DRIVER CONGESSION ALERTING SYSTEM

A PROJECT REPORT

Submitted by

MANIMARAN V	**513121106302**
MOHAN J	**513121106303**
PREM KUMAR S	**513121106306**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF

TECHNOLOGY, VELLORE-02.

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"DRIVER CONGESSION ALERTING SYSTEM"** is the bonafide work of **"MANIMARAN V – 513121106302, MOHAN J – 513121106303, PREM KUMAR S – 513121106306"** who carried out the project work under my supervision.

SIGNATURE
Dr. S. LETITIA, M.E., Ph.D.,
HEAD OF THE DEPARTMENT,
PROFESSOR,
Department of Electronics and
Communication Engineering,
Thanthai Periyar Govt. Institute
of Technology, Vellore-632002.

SIGNATURE
Prof.S.SABARI RAJAN,M.E.,
ASSISTANT PROFESSOR,
INTERNAL GUIDE,
Department of Electronics and
Communication Engineering,
Thanthai Periyar Govt. Institute
of Technology, Vellore-632002.

Project work viva voce held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

This single-page application integrates a live GPS tracking system with real-time traffic congestion alerts, providing users with dynamic navigation assistance. The system employs asyncio, websockets, and serial communication to continuously transmit GPS data from a device to a web browser.

Upon establishing a connection, the server retrieves GPS data, extracts latitude and longitude coordinates, and sends them to the client via websockets. The client-side JavaScript dynamically updates a Leaflet map with the received coordinates, enabling users to visualize their current location.

The interface features an interactive map where users can input destinations through a search box. Utilizing the Nominatim API for geocoding, the system calculates and displays routes to the specified destinations using Leaflet Routing Machine. Users can interact with the map to plan and adjust their routes.

Furthermore, the system incorporates a traffic congestion alert mechanism. Predefined traffic zones, represented by colored polylines on the map, indicate varying levels of congestion. When the GPS marker enters a congested area, an alert is triggered, providing users with immediate awareness of the traffic condition. Additionally, a synthesized voice message enhances user experience by delivering auditory alerts.

In summary, this application combines live GPS tracking, interactive mapping, route planning, geocoding, and real-time traffic congestion alerts into a single-page interface, offering users comprehensive navigation assistance with dynamic visualization and congestion awareness.

TABLE OF CONTENT

INTRODUCTION

In today's fast-paced world, efficient navigation is essential for optimizing travel routes and avoiding traffic congestion. Traditional navigation systems often lack real-time updates on traffic conditions, leading to inefficient routes and unexpected delays. To address this challenge, this project introduces a novel single-page application that integrates live GPS tracking with a dynamic traffic congestion alert system.

The application leverages modern web technologies such as websockets, asyncio, and Leaflet mapping library to provide users with an intuitive and interactive navigation experience. By continuously transmitting GPS data from a device to a web browser, users can visualize their current location in real-time on an interactive map interface.

In addition to live GPS tracking, the application incorporates a sophisticated traffic congestion alert mechanism. By analyzing predefined traffic zones and calculating distances between the user's location and these zones, the system promptly alerts users of potential congestion ahead. This proactive approach empowers users to make informed decisions and adjust their routes to avoid traffic delays.

Through the seamless integration of live GPS tracking and real-time traffic congestion alerts, this application offers users a comprehensive solution for efficient navigation. Whether navigating urban streets or highways, users can rely on this system to navigate with confidence, optimize their routes, and arrive at their destinations in a timely manner.

DESIGN STAGES FOR IMPLEMENTATION

1. Requirement analysis:

- Identify the primary goals and objectives of the application, including live GPS tracking and traffic congestion alerts.
- Determine the target audience and their specific needs and preferences.
- Gather requirements through user interviews, surveys, and market research to ensure the application meets user expectations.

2. System Architecture Design:

- Define the overall system architecture, including server-client communication protocols and data flow.
- Select appropriate technologies and frameworks for implementing live GPS tracking and traffic congestion alert features.
- Design the server-side components responsible for receiving GPS data, processing traffic information, and transmitting alerts to clients.

