

PROJECT TITLE

College Name: Government Arts and Science College, Sathyamangalam.

College Code: bru1021008

TEAM ID: NM2025TMID21285

TEAM MEMBERS:04

TeamMember: Nisha T

Email:2006nishastr@gmail.com

TeamMember1:Navanitha P

Email:navinavanitha68@gmail.com

TeamMember2:Prethika M

Email:prethikapraveena@g

mail.com

TeamMember3:Soundh

ar M

Email:soundhar32857@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.



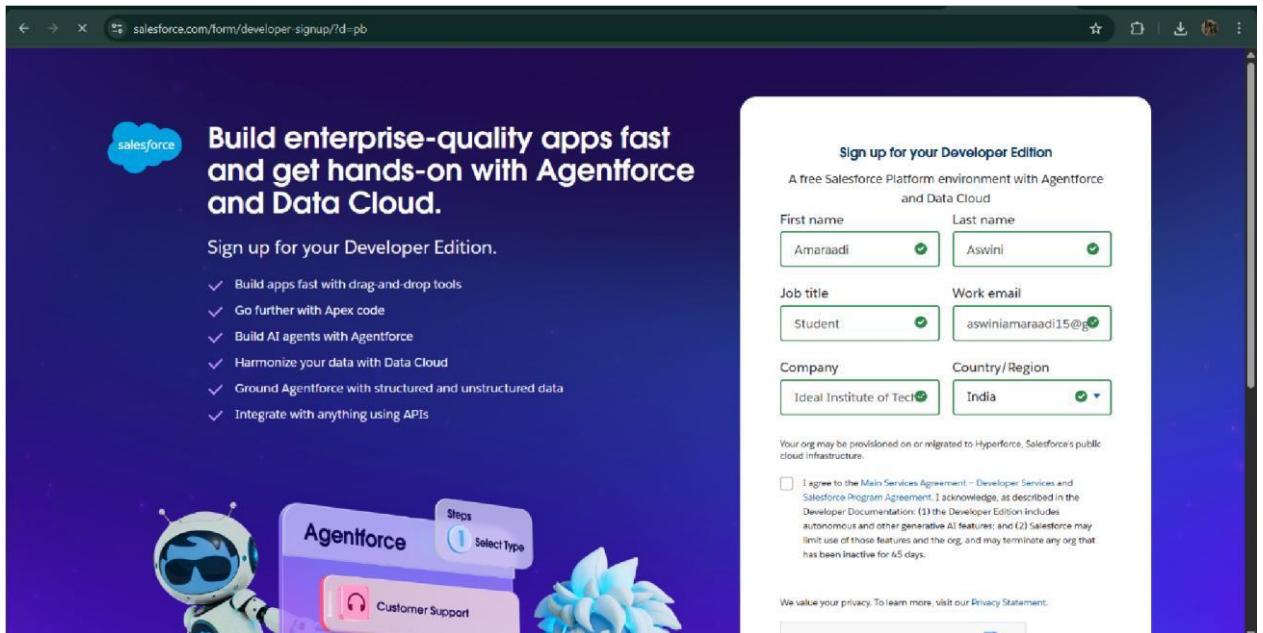
1.2 Purpose

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:

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



- Created objects: Property, Tenant, Lease, Payment

Details	
Description	API Name property__c Custom ✓ Singular Label property
	Plural Label property
	Enable Reports ✓ Track Activities ✓ Track Field History ✓ Deployment Status Deployed Help Settings Standard salesforce.com Help Window

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

This screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. The left sidebar lists various configuration options like Fields & Relationships, Page Layouts, and Record Types. The main 'Details' tab is selected, showing the API name 'Tenant__c', a custom label 'Tenant', and a plural label 'Tenants'. Under 'Enable Reports', 'Track Activities' and 'Track Field History' are checked. Deployment status is set to 'Deployed'. Help settings point to the standard Salesforce help window.

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

Edit Delete

This screenshot shows the Salesforce Object Manager interface for the 'lease' object. Similar to the 'Tenant' object, it has an API name 'lease__c', a custom label 'lease', and a plural label 'lease'. It also includes 'Enable Reports' settings for tracking activities and field history, and is currently deployed. The left sidebar shows the same range of configuration options as the 'Tenant' screen.

The screenshot shows the Salesforce Setup interface under the Object Manager. A sidebar on the left lists various configuration options like Fields & Relationships, Page Layouts, and Record Types. The main pane displays the 'Details' section for the 'Payment for tenant' object. It includes fields for API Name ('Payment_for_tenant__c'), Singular Label ('Payment for tenant'), and Plural Label ('Payment'). On the right, there are sections for 'Enable Reports' (checkboxes for Track Activities, Track Field History, Deployment Status, and Help Settings), and a link to the Standard Salesforce.com Help Window.

- Configured fields and relationships

The screenshot shows the Salesforce Setup interface under the Object Manager. The sidebar lists configuration options. The main pane shows the 'Fields & Relationships' section for the 'property' object, which contains 9 items. The table details the fields:

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)	<input checked="" type="checkbox"/>	
property	property__c	Lookup(property)	<input checked="" type="checkbox"/>	
property_Name	Name	Text(80)	<input checked="" type="checkbox"/>	
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

Setup > Object Manager

Payment for tenant

Fields & Relationships				
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 > Object Manager

lease

Fields & Relationships				
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

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

Lease Management
Application to efficiently handle the processes relate...

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

App Settings

[App Details & Branding](#)

[App Options](#)

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

Navigation Items

[User Profiles](#)

