

ASIAN
College of Engineering and Technology

Asian College Road, Kondayampalayam, Near Saravanampatti, Coimbatore 641 110

Department of
Computer science and Engineering

LABORATORY RECORD

Name : _____ University Reg.No : _____
Class : _____ Branch/Sem : _____

Certified bonafide Record of work done by _____
in _____ Laboratory.

Place : _____ *Staff Incharge* _____ *Head of the department* _____
Date: _____

Submitted for the University Practical Examination held on _____

Internal Examiner

External Examiner

Abstract :

The Garage Management System (GMS) is a comprehensive solution designed to streamline operations within an automotive garage, enhancing efficiency, and improving customer service. This system facilitates the management of various processes, including vehicle registration, service booking, inventory control, customer management, and billing. By automating these key functions, the GMS reduces manual errors, minimizes time spent on administrative tasks, and provides a centralized platform for tracking all garage activities.

The system allows garage staff to easily register new vehicles, schedule service appointments, and manage repair and maintenance tasks. It includes inventory management capabilities to monitor parts availability, ensuring the timely procurement of necessary supplies. The GMS also supports customer relationship management (CRM), storing customer profiles and service history, which helps in building long-term client relationships through personalized communication and service recommendations.

In addition to operational management, the GMS generates detailed reports on daily activities, financial transactions, and inventory status, enabling managers to make informed decisions and enhance business performance. The system is user-friendly, secure, and scalable, making it suitable for garages of various sizes. Overall, the Garage Management System provides an effective digital solution to the challenges faced by automotive service providers, promoting productivity and customer satisfaction.

Index for Garage Management System

1. Introduction

- **1.1 Overview of the System**
- **1.2 Objectives of the System**
- **1.3 Scope of the System**
- **1.4 Definitions, Acronyms, and Abbreviations**

2. System Requirements

- **2.1 Functional Requirements**
 - **2.1.1 Vehicle Registration and Management**
 - **2.1.2 Service Booking and Scheduling**
 - **2.1.3 Parts and Inventory Management**
 - **2.1.4 Staff Management**
 - **2.1.5 Customer Management**
 - **2.1.6 Billing and Payment**
 - **2.1.7 Report Generation**
- **2.2 Non-Functional Requirements**
 - **2.2.1 Usability**
 - **2.2.2 Performance**
 - **2.2.3 Security**
 - **2.2.4 Scalability**
- **2.3 Hardware and Software Requirements**

3. System Design

- **3.1 Architectural Design**

- **3.1.1 System Architecture Overview**
- **3.1.2 Database Design**
- **3.1.3 Component Design**
- **3.2 Data Flow Diagrams (DFDs)**
 - **3.2.1 Context Diagram**
 - **3.2.2 Level 1 DFD**
 - **3.2.3 Level 2 DFD**
- **3.3 User Interface Design**
 - **3.3.1 Login and Dashboard**
 - **3.3.2 Customer and Vehicle Management**
 - **3.3.3 Service Booking and Scheduling**
 - **3.3.4 Billing and Reports**

4. Implementation

- **4.1 Development Environment and Tools**
- **4.2 Coding Standards and Practices**
- **4.3 System Modules and Classes**
- **4.4 Database Schema**
- **4.5 Integration and Testing**
 - **4.5.1 Unit Testing**
 - **4.5.2 Integration Testing**
 - **4.5.3 System Testing**
 - **4.5.4 User Acceptance Testing (UAT)**

5. Testing and Quality Assurance

- **5.1 Testing Methodologies**

- **5.2 Test Cases and Scenarios**
- **5.3 Bug Tracking and Resolution**
- **5.4 Performance Testing**
- **5.5 Security Testing**

6. User Manual

- **6.1 Getting Started**
 - **6.1.1 Logging In and Registration**
- **6.2 Managing Vehicles**
- **6.3 Booking Services**
- **6.4 Billing and Payments**
- **6.5 Reports and Analytics**
- **6.6 Logout and System Navigation Tips**

7. System Maintenance and Support

- **7.1 Maintenance Plan**
- **7.2 Data Backup and Recovery**
- **7.3 Software Updates and Patches**
- **7.4 Troubleshooting Common Issues**

8. Appendices

- **8.1 Glossary of Terms**
- **8.2 Acronyms**
- **8.3 References**
- **8.4 Change Log / Version History**

Salesforce

Introduction:

Are you new to Salesforce? Not sure exactly what it is, or how to use it? Don't know where you should start on your learning journey? If you've answered yes to any of these questions, then you're in the right place. This module is for you.

Welcome to Salesforce! Salesforce is game-changing technology, with a host of productivity-boosting features, that will help you sell smarter and faster. As you work toward your badge for this module, we'll take you through these features and answer the question, "What is Salesforce, anyway?".

What Is Salesforce?

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers.

Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud.

So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this:

<https://youtu.be/r9EX3lGde5k>

Requirements :

These are the core functionalities that the system must perform.

1.1 Vehicle Management

- Ability to register and store vehicle details, including make, model, year, VIN (Vehicle Identification Number), and customer information.
- Maintain service history for each vehicle.

1.2 Customer Management

- Register and store customer details (name, contact info, vehicle ownership).
- Link customers to their respective vehicles and service records.
- Enable search for customers based on name, contact, or vehicle.

1.3 Service Booking and Scheduling

- Allow customers to book appointments for services and repairs.
- Schedule appointments, track service status, and notify customers of service completion.
- Display available times, mechanics, and bays for booking.

1.4 Inventory and Parts Management

- Track parts and inventory levels, including stock quantities and costs.
- Update inventory when parts are used in repairs or maintenance.
- Provide alerts or notifications for low inventory levels.

1.5 Staff Management

- Maintain staff records, including mechanics and other staff details.
- Track mechanic availability, workload, and assigned tasks.
- Allow assignment of specific mechanics to service bookings.

1.6 Billing and Payment Processing

- Generate bills for services rendered, including parts, labor, and taxes.

- Track payment statuses and record completed transactions.
- Provide invoices to customers in printable and email formats.

1.7 Reporting and Analytics

- Generate reports on various metrics, including service types, revenue, inventory usage, and customer visits.
- Provide data visualizations to help analyze business performance.
- Track and report on inventory, service bookings, and customer interactions over time.

1.8 Notifications and Alerts

- Send notifications for service updates, reminders, and promotional offers.
- Notify staff of scheduled appointments, low stock levels, and upcoming tasks.
- Allow both SMS and email notification options

2. Non-Functional Requirements

These are the quality attributes of the system, covering performance, security, usability, and more.

2.1 Usability

- User-friendly interfaces, accessible and intuitive for both staff and customers.
- Simple navigation and clear labeling of functions for seamless user experience.

2.2 Performance

- System should handle multiple users (garage staff, customers) simultaneously without noticeable lag.
- Quick response times for searches, loading service history, and managing inventory.

2.3 Reliability and Availability

- The system should be available 99% of the time during business hours.

- Implement backup and recovery solutions to prevent data loss.

2.4 Security

- Secure login and authentication for all users (staff, managers).
- Role-based access control: Only authorized users can view or edit certain data.
- Secure data storage with encryption for sensitive data, such as customer details and payment records.

2.5 Scalability

- The system should be scalable to support additional garages, service bays, and staff as the business grows.
- Designed to accommodate growing databases without performance issues.

2.6 Data Integrity

- Ensure data consistency and integrity across the system, especially for inventory, billing, and service records.
- Implement input validation and error handling for data entry.

2.7 Maintainability

- Code should be modular and documented to allow for easy maintenance, updates, and upgrades.
- Clear error messages and logs for diagnosing system issues.

3. Hardware Requirements

3.1 Server Requirements

- Processor: Intel Xeon or equivalent, 2.4 GHz or higher
- RAM: Minimum 8GB (16GB recommended for larger databases)
- Storage: Minimum 500GB SSD (expandable based on data volume)
- Network: Gigabit Ethernet

3.2 Client Workstations (for staff)

- Processor: Intel i5 or equivalent
- RAM: Minimum 4GB
- Storage: 100GB or more
- Display: 1080p resolution
- Operating System: Windows 10, Mac OS, or Linux

3.3 Mobile Device (for customer interaction, if applicable)

- OS: iOS 11 or Android 8.0 and above
- Internet Access: WiFi or mobile data for accessing the system remotely

4. Software Requirements

4.1 Server Software

- Operating System: Windows Server, Linux (Ubuntu, CentOS)
- Database: MySQL, PostgreSQL, or MS SQL Server
- Web Server: Apache or NGINX (for web-based access)
- Programming Language: Java, PHP, or Python (based on team expertise)
- Frameworks: Laravel (PHP), Django (Python), or Spring (Java) for backend development

4.2 Client Software (Workstations)

- Browser: Chrome, Firefox, or Safari (for web-based interface)
- PDF Reader: Adobe Acrobat or any PDF viewer (for reports)
- Text Editor: Optional for note-taking (Notepad++, Atom, etc.)

4.3 Other Software (if required)

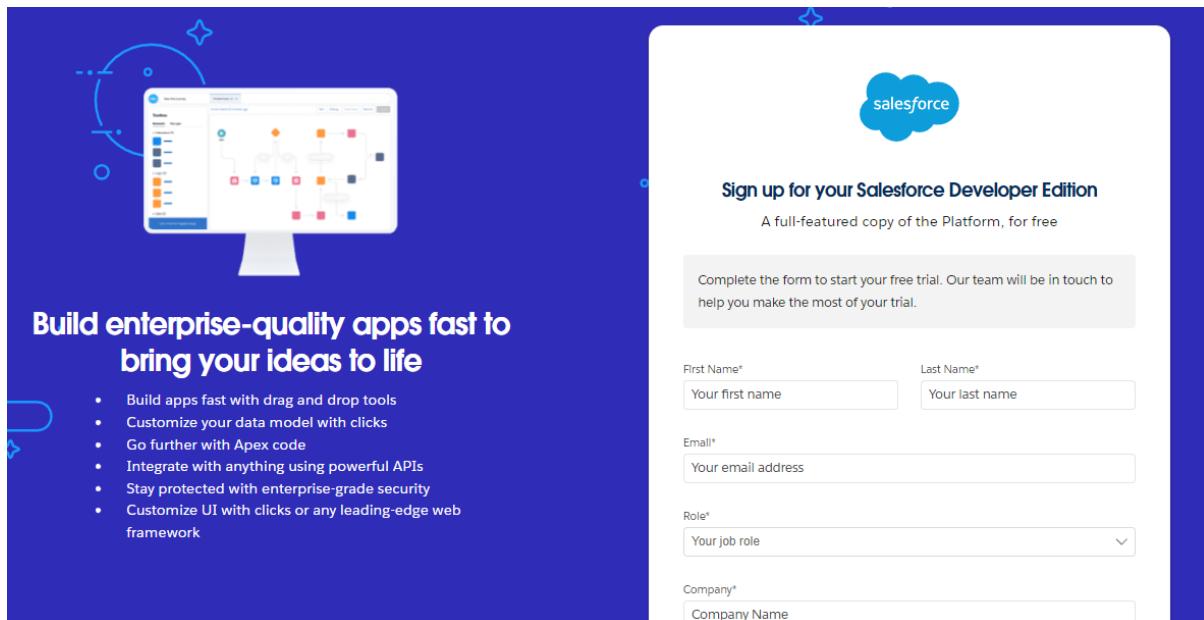
- Backup Solution: Automated backup software to secure data regularly.
- Anti-virus: Reliable anti-virus software for data security.
- Payment Gateway: If online payment is implemented, integration with a payment provider (e.g., Stripe, PayPal).

Project Description:

Creating a developer org in salesforce.

1. Go to <https://developer.salesforce.com/signup>

2. On the sign up form, enter the following details :



1. First name & Last name

2. Email

3. Role : Developer

4. Company : College Name

5. County : India

6. Postal Code : pin code

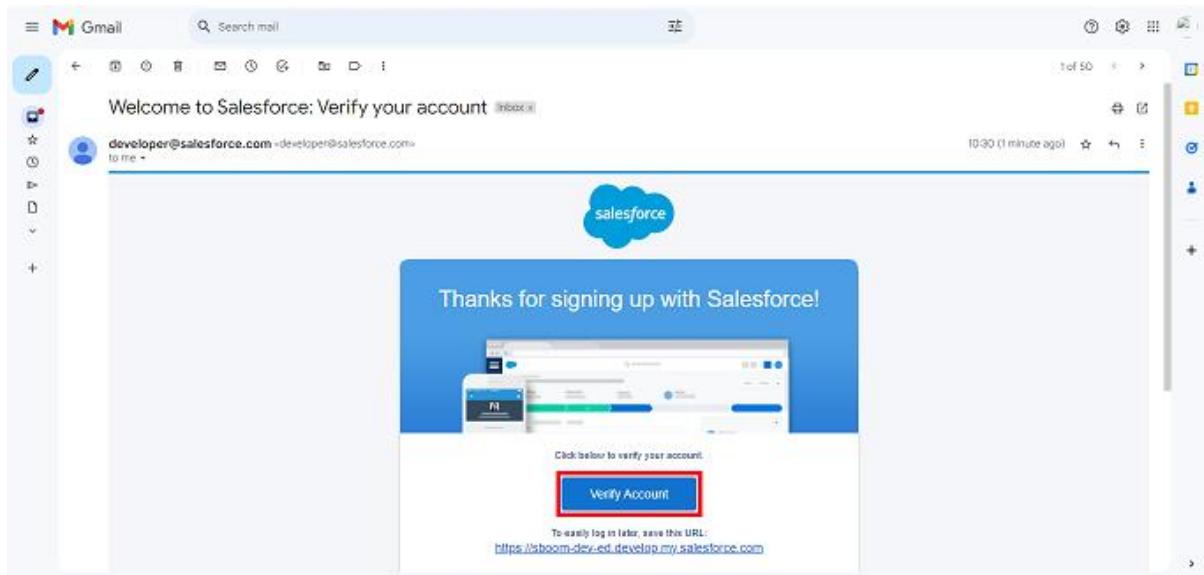
7. Username : should be a combination of your name and company

This need not be an actual email id, you can give anything in the format : username@organization.com

Click on sign me up after filling these.

Account Activation

1. Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.



Click on Verify Account

2. Give a password and answer a security question and click on change password.

Change Your Password

Enter a new password for lead@sb.oom.
Make sure to include at least:

- 8 characters
- 1 letter
- 1 number

* New Password
..... Good

* Confirm New Password
..... Match

Security Question
In what city were you born?

* Answer
asdfghjkl

Change Password

4. Then you will redirect to your salesforce setup page.

Setup Home

Service Setup Assistant

Multi-Factor Authentication Assistant

Release Updates

Lightning Experience Transition Assistant

Salesforce Mobile App

Lightning Usage

Optimizer

ADMINISTRATION

> Users

SETUP Home

Get Started with Einstein Bots

Launch an AI-powered bot to automate your digital connections.

Get Started

Mobile Publisher

Use the Mobile Publisher to create your own branded mobile app.

Learn More

Real-time Collaborative Docs

Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce.

Get Started

Create Customer DetailsObject

To create an object:

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

Create Appointment Object

To create an object:

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

Create Service records Object

To create an object:

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

Create Billing details and feedback Object

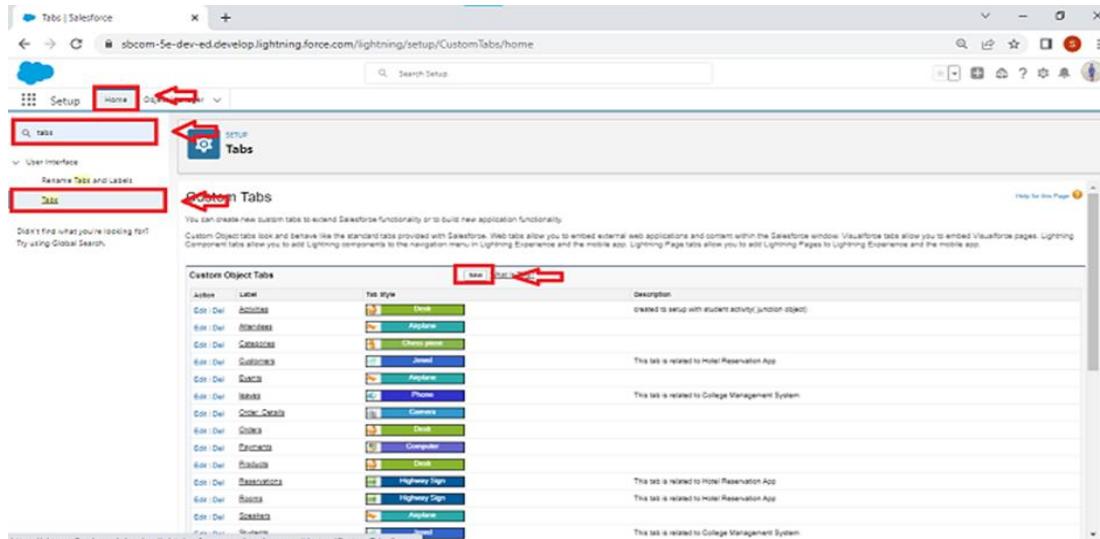
To create an object:

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

Creating a Custom Tab

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.

