

Project Files

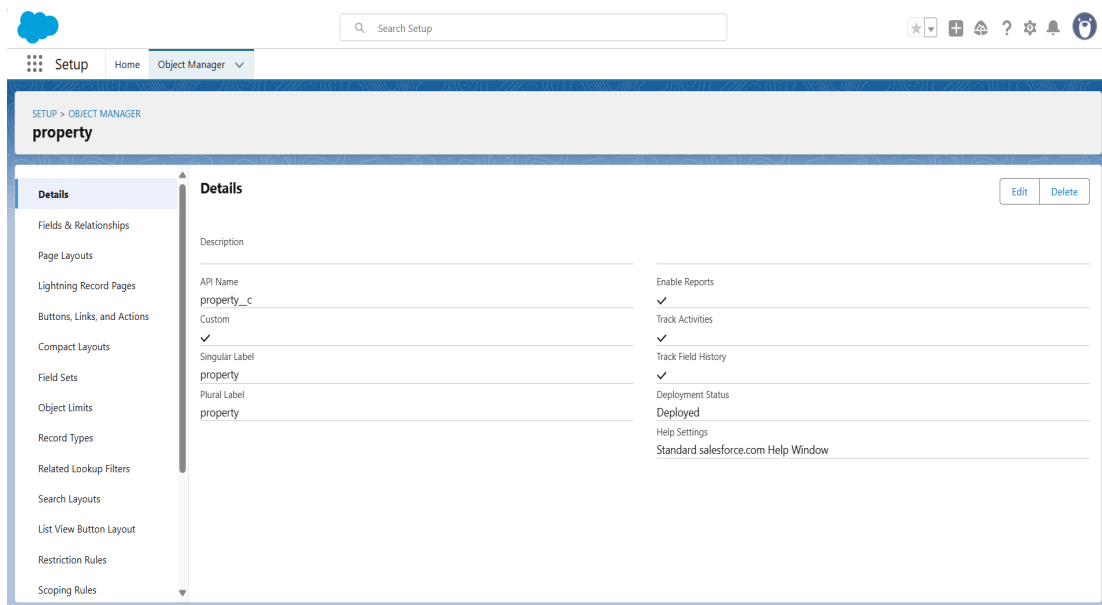
Date	25 JUNE 2025
Team ID	LTVIP2025TMID31649
Project Name	Lease Management
Maximum Marks	