Navigation Items

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

Available Items

Type to filter list... [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

Up ▲ Down ▼

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

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

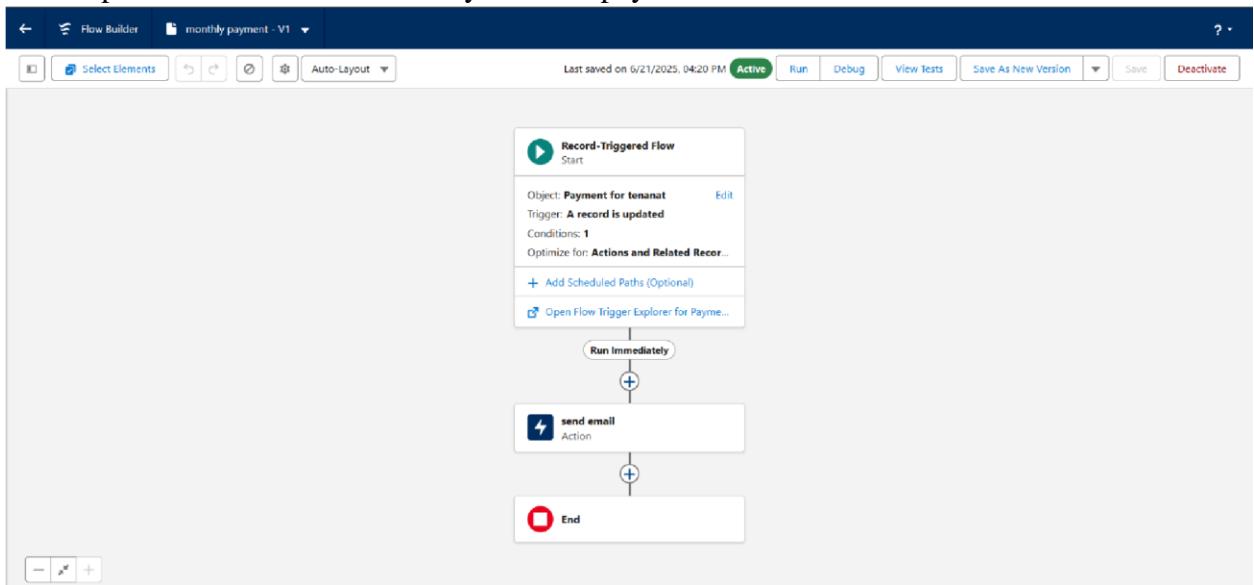
- System Administrator

A screenshot of a Salesforce Lightning interface. The top navigation bar shows the URL as orgId=arm-5d1fe805f2-dev-ed.lightning.force.com/lightning/o/Payment_for_tenant__c/list?filterName=Recent. The page title is "Lease Management" and the tab is "Payment". The main content area displays a list titled "Recently Viewed" with 5 items. The items are listed as follows:

	Payment Name
1	Rahul
2	Jack
3	Raj
4	Sam
5	Lahari

At the top right of the list view, there are buttons for "New", "Import", "Change Owner", and "Assign Label". Below the list, there is a search bar labeled "Search this list..." and various list management icons.

- 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 following details:

- Page Header:** orgfarm-5d1fe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK00000zTujValidationRules/page?address=%2F03dgK00000C8qPQAG%2Fe%3freURL%3D%2...
- Page Title:** Validation Rule Edit
- Object:** lease
- Fields:**
 - Rule Name: lease_end_date
 - Active: checked
 - Description: (empty)
- Error Condition Formula:**

```
End_date_c <= start_date_c
```

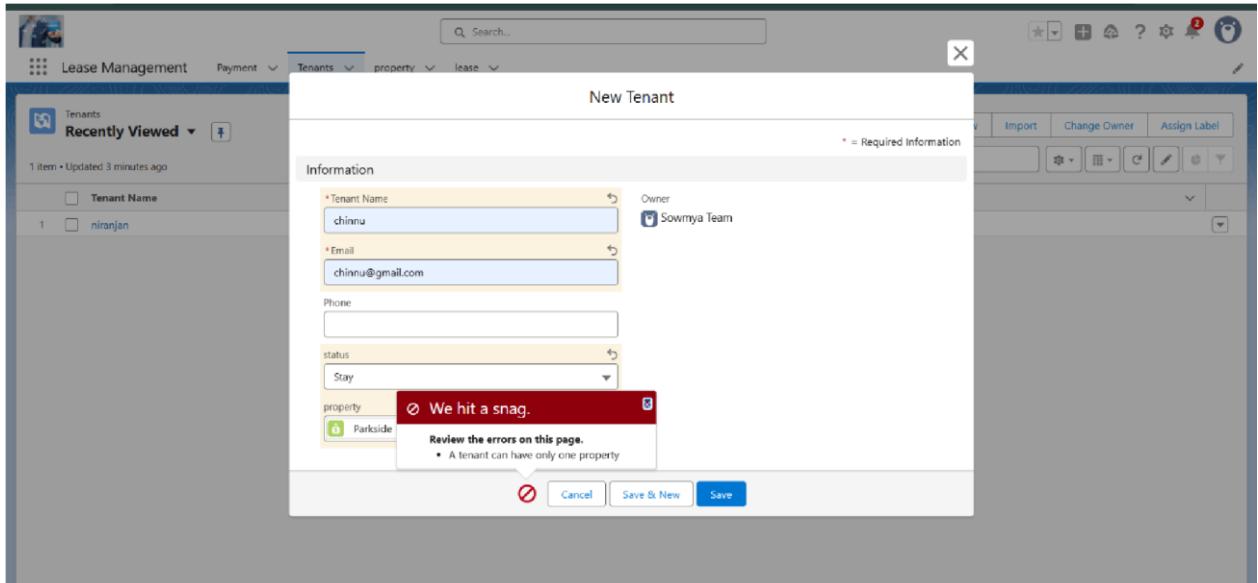
Example: Discount_Percent_C>30 More Examples...
Display an error if Discount is more than 30%
If this formula expression is true, display the text defined in the Error Message area
- Functions:**
 - All Function Categories
 - ABS
 - ACOS
 - ADDMONTHS
 - AND
 - ASCII
 - ASIN
- Buttons:**
 - Save
 - Save & New
 - Cancel
 - Check Syntax
- Help:** Quick Tips (Operators & Functions)

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

- Page Header:** orgfarm-5d1fe805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgK00000zTujValidationRules/03dgK00000C8qPQAG/view
- Page Title:** lease Validation Rule
- Object:** lease
- Fields:**

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	(empty)
Created By	Sowmya Team 6/19/2025, 5:37 AM
Modified By	Sowmya Team 6/26/2025, 7:47 AM
- Buttons:**
 - Edit
 - Close
- Help:** Help for this Page

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using 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     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20     for (Tenant__c tenant : tenants) {
21
22         String recipientEmail = tenant.Email__c;
23
24         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';
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30         email.setToAddresses(new String[]{recipientEmail});
31
32         email.setSubject(emailSubject);
33
34         email.setPlainTextBody(emailContent);
35 }
```

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

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. Under the 'Email' section, 'Classic Email Templates' is selected. A search result for 'Leave approved' is displayed. The 'Email Template Detail' section shows the following information:

Field	Value
Email Template Name	Leave approved
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Changed]
Description	
Created By	Sowmya Team, 6/20/2025, 1:06 AM
Modified By	Sowmya Team, 6/20/2025, 1:06 AM

The 'Email Template' preview area shows the following content:

```

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.
  
```

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. Under the 'Email' section, 'Classic Email Templates' is selected. A search result for 'tenant leaving' is displayed. The 'Email Template Detail' section shows the following information:

Field	Value
Email Template Name	tenant leaving
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	Sowmya Team [Changed]
Description	
Created By	Sowmya Team, 6/20/2025, 1:06 AM
Modified By	Sowmya Team, 6/20/2025, 1:06 AM

The 'Email Template' preview area shows the following content:

```

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 'Email' section selected. Under 'Email', 'Classic Email Templates' is chosen. A search bar at the top left shows 'email template'. The main content area displays a 'Text Email Template' named 'Leave rejected'. The 'Email Template Detail' section includes fields for Email Template Name (Leave rejected), Template Unique Name (Leave_rejected), Encoding (UTF-8), Author (Sowmya Team [Change]), Description (Created By Sowmya Team 6/20/2025, 1:11 AM), Available For Use (checked), Last Used Date, and Times Used. Below this is a preview pane titled 'Email Template' with a 'Plain Text Preview' containing the following text:

```
Subject : Leave rejected
Dear {Tenant_c_Name}.

I hope this email finds you well. Your contract has not ended. So we can't approve your leave.
your leave has rejected
```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Email', 'Classic Email Templates' is chosen. A search bar at the top left shows 'email template'. The main content area displays a 'Text Email Template' named 'Tenant Email'. The 'Email Template Detail' section includes fields for Email Template Name (Tenant Email), Template Unique Name (Tenant_Email), Encoding (UTF-8), Author (Sowmya Team [Change]), Description (Created By Sowmya Team 6/20/2025, 1:12 AM), Available For Use (checked), Last Used Date, and Times Used. Below this is a preview pane titled 'Email Template' with a 'Plain Text Preview' containing the following text:

```
Subject : Urgent: Monthly Rent Payment Reminder
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.
```

Email Template Detail

- Email Template Name: tenant payment
- Template Unique Name: tenant_payment
- Encoding: Unicode (UTF-8)
- Author: Sowmya Team [Change]
- Description: Created By: Sowmya Team | 6/20/2025, 1:13 AM
- Modified By: Sowmya Team | 6/20/2025, 1:13 AM

Email Template

Subject: Confirmation of Successful Monthly Payment

Plain Text Preview:

Dear {Tenant_c_Email__c},

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

- Approval Process creation

For Tenant Leaving:

Process Definition Detail

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description: Tenant: status equals Stay
- Entry Criteria: Tenant: status equals Stay
- Record Editability: Administrator ONLY
- Next Automated Approver Determined By:
- Allow Submitters to Recall Approval Requests:

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: Sowmya Team	Final Rejection

For Check for Vacant:

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

- Approval Processes** page.
- Tenant: check for vacant** process selected.
- Process Definition Detail** section:
 - Process Name: check for vacant
 - Unique Name: check_for_vacant
 - Description: Tenant: status EQUALS Leaving
 - Entry Criteria: Tenant: status EQUALS Leaving
 - Record Editability: Administrator ONLY
 - Next Automated Approver Determined By: Active (checked)
 - Approval Assignment Email Template: LeaveApproved
 - Initial Submitters: Tenant Owner
 - Created By: Sowmya Team
 - Modified By: Sowmya Team
 - Created Date: 6/20/2025, 3:18 AM
 - Modified Date: 6/26/2025, 11:02 PM
- Initial Submission Actions** section:
 - Action: Record Lock
 - Type: Description: Lock the record from being edited
 - Action: Email Alert
 - Type: Description: please approve my leave
- Approval Steps** section:

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

```

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

```

The screenshot shows the Salesforce Developer Console with the following details:

- trigger test on Tenant__c (before insert)**
- if(trigger.isInsert && trigger.isBefore){**
- testHandler.preventInsert(trigger.new);**
- Open** dialog (Entity Type: Triggers, Name: test)
- Logs**, **Tests**, **Checkpoints**, **Query Editor**, **View Status**, **Progress**, **Problems** tabs at the bottom.

Developer Console - Google Chrome
 orgfarm-5d1fe805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```
File Edit Debug Test Workspace Help < >
test.apex testHandler.apc MonthlyEmailScheduler.apc
Code Coverage Name: API Version: 64 Go To
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

Name Line Problem

Create an Apex Handler class

Developer Console - Google Chrome
 orgfarm-5d1fe805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```
File Edit Debug Test Workspace Help < >
test.apex testHandler.apc MonthlyEmailScheduler.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(exi
10     }
11
12
13
14     for (Tenant__c newTenant : newList) {
15
16
17         if (newTenant.Property__c != null) {
18
19             newTenantaddError('A
20
21
22     }
23 }
```

Entity Type Entity Name Namespace Related

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

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Developer Console - Google Chrome

orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

File Edit Debug Test Workspace Help

test.apex testHandler.apxc MonthlyEmailScheduler.apac

Code Coverage Name API Version 64 Go To

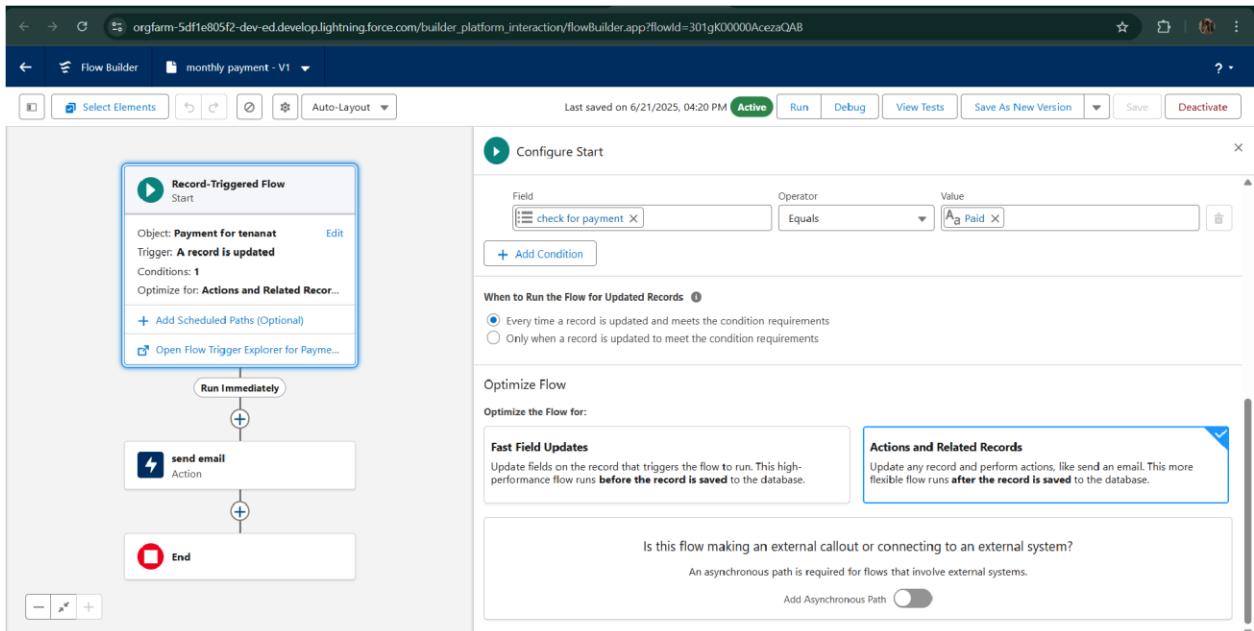
```

