

LEASE MANAGEMENT

College Name: SHREE VENKATESHWARA ARTS AND SCIENCE COLLEGE

College Code: brubd

TEAM ID : NM2025TMID20900

TEAM MEMBERS : 4

Team LeaderName: Santhiya.S

Email: santhiyas2023cs@gmail.com

Team Member1: Priyadharshini.V

Email: vpriyadharshini444@gmail.com

Team Member: Bhavadharani.S

Email: bhavadharani2023cs@gmail.com

Team Member: Gowsalya.A

Email : gowsalyaa2023cs@gmail.com

1. INTRODUCTION

1.1 Project Overview

The Lease Management System is a Salesforce-based application designed to streamline the processes associated with leasing real estate properties. It handles tenant management, lease

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

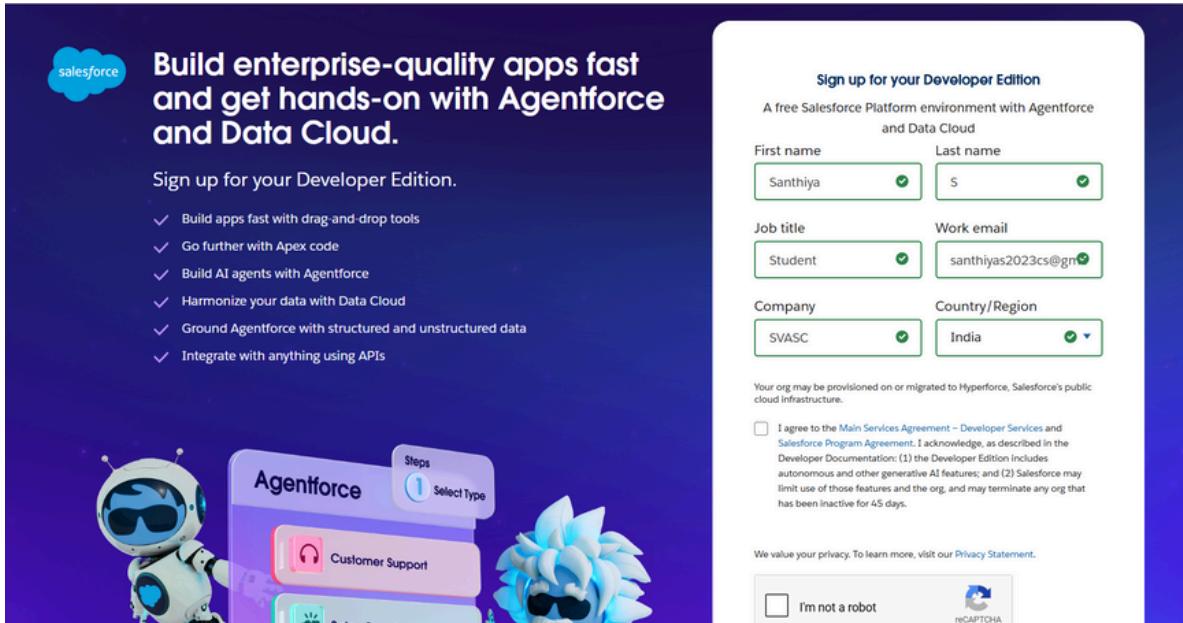


1. Purpose

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:



- Created objects: Property, Tenant, Lease, Payment

SETUP > OBJECT MANAGER

Tenant

Details

Description

API Name: Tenant__c

Custom:

- Singular Label: Tenant
- Plural Label: Tenants

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

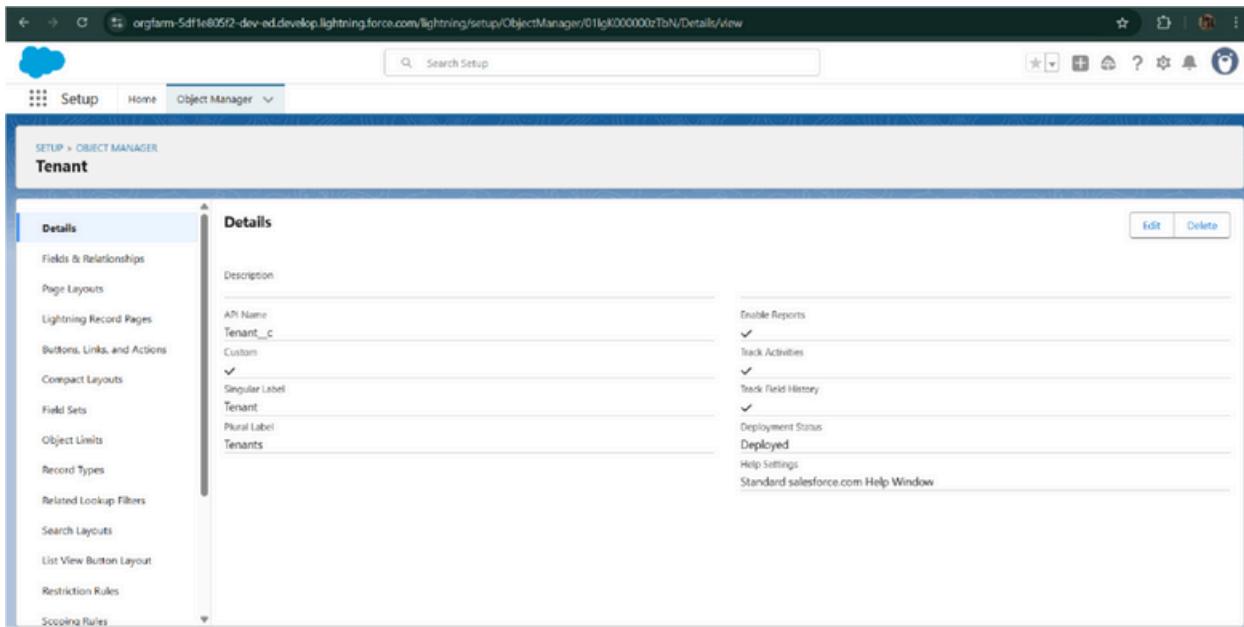
Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Edit Delete



