

# Lease Management System

**JAVVADI KARTHIK**  
**(22PA1A0561)**

**College : VISHNU INSTITUTE OF TECHNOLOGY**

**Branch: CSE**

Date	11 JUNE 2025
Team ID	LTVIP2025TMID30578
Project Name	Lease Management
Maximum Marks	4 Marks

## **Project Report Format**

1. **INTRODUCTION**
  - 1.1 Project Overview
  - 1.2 Purpose
2. **IDEATION PHASE**
  - 2.1 Problem Statement
  - 2.2 Empathy Map Canvas
  - 2.3 Brainstorming
3. **REQUIREMENT ANALYSIS**
  - 3.1 Customer Journey map
  - 3.2 Solution Requirement
  - 3.3 Data Flow Diagram
  - 3.4 Technology Stack
4. **PROJECT DESIGN**
  - 4.1 Problem Solution Fit
  - 4.2 Proposed Solution

#### 4.3 Solution Architecture

### 5. PROJECT PLANNING & SCHEDULING

#### 5.1 Project Planning

### 6. FUNCTIONAL AND PERFORMANCE TESTING

#### 6.1 Performance Testing

### 7. RESULTS

#### 7.1 Output Screenshots

### 8. ADVANTAGES & DISADVANTAGES

### 9. CONCLUSION

### 10. FUTURE SCOPE 11. APPENDIX

Source Code(if any)

Dataset Link

GitHub & Project Demo Link

## Abstract:

*This project is based on Lease MANAGEMENT SYSTEM, which, in a strategic alliance with Salesforce.com, develops an application for managing properties. LMS is an application that incorporates all the functions necessary to manage different processes of the real estate sector with the best practices in mind. It is delivered as a service (SaaS) and accessed via Internet from any device. Salesforce.com ([www.salesforce.com](http://www.salesforce.com)) provides the technological platform that supports the data storage, the security elements and protocols and the information backup in redundant servers in the EEUU, Europe and Asia, and a series of transversal functions such as messaging, information intelligence system, user profiles, among many others. It manages the relationship with clients and owner, it maintains Management processes, Buildings and units' information, areas information, and various types of information and update about it. This study aimed to design and develop a cloud-based Lease management system that provides detailed and summarized information on these critical areas to guide LMS administrators in planning and decision-making. It is accessible anywhere anytime as data is stored remotely and made available to users over the internet. This project is user-friendly, usable, performance, and security; and good in terms of robustness. This implies that other owners may*

*adopt this system for more efficient information management and more effective management decisions and to maintain good relationships with Clients LMS allows the management of real estate businesses.*

**Keywords:** LMS, lease management, Rental Property, Property Management

## **I. INTRODUCTION**

This Lease Management System project developed using salesforce(a cloud-based system). The main Objective/ Aim of this system, is to reduce the Consumption of Time, reducing the Redundancy etc. during maintaining the records of Hostel management. Separate divisions are provided to maintain the records of Building, Unit, Clients Details. Admin is the Superuser of this project. The proposed software will also reduce the clumsy paperwork, manual labour as well as communication cost.

Salesforce is a customer relationship management system widely used in business management but here we are using it for college management. Maintaining the relationship between college, students and faculties is the main moto of this project.

In the world of technology computers affect our lives in further ways than we probably are Apprehensive of. We've Computerized Management Systems maintaining the information of Lease and this LMS project shows all the working that's espoused by an Institution. It's delicate to prepare the homemade work to store the information about a Clients and Owner etc. so this LMS helps to store those types of information using salesforce. The System Lease Management System can be used to manage the data of all type of Building. It will support both stand alone and also networking environment. The system uses SALESFORCE TECHNOLOGY.

## MAIN MODULES OF THE SYSTEM

- **Admin Login:** The system is under supervision of admin who manages the all the Activity
- **User login/registration:** Users have to first register themselves to login into the system.
- **Add Building:** The system will provide this Functionality to Director different locations.

**Coordinator:** Coordinator will provide the pictures of building and unites of client's interest via email and schedule the meet with owner.

Create Lease Contract: LMS Provide the Digital Contract to the client and easily

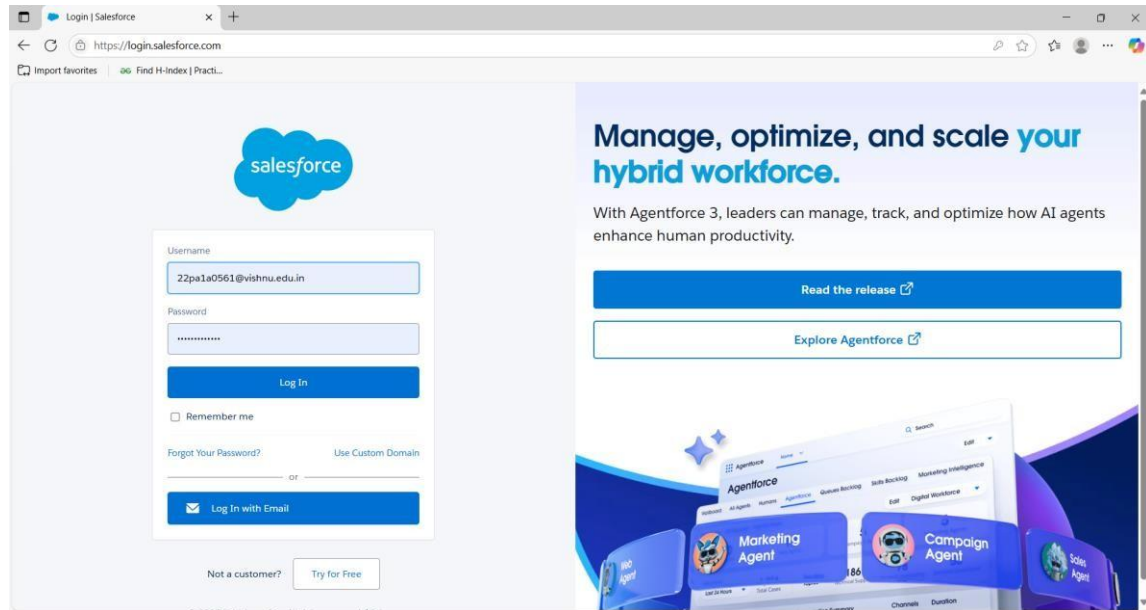
**Alerts :** LMS is Provide Alerts to clients using the various functionality of salesforce Platform.

