sree navaneeth polimetla

report internship_ Navanneth.docx

SRM University AP Amravati

Document Details

Submission ID
trn:oid:::8044:105494376

Submission Date
Jul 23, 2025, 12:05 PM GMT+5:30

Download Date
Jul 23, 2025, 12:07 PM GMT+5:30

File Name
report internship_ Navanneth.docx

File Size
1.9 MB

27 Pages

3,790 Words

22,477 Characters

0% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

Match Groups

1 Not Cited or Quoted 0%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

0% Internet sources
0% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **1** Not Cited or Quoted 0%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

0% Internet sources
0% Publications
0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Submitted works

The American University of Kurdistan on 2021-09-15 **<1%**

A REPORT

ON

ENHANCING CUSTOMER ORDERING EXPERIENCE IN SALESFORCE CRM FOR

WHATSNEXT VISION MOTORS

By

Sree Navaneeth Polimetla

Registration No: AP23110010757

Prepared in the partial fulfillment of the

Summer Internship Course

AT

SMARTBRIDGE EDUCATIONAL SERVICES PRIVATE LIMITED
(Virtual Internship)

In Collaboration with AICTE and Salesforce

SRM UNIVERSITY, AP

(July, 2025)

INTERNSHIP CERTIFICATE

JOINING REPORT

To

[mentor name]

SRM University – AP

Subject: Internship Joining Confirmation

Respected Sir/Madam,

I, **Sree Navaneeth Polimetla**, bearing Registration Number **AP23110010757**, hereby confirm that I have commenced my virtual internship under the **Salesforce Supported Virtual Internship Program 2025,** organized by **SmartBridge Educational Services Pvt. Ltd**. in collaboration with **AICTE** and **Salesforce**.

Start Date: 1st June 2025

End Date: 25th July 2025

During my internship, I took part in instructor-led workshops, finished Trailhead courses at my speed, and worked on a guided project called:

"Enhancing Customer Ordering Experience in Salesforce CRM for WhatNext Vision Motors."

I request you to kindly consider this report for evaluation under the Summer Internship Course.

Sincerely,

Sree Navaneeth Polimetla

Reg. No: AP23110010757

SRM University – AP

ACKNOWLEDGEMENTS

I want to thank everyone who helped and mentored me throughout my internship from the bottom of my heart.

First and foremost, I want to thank SmartBridge Educational Services Pvt. Ltd. and Salesforce gave me the chance to take part in the Salesforce Supported Virtual Internship Program 2025, which was put up by AICTE and Salesforce.

I want to thank the SmartBridge mentors and trainers very much. Their professional sessions and technical advice made it easier for me to understand difficult Salesforce ideas.

I'd also want to thank Dr. Mekala Ratna Raju, my Faculty Mentor from SRM University – AP, for always being there for me, giving me comments, and encouraging me throughout my internship.

1 Lastly, I want to thank my parents and friends for being there for me, being patient, and keeping me motivated throughout the program.

Sree Navaneeth Polimetla

Reg. No: AP23110010757

ABSTRACT

This report shows what happened during a virtual internship assignment that was part of the "Salesforce Supported Virtual Internship Program 2025" put on by SmartBridge Educational Services Pvt. Ltd. worked with AICTE and Salesforce. The main goal of the internship was to learn how to use Salesforce CRM by using it on a real project.

The project, called "Enhancing Customer Ordering Experience in Salesforce CRM for WhatNext Vision Motors," is all about making things better for customers in the car business. Some of the main aims were to automate the order processing workflow, check stock availability in real time, and distribute orders to the closest dealership depending on where the client is located.

The project included developing data models with custom objects and connections, automating business logic using record-triggered flows, and checking stock with Apex triggers. Also, batch Apex tasks were set up to constantly check the progress of bulk orders depending on how many vehicles were available.

This solution makes sure that clients can only purchase automobiles that are in stock and gives the sales staff real-time updates. This makes things more transparent, efficient, and satisfying for customers. The internship also gave me a chance to learn about Lightning App Builder, Flow Builder, Apex programming, and Salesforce best practices.

TABLE OF CONTENTS

AN OVERVIEW OF THE BUSINESS SECTOR OF THE ORGANIZATION

India's car industry is one of the biggest in the world and is very important for the country's economic growth. It includes a broad range of tasks, such as making, distributing, selling, repairing, and providing after-sales assistance for two-wheelers, passenger cars, and commercial vehicles. The business is still going through big changes because of things like expanding urbanization, rising disposable incomes, and improvements in electric and hybrid automobiles.

But the car sector also has a lot of problems to cope with, such keeping track of enormous stocks, making sure customers get help on time, and maintaining strong dealer networks. In this highly competitive environment, using digital tools is now necessary to make operations more efficient and get more customers involved.

Automotive firms have found that Customer Relationship Management (CRM) platforms like Salesforce are quite useful. Salesforce helps handle large amounts of customer data, automate tasks, and make it easier for dealerships, customers, and service centers to talk to each other. In this field, features like lead management, automatic order processing, real-time alerts, and making decisions based on data are quite important.

