

# Lease Management

College: 7155 - PSG Institute of Technology and Applied Research

Team ID: NM2024TMID00663

Team Lead: Harshavardhan K K (53399036FEF77E7BD274B6451DCEC1DA)

Team Members:

HARSHAVARDHAN K K	53399036FEF77E7BD274B6451DCEC1DA
KAVIYAVALLI A	F2916DDE85DA574914841EEB3EDF9374
MADHUMITHA S	54F0A2229B763EBBD834575A393DE970
SANTHOSHWAR R	FA976340C0BC8F9B5C046FDABB82A426
VIGNESHWARAN M	556695D2FF83B6AA02B42B7A10CA1986

## 1. Project Overview

This Lease Management project focuses on optimizing and automating the handling of lease agreements for real estate properties, equipment, or other assets. Built on Salesforce's platform, it aims to simplify lease management tasks, enhance communication, and maintain regulatory compliance. The system will support efficient tracking, streamlined communication, and accurate record-keeping for all lease transactions, improving both operational workflows and data reliability.

## 2. Objectives

### Business Goals:

- Develop a streamlined and automated lease management system that enhances tracking, compliance, and tenant relations.

- The project seeks to minimize administrative tasks, strengthen tenant communication, and ensure adherence to lease terms and regulatory standards.

### **Specific Outcomes:**

- A centralized dashboard to monitor lease terms, payments, and tenant information.
- Automated alerts and notifications for lease renewals, upcoming payments, and tenant communications.
- Streamlined approval workflows and validation mechanisms to maintain data accuracy and regulatory compliance.

## **3. Salesforce Key Features and Concepts Utilized**

This project leverages essential Salesforce features to build a powerful lease management system:

### **1. Custom Objects and Relationships**

- Created custom objects like Property, Tenant, Lease, and Payment to store crucial lease information.
- Defined lookup relationships to link lease records with tenants and properties, enabling efficient data access and record association.

### **2. Tabs and Lightning App (Lease Management)**

- Added tabs for each custom object to enhance navigation within the Lease Management Lightning app.
- Customized the app layout for streamlined functionality and consistent branding.

### **3. Validation Rules**

- Developed validation rules to uphold data integrity, ensuring accurate Lease entries and avoiding incomplete or erroneous submissions.

### **4. Email Templates**

- Created custom email templates for tenant communications, including:
  - Tenant Leaving Notification
  - Leave Approval/Rejection
  - Monthly Payment Reminder
  - Payment Confirmation

### **5. Approval Process**

- Configured an approval process for vacant properties:
  - **Initial Submission Action:** Sends for review upon submission.
  - **Final Approval Action:** Updates property status to leased and initiates tenant communication.
  - **Final Rejection Action:** Notifies tenant with rejection details.

## 6. Apex Triggers and Classes

- Developed an Apex Trigger for automating key lease tasks.
- Created an Apex handler class for functions like lease term validation and payment due date calculations.

## 7. Flows

- Designed a Monthly Payment Flow for streamlined tenant payment processing, with reminders and status tracking.

