GARAGE MANAGEMENT SYSTEM

COLLEGE NAME: L.R.G.Government Arts College for Women-Tirupur.

COLLEGE CODE: bru07

<u>TEAM ID:NM2025TMID28045</u>

<u>TEAM MEMBERS:-</u>

TEAM LEADER NAME: YUVARAJI C

EMAIL:yuvachandru2006@gmail.com

TEAM MEMBER1: DEVAYANI E

EMAIL:e.devayani0064@gmail.com

TEAM MEMBER2: DHARSHINI SHREE K

EMAIL:rr6709922@gmail.com

TEAM MEMBER3: DHARSINIPRIYA G

EMAIL:dharshinipriya88811@gmail.com

TEAM MEMBER4: DIVYADHARSHINI S

EMAIL:dividivya978@gmail.com

1. INTRODUCTION:-

1.1 Project Overview:-

The Garage management System is a software application designed to manage and streamline the day-to-day operations of a garage or vehicle service center. It helps automate processes such as customer details, appointments, service records, billing and reports.

1.2 Purpose:-

The main purpose of the project is to:

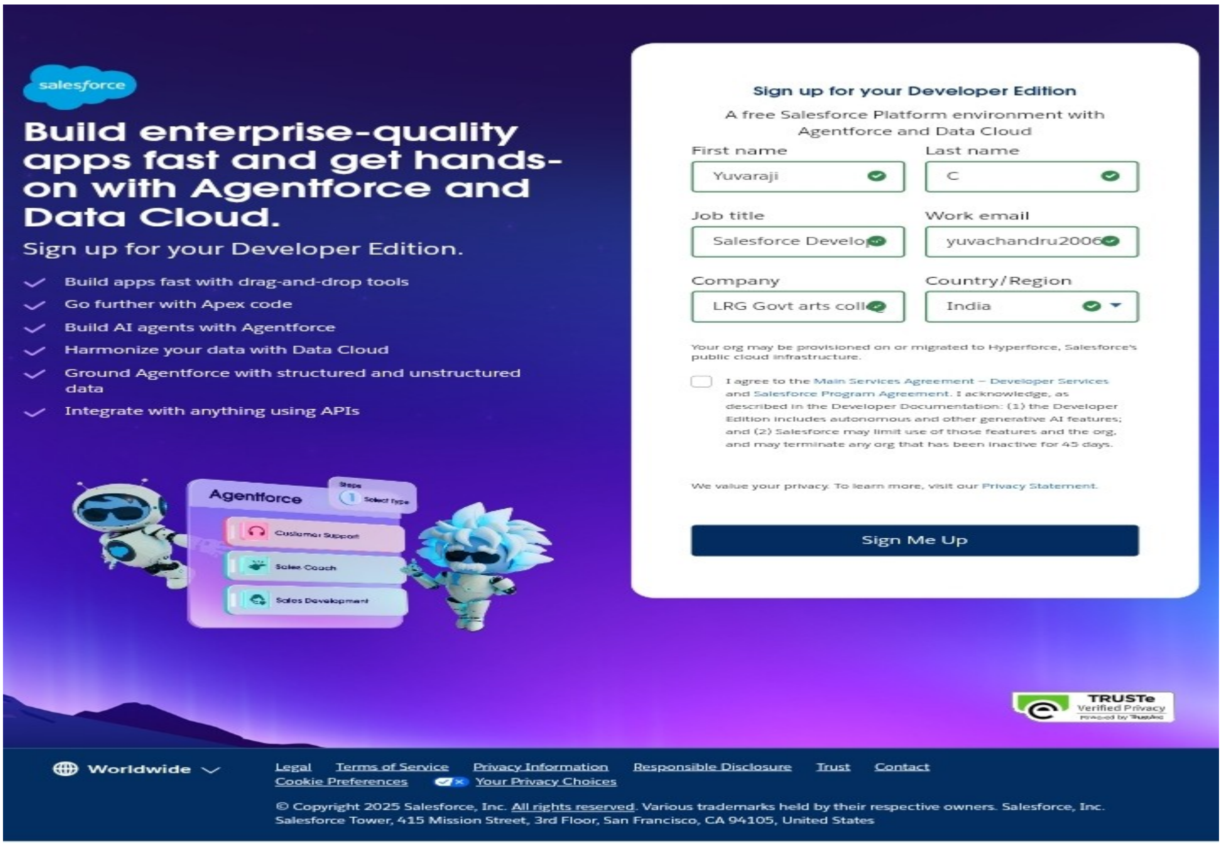
- Track vehicle services and spare parts inventory.
- Maintain accurate records of customers and their vehicles.
- Simplify job scheduling and mechanic assignments.
- Generate quick and error-free bills and invoices.
- Automate and streamline the operations of a vehicle service center or garage.
- Provide useful reports for management decisions.

2. DEVELOPMENT PHASE:-

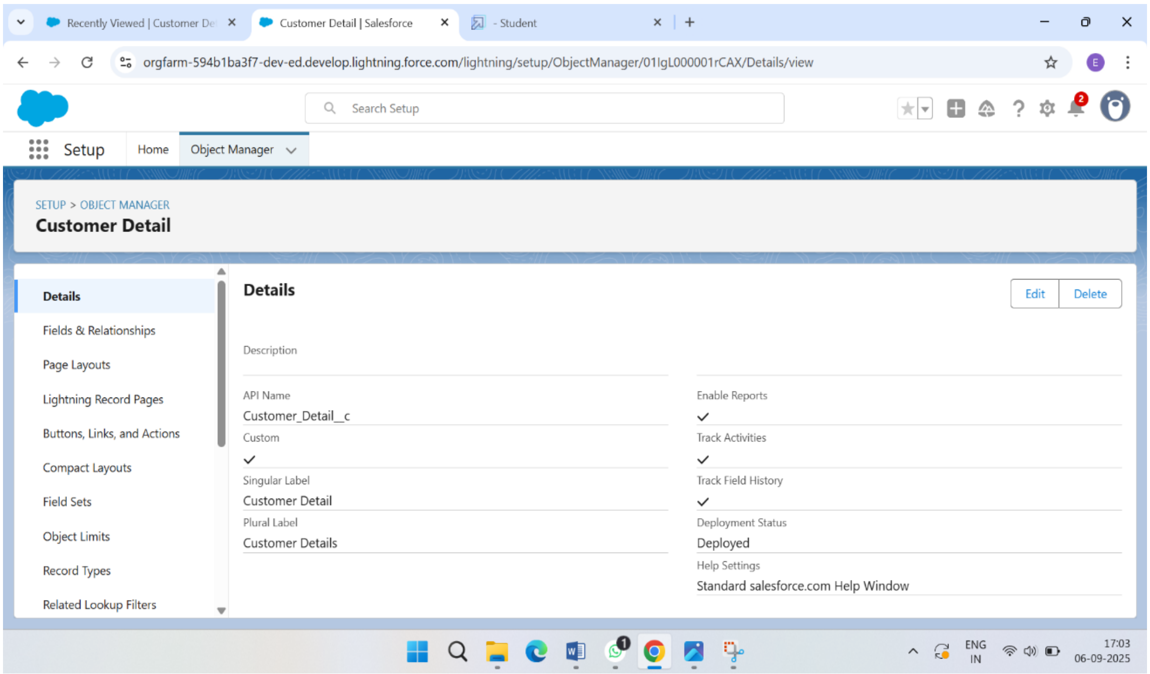
Creating Developer Account:

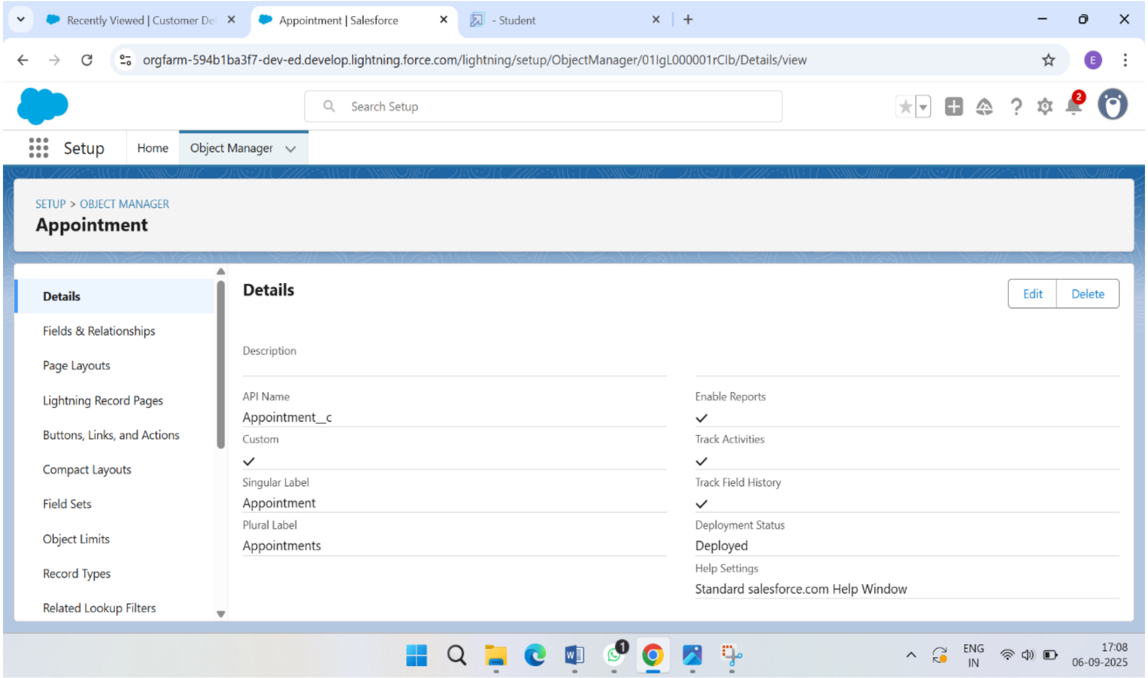
The project was developed on salesforce developer org, created via:

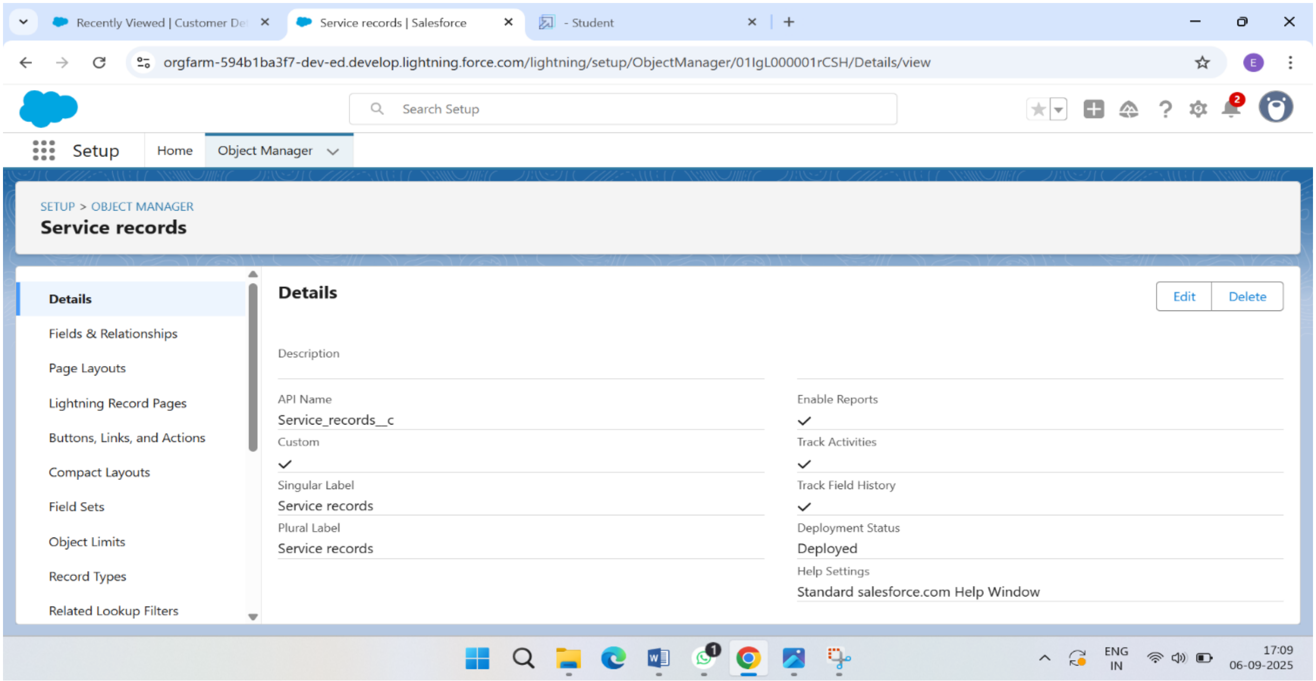
https://developer.salesforce.com/signup.