3. User Interface Design:

- Create wireframes and mockups to visualize the user interface components, such as the map interface, search box, and route controls.
- Design an intuitive and user-friendly interface that enables seamless interaction with the application.
- Incorporate feedback from usability testing to refine the user interface design and improve usability.

4. GPS Data Processing:

- Implement the functionality to read GPS data from the device using serial communication.
- Parse the incoming GPS data to extract latitude and longitude coordinates.
- Ensure robust error handling and data validation to handle edge cases and ensure data integrity.

5. Websockets and Real-Time Communication:

- Set up websockets on the server-side to establish real-time communication with clients.
- Implement asynchronous event handling to transmit GPS data updates to connected clients.
- Handle connection errors and client disconnects gracefully to maintain uninterrupted communication.

6. Mapping Integration:

- Integrate the Leaflet mapping library into the application to display interactive maps.
- Implement functionality to display the user's current location as a marker on the map.
- Enable users to input destinations and display route information using Leaflet Routing Machine.

7. Traffic Congestion Alert System:

- Define predefined traffic zones and congestion levels based on historical data or real-time traffic information sources.
- Calculate distances between the user's location and predefined traffic zones to determine proximity.
- Implement logic to trigger congestion alerts when the user enters a congested area, providing timely warnings.

8. User Interaction and Feedback:

- Implement user interaction features such as destination input, route planning, and route adjustments.
- Provide visual and auditory feedback to users, such as route highlights and traffic congestion alerts.
- Ensure responsiveness and smooth interaction across different devices and screen sizes.

9. Testing and Quality Assurance:

- Conduct comprehensive testing to verify the functionality and performance of the application.
- Perform unit tests, integration tests, and end-to-end tests to identify and fix bugs and ensure system reliability.
- Solicit feedback from beta testers and iterate on the design based on user input.

10.Deployment and Maintenance:

- Deploy the application to a production environment, ensuring scalability and reliability.
- Monitor system performance and user feedback post-deployment to identify areas for improvement.
- Provide ongoing maintenance and updates to address any issues and introduce new features based on user needs and technological advancements.

SOURCE CODE

PYTHON SCRIPT TO READ GPS DATA AND SEND DATA TO MAP:

```
import asyncio
import websockets
import serial
import pynmea2
import webbrowser
import time
async def send_location(websocket, path):
    port = "/dev/ttyAMA0"
    ser = None
    try:
        ser = serial.Serial(port, baudrate=9600, timeout=5)
        while True:
            data = ser.readline().decode('latin-1')
            if data.startswith('$GPRMC'):
                msg = pynmea2.parse(data)
                lat = msg.latitude
                lon = msg.longitude
                await websocket.send(f'{lat},{lon}')
                print(f"Sent: latitude: {lat}, longitude: {lon}")
    except serial.SerialException as e:
        print(f"Serial port error: {e}")
    except websockets.exceptions.ConnectionClosedError:
        print("WebSocket connection closed")
    except KeyboardInterrupt:
        print("Script terminated by user")
    finally:
        if ser and ser.is_open:
            ser.close()

start_server = websockets.serve(send_location, "localhost", 8765)
```

```
print("Server started")
asyncio.get_event_loop().run_until_complete(start_server)
print("WebSocket server running")
webbrowser.open("updmap.html",new=2)
time.sleep(10)
asyncio.get_event_loop().run_forever()
```

