

Final Project Report

BY: Nallamsetty Sri Krishna Manohar

Batch: 1 - TCS LastMile

Mentor: Ms. Mounika Katta

College: Aditya College of Engineering and Technology, Surampalem

Department: CSE (Artificial Intelligence and Machine Learning)

Phase 1: Problem Understanding & Industry Analysis

Introduction

Every successful project begins with a clear understanding of the problem it aims to solve. For AutoFlow CRM, the first step was identifying the pain points in the automotive industry, particularly in how dealerships managed customer interactions, order processing, and communication. Without this analysis, any solution would risk being incomplete or irrelevant to real-world needs.

The automobile industry is highly competitive, where customer satisfaction depends heavily on timely service, transparent communication, and efficient handling of requests. Companies like *DriveNext Motors* were struggling to manage increasing customer demands, leading to inefficiencies that damaged customer trust. Understanding these challenges laid the foundation for creating a solution that would not only automate processes but also transform customer experiences.

Industry Challenges

1. Delayed Order Processing

- **Customers often faced long delays when placing a vehicle order.**
- **Orders were routed manually, sometimes to dealerships that did not have the requested model in stock.**
- **This resulted in frustrated customers and missed sales opportunities.**

2. Lack of Real-Time Updates

- **Customers expected instant updates about their bookings, test drives, or delivery.**
- **Instead, they relied on repeated phone calls or in-person visits, causing dissatisfaction.**

3. Manual Test Drive Coordination

- **Test drives were mostly arranged through phone calls.**
- **This manual process caused confusion, overlapping schedules, and errors in communication.**

4. Inventory Mismanagement

- **Dealerships frequently confirmed bookings without checking live stock availability.**

- This led to situations where customers booked vehicles that were no longer in stock.

5. Customer Dissatisfaction

- In a competitive market, poor communication and delays often pushed customers to choose other brands.
 - A single negative experience significantly reduced customer loyalty.
-

Need for a Centralized System

By analyzing these issues, it became clear that the industry required a centralized digital solution capable of:

- Routing customer orders intelligently to the nearest dealership.
- Validating vehicle availability before booking.
- Automating communication with customers and dealers.
- Providing real-time tracking of orders and test drives.
- Reducing manual work for dealership staff.

Such a system would not only streamline dealership operations but also enhance customer satisfaction through transparency and reliability.

Stakeholder Analysis

- Customers (Buyers): Expect easy booking, instant updates, and smooth communication.
- Dealership Partners: Need efficient inventory management, quick order assignment, and minimal manual coordination.
- Sales Administrators: Require end-to-end visibility of dealership performance, order flow, and customer satisfaction.

Aligning system goals with stakeholder needs ensured better adoption and long-term sustainability of AutoFlow CRM.

Challenges During Analysis

- Collecting accurate data about dealership workflows, since many processes were informal.
- Resistance from staff who were accustomed to manual methods.
- Identifying common pain points shared across multiple dealerships, not just one.

How These Were Overcome:

- Conducting interviews and process observations.

- Benchmarking against CRM solutions in retail and healthcare industries.
-

Solutions Identified

- **Automation of Processes:** Reduce reliance on manual calls for test drives and updates.
- **Geolocation-Based Routing:** Automatically assign orders to the nearest dealership.
- **Stock Validation:** Confirm booking only after checking live vehicle availability.
- **Real-Time Notifications:** Keep customers informed through SMS and email.

Phase 2: Org Setup & Initial Configuration

Introduction

Once the problem areas were identified in Phase 1, the next step was to design the technical foundation for AutoFlow CRM. This involved setting up the Salesforce Organization (Org), configuring user roles, permissions, and business rules. Without a strong initial setup, later phases like automation, integration, and reporting would not function smoothly.

The goal of this phase was to create a robust, scalable, and secure CRM environment tailored to the unique needs of automobile dealerships. This setup ensured that customer requests could be routed effectively, dealership operations were streamlined, and the system could grow with future requirements.

Salesforce Org Setup

AutoFlow CRM was implemented on Salesforce, chosen for its flexibility, automation capabilities, and scalability. The initial configuration involved:

1. Org Creation

A new Salesforce environment was provisioned for DriveNext Motors. This served as the central hub where all customer, dealership, and vehicle data would be managed.

2. Business Hours Configuration

Dealership working hours were configured in the system to ensure that test drives, bookings, and customer communication aligned with real-world operations. This prevented scheduling outside dealership hours.

3. User Roles & Profiles

- **Administrators:** Full control over the CRM system.
- **Dealership Staff:** Managed orders, test drives, and inventory.
- **Sales Executives:** Engaged directly with customers.

- Managers:** Monitored performance through reports and dashboards.

Profiles controlled what each user could access, while role hierarchies determined data visibility.

4. Access Permissions

Proper permissions were set to prevent unauthorized access to sensitive data. For example, dealership staff could view customer data relevant to their branch but not for other branches.

The screenshot shows the Salesforce Setup interface with the following details:

- Left Sidebar:** Shows the navigation tree under "Company Information". The "Company Settings" section is expanded, showing "Business Hours", "Calendar Settings", "Public Calendars and Resources", and "Company Information".
- Page Header:** "Search Setup" and various icons for navigation and help.
- Page Title:** "Company Information" under "SETUP".
- Section Headers:** "Company Information" and "Technical Hub".
- Content Area:**
 - Organization Detail:** Displays organization information like Name (Technical Hub), Primary Contact (Nallamsetty Sri Krishna manohar), Address (IN), Fiscal Year Starts In (January), and various checkboxes for newsletter and admin newsletter.
 - Usage Metrics:** API Requests, Last 24 Hours (20 / 15,000 max); Streaming API Events, Last 24 Hours (0 / 10,000 max); Restricted Logins, Current Month (0 / 0 max).
 - System Info:** Salesforce.com Organization ID (000dL00000cgjOX), Organization Edition (Developer Edition), and Instance (IND154).

Initial Configuration of Core Features

To address the issues identified in Phase 1, the following configurations were implemented:

1. Order Routing Rules

Auto-assignment logic was established so that when a customer submitted a booking request, the system automatically routed it to the nearest dealership based on the customer's geographical location.

2. Vehicle Stock Validation

Before confirming a booking, the system validated the requested vehicle's availability in real time. If the vehicle was unavailable, the booking was restricted.

3. Queues & Assignment Rules

Service requests and customer inquiries were directed to the appropriate team members through assignment rules and queues. This prevented bottlenecks and improved response times.

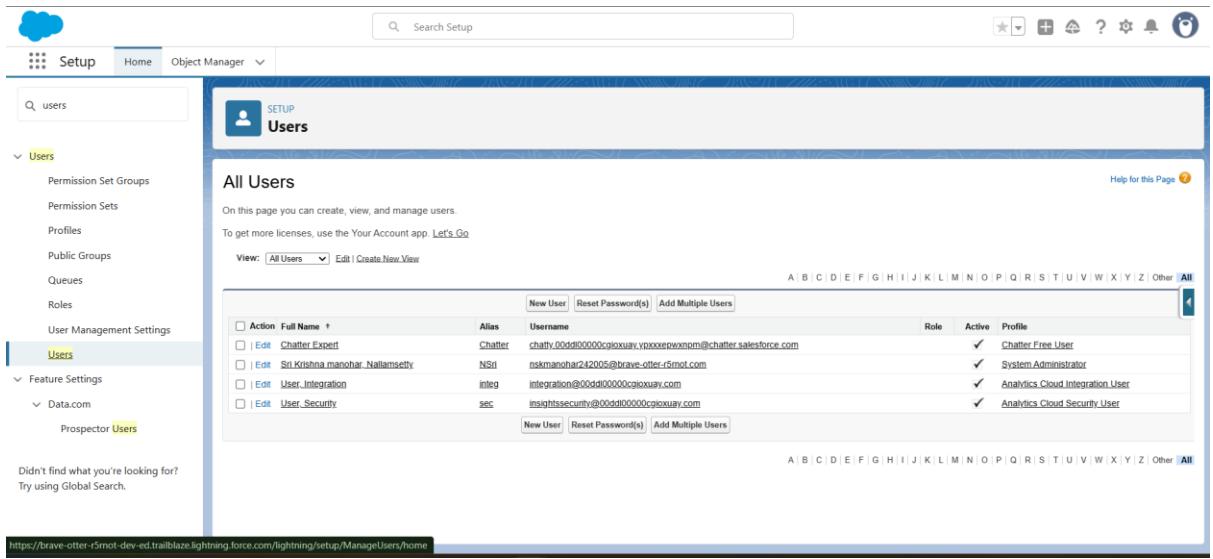
4. Validation Rules

Salesforce's declarative tools were used to enforce rules such as preventing duplicate bookings or restricting order confirmation without available stock.

Challenges Faced During Setup

While setting up the organization, the team encountered several challenges:

- **Data Segmentation:** Dealerships operated in different cities, making it essential to carefully separate data access while still enabling company-wide visibility for administrators.
- **Complex Role Hierarchy:** Multiple layers of users (executives, managers, regional heads) made designing the role structure challenging.



The screenshot shows the Salesforce Setup interface with the 'Users' page selected. The left sidebar shows navigation options like 'Setup', 'Home', and 'Object Manager'. Under 'Users', there are links for 'Permission Set Groups', 'Permission Sets', 'Profiles', 'Public Groups', 'Queues', 'Roles', 'User Management Settings', and 'Users'. The main content area displays a table of users with columns for 'Action', 'Full Name', 'Alias', 'Username', 'Role', 'Active', and 'Profile'. The table includes rows for 'Chatter Expert', 'Sri Krishna manohar_Nallamsetty', 'integ', and 'sec'. Buttons for 'New User', 'Reset Password(s)', and 'Add Multiple Users' are at the top of the table. A URL bar at the bottom shows the address: <https://brave-otter-r5mot-dev-ed.lightning.force.com/lightning/setup/ManageUsers/home>.

- **Customization vs. Standard Features:** Balancing between using Salesforce's out-of-the-box capabilities and customizing features required thoughtful decisions to avoid overengineering.

These challenges were addressed by adopting Salesforce best practices, such as:

- **Setting up role hierarchies**
 - **Leveraging record types**
 - **Ensuring modular design for future scalability**
-

Benefits of Org Setup & Initial Configuration

The results of this phase provided a strong technical foundation for AutoFlow CRM:

- **Efficiency in Order Handling:** Automatic routing reduced manual delays.
- **Data Accuracy:** Stock validation ensured customers could not book unavailable vehicles.
- **Security & Compliance:** Controlled access prevented misuse of customer information.
- **Operational Alignment:** Business hours and rules reflected real dealership practices.

- **Scalability:** The setup ensured the system could handle future expansions, including new dealerships and regions.

Phase 3: Data Modeling & Relationships

Introduction

At the heart of any CRM lies its data model—the blueprint that defines how different entities interact. For AutoFlow CRM, Phase 3 focused on designing a robust, flexible, and logical data model capable of handling the complex ecosystem of buyers, dealerships, vehicles, and orders.

The primary objective was to ensure that all key business entities were represented as Salesforce objects with meaningful relationships between them. A strong data model not only guarantees data integrity but also enables automation, reporting, and integrations to function seamlessly.

