

# GARAGE MANAGEMENT SYSTEM

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

**College Code:**

**TEAM ID:**

**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

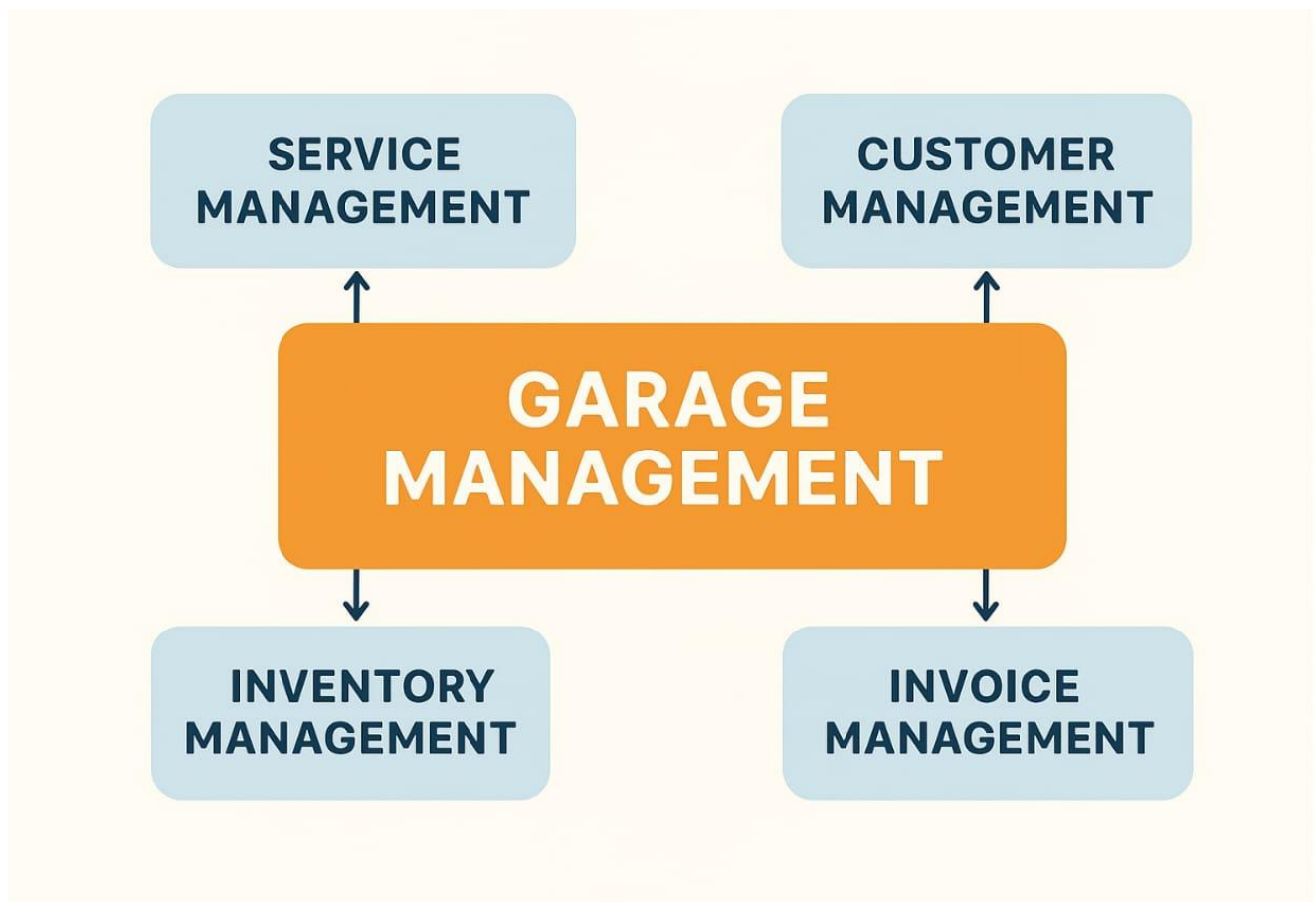
**Team Member3:** Hari Prasath P

**Email:** [23bsai217hariprasathp@skacas.ac.in](mailto: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.



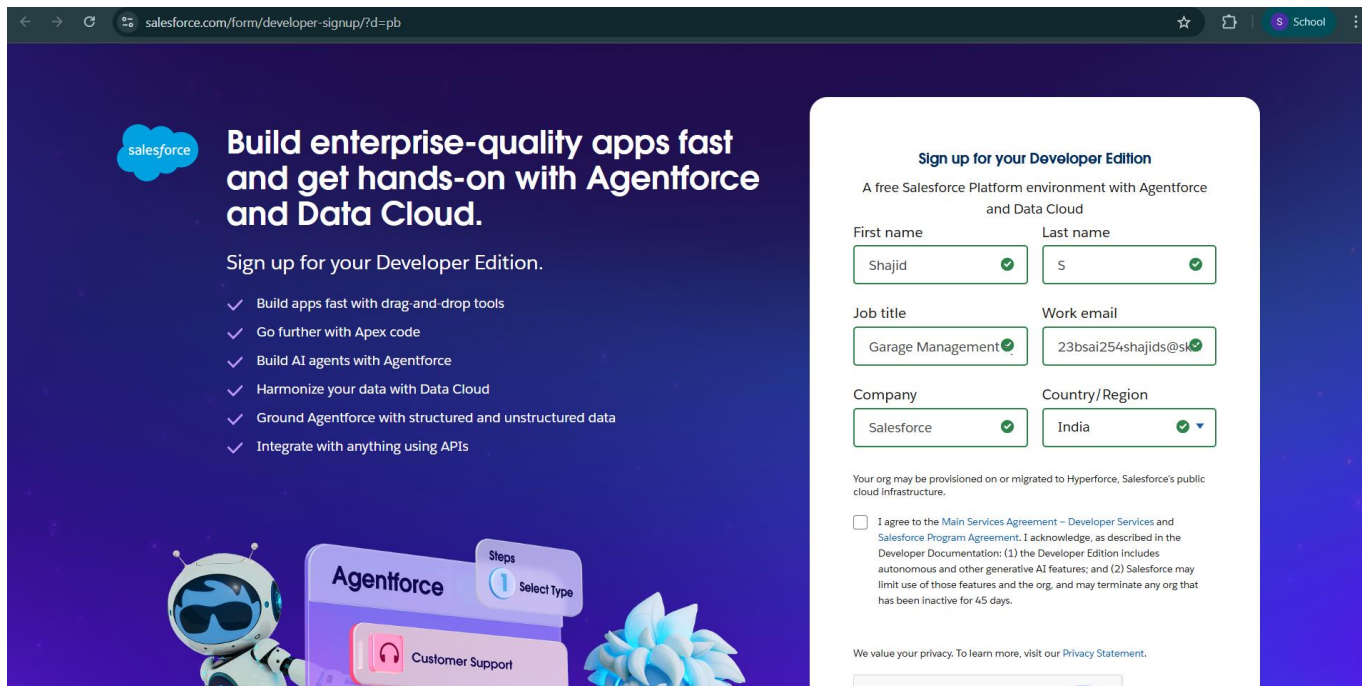
## 1.2 Purpose

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>




The screenshot shows the Salesforce Developer Edition sign-up page. On the left, there's a promotional banner with the Salesforce logo and text: "Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud." Below this, it says "Sign up for your Developer Edition." and lists five benefits: "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", and "Integrate with anything using APIs". At the bottom of the banner is an illustration of an Agentforce robot and a "Customer Support" button.

On the right, the sign-up form is titled "Sign up for your Developer Edition" and describes it as "A free Salesforce Platform environment with Agentforce and Data Cloud". The form fields are as follows:







Field	Value
First name	Shajid
Last name	S
Job title	Garage Management
Work email	23bsai254shajids@sk
Company	Salesforce
Country/Region	India

Below the form, there is a checkbox for "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." and a link to the "Privacy Statement".

