

LeaseManagement

College Name: AYYAN THIRUVALLUVAR COLLEGE OF ARTS &
SCIENCE
College Code: BRU4m

TEAM ID : NM2025TMID27906

TEAM MEMBERS : 4

Team Leader : S.LAKSHANA

Email: lakshanashanmugam695@gmail.com

TeamMember1 : S.VINITHA

Email:vini2312vini@gmail.com

TeamMember2 : S.LAKSHMIPRIYA

Email: lashupriyal@gmail.com

TeamMember3: R.DIVYA

Email: divyaseetha212@gmail.com

1.INTRODUCTION

Project Overview

The Lease Management System is a Sales force –based application designed to stream line 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.



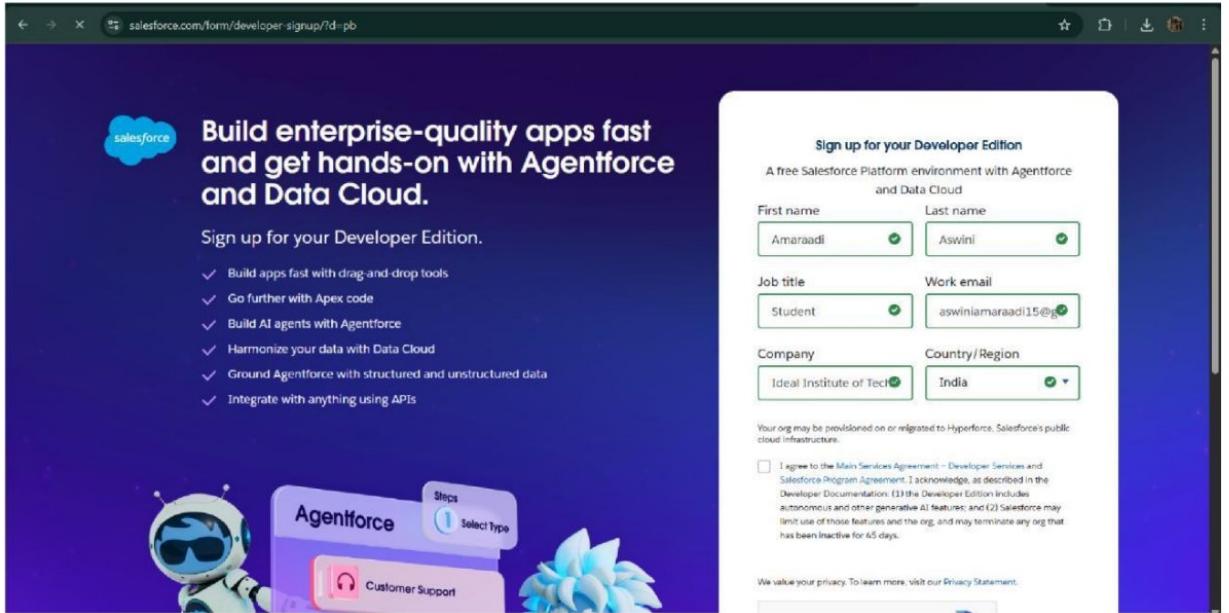
Purpose

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

The screenshot shows the Salesforce Setup Object Manager interface. The left sidebar lists various object configuration options: 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 main content area displays the details for the 'Tenant' object. The 'Details' tab is selected. The API Name is set to 'Tenant__c'. The object is defined as 'Custom'. The Singular Label is 'Tenant' and the Plural Label is 'Tenants'. On the right, there are sections for 'Enable Reports' (checked for Track Activities, Track Field History, and Deployment Status), 'Deployment Status' (set to 'Deployed'), and a link to the 'Standard salesforce.com Help Window'.

The screenshot shows the Salesforce Setup Object Manager interface. The left sidebar lists various object configuration options: 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 main content area displays the details for the 'lease' object. The 'Details' tab is selected. The API Name is set to 'lease__c'. The object is defined as 'Custom'. The Singular Label is 'lease' and the Plural Label is 'lease'. On the right, there are sections for 'Enable Reports' (checked for Track Activities, Track Field History, and Deployment Status), 'Deployment Status' (set to 'Deployed'), and a link to the 'Standard salesforce.com Help Window'.

