

PROJECT TITLE

College Name: VET INSTITUTE OF ARTS & SCIENCE

College Code: 65651

TEAM ID: NM2025TMID25836

TEAM MEMBERS:

Team LeaderName: ABI G

EMAIL: abig23aid@vetias.ac.in

Team member : DHARSHINI S

Team member : KANNAN T

Team member : KAVIPRIYANGAA J P

Team member : SOWKIYA SHREE R

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.



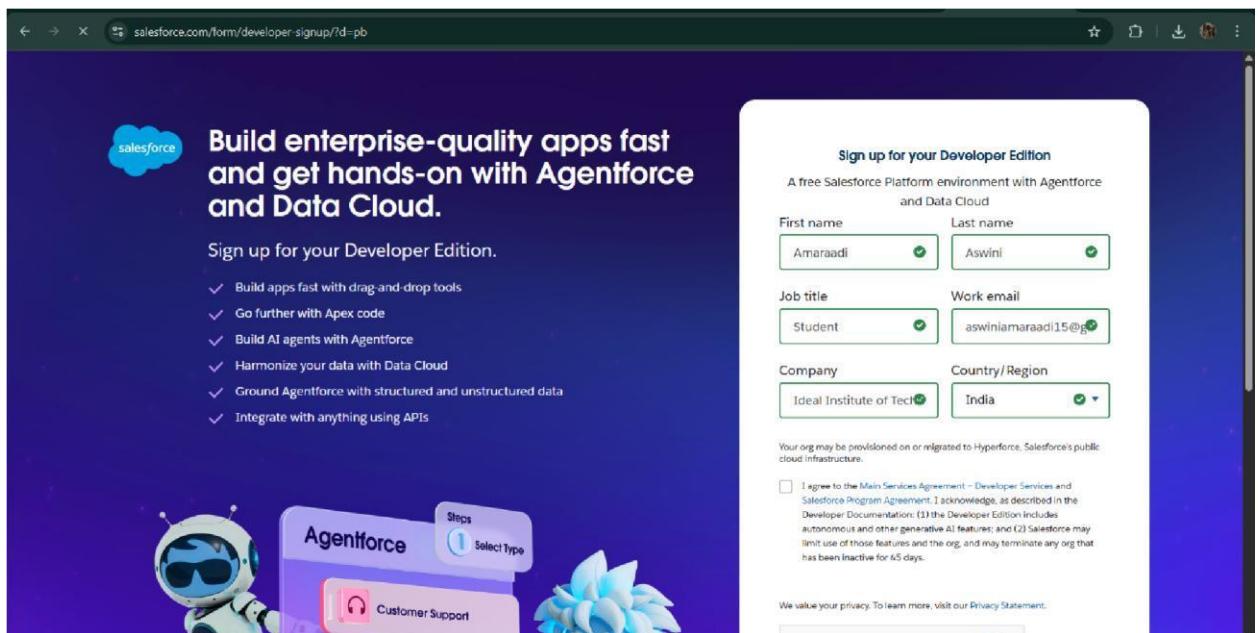
1.2 Purpose

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

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000z2TfI/Details/view

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER property'. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The 'Details' tab is selected. The main content area displays the 'property' object's details. The 'Description' section contains the API Name 'property_c', which is custom and singularly labeled 'property'. The 'Enable Reports' section has four checked options: Track Activities, Track Field History, Deployment Status (set to 'Deployed'), and Help Settings (linking to 'Standard salesforce.com Help Window').

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK00000zTbN/Details/view

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER Tenant'. On the left, a sidebar lists various configuration tabs: Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The 'Details' tab is selected. The main content area displays the 'Tenant' object's details. The 'Description' section contains the API Name 'Tenant_c', which is custom and singularly labeled 'Tenant'. The 'Enable Reports' section has four checked options: Track Activities, Track Field History, Deployment Status (set to 'Deployed'), and Help Settings (linking to 'Standard salesforce.com 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**

SETUP > OBJECT MANAGER

Payment for tenant

Details

Description

API 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

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**

- Configured fields and relationships

Fields & Relationships
9 Items, Sorted by Field Label

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(80)		✓
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Number(18, 0)		
check-for-payment	check_for_payment__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(80)		✓

Setup Home Object Manager

SETUP > OBJECT MANAGER
lease

Fields & Relationships 7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End date	End_date_c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User/Group)		✓
property	property_c	Lookup(property)		✓
start date	start_date_c	Date		