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



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

Details

Description

API Name

Customer\_Details\_\_c

Custom

✓

Singular Label

Customer Detail

Plural Label

Customer Detail

Enable Reports

✓

Track Activities

✓

Track Field History

✓


Deployment Status

Deployed







Help Settings

Standard salesforce.com Help Window

EditDelete



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

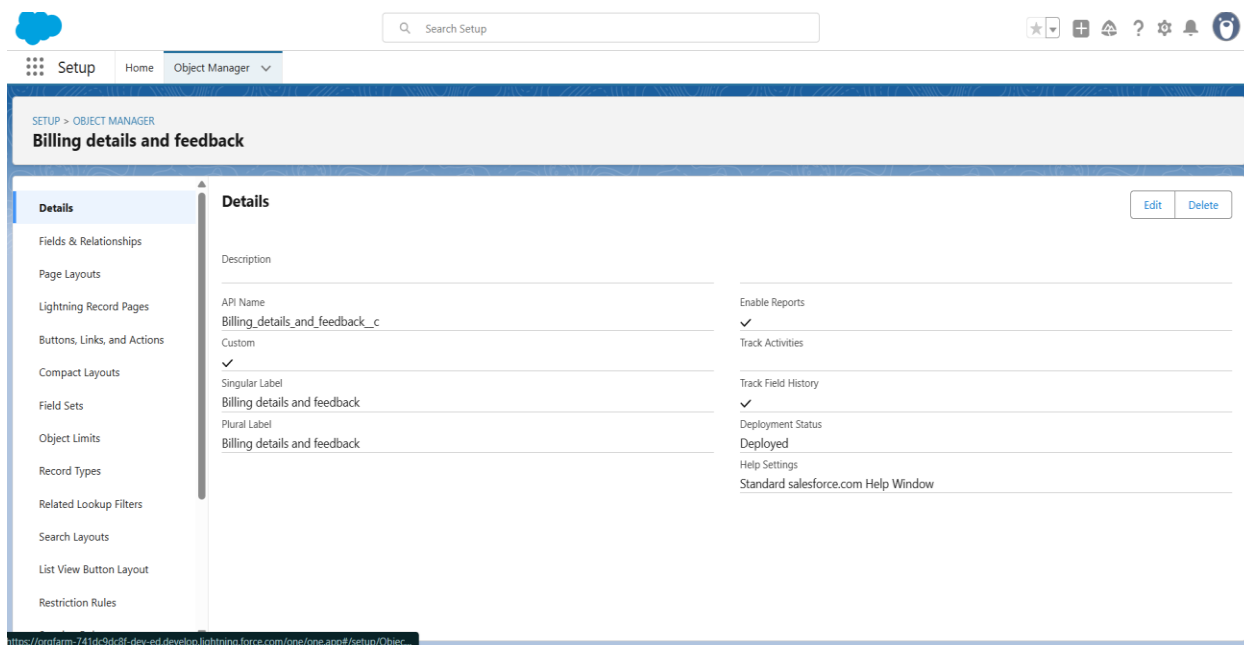
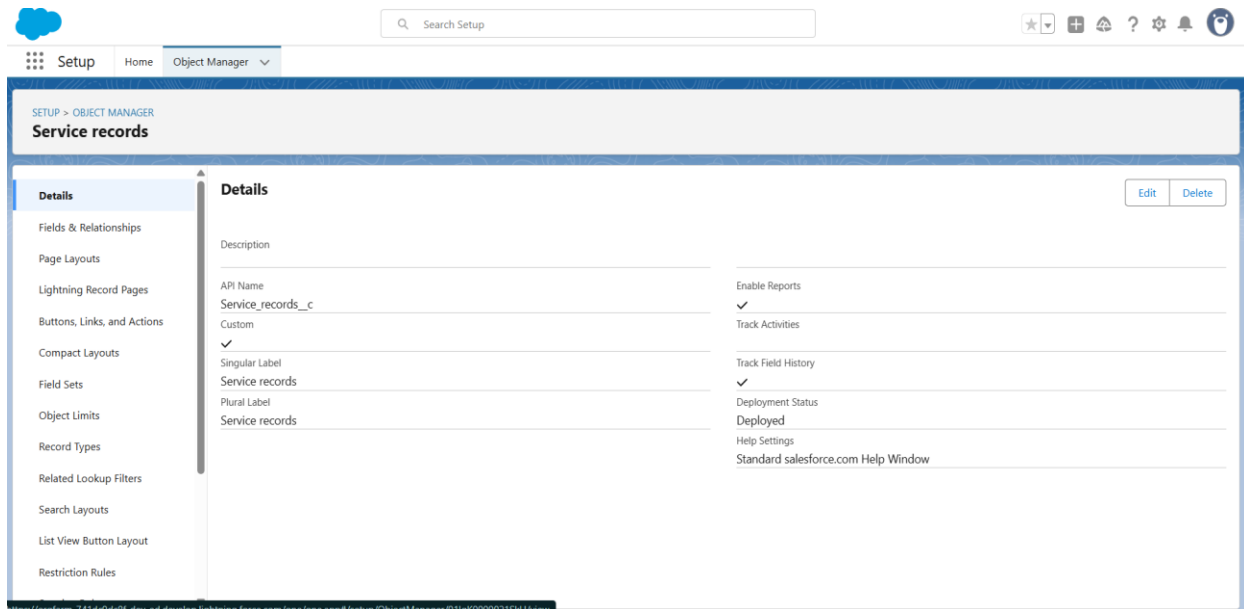
Deployed

Help Settings

Standard salesforce.com Help Window

EditDelete

<https://crafram-741dc3dc3f-dev-ed.develop.lightning.force.com/one/one.app#/setup/ObjectManager/01t0000000041820/Details/Details>



Setup

Home

Object Manager

Search Setup

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

Scoping Rules

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)		✓

Setup

Home

Object Manager

Search Setup

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

New

Deleted Fields

Field Dependencies

Set 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		

Setup

Home

Object Manager

Search Setup

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

Q Quick Find


New

Deleted Fields








Field Dependencies

Set 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		



[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](#)

### Fields & Relationships

11 Items, Sorted by Field Label

[New](#)
[Deleted Fields](#)
[Field Dependencies](#)
[Set History Tracking](#)

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date__c	Date		
Appointment Name	Name	Auto Number		
Created By	CreatedById	Lookup(User)		
Customer Detail	Customer_Detail__c	Lookup(Customer Detail)		
Last Modified By	LastModifiedById	Lookup(User)		
Maintenance service	Maintenance_service__c	Checkbox		
Owner	OwnerId	Lookup(User,Group)		
Repairs	Repairs__c	Checkbox		
Replacement Parts	Replacement_Parts__c	Checkbox		

- Developed Lightning App with relevant tabs

← 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

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 Branding

\* App Name ⓘ

Garage Management Application

\* Developer Name ⓘ

Garage\_Management\_Application

Description ⓘ

Enter a description...

Image ⓘ

Upload

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

GM

Garage Management Appli...

← 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

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

Selected Items

Q. Type to filter list...

Accounts

Activation Targets

Activations

All Sites

Alternative Payment Methods

Analytics

App Launcher

Appointment Categories

Appointment Invitations

Approval Requests

...

