

PROJECT TITLE

**CRM APPLICATION FOR JEWEL MANAGEMENT –
(DEVELOPER)**

College Name: GOVERNMENT ARTS&SCIENCE ,SATHY

College Code: BRU1008

TEAM ID:

TEAM MEMBERS:

Team LeaderName: **NANDHITHA R**

Email: nandhitharamu6@gmail.com

Team Member1: AARTHI M

Email: aarthi15062006@gmail.com

Team Member: NAVEEN M

Email: navijoker2006@gmail.com

Team Member: IYYAPPA P

Email: iyyappaiyyappa941@gmail.com

1.INTRODUCTION

1.1 Project Overview

The Jewel Management CRM is a Salesforce-based application designed to manage the end-to-end lifecycle of jewelry business operations. It supports customer management, inventory tracking, pricing, order management, and billing. The application leverages Salesforce automation features such as flows, validation rules, approval processes, and reports to enhance operational efficiency and customer engagement.

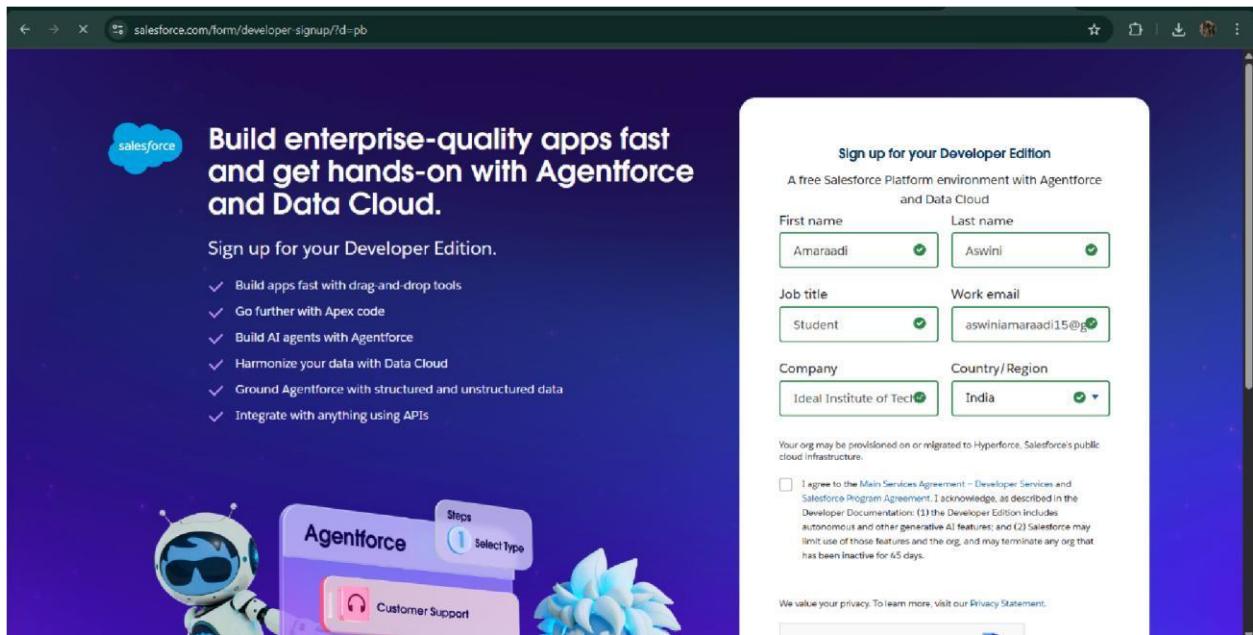
1.2 Purpose

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: **Customer Order, Price, Billing**

SETUP > OBJECT MANAGER
property

Details

Description

API Name:
property__c

Custom
✓

Singular Label:
property

Plural Label:
property

Enable Reports
✓

Track Activities
✓

Track Field History
✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK000000zTbN/Details/view

The screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. The left sidebar lists various configuration tabs: 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 'Details' tab is selected. The 'Description' field is empty. The 'API Name' field is set to 'Tenant__c'. Under the 'Custom' section, 'Singular Label' is set to 'Tenant' and 'Plural Label' is set to 'Tenants'. On the right, there are sections for 'Enable Reports' (with 'Track Activities' checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window').

orgfarm-5df1e805f2-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lgK000000zTuJ/Details/view

The screenshot shows the Salesforce Object Manager interface for the 'lease' object. The left sidebar lists the same configuration tabs as the Tenant object. The main 'Details' tab is selected. The 'Description' field is empty. The 'API Name' field is set to 'lease__c'. Under the 'Custom' section, 'Singular Label' is set to 'lease' and 'Plural Label' is set to 'lease'. On the right, there are sections for 'Enable Reports' (with 'Track Activities' checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings' (set to 'Standard salesforce.com Help Window').

