

OXFORD ENGINEERING COLLEGE

Department of B.TECH (INFORMATION TECHNOLOGY)

PROJECT TITLE:

LEASE MANAGEMENT SYSTEM

Submitted by (Team Members):

- 1.Arulselvan.S(Team Leader)
- 2.Balamurugan.P
- 3.Balamurugan.S
- 4.Santhosh kumar.S

Guided by:

Mrs. Sweetha.V

Platform: Salesforce Developer

Date of Submission: 05/11/2025

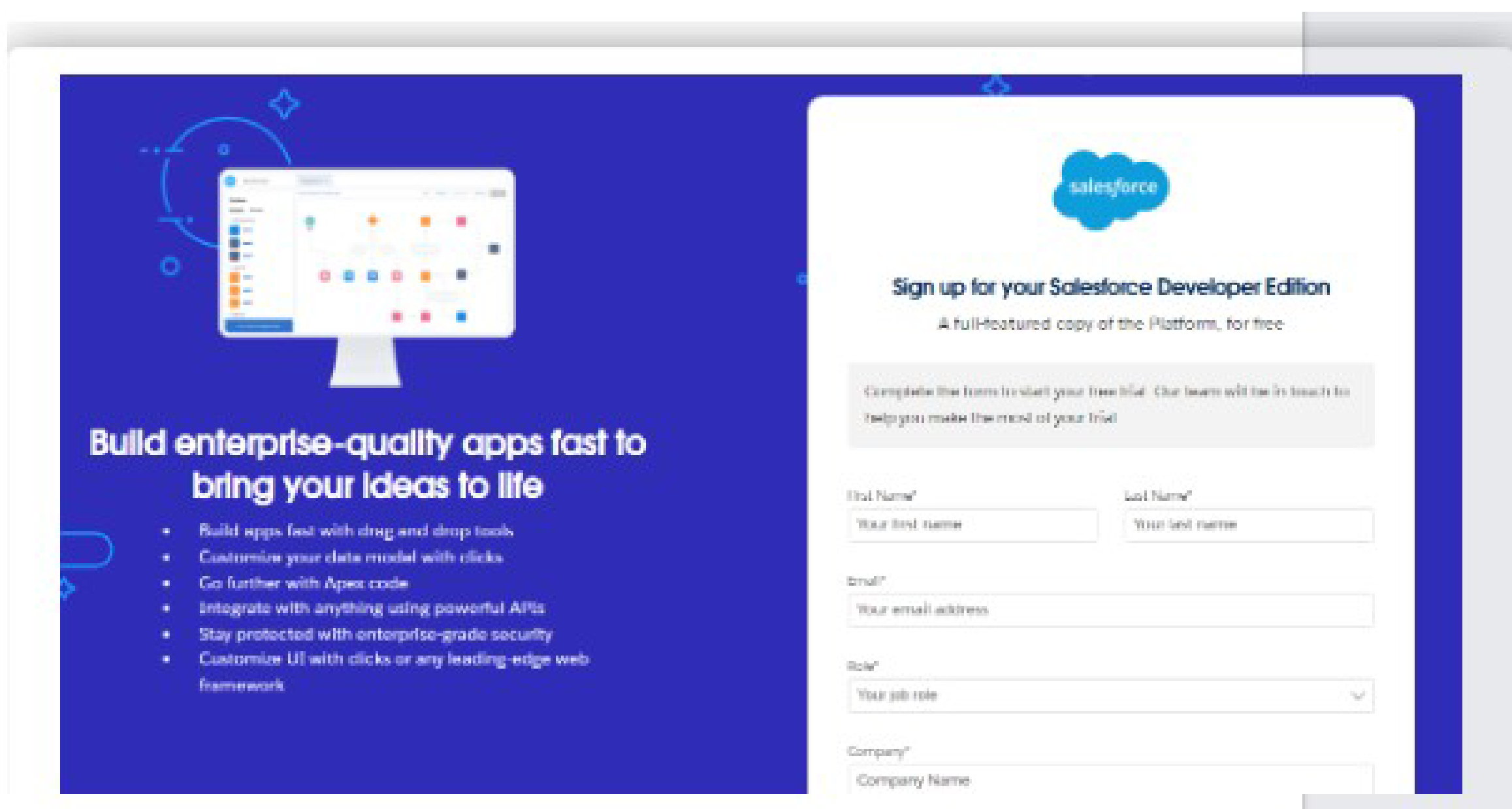
Project Description

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

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

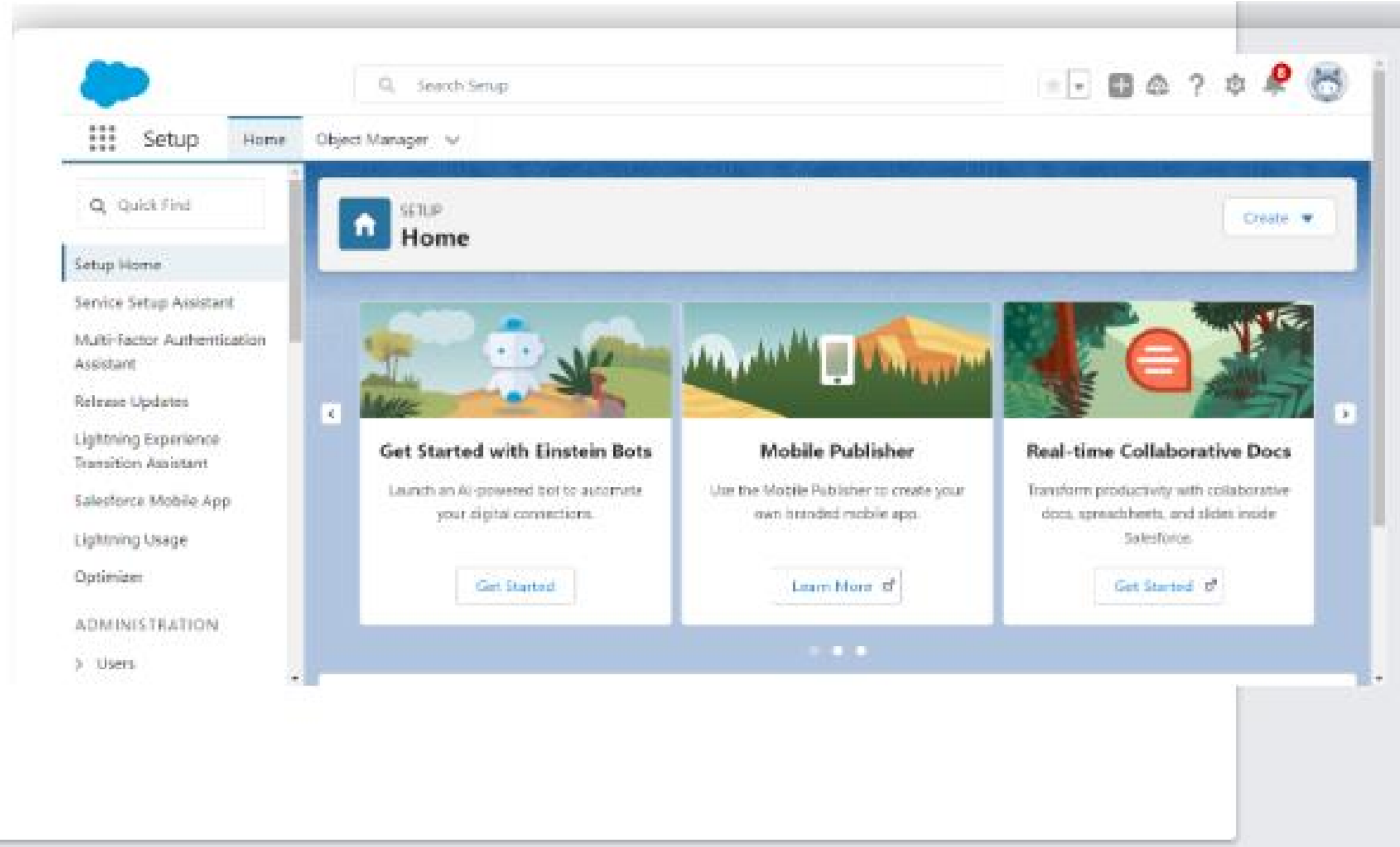
Creating Developer Account

