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

Overview

WhatNext Vision Motors is dedicated to modernizing mobility through a strategic Salesforce CRM upgrade designed to enhance both operational efficiency and the customer journey. The core of this initiative involves ensuring that orders are reliable; the system prevents customers from ordering vehicles that are not in stock and automatically routes orders to the dealership closest to the buyer. To support this, background automation manages tasks such as sending polite test drive reminders and updating order statuses without manual interference. These processes are powered technically by Apex triggers for inventory monitoring and scheduled batch jobs that maintain accurate stock levels, ultimately simplifying the purchase process and minimizing business errors.

Core Objectives

The primary goals are to boost customer satisfaction and streamline internal workflows by reducing manual mistakes. Technically, the project aims to deliver:

- Apex triggers to handle stock validation.
- Batch jobs to manage stock updates.
- Scheduled Apex to automate the processing of orders.

Phase 1: Requirement Analysis & Planning

The project strategy was built around four essential requirements:

1. **CRM Implementation:** The system must centralize the management of vehicle details, stock availability, and dealer data, while efficiently tracking service requests and customer test drives. Workflows were required to assign orders based on location.
2. **Process Automation:** Crucial business rules included preventing orders for out-of-stock items, auto-assigning local dealers, and triggering automated email reminders for appointments.
3. **Apex and Triggers:** The team needed to implement triggers to enforce logic like stock validation and ensure code maintainability through modular trigger handlers.
4. **Batch Jobs:** Development of batch Apex was required to periodically audit vehicle stock levels and send notifications regarding replenishment and order processing.

Phase 2: Backend Development & Configuration

This phase focused on constructing the data architecture within Salesforce. The team created specific Custom Objects to manage the data relationships effectively.

- **Custom Objects Created:** The system architecture includes objects for Vehicle, Vehicle Customer, Vehicle Dealer, Vehicle Order, Vehicle Service Request, and Vehicle Test Drive.
- **Tab Configuration:** To make these objects accessible to users, custom tabs were configured for each object (e.g., specific tabs for Vehicles, Customers, and Orders), allowing for organized data management within the Salesforce window.

