



## LEASE MANAGEMENT

Deepan S – 922222104006

Logesh S – 922222104026

Thirumani C – 922222104049

Panick Siranjan C – 922222104035

## Project Overview

The **Lease Management System** is a Salesforce-based application designed to simplify and automate property lease operations. The project focuses on managing properties, tenants, leases, and payments efficiently through a centralized system.

This application enables property owners and administrators to track lease agreements, tenant details, and payment records. It ensures smooth communication between landlords and tenants through automated workflows, validation rules, approval processes, and scheduled notifications.

## Objectives

The main goal of the Lease Management System is to digitalize and automate the lease lifecycle, from property registration to payment collection and lease termination.

### Business Goals:

- **Efficient Property Management:** Maintain detailed records of properties, including type, address, and square footage.
- **Tenant Information Management:** Store tenant contact details and lease status for easy access and updates.
- **Automated Payment Tracking:** Record and monitor monthly payments with automated email alerts for due and successful payments.
- **Streamlined Lease Operations:** Manage lease start and end dates, ensure data validation, and prevent overlapping tenancies.
- **Communication Automation:** Use email templates, flows, and Apex classes to automatically notify tenants regarding payment and lease status.

## Salesforce Key Features and Concepts Utilized

The system utilizes several key Salesforce features to provide a robust and automated environment:

### Custom Objects:

Created four main objects — *Property*, *Tenant*, *Lease*, and *Payment for Tenant* — to store and manage all data related to lease operation.

The screenshot shows the Salesforce Object Manager page. At the top, there are several tabs: Welcome to Salesforce, Recently Viewed, Lease Management - Light, - Student, Object Manager | Salesforce, and Developer Console. The URL in the address bar is orgfarm-0f175797f9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/home. Below the tabs, there's a search bar with 'Search Setup' and a navigation bar with 'Setup', 'Home', and 'Object Manager'. The main area is titled 'Object Manager' with a 'SETUP' icon. It displays a table with 52+ items, sorted by Last Modified. The columns are labeled: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The table includes rows for Tenant, lease, Payment for tenant, property, Work Type Group Member, Work Type Group, Work Type, Work Step Template, Work Step, Work Plan Template Entry, and Work Plan Template. The 'DEPLOYED' column shows checkmarks for most objects.

## Custom Tabs:

Tabs were created for each object to ensure easy navigation within the Lightning App.

## Lightning App:

Developed a custom app named **Lease Management** to bring together all the objects and features under a unified interface.

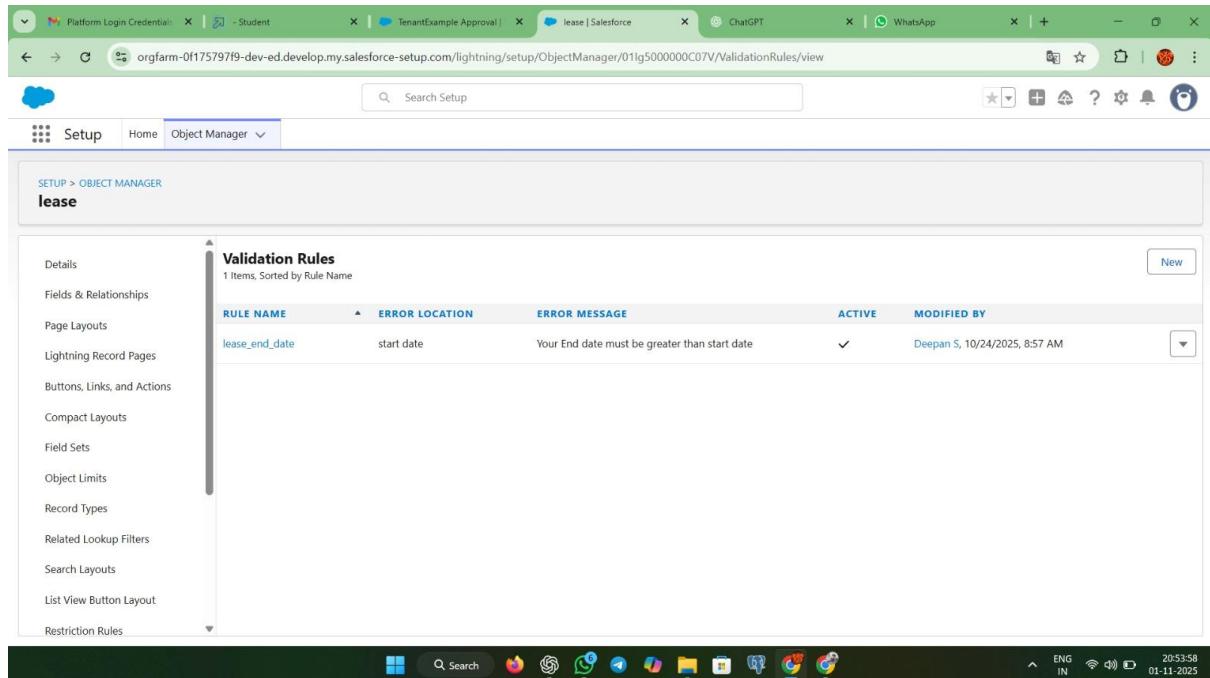
The screenshot shows the Lightning App Builder interface. The top navigation bar includes tabs for Lightning App Builder, App Settings, Pages, and Lease Management. The main content area is titled 'App Details & Branding'. On the left, a sidebar lists 'App Settings' and 'App Details & Branding' (which is currently selected). Under 'App Details & Branding', there are fields for 'App Name' (Lease Management), 'Developer Name' (Lease\_Management), and 'Description' (Enter a description...). To the right, there are sections for 'App Branding' (with an 'Image' field and 'Primary Color Hex Value' set to #0070D2) and 'Org Theme Options' (with a checkbox for 'Use the app's image and color instead of the org's custom theme'). Below this is an 'App Launcher Preview' section showing a blue square with 'LM' and the text 'Lease Management'. The bottom of the screen shows a taskbar with various application icons and system status indicators.

