International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Real Time Device Location Tracking

[1]Prasanna Kumari M
Assistant Professor
Vidya Jyothi Institute of Technology
Hyderabad
mprasanna.btech@gmail.com

[2]Bhavya Sree Nelli
UG Student
Vidya Jyothi Institute of Technology
Hyderabad
bhavyasree.nelli@gmail.com

[3]Sathwik Boilla
UG Student
Vidya Jyothi Institute of Technology
Hyderabad
sathwikboilla102@gmail.com

[4]Asritha Malgari
UG Student
Vidya Jyothi Institute of Technology
Hyderabad
asrithareddymalgari@gmail.com

[5]Praizy Yathakula
UG Student
Vidya Jyothi Institute of Technology
Hyderabad
yathakulapraizy0@gmail.com

Abstract: Real-time device tracking has become very crucial for many modern web based systems applications, the increasing demand for location-aware services including fleet management, security monitoring, navigation, and collaborative systems, due to the quick development of web technologies. The design and implementation of a Real-Time Device Location Tracking System using both Frontend and Backend Technologies is presented in this project, allowing for the real-time monitoring of one or more than one device simultaneously without requiring manual page refreshes via a web browser.The proposed system uses Socket.IO to create real-time, bidirectional communication between the server and connected clients, while Node.js and Express.js are used to create a scalable backend server. EJS is an express js template used to create dynamic web applications and HTML templates also allows React functions like enabling page refreshing each time when clients location updates. The client-side program continuously records the device's latitude and longitude Coordinates using the Browser Geolocation API, OpenStreetMap(OSM) is the map data source, while Leaflet.js is the open-source library used to display that data as an interactive map in a web browser. All connected users or clients receive these location updates, which are instantly sent to the server. Leaflet.js is a javascript library used to create an interactive map interface where device locations are dynamically shown as markers and updated in real time without requiring a page refresh. The system also handles device disconnections by removing inactive markers, ensuring reliable tracking results.

KEYWORDS: Express.js, Geolocation API, Leaflet.js, Node.js, OpenStreetMap, Socket.io.

1. Introduction

Real-time device tracking is now a crucial component of many contemporary applications due to the quick development of web technologies and the growing need for location-based services. In domains like navigation, security, and collaborative systems, the capacity to track the real-time location of devices via a web interface is essential. The goal of this project is to create a Node.js Real-Time Device Location Tracking System that allows for accurate and continuous tracking of several devices via a web browser and the ability to monitor the live location of devices through a web interface. Designing and implementing a web-based system that can track and display device locations in real time with low latency is the goal of this project. The project was started in order to get around the drawbacks of conventional location tracking techniques, which depend on delayed updates or frequent page refreshes. The suggested system guarantees quick location updates and seamless user interaction by employing real-time communication technologies.

The project is developed using Node.js as the backend platform due to its event-driven and non-blocking architecture, which is well suited for real-time applications. Express.js is used to create and manage the web server, while Socket.IO a web socket technology enables real-time, bidirectional communication between the server and connected clients. Instead of using HTTP server for tracking client information we have used web sockets which are Full duplex connections.The Browser Geolocation API is used to fetch live latitude and longitude Coordinates data from the user's device and update for every 5 seconds on interactive map which enables live device tracking efficiently. For visualization, Leaflet.js is integrated to display device locations on an interactive map interface. OpenStreetMap tile is

used to display clear visuals of map, templates, roads, buildings with names. For building this project we have used Frontend technologies HTML, CSS, Index.ejs to develop web applications and for logic functioning javascript is used.

The system supports multiple users simultaneously, assigns unique identifiers to each connected device, and dynamically updates or removes location markers based on device connectivity. This ensures accurate and efficient tracking of all active devices in real time.

The proposed system has wide-ranging applications, including live device tracking, fleet and asset monitoring, emergency response systems, campus or event monitoring, and location-based collaborative applications. Due to its lightweight and scalable design, the system can be further extended with features such as authentication, database storage, geofencing, and enhanced security mechanisms.

