



A CRM Application to Manage the Mall

College: 7155 - PSG Institute of Technology and Applied Research

Team Lead: Sridhar R - 84BCCE07A4B2DE6817F4A42A8154296E

Team Id: NM2024TMID00656

Team Members:

Bhavadharini C	-	892609CC0580BBB8F1246594F016266B
Gayathri C	-	A8F6D996BF44D94A97A47E2FB8854ACC
Krishnaprabha K R	-	2D17CB2ABD7AA8780354D382B2342DC9
Sweda A S	-	336612FAD4217B1B089411E11C6B41E2
Sridhar R	-	84BCCE07A4B2DE6817F4A42A8154296E

1. Project Overview

This project is focused on the Mall Management CRM Application, designed to address the challenge of efficiently managing tenant relationships, lease tracking, and mall operations. The goal is to deliver a comprehensive solution by leveraging the Salesforce platform to centralize tenant information, streamline communication, and automate key processes. Through this project, we aim to enhance operational efficiency, tenant satisfaction, and data accuracy and support the long-term goals of optimizing space utilization, improving decision-making, and ensuring sustainable mall management.

2. Objectives

Business Goals:

- To improve the efficiency and transparency of mall operations by leveraging Salesforce. The project aims to optimize lease management, enhance tenant satisfaction, reduce administrative overhead, and support data-driven decision-making, ultimately contributing to the smooth functioning and long-term success of the mall.

Specific Outcomes:

- Automated Lease Tracking: A centralized system that tracks lease agreements, renewal dates, and rental payments to avoid manual errors and missed deadlines.
- Tenant Management Dashboard: A user-friendly dashboard providing real-time access to tenant information, communication history, and support requests.
- Maintenance and Service Tracking: Integration of a maintenance request and tracking system, ensuring all tenant service needs are logged and addressed promptly.
- Real-Time Reporting and Analytics: Custom reports and analytics to monitor mall performance metrics, such as space utilization, rental income, and tenant satisfaction.
- Improved Operational Workflows: Automation of tasks like payment processing, tenant communication, and lease renewals to reduce manual intervention and save time.

3. Salesforce Key Features and Concepts Utilized

This project leverages key Salesforce functionalities and concepts to optimize mall management operations:

1. Custom Objects and Relationships

- Created custom objects like Tenant, Lease Tracking, and Tenant Issues to organize and track all relevant data related to tenants, lease agreements, and reported issues.

- Defined relationships between these objects, including lookup relationships, to ensure smooth data flow and consistency across all records.

2. Tabs and Lightning App (Mall Management CRM)

- Set up custom tabs for each object (Tenant, Lease Tracking, Tenant Issues) to provide easy access and navigation within the Mall Management CRM Lightning app.
- Customized the Lightning app to streamline user access, optimize the interface, and enhance overall navigation and branding.

3. Screen Flows

- Developed a Tenant Registration Form using Screen Flow to capture important tenant details such as name, contact information, and lease terms. This ensures efficient data entry and organization for tenant management.

4. Apex Triggers

- Created Apex triggers to automate business logic, such as sending email notifications when a tenant has not paid 50% of the total rent by the due date, and showing an error message if an invalid PAN card number is entered.
- These triggers help automate processes and ensure timely and accurate actions are taken.

5. Users and Public Groups

- Created specific users and public groups for different roles, such as Mall Managers, Tenant Support Teams, and Finance Team, to ensure proper access control and collaboration across the organization.
- This ensures that users only have access to the data they need while maintaining security and efficiency.

6. Reports and Dashboards

- Designed custom report types and reports for Lease Management, Tenant Issues, and Tenant Records to monitor key performance metrics such as lease renewal dates, unresolved issues, and tenant payment status.
- Created a Dashboard integrated with the homepage to provide real-time insights into these metrics, offering a comprehensive view of mall operations and tenant satisfaction.

7. Homepage Integration

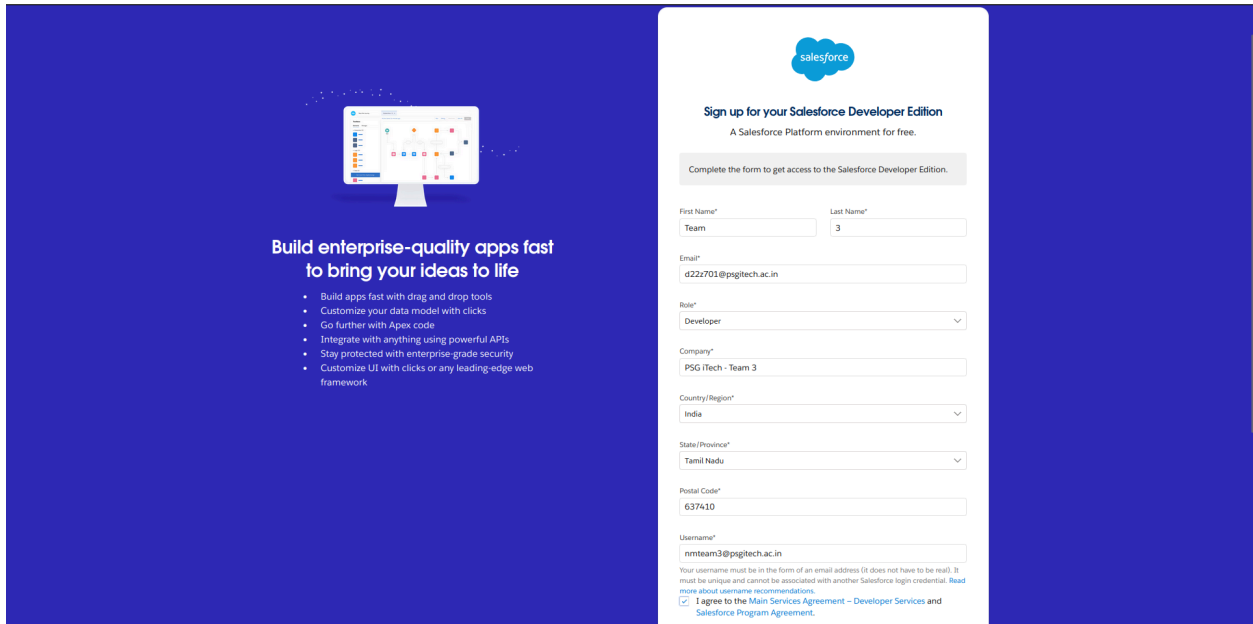
- Integrated the Tenant and Lease Management Dashboard on the homepage for centralized data monitoring, allowing mall management to track key operational metrics, identify trends, and make informed decisions at a glance.

These Salesforce features collectively enhance the efficiency and transparency of mall management, enabling better decision-making, improved tenant relationships, and streamlined operations.

4. Detailed Steps to Solution Design

1. Created Salesforce Developer Account

We began by creating a Salesforce Developer Account to access the development environment, enabling us to customize and build the application. This setup provided the necessary tools and resources to design, test, and deploy the CRM system for mall management.



Build enterprise-quality apps fast to bring your ideas to life

- Build apps fast with drag and drop tools
- Customize your data model with clicks
- Go further with Apex code
- Integrate with anything using powerful APIs
- Stay protected with enterprise-grade security
- Customize UI with clicks or any leading-edge web framework

Sign up for your Salesforce Developer Edition
A Salesforce Platform environment for free.

Complete the form to get access to the Salesforce Developer Edition.

