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

ABSTRACT

WhatNext Vision Motors, a rising company in the automotive industry, sought to modernize its customer interactions and operational processes by implementing a custom **Salesforce CRM** solution. The project's main goal was to **streamline vehicle order management**, ensure accurate **dealer assignment**, and improve ** customer engagement** through automation.

The company's prior **manual processes** frequently caused delays, issues with stock management, and customer dissatisfaction. To resolve these problems, the new CRM system was designed with features like **real-time stock validation**, automatic dealer assignment based on a customer's location, email reminders for test drives, and backend automation using **Apex triggers and batch classes**. The platform also offers a user-friendly experience for internal staff via **Lightning Apps** and **Dynamic Forms**. Ultimately, this solution boosts efficiency, minimizes errors, and establishes a scalable foundation for future improvements, such as AI-based vehicle recommendations or chatbot support.

Objective

The core objectives for the **WhatsNext Vision Motors Salesforce** project were**:**

- **Improve Customer Satisfaction:** Create mechanisms to automatically suggest the nearest authorized dealer to a customer based on their location for a smooth ordering experience.
- **Ensure Order Accuracy and Fulfillment**: Implement real-time stock validation to prevent customers from placing orders for vehicles that are currently out of stock.
- **Streamline Operations with Automation:** Develop advanced automation using Apex and Flows to efficiently manage customer orders, test drives, and internal data processes.
- **Maintain Data Integrity and Scalability:** Utilize Batch Apex to periodically and reliably update vehicle stock data, ensuring the system’s information remains current and is scalable for large volumes of data.
- **Proactive Customer Communication:** Automate email reminders for scheduled test drives to reduce no-shows and enhance customer engagement.

Technology Description

The project is built entirely on the **Salesforce platform**, leveraging its CRM and customization features:

- **Salesforce CRM:** Serves as the central platform for storing and managing vehicle details, stock availability, dealer information, customer details, and tracking customer orders, test drives, and service requests.
- **Apex and Apex Triggers:** Used to enforce complex business rules, such as **stock validation** upon order creation and **automatic dealer assignment** based on customer location. **Trigger handlers** are used for code modularity and maintainability.
- **Batch Apex and Scheduled Apex:** Employed to periodically process large data volumes, specifically to check vehicle stock levels and update availability in Salesforce, in addition to handling scheduled order processing and notifications.

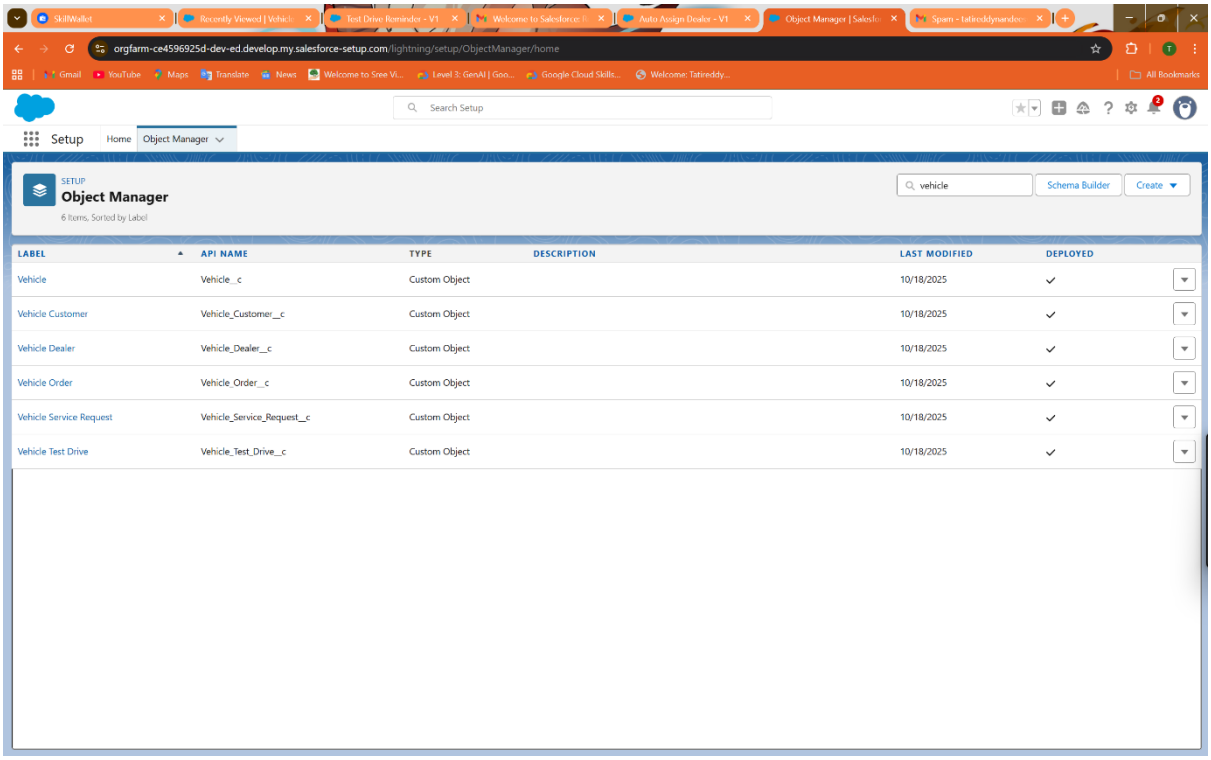
- **Data Modelling:** Custom objects and fields were created for efficient data structuring.
- **Process Automation (Flows): Record-Triggered Flows** and other automation tools are used for sending automated email reminders.
- **Lightning App Builder:** Used to create a dedicated **Lightning App** to provide a modern, consolidated user interface for internal users.

