

Garage Management System Using Salesforce

College Name: Sri Krishna Adithya College of Arts and Science

College Code: bruag

Team ID: NM2025TMID22779

Team Members: 4

Team Leader Name: HARI KISAN SURJITH BS

Email: 23bsds166harikisansurjithbs@skacas.ac.in

Team Member 1: AJAY RATHNAVEL K

Email: 23bsds103ajayrathnavelk@skacas.ac.in

Team Member 2: MOHAMMED SUHAIL AKTHAR J

Email: 23bsds135mohammedsuhailaktharj@skacas.ac.in

Team Member 3: MOHAMMED ASIF ,

Email: 23bsds134mohammedasifa@skacas.ac.in

Garage Management System Using Salesforce

Project Overview: The Garage Management System is a valuable tool designed for automotive repair facilities to help them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. This project uses the Salesforce platform to create a comprehensive application that manages the entire customer and vehicle service lifecycle. The system connects the garage's staff with customer details, appointments, service records, and billing information in one centralized location. Through the use of custom objects, automated flows, Apex triggers, and insightful reports, the process of managing a garage is streamlined and made more efficient, ensuring a seamless experience for both customers and staff.

Objectives:

- **Increase Operational Efficiency:** Streamline garage operations from appointment booking to final billing.
- **Enhance Service Delivery:** Provide staff with the tools and information needed to deliver high-quality, consistent service to every customer.
- **Build Lasting Customer Relationships:** Maintain a comprehensive record of customer interactions and service history to foster loyalty and trust.
- **Ensure Data Accuracy and Integrity:** Use validation and duplicate rules to maintain clean and reliable data for customers and their vehicles.

- **Provide Actionable Insights:** Leverage reports and dashboards to track service performance, revenue, and customer feedback for informed decision-making.
- **Automate Key Processes:** Reduce manual work by automating service amount calculation, status updates, and customer notifications.

Student Outcomes:

- **Hands-on Experience with Salesforce Platform:** Gained practical skills in building a complete application by creating custom objects, fields, and relationships.
- **Proficiency in Business Process Automation:** Learned to automate complex logic using record-triggered Flows and Apex triggers to solve real-world business problems.
- **Understanding of Salesforce Security Model:** Developed a strong understanding of data security by configuring Profiles, Roles, and Sharing Rules to control user access.
- **Enhanced Analytical and Reporting Skills:** Gained experience in creating custom report types, building detailed reports, and visualizing data with dashboards to provide business insights.
- **End-to-End Project Lifecycle Experience:** Learned the complete process from data model design and UI configuration to testing and deployment of a functional Salesforce application.

System Requirements

- **Hardware Requirements:**
 - Computer with a modern processor
 - Minimum 4 GB RAM
 - A stable internet connection

- **Software Requirements:**

- Salesforce Developer Edition Org
- A modern web browser (e.g., Google Chrome, Firefox)

Project Phases Overview:

Phase No.	Phase Name	Description	Page Numbers
1	Requirement Analysis & Planning	Defining project goals, identifying key features for a garage, and planning the data model and user interactions.	5
2	Salesforce Development - Backend & Configurations	Creating the foundational data structure including custom objects, fields, and relationships to store all garage-related data.	5-13
3	UI/UX Development & Customization	Building the user-facing components, including creating tabs and a custom Lightning App to ensure an intuitive user experience.	13-15
4	Automation, Security & Analytics	Implementing business logic with Flows and Apex, securing data with profiles and sharing rules, and creating reports and dashboards for analysis.	15-59
5	Deployment, Documentation & Maintenance	Finalizing the application, performing end-user testing by creating records, and documenting the project for future maintenance and scalability.	60-62

Phase 1: Requirement Analysis & Planning

This initial phase involved defining the core needs of a modern auto garage. The primary requirement was to create a unified system to manage customer information, vehicle appointments, the services performed on vehicles, and the associated billing and feedback processes. The plan focused on creating a relational data model that logically connects these entities for efficient data management and reporting.

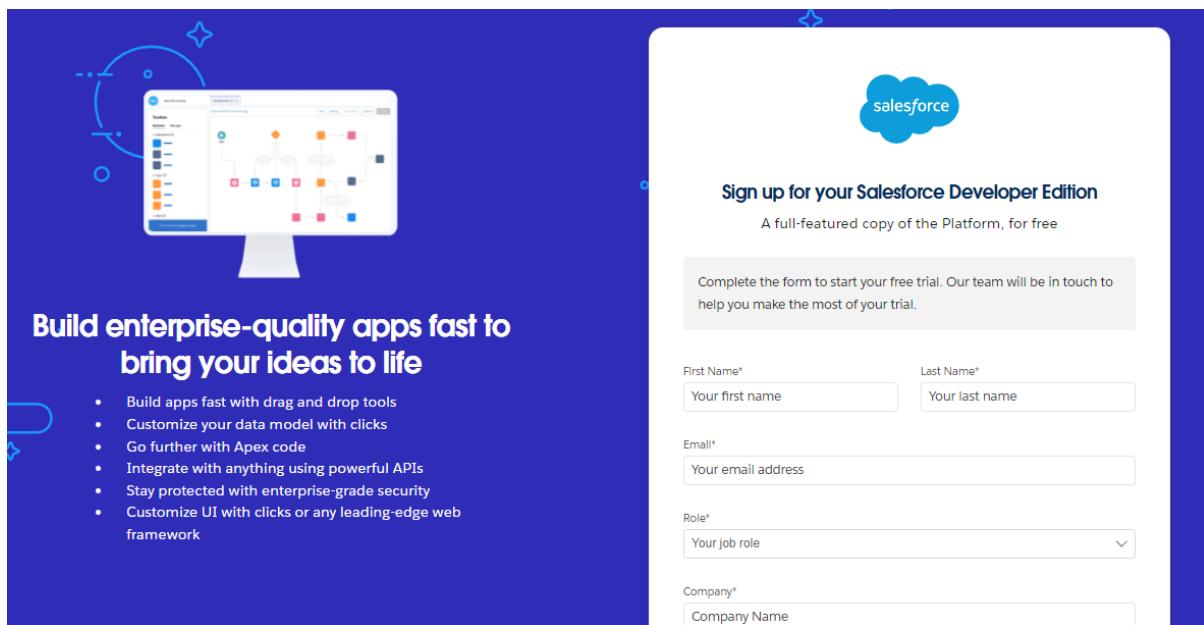
Phase 2: Salesforce Development – Backend & Configurations:-

Milestone 1: Salesforce developer account creation

Activity 1: Creating Developer Account

Creating a developer org in salesforce

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



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

- i. First name & Last name
- ii. Email
- iii. Role : Developer

- iv. Company : College or Company Name
- v. County : India
- vi. Postal Code : pin code
- vii. Username : should be a combination of your name and company

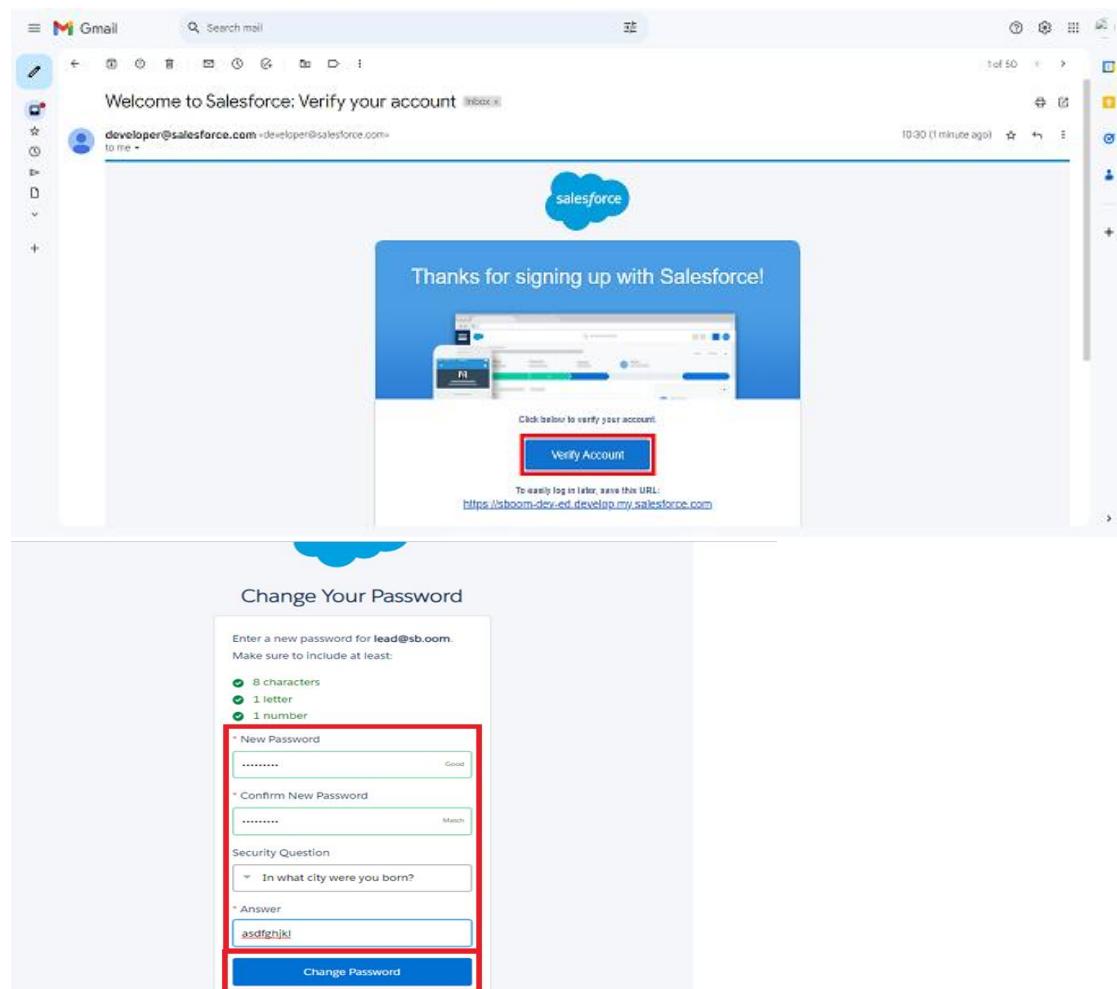
This need not be an actual email id, you can give anything in the format :

username@organization.com

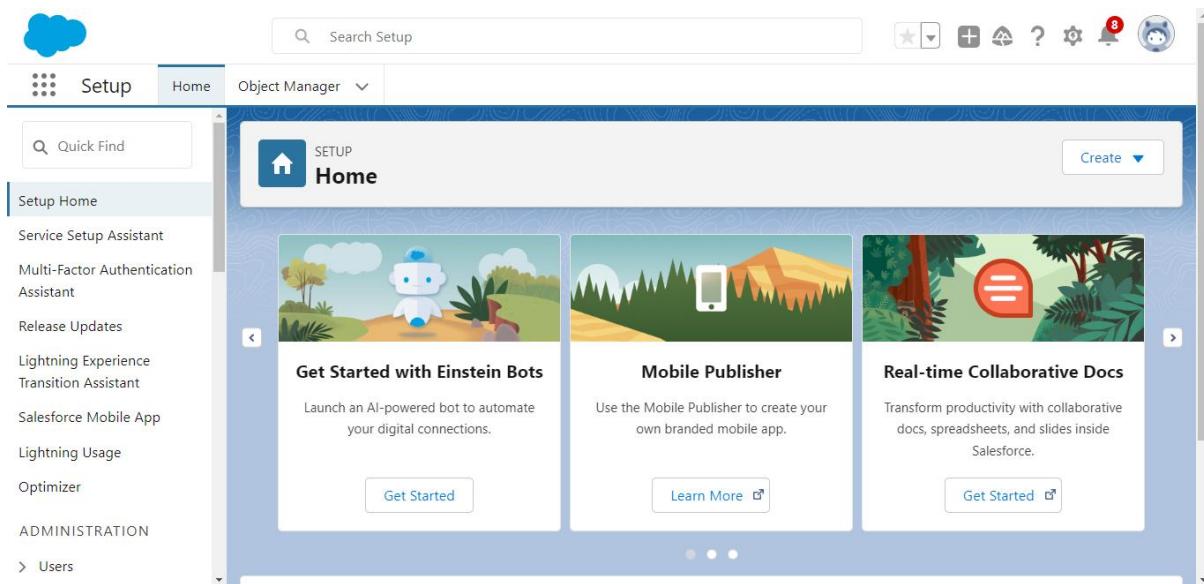
Click on sign me up after filling these.