▶

◀

Customer Detail

Appointment

Service records

Billing details and feedback

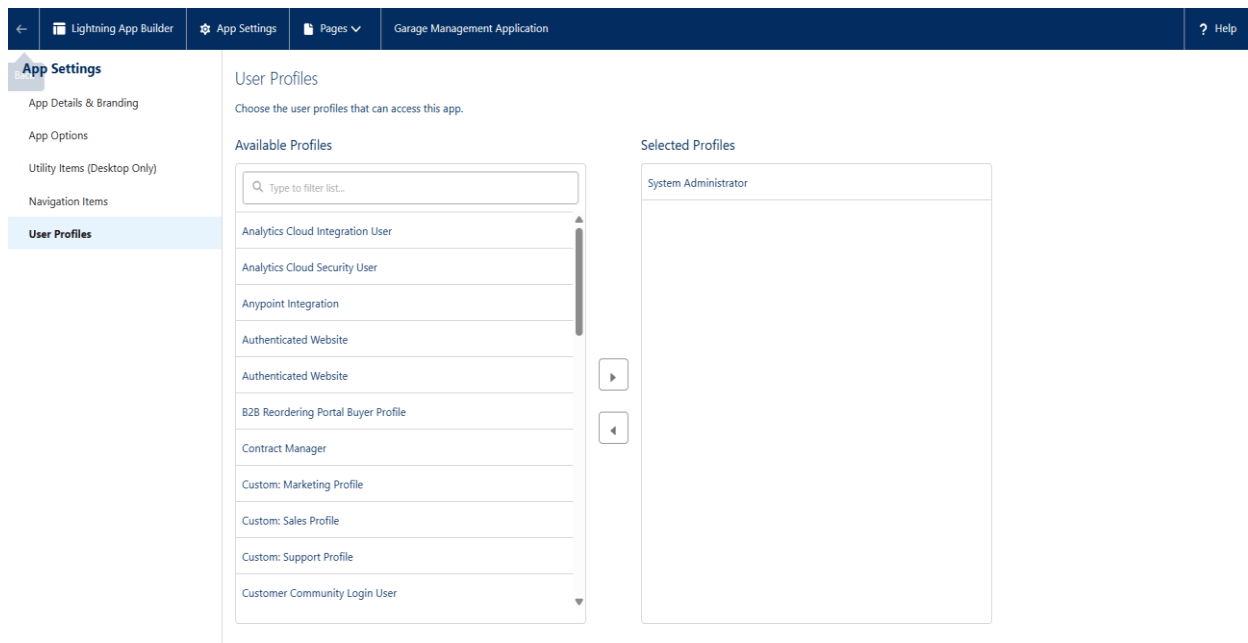
Reports

Dashboards

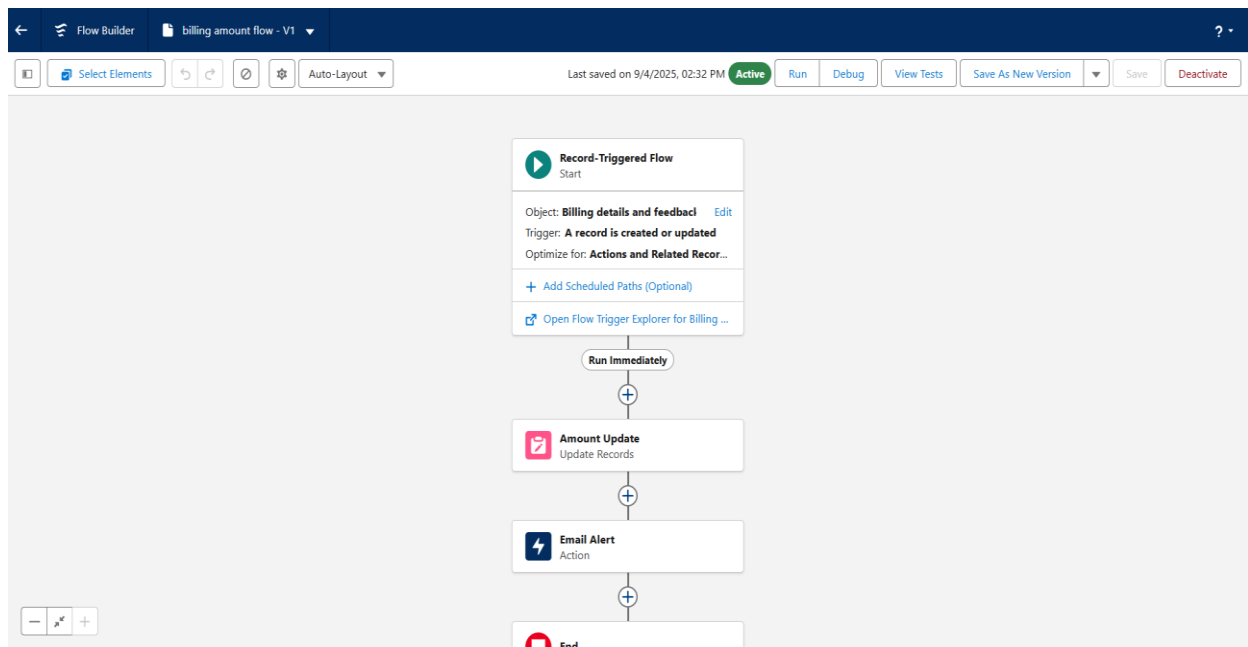
▲

▼





- 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 interface for the 'Appointment' object. The 'Validation Rules' section is active, displaying a table with one rule: 'Vehicle' with an error location of 'Vehicle number plate' and an error message of 'Please enter valid number'.

