

# Lease Management

**college name:** TIRUPPUR KUMARAN COLLEGE FOR WOMEN

**College code:** bru3j

**TEAM ID: NM2025TMID28384**

**TEAM MEMBERS:**

**Team LeaderName:** SIEYD KANI A

**Email:** [asrhakkem@gmail.com](mailto:asrhakkem@gmail.com)

**Team member:** ANGEL KIRAN E

**Email:** [angelkiran2006@gmail.com](mailto:angelkiran2006@gmail.com)

**Team member:** BAVIDHRA C

**Email:** [bavidhra551@gmail.com](mailto:bavidhra551@gmail.com)

**Team member:** KOWSHIKA M

**Email:** [manikandan8383@gmail.com](mailto:manikandan8383@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.

**LEASE ABSTRACTION**

**PAYMENT  
PROCESSING**

**lease  
management**

**DOCUMENT  
MANAGEMENT**

**PROPERTY  
MAINTENANCE**

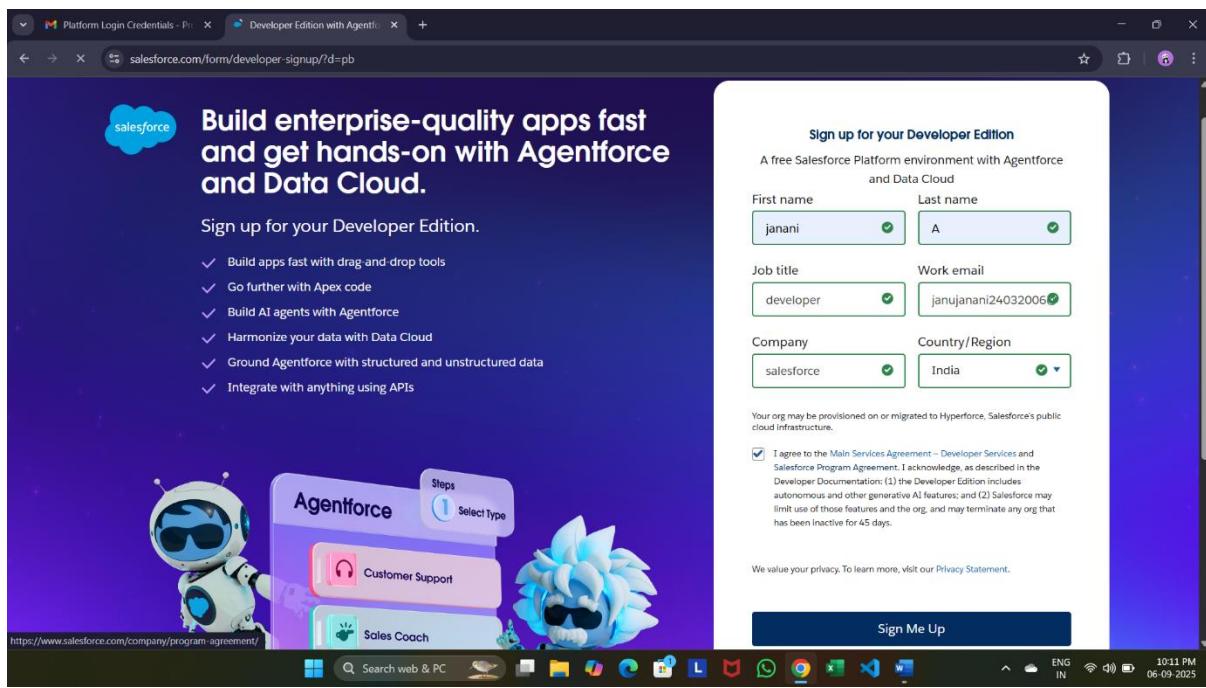
## **1.2 Purpose**

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 URL-<https://developer.salesforce.com/signup>