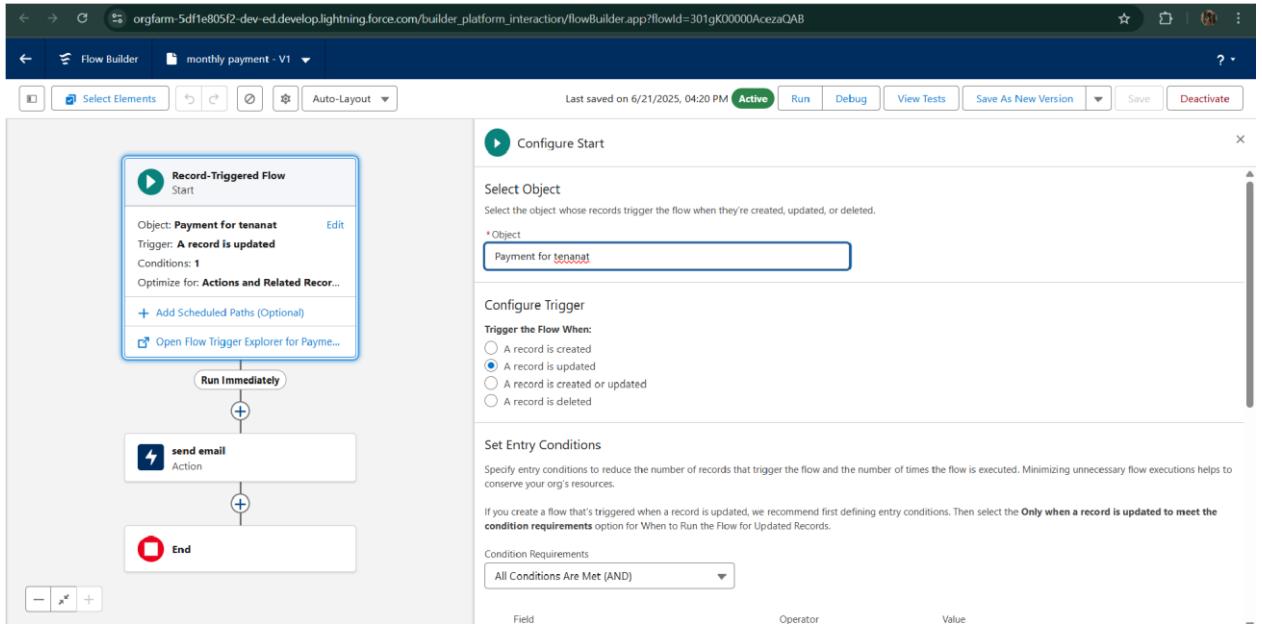
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
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 State 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
22             String recipientEmail = tenant.Email__c;

```

Entity Type

Entity Type	Entities	Related
Classes	testHandler	MonthlyEmailScheduler
Triggers		Email
Pages		← Tenant__c
Page Components		Object
Objects		← Tenant__c
Static Resources		References
Packages		

Developer Console - Google Chrome

org:farm-5d1fe805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```

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        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20        for (Tenant__c tenant : tenants) {
21
22            String recipientEmail = tenant.Email__c;
23
24            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.' ;
25
26            String emailSubject = 'Rent Remind: Monthly Rent Payment Due';
27
28            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30            email.setPlainTextContent(new String[]{recipientEmail});
31
32            email.setSubject(emailSubject);
33
34            email.setHtmlBody(emailContent);
35
36            messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
37
38        }
39
40    }
41
42

```

Page | Editor | Chatter | Home | Query Editor | Visualforce | Integration | Monitoring

Schedule Apex class

Setup | Home | Object Manager | Search Setup

Apex Classes

Apex Class Detail

Name	Namespace Prefix	Status
MonthlyEmailScheduler		Active
	Created By	Somya Team, 6/23/2025, 2:46 AM
	Code Coverage	0% (0/15)
	Last Modified By	Somya Team, 6/23/2025, 2:47 AM

Class Body

```

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        List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20        for (Tenant__c tenant : tenants) {
21
22            String recipientEmail = tenant.Email__c;
23
24            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.' ;
25
26            String emailSubject = 'Rent Remind: Monthly Rent Payment Due';
27
28            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30            email.setPlainTextContent(new String[]{recipientEmail});
31
32            email.setSubject(emailSubject);
33
34            email.setHtmlBody(emailContent);
35
36            messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
37
38        }
39
40    }
41
42

```

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

The screenshot shows a Salesforce Lightning page for a tenant named 'Aswini'. The 'Details' tab is selected. The form fields include:

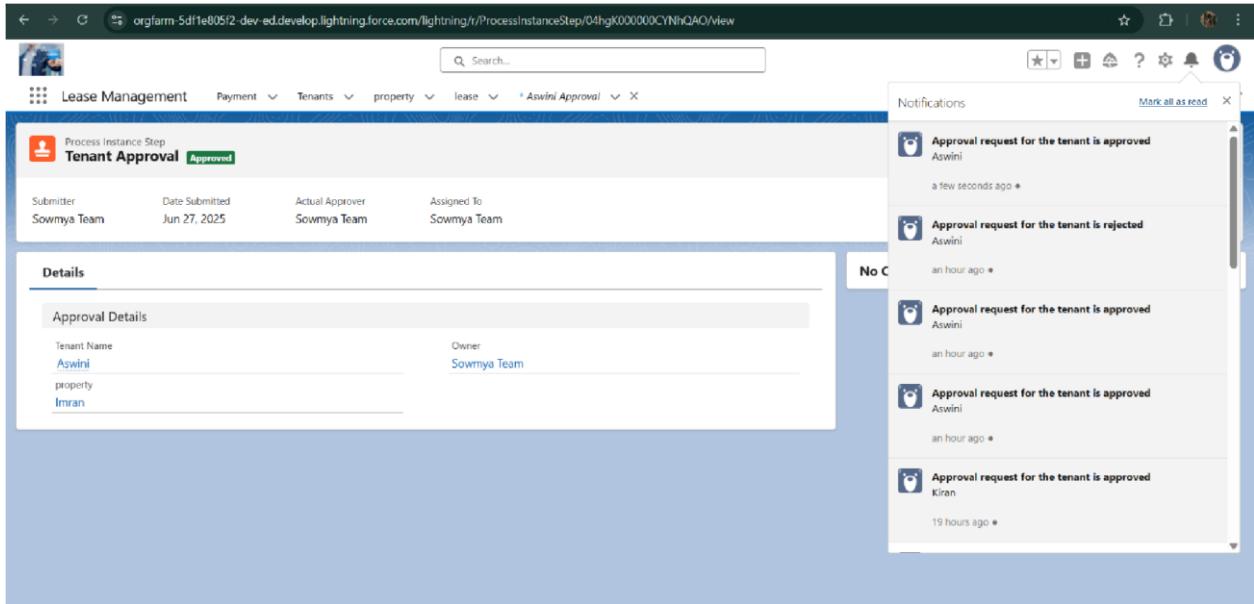
- * Tenant Name: Aswini
- * Email: aswiniamaraadi15@gmail.com
- Phone: (905) 223-5567
- Status: Leaving
- Property: Imran

Owner: Sowmya Team. Created By: Sowmya Team. Last Modified By: Sowmya Team.