Object Manager

Payment for tenant

Details

Description

API Name: Payment_for_tenant_c
Custom
✓
Singular Label: Payment for tenant
Plural Label: Payment

Enable Reports
✓ Track Activities
✓ Track Field History
✓ Deployment Status: Deployed
Help Settings: Standard salesforce.com Help Window

- Configured fields and relationships

Fields & Relationships

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(80)		
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

Setup > Object Manager
Payment for tenantat

Fields & Relationships					
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 > Object Manager
lease

Fields & Relationships					
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		

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

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

App Branding

Image: (Placeholder image)

Primary Color Hex Value: #0070D2

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

App Launcher Preview

Lease Management
Application to efficiently handle the processes relate...

Lightning App Builder | App Settings | Pages | Lease Management | Help

App Settings

App Details & Branding
App Options
Utility Items (Desktop Only)

Navigation Items

User Profiles

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)

Navigation Items

User Profiles

Choose the user profiles that can access this app.

Available Profiles

Type to filter list...

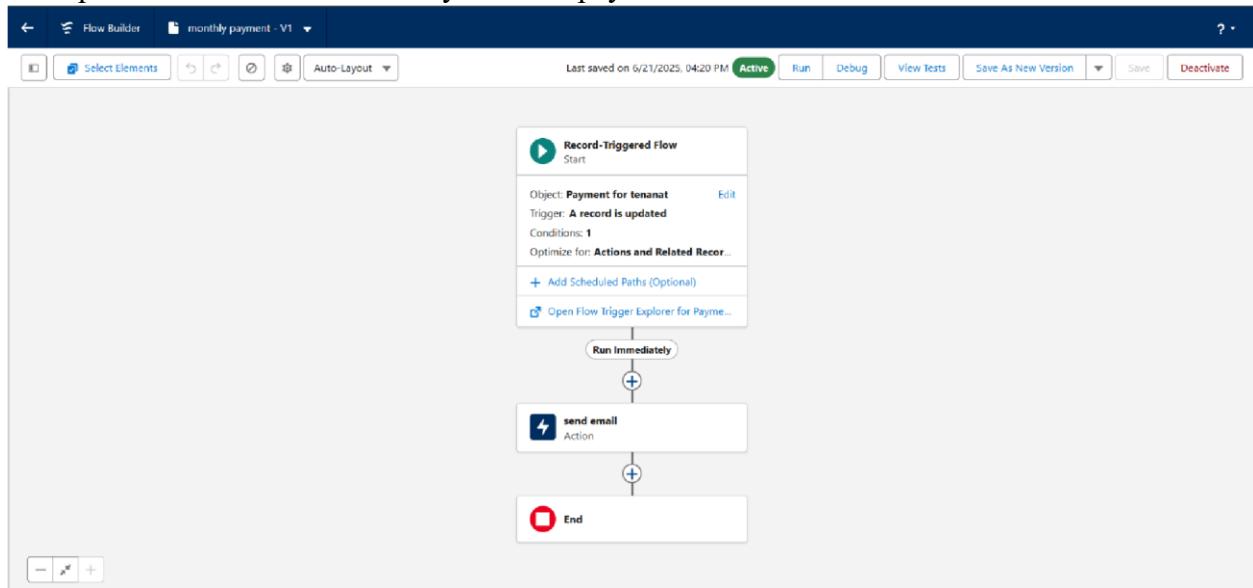
- Analytics Cloud Integration User
- Analytics Cloud Security User
- Anypoint Integration
- Authenticated Website
- Authenticated Website
- B2B Reordering Portal Buyer Profile
- Contract Manager
- Custom: Marketing Profile
- Custom: Sales Profile
- Custom: Support Profile
- Customer Community Login User

Selected Profiles

- System Administrator

The screenshot shows a Salesforce Lightning interface. At the top, there's a navigation bar with icons for back, forward, search, and other system functions. Below it, the page title is "Lease Management" with a sub-tab "Payment". A breadcrumb trail shows "Tenants > property > lease". The main content area has a header "Recently Viewed" with a "Payment" icon. It displays a list of 5 items, each with a checkbox and a name: 1. Rahul, 2. Jack, 3. Raj, 4. Sam, 5. Lahari. To the right of the list are several standard Salesforce list actions: New, Import, Change Owner, Assign Label, Search this list, and various edit and delete icons.

