

GARAGE MANAGEMENT SYSTEM



College Name : IDEAL INSTITUTE OF TECHNOLOGY

Team ID : LTVIP2025TMID29772

Team members:

1. Team Leader : Bhagyasri Muddaka

Email: Bhagyasrimuddaka@gmail.com

2. Team member : Akula Venkata Rama Krishna

Email: venkataramakrishna2705@gmail.com

3. Team member : Aishwarya Kakarlapudi

Email: Aishwaryakakarlapudi07@gmail.com

4. Team member : Abbireddy Naga Gangadhar

Email: nagagangadharabbireddy0@gmail.com

1. INTRODUCTION:

1.1 Project Overview:

The Garage Management System (GMS) is a comprehensive Salesforce-based application designed to streamline and automate the core operations of an automotive garage. Built using Salesforce's low-code/no-code tools—including custom objects, tabs, automation flows, reports, and dashboards—this system enhances service delivery, ensures better customer engagement, and boosts overall efficiency.

The system allows garage managers and employees to handle the full lifecycle of a customer service request, from customer registration and appointment scheduling to service tracking, billing, and feedback collection. With various modules like Customer Details, Appointments, Service Records, and Billing and Feedback, each built as custom Salesforce objects, the application enables accurate and real-time tracking of all garage activities.

Additionally, the system is enriched with tools like Validation Rules, Apex Triggers, Email Alerts, and Dashboards, providing a highly interactive and automated environment. The solution supports role hierarchies, public groups, and sharing settings to ensure appropriate data access and collaboration among users such as managers and sales personnel.

1.2 Purpose:

The primary purpose of the Garage Management System is to digitize and centralize the operations of a garage to:

- Improve service efficiency: Automate routine tasks like appointment bookings, service tracking, and invoice generation.
- Enhance customer satisfaction: Keep customers informed via email alerts and provide transparency on services and payments.
- Enable better decision-making: Utilize reports and dashboards to monitor trends like service quality, revenue generation, and customer feedback.
- Ensure data integrity and control: Implement validations, role hierarchies, and sharing settings to maintain secure and accurate records.
- Simplify employee management: Assign user roles and permissions to managers and sales staff for better organizational structure.

2. IDEATION PHASE:

2.1 Problem Statement:

Current Challenge:

Many small to mid-sized automotive garages still manage their customer data, service records, appointments, and billing manually or using basic software that lacks integration, automation, and real-time tracking. This results in:

- Missed or double-booked appointments
- Unclear service tracking and quality status
- Inaccurate or delayed billing
- Poor customer communication and feedback management
- Lack of analytical tools to make business decisions

Problem Statement:

"Garage businesses lack a centralized, automated system to manage service operations, customer interactions, and billing, which hampers their efficiency, service quality, and decision-making ability."

2.2 Empathy Map Canvas:

The goal of this empathy map is to understand the **customer's perspective** when they interact with the garage for vehicle servicing or repair.

Key Components:

SAYS:

- "We need a better way to track services and customers."
- "Paperwork takes too much time."
- "It's hard to manage appointments properly."
- "Billing is confusing and slow."
- "We miss out on collecting customer feedback."

THINKS:

- "Are customers unhappy with our delays?"

- "We might be losing business due to poor tracking."
- "Can we automate this to save time?"
- "What if we forget to bill someone?"
- "Is there a better way to view our daily workload?"

DOES:

- Manually enters customer and service details in notebooks or Excel.
- Calls or messages customers to confirm appointments.
- Calculates bills using a calculator or spreadsheet.
- Relies on memory for tracking service progress.
- Collects feedback only when customers complain.

FEELS:

- Frustrated by repetitive manual work.
- Anxious about making errors in appointments or billing.
- Stressed due to disorganized records.
- Motivated to modernize operations.
- Curious about how technology can help.

PAINS:

- Time wasted in manual data entry.
- Missed or overlapping appointments.
- Incorrect or forgotten billing.
- No customer history tracking.
- Poor visibility into service quality.

GAINS:

- Desire for a simple system to manage customers and services.
- Need for automatic billing and service tracking.
- Wish to send professional email updates to customers.
- Aim to generate insightful reports and dashboards.
- Want to focus more on service quality and less on admin tasks.

2.3 Brainstorming:

Goal:

To generate innovative and practical ideas for solving the core problems identified in the problem statement using Salesforce tools.

Brainstormed Ideas and Features:

1. Centralized Customer Profile Management

- Store customer details including phone, email, vehicle number plate.
- Use validation and duplicate rules to prevent data redundancy.

2. Automated Appointment Scheduling

- Use custom objects and date fields for booking.
- Implement validation rules to ensure no overlap or incorrect data entry.

3. Service Tracking via Checkboxes & Picklists

- Checkboxes for services selected (maintenance, repair, replacement).
- Picklist for service status: Started / Completed.
- Use flows to automatically update statuses based on field actions.

4. Dynamic Billing and Payment Flow

- Auto-calculate service amount using Apex Triggers based on selected services.
- Generate payment records upon completion.

5. Automated Customer Email Alerts

- Send thank-you emails after payment using record-triggered flow and email actions.

6. Quality Check & Service Completion

- Use checkbox to mark quality check.
- Automatically update service status upon completion.

7. Role-Based Access

- Create roles like Manager and Salesperson.
- Apply sharing rules and profiles for secure access.

8. Public Groups & Sharing Rules

- Share data appropriately across roles (e.g., Salesperson to Manager with read/write access).

9. Reports & Dashboards

- Line charts to track service ratings and payments.
- Weekly dashboard email subscriptions to monitor performance.

10. Mobile-Friendly Lightning App

- Create a branded Lightning App with relevant tabs and navigation.

3.REQUIREMENT ANALYSIS:

3.1Customer Journey Map:

Stage	Customer Actions	Touchpoints	Emotions	Pain Points	Opportunities
1. Awareness	Learns about garage through ads or referrals	Social media, Flyers, Word of mouth	Curious, Cautious	No online presence, no way to check reviews	Add social proof & ratings in Salesforce record or website
2. Appointment Booking	Calls or visits to book appointment	Phone call, Walk-in, Online form (future)	Hopeful, Slightly anxious	No booking confirmation, slow manual entry	Use Salesforce Appointment object with validation & date checks
3. Service Check-In	Brings vehicle for service	Reception desk, Service staff	Trusting, Expectant	Long wait time, manual check-in	Streamline check-in using Salesforce record lookup & mobile tabs
4. Service in Progress	Waits or receives updates	Mechanic updates, WhatsApp/SMS (manual)	Anxious, In the dark	No updates on progress	Add service status field + dashboard tracking in customer view
5. Billing & Payment	Receives bill, makes payment	Paper invoice, Cash/Card	Confused, Sometimes frustrated	No clarity on billing items, manual totals	Auto-calculate bill using Apex logic based on service checkboxes

6. Feedback & Follow-up	Leaves or forgets to leave feedback	Verbal, Sometimes no feedback collection	Neutral, Disengaged	Feedback not recorded or lost	Collect via Salesforce form, validate 1–5 rating, send thank-you mail
7. Retention	Decides whether to return or recommend	None (unless garage follows up)	Indifferent or Loyal (varies)	No follow-up or customer appreciation	Use automation to send service reminders or loyalty offers

3.2 Solution Requirement:

Functional Requirements:

1. Customer Module:

- Create and manage customer profiles
- Validate emails and phone numbers
- Prevent duplicate entries

2. Appointment Management:

- Schedule appointments with auto-generated IDs
- Track service status using checkboxes
- Date and time validations

3. Service Records:

- Track start/completion of service
- Capture service type and quality check
- Use picklists for service status

4. Billing and Feedback:

- Auto-calculate cost using Apex trigger
- Collect and validate customer ratings
- Capture payment status (pending/completed)

5. Notifications:

- Send automated emails post payment
- Trigger alerts for service status update

6. Reports & Dashboards:

- Generate service reports
- Visualize service quality and earnings via dashboards

7. Security & Roles:

- Role-based access: Manager, Sales Person
- Public groups for team access
- Sharing rules to manage data access

Non-Functional Requirements:

- **Scalability:** Should support multiple users and records
- **Availability:** Accessible 24/7 on the Salesforce cloud
- **Security:** Field-level and object-level permissions
- **User-Friendly UI:** Tabs and Lightning App for intuitive navigation
- **Automation:** Use of Flows, Validation Rules, and Triggers

3.3 Data Flow Diagram:

A Data Flow Diagram (DFD) for the Garage Management System visually represents how data moves between users, processes, and data stores within the system. It shows how customers, salespersons, and managers interact with processes like booking appointments, recording services, billing, and report generation, and how this data is stored and updated across different modules in a structured, organized way.

Context Level (Level 0) DFD:

This is the **highest-level** view of the system, showing the interaction between **external entities** and the **main system**.

Entities:

1. Customer
2. Manager
3. Sales Person

Process:

- **Garage Management System (GMS)** – The core process that handles all operations.

Data Flows:

- Customers send **appointment requests, vehicle details**, and receive **service status, bill, and feedback confirmation**.
- Managers monitor service records, billing, feedback, and generate **reports**.
- Salespersons handle **appointment entries** and customer coordination.

Data Stores (Implied in Level 0):

- Appointment Data
- Customer Data
- Service Record
- Billing & Feedback

Level 1 DFD: Breaks down the core system into major sub-processes

Process 1: Manage Customer Details

- **Input:** Customer personal info (name, phone, email)
- **Output:** Stores data in **Customer Database**
- **Entity involved:** Customer, Sales Person

Process 2: Book Appointments

- **Input:** Vehicle details, services selected, date & time
- **Processing:** Validate entries, check availability
- **Output:** Confirmation, updates appointment store
- **Entities involved:** Customer, Sales Person

Process 3: Record Service Details

- **Input:** Appointment ID, service status (started/completed), quality check
- **Output:** Updates **Service Records**
- **Entity involved:** Garage Technicians (via internal system)

