



SB8067 Salesforce Developer



S.Veerasamy Chettiar College of Engineering And
Technology,Puliyangudi.

Tittle : Garage Management System

Submitted By,

E. Suvetha,

B.E - CSE, Final Year,

952622104050.

Garage Management System

College code: 9526

**College Name: S.Veerasamy Chettiar College of Engineering And
Technology,Puliyangudi.**

Total Number of Student in a group: 4

Department: B.E – Computer Science and Engineering

Year: Final Year(2022-2026)

Team ID: NM2025TMID02613

Team Leader : Suvetha. E

Team member : Padma. E

Team member : Divya .T

Team member : Gowsalya .K

GARAGE MANAGEMENT SYSTEM

1. Ideation Phase

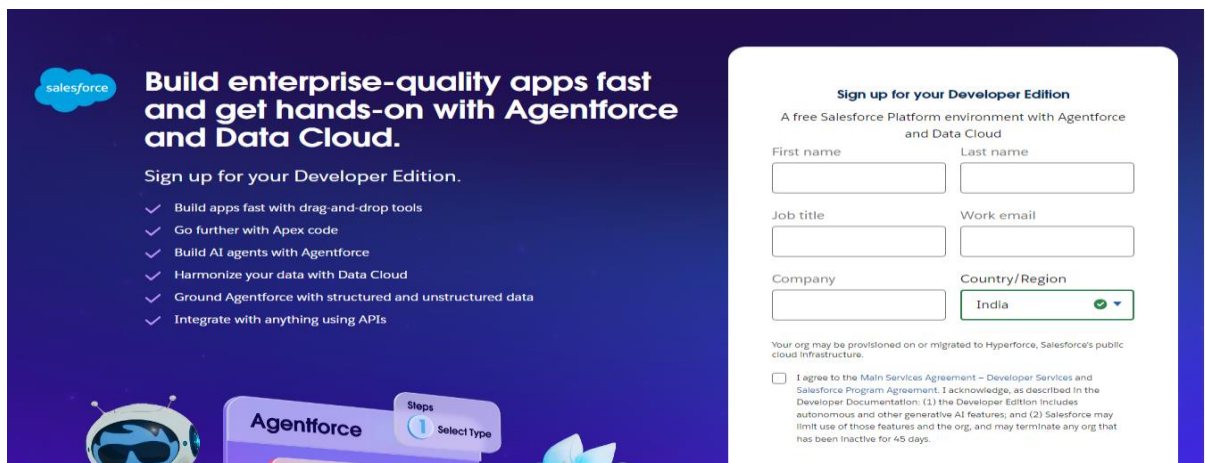
The Garage Management System is designed to simplify and automate the daily operations of a garage using Salesforce CRM.

Traditional garages depend on manual record-keeping for customer details, appointments, and billing, which leads to inefficiency and errors. The idea behind this project is to use Salesforce's cloud-based capabilities to:

- Manage customer and vehicle information digitally.
- Schedule and track service appointments.
- Automate billing and feedback collection.
- Maintain transparency between garage owners, staff, and customers.

This phase focuses on identifying challenges such as delayed service tracking, missing customer data, and billing confusion — and proposing Salesforce automation as the solution.