## **Relationships:**

- *Lookup Relationships* between Lease and Property, and Payment for Tenant and Tenant.
- *Master-Detail Relationship* between Property and Payment for Tenant.

## **Validation Rule:**

Ensured logical consistency by validating that the *End Date* of a lease is greater than the *StartDate*.



The screenshot shows the Salesforce Setup interface for the 'lease' object. On the left, a sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Validation Rules' and shows one item: 'lease\_end\_date'. The table has columns for RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The rule details are: RULE NAME: lease\_end\_date, ERROR LOCATION: start date, ERROR MESSAGE: Your End date must be greater than start date, ACTIVE: checked, and MODIFIED BY: Deepan S, 10/24/2025, 8:57 AM.

## **Approval Process:**

Implemented an approval workflow for tenant status (“Stay” or “Leaving”) that triggers automated email alerts for approval or rejection.

When the tenant submits the request, an email using the “**Tenant Leaving**” template is sent for approval. If **approved**, a “**Leave Approved**” email confirms the tenant can vacate. If **rejected**, a “**Leave Rejected**” email notifies that the request is denied.

**Email Templates Used** Various email templates are used to automate communication:

**Tenant Leaving:** Request for leave approval.

**Leave Approved / Leave Rejected:** Notify decision on the request.

**Monthly Payment Reminder:** Sends payment reminders and auto-confirmation when marked as “Paid” (via Flow).

**Successful Payment:** Confirms the rent payment has been received

The screenshot shows the Salesforce Setup interface for 'Approval Processes'. The top navigation bar includes tabs for Platform Login Credential, Student, TenantExample Approval, Approval Processes | Sales, ChatGPT, WhatsApp, and others. The main content area is titled 'Approval Processes' under the 'SETUP' tab. A sidebar on the left lists 'Process Automation' and 'Approval Processes'. A search bar at the top says 'Search Setup'.

**Active Approval Processes:**

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

**Inactive Approval Processes:**

No approval processes available			
---------------------------------	--	--	--

At the bottom, there's a toolbar with icons for Q Search, Firefox, ChatGPT, WhatsApp, and others, along with system status indicators like ENG IN, 20:57:00, and 01-11-2025.

This screenshot shows the 'Approval Processes' setup page with a different configuration. It includes sections for 'Initial Submission Actions', 'Approval Steps', 'Final Approval Actions', and 'Final Rejection Actions'.

**Initial Submission Actions:**

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

**Approval Steps:**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions   Edit	1	step1		Tenant: status NOTEQUALTO Leaving , else Approve	Manually Chosen	Final Rejection

**Final Approval Actions:**

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

**Final Rejection Actions:**

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

At the bottom, there's a toolbar with icons for Q Search, Firefox, ChatGPT, WhatsApp, and others, along with system status indicators like ENG IN, 20:58:47, and 01-11-2025.

## Email Templates:

In this project, various email templates are created to automate communication between tenants and property managers. These templates help in managing tenant requests, approvals, and payments efficiently without manual effort.

