DEPARTMENT OF COMPUTER &
SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Introduction To IOTs

Project Report

SUBMITTED TO:
Dr. Ali Hassan
LE Malaika Awais

SUBMITTED BY:

Faiqa Ummarah Ahmad	**Reg# 405052**
Fatima Tul Zohra	**Reg # 405985**
Shaheer Mukhtiar	**Reg# 432017**

DE- 44 Dept CE

Submission Date:1/10/2024

Project: GEO TRACER

Objectives:

The GEO TRACER endeavors to redefine location-based tracking by incorporating advanced IoT technologies, mainly focusing on achieving highly accurate geospatial information, the project prioritizes the development of a robust geolocation system, leveraging technologies like GPS for enhanced precision providing versatile tracking solution for diverse applications. Real-time monitoring, energy efficiency, scalability, user-friendly interfaces, and seamless integration with external systems constitute key project objectives.

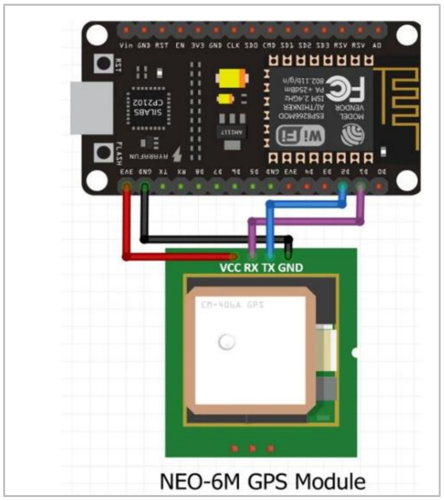
Project Brief Introduction:

- **Hardware Setup:**
  The NodeMCU (ESP8266) serves as the main microcontroller and connects to the GPS module. The GPS module receives signals from satellites and provides the current geographical coordinates (latitude and longitude). Wiring connections involve connecting the serial communication pins (TX and RX) of the GPS module to the corresponding pins on the NodeMCU.

- **Programming & User Interface:**
  The GPS tracker system employs the HTTP protocol and ThingSpeak to establish read and write API keys, enabling seamless integration with a JavaScript-based website. The NodeMCU, equipped with a GPS module, captures geographical coordinates (latitude and longitude) through serial communication. These coordinates are then transmitted to the ThingSpeak platform via HTTP requests, utilizing the designated API keys for secure data exchange.

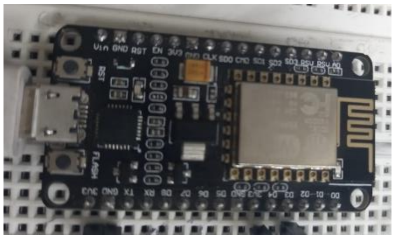
  ThingSpeak serves as a central hub for data storage, offering a convenient interface to visualize and analyze the received GPS information. The platform not only displays latitude and longitude but also facilitates the creation of a user-friendly dashboard. This dashboard, powered by JavaScript, integrates Google Maps for an intuitive representation of the tracked location. Users can access the website to view real-time location updates and historical data, providing a comprehensive and interactive experience.

Schematic Diagram:



Hardware Requirements:

Component	Use Case
NodeMCU (ESP8266) Module	Microcontroller for Edge Computing
NEO 6M GPS Sensor	Getting Longitudes and Latitudes for Location.
Lithium Batteries (3.7 V)	Powering up the system
Buck Converter	Managing the voltage for components



Software Tools:

- **Arduino IDE:**
  Utilized for facilitating the integration of the GPS module with Node MCU.
- **ThingSpeak Platform:**
  Provides an IoT analytics platform for storing, analyzing, and visualizing data received from the GPS tracker. Offers API keys for secure communication between the NodeMCU and the ThingSpeak server through HTTP requests.
- **JavaScript:**
  Employ JavaScript to develop a website which integrates Google Maps for a visually intuitive representation of the tracked location.

Program/Code:

Some instances of code are discussed below, whereas, complete code files are submitted on LMS:

ARDUINO IDE:

```
void loop()
{
  // This sketch displays information every time a new sentence is
correctly encoded.
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
    digitalWrite(gpsFoundLED, LOW);      // Turn off GPS found LED
    digitalWrite(gpsNotFoundLED, HIGH);   // Turn on GPS not found LED
    while(true);
  }
}
```

Explanation:

The loop () function continuously reads GPS data, processes it, and displays relevant information using the displayInfo() function. Additionally, it checks for potential issues with GPS detection and handles error conditions by indicating a wiring problem and entering an infinite loop.

