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

ABSTRACT

The Salesforce CRM implementation project at WhatsNext Vision Motors aims to revolutionize the customer ordering and service experience in the automotive industry. By leveraging Salesforce's robust platform capabilities, the project focuses on improving order accuracy, enhancing customer satisfaction, and increasing operational efficiency. Core functionalities include automatic dealer assignment based on customer location, real-time stock validation, and scheduled automation for bulk order status updates. This initiative is designed to minimize manual intervention, reduce processing delays, and provide a seamless customer journey from inquiry to vehicle delivery.

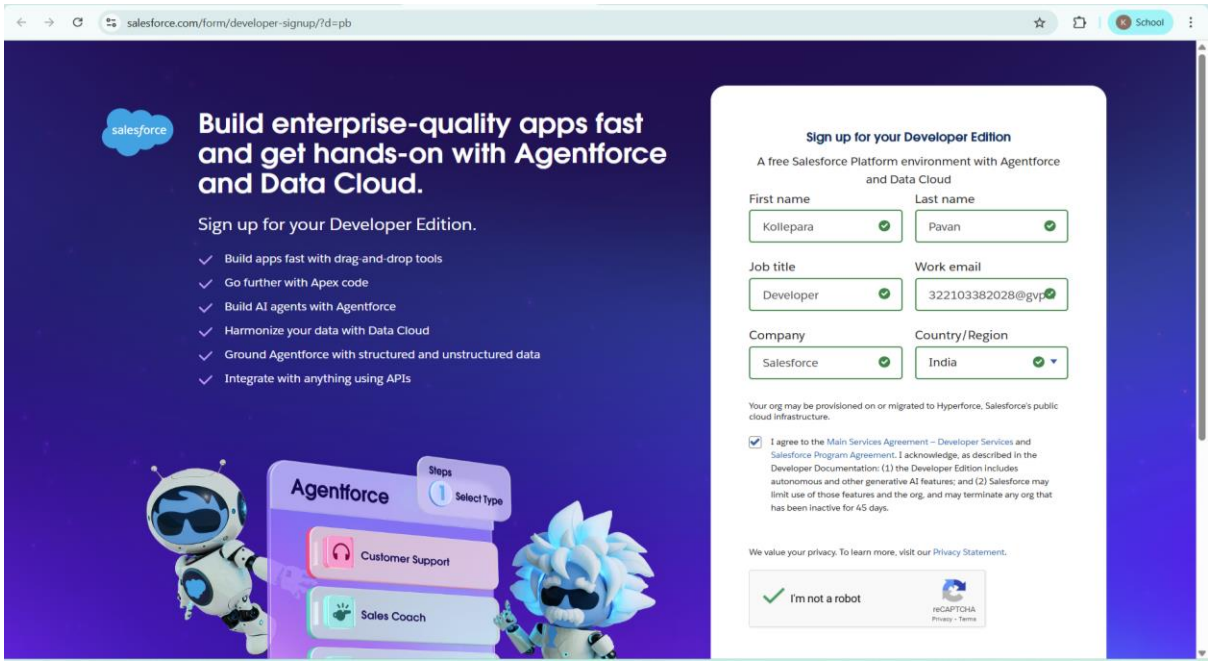
OBJECTIVE

- To implement Salesforce CRM for centralized management of vehicle inventory, dealer details, customer orders, test drives, and service requests.
- To automate key workflows including:
    - Nearest dealer assignment based on customer address.
    - Order restriction if vehicle stock is unavailable.
    - Email notifications for test drives and stock alerts.
- To develop Apex triggers and trigger handlers that enforce stock validation and automatic dealer assignment logic in a modular and maintainable way.
- To build batch Apex jobs for periodically updating stock levels and sending scheduled notifications for inventory management and order status tracking.
- To improve overall customer satisfaction and reduce administrative overhead through efficient process automation.

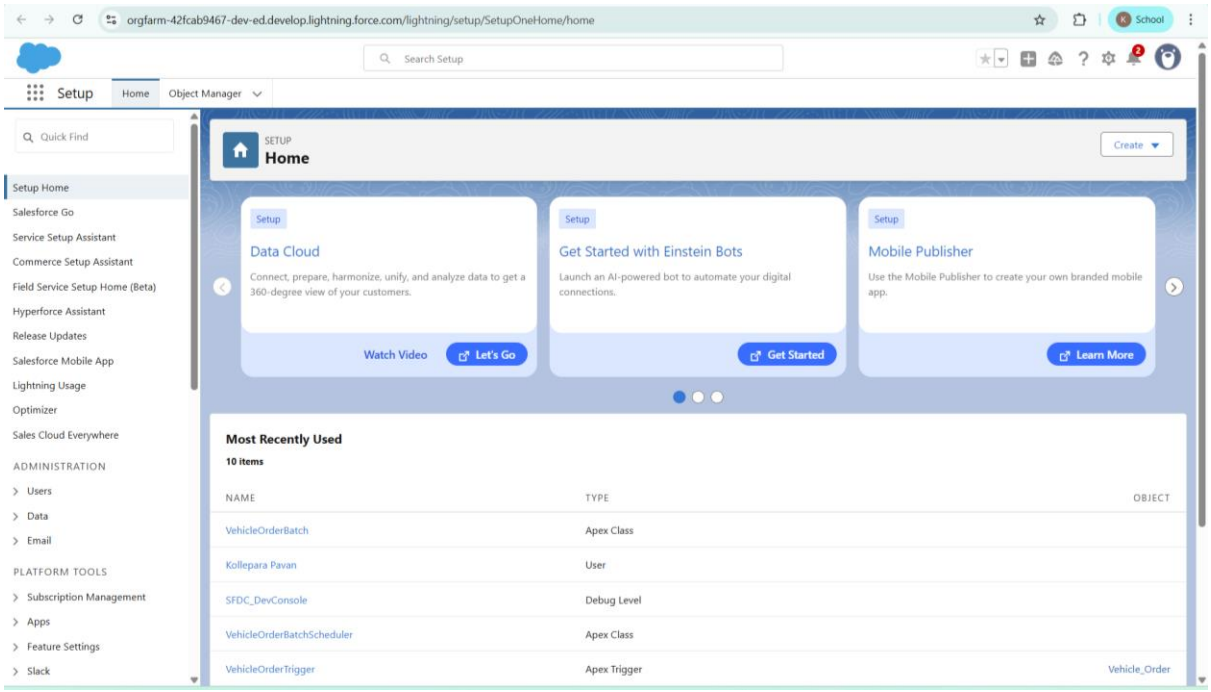
DETAILED EXECUTION OF PROJECT PHASES

Epic-1: Salesforce Credentials Creation

I have created a Salesforce Developer Account using my personal details and set up the necessary credentials to begin my project titled "WhatNext Vision Motors: Shaping the Future of Mobility with Innovation and Excellence." This account will serve as the development environment for building and showcasing my work. With the verifier granted access, they can now monitor and evaluate the progress and contributions I make toward this innovative mobility solution.

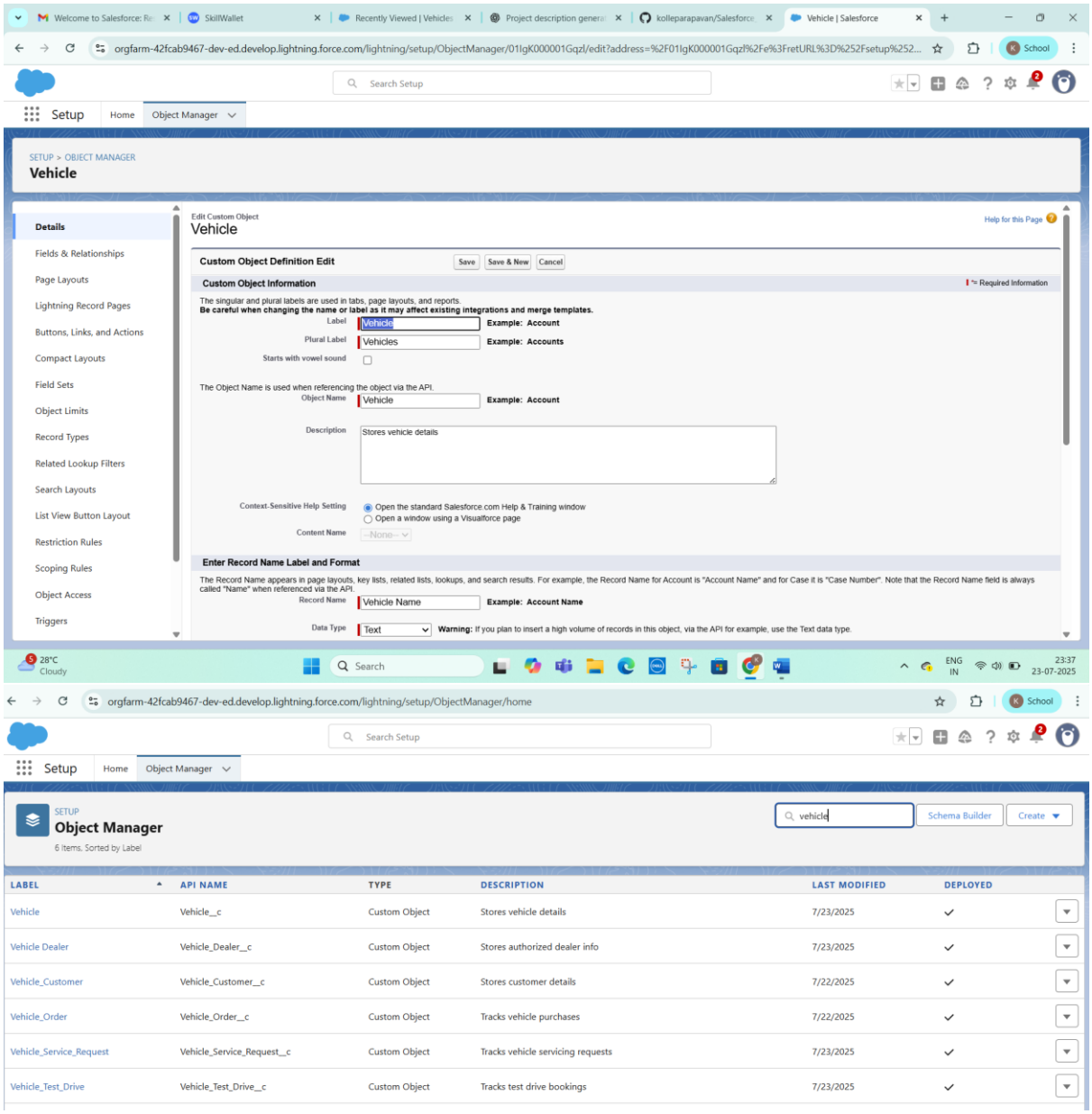


After Resetting the password login to the trailhead page.