First Name* Last Name*
Team 3

Email*
d22z701@psgitech.ac.in

Role*
Developer

Company*
PSG Tech - Team 3

Country/Region*
India

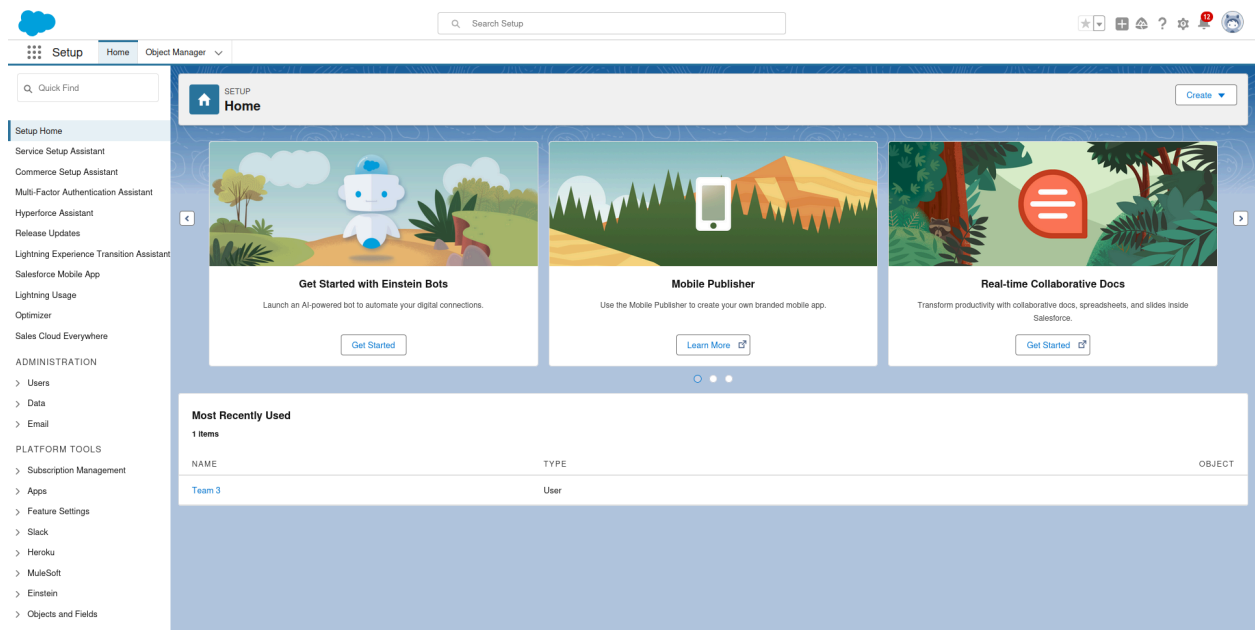
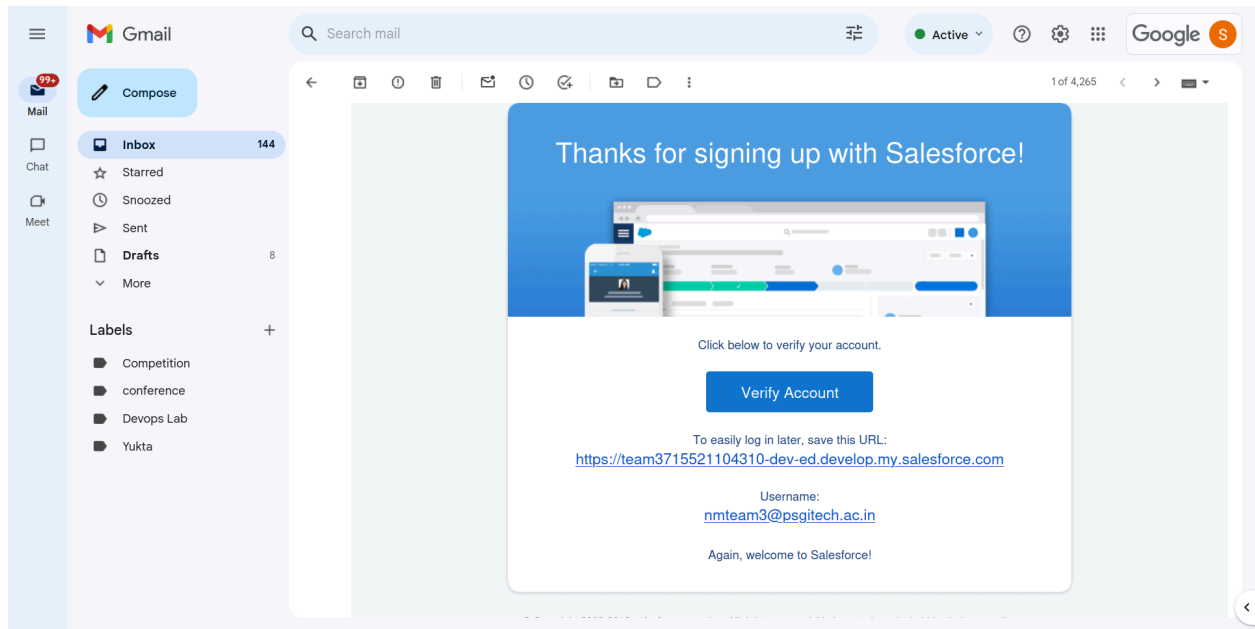
State/Province*
Tamil Nadu

Postal Code*
637410

Username*
nmteam3@psgitech.ac.in

Your username must be in the form of an email address (it does not have to be real). It must be unique and cannot be associated with another Salesforce login credential. [Read more about username recommendations.](#)

☒ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement.



2. Create Custom Objects

We created three main custom objects—Tenant, Lease Tracking, and Tenant Issues—in Object Manager.

- Tenant stores tenant-specific information, such as name, contact details, and lease terms.

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, Scoping Rules, Triggers, Flow Triggers, and Validation Rules. The 'Details' section is selected, showing the following configuration:

Field	Value
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

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

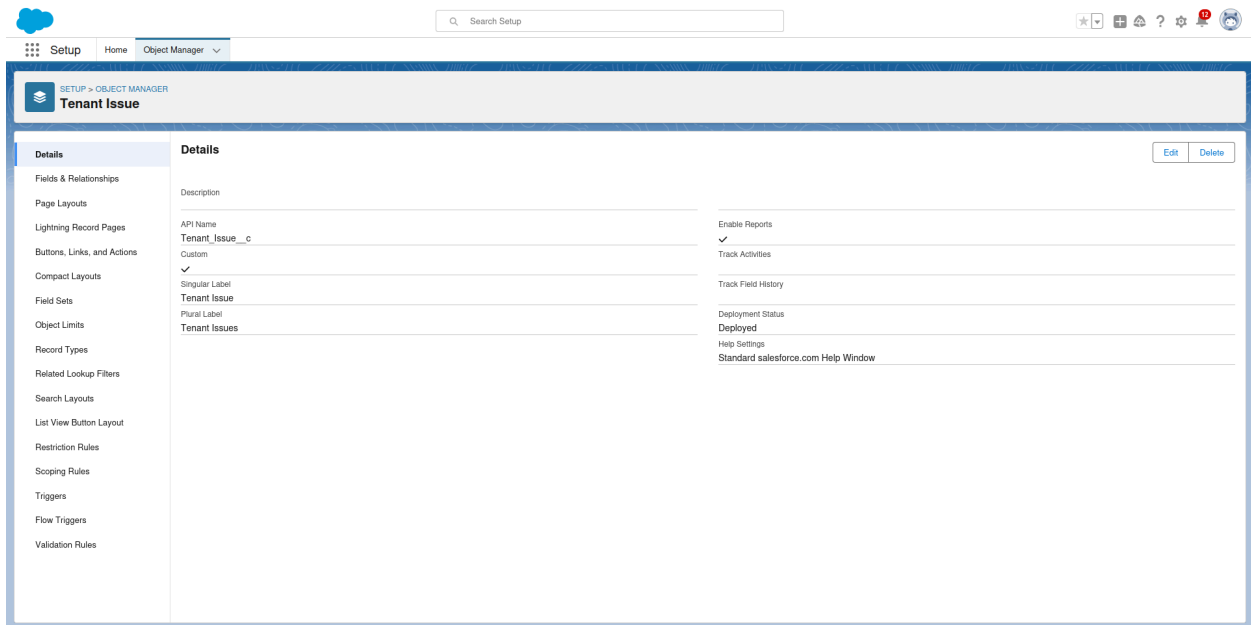
- Lease Tracking records lease details, such as lease agreement numbers, renewal dates, and payment status.

