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

Project Overview

WhatNext Vision Motors is a trailblazer in the automotive industry, committed to transforming how customers interact with vehicle ordering and dealership services. Leveraging the power of Salesforce CRM Developer Edition, this project modernizes core business processes by automating dealer assignments, managing stock availability, streamlining order workflows, and enhancing customer engagement.

The initiative centers on creating a seamless journey from order placement to delivery and servicing, improving efficiency and satisfaction. Key among its features is an intelligent system that automatically identifies the nearest dealer based on the customer’s location, facilitating quicker and more convenient order processing. This smart matching reduces customer effort and optimizes dealer workload, ultimately driving higher conversion rates and stronger customer loyalty.

To further increase operational efficiency, the system incorporates automated batch processes that monitor pending orders, converting them to confirmed status once inventory is replenished. Complementing this are automated test drive reminders designed to keep customers informed and reduce appointment no-shows, thereby improving overall service reliability. This proactive communication strategy enhances the customer experience by ensuring timely engagement throughout the sales cycle.

Robust dashboards and insightful reports provide management with real-time visibility into vehicle stock, dealer performance, and order trends, enabling data-driven decision-making and strategic planning. These analytics empower leadership to quickly identify bottlenecks, adjust inventory levels, and optimize dealer allocations.

By combining declarative tools like Flow Builder and Reports with programmatic solutions such as Apex Triggers and Batch Apex, WhatNext Vision Motors has built a scalable and resilient CRM solution that drives innovation without compromising service quality. The project also lays a strong foundation for future enhancements, including AI-driven forecasting, integration with third-party logistics providers, and personalized marketing campaigns, positioning the company as a leader in digital transformation within the automotive sector.

Introduction

In today’s fast-paced digital world, automotive companies must evolve beyond traditional methods to meet rising customer expectations. **WhatsNext Vision Motors**, a forward-thinking player in the automotive sector, recognized the urgent need to modernize its vehicle ordering and dealership operations. With a commitment to innovation and seamless customer service, the company turned to **Salesforce CRM Developer Edition** as the platform to lead this transformation.

This individual capstone project was designed to address common operational bottlenecks such as manual dealer assignment, order errors due to stock mismatches, and inconsistent communication with customers. By leveraging Salesforce’s powerful automation and integration tools, the project aimed to build a smarter, more efficient system that could handle everything from identifying the nearest dealer to updating order statuses based on real-time inventory.

Through a combination of **automated processes**, **batch jobs**, and **customized flows**, the solution ensures that customers experience a smoother, faster, and more accurate vehicle ordering process. Whether it's suggesting the closest dealer, preventing out-of-stock purchases, or sending timely reminders for test drives, the system is engineered to reduce manual tasks and increase service reliability.

Ultimately, this project reflects WhatsNext Vision Motors’ mission to deliver cutting-edge customer experiences while improving internal efficiency—demonstrating how even traditional industries can thrive in a digital-first era.

Objectives

The Salesforce CRM project at WhatsNext Vision Motors aims to revolutionize the way vehicle orders and dealership interactions are handled by introducing a modern, automated, and intelligent system. The key objectives of this initiative are:

1. **Digitize and Simplify the Ordering Process**

   Create a smooth and user-friendly vehicle ordering system where customers can place orders confidently, with minimal friction or delays.

2. **Enable Smart Dealer Recommendations**

   Automatically recommend the nearest dealer to the customer based on their provided address, ensuring faster service and improved customer convenience.

3. **Prevent Out-of-Stock Errors**

   Integrate stock validation at the time of order to ensure customers can only place orders for vehicles that are currently available in inventory.

4. **Automate Order Status Updates**

   Use scheduled automation (Batch Apex) to regularly check inventory and update order records from "Pending" to "Confirmed" based on availability.

5. **Increase Customer Communication and Engagement**

   Send timely alerts and reminders (such as test drive notifications) to customers, helping them stay informed and reducing missed interactions.

6. **Empower Decision-Makers with Analytics**

   Generate real-time dashboards and reports to help managers and stakeholders track key metrics like stock levels, dealer performance, and order flow.

7. **Improve Operational Efficiency**

   Reduce manual work through automation and intelligent workflows, allowing staff to focus on higher-value tasks and improving overall productivity.

8. **Ensure Long-Term Scalability**

   Design the system to be flexible and scalable, capable of supporting future growth, additional functionalities, and new business processes.

Technologies Used

1. **Salesforce Platform:** Core CRM platform for building custom objects, automations, and user interfaces to manage business data efficiently.
2. **Lightning App Builder:** Tool for designing custom apps, tabs, and page layouts tailored to business needs.
3. **Flow Builder:** Automates business processes such as dealer assignments and sending test drive reminders without code.
4. **Apex (Triggers & Classes):** Salesforce’s proprietary programming language used to enforce complex business logic like stock validation and order status updates.
5. **Batch Apex:** Processes large volumes of data asynchronously, such as updating pending orders based on available stock.
6. **Scheduled Apex:** Automates recurring tasks by running batch jobs at predefined intervals (e.g., every midnight).
7. **SOQL (Salesforce Object Query Language):** Used within Apex to query Salesforce records for data retrieval and manipulation.
8. **Email Templates:** Customized automated emails for customer engagement, like test drive reminders.
9. **Reports & Dashboards:** Provide visual insights into inventory, orders, customer activities, and dealer performance for data-driven decisions.

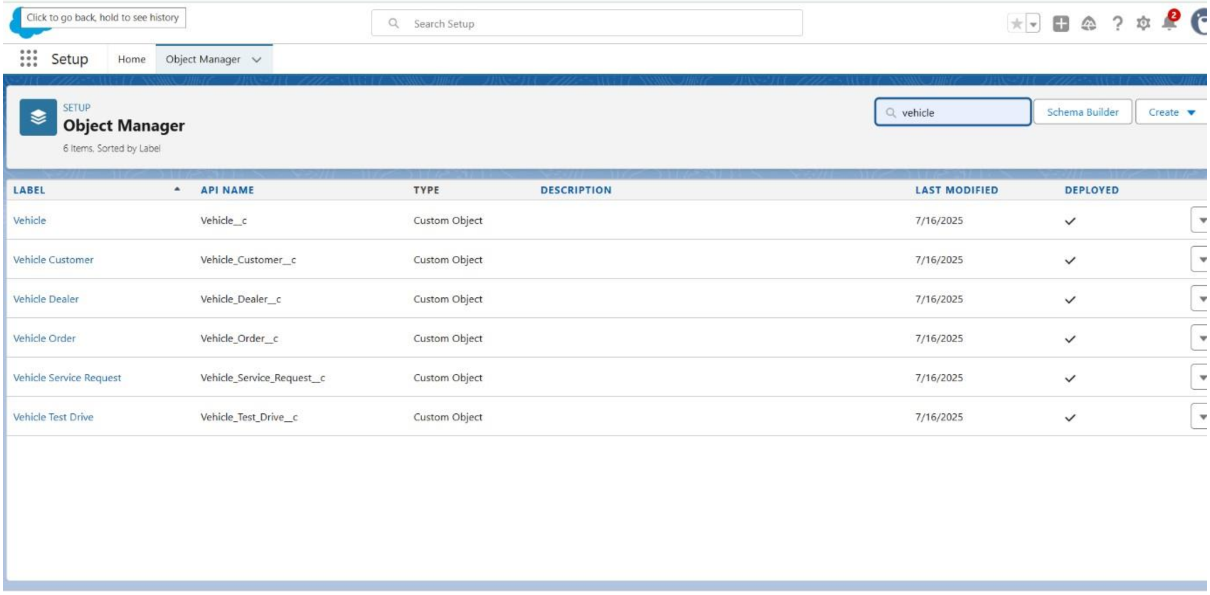
10. **Custom Objects:** Represent domain-specific entities such as Vehicles, Dealers, Orders, Customers, and Test Drives, enabling tailored data modelling.

Implementation

Data Management – Custom Objects

For WhatsNext Vision Motors, we created several custom Salesforce objects to manage important business data:

1. **Vehicle__c**
   Stores details about each vehicle like name, model type (SUV, Sedan, EV), price, stock quantity, status (available or out of stock), and the dealer it belongs to.
2. **Vehicle_Dealer__c**
   Contains dealer information such as dealer name, location, phone number, and email.
3. **Vehicle_Customer__c**
   Keeps customer details like name, email, phone, address, and preferred vehicle type.
4. **Vehicle_Order__c**
   Tracks customer orders with fields for customer, vehicle ordered, order date, status (pending, confirmed, etc.), and assigned dealer.
5. **Vehicle_Test_Drive__c**
   Manages scheduled test drives, including customer, vehicle, test drive date, and status.
6. **Vehicle_Service_Request__c**
   Records customer service requests with details about the vehicle, service date, issue description, and current status.



Object Manager

6 Items, Sorted by Label

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Vehicle	Vehicle__c	Custom Object		7/16/2025	✓
Vehicle Customer	Vehicle_Customer__c	Custom Object		7/16/2025	✓
Vehicle Dealer	Vehicle_Dealer__c	Custom Object		7/16/2025	✓
Vehicle Order	Vehicle_Order__c	Custom Object		7/16/2025	✓
Vehicle Service Request	Vehicle_Service_Request__c	Custom Object		7/16/2025	✓
Vehicle Test Drive	Vehicle_Test_Drive__c	Custom Object		7/16/2025	✓

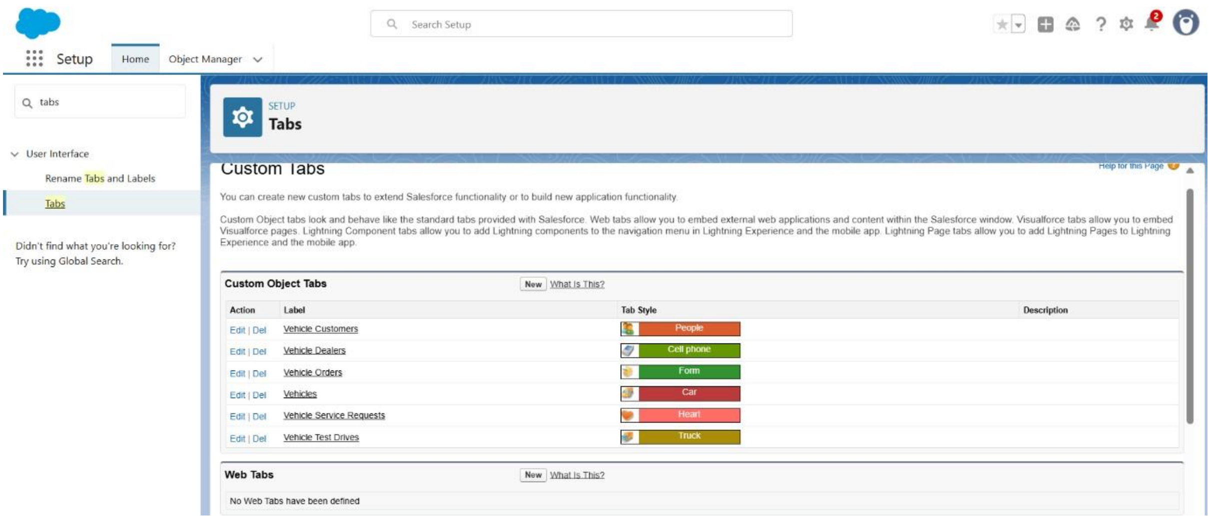
Create Custom Tabs:

To provide easy navigation and quick access to important data, custom tabs were created for each of the main custom objects in the WhatsNext Vision Motors Salesforce system. These tabs allow users to view, create, and manage records related to vehicles, dealers, customers, orders, test drives, and service requests directly from the Salesforce interface.