New Custom Object Tab

Help for this Page 

Step 1. Enter the Details

Step 1 of 3

Choose the custom object for this new custom tab. Fill in other details.

Select an existing custom object or [create a new custom object now](#).

Object 

Tab Style 

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link

Enter a short description:

Description

Tab Style Selector

Create your own style

[Hide styles which are used on other tabs](#)

	Airplane		Alarm clock		Apple		Balls
	Bank[1]		Bell		Big top		Boat[1]
	Books		Bottle		Box		Bridge
	Building		Building Block		Caduceus		Camera
	Can		Car		Castle		CD/DVD
	Cell phone		Chalkboard		Chess piece		Chip
	Circle		Compass		Computer		Credit card
	CRT TV		Cup		Desk[1]		Diamond
	Dice		Factory		Fan		Flag
	Form		Gears		Globe		Guitar
	Hammer		Hands		Handsaw		Headset
	Heart[1]		Helicopter		Hexagon		Highway Sign
	Hot Air Balloon		Insect		IP Phone		Jewel
	Keys		Laptop		Leaf		Lightning

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 .

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 .

Create a lightning app

To create a lightning app page:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on New lightning App.

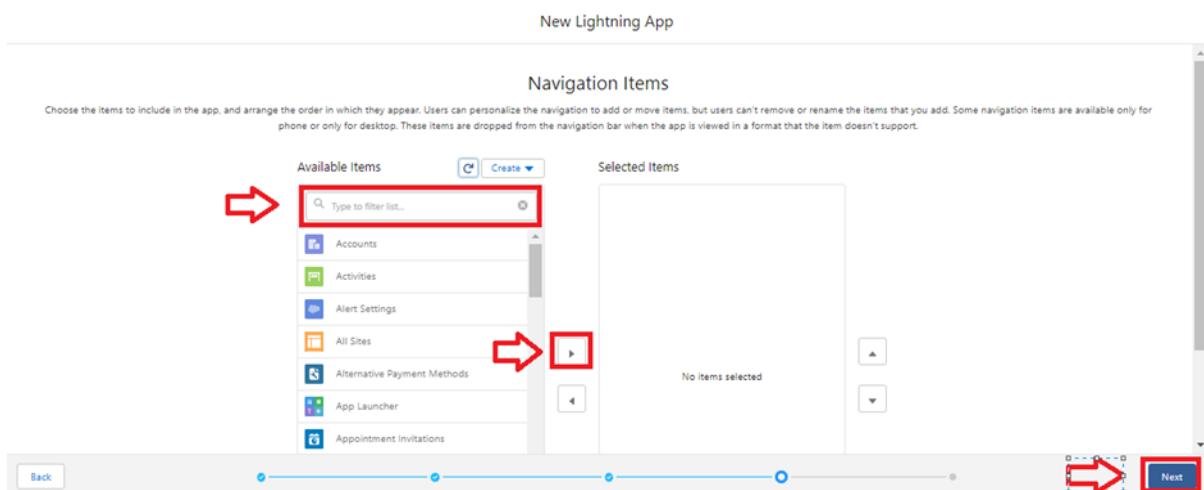
The screenshot shows the Salesforce App Manager interface. At the top, there are three search bars: 'app manager', 'Apps', and 'AppManager'. Below them is a section titled 'Lightning Experience App Manager' with a sub-section 'Clone Apps(Beta)'. A red arrow points to the 'New Lightning App' button in the top right corner of this section. The main area displays a table of existing apps, with a red arrow pointing to the 'Next' button at the bottom right of the table header.

App Name	Developer Name	Description	Last Modified	App Type	Versions
All Tabs	Lightning	Build CRM Analytics dashboards and apps	04/12/2022, 10:13 am	Classic	
Analytics Studio	Insights	Build CRM Analytics dashboards and apps	04/12/2022, 10:13 am	Classic	
App Launcher	AppLauncher	App Launcher tabs	04/12/2022, 10:13 am	Classic	
Built Solutions	LightningBox	Discover and manage business solutions designed for your industry	04/12/2022, 10:13 am	Lightning	
Chatter Desktop	Chatter/Desktop	Chatter Desktop is an Adobe AIR-based desktop application that lets Chatter users stay connected...	29/12/2022, 4:04 pm	Connected (Managed)	
Chatter Mobile for BlackBerry	ChatterForBlackBerry	The Salesforce.com Chatter Mobile app lets you access Chatter data on the go. Use it to view feed...	29/12/2022, 4:05 pm	Connected (Managed)	
College Management System	Academic	demo app	08/12/2022, 4:16 pm	Lightning	
Community	Community	Salesforce CRM Communities	04/12/2022, 10:13 am	Classic	
Content	Content	Salesforce CRM Content	04/12/2022, 10:13 am	Classic	
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage recipes.	04/12/2022, 10:13 am	Lightning	

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.

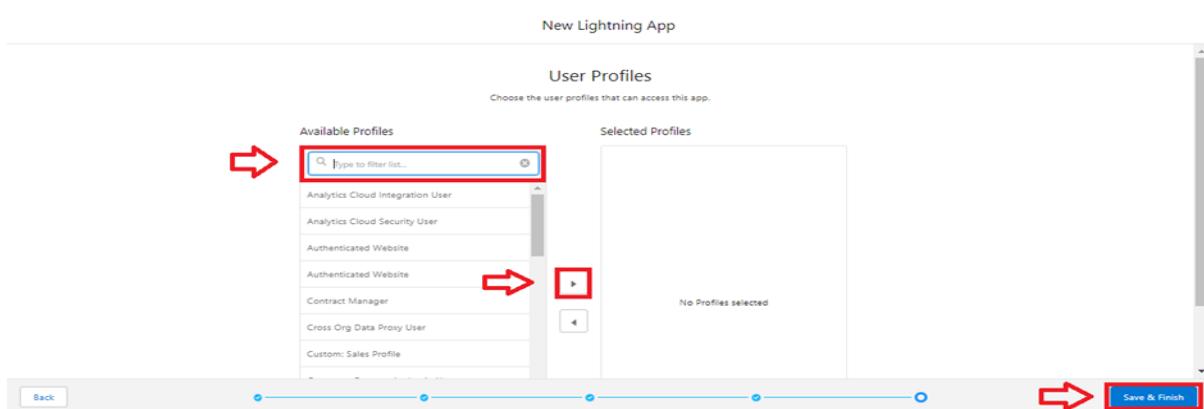
The screenshot shows the 'New Lightning App' configuration page. The 'App Details & Branding' section is displayed. In the 'App Details' tab, the 'App Name' field is highlighted with a red box and a red arrow pointing to it. In the 'App Branding' tab, the 'Primary Color Hex Value' field is also highlighted with a red box and a red arrow pointing to it. At the bottom right of the page, a large red arrow points to the 'Next' button.

3. To Add Navigation Items:



4. 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.

5. To Add User Profiles:



Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

Creation of fields for the Customer Details object

1. To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name(Customer Details) in search bar >> click on the object.

The screenshot shows the Salesforce Object Manager interface. At the top, there is a search bar with the text "cus", a "Schema Builder" button, and a "Create" button. Below the header, it says "2 Items. Sorted by Label". There are two rows in the main table:

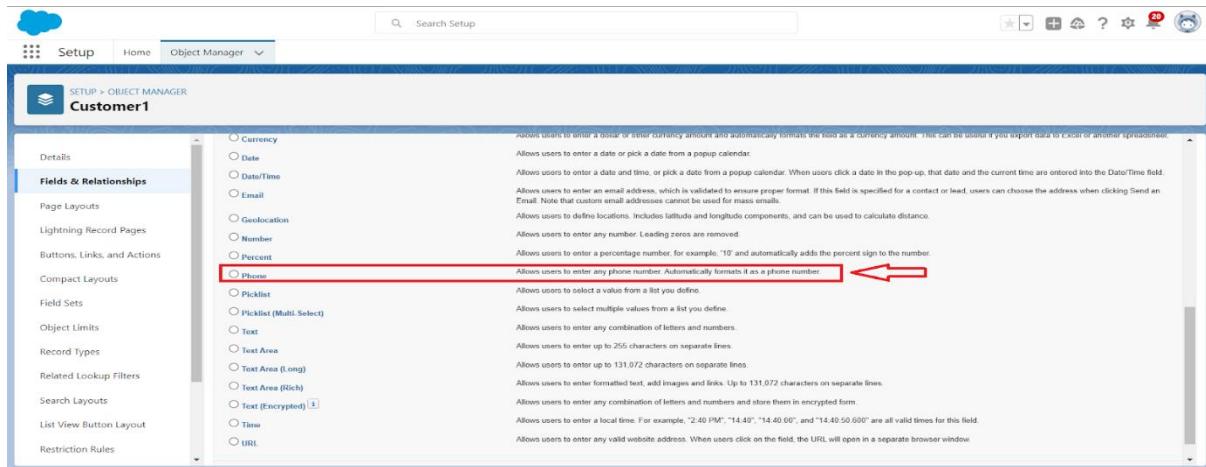
- Customer: API Name "Customer", Type "Standard Object".
- Customer Details: API Name "Customer_Details__c", Type "Custom Object". This row is highlighted with a red border.

2. Now click on “Fields & Relationships” >> New

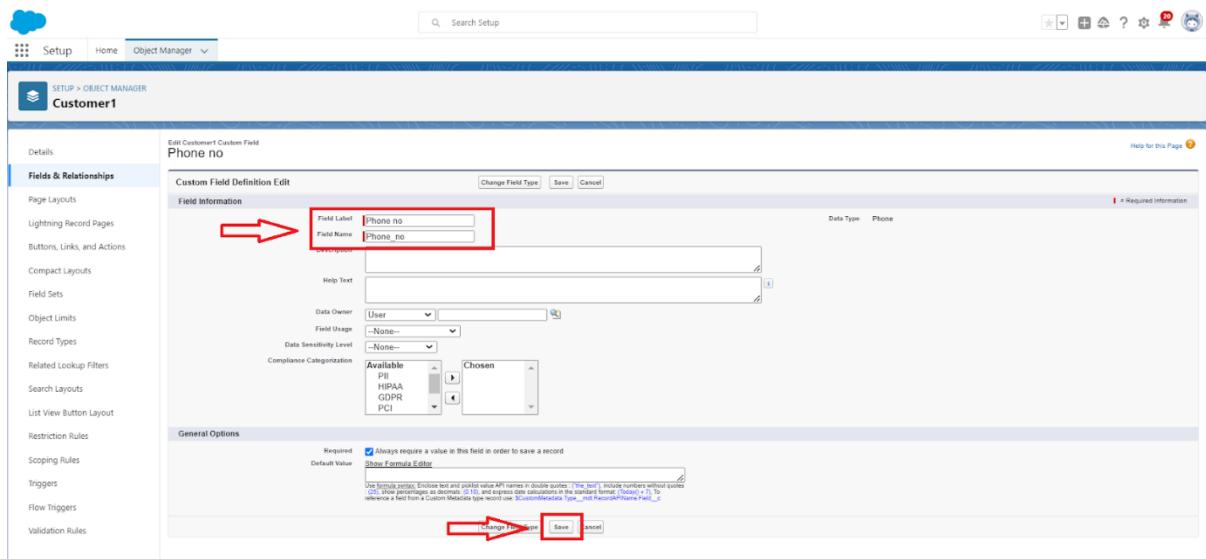
The screenshot shows the "Fields & Relationships" page for the "Customer1" object. At the top, there is a "Search Setup" bar and a toolbar with various icons. The left sidebar lists navigation options like Details, Fields & Relationships (which is selected and highlighted with a red box), Page Layouts, Lightning Record Pages, etc. The main content area is titled "Fields & Relationships" and shows a table of existing fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		✓
current Status	current_Status__c	Picklist		✓
Customer Name	Name	Text(80)		✓
Email Id	Email_Id__c	Email (Unique)		✓
Last Modified By	LastModifiedById	Lookup(User)		✓
Owner	OwnerId	Lookup(User,Group)		✓
Permanent Address	Permanent_Address__c	Text Area(255)		✓
Phone no	Phone_no__c	Phone		✓

3. Select Data Type as a “Phone”



4. Click on next.



5. Fill the Above as following:

- Field Label: Phone number
- Field Name : gets auto generated
- Click on Next >> Next >> Save and new.

Note: Follow the above steps for the remaining field for the same object.

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 Above as following:
 - Field Label : Gmail
 - Field Name : gets auto generated
 - Click on Next >> Next >> Save and new.

Creation of Lookup fields

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.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main area is titled 'Object Manager' with a sub-header '7 Items. Sorted by Label'. A table lists objects with columns: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The 'Appointment' object is highlighted with a red border. The table data is as follows:

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Appointment	Appointment_c	Custom Object		24/08/2023	✓
Appointment Category	AppointmentCategory	Standard Object			
Appointment Invitation	AppointmentInvitation	Standard Object			
Appointment Invitee	AppointmentInvitee	Standard Object			

2. Now click on “Fields & Relationships” >> New

The screenshot shows the 'Fields & Relationships' page for the 'Appointment' object. The left sidebar has options: 'Details', 'Fields & Relationships' (which is selected and highlighted with a red border), 'Page Layouts', and 'Lightning Record Pages'. The main area is titled 'Fields & Relationships' with a sub-header '14 Items. Sorted by Field Label'. A table lists fields with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. Two fields are listed: 'Appointment Date' and 'Appointment Name'. A red box highlights the 'New' button at the top right of the table header.

3. Select “Look-up relationship” as data type and click Next.

The screenshot shows the 'Data Type' configuration step. The top right corner has a red arrow pointing to the 'Next' button. The page instructions say 'Specify the type of information that the custom field will contain.' Under 'Data Type', the 'Lookup Relationship' option is selected (indicated by a red circle) and highlighted. Other options shown are 'None Selected', 'Auto Number', 'Formula', 'Roll-Up Summary', and 'Master-Detail Relationship'. The 'Lookup Relationship' description states: 'Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.' Below this, there are two bullet points: 'The relationship field is required on all detail records.' and 'The ownership and sharing of a detail record are determined by the master record.'

4. Select the related object “ Customer Details” and click next.

5. Next >> Next >> Save.

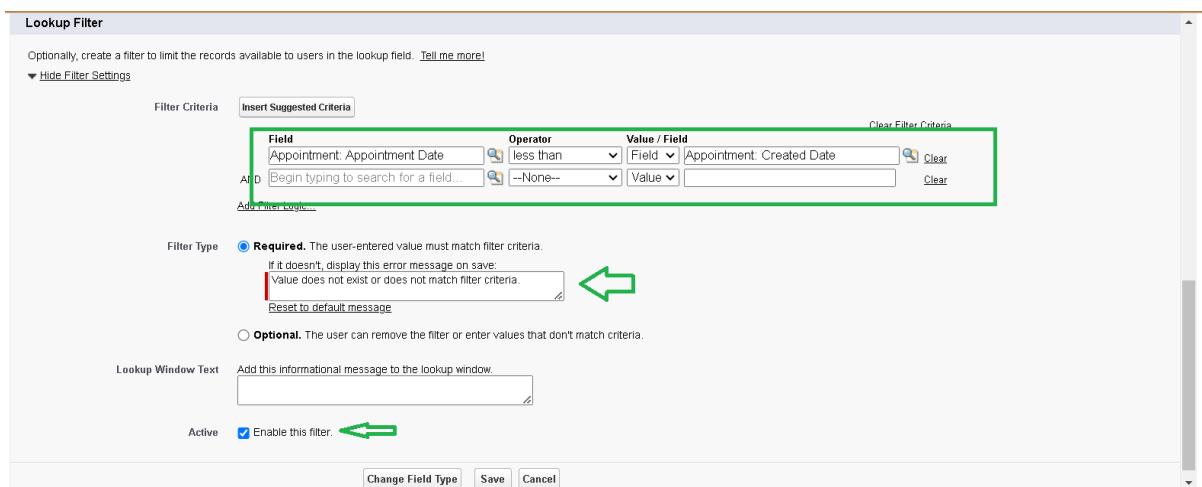
Note: Make sure you complete Activity 4 Before continuing.

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
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.