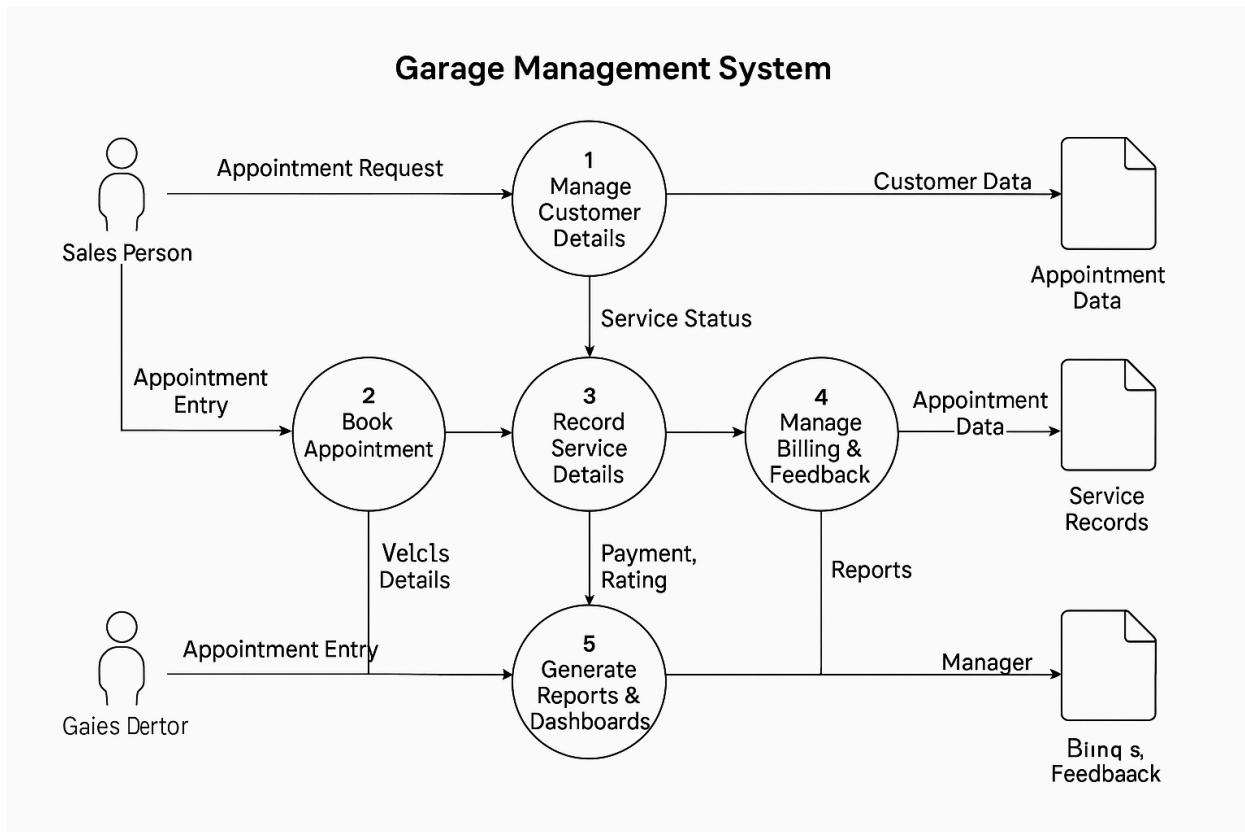
Process 4: Manage Billing & Feedback

- **Input:** Service completed, payment status, rating, comments
- **Processing:** Calculate total, auto-update if “Completed”
- **Output:** Stores in **Billing & Feedback** data store
- **Entity involved:** Customer, Manager

Process 5: Generate Reports & Dashboards

- **Input:** All data stores (Customer, Appointment, Service, Billing)
- **Processing:** Filter, sort, group, calculate

- **Output:** Visual charts, downloadable reports
- **Entity involved:** Manager



3.4 Technology Stack:

The technology stack for the Garage Management System is a set of Salesforce-based tools, frameworks, and configurations used to build, automate, secure, and manage all functionalities of the garage—such as customer management, service appointments, billing, and reporting—within a cloud environment.

This stack includes:

- **Salesforce Lightning Platform** as the core platform
- **Custom and standard objects** for data management
- **Apex classes and triggers** for custom logic
- **Lightning App Builder** for creating user interfaces
- **Flows and validation rules** for automation and data integrity
- **Profiles, roles, and sharing settings** for security
- **Reports and dashboards** for real-time analytics

- **Developer Console & Setup tools** for system configuration and deployment

Platform:

- **Salesforce Lightning Platform** – Core platform to build, run, and manage the entire application in the cloud.

Frontend (User Interface):

- **Lightning App Builder** – For building custom pages, tabs, and navigation without coding.
- **Lightning Experience UI** – Modern, responsive user interface for desktop and mobile users.
- **Visualforce (optional)** – For creating advanced or customized UI screens (if needed).

Backend (Business Logic):

- **Apex Classes** – Used for implementing custom logic like billing automation.
- **Apex Triggers** – Automatically execute logic before/after data changes.
- **Record-Triggered Flows** – Automate updates, notifications, and calculations without code.

Database Layer:

- **Standard & Custom Objects** – Store and manage data (e.g., Customer, Appointment, Service, Billing).
- **SOQL (Salesforce Object Query Language)** – Used to query data from Salesforce objects.

Automation:

- **Flows & Process Builder** – For automating service status updates, email alerts, and field updates.
- **Validation Rules** – Ensure data correctness (e.g., vehicle number format, rating limit).
- **Duplicate Rules** – Prevent duplicate customer entries.

Security:

- **Profiles & Roles** – Control access based on job role (Manager, Salesperson).
- **Sharing Settings** – Manage record-level access and visibility.

Reporting & Analytics:

- **Reports** – Generate service summaries, payment history, and feedback analysis.

- **Dashboards** – Visualize KPIs like rating trends, appointment volume, and revenue.

Communication:

- **Email Alerts (via Flow)** – Send automated thank-you or payment confirmation emails.

Development & Deployment Tools:

- **Salesforce Setup** – Configure the entire system (objects, fields, permissions, etc.).
- **Developer Console** – Write, test, and debug Apex code.
- **Data Loader** – Bulk data import/export.
- **Workbench** – Tool for querying and inspecting data.

Authentication & Access:

- **Salesforce User Login** – Role-based login with password policies and session timeout settings.

Summary:

The Garage Management System uses the **Salesforce Lightning Platform** as its core technology. It leverages **custom objects** for data storage, **Apex triggers and flows** for automation, **Lightning App Builder** for UI design, and **reports & dashboards** for analytics. Security is managed through **profiles, roles, and sharing settings**, while all logic and configuration are handled within Salesforce's no-code/low-code environment.

4. PROJECT DESIGN

4.1 Problem-Solution Fit:

Problem Recap:

Small and mid-sized garage businesses struggle with:

- Manual customer and appointment tracking
- Service delays and miscommunication
- Errors in billing and feedback collection
- Lack of visibility into garage performance

Solution Fit:

The **Garage Management System (GMS)** fits the problem by providing a fully automated, cloud-based solution using Salesforce. It:

- Digitizes customer and appointment data
- Automates service status tracking and billing
- Sends real-time email notifications
- Visualizes performance through reports and dashboards
- Offers role-based access for better collaboration and data security

Conclusion:

GMS solves the exact operational inefficiencies the garages face by replacing manual, fragmented processes with a smart, integrated Salesforce application.

4.2 Proposed Solution

The Proposed Solution is a Salesforce-based Garage Management Application that includes:

Core Modules:

1. Customer Details Module:

- Store name, contact info, and vehicle number
- Duplicate and validation rules to ensure clean data

2. Appointment Module:

- Auto-generated appointment IDs
- Capture date, selected services (checkboxes), and vehicle info
- Validation for license plate format and date

3. Service Records Module:

- Track service progress
- Use picklists and checkboxes for service status and quality check
- Lookup relationship with Appointment

4. Billing & Feedback Module:

- Auto-calculated service amount via Apex trigger
- Capture payment status and rating
- Email confirmation sent post-payment

5. Reports & Dashboards:

- Generate real-time reports (e.g., payments, services)
- Dashboards to monitor trends like ratings and revenues

Automation & Intelligence:

- **Flows:** Automatically update fields and send emails
- **Apex:** Handles service pricing logic
- **Validation Rules:** Ensure clean data entry
- **Email Alerts:** Notify customers after service or payment

Access Management:

- Roles: Manager, Salesperson
- Public Groups & Sharing Settings: Secure yet collaborative data access
- Profiles: Define what each user can see or do

4.3 Solution Architecture (in Detail):

Here's the detailed architecture broken into logical components:

1. Data Layer:

- **Custom Objects:**
 - Customer_Details__c
 - Appointment__c
 - Service_Records__c
 - Billing_Feedback__c
- **Relationships:**
 - Lookup between objects (e.g., Appointment → Customer, Service → Appointment)

- **Fields:**

- Text, Picklist, Date, Currency, Checkbox, Formula

2. Business Logic Layer:

- **Apex Trigger:**

- **AmountDistribution** handles pricing logic based on services selected

- **Apex Class:**

- **AmountDistributionHandler** calculates amount dynamically

- **Flows:**

- Flow 1: Updates payment amount and sends email upon payment status = "Completed"
 - Flow 2: Updates service status when quality check is true

- **Validation Rules:**

- License plate format
 - Rating range (1–5)

3. Application Layer:

- **Lightning App:**

- Navigation tabs for each module

- **Custom Tabs:**

- One for each object (Customer, Appointment, etc.)

- **UI Controls:**

- Checkbox fields for service types
 - Picklists for service/payment status

4. Security & Access Layer:

- **Roles & Role Hierarchy:**

- Manager (higher role)
- Salesperson (lower role)

- **Public Groups:**

- “Sales Team” group includes all salespeople

- **Sharing Rules:**

- Share records from Salesperson to Manager (read/write)

- **Profiles:**

- Controls access to tabs, fields, and objects

5. Reporting & Analytics Layer:

- **Custom Reports:**

- Joined reports for all modules (Service Info Report)

- **Dashboards:**

- Weekly line chart for service ratings
- Email subscription enabled for monitoring

6. Notification Layer:

- **Flow Email Actions:**

- Personalized emails with customer name, service amount, and thank-you note

- **Text Template:**

- Used in the body of the flow email

5. PROJECT PLANNING & SCHEDULING:

5.1 Project Planning:

1. Project Overview:

Objective:

Develop a web-based Garage Management System to streamline customer bookings, manage vehicle data, handle invoices, monitor inventory, and provide admin dashboards.

Target Users:

- Admin (Garage owner)
- Mechanic
- Customer

Technology Stack (Suggested):

- **Frontend:** HTML, CSS, JS (or React)
- **Backend:** Python (Flask or Django) / Node.js
- **Database:** MySQL / PostgreSQL / MongoDB
- **Deployment:** Heroku / Render / AWS

2. Core Modules (Epics):

Epic	Description
User Management	Signup/Login, Role-based access
Vehicle Management	Add/Edit vehicles for each customer
Service Booking	Book service, assign mechanic, update status
Invoice & Payment	Generate invoice, mark payment
Inventory Management	Spare parts, availability tracking

3. User Stories by Sprint:

Sprint	Focus	Story Points
Sprint 1	User & Vehicle Management	8 SP
Sprint 2	Booking & Job Status Flow	10 SP
Sprint 3	Invoice + Inventory System	10 SP
Sprint 4	Admin Dashboard & Alerts	10 SP

Sprint 5	Testing + Deployment	11 SP
----------	----------------------	-------

4. Story Point Estimation (Effort Guide):

- 1 SP – Very Easy (UI Form)
- 2 SP – Easy (CRUD Operation)
- 3 SP – Moderate (DB + Logic)
- 5 SP – Complex (Integrations)

5. Velocity & Timeline:

- Total SP: 49
- Team Velocity: 10 SP/Sprint
- Total Duration: 5 Weeks

6. Weekly Milestones:

- Week 1 – Auth + Vehicle Modules Ready
- Week 2 – Booking Flow Completed
- Week 3 – Billing + Inventory Added
- Week 4 – Admin Tools + Notifications Done
- Week 5 – App Tested + Deployed Online

7. Final Outcomes:

- Web App with Login, Vehicle Mgmt, Booking
- Role-based UI (Admin, Mechanic, Customer)
- Invoicing, Spare Parts Mgmt
- Dashboard with Reports
- Hosted Live App (Heroku/Render)

6. FUNCTIONAL AND PERFORMANCE TESTING:

6.1 Performance Testing:

Project Files:

Project files are essential documents created and maintained during the execution of the Garage Management System to ensure smooth development, testing, deployment, and delivery.

