VEHICLE SERVICE MANAGEMENT SYSTEM WITH DEALER AUTO-ASSIGNMENT – WHATNEXT VISION MOTORS

ABSTRACT

The automotive industry demands high efficiency in service and order fulfillment. WhatNext Vision Motors implemented a Salesforce-based system to automate vehicle order processing, assign orders to nearby dealers, prevent orders for out-of-stock vehicles, and streamline customer communication. The project involved creating custom objects, automating processes with Flows, and developing Apex and Batch classes to manage stock and trigger updates. This system enhances customer experience, minimizes manual effort, and improves operational accuracy.

OBJECTIVES

- Fully digitize the end-to-end vehicle order and service management process.
- Automatically assign the nearest available dealer based on a customer's address using flows.
- Prevent customers from placing orders for vehicles that are out of stock using Apex triggers.
- Continuously track and update vehicle stock availability using batch and scheduled Apex jobs.
- Automate service scheduling and send timely email reminders to customers for test drives or services.
- Reduce manual data entry and ensure accuracy with validation rules.
- Improve customer engagement through automated status updates and communication.
- Enable scalable architecture that supports future business expansion or integration.

TECHNOLOGY STACK

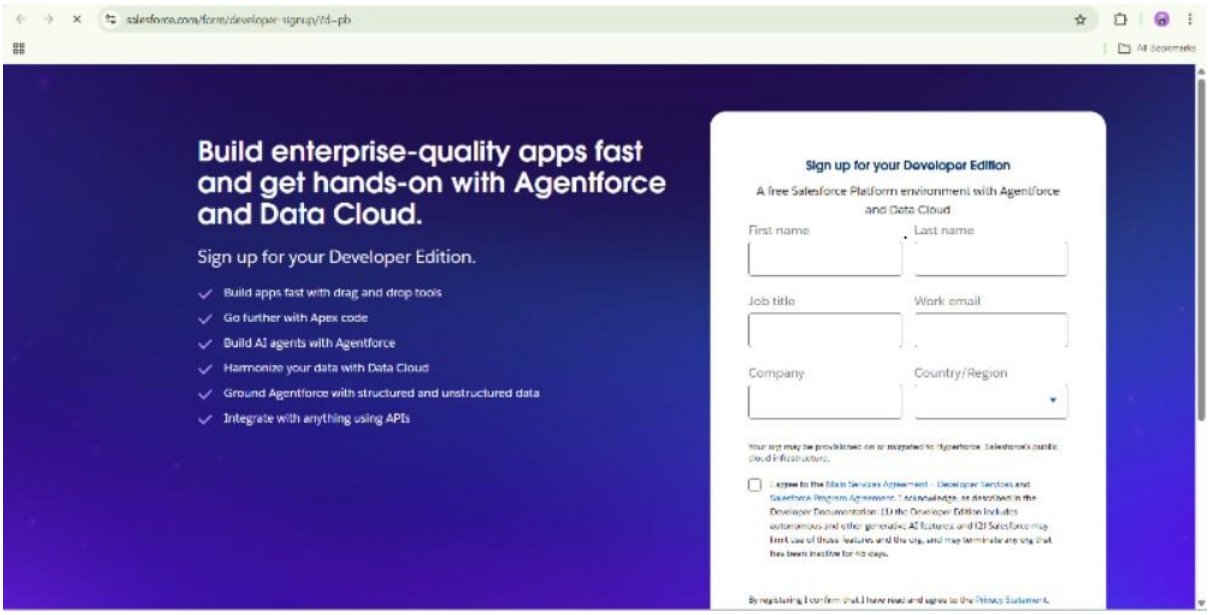
- **Salesforce CRM Platform:** The cloud-based platform used to build the entire application. It provides the foundation for object creation, automation tools, and UI configuration.
- **Lightning App Builder:** A point-and-click tool used to design custom apps and user interfaces. It allows the addition of tabs, pages, and components to tailor the experience for different users.

- **Flow Builder:** Used for building automation without writing code. Two types of flows were used:
    - **Record-Triggered Flows:** Triggered automatically when a record is created or updated. Used to assign dealers, update statuses, and send emails.
    - **Scheduled Flows:** Run at specific times to automate actions like status checks.
- **Apex Classes and Triggers:** Apex was used to implement advanced logic such as stock validation. Triggers enforce business rules during record operations.
- **Batch Apex:** Enables the processing of large data volumes in batches. Used to periodically check stock levels and update vehicle availability.
- **Scheduled Apex:** Apex jobs scheduled to run at defined intervals (e.g., daily). Used to send summaries and automate order updates.
- **Email Alerts and Notifications:** Configured through Flow Builder to send reminders and confirmations to customers and dealers. Ensures communication is automated and timely.

DETAILED EXECUTION OF PROJECT PHASES

Phase 1: Developer Org Setup

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

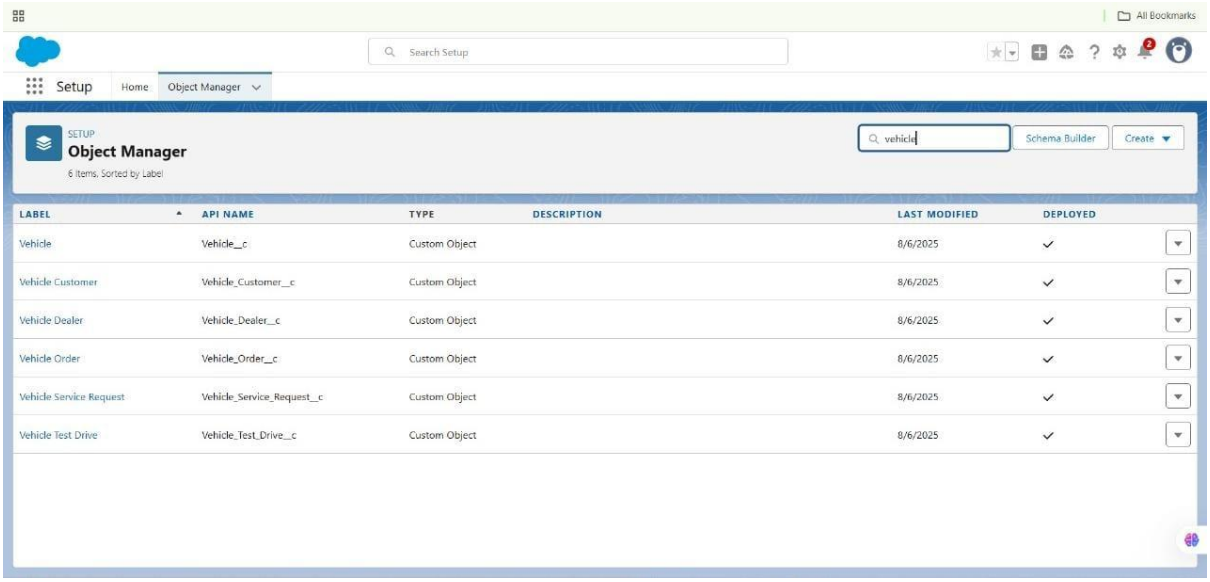


Phase 2: Requirement Gathering & Planning

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

Phase 3: Custom Object Creation

- Created six main custom objects:
  - **Vehicle__c** – Name, Type, Price, Stock Quantity
  - **Vehicle_Customer__c** – Name, Email, Phone, Address
  - **Vehicle_Dealer__c** – Name, Location, Assigned Vehicles
  - **Vehicle_Order__c** – Vehicle (lookup), Customer (lookup), Dealer (lookup), Status, Date
  - **Vehicle_Service_Request__c** – Customer (lookup), Vehicle (lookup), Request Date, Description, Status
  - **Vehicle_Test_Drive__c** – Customer (lookup), Vehicle (lookup), Preferred Date
- Configured relationships using Lookup fields.

