SPRINT-2

TEAM ID: **PNT2022TMID29935**

GPS tracking

CODE:

```
#include <SoftwareSerial.h>

//Install the following Libraries
#include <TinyGPS++.h> //https://github.com/mikalhart/TinyGPSPlus
#include <ESP8266WiFi.h>

//GPS Module RX pin to NodeMCU D1
//GPS Module TX pin to NodeMCU D2
const int RXPin = 4, TXPin = 5;
SoftwareSerial neo6m(RXPin, TXPin);

TinyGPSPlus gps;

const char *ssid = "u";
const char *password = "987654321";

String GMAP_API_KEY = "AIzaSyCLKeJgxNB77U5R6E4Gg4vQ5HxD77pWsMQ";

WiFiServer server(80);

String html;
```

```
void setup()
{
  Serial.begin(115200);
  Serial.println();
  neo6m.begin(9600);

    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");    // this is the address to use for
viewing the map
    Serial.println(WiFi.localIP());
    server.begin();

}

void loop()
{
  smartdelay_gps(1000);

  if (gps.location.isValid())
  {
```

```
//Storing the Latitude. and Longitude
    String latitude = String(gps.location.lat(), 6);
    String longitude = String(gps.location.lng(), 6);

    //Send to Serial Monitor for Debugging
    //Serial.print("LAT:  ");
    //Serial.println(latitude);  // float to x decimal places
    //Serial.print("LONG: ");
    //Serial.println(longitude);

    // listen for incoming clients
    WiFiClient client = server.available();
    if(client) {
    Serial.println("new client");
    String currentLine = "";                // make a String to hold incoming
data from the client
    while (client.connected()) {
      if (client.available()) {             // if there's client data
        char c = client.read();          // read a byte
          if (c == '\n') {                        // check for newline
character,
          if (currentLine.length() == 0) {  // if line is blank it means its
the end of the client HTTP request



//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM
    html="<!DOCTYPE html>";
    html+="<html lang='en'>";
    html+="<head>";
    html+="<meta charset='UTF-8'>";
```

```
   html+="<meta name='viewport' content='width=device-width, initial-
scale=1.0'>";

    html+="<meta http-equiv='X-UA-Compatible' content='ie=edge'>";

    html+="<title>My Google Map</title>";

    html+="<style>#map{height:400px;width:100%;}</style>";

    html+="</head>";

    html+="<body>";

    html+="<h1>My Google Map</h1>";

    html+="<div id='map'></div>";

    html+="<script>";

//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM


    html+="var map;";

    html+="var marker;";


    //5000ms means 5000/1000 = 5 Seconds

    //20000ms means 20000/1000 = 20 Seconds

    html+="var INTERVAL = 5000;";



//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM

    html+="function initMap(){";

      //NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

      html+="var options = {";

          html+="zoom:16,";
```

```
    html+="center:{lat:"+latitude+",lng:"+longitude+"},";
          html+="mapTypeId: google.maps.MapTypeId.ROADMAP,";
      html+="};";
      //NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
      html+="map = new google.maps.Map(document.getElementById('map'),
options);";
    html+="}";

//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM


//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM
    html+="function getMarkers() {";
      //html+="console.log("+latitude+");";
      //html+="console.log("+longitude+");";

      html+="var newLatLng = new google.maps.LatLng("+latitude+",
"+longitude+");";
      //NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
      html+="marker = new google.maps.Marker({";
        html+="position: newLatLng,";
        html+="map: map";
      html+="});";
      //NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
    html+="}";

//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM


//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM
    html+="window.setInterval(getMarkers,INTERVAL);";
```

```
    html+="</script>";

    html+="<script async defer
src='https://maps.googleapis.com/maps/api/js?key="+GMAP_API_KEY+"&callback=ini
tMap'>";

    html+="</script>";

    html+="</body></html>";

//MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMM


client.print(html);


            // The HTTP response ends with another blank line:

            client.println();

            // break out of the while loop:

            break;

          } else {   currentLine = ""; }

        } else if (c != '\r') {  // if you got anything else but a carriage
return character,

          currentLine += c; // add it to the end of the currentLine

        }

         // here you can check for any keypresses if your web server page has
any

      }

    }

    // close the connection:

    client.stop();

    Serial.println("client disconnected");

    }

  }

}
```

```
static void smartdelay_gps(unsigned long ms)
{
  unsigned long start = millis();
  do
  {
    while (neo6m.available())
      gps.encode(neo6m.read());
  } while (millis() - start < ms);
}
```

OUTPUT: