PUBLIC TRANSPORT OPTIMIZATION USING IOT

Phase 5 Submission document

1.ABSTRACT:

The Public Transport Optimization Project aims to revolutionize urban mobility by deploying cutting-edge technologies and data-driven strategies. This project focuses on optimizing public transportation systems to be more reliable, efficient and environmentally friendly. By leveraging advanced data analytics, machine learning algorithms, and real-time tracking solutions, the project intends to enhance the overall commuter experience, reduce traffic congestion, and mitigate the environmental impact of urban transportation.

2.INTRODUCTION:

2.1 PROJECT OVERVIEW:

Data Collection and Analysis: Gathering data on passenger demand, travel patterns, routes, and schedules is crucial. Advanced technologies like GPS, sensors, and passenger counting systems are used to collect this data.

Route Optimization: Analyzing existing routes and determining the most efficient paths based on demand, traffic patterns, and geographical factors. This may involve restructuring or consolidating routes to minimize travel time and maximize coverage.

Schedule Optimization: Adjusting schedules to match peak demand periods, ensuring buses or trains are available when and where they are needed the most. Real-time scheduling adjustments can also be implemented based on live data.

Fleet Management: Optimizing the size and composition of the transportation fleet. This involves assessing the types of vehicles needed, their capacities, and ensuring maintenance schedules are optimized to minimize downtime.

Technological Integration: Implementing smart technologies like real-time tracking, mobile apps, and digital payment systems to enhance passenger experience and provide up-to-date information.

Cost-Benefit Analysis: Evaluating the costs associated with implementing optimization strategies against the benefits, such as reduced travel time, increased ridership, and environmental savings.

2.2 PURPOSE:

Reducing Congestion: By optimizing routes, schedules, and capacities, public transport can reduce congestion on roads, making it a more attractive option for commuters.

Enhancing Accessibility: Public transport optimization can ensure that transportation services reach underserved or remote areas, providing mobility options to a broader population.

Cost Efficiency: Optimizing routes and schedules can help reduce operating costs, making public transport more financially sustainable and potentially leading to lower fares for passengers.

Environmental Benefits: By optimizing routes and encouraging the use of cleaner and more efficient vehicles, public transport can contribute to a reduction in greenhouse gas emissions and air pollution.

Improved Service Quality: Optimization can lead to more reliable and punctual services, making public transport a more reliable mode of transportation for users.

Economic Development: Well-optimized public transport can stimulate economic development by increasing access to job opportunities, education, and services.

Equity and Inclusivity: Public transport optimization should ensure that it serves all segments of the population, including those with limited mobility or financial resources.

3. LITERATURE SURVEY:

3.1 EXISTING PROBLEM:

Inefficiency and Delays: Public transport systems often face issues with inefficiency and delays due to outdated schedules, inadequate routes, and congestion, leading to longer travel times for passengers.

Overcrowding: Buses, trains, and trams can become overcrowded during peak hours, leading to discomfort for passengers and potential safety concerns.

Optimizing routes and schedules can help manage passenger loads more effectively.

Environmental Impact: Public transport systems often rely on fossil fuels, contributing to air pollution and greenhouse gas emissions. Transitioning to eco-friendly options like electric or hybrid vehicles is a concern for many cities aiming to reduce their environmental footprint.

Data Utilization: Public transport agencies might not be effectively using available data to optimize routes and schedules. Lack of data-driven decision-making can lead to suboptimal services.

3.2 REFERENCE

1. "Internet of Things in Transportation" by Paulo André da Conceição Menezes: This book provides insights into IoT applications in transportation, including public transport optimization.

2. "Smart Public Transportation: A Real-Time Bus Tracking System Using IoT" by A. Shijila and M. Mohamed Sathik: This research paper discusses the implementation of an IoT-based real-time bus tracking system to enhance public transportation.

3. "Smart Public Transportation System Using IoT and Big Data Analytics" by T. T. Kiran and R. N. Raja Kumar: This paper explores the use of IoT and big data analytics for smart public transportation systems.

4. "Intelligent Transportation Systems (ITS) and Internet of Things (IoT): Optimizing Traffic Management" by Siddhartha S. Ray: This book chapter discusses how IoT can be integrated into intelligent transportation systems to optimize traffic and public transportation.

5. "Smart Cities and the Internet of Things" by Ovidiu Vermesan and Peter Friess: This book includes chapters on IoT applications in urban transportation and how they contribute to smart city initiatives.

6. "Real-Time Passenger Information Systems for Public Transport Using IoT and Cloud" by Xun Zhou, Ruili Wang, and Xueping Xu: This research paper focuses on real-time passenger information systems using IoT and cloud computing for public transportation.

7. "IoT-Enabled Smart Public Transportation: Opportunities and Challenges" by Tauseef Ahmad, et al.: This paper discusses the opportunities and challenges in implementing IoT for smart public transportation.

8. "IoT Applications in Smart Cities: A Case Study of Smart Public Transportation" by Tauseef Ahmad and Shahzad Sarwar: This paper provides a case study of IoT applications in smart public transportation within a smart city context.

3.3 PROBLEM DESCRIPTION:

Integrate IoT sensors in public transportation vehicles to monitor ridership, track locations, and predict bus or train arrival times. This data can be shared on a public platform to improve transit services.

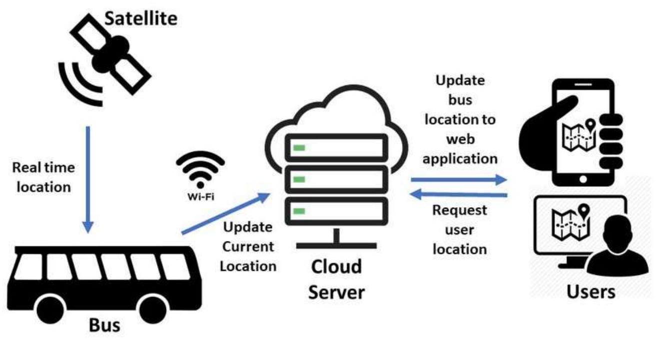
4. IDEATION:

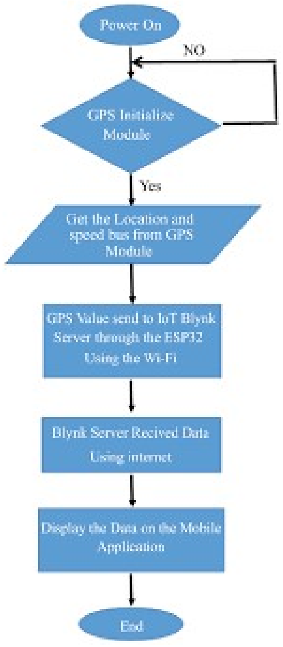
4.1 PROPOSED SOLUTION:

Using sensors such as GPS sensor to monitor location of each vehicle. This information is then fed to Arduino and then uploaded to cloud using an esp8266 module. This information is then used to monitor the locations of the public vehicle. Scheduling public transport and making this information available to the users.

5. PROJECT DESIGN:

5.1 FLOW CHART




Power On
NO
GPS Initialize Module
Yes
Get the Location and speed bus from GPS Module
GPS Value send to IoT Blynk Server through the ESP32 Using the Wi-Fi
Blynk Server Recived Data Using internet
Display the Data on the Mobile Application
End

6. TECHNICAL ARCHITECTURE:

Data Collection and Sensors:

- Utilize GPS, RFID, or other sensors on vehicles to track their real-time location.
- Collect data on passenger counts, traffic conditions, and weather.