Detailed Execution of Project Phases

The project followed a structured approach, from foundational data architecture to advanced programmatic and declarative automation:

1. Data Management and Modeling

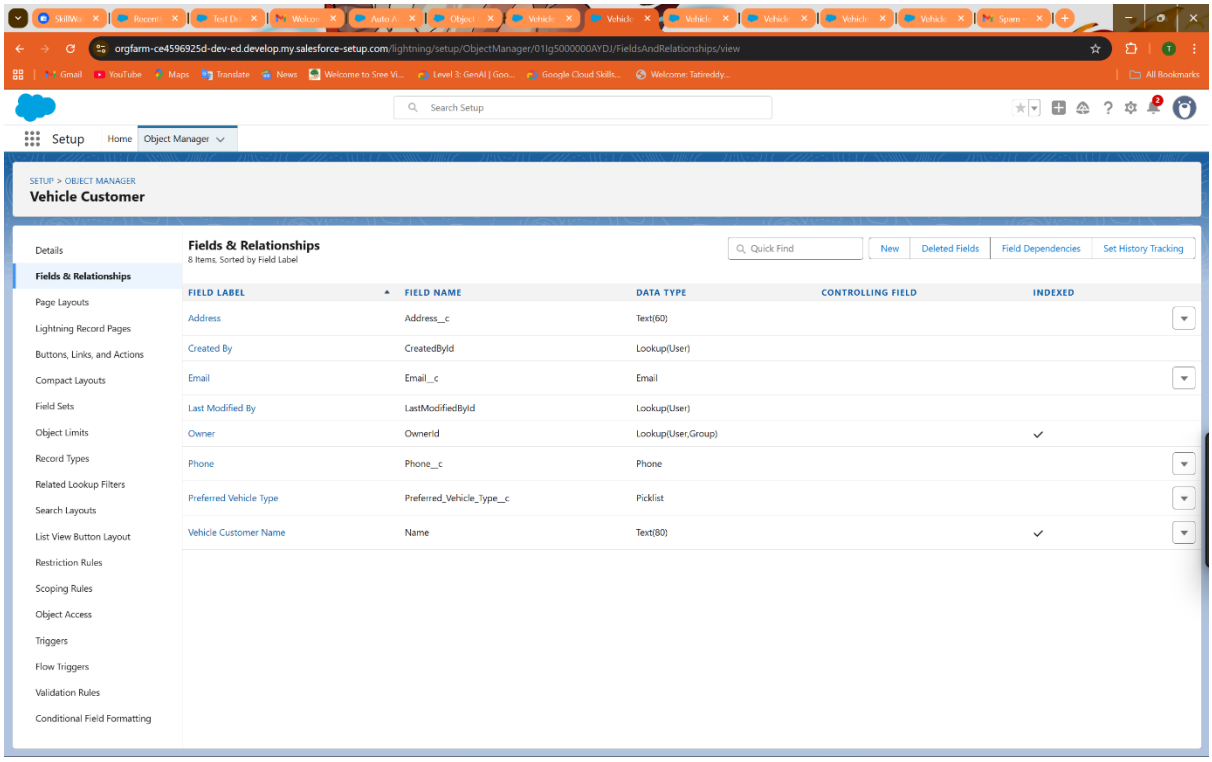
- Custom Objects: Six custom objects were created to manage the vehicle lifecycle: Vehicle__c, Vehicle_Dealer__c, Vehicle_Customer__c, Vehicle_Order__c, Vehicle_Test_Drive__c, and Vehicle_Service_Request__c.



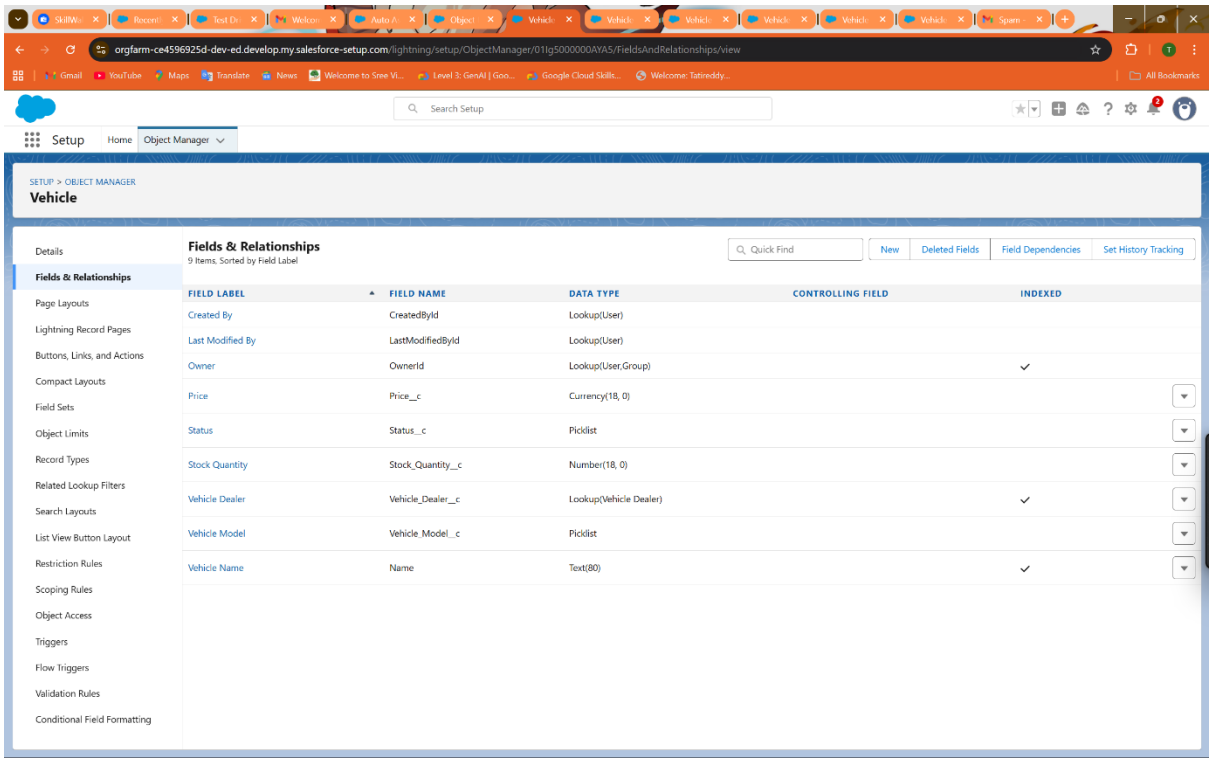
- Fields and Relationships: The foundation of an effective **Customer Relationship Management (CRM)** system lies in meticulously defined fields and strong record relationships. Essential fields such as **Stock_Quantity__c** for inventory tracking, **Dealer_Location__c** for geographical segmentation, and **Status__c** to manage workflow states (e.g., *New, In Progress, Closed*) are crucial for business operations and reporting.
- These core data points are then logically connected using **Lookup** or **Master-Detail** relationships. A **Lookup relationship**, such as linking an **Order** record to a **Customer** record, allows you to associate two objects while maintaining independent ownership and security rules. Conversely, a **Master-Detail relationship** between **Order** and **Vehicle** often indicates a tighter, required association where the detail record (Vehicle) is conceptually a child of the master record (Order), inheriting its security and deletion behavior. Establishing

