

GARAGE MANAGEMENT SYSTEM

**College Name: SRI KRISHNA ADITHYA COLLEGE OF ARTS
AND SCIENCE**

College Code: bruag

TEAM ID: NM2025TMID22201

TEAM MEMBERS:

Team Leader Name: Prathab Adithya G B

Email: 23bsai245prathabadithyagb@skacas.ac.in

Team Member1: Shajid S

Email: 23bsai254shajids@skacas.ac.in

Team Member2: Ashiq S

Email: 23bsai211ashiqs@skacas.ac.in

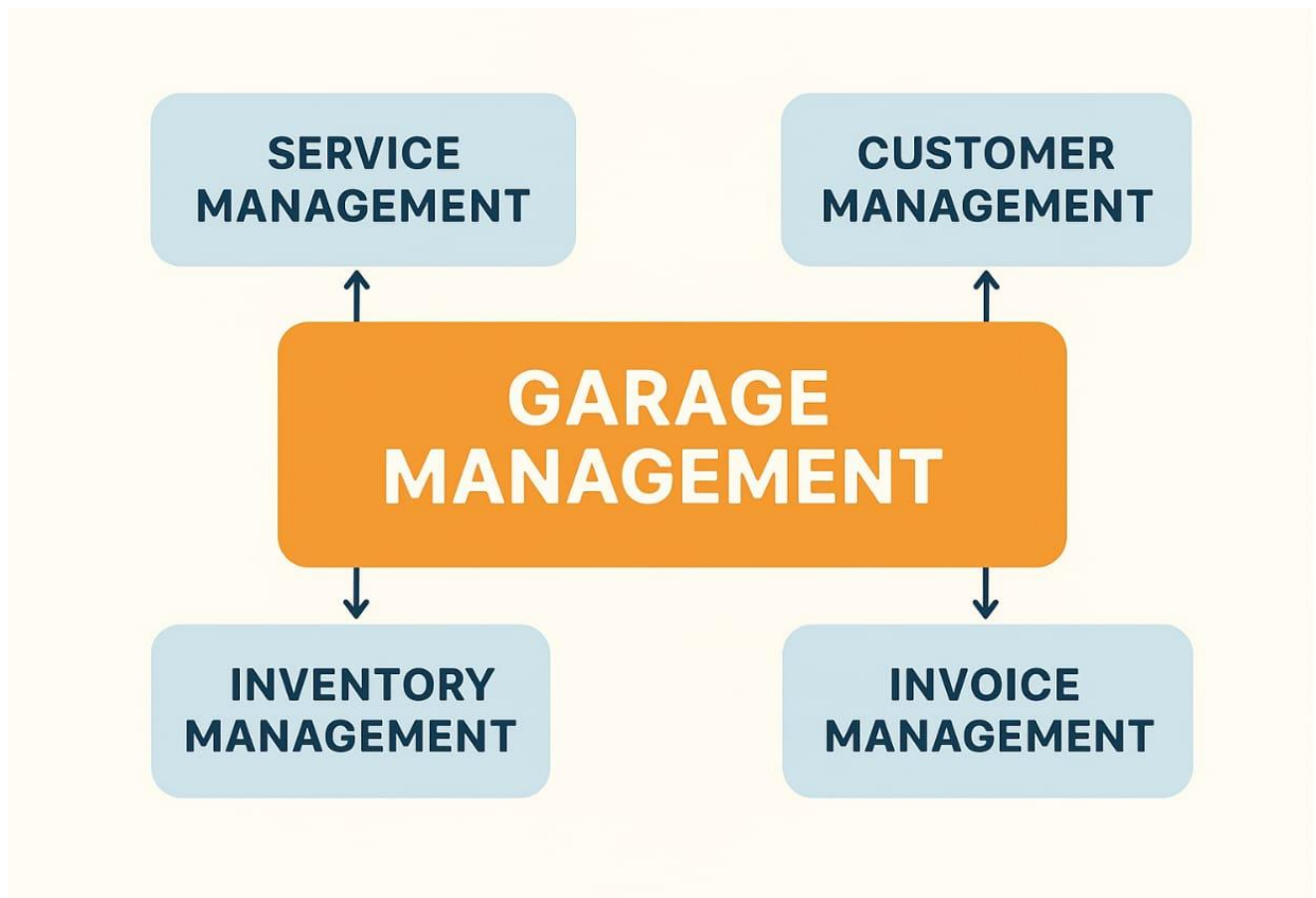
Team Member3: Hari Prasath P

Email: 23bsai217hariprasathp@skacas.ac.in

1.INTRODUCTION

1.1 Project Overview

The Garage Management System is a valuable tool for automotive repair facilities, helping them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. With its user-friendly interface and powerful features, GMS empowers garages to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff.



1.2 Purpose

The main objective of the project is to enable organizations to efficiently manage cars, service schedules and other 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>

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name: Shajid ✓ Last name: S ✓

Job title: Garage Management ✓ Work email: 23bsai254shajids@sk ✓

Company: Salesforce ✓ Country/Region: India ✓

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

☐ I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our Privacy Statement.

Created objects: Customer details, Appointment, Service records, Billing details and feedback.

Setup > OBJECT MANAGER

Customer Detail

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

Details

Description

API Name: Customer_Details__c

Custom: ✓

Singular Label: Customer Detail

Plural Label: Customer Details

Enable Reports: ✓


Track Activities: ✓

Track Field History: ✓








Deployment Status: Deployed

Help Settings: Standard salesforce.com Help Window

Edit Delete



Search Setup



SetupHomeObject Manager

SETUP > OBJECT MANAGER

Appointment

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

Details

Description

API Name

Appointment_c

Custom

✓

Singular Label

Appointment

Plural Label

Appointment

Enable Reports

✓

Track Activities

Track Field History

✓

Deployment Status


Deployed

Help Settings








Standard salesforce.com Help Window

EditDelete

https://orgfarm-741dc9dc8f-dev-ed.develop.lightning.force.com/one/one.app#/setup/Objec...



