

LEASE MANAGEMENT

COLLEGE NAME : SREE NARAYANA GURU COLLEGE

COLLEGE CODE : Bru36

TEAM ID: NM2025TMID26405

TEAM MEMBERS:

Team LeaderName : Krishna Kumar R

Email : krishnakumarrm3058@gmail.com

Team Member 1 : Antro Infant J

Email : antroinfant103@gmail.com

Team Member 2 : Anfal Dharvesh K

Email : anfalanfu2727@gmail.com

Team Member 3 : Anshil Rosario S

Email : Anshilrosario2005@gmail.com

Team Member 4 : Anas Muhammad M

Email : anasanasmuhammed01@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.



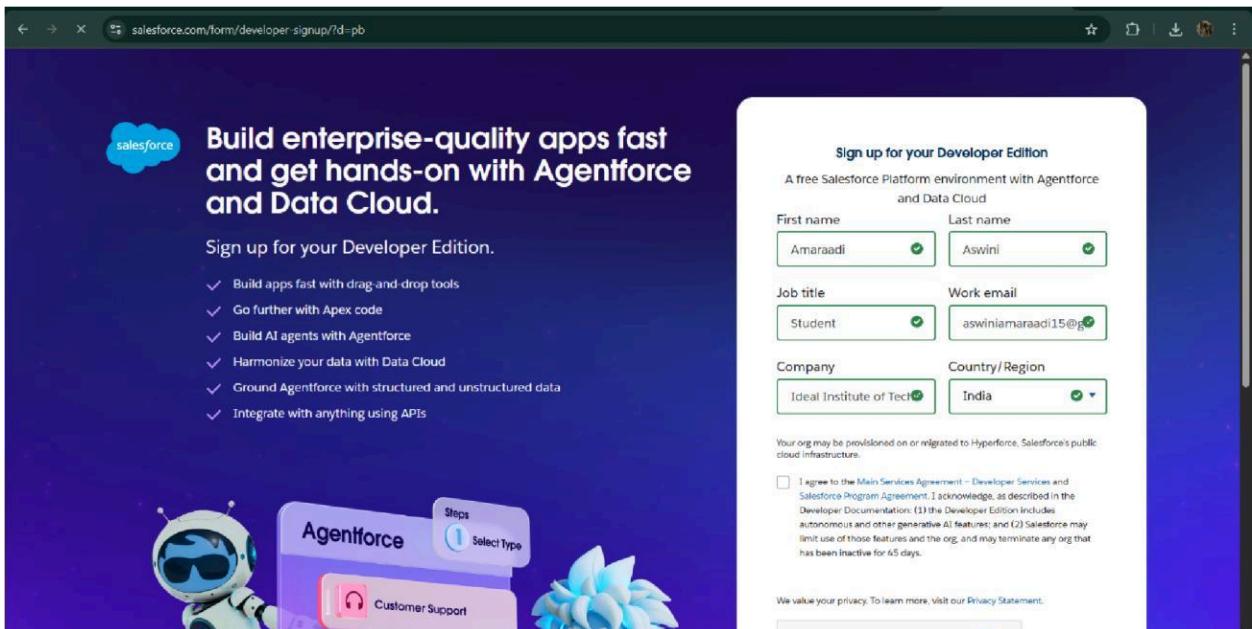
1.2 Purpose

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

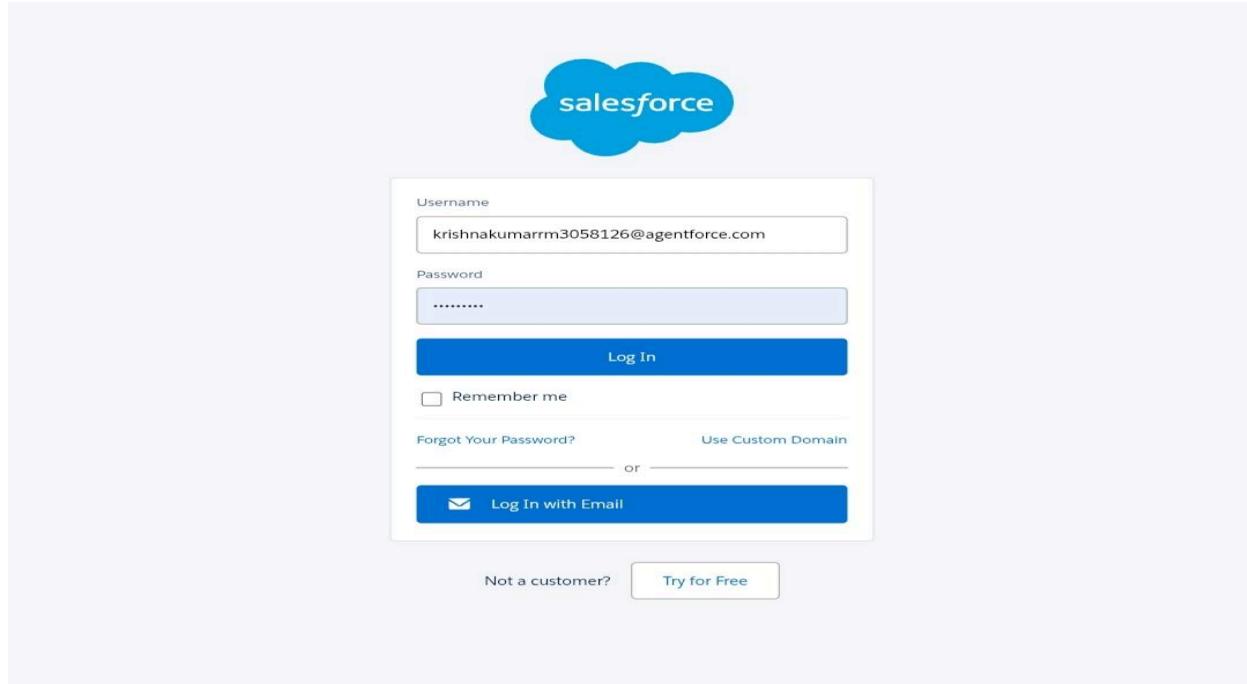
- **Creating Developer Account:**

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 > Object Manager interface. The page title is 'SETUP > OBJECT MANAGER' and the object name is 'Tenant'. On the left, there's a sidebar with various configuration options like Fields & Relationships, Page Layouts, and Record Types. The main area shows the 'Details' tab for the 'Tenant' object. It contains sections for 'Description' (with API Name set to 'Tenant__c'), 'Custom' (with Singular Label 'Tenant' and Plural Label 'Tenants'), and 'Enable Reports' (with checkboxes for Track Activities, Track Field History, Deployment Status, and Help Settings). At the bottom right of the main area are 'Edit' and 'Delete' buttons. The status bar at the bottom shows system information including weather (29°C, Sunny), quick search, and system status (ENG IN 06:45 08-09-2025).

The screenshot shows the Salesforce Setup interface for managing objects. The top navigation bar includes tabs for WhatsApp, Developer Edition with Agents, Home | Salesforce, lease | Salesforce, and New Tab. The main title is "SETUP > OBJECT MANAGER". The object being edited is "lease".

