IOT BASED VEHICLE TRACKING SYSTEM

Name: Venkatesh.Nayak(21211A04R2)

Institutional Affiliation: BVRIT,Narsapur

Date:31/01/2024

Abstract:

In today's society, security is a top priority, whether safeguarding our assets, homes, or loved ones. GPS tracker devices serve as invaluable tools in addressing these concerns by offering real-time monitoring capabilities for vehicles, assets, and even children. With the ability to track locations swiftly, they provide crucial assistance during emergencies such as theft or accidents. In our project, we aim to develop a GPS tracking device for monitoring vehicle locations remotely. Utilizing the Thingspeak IoT cloud, we'll archive the vehicle's journey history, enhancing security measures further. Building upon our previous experience interfacing GPS with NodeMCU ESP8266, we'll display location coordinates on a webpage. Additionally, this IoT Vehicle Tracking System will incorporate a feature allowing users to access a link on the webpage, directing them to Google Maps for a visual representation of the vehicle's location. By combining advanced technology with user-friendly interfaces, we empower individuals to proactively ensure the safety and security of their assets and loved ones.

1. Introduction:

In today's security-conscious society, safeguarding assets and loved ones is paramount, prompting the widespread adoption of GPS tracker devices. These devices offer real-time tracking capabilities crucial for monitoring vehicles, assets, and even children, aiding in emergencies like theft or accidents. This project focuses on building a GPS tracking device to remotely monitor vehicle locations, utilizing the Thingspeak IoT cloud to securely store journey histories. Drawing from prior experience with interfacing GPS via NodeMCU ESP8266 and displaying coordinates on a webpage, this IoT Vehicle Tracking System introduces a new feature: a webpage link directing users to Google Maps for visualizing the vehicle's current location. By integrating advanced technology with user-friendly interfaces, the project aims to empower individuals to proactively address security concerns, enabling them to effectively safeguard their assets and loved ones in today's dynamic environment..

2. Literature Review:

1. GPS Tracker Devices in Security: Recent research emphasizes GPS tracker devices' role in enhancing security for vehicles, homes, and personal safety. Their real-time tracking capabilities offer proactive measures against theft and accidents, ensuring peace of mind for users concerned about asset security and the well-being of loved ones.

2. IoT Integration for Real-time Monitoring: Integration of GPS tracker devices with IoT platforms enables real-time monitoring and data management. Utilizing services like the Thingspeak IoT cloud facilitates seamless storage and analysis of location data, empowering users to track vehicle movements, access historical data, and make informed decisions regarding security measures.

3. ESP8266 and GPS Integration: Studies demonstrate the integration of GPS modules with ESP8266 microcontrollers for IoT applications. This integration enables efficient retrieval and processing of location data, facilitating real-time tracking and monitoring of vehicles. ESP8266's capabilities make it a versatile platform for building GPS tracking systems with enhanced functionalities and connectivity options.

3.Motivation of the Project:

The project's motivation arises from the paramount importance of security in today's world, where safeguarding assets and loved ones against theft, accidents, and other emergencies is crucial. Recognizing the effectiveness of GPS tracker devices in providing real-time tracking capabilities, the project seeks to develop a reliable system for monitoring vehicle locations remotely. By utilizing the Thingspeak IoT cloud, the project ensures secure storage of location data, enabling users to access a comprehensive history of the vehicle's movements. Additionally, the incorporation of a webpage link to Google Maps enhances user accessibility and convenience, facilitating easy visualization of the vehicle's real-time location. Ultimately, the project aims to address the pressing need for robust security solutions by leveraging GPS technology and IoT integration, thereby offering individuals peace of mind regarding the safety and security of their assets and loved ones.

4.List of Used Components:

→HARDWARE COMPONENTS:

- ESP8266 NodeMCU - 1
- NE06M GPS Receiver - 1
- 16*2 LCD - 1
- 16*2 LCD I2C module - 1
- Breadboard
- Connectors
- Power supply

→SOFTWARE REQUIRED:

- Arduino IDE
- Thingspeak Database
- HTML Webpage

5. Methodology:

Hardware Setup:

- Connect the ESP8266 NodeMCU to the breadboard.
- Connect the NE06M GPS Receiver to the ESP8266 NodeMCU.
- Connect the 16x2 LCD display to the ESP8266 NodeMCU using the I2C module.
- Ensure all connections are secure and properly configured.

Software Configuration:

- Install the necessary libraries for ESP8266, GPS, and LCD display (if not already installed).
- Write code to initialize the GPS module and LCD display, and establish Wi-Fi connection.
- Implement code to read GPS data and display it on the LCD display.
- Configure the code to send GPS data to the Thingspeak IoT cloud for storage.
- Implement code to create a web server on the ESP8266 NodeMCU and serve a webpage.
- Write code to include a link on the webpage that directs users to Google Maps to visualize the vehicle's location.

Testing and Integration:

- Upload the code to the ESP8266 NodeMCU and ensure proper functioning.
- Test the GPS tracking system by monitoring real-time location updates.
- Verify that GPS data is successfully sent to the Thingspeak IoT cloud and stored.
- Test the web server functionality by accessing the webpage and verifying the link to Google Maps.
- Integrate all components together and conduct comprehensive testing to ensure seamless operation of the IoT Vehicle Tracking System.

Hardware Specifications:

ESP8266 NodeMCU:

- Microcontroller: ESP8266
- Operating Voltage: 3.3V
- GPIO Pins: 11 digital I/O pins, 1 analog input pin
- Communication: Wi-Fi connectivity

Programming Environment:

- Arduino IDE
- NE06M GPS Receiver:
- Receiver Type: GPS module
- Communication Protocol: UART (Serial)

- Operating Voltage: 3.3V to 5V
- Interface: TTL level serial port

16x2 LCD Display:

- Display Type: Liquid Crystal Display (LCD)
- Size: 16 characters x 2 lines
- Communication: Parallel or I2C (Inter-Integrated Circuit)
- Operating Voltage: Typically 5V

16x2 LCD I2C Module:

- Interface: I2C (Inter-Integrated Circuit)
- Compatibility: Compatible with standard 16x2 character LCD displays
- Operating Voltage: Typically 5V
- Provides I2C communication for the LCD display, reducing GPIO usage on the ESP8266 NodeMCU.

Connectors:

- Various types of connectors such as jumper wires, Dupont wires, and male-female jumper cables for establishing connections between components on the breadboard.

Power Supply:

- Voltage: Depends on individual component requirements (e.g., 3.3V for ESP8266 NodeMCU, 5V for LCD)
- Source: Can be provided via USB cable, battery pack, or external power supply. Ensure sufficient current capacity to power all components simultaneously.

Additional Components and Software:

Arduino IDE:

- Integrated Development Environment for programming the ESP8266 NodeMCU.
  Thingspeak

Database:

- Cloud-based IoT platform for storing and retrieving historical GPS location data.

HTML Webpage:

- Webpage interface for displaying real-time vehicle location and providing a link to Google Maps.