- Implemented Flows for monthly rent and payment success

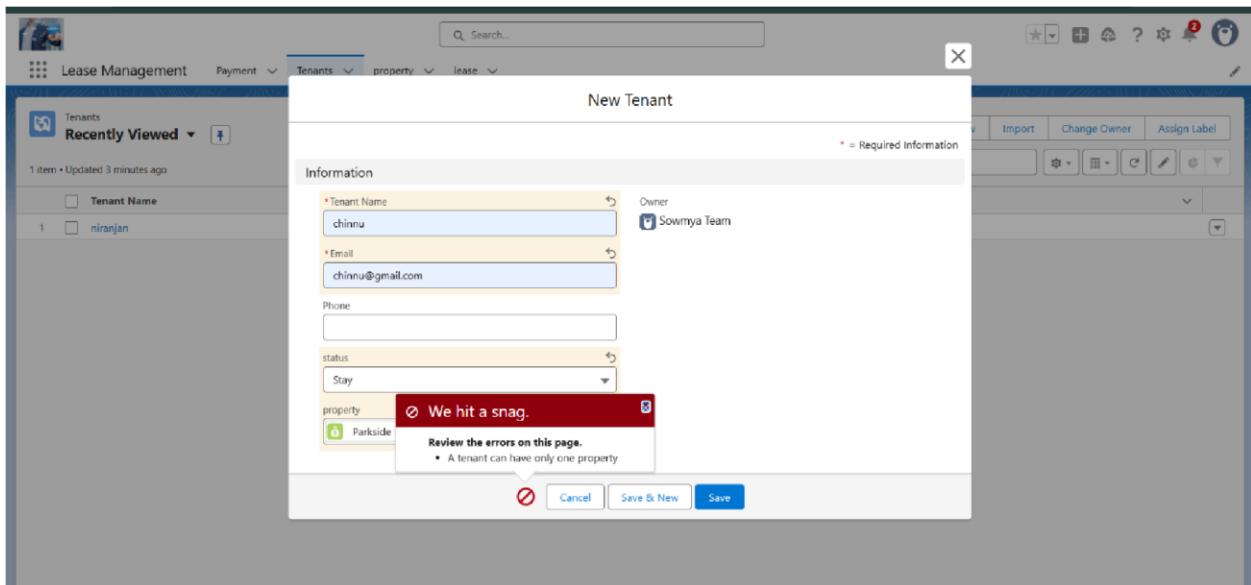


- To create a validation rule to a silver&gold purpose

The screenshot shows the 'Validation Rule Edit' page for the 'lease' object. The 'Rule 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 'Description' field is empty.

The screenshot shows the 'Validation Rule Detail' page for the 'lease Validation Rule'. The rule name is 'lease_end_date'. The error condition formula is 'End_date_c <= start_date_c'. The error message is 'Your End date must be greater than start date'. The error location is 'start date'. The rule was created by 'Sowmya Team' on 6/19/2025, 5:37 AM, and modified by 'Sowmya Team' on 6/26/2025, 7:47 AM.

- Added Apex trigger to restrict multiple tenants per property



- Scheduled monthly reminder emails using Apex class

```

File -> Edit -> Debug -> Test -> New Test -> Run
Test Class: MonthlyEmailScheduler
Code Coverage: 44%
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 ensures the smooth functioning of our rental arrangement and helps maintain';
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

```

- 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 'Classic Email Templates', the 'Leave approved' template is displayed. The template details are as follows:

Email Template Detail

- Email Template Name: Leave approved
- Template Unique Name: Leave_approved
- Encoding: Unicode (UTF-8)
- Author: Sowmya Team [Change]
- Description: Created By: Sowmya Team, 6/20/2025, 1:08 AM
- Modified By: Sowmya Team, 6/20/2025, 1:08 AM
- Available For Use: ✓
- Last Used Date: Times Used: 0

Email Template

Subject: Leave approved

Plain Text Preview:

```
dear{!Tenant__c.Name}.
```

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

Your leave is approved. You can leave now.

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Classic Email Templates', the 'tenant leaving' template is displayed. The template details are as follows:

Email Template Detail

- Email Template Name: tenant leaving
- Template Unique Name: tenant_leaving
- Encoding: Unicode (UTF-8)
- Author: Sowmya Team [Change]
- Description: Created By: Sowmya Team, 6/20/2025, 1:06 AM
- Modified By: Sowmya Team, 6/20/2025, 1:06 AM
- Available For Use: ✓
- Last Used Date: Times Used: 0

Email Template

Subject: request for approve the leave

Plain Text Preview:

```
Dear {Tenant__c.CreatedBy}.
```

Please approve my leave

The screenshot shows the Salesforce Setup interface with the search bar set to "email template". The main content area displays the "Classic Email Templates" page for a "Text Email Template" named "Leave rejected".