Key Project Files Include:

- **Project Plan** – Outlines timeline, team roles, and milestones.
- **Requirements Document** – Captures functional and non-functional system needs.
- **Use Case/User Stories** – Describes how users interact with the system.
- **Data Model & ER Diagram** – Shows object relationships (Customer → Appointment → Billing).
- **Configuration Sheets** – Details all objects, fields, and validations.
- **Apex/Flow Logic Docs** – Explains automation rules and triggers.
- **Test Plan & Cases** – Ensures all features are working as expected.
- **Deployment Checklist** – Guides pre/post deployment tasks.
- **User Manual** – Helps users understand system usage.
- **Reports & Dashboards File** – Describes analytics and metrics used.
- **Client Sign-Off Form** – Confirms approval from stakeholders.

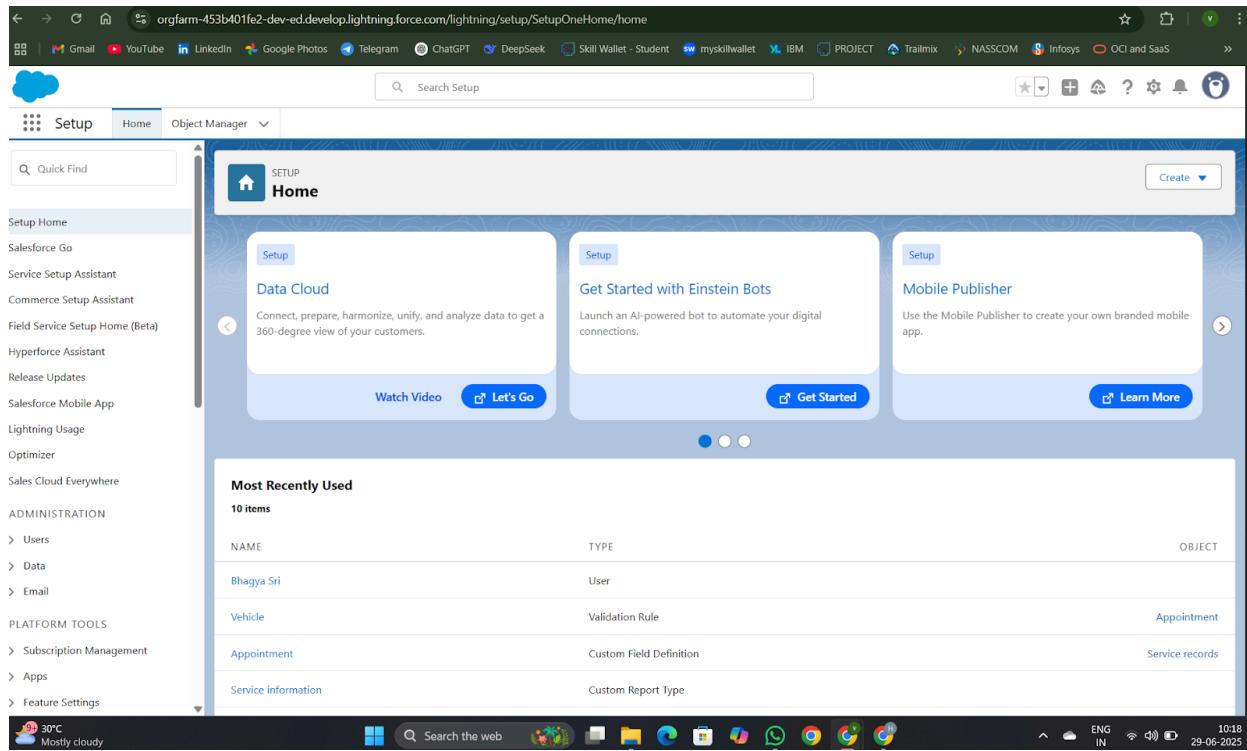
Project Overview:

The Garage Management System is a Salesforce-based application designed to streamline operations in a vehicle repair/maintenance garage. It manages customer records, service appointments, billing, feedback, and reporting — improving customer experience and business efficiency.

Developer Account Setup:

Creating Developer Org

- Go to: <https://developer.salesforce.com/signup>
- Fill in details like Name, Email, Role (Developer), etc.
- Create a username (e.g., yourname@yourcollege.com)
- Activate via email and set password.



Object Creation:

Objects are used to store data like customers, appointments, services, etc.

Custom Objects Created:

- Customer Details
- Appointments
- Service Records
- Billing Details & Feedback

Each object has its own **Record Name**, **Data Type**, and is enabled for **Reports**, **Search**, and **Field Tracking**.

The screenshot shows the Salesforce Setup interface with the following details:

- Setup > OBJECT MANAGER**
- Customer Details** object selected.
- Details** tab selected in the sidebar.
- Description**: API Name - Customer_Details_c, Custom - ✓, Singular Label - Customer Details, Plural Label - Customer Details.
- Enable Reports**: ✓
- Track Activities**: ✓
- Track Field History**: ✓
- Deployment Status**: Deployed
- Help Settings**: Standard salesforce.com Help Window
- Buttons, Links, and Actions** section visible in the sidebar.

Tabs:

Tabs are created for UI access to each object:

- Go to Setup → Tabs → Create New Tab
- Tabs created for: Customer Details, Appointments, Service Records, Billing Details & Feedback

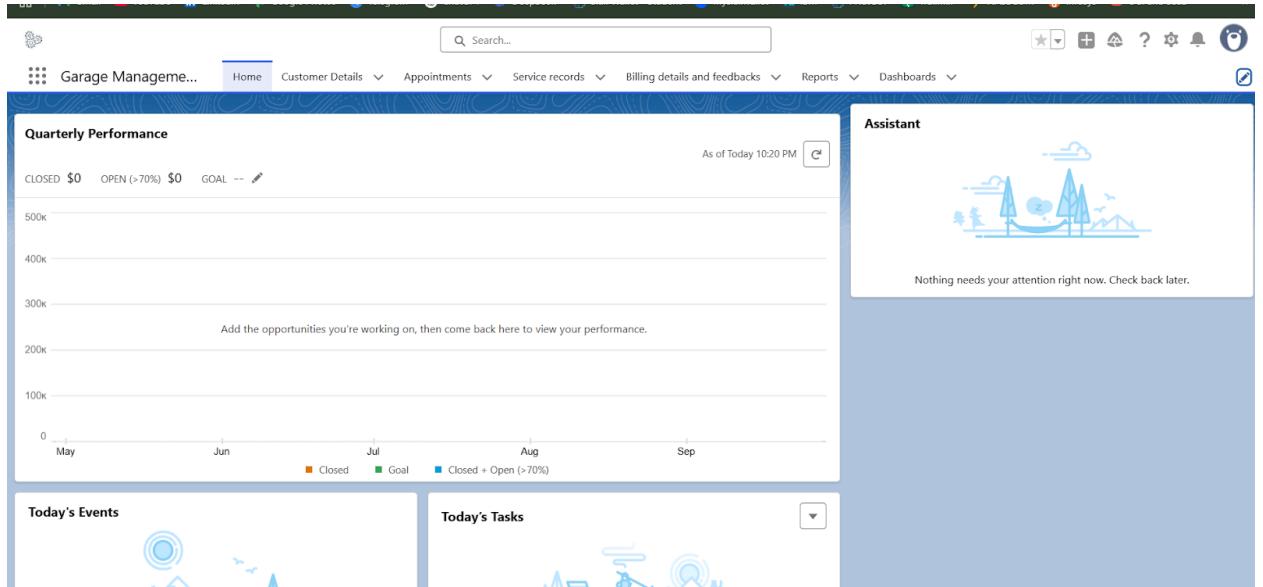
The screenshot shows the Salesforce Setup interface with the following details:

- Setup > TABS**
- Custom Tabs** section:
 - You can create new custom tabs to extend Salesforce functionality or to build new application functionality.
 - Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.
- Custom Object Tabs** table:

Action	Label	Tab Style	Description
Edit Del	Appointments	Alarm clock	
Edit Del	Billing details and feedbacks	Balls	
Edit Del	Customer Details	Airplane	
Edit Del	Service records	Apple	
- Web Tabs** section: No Web Tabs have been defined.
- Visualforce Tabs** section: No Visualforce Tabs have been defined.
- Lightning Component Tabs** section: No Lightning component tabs have been defined.

Lightning App Setup:

- Created a custom app: **Garage Management Application**
- Added navigation items: all custom objects, Reports, Dashboards
- Set up default user profile access.



Field Creation:

Each object is enriched with standard and custom fields.

Example Fields:

- **Customer Details:** Phone number, Gmail

Customer Details

Fields & Relationships
6 Items, Sorted by Field Label

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		✓

- **Appointments:** Vehicle number plate, Appointment date, Service checkboxes (Maintenance, Repairs, Replacements), Service Amount

Appointment

Fields & Relationships
11 Items, Sorted by Field Label

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 Details	Customer_Details__c	Lookup(Customer Details)		✓
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		
Service Amount	Service_Amount__c	Currency(18, 0)		
Vehicle number plate	Vehicle_number_plate__c	Text(10) (Unique Case Insensitive)		✓

- **Service Records:** Service Status (Picklist), Quality Check (Checkbox)

Service records

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment	Appointment_c	Lookup(Appointment)		✓
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	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		

- **Billing & Feedback:** Payment Status, Rating, Payment Paid

Billing details and feedback

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Billing details and feedback Name	Name	Auto Number		✓
Created By	CreatedBy	Lookup(User)		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment Paid	Payment_Paid_c	Currency(18, 0)		
Payment Status	Payment_Status_c	Picklist		
Rating for service	Rating_for_service_c	Text(1)		
Service records	Service_records_c	Lookup(Service records)		✓

Lookup Relationships

Enables linking between objects:

- Appointment → Customer Details
- Service Records → Appointment
- Billing Details → Service Records

Also includes **lookup filters** (e.g., appointment date < created date).