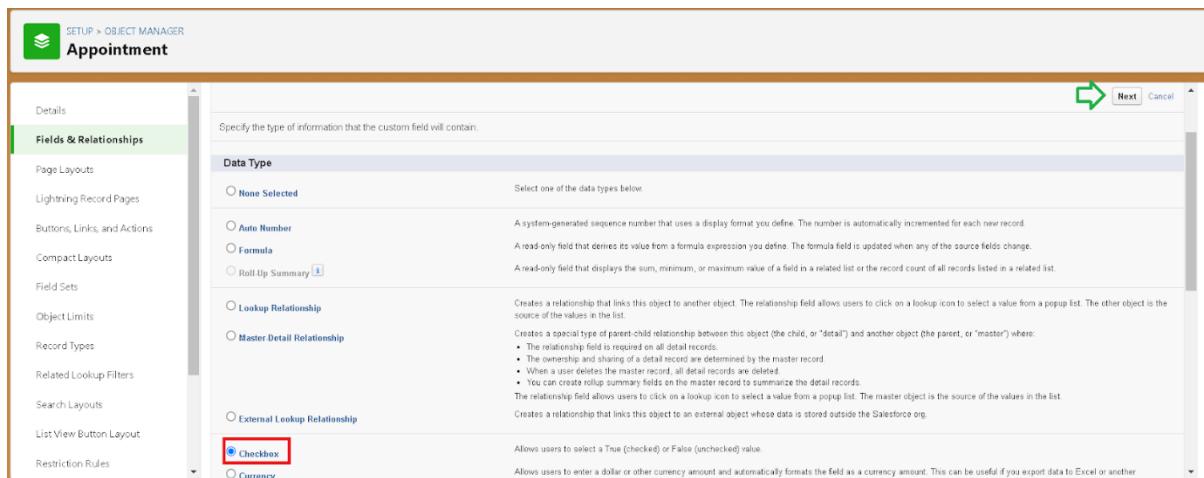
Creation of Lookup Field on Billing details and feedback Object :

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

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

Step 2. Enter the details Step 2 of 4

Field Label 

Default Value Checked Unchecked 

Field Name 

Description

Help Text

Auto add to custom report type Add this field to existing custom report types that contain this entity 

Previous Next Cancel 

7. Click on next >> next >> save.

Creation of Another Checkbox Field on Appointment Object :

1. Repeat the steps form 1 to 3.
2. Give the Field Label : Repairs
3. Field Nme : 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 Nme : is auto populated
9. Default value : unchecked
10. Click on next >> next >> save.

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 Nme : is auto populated
6. Default value : unchecked
7. Click on next >> next >> save

Creation of date fields

Creation of Date 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 “Date” as data type and click Next.
4. Give the Field Label : Appointment Date
5. Field Name : is auto populated
6. Make it as a Required field by click on the Required option.
7. Click on next >> next >> save.

Appointment
New Custom Field

Help for this Page 

Step 2 of 4

Step 2. Enter the details

Previous  Next Cancel

Field Label	<input type="text" value="Appointment Date"/> 
Field Name	<input type="text" value="Appointment_Date"/> 
Description	<input type="text"/>
Help Text	<input type="text"/>
Required	<input checked="" type="checkbox"/> Always require a value in this field in order to save a record
Auto add to custom report type	<input checked="" type="checkbox"/> Add this field to existing custom report types that contain this entity 
Default Value	<input type="text" value="Show Formula Editor"/> 

Creation of Currency fields

Creation of Currency 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 “Currency” as data type and click Next.
4. Give the Field Label : Service Amount
5. Field Nme : is auto populated

This screenshot shows the 'Step 2. Enter the details' screen for creating a new custom field. The field label is set to 'Service Amount'. The length is set to 18, and the decimal places are set to 0. The field name is 'Service_Amount'. There is a description and help text section, both of which are currently empty. Under the 'Required' section, there is a checkbox for 'Always require a value in this field in order to save a record' which is unchecked. Below that, there is a checkbox for 'Add this field to existing custom report types that contain this entity' which is checked. At the bottom right of the screen, there are 'Previous', 'Next', and 'Cancel' buttons.

6. Click on next
7. Give read only for all the profiles in field level security for profile.

This screenshot shows the 'Step 3. Establish field-level security' screen. It displays the field details: Field Label: Service Amounts, Data Type: Currency, Field Name: Service_Amounts, and Description. Below this, it says 'Select the profiles to which you want to grant edit access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.' A table titled 'Field-Level Security for Profile' lists various profiles: Analytics Cloud Integration User, Analytics Cloud Security User, Authenticated Website, Authenticated Website, Contract Manager, and Cross Org Data Proxy User. For each profile, there are two checkboxes: 'Visible' (checked for all) and 'Read-Only' (checked for all). A green arrow points to the 'Read-Only' column header. At the bottom right of the screen, there are 'Previous', 'Next', and 'Cancel' buttons.

8. Click on next >> save.

Creation of Currency Field on Billing details and feedback Object :

1. Follow the same steps as mentioned above in Billing details and feedback Object.
2. Change the label name as mentioned.
3. Give the Field Label : Payment Paid
4. Field Nme : is auto populated

Creation of Text fields

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 “Text” as data type and click Next.
4. Give the Field Label : Vehicle number plate
5. Field Name : is auto populated
6. Length : 10
7. Make field as Required and Unique.

The screenshot shows the 'Step 2. Enter the details' configuration page for creating a new text field. The page has a header 'Step 2 of 4' and navigation buttons 'Previous', 'Next', and 'Cancel'. The form fields include:

- Field Label:** Vehicle number plate
- Length:** 10 (with a note: "Please enter the maximum length for a text field below.")
- Field Name:** Vehicle_number_plate
- Description:** (empty text area)
- Help Text:** (empty text area)
- Required:** Always require a value in this field in order to save a record
- Unique:** Do not allow duplicate values
 - Treat "ABC" and "abc" as duplicate values (case insensitive)
 - Treat "ABC" and "abc" as different values (case sensitive)
- External ID:** Set this field as the unique record identifier from an external system
- Auto add to custom report type:** Add this field to existing custom report types that contain this entity

8. Click on next >> next >> save.

Creation of Text Fields in 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 “text” as data type and click Next.
4. Give the Field Label : Rating for service
5. Field Name : is auto populated
6. Length : 1
7. Make field as Required and Unique.
8. Click on next >> next >> save

Creation of picklist fields

Creation of Picklist Fields in Service records object :

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Service Status”, under values select “Enter values, with each value separated by a new line” and enter values as shown below.
5. The values are: Started, Completed.

New Custom Field

Step 2. Enter the details Step 2 of 4

Field Label [i]

Values Use global picklist value set
 Enter values, with each value separated by a new line

Display values alphabetically, not in the order entered
 Use first value as default value [i]
 Restrict picklist to the values defined in the value set [i]

Field Name [i]

Description

Previous Next Cancel

6. Click Next.
7. Next >> Next >> Save.

Creation of Picklist Fields in Billing details and feedback object :

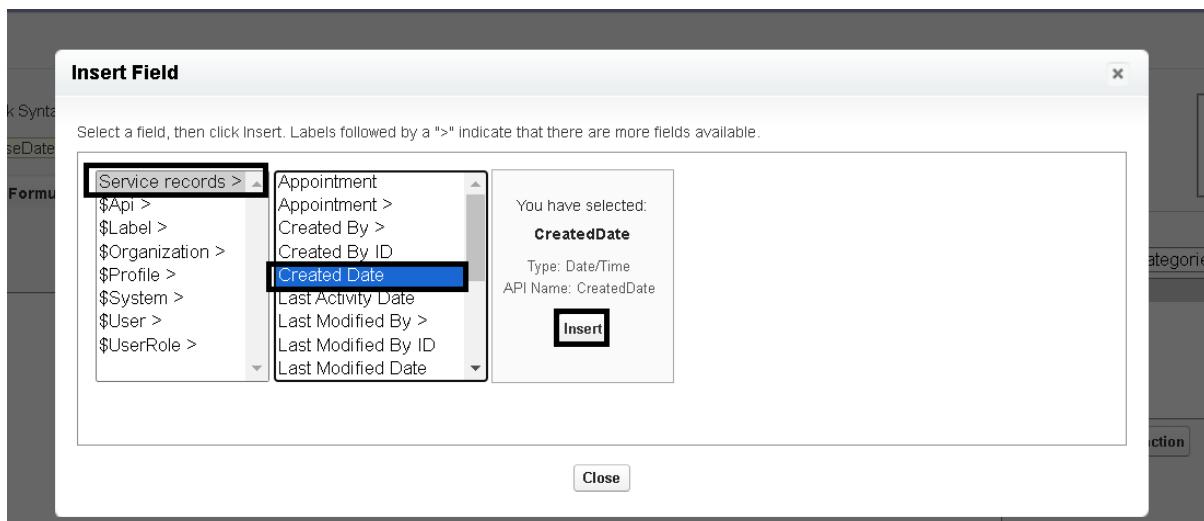
1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Payment Status”, under values select “Enter values, with each value separated by a new line” and enter values as shown below.
5. The values are: Pending, Completed.
6. Click Next.
7. Next >> Next >> Save.

Creating Formula Field in Service records Object

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “service date” and select formula return type as “Date” and click next.



5. Insert field formula should be : CreatedDate



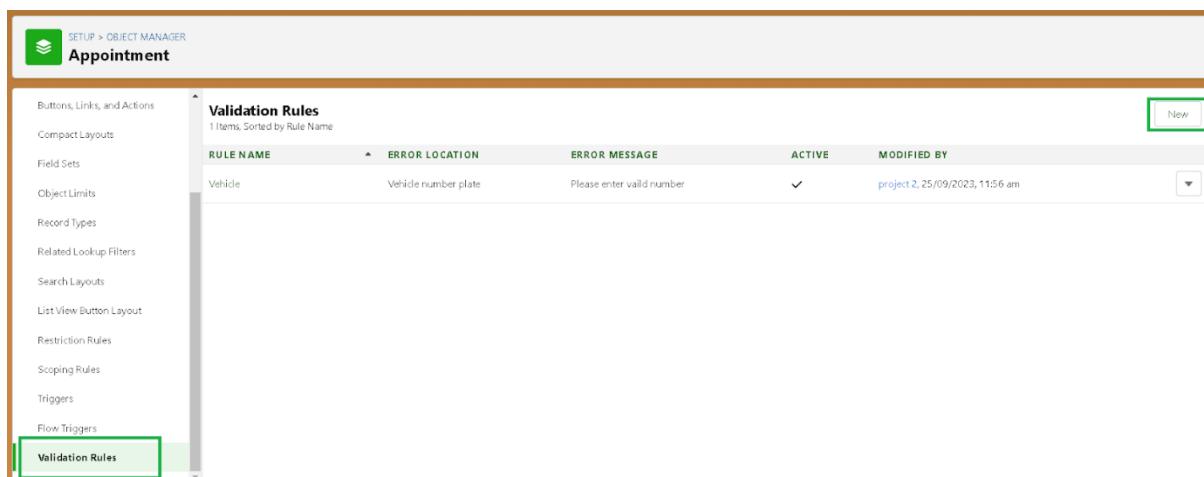
6. click “Check Syntax” .
7. Click next >> next >> Save.

Validation Rule

Validation rules are applied when a user tries to save a record and are used to check if the data meets specified criteria. If the criteria are not met, the validation rule triggers an error message and prevents the user from saving the record until the issues are resolved.

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.



The screenshot shows the Salesforce Object Manager for the 'Appointment' object. On the left, there's a sidebar with various tabs like Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, Triggers, and Flow Triggers. The 'Validation Rules' tab is highlighted with a green box. The main area is titled 'Validation Rules' and shows a table with one item: 'Vehicle'. The table has columns: RULE NAME, ERROR LOCATION, ERROR MESSAGE, ACTIVE, and MODIFIED BY. The 'Vehicle' row has 'Vehicle' in the RULE NAME column, 'Vehicle number plate' in the ERROR LOCATION column, 'Please enter valid number' in the ERROR MESSAGE column, an active checkbox checked, and 'project 2, 25/09/2023, 11:56 am' in the MODIFIED BY column. A 'New' button is visible in the top right corner of the main area.

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Vehicle	Vehicle number plate	Please enter valid number	✓	project 2, 25/09/2023, 11:56 am

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}"))`

Validation Rule Edit

Save Save & New Cancel

Rule Name Vehicle

Active

Description

Error Condition Formula

Example: More Examples...
Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

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

Functions

-- All Function Categories --

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

ABS(number)
Returns the absolute value of a number, a number without its sign
[Help on this function](#)

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.

Error Message

Example:

This message will appear when Error Condition formula is **true**

Error Message Please enter valid number

This error message can either appear at the top of the page or below a specific field on the page

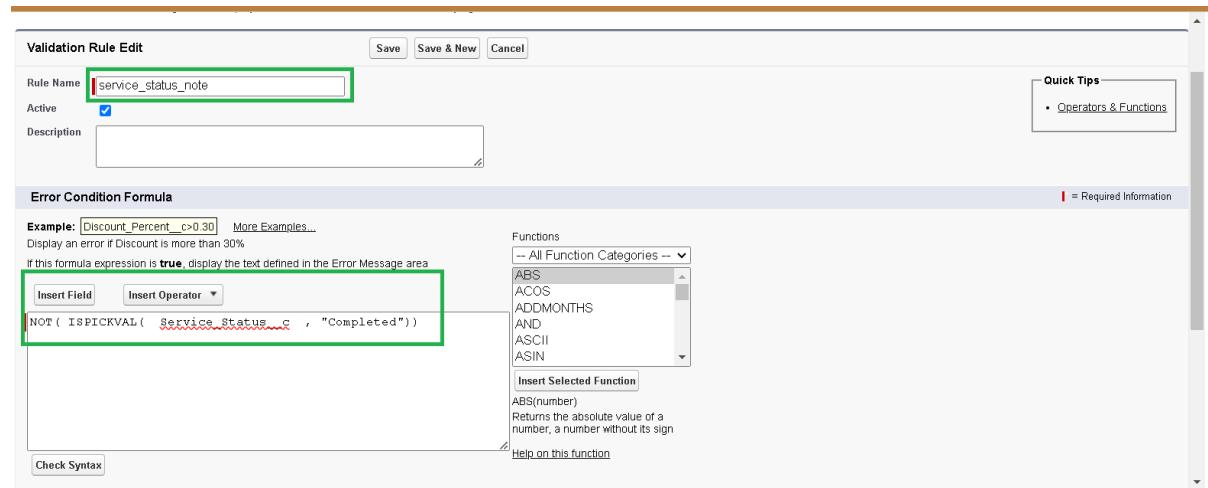
Error Location Top of Page Field Vehicle number plate 

 Save Save & New Cancel

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"))



The screenshot shows the 'Validation Rule Edit' window. In the 'Error Condition Formula' section, the formula `NOT(ISPICKVAL(Service_Status__c , "Completed"))` is entered in the text area and highlighted with a green box. To the right of the formula, a dropdown menu titled 'Functions' lists several functions: ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, with 'ABS(number)' being the selected function. A tooltip for 'ABS(number)' is displayed below the dropdown. The 'Rule Name' field contains 'service_status_note' and has a green border. The 'Active' checkbox is checked. The 'Description' field is empty. The 'Save' button is visible at the top right.

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.



The screenshot shows the 'Error Message' screen. In the 'Error Message' section, the message `still it is pending` is entered in the text area and highlighted with a green box. Below the message, the 'Error Location' section shows the 'Field' radio button selected, and 'Service Status' is chosen from the dropdown. A large green arrow points to the 'Save' button at the bottom right of the screen.

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}"))

The screenshot shows the 'Validation Rule Edit' interface. The 'Rule Name' field contains 'service_status_note'. The 'Error Condition Formula' field contains the formula `NOT(ISPIKVAL(Service_Status__c , "Completed"))`, which is also highlighted with a green box. A dropdown menu titled 'Functions' is open, showing various functions like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. The 'Insert Selected Function' button is visible. The 'Check Syntax' button is at the bottom left. A 'Quick Tips' panel on the right suggests 'Operators & Functions'.

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 'Error Message' screen. The 'Error Message' field contains 'still it is pending'. The 'Error Location' section shows 'Field Service Status' selected, with a radio button next to 'Field'. A large green arrow points to the 'Save' button at the bottom. The 'Save' button is highlighted with a green box.

Duplicate rule:

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.

The screenshot shows the Salesforce Setup interface. In the top navigation bar, 'Setup' is selected. Below it, the 'Object Manager' section is open, with 'Matching Rules' highlighted. A green arrow points to the 'Matching Rules' link. The main content area is titled 'Matching Rules' and displays a table of 'All Matching Rules'. The table has columns for Action, Rule Name, Object, Status, Description, Last Modified Date, and Last Modified By. A green arrow points to the 'New Rule' button at the top right of the table. The URL in the browser address bar is /setup/ObjectManager/MatchingRules

3. Select the object as Customer details and click Next.

The screenshot shows the 'New Matching Rule' wizard. It is on 'Step 1 of 2'. The title is 'Matching Rule' and 'New Matching Rule'. The sub-step title is 'Step 1: Select object'. The instructions say 'Select the object to which this matching rule applies.' A dropdown menu labeled 'Object' is open, with 'Customer Details' selected. A green arrow points to this dropdown. At the bottom right of the step, there are 'Next' and 'Cancel' buttons. The URL in the browser address bar is /setup/ObjectManager/MatchingRules/new

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.

Save Cancel

Rule Details

Object: Customer Details
 Rule Name: matching Customer data 
 Unique Name: matching_Customer_det 
 Description:

Matching Criteria

Tell the rule which fields to compare and how.

Field	Matching Method	Match Blank Fields
Gmail	Exact	<input type="checkbox"/> AND
Phone Number	Exact	<input type="checkbox"/> AND
--None--	Exact	<input type="checkbox"/> AND
--None--	Exact	<input type="checkbox"/> AND
--None--	Exact	<input type="checkbox"/>

Add Filter Logic... 

Save Cancel

Help for this Page ?

Matching Rule
matching Customer details

Matching Rule Detail

Object: Customer Details
 Rule Name: matching Customer details
 Unique Name: matching_Customer_details
 Description:
 Matching Criteria: (Customer Details: Gmail EXACT MatchBlank = FALSE) AND (Customer Details: Phone_Number EXACT MatchBlank = FALSE)
 Status: Inactive
 Created By: project2, 25/09/2023, 10:15 am
 Modified By: project2, 10/10/2023, 3:32 pm

Edit Delete Clone Activate 

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.

The screenshot shows the 'Duplicate Rules' page in the Salesforce Setup. The left sidebar has a 'Data' section with 'Duplicate Management' and 'Duplicate Error Logs' under 'Duplicate Rules' (which is selected and highlighted in green). Below that is 'Matching Rules'. A global search bar at the top says 'Didn't find what you're looking for? Try using Global Search.' The main area is titled 'All Duplicate Rules' and contains a table with columns: Rule Name, Description, Matching Rule, Active, Last Modified By, and Last Modified Date. The table lists several standard rules like 'Standard Account Duplicate Rule' and 'Standard Contact Duplicate Rule'. A new rule named 'Customer Detail duplicate' is being created, with its matching rule set to 'Matching customer details'.

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.

The screenshot shows the 'Edit Duplicate Rule' page for 'Customer Detail duplicate'. The top bar has 'Edit Duplicate Rule' and 'Customer Detail duplicate'. Below it is a toolbar with 'Save', 'Save & New', and 'Cancel'. The main form is divided into sections: 'Rule Details' and 'Actions'. In 'Rule Details', there's a 'Rule Name' field containing 'Customer Detail duplicate' with a green arrow pointing to it. Other fields include 'Description' (empty), 'Object' (set to 'Customer Details'), and 'Record Level Security' (radio button selected for 'Enforce sharing rules'). In the 'Actions' section, there are dropdown menus for 'Action On Create' (set to 'Allow') and 'Action On Edit' (set to 'Allow'), both with checked 'Alert' boxes. An 'Alert Text' field contains the placeholder 'Use one of these records?'.

Matching Rules

Define how duplicate records are identified.

Compare Customer Details With Matching Rule 

Matching Criteria
(Customer Details: Gmail EXACT MatchBlank = FALSE) AND
(Customer Details: Phone_Number EXACT MatchBlank = FALSE)

Field Mapping  Mapping Selected

Conditions

Optionally, specify the conditions a record must meet for the rule to run.

Field	Operator	Value	AND
--None--	--None--		AND
--None--	--None--		

[Add Filter Logic...](#)



Profiles

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

Types of profiles in salesforce

1. Standard profiles:

By default salesforce provides below standard profiles.

- Contract Manager
- Read Only
- Marketing User
- Solutions Manager
- Standard User
- System Administrator.

We cannot deleted standard ones

Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

2. Custom Profiles:

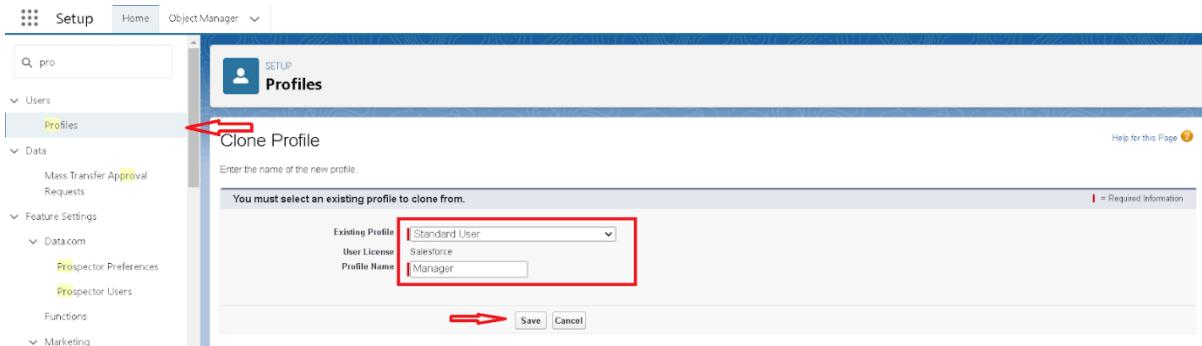
Custom ones defined by us.

They can be deleted if there are no users assigned with that particular one.

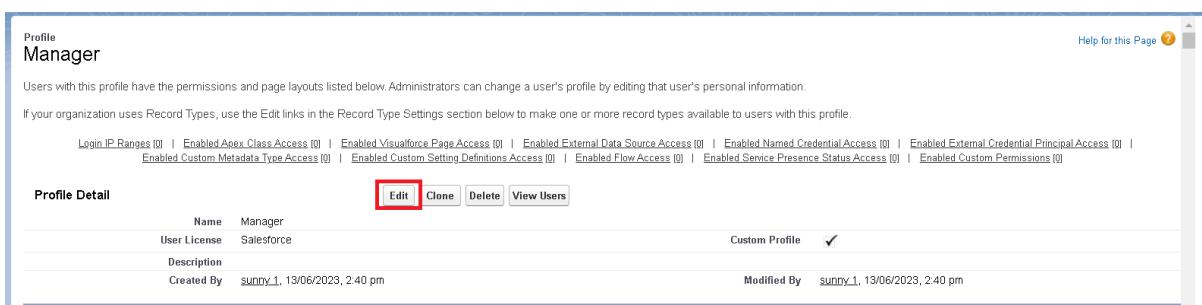
Manager Profile

To create a new profile:

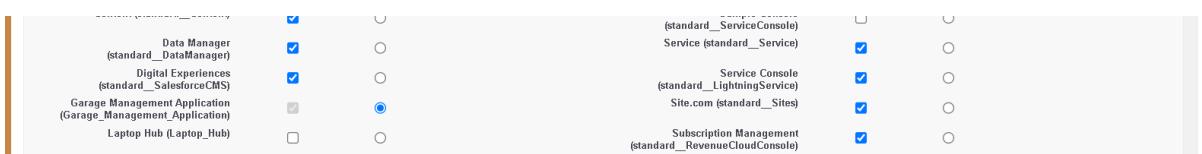
1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.



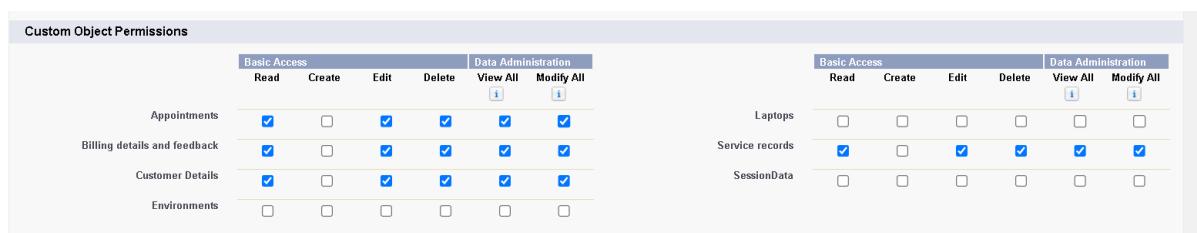
2. While still on the profile page, then click Edit.



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.

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.

Custom Object Permissions						
	Basic Access				Data Administration	
	Read	Create	Edit	Delete	View All	Modify All
Appointments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Billing details and feedback	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customer Details	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Environments	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Laptops	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Service records	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SessionData	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5. And click save.

Role & Role Hierarchy :

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Creating Manager role :

Creating Manager Role:

1. Go to quick find >> Search for Roles >> click on set up roles.

Setup Home Object Manager ▾

Q: roles

Users Roles

Feature Settings

Sales

Contact Roles on Contracts

Contact Roles on Opportunities

Service

Case Teams

Case Team Roles

Contact Roles on Cases

Did you find what you're looking for? Try using Global Search.

Help for this Page ⓘ

SETUP Roles

Understanding Roles

Sample Role Hierarchy

View other sample Role Hierarchies: Territory-based Sample ▾

Executive Staff

CEO President CFO VP, Sales

Western Sales Director

Eastern Sales Director

International Sales Director

Western Sales Rep

Eastern Sales Rep

International Sales Rep

Set Up Roles

Don't show this page again

2. Click on Expand All and click on add role under whom this role works.

Your Organization's Role Hierarchy

Collapse All Expand All

Nick Enterprises

Add Role

CFO Edit | Del | Assign

HR Edit | Del | Assign

Manager Edit | Del | Assign

On Site Emp Edit | Del | Assign

Remote Emp Edit | Del | Assign

3. Give Label as "Manager" and Role name gets auto populated. Then click on Save.

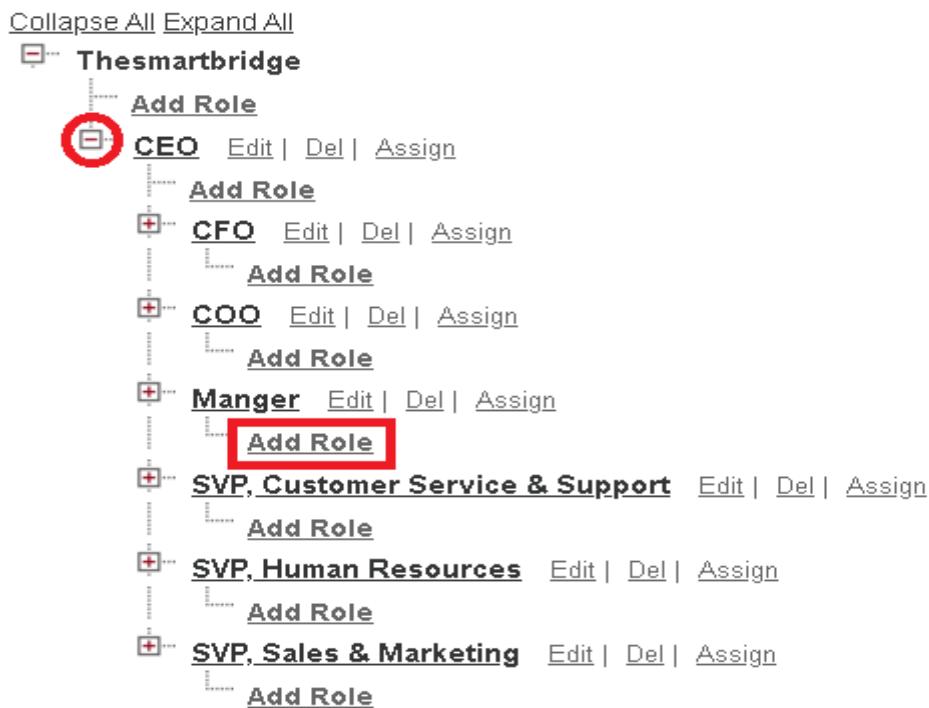
Role Edit

Label	<input type="text" value="Manger"/> 
Role Name	<input type="text" value="Manger"/> 
This role reports to	<input type="text" value="CEO"/> 
Role Name as displayed on reports	<input type="text"/>
 <input type="button" value="Save"/> <input type="button" value="Save & New"/> <input type="button" value="Cancel"/>	

Creating another roles

Creating another two roles under manager

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.

Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Create User

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

New User

User Edit

General Information

Required Information

Save Save & New Cancel

Field	Value
First Name	Niklaus
Last Name	Mikaelson
Alias	Nnika
Email	
Username	Mikaelson@Niklaus
Nickname	Nik
Title	
Company	
Department	
Division	
Role	Manager
User License	Salesforce
Profile	Manager
Active	✓

3. Save.

4. Create User

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

New User

User Edit [Save](#) [Save & New](#) [Cancel](#)

General Information

First Name	Niklaus
Last Name	Mikaelson
Alias	Jmika
Email	
Username	Mikaelson@Niklaus
Nickname	Nik
Title	
Company	
Department	
Division	

Role: Manager User License: Salesforce Profile: Manager Active: ✓

Marketing User Offline User Knowledge User Flow User Service Cloud User

Site.com Contributor User Site.com Publisher User WDC User Data.com User Type: --None--

Help for this Page ?

3. Save.

Creating another users

1. Repeat the steps and create another user using
 - a. Role : sales person
 - b. User licence : Salesforce Platform
 - c. Profile : sales person

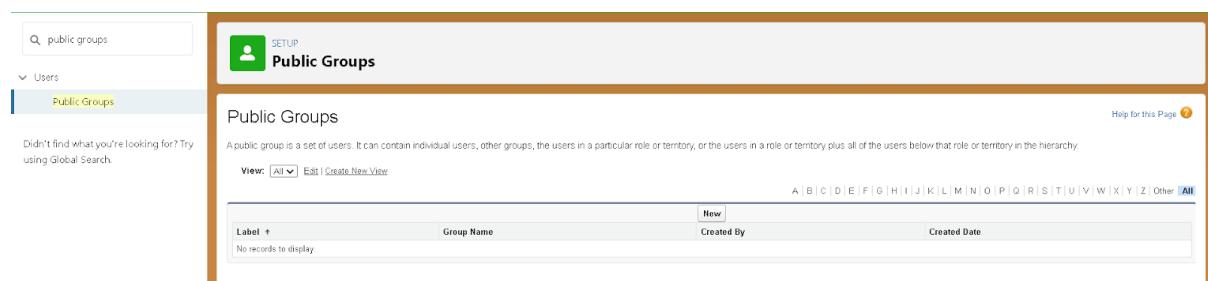
Note : create atleast 3 users with these permissions.

Public groups

Public groups are a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings. By creating and using public groups effectively, you can maintain a secure and organized Salesforce environment while ensuring that users have appropriate access to the resources they need.

Creating new public group

1. Go to setup >> type users in quick find box >> select public groups >> click New.



2. Give the Label as “sales team”.
3. Group name is autopopulated.
4. Search for Roles.
5. In Available Members select Sales person and click on add it will be moved to selected member.
6. Click on save.

The screenshot shows the 'Group Information' dialog box with the title 'New Public Group'. It includes fields for 'Label' (Sales Team) and 'Group Name' (Sales_Team). A checkbox 'Grant Access Using Hierarchies' is checked. Below these are sections for 'Available Members' (a list of various roles like Customer Support, Director, Sales Manager, etc.) and 'Selected Members' (a list containing 'Role: Sales person'). Buttons for 'Add' and 'Remove' are between the two lists. At the bottom are 'Save' and 'Cancel' buttons, and a note '! = Required Information'.

Sharing Setting

Salesforce allows you to configure sharing settings to control how records are accessed and shared within your organization. These settings are crucial for maintaining data security and privacy. Salesforce provides a variety of tools and mechanisms to define and enforce sharing rules, such as:

Organization-Wide Default (OWD) Settings:

These settings define the default level of access for all objects within your Salesforce org.

OWD settings include Private, Public Read-Only, Public Read/Write, and Controlled by Parent.

OWD settings can be configured for each standard and custom object.

Role Hierarchy:

Salesforce uses a role hierarchy to determine record access.

Users at higher levels in the hierarchy have greater access to records owned by or shared with users lower in the hierarchy.

The role hierarchy is often used in combination with OWD settings to grant different levels of access.

Profiles and Permission Sets:

Profiles and permission sets allow administrators to specify object-level and field-level permissions for users.

Profiles are typically used to grant general object and field access, while permission sets can be used to extend those permissions to specific users.

Sharing Rules:

Sharing rules are used to extend access to records for users who meet specific criteria.

They can be used to grant read-only or read-write access to records owned by other users.

Manual Sharing:

Administrators and record owners can manually share specific records with other users or groups.

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.