Email Template Detail:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Soumya_Team [Change]
Description	Created By: Soumya_Team, 6/20/2025, 1:11 AM
Created By	Soumya_Team, 6/20/2025, 1:11 AM
Modified By	Soumya_Team, 6/20/2025, 1:11 AM

Email Template Preview:

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
```

The screenshot shows the Salesforce Setup interface with the search bar set to "email template". The main content area displays the "Classic Email Templates" page for a "Text Email Template" named "Tenant Email".

Email Template Detail:

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Tenant_Email
Template Unique Name	Tenant_Email
Encoding	Unicode (UTF-8)
Author	Soumya_Team [Change]
Description	Created By: Soumya_Team, 6/20/2025, 1:12 AM
Created By	Soumya_Team, 6/20/2025, 1:12 AM
Modified By	Soumya_Team, 6/20/2025, 1:12 AM

Email Template Preview:

Subject: Urgent: Monthly Rent Payment Reminder

Plain Text Preview:

```
Dear {[Tenant__c.Name]},  
  
I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.
```

The screenshot shows the Salesforce Setup interface with the 'Email' section selected. Under 'Email', 'Classic Email Templates' is chosen. A search bar at the top left shows 'email template'. The main content area displays a 'Text Email Template' named 'tenant payment'. The 'Email Template Detail' section shows the following information:

- Email Templates from Salesforce: Unfiled Public Classic Email Templates
- Email Template Name: tenant payment
- Template Unique Name: tenant_payment
- Encoding: Unicode (UTF-8)
- Author: Scamya Team [Change]
- Description: Created By: Scamya Team, 6/20/2025, 1:13 AM
- Available For Use: checked
- Last Used Date: Times Used: 0
- Modified By: Scamya Team, 6/20/2025, 1:13 AM

The 'Email Template' section below shows the subject 'Confirmation of Successful Monthly Payment' and a plain text preview:

```

Subject : Confirmation of Successful Monthly Payment
Plain Text Preview
Dear {Tenant__c_Email__c}.

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

```

- Approval Process creation

For Tenant Leaving:

The screenshot shows the Salesforce Setup interface with the 'Data' section selected. Under 'Data', 'Approval Processes' is chosen. A search bar at the top left shows 'approval'. The main content area displays an 'Approval Process' named 'TenantApproval'. The 'Process Definition Detail' section shows the following information:

- Process Name: TenantApproval
- Unique Name: TenantApproval
- Description:
- Entry Criteria: tenant__c.status EQUALS Slay
- Record Editability: Administrator ONLY
- Active: checked
- Next Automated Approver Determined By:
- Allow Submitters to Recall Approval Requests: unchecked
- Created By: Scamya Team, 6/23/2025, 3:41 AM
- Modified By: Scamya Team, 6/26/2025, 11:57 PM

The 'Initial Submission Actions' section shows:

Action Type	Description
Record Lock	Lock the record from being edited

The 'Approval Steps' section shows:

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Step 1		User:Scamya Team		Final Rejection

For Check for Vacant:

The screenshot shows the Salesforce Setup interface with the 'Approval Processes' page open. The process name is 'check for vacant'. The 'Process Definition Detail' section includes fields like Unique Name ('check_for_vacant'), Description ('Tenant: check for vacant'), Entry Criteria ('Tenant: status EQUALS Leaving'), and Recipient ('Administrator ONLY'). The 'Initial Submission Actions' section contains an action 'Record Lock' with a description 'Lock the record from being edited'. The 'Approval Steps' section shows one step named 'step1' assigned to 'User:Sowmya Team'. The page also includes a sidebar with various setup categories and a global search bar.

- Apex Trigger

Create an Apex Trigger

The screenshot shows the Salesforce Developer Console with an Apex trigger named 'testHandler'. The code defines a trigger on the 'Tenant_c' object that prevents insertions if the record is being created before. A screenshot of the 'Open' dialog is overlaid on the code editor, showing the 'Triggers' entity type selected. The developer console also displays tabs for Logs, Tests, Checkpoints, Query Editor, View Status, Progress, and Problems.