PROCESSING CODE:

```
<!DOCTYPE html>
<html>
<head>
  <title>Live GPS Map</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/leaflet@1.7.1/dist/leaflet.css"/>
  <script src="https://cdn.jsdelivr.net/npm/leaflet@1.7.1/dist/leaflet.js"></script>
  <link rel="stylesheet" href="https://unpkg.com/leaflet-routing-machine/dist/leaflet-routing-machine.css" />
  <script src="https://unpkg.com/leaflet-routing-machine/dist/leaflet-routing-machine.min.js"></script>
  <style>
    #mapContainer {
      height: 100vh;
    }
  </style>
</head>
<body>
  <div id="mapContainer"></div>
  <div id="searchBoxContainer" style="position: absolute; top: 10px; left: 10px; z-index: 1000;">
    <input type="text" id="searchBox" placeholder="Enter destination..." style="width: 200px;"/>
    <button id="searchButton">Go</button>

    <button id="clearButton">Clear Route</button>
    <button id="reverseButton">Change Direction</button>
```

```
    <div id="suggestions"></div>
</div>
<script>
    document.getElementById('reverseButton').addEventListener('click', function() {
    var waypoints = control.getWaypoints();
    waypoints.reverse();
    control.setWaypoints(waypoints);
});
    var map = L.map('mapContainer').setView([12.880555, 79.121189], 18);
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        maxZoom: 19,
    }).addTo(map);
    var marker = L.marker([12.880555, 79.121189]).addTo(map);
    function geocodeDestination() {
        var destination = document.getElementById('searchBox').value;
        if (destination) {
            fetch('https://nominatim.openstreetmap.org/search?format=json&q=' + destination)
                .then(response => response.json())
                .then(data => {
                    if (data.length > 0) {
                        var lat = parseFloat(data[0].lat);
                        var lon = parseFloat(data[0].lon);
                        control.setWaypoints([
                            marker.getLatLng(),
                            L.latLng(lat, lon)
                        ]);
                    } else {
                        alert('Destination not found');
                    }
                })
                .catch(error => console.error('Error:', error));
        }
    }
```

```
function clearRoute() {
    control.setWaypoints([]);
}
document.getElementById('searchButton').addEventListener('click', geocodeDestination);
document.getElementById('clearButton').addEventListener('click', clearRoute);
document.getElementById('searchBox').addEventListener('keypress', function(event) {
    if (event.key === 'Enter') {
        geocodeDestination();
    }
});
var control;
var ws = new WebSocket("ws://localhost:8765");
ws.onmessage = function(event) {
    var data = event.data.split(",");
    var lat = parseFloat(data[0]);
    var lon = parseFloat(data[1]);
    marker.setLatLng([lat, lon]);
    map.setView([lat, lon]);
    if (!control) {
        control = L.Routing.control({
            waypoints: [
                L.latLng(lat, lon),
                L.latLng(lat, lon)
            ],
            routeWhileDragging: true
        }).addTo(map);
    } else {
        control.setWaypoints([
            L.latLng(lat, lon),
            control.getWaypoints()[1]
        ]);
    }
};
```

```
ws.onerror = function(event) {
    console.error("WebSocket error:", event);
};
ws.onclose = function(event) {
    console.log("WebSocket connection closed");
};
function showSuggestions(suggestions) {
    var suggestionsDiv = document.getElementById('suggestions');
    suggestionsDiv.innerHTML = '';
    suggestions.forEach(function(suggestion) {
        var suggestionItem = document.createElement('div');
        suggestionItem.textContent = suggestion.label;
        suggestionItem.addEventListener('click', function() {
            document.getElementById('searchBox').value = suggestion.label;
            control.setWaypoints([
                marker.getLatLng(),
                L.latLng(suggestion.lat, suggestion.lon)
            ]);
            suggestionsDiv.innerHTML = '';
        });
        suggestionsDiv.appendChild(suggestionItem);
    });
}
document.getElementById('searchBox').addEventListener('input', function(event) {
    var input = event.target.value;
    if (input) {
        fetch('https://nominatim.openstreetmap.org/search?format=json&q=' + input)
            .then(response => response.json())
            .then(data => {
                var suggestions = data.map(function(item) {
                    return {
                        label: item.display_name,
                        lat: parseFloat(item.lat),
```

```
                lon: parseFloat(item.lon)
              };
            });
            showSuggestions(suggestions);
          })
          .catch(error => console.error('Error:', error));
    } else {
      document.getElementById('suggestions').innerHTML = '';
    }
  });
  function simulateTraffic() {
    var trafficData = [
      { coords: [[12.879423, 79.134560], [12.881482, 79.135028]], color: 'blue' },//high
      { coords: [[12.879727, 79.134515], [12.880148, 79.133458]], color: 'orange' },//medium
      { coords: [[12.89860, 79.04720], [12.89900, 79.04830]], color: 'green' } //less
    ];
    var thresholdDistance = 2000;// in meters
    function getDistance(lat1, lon1, lat2, lon2) {
      var point1 = L.latLng(lat1, lon1);
      var point2 = L.latLng(lat2, lon2);
      return point1.distanceTo(point2);
    }
    function speak(message) {
      var synth = window.speechSynthesis;
      var utterance = new SpeechSynthesisUtterance(message);
      synth.speak(utterance);
    }
    function checkTraffic(lat, lon) {
      trafficData.forEach(function(traffic) {
        var polyline = L.polyline(traffic.coords);
        polyline.getLatLngs().forEach(function(latlng) {
          var distance = getDistance(lat, lon, latlng.lat, latlng.lng);
          if (distance <= thresholdDistance) {
```

```
                    var message = 'Entering ' + traffic.color + ' traffic area within ' + thresholdDistance
+ ' meters!';
                    alert(message);
                    speak(message);
                }
            });
        });
    }
    ws.onmessage = function(event) {
        var data = event.data.split(",");
        var lat = parseFloat(data[0]);
        var lon = parseFloat(data[1]);
        marker.setLatLng([lat, lon]);
        map.setView([lat, lon]);
        checkTraffic(lat, lon);
    };
    trafficData.forEach(function(traffic) {
        L.polyline(traffic.coords, { color: traffic.color }).addTo(map);
    });
}
simulateTraffic();
</script>
</body>
</html>
```

WORKING

Live GPS Tracking:

1. Initialization:

- Upon launching the application, the server initializes and starts listening for incoming websocket connections on the specified port.

2. GPS Data Acquisition:

- The application establishes a connection with the GPS device through the specified serial port.
- As the GPS device emits data, the application reads the incoming data stream, filtering out sentences starting with "$GPRMC," which contain essential location information.

3. Data Parsing and Transmission:

- Using the pynmea2 library, the application parses the relevant latitude and longitude coordinates from the received GPS data.
- Once parsed, the latitude and longitude coordinates are transmitted to connected clients via websockets, enabling real-time updates of the user's location.

4. Client-Side Rendering:

- Clients, typically web browsers, establish websocket connections with the server to receive live GPS updates.
- Upon receiving latitude and longitude coordinates from the server, the client-side JavaScript updates the Leaflet map interface, moving the marker to reflect the user's current location.

Traffic Congestion Alert System:

1. Predefined Traffic Zones:

- The application defines predefined traffic zones based on historical data or real-time traffic information.
- Each traffic zone is associated with a congestion level, categorized as high, medium, or low congestion.

2. Proximity Calculation:

- As the user's location updates are received from the server, the application calculates the distance between the user's location and the predefined traffic zones.
- Using the Haversine formula or similar distance calculation methods, the application determines the proximity of the user to each traffic zone.

3. Congestion Alert Triggering:

- When the user enters a predefined traffic zone, the application triggers a congestion alert based on the congestion level associated with that zone.
- The alert may manifest as a visual indication on the map, such as highlighting the congested area, and/or an auditory alert using synthesized speech.