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

Setup

Home

Object Manager

Search Setup

Star

Grid

Home

Help

Settings

Notifications

User

SETUP > OBJECT MANAGER

Tenant

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

Details

Description

API Name

Tenant__c

Custom

✓

Singular Label

Tenant

Plural Label

Tenants

Enable Reports

✓

Track Activities

✓

Track Field History

✓

Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

Edit

Delete

- Lease

Setup

Home

Object Manager

Search Setup

Star

Grid

Home

Help

Settings

Notifications

User

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

Details

Description

API Name

lease__c

Custom

✓

Singular Label

lease

Plural Label

lease

Enable Reports

✓

Track Activities

✓

Track Field History

✓

Deployment Status

Deployed

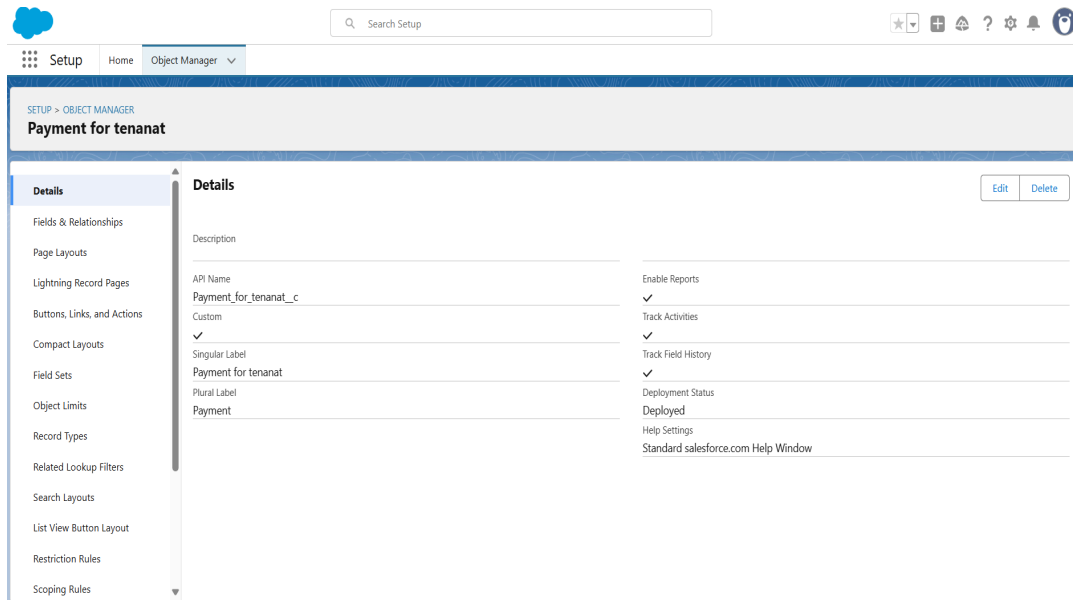
Help Settings

Standard salesforce.com Help Window

Edit

Delete

- Payment for Tenant

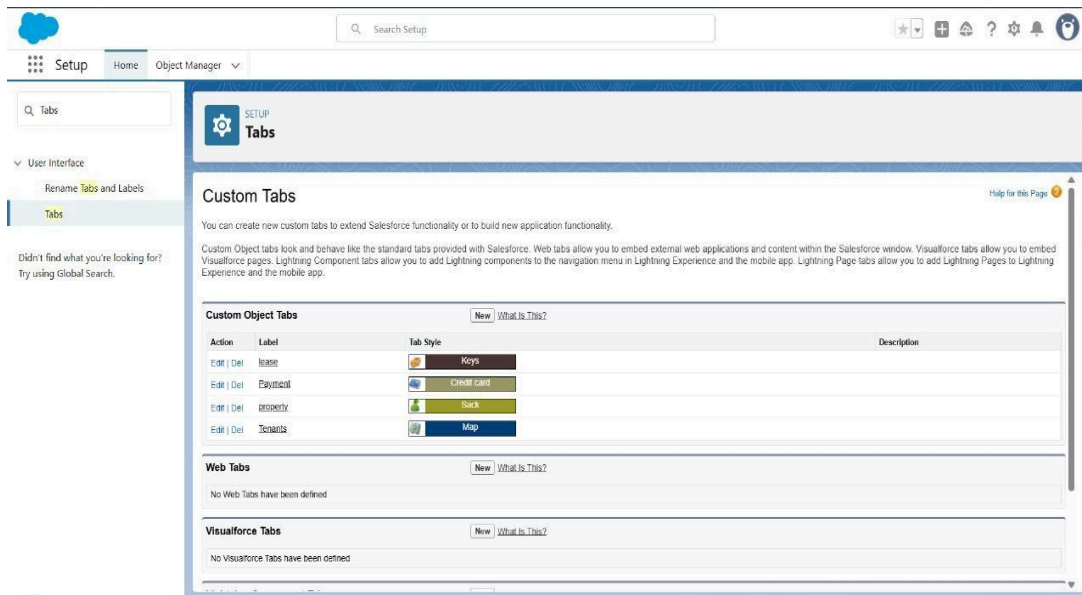


The screenshot shows the Salesforce Setup interface for the 'Payment for tenantat' object. The left sidebar contains a navigation menu with options like Fields & Relationships, Page Layouts, and Record Types. The main content area is titled 'Details' and includes a description, API Name (Payment_for_tenantat_c), and various configuration options such as Enable Reports, Track Activities, and Deployment Status.