**Feedback:** The system has a feedback form, where user can provide feedback into the system.

## WORKING

This Lease Management System (CMS) is composed of salesforce components using apex language and salesforce tools. Salesforce is a cloud service providers that offer free software as a Service (SaaS). Using any browser like Google Chrome, Mozilla Firefox, Opera mini, Safari, and Brave, the user can access this software. Administrator can add, edit, delete, and deactivate the processes, data and flows. Also, it can set, update/ change the settings for the profiles present in App.


### 3.1 Login Window



This is LOGIN PAGE. The registered user has to Enter the USERNAME and PASSWORD to enter in the org . The unregistered user has to click on the Log In option.

**Figure 1:** Login Window

In the REGISTRATION form the user has to fill details like First Name, Last Name, Email, etc



**Sign up for your Salesforce Developer Edition**  
A full-featured copy of the Platform, for free

First Name\*

Last Name\*

Email\*

Role\*

Company\*

Country/Region\*

Postal Code\*

Username\*

Your username must be in the form of an email address (it does not have to be real). It must be unique and cannot be associated with another Salesforce login credential.  
[Read more about username recommendations.](#)

By registering, you confirm you have read and agree to the [Terms of Use](#) and the [Main Services Agreement](#) and to the storing and processing of your personal data by Salesforce as described in the [Privacy Statement](#), including use for marketing purposes.

**Sign me Up**

Already have a Salesforce Developer Environment?

[Log in](#)

**Figure 2:** Registration Window

## 2. IDEATION PHASE

### 2.1 Problem Statement

Organizations that manage properties or equipment on lease often face challenges in tracking lease agreements, tenant information, payment schedules, and lease renewals using manual or disconnected systems. This leads to data inconsistencies, missed payments, lack of centralized visibility, and poor tenant communication.

There is a critical need for a **centralized Lease Management System** built on the **Salesforce platform** to automate and streamline lease lifecycle processes. This includes managing tenant records, creating lease agreements, automating monthly rent invoicing, tracking payment status, handling lease renewals or terminations, and generating insightful reports.

The goal is to enhance operational efficiency, ensure timely payments, reduce manual workload, and provide better service to tenants—all within the robust and customizable Salesforce ecosystem.

## 2.2 Empathy Map Canvas

Category	Details
Who are we empathizing with?	Property Managers, Lease Administrators, and Tenants
What do they need to do?	- Manage lease agreements efficiently- Track rent payments and due dates- Handle lease renewals or terminations- Communicate clearly with tenants- Generate reports for lease tracking and compliance
What do they see?	- Disorganized lease data- Missed payment alerts- Complex manual processes- Multiple disconnected tools like Excel sheets or offline files
What do they say?	- "It's hard to track who paid and who didn't."- "Renewal deadlines often sneak up on us."- "I wish everything was in one system."- "Tenant communication takes too long."
What do they do?	- Use spreadsheets or manual files- Follow up on payments via calls or emails- Spend time manually updating lease records
What do they hear?	- Feedback from tenants on late reminders or errors- Management demands for accurate reports- Complaints about miscommunication
Pain Points	- Time-consuming data entry- Missed rent payments- Manual followups and renewal tracking- Lack of real-time visibility
Gains	- Centralized lease data in Salesforce- Automated rent reminders- Easy-to-use dashboards- Efficient communication and tracking

## 2.3 Brainstorming

The brainstorming phase involved identifying key features, components, and processes required to build an effective Lease Management solution in Salesforce. The goal was to understand the problems faced by users and to determine how Salesforce tools (like custom objects, flows, and automation) can be used to solve them.

### Key Questions Asked:

- How can we store and manage lease agreements in Salesforce?
- How do we track rent payments and due dates?
- Can we automate monthly rent reminders to tenants?
- How do we handle lease renewals or terminations?
- What kind of dashboards or reports are useful for management?

### Ideas Generated:

<b>Idea</b>	<b>Description</b>
<b>Custom Objects</b>	Create custom objects for Lease, Tenant, Payment, Property, and Lease Terms.
<b>Automation with Flows</b>	Use Salesforce Flows to automate rent reminder emails and lease renewal notices.
<b>Scheduled Jobs</b>	Automate monthly invoice generation using scheduled flows or Apex.
<b>Tenant Portal</b>	Provide a self-service portal for tenants to view their lease details and payment history (optional).
<b>Reports &amp; Dashboards</b>	Visual dashboards for admin to track active leases, due payments, and lease expirations.
<b>Validation Rules</b>	Ensure lease records are correctly filled before saving (e.g., mandatory fields like start date, rent amount).
<b>Email Templates</b>	Predefined email formats for payment reminders, lease renewal notices, and welcome messages.
<b>Chatter Integration</b>	Internal collaboration and updates using Salesforce Chatter.