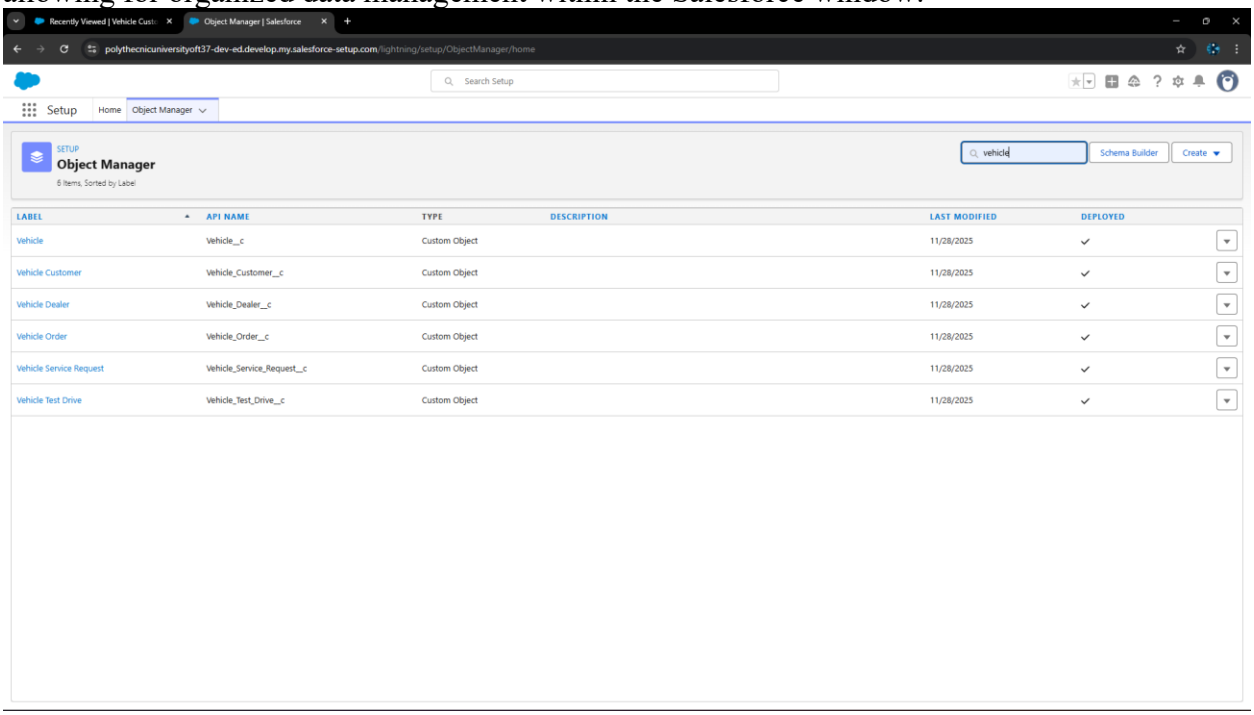

Figure 1. Object Creation

Phase 3: UI/UX Development & Automation

To improve the user experience, a custom "New Lightning App" was built using the App Manager. This allows users to personalize their navigation bar, ensuring that essential items—like Vehicle Customers and Orders—are easily accessible on both desktop and phone interfaces.

- **Field Creation:** Specific data points were defined for the objects. For example, the Vehicle Customer object was configured with fields for Address, Email, Phone, and Customer Name to ensure comprehensive data capture. The Vehicle Order object tracks

the Order Date, Dealer, and Customer.

