

PROJECT TITLE

College Name : Puratchi Thalaivi Amma Government College of Arts and Science

College Code: bruaq

TEAM ID: NM2025TMID21739

TEAM MEMBERS:

Team Leader Name: YOGESH P

Email: yogiyogeshyogiyoges@gmail.com

Team Member 1: DHANUSH K

Email: dhanushkrishna021@gmail.com

Team Member 2: SATHEESH R

Email: ramanraman60991@gmail.com

Team Member 3: SUBASH V

Email: jeevasupash@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.



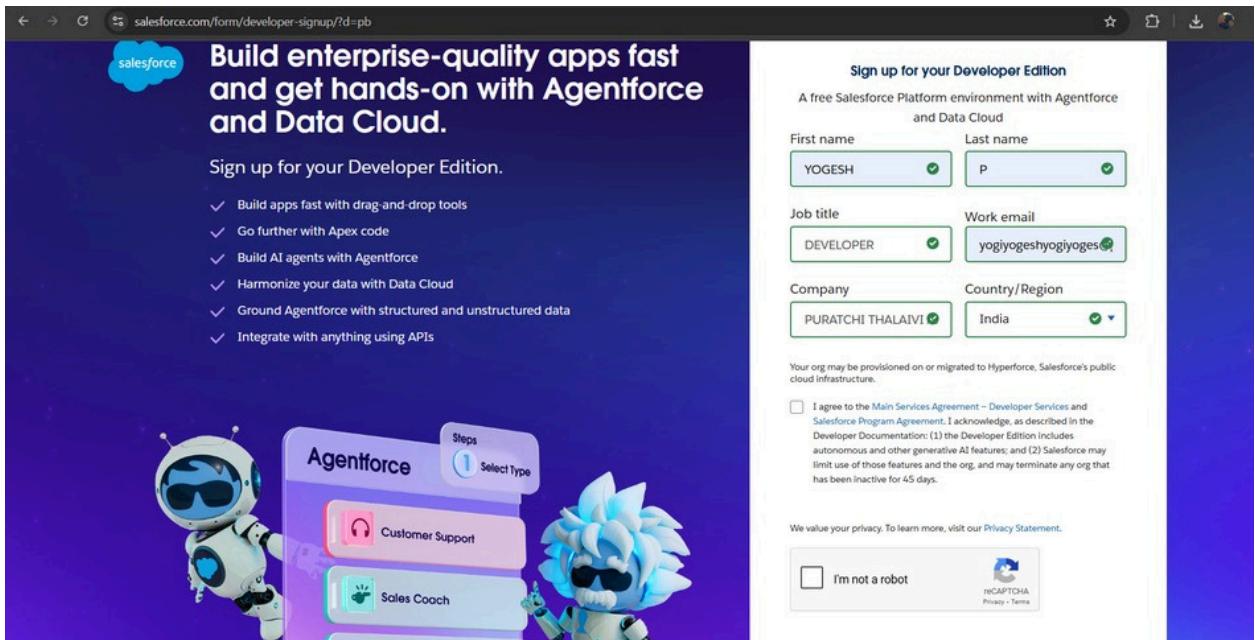
1. Purpose

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:

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



- 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

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', singular label 'Tenant', and plural label 'Tenants'. On the right, there are sections for enabling reports, tracking activities, and field history, all of which are checked. Deployment status is set to 'Deployed' and help settings point to the standard Salesforce help window. The top navigation bar includes links for Setup, Home, and Object Manager.

SETUP > OBJECT MANAGER

lease

Details

Description

API Name: lease__c

Custom:

- Singular Label: lease
- Plural Label: lease

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

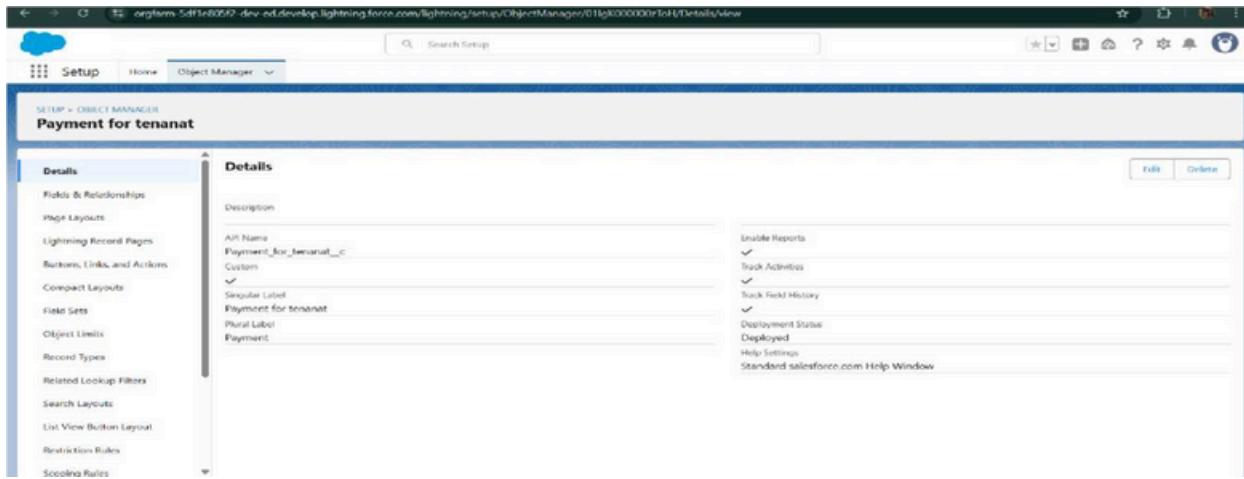
Deployment Status: Deployed

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 of 'lease__c', a custom label, and a plural label. It also includes sections for reports, activities, and field history tracking, all marked with checkmarks. Deployment status is set to 'Deployed' and help settings are standard. The left sidebar and top navigation bar are identical to the 'Tenant' configuration screen.



- 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)		
sfq	sfq__c	Text(18)		
Type	Type__c	Picklist		

Setup > Object Manager

Payment for tenantat

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 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...

