



SB8067 Salesforce Developer



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

Title : 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,Puliangudi.**

**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

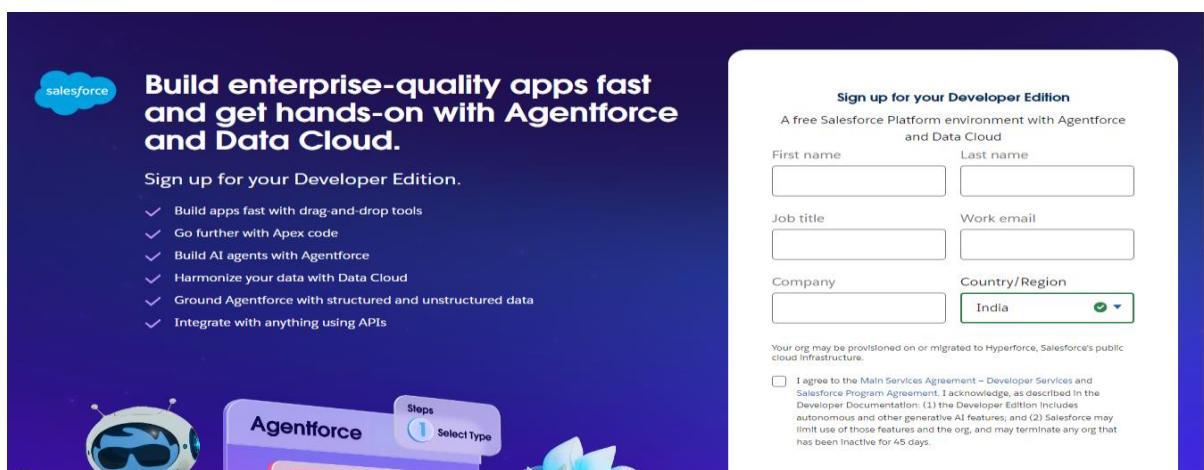
The Garage Management System is designed to simplify and automate the daily operations of a garage using Salesforce CRM.

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.

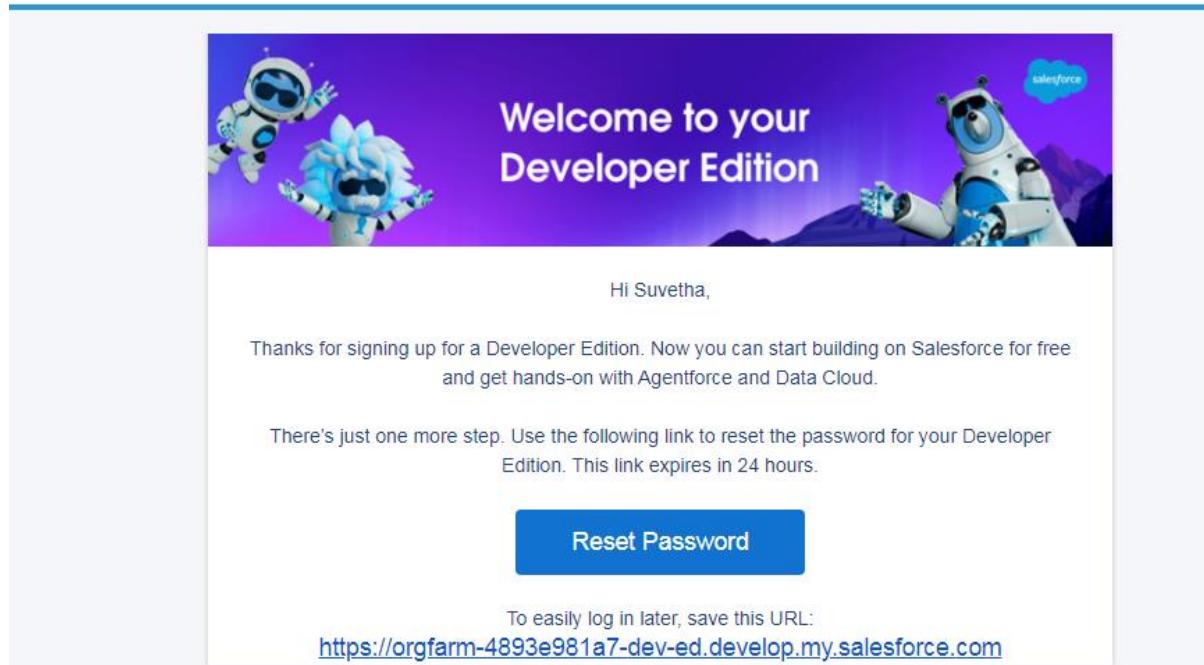
This phase focuses on identifying challenges such as delayed service tracking, missing customer data, and billing confusion — and proposing Salesforce automation as the solution.