Activity 2: 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.



1. Click on Verify Account
2. Give a password and answer a security question and click on change password.
3. Then you will redirect to your salesforce setup page.



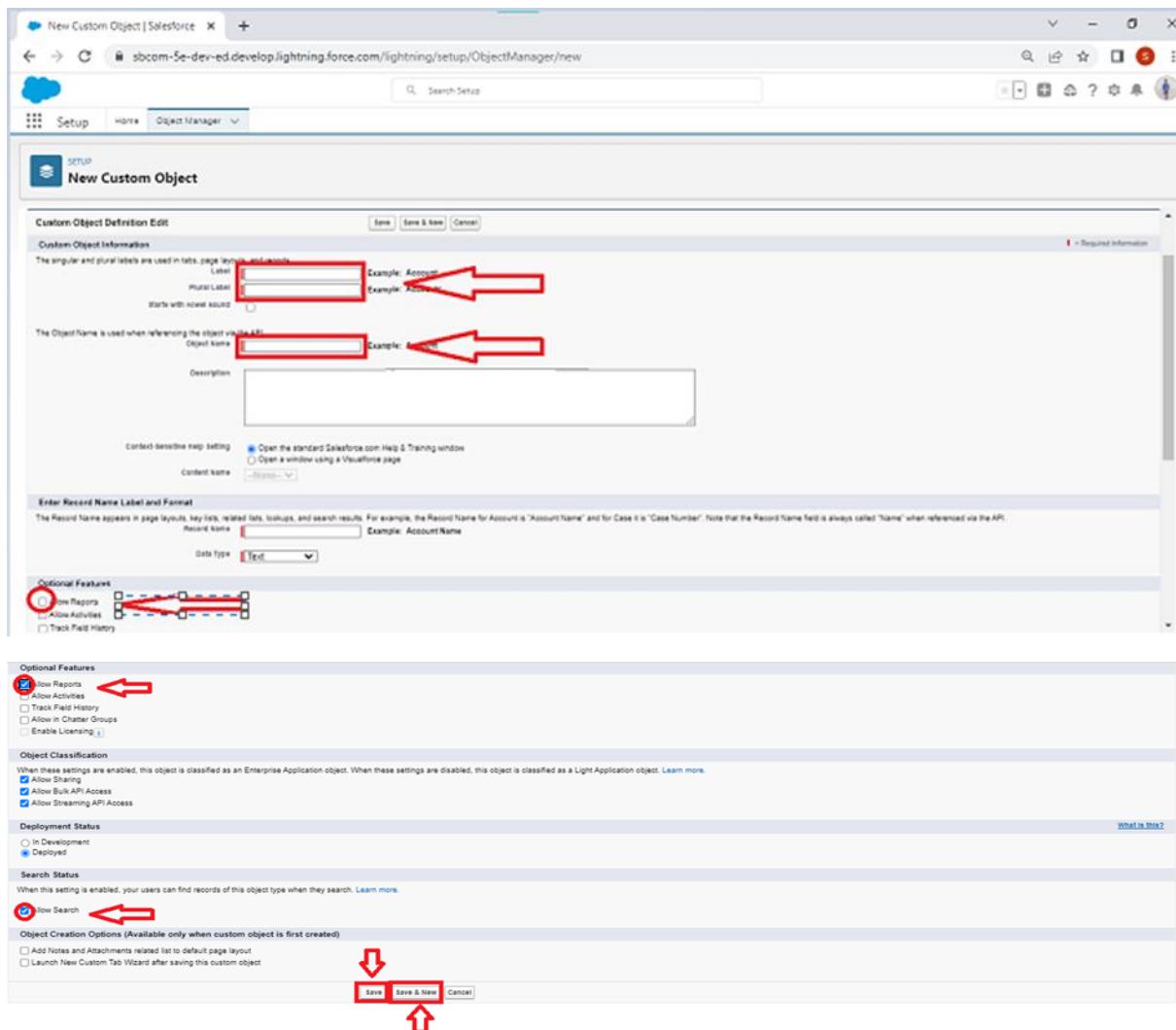
Milestone 2: OBJECTS

To create an object:

1. From the setup page Click on Object Manager Click on Create Click on Custom Object.



2. On Custom object defining page:
3. Enter the label name, plural label name, click on Allow reports, Allow search.



4. Click on Save.

Activity 1: Create Customer Details Object

Create Customer Details 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>>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.

Activity 2: 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.

Activity 3: 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.

Activity 4: Create Billing details and feedback Object

To create an object:

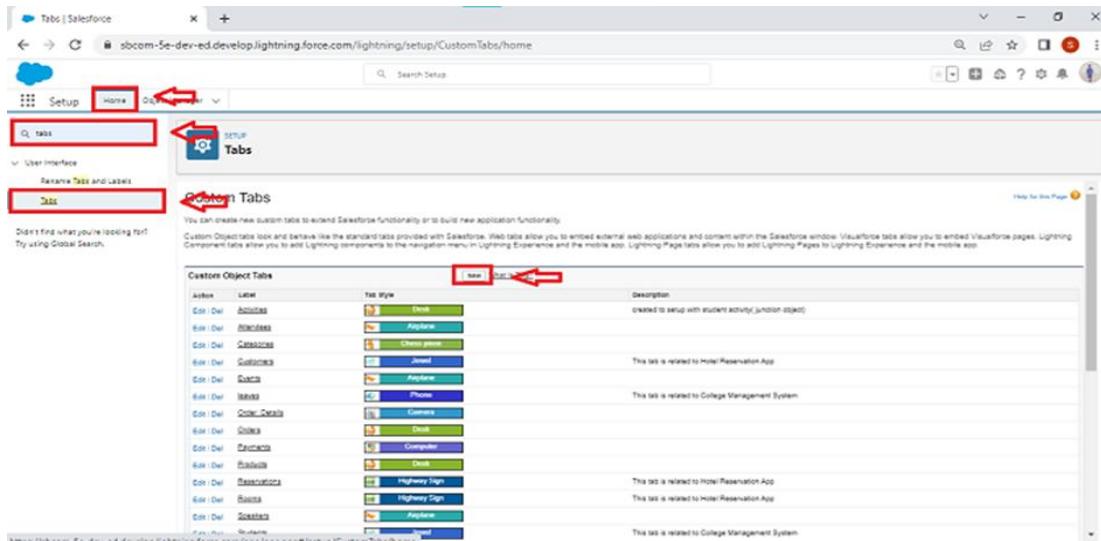
1. From the setup page>>Click on Object Manager>>Click on Create>>Click on Custom Object.
2. Enter the label name>>Billing details and feedback
3. Plural label name>>Billing details and feedback
4. Enter Record Name Label and Format
5. Record Name>>Billing details and feedback Name
6. Data Type>>Auto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

Milestone 3: TABS

Activity 1: Creating Custom Tabs

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

Step 1. Enter the Details

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	<input type="text" value="Customer Details"/>
Tab Style	<input type="button" value=""/>

(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	<input type="button" value="None"/>
-------------------------	-------------------------------------

Enter a short description

Description	<input type="text"/>
-------------	----------------------

[Next](#) [Cancel](#)

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

Save **Cancel**

Step 3. Add to Custom Apps

Step 3 of 3

Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each CustomApp.

Custom App	<input type="checkbox"/> Include Tab
Platform (standard__Platform)	<input type="checkbox"/>
Sales (standard__Sales)	<input type="checkbox"/>
Service (standard__Service)	<input type="checkbox"/>
Marketing (standard__Marketing)	<input type="checkbox"/>
Sample Console (standard__ServiceConsole)	<input type="checkbox"/>
High Volume Customer Portal User	<input type="checkbox"/>
Authenticated Website User	<input type="checkbox"/>
App Launcher (standard__AppLauncher)	<input type="checkbox"/>
Analytics Studio (standard__Insights)	<input type="checkbox"/>
Sales Console (standard__LightningSalesConsole)	<input type="checkbox"/>
Service Console (standard__LightningService)	<input type="checkbox"/>
Sales (standard__LightningSales)	<input type="checkbox"/>
Lightning Usage App (standard__LightningInstrumentation)	<input type="checkbox"/>
Digital Experiences (standard__SalesforceCMS)	<input type="checkbox"/>
Queue Management (standard__QueueManagement)	<input type="checkbox"/>
Bolt Solutions (standard__LightningBolt)	<input type="checkbox"/>
Data Manager (standard__DataManager)	<input type="checkbox"/>
Salesforce Scheduler Setup (standard__LightningScheduler)	<input type="checkbox"/>

Append tab to users' existing personal customizations

Previous **Save** **Cancel**

Activity 2: Creating Remaining Tabs

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.

Phase 3: UI/UX Development & Customization

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's a search bar with "Search Setup" and a "New" button. Below it, there are tabs for "Setup", "Home", and "Object Manager". A red box highlights the search bar, and another red box highlights the "New" button. In the center, there's a section titled "Lightning Experience App Manager" with a "Clone Apps(Beta)" link. A third red box highlights this link. On the right, there are buttons for "New Lightning App" and "New Connected App". Below these sections, there's a message about cloning existing apps. At the bottom, there's a table listing 35 apps, with columns for "App Name", "Developer Name", "Description", "Last Modified", "App Type", and "VLR". The table includes entries like "All Tabs", "Analytics Studio", "App Launcher", "Bolt Solutions", "Chatter Desktop", "Chatter Mobile for BlackBerry", "College Management System", "Community", "Content", and "Data Manager".

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.

New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

* App Name

* Developer Name

Description

App Branding

Image Primary Color Hex Value

Org Theme Options Use the app's image and color instead of the org's custom theme

App Launcher Preview

Next

3. To Add Navigation Items:

New Lightning App

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

Available Items

- Accounts
- Activities
- Alert Settings
- All Sites
- Alternative Payment Methods
- App Launcher
- Appointment Invitations

Selected Items

No items selected

Next

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:

New Lightning App

User Profiles

Choose the user profiles that can access this app.

Available Profiles

- Analytics Cloud Integration User
- Analytics Cloud Security User
- Authenticated Website
- Authenticated Website
- Contract Manager
- Cross Org Data Proxy User
- Custom: Sales Profile

Selected Profiles

No Profiles selected

Save & Finish

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

Phase 4: Automation, Security & Analytics

Milestone 5: FIELDS

Activity 1: Creation of fields for the Customer Details object

To create fields in an object:

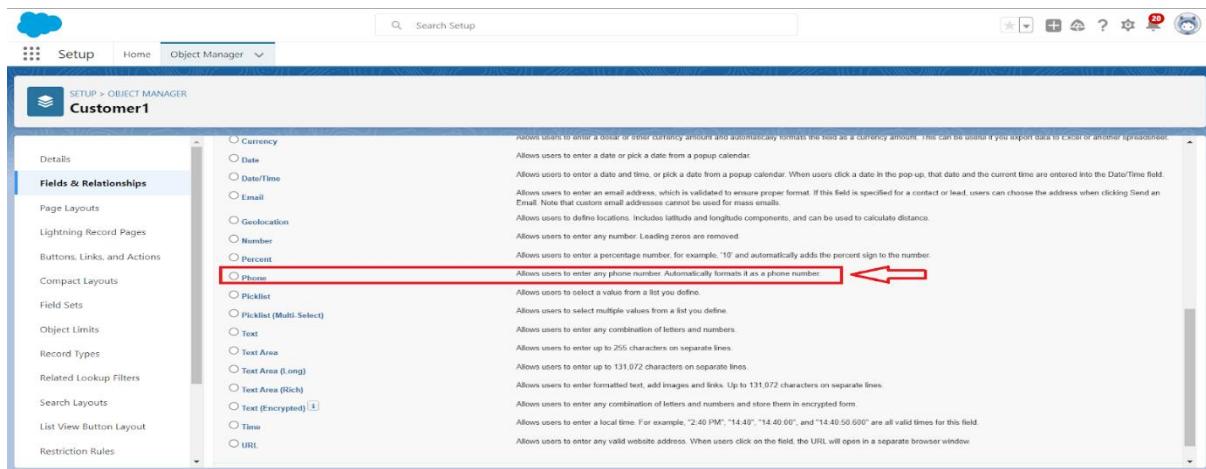
1. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.