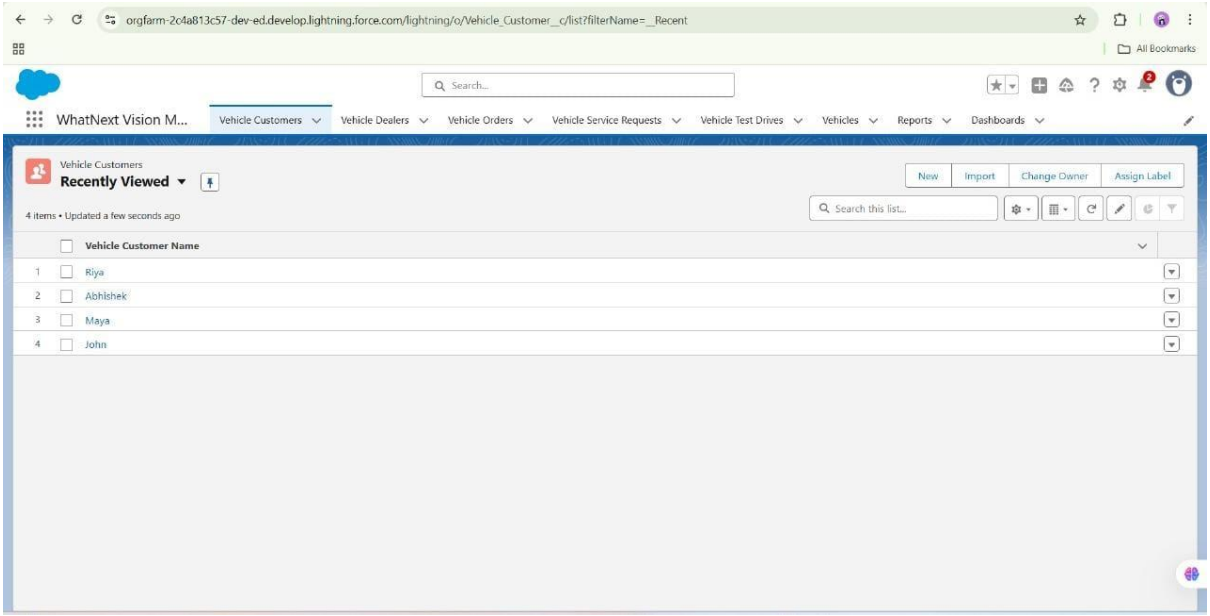


Phase 4: Field Creation and Validation Rules

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

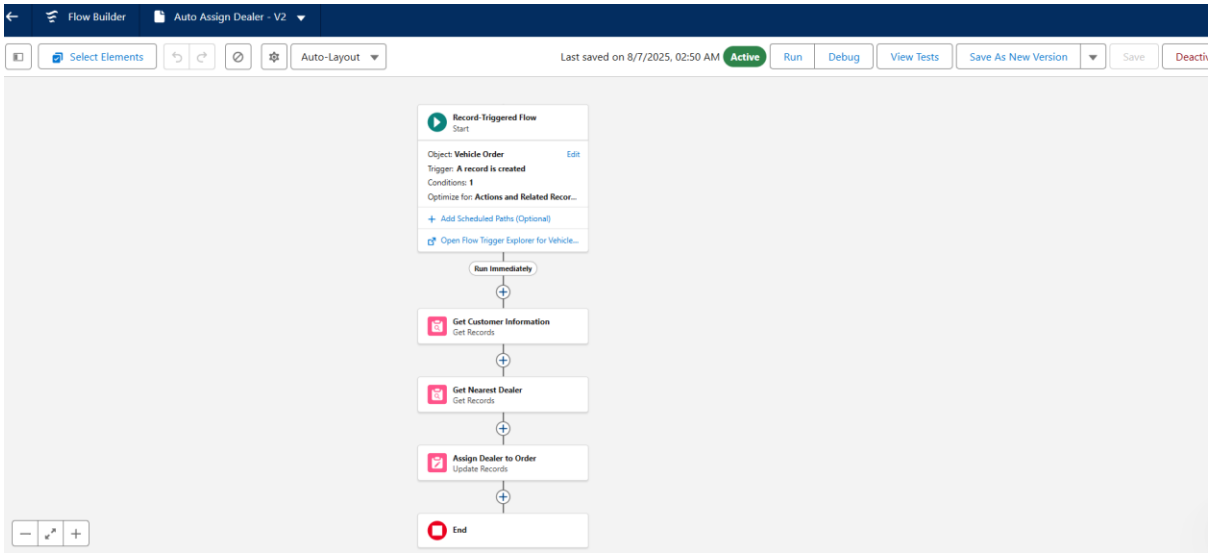
Phase 5: Lightning App Development

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



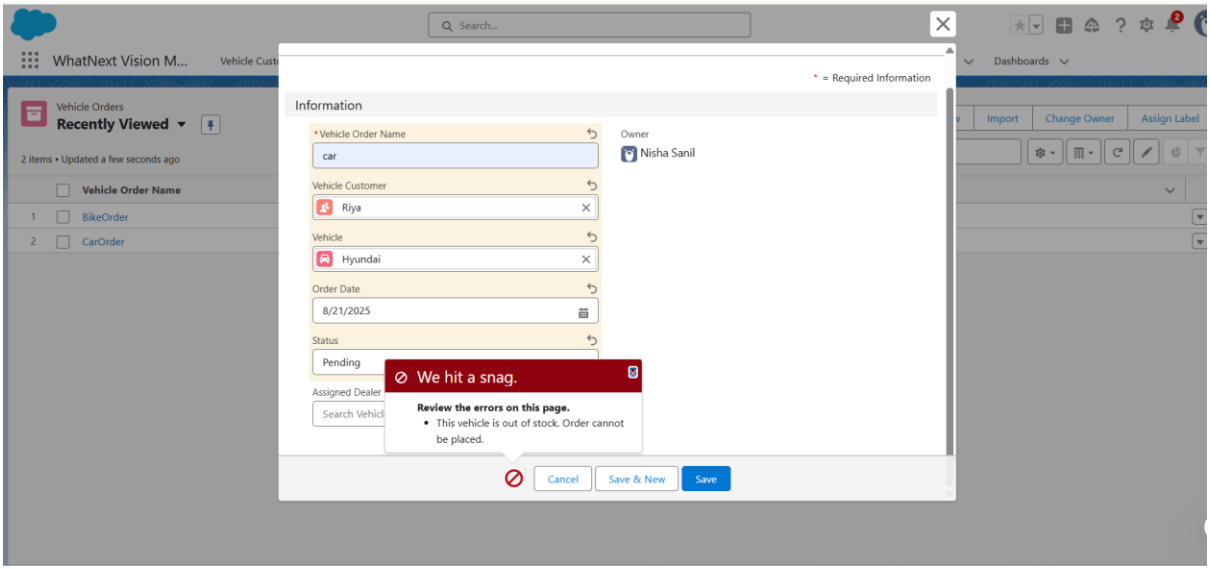
Phase 6: Flow Implementation

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



Phase 7: Apex Trigger and Classes

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



Phase 8: Batch and Scheduled Apex Jobs

- Created a **Batch Apex Class** to scan and update vehicle stock levels regularly.
- Implemented a **Scheduled Apex Job** to run the batch every night.
- Used logs and debug tools to verify batch execution.

Phase 9: Testing and Debugging

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

REAL-WORLD USE CASE

Imagine a customer named Riya visits the WhatNext Motors website or walks into a dealership to place an order for a car. As soon as her information is captured in Salesforce:

1. **Dealer Assignment:**
    - The system checks her address or pin code.
    - A Flow searches for the nearest dealer that has the requested vehicle in stock.
    - That dealer is automatically assigned to the order.
2. **Stock Validation:**
    - Before the order is confirmed, an Apex trigger checks if the vehicle is in stock.

- If it's out of stock, the system prevents order creation and shows an error message.

3. **Order Status Update:**
    - Once the order is created, a scheduled Apex job checks stock every night.
    - If stock is replenished, the order status is updated from "Pending" to "Confirmed."
4. **Email Reminders:**
    - If Riya books a test drive, an email reminder is sent to her a day before the scheduled drive.
5. **Customer Transparency:**
    - At any point, Riya can contact the dealer and receive accurate status updates on her order.

This replicates how leading vehicle brands like Tata Motors or Hyundai use smart CRM systems to reduce confusion, manage inventory efficiently, and provide great customer service.

LIGHTNING APP CREATION

Created a custom app named **WhatNext Vision Motors**

- Tabs: Vehicles, Vehicle Customers, Vehicle Dealers, Vehicle Orders, Vehicle Service Requests, Vehicle Test Drives
- Utility Bar: Quick access to test drive scheduler and stock monitor
- Components: Charts, Related Lists, Actions for fast operations

EXPECTED OUTCOMES

- Dealers receive relevant orders without manual input.
- Customers never face the issue of ordering out-of-stock vehicles.
- Admins are relieved from repetitive email tasks.
- Accurate vehicle stock data updated daily.