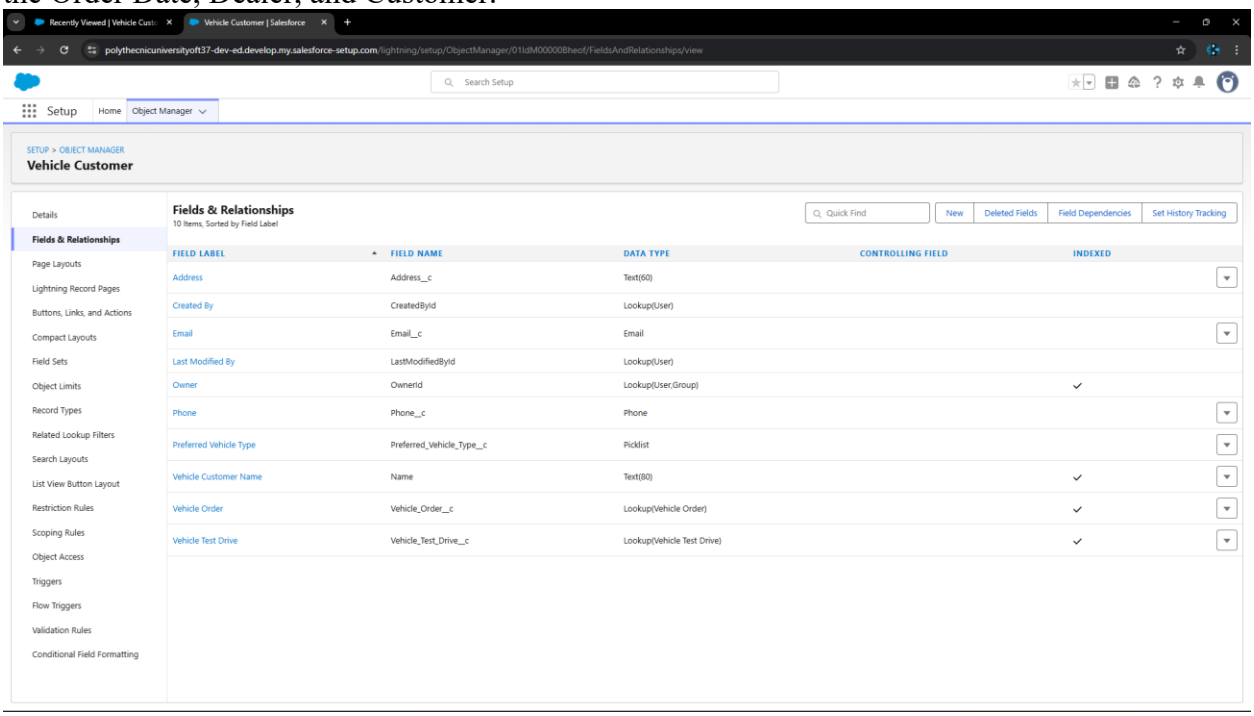

Figure 2. Field Creation1

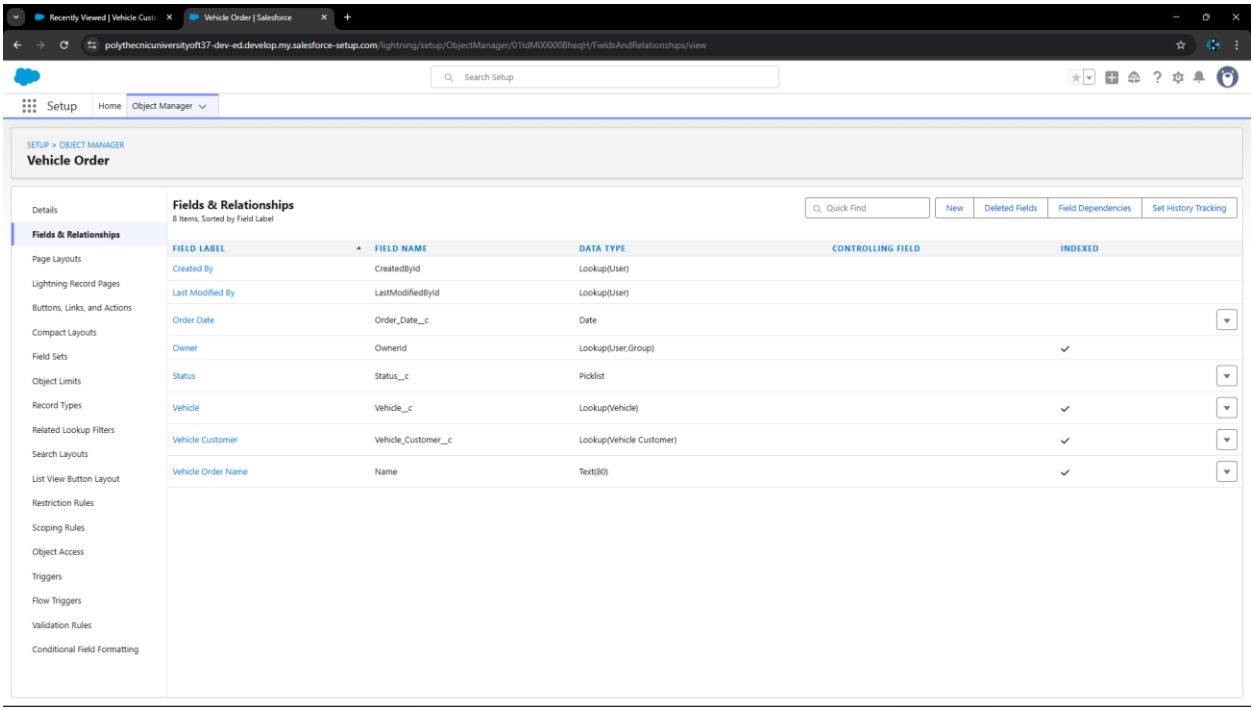

Figure 3. Field Creation2

- **Flow Automation:** Two primary automation flows were built:

1. **Auto Assign Dealer:** A record-triggered flow that gathers customer information, identifies the nearest dealer, and updates the order record to assign that dealer.

