



**KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)**

LEASE MANAGEMENT

Submitted by:

SHAIK RUHEE

218x1a1235@khitguntur.ac.in

JANGA KEERTHANA

218x1a1254@khitguntur.ac.in

SHAIK SAJID

218x1a1236@khitguntur.ac.in

YERVA VENKATA ASHOK REDDY

218x1a1266@khitguntur.ac.in

Project overview

Lease Management in Salesforce helps businesses efficiently track, manage, and optimize lease agreements for properties, equipment, and vehicles. The platform provides a centralized system to oversee the entire lease lifecycle, from initiation and approval to execution, renewal, and termination. By leveraging Salesforce's automation capabilities, organizations can streamline workflows, set up automated reminders for lease renewals and payments, and ensure compliance with regulatory standards such as IFRS 16 and ASC 842.

One of the key advantages of using Salesforce for lease management is its ability to store and organize lease contracts and related documents within the system. Integration with tools like DocuSign and SharePoint allows seamless contract management, while Salesforce Flow automates approval processes, reducing manual effort and errors. Financial management is also enhanced through integration with Salesforce Billing and third-party ERP solutions, enabling automated rent collection, revenue recognition, and tracking of outstanding dues.

Additionally, Salesforce provides robust reporting and analytics features, allowing businesses to gain real-time insights into lease performance, occupancy rates, and financial forecasts. With AI-powered analytics from Einstein, companies can make data-driven decisions to optimize lease portfolios. Lease Management in Salesforce is particularly beneficial for industries such as real estate, equipment leasing, vehicle rental, and retail chains managing multiple locations. By offering enhanced visibility, efficiency, and compliance, Salesforce ensures that organizations can manage their leases more effectively while reducing operational risks.

Objectives

1. Business Goals:

1. Cost Optimization and Financial Control

Managing leases effectively helps businesses reduce unnecessary costs, avoid penalties, and optimize lease expenditures. By tracking payments, automating billing, and renegotiating terms, businesses can maintain financial stability and improve cash flow management.

2. Regulatory Compliance and Risk Mitigation

Compliance with lease accounting standards such as IFRS 16 and ASC 842 is essential for accurate financial reporting. Proper lease management ensures transparency, audit readiness, and adherence to legal requirements, reducing the risk of financial penalties or legal disputes.

3. Improved Operational Efficiency

Automating lease processes such as contract approvals, payment schedules, and renewals minimizes manual effort and enhances productivity.

4. Better Lease Portfolio Visibility

A centralized lease management system provides real-time insights into lease terms, expiration dates, and obligations.

5. Maximizing Asset Utilization

Ensuring that leased properties, equipment, or vehicles are optimally used prevents inefficiencies and unnecessary expenses. Lease management helps track asset performance, usage patterns, and maintenance schedules to maximize return on investment.

2. Business Outcomes:

1. Cost Reduction and Financial Optimization

Lease management helps businesses control lease-related expenses, avoid unnecessary penalties, and optimize financial planning. By tracking payments, automating rent schedules, and renegotiating lease terms, companies can reduce costs and improve overall profitability.

2. Regulatory Compliance and Risk Mitigation

Ensuring adherence to global lease accounting standards (e.g., IFRS 16, ASC 842) enhances financial transparency and prevents legal and financial risks. A structured lease management system enables audit readiness and compliance with contractual obligations.

3. Operational Efficiency and Process Automation

Automating lease-related workflows, such as approvals, renewals, and payment tracking, minimizes manual errors and administrative overhead. This results in improved productivity and better allocation of resources.

4. Enhanced Visibility and Decision-Making

Centralized lease data provides businesses with real-time insights into lease portfolios, contract terms, and financial obligations. This visibility supports strategic decision-making regarding lease renewals, terminations, and asset investments.

5. Maximized Asset Utilization

Effective lease management ensures optimal use of leased properties, equipment, or vehicles. By tracking lease terms and asset performance, businesses can prevent underutilization and improve return on investment.

Salesforce Key Features And Concepts Utilized

Salesforce provides a powerful and flexible platform for lease management by leveraging various features and concepts that enhance automation, visibility, and efficiency. Below are the key Salesforce functionalities used in lease management:

1. Lease Lifecycle Management with Standard & Custom Objects

- Salesforce uses **Standard Objects** (Accounts, Contacts, Opportunities, and Contracts) and **Custom Objects** (e.g., Lease Agreements, Properties, Assets) to structure lease data.
- Lease agreements are stored and linked to tenants, properties, and payment schedules for easy tracking.

2. Workflow Automation & Process Management

- **Salesforce Flow & Process Builder:** Automates lease approval processes, contract renewals, and notifications.
- **Approval Processes:** Ensures that lease agreements go through the necessary approval hierarchies before execution.
- **Scheduled Reminders & Notifications:** Automated alerts for lease payments, expiration dates, and compliance deadlines.

3. Document & Contract

- **Salesforce Files & Attachments:** Stores lease agreements, amendments, and supporting documents.
- **DocuSign & Adobe Sign Integration:** Enables electronic signing of lease contracts directly within Salesforce.

4. Payment & Financial Management

- **Salesforce Billing:** Automates recurring lease payments, rent calculations, and invoicing.
- **Revenue Recognition & Compliance:** Integrates with accounting systems to ensure adherence to lease accounting standards (IFRS 16, ASC 842).

5. Asset & Property Tracking

- **Salesforce Asset Management:** Tracks leased assets (e.g., equipment, vehicles, properties) linked to lease contracts.
- **Field Service Management:** Helps with maintenance tracking and service requests for leased properties or equipment.

6. Reports & Analytics for Lease Performance

- **Salesforce Reports & Dashboards:** Provides real-time insights into lease portfolios, payment trends, and contract statuses.
- **Einstein Analytics:** AI-powered insights for forecasting lease revenues and optimizing lease terms.

7. Integration with Third-Party Lease Management Solutions

- **AppExchange Solutions:** Salesforce integrates with lease management apps like **LeaseAccelerator**, **Nakisa Lease Administration**, and **Visual Lease** for advanced lease tracking and compliance.
- **ERP & Financial System Integration:** Connects with SAP, Oracle, or QuickBooks for end-to-end financial reporting.

8. Customer & Tenant Relationship Management

- **Salesforce CRM:** Enhances lease negotiations and customer interactions by tracking communication history, lease terms, and tenant feedback.

- **Omni-Channel Support:** Enables support teams to handle lease-related inquiries across multiple channels (email, chat, phone).

9. Mobile Accessibility & Self-Service Portals

- **Salesforce Mobile App:** Allows lease managers to access lease records and track payments on the go.
- **Experience Cloud (Self-Service Portals):** Enables tenants or leaseholders to view their lease status, payment schedules, and submit requests.

Detailed Steps to Solution Design

Creating Developer Account:

Creating a salesforce developer org for <https://developer.salesforce.com/signup>.

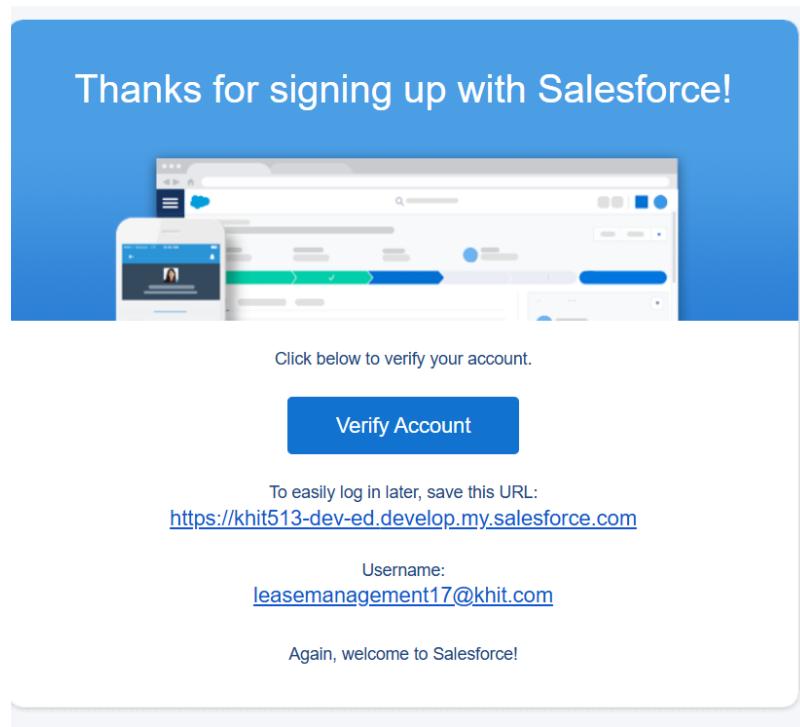
The screenshot shows a registration form for a Salesforce developer account. The fields are as follows:

- * First Name: Ruhee
- * Last Name: team17
- * Email address: ruheeshai177@gmail.com
- * Role: Developer
- * Company: KHIT
- * Country: India
- * Postal Code: 522019
- * Username: leasemanagement17@khit.com

At the bottom, there is a checked checkbox with the text: "By registering, you confirm you have read and agree to the [Terms of Use](#) and the [Master Subscription Agreement](#) and agree that my data is subject to the [Privacy Statement](#), including use for marketing purposes.*"

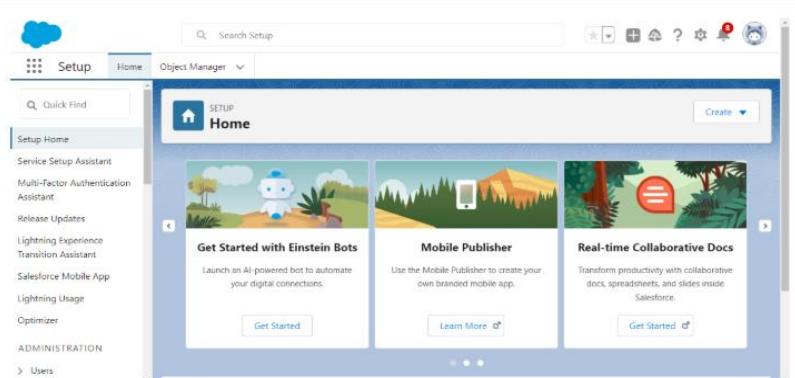
Account Activation

After creation we have got accepted mail to verify the account.



After clicking on the verify account, We have to set a password for the org and use it of the project.

After successfully login to the account, Then you will redirect to your salesforce setup page.



