

LEASE MANAGEMENT

College Name: Sri Ramakrishna College of Arts and Science for Women

College Code: Bru28

TEAM ID: NM2025TMID24279

TEAM MEMBERS:

Team LeaderName: Harini T

Email: srcw2326j161@srcw.ac.in

Team Member 1: ANGEL PRIYA R

Email: srcw2326j106@srcw.ac.in

Team Member 2: ARYA B

Email: srcw2326j108@srcw.ac.in

Team Member 3: BHARATHI R

Email: srcw2326j109@srcw.ac.in

1.INTRODUCTION

1.1 Project Overview

The Lease Management project involves creating a system or application to efficiently handle the processes related to leasing real estate properties, equipment, or other assets. The goal is to streamline and automate various tasks associated with lease agreements, ensuring accurate

record-keeping, compliance with regulations, and effective communication between parties involved.



1.2 Purpose

The main objective of the project is to streamline the process of managing leases, tenants, and payments. To reduce manual intervention by automating approvals, reminders, and validations. To provide real-time insights into tenant and property data for better decision making. To improve customer experience through timely communication and efficient service.

DEVELOPMENT PHASE

Creating Developer Account:

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

- Created objects: Property, Tenant, Lease, Payment

orgfarm-07e6012109-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK0000022tgn/Details/view

Setup Home Object Manager

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

The screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. The left sidebar lists various setup categories like Fields & Relationships, Page Layouts, and Buttons, Links, and Actions. The main 'Details' section shows the API name 'Tenant_c', which is a custom object. It also displays the singular label 'Tenant' and the plural label 'Tenants'. On the right, under 'Enable Reports', 'Track Activities' and 'Track Field History' are checked. The deployment status is set to 'Deployed'. At the bottom, there are 'Edit' and 'Delete' buttons.

orgfarm-07e6012109-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK0000022tor/Details/view

Setup Home Object Manager

SETUP > OBJECT MANAGER lease

Details

Description

API Name lease_c

Custom ✓

Singular Label lease

Plural Label leases

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

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. The left sidebar lists various setup categories like Fields & Relationships, Page Layouts, and Buttons, Links, and Actions. The main 'Details' section shows the API name 'lease_c', which is a custom object. It also displays the singular label 'lease' and the plural label 'leases'. On the right, under 'Enable Reports', 'Track Activities' and 'Track Field History' are checked. The deployment status is set to 'Deployed'. At the bottom, there are 'Edit' and 'Delete' buttons.

Payment for tenant

Details

All Name: Payment_for_tenant_c
Custom: ✓
Singular Label: Payment for tenant
Plural Label: Payment

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

- Configured fields and relationships

Property

Fields & Relationships
4 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
property Name	Name	Text(80)		✓

Setup > OBJECT MANAGER
Payment for tenantat

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment for tenantat Name	Name	Text(80)		✓

Details

Fields & Relationships

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

Setup > OBJECT MANAGER
lease

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓

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

The screenshot shows the Salesforce Setup interface under Object Manager for the Tenant object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main content area is titled 'Fields & Relationships' and displays four items sorted by Field Label. The table columns include Field Label, Field Name, Data Type, Controlling Field, and Indexed status. The indexed status for the Owner field is checked.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Tenant Name	Name	Text(80)		✓

- Developed Lightning App with relevant tabs

The screenshot shows the 'New Lightning App' configuration screen. The left sidebar lists 'Apps' and 'Extensions'. The main form has two tabs: 'App Details' and 'App Branding'. In 'App Details', fields include 'App Name' (Lease ManagementDeveloper), 'Developer Name' (Lease_ManagementDeveloper), and 'Description' (Enter a description...). In 'App Branding', fields include 'Image' (Upload button), 'Primary Color Hex' (#0070D2), and 'Org Theme Options' (checkbox for using the app's image and color instead of the org's custom theme). Below the form is an 'App Launcher Preview' showing a blue icon with 'LM' and the app name. At the bottom, a progress bar indicates step 1 of 14, and a 'Next' button is visible.