Details tab selected:

- Description: [empty]
- API Name: **lease__c**
- Type: **Custom**
- Singular Label: **lease**
- Plural Label: **lease**

Details section:

- Enable Reports:
- Track Activities:
- Track Field History:
- Deployment Status: **Deployed**
- Help Settings: Standard salesforce.com Help Window

Left sidebar:

- 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
- Scoping Rules

System status bar at the bottom:

- 29°C Sunny
- Quick search
- Cloud icon
- File icon
- Print icon
- Help icon
- Logout icon
- ENG IN
- 06:46
- 08-09-2025

The screenshot shows the Salesforce Setup interface for managing objects. The top navigation bar includes tabs for WhatsApp, Developer Edition with Agents, Home | Salesforce, Payment for tenant | Salesforce, and New Tab. The main title is "SETUP > OBJECT MANAGER". The object being edited is "Payment for tenant".

Details tab selected:

- Description: [empty]
- API Name: **Payment_for_tenant__c**
- Type: **Custom**
- Singular Label: **Payment for tenant**
- Plural Label: **Payment**

Details section:

- Enable Reports:
- Track Activities:
- Track Field History:
- Deployment Status: **Deployed**
- Help Settings: Standard salesforce.com Help Window

Left sidebar:

- 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
- Scoping Rules

System status bar at the bottom:

- 29°C Sunny
- Quick search
- Cloud icon
- File icon
- Print icon
- Help icon
- Logout icon
- ENG IN
- 06:47
- 08-09-2025

- Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main content area displays a table titled 'Fields & Relationships' with 9 items. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(32768)		
Created By	CreatedBy/d	Lookup(User)		
Last Modified By	LastModifiedBy/d	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 Object Manager interface for the 'Payment for tenant' object. The left sidebar lists various setup options. The main content area displays a table titled 'Fields & Relationships' with 8 items. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Number(18, 0)		
check for payment	check_for_payment_c	Picklist		
Created By	CreatedBy/d	Lookup(User)		
Last Modified By	LastModifiedBy/d	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)	✓	
Payment date	Payment_date_c	Date		
Payment Name	Name	Text(80)	✓	
Tenant	Tenant__c	Lookup(Tenant)	✓	

Screenshot of the Salesforce Setup interface showing the Fields & Relationships for the 'lease' object.

Fields & Relationships (7 items, Sorted by Field Label)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End date	End_date__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓
property	property__c	Lookup(property)		✓
start date	start_date__c	Date		

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
- List View Button Layout
- Restriction Rules
- Scoping Rules

Screenshot of the Salesforce Setup interface showing the Fields & Relationships for the 'Tenant' object.

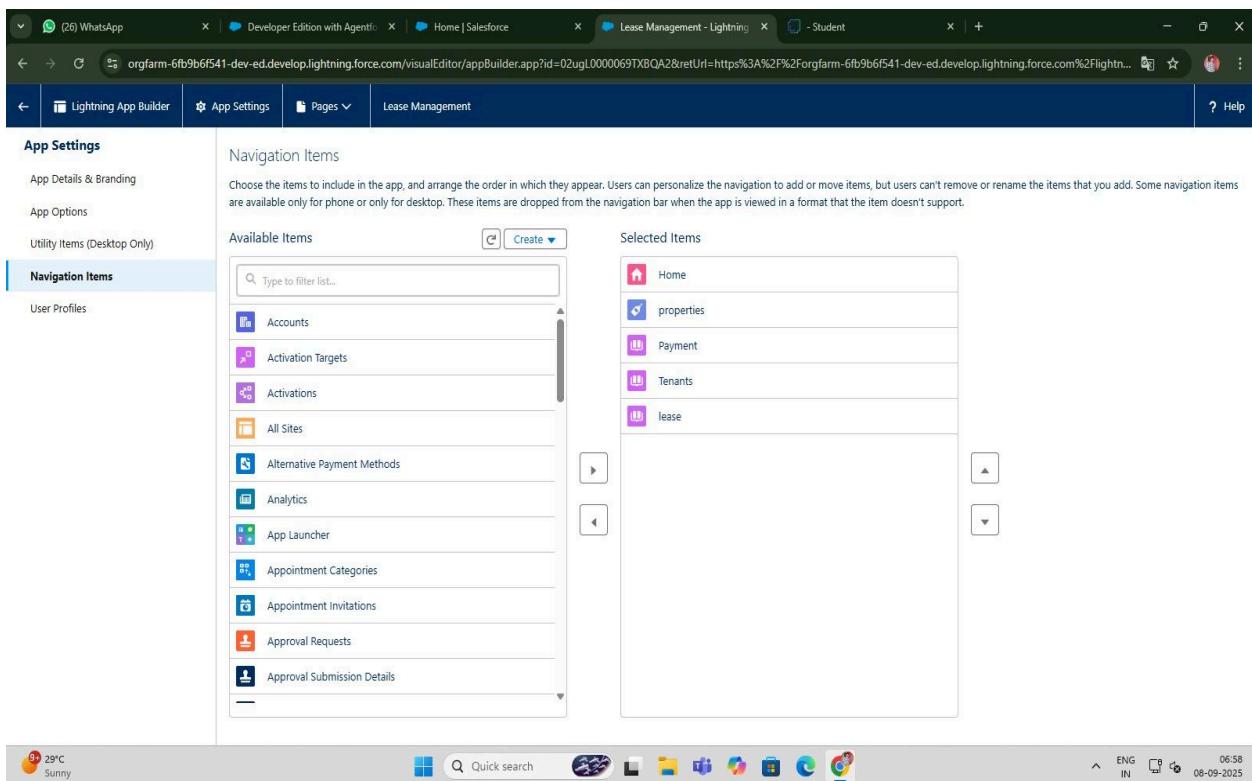
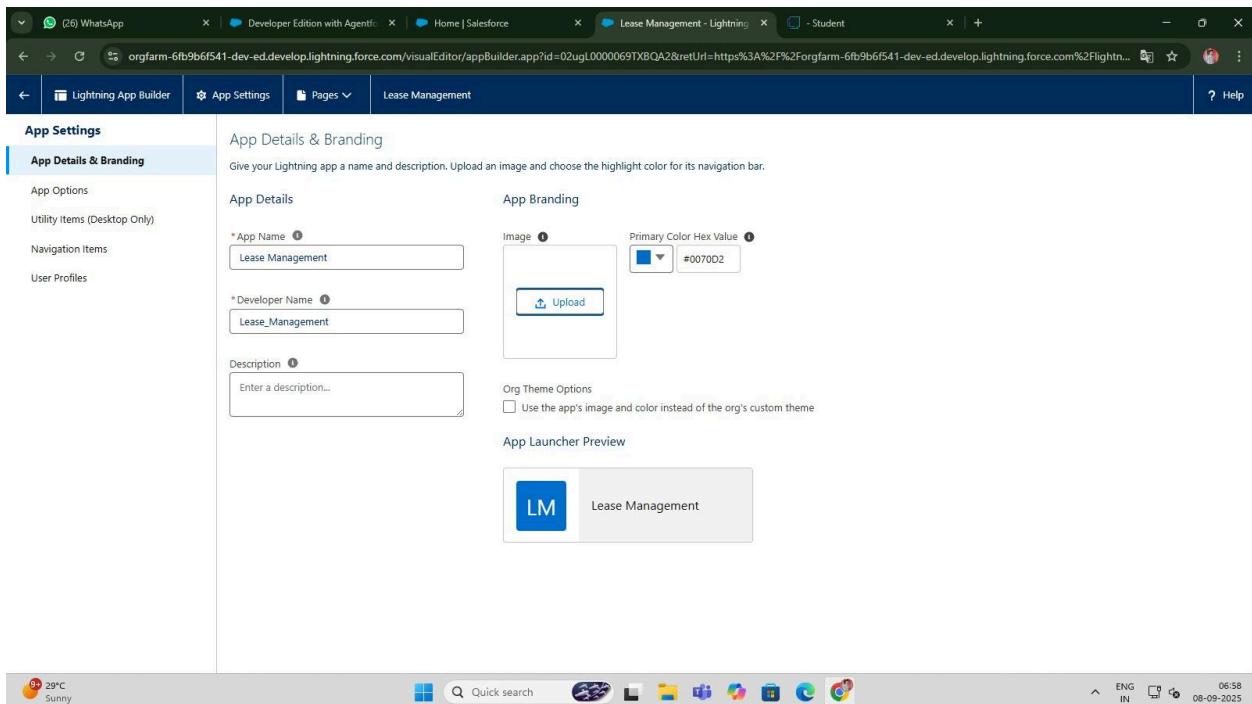
Fields & Relationships (8 items, Sorted by Field Label)

