**GARAGE MANAGEMENT SYSTEM**

College Name : Shree Venkateshwara Arts And Science (co-education) College,Gobi.

College Code : BRUBD

**TEAM ID :** NM2025TMID20887

**TEAM MEMBERS :**

**Team** **LeaderName** **:** PRANESH.M

**Email :** [pranesh.m75@gmail.com](mailto:pranesh.m75@gmail.com)

**Team Member :** SABARI.B

**Email :** [sabarib26082023cs@gmail.com](mailto:sabarib26082023cs@gmail.com)

**Team Member :** BHARATH.M

**Email :** [bharath12g16052006@gmail.com](mailto:bharath12g16052006@gmail.com)

**Team Member :** GOKUL PRASATH.V

**Email :** [gokulprasath1718@gmail.com](mailto:gokulprasath1718@gmail.com)

**Team member :** SAIYATH AFRITHR.R

**Email :** [saiyathafrithr2023cs@gmail.com](mailto:saiyathafrithr2023cs@gmail.com)

**INTRODUCTION:**

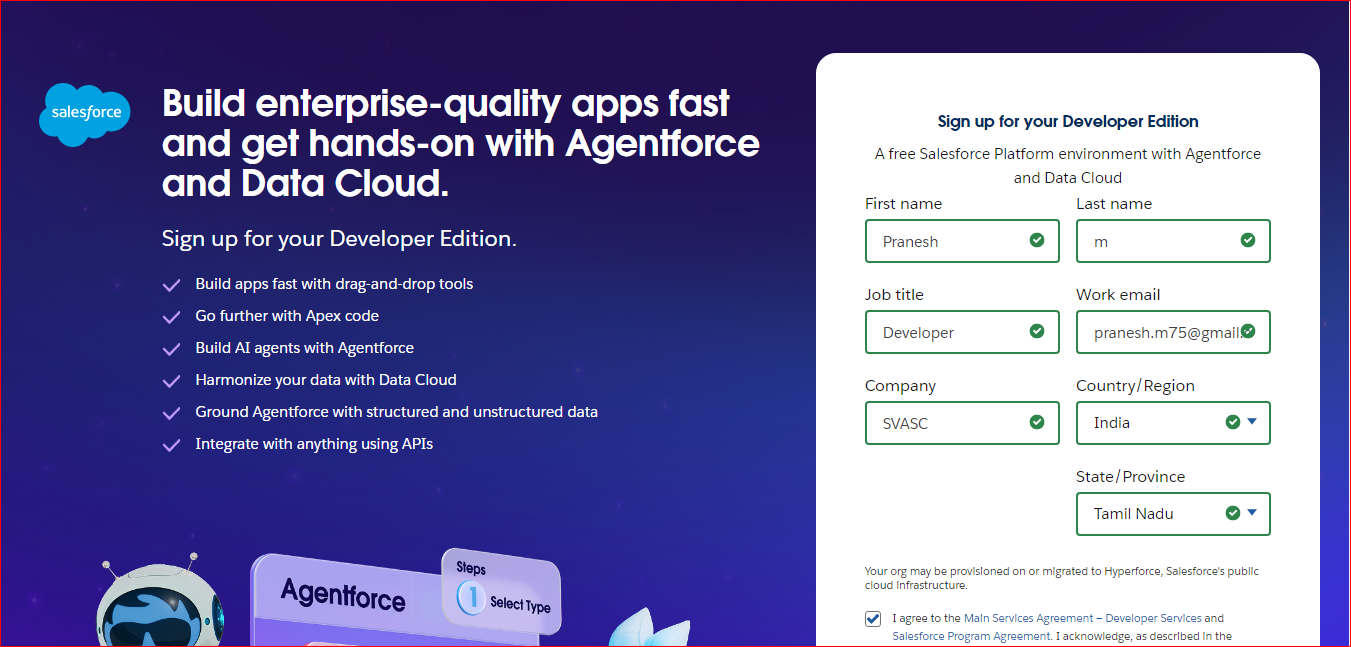
A Garage Management System (GMS) is a software solution designed to streamline and automate the daily operations of a garage or automobile service center. Traditionally, garages rely on manual record-keeping for customer details, vehicle history, job assignments, billing, and inventory management, which often leads to inefficiencies, errors, and time delays.

With a Garage Management System, all essential functions—such as customer registration, vehicle tracking, appointment scheduling, job card creation, spare parts inventory, invoicing, and payment processing—are integrated into a single digital platform. This not only improves accuracy and efficiency but also enhances customer satisfaction by ensuring faster service and transparent communication.

**Salesforce:**

Salesforce is highly customizable with Service Cloud, Sales Cloud, Experience Cloud, and AppExchange apps.

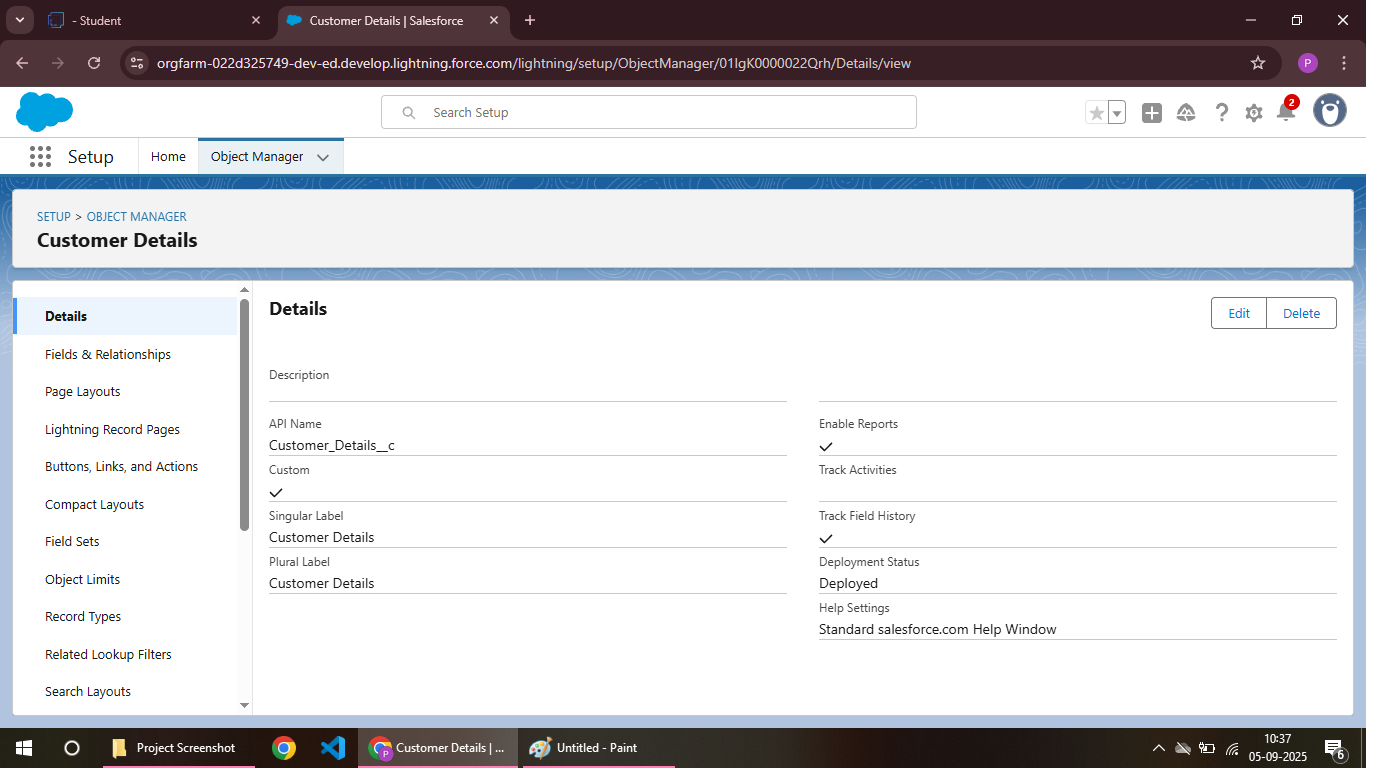
Creating Developer Account:

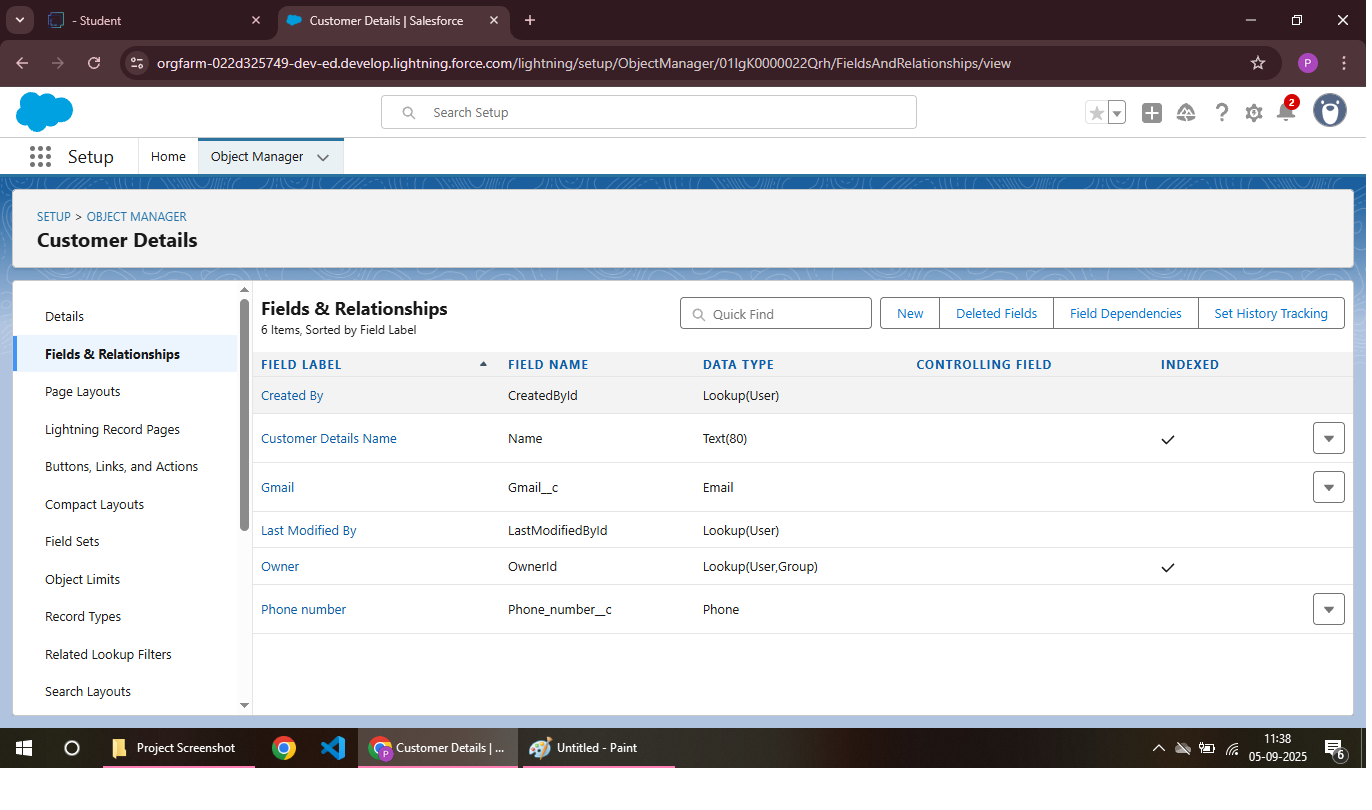
****

**Object :**

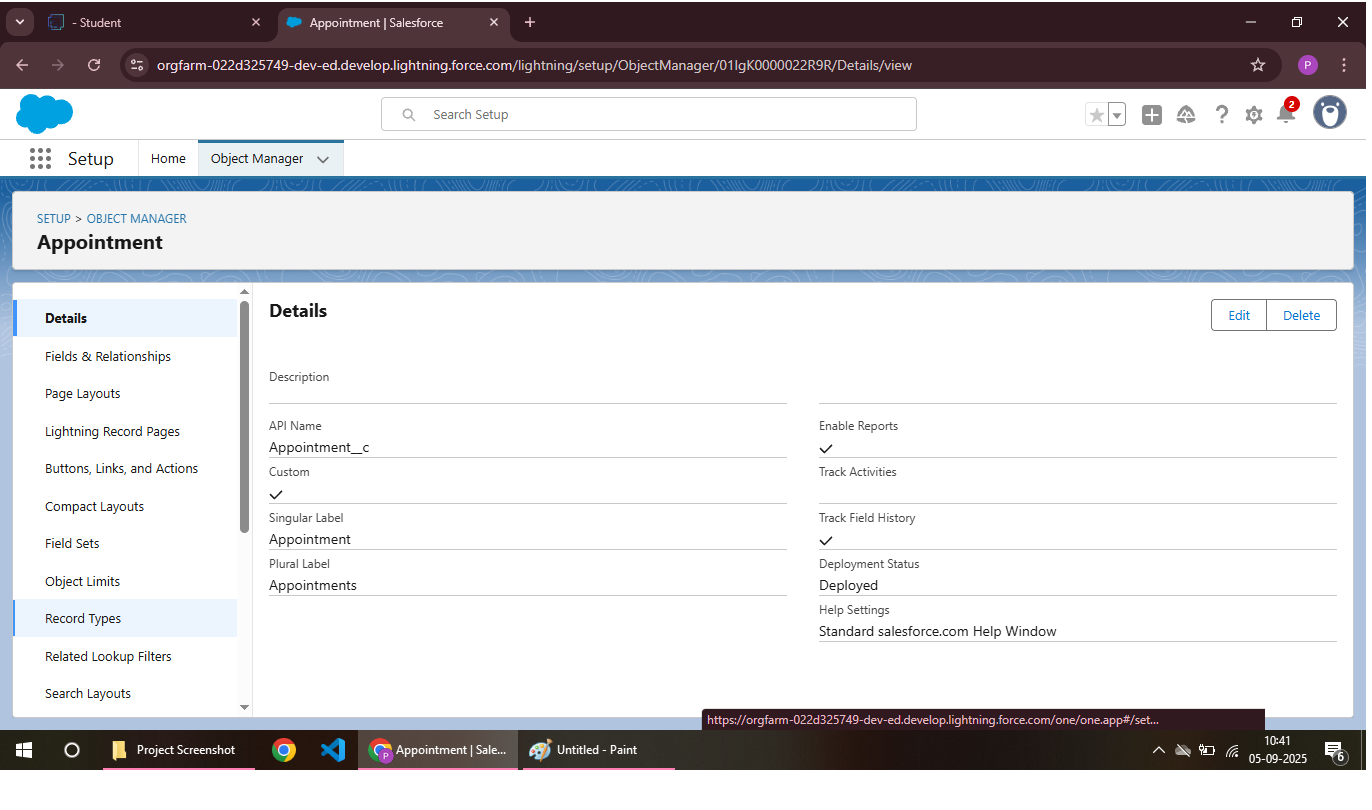
Manages vehicles, services, and appointments. \*Vehicle\* objects store vehicle details, \*Service\* objects define services offered, and \*Appointment\* objects schedule vehicle services. The system provides methods to add, view, and manage these objects efficiently.

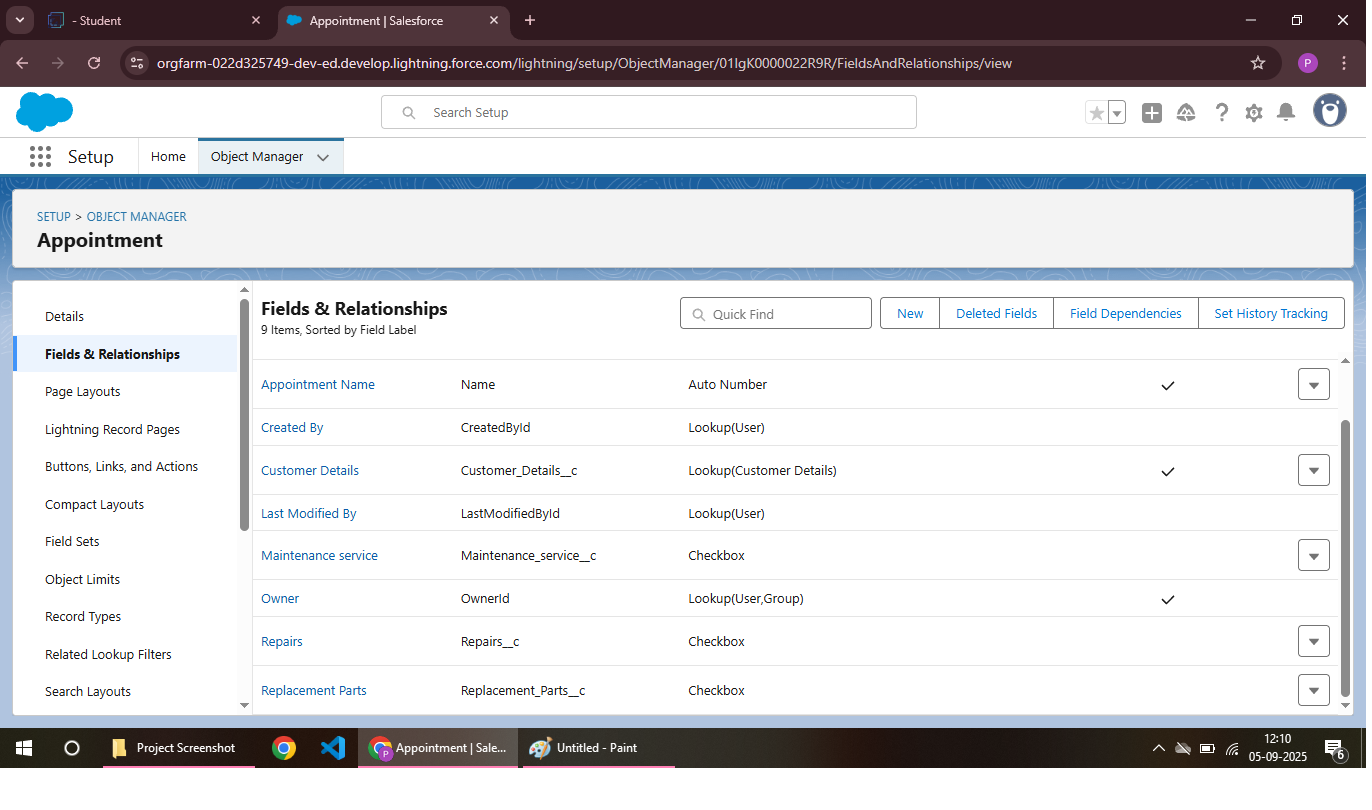
Create Customer Details Object:



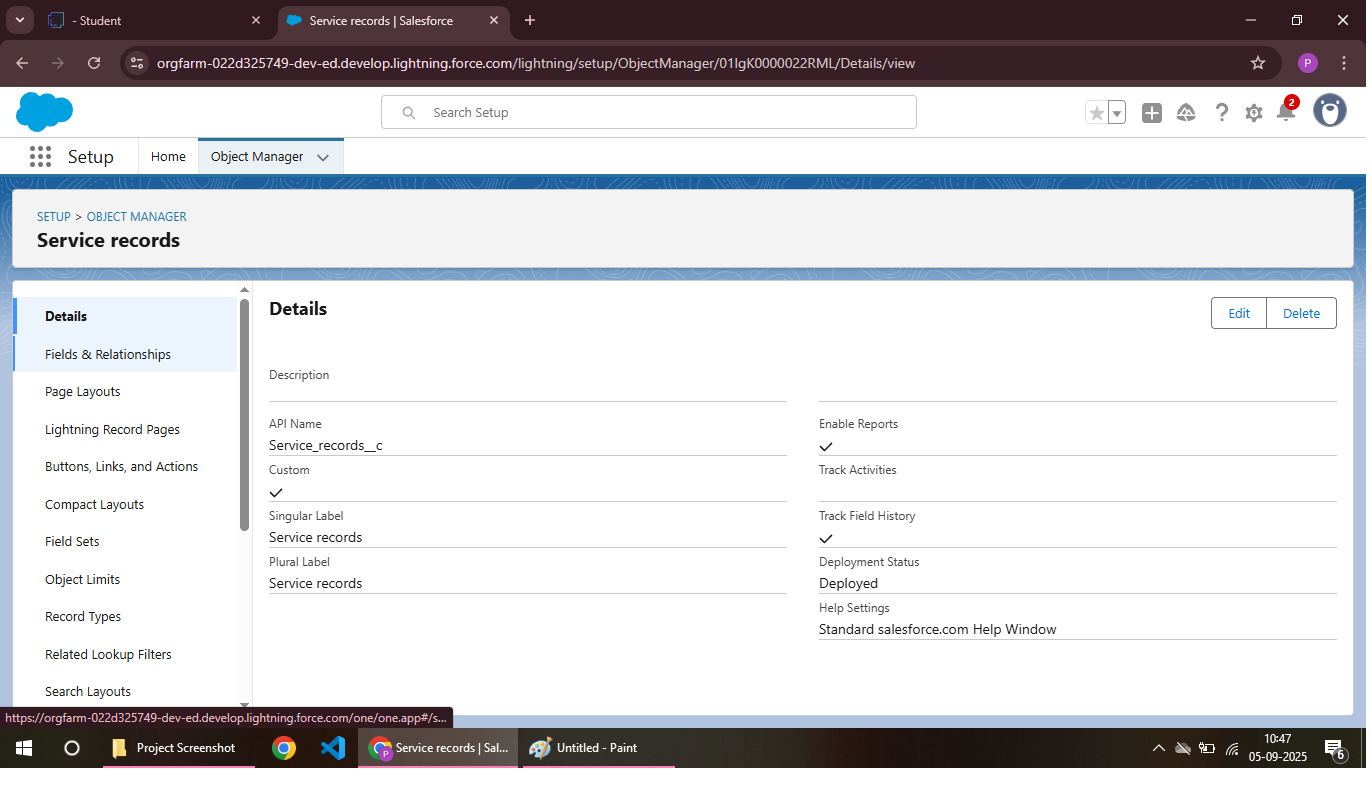


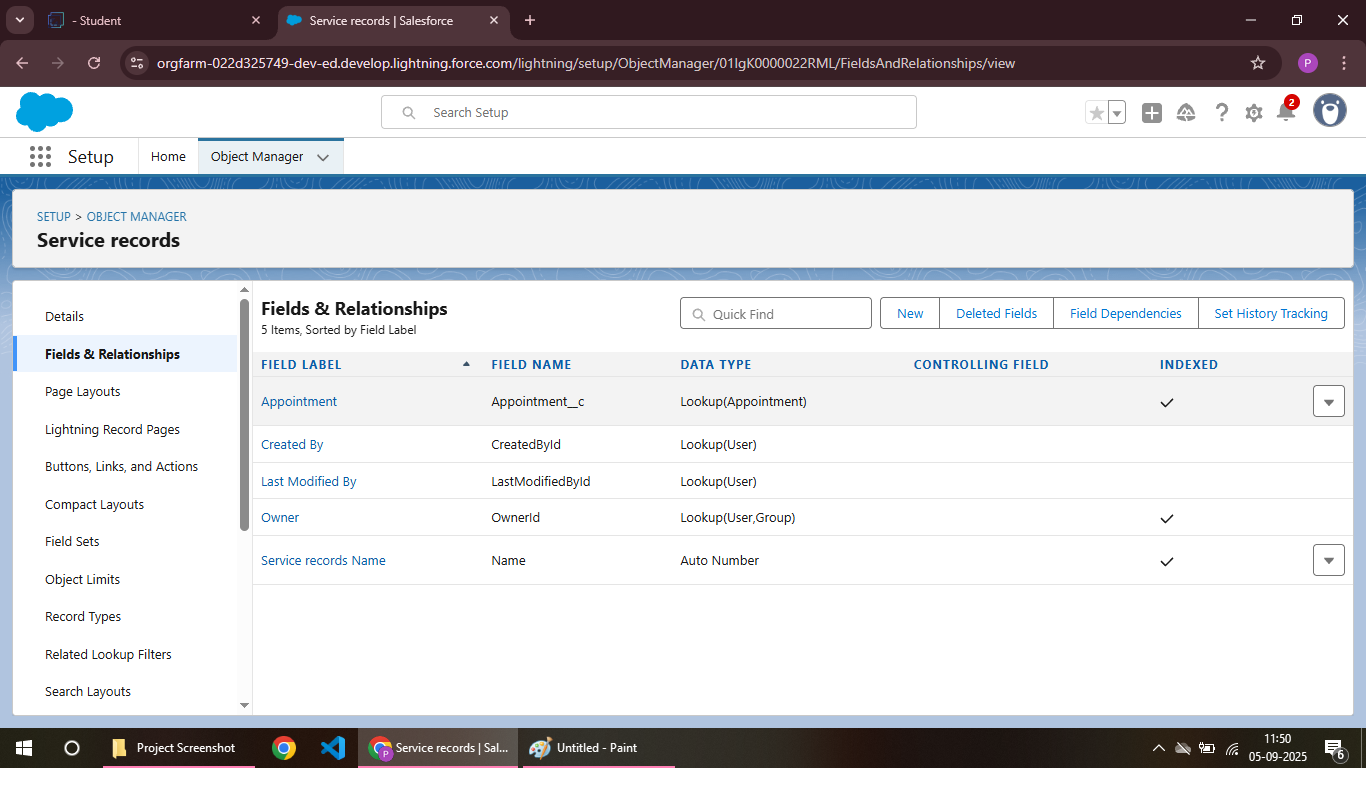
Create Appointment Object:



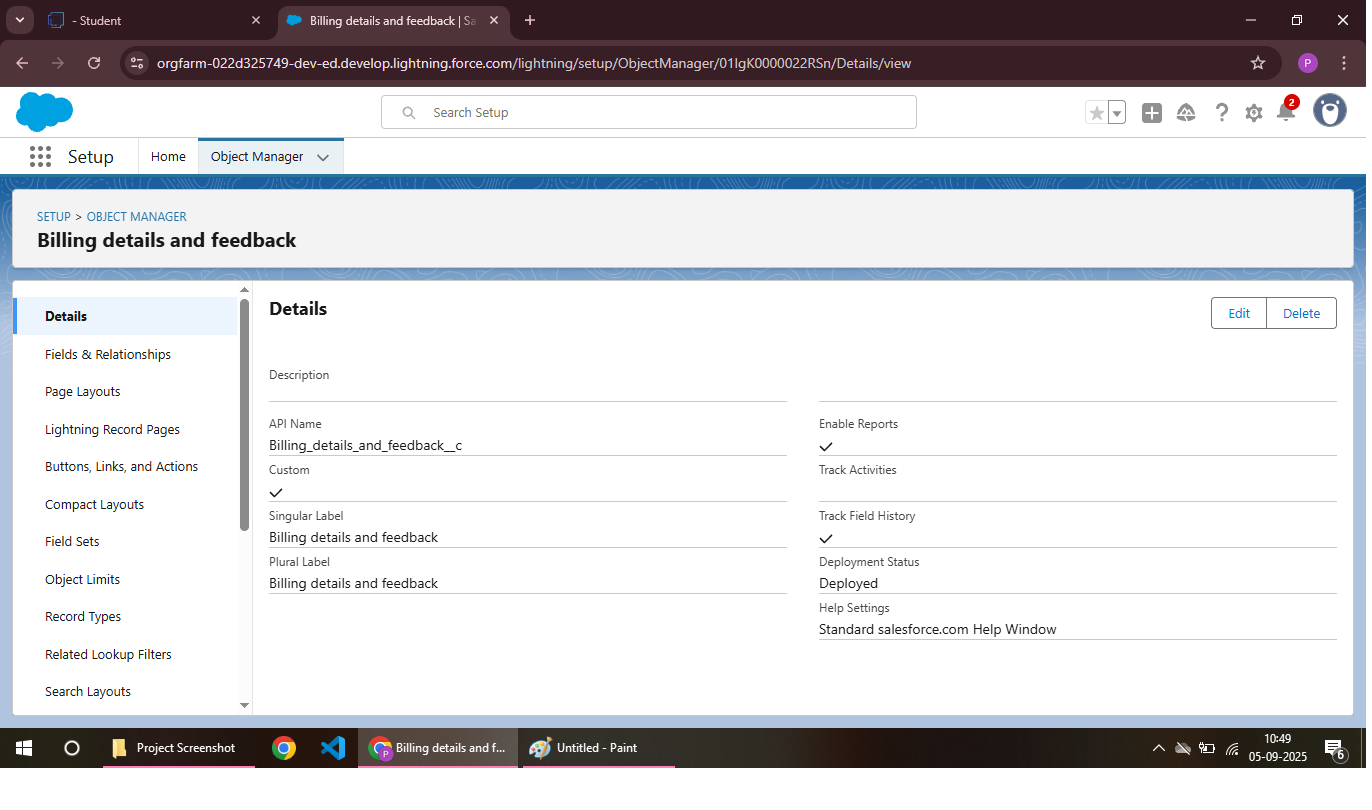


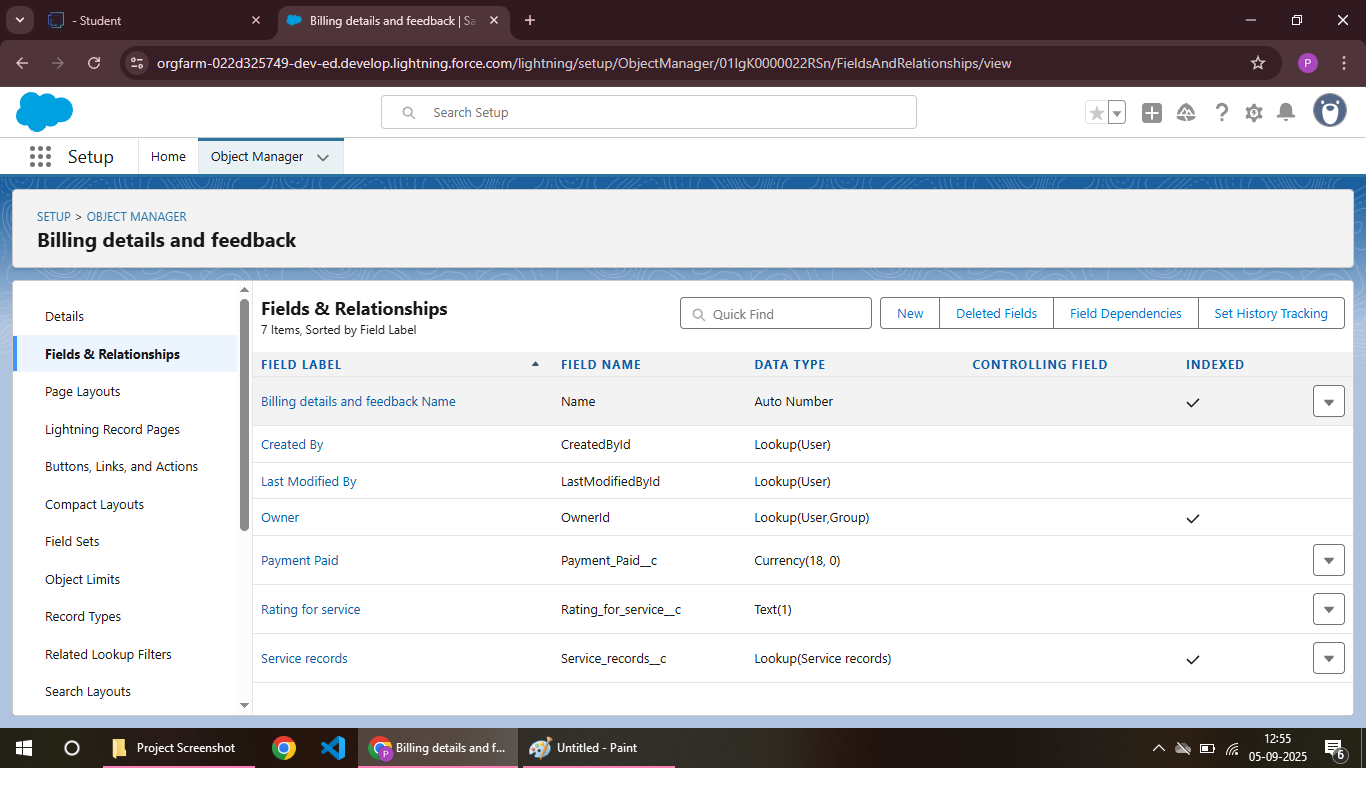
Create Service Record Object:





Create Billing Details And Feedback Object:

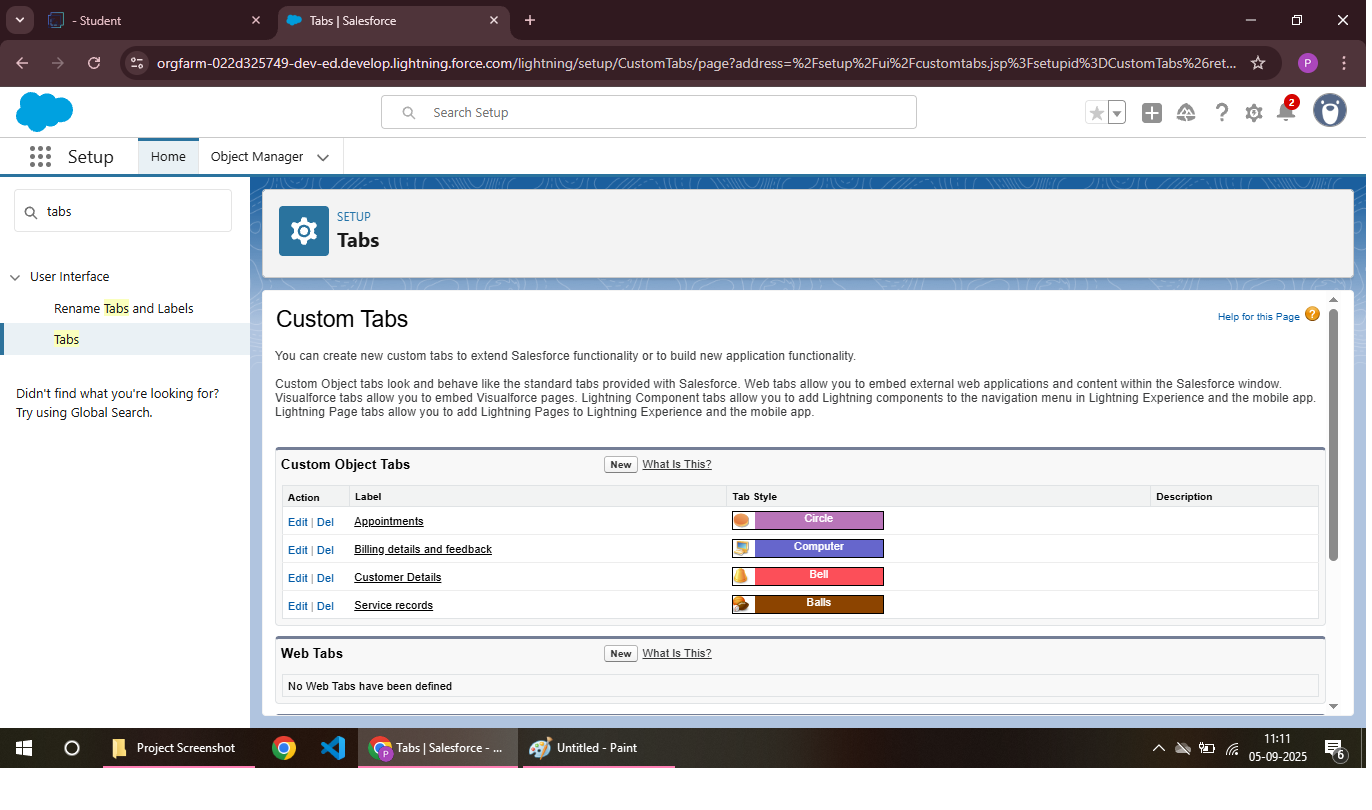




**Tabs:**

A Garage Management System (GMS) typically includes tabs for \*Appointments\*, \*Customers\*, \*Vehicles\*, and \*Inventory\*. These tabs organize system features and functions for easy access.

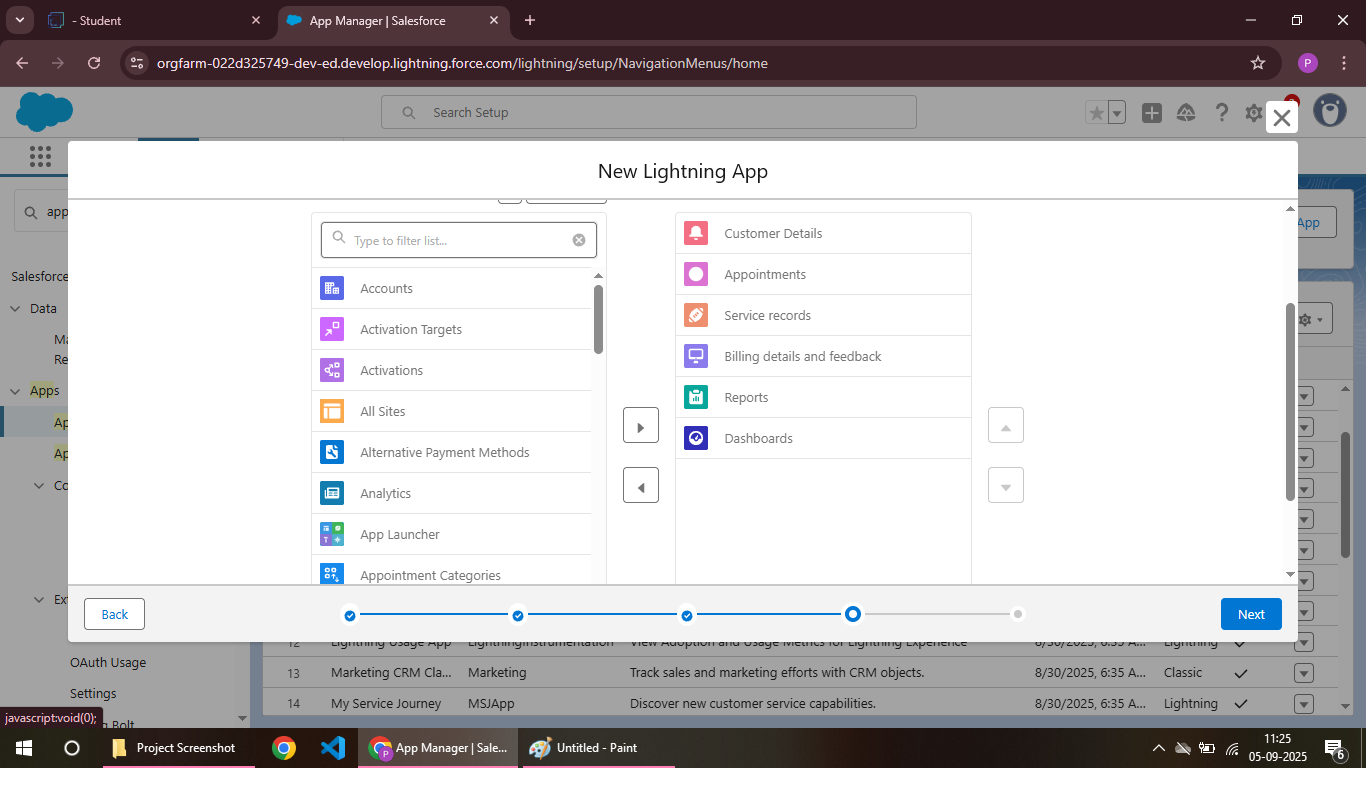
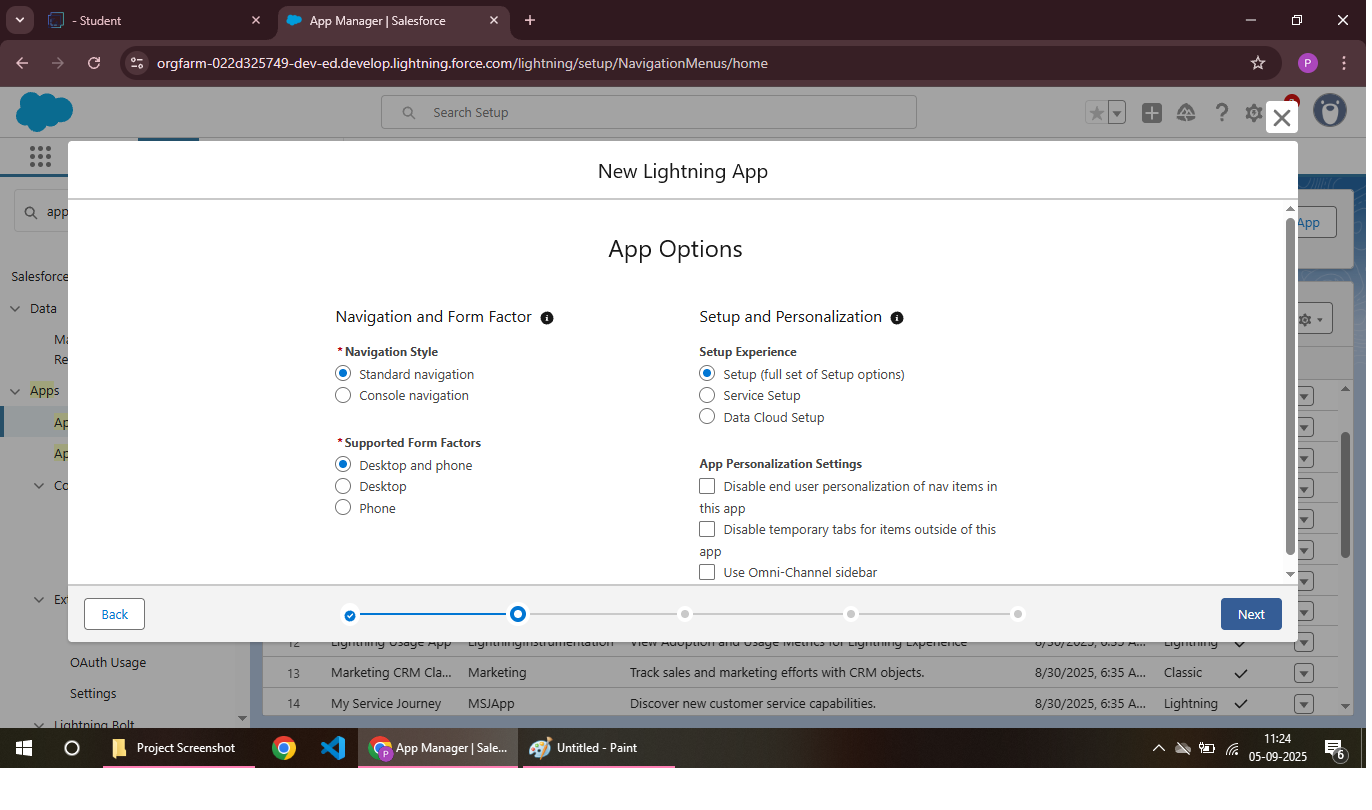
Creating A Custom Tabs:

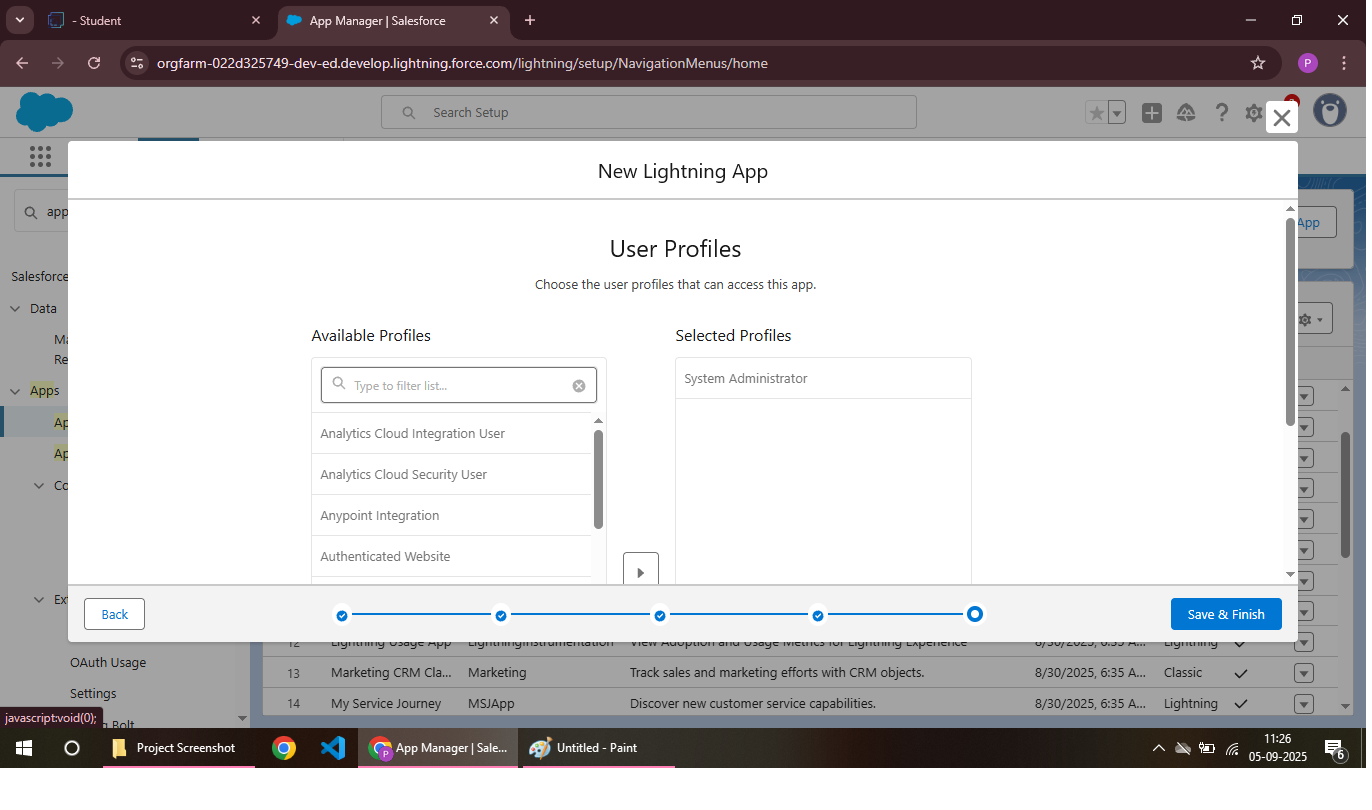


**The Lighting App:**

A Garage Management System in Salesforce Lightning App is a customized application built on Salesforce’s Lightning Platform to manage all garage operations such as vehicle servicing, customer management, spare parts, and invoicing.

Create A Lighting App:





**Fields:**

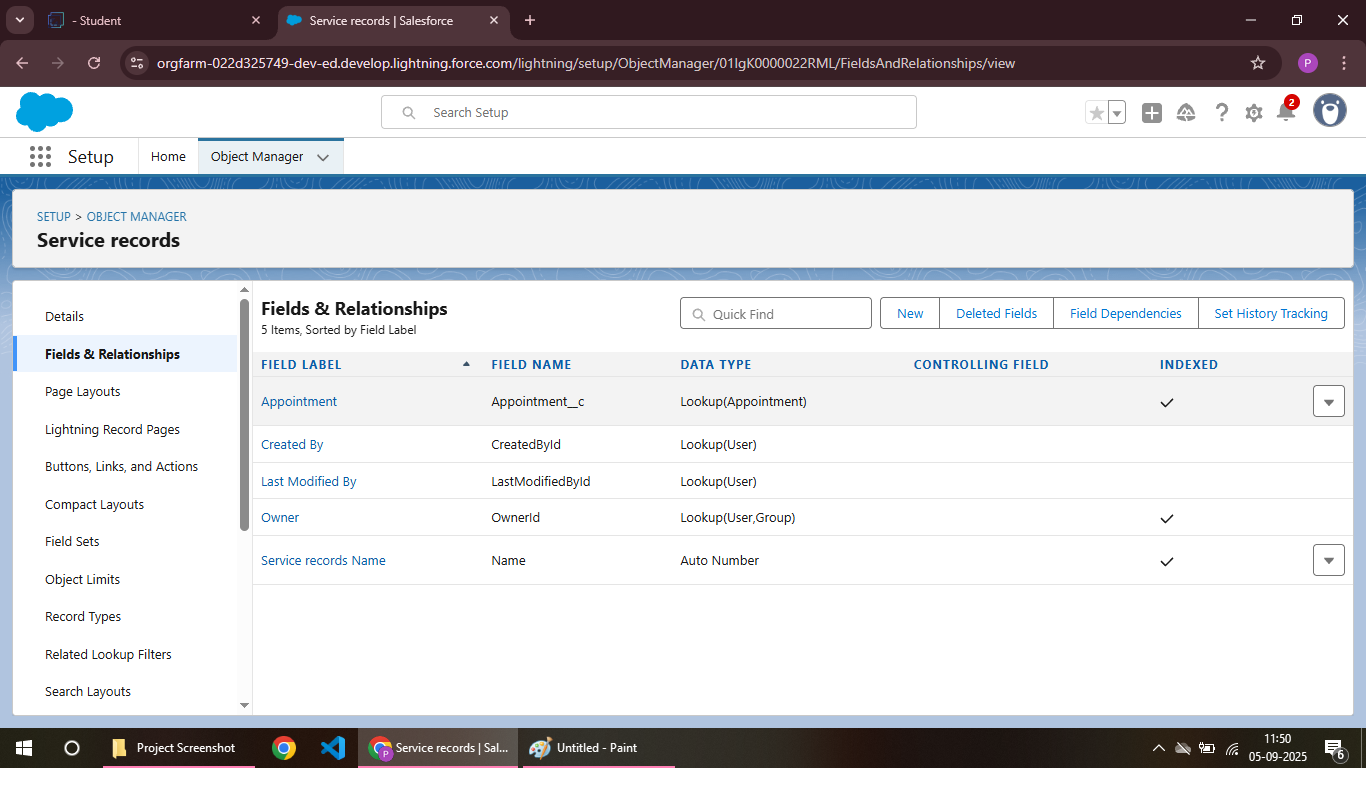
1. \*Vehicle Information\*: Vehicle ID, customer name, vehicle type, license plate, etc.

2. \*Service Details\*: Service type, cost, description, etc.

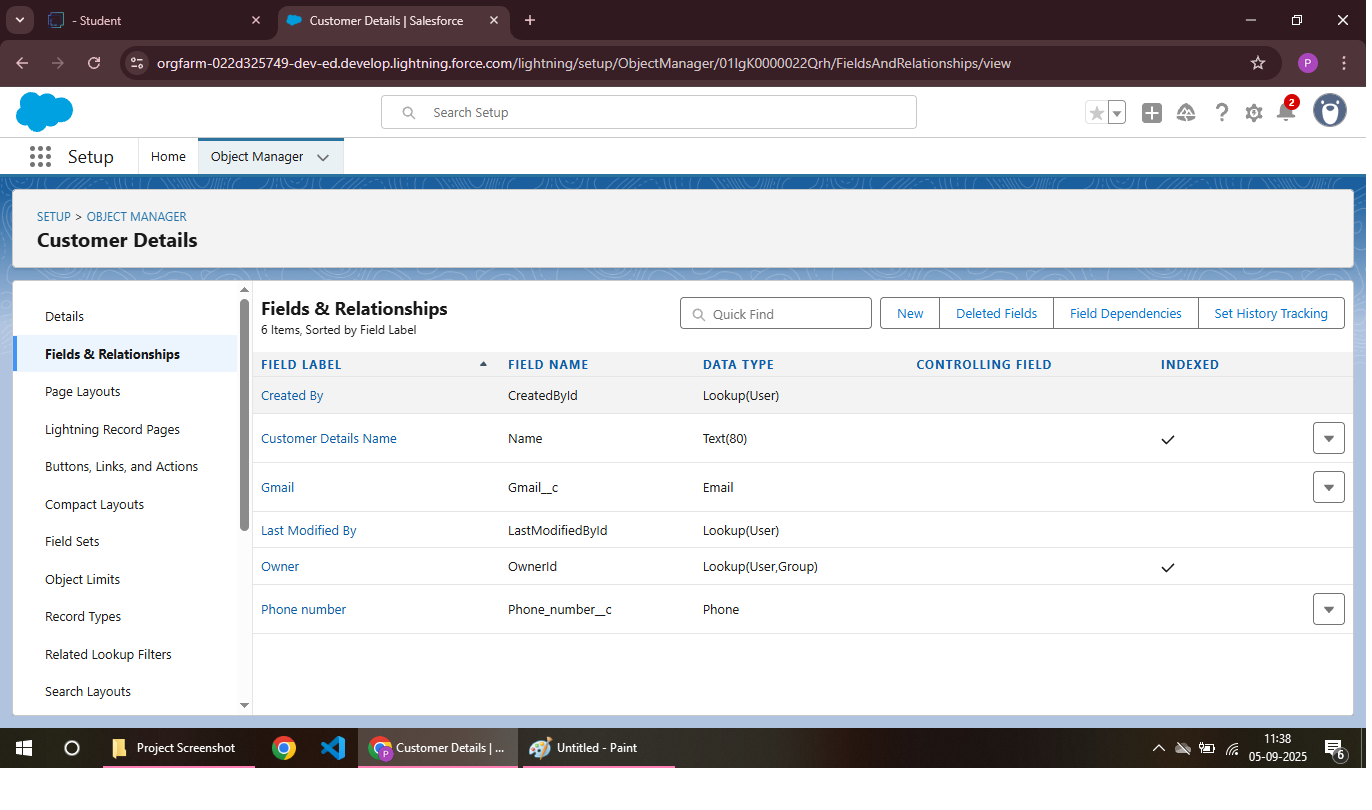
3. \*Customer Information\*: Customer name, contact details, service history, etc.

4. \*Payment and Billing\*: Invoice numbers, payment status, amount, etc.

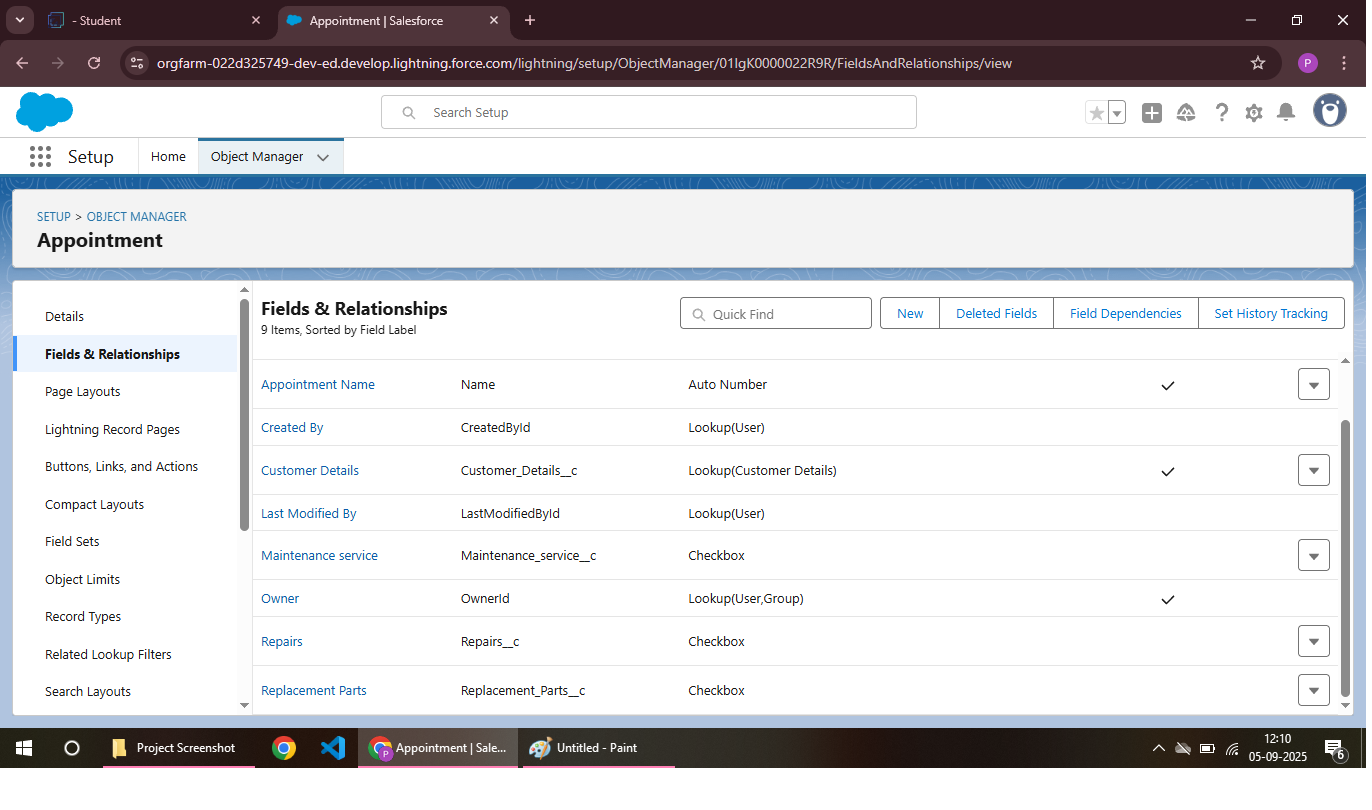
Creation Of Fields For The Customer Details Object:



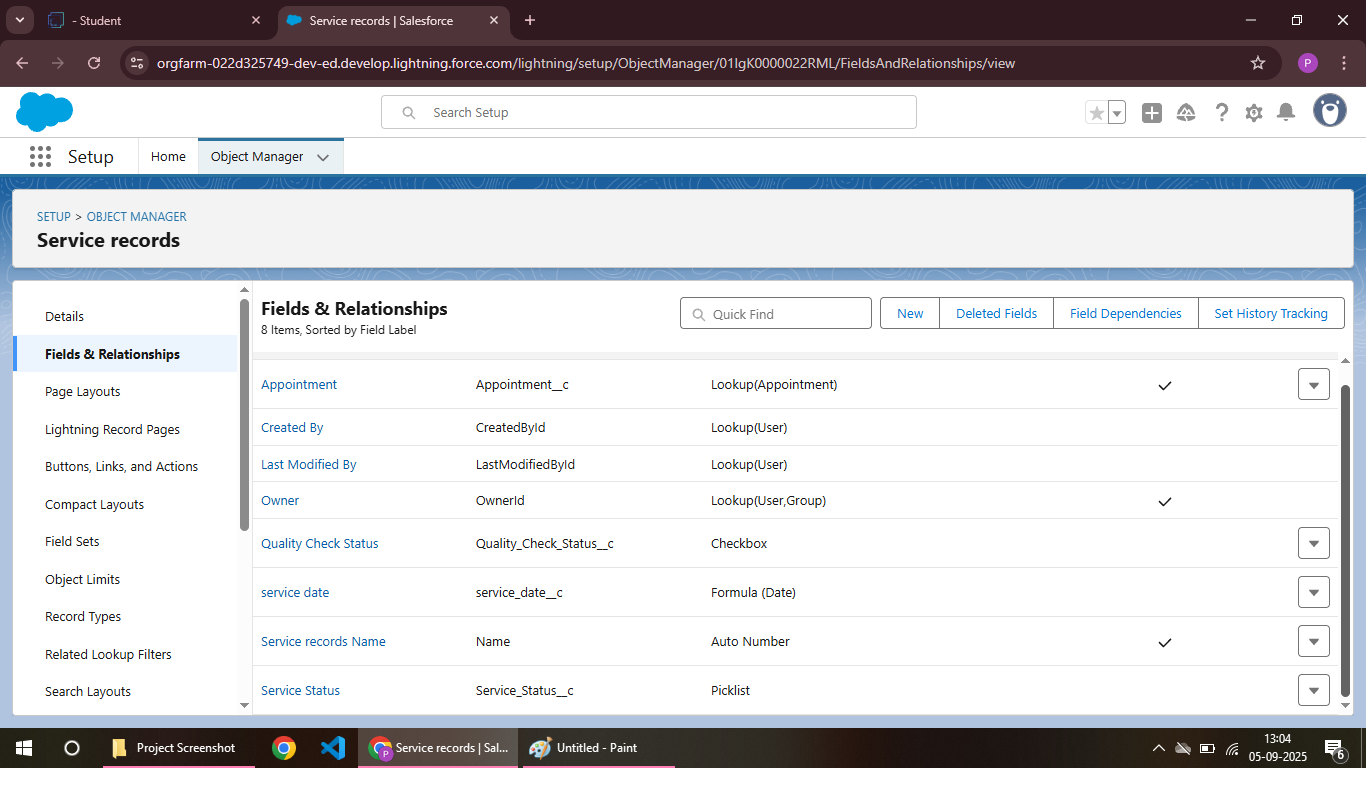
Creation Of Lookup Fields:



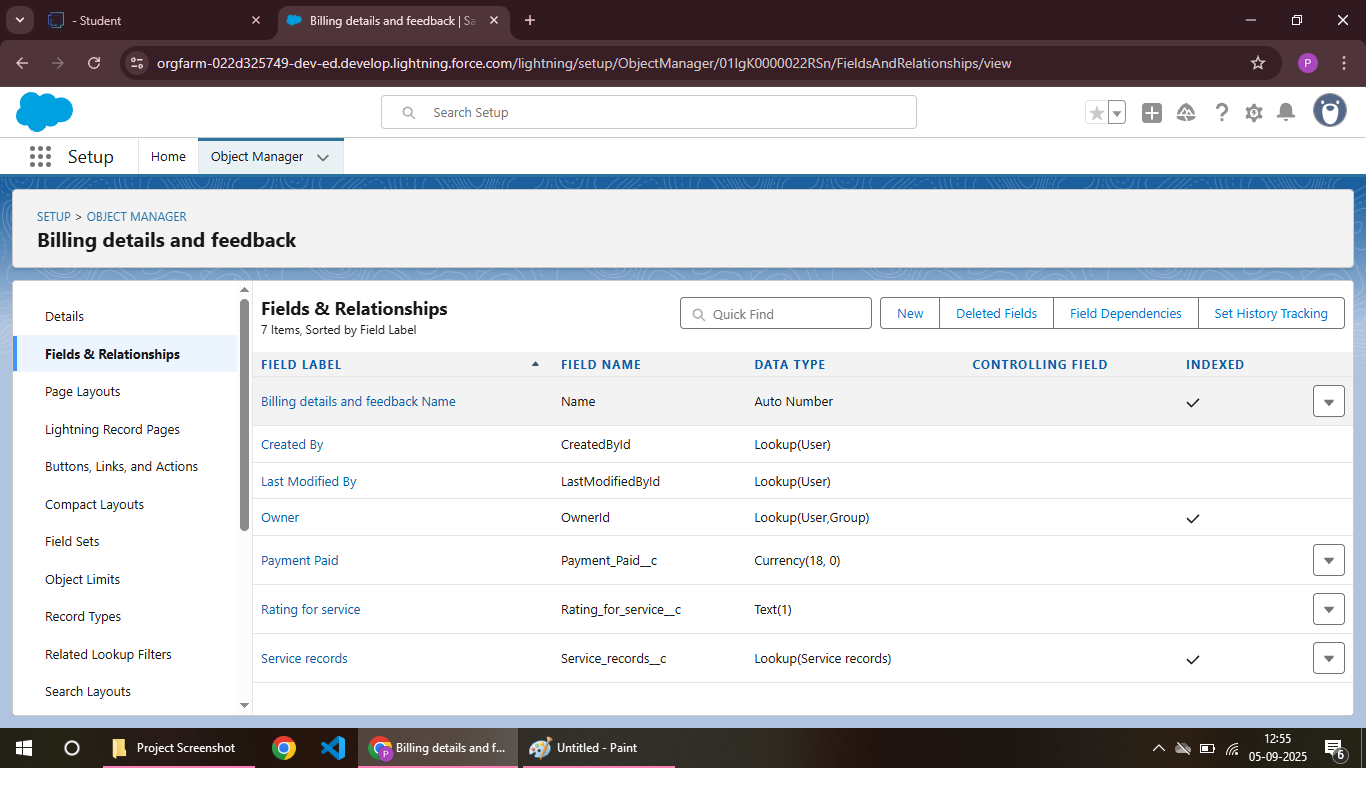
Creation Of Checkbox Fields:



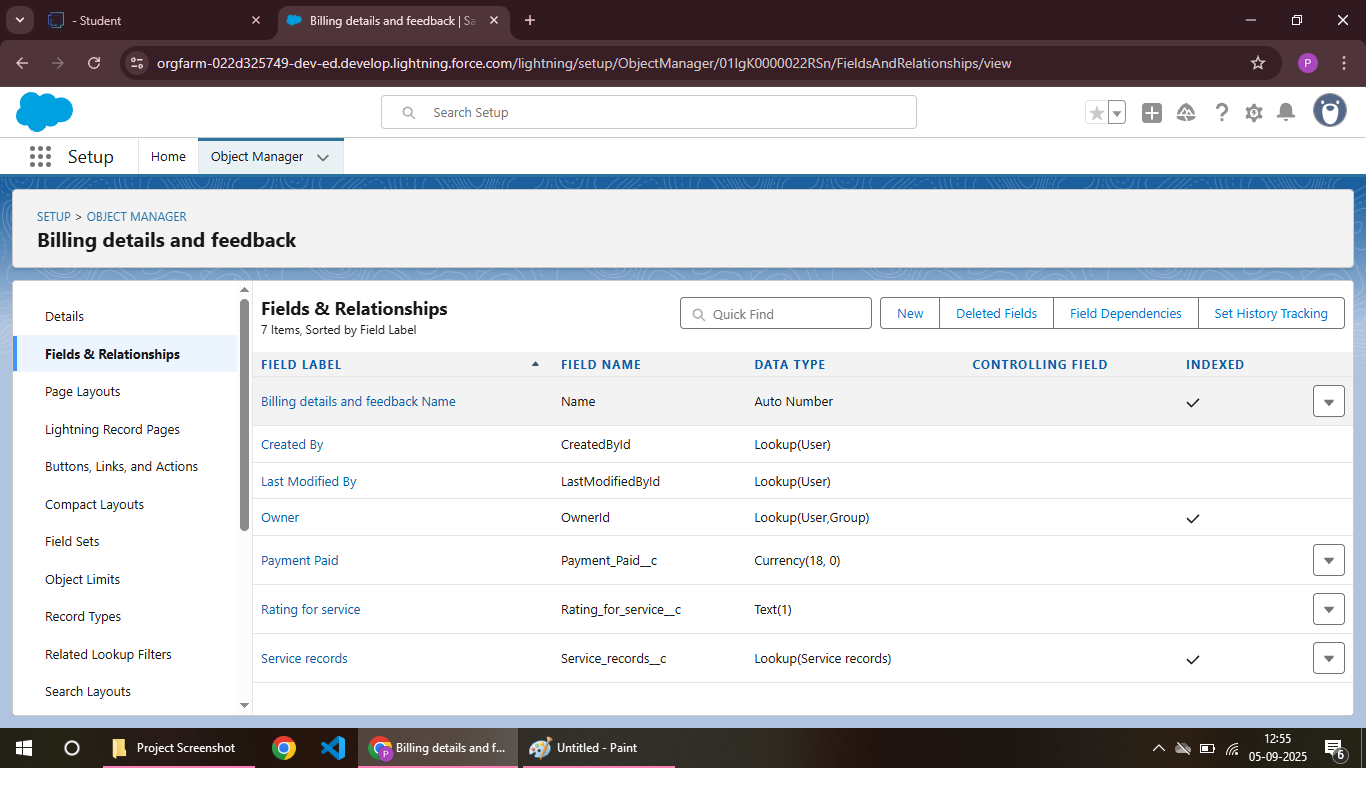
Creation Of Date Fields:



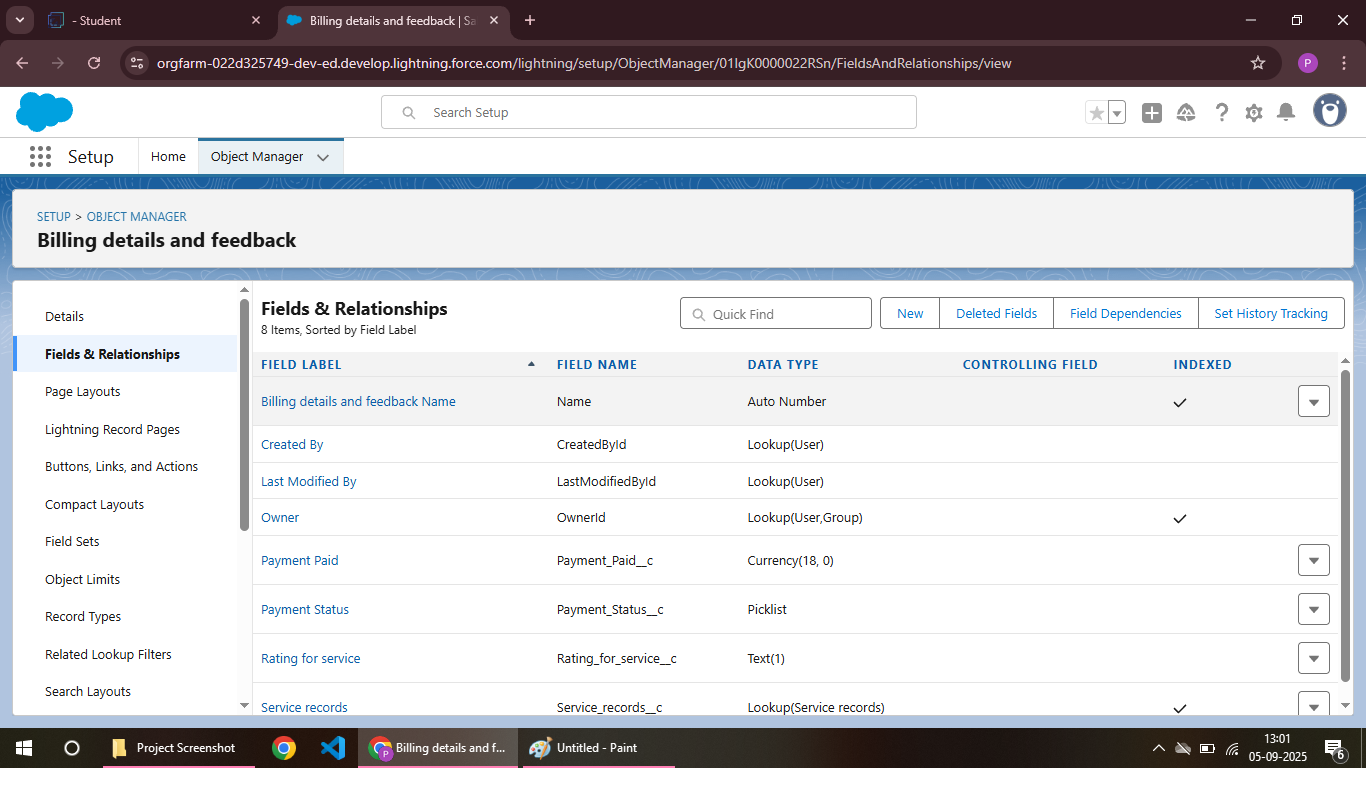
Creation Of Currency Fields:



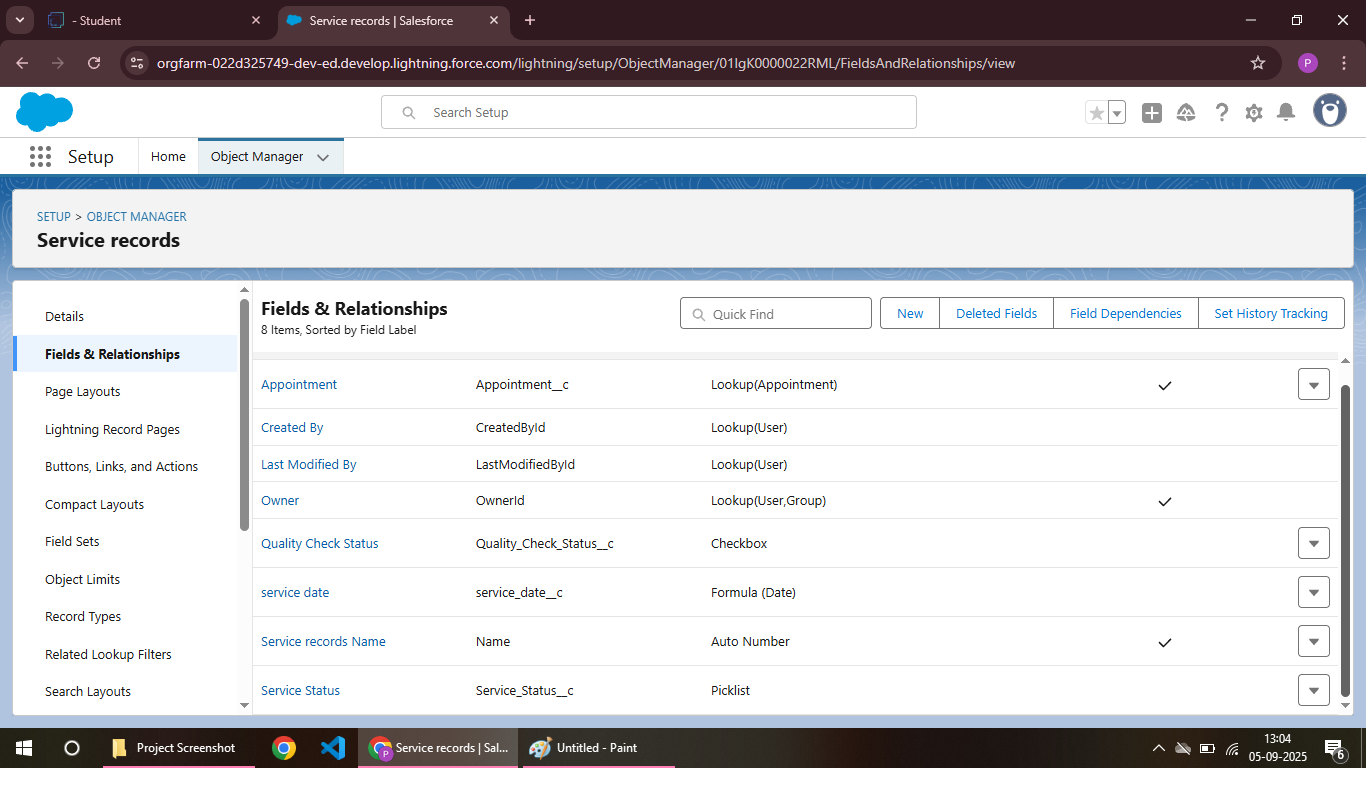
Creation Of Text Fields:



Creation Of Picklist Fields:



Creating Formula Field In Service Record Object:



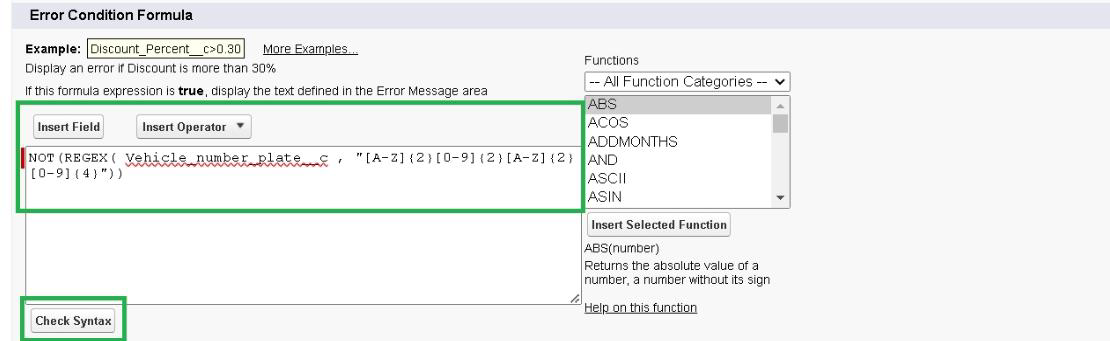
**Validation Rules:**

A Garage Management System (GMS) includes validation rules to ensure data accuracy and integrity, such as \*Vehicle ID format\*, \*Service Date\*, and \*Payment Amount\* validation. These rules prevent errors and inconsistencies in the system.

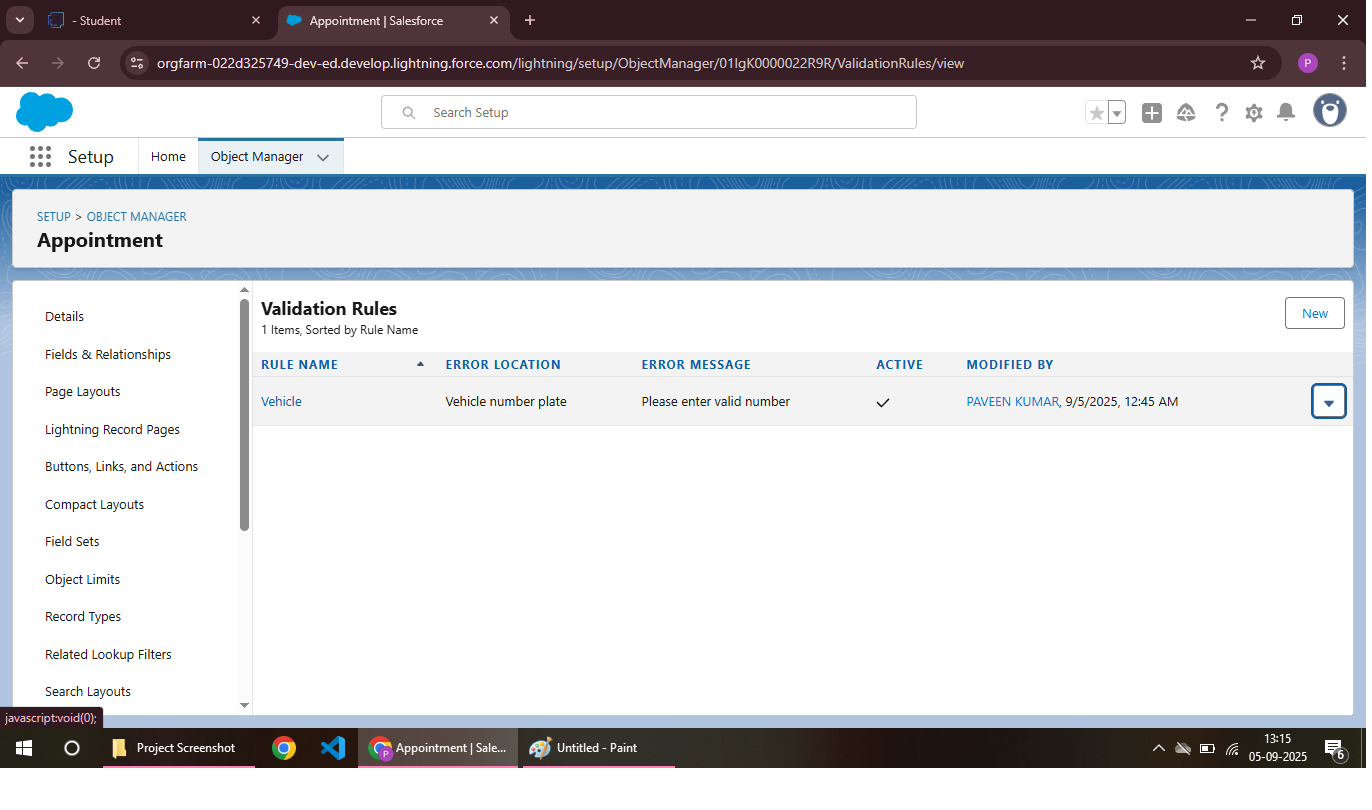
Rules:



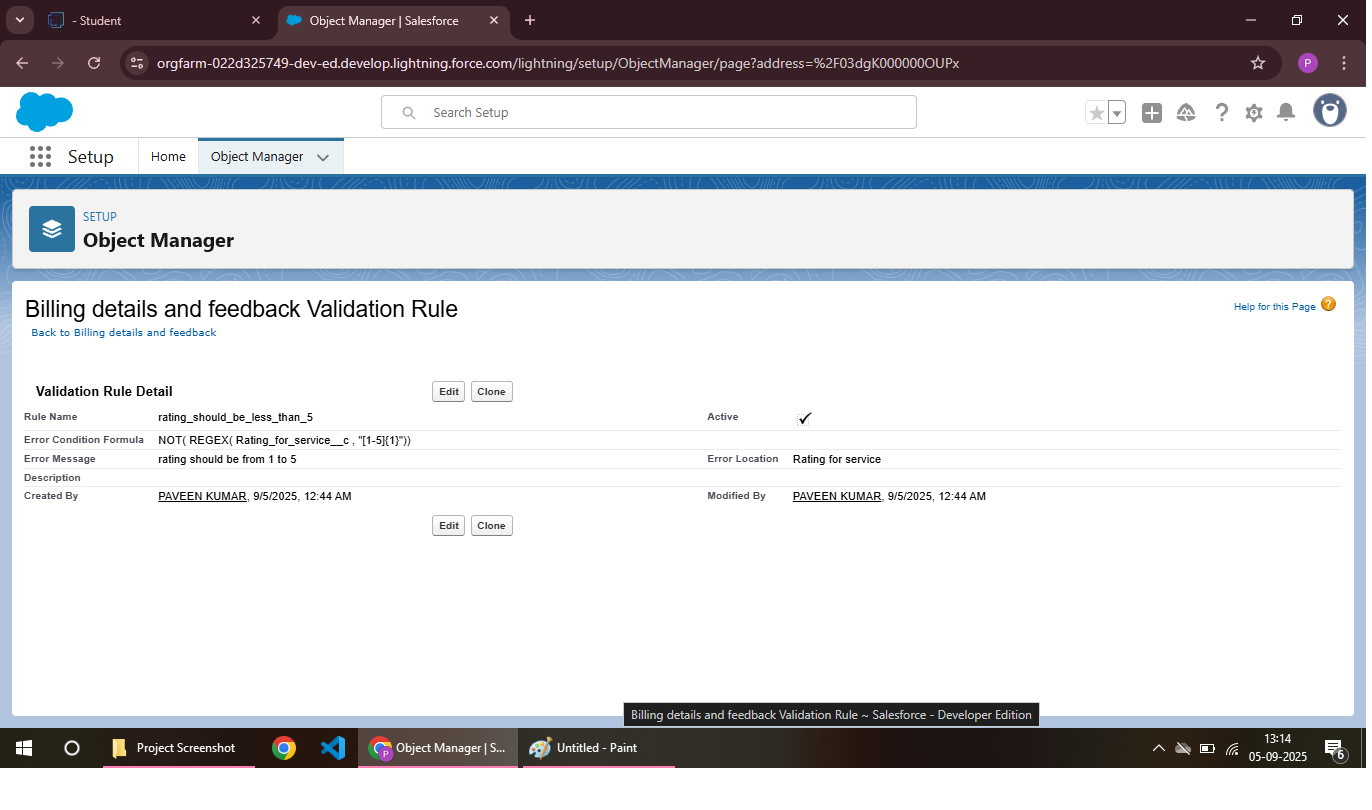
Condition As Formula:



To Create A Validation Rules To An Appointment Object:



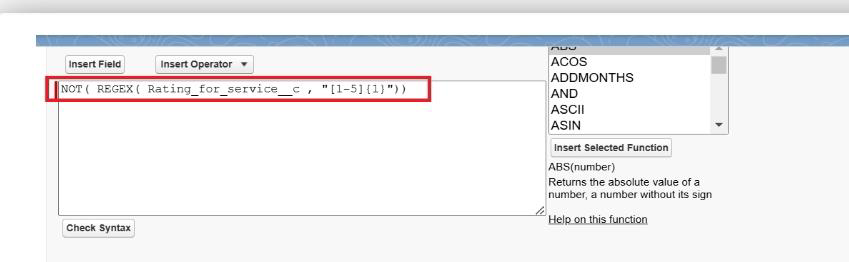
To Create A Validation Rule To An Billing Details And Feedback Object:



Rule Edit:



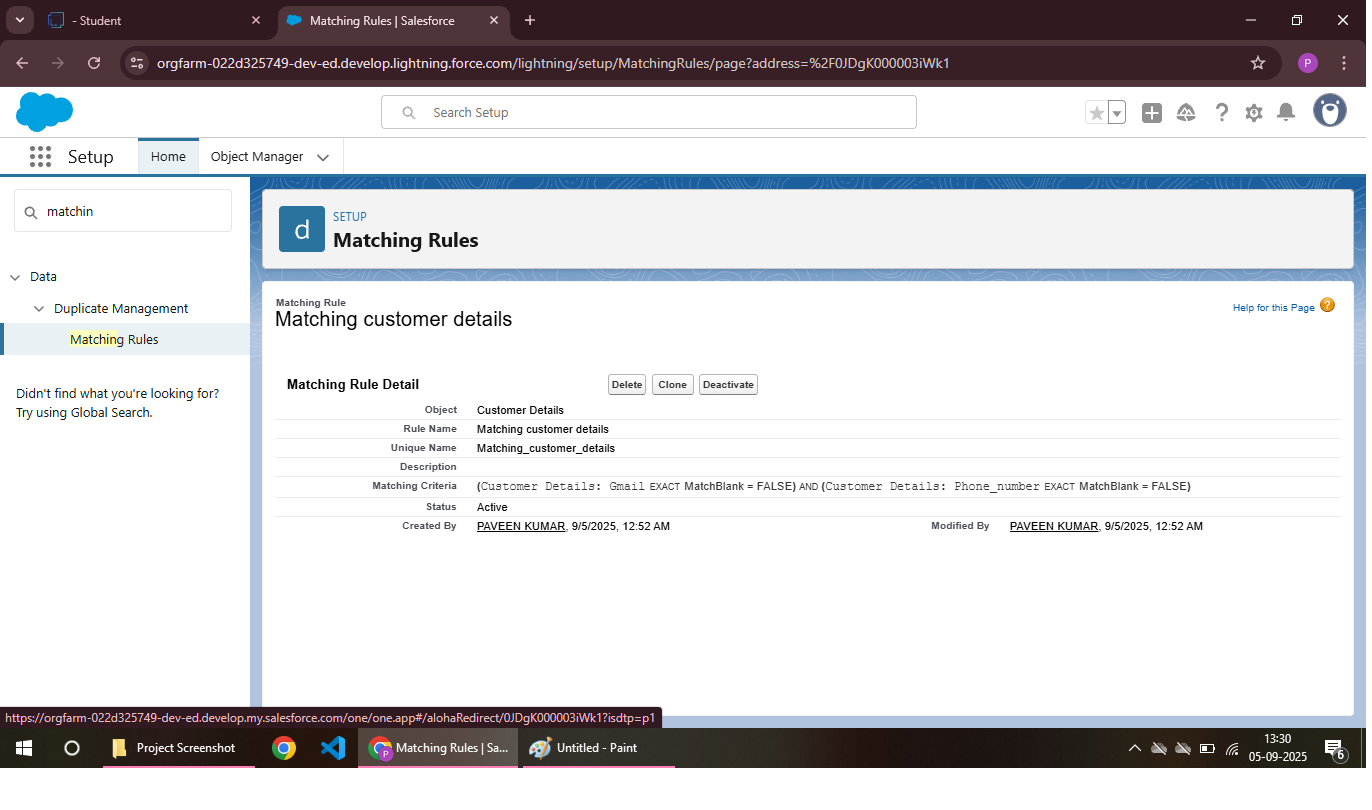
Condition Formula:



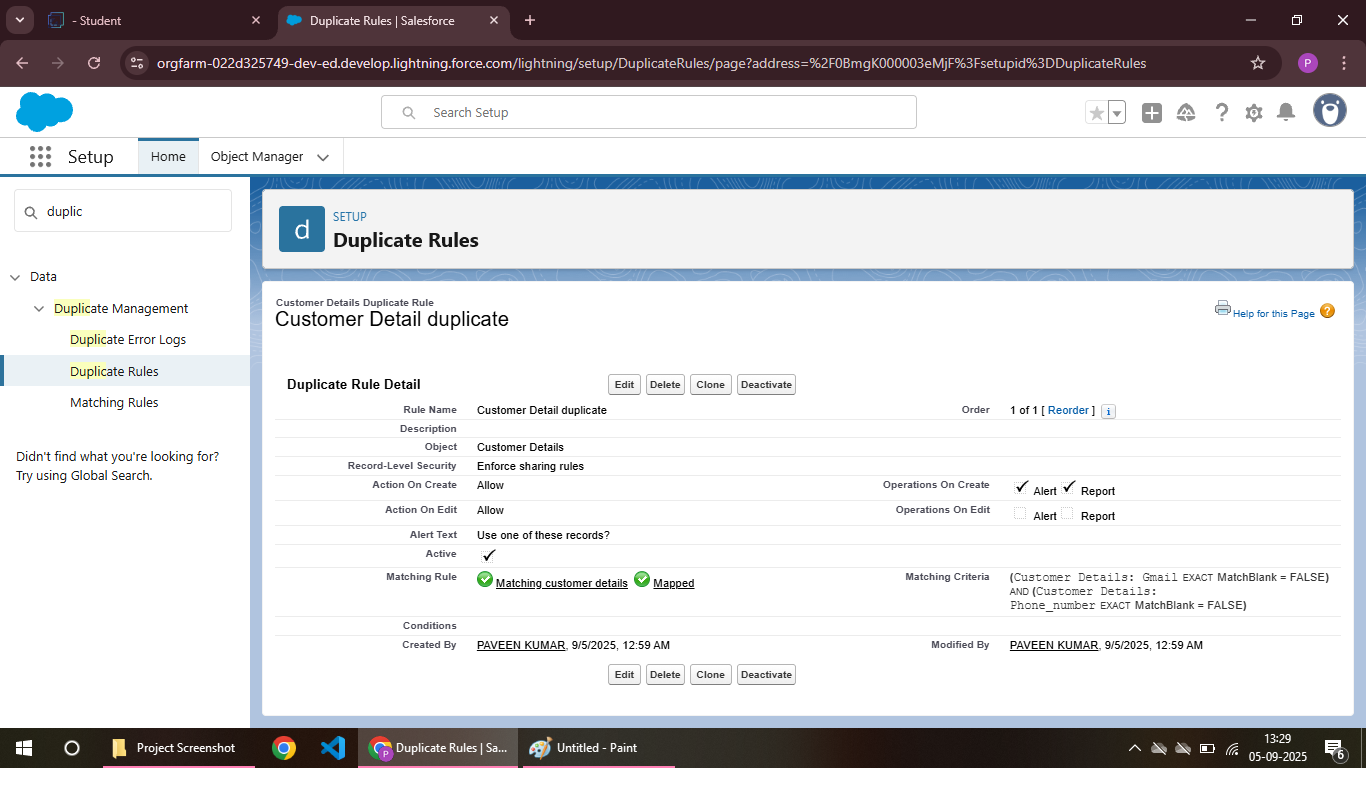
**Duplicate Rule :**

A duplicate rule in a Garage Management System (GMS) prevents \*duplicate customer records\* and \*duplicate vehicle records\*, ensuring data accuracy and integrity.

To Create A Matching Rule To An Customer Details Object:



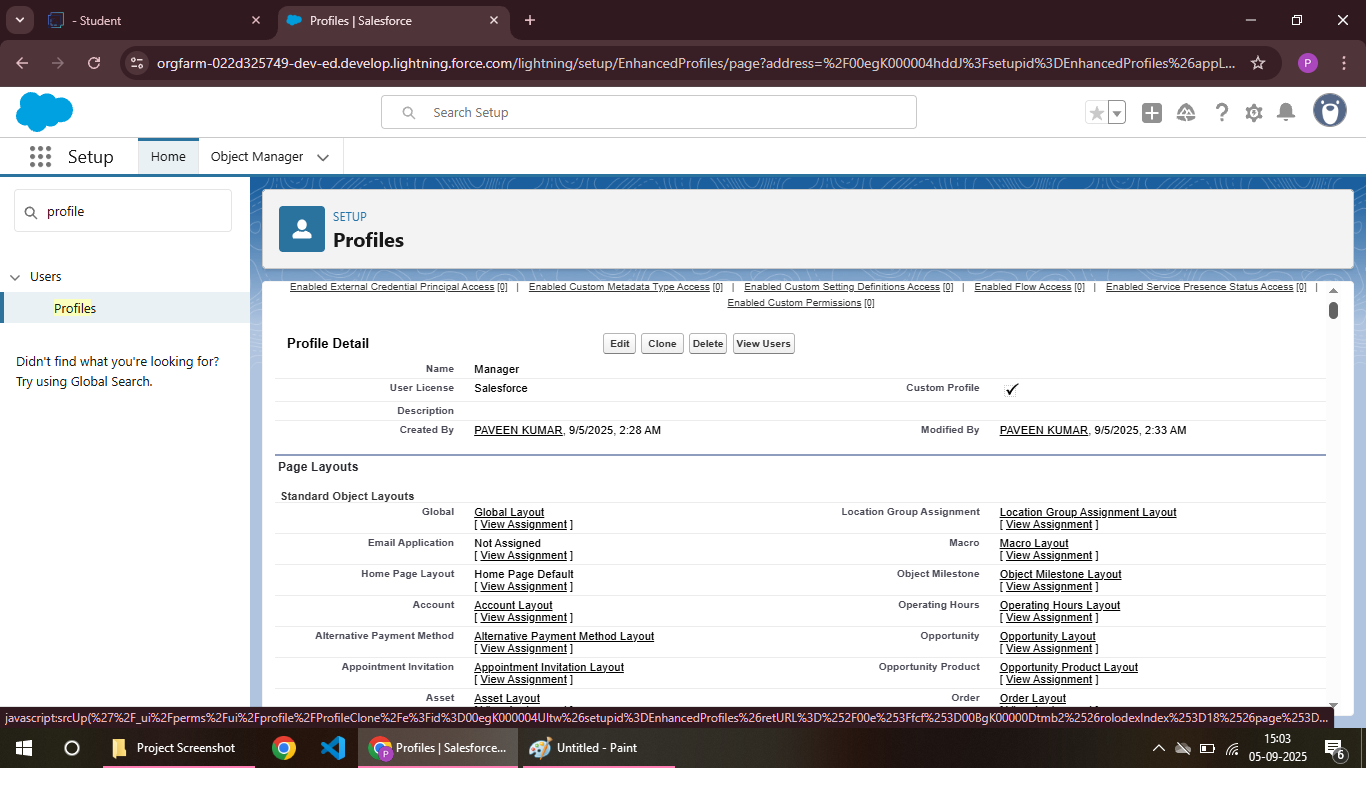
To Create A Duplicate Rule To An Customer Details Object:



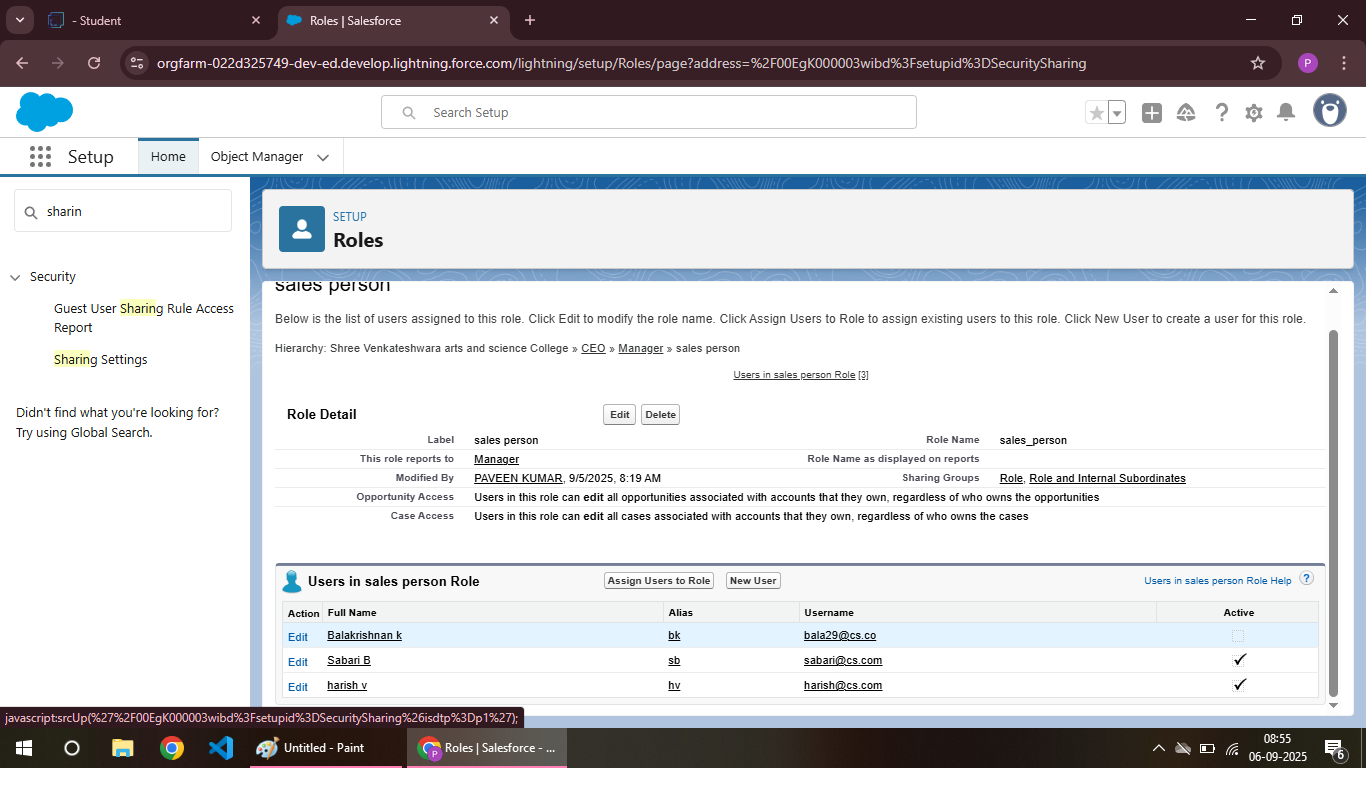
**Profiles:**

A Garage Management System (GMS) profile includes garage information, services offered, and technician details. It provides a comprehensive overview of the garage's operations and capabilities.

Manager Profile:



Sales Person Profile:



**Role & Role Hierarchy :**

In a Garage Management System (GMS), roles and role hierarchy may include:

1. \*Admin\*: Full access to system settings, reports, and management.

2. \*Manager\*: Oversees operations, manages staff, and reviews performance.

3. \*Technician\*: Performs vehicle services and repairs.

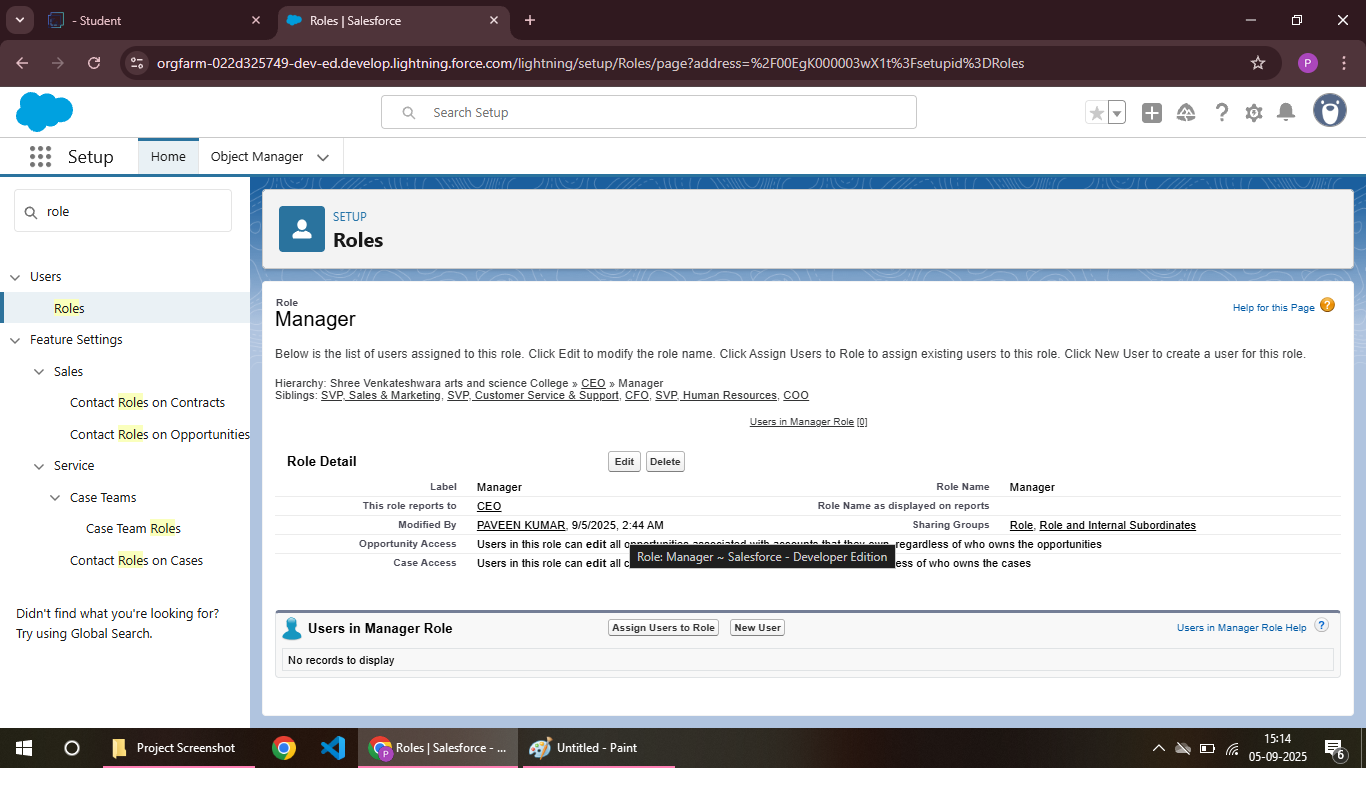
4. \*Receptionist\*: Handles customer inquiries, scheduling, and billing.