6. Flow Diagram/Chart:

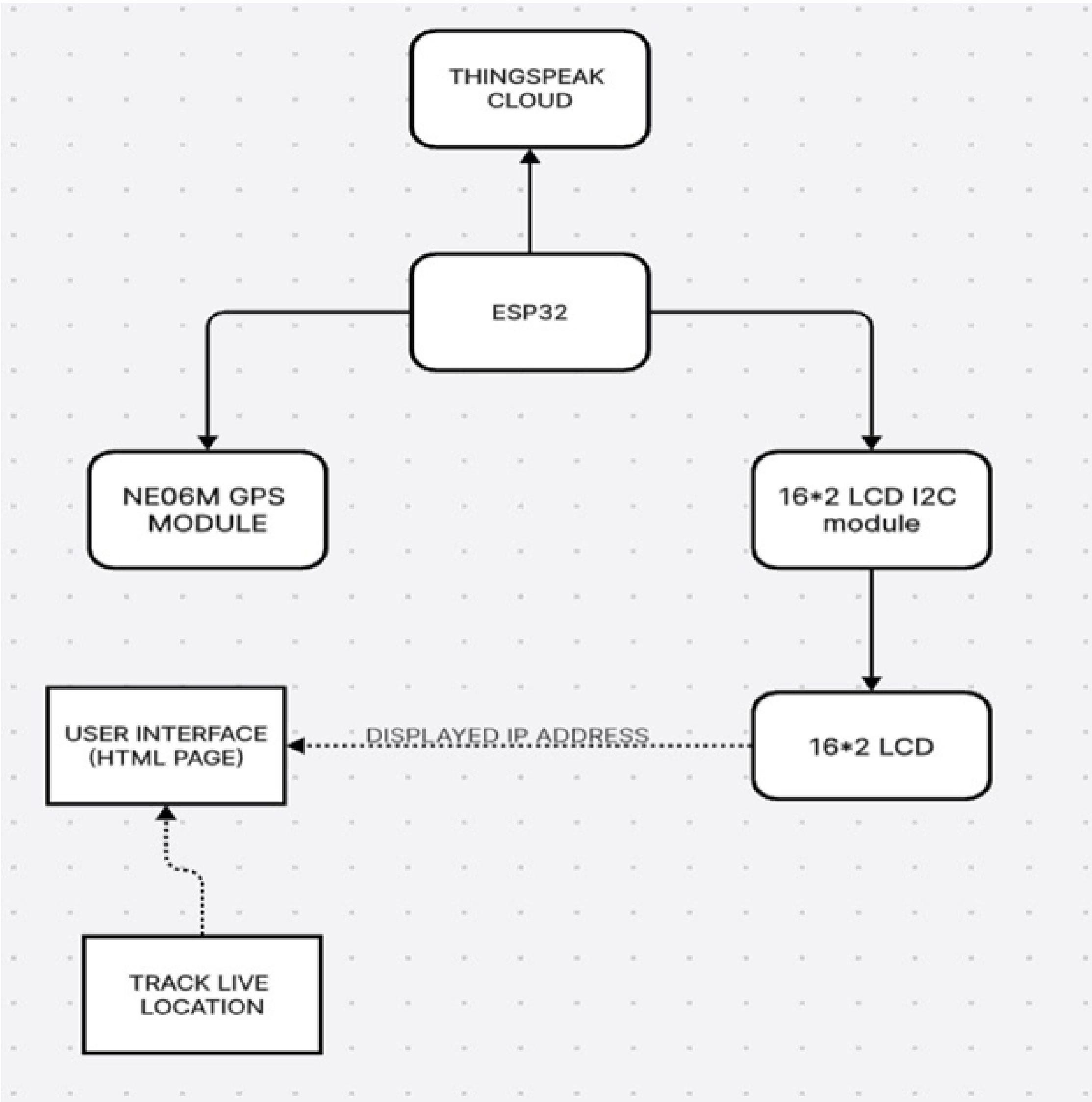

Fig1:Flow chart

6. Implementation:

Implementing the IoT Vehicle Tracking System involves connecting the ESP8266 NodeMCU to the NE06M GPS Receiver and 16x2 LCD Display. Using Arduino IDE, code is developed to read GPS data, display coordinates on the LCD, and transmit them to Thingspeak IoT cloud for storage. Additionally, a web server is created to display real-time location data on a webpage, with a link directing users to Google Maps for visualization. Hardware connections are established on a breadboard, and software functionalities are programmed to ensure seamless integration of GPS tracking capabilities with cloud storage and web-based visualization for efficient vehicle monitoring from anywhere