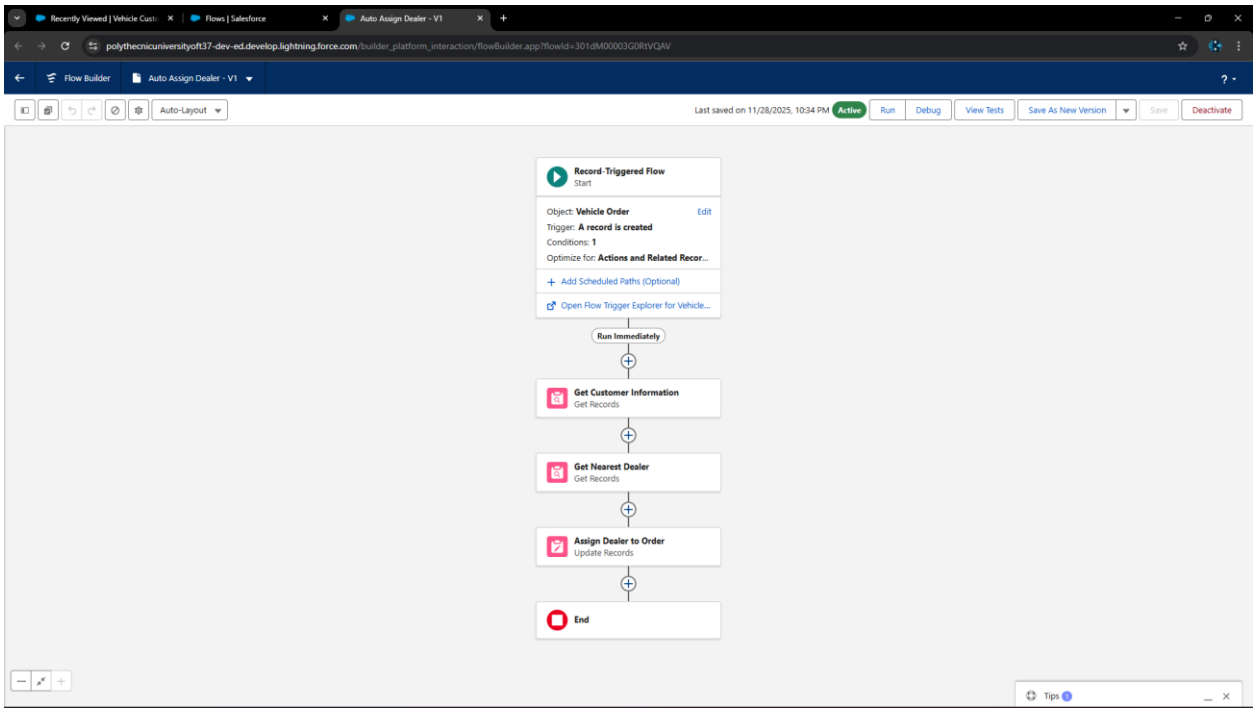

Figure 4. Auto Assign Dealer Flow

2. **Test Drive Reminder:** A flow designed to automatically send email reminders to customers regarding their upcoming test drives.

