**Problem 3: Real-Time Traffic Monitoring System**

**Scenario:**

**1.Model the data flow for fetching real-time traffic information from an external API and displaying it to the user.**

from flask import Flask, jsonify, request

import requests

app = Flask(\_\_name\_\_)

@app.route('/traffic', methods=['GET'])

def get\_traffic\_data():

location = request.args.get('location') # Example query parameter

api\_key = 'YOUR\_API\_KEY' # Replace with your API key

api\_url = f'https://api.example.com/traffic?location={location}&apiKey={api\_key}' # Example API URL

try:

response = requests.get(api\_url)

response.raise\_for\_status() # Raise an exception for 4xx/5xx status codes

traffic\_data = response.json()

return jsonify(traffic\_data)

except requests.exceptions.RequestException as e:

print(f'Error fetching traffic data: {e}')

return jsonify({'error': 'Failed to fetch traffic data'}), 500

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**2.Implement a Python application that integrates with a traffic monitoring API (e.g., Google Maps Traffic API) to fetch real-time traffic data.**

import requests

import json

def fetch\_traffic\_data(api\_key, location):

url = f'https://maps.googleapis.com/maps/api/distancematrix/json?origins={location}&destinations={location}&key={api\_key}&departure\_time=now'

try:

response = requests.get(url)

Location)

Console

**3**. **Display current traffic conditions, estimated travel time, and any incidents or delays.**

import requests

def fetch\_traffic\_data(api\_key, origin, destination):

url = f'https://maps.googleapis.com/maps/api/distancematrix/json'

params = {

'origins': origin,

'destinations': destination,

'key': api\_key,

'departure\_time': 'now',

'traffic\_model': 'best\_guess',

}

try:

response = requests.get(url, params=params)

data = response.json()

# Check if response status is OK

if response.status\_code == 200 and data['status'] == 'OK':

# Extract relevant information

duration\_in\_traffic = data['rows'][0]['elements'][0]['duration\_in\_traffic']['text']

distance = data['rows'][0]['elements'][0]['distance']['text']

traffic\_condition = data['rows'][0]['elements'][0]['status']

# Print results

print(f"Origin: {origin}")

print(f"Destination: {destination}")

print(f"Distance: {distance}")

print(f"Estimated Travel Time in Current Traffic: {duration\_in\_traffic}")

print(f"Traffic Condition: {traffic\_condition}")

# Check for incidents or delays

if traffic\_condition != 'OK':

print("Traffic incidents or delays may be affecting travel time.")

else:

print(f"Error: {data['error\_message']}") if 'error\_message' in data else print("Request failed")

except requests.exceptions.RequestException as e:

print(f"Error fetching data: {e}")

# Replace with your Google Maps API key

API\_KEY = 'YOUR\_GOOGLE\_MAPS\_API\_KEY'

# Example usage

origin = 'New York, NY'

destination = 'Los Angeles, CA'

fetch\_traffic\_data(API\_KEY, origin, destination)

**4.Allow users to input a starting point and destination to receive traffic updates and alternative routes.**

import requests

def fetch\_traffic\_updates(api\_key, origin, destination):

url = f'https://maps.googleapis.com/maps/api/directions/json'

params = {

'origin': origin,

'destination': destination,

'key': api\_key,

'departure\_time': 'now',

'traffic\_model': 'best\_guess',

'alternatives': 'true',

}

try:

response = requests.get(url, params=params)

data = response.json()

# Check if response status is OK

if response.status\_code == 200 and data['status'] == 'OK':

# Print traffic updates and alternative routes

print(f"Traffic updates from {origin} to {destination}:\n")

# Print each route's summary

for i, route in enumerate(data['routes'], start=1):

print(f"Route {i}:")

print(f"Distance: {route['legs'][0]['distance']['text']}")

print(f"Duration in current traffic: {route['legs'][0]['duration\_in\_traffic']['text']}")

# Print each step of the route

print("Steps:")

for step in route['legs'][0]['steps']:

print(f"{step['html\_instructions']} ({step['distance']['text']})")

print() # Empty line for readability

else:

print(f"Error: {data['error\_message']}") if 'error\_message' in data else print("Request failed")

except requests.exceptions.RequestException as e:

print(f"Error fetching data: {e}")

# Replace with your Google Maps API key

API\_KEY = 'YOUR\_GOOGLE\_MAPS\_API\_KEY'

# Example usage

origin = input("Enter starting point: ")

destination = input("Enter destination: ")

fetch\_traffic\_updates(API\_KEY, origin, destination)