Customers want experiences that are more customized, faster, and more open. By using cloud-based CRM solutions in automotive operations, service delivery will improve and brand loyalty will grow. The goal of this project is to meet these business goals by utilizing Salesforce CRM for a made-up car company called WhatNext Vision Motors.

A LOOK AT THE ORGANIZATION

1. A Short History

SmartBridge Educational Services Pvt. Ltd., which was founded in 2015, is an edtech firm situated in India with its main office in Hyderabad, Telangana. The goal of starting it was to close the gap between what businesses want and what schools teach, particularly for students in engineering and technology. The firm started because academics were worried that students were graduating without enough real-world experience with technology and business methods.

SmartBridge started offering project-based programs under the name SmartInternz because it knew that students needed to learn by doing in new fields like AI, IoT, cybersecurity, and Salesforce CRM. The platform immediately became popular and spread throughout the country, giving students in both urban and rural India the chance to take part in live virtual internships, guided projects, and superbadge training provided by Salesforce Trailhead.

SmartBridge has also inked institutional MoUs with hundreds of Indian colleges and universities. These agreements are in line with the activities of government authorities like AICTE and UGC. This way, the corporation makes sure that its programs are formally recognized and fit within the school curriculum. Over time, it became a well-known link between academia and business and an important part of virtual talent development.

SmartBridge has set a big aim in the last several years: to help 1 million virtual internships by 2026, making it a "talent factory" for the Indian IT sector.

2. Size of the Business

SmartBridge is a medium-sized edtech startup that has a significant online presence. Starting in 2025:

Employees: About There are 100 to 150 full-time employees that work on product development, academic alliances, training, support, and commercial growth.

Students Trained: More than 1.25 million students have taken part in seminars, internships, and certification programs.

Academic Reach: worked with more than 2,700 colleges and institutions in India via virtual partnerships.

Corporate Partners: Worked with more than 50 industry customers and recruiting partners to provide students real-world examples to work with.

Websites:

SmartInternz.com is a website for internships and project-based learning.

Skill Wallet is a dashboard for students to keep track of their credentials.

SmartBridge LMS is an internal learning management system for mentors and groups.

Infrastructure for technology: Cloud-based systems for hosting internships, GitHub-based project repositories, automatic progress monitoring, mentor chat, and assessment routines.

SmartBridge can conduct numerous groups of learners at the same time across a wide range of technological areas, such as Salesforce, Azure, AI/ML, IoT, and cybersecurity.

3. Services and product lines

SmartBridge offers a wide range of services for students, including:

a. SmartInternz

The main platform that works with AICTE, Salesforce, and Microsoft to provide virtual internships and supervised capstone projects. Students work on actual projects in simulated industrial settings with technologies like GitHub, Trailhead, and project workspaces.

b. Skill Wallet

A website where students may keep track of their progress and keep a digital record of the badges, certifications, and internship work they have done. It also has:

- Calendar for internships
- Tracker for milestones
- Download of certificate
- Links to submit projects

c. Curriculum-Integrated Programs

SmartBridge works with teachers and department heads to make sure that its projects are part of B.Tech classes. SmartBridge mentors and university instructors jointly grade these modules, which are worth elective credit.

d. Hackathons, Build-a-thons, and Faculty Development

- SmartBridge conducts more than only student programs:
- Salesforce and AI Faculty Development Programs (FDPs)
- National Build-a-thon Challenges to get people to come up with new ideas
- Bootcamps to be ready for placement and help with writing resumes

SmartBridge's distinctive strength is its outcome-based strategy, which makes sure that learning goes beyond just consuming information and includes delivering and submitting projects via GitHub and video demo review.

4. Competitors

As a platform blending virtual internships, academic integration, and hands-on delivery, SmartBridge has few direct competitors. However, it operates in a space that overlaps with:

Competitor	Description
Internshala	Offer certification courses with laboratories that students may use, but charge a lot for them.
Simplilearn, Great Learning	Provide certification courses with hands-on labs but at premium pricing
NASSCOM FutureSkills	Partnership between the government and businesses for training and certifying based on skills
TCS iON	Provides programs that get people ready for work and give them college credits
Udemy and Coursera	MOOC providers that work all over the world, but not on Indian university credits or internships

SmartBridge's virtual internships that are ready for credit and may be immediately linked to a student's degree or semester project give it an advantage over other MOOC platforms throughout the world.

5. Organizational Structure & Departments

SmartBridge is set up as a horizontal team-driven organization that encourages flexibility, mentoring, and quick invention. The most important departments are:

- Product & Technology: Builds and keeps up SmartInternz, Skill Wallet, GitHub integrations, and mentor dashboards.
- The Academic and Partnership Cell makes sure that the internship curriculum is in line with the academic boards, oversees institutional MoUs, and gets universities involved.
- Internship Operations is in charge of assigning mentors, making schedules, reviewing tasks, and keeping track of cohort progress.
- Marketing and Outreach: This person brings on new college partners, does webinars, and handles contacts with students who are just starting out.
- Corporate Relations finds real-world initiatives and establishes employment pipelines with partners in the sector.

Leadership Team – The following people are in charge of the strategic vision:

  - Amarender Katkam is the founder and CEO.