The screenshot shows the Salesforce Setup interface for the 'Lease Tracking' 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, Scoping Rules, Triggers, Flow Triggers, and Validation Rules. The 'Details' section is selected, showing the following configuration:

Field	Value
Description	
API Name	Lease_Tracking__c
Custom	✓
Singular Label	Lease Tracking
Plural Label	Leases Tracking
Enable Reports	✓
Track Activities	✓
Track Field History	
Deployment Status	Deployed
Help Settings	Standard salesforce.com Help Window

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

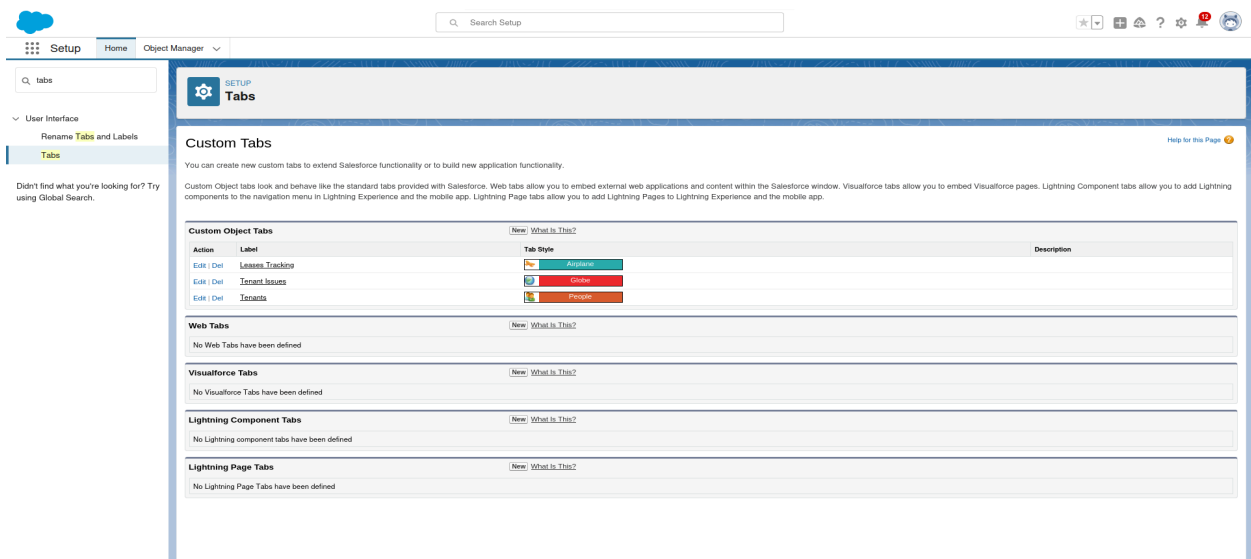
- Tenant Issues tracks tenant-reported issues, including issue type, status, and resolution progress.



3. Tabs

We set up tabs for each custom object to ensure easy navigation within the CRM. The tab created for Tenant allows users to access and manage these objects directly. This setup improves user experience by simplifying the flow of data management.

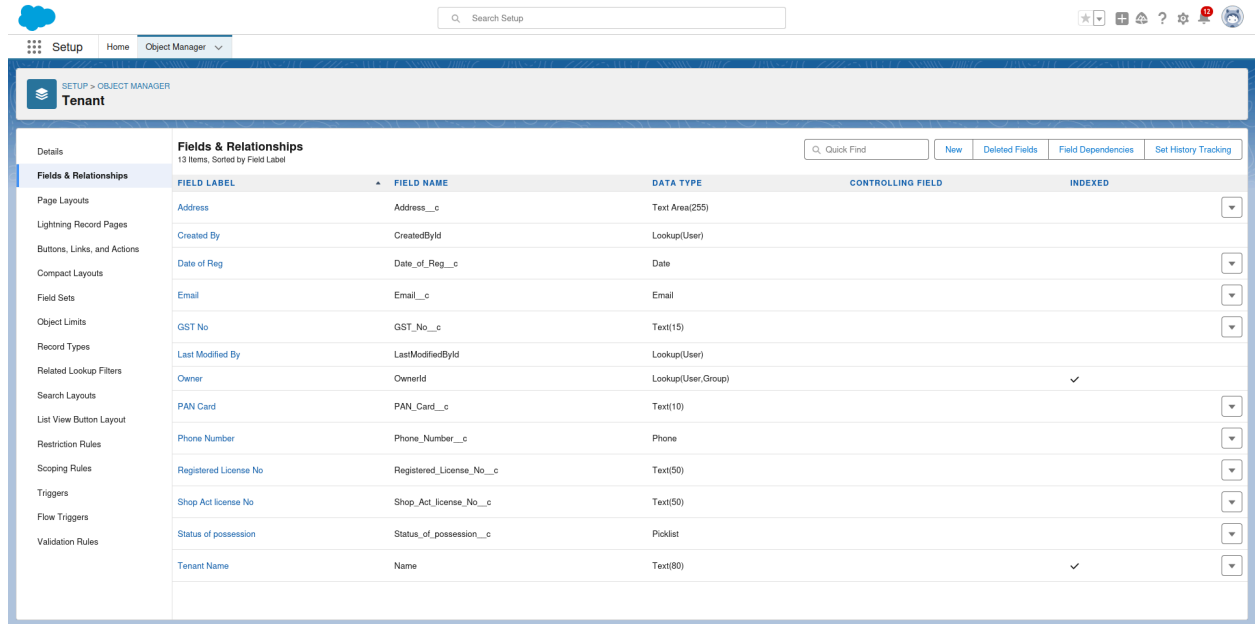
Create a custom tab for tenant object



4. Create Fields and Relationships

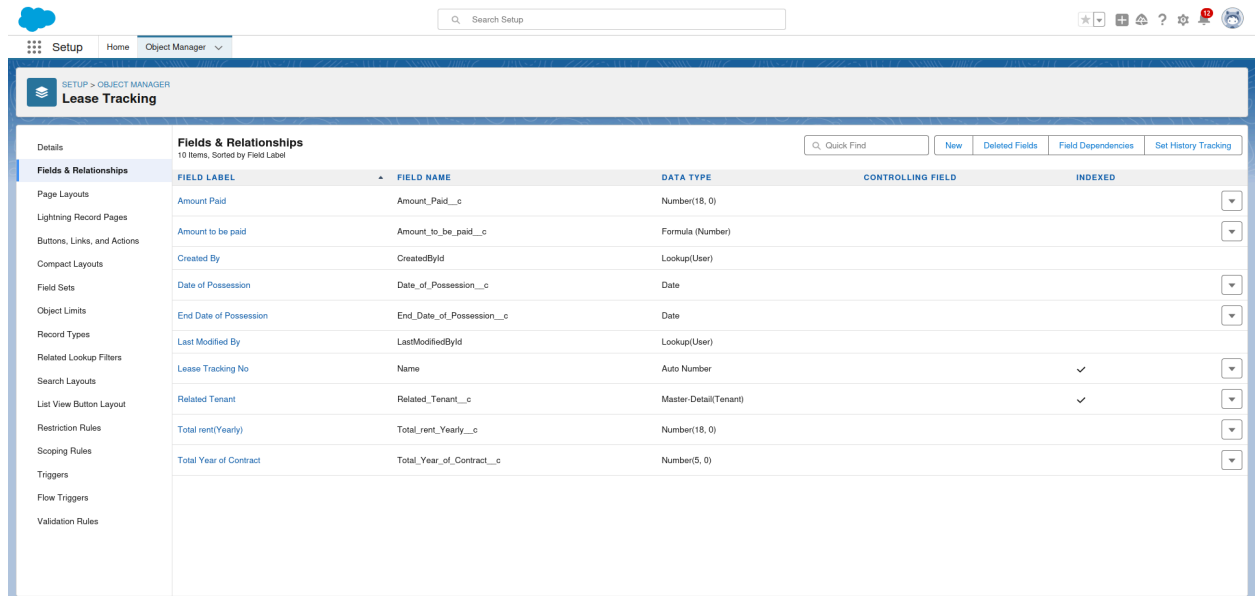
Custom fields were added to each object to capture necessary details:

- Tenant Object: Fields like tenant name, contact information, lease start and end dates.