Key Objects in AutoFlow CRM

To represent real-world entities, several custom objects were created in addition to standard Salesforce objects. The most critical ones included:

1. Buyers

- Captures customer information (name, contact details, preferences, and location).
- Stores history of interactions such as test drives, bookings, and past purchases.

2. Dealerships

- Stores details about dealer partners, their branches, addresses, and contact staff.
- Maintains operating hours and assigned inventory.

3. Orders

- Represents booking requests, test drives, and confirmed purchases.
- Tracks the order lifecycle: *Requested* → *Booked* → *Confirmed* → *Delivered*.

4. Vehicles

- Holds details of available models, variants, colors, and current stock levels.
 - Linked with dealership inventory for real-time availability.
-

Object Manager

4 Items. Sorted by Label

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Vehicle	Vechile_c	Custom Object		18/09/2025	✓
Vehicle Order	Order_c	Custom Object		18/09/2025	✓
Vehicle service request	Vehicle_service_request_c	Custom Object		18/09/2025	✓
Vehicle Test Drive	Vehicle_Test_Drive_c	Custom Object		18/09/2025	✓

<https://brave-otter-53m0t-dev.lightning.force.com/lightning/setup/ObjectManager/home>

Object Manager

1 Items. Sorted by Label

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Vechile Customer	Vechile_Customer_c	Custom Object		18/09/2025	✓

Object Manager

1 Items. Sorted by Label

LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Dealer	Dealer_c	Custom Object		18/09/2025	✓

Relationships Between Objects

To reflect real-world workflows, relationships were designed using Salesforce lookup and master-detail features:

1. Buyer ↔ Order
 - One buyer can place multiple orders.
 - A master-detail relationship ensured that if a buyer record was deleted, associated orders were also removed to maintain data integrity.
2. Dealership ↔ Vehicle
 - A junction object was introduced to manage many-to-many relationships.
 - This allowed:
 - A single dealership to stock multiple vehicles.
 - The same vehicle model to exist across different dealerships.
3. Order ↔ Dealership
 - A lookup relationship linked each order to the dealership fulfilling it.
4. Technicians ↔ Test Drives
 - Lookup relationships assigned test drive slots to available technicians.
 - Ensured accountability and proper scheduling.

This relational setup ensured that AutoFlow CRM could track customer interactions from start to finish while maintaining accurate dealership and inventory data.

Data Integrity & Validation

To prevent errors and maintain consistency:

- Validation Rules – Restricted duplicate orders or bookings without stock.
- Required Fields – Ensured key details (buyer contact number, dealership location, vehicle model) were mandatory.
- Unique Constraints – Prevented duplicate vehicle entries across dealerships.

These measures ensured clean, reliable, and trustworthy data for decision-making.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Dealer Code	Dealer_Code__c	Auto Number		
Dealer location	Dealer_Location__c	Text(60)		
Dealer Name	Name	Text(80)		
Email	Email__c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone	Phone__c	Phone		

Challenges in Data Modeling

Designing the data model presented several challenges:

- Balancing Complexity** – The automotive ecosystem includes multiple touchpoints (dealers, buyers, technicians). Too many objects could overcomplicate the system, while too few could reduce flexibility.
- Many-to-Many Relationships** – Vehicle–dealership assignment required a custom junction object, which added complexity.
- Scalability** – The model had to be future-proof to accommodate new dealerships, new vehicle types, and advanced features like IoT integrations.

The team resolved these issues by iteratively testing relationships and validating scenarios against real dealership workflows before finalizing the structure.

Phase 4: Process Automation (Admin)

Introduction

After building the core data model in Phase 3, the next priority was to streamline dealership operations by reducing manual workload for staff and sales administrators. Traditionally, dealership employees spent significant time on repetitive tasks such as updating order statuses, reminding customers about test drives, and verifying vehicle stock. These manual processes were prone to errors and consumed time that could otherwise be focused on sales and customer engagement.

To address these challenges, Salesforce's declarative automation tools—including **Flows, Process Builder, and Workflow Rules**—were leveraged. By automating critical processes, **AutoFlow CRM** reduced dependency on manual interventions, resulting in faster service, fewer errors, and improved customer satisfaction.

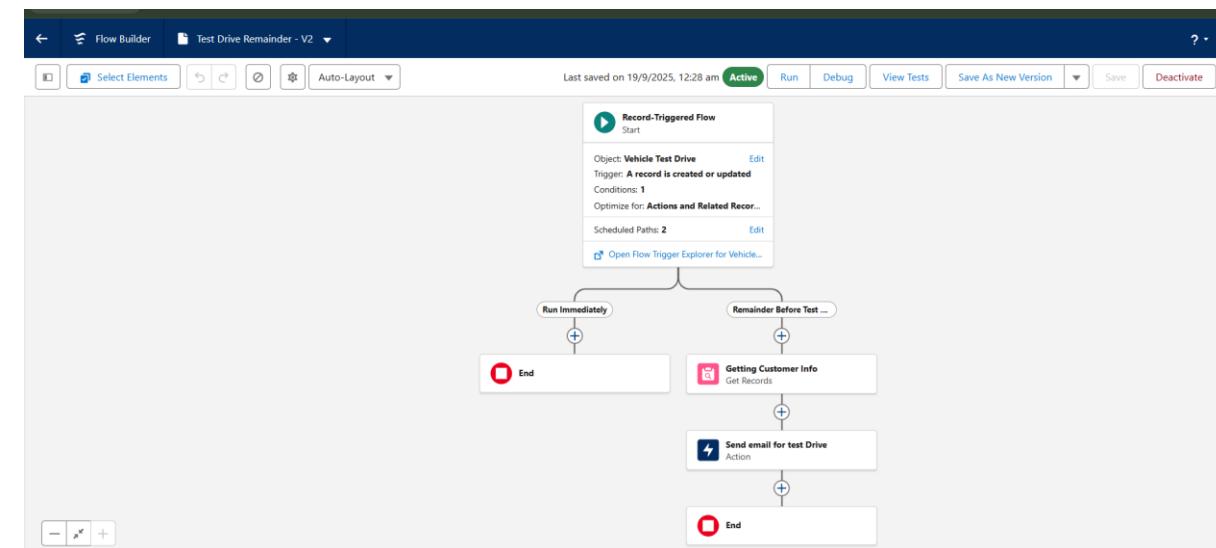
Key Automations Implemented

1. Order Lifecycle Automation

- Orders moved through predefined stages: *Requested* → *Booked* → *Confirmed* → *Delivered*.
- Salesforce Flows automatically updated order statuses as actions were completed (e.g., dealership confirmation, vehicle delivery).
- This eliminated the need for staff to manually update order records.

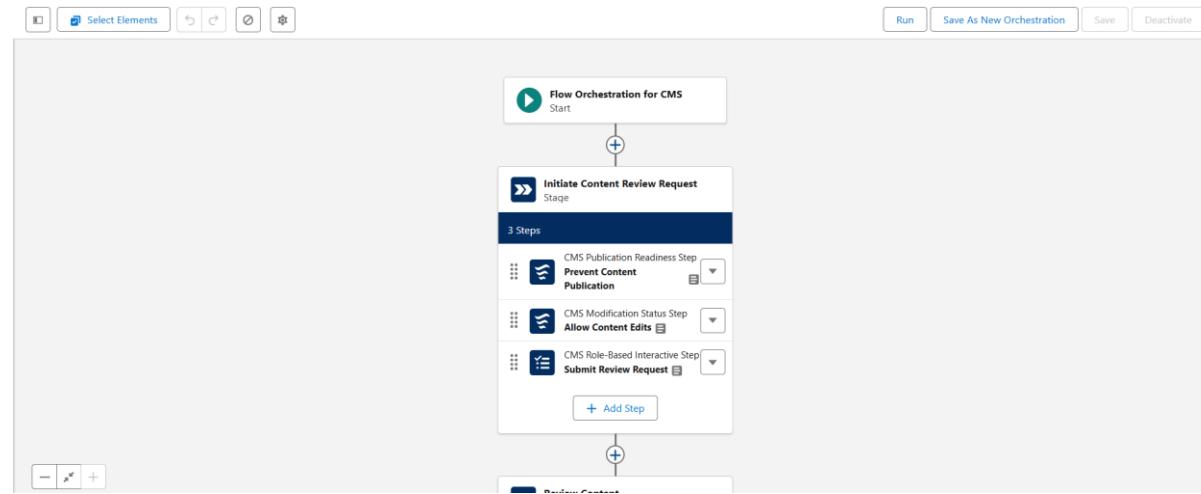
2. Test Drive Scheduling & Notifications

- When a test drive was scheduled, the system triggered automatic SMS and email notifications to both the customer and the assigned technician.
- Reminder notifications were sent before the scheduled time, reducing missed appointments.



3. Stock Validation Rules

- Automations prevented customers from booking vehicles that were out of stock.
- If a requested model was unavailable, the system prompted sales executives to suggest alternative vehicles or variants.



4. Lead Assignment Rules

- New buyer inquiries were automatically routed to the appropriate dealership based on location.
- Within each dealership, leads were fairly distributed among sales executives, ensuring balanced workloads.

5. Follow-Up Tasks

- Workflow rules automatically created follow-up tasks for sales staff after test drives.

Fields & Relationships 9 items, Sorted by Field Label					
	FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Page Layouts	Created By	CreatedById	Lookup(User)		
Lightning Record Pages	Dealer	Dealer_c	Lookup(Dealer)	✓	
Buttons, Links, and Actions	Last Modified By	LastModifiedById	Lookup(User)		
Compact Layouts	Owner	OwnerId	Lookup(User,Group)	✓	
Field Sets	Price	Price_c	Currency(18, 0)		
Object Limits	Status	Status_c	Picklist		
Record Types	Stock Quantity	Stock_Quantity_c	Number(18, 0)		
Related Lookup Filters	Vechile Name	Name	Text(80)	✓	
Search Layouts	Vechile Model	Vechile_Model_c	Picklist		
List View Button Layout					
Restriction Rules					

- This ensured consistent engagement with potential buyers and reduced missed opportunities.

Tools Used

- **Workflow Rules** – For simple tasks like sending notifications and creating follow-up activities.
- **Process Builder** – For multi-step logic, such as updating related records when an order status changed.
- **Salesforce Flows** – For advanced automation, including order lifecycle progression and stock validation across dealerships.