2. Literature Survey

Real-time location tracking systems have developed rapidly in recent years due to the growth of GPS technology, mobile applications, IoT devices, and cloud services. Earlier systems mainly depended on basic GPS modules and standalone tracking devices, which required dedicated hardware and manual monitoring. These systems lacked flexibility, scalability, and user-friendly access. With the advancement of web and mobile technologies, modern tracking solutions now provide real-time monitoring, remote access, and improved interaction through simple and intuitive interfaces.

One of the key areas of research in location tracking systems is security and user privacy. Many studies focus on developing secure mobile-based applications that collect real-time location data while using authentication and permission-based access. These systems ensure that only authorized users can view the location data and that information is transmitted securely. Although these methods improve data protection and user trust, maintaining a balance between usability and strict privacy remains a challenge.

Another common approach is the use of GPS modules combined with microcontrollers and mapping services like Google Maps. These systems are widely used in applications such as vehicle tracking, where location data is collected, processed, and displayed in real time. While they offer accurate tracking, they rely heavily on hardware components, which increases cost and limits their usage to specific areas like transportation.

To improve remote monitoring, IoT-based tracking systems have been introduced. In these systems, GPS-enabled devices send real-time data to cloud platforms for processing and visualization. This allows users to monitor objects or individuals from anywhere through mobile or web applications. However, these systems require continuous internet connectivity and proper infrastructure, which makes them more complex and less suitable for simple applications.

Hybrid systems that combine GPS and GSM technologies are also widely used. GPS is used to determine location, while GSM is used to transmit data to servers or mobile devices. Although these systems are efficient, their performance depends on network availability. In areas with weak connectivity, their reliability decreases, which limits their effectiveness.Recent developments also include mobile-based and decentralized tracking systems, especially in applications like contact tracing. These systems focus on protecting user privacy and ensuring secure data handling while supporting large-scale usage. However, challenges such as data accuracy, scalability, and user acceptance still exist.

In smart city environments, intelligent tracking systems are used to monitor public transportation and improve operational efficiency. These systems use real-time data and GPS-based tracking to provide better management and user experience. However, they require advanced infrastructure and are mainly limited to large urban areas.Researchers have also emphasized the importance of data protection techniques such as encryption, anonymization, and access control. These methods are essential in preventing unauthorized access and ensuring safe data handling, especially in cloud-based systems.

Some studies focus on handling large amounts of real-time location data using scalable systems. These approaches highlight the need for efficient data processing frameworks. However, many of them focus more on data analysis rather than real-time visualization and user interaction.Another approach involves combining GPS data with cellular network information to improve location accuracy. While this method provides better results in outdoor environments, it still depends on network infrastructure, which limits its use in lightweight web-based systems.

Even though many advancements have been made, several challenges still remain. Many existing systems are designed for specific applications, which reduces their general usability. Dependence on hardware, internet connectivity, and large infrastructure increases system complexity. Privacy concerns, scalability issues, and cost factors continue to be major limitations.Overall, the literature shows a shift

from hardware-based tracking systems to more flexible, software-driven solutions. While modern systems offer better accuracy and real-time performance, there is still a need for a system that balances security, scalability, cost, and ease of use. This creates an opportunity to develop more efficient and user-friendly tracking solutions.

3. Objectives

The primary objective of this work is to develop a platform-independent and efficient framework for real-time location tracking. The study focuses on designing a unified system that enables continuous tracking of user locations through a browser-based interface, eliminating the need for page refreshing and ensuring seamless real-time updates. Another key objective is to provide an accurate and responsive tracking experience by leveraging modern web technologies for smooth visualization and interaction.

The research also aims to ensure secure data handling by implementing permission-based access, allowing location information to be shared only with authorized users. In addition, it seeks to maintain stable and continuous connectivity between multiple devices, enabling users to access the system from mobiles, laptops, and tablets through a single browser interface. Finally, the study intends to enhance system reliability by automatically managing client connections, including disabling location markers when a user device is disconnected, thereby ensuring accurate and up-to-date tracking information..

3.1 Problem Statement

Providing accurate and efficient real-time location tracking is still a challenge due to limitations in existing systems and changing network conditions. Traditional tracking methods often depend on complex infrastructure, dedicated hardware, or platform-specific applications, which makes them less accessible and difficult for general users to use. These systems may also require manual updates or frequent page refreshing, leading to delays and a poor user experience.

