M.Kumarasamy College of Engineering
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 Certified Institution
Thalavapalayam, Karur - 639 113, TAMILNADU.

GARAGE MANAGEMENT SYSTEM

A SALESFORCE PROJECT REPORT

Submitted by

DHARANEESH MK **927622BCB013**

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

KARUR – 639 113

NOVEMBER 2025

M.KUMARASAMY COLLEGE OF ENGINEERING, KARUR

BONAFIDE CERTIFICATE

Certified that this **Salesforce Minor Project IV** report "**GARAGE MANAGEMENT SYSTEM"** is the Bonafide work of **DHARANEESH MK (927622BCB013)** who carried out the project work under my supervision in the academic year 2025 - 2026 **ODD**.

This report has been submitted for the **Salesforce Project - GARAGE MANAGEMENT SYSTEM** review held on_____.

PROJECT COORDINATOR

PROBLEM STATEMENT

The problem statement of this project is “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”.

Automobile repair workshops often face difficulties in managing day-to-day activities such as tracking service jobs, handling customer information, scheduling repairs, and maintaining accurate billing records. Without a proper system, these processes can become disorganized, slow, and prone to errors, affecting customer satisfaction and business growth.

The **Garage Management System (GMS)** is developed to address these issues by offering a simple yet effective platform that enables staff to manage tasks efficiently. It provides features for monitoring repair status, scheduling work, managing spare parts inventory, and ensuring clear communication with clients. By automating repetitive processes and centralizing data, GMS helps garages improve productivity, reduce operational costs, and offer a smoother experience to customers and employees alike.

ABSTRACT

Automobile repair workshops often encounter challenges in managing their daily operations efficiently, including tracking ongoing repairs, maintaining customer details, handling billing, and managing spare parts inventory. Manual processes can lead to delays, data inconsistencies, and reduced customer satisfaction. The Garage Management System (GMS) is developed to overcome these issues by offering a centralized, user-friendly platform that integrates all essential functions into a single application.The system enables workshop staff to record and monitor service requests, allocate jobs to technicians, manage stock levels, and generate accurate invoices with minimal effort. Real-time status updates and automated notifications improve transparency for customers, while comprehensive reporting tools assist management in decision-making. By automating repetitive tasks, reducing paperwork, and ensuring data accuracy, GMS helps garages enhance productivity, reduce operational costs, and maintain strong customer relationships. This solution not only streamlines workflows but also ensures a professional and satisfying experience for both customers and employees.

Keywords: Garage Management, Customer, Appointment, Service, Billing, Records, Feedback, Review, Dashboard, Report, Objects, Fields, Tabs, Lookup Relationship, Flow, Trigger, Apex Class.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ACRONYM	ABBREVIATION
CRM	Customer Relationship Management
GMS	Garage Management System
LWC	Lightning Web Components
API	Application Programming Interface
SOQL	Salesforce Object Query Language
SOSL	Salesforce Object Search Language
VF	Visualforce
DML	Data Manipulation Language
PB	Process Builder
FLO	Flow Builder

CHAPTER 1

INTRODUCTION

1.1 Project Details

The automotive service industry is evolving rapidly, and efficient management of day-to-day operations has become a key factor in delivering high-quality service. A **Garage Management System (GMS)** is a specialized software solution designed to streamline the workflow of automotive repair shops, service centers, and garages. It enables businesses to handle essential activities such as appointment scheduling, job card management, vehicle servicing, billing, inventory control, and customer communication from a unified platform.

Traditional garage operations often rely on manual record-keeping and paper-based processes, which are prone to human error, delays, and mismanagement. This can lead to inefficient service delivery, customer dissatisfaction, and revenue loss. A well-implemented GMS replaces these outdated methods with a digital, automated environment that improves accuracy, speeds up processes, and ensures better coordination between staff and customers.

In this system, customers can easily book appointments and track service progress, while the garage staff can manage repair orders, update service records, and monitor spare parts availability in real time. The integration of features like automated notifications, digital invoices, and service history tracking further enhances transparency and customer trust.

By adopting a Garage Management System, automotive service providers can significantly reduce operational costs, improve service quality, and maintain long-

term relationships with their clients. The system not only supports business growth but adapts to modern customer expectations by providing a user-friendly interface and seamless service experience.

CHAPTER 2

OBJECTIVE

The primary objective of the Garage Management System is to develop a comprehensive, user-friendly, and centralized platform that integrates all essential functions of an automotive service facility into a single, efficient digital solution. It is designed to replace traditional manual processes with automated workflows that handle service bookings, job assignments, repair tracking, spare parts inventory management, billing, and customer communication with speed and accuracy. By reducing repetitive tasks and minimizing human errors, the system enhances operational efficiency and ensures seamless coordination between customers, technicians, and administrative staff. One of its major goals is to improve customer satisfaction by offering real-time service updates, transparent billing, and easy access to vehicle service history. In addition, the system strengthens resource utilization by monitoring spare parts usage, forecasting inventory needs, and preventing wastage. It also provides a structured workflow that enables better time management, faster decision-making, and improved profitability. Beyond daily operations, the Garage Management System offers valuable analytical insights for identifying trends, planning future services, and making informed business decisions, thereby supporting sustainable growth and long-term success in the automotive.

CHAPTER 3

METHODOLGY

The development of the Garage Management System using **Salesforce CRM** follows a systematic approach to ensure efficient design, implementation, and deployment. Since Salesforce is a cloud-based platform, the project leverages its robust features such as **custom objects, workflows, automation tools, and Lightning Web Components (LWC)** to create a scalable and user-friendly solution.

The methodology begins with **requirement gathering and analysis**, where the specific needs of the garage, such as appointment scheduling, job tracking, inventory management, and billing processes, are identified. Based on these requirements, a **custom data model** is designed in Salesforce by creating **custom objects, fields, and relationships** to store and manage information related to customers, vehicles, spare parts, and services.

Next, **automation tools** such as **Flow Builder, Process Builder, and Validation Rules** are used to reduce manual effort, enforce business rules, and ensure data accuracy. **Lightning Web Components (LWC)** are developed for creating an interactive, responsive, and modern user interface that allows users to easily book appointments, update service records, and generate invoices. **Apex classes and triggers** are implemented where advanced business logic is required, enabling backend processing such as inventory updates and automated notifications.

Integration capabilities of Salesforce are utilized to connect the system with **email services and third-party tools** for sending reminders, generating reports, and improving communication with customers. Data security is ensured using **Role-Based Access Control (RBAC)**, **Field-Level Security (FLS)**, and Salesforce's built-in authentication mechanisms.

Finally, the system undergoes **testing and deployment** using Salesforce's **Sandbox environment**, ensuring that all functionalities work as intended before going live. Once deployed, the system can be easily maintained and upgraded due to Salesforce's flexible, cloud-based infrastructure, ensuring the Garage Management System remains adaptable to future business needs.

CHAPTER 3

PROJECT OVERVIEW

The Salesforce-based Garage Management Systemis a customized cloud application built to simplify and enhance the operations of automotive service centers. Using Salesforce CRM as the foundation, the system brings together all critical activities—suchas customer registration, vehicle profiling, service scheduling, repair tracking, parts inventory, and billing—into a unified and accessible platform. Through Salesforce's automation features, routine processes like sending service notifications, generating quotations, updating job progress, and preparing invoices are handled efficiently, reducing manual effort and minimizing errors. Real-time dashboards and detailed reports offer garage managers valuable insights into performance, resource utilization, and customer trends, helping them make informed business decisions. Since the system is hosted in the cloud, it provides secure, anytime access for staff and management, ensuring seamless communication and collaboration. By merging CRM capabilities with garage operations, the solution not only improves workflow efficiency but also delivers a better customer experience, builds stronger client relationships, and supports Sustainable growth in the automotive service industry.

