

PROJECT TITLE

College Name: Nandha arts and science college

College Code: BRU4J

TEAM ID:

TEAM MEMBERS:

Team LeaderName: Manimegalai D

Email: manimegalaits3257@gmail.com

Team Member1: Shasmathi B

Email: Shasmathi111@gmail.com

Team Member: Dhanusri P

Email: Dhanusri2830@gmail.com

1.INTRODUCTION

1.1ProjectOverview

TheLease ManagementSystem is aSalesforce-based application designed to streamline the processes associated with leasing real estate properties. It handles tenant management, lease

contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.



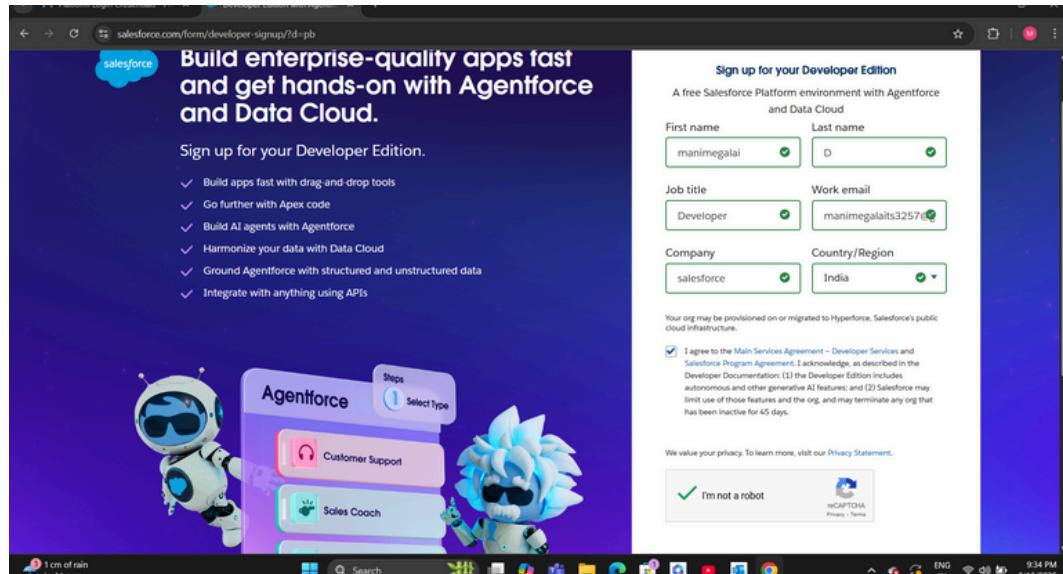
1.2 Purpose

The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

DEVELOPMENT PHASE

Creating Developer Account:

By using this URL - <https://www.salesforce.com/form/developer-signup/?d=pb>



- Created objects: Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface for the Object Manager. The left sidebar lists various configuration options: 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, and Sourcing Rules. The main content area is titled "Details" for the "Tenant" object. It shows the API Name as "Tenant__c" and the Singular Label as "Tenant". On the right, there are sections for "Enable Reports", "Track Activities", "Track Field History", "Deployment Status", and "Help Settings". The status is listed as "Deployed" and the help setting as "Standard salesforce.com Help Window".

The screenshot shows the Salesforce Setup interface for the Object Manager. The left sidebar lists the same configuration options as the previous screenshot. The main content area is titled "Details" for the "lease" object. It shows the API Name as "lease__c" and the Singular Label as "lease". On the right, there are sections for "Enable Reports", "Track Activities", "Track Field History", "Deployment Status", and "Help Settings". The status is listed as "Deployed" and the help setting as "Standard salesforce.com Help Window".

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The search bar says 'Search Setup'. The main title is 'SETUP > OBJECT MANAGER' followed by 'Payment for tenantat'. On the left, a sidebar lists various configuration options under 'Details'. The main content area has two tabs: 'Details' (selected) and 'Fields & Relationships'. The 'Details' tab contains fields for 'API Name' (Payment_for_tenantat_c), 'Custom' (checked), 'Singular Label' (Payment for tenantat), 'Plural Label' (Payments), and 'Description'. To the right, there are sections for 'Unfiled Reports' (Trunk Activities, Trunk Field History, Deployment Status, Help Settings, Standard Salesforce.com Help Window, all checked), 'Edit' and 'Delete' buttons, and a 'Standard Salesforce.com Help Window' link.