The screenshot shows the Salesforce Object Manager interface. The search bar at the top contains 'cus'. A red box highlights the 'Customer Details' row in the list, which has 'Customer_Details__c' as its API name and 'Custom Object' as its type. The 'Customer' object is also listed above it.

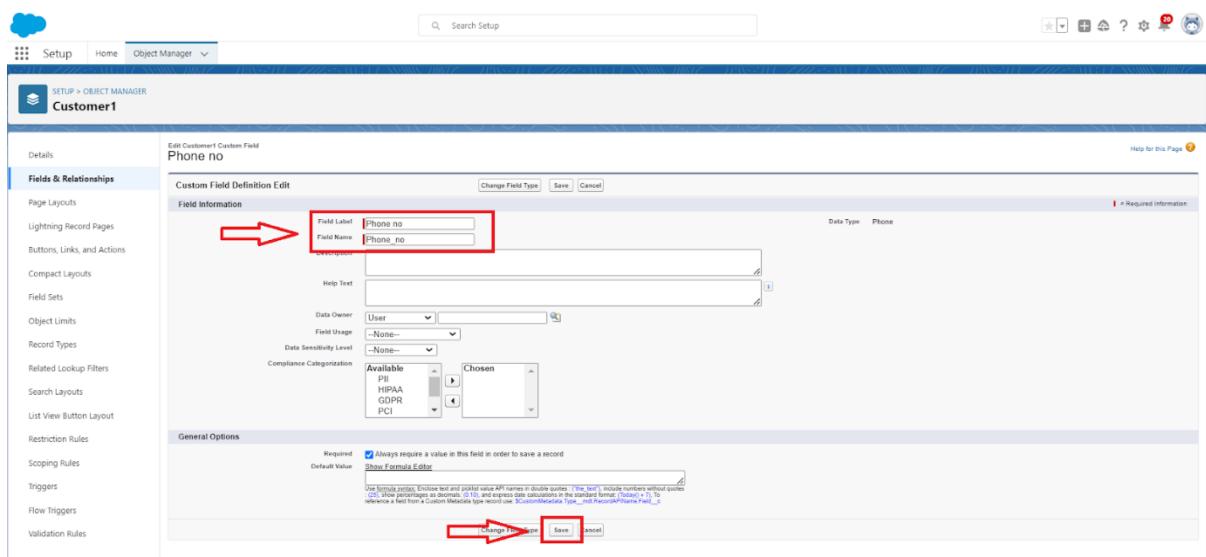
2. Now click on “Fields & Relationships” >> New

The screenshot shows the 'Fields & Relationships' page for the 'Customer1' object. A red box highlights the 'Customer1' label in the breadcrumb navigation. Another red box highlights the 'Fields & Relationships' link in the left sidebar. A third red box highlights the 'New' button in the top right corner of the main table area. The table lists various fields like 'Created By', 'current_Status', etc., with their respective field labels, names, data types, and other properties.

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.

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.

Activity 2: 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', 'Object Manager', and a search bar with placeholder 'app'. Below the header, the title 'Object Manager' is displayed with a green icon. A sub-header indicates '7 Items, Sorted by Label'. The main table lists four objects: Appointment, Appointment Category, Appointment Invitation, and Appointment Invitee. The 'Appointment' row is highlighted with a red border. The columns are labeled: LABEL, API NAME, TYPE, DESCRIPTION, LAST MODIFIED, and DEPLOYED. The 'Appointment' entry has 'Appointment__c' in the API Name column and 'Custom Object' in the Type column.

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

The screenshot shows the 'Fields & Relationships' section for the Appointment object. The left sidebar has tabs for 'Details', 'Fields & Relationships' (which is selected and highlighted with a green background), 'Page Layouts', and 'Lightning Record Pages'. The main area displays a table titled 'Fields & Relationships' with 14 items. The first two rows shown are 'Appointment Date' (Field Label: Appointment_Date__c, Data Type: Date) and 'Appointment Name' (Field Label: Appointment_Name, Data Type: Auto Number). The table includes columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. A red box highlights the 'New' button at the top right of the table.

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

The screenshot shows the 'Data Type' selection screen. At the top, it says 'Specify the type of information that the custom field will contain.' and has 'Next' and 'Cancel' buttons with a red arrow pointing to 'Next'. Below this, there's a 'Data Type' section with a table. The first row has a radio button 'None Selected' and a note 'Select one of the data types below'. The second row has a radio button 'Auto Number' and a note 'A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.'. The third row has a radio button 'Formula' and a note 'A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.'. The fourth row has a radio button 'Roll-Up Summary' and a note 'A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.'. The fifth row has a radio button 'Lookup Relationship' (which is selected and highlighted with a red circle) and a note '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.'. The sixth row has a radio button 'Master-Detail Relationship' and a note 'Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master") where:

- The relationship field is required on all detail records.
- 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.

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.

Lookup Options

Related To: Appointment

Related List Label: Service records

Child Relationship Name: Service_records

Required: Always require a value in this field in order to save a record

What to do if the lookup record is deleted?: Clear the value of this field. You can't choose this option if you make this field required. Don't allow deletion of the lookup record that's part of a lookup relationship.

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.

Lookup Filter

Optional, create a filter to limit the records available to users in the lookup field. [Tell me more!](#)

Hide Filter Settings

Filter Criteria

Field	Operator	Value / Field
Appointment: Appointment Date	less than	Field: Appointment: Created Date

Clear Filter Criteria

Filter Type: Required. The user-entered value must match filter criteria.
If it doesn't, display this error message on save:
Value does not exist or does not match filter criteria.

Optional. The user can remove the filter or enter values that don't match criteria.

Lookup Window Text: Add this informational message to the lookup window.

Active: Enable this filter.

10. Error Message : Value does not match the criteria.
11. Enable the filter by click on Active.
12. Next >> Next >> Save.

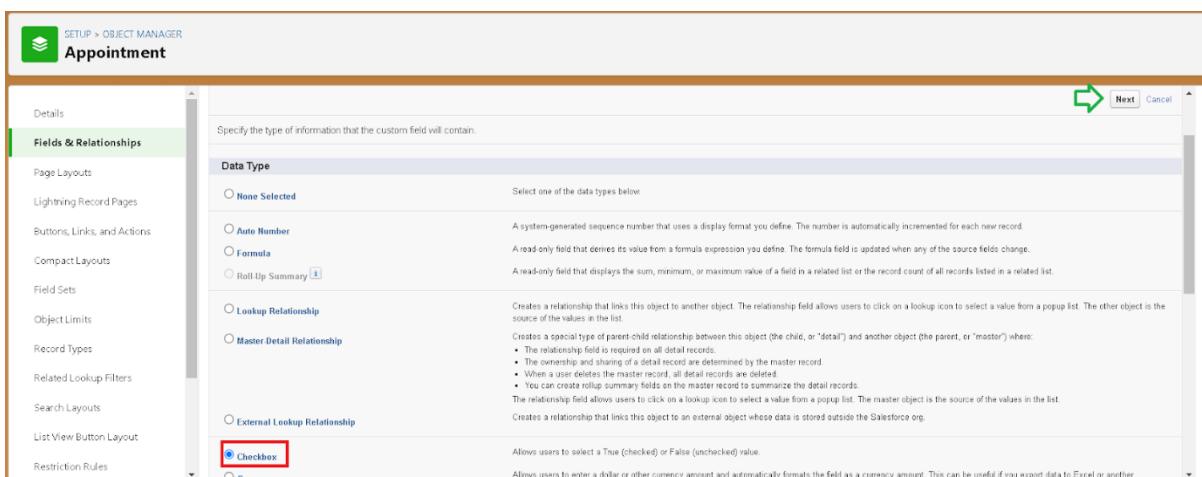
Creation of Lookup Field on Billing details and feedback Object :

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.

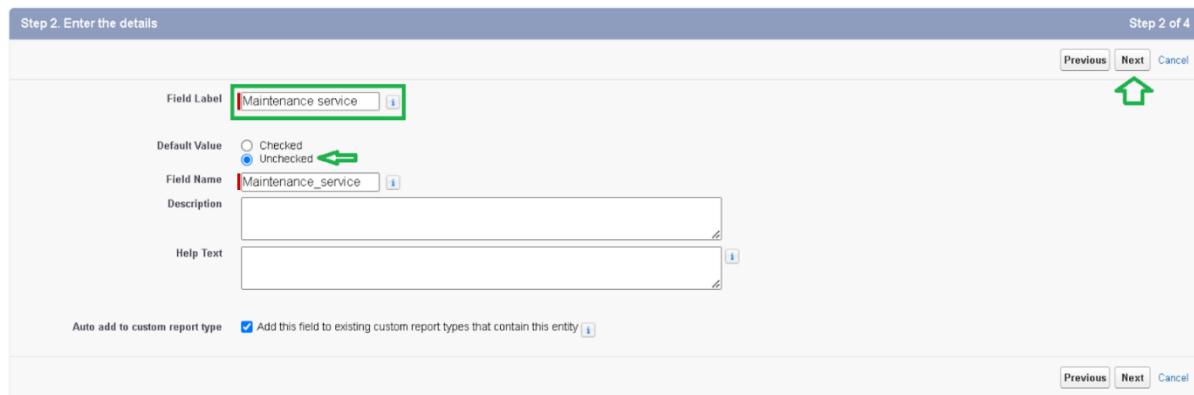
Activity 3: Creation of Checkbox Fields

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: Maintenance service 

Default Value: Checked Unchecked 

Field Name: Maintenance_service 

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 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.

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

Activity 4: 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. Enter the details Step 2 of 4

Previous Next Cancel

Field Label

Field Name

Description

Help Text

Required Always require a value in this field in order to save a record

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

Default Value Show Formula Editor

Activity 5: 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 Name : 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 specified as 18 digits, 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 an unchecked checkbox for 'Always require a value in this field in order to save a record'. Below that, there is a checked checkbox for 'Add this field to existing custom report types that contain this entity'. Navigation buttons at the top right include 'Previous', 'Next', and 'Cancel'.

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 is 'Service Amounts', Data Type is 'Currency', Field Name is 'Service_Amounts', and Description is empty. Below this, it asks to select profiles for edit access. A table lists various profiles with checkboxes for 'Visible' and 'Read-Only'. A green arrow points to the 'Read-Only' column header, indicating that the 'Read-Only' checkbox should be selected for all profiles. Navigation buttons at the top right include 'Previous', 'Next', and 'Cancel'.

Field Level Security for Profile	Visible	Read-Only
Analytics Cloud Integration User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Analytics Cloud Security User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Authenticated Website	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Authenticated Website	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Contract Manager	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cross Org Data Proxy User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

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' screen for creating a new field. The field label is 'Vehicle number plate'. The length is set to 10. The field name is 'Vehicle_number_plate'. The field is marked as required and unique. The help text and description fields are empty. The status bar at the top right indicates 'Step 2 of 4'.

Field Label	Vehicle number plate
Length	10
Field Name	Vehicle_number_plate
Description	(empty)
Help Text	(empty)
Required	<input checked="" type="checkbox"/> Always require a value in this field in order to save a record
Unique	<input checked="" type="checkbox"/> Do not allow duplicate values <input type="radio"/> Treat "ABC" and "abc" as duplicate values (case insensitive) <input type="radio"/> Treat "ABC" and "abc" as different values (case sensitive)
External ID	<input type="checkbox"/> Set this field as the unique record identifier from an external system
Auto add to custom report type	<input checked="" type="checkbox"/> 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.
8. Click on next >> next >> save

Activity 6: 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.

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.

Step 2. Choose output type

Field Label Field Name Up

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

Formula Return Type

None Selected Select one of the data types below.

