ALAGAPPA CHETTIAR GOVERNMENT COLLEGE OF ENGINEERING AND TECHNOLOGY, KARAIKUDI – 630003

(An Autonomous Institution Affiliated to ANNA UNIVERSITY, Chennai - 600025)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Project Report on

SALESFORCE - Lease Management system

BY

SUBIKSHA M V - 91762315205

DEEPA M - 91762215010

GANIKA SRI T - 91762215017

NISHANTHI S – 91762215035

PHASE 1-IDEATION

1.PROJECT OVERVIEW

The Lease Management System is a smart platform that simplifies how businesses handle lease agreements. It brings all lease-related activities into one place, helping track payments, renewals, and compliance more efficiently. By replacing manual processes, it reduces errors, delays, and confusion.

One of its main strengths is automation. It sends alerts for rent payments, renewals, and contract expirations ensuring deadlines are not missed. This reduces human error and keeps operations smooth. The system provides detailed reports on lease status, payment history, and upcoming events. This helps businesses make informed decisions based on real-time data.

By digitizing workflows, it saves time, improves accuracy, and ensures all teams access consistent information. It's especially useful for businesses handling multiple properties.

The system covers the entire lease lifecycle from creation to renewal or termination helping teams focus on key tasks instead of paperwork.

1.1 Technical Approach

The Lease Management System is designed to be simple, fast, and secure. It uses a central database to store all lease-related data, ensuring quick access and better data management.

To reduce manual work, the system uses automation for tasks like payment reminders and compliance checks. This helps avoid delays and improves accuracy.

Real-time updates ensure that any change made like payment status or lease details—is instantly visible to all users, keeping information consistent.

The platform is built to be flexible and scalable, allowing future upgrades like adding new reports or integrating with other tools.

The user interface is clean and easy to use, designed even for non-technical users to work without confusion.

1.2 Benefits:

- **Centralized Data Access**
  All lease information is stored in one secure place for easy access and updates.
- **Time-Saving Automation**
  Automatic alerts and checks reduce manual work and prevent missed deadlines.
- **Real-Time Updates**
  Any change is instantly reflected for all users, ensuring accurate and current data.
- **Improved Decision-Making**
  Built-in reports provide insights into leases, helping businesses plan better.
- **User-Friendly Interface**
  Simple design makes it easy for anyone to use, even without technical skills.
- **Enhanced Security**
  Login control, user roles, and data protection keep sensitive info safe.
- **Scalability and Flexibility**
  The system can grow with the business and easily integrate with other tools.

2. OBJECTIVE

This section highlights the clear and measurable goals that the Lease Management System aims to achieve. These objectives focus on solving key business challenges, streamlining operations, and delivering real value to both property managers and tenants.

2.1 Business Goals

- **Streamline Lease Management:**
  Develop a user-friendly platform that simplifies the management of lease agreements, tenant profiles, and property data, minimizing manual effort and bringing structure to the entire process.
- **Enhance Data Security and Controlled Access:**

Securely store tenant and property information, ensuring that only authorized individuals can view or modify sensitive data.

- **Automate Lease Renewal and Termination Notifications:**
  Automatically send timely alerts for upcoming lease renewals or terminations to keep property managers and tenants informed and prepared.
- **Efficient Payment Tracking:**
  Monitor rent payments and due dates effectively, giving property managers a clear view of payment history and overall financial performance.
- **Generate Insightful Reports and Analytics:**
  Provide real-time reports on lease status, payment trends, and property performance to support better decision-making and long-term planning.

2.2 Specific Outcomes

- **Intuitive User Interface:**
  Provide a clean, easy-to-navigate interface that allows both tenants and managers to manage and access lease data without confusion.
- **Automatic Reminders:**
  Implement alert systems to notify users of important events like rent due dates, lease renewals, and contract expirations.
- **Live Data Synchronization:**
  Ensure that all updates to lease records are reflected instantly across the platform, giving users access to the most accurate and up-to-date information.
- **Report Generation:**
  Enable property managers to create reports summarizing lease activities, payment statuses, and property performance for better analysis and decision-making.
- **Online Rent Payment:**
  Integrate online payment options to allow tenants to pay rent digitally, making the process more convenient and reducing delays.

PHASE 2 -REQUIREMENT ANALYSIS

3.REQUIREMENT ANALYSIS & PLANNING

3.1 Understanding Business Requirements

The first phase of the project focused on understanding the real-world challenges faced by property managers and tenants in managing lease agreements. Through stakeholder meetings, user interviews, and domain research, key pain points were identified to shape the system's core functionalities.

Identified Problems & User Needs:

• Manual handling of lease documents often leads to data entry errors, delays, and lack of consistency.

• Property managers face difficulties in tracking rent payments, lease expirations, and due dates across multiple tenants.

• Lease data is scattered and unstructured, making it hard to access and manage centrally.

• No alert mechanism for upcoming lease renewals or terminations causes missed actions and delays.

• Current systems lack data security, putting sensitive tenant and financial data at risk.

Solution Approach:

The Lease Management System addresses these problems by offering an automated, centralized, and secure digital platform. The system is designed to streamline daily operations, enhance visibility, and improve accuracy in lease tracking and management.

3.2 Defining Project Scope and Objectives

This step involves outlining what Phase 1 of the Lease Management System will deliver and how it will align with the overall business goals. The focus is on building a core functional system that can later be expanded.

Phase 1 Scope Includes**:**

- Centralized platform for managing leases, tenant records, properties, and payments
- User access through secure login based on roles (admin, manager, tenant)
- Lease lifecycle automation: creation, renewal reminders, and termination alerts
- Dashboard displaying lease status, payment due dates, and tenant activity
- Real-time synchronization of updates to ensure consistent and current data
- Flexibility to scale across multiple properties, locations, and user roles
- Data security measures like authentication, access control, and audit trails

Built-in reporting for financial summaries and lease performance Objectives:

- Improve operational efficiency by digitizing manual lease management
- Provide transparency and real-time visibility into lease data
- Enhance communication between tenants and property managers
- Support better decision-making through reporting and analytics
- Ensure system stability and ease of use for non-technical users
- Maintain long-term sustainability with modular and scalable design

The objective is to deliver a reliable MVP (Minimum Viable Product) that solves immediate pain points and forms the foundation for future enhancements.

3.3 Design: Data Model and Security Model

A. Data Model Design

The data model is the backbone of the Lease Management System. It defines how data is structured, stored, and linked across the application. The model ensures efficient data retrieval and clear relationships between different business entities.

Core Entities:

• **Users:** Stores login credentials and role information (Admin, Property Manager, Tenant)

• **Tenants:** Contains personal information, lease history, contact details, and tenant-specific notes

• **Properties:** Stores property details like ID, name, address, and ownership information
• **Leases:** Connects tenants to properties and includes start/end dates, rent amount, and lease status
• **Payments:** Tracks payment records including amount, due date, payment status, method, and remarks Relationships:
• One Property → Many Leases
• One Tenant → Many Leases (historical)
• One Lease → Many Payments

These relationships ensure a normalized, scalable data structure that supports reporting and performance optimization.

B. Security Model Design

Given the sensitivity of lease and financial data, the system incorporates strong security practices to safeguard information and ensure data privacy.

Security Features:

• **User Authentication:**
A secure login system requiring valid username and password combinations, with session control to prevent unauthorized access.

• **Role-Based Access Control (RBAC):**
Users are assigned roles which define what data they can view or modify.

- Tenants can view their own leases and payments.
- Property Managers can manage leases, payments, and tenant profiles.
- Admins have full access to all modules.

• **Data Encryption:**
Sensitive information (e.g., payment methods, user credentials) is encrypted both during transmission (SSL/TLS) and at rest in the database.

• **Audit Logging:**
Every critical operation (e.g., lease updates, payment status changes) is recorded for traceability. This ensures accountability and helps in debugging or compliance.

• **Input Validation & Protection:**
All user input is validated to prevent common web vulnerabilities such as:

- SQL Injection
- CrossSite Scripting (XSS)
- CSRF (Cross-Site Request Forgery)

These security layers protect the system from unauthorized access, data loss, and malicious attacks, ensuring the platform is trustworthy and compliant with best practices.

PHASE 3 -PROJECT DESIGN

4. SALESFORCE DEVELOPMENT – BACKEND & CONFIGURATIONS

4.1 Setup Environment & DevOps Workflow

Milestone 1: Salesforce Account

We established a robust Salesforce development environment using:

• **Developer Org Strategy:** Created Developer Org sandboxes for development and testing.

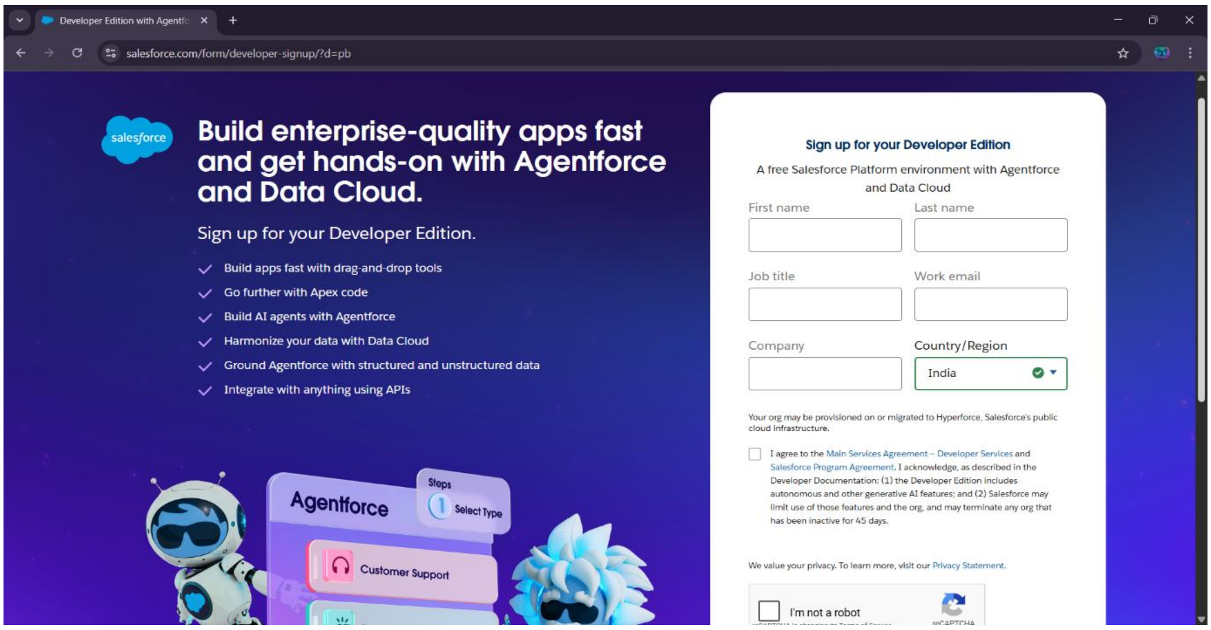
You can create a Developer Org using this official Salesforce link: https://developer.salesforce.com/signup

• **Source Control:** Used GitHub to manage code versions, collaborate with the team, and track all changes efficiently.

• **Deployment Tools:** Utilized Change Sets and Salesforce DX (SFDX CLI) to deploy components between Sandbox and Production environments.

• **Testing Environment:** All features were validated in the Sandbox to ensure quality and stability before live deployment.

This setup supports faster development, safer releases, and better tracking of changes during the entire project lifecycle.


salesforce.com/form/developer-signup/?d=pb

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- Build apps fast with drag-and-drop tools
- Go further with Apex code
- Build AI agents with Agentforce
- Harmonize your data with Data Cloud
- Ground Agentforce with structured and unstructured data
- Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name	Last name
Job title	Work email
Company	Country/Region: India

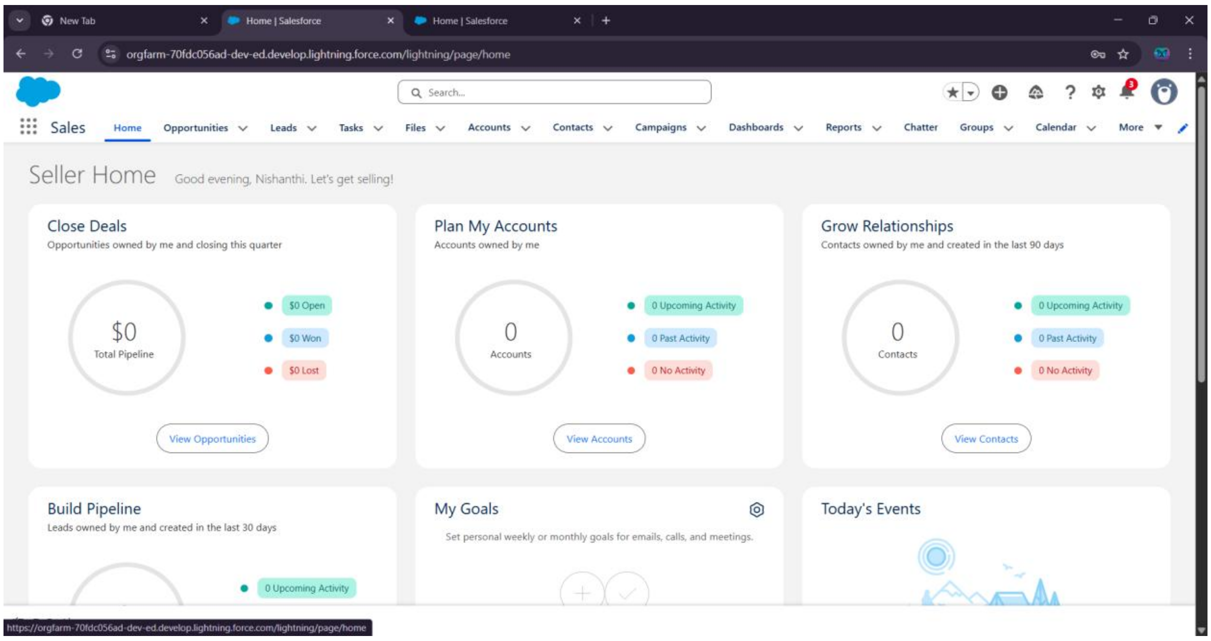
Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☐ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our Privacy Statement.

☐ I'm not a robot

Now, enter your credentials to login into the Salesforce Developer Org.


orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/page/home

Seller Home — Good evening, Nishanthi. Let's get selling!

Close Deals — Opportunities owned by me and closing this quarter: $0 Total Pipeline; $0 Open; $0 Won; $0 Lost. View Opportunities

Plan My Accounts — Accounts owned by me: 0 Accounts; 0 Upcoming Activity; 0 Past Activity; 0 No Activity. View Accounts

Grow Relationships — Contacts owned by me and created in the last 90 days: 0 Contacts; 0 Upcoming Activity; 0 Past Activity; 0 No Activity. View Contacts

Build Pipeline — Leads owned by me and created in the last 30 days: 0 Upcoming Activity

My Goals — Set personal weekly or monthly goals for emails, calls, and meetings.