- Configured fields and relationships

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The search bar says 'Search Setup'. The main title is 'SETUP > OBJECT MANAGER' followed by 'property'. On the left, a sidebar lists various configuration options under 'Details' and 'Fields & Relationships'. The 'Fields & Relationships' tab is selected, showing a table titled 'Fields & Relationships' with 9 items. The table 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
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	property_c	Lookup(property)		
property Name	Name	Text(30)		
sft	sft_c	Text(18)		
Type	Type_c	Picklist		

Setup Home Object Manager

SETUP > OBJECT MANAGER
Payment for tenantat

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

Fields & Relationships
7 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)		
Owner	OwnerId	Lookup(User,Group)		
Payment date	Payment_date_c	Date		
Payment Name	Name	Text(80)		

Setup Home Object Manager

SETUP > OBJECT MANAGER
lease

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 Scoping Rules

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		

Fields & Relationships
7 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		
status	status__c	Picklist		
Tenant Name	Name	Text(80)		✓

- Developed Lightning App with relevant tabs

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: Lease Management

*Developer Name: Lease_Management

App Branding

Image:

Primary Color Hex Value: #0070D2

Description: Application to efficiently handle the processes related to leasing real estate properties.

Org Theme Options: Use the app's image and color instead of the org's custom theme.

App Launcher Preview

[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

App Settings

[App Details & Branding](#)

[App Options](#)

[Utility Items \(Desktop Only\)](#)

Navigation Items

[User Profiles](#)

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

Type to filter list... [Create](#)

- Accounts
- Activation Targets
- Activations
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests

Selected Items

- Payment
- Tenants
- property
- lease

[Lightning App Builder](#) [App Settings](#) [Pages](#) [Lease Management](#) [Help](#)

App Settings

[App Details & Branding](#)

[App Options](#)

[Utility Items \(Desktop Only\)](#)

User Profiles

Choose the user profiles that can access this app.

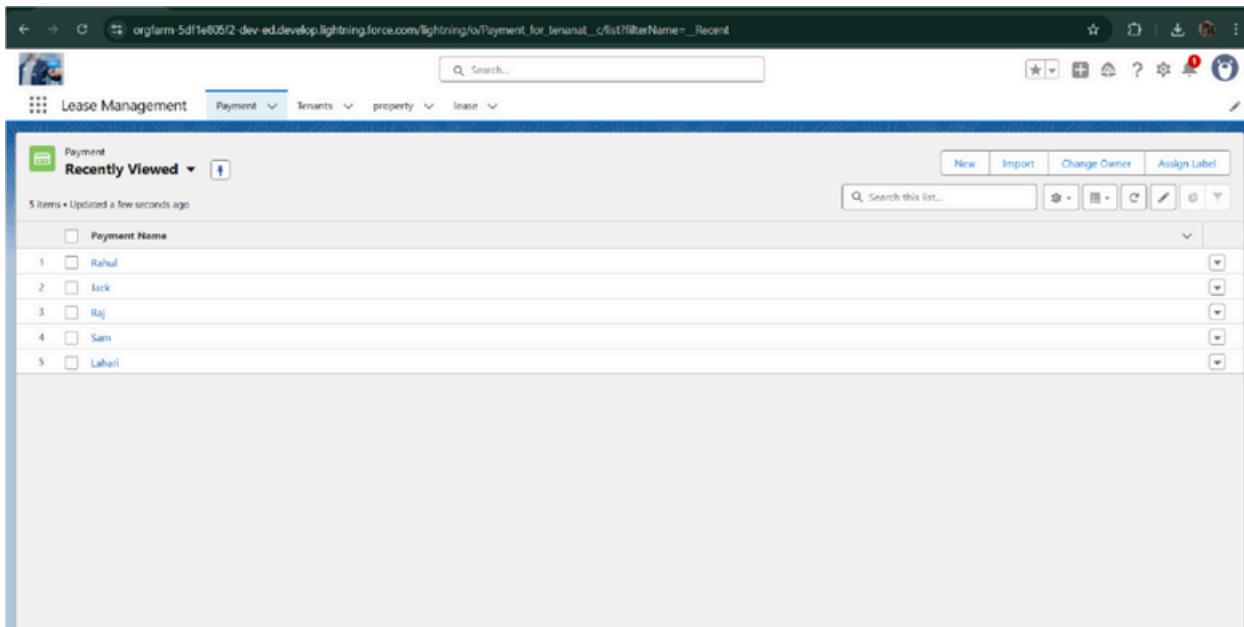
Available Profiles

Type to filter list...