Data Storage and Processing:

- Store the collected data in a robust database system, like SQL or NoSQL databases.
- Implement a data processing layer for real-time and batch data analytics.

Routing and Scheduling Algorithms:

- Develop algorithms for optimizing routes, schedules, and vehicle assignments.
- Consider factors like demand patterns, traffic congestion, and passenger preferences.

GIS (Geographic Information System):

- Use GIS data for mapping and geospatial analysis to optimize routes and stops.
- Integrate with external mapping services for real-time navigation.

Communication and Control Center:

- Implement a control center for monitoring and managing the transportation system.
- Enable real-time communication with drivers and passengers through mobile apps or other channels.

Fare Collection and Payment Systems:

- Integrate electronic fare collection systems, such as contactless cards or mobile payments.
- Ensure secure and efficient payment processing.

Predictive Analytics:

- Apply predictive models to anticipate demand, delays, and other operational factors.
- Adjust routes and schedules dynamically based on predictions.

Passenger Information Systems:

- Provide real-time information to passengers about vehicle locations, arrival times, and service updates.
- Develop mobile apps, websites, and digital signage for passenger communication.

IoT and Connectivity:

- Use IoT devices for vehicle monitoring and maintenance.
- Ensure reliable connectivity for data exchange between vehicles and the central system.

Security and Safety:

- Implement security measures to protect passenger data and system integrity.
- Ensure safety features for passengers and drivers.

Feedback and Reporting:

- Set up mechanisms for passengers and staff to report issues or provide feedback.
- Use this data to continuously improve the system.

Scalability and Redundancy:

- Design the architecture to scale with growing demand.
- Implement redundancy and failover mechanisms for high availability.

APIs and Integration:

- Provide APIs for third-party developers to build apps and services on top of the public transport system.
- Integrate with other transportation services, such as ride-sharing and bike-sharing.

Environmental Considerations:

- Optimize routes and schedules to reduce emissions and fuel consumption.
- Consider eco-friendly vehicle options, like electric buses.

Regulatory Compliance:

- Ensure compliance with local and national regulations related to public transportation.

7. CODING AND SOLUTION:

7.1 PYTHON CODE:

```
import time

from adafruit_bme280 import Adafruit_BME280

 import board

 import busio

 from adafruit_ads1x15.analog_in import AnalogIn
from adafruit_tsl2591 import TSL2591
 from adafruit_pn532 import PN532
 from digitalio import DigitalInOut, Direction
import adafruit_dht
 from pyblynk import Blynk
from tinygps import TinyGPS
# BME280 Environmental Sensor
 i2c = busio.I2C(board.SCL, board.SDA)
bme = Adafruit_BME280(i2c)
 # ADS1115 ADC
 ads = AnalogIn(board.A0)
# TSL2591 Light Sensor
 tsl = TSL2591()
# PN532 NFC Module
 pn532 = PN532(board.SDA, board.SCL)
 # DHT Sensor
dht_sensor = adafruit_dht.DHT22(board.D2)
# GPS Pins
gps_tx_pin = board.TX
gps_rx_pin = board.RX
gps_serial = TinyGPS()
# Blynk Setup
 BLYNK_AUTH = "Your_Blynk_Auth_Token"
```

```
blynk = Blynk(BLYNK_AUTH)

# Motion Sensor Pin

 motion_sensor_pin = DigitalInOut(board.D3)

motion_sensor_pin.direction = Direction.INPUT

 @blynk.VIRTUAL_WRITE(1)

 def bme280_data(pin):

 temperature = bme.temperature

humidity = bme.humidity

 pressure = bme.pressure / 100.0

 blynk.virtual_write(1, temperature)

blynk.virtual_write(2, humidity)

 blynk.virtual_write(3, pressure)

@blynk.VIRTUAL_WRITE(4)

def ads1115_data(pin):

 voltage = ads.value * (5.0 / 32767.0)

blynk.virtual_write(4, voltage)

@blynk.VIRTUAL_WRITE(5)

 def tsl2591_data(pin):

 luminosity = tsl.lux

desired_light_level = int(map_range(luminosity, 0, 5000, 0, 255))
blynk.virtual_write(5, desired_light_level)

 @blynk.VIRTUAL_WRITE(6)

def nfc_data(pin):

uid = pn532.read_passive_target()
```

```
 if uid:
card_uid = ''.join([format(byte, '02X') for byte in uid])
 process_card(card_uid)
 time.sleep(1)
 def process_card(card_uid):
 blynk.virtual_write(6, card_uid)
@blynk.VIRTUAL_WRITE(7)
def dht_data(pin):
temperature = dht_sensor.temperature
 humidity = dht_sensor.humidity
@blynk.VIRTUAL_WRITE(9)
def gps_data(pin):
sentence = gps_serial.read()
 if sentence:
if sentence.startswith(b'$GPGGA'):
lat, lon, _ = gps_serial.parse_gga(sentence)
blynk.virtual_write(9, lat)
blynk.virtual_write(10, lon)
 @blynk.VIRTUAL_WRITE(11)
def motion_sensor_data(pin):
motion_value = motion_sensor_pin.value
if motion_value == True:
blynk.notify("Motion detected!")
```

```
 def map_range(x, in_min, in_max, out_min, out_max):
 return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min
 while True:
 blynk.run()
time.sleep(1)
```

7.2 ARDUINO PROGRAM :

```
#include<Wire.h>
 #include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
 #include <Adafruit_ADS1015.H>
#include <Adafruit_PN532.h>
 #include <TinyGPS++.h>
 #include <DTH.h>
#include <WiFi.h>
#include<BlynkSimpleEsp32.h>
 char auth[] = "Your_Blynk_Auth_Token";
 char ssid[] = "Your_WiFi_SSID";
char pass[] = "Your_WiFi_Password";
// BME280 Environmental Sensor
Adafruit_BME280 bme;
// ADS1115 ADC
Adafruit_ADS1115 ads;
 const int fuelSensorPin = A0;
 // Motion Sensor

int motionSensorPin = 14;

 // GPS

TinyGPSPlus gps;

#define GPS_TX_PIN 17

#define GPS_RX_PIN 16

// DHT Sensor

 #define DHTPIN 2

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// NFC Module

 Adafruit_PN532 nfc(SDA_PIN, SCL_PIN);

 // Panic Button

 #define PANIC_BUTTON_PIN 2

 // Congestion Sensor

#define SENSOR_PIN 15

#define LED_PIN 13 BlynkTimer timer;

 void setup() {

Serial.begin(115200);

 Blynk.begin(auth, ssid, pass);

 if (!bme.begin(0x76)) {

 Serial.println("Could not find a valid BME280 sensor, check wiring!");

while (1);
```

