

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

### **ABSTRACT**

WhatNext Vision Motors is enhancing both its customer service capabilities and internal operational efficiency through the implementation of a robust Salesforce CRM ecosystem. The system optimizes the vehicle purchasing workflow by automatically assigning customer orders to the nearest dealership based on geographic proximity and preventing transactions for units that are no longer in stock. Automated processes maintain accurate, real-time order status updates and generate reminder emails prior to scheduled test-drive appointments. The platform's functionality is supported by Apex triggers to enforce inventory rules, Batch Apex for routine stock validation, and Scheduled Apex for autonomous processing of pending orders. Collectively, these advancements improve customer satisfaction, minimize ordering inaccuracies, and streamline day-to-day dealership operations.

### **OBJECTIVES**

This capstone project aims to fully digitize the entire vehicle order and service management workflow by integrating automation across all operational stages. It seeks to assign customers to the nearest eligible dealer based on address using Salesforce Flows, while preventing orders for vehicles that are no longer available through Apex triggers. To ensure real-time accuracy in inventory records, the system continuously monitors and updates stock availability using batch and scheduled Apex jobs. The platform also streamlines service scheduling and sends automated email reminders for test drives and maintenance appointments. Additionally, the use of validation rules reduces manual data entry and promotes data accuracy, while automated status notifications enhance customer engagement throughout the ordering and servicing lifecycle. Overall, the system is designed with scalability in mind to support future business expansion and potential third-party integrations.

## TECHNOLOGY STACK

- **Lightning App Builder:** A drag-and-drop tool for creating custom apps and user interfaces. It allows the addition of pages, components, and tabs to deliver a tailored experience for different users.
- **Salesforce CRM Platform:** The cloud-based foundation for the application, supporting object creation, automation, and interface configuration.
- **Flow Builder:** Enables no-code automation for various business processes, including:
  - **Scheduled Flows:** Run at specific times to automate tasks like stock monitoring and status updates.
  - **Record-Triggered Flows:** Automatically triggered when a record is created or updated, used for dealer assignment, status updates, and sending emails.
- **Apex Classes and Triggers:** Implements advanced logic and enforces business rules during record operations, such as vehicle stock validation.
- **Scheduled Apex:** Executes automated jobs at defined intervals, for example, sending order updates or summaries daily.
- **Batch Apex:** Handles large data volumes in batches to maintain accurate stock levels and vehicle availability.
- **Email Alerts and Notifications:** Configured through Flow Builder to automatically send reminders, confirmations, and updates to customers and dealers, ensuring consistent communication.

## SYSTEM REQUIREMENTS & FUNCTIONAL HIGHLIGHTS

### Salesforce CRM Core Setup

- Maintain comprehensive and up-to-date records of all vehicles, inventory levels, and dealership branch information.
- Efficiently document and monitor customer transactions, scheduled test drives, and service-related inquiries.
- Automatically assign customer orders to the dealership geographically closest to the customer's location.

## **Business Process Automation**

- Prevent order submissions when the selected vehicle is out of stock.
- Initiate automated dealership assignment based on the customer's address or location data.
- Dispatch scheduled email notifications to provide timely reminders for upcoming test-drive appointments.

## **Apex Development & Trigger Logic**

- Utilize Apex triggers to enforce critical business rules, including inventory validation and automated order routing.
- Adopt a structured trigger handler architecture to promote clean, scalable, and easily maintainable code.

## **Batch & Scheduled Processes**

- Employ Batch Apex to regularly monitor inventory levels and update vehicle availability records.
- Utilize Scheduled Apex to automate routine stock updates and facilitate order-related notifications.

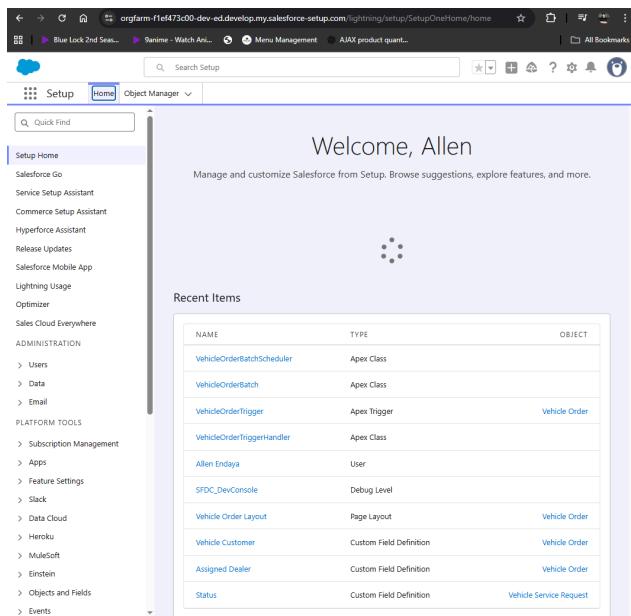
### **What you'll learn:**

1. Data Modelling
2. Fields and Relationships
3. Lightning App Builder
4. Record Triggered Flows
5. Apex and Apex Triggers
6. Batch Apex
7. Scheduled Apex

## DETAILED EXECUTION OF PROJECT PHASES

### Phase 1: Developer Org Setup

- Created a free Salesforce Developer Org from <https://developer.salesforce.com/signup>
- Verified the account via email and logged in.
- Enabled Developer Console and activated Lightning Experience.
- Personalized the org by enabling necessary features for custom object and app creation.



