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

ABSTRACT

WhatNext Vision Motors is an innovative mobility-tech project that aims to revolutionize the transportation industry through the integration of Artificial Intelligence (AI), Internet of Things (IoT), and Electric Vehicle (EV) technologies. The project envisions a future where vehicles are not just modes of transport but intelligent, connected systems that enhance safety, sustainability, and user experience. By developing AI-powered smart vehicles with autonomous navigation, predictive maintenance, and natural language interaction, the project delivers futuristic solutions tailored to modern mobility challenges.

The vehicles are built on a modular EV architecture using recyclable materials, solar-augmented charging systems, and fast-charging technology. Additionally, IoT sensors enable real-time diagnostics, over-the-air updates, and smart fleet coordination. Augmented Reality (AR) is used to create immersive infotainment and driver assistance through heads-up displays (HUDs).

WhatNext Vision Motors also introduces an inclusive, scalable design model suitable for both urban and rural environments, including use cases for smart campuses, logistics, and elderly mobility. With a strong commitment to carbon-negative operations and green energy partnerships, the project aligns with global sustainability goals. Ultimately, WhatNext Vision Motors is not just shaping vehicles it is redefining the future of mobility with innovation, intelligence, and ecological responsibility at its core.

OBJECTIVE

The primary objective of the WhatNext Vision Motors project is to build a comprehensive Salesforce-based CRM and operations system that empowers the automotive business to manage its vehicle sales, customer relationships, and test drive processes efficiently. The project aims to digitize and streamline every key operation from vehicle listings and dealer authorizations to customer onboarding and test drive scheduling ensuring real-time visibility, reduced manual errors, and enhanced customer satisfaction through intelligent automation.

Goals:

- Create a centralized system to manage vehicles, dealers, customers, and orders.
- Automate test drive scheduling and send timely reminders via Salesforce Flows.
- Ensure data accuracy and integrity through validation rules on key fields.
- Implement Apex Triggers to update vehicle availability and customer status dynamically.
- Provide role-based access for dealers, customers, and management via custom profiles.
- Develop dashboards and reports to track sales trends, test drives, and customer engagement.
- Enable seamless user navigation through custom Lightning App and tab structure.

TECHNOLOGY DESCRIPTION

The WhatNext Vision Motors project is developed on the Salesforce platform, aiming to modernize and automate the automotive sales and service experience. By leveraging both declarative tools and programmatic customization, the solution streamlines core business functions such as vehicle management, customer engagement, dealer operations, order processing, test drives, and service requests.

The following Salesforce technologies and components were used in this project:

1.Custom Objects :

Custom Objects were created to support the unique data requirements of WhatNext Vision Motors. These objects include:

- **Vehicle__c** – Stores all vehicle-related data such as model name, type, price, engine specifications, and stock availability.
- **Vehicle_Customer__c** – Maintains information about customers, their preferences, and interactions with WhatNext.
- **Vehicle_Dealer__c** – Captures data about authorized dealers and their assigned vehicle stock.
- **Vehicle_Order__c** – Tracks vehicle purchases, payment status, and delivery progress.
- **Vehicle_Test_Drive__c** – Records test drive bookings, time slots, and feedback from customers.

Each object is related through lookups and master-detail relationships to maintain data integrity and represent real-world connections between customers, vehicles, dealers, and orders.

2.Tabs

Custom Tabs were created to provide quick and organized access to all major objects in the WhatNext Vision CRM app. These tabs allow users to efficiently view, create, and manage records relevant to their roles.

Key Tabs:

- **Vehicle Tab** – Lists all available vehicle models with specs and pricing.
- **Customer Tab** – Stores customer profiles and preferences.
- **Dealer Tab** – Displays authorized dealers and their inventory.
- **Order Tab** – Tracks vehicle purchase orders and delivery status.
- **Test Drive Tab** – Manages test drive bookings and schedules.
- **Service Request Tab** – Logs and tracks vehicle service bookings.

Each tab is customized with relevant list views, quick actions, and Lightning pages to enhance user experience and streamline business processes.

3.Lightning App

In the WhatNext Vision Motors project, a custom Lightning App named "WhatNext Vision CRM" was built to unify all key functionalities into a single, streamlined workspace.

Key Highlights:

- **Included Custom Tabs** like Vehicle, Customer, Dealer, Order, Test Drive, and Service Request for easy navigation.
- **Tailored App Settings** based on user profiles (Sales Executive, Dealer, Manager, etc.) for role-specific access.
- **Lightning Navigation Bar** designed for quick access to high-priority objects and records.
- Integrated with **custom branding** and icons for a professional and intuitive interface.
- Optimized for both **desktop and mobile views**, ensuring flexibility across devices.

This app served as the central hub for all users, enabling seamless execution of sales, service, and dealership operations.

4.Fields & Relationships

In the WhatNext Vision Motors project, custom fields and relationships were defined across all key objects to ensure structured data storage, seamless automation, and meaningful reporting.

Custom Fields :-

Each object was enhanced with custom fields to capture specific business information. Examples include:

- Vehicle__c - Model_Name, Fuel_Type, Variant, Color, Price, Stock_Status
- Vehicle_Customer__c - Customer_Name, Phone_Number, Email, Preferred_Model, Previous_Purchase
- Vehicle_Order__c - Order_Date, Delivery_Status, Payment_Mode, Total_Amount
- Vehicle_Test_Drive__c - Scheduled_Date, Test_Drive_Status, Vehicle_Selected, Assigned_Executive
- Vehicle_Service_Request__c - Service_Type, Requested_Date, Service_Status, Assigned_Manager

These fields were configured with data types, picklists, formulas, and validation rules to support accuracy and business logic.

Relationships :-

Proper object relationships were implemented to represent real-world connections and ensure data integrity:

- Lookup Relationships
    - Vehicle_Order__c → Vehicle__c, Vehicle_Customer__c
    - Vehicle_Test_Drive__c → Vehicle__c, Vehicle_Customer__c
    - Vehicle_Service_Request__c → Vehicle__c, Vehicle_Customer__c
- Master-Detail Relationships (where control over child records was required)
    - For tightly bound objects like service history tied to customers or orders linked to a specific vehicle.

These relationships enable roll-up summaries, related lists, and automated updates, making the data model both powerful and scalable.

5.Profiles and Role Hierarchies :-

Profile: A profile in Salesforce defines a user's permissions, object access, and interface settings, controlling what they can do in the system.

Role: A role in Salesforce determines a user's record-level access and data visibility based on their position in the role hierarchy, controlling what data they can see.

