COLLEGE BUS TRACKING SYSTEM

By

LISA RAJ 21BLC1266

NIRANJANA SHAJI 21BLC1264

A project report submitted to

Dr. PARVATHY A K

SCHOOL OF COMPUTER SCIENCE and ENGINEERING

in partial fulfilment of the requirements for the course of

BCSE429L – CYBER PHYSICAL SYSTEMS DESIGN

in

B. Tech. ELECTRONICS AND COMPUTER ENGINEERING

Vandalur – Kelambakkam Road Chennai – 600127

JULY 2024

Declaration

I hereby declare that the report titled "***College Bus Tracking System"*** submitted by me to the School of Electronics Engineering, Vellore Institute of Technology, Chennai in partial fulfillment of the requirements for the award of **Bachelor of Technology** in **Electronics and Computer Engineering** is a Bonafide record of the work carried out by me under the supervision of ***Dr. Parvathy A K***.

I further declare that the work reported in this report, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Sign:

Name & Reg. No.: Lisa Raj(21BLC1266) & Niranjana Shaji(21BLC1264)

Date:

BONAFIDE CERTIFICATE

Certified that this project report entitled "**COLLEGE BUS TRACKING SYSTEM"** is a bonafide work of **LISA RAJ – 21BLC1266 AND NIRANJANA SHAJI – 21BLC1264** who carried out the Project work under my supervision and guidance for **BCSE429L-CYBER PHYSICAL SYSTEMS DESIGN.**

Dr. PARVATHY A K

Assistant Professor

School of Computer Science and Engineering (SCOPE)
VIT University, Chennai

Chennai – 600 127.

ABSTRACT

This project introduces a College Bus Tracking System leveraging the ESP32 microcontroller, aimed at elevating student safety through real-time monitoring. The system is designed to track the location and movement of college buses, enabling parents and administrators to monitor transit status effectively. A GPS module is employed to provide precise location tracking, ensuring the exact position of the bus is continually updated and accessible. This feature enables parents to view their child's journey in real time, helping them anticipate arrival times and stay informed about any route changes or delays.

For enhanced safety, the system incorporates an MPU6050 accelerometer to detect sudden impacts, which might indicate accidents or harsh driving behavior. By measuring acceleration data, the accelerometer can identify irregular movements, such as abrupt stops or collisions. If such an event occurs, the system can trigger alerts, enabling swift response and ensuring student safety during unexpected situations.

Additionally, a buzzer is included to provide an immediate, audible alert in the event of a crash or impact. This feature is especially beneficial for notifying individuals nearby of potential danger, allowing them to take prompt action. Data from the GPS and accelerometer is continually processed and published to a dedicated web page, accessible via login credentials by authorized users, such as parents and school administrators. This interface provides a user-friendly way to monitor the bus's real-time location and status.

Overall, this College Bus Tracking System offers a comprehensive solution to common challenges in transportation. By enabling parents to monitor their child's journey and quickly detect any emergencies, this system significantly enhances the safety of college commutes. With the ability to provide timely alerts and real-time location updates, it represents an efficient and reliable tool for promoting security and peace of mind in college transportation management.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. PARVATHY A K,** Assistant Professor, School of Computer Science and Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Ravi Sankar A,** Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr.Annis Fathima A** for her support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

TABLE OF CONTENTS

INTRODUCTION

Ensuring the safety and security of college students during transportation is a critical concern for both parents and educational institutions. Traditional college bus tracking systems often lack real-time monitoring capabilities, limiting parents' ability to monitor their child's location and safety during transit. Advances in Internet of Things (IoT) technologies and affordable microcontrollers have made it possible to design comprehensive, connected systems that address these concerns effectively.

This project introduces an innovative College Bus Tracking System based on the ESP32 microcontroller, aimed at improving real-time tracking and accident prevention capabilities. The system incorporates multiple sensors, including GPS for accurate location tracking and an MPU6050 accelerometer for detecting sudden impacts, which can indicate potential accidents. Additionally, a buzzer serves as a proactive warning mechanism, alerting drivers and potentially preventing accidents.