SCREENSHOTS

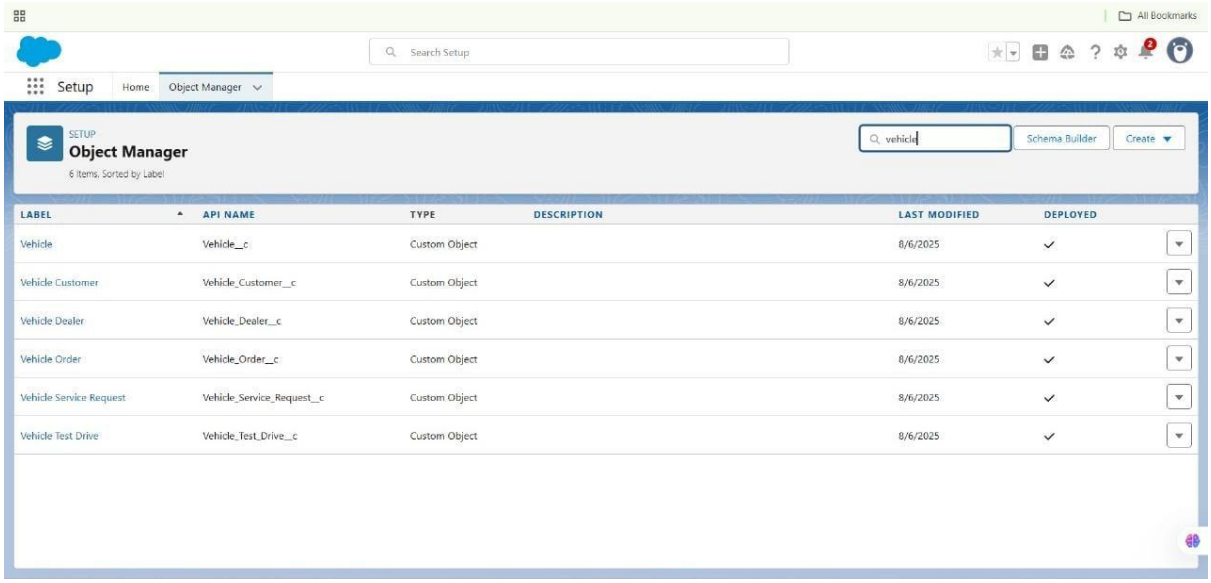


Figure 1:Custom Object Setup

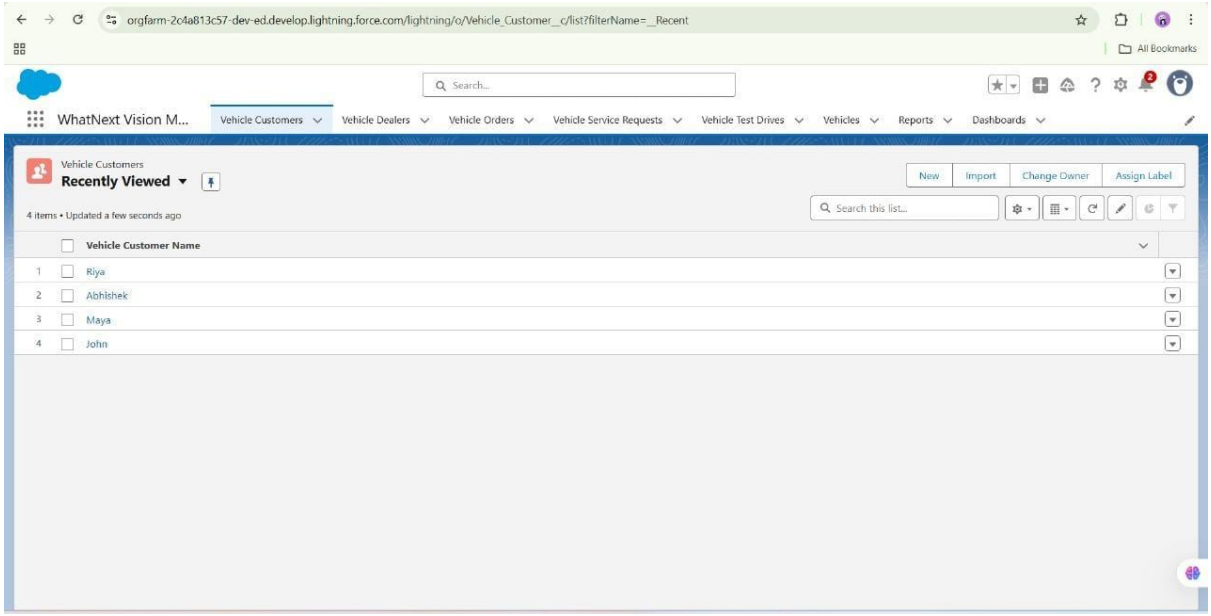


Figure 2: Lightning App Interface with Tabs

