

# Project Files

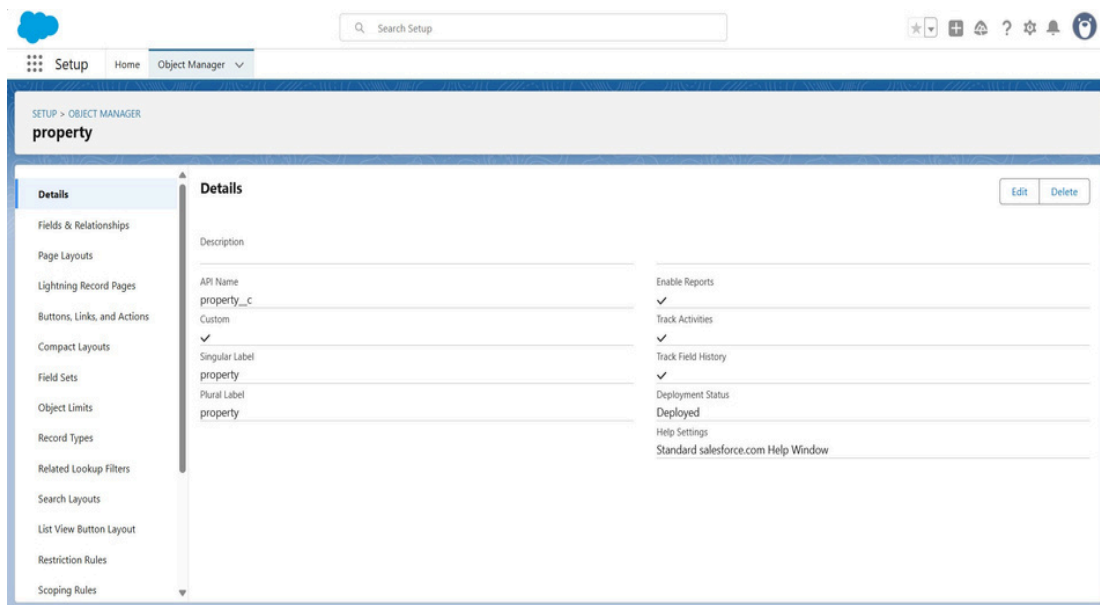
Date	20 JUNE 2025
Team ID	LTVIP2025TMID29234
Project Name	Lease Management
Maximum Marks	

This is a Salesforce-based implementation, so the "executables" are not traditional .exe files but configurations and custom code within the Salesforce ecosystem. The project includes:

## A. Custom Salesforce Objects

Created via Salesforce Setup:

### ● Property



## ● Tenant

The screenshot shows the Salesforce Setup interface for the 'Tenant' object. The left sidebar contains a list of configuration options: 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, and Scoping Rules. The main content area is titled 'Tenant' and includes a 'Details' section with the following fields:

Field	Value
Description	
API Name	Tenant_c
Custom	<input checked="" type="checkbox"/>
Singular Label	Tenant
Plural Label	Tenants
Enable Reports	<input checked="" type="checkbox"/>
Track Activities	<input checked="" type="checkbox"/>
Track Field History	<input checked="" type="checkbox"/>
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Buttons for 'Edit' and 'Delete' are located in the top right corner of the details section.

## ● Lease

The screenshot shows the Salesforce Setup interface for the 'Lease' object. The left sidebar contains a list of configuration options: 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, and Scoping Rules. The main content area is titled 'Lease' and includes a 'Details' section with the following fields:

Field	Value
Description	
API Name	Lease_c
Custom	<input checked="" type="checkbox"/>
Singular Label	Lease
Plural Label	Leases
Enable Reports	<input checked="" type="checkbox"/>
Track Activities	<input checked="" type="checkbox"/>
Track Field History	<input checked="" type="checkbox"/>
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

Buttons for 'Edit' and 'Delete' are located in the top right corner of the details section.

