ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES, KADAPA

Project Report on

SALESFORCE - Garage Management system

BY – B.Kavitha(22HM1A0508)

bkavitha9786@gmail.com

Table of Contents

1. PROJECT OVERVIEW

1.1 Overview

The Garage Management System is a valuable tool for automotive repair facilities, helping them deliver top-notch service, increase operational efficiency, and build lasting customer relationships. With its user-friendly interface and powerful features, GMS empowers garages to thrive in a competitive market while ensuring a seamless and satisfying experience for both customers and staff. A Garage Management System (GMS) in Salesforce is a cloud-based solution designed to manage all operations of a car garage or service center using Salesforce's CRM, automation, and cloud capabilities. It provides an integrated view of customer interactions, vehicle service history, job scheduling, and billing.
GMS empowers garages to:

- Provides access to add customers details.
- Automate service appointments.
- Vehicle service records update.
- Updates reports about the bills based on the records updates.
- Provide data-driven insights through reports and dashboards.

1.2 Key Features & Business Needs

1.2.1 Key Features

1.Customer & Vehicle Management:

- Store customer profiles with contact details.
- Link customers to multiple vehicles.
- Maintain vehicle details: make, model, VIN, year, registration number.

2.Service Booking & Appointment Scheduling:

- Book appointments via phone, web, or mobile app.
- View mechanic availability using calendar.
- Send automated SMS/email reminders.

3.Job Card & Work Order Management:

- Create job cards for each vehicle service.
- Assign mechanics or technicians to each job.
- Track job progress (Pending → In Progress → Completed).
- Attach parts used and technician notes.

4.Billing & Invoicing:

- Auto-generate invoices from job cards.
- Apply discounts, tax, labor charges.
- Integrate with payment gateways or external finance systems.

5.Service History Tracking:

- Maintain detailed records of each vehicle's past services.
- View previous repairs, replaced parts, and costs.

6.Reports & Dashboards:

- View mechanic performance, customer visits, service trends.
- Track daily/monthly revenue, popular services, inventory status.

7. Automation via Salesforce Flow:

- Auto-assign jobs based on mechanic skills.
- Trigger workflows for approvals or follow-up.
- Automatically update job status or notify managers.

1.2.2 Business Needs:

Business Need	How Salesforce GMS Solves It
Lack of visibility into operations	Dashboards and reports offer full transparency
Manual customer tracking	Centralized CRM for all customer and vehicle data
Overlapping appointments	Smart scheduling and availability calendars
Poor tracking of services performed	Job cards and service history maintain full records
Inventory loss or mismanagement	Real-time inventory tracking and low-stock alerts
Billing errors	Automated invoicing with tax/labor/parts calculation
Low customer retention	Personalized follow-ups, service reminders, loyalty offers
Inefficient technician workload	Auto-assignment based on workload and specialization
Limited access to data in field	Mobile app access for on-site technicians
Manual reporting	Real-time dynamic dashboards and scheduled reports

1.3 Technologies & Salesforce Tools Used:

- **Salesforce CRM**: Core customer and contact management.
- **Service Cloud**: Case tracking, knowledge base, technician management.
- **Salesforce Flow**: Automating repair approvals, notifications, and job assignments.
- **Lightning App Builder**: UI customization for service advisors or mechanics.
- **Custom Objects**: For Vehicle, Job Card, Part Inventory, etc.
- **Reports & Dashboards**: Visual performance tracking.
- **Mobile App (Salesforce1)**: Field service mechanics can update job status via mobile.

1.4 Benefits:

- Real-time visibility of garage operations.
- Improved customer experience with timely updates and reminders.
- Efficient management of service workflow and inventory.
- Centralized data for all branches or service centers.
- Customizable and scalable as garage expands.

2.OBJECTIVES

2. Objectives

1.Improve Customer Satisfaction: Deliver timely updates, service reminders, and digital invoices to enhance customer trust and retention.

2.Automate Garage Operations: Streamline key tasks like appointment booking, job card creation, billing, and notifications using automation tools in Salesforce.

3.Track Vehicle and Service History: Maintain detailed records of each vehicle's past services, repairs, and part replacements for accurate diagnostics.

4. Manage Inventory Effectively: Monitor spare part stock levels, usage, and reorder needs to avoid service delays and reduce waste.

5. Generate Accurate Invoices: Auto-generate invoices based on parts used, labor, and taxes, reducing billing errors and manual effort.

6. Provide Real-Time Reports: Use dashboards and reports to monitor garage performance, revenue, customer visits, and inventory trends.

3. REQUIREMENT ANALYSIS & PLANNING

3.1 Understanding Business Requirements

Current System Gaps:

1.Data Duplication:

- No standardized duplicate rules for critical objects (e.g., customers, vehicles).
- Manual checks required (matches your "Duplicate Rule" screenshot).

2.Access Control Issues:

Roles/Profiles (shown in your UI) lack granular permissions for:

- Technicians (need edit access to Work_Order_c but not Invoice_c).
- Public Groups not optimally used for sharing records.

3.Process Inefficiencies:

Flows/Triggers (referenced in your screenshots) not fully automating:

- Appointment reminders.
- Inventory reorder alerts.

3.2 Defining Project Scope & Objectives

Project Scope

Included Components:

The Garage Management System (GMS) will implement the following Salesforce components:

- **Custom Objects:** Vehicle_c, Service_Booking_c, and Inventory_c with defined relationships to centralize customer, service, and parts data.
- **Duplicate Rules:** Prevent duplicate customer entries (matching on email/phone) and vehicle records (matching on VIN) to eliminate manual cleanup efforts.
- **Role Hierarchy:** A three-tier structure (Admin > Service Manager > Technician) to enforce approval workflows, such as discounts exceeding 15%.
- **Automation:** Flows for auto-assigning technicians and sending SMS/email appointment reminders to reduce no-shows.
- **Validation Rules:** Enforce mandatory fields like Mileage_c during service bookings to ensure data accuracy.
- **Reports & Dashboards:** Real-time tracking of daily appointments, revenue, and inventory levels for data-driven decisions.

Objectives

1. **Operational Efficiency:** Reduce appointment scheduling time by 50% (from 10 minutes to 5 minutes) through self-service portals and automated workflows. Track progress via time-tracking reports.

2. **Customer Experience:** Cut no-show rates by 25% using automated reminders (SMS/email) triggered 24 hours before appointments. Measure success through attendance analytics.

3. **Inventory Optimization:** Decrease stockouts by 40% by implementing real-time inventory alerts and reorder triggers. Monitor results via inventory turnover reports.

4. **Compliance:** Ensure 100% audit trails for financial transactions (SOX compliance) using Salesforce's built-in audit logging.

5. **Maintain Security and Control:** Implement Salesforce role hierarchy, profiles, and sharing rules to enforce secure access to system data.

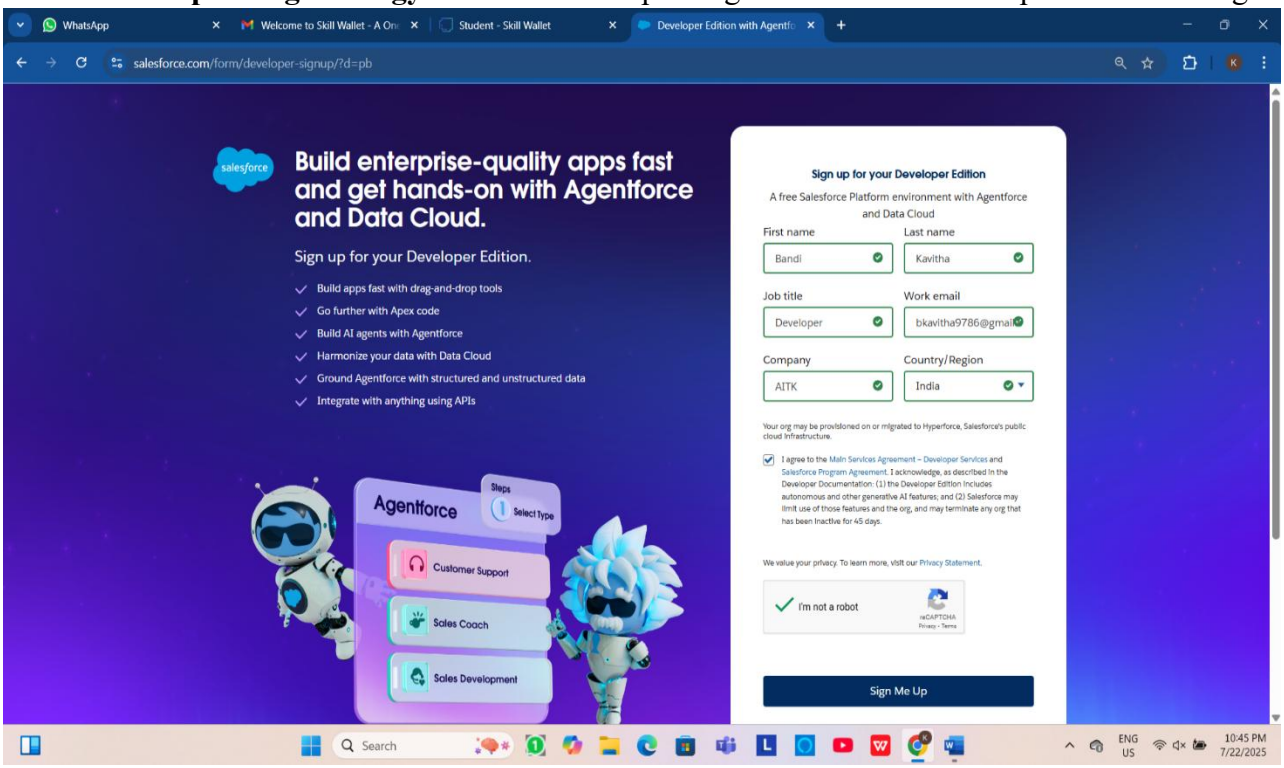
6. **Enable Better Decision-Making:** Use reports and dashboards to analyze daily operations, revenue, stock levels, and employee performance.

