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

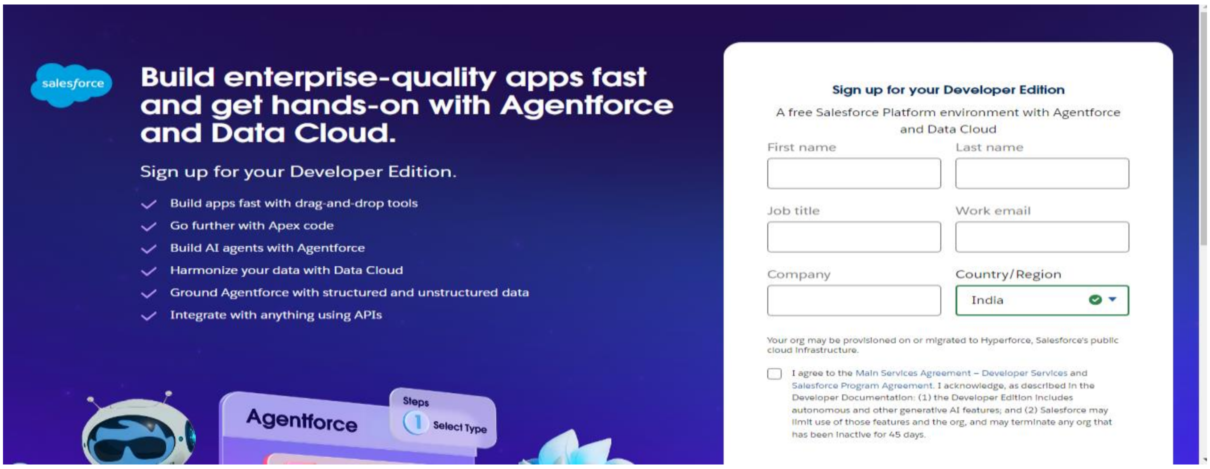
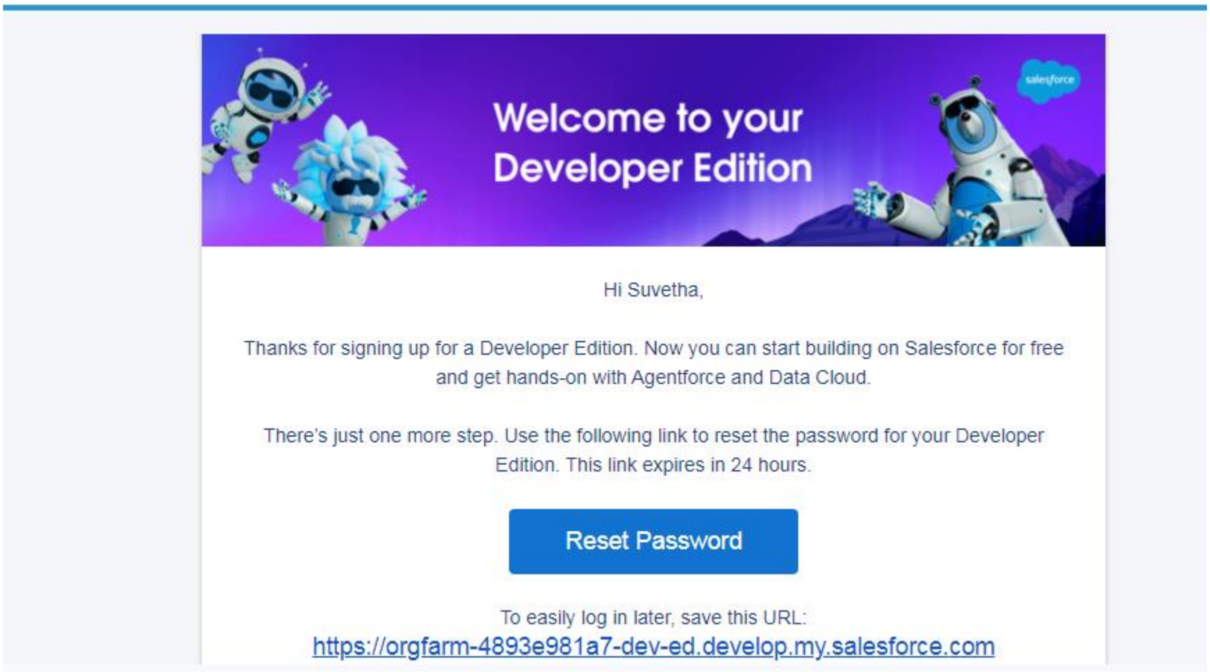


Figure 1.1: Salesforce Developer Org Signup Page

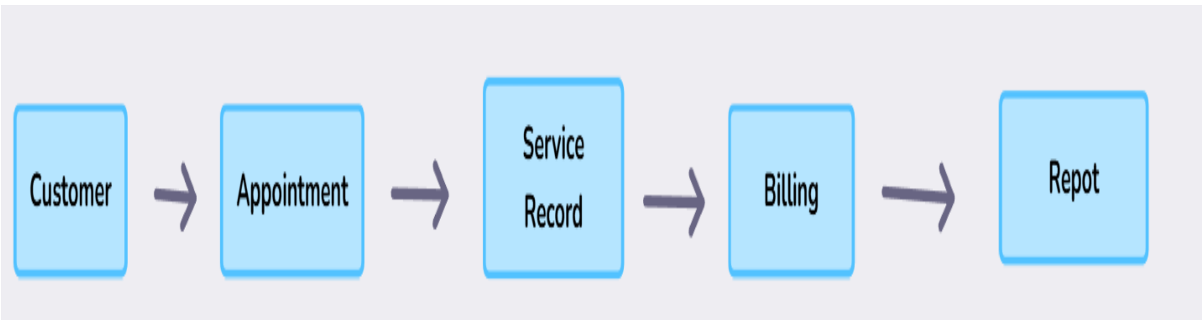
2.Salesforce Setup Home Page



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.

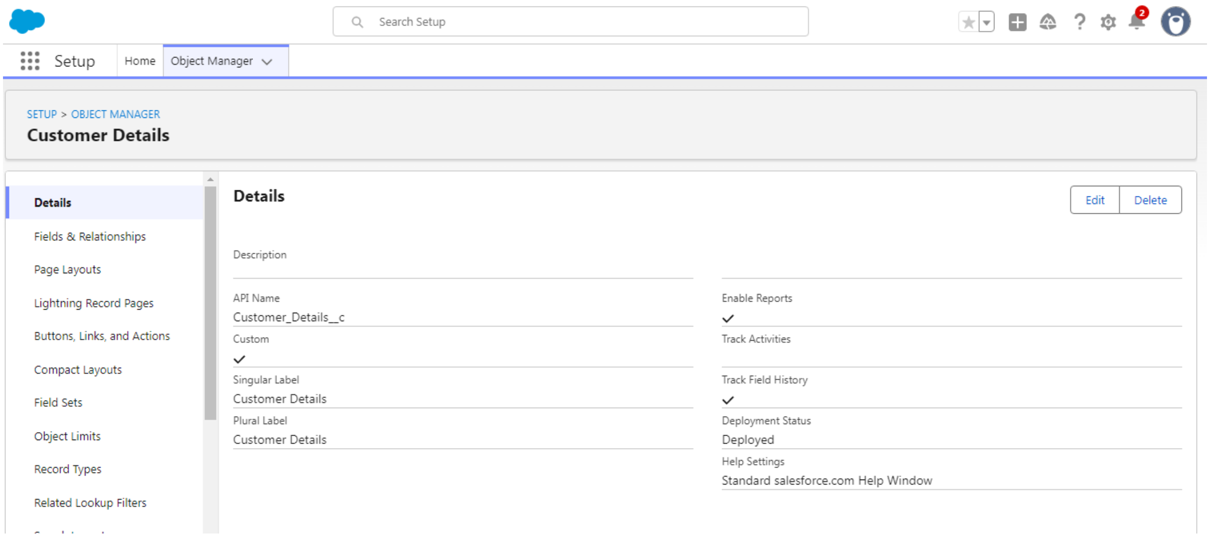
Roles and responsibilities:

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

A structured timeline was followed:

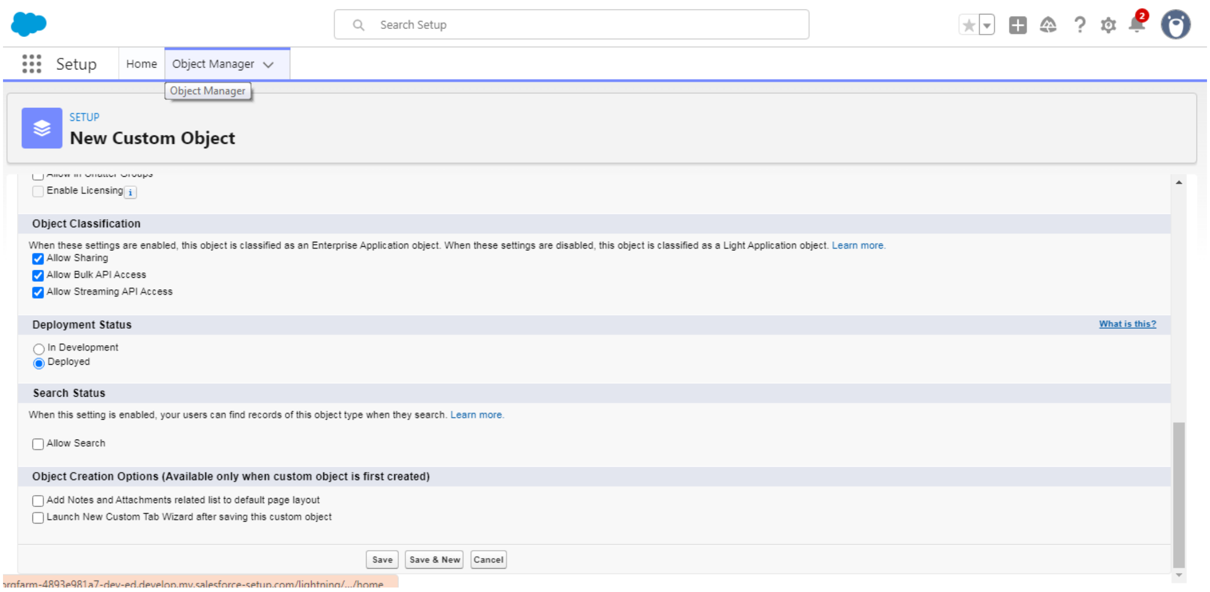
1. Object creation and setup
2. Tab and Lightning app configuration
3. Profile & Role Hierarchy setup
4. Sharing settings and automation flows
5. Testing and deployment