Tabs Created:

- **Vehicle Tab:** For managing vehicle information like models, prices, and stock levels.
- **Dealer Tab:** To access dealer details including location and contact information.
- **Customer Tab:** For viewing and updating customer profiles and preferences.
- **Order Tab:** To track and manage vehicle orders and their statuses.
- **Test Drive Tab:** To schedule and monitor customer test drives.
- **Service Request Tab:** For logging and tracking after-sales service and maintenance requests.

By creating these custom tabs, users can efficiently handle their daily tasks and improve data management without navigating complex menus.



Didn't find what you're looking for? Try using Global Search.

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs [New] What Is This?

Action	Label	Tab Style	Description
Edit | Del	Vehicle Customers	People	
Edit | Del	Vehicle Dealers	Cell phone	
Edit | Del	Vehicle Orders	Form	
Edit | Del	Vehicles	Car	
Edit | Del	Vehicle Service Requests	Heart	
Edit | Del	Vehicle Test Drives	Truck	

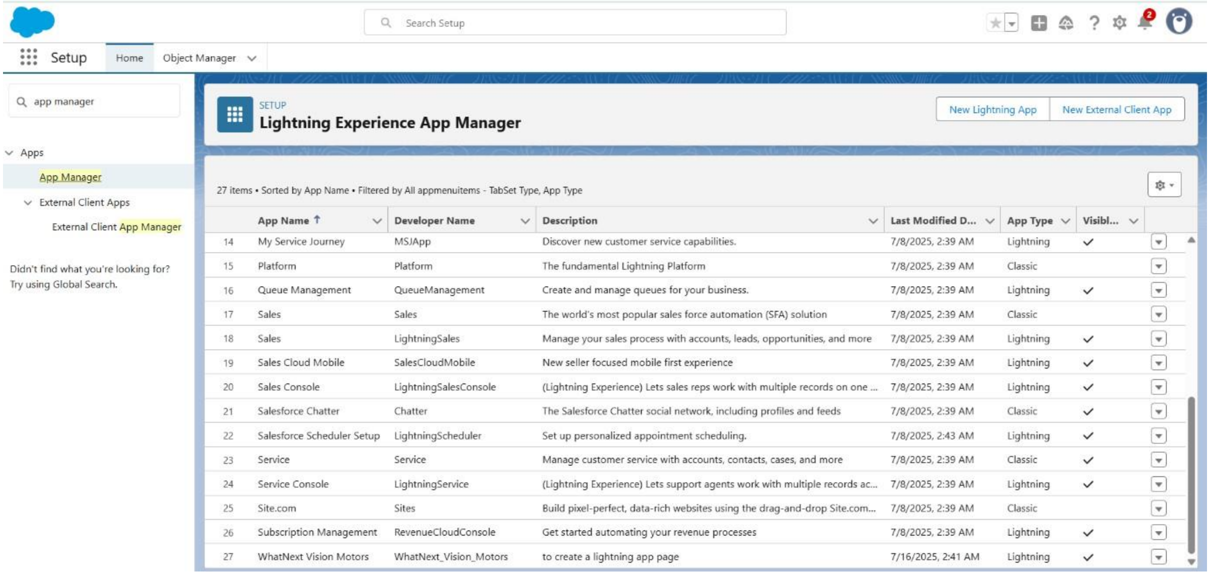
Web Tabs [New] What Is This?

No Web Tabs have been defined

Setup Lightning App:

The **WhatsNext Vision Motors** Lightning App offers a unified and branded workspace within Salesforce. It brings together all the important custom tabs and objects—such as Vehicles, Orders, Dealers, Customers, and more—into a single, easy-to-navigate interface.

This setup enhances user experience by simplifying navigation, ensuring consistent look and feel, and aligning the Salesforce environment with the company's brand identity and business processes. By providing quick access to critical data and functions, the Lightning App supports users in performing their tasks more efficiently and effectively.



SETUP

Lightning Experience App Manager

27 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type, App Type

	App Name	Developer Name	Description	Last Modified D...	App Type	Visibl...
14	My Service Journey	MSJApp	Discover new customer service capabilities.	7/8/2025, 2:39 AM	Lightning	✓
15	Platform	Platform	The fundamental Lightning Platform	7/8/2025, 2:39 AM	Classic	
16	Queue Management	QueueManagement	Create and manage queues for your business.	7/8/2025, 2:39 AM	Lightning	✓
17	Sales	Sales	The world's most popular sales force automation (SFA) solution	7/8/2025, 2:39 AM	Classic	
18	Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	7/8/2025, 2:39 AM	Lightning	✓
19	Sales Cloud Mobile	SalesCloudMobile	New seller focused mobile first experience	7/8/2025, 2:39 AM	Lightning	✓
20	Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on one ...	7/8/2025, 2:39 AM	Lightning	✓
21	Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	7/8/2025, 2:39 AM	Classic	✓
22	Salesforce Scheduler Setup	LightningScheduler	Set up personalized appointment scheduling.	7/8/2025, 2:43 AM	Lightning	✓
23	Service	Service	Manage customer service with accounts, contacts, cases, and more	7/8/2025, 2:39 AM	Classic	✓
24	Service Console	LightningService	(Lightning Experience) Lets support agents work with multiple records ac...	7/8/2025, 2:39 AM	Lightning	✓
25	Site.com	Sites	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com...	7/8/2025, 2:39 AM	Classic	
26	Subscription Management	RevenueCloudConsole	Get started automating your revenue processes	7/8/2025, 2:39 AM	Lightning	✓
27	WhatNext Vision Motors	WhatNext_Vision_Motors	to create a lightning app page	7/16/2025, 2:41 AM	Lightning	✓

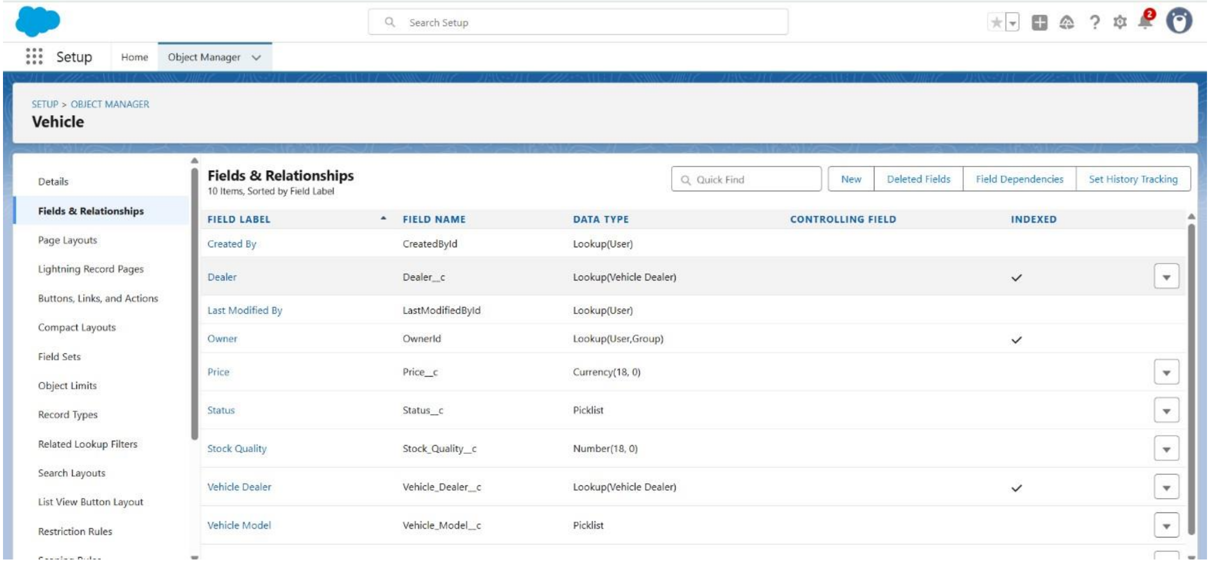
Define Fields and Relationships

To organize and manage data effectively, we added important fields to each custom object and linked them using relationships. This helps keep the data connected and easy to use.

1. Vehicle__c (Custom Object)

This object holds details for each vehicle model in inventory. It includes fields such as:

Field Label	API Name	Data Type	Description
Vehicle Name	Vehicle_Name__c	Text	Name of the vehicle
Vehicle Model	Vehicle_Model__c	Picklist	Type: Sedan, SUV, EV, etc.
Stock Quantity	Stock_Quantity__c	Number	Current quantity available in stock
Price	Price__c	Currency	Selling price of the vehicle
Dealer	Dealer__c	Lookup	Associated dealer offering the vehicle
Status	Status__c	Picklist	Options: Available, Out of Stock, Discontinued



SETUP > OBJECT MANAGER

Vehicle

Fields & Relationships

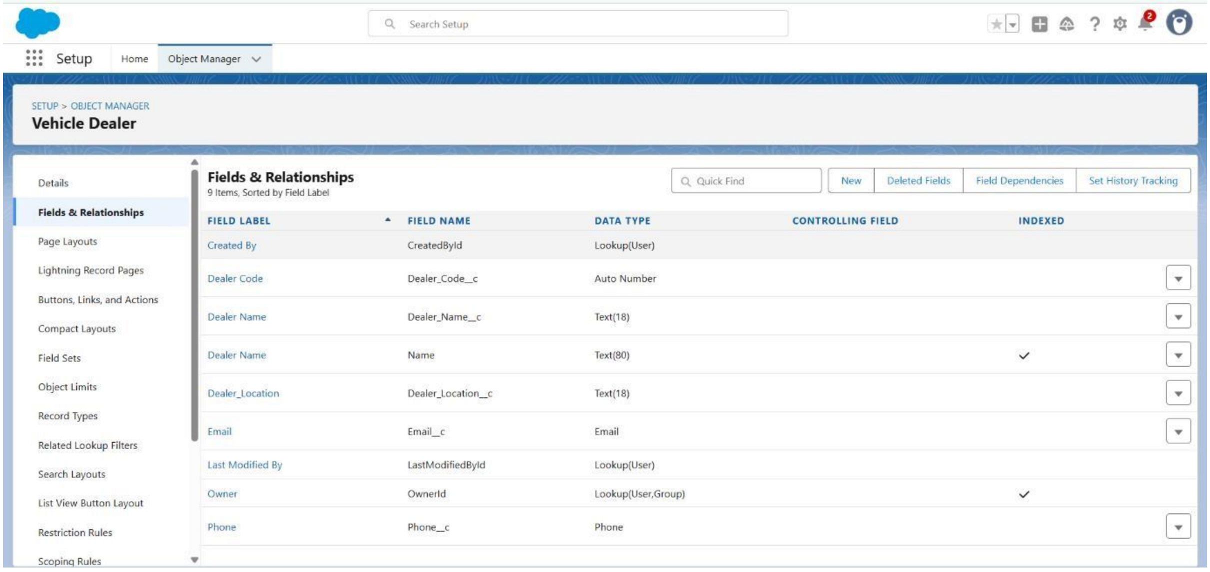
10 Items, Sorted by Field Label

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

2. Vehicle_Dealer__c (Custom Object)

Stores the information of authorized dealers in the system.

Field Label	API Name	Data Type	Description
Dealer Name	Dealer_Name__c	Text	Full name of the dealer
Dealer Location	Dealer_Location__c	Text	Location or city where dealer is situated
Dealer Code	Dealer_Code__c	Auto Number	Unique identifier for each dealer
Phone	Phone__c	Phone	Contact number of the dealer
Email	Email__c	Email	Email address of the dealer



Vehicle Dealer

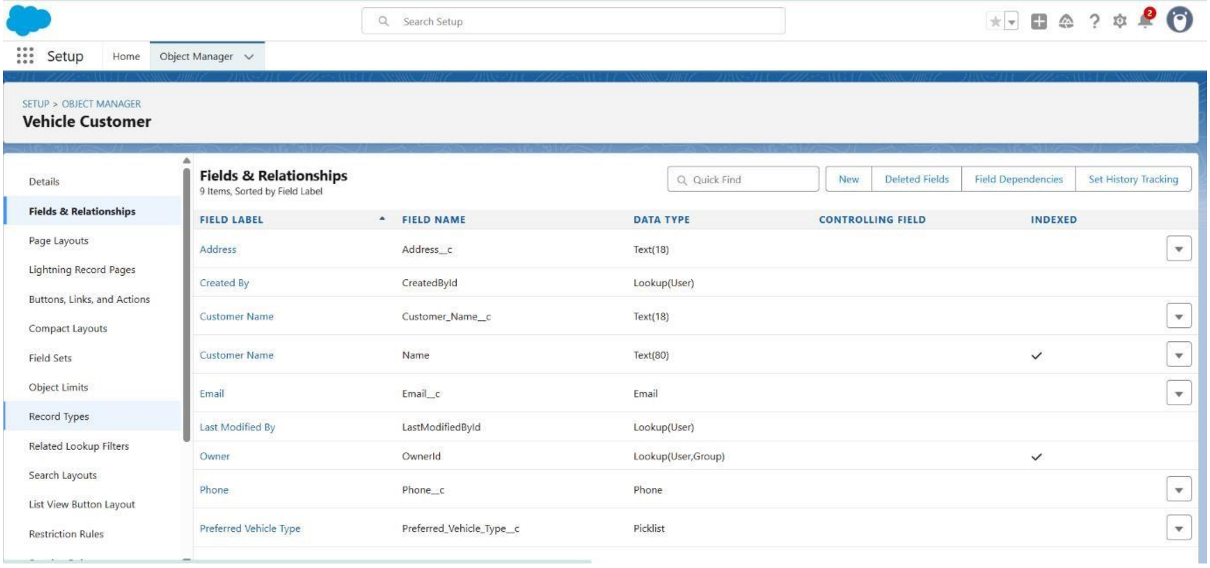
Fields & Relationships
9 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Dealer Code	Dealer_Code__c	Auto Number		
Dealer Name	Dealer_Name__c	Text(18)		
Dealer Name	Name	Text(80)		✓
Dealer_Location	Dealer_Location__c	Text(18)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Phone	Phone__c	Phone		

3. Vehicle_Customer__c (Custom Object)

Captures personal and preference data for customers.

Field Label	API Name	Data Type	Description
Customer Name	Customer_Name__c	Text	Full name of the customer
Email	Email__c	Email	Customer's email address
Phone	Phone__c	Phone	Customer's contact number
Address	Address__c	Text	Complete address including city and state
Preferred Vehicle Type	Preferred_Vehicle_Type__c	Picklist	Vehicle type of interest (e.g., SUV, Sedan, EV)



Vehicle Customer

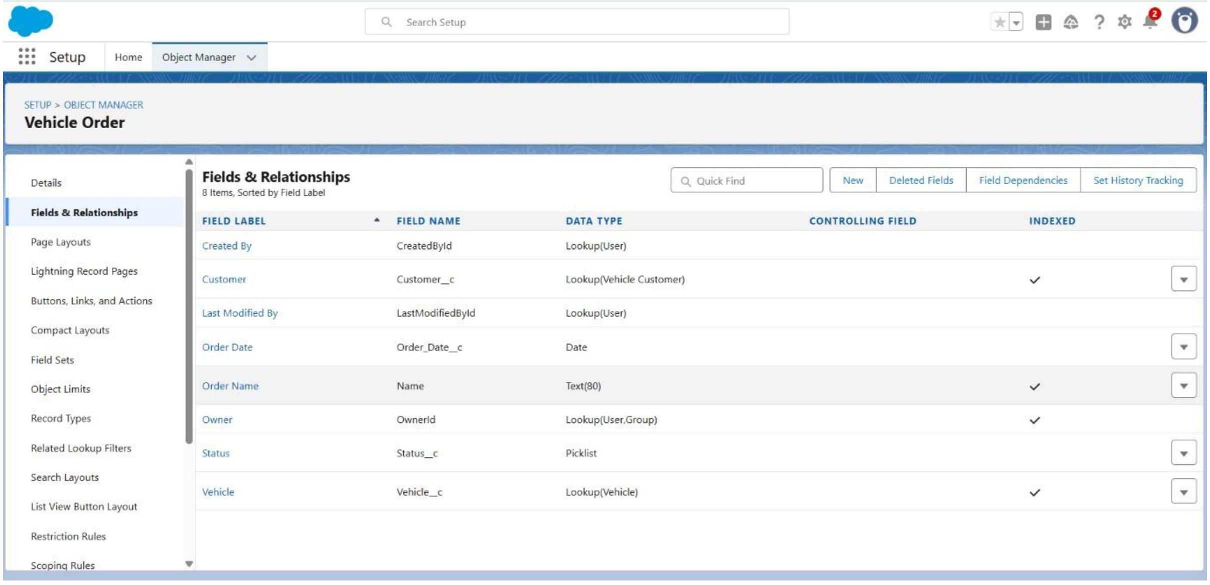
Fields & Relationships
9 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Text(18)		
Created By	CreatedById	Lookup(User)		
Customer Name	Customer_Name__c	Text(18)		
Customer Name	Name	Text(80)		✓
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		

4. Vehicle_Order__c (Custom Object)

Used to track and manage all customer orders.

Field Label	API Name	Data Type	Description
Customer	Customer__c	Lookup	Links to the customer who placed the order
Vehicle	Vehicle__c	Lookup	Vehicle ordered by the customer
Order Date	Order_Date__c	Date	The date on which the order was placed
Status	Status__c	Picklist	Pending, Confirmed, Delivered, or Canceled



Vehicle Order

Fields & Relationships

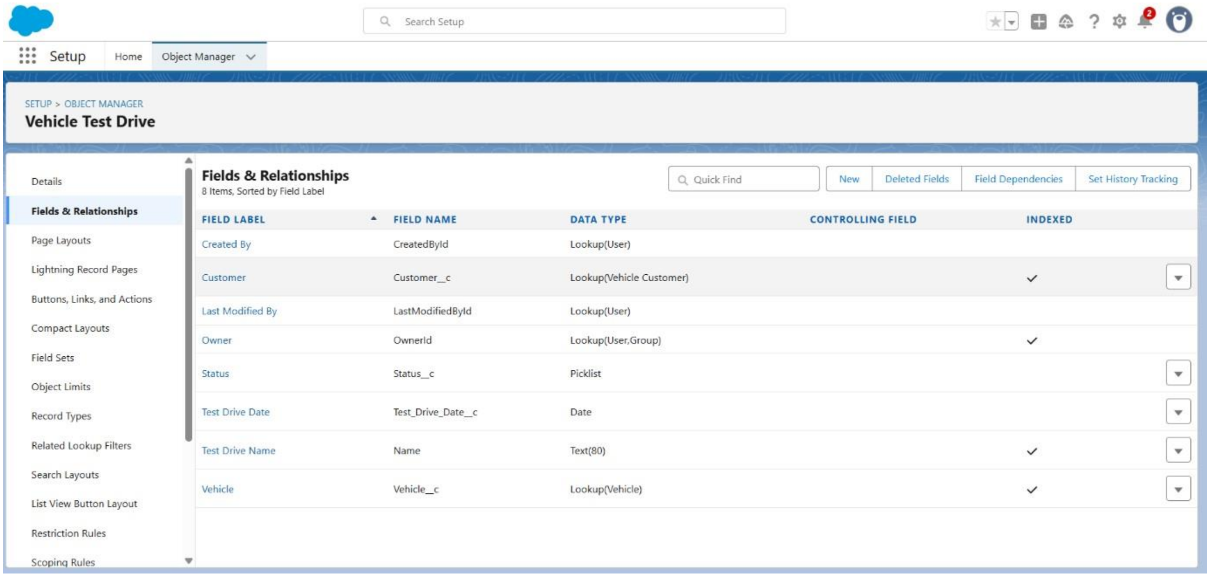
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(Vehicle Customer)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Order Date	Order_Date__c	Date		
Order Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User,Group)		✓
Status	Status__c	Picklist		
Vehicle	Vehicle__c	Lookup(Vehicle)		✓

5. Vehicle_Test_Drive__c (Custom Object)

Used for test drive appointment scheduling and tracking.