### 1. Salesforce Developer Signup Page



**Figure 1.1: Salesforce Developer Org Signup Page**

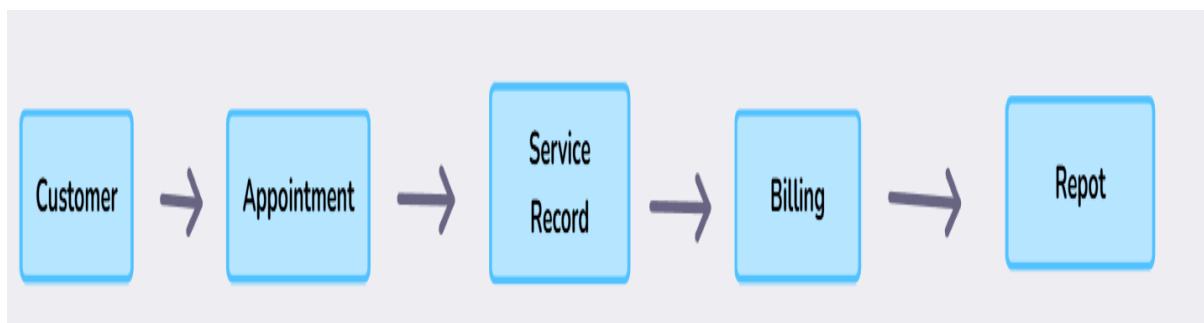
## **2.Salesforce Setup Home Page**



## **Salesforce Setup Page (Post Activation)**

## **3.Garage Management System Concept Diagram**

- o **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.

The screenshot shows the Salesforce Object Manager page. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'Customer Details'. On the left, a sidebar titled 'Details' lists various setup categories: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Details' and contains fields for 'Description', 'API Name' (set to 'Customer\_Details\_\_c'), 'Custom' (set to '✓'), 'Singular Label' (set to 'Customer Details'), and 'Plural Label' (set to 'Customer Details'). To the right, there are sections for 'Enable Reports' (checked), 'Track Activities', 'Track Field History' (checked), 'Deployment Status' (set to 'Deployed'), and 'Help Settings'. At the bottom right are 'Edit' and 'Delete' buttons.

## Object Manager – Creating Custom Objects

## 2. List of all custom objects created.

The screenshot shows the 'New Custom Object' creation page. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main title is 'New Custom Object'. The page has several sections: 'Allow in Master Groups' (unchecked), 'Enable Licensing' (unchecked). 'Object Classification' section: 'Allow Sharing' (checked), 'Allow Bulk API Access' (checked), 'Allow Streaming API Access' (checked). 'Deployment Status' section: 'In Development' (radio button) is selected. 'Search Status' section: 'Allow Search' (checkbox) is unchecked. 'Object Creation Options' section: 'Add Notes and Attachments related list to default page layout' (checkbox) and 'Launch New Custom Tab Wizard after saving this custom object' (checkbox) are both unchecked. At the bottom are 'Save', 'Save & New', and 'Cancel' buttons.