## ● Payment for Tenant

The screenshot shows the Salesforce Setup interface. The top navigation bar includes the Setup icon, a search bar labeled "Search Setup", and several utility icons. Below the navigation bar, the breadcrumb trail reads "SETUP > OBJECT MANAGER". The main heading is "Payment for tenant". On the left, a sidebar lists various configuration options: 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, and Scoping Rules. The "Details" section is selected and expanded, showing a table of configuration options. The table has two columns: "API Name" and "Description". The rows are: "Payment\_for\_tenant\_\_c" (Description: Enable Reports), "Custom" (Description: Track Activities), "Singular Label" (Description: Track Field History), "Payment for tenant" (Description: Deployment Status), "Plural Label" (Description: Deployed), and "Payment" (Description: Help Settings). The "Help Settings" row has a link to "Standard salesforce.com Help Window".

API Name	Description
Payment_for_tenant__c	Enable Reports
Custom	Track Activities
Singular Label	Track Field History
Payment for tenant	Deployment Status
Plural Label	Deployed
Payment	Help Settings

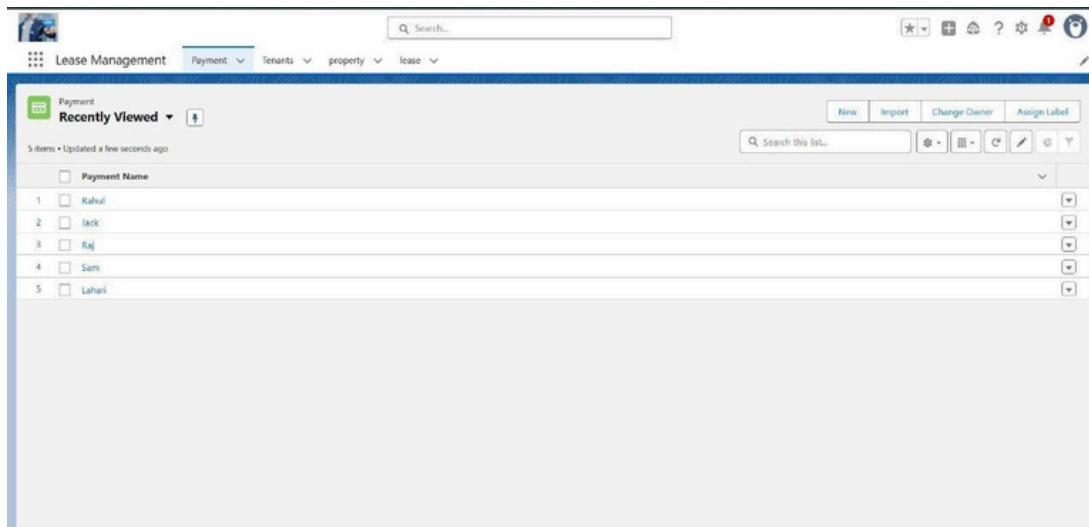
## B. Custom Tabs and App

### ● Tabs for each object

The screenshot shows the Salesforce Setup interface. The top navigation bar includes the Setup icon, a search bar labeled "Search Setup", and several utility icons. Below the navigation bar, the breadcrumb trail reads "Setup > Home > Object Manager". The main heading is "Tabs". On the left, a sidebar lists various configuration options: Tabs, User Interface, Rename Tabs and Labels, and Tabs. The "Tabs" section is selected and expanded, showing a table of configuration options. The table has two columns: "Action" and "Label". The rows are: "New" (Label: What Is This?), "Edit" (Label: Edit), "Delete" (Label: Delete), "Duplicate" (Label: Duplicate), "Clone" (Label: Clone), "Copy" (Label: Copy), and "Paste" (Label: Paste). The "New" row has a link to "What Is This?".