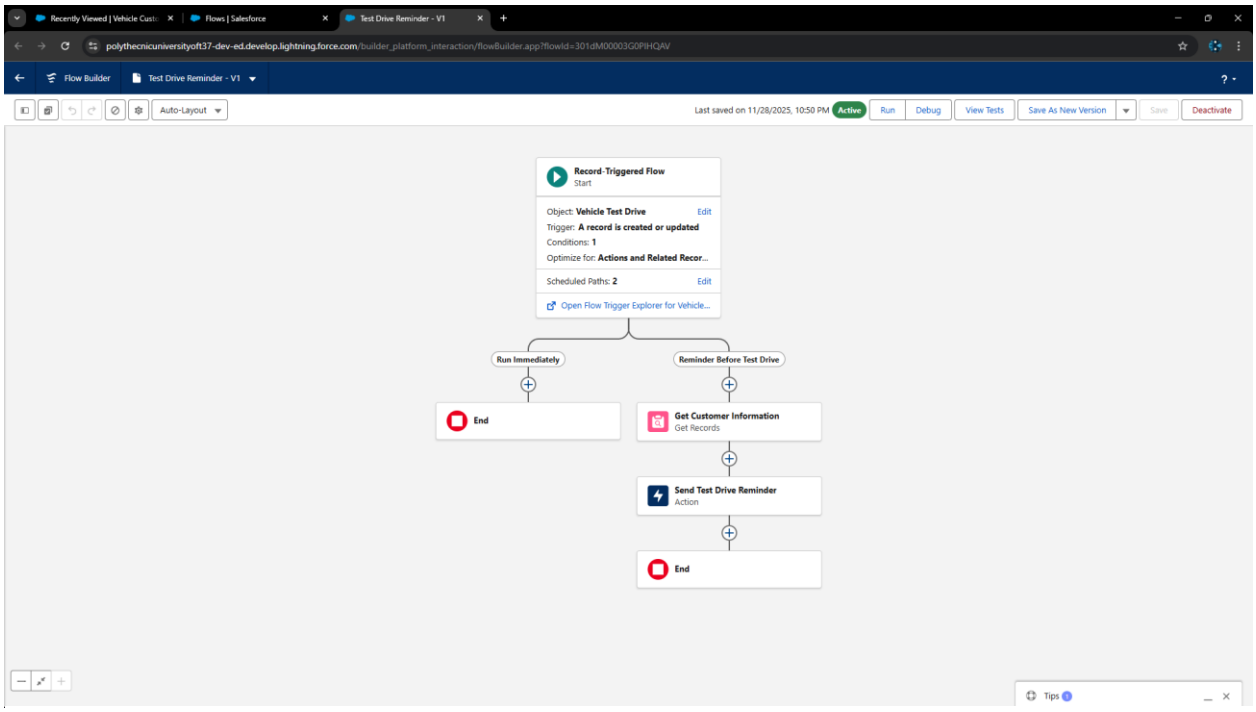

Figure 5. Test Drive Reminder Flow

Phase 4: Data Migration, Testing & Security

This phase involved the heavy lifting regarding code and logic implementation.

- **Apex Development:** Developers utilized the Developer Console to create the VehicleOrderTriggerHandler class. The trigger logic is set to run on Vehicle Order events (before/after insert and update) to enforce business rules.
- **Batch Processing:** A global class named VehicleOrderBatch was implemented to handle high-volume updates.
    - **Logic:** The batch job queries for orders with a "Pending" status. It collects Vehicle IDs and checks the stock map.
    - **Execution:** If the Vehicle__c is not null and the Stock_Quantity__c is greater than zero, the system updates the order status to "Confirmed".
- **Testing:** Test cases were run to verify inputs and outputs, such as checking a Honda Civic order against quantity levels and ensuring email reminders (e.g., "Reminder: Your Test Drive is Tomorrow!") were delivered correctly to the spam/inbox.