CHAPTER 4

PHASE I: REQUIREMENT, ANALYSIS & PLANNING

4.1 Understanding Business Requirements

The first step in the project involves closely analyzing the core business needs of a typical garage or automotive service center. The main goal is to address the challenges faced in manual operations such as inefficient appointment handling, lack of real-time service tracking, and poor inventory visibility. Through discussions with end-users or stakeholders, key functional requirements were identified — including the ability to book and manage service appointments, generate job cards, update service statuses, track spare parts, handle billing digitally, and maintain customer communication. This phase ensures a clear understanding of user pain points and sets the foundation for designing a solution tailored to daily garage operations.

4.2 Defining Project Scope and Objectives

The boundaries of the project and its deliverables are clearly defined to keep development focused and aligned with goals.

Key elements of the project scope include:

- Development of a centralized system to manage vehicle servicing, customer details, and billing.
- Enabling appointment scheduling and tracking via Salesforce interfaces.
- Managing spare parts and tools using custom inventory objects.
- Automating notifications and updates using Salesforce automation tools.

- Implementing dashboards and reports to monitor service performance and resource usage.

The project's primary objectives are to:

- Streamline daily garage operations through automation.
- Minimize human errors and manual data entry.
- Enhance customer satisfaction through transparency and real-time updates.
- Improve decision-making through insightful reporting tools.

4.3 Design Data Model and Security Model

Using Salesforce, a custom data model is designed to represent all entities involved in garage operations. Custom objects such as **Vehicles**, **Service Requests**, **Job Cards**, **Spare Parts**, and **Invoices** are created with appropriate fields and relationships to ensure smooth data flow. Master-detail and lookup relationships are defined to link objects like customers to vehicles or job cards to services performed.

For security, Salesforce's **Role Hierarchy** and **Profiles** are used to control access to data based on user responsibilities. **Field-Level Security (FLS)** is applied to hide or restrict access to sensitive information. **Permission Sets** are also configured to grant additional privileges when needed without changing the base profile. These measures ensure that mechanics, managers, and administrators only access data relevant to their roles, maintaining data integrity and confidentiality.

CHAPTER 5

PHASE II: SALESFORCE DEVELOPMENT – BACKEND & CONFIGURATION

5.1 Setup environment & DevOps workflow

The development process began by setting up the necessary Salesforce environments. A Salesforce Developer Org was used to build and test the core functionality of the Garage Management System. Additionally, a Sandbox environment was leveraged for safe testing of configurations and code before moving them into production.

To ensure a smooth DevOps workflow, change sets were used to deploy customizations from the sandbox to the production org. Version control practices were followed manually or with integration tools (if used) to track changes. This phase ensured that the team had a controlled and efficient development lifecycle, minimizing deployment risks and ensuring better collaboration.

5.2 Customization

In this project, custom Salesforce objects like Customer Details, Appointments, Service Records, and Billing were created to represent core garage operations. Various field types such as lookup, checkbox, date, currency, text, and picklists were added to capture structured data. Validation rules were applied to ensure correct data entry, such as preventing billing without completed service. Automation tools like Process Builder and Flows streamlined tasks like updating statuses and sending notifications.

As part of the Garage Management System development in Salesforce, several **custom objects** were created to model the key components of the garage workflow. These include:

- **Customer Details Object** – to store personal and contact information of vehicle owners.
- **Appointment Object** – for tracking service bookings with date, time, and assigned mechanic.
- **Service Records Object** – to record job details, parts used, and work status for each service.
- **Billing Details and Feedback Object** – to manage billing amounts, payment modes, and capture customer feedback after service completion.

After object creation, a variety of **custom fields** were added based on business needs. This includes:

- **Lookup Fields** – to create relationships between objects (e.g., linking Appointments to Customers).
- **Checkbox Fields** – to mark service status flags like "Completed" or "Payment Received".
- **Date Fields** – for appointment dates, billing dates, and service start/end times.
- **Currency Fields** – to input charges, taxes, and total amounts for service jobs.
- **Text Fields** – for notes, vehicle models, customer comments, and technician names.
- **Picklist Fields** – to select predefined options like payment mode (Cash, Card, UPI) or service type (Oil Change, Brake Check, etc.).

Additionally, **formula fields** were created in the **Service Records Object** to perform automatic calculations — for example, computing the total service cost by

adding parts and labor charges.

To ensure data quality, **validation rules** were applied:

- One validation rule ensures that **appointments cannot be created without selecting a customer**.
- Another rule in the Billing and Feedback Object **ensures billing cannot be completed if the service status is not marked as "Completed"**.

These customizations and validations help enforce business logic, improve user input accuracy, and create a smooth, error-free workflow within the Salesforce environment.

5.3 Apex classes

As part of the project, I developed two Apex components to handle amount distribution logic. The first is an Apex class named **AmountDistributionHandler**, which contains the main processing logic for distributing amounts based on certain conditions. The second is an Apex trigger called **AmountDistribution**, which activates on data changes and calls the handler class to execute the logic efficiently.

- **Apex Class:** AmountDistributionHandler
- **Apex Trigger:** AmountDistribution

The **AmountDistributionHandler** Apex class contains the core logic for distributing amounts based on predefined rules, keeping the business logic modular. The **AmountDistribution** Apex trigger listens for changes on a specific object and delegates processing to the handler class.

Apex Class: AmountDistributionHandler

The AmountDistributionHandler Apex class is developed to handle the backend logic related to salary or amount distribution in the system. This class encapsulates the core business functionality, ensuring a clean separation between trigger logic and processing logic. It improves code maintainability, testability, and adherence to Salesforce best practices.

