

Project

WhatNext Vision Motors: **Shaping the Future of Mobility with Innovation and Excellence**

ABSTRACT:

The automotive sector is rapidly evolving, requiring advanced digital solutions to enhance customer experience and improve sales processes. "WhatNext Vision Motors" is a **Salesforce-based CRM application** designed to modernize vehicle dealership operations by automating test drive bookings, streamlining lead management and enabling real time communication between customers and sales agents.

This project leverages **Lightning Web Components, Apex programming and Salesforce automation tools** to create a centralized platform that connects customers, agents and administrators seamlessly. Customers can book test drives online, receive instant notifications and track their booking status, while sales teams benefit from a structured workflow that ensures timely follow ups and better lead conversions.

By integrating automation, data-driven insights and a user-friendly interface, "WhatNext Vision Motors" delivers an innovative solution that transforms traditional dealership operations into a **future-ready, customer centric mobility platform**, driving efficiency, engagement and overall business growth.

OBJECTIVE:

The primary objective of this project is to **design and implement a Salesforce - based CRM solution** that transforms the traditional vehicle test drive process into a **seamless, automated and customer - focused digital experience**. This project aims to:

- **Streamline Test Drive Bookings:** Provide customers with a simple online platform to schedule and track test drives without manual intervention.
- **Enhance Customer Engagement:** Enable real-time notifications and updates to improve communication between customers and sales agents.
- **Optimize Sales Workflows:** Automate lead assignment, follow-ups and data management to increase efficiency and reduce human errors.
- **Provide Actionable Insights:** Deliver dashboards and reports to help dealership administrators make informed, data-driven decisions.
- **Create a Scalable Solution:** Build a future-ready CRM platform adaptable to evolving automotive industry needs and digital mobility innovations.

Technical Overview:

Salesforce Platform: -

Salesforce is a leading **cloud-based Customer Relationship Management (CRM)** platform that enables organizations to manage customer data, sales, and services efficiently. This project uses Salesforce's declarative and programmatic capabilities to design a centralized solution for vehicle test drive scheduling and lead management.

Custom objects: -

- **Vehicle__c** -- Stores vehicle details
- **Vehicle_Dealer__c** -- Stores authorized dealer info
- **Vehicle_Customer__c** -- Stores customer details
- **Vehicle_Order__c** -- Tracks test drive bookings
- **Vehicle_Test_Drive__c** -- Tracks test drive bookings
- **Vehicle_Service_Request__c** -- Tracks vehicle servicing requests

Tabs: -

Tabs provide a way to present object data within the Salesforce user interface. For example, creating a tab for **Vehicle__c** enables users to quickly access, view, and manage product records with ease.

Email Alert: -

These is actions in flows sends reminder emails to customers a day before their scheduled test drive.

For example, "This is a friendly reminder that your test drive (**a04gK00000hNm9QAE**) is scheduled for tomorrow."

Flows: -

Flows automate the quantity of the vehicles after conforming that order is conformed to sold and at other end stock quantity of a vehicles will decrease.

Apex: -

Apex is a salesforce's object-oriented programming language. It used to write the custom logic code.

For example, Updating the stock of the vehicles at every mid night and email alert for test drive before scheduled day.

Detail Execution of Project Phases:

1. Developer Org Setup

- A salesforce developer org was created using <https://developer.salesforce.com/signup>.
- Account was verified, password set and access was granted to the salesforce setup page.

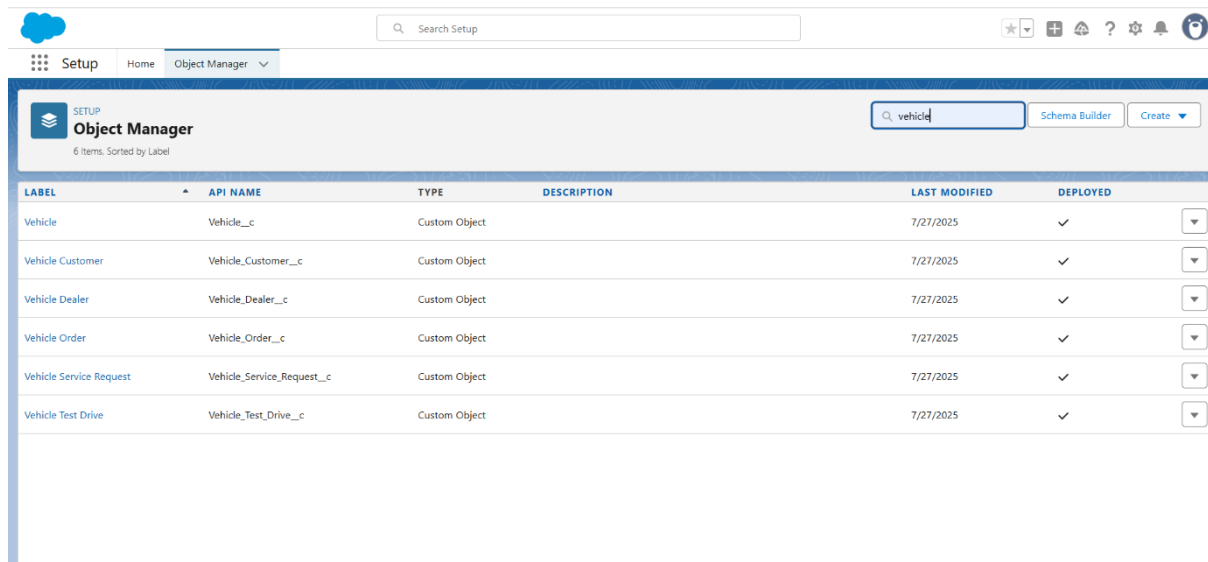
2. Custom Object Creation

Six custom objects were created to store data

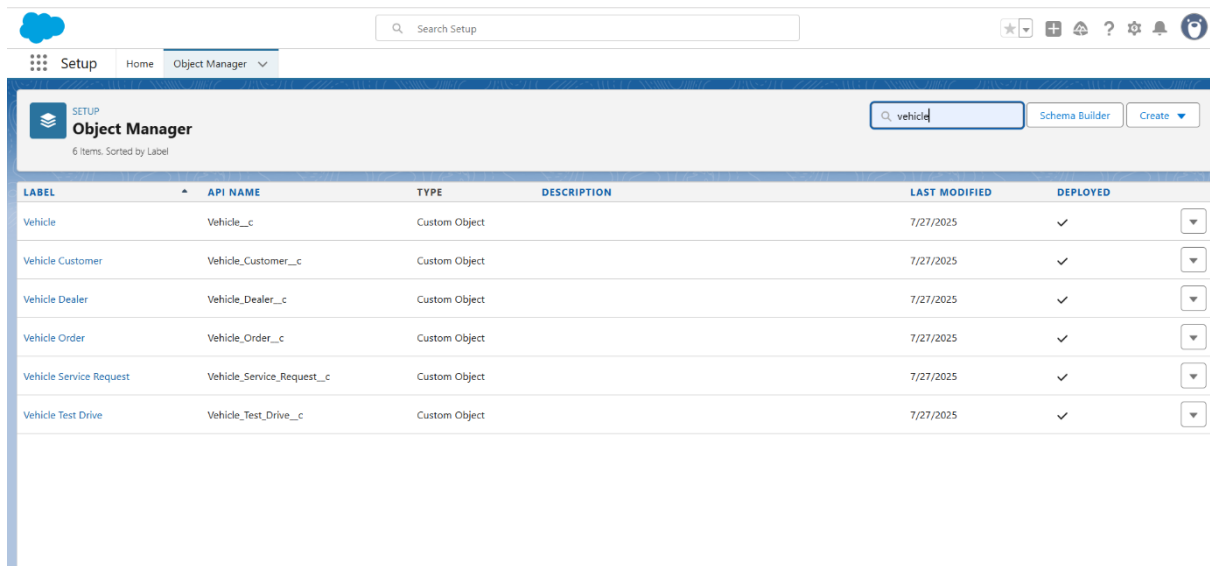
- **Vehicle__c** -- Stores vehicle details
- **Vehicle_Dealer__c** -- Stores authorized dealer info
- **Vehicle_Customer__c** -- Stores customer details
- **Vehicle_Order__c** -- Tracks test drive bookings
- **Vehicle_Test_Drive__c** -- Tracks test drive bookings
- **Vehicle_Service_Request__c** -- Tracks vehicle servicing requests
-

Step followed:

- Navigated to setup → Object Manager → Create → Custom Object
- Provided label, name and enabled search.
- Saved and created Tabs for each other.



