

PHASE IV PROJECT

PROJECT TITLE : PUBLIC TRANSPORT OPTIMIZATION
NAME : Bush Kaladevi.S
COLLEGE CODE : 9530
Reg No : 953021106013
COLLEGE NAME : ST.MOTHER THERESA ENGINEERING COLLEGE
TEAM NAME : proj 201030 Team_1

SOURCE CODE :

```
import time

import serial

import gpsd

from gsmmodem import GsmModem


# Define serial port for GPS communication

gps_serial = serial.Serial('/dev/ttyUSB0', 9600)


# Initialize GSM modem

modem = GsmModem(port='/dev/ttyUSB1', baudrate=9600)

modem.connect('<your_pin>', 'your_gsm_device')


# Function to send an SMS

def send_sms(message, recipient):

    modem.sendSms(recipient, message)


# Function to get GPS coordinates

def get_gps_coordinates():

    try:
```

```

packet = gpsd.get_current()
if packet.mode >= 2:
    return packet.lat, packet.lon
else:
    return None, None
except Exception as e:
    print(f"Error reading GPS data: {e}")
    return None, None

# Main loop
while True:
    try:
        command = input("Enter a command: ")
        if command == "Track Vehicle":
            lat, lon = get_gps_coordinates()
            if lat is not None and lon is not None:
                message = f"Vehicle Tracking Alert:\nYour Vehicle Current Location is:\nLatitude:
{lat:.6f}\nLongitude: {lon:.6f}\nGoogle Maps Link:
https://www.google.com/maps/@{lat},{lon},14z"
                recipient = '850xxxxxxx' # Replace with the actual phone number
                send_sms(message, recipient)
                print("SMS Sent")
            else:
                print("No GPS Fix")
        else:
            print("Invalid command")
    except KeyboardInterrupt:
        print("Exiting")
        break

```

```
# Disconnect GSM modem
```

```
modem.close()
```

SOURCE CODE

```
mapboxgl.accessToken = 'YOUR_MAPBOX_ACCESS_TOKEN'; // Replace with your Mapbox  
access token
```

```
const map = new mapboxgl.Map({  
  container: 'map',  
  style: 'mapbox://styles/mapbox/streets-v11', // You can use a different Mapbox style  
  center: [-73.981915, 40.747766], // Initial map center (longitude, latitude)  
  zoom: 12, // Initial zoom level  
});
```

```
const busMarker = new mapboxgl.Marker()  
  .setLngLat([-73.981915, 40.747766]) // Initial bus location (longitude, latitude)  
  .addTo(map);
```

```
// Set up a function to update bus location (you would replace this with real-time data)
```

```
function updateBusLocation() {  
  // Replace with code to fetch real-time bus location data  
  // For this example, we're simulating a moving bus  
  const newLocation = simulateBusMovement(busMarker.getLngLat());  
  busMarker.setLngLat(newLocation);  
  requestAnimationFrame(updateBusLocation);  
}
```

```
// Function to simulate bus movement (replace this with real real-time data)
```

```
function simulateBusMovement(currentLocation) {
```

```
// In this example, we're simulating bus movement by incrementing longitude
const newLongitude = currentLocation.lng + 0.0001;
return [newLongitude, currentLocation.lat];
}
```

```
updateBusLocation(); // Start updating the bus location
```

SOURCE CODE

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8">

  <title>Real-Time Bus Tracking</title>

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://api.mapbox.com/mapbox-gl-js/v2.6.1/mapbox-gl.css" rel="stylesheet">

  <script src="https://api.mapbox.com/mapbox-gl-js/v2.6.1/mapbox-gl.js"></script>

  <style>

    body {

      margin: 0;

      padding: 0;

    }

    #map {

      height: 100vh;

      width: 100%;

    }

  </style>

</head>

<body>
```

```
<div id="map"></div>
```

```
<script src="bus-tracking.js"></script>
```

```
</body>
```

```
</html>
```

OUTPUT:

