



## ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES (AUTONOMOUS)

UTUKUR (P), C. K. DINNE (V&M), KADAPA, YSR DIST.

Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu.

Approved by S.B.T.E.T. Andhra Pradesh.

Accredited by NAAC with 'A' Grade, Bangalore & NBA (EEE, ECE & CSE).



**Project Report on**  
**SALESFORCE - Garage Management system**  
**BY – B. Reddy Tejaswini(22HM1A0517)**  
**boosiredytejaswini517@gmail.com**

# **Table of Contents**

## **1. Project Overview**

- 1.1 Overview
- 1.2 Key Features & Business Needs

## **2. Objectives**

## **3. Requirement Analysis & Planning**

- 3.1 Understanding Business Requirements
- 3.2 Defining Project Scope & Objectives

## **4. Salesforce Development**

- 4.1 Setup Environment

## **5. Customizations & Automation**

- 5.1 Core System Customizations

## **6. Automation Components**

- 6.1 Flows
- 6.2 Apex Development

## **7. UI/UX Development & Customization**

- 7.1 Lightning App Setup
- 7.2 Reports & Dashboards

## **8. Data Migration, Testing & Security**

## **9. Key Features and Functionalities**

- 9.1 Work Order Management (Appointment & Service Records)
- 9.2 Customer Management
- 9.3 Billing & Feedback

## **10. Deployment, Documentation & Maintenance**

## **11. Conclusion**

# 1. PROJECT OVERVIEW

## 1.1 OVERVIEW

The Garage Management System is a comprehensive web-based tool designed to streamline and optimize the operations of a garage. Built on the Salesforce platform, this system offers a centralized platform for managing vehicle records, service appointments, billing, and customer feedback. The system aims to provide garage owners with a robust infrastructure to improve their business management and efficiency.

GMS empowers garages to:

- Provides access to add customers details.
- Automate service appointments.
- Vehicle service records update.
- Updates reports about the bills based on the records updates.

## 1.2 Key Features & Business Needs

### 1.2.1 Key Features

Feature Category	Key Features
<b>User Management</b>	- Admin and staff login/logout - Role-based access control (Admin, Mechanic, Receptionist) - User registration and profile management
<b>Customer Management</b>	- Add/edit/delete customer details - View customer history and vehicle records - Contact and communication tracking
<b>Vehicle Management</b>	- Register new vehicles - Track service history and upcoming services

### **1.2.2 Business Needs Addressed:**

<b>Business Need</b>	<b>How the Garage Management System Addresses It</b>
Manual record-keeping is time-consuming	Digitizes customer, vehicle, and service records
Lack of inventory control	Tracks spare parts usage and alerts when stock is low
Lack of performance insights	Custom dashboards and reports for day-to-day supervising.
Inefficient workflow management	Assigns jobs to mechanics and monitors service status.

## **2. OBJECTIVES**

### **2. Objectives**

#### **1. Automation of Garage Operations**

- Streamline daily tasks like service entry, billing, and scheduling.

#### **2. Customer Information Management**

- Store and manage customer and vehicle details efficiently.

#### **3. Service History Tracking**

- Maintain records of past services for each vehicle.

#### **4. Billing and Invoicing**

- Generate accurate bills based on service and parts used.

#### **5. Appointment Scheduling**

- Allow customers to book and manage service appointments.

#### **6. Data-Driven Decision Making**

➤ Build dashboards:

- Average service completion time.
- Revenue per customer/vehicle.

# 3. Requirement Analysis & Planning

## 3.1 Understanding Business Requirements

Understanding business requirements means knowing what the garage really needs in order to work better. Before developing the system, we must clearly understand the problems they face, what tasks they do daily, and what features will help them.

This step is important because it ensures the system is useful, practical, and solves real problems.

### Stakeholder Needs:

Component	Business Requirement	Pain Point Addressed
Role Hierarchy	Enforce tiered approvals (e.g., discounts > 20%)	Enforce tiered approvals (e.g., discounts > 20%)
Service Management	Track all service and repair activities	No proper tracking of services done on vehicles
Billing System	Auto-generate bills and invoices	Errors and delays in manual billing

## 3.2 Defining Project Scope & Objectives

### 3.2.1 Project Scope

The scope defines what the project will cover and deliver. For the Garage Management System, the scope includes:

- **Custom Objects:** Vehicle c, Service Booking, and Inventory c with defined relationships to centralize customer, service, and parts data.
- **Duplicate Rules:** Prevent duplicate customer entries (matching on email/phone) and vehicle records (matching on VIN) to eliminate manual cleanup efforts.

- **Role Hierarchy:** A three-tier structure (Admin > Service Manager > Technician) to enforce approval workflows, such as discounts exceeding 15%.
- **Automation:** Flows for auto-assigning technicians and sending SMS/email appointment reminders to reduce no-shows.
- **Validation Rules:** Enforce mandatory fields like Mileage during service bookings to ensure data accuracy.
- **Reports & Dashboards:** Real-time tracking of daily appointments, revenue, and inventory levels for data-driven decisions.

### **3.2.2 Project Objectives**

The objectives are the goals the system should achieve:

1. **Automate daily operations** like service tracking, billing, and inventory.
2. **Reduce manual errors** by digitizing customer and job records.
3. **Save time** for staff and improve work efficiency.
4. **Improve customer service** through fast billing and appointment reminders.
5. **Provide useful reports** to help the garage owner make better decisions.
6. **Ensure secure access** with login for authorized users only.