## Phase 2: Requirement Gathering & Planning

- Defined project scope based on real-world vehicle service processes.
- Identified major entities: Customer, Vehicle, Dealer, Order, and Test Drive.
- Outlined automation goals: automatic dealer assignment, stock validation, and email reminders.

## Phase 3: Custom Object Creation

- Created six main custom objects:
  - Vehicle\_\_c – Name, Type, Price, Stock Quantity
  - Vehicle\_Customer\_\_c – Name, Email, Phone, Address
  - Vehicle\_Dealer\_\_c – Name, Location, Assigned Vehicles
  - Vehicle\_Order\_\_c – Vehicle (lookup), Customer (lookup), Dealer (lookup), Status, Date
  - Vehicle\_Service\_Request\_\_c – Customer (lookup), Vehicle (lookup), Request Date, Description, Status
  - Vehicle\_Test\_Drive\_\_c – Customer (lookup), Vehicle (lookup), Preferred Date
- Configured relationships using Lookup fields.

The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with links like 'Blue Lock 2nd Seas...', '9anime - Watch Ani...', 'Menu Management', and 'AJAX product quant...'. Below the bar, the setup menu is open, and the 'Object Manager' tab is selected. The main area is titled 'Object Manager' and displays a table of six custom objects. The table has columns for Label, API Name, Type, Description, Last Modified, and Deployed. Each row contains a checkmark in the Deployed column and a dropdown arrow icon in the last column.

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Vehicle	Vehicle_c	Custom Object		11/20/2025	✓
Vehicle Customer	Vehicle_Customer_c	Custom Object		11/20/2025	✓
Vehicle Dealer	Vehicle_Dealer_c	Custom Object		11/20/2025	✓
Vehicle Order	Vehicle_Order_c	Custom Object		11/20/2025	✓
Vehicle Service Request	Vehicle_Service_Request_c	Custom Object		11/20/2025	✓
Vehicle Test Drive	Vehicle_Test_Drive_c	Custom Object		11/20/2025	✓

## Phase 4: Data Management-Table

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-f1ef473c00-dev-ed.develop.my.salesforce-setup.com/lightning/setup/CustomTabs/page?area=Custom%20Tabs>. The page title is "Custom Tabs".

The left sidebar shows a search bar and navigation links for "Setup", "Home", and "Object Manager". The "User Interface" section is expanded, showing "Rename Tabs and Labels" and "Tabs". A message says "Didn't find what you're looking for? Try using Global Search."

The main content area displays the "Custom Tabs" section with a heading "Custom Tabs" and a "Help for this Page" link. It includes a table for "Custom Object Tabs" and sections for "Web Tabs" and "Visualforce Tabs". Below these is a table for "Lightning Component Tabs".

**Custom Object Tabs**

Action	Label	Tab Style	Description
Edit   Del	<a href="#">Vehicle Customers</a>	People	
Edit   Del	<a href="#">Vehicle Dealers</a>	Building	
Edit   Del	<a href="#">Vehicle Orders</a>	Box	
Edit   Del	<a href="#">Vehicles</a>	Car	
Edit   Del	<a href="#">Vehicle Service Requests</a>	Form	
Edit   Del	<a href="#">Vehicle Test Drives</a>	Gears	

**Web Tabs**

No Web Tabs have been defined.