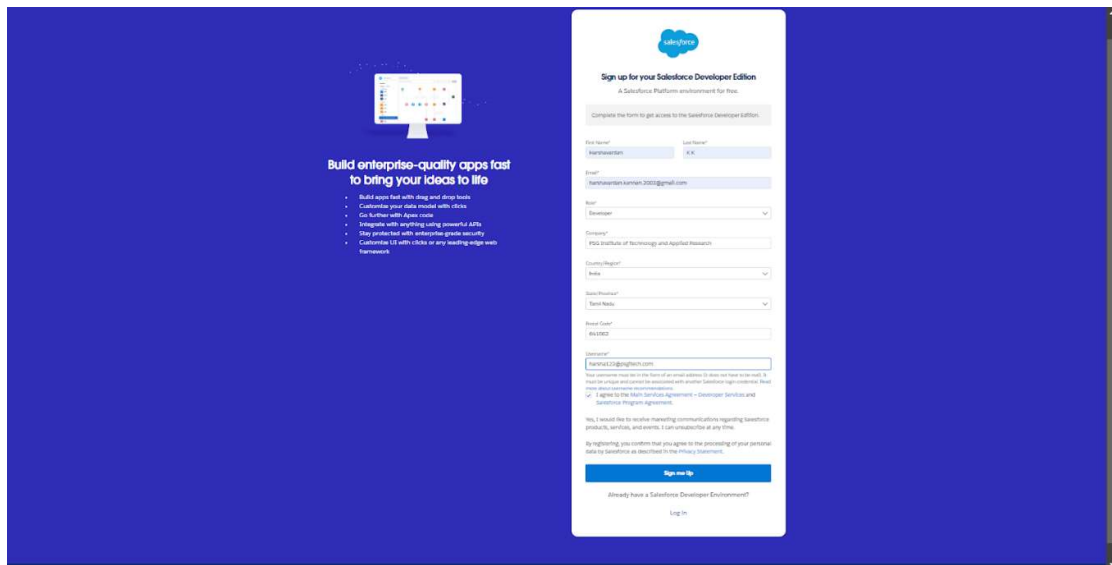
## 8. Scheduled Apex Class

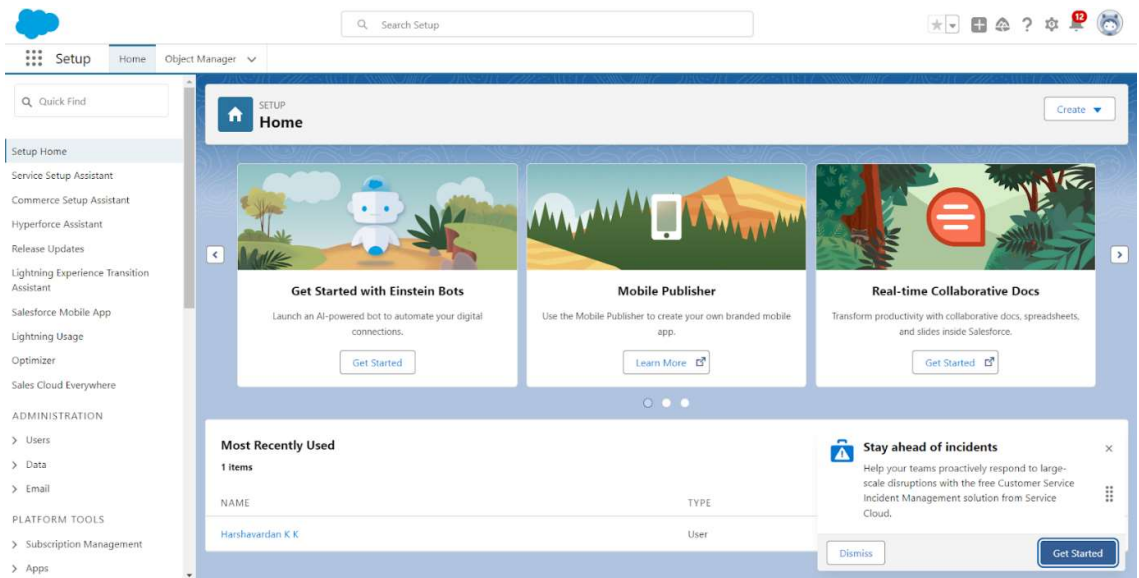
- Built an Apex class for scheduled tasks, automating monthly billing reminders and payment validations for timely notifications and compliance.