**Create objects:** property, Tenant, Lease, Payment

**SETUP > OBJECT MANAGER**

**property**

**Details**

Description

API Name: property\_\_c

Custom: ✓

Singular Label: property

Plural Label: property

Enable Reports: ✓

Track Activities: ✓

Track Field History: ✓

Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

**Edit | Delete**

The screenshot shows the Salesforce Setup interface for the Object Manager. The left sidebar lists various 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 is titled "Details" for the "Tenant" object. It includes sections for Description, API Name (set to "Tenant\_\_c"), Singular Label (set to "Tenant"), Plural Label (set to "Tenants"), and several checkboxes for Reports, Activities, Field History, and Deployment Status, all of which are checked. The status is set to "Deployed". A "Help Settings" link and a "Standard salesforce.com Help Window" link are also present. At the bottom right are "Edit" and "Delete" buttons.

The screenshot shows the Salesforce Setup interface for the Object Manager. The left sidebar lists the same configuration options as the previous screenshot. The main content area is titled "Details" for the "lease" object. It includes sections for Description, API Name (set to "lease\_\_c"), Singular Label (set to "lease"), Plural Label (set to "lease"), and several checkboxes for Reports, Activities, Field History, and Deployment Status, all of which are checked. The status is set to "Deployed". A "Help Settings" link and a "Standard salesforce.com Help Window" link are also present. At the bottom right are "Edit" and "Delete" buttons.

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-ed2820a810-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001PwMrn/Details/view>. The page title is "SETUP > OBJECT MANAGER" and the sub-page title is "Payment for tenantat". The left sidebar shows navigation options like 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, and Restriction Rules. The main content area displays the "Details" tab for the "Payment for tenantat" object. It includes fields for Description, API Name (Payment\_for\_tenantat\_c), Custom (✓), Singular Label (Payment for tenantat), Plural Label (Payment), and various checkboxes for Reports, Activities, Field History, and Deployment Status (Deployed). The status bar at the bottom shows the URL, a weather icon (Very humid), system icons, and the date/time (10:34 PM, 06-09-2025).

- Configured fields and relationships

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-ed2820a810-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgL000001PwBv/FieldsAndRelationships/view>. The page title is "SETUP > OBJECT MANAGER" and the sub-page title is "property". The left sidebar shows navigation options like 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, and Restriction Rules. The main content area displays the "Fields & Relationships" section for the "property" object. It lists 9 items, sorted by Field Label. The table shows columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data includes fields like Address, Created By, Last Modified By, Name, Owner, property, property Name, sfqt, and Type.

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		

Recently Viewed | Payment | S... Tenant | Salesforce

orgfarm-ed2820a810-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/011gL000001Pwfv/FieldsAndRelationships/view

Setup Home Object Manager

SETUP > OBJECT MANAGER Tenant

**Fields & Relationships**  
8 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		
property	property__c	Lookup(property)		✓
status	status__c	Picklist		
Tenant Name	Name	Text(80)		✓

Quick Find New Deleted Fields Field Dependencies Set History Tracking

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

Breaking news Why X fact-che...

Search web & PC 10:08 AM 07-09-2025 ENG IN

Recently Viewed | Payment | S... lease | Salesforce

orgfarm-ed2820a810-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/011gL000001PwQ1/FieldsAndRelationships/view

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		

Quick Find New Deleted Fields Field Dependencies Set History Tracking

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

Hot days ahead 28°C 10:13 AM 07-09-2025 ENG IN

**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)		✓
Tenant	Tenant_c	Lookup(Tenant)		✓

- Developed Lightning App with relevant tabs

**Lease Management - Lightning**

**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:  Primary Color Hex Value: #0070D2

Upload

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

**App Launcher Preview**

Lease Management

Application to efficiently handle the processes relate...