**Visualforce Tabs**

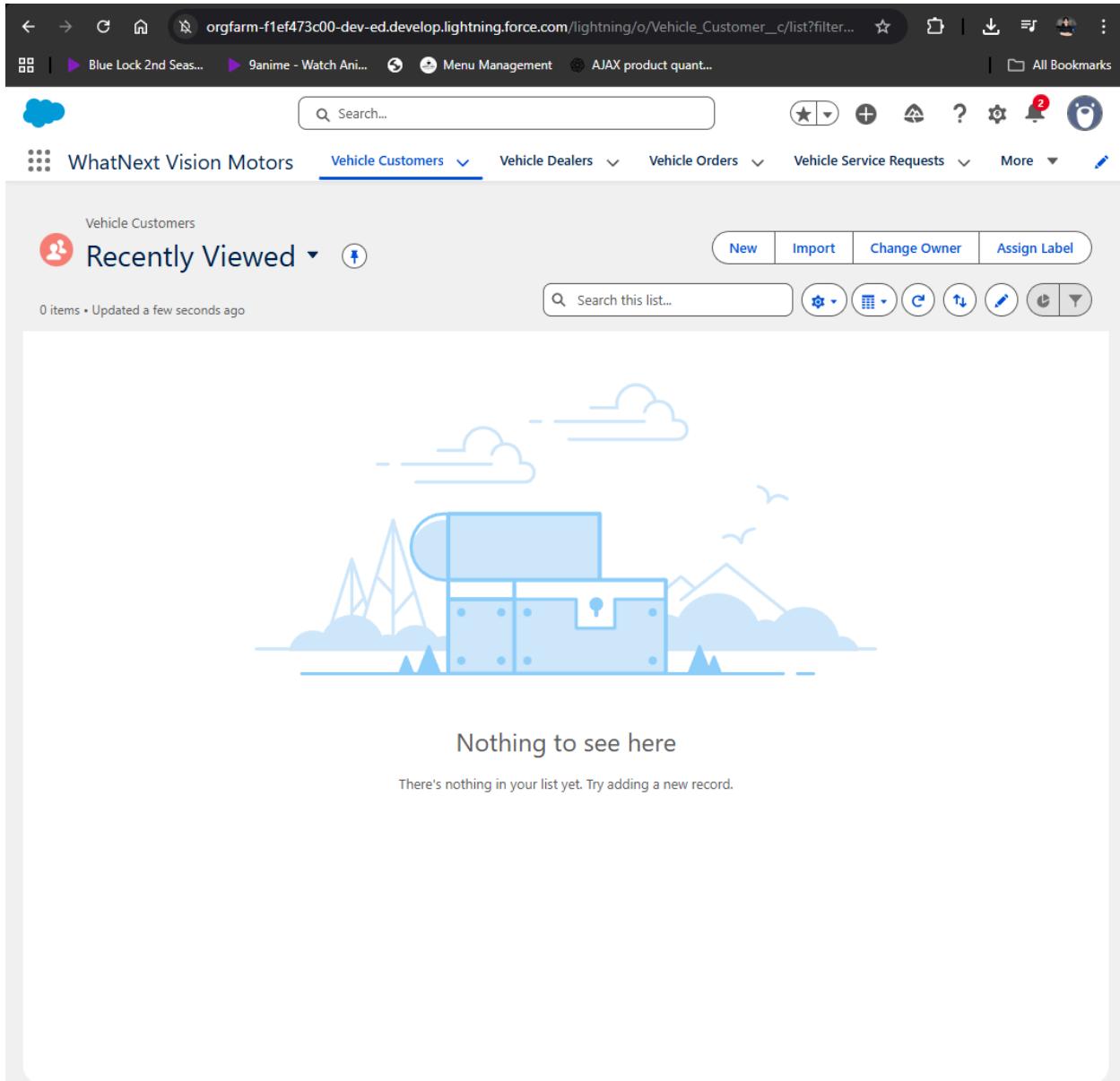
No Visualforce Tabs have been defined.

**Lightning Component Tabs**

Action	Label	Tab Style	Description
Edit	Get Started with Agentforce	Heart	
Edit	Get Started with Data Cloud	Map	
Edit	Get Started with MuleSoft	Heart	
Edit	Get Started with Salesforce DX	Building Block	

## Phase 5: Lightning App Development

- Used Lightning App Builder to create a custom app named WhatNext Vision Motors
- Added custom object tabs for seamless navigation.
- Customized record pages with components: Related Lists, Charts, and Action Buttons.



## Phase 6: Field Creation and Validation Rules

- Created all required fields with appropriate data types like Email, Phone, Picklist, and Lookup.
- Implemented validation rules to prevent incomplete or incorrect submissions.

### Vehicle Data Management-Fields

The screenshot shows the Salesforce Object Manager interface for the 'Vehicle' object. The left sidebar lists various setup categories. The main area displays the 'Fields & Relationships' section, which includes a table of fields and their properties. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The indexed column contains checkboxes, some of which are checked. The table rows are as follows:

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)		✓

## Vehicle Dealer Data Management-Fields

The screenshot shows the Salesforce Object Manager interface for the 'Vehicle Dealer' object. The top navigation bar includes links for 'Blue Lock 2nd Seas...', '9anime - Watch Ani...', 'Menu Management', and 'AJAX product quant...'. The main header features a cloud icon, a search bar with placeholder 'Search Setup', and various global navigation icons.

The left sidebar contains a tree view of setup categories: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, Object Access, Triggers, Flow Triggers, Validation Rules, and Conditional Field Formatting.

The right panel displays the 'Fields & Relationships' section for the 'Vehicle Dealer' object. It shows 8 items, sorted by Field Label. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Dealer Code	Dealer_Code__c	Auto Number		
Dealer Location	Dealer_Location__c	Text(60)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)	✓	
Phone	Phone__c	Phone		
Vehicle Dealer Name	Name	Text(80)	✓	

## Vehicle Order Data Management-Fields

The screenshot shows the Salesforce Object Manager interface for the 'Vehicle Order' object. The top navigation bar includes links for 'Blue Lock 2nd Seas...', '9anime - Watch Ani...', 'Menu Management', and 'AJAX product quant...'. The search bar contains 'Search Setup'. The main menu tabs are 'Setup', 'Home', and 'Object Manager' (selected).