1. Salesforce Developer Signup Page

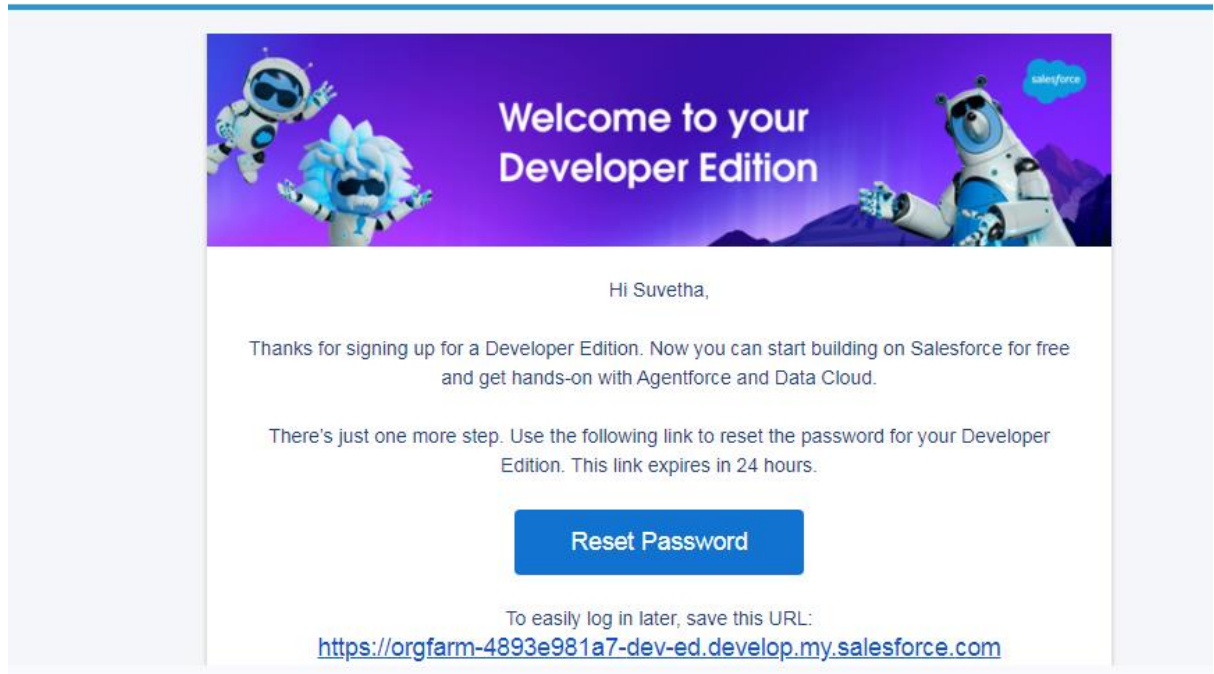


The screenshot shows the Salesforce Developer Edition Signup Page. On the left, there is a dark blue banner with the Salesforce logo and the text "Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud." Below this, it says "Sign up for your Developer Edition." and lists several benefits: "Build apps fast with drag-and-drop tools", "Go further with Apex code", "Build AI agents with Agentforce", "Harmonize your data with Data Cloud", "Ground Agentforce with structured and unstructured data", and "Integrate with anything using APIs". At the bottom of the banner, there is an illustration of a robot head and a box labeled "Agentforce" with a "Steps" indicator showing "1 Select Type".

On the right, there is a white form titled "Sign up for your Developer Edition". Below the title, it says "A free Salesforce Platform environment with Agentforce and Data Cloud". The form has several input fields: "First name", "Last name", "Job title", "Work email", "Company", and "Country/Region". The "Country/Region" field is currently set to "India" with a green checkmark. Below the form, there is a checkbox labeled "I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days."

Figure 1.1: Salesforce Developer Org Signup Page

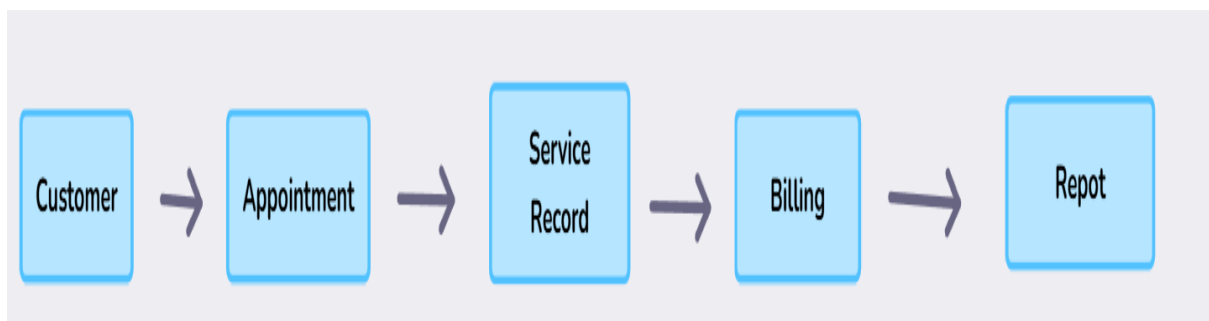
2.Salesforce Setup Home Page



Salesforce Setup Page (Post Activation)

3.Garage Management System Concept Diagram

- **Simple flow:**



2. Project Planning Phase

The planning phase involves defining the project modules, workflow, and access hierarchy within Salesforce. This stage ensures a clear vision of system development and the logical structure of objects and relationships.

Key modules planned:

- 1. Customer Details – Stores customer contact info and email.**
- 2. Appointment – Tracks service bookings, dates, and service types.**
- 3. Service Records – Maintains details of services performed.**
- 4. Billing Details & Feedback – Stores payment info and service ratings.**

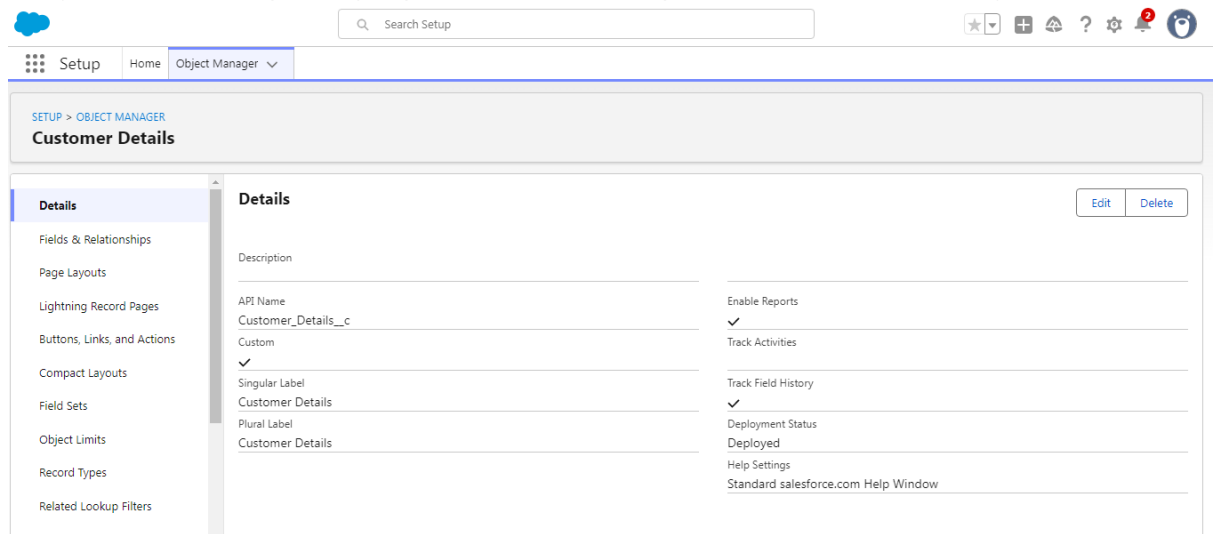
Roles and responsibilities:

- Manager: Access to all modules, can create reports and dashboards.**
- Salesperson: Limited access to assigned records.**

A structured timeline was followed:

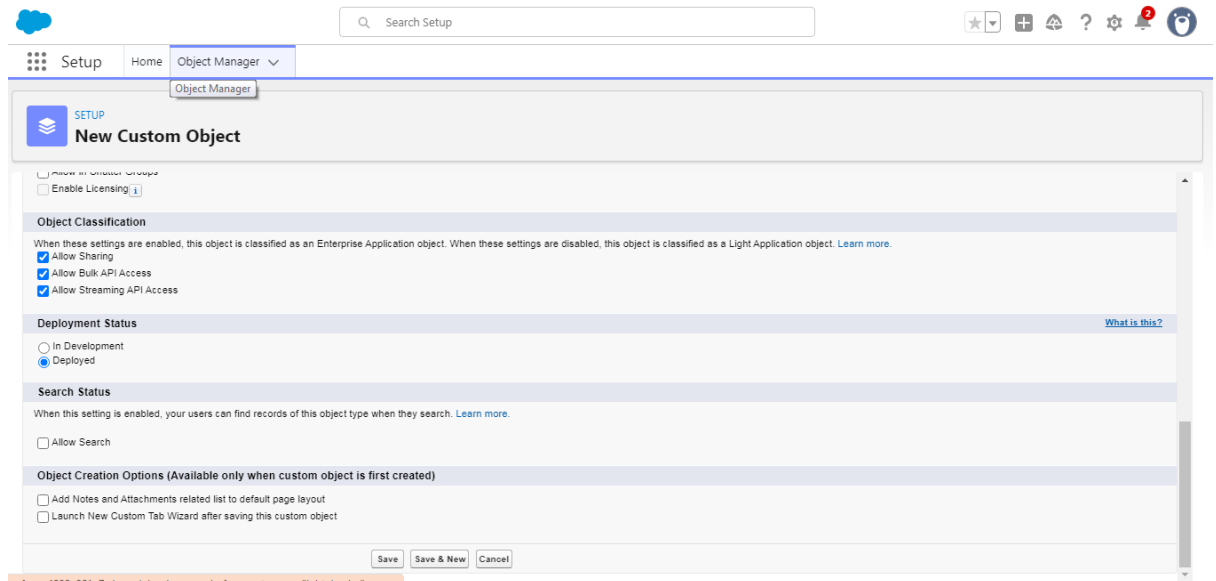
- 1. Object creation and setup**
- 2. Tab and Lightning app configuration**
- 3. Profile & Role Hierarchy setup**
- 4. Sharing settings and automation flows**
- 5. Testing and deployment**

1. Object Manager page – showing all custom objects.



Object Manager – Creating Custom Objects

2. List of all custom objects created.



List of Garage Management Custom Objects

3.Role hierarchy showing Manager → Salesperson.

Setup

Home

Object Manager

Home

SETUP > OBJECT MANAGER

Opportunity Contact Role

Details

Fields & Relationships

Page Layouts

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Triggers

Flow Triggers

Validation Rules

Fields & Relationships

6 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contact	ContactId	Lookup(Contact)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Opportunity	OpportunityId	Lookup(Opportunity)		✓
Primary	IsPrimary	Checkbox		
Role	Role	Picklist		

Role Hierarchy for Manager and Salesperson

4.Sharing settings configuration.

Setup

Home

Object Manager

Home

SETUP > OBJECT MANAGER

Service records

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Fields & Relationships

9 Items, Sorted by Field Label

Q Quick Find

New

Deleted Fields

Field Dependencies

Set History Tracking

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment	Appointment__c	Lookup(Appointment)		✓
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Payment Status	Payment_Status__c	Picklist		
Quality Check Status	Quality_Check_Status__c	Checkbox		
service date	service_date__c	Formula (Date)		
Service records Name	Name	Auto Number		✓

Sharing Settings for Service Records Object

3. Project Design Phase

The design phase focuses on creating the data structure and user interface.

Each object and its relationships were carefully defined to maintain consistency and data flow across modules.

Custom Objects Designed:

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

Field Types Used:

- Text Fields – Customer Name, Vehicle Number Plate
- Email, Phone, Date, Currency Fields
- Checkbox – Maintenance, Repairs, Replacement Parts
- Lookup Fields – To link related records (Customer ↔ Appointment ↔ Service ↔ Billing)
- Picklist – For service status and payment status
- Formula – Auto-calculated fields (e.g., Service Date = Created Date)

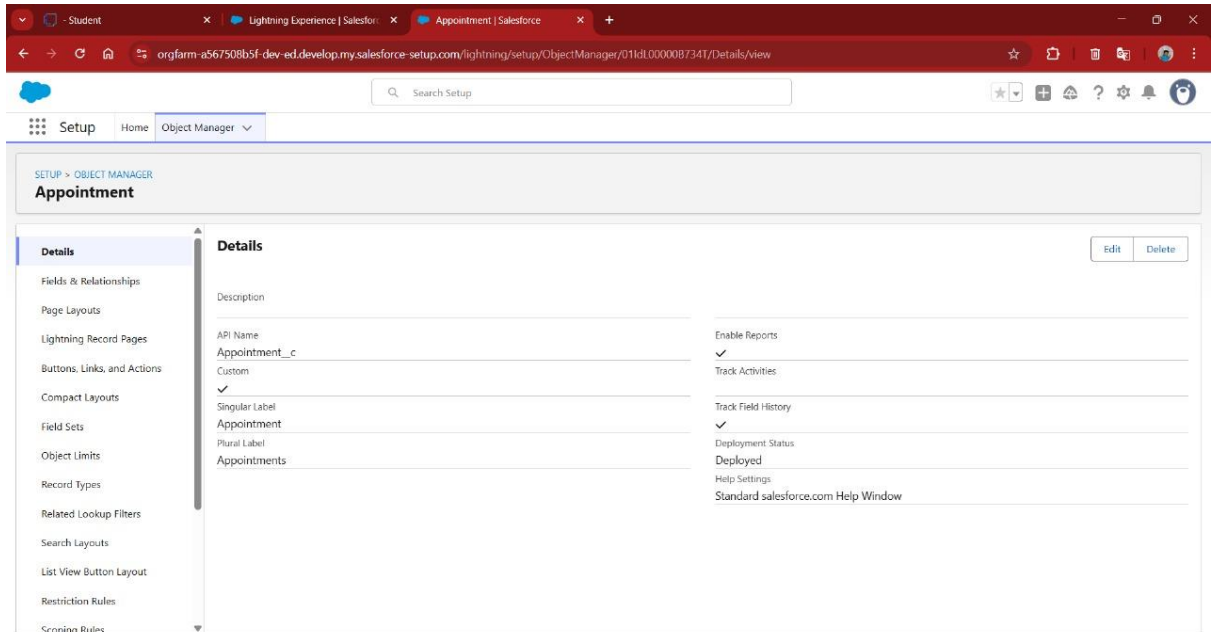
Lightning App Design:
All objects are grouped into one Garage Management Application.
Tabs for each module are added for easy navigation.