Field	Value
Description	
API Name	Payment_for_tenantat_c
Custom	✓
Singular Label	Payment for tenantat
Plural Label	Payment
Enable Reports	✓
Track Activities	✓
Track Field History	✓
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

B. Custom Tabs and App

- Tabs for each object



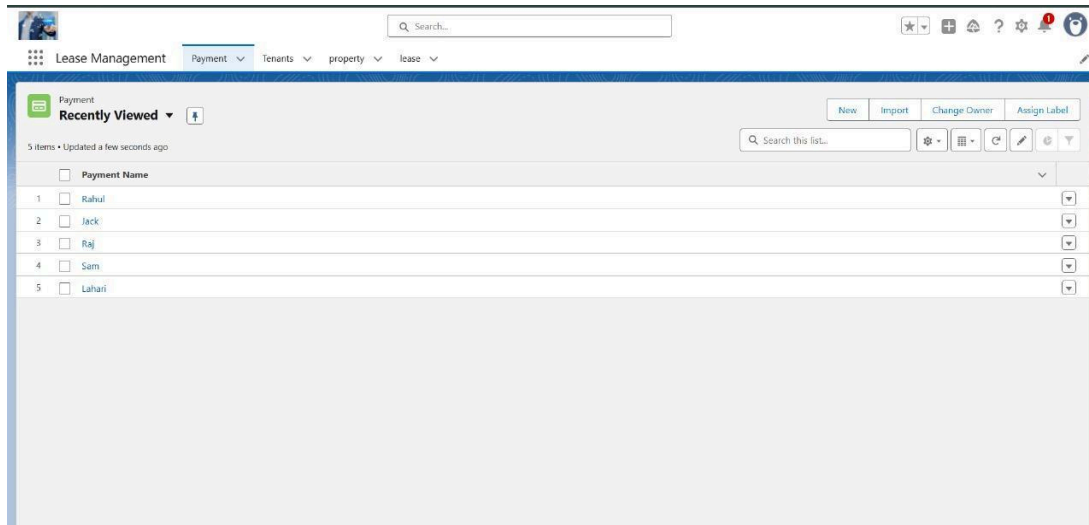
The screenshot shows the Salesforce Setup interface for the 'Tabs' page. The left sidebar contains a navigation menu with options like User Interface, Rename Tabs and Labels, and Tabs. The main content area is titled 'Custom Tabs' and includes a table of Custom Object Tabs, Web Tabs, and Visualforce Tabs.

Action	Label	Tab Style	Description
Edit Del	lease	Keys	
Edit Del	Payment	Credit card	
Edit Del	property	Sack	
Edit Del	Tenants	Map	

Web Tabs
No Web Tabs have been defined

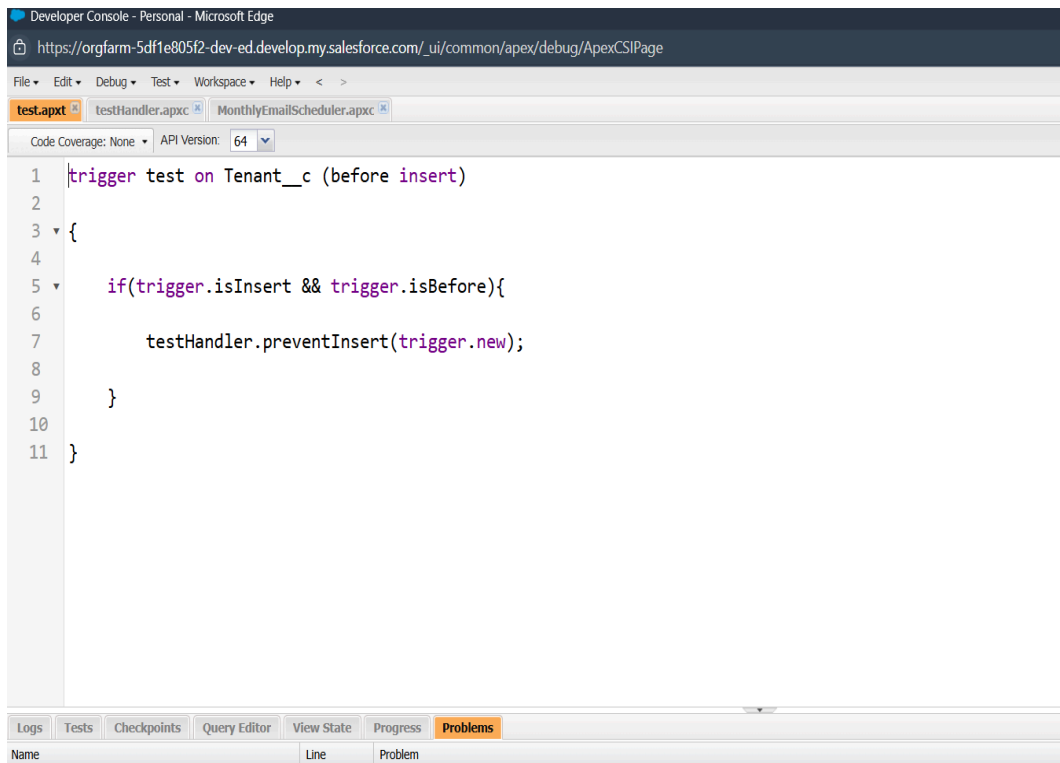
Visualforce Tabs
No Visualforce Tabs have been defined