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 Home Object Manager

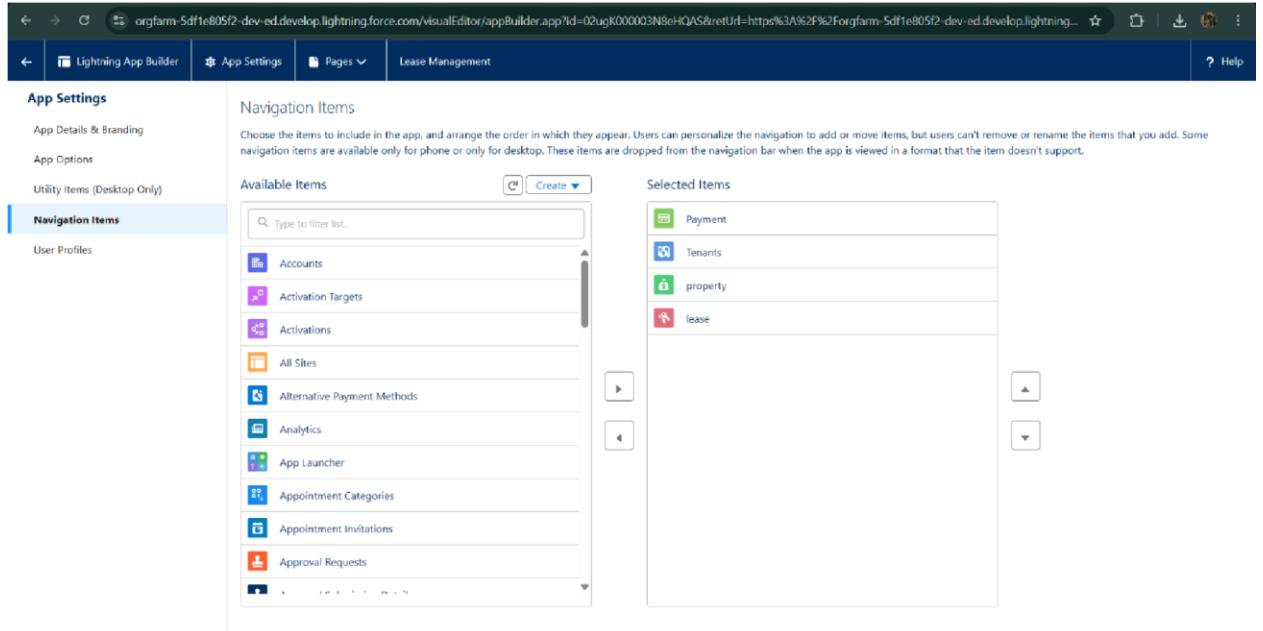
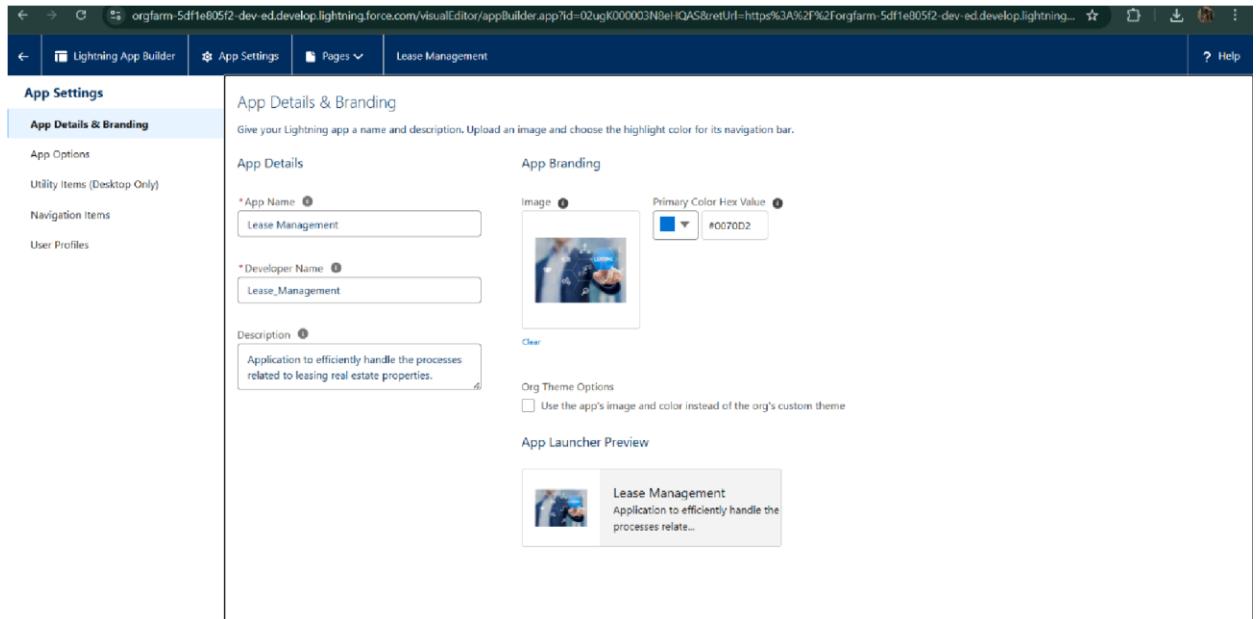
SETUP > OBJECT MANAGER
Tenant

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	Name		✓

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

- Developed Lightning App with relevant tabs

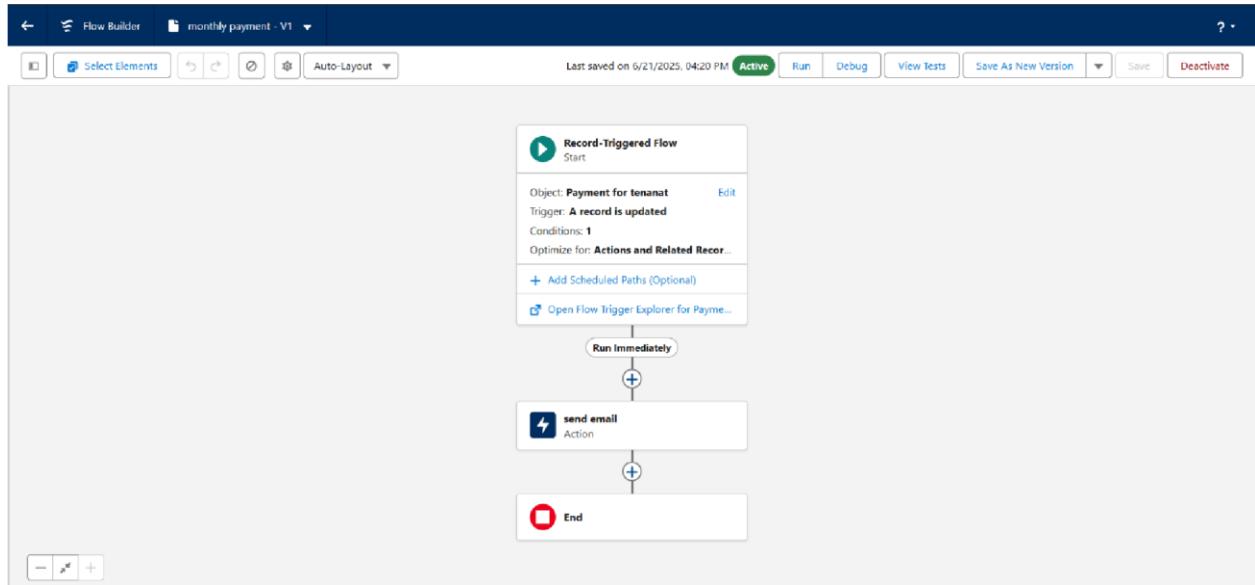


The screenshot shows the 'User Profiles' section of the Lightning App Builder. On the left, there's a sidebar with 'App Settings' and 'Navigation Items'. Under 'App Settings', 'User Profiles' is selected. The main area has two sections: 'Available Profiles' on the left and 'Selected Profiles' on the right. The 'Available Profiles' list includes: Analytics Cloud Integration User, Analytics Cloud Security User, Anypoint Integration, Authenticated Website, B2B Reordering Portal Buyer Profile, Contract Manager, Custom: Marketing Profile, Custom: Sales Profile, Custom: Support Profile, and Customer Community Login User. The 'Selected Profiles' list contains 'System Administrator'. A search bar at the top says 'Choose the user profiles that can access this app.'

The screenshot shows the 'Recently Viewed' list in the Lease Management section. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. The main area displays a table with columns for 'Payment Name' and 'Last Viewed'. The data in the table is as follows:

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

- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

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.