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' followed by 'Payment for tenantat'. On the left, a sidebar lists various configuration options under 'Details'. The main content area is titled 'Details' and contains sections for 'Description' and 'Fields & Relationships'. In the 'Description' section, fields like API Name ('Payment_for_tenantat_c'), Singular Label ('Payment for tenant'), and Plural Label ('Payment') are defined. The 'Fields & Relationships' section is collapsed. On the right, there are buttons for 'Edit' and 'Delete', and a list of 'Enabled Reports' which includes 'Track Activities', 'Track Field History', 'Deployment Status' (set to 'Deployed'), and 'Help Settings'. A link to 'Standard salesforce.com Help Window' is also present.

- Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'SETUP > OBJECT MANAGER' followed by 'property'. On the left, a sidebar lists various configuration options under 'Details' and 'Fields & Relationships'. The 'Fields & Relationships' section is expanded, showing a table titled 'Fields & Relationships' with 9 items. The table has columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. The data is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(25)		
Owner	OwnerId	Lookup(User,Group)	<input checked="" type="checkbox"/>	
property	property__c	Lookup(property)	<input checked="" type="checkbox"/>	
property Name	Name	Text(80)	<input checked="" type="checkbox"/>	
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

Setup Home Object Manager

SETUP > OBJECT MANAGER
Payment for tenantat

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	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	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	CreatedBy	Lookup(User)		
End date	End_date__c	Date		
Last Modified By	LastModifiedBy	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User, Group)		✓
property	property__c	Lookup(property)		✓
start date	start_date__c	Date		

SETUP > OBJECT MANAGER

Tenant

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedBy	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

Lightning App Builder | App Settings | Pages | Lease ManagementDeveloper Name | Help

App Settings

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 Branding
*App Name <input type="text" value="Lease ManagementDeveloper Name"/>	Image <input type="button" value="Upload"/> Primary Color Hex Value <input type="text" value="#0070D2"/>
*Developer Name <input type="text" value="Lease_ManagementDeveloper_Name"/>	
Description <input type="text" value="Enter a description..."/>	Org Theme Options <input type="checkbox"/>
App Launcher Preview	
Lease ManagementDeveloper Name	

Screenshot of the Lightning App Builder interface showing the "Navigation Items" configuration screen.

The left sidebar shows the "App Settings" section with "Navigation Items" selected. The main area displays the "Navigation Items" configuration.

Available Items:

- Accounts
- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories

Selected Items:

- Payment
- Tenants
- property
- lease
- Approval Processes

Buttons for managing navigation items include "Create" and arrows for moving items between lists.

Screenshot of the Lightning App Builder interface showing the "User Profiles" configuration screen.

The left sidebar shows the "App Settings" section with "User Profiles" selected. The main area displays the "User Profiles" configuration.

Available Profiles:

- 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

Buttons for managing user profiles include arrows for moving profiles between lists.

To create a validation rule to a Lease Object

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area displays a 'Validation Rule Detail' for a rule named 'lease_end_date'. The rule's formula is 'End_date__c <= start_date__c', with an error message stating 'Your End date must be greater than start date'. The rule is marked as Active. A 'Help for this Page' link is visible in the top right.

The screenshot shows the 'Validation Rule Edit' screen for creating a new rule. The 'Rule Name' field is set to 'lease_end_date'. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains the expression 'End_date__c > start_date__c'. A tooltip for the 'ABS' function is open, explaining it returns the absolute value of a number. The 'Save' button is at the top right of the form.

Email Templates

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

Page Header: WhatsApp, pdf to world - Se..., Download file | i..., salesforce - Search, Recently Viewed, Classic Email Temp, tamil calendar 20..., Update.

Page Title: Classic Email Templates

Left Navigation: Email, Classic Email Templates (selected), Lightning Email Templates.

Text Email Template: tenant leaving

Email Template Detail:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	tenant_leaving
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	poovarasu_p [Change]
Description	
Created By	poovarasu_p, 9/2/2025, 11:22 PM
Modified By	poovarasu_p, 9/2/2025, 11:22 PM

Email Template:

Subject: request for approve the leave

Plain Text Preview:

```
Dear {!Tenant__c.CreatedBy},
```

System Status Bar: NIFTY +0.12%, ENG 11-09-2025, 13:01

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

Page Header: WhatsApp, pdf to world - Se..., Download file | i..., salesforce - Search, Recently Viewed, Classic Email Temp, tamil calendar 20..., Update.