<https://orgfarm-07e6012109-dev-ed.develop.lightning.force.com/lightning/setup/NavigationMenus/home>

New Lightning App

Search: home

Icon	Name	Description	Created	Type	Active
House	Digital Experiences Home		8/31/2025, 5:09 PM	Lightning	✓
Document	Home		8/31/2025, 5:09 PM	Classic	✓
House	Home		8/31/2025, 5:09 PM	Lightning	✓
House	Journey Home		8/31/2025, 5:09 PM	Lightning	✓
House	Mobile Home		8/31/2025, 5:09 PM	Lightning	✓

Back Next

<https://orgfarm-07e6012109-dev-ed.develop.lightning.force.com/lightning/setup/NavigationMenus/home>

New Lightning App

Search: system

Icon	Name	Description	Created	Type	Active
System	Salesforce API Only System Integrations		8/31/2025, 5:09 PM	Lightning	✓

Back Save & Finish

The screenshot shows a web-based application titled "Lease Management". The main navigation bar includes "Payment", "Tenants", "property", and "lease". Below the navigation, there is a search bar with placeholder text "Search..." and a "Recently Viewed" section. The "Recently Viewed" section lists 5 items, all under the "Payment" category, with the first item being "Rahul". There are buttons for "New", "Import", "Change Owner", and "Assign Label". A large, empty list area is visible below the header.

- Implemented Flows for monthly rent and payment success

The screenshot shows the "Flow Builder" interface. A "Record-Triggered Flow" is currently selected, starting with a "Run Immediately" action followed by an "Email Alert" action and finally an "End" action. The "Email Alert" step is highlighted with a blue border. On the right side of the screen, there is a sidebar titled "Email Alert" containing a search bar and a list of 3 results: "please approve my leave", "Tenant leaving", and "your request for leave is rejected". The "please approve my leave" result is expanded, showing its trigger and description.

- To create a validation rule to a Lease Object

orgfarm-07e6012109-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK0000022tor/ValidationRules/new

lease Validation Rule

Validation Rule Edit

Rule Name: lease_end_date

Active:

Description:

Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples...

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

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

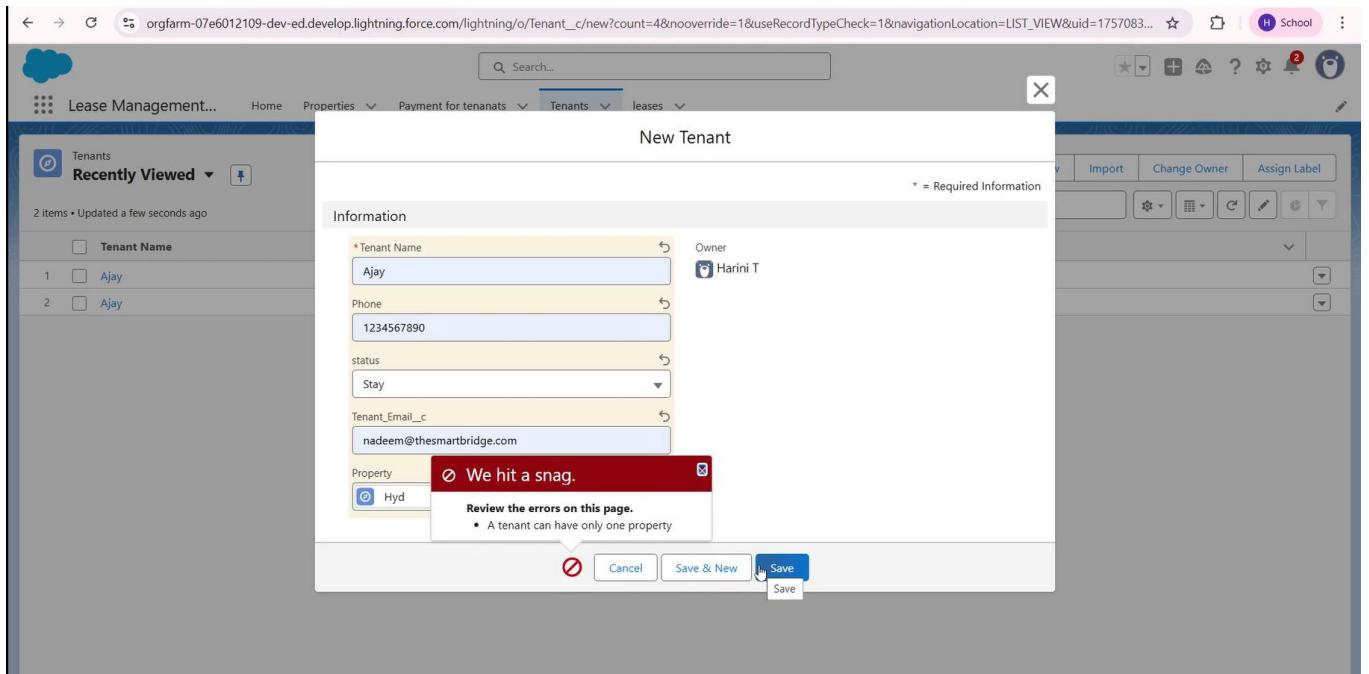
orgfarm-07e6012109-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/page?address=%2F03dgK000000aVN

lease Validation Rule

Validation Rule Detail

Rule Name	lease_end_date	Active	<input checked="" type="checkbox"/>
Error Condition Formula	End_date__c > start_date__c	Error Location	start date
Error Message	Your End date must be greater than start date		
Description			
Created By	Harini T 9/5/2025, 6:43 AM	Modified By	Harini T 9/5/2025, 6:43 AM