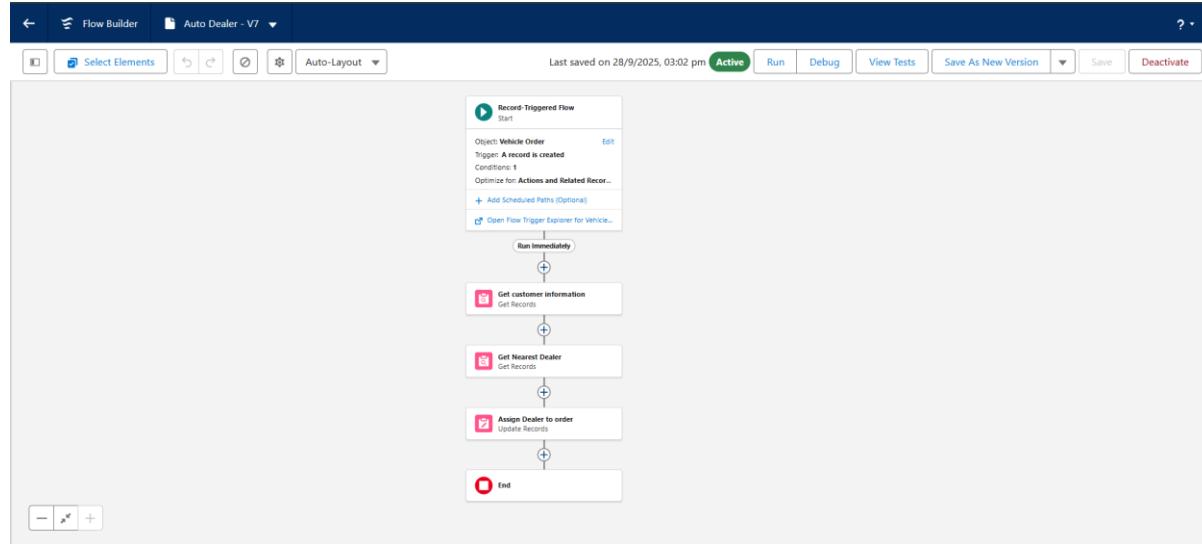
By combining these tools, AutoFlow CRM achieved a balance between simplicity and advanced automation capabilities.

The screenshot shows the Salesforce Setup interface for the Candidate object. The left sidebar lists various setup categories, and the main area displays the 'Candidate Validation Rule' configuration. The rule is named 'Require_Email_or_Phone' and is active. The error condition formula is set to AND(ISBLANK(Email__c), ISBLANK(Phone__c)). The error message is 'Either Email or Phone must be provided'. The rule was created by Tammiseti Venkatarao on 9/22/2025, 10:06 PM, and modified by the same user on the same date and time.

Challenges During Automation

1. Overlapping Logic

- Initially, workflows and processes overlapped, leading to duplicate notifications and conflicting updates.
- This was resolved by consolidating logic into Flows and deactivating redundant rules.



2. Scalability of Flows

- Complex flows with multiple branches risked performance issues as system usage grew.
- The team optimized performance by modularizing flows into smaller, reusable components.

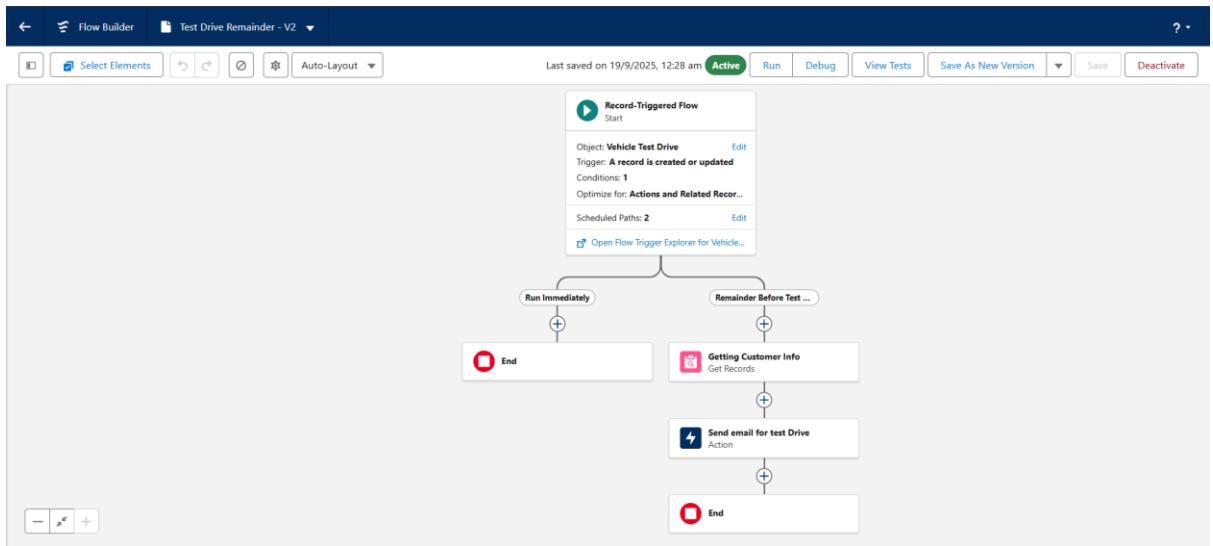
3. User Training

- Dealership staff required training to understand automated processes, especially where the system restricted bookings or enforced validation rules.

Impact of Automation

The automation framework delivered measurable improvements:

- **60% Reduction in Manual Communication** – Customers received automated SMS/email updates without constant staff involvement.
- **Faster Order Processing** – Automated status updates minimized delays in the order lifecycle.
- **Improved Accuracy** – Stock validation ensured bookings were restricted to available vehicles.
- **Enhanced Customer Experience** – Timely updates and reminders kept customers engaged.
- **Higher Staff Productivity** – Employees could dedicate more time to selling and building customer relationships rather than administrative tasks.



Example Scenario

- A customer requests a test drive of a new car.
- The system checks availability of the vehicle at the nearest dealership.
- If available, the test drive is automatically scheduled and notifications are sent to the customer and the technician.
- A follow-up task is created for the sales executive to connect with the customer post-drive.

This seamless workflow eliminated multiple manual calls and confirmations, ensuring both efficiency and a superior customer experience.

Phase 5: Apex Programming (Developer)

Introduction

While Salesforce's declarative tools (Flows, Process Builder, Workflow Rules) addressed most automation requirements in Phase 4, certain complex business needs required a programmatic approach. Declarative tools are powerful but limited when handling **large datasets, advanced conditional logic, and real-time synchronization**.

To overcome these limitations, Apex programming—Salesforce's proprietary object-oriented language—was introduced. Apex provided the flexibility to build triggers, service classes, batch jobs, and scheduled tasks, ensuring AutoFlow CRM could efficiently support **enterprise-scale dealership operations** while remaining scalable for future growth.

Key Apex Implementations

1. Triggers for Inventory Validation

- Apex triggers validated stock availability before confirming an order.
- Example: When a customer attempted to book a vehicle, the trigger cross-checked dealership inventory. If stock was unavailable, the order was restricted and an error message displayed.
- This **real-time validation** eliminated inaccurate bookings and improved customer trust.

2. Service Classes for Business Logic

- To promote **code reusability and maintainability**, complex logic was encapsulated into Apex service classes.
- These classes handled processes such as:
 - Assigning test drives
 - Confirming orders
 - Updating buyer histories

3. Batch Apex for Stock Synchronization

- Dealerships frequently updated vehicle stock levels.
- Batch Apex jobs refreshed inventory data across dealerships at regular intervals, even when **thousands of records** required updates simultaneously.
- This maintained system accuracy at scale.

4. Scheduled Apex Jobs

- Tasks such as sending follow-up reminders, refreshing daily reports, or reassigning pending leads were automated with Scheduled Apex.
- This reduced **administrative overhead** and ensured timely execution of repetitive tasks.

5. Test Coverage & Deployment Readiness

- Apex code in Salesforce requires **75% test coverage** for deployment.
- The development team wrote comprehensive test classes simulating realistic dealership scenarios.
- With **90%+ coverage**, the system became deployment-ready while minimizing production risks.
-
-
-
- `public class VehicleOrderTriggerHandler {`
-
- `// Status constant (better to use Custom Metadata or Custom Settings in real projects)`
- `private static final String STATUS_CONFIRMED = 'Confirmed';`
-
- `public static void handleTrigger(`
- `List<Order__c> newOrders,`
- `Map<Id, Order__c> oldOrders,`
- `Boolean isBefore,`
- `Boolean isAfter,`
- `Boolean isInsert,`
- `Boolean isUpdate`
- `) {`
- `if (isBefore && (isInsert || isUpdate)) {`
- `preventOrderIfOutOfStock(newOrders);`
- `}`
-
- `if (isAfter && (isInsert || isUpdate)) {`
- `updateStockOnOrderPlacement(newOrders, oldOrders);`
- `}`
- `}`
-
- `// ✗ Prevent placing an order if stock is zero or negative`

- private static void preventOrderIfOutOfStock(List<Order__c> orders) {
 - Set<Id> vehicleIds = new Set<Id>();
 - for (Order__c order : orders) {
 - if (Order.Vehicle__c != null) {
 - vehicleIds.add(order.Vehicle__c);
 - }
 - }
 - - if (!vehicleIds.isEmpty()) {
 - Map<Id, Vechcle__c> vehicleStockMap = new Map<Id, Vechcle__c>(
 - [SELECT Id, Stock_Quantity__c FROM Vechcle__c WHERE Id IN :vehicleIds]
 -);
 - - for (Order__c order : orders) {
 - Vechcle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
 - if (vehicle != null && vehicle.Stock_Quantity__c <= 0) {
 - orderaddError('This vehicle is out of stock. Order cannot be placed.');
 - }
 - }
 - }
- - // Decrease stock when an order is confirmed

- private static void updateStockOnOrderPlacement(List<Order__c> newOrders, Map<Id, Order__c> oldOrders) {
- Map<Id, Integer> vehicleOrderCounts = new Map<Id, Integer>();
- - // Collect vehicle order counts (only for new confirmed orders)
 - for (Order__c order : newOrders) {
 - Order__c oldOrder = (oldOrders != null && oldOrders.containsKey(order.Id)) ? oldOrders.get(order.Id) : null;

```

○      Boolean isNewlyConfirmed = order.Status__c == STATUS_CONFIRMED &&
○                  (oldOrder == null || oldOrder.Status__c != STATUS_CONFIRMED);

○
○      if (order.Vehicle__c != null && isNewlyConfirmed) {
○          Integer currentCount = vehicleOrderCounts.containsKey(order.Vehicle__c) ?
○              vehicleOrderCounts.get(order.Vehicle__c) : 0;
○          vehicleOrderCounts.put(order.Vehicle__c, currentCount + 1);
○      }
○
○      if (!vehicleOrderCounts.isEmpty()) {
○          // Lock rows to prevent race conditions
○          Map<Id, Vechcle__c> vehicleStockMap = new Map<Id, Vechcle__c>(
○              [SELECT Id, Stock_Quantity__c FROM Vechcle__c WHERE Id IN
○                  :vehicleOrderCounts.keySet() FOR UPDATE]
○          );
○
○          List<Vechcle__c> vehiclesToUpdate = new List<Vechcle__c>();
○
○          for (Id vehicleId : vehicleOrderCounts.keySet()) {
○              Vechcle__c vehicle = vehicleStockMap.get(vehicleId);
○              Integer orderCount = vehicleOrderCounts.get(vehicleId);
○
○              if (vehicle != null && vehicle.Stock_Quantity__c >= orderCount) {
○                  vehicle.Stock_Quantity__c -= orderCount;
○                  vehiclesToUpdate.add(vehicle);
○              } else if (vehicle != null) {
○                  vehicle.Stock_Quantity__c = 0;
○                  vehiclesToUpdate.add(vehicle);
○              }
○          }
○      }

```

-
- if (!vehiclesToUpdate.isEmpty()) {
- update vehiclesToUpdate;
- }
- }
- }

The screenshot shows the Salesforce IDE interface. The top part displays the Apex code for a trigger handler named VehicleOrderTriggerHandler. The code handles triggers on Order__c objects, checking for specific conditions before preventing an order from being inserted or updated if it's out of stock. The bottom part shows the Logs tab, which lists a single log entry for a user named Nallamsetty Sri Krishna manchar, indicating a successful operation in the browser at 9/29/2025, 2:12:34 PM.