Page Title: Classic Email Templates

Left Navigation: Email, Classic Email Templates (selected), Lightning Email Templates.

Text Email Template: Leave approved

Email Template Detail:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Leave_approved
Template Unique Name	Leave_approved
Encoding	Unicode (UTF-8)
Author	poovarasu_p [Change]
Description	
Created By	poovarasu_p, 9/2/2025, 11:23 PM
Modified By	poovarasu_p, 9/2/2025, 11:23 PM

Email Template:

Subject: Leave approved

Plain Text Preview:

```
dear {!Tenant__c.Name},
```

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

your leave is approved. You can leave now

System Status Bar: NIFTY +0.10%, ENG 11-09-2025, 13:03

Screenshot of the Salesforce Classic Email Templates page.

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	poovarasu_p [Change]
Description	
Created By	poovarasu_p 9/2/2025, 11:24 PM
Modified By	poovarasu_p 9/2/2025, 11:27 PM

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

Screenshot of the Salesforce Classic Email Templates page.

Text Email Template
tenant payment

Preview your email template below.

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	poovarasu_p [Change]
Description	
Created By	poovarasu_p 9/2/2025, 11:27 PM
Modified By	poovarasu_p 9/2/2025, 11:28 PM

Email Template

Subject: Confirmation of Successful Monthly Payment

Plain Text Preview:

Dear {Tenant__c Email__c},

The screenshot shows the Salesforce Setup interface under the 'Email' section, specifically the 'Classic Email Templates' page. A search bar at the top left contains 'EMAIL TE'. The main content area displays the 'Tenant Email' template details:

- Email Template Detail**
- Email Template Name:** Tenant Email
- Template Unique Name:** Tenant_Email
- Encoding:** Unicode (UTF-8)
- Author:** poovarasu_p [Change]
- Description:** Created by poovarasu_p, 9/2/2025, 11:26 PM
- Available For Use:** checked
- Last Used Date:** Times Used
- Modified By:** poovarasu_p, 9/2/2025, 11:26 PM

Below the detail section is a preview pane titled 'Email Template' containing the subject line 'Urgent: Monthly Rent Payment Reminder' and a plain text preview: 'Dear {Tenant__c.Name},'. The bottom status bar shows a Windows taskbar with various icons and the date/time '11-09-2025 13:06'.

Approval processor

The screenshot shows the Salesforce Setup interface under the 'Data' section, specifically the 'Approval Processes' page. A search bar at the top left contains 'appro'. The main content area displays the 'Approval Processes' list:

- Manage Approval Processes For:** Tenant
- Active Approval Processes:**
 - Action: Edit | Deactivate, Process Order: 1, Approval Process Name: check for vacant, Description:
- Inactive Approval Processes:** No approval processes available

A help section on the right provides steps for creating approval processes:

1. Read the help topic
2. View the checklist
3. Create a custom user hierarchical relationship field
4. Create email templates
5. Create an approval process using either the Jump Start or Standard Wizard
6. Add Approval History Related List to all page layouts
7. Activate the process to deploy to your users

The bottom status bar shows a Windows taskbar with various icons and the date/time '11-09-2025 13:08'.

Initial submission action-final approval action-final rejection action-approval set

The screenshot shows the Salesforce Approval Processes page. It includes the following sections:

- Initial Submission Actions:** Contains one entry: Action: Record Lock, Type: Email Alert, Description: Lock the record from being edited, and a note: please approve my leave.
- Approval Steps:** Contains one step: Step Number 1, Name: Manager Approval, Description: Lock the record from being edited, Criteria: Manually Chosen, and Reject Behavior: Final Rejection.
- Final Approval Actions:** Contains two entries: Action: Record Lock, Type: Email Alert, Description: Lock the record from being edited, and a note: Tenant leaving.
- Final Rejection Actions:** Contains two entries: Action: Record Lock, Type: Email Alert, Description: Unlock the record for editing, and a note: your request for leave is rejected.
- Recall Actions:** An empty section.

Apex trigger

Test apex

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

- Trigger:** testHandler.apxc
- Test Class:** testapxt
- Code:**

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

- Logs:** Logs tab is selected.
- Tests:** Tests tab is present.
- Checkpoints:** Checkpoints tab is present.
- Query Editor:** Query Editor tab is present.
- View State:** View State tab is present.
- Progress:** Progress tab is present.
- Problems:** Problems tab is selected.

