

Project Files

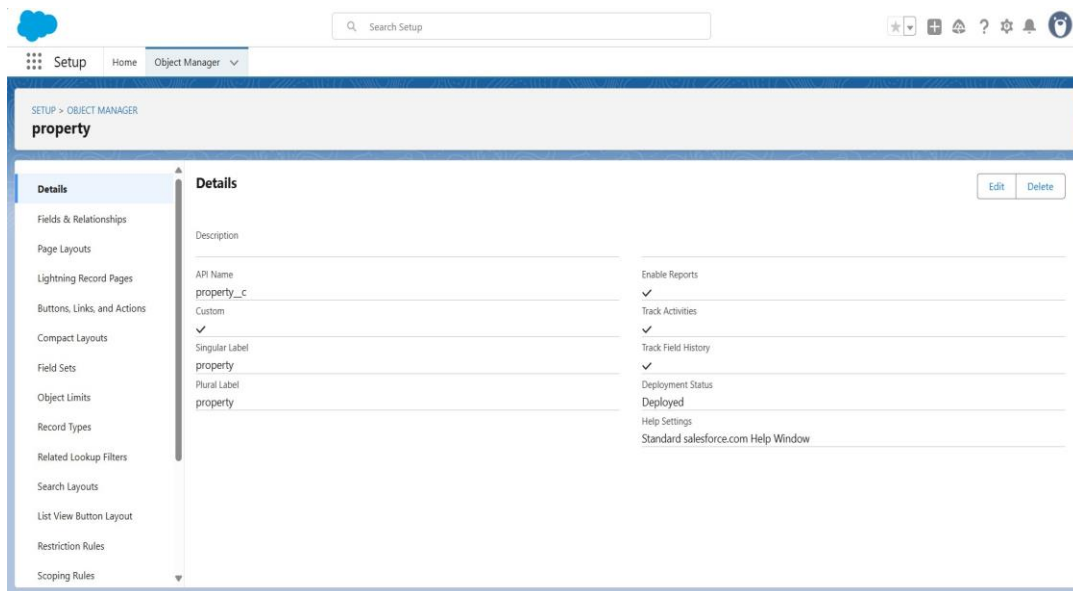
Date	25 JUNE 2025
Team ID	LTVIP2025TMID30830
Project Name	Lease Management
Maximum Marks	