Object:

What Is an Object?

Salesforce objects are database tables that permit you to store data that is specific to an organization. What are the types of Salesforce objects.

They are two types of objects:

1. Standard Objects.

2. Custom Objects.

Standard Objects:

Standard objects are pre-defined by Salesforce and form the core foundation of the CRM system. These objects are designed to store common business data that organizations typically use in sales, marketing, and customer support processes. Examples include Accounts, which represent companies or organizations, Contacts, which store details of individuals associated with accounts, and Opportunities, which track potential sales deals. Standard objects also include Leads (potential customers) and Cases (customer service issues).

Custom Objects:

Custom objects, on the other hand, are user-defined objects created to store data specific to an organization's unique needs. While standard objects cover general CRM use cases, businesses often require additional fields and objects to manage industry-specific processes. For instance, a hospital might create a custom object called Patient__c to track patient records, while a construction company may use Project__c to manage project details. Custom objects come with their own fields, relationships, page layouts, and security settings, allowing businesses to tailor Salesforce to their workflows.

Creation of Property Object:

To create a property object we should go to setup in a salesforce org.

Follow the steps below to create a custom object in Salesforce:

1. From the Setup page, click on 'Object Manager'.
2. Click on 'Create'.
3. Click on 'Custom Object'.
4. Enter the label name: Property.
5. Enter the plural label name: Property.
6. Enter the Record Name Label and Format:
 - Record Name: Property Name
 - Data Type: Text
7. Enable the following options:
 - Allow Reports
 - Track Field History
 - Allow Activities
8. Enable 'Allow Search'.
9. Click 'Save' to complete the process.

The screenshot shows the 'Custom Object Definition Edit' page. At the top, there are three buttons: 'Save', 'Save & New', and 'Cancel'. Below this is a section titled 'Custom Object Information' with a note: 'The singular and plural labels are used in tabs, page layouts, and reports. Be careful when changing the name or label as it may affect existing integrations and merge templates.' It includes fields for 'Label' (set to 'property') and 'Plural Label' (set to 'property'), both with examples ('Example: Account' and 'Example: Accounts'). There is also a checkbox for 'Starts with vowel sound'. The 'Object Name' field is set to 'property' with an example 'Example: Account'. A large 'Description' text area is empty. Under 'Content Name', there is a dropdown menu set to 'None'. At the bottom, there is a section titled 'Enter Record Name Label and Format' with a note about record names appearing in various contexts. A 'Record Name' field is shown with a value of 'Records'. The page has a light gray background with some darker gray sections for form fields.

Creation of Tenant Object:

1. From the Setup page, click on 'Object Manager'.
2. Click on 'Create'.
3. Click on 'Custom Object'.
4. Enter the label name: Tenant.
5. Enter the plural label name: Tenants.
6. Enter the Record Name Label and Format:
 - Record Name: Tenant Name
 - Data Type: Text
7. Enable the following options:
 - Allow Reports
 - Track Field History
 - Allow Activities
8. Enable 'Allow Search'.
9. Click 'Save' to complete the process.

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

- Header:** Shows the standard Salesforce navigation bar with links for Home, Object Manager, and various system icons.
- Page Title:** "SETUP" and "New Custom Object".
- Message Bar:** "New Custom Object" with a "Help for this Page" link.
- Message Area:** A yellow bar stating, "Permissions for this object are disabled for all profiles by default. You can enable object permissions in permission sets or by editing custom profiles." It includes "Tell me more" and "Don't show this message again" links.
- Form Fields (Custom Object Information):**
 - Label:** Tenant (Example: Account)
 - Plural Label:** Tenants (Example: Accounts)
 - Starts with vowel sound:** An unchecked checkbox.
- API Fields:**
 - Object Name:** Tenant (Example: Account)
- Description:** A large text input field.
- Help Settings:**
 - Context Sensitive Help Setting:** Radio buttons for "Open the standard Salesforce.com Help & Training window" (selected) and "Open a window using a Visualforce page".
 - Content Name:** A dropdown menu showing "-None-".
- Record Name Label and Format:** A section at the bottom of the form.

Creation Payment Object:

To create a payment object we have to follow this steps

1. From the Setup page, click on 'Object Manager'.
2. Click on 'Create'.
3. Click on 'Custom Object'.
4. Enter the label name: Payment for Tenant.
5. Enter the plural label name: Payment.
6. Enter the Record Name Label and Format:
 - Record Name: Payment Name
 - Data Type: Text
7. Enable the following options:
 - Allow Reports
 - Track Field History
 - Allow Activities
8. Enable 'Allow Search'.

Click 'Save' to complete the process

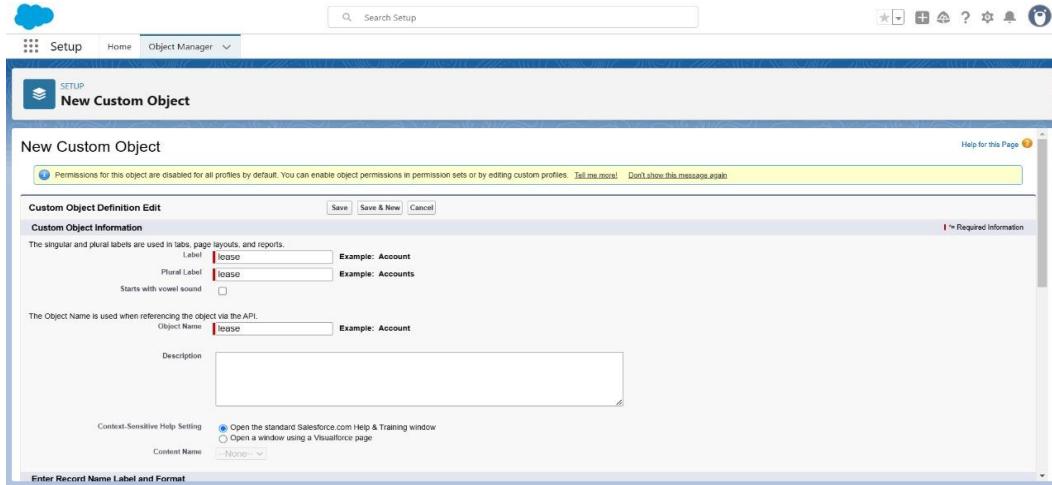
The screenshot shows the 'Custom Object Definition Edit' page in Salesforce. At the top, there are three buttons: 'Save', 'Save & New', and 'Cancel'. Below the buttons, a note says: 'The singular and plural labels are used in tabs, page layouts, and reports. Be careful when changing the name or label as it may affect existing integrations and merge templates.' The 'Label' field contains 'Payment for tenant' with an example 'Account'. The 'Plural Label' field contains 'Payment' with an example 'Accounts'. There is an unchecked checkbox for 'Starts with vowel sound'. The 'Object Name' field contains 'Payment' with an example 'Account'. A large empty text area labeled 'Description' is present. Under 'Context-Sensitive Help Setting', the radio button for 'Open the standard Salesforce.com Help & Training window' is selected. The 'Content Name' dropdown is set to 'None'. At the bottom, a note states: 'The Record Name appears in name layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always'. The entire form is enclosed in a light gray border.

Creation of Lease Object:

Follow the steps below to create a custom object in Salesforce:

1. From the Setup page, click on 'Object Manager'.

2. Click on 'Create'.
3. Click on 'Custom Object'.
4. Enter the label name: Lease.
5. Enter the plural label name: Lease.
6. Enter the Record Name Label and Format:
 - Record Name: Lease Name
 - Data Type: Text
7. Enable the following options:
 - Allow Reports
 - Track Field History
 - Allow Activities
8. Enable 'Allow Search'.
9. Click 'Save' to complete the process.



TABS

What is Tab:

A tab is like a user interface that is used to build records for objects and to view the records in the objects.

1. Visualforce Tabs

Visualforce Tabs are custom tabs that display a Visualforce page. Visualforce tabs look and behave like standard salesforce.com tabs such as accounts, contacts, and opportunities.

2.Lightning Component Tabs

Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app.

3.Lightning Page Tabs

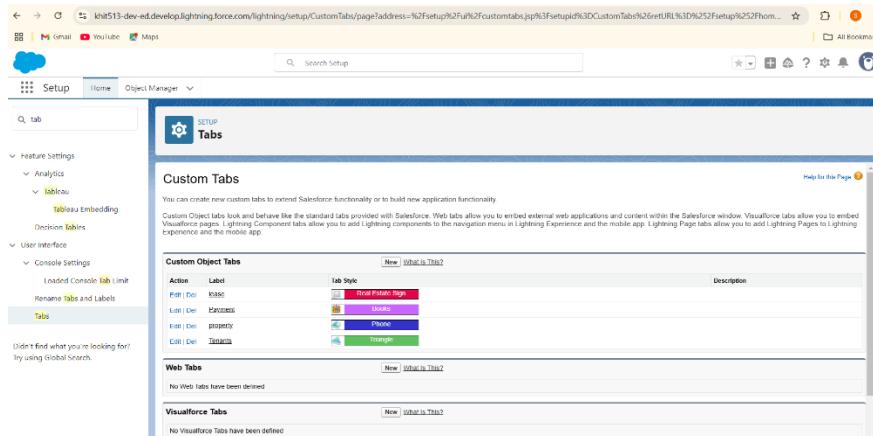
Lightning Page Tabs let you add Lightning Pages to the mobile app navigation menu.

Lightning Page tabs don't work like other custom tabs.

Creating a Custom Tab:

Follow the steps below to create a custom tab for the Property object in Salesforce:

1. Go to the Setup page.
2. Type 'Tabs' in the Quick Find bar and click on 'Tabs'.
3. Under 'Custom Object Tabs', click on 'New'.
4. Select the Object: Property.
5. Select the Tab Style.
6. Click 'Next'.
7. On the 'Add to Profiles' page, keep the default settings.
8. Click 'Next'.
9. On the 'Add to Custom App' page, uncheck the 'Include Tab' option.
10. Ensure that 'Append tab to users' existing personal customizations' is checked.
11. Click 'Save' to complete the process.



The Lightning App

An app is a collection of items that work together to serve a particular function. In Lightning Experience, Lightning apps gives users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar.

Lightning apps let you brand your apps with a custom color and logo. You can even include a utility bar and Lightning page tabs in your Lightning app. Members of your org can work more efficiently by easily switching between apps.