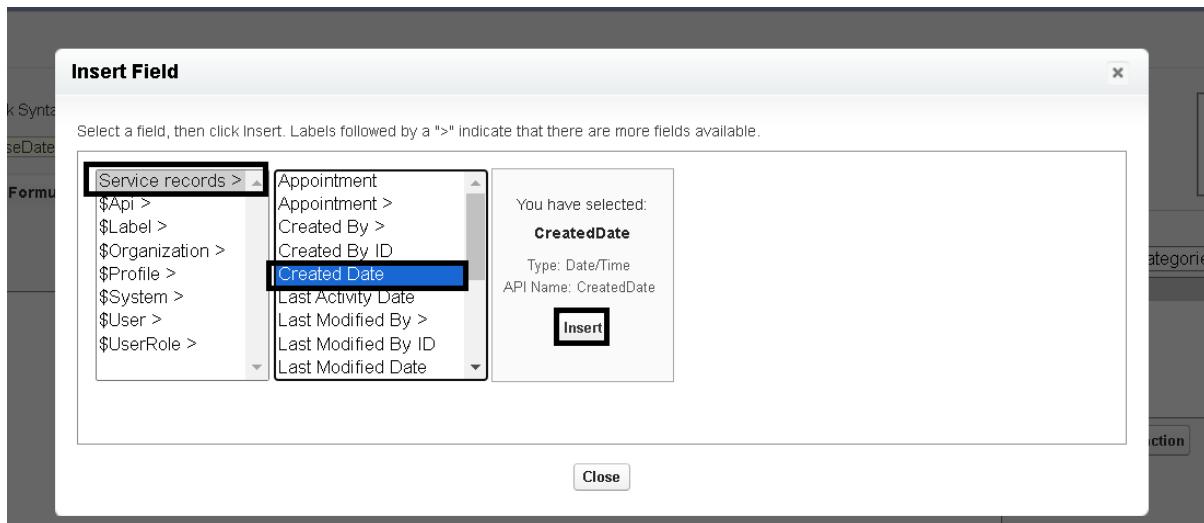
Checkbox Calculate a boolean value.
Example: `TODAY() > CloseDate`

Currency Calculate a dollar or other currency amount and automatically format the field as a currency amount.
Example: `Gross Margin = Amount - Cost__c`

Date Calculate a date, for example, by adding or subtracting days to other dates.
Example: `Reminder Date = CloseDate - 7`

Date/Time Calculate a date/time, for example, by adding a number of hours or days to another date/time.
Example: `Next = NOW() + 1`

5. Insert field formula should be : CreatedDate



Step 3. Enter formula

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.

Example: `Reminder Date = CloseDate - 7` [More Examples...](#)

Simple Formula Advanced Formula

service dates (Date) = CreatedDate Up

Insert Operator Up

Functions Up

ABS
ACOS
ADDMONTHS
AND
ASCII

Up

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

Milestone 6: VALIDATION & DUPLICATE RULES

Activity 1: Create Validation Rules

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 Setup interface for the 'Appointment' object. On the left, there's a sidebar with various configuration options like Buttons, Links, and Actions, Field Sets, Object Limits, etc. The 'Validation Rules' option is highlighted with a green box. The main area shows a table titled 'Validation Rules' with one item listed: 'Vehicle' (Rule Name), 'Vehicle number plate' (Error Location), and 'Please enter valid number' (Error Message). A 'New' button at the top right of the table is also highlighted with a green box.

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}"))

The screenshot shows the 'Validation Rule Edit' screen. The 'Rule Name' field is set to 'Vehicle'. The 'Active' checkbox is checked. The 'Description' field contains 'vehicle'. In the 'Error Condition Formula' section, the formula 'NOT (REGEX (Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))' is entered. A dropdown menu for functions is open, showing various mathematical and string manipulation functions like ABS, ACOS, ADDMONTHS, AND, ASCII, ASIN, etc. A 'Check Syntax' button at the bottom left is highlighted with a green box.

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.



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.

Activity 2: Create Duplicate Rules

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
	Gmail	Exact
	Phone Number	Exact

8. Click save.

9. After Saving Click on Activate.

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	AND
Phone Number	Exact	AND
—None—	Exact	AND
—None—	Exact	AND
—None—	Exact	AND

Add Filter Logic... Save Cancel

Matching Rule
matching Customer details

Help for this Page ?

Matching Rule Detail

Object: Customer Details
Rule Name: matching Customer details
Unique Name: matching_Customer_details
Description:
Matching Criteria: {Customer Details: Email EXACT MatchBlank = FALSE} AND {Customer Details: Phone_Number EXACT MatchBlank = FALSE}
Status: Inactive
Created By: project_2, 25/09/2023, 10:15 am
Modified By: project_2, 10/10/2023, 3:32 pm

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

Q: Duplicat

SETUP

Duplicate Management

- Duplicate Error Logs
- Duplicate Rules**
- Matching Rules

Didn't find what you're looking for? Try using Global Search.

All Duplicate Rules

What Are Duplicate Rules?

View: All Duplicate Rules

Role Name	Description	Matching Rule	Active	Last Modified By	Last Modified Date
CustomerDetailDuplicate	Identify accounts that duplicate other accounts	Matching Customer details	<input type="checkbox"/>	r2	10/10/2023
Standard Account Duplicate Rule	Identify contacts that duplicate other contacts and leads	Instant Account Matching Rule	<input checked="" type="checkbox"/>	r2	24/09/2023
Standard Contact Duplicate Rule	Identify leads that duplicate other leads and contacts	Instant Lead Matching Rule	<input checked="" type="checkbox"/>	r2	24/09/2023
Standard Lead Duplicate Rule	Environment	Instant Contact Matching Rule	<input checked="" type="checkbox"/>	r2	24/09/2023
	Individual				
	Laptop				
	Lead				

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.

Edit Duplicate Rule
Customer Detail duplicate

Duplicate Rule Edit

Rule Details

- Rule Name: Customer Detail duplicate
- Description:
- Object: Customer Details
- Record-Level Security:
 - Enforce sharing rules
 - Bypass sharing rules

Actions

Specify what happens when a user tries to save a duplicate record.

- Action On Create: Allow Alert Report
- Action On Edit: Allow Alert Report
- Alert Text: Use one of these records?

Matching Rules

Define how duplicate records are identified.

- Compare Customer Details With: Customer Details
- Matching Rule: matching Customer details
- 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
--None--	--None--	<input type="text"/>

Add Rule Remove Rule

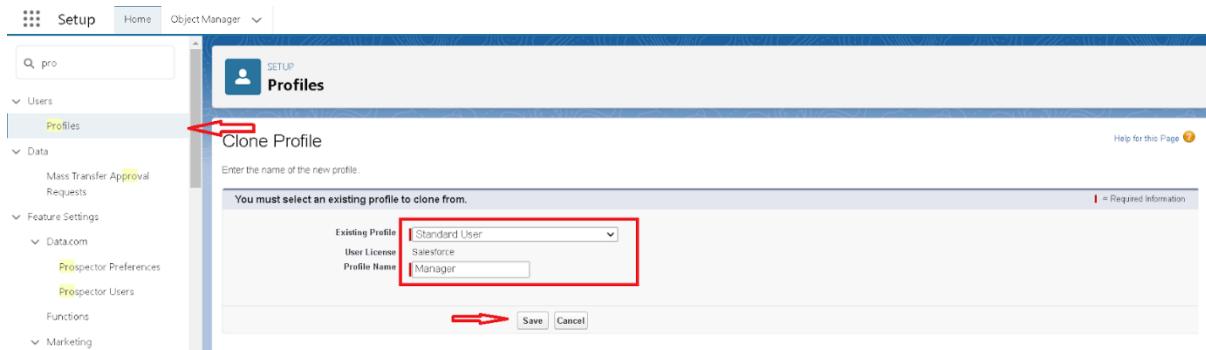
Save **Save & New** **Cancel**

Milestone 7: SECURITY (PROFILES, ROLES & SHARING)

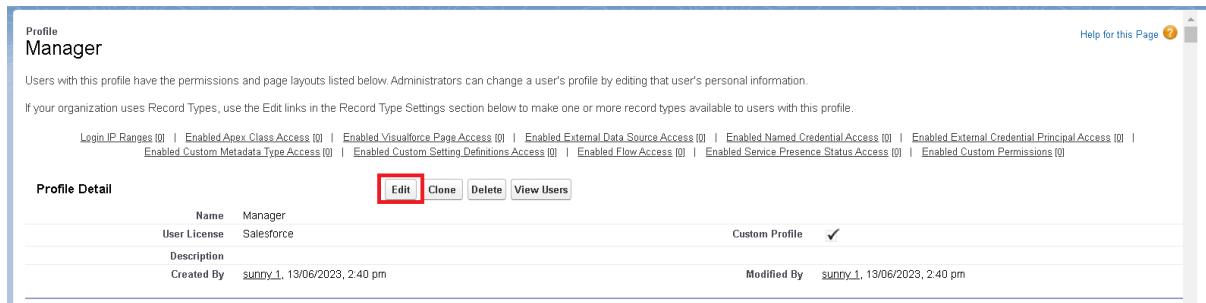
Activity 1: Create Custom Profiles

To create a new profile:

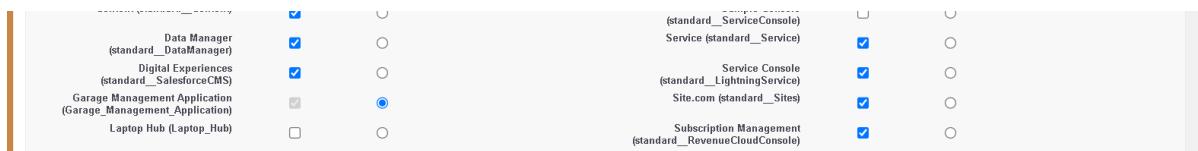
1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.



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.

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

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 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.

Activity 2: Create Roles & Public Groups

Creating Manager Role:

Creating Manager Role:

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

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



- Give Label as “Manager” and Role name gets auto populated. Then click on Save.

Role Edit

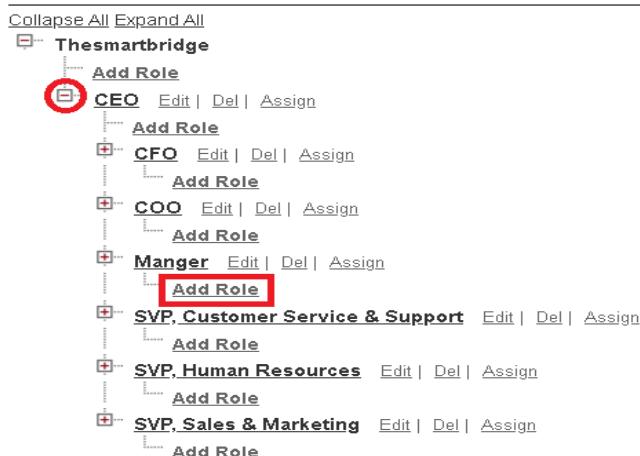
Label	Manger	←
Role Name	Manger	↻
This role reports to	CEO	🔍
Role Name as displayed on reports		

→ **Save** **Save & New** **Cancel**

Creating another roles

Creating another two roles under manager

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

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	hmika
Email	
Username	Mikaelson@Niklaus
Nickname	nik
Title	
Company	
Department	
Division	

Required Information

Role	Manager
User License	Salesforce
Profile	Manager
Active	<input checked="" type="checkbox"/>
Marketing User	<input type="checkbox"/>
Offline User	<input type="checkbox"/>
Knowledge User	<input type="checkbox"/>
Flow User	<input type="checkbox"/>
Service Cloud User	<input type="checkbox"/>
Site.com Contributor User	<input type="checkbox"/>
Site.com Publisher User	<input type="checkbox"/>
WDC User	<input type="checkbox"/>
Data.com User Type	--None--

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

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.

Activity 3: Configure Sharing Settings

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.

The screenshot shows the 'Sharing Settings' page in Salesforce Setup. The 'Service records' row in the main table has its 'OWD' dropdown set to 'Private', which is highlighted with a red box. Other objects like Work Plan Template, Work Step Template, etc., have their OWD set to 'Private'. Below the table are sections for 'User Visibility Settings' and 'Other Settings'. At the bottom are 'Save' and 'Cancel' buttons, with 'Save' also highlighted with a red box.

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

The screenshot shows the 'Service records Sharing Rules' page. At the top, there's a header with 'Service records Sharing Rules', a 'New' button (which is the target of a red arrow), and a 'Recalculate' button. Below the header, it says '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' page in Salesforce setup. It consists of five steps:

- Step 1: Rule Name**: Fields include Label ('sharing settings'), Rule Name ('sharing_settings'), and Description. A red arrow points to the Rule Name field.
- Step 2: Select your rule type**: Rule Type is set to 'Based on record owner'.
- Step 3: Select which records to be shared**: Service records are owned by members of 'Roles' (Sales person). A red arrow points to the 'Sales person' dropdown.
- Step 4: Select the users to share with**: Share with 'Roles' (Manager). A red arrow points to the 'Manager' dropdown.
- Step 5: Select the level of access for the users**: Access Level is 'ReadWrite'. A red arrow points to the 'ReadWrite' dropdown.