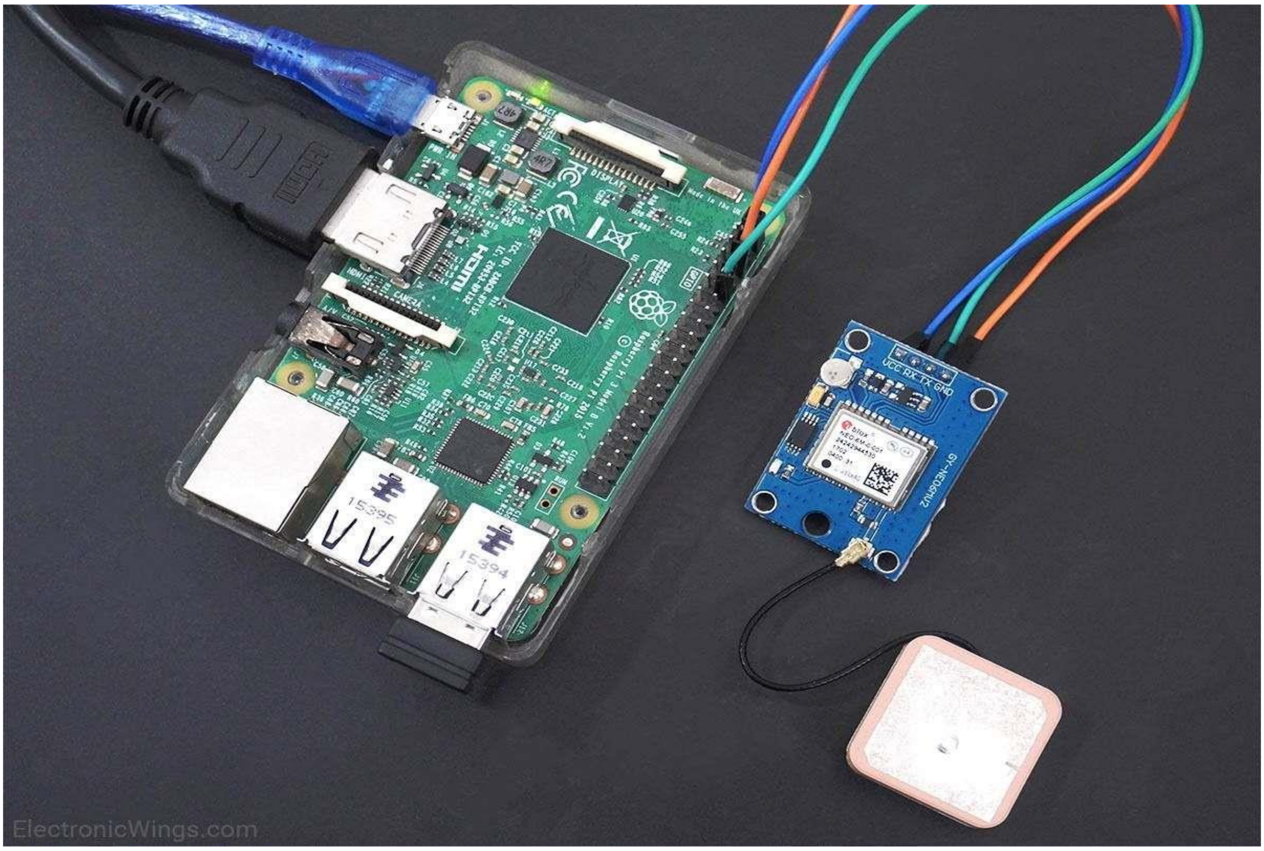
4. Alert Handling and User Response:

- Upon receiving a congestion alert, the user can take appropriate action, such as adjusting their route to avoid the congested area.
- The application may provide alternative route suggestions or dynamically update the current route based on real-time traffic conditions.