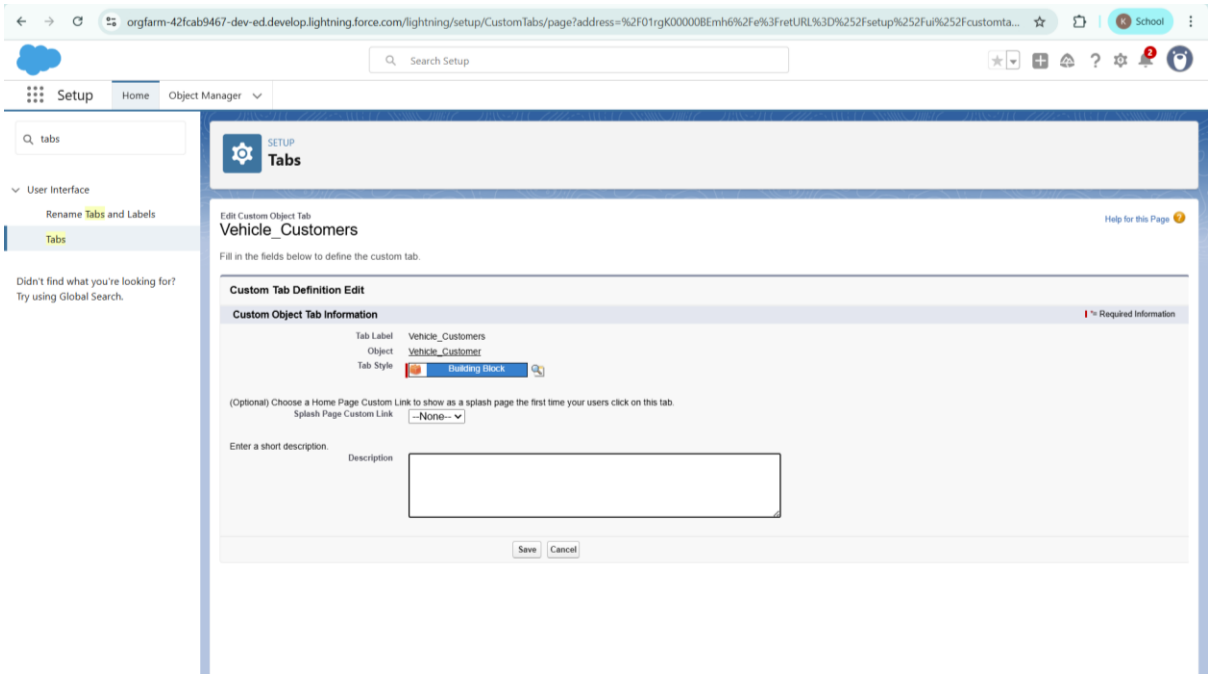
Epic-2: Data Management-Objects

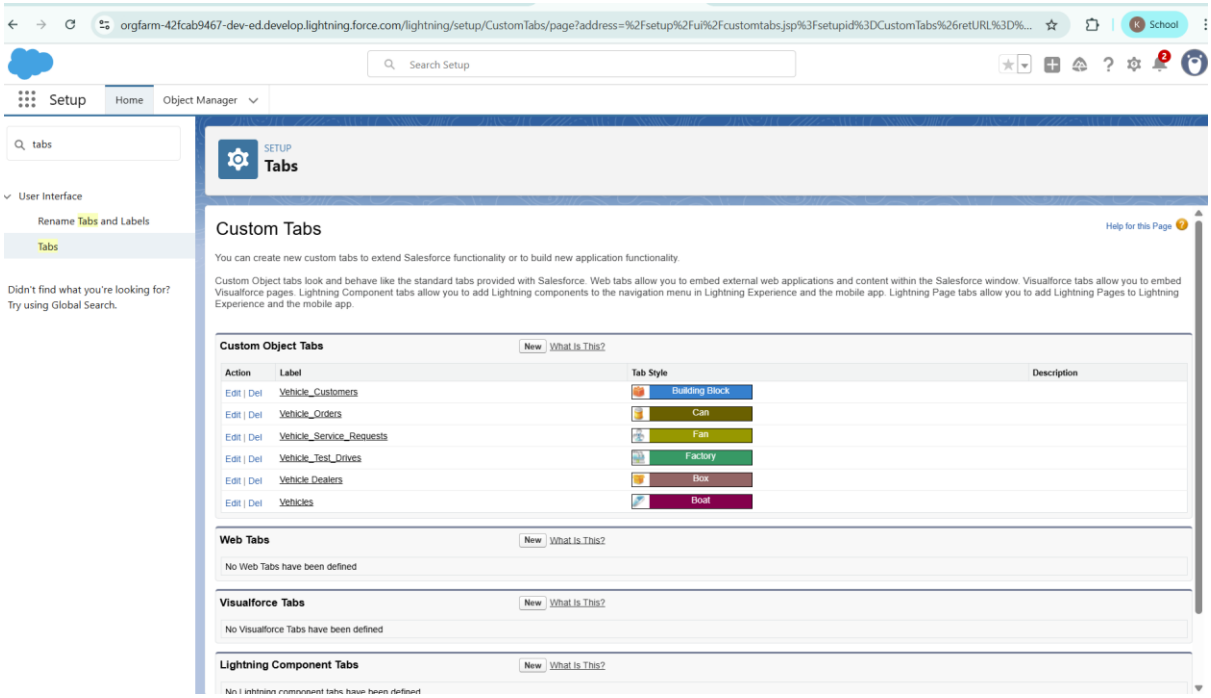
As part of my project "WhatNext Vision Motors: Shaping the Future of Mobility with Innovation and Excellence," I have successfully created six essential custom objects within Salesforce to manage and structure project data effectively. These objects include Vehicle__c for storing vehicle details, Vehicle_Dealer__c for managing authorized dealer information, Vehicle_Customer__c for capturing customer records, Vehicle_Order__c for tracking vehicle purchases, Vehicle_Test_Drive__c for handling test drive bookings, and Vehicle_Service_Request__c for managing service-related queries. I have established appropriate relationships among these objects to ensure smooth data flow and integration—for example, linking vehicles and dealers to orders, customers to both test drives and orders, and vehicles to service requests. This object structure forms the backbone of the entire project, supporting seamless data management and real-time interaction across all components.



Epic-3: Data Management-Tabs

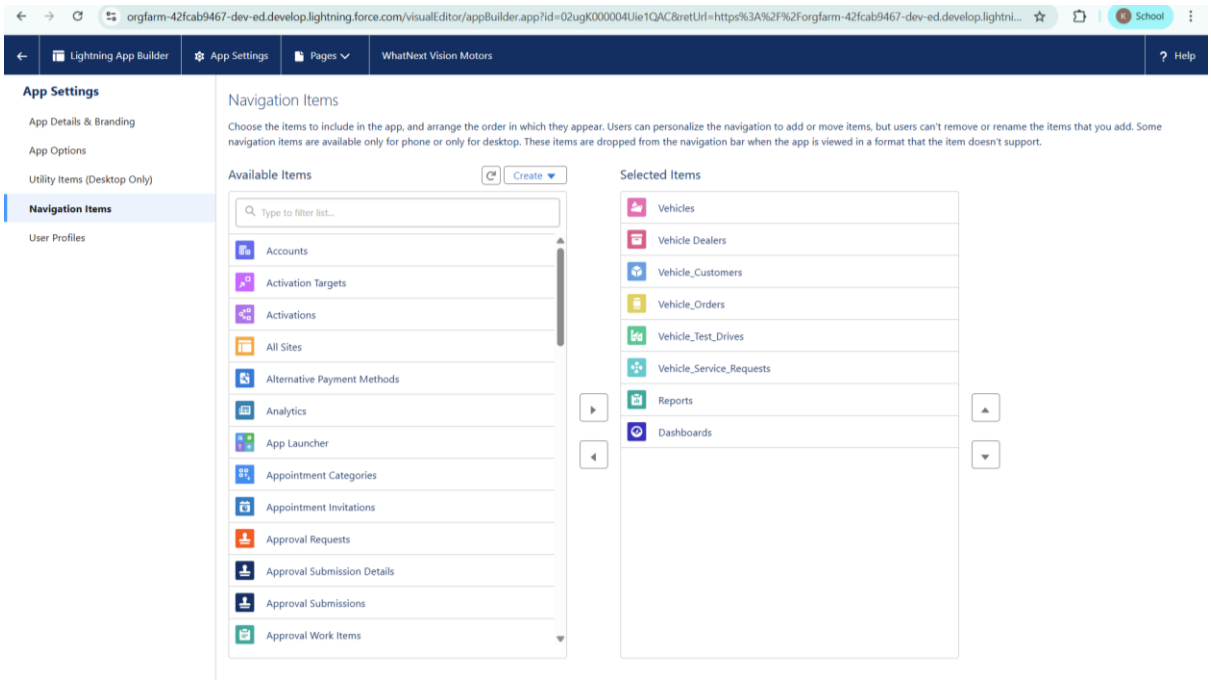
To enhance accessibility and usability within the Salesforce environment, I created custom tabs for all six custom objects used in the project: Vehicle__c, Vehicle_Dealer__c, Vehicle_Customer__c, Vehicle_Order__c, Vehicle_Test_Drive__c, and Vehicle_Service_Request__c. These tabs allow users to easily navigate, view, and manage records associated with each object directly from the Salesforce interface. By setting up these custom tabs, I ensured that all relevant data is readily available for stakeholders, streamlining the user experience and improving overall efficiency in managing vehicle-related processes within the project.