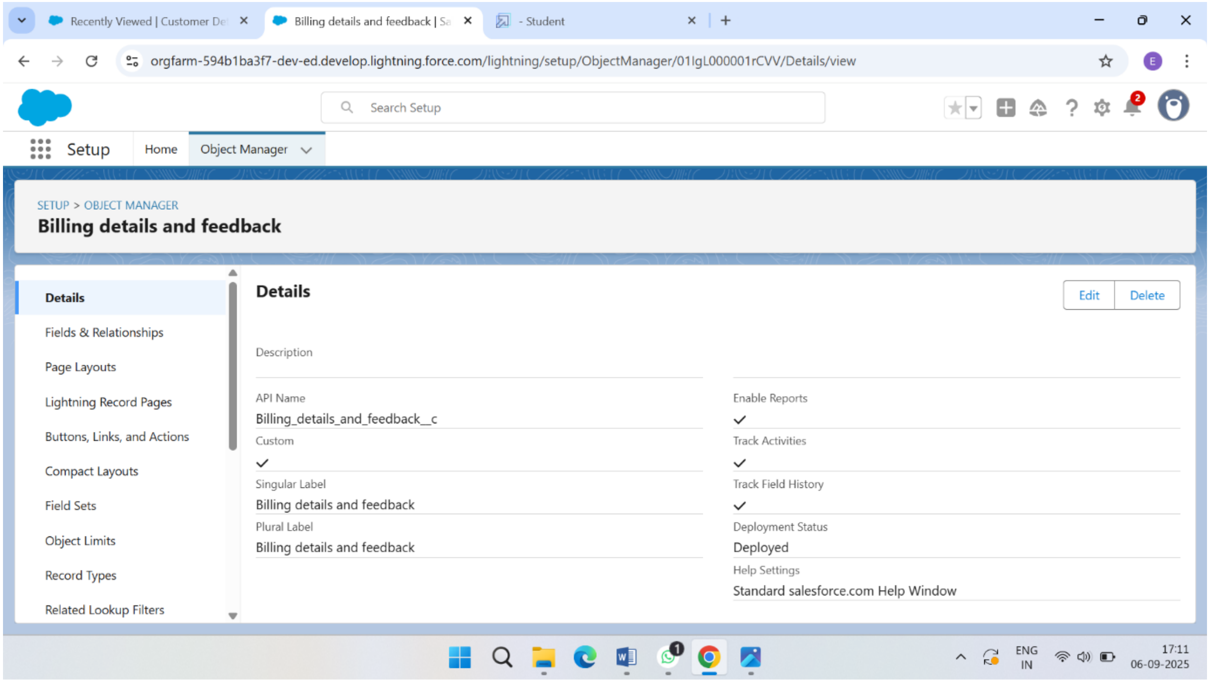


- Created objects: Customer Details, Appointment, Service record,Billing Details.

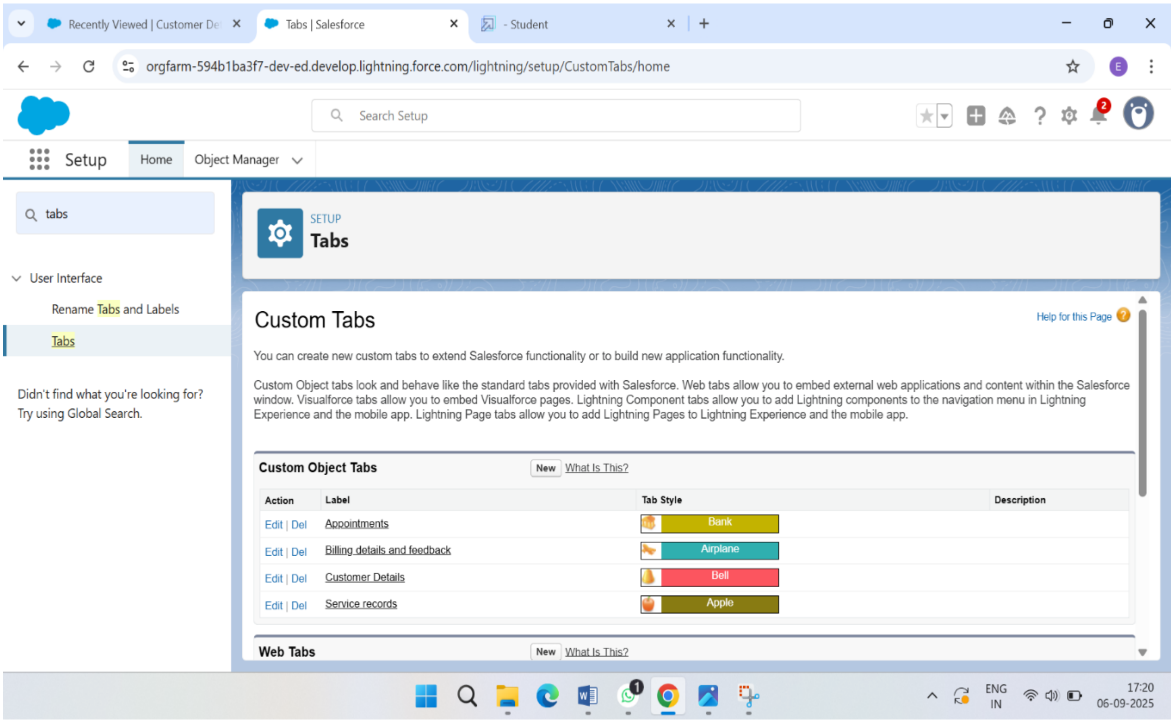



Recently Viewed | Customer De
Appointment | Salesforce
- Student
orgfarm-594b1ba3f7-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL000001rClb/Details/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Appointment
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Details
Edit
Delete
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
ENG
IN
17:08
06-09-2025

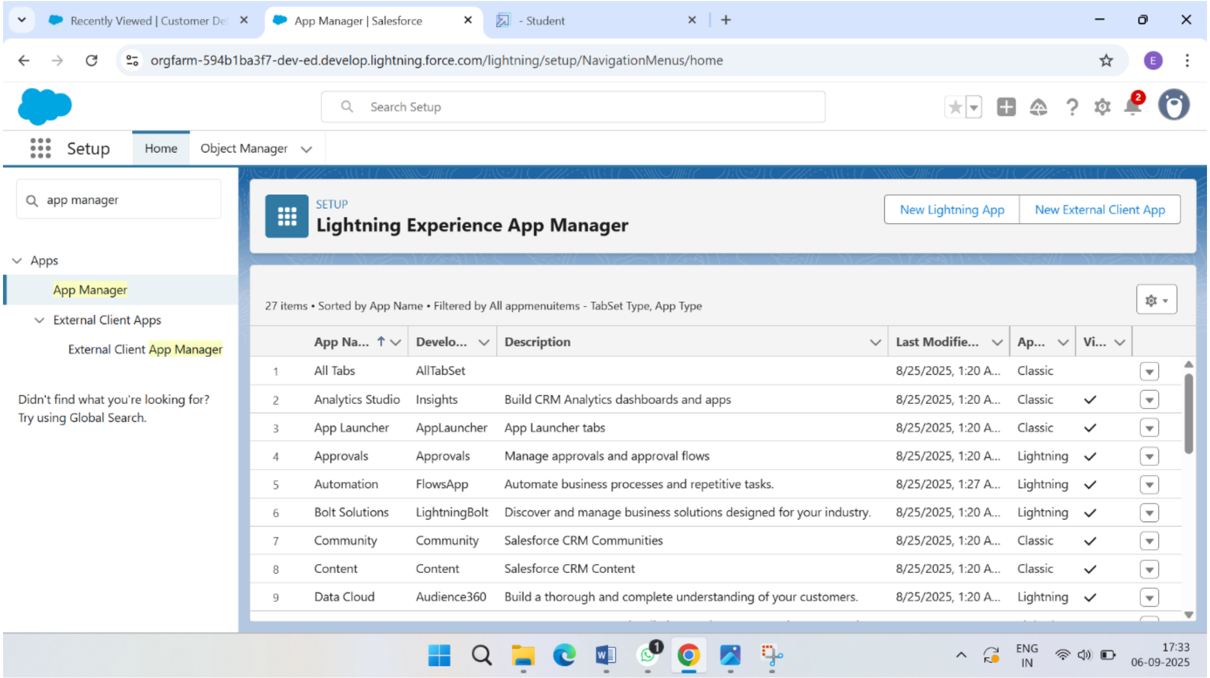

Recently Viewed | Customer De
Service records | Salesforce
- Student
orgfarm-594b1ba3f7-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL000001rCSH/Details/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Service records
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Details
Edit
Delete
Description
API Name
Service_records__c
Custom
Singular Label
Service records
Plural Label
Service records
Enable Reports
Track Activities
Track Field History
Deployment Status
Deployed
Help Settings
Standard salesforce.com Help Window
ENG
IN
17:09
06-09-2025



- Create the Tabs for the objects, they are "Customer Details, Appointment, Service Record and Billing Details and feedback".