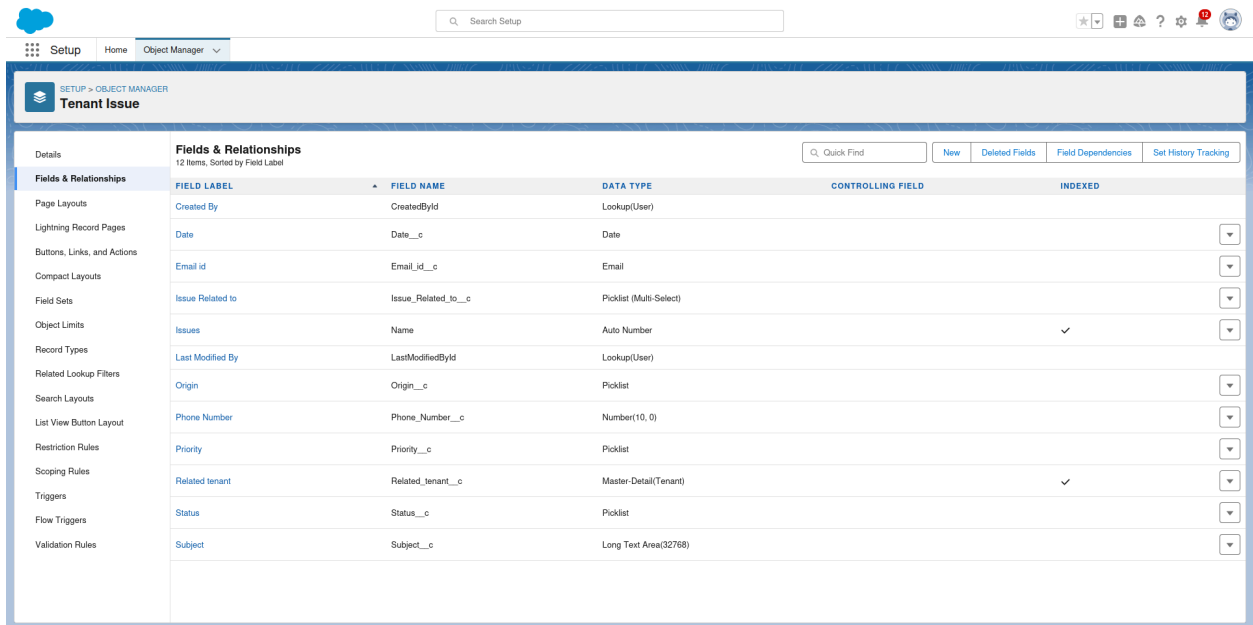
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text Area(255)		
Created By	CreatedBy	Lookup(User)		
Date of Reg	Date_of_Reg__c	Date		
Email	Email__c	Email		
GST No	GST_No__c	Text(15)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
PAN Card	PAN_Card__c	Text(10)		
Phone Number	Phone_Number__c	Phone		
Registered License No	Registered_License_No__c	Text(50)		
Shop Act license No	Shop_Act_license_No__c	Text(50)		
Status of possession	Status_of_posession__c	Picklist		
Tenant Name	Name	Text(80)		✓

- Lease Tracking Object: Fields like lease agreement number, lease type, renewal dates, payment status.



FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount Paid	Amount_Paid__c	Number(18, 0)		
Amount to be paid	Amount_to_be_paid__c	Formula (Number)		
Created By	CreatedBy	Lookup(User)		
Date of Possession	Date_of_Possession__c	Date		
End Date of Possession	End_Date_of_Possession__c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
Lease Tracking No	Name	Auto Number		✓
Related Tenant	Related_Tenant__c	Master-Detail(Tenant)		✓
Total rent (Yearly)	Total_rent_Yearly__c	Number(18, 0)		
Total Year of Contract	Total_Year_of_Contract__c	Number(5, 0)		

- Tenant Issues Object: Fields like issue description, status, and priority level.



The screenshot shows the Salesforce Setup interface. At the top, there's a navigation bar with 'Setup', 'Home', and 'Object Manager'. Below this, the 'Tenant Issue' object is selected. The main content area is titled 'Fields & Relationships' and shows a list of 12 items, sorted by Field Label. The table below lists the fields and their properties.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Date	Date__c	Date		
Email Id	Email_Id__c	Email		
Issue Related to	Issue_Related_to__c	Picklist (Multi-Select)		
Issues	Name	Auto Number		✓
Last Modified By	LastModifiedById	Lookup(User)		
Origin	Origin__c	Picklist		
Phone Number	Phone_Number__c	Number(10, 0)		
Priority	Priority__c	Picklist		
Related tenant	Related_tenant__c	Master-Detail(Tenant)		✓
Status	Status__c	Picklist		
Subject	Subject__c	Long Text Area(32768)		

We also implemented validation rules to maintain data integrity, such as ensuring that the lease end date is always later than the lease start date.

5. Create the Lightning App

We created a custom Lightning App named "SmartMall" to consolidate all the core functionalities. The app was configured with specific branding, navigation items, and user access settings, providing a user-friendly interface that simplifies the management of tenants, leases, and tenant issues.

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

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 ⓘ

SmartMall

* Developer Name ⓘ

SmartMall

Description ⓘ

Enter a description...

App Branding

Image ⓘ



Primary Color Hex Value ⓘ

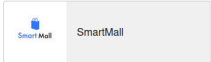
#0070D2

Clear

Org Theme Options

☐ Use the app's image and color instead of the org's custom theme

App Launcher Preview



SmartMall

Tenants ▾

Leases Tracking ▾

Tenant Issues ▾

Q Search...



Tenants

Recently Viewed ▾ ⓘ

1 item • Updated a minute ago

New

Import

Change Owner

Assign Label

Q Search this list...



☐ Tenant Name

1 ☐ Sidhar

SmartMail

Tenants

Leases Tracking

Tenant Issues

Search...

Star

Grid

Cloud

Help

15

Profile

Tenants

Recently Viewed

1 item • Updated a few seconds ago

Tenant Name

1

Sridhar

Information

* = Required Information

Tenant Name

Owner

Team 3

Phone Number

Address

PAN Card

Date of Reg

Email

GST No

Registered License No

Shop Act license No

Status of possession

--None--

Cancel

Save & New

Save

SmartMail

Tenants

Leases Tracking

Tenant Issues

Search...

Star

Grid

Cloud

Help

15

Profile

Leases Tracking

Recently Viewed

0 items • Updated a few seconds ago

Lease Tracking No

New Lease Tracking

* = Required Information

Lease Tracking No

Related Tenant

Search Tenants...

Date of Possession

End Date of Possession

Total Year of Contract

Total rent/Yearly

Amount Paid

Cancel

Save & New

Save

New Tenant Issue

Information

Report Target

Issue Related To

1) ELECTRICITY
2) INFRASTRUCTURE
3) PLUMBING
4) RENT
5) OTHER

Subject

Phone Number

Status

Priority

Origin

Email ID

Date

10/10/2024

Cancel Save & New Save

6. Record Insertion

Tenant Object:

We created records in the Tenant object by filling in the necessary fields, ensuring validation rules for phone numbers and registration dates were respected. We successfully inserted tenant records.

Tenant Sridhar

Related Details

Tenant Name Sridhar

Phone Number 9874563210

Address Hello World

PAN Card ASDFGA

Date of Reg 10/10/2024

Email abc@gmail.com

GST No 9876543210

Registered License No 789456313219687ygh

Shop Act license No 654986146876tgh

Status of possession Pending

Created By Team 3, 10/10/2024, 12:37 am

Last Modified By Team 3, 10/10/2024, 12:37 am

New Contact Edit New Opportunity

Lease Tracking Object:

We added records to the Lease Tracking object, ensuring each record was associated with a tenant. We validated fields like lease dates and rent amounts.

