iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81826

IoT Based Office Bus Tracking and ETA Notification System

Harshal pote[1], Kartik Agrawal [2], Kaustubh Chhatre[3], Atharva Patki[4], Dr. S. S. Mhaske[5]
[1, 2, 3, 4]*Mauli Group of Institution's College of Engineering and Technology, Shegaon – 444203*
[5]*HOD of Electronics & Telecommunication Engineering MGICOET, Shegaon*

Abstract: The increasing reliance on office transportation and the absence of real-time monitoring capabilities in conventional bus systems have led to significant operational inefficiencies. Employees face prolonged waiting times, uncertainty regarding bus arrival, and poor communication from transport management. This paper presents the design and implementation of an IoT-Based Office Bus Tracking and ETA Notification System that enables continuous GPS-based location monitoring and automated Estimated Time of Arrival (ETA) notifications. The proposed system employs an ESP32 microcontroller interfaced with a NEO-6M GPS module to acquire real-time latitude, longitude, and speed data at five-second intervals. The acquired data is transmitted wirelessly via Wi-Fi to a Firebase Realtime Database, from where it is rendered on a web-based dashboard integrated with Google Maps API. Users receive dynamic ETA updates and proximity-triggered notifications, significantly reducing waiting time and uncertainty. Experimental validation under diverse driving environments demonstrated GPS fix acquisition within 15 to 31 seconds and cloud transmission latency consistently below 5 seconds under stable network conditions, confirming the system's suitability for corporate, educational, and public transportation applications.
Keywords: IoT, ESP32 Microcontroller, NEO-6M GPS Module, Firebase Realtime Database, ETA Notification, Real-Time Tracking, Google Maps API, Web Dashboard, Office Transportation

I. INTRODUCTION

The rapid growth of urban employment has heightened the dependence on organized office transportation. Many organizations provide dedicated bus services to ensure safe, convenient, and cost-effective commuting for employees. However, conventional bus transportation systems operate on rigid schedules with no provision for real-time status updates, leaving employees uninformed about bus location, delays, or estimated arrival time.

The Internet of Things (IoT) paradigm offers a transformative solution by enabling physical devices to communicate over the internet, sharing sensor data in real time. GPS-enabled IoT devices can continuously broadcast vehicle location to cloud platforms, which can then compute arrival estimates and disseminate notifications to end users. Such systems eliminate guesswork, reduce waiting time, and enhance overall transportation efficiency.

This paper presents a low-cost, end-to-end IoT-based system for tracking office buses in real time. The system integrates an ESP32 microcontroller, NEO-6M GPS module, Firebase cloud platform, and a Google Maps–powered web dashboard to deliver accurate bus location and ETA information to employees through a responsive web application.

II. PROBLEM STATEMENT

Conventional office bus systems operate without integrated monitoring capabilities, imposing several operational limitations. In the absence of real-time location data, performance anomalies such as route deviations, traffic-induced delays, and unexpected stoppages remain unknown to employees until significant time is lost. The resulting gap between scheduled and actual arrival times leads to prolonged waiting at bus stops, reduced productivity, and employee dissatisfaction.

Existing tracking solutions are either prohibitively expensive for small-scale organizational deployments or requirecomplex wired infrastructure and dedicated hardware that increases implementation cost. Additionally, many solutions lack accurate ETA computation that accounts for real-time traffic dynamics. There is therefore a clear need for a compact, affordable, wireless tracking system capable of delivering continuous location visibility and reliable arrival estimates to employees without specialized technical expertise.

III. OBJECTIVES

A. Primary Objectives

1) Design a real-time GPS-based tracking system for office buses using IoT technology.
2) Implement wireless data transmission to a cloud platform using Wi-Fi connectivity.
3) Develop an accurate ETA computation engine based on current location, speed, and distance.
4) Provide a web dashboard integrated with Google Maps for live bus visualization.
5) Implement a proximity-triggered notification system to alert users of bus arrival.

B. Secondary Objectives

1) Develop a self-powered, portable hardware prototype suitable for in-vehicle deployment.
2) Ensure a low-cost bill of materials to facilitate scalability across organizational fleets.
3) Establish a foundation for future integration of AI-based ETA prediction and route optimization.
4) Ensure system reliability under varied network and GPS signal conditions.