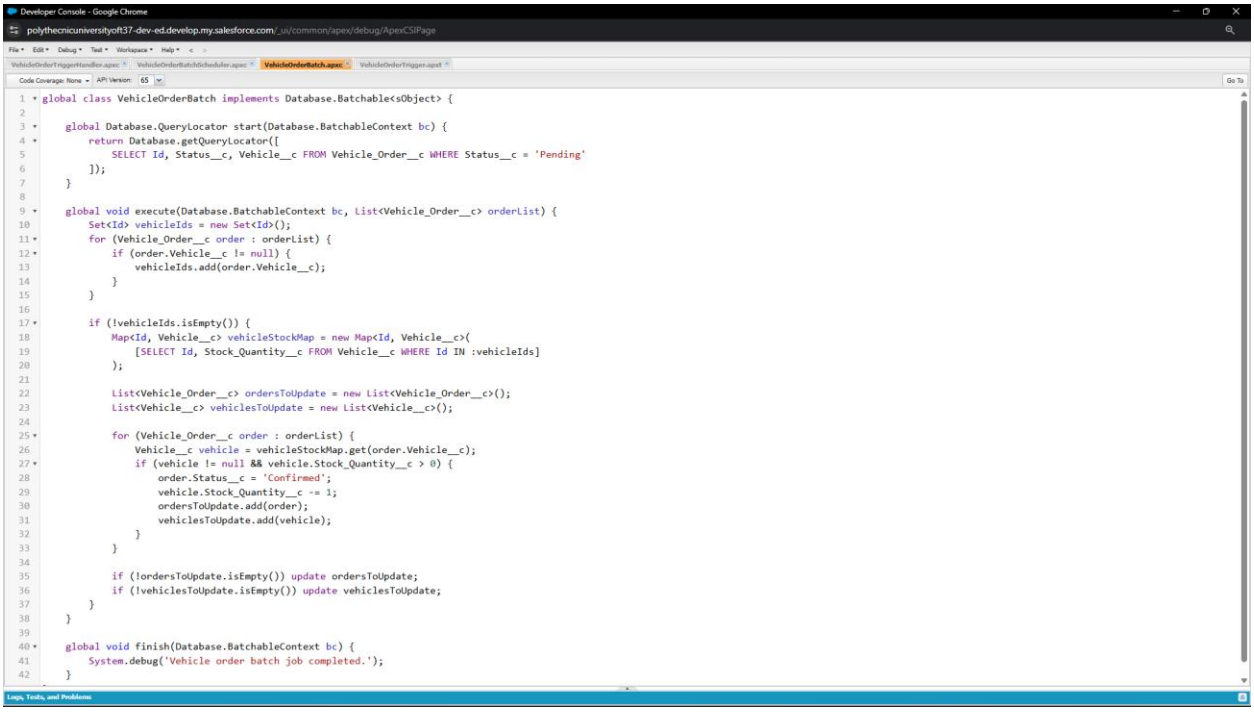

Figure 6. Vehicle Order Batch

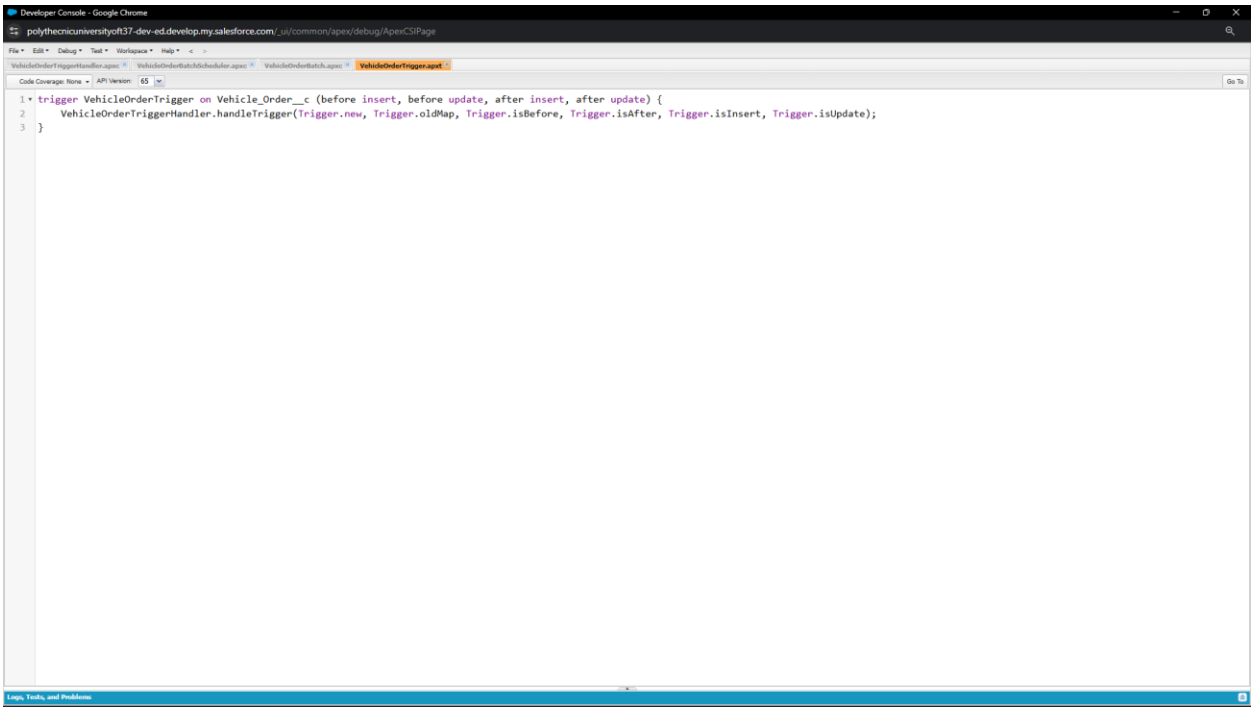

Figure 7. Vehicle Order Trigger

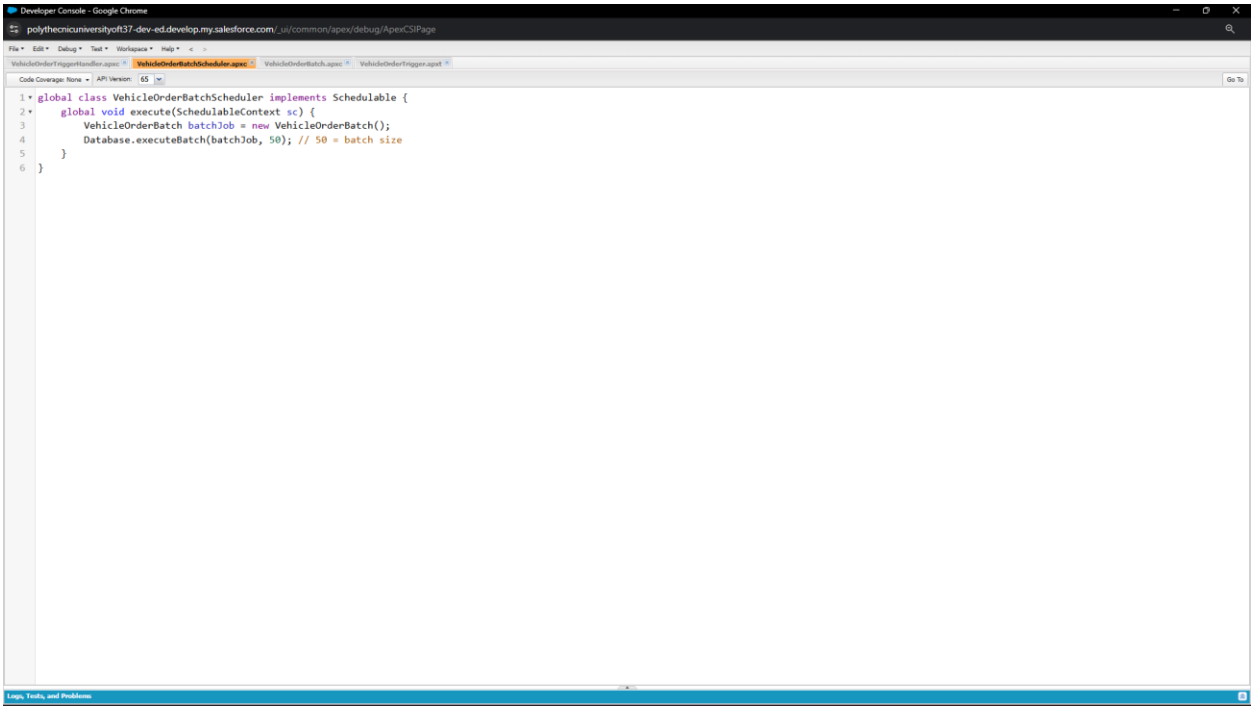

Figure 8. Vehicle Order Batch Scheduler

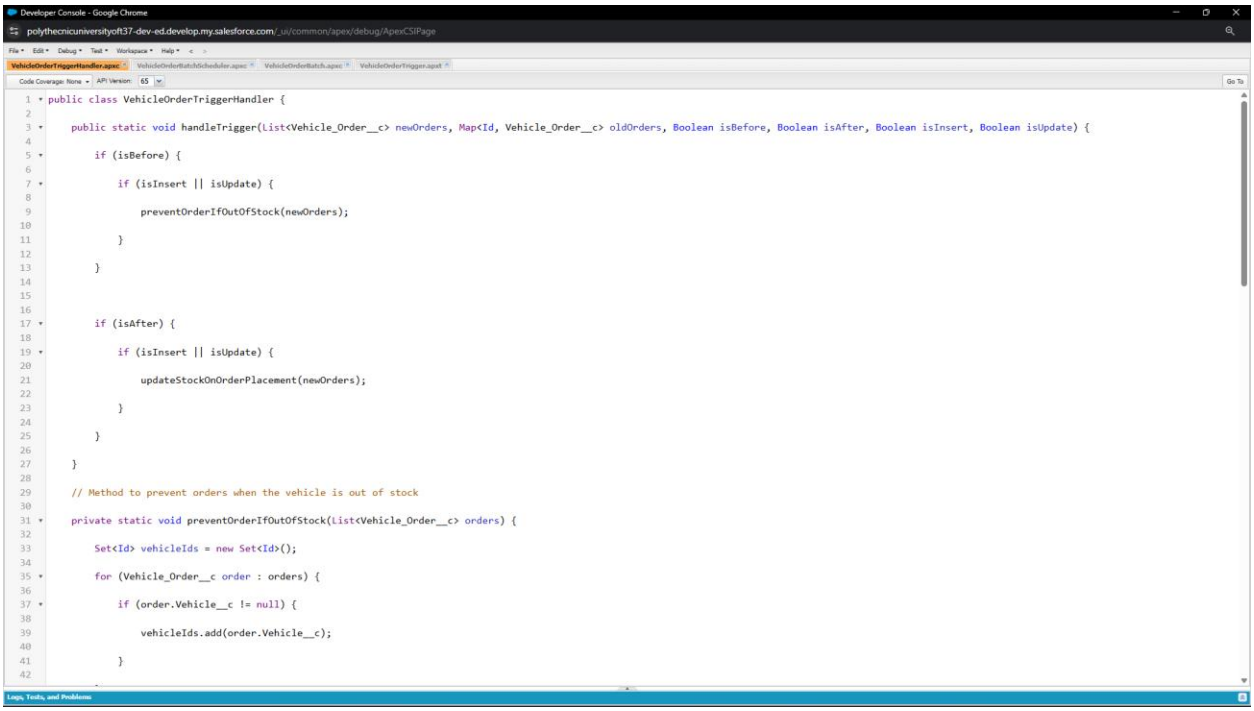

Figure 9. Vehicle Order Trigger Handler

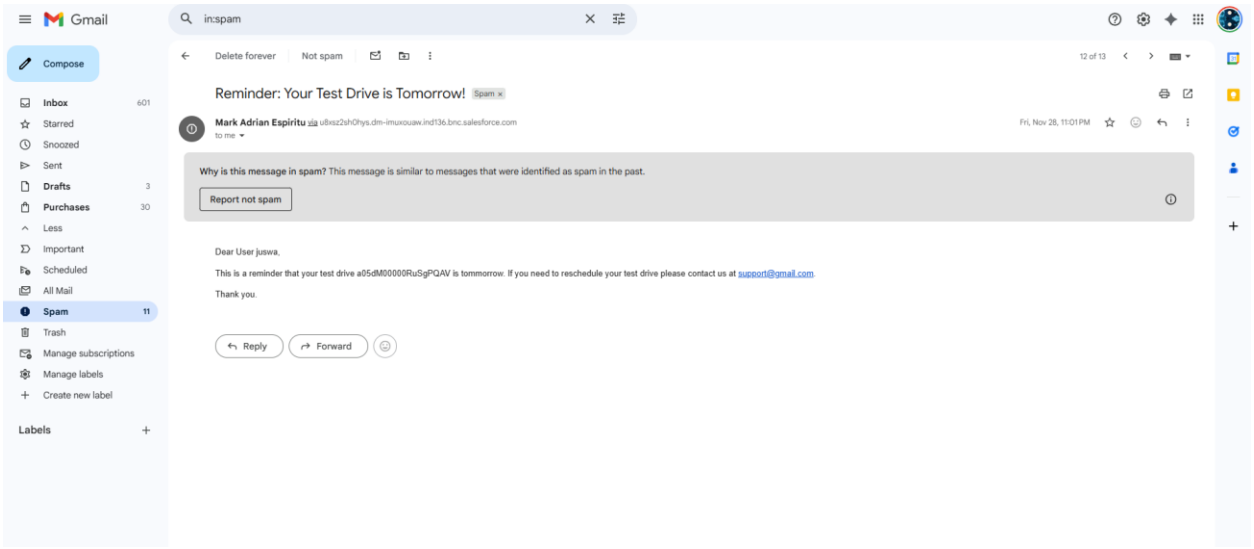

Figure 10. Testing of Test Drive Reminder

Phase 5: Deployment, Documentation & Maintenance

The final phase marked the full deployment of the system, including all custom objects, flows, Apex classes, and dashboards.

- **Validation:** Functional testing was conducted to ensure that automations—specifically dealer assignment and stock validation—executed without errors.
- **Maintenance Strategy:** To ensure continued accuracy and efficiency, the maintenance plan involves regular monitoring of Apex logs and reviewing flow executions.

Conclusion: By integrating real-time stock validation, automated dealer assignment, and proactive communication, WhatNext Vision Motors has established a reliable, error-reduced system that simplifies operations for admins and smooths the purchasing process for customers.