1. Object Manager page – showing all custom objects.



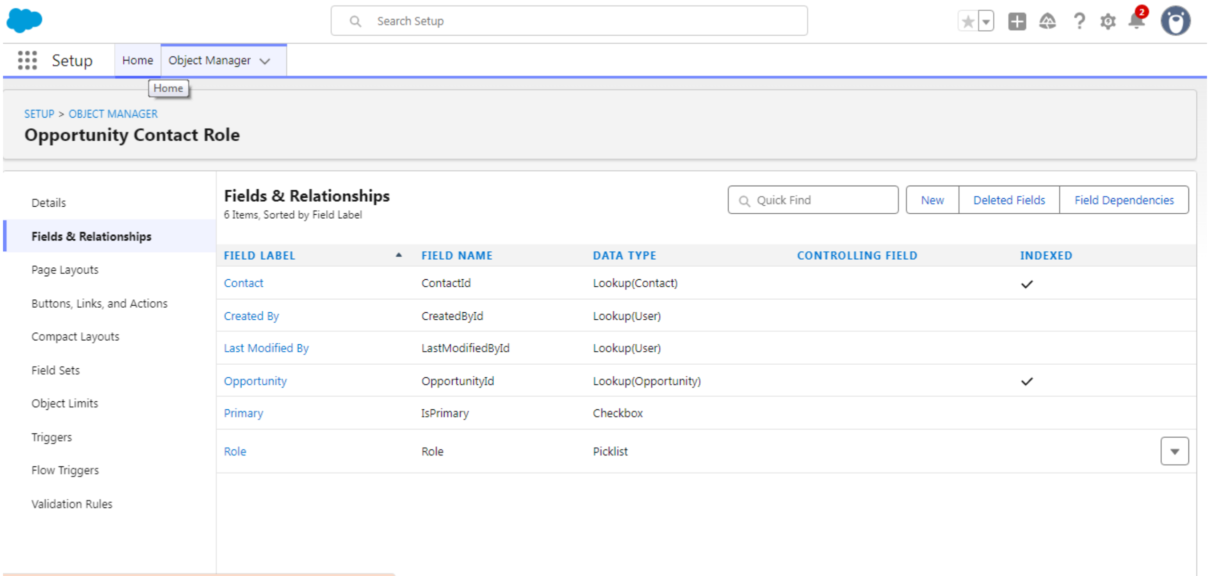
Object Manager – Creating Custom Objects

2. List of all custom objects created.



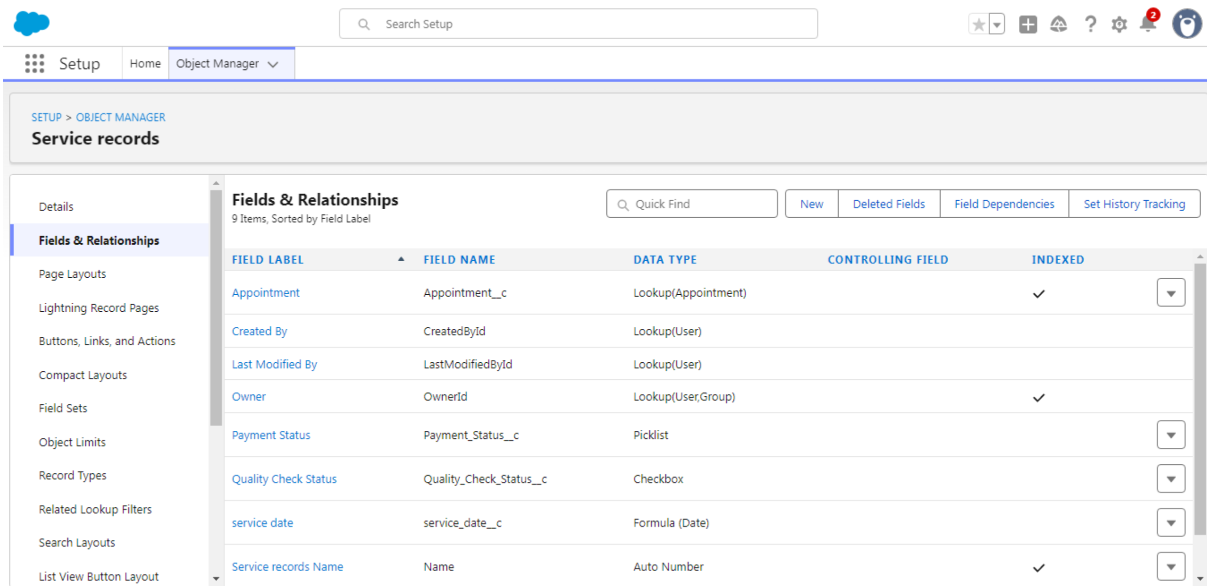
List of Garage Management Custom Objects

3.Role hierarchy showing Manager → Salesperson.



Role Hierarchy for Manager and Salesperson

4.Sharing settings configuration.



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)

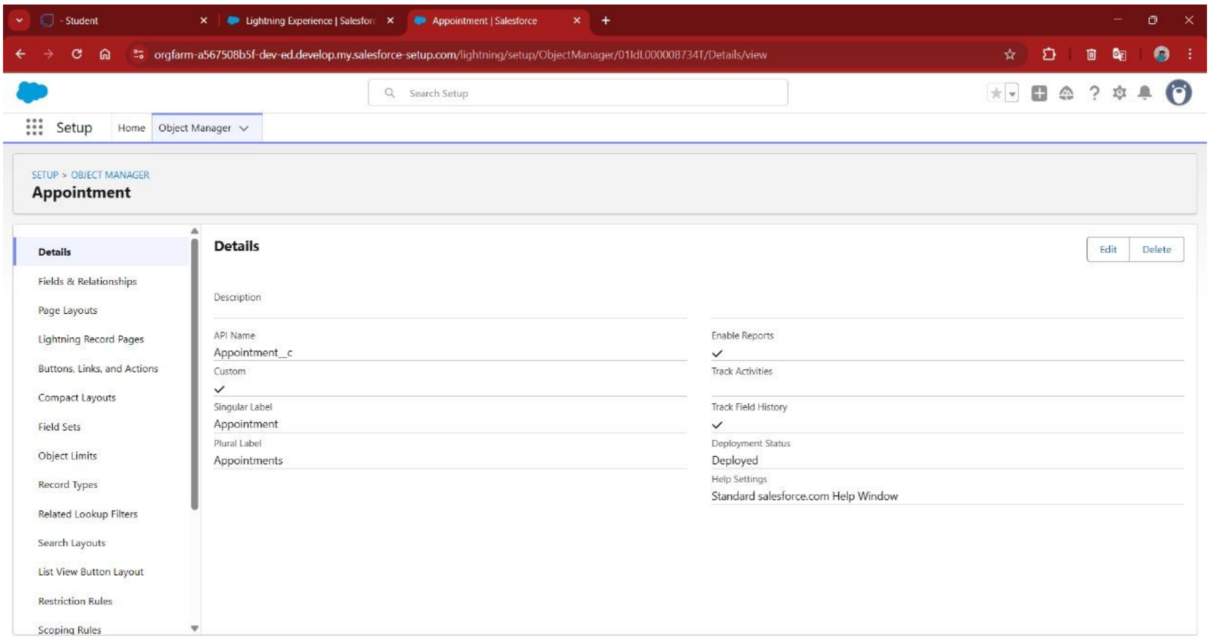
Lightning App Design:
All objects are grouped into one Garage Management Application. Tabs for each module are added for easy navigation.