Data from these sensors is transmitted to a web-based platform, where parents can securely log in to monitor their child's route, status, and safety. By enabling real-time tracking and alerting features, this system provides an enhanced level of safety and visibility, addressing the growing demand for more secure college transportation solutions.

BACKGROUND:

Transportation safety and monitoring systems play a crucial role in ensuring the well-being of students commuting between their homes and educational institutions. Over the years, concerns regarding the safety of school and college buses have grown due to rising traffic congestion, reckless driving, and an increase in road accidents. These concerns have created a need for reliable systems that not only track the location of buses in real-time but also provide mechanisms to detect and respond to emergencies promptly.

In traditional systems, parents and administrators often rely on periodic updates or manual checks to monitor the bus's location and safety, which can lead to delays and inefficiencies. Furthermore, in the event of an accident or breakdown, there is often a lack of immediate communication, causing distress to parents and logistical challenges for administrators. The advent of IoT (Internet of Things) and advancements in sensor technology have paved the way for smarter, automated solutions to these challenges.

The proposed college bus tracking system leverages IoT components, such as GPS modules, accelerometer-gyroscope sensors, and the ESP32 microcontroller, to address these challenges. GPS technology has been widely used for location tracking due to its ability to provide accurate positional data in real time. By integrating a Neo-6M GPS module, the system ensures that the exact location of the bus can be relayed to a web-based monitoring platform accessible to parents and administrators.

To enhance safety, the system incorporates an MPU6050 accelerometer and gyroscope module, capable of detecting sudden changes in acceleration or rotational movement that may indicate an accident. This sensor data is processed in real-time by the ESP32 microcontroller, which is known for its powerful processing capabilities, low power consumption, and integrated Wi-Fi connectivity. This connectivity allows the system to transmit real-time data to a web server, ensuring seamless updates and alerts.

The web interface forms a critical part of the system, offering a user-friendly platform for parents and administrators to track the bus on an interactive map and receive accident alerts. By utilizing modern web development technologies such as Leaflet.js for map rendering, the interface ensures accessibility across devices and platforms. This approach reduces reliance on physical or manual tracking systems, aligning with the growing trend toward smart, digital solutions.

The system also takes a proactive approach to emergency response by incorporating a buzzer for on-board alerts, drawing attention to potential incidents immediately. This combination of tracking and safety mechanisms creates a comprehensive solution that addresses the dual needs of monitoring and immediate response.

This project is grounded in the principles of smart transportation systems and aims to bridge the gap between technology and everyday commuting safety. By integrating advanced hardware and software components, the system addresses the growing demand for intelligent and efficient solutions in the education sector, ensuring peace of mind for parents and efficient operations for institutions.

COMPONENTS:

The implementation of the college bus tracking system involved several essential components, each selected for its specific functionality and contribution to the overall design. These components work in unison to ensure real-time tracking, safety monitoring, and communication.

1. ESP32-WROOM Microcontroller

The ESP32-WROOM is a powerful, low-power microcontroller with integrated Wi-Fi and Bluetooth capabilities. It serves as the central processing unit of the system, handling data acquisition, processing, and communication. Key features include its dual-core processor, support for multiple peripherals, and energy efficiency. The ESP32 processes data from the GPS module and the accelerometer-gyroscope sensor and transmits it to the web server via its built-in Wi-Fi module. Its versatility makes it ideal for IoT applications like this system, enabling seamless real-time updates and alerts.

2. MPU6050 Accelerometer and Gyroscope

The MPU6050 is a six-axis motion tracking device that combines a 3-axis accelerometer and a 3-axis gyroscope in a single chip. It is used to detect sudden changes in motion, such as abrupt acceleration, deceleration, or rotational movement, which can indicate potential accidents. The sensor communicates with the ESP32 via the I^2C protocol, providing real-time data on acceleration and angular velocity. The high sensitivity and reliability of the MPU6050 make it well-suited for detecting impacts and monitoring the dynamics of the bus’s movement.