# 4. Detailed Steps to Solution Design

## 1. Created Salesforce Developer Account

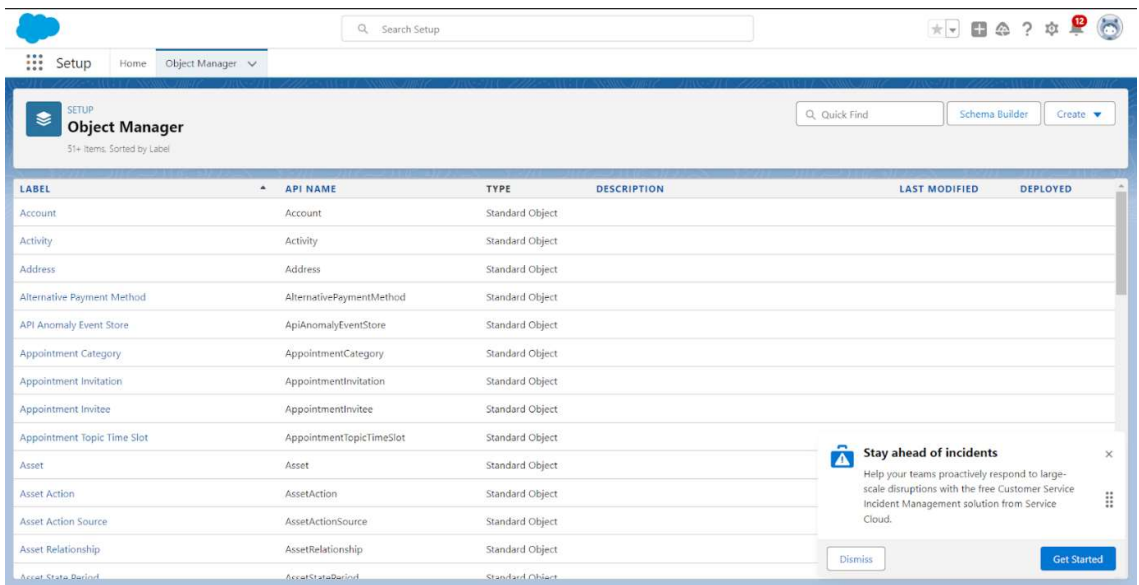
- We began by creating a Salesforce Developer Account to access a development environment where we could build and customize the application. This setup provided the necessary tools to design, test, and deploy the application.





## 2. Object Definitions

- Defined primary objects: **Property**, **Tenant**, **Lease**, and **Payment** within Object Manager.
- Each object was designed to store specific data: Property details, tenant information, lease agreements, and payment records.



Custom Object Definition Edit

Save Save & New Cancel

Custom Object Information

Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label  Example: Account

Plural Label  Example: Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name  Example: Account

Description

Context Sensitive Help Setting ☒ Open the standard Salesforce.com Help & Training window  
☐ Open a window using a Visualforce page

Content Name

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name  Example: Account Name

Data Type  Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

New Custom Object | Salesforce

psginstituteoftechnology-d7-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/new

Search Setup

Setup Home Object Manager

SETUP

New Custom Object

Custom Object Definition Edit

Save Save & New Cancel

Custom Object Information

Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label  Example: Account

Plural Label  Example: Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name  Example: Account

Description

Context Sensitive Help Setting ☒ Open the standard Salesforce.com Help & Training window  
☐ Open a window using a Visualforce page

Content Name

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name  Example: Account Name

Data Type  Warning: If you plan to insert a high volume of records in this object, via the API for example, use the Text data type.

Setup Home Object Manager

SETUP

New Custom Object

New Custom Object

Help for this Page

Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles. [Tell me more!](#) [Don't show this message again](#)

Custom Object Definition Edit

Save Save & New Cancel

Custom Object Information

Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label  Example: Account

Plural Label  Example: Accounts

Starts with vowel sound ☐

The Object Name is used when referencing the object via the API.

Object Name  Example: Account

Description

Context Sensitive Help Setting ☒ Open the standard Salesforce.com Help & Training window  
☐ Open a window using a Visualforce page

Content Name

Enter Record Name Label and Format

### 3. Tab Configuration

- Configured tabs for each main object to facilitate quick access.
- Enabled tabs for Property, Tenant, Lease, and Payment, allowing users to manage lease data seamlessly.

The screenshot shows the 'Edit Custom Object Tab' page for the 'property' object. The page title is 'property'. Below the title, it says 'Fill in the fields below to define the custom tab.' The main section is 'Custom Tab Definition Edit'. Under 'Custom Object Tab Information', there are fields for 'Tab Label' (property), 'Object' (property), and 'Tab Style' (Airplane). There is a note: '(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.' Below this is a dropdown menu for 'Splash Page Custom Link' with the value '--None--'. There is also a 'Description' field. At the bottom, there are 'Save' and 'Cancel' buttons.