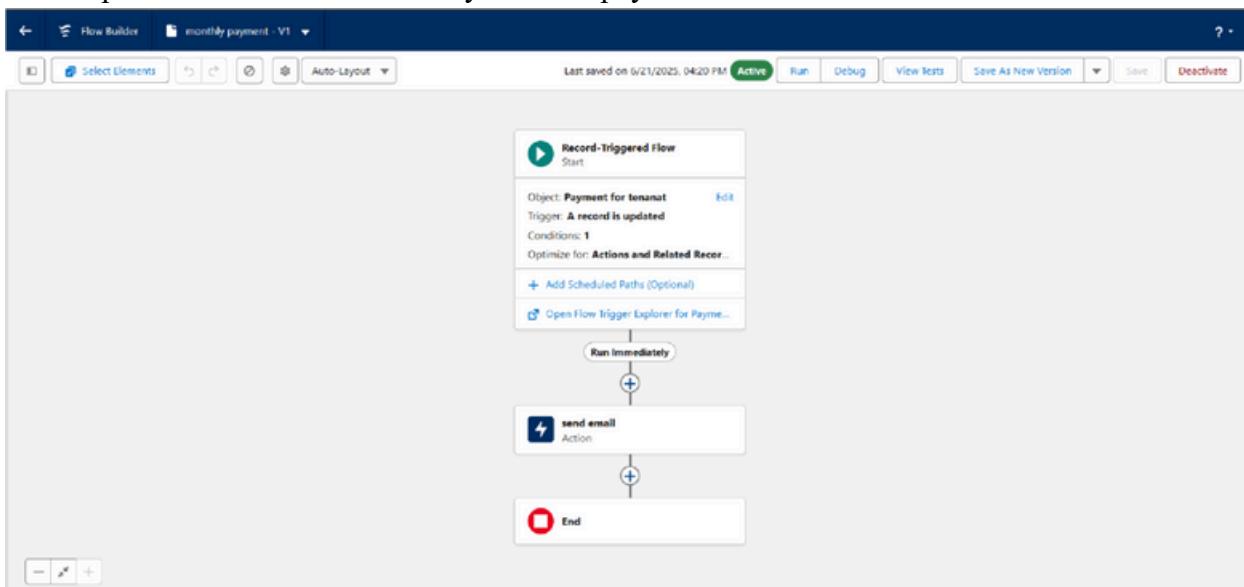
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

Selected Profiles

- System Administrator



- Implemented Flows for monthly rent and payment success



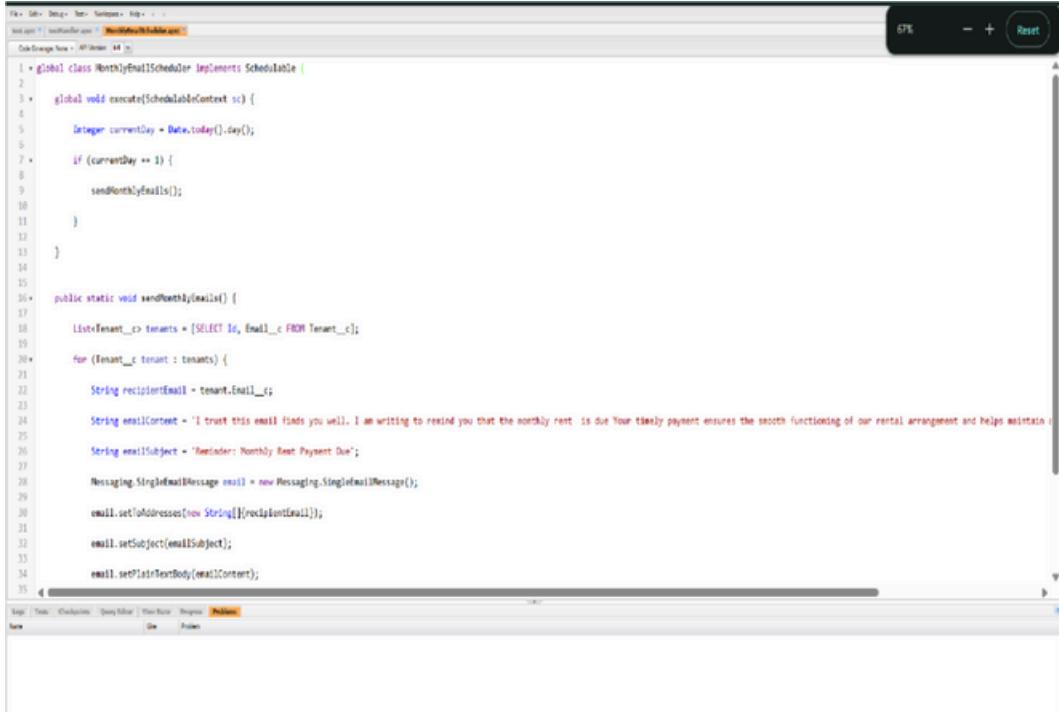
- To create a validation rule to a Lease Object