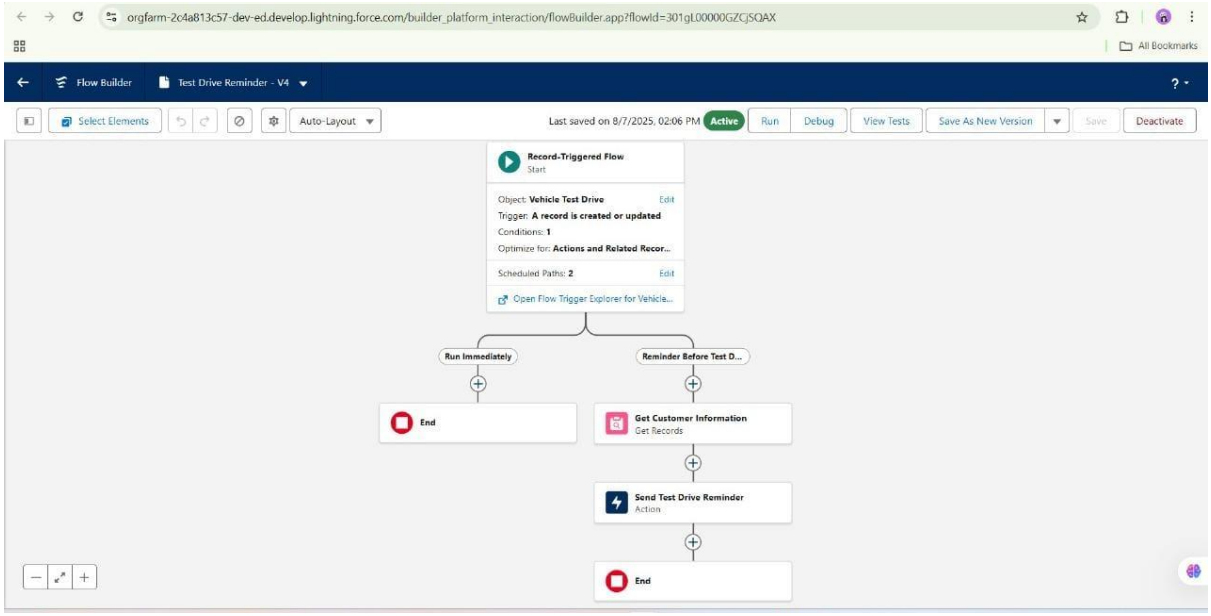


Figure 3: Email Reminder Flow

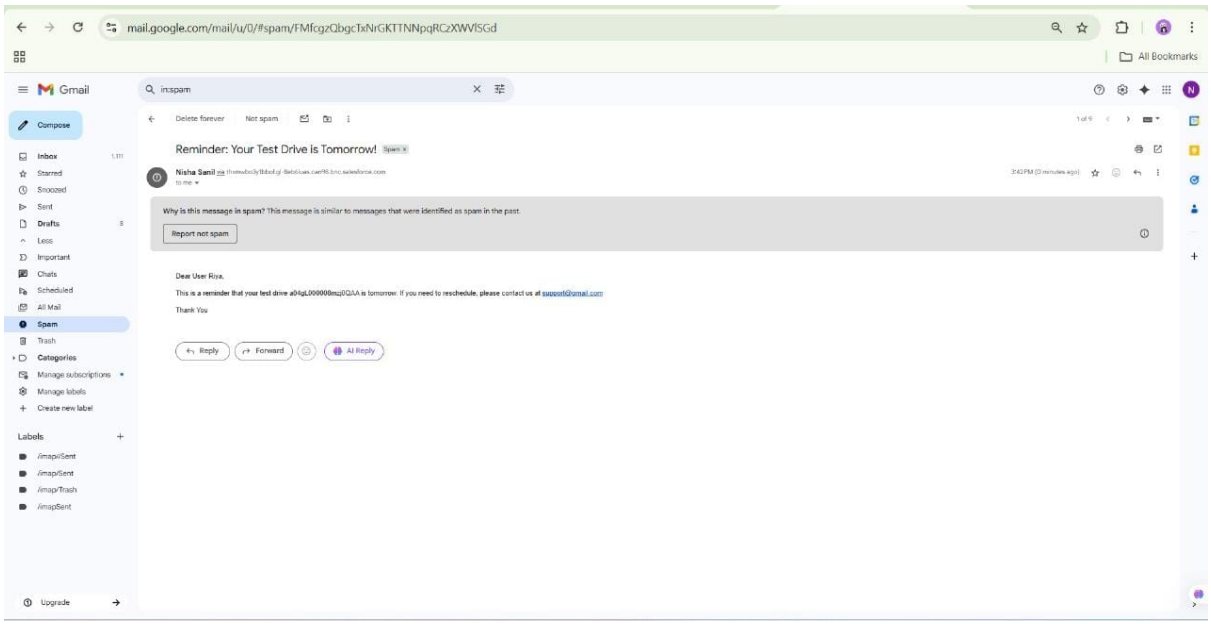


Figure 4:Email confirmation received by the customer for the scheduled test drive.

