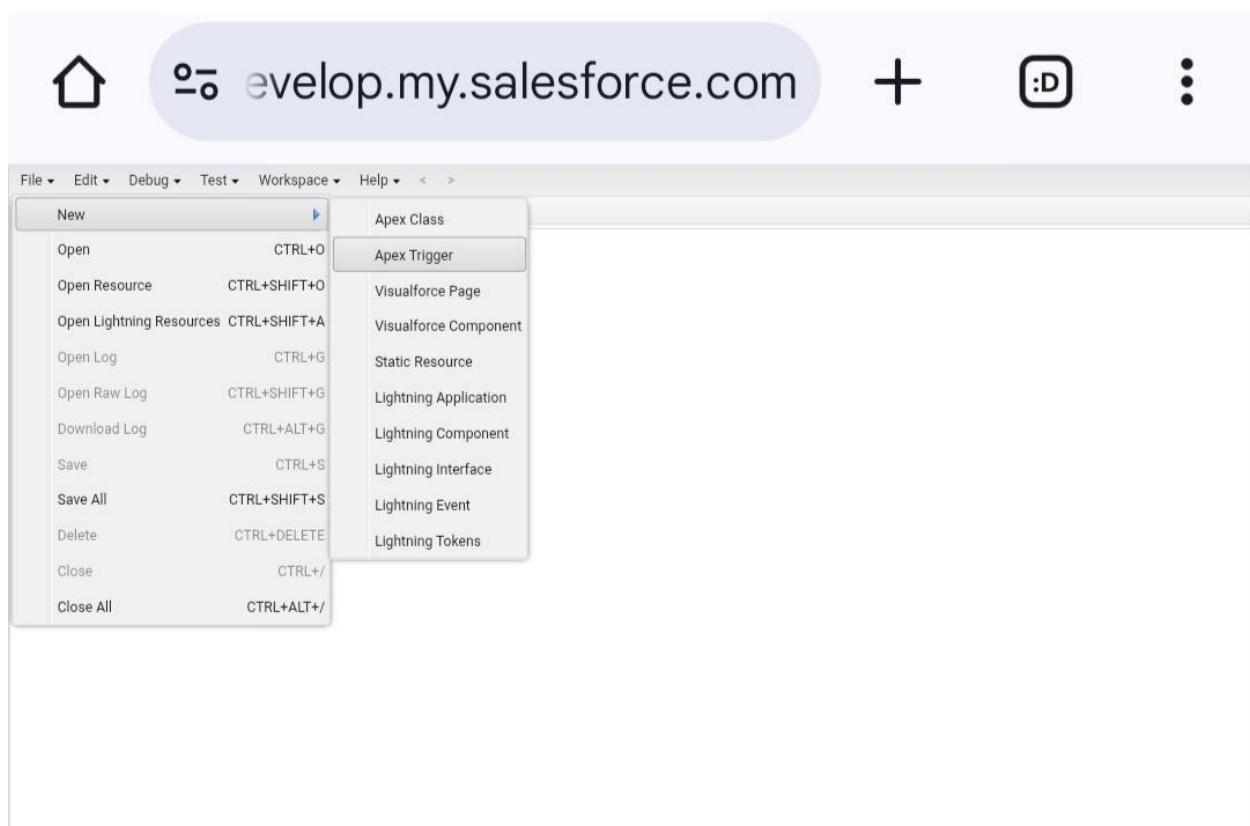
Flows Using This Email Alert