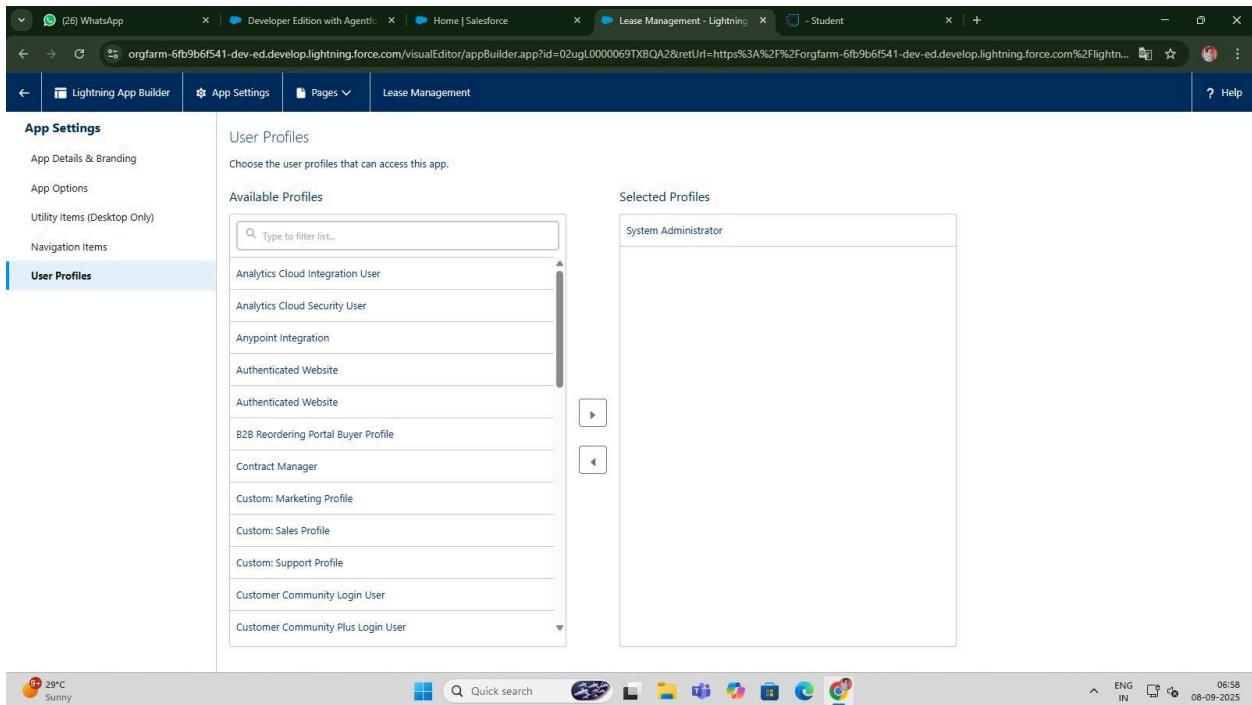
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Email	Email__c	Email		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		
property	property__c	Lookup(property)		✓
status	status__c	Picklist		
Tenant Name	Name	Text(80)		✓

Details

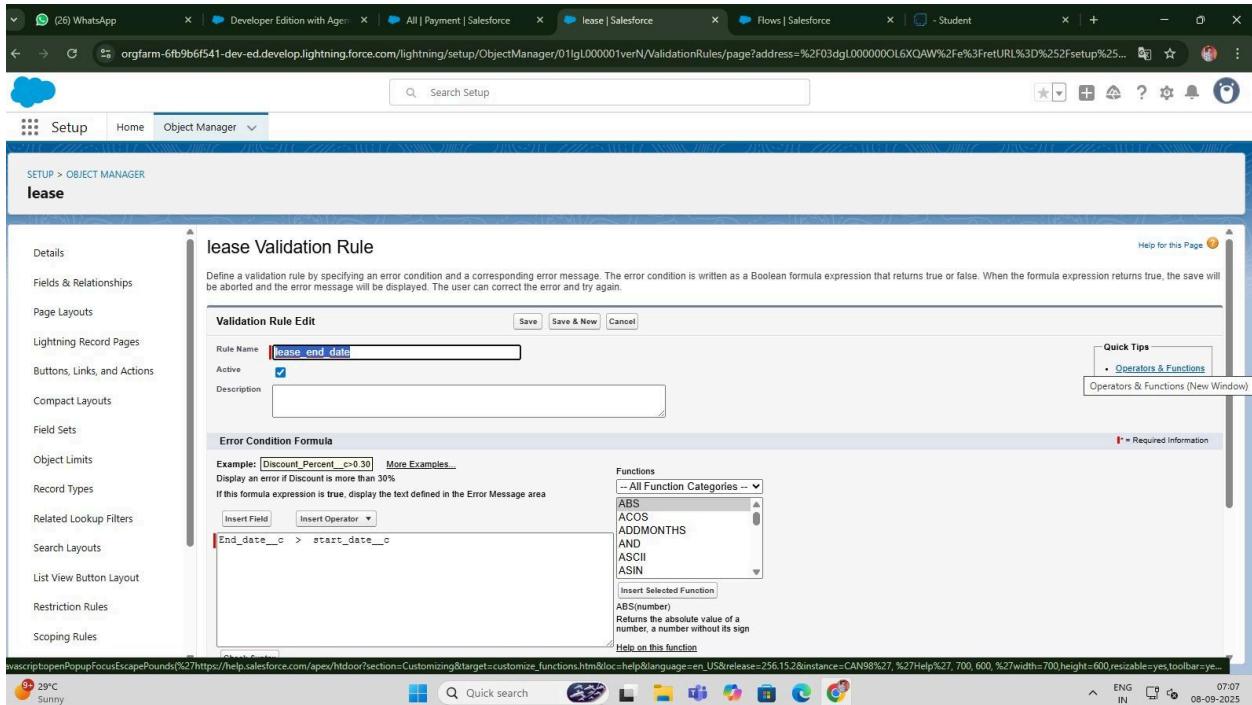
- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Restriction Rules
- Scoping Rules

- Developed Lightning App with relevant tabs





- create a validation rule to an Lease Object



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

The screenshot shows the Salesforce Setup interface with the following details:

Page Title: Classic Email Templates

Email Template Name: tenant leaving

Description: Preview your email template below.

