

# **SHIVANI ENGINEERING COLLEGE**

Department of Computer science and engineering

**PROJECT TITLE:**

## **LEASE MANAGEMENT SYSTEM**

**Submitted by (Team Members):**

Team ID: NM2025TMID04708

Team Size: 4

Team Leader: Atchaya S

Team member: Nandhini S

Team member: Kaviya C

Team member: Vincy J

Guide by: Archana N

**Platform:** Salesforce Developer

**Date of Submission:** 04/11/2025

## **bProject Description**

A lease management project involves creating a system or application to efficiently handle the processes related to leasing real estate properties, equipment, or other assets.

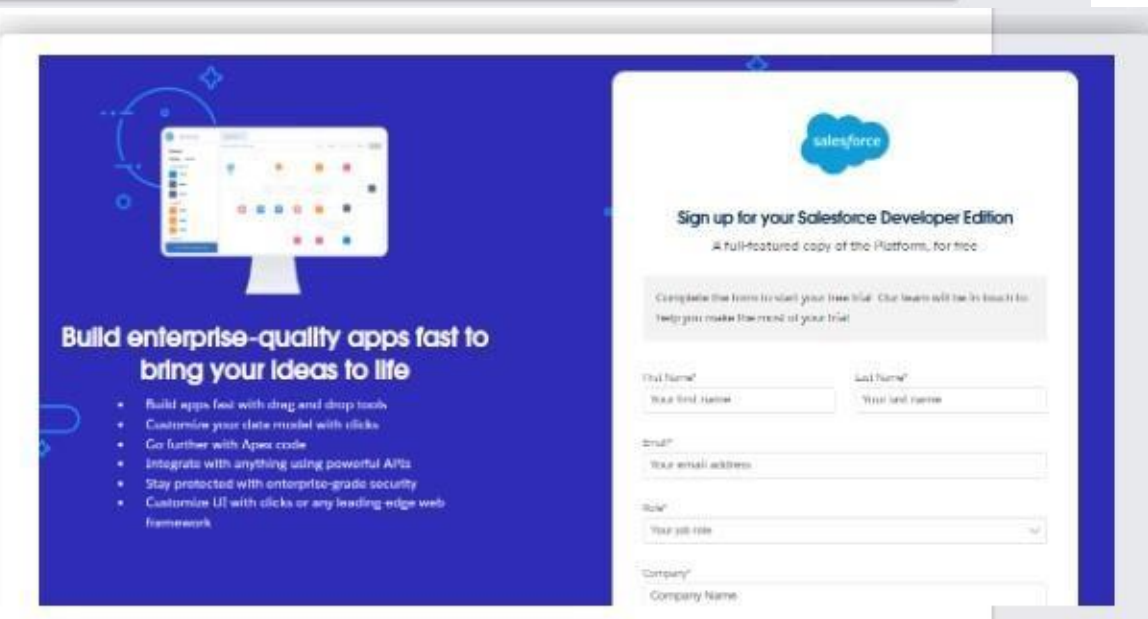
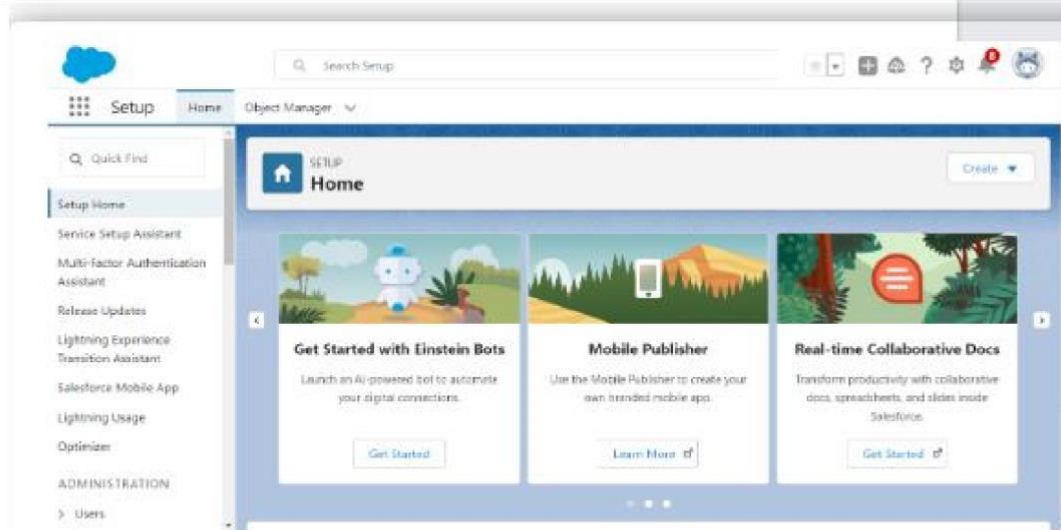
The goal is to streamline and automate various tasks associated with lease agreements, ensuring accurate record-keeping, compliance with regulations, and effective communication between parties involved.