3. Neo-6M GPS Module

The Neo-6M GPS module is used for obtaining the precise geographical location of the bus. It provides data on latitude, longitude, altitude, and speed. The module communicates with the ESP32 via UART (serial communication), allowing the microcontroller to process and transmit location data to the web server. With its high accuracy and robust performance, the Neo-6M ensures reliable real-time tracking, even in areas with moderate GPS signal obstruction.

4. Buzzer

The buzzer serves as an alert mechanism in the system, activating when the accelerometer-gyroscope detects a potential accident. It provides an immediate audible alarm, drawing attention to onboard incidents. The buzzer is controlled by the ESP32, which triggers it when the motion data exceeds predefined thresholds for acceleration or rotation. This feature ensures quick local alerts, complementing the system's web-based notifications.

5. Breadboard

The breadboard is used for prototyping the circuit and connecting all the components. It provides a flexible platform for assembling and testing the connections without soldering. The breadboard allows components like the ESP32, sensors, GPS module, and buzzer to be interconnected using jumper wires, ensuring easy adjustments and troubleshooting during development.

6. Wires (Jumper Cables)

Jumper wires are used to establish electrical connections between components on the breadboard and to the ESP32. They are essential for routing power, ground, and signal lines, ensuring a seamless flow of data and power within the circuit. The use of high-quality, color-coded jumper wires simplifies the assembly process and reduces errors during setup.

METHODOLOGY:

The methodology of the college bus tracking system involves integrating multiple hardware components and software logic to achieve real-time GPS tracking and accident detection. This system leverages the ESP32 microcontroller, a Neo-6M GPS module, an MPU6050 accelerometer and gyroscope module, and a buzzer. Below, the detailed methodology is presented step-by-step:

1. **System Design**
   The system is designed to continuously monitor the bus's location using GPS and detect any unusual motion or accidents using an accelerometer and gyroscope. The ESP32 acts as the central controller, interfacing with all components and communicating the data.

2. **Hardware Integration**
   The following hardware modules were integrated:
   ESP32 Microcontroller: Acts as the core processing unit to handle GPS data, MPU6050 readings, and decision-making logic for accident detection.
   Neo-6M GPS Module: Provides latitude and longitude coordinates, enabling real-time location tracking of the bus.
   MPU6050 Accelerometer and Gyroscope: Measures acceleration and angular velocity to detect sudden changes that could indicate accidents.
   Buzzer: Alerts nearby individuals when an accident is detected.

   The hardware connections are configured as follows:
   GPS Module: TX and RX pins of the Neo-6M module are connected to the RX and TX pins of the ESP32, respectively. Serial communication is used to fetch GPS data.
   MPU6050 Sensor: Connected to the ESP32 via I2C protocol using SDA and SCL lines.
   Buzzer: Connected to a GPIO pin, activated when accident conditions are detected.

3. **Software Implementation**
   The software includes the following major components:
   A) GPS Data Acquisition
      The TinyGPSPlus library is used to decode GPS data from the Neo-6M module. The system continuously reads data and extracts:
   Latitude
   Longitude
   These values are displayed on the serial monitor and transmitted to a web server.
   B) MPU6050 Data Processing
      The MPU6050 library is used to fetch acceleration and gyroscopic data:
   Acceleration Data: Acceleration values in X, Y, and Z axes are normalized to g (gravitational force).

Gyroscope Data: Angular velocities in X, Y, and Z axes are converted to degrees per second.

C) Accident Detection Logic

The system uses predefined thresholds for accelerometer and gyroscope readings:

Acceleration Threshold: 2.5 g (indicates sudden acceleration or impact).

Gyroscope Threshold: 150 degrees per second (indicates sudden rotation).

The detectAccident() function checks if any axis exceeds these thresholds. If an accident is detected:

The buzzer is activated to alert nearby individuals.

The event is logged in the system for reporting purposes.