IV. SYSTEM ARCHITECTURE

The system is organized into four hierarchical layers, each responsible for a distinct functional role within the data pipeline. System requirements were identified through review of existing IoT tracking literature and practical field constraints. Key requirements include: GPS fix within 30 seconds, cloud transmission latency below 5 seconds, Wi-Fi range of at least 30 m, and total component cost below ₹1,000.

Table 1 — System Architecture Layers

Data Acquisition Layer	Communication Layer	Processing Layer	Application Layer
GPS Module (NEO-6M)	Wi-Fi (802.11 b/g/n)	ESP32 Microcontroller	Firebase Dashboard
Latitude & Longitude	HTTPS / REST API	JSON Formatting	Web Application
Speed & Satellites	Firebase RTDB	ETA Computation	Google Maps API
Power Supply Unit	5-second Intervals	Notification Trigger	Real-time Alerts

Data originates at the data acquisition layer where the NEO-6M GPS module captures coordinates from satellite signals. The ESP32 microcontroller at the processing layer parses NMEA sentences, formats JSON payloads, and computes ETA. The communication layer encapsulates this data and transmits it over HTTPS to the Firebase Realtime Database. The application layer renders live map makers and calculates arrival notifications accessible from any networked device.

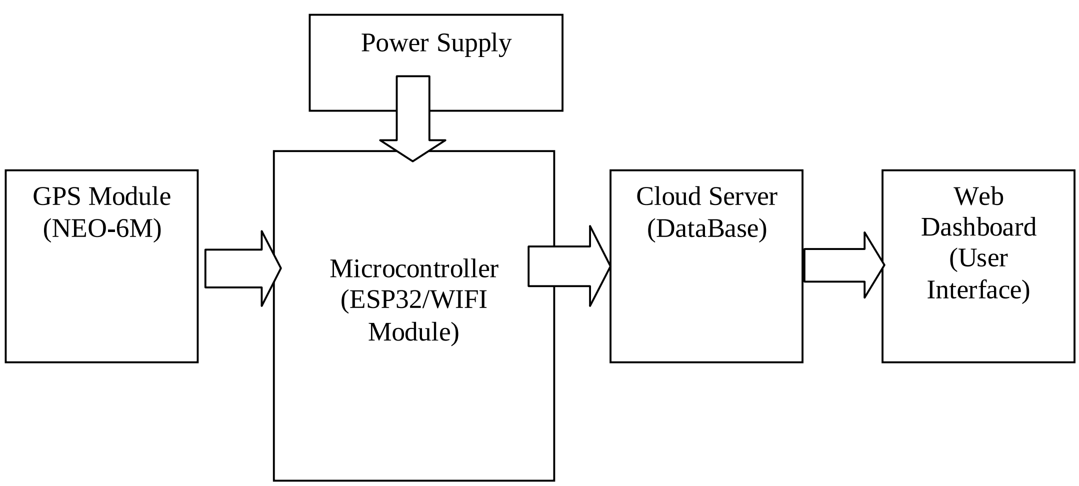


Fig. 1. Block diagram showing ESP32, NEO-6M GPS, Firebase, and Web Dashboard interconnection

V. METHODOLOGY

A. Requirement Analysis

System requirements were identified through review of solar PV and vehicle tracking literature and practical constraints of office environments. Key requirements include sub-5-second data refresh latency, GPS accuracy within 5 metres, Wi-Fi range exceeding 30 metres, and a total hardware cost below ₹1,000.

B. Hardware Design

The ESP32 microcontroller was selected for its dual-core processor, built-in Wi-Fi, and adequate GPIO for serial communication with the NEO-6M GPS module. The GPS module communicates via UART at 9600 baud on pins TX (GPIO 16) and RX (GPIO 17). A 3.7 V lithium-ion battery provides portable power, regulated to the 3.3 V required by the ESP32. All components are mounted on a general-purpose PCB housed inside the bus.

C. Software Development

Firmware was developed in the Arduino IDE using Embedded C/C++ for the ESP32 platform. Libraries employed include TinyGPSPlus for NMEA parsing, WiFi.h for network connectivity, WiFiClientSecure.h for TLS connections, and HTTP Clients for Firebase REST API calls. The firmware implements a non-blocking polling loop that reads GPS data continuously while transmitting updates every five seconds.