Insert Field: Insert Operator:

Formula: End_date_c <= start_date_c

Functions:

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

Quick Tips: Operators & Functions

F = Required Information

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Setup, Home, and Object Manager. The main title is "lease Validation Rule". The validation rule details are as follows:

- 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

The sidebar on the left lists various configuration options for the lease object, such as 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, and Scoping Rules.

- Added Apex trigger to restrict multiple tenants per property

The screenshot shows the Salesforce Lease Management interface. A modal window titled "New Tenant" is open, displaying an "Information" section with fields for Tenant Name (chinnu) and Email (chinnu@gmail.com). An error message at the bottom states: "We hit a snag. Review the errors on this page. * A tenant can have only one property". The background shows a list of tenants with one item visible: "1 niranjan".

- 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

Classic Email Templates

Email Template Detail

Email Template Name	Leave approved	Available For Use
Template Unique Name	Leave_approved	Last Used Date
Encoding	Unicode (UTF-8)	Times Used
Author	Sowmya Team (Change)	
Description		
Created By	Sowmya Team	Modified By
	6/20/2025, 1:08 AM	Sowmya Team

Email Template

Plain Text Preview

dear{!Tenant__c.Name},

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

The screenshot shows the Salesforce Setup interface. In the top left, there's a blue cloud icon. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Setup'. On the left sidebar under 'Email', 'Classic Email Templates' is selected. A search bar on the sidebar says 'email template'. The main content area has a title 'SETUP Classic Email Templates'. It shows a 'Text Email Template' named 'tenant leaving'. The 'Email Template Detail' section includes fields like 'Email Template Name' (tenant leaving), 'Template Unique Name' (tenant_leaving), 'Encoding' (Unicode (UTF-8)), 'Author' (Sowmya_Team [Change]), 'Description' (None), 'Created By' (Sowmya_Team), 'Modified By' (Sowmya_Team), and status indicators for 'Available For Use' (checked), 'Last Used Date' (blank), and 'Times Used' (0). Below this is a preview pane with a subject line 'request for approve the leave' and a plain text preview: 'Dear {Tenant__c.CreatedBy}, Please approve my leave'.

This screenshot shows the same Salesforce Setup interface as the first one, but with a different email template. The main content area has a title 'SETUP Classic Email Templates'. It shows a 'Text Email Template' named 'Leave rejected'. The 'Email Template Detail' section includes fields like 'Email Template Name' (Leave rejected), 'Template Unique Name' (Leave_rejected), 'Encoding' (UTF-8), 'Author' (Sowmya_Team [Change]), 'Description' (None), 'Created By' (Sowmya_Team), 'Modified By' (Sowmya_Team), and status indicators for 'Available For Use' (checked), 'Last Used Date' (blank), and 'Times Used' (0). Below this is a preview pane with a subject line 'Leave rejected' and a 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'.

Classic Email Templates

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

Email Template

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.
```

Classic Email Templates

Tenant payment

Email Template Detail

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant_payment
Template Unique Name	tenant_payment
Encoding	Unicode (UTF-8)
Author	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

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 for the 'Approval Processes' section. The process is named 'TenantApproval'.

Process Definition Detail:

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description: Tenant Approval
- Entry Criteria: tenant__r.status EQUALS Stay
- Record Editability: Administrator ONLY
- Next Automated Approver Determined By:
- Allow Submitters to Recall Approval Requests:
- Initial Submitters: Tenant Owner
- Created By: Sowmya Team
- Modified By: Sowmya Team

Initial Submission Actions:

- Action Type: Record Lock
- Description: 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 for the 'Approval Processes' section. The process is named 'check for vacant'.

Process Definition Detail:

- Process Name: check for vacant
- Unique Name: check_for_vacant
- Description: check for vacant
- Entry Criteria: Tenant: status EQUALS Leaving
- Record Editability: Administrator ONLY
- Next Automated Approver Determined By:
- Allow Submitters to Recall Approval Requests:
- Initial Submitters: Tenant Owner
- Created By: Sowmya Team
- Modified By: Sowmya Team

Initial Submission Actions:

- Action Type: Record Lock
- Description: Lock the record from being edited
- Action Type: Email Alert
- Description: SOWMYA.BOROVIC.MY.JCWS

Approval Steps:

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

- Apex Trigger

Create an Apex Trigger

Screenshot of the Salesforce IDE showing the Apex code editor and the Open dialog.

```

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

```

The Open dialog is displayed, showing the Entity Type list with Triggers selected. The Entities table shows one entry: Name: test, Namespace: .

Below the code editor, the tabs Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems are visible. The Problems tab is selected.

Developer Console screenshot:

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.apex [] MonthlyEmailScheduler.apex [] Go To

Code Coverage: None API Version: 64

```

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

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Create an Apex Handler class