Search Setup



SetupHomeObject Manager

SETUP > OBJECT MANAGER

Service records

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

Details

Description

API Name

Service_records_c

Custom

✓

Singular Label

Service records

Plural Label

Service records

Enable Reports

✓

Track Activities

Track Field History

✓

Deployment Status

Deployed

Help Settings

Standard salesforce.com Help Window

EditDelete

https://orgfarm-741dc9dc8f-dev-ed.develop.lightning.force.com/one/one.app#/setup/Objec...

Setup Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Billing details and feedback

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

Details

Description

API Name
Billing_details_and_feedback_c

Custom

✓

Singular Label
Billing details and feedback

Plural Label
Billing details and feedback

Enable Reports
✓

Track Activities

Track Field History

✓

Deployment Status
Deployed

Help Settings
Standard salesforce.com Help Window

Edit Delete

<https://orgfarm-741dc9dd8f-dev-ed.develop.lightning.force.com/one.app#/setup/Objec>

- Configured fields and relationships

Setup Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER


Billing details and feedback

Fields & Relationships








8 Items, Sorted by Field Label

Quick Find New Deleted Fields Field Dependencies Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid__c	Currency(10, 2)		
Payment Status	Payment_Status__c	Picklist		
Rating for service	Rating_for_service__c	Text(1)		
Service records	Service_records__c	Lookup(Service records)		✓



Search Setup



SetupHomeObject Manager

SETUP > OBJECT MANAGER

Service records

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

Fields & Relationships


8 Items, Sorted by Field Label

Quick Find








NewDeleted FieldsField DependenciesSet History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Appointment	Appointment__c	Lookup(Appointment)		✓	
Created By	CreatedById	Lookup(User)			
Last Modified By	LastModifiedById	Lookup(User)			
Owner	OwnerId	Lookup(User,Group)		✓	
Quality Check Status	Quality_Check_Status__c	Checkbox			
service date	service_date__c	Formula (Date)			
Service records Name	Name	Auto Number		✓	
Service Status	Service_Status__c	Picklist			