## **Creating Developer Account**

In this step, a new Salesforce Developer Account is created using a valid email address. This account provides access to the development environment for project setup.

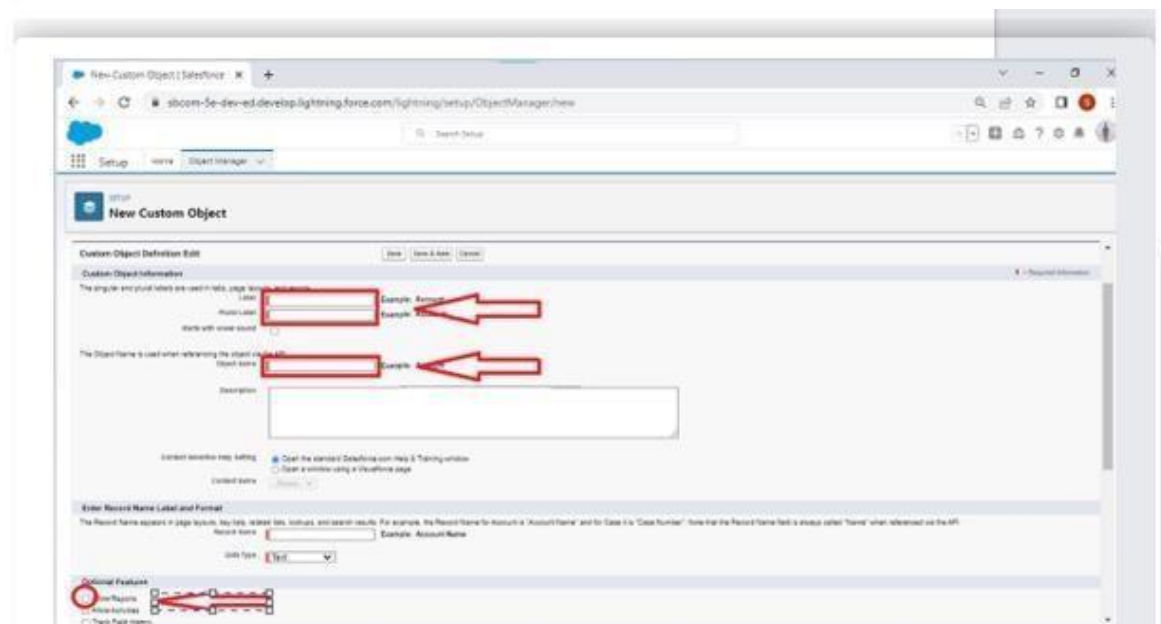
## **Account Activation**

Activate the Salesforce Developer Account by verifying the registered email and completing the activation process.



## Create Property Object

Create a new custom object named “Property” to store details related to lease properties such as property name, type, and location.



## Create Tenant Object

Create a “Tenant” custom object to manage tenant information including name, contact, and lease duration.

## Create Payment Object

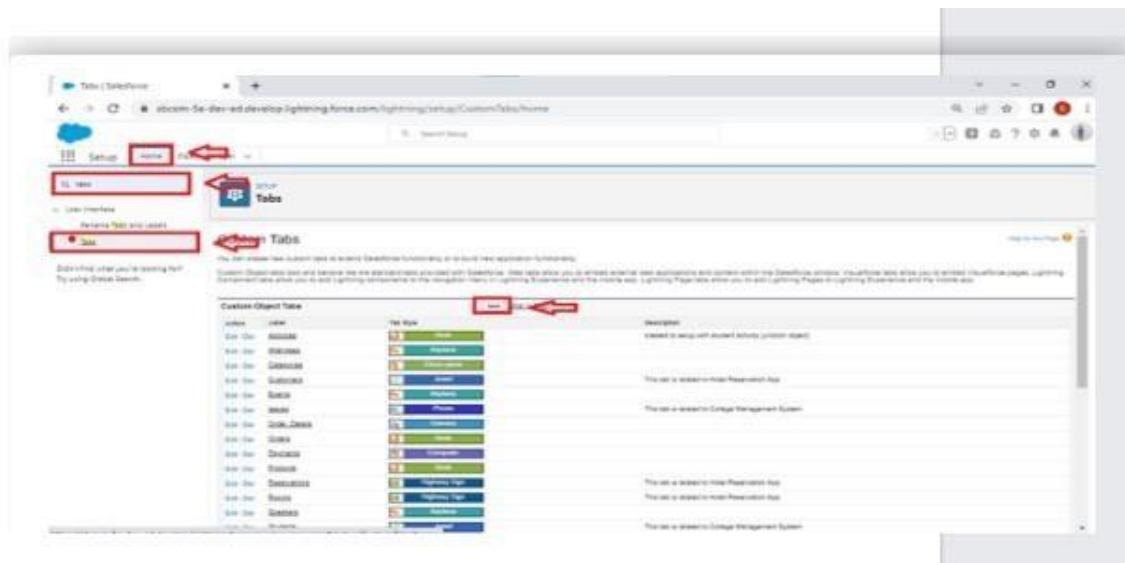
Design a “Payment” object to handle rent payments, due dates, and payment status tracking.