Custom **Profiles** and **Roles** were implemented to control access:

- **Sales Executives** – Manage leads, orders, test drives, and customer data.
- **Dealers** – View and update stock, accept test drives, and coordinate deliveries.
- **Service Managers** – Manage and assign service requests and maintain customer satisfaction.
- **Managers/Admins** – Full access to dashboards, all records, and configuration.

Security settings included **field-level security**, **record-level access** via sharing rules, and **object permissions** for fine-grained control.

6.Flows and Process Automation :-

To streamline key business actions and improve customer experience, the WhatNext Vision Motors project implemented two essential record-triggered flows as part of the automation strategy. These flows ensured timely communication and intelligent assignment of records based on logic, reducing manual intervention.

1. Assign Nearest Dealer to Customer (Record-Triggered Flow)

- **Objective**: Automatically assign the nearest available dealer based on the customer's location.
- **Trigger**: When a new **Vehicle_Customer__c** record is created.
- **Logic**: The flow compares the customer’s location (city or region field) with the dealer locations and assigns the most relevant dealer to the record.
- **Outcome**: Ensures faster lead follow-up and regional dealer mapping without manual effort.

2. Test Drive Reminder Email (Record-Triggered Flow)

- **Objective**: Send an automated reminder email to customers 24 hours before their scheduled test drive.
- **Trigger**: When a **Vehicle_Test_Drive__c** record is created or updated with a scheduled date.
- **Logic**: The flow checks the scheduled date and uses a scheduled path to send a reminder email exactly one day prior to the test drive.
- **Outcome**: Improves customer experience and reduces no-shows for test drives.

7.Apex Triggers and Classes

To handle complex business logic and ensure scalable data processing, the WhatNext Vision Motors project utilized Apex Triggers and Batch Apex Classes. These components allowed automation beyond point-and-click tools like Flows, especially for bulk operations and system-level processes.

1. VehicleOrderTrigger.apxt (Trigger)

- **Purpose**: This is the main trigger on the Vehicle_Order__c object.
- **Function**: It listens for before insert, before update, and after insert events to execute logic related to vehicle order validation and data handling.
- **Delegation**: Instead of writing logic directly in the trigger, it delegates the operations to a trigger handler class (VehicleOrderTriggerHandler), following best practices.

2. VehicleOrderTriggerHandler.apxc (Trigger Handler Class)

- **Purpose:** Acts as a controller class for managing logic associated with vehicle order processing.
- **Functionality:**
  - Applies validation rules (e.g., check if vehicle is in stock).
  - Calculates order value or status based on vehicle and customer inputs.
  - Ensures clean and modular code by separating logic from the trigger body.

3. VehicleOrderBatch.apxc (Batch Apex Class)

- Purpose: Handles processing of large volumes of vehicle order records asynchronously.
- Use Case: Used to periodically update the Order_Status__c field for orders that meet specific criteria (e.g., stuck in "Pending" for over X days).
- Components:
  - Implements Database.Batchable<SObject>.
  - Processes records in chunks for efficiency.
- Outcome: Enhances performance and avoids governor limit issues when working with large datasets.

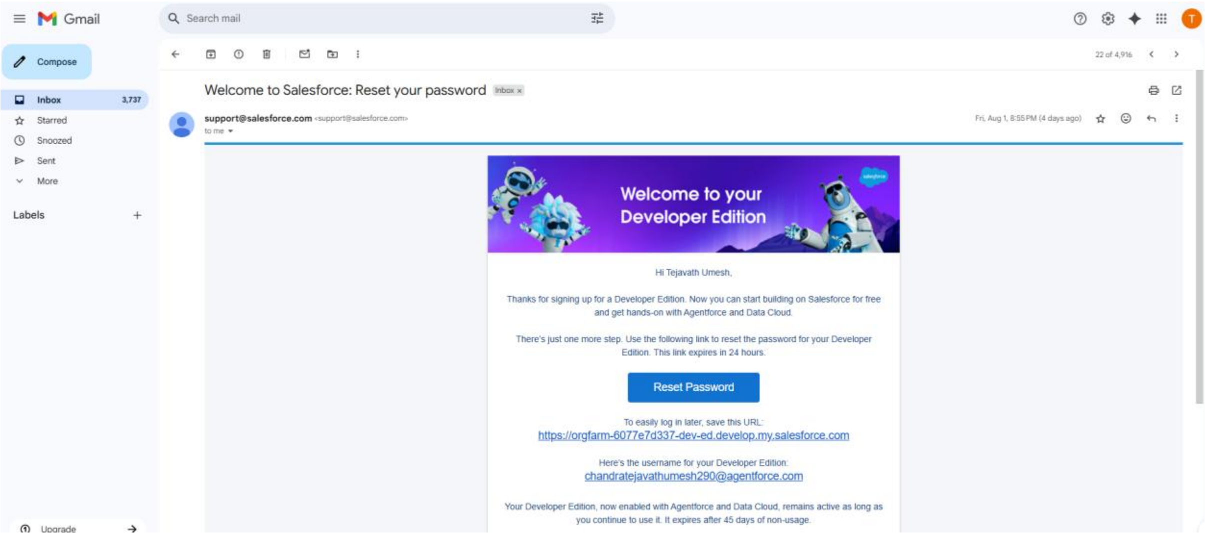
4. VehicleOrderBatchScheduler.apxc (Scheduler Class)

- Purpose: Automates the execution of the VehicleOrderBatch class at scheduled intervals.
- Functionality:
  - Implements Schedulable interface.
  - Runs the batch class (e.g., every night or weekly) to ensure order statuses are regularly evaluated and updated.
- Outcome: Ensures background operations are maintained without manual intervention.

DETAILED EXECUTION PROJECT PHASES

1.Developer OrgSetup : -

- A Salesforce Developer Org was created using https://developer salesforce.com/signup
- The account was verified, password set, and access was granted to the Salesforce Setup page.



Welcome to Salesforce: Reset your password

support@salesforce.com <support@salesforce.com>
to me

Fri, Aug 1, 8:55 PM (4 days ago)

Hi Tejavath Umesh,

Thanks for signing up for a Developer Edition. Now you can start building on Salesforce for free and get hands-on with Agentforce and Data Cloud.

There's just one more step. Use the following link to reset the password for your Developer Edition. This link expires in 24 hours.