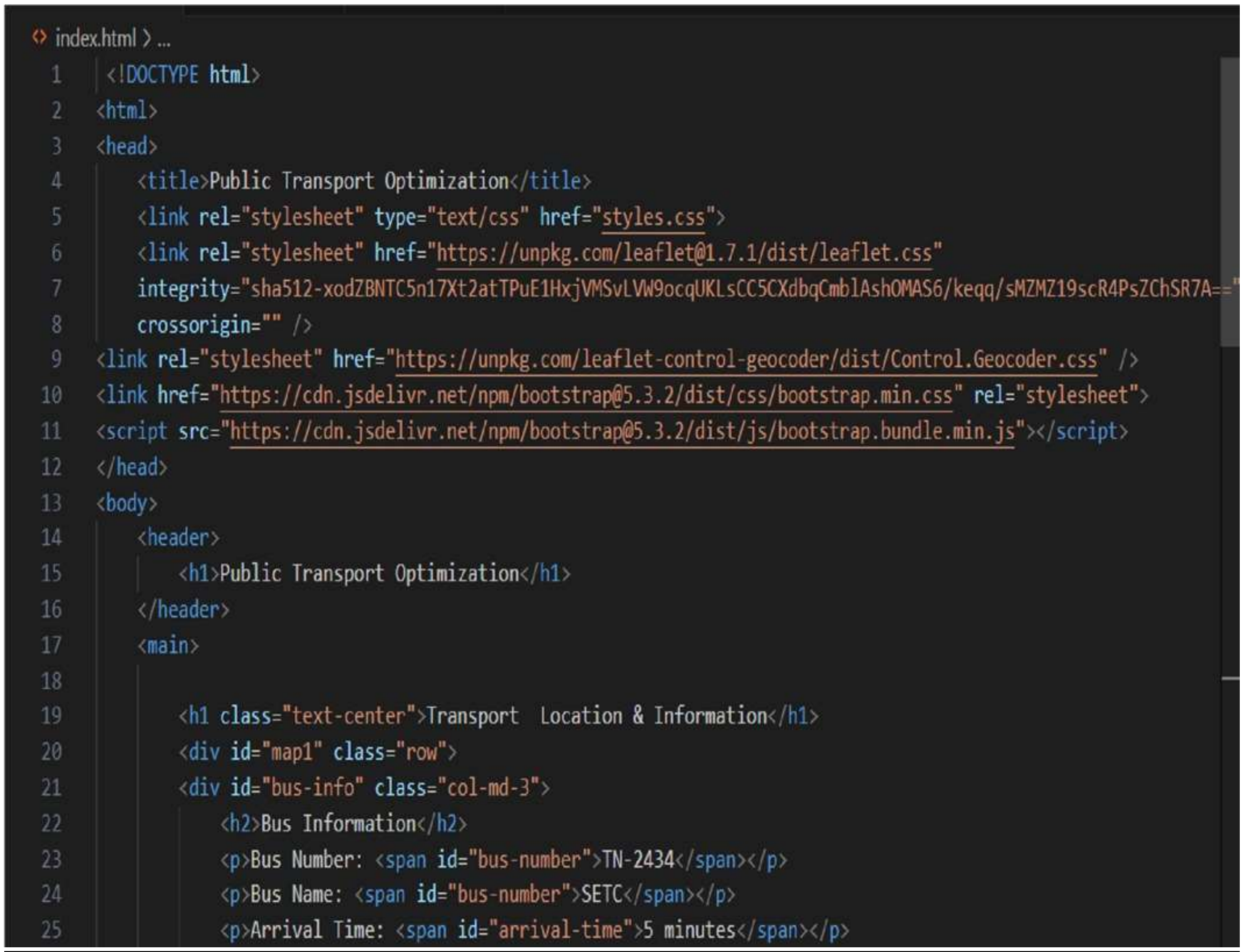
```
 }
 if (!ads.begin()) {
 Serial.println("Failed to initialize ADS1115");
 while (1);
}
pinMode(fuelSensorPin, INPUT);
pinMode(motionSensorPin, INPUT);
pinMode(PANIC_BUTTON_PIN, INPUT_PULLUP);
pinMode(SENSOR_PIN, INPUT);
pinMode(LED_PIN, OUTPUT);
Serial1.begin(9600, SERIAL_8N1, GPS_RX_PIN, GPS_TX_PIN);
 nfc.begin();
 uint32_t versiondata = nfc.getFirmwareVersion();
 if (!versiondata) {
Serial.print("Didn't find PN53x board");
while (1);
 }
nfc.SAMConfig();
dht.begin();
timer.setInterval(1000L, checkMotionSensor);
}
void loop() {
Blynk.run();
```

```
timer.run();
while (Serial1.available() > 0) {
if (gps.encode(Serial1.read())) {
 displayInfo();
 }
 }
 if (nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
&uidLength)) {
String cardUID = "";
for (uint8_t i = 0; i < uidLength; i++) {
cardUID += String(uid[i], HEX);
 }
processCard(cardUID);
}
Blynk.run();
int sensorValue = digitalRead(SENSOR_PIN);
 if (sensorValue == HIGH) {
digitalWrite(LED_PIN, HIGH);
} else {
 digitalWrite(LED_PIN, LOW);
 }
float temperature = dht.readTemperature();
 float humidity = dht.readHumidity();
```

```
uint16_t luminosity = tsl.getLuminosity(TSL2591_VISIBLE);

int desiredLightLevel = map(luminosity, 0, 5000, 0, 255);

Blynk.virtualWrite(V1, temperature);

 Blynk.virtualWrite(V2, humidity);

 Blynk.virtualWrite(V3, pressure);

 Blynk.virtualWrite(V4, motionValue);

 Blynk.virtualWrite(V5, voltage);

Blynk.virtualWrite(V6, cardUID);

Blynk.virtualWrite(V7, desiredLightLevel);

if (temperature >= 25.0) {

Blynk.virtualWrite(V8, HIGH);

 } else if (temperature <= 20.0) {

Blynk.virtualWrite(V8, LOW);

 } else {

 Blynk.virtualWrite(V8, LOW);

 }

} void displayInfo() {

 if (gps.location.isValid()) {

 Serial.print("Latitude: ");

Serial.println(gps.location.lat(), 6);

Serial.print("Longitude: ");

Serial.println(gps.location.lng(), 6);

 Blynk.virtualWrite(V1, gps.location.lat());
 Blynk.virtualWrite(V2, gps.location.lng());
 } else {
 Serial.println("Invalid GPS location");
}
 } void processCard(String cardUID) {
Blynk.virtualWrite(V9, cardUID);
}
void checkMotionSensor() {
int motionValue = digitalRead(motionSensorPin);
 if (motionValue == HIGH) {
Blynk.notify("Motion detected in public transport!");
}
 }
```