Action	Label
New	What Is This?
Edit	Edit
Delete	Delete
Duplicate	Duplicate
Clone	Clone
Copy	Copy
Paste	Paste

- A Lightning App called **Lease Management** with navigation setup



## C. Apex Code (Custom Backend Logic)

- **Apex Trigger:** test on Tenant\_\_c

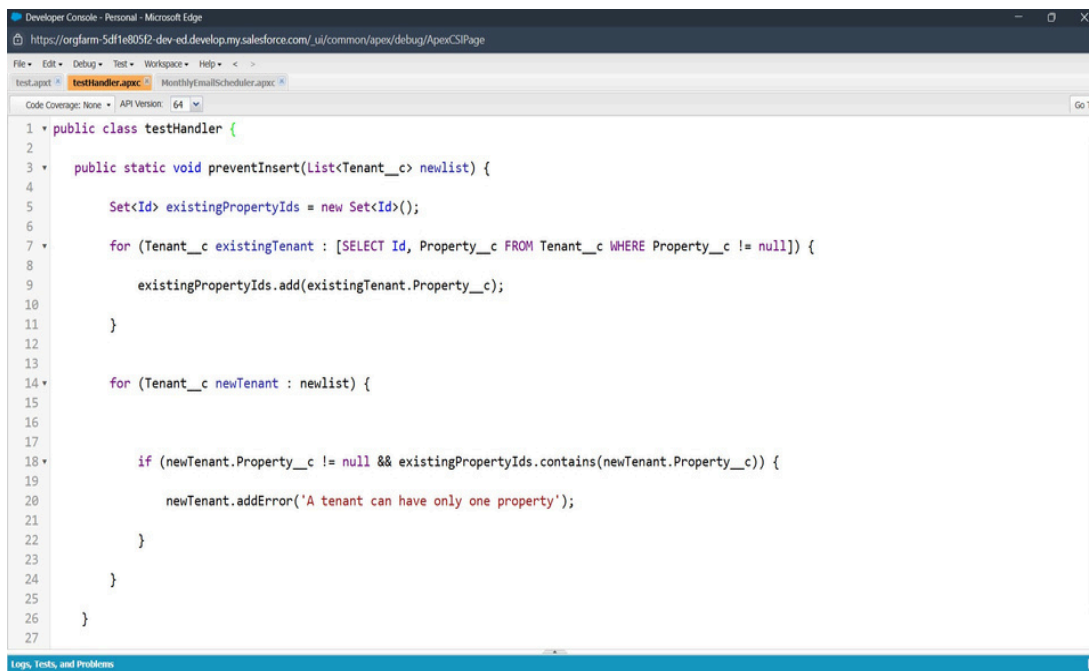
```
Developer Console - Personal - Microsoft Edge
https://orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >
test.apxt testHandler.apxc MonthlyEmailScheduler.apxc
Code Coverage: None API Version: 64
1 trigger test on Tenant__c (before insert)