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

```

File -> New -> Apex Class -> MonthlyEmailScheduler.apex
Test Apex: MonthlyEmailScheduler.apex -> Run
Code Coverage: None -> API Version: 44.0

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 'Classic Email Templates' page selected. The template 'Leave approved' is displayed. The 'Email Template Detail' section shows the following details:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	'Leave approved'
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	Harini.T [Change]
Description	
Created By	Harini.T 9/5/2025, 6:46 AM
Modified By	Harini.T 9/5/2025, 6:46 AM

The 'Email Template' section shows the subject 'Leave approved' and the plain text preview:

```

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 'Classic Email Templates' page selected. The template 'tenant leaving' is displayed. The 'Email Template Detail' section shows the following details:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant leaving
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	Harini.T [Change]
Description	
Created By	Harini.T 9/5/2025, 6:45 AM
Modified By	Harini.T 9/5/2025, 6:45 AM

The 'Email Template' section shows the subject 'request for approve the leave' and the plain text preview:

```

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

Text Email Template Leave rejected

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Harini T [Change]
Description	
Created By	Harini.T 9/5/2025, 6:47 AM
Modified By	Harini.T 9/5/2025, 6:47 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

Help for this Page

Setup Home Object Manager

Search Setup

Classic Email Templates

Text Email Template Tenant Email

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Harini T [Change]
Description	
Created By	Harini.T 9/5/2025, 6:48 AM
Modified By	Harini.T 9/5/2025, 6:48 AM

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.

This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding & outlined in

Send Test and Verify Merge Fields

Help for this Page

Classic Email Templates

Auto populated

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Auto populated
Template Unique Name	Auto_populated
Encoding	Unicode (UTF-8)
Author	Harini T [Change]
Description	
Created By	Harini T 9/5/2025, 6:49 AM
Modified By	Harini T 9/5/2025, 6:49 AM

Email Template

Plain Text Preview

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:

Approval Processes

Tenant: TenantApproval

Process Definition Detail

Process Name	TenantApproval	Active
Unique Name	TenantApproval	Next Automated Approver Determined By
Description		
Entry Criteria	Tenant__c.status EQUALS Stay	
Record Editability	Administrator ONLY	Allow Submitters to Recall Approval Requests
Approval Assignment Email Template		
Initial Submitters	Tenant Owner	
Created By	Sowmya Team 6/23/2025, 3:41 AM	Modified By Sowmya 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: Sowmya Team	Final Rejection

For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The process definition detail section includes:

- Process Name:** check for vacant
- Unique Name:** check_for_vacant
- Description:** Tenant: check for vacant
- Entry Criteria:** Tenant: status NOT EQUAL TO Leaving
- Record Editability:** Administrator ONLY
- Approval Assignment Email Template:** Initial Submitters set to Tenant Owner
- Created By:** Harini.T, 9/5/2025, 6:56 AM
- Modified By:** Harini.T, 9/5/2025, 7:29 AM

The 'Initial Submission Actions' section contains two items: 'Record Lock' (Description: Lock the record from being edited) and 'Email Alert' (Description: please approve my leave). The 'Approval Steps' section indicates 'You have not yet defined any approval steps'.

● Apex Trigger

Create an Apex Trigger

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'test' open. The trigger code is:

```

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

```

A modal window titled 'Open' is overlaid on the code editor, listing various entities like Classes, Triggers, Pages, etc., with 'test' selected. The modal has tabs for 'Entity Type', 'Entity', and 'Related'.

Developer Console - Google Chrome
 orgfarm-5d1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

test.apxc [testHandler.apxc MonthlyEmailScheduler.apxc]

Code Coverage: None 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 }

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

Create an Apex Handler class

Developer Console - Google Chrome
 orgfarm-5d1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

test.apxc [testHandler.apxc MonthlyEmailScheduler.apxc]

Code Coverage: None 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       newTenantaddError('A t
19
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