The page title is 'SETUP > OBJECT MANAGER' followed by 'Vehicle Order'. On the left, a sidebar lists various setup categories: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules, Object Access, Triggers, Flow Triggers, Validation Rules, and Conditional Field Formatting.

The main content area displays the 'Fields & Relationships' section. It shows 8 items, sorted by Field Label. The table has columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Order Date	Order_Date__c	Date		
Owner	OwnerId	Lookup(User,Group)	✓	
Status	Status__c	Picklist		
Vehicle	Vehicle__c	Lookup(Vehicle)	✓	
Vehicle Customer	Vehicle_Customer__c	Lookup(Vehicle Customer)	✓	
Vehicle Order Number	Name	Auto Number	✓	

## Vehicle Customer Data Management-Fields

The screenshot shows the Salesforce Object Manager interface for the 'Vehicle Customer' object. The left sidebar lists various setup categories. The main area displays the 'Fields & Relationships' section, which includes a table listing eight fields. The table columns are: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address_c	Text(60)		
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)	✓	
Phone	Phone_c	Phone		
Preferred Vehicle Type	Preferred_Vehicle_Type_c	Picklist		
Vehicle Customer Name	Name	Text(80)	✓	

## Vehicle Test Drive Data Management-Fields

The screenshot shows the Salesforce Object Manager interface for the 'Vehicle Test Drive' object. The left sidebar lists various setup categories, and the main area displays the 'Fields & Relationships' section. The table lists the following fields:

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)		✓
Status	Status_c	Picklist		▼
Test Drive Date	Test_Drive_Date_c	Date		▼
Vehicle	Vehicle_c	Lookup(Vehicle)	✓	▼
Vehicle Customer	Vehicle_Customer_c	Lookup(Vehicle Customer)	✓	▼
Vehicle Test Drive Name	Name	Text(80)	✓	▼

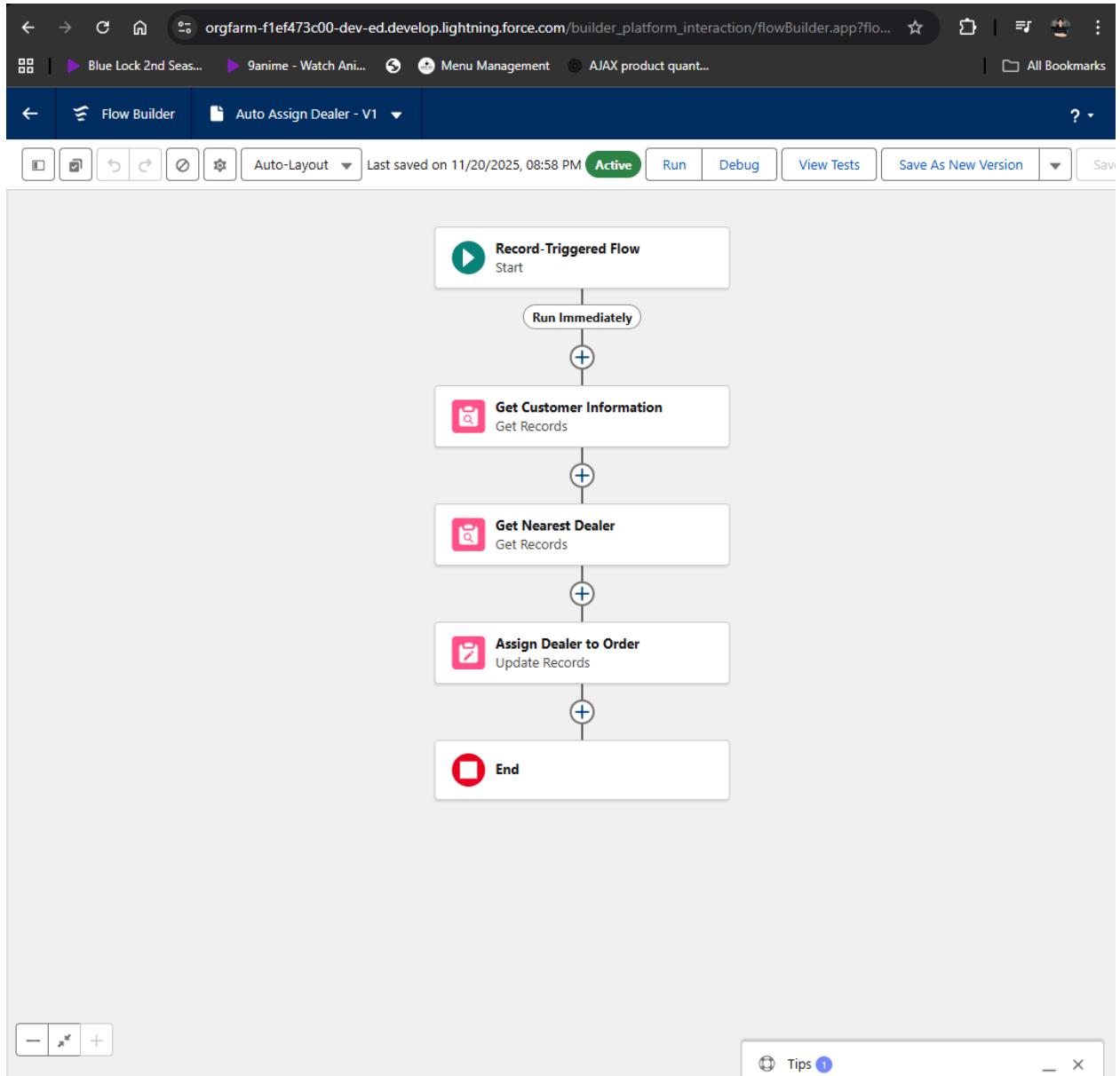
## Vehicle Service Request Data Management-Fields