Validation Rules:

Ensures data correctness:

- Vehicle number plate follows a specific format
- Rating for service must be between 1 to 5

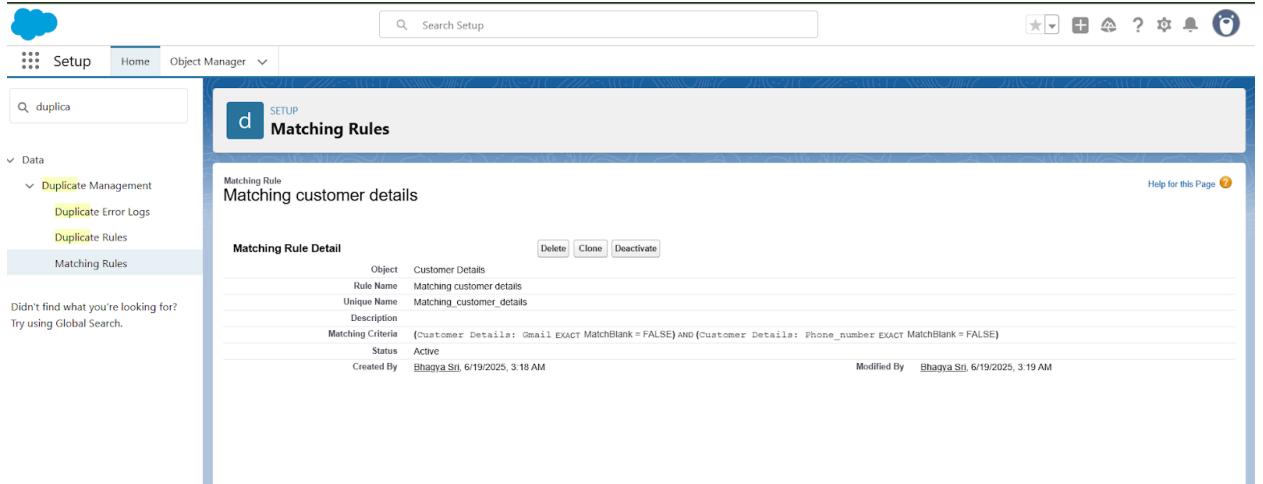
The screenshot shows the Salesforce Object Manager interface for the 'Appointment' object. The left sidebar lists various setup options like Details, Fields & Relationships, Page Layouts, etc. The main area is titled 'Validation Rules' and shows one item: 'Vehicle' with an error message 'Please enter valid number'. The table has columns for Rule Name, Error Location, Error Message, Active status, and Modified By.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter valid number	✓	Bhagya Sri, 6/19/2025, 3:08 AM

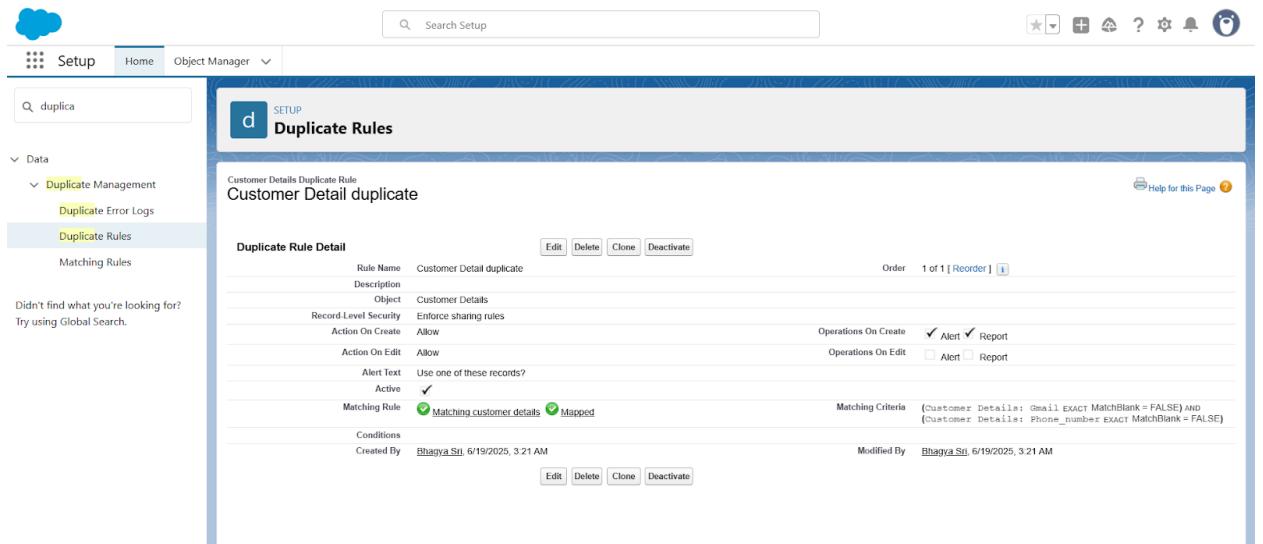
Duplicate Rules:

Ensures no duplicate customer entries:

- Created **Matching Rule** using Gmail and Phone Number



- Created **Duplicate Rule** using the matching rule for Customer Details object



Profiles:

Profiles control user permissions:

- **Manager Profile:** Full access to all objects

The screenshot shows the Salesforce Setup interface with the 'Profiles' page selected. The 'Manager' profile is displayed. The profile details include:

- Name:** Manager
- User License:** Salesforce
- Description:** Created by Bhagya Sri on 6/19/2025, 3:23 AM
- Modified By:** Bhagya Sri on 6/26/2025, 10:10 AM

Page Layouts

Object	Layout Type	Assignment	Location Group	Assignment
Global	Global Layout	[View Assignment]	Location Group	Location Group Layout [View Assignment]
Email Application	Not Assigned	[View Assignment]	Location Group Assignment	Location Group Assignment Layout [View Assignment]
Home Page Layout	Home Page Default	[View Assignment]	Macro	Macro Layout [View Assignment]
Account	Account Layout	[View Assignment]	Object Milestone	Object Milestone Layout [View Assignment]
Alternative Payment Method	Alternative Payment Method Layout	[View Assignment]	Operating Hours	Operating Hours Layout [View Assignment]
Appointment Invitation	Appointment Invitation Layout	[View Assignment]	Opportunity	Opportunity Layout [View Assignment]
Asset	Asset Layout	[View Assignment]	Opportunity Product	Opportunity Product Layout [View Assignment]

- **SalesPerson Profile:** Limited access for appointment and customer handling

The screenshot shows the Salesforce Setup interface with the 'Profiles' page selected. The 'sales person' profile is displayed. The profile details include:

- Name:** sales person
- User License:** Salesforce Platform
- Description:** Created by Bhagya Sri on 6/19/2025, 3:32 AM
- Modified By:** Bhagya Sri on 6/26/2025, 10:10 AM

Page Layouts

Object	Layout Type	Assignment	Location	Assignment	Lead	Assignment
Global	Global Layout	[View Assignment]	Location	Location Layout [View Assignment]	Lead Layout [View Assignment]	
Email Application	Not Assigned	[View Assignment]	Location Group	Location Group Layout [View Assignment]	Location Group Assignment Layout [View Assignment]	
Home Page Layout	Home Page Default	[View Assignment]	Object Milestone	Object Milestone Layout [View Assignment]	Object Milestone Layout [View Assignment]	
Account	Account Layout	[View Assignment]	Operating Hours	Operating Hours Layout [View Assignment]	Operating Hours Layout [View Assignment]	
Alternative Payment Method	Alternative Payment Method Layout	[View Assignment]	Order	Order Layout [View Assignment]	Order Layout [View Assignment]	
Appointment Invitation	Appointment Invitation Layout	[View Assignment]				
Asset	Asset Layout	[View Assignment]				

Custom profiles cloned and modified from:

- Standard User (Manager)
- Salesforce Platform User (Sales Person)

Roles and Role Hierarchy:

Defines data visibility levels:

- **Manager**
- **Sales Person under Manager**

The screenshot shows the Salesforce Setup interface with the 'Roles' page selected. The left sidebar shows navigation links for Users, Feature Settings, Sales, Service, and Case Teams. The main content area displays a hierarchical list of roles under the 'ideal institute technology' organization. The hierarchy includes:

- Add Role
- CEO (Edit | Del | Assign)
 - Add Role
- CFO (Edit | Del | Assign)
 - Add Role
- COO (Edit | Del | Assign)
 - Add Role
- Manager (Edit | Del | Assign)
 - Add Role
- sales person** (Edit | Del | Assign)
 - Add Role
- SVP, Customer Service & Support (Edit | Del | Assign)
 - Add Role
- Customer Support, International (Edit | Del | Assign)
 - Add Role
- Customer Support, North America (Edit | Del | Assign)
 - Add Role
- Installation & Repair Services (Edit | Del | Assign)
 - Add Role
- SVP, Human Resources (Edit | Del | Assign)
 - Add Role
- SVP, Sales & Marketing (Edit | Del | Assign)
 - Add Role

Users:

Created user accounts:

- **Manager** role & profile
- **Sales Person** role & profile

Public Groups:

Created a group:

- **Sales Team** includes all SalesPerson roles

The screenshot shows the Salesforce Setup interface. On the left, the navigation pane includes 'Setup' (selected), 'Home', and 'Object Manager'. Under 'Users', 'Public Groups' is selected. Other sections like 'Feature Settings' and 'Company Settings' are also visible. The main content area is titled 'Public Groups' and shows a group named 'sales team'. The group details include a Label ('sales team'), Group Name ('sales_team'), and 'Grant Access Using Hierarchies' checked. It was created by 'Bhagya Sri' on 6/19/2025, 4:04 AM, and modified by the same user on the same date. Below the details, a table lists a member: 'sales person' (Type: Role). A 'View All Users' link is also present.

Sharing Settings:

Customized object sharing:

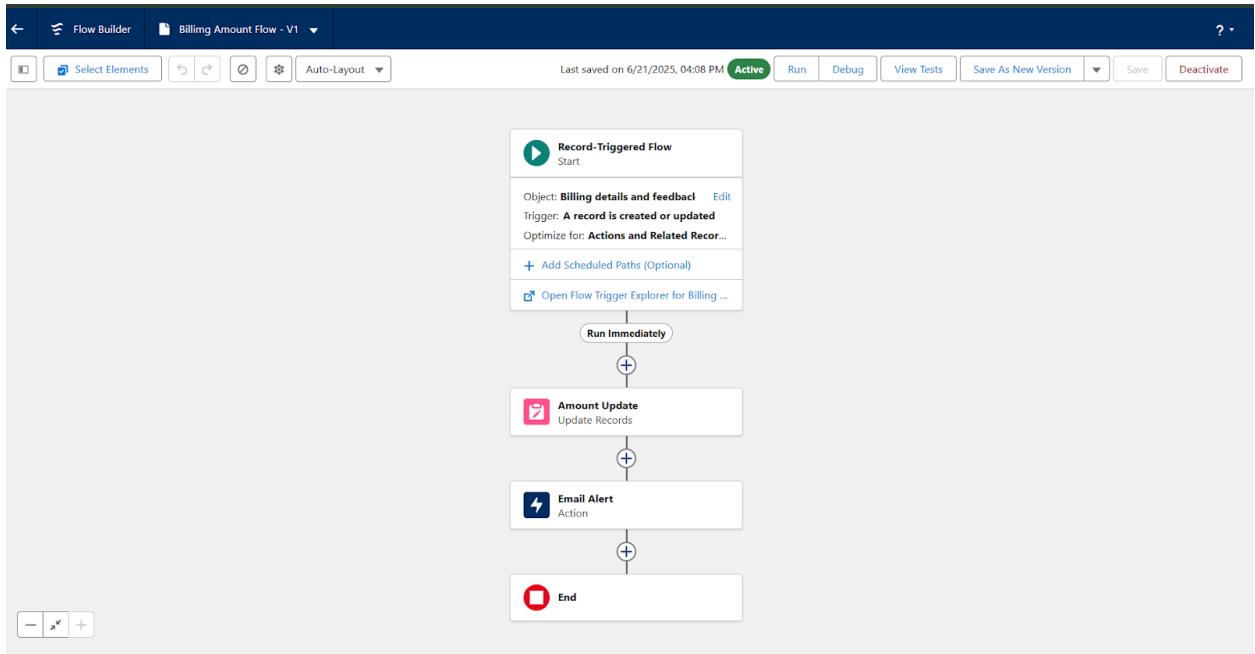
- **Service Records** set to Private (OWD)
- Created Sharing Rule to allow Sales Person to share data with Manager

The screenshot shows the 'Customer Details Sharing Rules' page. It has two main sections: 'Customer Details Sharing Rules' and 'Service records Sharing Rules'. The 'Customer Details Sharing Rules' section shows a message 'No sharing rules specified'. The 'Service records Sharing Rules' section shows a single rule: 'Owner in Role: sales person' is shared with 'Role: Manager' at the 'ReadWrite' access level. Buttons for 'New', 'Recalculate', and 'Help' are also visible.

