

# **LEASE MANAGEMENT**

**College Name: Ayyan Thiruvalluvar  
college of art's & science**

**College Code:BRU4m**

**TEAM ID:NM2025TMID27922**

**TEAM MEMBERS:4**

**Team Leader : S ASHA**

**Email: asha284asha@gmail.com**

**Team Member1:M NANDHINI**

**Email: nandhinimakali@gmail.com**

**Team Member2:D SRIDHAR**

**Email: sridh4858@gmail.com**

**Team Member3:P RAMESHKUMAR**

**Email: rameshk98903@gmail.com**

## **1. INTRODUCTION**

### **Project Overview**

The Lease Management System is a Sales Force- based application designed to streamline the process associated with leasing real properties. It handles tenant management, lease

contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.



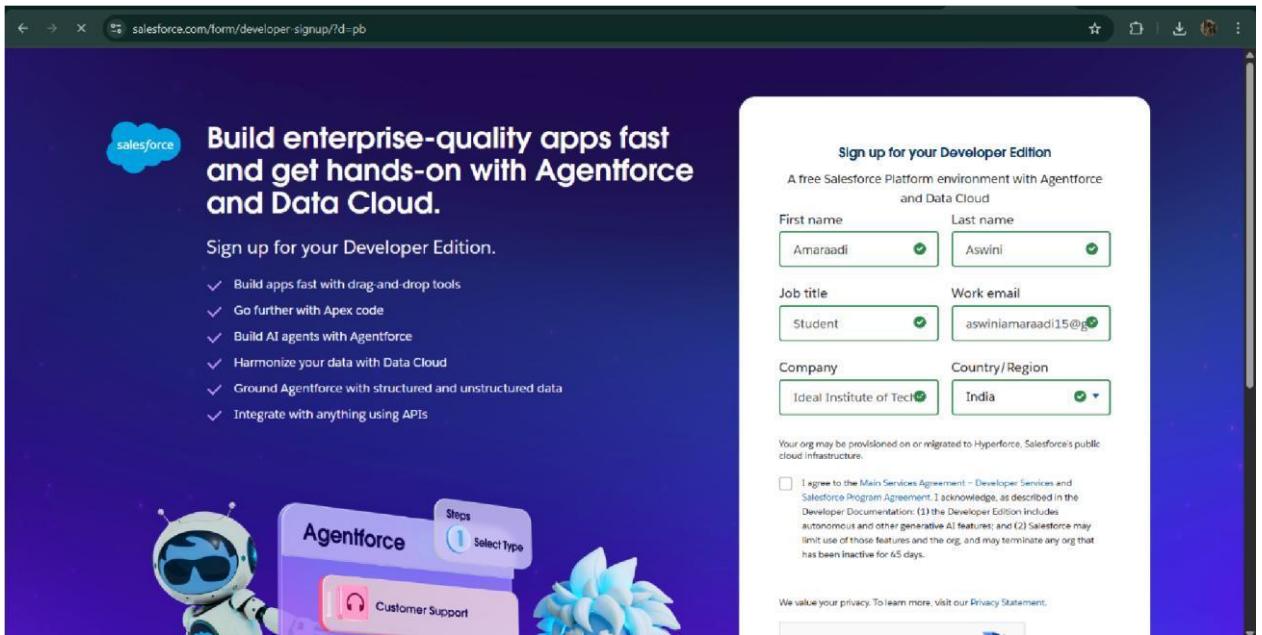
## Purpose

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

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 tabs like Fields & Relationships, Page Layouts, and Record Types. The main 'Details' tab is selected, showing the API name 'Tenant\_\_c', a custom field, and labels for both singular and plural forms. On the right, there are sections for enabling reports, tracking activities and field history, and deployment status. Buttons for Edit and Delete are at the bottom right.

SETUP > OBJECT MANAGER

## lease

Details

Description

API Name: lease\_\_c

Custom:

- Singular Label: lease
- Plural Label: lease

Enable Reports:

- ✓ Track Activities
- ✓ Track Field History

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Edit Delete

This screenshot shows the Salesforce Object Manager interface for the 'lease' object. Similar to the 'Tenant' object, it has an API name 'lease\_\_c', a custom field, and labels for singular and plural forms. It includes sections for reports, tracking, deployment status, and help settings. The left sidebar contains the same configuration tabs as the 'Tenant' screen.

