

# **Garage Management System**

By

**Ajay Sharma**  
[imajaysharma1604@gmail.com](mailto:imajaysharma1604@gmail.com)

# Project Abstract

The Salesforce Garage Management System (GMS) project is a comprehensive solution designed to enhance the operational efficiency and service quality of automotive service centers. Built on the Salesforce platform, this system integrates key functionalities such as customer relationship management (CRM), service scheduling, inventory management, invoicing, and real-time analytics, providing a seamless and automated workflow for garage operations.

The project began with an in-depth analysis of the existing processes in typical garage operations, identifying key pain points such as inefficient service scheduling, fragmented customer communication, and manual data entry errors. To address these challenges, the GMS was designed to centralize all essential operations on a single platform, allowing service centers to manage their day-to-day activities more effectively.

Key features of the system include:

1. **Customer Relationship Management (CRM):** The GMS provides a 360-degree view of customer interactions, allowing for personalized communication, targeted marketing, and improved customer retention. Customers can schedule appointments, track the status of their vehicle repairs, and receive updates in real-time.
2. **Service Scheduling and Management:** The system offers an intuitive interface for booking and managing service appointments. It ensures optimal resource allocation by matching available mechanics with service requests, reducing wait times and maximizing productivity.
3. **Invoicing and Payments:** The system automates the generation of invoices based on completed services and parts used, integrating with various payment gateways for smooth and secure transactions.
4. **Reporting and Analytics:** Advanced reporting tools provide insights into key performance indicators (KPIs), such as service turnaround time, customer satisfaction, and revenue trends. These analytics empower garage managers to make data-driven decisions to continuously improve operations.

The implementation of the Salesforce Garage Management System has resulted in a significant reduction in manual tasks, increased operational efficiency, and enhanced customer satisfaction. The system not only streamlines day-to-day operations but also provides a foundation for future growth, with the ability to incorporate additional features and integrations as business needs evolve.

# **Index Page**

<b>Topics</b>	<b>Page no</b>
Introduction	04
Task 1 (Object Creation)	06
Task 2 (Tabs Creation)	10
Task 3 (The Lightning App Creation)	11
Task 4 (Fields Creation)	12
Task 5 (Validation Rule Creation)	17
Task 6 (Duplicate Rule Creation)	18
Task 7 (Profiles Creation)	20
Task 8 (Role & Role Hierarchy Creation)	22
Task 9 (Users Creation)	23
Task 10 (Public Groups Creation)	24
Task 11 (Sharing Setting Creation).	25
Task 12 (Flows Creation)	26
Task 13 (Apex Trigger Creation)	28
Task 14 (Reports Creation)	30
Task 15 (Dashboards Creation)	31

# Introduction

The automotive industry is rapidly evolving, with customers demanding faster, more reliable, and personalized service experiences. In this competitive landscape, service centers must adopt innovative solutions to manage their operations efficiently while maintaining high standards of customer satisfaction. The Salesforce Garage Management System (GMS) project was conceived to address these needs by providing a comprehensive, cloud-based solution that streamlines the operations of automotive service centers.

## Project Background

Traditional garage management systems often rely on disparate tools and manual processes, leading to inefficiencies, errors, and a lack of transparency. Service centers frequently face challenges such as miscommunication with customers, delays in service due to inadequate scheduling, and inventory shortages that can disrupt repair work. These issues not only reduce operational efficiency but also negatively impact customer satisfaction and loyalty.

Recognizing these challenges, our team embarked on the development of a Salesforce-based Garage Management System. The primary objective was to create an integrated platform that could centralize all critical aspects of garage management, from customer relationship management (CRM) and service scheduling to inventory control and invoicing. Salesforce was chosen as the platform due to its robust customization capabilities, scalability, and ability to deliver real-time analytics, making it an ideal choice for modern service centers.

## Project Objectives

The Salesforce GMS was designed with several key objectives in mind:

1. **Enhance Operational Efficiency:** By automating routine tasks such as service scheduling, inventory management, and invoicing, the system reduces the manual workload for service center staff, allowing them to focus on delivering quality service.
2. **Improve Customer Communication and Satisfaction:** The integrated CRM features provide a 360-degree view of customer interactions, enabling personalized communication and real-time updates on vehicle status. This fosters transparency and builds trust with customers.
3. **Optimize Resource Allocation:** The system's scheduling module ensures that mechanics and service bays are optimally utilized, reducing wait times and improving service turnaround times.
4. **Ensure Data Accuracy and Integrity:** By centralizing data on a single platform, the GMS minimizes errors caused by manual data entry and ensures that accurate information is available to all stakeholders at all times.
5. **Provide Actionable Insights:** Advanced reporting and analytics tools offer insights into key performance indicators (KPIs), such as service efficiency, customer satisfaction, and inventory levels. These insights enable data-driven decision-making, helping service centers continuously improve their operations.

## **Conclusion**

The Salesforce Garage Management System represents a significant advancement in the management of automotive service centers. By leveraging the power of Salesforce, the system not only addresses existing inefficiencies but also sets the stage for future growth and innovation in the industry. The successful implementation of this project highlights the potential of cloud-based solutions to transform traditional business operations, making them more efficient, customer-centric, and data-driven.

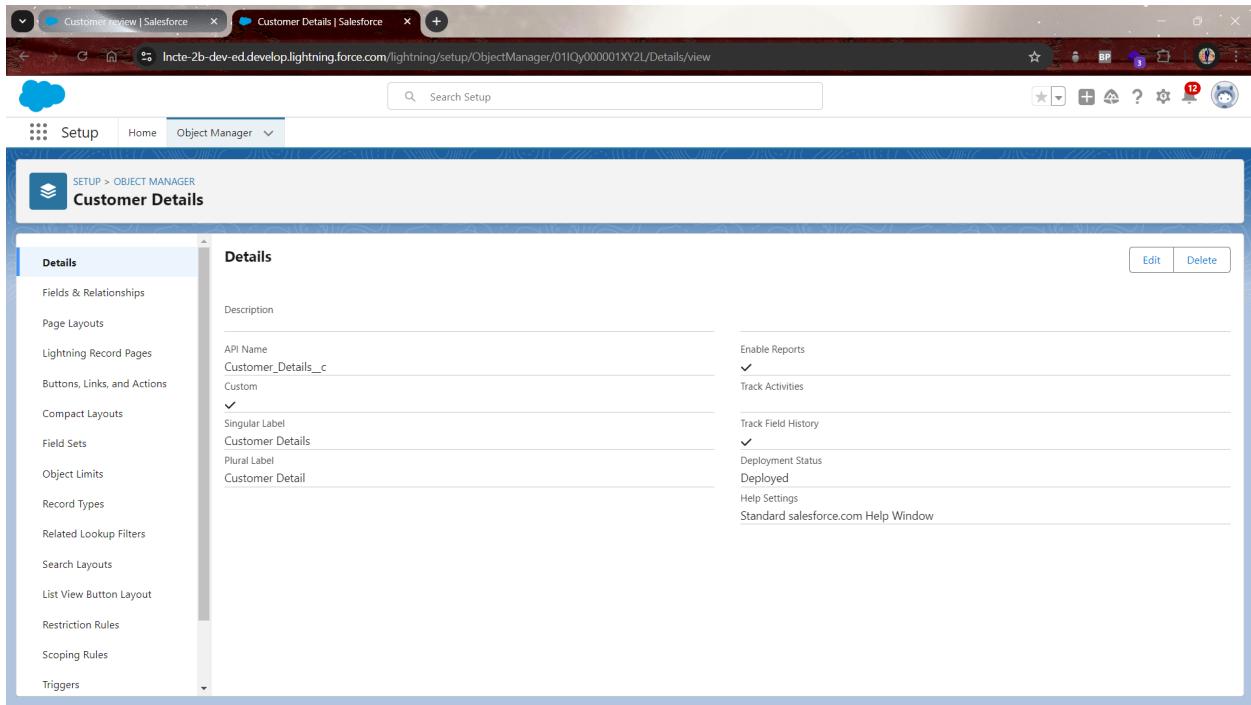
## Task 1 (Object Creation)