Sharing Settings

Work Plan Template	Private	Private	<input checked="" type="checkbox"/>
Work Step Template	Private	Private	<input checked="" type="checkbox"/>
Work Type	Private	Private	<input checked="" type="checkbox"/>
Work Type Group	Public Read/Write	Private	<input checked="" type="checkbox"/>
Appointment	Public Read/Write	Private	<input checked="" type="checkbox"/>
Billing details and feedback	Public Read/Write	Private	<input checked="" type="checkbox"/>
Customer Details	Public Read/Write	Private	<input checked="" type="checkbox"/>
Environment	Public Read/Write	Private	<input checked="" type="checkbox"/>
Laptop	Public Read/Write	Private	<input checked="" type="checkbox"/>
Service records	Private	Private	<input checked="" type="checkbox"/>
SessionData	Public Read/Write	Private	<input checked="" type="checkbox"/>

User Visibility Settings

Portal User Visibility

Site User Visibility

Other Settings

Standard Report Visibility

Manual User Record Sharing

Manager Groups

Minimize the number of roles created, which improves performance by cutting down processing loads

Grant site users access to related record cases

Secure guest user access

Require permission to view record names in lookup fields

Buttons: Save Cancel

3. Click on save and refresh.
 4. Scroll down a bit, Click new on Service records sharing Rules.
 - 5.
- Service records Sharing Rules

New Recalculate Service records Sharing Rules Help ?

No sharing rules specified.
6. Give the Label name as “ Sharing setting”
 7. Rule name is auto populated.
 8. In step 3 : Select which records to be shared, members of “ Roles ” >> “ Sales person ”
 9. In step 4: share with, select “ Roles ” >> “ Manager ”
 10. In step 5 : Change the access level to “ Read / write ”.
 11. Click on save.

The screenshot shows the 'Sharing Settings' wizard in Salesforce. It consists of five steps:

- Step 1: Rule Name**: The 'Label' field contains 'sharing settings' and the 'Rule Name' field contains 'sharing_settings'. A red arrow points to the 'sharing settings' label.
- Step 2: Select your rule type**: The 'Rule Type' section shows 'Based on record owner' selected. A red arrow points to the 'Based on record owner' radio button.
- Step 3: Select which records to be shared**: The 'Service records: owned by members of' dropdown is set to 'Roles' and the 'Sales person' checkbox is selected. A red arrow points to the 'Sales person' checkbox.
- Step 4: Select the users to share with**: The 'Share with' dropdown is set to 'Roles' and the 'Manager' checkbox is selected. A red arrow points to the 'Manager' checkbox.
- Step 5: Select the level of access for the users**: The 'Access Level' dropdown is set to 'ReadWrite'. A red arrow points to the 'ReadWrite' dropdown.

At the bottom, there are 'Save' and 'Cancel' buttons, with a red arrow pointing to the 'Save' button.

Flows

In Salesforce, a flow is a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created without any coding knowledge.

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.

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated to meet the condition requirements** option for When to Run the Flow for Updated Records.

Condition Requirements

None ▾

* Optimize the Flow for:

Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs *before* the record is saved to the database.

Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.

3

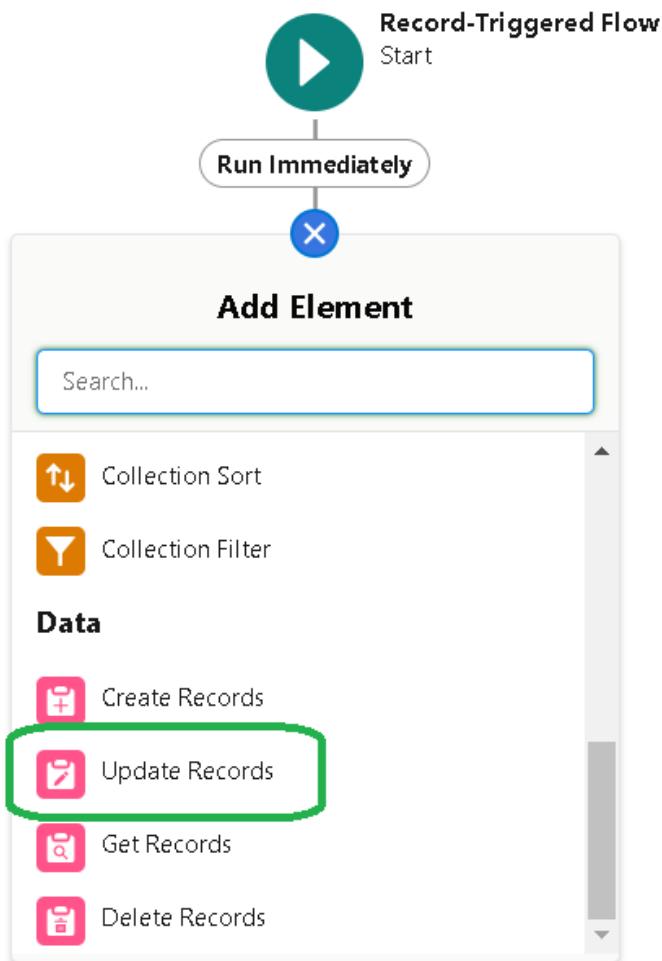
Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed

4

Cancel

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

Edit Update Records

Update Salesforce records using values from the flow.

*Label * API Name

Amount Update Amount_Update

Description

* How to Find Records to Update and Set Their Values

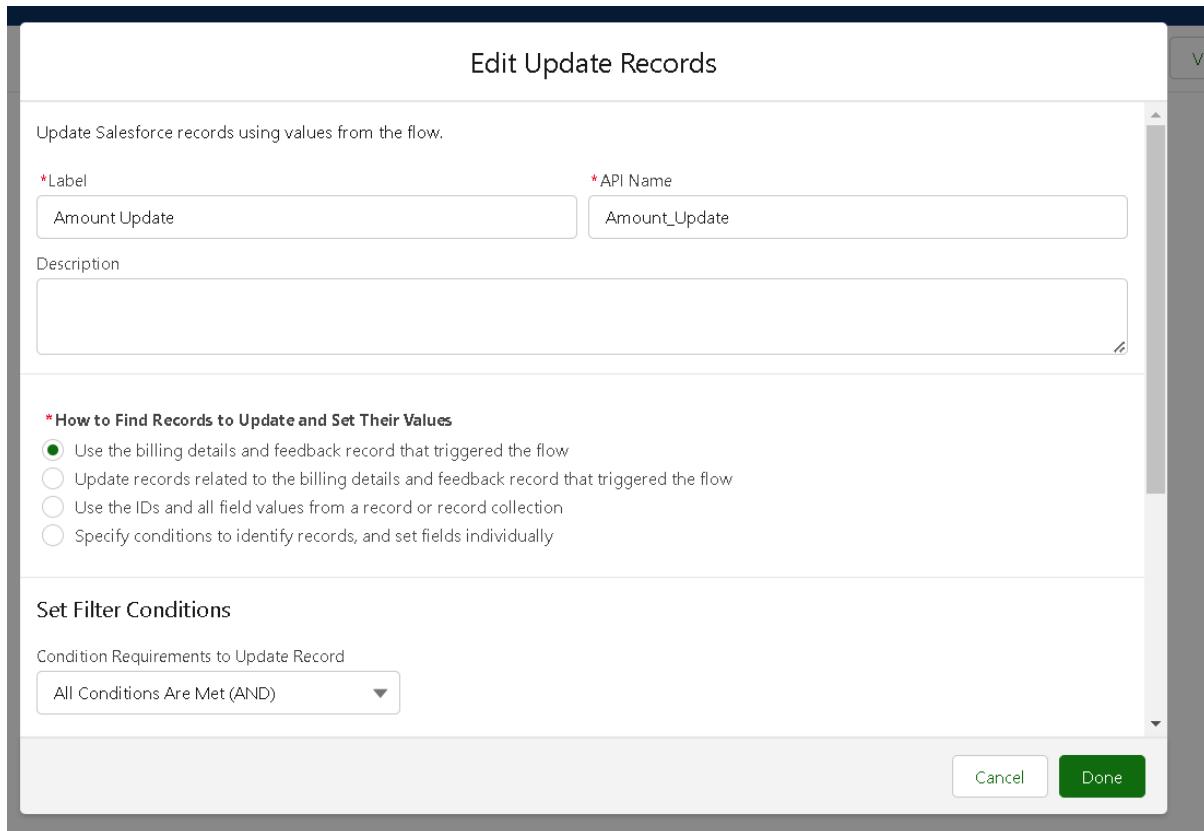
- Use the billing details and feedback record that triggered the flow
- Update records related to the billing details and feedback record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND) ▾

Cancel Done



Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND) ▾

Field Operator Value

Payment_Status__c Equals Completed

+ Add Condition

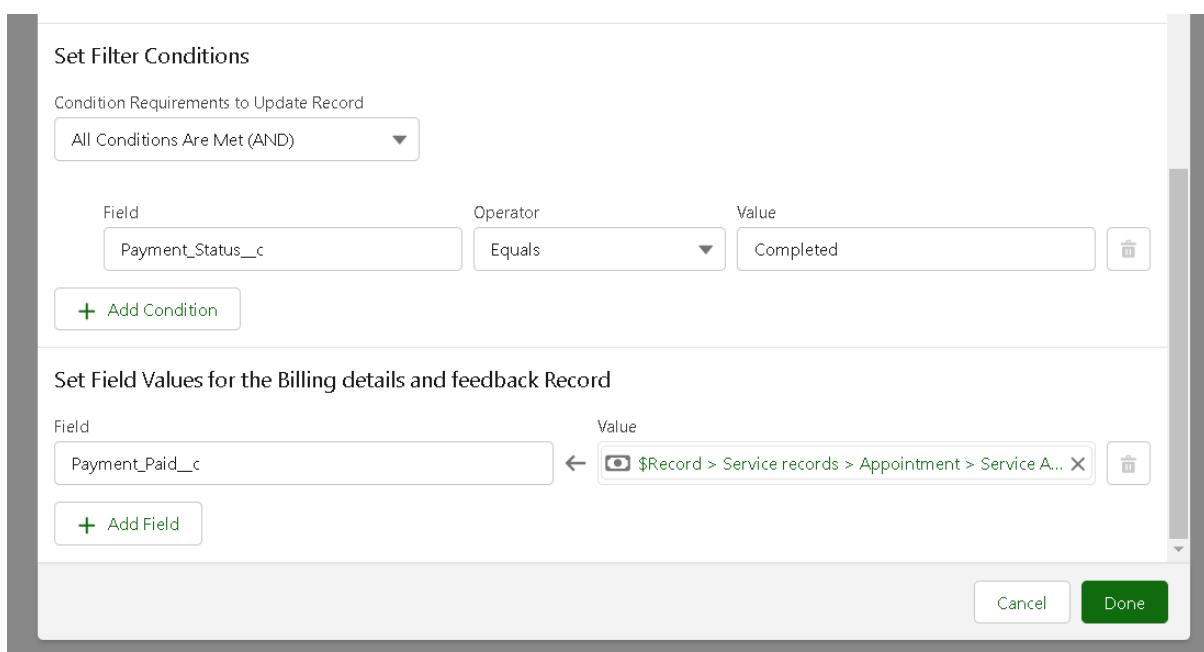
Set Field Values for the Billing details and feedback Record

Field Value

Payment_Paid__c \$Record > Service records > Appointment > Service A... X

+ Add Field

Cancel Done

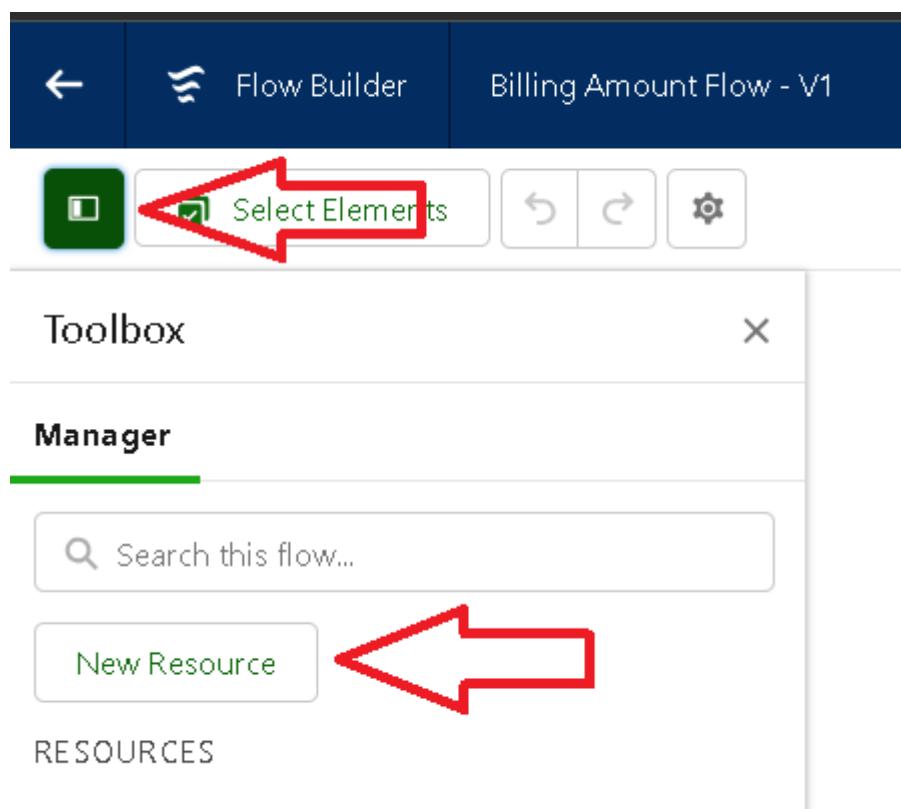


9. Set a filter condition : All Conditions are met(AND)

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 from 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.

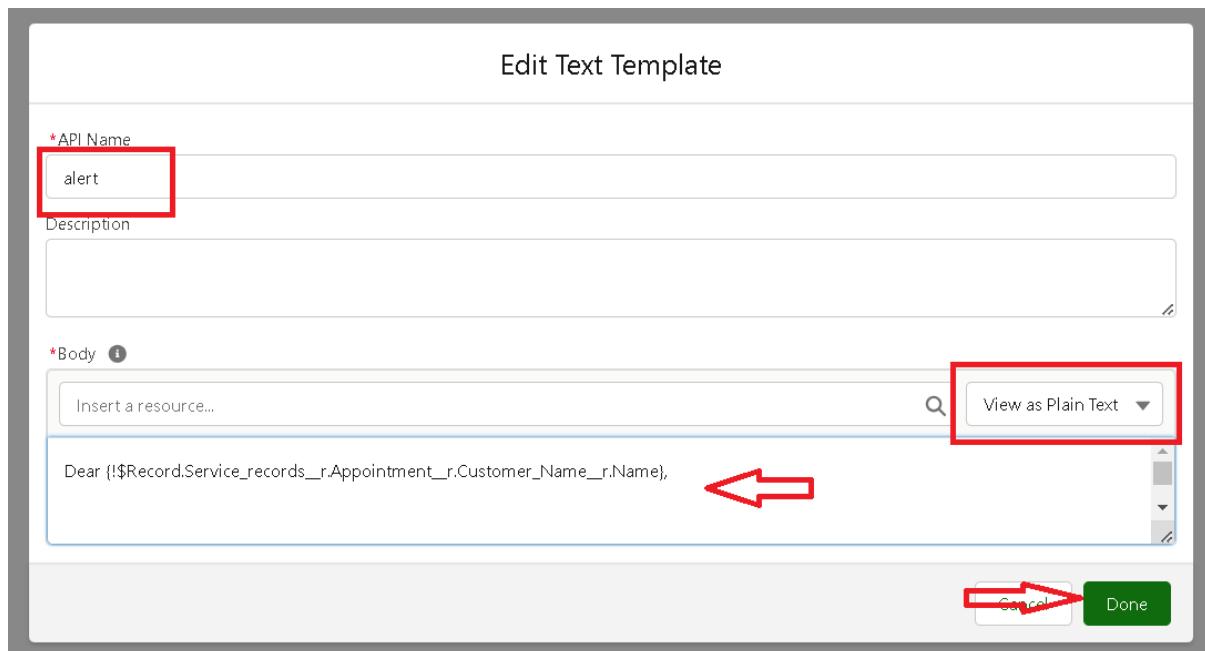
Dear {!\$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

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 .