The screenshot shows the Salesforce Setup interface under the Object Manager. A sidebar on the left lists various object configuration options like Fields & Relationships, Page Layouts, and Record Types. The main pane displays the 'Details' section for the 'Payment for tenant' object. It includes fields for API Name ('Payment\_for\_tenant\_\_c'), Singular Label ('Payment for tenant'), Plural Label ('Payment'), and several checkboxes for reporting and tracking features. Buttons for Edit and Delete are at the top right.

- Configured fields and relationships

The screenshot shows the Salesforce Setup interface under the Object Manager. The sidebar lists configuration options. The main pane shows the 'Fields & Relationships' section for the 'property' object. It lists nine fields: Address, Created By, Last Modified By, Name, Owner, property, property\_Name, sqft, and Type. The 'Owner' field is indexed, while others are not. A toolbar at the top of the list view includes buttons for Quick Find, New, Deleted Fields, Field Dependencies, and Set History Tracking.

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"/>	
sqft	sqft__c	Text(18)	<input checked="" type="checkbox"/>	
Type	Type__c	Picklist	<input checked="" type="checkbox"/>	

Setup > Object Manager

### Payment for tenant

Fields & Relationships				
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount_c	Number(18, 0)		
check for payment	check_for_payment_c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User, Group)		✓
Payment date	Payment_date_c	Date		
Payment Name	Name	Text(80)		✓

Setup > Object Manager

### lease

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

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

\* Developer Name

Description

**App Branding**

Image  Primary Color Hex Value

Org Theme Options  Use the app's image and color instead of the org's custom theme

**App Launcher Preview**

Lease ManagementDevelo...

The screenshot shows the Lightning App Builder interface for configuring navigation items. On the left, a sidebar lists 'App Settings' sections: 'App Details & Branding', 'App Options', 'Utility Items (Desktop Only)', and 'Navigation Items'. The 'Navigation Items' section is selected and expanded, showing a list of available items on the left and selected items on the right.

**Available Items**

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

**Selected Items**

- Payment
- Tenants
- property
- lease
- Approval Processes

Below the main interface is a Windows taskbar with various pinned icons and a search bar.

The screenshot shows the Lightning App Builder interface for configuring user profiles. The 'User Profiles' section is selected in the sidebar. It displays a list of available profiles on the left and a selected profile on the right.

**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

## To create a validation rule to a Lease Object

The screenshot shows the Salesforce Setup interface for the 'lease' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the 'Validation Rule Detail' for a rule named 'lease\_end\_date'. The rule is active and has the formula `End_date__c <= start_date__c`. The error message is 'Your End date must be greater than start date'. The 'Error Location' is set to 'start date'. The rule was created by 'Sowmya Team' on 6/19/2025, 5:37 AM, and modified by 'Sowmya Team' on 6/26/2025, 7:47 AM.

The screenshot shows the 'Validation Rule Edit' page for the 'lease' object. The rule name is 'lease\_end\_date' and it is active. The 'Error Condition Formula' is set to `End_date__c > start_date__c`. A tooltip for the formula explains that it displays an error if the end date is earlier than the start date. The formula editor includes fields for 'Insert Field' and 'Insert Operator', and a dropdown menu for 'Functions' with options like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. A 'Quick Tips' panel on the right provides links to 'Operators & Functions' and 'Help on this function'.

## Email Templates

The screenshot shows the Salesforce Setup interface under the 'Email' category, specifically the 'Classic Email Templates' section. A search bar at the top right contains the text 'Search Setup'. The main content area displays a 'Text Email Template' named 'tenant leaving'. The 'Email Template Detail' section shows the following information:

Email Template Name	Unfiled Public Classic Email Templates
tenant leaving	Available For Use <input checked="" type="checkbox"/>
tenant_leaving	Last Used Date
Unicode (UTF-8)	Times Used
poovarasu_p [Change]	
Description	
Created By poovarasu_p, 9/2/2025, 11:22 PM	Modified By poovarasu_p, 9/2/2025, 11:22 PM

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

Subject | request for approve the leave  
Plain Text Preview | Dear {!Tenant\_\_c.CreatedBy},

At the bottom of the page, there is a Windows taskbar with icons for search, file explorer, and other applications, along with a system tray showing the date and time.