Use Case:

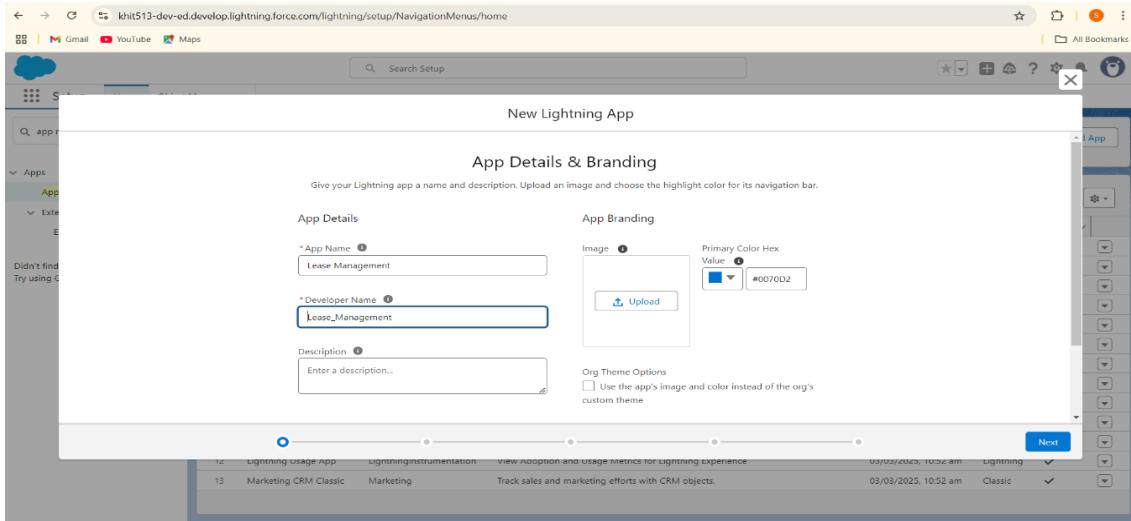
Well done you have reached close to your requirement by creating the objects to store the organisation's data. Making a database for an organisation is just not enough to reach out the requirements, the task is how the users at the organisation can access the objects you have created for them. As an Admin for the organisation it's your duty to make sure every user of the organisation is able to access the data modelling structure.

Steps to Create a Lightning App Page in Salesforce

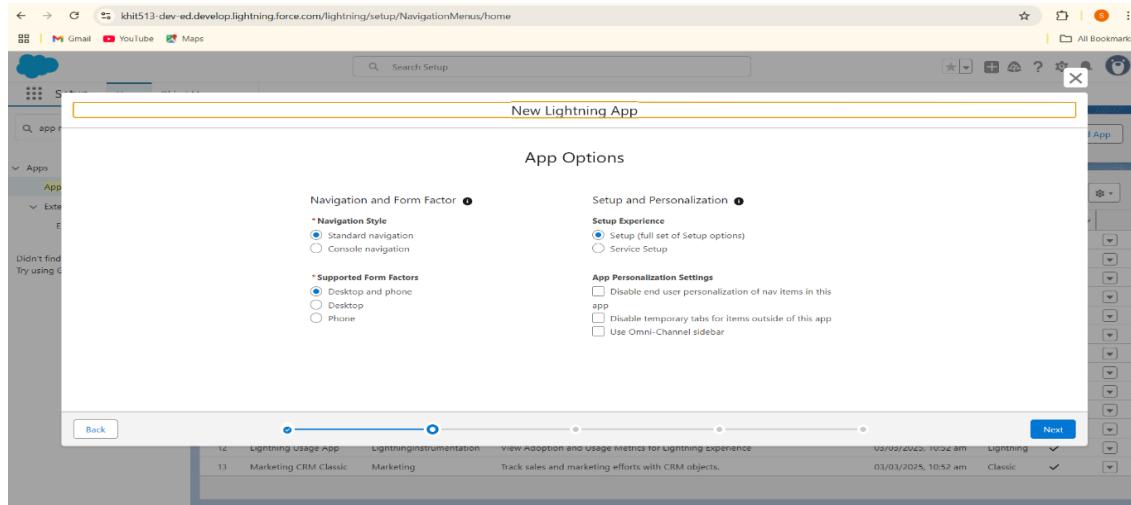
Follow the steps below to create a Lightning App Page in Salesforce:

Go to the Setup page.

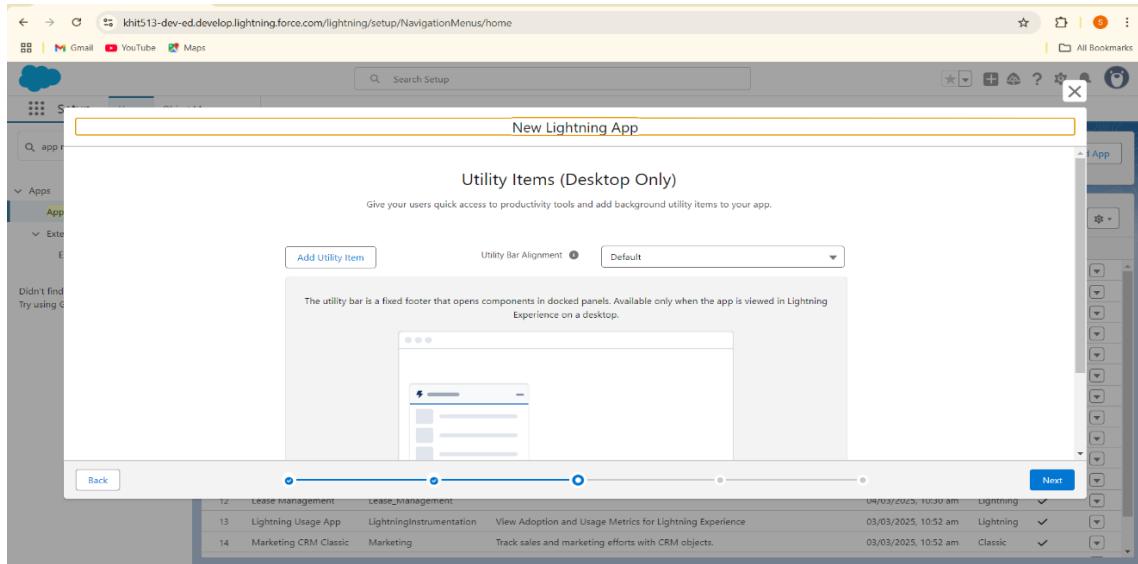
1. Search for 'App Manager' in the Quick Find bar and select 'App Manager'.
2. Click on 'New Lightning App'.
3. Fill in the app details and branding as follows:
 - App Name: Lease Management
 - Developer Name: Auto-populated
 - Primary Colour Hex Value: Keep as default



4. Click 'Next'.
5. On the 'App Options' page, set the Navigation Style as 'Standard Navigation'.

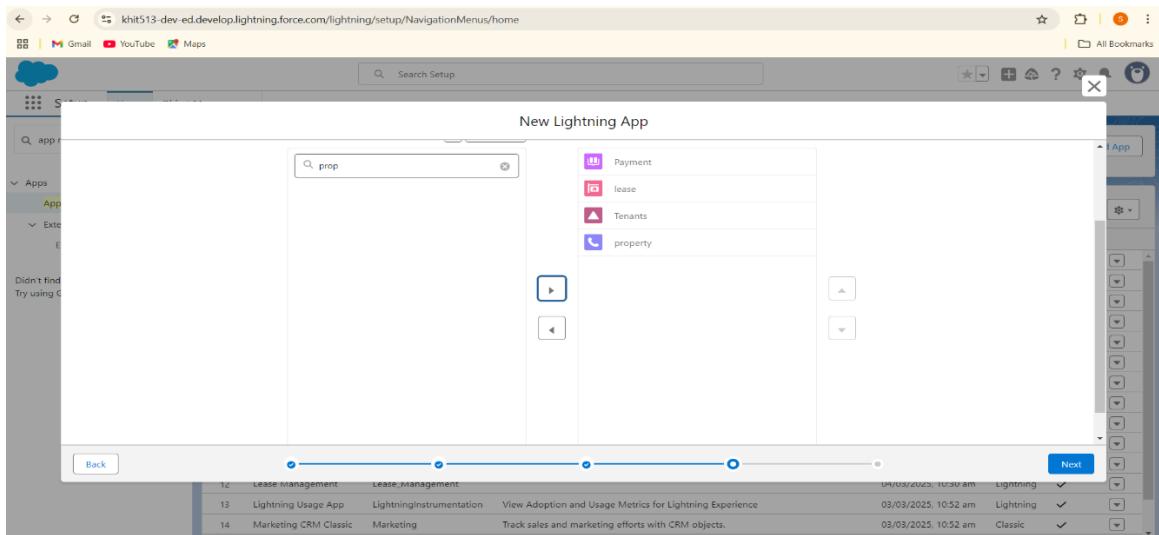


6. Click 'Next'.
7. On the 'Utility Items' page, keep it as default.
8. Click 'Next'.



9. To Add Navigation Items:

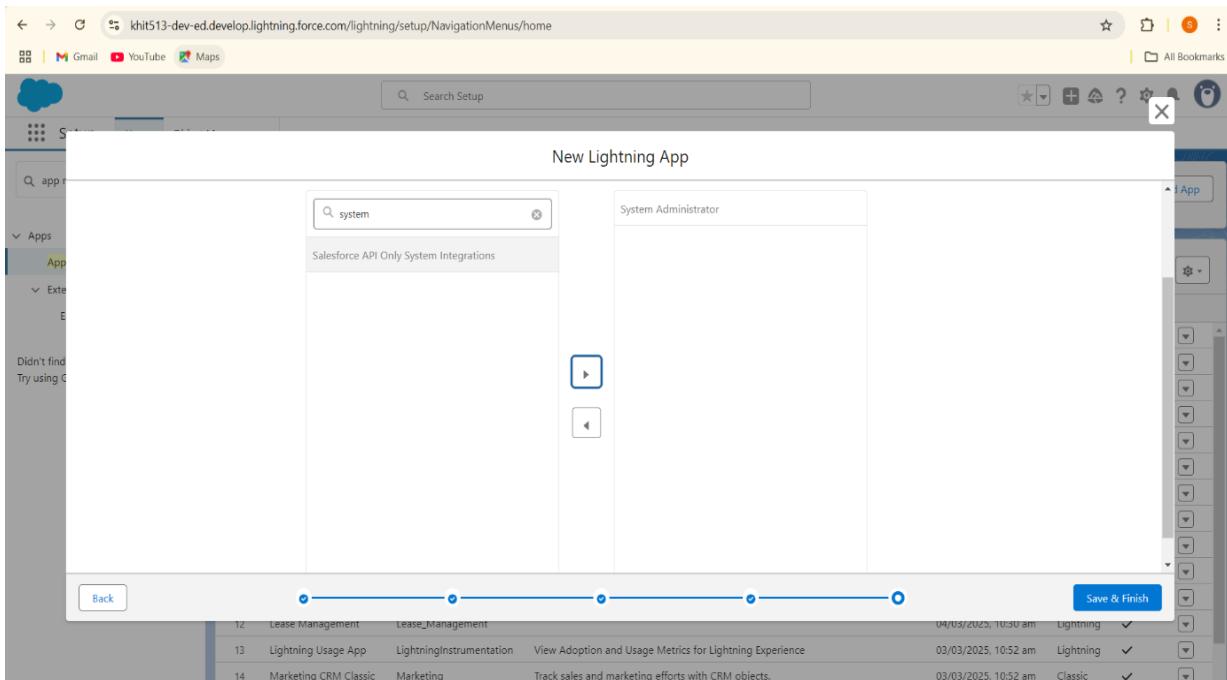
- Search for the items (Payment for Tenant, Tenants, Property, Lease) in the search bar.
- Move the selected items using the arrow button.
- Click 'Next'.