these links ensures data integrity, enables powerful cross-object reporting, and provides users with a comprehensive **360-degree view** of the customer and their interactions.

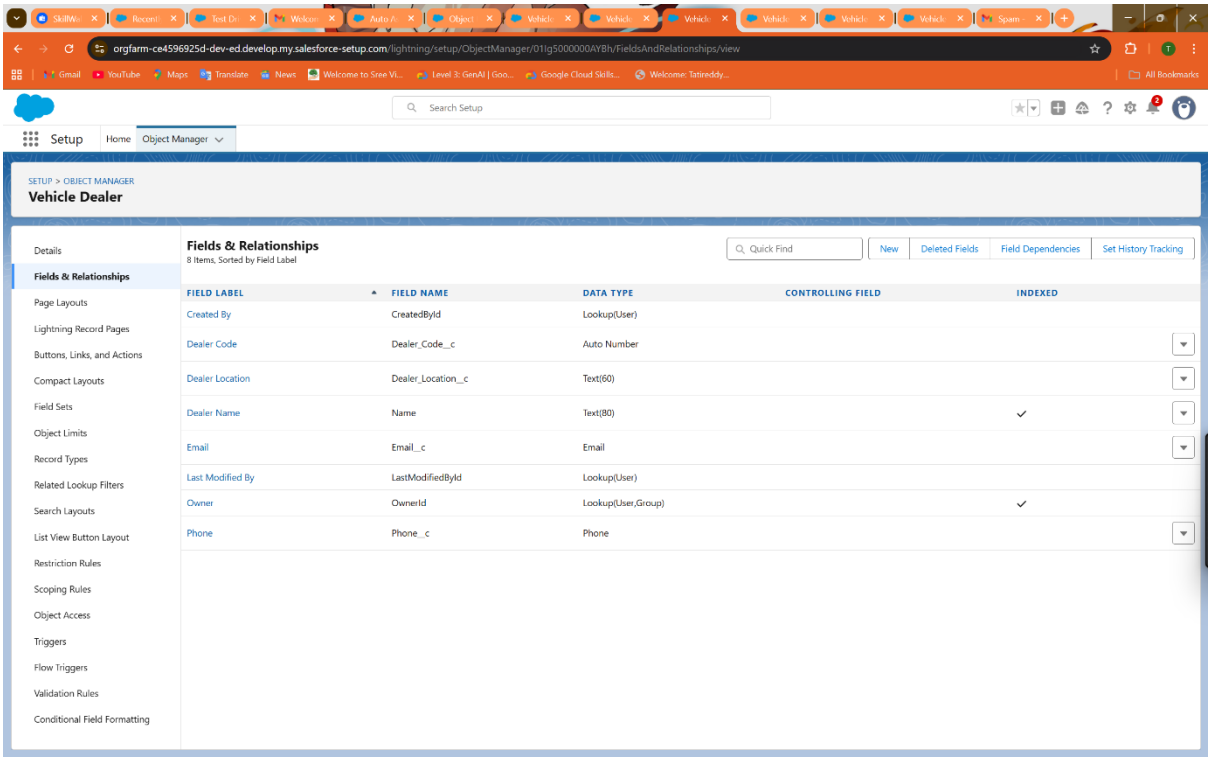
- VEHICLE CUSTOMER FIELDS AND RELATIONSHIPS