The screenshots show the 'Classic Email Templates' page in Salesforce. The left sidebar shows 'Email' categories: 'Classic Email Templates' (selected), 'Email to Salesforce', 'Filter Email Tracking', and 'Lightning Email Templates'. A search bar at the top has 'email t' typed in. The main area is titled 'Unfiled Public Classic Email Templates'.

**Screenshot 1:** Displays two templates: 'Leave approved' and 'Leave rejected'. Both are 'Text' type, marked as 'Available For Use', and created by 'sub' on 10/24/2025.

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit   Del	Leave approved	Text	✓		sub	10/24/2025
Edit   Del	Leave rejected	Text	✓		sub	10/24/2025

**Screenshot 2:** Displays three templates: 'tenant leaving', 'Tenant Email', and 'tenant payment'. All are 'Text' type, marked as 'Available For Use', and created by 'sub' on 10/24/2025.

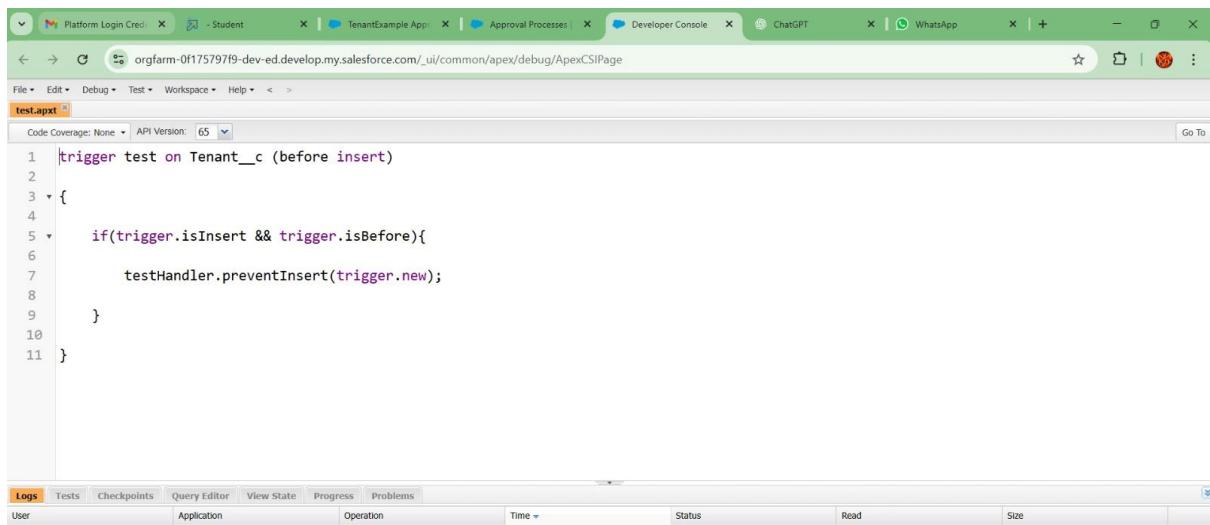
Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit   Del	tenant leaving	Text	✓		sub	10/24/2025
Edit   Del	Tenant Email	Text	✓		sub	10/24/2025
Edit   Del	tenant payment	Text	✓		sub	10/24/2025

**Screenshot 3:** Shows the URL 'javascript:srcSelf("%27%2F00X%3FappLayout%3Dsetup%26ln\_app\_id%3D%26setupid%3DComm...' in the address bar, indicating a page refresh or specific URL.