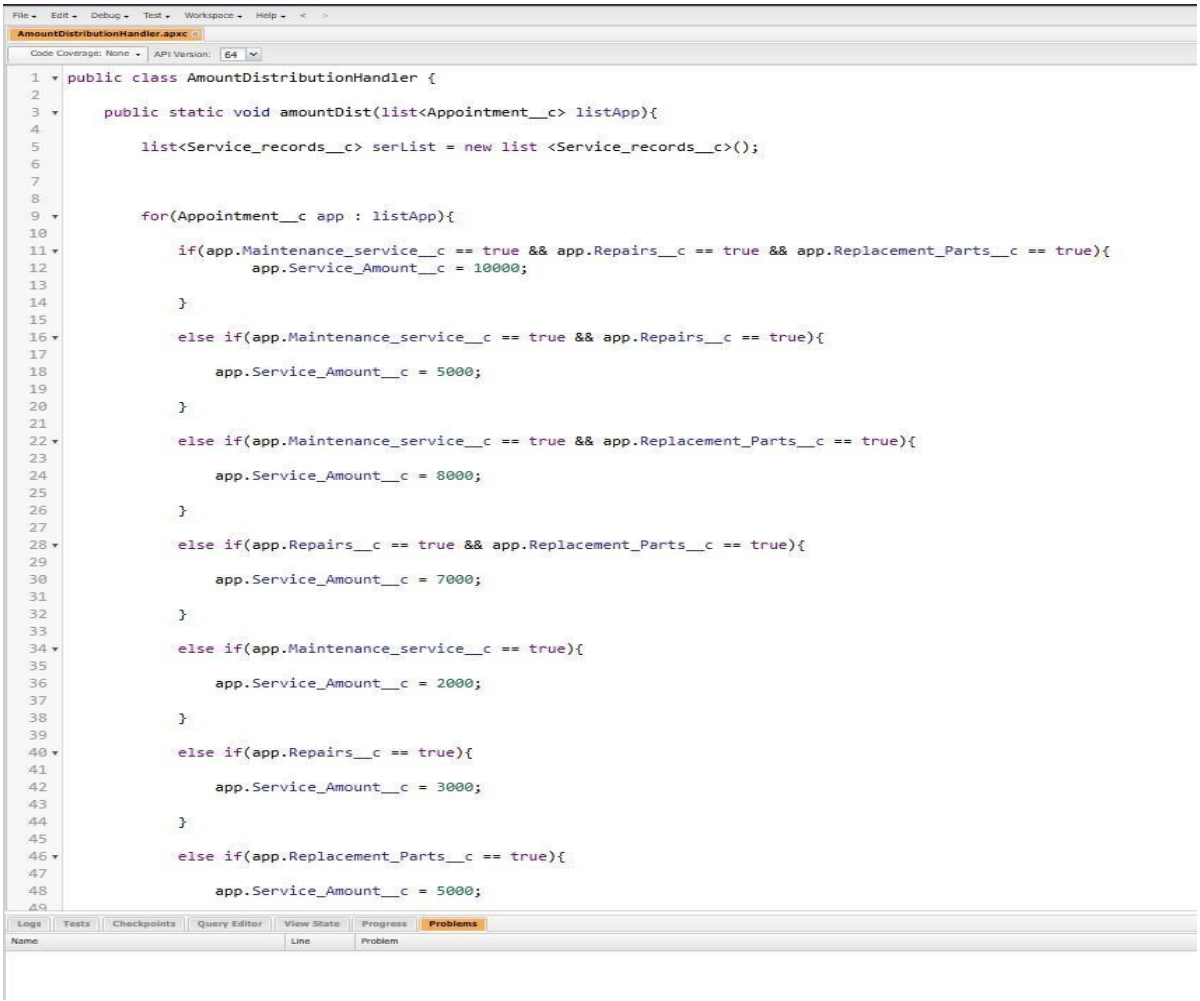

Fig.no.5.1 – Apex Class Program

Apex Trigger: AmountDistribution

The AmountDistribution trigger is written to automatically respond to specific changes on a particular object — most likely a custom object related to salary, payments, or amount management. This trigger is set to run before insert, before update, or after update (depending on your use case) to ensure that the distribution logic is applied consistently whenever relevant records are created or modified.

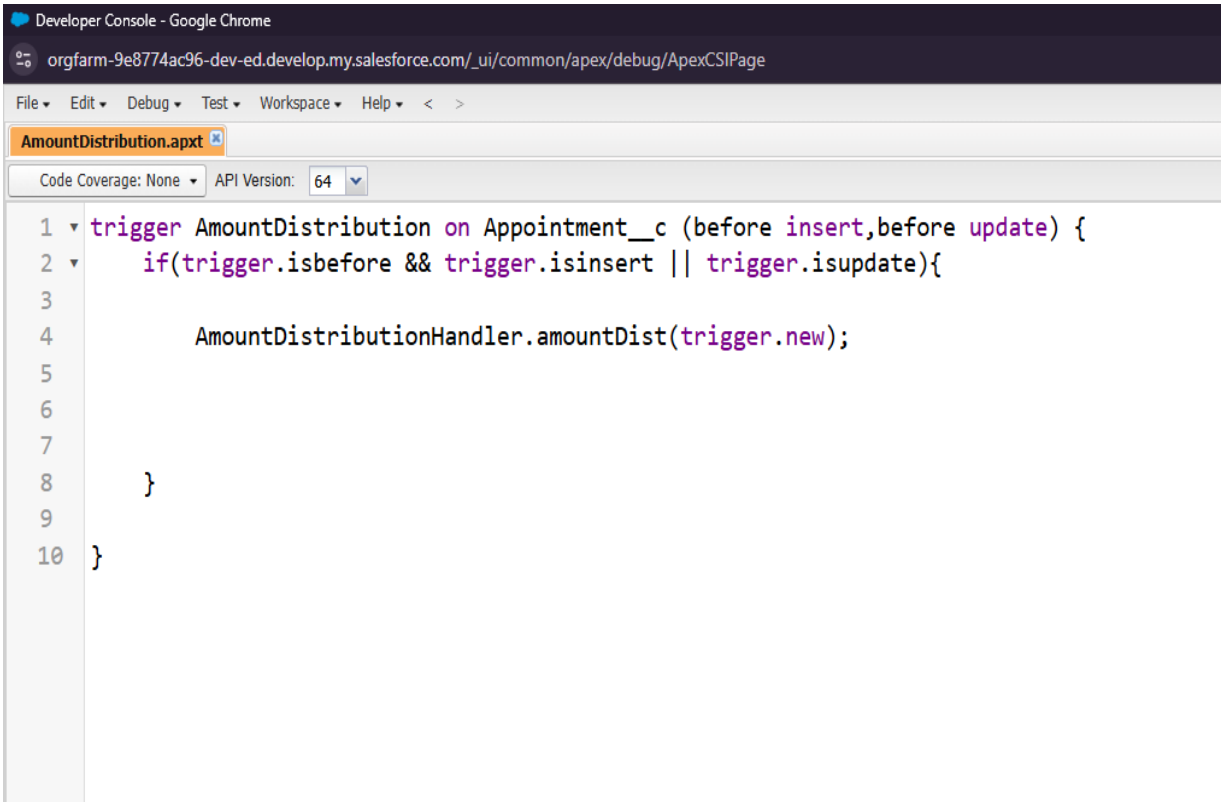

Fig.no.5.2 – Apex Trigger Program

CHAPTER 6

PHASE III: UI/UX DEVELOPMENT & CUSTOMIZATION

6.1 Lightning App setup through App Manager

A dedicated Lightning App was created using Salesforce's App Manager to provide a centralized workspace for the Garage Management System. This app included all the required custom objects, standard objects, tabs, and utilities needed for daily operations. The app design ensured smooth navigation, easy access to records, and a consistent interface across all devices.

6.2 Page Layouts & Dynamic Forms

Custom Page Layouts were designed to arrange fields, related lists, and sections in a logical and user-friendly manner for each object. Dynamic Forms were implemented to control field visibility based on specific conditions, ensuring that users only see relevant information. This reduced clutter, improved efficiency, and made the system more responsive to different use cases.

6.3 User Management

User Management was configured to assign appropriate profiles, roles, and permission sets to different types of users such as mechanics, managers, and administrators. This setup ensured role-based access control, data security, and clear segregation of responsibilities, allowing each user to work within their defined scope without compromising system integrity.

6.4 Reports and Dashboards

As part of the project, I created a report named “New Service Information Report” to display details of all newly logged service requests. This report includes fields such as customer name, vehicle details, service type, assigned mechanic, appointment date, and service status, helping the garage staff efficiently track and manage new work orders.

I also developed a dashboard named “Customer Review” to visually present customer feedback after service completion. This dashboard showcases charts and metrics such as rating distribution, satisfaction trends, and review counts, enabling management to monitor service quality and take necessary improvements based on customer experiences.

- **Report:** New Service Information Report
- **Dashboards:** Customer Review

Report: New Service Information Report

The **New Service Information Report** was created to display details of all newly registered service requests within the Garage Management System. This report includes key information such as the customer name, vehicle details, type of service requested, assigned mechanic, appointment date, and current service status. It helps the garage staff monitor new incoming work, allocate resources effectively, and ensure that all service requests are addressed promptly. Filters and sorting options were applied to make the report more interactive and user-friendly, allowing staff to quickly find specific service records.