Today's Events

https://orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/page/home

4.2 Customization of Objects, Fields, Validation Rules, and Automation

Salesforce's declarative (click-based) tools are used to customize the system:

4.2.1 Custom Objects:

Milestone 2: Objects

To model lease data, the following custom objects are created:

➢ **Property**

➢ **Tenant**

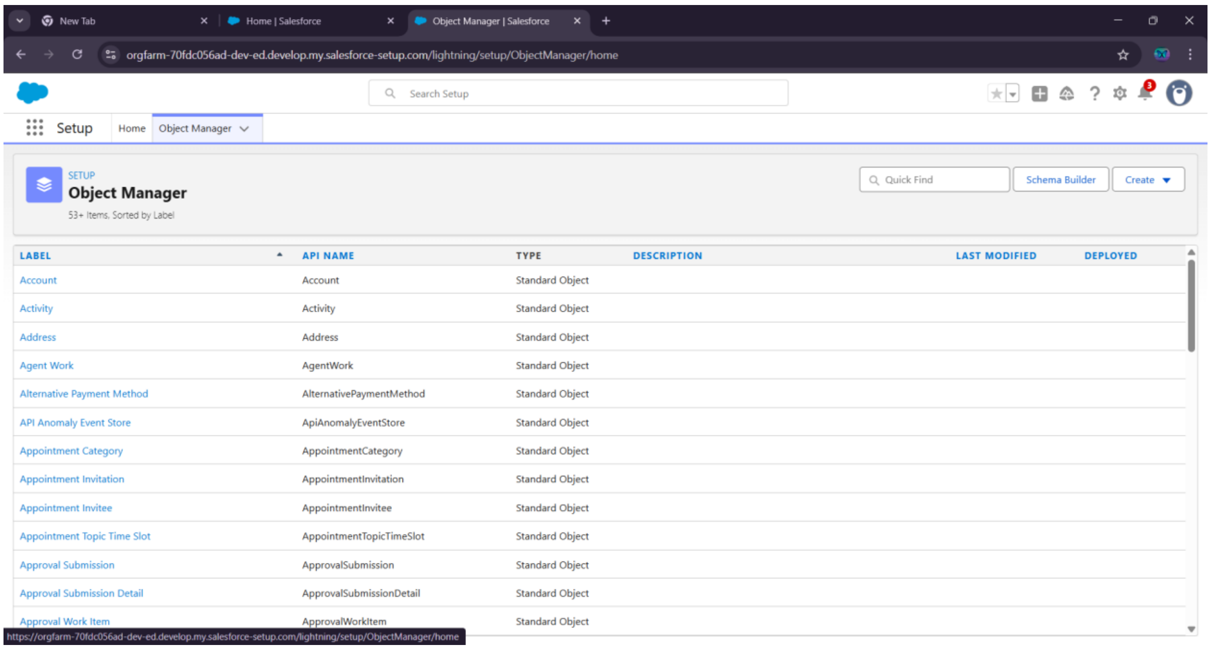
➢ **Lease**

➢ **Payment**

Salesforce's declarative tools were used to create custom objects tailored to the Lease Management System.

These objects include **Property, Tenant, Lease, and Payment,** each representing core business data.

They enable structured storage and management of lease-related records. Custom fields, relationships, and validations were added to meet specific project needs.


New Tab
Home | Salesforce
Object Manager | Salesforce
orgfarm-70fdc056ad-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/home
Search Setup
Setup
Home
Object Manager
SETUP
Quick Find
Schema Builder
Create

Object Manager

53+ Items, Sorted by Label

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Account	Account	Standard Object			
Activity	Activity	Standard Object			
Address	Address	Standard Object			
Agent Work	AgentWork	Standard Object			
Alternative Payment Method	AlternativePaymentMethod	Standard Object			
API Anomaly Event Store	ApiAnomalyEventStore	Standard Object			
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			
Appointment Topic Time Slot	AppointmentTopicTimeSlot	Standard Object			
Approval Submission	ApprovalSubmission	Standard Object			
Approval Submission Detail	ApprovalSubmissionDetail	Standard Object			
Approval Work Item	ApprovalWorkItem	Standard Object			

https://orgfarm-70fdc056ad-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/home

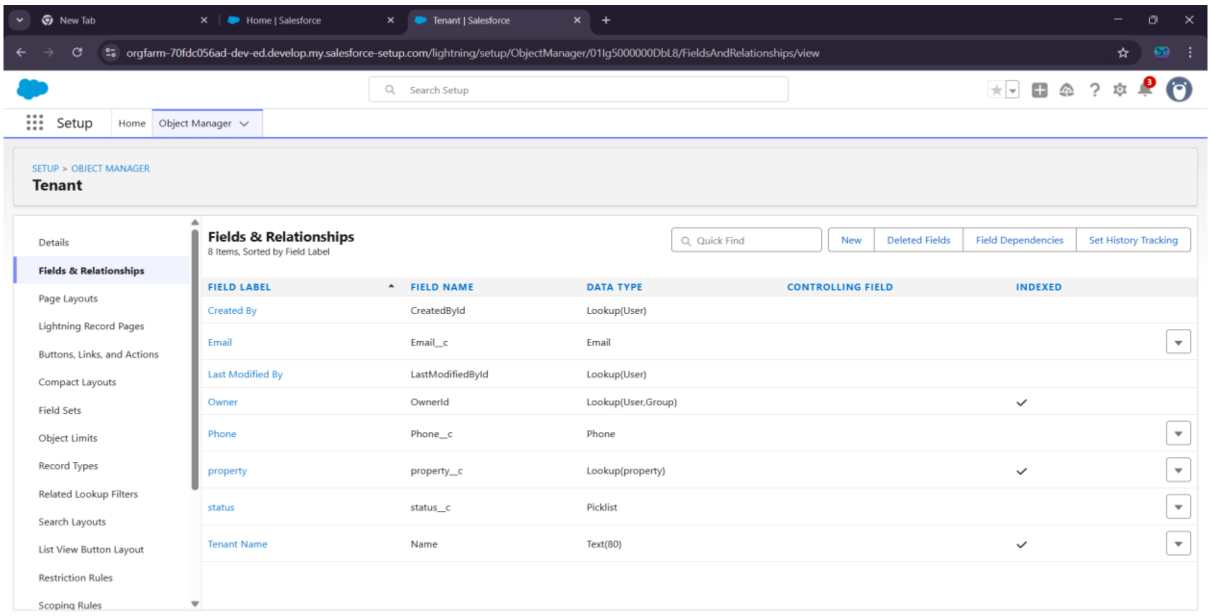
4.2.2 Custom Fields:

Milestone 5: Fields

Each object has custom fields such as:

➢ **Tenant:**

- Email__c
- Phone__c
- Property__c
- Status__c

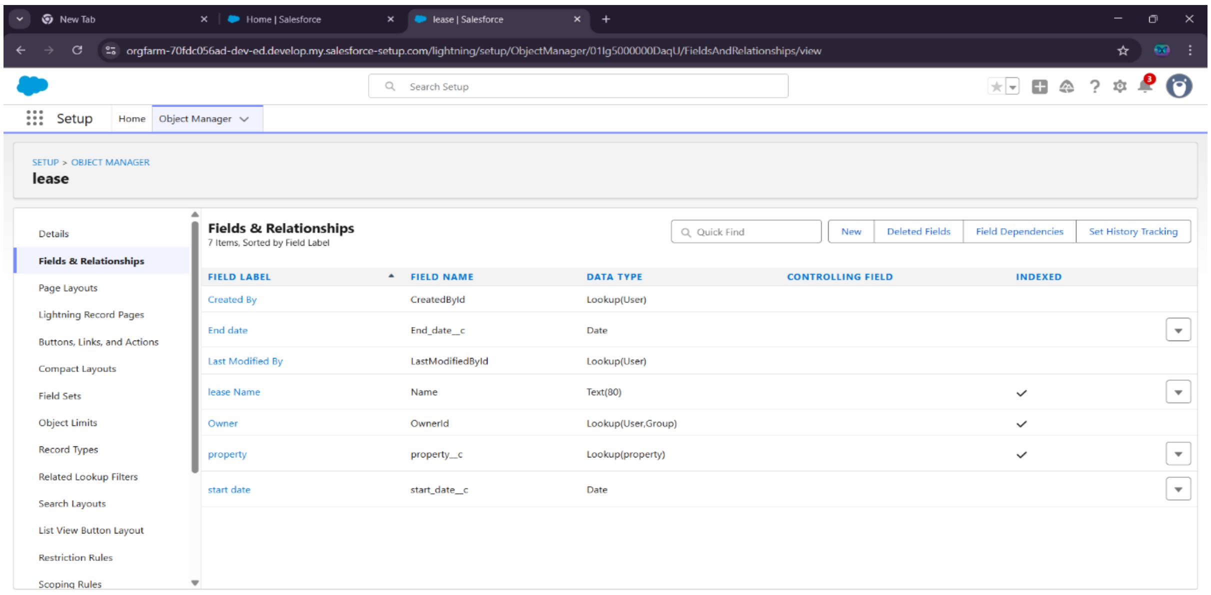


Fields & Relationships
8 Items, Sorted by Field Label

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)		✓

➢ Lease:

- End_date__c
- Property__c
- Start_date__c

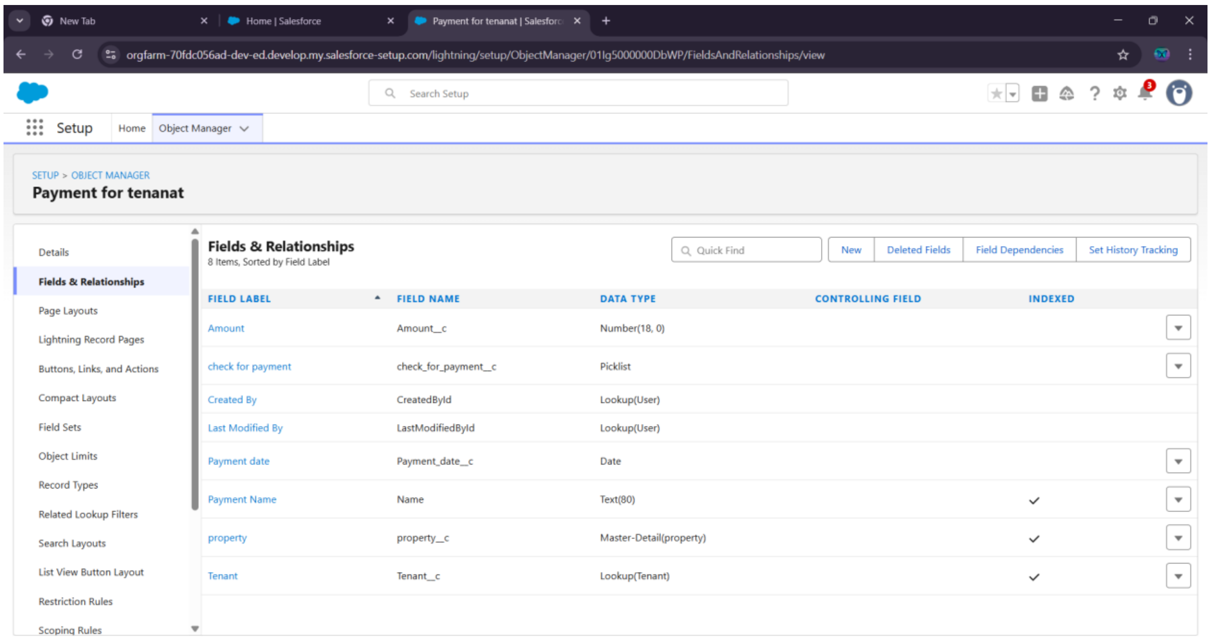


Fields & Relationships
7 Items, Sorted by Field Label

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		

➢Payment For Tenant:

- Amount__c
- Payment_date__c
- Check_for_payment__c
- Property__c
- Tenant__c



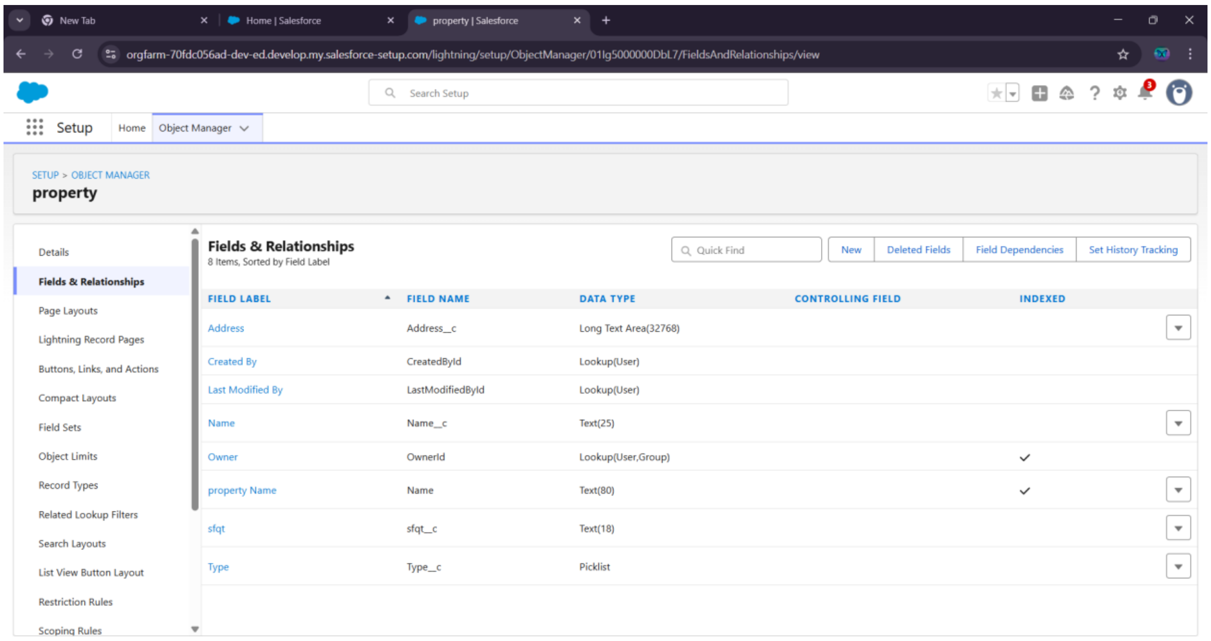
Payment for tenanat

Fields & Relationships
8 Items, Sorted by Field Label

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)		
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(80)		✓
property	property__c	Master-Detail(property)		✓
Tenant	Tenant__c	Lookup(Tenant)		✓