User	Application	Operation	Time	Status	Read	Size
Nallamsetty Sri Krishna manchar	Browser	/aura	9/29/2025, 2:12:34 PM	Success	Unread	314 bytes

Why Apex Was Necessary

Declarative tools alone could not address:

- **Complex conditional logic** – e.g., rerouting an order to another dealership if stock was unavailable locally.
- **Large-scale data processing** – e.g., updating thousands of records simultaneously.
- **Custom scheduling requirements** – beyond what Workflow Rules or Flows could provide.

Apex therefore became the **backbone** of AutoFlow CRM's advanced, large-volume operations.

Challenges in Apex Development

1. Governor Limits

- Salesforce enforces strict limits on database and execution resources.
- Exceeding these (e.g., too many SOQL queries in a loop) caused runtime errors.
- Developers mitigated this by:
 - Using **bulkified queries**

- Avoiding nested loops
- Applying best practices for handling large datasets

2. Error Handling

- With multiple dealerships and complex processes, error scenarios had to be carefully managed.
- Exception handling was built into triggers and batch jobs, ensuring smooth recovery without data loss.

3. Testing Real Scenarios

- Writing meaningful test classes was time-intensive.
- Developers simulated **real dealership workflows** to validate the stability of business logic under diverse conditions.
- The code

```
global class VehicleOrderBatch implements Database.Batchable<sObject> {

    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator([
            SELECT Id, Status__c, Vehicle__c FROM Order__c WHERE Status__c =
            'Pending'
        ]);
    }

    global void execute(Database.BatchableContext bc, List<Order__c> orderList) {
        Set<Id> vehicleIds = new Set<Id>();
        for (Order__c order : orderList) {
            if (order.Vehicle__c != null) {
                vehicleIds.add(order.Vehicle__c);
            }
        }
        if (!vehicleIds.isEmpty()) {
            Map<Id, Vechcle__c> vehicleStockMap = new Map<Id, Vechcle__c>(
                [SELECT Id, Stock_Quantity__c FROM Vechcle__c WHERE Id IN :vehicleIds]
            );
        }
    }
}
```

```

○     List<Order__c> ordersToUpdate = new List<Order__c>();

○     List<Vechcle__c> vehiclesToUpdate = new List<Vechcle__c>();

○

○     for (Order__c order : orderList) {

○         Vechcle__c vehicle = vehicleStockMap.get(order.Vehicle__c);

○         if (vehicle != null && vehicle.Stock_Quantity__c > 0) {

○             order.Status__c = 'Confirmed';

○             vehicle.Stock_Quantity__c -= 1;

○             ordersToUpdate.add(order);

○             vehiclesToUpdate.add(vehicle);

○         }

○     }

○

○     if (!ordersToUpdate.isEmpty()) update ordersToUpdate;

○     if (!vehiclesToUpdate.isEmpty()) update vehiclesToUpdate;

○ }

○ }

○

○     global void finish(Database.BatchableContext bc) {

○         System.debug('Vehicle order batch job completed.');

○     }

○ }

```

```

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
VehicleOrderTriggerHandler.apxc | VehicleOrderBatchApex.s
Code Coverage: None | API Version: 64 | Go To
1 ▪ global class VehicleOrderBatch implements Database.Batchable<sObject> {
2
3 ▪     global Database.QueryLocator start(Database.BatchableContext bc) {
4 ▪         return Database.getQueryLocator([
5 ▪             SELECT Id, Status__c, Vehicle__c FROM Order__c WHERE Status__c = 'Pending'
6 ▪         ]);
7
8 ▪     global void execute(Database.BatchableContext bc, List<Order__c> orderList) {
9 ▪         Set<Id> vehicleIds = new Set<Id>();
10 ▪        for (Order__c order : orderList) {
11 ▪            if (order.Vehicle__c != null) {
12 ▪                vehicleIds.add(order.Vehicle__c);
13 ▪            }
14 ▪        }
15
16 ▪        if (!vehicleIds.isEmpty()) {
17 ▪            Map<Id, Vechcle__c> vehicleStockMap = new Map<Id, Vechcle__c>(

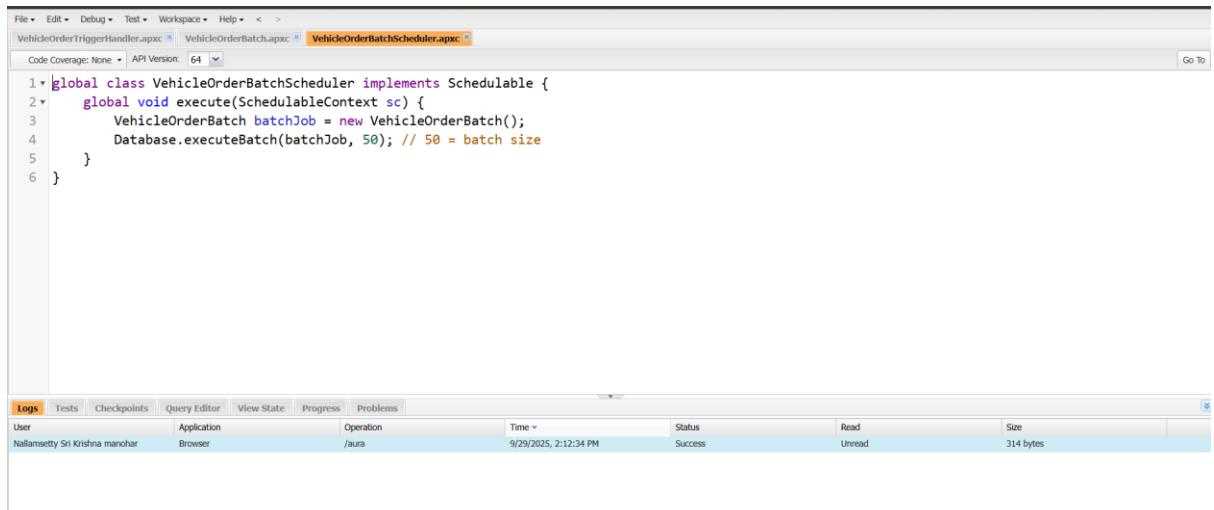
```

Logs	Tests	Checkpoints	Query Editor	View State	Progress	Problems
User Nallamsetty Sri Krishna manohar	Application Browser	Operation /aura	Time 9/29/2025, 2:12:34 PM	Status Success	Read Unread	Size 314 bytes

○

Impact of Apex Programming

- **Scalability** – Batch jobs enabled processing of thousands of vehicles and orders with consistent performance.
- **Accuracy** – Triggers guaranteed no order was confirmed without stock validation.
- **Automation** – Scheduled jobs eliminated repetitive manual tasks, saving dealership staff valuable time.
- **Reliability** – With 90%+ test coverage, the CRM achieved high stability and was ready for enterprise deployment.
- **The code**
- global class VehicleOrderBatchScheduler implements Schedulable {
- global void execute(SchedulableContext sc) {
- VehicleOrderBatch batchJob = new VehicleOrderBatch();
- Database.executeBatch(batchJob, 50); // 50 = batch size
- }
- }



The screenshot shows the Salesforce IDE interface. The top navigation bar includes File, Edit, Debug, Test, Workspace, Help, and a Go To button. Below the navigation bar, there are tabs for VehicleOrderTriggerHandler.apxc, VehicleOrderBatch.apxc, and VehicleOrderBatchScheduler.apxc, with VehicleOrderBatchScheduler.apxc currently selected. The code editor displays the following Apex code:

```
1 * global class VehicleOrderBatchScheduler implements Schedulable {
2 *     global void execute(SchedulableContext sc) {
3 *         VehicleOrderBatch batchJob = new VehicleOrderBatch();
4 *         Database.executeBatch(batchJob, 50); // 50 = batch size
5 *     }
6 }
```

Below the code editor is a logs table with the following data:

User	Application	Operation	Time	Status	Read	Size
Nallamsetty Sri Krishna manohar	Browser	/aura	9/29/2025, 2:12:34 PM	Success	Unread	314 bytes

Phase 6: User Interface Development

Introduction

A system may have powerful logic and automation in the backend, but its success largely depends on how intuitive and user-friendly the interface is for end users. For **AutoFlow CRM**, dealerships, administrators, and customers required a modern, simple, and responsive interface.

Phase 6 focused on designing and building the **Salesforce Lightning Application** with **custom Lightning Web Components (LWCs)** to ensure smooth user interaction. The objective was to create an interface that not only looked modern but also streamlined dealership operations by providing real-time visibility into orders, test drives, inventory, and sales performance.

A screenshot of the Salesforce Lightning App Builder interface. The top navigation bar includes 'Lightning App Builder', 'App Settings', 'Pages', 'HR Recruitment CRM.', and a help icon. On the left, a sidebar shows 'App Settings' with 'Utility Items (Desktop Only)' selected, and sub-options for 'Navigation Items' and 'User Profiles'. The main content area is titled 'Utility Items (Desktop Only)' with the sub-instruction 'Give your users quick access to productivity tools and add background utility items to your app.' It features a 'Recent Items' section containing 'Chatter Feed' and 'Notes', with an 'Add Utility Item' button. To the right, 'Utility Item Properties' are configured for 'Recent Items': 'Label' is 'Recent Items', 'Icon' is 'fallback X', 'Panel Width' is 340, 'Panel Height' is 480, and 'Start automatically' is unchecked. 'Component Properties' show 'Label' as 'Custom' (with options 'Custom' and 'Standard.RecentItems'), 'Objects' as 'API Anomaly Event Store' (with a 'Select...' button), and 'Number of Records to Display' as 3. There are also up/down arrows and a 'Remove' button for reordering items.

Utility Items (Desktop Only)
Give your users quick access to productivity tools and add background utility items to your app.

Add Utility Item

Utility Bar Alignment: Default

Recent Items

Chatter Feed

Notes

Utility Item Properties

Label: Recent Items

Icon: fallback X

Panel Width: 340

Panel Height: 480

Start automatically:

Component Properties

Label: Custom

Custom Label: Standard.RecentItems

Objects: API Anomaly Event Store

Select...

Number of Records to Display: 3

Core Features of the User Interface

1. Real-Time Dashboards

- Dealership managers could instantly view key metrics such as total bookings, test drives completed, vehicles delivered, and customer satisfaction scores.

- Dashboards refreshed automatically, eliminating the need for manual updates.

The screenshot shows the AutoFlow CRM dashboard. On the left, a sidebar titled 'Recent' lists categories like 'DASHBOARDS', 'FOLDERS', and 'FAVORITES'. The main area displays a placeholder message: 'Recent dashboards appear here. Go to All Dashboards to see what's available.' Below this is a 'View All Dashboards' button. The top navigation bar includes links for 'Vehicle Customers', 'Dealers', 'Vehicles', 'Vehicle Orders', 'Vehicle Test Drives', 'Vehicle service requests', 'Reports', and 'Dashboards'.