### Tools Considered:

- **Salesforce Objects** (Standard + Custom)
- **Process Builder / Flows**
- **Apex** (if needed for custom logic)
- **Reports and Dashboards**
- **Email Alerts**
- **Validation Rules**

### Brainstorm & Idea Prioritization Template:



<b>Feature / Idea</b>	<b>Description</b>	<b>Importance (High/Medium/Low)</b>	<b>Feasibility (Easy/Medium/Hard)</b>	<b>Priority (High/Medium/Low)</b>
<b>Custom Objects</b>	Lease, Tenant, Property, Payment, Lease Terms	High	Easy	High
<b>Flow Automation</b>	Automate monthly rent reminders and lease renewals	High	Medium	High
<b>Reports &amp; Dashboards</b>	Visual reports for payments, lease status, and expiry	Medium	Easy	High
<b>Email Templates</b>	Predefined templates for communication (rent, renewal)	Medium	Easy	Medium
<b>Validation Rules</b>	Ensure data integrity in lease and tenant records	High	Medium	High
<b>Scheduled Flows</b>	Monthly invoice creation and notifications	High	Medium	High
<b>Chatter Integration</b>	Internal communication about lease activities	Low	Easy	Low
<b>Tenant Portal (Optional)</b>	Tenants can view details and payment history	Low	Hard	Low

<b>Mobile Compatibility</b>	Access lease details on Salesforce mobile app	Medium	Medium	Medium
<b>Feature / Idea</b>	<b>Description</b>	<b>Importance (High/Medium/Low)</b>	<b>Feasibility (Easy/Medium/Hard)</b>	<b>Priority (High/Medium/Low)</b>
<b>Payment Tracking</b>	Record status of monthly payments	High	Easy	High

## System Design Overview

### Objects & Key Fields:

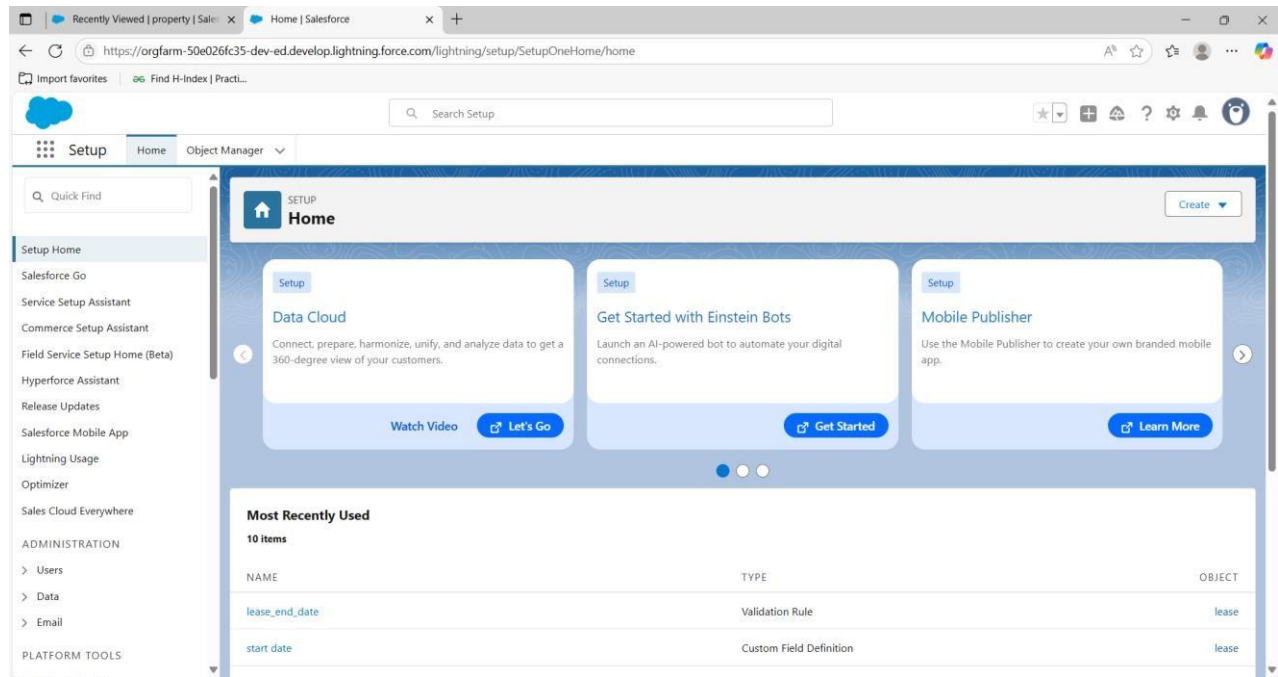
- **Tenant:** Name, Email, Phone, Address
- **Lease:** Lease Name, Tenant (Lookup), Start Date, End Date, Monthly Rent, Status (Active/Expired)
- **Payment:** Lease (Lookup), Payment Date, Amount, Status (Pending/Paid)

## Implementation

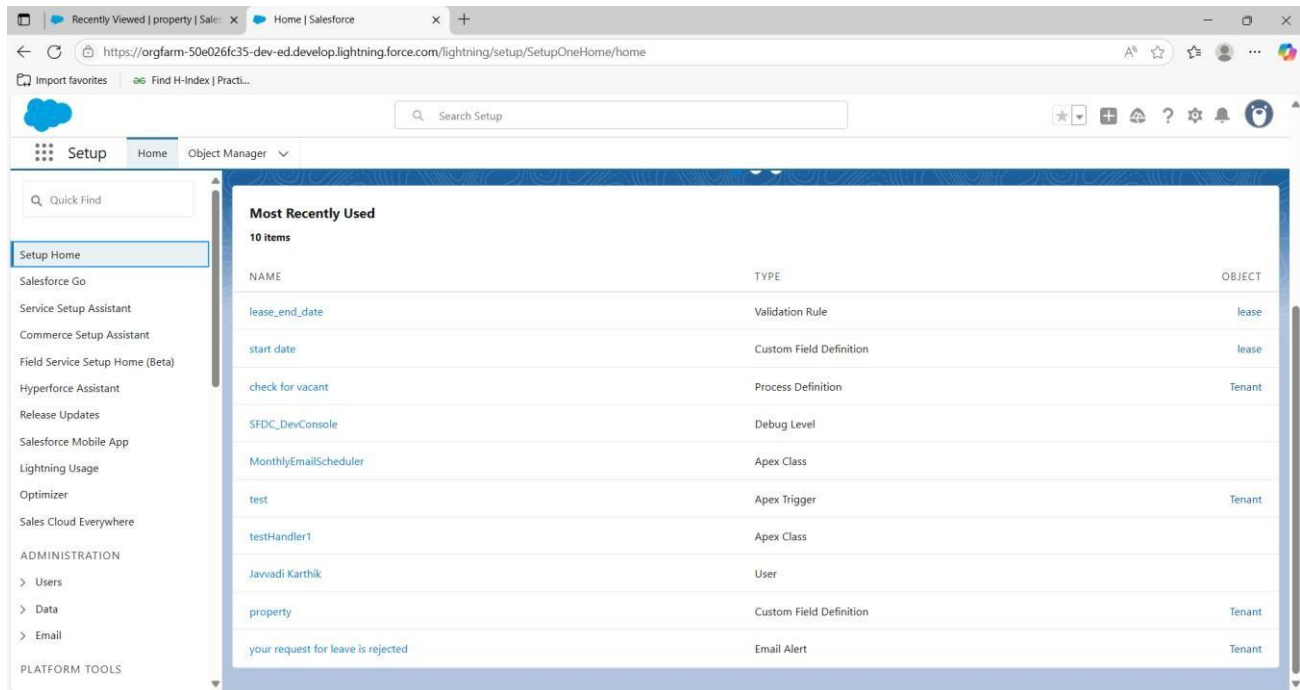
1. **Create “Lease Management” Lightning App** ○  
Include tabs: Tenant, Lease, Payment  
(see App Manager screenshot above)
2. **Define Custom Objects & Fields** ○ Tenant:  
Name, Email, etc.  
○ Lease: link to Tenant, dates, rent, status ○ Payment:  
link to Lease, amount, status
3. **Build Automation Flows** ○ **Payment Reminder Flow** (Scheduled):
  - Monthly run → email tenants with pending payments
 ○ **Renewal Alert Flow** (Scheduled):
  - Runs daily → alerts if lease ends in next 30 days
4. **Design Reports & Dashboard** ○ Reports:  
Active Leases, Expiring Soon, Payment Status ○  
Dashboard: Pie chart of lease status, bar chart of monthly rent collection