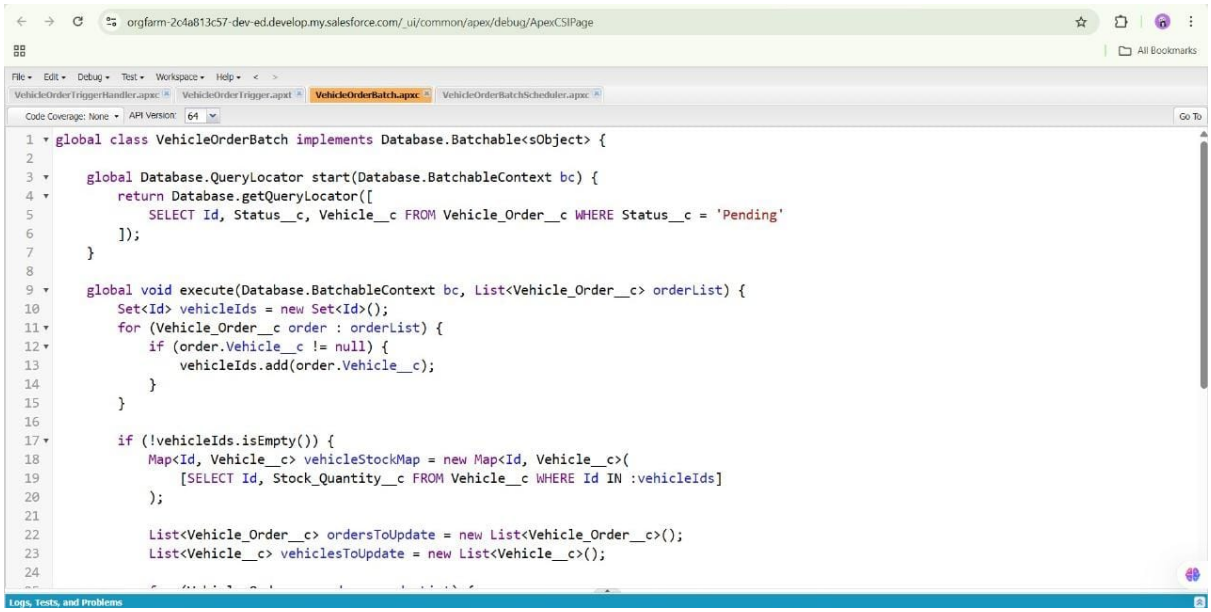


Figure 5: Batch Apex Class

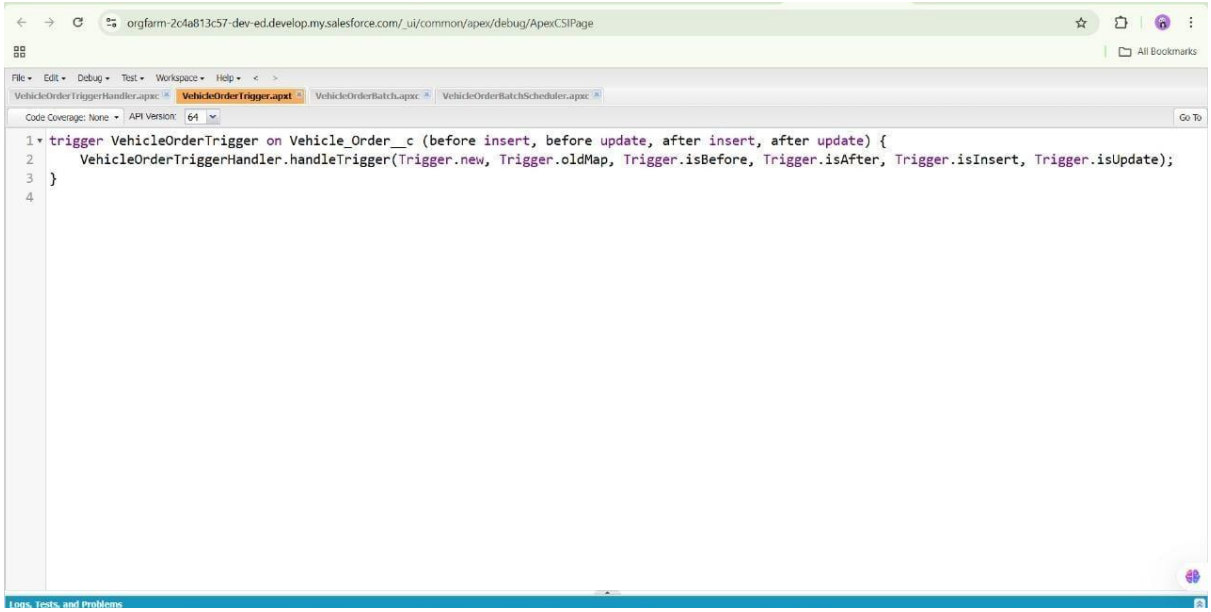


Figure 6:Apex Trigger

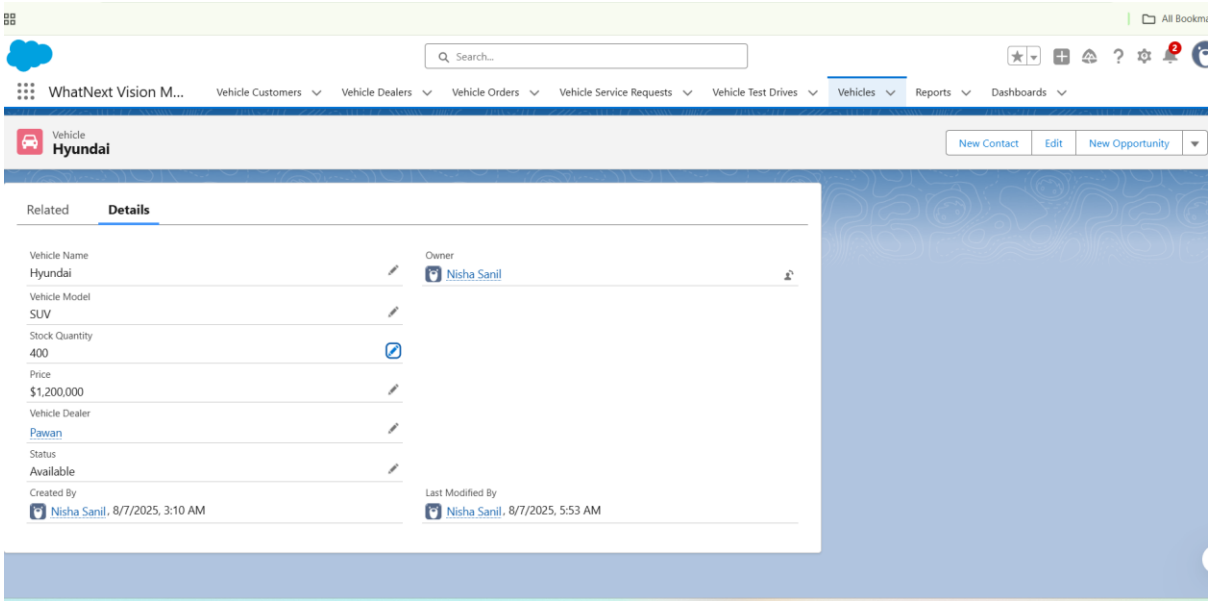


Figure 7:Sample Record

CONCLUSION

This capstone project demonstrates how Salesforce can be leveraged to solve real-world business challenges in the automotive industry using scalable, low-code solutions. By designing a custom Lightning App named **WhatNext Vision Motors**, the system digitalizes the entire vehicle order and service management process—from capturing customer requests to automating dealer assignments and sending timely notifications.

Key modules such as **custom objects**, **flows**, **Apex triggers**, and **batch classes** work together seamlessly to ensure data accuracy, timely communication, and operational efficiency. The implementation of email automation and stock validation not only reduces manual errors but also improves customer satisfaction and dealer coordination.

The use of **Record-Triggered Flows**, **Scheduled Flows**, and **Batch Apex Jobs** ensures the system is proactive rather than reactive, constantly monitoring stock levels and notifying users as needed. Furthermore, modular Apex coding and best practices like the **trigger handler framework** ensure that the system is easy to maintain and scale as business requirements grow.

Overall, the project successfully achieves its goals of automation, efficiency, and user engagement. It acts as a foundation upon which advanced features like AI-based recommendations, mobile integration, and real-time analytics can be added. This solution is a powerful example of how Salesforce can transform traditional dealership operations into a streamlined, intelligent, and customer-centric platform.

FUTURE SCOPE

- **Google Maps API Integration:** Automatically assign the geographically closest dealer using exact coordinates.
- **Mobile App Integration:** Enable customers to place and track orders via a mobile application.
- **AI Stock Prediction Engine:** Use AI to forecast future demand and suggest proactive stock adjustments.
- **Live Chat with Dealer Support:** Enable real-time messaging with the assigned dealer from within the app.
- **Service History Reports:** Generate downloadable service and order history for each vehicle.
- **Advanced Analytics:** Add dashboards that show dealer-wise performance, most demanded vehicle models, and delivery times.