```

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

```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage. The tab bar shows 'test.apex' is active. The code editor contains the following Apex trigger:

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

The bottom navigation bar includes 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is selected.

Create an Apex Handler class

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSPage. The tab bar shows 'test.apex' is active. The code editor contains the following Apex class:

```
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.Id);
        }
        for (Tenant__c newTenant : newList) {
            if (newTenant.Property__c != null) {
                newTenantaddError('A');
            }
        }
    }
}
```

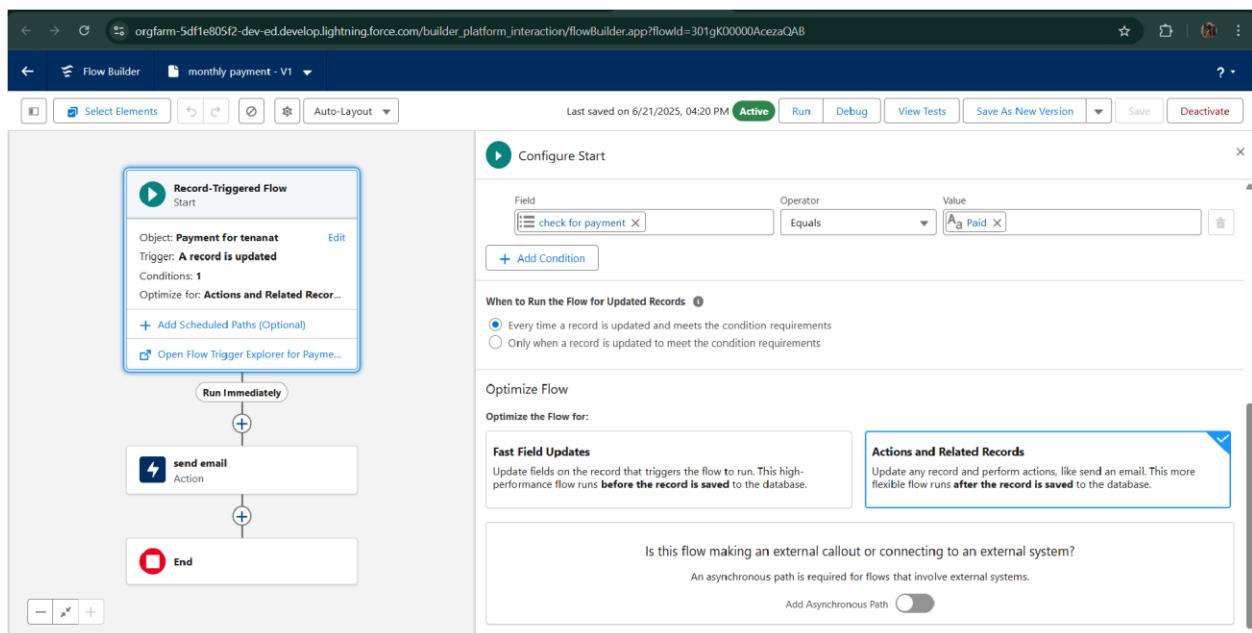
A modal dialog titled 'Open' is displayed over the code editor, listing available entities:

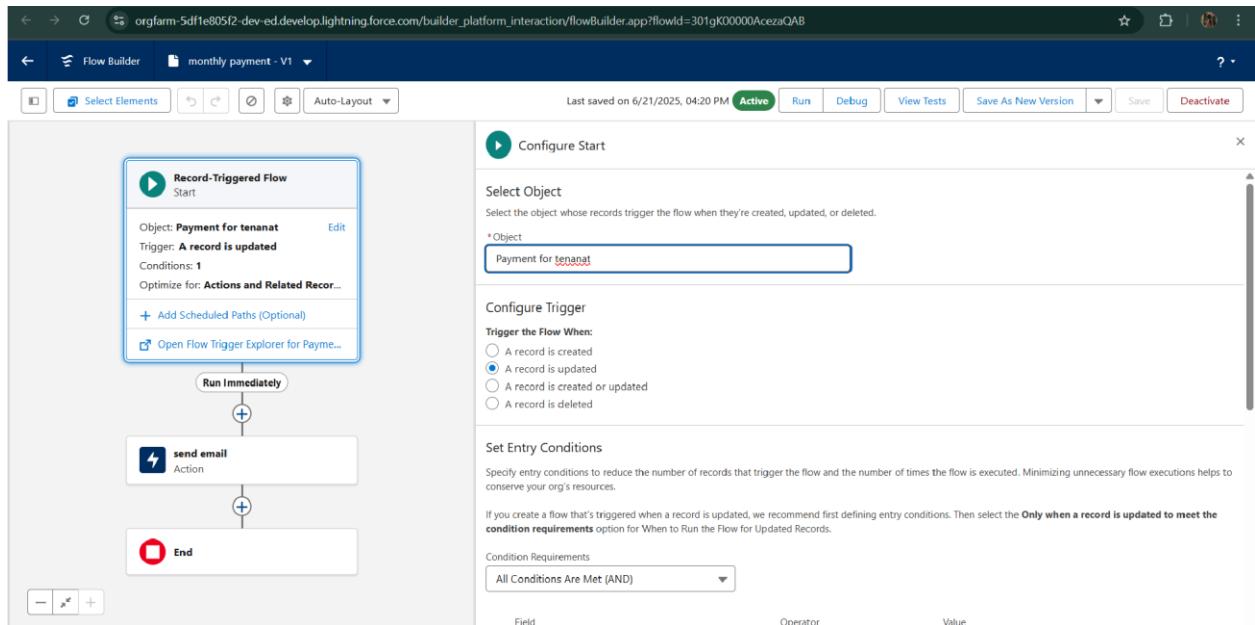
Entity Type	Name	Namespace	Related
Classes	testHandler	MonthlyEmailScheduler	↳ test ApexTrigger References
Triggers			↳ property CustomField References
Page Components			↳ Tenant__c SObject References
Objects			↳ Tenant__c SObject References
Static Resources			
Package			

The bottom navigation bar includes 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Problems' tab is selected.