### 1. Customer Details Object:

**Purpose:** This object stores all relevant information about customers, including their personal details, contact information, and vehicle details. It serves as the central repository for managing customer interactions and maintaining a comprehensive service history.

#### To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name >> Customer Details
3. Plural label name >> Customer Details
4. Enter Record Name Label and Format.
  - Record Name >> Customer Name
  - Data Type >> Text
5. Click on Allow reports and Track Field History,
6. Allow search >> Save.

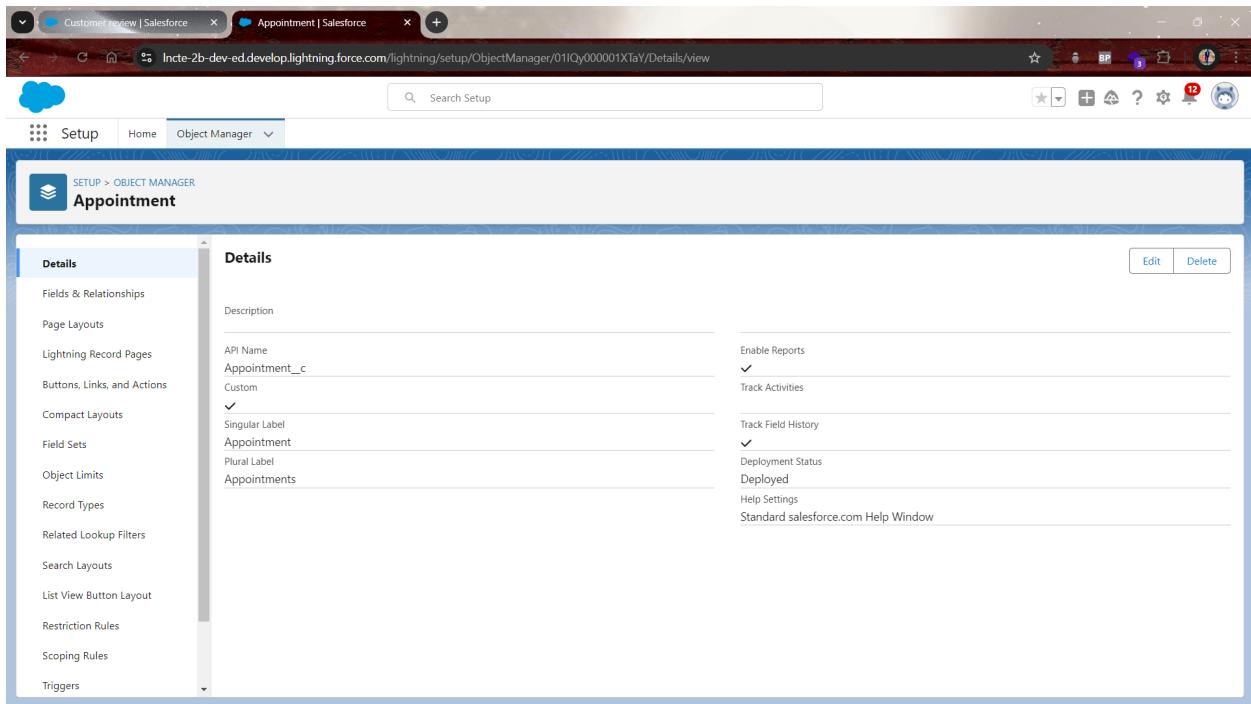


## 2. Appointment Object:

**Purpose:** This object is designed to manage and track service appointments. It allows service centers to schedule, update, and monitor appointments efficiently, ensuring that the right resources are allocated at the right time.

### To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name >> Appointment
3. Plural label name >> Appointments
4. Enter Record Name Label and Format.
  - Record Name >> Appointment Name
  - Data Type >> Auto Number
  - Display Format >> app-{000}
  - Starting number >> 1
5. Click on Allow reports and Track Field History,
6. Allow search >> Save.

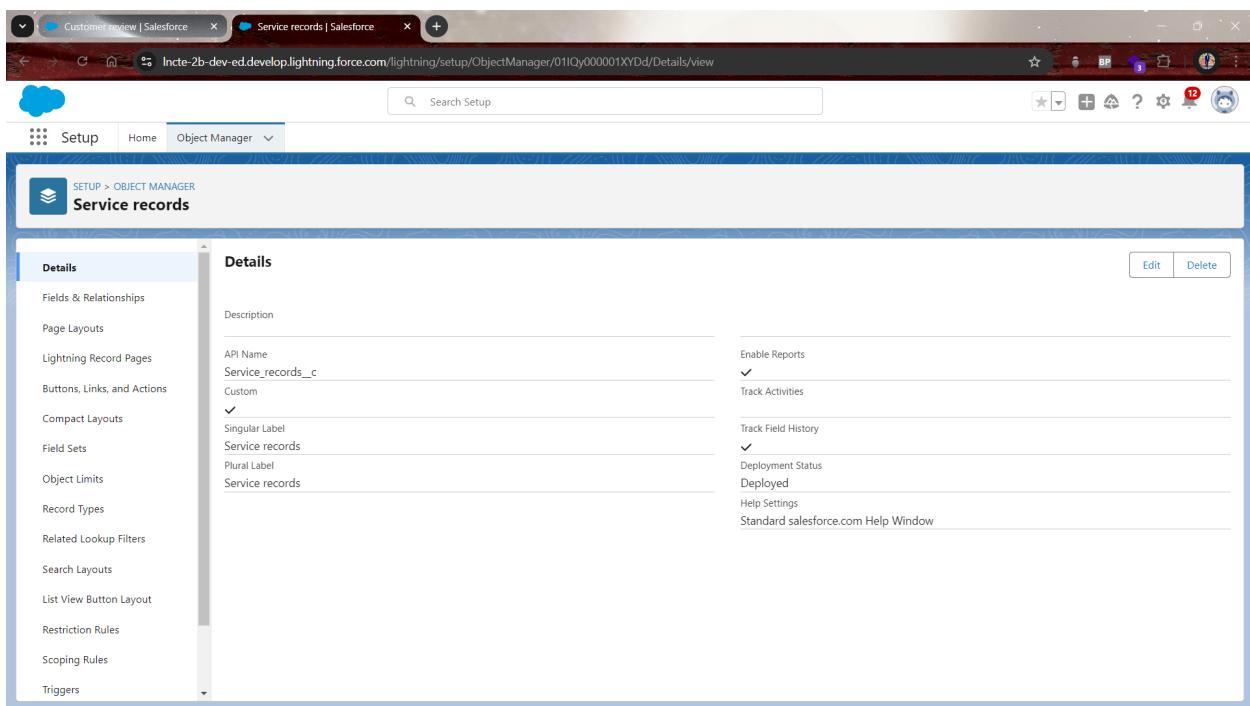


### 3. Service Records Object:

**Purpose:** The Service Records object keeps a detailed account of all services performed on a vehicle. It logs each service activity, parts used, and any additional notes, creating a complete service history for each vehicle.