4.SALESFORCE DEVELOPMENT

4.1 Set Environment

We established a robust Salesforce development environment using:

- **Developer Org Strategy:** Created Developer Org sandboxes for development and testing.



Link for creating the developer org https://developer.salesforce.com/signup

Fig: Signing up for a developer Org

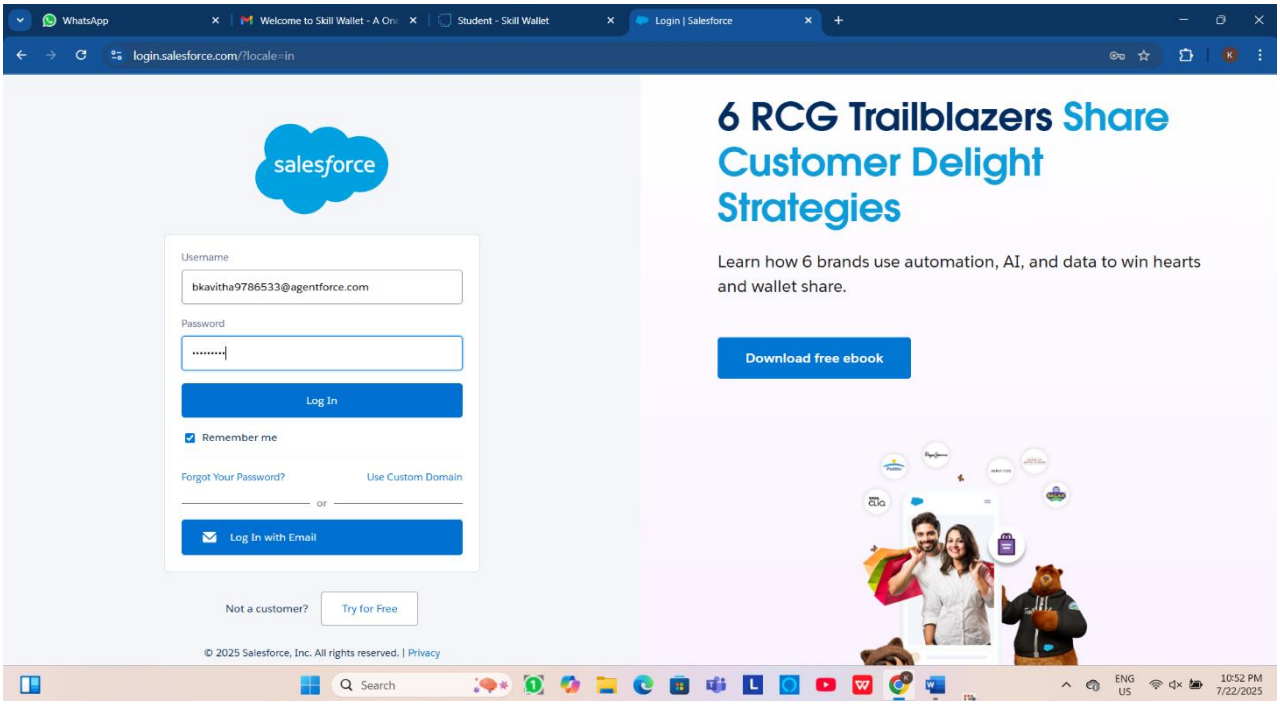


Fig: Developer Org Login

4.2 Customizations & Automation

4.2.1 Core System Customizations:

- **Object:**
  Salesforce objects are database tables that permit you to store data that is specific to an organization.

Create Customer DetailsObject:

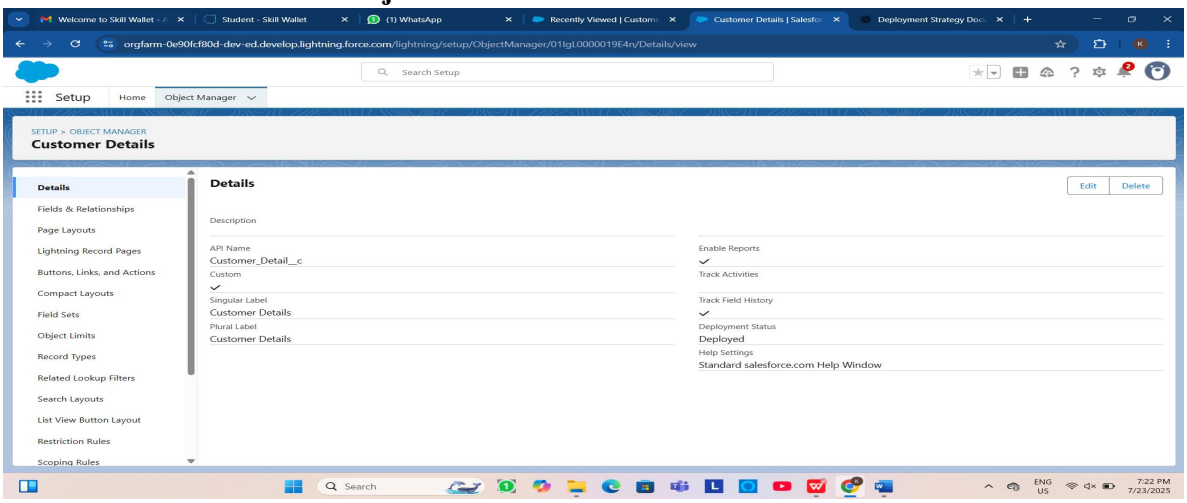


Fig:Customer Object

Create Appointment Object

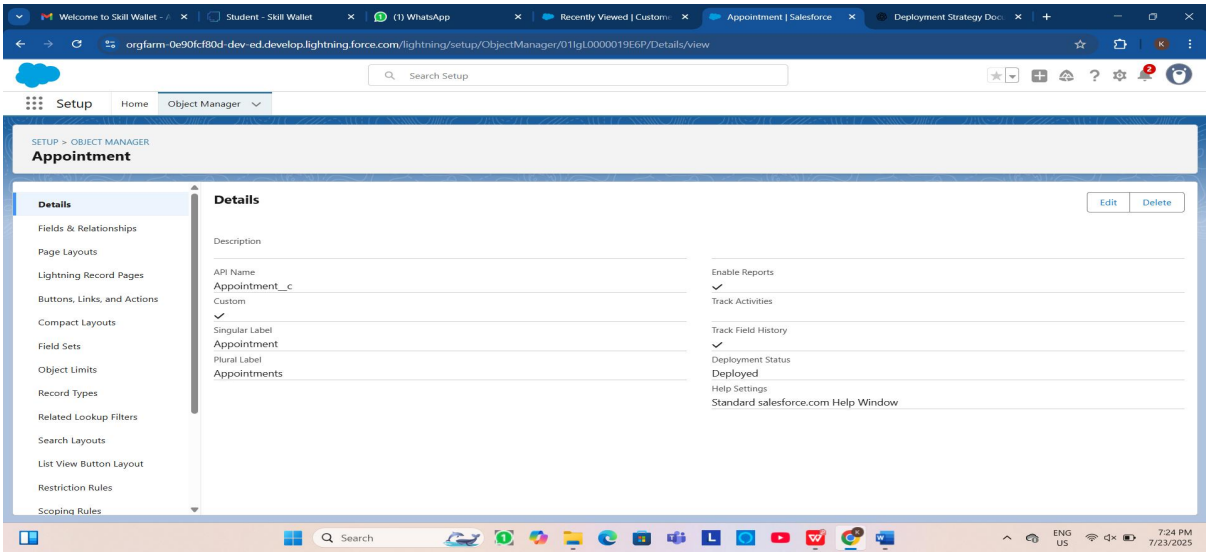


Fig:Appointment Object

Create Service records Object

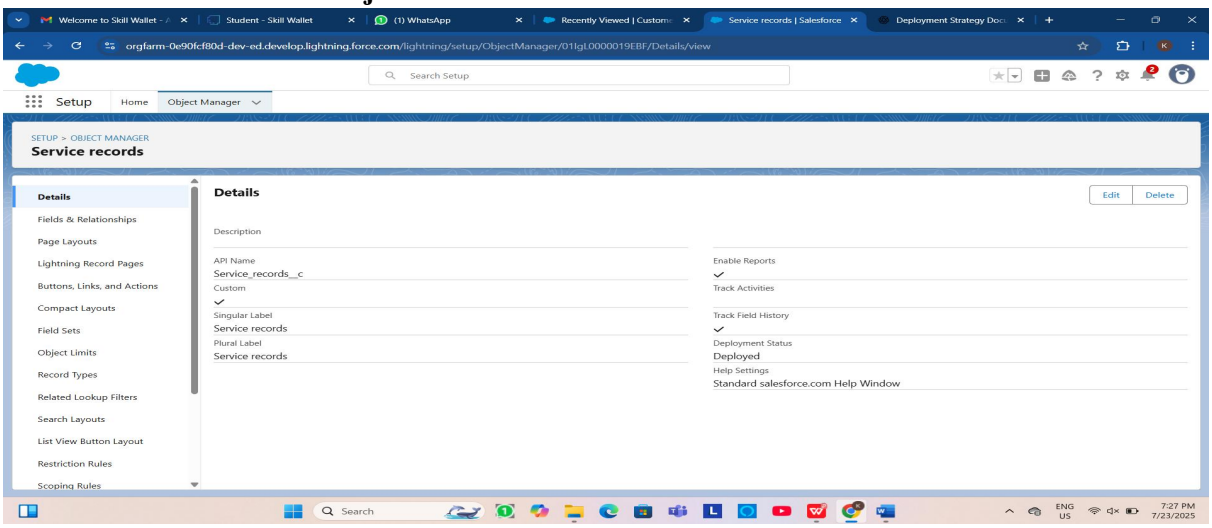


Fig:Service Object

Create Billing details and feedback Object

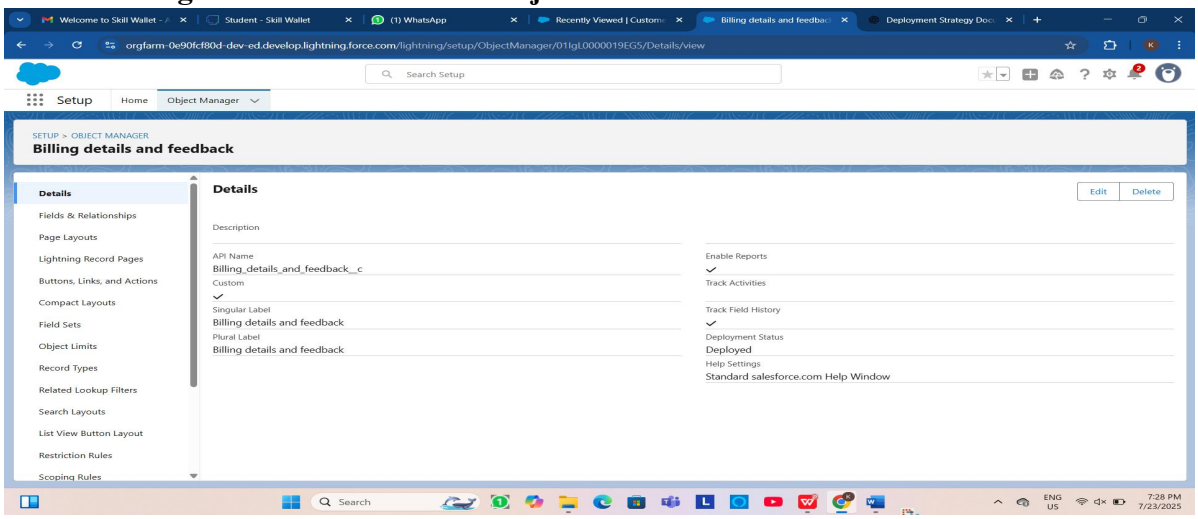


Fig:BD&F object

- **Fields:**

Fields represent the data stored in the columns of a relational database. It can also hold any valuable information that you require for a specific object. Hence, the overall searching, deletion, and editing of the records become simpler and quicker.

Creation of fields for the Customer Details object:

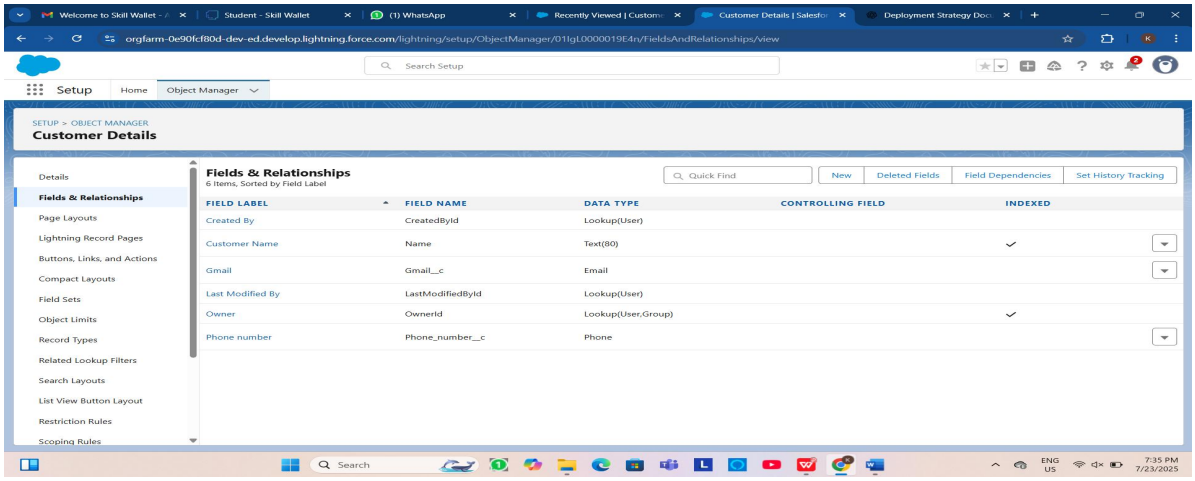


Fig:fields of Customer

Creation of fields for the Appointment Object:

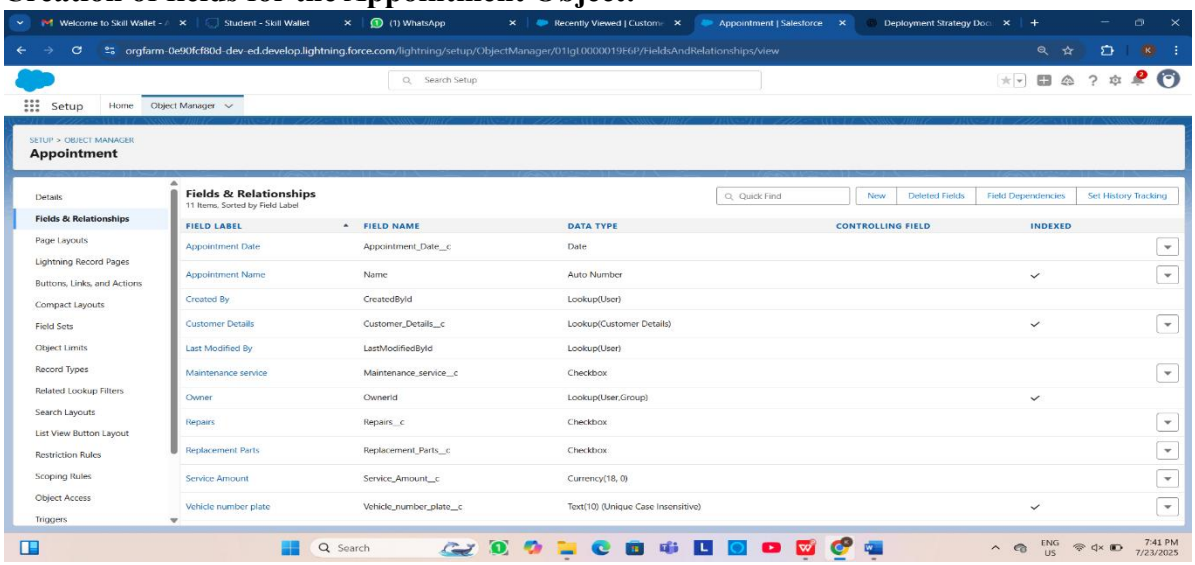


Fig:fileds of Appointment

Creation of fields for the Service Records Object:

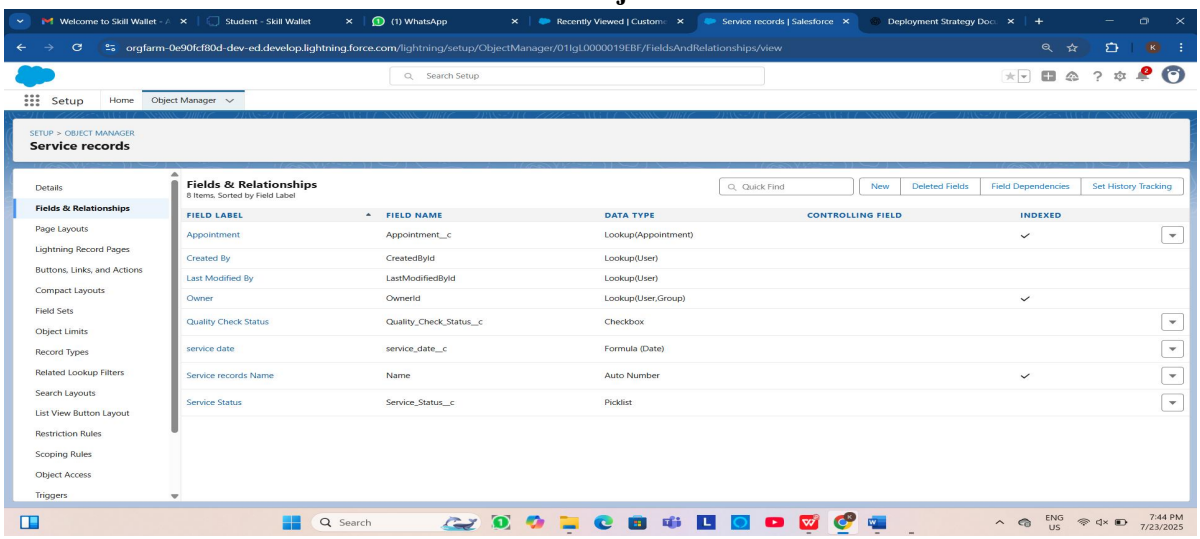


Fig: fields of Service

Creation of fields for Billing details and feedback Object:

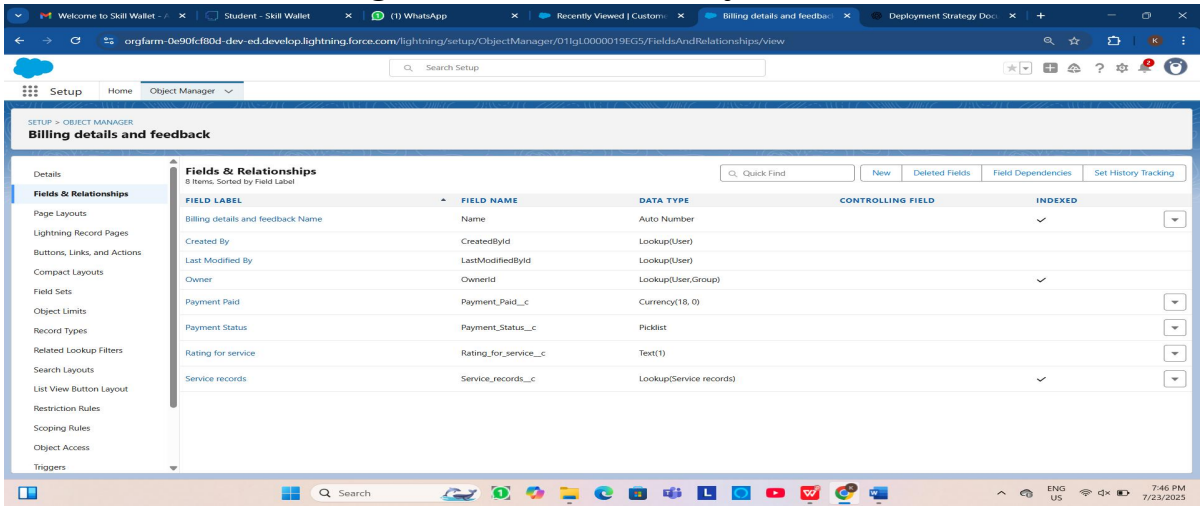


Fig:fields of BD&F

- **Validation Rules:**

Creation of Validation Rules to Appointment Object:

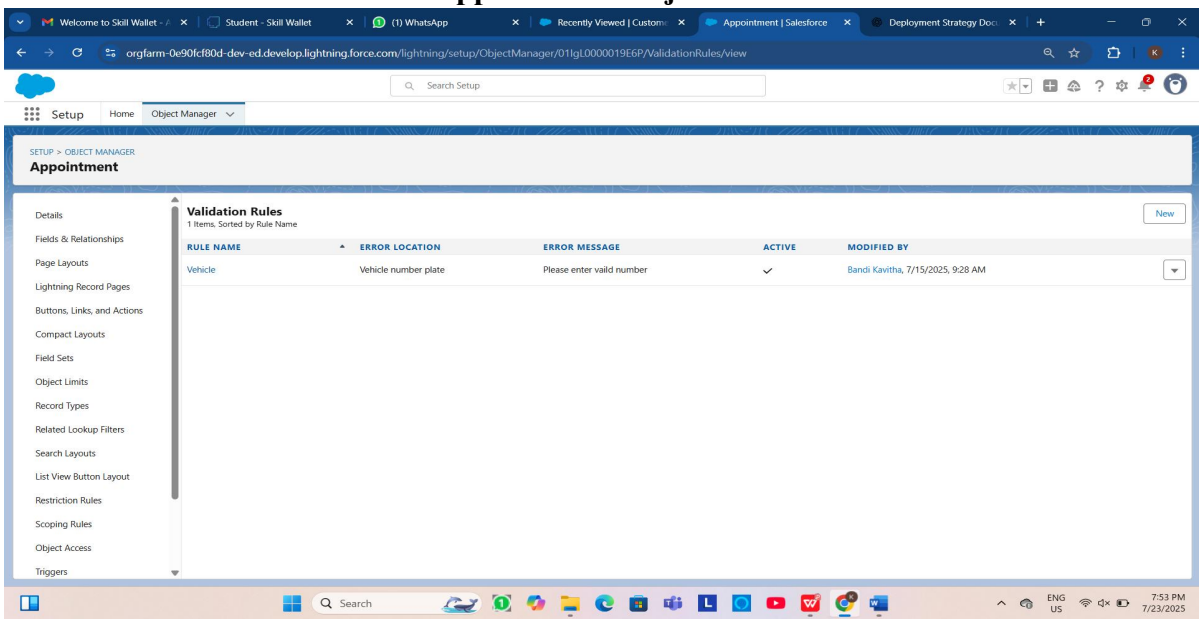


Fig : Validation rules of Appointment

Creation of Validation Rules to Billing details and feedback Object:

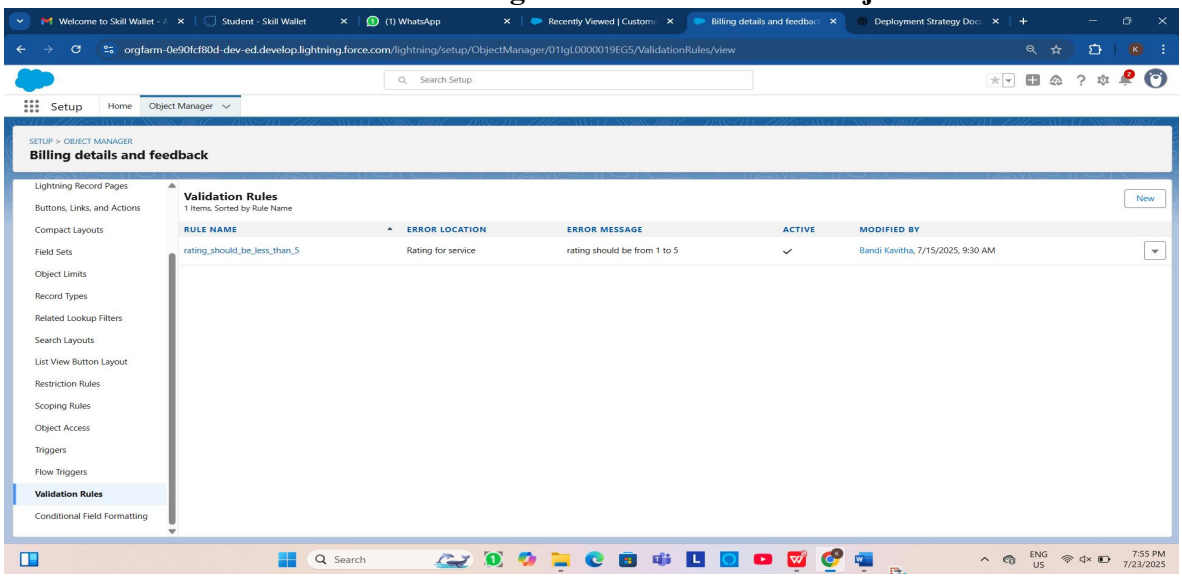


Fig : Validation rules of BD&F

4.3 Automation Components

4.3.1 Flows

In Salesforce, a flow is a powerful tool that allows you to automate business processes, collect and update data, and guide users through a series of screens or steps. Flows are built using a visual interface and can be created without any coding knowledge.

- Billing amount flow is created to send an email alert. Whenever the payment status in Billing details and feedback record is updated as completed for a particular service records the flow automatically sends an email alert as Thank You for Your Payment - Garage Management.

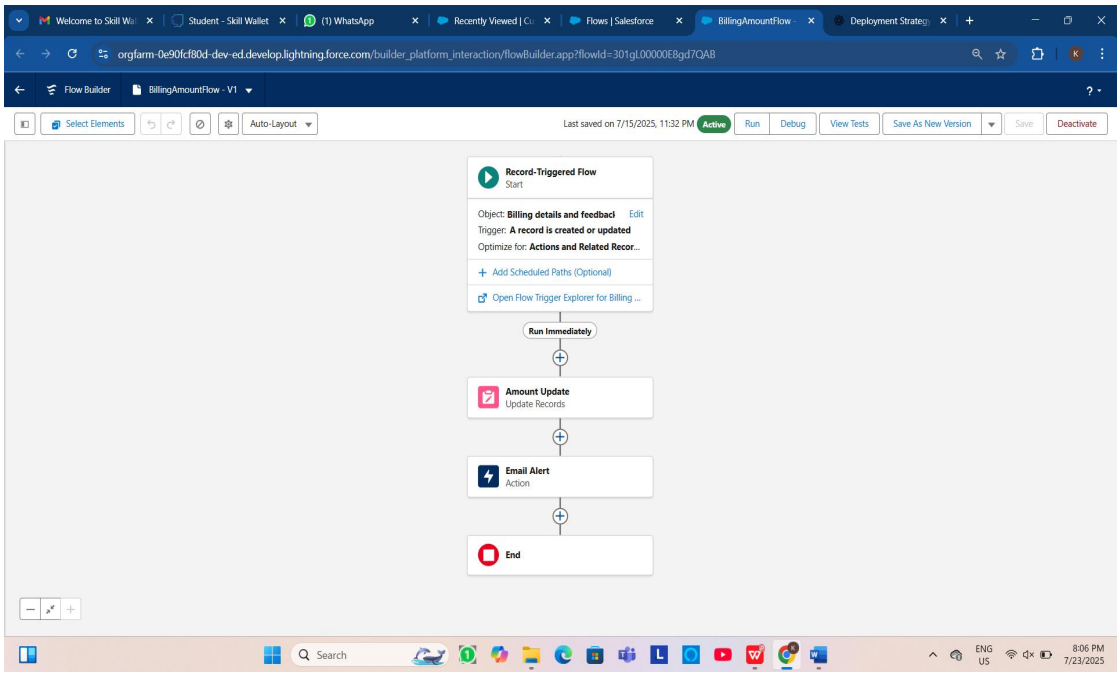


Fig: BillingAmountFlow

- The Update service status flow is designed for a purpose of updating the service status as completed when the quality service checkbox is selected when editing the service records.

