

LEASE MANAGEMENT

College Name: SREE VENKATESHWARA ARTS AND SCIENCE COLLEGE, GOBI(CO-EDUCATION)

CollegeCode: brubd

TEAM ID: NM2025TMID20898

TEAM MEMBERS:

Team LeaderName: K.ROOBAN SANJAY

Email: roobansanjayk2023cs@gmail.com

Team Member1: S.BALARAMAN

Email: balaramans2023cs@gmail.com

Team Member: S.SOORAJ

Email: soorajs2023cs@gmail.com

Team Member: L.PARTHASARATHY

Email: parthasarathy12023cs@gmail.com

1.INTRODUCTION

1.1ProjectOverview

TheLease ManagementSystem is aSalesforce-based application designed to streamline the processes associated with leasing real estate properties. It handles tenantmanagement, lease

contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.



1.2 Purpose

The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

DEVELOPMENT PHASE

Creating Developer Account:

The image shows a Salesforce sign-up page for the Developer Edition. The header features the Salesforce logo and the text: "Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud." Below this, a call-to-action button says "Sign up for your Developer Edition." To the right, there's a form for entering personal information: First name (rooban), Last name (sanjay), Job title (student), Work email (roobansanjayk2023), Company (svcas), Country/Region (India), and State/Province (Tamil Nadu). A note at the bottom states: "Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure." At the very bottom, there's a checkbox for agreeing to the Main Services Agreement and SalesForce Program Agreement.

● Created objects: Property, Tenant, Lease, Payment

This screenshot shows the Salesforce Object Manager interface. On the left, a sidebar lists various object settings like Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, and more. The main area displays the 'property' object's details. The 'Details' tab is selected, showing fields such as Description (with values like All Name, property_x, Custom, etc.), and a related list for Master Records (Task Activities, Task Record History, Deployment Status, Employee ID, and more). Buttons for Edit and Delete are visible at the top right of the detail pane.

The screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and a search bar. The left sidebar lists various configuration tabs: 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, Redirection Rules, Scoring Rules, and a blank tab at the bottom. The main content area displays the 'Details' tab for the 'Tenant' object. It contains sections for 'Description' (API Name: Tenant_c, Custom: Yes, Singular Label: Tenant, Plural Label: Tenants), 'Email Headers' (with a note about the standard header being used), and 'Email Templates' (with a note about the standard template being used). On the right side, there are 'Edit' and 'Delete' buttons.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes the Salesforce logo, a search bar, and standard navigation icons. Below the bar, the 'Setup' menu is open, with 'Object Manager' selected. The main content area is titled 'lease' under 'SETUP > OBJECT MANAGER'. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Component Layouts, Field Sets, Object Limits, Record Types, Related Lookup Fields, Search Layouts, List View Button Layout, and Record Action Rules. The 'Details' tab is currently active. The main panel displays the 'Details' section for the 'lease' object. It includes fields for 'Name' (set to 'lease'), 'Label' (set to 'lease'), and 'Type' (set to 'Standard'). A 'Description' field is present but empty. To the right, there are sections for 'Create Reports' (checkbox checked), 'Track Activities' (checkbox checked), 'Track Performance' (checkbox checked), 'Deployment Status' (checkbox checked), and 'Deployed' (checkbox checked). At the bottom right of the main panel, there are 'Edit' and 'Delete' buttons.

● Configured fields and relationships

The image displays two screenshots of the Salesforce Object Manager interface, specifically showing the 'Fields & Relationships' section for two different objects: 'property' and 'Payment for tenant'.

Property Object Fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(1000)		
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Name	Name__c	Text(255)		
Owner	Owner	Lookup(User,Group)		<input checked="" type="checkbox"/>
property	property__c	Lookup(property)		<input checked="" type="checkbox"/>
propertyName	Name	Text(255)		<input checked="" type="checkbox"/>
size	size__c	Text(10)		
Type	Type__c	Picklist		

Payment for tenant Object Fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Number(15, 2)		
Check for payment	check_for_payment__c	Picklist		
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	Owner	Lookup(User,Group)		<input checked="" type="checkbox"/>
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(255)		<input checked="" type="checkbox"/>

The screenshot shows the Salesforce Setup interface with the 'Object Manager' tab selected. In the top navigation bar, there are links for 'Setup', 'Home', and 'Object Manager'. A search bar at the top right contains the text 'Search Setup'. Below the navigation, the title 'SETUP > OBJECT MANAGER' and the object name 'lease' are displayed. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships (which is currently selected), 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 is titled 'Fields & Relationships' and shows a table with 7 items, sorted by Field Label. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data in the table is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End Date	EndDate_c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name	Text(30)		✓
Owner	OwnerId	Lookup(User/Group)		✓
Property	property_c	Lookup(Property)		✓
Start Date	StartDate_c	Date		

This screenshot shows the same Salesforce Setup interface, but the object selected is 'Tenant'. The title 'SETUP > OBJECT MANAGER' and the object name 'Tenant' are visible. The sidebar and the 'Fields & Relationships' table are identical to the 'lease' object, listing the same fields and their configurations.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Phone	Phone_c	Phone		
Status	Status_c	Picklist		
Suspect Name	SuspectName	Name		✓

● Developed Lightning App with relevant tabs

Lightning App Builder

App Settings

App Details & Branding

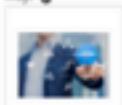
App Details

App Name: Lease Management

Developer Name: Lease_Management

Description: Application to efficiently handle the processes related to leasing real estate properties.