#### To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name >> Service record
3. Plural label name >> Service records
4. Enter Record Name Label and Format.
  - Record Name >> Service records Name
  - Data Type >> Auto Number
  - Display Format >> ser-{000}
  - Starting number >> 1
5. Click on Allow reports and Track Field History,
6. Allow search >> Save.

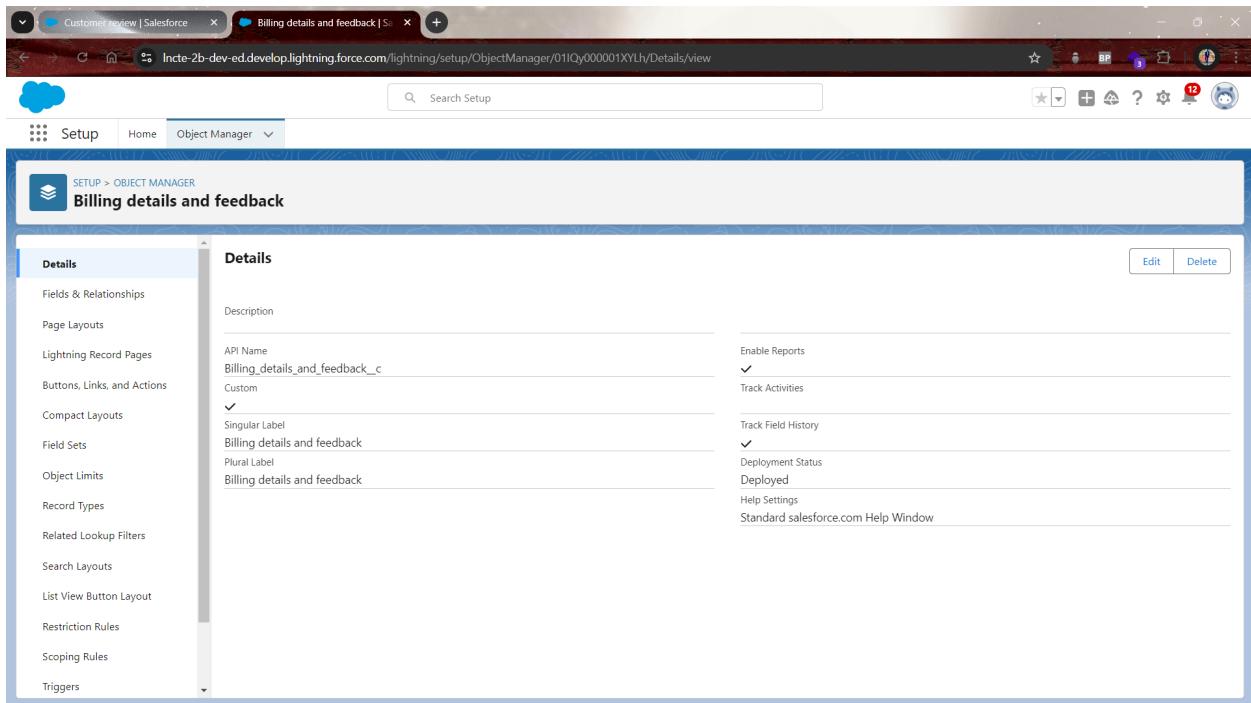


#### 4. Billing Details and Feedback Object:

**Purpose:** This object manages billing and customer feedback. It captures all financial transactions related to services, including invoices and payments, and also collects customer feedback to assess satisfaction and identify areas for improvement.

##### To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
2. Enter the label name >> Billing details and feedback
3. Plural label name >> Billing details and feedbacks
4. Enter Record Name Label and Format.
  - Record Name >> Billing details and feedback Name
  - Data Type >> Auto Number
  - Display Format >> bill-{000}
  - Starting number >> 1
5. Click on Allow reports and Track Field History,
6. Allow search >> Save.



## Task 2 (Tabs Creation)

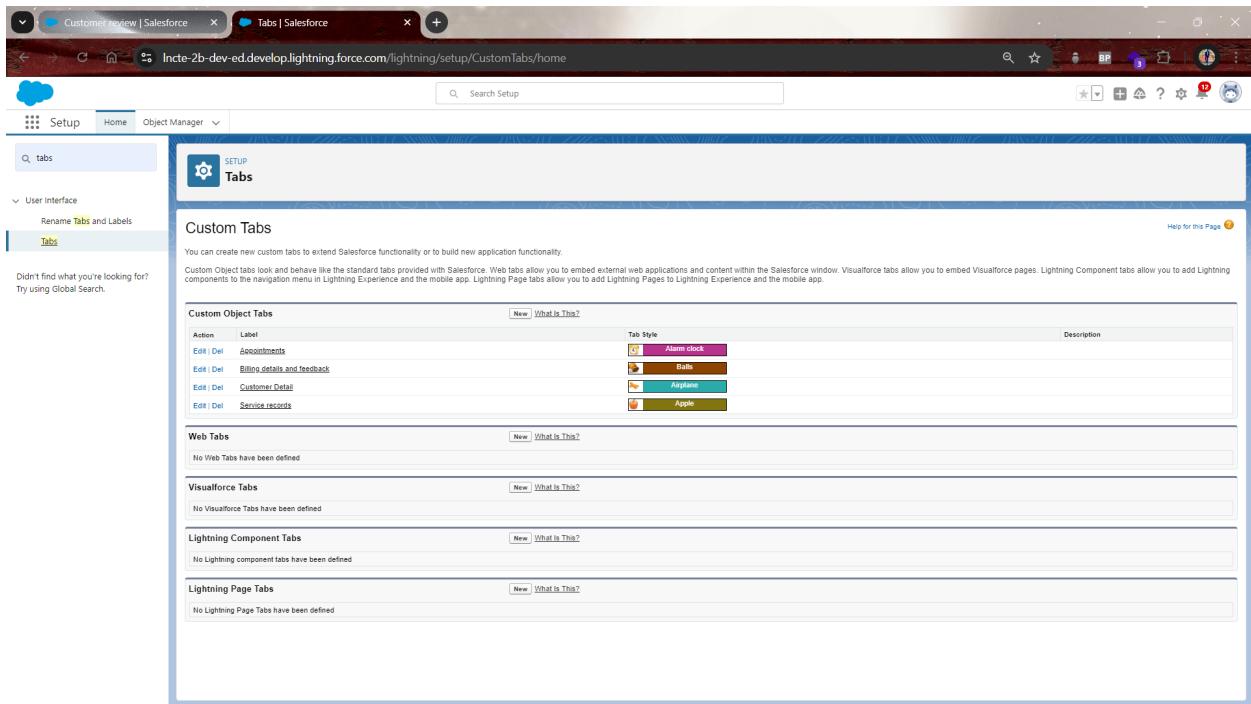
### Creating A Custom Tab

#### 1. To create a Tab:(Customer Details)

1. Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab)
2. Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .
3. Make sure that the Append tab to users' existing personal customizations is checked.
4. Click save.

