Shell Hacks 2021 AutoNation Challenge

Problem:

AutoNation is the largest automotive dealer in the US. We sell thousands of vehicles every day and we repair and service millions of vehicles every year. Our associates want to be able to deliver the best customer experience but there is limited time in the day. We want to save the associates time and deliver a better customer experience.

Challenge:

Build an application that can receive and send SMS messages and respond conversationally based on the context of the request.

We have provided some sample mocked customer and store data which we want you to use to come up with interesting and novel ways in which your application can recognize the customers intent and answer their queries.

Customers are human beings and can ask questions in a variety of ways, so your solution must be able to handle the variation of human speech and phrasing. It is also possible that you will need to ask follow up questions to give them an answer to their questions.

Suppose you want to know when the dealership is open a customer could say things like:

Example 1:
Customer: What are your service hours on Monday?
Bot: We are open from 8AM to 4PM on Monday
Example 2:
Customer: When are you open?
Bot: Do you want sales, service, or collision hours?
Customer: service
Bot: Our service drive is open from Monday-Friday 7AM to 7PM and Saturday 8AM to 5PM
Example 3:
Customer: Tell me your sales hours for the weekend
Bot: Our showroom is open Saturday 9AM to 8PM and on Sunday from 10AM to 7PM

Judging Criteria

- Creativity (50%)
- Complex Interactions (30%)
- SMS Integration (10%)
- Code Readability (10%)

Prizes

- Every member of the winning team will receive a set of Apple AirPods Pro

Tips:

1. We are hosting a session on building conversational AI using Amazon Lex (but feel free to use whatever tools you want to)
2. Twilio has SMS sending and receiving capabilities and offers $15.50 free trial credits no credit card needed
3. https://docs.aws.amazon.com/lex/latest/dg/twilio-bot-association.html

Appendix:

Explanation of sample data

1. storeInfo.json
    a. storeId – the primary key identifier for the store
    b. name – the display name of the store
    c. address – the location of the store
    d. storeTimeZone – time zone information for the store
        i. zoneId – the zoneId within the tz database https://en.wikipedia.org/wiki/List_of_tz_database_time_zones
        ii. gmtOffset – the number of hours to offset from Greenwich Mean Time (GMT)
        iii. dstSavings – if 1 is currently in daylight savings time if 0 in standard time
    e. phoneNumber – the primary phone number of the story
    f. latitude – the latitude of the store's location
    g. longitude – the longitude of the store's location
    h. faxNumber – the primary fax number of the store
    i. webAddress – the primary webpage for the store
    j. serviceUrl – the url to book an appointment at the store
    k. brands – the vehicle brands that are sold and serviced at the store
    l. salesHours – days and hours (local 24hour time) that the sales department is open
    m. serviceHours – days and hours (local 24hour time) that the service department is open
    n. collisionHours – days and hours (local24hour time) that the collision center is open
2. customerInfo.json
    a. customerId – the primary key identifier for the customer
    b. firstname – the first name of the customer
    c. lastName – the last name of the customer
    d. vehicles – the vehicles the customer owns
        i. vehicleId – the primary key identifier for the vehicle
        ii. year – the model year of the vehicle
        iii. make – the brand make of the vehicle
        iv. model – the model name of the vehicle
        v. mileage – the last recorded number of miles the vehicle has been driven
    e. appointments – service appointments the customer has upcoming
        i. appointmentId – the primary key identifier for the appointment
        ii. appointmentDateTime – the date and time of the appointment in iso8601 format https://en.wikipedia.org/wiki/ISO_8601

iii. vehicleId – the id of the vehicle the appointment is for
iv. storeId – the id of the store the appointment is for
v. reasons – a list of reasons the vehicle is being brought in for service
    1. for purposes of this exercise will be one or more of the following:
        a. oilChange
        b. tireRotation
        c. brakes
        d. batteryReplacement
        e. engineIssue
        f. manufacturerRecall
        g. inspection

f. repairOrders – the current and past repairs/services that the customer has had
    i. repairOrderId – the primary key identifier for the repair order
    ii. vehicleId – the id for the vehicle the repair order is for
    iii. storeId – the id for the store the repair order is for
    iv. status – the current state of the repair order will be one of the following values
        1. OPEN – the repair order is active and the car is being worked on
        2. COMPLETED – the service has been completed on the car but the customer has not picked up the vehicle
        3. CLOSED – the repair order has been completed and the customer has picked up the vehicle
    v. openDateTime – the date and time in iso8601 format that the repair order was created
    vi. closeDateTime – the date and time in iso8601 format that the repair order was closed, if the status is not “CLOSED” this will be null
    vii. lineItems – the different parts and labor, along with quantity and price associated with the repair or service
        1. name – a description of the type of item
        2. partNumber – if a part the part number will be there, if labor will be null
        3. quantity – the number of parts used or hours of labor
        4. pricePerUnit – the cost to the customer per unit, quantity x pricePerUnit = total cost for the lineItem