The screenshot shows the 'Custom Tabs' page in Salesforce Setup. The page title is 'Custom Tabs'. Below the title, it says 'You can create new custom tabs to extend Salesforce functionality or to build new application functionality.' The page is divided into four sections: 'Custom Object Tabs', 'Web Tabs', 'Visualforce Tabs', and 'Lightning Component Tabs'. Each section has a 'New' button and a 'What is This?' link. The 'Custom Object Tabs' section shows a table with columns 'Action', 'Label', 'Tab Style', and 'Description'. The table contains two rows: one for 'Lease' with a 'Alarm clock' icon and one for 'Leasehold' with an 'Airplane' icon. The other three sections (Web Tabs, Visualforce Tabs, and Lightning Component Tabs) all show 'No [type] Tabs have been defined'.

### 4. Lightning App Development

- Created a custom Lightning App named **Lease Management** to consolidate all functionalities.
- Configured branding, navigation items, and access settings to improve user experience and accessibility.

**New Lightning App**

**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 <sup>?</sup>

\* Developer Name <sup>?</sup>

Description <sup>?</sup>

**App Branding**

Image <sup>?</sup>

Primary Color Hex Value <sup>?</sup>

Org Theme Options

☐ Use the app's image and color instead of the org's custom theme

**Next**

**New Lightning App**

**User Profiles**

Choose the user profiles that can access this app.

**Available Profiles**

- Salesforce API Only System Integrations

**Selected Profiles**

- System Administrator

**Back** **Save & Finish**

## 5. Custom Field Creation

- Created custom fields within each object to capture required details:
  - Property:** Location, type, rental amount, availability status.

Setup Home Object Manager

SETUP > OBJECT MANAGER

### property

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

property

#### New Custom Field

Step 2. Enter the details Step 2 of 4

Field Label:

Values:

- ☐ Use global picklist value set
- ☒ Enter values, with each value separated by a new line

☐ Display values alphabetically, not in the order entered

☐ Use first value as default value

☒ Restrict picklist to the values defined in the value set

Field Name:

Description:

Previous Next Cancel

- **Tenant:** Contact information, lease start/end dates.

Setup Home Object Manager

SETUP > OBJECT MANAGER

### Tenant

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

7 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

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)		✓

- **Lease:** Lease terms, conditions, and expiration date.

Setup Home Object Manager

SETUP > OBJECT MANAGER

### lease

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

6 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

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)		✓
start date	start_date__c	Date		



- **Payment:** Amount due, due date, and payment status.

SETUP > OBJECT MANAGER

**Payment for tenant**

Details

**Fields & Relationships**  
7 Items, Sorted by Field Label

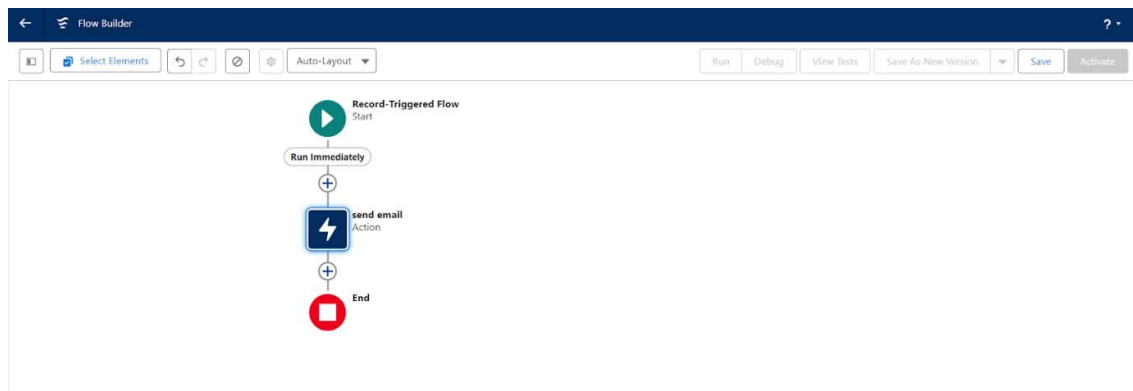
Q Quick Find New Deleted Fields Field Dependencies Set History Tracking

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)		✓

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

## 6. Flow Creation for Monthly Payments

- Developed a **Monthly Payment Flow** to guide users through the payment process for tenants.
- Included automated reminders and status tracking, streamlining monthly rent collection.