#### 2. Creating Remaining Tabs

1. Now create the Tabs for the remaining Objects, they are “ Appointments, Service records,Billing details and feedback”.
2. Follow the same steps as mentioned in Activity -1 .



## Task 3 (The Lightning App Creation)

### Create A Lightning App

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.
2. Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.  
To Add Navigation Items:
  3. Select the items (Customer Details, Appointments, Service records, Billing details and feedback, Reports and Dashboards) from the search bar and move it using the arrow button >> Next.
  4. To Add User Profiles: Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

The screenshot shows the Salesforce App Manager interface. The main area displays a grid of available apps, including Service, Marketing CRM Classic, Community, Salesforce Chatter, Content, Sales, Service Console, Digital Experiences, Salesforce Scheduler Setup, Lightning Usage App, Automation, and the newly created 'Garage Management Application'. The sidebar on the left contains sections for Data, Apps, AppExchange, Connect, External, Setups, Lightning Flow, Mobile, Sales, Service, and Bolt Solutions. The bottom of the screen shows the 'Salesforce Settings' section with options for Account, Change Requests, Enhanced Letterheads, Macros, and Recycle Bin.

The screenshot shows the 'Customer Detail' page for the 'Garage Management Application'. The top navigation bar includes links for Customer Detail, Appointments, Service records, Billing details and feedback, Reports, and Dashboards. The main content area displays a list titled 'Recently Viewed' with 10 items, each with a checkbox and a link to the customer record. The list includes: 1. user10, 2. user9, 3. user8, 4. user7, 5. user6, 6. user5, 7. user4, 8. user3, 9. user2, and 10. user1. The page also features a search bar at the top right and various edit and delete icons.

## **Task 4 (Fields Creation)**

### **Creation Of Fields For The Customer Details Object**

#### **1. To create fields in an Customer object:**

1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data Type as a “Phone”
4. Click on next.
5. Fill the details as Following:
  - Field Label: Phone number
  - Field Name : gets auto generated
  - Click on Next >> Next >> Save

#### **2. To create another fields in an object:**

1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Email” and Click on Next
4. Fill the details as following:
  - Field Label : Gmail
  - Field Name : gets auto generated
  - Click on Next >> Next >> Save and new.

### **Creation Of Lookup Fields**

#### **1. Creation of Lookup Field on Appointment Object :**

1. Go to setup >> click on Object Manager >> type object name( Appointment ) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select “Look-up relationship” as data type and click Next
4. Select the related object “ Customer Details” and click next.
5. Next >> Next >> Save.

#### **2. Creation of Lookup Field on Service records Object :**

1. Go to setup >> click on Object Manager >> type object name( Service records ) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select “Look-up relationship” as data type and click Next.

4. Select the related object “ Appointment ” and click next.
5. Make it a required field so click on Required.
6. Scroll down for Lookup Filter and click on Show filter settings.
7. Now add the filter criteria.
8. Field : Appointment: Appointment Date >> Operator : less than >> select field >> Appointment: Created Date(Create Appointment Date field in Appointment object) .
9. Filter type should be Required.
10. Error Message : Value does not match the criteria.
11. Enable the filter by click on Active.
12. Next >> Next >> Save.

### **3. Creation of Lookup Field on Billing details and feedback Object :**

1. Go to setup >> click on Object Manager >> type object name( Billing details and feedback ) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Service records” and click next.
5. Next >> Next >> Save & new.

## **Creation Of Checkbox Fields**

### **1.Creation of Checkbox Field on Appointment Object :**

1. Go to setup >> click on Object Manager >> type object name( Appointment ) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Check box” as data type and click Next.
4. Give the Field Label : Maintenance service
5. Field Name : is auto populated
6. Default value : unchecked
7. Click on next >> next >> save.

### **2.Creation of Another Checkbox Field on Appointment Object:**

1. Repeat the steps form 1 to 3.
2. Give the Field Label : Repairs
3. Field Name : is auto populated
4. Default value : unchecked
5. Click on next >> next >> save.
6. Follow the same and create another checkbox with given names

7. Give the Field Label : Replacement Parts
8. Field Name : is auto populated
9. Default value : unchecked
10. Click on next >> next >> save.

### **3.Creation of Checkbox Field on Service records Object :**

1. Go to setup >> click on Object Manager >> type object name( Service records ) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Check box” as data type and click Next.
4. Give the Field Label : Quality Check Status
5. Field Name : is auto populated
6. Default value : unchecked
7. Click on next >> next >> save

#### **Note:**

(Creation Of Date Fields in Appointment Object

Creation Of Currency Fields in appointment And bill feedback object

Creation Of Picklist Fields in Service and bill feedback object

Creating Formula Field In Service Records Object)

Do this same as the above step just choose the appropriate data type of fields

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Customer Name	Name	Text(80)		✓
Gmail	Gmail_c	Email		
Last Modified By	LastModifiedBy	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone number	Phone_number_c	Phone		

**Appointment**

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date__c	Date		
Appointment Name	Name	Auto Number		✓
Created By	CreatedBy	Lookup(User)		
Customer Details	Customer_Details__c	Lookup(Customer Details)		✓
Last Modified By	LastModifiedBy	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)		

**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		

The screenshot shows the Salesforce Setup interface. The top navigation bar includes tabs for Customer review, Recently Viewed, and Billing details and feedback. The main content area is titled "Billing details and feedback" under "OBJECT MANAGER". On the left, a sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Triggers. The "Fields & Relationships" section is currently selected. A table titled "Fields & Relationships" lists eight items, sorted by Field Label. The columns include FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The fields listed are:

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) (Unique Case Insensitive)		✓
Service records	Service_records__c	Lookup(Service records)		✓

