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

**Scenario:**

**You are developing a real-time traffic monitoring system as part of a smart city initiative. The system aims to provide live traffic updates and recommend alternative routes.**

**Tasks:**

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

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

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

**4. Enable users to input a starting point and destination to receive traffic updates and alternative route suggestions.**

**Deliverables:**

**1.Data flow diagram illustrating the interaction between the application and the API.**

**2.Pseudocode and implementation for the traffic monitoring system.**

**3.Documentation detailing the API integration and the methods used to retrieve and display traffic data.**

**4.Explanation of any assumptions made** **and suggestions for potential improvements.**

**Flow diagram**

+-----------------------------+

| Start |

+-----------------------------+

|

V

+-----------------------------+

| User Inputs: Starting Point,|

| Destination |

+-----------------------------+

|

V

+-----------------------------+

| Request Sent to |

| Traffic API |

+-----------------------------+

|

V

+-----------------------------+

| Traffic Data Received |

+-----------------------------+

|

V

+-----------------------------+

| Process Traffic Data |

+-----------------------------+

|

V

+-----------------------------+

| Display Traffic Conditions, |

| Estimated Travel Time, |

| Incidents |

+-----------------------------+

|

V

+-----------------------------+

| Optionally Display Alternative|

| Routes |

+-----------------------------+

|

V

+-----------------------------+

| End |

**Python Code**

import requests

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

BASE\_URL = 'https://maps.googleapis.com/maps/api/directions/json'

def fetch\_traffic\_data(start, destination):

params = {

'origin': start,

'destination': destination,

'key': API\_KEY,

'departure\_time': 'now'

}

response = requests.get(BASE\_URL, params=params)

data = response.json()

if data['status'] == 'OK':

route = data['routes'][0]

legs = route['legs'][0]

return {

'traffic\_conditions': legs['traffic\_speed\_entry'],

'travel\_time': legs['duration\_in\_traffic']['text'],

'incidents': route['warnings']

}

else:

raise Exception('Error fetching traffic data: ' + data['status'])

def display\_traffic\_data(traffic\_data):

print("Current Traffic Conditions:")

for condition in traffic\_data['traffic\_conditions']:

print(f" - Speed: {condition['speed']} km/h")

print(f"Estimated Travel Time: {traffic\_data['travel\_time']}")

print("Incidents or Delays:")

if traffic\_data['incidents']:

for incident in traffic\_data['incidents']:

print(f" - {incident}")

else:

print(" - No incidents or delays reported.")

def main():

start = input("Enter the starting point: ")

destination = input("Enter the destination: ")

try:

traffic\_data = fetch\_traffic\_data(start, destination)

display\_traffic\_data(traffic\_data)

except Exception as e:

print(f"An error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Pseudocode:**

1. Define constants for the API key and the base URL of the traffic monitoring API.

2. Create a function `fetch\_traffic\_data(start, destination)`:

* Construct the API request URL using the starting and destination points.
* Send a request to the API and receive the response.
* Parse the response to extract traffic data (conditions, travel time, incidents).
* Return the extracted traffic data.

3. Create a function `display\_traffic\_data(traffic\_data)`:

* Print the current traffic conditions.
* Print the estimated travel time.
* Print any incidents or delays.
* Suggest alternative routes if traffic is heavy.

4. Create a function `main()`:

* Prompt the user for a starting point and a destination.
* Call `fetch\_traffic\_data(start, destination)` to retrieve real-time traffic data.
* Call `display\_traffic\_data(traffic\_data)` to show the information.

5. Execute the `main()` function.