Flows:

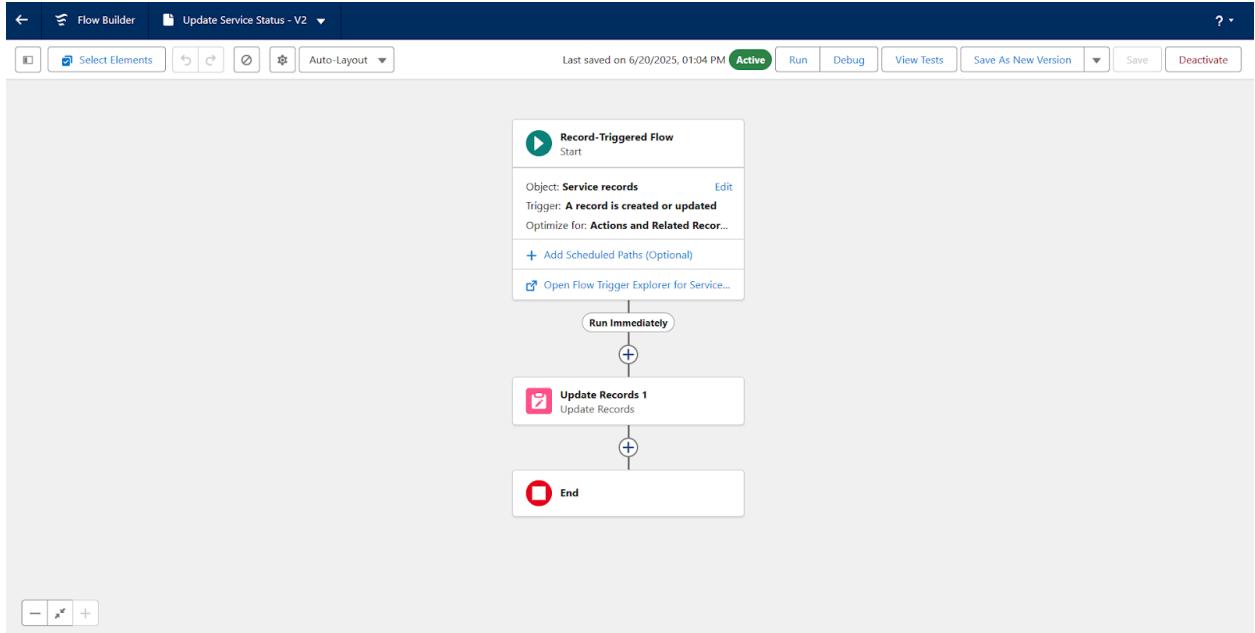
Flow 1: Payment Confirmation Email

- Trigger: Billing record updated with status = “Completed”
- Action: Send email with service amount and thank-you message



Flow 2: Auto Update Service Status

- Trigger: Quality Check marked true
- Action: Auto-update service status to “Completed”



Apex Trigger & Handler:

Use Case: Calculate Service Amount

- Trigger on **Appointment** object
- Handler calculates amount based on checkboxes:
 - Maintenance + Repairs = ₹5000
 - All 3 selected = ₹10,000
 - Individual selections calculated accordingly

The screenshot shows the Salesforce Setup interface under the Apex Triggers section. The left sidebar has a search bar with 'apex' and links for Email, Custom Code, Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, and Apex Triggers (which is selected). The main content area is titled 'Apex Triggers' with a sub-header 'Developer Console'. It displays a table with one row for the 'AmountDistribution' trigger. The table columns are Action, Name, Namespace Prefix, sObject Type, Api Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The trigger details are: Name: AmountDistribution, Namespace Prefix: null, sObject Type: Appointment, Api Version: 64.0, Status: Active, Size Without Comments: 211, Last Modified By: Bhagya Sri, and Has Trace Flags: false.

Action	Name	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del	AmountDistribution		Appointment	64.0	Active	211	Bhagya Sri, 6/20/2025, 12:04 AM	

Reports:

- Created a custom **Report Type** joining:
 - Customer → Appointment → Service Records → Billing
- Report: **New Service Information Report**
 - Fields: Customer Name, Appointment Date, Service Status, Payment Paid, Rating
 - Grouped & Charted (Line Chart)

The screenshot shows the Garage Management system Reports page. The top navigation bar includes links for Home, Customer Details, Appointments, Service records, Billing details and feedbacks, Reports (which is selected), and Dashboards. The left sidebar has sections for Reports, Recent (2 items), Created by Me, Private Reports, Public Reports, All Reports, and FOLDERS. The main content area shows a table of recent reports. The columns are Report Name, Description, Folder, Created By, Created On, and Subscribed. There are two entries: 'New Service information Report' created by Bhagya Sri on 6/27/2025 at 3:50 AM, and another 'New Service information Report' created by Bhagya Sri on 6/20/2025 at 1:00 AM.

Report Name	Description	Folder	Created By	Created On	Subscribed
New Service information Report	Garage Management Folder	Bhagya Sri	6/27/2025, 3:50 AM		
New Service information Report	Garage Management Folder	Bhagya Sri	6/20/2025, 1:00 AM		

Dashboards:

- Created **Dashboard Folder**: Service Rating Dashboard
- Used report above to build dashboard component (Line Chart)
- Subscribed to dashboard weekly (e.g., every Monday)

Dashboard...	Description	Folder	Created By	C..	Subscribed
Customer review	Service Rating dashboard	Bhagya Sri	6/20/22 025, 2:57 AM	✓	<input type="checkbox"/>
Customer review	Service Rating dashboard	Bhagya Sri	6/21/22 025, 2:10 AM	✓	<input type="checkbox"/>
Customer review	Service Rating dashboard	Bhagya Sri	6/20/22 025, 12:46 AM		<input type="checkbox"/>

Testing: Record Creation:

Created records in:

- Customer Details
- Appointments (with vehicle number format)
- Service Records
- Billing and Feedback

Tested automation:

- Auto-updated fields
- Triggered flows and emails

Outcome:

- Seamless end-to-end garage workflow in Salesforce
- Automated billing, updates, and communication
- Enhanced transparency, efficiency, and user satisfaction

Datasets:

A dataset is a structured collection of related data that is organized in rows and columns for easy processing, analysis, and testing. In the context of the Garage Management System (GMS), datasets represent the sample records created for customers, appointments, services, billing, and feedback, which are stored in different Salesforce objects.

Purpose of Datasets in This Project:

- To test how the system handles and displays real-time data.
- To validate automation like triggers, flows, and email alerts.
- To generate meaningful reports and dashboards.
- To simulate real-life user interactions and service workflows.

Customer Details Dataset:

The screenshot shows a Salesforce Lightning component for the 'Customer Details' dataset. The page title is 'Customer Details'. The top navigation bar includes links for Home, Customer Details, Appointments, Service records, Billing details and feedbacks, Reports, and Dashboards. A search bar at the top right contains the placeholder 'Search...'. Below the navigation is a toolbar with buttons for New, Import, Change Owner, and Assign Label. A 'Recently Viewed' section is visible on the left. The main content area displays a list of 10 customer names, each preceded by a checkbox. The names listed are: 1. Sai Neeraj, 2. Ram, 3. Sita, 4. Shailu, 5. Dhruv, 6. Vamsi Anand, 7. Rakesh, 8. Rupas, 9. Vijaya, and 10. Ramya. The URL in the browser address bar is https://orgfarm-453b401fe2-dev-ed.develop.lightning.force.com/lightning/o/Customer_Details__c/home.

Appointment Dataset:

The screenshot shows a Salesforce Lightning component for the 'Appointments' dataset. The page title is 'Appointments'. The top navigation bar includes links for Home, Customer Details, Appointments, Service records, Billing details and feedbacks, Reports, and Dashboards. A search bar at the top right contains the placeholder 'Search...'. Below the navigation is a toolbar with buttons for New, Import, Change Owner, and Assign Label. A 'Recently Viewed' section is visible on the left. The main content area displays a list of 11 appointment names, each preceded by a checkbox. The names listed are: 1. app-022, 2. app-021, 3. app-020, 4. app-019, 5. app-012, 6. app-014, 7. app-015, 8. app-016, 9. app-018, 10. app-011, and 11. app-013. The URL in the browser address bar is https://orgfarm-453b401fe2-dev-ed.develop.lightning.force.com/lightning/o/Appointments__c/home.

Service Records Dataset:

The screenshot shows a web-based application interface for a Garage Management system. The top navigation bar includes links for Home, Customer Details, Appointments, Service records (which is the active tab), Billing details and feedbacks, Reports, and Dashboards. A search bar labeled 'Search...' is located at the top right. Below the navigation is a toolbar with icons for New, Import, Change Owner, and Assign Label. A 'Recently Viewed' section displays a list of 10 service records, each with a checkbox and a dropdown menu. The list is as follows:

Index	Service Record Name
1	ser-012
2	ser-015
3	ser-013
4	ser-006
5	ser-014
6	ser-011
7	ser-010
8	ser-009
9	ser-008
10	ser-007

Billing & Feedback Dataset:

The screenshot shows the same web-based application interface for a Garage Management system. The top navigation bar and search bar are identical to the previous screenshot. The 'Billing details and feedbacks' tab is active. A 'Recently Viewed' section displays a list of 8 billing and feedback items, each with a checkbox and a dropdown menu. The list is as follows:

Index	Billing and Feedback Name
1	bill-008
2	bill-007
3	bill-006
4	bill-005
5	bill-004
6	bill-003
7	bill-002
8	bill-001

Summary:

A dataset is a group of sample entries created to represent actual records used in the Garage Management System, enabling testing, validation, and presentation of the project's functionality.