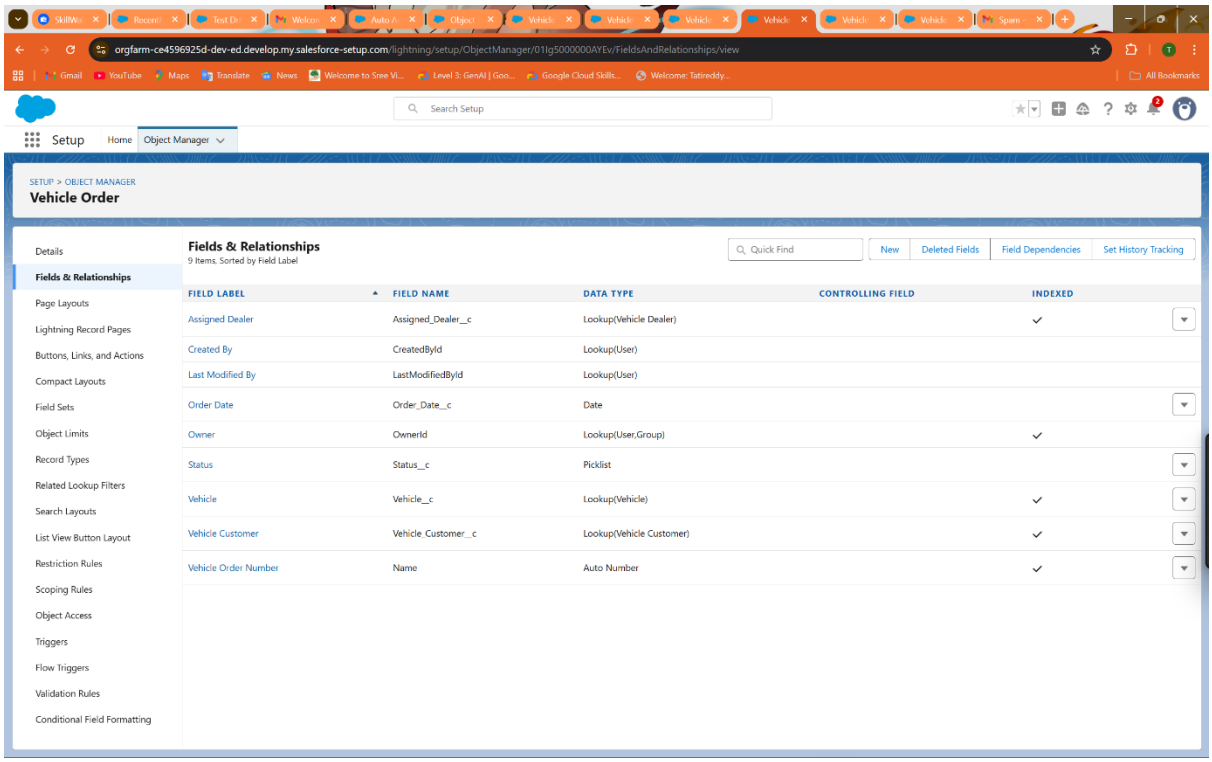
- VEHICLE FIELD AND RELATIONSHIPS



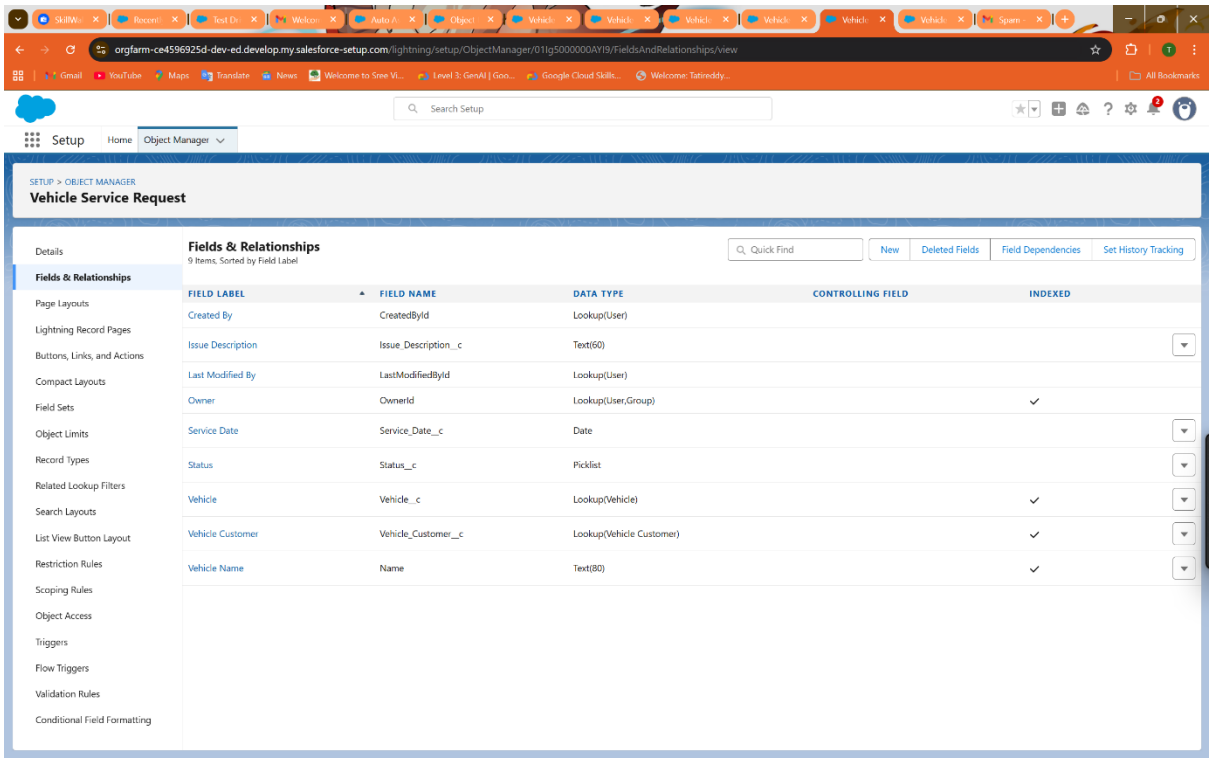
- VEHICLE DEALER FIELD AND RELATIONSHIPS