## Task 5 (Validation Rule Creation)

### 1. To Create A Validation Rule To An Appointment Object

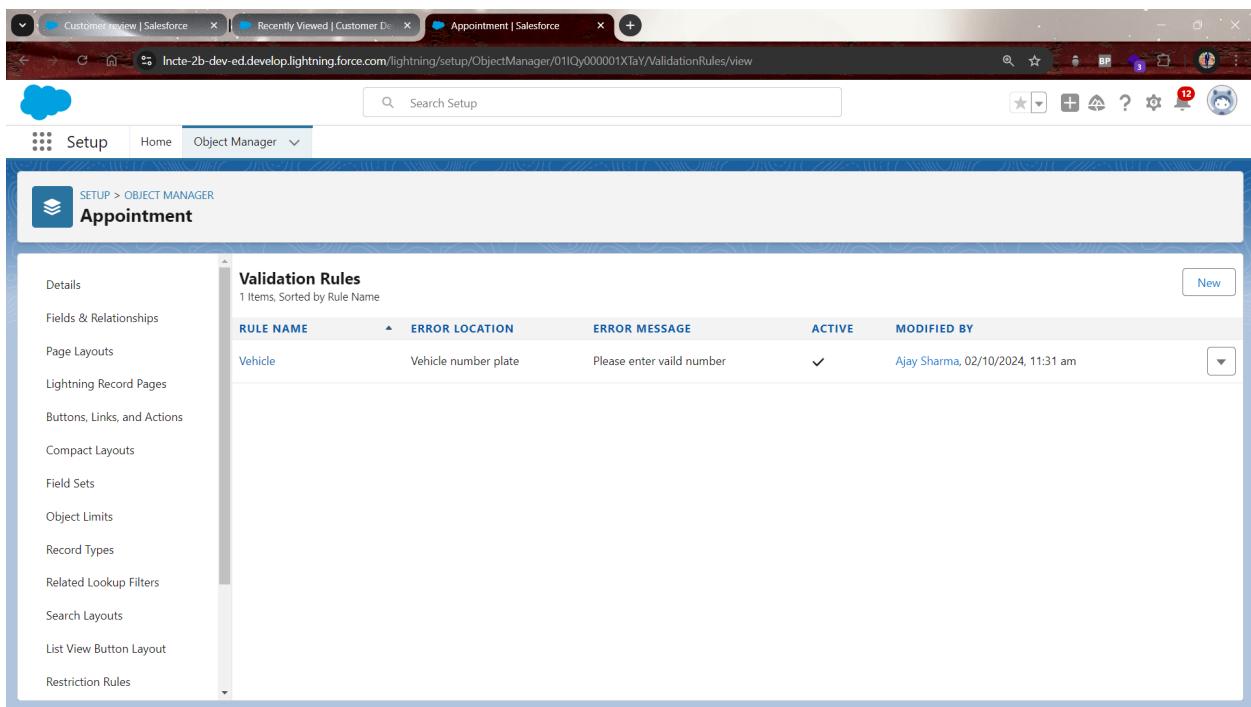
1. Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ Vehicle ”.
4. Insert the Error Condition Formula as : -  
NOT(REGEX( Vehicle\_number\_plate\_\_c , “[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}”))
5. Enter the Error Message as “Please enter valid number ”, select the Error location as Field and select the field as “Vehicle number plate”, and click Save.

### 2. To Create A Validation Rule To An Service Records Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Service records object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ service\_status\_note ”.
4. Insert the Error Condition Formula as : -  
NOT( ISPICKVAL( Service\_Status\_\_c , "Completed"))
5. Enter the Error Message as “still it is pending”, select the Error location as Field and select the field as “Service status”, and click Save.

### 3. To Create A Validation Rule To An Billing Details And Feedback Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Billing details and feedback object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ rating\_should\_be\_less\_than\_5”.
4. Insert the Error Condition Formula as :-  
`NOT( REGEX( Rating_for_service__c , "[1-5]{1}" ))`
5. Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Rating for Service”, and click Save.



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

- Header:** Customer review | Salesforce, Recently Viewed | Customer De..., Appointment | Salesforce
- Page:** Validation Rules (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	✓	Ajay Sharma, 02/10/2024, 11:31 am

Customer review | Salesforce | Recently Viewed | Customer De | Service records | Salesforce

Incte-2b-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lQy000001XYDd/ValidationRules/view

The screenshot shows the Salesforce Setup interface for the 'Service records' object. The left sidebar lists various setup categories like Details, Fields & Relationships, Page Layouts, etc. The main content area is titled 'Validation Rules' and shows one item: 'service\_status\_note'. The table columns are RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The entry details are: Rule Name: service\_status\_note, Error Location: Service Status, Error Message: still it is pending, Active: checked, Modified By: Ajay Sharma, 02/10/2024, 11:33 am.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
service_status_note	Service Status	still it is pending	✓	Ajay Sharma, 02/10/2024, 11:33 am

Customer review | Salesforce | Recently Viewed | Customer De | Billing details and feedback | Salesforce

Incte-2b-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01lQy000001XYLh/ValidationRules/view

The screenshot shows the Salesforce Setup interface for the 'Billing details and feedback' object. The left sidebar lists various setup categories. The main content area is titled 'Validation Rules' and shows one item: 'rating\_should\_be\_less\_than\_5'. The table columns are RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The entry details are: Rule Name: rating\_should\_be\_less\_than\_5, Error Location: Rating for service, Error Message: rating should be from 1 to 5, Active: checked, Modified By: Ajay Sharma, 02/10/2024, 11:35 am.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
rating_should_be_less_than_5	Rating for service	rating should be from 1 to 5	✓	Ajay Sharma, 02/10/2024, 11:35 am

## **Task 6 (Duplicate Rule Creation)**

### **1. To Create A Matching Rule To An Customer Details Object**

1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.
3. Select the object as Customer details and click Next.
4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7. 

Field	Matching Method
1. Gmail	Exact
2. Phone Number	Exact
8. Click save.
9. After Saving Click on Activate.

### **2. To Create A Duplicate Rule To An Customer Details Object**

1. Go to quick find box in setup and search for Duplicate rules.
2. Click on Duplicate rule >> click on New Rule >> select customer details object.
3. Give the Rule name as : Customer Detail duplicate
4. Scroll a little in Matching rule section
5. Select the matching rule : Matching customer details
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.

## Task 7 (Profiles Creation)

### Manager Profile

#### 1. To create a new profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Garage management.
4. Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.
5. Changing the session times out after should be “ 8 hours of inactivity”.
6. Change the password policies as mentioned :
7. User passwords expire in should be “ never expires ”.
8. Minimum password length should be “ 8 ”, and click save.

The screenshot shows the Salesforce Setup interface with the 'Profiles' page open. The left sidebar navigation includes 'Setup Home', 'Service Setup Assistant', 'Commerce Setup Assistant', 'Hyperforce Assistant', 'Release Updates', 'Lightning Experience Transition Assistant', 'Salesforce Mobile App', 'Optimizer', and sections for 'Users', 'Permission Set Groups', 'Permission Sets', 'Profiles' (which is selected), 'Public Groups', 'Data', 'Conversation Transcript Export', 'Data Export', 'Duplicate Management', 'Duplicate Error Logs', 'Duplicate Rules', 'Matching Rules', 'Mass Transfer Approval Requests', and 'Object Manager'. The main content area displays the 'Profile Detail' for the 'Manager' profile, which is a clone of the 'Standard User'. It shows the profile name, user license (Salesforce), description, creation date (02/10/2024, 11:43 am), and modification date (02/10/2024, 11:47 am). Below this, the 'Page Layouts' section lists standard object layouts for various objects like Global, Email Application, Home Page Layout, Account, Alternative Payment Method, Appointment Invitation, Asset, Asset Action, and Asset Action Source, along with their respective global and object-specific page layouts. A large table at the bottom maps these layouts to location group assignment, macro, object milestone, operating hours, opportunity, opportunity product, order, order product, and payment objects. The URL in the browser bar is 'https://incte-2b-dev-ed.lightning.force.com/lightning/setup/EnhancedProfiles/page?address=%2F00eQy000007csD3'.