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

```

File Edit Debug Test Workspace Help < >
testHandler.apc MonthlyEmailScheduler.apc
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(existingTenant.Id);
10    }
11
12    for (Tenant__c newTenant : newList) {
13
14      if (newTenant.Property__c != null) {
15
16        newTenantaddError('A tenant can have only one property');
17
18      }
19
20    }
21
22  }
23
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Open Entity Type Entity Name Namespace Related

Entry Type	Name	Namespace	Name	Extent	Direction
Classes	testHandler		test	ApexTrigger	Referenced
Triggers	MonthlyEmailScheduler		property	CustomField	References
Pages			Tenant__c	SObject	References
Components			Tenant__c	SObject	References
Objects					
Static Resources					
Package					

Open Filter Hide Managed Packages Refresh

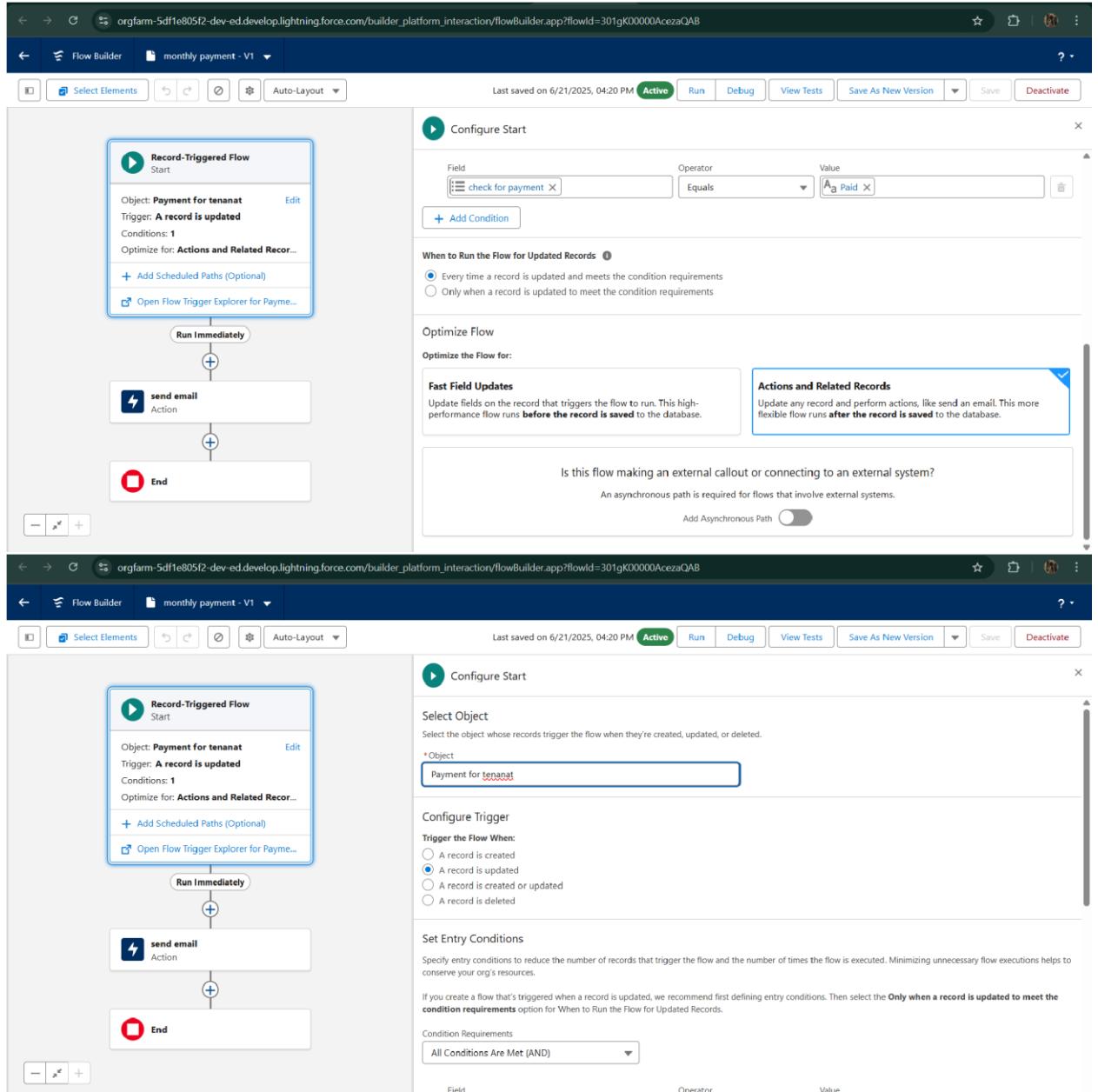
Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```

File Edit Debug Test Workspace Help < >
testHandler.apc MonthlyEmailScheduler.apc
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(existingTenant.Property__c);
10    }
11
12    for (Tenant__c newTenant : newList) {
13
14      if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
15
16        newTenantaddError('A tenant can have only one property');
17
18      }
19
20    }
21
22  }
23
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

- FLOWS



- Schedule class:
Create an Apex Class

Developer Console - Google Chrome

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

test.apex [testHandler.apex] MonthlyEmailScheduler.apex []

Code Coverage: None API Version: 64 Go To...

```
1 * global class MonthlyEmailScheduler implements Schedulable {  
2  
3     global void execute(SchedulableContext sc) {  
4  
5         Integer currentDay = Date.today().day();  
6  
7         if (currentDay == 1) {  
8             sendMonthlyEmails();  
9         }  
10    }  
11 }  
12  
13  
14  
15  
16 * public static void sendMonthlyEmail  
17  
18     List<Tenant__c> tenants = [SEL  
19  
20         for (Tenant__c tenant : tenants)  
21  
22             String recipientEmail = tenant.Email__c;  
23
```

Open Entity Type Entities Related

Entity Type	Name	Namespace	Name	Extent	Direction
Classes	testHandler		Email	CronTrigger	Referenced
Triggers	MonthlyEmailScheduler		← Tenant__c	SObject	References
Pages			← Tenant__c	SObject	References
Page Components					
Objects					
Static Resources					
Packages					

Open Filter Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View Status Progress Problems

Developer Console - Google Chrome

orgfarm-5df1e0805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

50% Reset

```
1 * @ISODLL11000 methEmailScheduler 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 = '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 *
36 *             Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
37 *
38 *         }
39 *
40 *     }
41 *
42 * }
```

Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The left sidebar has sections for Email, Custom Code (with Apex Classes selected), Environments, and a search bar. The main area displays the Apex Class MonthlyEmailScheduler. The Apex Class Detail section shows the class name, namespace prefix (MonthlyEmailScheduler), created by Sowmya Team on 6/23/2025, 2:46 AM, and status Active. The code editor shows the following Apex code:

```
1 global class MonthlyEmailScheduler implements Scheduleable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendMonthlyEmails();
6         }
7     }
8     public static void sendMonthlyEmails() {
9         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
10        for (Tenant__c tenant : tenants) {
11            ...
12        }
13    }
14}
15
16public static void sendMonthlyEmails() {
17    List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18    for (Tenant__c tenant : tenants) {
19        ...
20    }
21}
```

The screenshot shows the Lease Management detail page for tenant Aswini. The left sidebar has sections for Lease Management, Payment, Tenants (selected), property, and lease. The main area shows the tenant details: Tenant Name (Aswini), Owner (Sowmya Team), Email (aswinimaraadi15@gmail.com), Phone ((905) 223-5567), status (Leaving), and property (Imran). The activity sidebar shows a list of actions: New Case, Edit, New Opportunity, New Lead, Delete, Clone, Change Owner, Refresh, Printable View, Submit for Approval, and Edit Labels. It also displays the message "No activities to show. 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."

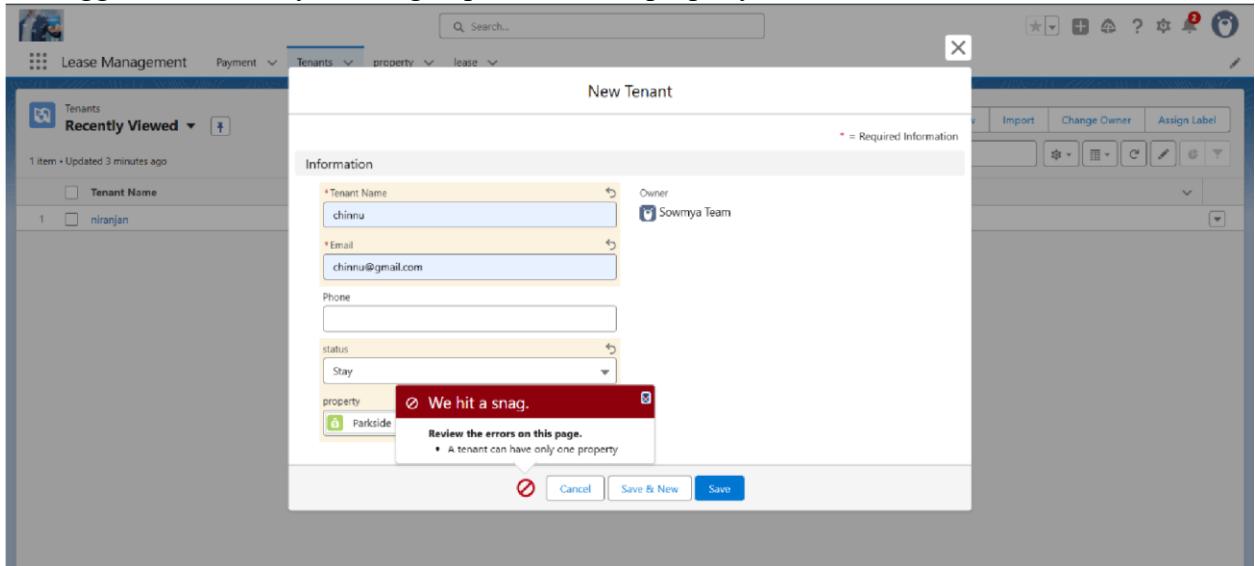