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



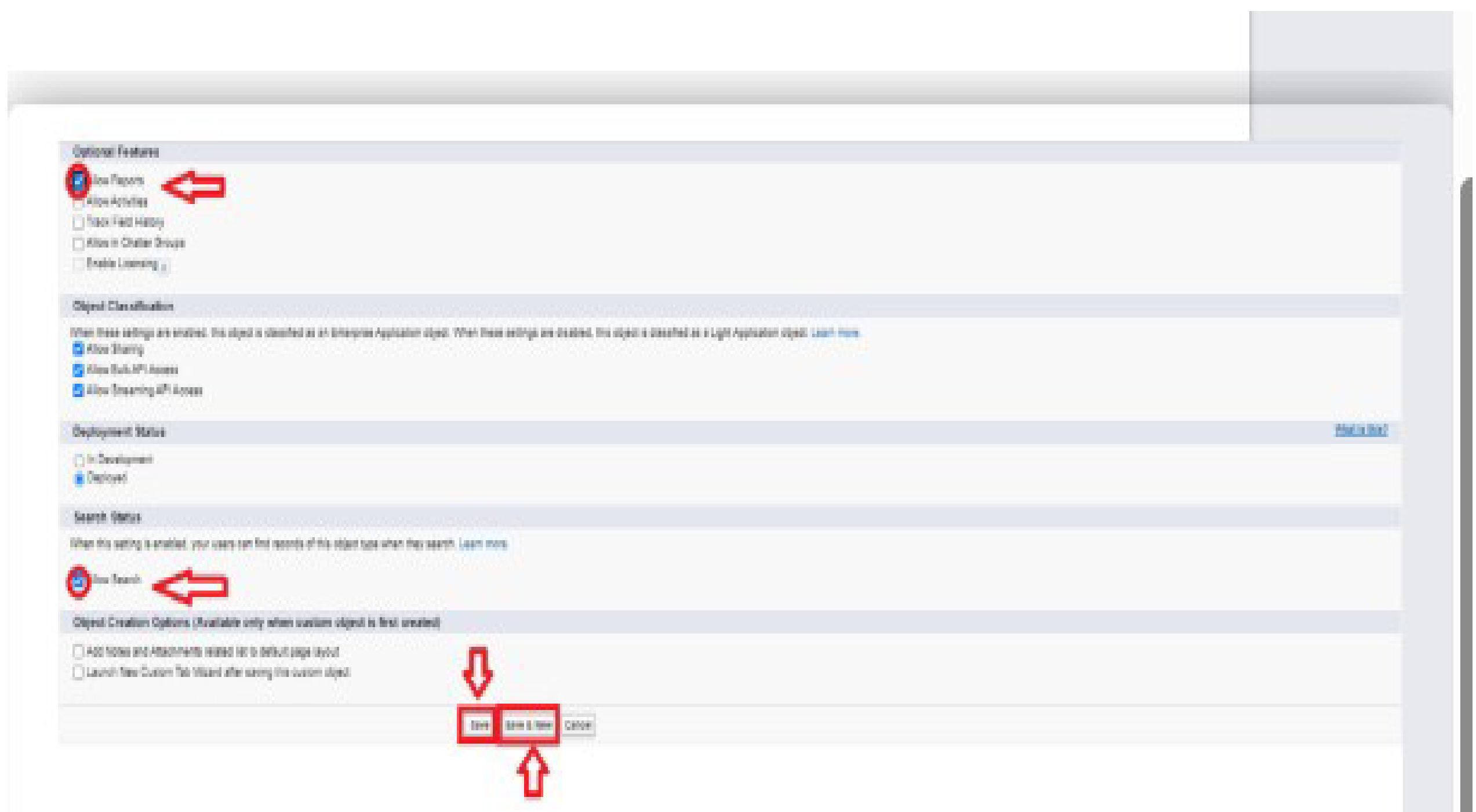
Account Activation

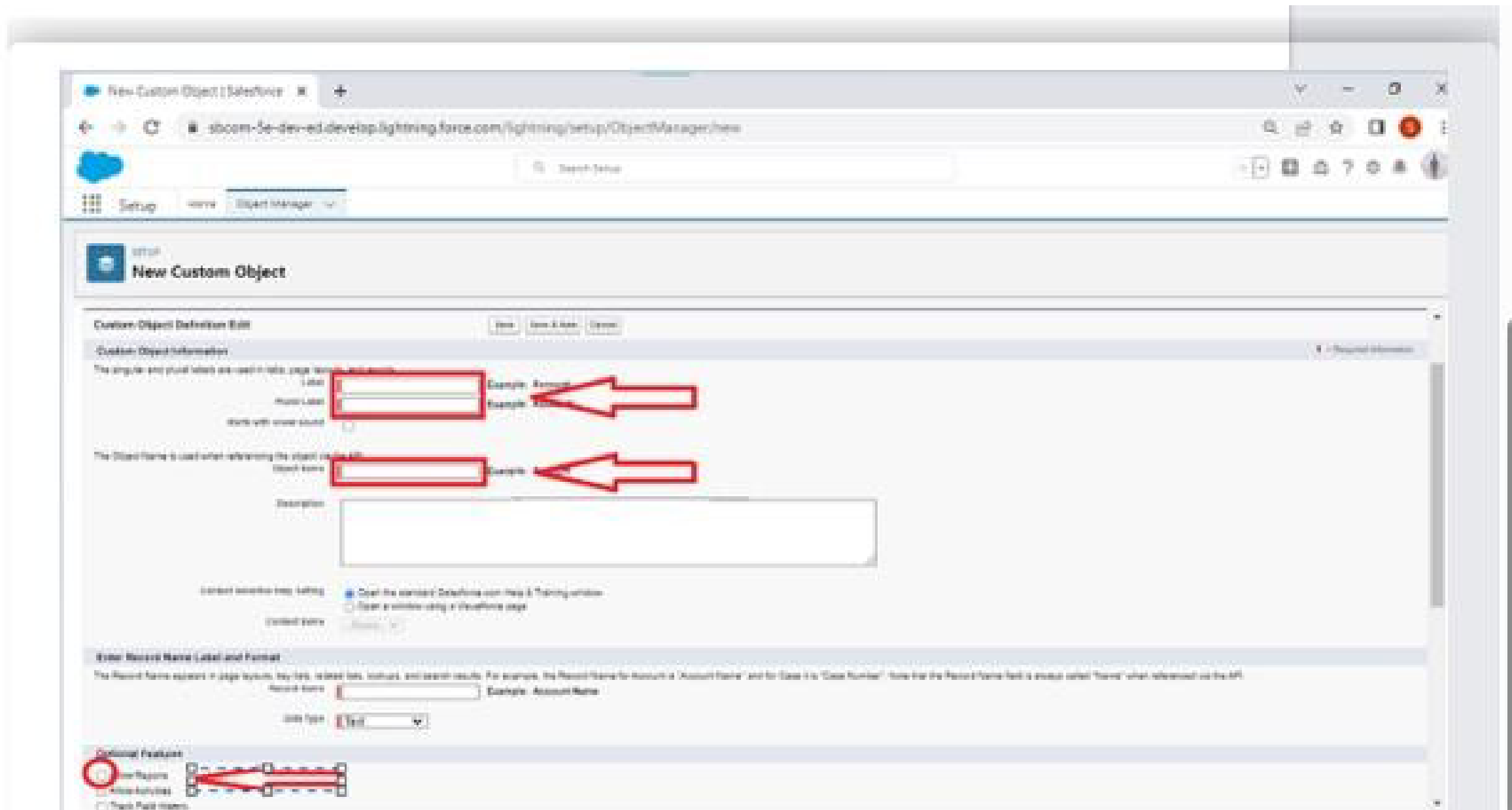
Activate the Salesforce Developer Account by verifying the registered email and completing the activation process.