SETUP > OBJECT MANAGER

lease

Details

Description

API Name: lease__c

Custom:

- Singular Label: lease
- Plural Label: lease

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

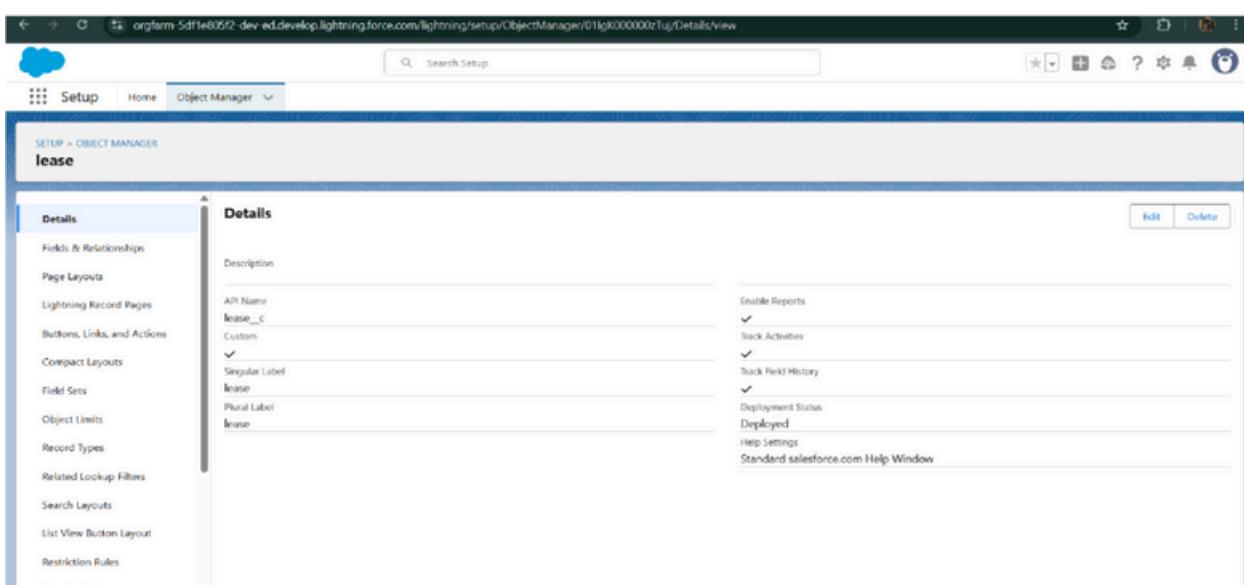
Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Edit Delete



Object Manager

Payment for tenant

Details

Description

- API Name: Payment_for_tenant__c
- Custom
- Singular Label: Payment for tenant
- Plural Label: Payments

Buttons, Links, and Actions

Reports

- Unfiled Reports
- Track Activities
- Track Field History
- Deployment Status
- Help Settings
- Standard Salesforce.com Help Window

- Configured fields and relationships

Fields & Relationships

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(30)		
sft	sft__c	Text(18)		
Type	Type__c	Picklist		

Setup Home Object Manager

SETUP > OBJECT MANAGER
Payment for tenantat

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

Fields & Relationships
7 Items, Sorted by Field Label

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)		

Setup Home Object Manager

SETUP > OBJECT MANAGER
lease

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

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		

Fields & Relationships
7 items. Sorted by Field Label

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		
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: Lease Management

*Developer Name: Lease_Management

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

App Branding

Image: [Placeholder image of a person in a suit]

Primary Color Hex Value: #0070D2

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

App Launcher Preview

[Placeholder image of the app launcher preview]

[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

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... [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

[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

App Settings

[App Details & Branding](#)

[App Options](#)

[Utility Items \(Desktop Only\)](#)

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Type to filter list...

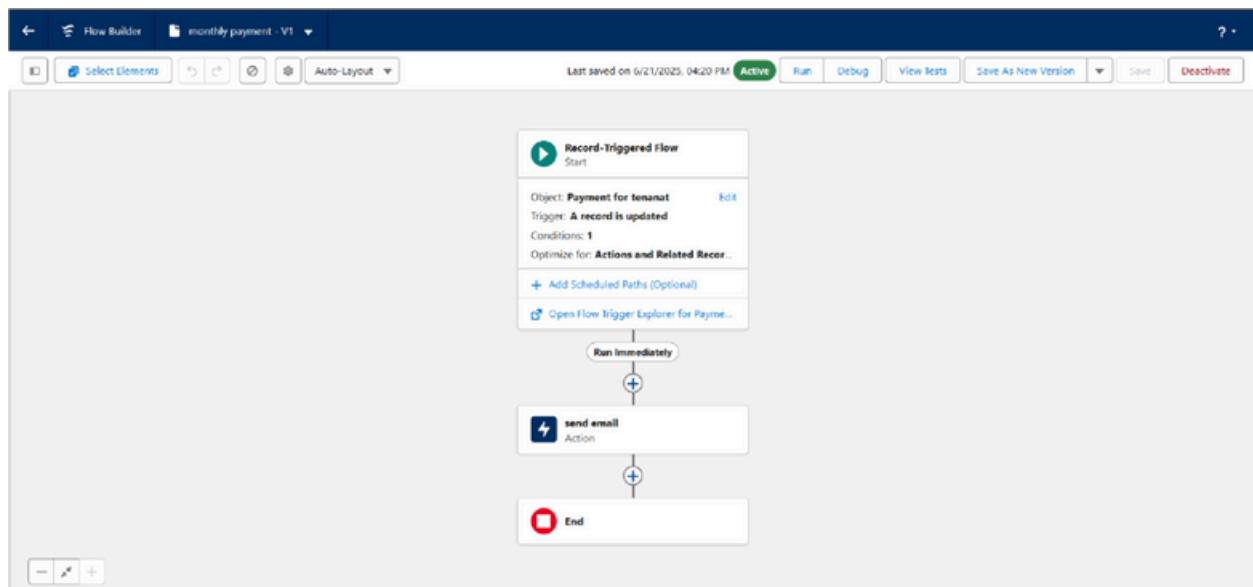
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

Selected Profiles

- System Administrator

The screenshot shows the Microsoft Dynamics 365 Lease Management Properties screen. The top navigation bar includes 'Lease Management', 'Home', 'Properties', 'Tenants', and 'Leases'. The main area displays a list of properties with one item visible: 'Hyd'. The list has a header 'Property Name ↑' and a search bar 'Search this list...'. Action buttons for 'New', 'Import', 'Change Owner', 'Printable View', and 'Assign Label' are at the top right. A toolbar with various icons is located at the bottom right.

- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

The screenshot shows the Salesforce Setup interface with the URL <https://org1a.m-5d1fe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lg0000000z1uq/ValidationRules/page?address=%2FO3dgk000000RgrQ%G%2Fe%3FneURL%3D%2...>

Validation Rule Edit

Role Name: lease_end_date

Active:

Description:

Error Condition Formula:

```
Example: Discount_Percent__c > 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
```

Formula: End_date__c < start_date__c

Functions:

- All Function Categories -
- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

Check Syntax

Quick Tips: Operators & Functions

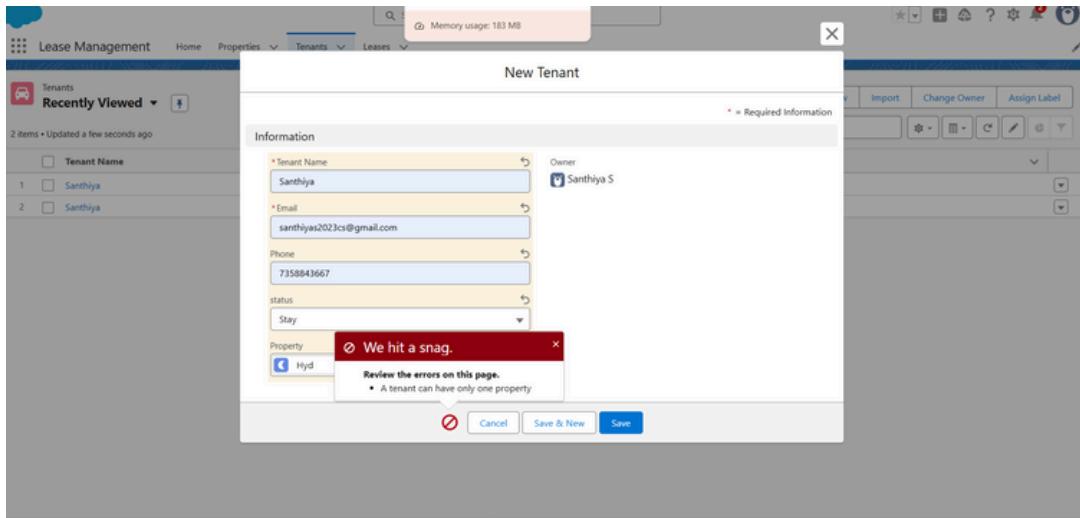
The screenshot shows the Salesforce Setup interface with the URL <https://org1a.m-5d1fe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lg0000000z1uq/ValidationRules/page?address=%2FO3dgk000000RgrQ%G%2Fe%3FneURL%3D%2...>

Lease Validation Rule

Validation Rule Detail:

Rule Name	lease_end_date	Active
Error Condition Formula	End_date__c > start_date__c	<input checked="" type="checkbox"/>
Error Message	Your End date must be greater than start date	Error Location: start date
Description		
Created By	Santhiya.S. 9/4/2025, 7:02 AM	Modified By: Santhiya.S. 9/4/2025, 7:02 AM

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

```
File Edit Insert View Favorites Help
File:///C:/Users/91991/Desktop/MonthlyEmailScheduler.java
MonthyEmailScheduler.java - 100% - + Reset
Code Change View API Version: 44
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+ public static void sendMonthlyEmails() {
16+
17+     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18+
19+     for (Tenant__c tenant : tenants) {
20+
21+         String recipientEmail = tenant.Email__c;
22+
23+         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';
24+
25+         String emailSubject = 'Reminder: Monthly Rent Payment Due';
26+
27+         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
28+
29+         email.setToAddresses(new String[]{recipientEmail});
30+
31+         email.setSubject(emailSubject);
32+
33+         email.setPlainTextBody(emailContent);
34+
35+     }
}
Top Test Checkstyle Run Unit Test View Source Request Problem
New
```

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

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', the 'Leave approved' template is displayed. The template details are as follows:

Email Template Detail

- Email Template Name: Leave approved
- Template Unique Name: leave_approved
- Encoding: Unicode (UTF-8)
- Author: Santhiya S [Change]
- Description: Created by Santhiya S 9/4/2025, 7:12 AM
- Modified By: Santhiya S 9/4/2025, 7:51 AM
- Available For Use: checked
- Last Used Date: Times Used

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.
```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', the 'tenant leaving' template is displayed. The template details are as follows:

Email Template Detail

- Email Template Name: tenant leaving
- Template Unique Name: tenant_leaving
- Encoding: Unicode (UTF-8)
- Author: Santhiya S [Change]
- Description: Created By Santhiya S 9/4/2025, 7:10 AM
- Modified By: Santhiya S 9/4/2025, 7:43 AM
- Available For Use: checked
- Last Used Date: Times Used

Email Template

Subject: request for approve the leave

Plain Text Preview:

```
Dear {Tenant__c CreatedBy}.