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

Insert Field Insert Operator

`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

Insert Selected Function

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 shajids

Maintenance service ☒

Repairs ☒

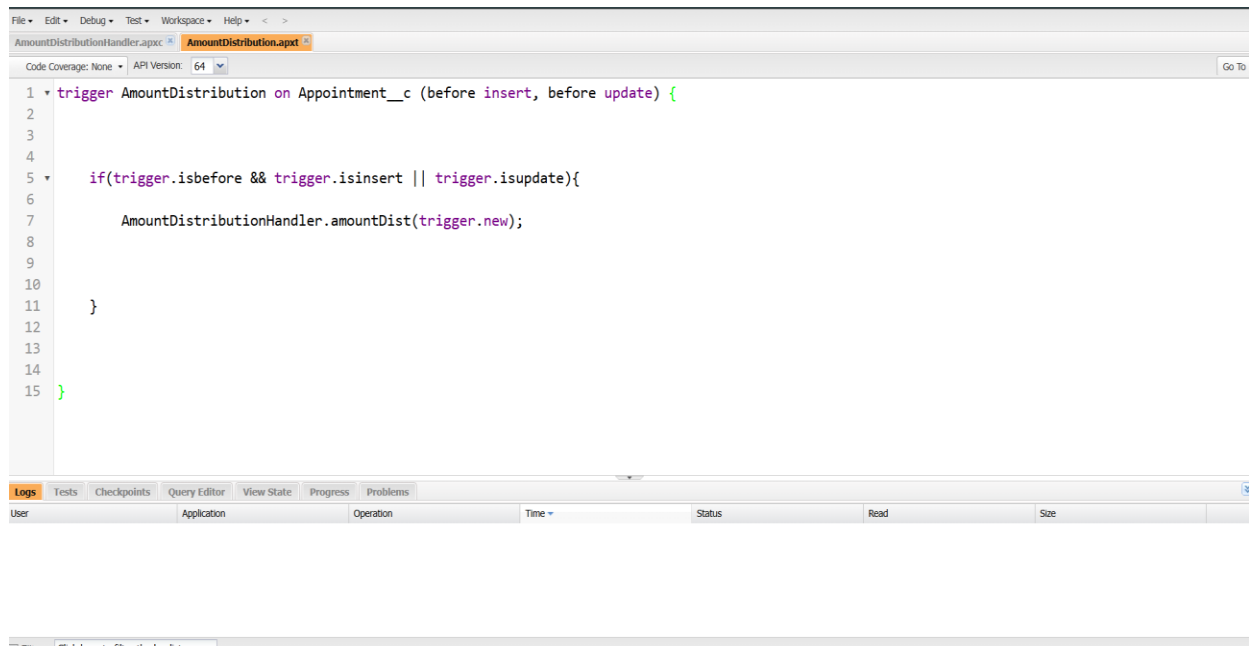
Replacement Parts ☒

\* Vehicle number plate TN01TS4002

Service Amount \$10,000

Cancel Save & New Save

- Scheduled notification reminder emails using Apex class.