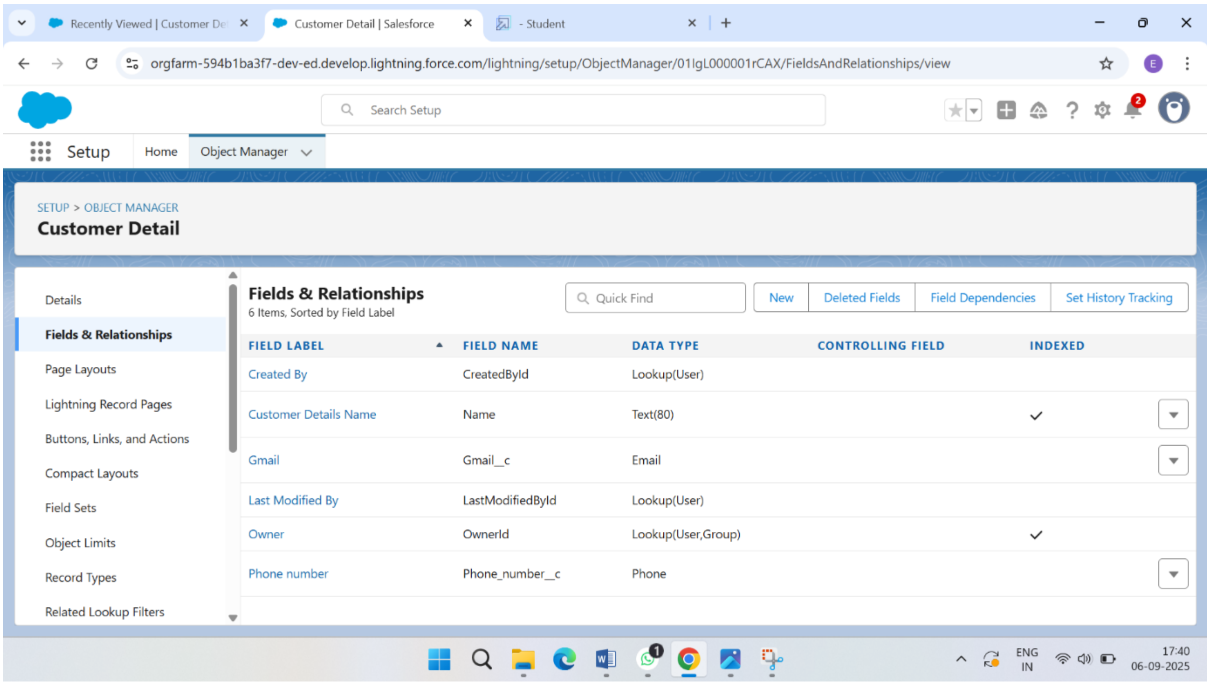


- Lightings App: Developed Lighting App with relevant tabs.

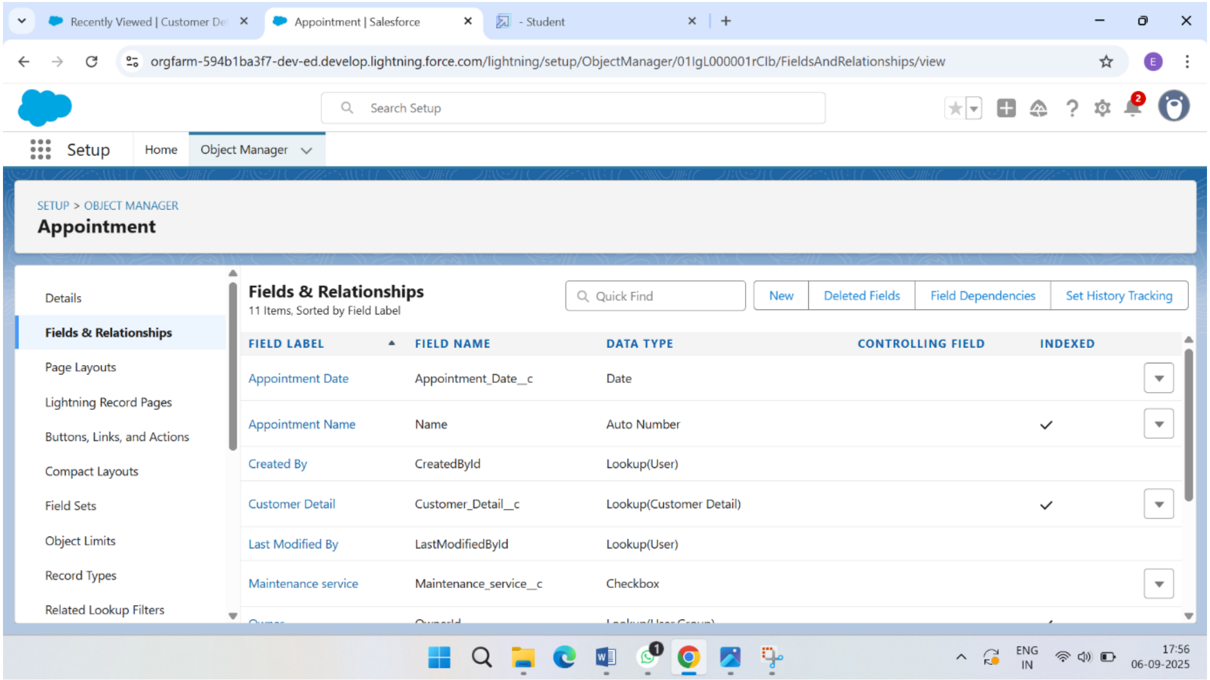


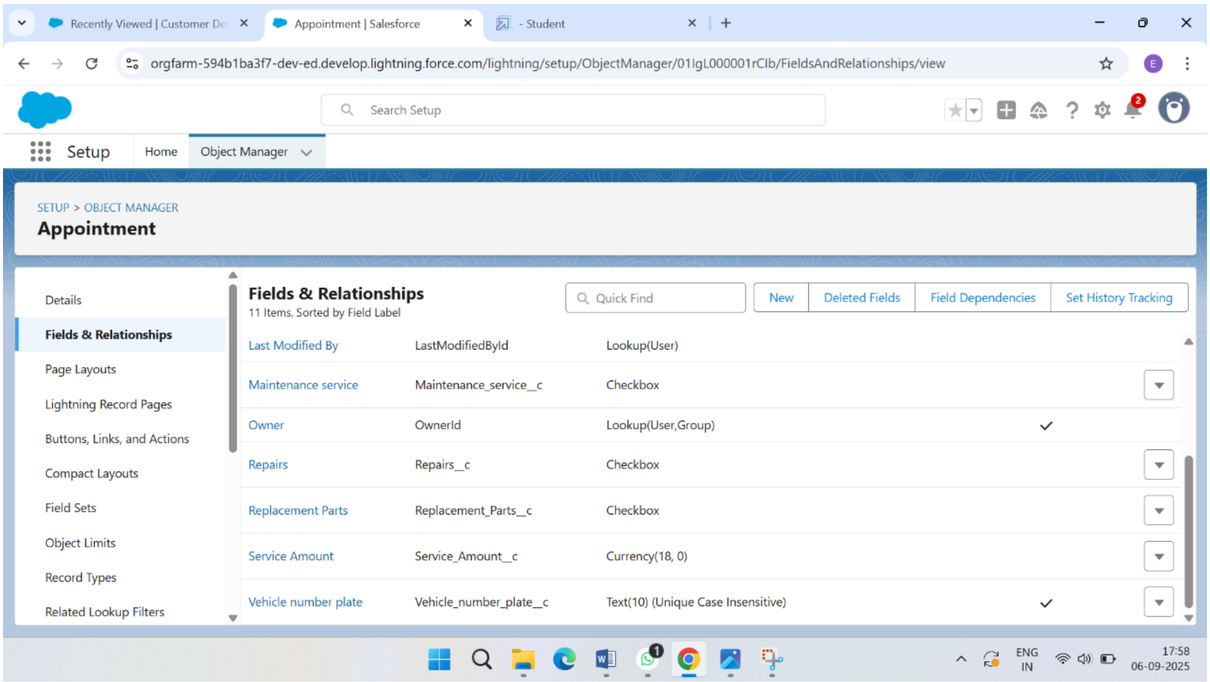
Fields:

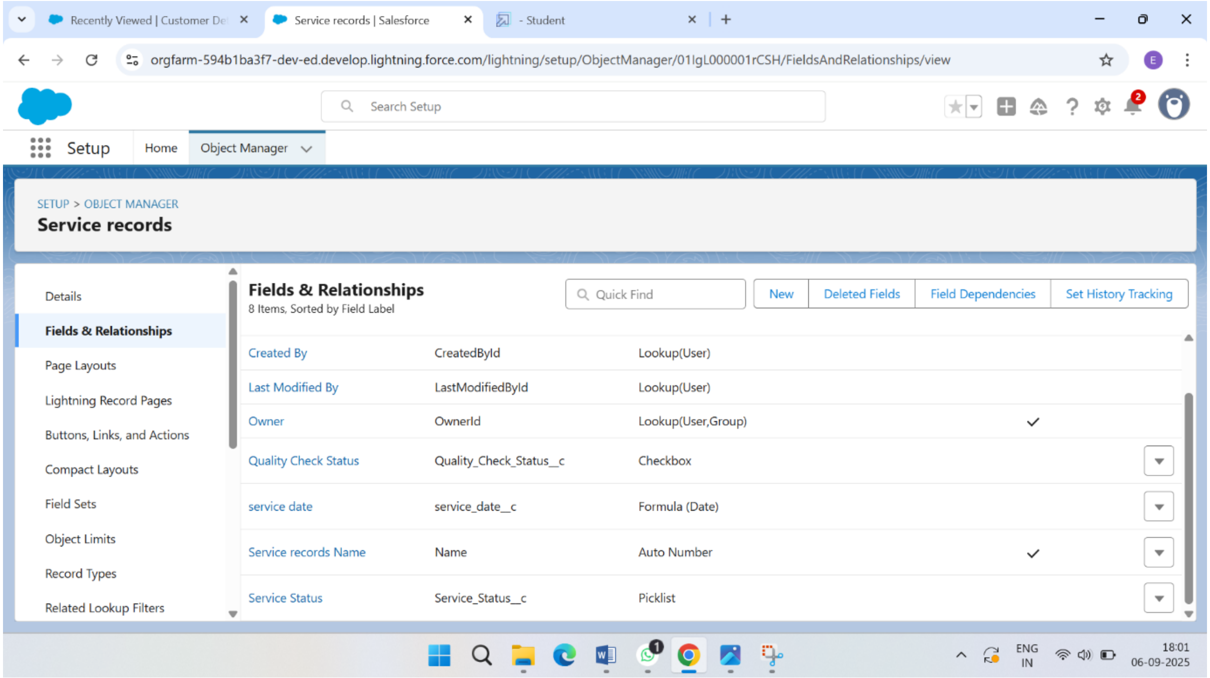
- Create a fields for the customer detail objects

