Send GPS Location to Firebase Using GPS Neo 6M and SIM800L

Method · November 2022

DOI: 10.13140/RG.2.2.10230.47685

3 authors, including:

Oscar Onyango
Strathmore University
14 PUBLICATIONS **83** CITATIONS

SEE PROFILE

Solomon Kamau
Strathmore University
13 PUBLICATIONS **195** CITATIONS

SEE PROFILE

Send GPS Location to Firebase Using GPS Neo 6M and SIM800L

Oscar Onyango, Solomon Kamau, and Stephen Ng’etich

IoT Lab, @iLab Africa, Strathmore University

Recently, we were tasked with developing smart trackers for electric vehicles operating in Kenya. The goal was to design and implement a prototype that would capture and transmit GPS location data to a local database. We figured it would be convenient to use Firebase to store the data and temporarily host our web platform for visualization. If you have worked with Firebase, you should know that it is a straightforward platform, especially when sending data through Wi-Fi using esp modules or LoRa modules using RN2483 or RFM96 chips. Of course, my friend Solomon was quick to develop a quick sensor node to send the GPS data to The Things Console through LoRa. He then integrated the console so that the data would stream directly to Firebase. We called this our ‘control set-up’ as we needed similar sensor data sending data to Firebase, this time through GPRS using SIM800L. An hour of research turned into days and soon we realized that things were not easy as we thought. From the onset, we realized two major problems.

i. Power Stability

SIM800L runs with voltage between 3.4V to 4.4V. If you connect 5V, it damages the module and if you connect any voltage less than 3.4V, it will not operate. It specifically requires 3.7V, 2A power supply to fully operate.

ii. Connectivity: HTTPS vs HTTP

Firstly, SIM800 only supports HTTP which makes it challenging because connection to Firebase by default requires HTTPS. Secondly, we did not find any Firebase library that works with GSM module such as SIM800. Lastly, SIM800 supports only HTTP POST and HTTP GET Request but we needed HTTP PUT Request to update data in Firebase database.

The technicality of these challenges made it nearly impossible for us to send gps data to firebase using SIM800L. As a result, we opted for a quick, or rather an interesting approach to solve our problem. We started by sending the data to server. After that, we had to redirect the data from HTTP server to the firebase. The sections below summarize the general approach we took during the entire development.

Part 1: Hardware Connection and Programming

Requirements

i. Arduino nano and suitable IDE
ii. SIM800L
iii. GPS Neo 6M
iv. Breadboard
v. Jumper wires
vi. 4 1000uF, 16V capacitors

Circuit Set-up

The pin set-up between SIM800L, gps module, and Arduino nano were as shown in Tables 1 and Table 2. I will explain why SIM800L was connected to 5V in this case.

Table 1: SIM800L and Arduino nano

SIM800L	**Arduino Nano**
Vcc	5V
ground	ground
Rx	2

Tx	3

Table 1: GPS Neo 6M and Arduino nano

Arduino Nano	GPS Neo 6M
5V	Vcc
ground	ground
8	Tx
9	Rx

We noted that SIM800L, when connected in the same circuit with gps module to the Arduino nano, does not draw 5V. As a result, the microcontroller board could not power it to full functionality. The same applied to the gps module that also faced power instabilities and could not easily connect to the satellite. We had to add capacitors in the circuit to stabilize the power. For the SIM800L, we added two 1000uF, 16V capacitors in parallel while for the gps module, we added two 1000uF, 16V capacitors in series. Figure 1 gives the overall architecture of our circuit.