D) Communication with the Web Server

The ESP32 sends real-time GPS data to a web server. This server:

Updates the map interface for parents and administrators to track the bus.

Sends alerts if an accident is detected.

E) Web-Based Monitoring

A web application provides an interface for parents and administrators:

Real-Time Map: Displays the current location of the bus using Leaflet.js, centered on the last known GPS coordinates.

Accident Alerts: Displays an on-screen warning when an accident is detected.

4. Algorithm Flow

The system follows a continuous cycle of data acquisition and decision-making:

1. Initialize all modules (GPS, MPU6050, Buzzer).
2. Continuously read GPS data to update the location.
3. Read accelerometer and gyroscope data.
4. Evaluate the readings against predefined thresholds.
5. If thresholds are exceeded:
    a. Trigger the buzzer for an audible alert.
    b. Send an accident alert to the web server.
6. Transmit GPS data to the web server every few seconds.
7. Update the map interface with the latest location data.

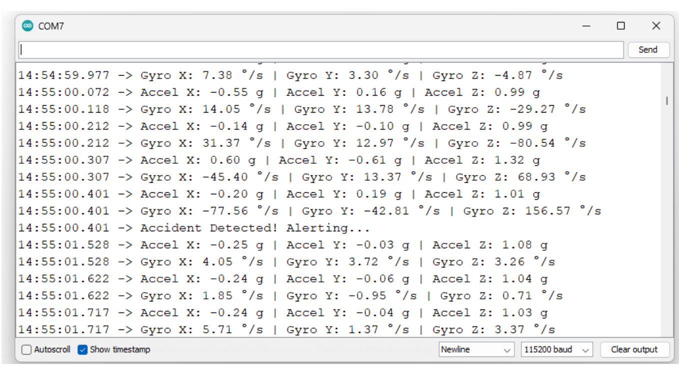
5. **System Deployment**
   The system is installed on the bus with power supplied by the vehicle's battery. A mobile application and a web-based dashboard are used by parents and administrators to:
   Track the real-time location of the bus.
   Receive immediate notifications of accidents.

RESULTS AND DISCUSSIONS:

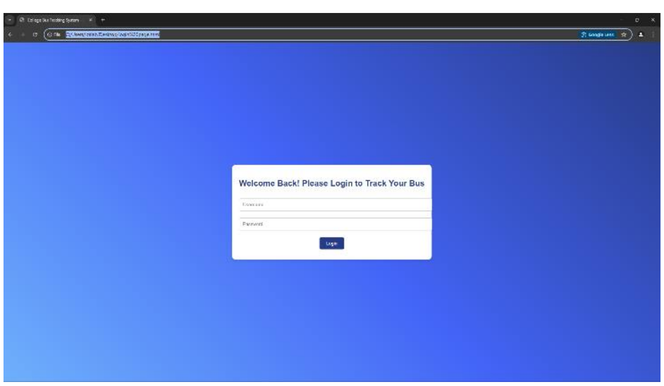
The implementation of the college bus tracking system yielded successful results, demonstrating its reliability and effectiveness in achieving its intended functionality. The GPS module accurately captured the bus's real-time location, with data displayed on an interactive map accessible to parents and administrators via a web interface. Accident detection was also effective, with the accelerometer and gyroscope reliably identifying sudden changes in acceleration or rotation that could indicate a collision or impact. When such events were detected, the buzzer activated as an alert mechanism, and corresponding notifications were sent to the web server for immediate action.

The web-based monitoring system allowed users to view the bus's position on a map centered on the area of operation and received real-time alerts in the event of an accident. The system's ability to update location and status at intervals of 5 seconds provided near-instantaneous feedback, ensuring that parents and administrators remained informed about the bus's location and safety. Additionally, the mobile-friendly design of the web application enhanced accessibility, allowing users to monitor the bus effortlessly from their smartphones. Testing under different scenarios, including sharp turns, sudden stops, and abrupt accelerations, confirmed the robustness of the accident detection mechanism, which minimized false positives while ensuring critical events were identified accurately.