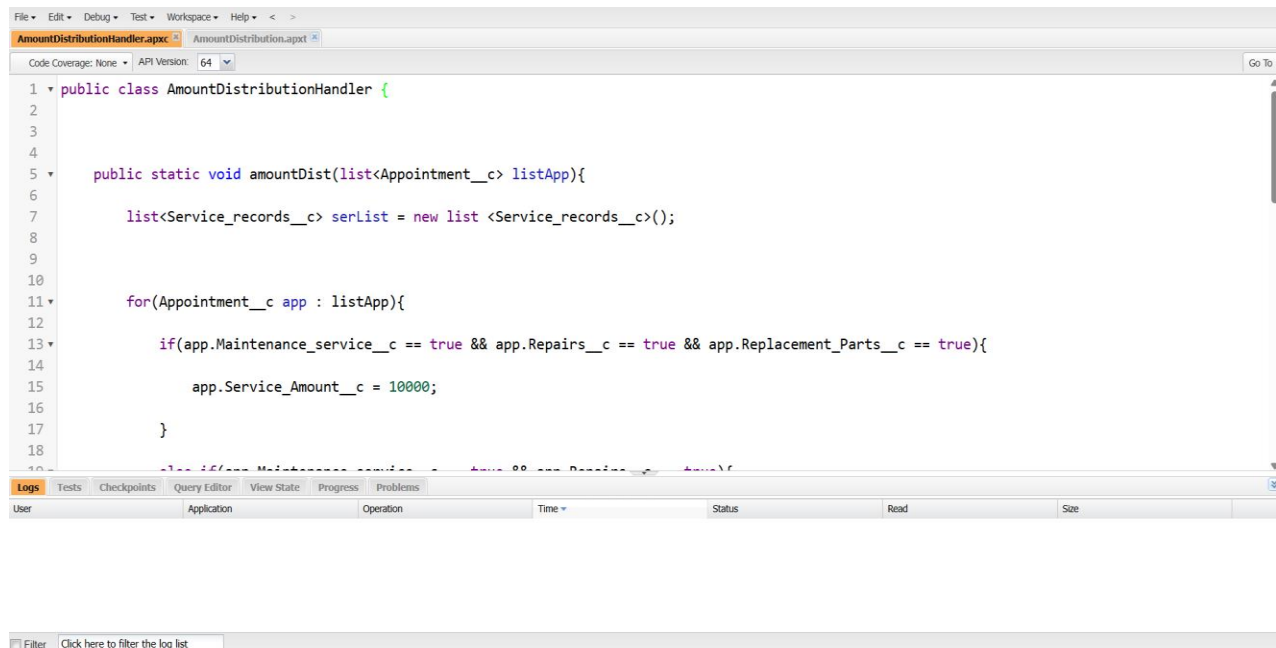


The screenshot shows an IDE window with the file 'AmountDistribution.apex' open. The code is an Apex trigger for the 'Appointment\_\_c' object, triggered before insert and update operations. The trigger calls a static method 'amountDist' on the 'AmountDistributionHandler' class. The IDE interface includes 