Testhandler apex

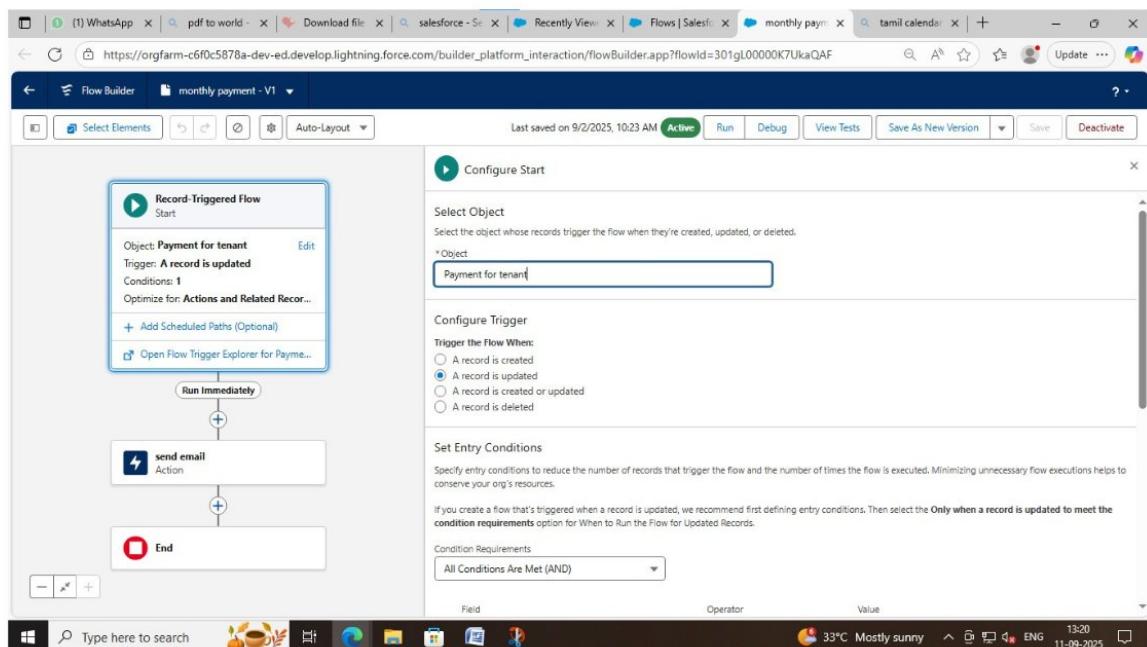
The screenshot shows the Salesforce Developer Console in Microsoft Edge. The URL is https://orgfarm-c6f0c5878a-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar shows 'testHandler.apxc' is active. The code editor contains the following Apex code:

```
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        for (Tenant__c newTenant : newList) {
14
15    }
```