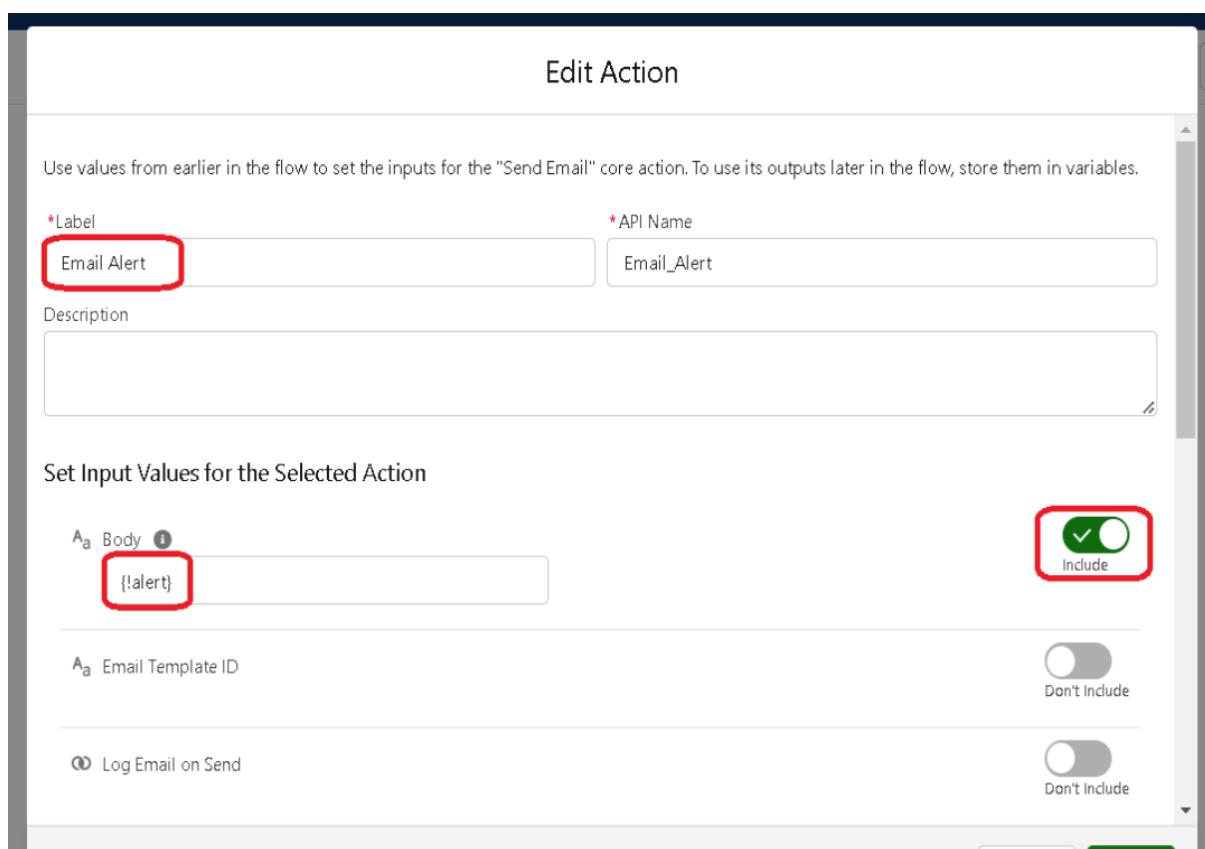
23. Click done.

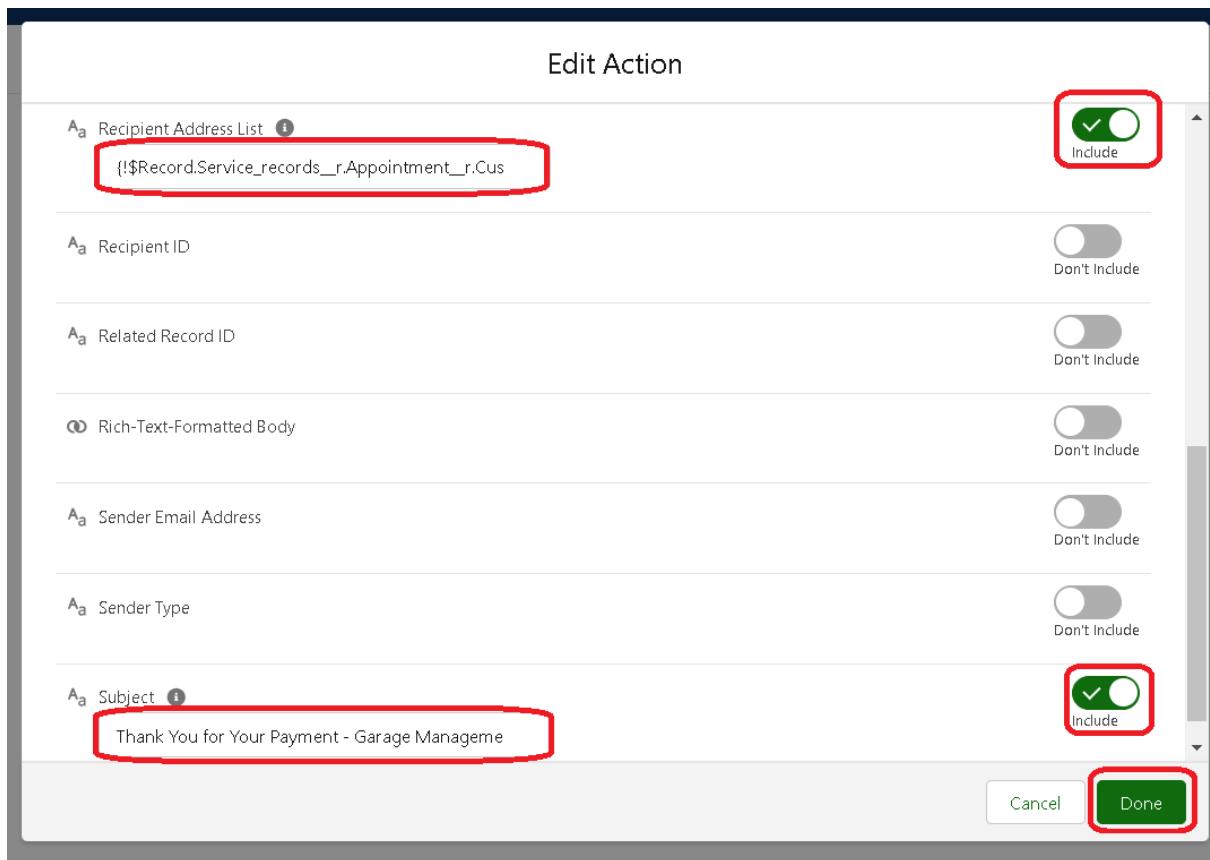


24. Now Click on Add Element , select Action.

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:

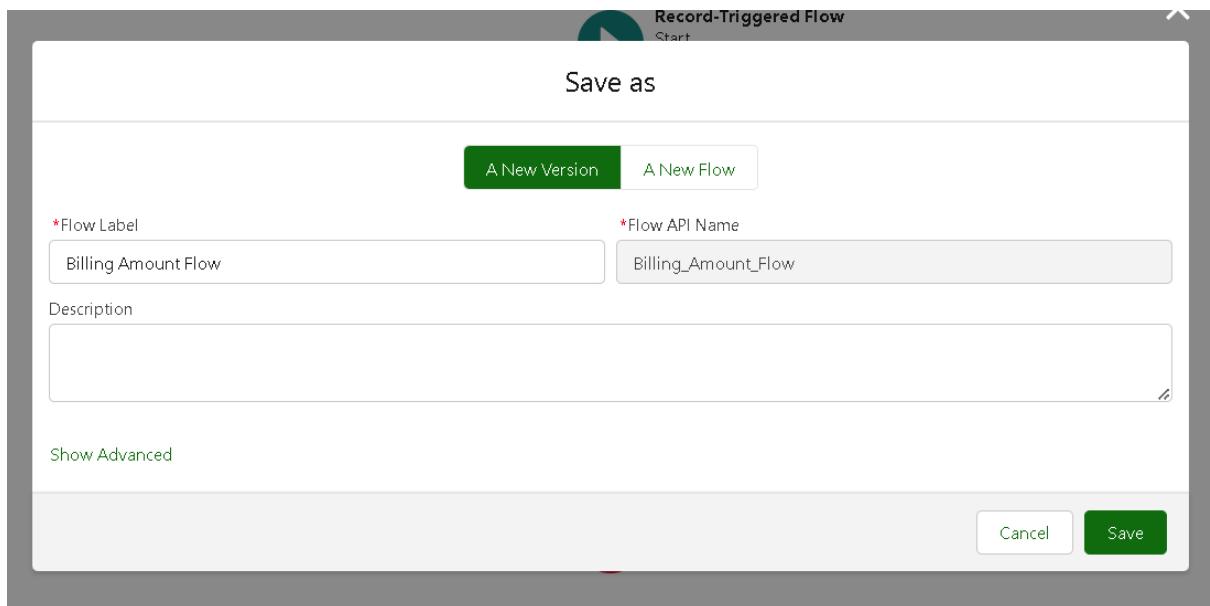
```
{!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}
```
32. Include subject as “ Thank You for Your Payment - Garage Management”.
33. Click done.

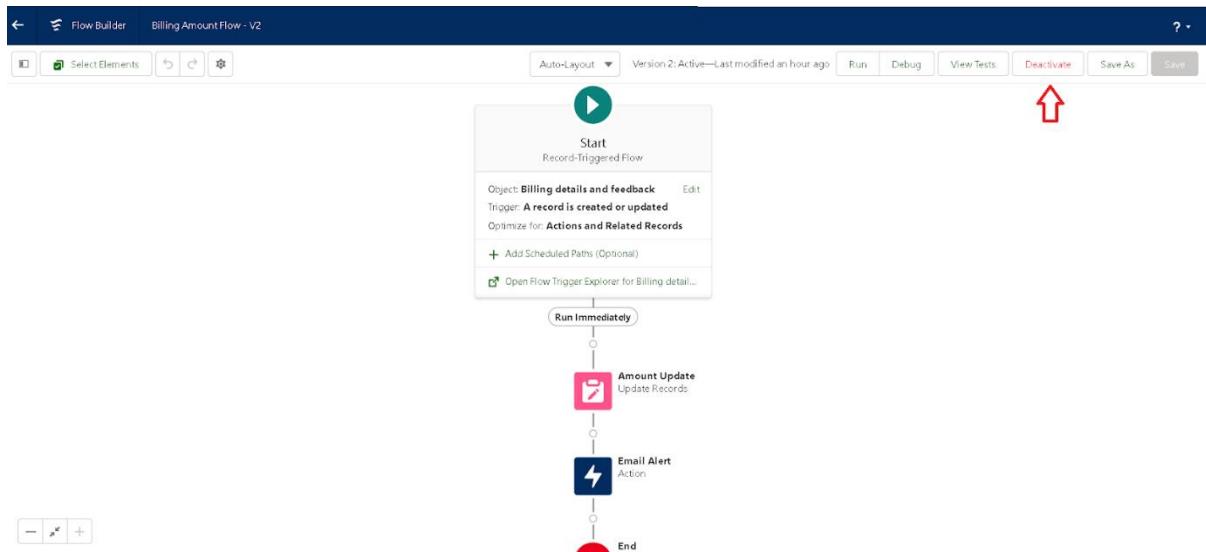




34. Click on save. Give the Flow label , Flow Api name will be autopopulated.

35. And click save, and click on activate.





Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions

before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- insert
- update
- delete
- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom

objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

Before Trigger:

This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

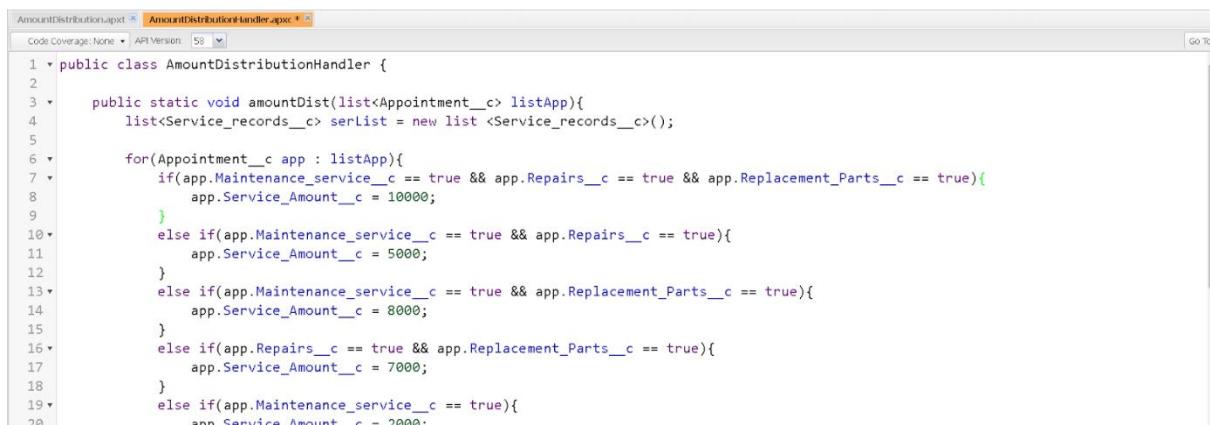
SOURCECODE:

After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

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 ”.



The screenshot shows the Salesforce Developer Console interface. The tabs at the top are "AmountDistribution.apxt" and "AmountDistributionHandler.apex". The status bar indicates "Code Coverage: None" and "API Version: 58". The main area contains the Apex code for the "AmountDistributionHandler" class:

```
1 * public class AmountDistributionHandler {  
2 *     public static void amountDist(list<Appointment__c> listApp){  
3 *         list<Service_records__c> serlist = new list <Service_records__c>();  
4 *         for(Appointment__c app : listApp){  
5 *             if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){  
6 *                 app.Service_Amount__c = 10000;  
7 *             }  
8 *             else if(app.Maintenance_service__c == true && app.Repairs__c == true){  
9 *                 app.Service_Amount__c = 5000;  
10 *            }  
11 *            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){  
12 *                app.Service_Amount__c = 8000;  
13 *            }  
14 *            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){  
15 *                app.Service_Amount__c = 7000;  
16 *            }  
17 *            else if(app.Maintenance_service__c == true){  
18 *                app.Service_Amount__c = 2000;  
19 *            }  
20 *        }  
21 *    }  
22 * }
```

```
AmountDistribution.apxt AmountDistributionHandler.apxc *
Code Coverage: None API Version: 58
12     }
13     else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
14         app.Service_Amount__c = 8000;
15     }
16     else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
17         app.Service_Amount__c = 7000;
18     }
19     else if(app.Maintenance_service__c == true){
20         app.Service_Amount__c = 2000;
21     }
22     else if(app.Repairs__c == true){
23         app.Service_Amount__c = 3000;
24     }
25     else if(app.Replacement_Parts__c == true){
26         app.Service_Amount__c = 5000;
27     }
28
29 }
30 }
31 }
```

```
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;
}

}

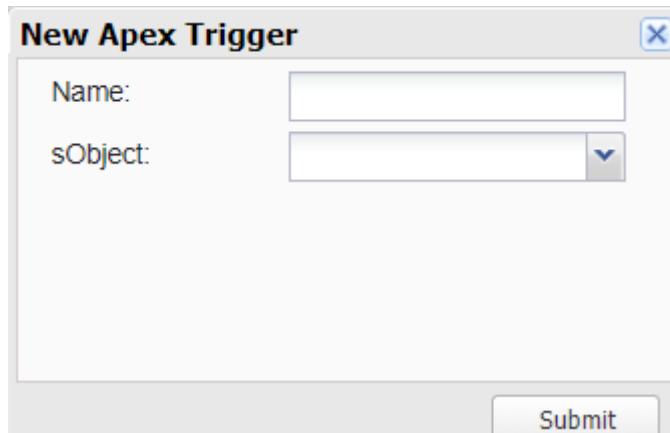
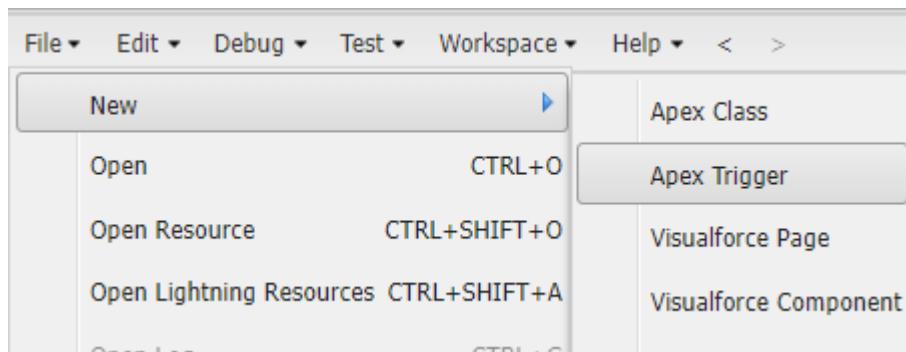
}
}

```

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

The screenshot shows a code editor interface with a menu bar at the top. The menu items include File, Edit, Debug, Test, Workspace, Help, and several navigation icons. Below the menu is a toolbar with a dropdown labeled 'Code Coverage: None' and another labeled 'API Version: 58'. There are two tabs open: 'AmountDistribution.apxt' (highlighted in orange) and 'AmountDistributionHandler.apxc *'. The main code area contains the following Apex trigger:

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

Code:

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

```
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
```

```
        AmountDistributionHandler.amountDist(trigger.new);
```

```
}
```

```
}
```