## List of Garage Management Custom Objects

### 3.Role hierarchy showing Manager → Salesperson.

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

- Header:** Search Setup, Home, Object Manager
- Page:** SETUP > OBJECT MANAGER, Opportunity Contact Role
- Left sidebar:** Details, Fields & Relationships (selected), Page Layouts, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Triggers, Flow Triggers, Validation Rules.
- Table:** Fields & Relationships (6 items, sorted by Field Label)

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.

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

- Header:** Search Setup, Home, Object Manager
- Page:** SETUP > OBJECT MANAGER, Service records
- Left sidebar:** Details, Fields & Relationships (selected), 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.
- Table:** Fields & Relationships (9 items, sorted by Field Label)

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:**

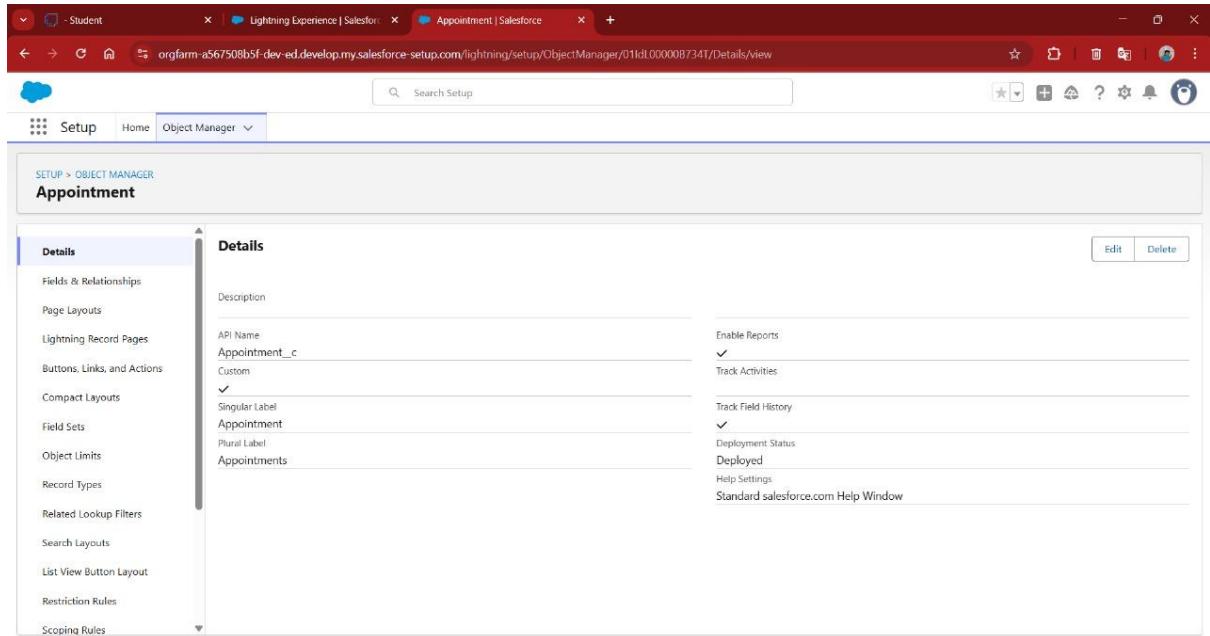
**All objects are grouped into one Garage Management Application. Tabs for each module are added for easy navigation.**