- Suman Akula is one of the co-founders.
- Nitin Jain is the Director of Business.
- Sridevi Sira is the Vice President of Partnerships.

PLAN FOR THE INTERNSHIP PROGRAM

I, Sree Navaneeth Polimetla(Reg. No. AP23110010757), finished an 8-week virtual internship via the Salesforce Supported Virtual Internship Program 2025, which was run by SmartBridge Educational Services Pvt. Ltd. worked with AICTE and Salesforce on this.

1. Internship Duration

- **Start Date:** 1st June 2025
- **End Date:** 25th July 2025

2. Type and Style of Internship

- Mode: Online (Work from Home)
- SmartInternz is the platform used.
- Kanban-based workspace for project management
- Mentor Support: Through a chat system and evaluations based on milestones
- Deliverables: a project report, a project demo video, and a GitHub repository submission

3. What Modules/Departments Are Covered

Because this internship was oriented on skills and technology, the organization was based on Salesforce domains instead of regular divisions. The modules covered (which are like "departments visited") are listed below:

Week	Domain/Module	Topics Covered
Week 1	Basics of Salesforce and Data Modeling	Objects, Fields, Schema Builder, and Validation Rules
Week 2	Declarative Automation	Flows, Approval Processes, and Flow Elements
Week 3	Apex Programming	SOQL, SOSL, Apex Triggers, Classes, and Testing

Week 4	Visualforce & Integration	REST/SOAP APIs, Callouts, and the Basics of LWC
Week 5	Flow Optimization & Authentication	Screen Flows, Record/Auto/Trigger Flows
Week 6	Apex & Batch Jobs	Bulk-safe Logic, Batch Apex, Scheduled Apex
Week 7	AgentBlazer (AI Module)	Building Agents, AI Governance
Week 8	Capstone Project	Final project implementation and submission

4. Roles & Responsibilities

As an intern, I had to do the following:

Trailhead Learning: Finished a lot of courses and superbadges on Apex, Flows, Authentication, and Object Access.

- **Hands-on Projects:**
- Built and changed Salesforce objects and connections for a made-up car manufacturer called WhatNext Vision Motors.
- Used Record-Triggered Flows to create automation that sends orders to the closest dealers.
- Created Apex triggers to check stock before processing client orders.
- Scheduled Batch Apex to change the status of an order depending on whether or not a car was available.

Project Submission: Kept the coding on GitHub, validated processes in Salesforce Developer Org, and sent in a video of the project showcase.

This course was really thorough and taught me how to utilize declarative tools, programmatic logic, and real-world examples on the Salesforce platform. Not only did it help me become better at technical things, but it also helped me get better at planning projects, writing documentation, and managing my time.

BACKGROUND AND EXPLANATION OF THE ISSUE

Like many other product-based industries, the automobile business has a lot of problems with customer service and operations when it comes to managing inventory, assigning dealers, and monitoring orders. To check car stock availability, find the nearest dealer, and update the order status, traditional systems generally need human involvement. This causes delays, inefficiencies, and unhappy customers.

In this case, the Salesforce CRM platform is a great place to create smart, automated solutions that meet client demands and fit in with how your business runs on the back end.

The fake car business WhatsNext Vision Motors is the focus of the internship's final assignment. The organization wanted to improve how it handled client orders to deal with the following main problems:

Customers may order automobiles that were out of stock, which caused uncertainty and loss of confidence since real-time stock validation wasn't available.

Manual Dealer Assignment: Sales teams had to manually designate the closest dealer, which slowed down processing and delivery.

No automated order updates: There was no way to automatically adjust the status of orders depending on supply availability, which caused communication problems.

These challenges are similar to what happens in many mid-sized car companies in India. Without a streamlined, data-driven, and automated system, company activities are slow and full of mistakes.

Problem Statement

"To automate the process of handling customer orders in Salesforce CRM for WhatNext Vision Motors, making sure that stock is checked in real time, dealers are assigned automatically, and order status updates are sent in batches."

Purpose of the Report

This report talks about how the Salesforce CRM system that was created during the internship was designed and put into use.

It uses:

- Tools that let you say what you want, like Flows
- Apex triggers and Batch Apex are examples of programmatic components.

- Features of the platform include custom objects, email automation, and record assignment.

MAIN TEXT

Creating and putting into action an order management system based on Salesforce CRM

1. Assumptions Made

To make the system in Salesforce behave like a real-world automotive process, the following assumptions were made:

- WhatsNext Vision Motors is in charge of a number of dealers in various cities in India.
- Based on the customer's address (using Zip/City logic), each order must be connected to the closest dealer that is accessible.
- There aren't many cars in stock, therefore you can only make an order if the car you want is in stock.
- If the item is out of stock, the order will be accepted but marked as "Pending." If it is in stock, it will be marked as "Confirmed."
- Regularly, inventory data (such the number of cars and supplies) is submitted by hand.
- The Batch Apex procedure must be able to handle large orders.

2. Data Modeling (Setting Up the Experiment)

The Salesforce data model was built using **custom objects and standard relationships**:

Custom Objects Created:

Object Name	Description
Vehicle__c	Keeps track of the name, model, variation, and stock of vehicles
Dealer__c	Shows vendors with fields for their locations
Customer__c	Shows customers who are making orders
Order__c	The main goal is to get the information of the order.