Recently Viewed | Payment | Lease Management - Lightning | Inbox (785) - janujanani240320 | Student

Lightning App Builder | App Settings | Pages | Lease Management

**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
- lease
- Tenants
- property

ENG - SA In 5 hours

Search web & PC

10:30 AM 07-09-2025

Recently Viewed | Payment | Lease Management - Lightning | Inbox (785) - janujanani240320 | Student

Lightning App Builder | App Settings | Pages | Lease Management

**App Settings**

App Details & Branding  
App Options  
Utility Items (Desktop Only)  
Navigation Items  
**User Profiles**

**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

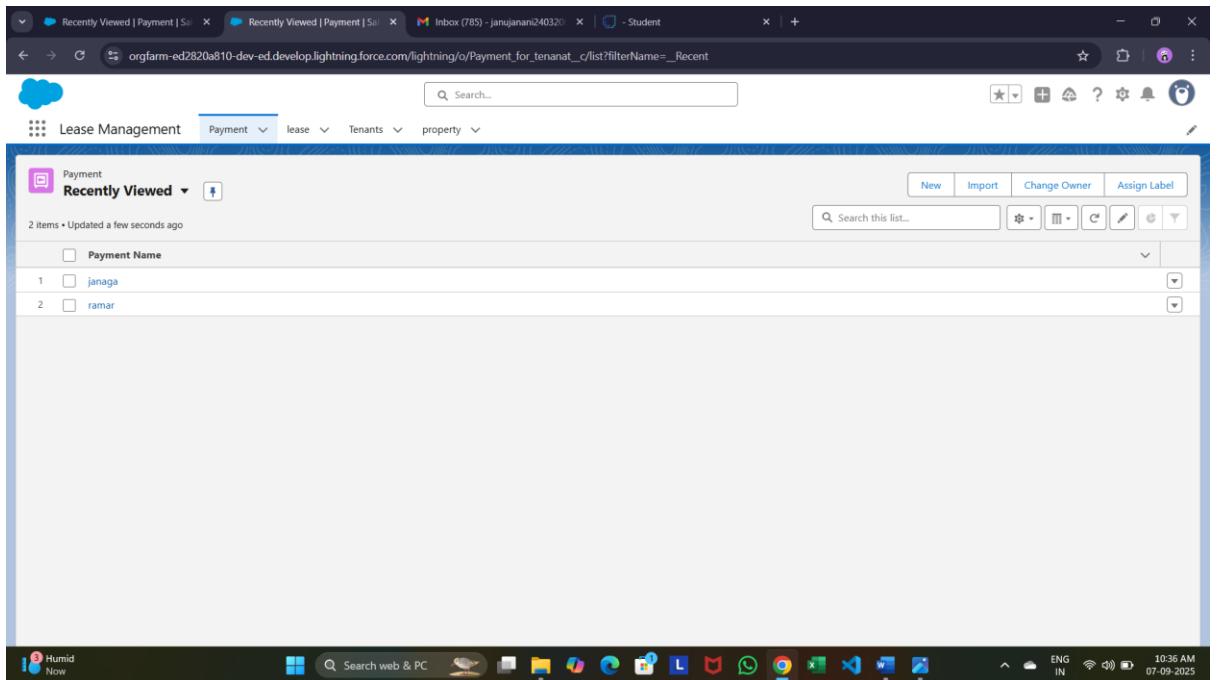
**Selected Profiles**

- System Administrator

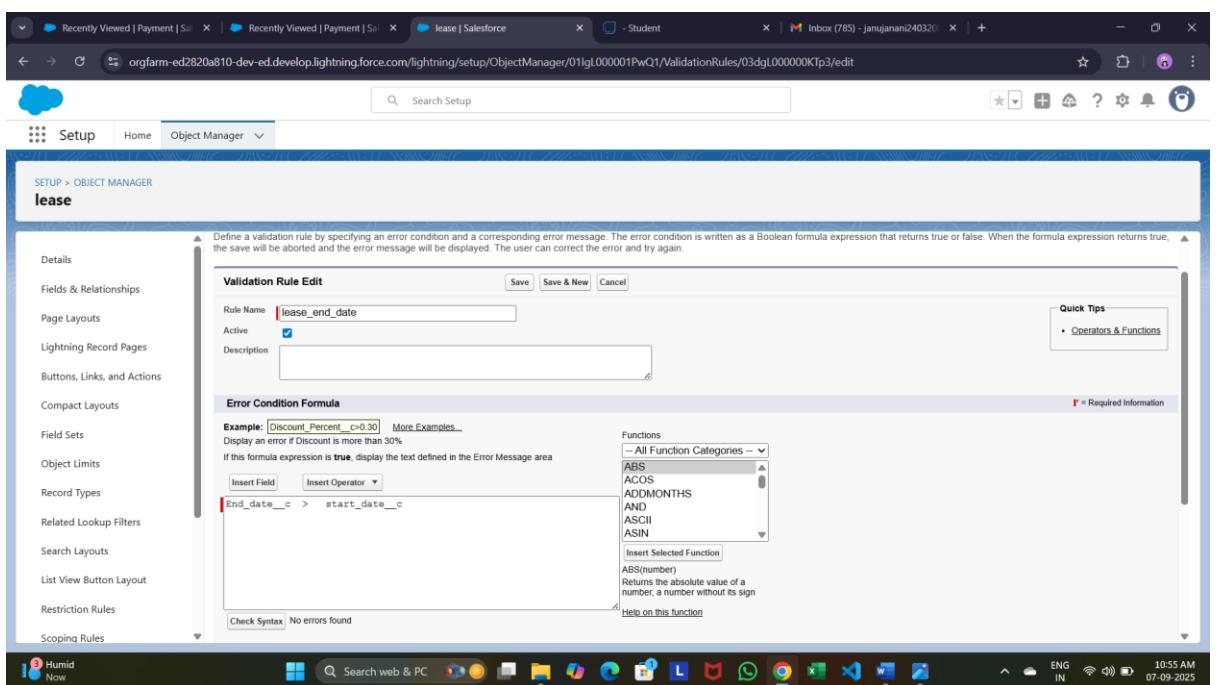
ENG - SA In 5 hours

Search web & PC

10:31 AM 07-09-2025



- To create validation rule to lease Object



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

- Page Header:** Recently Viewed | Payment | Sales | Recently Viewed | Payment | Sales | lease | Salesforce | - Student | Inbox (785) - janujanani240320
- Section:** SETUP > OBJECT MANAGER
- Object:** lease
- Section:** lease Validation Rule
- Validation Rule Detail:**
  - 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: janani\_A (8/12/2025, 8:13 AM)
  - Active: checked
  - Error Location: start date
  - Modified By: janani\_A (9/6/2025, 10:27 PM)
- Left Sidebar:** 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.
- Bottom:** Taskbar with various icons and system status.

## • Create Email Template

### Tenant leaving

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

- Page Header:** Recently Viewed | Payment | Sales | Recently Viewed | Payment | Sales | Classic Email Templates | Sales | - Student | Inbox (786) - janujanani240320
- Section:** SETUP > CLASSIC EMAIL TEMPLATES
- Search:** classic email
- Section:** Email > Classic Email Templates
- Email Template Edit:**
  - Email Template Information:**
    - Folder: Unfiled Public Classic Email Templates
    - Available For Use: checked
    - Email Template Name: tenant\_leaving
    - Template Unique Name: tenant\_leaving
    - Encoding: Unicode (UTF-8)
    - Description: (empty)
    - Subject: request for approve the leave
    - Email Body:

```
Dear {!Tenant__c.CreatedBy},  
Please approve my leave
```
  - Buttons:** Save, Save & New, Cancel
- Bottom:** Taskbar with various icons and system status.