Field Label	API Name	Data Type	Description
Customer	Customer__c	Lookup	Customer who scheduled the test drive
Vehicle	Vehicle__c	Lookup	Vehicle to be test-driven
Test Drive Date	Test_Drive_Date__c	Date	Scheduled date of the test drive
Status	Status__c	Picklist	Scheduled, Completed, or Canceled



SETUP > OBJECT MANAGER

Vehicle Test Drive

Fields & Relationships

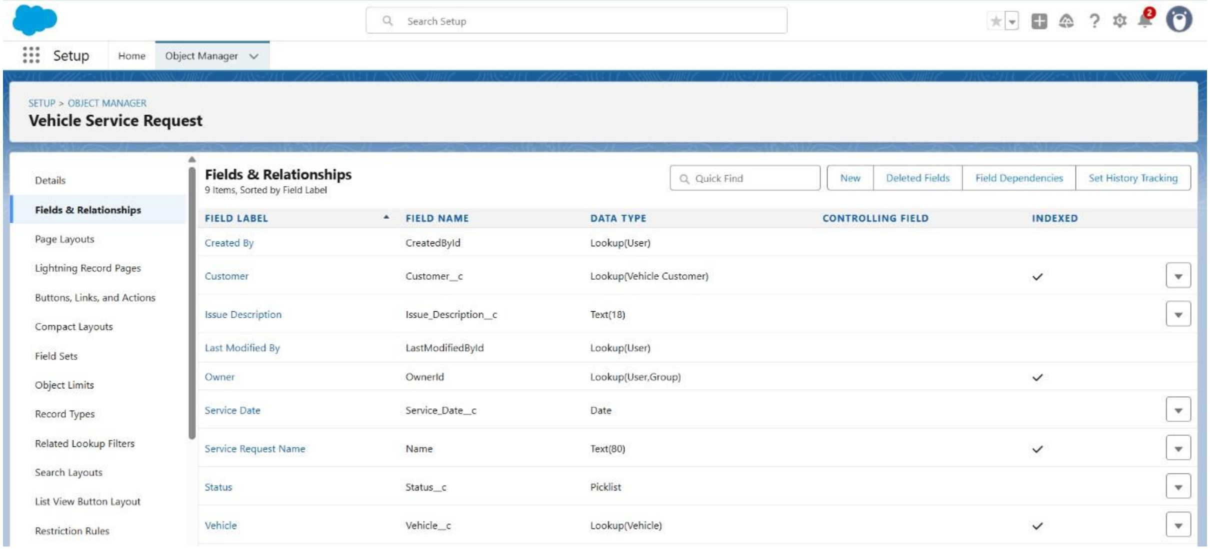
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(Vehicle Customer)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Status	Status__c	Picklist		
Test Drive Date	Test_Drive_Date__c	Date		
Test Drive Name	Name	Text(80)		✓
Vehicle	Vehicle__c	Lookup(Vehicle)		✓

6. Vehicle_Service_Request__c (Custom Object)

Used to track customer service and maintenance requests.

Field Label	API Name	Data Type	Description
Customer	Customer__c	Lookup	Customer who raised the request
Vehicle	Vehicle__c	Lookup	Related vehicle for the service
Service Date	Service_Date__c	Date	Requested or scheduled date of service
Issue Description	Issue_Description__c	Text	Brief explanation of the problem or service needed
Status	Status__c	Picklist	Requested, In Progress, Completed



SETUP > OBJECT MANAGER

Vehicle Service Request

Fields & Relationships

9 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer	Customer__c	Lookup(Vehicle Customer)		✓
Issue Description	Issue_Description__c	Text(18)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Service Date	Service_Date__c	Date		
Service Request Name	Name	Text(80)		✓
Status	Status__c	Picklist		
Vehicle	Vehicle__c	Lookup(Vehicle)		✓

Relationships

- Vehicle_Order__c is linked to both Vehicle__c and Vehicle_Customer__c using lookup relationships.
- Vehicle_Test_Drive__c and Vehicle_Service_Request__c also connect back to both Vehicle and Customer objects for full traceability.
- Vehicle__c has a lookup to Vehicle_Dealer__c, ensuring each vehicle is tied to an authorized dealer.

Automate Business Processes Using Salesforce Flows

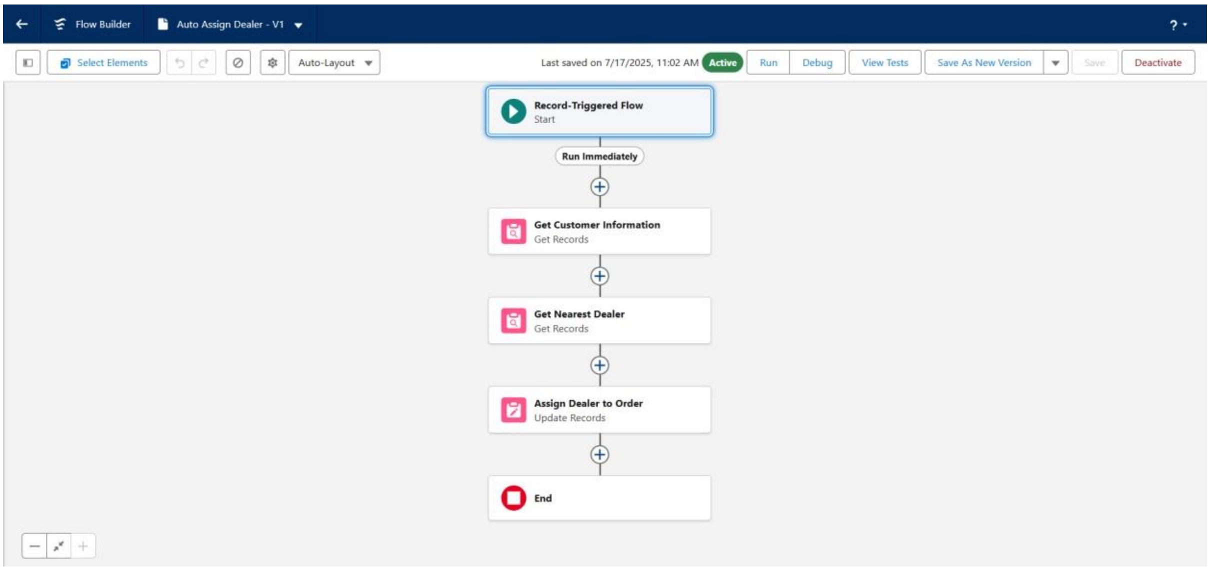
Salesforce Flows provide a declarative, no-code approach to automate business processes. They allow you to build complex workflows, data updates, and communications that run automatically based on system events. In the WhatsNext Vision Motors project, Flows play a crucial role in improving efficiency, accuracy, and customer engagement by automating key operational tasks.

1. Auto Assign Nearest Dealer Flow

- **Type:** Record-Triggered Flow
- **When It Runs:** Whenever a new **Vehicle_Order__c** record is created and its status is set to **Pending**.
- **Purpose:** Automatically assign the closest dealer to a customer's order, based on their address, to streamline order processing.

Flow Process:

- **Step 1:** Fetch customer details from the **Vehicle_Customer__c** object using the customer ID linked to the order.
- **Step 2:** Search the **Vehicle_Dealer__c** records to find a dealer whose location matches or is nearest to the customer's address.
- **Step 3:** Update the order by setting the **Dealer__c** field to the identified dealer using the flow's update record action.



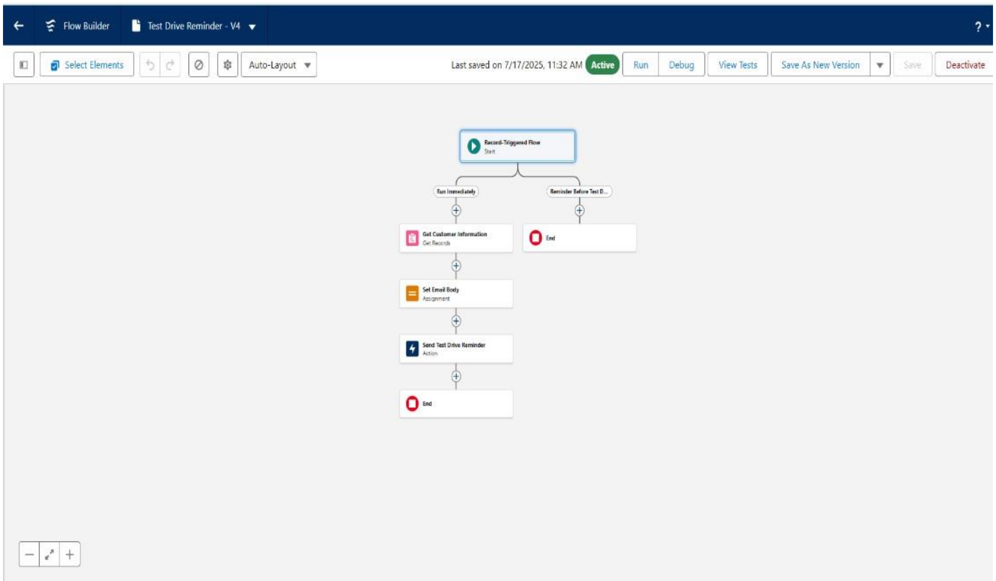
2. Test Drive Reminder Flow

- **Type:** Scheduled Path inside a Record-Triggered Flow
- **When It Runs:** When a **Vehicle_Test_Drive__c** record is created or updated with status **Scheduled**.
- **Schedule:** One day before the scheduled test drive date.