Relationships:

- Order__c can search up Customer__c, Vehicle__c, and Dealer__c.
- Vehicle__c has a stock count (Number field)
- Dealer__c has a location (city and pin code).

This model lets you filter cars based on their availability and attach the right dealer to them.

3. Automation using Flows (Declarative Work)

Record-Triggered Flow:

Trigger: When an Order__c record is created

- **Logic:**
    - Get the customer's city
    - Find all Dealer__c records in that city
    - Automatically assign the closest available dealer
    - Check to see whether Vehicle__c has more than 0 stock.
    - Set Order_Status__c to "Pending" if stock = 0.
    - Otherwise, "Confirmed"
    - **Outcome:** Dealer is auto-assigned and order is validated instantly.

Flow Builder Elements Used:

- Get Records
- Decision
- Assignment
- Update Records

4. Apex Trigger Logic (Programmatic Logic)

Flows took care of normal logic, while Apex triggers were utilized to enforce important business rules:

OrderTrigger.trigger (After Insert)

```
trigger OrderTrigger on Order__c (after insert) {
    for (Order__c ord : Trigger.new) {
        Vehicle__c v = [SELECT Stock__c FROM Vehicle__c WHERE Id = :ord.Vehicle__
        if (v.Stock__c <= 0) {
            ord.Order_Status__c = 'Pending';
        } else {
            ord.Order_Status__c = 'Confirmed';
            v.Stock__c -= 1;
            update v;
        }
    }
}
```

This logic ensures:

- Every time a new order is placed, the stock is checked.
- If verified, the number of vehicles in stock goes down.

Best Practices Used:

- Used the Trigger Handler pattern (moved code to a class to make it more modular)
- Checked design for bulk safety
- If required, used custom metadata to set thresholds

5. Scheduled & Batch Apex Job

A Batch Apex task was made to help with bulk processes by:

- Do it every night
- Look at all of your "Pending" orders.
- Check the supply of vehicles again
- Change the status if stock becomes available.

OrderBatch.cls (Batch Class Overview)

```
global class OrderBatch implements Database.Batchable<SObject> {
    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator('SELECT Id, Vehicle__c FROM Order__c WHERE Order_Status__
    }

    global void execute(Database.BatchableContext bc, List<Order__c> scope) {
        for (Order__c ord : scope) {
            Vehicle__c v = [SELECT Stock__c FROM Vehicle__c WHERE Id = :ord.Vehicle__c];
            if (v.Stock__c > 0) {
                ord.Order_Status__c = 'Confirmed';
                v.Stock__c -= 1;
                update v;
            }
        }
        update scope;
    }

    global void finish(Database.BatchableContext bc) {
        // Log or send completion email
    }
}
```

6. Testing and Validation

- We used Developer Console and manual test records to test all the triggers and flows.
- Made test classes to make sure that more than 75% of the code was covered.
- Enabled and watched flow debug logs.
- Verified end-to-end:
    - Making an order
    - Dealer assignment
    - Managing stock

7. Results and Interpretation

Feature	Result Achieved
Checking stocks in real time	Works well with Trigger & Flow
Assignment of auto dealers	How to do it using Flow
Update on the progress of the batch	Confirmed via a scheduled job
Scalability	System supports large data via Batch
Error Handling	Trigger with exception try-catch

There were no major differences between the predicted and actual outcomes throughout testing. All of the main use cases and logic routes were checked.

8. Tools & Platforms Used

- Salesforce Developer Organization
- Salesforce Flow Builder
- Console for Apex Developers
- GitHub is for keeping track of different versions of code.
- Trailhead: Finished modules and superbadges
- SmartInternz Workspace: a place to work on projects and keep track of tasks

RESULTS

At the completion of the WhatNext Vision Motors Salesforce CRM-based internship project, the following results were reached:

Results of Project Implementation

- Built a unique Salesforce CRM system for managing orders in a fake car firm.
- Created and set up bespoke objects like Vehicle__c, Order__c, Dealer__c, and Customer__c, making sure they had the right connections.
- Created and set up a record-triggered flow that automatically assigns dealers and checks stock when an order is made.
- Created and tested an Apex trigger to keep track of order status and decrease inventory in real time.
- Set up a Batch Apex task to check on outstanding orders and change their status depending on stock levels on a regular basis.
- The procedure of checking orders and assigning dealers is now fully automated.
- Both human and automated testing showed that all essential logic routes were correct.

Skills in Technology Gained

- Learned how to model data using custom objects and construct schemas in Salesforce.
- Got to use Flow Builder in real life, including record-triggered, screen, and auto-launched flows.

- Wrote and tested Apex code for things like triggers, classes, and batch tasks.
- Using the Developer Console, I learned how to troubleshoot and handle errors.
- Learned how to submit projects to GitHub, keep track of different versions, and do peer reviews.

Trailhead And Learning Successes

- Finished a number of Trailhead Superbadges that were connected to:
  - Relationships between objects
  - Checking the Data
  - Managing Flow
  - Programming in Apex
  - Basics of Screen Flow
- Understood and used AgentBlazer AI ideas (optional module)

Growth in your career

- Created a discipline for delivering real-world projects, including monitoring milestones, testing, and writing documentation.
- Regularly talking to mentors and sending updates helped me communicate better.
- Learned how to manage my time by having weekly tasks and being able to study at my own speed.