Dashboard: Customer Review

The **Customer Review** dashboard presents a visual summary of feedback provided by customers after their service is completed. It includes components such as rating distribution charts, service satisfaction trends, and counts of positive, neutral, and negative reviews. This dashboard enables management to assess service quality, identify areas of improvement, and track customer satisfaction over time. By having this data readily available in a visual format, the garage can maintain high service standards and build stronger customer relationships.

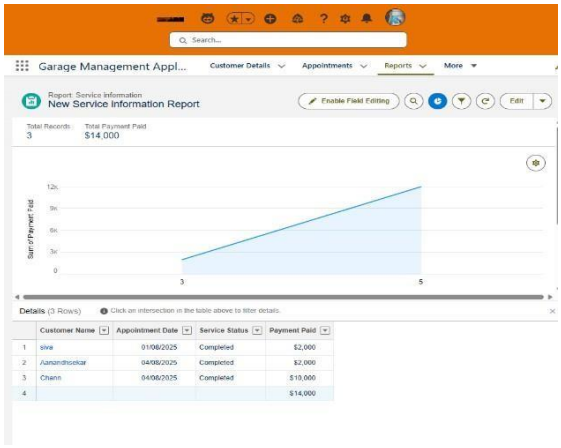

Fig.no.6.1 - Report

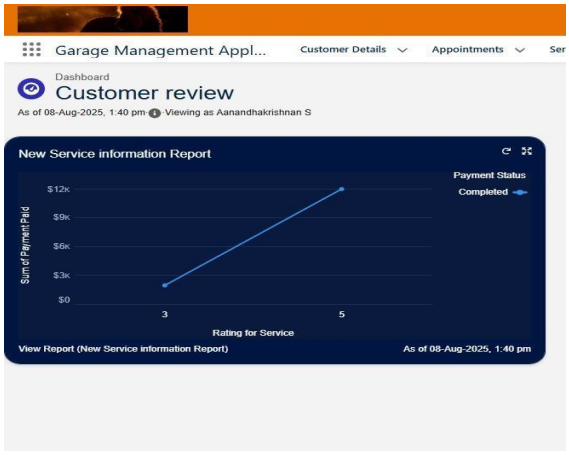

Fig.no.6.2 – Dashboards

6.5 Lightning Pages

As part of the Garage Management System, I designed **Lightning Pages** in Salesforce to provide a streamlined and role-based interface for users. Each Lightning Page was customized to display the most relevant information for the task at hand, combining record details, related lists, and interactive components in a single view. The pages were designed for both desktop and mobile use, ensuring consistent accessibility.

The Lightning Pages include:

- **Customer Details Page** – Displays complete customer information along with related vehicles, past service history, and contact details, enabling staff to quickly review and manage customer relationships.
- **Appointments Page** – Shows scheduled service bookings with appointment dates, assigned mechanics, and service status. Embedded list views and quick actions allow for faster updates and rescheduling.
- **Service Records Page** – Contains detailed job information, including service type, parts used, labor hours, and total costs. Related lists and highlights make it easy to track service progress.
- **Billing and Feedback Page** – Combines billing details with customer feedback, allowing managers to review payments and service ratings in one place.
- **Reports and Dashboards Page** – Provides direct access to the **New Service Information Report** and **Customer Review Dashboard**, giving users a quick, visual overview of operational performance and customer satisfaction without navigating away from their workspace.

By integrating objects, reports, and dashboards into well-structured Lightning Pages, the system delivers a smooth user experience, reduces navigation time, and improves overall productivity.

Lightning Pages in the Project

In this project, multiple Lightning Pages were created and customized using the **Lightning App Builder** to provide a streamlined and user-friendly interface. For each primary object — **Service Records**, **Appointments**, **Billing Details**, and **Customer Feedback** — a dedicated **Record Page** was designed. These pages were structured to display the most relevant fields at the top, followed by related lists, embedded charts, and quick action buttons, ensuring that users can view and update records efficiently.

A **Home Page** was also developed to act as a central hub for users. It includes quick links to key objects, highlights from the **New Service Information Report**, and a visual snapshot of the **Customer Review Dashboard**. This design ensures that important insights are available at a glance without navigating through multiple menus.

Additionally, the Lightning Pages were configured to dynamically adapt based on user roles, so each user sees information and actions most relevant to their responsibilities. By integrating both operational data from objects and visual analytics from reports and dashboards, the Lightning Pages significantly improved the usability and efficiency of the system.

CHAPTER 7

PHASE IV: DATA MIGRATION, TESTING & SECURITY

7.1 Data Loading Process

For migrating existing garage-related data into Salesforce, both the Data Import Wizard and Data Loader tools were utilized. The Data Import Wizard was applied for smaller, standard object datasets, ensuring quick uploads with minimal configuration. The Data Loader was used for large and complex datasets, such as historical service records and appointment logs, allowing efficient bulk operations and upserts.

7.2 Field History Tracking, Duplicate Rules, and Matching Rules

Field History Tracking was enabled for critical objects like *Service Records* and *Billing Details* to maintain a log of changes for auditing purposes. Duplicate Rules were implemented to prevent the creation of redundant customer or vehicle records, while Matching Rules ensured that data consistency was maintained by identifying similar records based on predefined criteria.

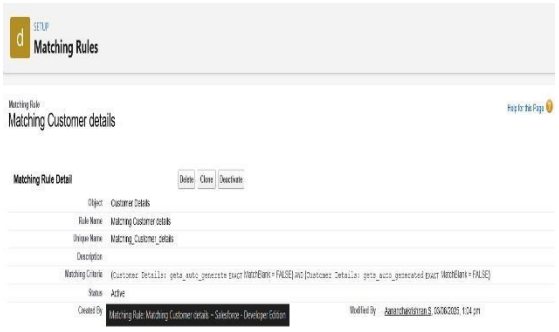

Fig.no.7.2.1- Matching rules

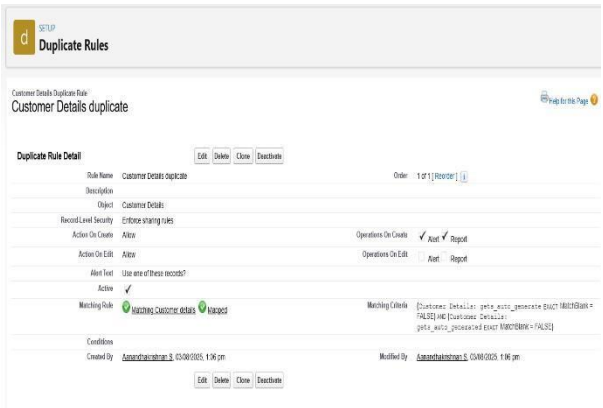

Fig.no.7.2.2 – Duplicate rules

7.3 Profiles, Roles, Role Hierarchy, Permission Sets, and Sharing Rules

Custom Profiles were created to define the baseline permissions for different user groups such as Service Advisors, Technicians, and Admins. A structured Role Hierarchy ensured that managers could view and control data for their teams, while Permission Sets provided additional, temporary access to specific features without altering base profiles. Sharing Rules were applied to securely share records across departments where collaboration was necessary, without compromising data security.

7.4 Creation of Test Class

Apex Test Classes were developed to validate the functionality of triggers, classes, and automation before deployment. These test classes simulated real-time scenarios for service bookings, billing updates, and record modifications, ensuring that all code met Salesforce's 75% coverage requirement and functioned as intended in production.