The screenshot shows the 'Validation Rule Edit' screen for the 'lease_end_date' rule. The 'Role Name' is set to 'lease_end_date'. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains the formula 'End_date__c < start_date__c'. A tooltip for the 'ABS' function is displayed, stating: 'Returns the absolute value of a number, a number without its sign'. The 'Error Message' field is empty.

The screenshot shows the 'Lease Validation Rule' detail screen. The rule is named 'lease_end_date' and has the formula 'End_date__c > start_date__c'. The error message is 'Your End date must be greater than start date'. The rule is active and was created by 'manimegala|D' on 9/6/2025, 1:52 AM. It was modified by 'manimegala|D' on 9/6/2025, 1:52 AM.

- Added Apex trigger to restrict multiple tenants per property

- Scheduled monthly reminder emails using Apex class

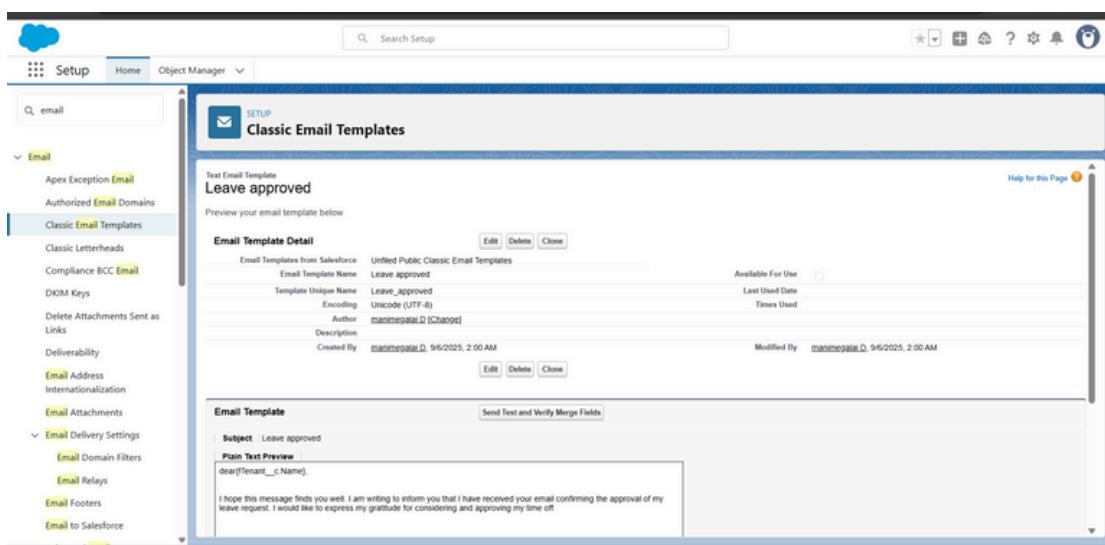


```

1 * Global class MonthlyEmailScheduler implements Scheduleable {
2 *
3 *     global void execute(SchedulableContext sc) {
4 *
5 *         Integer currentDay = Date.today().day();
6 *
7 *         if (currentDay == 1) {
8 *
9 *             sendMonthlyEmails();
10 *         }
11 *     }
12 *
13 * }
14 *
15 * public static void sendMonthlyEmails() {
16 *
17 *     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
18 *
19 *     for (Tenant__c tenant : tenants) {
20 *
21 *         String recipientEmail = tenant.Email__c;
22 *
23 *         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';
24 *
25 *         String emailSubject = 'Reminder: Monthly Rent Payment Due';
26 *
27 *         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
28 *
29 *         email.setToAddresses(new String[]{recipientEmail});
30 *
31 *         email.setSubject(emailSubject);
32 *
33 *         email.setPlainTextBody(emailContent);
34 *
35 *     }
36 *
37 * }
38 */

```

- Built and tested email templates for leave request, approval, rejection, payment, and reminders



The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Email', 'Classic Email Templates' is highlighted. A specific template, 'Leave rejected', is selected. The main pane displays the 'Email Template Detail' for 'Leave rejected'. The template is defined as 'Leave rejected' with 'Leave_rejected' as the unique name, 'UTF-8' as the encoding, and 'manimegalai_D [changed]' as the author. It was created by 'manimegalai_D' on 9/6/2025, 2:02 AM. The preview pane shows the email content:

Email Template

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

- Apex Trigger

Create an Apex Trigger

```

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

```

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'testHandler' in the 'testHandler.apex' file. The trigger is defined as a before insert trigger on the 'Tenant__c' object. The code prevents insertions if the trigger is insert and before. A screenshot of the 'Open' dialog is overlaid on the code editor, showing the 'Triggers' entity type selected.