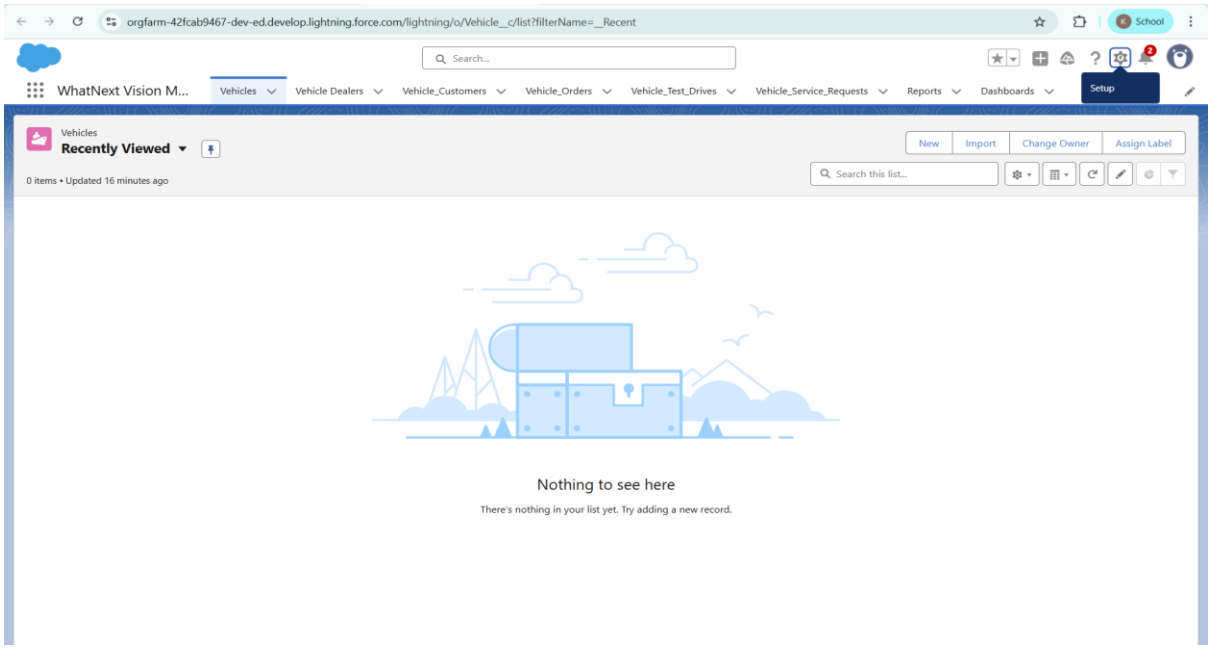




Epic-4: Data Management-App Manager

As part of organizing and streamlining the user experience in my Salesforce project, I created a Lightning App named "WhatNext Vision Motors" to serve as a centralized workspace for managing all aspects of the vehicle business process. This app integrates all key components of the project—including Vehicles, Dealers, Customers, Orders, Test Drives, and Service Requests—into a single, user-friendly interface. I included essential navigation items and custom objects to ensure quick access to relevant records and functionalities. Additionally, I assigned the System Administrator profile to ensure full access and control over the app's features. This app provides a cohesive and efficient way for users to interact with the system, improving workflow and data visibility across all departments involved in the mobility solution.

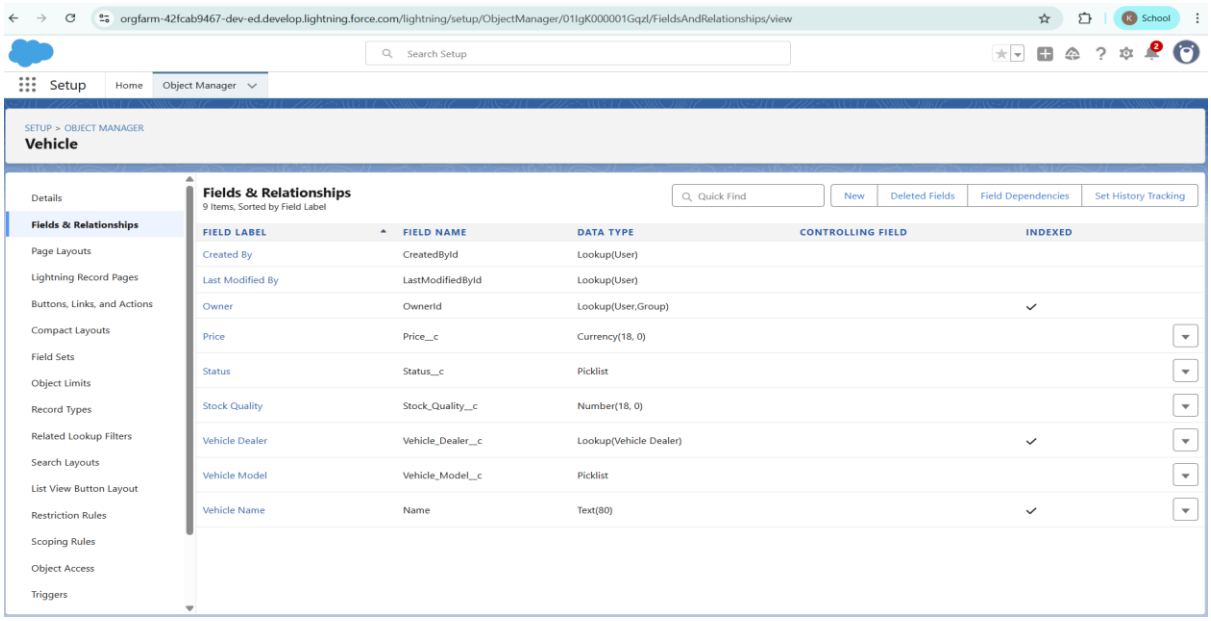




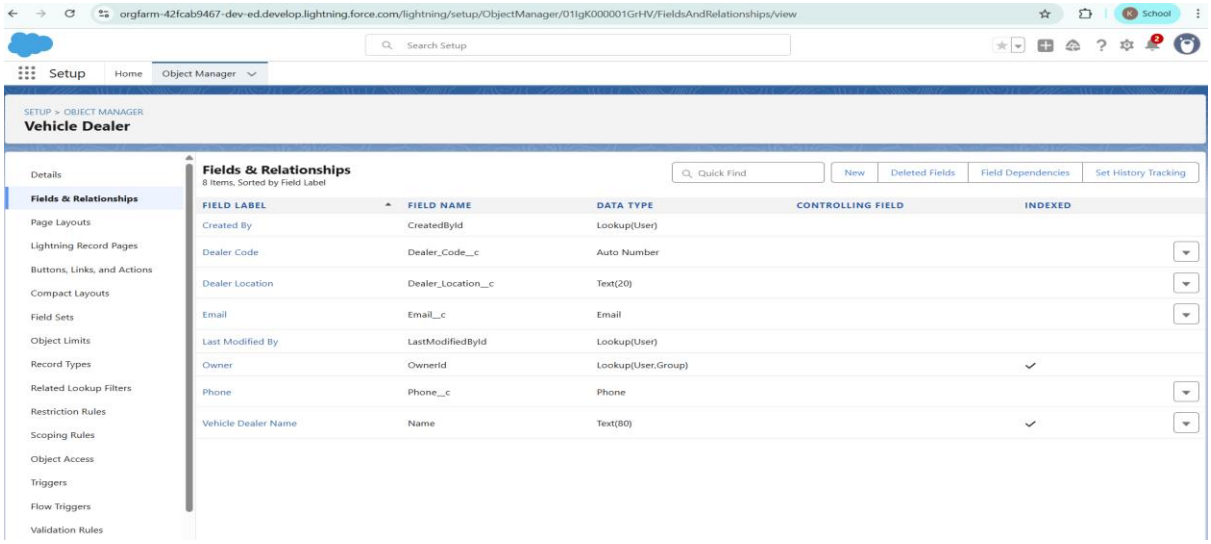
Epic-5: Data Management-Fields

As part of building the **"WhatNext Vision Motors"** Salesforce application, I created six custom objects with all the necessary fields to handle business operations like vehicle management, customer tracking, dealer coordination, orders, test drives, and service requests. Each object was carefully structured with appropriate field types such as Text, Picklist, Lookup, Date, Currency, and Auto Number to ensure accurate and efficient data storage.