## Leave approved

The screenshot shows the 'Classic Email Templates' page in the Salesforce setup. A new email template named 'Leave approved' is being edited. The template information includes:

- Email Template Information:**
  - Folder: Unfiled Public Classic Email Templates
  - Available For Use: Checked
  - Email Template Name: Leave approved
  - Template Unique Name: Leave\_approved
  - Encoding: Unicode (UTF-8)
  - Description: Leave approved
  - Subject: dear({Tenant\_\_c.Name})
  - Email Body:

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
- Available Merge Fields:** Contact Fields

## Leave rejected

The screenshot shows the 'Classic Email Templates' page in the Salesforce setup. A new email template named 'Leave rejected' is being edited. The template information includes:

- Email Template Information:**
  - Folder: Unfiled Public Classic Email Templates
  - Available For Use: Checked
  - Email Template Name: Leave rejected
  - Template Unique Name: Leave\_rejected
  - Encoding: Unicode (UTF-8)
  - Description: Leave rejected
  - Subject: Dear ({Tenant\_\_c.Name})
  - Email Body:

I hope this email finds you well. Your contract has not ended. So we can't approve your leave

your leave has rejected
- Available Merge Fields:** Contact Fields

## Tenant email

The screenshot shows the 'Classic Email Templates' page in the Salesforce setup. A new email template is being created with the following details:

- Email Template Information:**
  - Folder: Unfiled Public Classic Email Templates
  - Available For Use: Checked
  - Email Template Name: Tenant Email
  - Template Unique Name: Tenant\_Email
  - Encoding: Unicode (UTF-8)
  - Description: Urgent: Monthly Rent Payment Reminder
  - Email Body:

Dear {Tenant\_c\_Name},

I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding. As outlined in our rental agreement, the payment is due . To ensure the smooth operation of our property

## Tenant payment

The screenshot shows the 'Classic Email Templates' page in the Salesforce setup. A new email template is being created with the following details:

- Email Template Information:**
  - Folder: Unfiled Public Classic Email Templates
  - Available For Use: Checked
  - Email Template Name: tenant\_payment
  - Template Unique Name: tenant\_payment
  - Encoding: Unicode (UTF-8)
  - Description: Confirmation of Successful Monthly Payment
  - Email Body:

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.

- **Create Approval Process For check for vacant**

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The sidebar on the left has 'Approval Processes' selected under 'Process Automation'. The main content area includes a help section with 7 steps, a search bar for 'Manage Approval Processes For: Tenant', and a table for 'Active Approval Processes' with one row: 'check for vacant'. Below it is a section for 'Inactive Approval Processes' which says 'No approval processes available'. The status bar at the bottom indicates 'Hot days ahead 31°C' and the date '07-09-2025'.

## Initial Submission Action:

The screenshot shows the Salesforce Setup interface with the 'Email Alerts' page open. The sidebar on the left has 'Email Alerts' selected under 'Process Automation'. The main content area shows the creation of a new email alert with the subject 'please approve my leave'. It includes fields for 'Description' (please approve my leave), 'Unique Name' (please\_approve\_my\_leave), 'Object' (Tenant), and 'Email Template' (tenant\_leaving). The 'Recipients' section lists users: 'User: Integration User', 'User: OrgFarm EPIC', 'User: Security User', and 'User: janani A', with 'User: Integration User' selected. The status bar at the bottom indicates 'Trending videos A Big Bold Beau...' and the date '07-09-2025'.

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The left sidebar has sections like Data, Feature Settings, Process Automation, and Approval Processes, with 'Approval Processes' selected. The main area displays the 'Approval Processes' configuration screen, which includes tables for Approval Steps, Final Approval Actions, Final Rejection Actions, and Recall Actions. Each table lists actions (e.g., Record Lock, Email Alert), their types, descriptions, and assigned approvers. A search bar at the top and a message ' Didn't find what you're looking for? Try using Global Search.' are also visible.

## Apex Trigger

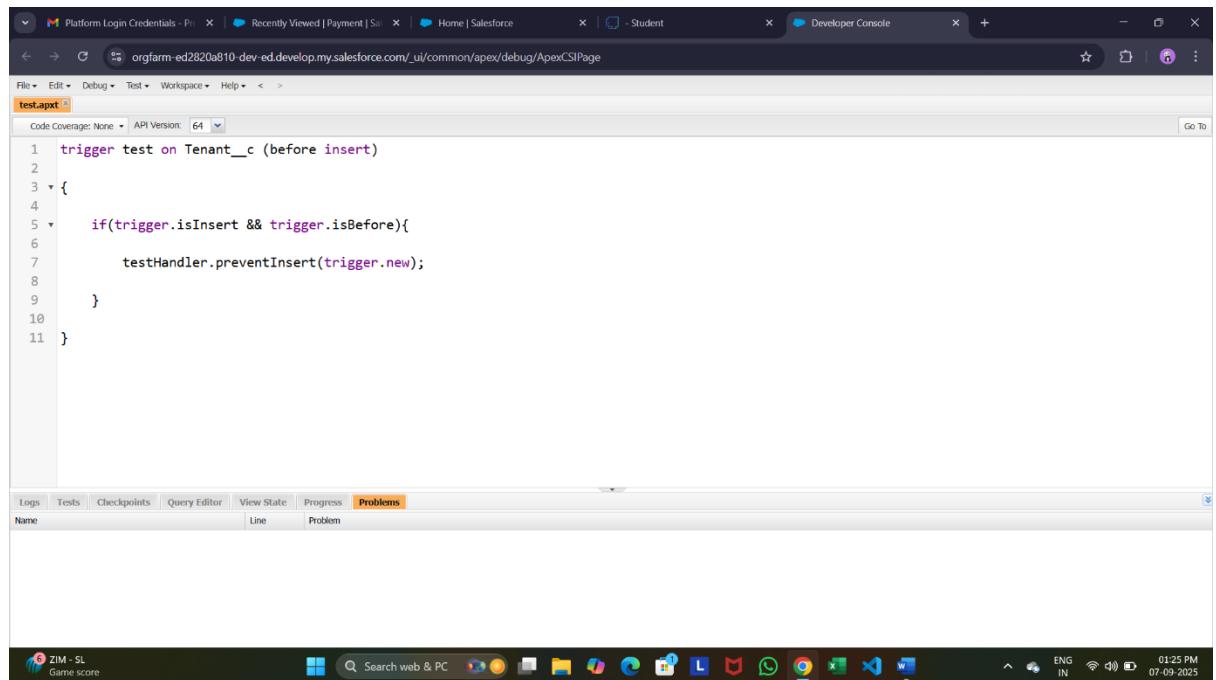
### Create an apex trigger:

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'test' on the Tenant\_\_c object. The trigger is defined as follows:

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

The developer console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected, showing an empty list. The bottom status bar indicates the device is connected to a network and shows the date and time.