Object Manager					
6 Items. Sorted by Label					
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Vehicle	Vehicle__c	Custom Object		7/27/2025	✓
Vehicle Customer	Vehicle_Customer__c	Custom Object		7/27/2025	✓
Vehicle Dealer	Vehicle_Dealer__c	Custom Object		7/27/2025	✓
Vehicle Order	Vehicle_Order__c	Custom Object		7/27/2025	✓
Vehicle Service Request	Vehicle_Service_Request__c	Custom Object		7/27/2025	✓
Vehicle Test Drive	Vehicle_Test_Drive__c	Custom Object		7/27/2025	✓

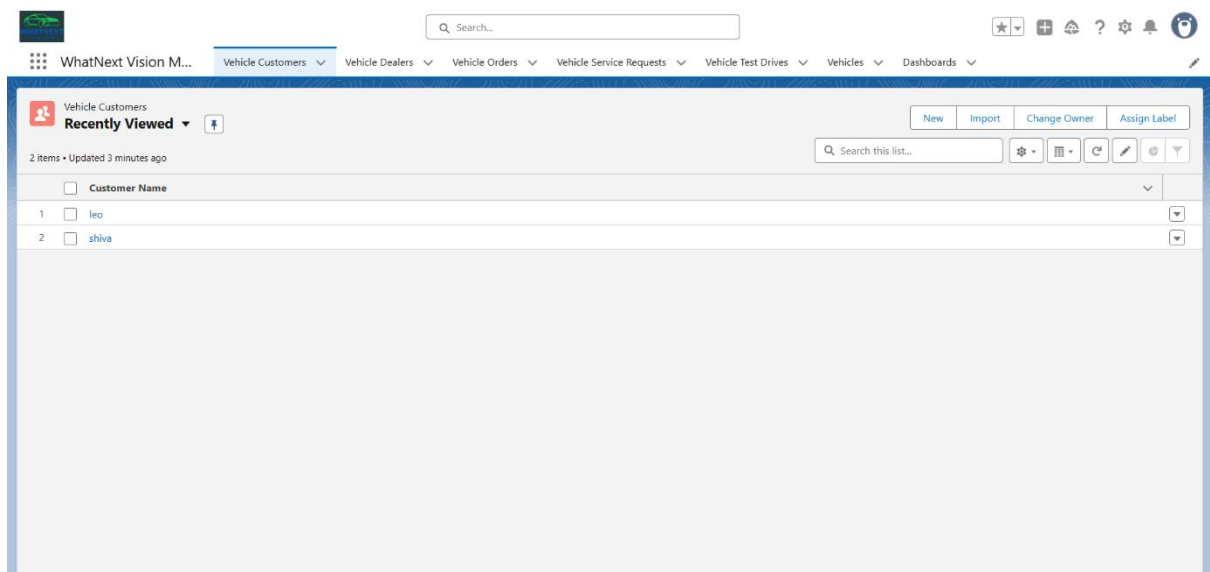


The screenshot shows the Salesforce Setup interface, specifically the Object Manager section. The top navigation bar includes the Setup icon, Home, and Object Manager. A search bar at the top right contains the text "Search Setup". Below the navigation bar, the Object Manager header shows "6 Items. Sorted by Label". A search bar on the right contains the text "vehicle". The main table lists six custom objects, all of which are "Custom Object" type and were last modified on 7/27/2025. Each object has a "DEPLOYED" status of "✓" and a dropdown arrow in the final column.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED	
Vehicle	Vehicle__c	Custom Object		7/27/2025	✓	▼
Vehicle Customer	Vehicle_Customer__c	Custom Object		7/27/2025	✓	▼
Vehicle Dealer	Vehicle_Dealer__c	Custom Object		7/27/2025	✓	▼
Vehicle Order	Vehicle_Order__c	Custom Object		7/27/2025	✓	▼
Vehicle Service Request	Vehicle_Service_Request__c	Custom Object		7/27/2025	✓	▼
Vehicle Test Drive	Vehicle_Test_Drive__c	Custom Object		7/27/2025	✓	▼

3. Creating the Lightning App

- A custom Lightning App names WhatNext Vision Motors was created.
- Included tabs: Vehicle's Customers, Dealers, Orders, Service request, Test drives and Vehicle and also additionally Dashboard.
- Assign to the System Administrator profile.