## 7. Apex Trigger and Handler Class

- Created an Apex Trigger to automate lease calculations, such as due dates and reminders.
- Developed an Apex handler class to process complex calculations and manage lease status changes.



SetupHomeObject Manager

Search Setup

12

email te

Classic Email TemplatesLightning Email Templates

Didn't find what you're looking for? Try using Global Search.

SETUP

Classic Email Templates

Text Email Template

tenant leaving

Help for this Page

Preview your email template below.

Email Template Detail

EditDeleteClone

Email Templates from Salesforce

Email Template Name

Unified Public Classic Email Templates

tenant leaving

Available For Use

✓

Template Unique Name

tenant\_leaving

Last Used Date

Encoding

Unicode (UTF-8)

Times Used

Author

Harshavardan K.K. (Change)

Description

Created By

Harshavardan K.K. 17/10/2024, 5:06 pm

Modified By

Harshavardan K.K. 17/10/2024, 5:06 pm

EditDeleteClone

Email Template

Send Text and Verify Merge Fields

Subject

request for approve the leave

Plain Text Preview

Dear (Tenant\_\_c.CreatedBy),

Please approve my leave

SetupHomeObject Manager

Search Setup

12

email te

Classic Email TemplatesLightning Email Templates

Didn't find what you're looking for? Try using Global Search.

SETUP

Classic Email Templates

Step 2. Text Email Template: New Template

Step 2 of 2

PreviousSaveCancel

Email Template Information

Required Information

Folder

Unified Public Classic Email Templates

Available For Use

✓

Email Template Name

Leave approved

Template Unique Name

Leave\_approved

Encoding

Unicode (UTF-8)

Description

Subject

Leave approved

Email Body

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

PreviousSaveCancel

SetupHomeObject Manager

Search Setup

12

email te

Classic Email TemplatesLightning Email Templates

Didn't find what you're looking for? Try using Global Search.

SETUP

Classic Email Templates

Text Email Template

Leave approved

Help for this Page

Preview your email template below.

Email Template Detail

EditDeleteClone

Email Templates from Salesforce

Email Template Name

Unified Public Classic Email Templates

Leave approved

Available For Use

✓

Template Unique Name

Leave\_approved

Last Used Date

Encoding

Unicode (UTF-8)

Times Used

Author

Harshavardan K.K. (Change)

Description

Created By

Harshavardan K.K. 17/10/2024, 5:08 pm

Modified By

Harshavardan K.K. 17/10/2024, 5:08 pm

EditDeleteClone

Email Template

Send Text and Verify Merge Fields

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

## 9. Approval Process for Vacant Properties

- Established an approval process to verify property availability before leasing:
  - Configured initial submission, final approval, and rejection actions to streamline the lease request process.

The screenshot shows the Salesforce Setup interface for creating a new approval process. The left sidebar contains a search bar with "approval process" and a list of navigation items including "Approval Processes". The main content area is titled "Approval Processes" and shows a "New Approval Process" form for "Tenants". The form is at "Step 1: Enter Name and Description". It includes fields for "Process Name" (filled with "check for vacant"), "Unique Name" (filled with "check\_for\_vacant"), and "Description". A "Next" button is visible at the bottom right of the form.

## 10. Scheduled Apex Class for Monthly Reminders

- Scheduled an Apex class to handle monthly billing reminders and lease renewals, ensuring tenants and administrators are up-to-date on upcoming obligations.

The screenshot shows the Salesforce Setup interface for scheduling an Apex class. The left sidebar contains a search bar and a list of navigation items including "Apex Classes". The main content area is titled "Apex Classes" and shows a "Schedule Apex" form. The form includes fields for "Job Name" (filled with "MonthlyEmailScheduler") and "Apex Class" (filled with "MonthlyEmailScheduler"). The "Schedule Using" section has "Schedule Builder" selected. The "Schedule Apex Execution" section shows "Frequency" set to "Monthly", "Start" date as "17/10/2024", "End" date as "17/11/2024", and "Preferred Start Time" as "7:00 pm". A "Save" button is visible at the bottom right of the form.