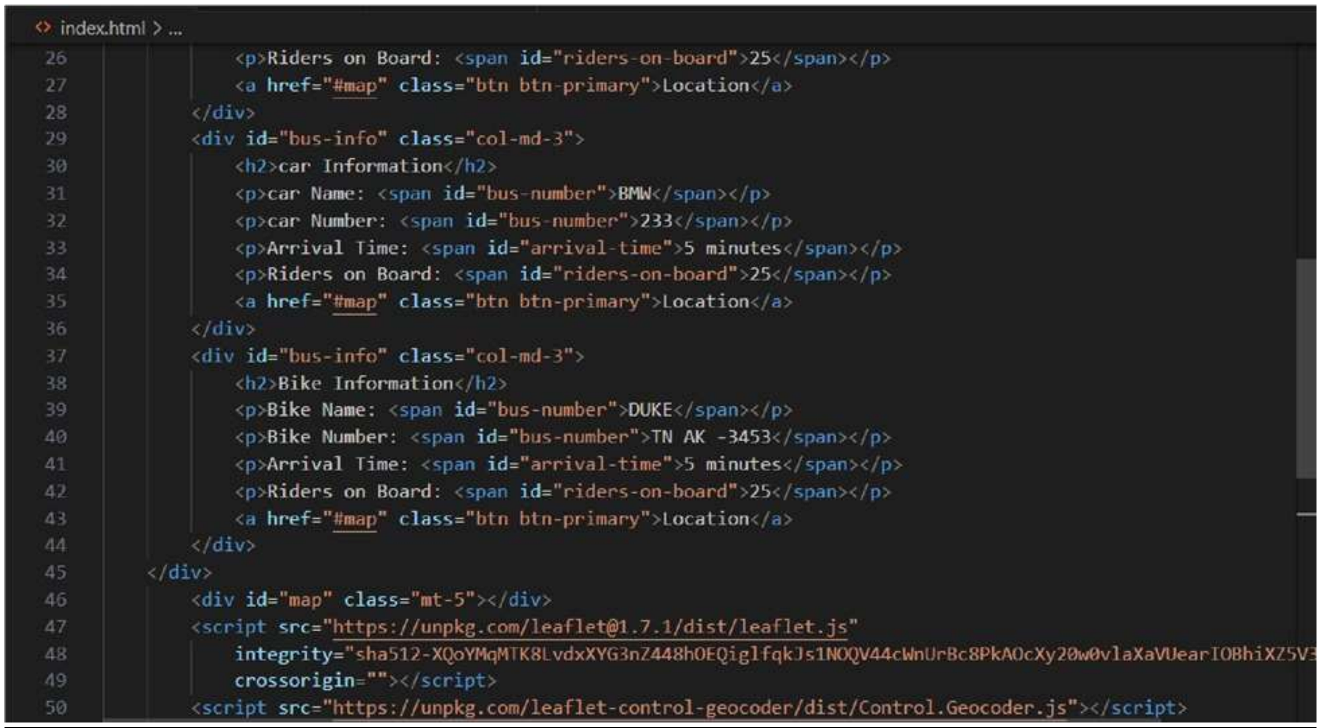
<u>Here we build the project by developing the real time transit information platform</u>

<u>INDEX.html</u>



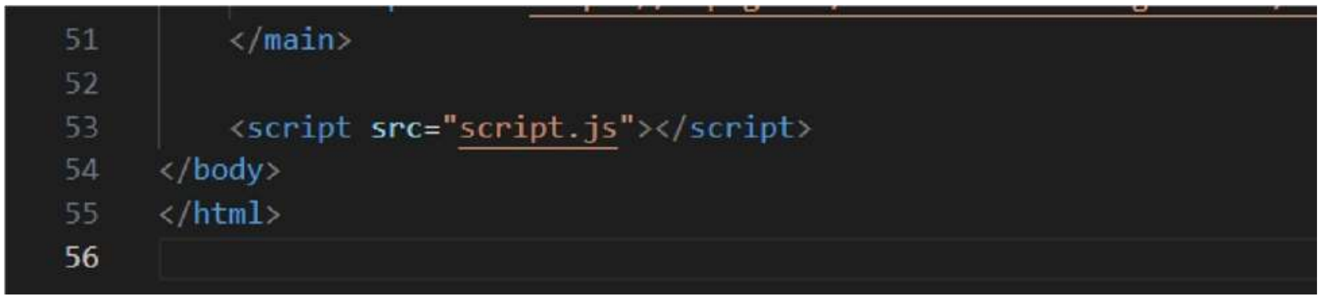
index.html > ...

```html
<!DOCTYPE html>
<html>
<head>
    <title>Public Transport Optimization</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
    integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A=="
    crossorigin="" />
<link rel="stylesheet" href="https://unpkg.com/leaflet-control-geocoder/dist/Control.Geocoder.css" />
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
    <header>
        <h1>Public Transport Optimization</h1>
    </header>
    <main>

        <h1 class="text-center">Transport  Location & Information</h1>
        <div id="map1" class="row">
        <div id="bus-info" class="col-md-3">
            <h2>Bus Information</h2>
            <p>Bus Number: <span id="bus-number">TN-2434</span></p>
            <p>Bus Name: <span id="bus-number">SETC</span></p>
            <p>Arrival Time: <span id="arrival-time">5 minutes</span></p>
```



index.html > ...

```html
            <p>Riders on Board: <span id="riders-on-board">25</span></p>
            <a href="#map" class="btn btn-primary">Location</a>
        </div>
        <div id="bus-info" class="col-md-3">
            <h2>car Information</h2>
            <p>car Name: <span id="bus-number">BMW</span></p>
            <p>car Number: <span id="bus-number">233</span></p>
            <p>Arrival Time: <span id="arrival-time">5 minutes</span></p>
            <p>Riders on Board: <span id="riders-on-board">25</span></p>
            <a href="#map" class="btn btn-primary">Location</a>
        </div>
        <div id="bus-info" class="col-md-3">
            <h2>Bike Information</h2>
            <p>Bike Name: <span id="bus-number">DUKE</span></p>
            <p>Bike Number: <span id="bus-number">TN AK -3453</span></p>
            <p>Arrival Time: <span id="arrival-time">5 minutes</span></p>
            <p>Riders on Board: <span id="riders-on-board">25</span></p>
            <a href="#map" class="btn btn-primary">Location</a>
        </div>
    </div>
        <div id="map" class="mt-5"></div>
        <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
            integrity="sha512-XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vlaXaVUearIOBhiXZ5V3
            crossorigin=""></script>
        <script src="https://unpkg.com/leaflet-control-geocoder/dist/Control.Geocoder.js"></script>
```



```
    </main>

    <script src="script.js"></script>
</body>
</html>

```

1. <!DOCTYPE html>: This declaration specifies that the document follows HTML5 standards.

2. <html>: The root element of an HTML document.

3. <head>: This section contains metadata and links to external resources. In this case, it links to an external CSS file and sets the document's character encoding.

4. <meta charset="UTF-8">: Specifies the character encoding of the document as UTF-8,

which is a widely used encoding for handling text in different languages.

5. <title>Public Transport Optimization Platform</title>: Sets the title of the web page, which appears in the browser's title bar or tab.

6. <link rel="stylesheet" type="text/css" href="styles.css">: Links an external CSS file named "styles.css" to style the web page.

7. <body>: The main content of the web page is contained within the <body> element.

8. <header>: The header section typically contains the title or logo of the website. In this case, it displays the title "Public transport optimization."

9. <nav>: This section contains navigation links. It's structured as an unordered list <ul>with list items <li>, each of which contains an anchor <a> element for navigation.

10.<p align="center">: This paragraph element aligns its text to the center. However, the align attribute is deprecated in HTML5, and it's recommended to use CSS for text alignment.

11.The <p> element provides information about environmental monitoring, its purpose, and what it encompasses. It's a description of the environmental monitoring concept.

12.<section id="updates">: This section is labeled "Latest Updates" and contains an unordered list <ul> of updates. Each update is presented as a list item <li> with a date and a description.

13.<main>: The main content of the web page, which contains various sections related to real-time data, data visualization, and information about the platform.

14.<section class="sensor-data">: This section is dedicated to displaying real-time sensor data. It contains two data items, "Temperature" and "Humidity," each with a heading

<h3> and a placeholder paragraph <p> for data to be loaded via JavaScript.