orgfarm-741dc9dc8f-dev-ed.develop.lightning.force.com/lightning/setup/_view



Search Setup



SetupHomeObject Manager

SETUP > OBJECT MANAGER

Customer Detail

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

Fields & Relationships

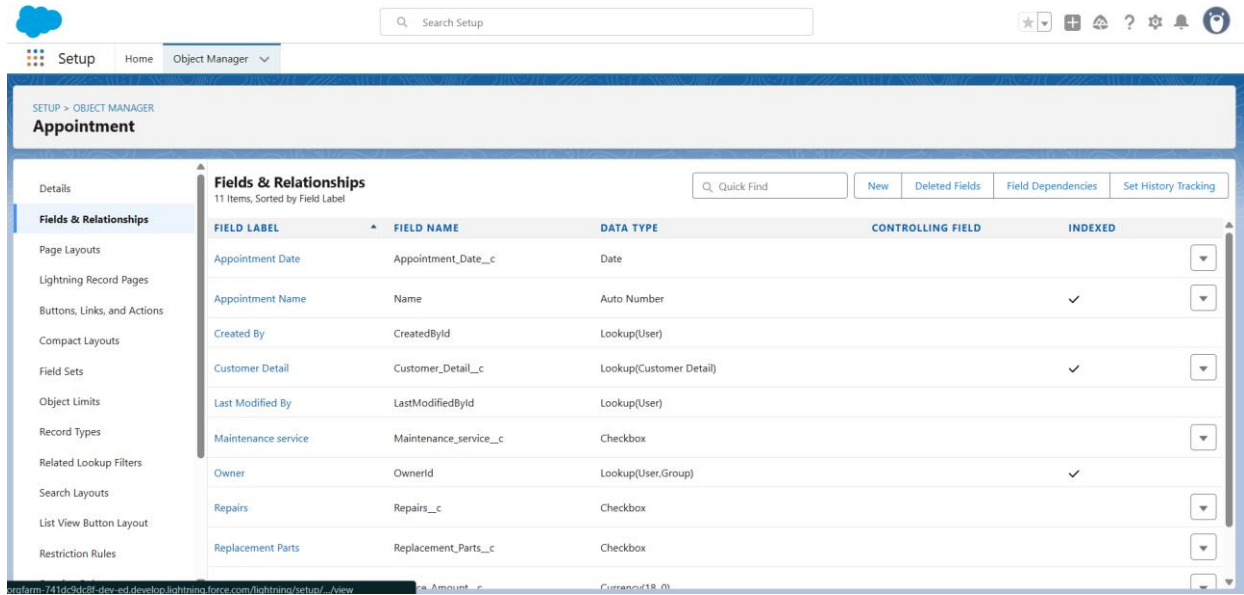
6 Items, Sorted by Field Label

Quick Find

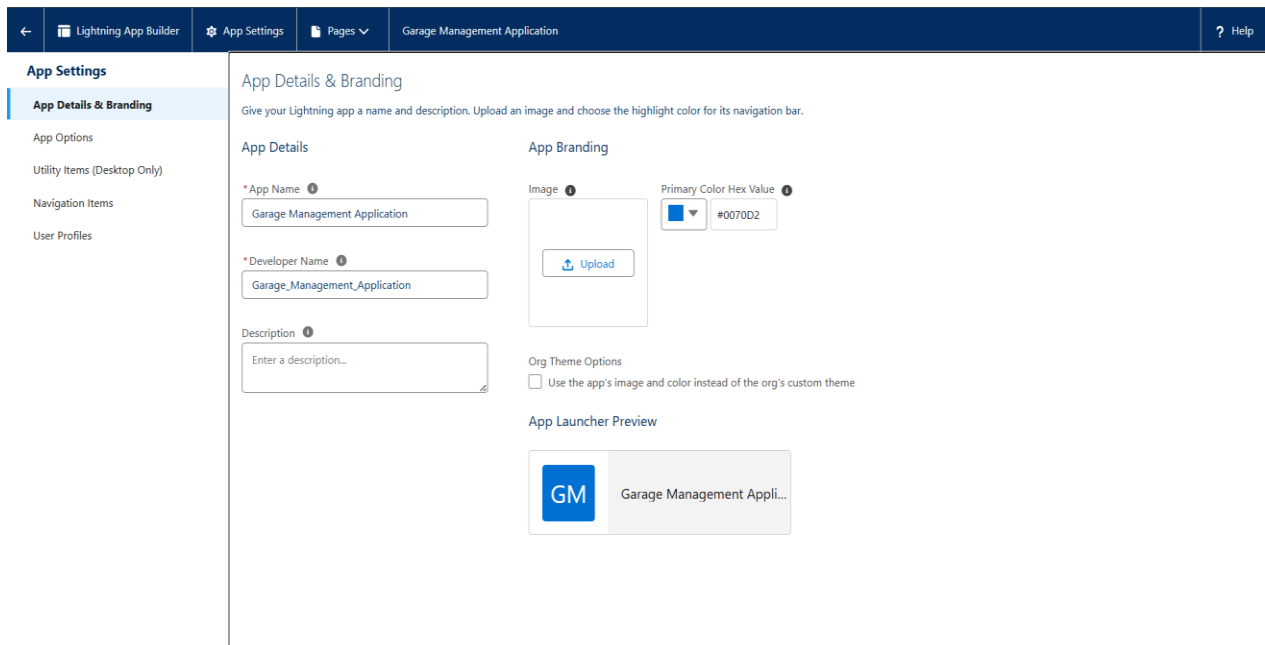
NewDeleted FieldsField DependenciesSet History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED	
Created By	CreatedById	Lookup(User)			
Customer Details Name	Name	Text(80)		✓	
Gmail	Gmail__c	Email			
Last Modified By	LastModifiedById	Lookup(User)			
Owner	OwnerId	Lookup(User,Group)		✓	
Phone number	Phone_number__c	Phone			

https://orgfarm-741dc9dc8f-dev-ed.develop.lightning.force.com/one/one-app#/setup/Obec



- Developed Lightning App with relevant tabs



App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

🔍 Type to filter list...

Accounts

Activation Targets

Activations

All Sites

Alternative Payment Methods

Analytics

App Launcher

Appointment Categories

Appointment Invitations

Approval Requests

...

▶

◀

Selected Items

🔔 Customer Detail

📅 Appointment

📄 Service records

📄 Billing details and feedback

📄 Reports

📄 Dashboards

▶

◀

▲

▼

←

Lightning App Builder

⚙️ App Settings

📄 Pages ▾

Garage Management Application

?

Help

App Settings

App Details & Branding

App Options

Utility Items (Desktop Only)

Navigation Items

User Profiles

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

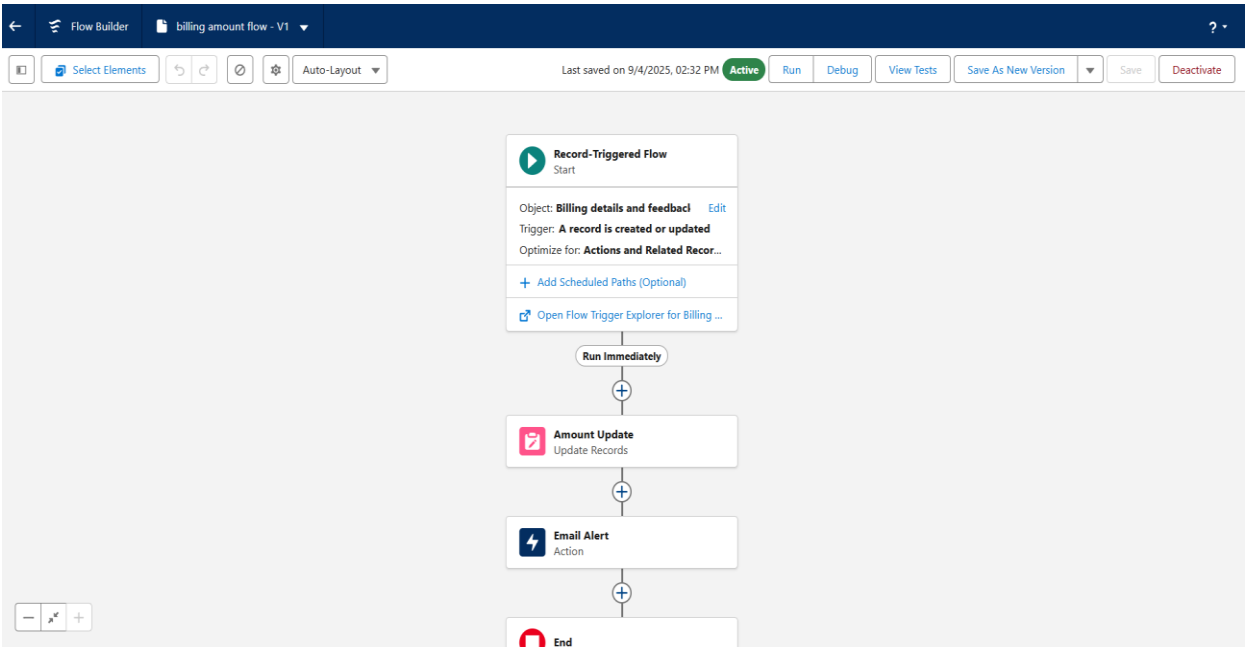
▶

◀

Selected Profiles

System Administrator

- Implemented Flows for billing details and feedback, Service records.



- To create a validation rule for billing and feedback, appointment objects.