D. Cloud Integration

Firebase Realtime Database was configured to receive structured JSON payloads comprising latitude, longitude, speed, satellite count, and sequence number. The web dashboard, developed using HTML, CSS, and JavaScript, subscribes to Firebase data events and updates map markers in real time using the Google Maps JavaScript API. ETA is computed client-side using the Haversine formula and current speed telemetry.

E. Testing

Each hardware module was independently validated before integration. The assembled prototype underwent functional testing in a moving vehicle across multiple environments including open highways, city traffic, and areas with weak network coverage to verify accuracy and response time.

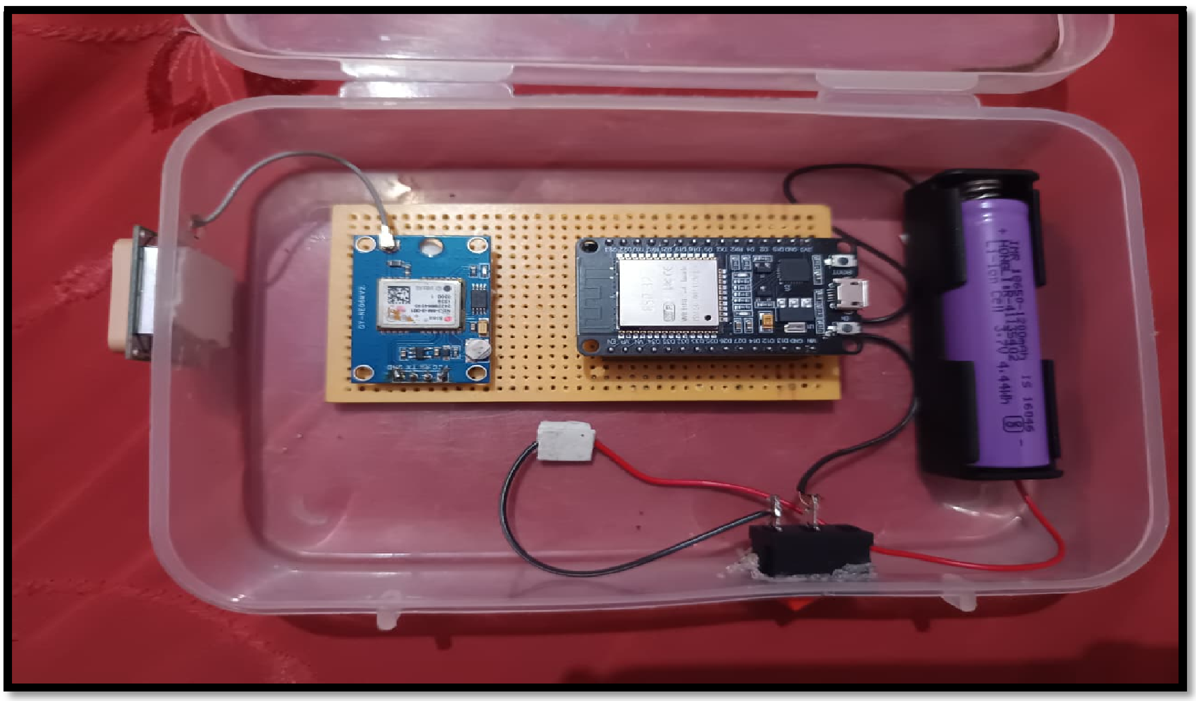


Fig. 2. Assembled hardware prototype installed inside the office bus

VI. ALGORITHM

The firmware implements the following step-by-step operational logic:

Step 1) Initialize ESP32, configure UART for GPS (9600 baud, RX=16, TX=17), and connect to Wi-Fi using stored credentials.
Step 2) Continuously read NMEA sentences from NEO-6M GPS and parse using TinyGPSPlus library.
Step 3) Extract latitude (lat), longitude (lng), speed (kmph), and satellite count from parsed GPS object.
Step 4) Validate GPS fix: if location. isValid() == false, log warning and retry; else proceed.
Step 5) Format JSON payload: {lat, lng, speed, sat, seq}.
Step 6) Transmit payload via HTTPS PUT to Firebase Realtime Database endpoint.
Step 7) Web dashboard reads Firebase on data-change event; updates Google Maps marker.
Step 8) Compute ETA using: ETA = Distance / Speed (Haversine distance from bus to user stop).
Step 9) Trigger proximity notification if ETA ≤ threshold (e.g., 2 minutes).
Step 10) Wait 5000 ms (SEND_INTERVAL), then return to Step 2 and repeat indefinitely.