2. Interactive Order Tracking

- Customers and administrators could monitor the status of any order (Requested → Booked → Confirmed → Delivered).
- Color-coded progress bars and timeline views made order stages clear and visually engaging.
-

3. Test Drive Scheduling Screen

- Staff could view upcoming test drives, assigned technicians, and vehicle availability.
- Customers received confirmations directly within the same interface, reducing dependency on calls or emails.

The screenshot shows the 'Vehicle Test Drives' screen. A sidebar on the left lists 'Recently Viewed' items, including 'test 2', 'test 1', 'test 0', 'm1', 'man2', and 'Man1'. The main area contains a search bar and several action buttons: 'New', 'Import', 'Change Owner', and 'Assign Label'. There is also a 'Search this list...' input field and a set of filter and sort icons. The top navigation bar includes links for 'Vehicle Customers', 'Dealers', 'Vehicles', 'Vehicle Orders', 'Vehicle Test Drives', 'Vehicle service requests', 'Reports', and 'Dashboards'.

4. Vehicle Stock Monitoring

- LWCs displayed real-time dealership stock levels with filters for model, variant, and color.
- Alerts notified managers when specific models were running low, enabling proactive restocking.

The screenshot shows a vehicle detail page in the AutoFlow CRM. The vehicle is a Bugatti, specifically a Sports Cars / Supercars model. It has a stock quantity of 99 and a price of ₹4,00,00,000. The dealer is listed as krishna, and the status is Available. The owner of the vehicle is Nallamsetty Sri Krishna manohar. The page also shows the last modified by user and the date and time of modification.

○

5. Custom Lightning Web Components (LWCs)

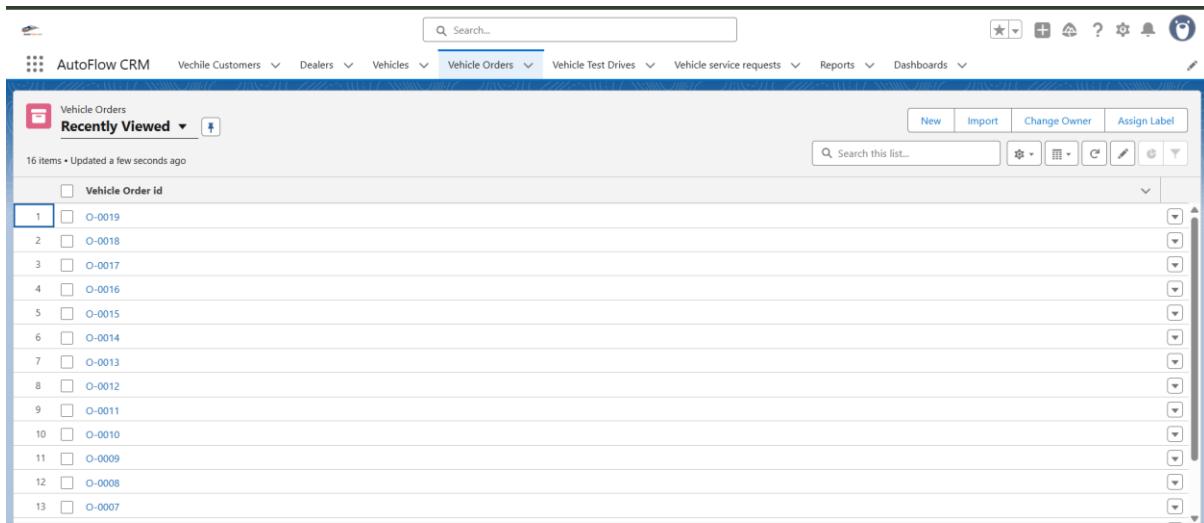
- **Sales Funnel Visualization:** Tracked how leads progressed into bookings and deliveries.
- **Inventory Heatmap:** Highlighted dealerships with the healthiest or lowest stock levels.
- **Performance Widgets:** Quick tiles displayed KPIs like “Orders This Week” or “Pending Deliveries.”

The screenshot shows a list of recently viewed dealers in the AutoFlow CRM. The list includes krishna, Elon Musk, and Stephan Winkelmann. The interface includes standard CRM navigation elements like New, Import, Change Owner, and Assign Label buttons.

○

User Experience (UX) Enhancements

- **Clean Layouts:** Grid-based layouts with minimal clutter ensured intuitive navigation.



- [History](#)
- **Responsive Design:** LWCs were optimized for desktops, tablets, and mobile devices.
- **Role-Based Views:** Administrators, dealership staff, and sales executives each had tailored dashboards aligned to their responsibilities.
- **Search & Filter Options:** Powerful search features enabled quick access to buyers, vehicles, or orders.

Challenges in UI Development

1. Balancing Simplicity with Detail

- Sales teams required comprehensive data, while customers preferred minimal details.
- *Solution:* Role-based dashboards with customized views.

2. Performance of LWCs

- Rendering large datasets (e.g., thousands of vehicle records) initially caused lag.
- *Solution:* Implemented lazy loading and pagination for smoother performance.

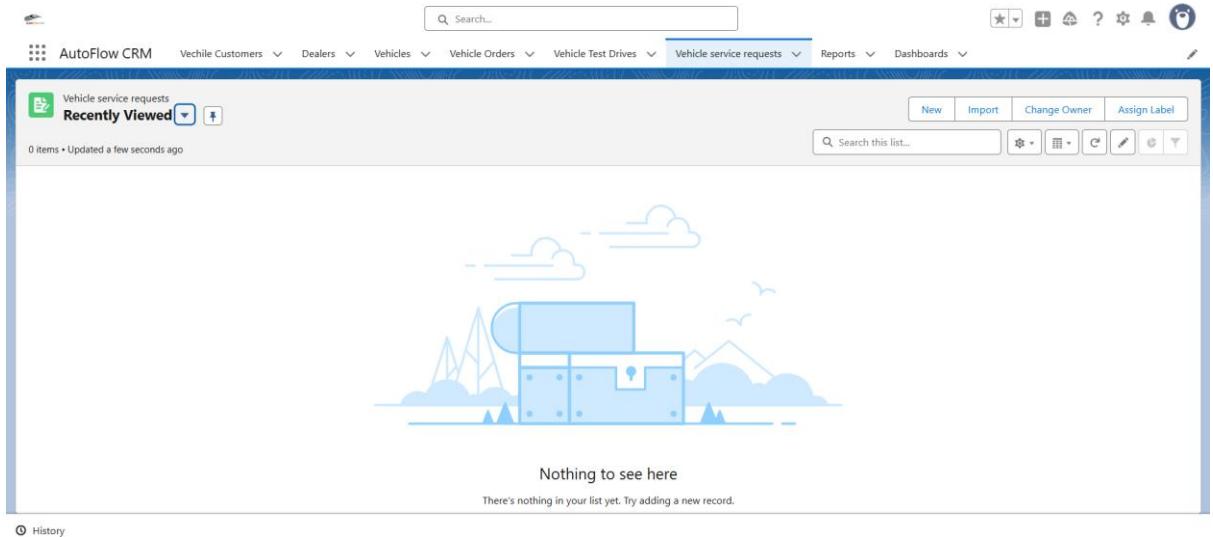
3. Training Dealership Staff

- Some staff were unfamiliar with digital dashboards.
- *Solution:* Conducted training sessions and added contextual help tooltips.

Benefits of a Modern UI

- **Faster Decision-Making:** Managers gained real-time insights for immediate business actions.
- **Customer Transparency:** Buyers tracked orders directly, reducing reliance on dealership calls.
- **Operational Efficiency:** Staff managed test drives, stock, and bookings in one unified app.
- **Data-Driven Insights:** Visualizations enabled quick understanding of dealership performance.

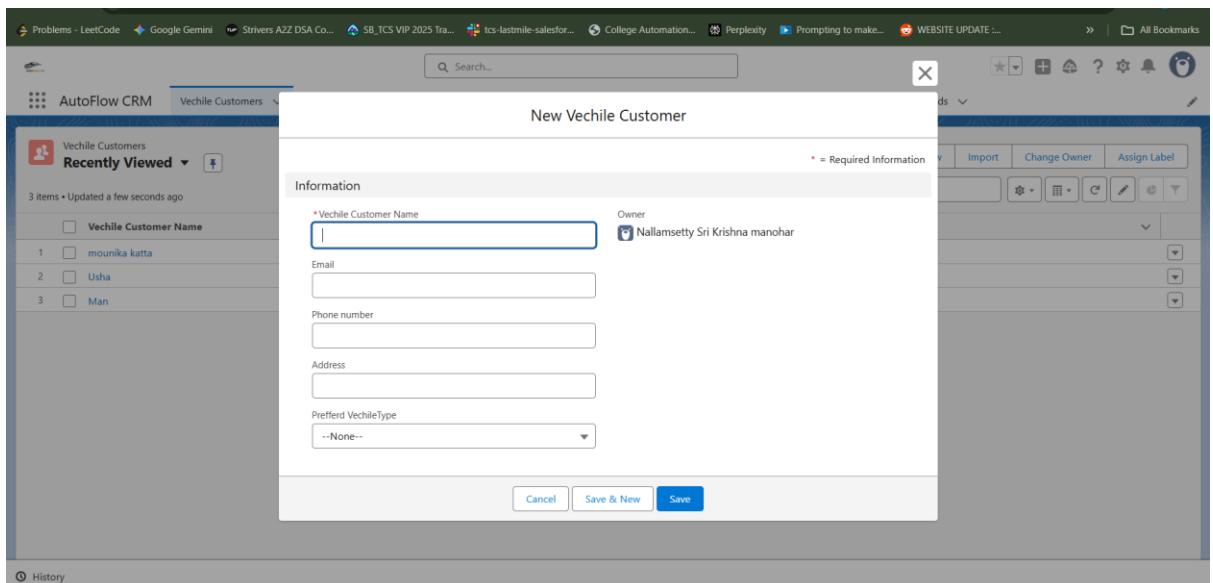
- **Adoption & Satisfaction:** A modern, simple design encouraged higher usage among staff previously resistant to digital tools.



Example Scenario

A customer books a test drive for a sedan model:

- Dealership staff log into the Lightning app.
- The **Test Drive Schedule LWC** instantly displays available slots and assigned technicians.
- Once confirmed, the **Order Tracking Dashboard** updates automatically.
- The customer receives an SMS/email notification in real time.



This streamlined process replaced multiple manual steps, enhancing efficiency and providing both staff and customers with transparent, real-time visibility.

Phase 7: Integration & External Access

Introduction

In today's interconnected business environment, no system functions effectively in isolation. For AutoFlow CRM to achieve its full potential, it required robust integration with external applications, APIs, and dealership systems. This phase focused on enabling seamless communication between AutoFlow CRM and external platforms to support inventory synchronization, customer communication, and real-time data access.

The primary goal of Phase 7 was to establish AutoFlow CRM as a centralized hub that connects dealerships, customers, and administrators across multiple channels, thereby eliminating the need for manual updates.

Core Integrations Implemented

1. Inventory Synchronization with Dealerships

- Integrated AutoFlow CRM with dealership inventory management systems via external APIs.
- Enabled real-time updates of vehicle availability across multiple dealership branches.
- For dealerships lacking automation, Batch Apex jobs were scheduled to pull data periodically.