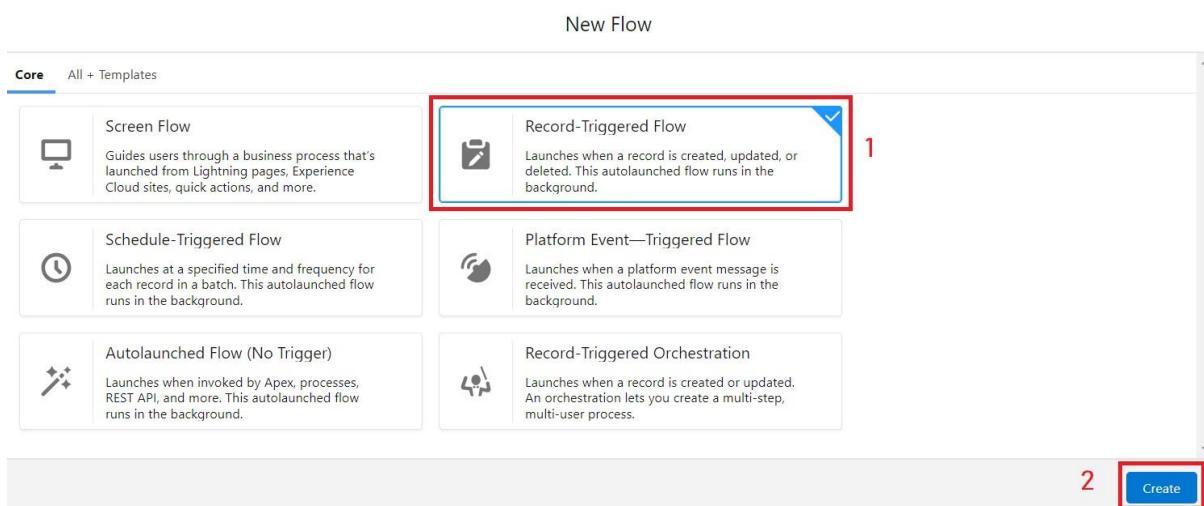
At the bottom are 'Save' and 'Cancel' buttons, with a red arrow pointing to the 'Save' button.

Milestone 8: AUTOMATION (FLOWS & APEX)

Activity 1: Create Flows

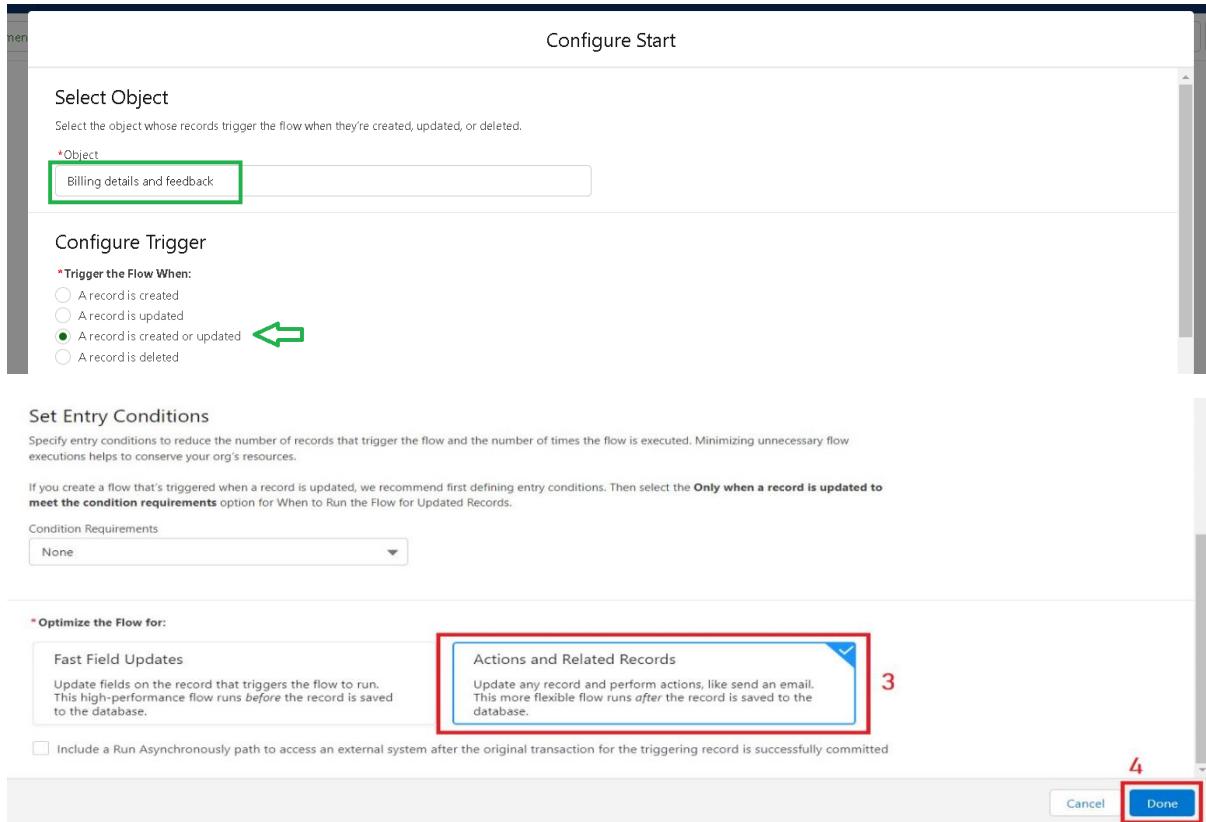
Create a Flow:

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.

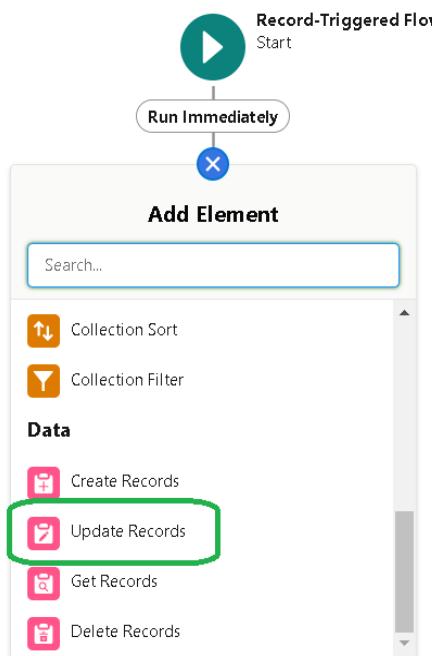


3. Select the Object as “Billing details and feedback” in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.

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

Edit Update Records

Update Salesforce records using values from the flow.

*Label * API Name

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

Cancel

Set Filter Conditions

Condition Requirements to Update Record

Field Operator Value

+ Add Condition

Set Field Values for the Billing details and feedback Record

Field Value

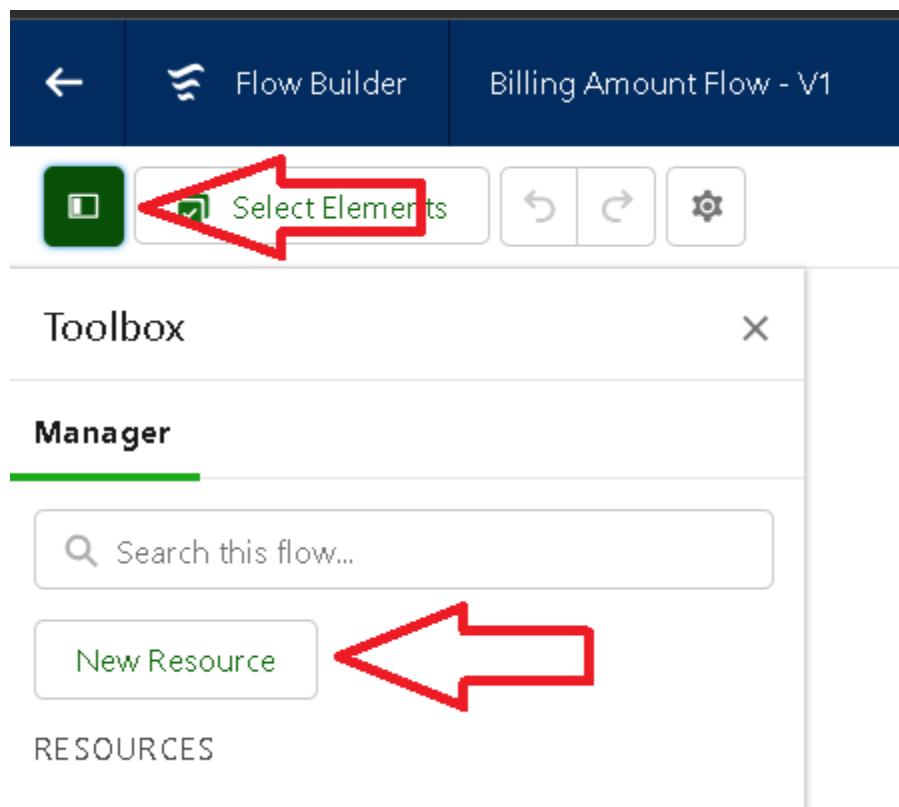
+ Add Field

Cancel

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

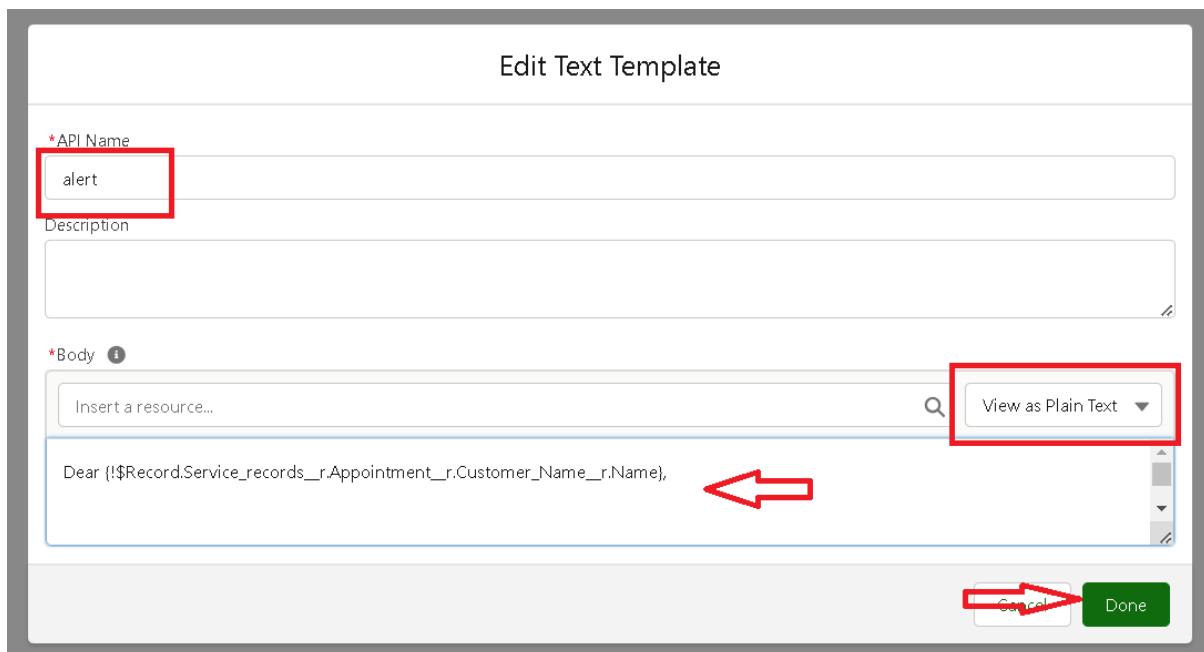
{!\$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

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.