Data Integrity:

- Validation rules ensure correct data (e.g., valid vehicle number format).

- Duplicate rules prevent repeated customer records.
- Profiles manage user permissions.

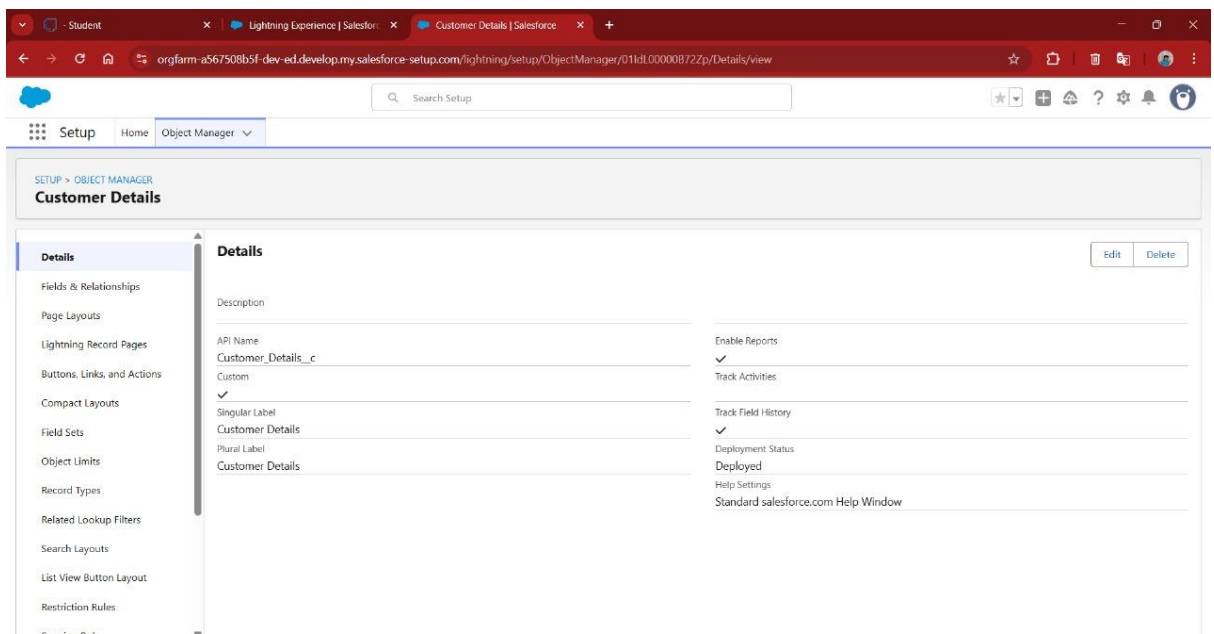
1. Field creation (e.g., Gmail, Phone, Lookup).



The screenshot shows the Salesforce Setup interface for the 'Appointment' object. The left sidebar contains a navigation menu with options like 'Details', 'Fields & Relationships', 'Page Layouts', 'Lightning Record Pages', 'Buttons, Links, and Actions', 'Compact Layouts', 'Field Sets', 'Object Limits', 'Record Types', 'Related Lookup Filters', 'Search Layouts', 'List View Button Layout', 'Restriction Rules', and 'Scoping Rules'. The main content area is titled 'Appointment' and includes a 'Details' section with the following fields:

- Description
- API Name: Appointment__c
- Custom: ☒
- Singular Label: Appointment
- Plural Label: Appointments
- Enable Reports: ☒
- Track Activities: ☒
- Track Field History: ☒
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

Buttons for 'Edit' and 'Delete' are visible in the top right corner of the details section.



The screenshot shows the Salesforce Setup interface for the 'Customer Details' object. The left sidebar contains a navigation menu with options like 'Details', 'Fields & Relationships', 'Page Layouts', 'Lightning Record Pages', 'Buttons, Links, and Actions', 'Compact Layouts', 'Field Sets', 'Object Limits', 'Record Types', 'Related Lookup Filters', 'Search Layouts', 'List View Button Layout', 'Restriction Rules', and 'Scoping Rules'. The main content area is titled 'Customer Details' and includes a 'Details' section with the following fields:

- Description
- API Name: Customer_Details__c
- Custom: ☒
- Singular Label: Customer Details
- Plural Label: Customer Details
- Enable Reports: ☒
- Track Activities: ☒
- Track Field History: ☒
- Deployment Status: Deployed
- Help Settings: Standard salesforce.com Help Window

Buttons for 'Edit' and 'Delete' are visible in the top right corner of the details section.