10. Click 'Next' again.

11. To Add User Profiles:

- Search for 'System Administrator' in the search bar.
- Click on the arrow button to add it.
- Click 'Save & Finish' to complete the setup.



Fields

When we talk about Salesforce, Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Types of Fields

1. Standard Fields
2. Custom Fields

Standard Fields:

As the name suggests, the Standard Fields are the predefined fields in Salesforce that perform a standard task. The main point is that you can't simply delete a Standard Field until it is a non-required standard field. Otherwise, users have the option to delete them at any point from the application freely. Moreover, we have some fields that you will find common in every Salesforce application. They are,

>>Created By >>Owner >>Last Modified >> Field Made During object Creation

Custom Fields:

On the other side of the coin, Custom Fields are highly flexible, and users can change them according to requirements. Moreover, each organizer or company can use them if necessary. It means you need not always include them in the records, unlike Standard fields. Hence, the final decision depends on the user, and he can add/remove Custom Fields of any given form.

Creation of fields for the property object

Steps to Create Multiple Fields in an Object in Salesforce

Follow the steps below to create multiple fields in the 'Property' object in Salesforce:

Create a Name Field in Property object

1. Go to the Setup page.
2. Click on 'Object Manager'.
3. Type the object name ('Property') in the search bar.
4. Click on the object to open its settings.
5. Click on 'Fields & Relationships'.
6. Click on 'New'.
7. Select the Data Type as 'Text'.
8. Click 'Next'.
9. Fill in the details as follows:
 - a. Field Label: Name
 - b. Field Name: Auto-generated
 - c. Length: 25
 - d. Required: Check the checkbox
10. Click 'Next'.
11. Click 'Next' again.
12. Click 'Save & New' to continue adding more fields.

Create a Address Field in Property Object

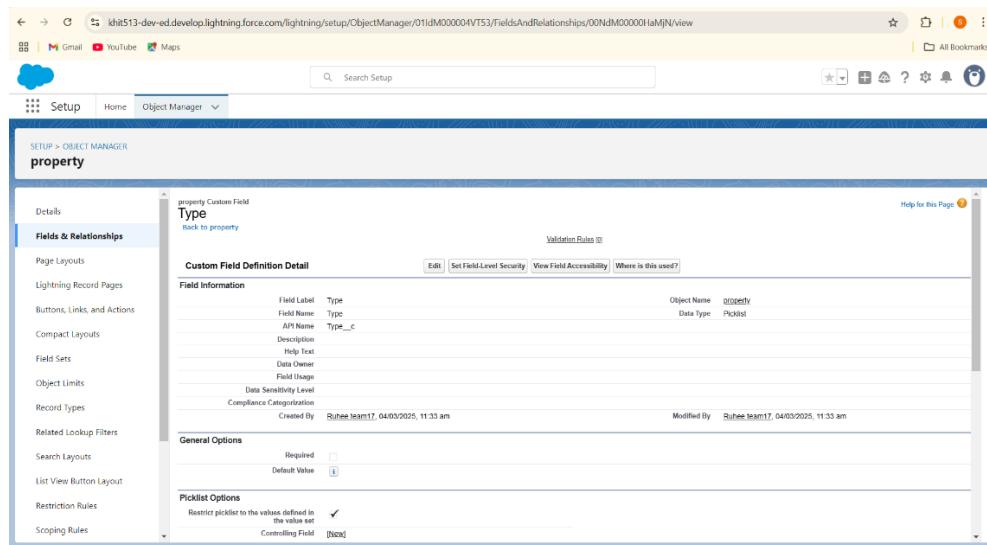
While we are still in property object

1. Select the Data Type as 'Long Text'.
2. Click 'Next'.
3. Fill in the details as follows:
 - a. Field Label: Address
 - b. Field Name: Auto-generated
4. Click 'Next'.
5. Click 'Next' again.
6. Click 'Save & New' to continue adding more fields.

Create a Type Field in Property Object

While we are still in property object

1. Select the Data Type as 'Picklist'.
2. Click 'Next'.
3. Fill in the details as follows:
 - a. Field Label: Type
 - b. Field Name: Auto-generated
4. Enter values, with each value separated by a new line:
 - a. 1BHK
 - b. 2BHK
 - c. 3BHK
5. Click 'Next'.
6. Click 'Next' again.
7. Click 'Save & New' to continue adding more fields.

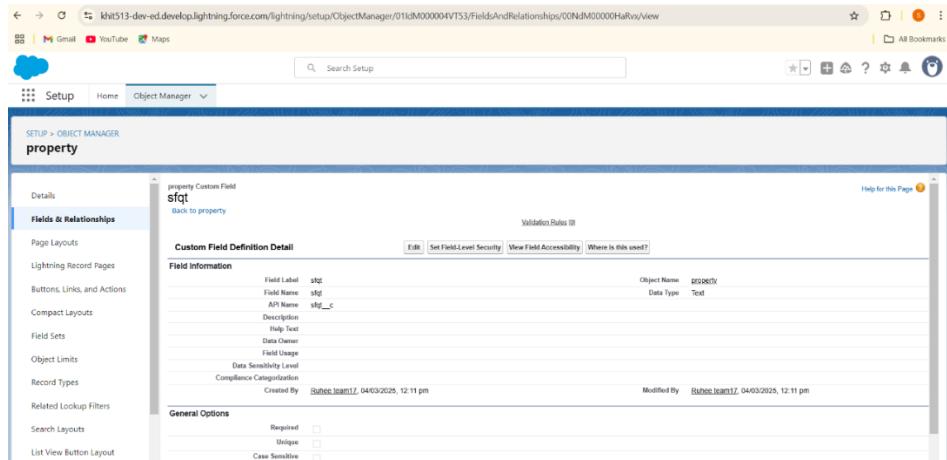


Create of sfqt Field in Property Object

While we are still in property object

1. Select the Data Type as 'Text'.

2. Click 'Next'.
3. Fill in the details as follows:
 - a. Field Label: sfqt
 - b. Field Name: Auto-generated
 - c. Length: 18
4. Click 'Next'.
5. Click 'Next' again.
6. Click 'Save' to complete the process.



Creation of fields for the Tenant object

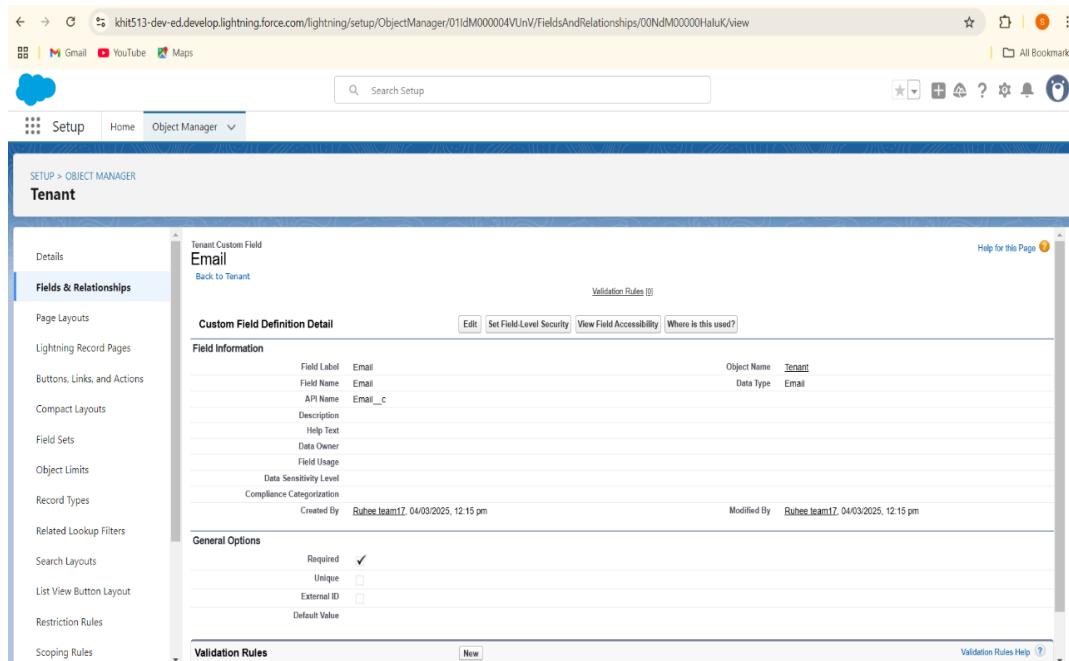
Steps to Create Multiple Fields in Tenant Object in Salesforce

Follow the steps below to create multiple fields in the 'Tenant' object in Salesforce:

Create an Email Field

1. Go to the Setup page.
2. Click on 'Object Manager'.
3. Type the object name ('Tenant') in the search bar.
4. Click on the object to open its settings.

5. Click on 'Fields & Relationships'.
6. Click on 'New'.
7. Select the Data Type as 'Email'.
8. Click 'Next'.
9. Fill in the details as follows:
 1. Field Label: Email
 2. Field Name: Auto-generated
 3. Click on the 'Required' checkbox.
10. Click 'Next'.
11. Click 'Next' again.
12. Click 'Save & New' to continue adding more fields.