Developer Console - Google Chrome
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

File • Edit • Debug • Test • Workspace • Help • < >

testHandler.apc [MonthlyTaskScheduler.apc]

Code Coverage: None • API Version: 41 • Go To

```
1 trigger test on Tenant__c (before insert)
2
3 {
4     if(trigger.isInsert && trigger.isBefore){
5         testHandler.preventInsert(trigger.new);
6     }
7 }
8
9
10
11 }
```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

Create an Apex Handler class

Developer Console - Google Chrome
orgfarm-5d1fe8052-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage

File • Edit • Debug • Test • Workspace • Help • < >

testHandler.apc [MonthlyTaskScheduler.apc]

Code Coverage: None • API Version: 41 • Go To

```
1 + public class testHandler {
2
3 +     public static void preventInsert(List<Tenant__c> newList) {
4
5     Set<Id> existingPropertyIds = new Set<Id>();
6
7     for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9         existingPropertyIds.add(existin
10     }
11
12
13
14+     for (Tenant__c newTenant : newList) {
15
16
17         if (newTenant.Property__c != null) {
18             newTenantaddError('A t
19
20
21
22
23 }
```

Entity Type Entity Name Namespace Related

Entity Type	Entity Name	Namespace	Related
Empty List			
Classes	testHandler	MonthlyTaskScheduler	+< test ApexTrigger References
Triggers			+< property CustomerId References
Pages			+< Tenant__c Subject References
Page Components			+< Tenant__c Subject References
Objects			
Item Resources			
Packages			

Open Filter: Filter the repository (* = any string) Hide Managed Packages Refresh

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

Developer Console - Google Chrome

orgfarm-Sd1t0e05f2-dev-ed.develop.my.salesforce.com/u/common/apex/debug/ApexCSPage

Test.Javascript testHandler.apc [] HostedDynamicScheduler.apc []