# Screenshots (Sample Images)

- **App Setup in Lightning**



- **Lightning App Builder - Adding Tabs**



## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey map

Stage	User Role	Action	System Interaction (Object)	Touchpoint	Outcome
1. Add Property	Admin	Adds a new <b>Property</b> with location, type, and rental terms	Property Object	Property Tab	Property inventory is maintained
2. Add Tenant	Admin	Creates a new <b>Tenant</b> with personal and contact details	Tenant Object	Tenant Tab	Tenant record is stored
3. Create Lease	Admin	Creates a <b>Lease</b> linked to both Tenant and Property	Lease Object (Lookup to Property, Tenant)	Lease Tab	Lease links Tenant to Property
4. Set Payment	Admin/Finance	Creates <b>Payment</b> records for lease duration	Payment Object (Lookup to Lease)	Payment Tab	Rent tracking enabled
5. Payment Reminder	Automated Flow	System sends <b>monthly payment reminder</b> to Tenant	Flow based on Payment Status	Email / Notification	Tenants are reminded automatically
6. Payment Update	Admin/Finance	Marks Payment as <b>Paid or Pending</b>	Payment Object	Payment Record	Payment history stays up to date
7. Lease Expiry	Admin	Updates Lease as <b>Renewed or Expired</b>	Lease Object	Lease Record	Lease lifecycle is managed efficiently

### 3.2Solution Requirement

#### Functional Requirements

These are the core capabilities the system **must perform**.

1. **Property Management**

- Create, view, edit, and delete property records.
- Track details: property name, address, type, ownership, status.

2. **Tenant Management**

- Maintain tenant profiles with contact info, lease history, and documents.
- Link tenants to properties and leases.

3. **Lease Agreement Tracking** ○ Generate new lease agreements.

- Track start/end dates, duration, terms, rent amount, status (active/expired).
- Upload scanned copies or PDFs.

4. **Rent/Payment Tracking** ○ Record monthly/quarterly payments. ○ Set due dates and automatic payment reminders.

- Integrate with Salesforce Flow to notify tenants or admin for late payments.

5. **Renewals & Terminations** ○ Track lease expiry dates and send renewal alerts.

- Allow early termination with reason tracking.

6. **Reporting and Dashboards** ○ Generate reports for:

- Active/expired leases
- Payment history
- Property occupancy ○ Visual dashboards for lease health and payment status.

7. **User Roles and Permissions** ○ Admin: Full access ○

Property Manager: Manage leases & payments ○

Tenant (Community User): View lease & payment status

## Non-Functional Requirements

These refer to **how** the system should perform.

1. **Scalability** ○ Should support 100+ properties and tenants with no performance lag.

2. **Usability** ○ User-friendly interface with clear navigation for both admin and tenants.

3. **Data Security** ○ Only authorized users can access/edit lease and payment records.

- Role-based sharing and field-level security enabled.

4. **Availability** ○ 99.9% uptime – accessible 24/7 via Salesforce Cloud.

5. **Integration** ○ Integration with email (Outlook/Gmail) and calendar for reminders.