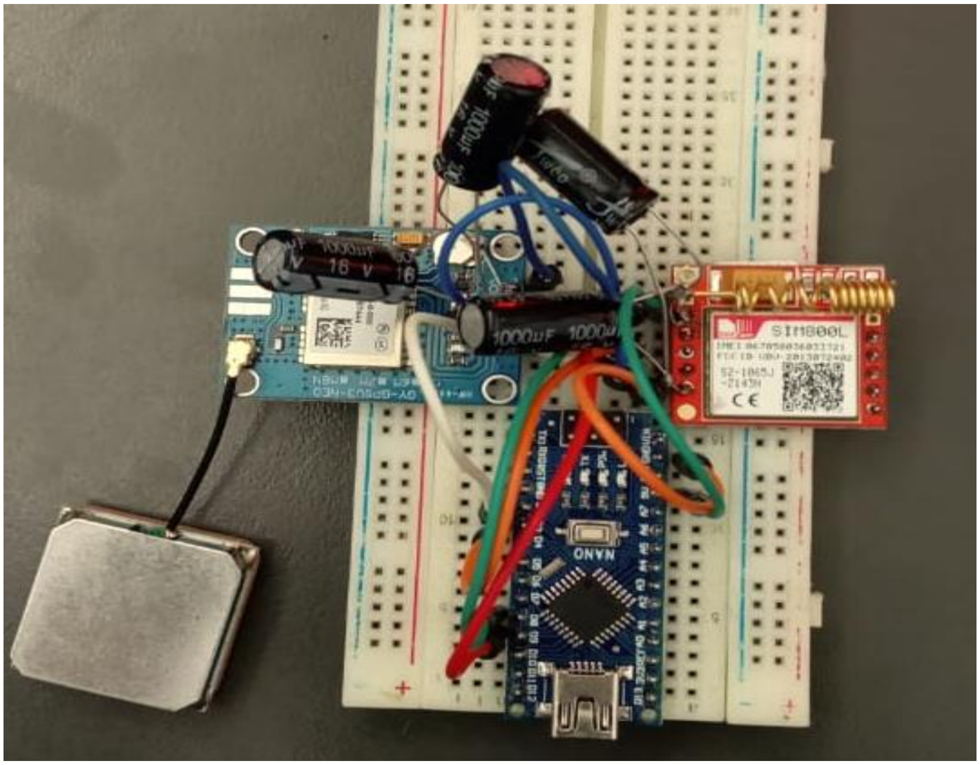


Figure 1: Integration of arduino, sim800l and gps module

The following libraries are needed to run the Arduino sketch. If using the Arduino IDE, you can easily download them by following the path: sketch-include library-manage libraries-search and install the particular library. We also working on another document explaining the specific AT commands used in this project. You can read about them in the next series. You can also download the same sketch from this LINK.

Arduino Sketch and how it works

First, we need to add the necessary libraries. The program requires three basic libraries to run.

i. TinyGPS++

ii. SoftwareSerial

iii. AltSoftSerial

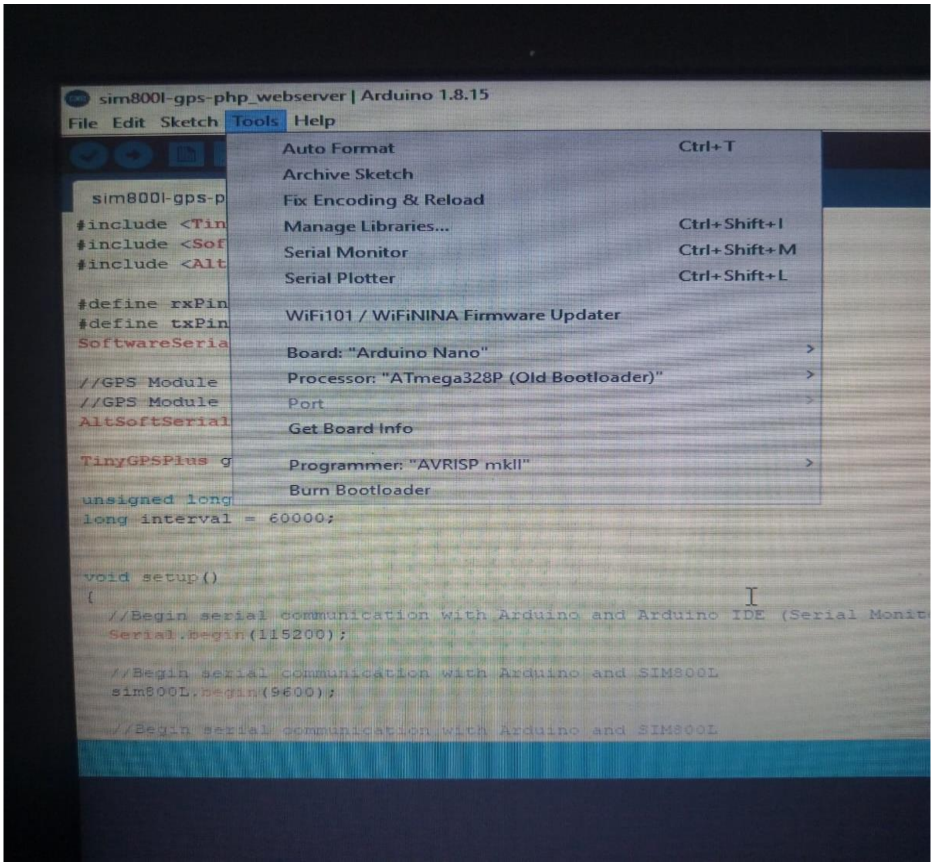
Here is the sketch for Arduino.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <AltSoftSerial.h>

#define rxPin 2
#define txPin 3
SoftwareSerial sim800L(rxPin,txPin);

//GPS Module RX pin to Arduino 9
//GPS Module TX pin to Arduino 8
AltSoftSerial neogps;

TinyGPSPlus gps;

unsigned long previousMillis = 0;
long interval = 60000;
```

```
void setup()
{
  //Begin serial communication with Arduino and Arduino IDE (Serial Monitor)
  Serial.begin(115200);

  //Begin serial communication with Arduino and SIM800L
  sim800L.begin(9600);

  //Begin serial communication with Arduino and SIM800L
  neogps.begin(9600);

  Serial.println("Initializing...");
  //delay(10000);

  //Once the handshake test is successful, it will back to OK
  sendATcommand("AT", "OK", 2000);
  sendATcommand("AT+CMGF=1", "OK", 2000);
  //sim800L.print("AT+CMGR=40\r");

}

void loop()
{
  while(sim800L.available()){
    Serial.println(sim800L.readString());
  }
  while(Serial.available())  {
    sim800L.println(Serial.readString());
  }

    unsigned long currentMillis = millis();
    if(currentMillis - previousMillis > interval) {
       previousMillis = currentMillis;
       sendGpsToServer();
    }
}

int sendGpsToServer()
{
    //Can take up to 60 seconds
    boolean newData = false;
    for (unsigned long start = millis(); millis() - start < 2000;){
      while (neogps.available()){
        if (gps.encode(neogps.read())){
          newData = true;
          break;
        }
      }
    }

    //If newData is true
    if(true){
      newData = false;

      String latitude, longitude;
      float altitude;
      unsigned long date, time, speed, satellites;

      latitude = String(gps.location.lat(), 6); // Latitude in degrees (double)
      longitude = String(gps.location.lng(), 6); // Longitude in degrees (double)
      altitude = gps.altitude.meters(); // Altitude in meters (double)
      date = gps.date.value(); // Raw date in DDMMYY format (u32)
      time = gps.time.value(); // Raw time in HHMMSSCC format (u32)
      speed = gps.speed.kmph();

      Serial.print("Latitude= ");
      Serial.print(latitude);
      Serial.print(" Longitude= ");
      Serial.println(longitude);

      //if (latitude == 0) {return 0;}

      String url, temp;
      url = "http://e-waka.000webhostapp.com/gpsdata.php?lat=“;  //you will need to change url
      url += latitude;
      url += "&lng=";
      url += longitude;
      url += "&source=";
      url += "gprs";
      url += "&device_id=";
      url += "ewaka_2";

      //url = "http://ahmadssd.000webhostapp.com/gpsdata.php?lat=222&lng=222";

      Serial.println(url);
      delay(300);

    sendATcommand("AT+CFUN=1", "OK", 2000);
    //AT+CGATT = 1 Connect modem is attached to GPRS to a network. AT+CGATT = 0, modem is not attached to GPRS to a network
    sendATcommand("AT+CGATT=1", "OK", 2000);
    //Connection type: GPRS - bearer profile 1
    sendATcommand("AT+SAPBR=3,1,\"Contype\",\"GPRS\"", "OK", 2000);
```