CONCLUSIONS AND/OR SUGGESTIONS

Conclusions

The Salesforce internship project called "Enhancing Customer Ordering Experience in Salesforce CRM for WhatNext Vision Motors" gave me a chance to use CRM and automation ideas in a real company setting.

By designing and building a bespoke order management system, I was able to show that I understood:

- Setting up the Salesforce platform (objects, fields, and relationships)
- Using Flows to automate business processes
- Apex programming for business logic and checking stock
- Batch Apex for processing large amounts of data and updating status

The project met all of its main goals:

- Stopped consumers from ordering autos that were out of stock
- Automatically given the closest dealer depending on where you are
- Status updates on pending orders are scheduled.
- Made the simulated company's service more reliable and its operations more efficient

Also, the internship helped me become better at Salesforce programming, Trailhead learning, and submitting professional projects.

SUGGESTIONS

Based on what I learned from this research, I think the following changes should be made for future versions or real-world use:

1. **Geolocation Integration**
   Integrate actual maps-based APIs (e.g., Google Maps) to calculate distance more accurately for assigning the nearest dealer.
2. **Customer Notifications via Email/SMS**
   Implement email alerts or mobile push notifications to inform customers about order confirmation, stock issues, or delays.
3. **Inventory Management Dashboard**
   Add a dashboard for internal teams to visually track vehicle stock, pending orders, and dealer availability.
4. **Role-Based Access**
   Apply Salesforce security model (profiles, roles, permission sets) to ensure data access based on user role (e.g., Sales Manager vs Dealer).
5. **Automated Reporting**
   Use Salesforce Reports and Dashboards to provide weekly insights on order trends, cancellations, and fulfillment rates

APPENDICES

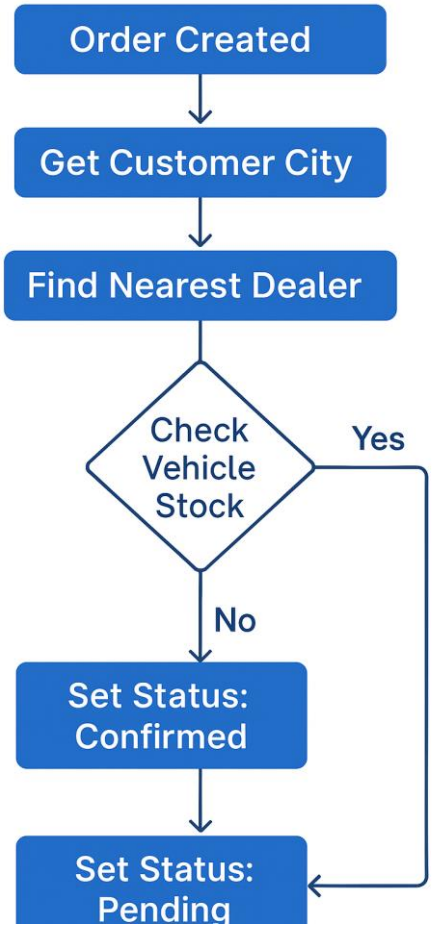
The project documentation includes the following materials:

Appendix A: GitHub Repository Link

Project files (Apex Classes, Flows, Schema, Batch Job Code) are uploaded to GitHub:
[Insert your GitHub Repository URL here]

Appendix B: Flow Design Diagram

A simplified visual of the **Order Validation and Dealer Assignment Flow**:



Appendix C: Sample Screenshots

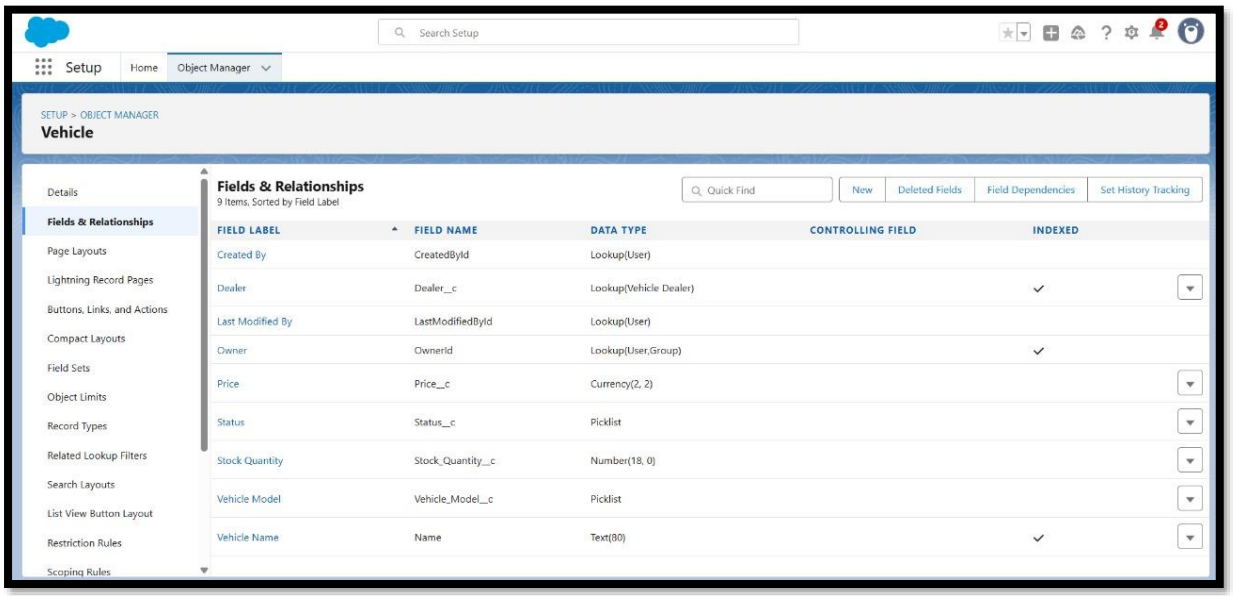


Figure 1 Custom Object Schema Builder View

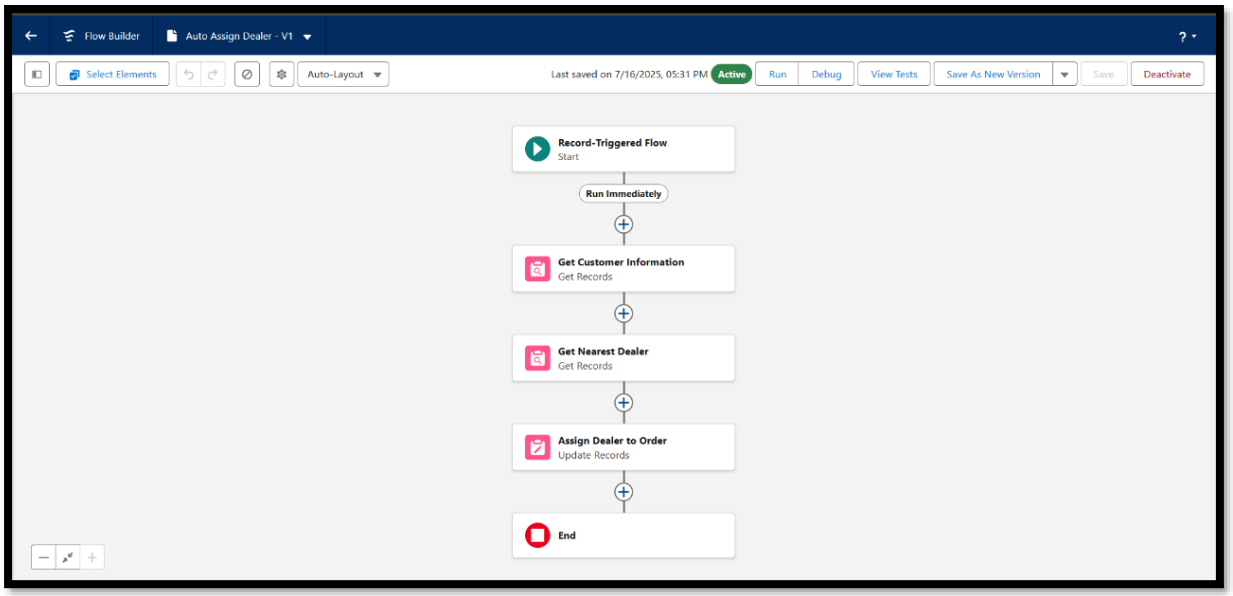


Figure 2 Flow Builder Design: Auto Assign Dealer

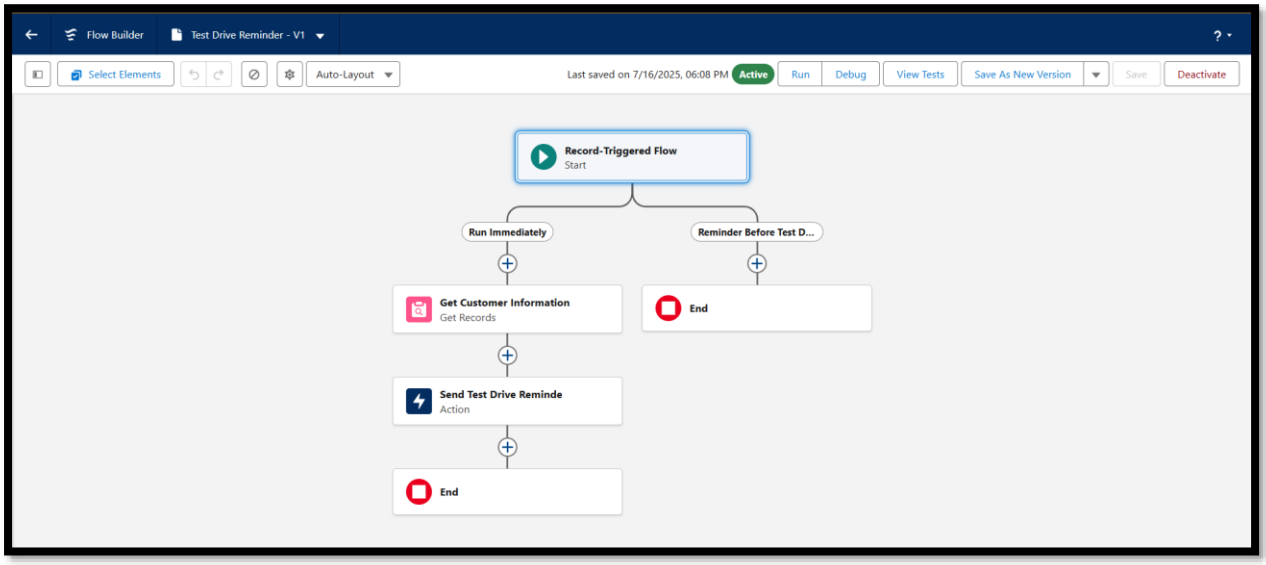


Figure 3 Flow Builder Design: Test Drive Reminder

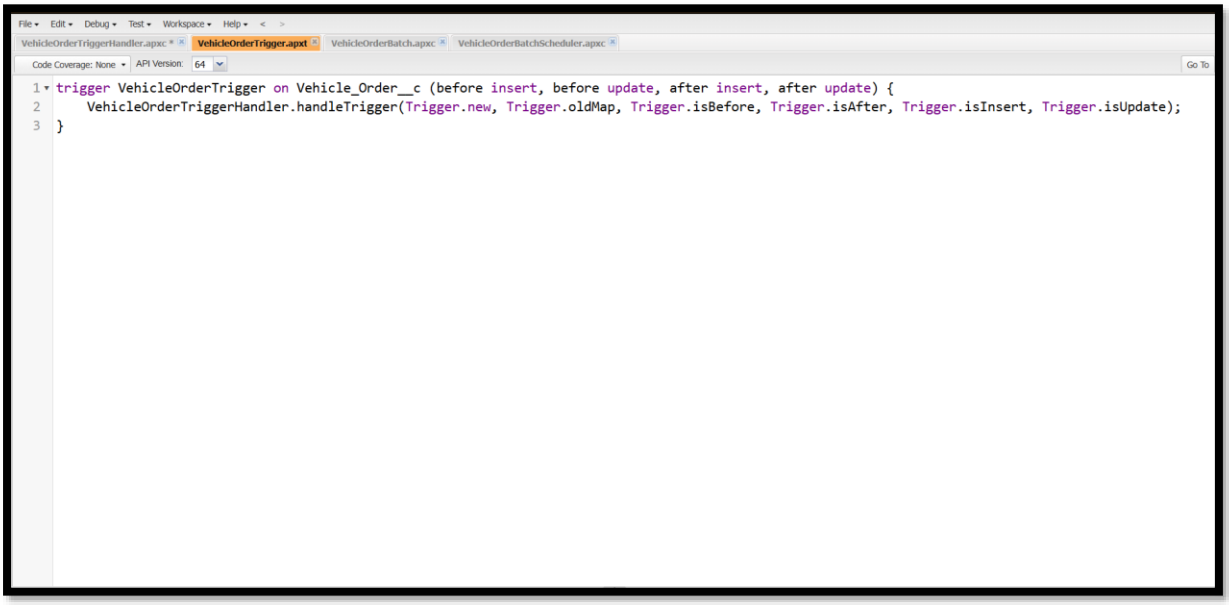
```
public class VehicleOrderTriggerHandler {
    public static void handleTrigger(List<Vehicle_Order__c> newOrders, Map<Id, Vehicle_Order__c> oldOrders, Boolean isBefore, Boolean isAfter, Boolean isInsert, Boolean isUpdate) {
        if (isBefore && (isInsert || isUpdate)) {
            preventOrderIfOutOfStock(newOrders);
        }
        if (isAfter && (isInsert || isUpdate)) {
            updateStockOnOrderPlacement(newOrders);
        }
    }
    private static void preventOrderIfOutOfStock(List<Vehicle_Order__c> orders) {
        Set<Id> vehicleIds = new Set<Id>();
        for (Vehicle_Order__c order : orders) {
            if (order.Vehicle__c != null) {
                vehicleIds.add(order.Vehicle__c);
            }
        }

        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(
                [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
            );

            for (Vehicle_Order__c order : orders) {
                if (vehicleStockMap.containsKey(order.Vehicle__c)) {
                    Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                    if (vehicle.Stock_Quantity__c <= 0) {
                        order.addError('This vehicle is out of stock. Order cannot be placed.');
                    }
                }
            }
        }
    }
    private static void updateStockOnOrderPlacement(List<Vehicle_Order__c> orders) {
        Set<Id> vehicleIds = new Set<Id>();
        for (Vehicle_Order__c order : orders) {
            if (order.Vehicle__c != null && order.Status__c == 'Confirmed') {
                vehicleIds.add(order.Vehicle__c);
            }
        }
        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(
                [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
            );

            List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();
            for (Vehicle_Order__c order : orders) {
                Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                if (vehicle.Stock_Quantity__c > 0) {
                    vehicle.Stock_Quantity__c -= 1;
                    vehiclesToUpdate.add(vehicle);
                }
            }
            if (!vehiclesToUpdate.isEmpty()) {
                update vehiclesToUpdate;
            }
        }
    }
}
```