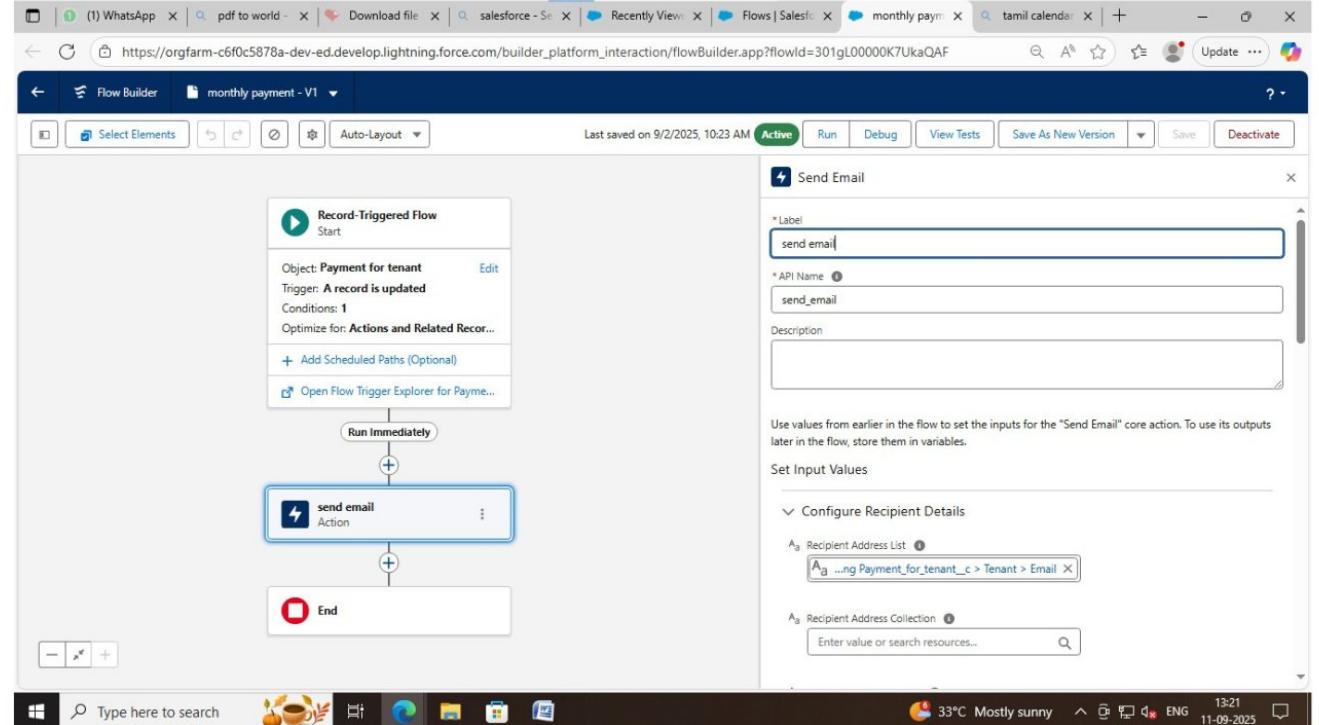
The 'Problems' tab is selected, showing no errors or warnings.

Flows

Configure Start



Send Email



- Scheduled monthly reminder emails using Apex class



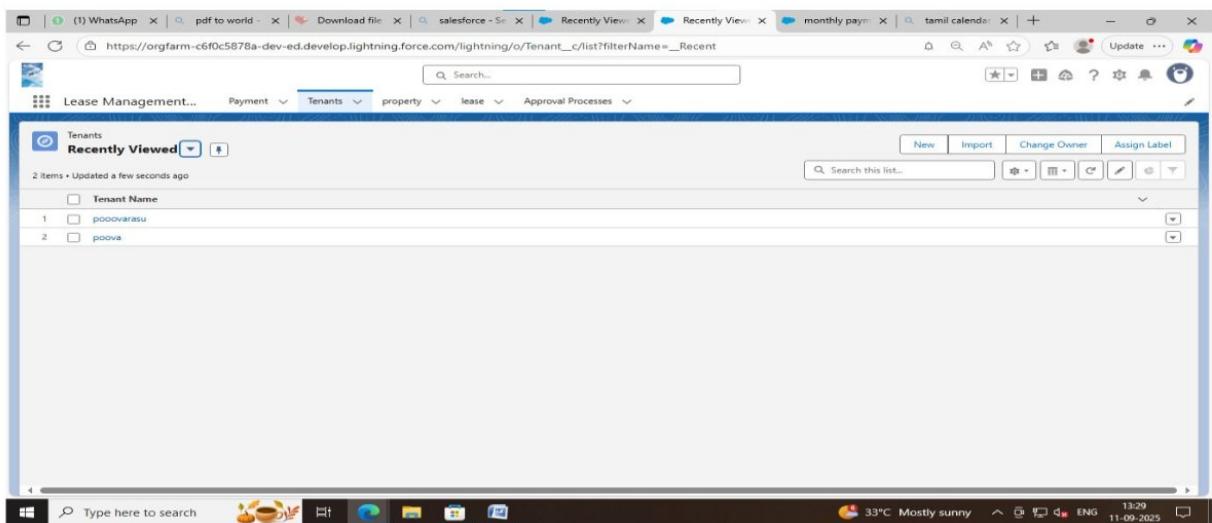
```
1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10
11         }
12
13     }
14
15
16     public static 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     }
36 }
```

```

globalclass Monthly Email Scheduler implements Schedulable {
    globalvoid execute(Schedulable Context sc) {
        Integer current Day = Date. today().day();
        if(current Day == 1) {
            send Monthly Emails();
        }
    }
    publicstatic void send Monthly Emails() {
        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});
        }
    }
}