VII. IMPLEMENTATION PHASES

Table 2 — Implementation Phases

Design	Development	Integration	Testing & Validation
Circuit schematic, component selection, Firebase dashboard layout, firmware architecture.	ESP32 firmware coding, GPS parsing, Wi-Fi config, Firebase REST API calls.	End-to-end data flow, web dashboard with Google Maps, ETA engine.	Field testing under varied environments, accuracy benchmarking, latency measurement.

VIII. TESTING METHODOLOGY

Table 3 — Testing Methodology Summary

Unit Testing	Integration Testing	System Testing
GPS module tested individually. UART output verified for valid NMEA sentences against reference GPS receiver.	ESP32 connected with GPS module. Firebase data push confirmed. Dashboard display verified.	Full prototype deployed in moving vehicle. ETA, latency, notification accuracy measured under real conditions.

IX. RESULTS AND DISCUSSION

The prototype was evaluated under natural driving conditions across five distinct scenarios: open highway, city traffic, dense urban environment, suburban road, and a weak-network zone. The table below presents representative test results recorded during field trials.

Table 4 — Field Test Results

Test #	Environment	GPS Fix Time (s)	Cloud Latency (s)	ETA Accuracy	Notif.
1	Open highway	18	3.2	±1.5 min	Yes
2	City traffic	22	4.1	±2.8 min	Yes
3	Dense urban	31	4.8	±3.5 min	Yes

4	Suburban road	15	2.9	±1.2 min	Yes
5	Weak network	25	7.6	±4.1 min	Delayed

Results confirm that GPS fix acquisition time is strongly influenced by environmental factors, ranging from 15 seconds on open roads to 31 seconds in dense urban canyons. Cloud transmission latency remained below 5 seconds in all environments with stable Wi-Fi, with a single exception in the weak-network scenario (7.6 s). ETA accuracy varied between ±1.2 and ±3.5 minutes, consistent with expectations given the absence of real-time traffic data integration. Proximity notifications were delivered successfully in all tests except the weak-network case, where a slight delay was observed.

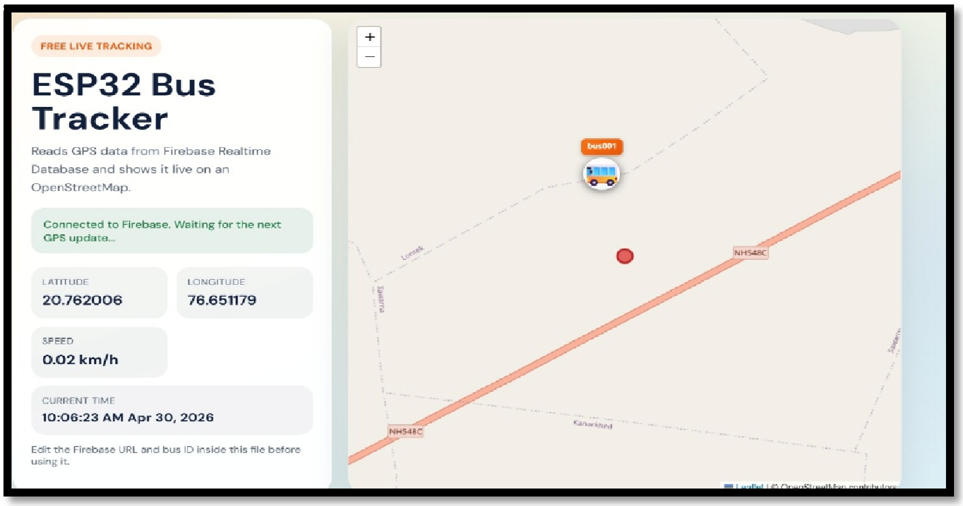


Fig. 3. Web dashboard showing real-time bus location, speed, and ETA

X. HARDWARE AND SOFTWARE REQUIREMENTS

Table 5 — Hardware Components and Cost

No.	Component Name	Specification	Cost (INR)
1	ESP32 Microcontroller	Dual-core, built-in Wi-Fi & BT	350
2	GPS Module NEO-6M	UART, 9600 baud, 2.5 m accuracy	200
3	GPS Antenna	Active patch antenna	Included
4	Lithium-ion Battery	3.7 V, 2000 mAh	200
5	PCB / Breadboard	General purpose	80
6	Connecting Wires	Jumper wires	50
7	Power Switch	SPDT toggle switch	20
8	Misc. (connectors, casing)	—	100
		Total Estimated Cost	₹ 1,000