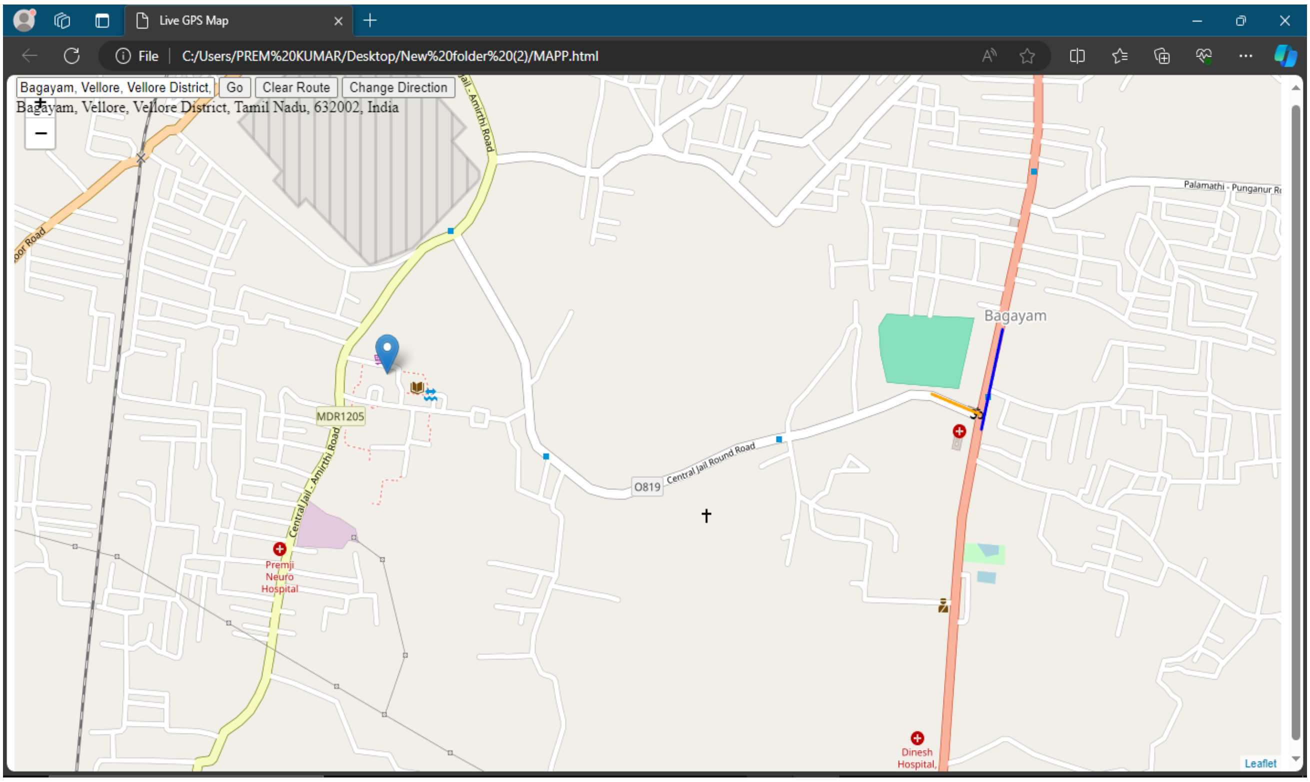
5. Continued Monitoring and Feedback:

- The application continuously monitors the user's location and updates traffic congestion information as the user navigates.
- Users receive ongoing feedback and alerts regarding traffic conditions, empowering them to make informed decisions and optimize their routes in real-time.

PHYSICAL DESIGN:



CREATED MAP:



ADVANTAGES

1. **Real-Time Location Awareness:**
   - Live GPS tracking provides users with up-to-date information about their current location, enabling them to accurately monitor their position on the map in real-time. This feature is particularly beneficial for users navigating unfamiliar areas or traveling through dynamic environments.
2. **Enhanced Navigation Accuracy:**
   - By continuously transmitting GPS data from the device to the application, users can rely on accurate and precise location updates, ensuring they stay on course and reach their destinations efficiently. This level of accuracy minimizes the risk of getting lost or taking incorrect routes.
3. **Dynamic Route Optimization:**
   - With live GPS tracking, users can dynamically adjust their routes based on real-time traffic conditions, road closures, or other unforeseen obstacles. This flexibility allows for on-the-fly route optimization, helping users avoid traffic congestion and minimize travel time.
4. **Improved Safety and Security:**
   - Live GPS tracking enhances safety and security by providing users with the ability to share their real-time location with trusted contacts or emergency services. In case of an emergency or unforeseen event, responders can quickly locate and assist individuals, improving overall safety and peace of mind.
5. **Efficient Fleet Management:**
   - For businesses or organizations managing fleets of vehicles, live GPS tracking offers valuable insights into vehicle locations, routes, and performance metrics. Fleet managers can optimize operations, reduce fuel consumption, and enhance productivity by monitoring and analyzing real-time fleet data.
6. **Seamless Integration with Mapping Services:**
   - Live GPS tracking seamlessly integrates with mapping services and navigation applications, providing users with a familiar and intuitive interface for route planning and navigation. This integration enhances user experience and accessibility, ensuring a smooth and efficient navigation process.

APPLICATIONS

1. Vehicle Tracking and Fleet Management:

- Live GPS tracking is widely used in vehicle tracking systems for fleet management in industries such as transportation, logistics, and delivery services. Fleet managers can monitor the real-time location of vehicles, optimize routes, improve fuel efficiency, and ensure timely deliveries.

2. Personal Navigation and Travel:

- Individuals rely on live GPS tracking for personal navigation during travel, hiking, biking, or exploring unfamiliar locations. With GPS-enabled devices and navigation apps, users can plan routes, find points of interest, and navigate efficiently, enhancing their travel experience and safety.

3. Asset Tracking and Security:

- Live GPS tracking is employed to monitor and secure valuable assets such as high-value shipments, equipment, and machinery. By tracking the real-time location of assets, businesses can prevent theft, mitigate risks, and ensure the security and integrity of their assets during transit or storage.

4. Emergency Response and Public Safety:

- Emergency services and public safety agencies utilize live GPS tracking to locate and dispatch resources quickly in response to emergencies, accidents, or natural disasters. Real-time location data enables responders to deploy assistance to the precise location of incidents, reducing response times and saving lives.

5. Geographic Information Systems (GIS):

- Live GPS tracking plays a crucial role in Geographic Information Systems (GIS) for mapping, surveying, and spatial analysis. GIS professionals use real-time GPS data to create accurate maps, conduct field surveys, and analyze spatial patterns for various applications, including urban planning, environmental management, and natural resource exploration.

DISADVANTAGES

1. Dependency on Satellite Signal:

- Live GPS tracking relies on satellite signals for accurate positioning, which may be obstructed or weakened in urban areas with tall buildings, dense foliage, or adverse weather conditions. This can result in signal loss or reduced accuracy, affecting the reliability of the tracking system.

2. Battery Drainage and Device Compatibility:

- Running live GPS tracking applications on mobile devices or GPS-enabled devices can drain battery life significantly, reducing the device's overall operational time. Additionally, not all devices may be compatible with live GPS tracking software, limiting the accessibility of the tracking system.

3. Data Usage and Network Connectivity:

- Live GPS tracking requires a constant data connection to transmit location updates to the server and receive real-time map data. This reliance on data connectivity may result in increased data usage and incur additional costs, particularly in areas with limited or unreliable network coverage.

4. Accuracy Limitations and Signal Interference:

- While GPS technology provides relatively accurate location information, it may encounter inaccuracies or signal interference in certain environments, such as tunnels, underground parking lots, or densely populated urban areas with tall buildings. These accuracy limitations can affect the precision of the tracking system.

5. Potential for Misuse and Security Risks:

- Live GPS tracking systems are susceptible to misuse, exploitation, or hacking, leading to unauthorized access to sensitive location data. Malicious actors may exploit vulnerabilities in the tracking system to track or monitor individuals without their consent, posing security risks and violating privacy rights.

CONCLUSION

Live GPS tracking technology has revolutionized the way we navigate, manage assets, and respond to emergencies in today's interconnected world. Despite some inherent disadvantages, the benefits and applications of live GPS tracking far outweigh the challenges, making it an indispensable tool for various industries and personal use.

From optimizing fleet management and ensuring timely deliveries to enhancing personal safety during travel and outdoor activities, live GPS tracking offers unparalleled convenience, efficiency, and peace of mind. Businesses leverage live GPS tracking to streamline operations, improve productivity, and enhance customer satisfaction, while individuals rely on it for seamless navigation, fitness tracking, and location-based services.

While privacy concerns and technical limitations remain valid considerations, advancements in GPS technology, data security measures, and regulatory frameworks continue to address these challenges and mitigate risks associated with live GPS tracking. With proper safeguards in place, live GPS tracking systems can deliver accurate location data, real-time insights, and actionable intelligence to empower users and organizations in their daily endeavors.

live GPS tracking technology represents a powerful tool for navigating the complexities of our modern world, offering limitless possibilities for innovation, efficiency, and connectivity. As we embrace the potential of live GPS tracking, it is essential to prioritize privacy, security, and ethical considerations to ensure its responsible and beneficial use for all stakeholders.

REFERANCES

1. https://www.raspberrypi.com/documentation/
2. https://www.raspberrypi.com/documentation/computers/raspberry-pi.html
3. https://www.raspberrypi.org/
4. https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf
5. https://leafletjs.com/reference.html
6. https://leafletjs.com/
7. https://osmfoundation.org/
8. https://www.u-blox.com/en/product/neo-6-series