The screenshot shows the Salesforce Setup page for the 'Appointment' object. The 'Validation Rules' section is expanded, showing a table with one rule. The table has columns for 'RULE NAME', 'ERROR LOCATION', 'ERROR MESSAGE', 'ACTIVE', and 'MODIFIED BY'. The rule 'Vehicle' has an error location of 'Vehicle number plate' and an error message of 'Please enter valid number'. The rule is active and was modified by 'Shajid S' on 9/3/2025 at 11:50 PM.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter valid number	✓	Shajid S, 9/3/2025, 11:50 PM

Setup Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER

Appointment

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

Appointment Validation Rule

Define a validation rule by specifying an error condition and a corresponding error message. The error condition is written as a Boolean formula expression that returns true or false. When the formula expression returns true, the save will be aborted and the error message will be displayed. The user can correct the error and try again.

Validation Rule Edit Save Save & New Cancel

Rule Name

Active ☒

Description

Error Condition Formula * = Required Information

Example: `Discount_Percent__c > 0.30` [More Examples...](#)
 Display an error if Discount is more than 30%
 If this formula expression is **true**, display the text defined in the Error Message area

`NOT (REGEX (Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))`

Functions
 -- All Function Categories --
 ABS
 ACOS
 ADDMONTHS
 AND
 ASCII
 ASIN

 ABS(number)
 Returns the absolute value of a number, a number without its sign

Quick Tips
 • Operators & Functions

- Added Apex trigger Amount Distribution Handler.

Garage Manage... Customer D...

Appointment

Recently Viewed

1 item • Updated a few seconds ago

☐ Appointment Name

1 ☐ app-014

Edit app-014

* = Required Information

Appointment Name app-014 Owner Shajid S

Appointment Date
 Format: 12/31/2024

Customer Detail

Maintenance service ☒

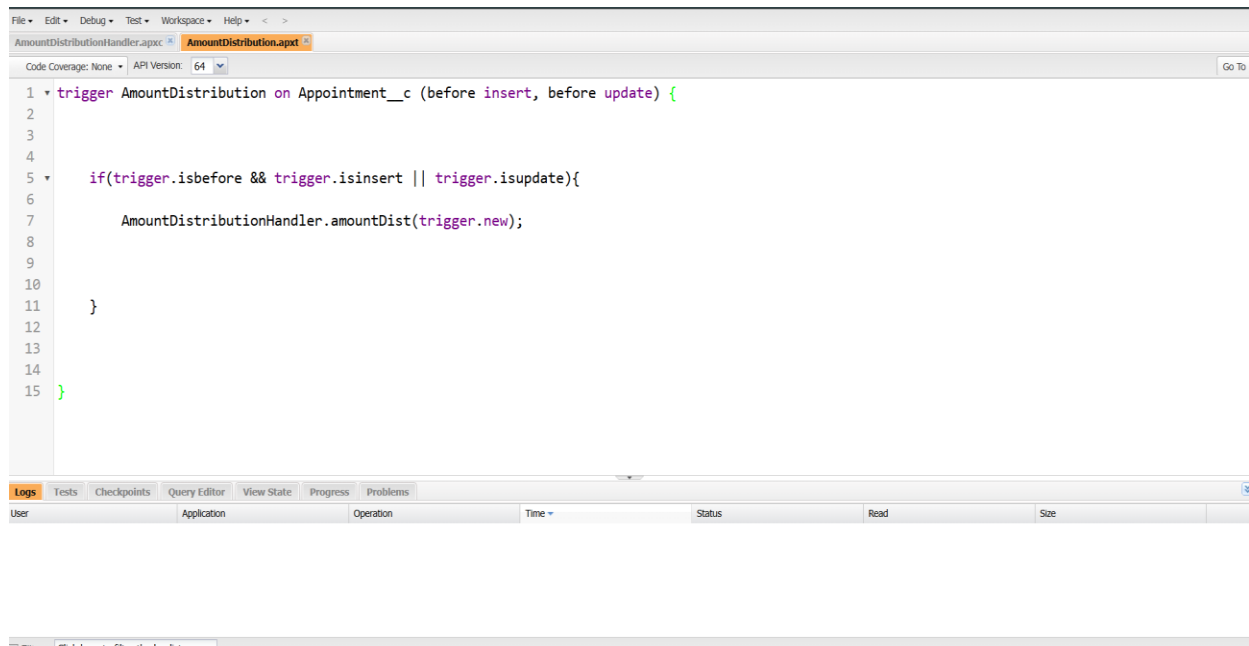
Repairs ☒

Replacement Parts ☒

* Vehicle number plate

Service Amount

- Scheduled notification reminder emails using Apex class.