The screenshot shows the 'Lease Tracking' object details for record 'TT - 000001'. The interface includes a top navigation bar with 'SmartMail', 'Tenants', 'Leases Tracking', and 'Tenant Issues'. A search bar is present in the top right. The main content area is divided into 'Related' and 'Details' tabs. The 'Details' tab is active, showing a form with the following fields:

Field	Value	Action
Lease Tracking No	TT - 000001	
Related Tenant	Sridhar	
Date of Possession	11/12/2024	Edit
End Date of Possession	10/12/2030	Edit
Total Year of Contract	5	Edit
Total rent(Yearly)	10,00,000	Edit
Amount Paid	1,00,000	Edit
Amount to be paid	9,00,000	
Created By	Team 3, 10/10/2024, 12:45 am	
Last Modified By	Team 3, 10/10/2024, 12:45 am	

Buttons for 'New Contact', 'Edit', and 'New Opportunity' are located in the top right corner of the details view.

Tenant Issues Object:

We entered data into the Tenant Issues object, linking each record to a tenant. We have inserted records for tenant-related issues.

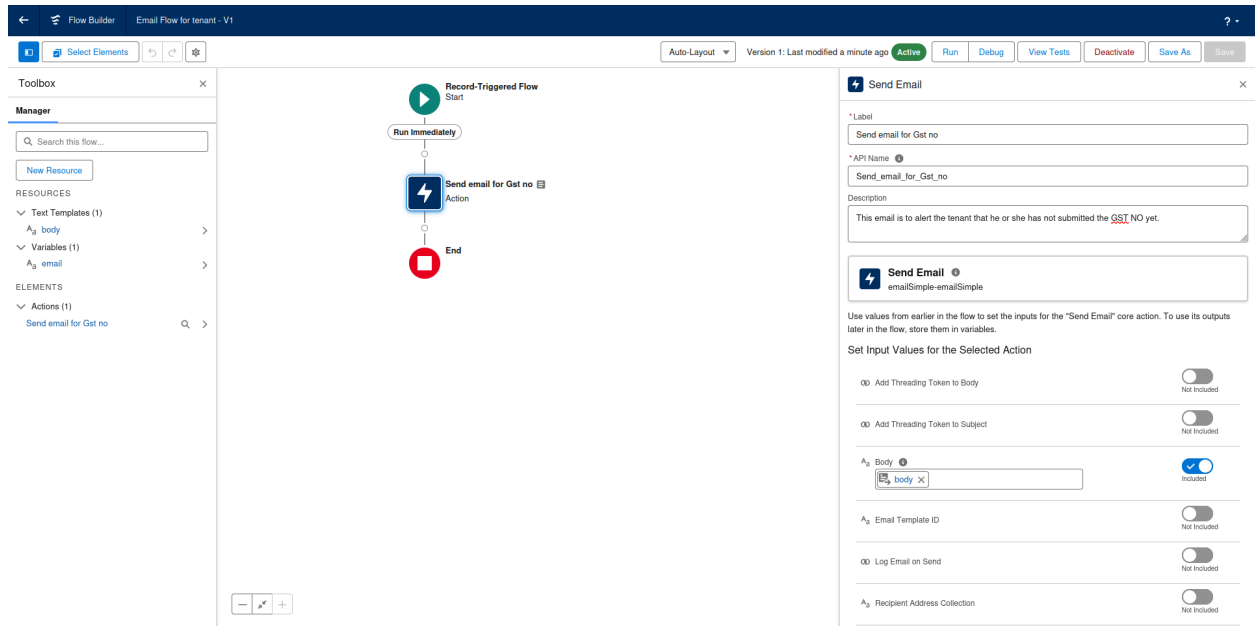
The screenshot shows the 'Tenant Issues' object details for record 'TI - 000001'. A green notification banner at the top states 'Tenant Issue "TI - 000001" was created.' The interface includes a top navigation bar with 'SmartMail', 'Tenants', 'Leases Tracking', and 'Tenant Issues'. A search bar is present in the top right. The main content area is divided into 'Related' and 'Details' tabs. The 'Details' tab is active, showing a form with the following fields:

Field	Value	Action
Issues	TI - 000001	
Related tenant	Sridhar	
Issue Related to	1) ELECTRICITY	Edit
Subject		Edit
Phone Number	9,87,45,63,210	Edit
Status	Not contacted	Edit
Priority	High	Edit
Origin		Edit
Phone		Edit
Email id	abc@gmail.com	Edit
Date	10/10/2024	Edit
Created By	Team 3, 10/10/2024, 12:48 am	
Last Modified By	Team 3, 10/10/2024, 12:48 am	

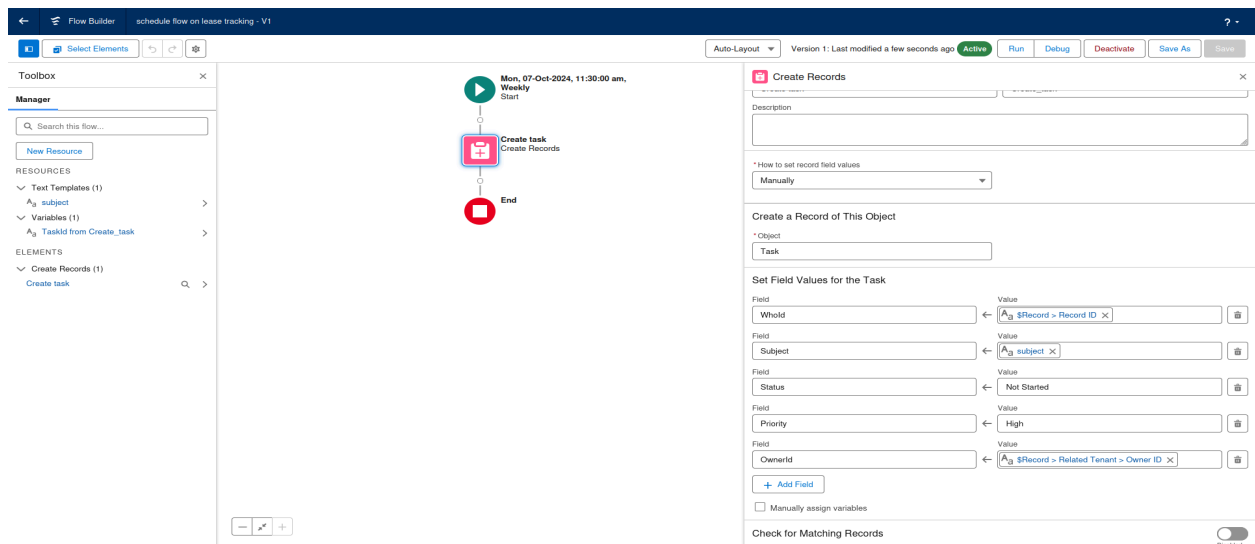
Buttons for 'New Contact', 'Edit', and 'New Opportunity' are located in the top right corner of the details view.

7. Create Flows

Created a Record-Triggered Flow on the Tenant Object to send an email if a new tenant record had an empty GST No. field. Configured conditions to check for a missing GST No., set up an email alert action, and activated the flow.