## Create an Apex Handler class

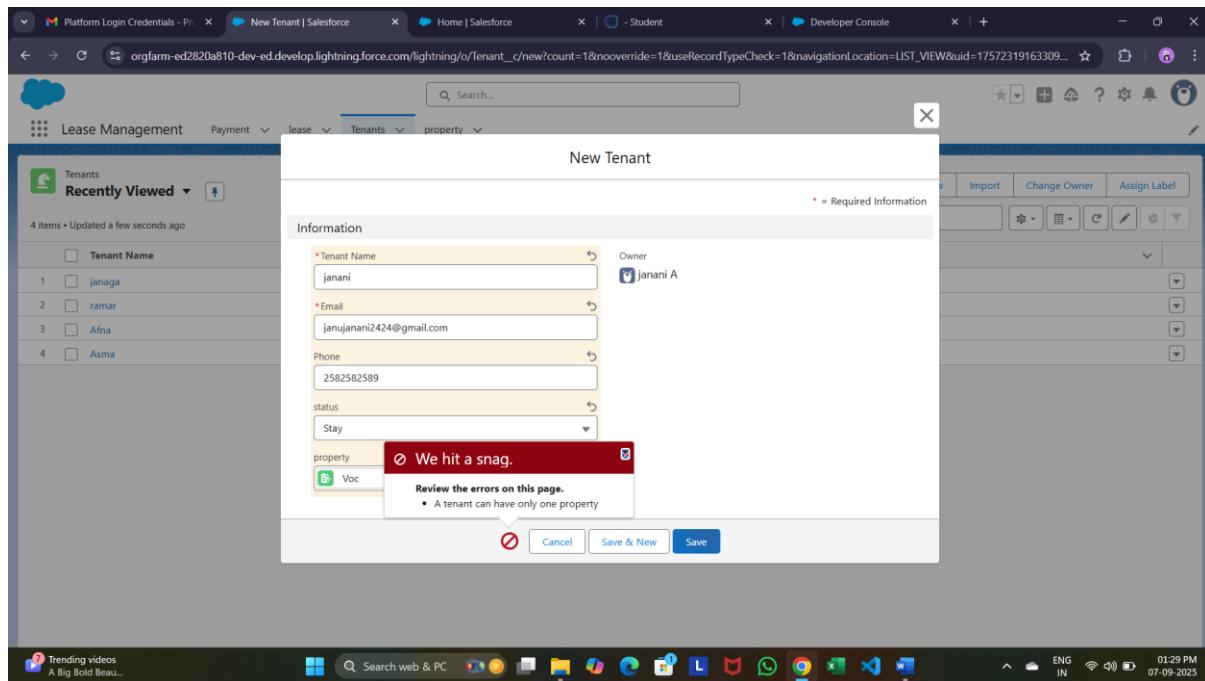


The screenshot shows the Salesforce Developer Console with an Apex trigger named `test.aprx`. The code is as follows:

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

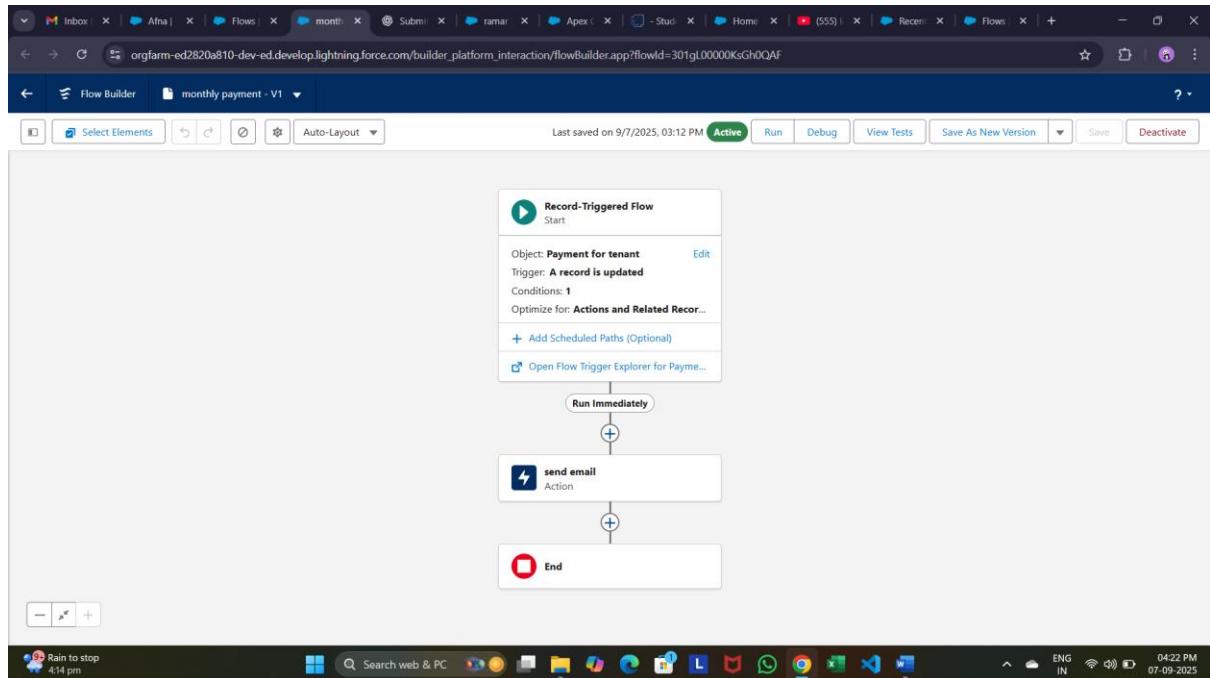
The console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is selected, showing no errors.