15.<section class="data-visualization">: This section is for data visualization. It currently includes a canvas element <canvas> with the ID "chart" where data visualization elements like charts or graphs can be added. Additionally, it includes a script to include the Chart.js library for creating charts and another canvas element with the ID "lineChart."

16.<section id="about">: This section provides information about the platform, its mission, and what it offers.

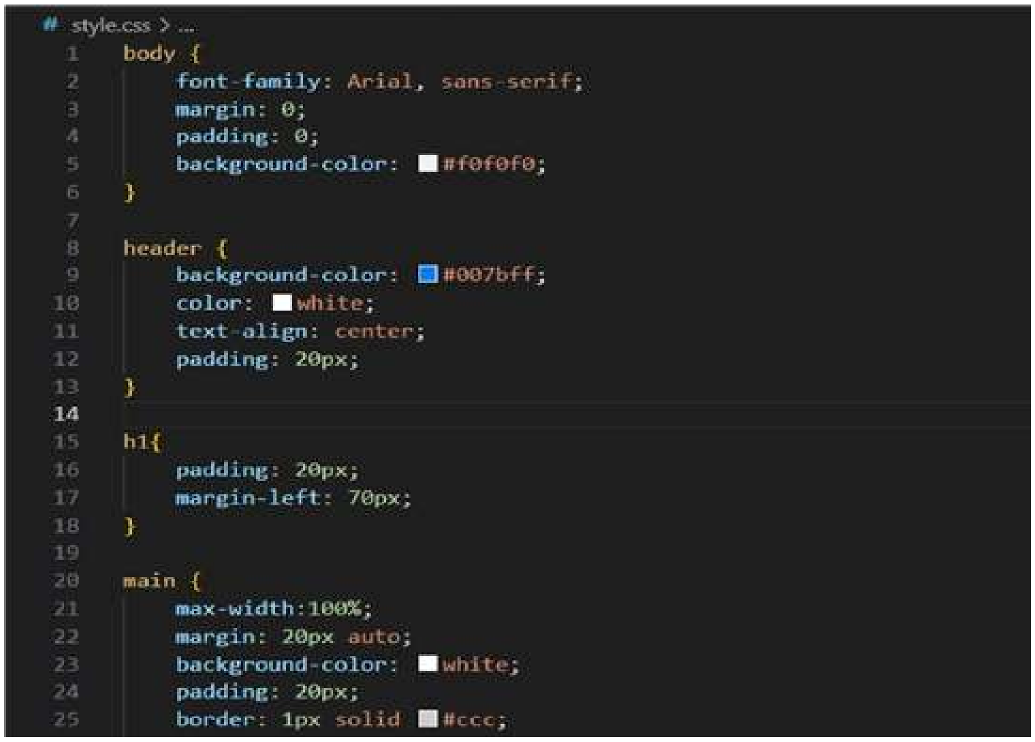
17.<section id="contact">: This section offers contact information, including an email address where users can reach out with questions or feedback.

18.<section id="disclaimer">: This section includes a disclaimer regarding the informational nature of the platform and advises users to consult with experts for critical decisions related to the environment.

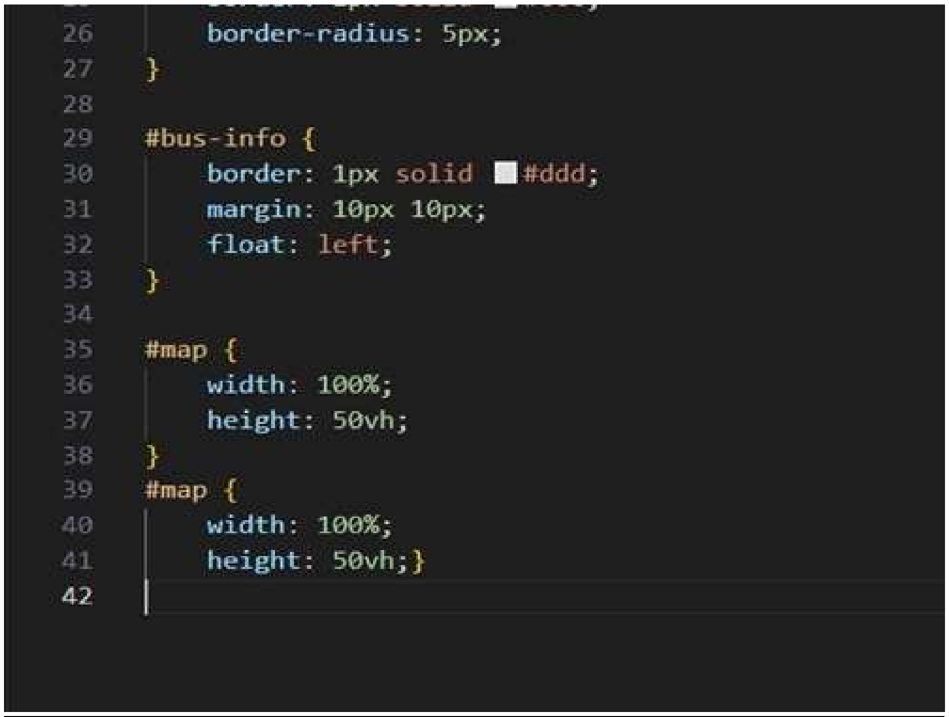
19.<footer>: The footer section displays a copyright notice for the year 2023, indicating ownership of the content.

20.<script src="node.js"></script>: This script element links to an external JavaScript file named "node.js." This file is used for real-time updates and functionality related to the platform.