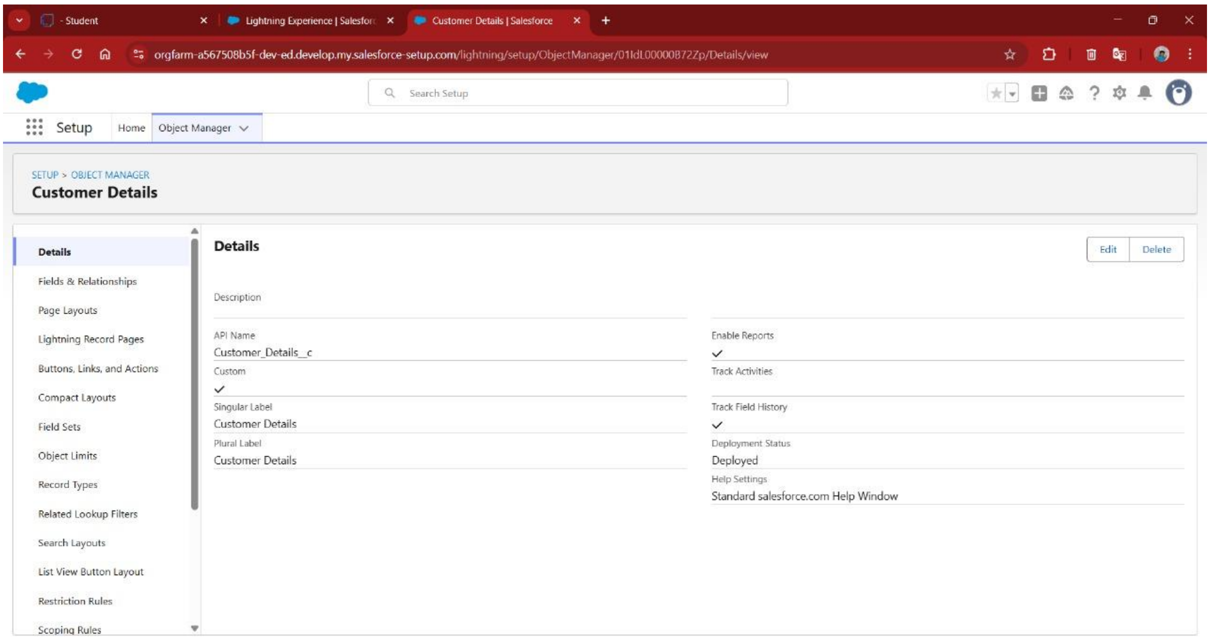
Data Integrity:

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

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

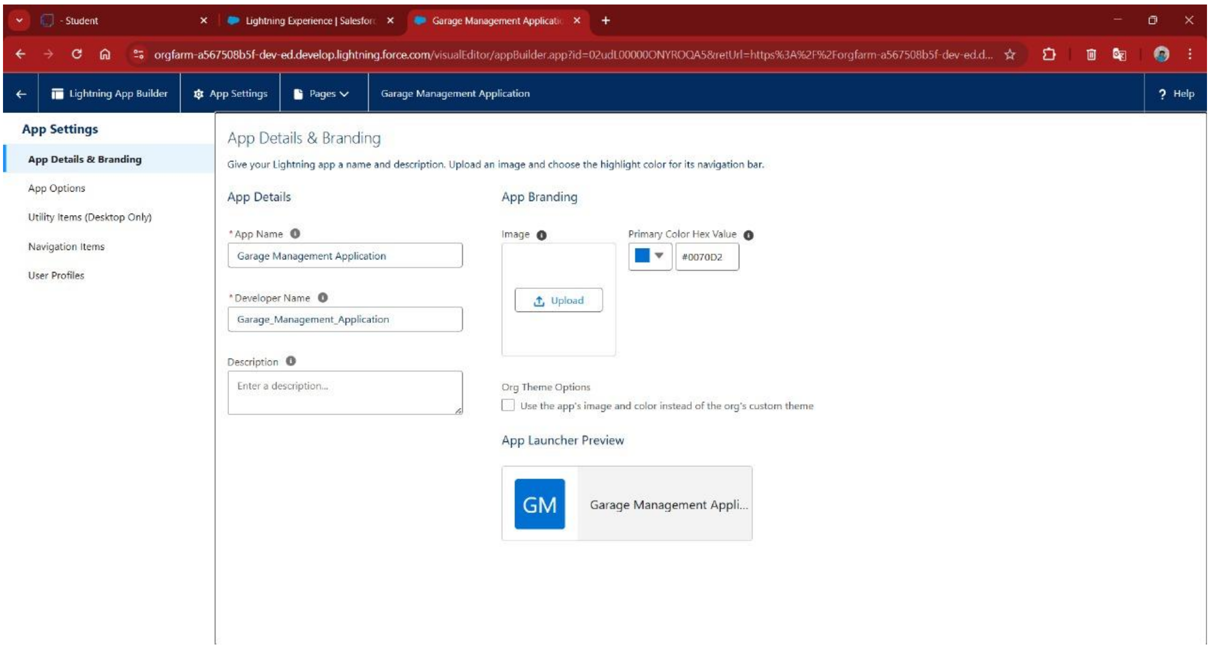
1. Field creation (e.g., Gmail, Phone, Lookup).