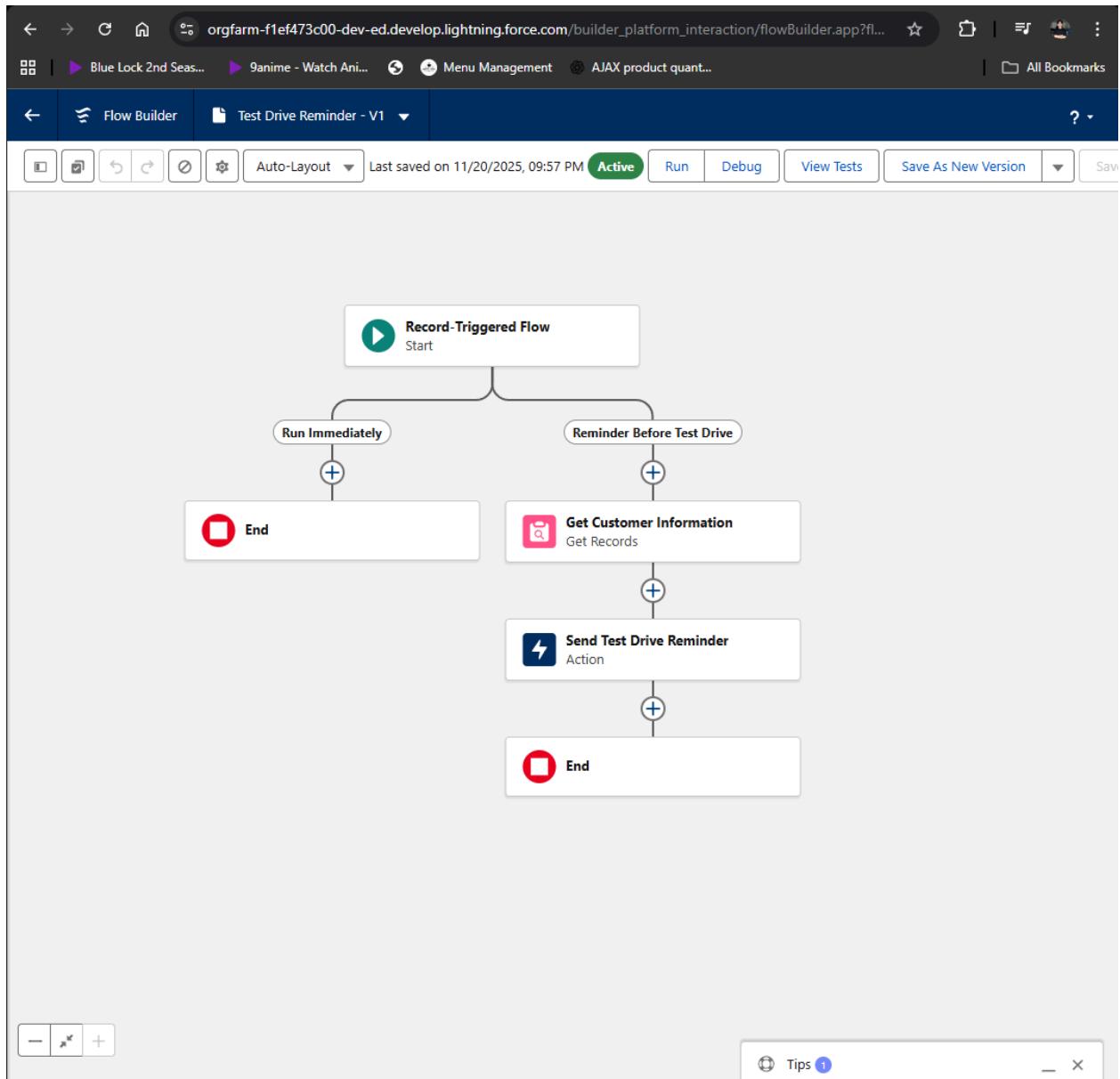
The screenshot shows the Salesforce Object Manager interface for the 'Vehicle Service Request' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, etc. The main area displays the 'Fields & Relationships' section with a table of fields.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Issue Description	Issue_Description__c	Text(60)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)	✓	
Service Date	Service_Date__c	Date		
Status	Status__c	Picklist		
Vehicle	Vehicle__c	Lookup(Vehicle)	✓	
Vehicle Customer	Vehicle_Customer__c	Lookup(Vehicle Customer)	✓	
Vehicle Service Name	Name	Text(80)	✓	