STYLE.CSS

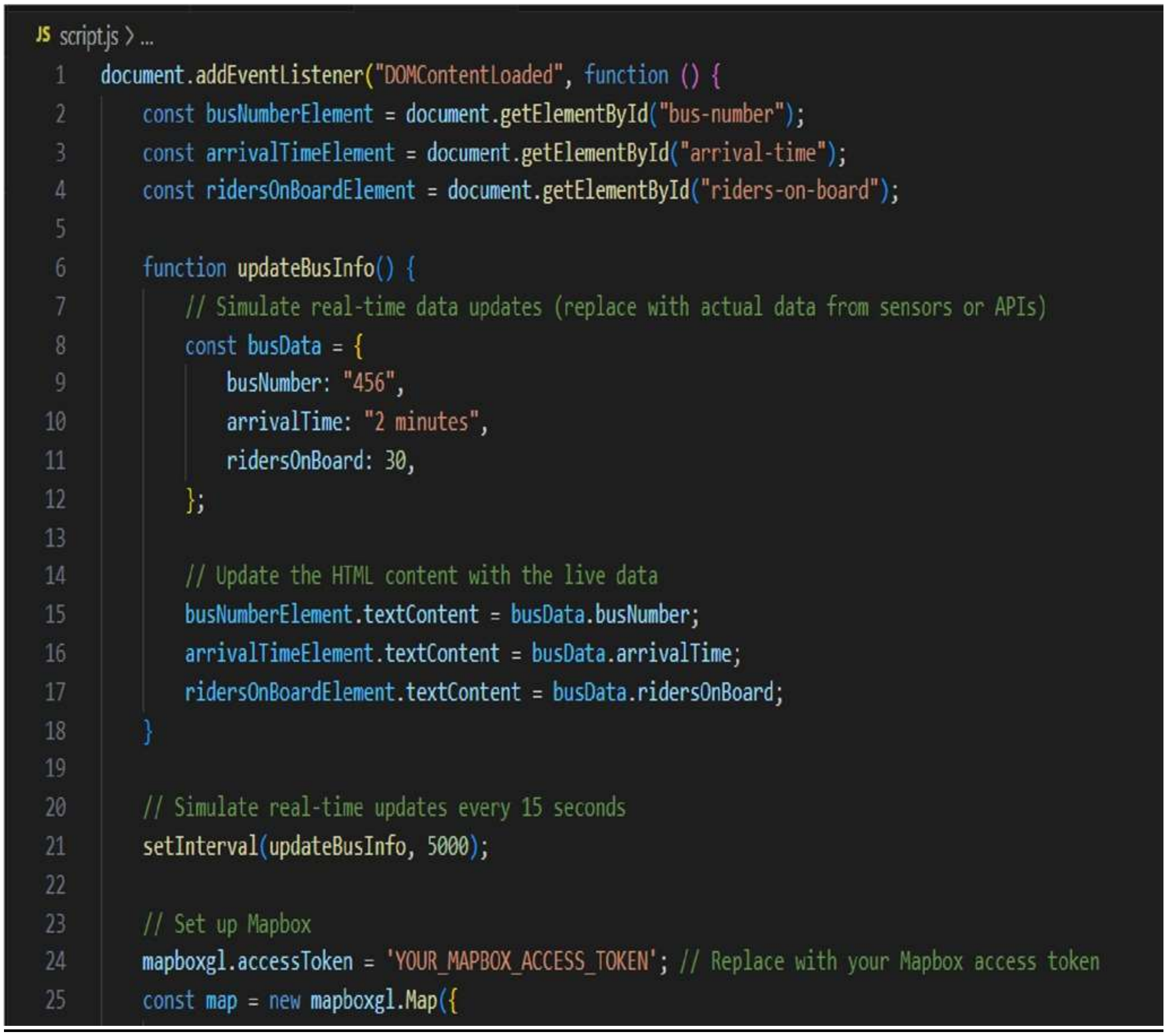


```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0;
}

header {
    background-color: #007bff;
    color: white;
    text-align: center;
    padding: 20px;
}

h1{
    padding: 20px;
    margin-left: 70px;
}

main {
    max-width:100%;
    margin: 20px auto;
    background-color: white;
    padding: 20px;
    border: 1px solid #ccc;
```

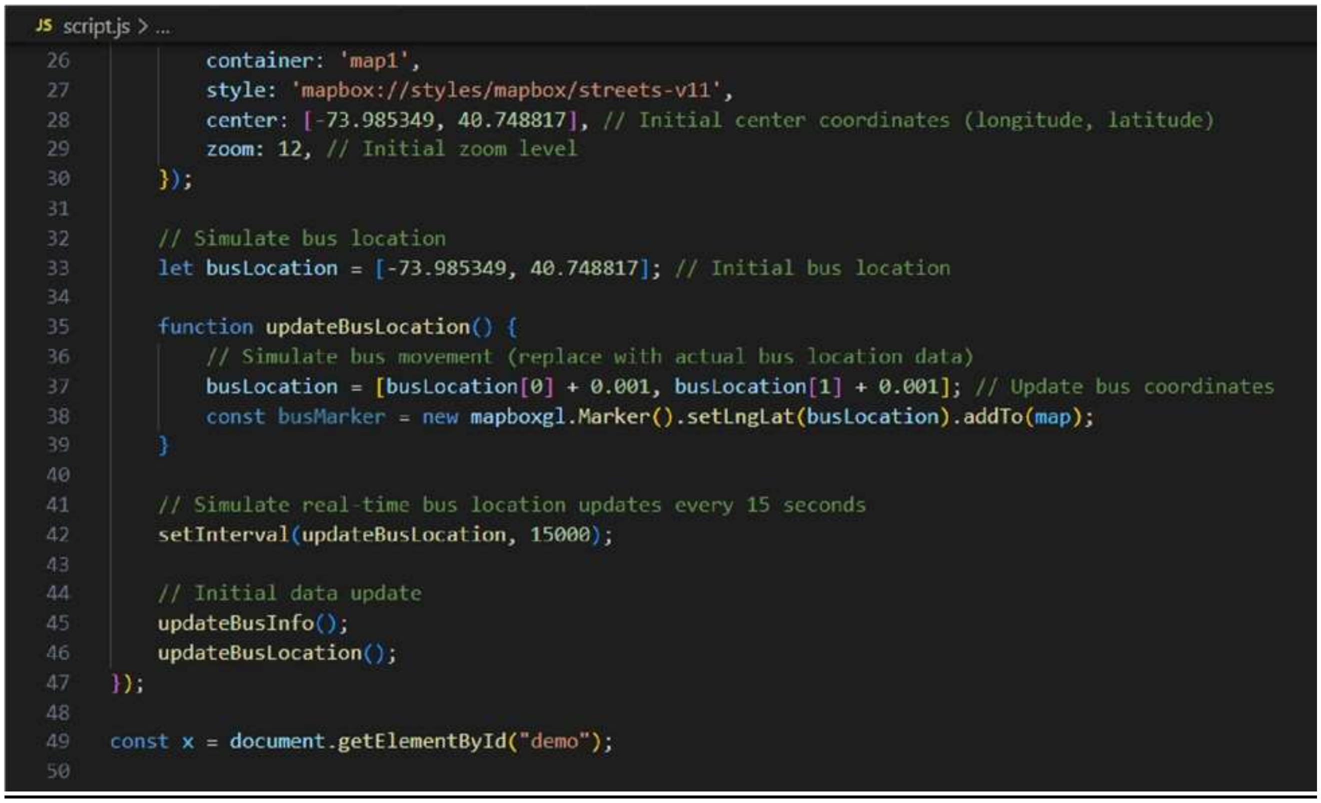


```css
    border-radius: 5px;
}

#bus-info {
    border: 1px solid #ddd;
    margin: 10px 10px;
    float: left;
}

#map {
    width: 100%;
    height: 50vh;
}
#map {
    width: 100%;
    height: 50vh;}