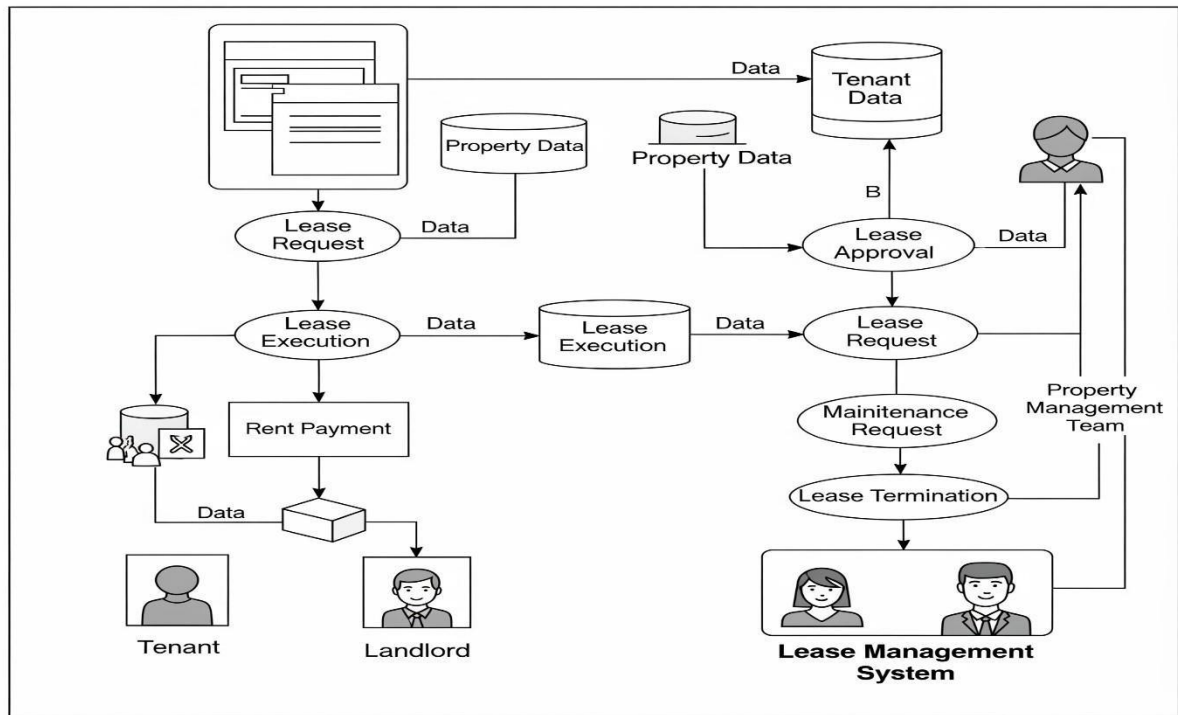
- Optional: Payment gateway (e.g., Razorpay, Stripe).
- 6. **Automation**
  - Salesforce Flows to automate:
    - Payment reminders
    - Lease renewal notices
    - Monthly reports

### System Requirements (Salesforce Specific)

Component	Requirement
Salesforce Edition	Enterprise/Developer
Objects Used	Custom Objects: Property, Tenant, Lease, Payment
Automation Tools	Flows, Process Builder, Approval Processes
Reports & Dashboards	Custom report types and dashboards
Page Layouts	Record pages customized using Lightning App Builder
App Type	Lightning App

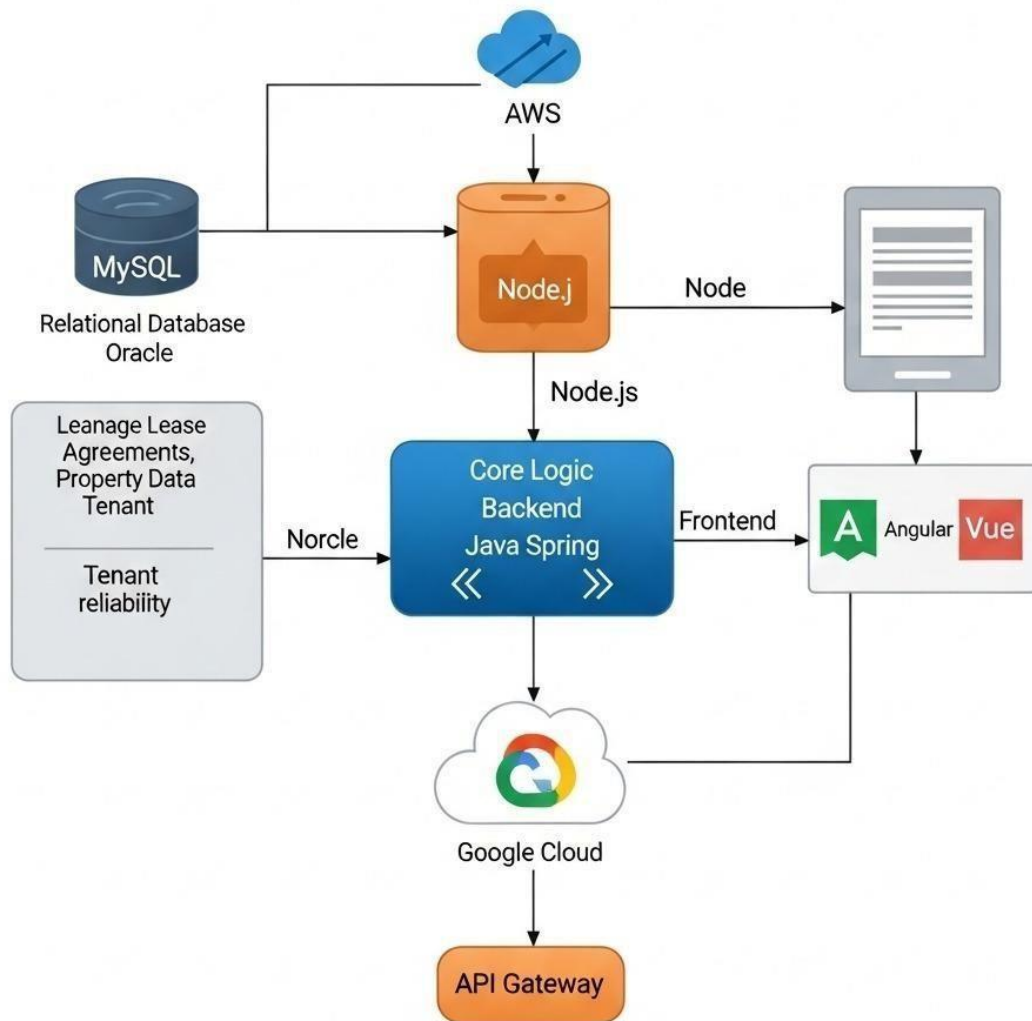
### 3.3 Data Flow Diagram

## Lease Management



### 3.4 Technology Stack:

# Lease Management System



The Lease Management System is built using tools and technologies provided by Salesforce. The main technologies used are:

## 1. Salesforce Platform

- Main platform used for building the application.
- Provides tools for development, automation, and data storage.

## 2. Frontend (User Interface)

- **Lightning App Builder** – to design pages.
- **Lightning Web Components (LWC)** – to create custom components.
- **Tabs & Page Layouts** – to organize data views.