Create a Phone Field

1. Select the Data Type as 'Phone'.
2. Click 'Next'.
3. Fill in the details as follows:
 1. Field Label: Phone
 2. Field Name: Auto-generated

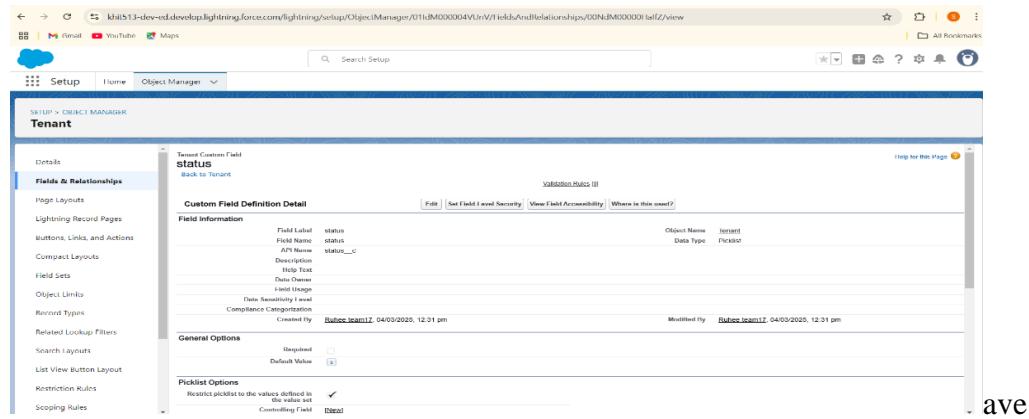
4. Click 'Next'.
5. Click 'Next' again.
6. Click 'Save & New' to continue adding more fields.

The screenshot shows the Salesforce Setup interface under 'Object Manager'. A custom field named 'Phone' has been created for the 'Tenant' object. The field details are as follows:

- Field Label:** Phone
- Field Name:** Phone
- API Name:** Phone_c
- Description:** Help Text
- Data Owner:** Field Usage
- Data Sensitivity Level:** Compliance Categorization
- Object Name:** Tenant
- Data Type:** Phone
- Created By:** Ruhee team17
- Modified By:** Ruhee team17
- Required:** No
- Default Value:** None
- Validation Rules:** None defined.

Create a Status Field

1. Select the Data Type as 'Picklist'.
2. Click 'Next'.
3. Fill in the details as follows:
 1. Field Label: Status
 2. Field Name: Auto-generated
4. Enter values, with each value separated by a new line:
 1. Stay
 2. Leaving
5. Click 'Next'.
6. Click 'Next' again.
7. Click 'Save'



Creation of fields for the Lease object

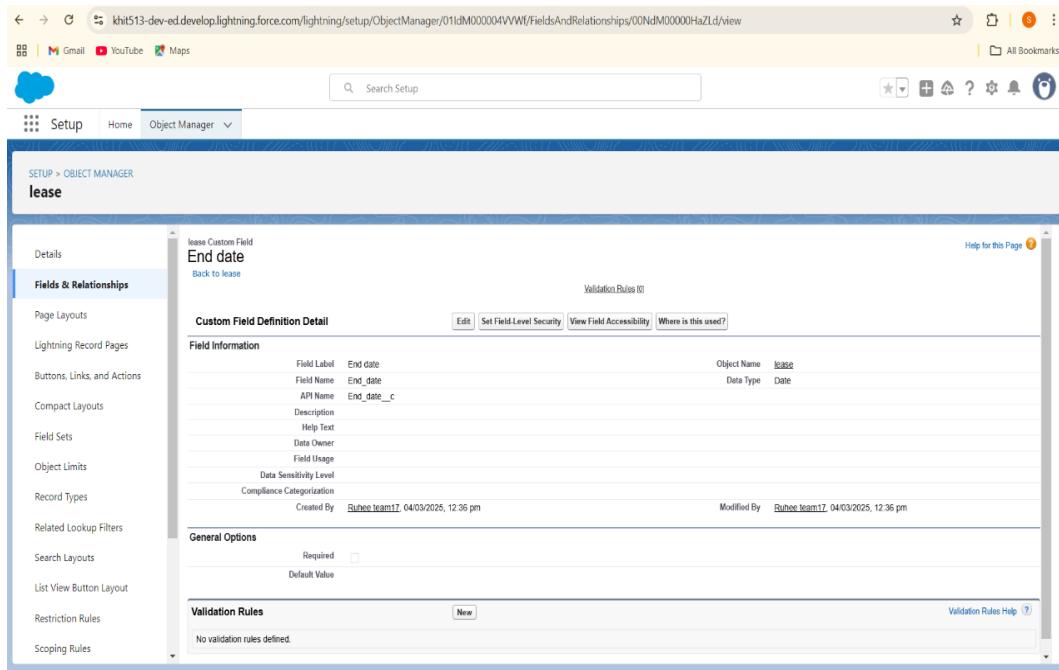
1. Create a Start Date Field

1. Go to the Setup page.
2. Click on 'Object Manager'.
3. Type the object name ('Lease') in the search bar.
4. Click on the object to open its settings.
5. Click on 'Fields & Relationships'.
6. Click on 'New'.
7. Select the Data Type as 'Date'.
8. Click 'Next'.
9. Fill in the details as follows:
 1. Field Label: Start Date
 2. Field Name: Auto-generated
10. Click 'Next'.
11. Click 'Next' again.
12. Click 'Save & New' to continue adding more fields.

The screenshot shows the Salesforce Setup interface under the Object Manager. A custom field named 'start date' is being created for the 'lease' object. The field is defined as a Date type. The 'Fields & Relationships' tab is selected. The 'Field Information' section shows the field label 'start date', field name 'start_date', API name 'start_date__c', and data type 'Date'. The 'General Options' section indicates the field is required. The 'Validation Rules' section is empty.

2. Create an End Date Field

1. Select the Data Type as 'Date'.
2. Click 'Next'.
3. Fill in the details as follows:
 1. Field Label: End Date
 2. Field Name: Auto-generated
4. Click 'Next'.
5. Click 'Next' again.
6. Click 'Save' to complete the process.



Creation of fields for the Payment for tenant object

Create a Payment Date Field

1. Go to the Setup page.
2. Click on 'Object Manager'.
3. Type the object name ('Payment for Tenant') in the search bar.
4. Click on the object to open its settings.
5. Click on 'Fields & Relationships'.
6. Click on 'New'.
7. Select the Data Type as 'Date'.
8. Click 'Next'.
9. Fill in the details as follows:
 1. Field Label: Payment Date
 2. Field Name: Auto-generated
10. Click 'Next'.
11. Click 'Next' again.
12. Click 'Save & New' to continue adding more fields.

The screenshot shows the Salesforce Setup interface under the Object Manager. A custom field named 'Payment for tenantat' has been created. The field details are as follows:

- Field Label:** Payment date
- Field Name:** Payment_date
- API Name:** Payment_date_c
- Description:** Help Text
- Data Owner:** Data Owner
- Field Usage:** Field Usage
- Data Sensitivity Level:** Data Sensitivity Level
- Compliance Categorization:** Compliance Categorization
- Created By:** Ruhee team17 / 04/03/2025, 12:45 pm
- Modified By:** Ruhee team17 / 04/03/2025, 12:45 pm

The 'Fields & Relationships' sidebar is visible on the left.

Create an Amount Field

1. Select the Data Type as 'Number'.
2. Click 'Next'.
3. Fill in the details as follows:
 1. Field Label: Amount
 2. Length: 18
 3. Field Name: Auto-generated
4. Click 'Next'.
5. Click 'Next' again.
6. Click 'Save & New' to continue adding more fields.

The screenshot shows the Salesforce Setup interface under the Object Manager. A custom field named 'Amount' has been created for the 'Payment for tenantat' object. The field is of type Number and is required. It has an API name of 'Amount__c'. The field label is 'Amount'. The field is owned by 'Refugee team' and was created on 04/03/2025 at 12:47 pm.

Field Label	API Name	Object Name	Data Type
Amount	Amount__c	Payment for tenantat	Number

General Options:

- Required:
- Unique:
- External ID:
- AI Prediction:
- Default Value:

Create a Check for Payment Field

1. Select the Data Type as 'Picklist'.
2. Click 'Next'.
3. Fill in the details as follows:
 1. Field Label: Check for Payment
 2. Field Name: Auto-generated
4. Enter values, with each value separated by a new line:
 1. Paid
 2. Not Paid
5. Click 'Next'.
6. Click 'Next' again.
7. Click 'Save' to complete the process.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Setup, Home, and Object Manager. The main title is "Payment for tenantat". On the left, a sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, and Record Types. The main content area displays a "Custom Field Definition Detail" for a field named "check_for_payment". The field details include:

Field Label	check for payment	Object Name	Payment for tenantat
Field Name	check_for_payment	Data Type	Picklist
API Name	check_for_payment_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Ruheet team17, 04/03/2025, 10:03 pm	Modified By	Ruheet team17, 04/03/2025, 10:03 pm

Below this, under "General Options", there is a "Required" checkbox which is unchecked. Under "Picklist Options", there is a checked checkbox for "Restrict picklist to the values defined in the value set". The "Controlling Field" dropdown is set to "[Next]".

Creation of Lookup Field on Lease Object

1. Go to setup>> click on Object Manager >> type object name(Lease) in the search bar click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select lookup relationship
4. Select the related object “ property” and click next.
5. Field Name : property
6. Field label : Auto generated
7. Click 'Next'.
8. Click 'Next' again.
9. Click 'Save' to complete the process.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Setup, Home, and Object Manager. The main title is 'LEASE > OBJECT MANAGER'. On the left, a sidebar lists various setup categories like Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The central panel displays the 'Custom Field Definition Detail' for a field named 'property'. The 'Field Information' section shows the field label as 'property', field name as 'property', API name as 'property__c', and data type as 'Lookup' (Object Name: lease). The 'Lookup Options' section shows the related object as 'lease'. The 'Validation Rules' section is empty. The 'Help for this Page' link is visible in the top right.