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

Cloud

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

Cloud

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

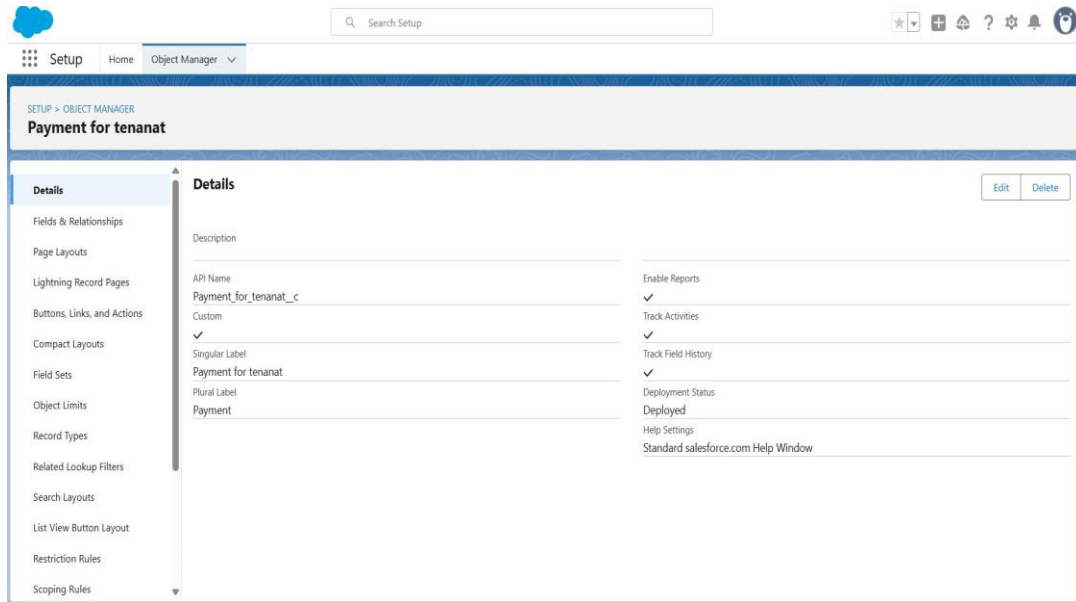
Help Settings

Standard salesforce.com Help Window

Edit

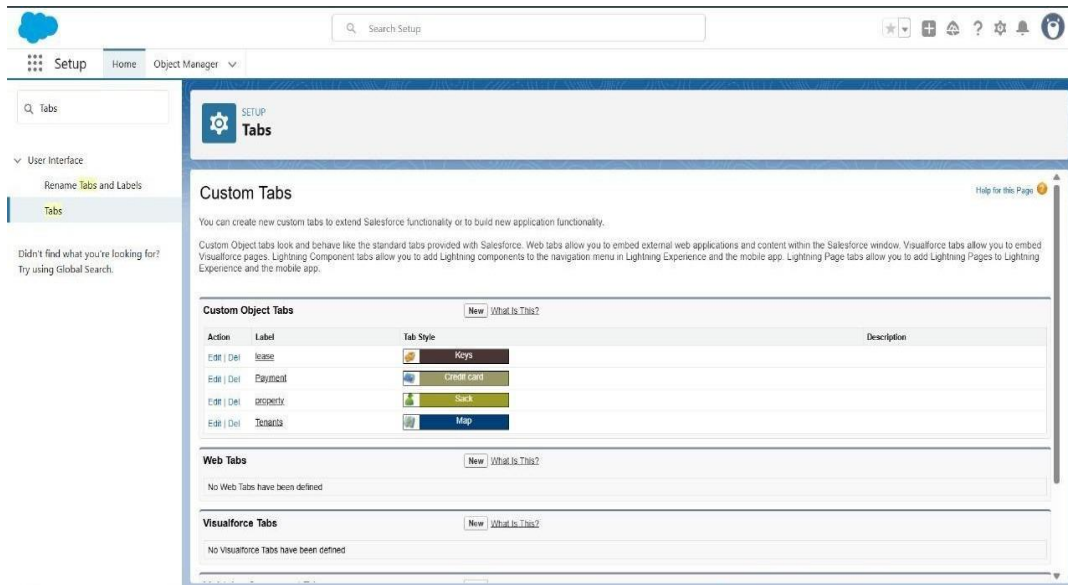
Delete

- Payment for Tenant

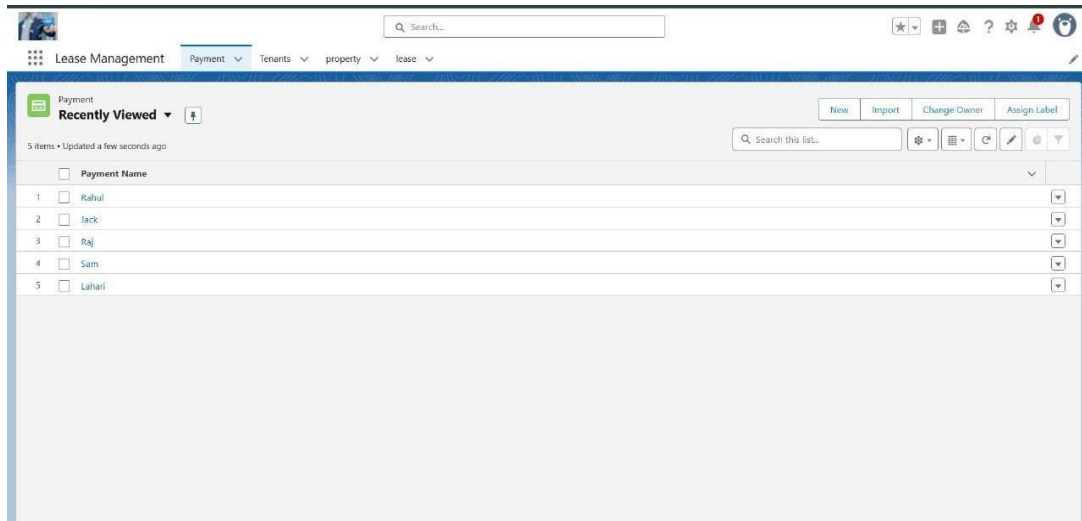


B. Custom Tabs and App

- Tabs for each object

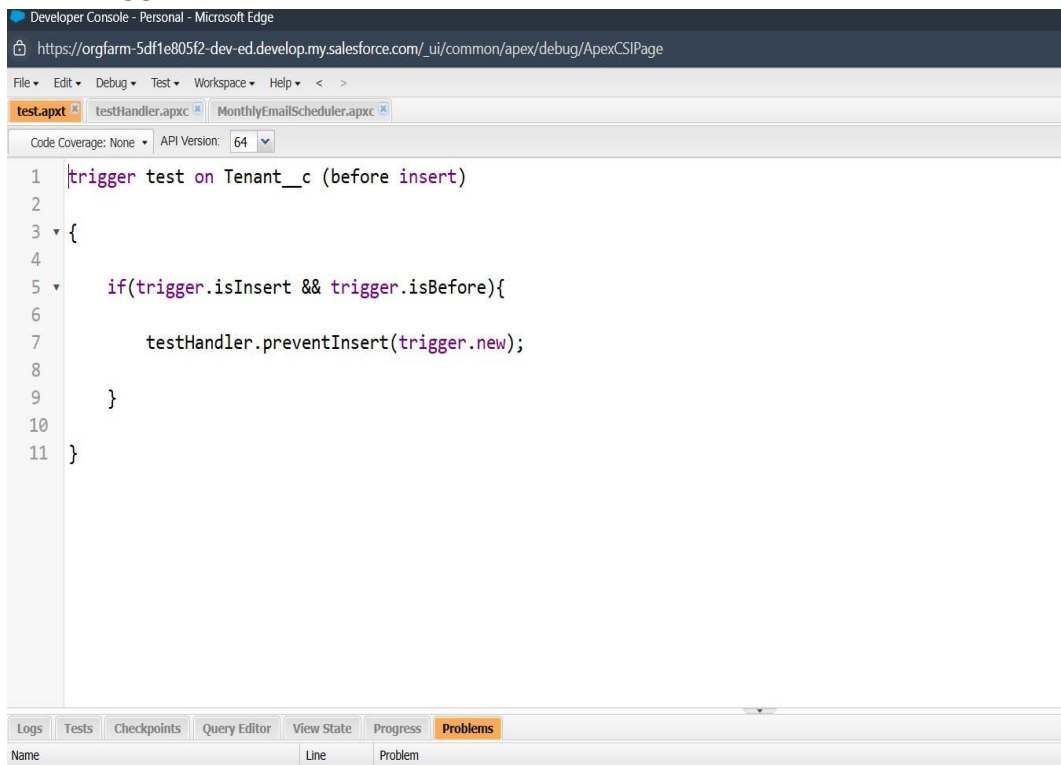


- A Lightning App called **Lease Management** with navigation setup

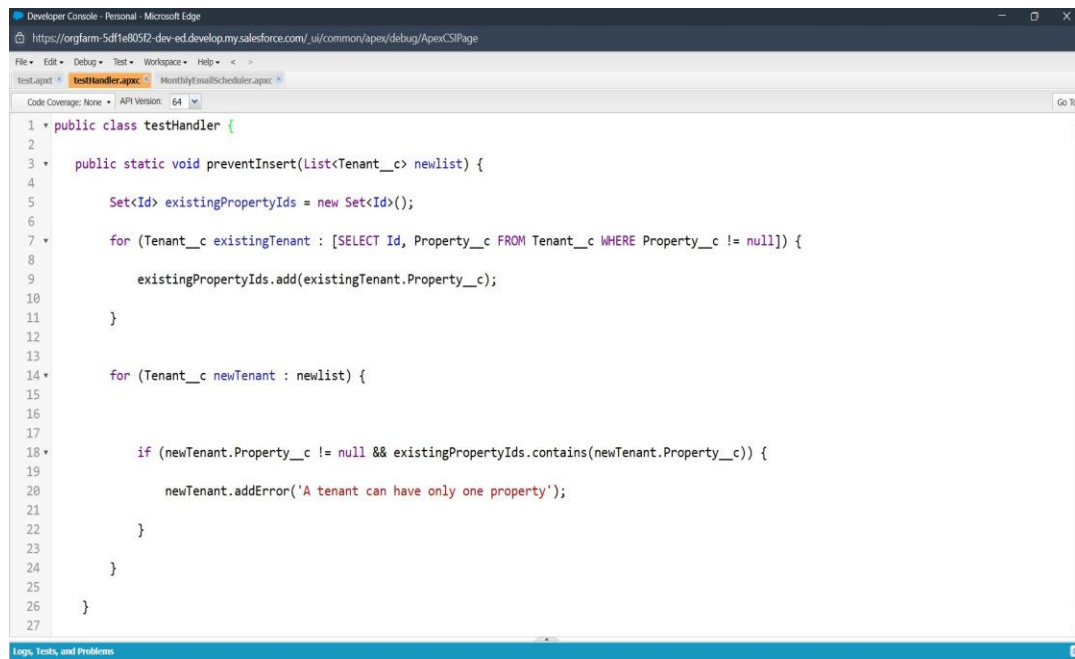


C. Apex Code (Custom Backend Logic)

- **Apex Trigger:** test on Tenant__c



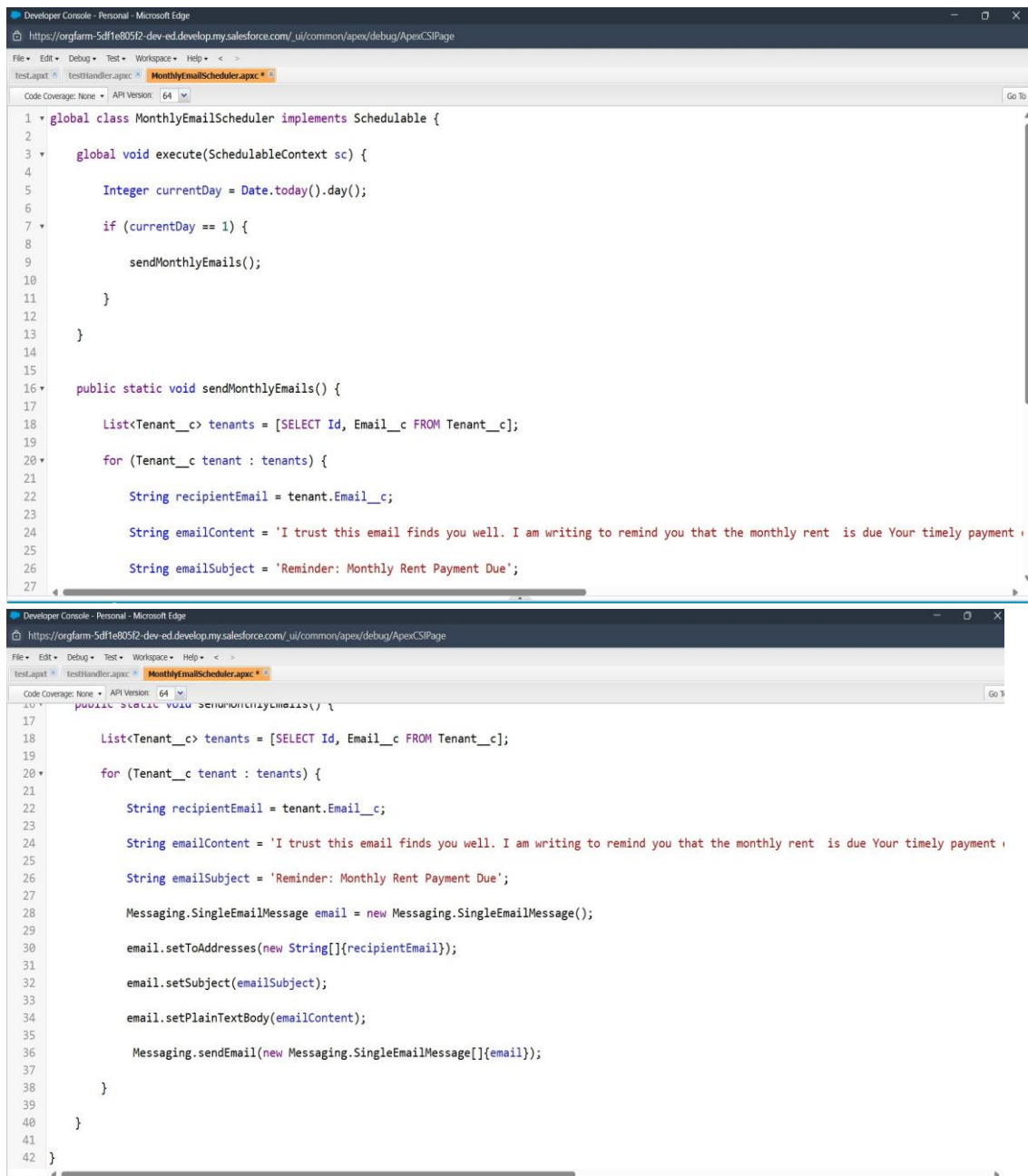
- **Apex Handler Class: testHandler**