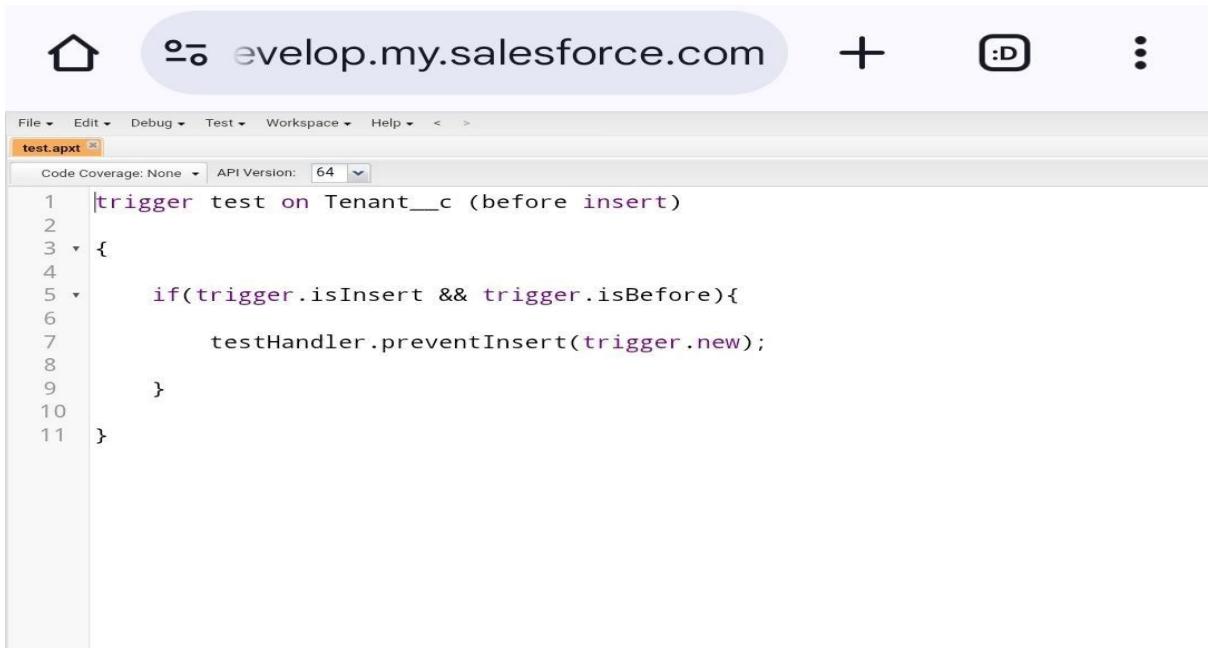
Flow Name	Version	Description	Object	Active