Model Performance Testing:

Performance Testing:

Performance Testing is a type of software testing that evaluates how a system performs under specific workloads. It checks the system's speed, responsiveness, stability, and scalability to ensure optimal user experience and operational efficiency.

Purpose:

- To ensure the application can **handle expected user load**
- To identify and eliminate **performance bottlenecks**
- To validate **system behavior under stress** (high data/traffic)
- To confirm that **response times** meet business requirements

Types of Performance Testing:

Type	Description
Load Testing	Checks how the system handles expected number of users and operations.
Stress Testing	Determines the system's breaking point by applying extreme load.
Spike Testing	Observes behavior when load suddenly increases or decreases.
Scalability Testing	Checks system's ability to scale up or down as users increase
Endurance Testing	Evaluates system performance over a long duration (e.g., 12–24 hours).

Tools Commonly Used for Performance Testing:

Tool	Usage Area
Salesforce Dev Console	For logs and timing apex/flows
Apache JMeter	Web app load testing
BlazeMeter	Cloud-based performance testing
LoadRunner	Enterprise-grade performance tool
Google Lighthouse	Front-end performance

Here are the Parameters to evaluate the performance of the project:

1.Apex Triggers:

In the **Garage Management System (GMS)** project, **Apex Triggers** are used to automate business logic in Salesforce before or after specific data operations occur on records.

Purpose:

- Automatically **calculate the service amount** based on selected services (Maintenance, Repairs, Replacement).
- Ensure **real-time updates** without manual intervention.

The screenshot shows the Salesforce Setup interface with the search bar set to 'apex'. Under the 'Apex Classes' section, 'Apex Triggers' is selected. The main content area is titled 'Apex Triggers' and contains a message stating 'Percent of Apex Used: 0.02%' and '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.' Below this is a table with one row, showing the trigger details:

Action	Name	Namespace Prefix	sObject Type	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags
Edit Del	AmountDistribution		Appointment	64.0	Active	211	Rbagya Sr.	6/20/2025, 12:04 AM

2. Flows :

In the **Garage Management System (GMS)** project, **Flows** in Salesforce are used to automate repetitive tasks and improve process efficiency **without writing code**.

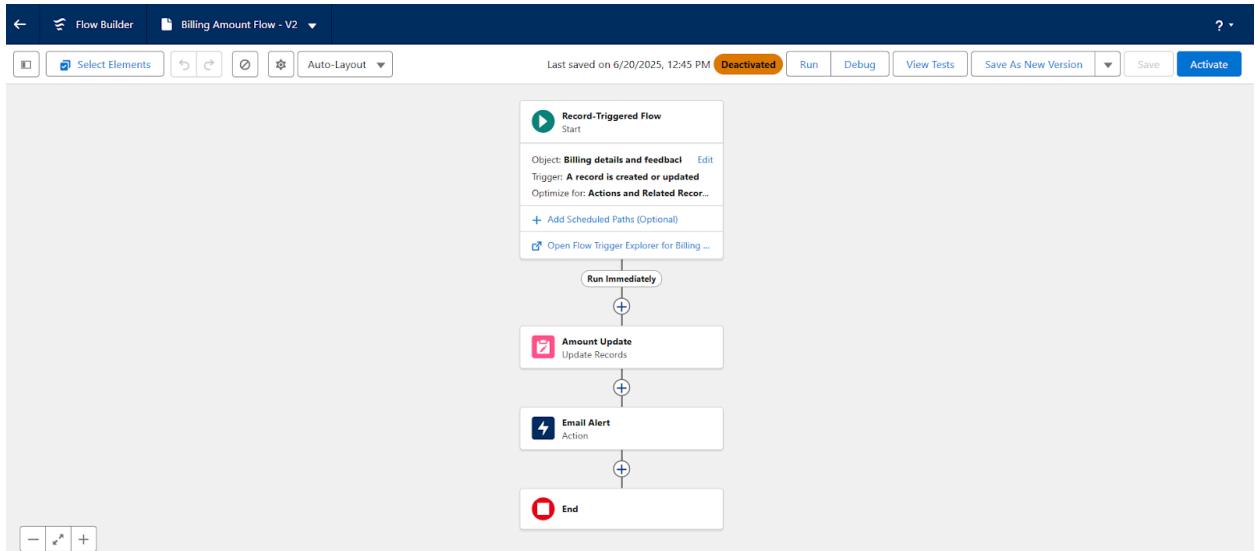
What Are Flows?

Flows are visual workflows in Salesforce that automate data updates, send notifications, and handle complex logic based on user-defined conditions.

Flows Used in This Project:

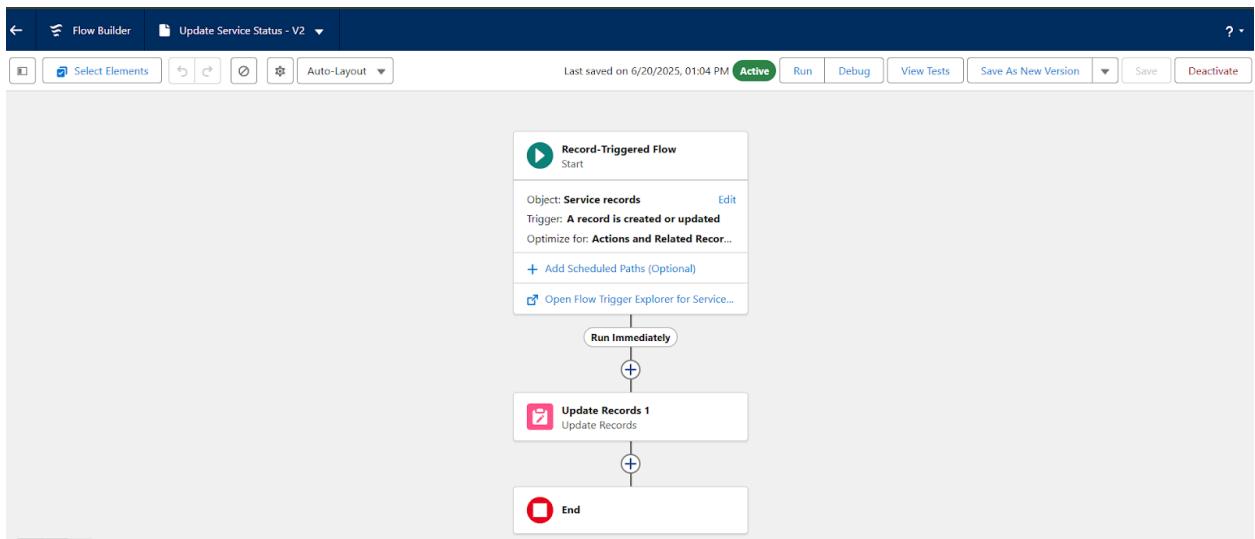
1. Payment Confirmation Flow

- **Type:** Record-Triggered Flow
- **Object:** `Billing_details_and_feedback__c`
- **Trigger:** When a record is **created or updated**
- **Logic:**
 - If **Payment Status = Completed** →
 - Auto-update **Payment Paid** field from linked service amount
 - Send a **thank-you email** to the customer



2. Service Status Update Flow

- **Type:** Record-Triggered Flow
- **Object:** `Service_records__c`
- **Trigger:** When **Quality Check Status = True**
- **Logic:**
 - Automatically sets **Service Status = Completed**

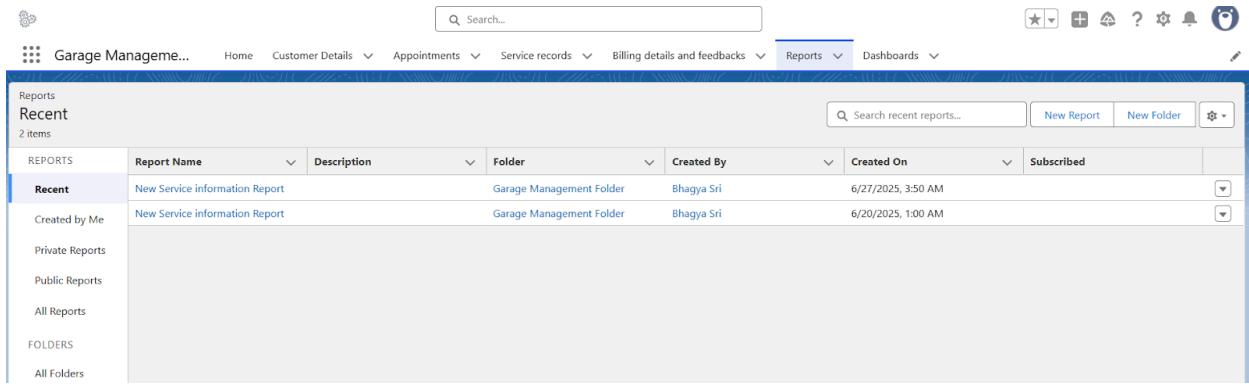


3. Reports & Dashboards :

In the **Garage Management System (GMS)** project, **Reports and Dashboards** are used to monitor service performance, customer feedback, and billing information in a **visual and analytical** format.

Reports:

Custom Report Created



The screenshot shows the 'Reports' section of the Garage Management System. On the left, there's a sidebar with categories: 'Recent' (2 items), 'REPORTS' (Recent), 'Created by Me', 'Private Reports', 'Public Reports', and 'All Reports'. Below that is a 'FOLDERS' section with 'All Folders'. The main area has a table titled 'Recent' with columns: Report Name, Description, Folder, Created By, Created On, and Subscribed. There are two entries: 'New Service information Report' created by Bhagya Sri on 6/27/2025, 3:50 AM, and another 'New Service information Report' created by Bhagya Sri on 6/20/2025, 1:00 AM. A search bar at the top and bottom, along with various icons for search, new report, and folder creation, are also visible.

Report Name	Description	Folder	Created By	Created On	Subscribed
New Service information Report		Garage Management Folder	Bhagya Sri	6/27/2025, 3:50 AM	
New Service information Report		Garage Management Folder	Bhagya Sri	6/20/2025, 1:00 AM	

- **Report Type:** Custom Report Type based on:
 - Customer Details
 - Appointments
 - Service Records
 - Billing & Feedback
- **Report Name:** New Service Information Report
- **Displayed Fields:**
 - Customer Name
 - Appointment Date
 - Service Status
 - Payment Paid
 - Rating for Service
 - Payment Status
- **Groupings:**
 - Grouped by Rating for Service
 - Grouped by Payment Status
- **Chart Used:** Line Chart

Dashboards:

Dashboard Created:

The screenshot shows the 'Garage Management System' interface with the 'Dashboards' tab selected. On the left, a sidebar lists categories like 'DASHBOARDS', 'Recent', 'Created by Me', 'Private Dashboards', 'All Dashboards', 'FOLDERS', 'All Folders', 'Created by Me', 'Shared with Me', and 'FAVORITES'. The main area displays a table titled 'Recent' with three rows of dashboard entries. Each row includes columns for 'Dashboard...', 'Description', 'Folder', 'Created By', 'C..', and 'Subscribed'. The first two rows have a 'Subscribed' status indicated by a checkmark and a dropdown menu icon. The third row has a dropdown menu icon only.