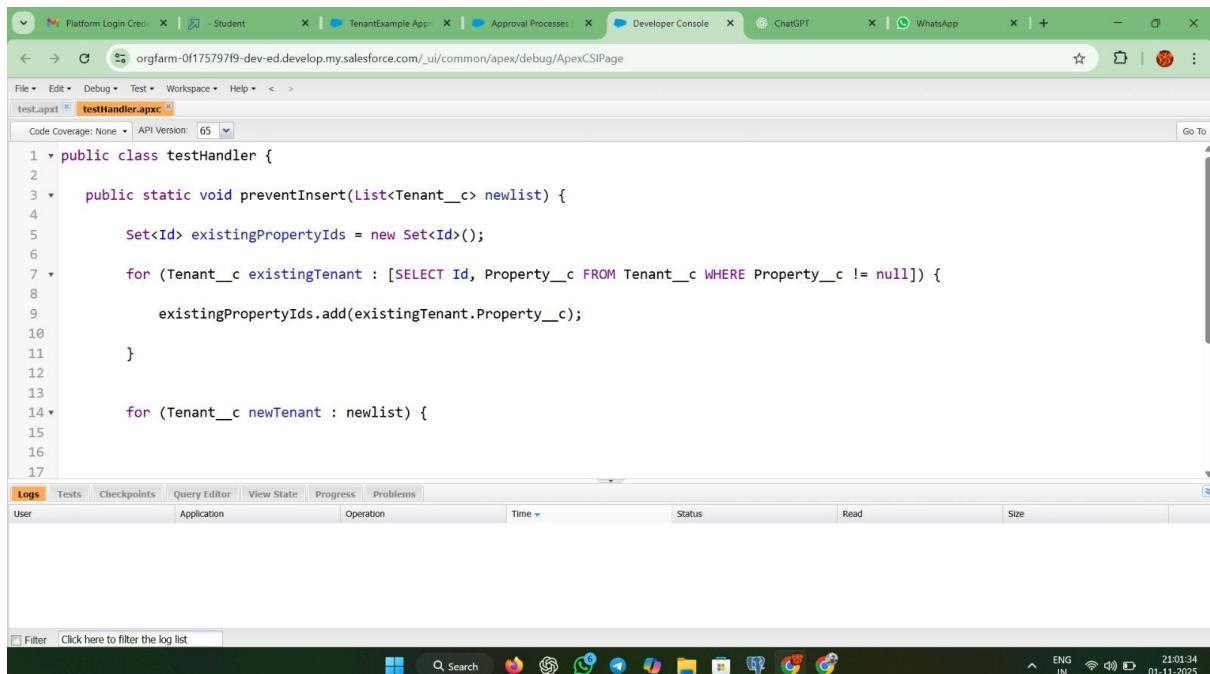
## Apex trigger and Handler class:

Implemented business logic to prevent multiple tenants from being assigned to the same property. An **Apex Trigger** and **Handler Class** are created to maintain data integrity between tenants and properties. The trigger, named “**test**”, runs **before inserting** a new tenant record. It ensures that each property can only be linked to **one tenant at a time**.

The **Handler Class (testHandler)** contains the main logic. It checks whether the property is already assigned to another tenant.



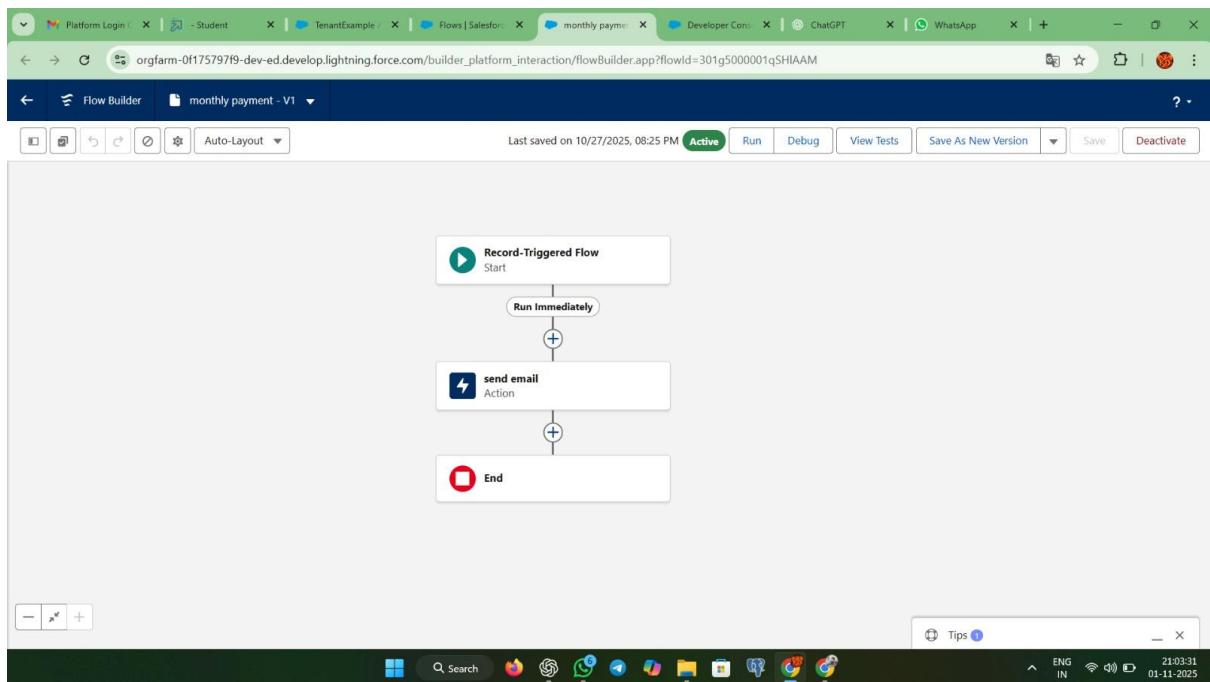
```
trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}
```



```
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) {
            existingPropertyIds.add(newTenant.Property__c);
        }
    }
}
```

## Flows and scheduled classes:

Automated monthly payment reminders using Record-Triggered Flows and an Apex Scheduler to send emails on the 1st of each month.