App Branding

Image: 

Primary Color Hex Value: #003366

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

App Launcher Preview



Lightning App Builder

App Settings

Navigation Items

Available Items

Type to filter list...

Create

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

Selected Items

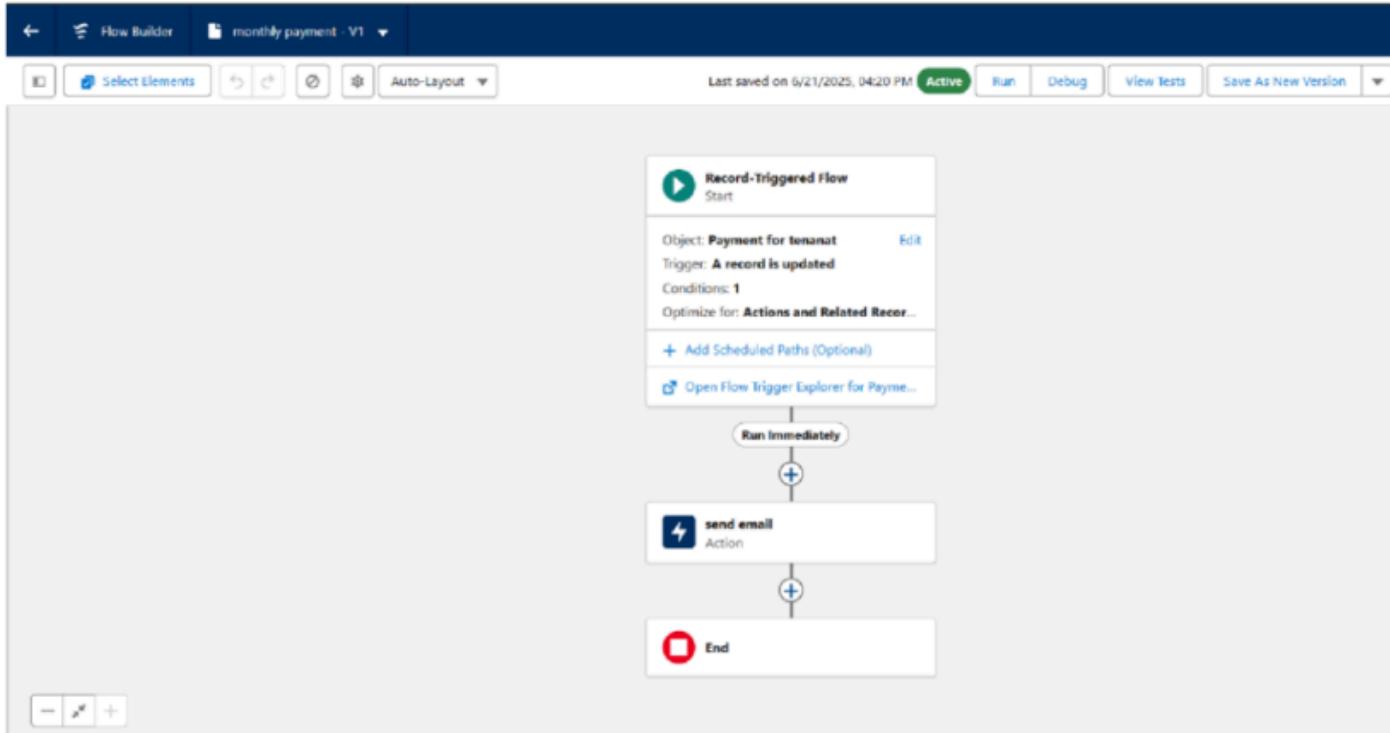
- Payment
- Tenants
- property
- lease

The screenshot shows the 'User Profiles' section of the Lightning App Builder. On the left, a sidebar lists 'App Settings' options: App Details & Branding, App Options, Utility Items (Desktop Only), Navigation Items, and User Profiles (which is currently selected). The main area is titled 'User Profiles' and contains a sub-instruction: 'Choose the user profiles that can access this app.' Below this, two sections are displayed: 'Available Profiles' and 'Selected Profiles'. The 'Available Profiles' section lists ten profiles: Analytics Cloud Integration User, Analytics Cloud Security User, Anypoint Integration, Authenticated Website, B2B Reordering Portal Buyer Profile, Contract Manager, Custom: Marketing Profile, Custom: Sales Profile, Custom: Support Profile, and Customer Community Login User. The 'Selected Profiles' section shows one profile: System Administrator.