7.5 Preparation of Test cases

As part of the testing phase, detailed test cases were designed to validate the proper functioning of all implemented Salesforce features in the Garage Management System. This included scenarios for service booking creation, ensuring accurate data capture and record generation; approval processes, verifying that workflow approvals followed the correct hierarchy and conditions; and automatic task creation, confirming that tasks were generated based on specific service events.

For the Apex classes and triggers, test scenarios were built to simulate real-time operations, such as amount distribution after service completion and automated updates to related records. Similarly, for the two developed flows, test cases were

prepared to check each decision path, input validation, and automation execution. Each test case documented the steps to reproduce, expected results, and actual results, along with input and output screenshots, to ensure clear traceability and proof of successful execution.

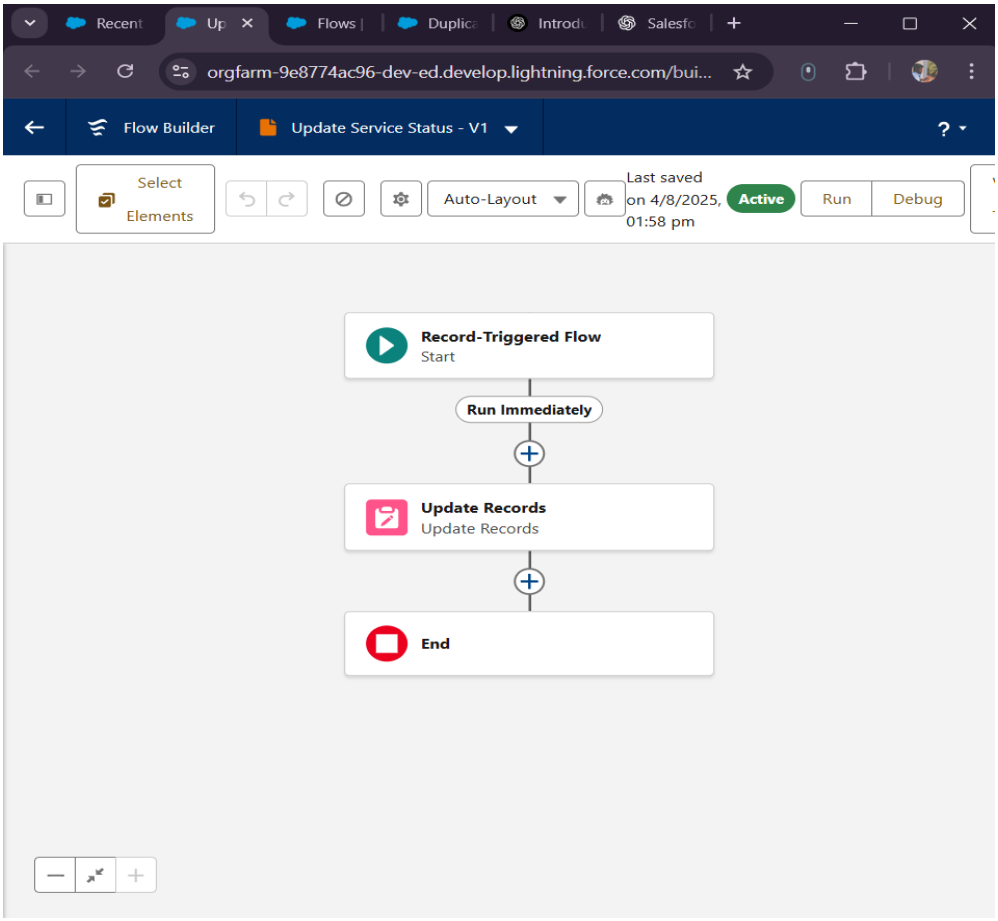

Fig.no.7.5.1 – Flow of Updating the Status of the Service

In the above flow (i.e.Fig.no.7.5.1), a **Salesforce Record-Triggered Flow** was developed to automatically update the *Service Status* field based on specific record changes. This flow ensures that service progress is tracked in real-time without the need for manual intervention. When certain conditions are met — such as completion of a service or change in appointment details — the flow updates the

Service Status to the appropriate value (e.g., “In Progress,” “Completed,” or “Pending”).

The automation was configured with decision elements to evaluate different service conditions and assignment elements to apply the status updates. By implementing this flow, the process of monitoring service stages became faster, more consistent, and error-free, enhancing overall operational efficiency in the system.