JAVA Script:

```
<meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ThingSpeak Data Display</title>
    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
    <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAkr0CzDtEPoW2zQYeAvny9
tWDtnaxKW4M&callback=initMap"
        async defer></script>
```

Explanation:

This HTML code sets the character set, viewport configuration, and page title. It also includes script references for jQuery and the Google Maps API, indicating the use of asynchronous loading and a callback function (initMap) for Google Maps initialization. The combination of these elements suggests the web page may involve dynamic content display, potentially involving ThingSpeak data and Google Maps integration.

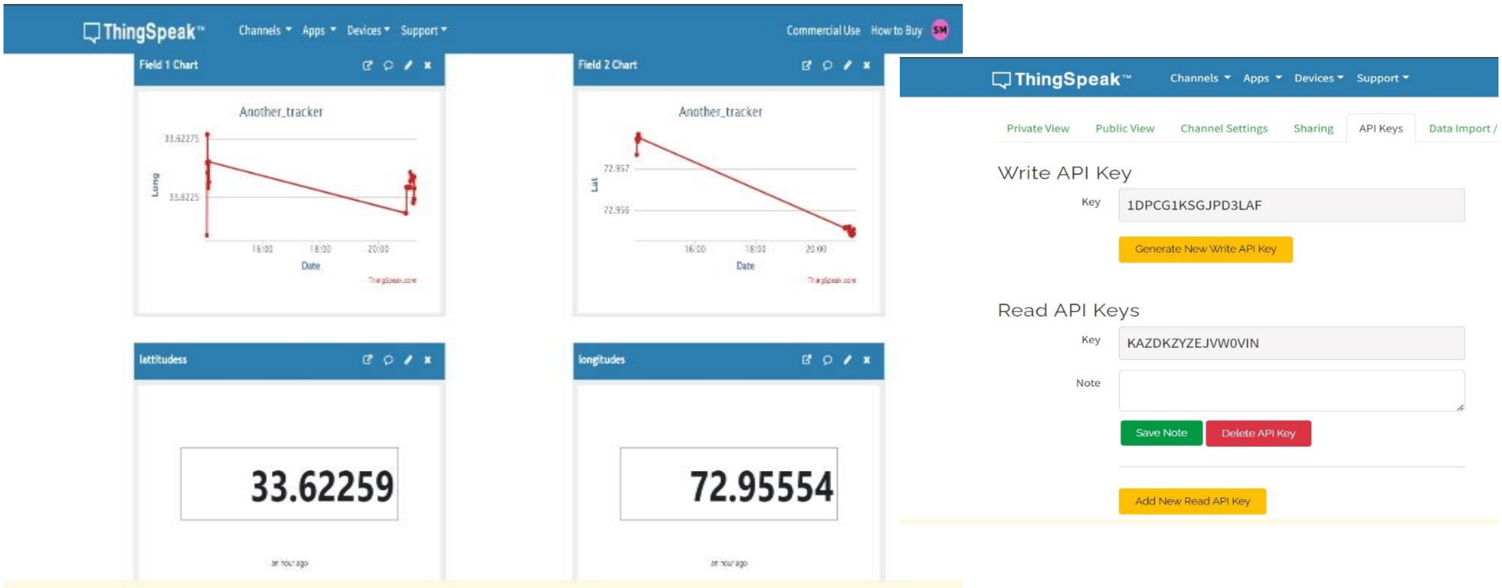
Function to update the map with new coordinates

```
function updateMap(newLat, newLng) {
    const mapCenter = { lat: parseFloat(newLat), lng: parseFloat(newLng) };

    // Get the map and marker
    const map = new google.maps.Map(document.getElementById('map'), {
        center: mapCenter,
        zoom: 18
    });

    const marker = new google.maps.Marker({
        position: mapCenter,
        map: map,
        title: 'Current Location'
    });
}
```

Explanation:

The updateMap function is a JavaScript function that takes latitude and longitude as parameters, updates a Google Map centered at those coordinates, and adds a marker at the specified location. This function is suitable for real-time updating of a map with changing geospatial data, such as the current location in a tracking application.