Lease Management

Tenants

property

Payment

lease

New Tenant

Information

\* Tenant Name

Harshavardan

\* Email

harshavardan.kannan.2003@gmail.com

Phone

Status

Stay

Owner

Harshavardan K K

Cancel Save & New Save

## 11. Create a Tenant Record

- Created a **Tenant** record within the system, entering all relevant details such as contact information, lease start and end dates, and tenant status.
- Ensured that the new tenant record is associated with the appropriate **Property** and **Lease** records through lookup relationships, enabling seamless data access and association.

Lease Management

Tenants

property

Payment

New Tenant

Information

\* Tenant Name

Srinivas

\* Email

sripaddep204@gmail.com

Phone

Status

Stay

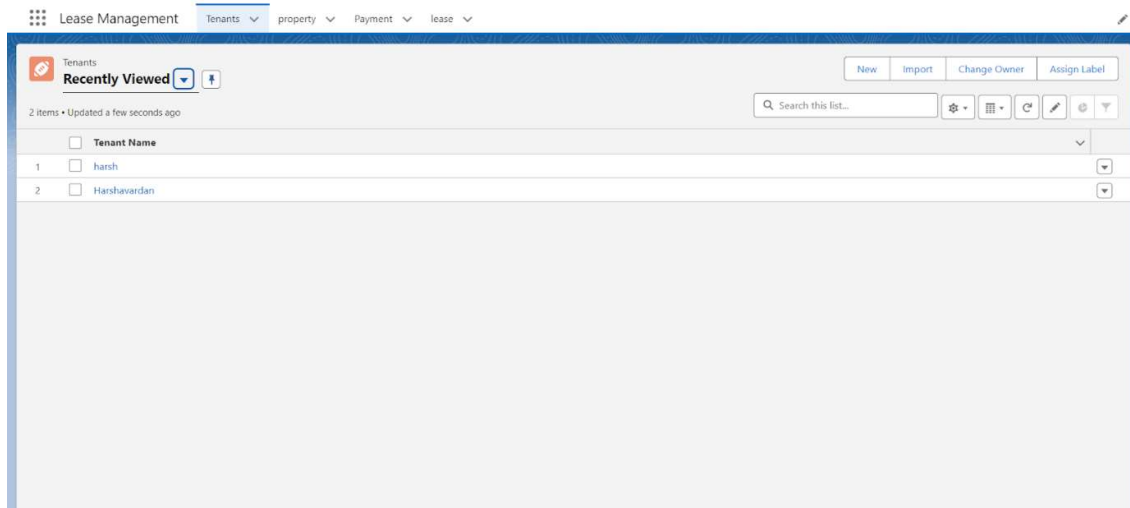
Owner

Srinivasa Pradeep S

Cancel Save & New Save

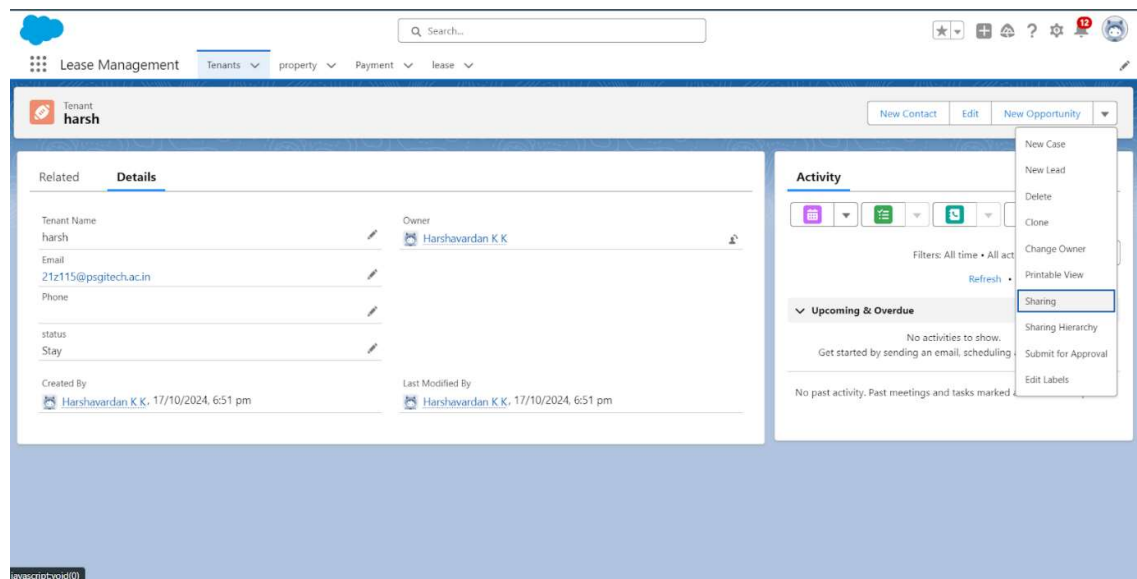
## 12. Submit for Approval

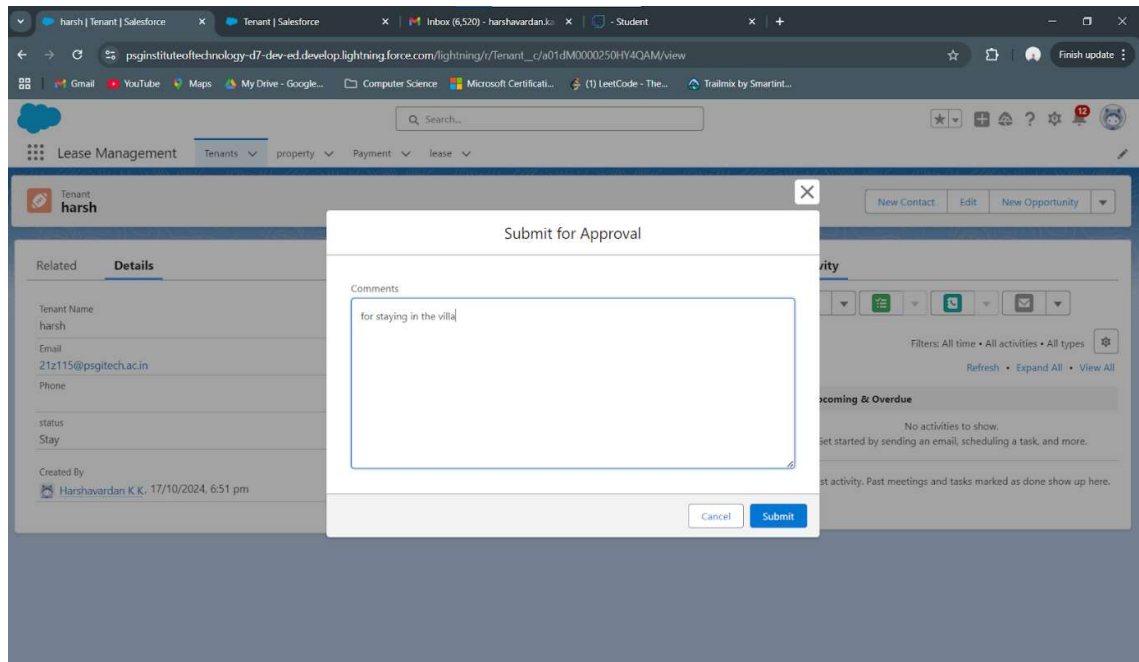
- Initiated the **Approval Process** for the new lease by submitting the **Lease** record for approval.
- The submission triggers an initial review, allowing the leasing team to validate all provided information before final approval.



### 13. Wait for Approval

- The **Approval Process** waits for the approver's action (approval or rejection). During this phase, the lease status remains in a "Pending Approval" state until final action is taken.