```

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         for (Tenant__c newTenant : newList) {
13
14             if (newTenant.Property__c != null &amp; existingPropertyIds.contains(newTenant.Property__c)) {
15
16                 newTenantaddError('A tenant can have only one property');
17
18             }
19
20         }
21
22     }
23

```

● FLOWS

Record-Triggered Flow Start

Object: Payment for tenant Trigger: A record is updated

Conditions: 1
Optimize for: Actions and Related Records

Configure Start

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)

Field	Operator	Value
check for payment	Equals	A Paid

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

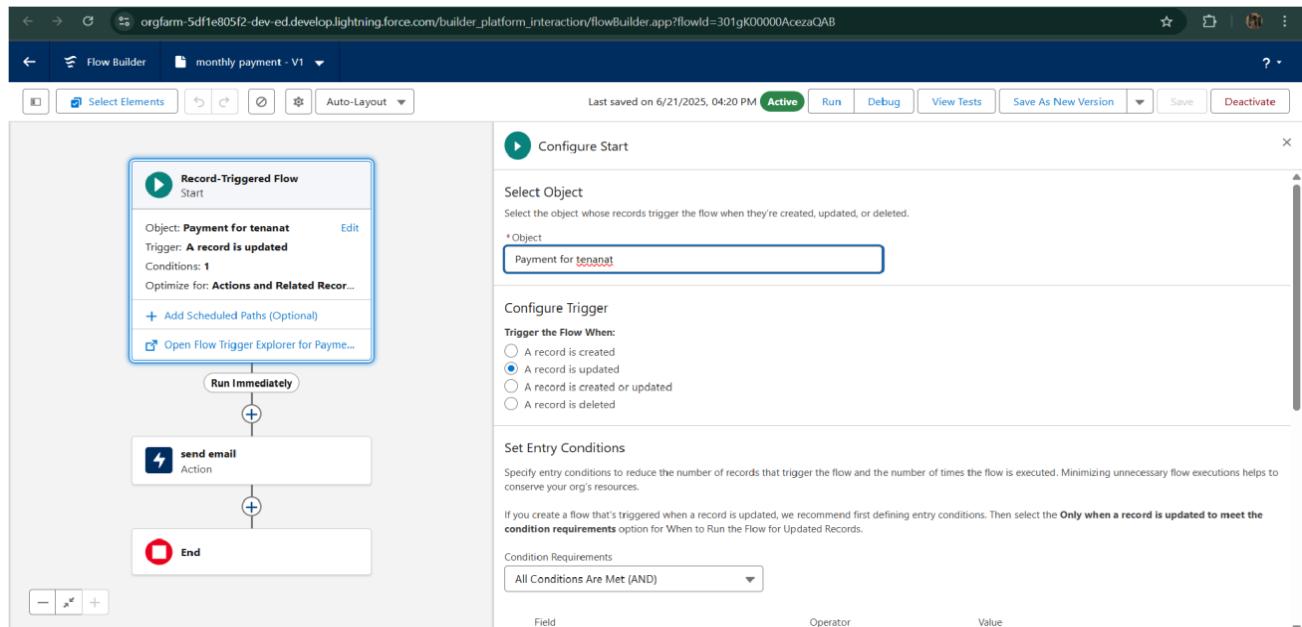
Optimize the Flow for:

- Fast Field Updates
- Actions and Related Records

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.



- Schedule class:
Create an Apex Class

The screenshot shows the Salesforce Developer Console. The code editor contains an Apex class named "MonthlyEmailScheduler" which implements the "Schedulable" interface. The class has a scheduled method "execute" that sends monthly emails if the current day is 1. The code also includes a static method "sendMonthlyEmail" that queries for tenants and sends emails to their recipient emails. A dependency browser window is overlaid on the code editor, showing the relationships between classes like "testHandler", "MonthlyEmailScheduler", "Email", "Tenant__c", and "CustomField__c".

```

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;
23

```

```

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 a positive living environment for all.';
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            email.setSubject(emailSubject);
31            email.setPlainTextBody(emailContent);
32
33            Messaging.SingleEmailMessage[] messages = new Messaging.SingleEmailMessage[]{email};
34
35            Messaging.sendEmail(messages);
36        }
37    }
38
39 }
40

```

Schedule Apex class

The screenshot shows the Salesforce Setup interface. In the top navigation bar, the user is in the Apex Classes section under Custom Code. The main content area displays the Apex Classes page with the following details:

- Apex Classes** heading.
- A message stating "Percent of Apex Used: 0.03%" and "You are currently using 1,674 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization."
- A "Estimate your organization's code coverage" button.
- A "Compile all classes" button.
- A "View" dropdown set to "All" with a "Create New View" option.
- A table listing two Apex classes:

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	MonthlyEmailScheduler		64.0	Active	1,143	Harmi T	9/5/2025, 9:42 AM
Edit Del Security	testHandler		64.0	Active	584	Harmi T	9/5/2025, 7:45 AM
- Dynamic Apex Classes** section with a note: "Dynamic Apex extends your programming reach by interacting with Lightning Platform components."
- A table showing no records displayed: "No records to display." The header for this table is "Class Name" and "Namespace Prefix".

Lease Management Payment Tenants property lease

Tenant: Aswini

Related Details * = Required Information

Tenant Name: Aswini Owner: Sowmya Team

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

New Contact Edit New Opportunity

Activity

New Case New Lead Delete Clone Change Owner Refresh Printable View Submit for Approval Edit Labels