Although modern web-based solutions have improved accessibility, achieving smooth and uninterrupted real-time updates across multiple devices remains difficult. Network instability and device synchronization issues can affect the performance and consistency of tracking systems.Another major concern is data security and user privacy. Many existing systems do not implement strong permission-based access control, which increases the risk of unauthorized access and misuse of sensitive location data. Ensuring that only authorized users can view and share location information is still a critical requirement.In addition, maintaining continuous connectivity and proper synchronization between multiple devices is challenging. These issues can reduce the accuracy and reliability of tracking results, especially in real-time environments.

Overall, these challenges highlight the need for a secure, platform-independent, and efficient system that can provide accurate, continuous, and user-friendly real-time location tracking.

3.2 Overview

The proposed system is a web-based solution designed for real-time location tracking of devices. It allows users to view live location updates directly through a browser without the need for manual page refreshing. By using the Browser Geolocation API, the system continuously collects location data and sends it to the server, ensuring accurate and up-to-date tracking.

The system supports real-time communication between devices, enabling instant updates and smooth interaction. It is capable of handling multiple users at the same time while maintaining synchronization across all active sessions.Location data is displayed on an interactive map, where markers are updated dynamically based on user movement. The system also manages device disconnections by automatically removing inactive markers, which helps maintain reliable and accurate tracking.

Overall, the system provides a simple, lightweight, and platform-independent solution for real-time location monitoring.

3.3 Scope

This project focuses on the design and implementation of a web-based system for real-time device location tracking using modern frontend and backend technologies. It includes capturing live location data using the Browser Geolocation API, enabling real-time communication through WebSocket technology, and displaying device locations on an interactive map interface.

The system supports multiple users at the same time and provides continuous location updates without the need for manual page refreshing. It also ensures smooth data transmission, dynamic updating of location markers, and accessibility across different devices through standard web browsers .However, the scope of this project is limited to browser-based tracking and depends on internet connectivity and user-granted location permissions. It does not include advanced features such as large-scale data storage,

integration with external databases, or deployment in enterprise-level environments.

Additionally, aspects such as advanced security mechanisms, large-scale infrastructure integration, and high-level scalability optimization are not covered in detail. These limitations open opportunities for future improvements, such as adding authentication features, cloud integration, and advanced data analysis for better system performance.

4. Proposed Methodology

The proposed methodology focuses on developing a web-based system for real-time device location tracking using modern communication and mapping technologies. The system is designed as an integrated framework where multiple components work together to provide accurate and continuous location updates.

Instead of using traditional request-response methods, the system uses real-time communication to enable instant data transfer and smooth user interaction. This approach helps in capturing live location data efficiently and displaying it dynamically on an interactive map.The methodology combines key processes such as location data collection, data transmission, server-side processing, and visualization into a single workflow. This ensures proper coordination between all system components and supports continuous tracking.

Real-time data exchange between the client and server allows synchronization of location information across multiple users. The system also manages device connections effectively by handling active sessions and updating location markers dynamically based on user activity.Overall, this methodology ensures accurate tracking, low latency, and a responsive user experience across different devices and platforms.

4.1 Framework

The proposed framework is designed as a real-time, event-driven architecture consisting of multiple interconnected modules that work together to provide continuous device location tracking.

The process begins with the client-side module, where the system captures the device's location using the Browser Geolocation API. The collected data is then prepared and sent to the server through a real-time communication channel.

The transmission module uses WebSocket technology to transfer location data from the client to the server with minimal delay. On the server side, the data is processed, and each connected device is assigned a unique identifier. The server then broadcasts updated location information to all active users, ensuring that every client receives synchronized and real-time updates.The visualization module displays device locations on an interactive map using mapping libraries. It dynamically updates markers based on user activity by adding, modifying, or removing them as needed. This helps maintain accurate tracking of active devices.

The framework also includes connection and session management, which monitors device status and handles disconnections efficiently. When a device becomes inactive, its corresponding marker is removed to maintain data accuracy. The overall system is supported by a lightweight backend that manages communication and ensures smooth data flow. A web-based interface allows users to access the system from different devices, making it flexible and easy to use. Overall, this framework ensures efficient data handling, low latency, and a smooth real-time tracking experience across multiple platforms.