### *3. Backend (Logic & Automation)*

- **Apex** – used to write custom logic (classes, triggers).
- **SOQL** – used to query data from Salesforce objects.
- **Flow Builder** – to automate tasks without code (e.g., send emails, update fields).

### *4. Database*

- **Custom Objects** – like Lease, Tenant, Property, and Payment to store project data.
- **Standard Objects** – like Account and Contact, used if needed.

### *5. Integration (if required)*

- **REST API / SOAP API** – to connect with other apps.
- **Named Credentials** – to store external API login info securely.

### *6. Development Tools*

- **Developer Console** – for writing and testing code.
- **VS Code + Salesforce CLI** – for professional development.
- **Change Sets** – for moving changes from sandbox to production.

### *7. Testing Tools*

- **Apex Test Classes** – to test code.
- **Debug Logs** – to find and fix errors.
- **Workbench** – to run queries and test APIs.

## 4. PROJECT DESIGN

### 4.1 Problem-Solution Fit

#### Problem Statement:

Organizations that manage properties or equipment on lease often face challenges in tracking lease agreements, tenant information, payment schedules, and lease renewals using manual or disconnected systems. This results in:

- Data inconsistencies
- Missed payments
- Lack of centralized visibility
- Poor tenant communication

#### Solution Fit:

There is a critical need for a centralized **Lease Management System** built on the **Salesforce platform** to automate and streamline the lease lifecycle. Salesforce offers a powerful low-code platform that supports custom objects, workflows, and automation tools. This solution aims to:

- Manage tenant records efficiently
- Create and approve lease agreements
- Automate monthly rent invoicing
- Track payments and lease renewals
- Generate real-time reports and dashboards

This centralized system will enhance operational efficiency, reduce manual workload, and improve tenant service quality within the robust Salesforce ecosystem.

### 4.2 Proposed Solution

To solve the identified challenges, the Lease Management System in Salesforce will include:

#### *Custom Objects:*

- **Tenant:** Stores contact, ID proof, and history.
- **Property:** Captures property info such as location, size, and status.
- **Lease:** Links tenant to property with lease dates, status, and terms.
- **Payment:** Tracks rent amount, due date, status (Paid/Unpaid).

#### *Automation & Flows:*

- **Monthly Rent Reminder:** Sends notifications to tenants before due date.
- **Lease Expiry Alerts:** Notifies admin and tenant before lease ends.
- **Validation Rules:** Ensure correct data entry for leases and payments.

#### *Approval Process:*

- **Lease Agreement Approval:** Once a new lease is created, it triggers an approval process.
  - Goes to Property Manager for review.
  - Status updates to “Approved” or “Rejected”.
  - Upon approval, related flows (e.g., payment scheduling) are activated.
  - Email alerts are sent to notify stakeholders.

#### *Reports & Dashboards:*

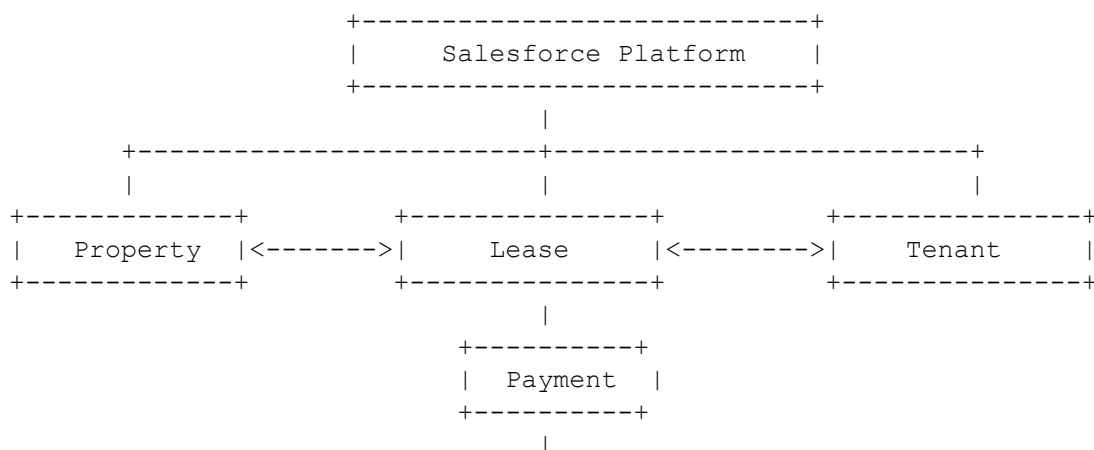
- Lease Status Overview
- Monthly Revenue
- Late Payment Summary
- Lease Renewal Pipeline

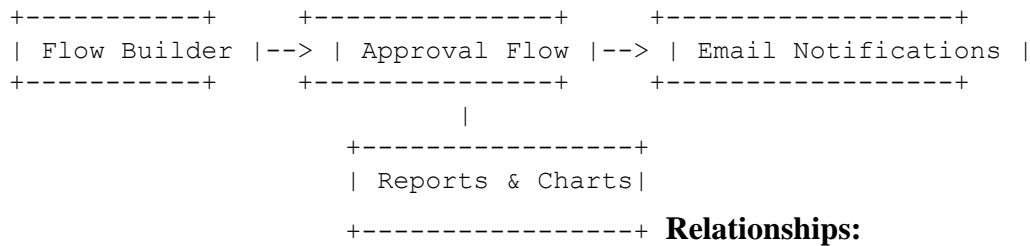
#### *User Interface:*

- Custom Tabs for Tenant, Property, Lease, Payment
- Lightning Pages for record-level views
- List Views for admin monitoring

## 4.3 Solution Architecture

The following diagram explains the structure and relationships of the system:





- One Property → Many Leases
  - One Tenant → Many Leases
  - One Lease → Many Payments
- Salesforce Tools Used:**

- **Object Manager** for custom objects
- **Flow Builder** for automation
- **Approval Process** for lease validation
- **Lightning App Builder** for user experience
- **Reports & Dashboards** for analytics

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Objectives

- Develop a centralized Lease Management System on Salesforce.
- Automate lease creation, approvals, rent invoicing, and renewals.
- Create custom objects for Property, Lease, Tenant, and Payment.
- Implement flows and approval processes to streamline tasks.
- Build dashboards and reports for real-time business insights.
- Enhance user experience using Salesforce Lightning interface.