**Approval request for the tenant is approved**  
TenantExample  
Oct 27, 2025, 8:44 PM

**New Guidance Center learning resource available**  
Set Up Accounts & Contacts  
Start storing information about your customers with accounts and contacts.  
Oct 24, 2025, 9:01 PM

**New Guidance Center learning resource available**  
Define Your Sales Process  
Learn how to guide reps through the sales process.  
Oct 24, 2025, 9:01 PM

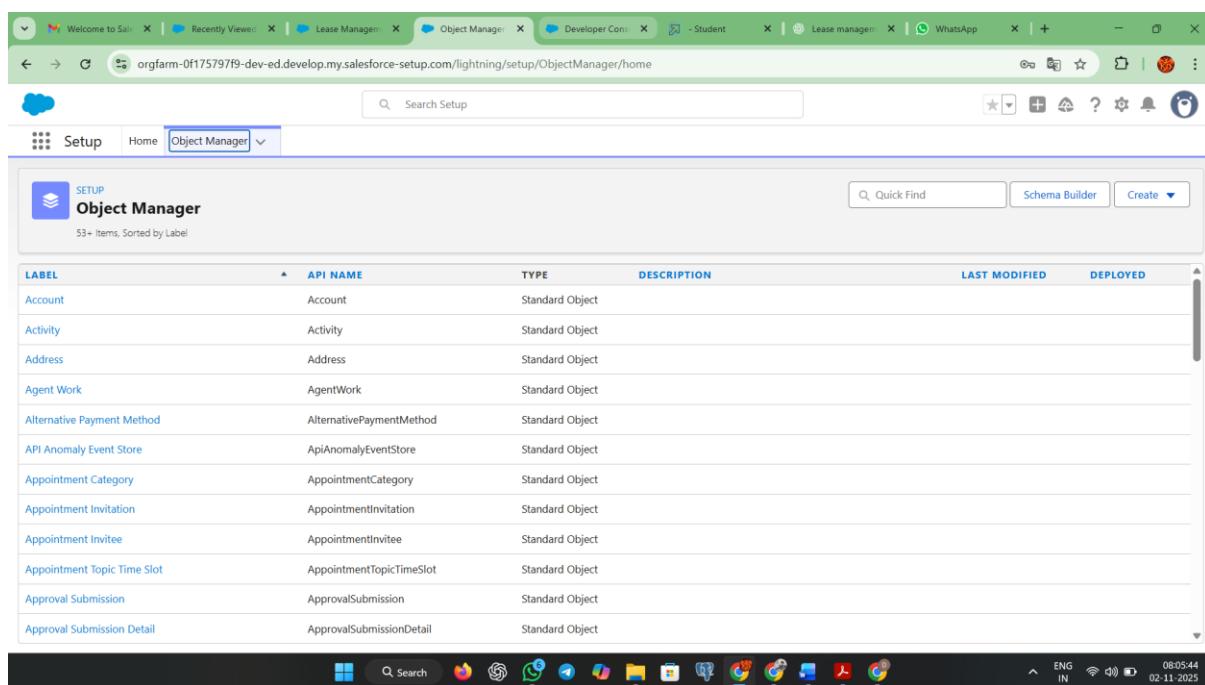
## Detailed Steps to Solution Design

### Step 1: Custom Object Creation

Created the following custom objects using the Object Manager:

- **Property** – to store property details.
- **Tenant** – to record tenant information.
- **Lease** – to manage lease agreements.
- **Payment for Tenant** – to track rent payments.

Each object was configured with “Allow Reports,” “Track Field History,” “Allow Activities,” and “Allow Search.”