# 4. SALESFORCE DEVELOPMENT

## 4.1 Setup Environment

We established a robust Salesforce development environment using:

- **Developer Org Strategy:** Created Developer Org sandboxes for development and testing.

link for creating the developer org <https://developer.salesforce.com/signup>

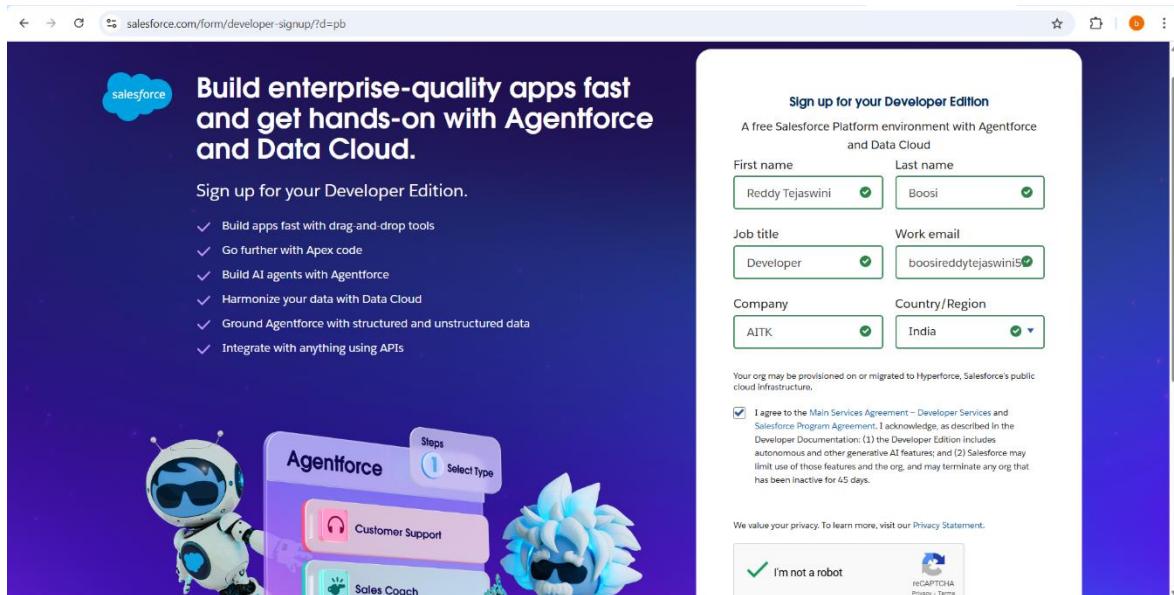


Fig: Signing up for a developer Org

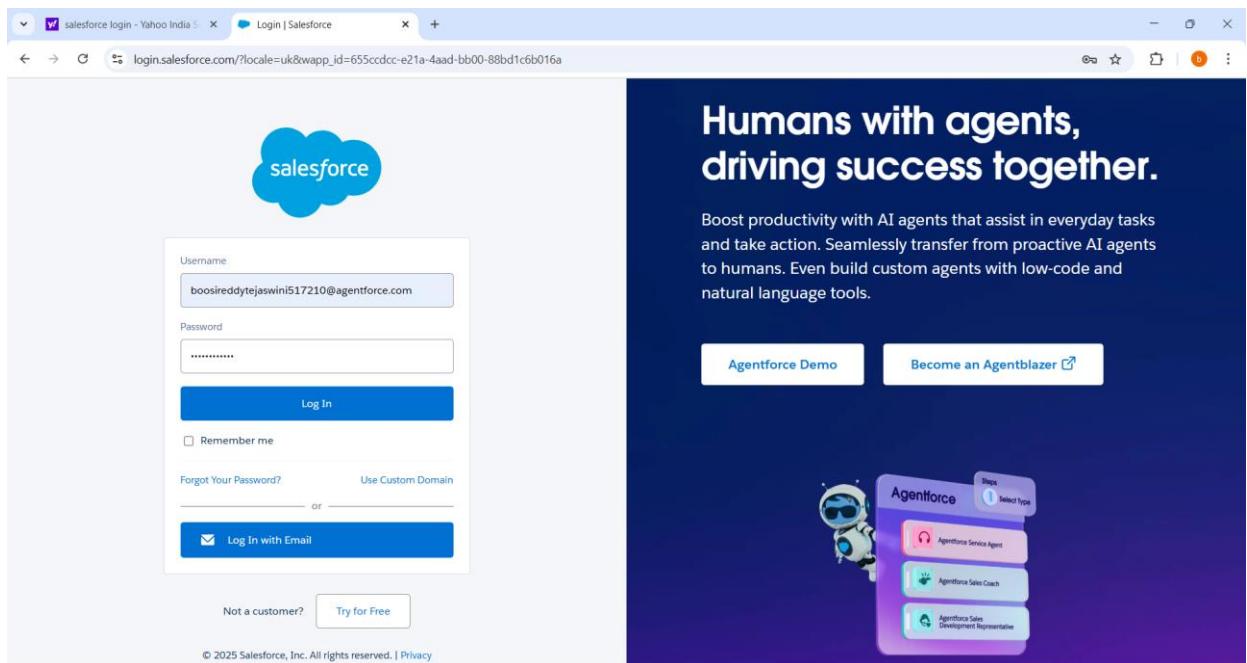


Fig: Developer Org Login

## 5. CUSTOMIZATIONS & AUTOMATION

### 5.1 Core System Customizations

- **Custom Objects & Fields:**

- **Object:**

Salesforce objects are database tables that permit you to store data that is specific to an organization.

#### Create Customer Details Object:

The Customer Details Object is a data structure designed to store and manage key information about a customer.