Custom Field Creation in Customer Details Object

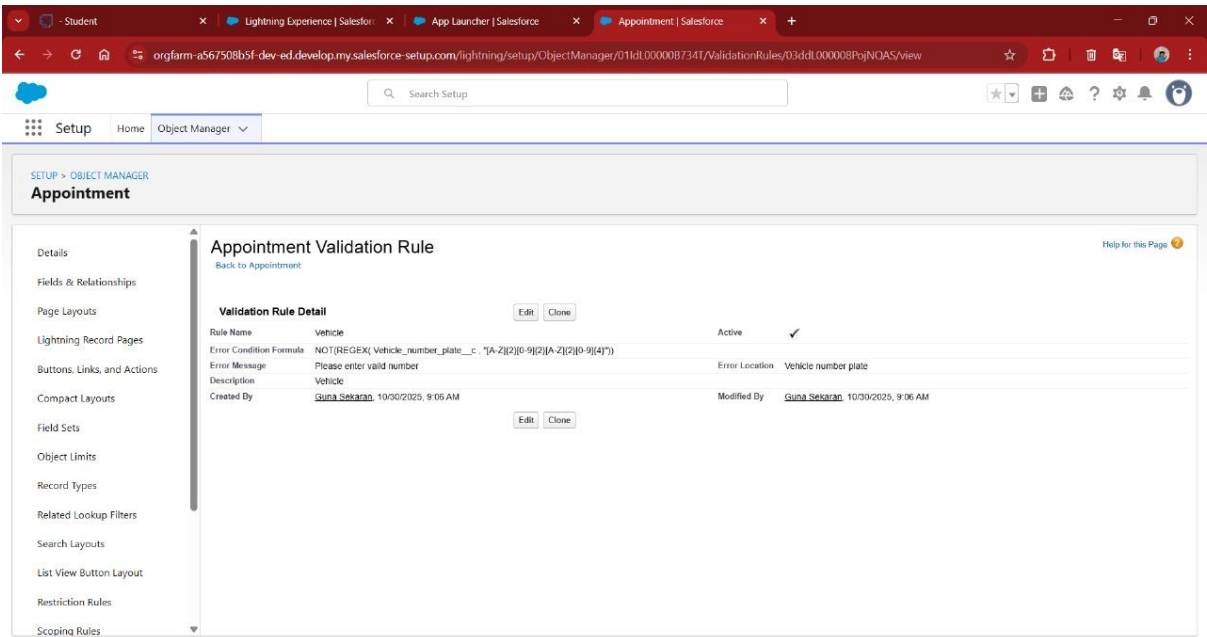
2. Lightning App setup showing all tabs.

The screenshot shows the 'App Details & Branding' configuration page in the Lightning App Builder. The left sidebar lists the navigation tabs: App Settings, App Details & Branding (selected), App Options, Utility Items (Desktop Only), Navigation Items, and User Profiles. The main content area is divided into two sections: 'App Details' and 'App Branding'. In the 'App Details' section, the 'App Name' is 'Garage Management Application', the 'Developer Name' is 'Garage_Management_Application', and the 'Description' field is empty. In the 'App Branding' section, an 'Image' upload button is present, and the 'Primary Color Hex Value' is set to '#0070D2'. Below this, the 'Org Theme Options' section has an unchecked checkbox for 'Use the app's image and color instead of the org's custom theme'. At the bottom, the 'App Launcher Preview' shows a blue square icon with 'GM' and the text 'Garage Management Appli...'.

The screenshot shows the 'User Profiles' configuration page in the Lightning App Builder. The left sidebar is the same as the previous screenshot, with 'User Profiles' now selected. The main content area is titled 'User Profiles' and includes the instruction 'Choose the user profiles that can access this app.' It is divided into 'Available Profiles' and 'Selected Profiles'. The 'Available Profiles' list includes: Analytics Cloud Integration User, Analytics Cloud Security User, Anypoint Integration, Authenticated Website (two entries), B2B Reordering Portal Buyer Profile, Contract Manager, Cross Org Data Proxy User, Custom: Marketing Profile, Custom: Sales Profile, and Custom: Support Profile. The 'Selected Profiles' list is currently empty. Navigation arrows are located between the two lists.

Garage Management Lightning App Setup

3.Validation rule page.



Validation Rule for Vehicle Number Plate

4. Requirement Analysis

In this phase, both functional and non-functional requirements of the system are clearly defined.

Functional Requirements:

- 1. Add, view, and manage Customer, Appointment, and Billing records.**
- 2. Automate updates — for example, when payment is completed, mark status as “Paid.”**
- 3. Send email alerts automatically upon successful billing.**
- 4. Allow users to generate reports and dashboards for performance tracking.**