1. Vehicle__c:
Vehicle_Name__c, Vehicle_Model__c, Stock_Quantity__c, Price__c, Dealer__c, Status__c

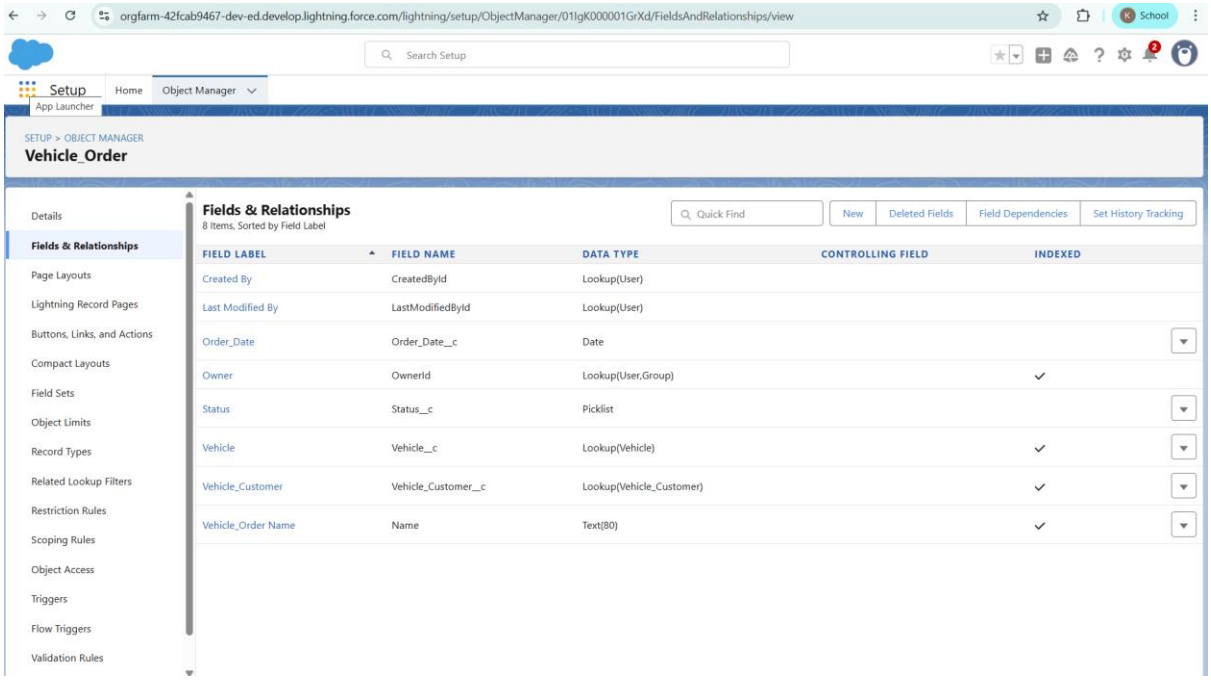


2. Vehicle_Dealer__c:
Dealer_Name__c, Dealer_Location__c, Dealer_Code__c, Phone__c, Email__c



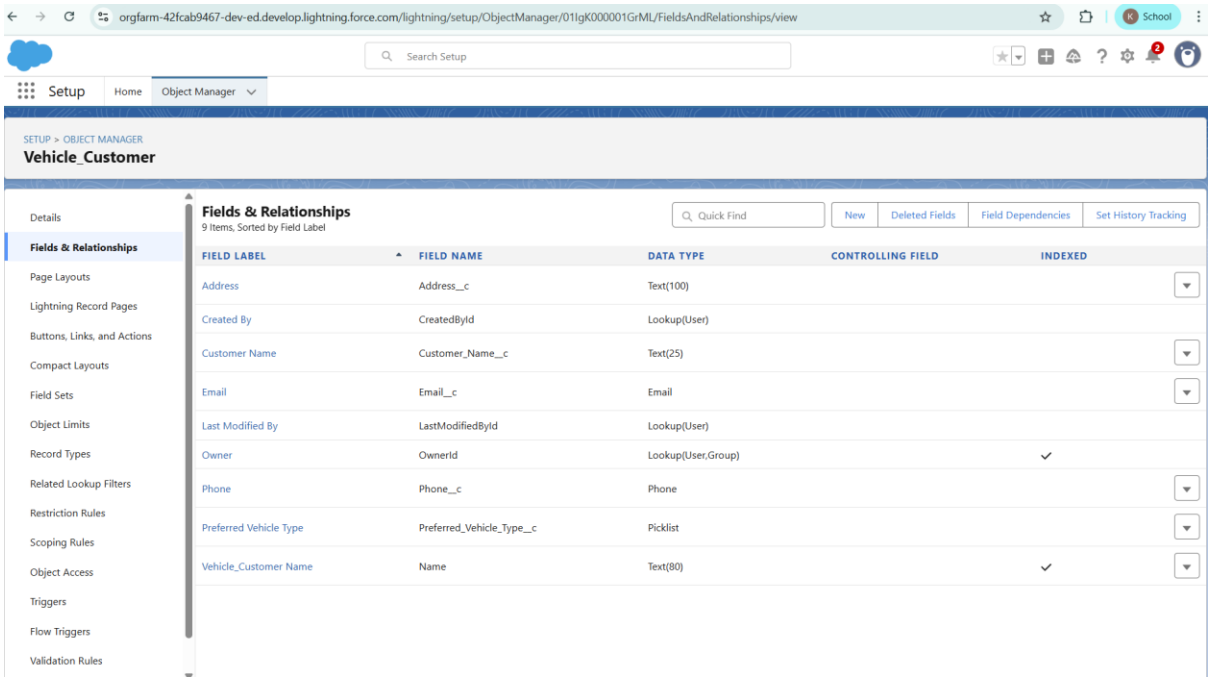
3. Vehicle_Order__c:

Customer__c, Vehicle__c, Order_Date__c, Status__c



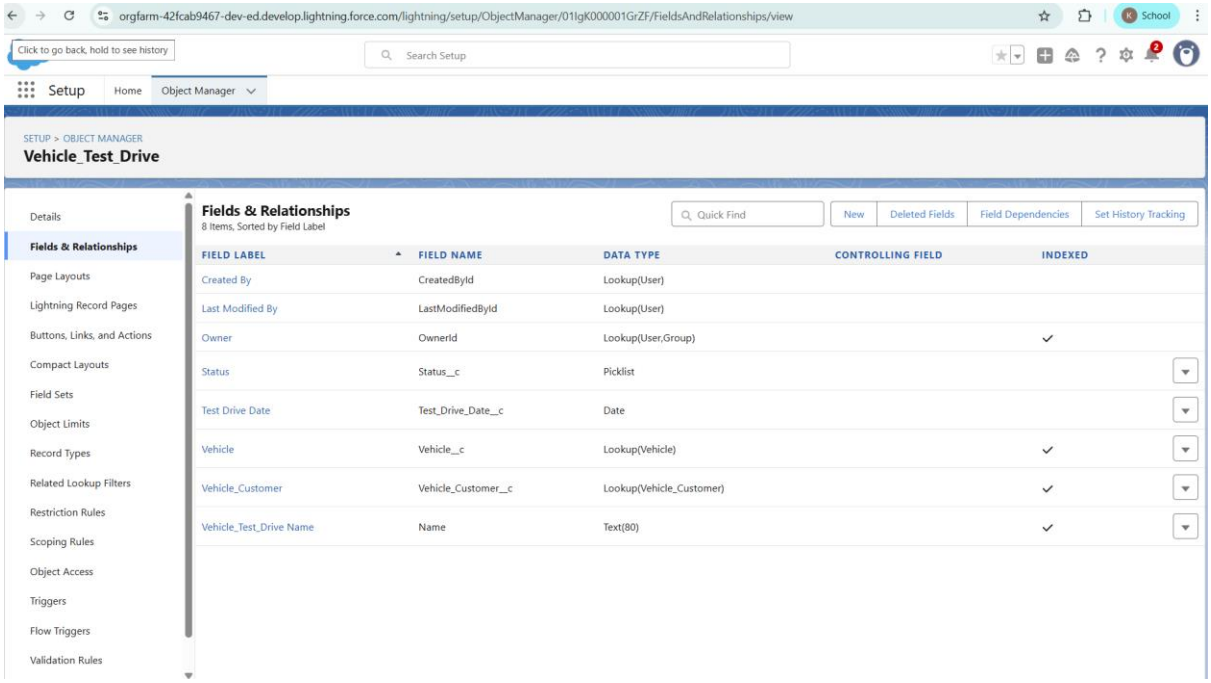
4. Vehicle_Customer__c:

Customer_Name__c, Email__c, Phone__c, Address__c, Preferred_Vehicle_Type__c



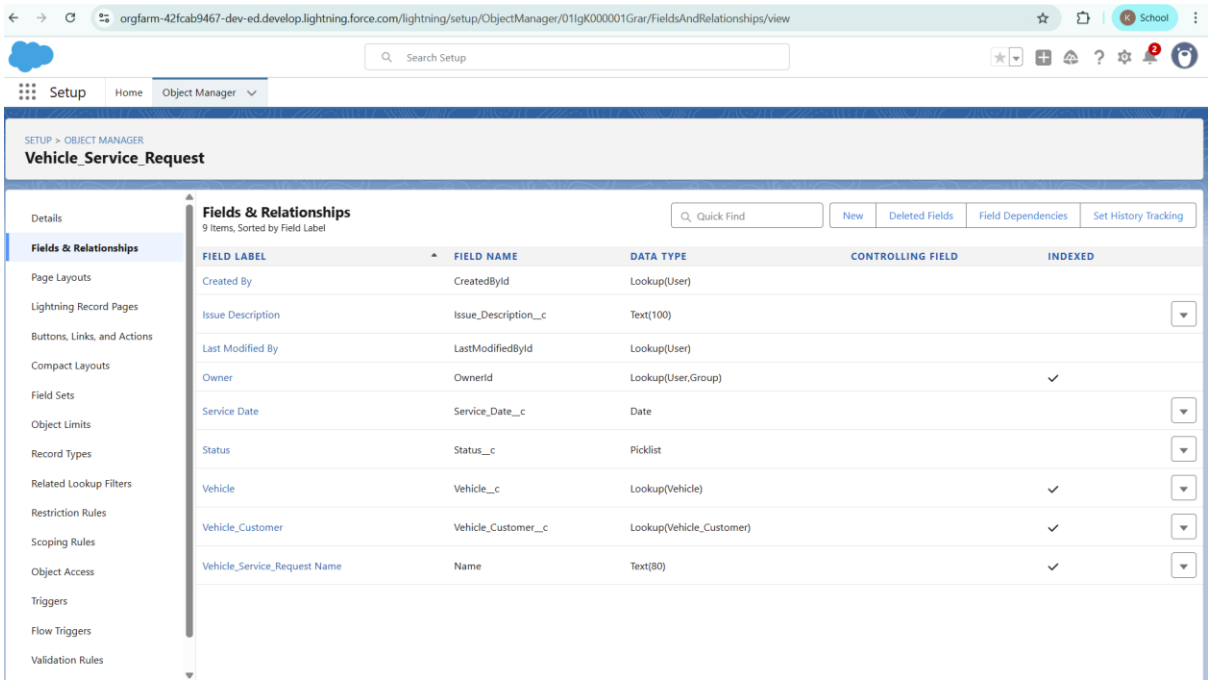
5. Vehicle_Test_Drive__c:

Customer__c, Vehicle__c, Test_Drive_Date__c, Status__c



6. Vehicle_Service_Request__c:

Customer__c, Vehicle__c, Service_Date__c, Issue_Description__c



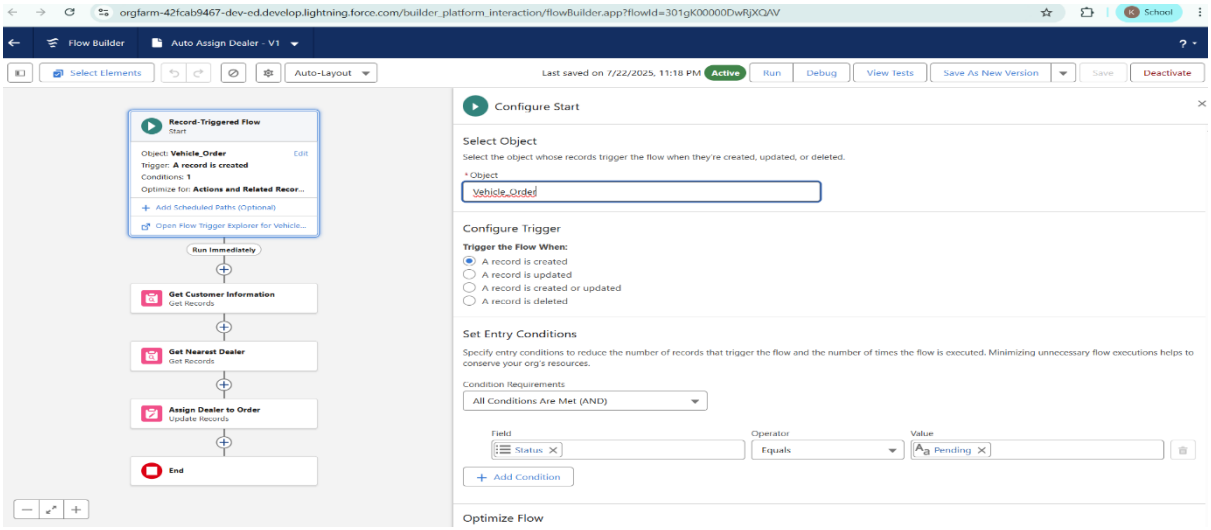
These custom fields enable full tracking of all interactions and transactions within the system. By designing them logically and linking them through relationships, the app provides a seamless experience for managing every aspect of the vehicle sales and service lifecycle.

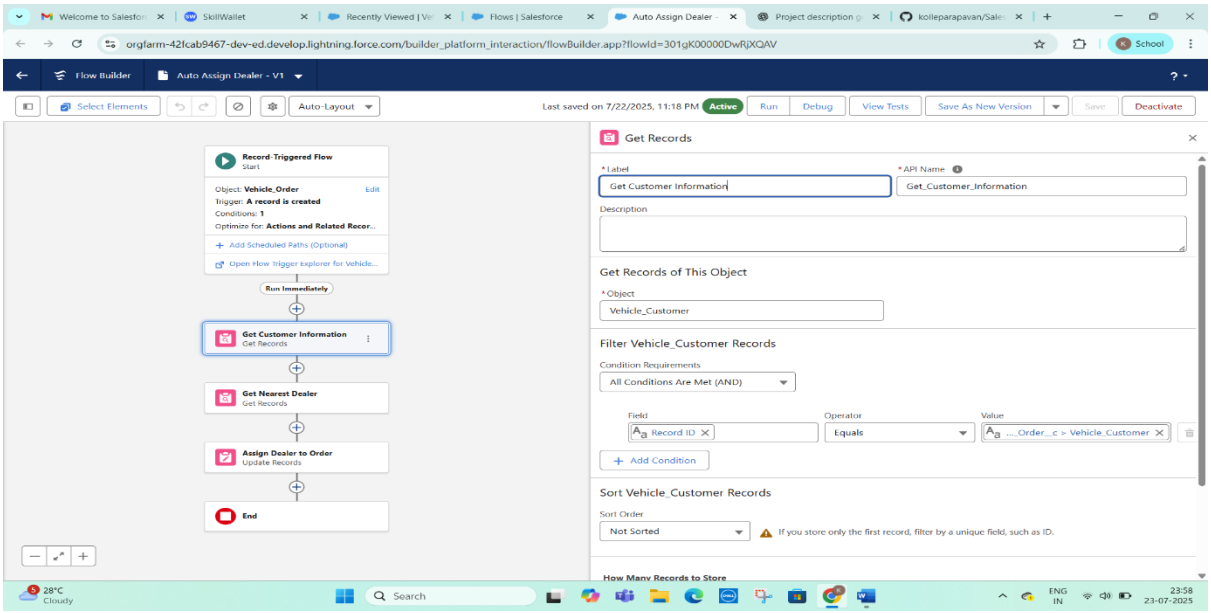
Epic-6: Automation

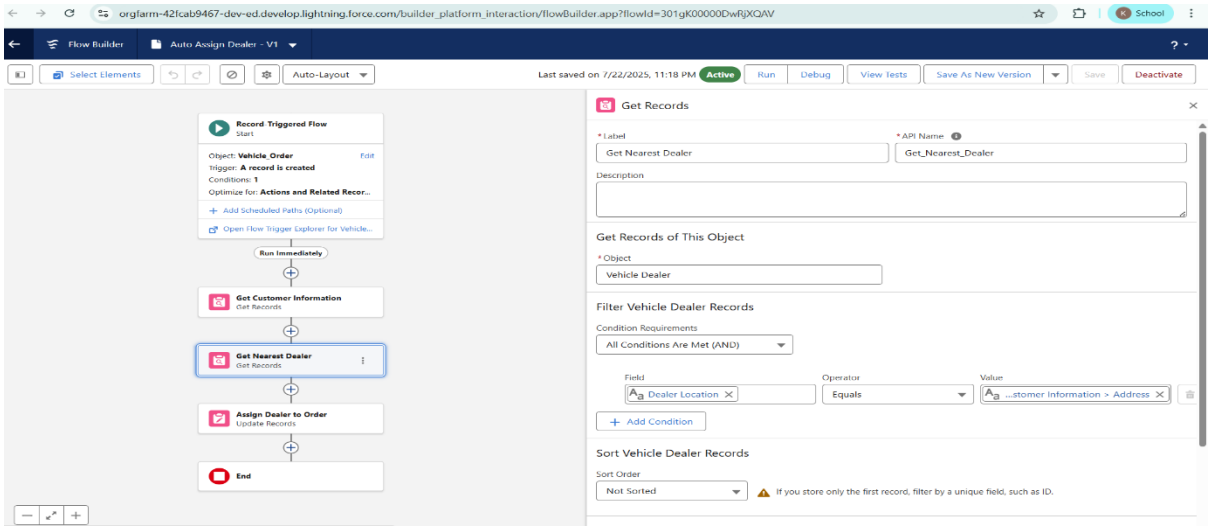
As part of automating key processes in the **"WhatNext Vision Motors"** Salesforce application, I created two **Record-Triggered Flows** to enhance efficiency and customer engagement.

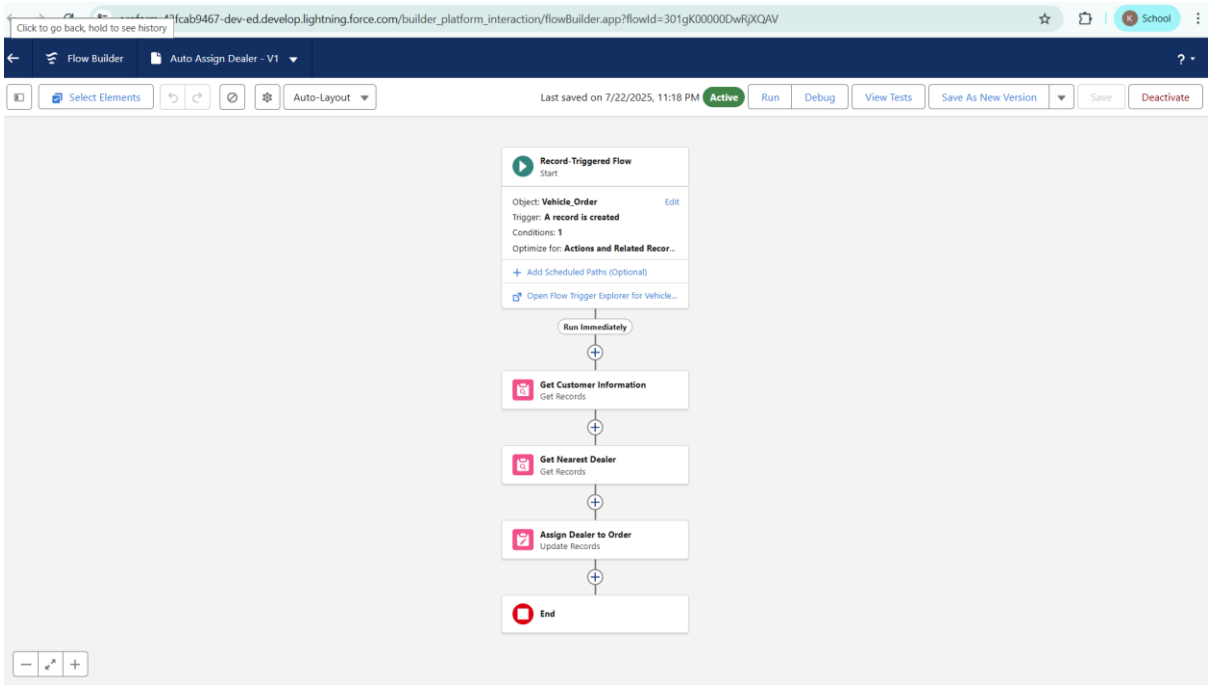
1. **Assign Nearest Dealer to Customer:**
   This flow is triggered when a new customer record is created. It automatically identifies and assigns the nearest authorized dealer based on the customer's location. This ensures a more personalized experience and helps in streamlining the order and service process by connecting customers with the most relevant dealer.