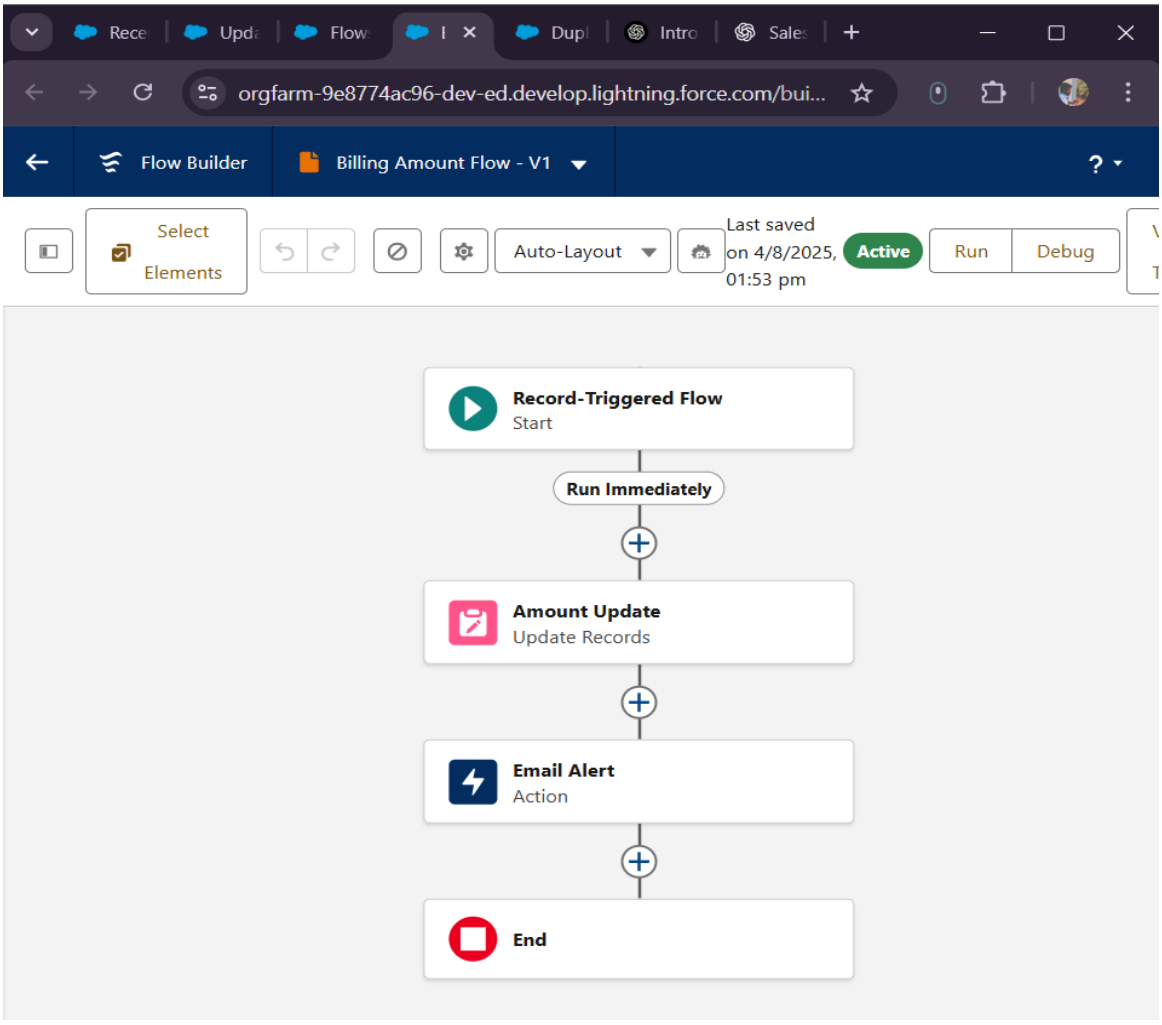

Fig.no.7.5.2 – Flow for Sending Email

In the above flow (i.e. Fig.no.7.5.2), a **Salesforce Record-Triggered Flow** named *Billing Amount Flow* was implemented to automate the final billing and customer notification process. This flow is triggered when a service record is marked as closed, indicating that all service activities have been completed. Once triggered, the flow calculates or retrieves the **Billing Amount** associated with the service and updates the relevant field in the record.

After the billing details are updated, the flow uses the customer's email address from the *Customer Details* object to automatically send a confirmation email containing the billing amount and service summary. This automation ensures that customers receive timely and accurate billing information without any manual follow-up, improving both efficiency and customer satisfaction.

7.6 Working of the Project

In the Garage Management System, a Salesforce Record-Triggered Flow named *Billing Amount Flow* was implemented to automate the final billing and customer notification process. This flow is triggered when a service record is marked as closed, indicating that all service activities have been completed. Once triggered, the flow calculates or retrieves the Billing Amount associated with the service and updates the relevant field in the record.

After the billing details are updated, the flow uses the customer's email address from the *Customer Details* object to automatically send a confirmation email containing the billing amount and service summary. This automation ensures that customers receive timely and accurate billing information without any manual follow-up, improving both efficiency and customer satisfaction.

The Garage Management System in Salesforce follows a streamlined, relationship-driven workflow to manage the complete service lifecycle. The process initiates with the creation of a **Customer Details** record, capturing essential client information such as name, contact details, and vehicle information. This record is linked via a **lookup relationship** to the **Appointment** object, where service scheduling and time allocation are managed.

Once the appointment is booked, the system uses the same relationship structure to create a corresponding record in the **Service Records** object, where all service-related activities, including maintenance, repairs, and parts replacement, are logged. Upon service completion, the workflow advances to the **Billing Details** object, where the total service cost is calculated and stored, ensuring transparent financial tracking. The process concludes in the **Feedback** object, where customers can share their experience, helping improve service quality. By leveraging lookup relationships and Salesforce's object customization, this workflow enables smooth data transfer across all stages, ensuring accuracy, traceability, and a connected customer experience from booking to feedback collection.

CHAPTER 8

RESULT

In the final stage of development, the Garage Management System application was configured using a Lightning Page in Salesforce to provide a centralized and user-friendly interface. The Lightning Page was designed to bring together all the core components of the system, including the four primary custom objects — Customer Details, Appointment, Service Records, and Billing Details — ensuring seamless navigation and data accessibility. Additionally, the Reports and Dashboards created during the project were embedded within the page, enabling real-time data visualization and performance tracking directly from the app interface. This approach provided a single, interactive workspace for managing all aspects of the garage operations, improving efficiency and enhancing the overall user experience.