## 2. Sales Person Profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Salesforce Platform User) >> enter profile name (sales person) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the Garage management.
4. Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.
5. And click save.

The screenshot shows the Salesforce Setup interface with the 'Profiles' page open. The profile being edited is named 'sales person'. The 'Profile Detail' section shows the user license as 'Salesforce Platform' and the creation date as '02/10/2024, 11:50 am'. The 'Custom Profile' checkbox is checked. The 'Page Layouts' section lists various standard object layouts such as Global, Email Application, Home Page Layout, Account, Alternative Payment Method, Appointment Invitation, Asset, Asset Relationship, and Assigned Resource. Each layout is associated with specific assignment links. The 'Location Group Assignment' section shows assignments for Lead, Location, Location Group, Object Milestone, Operating Hours, Order, Order Product, and Payment. The sidebar on the left shows navigation links for Setup Home, Service Setup Assistant, Commerce Setup Assistant, Hyperforce Assistant, Release Updates, Lightning Experience Transition Assistant, Salesforce Mobile App, Optimizer, Users, Permission Set Groups, and Data. The 'Profiles' link under 'Permission Set Groups' is currently selected.

## Task 8 (Role & Role Hierarchy Creation)

### 1. Creating Manager Role

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click on Expand All and click on add role under whom this role works.
3. Give Label as “Manager” and Role name gets auto populated. Then click on Save.

The screenshot shows the Salesforce Setup Roles page. In the left sidebar, under 'Users' > 'Roles', the 'Manager' role is selected. The main content area displays the 'Role Detail' for the 'Manager' role. The 'Label' field is set to 'Manager'. The 'Role Name' field is also 'Manager'. The 'Sharing Groups' section includes 'Role, Role and Internal Subordinates'. The 'Users in Manager Role' section lists one user: 'Abir Shama' (Full Name: Abir Shama, Alias: abir, Username: abir@inst.com, Active: checked). The status bar at the bottom indicates 'Role: Manager - Salesforce - Developer Edition'.

### 2. Creating Sales Person Roles

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role, and click add role under manager.
3. Give Label as “sales person” and Role name gets auto populated. Then click on Save.

The screenshot shows the Salesforce Setup Roles page. In the left sidebar, under 'Users' > 'Roles', the 'sales person' role is selected. The main content area displays the 'Role Detail' for the 'sales person' role. The 'Label' field is set to 'sales person'. The 'Role Name' field is also 'sales\_person'. The 'Sharing Groups' section includes 'Role, Role and Internal Subordinates'. The 'Users in sales person Role' section lists three users: 'Himash Beshamya' (Full Name: Himash Beshamya, Alias: himesh, Username: himesh@inst.com, Active: checked), 'Avantika' (Full Name: Avantika, Alias: avantika, Username: avantika@inst.com, Active: checked), and 'Vimal Sheraf' (Full Name: Vimal Sheraf, Alias: vimal, Username: vimal@inst.com, Active: checked). The status bar at the bottom indicates 'Role: sales person - Salesforce - Developer Edition'.

## Task 9 (Users Creation)

### 1.Create User(Manager Role):

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. Fill in the fields
3. First Name : Niklaus
4. Last Name : Mikaelson
5. Alias : Give a Alias Name
6. Email id : Give your Personal Email id
7. Username : Username should be in this form: text@text.text
8. Nick Name : Give a Nickname
9. Role : Manager
10. User licence : Salesforce
11. Profiles : Manager ->Save.

### 2..Creating Another Users(Sales Role):

1. Repeat the steps and create another user using

1. Role : sales person
2. User licence : Salesforce Platform
3. Profile : sales person

The screenshot shows the Salesforce Setup interface with the 'Users' tab selected. The main area displays a table of users with columns for Action, Full Name, Alias, Username, Role, Active, and Profile. The table includes rows for Chatter Expert, Hema Arun, Mikaelson Niklaus, Rehamniva Himesh, Sharma Ajay, Sheerf Vimal, User Integration, and User Security. The 'Active' column shows checkboxes for each user, and the 'Profile' column indicates the profile assigned to each user. The sidebar on the left shows the 'User Management Settings' section selected under 'User'. The top navigation bar shows the URL as 'https://incte-2b-dev.develop.lightning.force.com/lightning/setup/ManageUsers/home'.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/>	Chatter Expert	Chatter	chatty.00dov00000cumvhmai5v5lrbccy@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/>	Hema Arun	akami	akuno@inct.com		<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/>	Mikaelson Niklaus	nmika	niklaus@inct.com	Manager	<input checked="" type="checkbox"/>	Manager
<input type="checkbox"/>	Rehamniva Himesh	hresh	himesh@inct.com		<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/>	Sharma Ajay	AShar	sharma@ajay1604.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/>	Sheerf Vimal	vshar	vimal@inct.com		<input checked="" type="checkbox"/>	sales person
<input type="checkbox"/>	User Integration	integ	integration@00dov00000cumvhmai.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/>	User Security	se	inightssecurity@00dov00000cumvhmai.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

## Task 10 (Public Groups Creation)

**Public Groups** in Salesforce are collections of users that can be defined for sharing rules, access permissions, or collaboration purposes. These groups can include individual users, roles, or other groups, making them versatile for managing access and data visibility across the organization.

### Purpose of Public Groups:

- Access Control:** Public Groups allow administrators to define who has access to certain records or data. For example, you can use Public Groups to grant access to certain objects, records, or folders to a specific group of users.
- Sharing Rules:** Public Groups are often used in sharing rules to control data visibility. By assigning records to a Public Group, you can ensure that only members of the group have access to those records, without the need to individually configure permissions for each user.
- Collaboration:** Public Groups can be used to facilitate collaboration among users. For instance, a Public Group could be created for a specific department or project team, allowing members to easily share files, Chatter posts, and other resources.

### Creating New Public Group

- Go to setup >> type users in quick find box >> select public groups >> click New.
- Give the Label as “sales team”.
- Group name is autopopulated.
- Search for Roles.
- In Available Members select Sales person and click on add it will be moved to selected member.
- Click on save.

The screenshot shows the Salesforce Setup interface. The left sidebar is collapsed, and the main area displays the 'Public Groups' page under the 'SETUP' tab. The search bar at the top contains 'public'. The page title is 'Public Groups'. A sub-header 'Public Groups' is present with a help link. Below it, a description states: 'A public group is a set of users. It can contain individual users, other groups, the users in a particular role or territory, or the users in a role or territory plus all of the users below that role or territory in the hierarchy.' There are buttons for 'View: All' (selected), 'Edit', and 'Create New View'. A navigation bar at the bottom includes links for A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, and Other. A table lists one group entry: Action (Edit | Del), Label (sales\_team), Group Name (sales\_team), Created By (Sharma\_Ajay), and Created Date (02/10/2024, 12:13 pm). At the bottom left, there's a note: 'Didn't find what you're looking for? Try using Global Search.'