The screenshot shows the 'Lease Management' application interface. At the top, there are navigation tabs: 'Lease Management', 'Payment' (selected), 'Tenants', 'property', 'lease'. A search bar is located at the top right. The main content area displays a list of 'Recently Viewed' payments. The list header includes columns for 'Payment Name' and checkboxes. Below the header, five items are listed, each with a checkbox and a name: 1. Rahul, 2. Jack, 3. Raj, 4. Sam, 5. Lahari. To the right of the list are various action buttons: 'New', 'Import', 'Change Owner', 'Assign Label', and search/filter icons.

Implemented Flows for monthly rent and payment

success●



- To create a validation rule to a LeaseObject

Setup Home Object Manager

SETUP > OBJECT MANAGER

lease

Validation Rule Edit

Rule Name: lease_end_date

Active:

Description:

Error Condition Formula

Example: Discount_Percent > 30 More Examples...
Display an error if Discount is more than 30%
If this formula expression is true, display the text defined in the Error Message area

Insert Field Insert Operator End date <= start date

Functions

ABS
ACOS
ADDMONTHS
AND
ASCII
ASIN

Insert Selected Function ABS(number)
Returns the absolute value of a number, a number without its sign

Check Syntax

orgfaam-5d1e805f2-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01lgK00000z1u/ValidationRules/03dgk00000C8grQAg/view

lease Validation Rule

Validation Rule Detail

Field	Value
Rule Name	lease_end_date
Error Condition Formula	End_date_c <= start_date_c
Error Message	Your End date must be greater than start date
Description	
Created By	Sowmya Team, 6/19/2025, 5:37 AM
Modified By	Sowmya Team, 6/26/2025, 7:47 AM
Active	✓
Error Location	start 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

● Added Apex trigger to restrict multiple tenants per property

New Tenant

Information

* = Required Information

Tenant Name: chinnu

Email: chinnu@gmail.com

Phone:

Status: Stay

property: Parkside

We hit a snag.

Review the errors on this page.

* A tenant can have only one property

Cancel Save & New Save

● Scheduled monthly reminder emails using Apex class

```
67% - + Reset
```

```
IScheduler implements Schedulable {  
    SchedulableContext sc) {  
        ay = Date.today().day();  
        * 1) {  
            emails();  
  
            endMonthlyEmails() {  
                tenants = [SELECT Id, Email__c FROM Tenant__c];  
                tenant : tenants) {  
                    tenantEmail = tenant.Email__c;  
  
                    Content = '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';  
                    Subject = 'Reminder: Monthly Rent Payment Due';  
  
                    SingleEmailMessage email = new Messaging.SingleEmailMessage();  
                    addresses(new String[]{recipientEmail});  
                    subject(emailSubject);  
                    inTextBody(emailContent);  
                }  
            }  
        }  
    }  
}
```

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

Setup Home Object Manager

Q email template

v Email

Classic Email Templates Lightning Email Templates

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

SETUP Classic Email Templates

Text Email Template 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	Saamya_Team [Changed]
Description	
Created By	Saamya_Team, 6/20/2025, 1:06 AM
Modified By	Saamya_Team, 6/20/2025, 1:06 AM

Available For Use ✓

Last Used Date

Times Used

Send Test and Verify Merge Fields

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 status is confirmed. You can leave now.

Setup Home Object Manager

Q email template

v Email

Classic Email Templates Lightning Email Templates

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

SETUP Classic Email Templates

Text Email Template tenant leaving

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Tenant leaving
Template Unique Name	Tenant_leaving
Encoding	Unicode (UTF-8)
Author	Saamya_Team [Changed]
Description	
Created By	Saamya_Team, 6/20/2025, 1:06 AM
Modified By	Saamya_Team, 6/20/2025, 1:06 AM

Available For Use ✓

Last Used Date

Times Used

Send Test and Verify Merge Fields

Subject | request for approve the leave

Plain Text Preview |

Dear {Tenant__c.CreatedBy},

Please approve my leave

Cloud icon

Search Setup

Setup Home Object Manager

Q_email template

v Email

Classic Email Templates

Lightning Email Templates

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

Text Email Template

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	Sowmya_Team [Change]
Description	
Created By	Sowmya_Team 6/20/2025, 1:11 AM
Modified By	Sowmya_Team 6/20/2025, 1:11 AM

Available For Use ✓

Last Used Date

Times Used

Edit Delete Close

Email Template

Send Test and Verify Merge Fields

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

Cloud icon

Search Setup

Setup Home Object Manager

Q_email template

v Email

Classic Email Templates

Lightning Email Templates

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

Text Email Template

Tenant Email

Preview your email template below.