Email Template Detail:

Setting	Value
Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant leaving
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	Krishna Kumar R [Change]
Description	
Created By	Krishna Kumar R 9/2/2025, 3:13 AM
Modified By	Krishna Kumar R 9/2/2025, 3:13 AM

Email Template:

Subject: request for approve the leave

Plain Text Preview:

```
Dear {!Tenant__c.CreatedBy}.

Please approve my leave
```

The screenshot shows the Salesforce Setup interface with the following details:

Page Title: Classic Email Templates

Email Template Name: Leave approved

Description: Preview your email template below.

Email Template Detail:

Setting	Value
Email Templates from Salesforce	Unfiled 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
```

Screenshot of the Salesforce Classic Email Templates page for the 'Leave rejected' template.

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
```

Screenshot of the Salesforce Classic Email Templates page for the 'Tenant Email' template.

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
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

Email Template

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 screenshot shows the 'Classic Email Templates' page in the Salesforce Setup. A new template named 'tenant payment' has been created. The template details include:

- 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

The email template content is:

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 'Approval Processes' page in the Salesforce Setup. A new approval process named 'check for vacant' is being created. The process details are:

Step 1. Enter Name and Description

Enter a name and description for your new approval process.

Process Name	check for vacant	Required Information
Unique Name	check_for_vacant	
Description	(empty)	

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page selected. The process 'Tenant: check for vacant' is listed. The 'Process Definition Detail' section shows the process name, unique name, entry criteria ('Tenant: status NOT EQUAL TO Leaving'), and record editability ('Administrator ONLY'). The 'Initial Submission Actions' section contains a single entry: 'Record Lock' with a description 'Lock the record from being edited' and a note 'please approve my leave'. The 'Approval Steps' section is empty, showing a message 'You have not yet defined any approval steps'. A context menu is open over the 'Initial Submission Actions' section, listing options: Task, Email Alert, Field Update, Outbound Message, and Flow.

● Initial Submission Action

The screenshot shows the Salesforce classic interface with the 'Approval Processes' page selected. The process 'Tenant: check for vacant' is listed. The 'Process Definition Detail' section shows the process name, unique name, entry criteria ('Tenant: status NOT EQUAL TO Leaving'), and record editability ('Administrator ONLY'). The 'Initial Submission Actions' section contains a single entry: 'Record Lock' with a description 'Lock the record from being edited' and a note 'please approve my leave'. The 'Approval Steps' section is empty, showing a message 'You have not yet defined any approval steps'. A context menu is open over the 'Initial Submission Actions' section, listing options: Task, Email Alert, Field Update, Outbound Message, and Flow.

Email Alert Detail

please approve my leave

Description: please approve my leave
Unique Name: please_approve_my_leave
From Email Address: Current User's email address
Recipients: Email Field: Email
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: check for vacant
Description: Tenant
Type: Tenant
State: 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

Always show me **more records per related list**

- Final Approval Action

Email Alert Detail

Tenant leaving

Description: Tenant leaving
Unique Name: Tenant_leaving
From Email Address: Current User's email address
Recipients: Email Field: Email
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

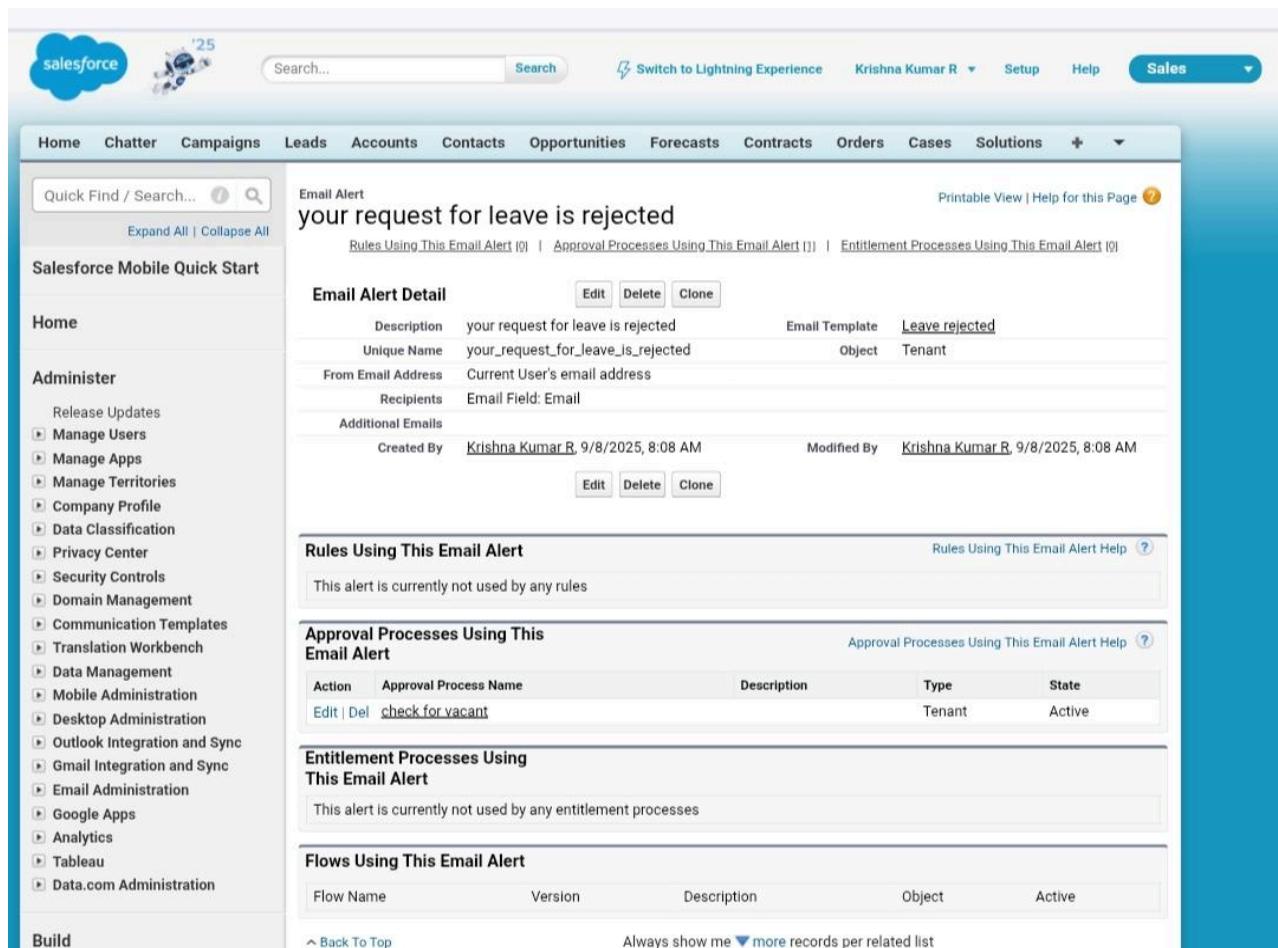
Approval Processes Using This Email Alert
Action: Approval Process Name: check for vacant
Description: Tenant
Type: Tenant
State: Active