## Phase 7: Flow Implementation

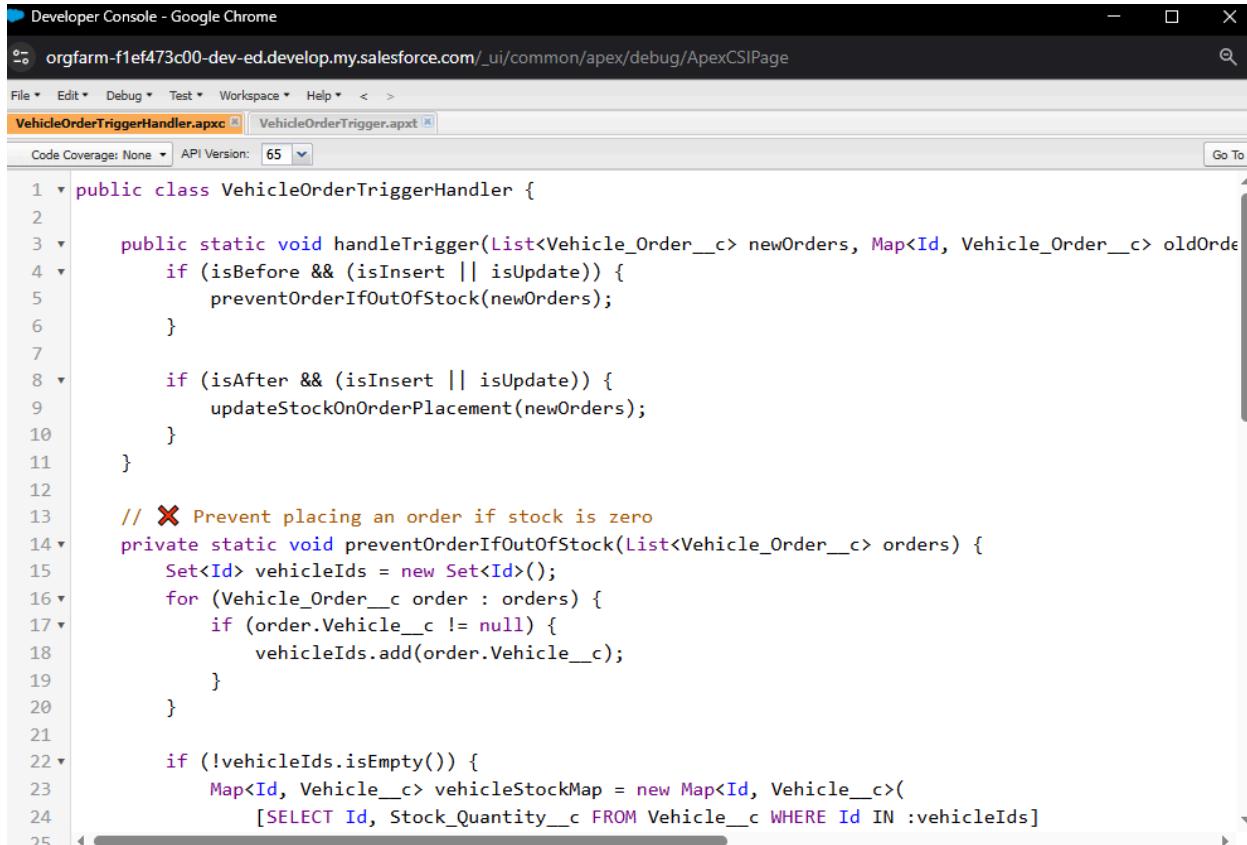
- Auto-Assignment Flow: Assigns the nearest dealer based on customer address.
- Status Update Flow: Automatically updates order status based on stock.
- Test Drive Reminder Flow: Sends email reminders to customers before their scheduled test drive.
- Configured flow triggers, elements, and conditions using Flow Builder.