4.2 System Design

The system is designed as a real-time, web-based architecture that enables continuous tracking of device locations through efficient communication between client and server components. On the client side, the Browser Geolocation API is used to capture the device's latitude and longitude at regular intervals. This data is then transmitted to the server using WebSocket communication, which ensures low latency and supports bidirectional data exchange. This approach removes the need for repeated HTTP requests and enables smooth real-time updates.

On the server side, a lightweight backend built using Node.js and Express.js handles incoming location data and manages active client connections. Each connected device is assigned a unique identifier, allowing the system to track multiple users efficiently. The server processes the received data and broadcasts updated location information to all connected clients using Socket.IO, ensuring synchronized tracking across devices.

The visualization component displays device locations on an interactive map using mapping libraries. Location markers are dynamically added, updated, or removed based on device activity, ensuring accurate representation of active users. The system also detects device disconnections and removes inactive markers automatically. Overall, the system design ensures efficient data flow, reduced latency, and a responsive user experience across different devices and platforms.

4.3 System Architecture

The system architecture is designed using a client-server model that supports real-time communication and efficient data exchange between multiple devices. On the client side, the web browser acts as the main interface, where location data is captured using the Geolocation API. This data is then sent to the server through WebSocket communication, ensuring low latency and continuous updates. The architecture clearly separates frontend and backend responsibilities, improving scalability and maintainability.

On the server side, Node.js and Express.js are used to handle incoming data and manage active client connections. Socket.IO enables real-time, bidirectional communication between the client and server. The server processes the received location data, assigns unique identifiers to connected devices, and broadcasts updated information to all active users.This architecture allows the system to efficiently support multiple users while maintaining synchronization and real-time performance across all connected devices.

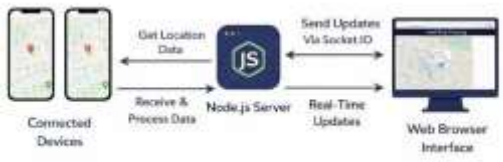

Fig 1: System Architecture of Real Time Device Location Tracking

4.4 Working Mechanism

The working of the system begins when a user accesses the application through a web browser and grants permission to access location data. Once permission is provided, the Geolocation API captures the device's latitude and longitude at regular intervals. This data is transmitted to the server using Socket.IO, enabling real-time communication without requiring page refresh.

The server receives the location data and immediately broadcasts it to all connected clients. On the client side, the received data is used to update markers on an interactive map. When a new user joins, a new marker is created, and when a user disconnects, the corresponding marker is removed automatically. This continuous process of data collection, transmission, and visualization ensures accurate, synchronized, and real-time tracking of all active devices.

4.5 System Features

The system provides continuous real-time location updates without the need for manual page refreshing, ensuring a smooth user experience. It supports multiple users simultaneously, allowing tracking of multiple devices on a single interface.

The system is platform-independent and can be accessed from different devices such as mobile phones, laptops, and tablets through a web browser. It also includes dynamic marker management, where markers are automatically added, updated, and removed based on user activity.

Secure and reliable data transmission is maintained through WebSocket communication. The system also handles device disconnections efficiently by removing inactive markers, ensuring accurate tracking. Overall, the system is lightweight, scalable, and suitable for real-time applications.

4.6 Algorithm

Step 1: User opens the application in a browser and grants location permission.
Step 2: Geolocation API captures the device's latitude and longitude.
Step 3: Location data is sent to the server using WebSocket communication.
Step 4: Server receives data, assigns a unique ID, and maintains the connection.
Step 5: Server broadcasts updated location data to all connected clients.
Step 6: Clients receive data and update markers on the map interface.
Step 7: If a client disconnects, the server detects it and removes the marker.
Step 8: The process repeats continuously for real-time tracking.

This step-by-step process ensures continuous location updates and synchronization across all connected devices.

5. Experimental Setup

The experimental setup is designed to evaluate the performance of the real-time location tracking system in a browser-based environment. The system is executed on a Node.js server, while multiple client devices access the application through web browsers. These devices provide location data using the Geolocation API, and real-time communication is maintained using Socket.IO.