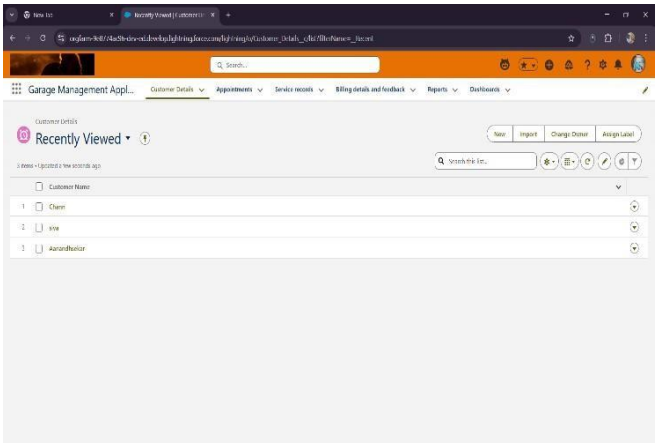

Fig.no.8.1 – App Page of Customer Details Object

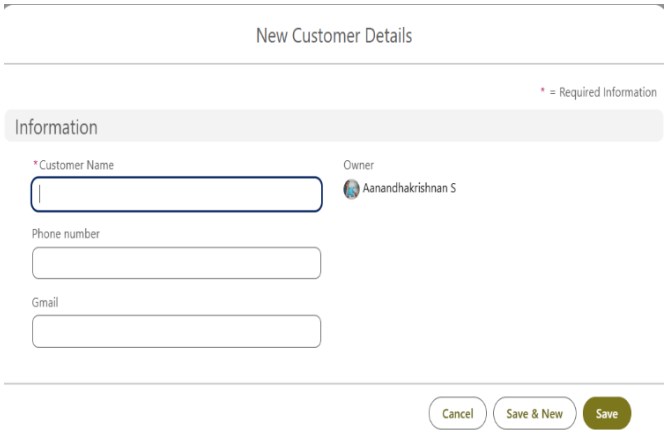

Fig.no.8.2 – Record Creating Page of Customer Details Object

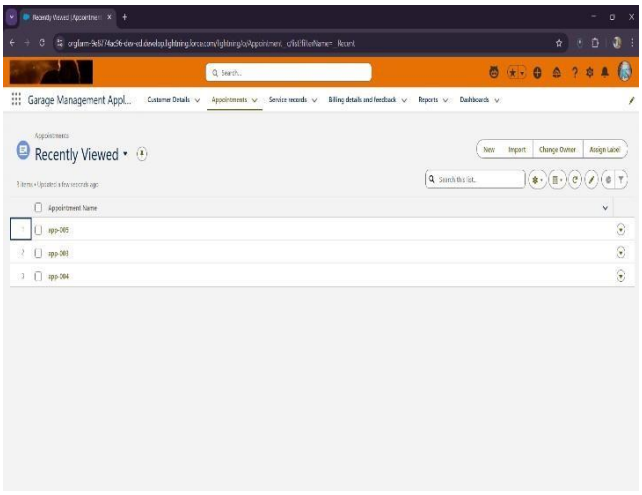

Fig.no.8.3 – App Page of Appointment Object

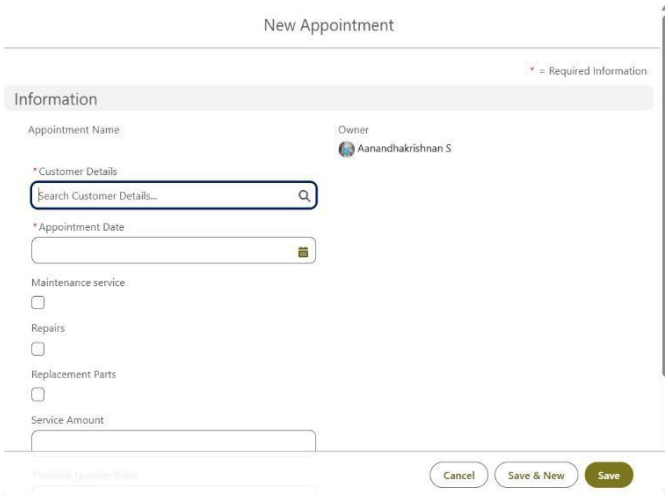

Fig.no.8.4 – Record Creating Page of Appointment Object

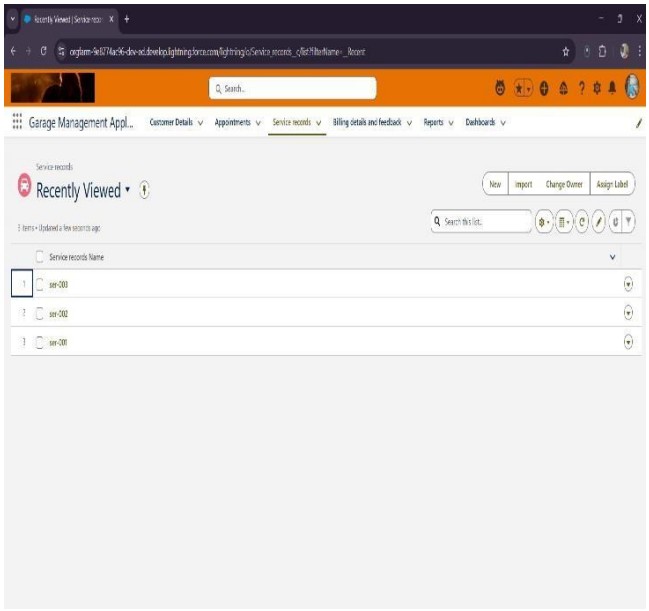

Fig.no.8.5 – App Page of Service records Object

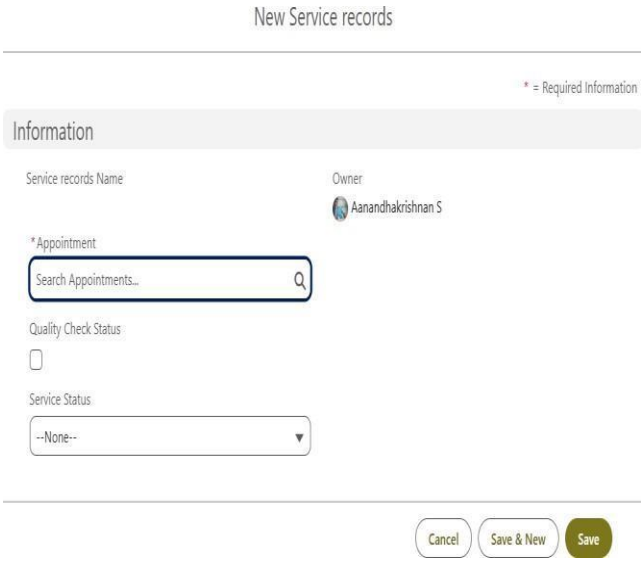

Fig.no.8.6 – Record Creating Page of Service records Object

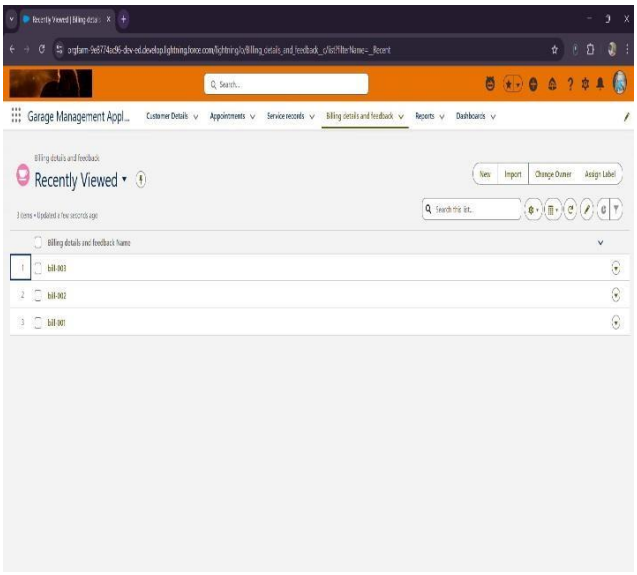

Fig.no.8.7 – App Page of Billing details and feedback Object

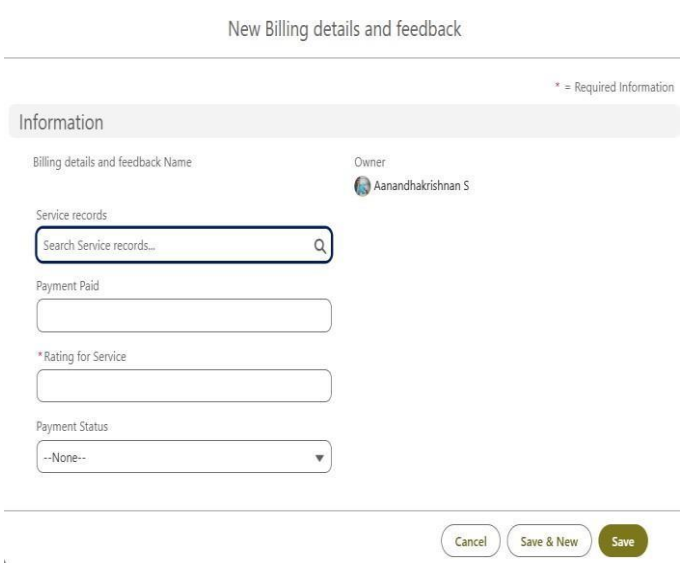

Fig.no.8.8 – Record Creating Page of Billing details and feedback Object

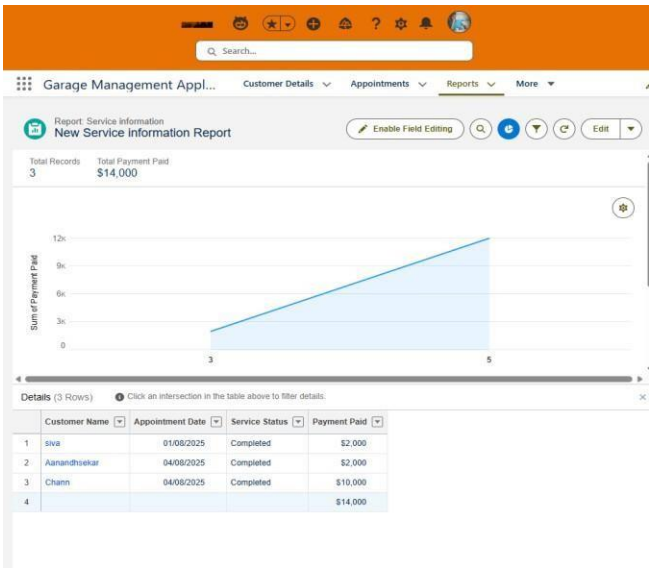

Fig.no.8.9 – Report Page for final Records

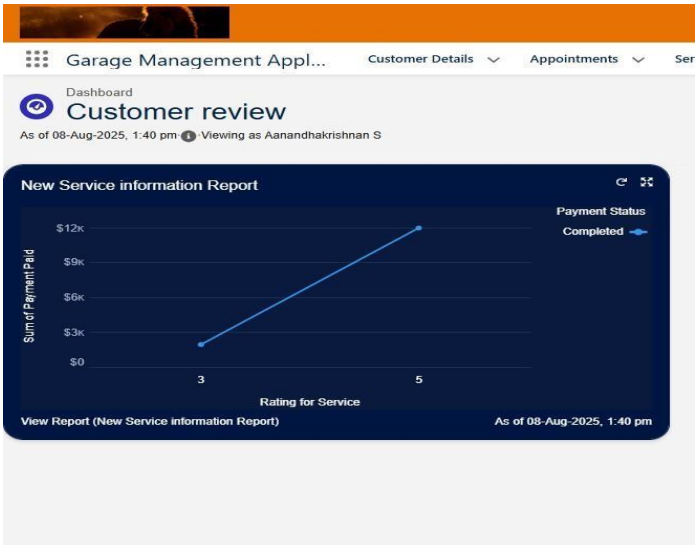

Fig.no.8.10 – Dashboard for the Customer Reviews

After the completion of all record creation and service-related processes, the system was configured to automatically send email notifications using a Salesforce Flow. This flow is triggered once the final steps, such as closing a service or generating the billing details, are completed. The customer's email address, stored in the Customer Details object, is dynamically fetched, and a pre-defined email template is used to send the confirmation or billing information. This automation ensures timely communication with customers, eliminates manual follow-up, and enhances service transparency within the Garage Management System**.**