[Back To Top](#) Always show me more records per related list

Apex Trigger

- Create an Apex Trigger

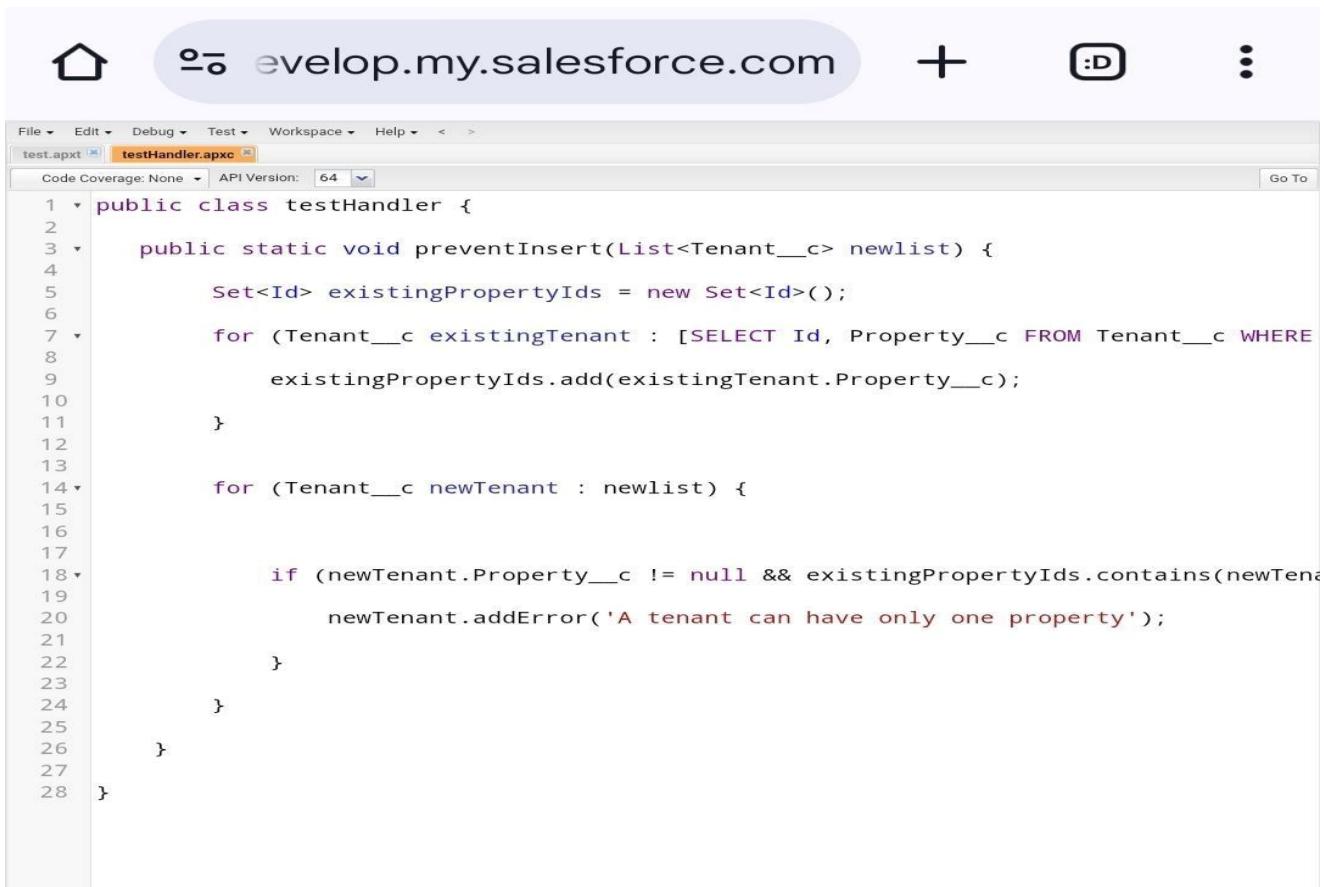




The screenshot shows the Salesforce developer console interface. At the top, there's a navigation bar with links for File, Edit, Debug, Test, Workspace, and Help. Below the navigation bar is a toolbar with icons for Home, Refresh, and a three-dot menu. The main area displays a code editor for a trigger named "test.apxt". The code is as follows:

```
1 trigger test on Tenant__c (before insert)
2
3 {
4     if(trigger.isInsert && trigger.isBefore){
5         testHandler.preventInsert(trigger.new);
6     }
7 }
```

- Create an Apex Handler class



The screenshot shows the Salesforce developer console interface. At the top, there's a navigation bar with links for File, Edit, Debug, Test, Workspace, and Help. Below the navigation bar is a toolbar with icons for Home, Refresh, and a three-dot menu. The main area displays a code editor for an Apex Handler class named "testHandler.apxc". The code is as follows:

```
1 public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4         Set<Id> existingPropertyIds = new Set<Id>();
5
6         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE
7             existingPropertyIds.add(existingTenant.Property__c);
8
9         }
10
11
12
13         for (Tenant__c newTenant : newList) {
14
15             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
16                 newTenant.addError('A tenant can have only one property');
17             }
18         }
19     }
20
21
22 }
```

- Testing the Trigger

Stay

property

Hyd

Created By Krishna Kumar R, 9/2/2025, 5:19 AM

Last Modified By Krishna Kumar R, 9/10/2025, 8:22 AM

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

New Tenant

* = Required Information

Information

* Tenant Name Krishna Kumar R

* Email krishnakumarrm3058@gmail.com

Phone

Status Stay

We hit a snag.

Review the errors on this page.