```
    //sets the APN settings for your network provider.
    sendATcommand("AT+SAPBR=3,1,\"APN\",\"internet\"", "OK", 2000);
    //enable the GPRS - enable bearer 1
    sendATcommand("AT+SAPBR=1,1", "OK", 2000);
    //Init HTTP service
    sendATcommand("AT+HTTPINIT", "OK", 2000);
    sendATcommand("AT+HTTPPARA=\"CID\",1", "OK", 1000);
    //Set the HTTP URL
sim800.print("AT+HTTPPARA="URL","http://ahmadssd.000webhostapp.com/gpsdata.php?lat=
222&lng=222"\r");
    sim800L.print("AT+HTTPPARA=\"URL\",\"");
    sim800L.print(url);
    sendATcommand("\"", "OK", 1000);
    //Set up the HTTP action
    sendATcommand("AT+HTTPACTION=0", "0,200", 1000);
    //Terminate the HTTP service
    sendATcommand("AT+HTTPTERM", "OK", 1000);
    //shuts down the GPRS connection. This returns "SHUT OK".
    sendATcommand("AT+CIPSHUT", "SHUT OK", 1000);

  }
  return 1;
}

int8_t sendATcommand(char* ATcommand, char* expected_answer, unsigned int timeout){

    uint8_t x=0,  answer=0;
    char response[100];
    unsigned long previous;

    //Initialice the string
    memset(response, '\0', 100);
    delay(100);

    //Clean the input buffer
    while( sim800L.available() > 0) sim800L.read();

    if (ATcommand[0] != '\0'){
      //Send the AT command
      sim800L.println(ATcommand);
    }

    x = 0;
    previous = millis();

    //this loop waits for the answer with time out
    do{
        //if there are data in the UART input buffer, reads it and checks for the asnwer
        if(sim800L.available() != 0){
            response[x] = sim800L.read();
            //Serial.print(response[x]);
            x++;
            // check if the desired answer (OK) is in the response of the module
            if(strstr(response, expected_answer) != NULL){
                answer = 1;
            }
        }
    }while((answer == 0) && ((millis() - previous) < timeout));

  Serial.println(response);
  return answer;
}
```

To program the board, ensure you select the appropriate board (Arduino nano) and port on the IDE in the tools tab. In case it fails to upload the code to the MCU, change the processor to the Old Bootloader. Figure 2 below shows the board set-up we used.



sim800l-gps-php_webserver | Arduino 1.8.15
File
Edit
Sketch
Tools
Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Port
Get Board Info
Burn Bootloader
sim800l-gps-p

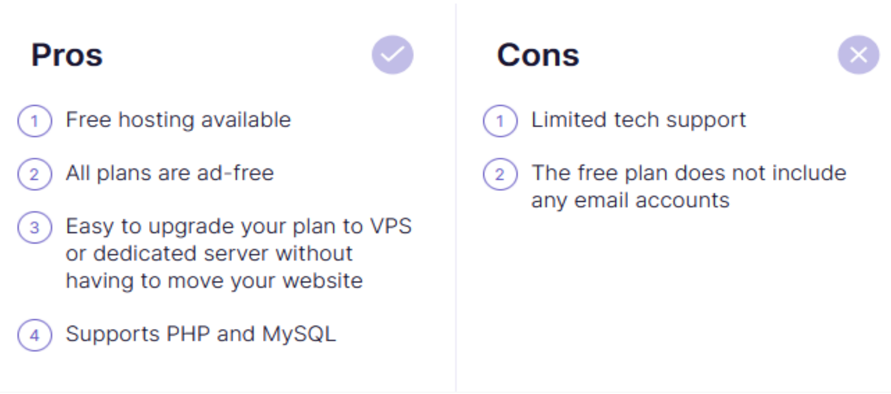
Board: "Arduino Nano"
Processor: "ATmega328P (Old Bootloader)"
Programmer: "AVRISP mkII"