### 5.2 Project Scope

*In Scope:*

- Custom object and relationship design (Property, Lease, Tenant, Payment).
- Flow automation for payment reminders and lease expiry.
- Approval process for lease agreement validation.
- Reports and dashboards for lease and payment tracking.
- User interface setup with tabs and list views.

*Out of Scope:*

- Integration with external payment gateways.
- Mobile application development.
- Tenant self-service portal (unless added later).

### 5.3 Project Timeline

Week	Milestones
Week 1	- Finalize requirements and user roles - Design data model & relationships - Plan flows & approval
Week 2	- Create custom objects & fields - Build Lightning record pages - Set up app and custom tabs
Week 3	- Configure Flows for rent reminders and expiry - Implement Approval Process - Add validation rules
Week 4	- Create dashboards & reports - Conduct UAT (User Acceptance Testing) - Final deployment & handover

### 5.4 Resource Allocation

Role	Responsibility
Project Manager	Oversees planning, ensures milestones are met
Salesforce Developer	Builds objects, flows, approvals, and dashboards
QA/User Testers	Runs tests and validates functionality

### 5.5 Risk Management Plan

Risk	Impact	Mitigation Strategy
Flow errors or logic failures	Medium	Use debug logs and sandbox testing
Late requirement changes	High	Lock scope and review weekly
Delays in approval process	Medium	Configure escalations in approval workflow
Insufficient user training	Medium	Provide demo sessions and training guides

## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 Functional Testing

Functional testing ensures that each feature of the Lease Management system performs as intended. Below are the key functional test scenarios:

Test Case ID	Test Scenario	Expected Result	Status
TC-001	Create a new Tenant record	Tenant record is created and visible in list view	✓ Passed
TC-002	Link a Lease to a Property and Tenant	Lease record links correctly to both	✓ Passed
TC-003	Trigger Lease Approval Process	Approval email sent to manager, status set to Pending	✓ Passed
TC-004	Approve a Lease	Status updates to "Approved", flows are triggered	✓ Passed
TC-005	Generate Payment record monthly	Payment record auto-created before due date	✓ Passed
TC-006	Send rent reminder email	Email received by tenant 3 days before due date	✓ Passed
TC-007	Lease expiry alert	Notification sent 7 days before lease end	✓ Passed
TC-008	Dashboard loads payment report	Correct payment data visible in dashboard	✓ Passed
TC-009	Validation rule on empty lease field	Error message shown if required fields are empty	✓ Passed

## 6.2 Performance Testing

Performance testing ensures the system responds efficiently under expected usage conditions.

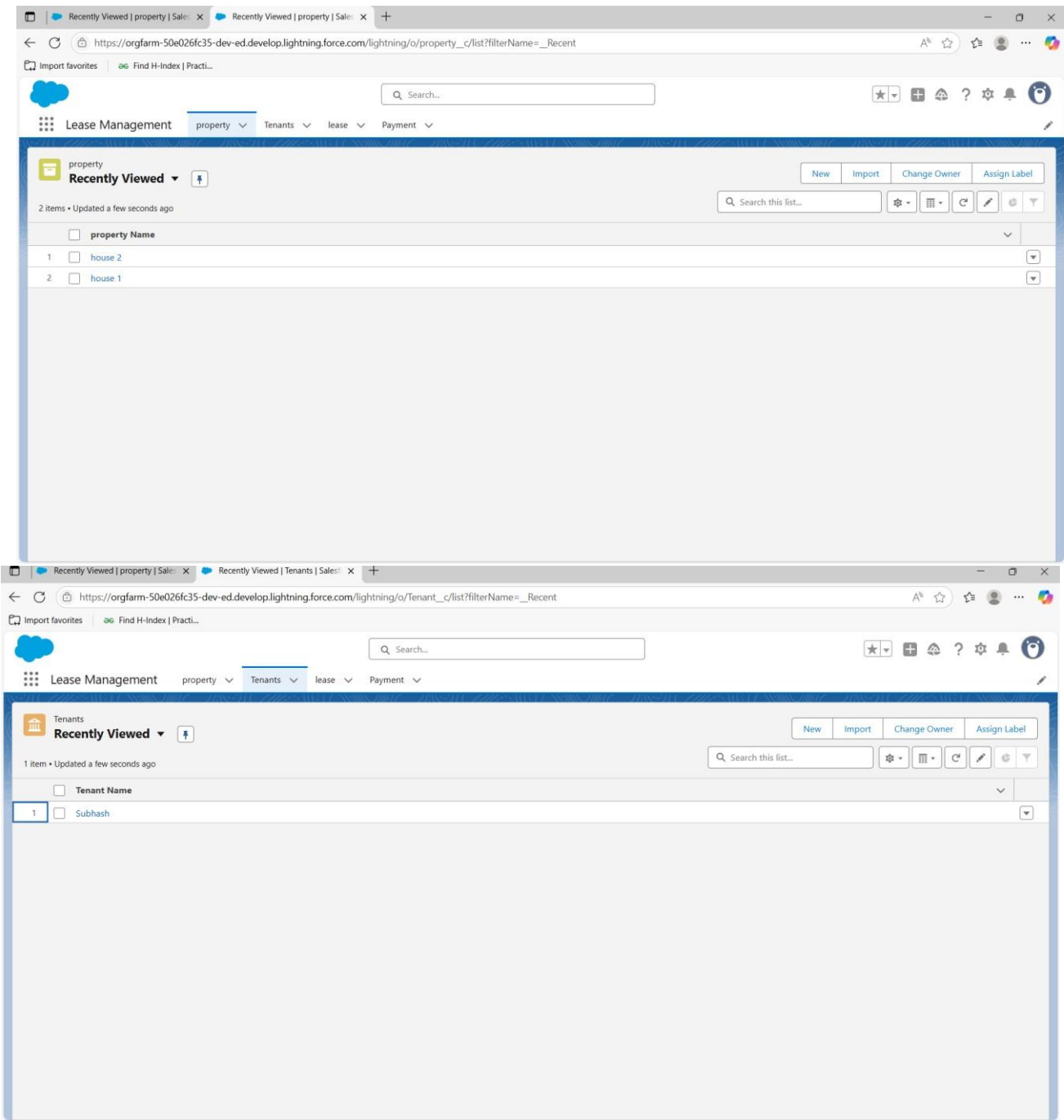
Test Type	Scenario	Expected Result	Status
Load Testing	Load dashboard with 1000+ lease records	Loads within 3 seconds	✓ Passed
Flow Execution	Trigger rent reminder flow for 100 tenants	Emails sent within 1 minute	✓ Passed
Approval Workflow	Process 50 lease approvals in sequence	All processed within 2 minutes	✓ Passed
Record Creation	Bulk import 500 tenants via Data Loader	All records imported successfully	✓ Passed
Test Type	Scenario	Expected Result	Status
Dashboard Refresh	Refresh lease summary dashboard	Refresh completed under 5 seconds	✓ Passed