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	Sowmya_Team [Change]
Description	
Created By	Sowmya_Team 6/20/2025, 1:12 AM
Modified By	Sowmya_Team 6/20/2025, 1:12 AM

Available For Use ✓

Last Used Date

Times Used

Edit Delete Close

Email Template

Send Test and Verify Merge Fields

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.

Home Object Manager

SETUP Classic Email Templates

Text Email Template tenant payment

Preview your email template below.

Email Template Detail	
Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant payment
Template Unique Name	tenant_payment
Encoding	Unicode (UTF-8)
Author	Soumya Team (Change)
Created By	Soumya Team, 6/20/2025, 1:13 AM
Modified By	Soumya Team, 6/20/2025, 1:13 AM

[Edit](#) [Delete](#) [Clone](#)

[Available For Use](#)

Last Used Date Times Used

[Edit](#) [Delete](#) [Clone](#)

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.

[Send Test and Verify Merge Fields](#)

Approval Process creation For Tenant Leaving:

Setup Home Object Manager

approval

Approval Processes

Tenant: TenantApproval

Process Definition Detail

Process Name	TenantApproval	Active	
Unique Name	TenantApproval	<input checked="" type="checkbox"/>	
Description		Next Automated Approver Determined By	
Entry Criteria	Tenant__r.status equals Stay		
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests	
Approval Assignment Email Template			
Initial Submitters	Tenant Owner		
Created By	Soumya Team, 6/23/2025, 3:41 AM	Modified By	Soumya Team, 6/26/2025, 11:57 PM

Initial Submission Actions

Action Type	Description
Record Lock	Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Step 1			User: Soumya Team	Final Rejection

ForCheck for Vacant:

orgfarm-5d1fe805f2-dev-ed.lightning.force.com/lightning/setup/ApprovalProcesses/page?address=%2F04ag0000004rbz

SETUP
Approval Processes
Tenant: check for vacant
< Back to Approval Process List

Process Definition Detail

Process Name:	check for vacant	Active:	✓
Unique Name:	check_for_vacant	Next Automated Approver Determined By:	
Description:			
Entry Criteria:	Tenant__r.status EQUALS Leaving		
Record Editability:	Administrator ONLY	Allow Submitters to Recall Approval Requests:	<input checked="" type="checkbox"/>
Approval Assignment Email Template:	Leave approved		
Initial Submitters:	Tenant Owner		
Created By:	Sowmya_Team	Modified By:	Sowmya_Team
Created On: 6/20/2025, 3:18 AM Modified On: 6/26/2025, 11:02 PM			

Initial Submission Actions

Action	Type	Description
Record Lock		Lock the record from being edited
Edit Remove	Email Alert	PLEASE APPROVE MY LEAVE

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	step1			User:Sowmya_Team	Final Rejection

● Apex Trigger

Create an Apex Trigger

Handler.apc HostByInnScheduler.apc

```
trigger test on Tenant__c (before insert)

if(trigger.isInsert && trigger.isBefore){

    testHandler.preventInsert(trigger.new);

    Open
    Entity Type: Triggers
    Edition: test
    Namespace: 
    Related: 
    Open Filter Hide Managed Packages Refresh
```

Checkpoints Query Editor View Status Progress Problems

Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >
testHandler.apc testHandler.apc MonthlyEmailScheduler.apc

Code Coverage: None API Version: 64

```

1 trigger test on Tenant__c (before insert)
2
3 {
4
5   if(trigger.isInsert && trigger.isBefore){
6
7     testHandler.preventInsert(trigger.new);
8
9   }
10
11 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Create an Apex Handler class

Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Test Workspace Help < >
testHandler.apc testHandler.apc MonthlyEmailScheduler.apc

API Version: 64

```

class testHandler {

  ic 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(exi
```

Entity Type	Edition	Related
Entity Type	Name: testHandler Namespace: MonthlyEmailScheduler	Name: test Extent: ApexTrigger Referenced Name: property Extent: CustomField References ← Tenant__c Extent: SObject References ← Tenant__c Extent: SObject References
Classes		
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

Developer Console - Google Chrome

orgfarm-Sdf1eB05f2-dev-ed develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

testHandler.apc MonthlyEmailScheduler.apc

Code Coverage: None API Version: 64

```

1 + public class testHandler {
2
3 +     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existin
10
11         }
12
13         for (Tenant__c newTenant : newList) {
14
15             if (newTenant.Property__c != null) {
16
17                 newTenantaddError('A tenant can have only one property');
18             }
19
20         }
21
22     }
23 }
```

Open

Entity Type	Entity Name	Namespace	Related
EMPTY TYPE			
Classes	testHandler		← test
Triggers			Apextigger
Pages			CustomTab
Page Components			References
Objects			Tenant__c
Static Resources			Object
Packages			References

Filter: Filter the repository (* = any string)

Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

● FLOWS

Developer Console - Google Chrome

orgfarm-Sdf1eB05f2-dev-ed develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

testHandler.apc MonthlyEmailScheduler.apc

API Version: 64