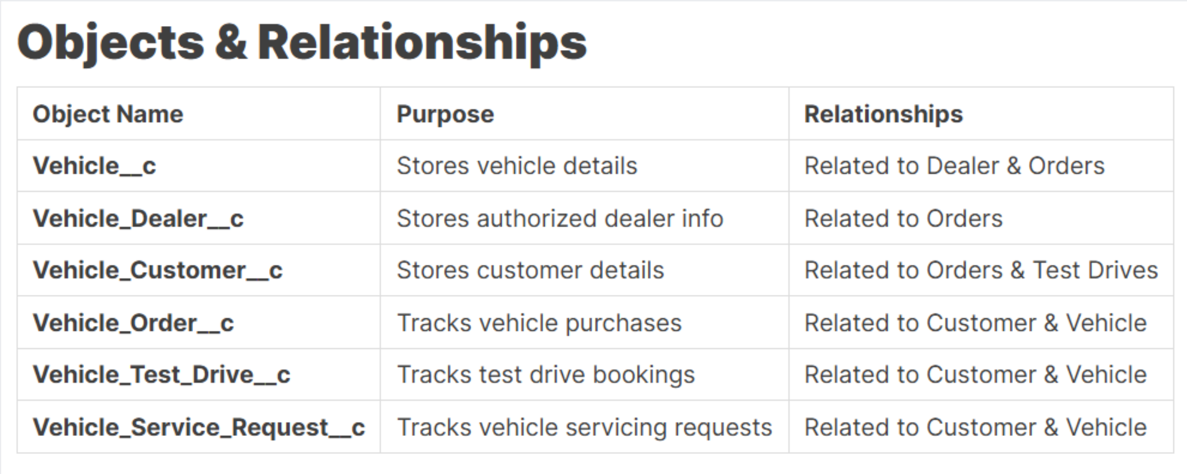
To easily log in later, save this URL:
https://orgfarm-6077e7d337-dev-ed.develop.my.salesforce.com

Here's the username for your Developer Edition:
chandratejavathumesh290@agentforce.com

Your Developer Edition, now enabled with Agentforce and Data Cloud, remains active as long as you continue to use it. It expires after 45 days of non-usage.

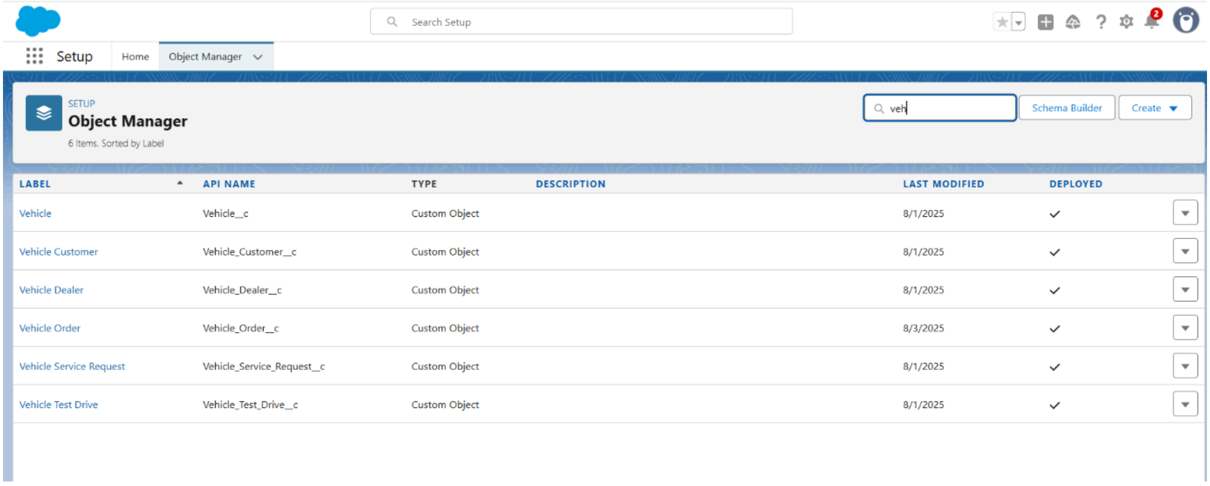
2. Creating Data Management-Objects :-

- Created custom objects like Vehicle, Customer, Order, Dealer, Test Drive, and Service Request to store and manage specific business data.
- Defined fields and relationships between objects to connect data and ensure smooth flow of information across modules.



Objects & Relationships

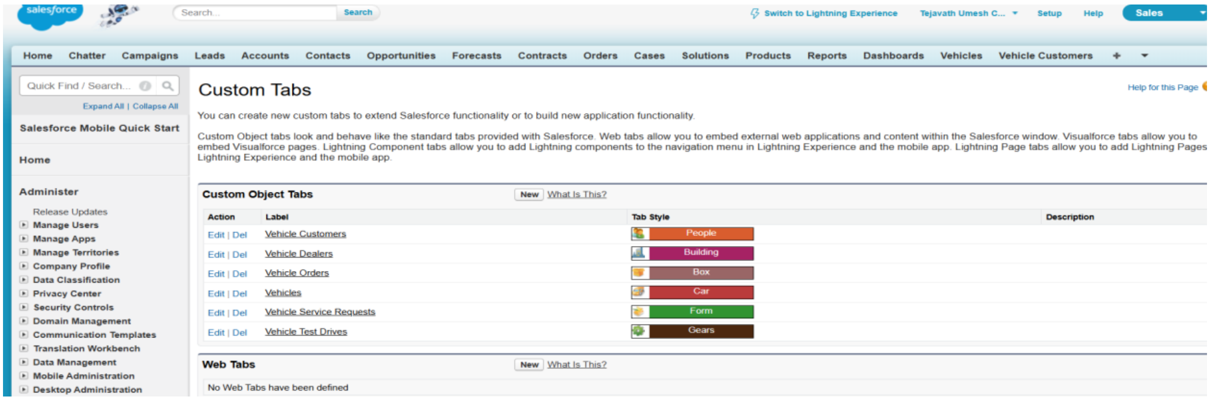
Object Name	Purpose	Relationships
Vehicle__c	Stores vehicle details	Related to Dealer & Orders
Vehicle_Dealer__c	Stores authorized dealer info	Related to Orders
Vehicle_Customer__c	Stores customer details	Related to Orders & Test Drives
Vehicle_Order__c	Tracks vehicle purchases	Related to Customer & Vehicle
Vehicle_Test_Drive__c	Tracks test drive bookings	Related to Customer & Vehicle
Vehicle_Service_Request__c	Tracks vehicle servicing requests	Related to Customer & Vehicle



LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Vehicle	Vehicle__c	Custom Object		8/1/2025	✓
Vehicle Customer	Vehicle_Customer__c	Custom Object		8/1/2025	✓
Vehicle Dealer	Vehicle_Dealer__c	Custom Object		8/1/2025	✓
Vehicle Order	Vehicle_Order__c	Custom Object		8/3/2025	✓
Vehicle Service Request	Vehicle_Service_Request__c	Custom Object		8/1/2025	✓
Vehicle Test Drive	Vehicle_Test_Drive__c	Custom Object		8/1/2025	✓

3.Creating Tabs :-

- Created custom tabs for each major object (like Vehicle, Dealer, Customer, Orders,test Drive, Service Request) to enable easy navigation and quick access to records within the Salesforce Lightning App.