Activity sidebar:

- New Contact, Edit, New Opportunity buttons.
- Activity section: Upcoming & Overdue (No activities to show).
- Filters: All time • All activities • All types.
- Actions: New Case, New Lead, Delete, Clone, Change Owner, Printable View, Submit for Approval, Edit Labels.

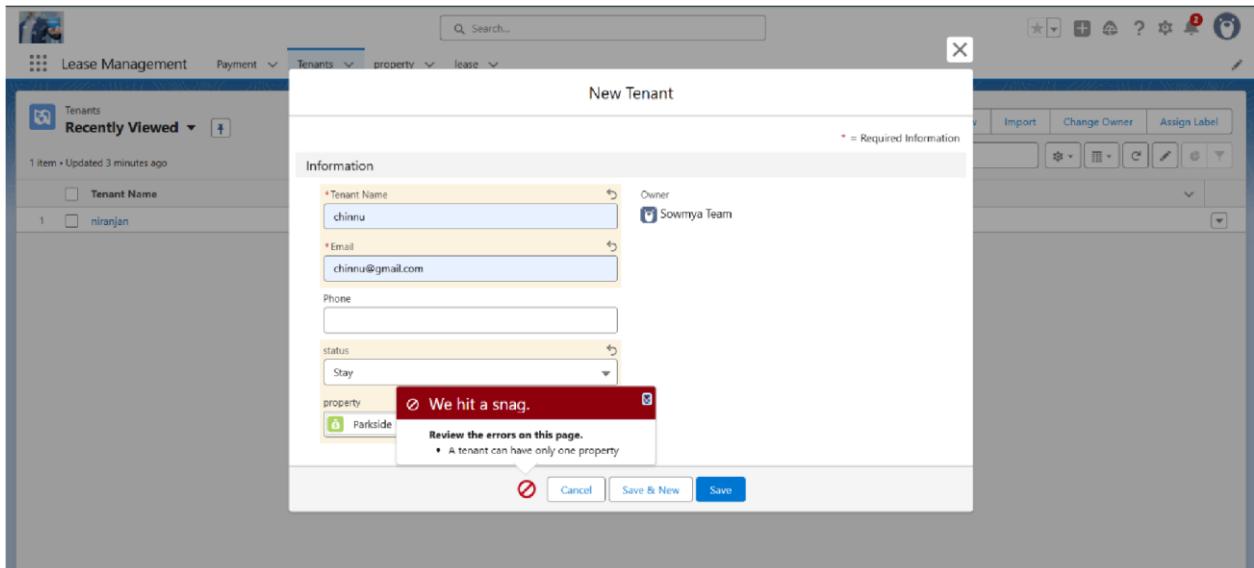
The screenshot shows the same Salesforce Lightning page for tenant 'Aswini'. A green success message at the top right says "Tenant was submitted for approval." The rest of the page is identical to the first screenshot, showing the tenant details and the activity sidebar.