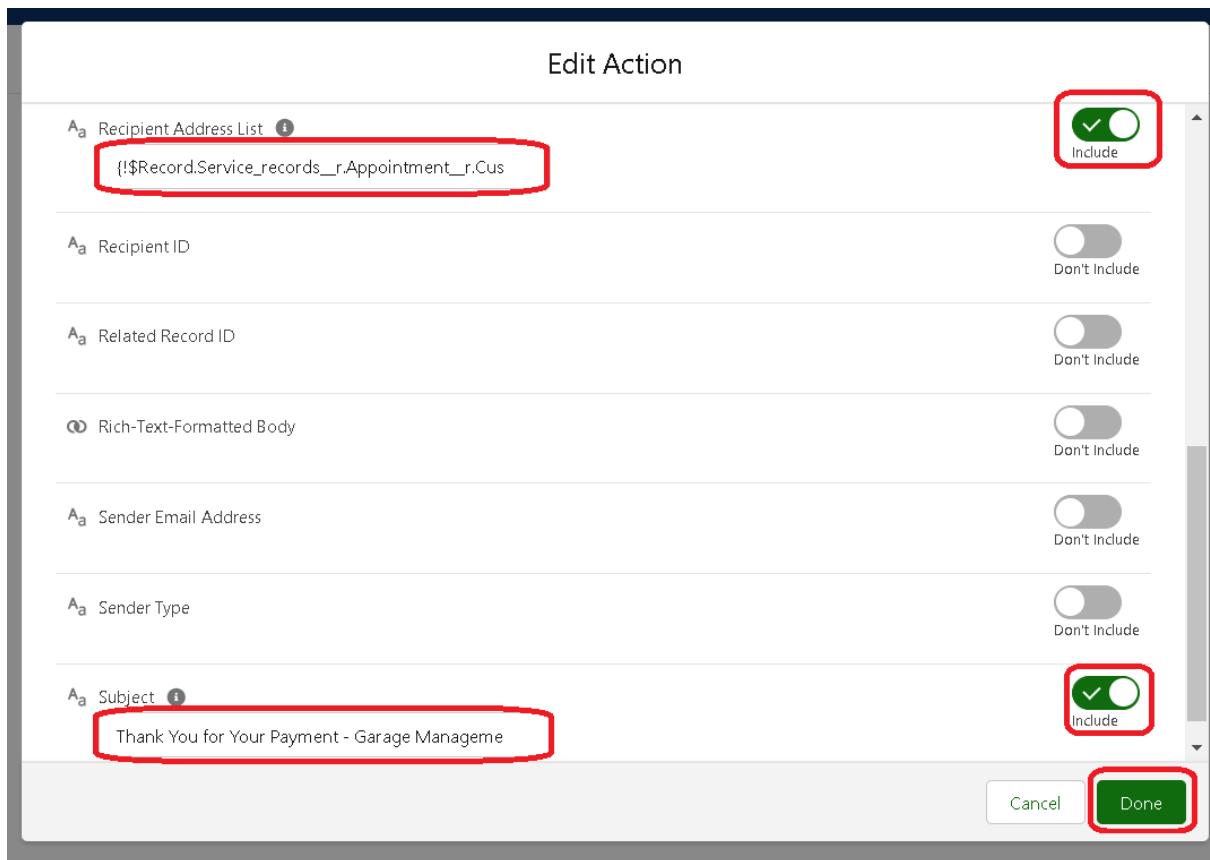
Edit Action

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variables.

*Label	*API Name
Email Alert	Email_Alert
Description	
<input type="text"/>	

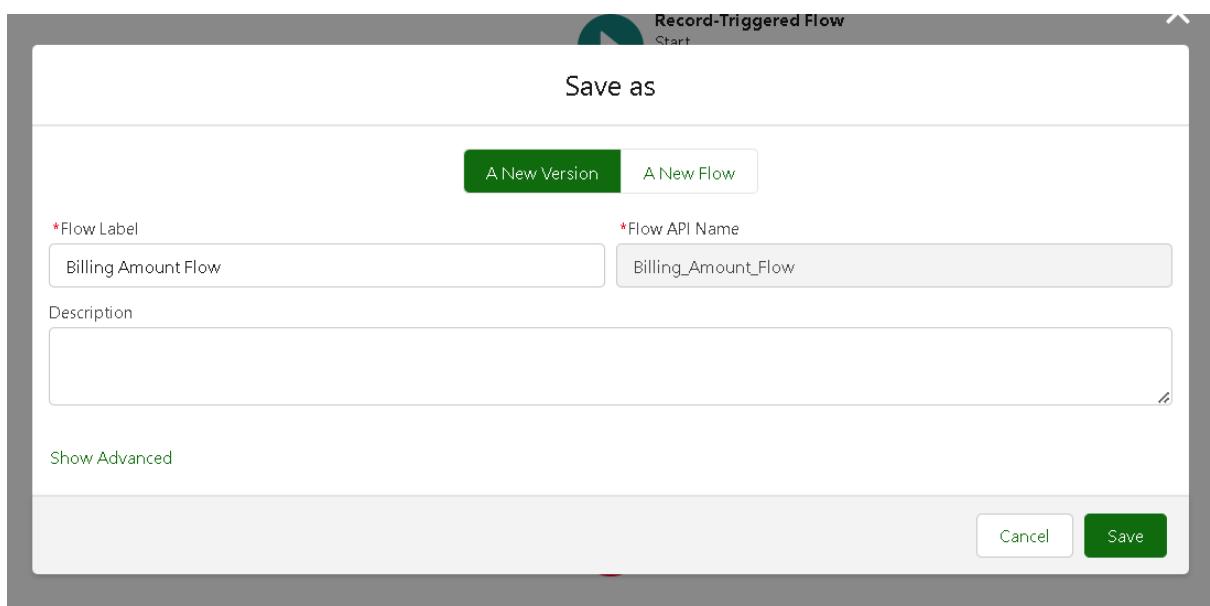
Set Input Values for the Selected Action

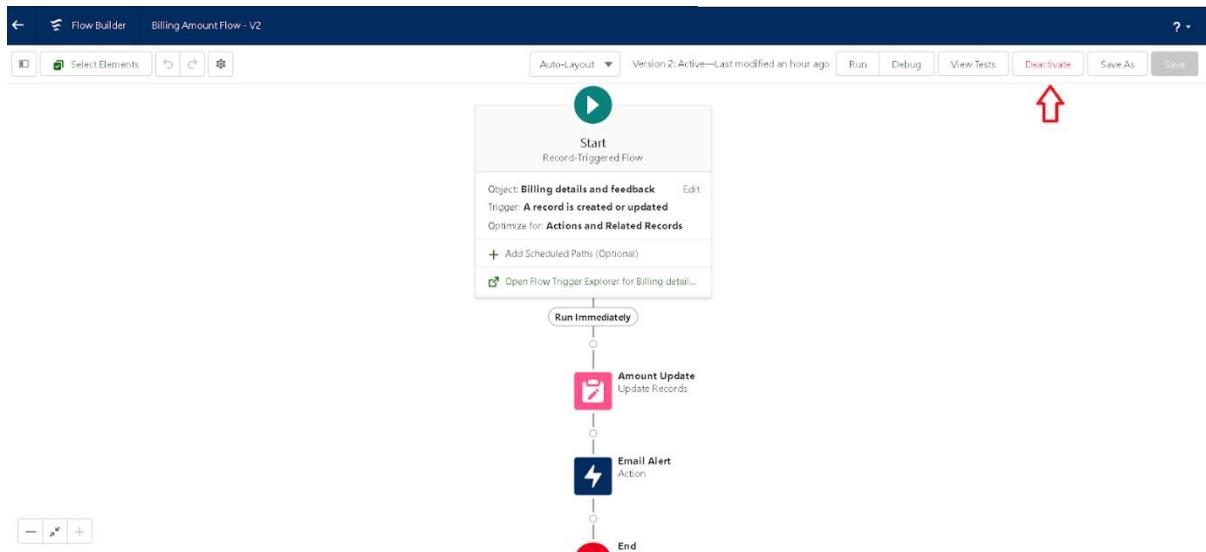
A_a Body	<input type="text" value="={!alert}"/> Include
A_a Email Template ID	<input type="checkbox"/> Don't Include
A_a Log Email on Send	<input type="checkbox"/> Don't Include



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

35. And click save, and click on activate.





Create another Flow:

1. Go to setup type Flow in quick find box Click on the Flow and Select the New Flow.

The screenshot shows the Salesforce Setup interface with the following navigation path:

- Setup
- Home
- Object Manager
- Flows

The main area displays a list of Flow Definitions:

Flow Label	Process Type	Ac...	Te...	Package State	Pa...	Last Modified By	Last Modified ...
Ac Amount update	Autolaunched Flow	<input type="checkbox"/>	<input type="checkbox"/>	Unmanaged	Veera Venkata Varaprasad Androthu	07/06/2023, 11:35 am	
Book Appointment from Invitation	Salesforce Scheduler Flow	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Cancel Item Flow	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Change Case Owner to Incident Owner	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			
Close Change Request & Related Issues	Screen Flow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Managed-Installed			

2. Select the Record-triggered flow and Click on Create.

The screenshot shows the "New Flow" creation page with the following options:

- Core All + Templates
- Screen Flow
- Record-Triggered Flow (highlighted with a red box)
- Schedule-Triggered Flow
- Platform Event—Triggered Flow
- Autolaunched Flow (No Trigger)
- Record-Triggered Orchestration

New Flow

Core All + Templates

Record-Triggered Flow (highlighted with a red box)
Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.

Create (highlighted with a red box)

3. Select the Object as “ **Service records**”in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimise 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. Set a filter condition : All Conditions are met(AND)
8. Field : **Quality_Check_Status__c**
9. Operator : **Equals**
10. Value : **True**
11. And Set Field Values for the Billing details and feedback Record
12. Field : **Service_Status__c**
13. Value : **Completed**

Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND) ▾

Field	Operator	Value
Quality_Check_Status__c	Equals	True X Delete

[+ Add Condition](#)

Set Field Values for the Service record Record

Field	Value
Service_Status__c	← Completed Delete

[+ Add Field](#)

14. Click On **Done**.
15. Click on **save**

16. Given the Flow label as **Update Service Status** , Flow Api name will be auto populated.

17. And click save, and click on **activate**.

Activity 2: Create Apex Trigger

Apex handler:

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”.

```
AmountDistribution.apxt [AmountDistributionHandler.apxc *]
Code Coverage: None API Version: 58 Go To
1 * public class AmountDistributionHandler {
2
3 *     public static void amountDist(List<Appointment__c> listApp){
4         List<Service_records__c> serlist = new List<Service_records__c>();
5
6         for(Appointment__c app : listApp){
7             if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
8                 app.Service_Amount__c = 10000;
9             }
10            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
11                app.Service_Amount__c = 5000;
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 }
```

```
AmountDistribution.apxt [AmountDistributionHandler.apxc *]
Code Coverage: None API Version: 58 Go To
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 }
```

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;

}

}

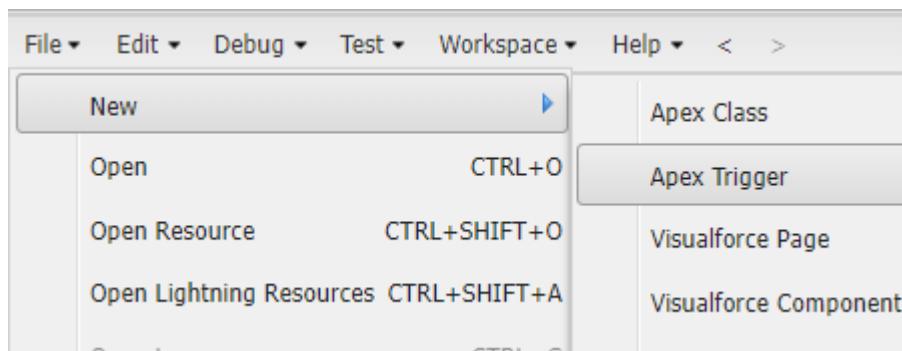
}

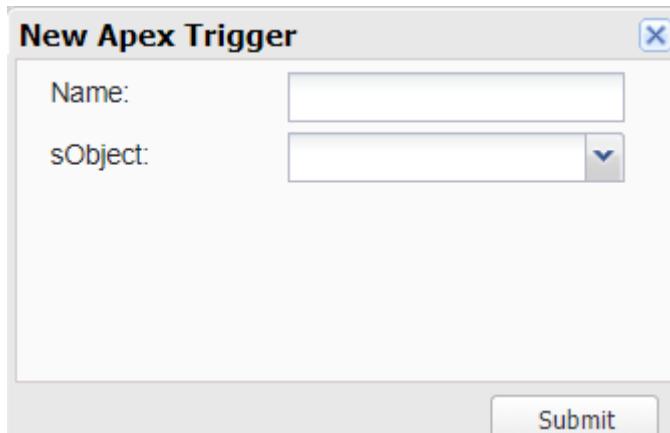
```

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)