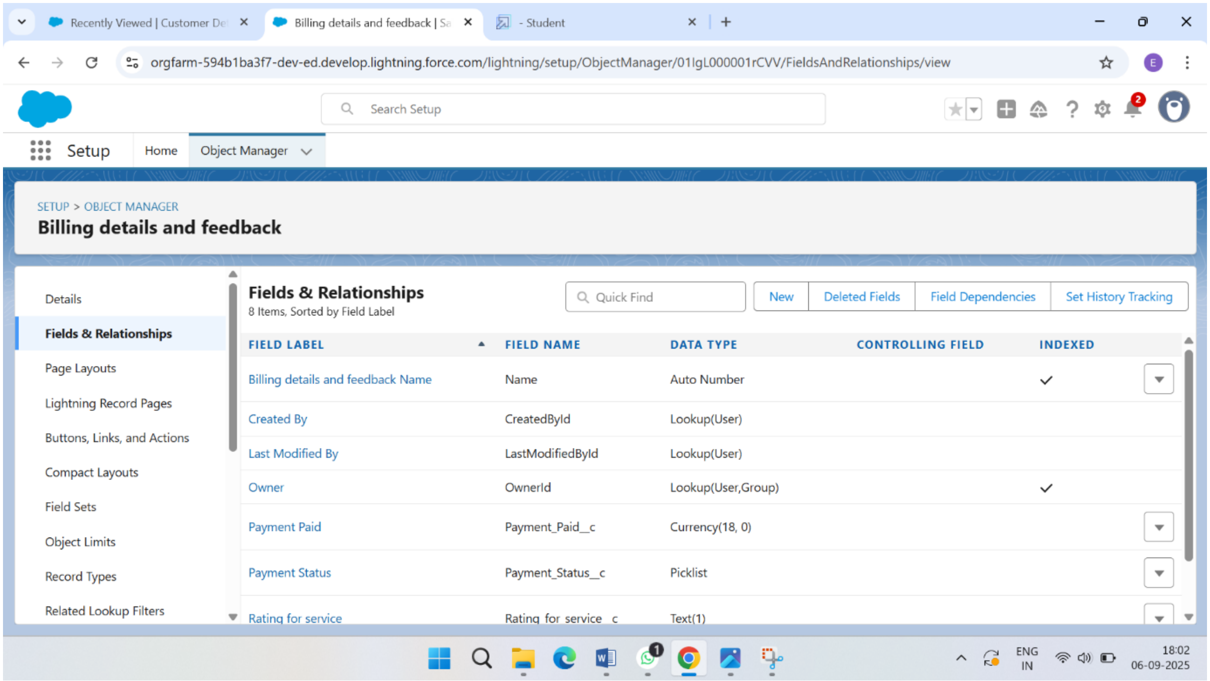


- The fields of lookup, check box, date, currency, text, picklist and formula fields in service record objects also created.

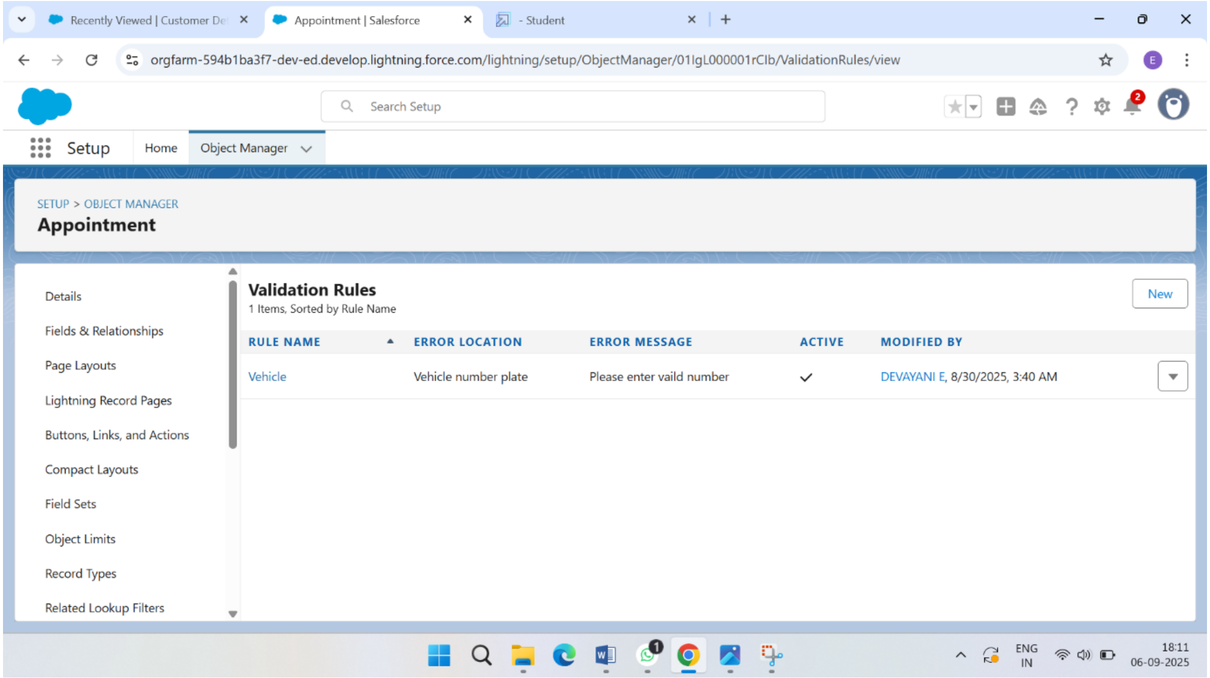


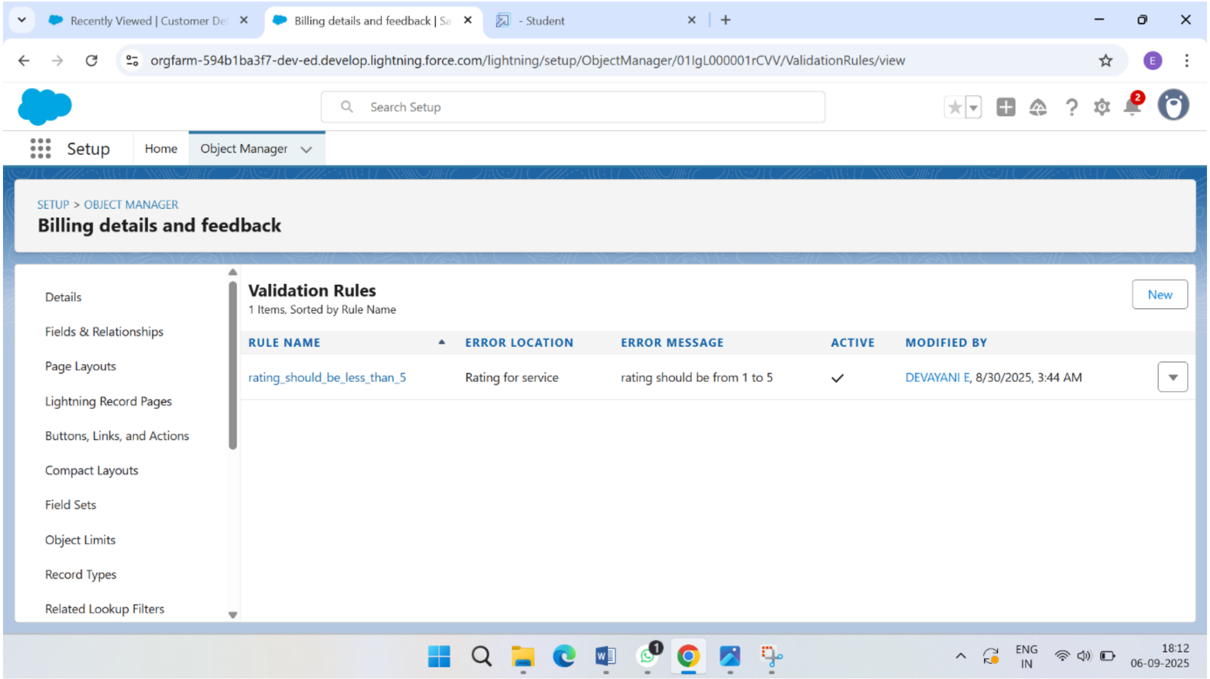




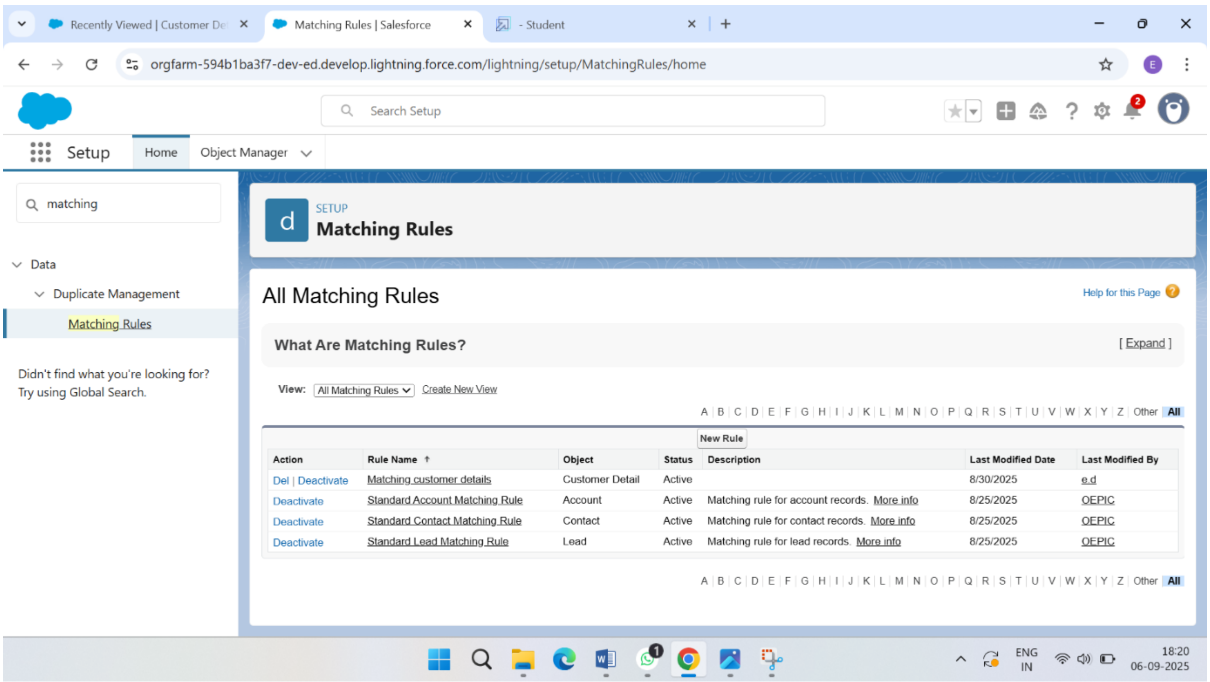


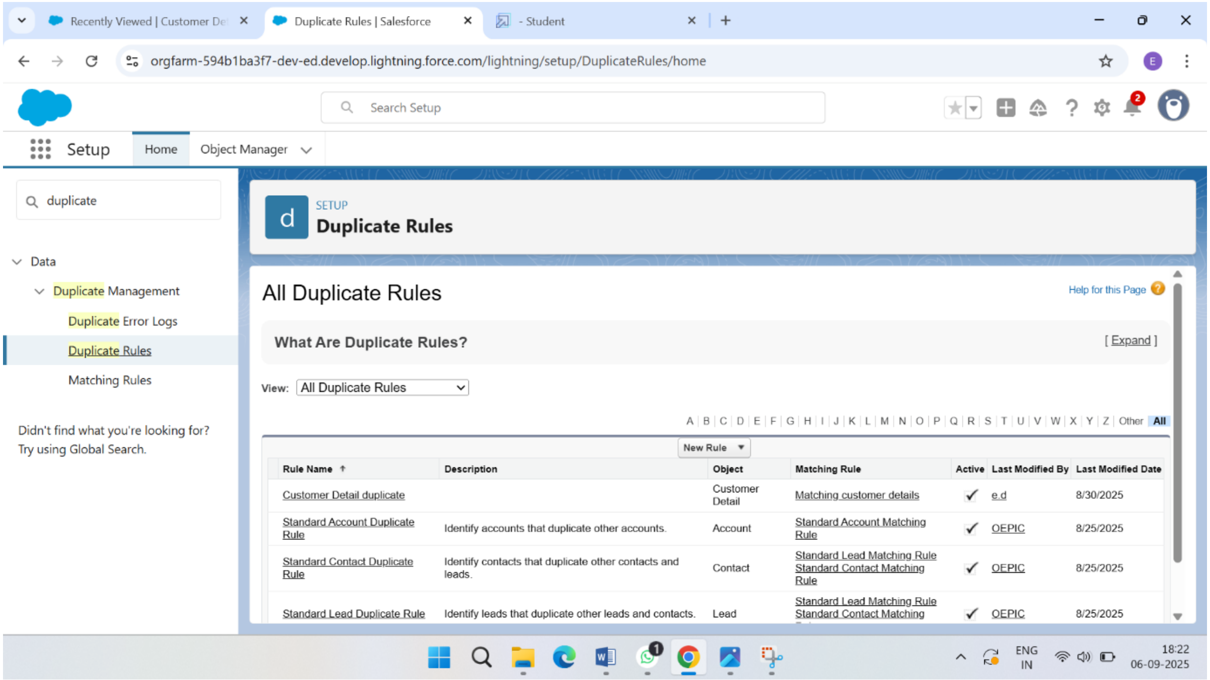
- Create a validation rules is appointment object and Billing details and feedback.