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

Always show me **more records per related list**

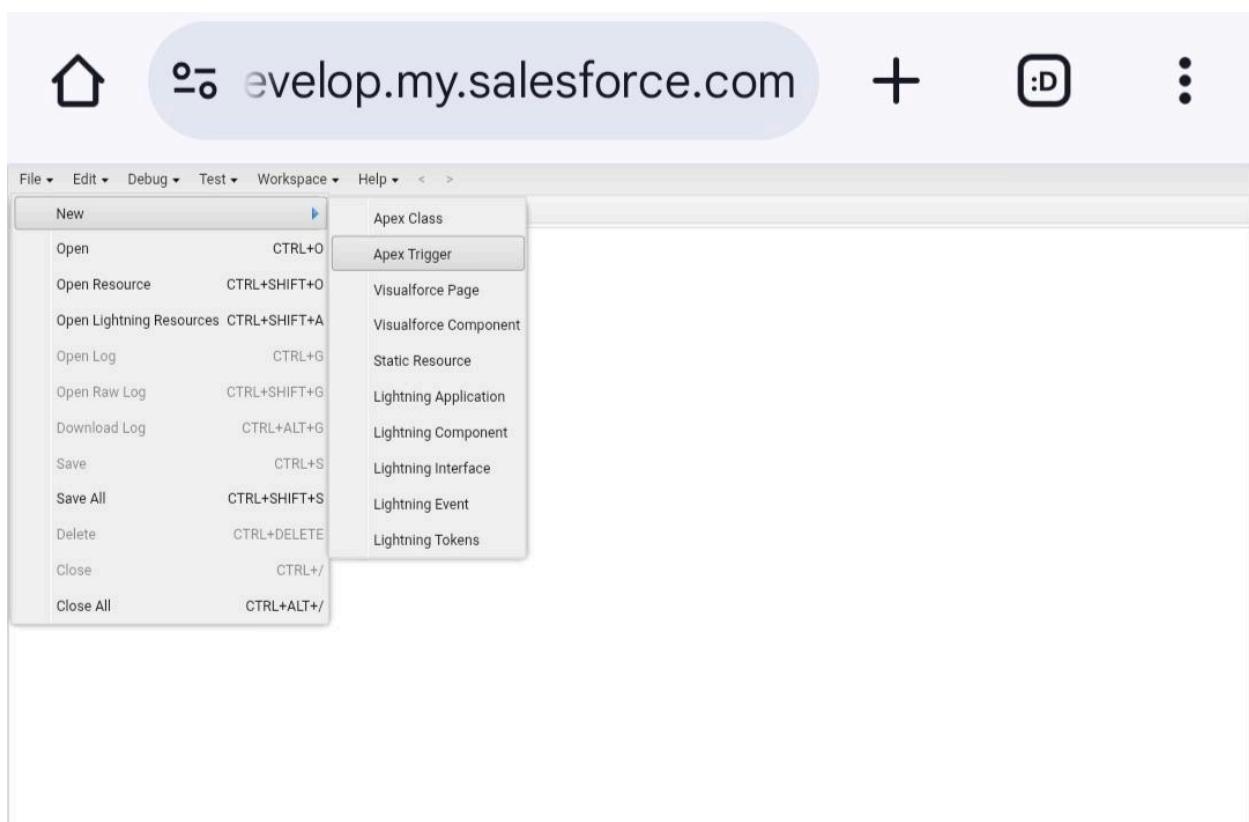
- Final Rejection Action

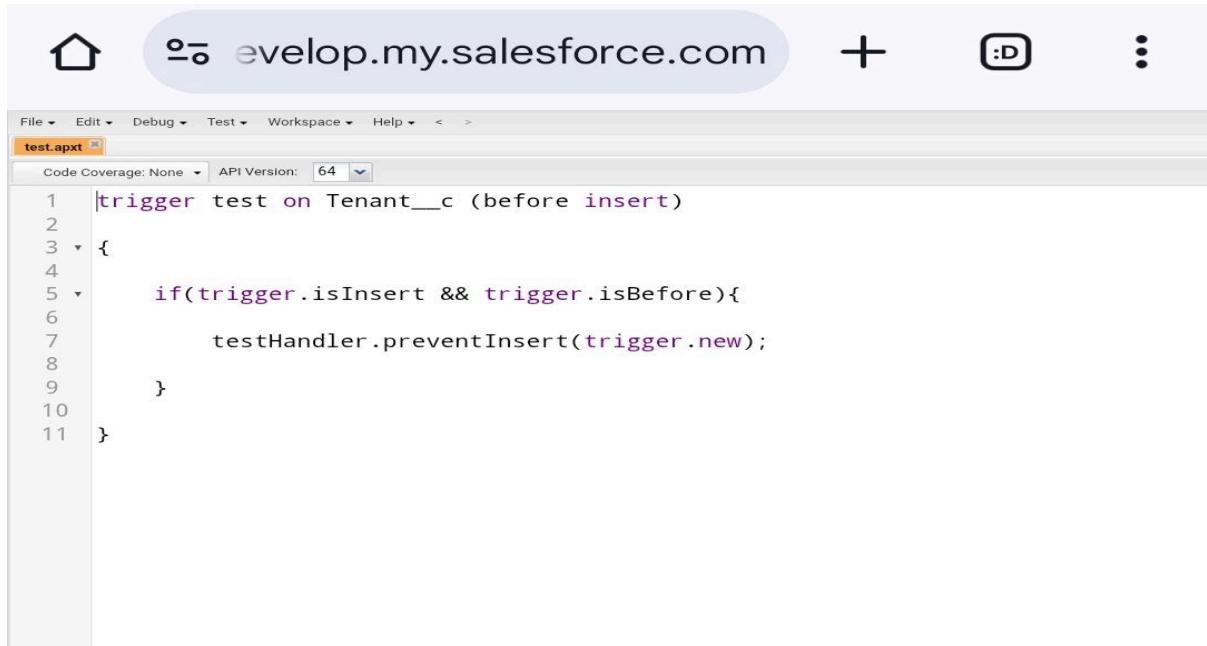


The screenshot shows the Salesforce Email Alert detail page. The main title is "your request for leave is rejected". The "Email Alert Detail" section includes fields for Description (your request for leave is rejected), Unique Name (your_request_for_leave_is_rejected), From Email Address (Current User's email address), Recipients (Email Field: Email), and Created By (Krishna Kumar R. 9/8/2025, 8:08 AM). The "Email Template" is set to "Leave rejected". Below this, there are sections for "Rules Using This Email Alert", "Approval Processes Using This Email Alert", "Entitlement Processes Using This Email Alert", and "Flows Using This Email Alert". The "Approval Processes Using This Email Alert" section shows one entry: "check_for_vacant" with Type "Tenant" and State "Active". The "Entitlement Processes Using This Email Alert" section and "Flows Using This Email Alert" section both show a message: "This alert is currently not used by any entitlement processes" and "Always show me ▾ more records per related list" respectively.