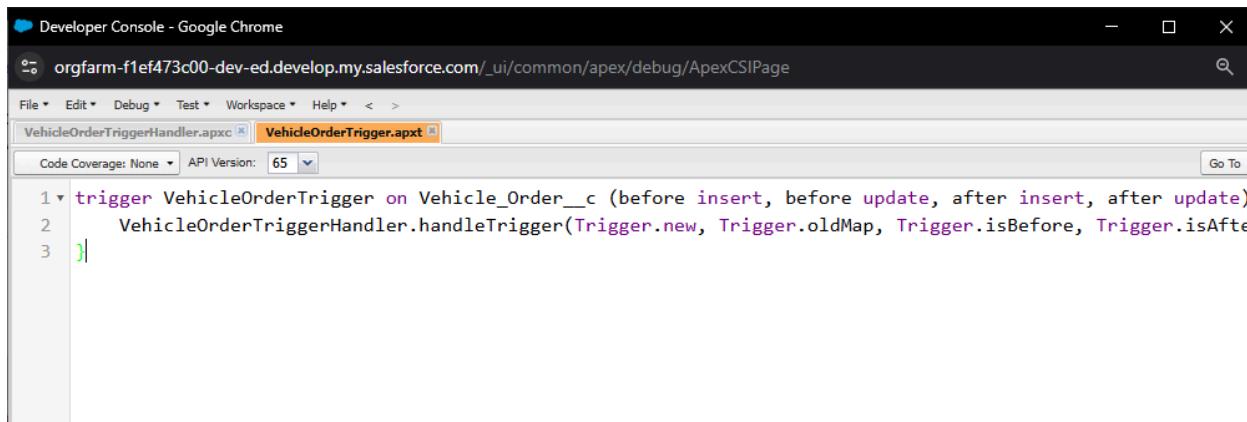
## Phase 8: Apex Trigger and Classes

- Developed an Apex Trigger to validate vehicle stock before allowing order creation.
- Created Apex Classes to modularize logic for dealer selection and stock handling.
- Applied Trigger Handler Framework for best practices.



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `orgfarm-f1ef473c00-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar has two tabs: `VehicleOrderTriggerHandler.apxc` (selected) and `VehicleOrderTrigger.apxt`. The code editor displays the `VehicleOrderTriggerHandler` class:

```
1 public class VehicleOrderTriggerHandler {  
2  
3     public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders,  
4         boolean isBefore, boolean isInsert, boolean isUpdate) {  
5             if (isBefore && (isInsert || isUpdate)) {  
6                 preventOrderIfOutOfStock(newOrders);  
7             }  
8  
9             if (isAfter && (isInsert || isUpdate)) {  
10                 updateStockOnOrderPlacement(newOrders);  
11             }  
12         }  
13  
14     // ✗ Prevent placing an order if stock is zero  
15     private static void preventOrderIfOutOfStock(List<Vehicle_Order__c> orders) {  
16         Set<Id> vehicleIds = new Set<Id>();  
17         for (Vehicle_Order__c order : orders) {  
18             if (order.Vehicle__c != null) {  
19                 vehicleIds.add(order.Vehicle__c);  
20             }  
21         }  
22         if (!vehicleIds.isEmpty()) {  
23             Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(  
24                 [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]  
25             );  
26             for (Vehicle_Order__c order : orders) {  
27                 if (order.Vehicle__c != null) {  
28                     Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);  
29                     if (vehicle != null && vehicle.Stock_Quantity__c < order.Quantity__c) {  
30                         preventOrderIfOutOfStock(order);  
31                     }  
32                 }  
33             }  
34         }  
35     }  
36 }  
37 }
```



The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is `orgfarm-f1ef473c00-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. The tab bar has two tabs: `VehicleOrderTriggerHandler.apxc` and `VehicleOrderTrigger.apxt` (selected). The code editor displays the `VehicleOrderTrigger` trigger:

```
1 trigger VehicleOrderTrigger on Vehicle_Order__c (before insert, before update, after insert, after update)  
2     VehicleOrderTriggerHandler.handleTrigger(Trigger.new, Trigger.oldMap, Trigger.isBefore, Trigger.isAfter)  
3 }
```

Developer Console - Google Chrome

orgfarm-f1ef473c00-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

VehicleOrderTriggerHandler.apxc VehicleOrderTrigger.apxt **VehicleOrderBatch.apxc**

Code Coverage: None API Version: 65 Go To

```
19     [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
20 );
21
22     List<Vehicle_Order__c> ordersToUpdate = new List<Vehicle_Order__c>();
23     List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
24
25     for (Vehicle_Order__c order : orderList) {
26         Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
27         if (vehicle != null && vehicle.Stock_Quantity__c > 0) {
28             order.Status__c = 'Confirmed';
29             vehicle.Stock_Quantity__c -= 1;
30             ordersToUpdate.add(order);
31             vehiclesToUpdate.add(vehicle);
32         }
33     }
34
35     if (!ordersToUpdate.isEmpty()) update ordersToUpdate;
36     if (!vehiclesToUpdate.isEmpty()) update vehiclesToUpdate;
37 }
38
39
40     global void finish(Database.BatchableContext bc) {
41         System.debug('Vehicle order batch job completed.');
42     }
43 }
```

Developer Console - Google Chrome

orgfarm-f1ef473c00-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

VehicleOrderTriggerHandler.apxc VehicleOrderTrigger.apxt VehicleOrderBatch.apxc **VehicleOrderBatchScheduler.apxc**

Code Coverage: None API Version: 65 Go To

```
1 global class VehicleOrderBatchScheduler implements Schedulable {
2     global void execute(SchedulableContext sc) {
3         VehicleOrderBatch batchJob = new VehicleOrderBatch();
4         Database.executeBatch(batchJob, 50); // 50 = batch size
5     }
6 }
```

## Phase 9: Testing and Debugging

- Created test records to simulate real customer orders and vehicle bookings.
- Triggered flows and observed real-time changes.
- Verified email delivery for test drives.