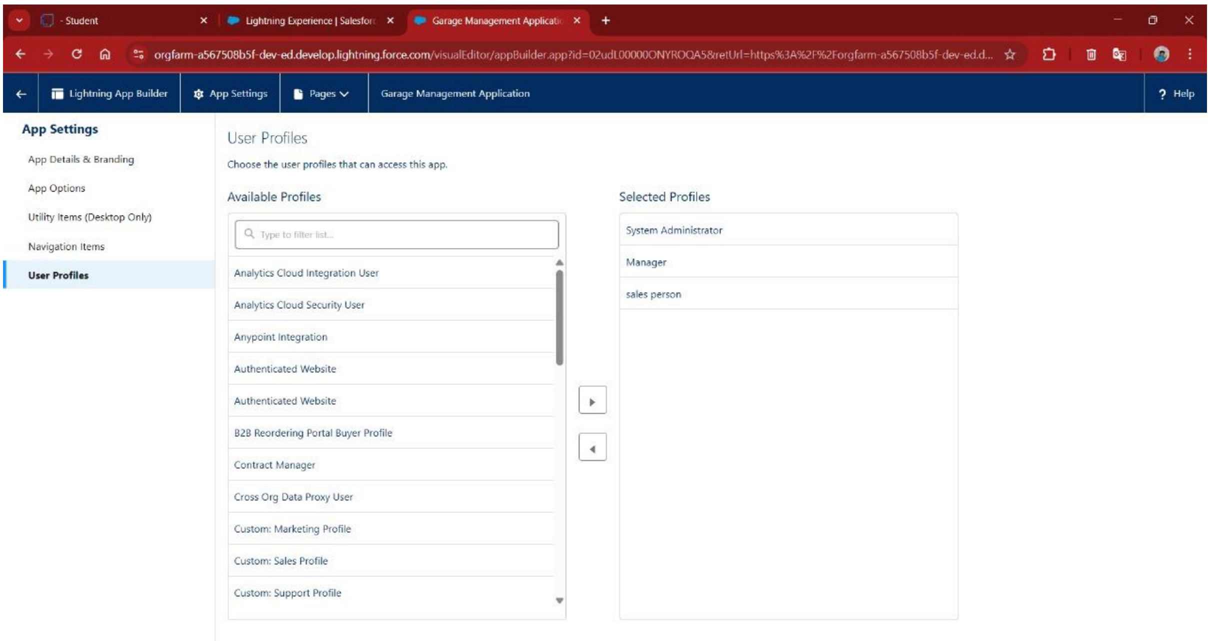




Custom Field Creation in Customer Details Object

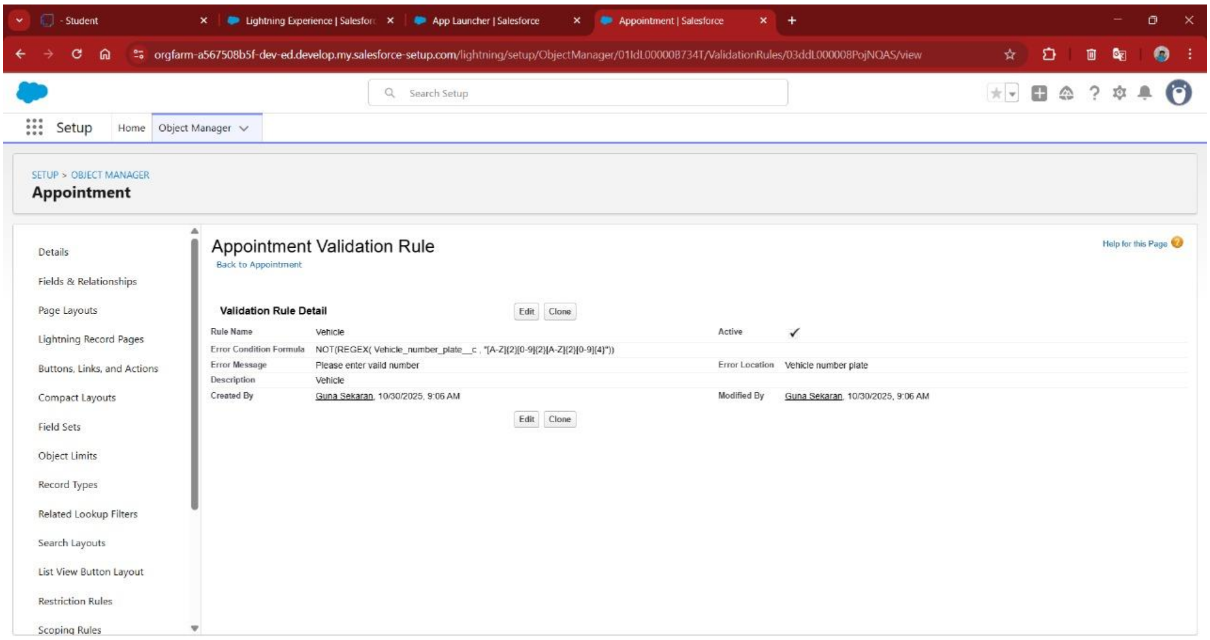
2. Lightning App setup showing all tabs.