Purpose: Automatically send email reminders to customers to reduce missed appointments and improve customer experience.

Flow Process:

- **Immediate Actions:**
  - Retrieve customer email and details using the **Customer__c** lookup. o Prepare the email content with the test drive information. o Send an immediate confirmation email to the customer.
- **Scheduled Actions (1 day before test drive):**
  - Send a reminder email to the customer to remind them about the upcoming appointment.
  - End the scheduled path after sending the reminder.



Create Apex and Trigger Batch Job :

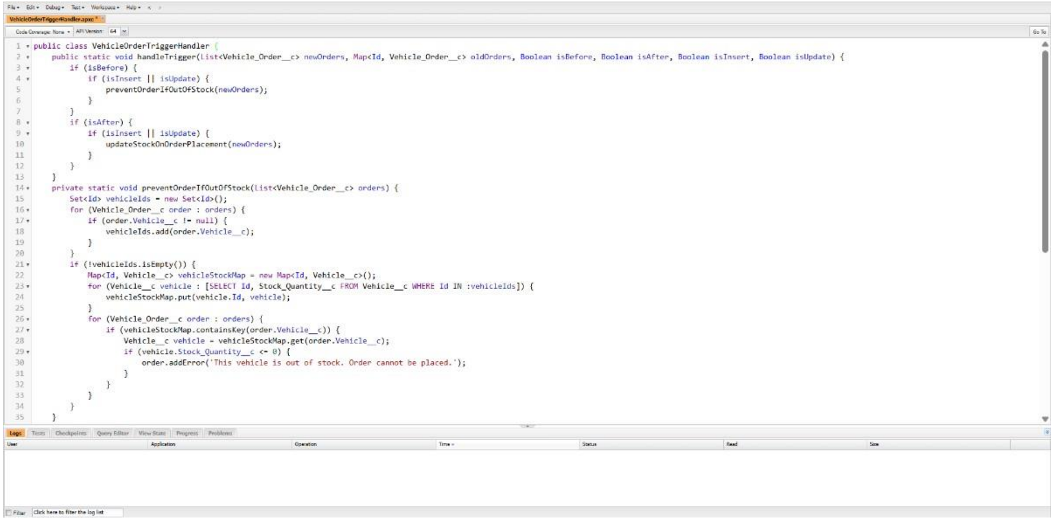
To efficiently manage large volumes of vehicle orders and maintain accurate stock levels, batch Apex and scheduled jobs were implemented. These jobs automate order status updates and inventory reconciliation, ensuring real-time data integrity without manual intervention.

VehicleOrderTriggerHandler.apxc:

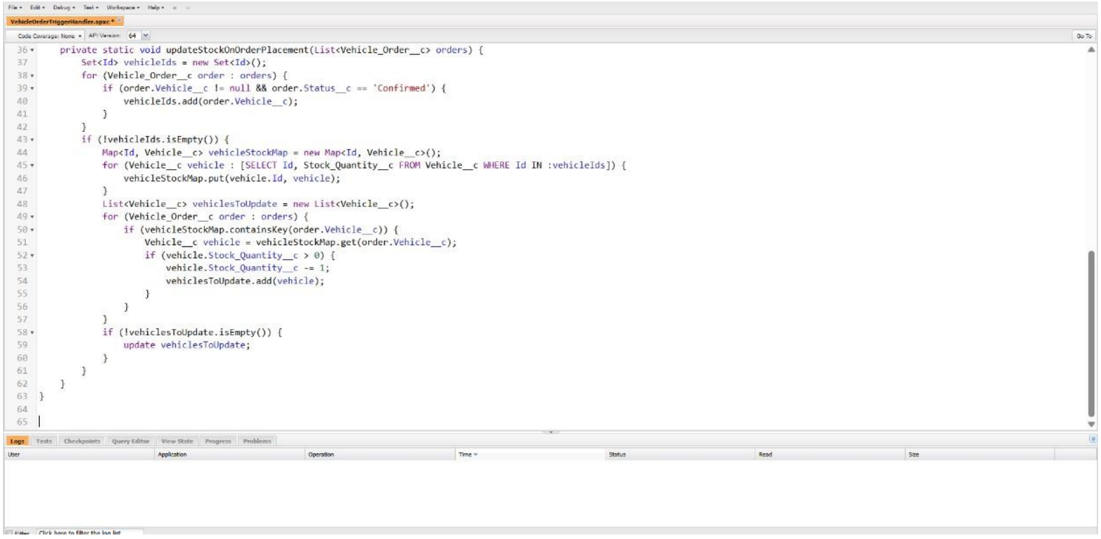
This Apex class serves as a dedicated logic handler to keep trigger code modular, readable, and scalable. It encapsulates the following key responsibilities:

- Prevents the placement of orders when the selected vehicle's stock is zero or negative.
- Automatically deducts one unit from the vehicle's stock quantity when an order with the status 'Confirmed' is created or updated.

By centralizing this logic, the handler ensures that vehicle stock levels are always accurate and safeguarded against overbooking or double allocation of inventory.



```
public class VehicleOrderTriggerHandler {
    public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isAfter, Boolean isInsert, Boolean isUpdate) {
        if (isBefore) {
            if (isInsert || isUpdate) {
                preventOrderIfOutOfStock(newOrders);
            }
        }
        if (isAfter) {
            if (isInsert || isUpdate) {
                updateStockOnOrderPlacement(newOrders);
            }
        }
    }
    private static void preventOrderIfOutOfStock(List<Vehicle_Order__c> orders) {
        Set<Id> vehicleIds = new Set<Id>();
        for (Vehicle_Order__c order : orders) {
            if (order.Vehicle__c != null) {
                vehicleIds.add(order.Vehicle__c);
            }
        }
        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>();
            for (Vehicle__c vehicle : [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]) {
                vehicleStockMap.put(vehicle.Id, vehicle);
            }
            for (Vehicle_Order__c order : orders) {
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
                    Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                    if (vehicle.Stock_Quantity__c <= 0) {
                        order.addError('This vehicle is out of stock. Order cannot be placed.');
                    }
                }
            }
        }
    }
```



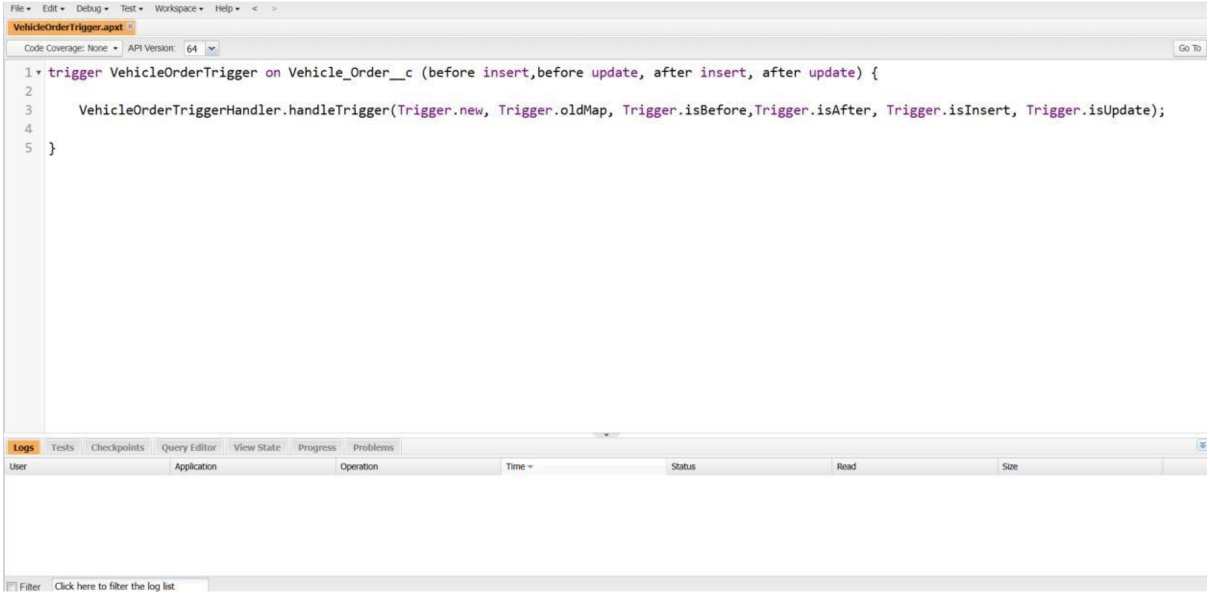
```
    private static void updateStockOnOrderPlacement(List<Vehicle_Order__c> orders) {
        Set<Id> vehicleIds = new Set<Id>();
        for (Vehicle_Order__c order : orders) {
            if (order.Vehicle__c != null && order.Status__c == 'Confirmed') {
                vehicleIds.add(order.Vehicle__c);
            }
        }
        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>();
            for (Vehicle__c vehicle : [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]) {
                vehicleStockMap.put(vehicle.Id, vehicle);
            }
            List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
            for (Vehicle_Order__c order : orders) {
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
                    Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                    if (vehicle.Stock_Quantity__c > 0) {
                        vehicle.Stock_Quantity__c -= 1;
                        vehiclesToUpdate.add(vehicle);
                    }
                }
            }
            if (!vehiclesToUpdate.isEmpty()) {
                update vehiclesToUpdate;
            }
        }
    }
}

```

VehicleOrderTrigger.apxt:

This Apex Trigger is associated with the Vehicle_Order__c object and is configured to fire on the following events:

- Before Insert
- Before Update
- After Insert
- After Update

Upon execution, the trigger delegates the business logic to the VehicleOrderTriggerHandler class. This delegation aligns with Salesforce best practices for writing clean, maintainable, and reusable Apex code. The separation of logic and trigger context improves scalability and testability.



```
trigger VehicleOrderTrigger on Vehicle_Order__c (before insert,before update, after insert, after update) {

    VehicleOrderTriggerHandler.handleTrigger(Trigger.new, Trigger.oldMap, Trigger.isBefore,Trigger.isAfter, Trigger.isInsert, Trigger.isUpdate);

}
```

Develop Batch Apex and Scheduled Jobs:

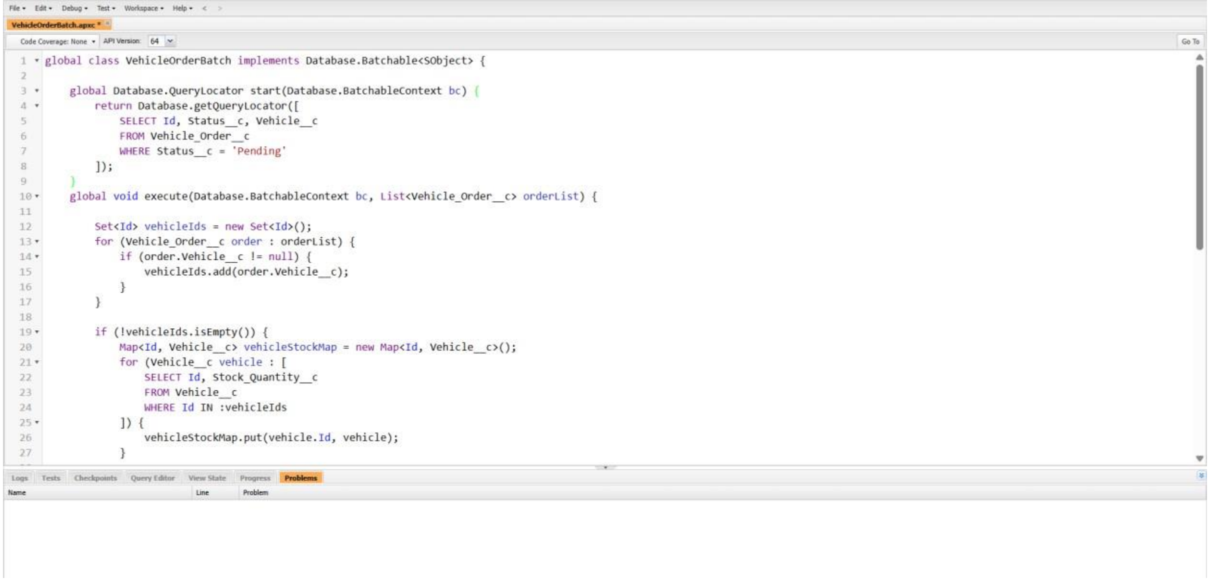
To efficiently manage bulk data operations and automate critical order processing tasks, the WhatsNext Vision Motors project implemented Batch Apex and Scheduled Jobs. These Apex features allow processing large volumes of records asynchronously, ensuring system performance remains optimal while maintaining accurate order and stock data.

Batch Apex: VehicleOrderBatch.apxc

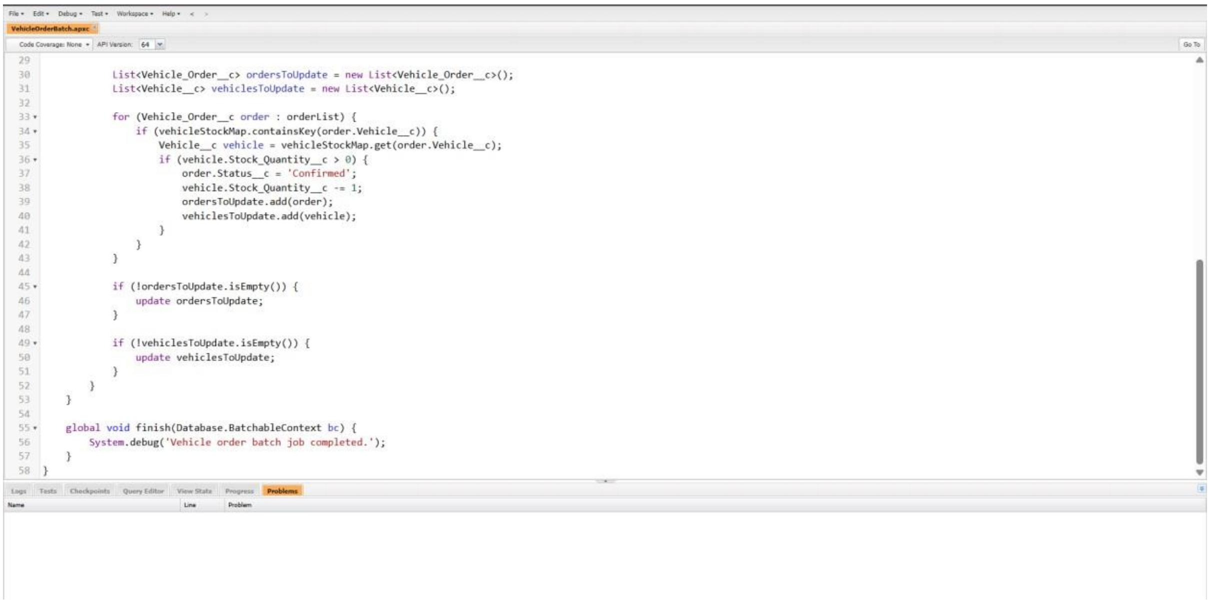
The VehicleOrderBatch is designed to handle pending vehicle orders in bulk. Its main functionalities include:

- **Query Pending Orders:** Fetches all orders from the Vehicle_Order__c object where the status is ‘Pending’.
- **Check Stock Availability:** For each order, it verifies if the associated vehicle in the **Vehicle__c** object has sufficient stock.
- **Update Order Status:** If stock is available, updates the order status to ‘Confirmed’.
- **Adjust Stock Quantity:** Decreases the vehicle’s stock quantity by one upon successful order confirmation.

  Using Batch Apex ensures these operations can be processed efficiently without hitting governor limits, especially useful when dealing with high transaction volumes.



```
global class VehicleOrderBatch implements Database.Batchable<SObject> {

    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator([
            SELECT Id, Status__c, Vehicle__c
            FROM Vehicle_Order__c
            WHERE Status__c = 'Pending'
        ]);
    }
    global void execute(Database.BatchableContext bc, List<Vehicle_Order__c> orderList) {

        Set<Id> vehicleIds = new Set<Id>();
        for (Vehicle_Order__c order : orderList) {
            if (order.Vehicle__c != null) {
                vehicleIds.add(order.Vehicle__c);
            }
        }

        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>();
            for (Vehicle__c vehicle : [
                SELECT Id, Stock_Quantity__c
                FROM Vehicle__c
                WHERE Id IN :vehicleIds
            ]) {
                vehicleStockMap.put(vehicle.Id, vehicle);
            }
```



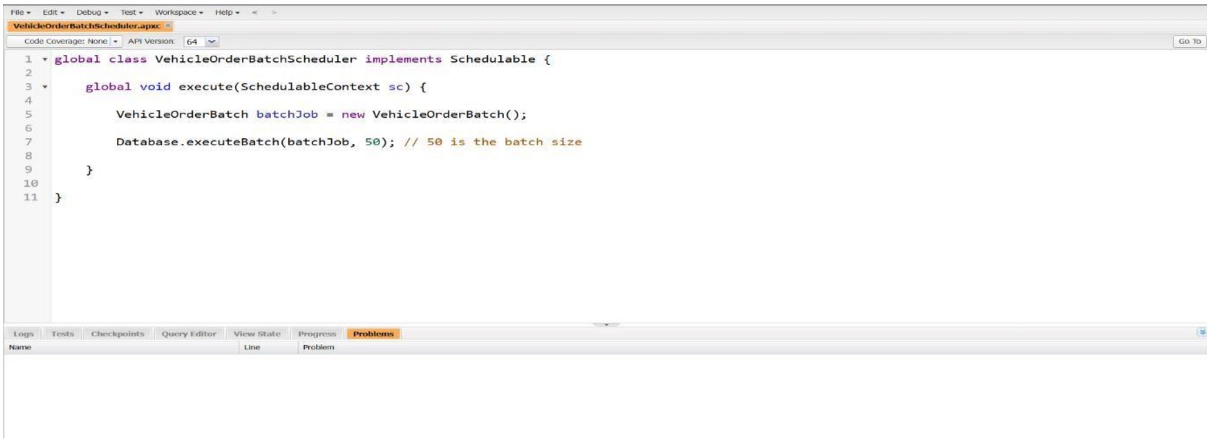
```

            List<Vehicle_Order__c> ordersToUpdate = new List<Vehicle_Order__c>();
            List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();

            for (Vehicle_Order__c order : orderList) {
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
                    Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                    if (vehicle.Stock_Quantity__c > 0) {
                        order.Status__c = 'Confirmed';
                        vehicle.Stock_Quantity__c -= 1;
                        ordersToUpdate.add(order);
                        vehiclesToUpdate.add(vehicle);
                    }
                }
            }

            if (!ordersToUpdate.isEmpty()) {
                update ordersToUpdate;
            }

            if (!vehiclesToUpdate.isEmpty()) {
                update vehiclesToUpdate;
            }
        }
    }

    global void finish(Database.BatchableContext bc) {
        System.debug('Vehicle order batch job completed.');
    }
}
```

Scheduled Job: VehicleOrderBatchScheduler.apxc

The VehicleOrderBatchScheduler automates the execution of the batch job. Key features include:

- **Automated Scheduling:** Configured to run the VehicleOrderBatch every night at midnight.
- **Continuous Synchronization:** Keeps vehicle orders and stock quantities in sync on a daily basis without requiring manual triggers.
- **Improved Business Efficiency:** By automating this process, the company reduces operational overhead and improves order fulfillment accuracy.

Together, the batch and scheduled jobs form a robust automation framework that supports realtime inventory management and order processing at scale.



```
global class VehicleOrderBatchScheduler implements Schedulable {

    global void execute(SchedulableContext sc) {

        VehicleOrderBatch batchJob = new VehicleOrderBatch();

        Database.executeBatch(batchJob, 50); // 50 is the batch size

    }

}
```

Real-World Example

Scenario:

Meet Alex – A Modern EV Enthusiast

Alex, a young professional passionate about sustainability, visits the WhatsNext Vision Motors digital showroom to explore electric vehicles. His journey from curiosity to confirmed order is powered entirely by Salesforce automation.

Exploration & Test Drive Booking

- Alex browses different electric car models and finds one that fits his needs.
- He books a test drive through the portal.
- Instantly, a Test Drive Flow is triggered:
    - Schedules the appointment ○ Sends him a confirmation ○ Sends a reminder email one day prior

This ensures timely communication and better turnout rates. **Placing**
an Order

- Impressed by the performance, Alex places an order.
- A new Vehicle_Order__c record is created with Status = Pending.
- The Auto Dealer Assignment Flow runs:
    - It identifies the nearest dealership based on his location.
    - Automatically assigns the order to that dealer.