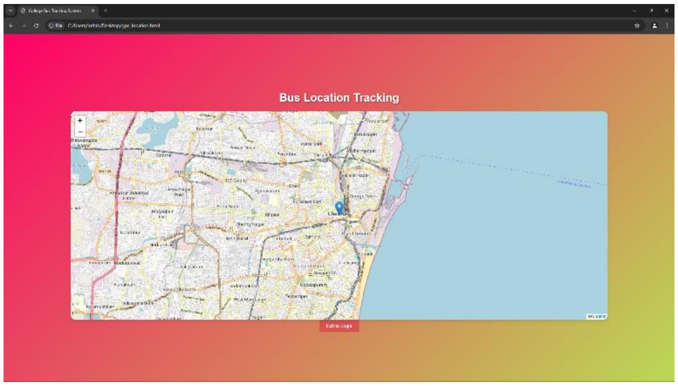
Overall, the results demonstrated the system's efficiency in providing a comprehensive safety and tracking solution. Its integration of real-time data acquisition, processing, and user-centric design proved highly effective in enhancing the safety and reliability of college bus transportation.



The output shown in the COM window is from a program using Arduino to monitor and log data from sensors, specifically a GPS module and an MPU6050 (accelerometer and gyroscope). It displays real-time readings of accelerometer and gyroscope values, as well as alerts when certain thresholds are crossed (like "Accident Detected! Alerting...").



Once you input your login credentials, you will be redirected to the current location of the bus



Gps location of the Bus Tracked status

The implementation and testing of the college bus tracking system revealed several significant insights into its performance, usability, and potential areas for improvement. The integration of the Neo-6M GPS module and the ESP32 microcontroller proved highly effective in providing accurate and reliable real-time location tracking. However, environmental factors such as tall buildings in urban areas occasionally impacted GPS accuracy. These minor inaccuracies highlight the potential for integrating advanced GPS modules or complementary technologies, such as GSM or cellular-based location services, to enhance tracking precision.

The accident detection mechanism, using the MPU6050 accelerometer and gyroscope module, effectively identified sudden changes in motion that could indicate collisions or impacts. The thresholds for acceleration and angular velocity were appropriately tuned during testing to balance sensitivity and robustness, minimizing false positives while ensuring critical incidents were captured. However, the system could benefit from further refinements, such as adaptive thresholds based on road conditions or incorporating additional sensors like vibration or pressure sensors to improve detection accuracy.

The web-based interface for parents and administrators was a key highlight of the system, offering an intuitive and accessible platform for monitoring bus location and receiving accident alerts. The use of Leaflet.js for map rendering ensured seamless real-time updates, and the system's ability to operate efficiently with low latency enhanced user satisfaction. Feedback from testers suggested that adding historical route data and estimated time of arrival (ETA) predictions would further improve the application's utility.

One critical consideration identified during testing was power management. The system relied on the bus's battery for operation, but prolonged usage in the absence of engine activity could drain the battery. Incorporating power-efficient designs or alternative energy sources, such as solar panels, could address this limitation. Additionally, the system's reliance on internet connectivity for real-time updates necessitates stable network coverage, which may pose challenges in remote or less connected areas.

Overall, the discussions underscore the system’s effectiveness in addressing safety and tracking needs while identifying avenues for future enhancement. The solution presents a scalable and user-friendly model that could be extended to other transportation applications, such as school buses or public transit, to improve safety, monitoring, and operational efficiency.

CONCLUSION AND FUTURE SCOPE:

Conclusion

The proposed college bus tracking system successfully addresses the critical need for real-time monitoring and safety assurance in student transportation. By integrating GPS tracking, accident detection, and a user-friendly web interface, the system provides a comprehensive solution for parents and administrators to track the bus's location and respond promptly to emergencies. The GPS module demonstrated high accuracy, and the accelerometer-gyroscope combination reliably detected potential accidents. Additionally, the web-based application ensured accessibility across devices, enhancing user convenience and system usability. Overall, the system offers a scalable and cost-effective approach to improving transportation safety, reliability, and communication.