The system is tested under different conditions such as varying time intervals and device movements to analyze tracking accuracy, responsiveness, and overall performance.

5.1 Hardware and Software Environment

The system is implemented using a standard computing environment suitable for real-time web applications. The hardware setup includes a laptop or desktop with a multi-core processor and sufficient memory to handle server operations. Mobile devices are also used to test the system across different platforms.

The software environment consists of Node.js as the backend runtime and Express.js for server-side processing. Socket.IO is used for real-time communication between the client and server. On the frontend, HTML, CSS, and JavaScript are used to build the interface, while EJS is used for rendering dynamic content. The Geolocation API is used to capture location data, and Leaflet.js with OpenStreetMap is used to display the data on an interactive map. The system is tested on browsers such as Chrome and Firefox to ensure compatibility.

5.2 Assumptions and Limitations

The system assumes that users grant permission to access their location, as tracking depends on the Geolocation API. It also requires a stable internet connection for real-time communication through WebSocket technology. Additionally, the system is designed to work on modern browsers that support these features.

There are certain limitations in terms of accuracy and external dependencies. Tracking accuracy may vary due to network delays, GPS signal fluctuations, and device movement. The system does not include features such as historical data storage or advanced analytics, as it focuses mainly on real-time tracking. Large-scale deployment may require further optimization of server performance.

5.3 Performance Metrics

The performance of the system is evaluated based on tracking accuracy, response time, and update frequency. The system continuously updates location data at regular intervals, ensuring a smooth tracking experience.

The use of WebSocket communication helps reduce delay and improves responsiveness. Accuracy is generally high at the initial stage but may vary slightly over time due to network and environmental factors. However, the system maintains a stable level of performance suitable for real-time applications.

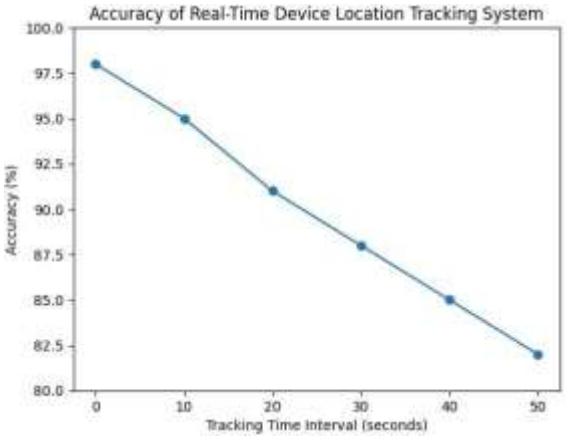

Figure 2:Accuracy of Real-Time Device Location Tracking System

Figure 3 presents the performance of the proposed model using accuracy, precision, and F1-score across evaluation epochs. The results show consistent improvement, demonstrating the model's effectiveness in reliable tumor classification.

5.4 Experimental Results

The system is tested under different conditions to evaluate its ability to track device locations in real time. The results show that the system successfully captures and updates location data continuously without requiring manual page refresh.

Multiple devices can connect to the system simultaneously, and their locations are displayed accurately on the map interface. The system also handles user activity effectively by adding markers when devices connect and removing them when devices disconnect. This ensures proper synchronization and reliable tracking.

5.5 Performance Analysis

The system performance is analysed based on its responsiveness and consistency during operation. It performs efficiently over short time intervals, providing fast and accurate updates.

Over longer durations, slight variations in accuracy may occur due to factors such as network latency and GPS signal strength. Despite these variations, the system continues to function smoothly and provides uninterrupted real-time updates. The lightweight design and efficient communication mechanism contribute to stable performance.

5.6 Visual Representation

The system provides a visual representation of device locations through an interactive map interface. Each device is shown as a marker that updates dynamically as the device moves.

The interface supports multiple users and displays all active devices simultaneously. Continuous marker updates provide a clear and accurate view of location changes, improving user understanding and interaction.

Fig 3: Real-Time Device Location Tracking

5.7 System Demonstration

The system demonstrates effective real-time tracking when users access the application and grant location permission. It continuously captures and transmits location data to the server, ensuring instant updates.