Please approve my leave
```

Setup Home Object Manager

Search Setup

Classic Email Templates

Leave rejected

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Untitled Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Santhya.S [Change]
Description	
Created By	Santhya.S 9/4/2025, 7:16 AM
Modified By	Santhya.S 9/4/2025, 7:51 AM

Available For Use

Last Used Date

Times Used

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

Send Test and Verify Merge Fields

Setup Home Object Manager

Search Setup

Classic Email Templates

Tenant Email

Preview your email template below.

Email Template Detail

Email Templates from Salesforce	Untitled Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Santhya.S [Change]
Description	
Created By	Santhya.S 9/4/2025, 7:19 AM
Modified By	Santhya.S 9/5/2025, 8:54 PM

Available For Use

Last Used Date

Times Used

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.

Send Test and Verify Merge Fields

The screenshot shows the Salesforce Setup interface with the search bar containing 'class'. The left sidebar has sections for Email (Classic Email Templates, Classic Letterheads), Custom Code (Apex Classes), and Data Classification (Data Classification Download, Data Classification Settings, Data Classification Upload). A message at the bottom says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'SETUP Classic Email Templates' and shows a 'Text Email Template' named 'tenant payment'. The template details include:

Email Template Detail	Value
Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant payment
Template Unique Name	tenant_payment
Encoding	Unicode (UTF-8)
Author	Santhiya S [Change]
Description	
Created By	Santhiya S 9/4/2025, 7:20 AM
Available For Use	<input checked="" type="checkbox"/>
Last Used Date	
Times Used	
Modified By	Santhiya S 9/5/2025, 8:55 PM

Below this is a preview section titled 'Email Template' with a 'Plain Text Preview' containing the following content:

```

Subject: Confirmation of Successful Monthly Payment
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 Tenant Leaving :

The screenshot shows the Salesforce Setup interface with the search bar containing 'approval pro'. The left sidebar has sections for Process Automation (Approval Processes). A message at the bottom says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'SETUP Approval Processes' and shows an 'Approval Process' named 'Tenant: Tenant Approval'. The process details include:

Process Definition Detail	Value
Process Name	Tenant Approval
Unique Name	Tenant_Approval
Description	
Entry Criteria	
Record Editability	Administrator ONLY
Approval Assignment Email Template	tenant_leaving
Initial Submitters	Tenant Owner
Created By	Santhiya S 9/5/2025, 8:31 PM
Modified By	Santhiya S 9/5/2025, 8:32 PM

Below this are sections for 'Initial Submission Actions' and 'Approval Steps'.

For Check for Vacant:

Approval Processes

Process Definition Detail

- Process Name: check for vacant
- Unique Name: check_for_vacant
- Description: Tenant__status NOT EQUAL TO Leaving
- Entry Criteria: Administrator ONLY
- Record Editability: Administrator ONLY
- Next Automated Approver Determined By: [checkbox]
- Allow Submitters to Recall Approval Requests: [checkbox]
- Approval Assignment Email Template: Tenant Owner
- Initial Submitters: Tenant Owner
- Created By: Santhiya_S, 9/4/2025, 7:32 AM
- Modified By: Santhiya_S, 9/4/2025, 7:56 AM

Initial Submission Actions

Action	Type	Description
Record Lock	Record Lock	Lock the record from being edited please approve my leave

Approval Steps

● Apex Trigger

Create an Apex Trigger

```

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

```

Open

Entity Type	Edition	Related
Classes	test	
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `orgfarm-5d1fe805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage`. The tab bar shows `testHandler.apex` is active. The code editor contains the following Apex trigger:

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

The status bar at the bottom indicates "None" under "Logs", "Tests", "Checkpoints", "Query Editor", "View Status", "Progress", and "Problems". The "Problems" tab is highlighted.

Create an Apex Handler class

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `orgfarm-5d1fe805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage`. The tab bar shows `testHandler.apex` is active. The code editor contains the following Apex class:

```
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
14         for (Tenant__c newTenant : newList) {
15
16             if (newTenant.Property__c != null) {
17
18                 newTenantaddError('A t
19
20             }
21
22         }
23     }
24 }
```

A dependency graph overlay is displayed over the code editor, showing relationships between classes and triggers. The graph includes nodes for `testHandler`, `ApexTrigger`, `Property`, `Tenant__c`, and `CustomerID`.

The status bar at the bottom indicates "None" under "Logs", "Tests", "Checkpoints", "Query Editor", "View Status", "Progress", and "Problems". The "Problems" tab is highlighted.