```

SCRIPT.JS

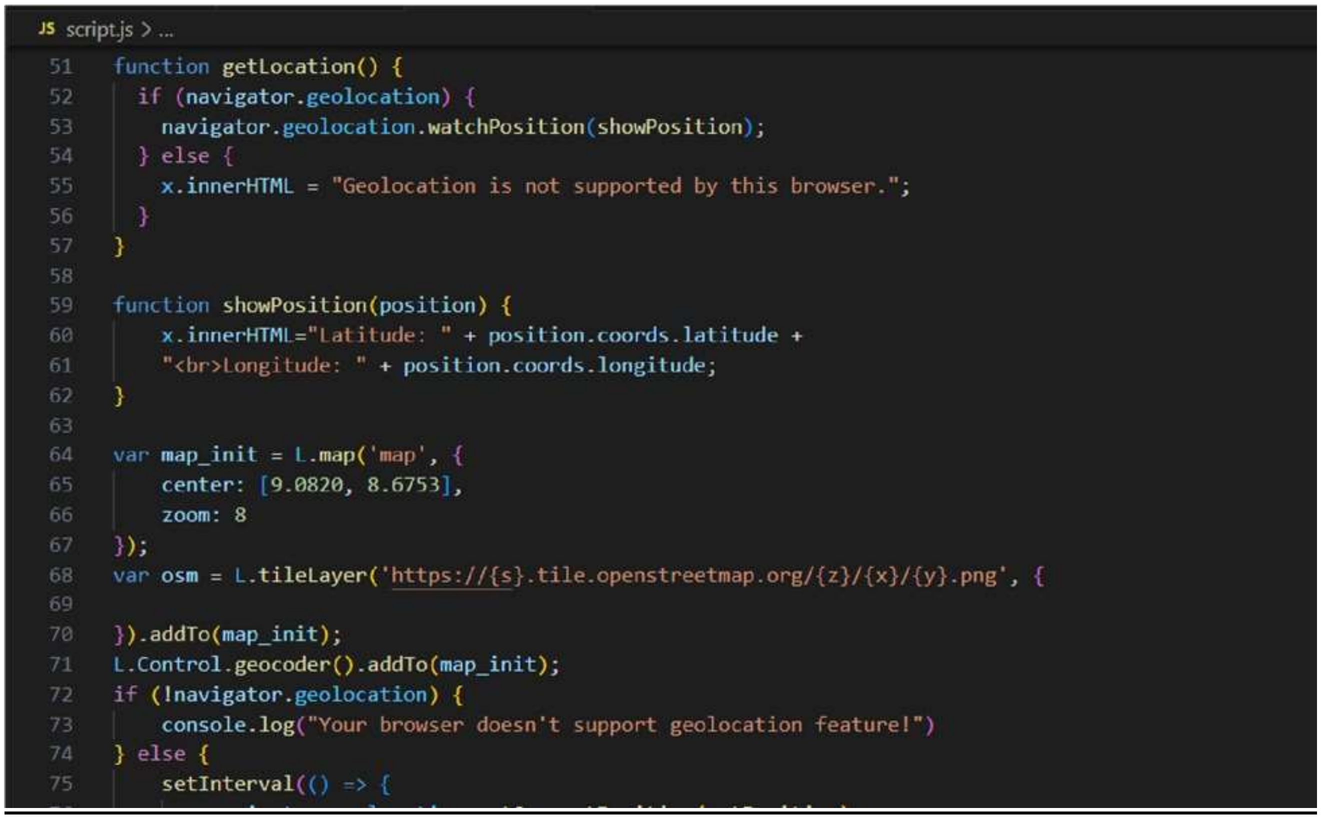

JS script.js > ...

```
document.addEventListener("DOMContentLoaded", function () {
    const busNumberElement = document.getElementById("bus-number");
    const arrivalTimeElement = document.getElementById("arrival-time");
    const ridersOnBoardElement = document.getElementById("riders-on-board");

    function updateBusInfo() {
        // Simulate real-time data updates (replace with actual data from sensors or APIs)
        const busData = {
            busNumber: "456",
            arrivalTime: "2 minutes",
            ridersOnBoard: 30,
        };

        // Update the HTML content with the live data
        busNumberElement.textContent = busData.busNumber;
        arrivalTimeElement.textContent = busData.arrivalTime;
        ridersOnBoardElement.textContent = busData.ridersOnBoard;
    }

    // Simulate real-time updates every 15 seconds
    setInterval(updateBusInfo, 5000);

    // Set up Mapbox
    mapboxgl.accessToken = 'YOUR_MAPBOX_ACCESS_TOKEN'; // Replace with your Mapbox access token
    const map = new mapboxgl.Map({
```



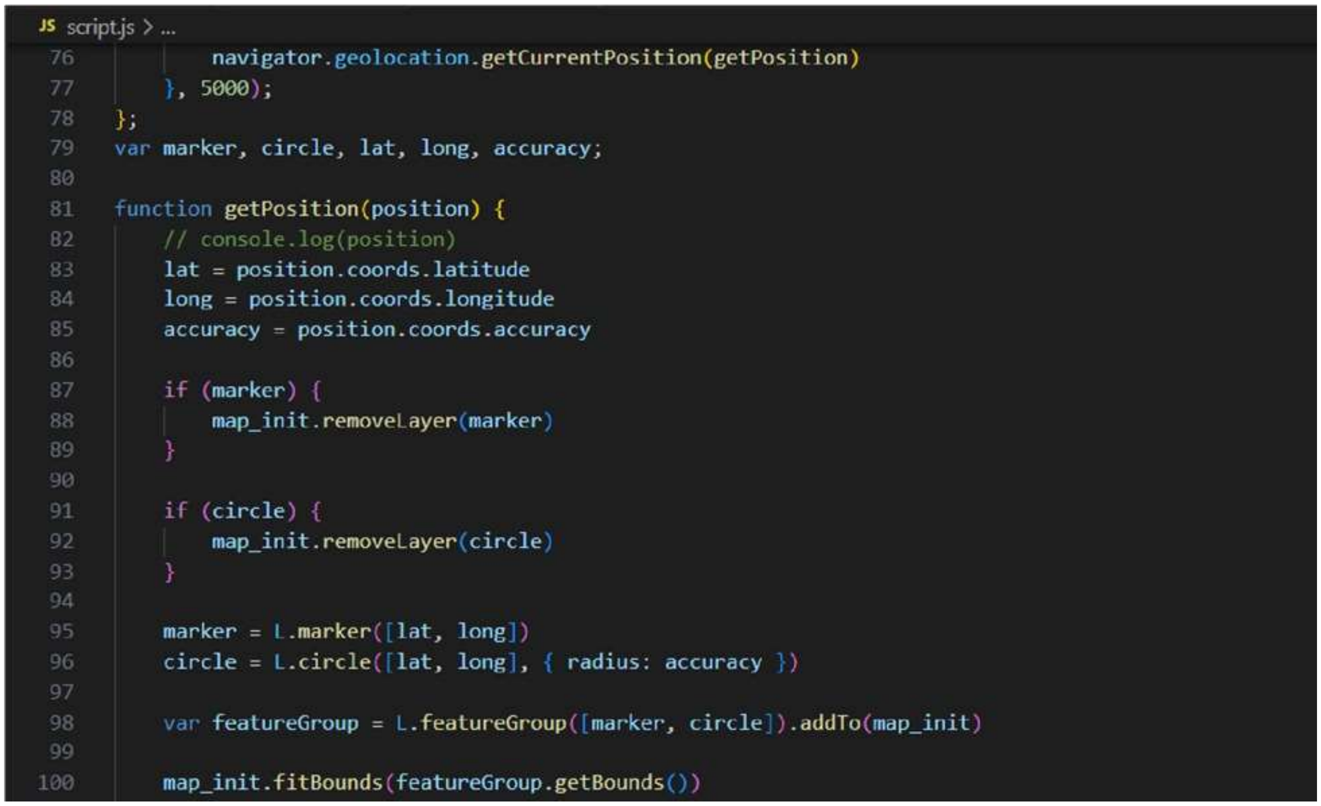
script.js > ...

```javascript
        container: 'map1',
        style: 'mapbox://styles/mapbox/streets-v11',
        center: [-73.985349, 40.748817], // Initial center coordinates (longitude, latitude)
        zoom: 12, // Initial zoom level
    });

    // Simulate bus location
    let busLocation = [-73.985349, 40.748817]; // Initial bus location

    function updateBusLocation() {
        // Simulate bus movement (replace with actual bus location data)
        busLocation = [busLocation[0] + 0.001, busLocation[1] + 0.001]; // Update bus coordinates
        const busMarker = new mapboxgl.Marker().setLngLat(busLocation).addTo(map);
    }

    // Simulate real-time bus location updates every 15 seconds
    setInterval(updateBusLocation, 15000);

    // Initial data update
    updateBusInfo();
    updateBusLocation();
});

const x = document.getElementById("demo");