```
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 }
```

• 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 sendMonthlyEmail(
17
18     List<Tenant__c> tenants = [SELECT
19
20         for (Tenant__c tenant : tenants
21
22             String recipientEmail = tenant.Email__c;
23

```

File Edit Debug Test Workspace Help < >

test.apex testHandler.apex MonthlyEmailScheduler.apc

Code Coverage: Name API Version: 64 Go To

Entity Type Edition Related

Name Namespace Name Extent ConTrigger Reference...

Classes testHandler MonthlyEmailScheduler Email CustomFiel...

Triggers MonthlyEmailScheduler Tenant__c Object References

Pages Tenant__c Subject References

Objects

Static Resources

Packages

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

Logs Tests Checkpoints Query Editor View Status Progress Problems

```

1 *@ISOBEL 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    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 a positive living environment for all.';
24
25            String emailSubject = 'Wesider: 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            Messaging.SingleEmailMessage[] emails = new Messaging.SingleEmailMessage[]{email};
36
37            Messaging.sendEmail(emails);
38
39        }
40    }
41
42 }

```

Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The sidebar on the left has a search bar with 'apex' typed in. Under 'Apex Classes', the 'MonthlyEmailScheduler' class is selected. The main area displays the Apex code for the class:

```

1 *@ISOBEL 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    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 a positive living environment for all.';
24
25            String emailSubject = 'Wesider: 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            Messaging.SingleEmailMessage[] emails = new Messaging.SingleEmailMessage[]{email};
36
37            Messaging.sendEmail(emails);
38
39        }
40    }
41
42 }

```

← → ⌂ orgfarm-75748232a1-dev-ed.develop.lightning.force.com/lightning/o/Report/home?queryScope=mru

Jewellery Inventory ... Reports New Prices Report Report Builder

Reports Recent 1 item

REPORTS Report Name Description Folder Created By Created On Subscribed

Recent New Prices Report Private Reports Nandhitha ramu 8/19/2025, 11:47 PM

Created by Me
Private Reports
Public Reports
All Reports

FOLDERS All Folders
Created by Me
Shared with Me

FAVORITES

A Snipping Tool Screenshot copied to clipboard and saved Select here to mark up and share.

← → ⌂ orgfarm-75748232a1-dev-ed.develop.lightning.force.com/lightning/o/Report/home?queryScope=mru