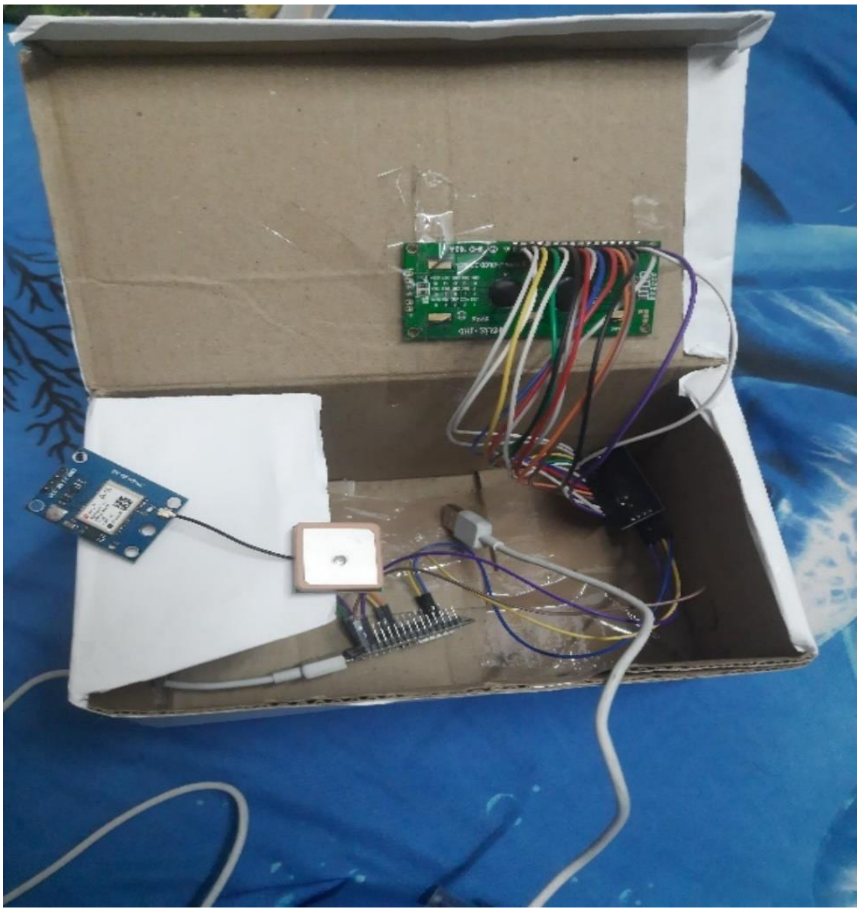

Fig2:Hardware Connections

Code explanation

- Here, *ThingSpeak.h* library is added to use the ThingSpeak platform with NodeMCU and *TinyGPS++.h The* library is used to get the GPS coordinates using the GPS receiver module.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

- For using the I2C module for 16x2 Alphanumeric LCD, configure it using the **LiquidCrystal_I2C** class. Here we have to pass the address, row, and column number which are 0x27, 16, and 2 respectively in our case

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

- Now, declare the network credentials- i.e. SSID and password. It is required to connect the NodeMCU to the internet and declare the connection pins of the GPS module and it's the default baud rate, which is 9600 in our case and declare the ThingSpeak account credentials such as the channel number and write API which was recorded earlier.

```
static const int RX = D6, TX = D7;
static const uint32_t GPSBaud = 9600;
const char *ssid = "M21";
const char *password = "123456780";
unsigned long ch_no = 2413442;
const char *write_api = "BMMUKEB00PKVZT3L";
```

- Then declare the objects for *TinyGPSPlus* and *WiFiClient* class. For using *WiFiServer* properties, the *server* object is defined with Port number 80.

```
TinyGPSPlus gps;
WiFiServer server(80);
WiFiClient client;
SoftwareSerial soft(RX, TX);
String latitude_data;
String longitude_data;
```

- Inside ***setup(),*** declare all the input pins and output pins. Then print a welcome message on the LCD and serial monitor which will be displayed during the initialization of the project. To connect NodeMCU to the internet, call **WiFi.begin** and pass network

SSID and password as its arguments. Check for the successful network connection using *WiFi.status()* and after a successful connection, print a message on LCD with IP address. Then connect to the ThingSpeak platform, using saved credentials. For this *ThingSpeak.begin* is used.

```
void setup() {
  Wire.begin();
  Serial.begin(115200);
  soft.begin(GPSBaud);
  WiFi.begin(ssid, password);
  server.begin();

  // Initialize the LCD
  lcd.init();
  lcd.backlight();

  lcd.setCursor(0, 0);
  lcd.print("WiFi connecting...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    lcd.setCursor(0, 1);
    lcd.print("  "); // Clear the previous IP address
    lcd.setCursor(0, 1);
    lcd.print("  ");
  }

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WiFi connected");
  lcd.setCursor(0, 1);
  lcd.print(" ");
  lcd.print(WiFi.localIP());

  ThingSpeak.begin(client);
}
```

- Inside loop(), ***encode ()*** is used to ensure a valid GPS sentence is received. When **encode ()** returns "true", a valid sentence has just changed the TinyGPS object's internal state. Here two functions namely *displaydata()* *and* *displaywebpage()* are called, when *encode()* returns *true*.

```
void loop() {
```

```
  while (soft.available() > 0) {
    if (gps.encode(soft.read())) {
      displaydata();
      displaywebpage();
    }
  }
  if (millis() > 5000 && gps.charsProcessed() < 10) {
    Serial.println(F("GPS Connection Error!!"));
    Serial.print(F("Number of characters processed by GPS: "));
    Serial.println(gps.charsProcessed());
    while (true);
  }
}
```

- Inside ***displaydata()*** function**,** *isValid()* method is used to ensure a valid latitude and longitude reception and they are stored in respective variables. Then to send these data to ThingSpeak, *setField()* method is used to set the fields and the *writeFields()* method is used to send these data to the cloud.

```
void displaydata() {
  if (gps.location.isValid()) {
    double latitude = gps.location.lat();
    double longitude = gps.location.lng();
    latitude_data = String(latitude, 6);
    longitude_data = String(longitude, 6);
    Serial.print("Latitude: ");
    Serial.println(latitude_data);
    Serial.print("Longitude: ");
    Serial.println(longitude_data);
    ThingSpeak.setField(1, latitude_data);
    ThingSpeak.setField(2, longitude_data);
    ThingSpeak.writeFields(ch_no, write_api);
    delay(20000);
  } else {
    Serial.println(F("Data error!!!"));
  }
}
```

Inside *displaywebpage(),* a **HTML code** is written which is sent to Client-side in string format using *client.print().* This HTML code contains a hyperlink which on clicking, takes you to the Google maps pointing the location of a tracked vehicle.

```
void displaywebpage() {
  WiFiClient client = server.available();
  if (client) {
    String page = "<html><head><style>";
    page += "body { font-family: Arial, sans-serif; background-
color: #f2f2f2; }";
    page += "h1 { color: #333; }";
```

```
    page += "img { max-width: 100%; height: auto; }";
    page += "</style></head><body>";
    page += "<center><h1>Real Time Vehicle Tracking using
IoT</h1></center>";
    page += "<center><img
src='https://via.placeholder.com/400x300' alt='Vehicle
Image'></center>";
    page += "<center><p><a style='color: red; font-size: 150%;'
href='http://maps.google.com/maps?&z=15&mrt=yp&t=k&q=";
    page += latitude_data;
    page += "+";
    page += longitude_data;
    page += "'>Click here For Live Location</a></p></center>";
    page += "</body></html>";

    client.print("HTTP/1.1 200 OK\r\n");
    client.print("Content-Type: text/html\r\n");
    client.print("Connection: close\r\n");
    client.print("\r\n");
    client.print(page);
    delay(100);
    client.stop();
  }
}
```

8.Data Validation and Inferences:

- **Security Concerns Validation:** The acknowledgment of security as a primary concern validates the project's relevance in addressing contemporary societal needs. This validates the importance of developing a GPS tracking device to enhance security measures for assets and individuals.
- **Utility of GPS Tracker Devices Validation:** Recognizing GPS tracker devices as valuable tools for real-time tracking reinforces the project's focus on leveraging this technology to monitor vehicle locations and ensure the safety of assets and individuals, including children.
- **Integration with ThingSpeak IoT Cloud Validation:** The decision to use ThingSpeak IoT cloud for storing location history validates the project's

approach to data management, ensuring that historical location data is securely stored and easily accessible for analysis and reference.

- **Previous Experience with ESP8266 and GPS Validation:** Mentioning prior experience with interfacing GPS with NodeMCU ESP8266 validates the project team's expertise and capability to execute the proposed solution effectively, indicating a solid foundation for implementation.
- **Enhanced Web Interface Inference:** The plan to include a link on the webpage directing users to Google Maps suggests a user-friendly approach to visualizing vehicle locations, enhancing accessibility and usability of the IoT Vehicle Tracking System. This emphasizes the project's focus on providing practical solutions for tracking and monitoring purposes.