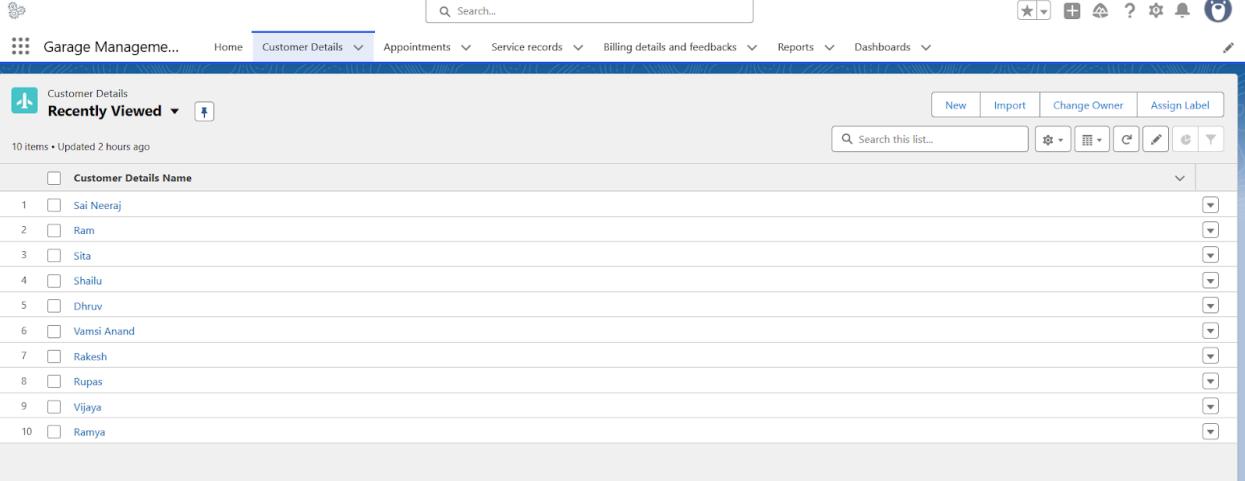
	Dashboard...	Description	Folder	Created By	C..	Subscribed
Customer review	Service Rating dashboard	Bhagya Sri	6/20/2 025, 2:57 AM	✓	<input type="button"/>	
Customer review	Service Rating dashboard	Bhagya Sri	6/21/2 025, 2:10 AM	✓	<input type="button"/>	
Customer review	Service Rating dashboard	Bhagya Sri	6/20/2 025, 12:46 AM		<input type="button"/>	

- **Name:** Service Rating Dashboard
- **Folder:** Service Rating Dashboard
- **Component Used:** Line Chart from above report
- **Insights Visualized:**
 - Ratings Distribution
 - Completed Payments vs Pending
 - Overall service performance
- **Automation:** Dashboard is **subscribed weekly**, delivering insights to the **Manager** role.

Record Insert/Update :

In the **Garage Management System (GMS)** project, **Record Insert/Update** refers to the process of adding new data or modifying existing records across various custom Salesforce objects like **Customer Details, Appointments, Service Records, and Billing & Feedback**.

Insert Process (Creating Records):



The screenshot shows the Salesforce interface for the 'Customer Details' object. At the top, there's a navigation bar with links for Home, Customer Details, Appointments, Service records, Billing details and feedbacks, Reports, and Dashboards. Below the navigation is a search bar labeled 'Search...'. On the right side of the header are various icons for star, plus, question mark, etc. The main area is titled 'Customer Details' and shows a list titled 'Recently Viewed'. It displays 10 items, all updated 2 hours ago. The list includes columns for 'Customer Details Name' and a checkbox column. The names listed are: 1. Sai Neeraj, 2. Ram, 3. Sita, 4. Shailu, 5. Dhruv, 6. Vamsi Anand, 7. Rakesh, 8. Rupas, 9. Vijaya, and 10. Ramya. Each row has a dropdown arrow icon on the far right.

- **Objects Involved:**

- Customer Details
- Appointment
- Service Records
- Billing & Feedback

- **Steps:**

- Navigate to the tab (e.g., Customer Details).
- Click on **New**, fill in required fields (e.g., name, phone, Gmail).
- Save the record – it gets stored in Salesforce.

- **Automation Triggered:**

- Apex triggers calculate **service amount** during appointment insertion.
- Flows auto-update fields like **Service Status** or **Payment Paid** after insert/update.

Update Process (Modifying Records):

- Records can be edited directly from their respective object tabs.
- Example:
 - Marking **Quality Check** as will **automatically update** **Service Status** to **Completed** (via Flow).
 - Updating **Payment Status** to **Completed** triggers **emailalert** and updates **Payment Paid**.

Validations During Insert/Update:

- **Vehicle Number** format validation (e.g., MH12AB1234)
- **Rating** must be between 1 and 5
- Duplicate records blocked (e.g., same email & phone)

The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Search Setup, Home, Object Manager.
- Breadcrumbs:** SETUP > OBJECT MANAGER > Appointment.
- Left Sidebar:** 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.
- Table:** Validation Rules (1 items, Sorted by Rule Name).

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter valid number	✓	Bhagya Sri, 6/19/2025, 3:08 AM

Purpose:

- Keeps the system up-to-date with real-time service activity.
- Enables automation like triggers, flows, and reporting.
- Ensures **data consistency** and **customer satisfaction**.

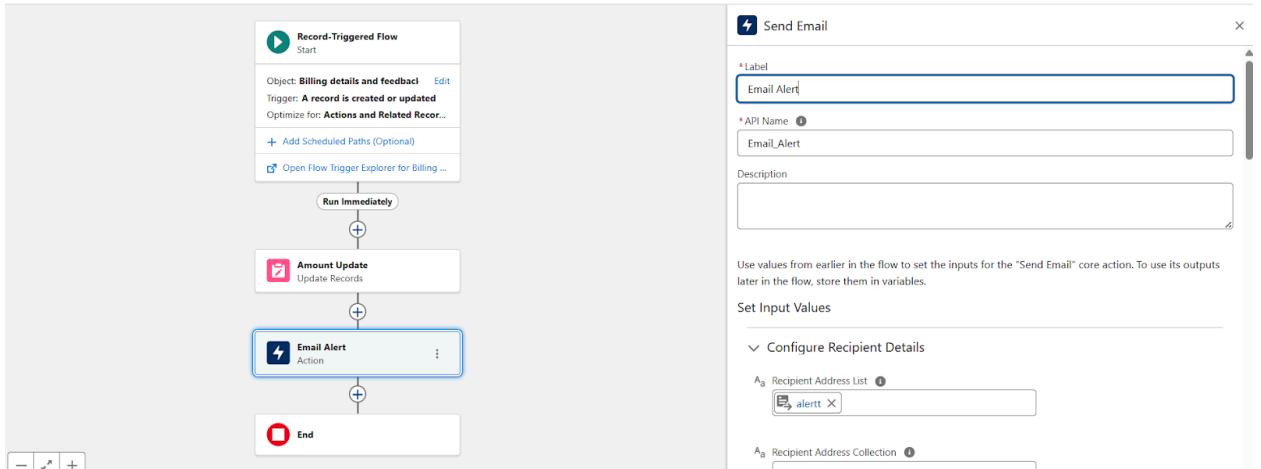
Mail Alerts:

In the **Garage Management System (GMS)** project, **Mail Alerts** are used to automatically send emails to customers after certain actions are completed, enhancing communication and customer satisfaction.

Purpose of Mail Alerts:

- To thank customers after payment.
- To confirm service completion.

- To provide a **personalized communication experience** using automated flows.



How Mail Alerts Work:

- **Tool Used:** Salesforce Flow Builder
- **Trigger:** When a **Billing & Feedback** record is created or updated.
- **Condition:** **Payment Status** is set to **Completed**.
- **Action:**
 - Flow fetches the customer's **name**, **email**, and **payment amount**.
 - Sends an **email alert** with a thank-you message and payment details.

7. RESULTS:

7.1 Output Screenshots:

Final Outputs:

1.Customer Management Output:

- New customer records are created with validated email and phone formats.
- Duplicate entries are prevented using duplicate rules.
- Customer details are accessible in a tabbed view in the Lightning app.

The screenshot shows a list of customer details. At the top, there is a search bar with the placeholder "Search..." and a toolbar with icons for star, plus, question mark, etc. Below the toolbar, a navigation bar includes "Home", "Customer Details", "Appointments", "Service records", "Billing details and feedbacks", "Reports", and "Dashboards". A "Recently Viewed" dropdown menu is open, showing "Customer Details Name". The main list displays 10 items, each with a checkbox and a name: 1. Sai Neeraj, 2. Ram, 3. Sita, 4. Shailu, 5. Dhruv, 6. Vamsi Anand, 7. Rakesh, 8. Rupas, 9. Vijaya, 10. Ramya. There are edit and delete icons next to each row.

2. Appointment Booking Output:

- Appointments are successfully scheduled.
- Vehicle number is validated (e.g., format: KA05AB1234).
- Service amount is automatically calculated based on selected checkboxes using Apex Trigger.

The screenshot shows a list of appointments. At the top, there is a search bar with the placeholder "Search..." and a toolbar with icons for star, plus, question mark, etc. Below the toolbar, a navigation bar includes "Home", "Customer Details", "Appointments", "Service records", "Billing details and feedbacks", "Reports", and "Dashboards". A "Recently Viewed" dropdown menu is open, showing "Appointments Name". The main list displays 11 items, each with a checkbox and a name: 1. app-022, 2. app-021, 3. app-020, 4. app-019, 5. app-012, 6. app-014, 7. app-015, 8. app-016, 9. app-018, 10. app-011, 11. app-013. There are edit and delete icons next to each row.

3.Service Record Output:

- New service records created with lookup to the corresponding appointment.
- Status changes automatically to “Completed” after quality check is enabled (Flow logic).
- Validation ensures service appointment date < record creation date.

The screenshot shows a list view of service records in a Microsoft Dynamics 365 application. The top navigation bar includes links for Home, Customer Details, Appointments, Service records (which is the active tab), Billing details and feedbacks, Reports, and Dashboards. A search bar at the top right contains the placeholder "Search...". Below the navigation is a toolbar with icons for New, Import, Change Owner, and Assign Label. A "Recently Viewed" section shows 10 items, all labeled "ser-012". A detailed list follows, numbered 1 to 10, each with a checkbox and the value "ser-012". The interface uses a clean, modern design with blue and white colors.

Index	Service records Name
1	ser-012
2	ser-015
3	ser-013
4	ser-006
5	ser-014
6	ser-011
7	ser-010
8	ser-009
9	ser-008
10	ser-007

4.Billing & Feedback Output:

- Payment amount is auto-filled if service is marked as completed.
- Rating validated (only 1 to 5 allowed).
- Automated thank-you email sent on payment completion (via Flow).