```

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');
            }
        }
    }
}
```

Checkpoints Query Editor View State Progress Problems

Line Problem

Builder monthly payment - V1

Last saved on 6/21/2025, 04:20 PM Active Run Debug View Tests Save As New Version Save Deactivate

Record-Triggered Flow

Start

Object: Payment for tenant Edit

Trigger: A record is updated

Conditions: 1

Optimize for: Actions and Related Records

+ Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Payment...

Run Immediately

send email Action

End

Configure Start

Field: check for payment 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 Flow

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.

Is this flow making an external callout or connecting to an external system?

An asynchronous path is required for flows that involve external systems.

Add Asynchronous Path

Schedule class: Create anApexClass

Builder monthly payment - V1

Last saved on 6/21/2025, 04:20 PM Active Run Debug View Tests Save As New Version Save Deactivate

Record-Triggered Flow

Start

Object: Payment for tenant Edit

Trigger: A record is updated

Conditions: 1

Optimize for: Actions and Related Records

+ Add Scheduled Paths (Optional)

Open Flow Trigger Explorer for Payment...

Run Immediately

send email Action

End

Configure Start

Select Object
Select the object whose records trigger the flow when they're created, updated, or deleted.
Object: Payment for tenant

Configure Trigger

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

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)

ScheduleApex class

file - Google Chrome

Te805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

50% Reset

```
public class EmailScheduler implements schedulable {
    execute(SchedulableContext sc) {
        currentDay = Date.today().day();
        if (currentDay == 1) {
            sendMonthlyEmails();
        }
    }

    void sendMonthlyEmails() {
        List<Tenants> 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.";
            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
            email.setToAddresses(new String[]{recipientEmail});
            email.setSubject('RENT DUE');
            email.setPlainTextEmailContent();
            Messaging.SingleEmailMessage[] emails = {email};
            Messaging.sendEmail(emails);
        }
    }
}
```

File Insert Refresh Help

Setup Home Object Manager

Search Setup apex

Email Apex Exception Email

Custom Code Apex Classes

Apex Settings

Apex Test Execution

Apex Test History

Apex Triggers

Environments Jobs

Apex Flex Queue

Apex Jobs

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

Apex Class MonthlyEmailScheduler

Apex Class Detail

Name	MonthlyEmailScheduler	Status	Active
Namespace Prefix		Code Coverage	0% (0/15)
Created By	Sowmya Team . 6/23/2025, 2:46 AM	Last Modified By	Sowmya Team . 6/23/2025, 2

Class Body Class Summary Version Settings Trace Flags