## Testing The Trigger:



The screenshot shows the Salesforce Lightning interface for the Lease Management application. A modal dialog titled "New Tenant" is open, displaying an error message: "We hit a snag. Review the errors on this page. • A tenant can have only one property". The "Owner" field is populated with "janani A". The "Tenant Name" field contains "janani". Other fields like "Email" and "Phone" are also visible. The background shows a list of existing tenants.

## Create Flow for monthly payment



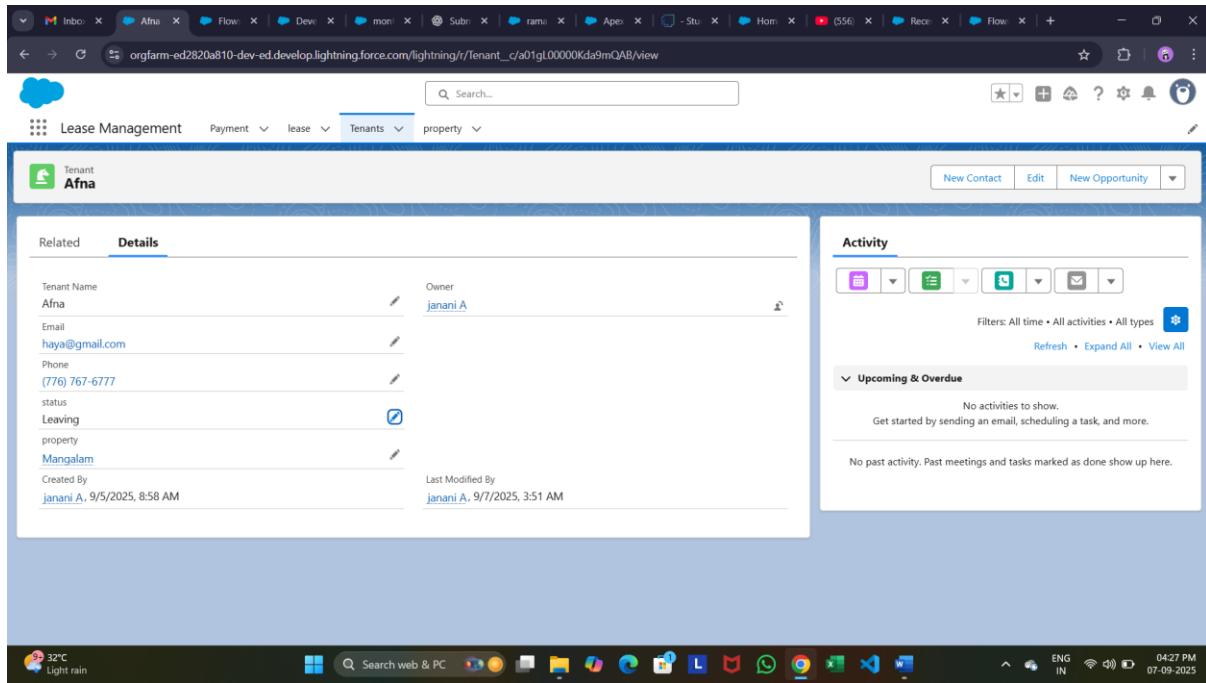
## Schedule class

### Create an Apex Class:

The screenshot shows the Salesforce Apex code editor. The class "MonthlyEmailScheduler" implements the "Schedulable" interface. It contains a single method "execute" which checks if the current day is 1, and if so, calls the static method "sendMonthlyEmails". The static method "sendMonthlyEmails" is also defined within the class. The code editor interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems, with the Problems tab currently selected. The desktop taskbar at the bottom shows various application icons and the date/time as 07-09-2025.

```
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 }
```

## Schedule Apex class



## 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.