```
{
}
```

Handler for the Appointment Object

```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
AmountDistribution.apxt AmountDistributionHandler.apxc *
Code Coverage: None API Version: 58
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);

    }

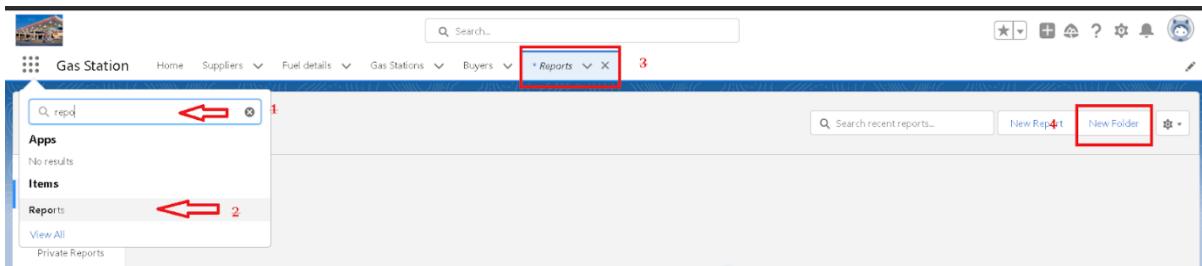
}
```

Milestone 9: ANALYTICS (REPORTS & DASHBOARDS)

Activity 1: Create Reports

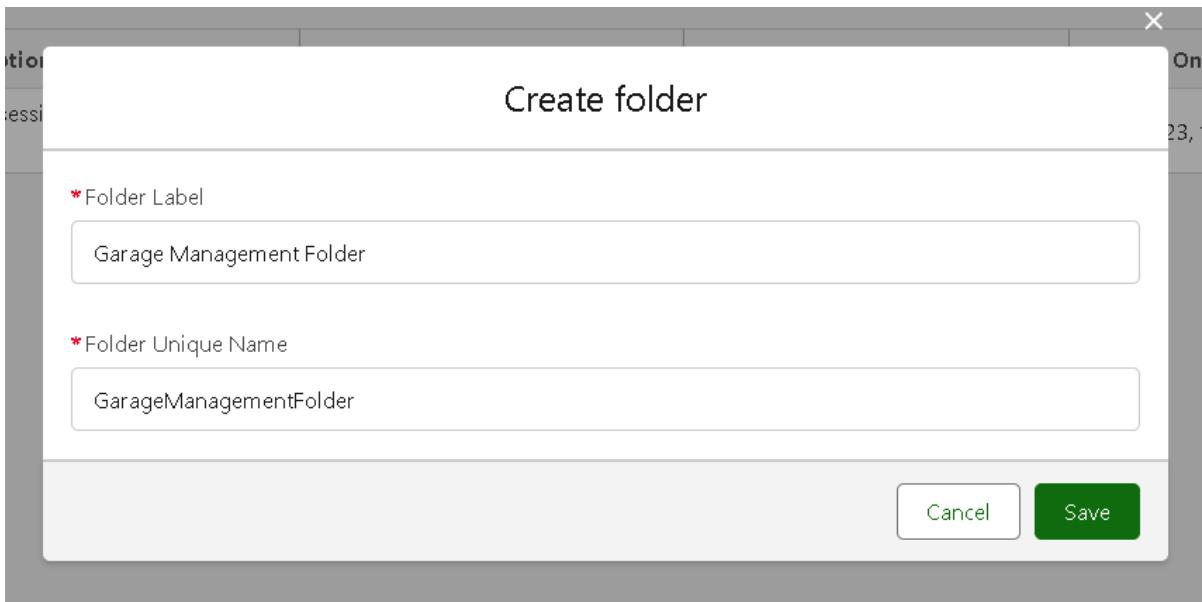
create a report folder:

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.

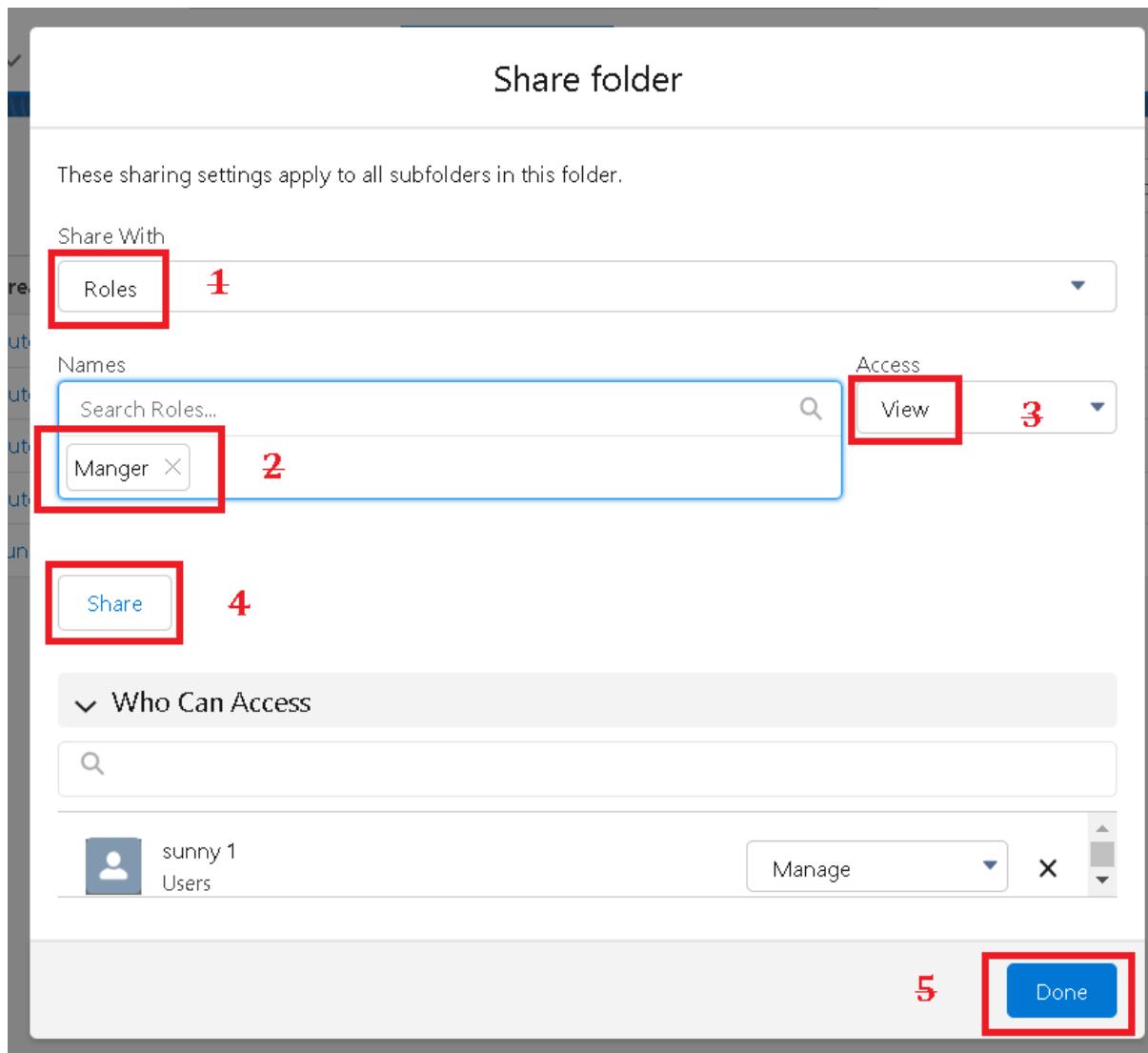
4. Click save.



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.

The screenshot shows the Salesforce Setup interface with the 'Report Types' page selected. The sidebar on the left has a 'Reports & Dashboards' section with a 'Report Types' link, which is highlighted with a green arrow. The main content area displays a list of existing report types like 'Bot Metrics Daily Summer '23' and 'Session Metrics Summer '23'. At the top of this list, there is a 'New Custom Report Type' button, also highlighted with a green arrow.

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' and 'Deployment'. In the 'Identification' section, fields are highlighted with green arrows: 'Report Type Label' (set to 'Service information'), 'Report Type Name' (set to 'Service_information'), 'Description' (set to 'Service information'), and 'Store in Category' (set to 'Other Reports'). In the 'Deployment' section, the 'Deployment Status' field is highlighted with a green arrow and set to 'Deployed'. A green arrow also points to the 'Next' button at the bottom right.

9. now , Click on Related object box.
10. Click on Select Object, choose Appointment Object as shown in fig.

New Custom Report Type
Service information

Step 2. Define Report Records Set Step 2 of 2

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 Select Object:

- Appointments (highlighted with a green arrow)
- Duplicate Record Items

Diagram: A Venn diagram showing overlapping circles A and B. Below the circles is a grid divided into two columns labeled A and B, each containing four horizontal bars.

Previous Save Cancel

Step 2. Define Report Records Set

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

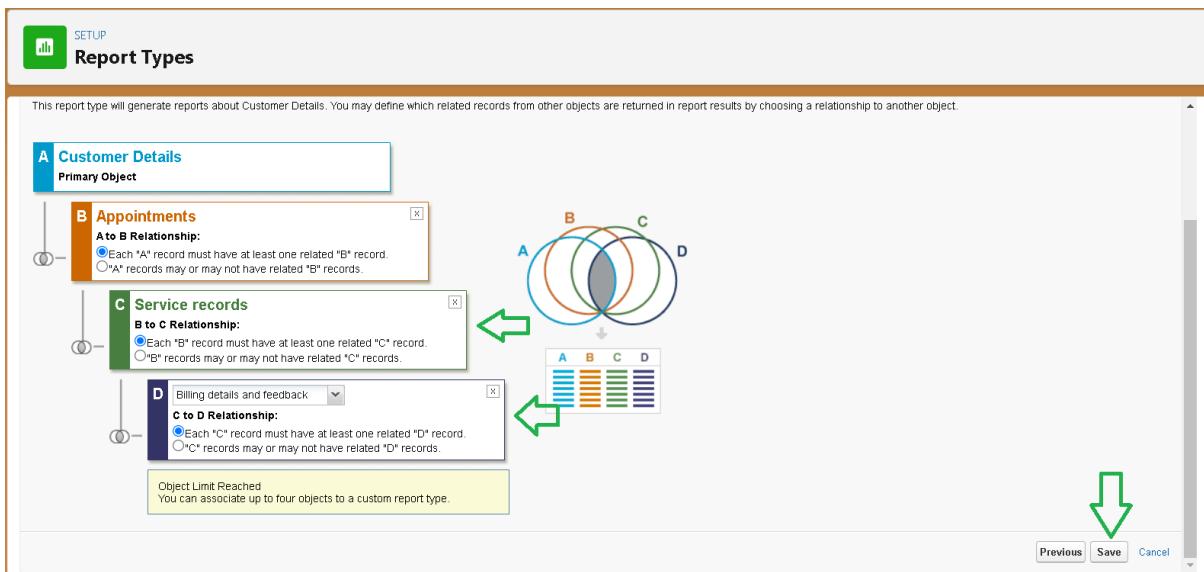
A to B Relationship:

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

(Click to relate another object)

Diagram: A Venn diagram showing overlapping circles A and B. Below the circles is a grid divided into two columns labeled A and B, each containing four horizontal bars.

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:

- 1) Go to the app >> click on the reports tab
- 2) Click New Report.

REPORTS	Report Name	Description	Folder	Created By	Created On	Subscribed
Recent	Employee's working on projects report		Private Reports	Employee Project	5/6/2023, 9:33 am	
Created by Me	Assets assigned to Employees		Private Reports	Employee Project	5/6/2023, 9:36 am	
Private Reports						
Public Reports						
All Reports						

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

Report Type Name	Category
Service records	Standard
Service records with Appointment	Standard
Service records History	Standard
Billing details and feedback with Service records	Standard
Service information	Custom

Details

Service information
Custom Report Type

Start Report

Description
Service information

Created By You
No Reports Yet

Created By Others
No Reports Yet

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.

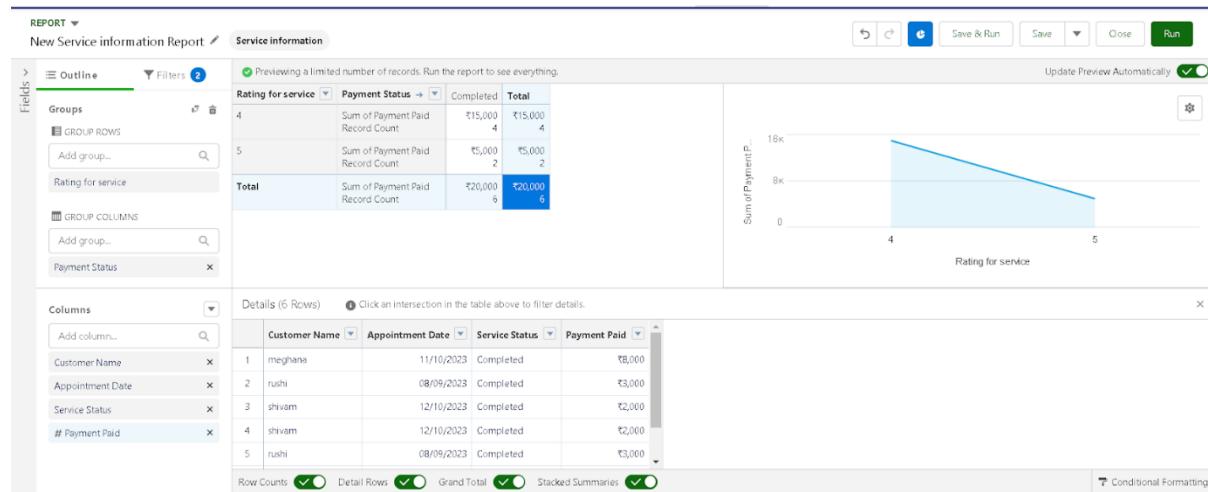
- a) Payment Status

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.



Save Report

*Report Name	<input type="text" value="New Service information Report"/>	
Report Unique Name	<input type="text" value="New_Service_information_Report_oVu"/>	
Report Description	<input type="text"/>	
Folder	<input type="text" value="Garage Management Folder"/>	
		<input type="button" value="Select Folder"/>
		<input type="button" value="Cancel"/> <input type="button" value="Save"/>

Activity 2: Create a Dashboard

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.

Create folder

*Folder Label	<input type="text" value="Service Rating"/>
*Folder Unique Name	<input type="text" value="ServiceRating"/>
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

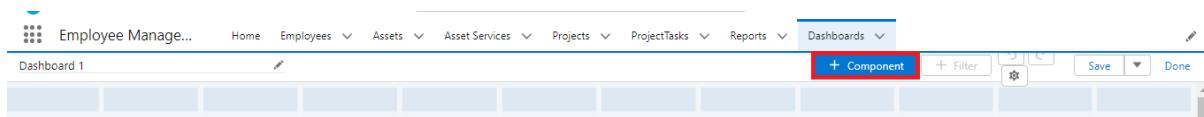
6. Follow the same steps, from Reports Milestone and Activity 2, and provide the sharing settings for the folder that was just created.

Create Dashboard:

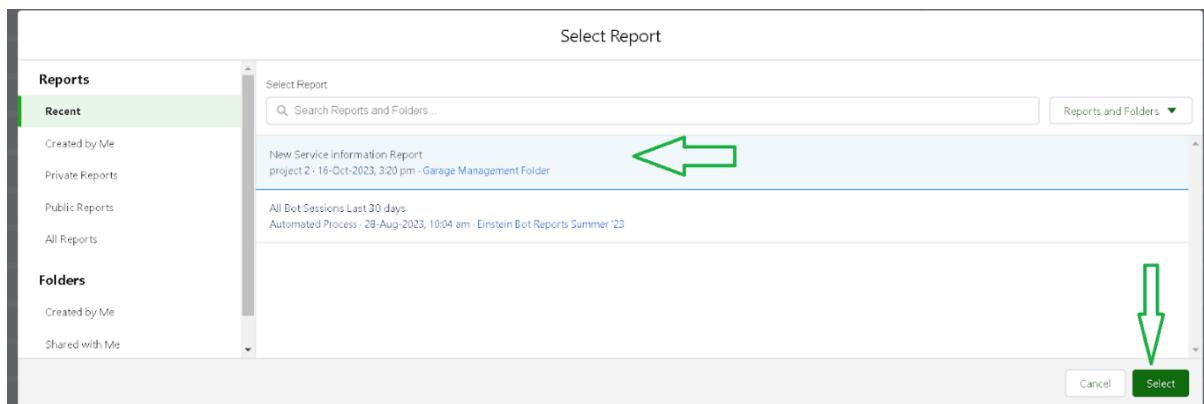
1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.

The screenshot shows a 'New Dashboard' dialog box. At the top is the title 'New Dashboard'. Below it is a field labeled 'Name' with the value 'Customer review'. This field is highlighted with a green rectangle. Below the name is a 'Description' field, which is empty. Under 'Folder', there is a dropdown menu showing 'Service Rating' and a 'Select Folder' button. At the bottom right are 'Cancel' and 'Create' buttons, with 'Create' also highlighted with a green rectangle.