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 Lightning Experience and the mobile app.

Custom Object Tabs New What Is This?

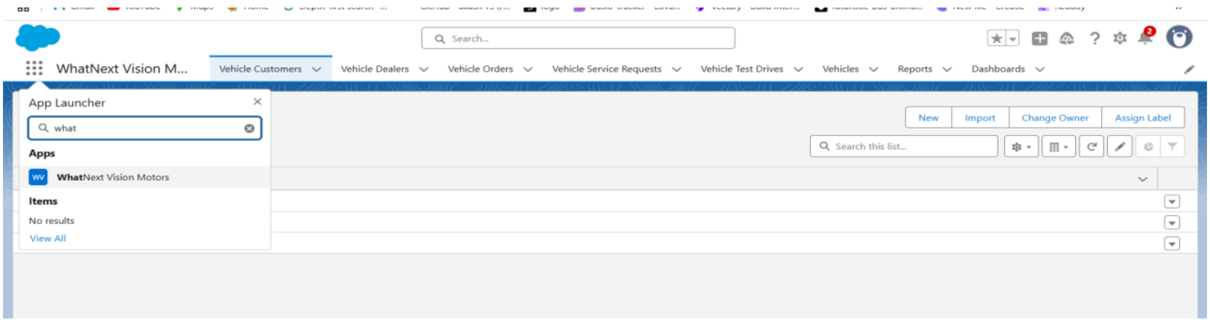
Action	Label	Tab Style	Description
Edit | Del	Vehicle Customers	People	
Edit | Del	Vehicle Dealers	Building	
Edit | Del	Vehicle Orders	Box	
Edit | Del	Vehicles	Car	
Edit | Del	Vehicle Service Requests	Form	
Edit | Del	Vehicle Test Drives	Gears	

Web Tabs New What Is This?

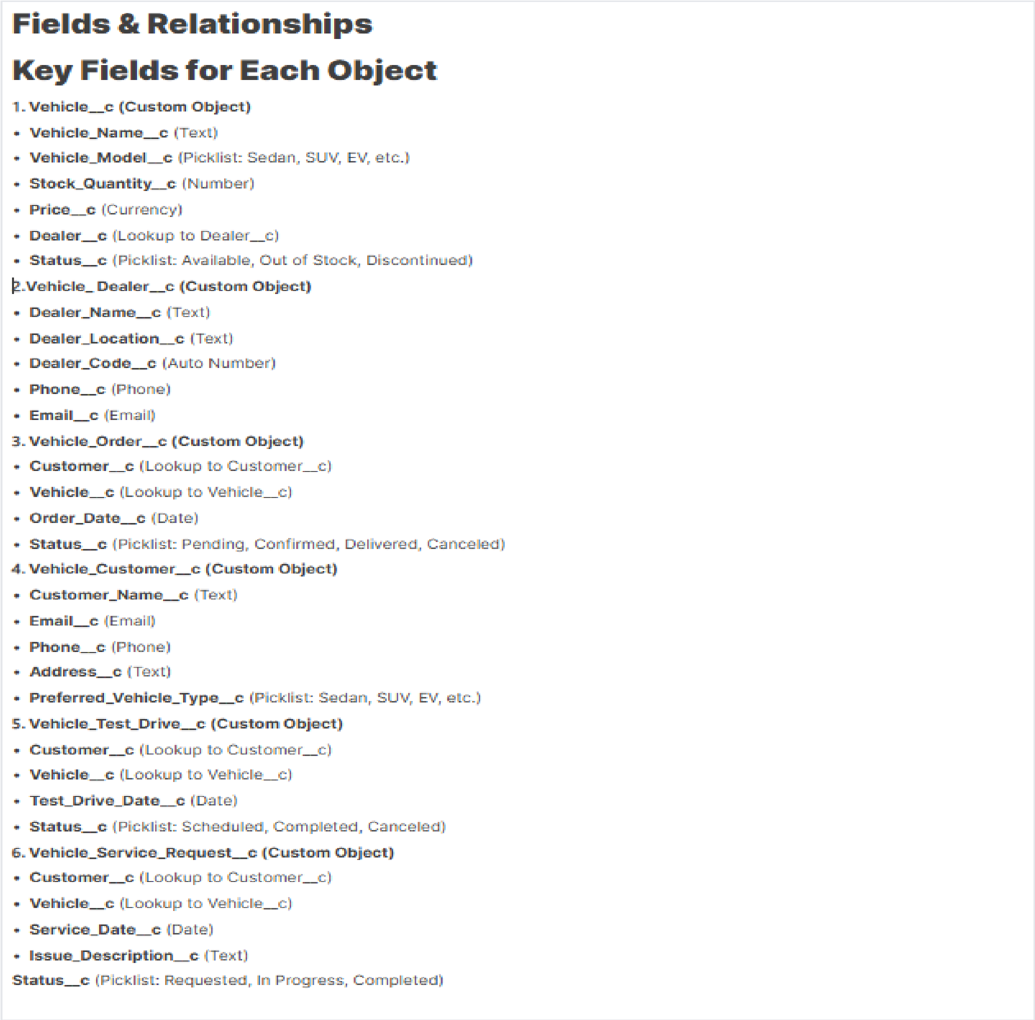
No Web Tabs have been defined

4.Creating the Lighting App :-

- We created a custom Lightning App called "WhatNext Vision CRM" to organize all important tabs like Vehicles, Customers, Orders, Test Drives, Dealers, and Service Requests in one place.
- This app helps users easily access and manage all records with a clean interface, customized navigation bar, and role-based access.



5. Creating Data Management Fields & Relationship for Each Object :-



Fields & Relationships

Key Fields for Each Object

1. Vehicle__c (Custom Object)

- **Vehicle_Name__c** (Text)
- **Vehicle_Model__c** (Picklist: Sedan, SUV, EV, etc.)
- **Stock_Quantity__c** (Number)
- **Price__c** (Currency)
- **Dealer__c** (Lookup to Dealer__c)
- **Status__c** (Picklist: Available, Out of Stock, Discontinued)

2.Vehicle_ Dealer__c (Custom Object)

- **Dealer_Name__c** (Text)
- **Dealer_Location__c** (Text)
- **Dealer_Code__c** (Auto Number)
- **Phone__c** (Phone)
- **Email__c** (Email)

3. Vehicle_Order__c (Custom Object)

- **Customer__c** (Lookup to Customer__c)
- **Vehicle__c** (Lookup to Vehicle__c)
- **Order_Date__c** (Date)
- **Status__c** (Picklist: Pending, Confirmed, Delivered, Canceled)

4. Vehicle_Customer__c (Custom Object)

- **Customer_Name__c** (Text)
- **Email__c** (Email)
- **Phone__c** (Phone)
- **Address__c** (Text)
- **Preferred_Vehicle_Type__c** (Picklist: Sedan, SUV, EV, etc.)