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.**

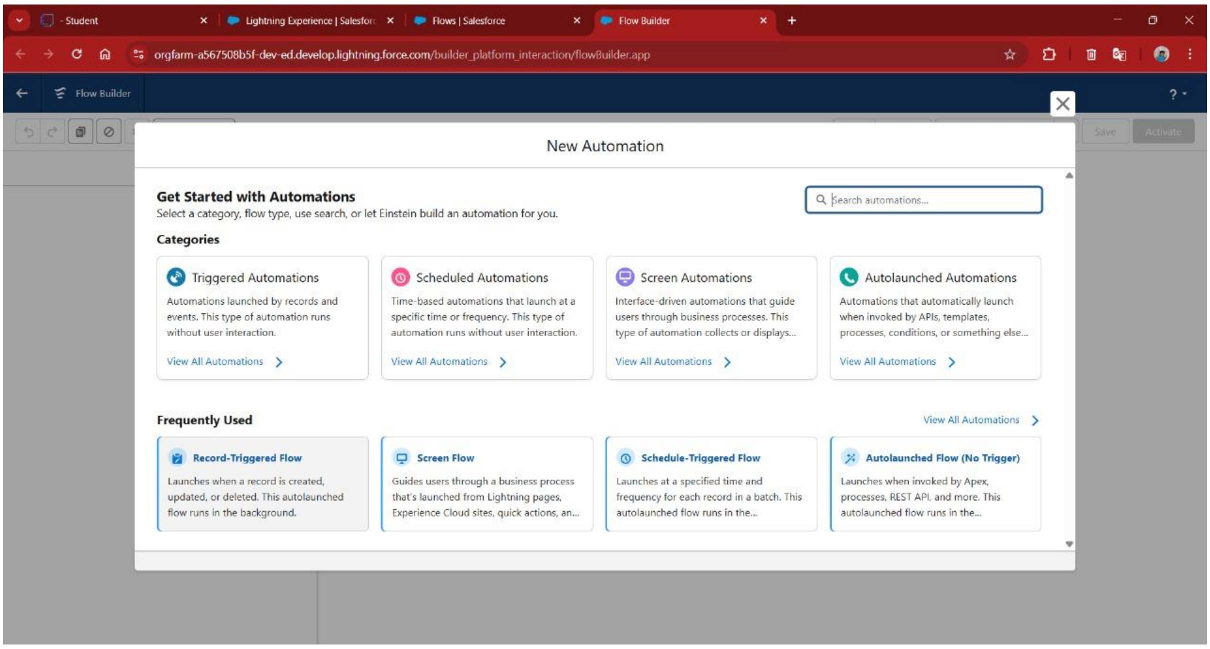
Non-Functional Requirements:

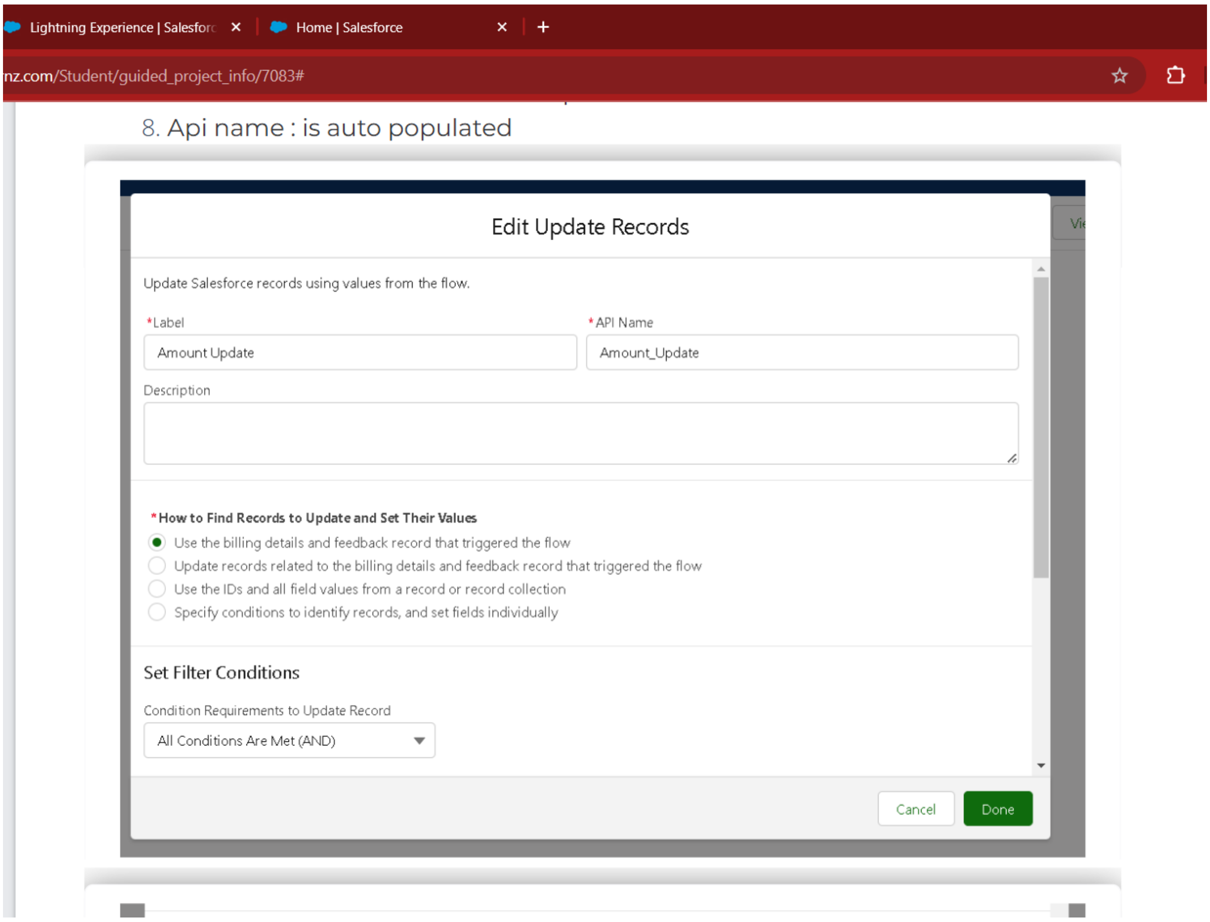
1. **The system should be user-friendly with a simple interface.**
2. **Data should be secured based on role-based access.**
3. **Performance should be optimized for fast operations.**
4. **Validation and duplicate prevention to ensure accuracy.**

Implementation Tools:

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

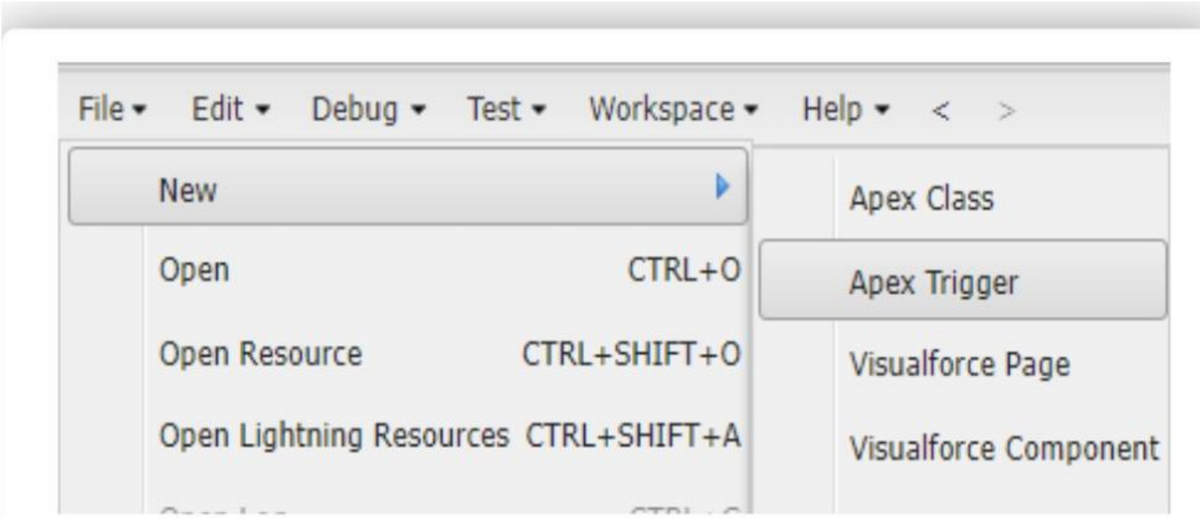
1. **Record-triggered flow showing automation.**

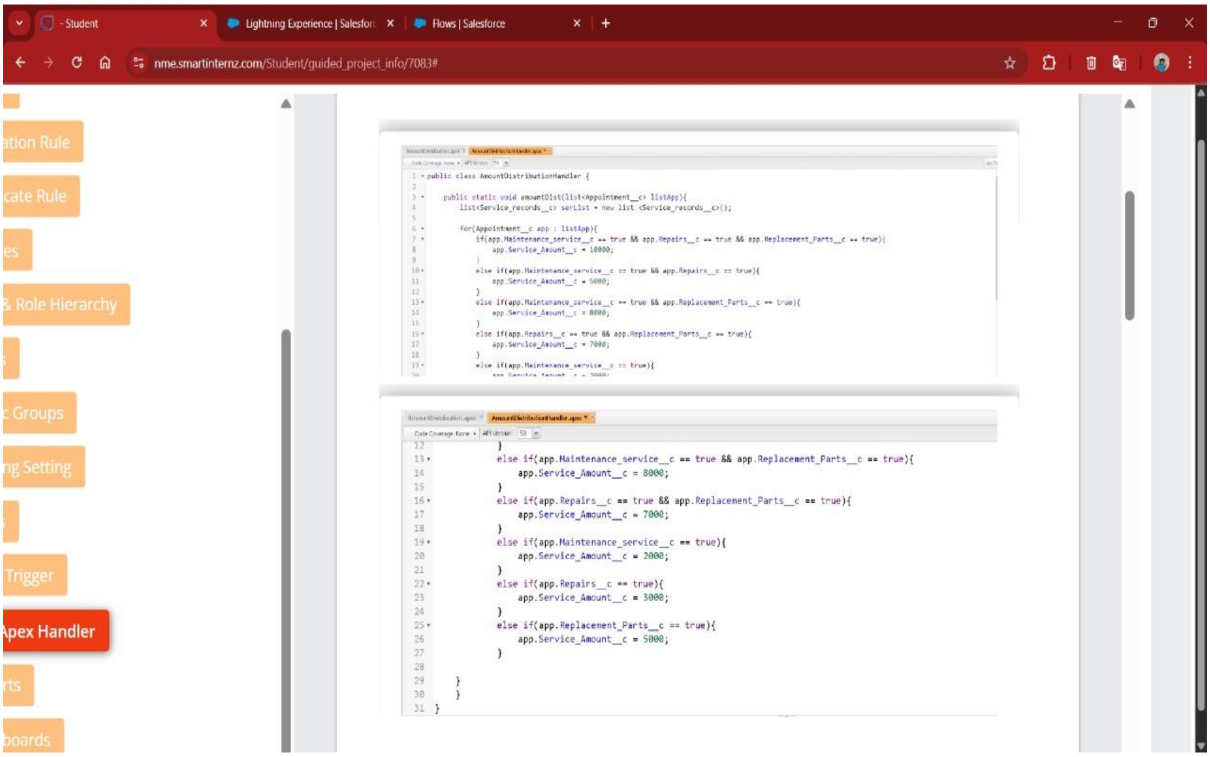




Record-Triggered Amount Update

2.Apex Trigger window in Developer Console.