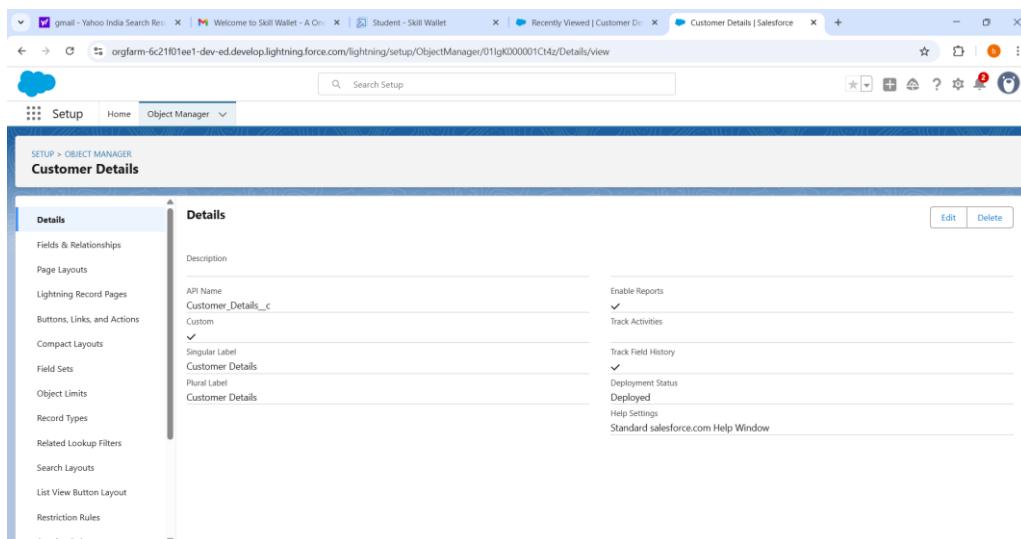
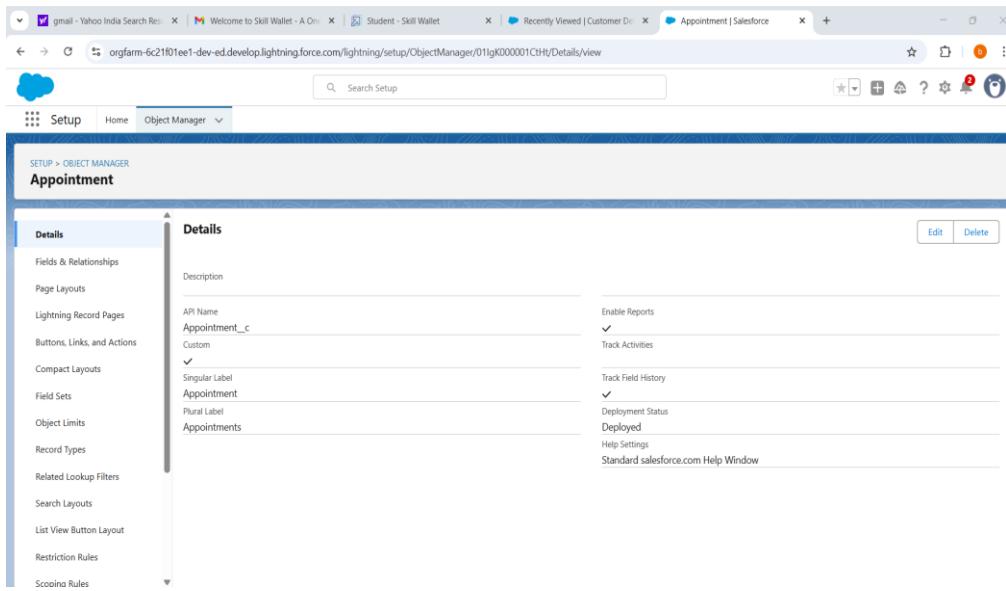


Fig: Customer Object

#### Create Appointment Object:

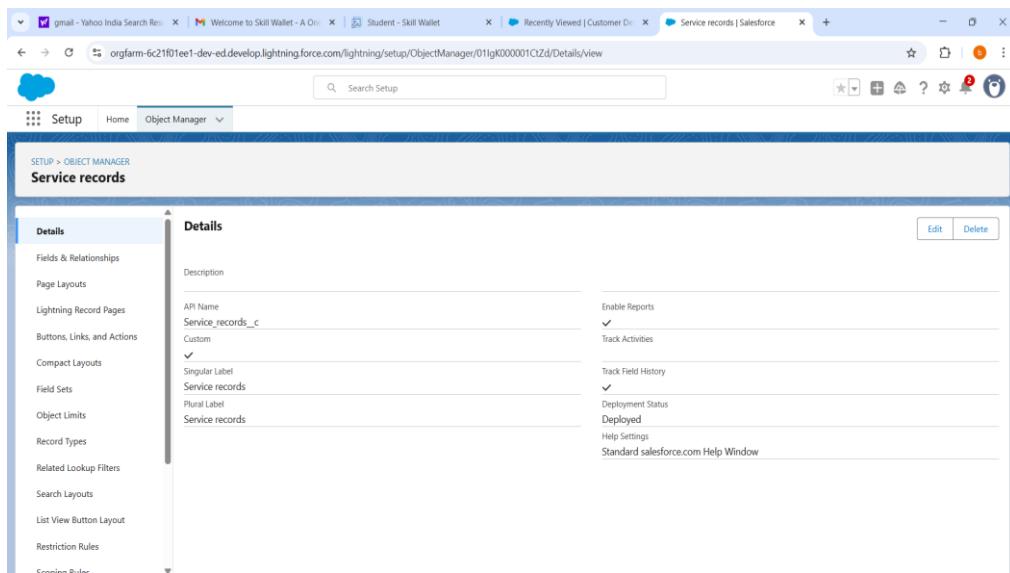
The Appointment Object is a structured data entity used to store and manage information about scheduled meetings, visits.



**Fig: Appointment Object**

## Create Service records Object:

The Service Records Object is a structured representation of all services provided to a customer over time.



**Fig: Service Object**

## Create Billing details and feedback Object:

The Billing Details Object is designed to capture all financial transactions and payment-related information for services or products provided to a customer. It ensures accurate billing, payment tracking, and supports financial reporting and auditing.