5. \*Customer\*: Views service history, schedules appointments**.**

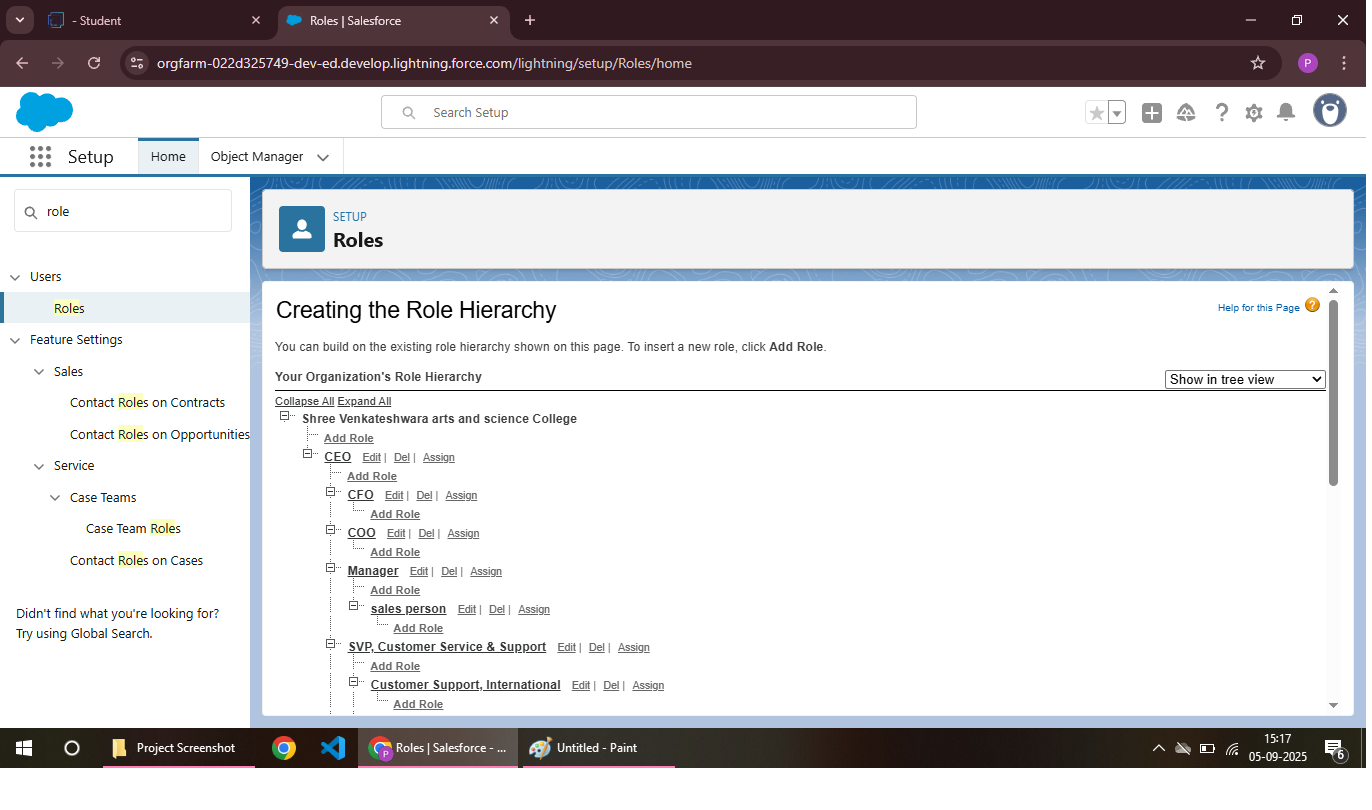
**Role hierarchy:**

**1. \***Admin\* > \*Manager\* > \*Receptionist\* > \*Technician\* > \*Customer\*

Creating Manager Role:

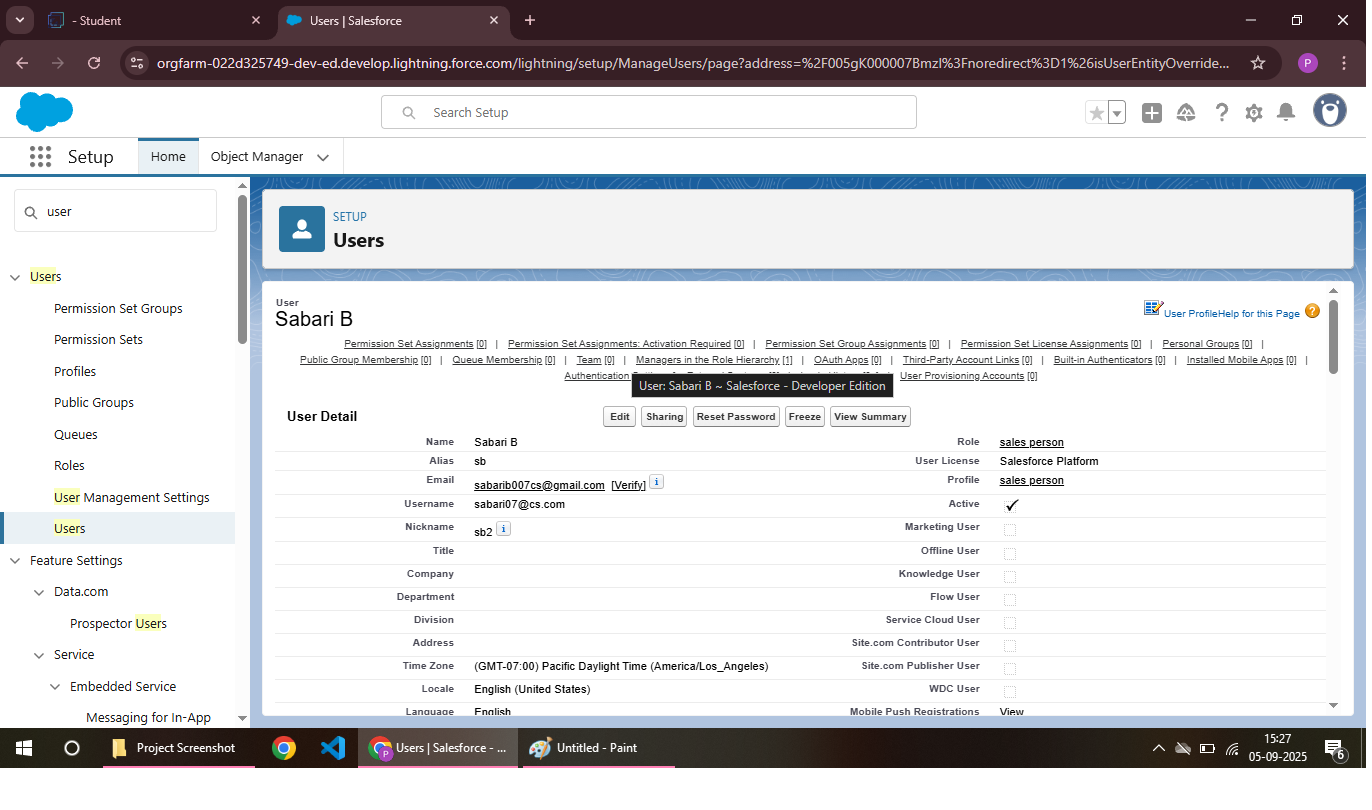


Creating Another Role:



**Users :**

Create User:

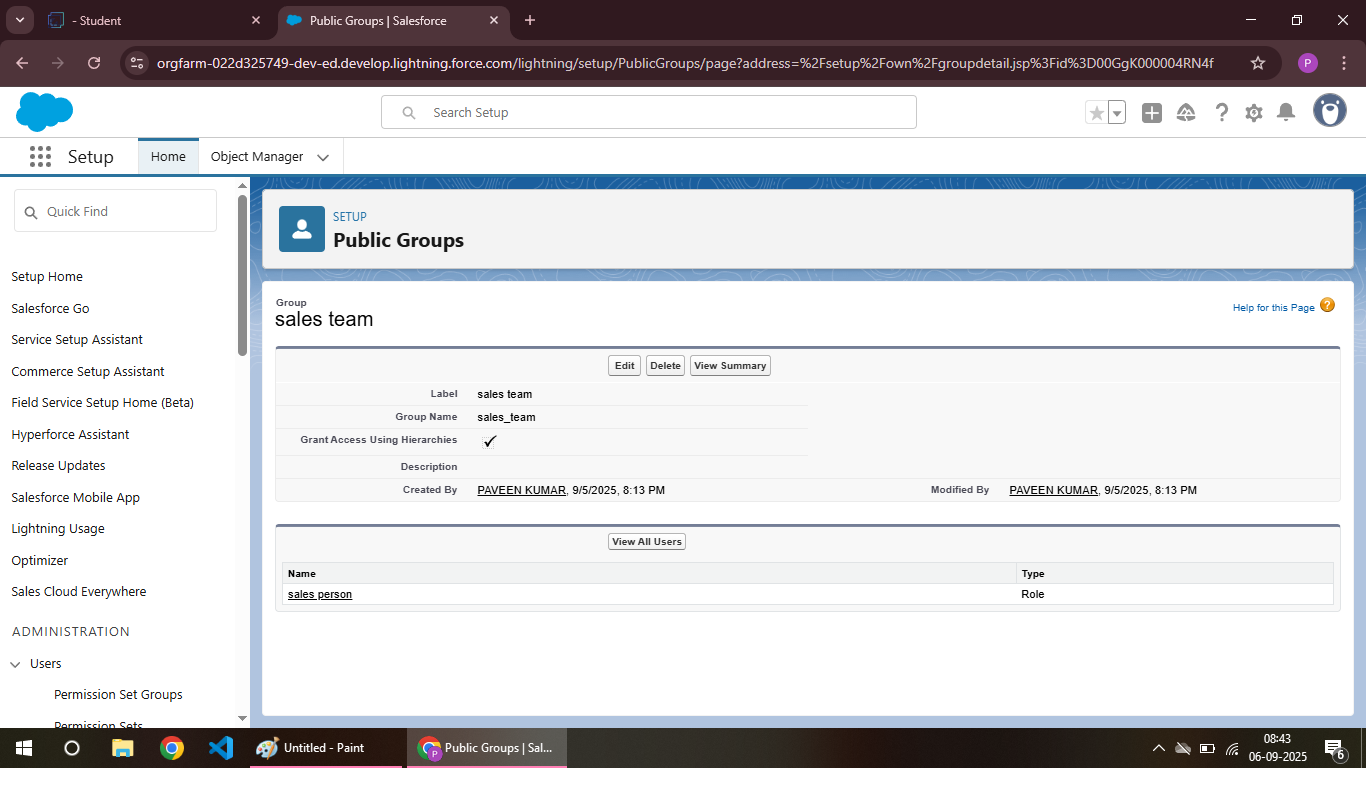


Creating Another User:

**Public Groups:**

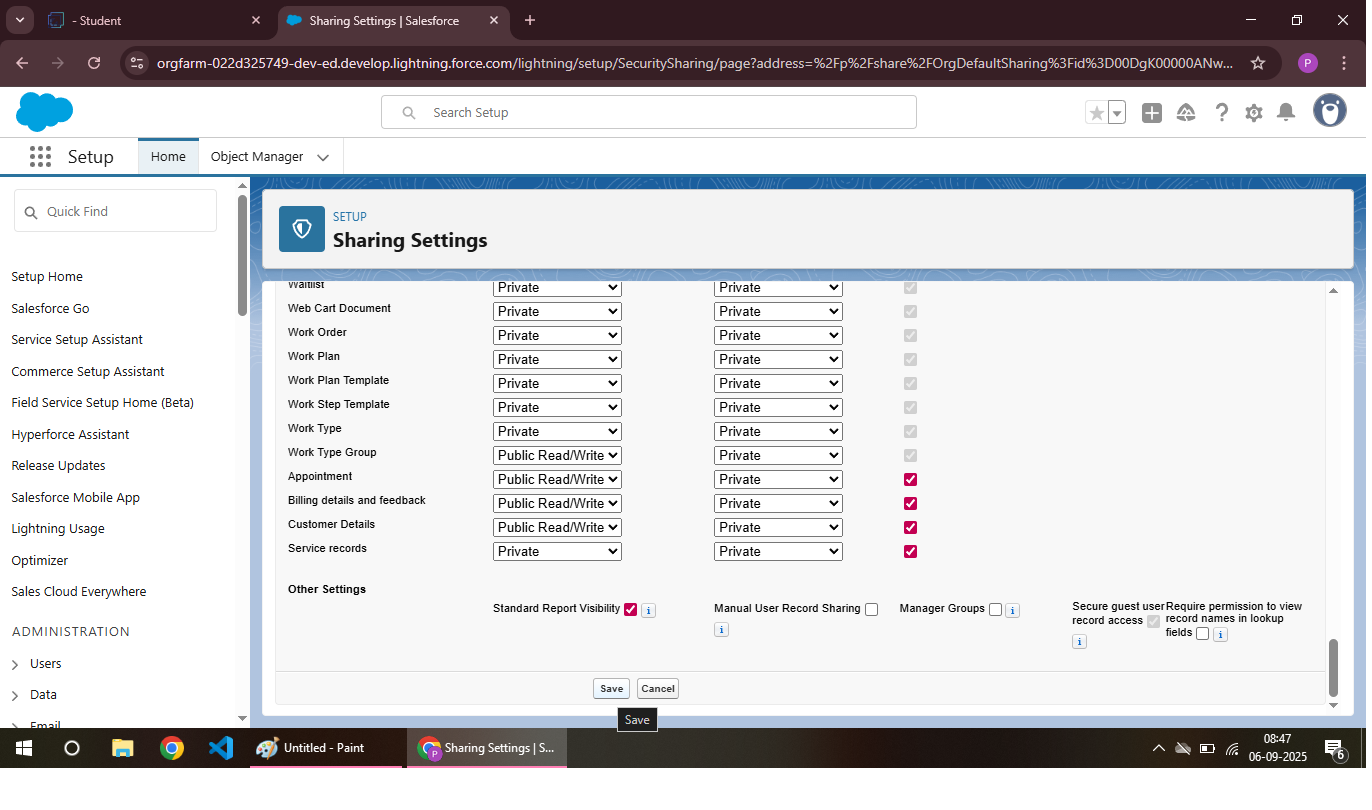
In a Garage Management System (GMS), public groups may include \*Administrators\*, \*Managers\*, \*Technicians\*, and \*Receptionists\*. These groups define access levels and permissions for system users**.**

Creating New PublicGroups:



**Sharing Setting:**

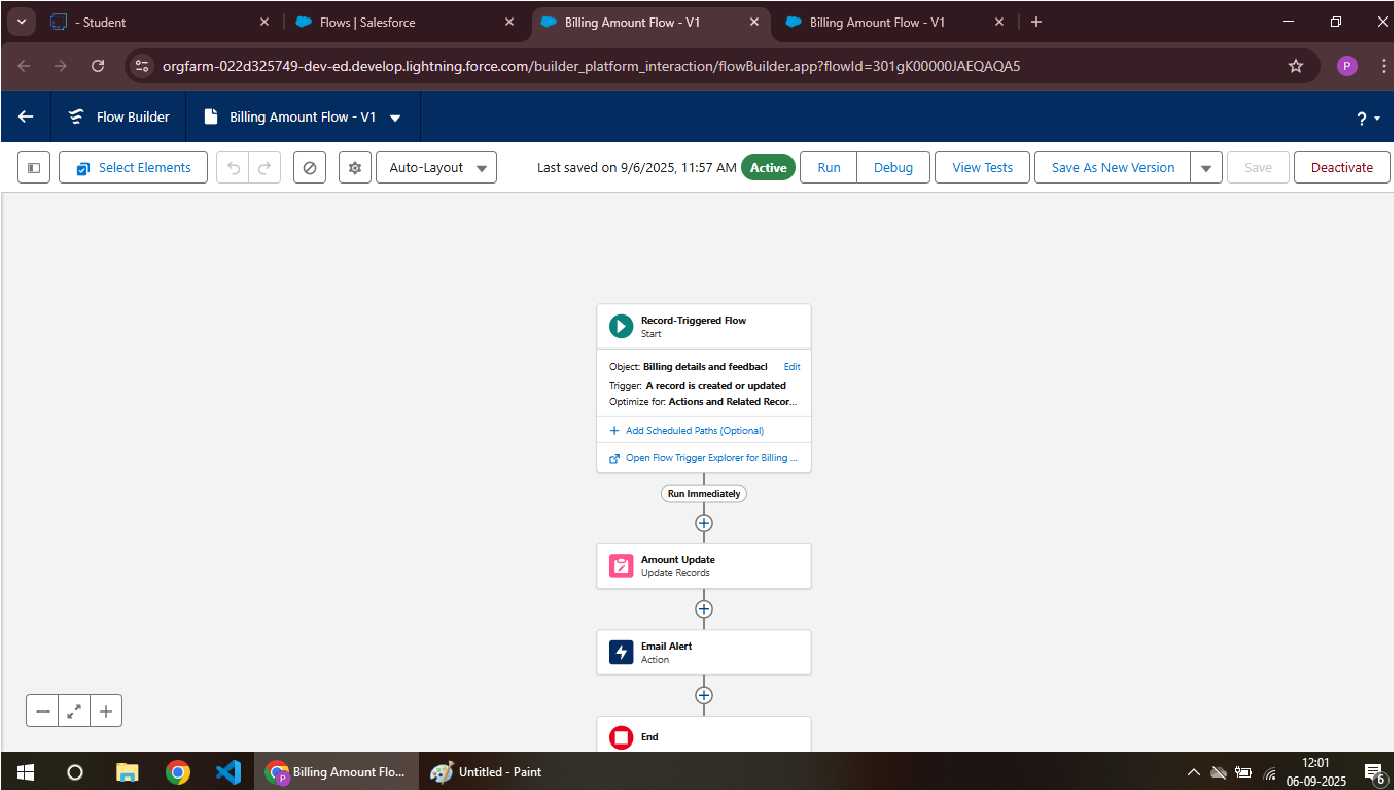
Creating Sharing Settings:



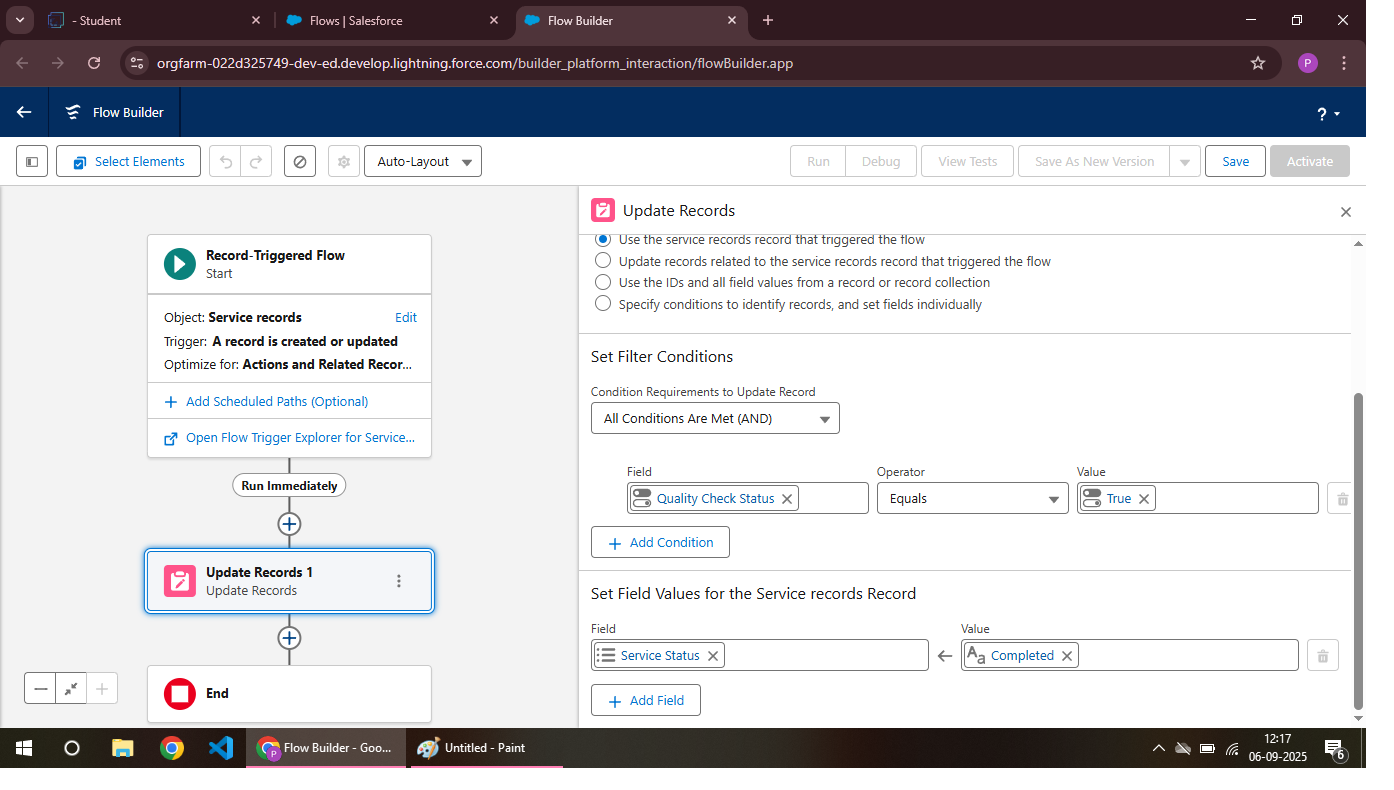
**Flows:**

In a Garage Management System (GMS), the flow typically involves \*Customer Booking\* and \*Service Fulfillment\*, where customers schedule appointments, vehicles are serviced, and payments are processed. This flow streamlines garage operations and enhances customer experience.