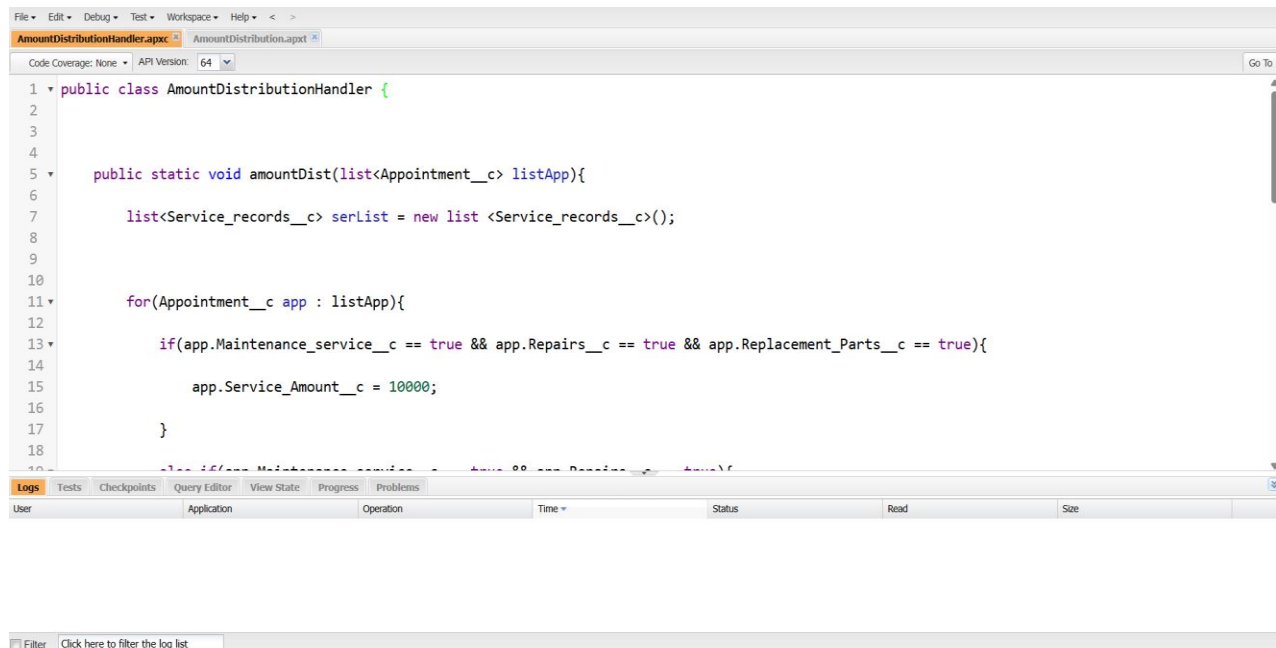


The screenshot shows an IDE window with the file 'AmountDistribution.apex' open. The code is an Apex trigger for the 'Appointment__c' object, triggered 'before insert, before update'. The trigger calls a static method 'amountDist' on the 'AmountDistributionHandler' class. The IDE interface includes a menu bar (File, Edit, Debug, Test, Workspace, Help), a toolbar with 'Code Coverage: None' and 'API Version: 64', and a bottom panel with tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. A log table is visible at the bottom with columns: User, Application, Operation, Time, Status, Read, and Size.

```
1 trigger AmountDistribution on Appointment__c (before insert, before update) {  
2  
3  
4  
5     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
6  
7         AmountDistributionHandler.amountDist(trigger.new);  
8  
9  
10    }  
11  
12  
13  
14  
15 }
```

- Apex Trigger

Create an Apex Trigger



The screenshot shows an IDE window with the file 'AmountDistributionHandler.apex' open. The code is an Apex class named 'AmountDistributionHandler' with a static method 'amountDist'. This method takes a list of 'Appointment__c' objects and iterates through them. For each appointment, it checks if 'Maintenance_service__c' is true, 'Repairs__c' is true, and 'Replacement_Parts__c' is true. If all conditions are met, it sets 'Service_Amount__c' to 10000. The IDE interface is similar to the previous screenshot, showing the same menu bar, toolbar, and bottom panel with the log table.

```
1 public class AmountDistributionHandler {  
2  
3  
4  
5     public static void amountDist(list<Appointment__c> listApp){  
6  
7         list<Service_records__c> serlist = new list <Service_records__c>();  
8  
9  
10  
11     for(Appointment__c app : listApp){  
12  
13         if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){  
14  
15             app.Service_Amount__c = 10000;  
16  
17         }  
18  
19     }  
20 }
```

```
1 trigger AmountDistribution on Appointment__c (before insert, before update) {
2
3
4
5     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
6
7         AmountDistributionHandler.amountDist(trigger.new);
8
9
10    }
11
12
13
14
15 }
```

User	Application	Operation	Time	Status	Read	Size
------	-------------	-----------	------	--------	------	------

Create an Apex Handler class

```
1 public class AmountDistributionHandler {
2
3
4
5     public static void amountDist(list<Appointment__c> listApp){
6
7         list<Service_records__c> serList = new list <Service_records__c>();
8
9
10
11        for(Appointment__c app : listApp){
12
13            if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
14
15                app.Service_Amount__c = 10000;
16
17            }
18
19        }
20    }
21 }
```

User	Application	Operation	Time	Status	Read	Size
------	-------------	-----------	------	--------	------	------

Schedule Apex class

Setup

Home

Object Manager

apex class

Custom Code

Apex Classes

Didn't find what you're looking for?

Try using Global Search.

Apex Classes

Help for this Page

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.02%

You are currently using 1,407 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage

Compile all classes

View: All Create New View

Developer Console

New

Generate from WSDL

Run All Tests

Schedule Apex

Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del Security	AmountDistributionHandler		64.0	Active	1,196	Shajid S. 9/4/2025, 2:13 AM	<input type="checkbox"/>

Dynamic Apex Classes

Dynamic Apex extends your programming reach by interacting with Lightning Platform components.

View: All Create New View

Class Name	Namespace Prefix	Api Version	Created By	Last Modified By
No records to display.				

Garage Manage...

Customer Detail

Appointment

Service records

Billing details and feedback

Reports

Dashboards

Customer Detail

shajids

New Contact

Edit

New Opportunity

Related

Details

Customer Details Name

shajids

Owner

Shajid S.

Phone number

(123) 456-7890

Gmail

gdfndfndf@gmail.com

Created By

Shajid S. 9/8/2025, 2:01 AM

Last Modified By

Shajid S. 9/8/2025, 2:01 AM

Activity

Filters: All time • All activities • All types

Refresh • Expand All • View All

Upcoming & Overdue

No activities to show.

Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.