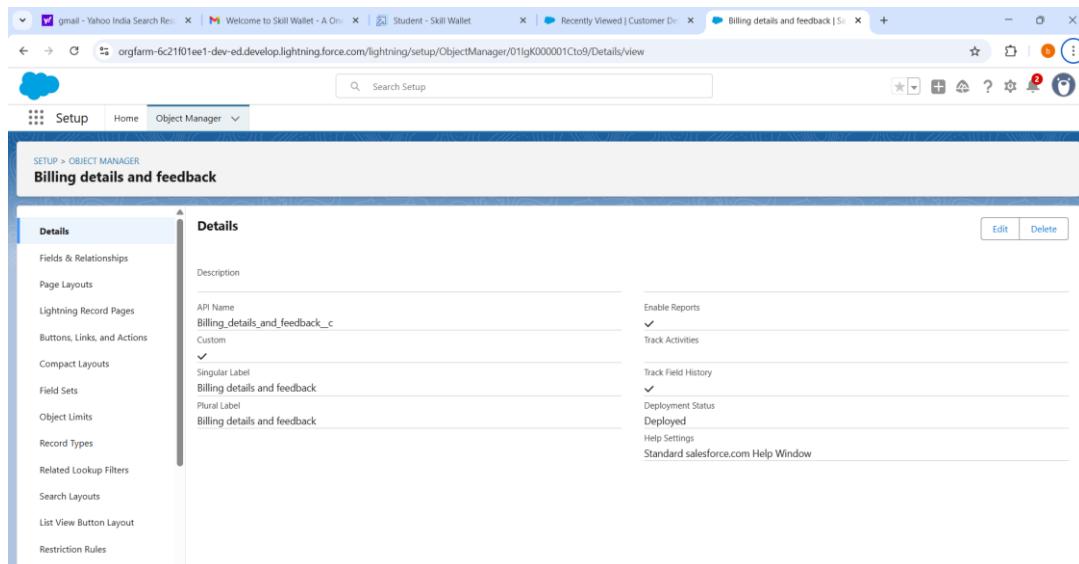


Fig: Billing details and feedback Object

## • Validation Rules:

Validation Rules are conditions or constraints applied to data fields to ensure that the input provided by users is accurate, consistent, and meets predefined business requirements. These rules help maintain data quality, prevent errors, and enforce logical correctness throughout the system.

## Validation Rules for the Appointment Object:

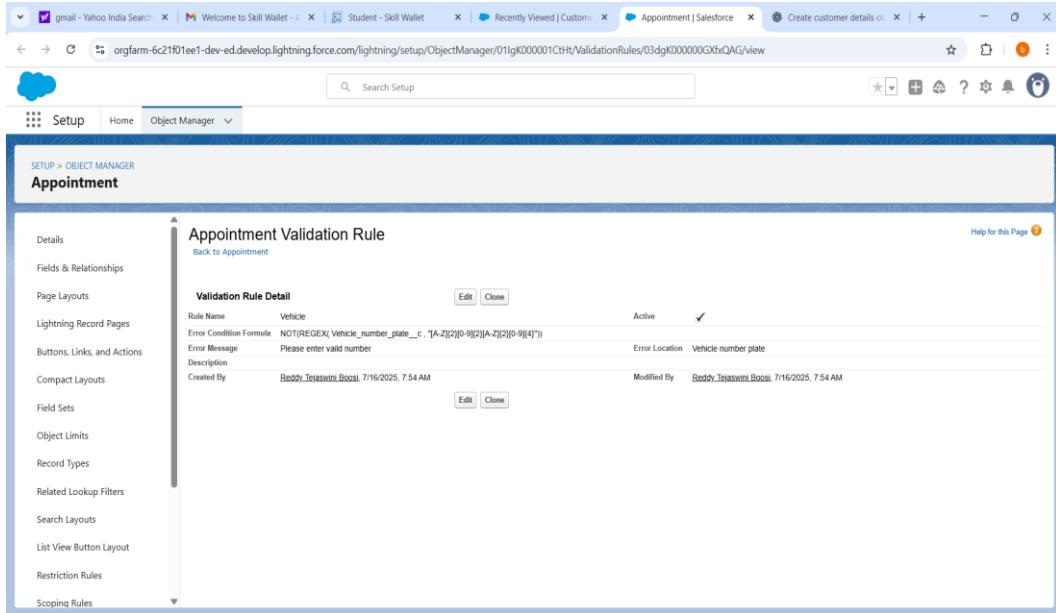


Fig: Validation rules of Appointment

## Validation Rules for Billing details and feedback Object:

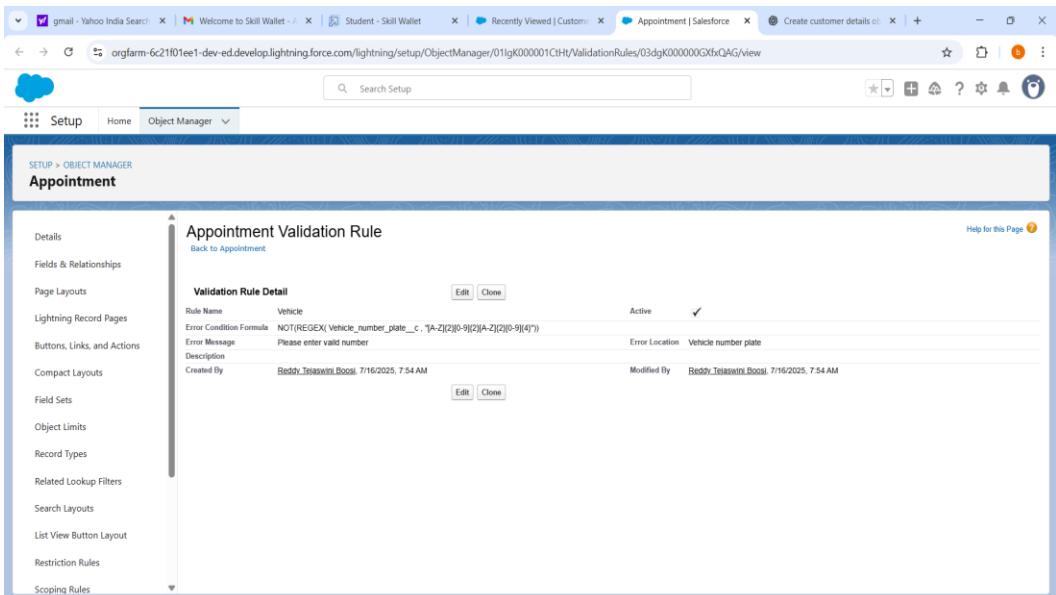


Fig: Validation rules of BD&F

# 6. AUTOMATION COMPONENTS

## 6.1 Flows:

- Billing amount flow is created to send an email alert. Whenever the payment status in Billing details and feedback record is updated as completed for a particular service records the flow automatically sends an email alert as Thank You for Your Payment - Garage Management.
- The Update service status flow is designed for a purpose of updating the service status as completed when the quality service checkbox is selected when editing the service records.