5. Vehicle_Test_Drive__c (Custom Object)

- **Customer__c** (Lookup to Customer__c)
- **Vehicle__c** (Lookup to Vehicle__c)
- **Test_Drive_Date__c** (Date)
- **Status__c** (Picklist: Scheduled, Completed, Canceled)

6. Vehicle_Service_Request__c (Custom Object)

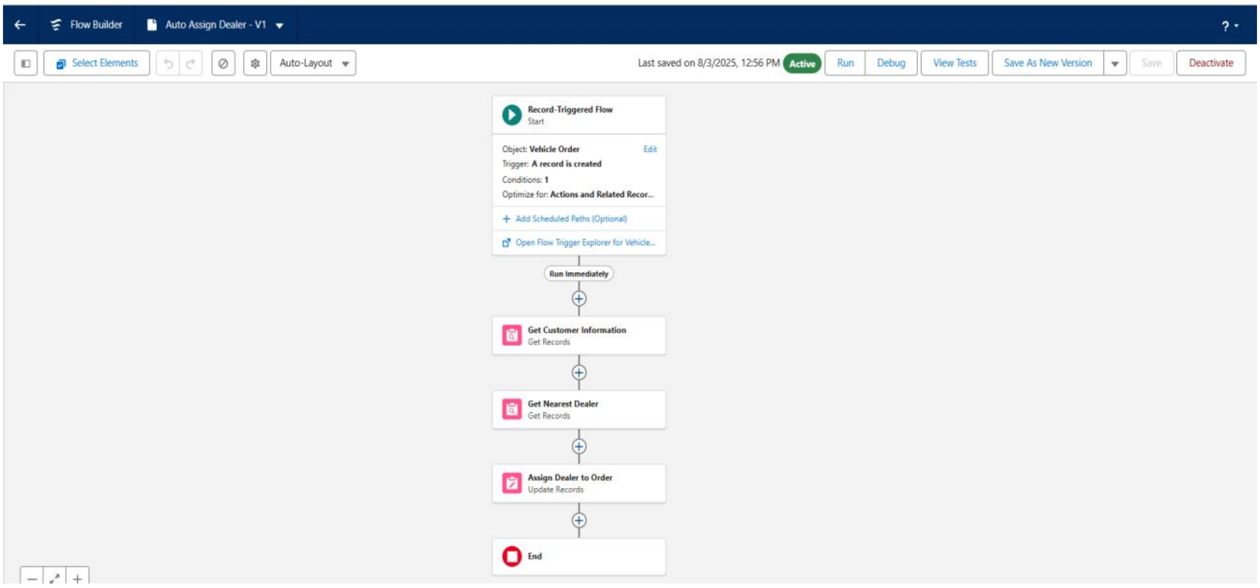
- **Customer__c** (Lookup to Customer__c)
- **Vehicle__c** (Lookup to Vehicle__c)
- **Service_Date__c** (Date)
- **Issue_Description__c** (Text)

Status__c (Picklist: Requested, In Progress, Completed)

6. Creating Automation :-

We created **2 record-triggered flows** in this project:

1. **Assign Nearest Dealer Flow** – automatically assigns the closest dealer based on the customer's location.
2. **Test Drive Reminder Flow** – sends an email reminder to the customer 24 hours before their scheduled test drive.



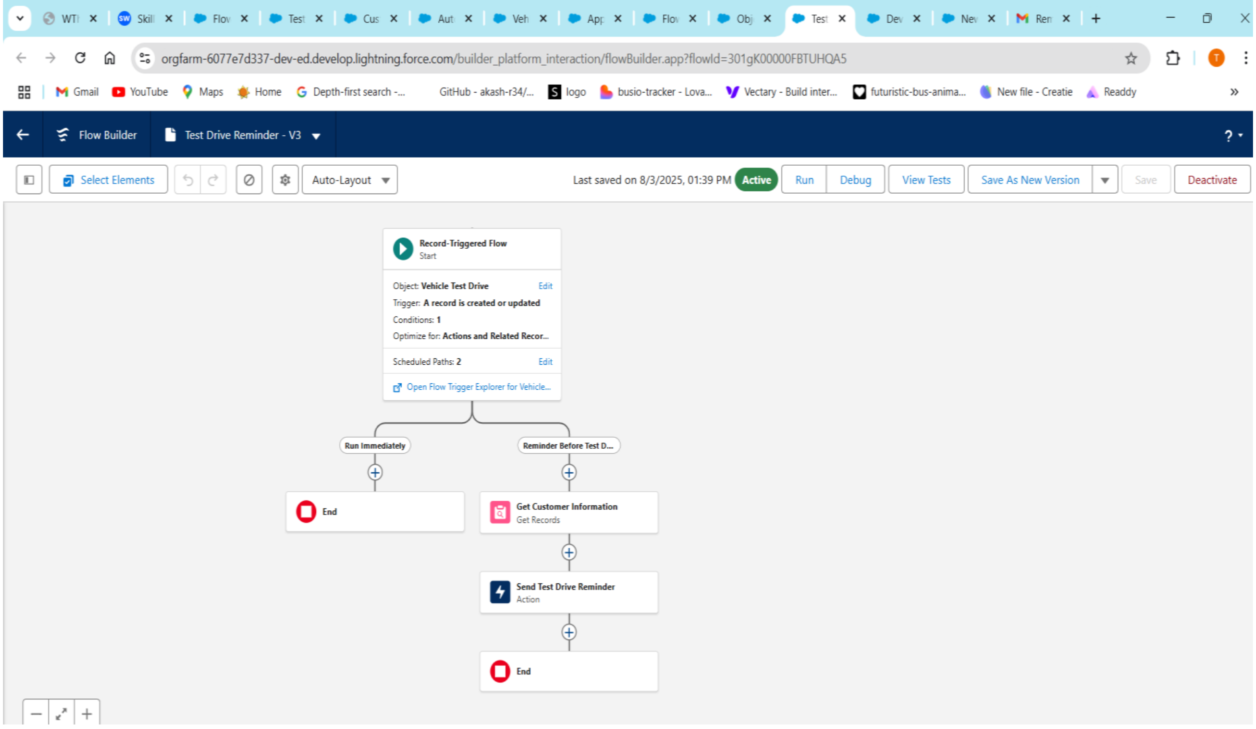
Object: Vehicle Order

Trigger: A record is created

Conditions: 1

Optimize for: Actions and Related Recor...