9.Results:

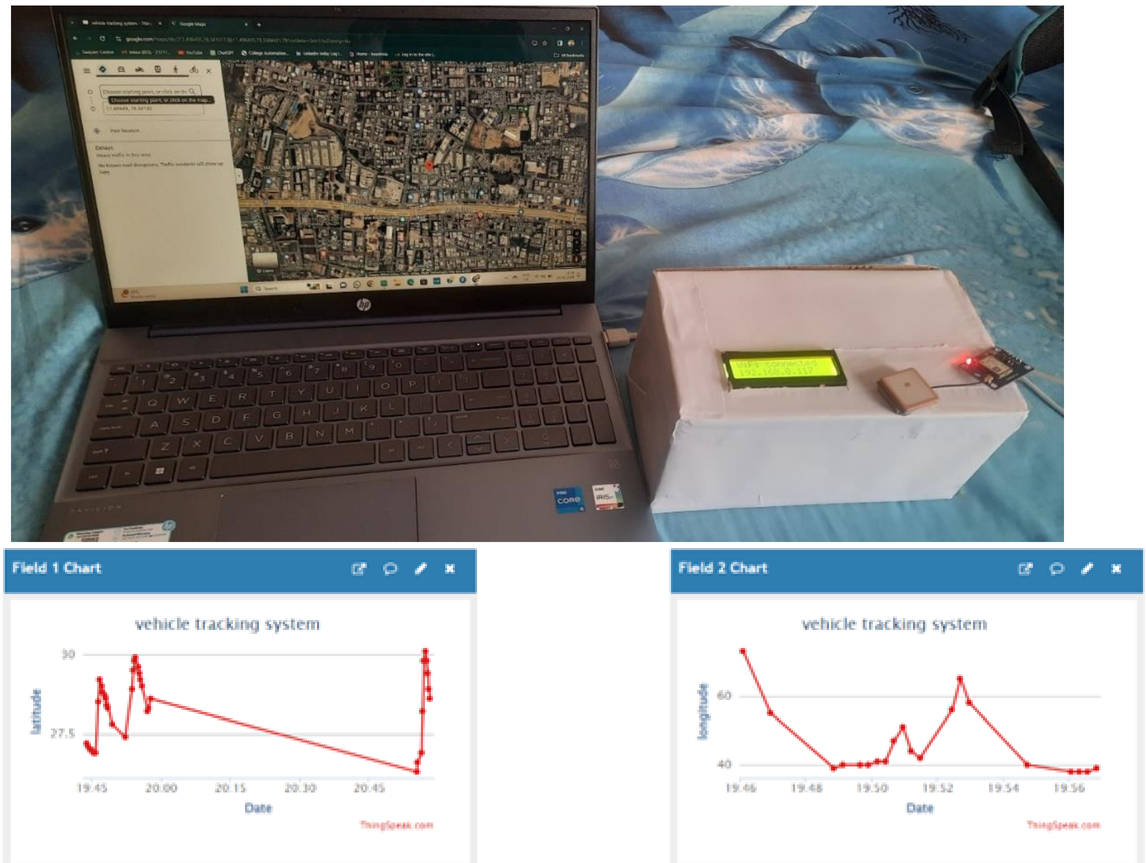

Fig3:Results

10. Conclusion:

In conclusion, the development of an IoT Vehicle Tracking System epitomizes a proactive approach to addressing contemporary security concerns regarding assets and personal safety. Leveraging the versatile capabilities of GPS tracker devices and the ESP8266 NodeMCU, this system empowers users with real-time monitoring of vehicle locations, ensuring prompt action in emergencies such as theft or accidents. Integration with the ThingSpeak IoT cloud facilitates seamless storage and retrieval of historical location data, providing valuable insights into past vehicle movements. Furthermore, the user-friendly web interface, coupled with the inclusion of a link to Google Maps for visualizing vehicle locations, enhances accessibility and usability for users. Through the utilization of innovative hardware components and software tools, including the NE06M GPS Receiver, 16*2 LCD, and Arduino IDE, this project underscores the effectiveness of modern technology in addressing security challenges. Ultimately, the IoT Vehicle Tracking System serves as a testament to the importance of embracing technological advancements to safeguard assets and loved ones in an increasingly interconnected world.

11. References:

1. A. Mounika and A. Chepuru, "Iot based vehicle tracking and monitoring system using gps and gsm", *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 2S111, pp. 2399-2403, 2019.

2. N. Pande, A. Somani, S. Prasad Samal and V. Kakkirala, "Enhanced web application and browsing performance through service-worker infusion framework", *2018 IEEE International Conference on Web Services (ICWS)*, pp. 195-202, 2018.

3. S. Lee, G. Tewolde and J. Kwon, "Design and implementation of vehicle tracking system using gps/gsm/gprs technology and smartphone application", *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 353-358, 2014.