Code Coverage Class Name API Version 54 Go To

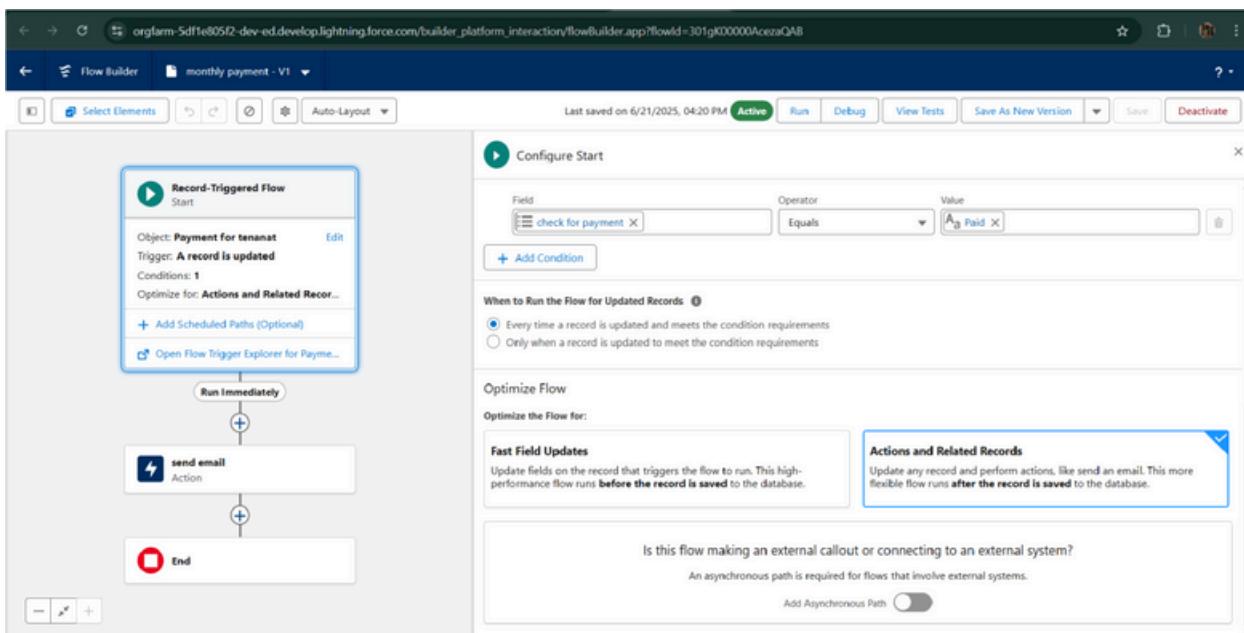
```

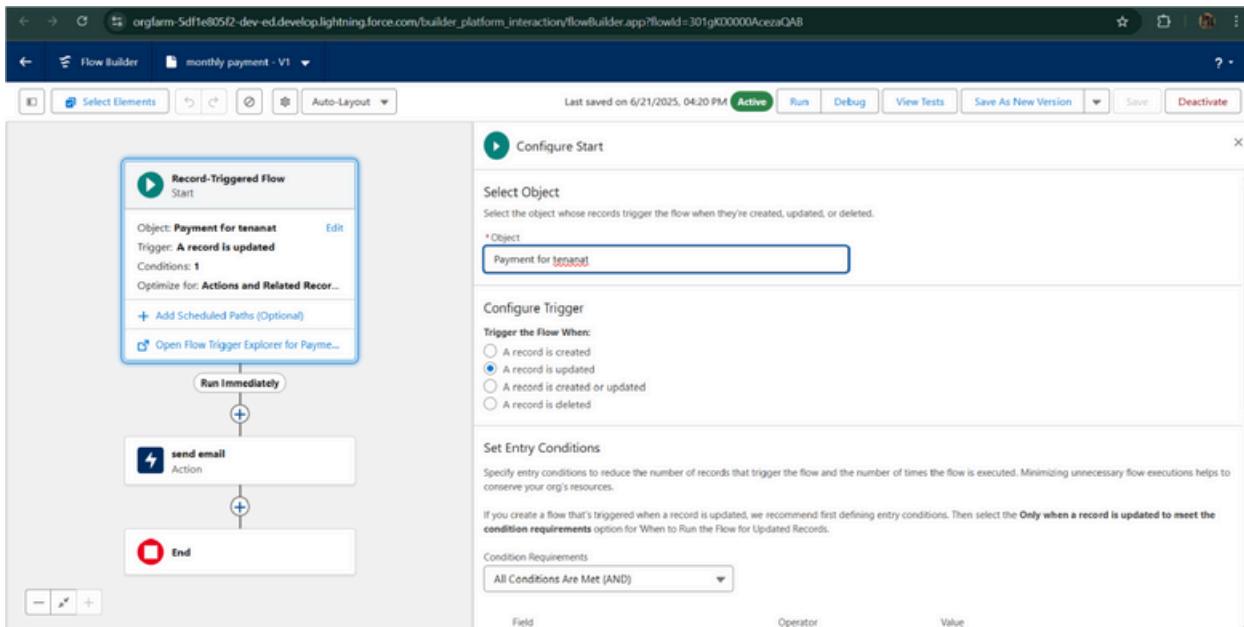
1 * public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);
10
11         }
12
13
14         for (Tenant__c newTenant : newList) {
15
16
17             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
18
19                 newTenantaddError('A tenant can have only one property');
20
21             }
22
23         }
24
25     }
26
27 }
```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Name Line Problem

- FLOWS





- Schedule class:
Create an Apex Class

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10
11        }
12
13    }
14
15
16    public static void sendMonthlyEmails() {
17
18        List<Tenant__c> tenants = [SELECT
19
20            for (Tenant__c tenant : tenants)
21                String recipientEmail = tenant.Email__c;
  
```

Entity Type: Classes

Entity Type	Edition	Related
Classes	testHandler	Name: testHandler Namespace: testHandler
Triggers	MonthlyEmailScheduler	Email: CustomEmail_ References: Tenant__c
Pages		← Tenant__c: Subject: References: Tenant__c
Page Components		
Objects		
Static Resources		
Packages		

```

1  global class MonthlyEmailScheduler implements schedulable {
2
3      public void execute(SchedulableContext sc) {
4
5          Integer currentDay = Date.today().day();
6
7          if (currentDay == 15) {
8
9              sendEmail();
10         }
11     }
12   }
13
14
15
16  public static void sendEmail() {
17
18      List<Email__c> emails = [SELECT Id, Email__c FROM Email__c];
19
20      for (Email__c email : emails) {
21
22          String recipientEmail = email.Email__c;
23
24          String emailSubject = 'Reminder: monthly rent payment due';
25
26          String emailContent = 'Dear Sir/Madam, I hope 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.' ;
27
28          Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
29
30          mail.setTo(RecipientEmail);
31          mail.setSubject(emailSubject);
32          mail.setPlainTextEmailContent();
33          mail.setHTMLEmailContent();
34
35          messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail});
36      }
37  }
38
39 }
40