Jewellery Inventory ... Reports New Prices Report Report Builder

Reports Recent 1 item

REPORTS Report Name Description Folder Created By Created On Subscribed

Recent New Prices Report Private Reports Nandhitha ramu 8/19/2025, 11:47 PM

Created by Me
Private Reports
Public Reports
All Reports

FOLDERS All Folders
Created by Me
Shared with Me

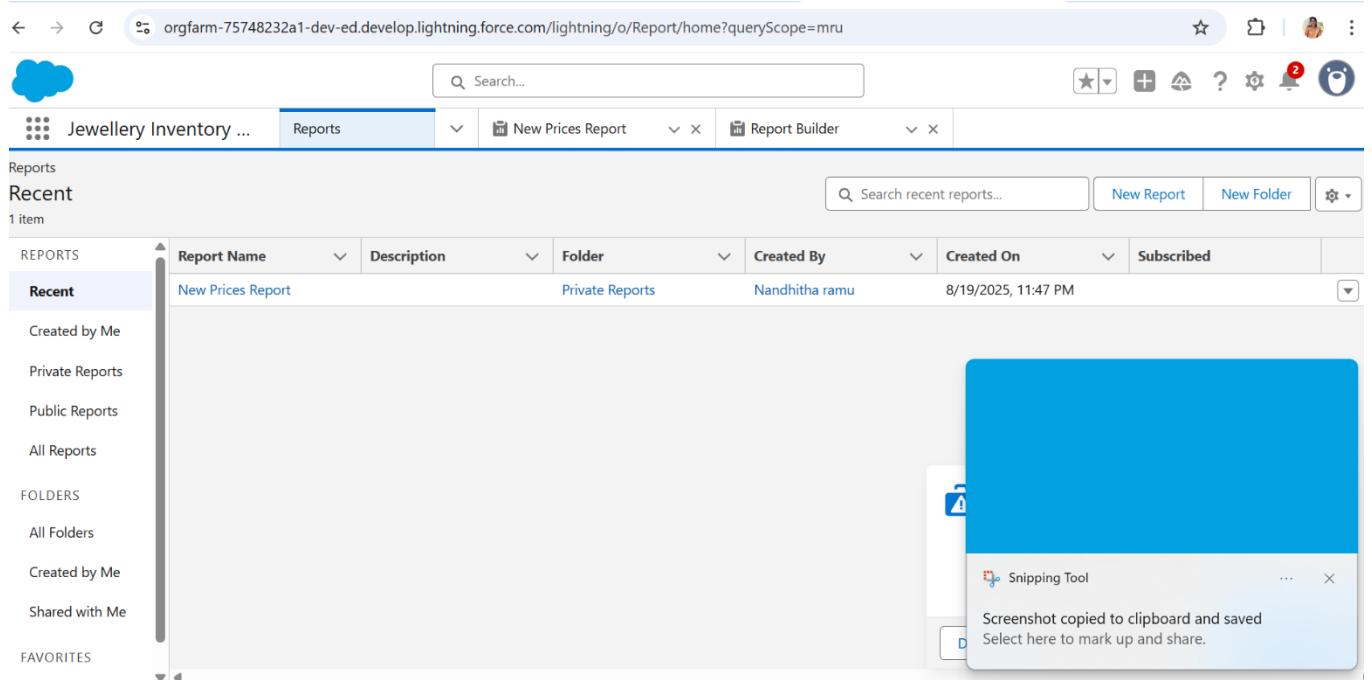
FAVORITES

A Snipping Tool Screenshot copied to clipboard and saved Select here to mark up and share.

FUNCTIONAL AND PERFORMANCE TESTING

Performance Testing

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



← → ⌂ orgfarm-75748232a1-dev-ed.develop.lightning.force.com/lightning/o/Report/home?queryScope=mru

Jewellery Inventory ... Reports New Prices Report Report Builder

Reports Recent 1 item

REPORTS Recent

Report Name	Description	Folder	Created By	Created On	Subscribed
New Prices Report	Private Reports	Nandhitha ramu	8/19/2025, 11:47 PM		

Created by Me
Private Reports
Public Reports
All Reports
FOLDERS
All Folders
Created by Me
Shared with Me
FAVORITES

Snipping Tool Screenshot copied to clipboard and saved Select here to mark up and share.

The screenshot shows the Salesforce Lightning Report Home page. The top navigation bar includes back, forward, and search icons. Below the header, there are tabs for 'Reports' and two recently viewed reports: 'New Prices Report' and 'Report Builder'. A sidebar on the left lists categories like 'Recent', 'Created by Me', 'Private Reports', etc. The main content area displays a table of recent reports with columns for 'Report Name', 'Description', 'Folder', 'Created By', 'Created On', and 'Subscribed'. A single row for 'New Prices Report' is shown. A blue callout box from the 'Snipping Tool' indicates that a screenshot has been copied to the clipboard. The URL in the address bar is 'orgfarm-75748232a1-dev-ed.develop.lightning.force.com/lightning/o/Report/home?queryScope=mru'.