```

Testing the approval process



Tenant approval access

The screenshot shows a Salesforce Lightning interface for a tenant record. The top navigation bar includes links for WhatsApp, pdf to world, Download file, salesforce, Recently Viewed, pooovarasu, monthly paym, tamil calendar, and Update. The main header shows "Lease Management..." and "Tenants". The page title is "Tenant: pooovarasu". The left sidebar has tabs for Related and Details. The Details tab displays the following fields:

- Tenant Name: pooovarasu
- Email: ppoovarasu614@gmail.com
- Phone: (956) 687-8849
- status: Stay
- property: pk
- check for payment: Paid
- Created By: pooovarasu p, 9/1/2025, 10:28 PM

The Owner field shows "pooovarasu.p". The right sidebar is titled "Activity" and includes sections for "Upcoming & Overdue" and "Past activity". The system status bar at the bottom shows a search bar, icons for various apps, and a weather forecast of 33°C Mostly sunny.

Submit for approval

The screenshot shows a Salesforce Lightning interface with a "Submit for Approval" dialog box overlaid on a tenant record. The dialog box contains a "Comments" section with the text "hey hi". The "Cancel" and "Submit" buttons are visible at the bottom of the dialog. The background shows the same tenant record details as the previous screenshot, including the "Details" tab and the "Activity" sidebar. The system status bar at the bottom shows a search bar, icons for various apps, and a weather forecast of 33°C Mostly sunny.

- Approval process notifications

The screenshot shows a Salesforce Lightning interface for managing tenants. The main page displays the details of a tenant named "poovarasu". The notifications sidebar on the right lists several messages related to the approval request:

- poovarasu p is requesting approval for tenant Tenant Name: poovarasu • Owner: poovarasu p a few seconds ago
- Approval request for the tenant is approved poovarasu a day ago
- Approval request for the tenant is approved poovarasu Sep 10, 2025, 11:34 AM
- Approval request for the tenant is approved poova Sep 8, 2025, 8:59 PM
- New Guidance Center learning resource available Define Your Sales Process Learn how to guide reps through the sales process. Sep 1, 2025, 1:55 PM

The screenshot shows a Salesforce Lightning interface for a process instance step named "Tenant Approval". The step is marked as "Approved". The page displays the following details:

Submitter	Date Submitted	Actual Approver	Assigned To
poovarasu p	Sep 11, 2025	poovarasu p	poovarasu p