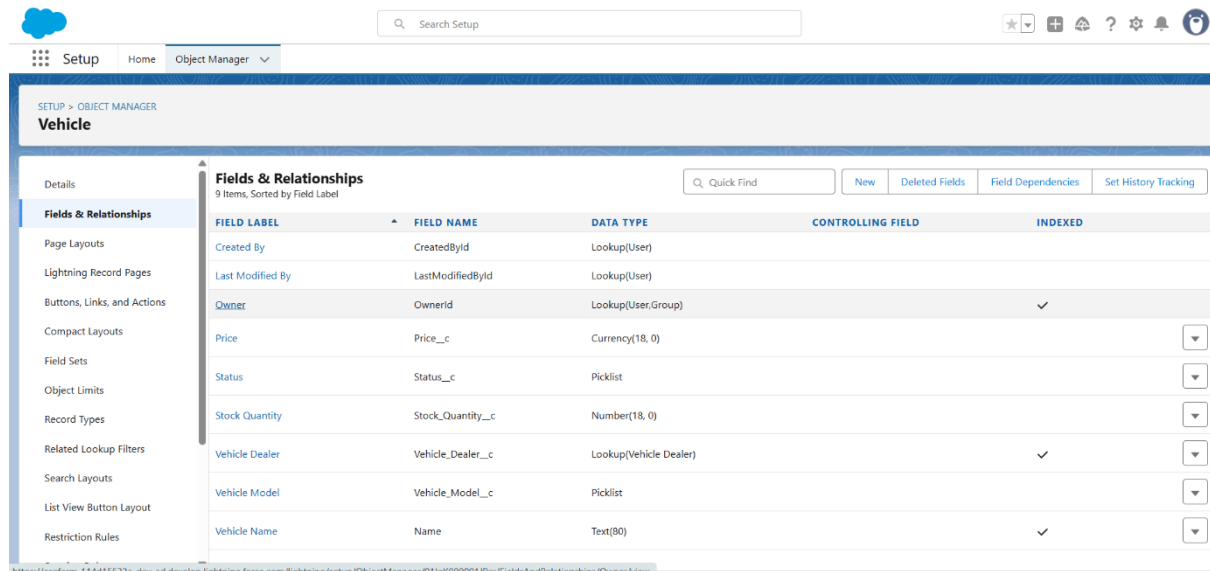


The screenshot shows the Salesforce Lightning App interface for "Vehicle Customers". The top navigation bar includes the app icon, the name "WhatNext Vision M...", and a search bar. Below the navigation bar, the app header shows "Vehicle Customers" and "Recently Viewed". A search bar on the right contains the text "Search this list...". The main table lists two items, both of which are "Customer Name" type and were last updated 3 minutes ago. Each item has a "DEPLOYED" status of "✓" and a dropdown arrow in the final column.

Customer Name	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED	
1	leo	Customer Name		Updated 3 minutes ago	✓	▼
2	shiva	Customer Name		Updated 3 minutes ago	✓	▼

4. Fields & Relationships

- For every custom object Field creation has been created.
- And also created a relationship by the lookup relation datatype for each object according to the given description.