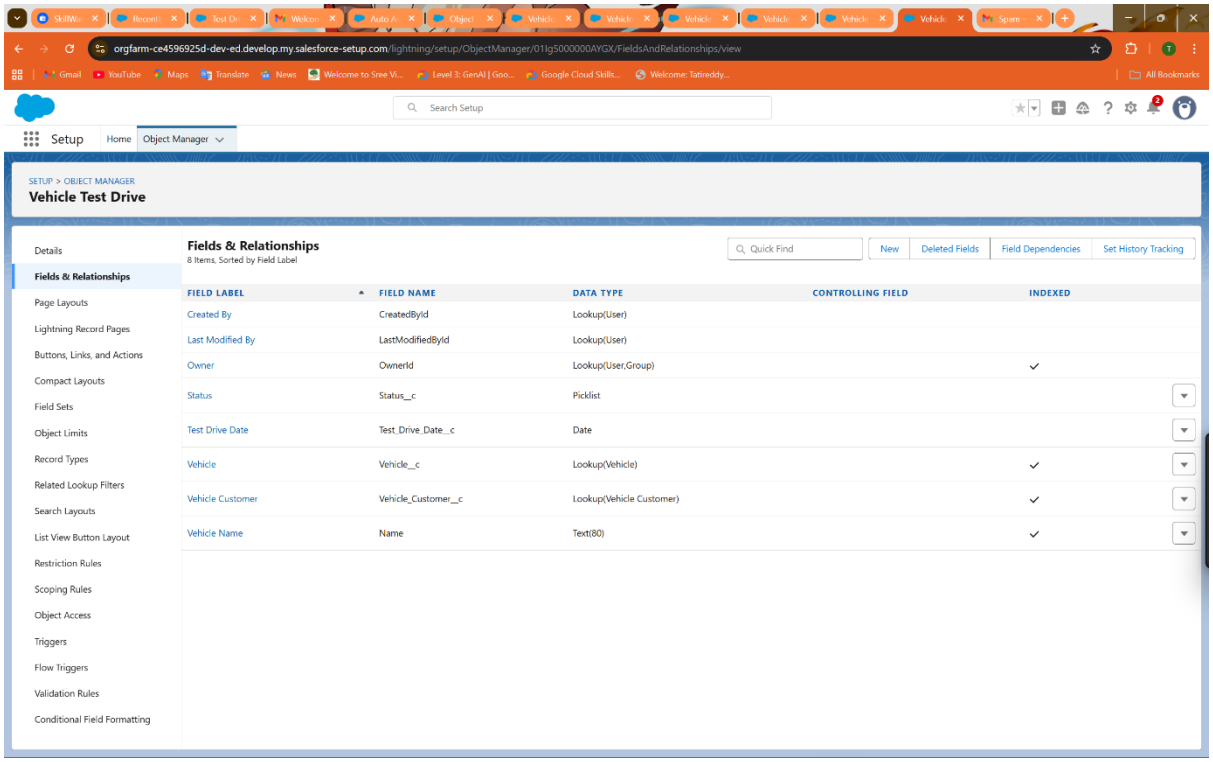
- VEHICLE ORDER FIELDS AND RELATIONSHIPS