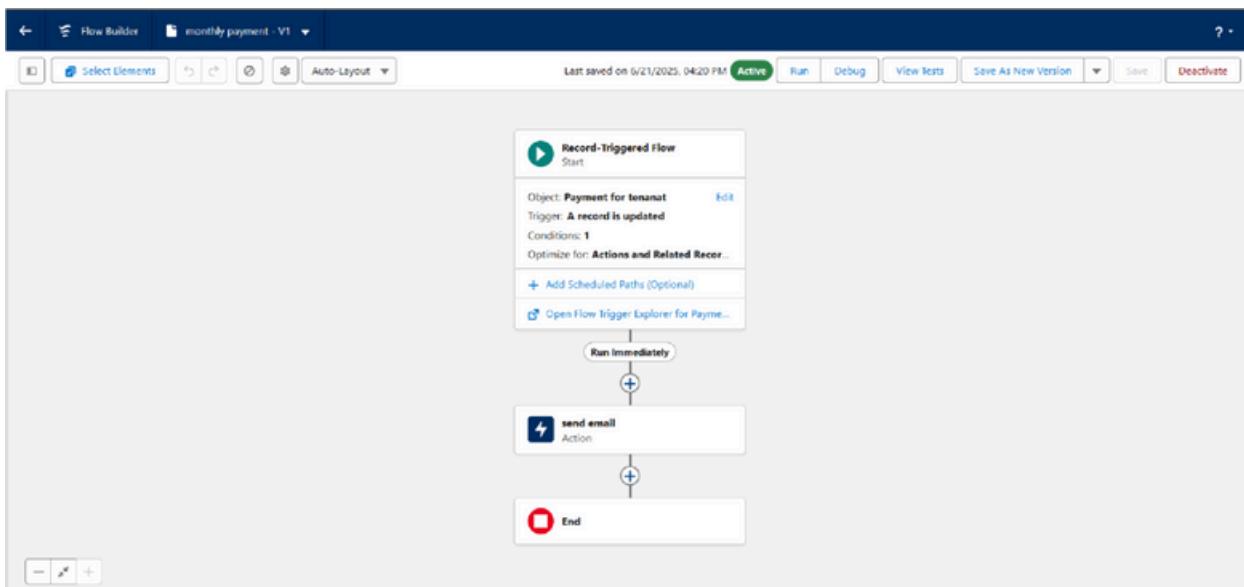
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

Selected Profiles

- System Administrator

A screenshot of a Salesforce Lightning interface. The top navigation bar shows 'Lease Management' and 'Payment'. Below it is a search bar and a toolbar with icons for New, Import, Change Owner, and Assign Label. The main area displays a list titled 'Recently Viewed' under the 'Payment' tab. It shows 5 items updated a few seconds ago, with columns for 'Payment Name' and 'Owner'. The items listed are: 1. Rahul, 2. Jack, 3. Raj, 4. Sam, and 5. Lahari.

- Implemented Flows for monthly rent and payment success

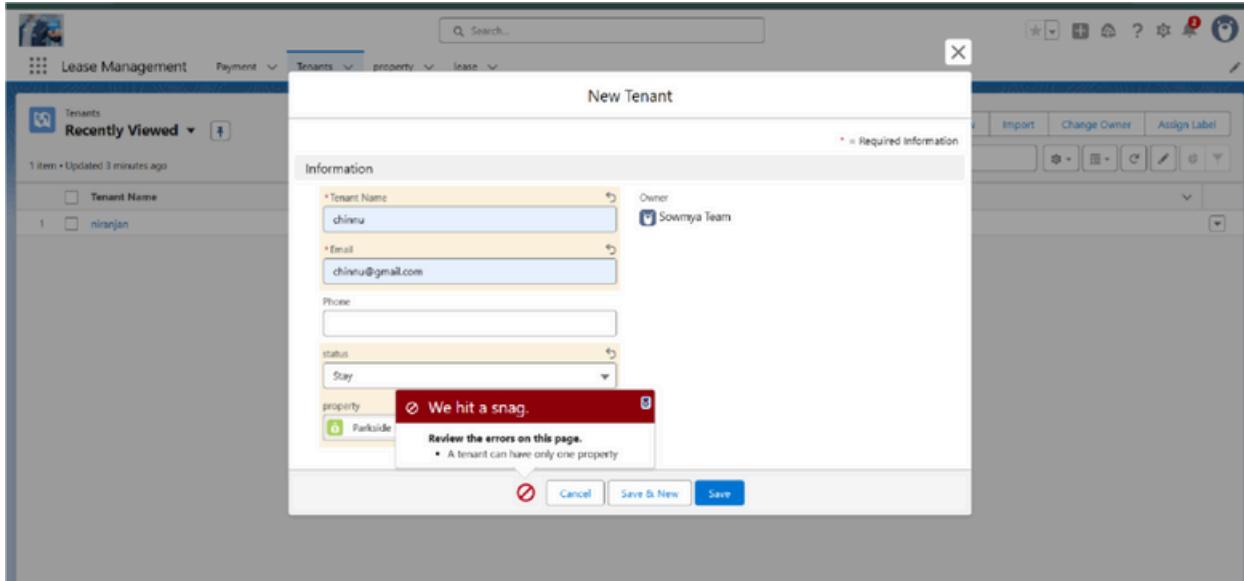


- To create a validation rule to a Lease Object

The screenshot shows the 'Validation Rule Edit' page for the 'lease' object. The 'Role Name' field contains 'lease_end_date'. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains the formula 'End_date_c <= start_date_c'. A tooltip for the 'ABS' function is displayed, stating 'Returns the absolute value of a number, a number without its sign'. The 'Error Message' field is empty.

The screenshot shows the 'Validation Rule Detail' page for the 'lease' validation rule. The rule name is 'lease_end_date'. The error condition formula is 'End_date_c <= start_date_c'. The error message is 'Your End date must be greater than start date'. The rule is active and was created by 'Sourya Team' on 6/19/2025 at 5:37 AM. It was modified by 'Sourya Team' on 6/26/2025 at 7:47 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/HelloWorld.apk
Java code APIDocs 44 %
67% - + Reset
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+
4
```

Log Test Checkstyle Run Status Request Policies

- 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', a specific template named 'Leave approved' is displayed. The template details include:

- Email Template Name:** Leave approved
- Template Unique Name:** leave_approved
- Encoding:** Unicode (UTF-8)
- Author:** Scorma Team [Changed]
- Description:** Created By: Scorma Team, 6/20/2025, 1:05 AM
- Last Used Date:** Not specified
- Times Used:** Not specified
- Modified By:** Scorma Team, 6/20/2025, 1:05 AM

The preview pane shows the email content:

```

Subject: Leave approved
Main 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.

where {tenant} is account. You can leave now.

```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', a specific template named 'tenant leaving' is displayed. The template details include:

- Email Template Name:** tenant leaving
- Template Unique Name:** tenant_leaving
- Encoding:** Unicode (UTF-8)
- Author:** Scorma Team [Changed]
- Description:** Created By: Scorma Team, 6/20/2025, 1:05 AM
- Last Used Date:** Not specified
- Times Used:** Not specified
- Modified By:** Scorma Team, 6/20/2025, 1:05 AM

The preview pane shows the email content:

```

Subject: request for approve the leave
Main Text Preview:
Dear {Tenant__c.CreatedBy}.