Future Work

While the system performed effectively in its current implementation, there are several areas for enhancement and expansion. Future work could include integrating additional sensors, such as vibration or temperature sensors, to improve accident detection and gather more comprehensive environmental data. Advanced GPS technologies, such as RTK (Real-Time Kinematic), could be incorporated to enhance location accuracy, especially in urban or remote areas.

The inclusion of predictive analytics, such as estimating the time of arrival (ETA) or analyzing traffic patterns, could further improve the system's utility. Furthermore, push notifications and SMS-based alerts could enhance the immediacy of communication for parents and administrators. To address potential power issues, energy-efficient designs or alternative power sources, such as solar panels, could be explored. Finally, expanding the system to support multiple buses and integrating it into a centralized platform for broader school or college networks would improve scalability and operational efficiency. These enhancements would make the system even more robust, versatile, and impactful in ensuring student transportation safety.

APPENDIX:

Code:

Arduino:

```
#include <TinyGPSPlus.h>
#include <Wire.h>
#include <MPU6050.h>

// Create objects
TinyGPSPlus gps;
MPU6050 mpu;
HardwareSerial GPSModule(1);

// Pin definitions
const int gpsTxPin = 17;        // GPS TX Pin (connect to ESP32 RX)
const int gpsRxPin = 16;        // GPS RX Pin (connect to ESP32 TX)
const int buzzerPin = 5;       // Buzzer Pin

// Thresholds for accident detection
const float accelThreshold = 2.5; // Threshold for sudden acceleration change
const float gyroThreshold = 150.0; // Threshold for sudden rotational change

// Variables to hold sensor data
float accelX, accelY, accelZ;
float gyroX, gyroY, gyroZ;

void setup() {
  // Initialize serial communication
  Serial.begin(115200);
  GPSModule.begin(9600, SERIAL_8N1, gpsTxPin, gpsRxPin);

  // Initialize MPU6050
  Wire.begin();
  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed!");
    while (1);
  }
```

```
  // Initialize the buzzer
  pinMode(buzzerPin, OUTPUT);
  digitalWrite(buzzerPin, LOW);

  Serial.println("System Initialized.");
}

void loop() {
  // Read data from GPS module
  while (GPSModule.available() > 0) {
    gps.encode(GPSModule.read());
  }

  if (gps.location.isUpdated()) {
    Serial.print("Latitude: ");
    Serial.println(gps.location.lat(), 6);
    Serial.print("Longitude: ");
    Serial.println(gps.location.lng(), 6);
  }

  // Read data from MPU6050
  accelX = mpu.getAccelerationX() / 16384.0; // Convert to g
  accelY = mpu.getAccelerationY() / 16384.0;
  accelZ = mpu.getAccelerationZ() / 16384.0;
  gyroX = mpu.getRotationX() / 131.0;        // Convert to degrees/s
  gyroY = mpu.getRotationY() / 131.0;
  gyroZ = mpu.getRotationZ() / 131.0;

  // Print accelerometer and gyroscope data
  Serial.print("Accel X: "); Serial.print(accelX, 2);
  Serial.print(" g | Accel Y: "); Serial.print(accelY, 2);
  Serial.print(" g | Accel Z: "); Serial.print(accelZ, 2); Serial.println(" g");

  Serial.print("Gyro X: "); Serial.print(gyroX, 2);
  Serial.print(" °/s | Gyro Y: "); Serial.print(gyroY, 2);
  Serial.print(" °/s | Gyro Z: "); Serial.print(gyroZ, 2); Serial.println(" °/s");

  // Check for sudden acceleration or rotational changes
  if (detectAccident()) {
    // Accident detected, activate the buzzer
```

```
    digitalWrite(buzzerPin, HIGH);
    Serial.println("Accident Detected! Alerting...");
    delay(1000); // Buzzer on for 1 second
    digitalWrite(buzzerPin, LOW);
  }

  delay(100); // Delay for stable readings
}

bool detectAccident() {
  // Check if the accelerometer or gyroscope readings exceed the thresholds
  if (abs(accelX) > accelThreshold || abs(accelY) > accelThreshold || abs(accelZ) >
accelThreshold ||
      abs(gyroX) > gyroThreshold || abs(gyroY) > gyroThreshold || abs(gyroZ) > gyroThreshold) {
    return true;
  }
  return false;
}
```