2. **Send Test Drive Reminder Email:**
   This flow is triggered when a new test drive record is created. It sends an automated email reminder to the customer with details about their scheduled test drive. This improves communication, reduces no-shows, and ensures that customers are well-informed and engaged with the brand.

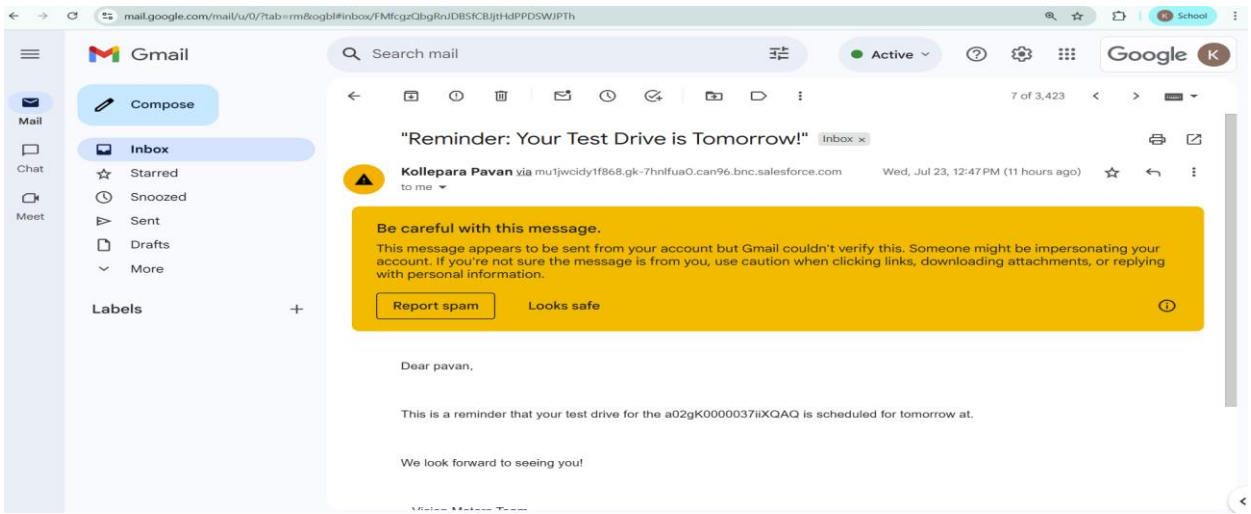








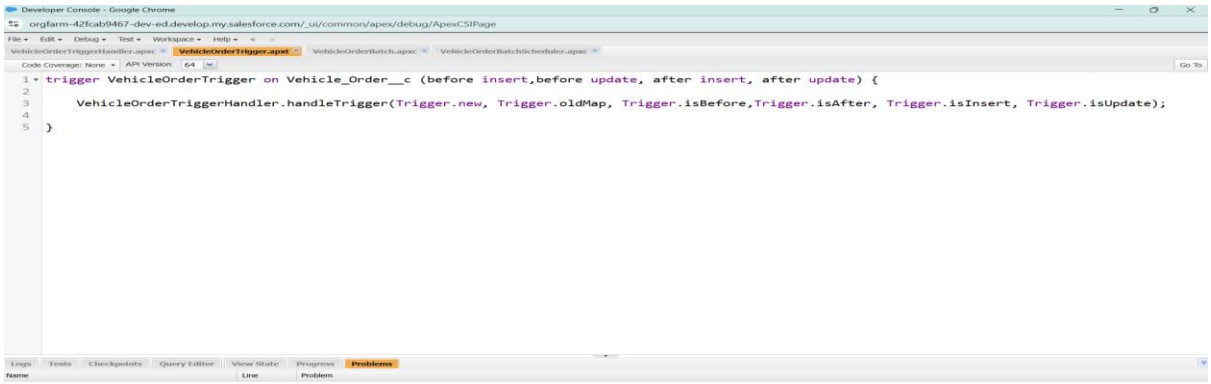
Tested the flow by creating the customer



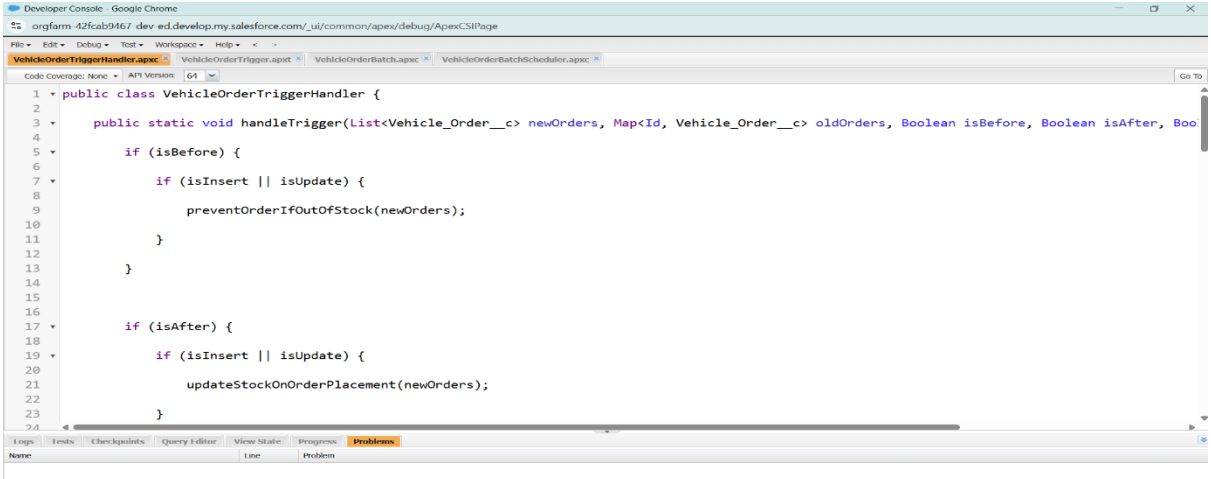
Epic-7: Apex and Batch Class

To automate the **vehicle availability tracking and order processing workflow** in the *WhatNext Vision Motors* project, I implemented Apex-based automation using **triggers, handler classes, batch jobs, and schedulers**. These components ensure that vehicle orders are processed automatically and efficiently on a daily basis, reducing manual intervention and increasing accuracy.

- **VehicleOrderTrigger**
  - This is the **Apex trigger** written on the Vehicle_Order__c object.
  - It is responsible for executing logic whenever a vehicle order is created or updated.
  - The trigger delegates its logic to the handler class to keep the code modular and maintainable.

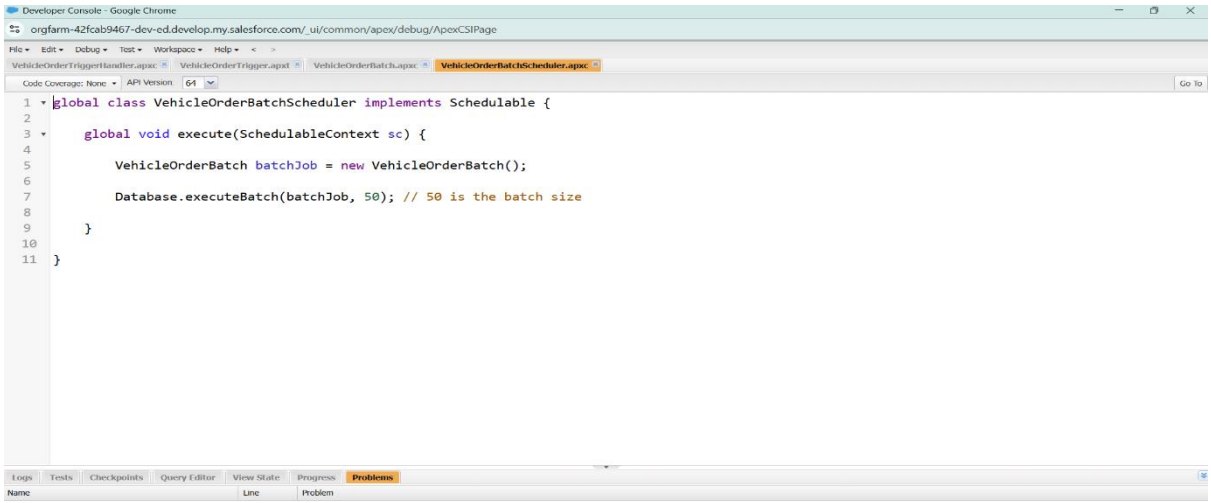


- **VehicleOrderTriggerHandler**
  - This class contains the **business logic** related to the VehicleOrderTrigger.
  - It checks vehicle availability, updates order status, and manages relationships with vehicles and dealers.
  - Keeping logic in this separate handler promotes **clean code architecture** and **reusability**.