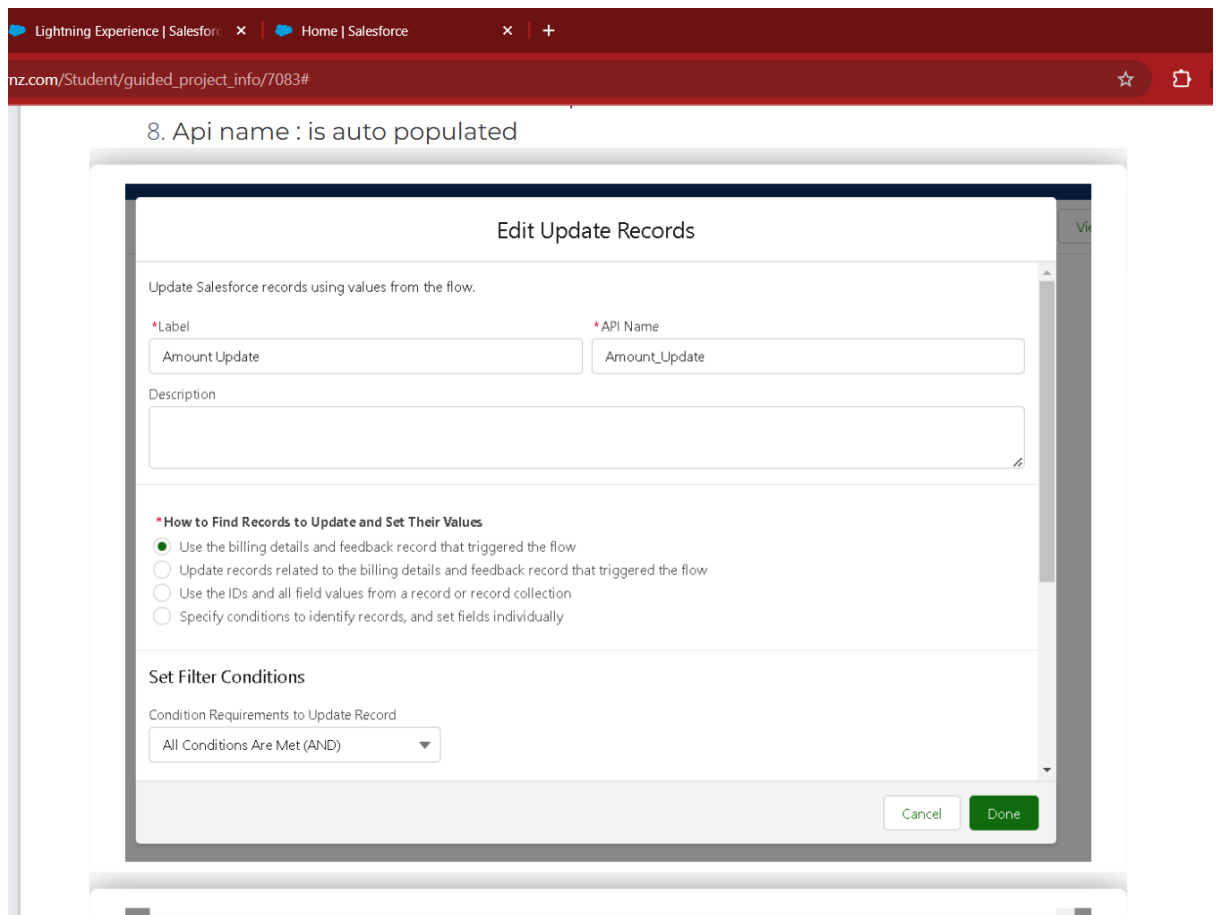
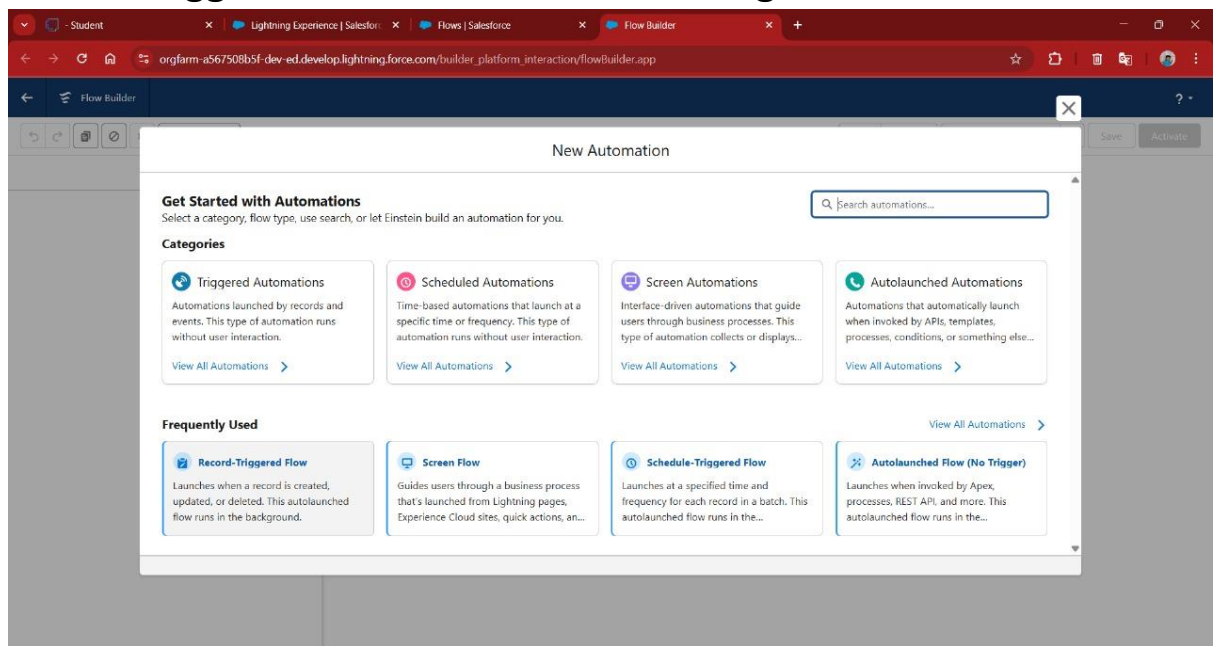
Non-Functional Requirements:

- 1. The system should be user-friendly with a simple interface.**
- 2. Data should be secured based on role-based access.**
- 3. Performance should be optimized for fast operations.**
- 4. Validation and duplicate prevention to ensure accuracy.**