**Data Integrity:**

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

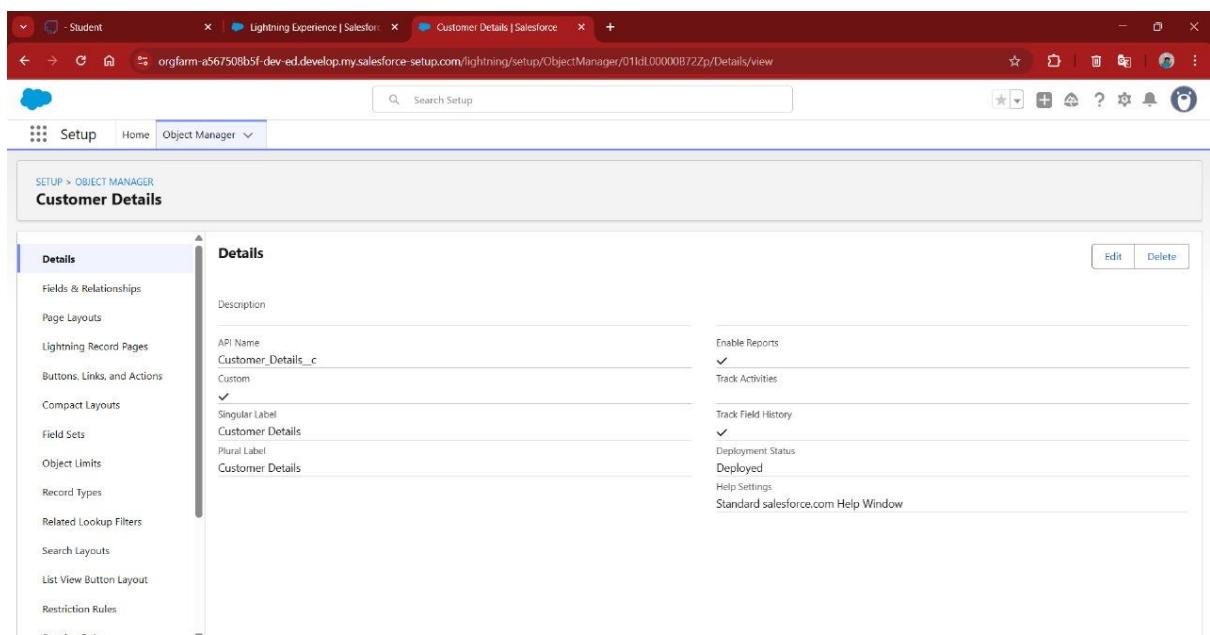
- **Duplicate rules prevent repeated customer records.**
- **Profiles manage user permissions.**

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



The screenshot shows the Salesforce Object Manager interface for the 'Appointment' object. The left sidebar lists various configuration options: 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 'Details' tab is selected, showing the following fields:

Setting	Value
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	



The screenshot shows the Salesforce Object Manager interface for the 'Customer Details' object. The left sidebar lists the same configuration options as the previous screenshot. The main 'Details' tab is selected, showing the following fields:

Setting	Value
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	

## Custom Field Creation in Customer Details Object

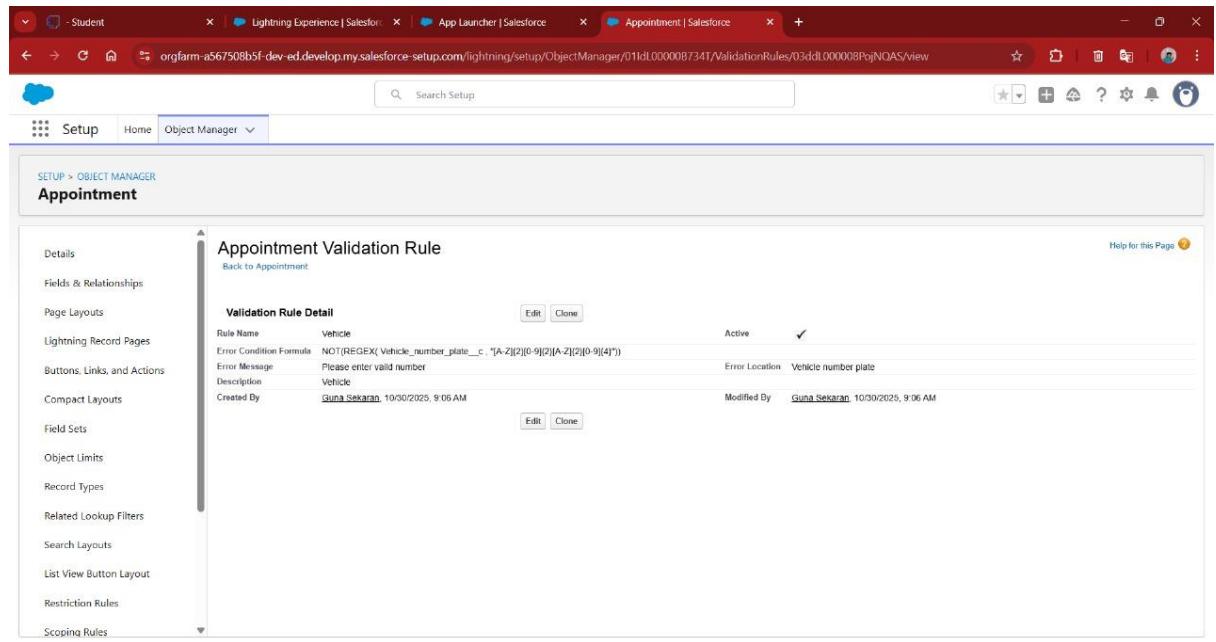
## 2. Lightning App setup showing all tabs.