Setup > OBJECT MANAGER
Vehicle

Fields & Relationships
9 Items, Sorted by Field Label

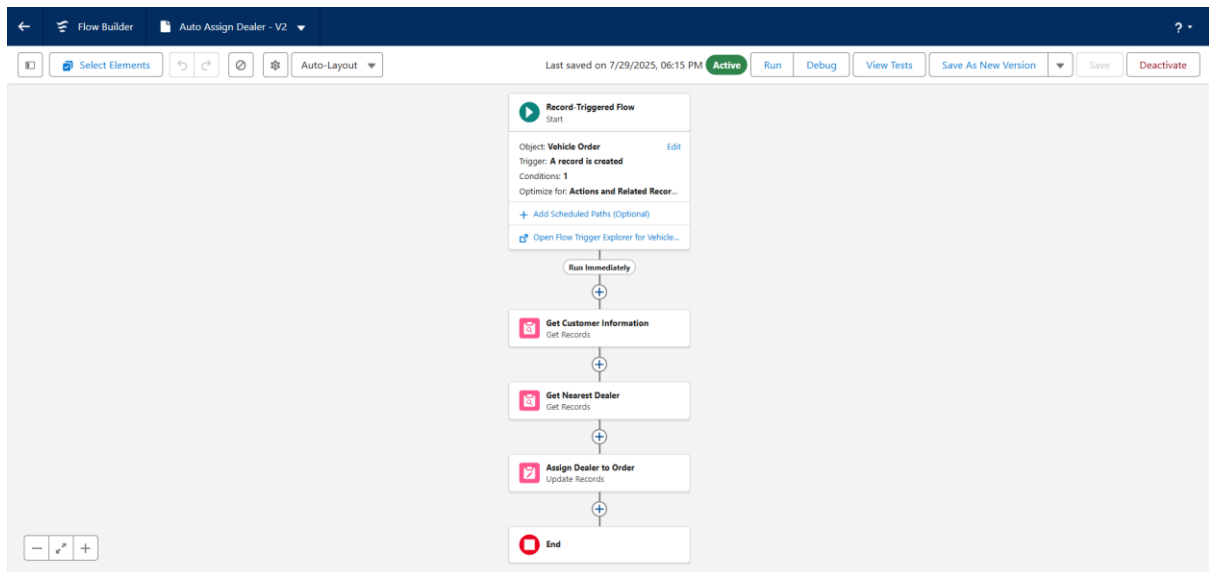
FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Price	Price__c	Currency(18, 0)		
Status	Status__c	Picklist		
Stock Quantity	Stock_Quantity__c	Number(18, 0)		
Vehicle Dealer	Vehicle_Dealer__c	Lookup(Vehicle Dealer)		✓
Vehicle Model	Vehicle_Model__c	Picklist		
Vehicle Name	Name	Text(80)		✓

FIG: Like this other fields & relationships were created for every custom objects.

5. Flows Implementation

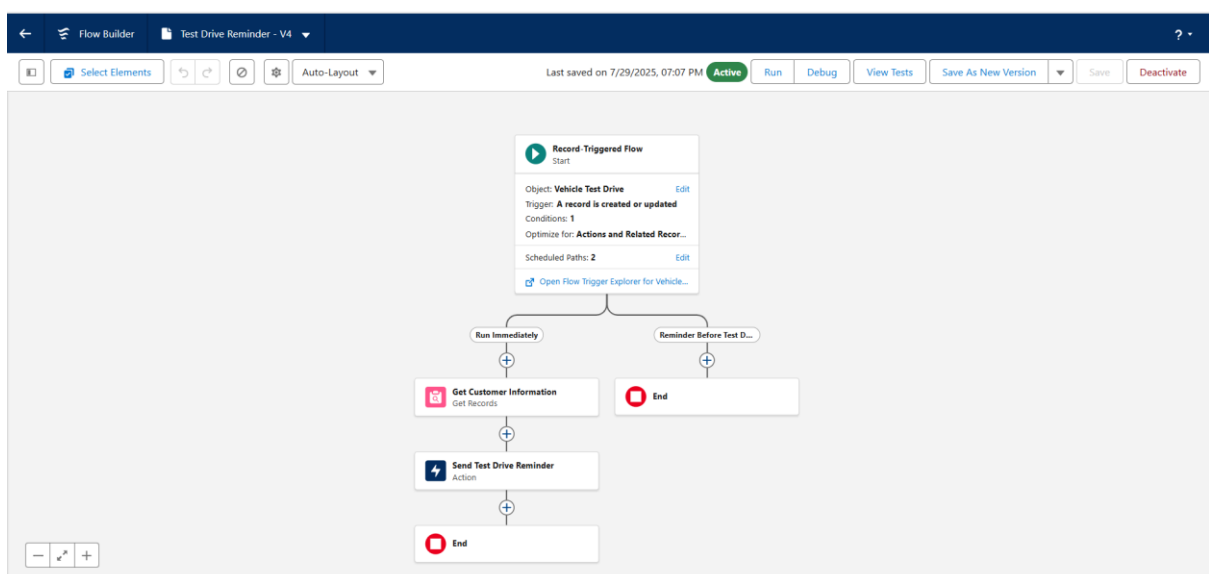
A) Auto Assign Dealer:

- It is Record-Triggered Flow to assign nearest dealer to the customer's location.
- Here I created a two Get Records one for customer information and another one for nearest dealer.
- And finally update records were created to assign dealer to order



B) Test Drive Reminder

- It is Record-Triggered Flow to send an email to the customer reminding about the test drive.
- In these Scheduled Path is created that to work the flow before one day.
- And also created one Get Record element for the customer information about test drive conformation.
- And another element Action is added to it to send a Email Alert for conformation of the Test Drive before one day of conforming.