Please approve my leave.

```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', there is a list item for 'Leave rejected'. The main content area displays the 'Leave rejected' email template details. The 'Email Template Detail' section includes fields like Email Template Name (Leave rejected), Template Unique Name (Leave_rejected), Encoding (Unicode (UTF-8)), Author (Sohma_Team (Changed)), Description (Created By: Sohma_Team, 6/20/2025, 1:11 AM), Available For Use (checked), Last Used Date, and Times Used. Below this is the 'Email Template' preview section, which shows the subject 'Leave rejected', the plain text preview (Dear [Tenant__c.Name]), and the HTML preview (I hope this email finds you well. Your contract has not ended. So we can't approve your leave. Your leave has rejected). A 'Send Test and Verify Merge Fields' button is also present.

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', there is a list item for 'Tenant Email'. The main content area displays the 'Tenant Email' email template details. The 'Email Template Detail' section includes fields like Email Template Name (Tenant Email), Template Unique Name (Tenant_Email), Encoding (Unicode (UTF-8)), Author (Sohma_Team (Changed)), Description (Created By: Sohma_Team, 6/20/2025, 1:12 AM), Available For Use (checked), Last Used Date, and Times Used. Below this is the 'Email Template' preview section, which shows the subject 'Urgent: Monthly Rent Payment Reminder', the plain text preview (Dear [Tenant__c.Name]), and the HTML preview (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.). A 'Send Test and Verify Merge Fields' button is also present.

The screenshot shows the Salesforce Setup interface with the search bar set to 'email template'. The left sidebar under 'Email' has 'Classic Email Templates' selected. A specific template, 'tenant payment', is highlighted. The main content area displays the 'Email Template Detail' for 'tenant payment', which is 'Untitled Public Classic Email Templates'. The template details include:

- Email Template Name: tenant payment
- Template Unique Name: tenant_payment
- Encoding: Unicode (UTF-8)
- Author: Soma_Team [Change]
- Description: Created by Soma_Team, 6/20/2025, 1:13 AM
- Last Used Date: Not specified
- Times Used: Not specified
- Modified By: Soma_Team, 6/20/2025, 1:13 AM

The 'Email Template' section shows the subject 'Confirmation of Successful Monthly Payment' and a plain text preview:

```

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:

The screenshot shows the Salesforce Setup interface with the search bar set to 'approval'. The left sidebar under 'Data' has 'Approval Processes' selected. An approval process named 'Tenant: TenantApproval' is selected. The main content area displays the 'Process Definition Detail' for 'TenantApproval', which is 'TenantApproval'. The process details include:

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description: Next Automated Approver Determined by
- Entry Criteria: IsStatus: status equals Stay
- Record Editability: Administrator ONLY
- Allow Submitters to Recall Approval Requests:
- Created By: Soma_Team, 6/20/2025, 3:41 AM
- Modified By: Soma_Team, 6/20/2025, 11:57 PM

The 'Initial Submission Actions' section shows a single action:

Action Type	Description
Record Lock	Lock the record from being edited

The 'Approval Steps' section shows one step:

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

For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page selected. The process name is 'Tenant: check for vacant'. The 'Process Definition Detail' section includes fields like Process Name (check for vacant), Unique Name (Check_for_vacant), Description (Tenant: status equals Leaving), Entry Criteria (Tenants: status equals Leaving), Record Visibility (Administrator ONLY), Next Automated Approver (Determined by), and Approval Assignment Email Template (Leave approved). The 'Initial Submission Actions' section contains an action 'Record Lock' with a description 'Lock the record from being edited'. The 'Approval Steps' section shows one step named 'step1' assigned to 'User: Bowmav.Team' with a reject behavior of 'Final Rejection'.

- Apex Trigger

Create an Apex Trigger

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'test' being created. The trigger code is as follows:

```

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

```

A modal window titled 'Open' is displayed, showing a list of entities under 'Entity Type': Classes, Triggers, Pages, Page Components, Objects, Static Resources, and Packages. The 'Triggers' option is selected. The 'Edition' dropdown shows 'test'.

Developer Console - Google Chrome
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

File • Edit • Debug • Test • Workspace • Help • < >

testHandler.apc [MonthlyTaskScheduler.apc]

Code Coverage: None • API Version: 41 • Go To

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

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

Create an Apex Handler class

Developer Console - Google Chrome
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

File • Edit • Debug • Test • Workspace • Help • < >

testHandler.apc [MonthlyTaskScheduler.apc]

Code Coverage: None • API Version: 41 • 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(existin
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
Empty List			
Classes	testHandler	MonthlyTaskScheduler	+< test ApexTrigger References
Triggers			+< property CustomerId References
Pages			+< Tenant__c Subject References
Page Components			+< Tenant__c Subject References
Objects			
Item Resources			
Packages			

Open Filter: Filter the repository (* = any string) Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