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 the same 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', 'Repairs\_\_c', and 'Replacement\_Parts\_\_c' are all true. If so, it sets 'Service\_Amount\_\_c' to 10000. The IDE interface is identical 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    }
```

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
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Security</a>	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

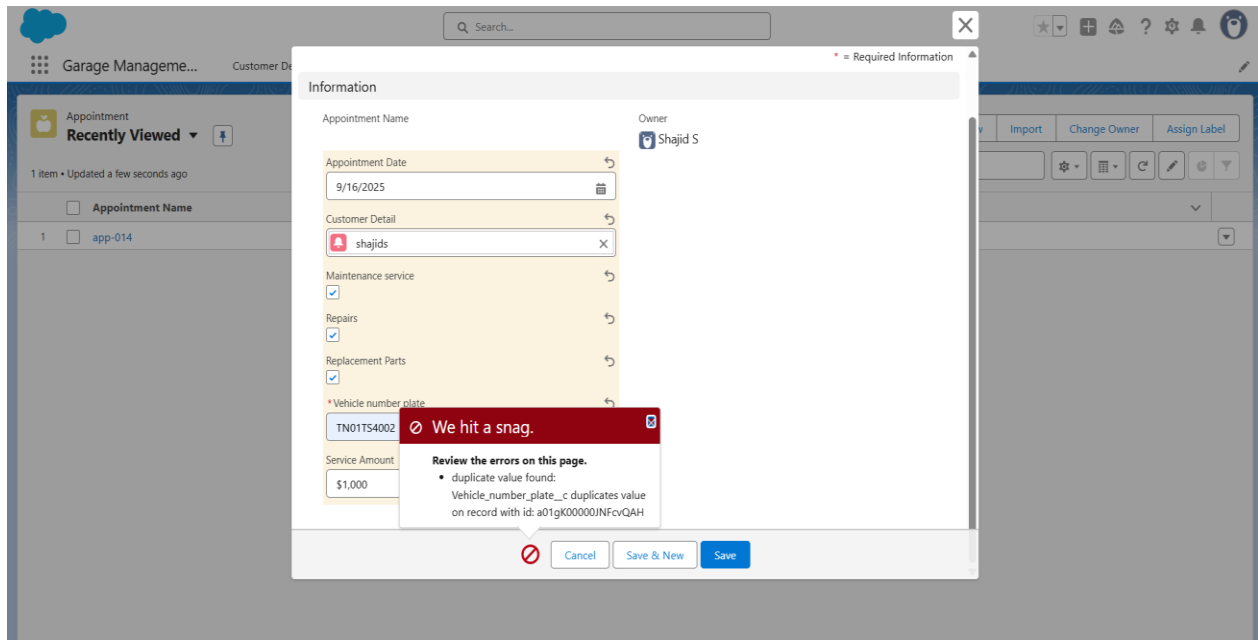
Related Details

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

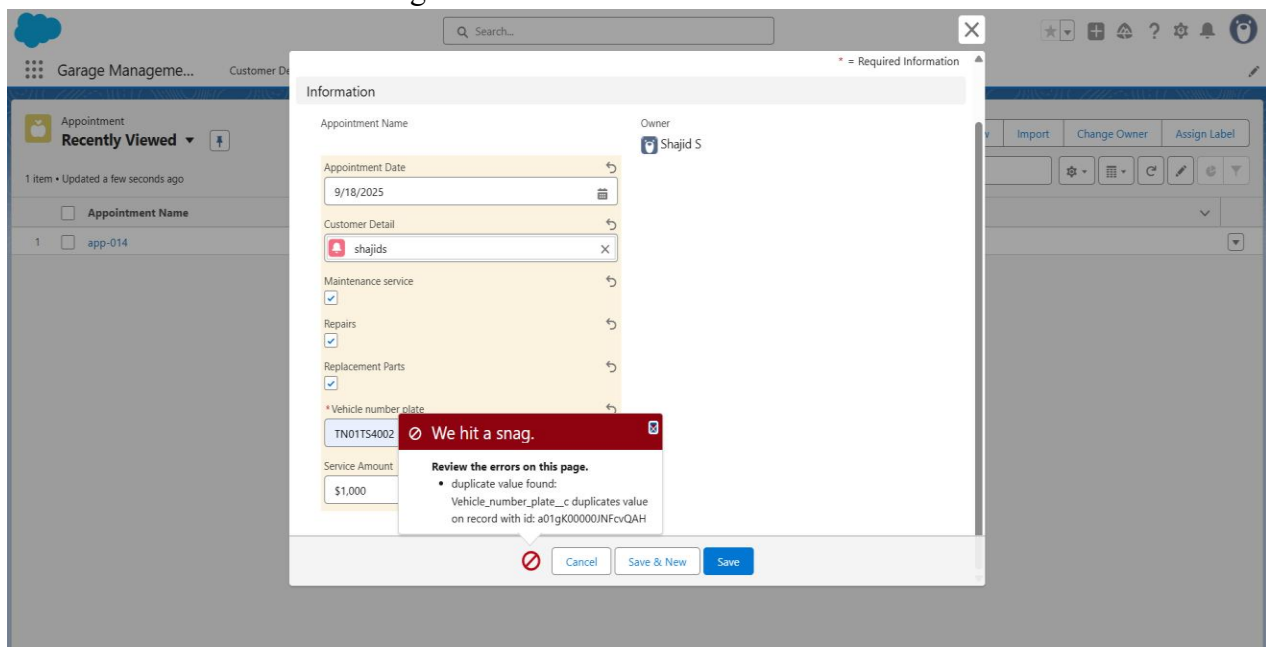
# FUNCTIONAL AND PERFORMANCE TESTING

## Performance Testing

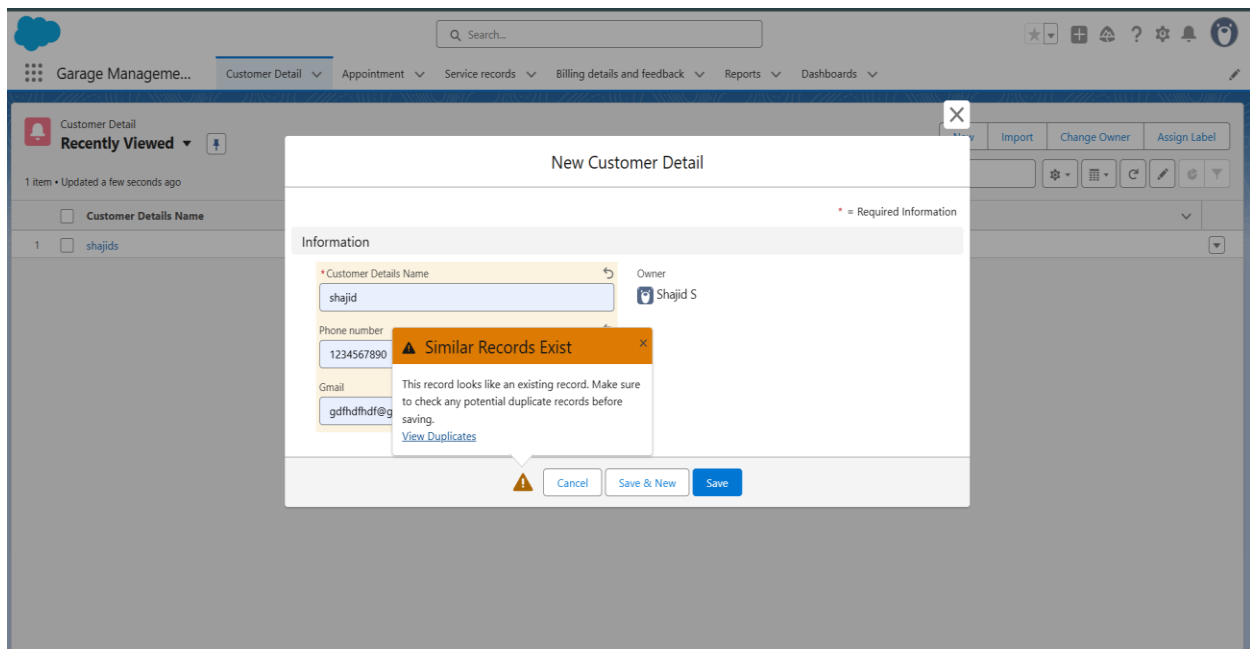