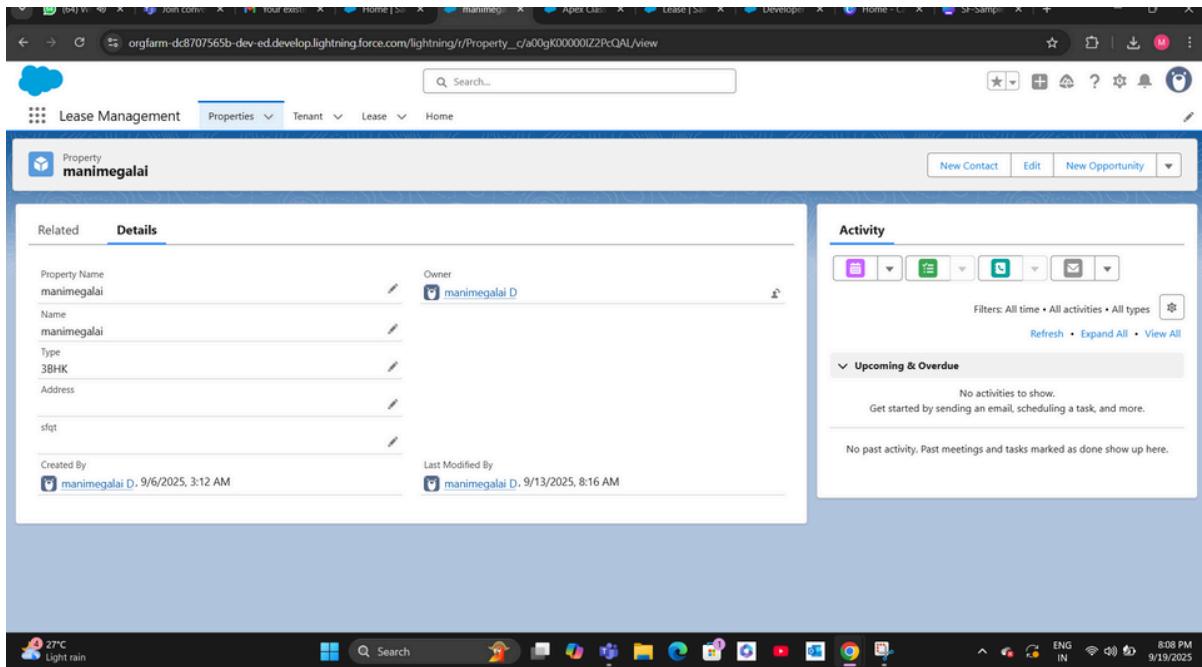
```

Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The sidebar on the left has sections for Email (Apex Exception Email), Custom Code (Apex Classes selected), Apex Settings, Apex Test Execution, Apex Test History, and Apex Triggers. The main content area has a title "Apex Classes" with a sub-section "Apex Classes". It displays the following information:

- Percent of Apex Used:** 0.00%
- Code Coverage:** You are currently using 1,896 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.
- Action Bar:** Developer Console, New, Generate from WSDL, Run All Tests, Schedule Apex.
- Table:** A list of Apex classes:

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	MonthlyEmailScheduler		64.0	Active	1,125	manimegalai_D	9/13/2025, 7:20 AM
Edit Del Security	MonthlyEmailScheduler1		64.0	Active	40	manimegalai_D	9/13/2025, 6:37 AM
Edit Del Security	testHandler		64.0	Active	584	manimegalai_D	9/6/2025, 3:43 AM
- Dynamic Apex Classes:** Dynamic Apex extends your programming reach by interacting with Lightning Platform components.



FUNCTIONAL AND PERFORMANCE TESTING

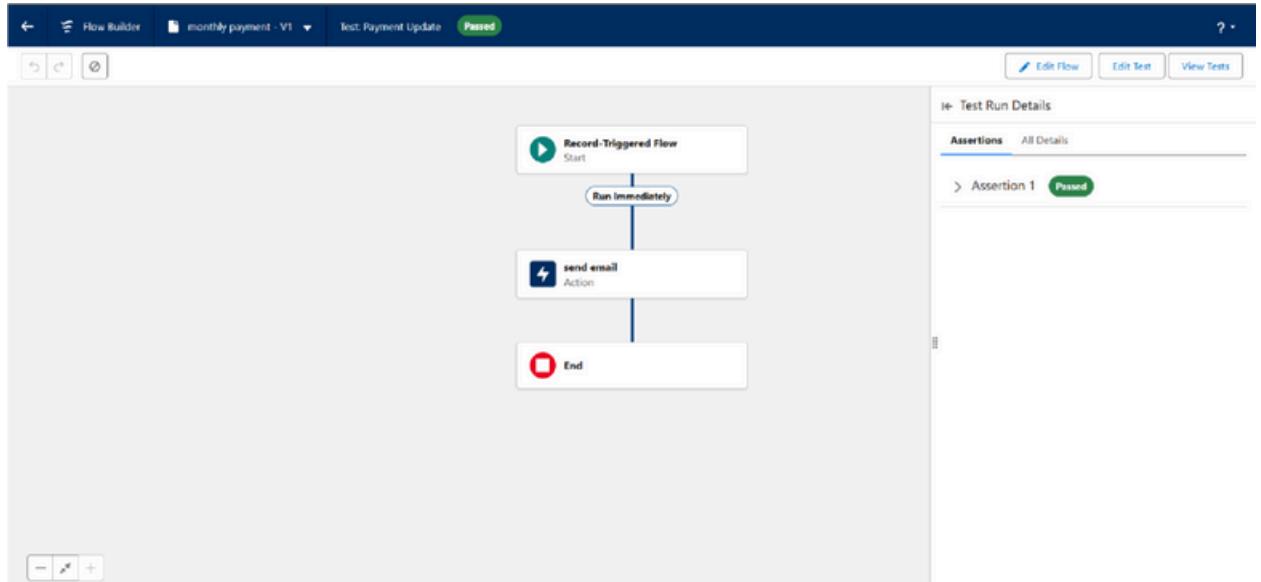
Performance Testing

- Trigger validation by entering duplicate tenant-property records
- Validation Rule checking

The screenshot shows a Salesforce Lease Management interface. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', 'lease', and a search bar. The main area displays a lease record for 'supriya'. The 'Details' tab is selected. The 'start date' field contains '6/20/2025' and the 'End date' field contains '6/19/2025', which is highlighted with a red border and an error message: 'Your End date must be greater than start date.' A modal window titled 'We hit a snag.' lists the error: 'Review the following fields: start_date'. The 'Owner' is listed as 'Sowmya Team'. The 'Activity' sidebar on the right shows no upcoming or overdue activities.