➢ **Property:**

- Address__c
- Name__c
- Sqft__c
- Type__c



property

Fields & Relationships
8 Items, Sorted by Field Label

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 Name	Name	Text(80)		✓
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

4.2.3 Validation Rules:

Milestone 6: Validation Rule

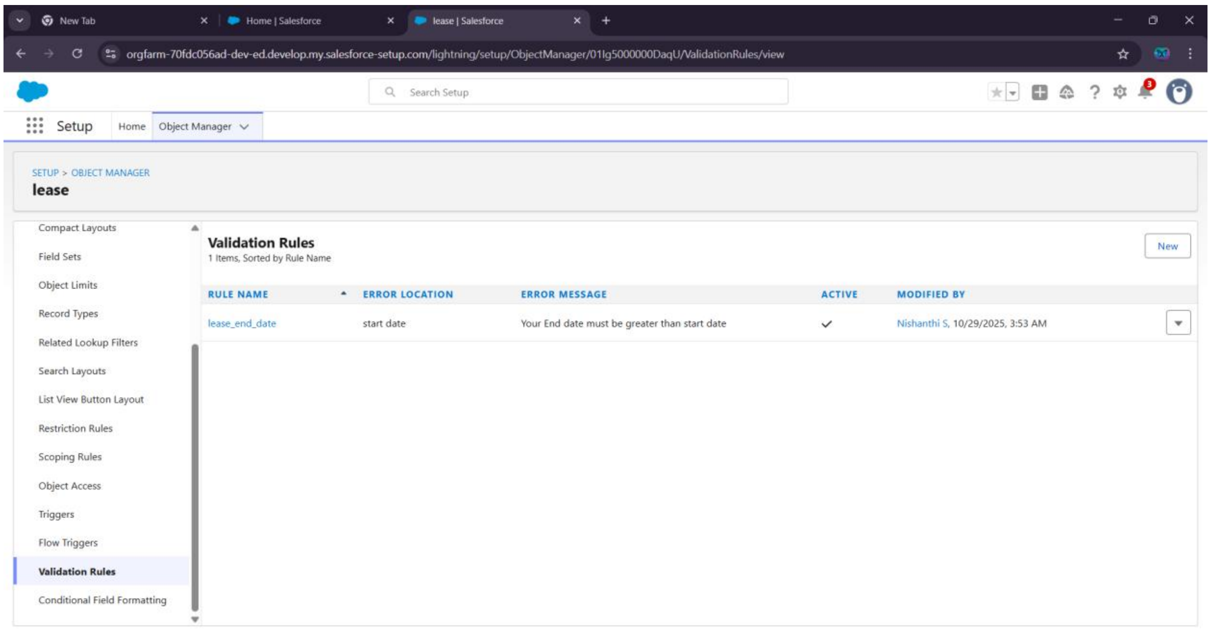
Validation rules ensure data accuracy.

Examples:

➢ Lease end date must be after the start date.

➢ Rent amount must be greater than zero.

➢ Payment date cannot be in the future (unless marked as scheduled).


orgfarm-70fdc056ad-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01Ig5000000DaqU/ValidationRules/view

SETUP > OBJECT MANAGER
lease

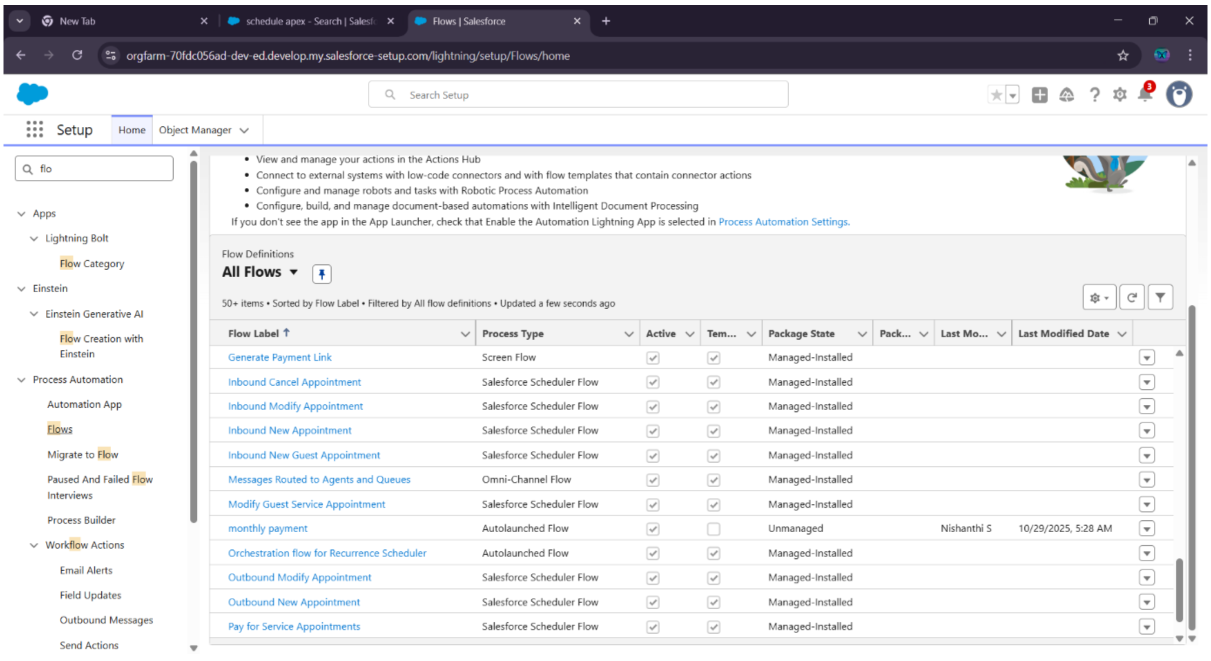
Validation Rules
1 Items, Sorted by Rule Name

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
lease_end_date	start date	Your End date must be greater than start date	✓	Nishanthi S, 10/29/2025, 3:53 AM

4.2.4 Automation Tools:

➢ **Flows:** Used to collect tenant input, auto-create payment records, and send reminders.

Milestone 10: Flows


orgfarm-70fdc056ad-dev-ed.develop.my.salesforce-setup.com/lightning/setup/Flows/home

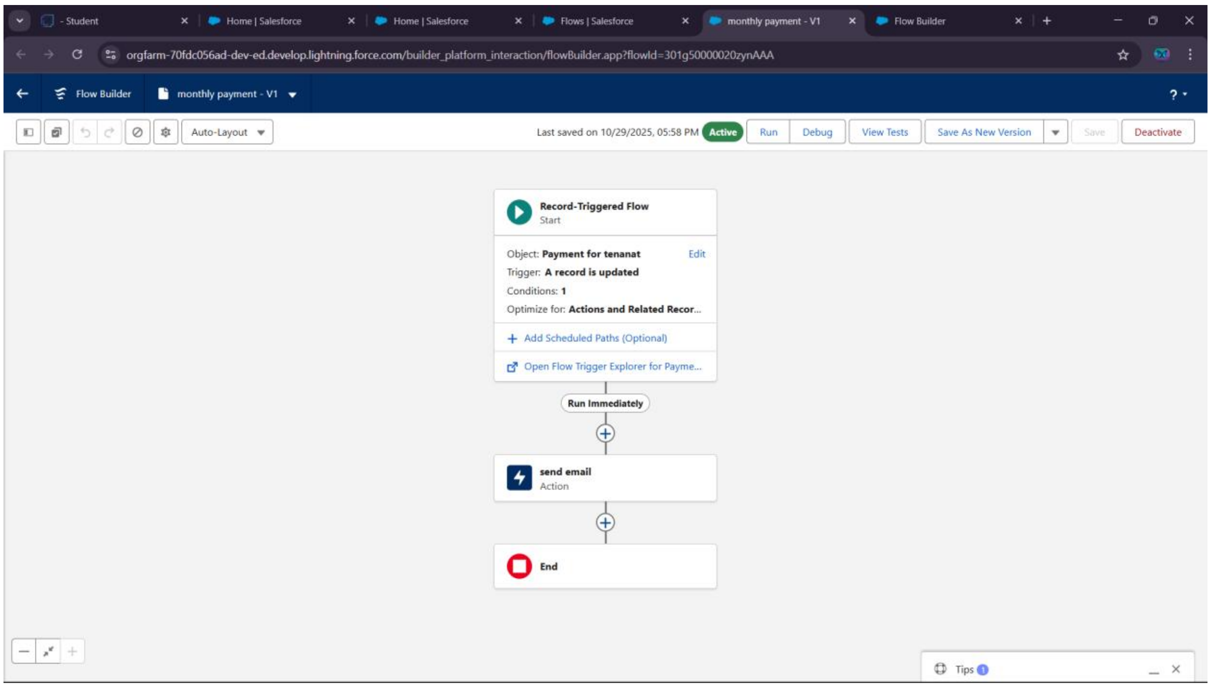
- View and manage your actions in the Actions Hub
- Connect to external systems with low-code connectors and with flow templates that contain connector actions
- Configure and manage robots and tasks with Robotic Process Automation
- Configure, build, and manage document-based automations with Intelligent Document Processing

If you don't see the app in the App Launcher, check that Enable the Automation Lightning App is selected in Process Automation Settings.

Flow Definitions
All Flows

50+ items • Sorted by Flow Label • Filtered by All flow definitions • Updated a few seconds ago

Flow Label	Process Type	Active	Tem...	Package State	Pack...	Last Mo...	Last Modified Date
Generate Payment Link	Screen Flow	✓	✓	Managed-Installed			
Inbound Cancel Appointment	Salesforce Scheduler Flow	✓	✓	Managed-Installed			
Inbound Modify Appointment	Salesforce Scheduler Flow	✓	✓	Managed-Installed			
Inbound New Appointment	Salesforce Scheduler Flow	✓	✓	Managed-Installed			
Inbound New Guest Appointment	Salesforce Scheduler Flow	✓	✓	Managed-Installed			
Messages Routed to Agents and Queues	Omni-Channel Flow	✓	✓	Managed-Installed			
Modify Guest Service Appointment	Salesforce Scheduler Flow	✓	✓	Managed-Installed			
monthly payment	Autolaunched Flow	✓		Unmanaged		Nishanthi S	10/29/2025, 5:28 AM
Orchestration flow for Recurrence Scheduler	Autolaunched Flow	✓	✓	Managed-Installed			
Outbound Modify Appointment	Salesforce Scheduler Flow	✓	✓	Managed-Installed			
Outbound New Appointment	Salesforce Scheduler Flow	✓	✓	Managed-Installed			
Pay for Service Appointments	Salesforce Scheduler Flow	✓	✓	Managed-Installed			


orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/builder_platform_interaction/flowBuilder.app?flowId=301g50000020zynAAA

Flow Builder | monthly payment - V1

Last saved on 10/29/2025, 05:58 PM Active

Record-Triggered Flow
Start

Object: **Payment for tenanat**
Trigger: **A record is updated**
Conditions: **1**
Optimize for: **Actions and Related Recor...**

+ Add Scheduled Paths (Optional)
Open Flow Trigger Explorer for Payme...

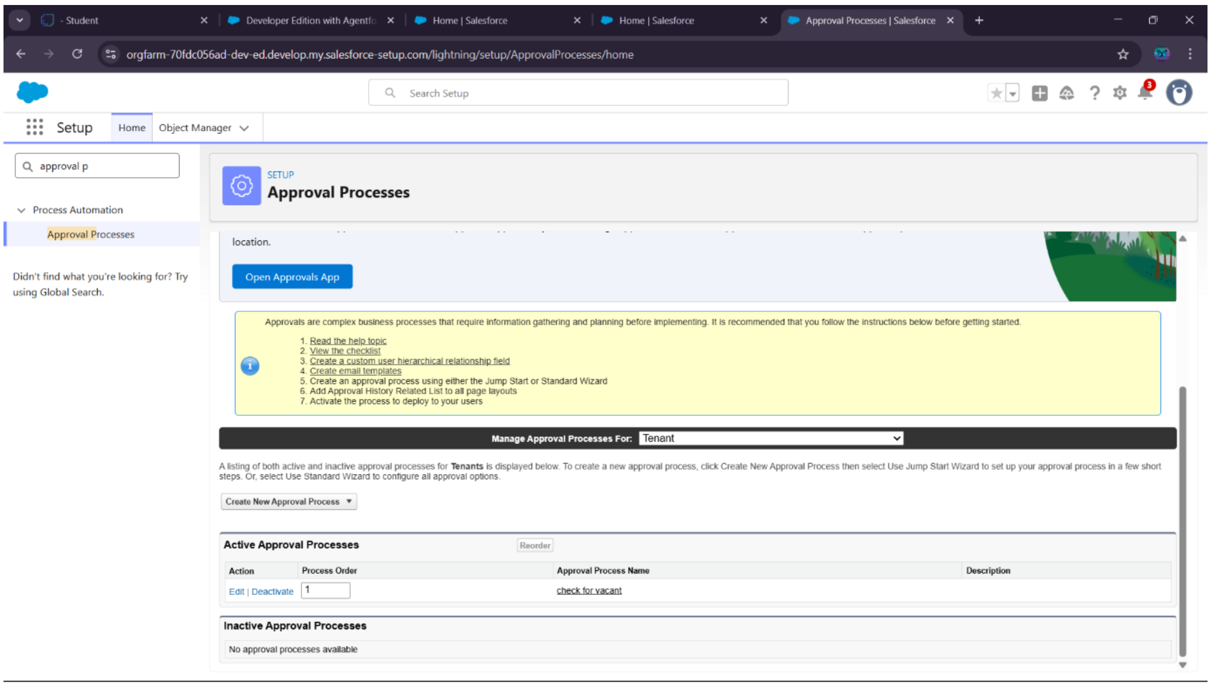
Run Immediately

send email
Action

End

➢ **Approval Processes:** Used for scenarios like lease renewal approval by a manager.

Milestone 8: Approval Process


orgfarm-70fdc056ad-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ApprovalProcesses/home

Setup | Home | Object Manager

approval p

Process Automation
Approval Processes

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

SETUP
Approval Processes

location.

Open Approvals App

Approvals are complex business processes that require information gathering and planning before implementing. It is recommended that you follow the instructions below before getting started.

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

Manage Approval Processes For: Tenant

A listing of both active and inactive approval processes for **Tenants** is displayed below. To create a new approval process, click Create New Approval Process then select Use Jump Start Wizard to set up your approval process in a few short steps. Or, select Use Standard Wizard to configure all approval options.