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

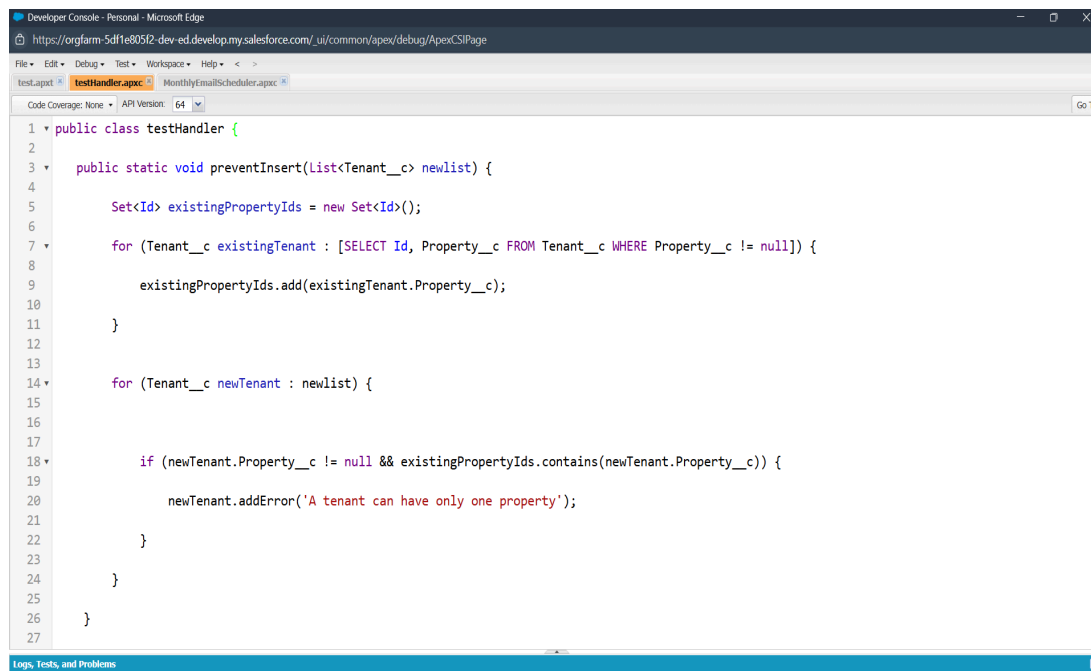


C. Apex Code (Custom Backend Logic)

- **Apex Trigger:** test on Tenant__c



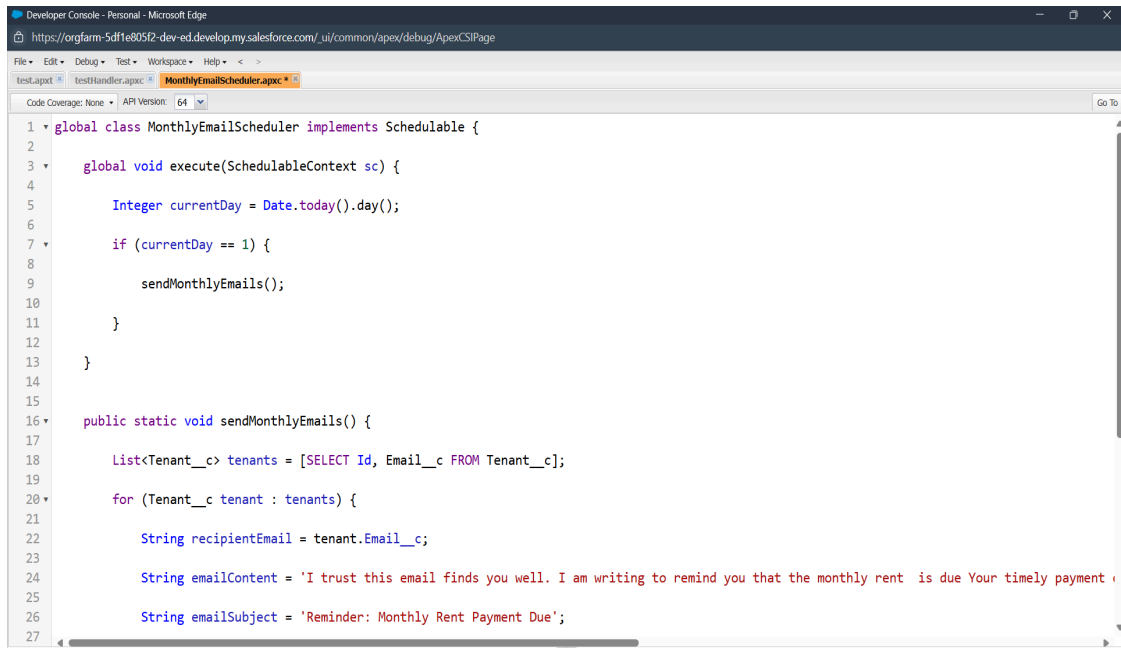
- **Apex Handler Class: testHandler**



The screenshot shows the Salesforce Developer Console with the Apex class `testHandler` open. The class is located in the `test.appt` file. The code defines a `testHandler` class with a `preventInsert` method. The method takes a `List<Tenant__c>` parameter and performs a database query to find existing tenants with non-null property IDs. It then iterates through the new list of tenants, checking if any have a property ID that already exists in the database. If a duplicate is found, an error is added to the new tenant object.