```
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            ...  
12        }  
13    }  
14}  
15  
16  
17  
18  
19  
20
```

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

* Required Information

* Tenant Name Aswini

* Email aswiniamaraadi15@gmail.com

Phone (905) 223-5567

status Leaving

property Imran

Created By Sowmya Team . 6/26/2025, 6:05 AM

Last Modified By Sowmya Team . 6/26/2025, 11:06 PM

Cancel Save

Activity

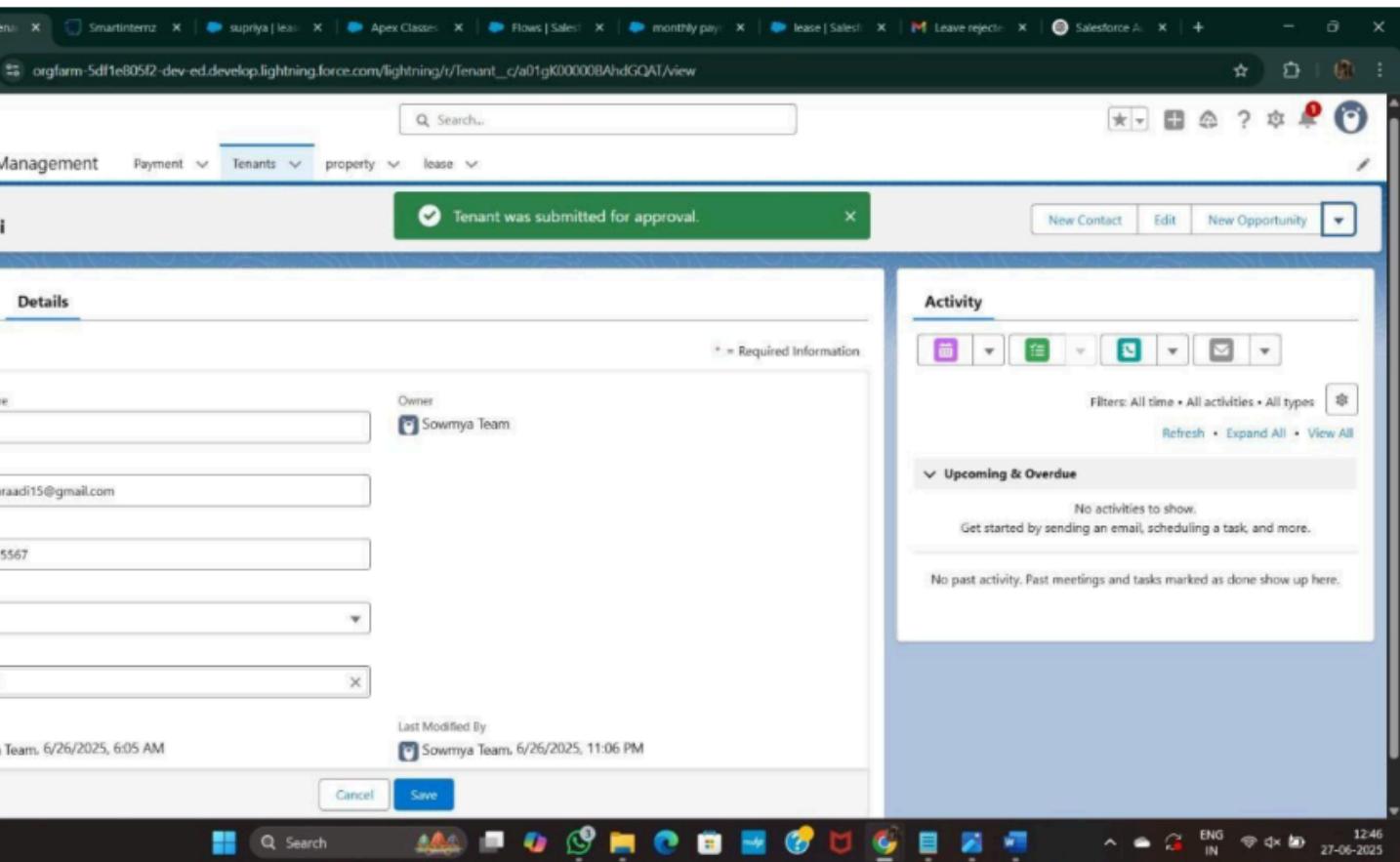
New Contact Edit

Filters: All time + All activities Refresh

Upcoming & Overage

No activities to show. Get started by sending an email, scheduling a task or meeting.

No past activity. Past meetings and tasks marked as complete.



FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

- Trigger validation by entering duplicate tenant-property records

Lease Management Payment Tenants property lease

Tenant Aswini

Related Details

* = Required Information

Tenant Name: Aswini

Email: aswiniamaraadi15@gmail.com

Phone: (905) 223-5567

Status: Leaving

Property: Imran

Owner: Sowmya Team

Created By: Sowmya Team, 6/26/2025, 6:05 AM

Last Modified By: Sowmya Team, 6/26/2025, 11:06 PM

Activity

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

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

● Validation Rule checking

Smartinternz X | supriya | lease X | Apex Classes X | Flows | Sales! X | monthly pay X | lease | Sales! X | Leave rejectin X | Salesforce A X | +

management Payment Tenants property lease

Tenant Aswini

Details

* = Required Information

Tenant was submitted for approval.

Email: adi15@gmail.com

Phone: (905) 223-5567

Status: Leaving

Property: Imran

Owner: Sowmya Team

Created By: Sowmya Team, 6/26/2025, 6:05 AM

Last Modified By: Sowmya Team, 6/26/2025, 11:06 PM

Activity

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.

orgfarm-5df1e805f2-dev-ed.lightning.force.com/lightning/r/ProcessInstanceStep/04hgK000000CYNhQAO/view