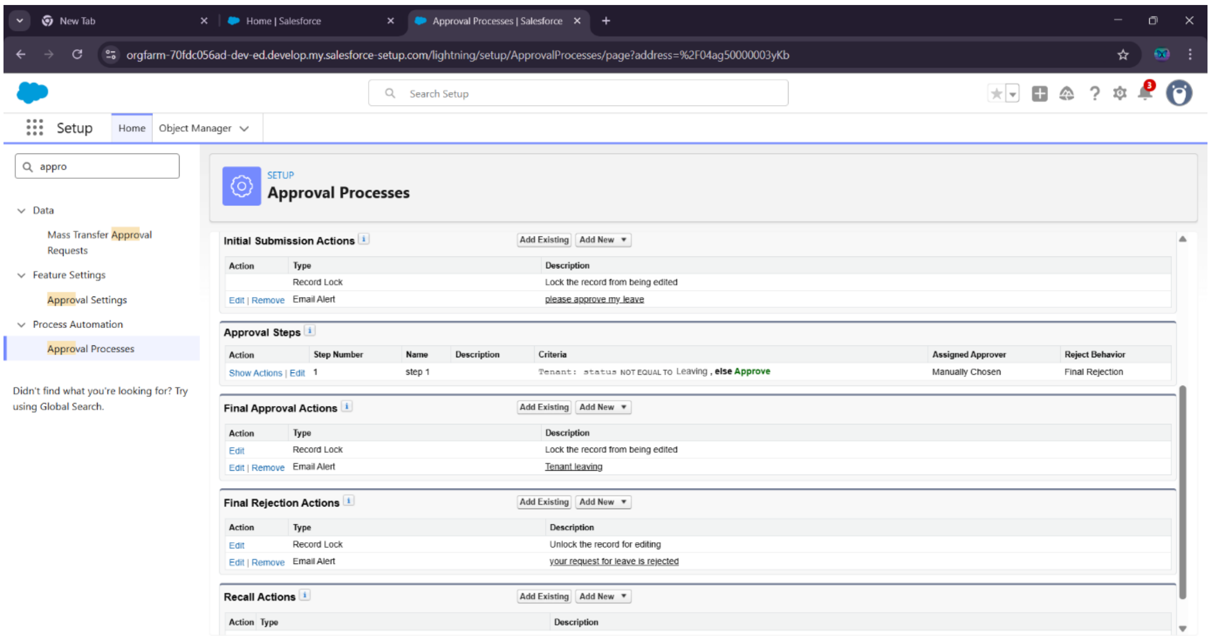
Create New Approval Process

Active Approval Processes

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

Inactive Approval Processes

No approval processes available


SETUP
Approval Processes

Initial Submission Actions

Action	Type	Description
	Record Lock	Lock the record from being edited
Edit | Remove	Email Alert	please approve my leave

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions | Edit	1	step 1		Tenant: status NOT EQUAL TO Leaving , else Approve	Manually Chosen	Final Rejection

Final Approval Actions

Action	Type	Description
Edit	Record Lock	Lock the record from being edited
Edit | Remove	Email Alert	Tenant leaving

Final Rejection Actions

Action	Type	Description
Edit	Record Lock	Unlock the record for editing
Edit | Remove	Email Alert	your request for leave is rejected

Recall Actions

Action	Type	Description

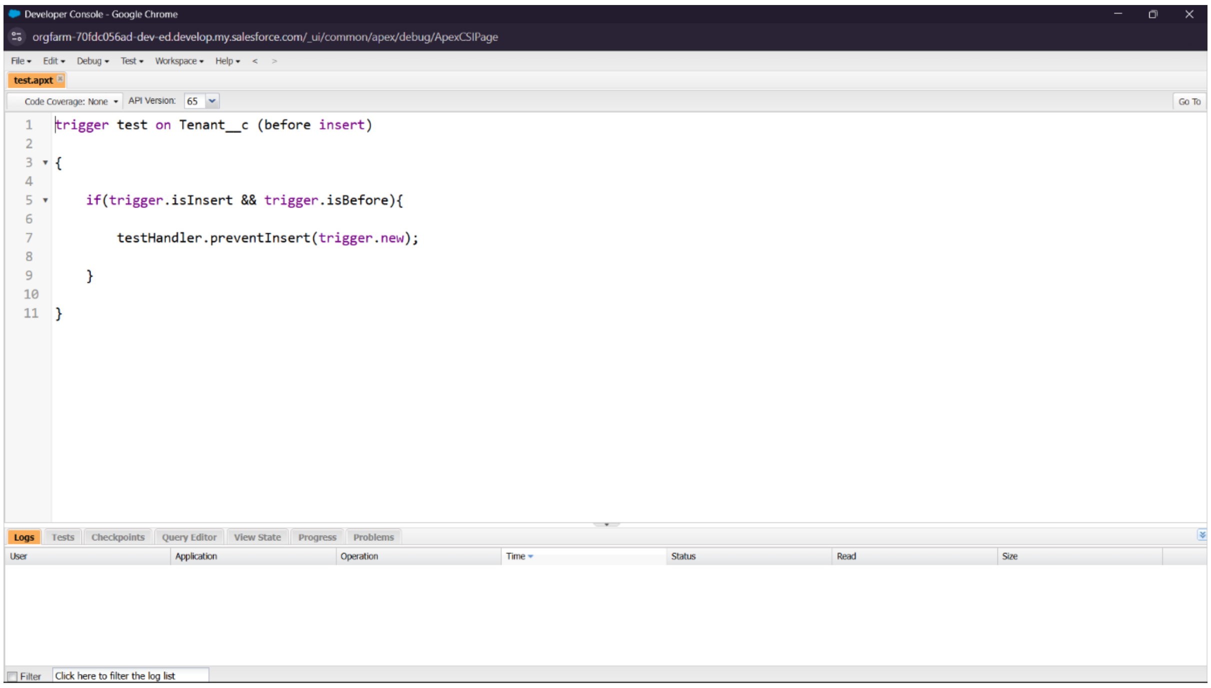
4.3 Apex Classes, Triggers, and Asynchronous Apex

Milestone 9: Apex Trigger

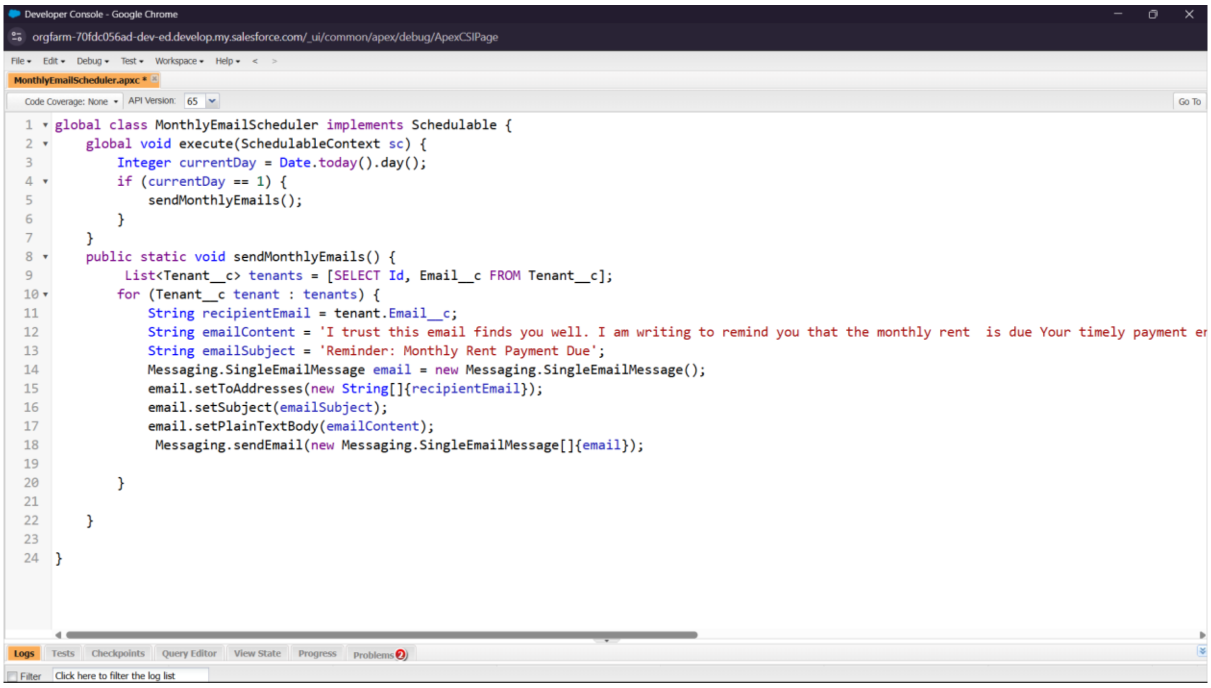
When point-and-click tools are not enough, Apex programming is used to implement custom logic.

4.3.1 Apex Classes:

These are reusable blocks of code used for custom backend logic.



```
trigger test on Tenant__c (before insert)

{

    if(trigger.isInsert && trigger.isBefore){

        testHandler.preventInsert(trigger.new);

    }

}
```


Developer Console - Google Chrome
orgfarm-70fdc056ad-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
MonthlyEmailScheduler.apxc *
Code Coverage: None
API Version: 65
Go To
Logs Tests Checkpoints Query Editor View State Progress Problems
Filter
Click here to filter the log list

```
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 er
            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});

        }

    }

}
```

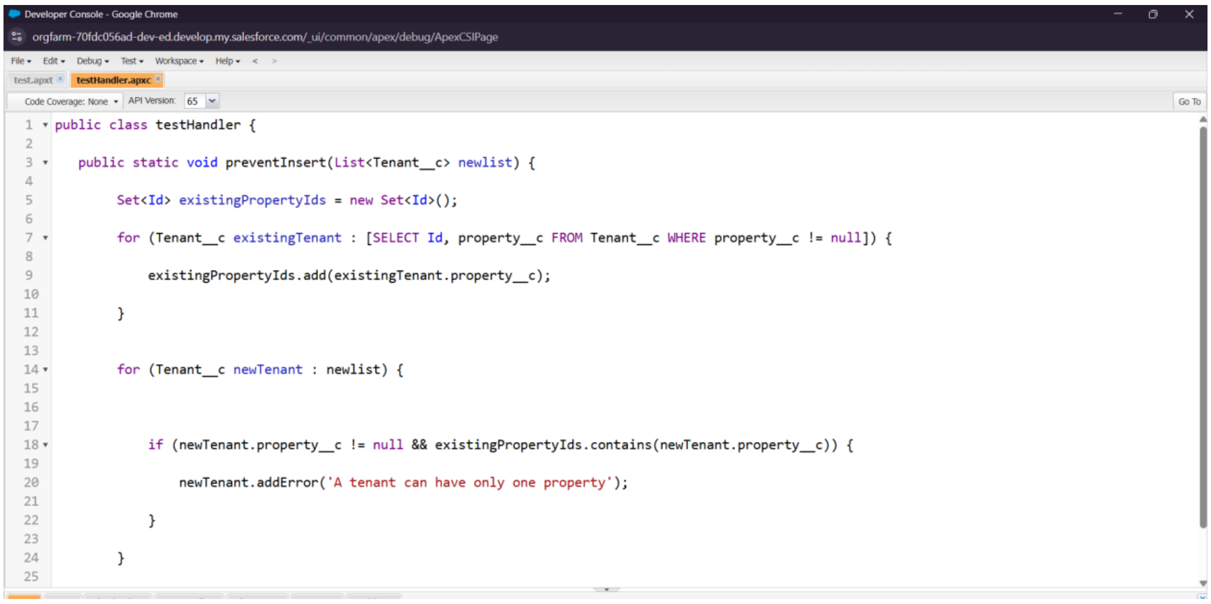
4.3.2 Apex Triggers:

Triggers are used to perform actions automatically when a record is inserted, updated, or deleted.

Example:

➢ On creation of a Lease record, auto-generate a related Payment schedule.

➢ On Payment update, update Lease status if fully paid.


Developer Console - Google Chrome
orgfarm-70fdc056ad-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
test.apxt
testHandler.apxc
Code Coverage: None
API Version: 65
Go To

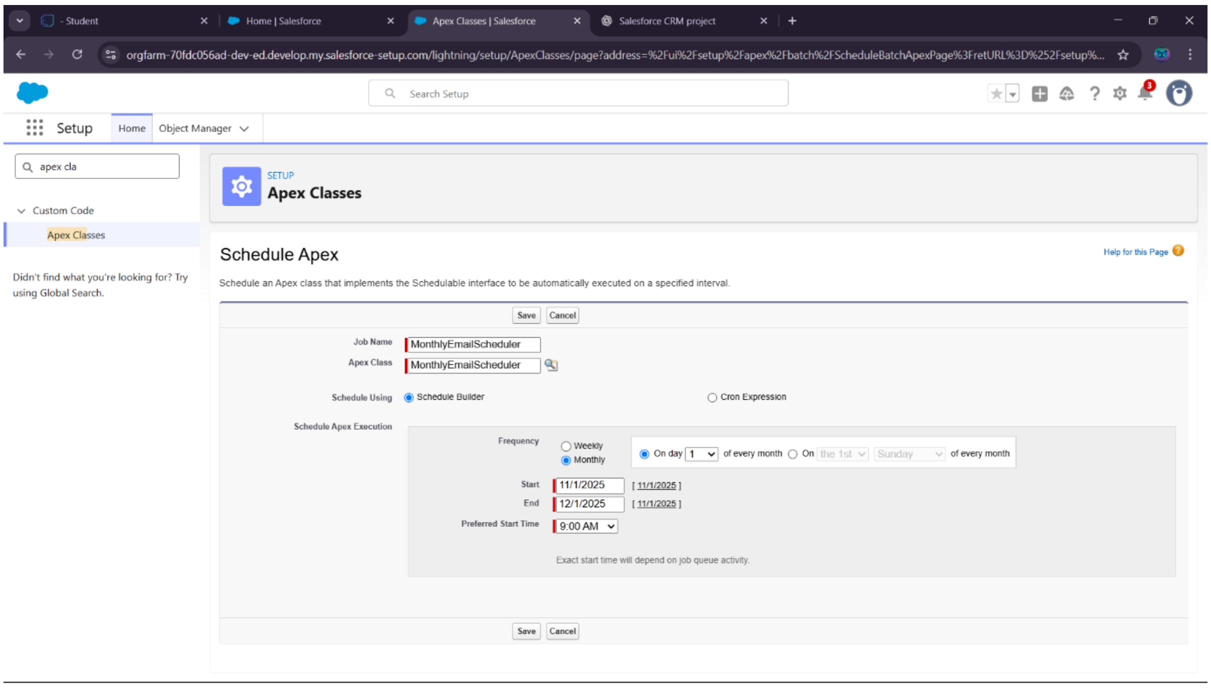
```
public class testHandler {

   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');

            }

        }

```

4.3.3 Asynchronous Apex:

Milestone 11: Schedule Class

Used when tasks take longer to run or need to happen in the background:

➢ **Scheduled Apex:** To check daily for upcoming lease expirations.