Implementation Tools:

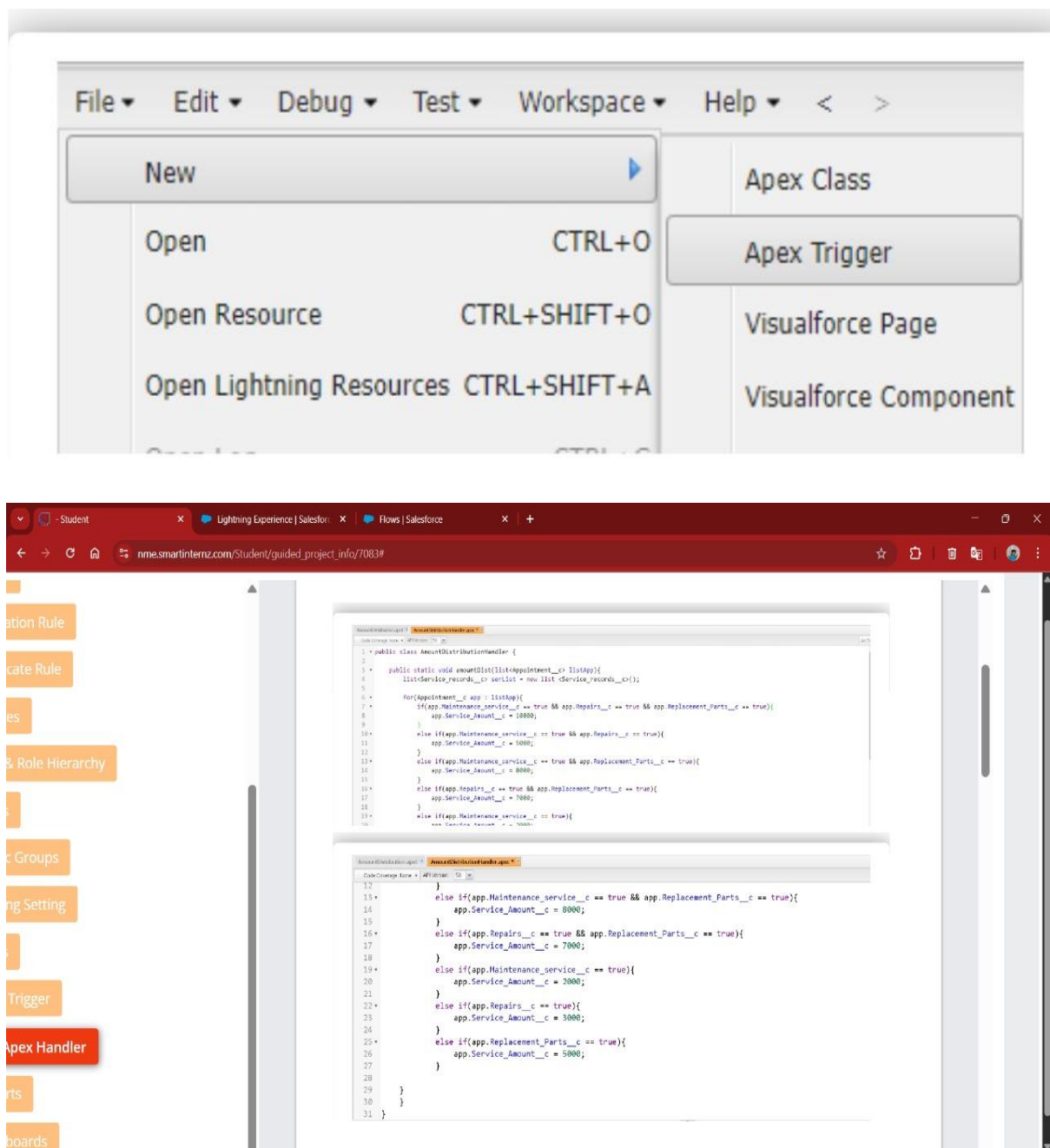
- Salesforce Objects, Flows, Apex Trigger, Validation & Duplicate Rules.**
- Reports and Dashboards for analysis.**

1. Record-triggered flow showing automation.



Record-Triggered Amount Update

2.Apex Trigger window in Developer Console.



Apex Trigger for Amount Distribution

5. Performance Testing

After implementation, the system underwent multiple testing cycles to verify accuracy, performance, and automation.

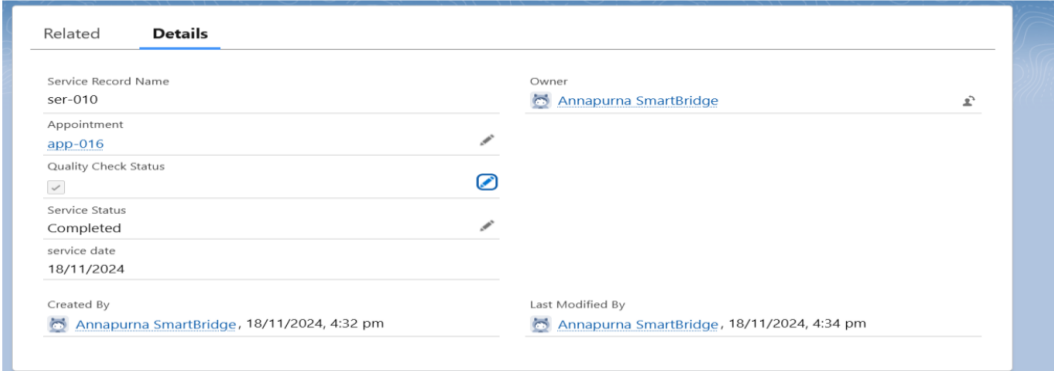
Tests Performed:

- **Flow Test:** Verified that upon “Payment Completed,” an email alert is automatically sent.
- **Trigger Test:** Ensured service amount is correctly updated based on selected checkboxes (Maintenance, Repairs, Replacement Parts).
- **Validation Rule Test:** Tested invalid inputs for vehicle number and rating.
- **Report & Dashboard Test:** Verified that reports accurately display service and payment data.