## **Task 11 (Sharing Setting Creation)**

**Sharing Settings** in Salesforce are configurations that determine the level of access users have to data across the organization. They control how records are shared among users and help define the default access level for each object in Salesforce. Sharing Settings are essential for balancing data security with collaboration needs.

### **Creating Sharing Settings**

1. Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.
2. Change the OWD setting of the Service records Object to private as shown in fig.
3. Click on save and refresh.
4. Scroll down a bit, Click new on Service records sharing Rules.
5. Give the Label name as “ Sharing setting”
6. Rule name is auto populated.
7. In step 3 : Select which records to be shared, members of “ Roles ” >> “ Sales person”
8. In step 4: share with, select “ Roles ” >> “ Manager ”
9. In step 5 : Change the access level to “ Read / write ”.
10. Click on save.

## Task 12 (Flows Creation)

**Flows in Salesforce** are a powerful tool for automating complex business processes through a visual interface. They allow administrators and developers to build custom workflows without needing to write code, making it easier to implement automation that meets specific business requirements.

### Create A Flow:

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.
2. Select the Record-triggered flow and Click on Create.
3. Select the Object as “Billing details and feedback” in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.
6. Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.
7. Give the Label Name : Amount Update
8. Api name : is auto populated
9. Set a filter condition : All Conditions are met(AND)
10. Field : Payment\_Status\_\_c
11. Operator : Equals
12. Value : Completed
13. And Set Field Values for the Billing details and feedback Record
14. Field : Payment\_Paid\_\_c
15. Value : {!\$Record.Service\_records\_\_r.Appointment\_\_r.Service\_Amount\_\_c}
16. Click On Done.
17. Before creating another Element. Create a New Resource form Toolbox form top left.
18. Click on the New Resource, And select Variable.
19. Select the resource type as text template.
20. Enter the API name as “ alert”.
21. Change the view as Rich Text ? View to Plain Text.
22. In body field paste the syntax that given below.

Dear {!\$Record.Service\_records\_\_r.Appointment\_\_r.Customer\_Name\_\_r.Name},

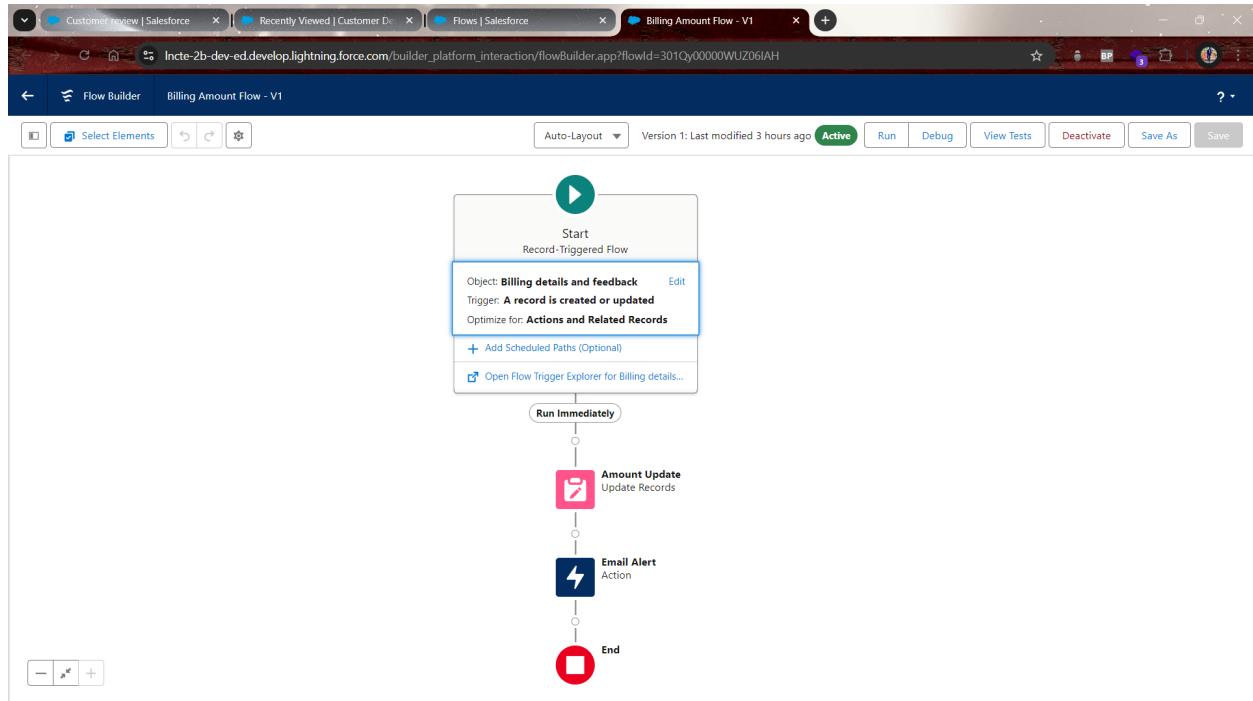
I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is greatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {!\$Record.Payment\_Paid\_\_c}

Thank you for Coming .

23. Click done.
24. Now Click on Add Element , select Action.
25. Their action bar will be opened in that search for “ send email ” and click on it.
26. Give the label name as “ Email Alert”
27. API name will be auto populated.

28. Enable the body in set input values for the selected action.
29. Select the text template that created , Body : {!alert}
30. Include recipient address list select the email form the record.
31. RecipientAddressList: {!\$Record.Service\_records\_\_r.Appointment\_\_r.Customer\_Name\_\_r.Gmail\_\_c}
32. Include subject as “ Thank You for Your Payment - Garage Management”.
33. Click done.
34. Click on save. Give the Flow label , Flow Api name will be autopopulated.
35. And click save, and click on activate.



## Task 13 (Apex Trigger Creation)

**Apex Triggers** in Salesforce are pieces of code that run automatically in response to specific events on a particular Salesforce object. These events include actions like creating, updating, deleting, or undeleting records. Triggers are used to perform operations before or after these events, allowing developers to customize and automate business processes in ways that standard configurations and workflows can't achieve.

### Apex Handler

UseCase : This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as “AmountDistributionHandler”.