## 6.3 Testing Tools Used

- **Salesforce Developer Console** – For debug logs and Apex test execution
- **Data Loader** – To bulk create records for testing
- **Flow Debugger** – To trace flow paths and variables
- **Manual UI Testing** – To verify field visibility, tab layouts, and record behavior
- **Email Logs** – For tracking approval and reminder email delivery

# 7. RESULTS

## 7.1 Output Screenshots





## 8. ADVANTAGES & DISADVANTAGES

### Advantages of Lease Management in Salesforce

1. **Centralized Data Management** ○ All lease-related data (tenants, leases, properties, and payments) are stored in one system for easy access and tracking.
2. **Automation of Lease Processes** ○ Automates tasks such as payment reminders, lease renewals, and notifications, reducing manual effort and errors.
3. **Customizable and Scalable** ○ Salesforce allows customization of objects, fields, flows, and automation as per specific business needs.
4. **Real-time Reporting & Dashboards** ○ Easily generate reports and dashboards to track lease performance, payments, upcoming expiries, and more.
5. **Improved Communication** ○ Integration with email and Chatter enhances communication between tenants, landlords, and internal teams.
6. **Mobile Accessibility** ○ Salesforce's mobile app allows users to view and update lease information on the go.
7. **Security & Role-Based Access** ○ Ensures only authorized users can view/edit data, protecting sensitive lease and tenant information.

### Disadvantages of Lease Management in Salesforce

1. **Initial Setup Complexity** ○ Setting up custom objects, flows, and reports requires admin or developer expertise.
2. **Licensing Cost** ○ Salesforce can be expensive for small businesses, especially when scaling up with more users and customizations.
3. **Training Required** ○ End-users (like property managers) may need training to fully use the system and understand Salesforce terminology.
4. **Dependency on Internet** ○ Being a cloud platform, Salesforce needs an active internet connection for access.
5. **Customization Limitations Without Code** ○ Some complex logic or automation requires Apex coding, which may need developer support.

## 9. CONCLUSION

The Lease Management System built on the Salesforce platform provides an efficient and streamlined solution for managing lease-related operations. By leveraging Salesforce's powerful CRM and automation capabilities, the system simplifies the tracking of tenants, properties, leases, and payments within a centralized environment.

The project successfully demonstrates how organizations can replace manual, disconnected methods with a digital platform that offers automation, real-time data access, security, and scalability. With features like automated reminders, real-time dashboards, and customizable flows, property managers and leasing agents can improve decision-making, ensure compliance, and enhance tenant relationships.

Despite initial setup complexity and cost considerations, the long-term benefits of automation, centralized management, and improved operational efficiency make this solution a valuable investment for any lease-based business.

This Lease Management solution not only meets current business needs but also lays a strong foundation for future growth and integration with other business systems.

## 10. FUTURE SCOPE

The Lease Management System developed on the Salesforce platform has strong potential for future enhancements and integrations. As the business grows, the system can be scaled and extended to offer additional features, improve automation, and integrate with third-party tools. Below are some of the key areas for future development:

### 1. Integration with External Payment Gateways

- Integration with payment systems like Razorpay, Stripe, or PayPal for automated rent collection and reconciliation.

### 2. Document Management & E-signatures

- Incorporate document generation (PDF lease contracts) and digital signatures using Salesforce integrations with tools like **DocuSign** or **Adobe Sign**.

### 3. Tenant Portal

- Develop a **tenant self-service portal** for online rent payments, lease renewals, and communication.

#### 4. Advanced Analytics with AI

- Use Salesforce **Einstein Analytics** or **AI tools** to predict lease renewals, identify late payments trends, or suggest rental pricing strategies.

#### Facility & Maintenance Management

- Add modules for **facility management**, **maintenance requests**, and **service scheduling**.

#### Multi-property & Multi-location Support

- Scale the system to manage multiple properties across various cities or regions with localized compliance settings.

#### Chatbot or WhatsApp Integration

- Enable chatbots or integrate WhatsApp for tenant communication, rent reminders, and query handling.

By planning for these enhancements, the Lease Management system can evolve into a fullfledged **Property Management Suite**, offering both operational efficiency and tenant satisfaction.

## 11. APPENDIX

### Source Code(if any)

#### Apex Trigger

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

```

#### Apex Handler class

```
for (Tenant__c newTenant : newList) {

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

```

#### Apex Class

```
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});
    }
}
}

```

**Dataset Link:**

<https://www.kaggle.com/code/azratuni/lease-document-processing-analysisleaseflow/input>

**GitHub & Project Demo Link :**

[https://github.com/JKarthik2207/Salesforce\\_lease\\_management](https://github.com/JKarthik2207/Salesforce_lease_management)