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

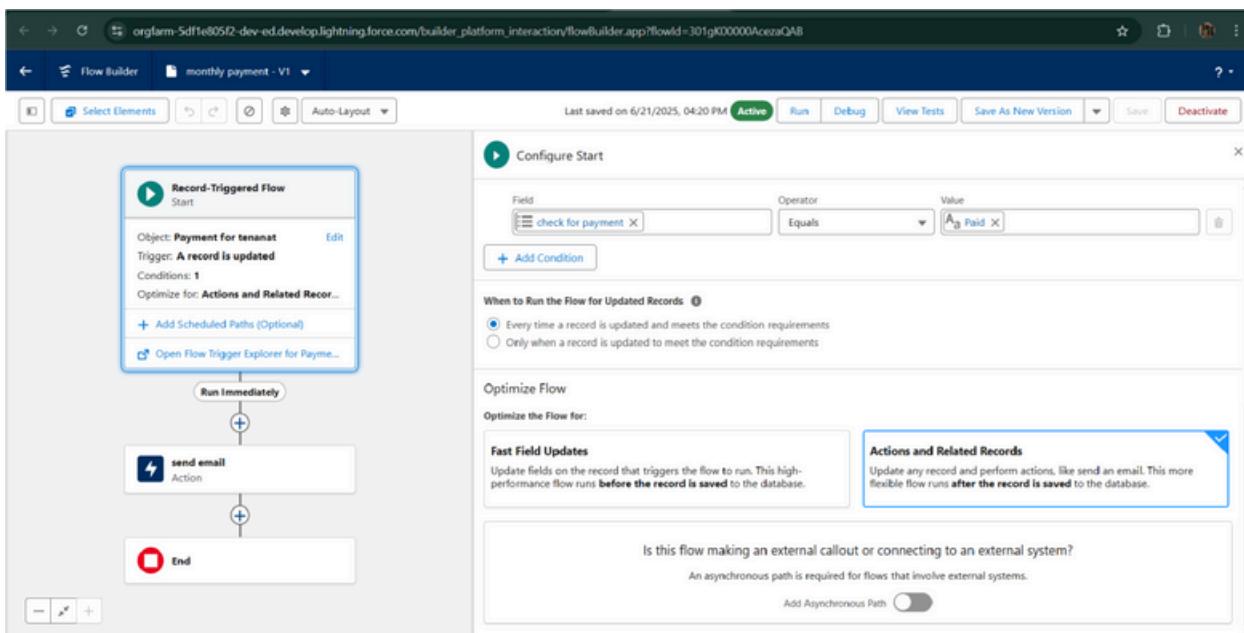
```

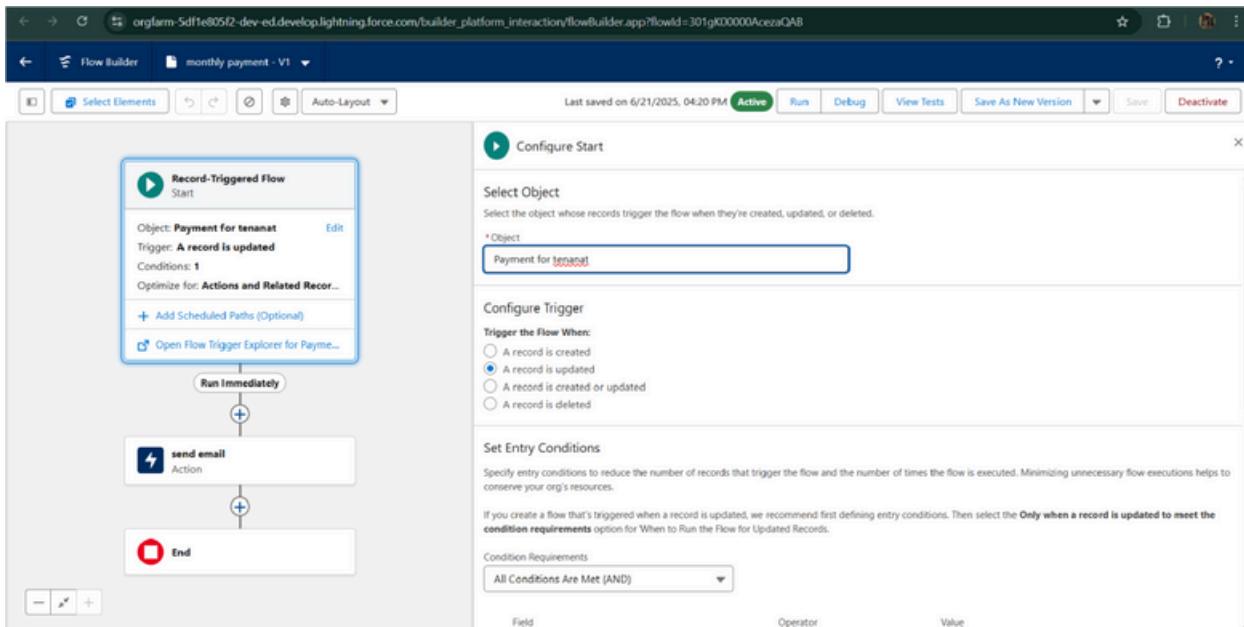
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);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

```

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

Entity Type	Edition	Related
Classes	Name: MonthlyEmailScheduler	Name: testhandler Object: CustomObject CustomField: Email CustomField: Tenant__c CustomField: Tenant__c CustomField: Subject
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
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 = 'Dear Sir/Madam, 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 = 'Rental Reminders: Monthly Rent Payment Due';
26
27             Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29             mail.setTo(recipientEmail);
30             mail.setSubject(emailSubject);
31             mail.setPlainTextEmailContent();
32             mail.setHTMLEmailContent();
33
34             messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
35         }
36     }
37 }
38

```

Schedule Apex class

The screenshot shows the Salesforce Setup interface with the search bar set to "apex". The left sidebar is expanded to show categories like Email, Custom Code, and Environments. Under "Apex Classes", the "MonthlyEmailScheduler" class is selected. The main pane displays the Apex Class Detail for "MonthlyEmailScheduler". The "Name" field is "MonthlyEmailScheduler", and the "Status" is "Active". The "Created By" field shows "Sonoya Team" and the "Last Modified By" field also shows "Sonoya Team". Below the details, there are tabs for "Class Body", "Class Summary", "Version Settings", and "Trace Flags". The "Class Body" tab is active, showing the Apex code for the 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     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 = 'Dear Sir/Madam, 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 = 'Rental Reminders: Monthly Rent Payment Due';
26
27             Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
28
29             mail.setTo(recipientEmail);
30             mail.setSubject(emailSubject);
31             mail.setPlainTextEmailContent();
32             mail.setHTMLEmailContent();
33
34             messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
35         }
36     }
37 }
38

```

Lease Management

Tenant sathish

Related Details

* = Required Information

Tenant Name: sathish

Email: sathish22@gmail.com

Phone:

Status: Leaving

Property: imran

Created By: Yogesh P. 9/2/2025, 2:17 AM

Last Modified By: Yogesh P. 9/2/2025, 2:29 AM

Activity

New Case New Lead Delete Clone Change Owner Printable View Refresh 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.

New Contact Edit New Opportunity

Lease Management

Tenant sathish

Related Details

* = Required Information

Tenant Name: sathish

Email: sathish22@gmail.com

Phone:

Status: Leaving

Property: imran

Created By: Yogesh P. 9/2/2025, 2:17 AM

Last Modified By: Yogesh P. 9/2/2025, 2:29 AM

Tenant was submitted for approval.

Activity

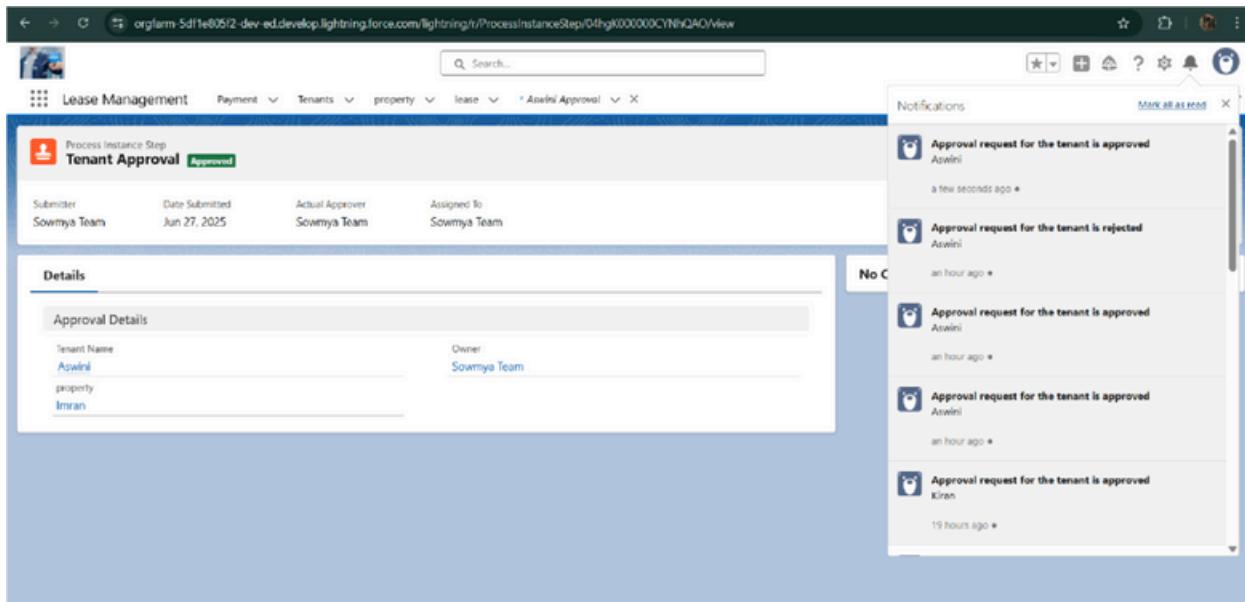
Filters: All time • All activities • All types Refresh • Expand All • View All

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.