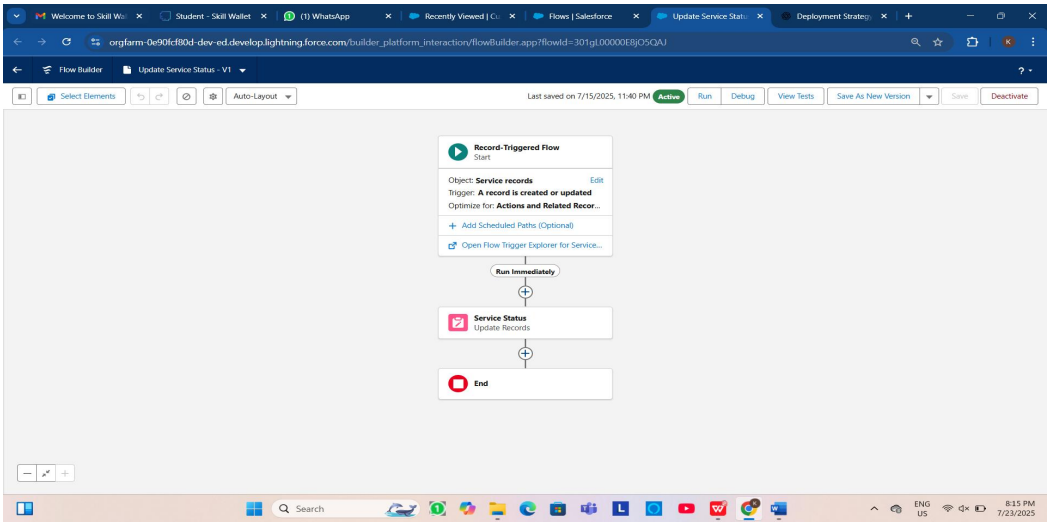


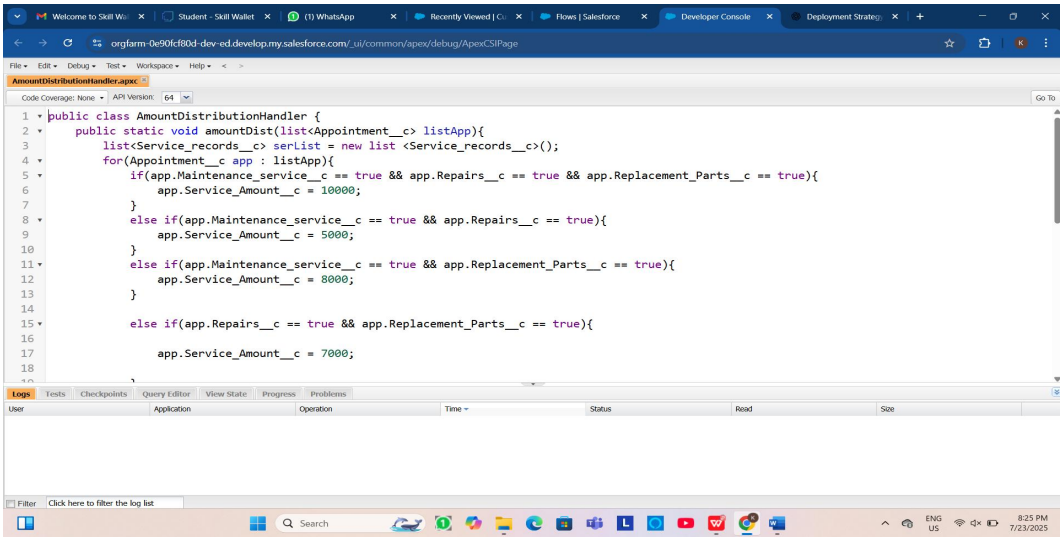
Fig:UpdateServiceStatus Flow

4.3.2 Apex Development:

- **Apex Handler:**

This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as “AmountDistributionHandler ”.



```
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;
```

Fig: Apex Handler

- **Trigger Handler:**

How to create a new trigger :

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment__c

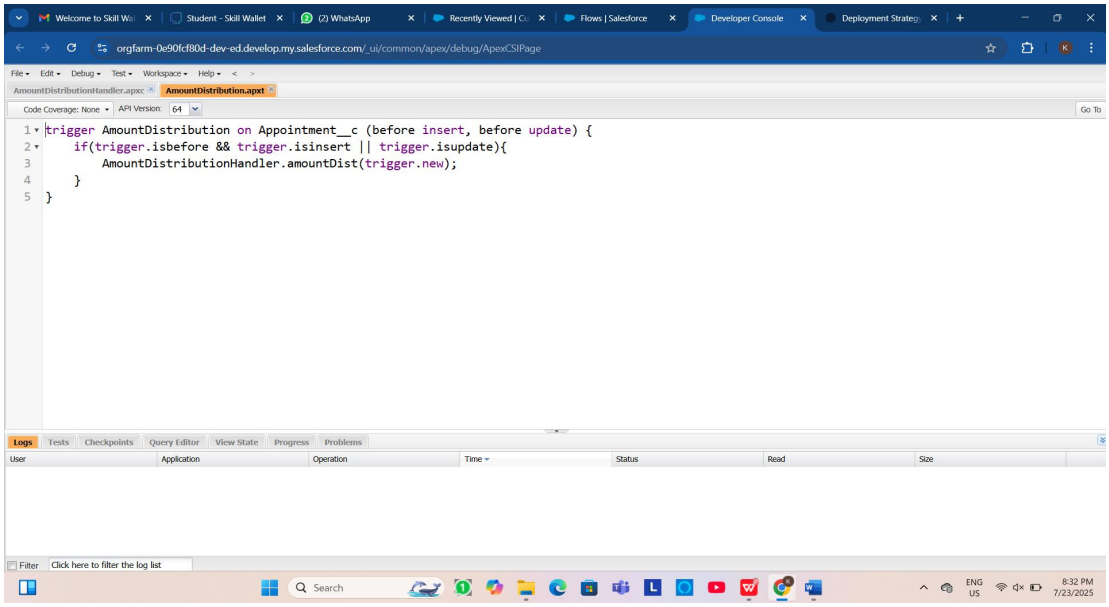


Fig: Trigger Handler

5.DATA MIGRATION,TESTING & SECURITY

- **Duplicate Rules:**

Duplicate Rules in Salesforce are used to prevent or manage the creation of duplicate records, such as leads, contacts, accounts, or custom objects.

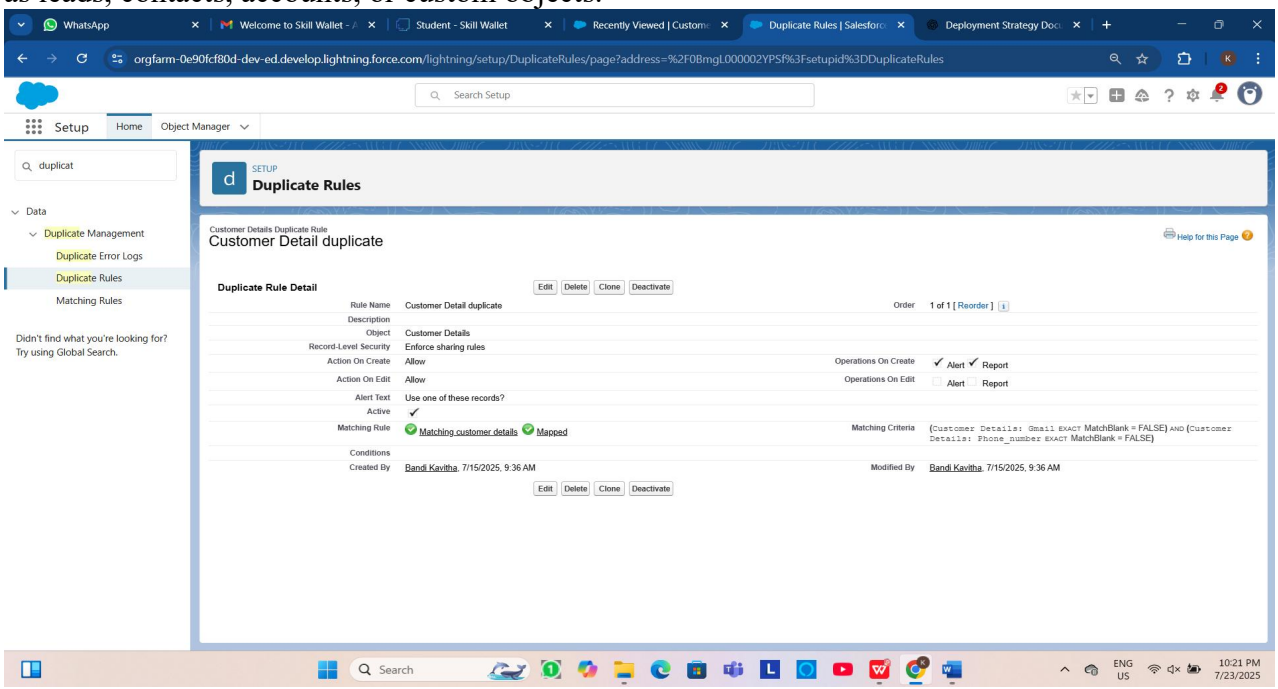


Fig:Duplicate Rules of Customer Detail

- **Matching Rules:**

Matching Rules in Salesforce define how records are compared to determine if they are duplicates. They are used in combination with Duplicate Rules to identify potential duplicate records based on specific fields and logic.