Fig 1: Assign Nearest Dealer Flow



Object: Vehicle Test Drive

Trigger: A record is created or updated

Conditions: 1

Optimize for: Actions and Related Recor...

Scheduled Paths: 2

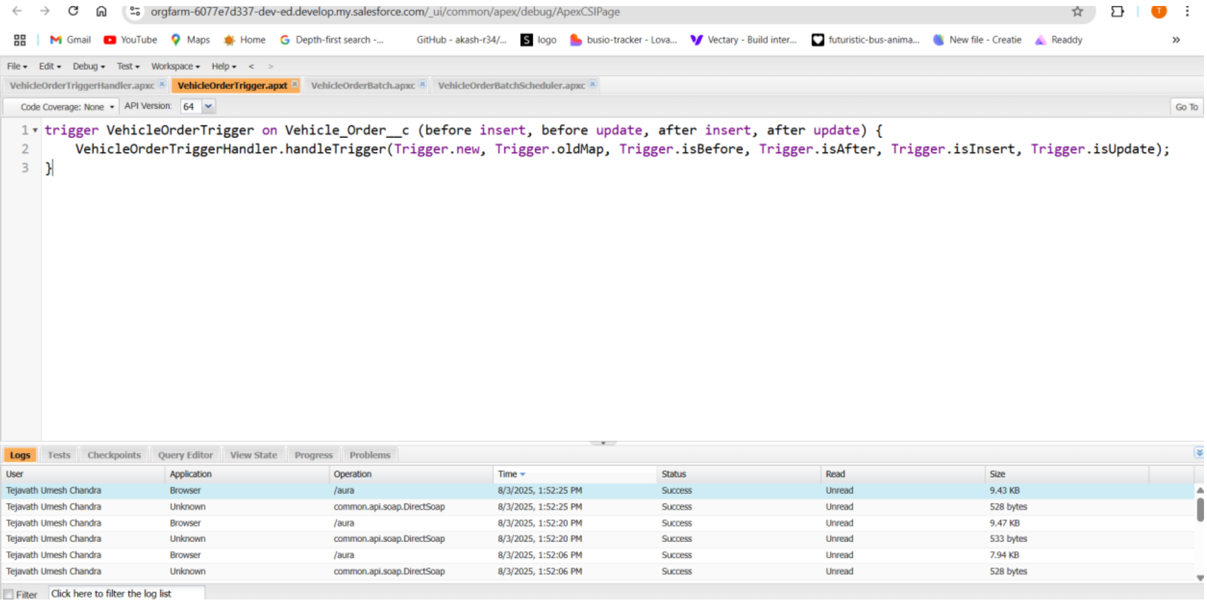
Fig 2: Test Drive Reminder Flow

7.Creating Apex and Trigger Batch Jobs :-

We created Apex Classes, Triggers, and Batch Jobs to handle advanced logic and automate large-scale operations like vehicle order processing.

The components created are:

1. **VehicleOrderTrigger.apxt** – Trigger to handle actions on vehicle orders.
2. **VehicleOrderTriggerHandler.apxc** – Apex class containing the logic for the trigger.
3. **VehicleOrderBatch.apxc** – Batch Apex class to update order statuses in bulk.
4. **VehicleOrderBatchScheduler.apxc** – Scheduler class to run the batch job at regular intervals.



```
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);
}
```

User	Application	Operation	Time	Status	Read	Size
Tejavath Umesh Chandra	Browser	/aura	8/3/2025, 1:52:25 PM	Success	Unread	9.43 KB
Tejavath Umesh Chandra	Unknown	common.api.soap.DirectSoap	8/3/2025, 1:52:25 PM	Success	Unread	528 bytes
Tejavath Umesh Chandra	Browser	/aura	8/3/2025, 1:52:20 PM	Success	Unread	9.47 KB
Tejavath Umesh Chandra	Unknown	common.api.soap.DirectSoap	8/3/2025, 1:52:20 PM	Success	Unread	533 bytes
Tejavath Umesh Chandra	Browser	/aura	8/3/2025, 1:52:06 PM	Success	Unread	7.94 KB
Tejavath Umesh Chandra	Unknown	common.api.soap.DirectSoap	8/3/2025, 1:52:06 PM	Success	Unread	528 bytes

Fig 3: Apex and Trigger Batch Jobs

PROJECT EXPLANATION WITH REAL-WORLD EXAMPLE

Let’s imagine a real customer scenario and see how Salesforce makes the whole vehicle-buying journey smooth, automated, and intelligent through the WhatNext Vision Motors solution.

1. CUSTOMER INQUIRY – MEET RAHUL :-

Rahul Verma, a working professional from Hyderabad, is looking to buy a premium EV car. He lands on the WhatNext Vision Motors website and fills out a test drive request form.

- **In Salesforce:**
  A new record gets created in the custom object **Vehicle_Customer__c** capturing his name, email, city, preferred vehicle, and contact details.
- **Validation Rule:**
  Before storing the data, a rule checks if his email format is valid (like rahul@gmail.com). If not, the record won't be saved ensuring clean data.

2. AUTOMATED DEALER ASSIGNMENT – NO HUMAN DELAY :-

Now, how do we assign Rahul to the nearest dealer?

- **Behind the scenes in Salesforce:**
  A **Record-Triggered Flow** automatically compares Rahul's city to the **Vehicle_Dealer__c** records and assigns the closest dealer.
- This is instant no manual lookup. The moment Rahul's record is saved, dealer assignment is done.

3. TEST DRIVE CONFIRMATION -TIMELY COMMUNICATION :-

- The assigned dealer contacts Rahul and schedules a test drive. A new record is created in **Vehicle_Test_Drive__c**, capturing test drive details.
- **Flow Magic Again :**
  Another **Record-Triggered Flow** sends Rahul a **personalized email reminder** about his test drive date and time creating a premium customer experience.

4. VEHICLE ORDER – THE BIG DECISION :-

After a great test drive experience, Rahul decides to purchase the car.

- A new **Vehicle_Order__c** record is created, linking the customer, selected vehicle, price, and dealer.
- **Apex Trigger in action:**
  The trigger automatically calculates the **Total_Amount__c** (based on the vehicle model and any add-ons). For example, ₹14,50,000 is computed and stored instantly.

5. BULK ORDER PROCESSING – BACKEND EFFICIENCY :-

At the end of the day, there are hundreds of such orders to process.

- **We built a Batch Apex class**:
- **VehicleOrderBatch.apxc** processes these orders in bulk.
- **VehicleOrderBatchScheduler.apxc** runs on schedule to ensure large data volumes don't slow down the system.