Recently Viewed | Customer De
Appointment | Salesforce
- Student
orgfarm-594b1ba3f7-dev-ed.develop.lightning.force.com/lightning/setup/ObjectManager/01IgL000001rClb/ValidationRules/view
Search Setup
Setup
Home
Object Manager
SETUP > OBJECT MANAGER
Appointment
Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Validation Rules
1 Items, Sorted by Rule Name
New
RULE NAME
ERROR LOCATION
ERROR MESSAGE
ACTIVE
MODIFIED BY
Vehicle
Vehicle number plate
Please enter vaild number
DEVAYANI E, 8/30/2025, 3:40 AM
ENG
IN
18:11
06-09-2025

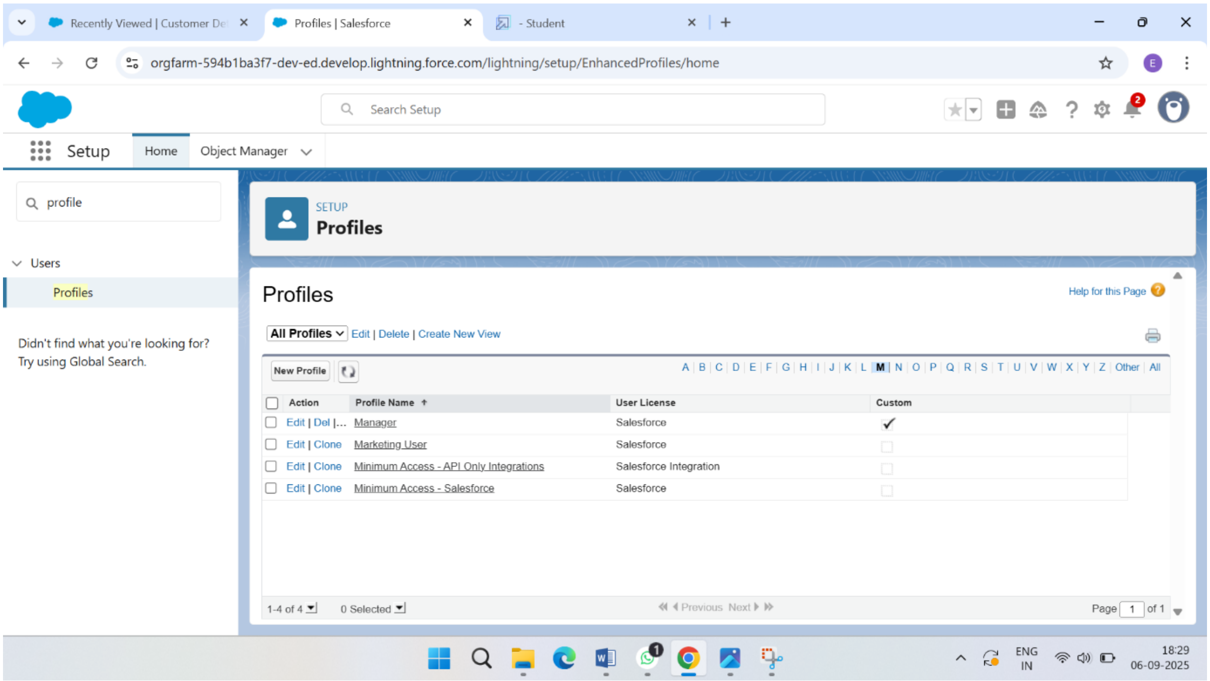


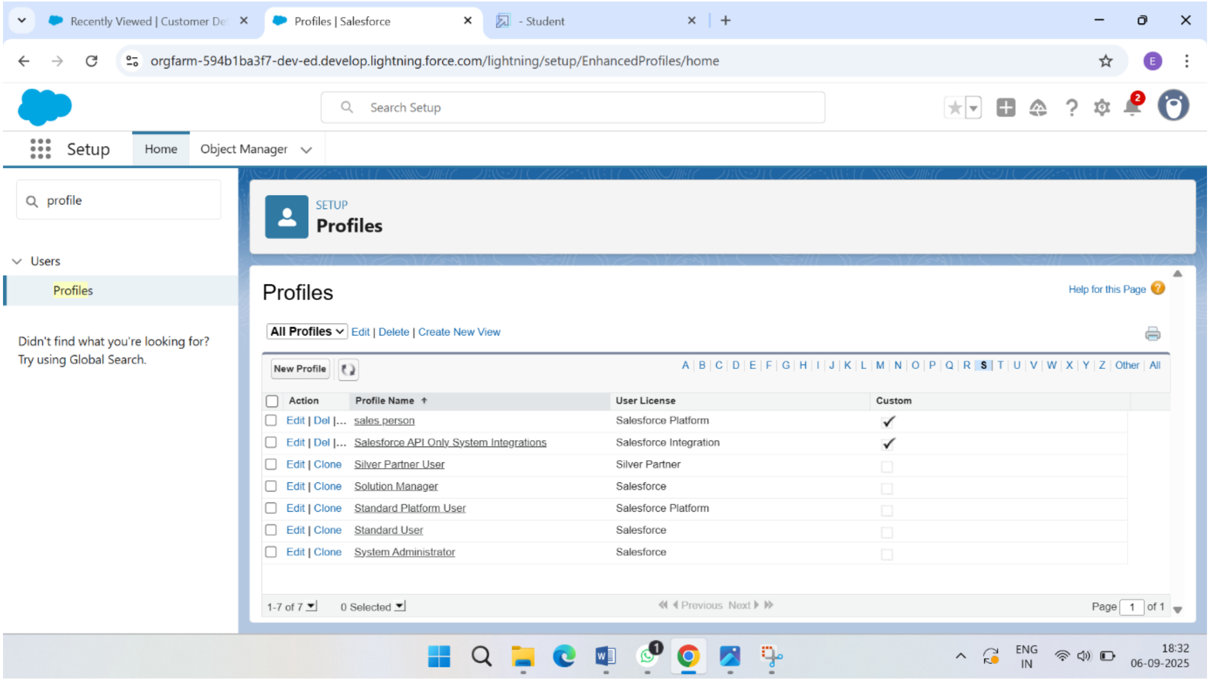
- The customer detail object is create the matching rule and duplicate rule.



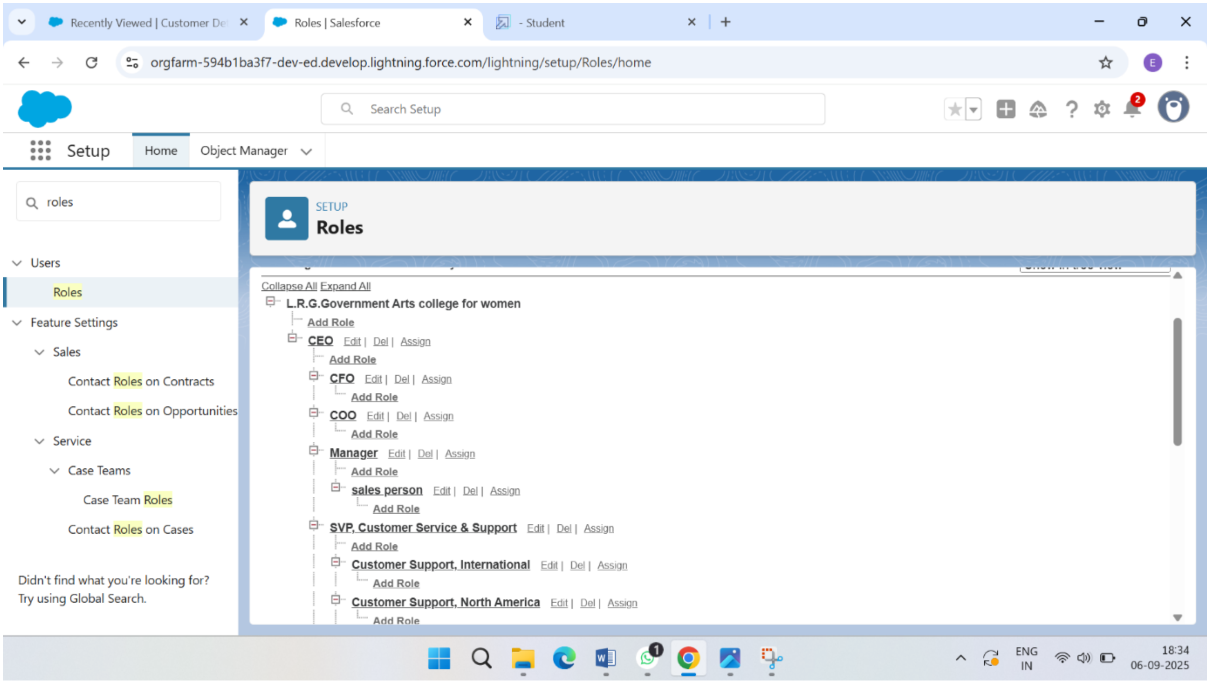


- Create the profiles.