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

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

Lease Management Payment Tenants property lease

Tenant: Aswini

Related Details * = Required Information

Tenant was submitted for approval.

Tenant Name: Aswini Owner: Sowmya Team

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

New Contact Edit New Opportunity

Activity

Filters: All time • All activities • All types Refresh • Expand All • View All

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

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

Lease Management

Process Instance Step Tenant Approval Approved

Submitter: Sowmya Team Date Submitted: Jun 27, 2025 Actual Approver: Sowmya Team Assigned To: Sowmya Team

Details

Approval Details

Tenant Name: Aswini Owner: Sowmya Team

No C

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

FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

- * Trigger validation by entering duplicate tenant-property records

Lease Management

Tenants Recently Viewed

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.

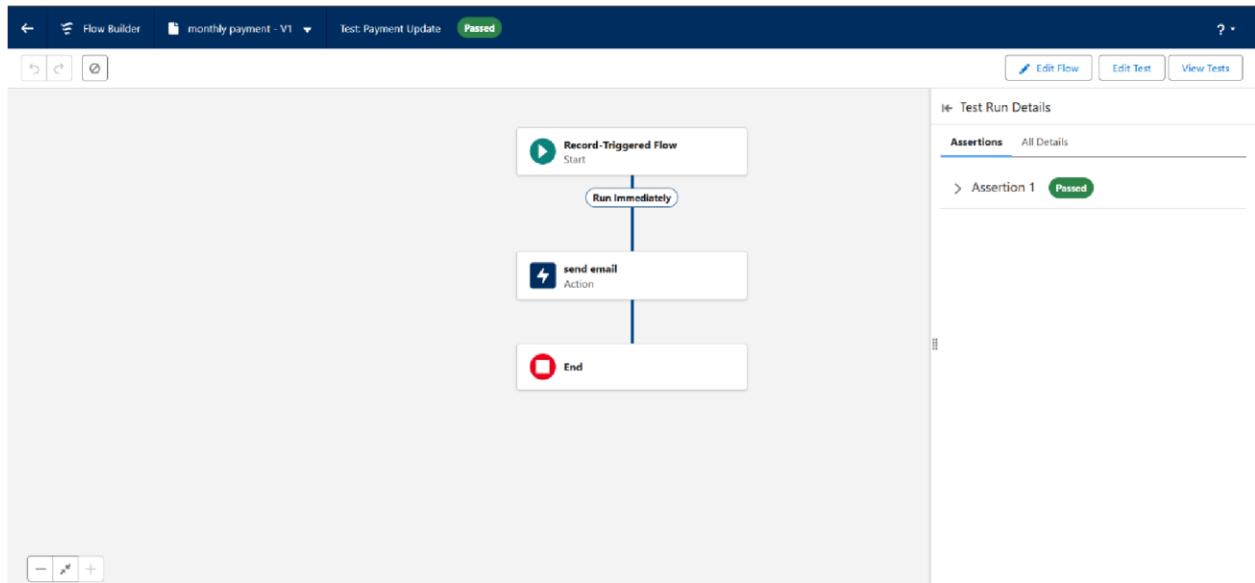
Cancel Save & New Save

- Validation Rule checking

The screenshot shows a Salesforce Lease Management page for a lease named "supriya". The "Details" tab is selected. A validation error message "Your End date must be greater than start date" is displayed over the "End date" field, which is set to "6/19/2025". The "start date" field is also highlighted in red. A modal window titled "We hit a snag." provides instructions to "Review the following fields" and lists "start_date". The "Save" button is disabled.

This screenshot is identical to the one above, showing the same lease record "supriya" in the "Lease Management" section. The "start date" field is again highlighted in red with the validation message "Your End date must be greater than start date". A modal window titled "We hit a snag." lists the validation rule "start_date". The "Save" button remains disabled.