Schedule Apex

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

Job Name: MonthlyEmailScheduler
Apex Class: MonthlyEmailScheduler
Schedule Using: [✓ Schedule Builder] Cron Expression
Schedule Apex Execution
Frequency: Weekly [✓ Monthly]
[✓ On day 1 of every month] On the 1st Sunday of every month
Start: 11/1/2025 [11/1/2025]
End: 12/1/2025 [11/1/2025]
Preferred Start Time: 9:00 AM

Exact start time will depend on job queue activity.

5. UI/UX DEVELOPMENT & CUSTOMIZATION

This phase focuses on designing and customizing the user interface to provide a smooth and efficient experience for both property managers and tenants. Salesforce Lightning Experience tools are used to build a clean, user-friendly, and role-specific UI.

5.1 Lightning App Setup through App Manager

Milestone 4: The Lightning App

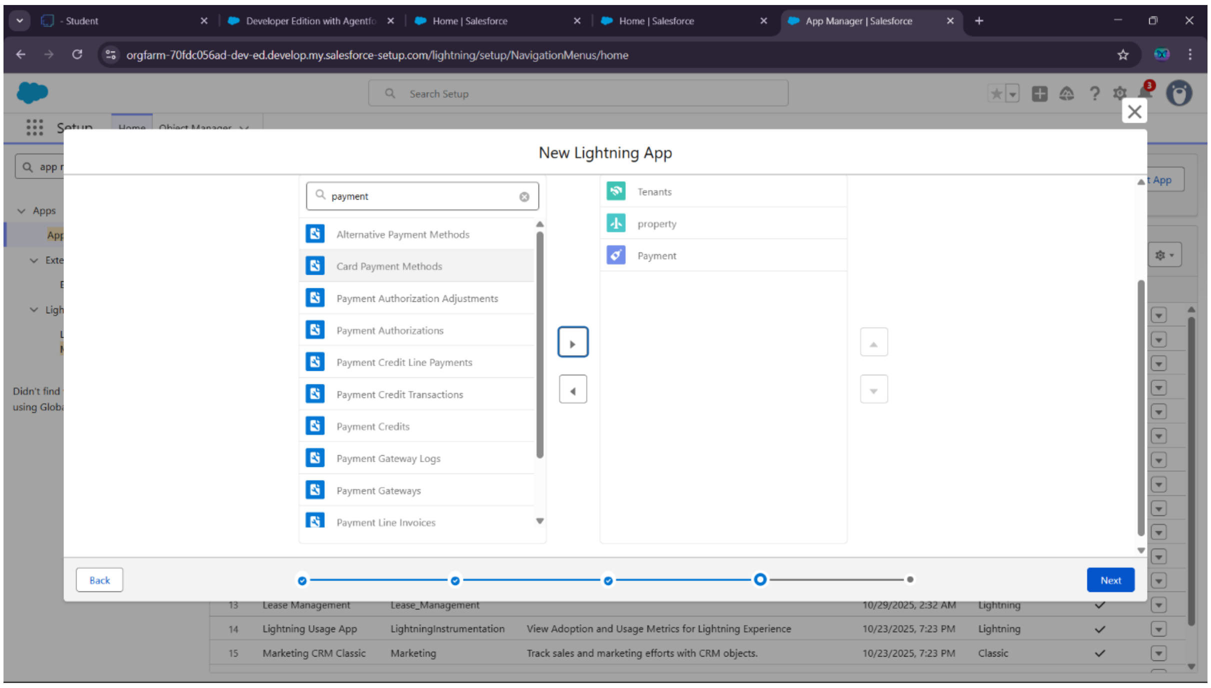
Using App Manager, a custom **"Lease Management" Lightning App** is created with a branded name, logo, and navigation items such as:

➢ Tenants

➢ Properties

➢ Payments

This setup ensures quick access to all key objects in one place, improving user workflow.

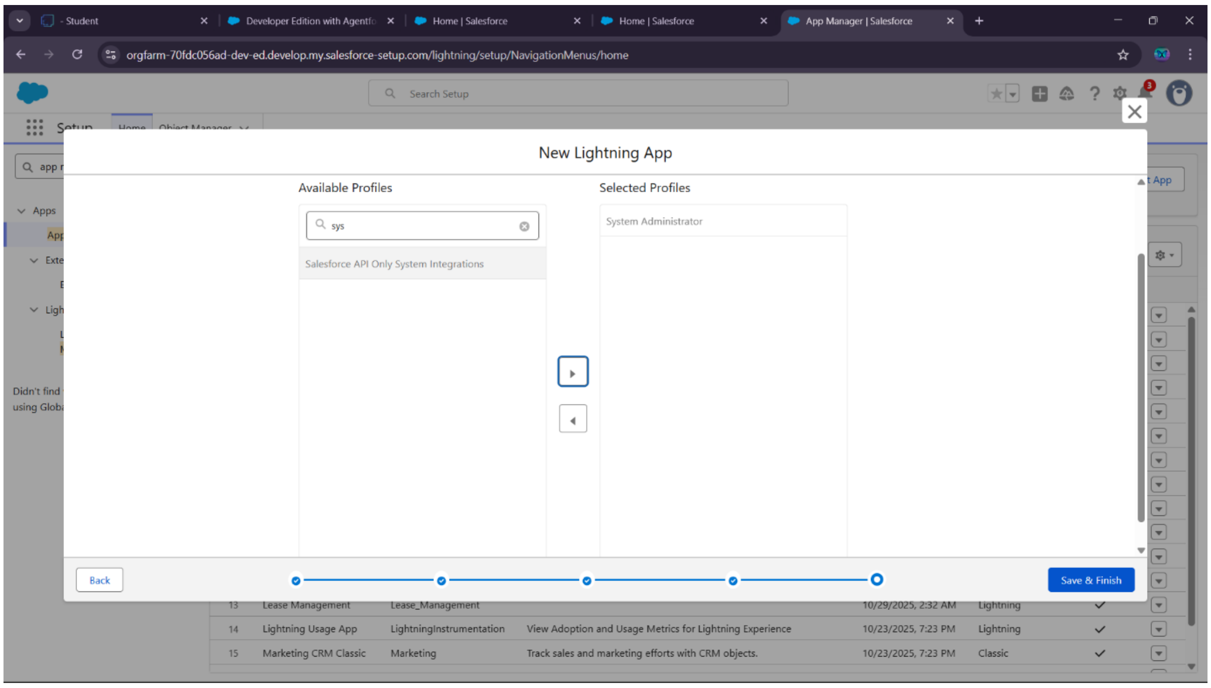

New Lightning App

payment

Alternative Payment Methods
Card Payment Methods
Payment Authorization Adjustments
Payment Authorizations
Payment Credit Line Payments
Payment Credit Transactions
Payment Credits
Payment Gateway Logs
Payment Gateways
Payment Line Invoices

Tenants
property
Payment

The next step involves in selecting the system administrator as the default user for the user profiles.


New Lightning App

Available Profiles

sys

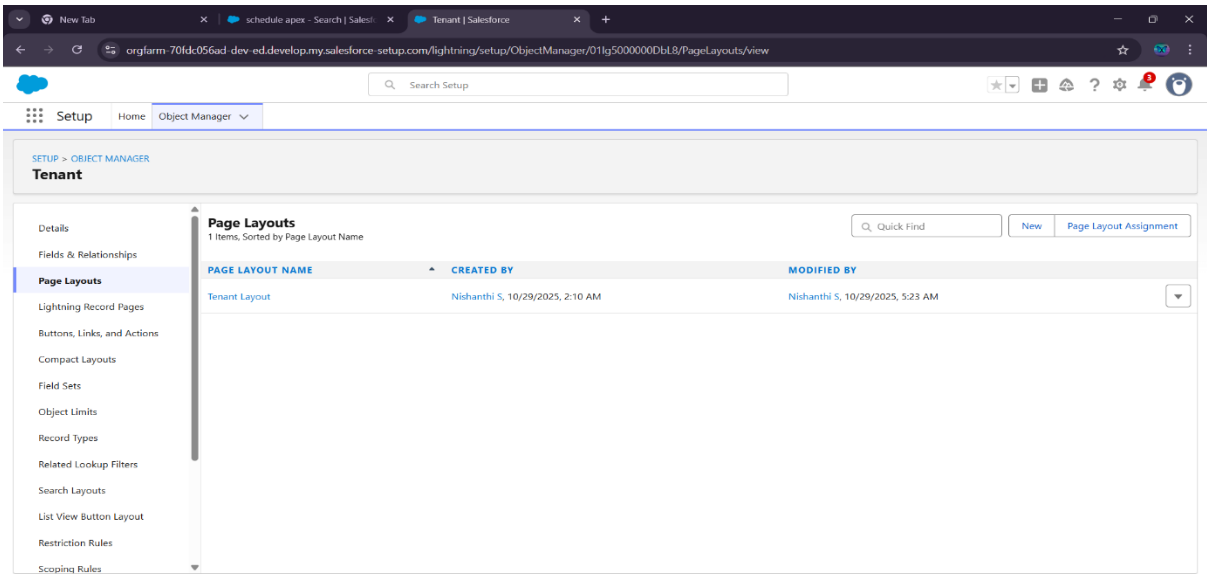
Salesforce API Only System Integrations

Selected Profiles

System Administrator

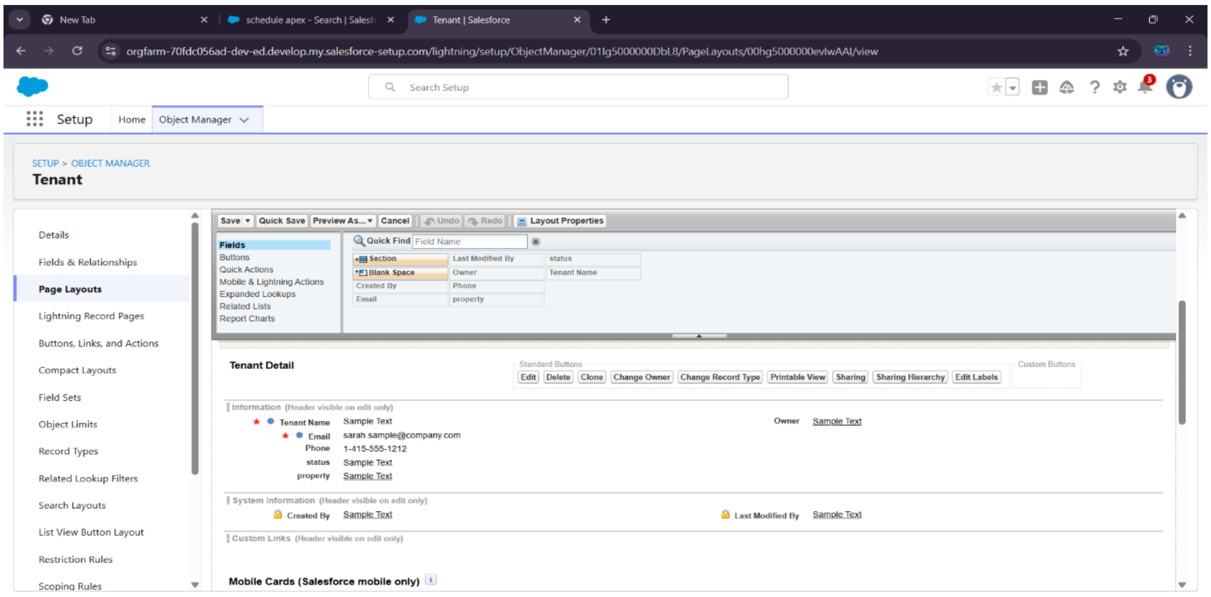
5.2 Page Layouts and Dynamic Forms

Page Layouts are customized for Tenant object to display only relevant fields for ach user type.



Page Layouts
1 Items, Sorted by Page Layout Name

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Tenant Layout	Nishanthi S, 10/29/2025, 2:10 AM	Nishanthi S, 10/29/2025, 5:23 AM



Tenant Detail

Information (Header visible on edit only)

Tenant Name	Sample Text	Owner	Sample Text
Email	sarah.sample@company.com		
Phone	1-415-555-1212		
status	Sample Text		
property	Sample Text		

System Information (Header visible on edit only)

Created By	Sample Text	Last Modified By	Sample Text

Custom Links (Header visible on edit only)

Mobile Cards (Salesforce mobile only)

This ensures the interface stays clean and relevant for users, reducing clutter and confusion.

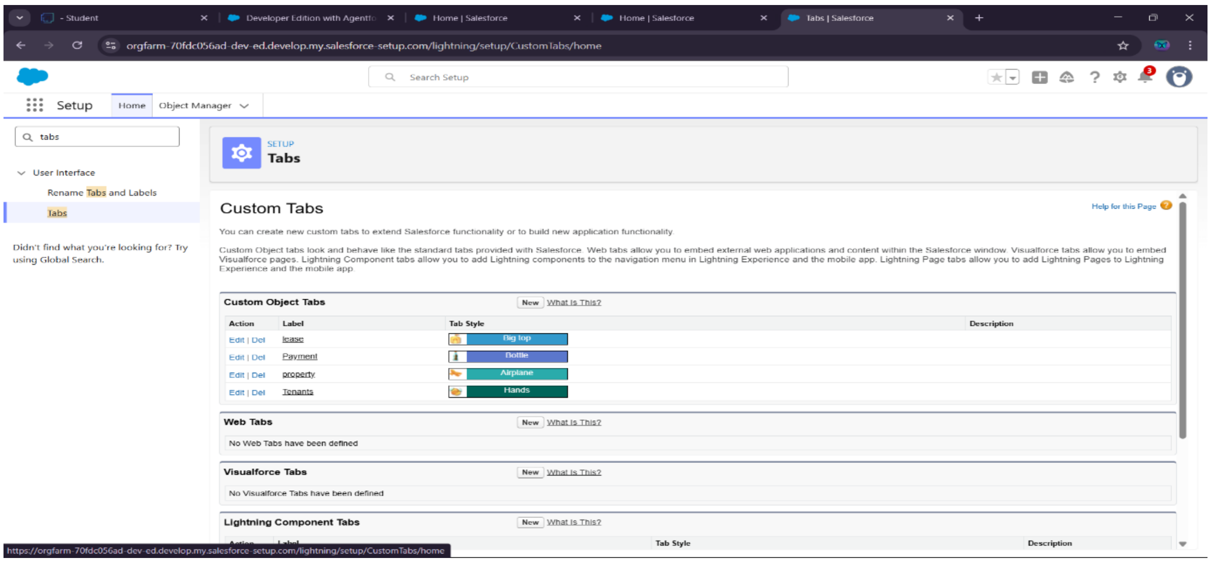
5.3 Lightning Pages

Milestone 3: Tabs

Lightning Record Pages were customized for each object to improve how data is displayed to users:

• **Use of Tabs:** Tabs were used to organize related information clearly within the record page, ensuring a clean and structured layout.