The system performed efficiently with accurate automation and responsive performance under typical garage workloads.

1. Created Status records for all objects.

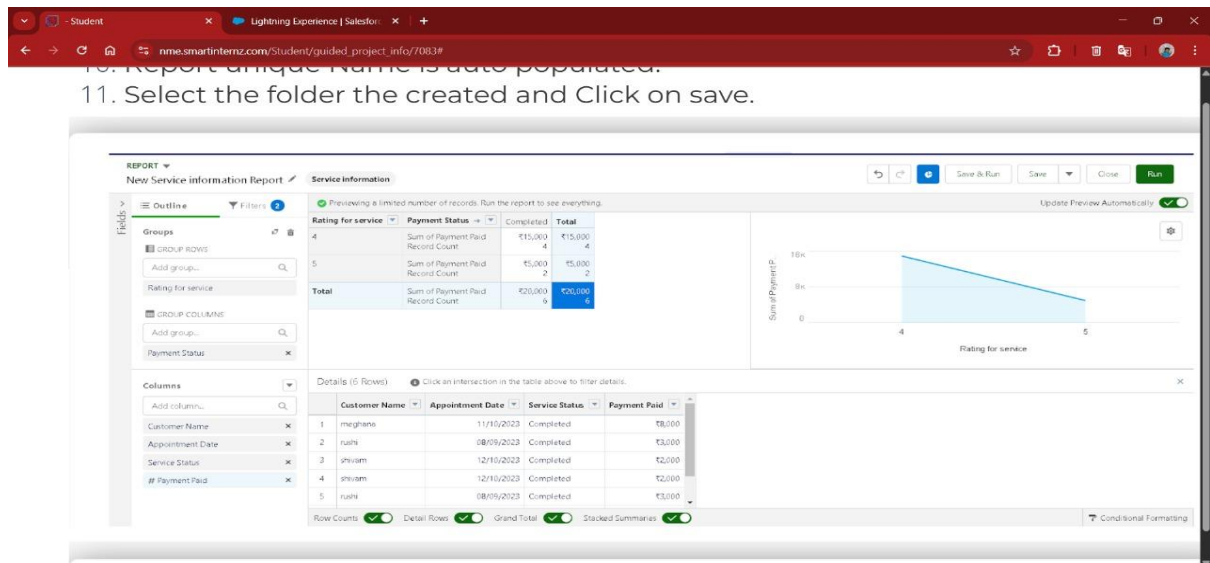
6. Now automatically Service status will be moved to completed.



The screenshot displays a 'Details' view of a service record. The interface includes a 'Related' tab and a 'Details' tab. The 'Details' tab shows the following information:

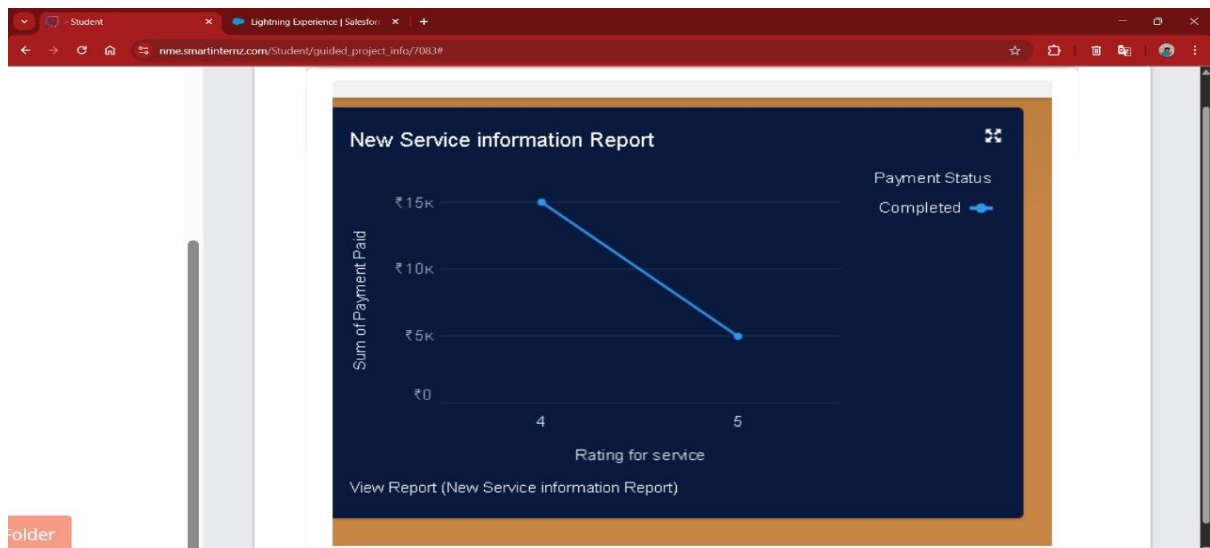
Field	Value
Service Record Name	ser-010
Appointment	app-016
Quality Check Status	<input checked="" type="checkbox"/>
Service Status	Completed
Service date	18/11/2024
Created By	Annapurna SmartBridge, 18/11/2024, 4:32 pm
Last Modified By	Annapurna SmartBridge, 18/11/2024, 4:34 pm

2.Report tab showing generated report.



Service Information Report

3.Dashboard line chart view.



Service Rating Dashboard