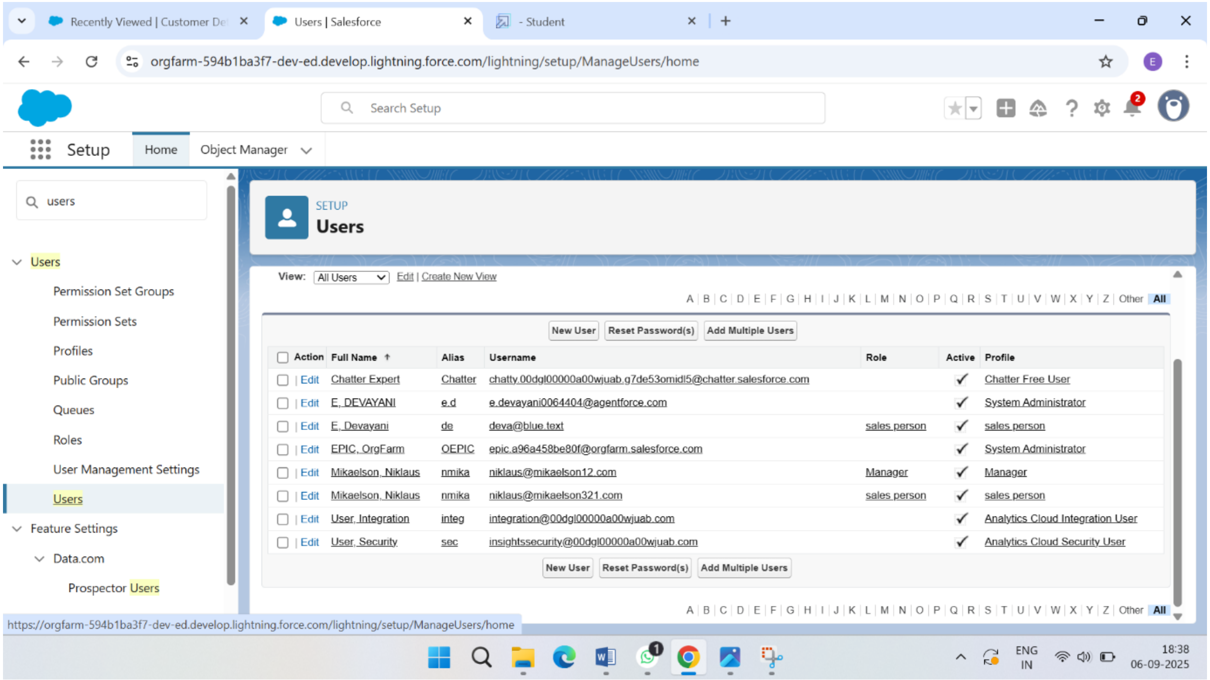


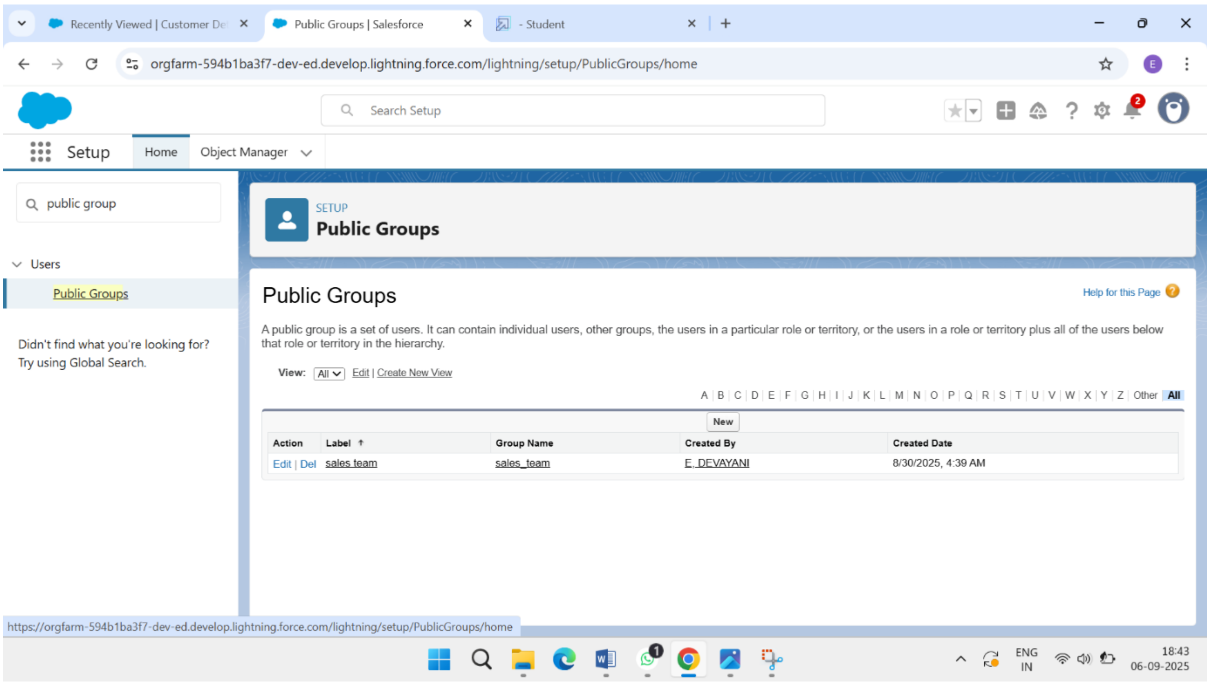


- Roles are created.

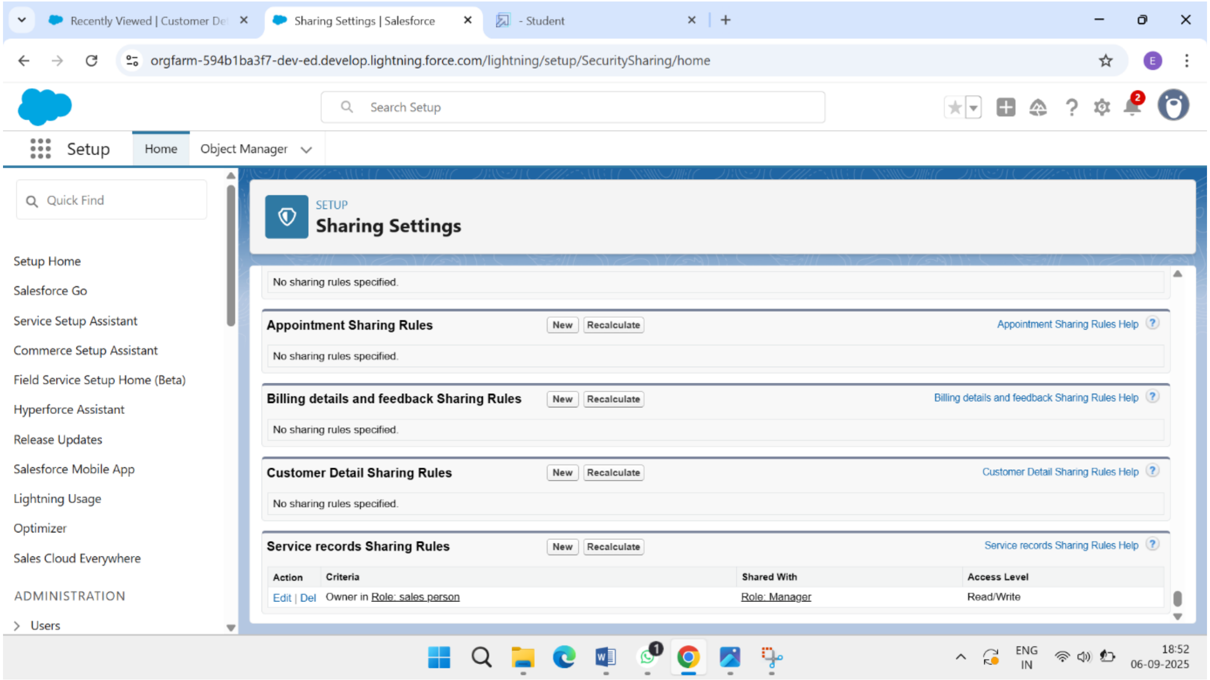


- It is create the users.

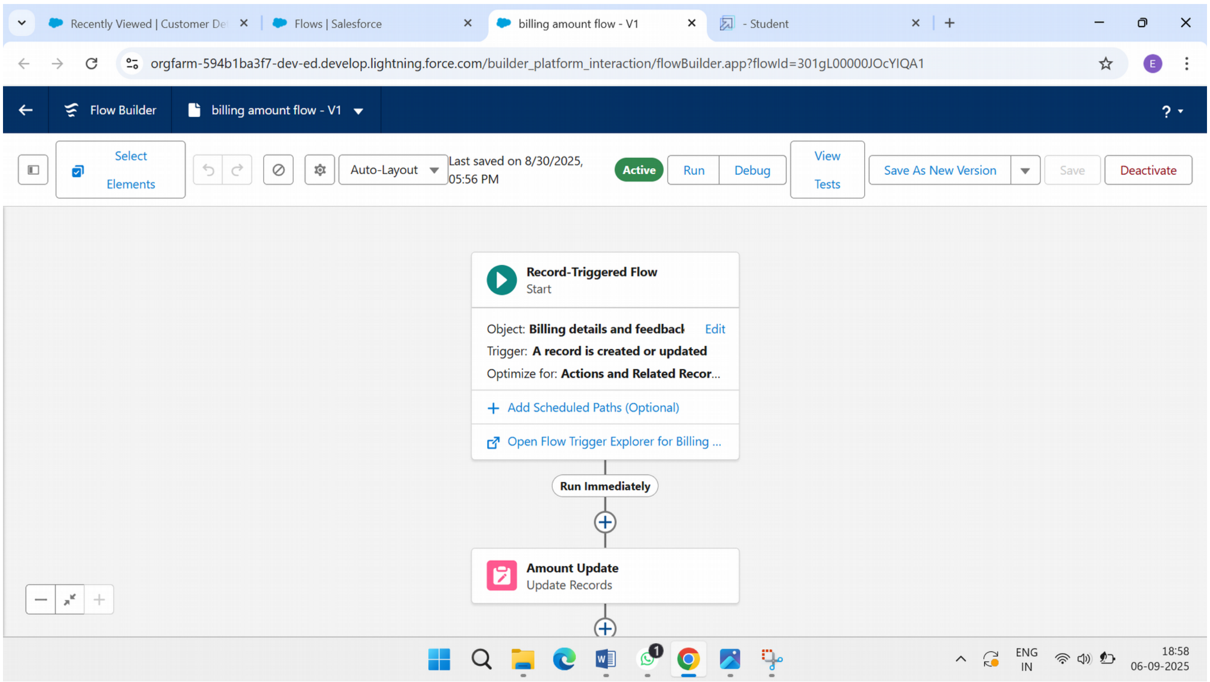


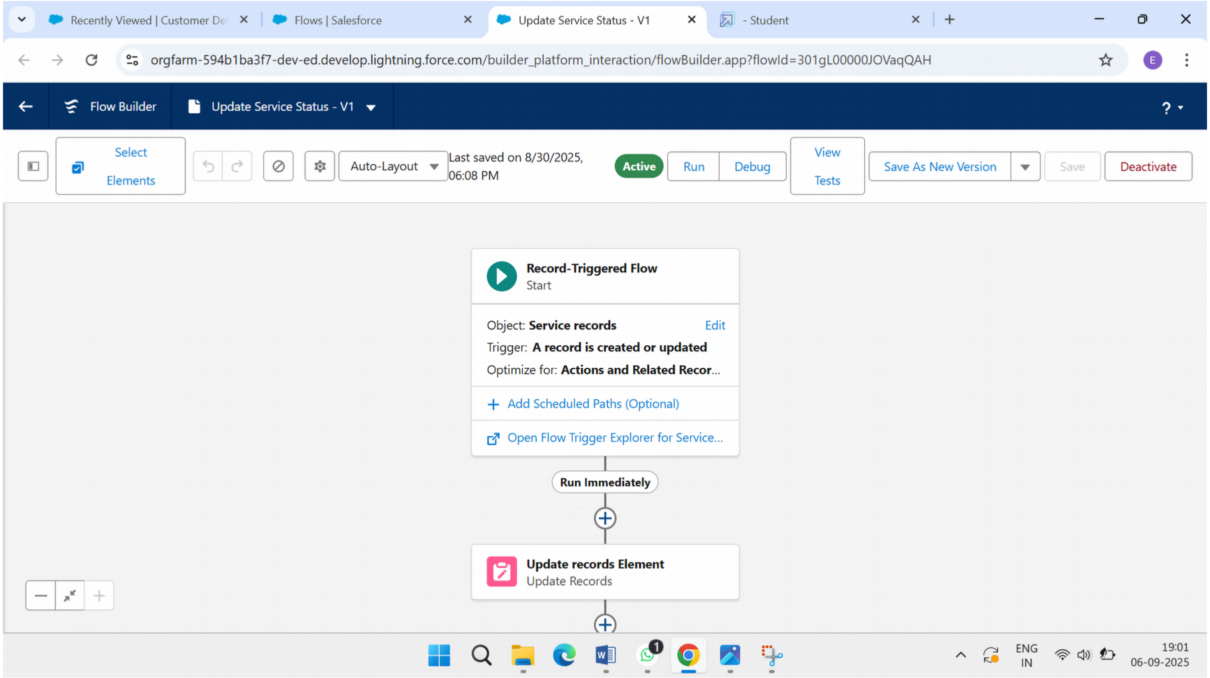


- To create sharing settings.