OUTPUT SCREENSHOT:

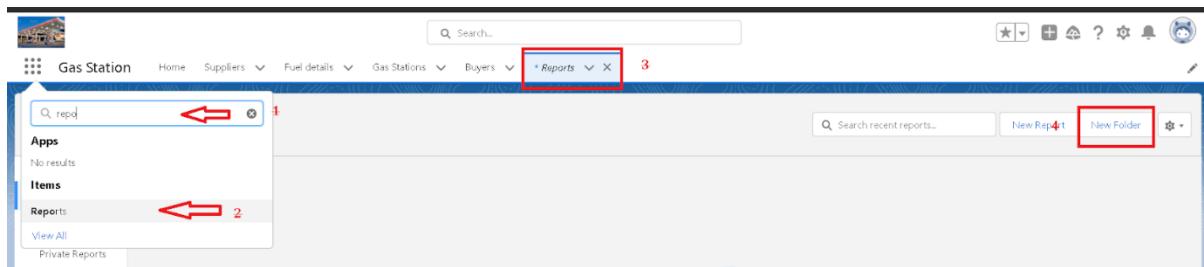
Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

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

create a report folder

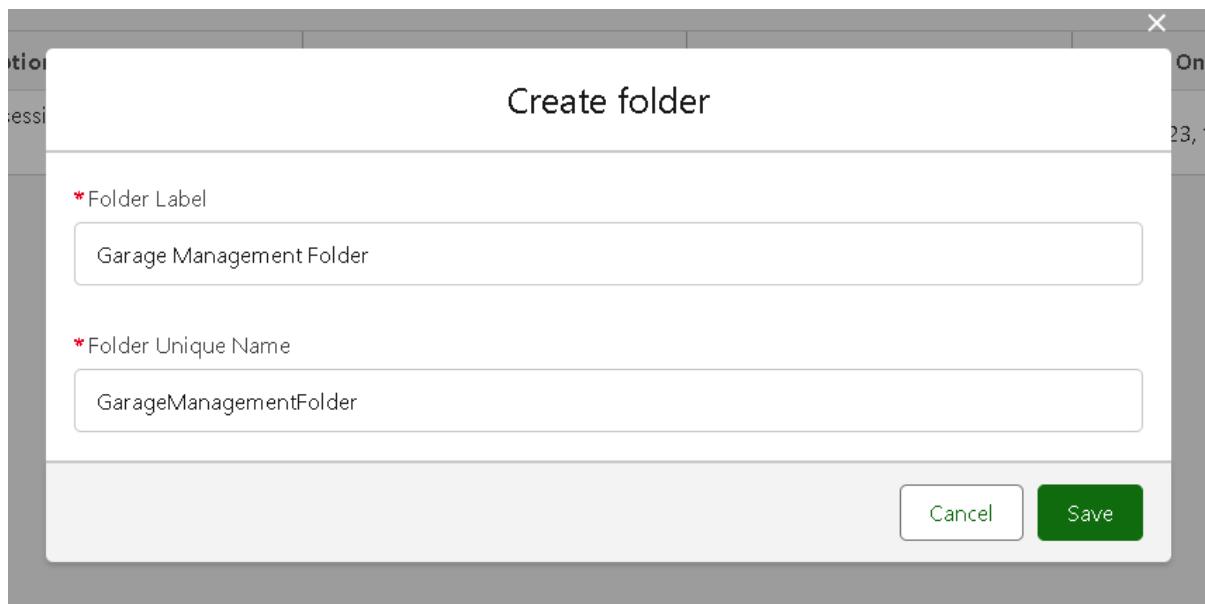
1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.



3. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.

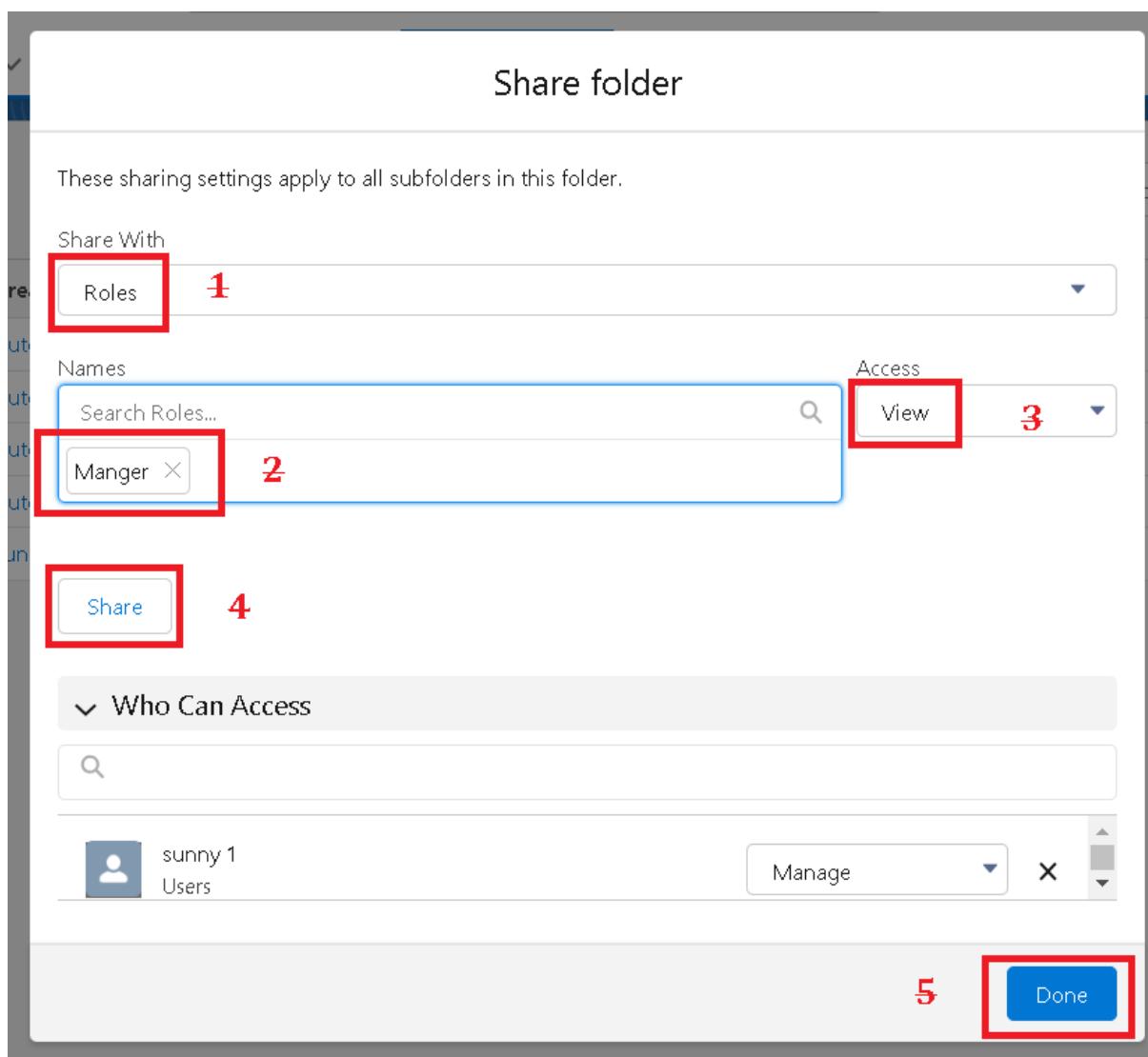
4. Click

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 a 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.

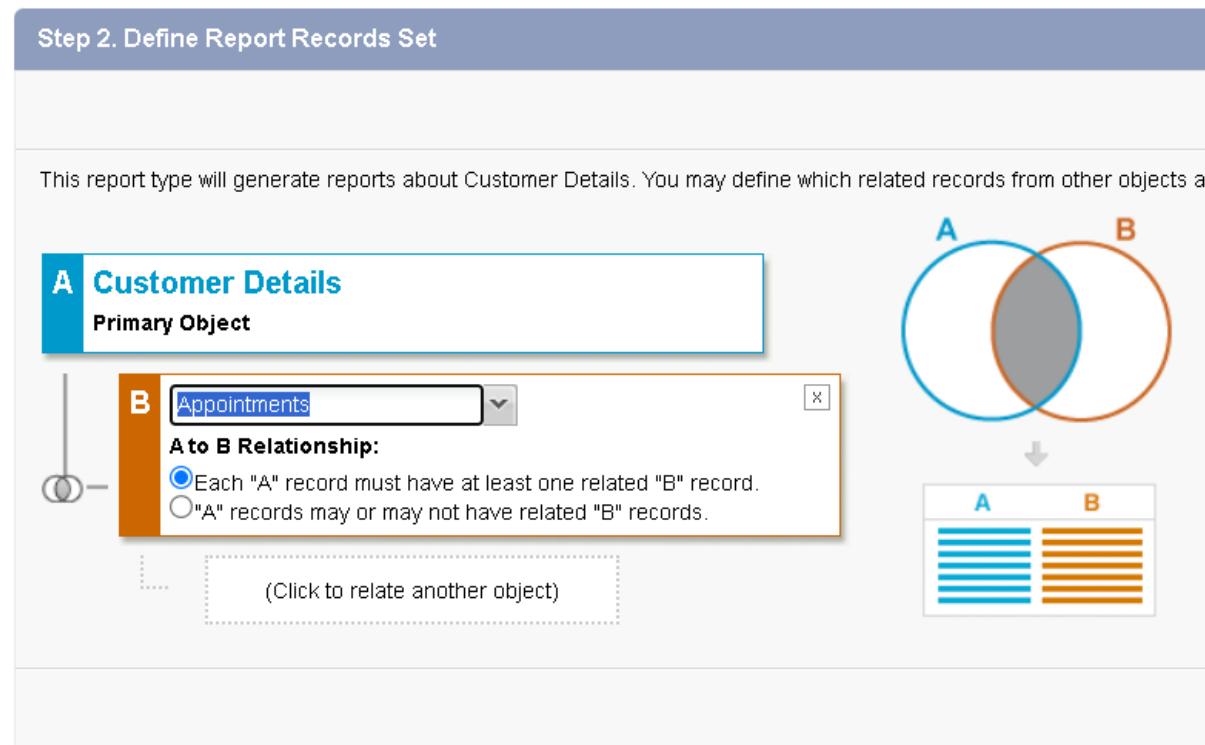
The screenshot shows the Salesforce Setup interface with the 'Report Types' page selected. The left sidebar has a 'Report Types' link highlighted with a green arrow. The main area displays a list of 'All Custom Report Types' with columns for Action, Label, Description, Category, Deployed, Created By Alias, and Created Date. A green arrow points to the 'New Custom Report Type' button at the top right of the list.

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.

The screenshot shows the 'Report Type Focus' configuration page. It includes sections for Identification (Primary Object, Report Type Label, Report Type Name, Description, and Store in Category) and Deployment (Deployment Status). The 'Report Type Label' and 'Report Type Name' fields are both set to 'Service information'. The 'Description' field also contains 'Service information'. The 'Store in Category' dropdown is set to 'Other Reports'. The 'Deployment Status' section shows 'Deployed' selected. A green arrow points to the 'Next' button at the bottom right.

9. now , Click on Related object box.

- Click on Select Object, choose Appointment Object as shown in fig.



- Again Click to relate another object.
- And select the related object as “ service records”.
- Repeat the process and select the related object as “ Billing details and feedback”.
- And click on save.

SETUP Report Types

This report type will generate reports about Customer Details. You may define which related records from other objects are returned in report results by choosing a relationship to another object.

A Customer Details
Primary Object

B Appointments
A to B Relationship:
 Each "A" record must have at least one related "B" record.
 "A" records may or may not have related "B" records.

C Service records
B to C Relationship:
 Each "B" record must have at least one related "C" record.
 "B" records may or may not have related "C" records.

D Billing details and feedback
C to D Relationship:
 Each "C" record must have at least one related "D" record.
 "C" records may or may not have related "D" records.

Object Limit Reached
You can associate up to four objects to a custom report type.

Previous Save Cancel

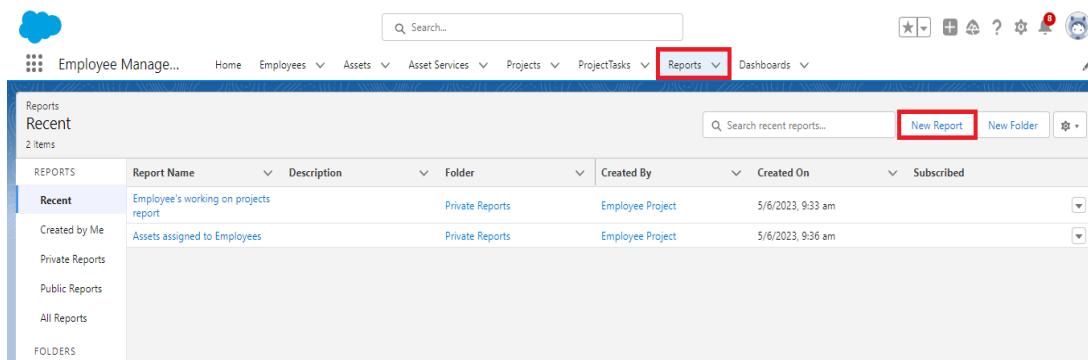
Create report

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

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

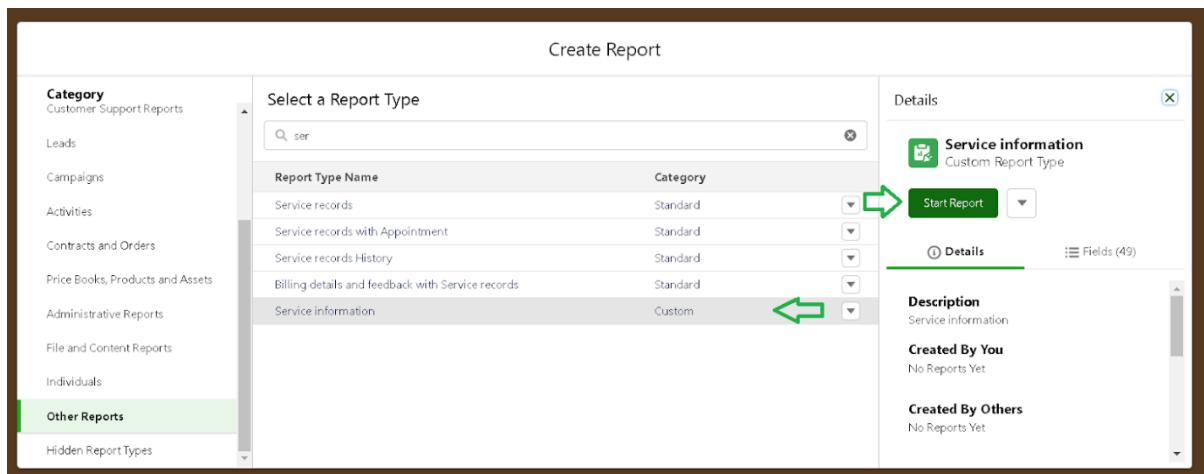
1. Go to the app >> click on the reports tab

2. Click New Report.



The screenshot shows the Employee Management application's interface. At the top, there is a navigation bar with links for Home, Employees, Assets, Asset Services, Projects, ProjectTasks, Reports (which is highlighted with a red box), and Dashboards. Below the navigation bar is a search bar labeled "Search..." and a toolbar with various icons. The main area is titled "Reports" and "Recent". It displays a table with two items: "Employee's working on projects report" and "Assets assigned to Employees". A sidebar on the left lists categories: Reports, Recent, Created by Me, Private Reports, Public Reports, and All Reports. At the bottom of the sidebar is a section for FOLDERS.

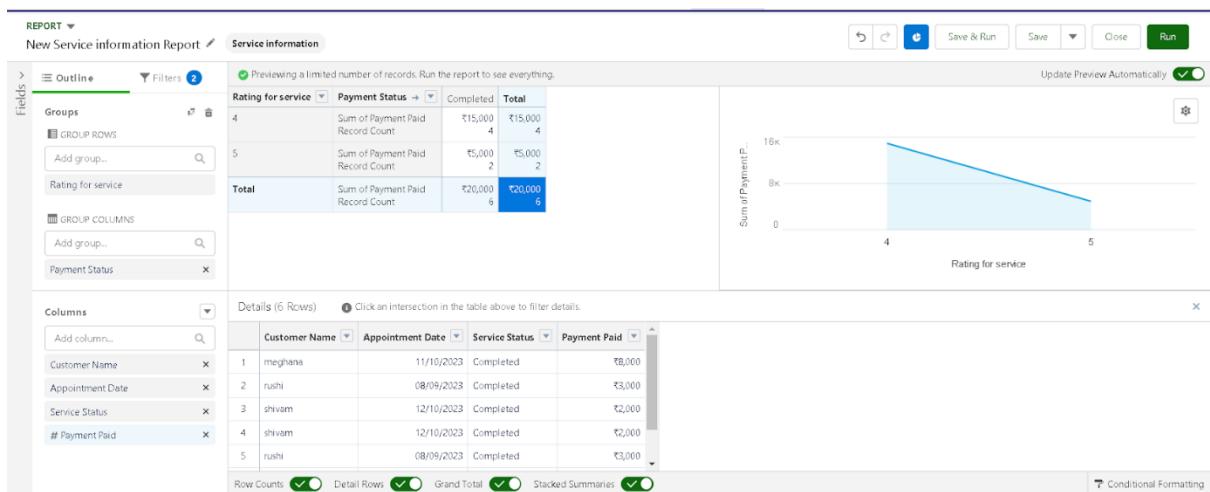
3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.