The screenshot shows a Salesforce Lightning page for a 'Vehicle Test Drive' record. The URL in the browser is `orgfarm-f1ef473c00-dev-ed.lightning.force.com/lightning/r/Vehicle_Test_Drive_c/a04gK000001...`. The page title is 'Vehicle Test Drive "test drive" was created.' The header includes navigation links for 'Vehicle Customers', 'Vehicle Dealers', 'Vehicle Orders', and 'Vehicle Test Drives'. The main content area displays the following details:

Field	Value
Vehicle Test Drive Name	test drive
Vehicle Customer	<a href="#">Allen Benedict O. Endaya</a>
Vehicle	<a href="#">Ford</a>
Test Drive Date	11/22/2025
Status	Scheduled
Created By	<a href="#">Allen Endaya</a> , 11/20/2025, 5:59 AM
Owner	<a href="#">Allen Endaya</a>
Last Modified By	<a href="#">Allen Endaya</a> , 11/20/2025, 5:59 AM

Below the details, there are tabs for 'Related' and 'Details', with 'Details' being the active tab. A blue success message box is visible at the top right of the page.

## **REAL WORLD USE CASE**

Picture a customer named Riya browsing the WhatNext Motors website or visiting a dealership to purchase a car. The moment her information is entered into Salesforce, the system begins working in the background:

### **1. Automatic Dealer Matching**

- Her pin code or address is analyzed.
- A Flow locates the closest dealer that has her chosen vehicle available.
- The order gets instantly assigned to that dealer — no manual routing needed.

### **2. Smart Stock Check**

- An Apex trigger verifies the vehicle's availability before finalizing the order.
- If the car isn't in stock, the system stops the order and notifies the user immediately.

### **3. Live Order Progress**

- A scheduled Apex job reviews stock levels every night.
- If supply becomes available, the order is automatically moved from "Pending" to "Confirmed."

### **4. Automated Follow-Ups**

- When Riya schedules a test drive, she receives an automated email reminder one day before the appointment.

### **5. Full Customer Visibility**

- Riya can reach out to the assigned dealer anytime and get accurate updates on her order status.

This mirrors the intelligent CRM workflows used by top automobile companies like Hyundai and Tata Motors to improve customer satisfaction and sharply reduce operational confusion.

## CONCLUSION

This capstone project highlights how Salesforce can effectively address operational challenges in the automotive sector through a scalable, low-code approach. The development of the customized Lightning App, WhatNext Vision Motors, modernizes vehicle ordering and service handling by automating the entire workflow from collecting customer inquiries to assigning nearby dealers and sending important updates.

Through the integration of custom objects, automation flows, Apex triggers, and batch processing, the system ensures consistent data accuracy and smooth communication between customers and dealers. Automated email alerts and real-time inventory validation help minimize human workload while enhancing service reliability and customer experience.

By leveraging Record-Triggered Flows, Scheduled Flows, and Batch Apex Jobs, the platform continuously oversees vehicle availability and responds proactively to status changes. The structured use of Apex classes along with a trigger handler framework ensures flexibility, making future enhancements easier to implement.

In summary, this solution successfully meets its objectives of boosting automation, improving efficiency, and strengthening user engagement. It also lays the groundwork for more advanced capabilities such as mobile app support, AI-driven insights, and enhanced analytics, demonstrating the full potential of Salesforce in transforming dealership operations into a smarter and more customer-focused environment.

## FUTURE SCOPE

1. **IoT Vehicle Tracking:** Integrate IoT devices to provide real-time location and status updates of vehicles during delivery.
2. **Voice Assistant Integration:** Allow customers to place orders or schedule services using voice commands through smart assistants.
3. **Predictive Maintenance Alerts:** Use AI to predict upcoming vehicle maintenance needs based on usage patterns and service history.
4. **Loyalty and Rewards System:** Implement a rewards program for repeat customers, tracking purchases and service activities.
5. **Augmented Reality (AR) Vehicle Preview:** Let customers visualize car models in their environment before purchase using AR technology.\
6. **Multi-Language Support:** Expand the system to support multiple languages for a wider customer base.
7. **Dealer Performance AI Insights:** Use AI to provide predictive insights and recommendations to dealers for improving service efficiency and sales strategies.