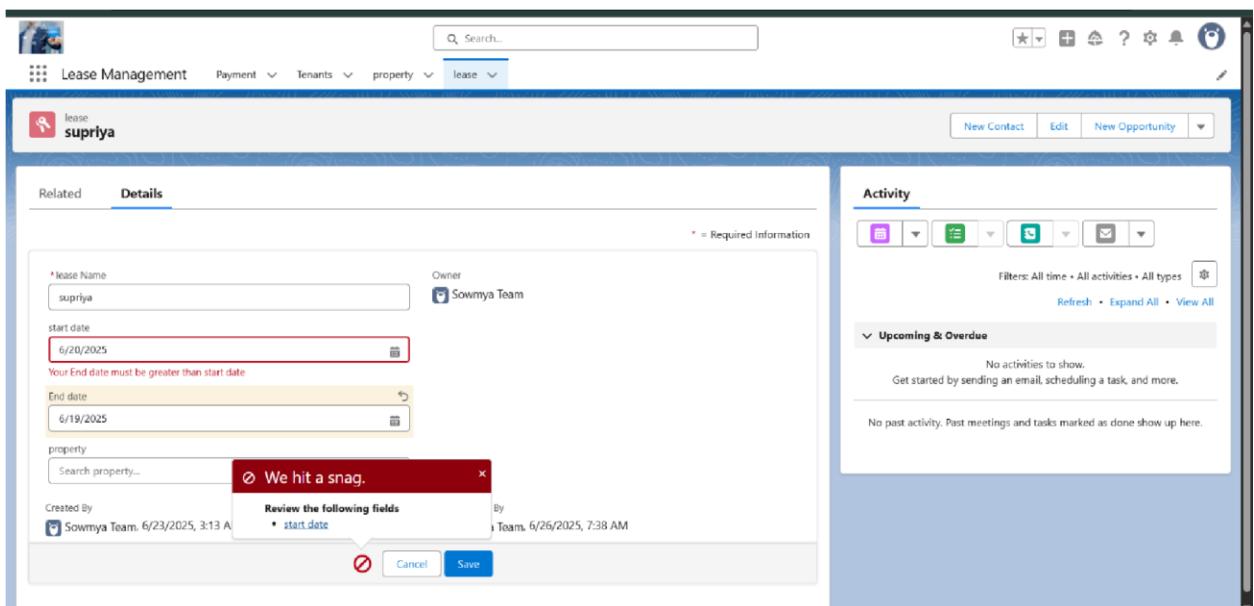
FUNCTIONAL AND PERFORMANCE TESTING

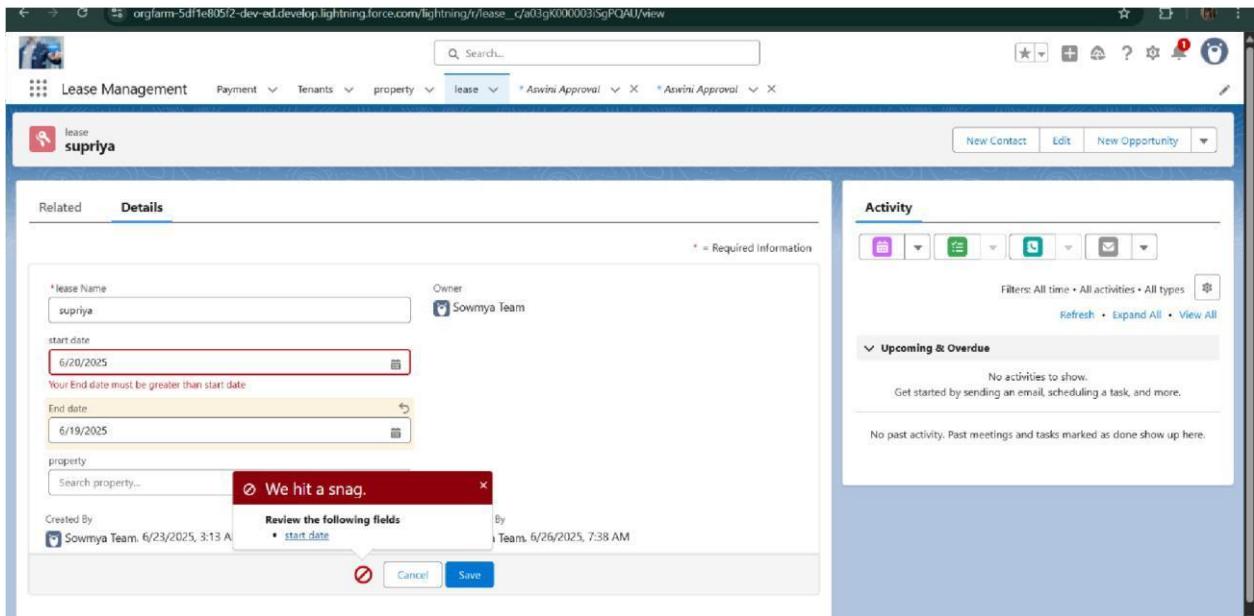
Performance Testing

- Trigger validation by entering duplicate tenant-property records

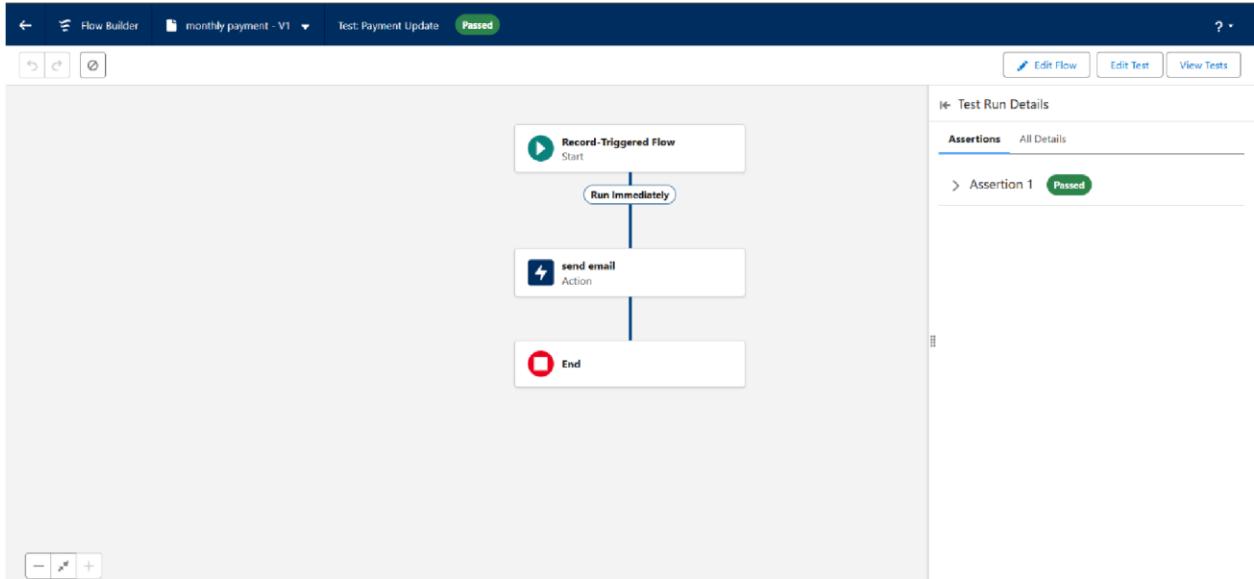


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows a CRM interface for 'Lease Management'. On the left, a 'Tenant' record for 'niranjan' is displayed with tabs for 'Related' and 'Details'. The 'Details' tab contains fields for Tenant Name ('niranjan'), Email ('niranjan1506@gmail.com'), Phone (empty), Status ('Stay'), and Property ('Parkside Lofts'). It also shows 'Created By' (Sowmya Team) and 'Last Modified By' (Sowmya Team). At the bottom are 'Cancel' and 'Save' buttons. On the right, a 'Notifications' sidebar lists several items: '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), and a 'New Guidance Center learning resource available' (Jun 20, 2025, 1:28 PM).

The screenshot shows a CRM interface for 'Lease Management'. On the left, the 'Approval History' section displays a table with 6+ rows. The columns are Step Name, Date, Status, and Assigned To. The data includes: Step 1 (Approved, 6/25/2025, 5:39 AM, Sowmya Team); Approval Request Submitted (Submitted, 6/25/2025, 5:39 AM, Sowmya Team); Step 1 (Rejected, 6/23/2025, 3:59 AM, Sowmya Team); Approval Request Submitted (Submitted, 6/23/2025, 3:58 AM, Sowmya Team); Step 1 (Approved, 6/23/2025, 3:55 AM, Sowmya Team); and Approval Request Submitted (Submitted, 6/23/2025, 3:55 AM, Sowmya Team). Below this is a 'View All' button. On the right, the 'Payment' section shows 2 entries: 'Jack' and 'Rahul'. A 'New' button is available for adding new payments. A sidebar on the right says 'Get started by sending an email, scheduling a task, and more.' and 'No past activity. Past meetings and tasks marked as done show up here.'

RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Tabs' section. It displays four categories of tabs:

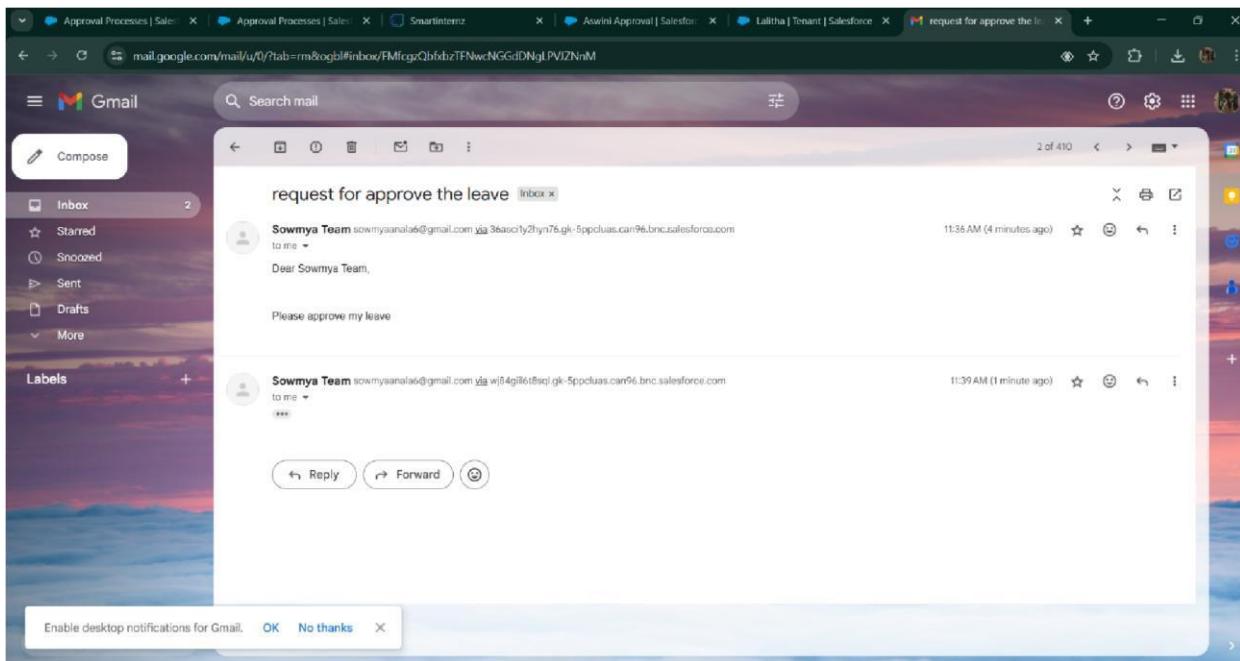
- Custom Object Tabs:** Shows tabs for 'Lease' (Tab Style: Keys), 'Payment' (Tab Style: Credit card), 'property' (Tab Style: Sack), and 'Tenant' (Tab Style: Map).
- Web Tabs:** Shows a message: "No Web Tabs have been defined".
- Visualforce Tabs:** Shows a message: "No Visualforce Tabs have been defined".

- Email alerts

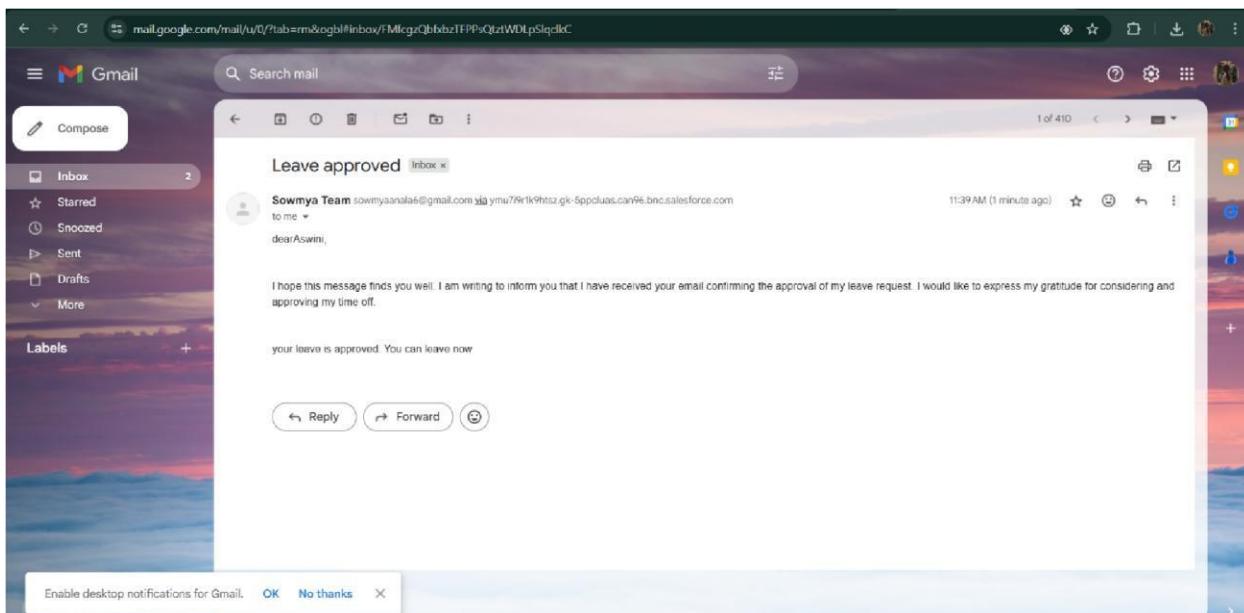
The screenshot shows the 'Approval History' section of the Lease Management application. It lists eight approval steps for a leave request:

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
2 Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team	Sowmya Team	leaving
3 Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team	Sowmya Team	Rejected
4 Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team	Sowmya Team	Leaving
5 Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team	Sowmya Team	Approved
6 Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team	Sowmya Team	leaving
7 Step 1	6/23/2025, 3:44 AM	Approved	Sowmya Team	Sowmya Team	Approval Approved
8 Approval Request Submitted	6/23/2025, 3:42 AM	Submitted	Sowmya Team	Sowmya Team	Leaving