Creation of Lookup Field on Payment Object :

1. Go to setup>> click on Object Manager >> type object name(Payment) in the search bar click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select lookup relationship
4. Select the related object “ Tenant” and click next.
5. Field Name : Tenant
6. Field label : Auto generated
7. Click 'Next'.
8. Click 'Next' again.
9. Click 'Save' to complete the process

The screenshot shows the Salesforce Setup interface for creating a custom field named 'Tenant' on the 'Payment' object. The 'Fields & Relationships' tab is active. The 'Custom Field Definition Detail' section shows the field's properties: Field Label 'Tenant', Field Name 'Tenant', API Name 'Tenant__c', Data Type 'Lookup', Object Name 'Payment', and Child Relationship Name 'Payments'. The 'Related To' field is set to 'tenant'. The 'Required' checkbox is checked. The 'What to do if the lookup record is deleted?' dropdown is set to 'Clear the value of this field'. The 'Edit' button is visible at the top right of the main form.

Creation of Lookup Field on Payment for tenant Object

1. Go to setup>> click on Object Manager >> type object name(Tenant) in the search bar click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Masterdetail relationship
4. Select the related object “ property” and click next.
5. Field Name : property
6. Field label : Auto generated
7. Click 'Next'.
8. Click 'Next' again.
9. Click 'Save' to complete the process

Creation of Validation Rule to a Lease Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Lease object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “lease_end_date”.
4. Insert the Error Condition Formula as :
End_date__c > start_date__c
5. Enter the Error Message as “Your End date must be greater than start date”, select the Error location as Field and select the field as “start date”, and click Save.

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, and Lightning Record Pages. The main content area is titled 'lease Validation Rule' and displays the 'Validation Rule Detail' for rule 'lease_end_date'. The rule details are as follows:

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	Ruheem team17 04/03/2025, 2:33 pm
Modified By	Ruheem team17 04/03/2025, 2:33 pm

The 'Active' checkbox is checked. The right side of the page includes a 'Help for this Page' link and standard browser controls.

Create Email Template

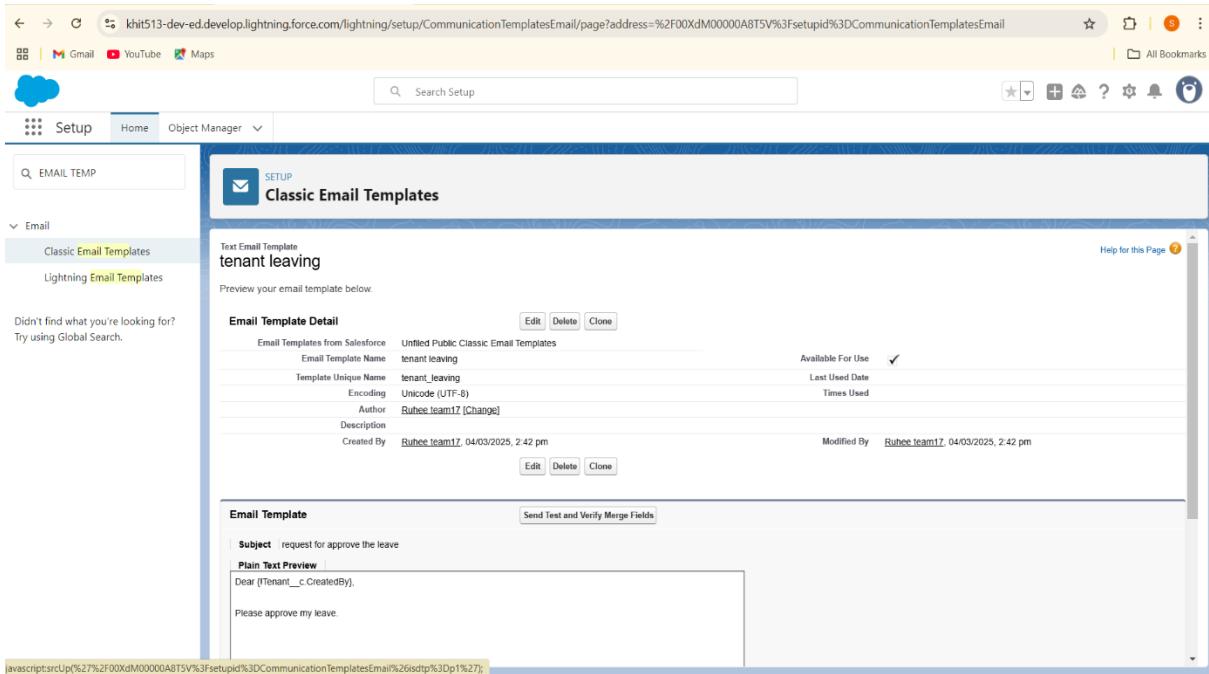
Creation of Email Template For Tenant Leaving

1. Go to setup in quick find box enter email template >> click on classic Email Template.
2. Click on >> New Email Template==>Choose text
3. Folder : Unfiled public Classic Email templates
4. Click on available for use
5. Email Template Name is “tenant leaving”
6. Template Unique Name : Auto populated
7. Subject : ” request for approve the leave”
8. Email body :

Dear {!Tenant__c.CreatedBy},

Please approve my leave

9. Save



Creation of Email Template For Leave Approved

1. Go to setup in quick find box enter email template >> click on classic Email Template.
2. Click on >> New Email Template===>Choose text
3. Folder : Unfiled public Classic Email templates
4. Click on available for use
5. Email Template Name is “Leave approved”
6. Template Unique Name : Auto populated
7. Subject : ” Leave approved”
8. Email body :

dear{!Tenant__c.Name},

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

your leave is approved. You can leave now

9. Save

The screenshot shows the Salesforce Setup interface. In the left sidebar, under the 'Email' section, 'Classic Email Templates' is selected. The main area displays the 'Text Email Template' page for 'Leave approved'. The template details include:

- Email Template Name: Leave approved
- Template Unique Name: Leave_approved
- Encoding: Unicode (UTF-8)
- Author: Ruhee team17 [Change]
- Description: Created By Ruhee team17 04/03/2025, 2:44 pm
- Modified By Ruhee team17 04/03/2025, 2:44 pm
- Available For Use: checked
- Last Used Date: (empty)
- Times Used: (empty)

The email template body contains the following text:

```

Subject : Leave approved
Plain Text Preview :
dear(!Tenant__c.Name),

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

your leave is approved. You can leave now

```

Creation of Email Template For rejection for leave

1. Go to setup in quick find box enter email template >> click on classic Email Template.
2. Click on >>New Email Template==>Choose text
3. Folder : Unfiled public Classic Email templates
4. Click on available for use
5. Email Template Name is “Leave rejected”
6. Template Unique Name : Auto populated
7. Subject : ” Leave rejected”
8. Email body :

Dear {!Tenant__c.Name},

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

your leave has rejected

9. Save

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

- Page Title:** Classic Email Templates
- Template Selected:** Leave rejected
- Email Template Detail:**
 - Email Template Name: Leave rejected
 - Template Unique Name: Leave_rejected
 - Encoding: Unicode (UTF-8)
 - Author: Ruhee team17 [Change]
 - Created By: Ruhee team17, 04/03/2025, 2:46 pm
 - Modified By: Ruhee team17, 04/03/2025, 2:46 pm
- Email Template Preview:**
 - Subject: Leave rejected
 - Plain Text Preview:

Dear {!Tenant__c.Name},
I hope this email finds you well. Your contract has not ended. So we can't approve your leave
your leave has rejected

Create Email Template For Monthly payment

1. Email Template Name is “Tenant Email”
2. Template Unique Name : Auto populated
3. Subject : ” Urgent: Monthly Rent Payment Reminder”
4. Email body :

Dear {!Tenant__c.Name},

I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding. As outlined in our rental agreement, the payment is due. To ensure the smooth operation of our property management and to avoid any inconvenience, we kindly request you to settle the payment at your earliest convenience.

5. Save

The screenshot shows the Salesforce Setup interface for managing email templates. The top navigation bar includes links for Setup, Home, Object Manager, and a search bar labeled 'Search Setup'. Below the header, there's a sidebar with a 'Email' section containing 'Classic Email Templates' (which is selected) and 'Lightning Email Templates'. A message says ' Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Classic Email Templates' and shows a specific template named 'Tenant Email'. The 'Email Template Detail' section displays the following information:

Email Templates from Salesforce	Unfiled Public Classic Email Templates
Email Template Name	Tenant Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Rubhee team17 [Change]
Description	
Created By	Rubhee team17 04/03/2025, 2:48 pm
Modified By	Rubhee team17 04/03/2025, 2:48 pm

Below the detail section is a preview pane titled 'Email Template' with a 'Plain Text Preview' button and a 'Send Test and Verify Merge Fields' button. The preview text is as follows:

Subject: Urgent: Monthly Rent Payment Reminder
Plain Text Preview:
 Dear {!Tenant__c.Name},
 I trust this email finds you well. We appreciate your continued tenancy at our property and hope you have been comfortable in your residence.
 This communication is a friendly reminder regarding your monthly rent payment, which is currently outstanding. As outlined in our rental agreement, the payment is due. To ensure the smooth operation of our property management and to avoid any inconvenience, we kindly request you to settle the payment at your earliest convenience.

Creation of Email Template For successful payment

1. Email Template Name is “tenant payment”
2. Template Unique Name : Auto populated
3. Subject : ” Confirmation of Successful Monthly Payment”
4. Email body :

Dear {!Tenant__c.Email__c},

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

5. Save

The screenshot shows the Salesforce Setup interface for managing email templates. The page title is "Classic Email Templates". A specific template named "tenant payment" is selected. The template details include:

- Email Template Detail:**
 - Email Templates from Salesforce: Unfiled Public Classic Email Templates
 - Email Template Name: tenant payment
 - Template Unique Name: tenant_payment
 - Encoding: Unicode (UTF-8)
 - Author: Ruhee.team17 [Change]
 - Description: Created By Ruhee.team17, 04/03/2025, 3:21 pm
 - Available For Use: checked
 - Last Used Date: Not specified
 - Times Used: Not specified
 - Modified By: Ruhee.team17, 04/03/2025, 3:21 pm
- Email Template:**
 - Subject: Confirmation of Successful Monthly Payment
 - Plain Text Preview: Dear {Tenant__c_Email__c}, We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

Create Approval Process

Create Approval Process For check for vacant

1. Go to setup >> Approval Processes in quick find bar>>click on it.
2. Manage Approval Process For >> “Tenant” from the drop down.
3. Click on “Create New Approval Process” >> Use standard setup wizard.
4. Process Name “check for vacant” >> Click Next.
5. Field “Tenant:status” >> Operator : Not equals , Value >> Click on the lookup filter icon and select “Leaving”.
6. Click insert field,then click Next.
7. Next Automated Approver determined by “None” from the drop down.
8. Select the “Administrators ONLY can edit records during the approval process”.Then Next.
9. Click on next leave the email template click on next
10. From the available fields select >> Tenant Name, and then add >>Add it to the selected.Then Next.
 - Make sure Display approver history is checked.