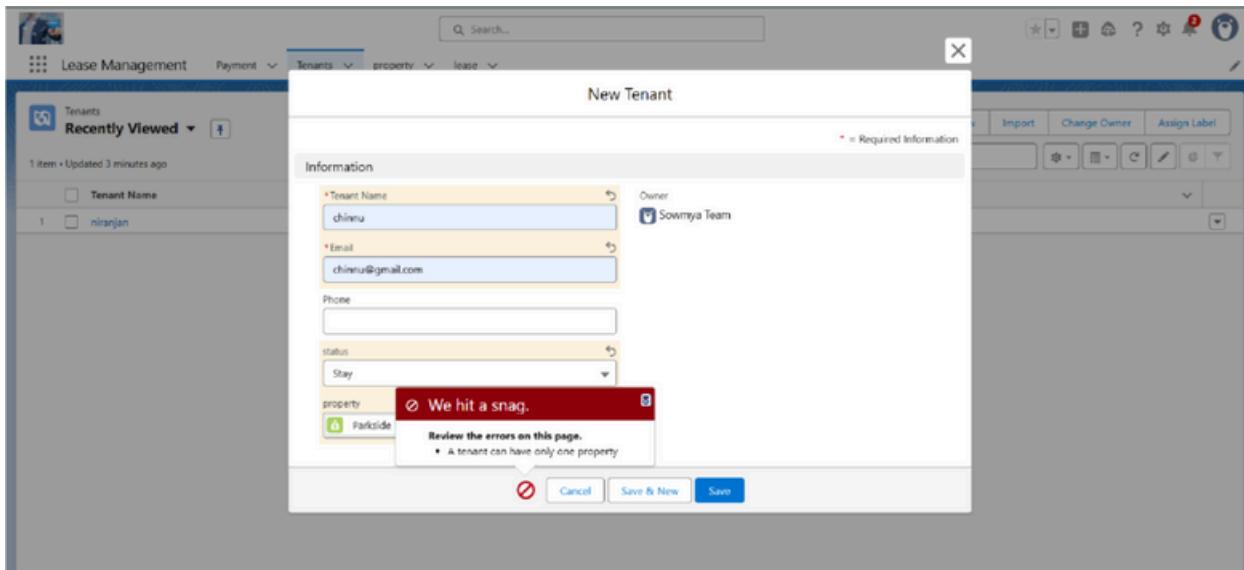
New Contact Edit New Opportunity



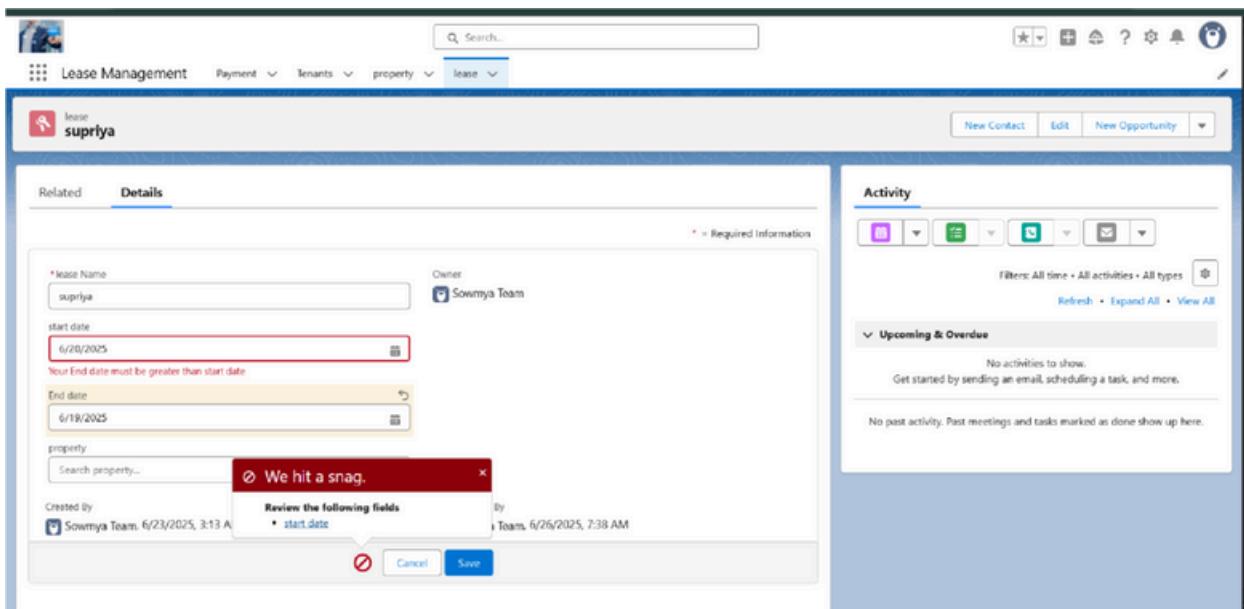
FUNCTIONAL AND PERFORMANCE TESTING

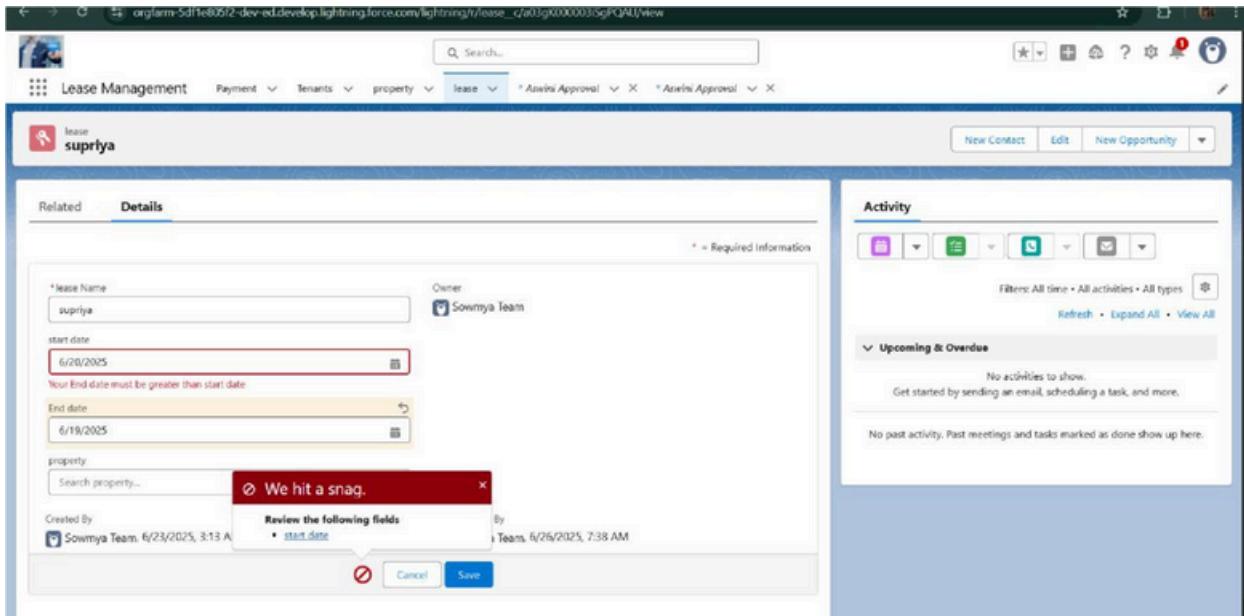
Performance Testing

- Triggervalidationbyenteringduplicate tenant-property records

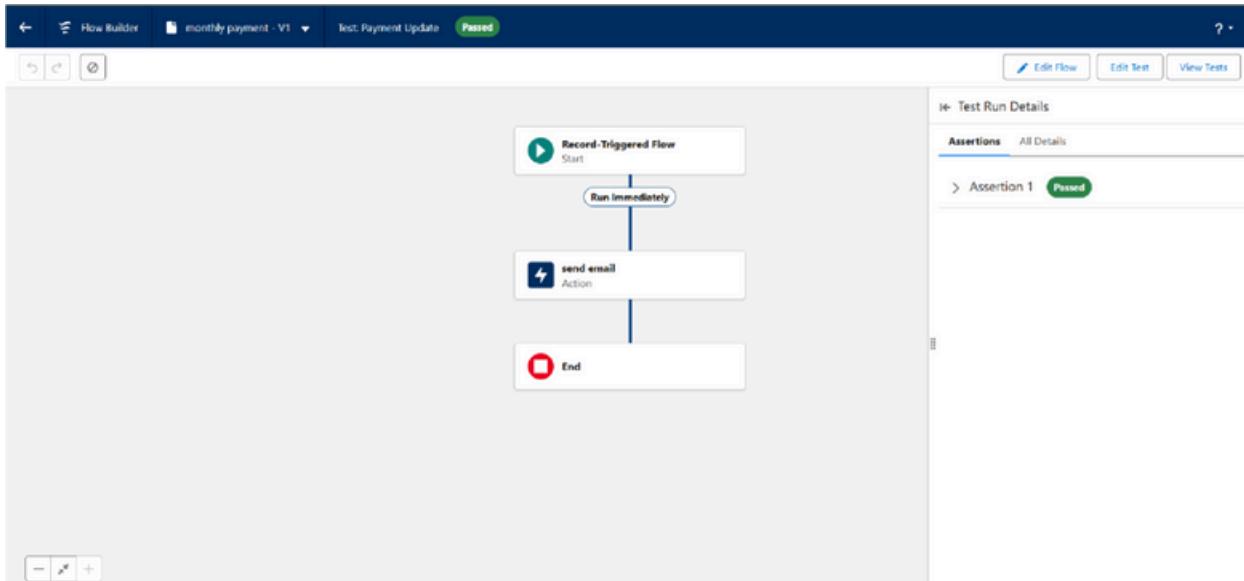


- Validation Rule checking





- 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 'Tenants' record for 'niranjan' is displayed. The 'Details' tab is selected, showing fields like 'Tenant Name' (niranjan), 'Email' (niranjan1506@gmail.com), 'Phone' (empty), 'status' (Stay), and 'property' (Parkade Lofts). The 'Owner' is listed as 'Sowmya Team'. Below the form are 'Created by' and 'Last Modified by' fields, both showing 'Sowmya Team' on June 23, 2025, at 2:33 AM and 3:58 AM respectively. At the bottom are 'Cancel' and 'Save' buttons. To the right, a sidebar titled 'Notifications' 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)
- New Guidance Center learning resource available (Jun 20, 2025, 1:28 PM)

The screenshot shows a CRM interface for 'Lease Management'. On the left, a 'Tenants' record for 'niranjan' is displayed. Under the 'Approval History' section, there are six entries:

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

Below the approval history is a 'Payment' section with two entries:

Payment Name
Jark
Rahul

To the right, a sidebar shows a message: 'Get started by sending an email, scheduling a task, and more.' Below it, a note says: 'No past activity. Past meetings and tasks marked as done show up here.'

RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

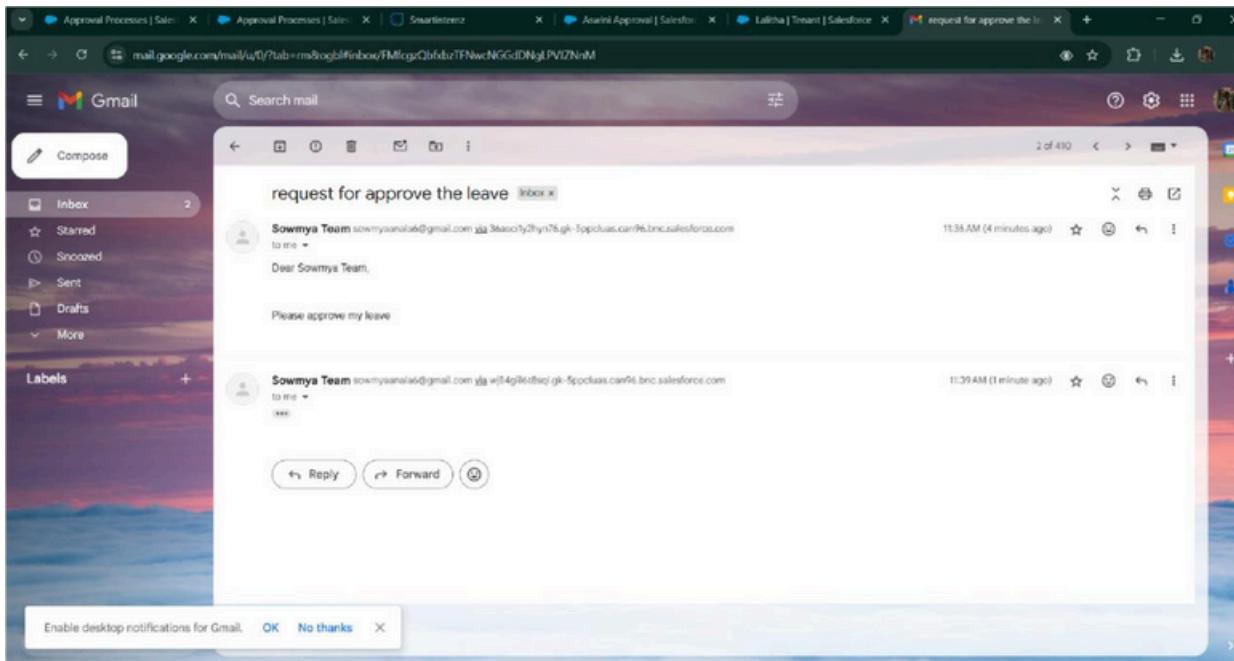
The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page open. The left sidebar has 'User Interface' expanded, with 'Tabs' selected. The main content area is titled 'Custom Tabs' and contains instructions: 'You can create new custom tabs to extend Salesforce functionality or to build new application functionality.' Below this are three sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section lists four tabs: 'Lease' (Style: Keys), 'Payment' (Style: Credit Card), 'Property' (Style: Back), and 'Records' (Style: Map). The 'Web Tabs' and 'Visualforce Tabs' sections both state 'No Web Tabs have been defined' and 'No Visualforce Tabs have been defined' respectively.