Developer Console - Google Chrome

orgfarm-Sd1t0e05f2-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage

Test.Javascript testHandler.apc [] HostedHTMLStaticScheduler.apc []

Code Coverage Name: API Version: 64 Go To

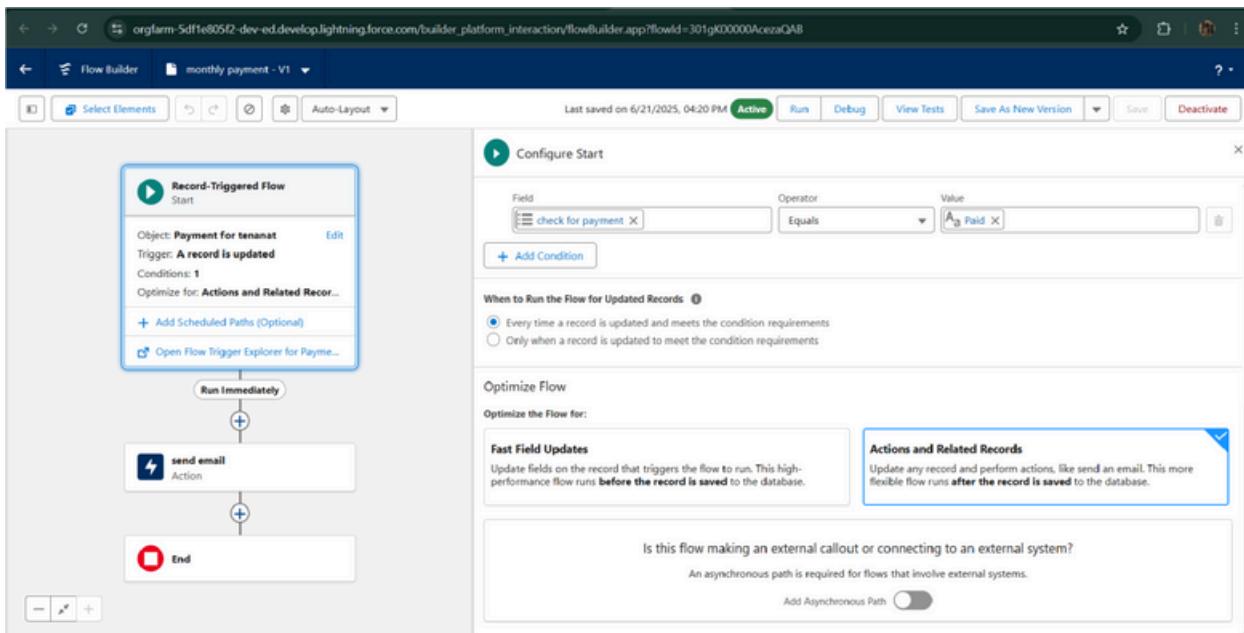
```

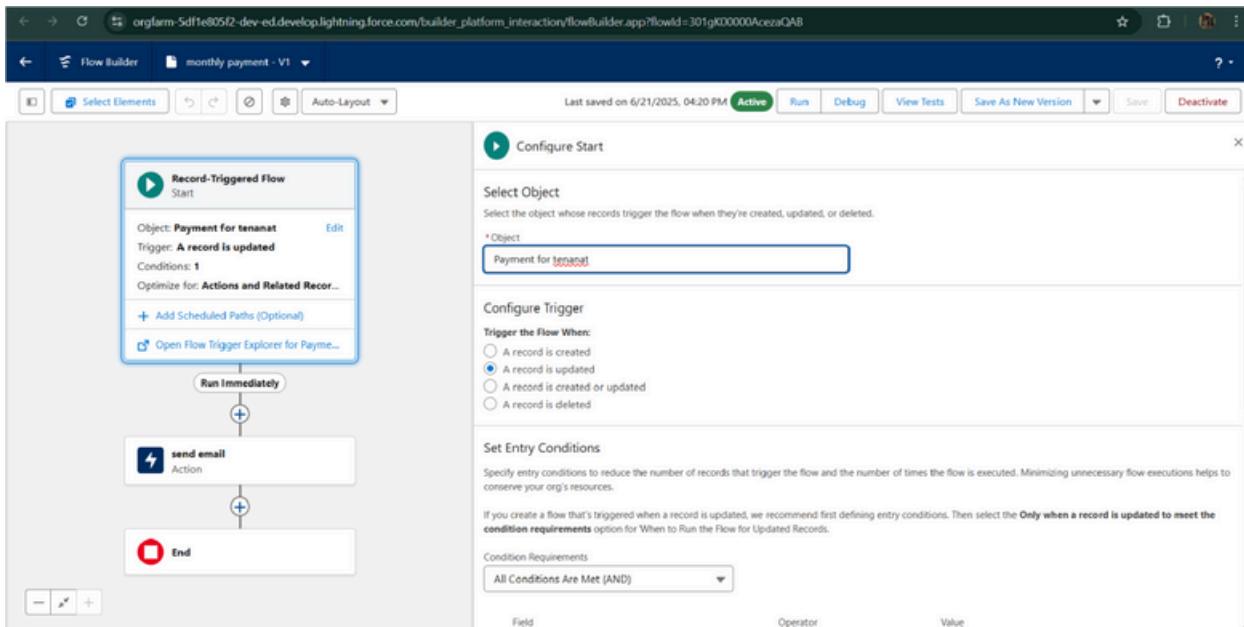
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11         }
12
13
14         for (Tenant__c newTenant : newList) {
15
16
17             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
18
19                 newTenantaddError('A tenant can have only one property');
20
21             }
22
23         }
24
25     }
26
27 }
```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

● FLOWS





- Schedule class:
Create an 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 sendMonthlyEmail
17
18     List<Tenant__c> tenants = [SELECT
19
20         for (Tenant__c tenant : tenants)
21             String recipientEmail = tenant.Email__c;
  
```

Entity Type: Classes
Name: MonthlyEmailScheduler Namespace: **testhandler**

Entity Type	Edition	Related
Classes	Name: MonthlyEmailScheduler	Name: testhandler CustomType: Email CustomField: Tenant__c CustomObject: Tenant__c CustomRelationship: Tenant__c CustomType: Tenant__c CustomField: Tenant__c CustomObject: Tenant__c CustomRelationship: Tenant__c
Triggers		
Pages		
Page Components		
Objects		
Static Resources		
Packages		