Create Property Object

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





Create Tenant Object

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

Create Payment Object

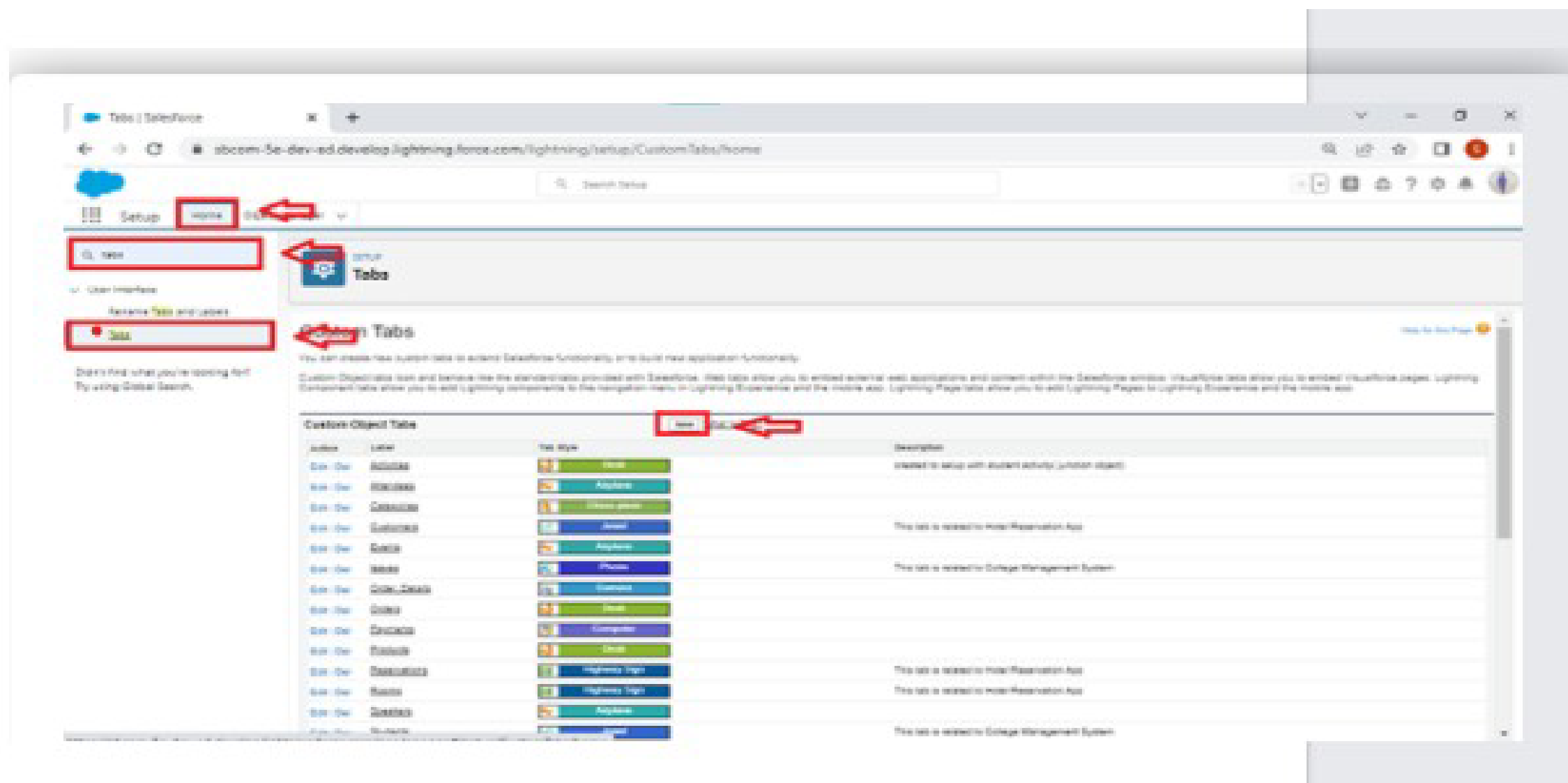
Design a “ Payment” object to handle rent payments, due dates, and payment status tracking.