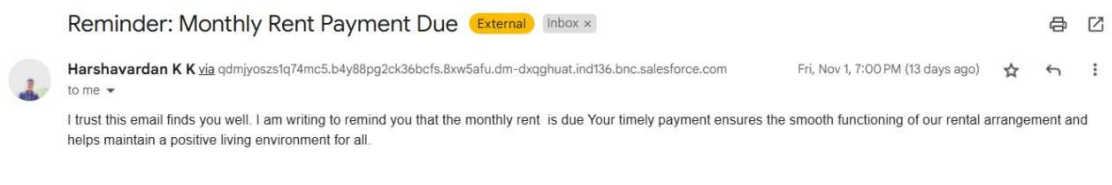


#### 14. Email Sent for Approval Confirmation

- Upon approval, an automated **Email Template** for lease approval is triggered and sent to the tenant, confirming the lease's approval status.



- The email outlines the lease's terms and conditions, ensuring clear communication with the tenant.



## 5. Testing and Validation

### Unit Testing (Apex Classes, Triggers)

#### Apex Trigger:

trigger test on Tenant\_\_c (before insert)

```
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}
```

#### Test Class:

```
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');
            }
        }
    }
}
```

#### MonthlyEmailScheduler:

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});

    }

}

```

## Steps:

1. Created a Test Property Record: Created a Property\_\_c record to serve as the property for assigning tenants.
2. Inserted the First Tenant: Inserted a Tenant\_\_c record linked to the test property, simulating the initial tenant assignment.
3. Prepared a Second Tenant for the Same Property: Created another Tenant\_\_c record with the same property ID to test the trigger's restriction on multiple tenants per property.
4. Inserted the Second Tenant: Attempted to insert the second tenant within a try-catch block to catch the expected error, confirming that the trigger prevented the insertion.

5. Verified Error Message: Checked if the error message matched the expected output: "A tenant can have only one property," verifying that the trigger works as intended.

6. Run Apex Test Execution:

- Go to Setup > Apex Test Execution.
- Select TenantTriggerTest and click Run.

Confirm that all assertions pass, indicating that the trigger prevents multiple tenants from being assigned to the same property.

The screenshot displays the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The left sidebar shows a 'Quick Find' search bar and a list of setup categories: Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Lightning Usage, Optimizer, Sales Cloud Everywhere, ADMINISTRATION (Users, Data, Email), PLATFORM TOOLS (Subscription Management, Apps), and a search bar. The main content area is titled 'Apex Triggers' and shows the 'Apex Trigger Edit' page for a trigger named 'test'. The trigger is active and its code is displayed in a text area. Below the trigger code, the 'Apex Test Execution' page is visible, showing a list of test results. The table below contains the test execution results.

User	Application	Operation	Time	Status	Read	Size
Harshvardan K K	Browser	/setup/build/testApexClass.apex	10/17/2024, 6:22:14 PM	Success	Unread	798 bytes
Harshvardan K K	Browser	/setup/build/testApexClass.apex	10/17/2024, 6:22:10 PM	Success	Unread	1.48 KB
Harshvardan K K	Browser	/setup/build/viewApexTrigger.apex	10/17/2024, 6:21:40 PM	Success	Unread	1.03 KB
Harshvardan K K	Browser	/setup/build/allTriggers.apex	10/17/2024, 6:21:32 PM	Success	Unread	1.74 KB
Harshvardan K K	Browser	/setup/build/viewApexTrigger.apex	10/17/2024, 6:21:26 PM	Success	Unread	1.03 KB
Harshvardan K K	Browser	/setup/build/editApexClass.apex	10/17/2024, 6:20:53 PM	Success	Unread	359 bytes

## 6. Conclusion

**Summary of Achievements:** The Lease Management project successfully developed a streamlined, transparent, and scalable solution for managing property leases. Key achievements include:

- **Organized Data Management:** Custom objects, fields, and tabs were created for structured and efficient lease tracking.
- **Process Automation:** Triggers, validation rules, and scheduled classes were implemented to streamline recurring tasks, including monthly payments and notifications.
- **Enhanced Communication:** Approval workflows and email templates were designed to automate tenant communications, improving clarity and tenant satisfaction.
- **Data-Driven Decision-Making:** Custom reports and dashboards provide stakeholders with real-time insights, enabling faster and more informed decision-making.

This project showcases the powerful application of Salesforce in lease management, driving operational efficiency, ensuring data accuracy, and enhancing tenant relations through optimized workflows.