The system efficiently manages multiple users by assigning unique identifiers to each device. It updates location data dynamically and handles connections properly. When a device disconnects, it is automatically removed from tracking, ensuring accurate and consistent results. This demonstrates the reliability and smooth operation of the system in real-time scenarios.

6. Discussions

The experimental results yield significant information regarding the performance and effectiveness of the proposed real-time device location tracking system. The system shows that it can continuously capture, send, and display location data with minimal delay, which proves that the web-based architecture is effective. Bringing together real-time communication tools with interactive map visualisation makes sure that location updates are displayed correctly, which makes the tracking process more accurate.

The system's ability to keep communication low-latency using WebSocket-based technology is one of the most important things to note about the results. The use of a socket.IO lets the client and server send and receive data in both directions, so there is no need to make multiple HTTP requests. This cuts down on communication costs by a lot and makes sure that updates are sent almost right away. Because of this, the system can track things smoothly and continuously, which is necessary for applications that need to monitor things in real time.

Another important thing is how well the interactive map visualisation works. Combining Leaflet.js with OpenStreetMap makes it easy to see where devices are located on a clear and responsive interface. Markers that update automatically let users see movement in real time without having to refresh the page. This makes the system easier to use and more intuitive, even when tracking more than one device at a time.

The system also shows that it can grow easily. It was noted during testing that several devices could be tracked at the same time with only small changes in response time. The backend's event-driven, non-blocking architecture makes it easy to handle many client connections at once. This means that the system can be expanded to handle big applications like managing a fleet, tracking logistics, and keeping an eye on assets.

The evaluation, however, identify a few issues. GPS signal strength and network conditions are two examples of outside factors that can affect how accurately location tracking works. In places where the signals are weak or the latency is high, location updates may be delayed or changed by a small amount. The system also needs a constant internet connection, which could slow things down when there isn't one or when the connection is weak.

The proposed system is more responsive and efficient than traditional tracking systems that use HTTP-based updates on a regular basis. Using WebSocket communication keeps data flowing all the time, which makes the device, server, and user interface work better together. This shows how useful modern real-time technologies can be for location-based services.

From an application standpoint, the system is validated as a functional, viable option for actual deployment. It can provide location data to increase safety (personal), vehicle management, and logistics. Because of the web-based user interface, it is very user-friendly; this helps make it more usable from multiple locations/platforms without the need to go through complicated installation procedures.

Overall the results show that the proposed system fulfills the main goals of real-time location tracking

with an acceptable level of accuracy, responsiveness to events, and capacity for further expansion. There are opportunities for additional enhancements to the system, including improving location identification accuracy, adding offline capabilities, and implementing advanced features (e.g., geofencing, historical tracking). With additional enhancements, the system can become an even more versatile application capable of supporting various types of physical environments.

7. Conclusion

This project uses the Geolocation API, Socket.IO, Express, and Leaflet to provide an efficient, user-friendly, and secure real-time location tracking solution. The system is very flexible and offers useful applications and multiple users can access server simultaneously , logistics, and navigation. It is a scalable solution for a variety of location-tracking requirements due to its platform independence and simplicity of integration.It uses web socket technology which is a full duplex bi-directional communication.

The requirements for performance include that location updates occur in real-time, with no noticeable lag between updates. The requirements for scalability allow for simultaneous tracking of multiple users without compromising performance by using Socket.IO as an underlying technology. Reliability means that the system will provide continuous tracking of a user's location without crashing or losing data while operating; therefore, there is no lack of reliability between the user's location and other users' locations.

Security is concerned with safe communication between the client and the server via secure socket connections. Usability refers to the fact that the system has a user interface that is simple and easy to use, allowing users to easily access and view live location updates on the map.

In conclusion, the developed system demonstrates the effective use of real-time web technologies in building a scalable and efficient location tracking solution. It highlights the potential of such systems in addressing real-world challenges and provides a basis for the development of more advanced location-based services in the future.

8. Future Work

The proposed real-time device position-tracking system has been proven to perform well; however, there are options for making improvements to enhance functionality or use for other applications. One option is incorporating a database such as MongoDB that would provide historical data storage capabilities, allowing for use in path tracking, data analysis, and long-term monitoring. This is very important for applications including fleet management and logistics.