The screenshot shows a Salesforce Lightning page for 'Lease Management'. The top navigation bar includes tabs for 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. A green banner at the top right indicates 'Tenant was submitted for approval.' The main area displays a form with fields for 'Tenant Name' (Aswini), 'Email' (aswiniamaraadi15@gmail.com), 'Phone' ((905) 223-5567), 'status' (Leaving), and 'property' (Imran). The 'Owner' field is set to 'Sowmya Team'. Below the form, it shows 'Created By' (Sowmya Team) and 'Last Modified By' (Sowmya Team). A 'Details' tab is selected. On the right side, there is an 'Activity' sidebar with sections for 'Upcoming & Overdue' and 'No past activity. Past meetings and tasks marked as done show up here.'

The screenshot shows a Salesforce Lightning page for a 'Process Instance Step' titled 'Tenant Approval'. The status is 'Approved'. The top navigation bar includes tabs for 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. A green banner at the top right indicates 'Aswini Approval'. The main area displays a table with columns for 'Submitter' (Sowmya Team), 'Date Submitted' (Jun 27, 2025), 'Actual Approver' (Sowmya Team), and 'Assigned To' (Sowmya Team). Below this is a 'Details' section for 'Approval Details' with fields for 'Tenant Name' (Aswini), 'Owner' (Sowmya Team), and 'property' (Imran). On the right side, there is a 'Notifications' sidebar showing five notifications all from 'Aswini' indicating approval requests.

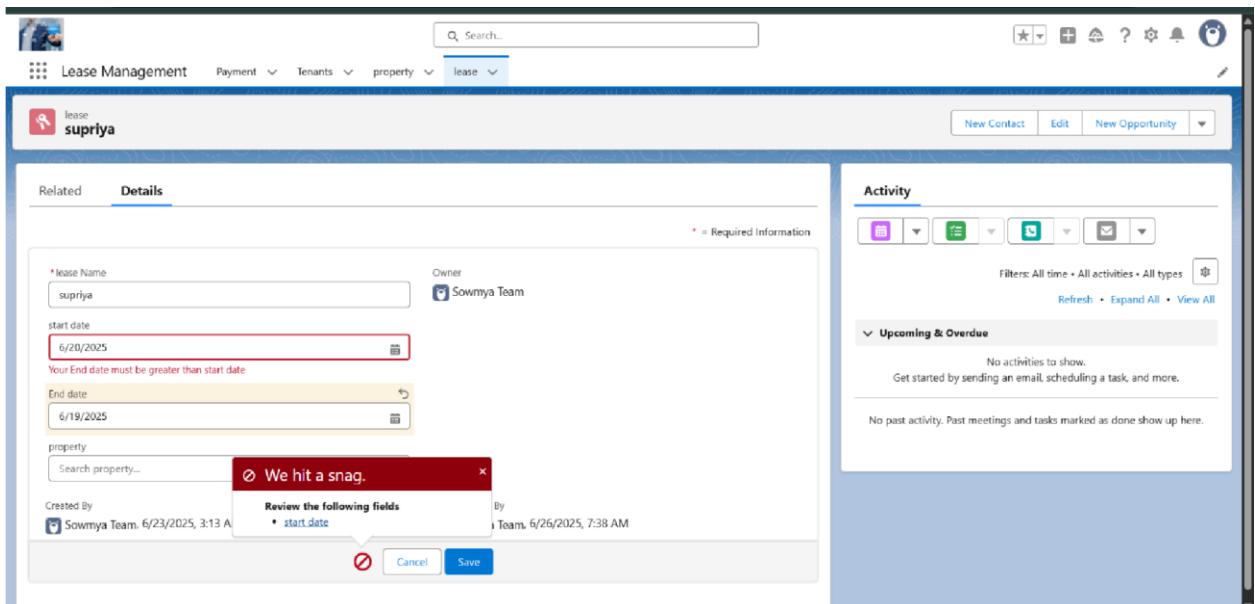
FUNCTIONAL AND PERFORMANCE TESTING

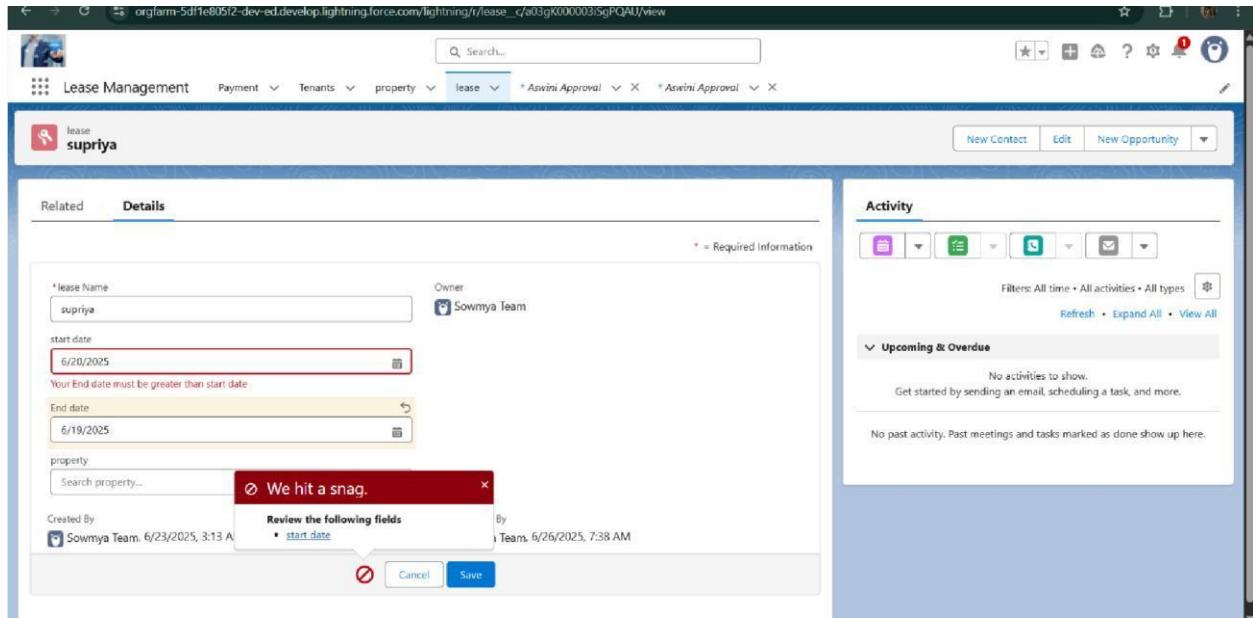
Performance Testing

- Trigger validation by entering duplicate tenant-property records

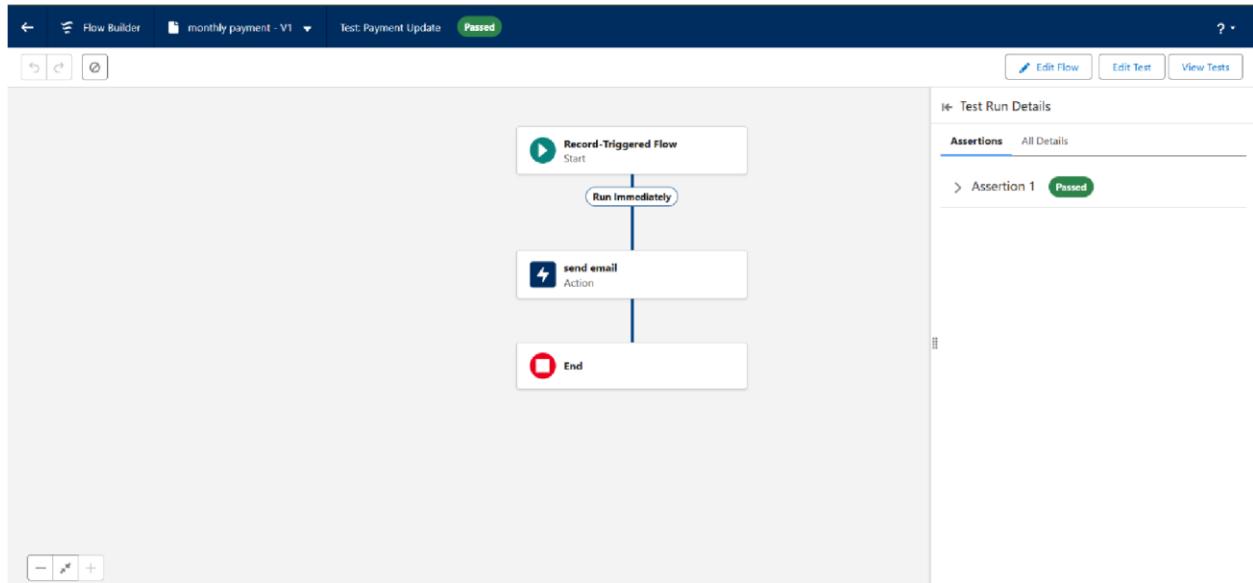


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

Tenant: nirajan

Related Details

* = Required Information

Tenant Name nirajan	Owner Sowmya Team
Email niranjan1506@gmail.com	
Phone	
status Stay	
property Parksid Lofts	

Created By
Sowmya Team, 6/23/2025, 2:33 AM

Last Modified By
Sowmya Team, 6/23/2025, 3:58 AM

Cancel **Save**

Notifications

- Approval request for the tenant is approved nirajan (a few seconds ago)
- Approval request for the tenant is rejected nirajan (Jun 23, 2025, 4:29 PM)
- Approval request for the tenant is approved nirajan (Jun 23, 2025, 4:25 PM)
- Approval request for the tenant is approved nirajan (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)

Tenant: nirajan

Approval History (6+)

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