```

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             sendMonthlyEmails();
9         }
10    }
11
12
13
14
15    public static void sendMonthlyEmails() {
16
17        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18
19        for (Tenant__c tenant : tenants) {
20
21            String recipientEmail = tenant.Email__c;
22
23            String emailSubject = 'Reminder: monthly rent payment due';
24
25            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.';
26
27            Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29            mail.setTo(recipientEmail);
30            mail.setSubject(emailSubject);
31            mail.setPlainTextEmailBody(emailContent);
32
33            messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
34
35        }
36    }
37}

```

Schedule Apex class

The screenshot shows the Salesforce Setup interface. The left sidebar navigation includes 'Email' (with 'Apex Exception Email'), 'Custom Code' (selected), 'Environments', and 'Jobs'. Under 'Custom Code', 'Apex Classes' is selected, showing the Apex Class Detail for 'MonthlyEmailScheduler'. The class details show it was created by Santhya S on 9/4/2025 at 9:58 AM, is active, and has 0% code coverage. The 'Class Body' tab displays the Apex code provided in the developer console.

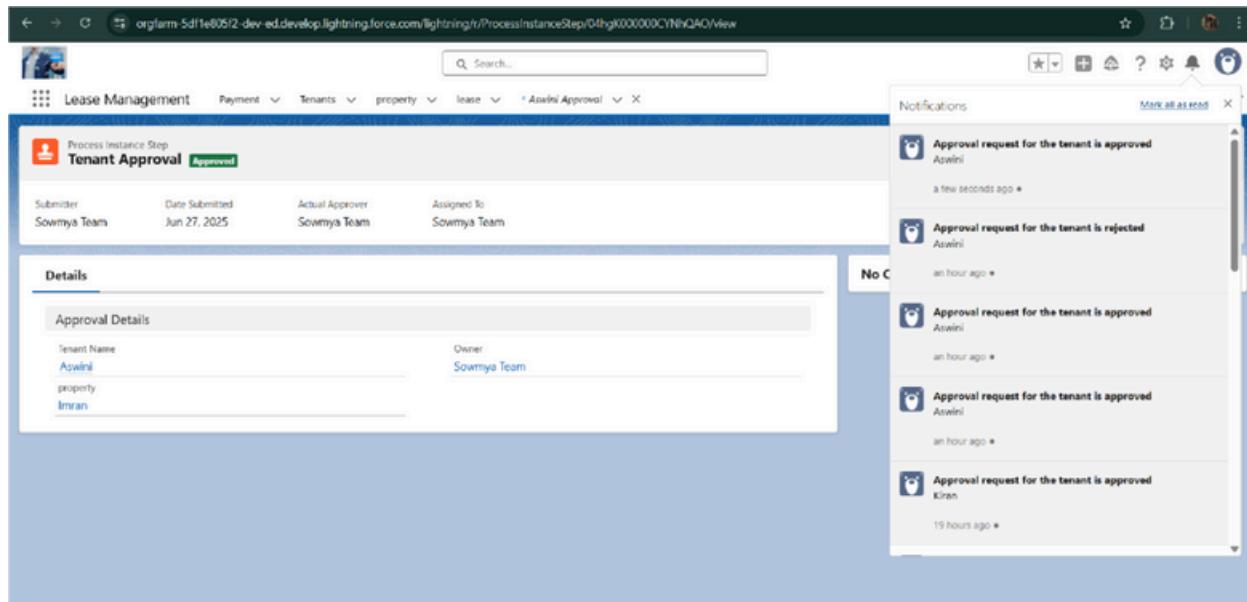
```