- And under security settings check the “Allow approvers to access the approval page only from within the Salesforce application. (Recommended)” option.
- Click on “i’ll do this later. Take me back to the listing of all approval process for this object”
- Click go

11. Submitter type Search>>Owner, Allowed Submitters>>Property Owner. Then Next.

- Then click save.

The screenshot shows the Salesforce Setup interface for 'Approval Processes'. A process named 'check for vacant' is selected. The 'Process Definition Detail' section includes fields like Process Name (check for vacant), Unique Name (check_for_vacant), Description (Tenant: check for vacant), Entry Criteria (Tenant: status NOT EQUAL TO Leaving), Record Editability (Administrator ONLY), and a checkbox for 'Allow Submitters to Recall Approval Requests'. The 'Initial Submission Actions' section contains an action 'Record Lock' with a description 'Lock the record from being edited' and 'please approve my leave'. The 'Approval Steps' section lists one step: 'Show Actions | Edit 1 Approval Process'.

Initial Submission Action:

1. Under initial submission action click on add new and then select email alert.
2. Description: “please approve my leave”.
3. unique name : auto populated

4. Email template : tenant leaving
5. Recipient type : Email field
6. Available Recipients : Email field : Email
7. From Email address : Current user's email
8. Click save

The screenshot shows the Salesforce Setup interface with the URL khit513-dev-ed.lightning.force.com/lightning/setup/ApprovalProcesses/page?address=%2F04adM00000fMNx. The left sidebar includes links for Data, Feature Settings, Process Automation, and Approval Processes (which is selected). The main content area is titled "Approval Processes". It contains four sections:

- Initial Submission Actions**: Contains two entries: "Record Lock" (Type: Record Lock) with description "Lock the record from being edited" and "Email Alert" (Type: Email Alert) with description "please_approve_my_leave".
- Approval Steps**: Shows one step named "Approval Process". The "Assigned Approver" is "User:Ruhee team17" and the "Reject Behavior" is "Final Rejection".
- Final Approval Actions**: Contains one entry: "Record Lock" (Type: Record Lock) with description "Lock the record from being edited" and "Email Alert" (Type: Email Alert) with description "Tenant leaving".
- Final Rejection Actions**: Contains one entry: "Record Lock" (Type: Record Lock) with description "Unlock the record for editing" and "Email Alert" (Type: Email Alert) with description "your request for leave is rejected".

Final Approval Action

1. Under Final approval action click on new and then select email alert.
2. Description: “Tenant leaving”.
3. unique name : auto populated
4. Email template : Leave approved
5. Recipient type : Email field
6. Available Recipients : Email field : Email
7. From Email address : Current user's email
8. Click save

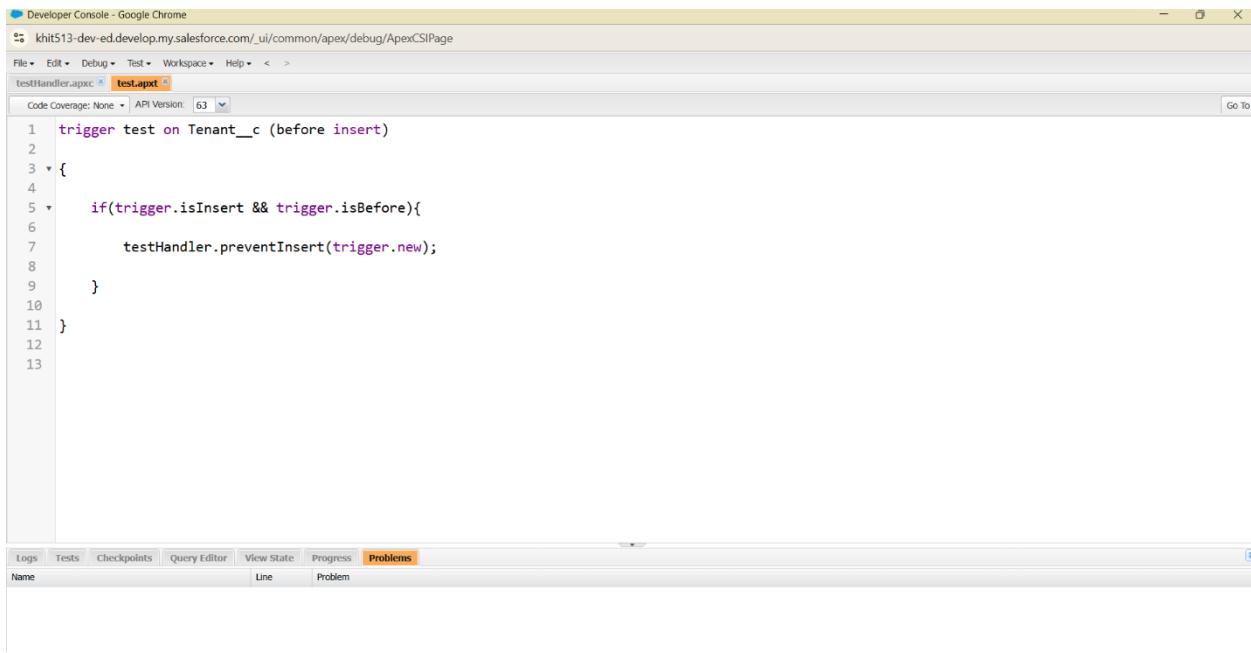
Final Rejection Action

1. Under final rejection action click on add new and then select email alert.
2. Description: “your request for leave is rejected”.
3. unique name : auto populated
4. Email template : leave rejected
5. Recipient type : Email field
6. Available Recipients : Email field : Email
7. From Email address : Current user’s email
8. Click save

Create an Apex Trigger

To create a new Apex Class follow the below steps:

- Click on the file >> New ? Apex Class.
- Give the Apex Trigger name as “test”, and select “Tenant__c” from the dropdown for sObject.
- Click Submit.
- Now write the code logic here



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is khit513-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar shows 'testHandler.apxc' and 'testLapac'. The code editor contains the following Apex trigger:

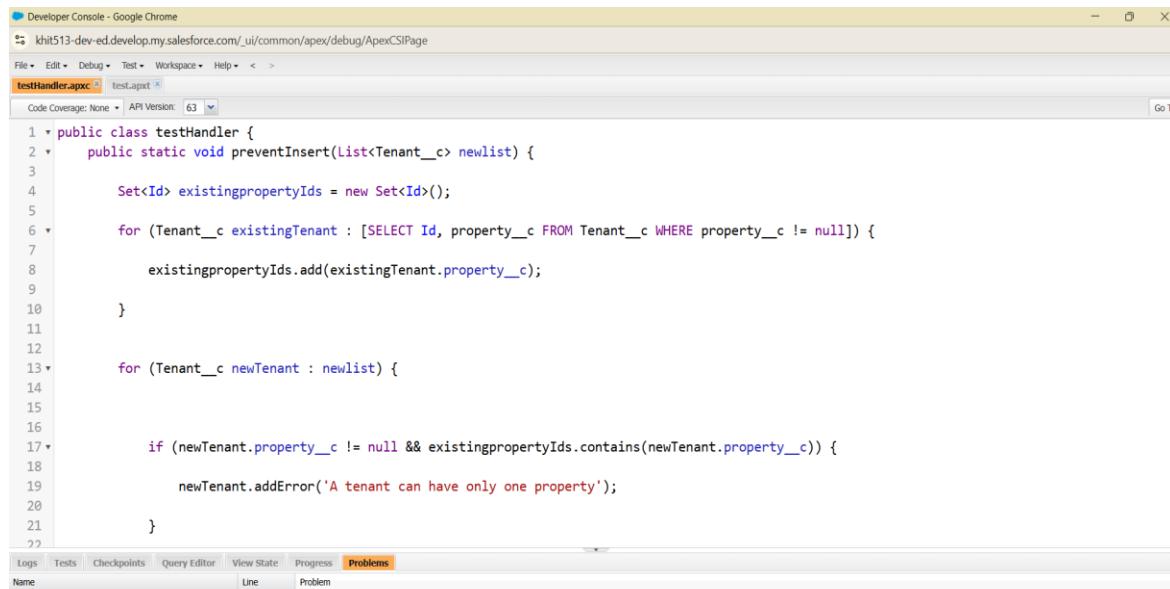
```
trigger test on Tenant__c (before insert)
{
    if(trigger.isInsert && trigger.isBefore){
        testHandler.preventInsert(trigger.new);
    }
}
```

The 'Problems' tab is selected in the bottom navigation bar.

Creation of an Apex Handler class

To create a new Apex Class follow the below steps:

- Click on the file >> New >>Apex Class.
- Enter class name as testHandler

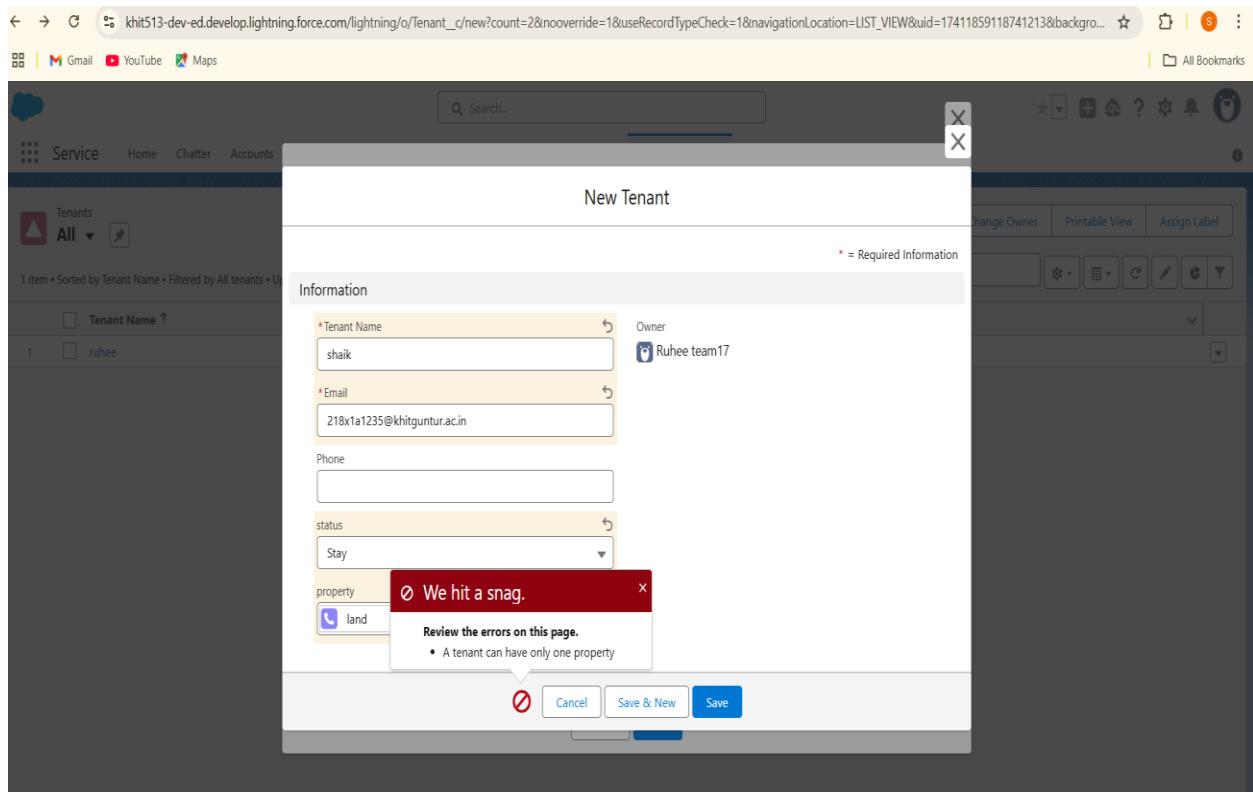


The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is khit513-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The tab bar shows 'testHandler.apxc' and 'testLapac'. The code editor contains the following Apex class:

```
public class testHandler {
    public static void preventInsert(List<Tenant__c> newList) {
        Set<Id> existingPropertyIds = new Set<Id>();
        for (Tenant__c existingTenant : [SELECT Id, property__c FROM Tenant__c WHERE property__c != null]) {
            existingPropertyIds.add(existingTenant.property__c);
        }
        for (Tenant__c newTenant : newList) {
            if (newTenant.property__c != null && existingPropertyIds.contains(newTenant.property__c)) {
                newTenantaddError('A tenant can have only one property');
            }
        }
    }
}
```

The 'Problems' tab is selected in the bottom navigation bar.

Testing the Trigger



Create Flow

Creation Flow for monthly payment

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.
2. Select the record Triggered flow. Click on create.
3. Under Object select "Payment for tenant". Click on A record is updated.
4. Set Entry Conditions
 - a. Under Condition Requirements
 - b. All Conditions are met

Field: check_for_payment__c	Operator: Equals	Value : paid
--------------------------------	---------------------	--------------

5. Click on : Every time a record is updated and meets the condition requirements
6. Click on : Actions and related records,done
7. Under record trigger flow click on “+” icon and select action
8. Label : send email
 - a. API Name : send_email
9. Label : send email
 - a. API Name : send_email
10. Enable Body
11. Click on new resource
12. Under resource type select “Text Template”
13. API Name : emailbody

Under body

Dear {!\$Record.Tenant__r.Name},

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

Click Done.

14. Enable recipient Address List { !\$Record.Tenant__r.Email__c }

15. Click Done

16. Enable subject

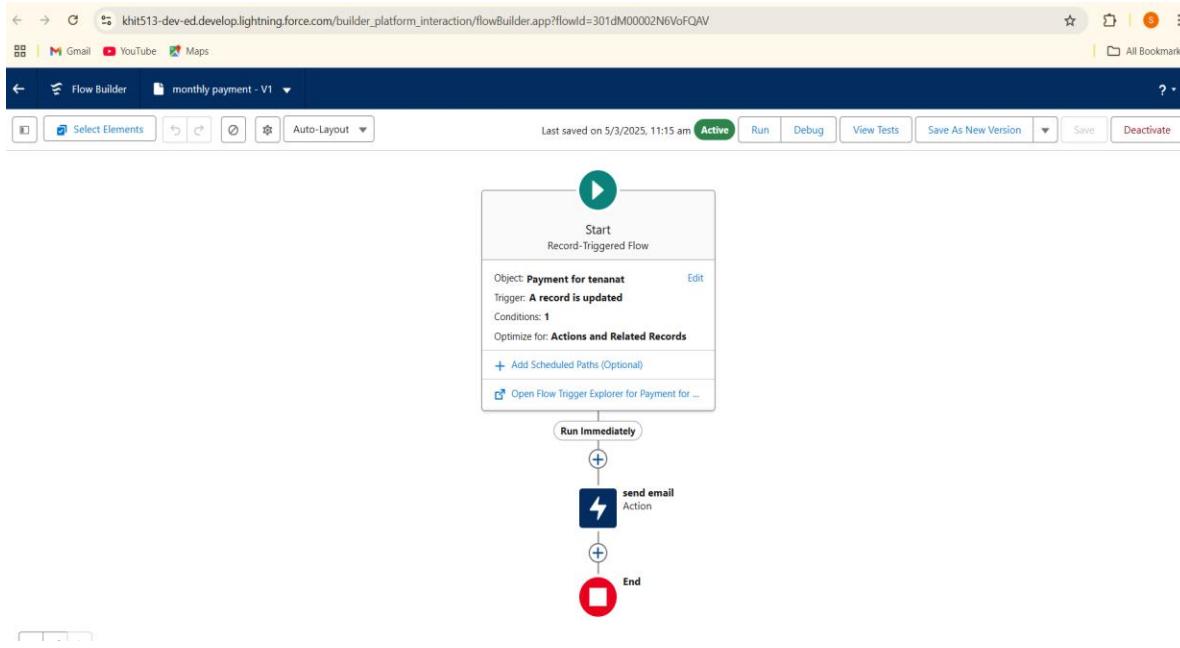
17. Confirmation of Successful Monthly Payment

18. Click on save

Flow label : monthly payment

Flow API Name : monthly_payment

19. Click on activate



Creation of Schedule Class

Create an Apex Class

1. To create a new Apex Class follow the below steps:

Click on the file >> New >> Apex Class.

2. Enter class name as MonthlyEmailScheduler

Apex logic:

```

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});
}
}
}

```

Save the code.

Schedule Apex class

1. Enter Apex class in quick find box
2. Select schedule Apex
3. Enter job Name : MonthlyEmailScheduler
4. Apex class : MonthlyEmailScheduler
5. Frequency : Monthly==>select on day 1
6. Start date : 04/12/2023
7. End date : 04/01/2024
8. Preferred start time : 09:00 am
9. save

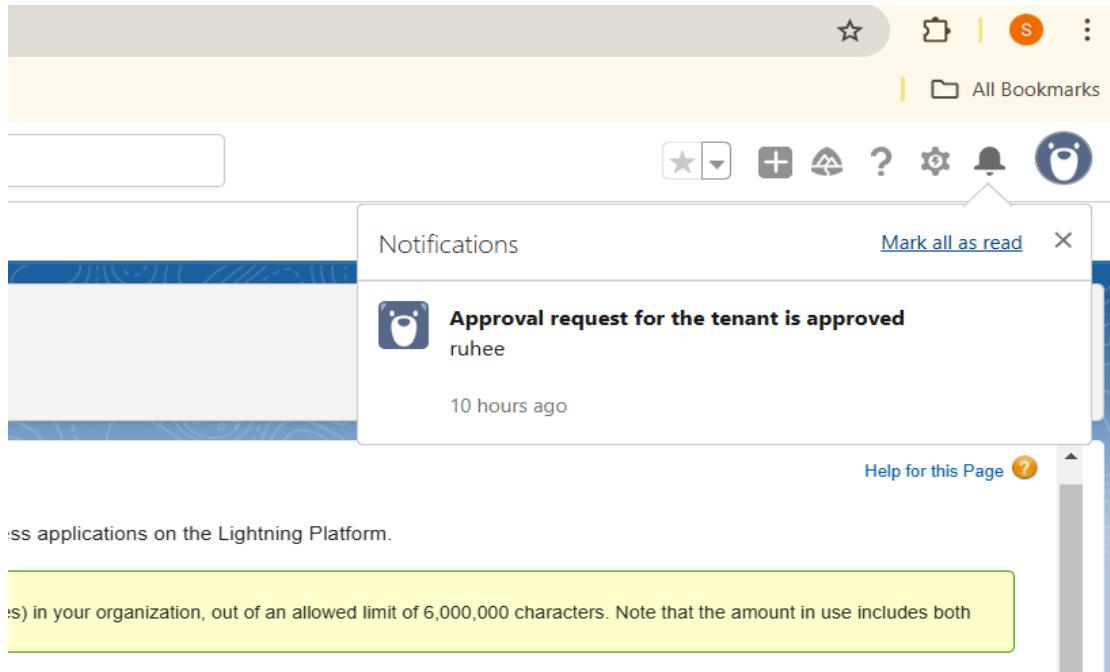
Testing the approval process

Enter any comment and click on submit

Click on that notification

click on approve

Give any comment and submit



Use Case Scenarios in Daily life

Tenant Onboarding & Lease Agreement: A property management company uses Salesforce to track new tenant applications. By storing tenant details and lease documents, it automates lease approval workflows with e-signatures.

Rent Collection & Payment Tracking: A landlord wants to track rent payments and send automated reminders through emails. It integrates with payment gateways for online rent collection.

Lease Expiry & Renewal Management: A lease is about to expire, and the tenant needs to renew or vacate. Sends automated notifications for lease renewal deadlines. Tracks lease status and allows tenants to renew online.

Challenges and Solutions

Challenge: Difficulty in managing late payments and penalties.

Solution : Set up payment reminder via email.

Challenge: Inefficient tracking of tenant issues.

Solution: Use Salesforce Management to automate ticket tracking and assignments.

Challenge: Manual tracking of lease expirations leads to missed renewals.

Solution: Use Salesforce Flow to streamline renewal approvals. Set up automated reminders and notifications for lease renewals.

Future Recommendations

AI-driven : Use Salesforce Einstein Analytics for AI-driven insights.

Mobile Access: Deploy Salesforce Mobile App for lease and tenant management.

Conclusion:

The Salesforce based lease management system successfully optimizes the lease management process by providing a structured, automated scheduling system. By leveraging Salesforce, automated notifications, and real-time scheduling, making lease management more efficient and user-friendly.

