# PYTHON ASSIGNMENT-02

**NAME:** N.HIMA HARSHITHA

**REGISTER N0:** 192371063

**DATE OF SUBMISSION:** 26/08/2024

**TITLE:** REAL_TIME TRAFFIC MONITORING SYSTEM

## Real-Time Traffic Monitoring System

**Scenario:**

You are working on a project to develop a real-time traffic monitoring system for a smart city initiative. The system should provide real-time traffic updates and suggest alternative routes.

**Tasks:**

1. **Model the data flow for fetching real-time traffic information from an external API and displaying it to the user.**
2. **Implement a Python application that integrates with a traffic monitoring API (e.g., Google Maps Traffic API) to fetch real-time traffic data.**
3. **Display current traffic conditions, estimated travel time, and any incidents or delays.**
4. **Allow users to input a starting point and destination to receive traffic updates and alternative routes.**

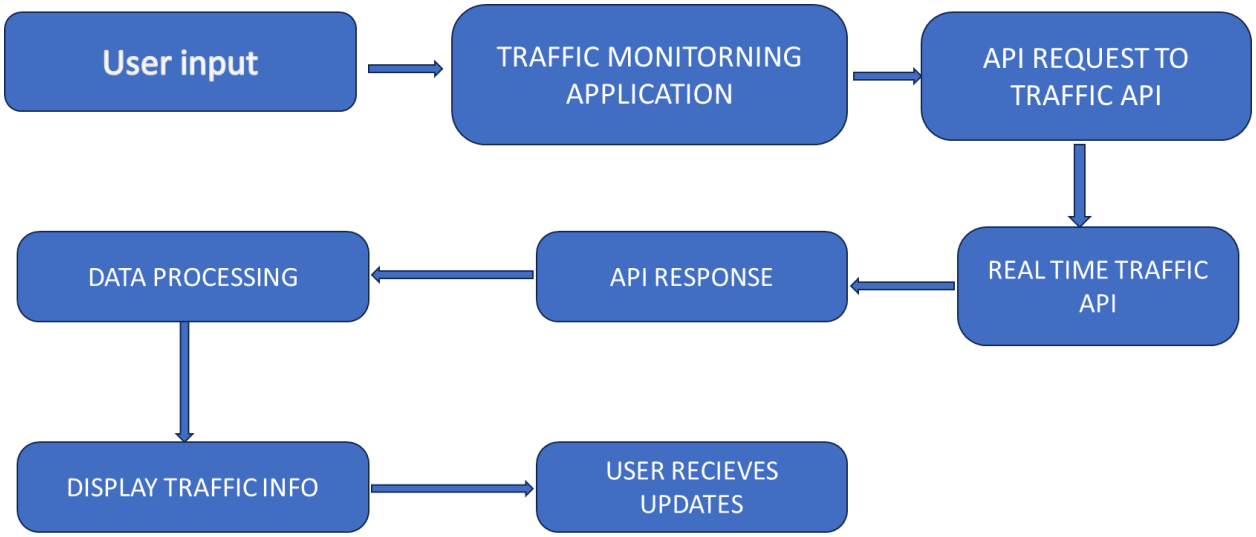
**Deliverables:**

- Data flow diagram illustrating the interaction between the application and the API.
- Pseudocode and implementation of the traffic monitoring system.
- Documentation of the API integration and the methods used to fetch and display traffic data.
- Explanation of any assumptions made and potential improvements

**Solution:**

## Real-Time Traffic Monitoring System

**FLOW CHAT:**




## IMPLEMENTATION:

```
API_KEY = "your_api_key_here"

API_ENDPOINT = "https://maps.googleapis.com/maps/api/directions/json"



def fetch_traffic_data(origin, destination):

    url =
f"{API_ENDPOINT}?origin={origin}&destination={destination}&key={API_KEY}&d
eparture_time=now&traffic_model=best_guess"

    response = requests.get(url)

    if response.status_code == 200:

        traffic_data = response.json()

        return traffic_data

    else:
```

```
        print("Error fetching traffic data. Status code:",
response.status_code)

        return None



def display_traffic_info(traffic_data):

    if traffic_data is not None:

        routes = traffic_data.get("routes", [])

        if routes:

            legs = routes[0].get("legs", [])

            if legs:

                duration_in_traffic = legs[0].get("duration_in_traffic",
{}).get("text", "Not available")

                print(f"Estimated duration in traffic:
{duration_in_traffic}")

            else:

                print("No legs found in the route.")

        else:

            print("No routes found.")

    else:

        print("Failed to fetch traffic data.")



def suggest_alternative_routes(traffic_data):

    if traffic_data is not None:
```

```
        routes = traffic_data.get("routes", [])

        if len(routes) > 1:

            print("Alternative routes:")

            for i in range(1, len(routes)):

                route_summary = routes[i].get("summary", "Route without
summary")

                route_duration = routes[i].get("legs",
[{}])[0].get("duration", {}).get("text", "Not available")

                print(f"- Route {i + 1}: {route_summary}, Estimated
duration: {route_duration}")

        else:

            print("No alternative routes available.")

    else:

        print("Failed to fetch alternative routes.")



def main():

    origin = input("Enter starting point: ")

    destination = input("Enter destination: ")

    traffic_data = fetch_traffic_data(origin, destination)

    if traffic_data is not None:

        display_traffic_info(traffic_data)

        suggest_alternative_routes(traffic_data)

    else:
```

```
        print("Failed to fetch traffic information. Please try again.")



if __name__ == "__main__":

    main()
```

**USER INPUT:**

+ Code + Text

```python
import requests

API_KEY = "your_api_key_here"  # Replace with your actual Google Maps API key
API_ENDPOINT = "https://maps.googleapis.com/maps/api/directions/json"

def fetch_traffic_data(origin, destination):
    url = (f"{API_ENDPOINT}?origin={origin}&destination={destination}&key={API_KEY}"
           "&departure_time=now&traffic_model=best_guess")
    response = requests.get(url)

    if response.status_code == 200:
        return response.json()
    else:
        print(f"Error fetching traffic data. Status code: {response.status_code}")
        return None

def display_traffic_info(traffic_data):
    if traffic_data:
        routes = traffic_data.get("routes", [])
        if routes:
            legs = routes[0].get("legs", [])
            if legs:
                duration_in_traffic = legs[0].get("duration_in_traffic", {}).get("text", "Not available")
                print(f"Estimated duration in traffic: {duration_in_traffic}")
            else:
                print("No legs found in the route.")
```

```python
        else:
            print("No routes found.")
    else:
        print("Failed to fetch traffic data.")

def suggest_alternative_routes(traffic_data):
    if traffic_data:
        routes = traffic_data.get("routes", [])
        if len(routes) > 1:
            print("Alternative routes:")
            for i in range(1, len(routes)):
                route_summary = routes[i].get("summary", "Route without summary")
                route_duration = routes[i].get("legs", [{}])[0].get("duration", {}).get("text", "Not available")
                print(f"- Route {i + 1}: {route_summary}, Estimated duration: {route_duration}")
        else:
            print("No alternative routes available.")
    else:
        print("Failed to fetch alternative routes.")

def main():
    origin = input("Enter starting point: ")
    destination = input("Enter destination: ")

    traffic_data = fetch_traffic_data(origin, destination)

    if traffic_data:
        display_traffic_info(traffic_data)
```

**OUTPUT:**

```
Enter starting point: nellore
Enter destination: naidupeta
No routes found.
No alternative routes available.
```