- Test flows on payment update



- Approval process validated through email alerts and status updates

The screenshot shows the Lease Management system interface. On the left, a tenant profile for "niranjan" is displayed with the "Details" tab selected. The profile includes fields for Tenant Name (niranjan), Email (niranjan1506@gmail.com), Phone, Status (Stay), and Property (Parkside Lofts). On the right, a "Notifications" sidebar lists several approval requests:

- 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)

The screenshot shows the Salesforce Lightning Experience interface. At the top, there's a navigation bar with 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. A search bar is at the top right. Below the navigation, it says 'Tenant: niranjan'. On the left, there are two tabs: 'Approval History (6+)' and 'Payment (2)'. The 'Approval History' tab lists six steps with details like date, status, and assigned team. The 'Payment' tab lists two entries: 'Jack' and 'Rahul'. To the right, there's a sidebar with a message: 'Get started by sending an email, scheduling a task, and more.' and a note: '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. The URL is 'orgfarm-5df1e805f2-dev-ed.lightning.force.com/lightning/setup/CustomTabs/page?address=%2Fsetup%2Fu%2Fcustomtabs.jsp%3Fsetupid%3DCustomTabs%26refURL%3D%252Fsetu...'. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'Custom Tabs' and contains sections for 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. In the 'Custom Object Tabs' section, there's a table with four rows:

Action	Label	Tab Style	Description
Edit Del	Lease	Keys	
Edit Del	Payment	Credit card	
Edit Del	property	Sack	
Edit Del	Tenants	Map	