The screenshot shows the Lightning App Builder interface with the "App Details & Branding" tab selected. On the left, a sidebar lists "App Settings" with "App Details & Branding" highlighted. The main area contains fields for "App Name" (Garage Management Application), "Developer Name" (Garage\_Management\_Application), and "Description" (Enter a description...). It also includes sections for "Image" (with a placeholder box and "Upload" button) and "Primary Color Hex Value" (#0070D2). Below these is a "Org Theme Options" checkbox. At the bottom, there's a preview of the app launcher icon, which is a blue square with "GM" and the text "Garage Management Appl...".

The screenshot shows the Lightning App Builder interface with the "User Profiles" tab selected. On the left, a sidebar lists "App Settings" with "User Profiles" highlighted. The main area shows a list of "Available Profiles" on the left and "Selected Profiles" on the right. The "Available Profiles" list includes: Analytics Cloud Integration User, Analytics Cloud Security User, Anypoint Integration, Authenticated Website, Authenticated Website, 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 on the right contains: System Administrator, Manager, and sales person.

## Garage Management Lightning App Setup

### 3. Validation rule page.



The screenshot shows the Salesforce Setup interface for the 'Appointment' object. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, and Lightning Record Pages. The main content area is titled 'Appointment Validation Rule' and displays the details of a validation rule named 'Vehicle'. The rule's formula is 'NOT(REGEX( Vehicle\_number\_plate\_\_c, "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}") )', which checks if the vehicle number plate does not match the specified regex pattern. The rule is active and was created by 'Guna Sekaran' on 10/30/2025 at 9:06 AM. It has an error message: 'Please enter valid number' and is located on the 'Vehicle' field.

## 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.

The screenshot shows the Salesforce Flow Builder interface with the title 'New Automation'. The 'Get Started with Automations' section displays four categories: 'Triggered Automations', 'Scheduled Automations', 'Screen Automations', and 'Autolaunched Automations'. Below this, the 'Frequently Used' section lists four types of flows: 'Record-Triggered Flow' (selected), 'Screen Flow', 'Schedule-Triggered Flow', and 'Autolaunched Flow (No Trigger)'. A search bar at the top right is labeled 'Search automations...'. A 'Save' and 'Activate' button are located in the top right corner of the main window.