2
3 {
4
5     if(trigger.isInsert && trigger.isBefore){
6
7         testHandler.preventInsert(trigger.new);
8
9     }
10
11 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name	Line	Problem
------	------	---------

## ● Apex Handler Class: testHandler

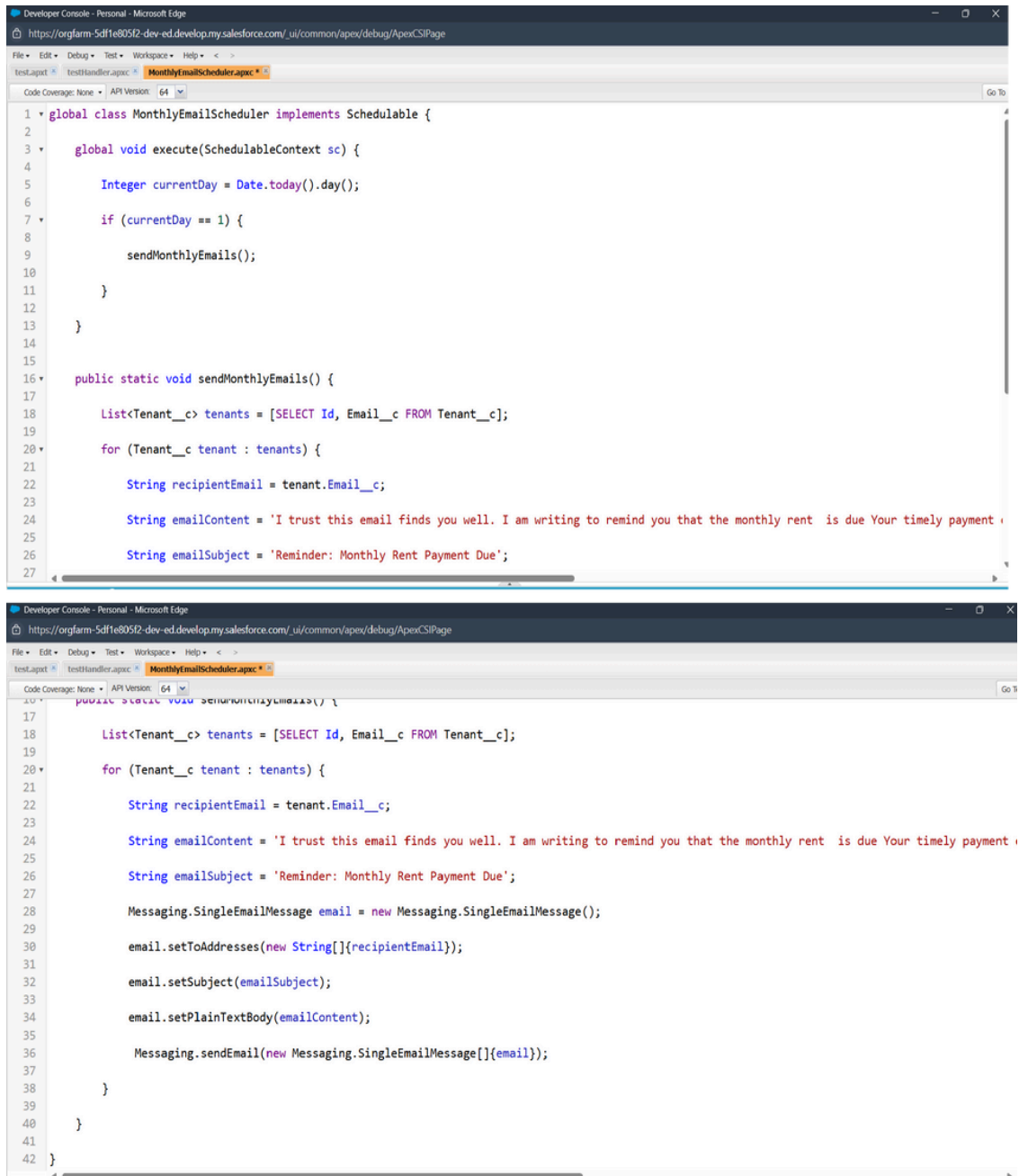


The screenshot shows the Salesforce Developer Console with the Apex class `testHandler` open. The class contains a static method `preventInsert` that checks for existing properties before inserting new tenants. The code is as follows:

```
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         for (Tenant__c newTenant : newList) {  
14  
15  
16  
17             if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {  
18  
19                 newTenant.addError('A tenant can have only one property');  
20  
21             }  
22  
23         }  
24     }  
25 }  
26  
27 }
```

The interface at the bottom shows 'Logs, Tests, and Problems'.

## ● Scheduled Apex Class: MonthlyEmailScheduler



```
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
17 public static void sendMonthlyEmails() {
18
19     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
20
21     for (Tenant__c tenant : tenants) {
22
23         String recipientEmail = tenant.Email__c;
24
25         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due Your timely payment';
26
27         String emailSubject = 'Reminder: Monthly Rent Payment Due';
28
29     }
30 }
31
32
33
34
35
36
37
38
39
40
41
42 }
```

```
17
18 List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20 for (Tenant__c tenant : tenants) {
21
22     String recipientEmail = tenant.Email__c;
23
24     String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due Your timely payment';
25
26     String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28     Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30     email.setToAddresses(new String[]{recipientEmail});
31
32     email.setSubject(emailSubject);
33
34     email.setPlainTextBody(emailContent);
35
36     Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
37
38 }
39
40 }
41
42 }
```

## D. Validation Rule

- Lease object has a rule: End\_date\_\_c > Start\_date\_\_c

The screenshot shows the Salesforce Setup interface, specifically the Object Manager for the 'lease' object. The left sidebar contains a navigation menu with options like 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, and Scoping Rules. The main content area is titled 'Validation Rules' and shows a table with one rule.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
lease_end_date	start date	Your End date must be greater than start date	✓	Sowmya Team, 6/19/2025, 5:37 AM

## E. Approval Process

- For Tenant object: "Check for vacant"

The screenshot shows the Salesforce Setup interface, specifically the Approval Processes for the 'Tenant' object. The left sidebar contains a navigation menu with options like Approval Processes, Process Automation, and Approval Processes. The main content area is titled 'Approval Processes' and shows the details for the 'Tenant: check for vacant' process.

**Process Definition Detail**

Field	Value
Process Name	check for vacant
Unique Name	CHECK_FOR_VACANT
Entry Criteria	STATUS: STATUS NOT EQUAL TO LEAVING
Record Eligibility	Administrator ONLY
Approval Assignment Email Template	LEASE_APPROVAL
Initial Submitters	Tenant Owner
Created By	Sowmya Team, 6/20/2025, 3:18 AM
Modified By	Sowmya Team, 6/25/2025, 4:46 AM

**Initial Submission Actions**

Action	Type	Description
Record Lock	Record Lock	Lock the record from being edited
Email Alert	Email Alert	Please approve my lease

**Approval Steps**

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
--------	-------------	------	-------------	----------	-------------------	-----------------

## F. Flows

- **Record-Triggered Flow** for monthly payment email when payment is marked as “Paid”

The screenshot displays the Microsoft Power Automate interface. At the top, there's a navigation bar with 'Lease Management' selected, and sub-menus for 'Payment', 'Tenants', and 'property'. A search bar is present on the right. Below the navigation bar, the flow 'monthly payment' is highlighted. The flow's details are shown in a card-like view. The flow is labeled 'monthly payment' and is of type 'Record—Run After Save'. It is currently 'Activated'. The flow owner is 'Sri Anjhanee Kunchanapalli'. The flow was last modified on 7/1/2025 at 5:04 AM. The flow is categorized under 'monthly payment' with the API name 'monthly\_payment'. The flow type is 'Record-Triggered After Save Flow'. The flow was created by 'Sri Anjhanee Kunchanapalli' on 7/1/2025 at 5:01 AM. The flow was last modified by 'Sri Anjhanee Kunchanapalli' on 7/1/2025 at 5:04 AM. The flow is categorized under 'monthly payment' with the API name 'monthly\_payment'.

Flow: **monthly payment**

Type: Record—Run After Save

Associated Record: **Activated**

Last Modified Date: 7/1/2025, 5:04 AM

Flow Owner: Sri Anjhanee Kunchanapalli

Related: **Details**

Information

Flow Label	monthly payment	API Name	monthly_payment
Description		Flow Type	Record-Triggered After Save Flow
Associated Record		Segment	
Created By	Sri Anjhanee Kunchanapalli, 7/1/2025, 5:01 AM	Created Date	7/1/2025, 5:01 AM
Last Modified	Sri Anjhanee Kunchanapalli, 7/1/2025, 5:04 AM	Last Modified Date	7/1/2025, 5:04 AM
Category		Subcategory	