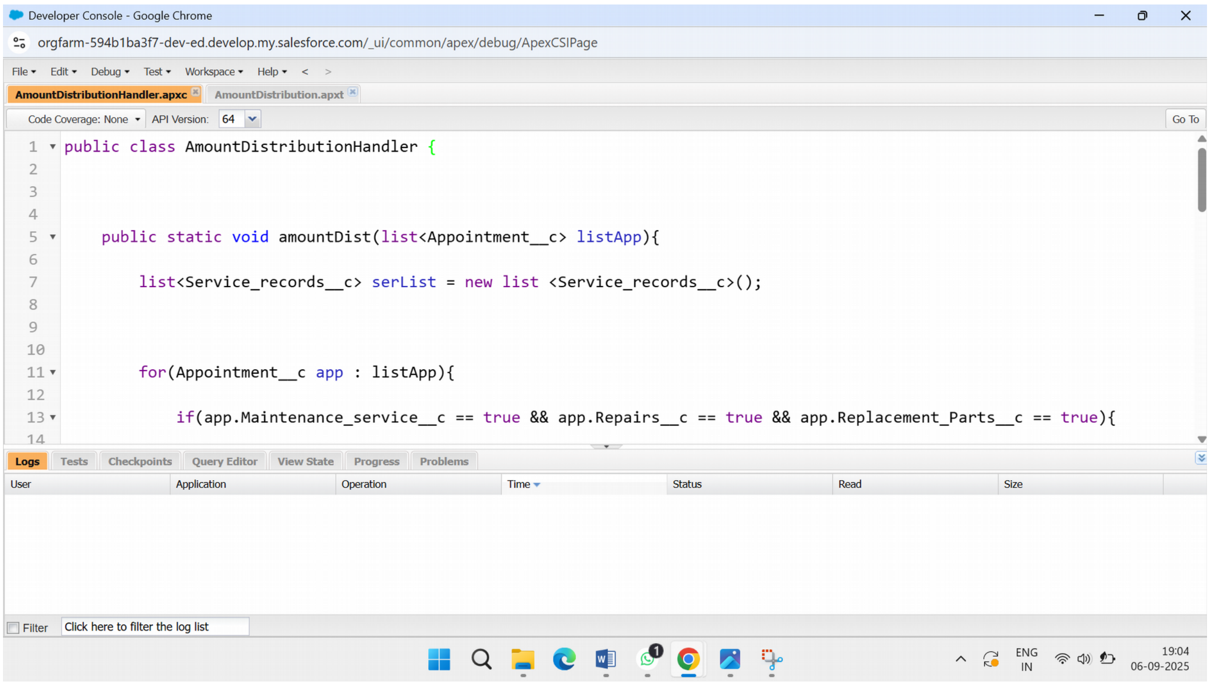
- Flows.



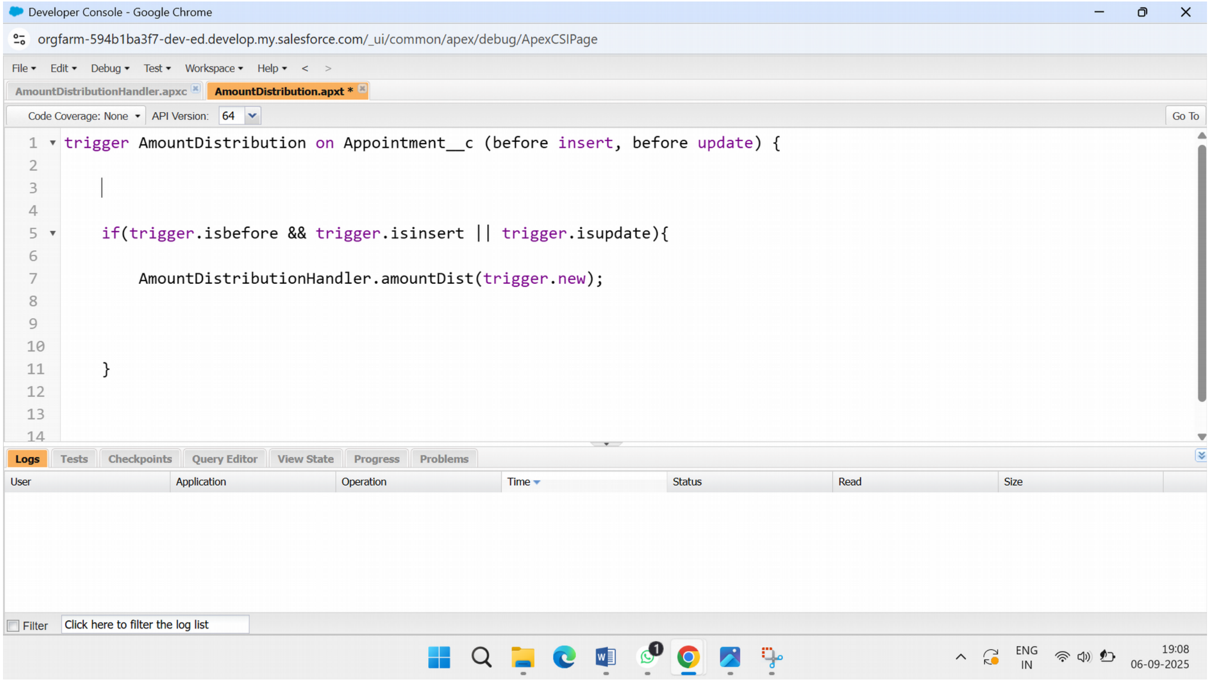


- Schedule class:

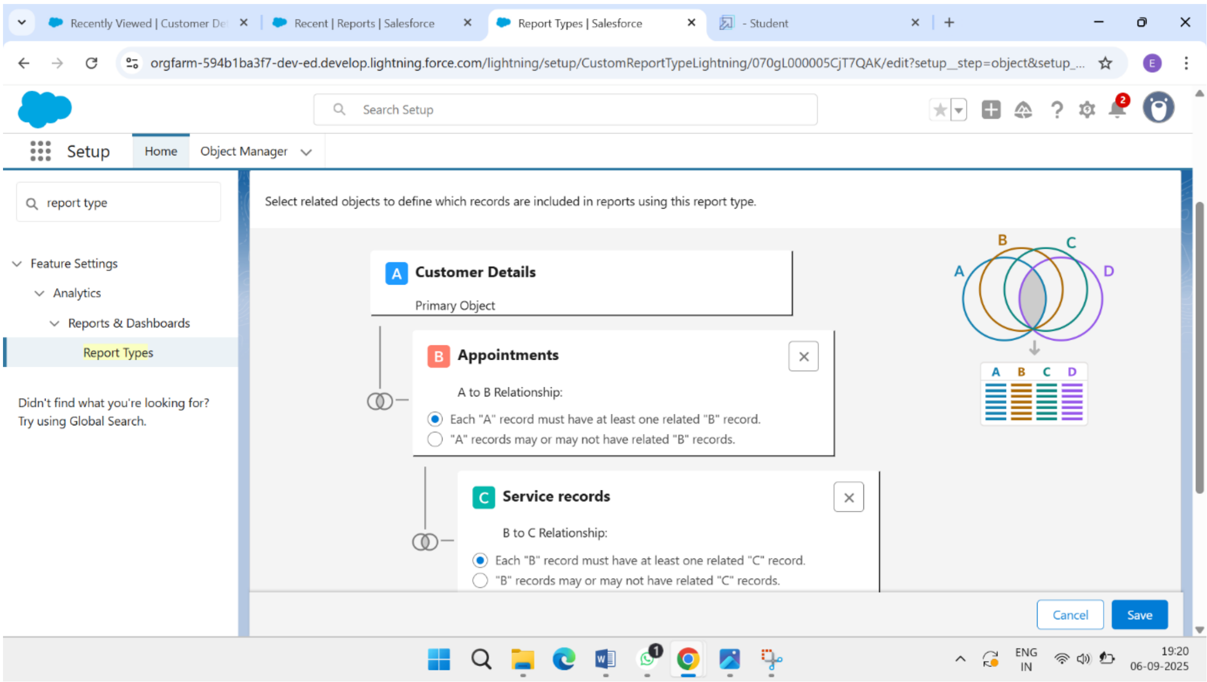
Create an Apex Class.