Appointment
app-014

New Contact Edit New Opportunity

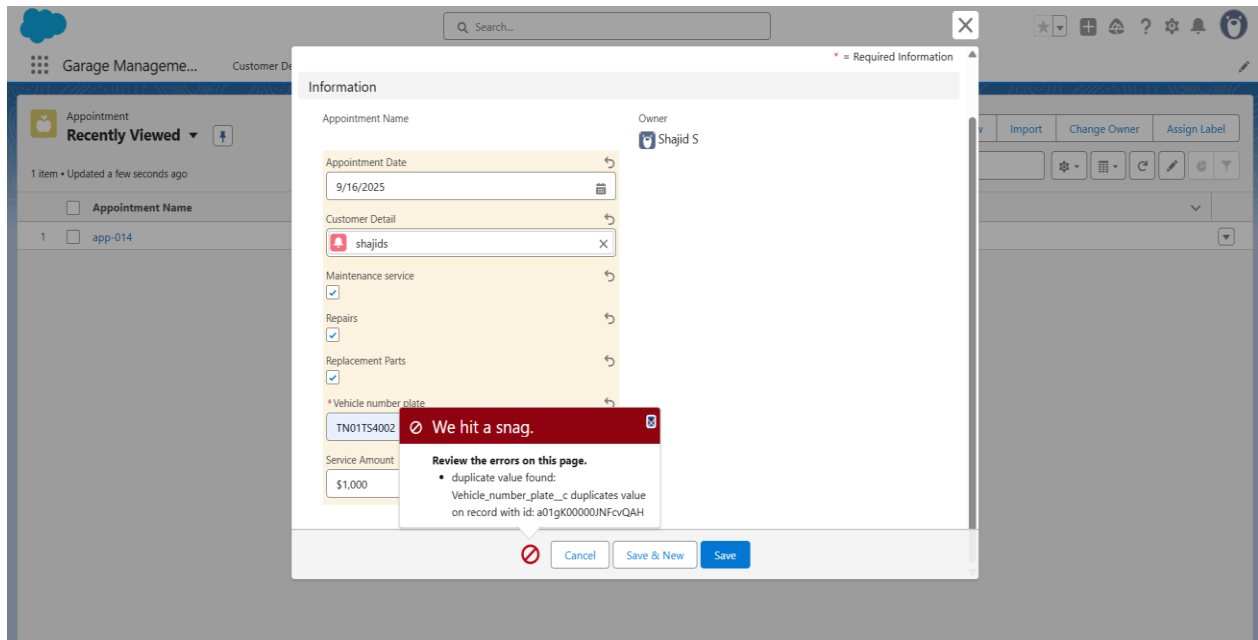
Related Details

Appointment Name	app-014	Owner	Shajid S
Appointment Date	9/24/2025		
Customer Detail	shajids		
Maintenance service	<input checked="" type="checkbox"/>		
Repairs	<input checked="" type="checkbox"/>		
Replacement Parts	<input checked="" type="checkbox"/>		
Vehicle number plate	TN01TS4002		
Service Amount	\$10,000		
Created By	Shajid S, 9/8/2025, 2:04 AM	Last Modified By	Shajid S, 9/8/2025, 2:04 AM