Create Lease Object

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

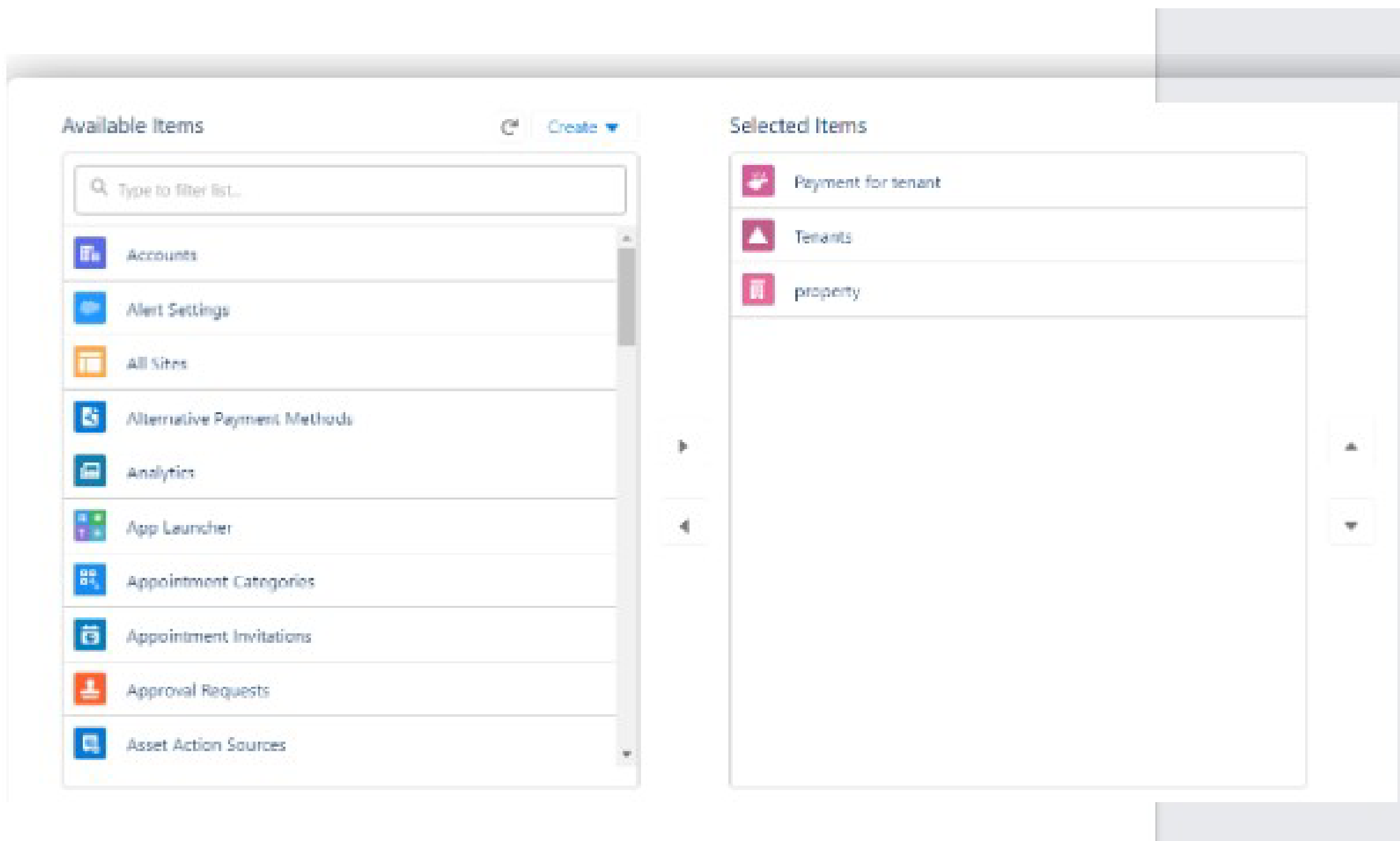
Creating a Custom Tab

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



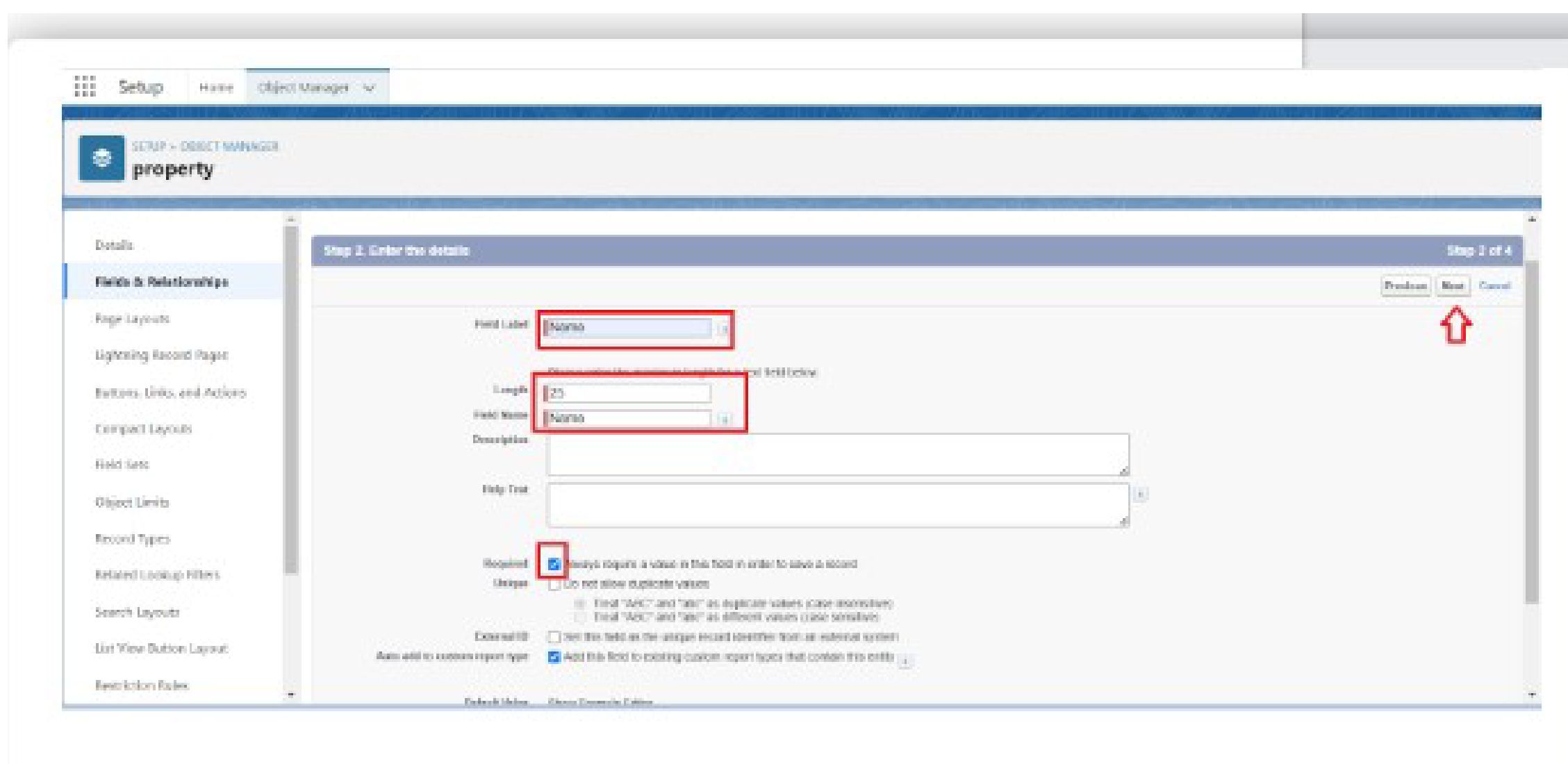
Lightning App Creation

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



Fields Creation

In Salesforce, **Fields** are used to store different types of data inside each **Object**. Each object in the **Lease Management System** has specific fields that help to manage properties, tenants, leases, and payments efficiently.