6. Apex and Trigger Batch Jobs

A) VehicleOrderTriggerHandler (Apex Class)

- **Business Logic Handling:** The VehicleOrderTriggerHandler Apex class is used to manage and execute business rules for vehicle orders, such as preventing orders when vehicles are out of stock and updating stock quantities after an order is confirmed.
- **Clean Code Structure:** It helps separate trigger logic from the trigger itself, making the code reusable, organized, and easier to maintain.

B) VehicleOrderBatch (Apex Class)

- It automates the processing of pending vehicle orders in bulk using Salesforce Batch Apex.
- The class checks vehicle stock availability and updates order statuses to '**Confirmed**' while decrementing the stock quantity accordingly.
- This ensures efficient handling of large volumes of vehicle orders without hitting governor limits.

C) VehicleOrderBatchScheduler

- **Automated Scheduling:** The VehicleOrderBatchScheduler Apex class is used to schedule the execution of the VehicleOrderBatch class at specific intervals without manual intervention.
- **Efficient Processing:** It ensures large volumes of vehicle order records are processed in manageable chunks (batch size 50) for better performance and scalability.

D) VehicleOrderTrigger

- The **VehicleOrderTrigger** Apex class is used to automatically execute predefined logic whenever a new Vehicle Order record is inserted or updated.
- It helps maintain **data integrity and business automation** by invoking the VehicleOrderTriggerHandler to handle specific operations without manual intervention.

E) Schedule the Batch Job To run the batch job every night at midnight

- The batch job is scheduled to run every night at midnight to ensure all vehicle orders are processed on time without manual intervention.
- Running it during off-peak hours reduces system load and ensures data is updated before the next business day.

Step followed:

- Navigate to Developer console → Debug → Open Execute Anonymous Window.
- Copy and pasted the given code and execute.

Project Explanation with Example:

1. Vehicle Tab:

A dealership adds new vehicles to the system with details like model, price, and availability.

Example: A dealer adds 10 units of "Honda SUV 2025" to the inventory.

2. Vehicle Dealer Tab:

Dealers register with their dealership details, enabling them to sell vehicles online.

Example: "Auto World Motors" registers and lists multiple vehicles for customers to view.

3. Vehicle Customer Tab:

Customers create profiles, browse available vehicles, and interact with dealers.

Example: A customer named Alex signs up to explore available SUVs for his family.

4. Vehicle Order Tab:

Customers can place orders directly from the portal once they decide to purchase.

Example: Alex orders a "Honda SUV 2025" after finalizing the price with the dealer.

5. Vehicle Test Drive Tab:

Customers can schedule a test drive before making a final decision. Automated emails remind them of the test drive date.

Example: Alex books a test drive for 5 PM the next day and receives a confirmation email automatically.

6. Vehicle Service Request Tab:

After purchase, customers can log service requests for regular maintenance or repairs.

Example: Alex submits a service request for his SUV's first free maintenance, and the dealer schedules it promptly.

Conclusion:

The Test Drive Automation Email Alert System plays a vital role in simplifying and streamlining dealership operations. By automating reminder emails, it ensures customers are promptly informed about their scheduled test drives, reducing the chances of missed appointments. The use of scheduled batch jobs allows the system to run efficiently without requiring manual intervention, saving both time and resources. Personalized messages improve customer engagement and enhance their overall experience with the dealership. This solution not only optimizes the workflow for the sales team but also fosters better communication and reliability. Implementing such an automated approach strengthens customer relationships and builds trust in the brand. Ultimately, this project demonstrates how automation can significantly enhance service quality in the automotive sector. It lays a strong foundation for integrating more advanced features in the future, such as SMS notifications and AI-Powered scheduling.

Future Scope

1. **AI-Powered Recommendations:** Introduce AI-based suggestions for customers based on preferences and browsing history.
2. **Mobile Application Support:** Develop a mobile app version for quick access and real-time notifications.
3. **Payment Gateway Integration:** Allow secure online payments for vehicle bookings, orders and service requests.
4. **Chatbot Assistance:** Implement AI chatbots to handle customer queries instantly.
5. **Advanced Analytics:** Provide dealers with real-time sales and service insights to improve decision-making.
6. **Integration with IoT:** Connect vehicles for remote diagnostics and predictive maintenance alerts.