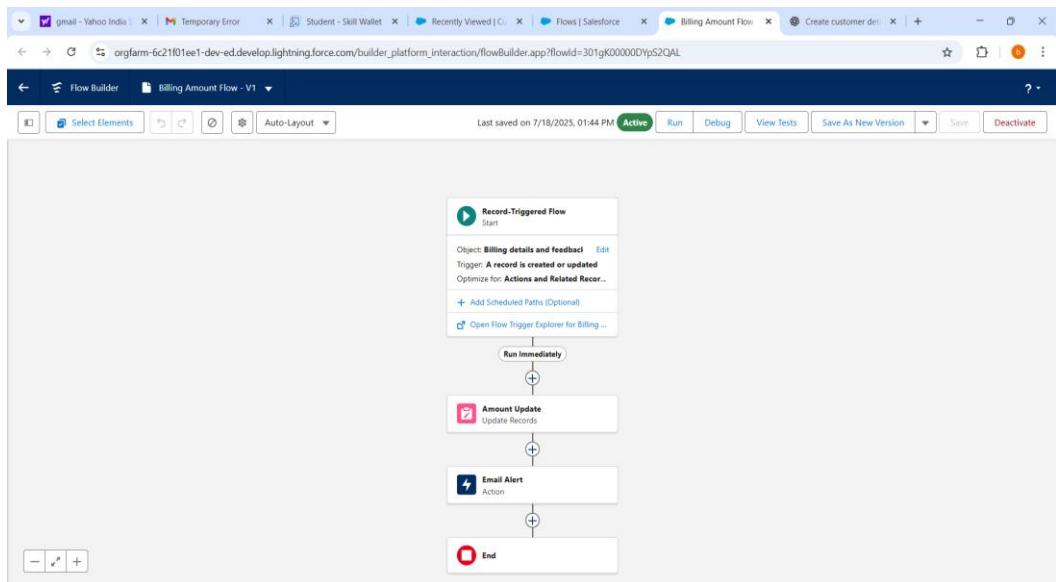


Fig: Billing amount flow

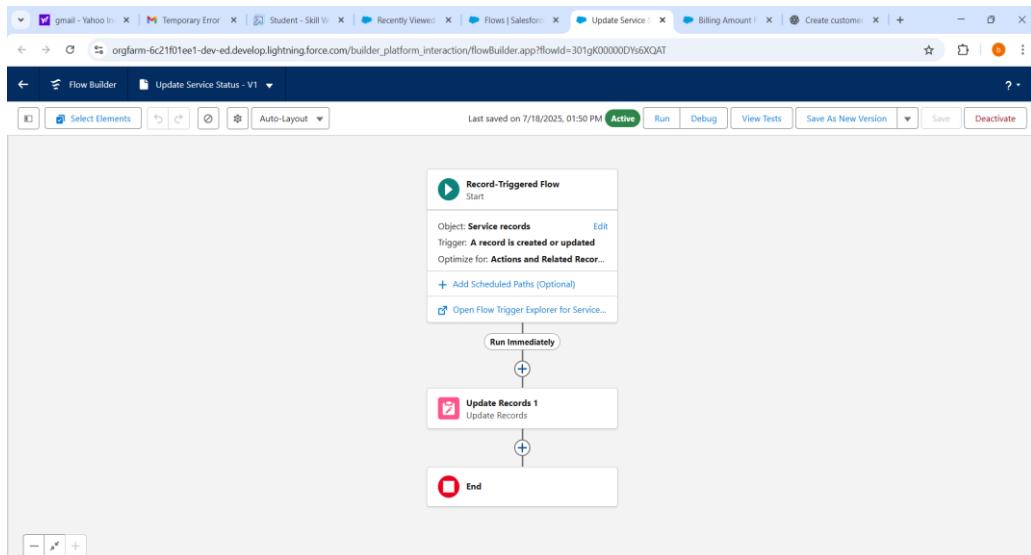


Fig: Update service status

## 6.2 Apex Development:

### Apex Handler:

An Apex Handler is a class in Salesforce that contains the business logic for specific operations like triggers, batch jobs, or flow actions. It is a part of the Trigger Handler Pattern, which promotes separation of concerns, better organization, and improved maintainability of Apex code.

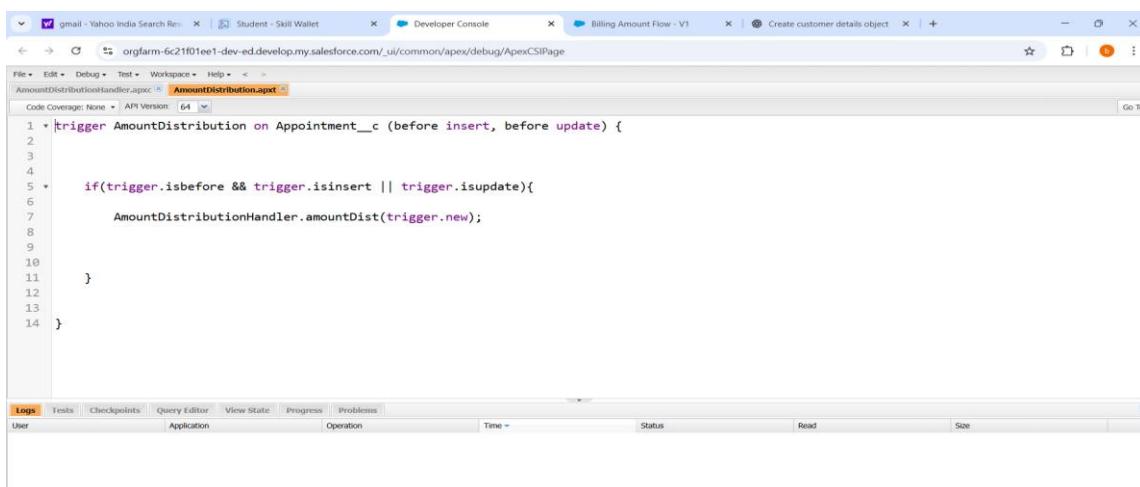
```

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        for(Appointment__c app : listApp){
11            if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
12                app.Service_Amount__c = 10000;
13            }
14        }
15    }
16
17 }
18

```

## Apex Trigger:

- An Apex Trigger is a piece of code in Salesforce that allows developers to perform custom actions before or after events occur on Salesforce records such as insert, update, delete, or undelete.
- Triggers help automate complex business processes by executing logic automatically when a data change happens.