Email Template

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

1. Tenant Leaving Notification

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

2. Lease Approved

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

3. Lease Rejection Email

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

4 . Monthly Payment Reminder

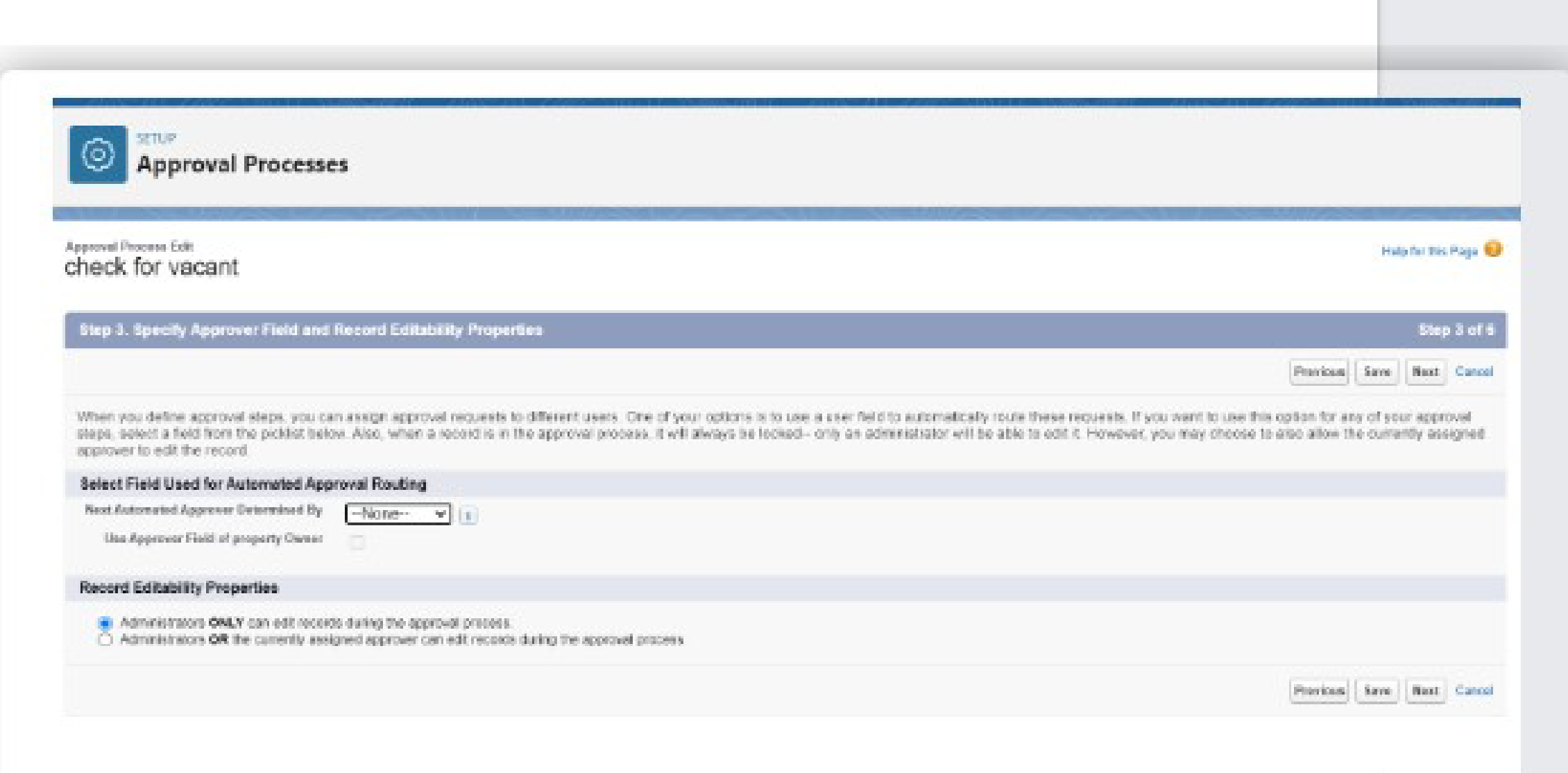
Automatically reminds the tenant each month about the upcoming rent payment due date.