- A tenant can have only one property

The screenshot shows a 'New Tenant' form within a software interface. At the top left, there's a sidebar with 'Stay' and 'property' sections, and a note 'No past activity. Past meetings and tasks marked as done show up here.' On the right, there's a close button (X). The main form has a title 'New Tenant' and a note '* = Required Information'. It contains an 'Information' section with fields for 'Tenant Name' (Krishna Kumar R) and 'Email' (krishnakumarrm3058@gmail.com), both marked with a red asterisk. Below these are empty fields for 'Phone' and 'Status' (set to 'Stay'). A prominent red error box at the bottom left says 'We hit a snag.' with a red circle icon. It lists an error: 'Review the errors on this page.' followed by a bullet point: 'A tenant can have only one property'. At the bottom are three buttons: 'Cancel' (gray), 'Save & New' (light blue), and 'Save' (blue).

Create Flow for monthly payment

The screenshot shows the Flow Builder interface for a 'monthly payment' flow. The flow starts with a 'Record-Triggered Flow' trigger for 'Payment for tenantat' objects. It triggers on 'A record is updated' with one condition: 'check for payment' equals 'Paid'. The flow then runs immediately and performs an 'send email' action. Finally, it ends.

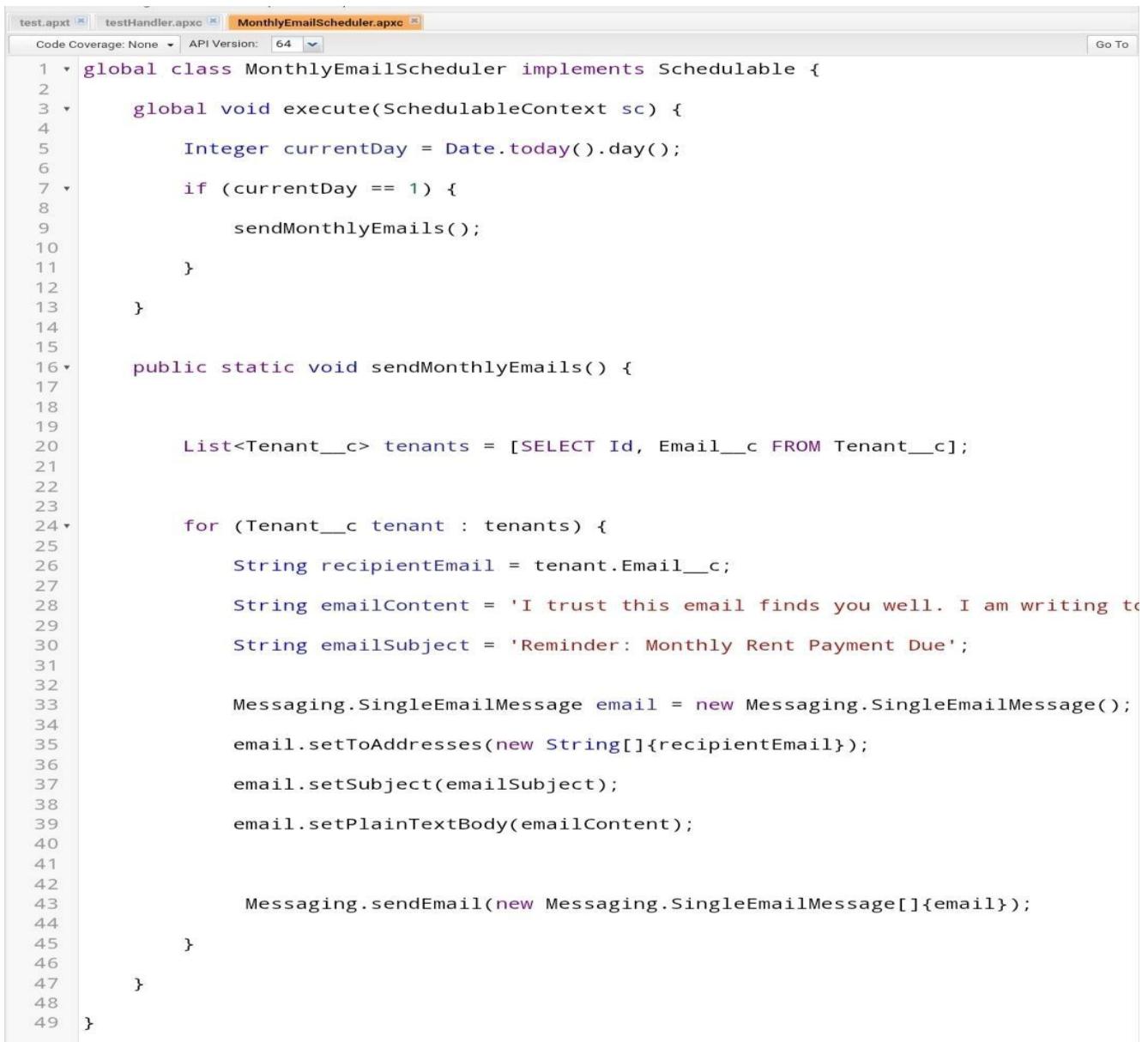
Configure Start

- Select Object:** Payment for tenantat
- Trigger the Flow When:**
 - A record is created
 - A record is updated
 - A record is created or updated
 - A record is deleted
- Set Entry Conditions:** All Conditions Are Met (AND)
- Condition Requirements:** check for payment Equals A Paid

The screenshot shows the same flow configuration with additional settings. Under 'When to Run the Flow for Updated Records', the option 'Every time a record is updated and meets the condition requirements' is selected. In the 'Optimize Flow' section, 'Fast Field Updates' is selected. The 'Actions and Related Records' section is expanded, stating: 'Update any record and perform actions, like send an email. This more flexible flow runs after the record is saved to the database.' At the bottom, there is a note about external callouts and an 'Add Asynchronous Path' toggle switch.

Schedule class

- Create an Apex Class