The 'Web Tabs' and 'Visualforce Tabs' sections both indicate 'No Web Tabs have been defined' and 'No Visualforce Tabs have been defined' respectively.

- Email alerts

Approval History

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

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

request for approve the leave

Sowmya Team sowmyaanala6@gmail.com via 36ascity2hy76.gk-5ppcluas.car96.bnc.salesforce.com to me

Dear Sowmya Team,

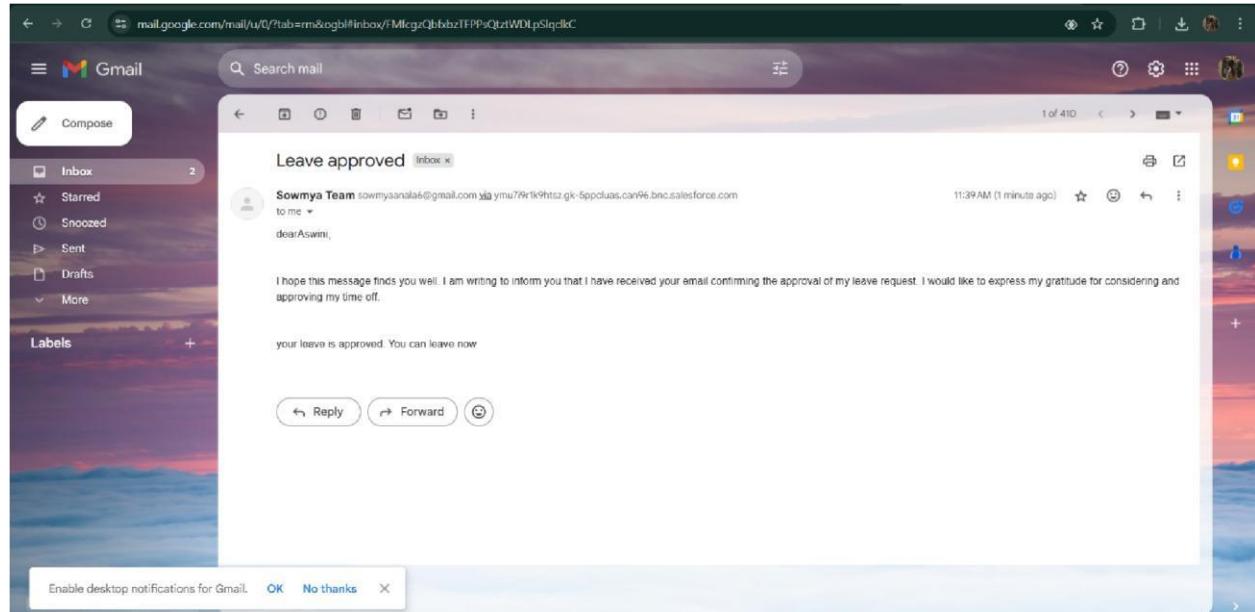
Please approve my leave

Sowmya Team sowmyaanala6@gmail.com via wj04gj6ifsqj.gk-5ppcluas.car96.bnc.salesforce.com

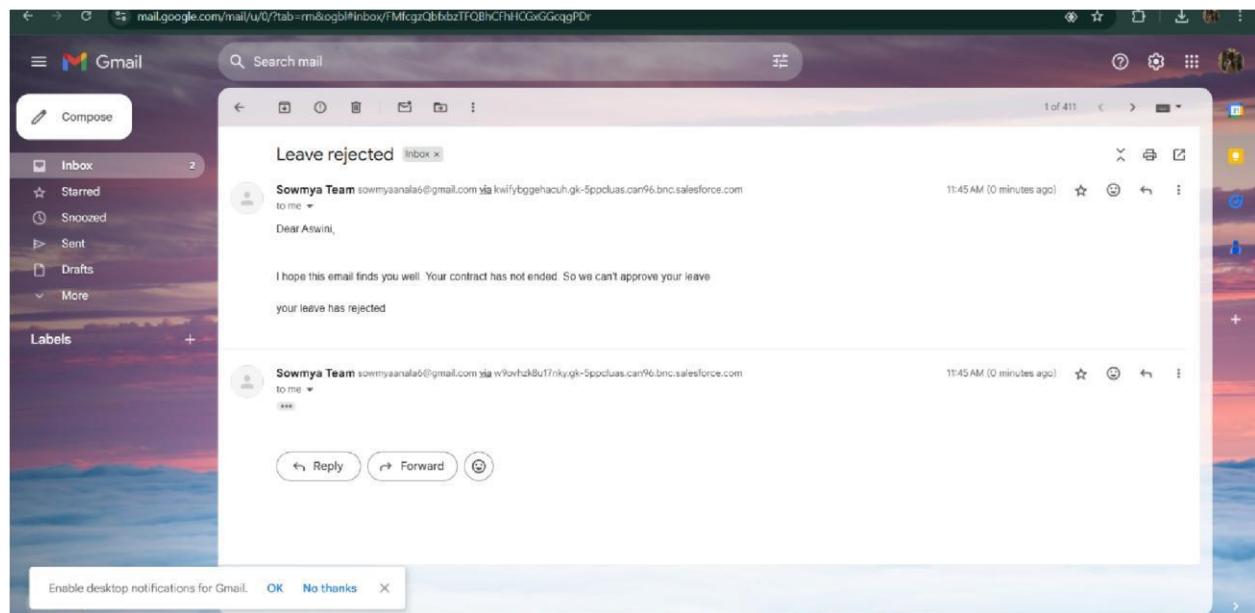
11:38 AM (4 minutes ago)

Enable desktop notifications for Gmail. OK No thanks

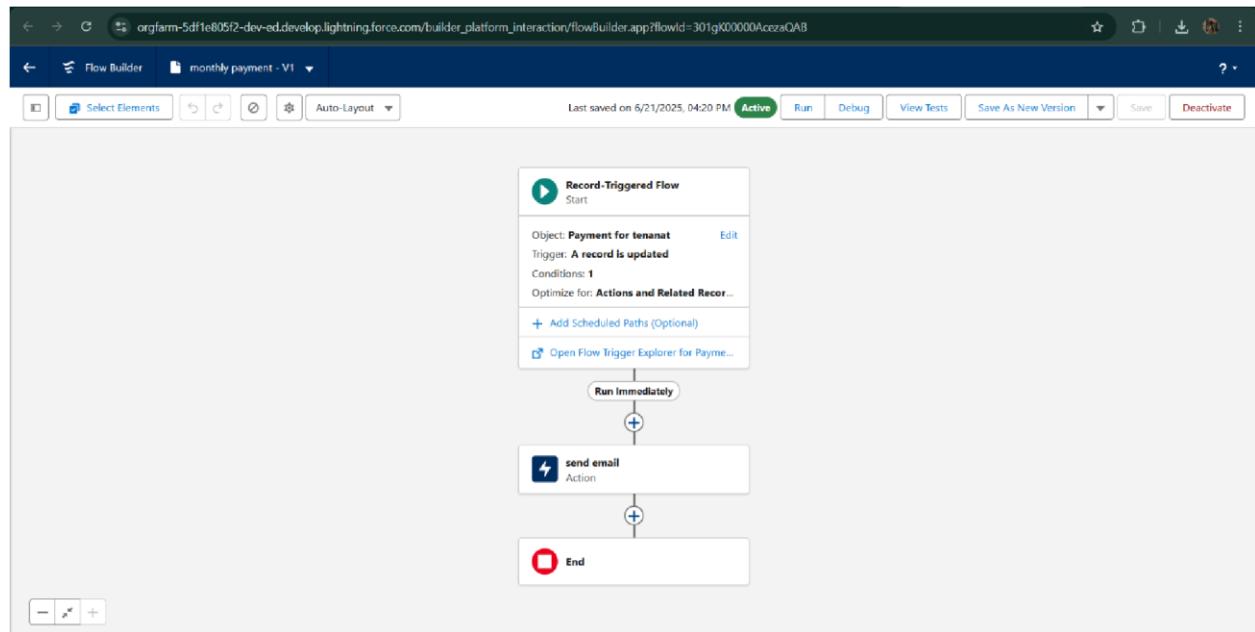
- Leave approved



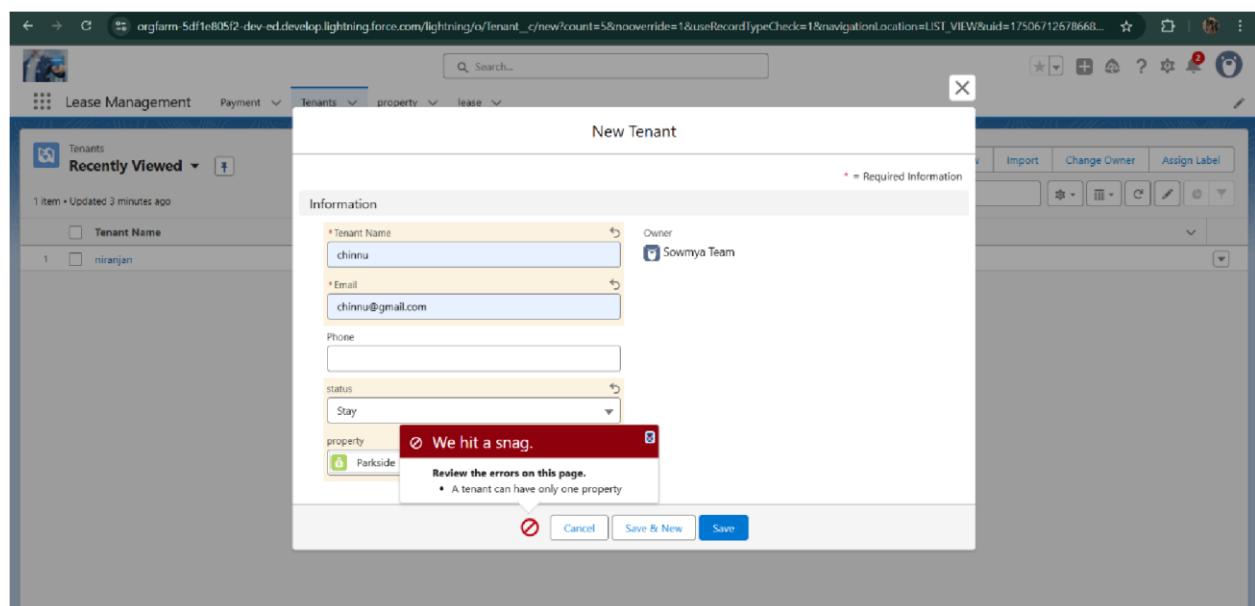
- Leave rejected



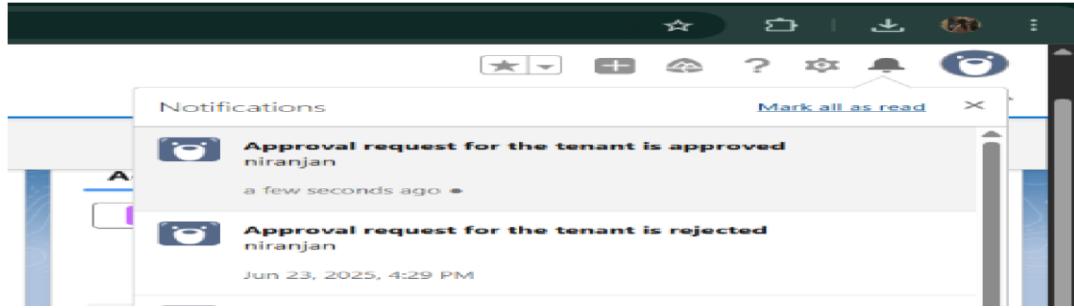
- Flow runs



- Trigger error messages



- Approval process notifications



ADVANTAGES & DISADVANTAGES

Advantages:

- Automates routine tasks like rent reminders, approvals, and payments.
- Reduces manual errors by using validation rules and flows.
- Improves efficiency and saves time for administrators.
- Provides real-time tracking of properties, tenants, and payments.
- Enhances communication through automated email alerts and notifications.
- Centralized data storage ensures easy access and better decision-making.

Disadvantages:

- Requires initial learning effort to understand salesforce setup and features.
 - Internet connectivity is mandatory to use salesforce effectively.
 - Advanced customization (like Apex coding for triggers/schedulers) may be difficult for beginners.
 - In real-world business, Salesforce licensing costs can be high (though the developer edition is free for projects).
-

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)) {

            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});

}

}

```