The screenshot shows the Salesforce Object Manager page. At the top, there's a navigation bar with tabs for Setup, Home, and Object Manager (which is selected). Below the navigation is a search bar labeled "Search Setup". The main area is titled "Object Manager" and shows a list of 53+ items, sorted by Label. The list includes various standard objects like Account, Activity, Address, Agent Work, Alternative Payment Method, API Anomaly Event Store, Appointment Category, Appointment Invitation, Appointment Invitee, Appointment Topic Time Slot, Approval Submission, and Approval Submission Detail. Each row in the table has columns for Label, API Name, Type, Description, Last Modified, and Deployed. At the bottom of the page, there's a toolbar with icons for different functions and a status bar showing network connectivity and the date/time (02-11-2025).

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			

### Step 2: Creating Custom Tabs

Custom tabs were created for all four objects (Property, Tenant, Lease, and Payment for Tenant) to allow easy navigation and data entry within the Lightning App.

The screenshot shows the Salesforce Setup interface under the 'Tabs' section. The 'Custom Tabs' section is active, displaying a table of custom tabs:

Action	Label	Tab Style	Description
Edit   Del	lease	Globe	
Edit   Del	Payment	Shopping Cart	
Edit   Del	property	Building	
Edit   Del	Tenants	People	

### Step 3: Lightning App Creation

Developed a **Lightning App** named *Lease Management* with navigation items for:

- Property
- Tenant
- Lease
- Payment for Tenant

Set the navigation style as *Standard Navigation* and assigned the *System Administrator* profile for app access.

The screenshot shows the Lightning App Builder interface under the 'App Details & Branding' tab. The 'App Details' section contains the following information:

- App Name:** Lease Management
- Developer Name:** Lease\_Management
- Description:** Enter a description...

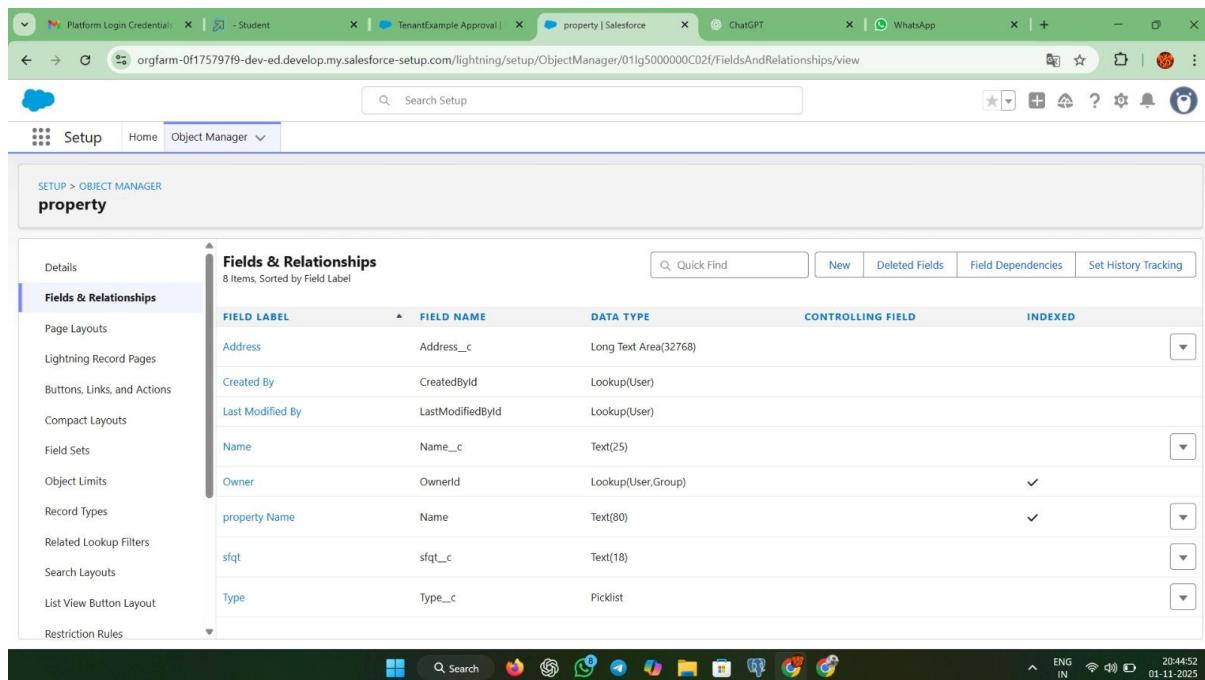
The 'App Branding' section includes:

- Image:** A placeholder for an image, with a 'Upload' button.
- Primary Color Hex Value:** #007002

### Step 4: Field Creation for Property Object

Added the following fields:

- **Name** (Text, Required)
- **Address** (Long Text Area)
- **Type** (Picklist: 1BHK, 2BHK, 3BHK)
- **Sqft** (Text, Length: 18)



The screenshot shows the Salesforce Setup interface with the 'property' object selected. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main content area displays the 'Fields & Relationships' section for the 'property' object. It shows 8 items, sorted by Field Label. The table includes columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Long Text Area(32768)		
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	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		

## Step 5: Field Creation for Tenant Object

Added key fields:

- **Email** (Email, Required)
- **Phone** (Phone)
- **Status** (Picklist: Stay, Leaving)

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-0f175797f9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01lg5000000By0s/FieldsAndRelationships/view>. The page title is "SETUP > OBJECT MANAGER" and the object name is "Tenant". The left sidebar shows various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled "Fields & Relationships" and lists 8 items. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

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)		✓

## Step 6: Field Creation for Lease Object

Configured:

- Start Date (Date)
- End Date (Date)

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-0f175797f9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01lg5000000C07V/FieldsAndRelationships/view>. The page title is "SETUP > OBJECT MANAGER" and the object name is "lease". The left sidebar shows various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled "Fields & Relationships" and lists 7 items. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

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		

## Step 7: Field Creation for Payment for Tenant Object

Defined fields:

- **Payment Date** (Date)
- **Amount** (Number, Length: 18)
- **Check for Payment** (Picklist: Paid, Not Paid)

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-0f175797f9-dev-ed.develop.my.salesforce-setup.com/lightning/setup/ObjectManager/01lg5000000C05t/FieldsAndRelationships/view>. The page title is "Payment for tenant". The left sidebar shows "FIELDS & RELATIONSHIPS" selected. The main content area displays a table titled "Fields & Relationships" with 8 items. The columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data is as follows:

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)		

## Step 8: Relationship Creation

- **Lease → Property:** Lookup Relationship
- **Payment for Tenant → Tenant:** Lookup Relationship
- **Payment for Tenant → Property:** Master-Detail Relationship

These relationships ensure linked and consistent data across all objects.

## Step 9: Validation Rule

Created a validation rule on **Lease Object**:

- **Rule Name:** lease\_end\_date
- **Condition:** End\_date\_c < Start\_date\_c
- **Error Message:** “Your End date must be greater than start date.”
- **Error Location:** Field → Start Date

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

- Page Header:** Welcome to Sales[redacted] | Recently Viewed | Lease Management | lease | Salesforce | Developer Cons[redacted] | - Student | Lease manager[redacted] | WhatsApp
- Search Bar:** Search Setup
- Breadcrumbs:** SETUP > OBJECT MANAGER > lease
- Left Sidebar:** Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules.
- Main Content:**
  - Section:** lease Validation Rule
  - Validation Rule Detail:**
    - Rule Name:** lease\_end\_date
    - Error Condition Formula:** End\_date\_\_c > start\_date\_\_c
    - Error Message:** Your End date must be greater than start date
    - Description:**
    - Created By:** Deepan\_S, 10/24/2025, 8:57 AM
    - Active:** ✓
    - Error Location:** start date
    - Modified By:** Deepan\_S, 10/24/2025, 8:57 AM
- Bottom:** Windows taskbar with various icons and system status.

## Step 10: Email Templates

Created multiple Classic Email Templates for communication:

**Tenant Leaving** – request approval for leave.

**Leave Approved** – confirmation email for approved leave.

**Leave Rejected** – notification for rejected leave.

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

- Page Header:** Platform Login Credentials | Student | TenantExample Approval | Classic Email Templates | S[redacted] | ChatGPT | WhatsApp
- Search Bar:** Search Setup
- Breadcrumbs:** SETUP > Email > Classic Email Templates
- Left Sidebar:** Email, Classic Email Templates (selected), Email to Salesforce, Filter Email Tracking, Lightning Email Templates.
- Main Content:**
  - Section:** Unified Public Classic Email Templates
  - Classic Email Template Availability:**
    - Folder:** Unified Public Classic Email Templates
    - Create New Folder:**
  - Table:** List of templates

Action	Email Template Name	Template Type	Available For Use	Description	Author	Last Modified Date
Edit   Del	tenant_leaving	Text	✓		sub	10/24/2025
Edit   Del	Tenant_Email	Text	✓		sub	10/24/2025
Edit   Del	tenant_payment	Text	✓		sub	10/24/2025
- Bottom:** Windows taskbar with various icons and system status.