3. Select add component.

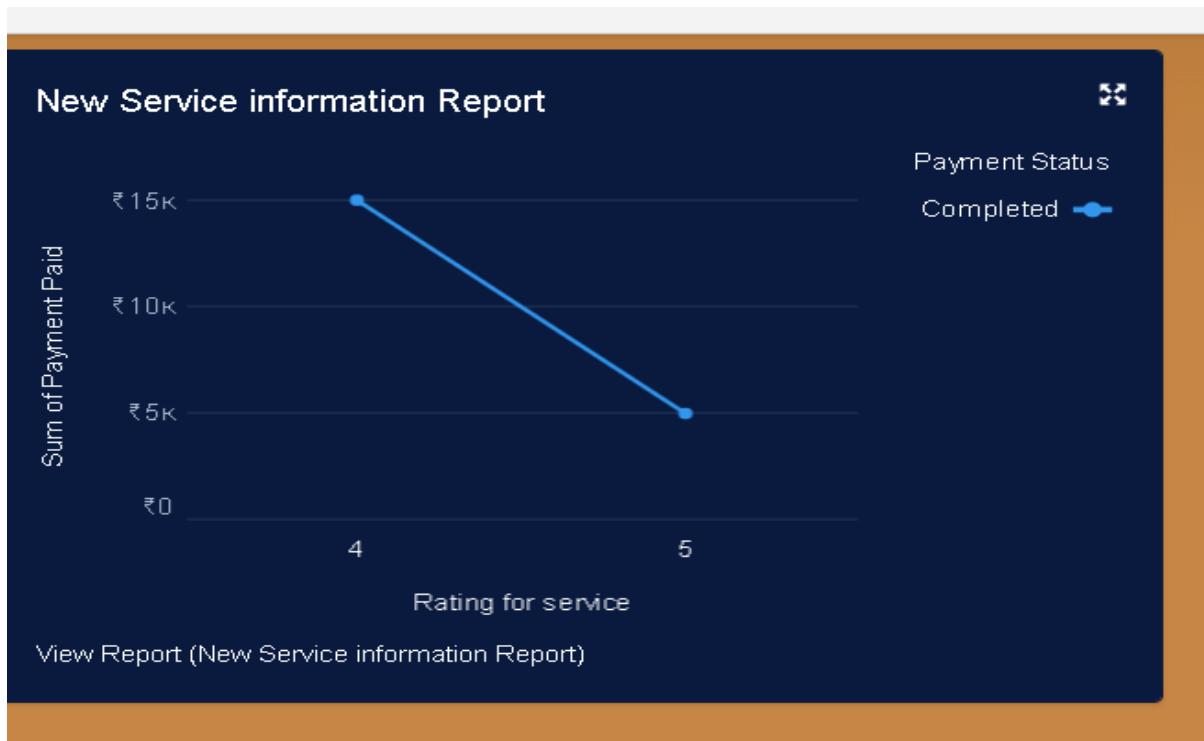


4. Select a Report and click on select.



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.

The form is titled 'Edit Subscription'. It includes fields for 'Frequency' (set to 'Weekly'), 'Days' (set to 'Mon'), 'Time' (set to '3:00 pm'), and 'Recipients' (checkbox checked for email notifications). Green arrows highlight the 'Frequency' dropdown, the 'Days' selector, and the 'Save' button at the bottom right.

Phase 5: Deployment, Documentation & Maintenance

creating records:

To create a record in the follow objects follow these steps

1. Click on the app launcher located at the left side of the screen.
2. Search for “ **Garage Management**” and click on it.
3. Click on the “ **Consumer details tab**”.
4. Click on new and fill the details as shown below figs, and click save.

New Customer Detail

* = Required Information

Information

*Customer Name: Mac

Phone number: 5678765567

Gmail: mac@gmail.com

Owner: Annapurna SmartBridge

Cancel Save & New Save

Now, Create the Appointment Record

1. Click on the “ **Appointment tab**”.
2. Enter the customer details as created, while entering Appointment Date enter the date less than the created date.
3. Match the validation while entering the vehicle number plate.
4. Select the services you need.
5. Click on save to see the Service Amount.

Garage Management... Customer Details Appointments Service records Billing details and feedback Reports Dashboards

Appointment app-016

Appointment Name: app-016

Customer Details: Mac

* Appointment Date: 13/11/2024

Maintenance service:

Repairs:

Replacement Parts:

Service Amount:

* Vehicle number plate: TS30EU0443

Created By: Annapurna SmartBridge, 18/11/2024, 3:28 pm Updated By: Annapurna SmartBridge, 18/11/2024, 3:28 pm

Cancel **Save**

Now, Create a service Record

1. Click on the “Service record tab”.
2. Enter the Appointment, and started is selected as default.
3. Click on save.

New Service record

* = Required Information

Information

Service Record Name: app-016

Owner: Annapurna SmartBridge

* Appointment: app-016

Quality Check Status:

Service Status: Started

Cancel **Save & New** **Save**

4. Open the record and click on Quality check status as true.

5. Click on save.

The screenshot shows a service record creation form. The fields are as follows:

- Service Record Name: ser-010
- Owner: Annapurna SmartBridge
- * Appointment: app-016
- Quality Check Status:
- Service Status: Started
- service date: 18/11/2024
- A note below the date says: *This field is calculated upon save*
- Created By: Annapurna SmartBridge, 18/11/2024, 4:32 pm
- Buttons: Cancel, Save

6. Now automatically Service status will be moved to completed.

The screenshot shows the service record details page. The status has been updated to "Completed". Other details remain the same as in the previous screenshot.

Related **Details**

Service Record Name: ser-010
Owner: Annapurna SmartBridge

Appointment: app-016

Quality Check Status:

Service Status: Completed

service date: 18/11/2024

Created By: Annapurna SmartBridge, 18/11/2024, 4:32 pm

Last Modified By: Annapurna SmartBridge, 18/11/2024, 4:34 pm

Conclusion

The "Garage Management System" project successfully demonstrates the power of the Salesforce platform in creating a specialized, efficient, and user-friendly application for a real-world business need. By automating key tasks, ensuring data integrity, and providing clear analytics, the system empowers automotive repair facilities to improve their operations and enhance customer satisfaction.

- **Project Achievements:**

- Successfully built a streamlined process for managing customers, appointments, services, and billing.
- Automated complex calculations and status updates using a combination of Flows and Apex triggers.
- Implemented a robust security model to ensure users can only access appropriate data.
- Provided clear, actionable insights through customized reports and dashboards.

- **Future Scope:**

- **Mobile Integration:** Develop mobile-friendly components to allow technicians to update service records directly from the workshop floor.
- **Inventory Management:** Add custom objects to track spare parts inventory, automating reorder processes when stock is low.
- **Customer Portal:** Create a Salesforce Experience Cloud site to allow customers to book their own appointments, view their service history, and pay bills online.
- **AI Integration:** Use Salesforce Einstein to predict service needs based on vehicle history and mileage, allowing for proactive customer outreach.