Apex Trigger

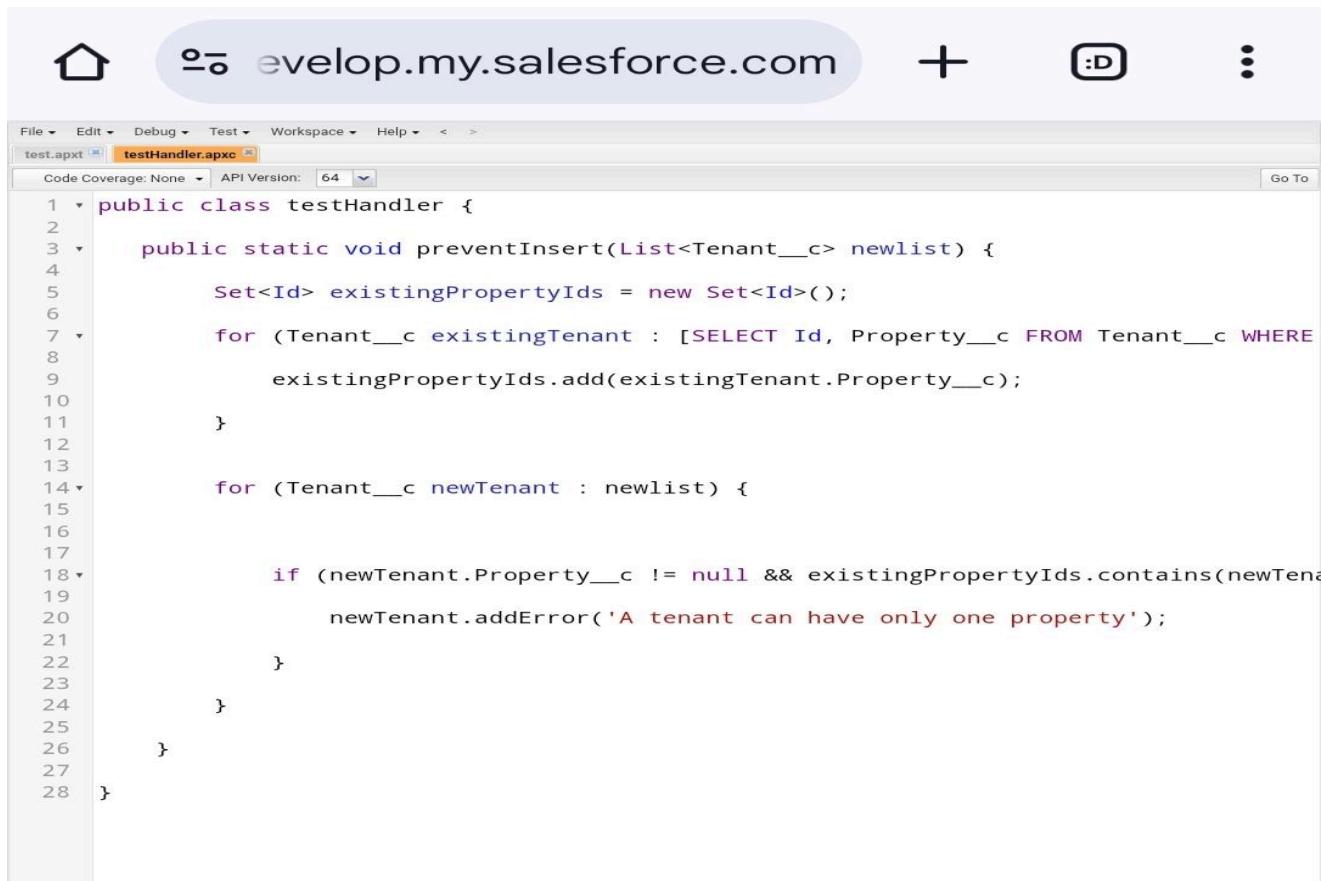
- Create an Apex Trigger





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

- Create an Apex Handler class



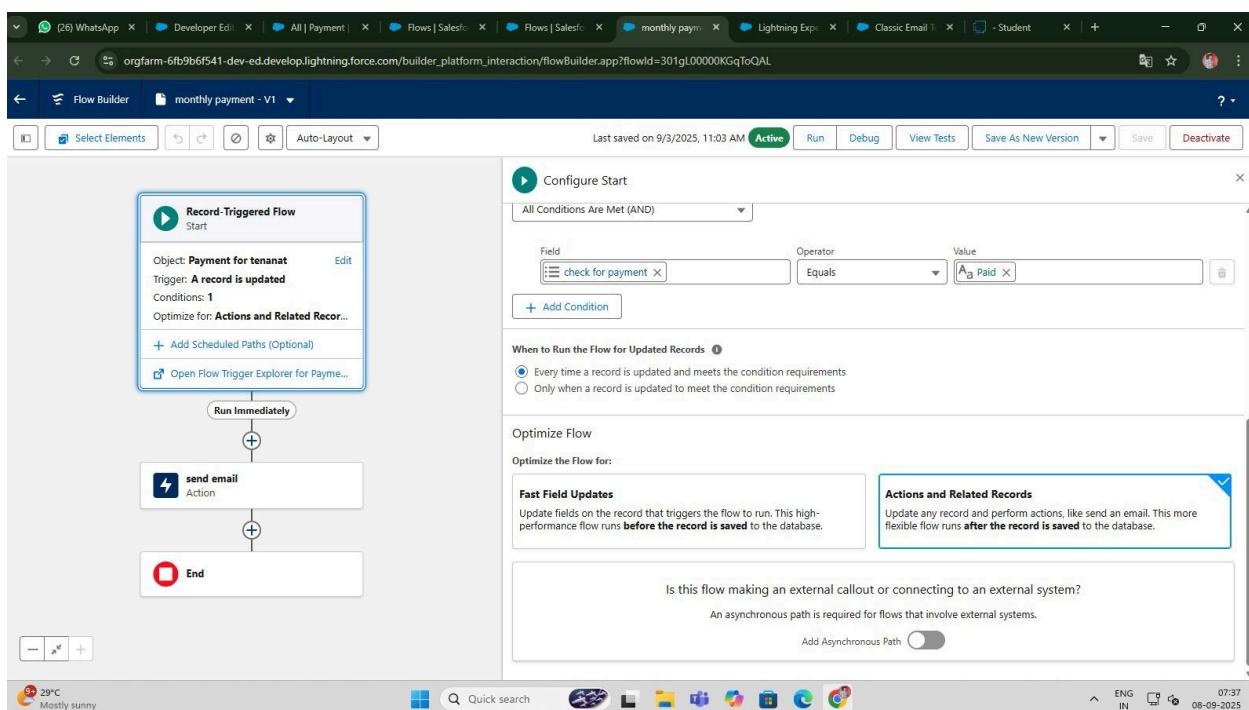
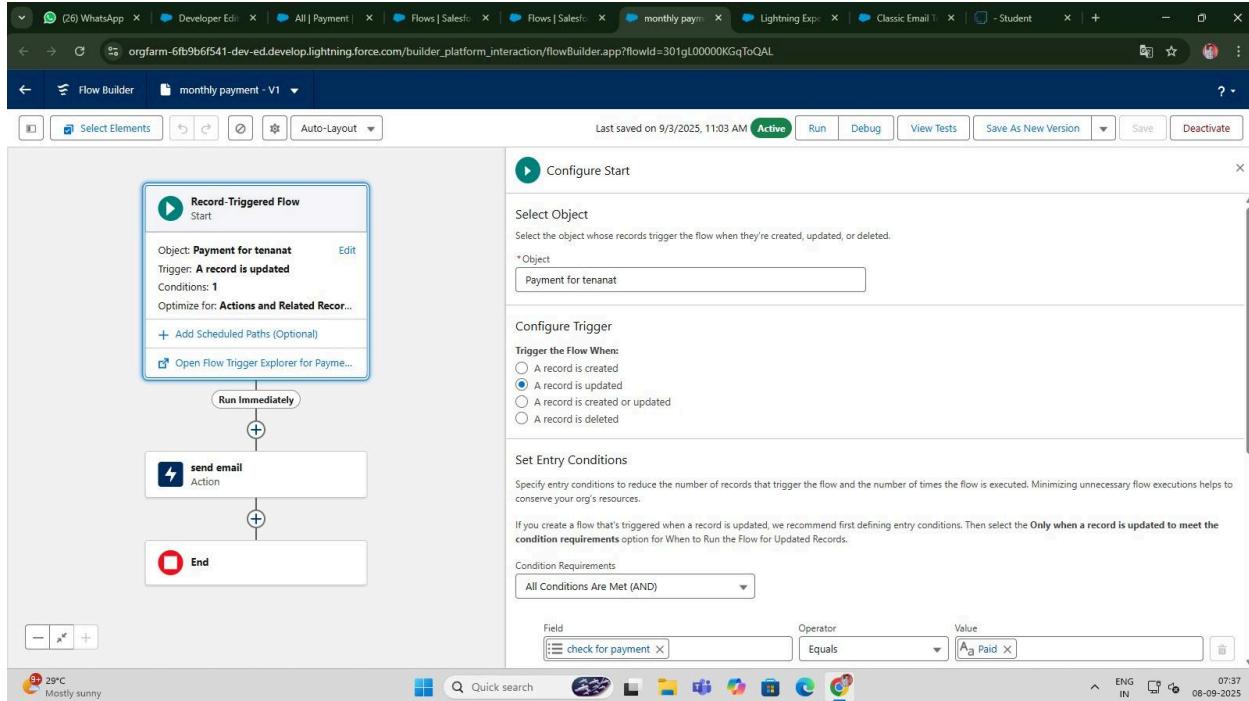
```
public class testHandler {
    public static void preventInsert(List<Tenant__c> newlist) {
        Set<Id> existingPropertyIds = new Set<Id>();
        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE
                                         existingPropertyIds.add(existingTenant.Property__c);
        }
        for (Tenant__c newTenant : newlist) {
            if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c))
                newTenant.addError('A tenant can have only one property');
        }
    }
}
```