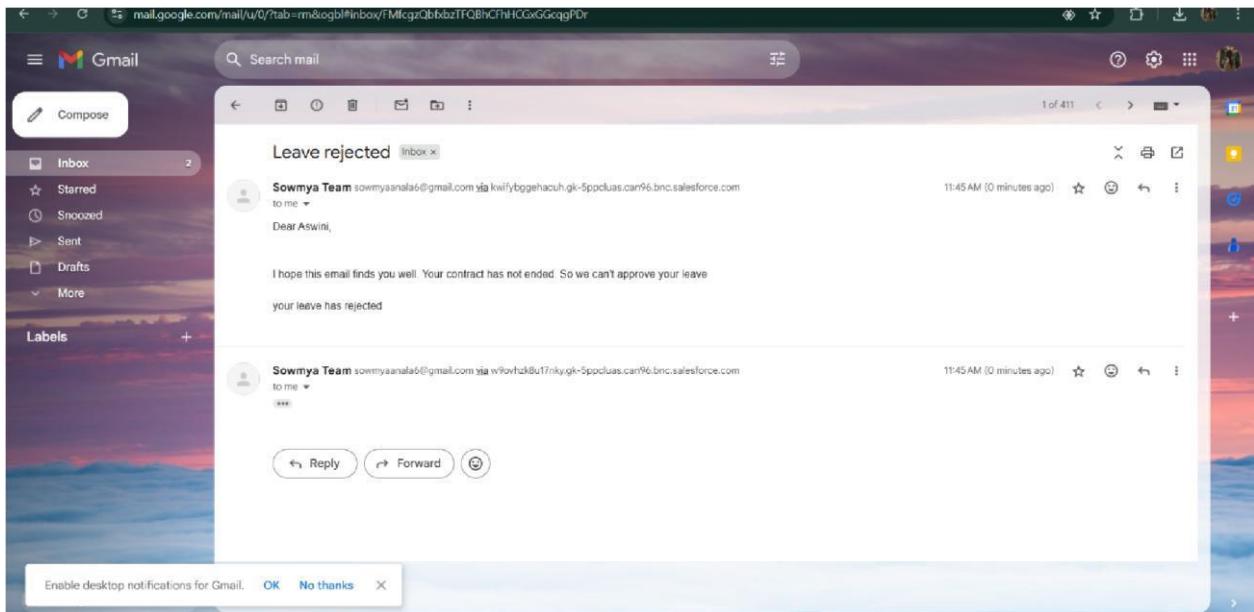
- Request for approve the leave



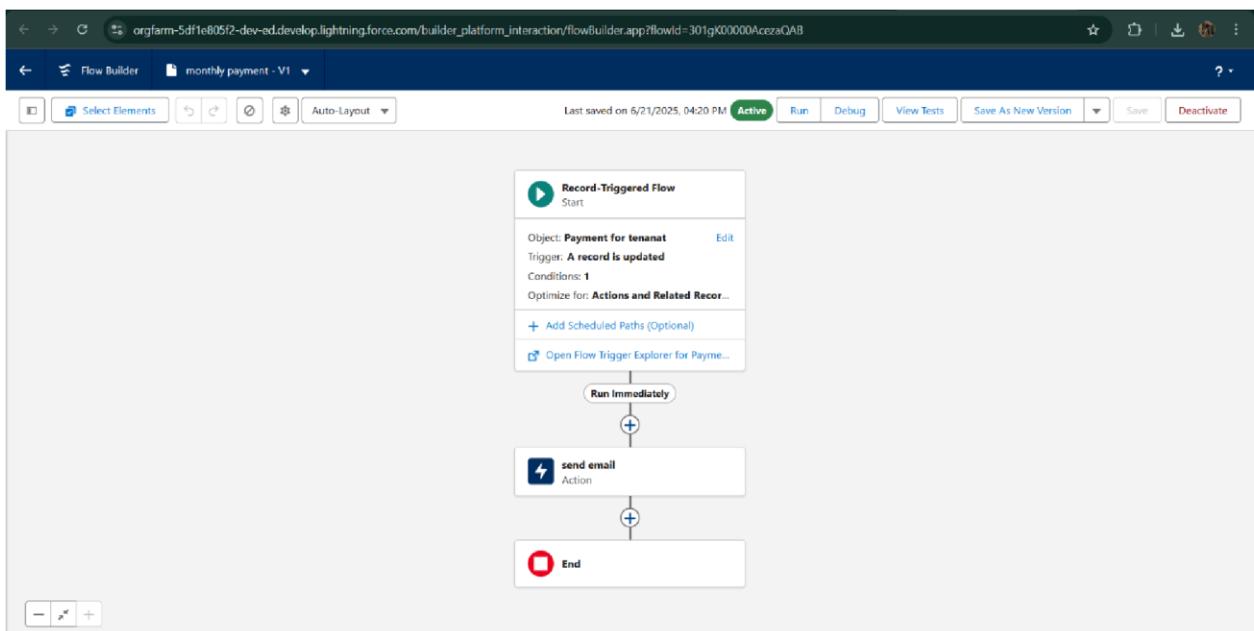
- Leave approved



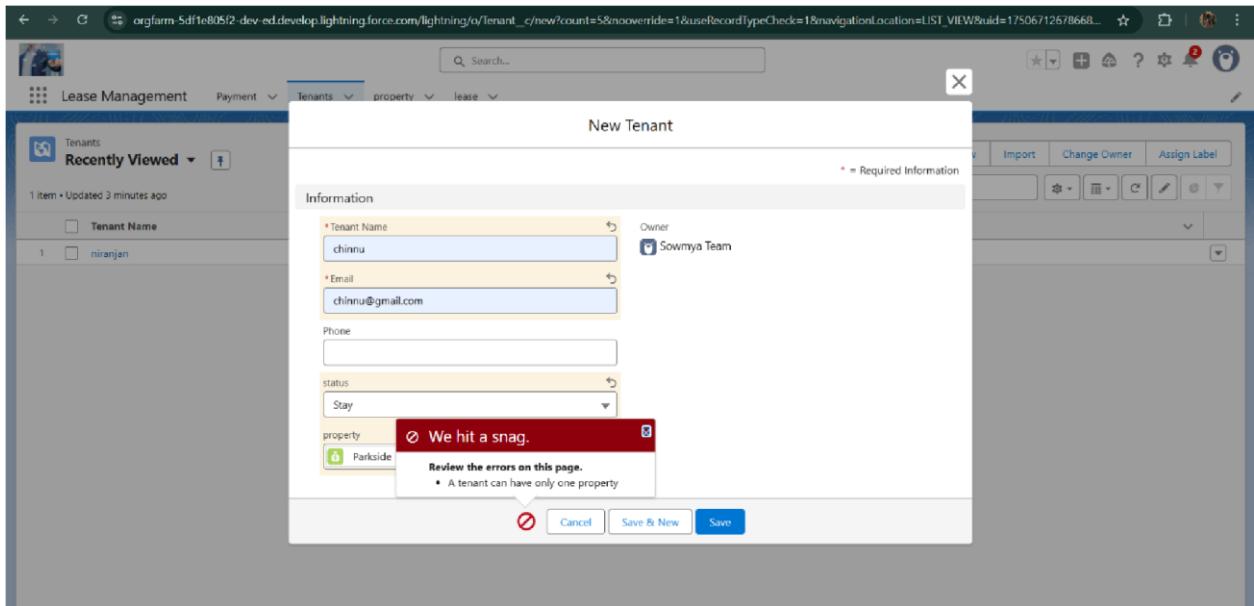
- Leave rejected



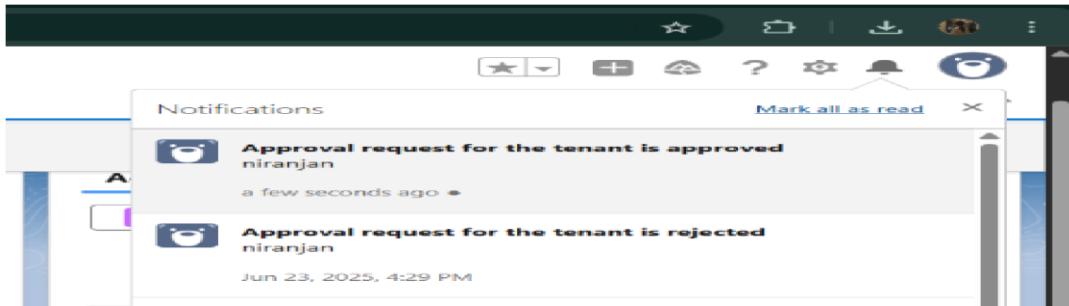
- Flow runs



- Trigger error messages



- 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) {  

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});  
    }  
}  
}
```