This screenshot shows the same Salesforce setup interface and navigation as the first one. It displays a different email template, 'Leave approved'. The 'Email Template Detail' section shows:

Email Template Name	Unfiled Public Classic Email Templates
Leave approved	Available For Use <input checked="" type="checkbox"/>
Leave_approved	Last Used Date
Unicode (UTF-8)	Times Used
poovarasu_p [Change]	
Description	
Created By poovarasu_p, 9/2/2025, 11:23 PM	Modified By poovarasu_p, 9/2/2025, 11:23 PM

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

Subject | Leave approved  
Plain Text Preview | dear {!Tenant\_\_c.Name},  
I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.  
your leave is approved. You can leave now

At the bottom of the page, there is a Windows taskbar with icons for search, file explorer, and other applications, along with a system tray showing the date and time.

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.

**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 category, specifically the Classic Email Templates section. A search bar at the top left contains "EMAIL TE". The main content area displays the "Classic Email Templates" page with a header "SETUP" and "Classic Email Templates". Below the header, it says "Text Email Template" and "Tenant Email". A message "Preview your email template below." is present. The "Email Template Detail" section shows the following information:

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

Below the detail section is a preview area titled "Email Template" with a "Plain Text Preview" containing the text "Dear {!Tenant\_\_c.Name},". The status bar at the bottom shows "javascript:srcUp%27%2F00XgL000005PGBZ%3Fsetupid%3DCommunicationTemplatesEmail%26isdtp%3Dp1%27;".

## Approval processor

The screenshot shows the Salesforce Setup interface under the Approval Processes section. The left sidebar has a search bar with "appro" and categories: Data, Feature Settings, Process Automation, and Approval Processes (which is selected). A message "Didn't find what you're looking for? Try using Global Search." is at the bottom of the sidebar.

The main content area displays the "Approval Processes" page with a header "SETUP" and "Approval Processes". It includes a "Manage Approval Processes For" dropdown set to "Tenant". A list of steps for starting an approval process is shown:

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 "Active Approval Processes" table lists one process:

Action	Process Order	Approval Process Name	Description
Edit   Deactivate	1	check for vacant	

The "Inactive Approval Processes" section states "No approval processes available". The status bar at the bottom shows "Who is the world's ric... 13:08 11-09-2025".

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

The screenshot shows the Salesforce Setup interface for managing Approval Processes. The left sidebar is collapsed, and the main area displays the 'Approval Processes' configuration screen. The page is divided into several sections:

- Initial Submission Actions:** A table with one row: Action (Record Lock), Type (Record Lock), Description (Lock the record from being edited), and a note (please approve my leave). Buttons for 'Add Existing' and 'Add New' are at the top.
- Approval Steps:** A table with one row: Action (Show Actions), Step Number (1), Name (Manager Approval), Description (Manager Approval), Criteria (Manually Chosen), Assigned Approver (Manually Chosen), and Reject Behavior (Final Rejection).
- Final Approval Actions:** A table with one row: Action (Record Lock), Type (Record Lock), Description (Lock the record from being edited), and a note (Tenant leaving). Buttons for 'Add Existing' and 'Add New' are at the top.
- Final Rejection Actions:** A table with one row: Action (Record Lock), Type (Record Lock), Description (Unlock the record for editing), and a note (your request for leave is rejected). Buttons for 'Add Existing' and 'Add New' are at the top.
- Recall Actions:** An empty table with a header row for Action and Type, and a 'Description' column.

## Apex trigger

Test apex

The screenshot shows the Microsoft Edge Developer Console with the URL [https://orgfarm-c6f0c5878a-dev-ed.develop.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://orgfarm-c6f0c5878a-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The console is displaying an Apex trigger named `testHandler.apxc` with the file tab `test.apxt` selected. The code is as follows:

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

The developer console includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently active, showing no errors. The bottom status bar indicates the date and time as 11-09-2025 and 13:14, along with system status icons.

## Testhandler apex

Developer Console - Profile 1 - Microsoft Edge

https://orgfarm-c6f0c5878a-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

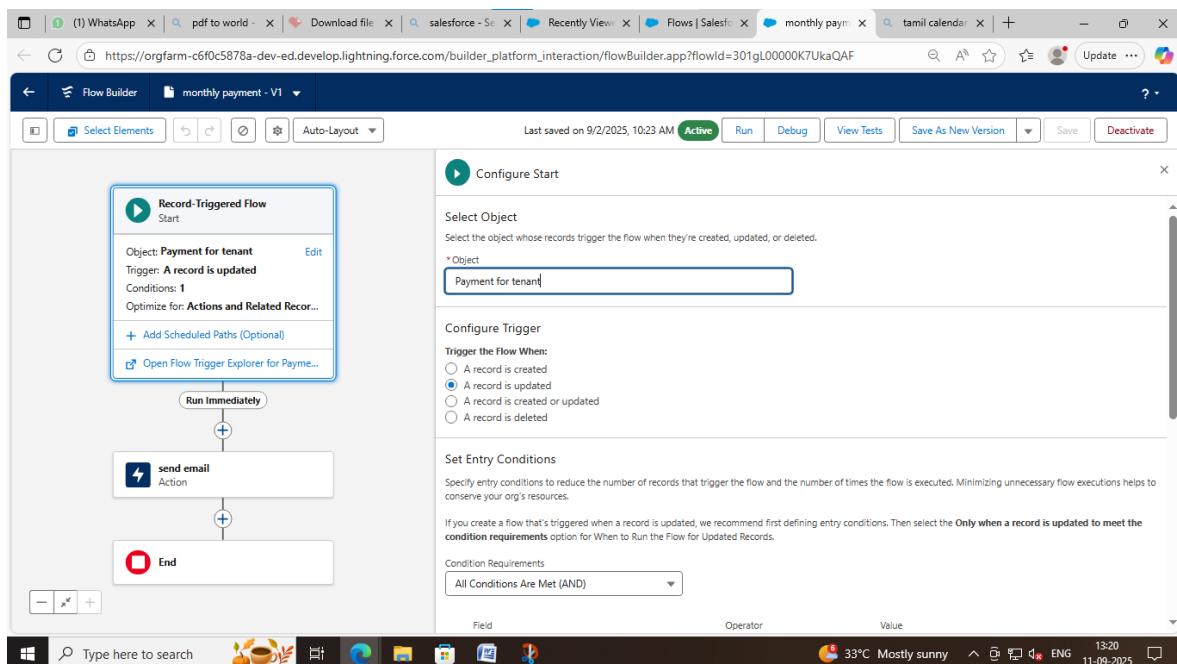
testHandler.apxc test.apxt

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             existingPropertyIds.add(existingTenant.Property__c);  
9         }  
10    }  
11  
12    for (Tenant__c newTenant : newList) {  
13  
14    }
```

## Flows

# Configure Start



## Send Email

The screenshot shows the Salesforce Flow Builder interface. A Record-Triggered Flow named "monthly payment - V1" is active. The flow starts with a "Record-Triggered Flow" step (Payment for tenant, Trigger: A record is updated), followed by a "Run Immediately" step, then a "send email" action step, and finally an "End" step. The "send email" step is selected, showing its configuration details: Label: send\_email, API Name: send\_email, and a description. The recipient details section shows a recipient address list: ...ng Payment\_for\_tenant\_c > Tenant > Email.

- Scheduled monthly reminder emails using Apex class

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendMonthlyEmails();
6         }
7     }
8
9     public static void sendMonthlyEmails() {
10        List<Tenant_c> tenants = [SELECT Id, Email__c FROM Tenant_c];
11        for (Tenant_c tenant : tenants) {
12            String recipientEmail = tenant.Email__c;
13            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';
14            String emailSubject = 'Reminder: Monthly Rent Payment Due';
15            Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
16            email.setToAddresses(new String[]{recipientEmail});
17            email.setSubject(emailSubject);
18            email.setPlainTextBody(emailContent);
19        }
20    }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```

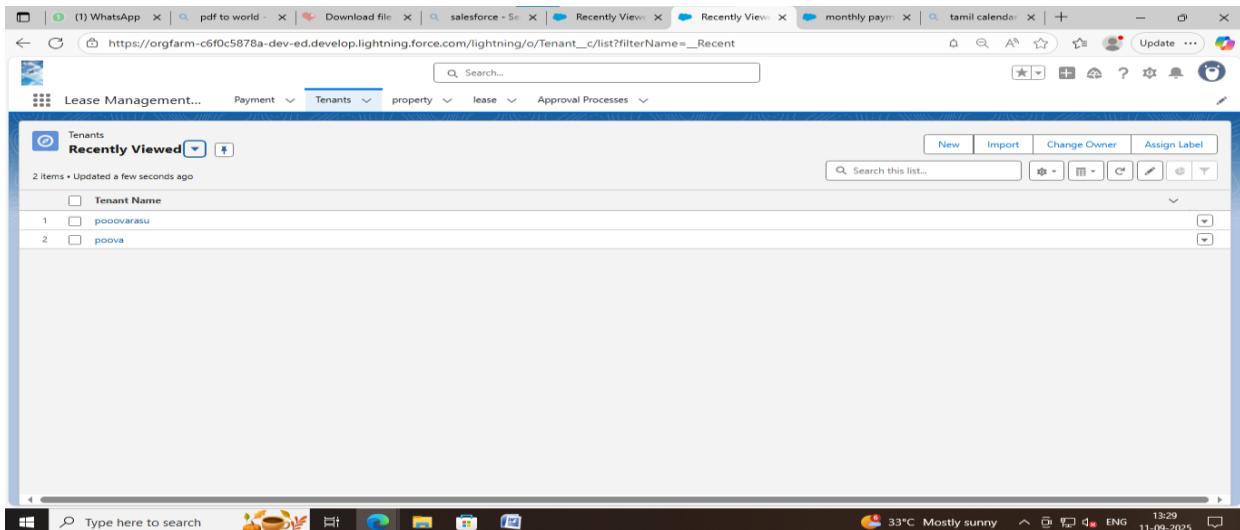
The screenshot shows the Salesforce Developer Console with the code for the MonthlyEmailScheduler apex class. The class implements the Scheduleable interface and contains methods for executing scheduled tasks and sending monthly emails. The code uses the Messaging.SingleEmailMessage class to send emails to tenants.

