**Assignment: Python Programming for GUI Development**

Name: Dhanajeyan J

Register Number: 192324211

Department: AI&DS

Date of Submission: 26-08-2024

**Problem 1:** **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.**

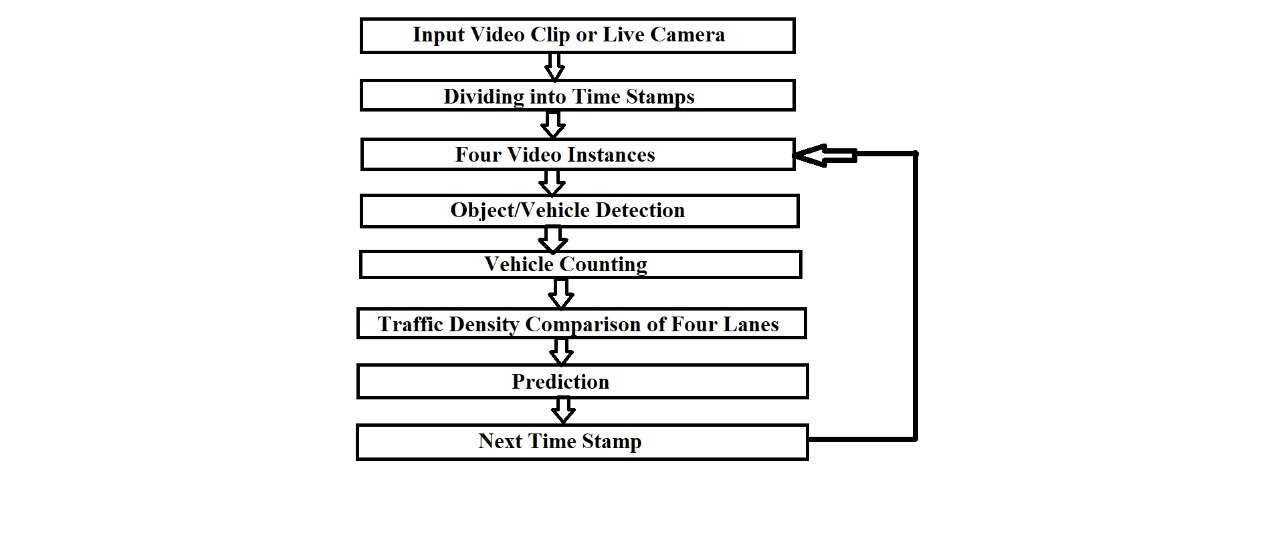
**Deliverables:**

* 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

# 1.Data Flow Diagram:



# 2. Implementation

|  |
| --- |
| START  Initialize the application  User inputs starting point and destination  Send request to traffic monitoring API with starting point and destination  IF request successful THEN  Receive traffic data response  Parse the response to extract:  - Current traffic conditions  - Estimated travel time  - Incidents or delays    Display the traffic data to the user    IF alternative routes available THEN  Calculate alternative routes  Display alternative routes with updated traffic conditions and travel time  END IF  ELSE  Display error message to user  END IF  END |