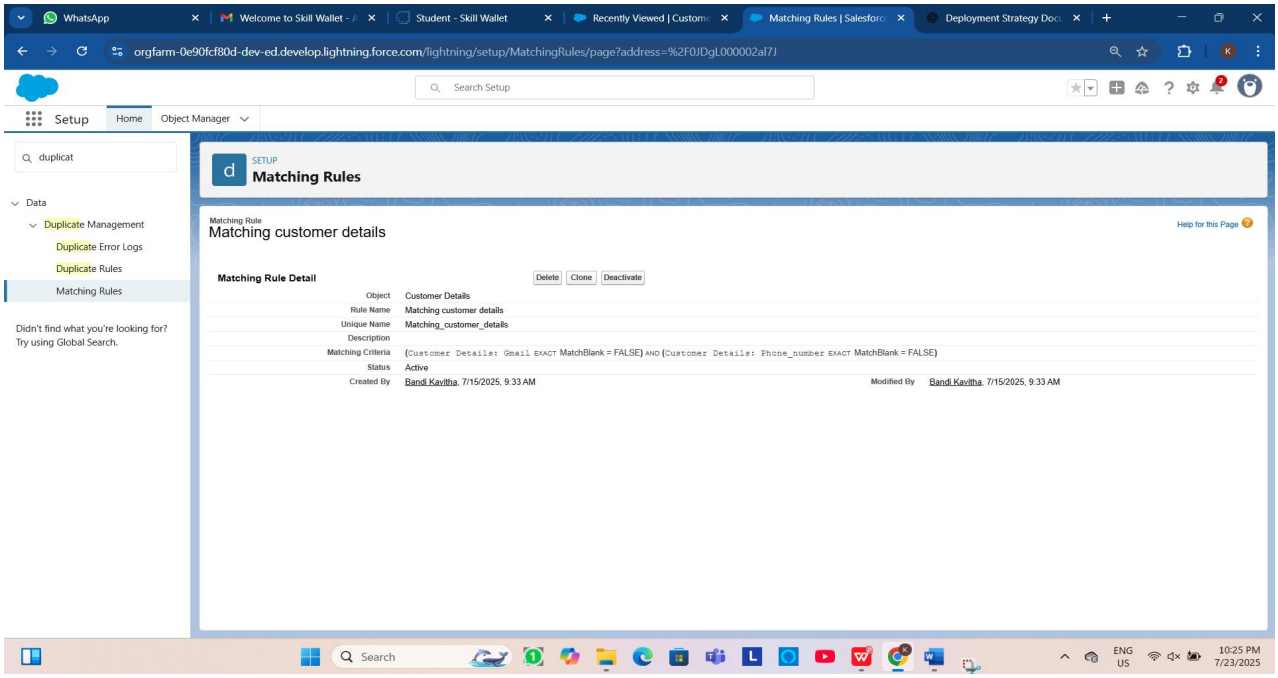


Fig:Matching Rules

- **Profiles:**

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

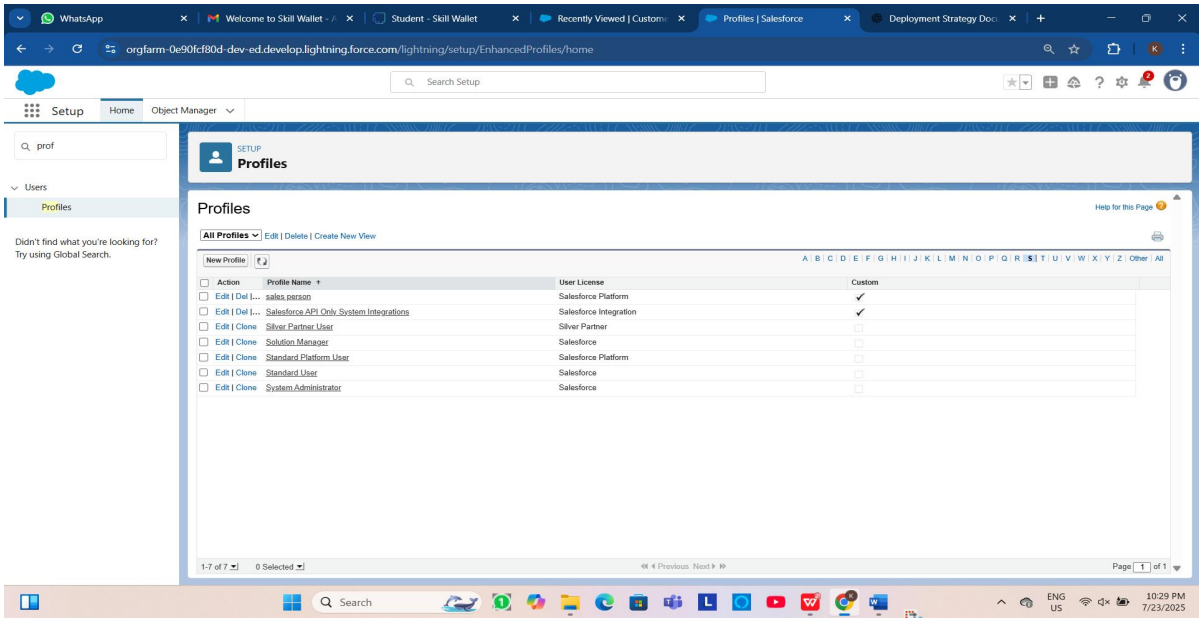


Fig:Profiles

- **Roles & Role Hierarchy:**

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data.

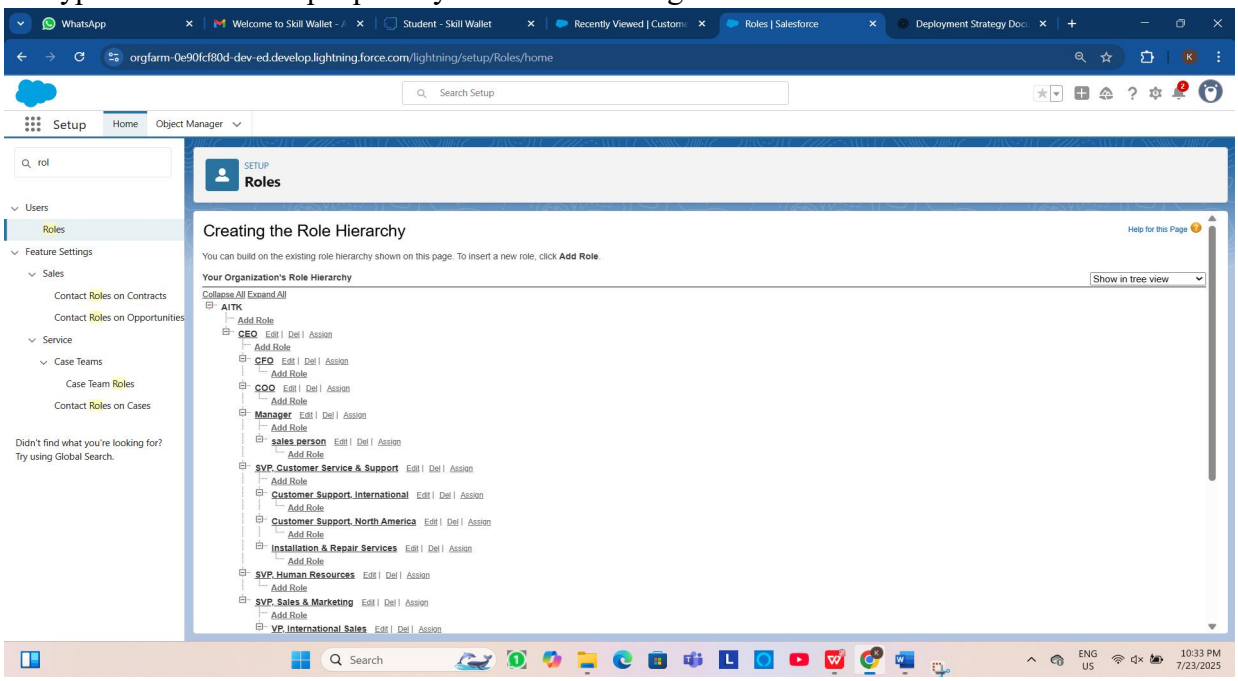


Fig:Roles & Role Hierarchy

- **Sharing Rules:**

Sharing rules are used to extend access to records for users who meet specific criteria.
They can be used to grant read-only or read-write access to records owned by other users.

Manual Sharing:

Administrators and record owners can manually share specific records with other users or groups.

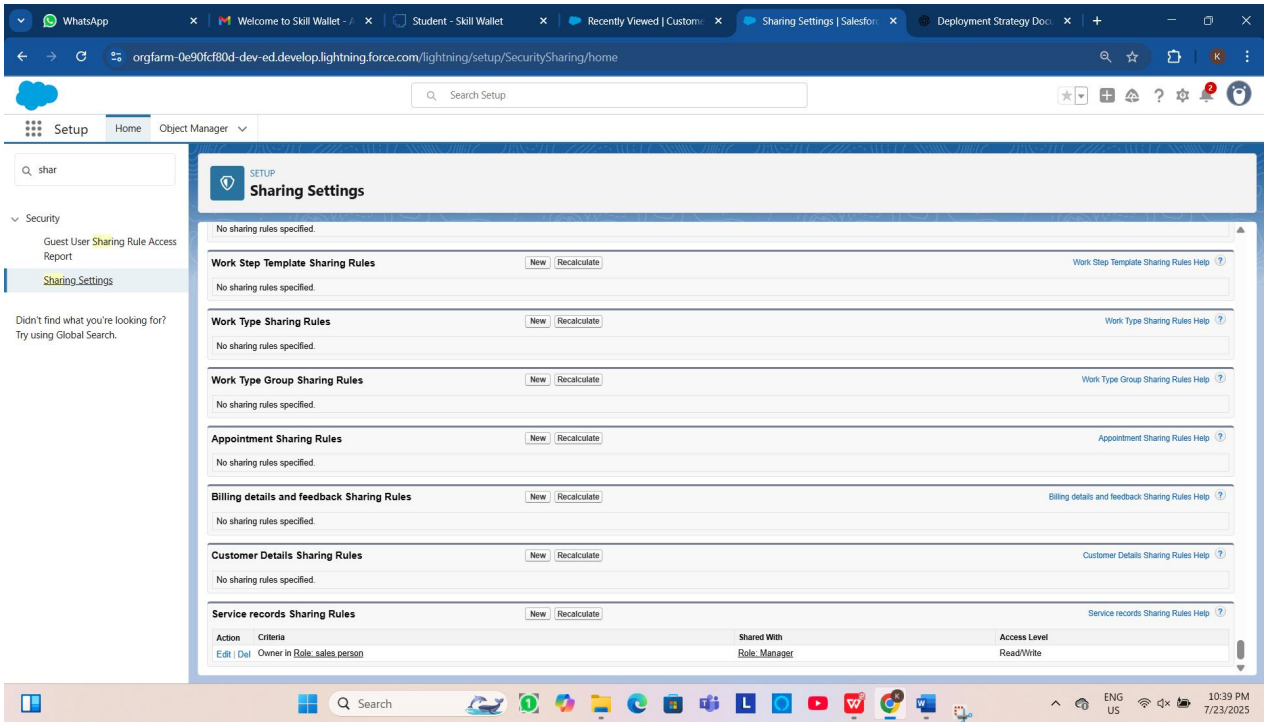


Fig: Sharing rules

6.KEY FEATURES AND FUNCTIONALITIES

The Garage Management System's core functionality suite delivers a comprehensive set of tools designed to revolutionize garage operations management, leveraging Salesforce's robust automation capabilities.