Details

Approval Details	
Tenant Name	poovarasu
Owner	poovarasu p

Approver Comments

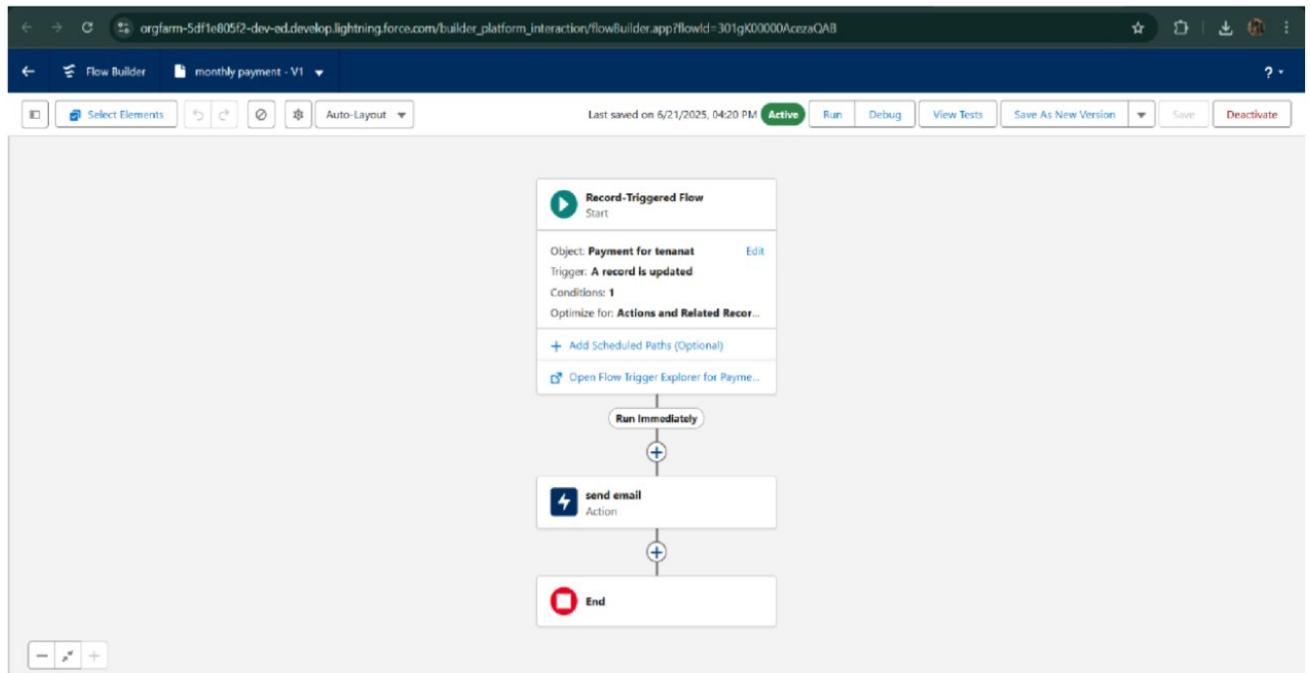
poovarasu p
hv
Sep 11, 2025, 1:06:59 AM

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 shows 'User Interface' and 'Tabs' selected. The main content area is titled 'Custom Tabs' and contains three sections: 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section lists tabs for 'Lease', 'Payment', 'Property', and 'Tenant', each with a 'Tab Style' icon (Keys, Credit Card, Back, Map) and a 'Description' column. The 'Web Tabs' and 'Visualforce Tabs' sections both state 'No Web Tabs have been defined' and 'No Visualforce Tabs have been defined' respectively.



ADVANTAGES & DISADVANTAGES

Advantages of Lease Management

Advantage	Description
1. Cost Predictability	Regular lease payments make budgeting easier and reduce unexpected costs.
2. Reduced Upfront Costs	Leasing avoids the need for large capital outlay, especially useful for startups or growing businesses.
3. Flexibility	Short- or long-term lease options allow businesses to scale up or down based on need.
4. Risk Mitigation	Maintenance and repair responsibilities may fall to the lessor, reducing liability for the lessee.
5. Asset Upgrades	Leases can be structured to allow regular upgrades to newer equipment or properties.

Disadvantages of Lease Management

Disadvantage	Description
1. Long-Term Cost	Over time, leasing can be more expensive than owning the asset outright.
2. Limited Control	The lessee may have restrictions on property use or modification.
3. Renewal Risks	Lease terms may change unfavorably at renewal or the lease may not be renewed at all.
4. Contract Complexity	Lease agreements can be legally complex, requiring specialized management and review.
5. Liability for Breach	Failure to comply with lease terms can result in penalties or legal consequences.

Disadvantages of Lease Management

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

- SourceCode: Provided in Apex Classes and Triggers

Test.apxt:

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

testHandler.apxc:

```
public class
testHandler {
    public static void
    preventInsert(List<
        Tenantc> newList)
    {
        Set<Id>
        existingPropertyIds
        = newSet<Id>()
            for(Tenantc existingTenant:[SELECT Id, Propertyc FROM Tenantc WHERE
Propertyc != null])
            {
                existingPropertyIds.add(existingTenant.Propertyc);
            }
    }
}
```

```

}for(TenantcnewTenant:
newlist) {

    if (newTenant.Propertyc != null &&
existingPropertyIds.contains(newTenant.Propertyc)){newTenant.addError('A
tenantcanhaveonlyoneproperty');

}

}

}

```

MothlyEmailScheduler.apxc:

```

globalclassMonthlyEmailSchedulerimplementsSchedulable{ global
void execute(SchedulableContext sc) { Integer currentDay =
Date.today().day(); if (currentDay == 1) {
sendMonthlyEmails();

}

}public static void
sendMonthlyEmails(){List<Tenantc>
tenants=[SELECTId,EmailcFROM
Tenantc]; for (Tenantc tenant :
tenants){

StringrecipientEmail=tenant.Emailc;
StringemailContent='Itrustthisemailfindsyouwell.Iamwritingtoremindyou that the
monthly rentis dueYour timely payment ensures the smooth functioning of our rental
arrangement and helps maintain a positive living environment for all.';

StringemailSubject='Reminder:MonthlyRentPaymentDue';

```

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

Thank you!