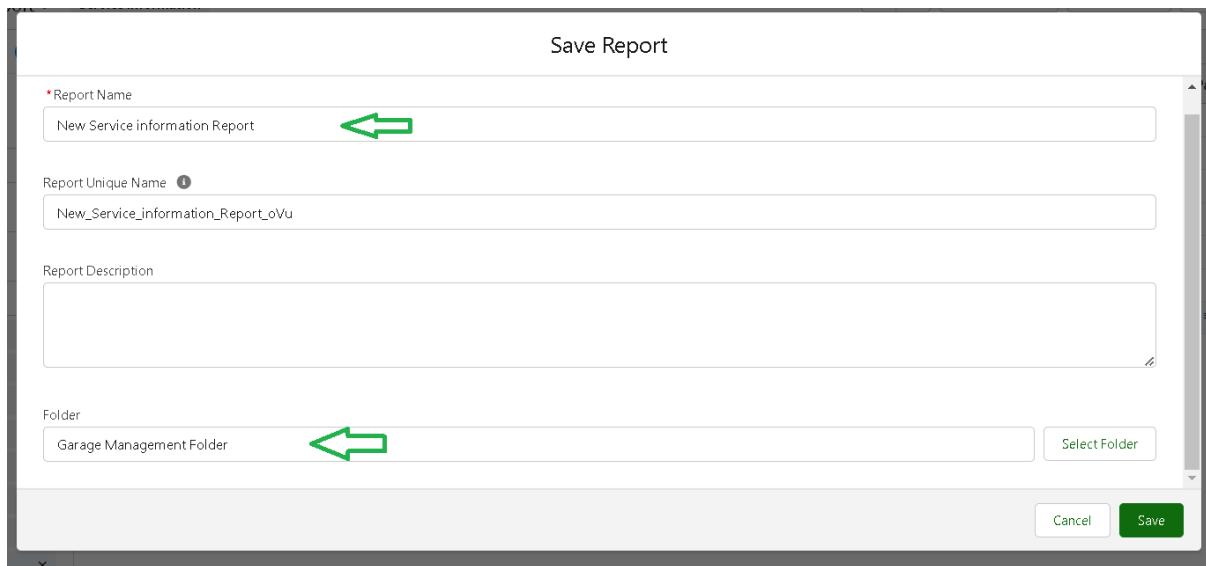


The screenshot shows the "Create Report" dialog box. On the left, there is a sidebar with a tree view under "Category" showing various report types like Customer Support Reports, Leads, Campaigns, etc., with "Other Reports" selected. In the center, there is a "Select a Report Type" panel with a search bar and a list of report types. One item, "Service information", is highlighted with a green arrow pointing to it. On the right, there is a "Details" panel for the selected report type. The "Service information" section is expanded, showing "Custom Report Type", a "Start Report" button, and sections for "Description" (Service information) and "Created By You" (No Reports Yet). Another green arrow points to the "Start Report" button.

4. Their outline pane is opened already, select the fields that mentioned below in column section.

- a. Customer name
 - b. Appointment Date
 - c. Service Status
 - d. Payment paid
 5. Remove the unnecessary fields.
 6. Select the fields that mentioned below in GROUP ROWS section.
 - a. Rating for Service
 7. Select the fields that mentioned below in GROUP ROWS section.
 - a. 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.



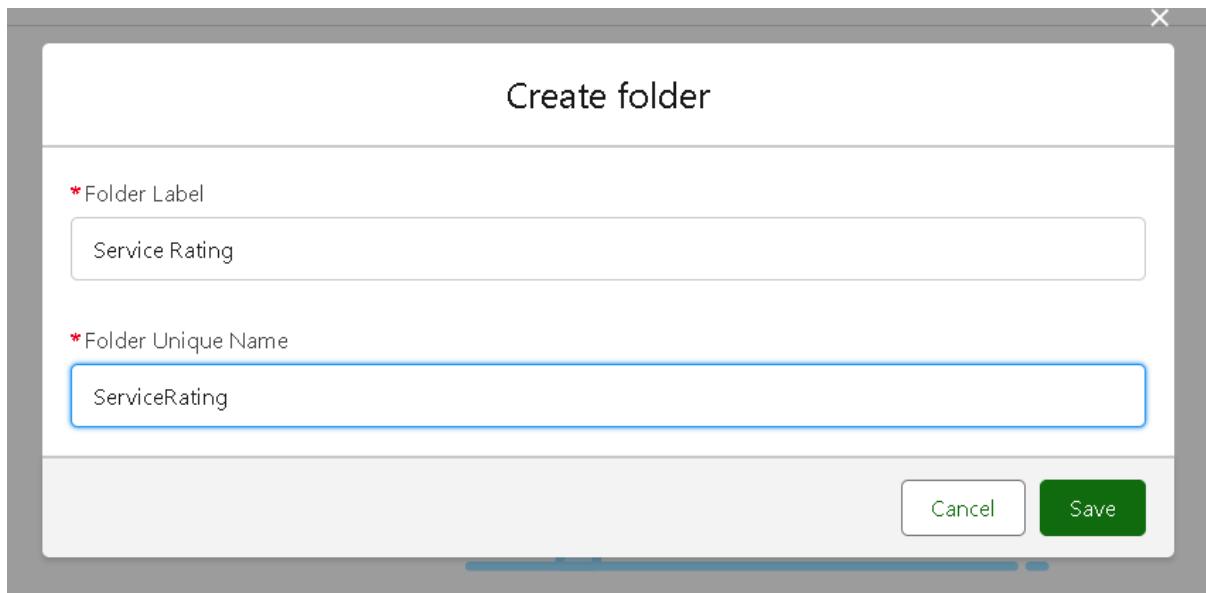


Dashboards

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 a dashboard folder

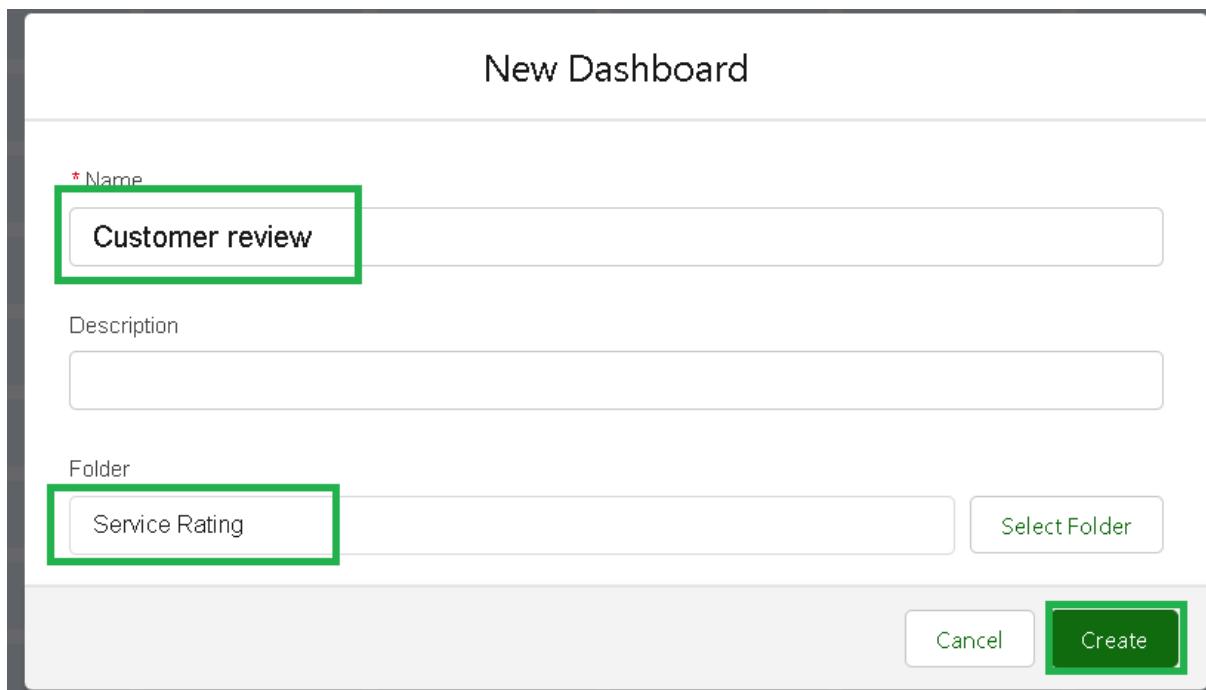
1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “Service Rating dashboard”.
4. Folder unique name will be auto populated.
5. Click save.



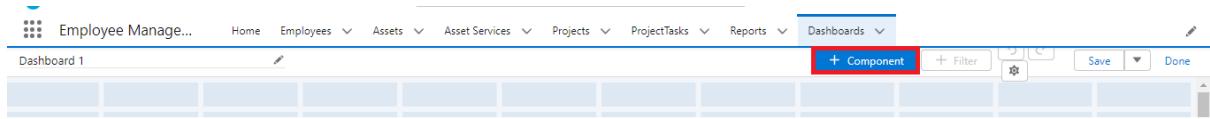
6. Follow the same steps, from milestone 15, and activity 2, and provide the sharing settings for the folder that just created.

Creation 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.

Select Report

Reports

Recent

- Created by Me
- Private Reports
- Public Reports
- All Reports

Folders

- Created by Me
- Shared with Me

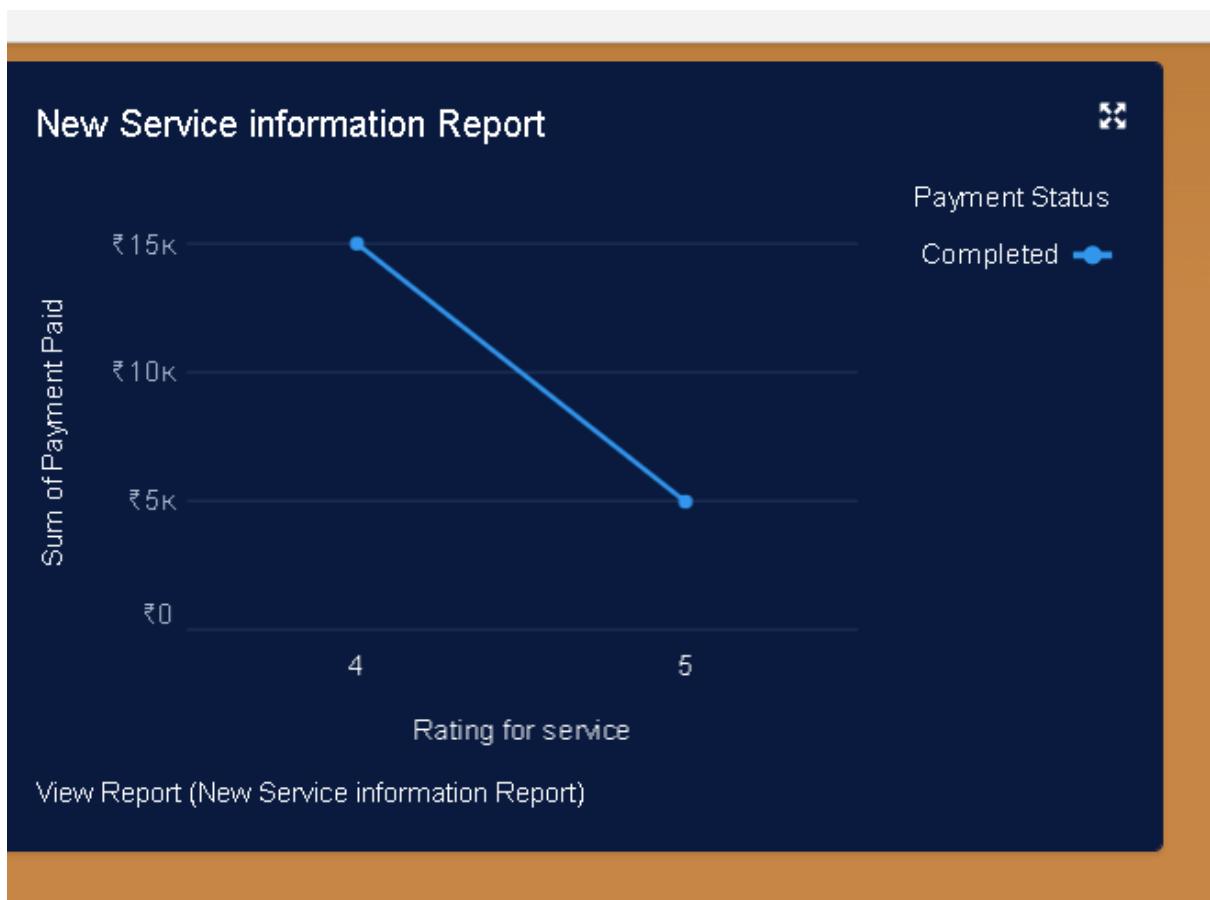
Search Reports and Folders...

New Service Information Report
Project 2 - 16-Oct-2023, 3:20 pm - Garage Management Folder

All Bot Sessions Last 30 days
Automated Process - 28-Aug-2023, 10:04 am - Einstein Bot Reports Summer '23

Cancel 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.

Conclusion :

The Garage Management System (GMS) has been developed as an effective tool to streamline operations in automotive garages. By integrating features such as vehicle registration, service booking, inventory tracking, customer management, and billing, the system significantly reduces the need for manual processes, thereby increasing efficiency and accuracy. This system not only helps staff in tracking and managing daily tasks but also improves customer satisfaction by providing timely service updates, accurate billing, and reminders.

With real-time data tracking and automated reporting, garage managers can make more informed decisions, optimize inventory levels, and manage staff schedules effectively. The GMS's user-friendly interface makes it accessible to all employees, reducing the need for extensive training and enabling faster onboarding. Overall, this system has the potential to elevate the quality of service a garage can offer, enhance operational productivity, and ultimately contribute to business growth.

While the current implementation achieves its primary objectives, there are several areas that could be further enhanced to extend the system's functionality and value

Future Enhancements :

1. Mobile Application Integration

- Developing a mobile app for customers to schedule appointments, view service history, and receive notifications directly on their smartphones. This would make the GMS more accessible to customers and help them stay informed about their service status.

2. Enhanced Reporting and Analytics

- Adding more advanced analytics capabilities to provide predictive insights into inventory needs, customer behavior, and seasonal service demands. Machine learning models could help in forecasting high-demand parts or predicting potential service needs based on historical data.

3. Online Payment Integration

- Integrating online payment gateways (like PayPal or Stripe) to allow customers to pay bills directly through the system. This would provide greater convenience to customers and streamline the billing and payment process.

4. Customer Loyalty and Rewards Program

- Introducing a loyalty program to reward repeat customers. By tracking customer visits and spending, the system could offer discounts, special offers, or points, encouraging customer retention and enhancing the customer experience.

5. Integration with Third-Party Services

- Enabling integrations with third-party suppliers for automated inventory replenishment based on predefined stock levels. This would ensure that parts are ordered automatically, reducing the risk of stockouts and improving turnaround times for repairs.

6. Automated Service Reminders and Follow-Up

- Enhancing the system's notification capabilities by sending automated reminders for upcoming service appointments, annual inspections, or

maintenance requirements. Additionally, the system could send follow-up messages after services to collect customer feedback.

7. AI-Powered Diagnostics

- Incorporating an AI-driven diagnostics tool that assists mechanics by suggesting possible issues based on symptoms entered into the system. This could speed up the diagnostic process, ensuring faster and more accurate repairs.

8. Integration with IoT Devices

- Using IoT (Internet of Things) technology to connect with vehicles directly, providing real-time health monitoring of vehicles. This would enable predictive maintenance alerts for customers and help garages prepare for incoming service requirements.

9. Scalability for Multi-Branch Management

- Expanding the system to support multiple garage branches within the same organization, allowing centralized management, unified customer records, and resource allocation across locations.

10. Data Backup and Cloud Storage

- Moving data storage to the cloud, ensuring data safety through regular backups and enabling system access from multiple locations. This would also increase data security and accessibility for both staff and customers.