## Create Lease Object

Establish a “Lease” object to connect Property and Tenant objects and manage lease terms, start and end dates.

## Creating a Custom Tab

Add custom tabs for each created object to make them accessible in the Salesforce app navigation bar.



## Lightning App Creation

Create a custom Lightning App that includes all the custom objects and tabs for better project navigation. **Fields Creation**



## **Email Template**

An **Email Template** in Salesforce is a pre-designed message format that allows users to send standardized emails automatically or manually. It saves time, ensures consistency, and helps communicate important information like **lease confirmation**, **payment receipts**, and **reminders** to tenants

### **1. Tenant Leaving Notification**

Sends an alert to the admin when a tenant requests to vacate the property.

### **2. Lease Approved**

Notifies the tenant that their lease request has been approved successfully.

### **3. Lease Rejection Email**

Informs the tenant that their lease application has been rejected due to missing details or eligibility.

### **4 . Monthly Payment Reminder**

Automatically reminds the tenant each month about the upcoming rent payment due date.

### **5. Successful Payment Confirmation**

Sends a thank-you message and confirmation once the tenant's rent payment is received successfully.

## **Approval Process**

An **Approval Process** automates how records are approved in Salesforce. In this project, an approval process is created for the **Lease or Payment** object to ensure that certain records (for example, rent amount > ₹50,000) need admin or manager approval before final confirmation.

The screenshot shows the 'Approval Processes' setup page in Salesforce. The process is named 'check for vacant'. It is at 'Step 3 of 4: Specify Approver Field and Record Editability Properties'. The 'Next Automated Approver Determined By' dropdown is set to 'None'. Under 'Record Editability Properties', the option 'Administrators ONLY can edit records during the approval process' is selected. Navigation buttons 'Previous', 'Save', 'Next', and 'Cancel' are visible at the bottom right.

## Apex Trigger

An **Apex Trigger** is used to perform automatic actions when a record is created or updated. In this project, a trigger is written on the **Payment** object to automatically update the **Lease Status** once the payment is completed.

The screenshot shows the Salesforce IDE with an Apex trigger named 'test' on the 'Tenant\_\_c' object, set to trigger 'before insert'. The code is as follows:

```

1 trigger test on Tenant__c (before insert)
2 {
3     if(trigger.isInsert && trigger.isBefore){
4         testHandler.preventInsert(trigger.new);
5     }
6 }