```

global class Monthly Email Scheduler implements Schedulable {
    global void execute(Schedulable Context sc) {
        Integer current Day = Date. today().day();
        if (current Day == 1) {
            send Monthly Emails();
        }
    }
    public static 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 tabs for Payment, Tenants, property, lease, and Approval Processes. The main content area displays the tenant's details: Name (poovarasu), Email (ppoovarasu614@gmail.com), Phone ((956) 687-8849), Status (Stay), and Property (pk). The record was created by poovarasu p on 9/1/2025 at 10:28 PM and last modified by poovarasu p on 9/8/2025 at 5:57 AM. To the right, there is an 'Activity' sidebar with sections for Upcoming & Overdue, and a note indicating no activities to show.

## Submit for approval

The screenshot shows a Salesforce Lightning interface with a 'Submit for Approval' modal overlaid. The modal contains a 'Comments' text area with the text 'hey hi'. At the bottom of the modal are 'Cancel' and 'Submit' buttons. The background shows the same tenant record details as the previous screenshot, including the activity sidebar.

- Approval process notifications

The screenshot shows a web browser window with a tab bar at the top containing various links like WhatsApp, pdf to world, Download file, salesforce, Recently Viewed, poovarasu, monthly paym, and tamil calendar. The main content area displays a tenant profile for "poovarasu". The profile includes fields for Tenant Name (poovarasu), Email (poovarasu614@gmail.com), Phone (956) 687-8849, status (Stay), property (pk), and check for payment (Paid). It also shows the Created By field (poovarasu p) and Last Modified By field (poovarasu p, 9/8/2025, 5:57 AM). To the right of the profile, a sidebar titled "Notifications" lists several recent messages:

- 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 bottom of the screen shows a Windows taskbar with icons for Start, Search, File Explorer, Edge, Task View, and others. The system tray indicates "Rain coming" and shows the date and time as 13:35 11-09-2025.