The screenshot shows the Salesforce Developer Console interface. The top navigation bar includes tabs for 'File', 'Edit', 'Debug', 'Test', 'Workspace', and 'Help'. Below the navigation bar, the URL is orgfarm-6c21f01ee1-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage. The main content area displays the code for 'AmountDistribution.apex'. The code is as follows:

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

Below the code editor, there is a toolbar with tabs for 'Logs' (which is selected), 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. There is also a 'Go To' button. At the bottom of the interface, there is a table with columns for 'User', 'Application', 'Operation', 'Time', 'Status', 'Read', and 'Size'.

## 7. UI/UX DEVELOPMENT & CUSTOMIZATION

### 7.1 Lightning App Setup

#### 7.1.1 Custom App Configuration:

- Created "Garage Management Application" Lightning App for accessing all objects, reports and dashboards as follows:

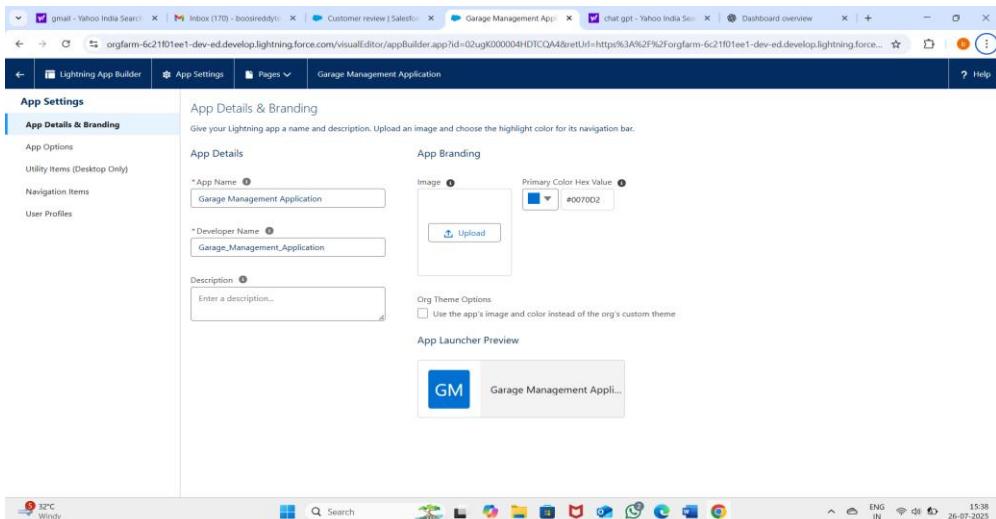


Fig: Lightning app Structure

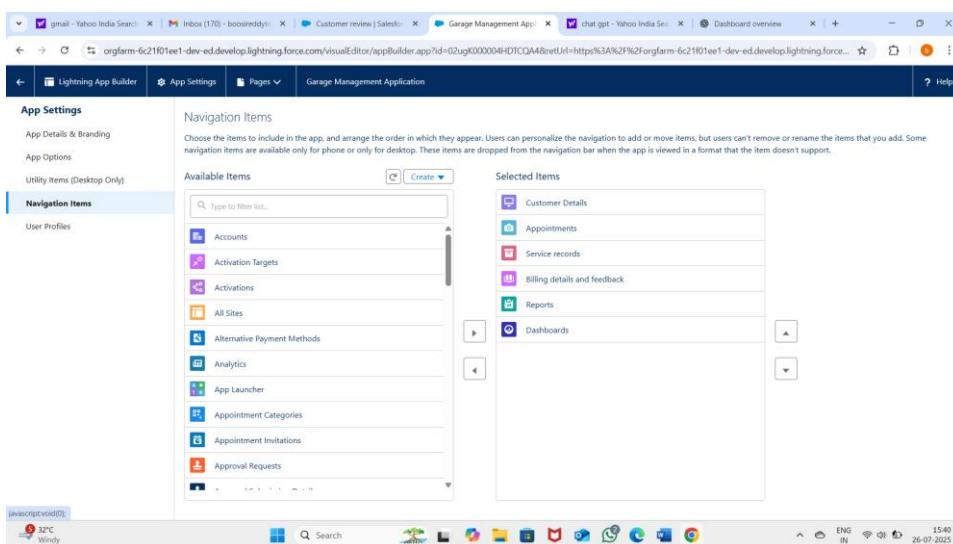
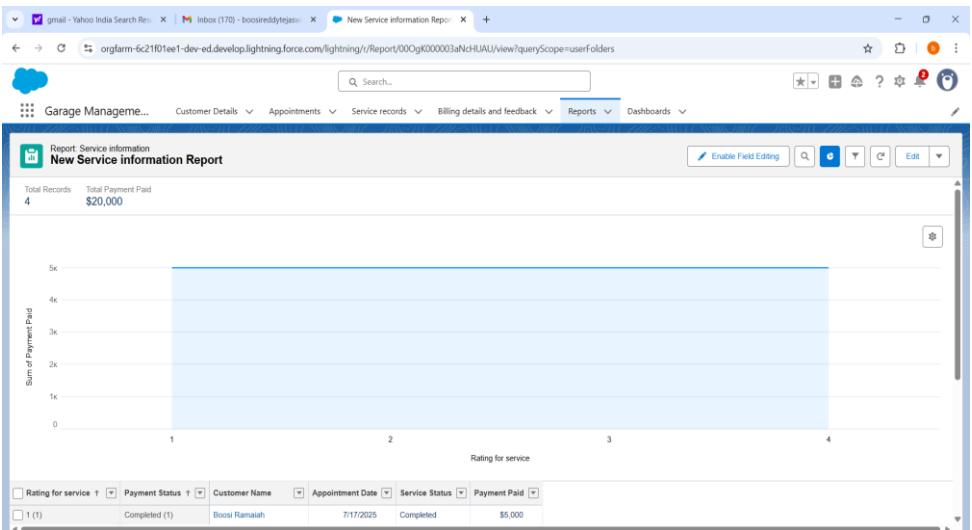