2. Communication Services

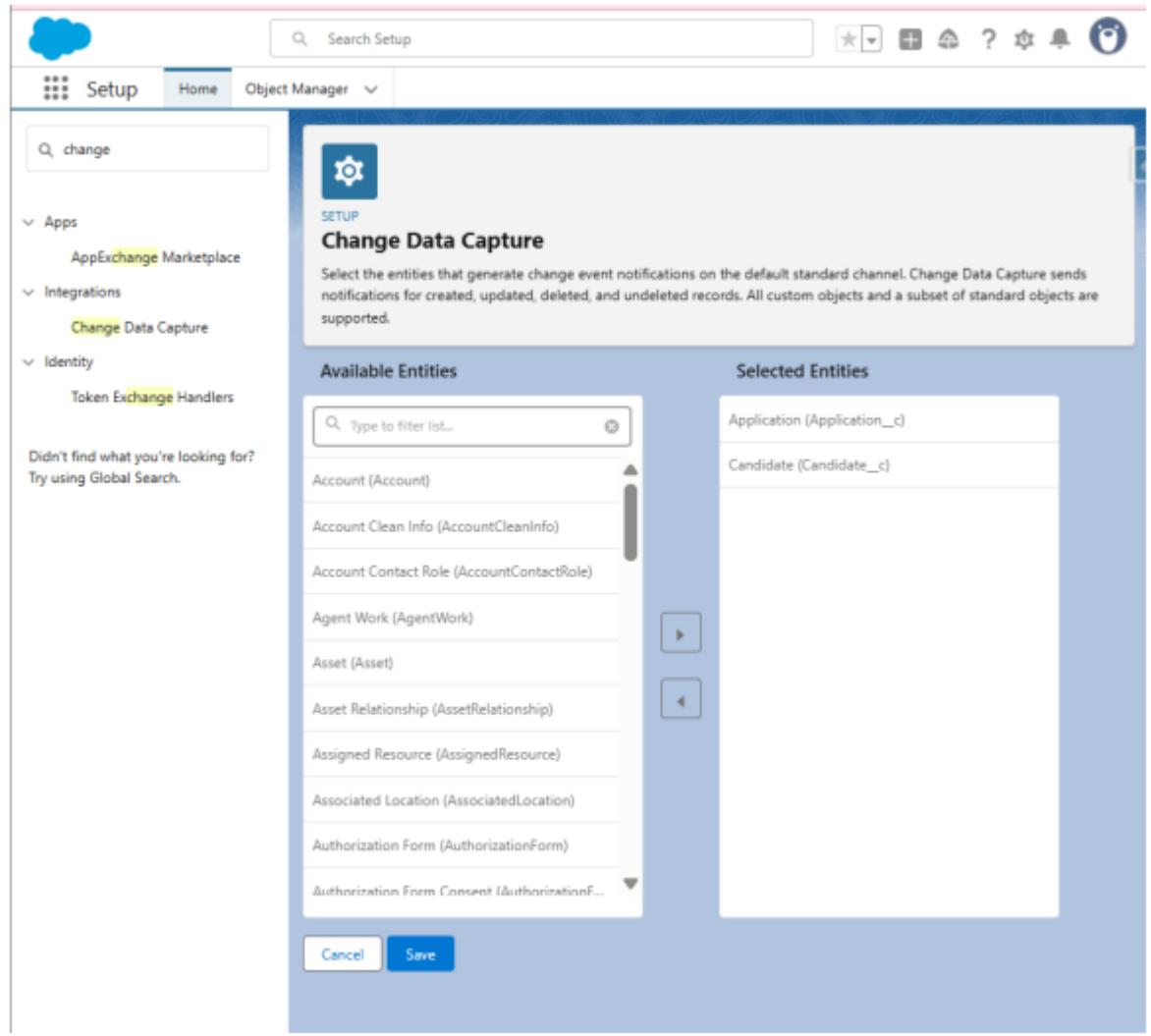
- Integrated with external SMS and email providers to boost customer engagement.
- Automated notifications such as booking confirmations, test drive reminders, and delivery updates were sent without staff intervention.

3. Salesforce Connect for External Data

- Allowed access to external databases without duplicating information inside the CRM.
- Provided administrators with a unified view of dealership operations while keeping external systems intact.

4. Third-Party Services

- Future-ready integration for payment gateways to support booking advance payments.
- Integrated maps and geolocation services to automatically assign orders to the nearest dealership.



External Access for Stakeholders

1. Customer Access

- Customers received automated SMS and email updates without logging into the CRM.
- A future customer portal was planned to allow order tracking, test drive rescheduling, and preference updates.

2. Dealership Staff Access

- Staff accessed the CRM via a Salesforce Lightning App designed for their role.
- Mobile-friendly layouts enabled staff to manage bookings and inventory on the go.

3. Administrators & Managers

- Administrators had full access to dashboards and real-time performance reports.
- Role-based access ensured managers only viewed data relevant to their responsibilities.

Challenges in Integration

1. Data Consistency

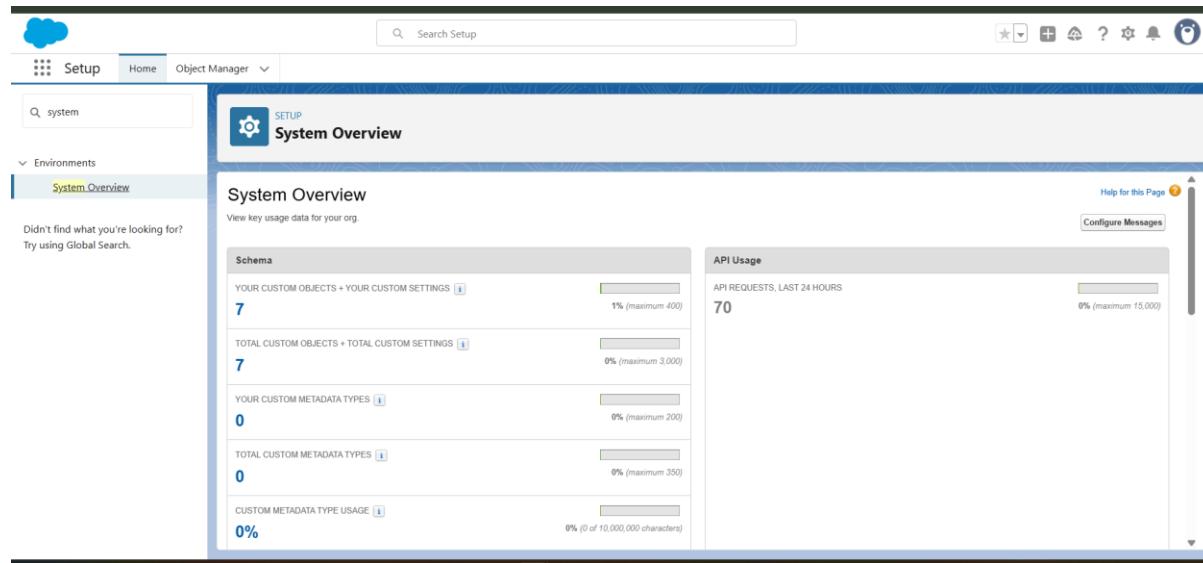
- Risk of duplication and mismatched inventory across dealerships.
- **Solution:** Applied validation rules and scheduled reconciliation jobs.

2. API Rate Limits

- Transaction limits on external APIs occasionally delayed updates.
- **Solution:** Leveraged Batch Apex and scheduled non-critical updates during off-peak hours.

3. Security Concerns

- External access introduced security risks.
- **Solution:** Implemented OAuth 2.0, IP restrictions, and encrypted tokens for secure connections.



Benefits of Integration & External Access

- **Unified Data Management:** Single source of truth for orders and inventory across dealerships.
- **Real-Time Updates:** Customers received instant updates, improving trust and satisfaction.
- **Reduced Manual Effort:** Staff no longer needed to manually update stock or send reminders.
- **Scalability:** Easier to expand with new dealerships, communication tools, or third-party services.
- **Competitive Advantage:** Positioned AutoFlow CRM as a smart, industry-ready solution through interconnected systems.

Example Scenario

A dealership in Hyderabad updates its vehicle stock in their local system:

- The external API automatically pushes the updated inventory into AutoFlow CRM.
- When a customer in Hyderabad books the same model online, the CRM confirms real-time availability.
- The customer instantly receives an automated SMS confirming the booking.
- Simultaneously, dealership staff see the updated booking reflected on their dashboard.

This seamless flow of information was possible only through strong integration and controlled external access.

Phase 8: Data Management & Deployment

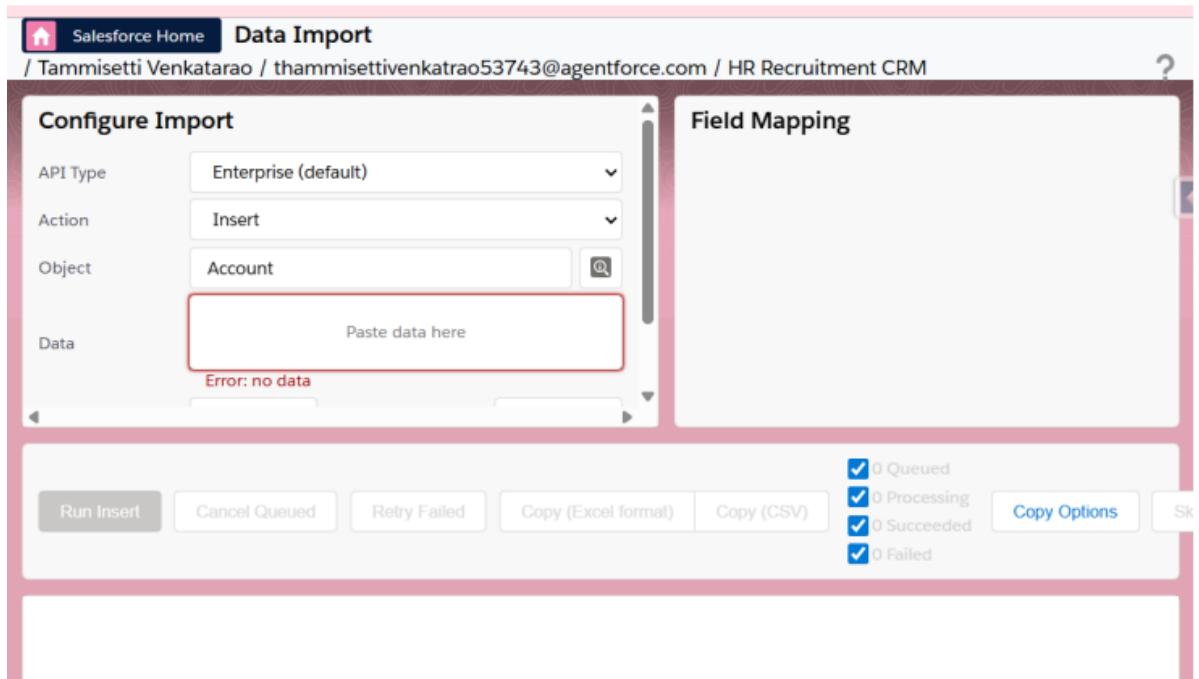
Introduction

In this phase, robust data management and deployment practices were established to ensure the CRM system remained accurate, consistent, and secure. Since dealership, vehicle, and customer data formed the foundation of the AutoFlow CRM, careful attention was given to how this information was imported, validated, and deployed.