- Email alerts

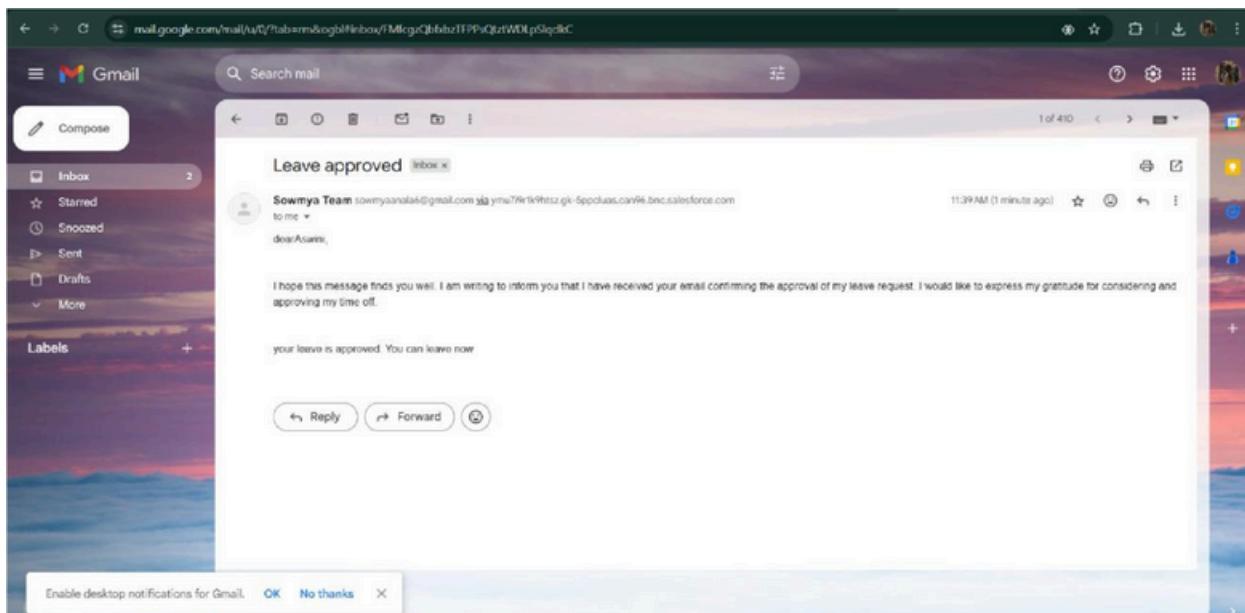
The screenshot shows the 'Lease Management' application interface with the 'Approval History' page open. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. The page title is 'Approval History' and it shows '8 items • Sorted by Is Pending • Updated a few seconds ago'. A table lists eight approval steps with columns: Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The steps are numbered 1 through 8, with various status and approver details.

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 Step 1	6/23/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
2 Approval Request Submitted	6/23/2025, 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