- Trigger validation by entering duplicate vehicle number-property records



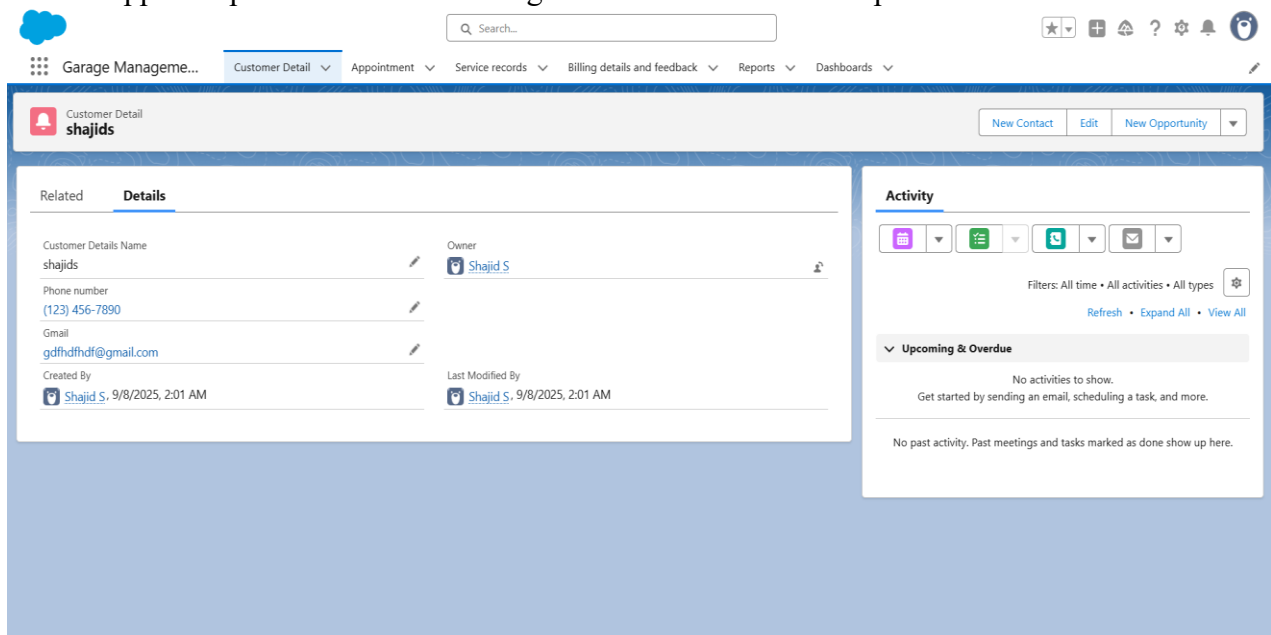
- Validation Rule checking







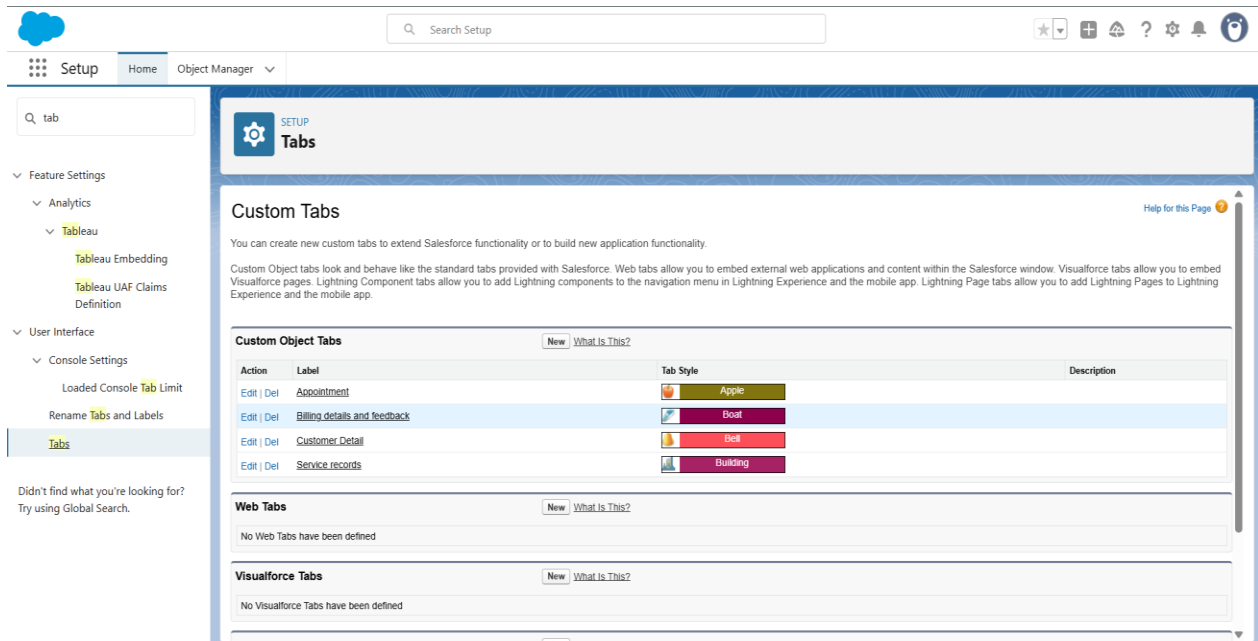
- Approval process validated through email alerts and status updates



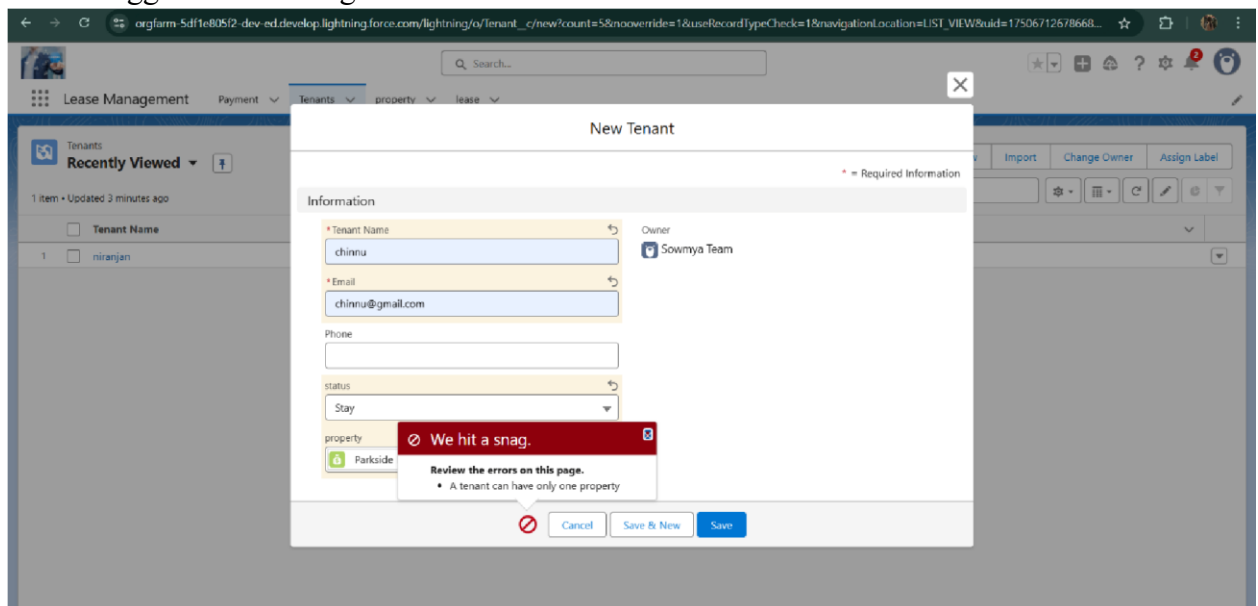
# RESULTS

## 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);  
  
}  
  
}
```