Fig: Objects under Garage Management Application

## 7.2 Reports & Dashboards

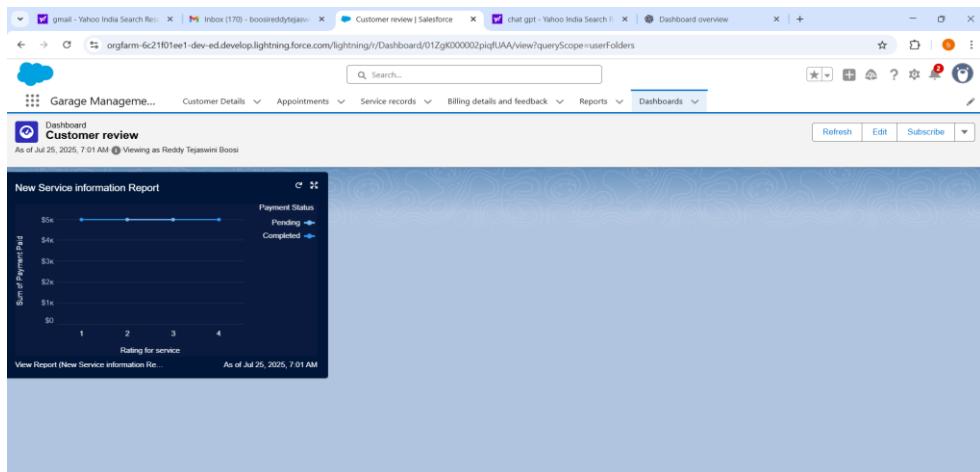
### 7.2.1 Reports:

Reports are structured presentations of data retrieved from the system, used to monitor, and visualize business information. They help users understand performance, identify trends, and make informed decisions based on real-time or historical data.



## 7.2.2 Dashboards:

A Dashboard is a visual representation of data, often displayed using charts, graphs, tables, and key performance indicators (KPIs). It is used in business, project management, marketing, sales, IT, and many other domains.



# 8. DATA MIGRATION, TESTING & SECURITY

## 8.1 Duplicate Rules & Matching Rules:

### Matching Rules:

Matching Rules are logic-based rules used in systems like Salesforce to identify duplicate records or similar entries by comparing fields across objects (e.g., Leads, Contacts, Accounts).

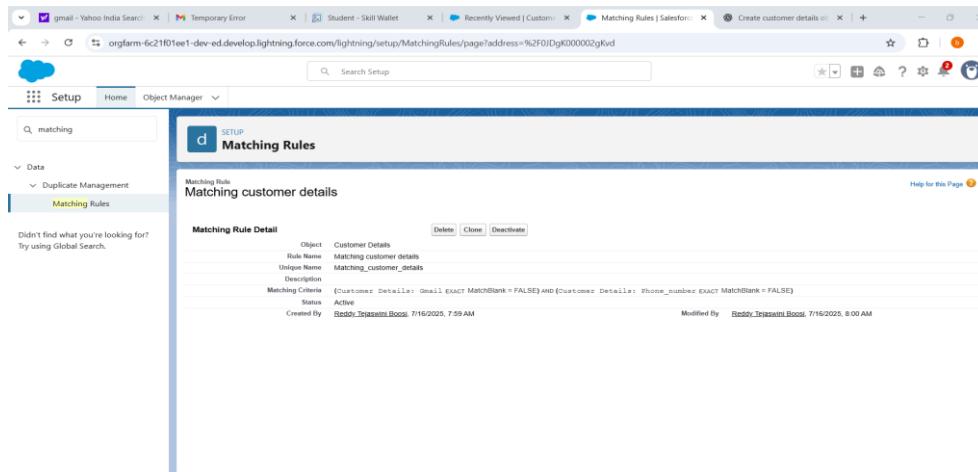
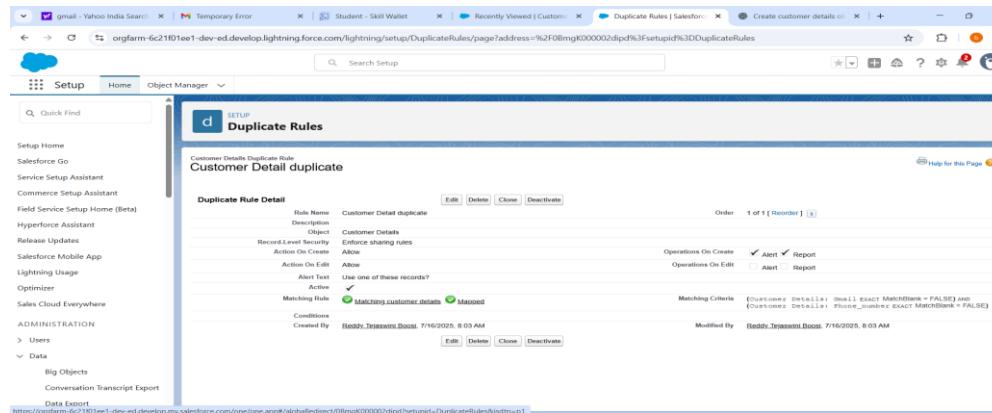