Data Management

Salesforce tools such as **Data Loader** and **Import Wizard** were leveraged for uploading data from external sources.

- *Data Loader* handled large volumes of structured data, including dealership details, vehicle inventory, and customer records.
- *Import Wizard* was used for smaller datasets, offering a guided process with simplified field mapping.



To preserve database integrity and prevent redundancies, **duplicate management rules** were configured. These rules automatically flagged or blocked duplicate entries for key objects such as Contacts, Accounts, and Leads. In addition, **validation rules** and **required field settings** were applied to enforce data quality standards, ensuring that every record captured was both complete and consistent.

The screenshot shows the Salesforce Data Export interface. At the top, there are navigation links: 'Salesforce Home', 'Data Export / Tammisetti Venkatarao / thammisettivenkatrao53743@agentforce.com / HR Recruitment CRM', and a help icon. Below this is a toolbar with 'Templates', 'Query History', 'Clear', 'Saved Queries', 'Query Label', 'Save Query', and checkboxes for 'Deleted/Archived Records?' and 'Tooling API?'. The main area contains a query editor with the following code:

```
SELECT Id FROM Interview_c WHERE Application_c = 'a02gL000007SrgUQAS'
```

Below the query are buttons for 'Run Export', 'Export Query', 'Query Plan', and a dropdown menu. The results section is titled 'Export Result' and shows a single record:

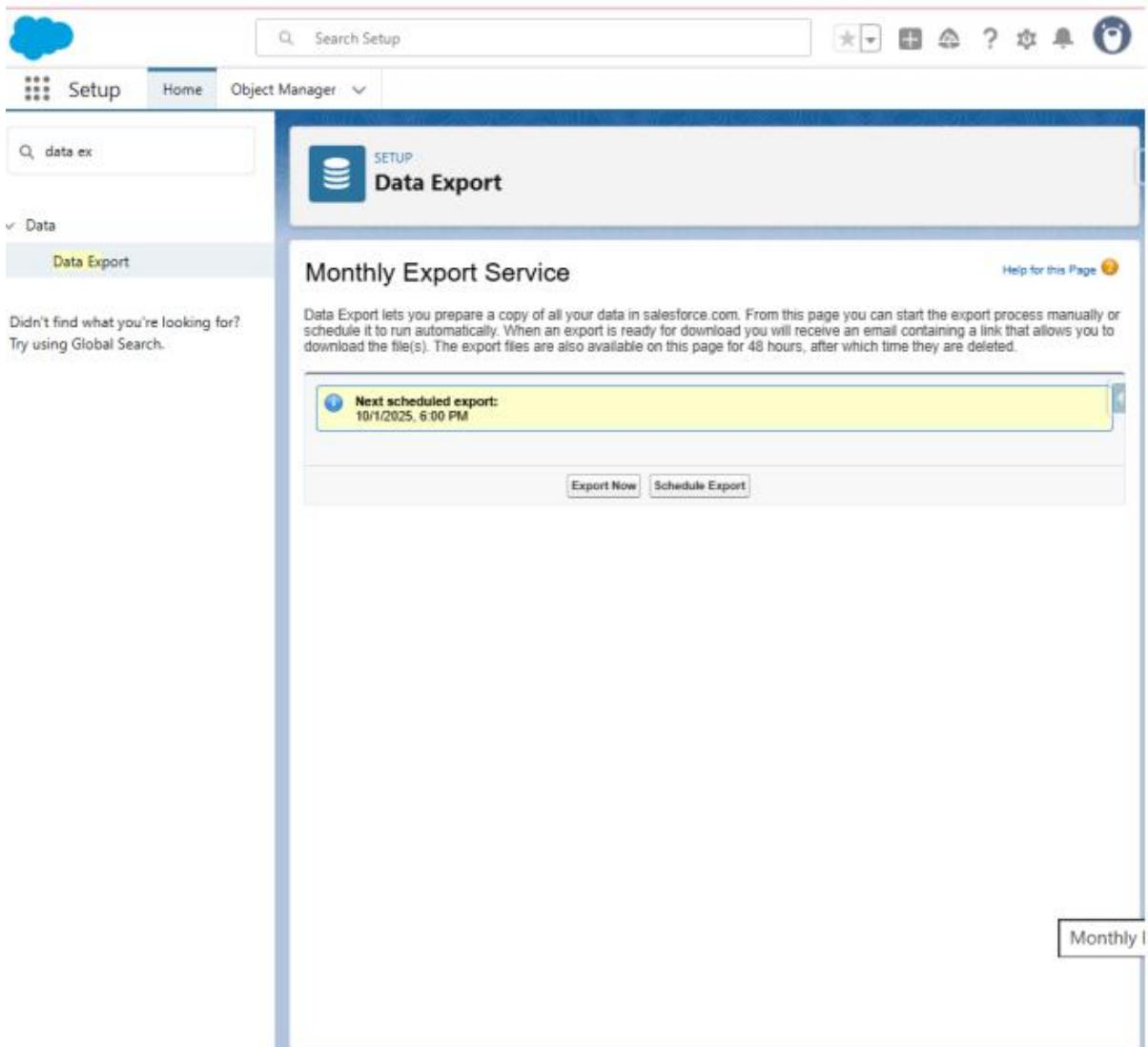
-	Id
Interview_c	a02gL00000D9vZdQAB

Actions available for this record include 'Copy (Excel)', 'Copy (CSV)', 'Copy (JSON)', a download icon, a refresh icon, and a red 'Delete Records' button. There is also a 'Filter Result' search bar and a status message 'Exported 1 record 410.6ms'.

Deployment Management

A structured deployment strategy was followed to migrate changes from the development environment to production:

- **Change Sets** were used for packaging and deploying configuration updates, custom objects, workflows, and automation rules.
- For complex or large-scale updates, **Salesforce DX (SFDX) CLI** was employed to manage version control, push metadata, and automate deployment tasks via the command line.



This hybrid approach enabled smooth, low-risk transitions with minimal downtime, while also reducing the chances of conflicts during production rollout.

Data Security & Backup

To safeguard business-critical information, **scheduled automated backups** were implemented. Regular snapshots of key objects—including Leads, Opportunities, Accounts, and vehicle inventory—were maintained. These backups ensured quick recovery in case of accidental deletions, system errors, or data corruption, thereby preserving business continuity.

Phase 9: Reporting, Dashboards & Security Review

Phase 9 centered on maximizing the CRM's analytics and reporting capabilities to strengthen organizational insights, enable data-driven decision-making, and improve operational efficiency. Reporting and visualization tools were leveraged to track dealership performance, monitor sales trends, and highlight areas for improvement.

Reporting

A variety of report types were created within the CRM to meet diverse analytical needs:

- **Tabular Reports** – Provided simple lists of records, such as open leads, completed sales, and vehicle inventory levels, enabling quick data access.
- **Summary Reports** – Aggregated data by categories such as dealership region, sales representative, or vehicle type, giving a clear view of performance trends.
- **Matrix Reports** – Delivered multidimensional analysis, e.g., sales performance by both month and region, supporting deeper cross-sectional insights.
- **Joined Reports** – Linked data across multiple objects, such as customer satisfaction surveys with completed sales and service records, to provide a holistic view of customer engagement and operational efficiency.

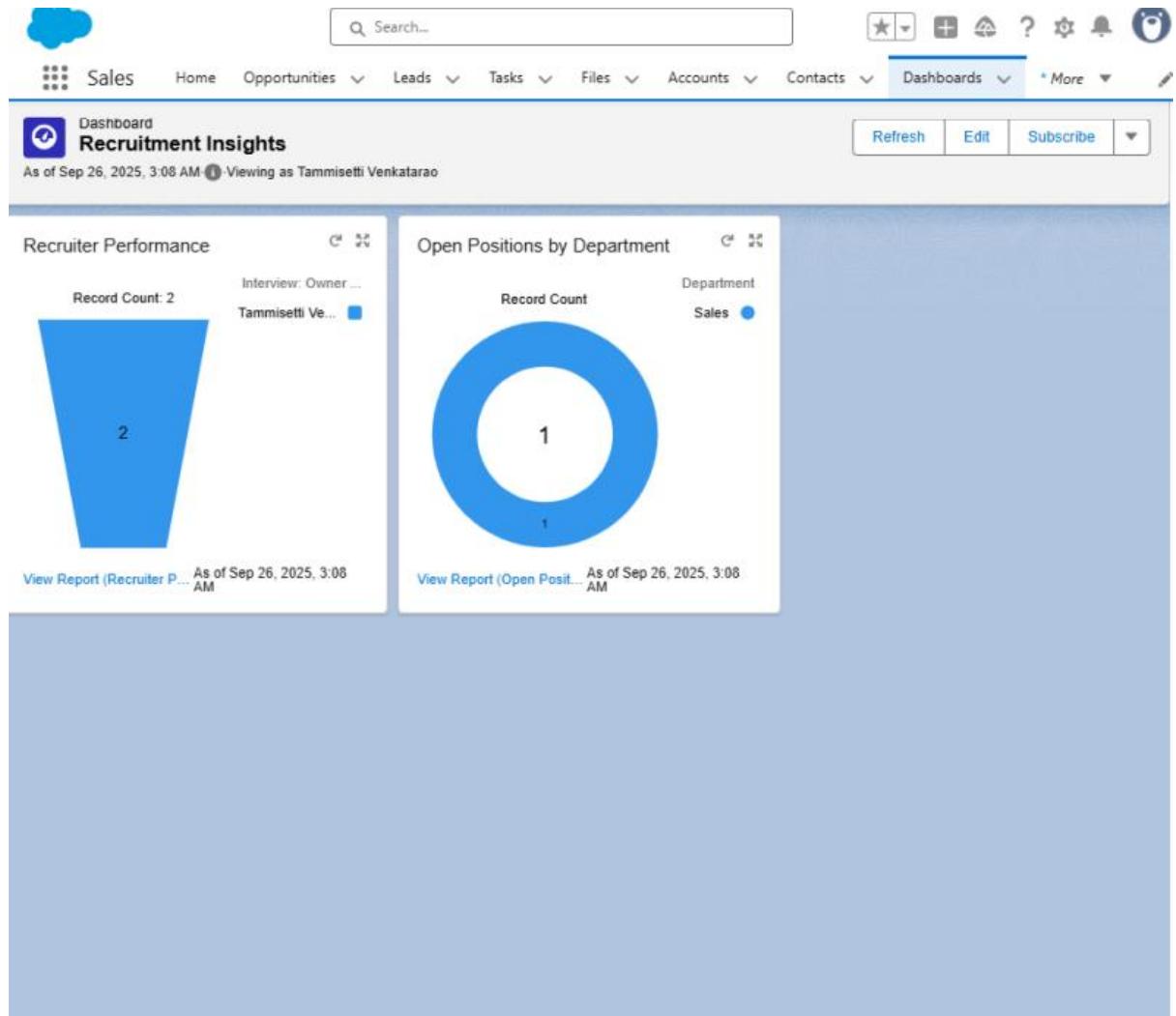
The screenshot shows the 'Reports' section of the AutoFlow CRM. On the left, there's a sidebar with navigation links: 'Reports', 'All Reports' (selected), '6 items', 'RECENT', 'Created by Me', 'Private Reports', 'Public Reports', 'FOLDERS', 'All Folders', 'Created by Me', 'Shared with Me', 'FAVORITES', 'All Favorites'. The main area displays a table of reports with columns: Report Name, Description, Folder, Created By, Created On, and Subscribed. There are six rows of sample reports listed:

Report Name	Description	Folder	Created By	Created On	Subscribed
Sample Flow Report: Screen Flows	Which flows run, what's the status of each interview, and how long do users take to complete the screens?	Public Reports	Automated Process	15/9/2025, 11:13 pm	
Sample Report: Orchestration Run Logs	What orchestration run logs were created and what happened in their associated orchestration runs?	Public Reports	Automated Process	15/9/2025, 11:13 pm	
Sample Report: Orchestration Runs	What orchestration runs have been created and what's the current status of each run?	Public Reports	Automated Process	15/9/2025, 11:13 pm	
Sample Report: Orchestration Stage Runs	What orchestration stage runs have been created and what's the current status of each run?	Public Reports	Automated Process	15/9/2025, 11:13 pm	
Sample Report: Orchestration Step Runs	What orchestration step runs have been created and what's the current status of each run?	Public Reports	Automated Process	15/9/2025, 11:13 pm	
Sample Report: Orchestration Work Items	What orchestration work items were created and what's the current status of each item?	Public Reports	Automated Process	15/9/2025, 11:13 pm	

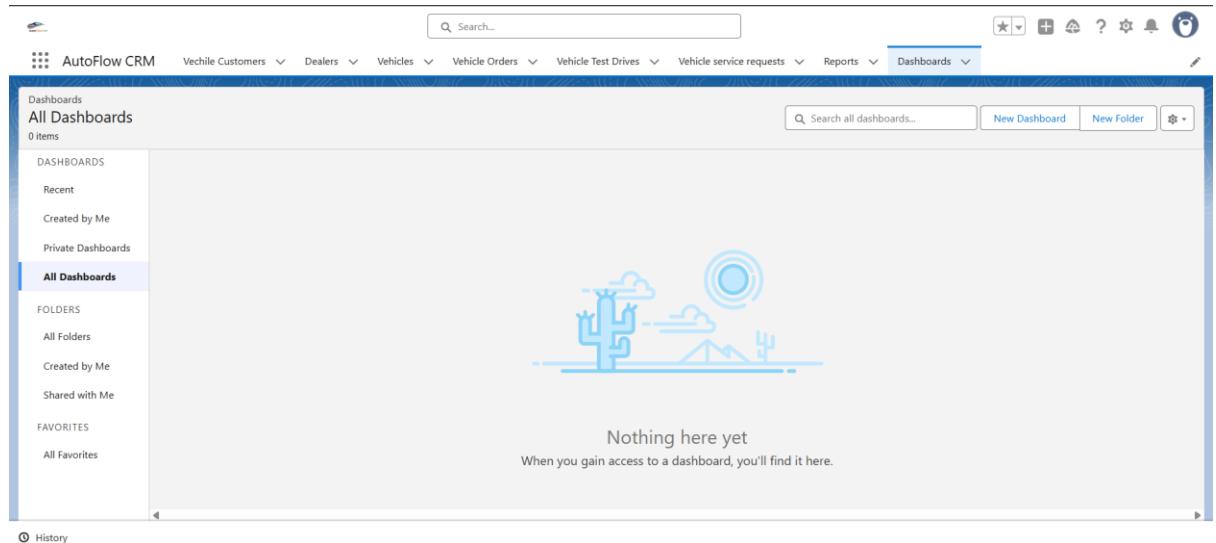
Dashboards

To complement the reports, **interactive dashboards** were developed with real-time visualizations including graphs, charts, and gauges. Key dashboard features included:

- Tracking dealership performance and stock availability.
- Monitoring sales funnels and lead conversion rates.



- Enabling drill-down capabilities for managers to explore individual metrics and uncover actionable insights.



Security Review

Alongside analytics, this phase prioritized **data security and compliance** to protect sensitive information and ensure trust in the CRM system:

- **Field-Level Security (FLS):** Restricted access to sensitive fields so only authorized personnel could view or update them.

Action	Profile Name	User License	Custom
<input type="checkbox"/>	Action	Analytics Cloud Integration User	
<input type="checkbox"/>	Edit Clone	Analytics Cloud Security User	
<input type="checkbox"/>	Edit Clone	Authenticated Website	
<input type="checkbox"/>	Edit Clone	Authenticated Website	
<input type="checkbox"/>	Edit Clone	External Apps Login	✓
<input type="checkbox"/>	Edit Clone	Chatter External User	
<input type="checkbox"/>	Edit Clone	Chatter Free User	
<input type="checkbox"/>	Edit Clone	Chatter Moderator User	
<input type="checkbox"/>	Edit Clone	Contract Manager	
<input type="checkbox"/>	Edit Clone	Cross Org Data Proxy User	
<input type="checkbox"/>	Edit Del ...	Custom: Marketing Profile	
<input type="checkbox"/>	Edit Del ...	Custom: Sales Profile	✓
<input type="checkbox"/>	Edit Del ...	Custom: Support Profile	✓

- <https://brave-offer-5m0t-dev-ed.trailblaze.lightning.force.com/lightning/setup/EnhancedProfiles/home>
- **IP Restrictions:** Limited CRM access to trusted networks, preventing unauthorized entry.
- **Audit Trails:** Captured record changes (who made them and when), ensuring accountability and supporting compliance reporting and risk management.
- Create Permission Sets for Sensitive Fields 1. Setup → Permission Sets → New. 2. Label: CandidateSensitiveAccess. 3. Object Settings → Candidate__c → Enable visibility for Salary_Expectation__c and Offer_Details__c. 4. Click Manage Assignments → Assign only to Dealer users

Action	Permission Set Name	Description	License
<input type="checkbox"/>	Authenticated Buyer	An authenticated external user with the ability to make and manage their...	Salesforce Payments External
<input type="checkbox"/>	Buyer	Allows access to the store. Lets users see products and categories, ma...	B2B Buyer Permission Set One Seat
<input type="checkbox"/>	Buyer Manager	Includes all Buyer capabilities, and allows access to manage carts and...	B2B Buyer Manager Permission Set One Seat
<input type="checkbox"/>	C360 High Scale Flow Integration User	Allows integration user to access features specific to C360 High Scale ...	Cloud Integration User
<input type="checkbox"/>	CRM User	Denotes that the user is a Sales Cloud or Service Cloud user.	CRM User
<input type="checkbox"/>	Commerce Admin	Allow access to commerce admin features.	Commerce Admin Permission Set License Seat
<input type="checkbox"/>	ConnectivityServiceCASCPPermSet		Cloud Integration User
<input type="checkbox"/>	Data Cloud Home Org Integration User	Allows integration user to access entities specific to Remote Data Cloud.	Cloud Integration User
<input type="checkbox"/>	DeliveryEstimationServicePermSet		Cloud Integration User
<input type="checkbox"/>	Event Monitoring User	Query all Event Monitoring data, including Event Log Files, Event Log O...	Salesforce
<input type="checkbox"/>	FieldServiceMobileStandardPermSet	Give your mobile workforce access to the Field Service mobile app. Set...	Field Service Mobile
<input type="checkbox"/>	Marketing Cloud Reporting C2C Perm	Allows MCR app to access data from core	Cloud Integration User

Configure Field-Level Security (FLS) 1. Setup → Object Manager → Select Candidate__c. 2. Click Fields & Relationships → Choose Salary_Expectation__c. 3. Click Set Field-Level Security. 4. Uncheck

Recruiter visibility → Save.

The screenshot shows the Salesforce Setup interface with the 'Sharing Settings' tab selected. The main title is 'Organization-Wide Sharing Defaults Edit'. A note at the top states: 'Edit your organization-wide sharing defaults below. Changing these defaults will cause all sharing rules to be recalculated. This could require significant system resources and time depending on the amount of data in your organization. Setting an object to Private makes records visible to record owners and those above them in the role hierarchy, and access can be extended using sharing rules.' Below this are two tables: 'Default Internal Access' and 'Default External Access', each with dropdown menus for various object types like Lead, Account, Order, etc. The 'Grant Access Using Hierarchies' column contains checkboxes. At the bottom right is a 'Help for this Page' link.

Outcome

Phase 9 reinforced the CRM's role as both a secure and strategic platform. By combining powerful reporting, dynamic dashboards, and rigorous security measures, the organization gained the ability to:

- Monitor dealership and sales performance in real time.
- Make well-informed, data-driven decisions.
- Protect the confidentiality, integrity, and compliance of customer and business data.

Together with **Phase 8 (Data Management & Deployment)**, this phase ensured the CRM was not only operationally efficient but also strategically valuable for the organization.

Phase 10: Final Presentation & Outcomes

The final phase of the AutoFlow CRM implementation focused on presenting the fully integrated system to stakeholders, including management, sales teams, and IT staff. This phase was critical to demonstrate the system's capabilities, validate that all business requirements had been met, and highlight the measurable benefits achieved during implementation.

Live Demonstration of Core Functionalities

The presentation was structured around a live demonstration of key CRM functionalities, emphasizing end-to-end business processes that had been streamlined through the system:

1. Secure Access and Personalized Dashboards

- Demonstrated logging in as a dealership administrator.
- Showcased role-based permissions and secure access protocols.
- Highlighted personalized dashboard views, including performance metrics, sales figures, stock levels, customer requests, and pending actions.
- Illustrated the system's ability to manage multiple dealerships on a centralized platform while maintaining operational efficiency and data security.

2. Customer Order Management and Dealer Assignment

- Showed creation of a customer order and automatic assignment to the nearest available dealer.
- Utilized geolocation and automated routing to suggest the most appropriate dealership based on proximity, vehicle availability, and workload.
- Highlighted CRM's ability to optimize resource allocation, reduce response times, and enhance customer satisfaction.

3. Test Drive Scheduling Automation

- Demonstrated scheduling of test drives with automated SMS and email notifications sent to both customers and dealerships.
- Reduced manual follow-ups and minimized scheduling errors.
- Showcased how automation streamlines daily tasks, improves accuracy, and enhances the overall customer experience.

4. Order Lifecycle Monitoring

- Tracked all stages of the order lifecycle, from initial request to final delivery.
- Enabled sales representatives and managers to view status updates, pending tasks, and potential bottlenecks.
- Provided transparency, improved operational control, and strengthened trust between dealerships and customers.

5. Analytics and Insights via Dashboards

- Presented interactive dashboards visualizing dealership efficiency, vehicle stock levels, sales performance, and customer satisfaction metrics.
- Allowed stakeholders to drill down into datasets, identify trends, and make data-driven decisions.
- Positioned the CRM as a central hub for both tactical and strategic planning.

Demo Link

Stakeholders and reviewers can access a full demonstration of the AutoFlow CRM system via the following link:

https://drive.google.com/file/d/1uCP_9_ymXLO4XrohBPH0UN64po5n_8rE/view?usp=sharing