The screenshot shows a Salesforce Lightning page for a process instance step. The top navigation bar includes links for Management, Payment, Tenants, property, lease, Aswini Approval, and Notifications. The main content area displays the approval status as 'Approved' with a timestamp of 'Jun 27, 2025'. Below this, there's a 'Details' section showing the owner as 'Sowmya Team'. On the right side, a sidebar titled 'Notifications' lists five recent messages from 'Aswini' and 'Kiran' regarding approval requests.

Management Payment Tenants property lease Aswini Approval Notifications

instance Step Approval Approved

Date Submitted Jun 27, 2025

Actual Approver Sowmya Team

Assigned To Sowmya Team

Owner Sowmya Team

Notifications

Approval request for the tenant is approved Aswini a few seconds ago *

Approval request for the tenant is rejected Aswini an hour ago *

Approval request for the tenant is approved Aswini an hour ago *

Approval request for the tenant is approved Aswini an hour ago *

Approval request for the tenant is approved Kiran 19 hours ago *

● Testflows on payment update

The screenshot shows a 'New Tenant' dialog box. The 'Information' section contains fields for Tenant Name ('chinnu'), Email ('chinnu@gmail.com'), Phone, status ('Stay'), and property ('Parkside'). A red error message box states: 'We hit a snag.' and 'Review the errors on this page.' with the note: 'A tenant can have only one property.' The dialog has buttons for 'Cancel', 'Save & New', and 'Save'.

e Management Payment Tenants property lease

Recently Viewed

3 minutes ago

Tenant Name niranjan

New Tenant

Information

* = Required Information

Tenant Name chinnu

Email chinnu@gmail.com

Phone

status Stay

property Parkside

We hit a snag.

Review the errors on this page.

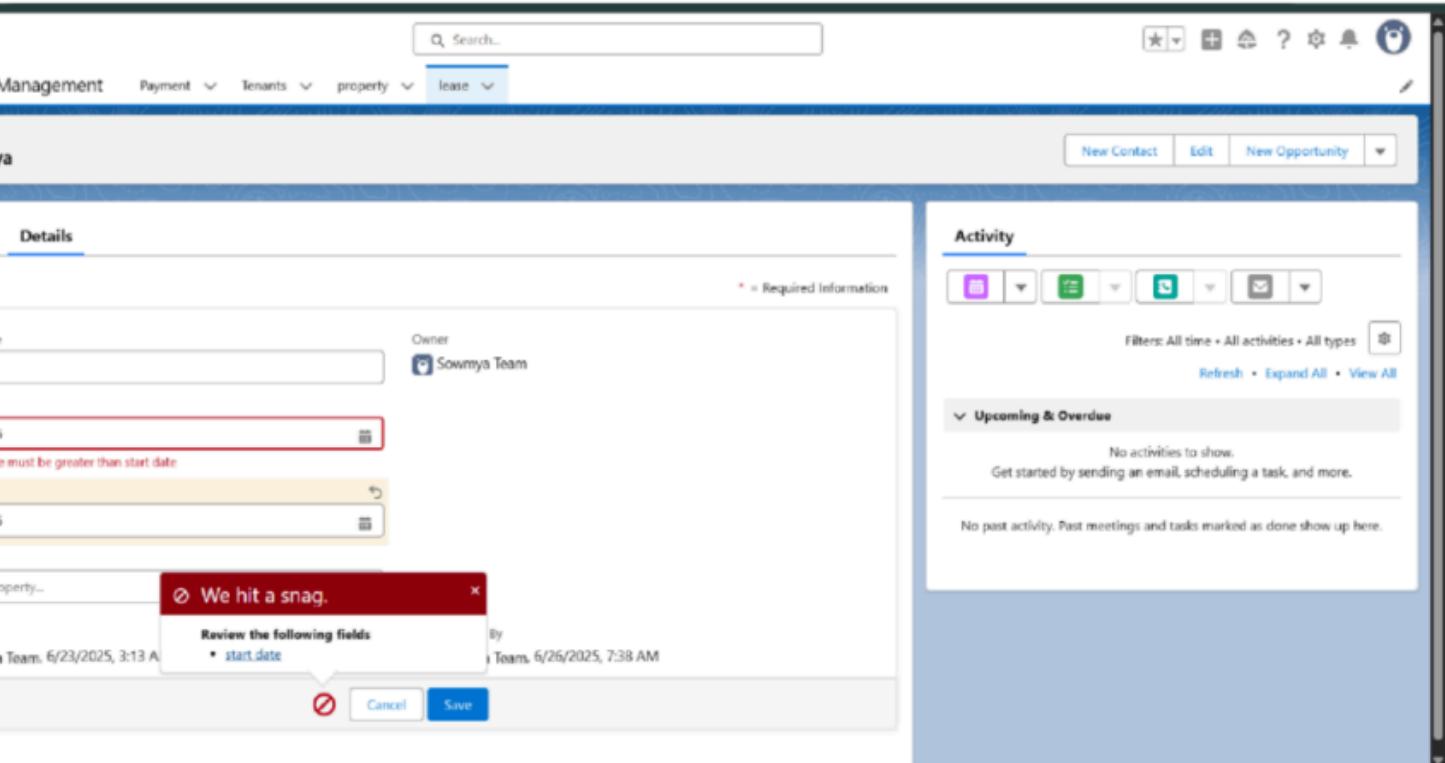
A tenant can have only one property.

Cancel Save & New Save

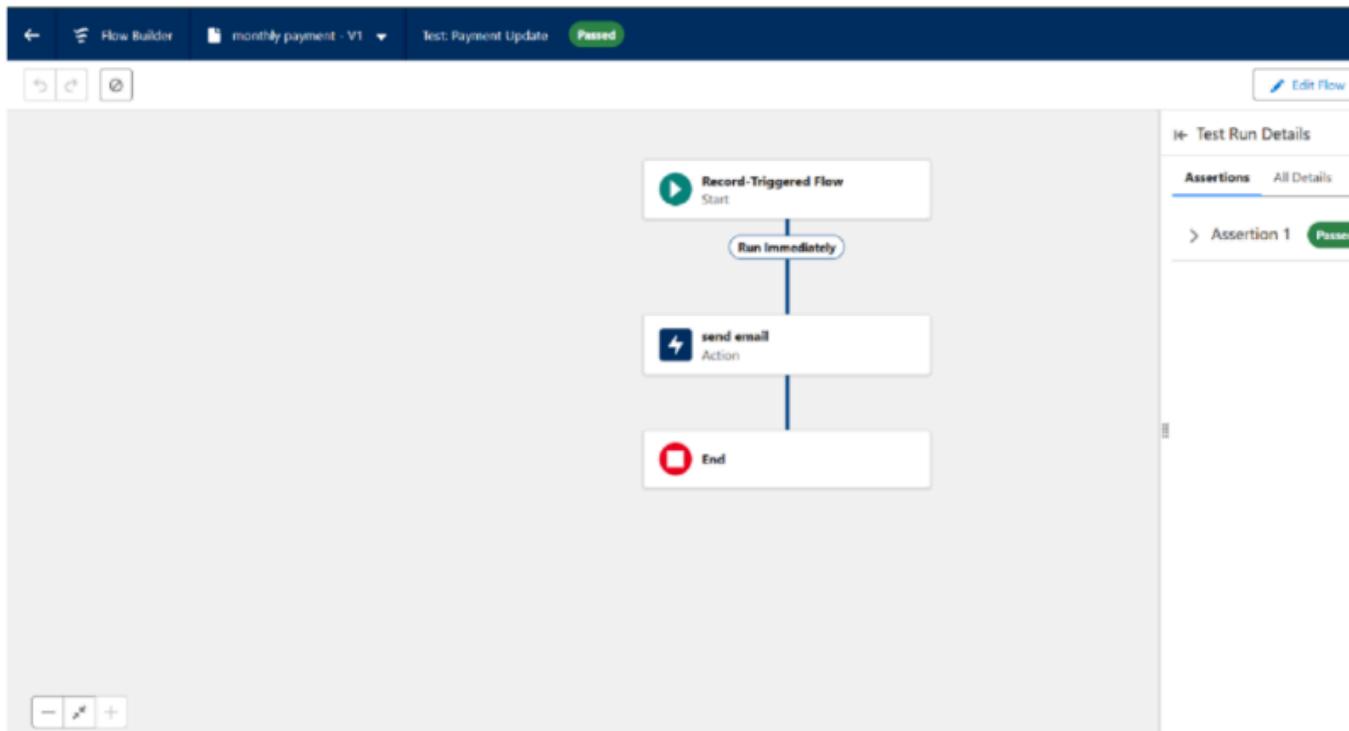
RESULTS

Output Screenshots

● Tabs for Property, Tenant, Lease, Payment



● Flowruns



Trigger error messages •

The screenshot shows the Lease Management application with the following details:

- Tenant:** niranjan