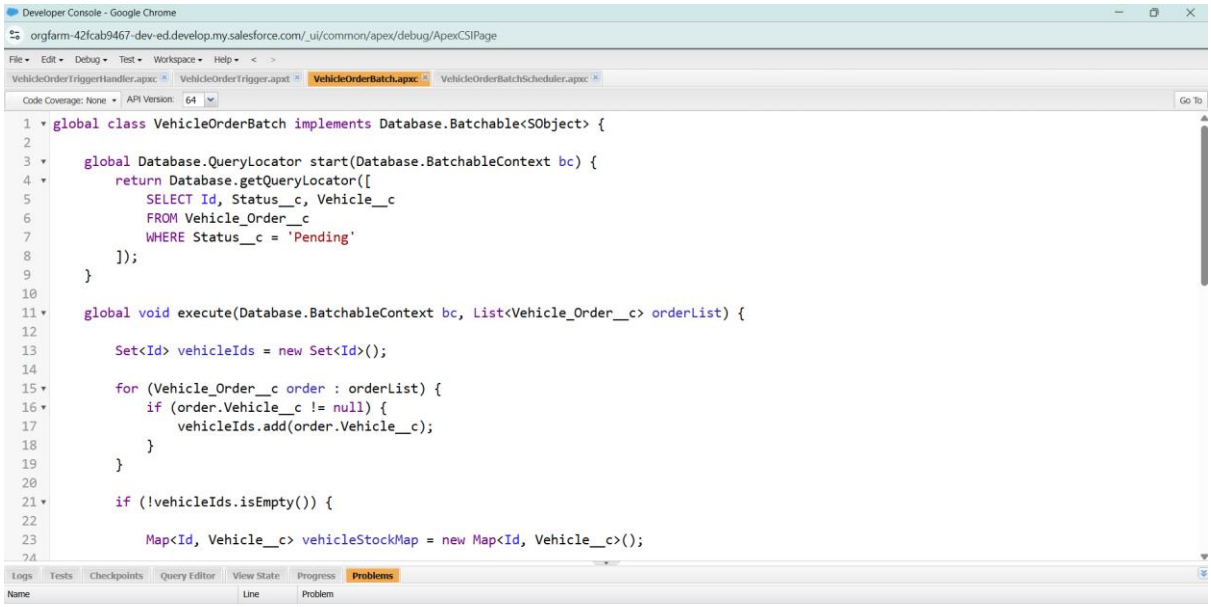
🔹 VehicleOrderBatch

- This is a **batch class** that processes large numbers of vehicle orders in bulk.
- It queries all pending vehicle orders and performs actions such as assigning vehicles, updating availability, or marking orders as processed.
- Batch processing is used here to handle **large data volumes efficiently** and without hitting governor limits.

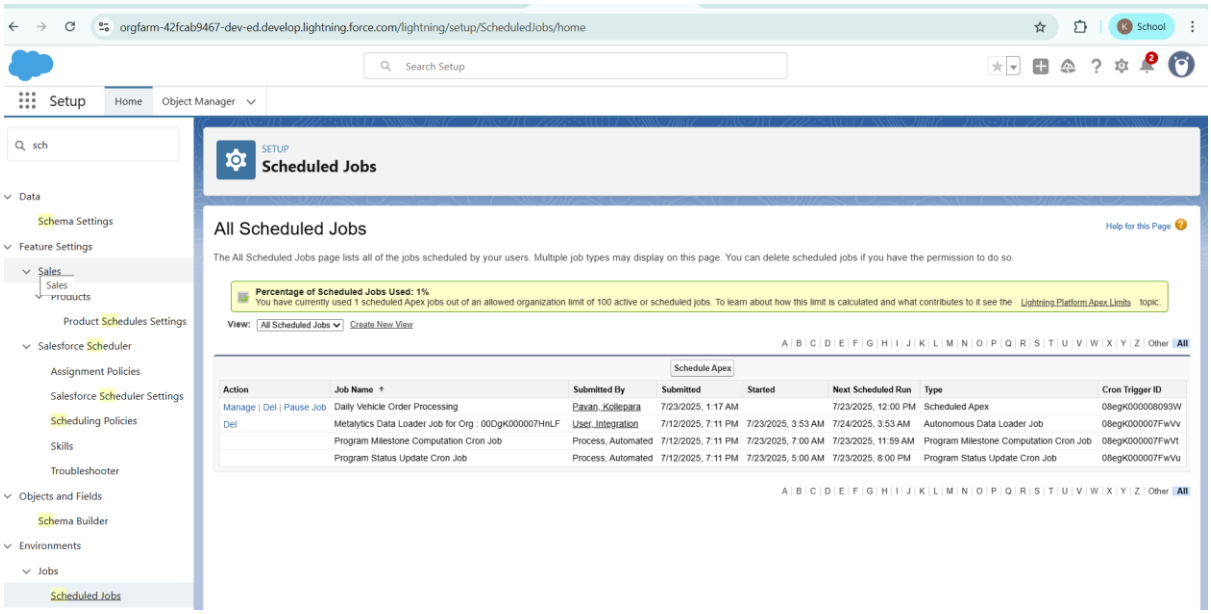


🔹 VehicleOrderBatchScheduler

- This class is used to **schedule the batch job to run automatically** at a specific time.
- It leverages the System.schedule method to ensure the VehicleOrderBatch job runs every day without manual triggering.



- **Scheduled Job Monitoring**
  - Once the scheduler is deployed, you can monitor the scheduled job from **Setup > Scheduled Jobs** in Salesforce.
  - This allows tracking of batch job execution, checking for errors, and ensuring it runs as intended.



Project Demonstration – Functional Walkthrough with Sample Data

Sample Data Setup:

Vehicle Customers:

- Customer-1
  - Customer Name : Pavan
  - Email : 322103382028@gvpce.ac.in
  - Phone :6303643871
  - Address :Visakhapatnam
- Customer-2
  - Customer Name :Madhu
  - Email : abc@gmail.com
  - Phone :6303643871
  - Address : Hyderabad

Vehicle Dealer:

- Dealer-1
  - Dealer Name :Satya
  - Dealer Location :Visakhapatnam
  - Phone :6303643871
  - Email :abc@gmail.com
- Dealer-2
  - Dealer Name :Naga
  - Dealer Location :Hyderabad
  - Phone :6303643871
  - Email :xyz@gmail.com

Vehicles:

- Vehicle-1
  - Vehicle Name :RX_100
  - Vehicle Model :SUV
  - Stock Quantity :5
  - Vehicle Dealer :Madhu
  - Price :500000
  - Status :Available
- Vehicle-2
  - Vehicle Name :AUDI
  - Vehicle Model :Sedan

- Stock Quantity :0
- Vehicle Dealer :Naga
- Price :1000000
- Status :Out of Stock

Vehicle Orders:

- Vehicle-1
  - Vehicle Order Name :car
  - Vehicle Customer :Pavan
  - Vehicle :RX_100
  - Order_Date :7/28/2025
  - Status :Pending
- Vehicle-1
  - Vehicle Order Name :Bike
  - Vehicle Customer :Pavan
  - Vehicle :RX_100
  - Order_Date :7/29/2025
  - Status :Pending

Flow Execution

After the above data is created, here's what happens automatically:

Dealer Assignment Flow:

When a new Vehicle Order is created with status Pending, the flow:

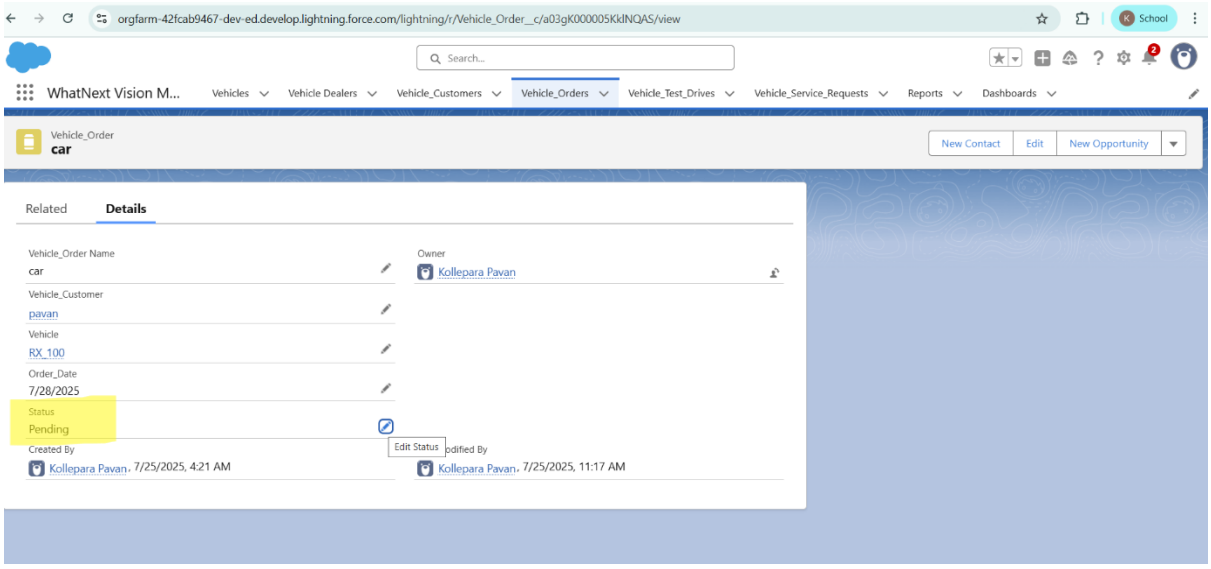
- Fetches the selected Vehicle__c
- Retrieves its associated Dealer__c
- Assigns that dealer to the Vehicle_Order__c (via lookup)
- Status remains Pending until confirmed

Trigger/Batch Apex:

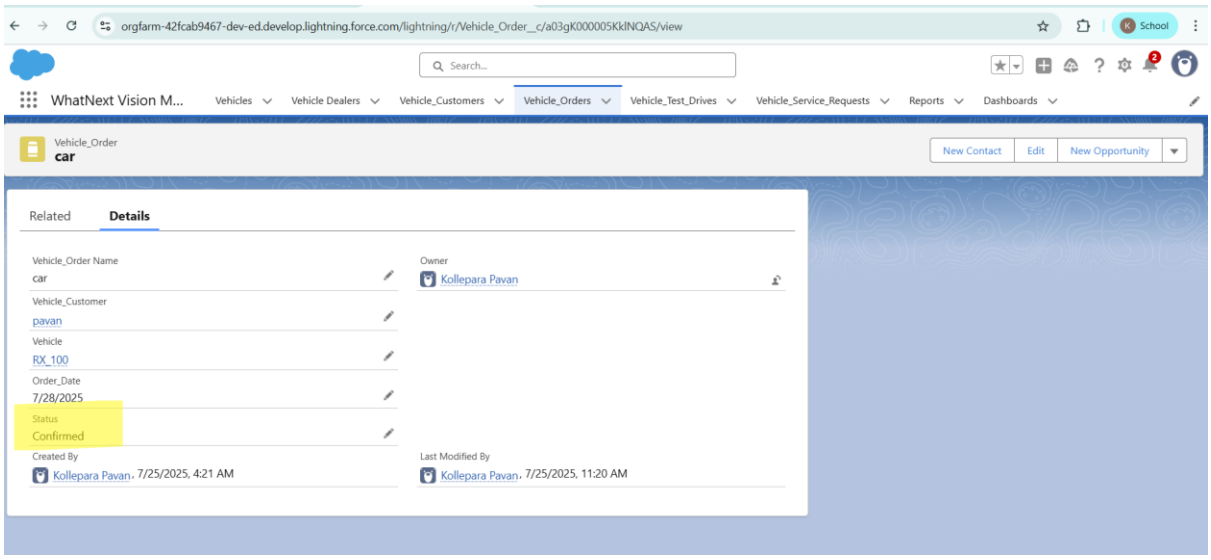
If a vehicle is out of stock, order cannot be placed (error shown via addError() in trigger)