Snapshots of simulations:



33.62259

72.95554

Write API Key

Key 1DPCG1KSGJPD3LAF

Generate New Write API Key

Read API Keys

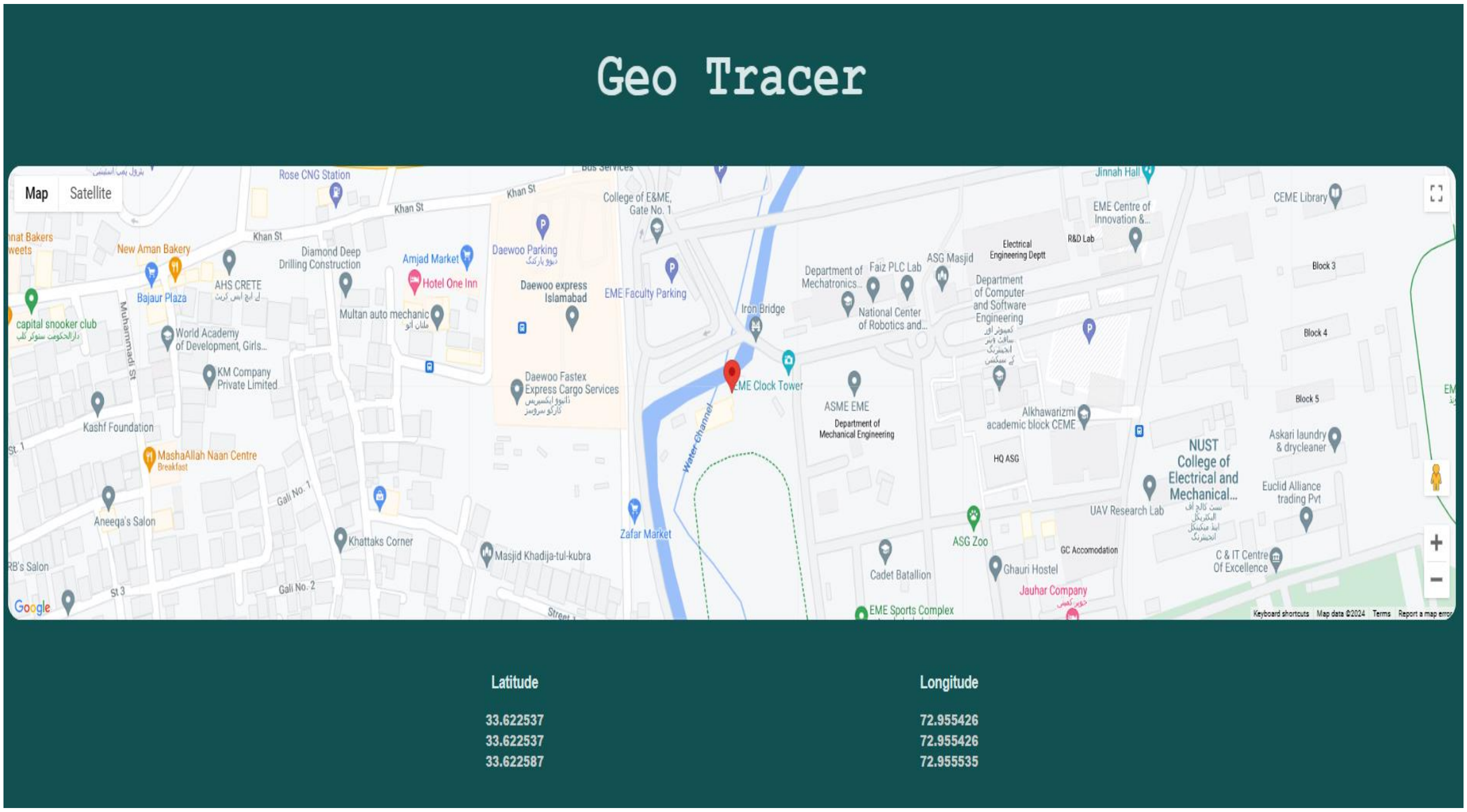
Key KAZDKZYZEJVW0VIN

Note

Save Note

Delete API Key

Add New Read API Key



Geo Tracer

Latitude	Longitude
33.622537	72.955426
33.622537	72.955426
33.622587	72.955535

Conclusion:

In conclusion, the project has achieved its objectives of creating an integrated GPS tracking system using IoT technologies. The combination of Arduino, ESP8266, and Google Maps API provides a comprehensive solution for real-time location monitoring. The TinyGPS++ library is effectively utilized to parse GPS data, while the ThingSpeak platform serves as a centralized hub for storing and visualizing the information. The JavaScript function ensures dynamic updating of a Google Map on the web interface, enhancing user experience. The project successfully meets key goals, including geospatial accuracy, real-time monitoring, and user-friendly interfaces.