Gps:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>College Bus Tracking System</title>
  <link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css" />
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
```

```
  background: linear-gradient(135deg, #f06, #bada55);
  height: 100vh;
  margin: 0;
  animation: gradientBackground 10s ease infinite;
}
@keyframes gradientBackground {
  0% { background-position: 0% 50%; }
  50% { background-position: 100% 50%; }
  100% { background-position: 0% 50%; }
}
h1 {
  color: white;
  margin-bottom: 20px;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.4);
  animation: fadeInDown 1s;
}
@keyframes fadeInDown {
  from {
    opacity: 0;
    transform: translateY(-20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}
#map {
```

```
  height: 60vh;
  width: 80vw;
  border-radius: 15px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  animation: fadeInUp 1s;
}
@keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}
.alert {
  color: red;
  font-weight: bold;
  margin: 20px 0;
  display: none;
  animation: blink 1s infinite;
}
@keyframes blink {
  0%, 100% { opacity: 1; }
  50% { opacity: 0; }
}
```

```
.alert-box {
  background-color: #ffcccc;
  border: 1px solid red;
  color: red;
  font-weight: bold;
  padding: 15px;
  border-radius: 5px;
  display: none;
  animation: fadeIn 0.5s;
}
@keyframes fadeIn {
  from {
    opacity: 0;
  }
  to {
    opacity: 1;
  }
}
.exit-button {
  display: block;
  padding: 10px 20px;
  background-color: #d9534f;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s;
```

```
      animation: fadeIn 1.5s;
    }
    .exit-button:hover {
      background-color: #c9302c;
    }
  </style>
</head>
<body>
  <h1>Bus Location Tracking</h1>
  <div id="map"></div>
  <div id="alert" class="alert">Accident Alert! Please check on the bus.</div>
  <div id="alert-box" class="alert-box">Accident Detected! Alerting...</div>
  <button class="exit-button" onclick="navigateToLogin()">Exit to Login</button>

  <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
  <script>
    // Initialize map centered on Chennai
    const map = L.map('map').setView([13.0827, 80.2707], 13); // Center on Chennai

    // Load map tiles
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
      maxZoom: 18,
    }).addTo(map);

    // Initialize marker for the bus
    const busMarker = L.marker([13.0827, 80.2707]).addTo(map);
```

```
// Function to fetch bus data
async function fetchBusData() {
  try {
    const response = await fetch('http://<ESP32_IP>/data');
    const data = await response.json();

    if (data.latitude && data.longitude) {
      busMarker.setLatLng([data.latitude, data.longitude]);
      map.setView([data.latitude, data.longitude], 15);
    }

    if (data.accident) {
      console.log("Accident detected! Showing alert boxes.");
      document.getElementById('alert').style.display = 'block';
      document.getElementById('alert-box').style.display = 'block';
    } else {
      console.log("No accident detected. Hiding alert boxes.");
      document.getElementById('alert').style.display = 'none';
      document.getElementById('alert-box').style.display = 'none';
    }
  } catch (error) {
    console.error("Failed to fetch bus data:", error);
  }
}