This setup creates a simple, role-friendly interface, making navigation easier and enhancing the overall user experience.


- Student
Developer Edition with Agentfo
Home | Salesforce
Home | Salesforce
Tabs | Salesforce
orgfarm-70fdc056ad-dev-ed.develop.my.salesforce-setup.com/lightning/setup/CustomTabs/home
Search Setup
Setup
Home
Object Manager
tabs
User Interface
Rename Tabs and Labels
Tabs
Didn't find what you're looking for? Try using Global Search.
SETUP
Tabs
Help for this Page

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs — New — What Is This?

Action	Label	Tab Style	Description
Edit | Del	lease	Big top	
Edit | Del	Payment	Bottle	
Edit | Del	property	Airplane	
Edit | Del	Tenants	Hands	

Web Tabs — New — What Is This?

No Web Tabs have been defined

Visualforce Tabs — New — What Is This?

No Visualforce Tabs have been defined

Lightning Component Tabs — New — What Is This?

Action	Label	Tab Style	Description

https://orgfarm-70fdc056ad-dev-ed.develop.my.salesforce-setup.com/lightning/setup/CustomTabs/home

<u>PHASE 4- PROJECT DEVELOPMENT</u>

6. DATAMIGRATION, TESTING & SECURITY

This phase covers the essential activities of migrating lease-related data into Salesforce, thorough testing of the developed features, and ensuring security measures are properly implemented. The objective is to guarantee data accuracy, system reliability, and protection of sensitive information before the system goes live.

6.1 Testing

Testing was performed to validate the functionality and accuracy of all implemented automation and business logic in the system. This phase ensures that the Lease Management System performs as expected under real-world scenarios.

6.1.1 Apex Trigger Testing

A custom Apex trigger was developed to enforce the rule that each property can be assigned to only one tenant at a time.

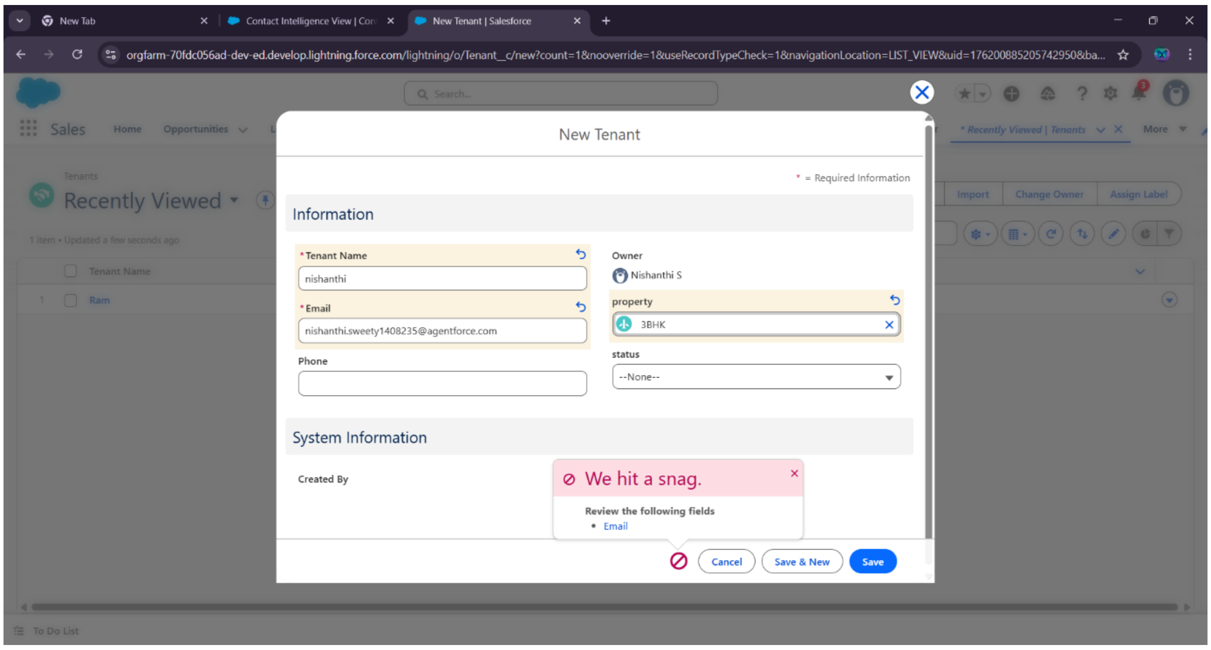
To test this:

• A test class was created simulating insertion of tenant records.

• It confirmed successful creation when a property was free.

• It verified the trigger correctly throws an error if a property is already assigned, preventing data inconsistencies.

Test logs and results demonstrated the trigger’s effectiveness in maintaining data integrity.


New Tab
Contact Intelligence View | Con
New Tenant | Salesforce
orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/o/Tenant__c/new?count=1&nooverride=1&useRecordTypeCheck=1&navigationLocation=LIST_VIEW&uid=176200885205742950&ba...
Sales
Home
Opportunities
Recently Viewed
Import
Change Owner
Assign Label
To Do List

New Tenant

* = Required Information

Information

*Tenant Name: nishanthi

*Email: nishanthi.sweety1408235@agentforce.com

Phone:

Owner: Nishanthi S

property: 3BHK

status: --None--

System Information

Created By

We hit a snag.

Review the following fields

- Email

Cancel | Save & New | Save

6.1.2 Approval Process Testing

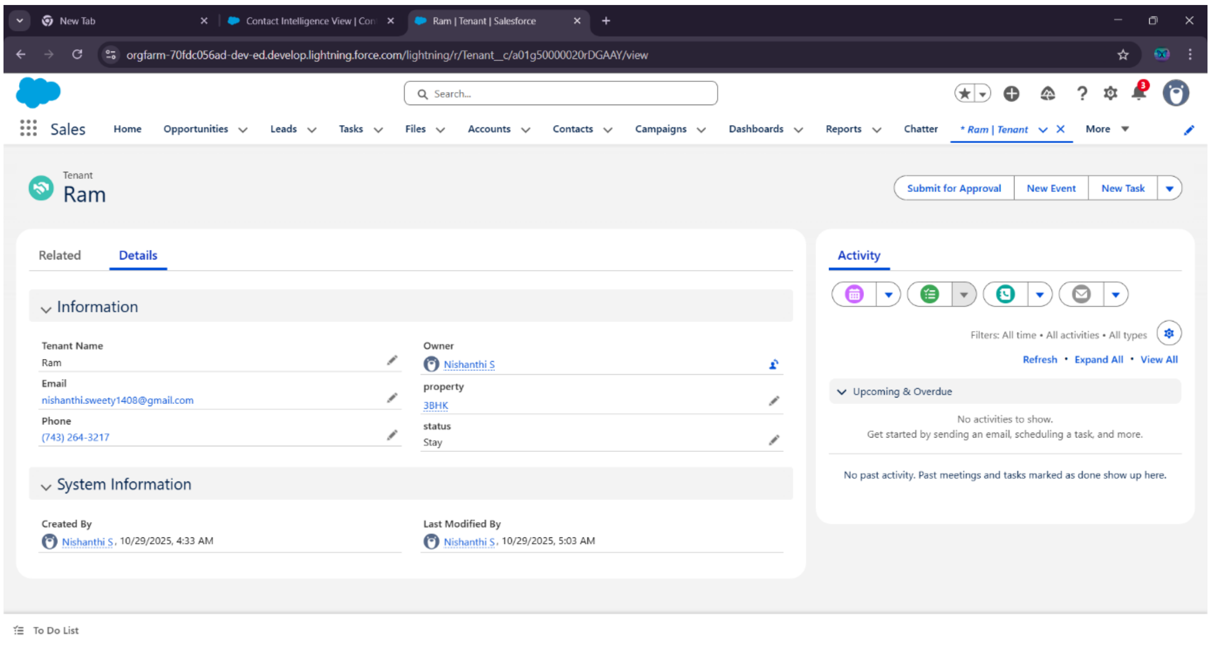
The tenant leave request approval workflow was tested through multiple scenarios:

• Submission of leave requests by tenants triggered the approval process.

• Approvers received real-time notifications and emails.

• Both approval and rejection paths were tested to ensure that tenant status updated accordingly, and the proper email templates were sent to notify tenants.

Screenshots of submission forms, approval screens, and notification emails are documented as evidence of successful workflow execution.


New Tab | Contact Intelligence View | Con | Ram | Tenant | Salesforce

orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/r/Tenant__c/a01g50000020rDGAAY/view

Sales | Home | Opportunities | Leads | Tasks | Files | Accounts | Contacts | Campaigns | Dashboards | Reports | Chatter | * Ram | Tenant | More

Tenant
Ram

Submit for Approval | New Event | New Task

Related | Details

Information

Tenant Name: Ram	Owner: Nishanthi S
Email: nishanthi.sweety1408@gmail.com	property: 3BHK
Phone: (743) 264-3217	status: Stay

System Information

Created By: Nishanthi S, 10/29/2025, 4:33 AM	Last Modified By: Nishanthi S, 10/29/2025, 5:03 AM

Activity

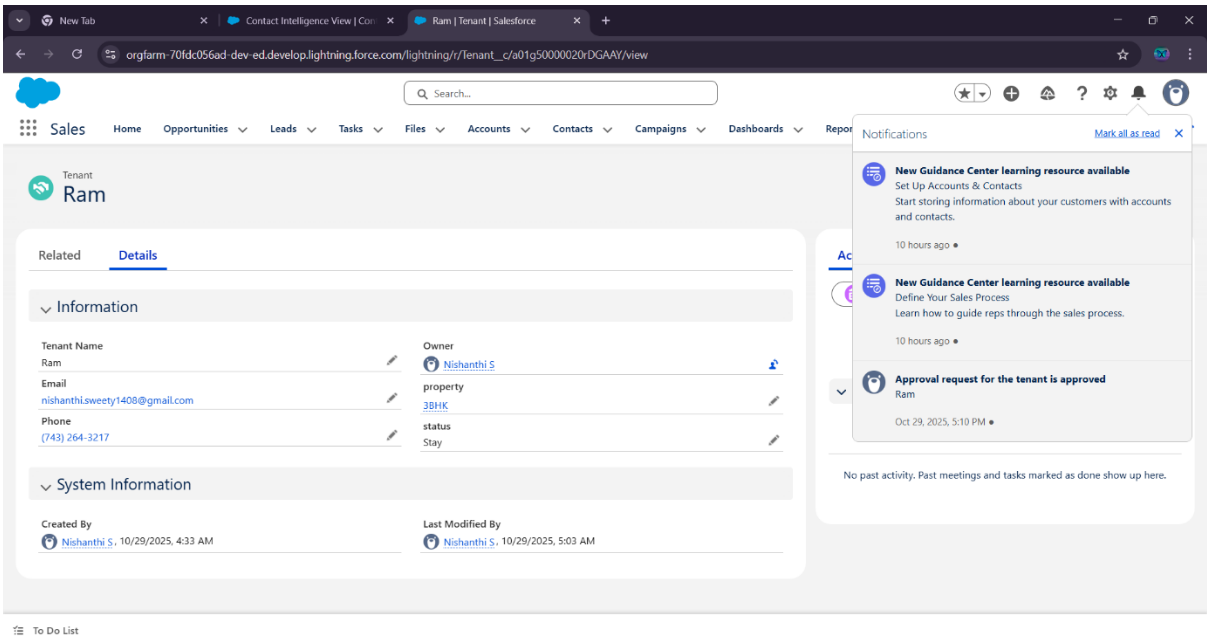
Filters: All time • All activities • All types
Refresh • Expand All • View All

Upcoming & Overdue

No activities to show.
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

To Do List


New Tab | Contact Intelligence View | Con | Ram | Tenant | Salesforce

orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/r/Tenant__c/a01g50000020rDGAAY/view

Sales | Home | Opportunities | Leads | Tasks | Files | Accounts | Contacts | Campaigns | Dashboards | Repor

Tenant
Ram

Related | Details

Information

Tenant Name: Ram	Owner: Nishanthi S
Email: nishanthi.sweety1408@gmail.com	property: 3BHK
Phone: (743) 264-3217	status: Stay

System Information

Created By: Nishanthi S, 10/29/2025, 4:33 AM	Last Modified By: Nishanthi S, 10/29/2025, 5:03 AM

Notifications — Mark all as read

New Guidance Center learning resource available
Set Up Accounts & Contacts
Start storing information about your customers with accounts and contacts.
10 hours ago

New Guidance Center learning resource available
Define Your Sales Process
Learn how to guide reps through the sales process.
10 hours ago

Approval request for the tenant is approved
Ram
Oct 29, 2025, 5:10 PM

No past activity. Past meetings and tasks marked as done show up here.

To Do List

6.1.3 Flow Testing

The system includes a record-triggered flow that sends payment confirmation emails automatically when a tenant's payment status changes to "paid."

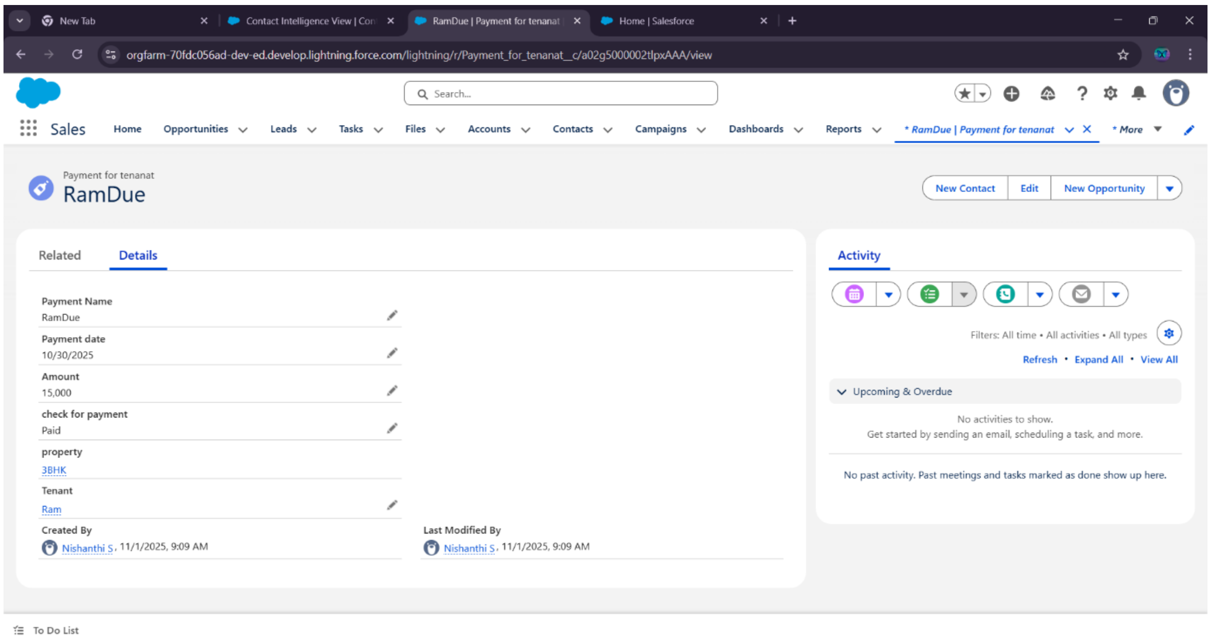
Testing involved:

• Updating payment records to "paid" status.