Create A Flow:



Create Another Flow:



**Apex Trigger**:

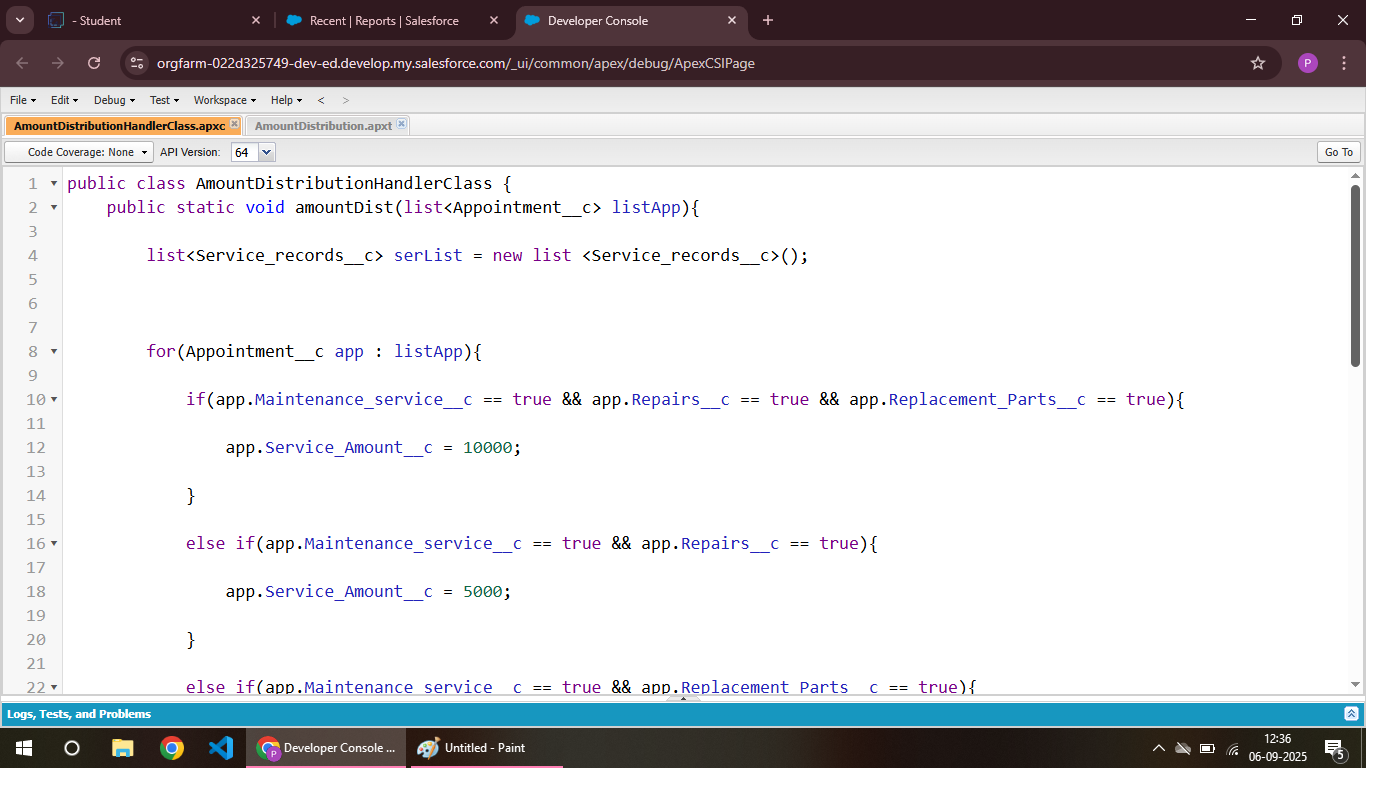
In Salesforce, an Apex trigger for a Garage Management System (GMS) could be used to:

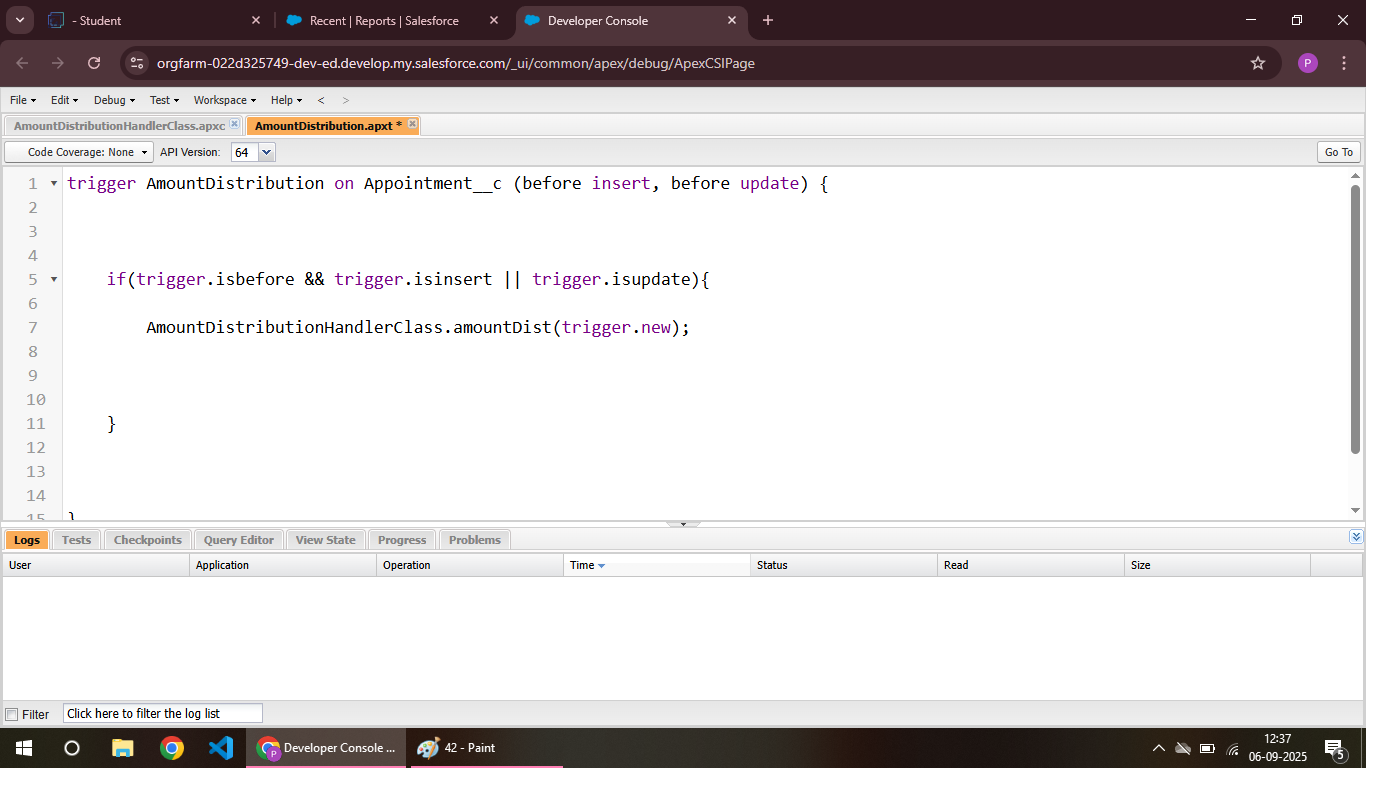
1. \*Automate tasks\*: Trigger actions when records are inserted, updated, or deleted.

2. \*Validate data\*: Ensure data consistency and accuracy.

3. \*Send notifications\*: Notify stakeholders of changes or updates.

Apex Handler:

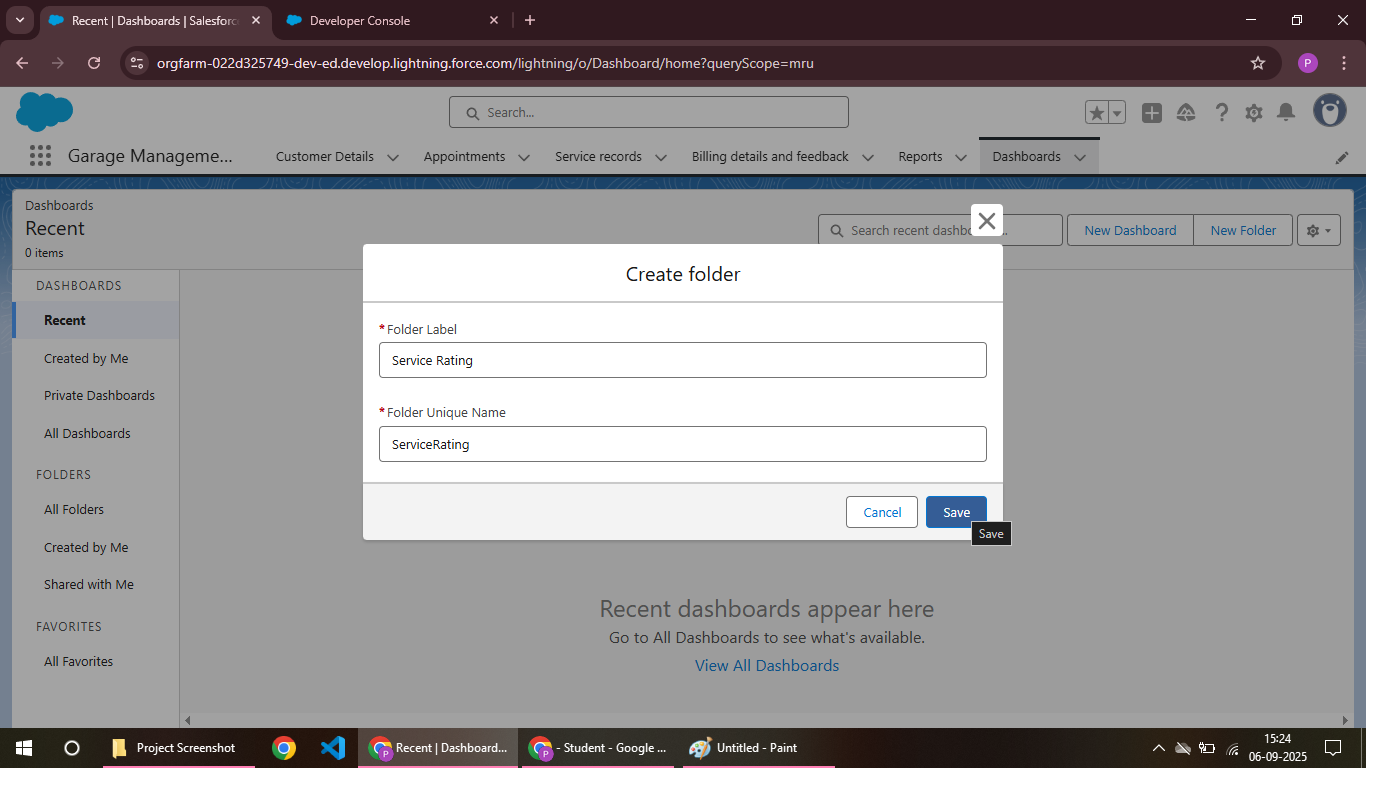




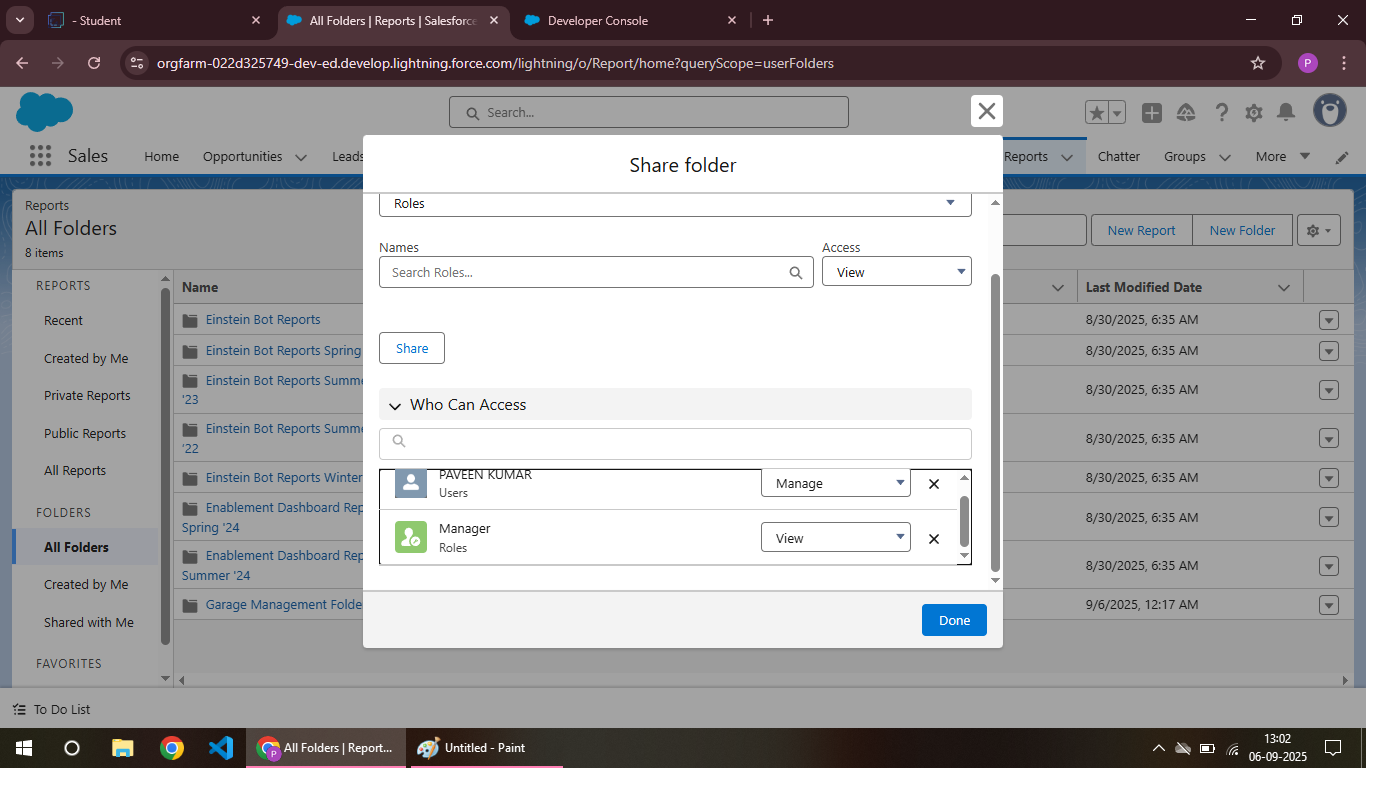
**Reports:**

A Garage Management System (GMS) generates reports on \*Service History\*, \*Revenue\*, \*Inventory\*, and \*Technician Performance\*. These reports provide valuable insights to optimize garage operations and inform business decisions**.**

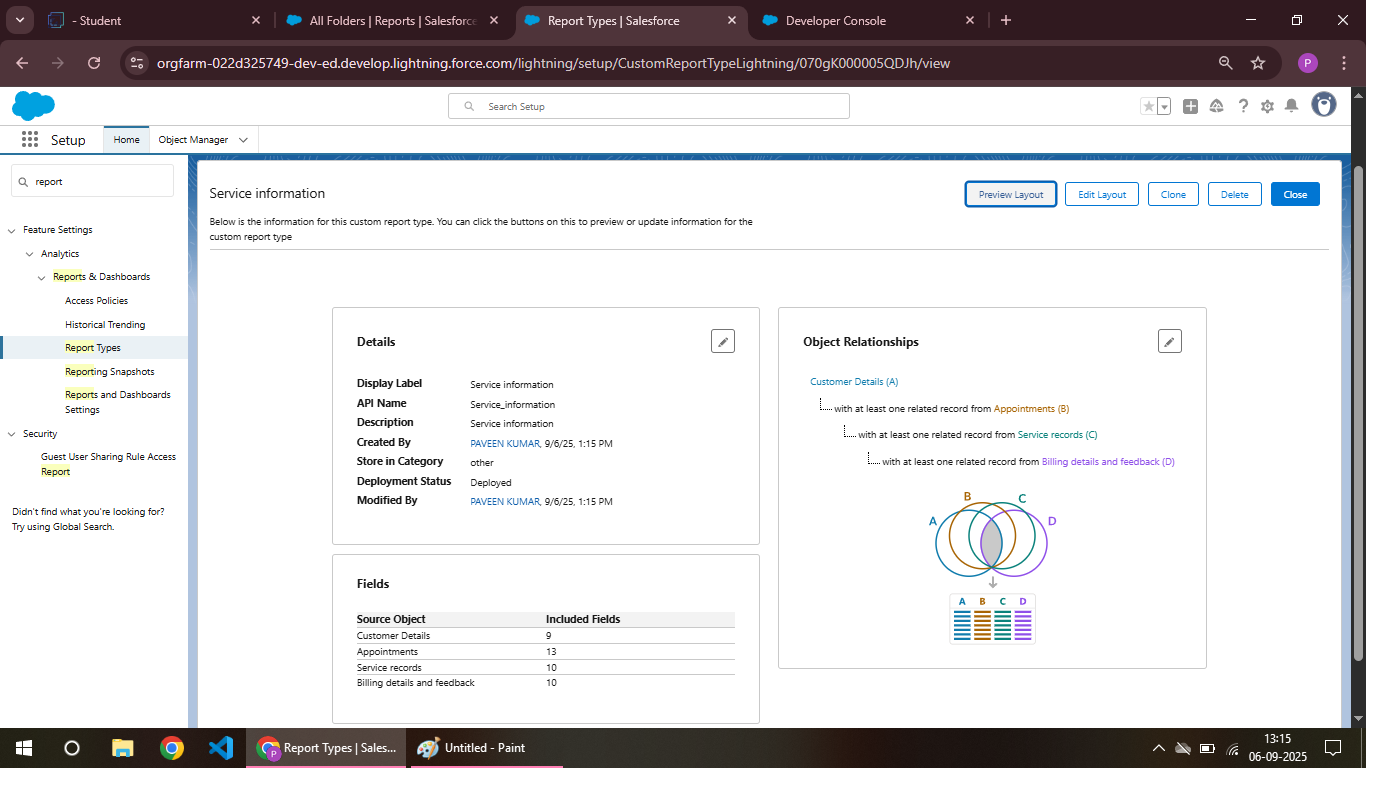
Create A Report Folder:



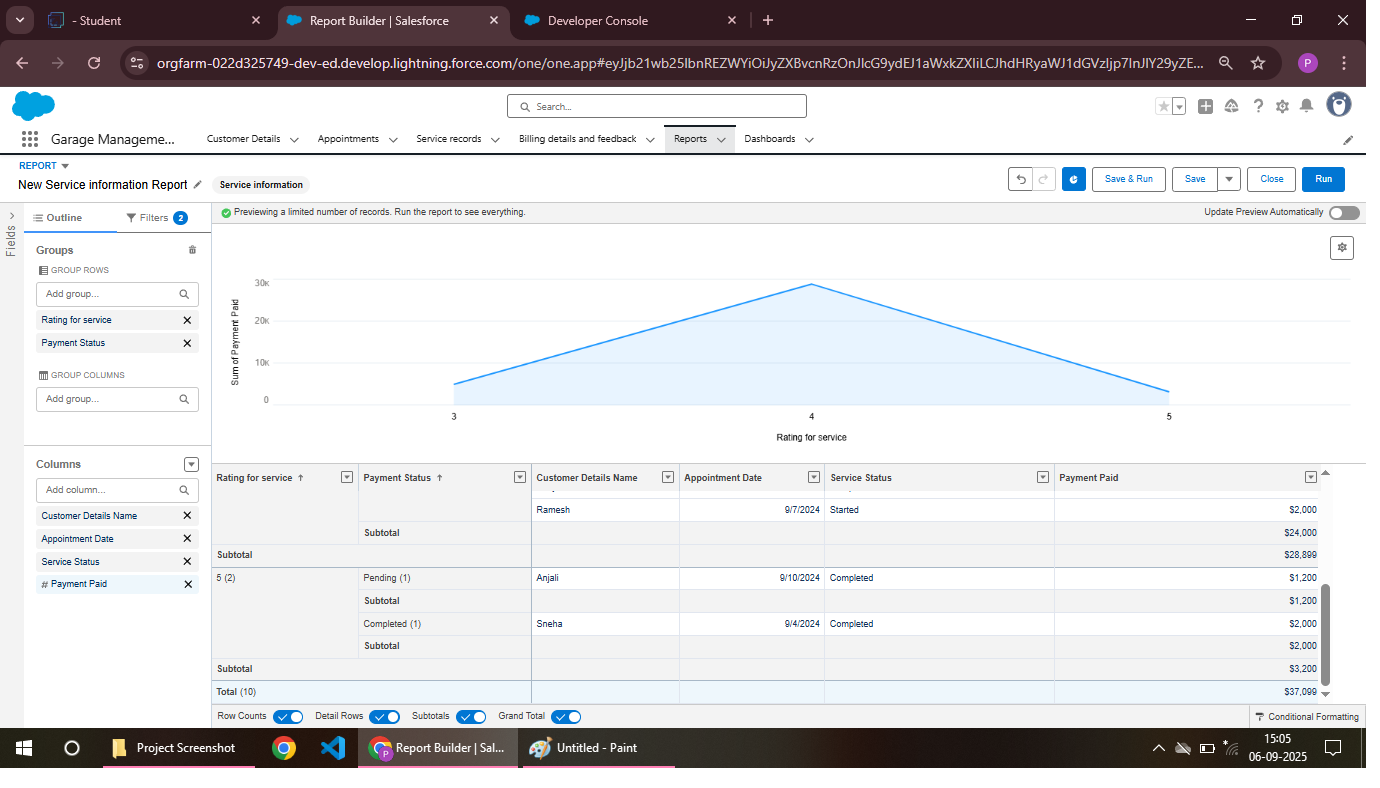
Sharing A Report Folder:



Create Report Type:



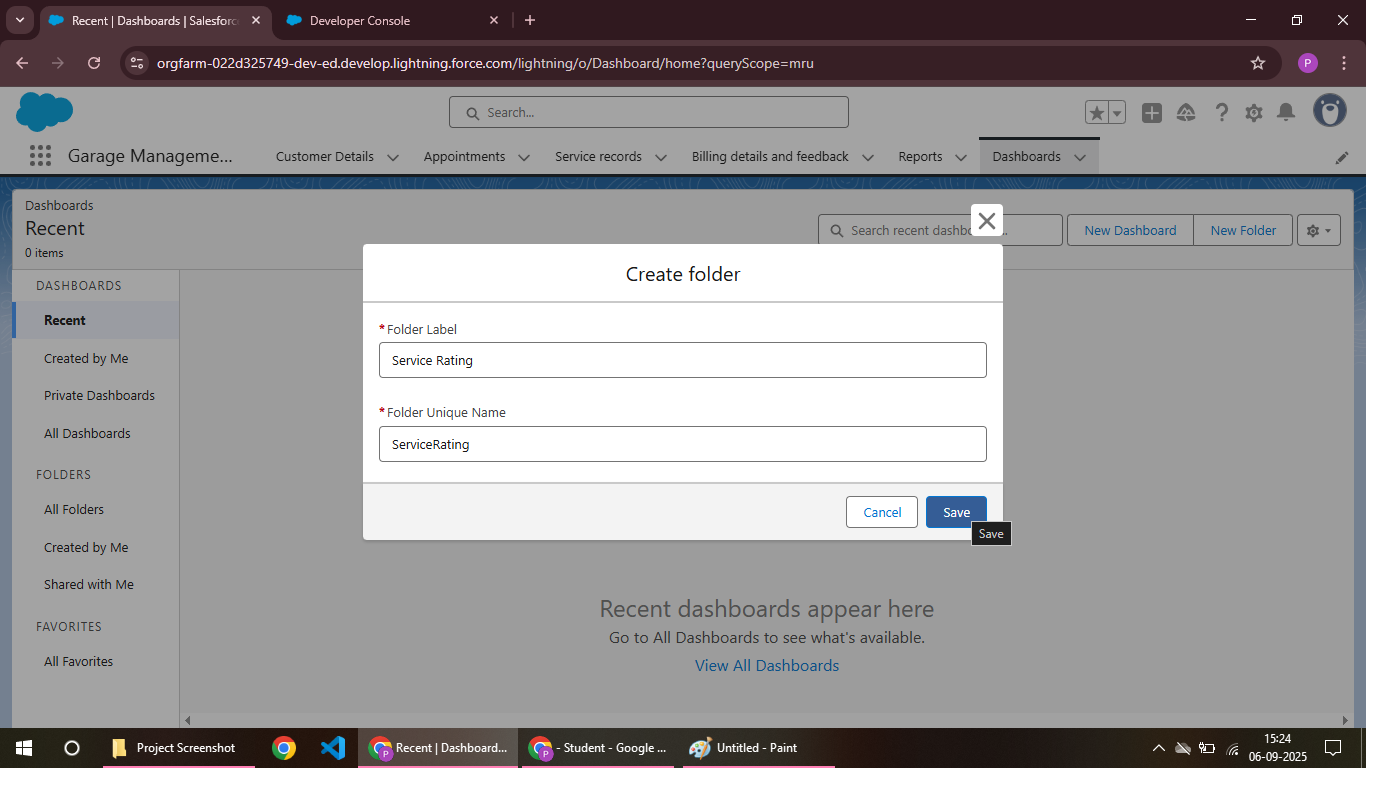
Create Report:

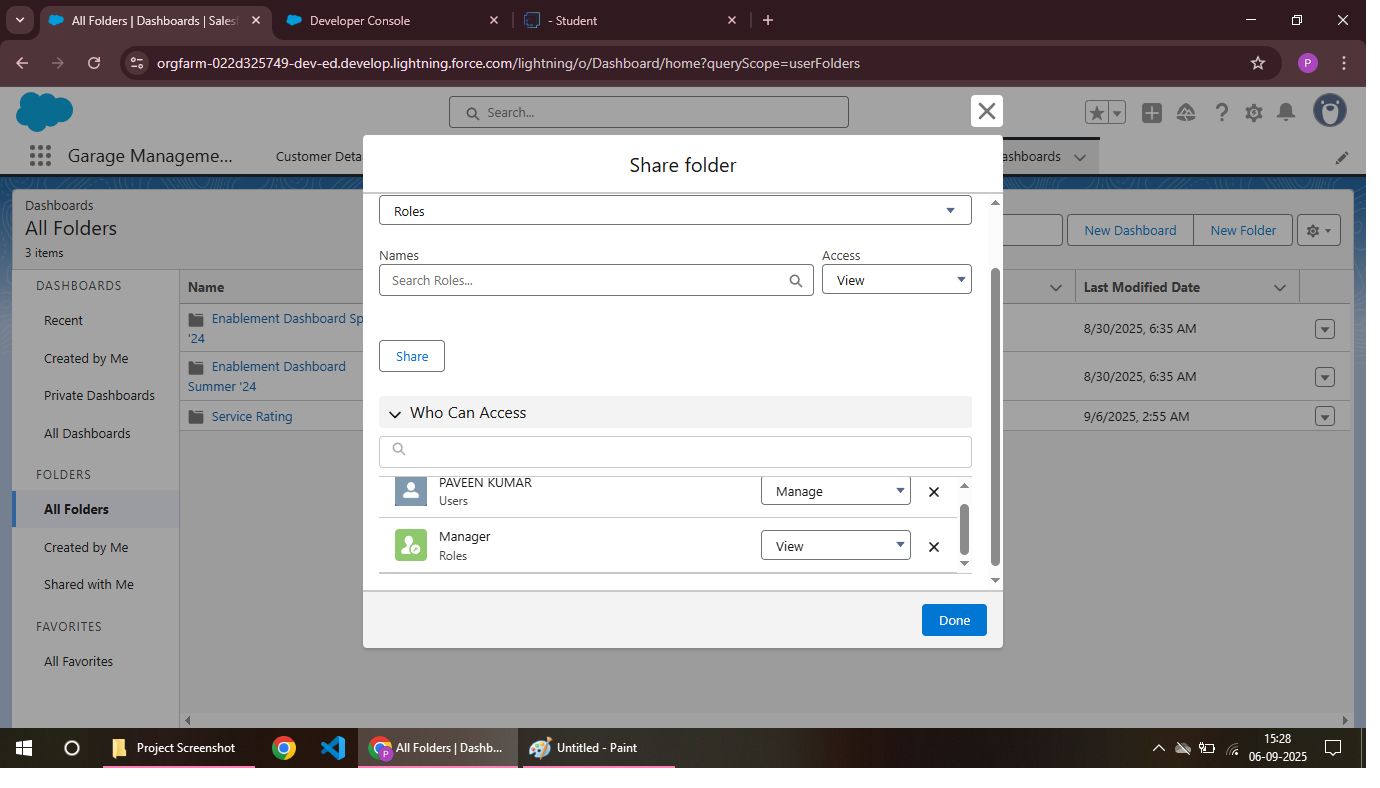


**Dashboards:**

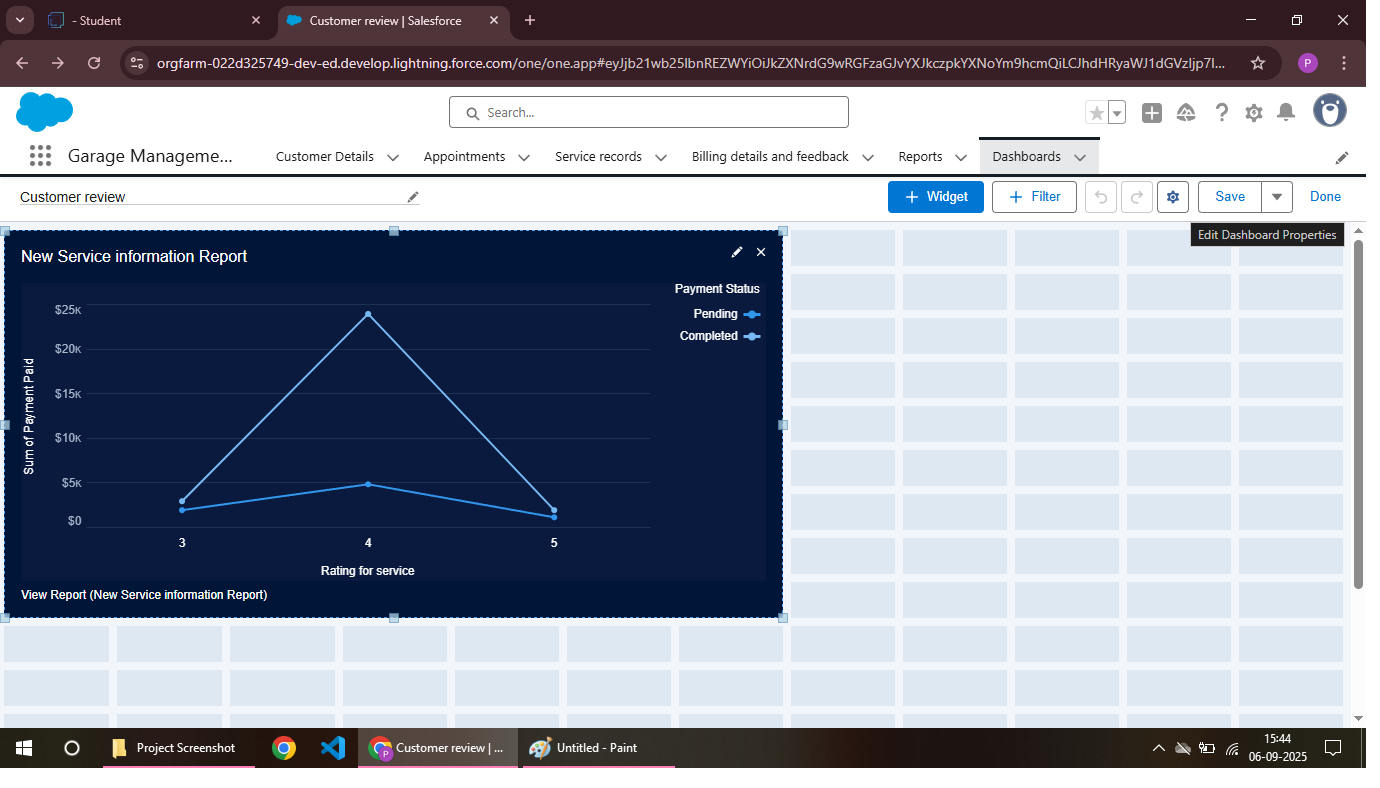
A Garage Management System (GMS) dashboard displays key metrics like \*Appointment Schedule\*, \*Revenue\*, and \*Inventory Levels\*. It provides a centralized view of garage operations and performance.

Create Dashboard Folder:





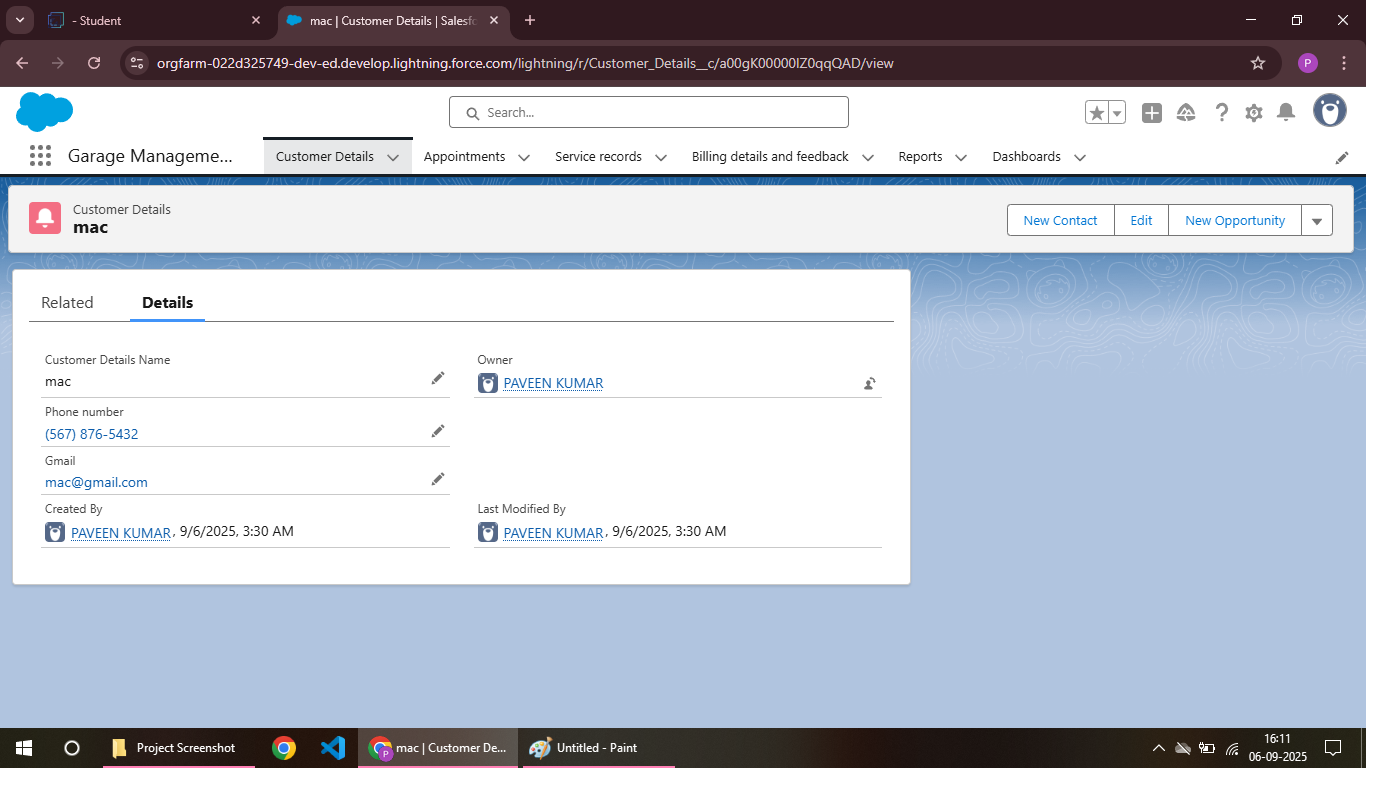
Create Dashboard:

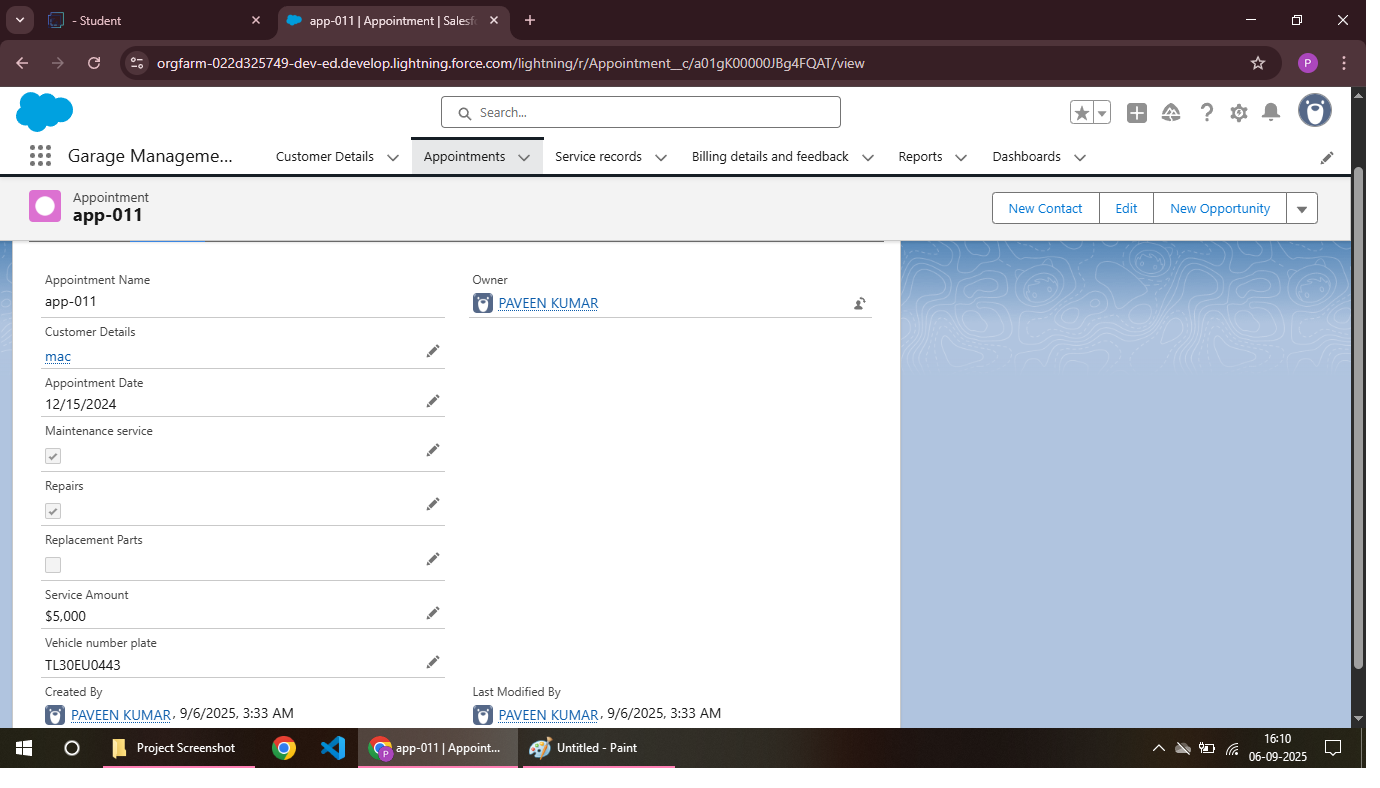


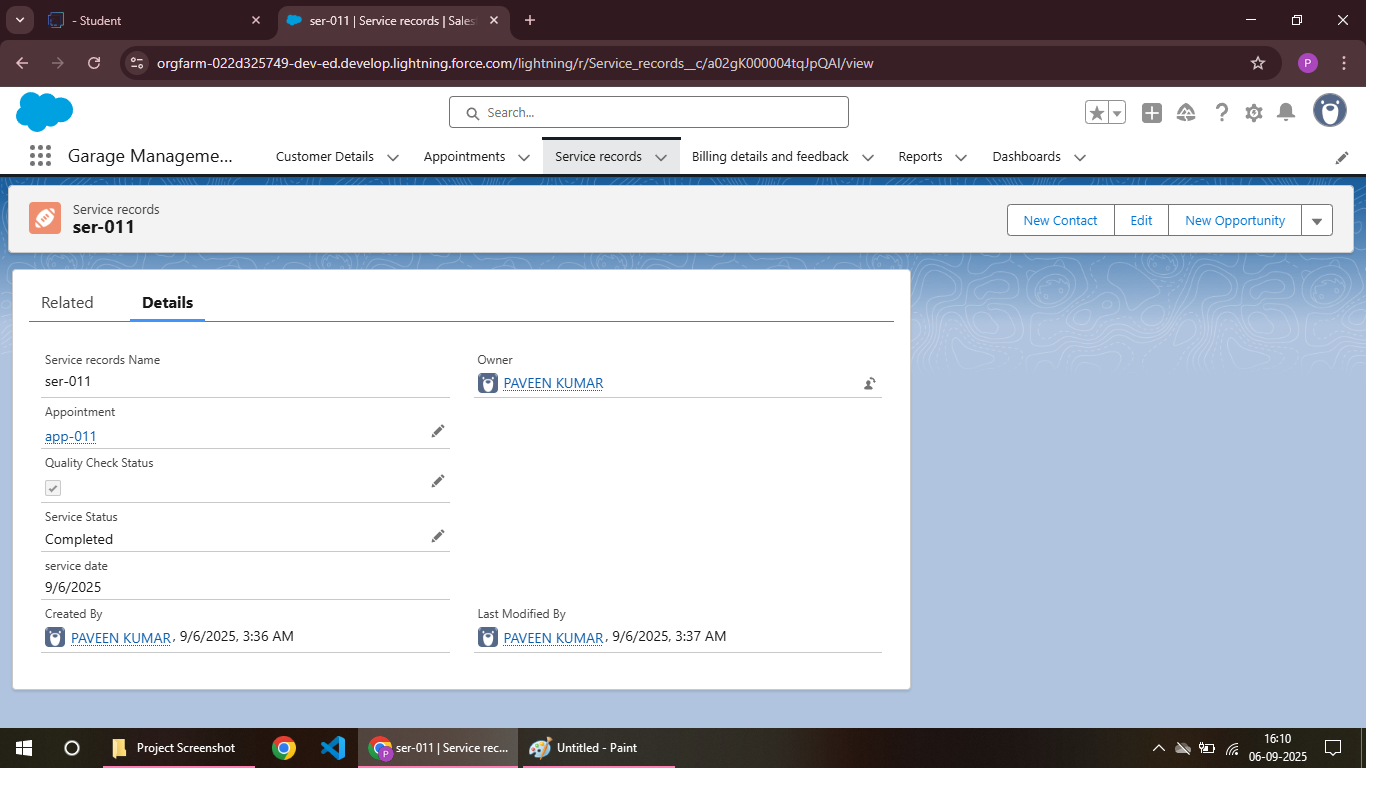
**User Adoption:**

User adoption of a Garage Management System (GMS) requires \*effective training\* and \*ongoing support\* to ensure staff can utilize the system efficiently. This leads to improved productivity and customer satisfaction.

Creating Records:







**CONCLUSION :**

The Garage Management System (GMS) plays a vital role in modernizing and streamlining the operations of automobile service centers. By automating tasks such as customer management, vehicle tracking, service scheduling, billing, and inventory control, the system reduces manual effort, minimizes errors, and saves valuable time.

**APPENDIX :**

**Source Code: Provided in Apex Classes and Triggers**

**AmountDistributionHandler. apxc:**

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;

trigger AmountDistribution on Appointment\_\_c (before insert, before update) {

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

AmountDistributionHandler.amountDist(trigger.new);

}

}

}

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;

}

}

}

}

**AmountDistribution.apxt:**

trigger AmountDistribution on Appointment\_\_c (before insert, before update) {

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

AmountDistributionHandler.amountDist(trigger.new);

}

}