- VEHICLE SERVICE REQUEST FIELDS AND RELATIONSHIPS

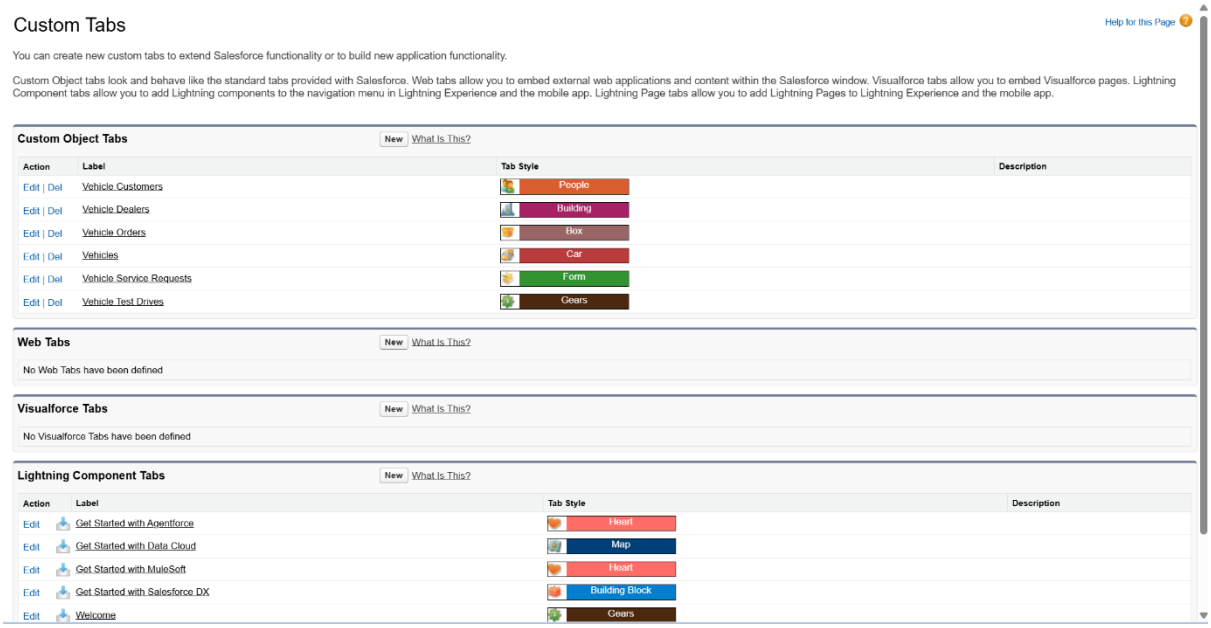


- VEHICLE TEST DRIVE FIELDS AND RELATIONSHIPS

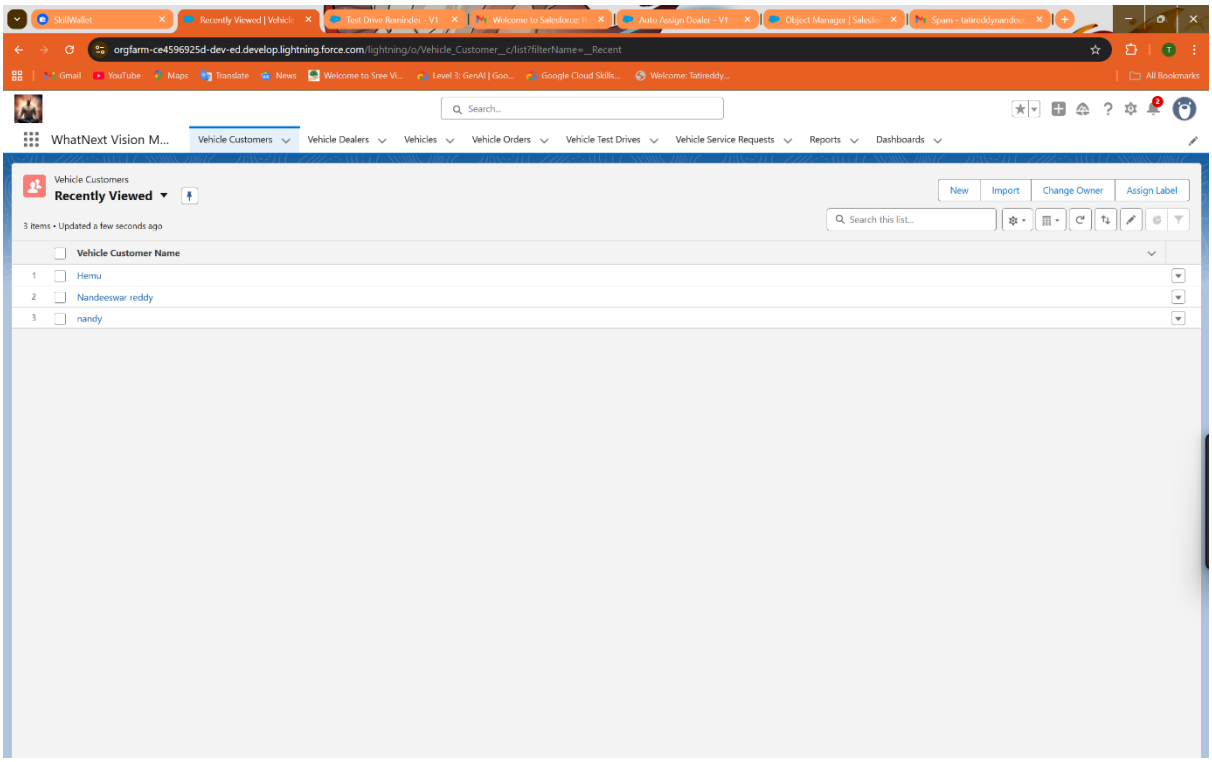


2. User Interface Setup

- **Custom Tab Creation:** A custom tab for the Vehicle__c object was created to allow user access to vehicle data.