**Tenant Email** – monthly rent payment reminder.

**Tenant Payment** – successful payment confirmation.

The screenshot shows the Salesforce Setup interface with the 'Classic Email Templates' page open. The left sidebar shows 'Email' categories: 'Classic Email Templates' (selected), 'Email to Salesforce', 'Filter Email Tracking', and 'Lightning Email Templates'. A search bar at the top right contains 'Search Setup'. The main content area is titled 'Classic Email Templates' and shows 'Unfiled Public Classic Email Templates'. It includes a table with columns: Action, Email Template Name, Template Type, Available For Use, Description, Author, and Last Modified Date. Two entries are listed: 'Leave approved' (Text, Available, sub, 10/24/2025) and 'Leave rejected' (Text, Available, sub, 10/24/2025). Navigation links at the bottom include letters from A to Z and 'Other'.

## Step 11: Approval Process

Created an approval process named **Check for Vacant** for the *Tenant* object. It automates email notifications and record updates during the approval lifecycle using the templates created above.

Actions:

- Initial Submission:** Sends *Tenant Leaving* email.
- Final Approval:** Sends *Leave Approved* email
- Final Rejection:** Sends *Leave Rejected* email

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The left sidebar shows 'Process Automation' categories: 'Approval Processes' (selected), 'Process Flow', and 'Process Builders'. A search bar at the top right contains 'Search Setup'. The main content area is titled 'Approval Processes' and shows 'Initial Submission Actions', 'Approval Steps', and 'Final Approval Actions'. The 'Initial Submission Actions' section has one entry: 'Record Lock' with description 'Lock the record from being edited please approve.my.leave' and 'Edit | Remove'. The 'Approval Steps' section has one step: 'step1' with action 'Tenant: status NOTEQUALTO Leaving , else Approve', assigned approver 'Manually Chosen', and reject behavior 'Final Rejection'. The 'Final Approval Actions' section has one entry: 'Record Lock' with description 'Lock the record from being edited Tenant leaving' and 'Edit | Remove'. Navigation links at the bottom include letters from A to Z and 'Other'.

## **Step 12: Apex Trigger and Handler Implementation**

- Created an **Apex Trigger** and **Handler Class** to ensure one property is assigned to only one tenant.
- Prevents duplicate tenant assignments and maintains clean, accurate data.

## **Step 13: Scheduled Apex Class**

- Implemented a **Scheduled Apex Class (MonthlyEmailScheduler)** to send rent reminder emails automatically on the first day of every month.
- This helps maintain regular rent payments and consistent communication with tenants.

## **Key Scenarios Addressed by Salesforce**

- **Property Management:** Salesforce custom objects simplify property record maintenance.
- **Tenant Tracking:** Ensures every tenant is linked to a property with no duplication.
- **Lease Validation:** Validation rule ensures accurate date management.
- **Automated Communication:** Email templates and flows improve tenant engagement.
- **Approval Workflow:** Automates tenant status updates with admin approval.

**Payment Tracking:** Payments and reminders are handled via flows and scheduled emails

## **Conclusion**

The **Lease Management System** project demonstrates the effective use of Salesforce to streamline rental operations. Through automation, relationship management, and workflow logic, the system enhances accuracy, transparency, and communication.

Key accomplishments include:

- End-to-end automation of lease, payment, and tenant processes.
- Secure and relational data structure.
- Automatic email notifications for tenants.
- Validation and approval systems for better control.
- Improved operational efficiency for property administrators.

The screenshot shows a Salesforce Lightning interface for a process instance step. The title bar indicates the URL is orgfarm-0f175797f9-dev-ed.lightning.force.com/lightning/r/ProcessInstanceStep/04hg5000000lwDAAU/view. The top navigation bar includes links for Platform Login, Student, TenantExample, Flows | Salesforce, monthly payme, Developer Com., ChatGPT, WhatsApp, and a plus sign for new items. The main header shows a blue cloud icon, the text 'Lease Management', and 'No Items'. A 'Setup' button is visible in the top right. The main content area displays a 'Process Instance Step' record for 'Tenant Approval' (status: Approved). It shows details: Submitter (Deepan S), Date Submitted (Oct 27, 2025), Actual Approver (Deepan S), and Assigned To (Deepan S). Below this, a 'Details' section contains 'Approval Details' with fields for Tenant Name (TenantExample) and Owner (Deepan S). To the right, an 'Approver Comments' section shows a comment from Deepan S: 'am' at Oct 27, 2025, 8:14:43 AM. The bottom of the screen shows a Windows taskbar with various pinned icons and system status indicators like battery level and date/time (21:05:57, 01-11-2025).