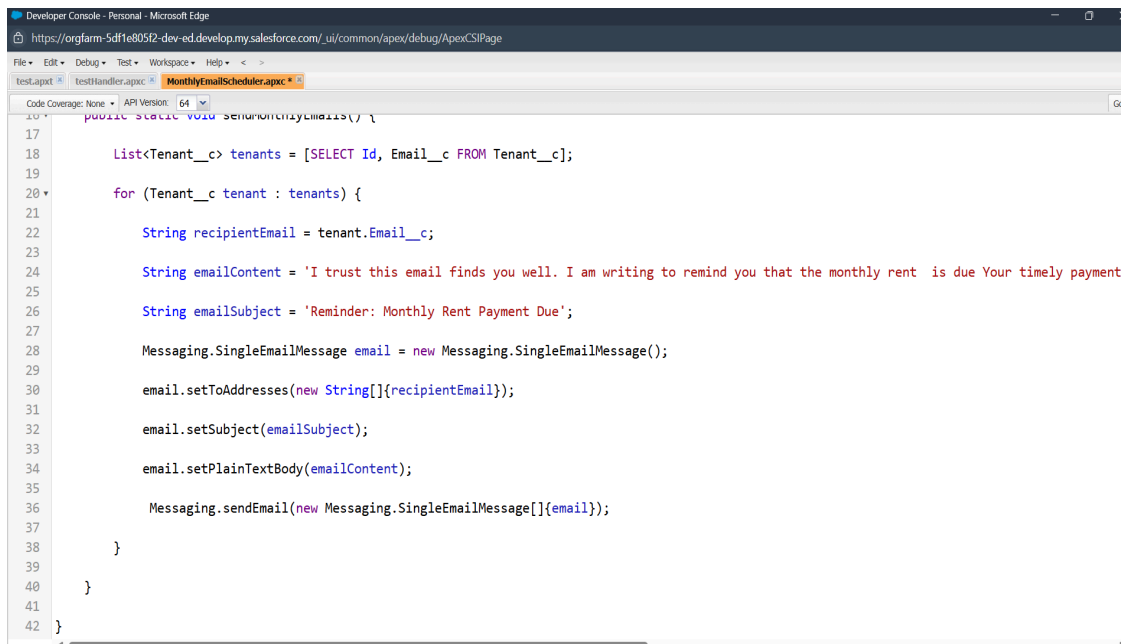
```
1 public class testHandler {
2
3     public static void preventInsert(List<Tenant__c> newList) {
4
5         Set<Id> existingPropertyIds = new Set<Id>();
6
7         for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9             existingPropertyIds.add(existingTenant.Property__c);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

- **Scheduled Apex Class: MonthlyEmailScheduler**



The screenshot shows the Salesforce Developer Console with the 'MonthlyEmailScheduler.apex' file open. The code defines a global class that implements the Schedulable interface. It includes an execute method that checks if the current day is the 1st of the month and a static sendMonthlyEmails method that queries for tenants and sends them reminder emails.

```
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 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 o
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27     }
```



This screenshot shows the continuation of the 'sendMonthlyEmails' method. It details the creation of a SingleEmailMessage object, setting its recipient, subject, and body, and then sending it via the Messaging.sendEmail method.

```
20 public static void sendMonthlyEmails() {
21
22     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
23
24     for (Tenant__c tenant : tenants) {
25
26         String recipientEmail = tenant.Email__c;
27
28         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due Your timely payment o
29
30         String emailSubject = 'Reminder: Monthly Rent Payment Due';
31
32         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
33
34         email.setToAddresses(new String[]{recipientEmail});
35
36         email.setSubject(emailSubject);
37
38         email.setPlainTextBody(emailContent);
39
40         Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
41
42     }
43 }
```

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 for the 'Approval Processes' section. 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. The process definition details include:

- Process Name: check for vacant
- Unique Name: check_for_vacant
- Description: Tenant: status NOT EQUAL TO Leaving
- Entry Criteria: Administrator ONLY
- Record Editability: Allow Submitters to Recall Approval Requests
- Approval Assignment Email Template: Leave_approved
- Initial Submitters: Tenant Owner
- Created By: Sowmya Team, 6/20/2025, 3:18 AM
- Modified By: Sowmya Team, 6/25/2025, 4:46 AM

The 'Initial Submission Actions' section shows two actions:

Action	Type	Description
Record Lock	Record Lock	Lock the record from being edited
Email Alert	Email Alert	please approve my leave

The 'Approval Steps' section shows a table with columns for Action, Step Number, Name, Description, Criteria, Assigned Approver, and Reject Behavior.

F. Flows

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

The screenshot shows the Salesforce interface for a Flow named "monthly payment". The top navigation bar includes a search bar and various utility icons. The main header shows the flow name and buttons for "Open Flow", "Open Latest Version", and "Sharing". Below this, a summary bar displays the flow's type, status, last modified date, and owner.

Flow Details:

Information	
Flow Label	monthly payment
Description	
Associated Record	
Created By	Sowmya Team, 6/21/2025, 3:50 AM
Last Modified	Sowmya Team, 6/21/2025, 3:50 AM
Category	
API Name	monthly_payment
Flow Type	Record-Triggered After Save Flow
Segment	
Created Date	6/21/2025, 3:50 AM
Last Modified Date	6/21/2025, 3:50 AM
Subcategory	

At the bottom left, there is a "To Do List" icon.