Another way to improve the current system is implementing geofencing and real-time alerting capabilities. By setting virtual boundaries, the system will be able to send alerts whenever a device enters or leaves a specified area, or if abnormal conditions exist, such as excessive speed or loss of connection. This would provide greater value in the security and monitoring applications.

Future work may also be directed toward improving the accuracy of location via GPS data, combined with advanced techniques such as sensor fusion and machine-learning prediction models. This will reduce the impact of weak-signal conditions or environmental factors when a device is located indoors or in highly populated urban areas. At the same time, expanding on this idea would allow for greater flexibility in deploying the system with offline capability through temporary storage of location information on mobile devices; when network connectivity is available again, that stored data would upload to the server. Reliability would therefore increase in areas with unreliable networks.

Finally, the system could integrate or interface with mobile apps as well as IoT (devices) thus creating further potential for broader application. Security and privacy could be enhanced through additional features including verifying user identity before giving access, controlling who has what kind of access to what kinds of data, and providing encryption of sensitive data stored within the service (i.e., using symmetric or asymmetric methods).

9. References

[1] Secure Android Location Tracking Application with Privacy Enhanced Technique, Proc. IEEE CCiCT, 2022, doi: 0.1109/CCiCT56684.2022.00050.

[2] Design and Implement a GPS Car Tracker on Google Maps Using Arduino, Proc. FRUCT Conf., 2024, doi: 10.23919/FRUCT61870.2024.10516353.

[3] Google Maps API Implementation on IoT Platform for Tracking an Object Using GPS, Proc. IEEE,APWiMob,2020,doi:10.1109/APWiMob48441 .2019.8964139.

[4] A GPS-GSM Predicated Vehicle Tracking System Monitored in a Mobile App Based on Google Maps, Proc.IEEE,ICECDS,2017,doi:10.1109/ICECDS.2017 .8389989.

[5] Survey of Decentralized Solutions with Mobile Devices for User Location Tracking, Proximity Detection, and Contact Tracing in the COVID-19 Era, Data (MDPI), 2020, doi: 10.3390/data5040087.

[6] An Intelligent Metro Tracking System for Riyadh Smart City, Springer, 2020, doi: 10.1007/s41870-020-00435-7.

[7] Comprehensive Review: Privacy Protection of User in Location-Aware Services of Mobile Cloud Computing, Springer, 2020, doi: 10.1007/s11277-019-06872-3.

[8] Applicability of Mobile Contact Tracing in Fighting Pandemic (COVID-19): Issues, Challenges and Solutions, Computer Science Review (Elsevier), 2020, doi: 10.1016/j.cosrev.2020.100307.

[9] Real-Time Uber Data Analysis of Popular Uber Locations in Kubernetes Environment, Proc. IEEE ICITR,2020,doi:10.1109/ICITR51448.2020.9310851

[10] Outdoor Location Tracking of Mobile Devices in Cellular Networks, EURASIP Journal on Wireless Communications and Networking (Springer), 2019, doi: 10.1186/s13638-019-1459-4.

[11] Bus Tracking System Using Mobile GPS Technology, Proc. IEEE ICICT, 2024, doi:10.1109/ICICT60155.2024.10544750.

[12] Estimation of Bus Arrival by Tracking and Analyzing the Users Real-Time Location, Proc. IEEE ICSCAN,2023,doi:10.1109/ICSCAN58655.2023.10395687.

[13] Mobile Application Based Tracking Using GPS and GSM, Proc. IEEE ICSC, 2022, doi:10.1109/ICSC56524.2022.10009250.

[14] Secure Android Location Tracking Application with Privacy Enhanced Technique, Proc. IEEE CCiCT, 2022, doi:10.1109/CCiCT56684.2022.00050.

[15] Understanding Travel Tracking Mobile Application Usage: An Integration of Self Determination Theory and UTAUT2, Tourism Management Perspectives (Elsevier), 2022, doi: 10.1016/j.tmp.2022.100949.

[16] The Use of Leaflet and GeoJSON Files for Creating the Interactive Web Map of the Preindustrial State of the Natural Environment, Cartography and Geographic Information Science (Taylor & Francis), 2020, doi:10.1080/14498596.2020.1713237.