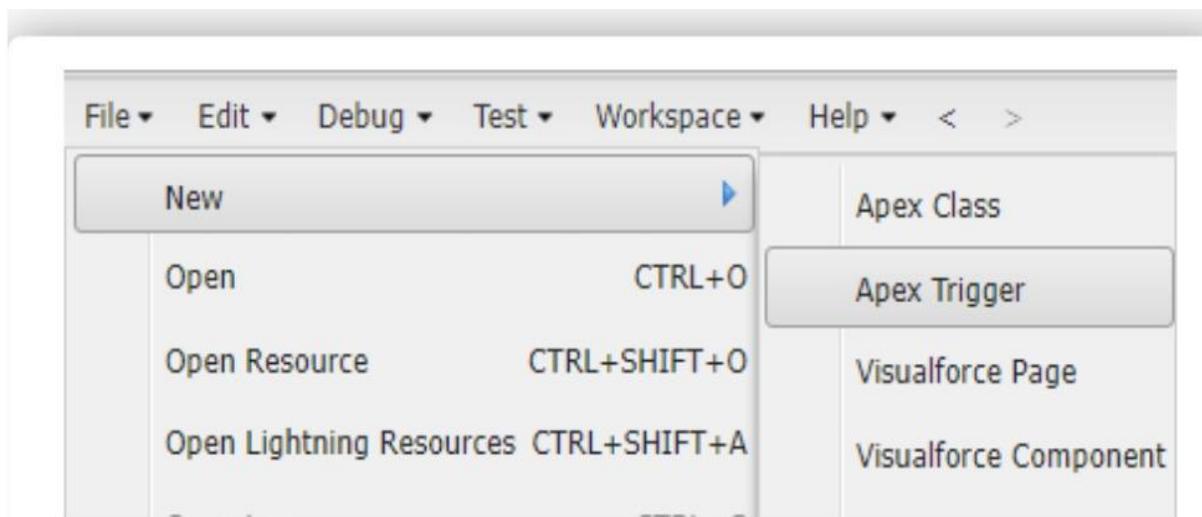
The screenshot shows the 'Edit Update Records' configuration screen. The title bar indicates the URL is nz.com/Student/guided\_project\_info/7083#. The main area is titled 'Edit Update Records' and contains fields for 'Label' (Amount Update) and 'API Name' (Amount\_Update). A 'Description' field is present but empty. Below these fields is a section titled 'How to Find Records to Update and Set Their Values' with the following options:

- 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

Below this is a 'Set Filter Conditions' section with a dropdown menu set to 'All Conditions Are Met (AND)'. At the bottom right are 'Cancel' and 'Done' buttons.

**Record-Triggered Amount Update**

## 2. Apex Trigger window in Developer Console.



The screenshot shows the Apex trigger code for 'AmountDistributionHandler'. The code logic is as follows:

```
public class AmountDistributionHandler {
    public static void amountDistribution(List<Appointment> listApp) {
        List<ServiceRecord> serviceList = new List<ServiceRecord>();
        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;
            }
        }
    }
}
```

**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 shows a software application window with a light blue header bar. Below it is a white form with a blue border. At the top left of the form, there are two tabs: "Related" and "Details", with "Details" being the active tab. The main area contains several data fields:

- Service Record Name: Ser-010
- 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

On the right side of the form, there is an "Owner" section with a small profile icon and the text "Annapurna SmartBridge". There are also three small edit icons (pencil, magnifying glass, and trash) aligned vertically next to the status and service date fields. The bottom right corner of the form has a small circular icon with a question mark.

## 2. Report tab showing generated report.

10. Report unique Name is auto populated.

11. Select the folder the created and Click on save.

The screenshot shows a Salesforce report titled "New Service information Report". On the left, there's a sidebar with "Fields" and "Groups" sections. The main area has a summary table and a line chart. The summary table shows:

	Completed	Total
4	₹15,000 Record Count: 4	₹15,000
5	₹15,000 Record Count: 2	₹15,000
Total	₹30,000 Record Count: 6	₹30,000

The line chart shows a downward trend from a rating of 4 to 5, with values around ₹15,000 for rating 4 and ₹15,000 for rating 5.

## Service Information Report

## 3. Dashboard line chart view.

The screenshot shows a Salesforce dashboard card titled "New Service information Report". The card contains a line chart with the title "Service Rating Dashboard". The Y-axis is labeled "Sum of Payment Paid" and ranges from ₹0 to ₹15k. The X-axis is labeled "Rating for service" and shows points at 4 and 5. A blue line connects two data points: (4, ₹15k) and (5, ₹5k). A legend on the right indicates "Payment Status: Completed". A red button labeled "Folder" is visible on the left side of the card.

## Service Rating Dashboard