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             sendMonthlyEmails();
9         }
10    }
11
12
13
14
15    public static void sendMonthlyEmails() {
16
17        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18
19        for (Tenant__c tenant : tenants) {
20
21            String recipientEmail = tenant.Email__c;
22
23            String emailSubject = 'Reminder: monthly rent payment due';
24
25            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.';
26
27            Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29            mail.setTo(recipientEmail);
30            mail.setSubject(emailSubject);
31            mail.setPlainTextEmailBody(emailContent);
32
33            messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
34
35        }
36    }
37}

```

The screenshot shows a tenant record for "Santhiya". The "Details" tab is selected. The tenant's name is "Santhiya", email is "santhiyas2023cs@gmail.com", phone is "(735) 884-3667", and property is "Hyd". The record was created by "Santhiya S" on 9/4/2025 at 8:28 AM and last modified by "Santhiya S" on 9/5/2025 at 4:28 AM. The "Activity" sidebar shows no upcoming or overdue activities.

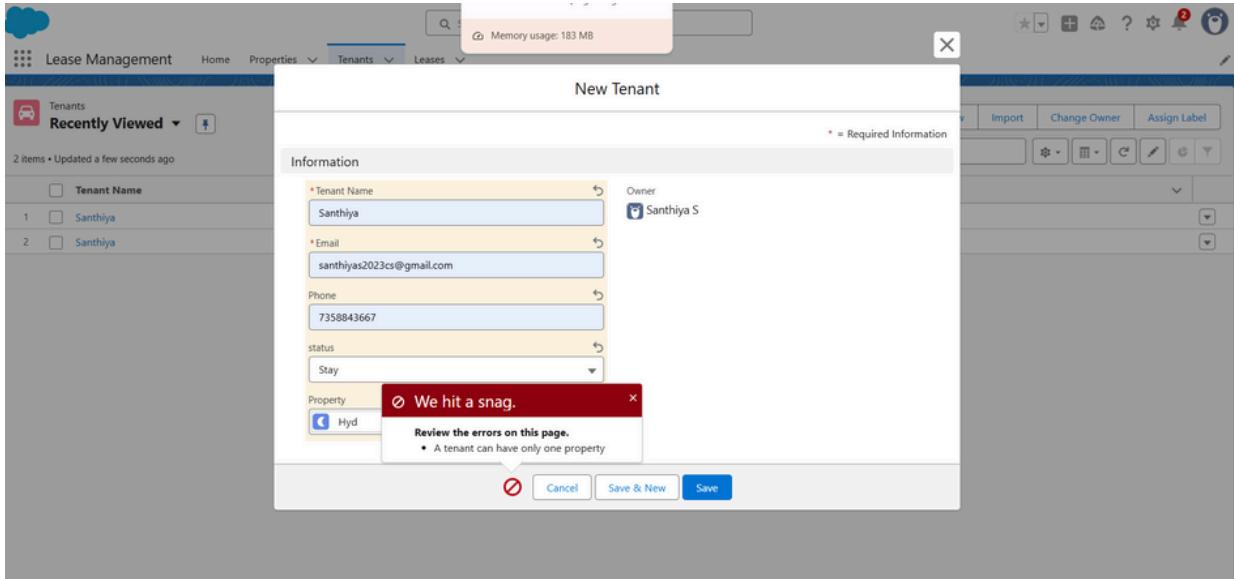
The screenshot shows a tenant record for "Aswini". The "Details" tab is selected. The tenant's name is "Aswini", email is "aswiniamaraadi15@gmail.com", phone is "(905) 223-5567", and property is "Imran". The record was created by "Sowmya Team" on 6/26/2025 at 6:05 AM and last modified by "Sowmya Team" on 6/26/2025 at 11:06 PM. A green banner indicates "Tenant was submitted for approval". The "Activity" sidebar shows no upcoming or overdue activities.



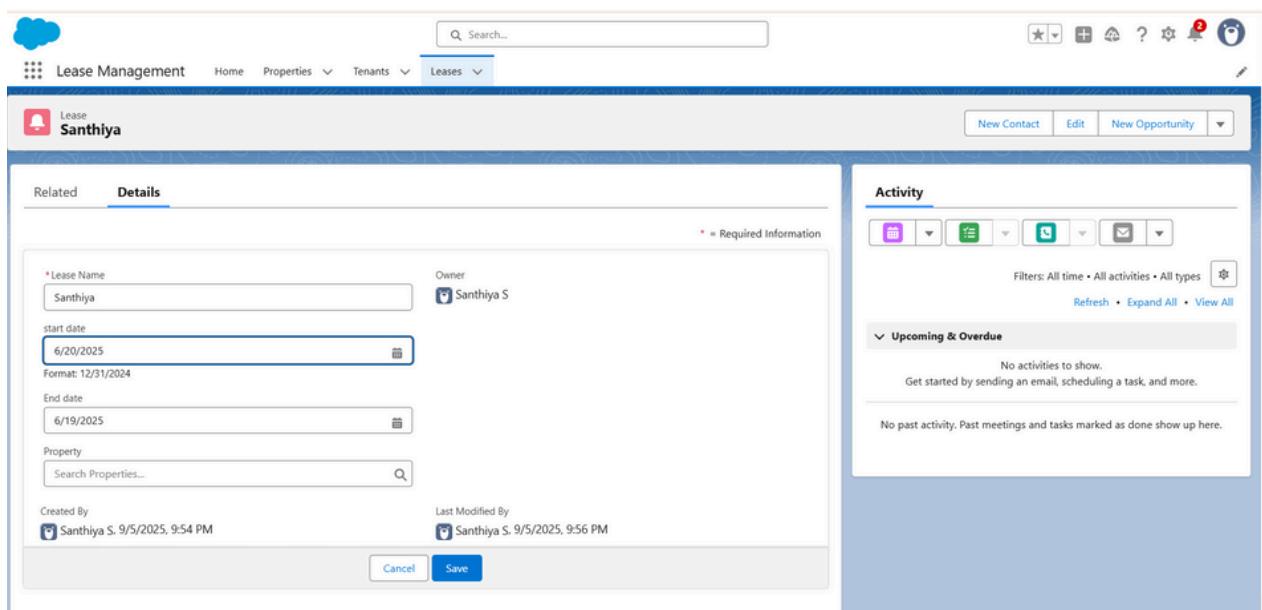
FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