• Monitoring flow execution in Salesforce.

• Confirming that tenants received correctly formatted confirmation emails.

This automated communication improves tenant engagement and reduces manual workload.



orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/r/Payment_for_tenanat__c/a02g5000002tlpxAAA/view

Payment for tenanat

RamDue

Payment Name	RamDue
Payment date	10/30/2025
Amount	15,000
check for payment	Paid
property	3BHK
Tenant	Ram
Created By	Nishanthi S, 11/1/2025, 9:09 AM
Last Modified By	Nishanthi S, 11/1/2025, 9:09 AM

Activity

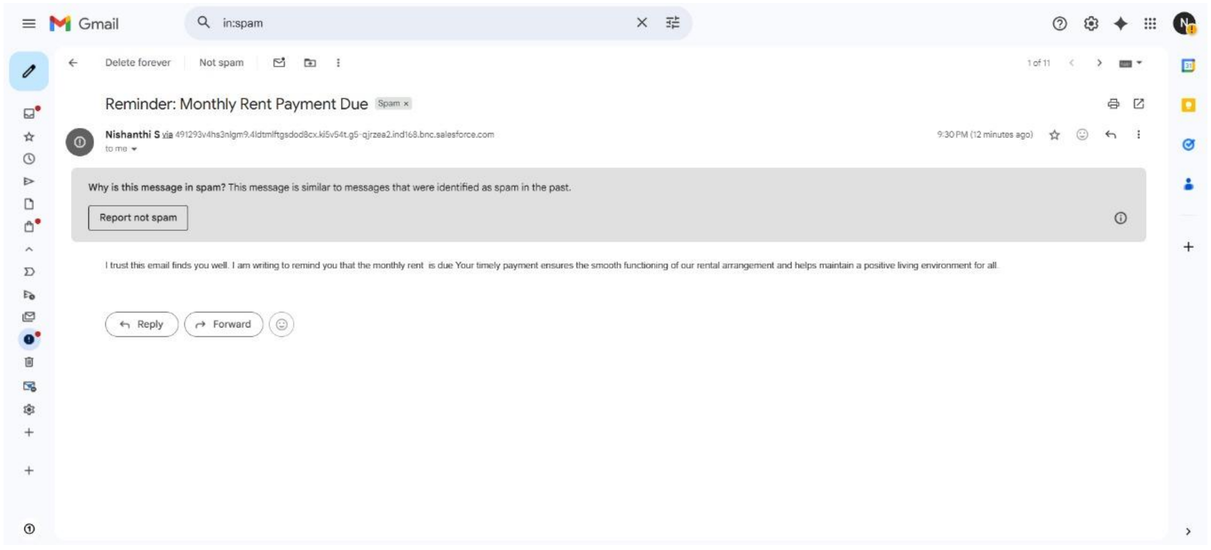
Filters: All time • All activities • All types

Upcoming & Overdue

No activities to show.
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

A confirmation mail received to the tenant about the payment of the monthly rent.



Reminder: Monthly Rent Payment Due

Nishanthi S via 491293v4hs3nlgm9.4ldtmlftgsdod8cx.ki5v54t.g5-qjrzea2.ind168.bnc.salesforce.com
to me

9:30 PM (12 minutes ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

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.

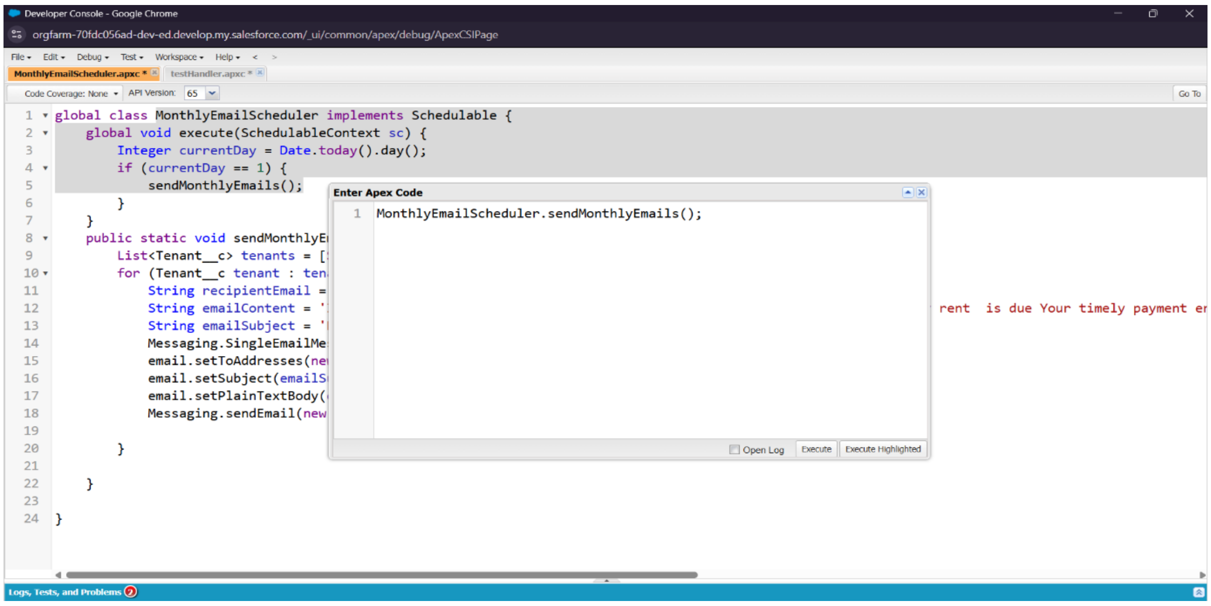
6.1.4 Scheduled Apex Testing

Milestone 7: Email Templates

To ensure timely reminders for rent payments, a scheduled Apex class was developed and tested. The scheduler runs monthly on the 1st day to send payment reminder emails. Testing involved:

- Manually triggering the scheduled job to simulate its execution.
- Verifying emails were dispatched to all tenants with pending payments.

Test documentation includes logs from the scheduler and screenshots of sent emails.

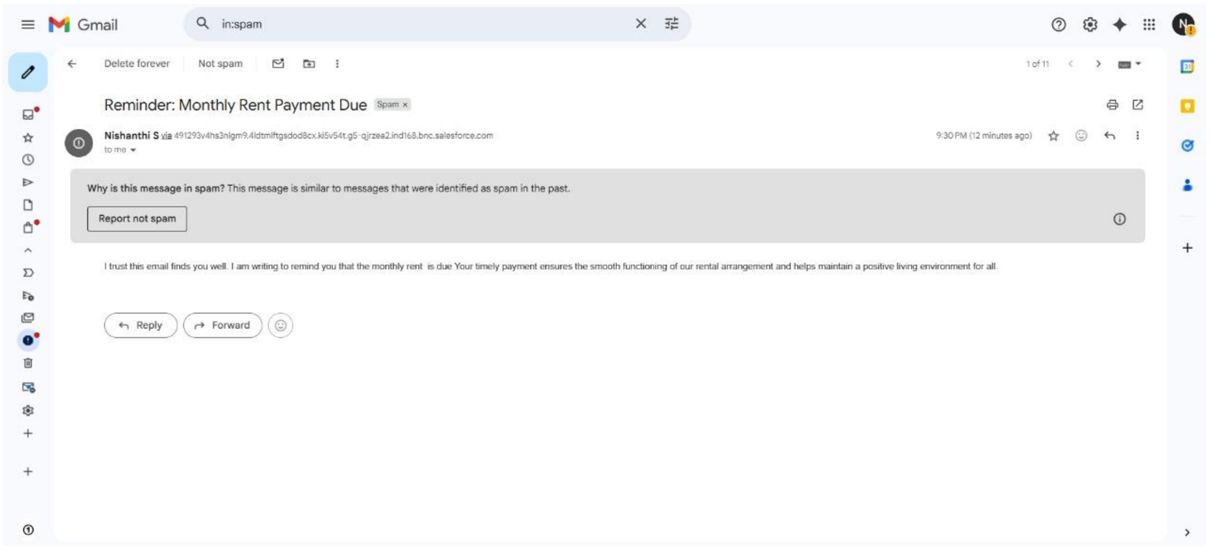


```
global class MonthlyEmailScheduler implements Schedulable {
    global void execute(SchedulableContext sc) {
        Integer currentDay = Date.today().day();
        if (currentDay == 1) {
            sendMonthlyEmails();
        }
    }
    public static void sendMonthlyE
        List<Tenant__c> tenants = [
        for (Tenant__c tenant : ten
            String recipientEmail =
            String emailContent = '                                                    rent  is due Your timely payment er
            String emailSubject = '
            Messaging.SingleEmailMe
            email.setToAddresses(ne
            email.setSubject(emailS
            email.setPlainTextBody(
            Messaging.sendEmail(new

        }

    }

}
```

Enter Apex Code

```
MonthlyEmailScheduler.sendMonthlyEmails();
```

The execution of the MonthlyEmailSchedular class.



Reminder: Monthly Rent Payment Due

Nishanthi S via 491293v4hs3nlgm9.4ldtmlftgsdod8cx.ki5v54t.g5-qjrzea2.ind168.bnc.salesforce.com
to me

9:30 PM (12 minutes ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

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.

6.2 Documentation and Evidence

• Detailed test cases were created covering all implemented features.

• Input data, execution steps, and expected outcomes were recorded for each test.

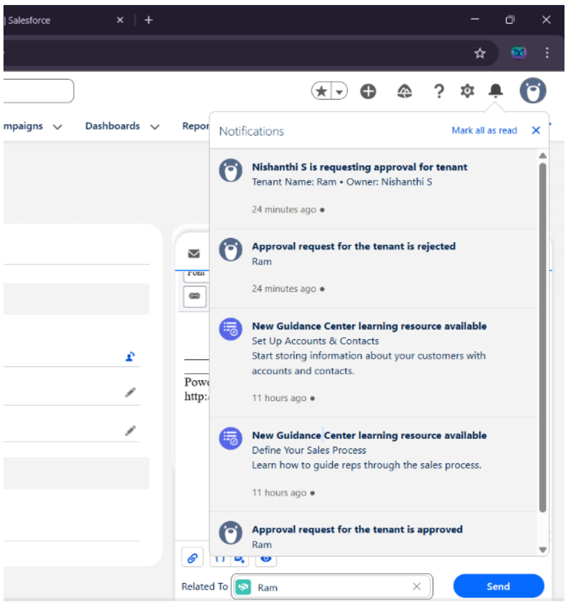
• Screenshots captured actual results to verify correct functionality.

• Documentation ensures transparency and aids future maintenance.

• Provides confidence in system readiness for deployment.

The execution of the Apex class MonthlyEmailSchedular. The following figure shows the execution logs of the Apex class.

The notifications are received to the user.



Notifications

Mark all as read

Nishanthi S is requesting approval for tenant
Tenant Name: Ram • Owner: Nishanthi S
24 minutes ago •

Approval request for the tenant is rejected
Ram
24 minutes ago •

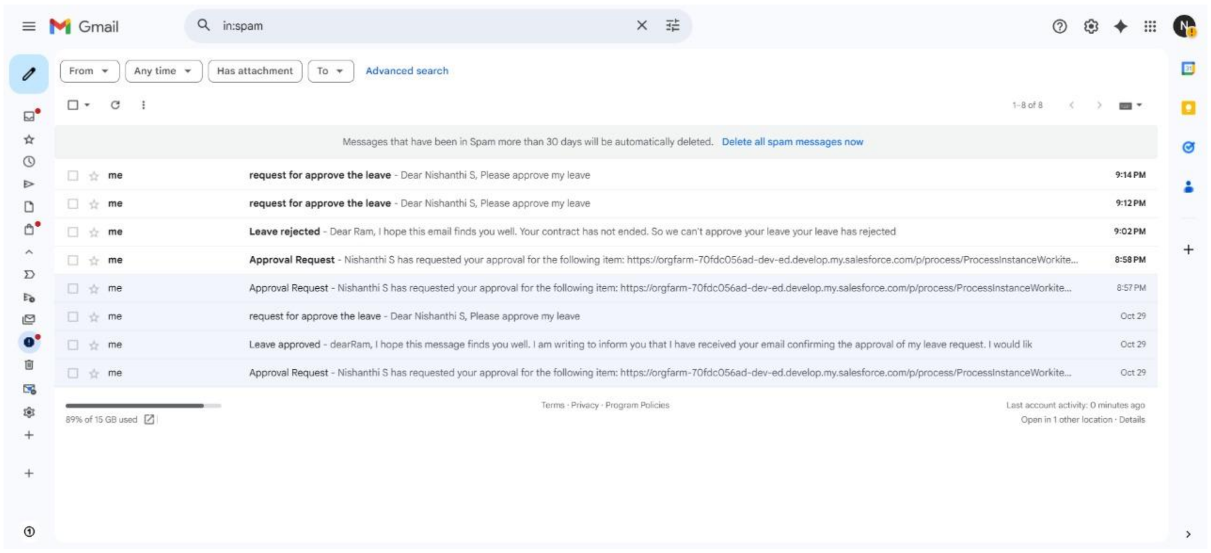
New Guidance Center learning resource available
Set Up Accounts & Contacts
Start storing information about your customers with accounts and contacts.
11 hours ago •

New Guidance Center learning resource available
Define Your Sales Process
Learn how to guide reps through the sales process.
11 hours ago •

Approval request for the tenant is approved
Ram

Related To Ram
Send

Emails received by the tenant sent by the user or owner through automated process.



