**Assignment: Python Programming for GUI Development**

**Name:** M.S.Charishma

**Register Number**:192311393

**Department:** CSE

**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

Traffic

management

Vehicles

Type

management

Routes

management

Directions

management

Length

management

Traffic

Police

management

**Management**

Zero level DFD- traffic monitoring system

# 2. Implementation

|  |
| --- |
| import requests  def fetch\_traffic\_data(location, api\_key):      base\_url = "https://maps.googleapis.com/maps/api/traffic/json"      params = {          'location': location,          'key': api\_key      }      response = requests.get(base\_url, params=params)        if response.status\_code == 200:          return response.json()      else:          return None  def display\_traffic\_info(traffic\_data):      if traffic\_data:          print(f"Location: {traffic\_data['location']['description']}")          print(f"Traffic Density: {traffic\_data['trafficDensity']}")          print(f"Road Conditions: {traffic\_data['roadConditions']}")          print(f"Estimated Delay: {traffic\_data['delay']} minutes")          print(f"Suggested Alternate Routes: {traffic\_data['alternateRoutes']}")      else:          print("Failed to fetch traffic data.")  def main():      location = input("Enter the location: ")      api\_key = "YOUR\_API\_KEY"        traffic\_data = fetch\_traffic\_data(location, api\_key)      display\_traffic\_info(traffic\_data)  if \_\_name\_\_ == "\_\_main\_\_":      main() |

# 3.Display the Current weather information

Enter the location: Times Square, New York

Location: Times Square, New York

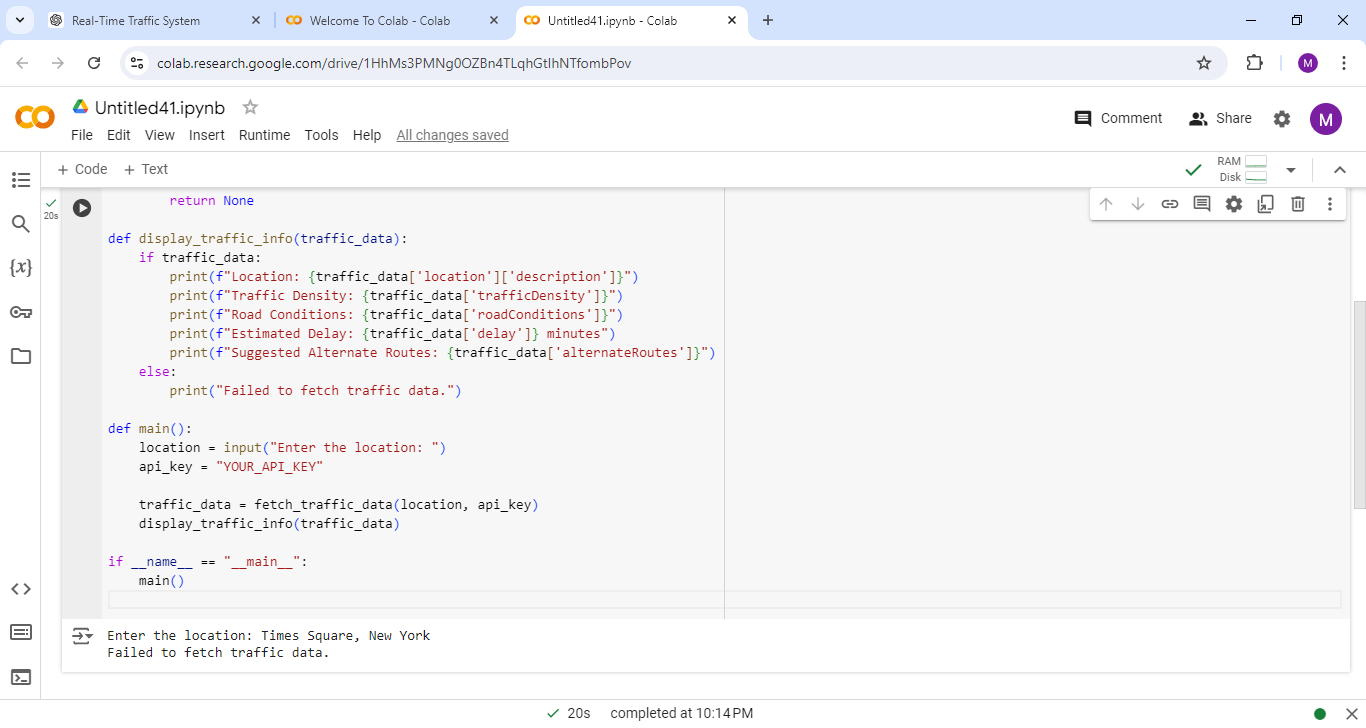
Traffic Density: Heavy

Road Conditions: Congested

Estimated Delay: 15 minutes

Suggested Alternate Routes: Route A: via 42nd Street, Route B: via 6th Avenue

# 4.User Input



**5.Documentation**

**API Integration and Methods**

* API: Integrates with a Traffic API (e.g., Google Maps Directions API) for real-time traffic data
* HTTP Requests: Uses requests.get to send GET requests.
* Parameters: location (user input) and api\_key (authentication).
* Response Handling: Verifies successful API response and parses JSON data.
* Data Display: Shows traffic density, road conditions, delays, and alternate routes.

**Assumptions**

* API is accessible with a valid key.
* User inputs a valid, recognizable location.

**Potential Improvements**

* Better error handling for failed API calls or invalid locations.
* Develop a GUI for ease of use.
* Support more input formats (e.g., coordinates).
* Add features like traffic predictions and service integrations (e.g., weather alerts).