The screenshot shows the Salesforce code editor interface. The tab bar at the top has three tabs: 'test.apxt' (closed), 'testHandler.apxc' (closed), and 'MonthlyEmailScheduler.apxc' (selected). The status bar at the bottom shows 'Code Coverage: None' and 'API Version: 64'. A 'Go To' button is also visible. The main code area contains the following Apex class:

```
1  global class MonthlyEmailScheduler implements Schedulable {
2
3      global void execute(SchedulableContext sc) {
4
5          Integer currentDay = Date.today().day();
6
7          if (currentDay == 1) {
8
9              sendMonthlyEmails();
10
11         }
12     }
13
14
15
16     public static void sendMonthlyEmails() {
17
18
19
20         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
21
22
23
24         for (Tenant__c tenant : tenants) {
25
26             String recipientEmail = tenant.Email__c;
27
28             String emailContent = 'I trust this email finds you well. I am writing to';
29
30             String emailSubject = 'Reminder: Monthly Rent Payment Due';
31
32
33             Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
34
35             email.setToAddresses(new String[]{recipientEmail});
36
37             email.setSubject(emailSubject);
38
39             email.setPlainTextBody(emailContent);
40
41
42             Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
43
44         }
45
46     }
47
48 }
49 }
```

Schedule Apex class

The screenshot shows the Salesforce Apex Classes page. At the top, there's a navigation bar with Home, Chatter, Campaigns, Leads, Accounts, Contacts, Opportunities, Forecasts, Contracts, Orders, Cases, Solutions, and a search bar. Below the navigation is a sidebar titled "Salesforce Mobile Quick Start" with sections for Home, Administer, Build, and Customize.

The main content area is titled "Apex Classes". It displays a message: "Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform." Below this is a yellow box containing a green checkmark icon and the text "Percent of Apex Used: 0.03%" followed by a detailed description of character usage.

Under "Estimate your organization's code coverage", there are links for "Compile all classes" and "View: All". A table lists two Apex classes: "MonthlyEmailScheduler" and "testHandler".