- Details:**
 - Tenant Name: niranjan
 - Email: niranjan1506@gmail.com
 - Phone: (empty)
 - Status: Stay
 - property: Parkside Lofts
 - Created By: Sowmya Team, 6/23/2025, 2:33 AM
 - Last Modified By: Sowmya Team, 6/23/2025, 3:58 AM
- Notifications:**
 - Approval request for the tenant is approved niranjan (a few seconds ago)
 - Approval request for the tenant is rejected niranjan (Jun 23, 2025, 4:29 PM)
 - Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:25 PM)
 - Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:14 PM)
 - New Guidance Center learning resource available Define Your Sales Process Learn how to guide reps through the sales process. (Jun 20, 2025, 1:28 PM)

● Approval process notifications

The screenshot shows a Salesforce application interface for Lease Management. At the top, there are navigation tabs: Lease Management, Payment, Tenants, property, lease, and a search bar. Below the tabs, it says "Tenant niranjan". On the right side, there are buttons for New Contact, Edit, and a gear icon. A sidebar on the right says "Get started by sending an email, scheduling a meeting, or creating a task." It also notes "No past activity. Past meetings and tasks marked as completed." The main content area has two sections: "Approval History (6+)" and "Payment (2)".

Step Name	Date	Status	Assigned To
Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team
Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team
Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team
Approval Request submitted	6/23/2025, 3:55 AM	submitted	Sowmya Team

[View All](#)

Payment Name
Jack
Rahul

[View All](#)

ADVANTAGES & DISADVANTAGES

CONCLUSION

The Lease Management System successfully streamlines the operations of leasing through a structured, automated Salesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.

APPENDIX

- SourceCode:
Provided in Apex Classes and Triggers

Test.apxt:

```

trigger teston 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

= newSet<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}); }  
}  
}
```