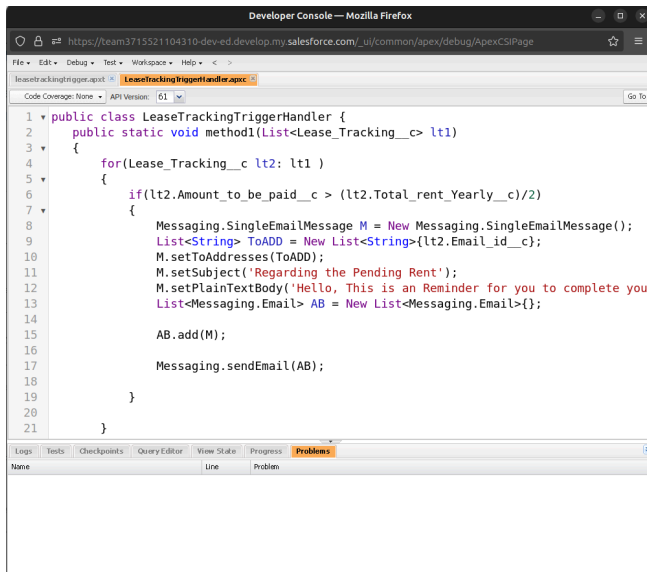
Set up a Schedule-Triggered Flow on the Lease Management Object to create a task if the possession end date was within the next year. Scheduled it to run weekly on Mondays, defined conditions for the end date, created a task with a reminder to contact the manager, and saved and activated the flow.



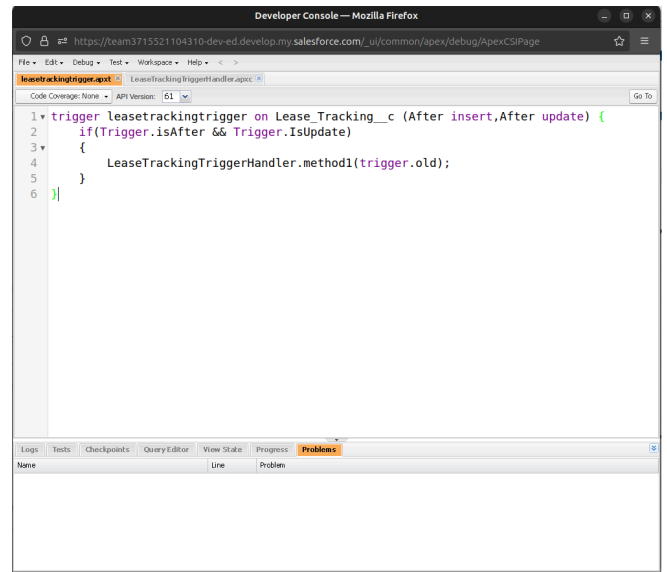
8. Apex Triggers

Apex Triggers were created to automate key actions:

- A trigger was developed to send email notifications if a tenant's payment is overdue by more than 30 days.

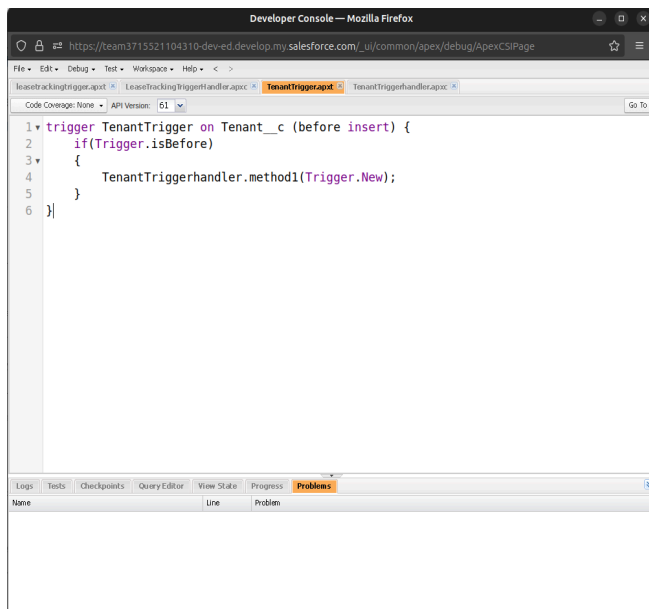


```
1 public class LeaseTrackingTriggerHandler {
2     public static void method1(List<Lease_Tracking__c> lt1)
3     {
4         for(Lease_Tracking__c lt2: lt1)
5         {
6             if(lt2.Amount_to_be_paid__c > (lt2.Total_rent_Yearly__c)/2)
7             {
8                 Messaging.SingleEmailMessage M = New Messaging.SingleEmailMessage();
9                 List<String> ToADD = New List<String>{lt2.Email_id__c};
10                M.setToAddresses(ToADD);
11                M.setSubject('Regarding the Pending Rent');
12                M.setPlainTextBody('Hello, This is an Reminder for you to complete your');
13                List<Messaging.Email> AB = New List<Messaging.Email>{};
14
15                AB.add(M);
16
17                Messaging.sendEmail(AB);
18            }
19        }
20    }
21 }
```

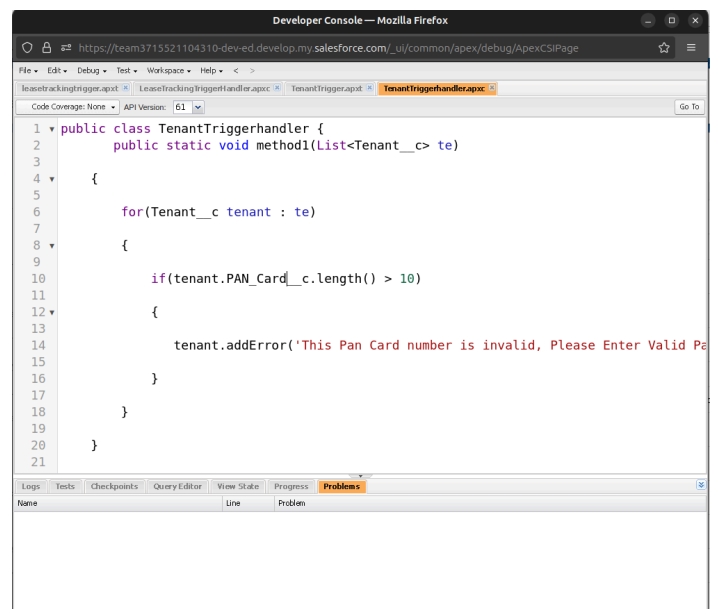


```
1 trigger LeaseTrackingTrigger on Lease_Tracking__c (After insert,After update) {
2     if(Trigger.isAfter && Trigger.IsUpdate)
3     {
4         LeaseTrackingTriggerHandler.method1(trigger.old);
5     }
6 }
```

- Another trigger checks if the tenant's PAN card is valid when entered into the system, showing an error if the card number is invalid. These triggers help automate manual tasks, ensuring timely actions are taken.



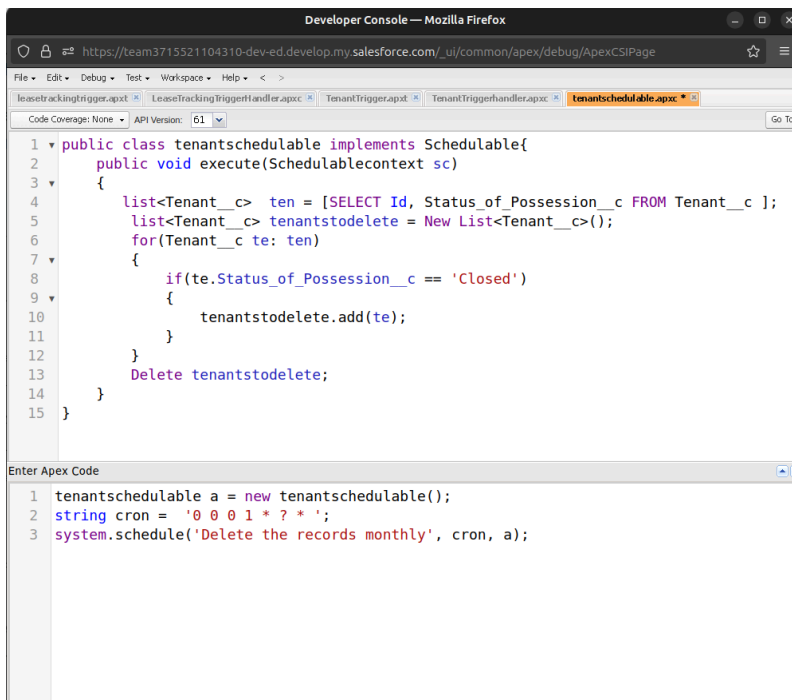
```
1 trigger TenantTrigger on Tenant__c (before insert) {
2     if(Trigger.isBefore)
3     {
4         TenantTriggerHandler.method1(Trigger.New);
5     }
6 }
```



```
1 public class TenantTriggerHandler {
2     public static void method1(List<Tenant__c> te)
3     {
4         for(Tenant__c tenant : te)
5         {
6             if(tenant.PAN_Card__c.length() > 10)
7             {
8                 tenant.addError('This Pan Card number is invalid, Please Enter Valid Pa');
9             }
10        }
11    }
12 }
```