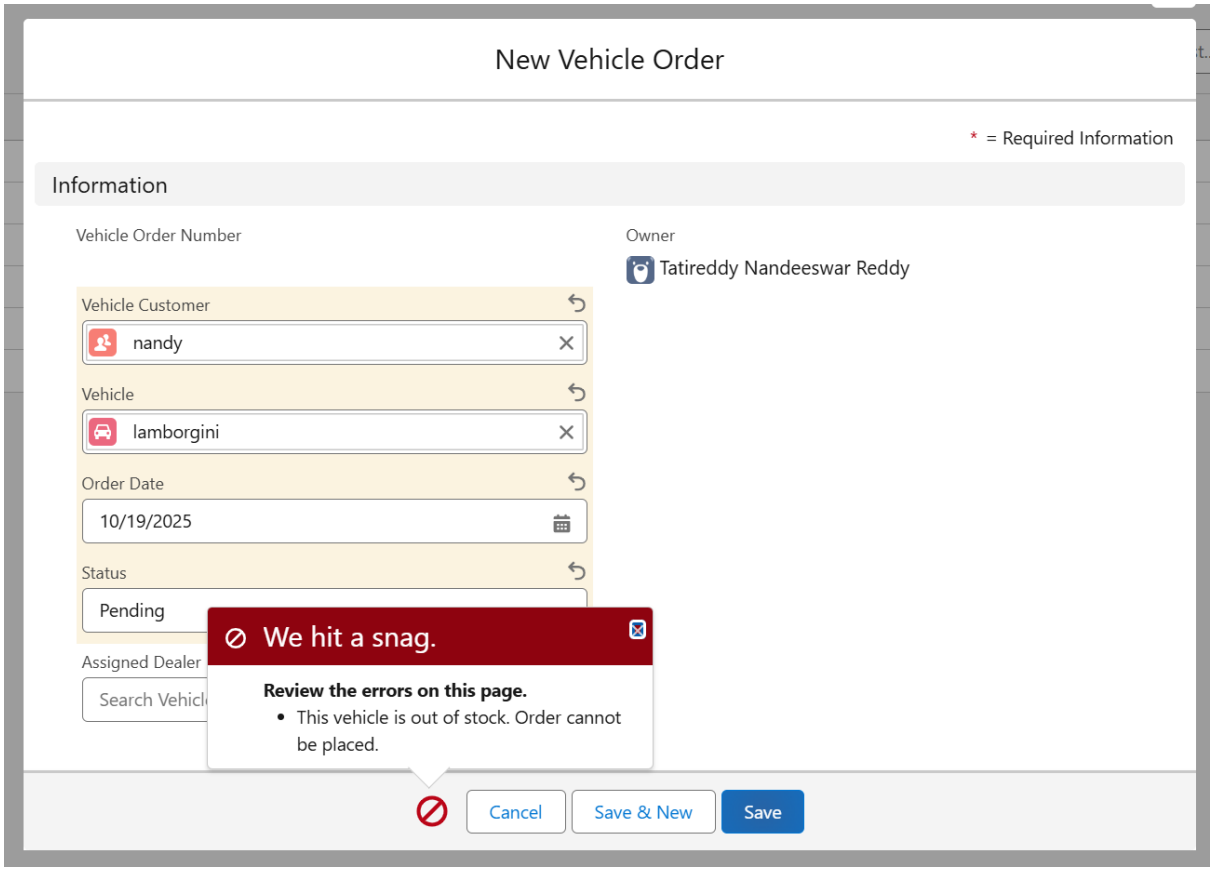


- **Lightning App Creation:** A dedicated WhatsNext Vision Motors Lightning App was configured, integrating all relevant custom object tabs into the navigation bar for centralized user access.



3. Apex and Trigger Framework

- A modular Apex architecture was implemented using a handler class (VehicleOrderTriggerHandler) invoked by a simple trigger (VehicleOrderTrigger) to manage all events on the Vehicle_Order__c object.
- **Stock Validation Logic:** The handler class includes logic to perform real-time stock validation *before* a Vehicle_Order__c record is inserted. This logic prevents the order from being created if the selected vehicle's Stock_Quantity__c is zero or less.



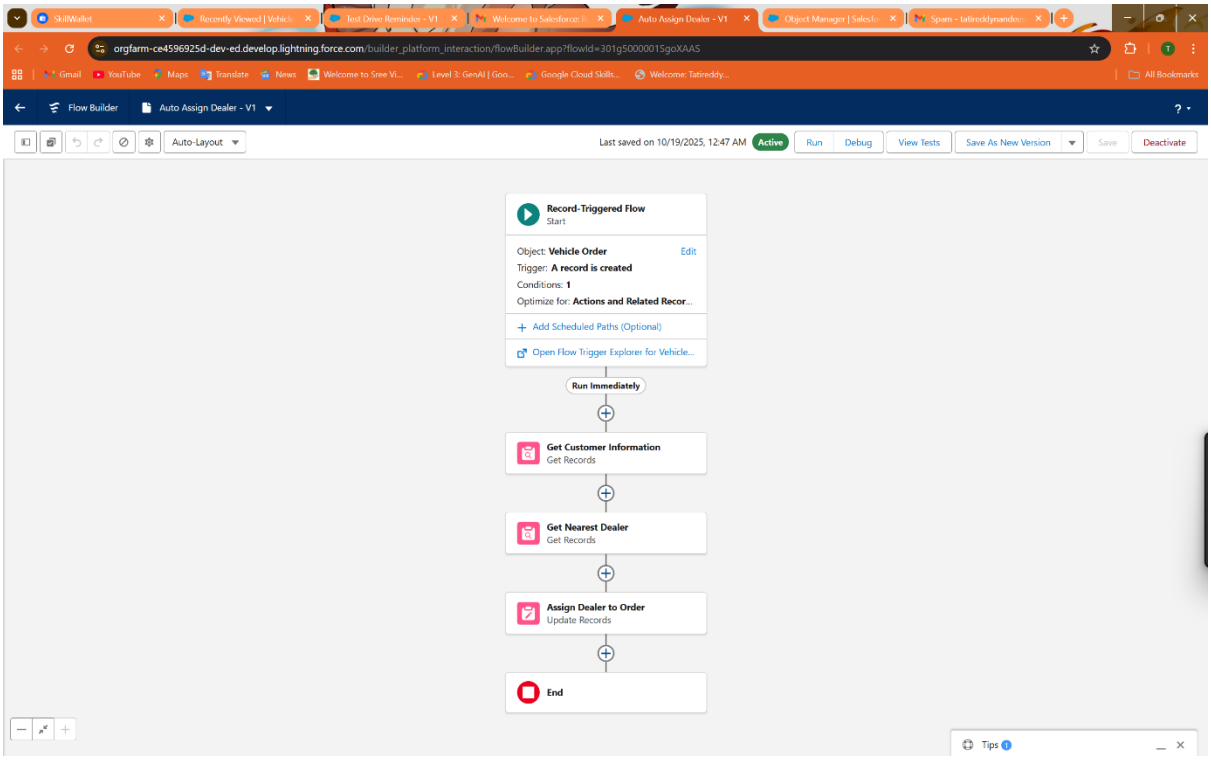
4. Process Automation (Flows)

Two key business processes were automated using Record-Triggered Flows:

A. Auto Assign Dealer Flow

- **Type:** A Record-Triggered Flow that fires when a Vehicle_Order__c record is Created and its Status__c is 'Pending'.
- **Steps:**