Fig: Matching Rules

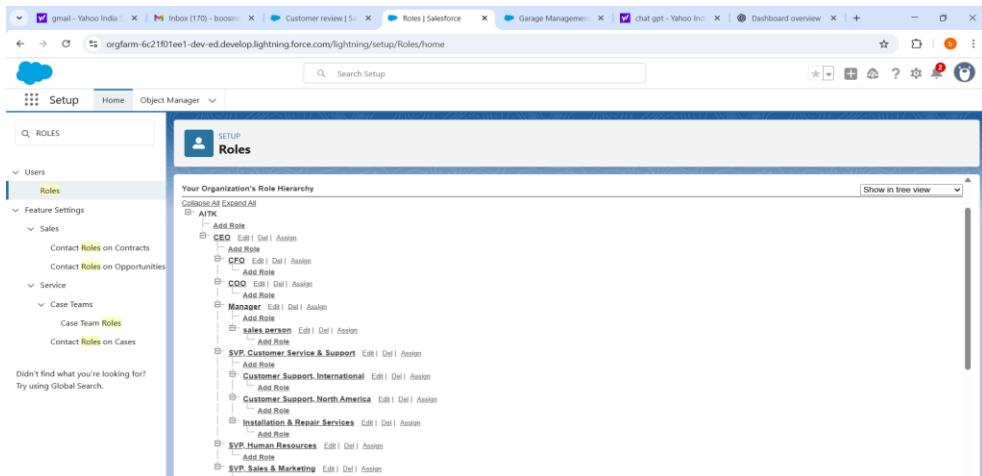
## Duplicate Rules:

Duplicate Rules work with Matching Rules to detect and manage duplicate records in systems like Salesforce. While Matching Rules define *how* duplicates are identified, Duplicate Rules define *what happens* when a potential duplicate is found.



## Roles & Role Hierarchy:

Roles define a user's position in the organization (e.g., Sales Manager, Support Rep) and control data visibility vertically — who can see or access whose records in the system.



## 9. KEY FEATURES AND FUNCTIONALITIES

### 9.1 Work Order Management (Appointment & Service Records)

#### ✓ Overview

Work Order Management refers to the process of creating, tracking, updating, and completing service tasks (or "work orders") that are assigned to field technicians or service agents. It ensures proper scheduling, tracking, and recording of service-related activities.

#### 📁 Key Components

##### 1. Work Order

- A formal request for service or maintenance work.
- Linked to a customer, asset, or product.
- Contains details like status, priority, location, and instructions.

##### 2. Appointment

- A scheduled time slot when a service technician will perform the task.
- Includes time window, assigned technician, and customer availability.

- Often optimized using scheduling algorithms to reduce travel and improve efficiency.

### **3. Service Records**

- Detailed log of what service was performed during the appointment.
- Includes:
  - Tasks completed
  - Materials used
  - Time spent
  - Customer feedback
  - Technician notes
- Can generate invoices, updates to asset history, or future follow-up.

### **9.2 Customer Management**

- Centralized Customer Profiles: The Customer Details object acts as a single source of truth for all customer information, including contact details (Phone number, Gmail) and a unique Customer Name.
- Relationship Tracking: All appointments, service records, and billing details are linked back to the Customer Details, providing a comprehensive view of a customer's history with the garage. This enables personalized service delivery and proactive communication.

### **9.3 Billing & Feedback**

- Automated Billing Integration: The Billing details and feedback object captures payment information. The Payment Paid field is designed to be populated automatically via a Flow when the Payment Status is 'Completed', drawing the amount from the related Service Amount on the Appointment.
- Payment Status Tracking: A picklist field (Payment Status) allows for clear tracking of billing states ('Pending', 'Completed').

## 10. DEPLOYMENT, DOCUMENTATION & MAINTENANCE

### 10.1 DEPLOYMENT:

Deployment is the process of moving metadata (like Apex classes, objects, flows, layouts) from one Salesforce environment (e.g., sandbox or developer org) to another (typically production).

#### Key Methods:

- **Change Sets:** Point-and-click tool in Salesforce UI for deploying between related orgs.
- **ANT Migration Tool:** Java/command-line based; used for scripted deployments.
- **Salesforce DX (SFDX):** CLI-based deployment for modern development (source-driven, scratch orgs).
- **Third-Party Tools:** Copado, Gearset, simplify CI/CD, rollback, and change tracking.

### 10.2 DOCUMENTATION:

Documentation ensures that developers, admins, and users understand how the Garage Management System works, how to maintain it, and how to use it effectively.

#### Key Points:

- **Technical Documentation:**
  - Object schema (e.g., Vehicle, Service Record, Invoice)
  - Apex logic (e.g., triggers on Service Record or status updates)
  - Flows and process builders used
- **Functional Documentation:**
  - Step-by-step guide for using each module (booking service, billing, updating vehicle details)
  - Screenshots and field descriptions

- **Admin Guide:**
  - Setup instructions for new users, role hierarchy, and profiles
  - Permissions for staff (mechanic, cashier, admin)
- 

## **10.3 MAINTENANCE:**

Maintenance refers to ongoing tasks to ensure the Garage Management System remains efficient, secure, and up-to-date with business needs.

### **Key Points:**

- **Error Monitoring:**
  - Use debug logs to catch Apex or flow errors (e.g., servicebilling)
- **Performance Tuning:**
  - Optimize report performance (e.g., Monthly Service Summary)
  - Archive old records to improve speed
- **User Feedback Loop:**
  - Collect feedback from garage staff and customers
  - Improve features like appointment rescheduling or invoice printing
- **Security Maintenance:**
  - Regularly review user permissions
  - Run Salesforce Health Check for vulnerabilities

## **11.CONCLUSION**

The Garage Management System (GMS) simplifies and streamline all garage operations—right from customer appointments to service tracking, billing, and feedback management. With its automation features and integration capabilities (especially when deployed in platforms like Salesforce), it enhances operational efficiency, customer satisfaction, and data accuracy. It helps garage staff handle daily workflows more effectively while providing customers with a smooth and transparent service experience. Overall, GMS is a reliable, scalable, and smart solution for modern garage businesses aiming to improve productivity, maintain service records, and grow sustainably.