This screenshot shows a similar Salesforce Lease Management interface. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', 'lease', and a search bar. The main area displays a lease record for 'supriya'. The 'Details' tab is selected. The 'start date' field contains '6/20/2025' and the 'End date' field contains '6/19/2025', which is highlighted with a red border and an error message: 'Your End date must be greater than start date.' A modal window titled 'We hit a snag.' lists the error: 'Review the following fields: start_date'. The 'Owner' is listed as 'Sowmya Team'. The 'Activity' sidebar on the right shows no upcoming or overdue activities.

- Test flows on payment update



- Approval process validated through email alerts and status updates

RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface with the 'Tabs' page selected under 'User Interface'. The left sidebar has 'Setup' selected. The main content area is titled 'Custom Tabs' and includes sections for 'Custom Object Tabs', 'Web Tabs', and 'Visualforce Tabs'. The 'Custom Object Tabs' section lists tabs for 'Issue', 'Payment', 'Accedit', and 'Tentant', each with a 'Tab Style' icon (e.g., Keys, Credit card, Back, Map). A note at the bottom of this section states: '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.' A 'Help for this Page' link is in the top right.

- Approval process notifications

The screenshot shows the Salesforce mobile app's notification center. It displays two notifications from 'niranjan': 'Approval request for the tenant is approved' (timestamp: 'a few seconds ago') and 'Approval request for the tenant is rejected' (timestamp: 'Jun 23, 2025, 4:29 PM'). The notifications are shown in a scrollable list with a blue header bar and a white background.

ADVANTAGES & DISADVANTAGES

CONCLUSION

The Lease Management System successfully streamlines the operations of leasing through a structured, automated Salesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.

APPENDIX

Provided in Apex Classes and Triggers

- **Source Code:**

Test.apxt:

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

```
    } }

testHandler.apxc:

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');

}
}

}
```

```
}
```

```
}
```

MothlyEmailScheduler.apxc:

```
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});  
  
    }  
  
}
```