- When the Status__c of order changes to Confirmed, the Trigger or Batch Apex:
  - Reduces the vehicle stock by 1
  - Prevents over-ordering
  - Keeps vehicle stock synced

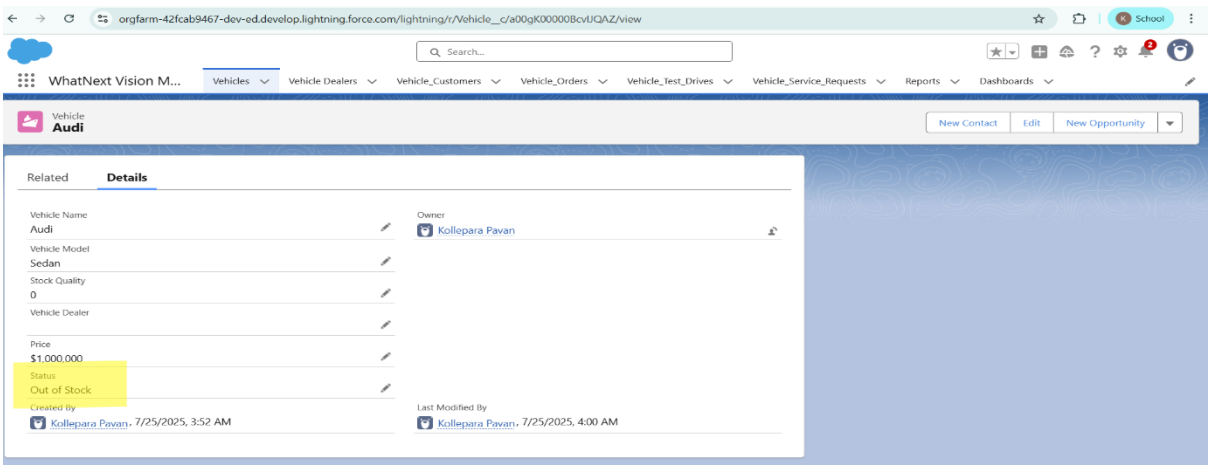
Before the Flow executed the Status is Pending:

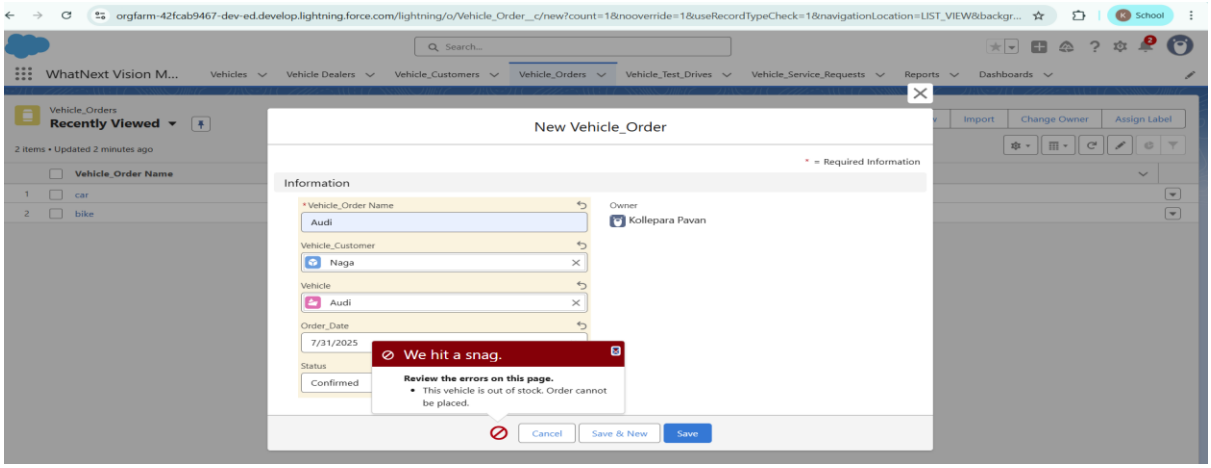


After the Execution of Flow

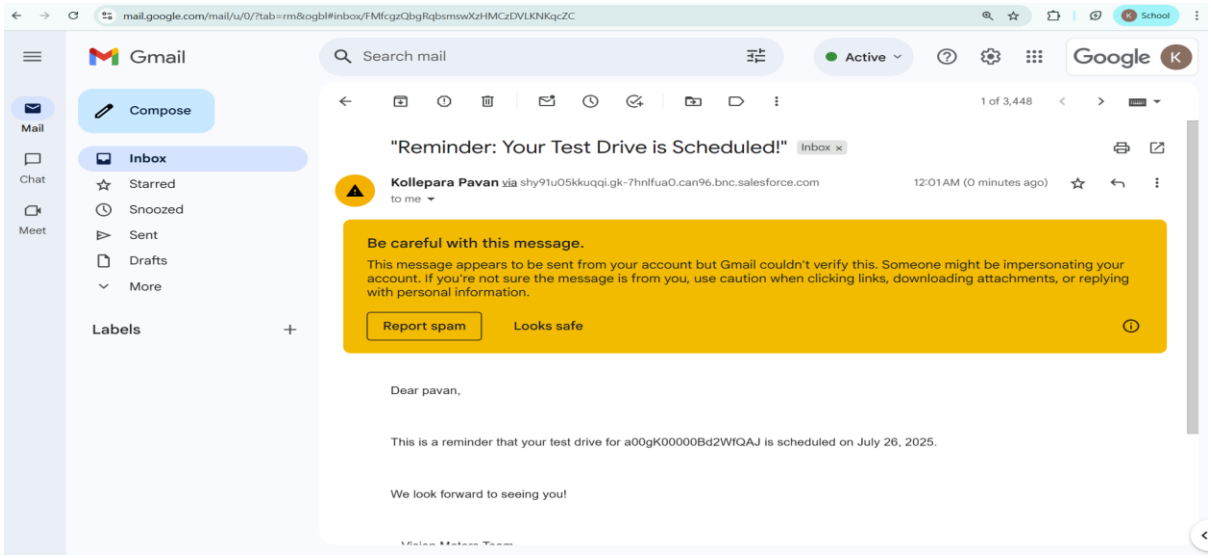


Here Audi car is Out of stock.lets see the order is creating or not:

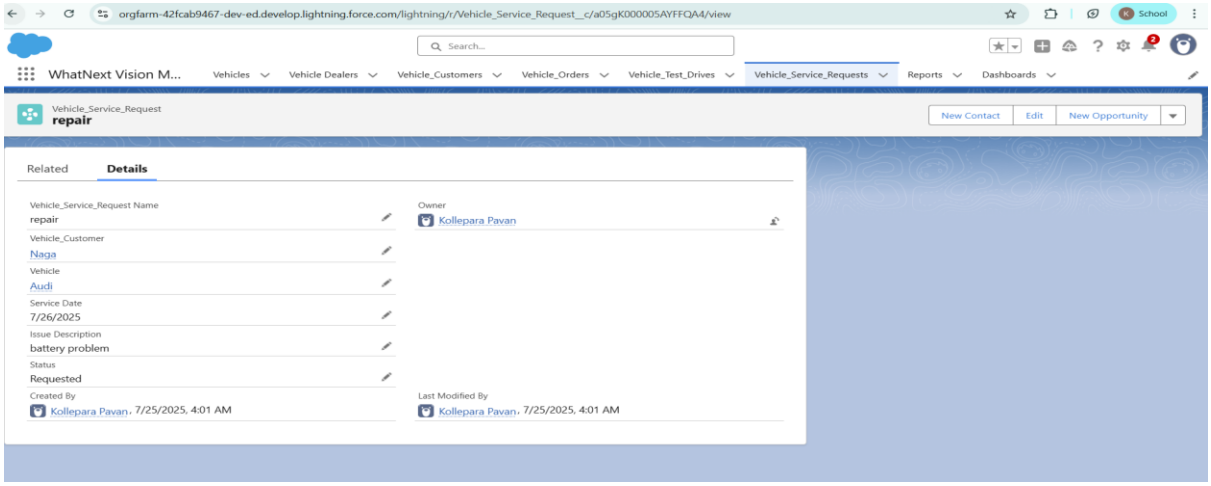




When a test drive is scheduled then email will be send Automatically to customer like this:



This is how a customer can raise a vehicle service request in the system:



Conclusion:

Through the development of the **"WhatNext Vision Motors"** Salesforce application, I gained hands-on experience in building a complete, scalable business solution using Salesforce tools and automation features. I successfully designed and created custom objects, fields, and relationships to represent real-world entities like vehicles, customers, dealers, orders, test drives, and service requests. I enhanced user experience by setting up a custom **Lightning App**, creating **custom tabs**, and configuring **navigation items**.

On the automation side, I developed **record-triggered flows** to assign the nearest dealer to a customer and to send automated test drive reminders, improving customer engagement. I implemented Apex-based automation by creating a structured set of classes including **triggers, handler classes, batch classes**, and a **scheduler** to process vehicle orders automatically on a daily basis. This taught me how to handle real-time and scheduled processes in Salesforce effectively.

Overall, this project helped me understand and apply **Salesforce development principles**, automation best practices, and scalable design. It not only strengthened my technical skills but also gave me practical insights into building enterprise-level CRM solutions.