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	MonthlyEmailScheduler		64.0	Active	1,125	Krishna Kumar B. 9/2/2025, 10:13 AM	<input type="checkbox"/>
Edit Del Security	testHandler		64.0	Active	584	Krishna Kumar B. 9/2/2025, 5:25 AM	<input type="checkbox"/>

Below this is a section titled "Dynamic Apex Classes" with a message: "Dynamic Apex extends your programming reach by interacting with Lightning Platform components." It shows a table for dynamic apex classes with one record: "No records to display".

This screenshot shows the "Apex Class Detail" page for the "MonthlyEmailScheduler" class. The top navigation bar and sidebar are identical to the previous screenshot.

The main content area shows the class detail for "MonthlyEmailScheduler". It includes tabs for "Class Body", "Class Summary", "Version Settings", and "Trace Flags". The "Class Body" tab displays the Apex code:

```

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

```

The "Class Summary" tab shows the class details: Name (MonthlyEmailScheduler), Namespace Prefix (empty), Created By (Krishna Kumar R.), Status (Active), Code Coverage (0% / 0/15), and Last Modified By (Krishna Kumar R. - 9/2/2025, 10:13 AM).

- Testing the approval process

Tenant Krishna

Related Details

Tenant Name	Krishna	Owner	Krishna Kumar R.
Email	krishnakumarrm3058@gmail.com		
Phone	(730) 632-6064		
status	Stay		
property	Hyd		
Created By	Krishna Kumar R.	Last Modified By	Krishna Kumar R.
	9/2/2025, 5:19 AM		9/10/2025, 8:22 AM

Activity

New Case

New Lead
Delete
Clone
Change Owner
Printable View
Sharing
Sharing Hierarchy
Submit for Approval
Edit Labels

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.

Tenant Krishna

Related Details

Tenant Name	Krishna	Owner	Krishna Kumar R.
Email	krishnakumarrm3058@gmail.com		
Phone	(730) 632-6064		
status	Stay		
property	Hyd		
Created By	Krishna Kumar R.	Last Modified By	Krishna Kumar R.
	9/2/2025, 5:19 AM		9/10/2025, 8:22 AM

Activity

Filters: All time • All activities • All types

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.

Submit for Approval

Comments
Leaving

Cancel Submit

Tenant Krishna

Details

- Tenant Name: Krishna
- Email: krishnakumarrm3058@gmail.com
- Phone: (730) 632-6064
- Status: Stay
- Property: Hyd
- Created By: Krishna Kumar R.
- Last Modified By: Krishna Kumar R., 9/10/2025, 8:22 AM

Activity

No activities to show. Get started by sending an email, scheduling a task, and more.

Notifications

- Krishna Kumar R is requesting approval for tenant
Tenant Name: Krishna • Owner: Krishna Kumar R
a few seconds ago
- Approval request for the tenant is approved
Krishna
Sep 9, 2025, 3:19 PM
- Approval request for the tenant is approved
Krishna
Sep 9, 2025, 11:51 AM
- New Guidance Center learning resource available
Define Your Sales Process
Learn how to guide reps through the sales process.
Sep 2, 2025, 5:41 PM
- New Guidance Center learning resource available
Set Up Accounts & Contacts
Start storing information about your customers with accounts and contacts.
Sep 2, 2025, 5:41 PM

Step Name	Date	Status	Assigned To
Approved	9/10/2025, ...	Approved	Krishna Ku...
Approval ...	9/10/2025, ...	Submitted	Krishna Ku...
Approved	9/9/2025, 2:...	Approved	Krishna Ku...
Approval ...	9/9/2025, 2:...	Submitted	Krishna Ku...
Approved	9/8/2025, 1:...	Approved	Krishna Ku...
Approval ...	9/8/2025, 8:...	Submitted	Krishna Ku...