← → ⌂ orgfarm-75748232a1-dev-ed.lightning.force.com/lightning/o/Report/home?queryScope=mru

Cloud icon Search... ⋮

Jewellery Inventory ... Reports New Prices Report Report Builder

Reports Recent 1 item

REPORTS Report Name Description Folder Created By Created On Subscribed

Recent New Prices Report Private Reports Nandhitha ramu 8/19/2025, 11:47 PM

Created by Me

Private Reports

Public Reports

All Reports

FOLDERS All Folders

Created by Me

Shared with Me

FAVORITES

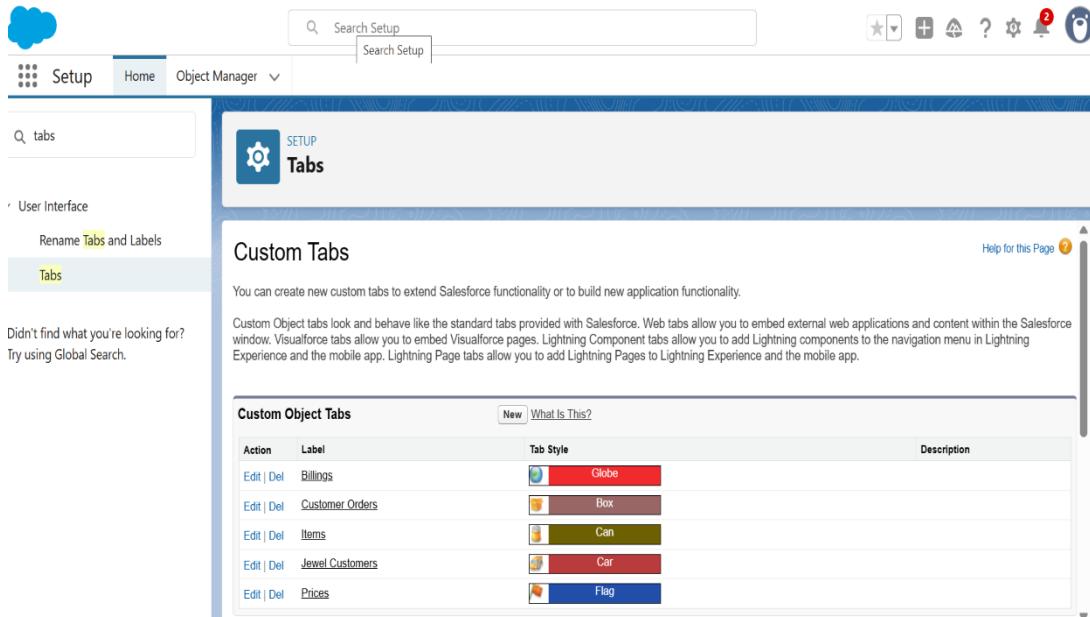
Snipping Tool Screenshot copied to clipboard and saved Select here to mark up and share.

Report Name	Description	Folder	Created By	Created On	Subscribed
New Prices Report	Private Reports	Nandhitha ramu	8/19/2025, 11:47 PM		

RESULTS :

Output Screenshots :

Tabs for billing, customer, orders items, jewel customers, prices.



The screenshot shows the Salesforce Setup interface with the 'Custom Tabs' page selected. The left sidebar has 'User Interface' expanded, with 'Rename Tabs and Labels' and 'Tabs' selected. A search bar at the top left contains 'tabs'. The main content area is titled 'Custom Tabs' and includes a brief description of what custom tabs are. Below this is a table titled 'Custom Object Tabs' listing five tabs:

Action	Label	Tab Style	Description
Edit Del	Billing	Globe	
Edit Del	Customer Orders	Box	
Edit Del	Items	Can	
Edit Del	Jewel Customers	Car	
Edit Del	Prices	Flag	

ADVANTAGES & DISADVANTAGES

Advantages

- **Centralized Data** – All customer, inventory, and billing records stored in one platform.
- **Automation** – Flows and approval processes save time and reduce errors.
- **Scalability** – Easily adaptable as the jewelry business grows.
- **Real-Time Reports** – Sales, stock, and payment insights available anytime.
- **Customer Engagement** – Better communication through email alerts and history tracking.

Disadvantages

- **Implementation Cost** – Salesforce licenses and customization can be expensive.
- **Training Required** – Staff needs to be trained to use Salesforce effectively.

- **✗ Customization Effort** – Developers must build custom objects, flows, and automation.
- **✗ Dependency on Internet** – Requires stable connectivity to access data.
- **✗ Ongoing Maintenance** – Regular updates, bug fixes, and support needed.

CONCLUSION:

- The Jewel Management CRM system provides a structured and automated solution for managing jewelry customers, inventory, pricing, orders, and billing within Salesforce. By integrating automation, approval processes, and reporting, it reduces manual effort, improves accuracy, and enhances customer relationships.
-