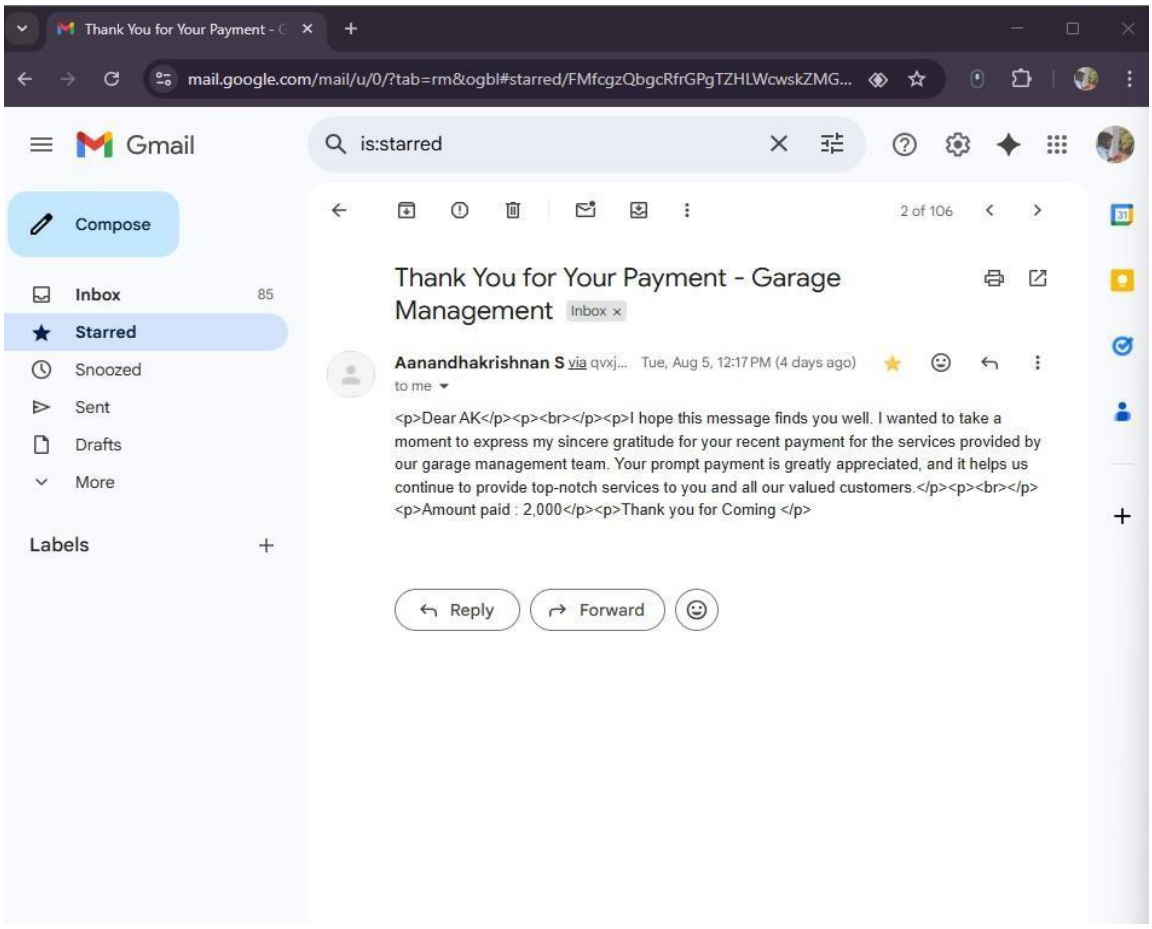

Fig.no.8.11 – Output of the Project by Sending Mail

CHAPTER 9

FUTURE SCOPE

While the current Garage Management System is primarily designed for booking and managing vehicle service appointments, it has strong potential for expansion into other areas of the automotive industry. In the future, the application can be enhanced to include a vehicle rental module, allowing customers to book cars or bikes for short or long-term use. Additionally, it can be extended to support buying and selling of pre-owned vehicles, connecting dealers and customers within the same platform. Another planned enhancement is to introduce automotive consulting services, providing expert guidance on vehicle maintenance, upgrades, and purchases.Furthermore, the system can incorporate insurance renewal and management features, enabling customers to track policy expiry dates, receive renewal reminders, and complete the process directly through the application. These improvements will transform the platform from a service booking tool into a complete automotive management ecosystem

CHAPTER 10

REFERENCES

[1] Salesforce, "Reports and Dashboards Overview," *Salesforce Help Documentation*, 2025. [Online]. Available: https://help.salesforce.com.

[2] Salesforce, "Lightning App Builder Overview," *Salesforce Developer Guide*, 2025. [Online]. Available: https://developer.salesforce.com.

[3] Salesforce, "Flow Builder – Automating Business Processes," *Salesforce Automation Tools*, 2025. [Online]. Available: https://help.salesforce.com/s/articleView?id=sf.flow_build.htm.

[4] SmartBridge, "Salesforce CRM Projects – SmartBridge Industry Connect Programs," *SmartBridge Official Portal*, 2025. [Online]. Available: https://smartbridge.in.

[5] SmartInternz, "Salesforce Virtual Internship Program," *SmartInternz Platform*, 2025. [Online]. Available: https://smartinternz.com.

[6] Admin Hero, "Best Practices for Designing Salesforce Reports and Dashboards," *Salesforce Community Blog*, 2024.

[7] Automation Champion, "Practical Use Cases for Salesforce Flows and Process Automation," *Automation Champion Blog*, 2024.

[8] Trailhead by Salesforce, "Lightning Experience Customization," *Salesforce Trailhead Modules*, 2025. [Online]. Available: https://trailhead.salesforce.com.

CHAPTER 11

CONCLUSION

The Garage Management System developed in Salesforce ensures efficient handling of customer service requests, appointments, and billing through an integrated and automated workflow. The system will be maintained by regularly monitoring performance, updating records, and optimizing automation flows to match evolving business needs. Periodic checks of reports, dashboards, and object relationships will help ensure data accuracy and smooth operation.

For troubleshooting, a structured approach will be followed — starting with identifying the issue from user feedback or system logs, replicating the problem in a test environment, and analyzing object relationships, automation rules, and field dependencies. Necessary adjustments will be made in Flow Builder, page layouts, or validation rules, followed by testing and redeployment. Comprehensive documentation of system configuration, automation logic, and troubleshooting steps will be maintained, allowing future administrators to easily manage, monitor, and enhance the application.