The screenshot shows the Salesforce Developer Console with the Apex class `testHandler` open. The class contains two methods: `preventInsert` and a loop for `newlist`. The `preventInsert` method initializes a `Set<Id>` `existingPropertyIds` and iterates over existing tenants to populate this set. The loop for `newlist` checks if a new tenant's `Property__c` is already in the `existingPropertyIds` set and adds an error message if it is.

```
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  
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**



```
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';
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
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. 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' for the 'lease' object. It 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 Approval Processes. The left sidebar has a search bar and a list of process automation options, with 'Approval Processes' selected. The main content area is titled 'Approval Processes' and shows the details for a process named 'Tenant: check for vacant'. The process is active and was created by Sowmya Team on 6/20/2025. 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: Administrator ONLY
- Approval Assignment Email Template: Leave approved
- Initial Submitters: Tenant Owner
- Created By: Sowmya Team, 6/20/2025, 3:18 AM
- Modified By: Sowmya Team, 6/20/2025, 4:46 AM

The process definition also includes a section for 'Initial Submission Actions' and a table for 'Approval Steps'.

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

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Record Lock	1	Lock the record from being edited			Sowmya Team	Lock the record
Email Alert	2	please approve my leave			Sowmya Team	Send email alert

F. Flows

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

The screenshot displays the Salesforce interface for a Flow named "monthly payment". The top navigation bar includes the Salesforce logo, a search bar, and various utility icons. The main header shows the flow name "monthly payment" with buttons for "Open Flow", "Open Latest Version", and "Sharing". Below the header, a summary section provides key details:

- Type:** Record—Run After Save
- Associated Record:** (empty)
- Progress Status:** Activated
- Last Modified Date:** 6/21/2025, 3:50 AM
- Flow Owner:** Sowmya Team

The "Details" tab is selected, showing a table of flow information:

Information	
Flow Label	monthly_payment
Description	Record-Triggered After Save Flow
Associated Record	Segment
Created By	Sowmya Team
Last Modified	Sowmya Team
Category	

The bottom of the page features a "To Do List" icon.