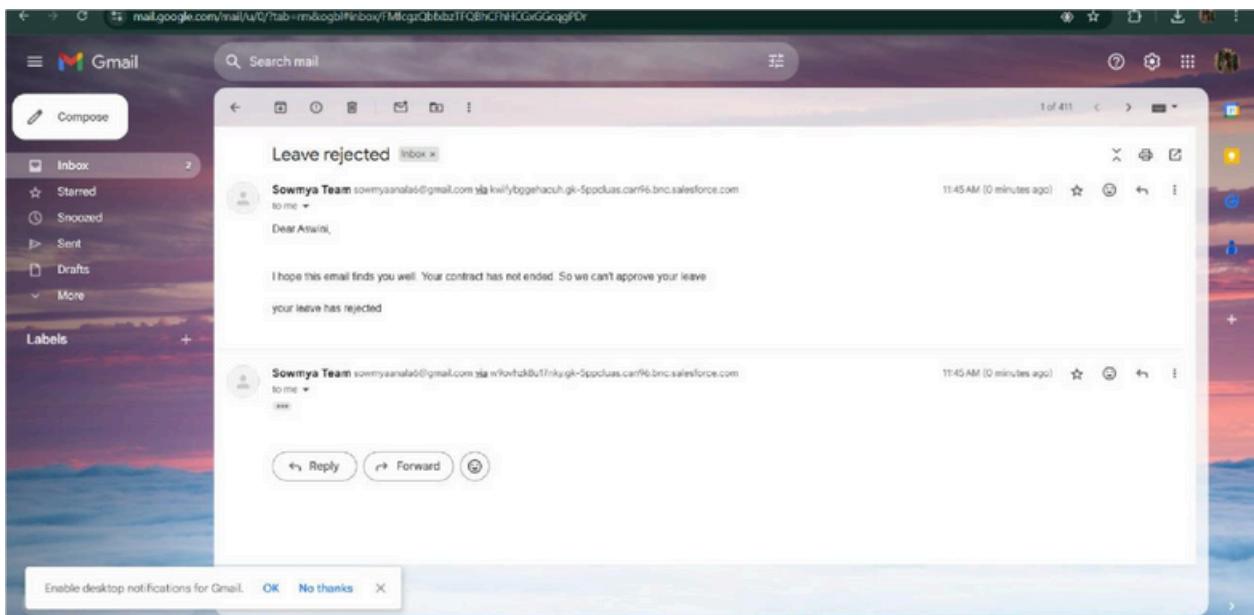
- Request for approve the leave



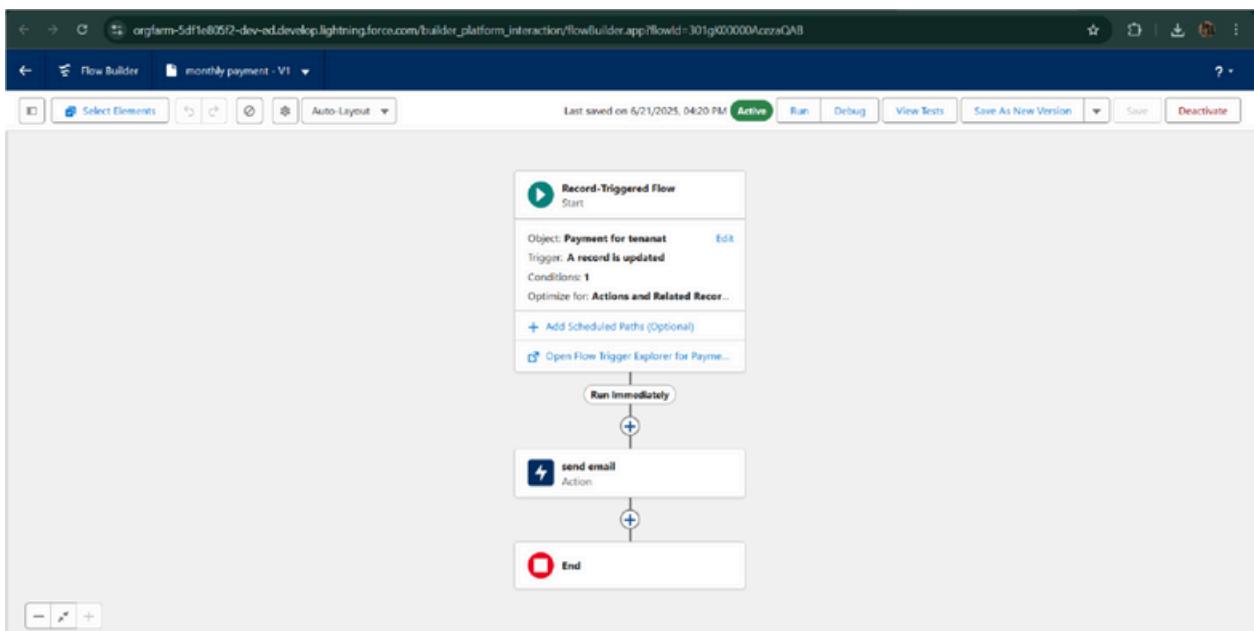
- Leave approved



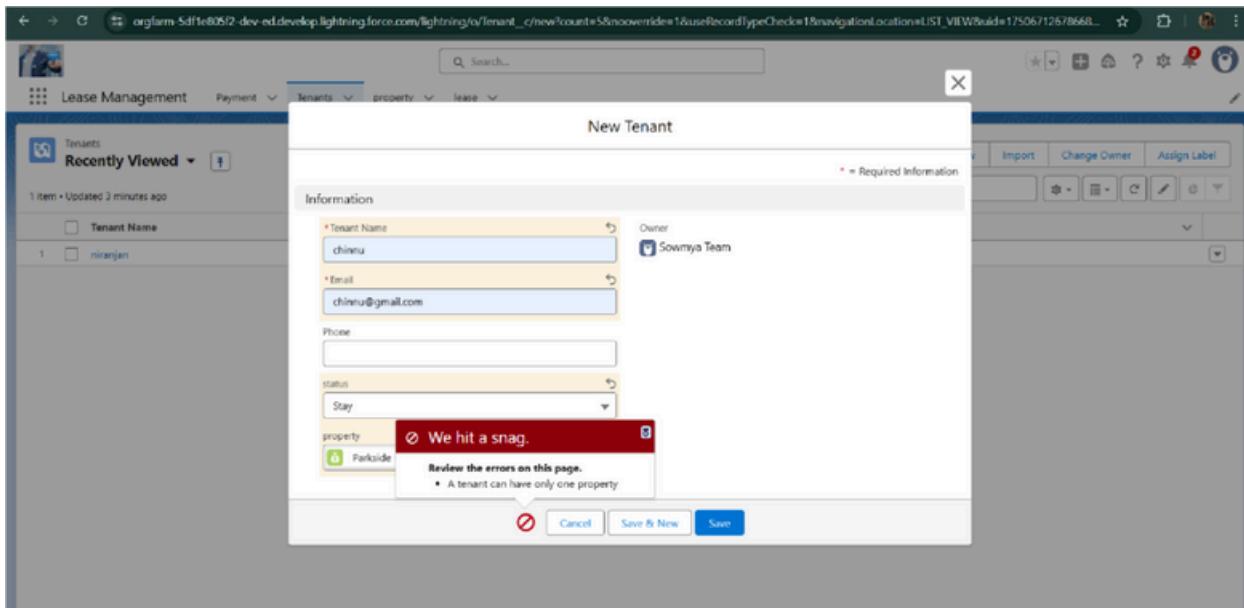
- Leave rejected



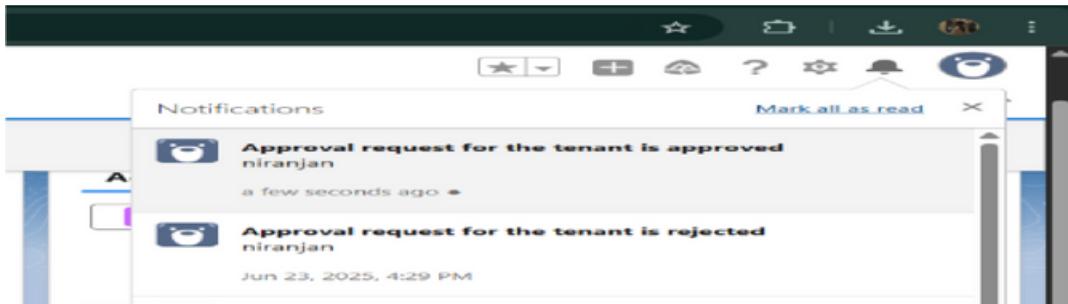
- Flow runs



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

testHandler.apxc:

```
public class  
testHandler {  
    public static void  
    preventInsert(List<  
        Tenant__c> newlist)  
  
    {  
        Set<Id>  
        existingPropertyIds  
        = new Set<Id>()  
  
        for (Tenant__c existingTenant : [SELECT Id, Property  
        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  
            that the monthly rent is due Your timely payment ensure  
            rental arrangement and helps maintain a positive living  
            environment for all.  
              
            Thank you for your cooperation.';  
            String emailSubject = 'Reminder: Monthly Rent Payment Due';  
            String body = '

Subject: ' + emailSubject + '



Content: ' + emailContent + '

';  
            Messaging.SingleEmailMessage message = new Messaging.SingleEmailMessage();  
            message.setTo(recipientEmail);  
            message.setSubject(emailSubject);  
            message.setPlainTextBody(body);  
            message.setHtmlBody(body);  
            Messaging.sendEmail(new List<Messaging.SingleEmailMessage>{message});  
        }  
    }  
}
```

```
Messaging.SingleEmailMessage email = new  
Messaging.SingleEmailMessage(); email.setToAddresses(new  
String[]{recipientEmail}); email.setSubject(emailSubject);  
email.setPlainTextBody(emailContent);  
Messaging.sendEmail(new Messaging.SingleEmailMessage (email));
```