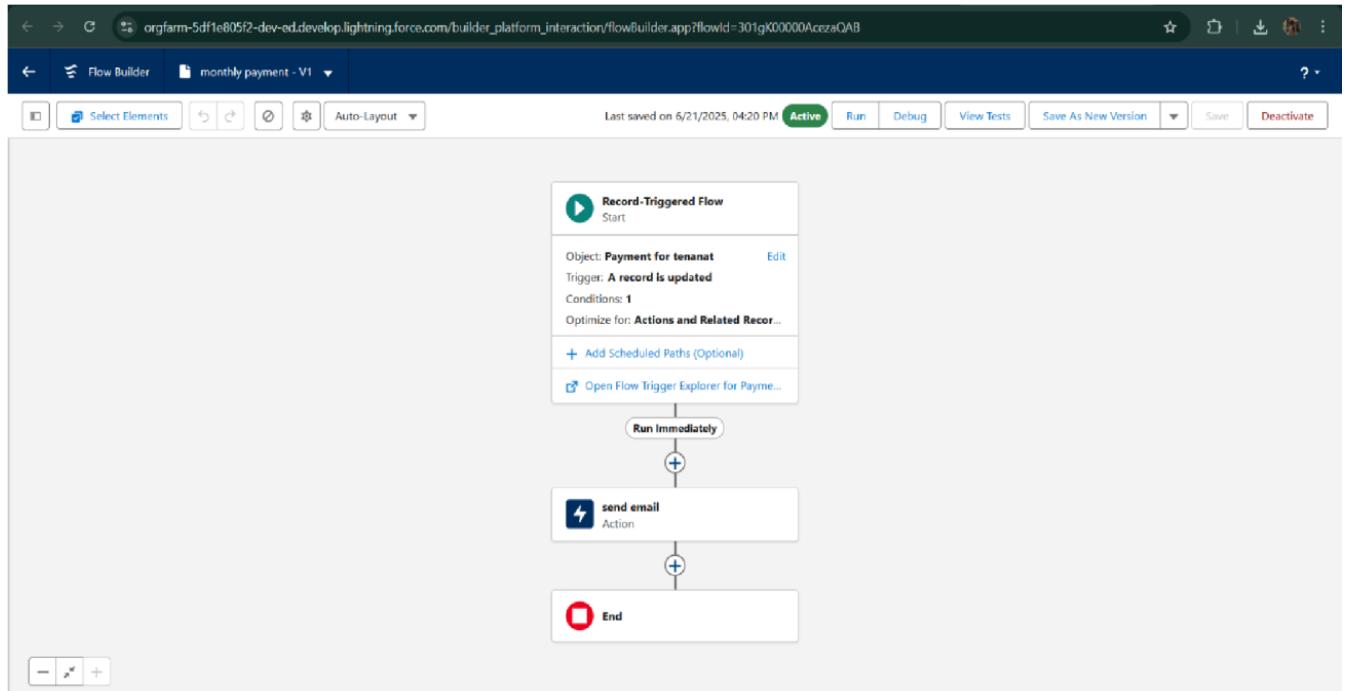
The screenshot shows a web browser window with a tab bar at the top containing various links like WhatsApp, pdf to world, Download file, salesforce, Recently Viewed, poovarasu A, monthly paym, tamil calendar, and others. The main content area displays a "Process Instance Step" for "Tenant Approval". The step status is shown as "Approved". The details section includes fields for Submitter (poovarasu p), Date Submitted (Sep 11, 2025), Actual Approver (poovarasu p), and Assigned To (poovarasu p). Below this, the "Details" section shows "Approval Details" with Tenant Name (poovarasu) and Owner (poovarasu p). To the right, the "Approver Comments" section shows a comment from "poovarasu p" (hv) dated Sep 11, 2025, 1:06:59 AM. The bottom of the screen shows a Windows taskbar with icons for Start, Search, File Explorer, Edge, Task View, and others. The system tray indicates "Result" and shows the date and time as 13:37 11-09-2025.

## RESULTS

# Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Tabs' section. It displays four custom object tabs: 'Lease' (Tab Style: Keys), 'Payment' (Tab Style: Credit card), 'property' (Tab Style: Sack), and 'Tenant' (Tab Style: Map). Each tab has an 'Edit | Del' button next to its name.



---

# ADVANTAGES & DISADVANTAGES

---

## Advantages of Lease Management

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

## Disadvantages of Lease Management

<b>Disadvantage</b>	<b>Description</b>
<b>1. Long-Term Cost</b>	Over time, leasing can be more expensive than owning the asset outright.
<b>2. Limited Control</b>	The lessee may have restrictions on property use or modification.
<b>3. Renewal Risks</b>	Lease terms may change unfavorably at renewal or the lease may not be renewed at all.
<b>4. Contract Complexity</b>	Lease agreements can be legally complex, requiring specialized management and review.
<b>5. Liability for Breach</b>	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});  
    }  
}  
}
```