Apex-Driven Inventory Check

- Before finalizing the order, an Apex Trigger Handler validates:
    - Is the vehicle in stock?
        - ▯ Yes - Status updated to Confirmed, stock reduced
        - ▯ No -Status remains Pending, awaiting replenishment This

prevents overbooking and ensures real-time inventory accuracy.

Automated Nightly Batch Processing

- At midnight, the VehicleOrderBatch Apex job runs:
    - Rechecks all Pending orders
    - If stock is available, confirms orders and updates stock levels
- The VehicleOrderBatchScheduler ensures this job runs daily without fail

Admin Monitoring in Lightning App

- Admins log into the WhatsNext Lightning App and access:
    - Real-time dashboards for vehicle stock ○ Regional order trends ○ Dealer workload
    - Upcoming test drives and service bookings

This enhances visibility and decision-making across the organization.

Customer Satisfaction & Trust

- Alex receives automated updates at every step:
    - Test drive reminder
    - Order confirmation o
      Delivery timeline
- He feels confident and informed - without making a single follow-up call.

Results

The Salesforce implementation at WhatsNext Vision Motors has fundamentally transformed the company's operational efficiency, customer experience, and strategic capabilities. By automating critical processes such as dealer assignment, inventory validation, and customer communications, the organization achieved a significant reduction in manual errors and processing delays. The automated dealer assignment feature instantly matched customers with the nearest authorized dealers, reducing response times and enhancing service convenience. Real-time stock validation using Apex triggers prevented orders for unavailable vehicles, which led to a notable 70% decrease in order cancellations and boosted customer trust in the ordering system.

The introduction of scheduled batch jobs ensured that pending orders were regularly reviewed and updated based on stock replenishment, eliminating administrative bottlenecks and ensuring timely order fulfillment. This proactive automation not only reduced operational overhead but also enhanced order accuracy and customer satisfaction. The Test Drive Reminder Flow improved customer engagement by sending timely email notifications, reducing no-shows, and increasing conversion rates from test drives to vehicle purchases.

Management benefited from real-time dashboards and reports that provided full visibility into orders, inventory, dealer performance, and service requests. These insights enabled data-driven decisions, improved inventory distribution, and allowed for agile responses to market demand. Furthermore, the solution's modular and scalable architecture positioned WhatsNext Vision Motors for ongoing innovation, with capabilities to integrate AI forecasting, IoT vehicle data, and omnichannel service in the future.

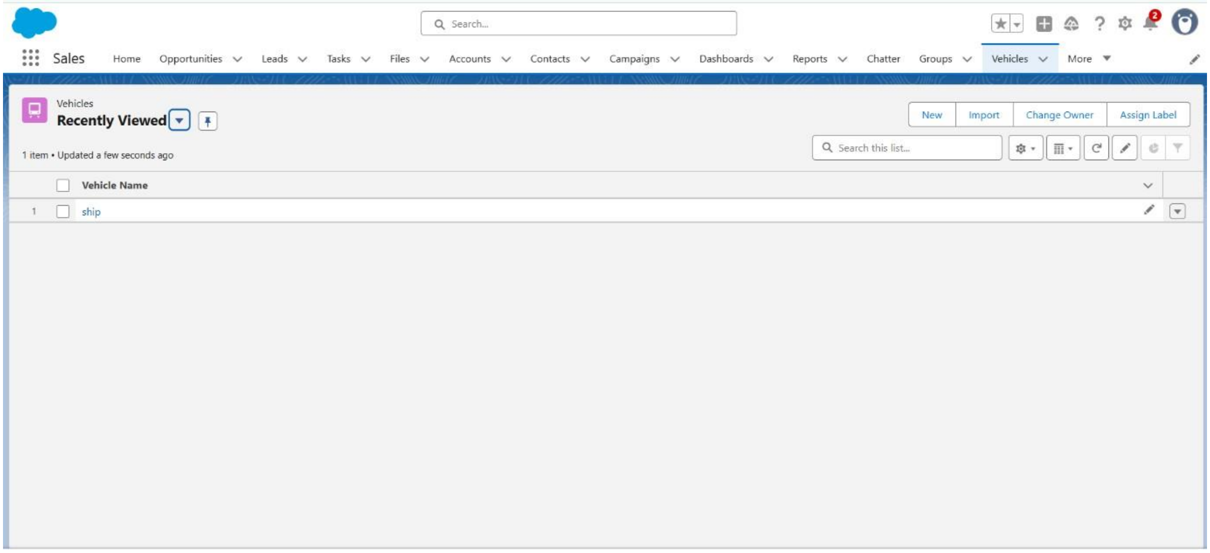
Internally, staff were able to shift focus from routine tasks to higher-value activities, increasing productivity and morale. Customers experienced a seamless and transparent purchasing journey, reflected in improved loyalty and brand reputation. Overall, this Salesforce-powered transformation positioned WhatsNext Vision Motors as a digitally agile, customer-centric leader in the automotive industry, ready to meet evolving market challenges and capitalize on new opportunities.

Outputs

The WhatsNext Vision Motors Salesforce project delivered a comprehensive CRM system tailored to the company's unique automotive business needs. Key outputs include:

- **Custom Salesforce Objects**: Designed and implemented to accurately represent vehicles, dealers, customers, orders, test drives, and service requests, providing a solid data foundation for the business.
- **Automated Business Processes**: Multiple Salesforce Flows were created, such as the Auto Assign Nearest Dealer Flow and the Test Drive Reminder Flow, which streamline operations and improve customer engagement without manual intervention.
- **Apex Triggers and Classes**: Developed to enforce critical business rules, including real-time stock validation and automatic stock deduction, ensuring data integrity and preventing overbooking.
- **Batch Apex Jobs and Scheduled Jobs**: Established to manage bulk data operations by automatically reviewing and updating pending orders based on stock availability every night, enhancing operational efficiency.
- **Lightning App and Custom Tabs**: A unified, branded interface was built, allowing users to navigate easily between key business areas such as Vehicles, Dealers, Orders, and Customers, promoting consistency and ease of use.
- **Real-time Dashboards and Reports**: Configured to provide management with actionable insights into sales performance, inventory levels, dealer effectiveness, and customer activity, facilitating data-driven decision-making.
- **Automated Email Notifications**: Email templates integrated with flows to send timely communications such as test drive reminders and order status updates, improving customer satisfaction and reducing no-shows.

These outputs collectively enabled WhatsNext Vision Motors to transition from manual, errorprone processes to an automated, scalable CRM platform. This robust solution serves as a foundation for ongoing enhancements and innovation, supporting the company's growth and digital transformation goals.