```

## Apex logic:

```

:
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) {

```

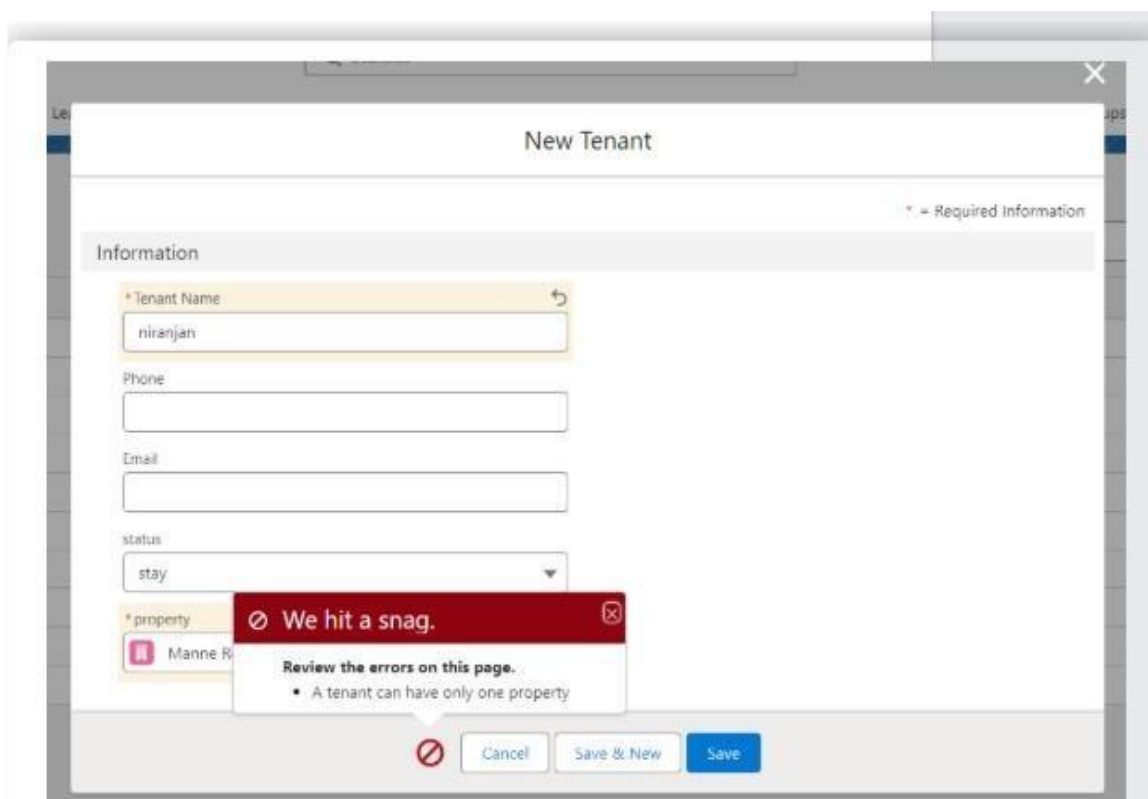


```

        if (newTenant.Property__c != null &&
existingPropertyIds.contains(newTenant.Property__c)) {
            newTenant.addError('A tenant can have only one property');
        }
    }
}
}
}

```

## Testing the Trigger:



## Flow Creation

A **Flow** is created to automate record updates or send email alerts without coding.

In this project, a **Record-Triggered Flow** is designed on the **Tenant** or **Payment** object to send an automatic email when a new payment is recorded.

Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.

For example — sending payment reminders, lease expiry alerts, or autoupdating records every morning.

### Schedule Apex

Schedule an Apex class that implements the Schedulable interface to be automatically executed on a weekly or monthly interval.

Job Name:

Apex Class:

Schedule Apex Location

Frequency: ☐ Weekly ☒ Monthly

On day:  of every month ☐ On:  of every month

Start:

End:

Preferred Start Time:

Exact start time will depend on job queue activity.

### Apex Classes

Apex Class is an object model programming language that allows developers to develop custom business applications on the Lightning Platform.

**Percent of Apex Code DTS**  
For all Lightning, using a mix of Apex Class (including interfaces) and Apex Test (including interfaces) in your organization. % of all classes with DTS DCCO (includes Apex Test classes and Apex Test classes) in your organization.

Customize your search results: public access only

Cancel All Search

PM:

**1** **2** **3** **4** **5** **6** **7** **8** **9** **10** **11** **12** **13** **14** **15** **16** **17** **18** **19** **20** **21** **22** **23** **24** **25** **26** **27** **28** **29** **30** **31** **Next** **Previous**

Author	Name	Namespace	Apex Profile	Status	Has Without Comments	Last Modified By	Last Modified Date
John Doe	MonthlyEmailScheduler	MonthlyEmailScheduler	001	Active	0/0	John Doe	04/12/2023 10:00 AM
John Doe	MonthlyEmailScheduler	MonthlyEmailScheduler	001	Active	0/0	John Doe	04/12/2023 10:00 AM
John Doe	MonthlyEmailScheduler	MonthlyEmailScheduler	001	Active	0/0	John Doe	04/12/2023 10:00 AM
John Doe	MonthlyEmailScheduler	MonthlyEmailScheduler	001	Active	0/0	John Doe	04/12/2023 10:00 AM

## Conclusion

The *Lease Management System* project was successfully created using Salesforce. This project demonstrates how different Salesforce features such as **Objects, Tabs, Validation Rules, Email Templates, Approval Process, Flows, and Apex Triggers** can be used to automate and simplify lease management tasks. The system ensures efficient handling of tenant records, payments, and lease approvals with minimal manual work. Overall, this project improves productivity, reduces human error, and provides a digital solution for real-time property management.