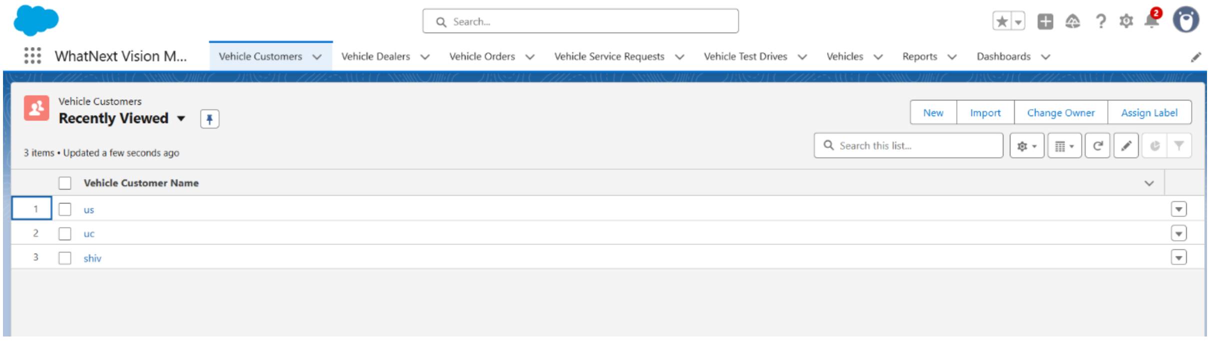
Inventory gets adjusted accordingly ensuring real-time stock visibility.

6. POST-SALE SERVICE – CUSTOMER COMES BACK :-

Two months later, Rahul faces a minor issue and books a service.

- A record in **Vehicle_Service_Request__c** is created with his service type, issue, and preferred date.
- Dealers can now access his full history — order, test drive, and service — in one place.