Billing details and feedbacks

Recently Viewed

8 items • Updated a few seconds ago

	Billing details and feedback Name
1	bill-008
2	bill-007
3	bill-006
4	bill-005
5	bill-004
6	bill-003
7	bill-002
8	bill-001

5. Reports Output:

- Custom report “Service Information Report” generated using related objects.
- Shows appointment dates, service statuses, payment amounts, and customer ratings.
- Grouped by “Rating for Service” and “Payment Status”.

Garage Management...

Reports

All Folders

8 items

REPORTS	Name	Created By	Created On	Last Modified By	Last Modified Date
Recent	Einstein Bot Reports	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
Created by Me	Einstein Bot Reports Spring '23	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
Private Reports	Einstein Bot Reports Summer '23	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
Public Reports	Einstein Bot Reports Summer '22	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
All Reports	Einstein Bot Reports Winter '23	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
FOLDERS	Enablement Dashboard Reports Spring '24	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
All Folders	Enablement Dashboard Reports Summer '24	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
Created by Me	Garage Management Folder	Bhagya Sri	6/20/2025, 12:06 AM	Bhagya Sri	6/20/2025, 12:06 AM

6.Dashboard Output:

- Visual dashboard component built from the above report.
- Chart Type: Line Chart or Bar Graph.
- Weekly subscription emails sent to Manager.

The screenshot shows a list of dashboards and folders in the Garage Management system. The left sidebar has sections for Dashboards, Folders, and Favorites. The main area lists 'DASHBOARDS' with three items: 'Recent', 'Created by Me', and 'Private Dashboards'. Under 'Private Dashboards', there is one item: 'Service Rating dashboard'. The 'Folders' section shows 'All Folders' with 'Created by Me' and 'Shared with Me' options. The 'Favorites' section shows 'All Favorites'. The top navigation bar includes search, dashboard creation, and other system icons.

Name	Created By	Created On	Last Modified By	Last Modified Date
Enablement Dashboard Spring '24	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
Enablement Dashboard Summer '24	Automated Process	6/13/2025, 5:41 AM	Automated Process	6/13/2025, 5:41 AM
Service Rating dashboard	Bhagya Sri	6/20/2025, 12:38 AM	Bhagya Sri	6/20/2025, 12:38 AM

7.Validation Rules Output:

- Vehicle number plate follows regex format.
- Ratings outside 1–5 are blocked.

The screenshot shows the 'Validation Rules' section for the 'Appointment' object in the Salesforce Setup. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main area shows a table of validation rules. There is one rule named 'Vehicle' with the error location 'Vehicle number plate' and the error message 'Please enter valid number'. The rule is active and was modified by Bhagya Sri on 6/19/2025, 3:08 AM.

Rule Name	Error Location	Error Message	Active	Modified By
Vehicle	Vehicle number plate	Please enter valid number	✓	Bhagya Sri, 6/19/2025, 3:08 AM

8.Sharing & Security Output:

- Service records are private by default (OWD).
- Sharing rule allows Sales Person to share records with Manager.

The screenshot shows the 'Customer Details Sharing Rules' page. It displays two sections: 'Customer Details Sharing Rules' and 'Service records Sharing Rules'. Under 'Service records Sharing Rules', there is a single row for a sharing rule. The rule details are: Action: Owner in Role: sales.person, Shared With: Role: Manager, and Access Level: Read/Write. Buttons for 'New', 'Recalculate', 'Edit | Del', and 'Service records Sharing Rules Help' are visible at the top and bottom of the section.

9.Apex Trigger Output

- On saving an appointment with services selected, Apex calculates the service cost.

The screenshot shows the 'Apex Triggers' page in the Salesforce Setup. The left sidebar is collapsed. The main area shows a table of triggers. One trigger is highlighted: 'AmountDistribution' (Action: Edit | Del, Name: AmountDistribution, Namespace Prefix: , sObject Type: Appointment, Api Version: 64.0, Status: Active, Size Without Comments: 211, Last Modified By: Bhagya Sri, Last Modified: 6/20/2025, 12:04 AM). A yellow box highlights the trigger name. A message bar at the top says 'Percent of Apex Used: 0.02%' and '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.' Below the message bar are buttons for 'Compile all triggers' and 'View: All | Create New View'. A developer console link is also present.

8. ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

1. Automation of Garage Operations:

- Automates appointments, billing, service tracking, and feedback collection.
- Reduces human errors and manual workload.

2. Centralized Customer Management:

- Stores all customer and vehicle data in one place.
- Prevents duplication using matching and duplicate rules.

3. Real-time Service Monitoring:

- Staff can track service progress (Started, Completed) via picklists and checkbox fields.
- Management can see live updates on service delivery.

4. Dynamic Billing System:

- Service amount is auto-calculated based on selected services using Apex triggers.
- Ensures transparent and accurate billing.

5. Automated Communication:

- Sends personalized emails upon payment or service updates via Flows.
- Improves customer engagement and professionalism.

6. Reports and Dashboards:

- Visualizes service performance, customer feedback, and financial data.
- Supports informed decision-making through data-driven insights.

7. Role-Based Security:

- Uses Salesforce roles, profiles, and sharing rules to secure data.
- Ensures only authorized users can view or modify sensitive information.

8. Scalability and Cloud Access:

- Built on Salesforce, it's accessible from any device, anywhere.
- Scales easily for small to large garage businesses.

9. User-Friendly Interface:

- Lightning App provides easy navigation with custom tabs.
- Simplifies data entry and viewing for staff with minimal training.

DISADVANTAGES:

1. Platform Dependency:

- Entire system depends on the Salesforce ecosystem.
- Limited flexibility if migrating to other platforms in the future.

2. Learning Curve for New Users:

- Users unfamiliar with Salesforce may need training to use the system effectively.
- Admin-level tasks (like creating flows or objects) require technical know-how.

3. Customization Limitations:

- Some advanced features may require Apex coding, which is not always low-code.
- Over-customization can complicate maintenance and upgrades.

4. Cost Considerations (Outside Developer Org):

- Salesforce licensing can be expensive for production use (though free for Developer Org).
- Might not be viable for very small or low-revenue garages if commercialized.

5. Dependence on Internet Connectivity:

- Being a cloud-based solution, requires a stable internet connection.
- Offline access is not supported without third-party tools.

6. Limited Offline Functionality:

- Cannot be used during network outages unless integrated with mobile/offline apps.
- Could affect garages in remote areas with unreliable internet.

9. CONCLUSION:

The **Garage Management System (GMS)** developed on the Salesforce platform successfully addresses the key operational challenges faced by modern automotive service centers. By digitizing and automating core functions such as **customer management, appointment scheduling, service tracking, billing, and feedback collection**, the project provides a complete end-to-end solution that enhances efficiency, transparency, and customer satisfaction.

Through the use of **custom objects, automation tools** like Flows and Apex triggers, and **robust data controls** via roles and sharing settings, GMS ensures that the garage operations are not only well-organized but also scalable and secure. Real-time dashboards and reports empower managers with actionable insights, enabling smarter decision-making and continuous performance monitoring.

Furthermore, by implementing **automated communications** and service workflows, the system reduces manual dependency and elevates the customer experience. This project demonstrates the powerful capabilities of Salesforce as a low-code platform for building real-world business applications with minimal development overhead.

In conclusion, the Garage Management System not only improves the internal processes of a garage but also establishes a foundation for **digital transformation** in the automotive service industry, making it future-ready and customer-centric.

10. FUTURE SCOPE:

The current version of the **Garage Management System (GMS)** lays a strong foundation for managing key garage operations efficiently using Salesforce. However, the system can be extended and enhanced in several ways to provide more advanced functionality and better scalability in the future.

1. Integration with External Systems:

- **Payment Gateway Integration:** Link with platforms like Razorpay, Stripe, or PayPal to enable online payment directly from the billing module.
- **SMS Gateway Integration:** Send real-time SMS notifications for appointment confirmations, reminders, and service updates.
- **Google Calendar Sync:** Automatically sync customer appointments with calendars for better scheduling and reminders.

2. Mobile App Development:

Build a **dedicated mobile app** (using Salesforce Mobile SDK or other platforms like Flutter) to allow:

- Customers to book/view appointments
- Staff to update service records on the go
- Managers to view dashboards anytime, anywhere

3. AI & Predictive Analytics:

Integrate **Einstein Analytics (Salesforce AI)** to:

- Predict upcoming service needs based on history
- Analyze customer behavior and recommend upselling services
- Detect anomalies in service trends and customer feedback

4. Inventory Management Module:

- Add functionality to track **spare parts, tools, and stock levels**.
- Auto-update inventory based on services rendered.
- Notify when stock is low and integrate with suppliers for procurement.

5. Maintenance History & Service Plans:

- Maintain a detailed **vehicle maintenance logbook** for each customer.
- Allow creation of **custom service plans or packages** (e.g., yearly maintenance contracts).
- Set up **automated follow-up** notifications for recurring services.

6. Multi-Branch / Franchise Support:

- Extend the architecture to support **multi-location garages** or franchises.
- Use Salesforce's **Record Types** or **Business Units** to isolate data per branch.
- Centralized reporting across all branches for top-level management.

7. Chatbot for Customer Support:

- Integrate a **Salesforce Chatbot** to handle customer FAQs, booking assistance, and feedback collection.
- Provide 24/7 customer service via website or WhatsApp.

8. Advanced Reporting & KPI Tracking:

- Add **custom KPIs** like average turnaround time, top services, or customer satisfaction scores.
- Schedule weekly/monthly auto-reports to key stakeholders.

9. Enhanced User Roles and Permissions:

- Introduce more granular user roles like **Mechanic**, **Receptionist**, or **Accountant**.
- Apply **Field-Level Security** to ensure each user sees only what they need.

10. Customer Self-Service Portal:

- Create a **Salesforce Experience Cloud Portal** where customers can:
 - View past services
 - Download invoices
 - Submit feedback
 - Book new appointments

Summary:

The GMS has significant potential to evolve into a comprehensive **Garage ERP System**, capable of managing not just appointments and services, but also inventory, payments, customer relationships, and business growth strategies. As technology advances and customer

expectations rise, incorporating features like **mobile access**, **AI, integrations**, and **self-service portals** will future-proof the solution and maintain competitive advantage.

11.Appendix:

Validation Rules:

1.Vehicle Number Format:

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

2.Rating Range:

```
NOT(REGEX( Rating_for_service__c , "[1-5]{1}" ))
```

Apex Trigger:

```
trigger AmountDistribution on Appointment__c (before insert, before update) {  
    if(trigger.isbefore && (trigger.isinsert || trigger.isupdate)){  
        AmountDistributionHandler.amountDist(trigger.new);  
    }  
}
```

Github Link:

<https://github.com/venkataramakrishnaakula/Garage-Management-System>

Project Demo Link:

https://drive.google.com/file/d/1DPB-m7Kn4s-Oa7oq_IsQZqqEUW15VU9r/view?usp=sharing