9. Asynchronous Apex

We created an Apex class `tenantschedulable` that implements the `Schedulable` interface to automate the deletion of tenant records whose Status of Possession is marked as "Closed". Created a class named `tenantschedulable` with the `Schedulable` extension. Implemented the logic to query tenants with a closed status and add them to a list for deletion. Scheduled the class to run monthly using a cron expression `(0 0 0 1 * ? *)`, ensuring it executes on the 1st day of each month. The class is scheduled to run monthly, automatically deleting tenant records with a closed status.



```
1 public class tenantschedulable implements Schedulable{
2     public void execute(SchedulableContext sc)
3     {
4         list<Tenant__c> ten = [SELECT Id, Status_of_Possession__c FROM Tenant__c ];
5         list<Tenant__c> tenantstodelete = New List<Tenant__c>();
6         for(Tenant__c te: ten)
7         {
8             if(te.Status_of_Possession__c == 'Closed')
9             {
10                 tenantstodelete.add(te);
11             }
12         }
13         Delete tenantstodelete;
14     }
15 }
```

```
1 tenantschedulable a = new tenantschedulable();
2 string cron = '0 0 0 1 * ? *';
3 system.schedule('Delete the records monthly', cron, a);
```

10. Created Custom Reports

Custom report types were created to structure the data relationships, such as "Tenants with Lease Tracking and Issues." These report types laid the foundation for generating detailed reports to monitor tenant status, lease performance, and tenant-reported issues. Several custom reports were built to monitor important performance metrics, including:

- Tenant Reports: Showing details about current tenants, lease renewals, and payment status.

The screenshot shows the SmartMail Report Builder interface for a 'Tenant Details' report. The left sidebar contains a 'Groups' section with 'GROUP ROWS' (Add group..., Status of possession) and 'GROUP COLUMNS' (Add group..., Date of Reg). The 'Columns' section lists 'Tenant: Tenant Name', 'GST No', and 'PAN Card'. The main area displays a table with columns 'Status of possession', 'Date of Reg', and 'Total'. The 'Status of possession' column has a dropdown menu with 'Pending' selected. The 'Date of Reg' column has a dropdown menu with '10/10/2024' selected. The 'Total' column shows a value of '2'. Below the table, there is a 'Details (2 Rows)' section with a table showing tenant information.

Status of possession	Date of Reg	Total
Pending	10/10/2024	2
Total	Record Count	2

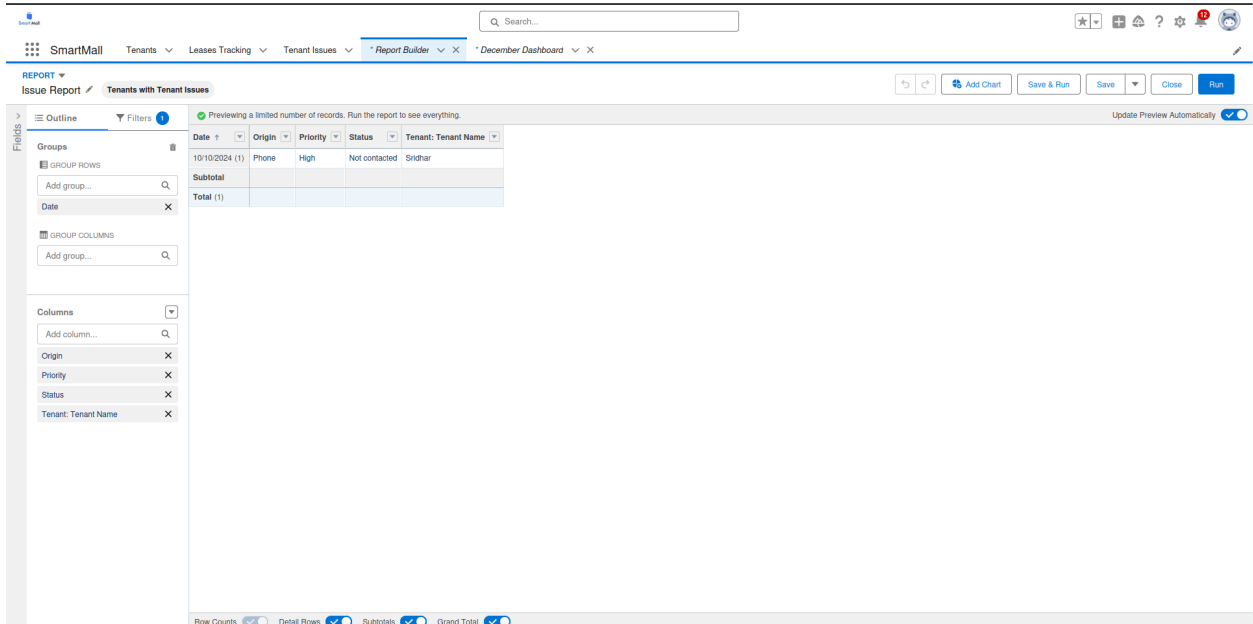
Tenant: Tenant Name	GST No	PAN Card
1 Sridhar	9876543210	ASDFGA
2 drgh	2342345	4545454545
3		

- Lease Tracking Reports: Tracking lease durations, upcoming renewals, and overdue payments.

The screenshot shows the SmartMail Report Builder interface for a 'Lease Tracking' report. The left sidebar contains a 'Groups' section with 'GROUP ROWS' (Add group..., Date of Reg) and 'GROUP COLUMNS' (Add group...). The 'Columns' section lists '# Amount to be paid', 'Date of Possession', and 'Tenant: Tenant Name'. The main area displays a table with columns 'Date of Reg', 'Amount to be paid', 'Date of Possession', and 'Tenant: Tenant Name'. The 'Date of Reg' column has a dropdown menu with '10/10/2024 (1)' selected. The 'Amount to be paid' column shows a value of '9,00,000'. The 'Date of Possession' column has a dropdown menu with '11/12/2024' selected. The 'Tenant: Tenant Name' column shows a value of 'Sridhar'. Below the table, there is a 'Subtotal' row and a 'Total (1)' row.

Date of Reg	Amount to be paid	Date of Possession	Tenant: Tenant Name
10/10/2024 (1)	9,00,000	11/12/2024	Sridhar
Subtotal	9,00,000		
Total (1)	9,00,000		

- Tenant Issue Reports: Displaying the number of issues reported by tenants and their resolution statuses.