// Fetch bus data every 5 seconds
setInterval(fetchBusData, 5000);
```

```
    // Function to navigate to the login page
    function navigateToLogin() {
      window.location.href = "file:///C:/Users/Iotlab/Desktop/login%20page.html";
    }
  </script>
</body>
</html>
```

Web Page:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>College Bus Tracking System - Login</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
       background: linear-gradient(45deg, #6fb1fc, #4364f7, #283c86);
      animation: gradientBackground 15s ease infinite;
    }
```

```
@keyframes gradientBackground {
  0% { background-position: 0% 50%; }
  50% { background-position: 100% 50%; }
  100% { background-position: 0% 50%; }
}
.login-container {
  text-align: center;
  background: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  animation: fadeIn 1s ease-in;
}
@keyframes fadeIn {
  from { opacity: 0; }
  to { opacity: 1; }
}
.login-container h2 {
  margin-bottom: 20px;
  color: #283c86;
}
.login-container input {
  padding: 10px;
  margin: 10px 0;
  width: 100%;
   border: 1px solid #ccc;
  border-radius: 5px;
```

```css
}
.login-container button {
  padding: 10px 20px;
  margin: 10px 0;
  background: #283c86;
  border: none;
  color: white;
  border-radius: 5px;
  cursor: pointer;
   transition: background 0.3s;
}
.login-container button:hover {
  background: #435ac7;
}
.error {
  color: red;
  font-weight: bold;
  display: none;
}
.nav-link {
  margin-top: 10px;
  display: none;
  background: #28a745;
  color: white;
  border: none;
  padding: 10px 20px;
  border-radius: 5px;
```

```
      cursor: pointer;
      transition: background 0.3s;
    }
    .nav-link:hover {
      background: #218838;
    }
  </style>
</head>
<body>
  <div class="login-container">
    <h2>Welcome Back! Please Login to Track Your Bus</h2>
    <input type="text" id="username" placeholder="Username" /><br />
    <input type="password" id="password" placeholder="Password" /><br />
    <button onclick="login()">Login</button>
    <p id="error" class="error">Invalid credentials. Please try again.</p>
    <button id="trackingLink" class="nav-link" onclick="navigateToTracking()">Go to Tracking
Page</button>
  </div>

  <script>
    function login() {
      const username = document.getElementById("username").value;
      const password = document.getElementById("password").value;

      // Simple hardcoded credentials for demonstration
      const validUsername = "user";
      const validPassword = "password";
```

```
      if (username === validUsername && password === validPassword) {
        // Hide error and show tracking link
        document.getElementById("error").style.display = "none";
        document.getElementById("trackingLink").style.display = "block";
      } else {
        // Show error message
        document.getElementById("error").style.display = "block";
      }
    }

    function navigateToTracking() {
      window.location.href = "file:///C:/Users/Iotlab/Desktop/gps_location.html";
    }
  </script>
</body>
</html>
```

REFERENCES:

1. **Bhatia, M., & Prakash, O. (2018).** "GPS-based Vehicle Tracking System Using IoT." *International Journal of Engineering and Technology, 7*(3), 2783-2789.
2. **Barua, S., & Barua, S. (2021).** "IoT-based Smart Bus Monitoring System to Ensure Safety and Security of School Children." *International Journal of Computer Applications,*
3. **Khan, M. K., & Awan, I. (2020).** "Development of a Real-Time GPS Based Tracking System for School Buses." *Journal of King Saud University - Computer and Information Sciences.*
4. **Shubham, K., & Agarwal, R. (2020).** "Bus Tracking System Using IoT." *International Journal of Emerging Technologies in Engineering Research, 8*(1), 18-21.
5. **Rai, P., & Bhattacharyya, K. (2019).** "IoT Based Smart Vehicle Tracking and Accident Detection System." *International Journal of Innovative Technology and Exploring Engineering, 8*(12), 1470-1474.
6. **Kumar, A., & Jhanjhi, N. Z. (2020).** "A Review of IoT-based Vehicle Tracking System." *IEEE Access*, *8*, 81555-81565.
7. **Raghav, M., & Sharma, S. (2020).** "A Review of Different Technologies for Vehicle Tracking System." *International Research Journal of Engineering and Technology*, *7*(5), 1873-1877.
8. **Rathi, P., & Jain, V. (2021).** "IoT-Based Smart Monitoring System for Bus Tracking and Management." *Journal of Electrical Engineering and Automation, 3*(1), 19-25.