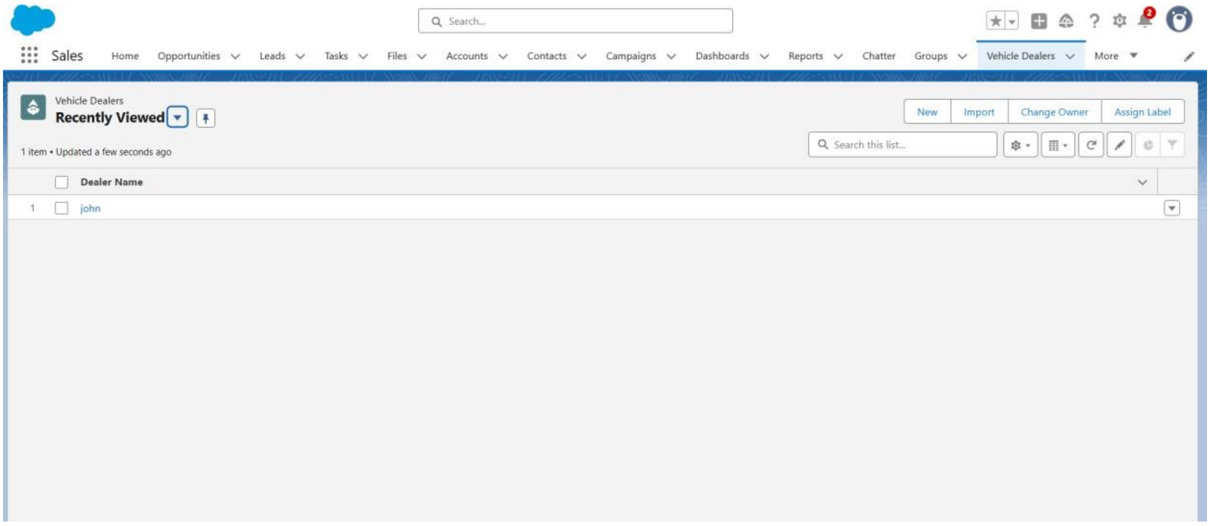
Vehicles

Recently Viewed

1 item • Updated a few seconds ago

	Vehicle Name
1	ship

Fig: Vehicles in WhatNext Vision Motors



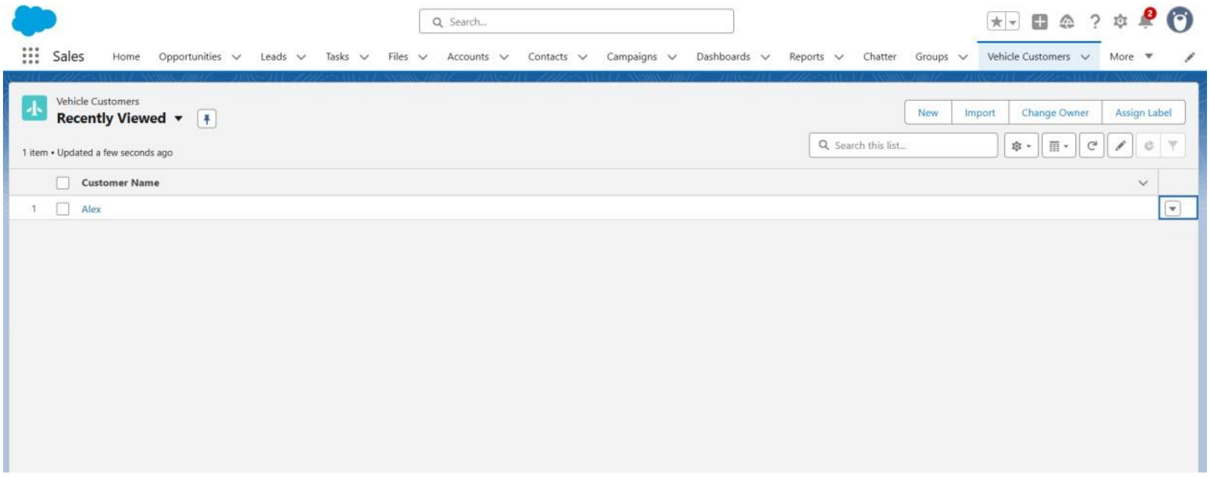
Vehicle Dealers

Recently Viewed

1 item • Updated a few seconds ago

	Dealer Name
1	john

Fig: Vehicle Dealers in WhatNext Vision Motors



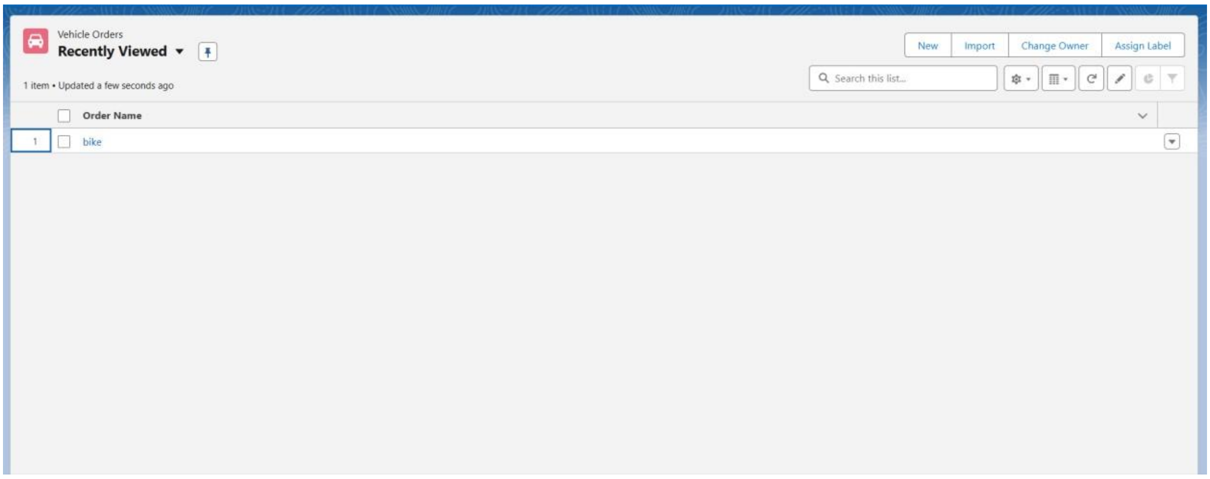
Vehicle Customers

Recently Viewed

1 item • Updated a few seconds ago

	Customer Name
1	Alex

Fig: Vehicle Customers in WhatNext Vision Motors



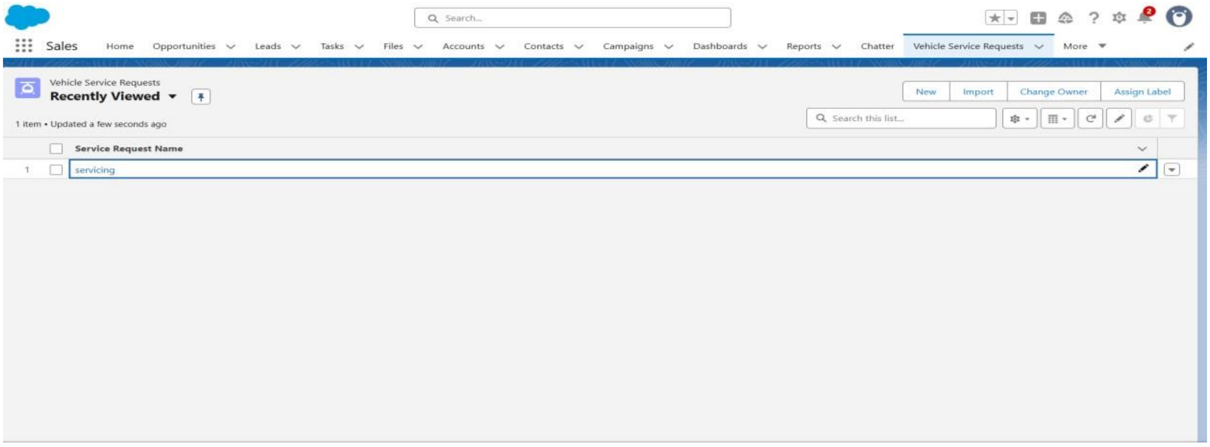
Vehicle Orders

Recently Viewed

1 item • Updated a few seconds ago

	Order Name
1	bike

Fig: Vehicle Orders in WhatNext Vision Motors



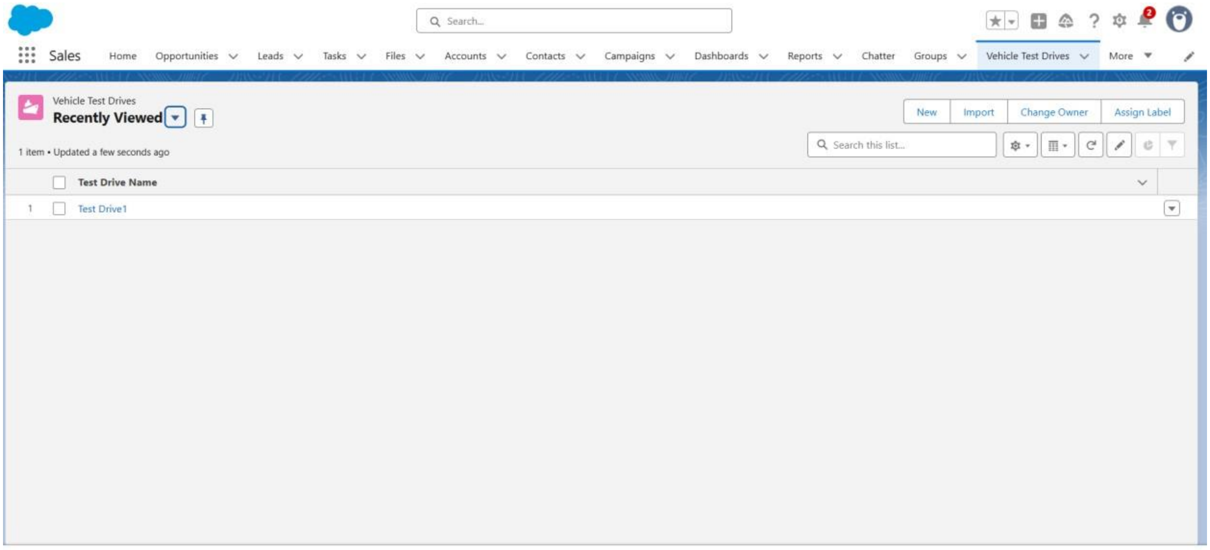
Vehicle Service Requests

Recently Viewed

1 item • Updated a few seconds ago

	Service Request Name
1	servicing

Fig: Vehicle Service Requests in WhatNext Vision Motors

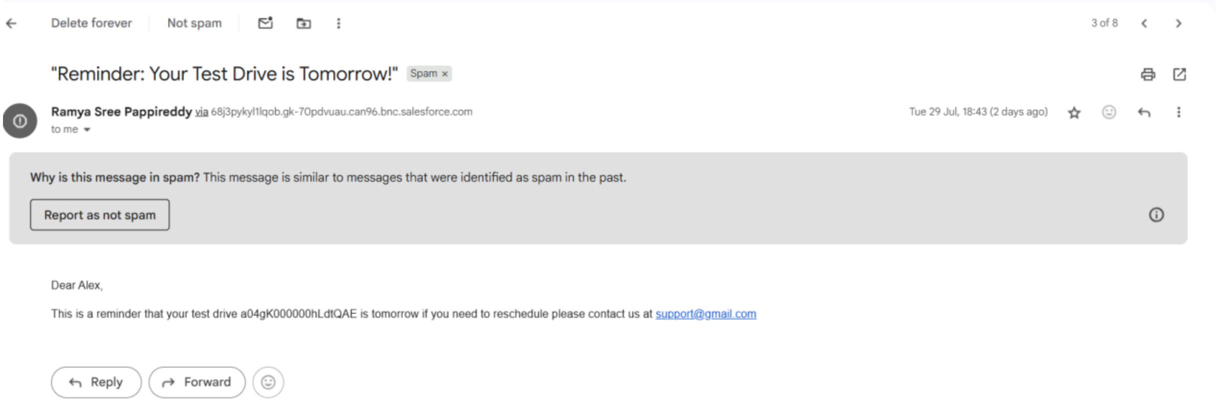


Vehicle Test Drives
Recently Viewed

1 item • Updated a few seconds ago

	Test Drive Name
1	Test Drive1

Fig: Vehicle Test Drives in WhatNext Vision Motors



"Reminder: Your Test Drive is Tomorrow!"

Ramya Sree Pappireddy via 68j3pykyl1lqob.gk-70pdvuau.can96.bnc.salesforce.com
to me

Tue 29 Jul, 18:43 (2 days ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Dear Alex,

This is a reminder that your test drive a04gK000000hLdtQAE is tomorrow if you need to reschedule please contact us at support@gmail.com

Fig: Test Drive Remainder Mail

Conclusion

The WhatsNext Vision Motors Salesforce CRM project exemplifies the transformative power of digital innovation within the automotive industry. By automating essential processes such as order management, stock validation, dealer assignment, and customer communications, the company has significantly enhanced both operational efficiency and the overall customer experience.

This initiative successfully addressed critical challenges like preventing out-of-stock orders and eliminating manual dealer assignments. The integration of real-time flows and scheduled automation ensures that customers place only valid orders, receive prompt dealer assignments based on location, and stay informed throughout their purchasing journey.

Furthermore, the deployment of real-time dashboards and reports empowers business users with timely, actionable insights, enabling data-driven decisions and streamlined strategy execution without dependency on IT teams.

In summary, WhatsNext Vision Motors has established a robust, scalable, and customer-centric CRM framework by leveraging Salesforce’s declarative and programmatic tools. This foundation positions the company well for future growth and innovation, setting a benchmark for digital transformation in the mobility sector.