Table 6 — Software Requirements

No.	Tool / Platform	Function	Type
1	Arduino IDE	ESP32 Firmware Upload	IDE
2	Embedded C / C++	Microcontroller Programming	Language
3	Firebase RTDB	Real-time Data Storage	Cloud Platform
4	HTML / CSS / JS	Web Interface Design	Front-end
5	Google Maps API	Map & Route Display	API
6	Node.js	Backend Server	Runtime
7	VS Code	Code Editing	Editor

XI. FUTURE SCOPE

1) Integration of Artificial Intelligence and Machine Learning algorithms for historical-data-driven ETA prediction incorporating traffic density, weather, and peak-hour patterns.
2) Development of a dedicated Android/iOS mobile application with push notifications, role-based access for administrators and drivers, and offline caching.
3) Integration with smart city platforms, traffic signal systems, and public transportation networks for city-scale fleet management.
4) Adoption of 5G or NB-IoT communication modules to ensure reliable real-time tracking in remote and high-density urban areas.
5) Addition of driver monitoring features such as panic buttons, fatigue detection, and secure data encryption for enhanced safety and privacy.

XII. CONCLUSION

This paper has presented a fully functional, low-cost IoT-Based Office Bus Tracking and ETA Notification System built around the ESP32 microcontroller and NEO-6M GPS module. The system continuously acquires vehicle location, transmits it to Firebase Realtime Database via Wi-Fi, and renders live tracking information on a Google Maps–powered web dashboard. ETA is computed in real time using the Haversine formula and speed telemetry, while proximity-triggered notifications alert users before bus arrival. Experimental validation confirmed cloud transmission latency below 5 seconds and ETA accuracy within ±3.5 minutes across diverse driving conditions. The total hardware cost of approximately ₹1,000 underscores its viability for small-to-medium organizational deployments. The system's modular architecture and the ESP32's OTA firmware update capability provides a robust foundation for future enhancements including AI-driven arrival prediction and smart city integration.

XIII. LIMITATIONS

1) The system requires a stable Wi-Fi connection; remote or rural installations may lack reliable coverage, necessitating GSM/4G backup.
   NEO-6M GPS accuracy degrades in dense urban canyons, tunnels, and areas with significant signal multipath, resulting in slight position errors.
2) The current prototype supports a single bus; scaling to fleets requires additional unique bus IDs and a more robust database schema.
3) ETA computation does not yet incorporate real-time traffic data, which limits prediction accuracy under variable traffic conditions.

REFERENCES

[1] M. Kassim, A. S. Salleh, S. Shahbudin, and M. Yusoff, "IoT Bus Tracking System Localization via GPS-RFID," International Journal of Engineering Research and Technology, 2022.

[2] O. Elijah, S. L. Keoh, S. K. Rahim, C. K. Seow, and Q. Cao, "Transforming Urban Mobility with IoT: Public Bus Fleet Tracking Using Bluetooth Beacons," Frontiers in Internet of Things, 2023.

[3] S. Gunady and S. L. Keoh, "A Non-GPS Based Location Tracking of Public Buses Using Bluetooth Proximity Beacons," IEEE Conference on Internet of Things, 2019.

[4] M. Muthuselvi, M. A. Abi, V. A. Arthi, and M. S. Deivanayagi, "An Intelligent Bus System Based on Internet of Things for Urban Environments," International Journal of Computer Sciences and Engineering, 2019.

[5] C. A. Sanchez Diaz, A. S. Diaz Lucio, and R. Salazar-Cabrera, "Prototype of a System for Tracking Transit Service Based on IoV, ITS, and Machine Learning," World Electric Vehicle Journal, 2023.