Screenshots

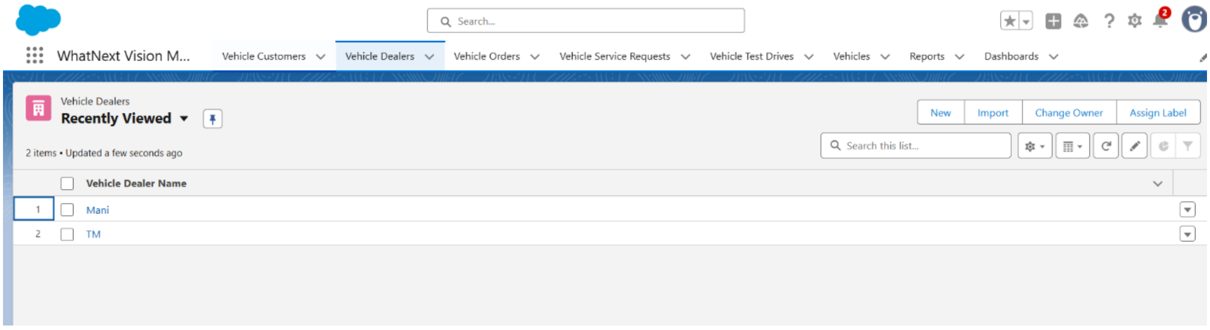


Vehicle Customers
Recently Viewed

3 items • Updated a few seconds ago

	Vehicle Customer Name
1	us
2	uc
3	shiv

Fig 1: Vechicle Customer page

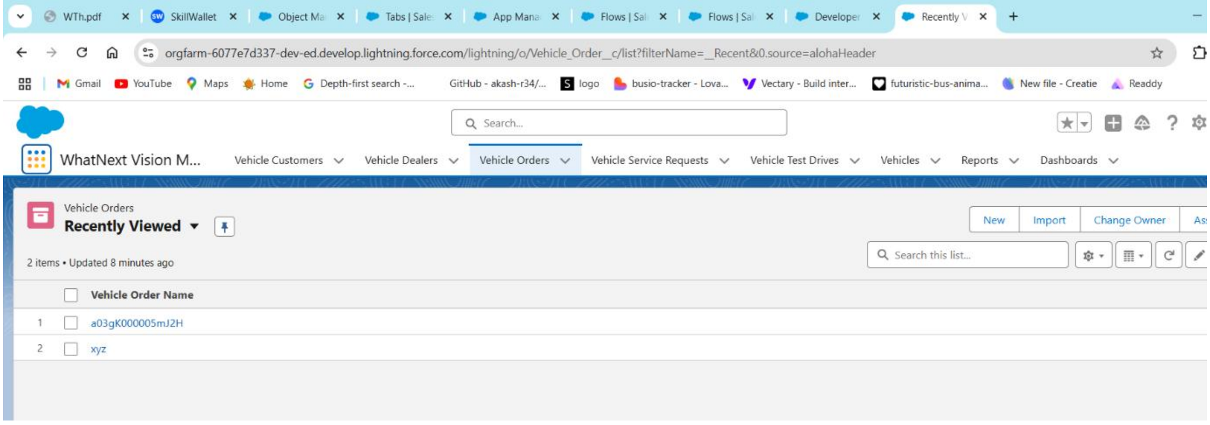


Vehicle Dealers
Recently Viewed

2 items • Updated a few seconds ago

	Vehicle Dealer Name
1	Mani
2	TM

Fig 2: Vechicle Dealer page

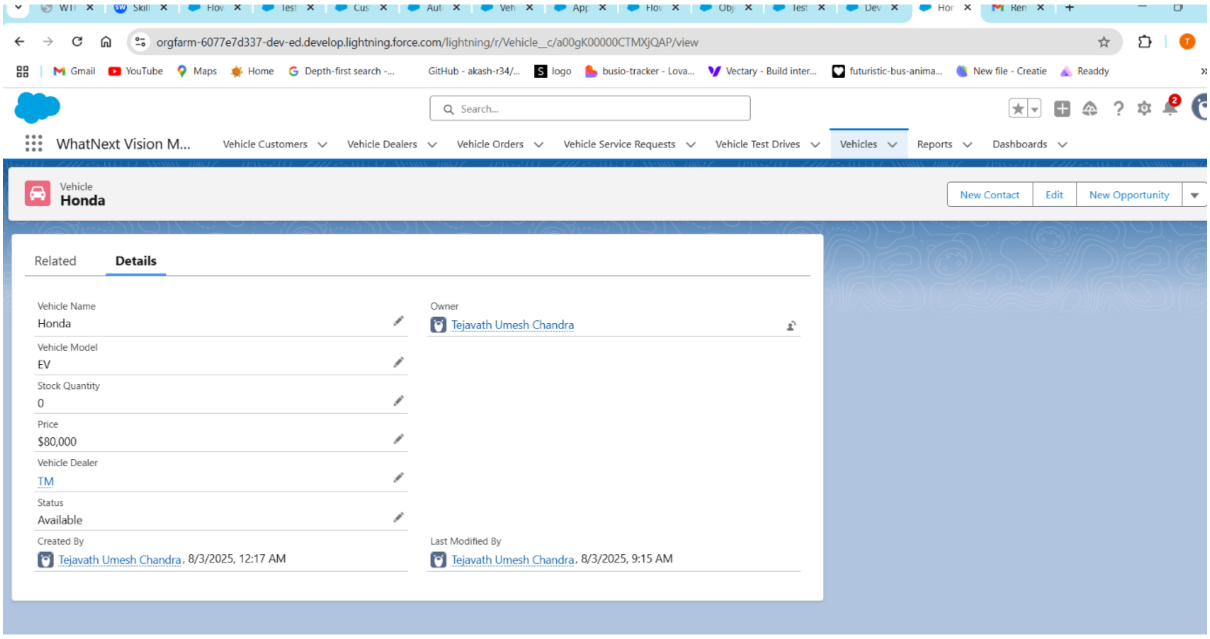


Vehicle Orders
Recently Viewed

2 items • Updated 8 minutes ago

	Vehicle Order Name
1	a03gK000005mJ2H
2	xyz

Fig 3: Vechicle Order page



Vehicle
Honda

Vehicle Name: Honda

Vehicle Model: EV

Stock Quantity: 0

Price: $80,000

Vehicle Dealer: TM

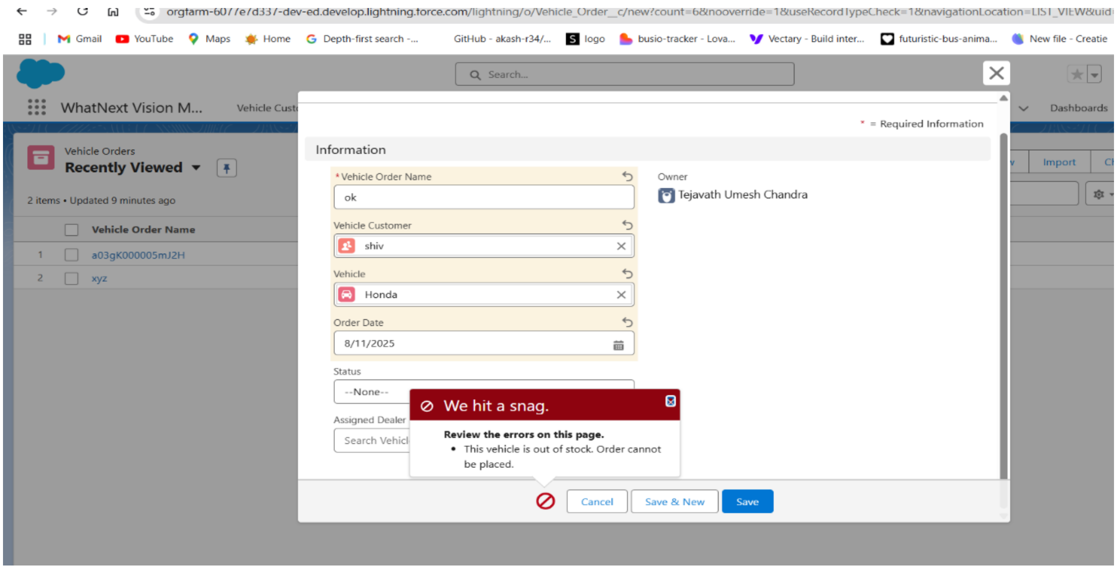
Status: Available

Created By: Tejavath Umesh Chandra, 8/3/2025, 12:17 AM

Owner: Tejavath Umesh Chandra

Last Modified By: Tejavath Umesh Chandra, 8/3/2025, 9:15 AM

Fig 4: Vechicles page


orgfarm-6077e7d337-dev-ed.develop.lightning.force.com/lightning/o/Vehicle_Order__c/new?count=6&nooverride=1&useRecordTypeCheck=1&navigationLocation=LIST_VIEW&uid
WhatNext Vision M...
Vehicle Orders
Recently Viewed
2 items • Updated 9 minutes ago
Vehicle Order Name
a03gK000005mJ2H
xyz
Import
Dashboards
Cancel
Save & New
Save

* = Required Information

Information

*Vehicle Order Name: ok

Vehicle Customer: shiv

Vehicle: Honda

Order Date: 8/11/2025

Status: --None--

Assigned Dealer: Search Vehicl

Owner: Tejavath Umesh Chandra

We hit a snag.

Review the errors on this page.

- This vehicle is out of stock. Order cannot be placed.

Fig 5: Vehicle Ordering Page if the selected vehicle is currently out of stock, the customer will be restricted from placing an order for that product until it becomes available again in the inventory.

Key Business Outcomes

- Centralized management of customer, dealer, vehicle, and service data in Salesforce.
- Real-time automation of key processes like **dealer assignment** and **test drive reminders** using Flows.
- Enhanced data visibility and transparency across departments via dashboards and role-based access.
- Improved operational efficiency by reducing manual work and human errors.
- Streamlined order-to-delivery process for vehicles and post-sales service requests.

Business Value / Impact

- Reduced manual interventions by automating repetitive business processes.
- Enhanced customer engagement with timely email reminders and accurate dealer mapping.
- Increased data security and access control using custom Profiles and Roles.
- Boosted decision-making through real-time reporting and dashboards.
- Improved team collaboration by offering a unified system for all departments.

Future Scope

- Integration with online **payment gateways** for vehicle booking and service charges.
- Development of a **customer mobile app** for tracking orders, services, and updates.
- Implementation of **Einstein AI or GPT-based chatbots** for customer support and lead prediction.
- Expansion to include **pickup & drop scheduling**, **maintenance alerts**, and **inventory forecasting**.
- Use of **Salesforce Experience Cloud** to give customers a self-service portal experience.

Conclusion

The WhatNext Vision Motors project serves as an example of how Salesforce can be used as a single platform to automate, optimize, and scale important business processes in the automotive industry. The entire vehicle journey from customer inquiry to test drive, order placement, and after-sales service is managed effectively and smoothly through the use of custom objects, flows, Apex triggers, batch classes, and role-based access.

This solution has greatly decreased manual labor, increased customer satisfaction, and allowed for more intelligent business decision-making through real-time data visibility, automated communications, and intelligent process execution. In addition to meeting present operational requirements, it was designed to be scalable to support upcoming additions like customer portals, mobile access, and AI integrations.

Overall, this Salesforce deployment enables WhatNext Vision Motors to provide innovative customer experiences while upholding operational excellence at every organizational level.