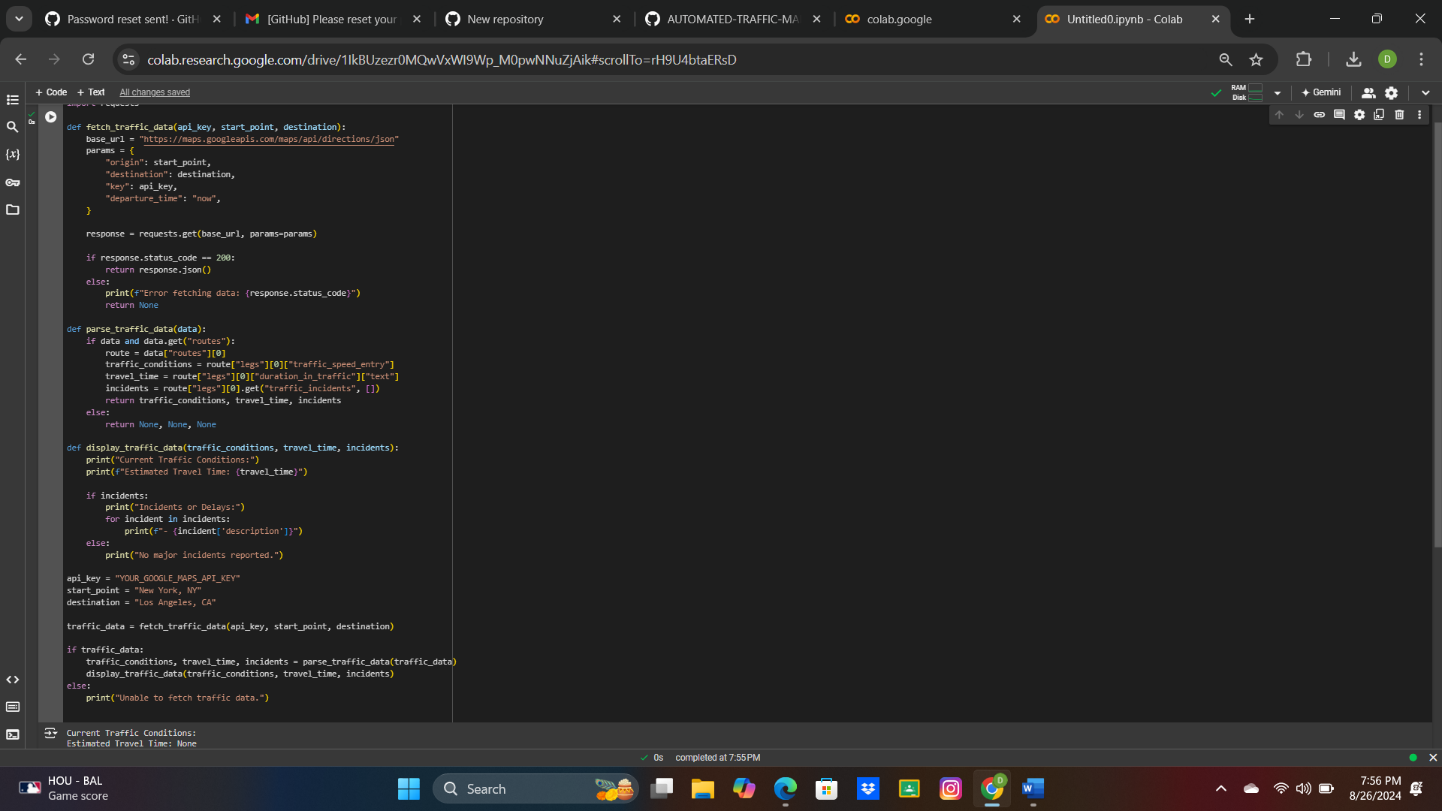
# 3.Display the Current Traffic information

**Current Traffic Conditions:**

**Estimated Travel Time: None**

**No major incidents reported.**

# 4.User Input



**5.Documentation**

**Algorithm Explanation:**

The algorithm begins by initializing the application and prompting the user to input a starting point and destination. It then sends a request to a real-time traffic monitoring API with these inputs to fetch current traffic data. If the request is successful, the algorithm extracts and processes key information such as current traffic conditions, estimated travel time, and any incidents or delays. This data is then displayed to the user, along with any available alternative routes. If errors occur during the API request, an error message is shown. Finally, the process concludes after displaying all necessary information to the user.

Assumptions Made During **Real-Time Traffic Monitoring System:**

The following assumptions were made in designing the real-time traffic monitoring system.

1. **Valid API Access**: The user has a valid API key to access the traffic monitoring service (e.g., Google Maps Traffic API).
2. **User Input**: The starting point and destination provided by the user are valid and recognized by the API.
3. **API Response**: The API is functioning correctly, providing accurate and real-time traffic data without significant delays or downtimes.
4. **Network Connectivity**: The system has stable internet connectivity to send requests and receive responses from the external API.
5. **Real-Time Data**: The traffic monitoring API provides reliable real-time data, reflecting current traffic conditions accurately.
6. **Default Mode of Transport**: The application assumes the user is traveling by car unless otherwise specified (e.g., for walking or biking directions).
7. **Simplified Error Handling**: The system only handles basic errors, such as failed API requests, and does not account for more complex scenarios like partial data retrieval.
8. **Basic User Interface**: The application has a text-based interface, and no complex graphical user interface (GUI) is implemented.

**Conclusion:**

The real-time traffic monitoring system is designed to provide users with up-to-date traffic conditions, estimated travel times, and potential incidents or delays based on real-time data from an external API. The system allows users to input their starting point and destination and suggests alternative routes when possible, enhancing route planning. While the basic implementation meets the core requirements, future improvements could include better error handling, a more interactive user interface, and expanded functionality for different modes of transportation. This project demonstrates the integration of real-time data into a user-centric application for smarter city traffic management.