RESULTS

Output Screenshots

Tabs for Property, Tenant, Lease, Payment

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object Tabs

Action	Label	Tab Style	Description
Edit Del	lease	Books	
Edit Del	Payment	Books	
Edit Del	properties	Bottle	
Edit Del	Tenants	Books	

Web Tabs

No Web Tabs have been defined

Visualforce Tabs

No Visualforce Tabs have been defined

Lightning Component Tabs

No Lightning component tabs have been defined

Lightning Page Tabs

No Lightning Page Tabs have been defined

Approval History

6 items • Sorted by Is Pending • Updated a few seconds ago

Step Name	Date	Status	Assignee	Actual Approver	Comments
1 Approved	9/10/2025, 9:04 AM	Approved	Krishna Ku...	Krishna Kumar R	Approved
2 Approval Re...	9/10/2025, 9:03 AM	Submitted	Krishna Ku...	Krishna Kumar R	Leaving
3 Approved	9/9/2025, 2:49 AM	Approved	Krishna Ku...	Krishna Kumar R	Approve tenant
4 Approval Re...	9/9/2025, 2:44 AM	Submitted	Krishna Ku...	Krishna Kumar R	Leaving
5 Approved	9/8/2025, 11:21 PM	Approved	Krishna Ku...	Krishna Kumar R	Leaving
6 Approval Re...	9/8/2025, 8:39 PM	Submitted	Krishna Ku...	Krishna Kumar R	Leaving

ADVANTAGES & DISADVANTAGES

Advantages of Lease Management System

Centralized Data Management – All tenant, property, and lease records are stored in one place, reducing duplication and errors.

Automation of Processes – Workflows, approvals, and reminders minimize manual intervention, saving time and effort.

Improved Accuracy – Automated tracking of payments and contracts reduces chances of mistakes and missed deadlines.

Better Communication – Email alerts and notifications keep tenants and property managers informed in real time.

Enhanced Compliance – Role-based access ensures data security and adherence to legal requirements.

Data Analytics & Reporting – Dashboards provide insights into occupancy rates, payment status, and lease renewals.

Scalability – Can be extended to manage multiple properties and tenants as the business grows.

Tenant Satisfaction – Timely updates and smoother processes improve the overall tenant experience.

Disadvantages of Lease Management System

High Initial Setup Cost – Implementing Salesforce and customizing it for lease management may require investment.

Training Requirement – Staff need to be trained to use Salesforce effectively, which can take time

Dependency on Internet – Being a cloud-based solution, it requires stable internet connectivity.

Customization Complexity – Advanced customizations may need skilled Salesforce developers, adding to costs.

Data Migration Challenges – Transferring existing lease and tenant records into the system can be time-consuming.

Subscription Costs – Ongoing Salesforce licensing fees may be expensive for smaller organizations

CONCLUSION

The Lease Management System streamlines leasing operations by centralizing tenant data, contracts, payments, and communication within Salesforce. It enhances accuracy, reduces manual effort, and ensures timely rent collection and contract renewals through automation. With features like Flows, Approval Processes, and Email Alerts, the system improves

efficiency and communication, while dashboards provide real-time insights for decision-making. Overall, it boosts productivity, tenant satisfaction, and compliance, offering a scalable foundation for future growth.

APPENDIX

Source Code: Provided in Apex Classes and Triggers

Test.apxt:

```
trigger test on Tenant c (before insert) { if (trigger.isInsert && trigger.isBefore){  
testHandler.preventInsert(trigger.new);  
} } testHandler.apxc: public class testHandler { public static  
void preventInsert(List<  
Tenant c> newlist)  
{Set<Id>existingPropertyIds= new Set<Id>() for (Tenant c existingTenant : [SELECT Id,  
Property c FROM Tenant c WHERE Property c  
!= null]) {existingPropertyIds.add(existingTenant.Property c;  
} for (Tenant c newTenant :newlist) { if (newTenant.Property c != null &&  
existingPropertyIds.contains(newTenant.Property c)) { newTenantaddError('A tenant can  
have only one property');  
}}}}
```

MothlyEmailScheduler.apxc:

```

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});

}
}
}

```