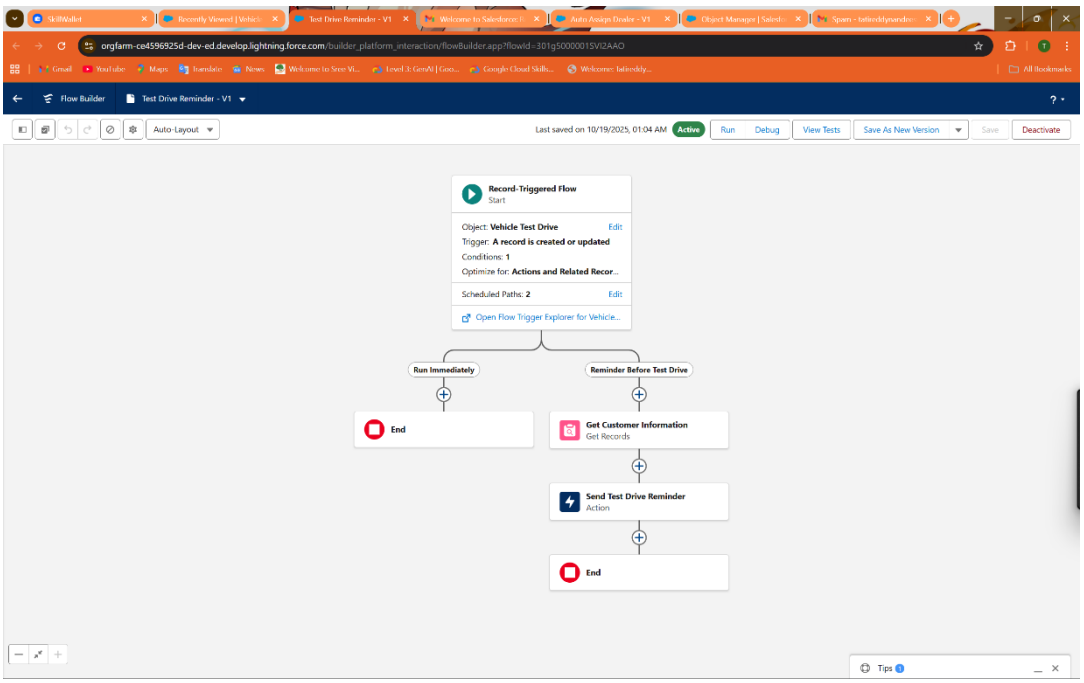
    1. Retrieves the related Vehicle_Customer__c record.
    2. Finds the nearest dealer by retrieving the Vehicle_Dealer__c record where the Dealer_Location__c Equals the customer's address.
    3. Updates the Order record to assign the ID of the retrieved dealer.



B. Test Drive Reminder Flow

- **Type**: A Record-Triggered Flow that fires when a Vehicle_Test_Drive__c record is Created or Updated**.**
- **Scheduling:** Uses a Scheduled Path to run 1 Day Before the Test_Drive_Date__c.
- **Steps:**

    1. Retrieves the related Vehicle_Customer__c record.
    2. Sends a reminder email to the customer's Email__c address with the subject "Reminder: Your Test Drive is Tomorrow!".



5. Batch and Scheduled Apex

A nightly job was implemented for consistent stock availability and order processing:

- **Batch Apex (VehicleOrderBatch):** Contains the logic to periodically check for vehicles whose stock has been replenished and update the status of any associated Vehicle_Order__c records from 'Pending' to 'Confirmed'**.**
- **Scheduled Apex (VehicleOrderBatchScheduler):** Implements the Schedulable interface to execute the Batch job.
- **Scheduling:** The job is scheduled to run daily at 12:00 PM.

Project Explanation with Real-World Example

Scenario: Processing an Order for an Out-of-Stock Vehicle

The system's integrity is demonstrated by how it handles a critical edge case:

1. **Order Placement:** A customer attempts to place a new Vehicle_Order__c for a **WhatsNext X1 SUV**.
2. **Stock Validation (Apex):** The VehicleOrderTrigger fires before insert. The code checks the Vehicle__c.Stock_Quantity__c for the X1 SUV, which is currently **0**. The Apex logic **prevents the insertion** of the Order record, notifying the user that the vehicle is out of stock. *Result: Customer disappointment is mitigated, and no unfulfillable order enters the system.*
3. **Stock Replenishment (Batch Apex):** The next day, a new shipment of X1 SUVs arrives, and the external inventory system updates. The **VehicleOrderBatchScheduler** runs the **VehicleOrderBatch** job at 12:00 PM. The batch job finds all pending orders (e.g., if a prior order was made when stock was 0) and, upon confirming stock is now available, **updates the Order Status__c from 'Pending' to 'Confirmed'**.
4. **Dealer Assignment (Flow):** As a new, fulfillable order is confirmed, the **Auto Assign Dealer Flow** is initiated. It uses the customer's location to query the Vehicle_Dealer__c object and finds the nearest authorized dealer (e.g., "WhatsNext Motors Dallas"). The flow then automatically populates the Dealer lookup field on the Order. *Result: The order is instantly ready for fulfillment by the correct dealer.*
5. **Proactive Communication (Flow):** If the customer simultaneously books a test drive for a different model, the **Test Drive Reminder Flow** schedules an email to be sent exactly 24 hours before the appointment. *Result: Reduced logistical errors and enhanced customer service.*

Conclusion

The WhatsNext Vision Motors Salesforce implementation represents a significant advancement in leveraging CRM technology to address core business needs in the automotive sector. By seamlessly integrating **Apex triggers for real-time validation**, **Flows for operational automation**, and **Batch/Scheduled Apex for scalable data management**, the project has successfully built a system that is efficient, accurate, and customer-centric. This robust solution not only meets all defined requirements—from automated dealer assignment to guaranteed stock accuracy—but also positions WhatsNext Vision Motors for enhanced customer loyalty and sustained operational excellence in the future.