Payment (2)

Payment Name
Jack
Rahul

New Contact **Edit** **New Opportunity**

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 a table of custom object tabs with columns for Action, Label, Tab Style, and Description. The tabs listed are:

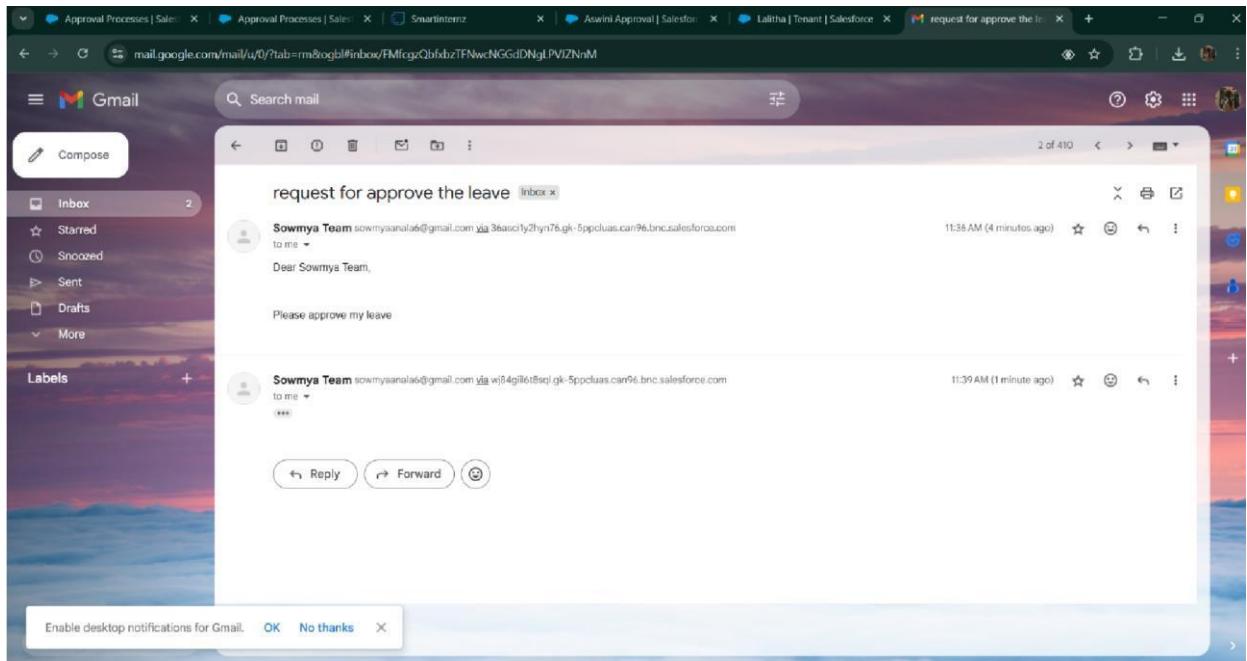
Action	Label	Tab Style	Description
Edit Del	lease	Key	
Edit Del	Payment	Credit card	
Edit Del	property	Back	
Edit Del	Tenants	Map	

- Email alerts

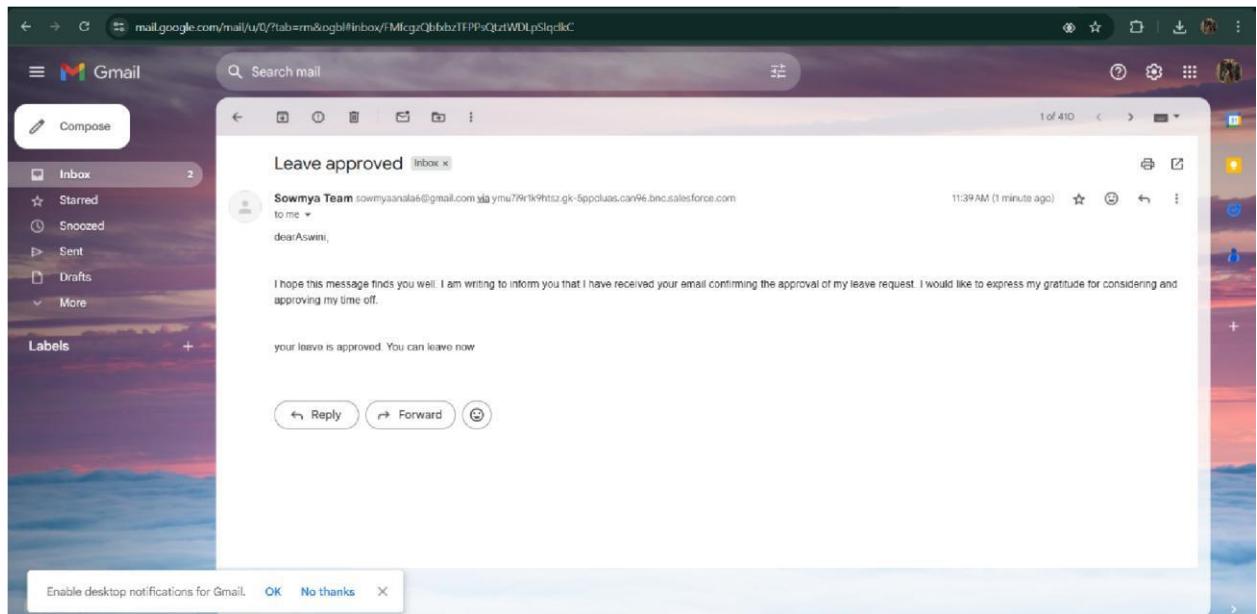
The screenshot shows the 'Lease Management' application interface. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. Below the navigation is a search bar and a toolbar with various icons. The main area is titled 'Approval History' and shows a table with 8 items. The table columns are: Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The data in the table is as follows:

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

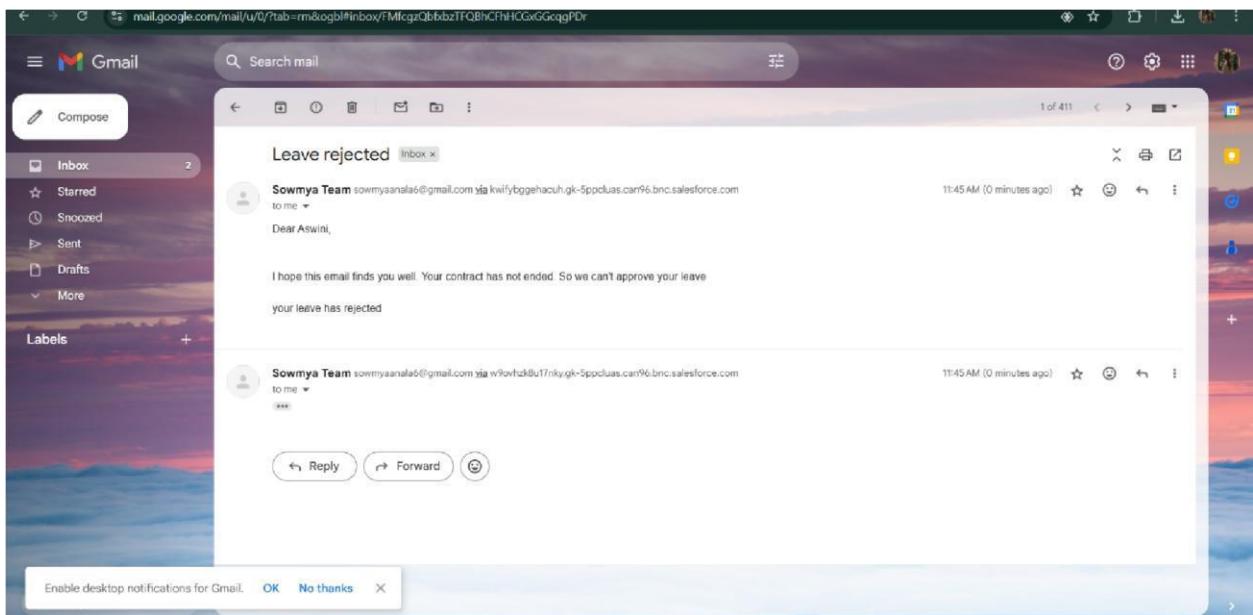
- Request for approve the leave



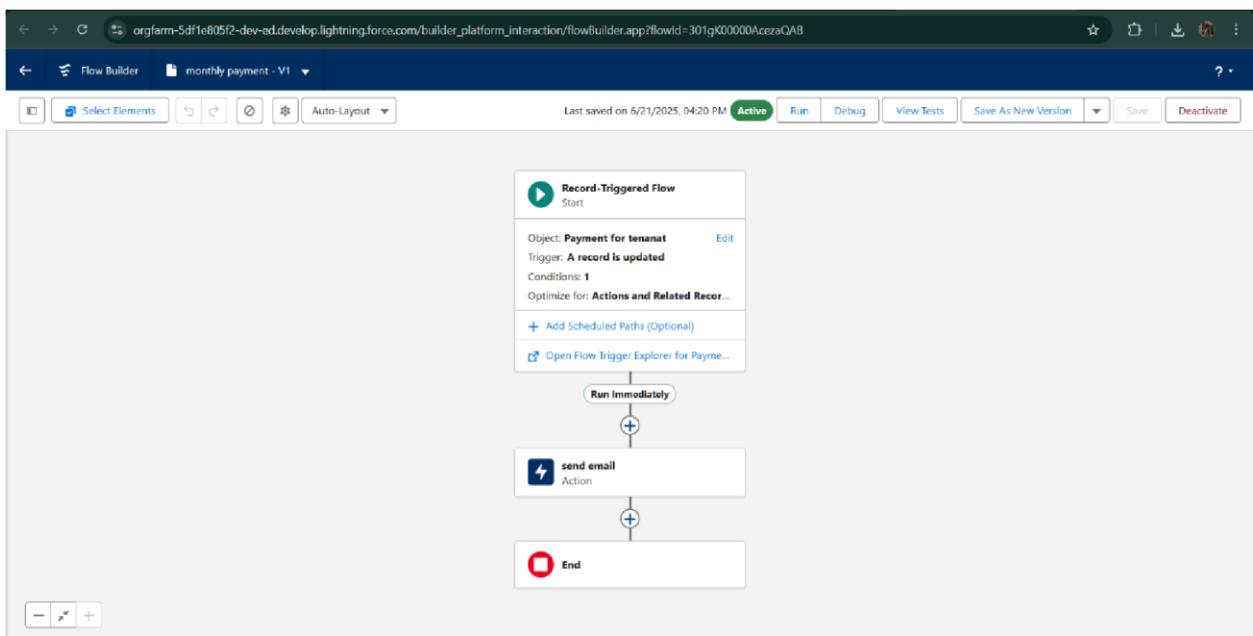
- Leave approved



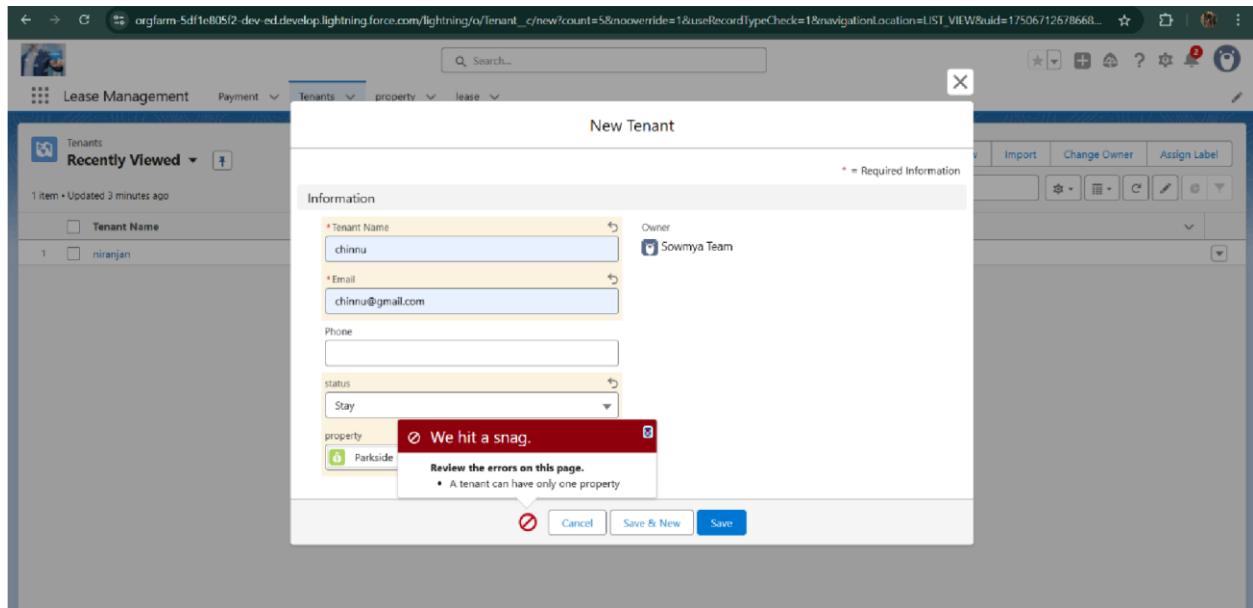
- Leave rejected



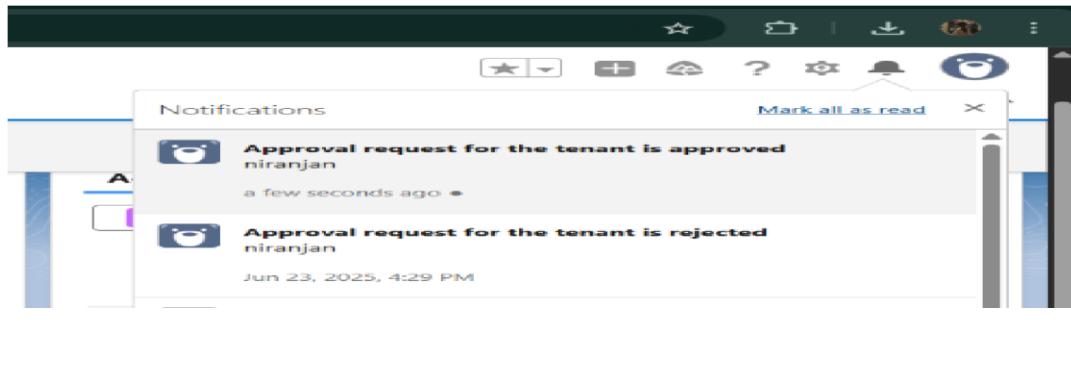
- Flow runs



- 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});

    }

}
```