Problem set 8.

Exercise 1. Webhook Integration Challenge

Webhook Development: Develop a webhook that integrates with a Dialogflow CX agent. This webhook should perform a specific action, such as querying a database, interacting with a third-party API, or executing a complex task based on user input.

Integration and Testing: Integrate the webhook with a Dialogflow CX agent and demonstrate how it enhances the conversational experience.

Deliverables:

Source code of the webhook.

Documentation of the webhook's functionality and integration steps.

(PS.: To solve exercise 1 you can use the steps and also the solution of exercise 2.)

Exercise 2. Weather Forecast Assistant

Write a Webhook in Google Dialogflow CX that retrieves real-time weather data (you can use the example below):

1. Create a weather forecast intent: Create an intent named "getWeatherForecast" that captures user requests for weather information.
2. Implement a webhook fulfillment: Develop a webhook fulfillment that retrieves real-time weather data from an external weather API using the user's location.
3. Parse weather data: Extract relevant weather information, such as current temperature, forecast for the next 24 hours, and weather conditions, from the API response.
4. Format weather response: Structure the weather information into a coherent and informative response for the user.
5. Return weather response to Dialogflow: Send the formatted weather response back to Dialogflow CX to be presented to the user.

Assessment:

1. Real-time data: Verify that the agent retrieves weather data in real-time and provides up-to-date information.

2. Location accuracy: Ensure that the agent correctly identifies the user's location and retrieves weather data for the correct location.
3. Weather information: Evaluate the accuracy and completeness of the weather information provided to the user.

Example:

Here is an example of how to write a webhook in Google Dialogflow CX that retrieves real-time weather data from an external weather API using the user's location:

External Weather API:

You can use the **OpenWeatherMap API** https://home.openweathermap.org/ to retrieve weather data. You'll need to create an account and obtain an API key.

Webhook Code:

https://colab.research.google.com/drive/1aKu1Srm5YZGQJiGE1A_dGSID__fybk6I?usp=sharing

```python
import json
import requests

def fulfill_webhook(session_id, request):
    """Fulfillment callback for the 'getWeatherForecast' intent."""

    # Extract user's location from the request
    location = request['queryResult']['parameters']['location']

    # Construct the OpenWeatherMap API URL
    api_url = "https://api.openweathermap.org/data/2.5/weather?q={location}&appid={api_key}".format(
        location=location, api_key={YOUR_API_KEY}
    )

    # Send the API request and receive the response
    response = requests.get(api_url)
    weather_data = response.json()

    # Extract relevant weather information from the API response
    current_temperature = weather_data['main']['temp']
    weather_description = weather_data['weather'][0]['description']

    # Format the weather information into a coherent response
    response_text = "The current temperature in {location} is {temperature} degrees Celsius. The weather is {description}.".format(
        location=location, temperature=current_temperature, description=weather_description
    )

    # Return the weather response to Dialogflow
    return {
        "fulfillmentText": response_text,
    }
```

Integration with Dialogflow CX:

1. Create a webhook integration: In your Dialogflow CX agent, go to the "Fulfillments" section and click "Create Integration." Select "HTTP Webhook" as the integration type and provide the URL of your webhook script.

2. Map intent to webhook: Create an intent named "getWeatherForecast" and map it to the webhook integration. This will ensure that whenever a user utterance matches the intent, the webhook will be triggered to handle the request.
3. Set location parameter: In the intent parameters, add a parameter named "location". This parameter will capture the user's location from the conversation and pass it to the webhook.
4. Test the weather forecast: Trigger the "getWeatherForecast" intent with a user utterance that specifies the location, such as "What is the weather in Berlin?". Ensure that the webhook retrieves the correct weather data and provides a relevant response to the user.