5. Successful Payment Confirmation

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

Approval Process

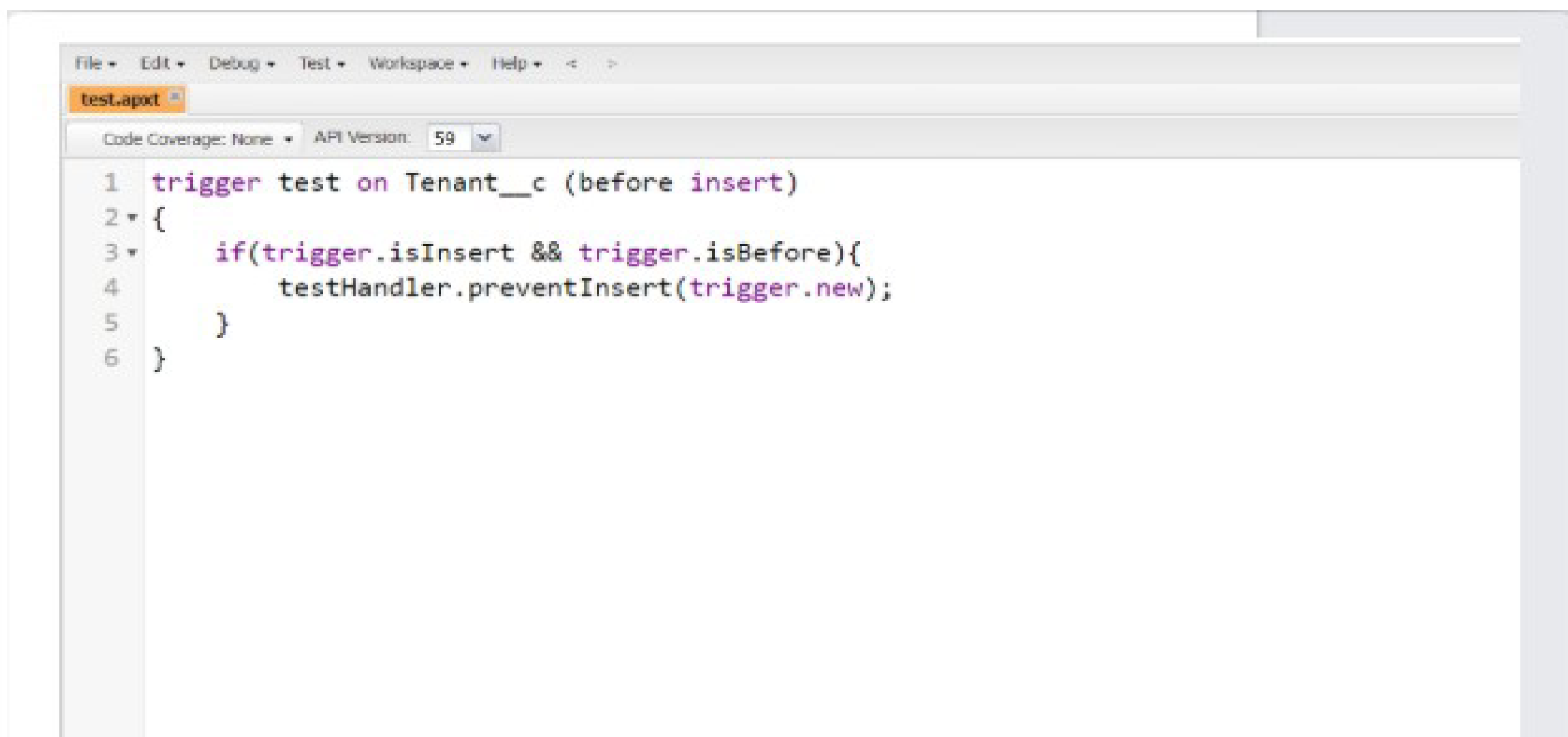
An **Approval Process** automates how records are approved in Salesforce. In this project, an approval process is created for the **Lease or Payment** object to ensure that certain records (for example, rent amount > ₹ 50,000) need admin or manager approval before final confirmation.



The screenshot shows the Salesforce Setup interface for configuring an approval process. The page title is "Approval Processes" under the "SETUP" menu. The specific process being edited is "check for vacant". The current step is "Step 3. Specify Approver Field and Record Editability Properties", which is the third of four steps. The interface includes navigation buttons: "Previous", "Save", "Next", and "Cancel". A help link "Help for this Page" is also present. The main content area contains instructions on assigning approval requests to different users and selecting a field for automated approval routing. A dropdown menu for "Next Automated Approver Determined By" is set to "--None--". There is a checkbox for "Use Approver Field of property Owner" which is currently unchecked. Below this, the "Record Editability Properties" section has two radio button options: "Administrators ONLY can edit records during the approval process" (which is selected) and "Administrators OR the currently assigned approver can edit records during the approval process".

Apex Trigger

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



Apex logic:

:

```
public class testHandler {
    public static void preventInsert(List<Tenant__c> newList) {
        Set<Id> existingPropertyIds = new Set<Id>();
        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE
Property__c != null]) {
            existingPropertyIds.add(existingTenant.Property__c);
        }

        for (Tenant__c newTenant : newList) {

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

Testing the Trigger:

The screenshot shows a web form titled "New Tenant" with a close button (X) in the top right corner. A legend indicates that a red asterisk (*) denotes "Required Information". The form contains the following fields:

- Information** (Section Header)
- Tenant Name** (Required): A text input field containing "niranjan".
- Phone**: An empty text input field.
- Email**: An empty text input field.
- status**: A dropdown menu with "stay" selected.
- property** (Required): A dropdown menu showing "Manne R".

A red error message box is overlaid on the form, stating: "We hit a snag. Review the errors on this page. A tenant can have only one property". At the bottom of the form, there are three buttons: "Cancel", "Save & New", and "Save". A red prohibition icon is located to the left of the "Cancel" button.

Flow Creation


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

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

2

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

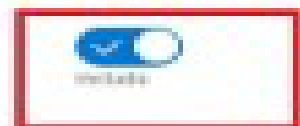
All Conditions Are Met (AND)

When to Run the Flow for Updated Records 

☒ Every time a record is updated and meets the condition requirements

☐ Only when a record is updated to meet the condition requirements

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



A **Scheduled Apex Class** is used to run Apex code automatically at a specific time (daily, weekly, monthly).
For example — sending payment reminders, lease expiry alerts, or autoupdating records every morning.

reduces human error, and provides a digital solution for real-time property management.