Code:

```
public class AmountDistributionHandler {  
  
    public static void amountDist(list<Appointment__c> listApp){  
        list<Service_records__c> serList = new list <Service_records__c>();  
  
        for(Appointment__c app : listApp){  
            if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){  
                app.Service_Amount__c = 10000;  
            }  
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){  
                app.Service_Amount__c = 5000;  
            }  
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){  
                app.Service_Amount__c = 8000;  
            }  
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){  
                app.Service_Amount__c = 7000;  
            }  
            else if(app.Maintenance_service__c == true){  
                app.Service_Amount__c = 2000;  
            }  
            else if(app.Repairs__c == true){  
                app.Service_Amount__c = 3000;  
            }  
            else if(app.Replacement_Parts__c == true){  
                app.Service_Amount__c = 5000;  
            }  
        }  
    }  
}
```

Trigger Handler :

How to create a new trigger :

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment\_\_c

Syntax For creating trigger :

The syntax for creating trigger is :

```
Trigger [trigger name] on [object name]( Before/After event)
{
```

```
}
```

In this project , trigger is called whenever the particular records sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

1. Handler for the Appointment Object

Code:

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

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

    }
}
```

## **Task 14 (Reports Creation)**

### **Reports**

Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

1. Tabular
2. Summary
3. Matrix
4. Joined Reports

### **Create A Report Folder**

1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.
3. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.
4. Click save.

### **Sharing A Report Folder**

1. Go to the app >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles” , in name field search for “manager” , give “view” as access for that role.
4. Then click share, and click on Done.

### **Create Report Type**

1. Go to setup >> type users in quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.
3. Select the Primary object as “ Customer details” .
4. Give the Report type Label as “ Service information ”
5. Report type Name is autopopulated.
6. Keep the Description as same.
7. Select Store in Category as “ other Reports ”
8. Select the deployment status as “ Deployed ”, click on Next.
9. now , Click on Related object box.
10. Click on Select Object, choose Appointment Object a
11. Again Click to relate another object.
12. And select the related object as “ service records”.
13. Repeat the process and select the related object as “ Billing details and feedback”.
14. And click on save.

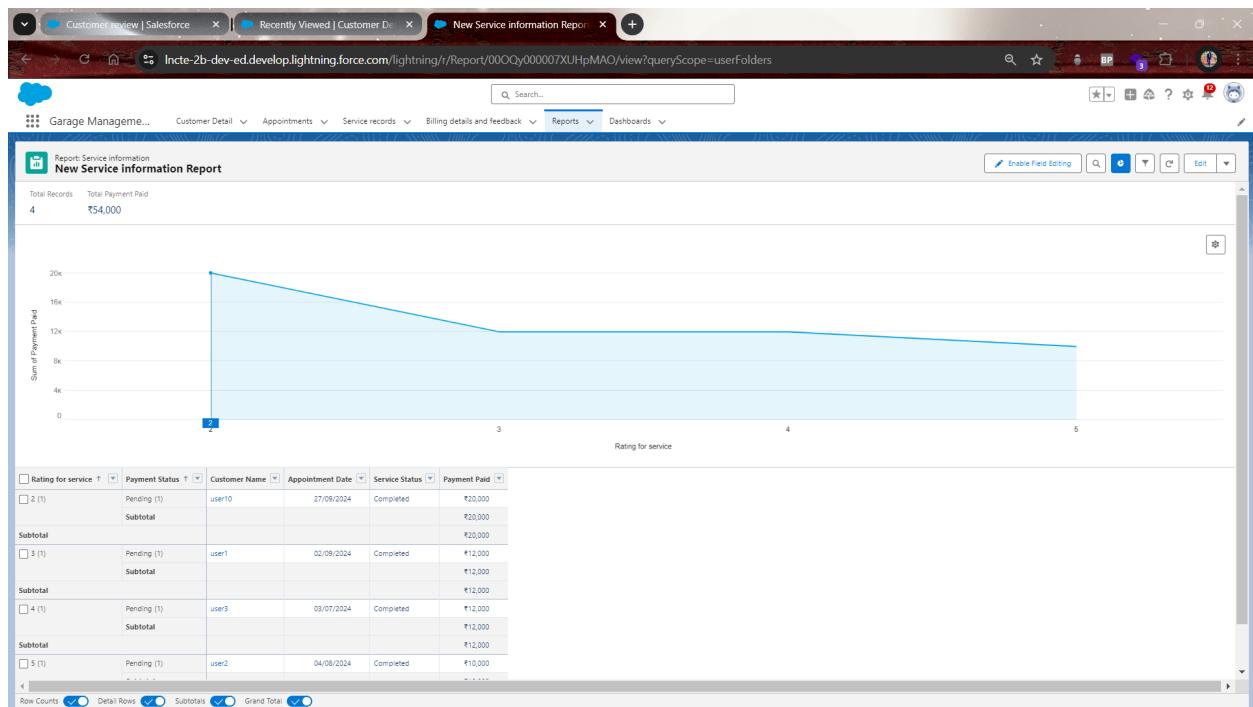
### **Create Report**

Note : Before creating report, create latest “10” records in every object.

Try to fill every field in each record for better experience.

1. Go to the app >> click on the reports tab

2. Click New Report.
3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.
4. Their outline pane is opened already, select the fields that mentioned below in column section.
  1. Customer name
  2. Appointment Date
  3. Service Status
  4. Payment paid
2. Remove the unnecessary fields.
3. Select the fields that mentioned below in GROUP ROWS section.
  1. Rating for Service
7. Select the fields that mentioned below in GROUP ROWS section.
  1. Payment Status
8. Click on Add Chart , Select the Line Chart.
9. Click on save, Give the report Name : New Service information Report
10. Report unique Name is auto populated.
11. Select the folder the created and Click on save



## Task 15 (Dashboards Creation)

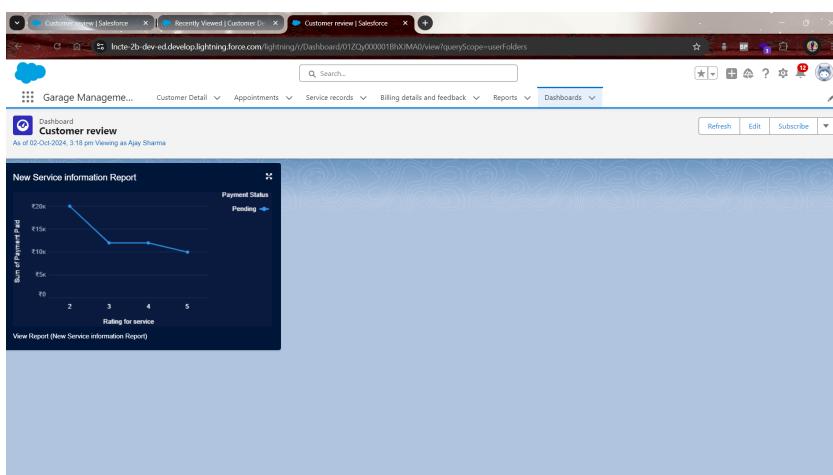
**Dashboards** help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

### Create Dashboard Folder

1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “ Service Rating dashboard”.
4. Folder unique name will be auto populated.
5. Click save.
6. Follow the same steps, from milestone 15, and activity 2, and provide the sharing settings for the folder that just created.

### Reate Dashboard

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.
3. Select add component.
4. Select a Report and click on select.
5. Select the Line Chart. Change the theme.
6. Click Add then click on Save and then click on Done.
7. Preview is shown below.
8. After that Click on Subscribe on top right.
9. Set the Frequency as “ weekly ”.
10. Set a day as monday.
11. And Click on save.



**THANKYOU**