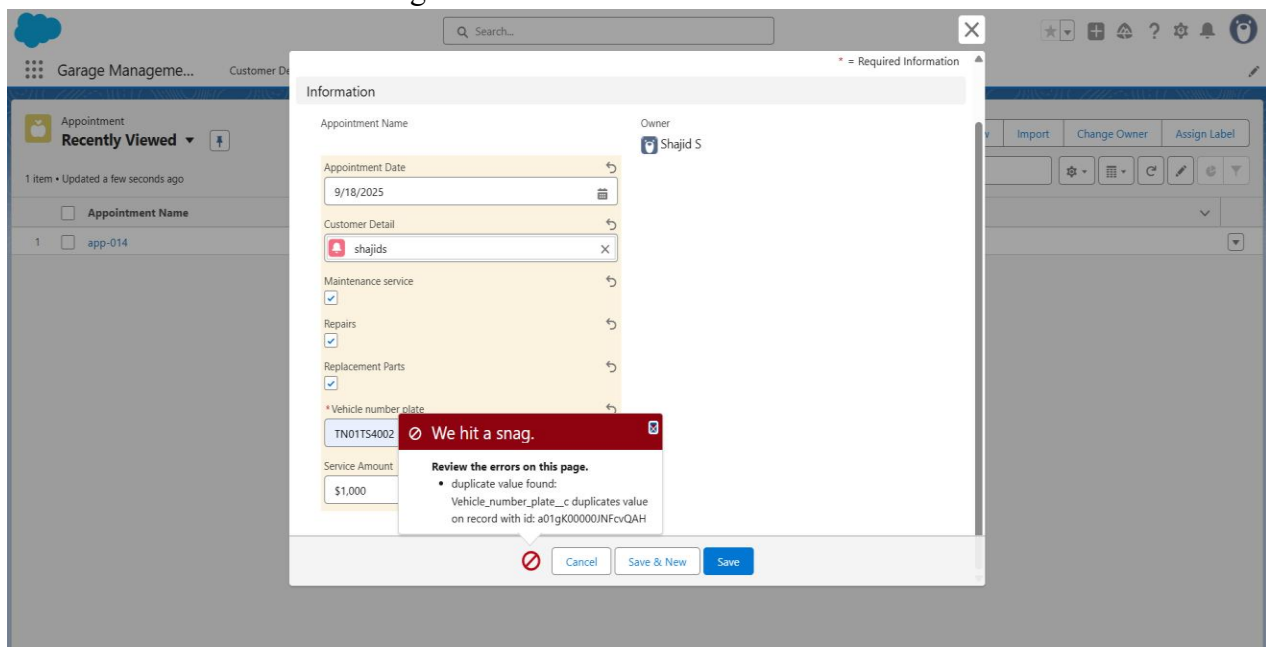
FUNCTIONAL AND PERFORMANCE TESTING

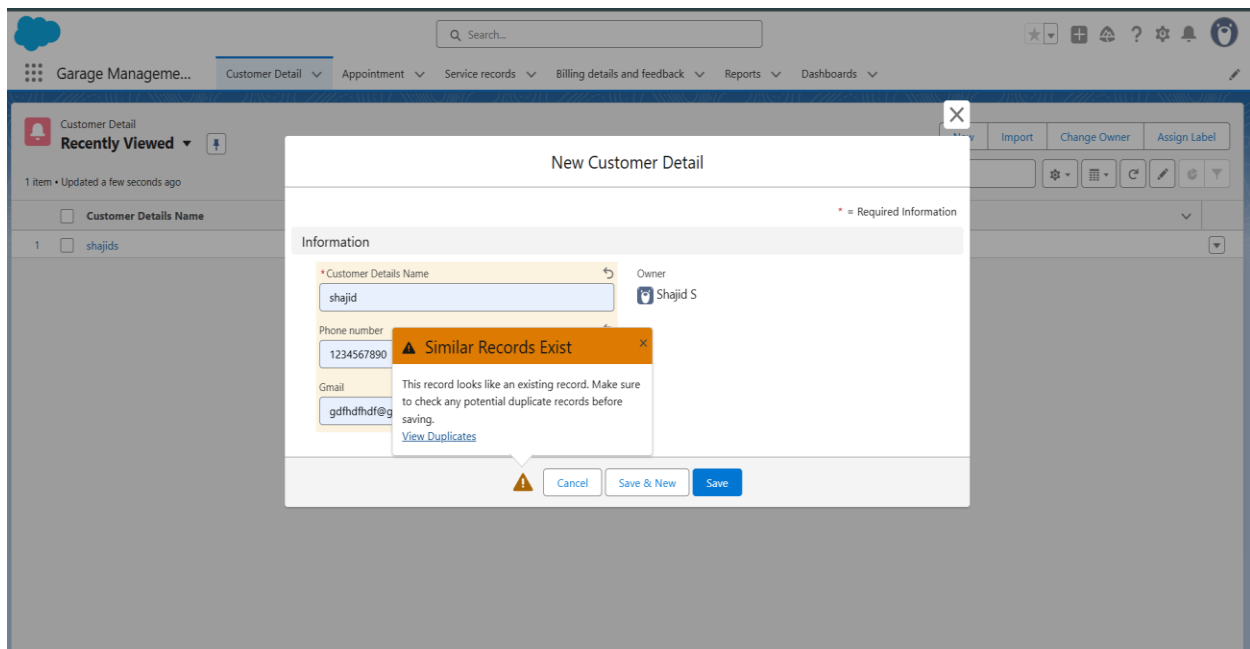
Performance Testing

- Trigger validation by entering duplicate vehicle number-property records

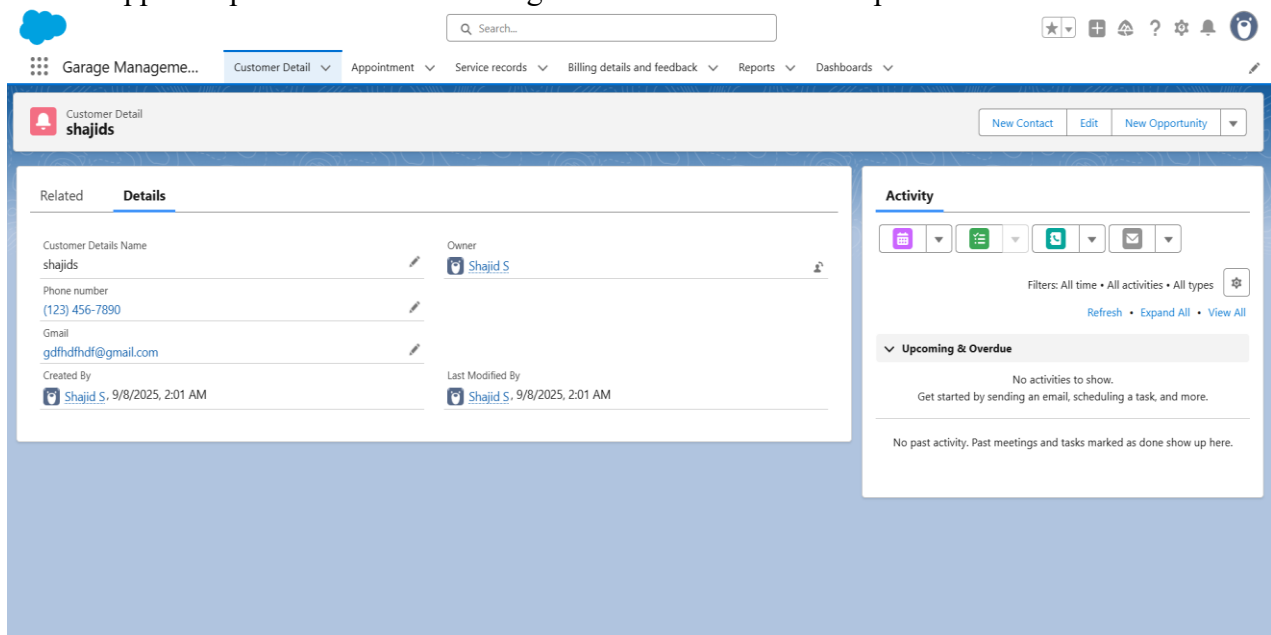


- Validation Rule checking





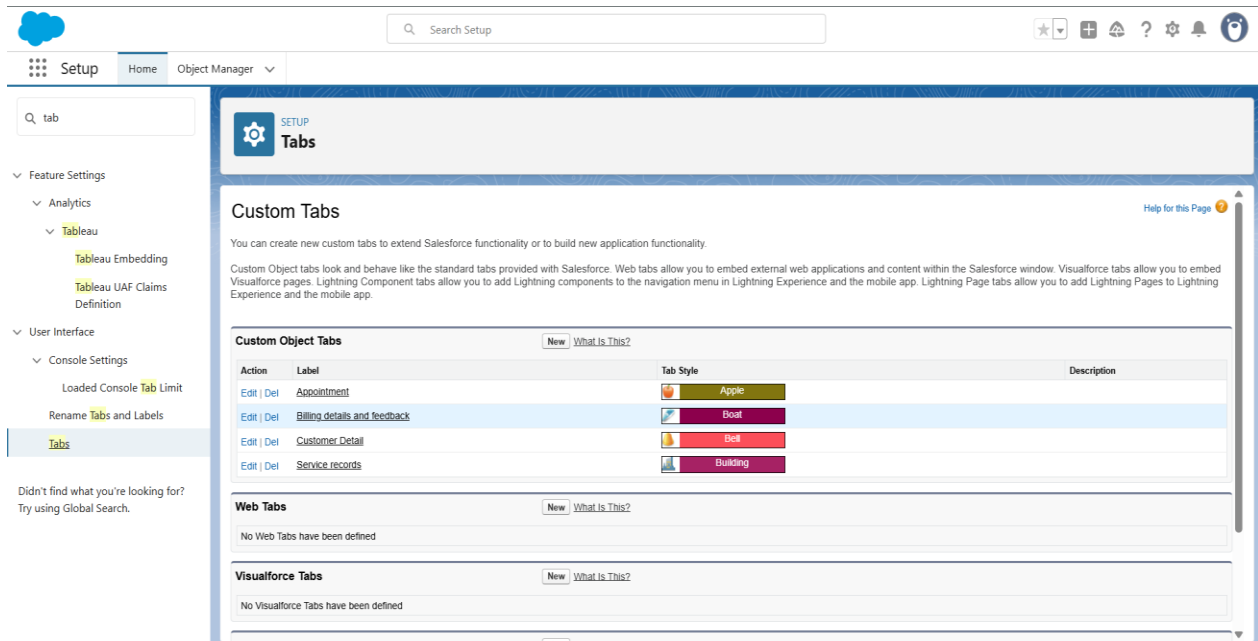
- Approval process validated through email alerts and status updates



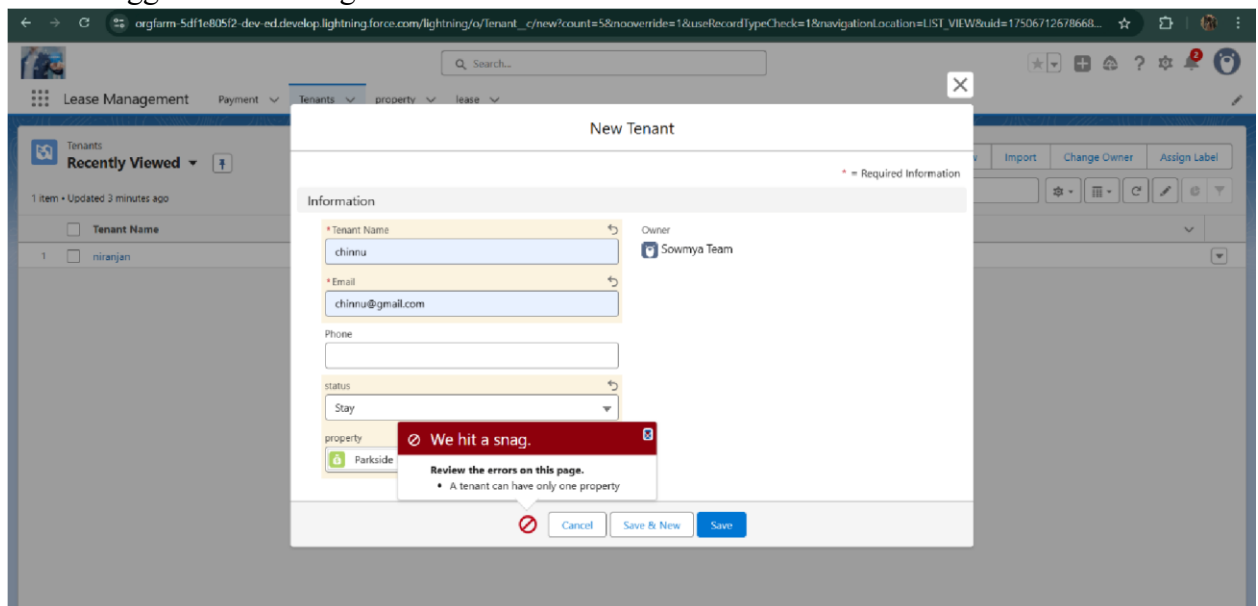
RESULTS

Output Screenshots

- Tabs for Customer details, Appointment, Service records, Billing details and feedback.



- Trigger error messages



ADVANTAGES & DISADVANTAGES

Advantages

- Centralized data storage for customer, vehicle, and service details.
- Better customer service with service reminders and history tracking.
- Automation of job cards, invoices, and service records.
- Integration with Salesforce CRM for a 360° customer view.
- Effective spare parts and inventory management.
- Detailed reports and analytics for business decisions.
- Scalable and customizable for any size of garage.

Disadvantages

- Complex setup and requires proper Salesforce configuration.
- High cost of Salesforce licensing and maintenance.
- Needs stable internet connection for smooth operation.
- Learning curve for staff with limited technical knowledge.
- Customization effort required to fit unique workflows.
- Data security and privacy concerns in cloud storage.

CONCLUSION

The Garage Management System successfully streamlines the operations of leasing through a structured, automated Salesforce application. It improves efficiency, communication, and data accuracy for both admins and customers.

APPENDIX

• Source Code:

```
public class AmountDistributionHandler {

    public static void amountDist(list<Appointment__c> listApp){
        list<Service_records__c> serList = new list <Service_records__c>();

        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true &&
app.Replacement_Parts__c == true){
                app.Service_Amount__c = 10000;
            }
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
                app.Service_Amount__c = 5000;
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
            }
            else if(app.Repairs__c == true){
                app.Service_Amount__c = 3000;
            }
            else if(app.Replacement_Parts__c == true){
                app.Service_Amount__c = 5000;
            }

        }
    }
}
```

The syntax for creating trigger is :

```
trigger AmountDistribution on Appointment__c (before insert, before update) {
```

```
if(trigger.isbefore && trigger.isinsert || trigger.isupdate){  
    AmountDistributionHandler.amountDist(trigger.new);  
  
}  
  
}
```