- Trigger validation by entering duplicate tenant–property records



- Validation Rule checking



Lease Management

Lease Name: Santhiya

Start Date: 6/20/2025

End Date: 6/19/2025

Property: Search Properties...

Created By: Santhiya S. 9/5/2025, 9:54 PM

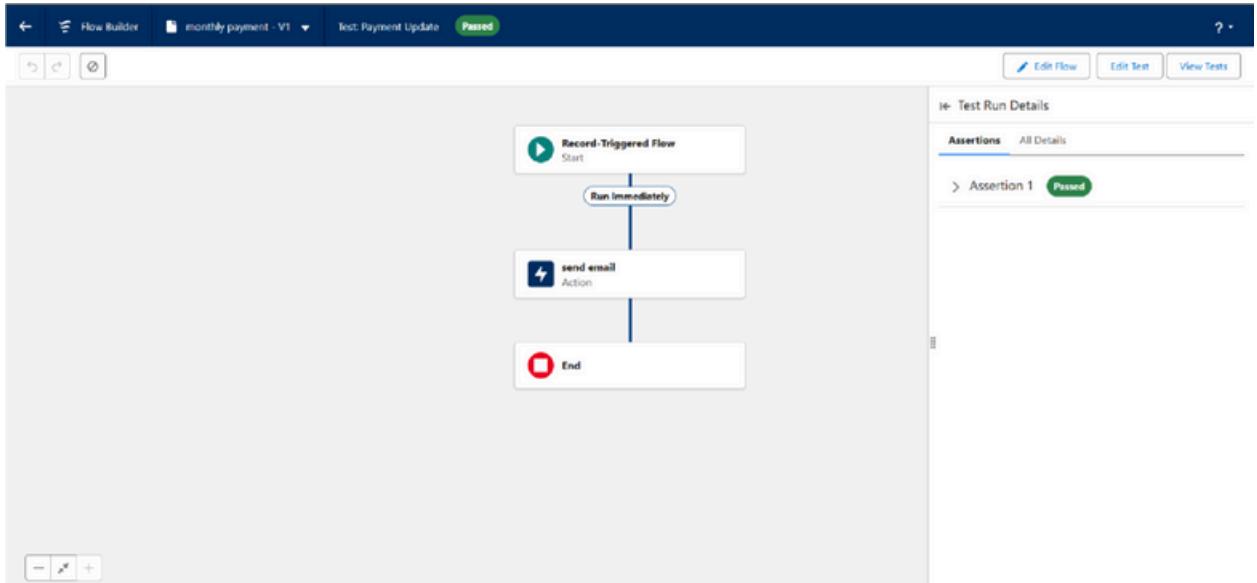
Last Modified By: Santhiya S. 9/5/2025, 9:56 PM

Activity

No activities to show.

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

- Test flows on payment update



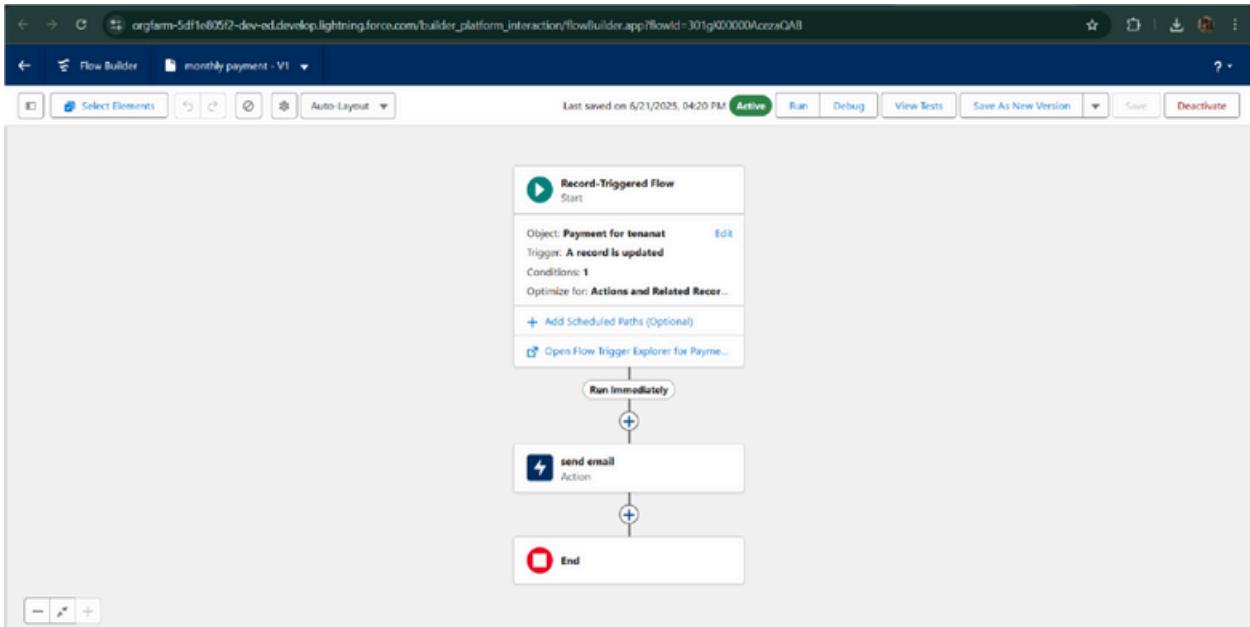
RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Custom Tabs' section. The left sidebar has 'Tabs' selected. The main area displays three sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. In 'Custom Object Tabs', there are four entries: 'Issue' (Tab Style: Keys), 'Payment' (Tab Style: Credit card), 'Lease' (Tab Style: Back), and 'Records' (Tab Style: Map). A note at the bottom says 'You can create new custom tabs to extend Salesforce functionality or to build new application functionality.' and 'Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.'

- Flow runs



- Trigger error messages

New Tenant

Information

- * Tenant Name: Santhiya
- * Email: santhiyas2023cs@gmail.com
- Phone: 7358843667
- Status: Stay
- Property: Hyd

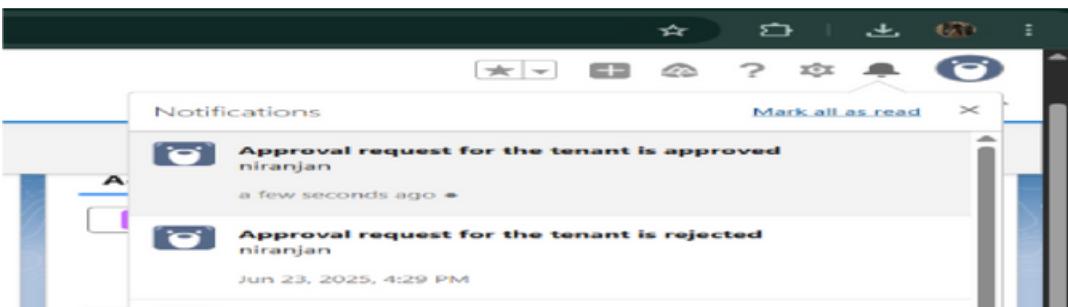
We hit a snag.

Review the errors on this page.

- A tenant can have only one property

Cancel Save & New Save

- Approval process notifications



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

- 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) 

Stu

String emailContent = 'I trust this email finds you well.
that the monthly rent is due. Your timely payment

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});
```