- Testing the Trigger

The screenshot shows a software application window. At the top left, there's a header with 'Stay', 'property', and 'Hyd'. Below it, 'Created By' is listed as 'Krishna Kumar R.' with a creation date of '9/2/2025, 5:19 AM'. To the right, 'Last Modified By' is also 'Krishna Kumar R.' with a modification date of '9/10/2025, 8:22 AM'. A note says 'No past activity. Past meetings and tasks marked as done show up here.' On the far right is a close button ('X').

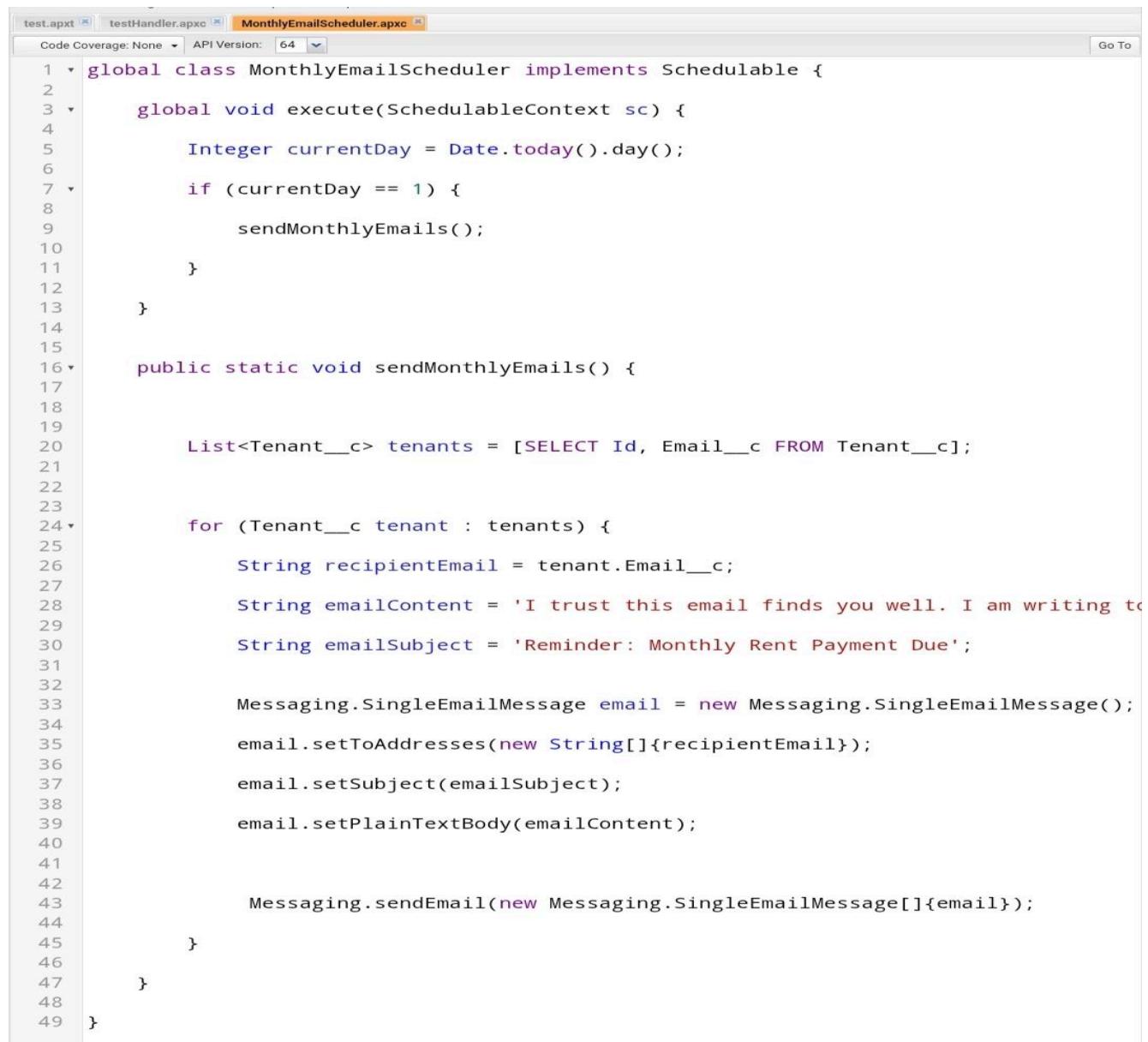
The main area is a modal titled 'New Tenant'. It has a note at the top right: '* = Required Information'. The 'Information' section contains fields for 'Tenant Name' (set to 'Krishna Kumar R.'), 'Email' (set to 'krishnakumarrm3058@gmail.com'), 'Phone' (empty), and 'status' (set to 'Stay'). To the right of these fields is an 'Owner' section showing 'Krishna Kumar R.'. A red error message box is displayed, containing a red circle with a slash icon, the text 'We hit a snag.', and a link 'Review the errors on this page.' with the note '• A tenant can have only one property'. At the bottom are buttons for 'Cancel', 'Save & New' (disabled), and 'Save'.

