import socket

import time

import json

def get\_user\_input():

start\_point = input("Enter starting point: ")

destination = input("Enter destination: ")

return start\_point, destination

def send\_api\_request(start\_point, destination, api\_key):

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

sock.connect(("maps.googleapis.com", 443))

request = f"GET /maps/api/directions/json?origin={start\_point}&destination={destination}&key={api\_key}&traffic\_model=best\_guess HTTP/1.1\r\nHost: maps.googleapis.com\r\n\r\n"

sock.sendall(request.encode())

response = b""

while True:

data = sock.recv(1024)

if not data:

break

response += data

sock.close()

# Split the response into headers and body

response\_parts = response.decode().split("\r\n\r\n", 1)

# Check the response status code

status\_code = int(response\_parts[0].split("\r\n")[0].split(" ")[1])

if status\_code == 200:

return json.loads(response\_parts[1])

else:

raise Exception(f"API request failed with status code {status\_code}")

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

if "routes" in traffic\_data:

for route in traffic\_data["routes"]:

for leg in route["legs"]:

print(f"Route: {leg['start\_address']} to {leg['end\_address']}")

print(f"Estimated Travel Time: {leg['duration']['text']}")

for step in leg["steps"]:

print(f"Step: {step['html\_instructions']}")

if "traffic\_speed\_entry" in step:

print(f"Traffic Speed: {step['traffic\_speed\_entry']['speed']} km/h")

if step['traffic\_speed\_entry']['congestion'] == True:

print("Congestion Detected")

print()

else:

print("Error: Unable to fetch traffic data.")

def main():

api\_key = "YOUR\_API\_KEY"

while True:

start\_point, destination = get\_user\_input()

try:

traffic\_data = send\_api\_request(start\_point, destination, api\_key)

display\_traffic\_data(traffic\_data)

except Exception as e:

print(f"Error: {e}")

print(f"Last updated: {time.strftime('%Y-%m-%d %H:%M:%S')}")

print()

input("Press Enter to continue...")

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

main()