The screenshot shows a report builder interface for 'SmartMail'. The report is titled 'Issue Report' and is filtered by 'Tenants with Tenant Issues'. The table displays the following data:

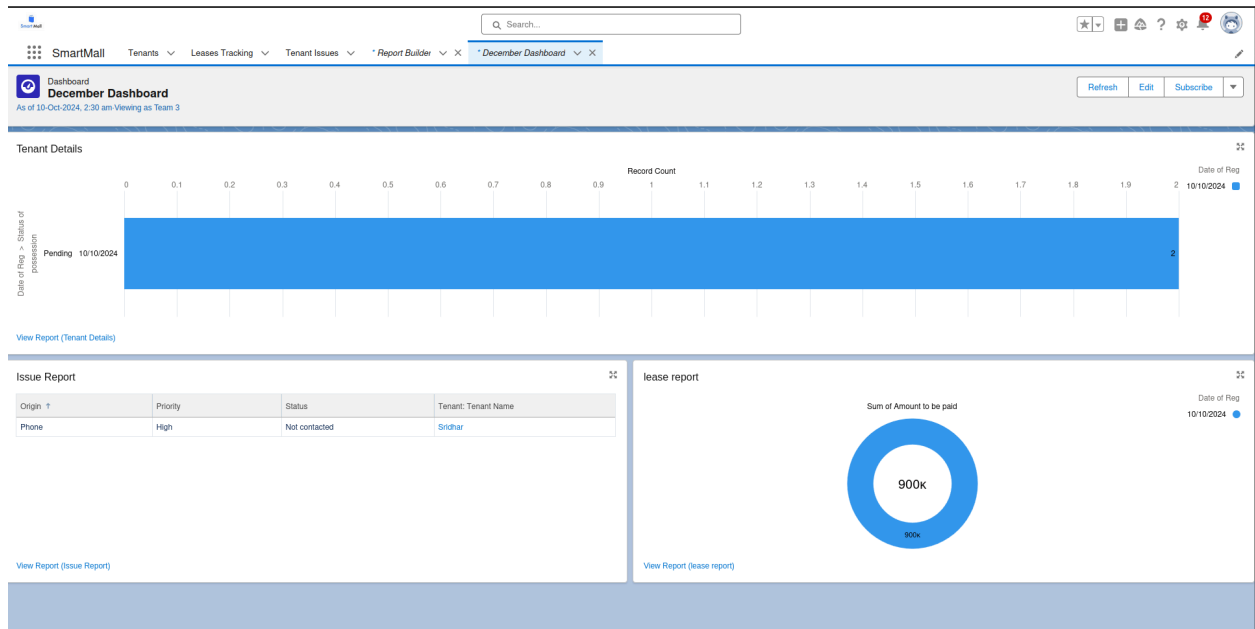
Date	Origin	Priority	Status	Tenant: Tenant Name
10/10/2024 (1)	Phone	High	Not contacted	Sridhar
Subtotal				
Total (1)				

The interface includes a sidebar with 'Groups' and 'Columns' sections. The 'Columns' section lists the following columns: Date, Origin, Priority, Status, and Tenant: Tenant Name. The 'Groups' section lists the following groups: Date, Origin, Priority, Status, and Tenant: Tenant Name. The 'Columns' section also includes a search bar and a dropdown menu. The 'Groups' section also includes a search bar and a dropdown menu. The 'Columns' section also includes a search bar and a dropdown menu. The 'Groups' section also includes a search bar and a dropdown menu.

These reports were configured to run automatically, sending updates to relevant stakeholders.

11. Developed a Dashboard

A centralized Dashboard was designed to provide an overview of key performance metrics such as active leases, issue resolution progress, and tenant satisfaction. The dashboard was designed to offer real-time insights for quick decision-making, enhancing visibility for mall management.



5. Testing and Validation

1. Testing LeaseTrackingTrigger

- **Trigger Logic:** The trigger checks if Amount_to_be_paid__c is greater than half of Total_rent_Yearly__c and sends an email if true.

```
public class LeaseTrackingTriggerHandler {

    public static void method1(List<Lease_Tracking__c> lt1)

    {

        for(Lease_Tracking__c lt2: lt1 )

        {

            if(lt2.Amount_to_be_paid__c > (lt2.Total_rent_Yearly__c)/2)

            {

                Messaging.SingleEmailMessage M = New
                Messaging.SingleEmailMessage();
```

```

        List<String> ToADD = New List<String>{lt2.Email_id__c};

        M.setToAddresses(ToADD);

        M.setSubject('Regarding the Pending Rent');

        M.setPlainTextBody('Hello, This is an Reminder for you to
complete your due rent by the end this month, your due rent thatneeds to
be paid is' +lt2.Amount_to_be_paid__c);

        List<Messaging.Email> AB = New
List<Messaging.Email>{};

        AB.add(M);

        Messaging.sendEmail(AB);

    }

}

}

}

```

- **Steps:**

1. We created a test class LeaseTrackingTriggerTest with a method testAfterUpdateTrigger to test the trigger functionality.
2. We inserted a Lease_Tracking__c record with the required fields, such as Amount_to_be_paid__c, Total_rent_Yearly__c, and Email_id__c.
3. We updated the record by modifying the Amount_to_be_paid__c field to test the "after update" trigger.
4. We queried the Messaging.SingleEmailMessage to check that the email was sent correctly after the record update.
5. We validated the email's subject, recipient, and content to ensure the email notification was accurate.

2. Testing TenantTrigger

- **Trigger Logic:** The trigger checks if the PAN_Card__c field has more than 10 characters and adds an error if true.

```
public class TenantTriggerhandler {  
  
    public static void method1(List<Tenant__c> te)  
  
    {  
  
        for(Tenant__c tenant : te)  
  
        {  
  
            if(tenant.PAN_Card__c.length() > 10)  
  
            {  
  
                tenant.addError('This Pan Card number is invalid, Please Enter  
Valid Pan Card number');  
  
            }  
  
        }  
  
    }  
  
}
```

- **Steps:**
 1. We created a test class TenantTriggerTest with a method testInvalidPANCard to test the validation rule for invalid PAN card numbers.
 2. We inserted a Tenant__c record with an invalid PAN_Card__c value (more than 10 characters).

3. We attempted to insert the record and expected an error to be thrown due to the validation rule on the PAN_Card__c field.
4. We verified that the error message was correctly displayed, indicating that the PAN card number was invalid.

3. Testing tenantschedulable Class

- **Class Logic:** This class deletes Tenant__c records with Status_of_Possession__c set to "Closed".

```
public class tenantschedulable implements Schedulable{

    public void execute(Schedulablecontext sc)

    {

        list<Tenant__c> ten = [SELECT Id, Status_of_Possession__c FROM
Tenant__c ];

        list<Tenant__c> tenantstodelete = New List<Tenant__c>();

        for(Tenant__c te: ten)

        {

            if(te.Status_of_Possession__c == 'Closed')

            {

                tenantstodelete.add(te);

            }

        }

        Delete tenantstodelete;

    }

}
```

- **Steps:**

1. Created a test class TenantSchedulableTest with testScheduledJob method.
2. Inserted two Tenant__c records with different Status_of_Possession__c values.
3. Scheduled and executed the cleanup job.
4. Verified that only the "Closed" tenant was deleted.

6. Conclusion

Summary of Achievements:

The Tenant Management and Lease Tracking System was successfully developed to automate and streamline tenant-related processes while ensuring data accuracy and operational efficiency. Key accomplishments include:

- 1. Efficient Data Management:**

Custom objects, fields, and validation rules were implemented to manage tenant details, lease tracking, and tenant issues, ensuring accurate and organized data across the system.

- 2. Automated Processes:**

Developed Apex triggers and scheduled jobs to automate key tasks such as sending reminders for overdue rent and deleting tenant records with a "Closed" status, reducing manual intervention and improving task efficiency.

- 3. Real-Time Alerts:**

Automated email notifications were set up to alert tenants regarding pending payments, enabling proactive communication and timely follow-ups.

- 4. Enhanced Data Integrity:**

Validation rules were enforced to ensure data accuracy, such as verifying the length of PAN Card numbers and preventing invalid dates for tenant registrations.

- 5. Improved Operational Efficiency:**

A systematic approach to managing tenant records, lease tracking, and

tenant issues was implemented, improving the speed and accuracy of data processing and reporting.

6. Scalable Solution:

The system was designed to scale easily, with the ability to handle a growing number of tenant records and lease agreements while maintaining data integrity and process automation.

The project successfully achieved its objectives by utilizing Salesforce's platform capabilities to automate key tasks, improve data management, and ensure accurate, real-time communication. This initiative has optimized tenant and lease tracking operations, making the process more efficient, reliable, and transparent.