Gmail — in:spam

1-8 of 8

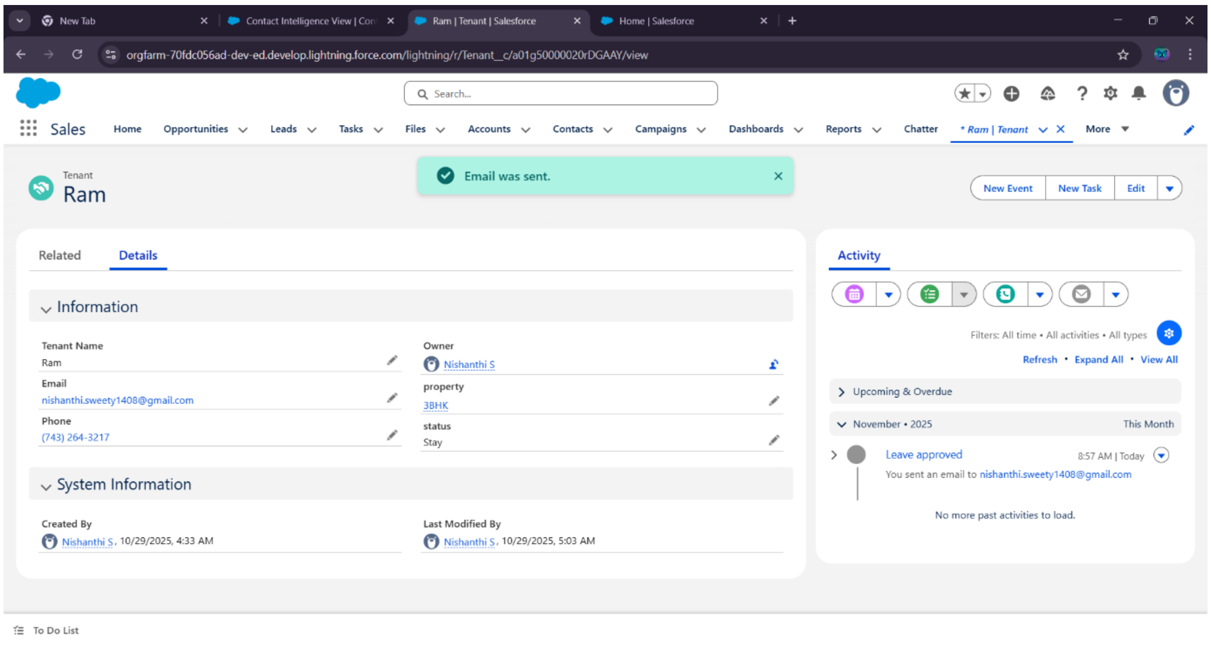
Messages that have been in Spam more than 30 days will be automatically deleted. Delete all spam messages now

me	**request for approve the leave** - Dear Nishanthi S, Please approve my leave	9:14 PM
me	**request for approve the leave** - Dear Nishanthi S, Please approve my leave	9:12 PM
me	**Leave rejected** - Dear Ram, I hope this email finds you well. Your contract has not ended. So we can't approve your leave your leave has rejected	9:02 PM
me	**Approval Request** - Nishanthi S has requested your approval for the following item: https://orgfarm-70fdc056ad-dev-ed.develop.my.salesforce.com/p/process/ProcessInstanceWorkite...	8:58 PM
me	Approval Request - Nishanthi S has requested your approval for the following item: https://orgfarm-70fdc056ad-dev-ed.develop.my.salesforce.com/p/process/ProcessInstanceWorkite...	8:57 PM
me	request for approve the leave - Dear Nishanthi S, Please approve my leave	Oct 29
me	Leave approved - dearRam, 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 lik	Oct 29
me	Approval Request - Nishanthi S has requested your approval for the following item: https://orgfarm-70fdc056ad-dev-ed.develop.my.salesforce.com/p/process/ProcessInstanceWorkite...	Oct 29

89% of 15 GB used

Terms · Privacy · Program Policies

Last account activity: 0 minutes ago
Open in 1 other location · Details


New Tab | Contact Intelligence View | Con | Ram | Tenant | Salesforce | Home | Salesforce

orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/r/Tenant__c/a01g50000020rDGAAY/view

Sales | Home | Opportunities | Leads | Tasks | Files | Accounts | Contacts | Campaigns | Dashboards | Reports | Chatter | * Ram | Tenant | More

Email was sent.

Tenant
Ram

New Event | New Task | Edit

Related | Details

Information

Tenant Name: Ram
Email: nishanthi.sweety1408@gmail.com
Phone: (743) 264-3217
Owner: Nishanthi S
property: 3BHK
status: Stay

System Information

Created By: Nishanthi S, 10/29/2025, 4:33 AM
Last Modified By: Nishanthi S, 10/29/2025, 5:03 AM

Activity

Filters: All time • All activities • All types
Refresh • Expand All • View All

Upcoming & Overdue

November • 2025 — This Month

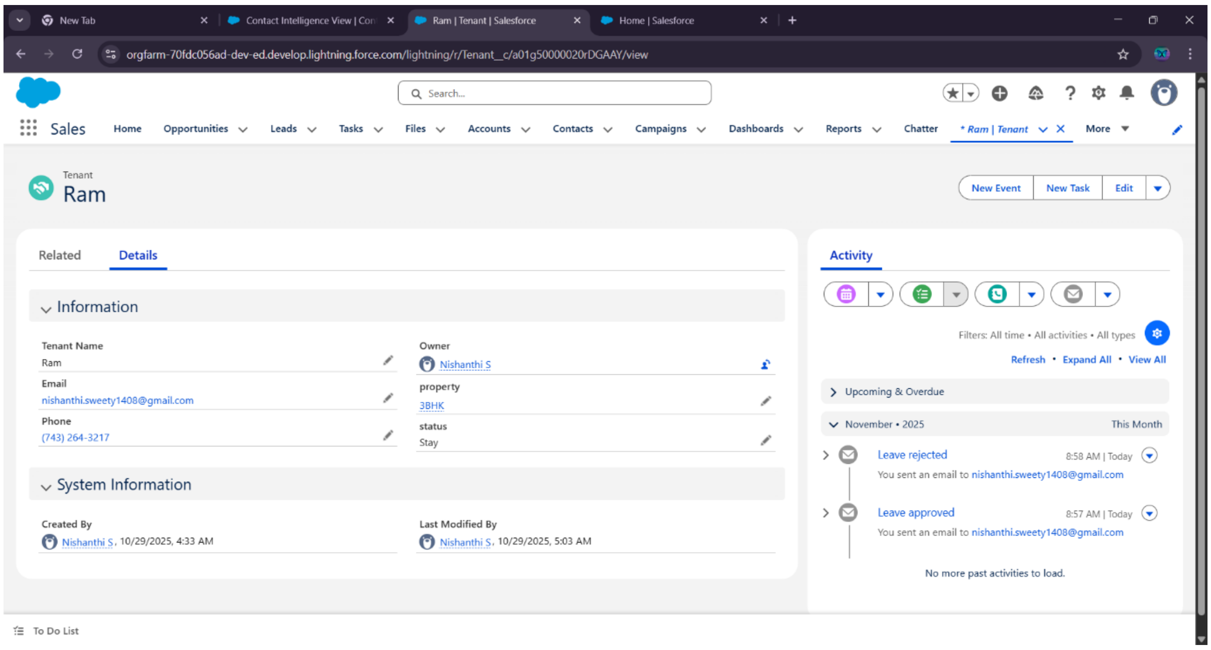
Leave approved — 8:57 AM | Today
You sent an email to nishanthi.sweety1408@gmail.com

No more past activities to load.

To Do List

The above figure determines the approval page for the owner. The owner can approve, reject or reassign the tenant request.

The approver can accept or reject the user request.


New Tab | Contact Intelligence View | Con | Ram | Tenant | Salesforce | Home | Salesforce

orgfarm-70fdc056ad-dev-ed.develop.lightning.force.com/lightning/r/Tenant__c/a01g50000020rDGAAY/view

Sales | Home | Opportunities | Leads | Tasks | Files | Accounts | Contacts | Campaigns | Dashboards | Reports | Chatter | * Ram | Tenant | More

Tenant
Ram

New Event | New Task | Edit

Related | Details

Information

Tenant Name: Ram
Email: nishanthi.sweety1408@gmail.com
Phone: (743) 264-3217
Owner: Nishanthi S
property: 3BHK
status: Stay

System Information

Created By: Nishanthi S, 10/29/2025, 4:33 AM
Last Modified By: Nishanthi S, 10/29/2025, 5:03 AM

Activity

Filters: All time • All activities • All types
Refresh • Expand All • View All

Upcoming & Overdue

November • 2025 — This Month

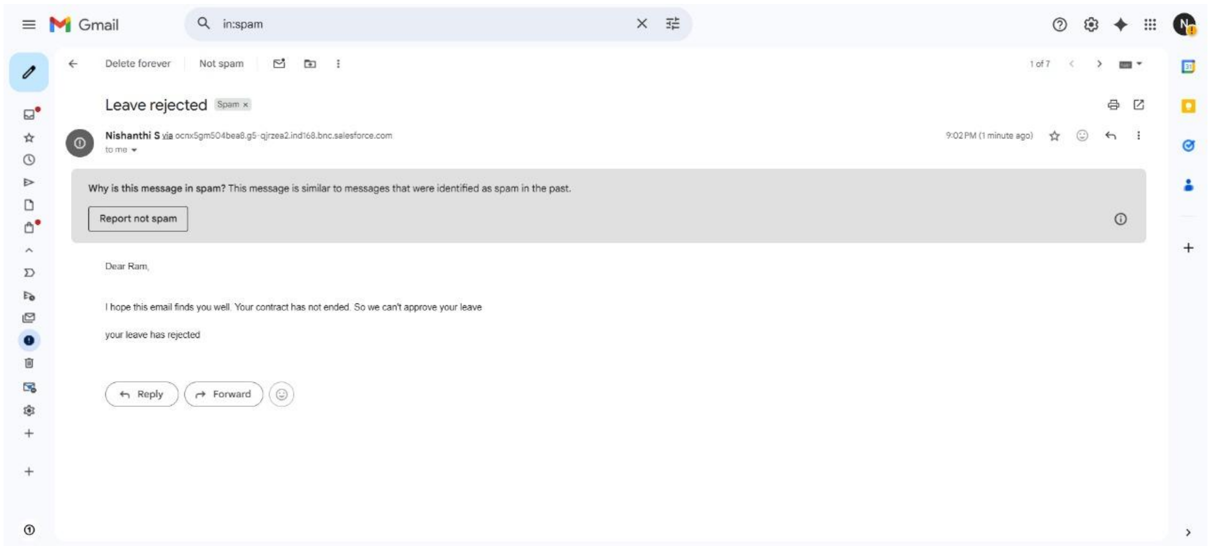
Leave rejected — 8:58 AM | Today
You sent an email to nishanthi.sweety1408@gmail.com

Leave approved — 8:57 AM | Today
You sent an email to nishanthi.sweety1408@gmail.com

No more past activities to load.

To Do List

The approval or rejection email is received to the tenant.



Leave rejected Spam

Nishanthi S via ocnx5gm504bea8.g5-qjrzea2.ind168.bnc.salesforce.com
to me

9:02 PM (1 minute ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Report not spam

Dear Ram,

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

your leave has rejected

7. DEPLOYMENT, DOCUMENTATION & MAINTENANCE

7.1 Deployment Strategy

To ensure a smooth transition from development to production, a well-defined deployment strategy has been implemented. The primary method used for deployment is Salesforce Change Sets, allowing for the structured movement of metadata between environments. This approach reduces risks, ensures stability, and maintains system integrity.

7.1.1 Key Components of Deployment:

• **Developer Sandbox Usage:**

All development and testing activities are conducted in Salesforce Developer Edition or Sandbox environments. This isolates testing from the live system and allows teams to experiment and validate without affecting end users.

Outbound Change Sets:

These are created in the Sandbox environment and include all the necessary components such as:

- Custom Objects and Fields
- Validation Rules o Classes and Triggers
- Process Builders and Flows
- Lightning Pages and Components Apex

Inbound Change Sets:

The packaged Change Sets are received in the Production Org and reviewed. Once verified, they are deployed after proper validation and testing, ensuring that only stable and approved features go live.

• **Pre-deployment Testing:**

Before any deployment, a complete testing cycle is conducted in a **Full Sandbox** environment.

This includes:

- Functional Testing
- Integration Testing
- Regression Testing

This helps identify potential bugs, configuration issues, or deployment errors early in the process.

7.2 Advanced Deployment Tools (Optional):

For larger projects or distributed teams, additional tools are considered for improved control and automation:

• SFDX CLI (Salesforce DX) for script-based deployment and continuous integration

• Git for version control and team collaboration

This deployment approach ensures consistency, minimizes risks, and supports scalable growth in the future.

7.3 System Maintenance & Monitoring

Post-deployment, the system needs to be actively monitored and maintained to ensure continued performance and data reliability. A structured maintenance plan has been put in place:

Key Activities:

• **Regular Data Backups:** Scheduled backups are configured to safeguard critical information such as lease contracts, tenant records, payment data, and related documents. Backups ensure business continuity in case of data loss or system failure.

• **Performance Monitoring:** Tools provided by Salesforce are used to continuously monitor system performance:

- Debug Logs to track system executions and identify performance bottlenecks
- Setup Audit Trail to monitor configuration changes and user actions
- Salesforce Health Check to assess security settings and overall system actions.

• **User Feedback Collection**: Regular feedback is gathered from end-users (property managers, admins, tenants) to identify UX/UI improvements and minor bugs. Feedback is logged and analyzed to drive iterative enhancements.

• **Scheduled Enhancements & Upgrades:** Based on usage patterns and evolving business needs, new features or process optimizations are scheduled periodically.

These may include:

- New report types
- Enhanced dashboards
- Additional automation (e.g., lease expiry followups)
- Workflow optimizations

• Security & Permission Reviews: As teams grow or roles change, it's crucial to ensure access levels are aligned.

Periodic reviews are conducted for:

- Role hierarchy
- Profile settings
- Permission sets and sharing rules

These reviews ensure that users only access the data they’re authorized to, maintaining trust and data protection compliance.

7.4 Documentation & Troubleshooting

Proper documentation is essential for smooth adoption, ongoing support, and future scalability. Two levels of documentation have been created:

7.4.1 User Documentation (User Guide & Admin Manual):

Designed for end-users and system administrators, this includes:

• **Step-by-step instructions** for key tasks such as:

- Creating a new lease record
- Viewing tenant profiles and payment history
- Generating and exporting reports
- Updating lease statuses (active, expired, terminated)

• **Common Troubleshooting Tips:**

- Handling record save errors caused by validation rules
- What to do when approval processes or flows don't trigger
- Resolving login issues and access permission problems

• **FAQs Section:**

Answers to frequently asked questions, separated for general users and system administrators. Helps reduce support requests and onboarding time for new users.

7.4.2 Technical Documentation:

Created for developers and future maintainers, this includes:

• **Data Model Diagram:**

A visual representation showing relationships between key custom objects like Leases, Tenants, Properties, and Payments.

• **Custom Objects & Fields List:**

A detailed list of all custom-built objects and fields, including field types, API names, and usage.

• **Automation & Apex Details:**

Documentation of flows, process builders, triggers, Apex classes, and any scheduled jobs implemented in the system.

• **Deployment Checklist:** A pre-deployment and post-deployment checklist used to ensure all components were tested, dependencies addressed, and configurations verified.