Create Flow for monthly payment



Schedule class

- Create an Apex Class



The screenshot shows the Salesforce IDE interface with the code editor open. The tab bar at the top has three tabs: 'test.apxt' (grayed out), 'testHandler.apxc' (grayed out), and 'MonthlyEmailScheduler.apxc' (highlighted in orange). Below the tabs, there are dropdown menus for 'Code Coverage: None' and 'API Version: 64'. A 'Go To' button is on the right. The code editor displays 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
43             Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
44
45         }
46
47     }
48
49 }
```

Schedule Apex class

The screenshot shows the Salesforce Apex Classes page. At the top, there's a green box indicating "Percent of Apex Used: 0.03%" and "You are currently using 1,856 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." Below this, there are buttons to "Estimate your organization's code coverage" and "Compile all classes". The main list of Apex classes is shown in a table:

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 the table, there's a section titled "Dynamic Apex Classes" with the sub-section "MonthlyEmailScheduler". It shows the Apex class detail with the following code:

```

Apex Class
MonthlyEmailScheduler
+ Back to List: Apex Classes

Apex Class Detail
Name: MonthlyEmailScheduler
Namespace Prefix: 
Created By: Krishna Kumar R - 9/2/2025, 10:10 AM
Status: Active
Code Coverage: 0% (0/15)
Last Modified By: Krishna Kumar R - 9/2/2025, 10:13 AM

Class Body
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     public static void sendMonthlyEmails() {
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.SingleEmailMessage[] emails = new Messaging.SingleEmailMessage[1];
19            emails[0] = email;
20            Messaging.sendEmail(emails);
21        }
22    }
23}
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

```

This screenshot shows the Apex Class Detail page for the "MonthlyEmailScheduler" class. The page includes the class header, namespace prefix, creation date, status, and code coverage information. The "Class Body" tab is selected, displaying the complete Apex code:

```

Apex Class
MonthlyEmailScheduler
+ Back to List: Apex Classes

Apex Class Detail
Name: MonthlyEmailScheduler
Namespace Prefix: 
Created By: Krishna Kumar R - 9/2/2025, 10:10 AM
Status: Active
Code Coverage: 0% (0/15)
Last Modified By: Krishna Kumar R - 9/2/2025, 10:13 AM

Class Body
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     public static void sendMonthlyEmails() {
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.SingleEmailMessage[] emails = new Messaging.SingleEmailMessage[1];
19            emails[0] = email;
20            Messaging.sendEmail(emails);
21        }
22    }
23}
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

```

At the bottom of the page, there are buttons for "Edit", "Delete", "Download", "Security", and "Show Dependencies".

- Testing the approval process

The screenshot shows a CRM application for Lease Management. At the top, there's a navigation bar with icons for Home, properties, Payment, Tenants, and Lease. A search bar is also present. The main area displays a tenant profile for "Krishna". The "Details" tab is selected, showing information such as Tenant Name (Krishna), Email (krishnakumarrm3058@gmail.com), Phone ((730) 632-6064), status (Stay), property (Hyd), Created By (Krishna Kumar R.), and Last Modified By (Krishna Kumar R., 9/10/2025, 8:22 AM). To the right, an "Activity" section is shown with a sub-menu open for "New Case". The sub-menu includes options like New Lead, Delete, Clone, Change Owner, Printable View, Sharing, Sharing Hierarchy, Submit for Approval, and Edit Labels.

This screenshot shows the same CRM interface as above, but with a modal dialog box titled "Submit for Approval" overlaid. The dialog contains a text area labeled "Comments" with the text "Leaving" entered. At the bottom of the dialog are "Cancel" and "Submit" buttons. The background of the CRM interface is dimmed, indicating the dialog is active.

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., 9/2/2025, 5:19 AM

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

Activity

Filters: All time • All activities • All types

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.

Notifications

Mark all as read

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

The screenshot shows a web-based application interface for Lease Management. At the top, there is a header bar with various icons and a search bar. Below the header, the main navigation menu includes 'Lease Management' (highlighted), 'Home', 'properties', 'Payment', 'Tenants', 'lease', and a dropdown for 'Tenant_c'. A sub-menu for 'Tenant Approval' is open, showing a status of 'Pending'. On the left, there is a sidebar with sections for 'Details' and 'Approval Details'. In the 'Approval Details' section, it shows 'Submitter' as 'Krishna Kumar R', 'Date Submitted' as 'Sep 10, 2025', 'Actual Approver' as 'Krishna Kumar R', and 'Assigned To' as 'Krishna Kumar R'. To the right, there is a 'Submitter Comments' section showing a comment from 'Krishna Kumar R' stating 'Leaving' on 'Sep 10, 2025, 9:03:29 AM'. There are also 'Approve', 'Reject', and 'Reassign' buttons at the top right of the approval request area.

The screenshot shows a tenant profile page for 'Krishna'. The top navigation bar includes 'Approvals', 'Tenants', 'All Flow A...', 'Pending A...', and a 'New Contact' button. The main content area is divided into two main sections: 'Related' and 'Activity'. The 'Related' section contains tabs for 'Payments (0)', 'Payment (0)', and 'Approval History (6)'. The 'Approval History' table has columns for 'Step Name', 'Date', 'Status', and 'Assigned To'. The 'Activity' section includes a 'Filters: All t' button and a 'Upcoming & Overdue' section which states 'No activities' and 'Get started by sending an email'. It also mentions 'No past activity. Past meetings and ta'.

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

The screenshot shows the Salesforce Setup interface under the 'Sales' tab. On the left, there's a sidebar with sections like Home, Chatter, Campaigns, Leads, Accounts, Contacts, Opportunities, Forecasts, Contracts, Orders, Cases, Solutions, Administer, and Build. Under Administer, there's a long list of management options. Under Build, there are 'Customize' and 'Create' buttons. The main content area is titled 'Custom Tabs' and contains sections for 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tabs', 'Lightning Component Tabs', and 'Lightning Page Tabs'. Each section has a 'New' button and a 'What Is This?' link. In the 'Custom Object Tabs' section, there are four tabs listed: 'lease' (Tab Style: Books), 'Payment' (Tab Style: Books), 'properties' (Tab Style: Bottle), and 'Tenants' (Tab Style: Books). All other sections show 'No [Tab Type] have been defined'.

The screenshot shows a custom application titled 'Lease Management'. The top navigation bar includes links for Home, properties, Payment, Tenants, lease, *Tenant_c, and a search bar. Below the navigation is a header for 'Tenants > Krishna' with a sub-header 'Approval History'. A message indicates '6 items • Sorted by Is Pending • Updated a few seconds ago'. To the right are filter and sort icons. The main content is a table with the following data:

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)) {
newTenant.addError('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});
```

```
}
```

```
}
```

```
}
```