```



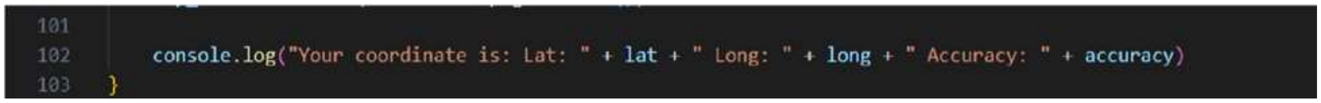
script.js > ...

```javascript
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.watchPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
    x.innerHTML="Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}

var map_init = L.map('map', {
    center: [9.0820, 8.6753],
    zoom: 8
});
var osm = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

}).addTo(map_init);
L.Control.geocoder().addTo(map_init);
if (!navigator.geolocation) {
    console.log("Your browser doesn't support geolocation feature!")
} else {
    setInterval(() => {
```



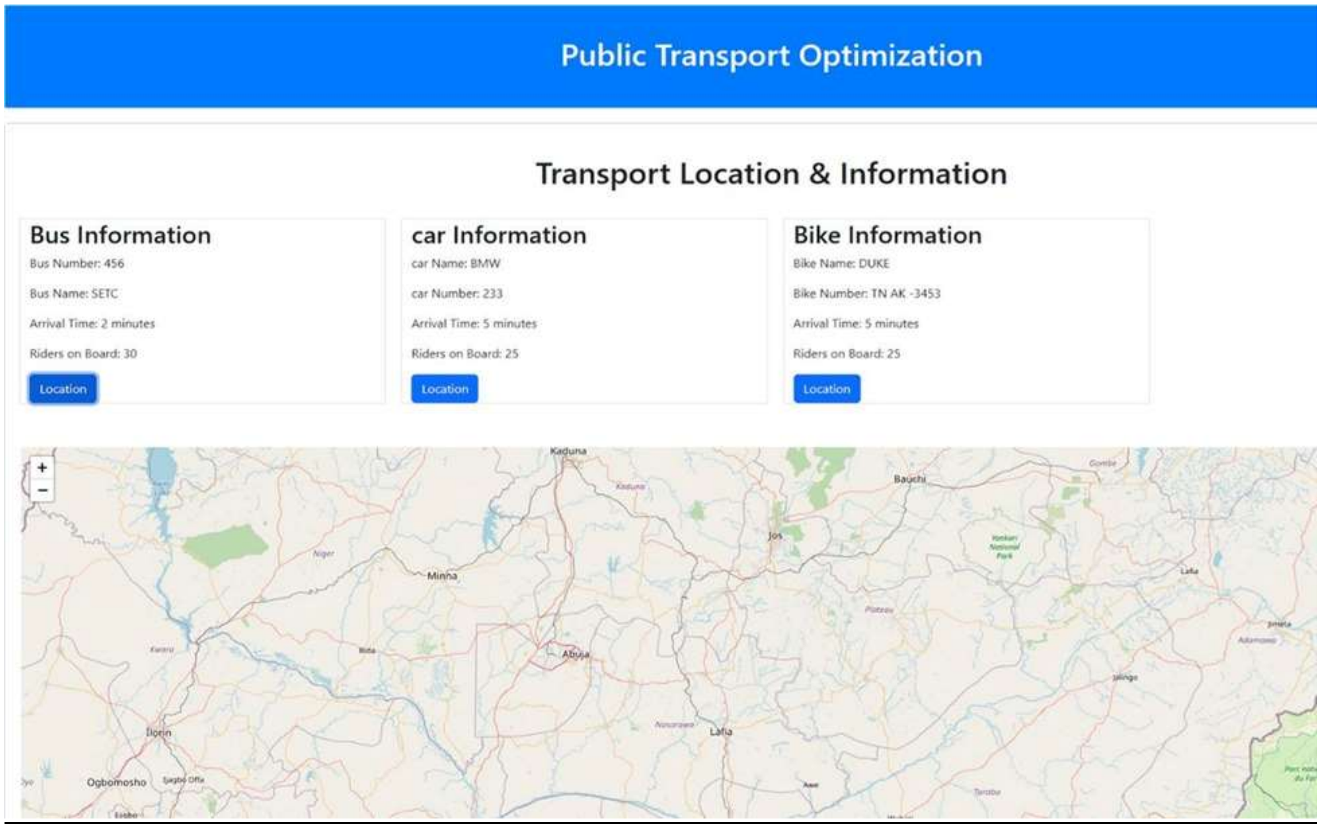
JS script.js > ...

```
            navigator.geolocation.getCurrentPosition(getPosition)
        }, 5000);
    };
    var marker, circle, lat, long, accuracy;

    function getPosition(position) {
        // console.log(position)
        lat = position.coords.latitude
        long = position.coords.longitude
        accuracy = position.coords.accuracy

        if (marker) {
            map_init.removeLayer(marker)
        }

        if (circle) {
            map_init.removeLayer(circle)
        }

        marker = L.marker([lat, long])
        circle = L.circle([lat, long], { radius: accuracy })

        var featureGroup = L.featureGroup([marker, circle]).addTo(map_init)

        map_init.fitBounds(featureGroup.getBounds())
```



```

        console.log("Your coordinate is: Lat: " + lat + " Long: " + long + " Accuracy: " + accuracy)
    }
```

OUTPUT:



Public Transport Optimization

Transport Location & Information

Bus Information

Bus Number: 456

Bus Name: SETC

Arrival Time: 2 minutes

Riders on Board: 30

Location

car Information

car Name: BMW

car Number: 233

Arrival Time: 5 minutes

Riders on Board: 25

Location

Bike Information

Bike Name: DUKE

Bike Number: TN AK -3453

Arrival Time: 5 minutes

Riders on Board: 25

Location

8.ADVANTAGES:

- Reduced Traffic Congestion: Efficient public transport can reduce the number of private vehicles on the road, easing traffic congestion and promoting smoother traffic flow.

- Environmental Benefits: Public transport produces fewer emissions per passenger compared to individual vehicles, contributing to lower air pollution and greenhouse gas emissions.

- Cost-Effectiveness: Public transport optimization can lead to more cost-effective use of resources, ensuring that public funds are utilized efficiently.

9. DISADVANTAGES:

- Initial Cost: Implementing an optimized public transport system often requires significant initial investments in infrastructure, technology, and vehicles.

- Maintenance Challenges: Public transport systems require regular maintenance to ensure safety and efficiency. Maintenance costs can be substantial, especially for large and complex systems.

10. FUTURE SCOPE:

Certainly! When considering the features scope for a public transport optimization project, there are several key aspects to take into account. Here's a breakdown of potential features to consider:

Intelligent route planning algorithms to optimize bus/train/tram routes for efficiency.Real-time route adjustments based on traffic, weather, or other external factors.