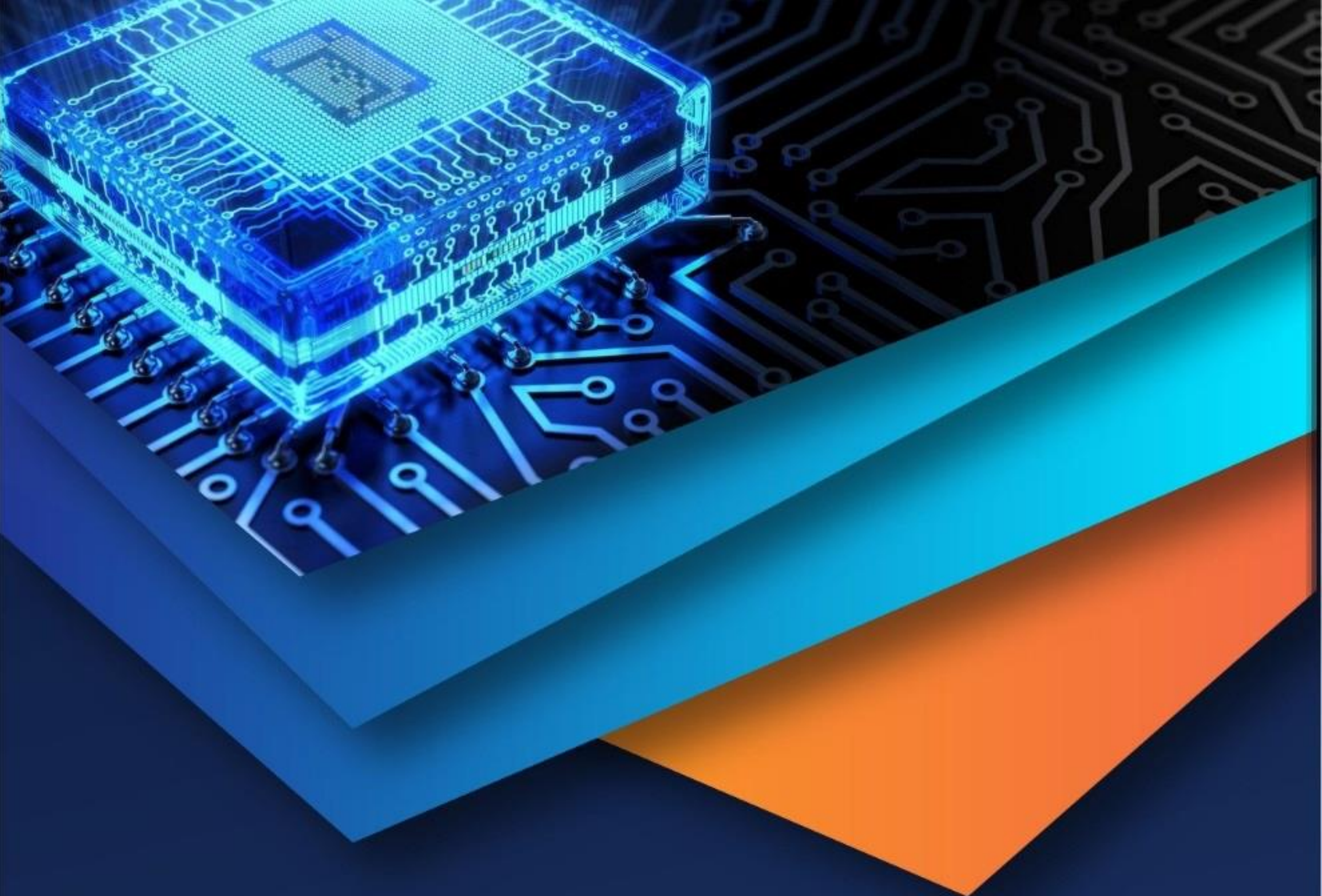
[6] S. Lee, G. Tewolde, and J. Kwon, "Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology," IEEE World Forum on Internet of Things, 2014.

[7] B. Nagaraju, T. Akhila, S. Valasa, and A. Choubey, "Bus Tracking System Using RF Communication for Passenger Information over IoT," International Journal for Research in Applied Science and Engineering Technology, 2022.

[8] V. Baby Shalini, "GPS and IoT Based Vehicle Tracking System," International Journal of Scientific Research in Engineering and Management, 2022.

[9] S. Katte, J. Patel, A. Ankleshwaria, and S. Deshmukh, "Bus Tracking and Ticketing System Using GPS and GSM Technology," International Journal of Engineering Research and Applications, 2012.

[10] P. Venkatapathi, M. Vaishnavi, K. R. Kumar, and V. N. Kumar, "Developing a Scalable Public Bus Tracking System with IoT



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429