Figure 4 Apex Class Name: VehicleOrderTriggerhandler



```
trigger VehicleOrderTrigger on Vehicle_Order__c (before insert, before update, after insert, after update) {
    VehicleOrderTriggerHandler.handleTrigger(Trigger.new, Trigger.oldMap, Trigger.isBefore, Trigger.isAfter, Trigger.isInsert, Trigger.isUpdate);
}
```

Figure 5 Apex Class in Trigger Class: VehicleOrderTrigger

```
global class VehicleOrderBatch implements Database.Batchable<sObject> {
    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator([
            SELECT Id, Status__c, Vehicle__c
            FROM Vehicle_Order__c
            WHERE Status__c = 'Pending'
        ]);
    }
    global void execute(Database.BatchableContext bc, List<Vehicle_Order__c> orders) {
        Set<Id> vehicleIds = new Set<Id>();
        for (Vehicle_Order__c order : orders) {
            if (order.Vehicle__c != null) {
                vehicleIds.add(order.Vehicle__c);
            }
        }
        if (!vehicleIds.isEmpty()) {
            Map<Id, Vehicle__c> vehicleStockMap = new Map<Id, Vehicle__c>(
                [SELECT Id, Stock_Quantity__c FROM Vehicle__c WHERE Id IN :vehicleIds]
            );

            List<Vehicle_Order__c> ordersToUpdate = new List<Vehicle_Order__c>();
            List<Vehicle__c> vehiclesToUpdate = new List<Vehicle__c>();

            for (Vehicle_Order__c order : orders) {
                Vehicle__c vehicle = vehicleStockMap.get(order.Vehicle__c);
                if (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.');
    }
}
```

Figure 6 Apex Class Name: VehicleOrderBatch

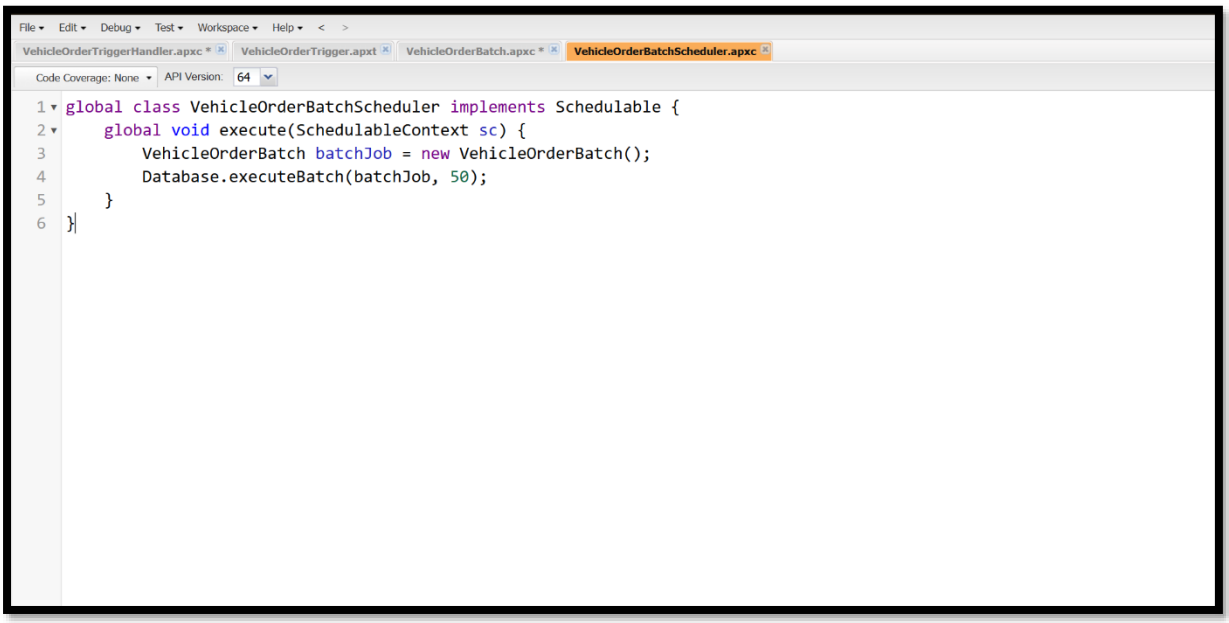


Figure 7 Apex Class Name: VehicleOrderBatchScheduler

REFERENCES

1. Salesforce Developer Documentation
   https://developer.salesforce.com/docs
2. Salesforce Trailhead
   https://trailhead.salesforce.com
3. Trailhead Modules & Superbadges:
   - Object Relationships Superbadge
   - Data Validation Superbadge
   - Flow Administration Superbadge
   - Apex Specialist Superbadge
   - User Authentication Troubleshooting Superbadge
4. SmartBridge Internship Platform (SmartInternz)
   https://www.smartinternz.com
5. GitHub Docs – Repository Hosting
   https://docs.github.com
6. Internal Mentor Resources – SmartBridge Guided Project Docs
7. Code examples – Custom-written, tested in Developer Org