Apex Trigger for Amount Distribution

5. Performance Testing

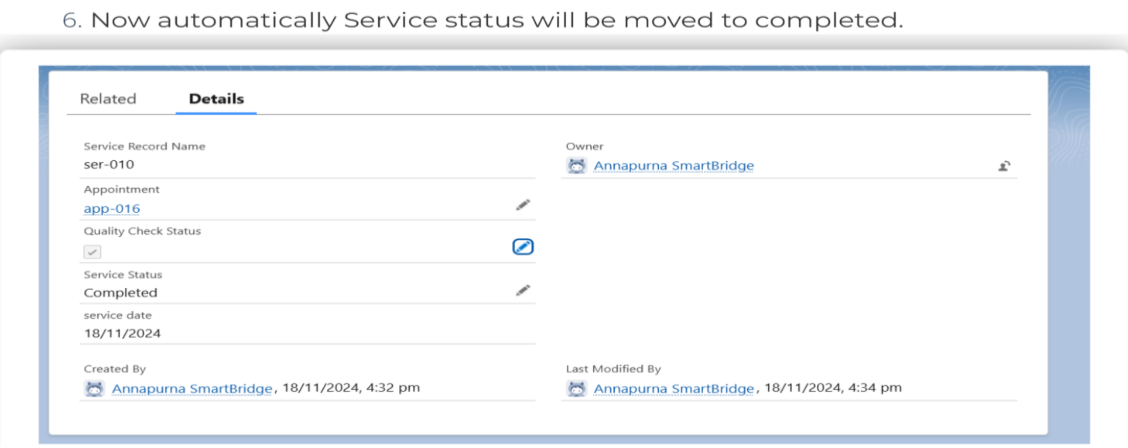
After implementation, the system underwent multiple testing cycles to verify accuracy, performance, and automation.

Tests Performed:

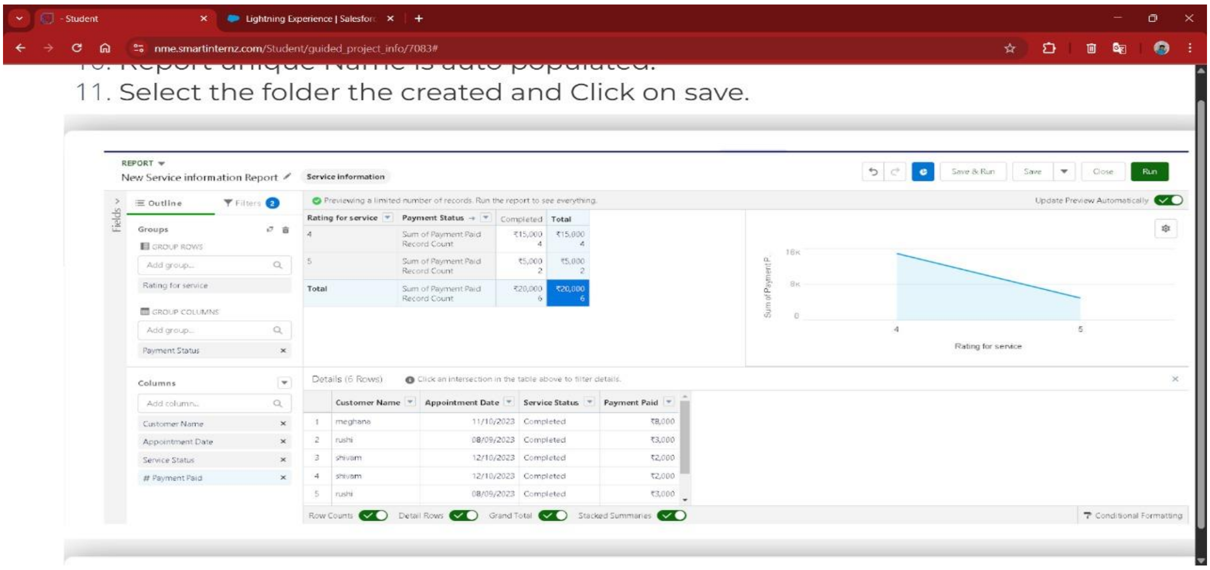
- **Flow Test: Verified that upon "Payment Completed," an email alert is automatically sent.**
- **Trigger Test: Ensured service amount is correctly updated based on selected checkboxes (Maintenance, Repairs, Replacement Parts).**
- **Validation Rule Test: Tested invalid inputs for vehicle number and 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.**

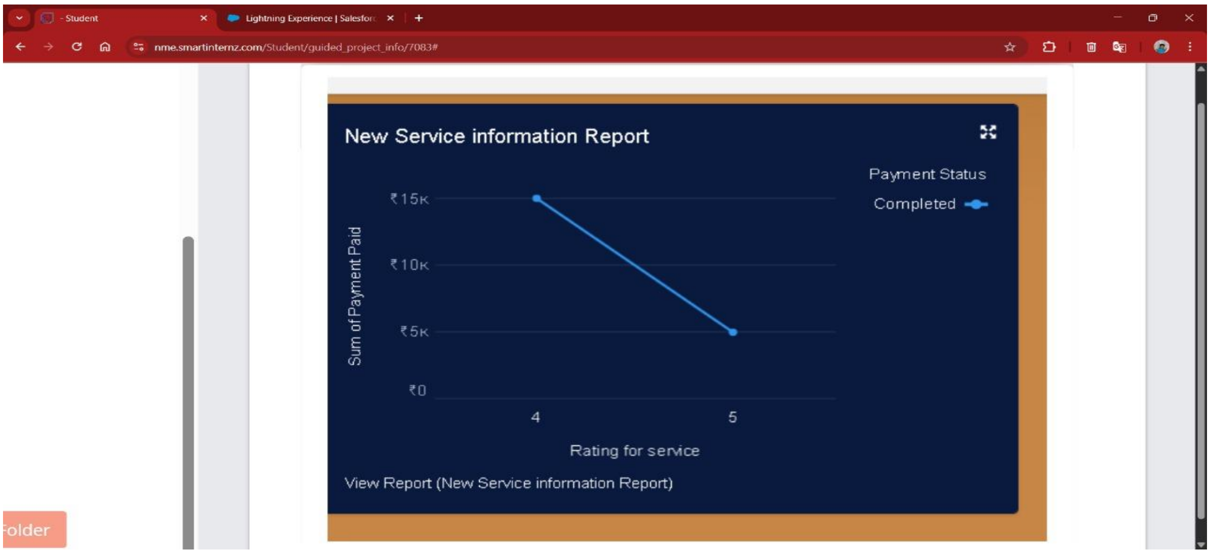


2.Report tab showing generated report.



Service Information Report

3.Dashboard line chart view.



Service Rating Dashboard