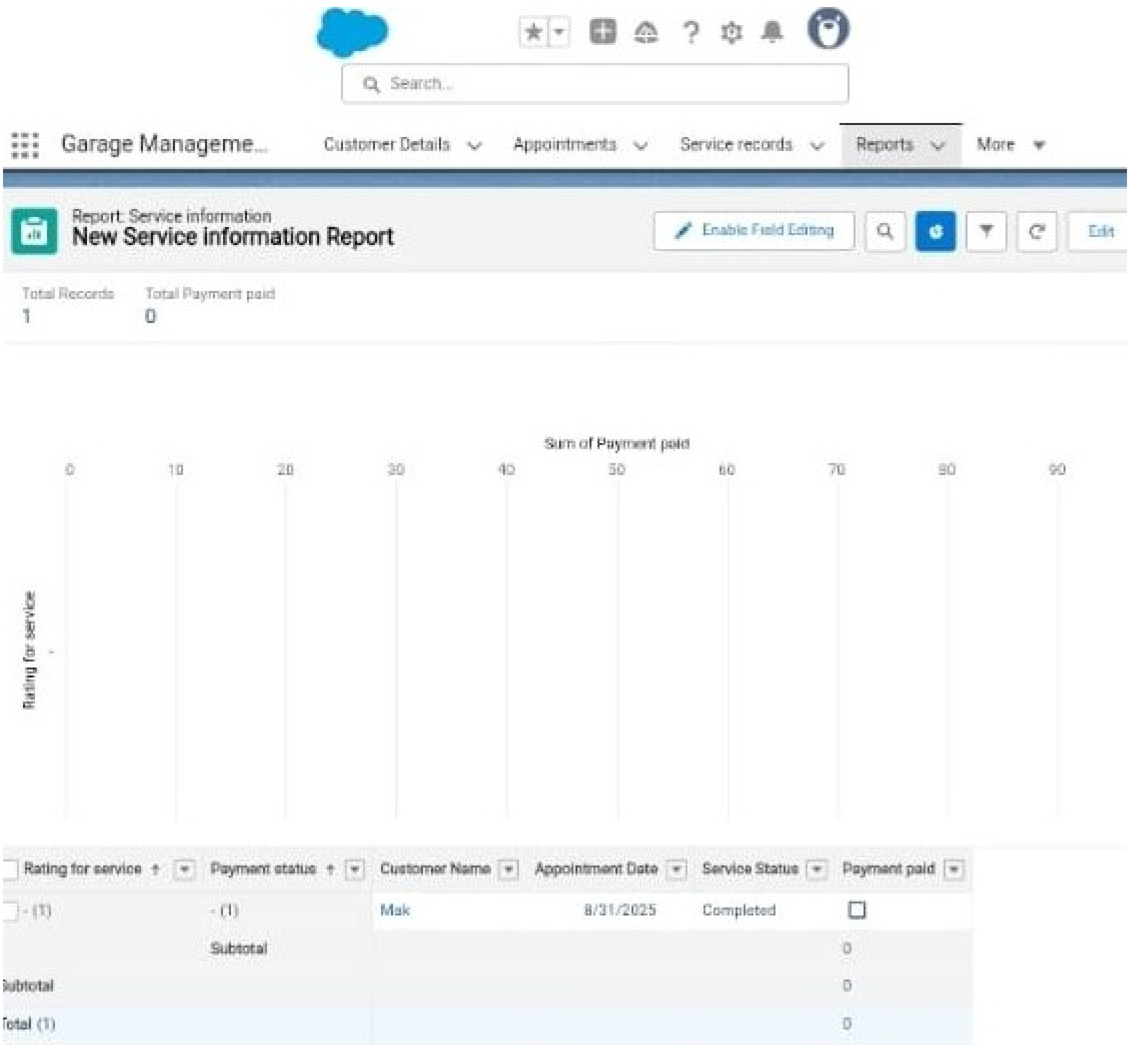


Schedule Apex Trigger.

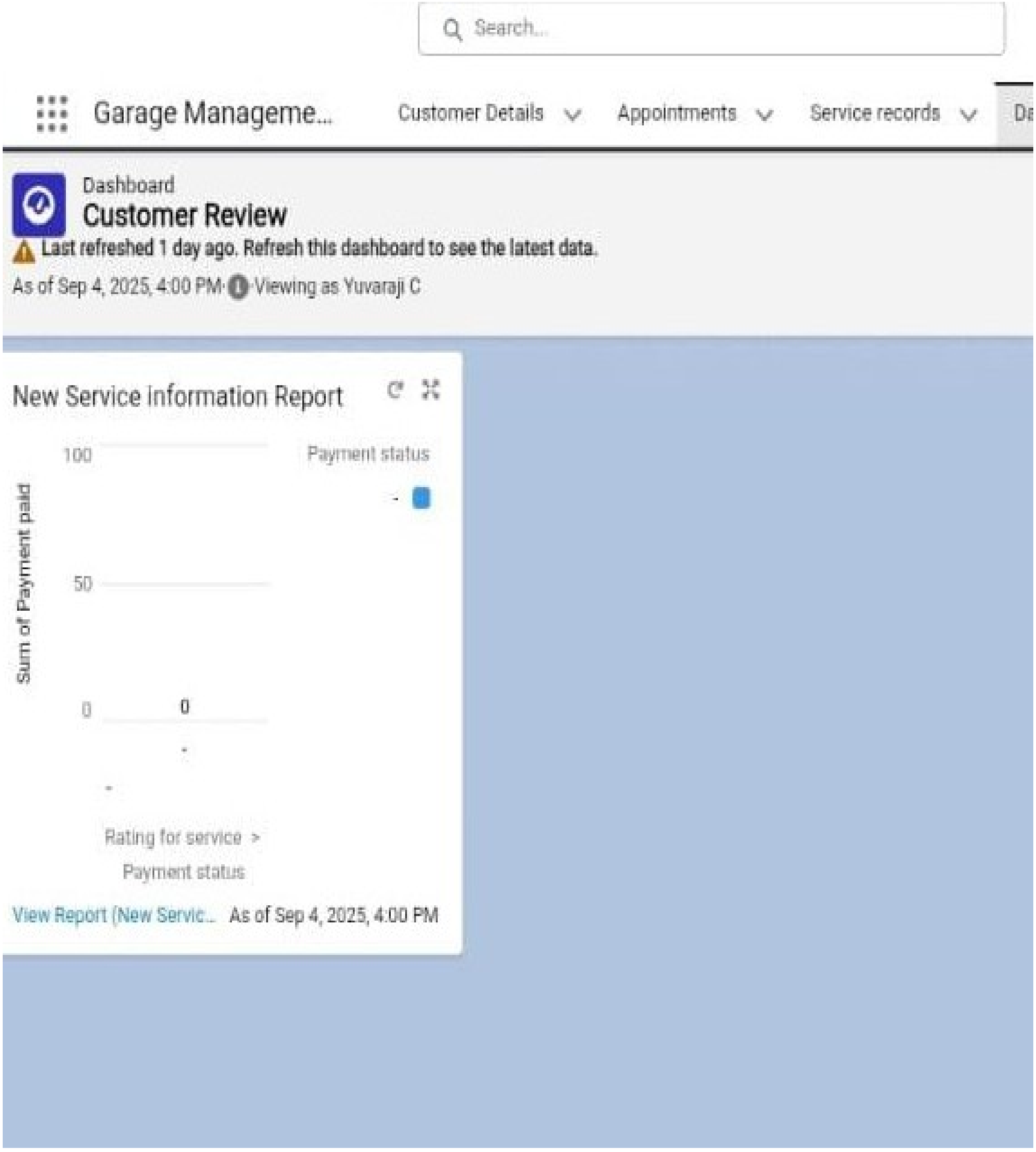


- To create the report folders.



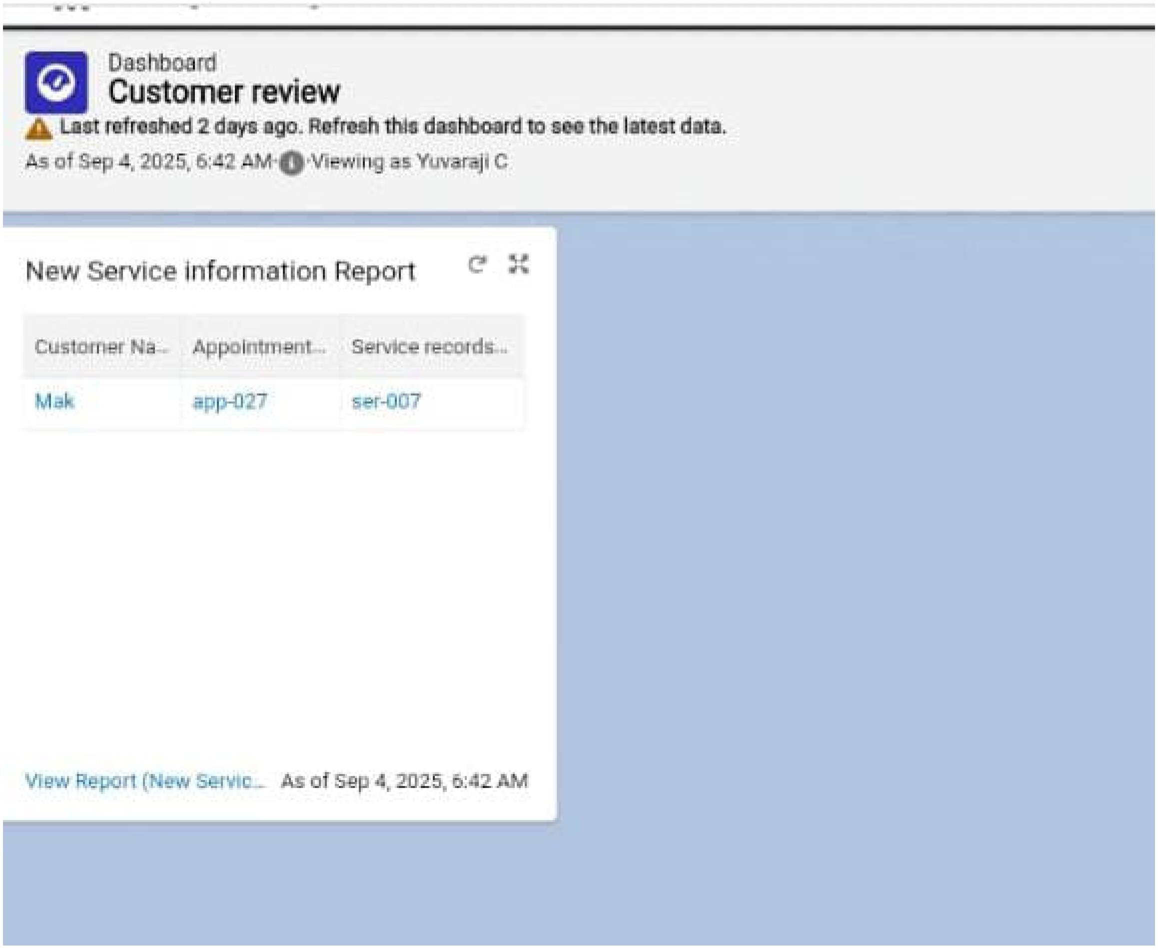


- The Dashboards is created.



- The Garage Management System is used the user adoption.

- Create a Customer detail, Appointment, Service record and Billing details and feedback.



3. ADVANTAGES AND DISADVANTAGES:-

ADVANTAGES:-

- ✓ Stores customer details, vehicle and service record in one place without errors.
- ✓ Provides service history, billing, and updates quickly to customers.
- ✓ Time saving.

- ✓ Efficiency improvement.
- ✓ Ensures safe storage of records compared to traditional registers.
- ✓ Helps management make better decisions using service and revenue reports.
- ✓ Inventory are controls.

DISADVANTAGES:-

- ✓ Garage operations may slow down if there are technical issues or downtime.
- ✓ If not properly secured, customer and vehicle data could be exposed.
- ✓ Initial cost.
- ✓ Technical knowledge required.
- ✓ Maintenance.
- ✓ Customer satisfaction.
- ✓ Regular software updates.
- ✓ Technical support are required.

4. FUNCTIONAL AND PERFORMANCE TESTING:-

- Functional testing makes sure the system features work correctly.
- Performance testing makes sure the system works efficiently under real-world usage and load.

CONCLUSION:-

The Garage Management System provides an efficient way to manage day-to-day garage operations by automating tasks such as Customer details, Appointment, Service record and Billing details and feedback. The system ensures that garage operations are faster, more reliable and customer-friendly, though it requires proper setup, training and maintenance.

APPENDIX:-

```
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 AmountDistribution on Appointment__c (before insert, before update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);

    }

}
```