6.1 Work Order Management(Appointment & Service Records)

The system meticulously manages the lifecycle of a service request:

- Appointment Scheduling: Allows customers to book appointments, capturing essential details like customer information, desired service types (Maintenance, Repairs, Replacement Parts), and vehicle number plates. The Appointment Date is a crucial field, ensuring all necessary information is collected upfront.
- Dynamic Service Amount Calculation: Based on the services selected during appointment creation (Maintenance, Repairs, Replacement Parts), an Apex Trigger dynamically calculates and populates the Service Amount field. This ensures accurate upfront estimates.
- Service Execution Tracking: Once an appointment is confirmed, a Service record is created, automatically assigned a unique ser-{000} ID. The Service Status defaults to 'Started'.
- Quality Control Checkpoints: The Quality Check Status checkbox on the Service records object allows technicians to confirm critical quality checks have been performed.
- Automated Status Update: Upon marking Quality Check Status as true, the Service Status automatically updates to 'Completed', providing real-me visibility into service progression.
- Lookup Filter for Service Records: A validation on the Appointment lookup ensures that the Appointment Date for a service record is logically less than the Service records Created Date, maintaining data integrity.

6.2 Customer Management

- Centralized Customer Profiles: The Customer Details object acts as a single source of truth for all customer information, including contact details (Phone number, Gmail) and a unique Customer Name.
- Relationship Tracking: All appointments, service records, and billing details are linked back to the Customer Details, providing a comprehensive view of a customer's history with the garage. This enables personalized service delivery and proactive communication.

6.3 Billing & Feedback

- Automated Billing Integra on: The Billing details and feedback object captures payment information. The Payment Paid field is designed to be populated automatically via a Flow when the Payment Status is 'Completed', drawing the amount from the related Service Amount on the Appointment.
- Payment Status Tracking: A picklist field (Payment Status) allows for clear tracking of billing states ('Pending', 'Completed').
- Customer Feedback Collection: The Ra ng for service (1-5) field on the Billing details and feedback object provides a direct mechanism for customers to rate their experience, enabling continuous service improvement.

6.4 Dynamic Service Pricing(Apex)

The AmountDistributionHandler Apex class, triggered before insert and before update on the Appointment_c object, implements the business logic for calculating the Service_Amount_c based on the selected services:

- Logic:
    - Maintenance, Repairs, and Replacement Parts: $10,000
    - Maintenance and Repairs: $5,000
    - Maintenance and Replacement Parts: $8,000
    - Repairs and Replacement Parts: $7,000
    - Maintenance Service only: $2,000
    - Repairs only: $3,000
    - Replacement Parts only: $5,000

This ensures that the service amount is dynamically calculated and displayed to the customer based on their selections.

6.5 Automated Email Notifications(Flow)

A Record-triggered Flow on the Billing details and feedback object automates customer communication:

- Trigger: When a Billing details and feedback record is Created or Updated.
- Condition: Executes only when Payment_Status_c is 'Completed'.
- Action:
    1. Update Records: Sets the Payment_Paid_c field on the Billing details and feedback record to the value of Service_Amount_c from the related Appointment_c record.
    2. Email Alert: Sends a personalized "Thank You for Your Payment" email to the customer using their Gmail_c from the Customer Details record. The email includes the customer's name and the Amount paid. This enhances customer experience and provides automated confirmation.

7. UI/UX DEVELOPMENT & CUSTOMIZATION

7.1.Lightning App Setup

7.1.1 Customer App Configuration

Created "Garage Management Application" Lightning App for accessing all objects, reports and dashboards as follows:

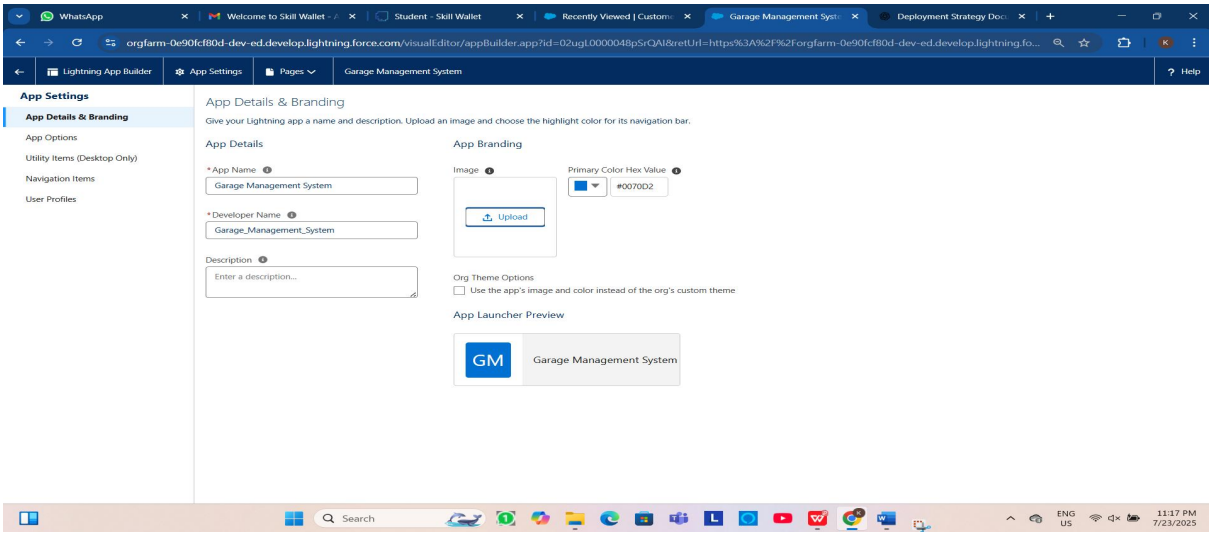


Fig:Lightning app Structure

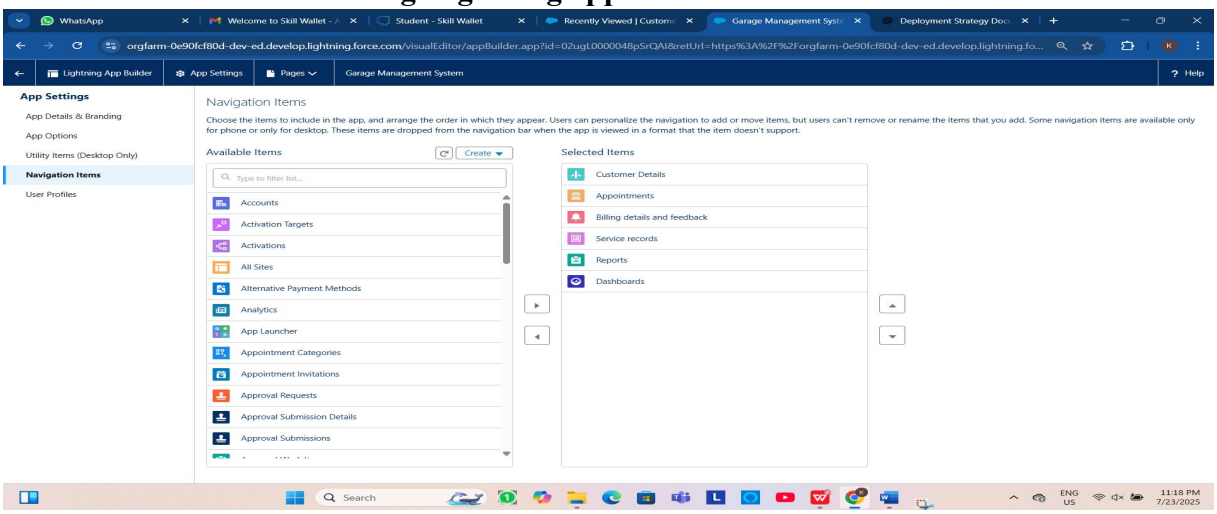


Fig:Object under Garage Management System Application

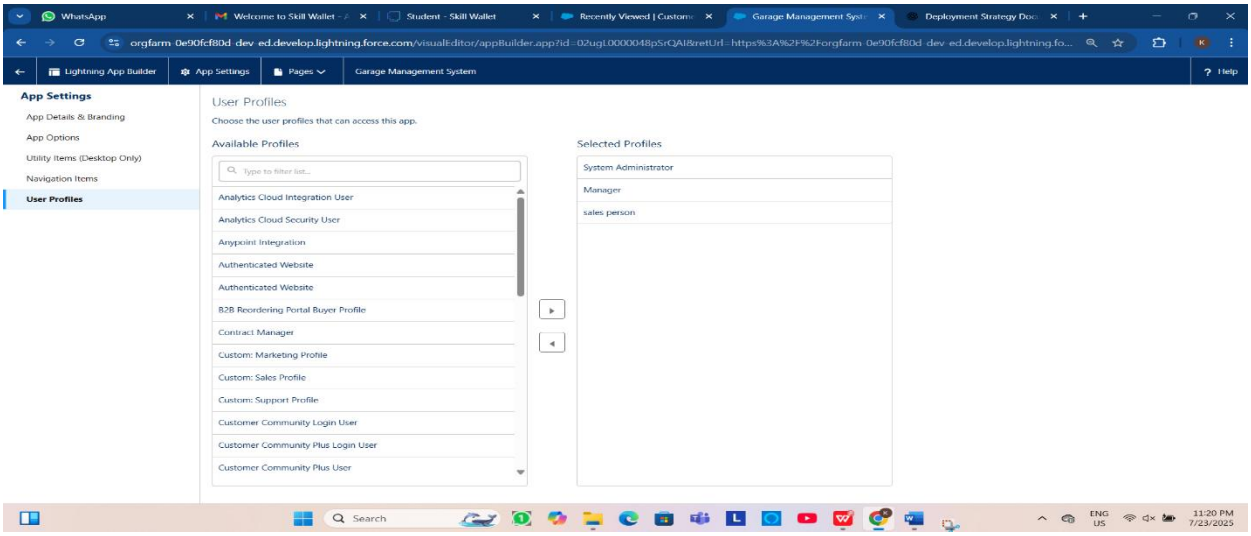


Fig:Users Who Access the Application

7.2 Reports & Dashboards

7.2.1 Reports

The “New service information Report” is created to display the services payments details summary and a chart representing the ratio between the rating given by customers while payment and number of payments rated same.

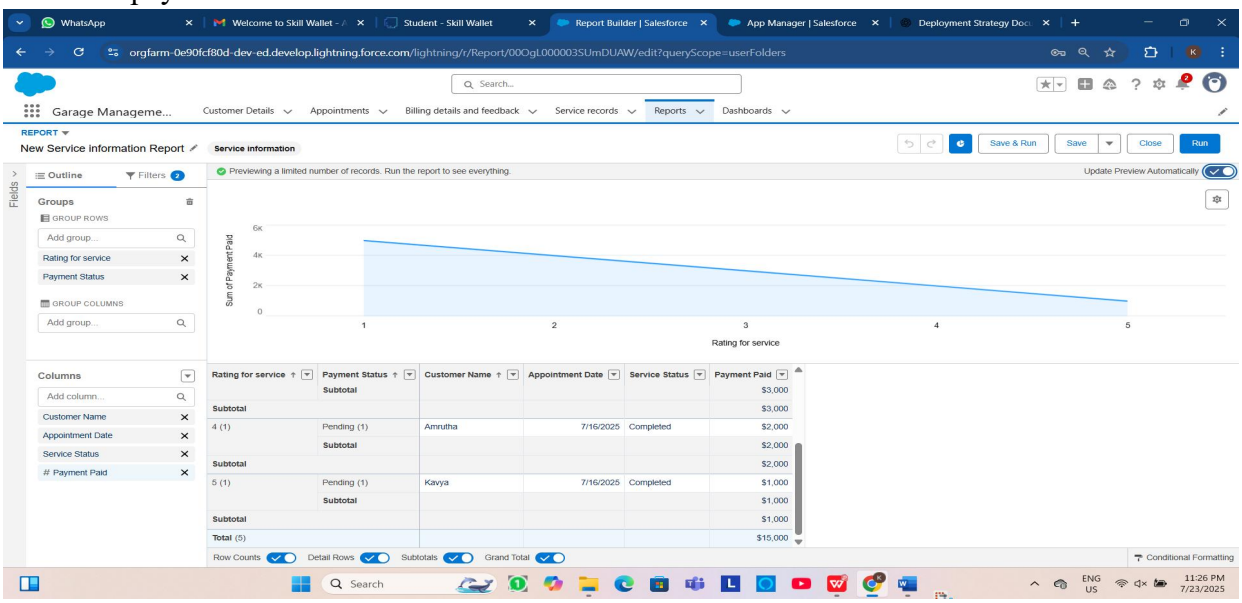


Fig: Reports

7.2.2 Dashboards

The “Customer Review” dashboard is as similar as report without the other data rather than line chart. The line chart of “New service information report” is displayed as a widget in the dashboard.

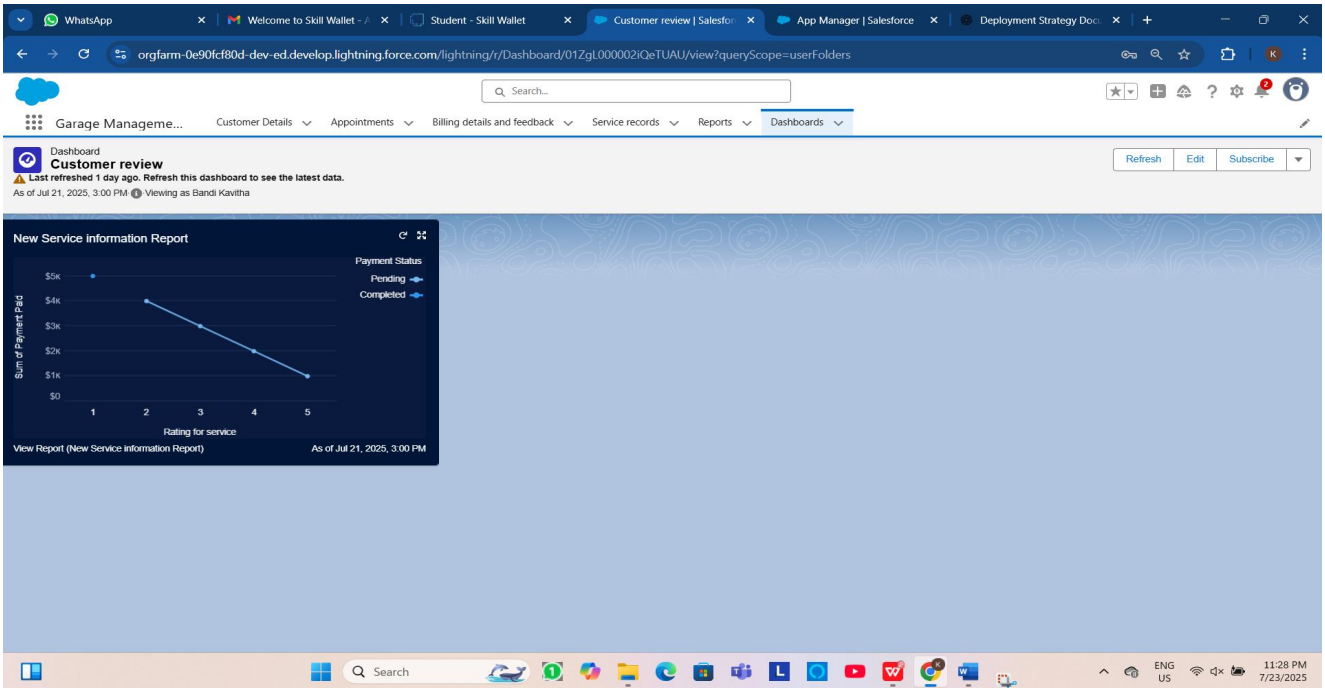


Fig:Dashboards

8.DEPLOYMENT,DOCUMENTATION & MAINTENANCE

8.1 Deployment Strategy

The deployment strategy will depend on the scale and complexity of the solution. For the Garage Management System in Salesforce, the following approaches can be used**:**

a. Change Sets (for Sandbox to Production):

- **Use Case:** Best for small-to-medium-sized orgs.
- **Steps:**
    - Create an outbound change set in sandbox.
    - Add components (Apex classes, triggers, objects, etc.).
    - Upload to production org.
    - Deploy after validation.
- **Pros**: Native to Salesforce, easy to track.
- **Cons:** Doesn't support all metadata (e.g., profiles, standard value sets).

b. Salesforce CLI + Metadata API (SFDX):

- **Use Case:** Suitable for CI/CD and version control integration.
- **Steps:**
    - Use Salesforce DX commands to retrieve, deploy, and test components.
    - Integrate with Git for source control.
    - Deploy using CI/CD tools (e.g., Jenkins, GitHub Actions).
- **Pros:** Automation, better for large teams, supports DevOps best practices.
- **Cons:** Steeper learning curve.

c. ANT Migration Tool (Legacy):

- **Use Case:** For automated deployments where SFDX is not available.
- **Steps:** Use build.xml scripts to move metadata between orgs.
- **Pros:** Scripted and repeatable.
- **Cons:** Older method, limited metadata types.

8.2 System Maintenance & Monitoring

To ensure continued functionality, performance, and reliability:

a. Maintenance Activities:

- Regular updates for Apex classes, triggers, and validation rules.
- Monitor Salesforce governor limits and optimize SOQL/SOSL usage.
- Review and clean up unused fields, flows, and reports periodically.

b. Monitoring Tools:

- **Salesforce Health Check:** Identify org security risks.
- **Debug Logs:** Monitor errors and system events.
- **Scheduled Reports:** Track KPIs and system usage.
- **Third-party Tools:** New Relic, Splunk, or Salesforce Shield for advanced logging and event monitoring.

9.CONCLUSION

The Garage Management System has been successfully deployed with a structured and reliable strategy, ensuring a smooth transition from development to production. Comprehensive documentation and a clear maintenance plan have been established to support ongoing system stability, user efficiency, and scalability. With proactive monitoring, regular updates, and well-defined troubleshooting procedures, the system is well-prepared to meet current operational needs and adapt to future enhancements, providing long-term value to garage operations.