```
#include <Tin
#include <Sof
#include <Alt

#define rxPin
#define txPin
SoftwareSeria

//GPS Module
//GPS Module
AltSoftSerial

TinyGPSPlus g

unsigned long
long interval = 60000;


void setup()
{
  //Begin serial communication with Arduino and Arduino IDE (Serial Monit
  Serial.begin(115200);

  //Begin serial communication with Arduino and SIM800L
  sim800L.begin(9600);

  //Begin serial communication with Arduino and SIM800L
```

Figure 2: Arduino nano board selection on IDE

Part 2: Server Set-up and Integration with Firebase

Begin by registering account with 000webhost. You can register through email or google. 000webhost is a free website hosting solution that provides an array of valuable features, including a website builder, WordPress support, and no ads. Ensure you confirm the registration by click on the link sent to your email.



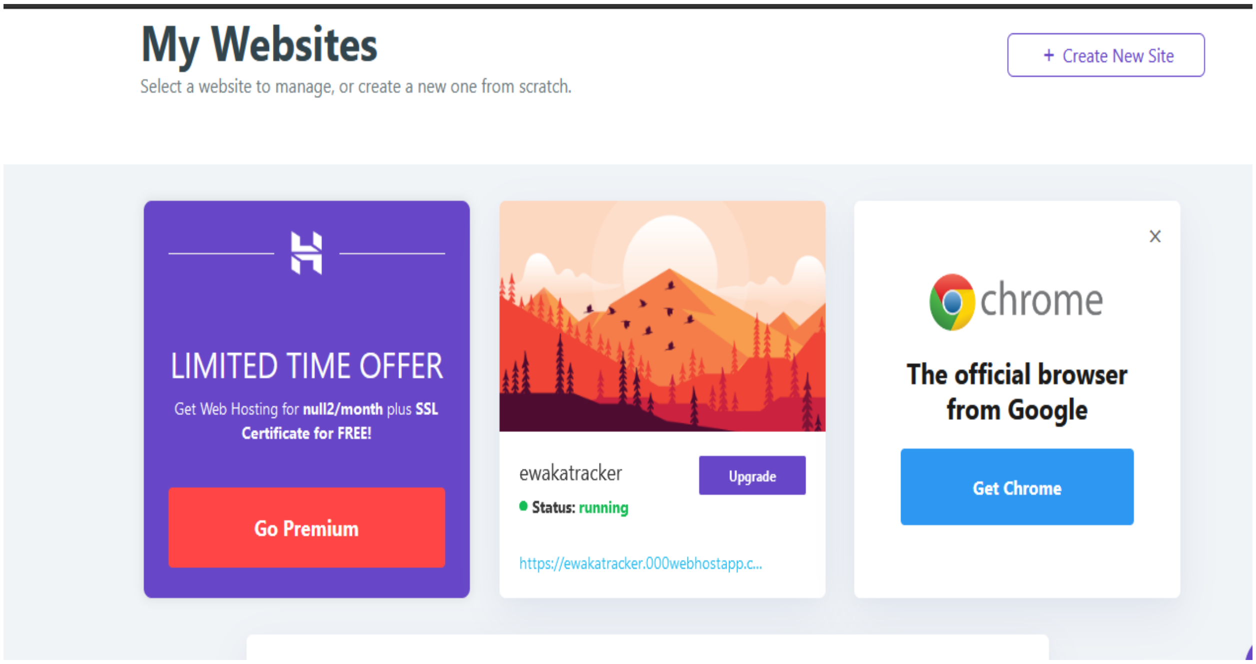
Pros

1. Free hosting available
2. All plans are ad-free
3. Easy to upgrade your plan to VPS or dedicated server without having to move your website
4. Supports PHP and MySQL

Cons

1. Limited tech support
2. The free plan does not include any email accounts

If the registration is successful, you should land in page that looks like this.



My Websites

Select a website to manage, or create a new one from scratch.

LIMITED TIME OFFER

Get Web Hosting for **null2/month** plus **SSL Certificate for FREE!**

ewakatracker

● **Status:** running

https://ewakatracker.000webhostapp.c...

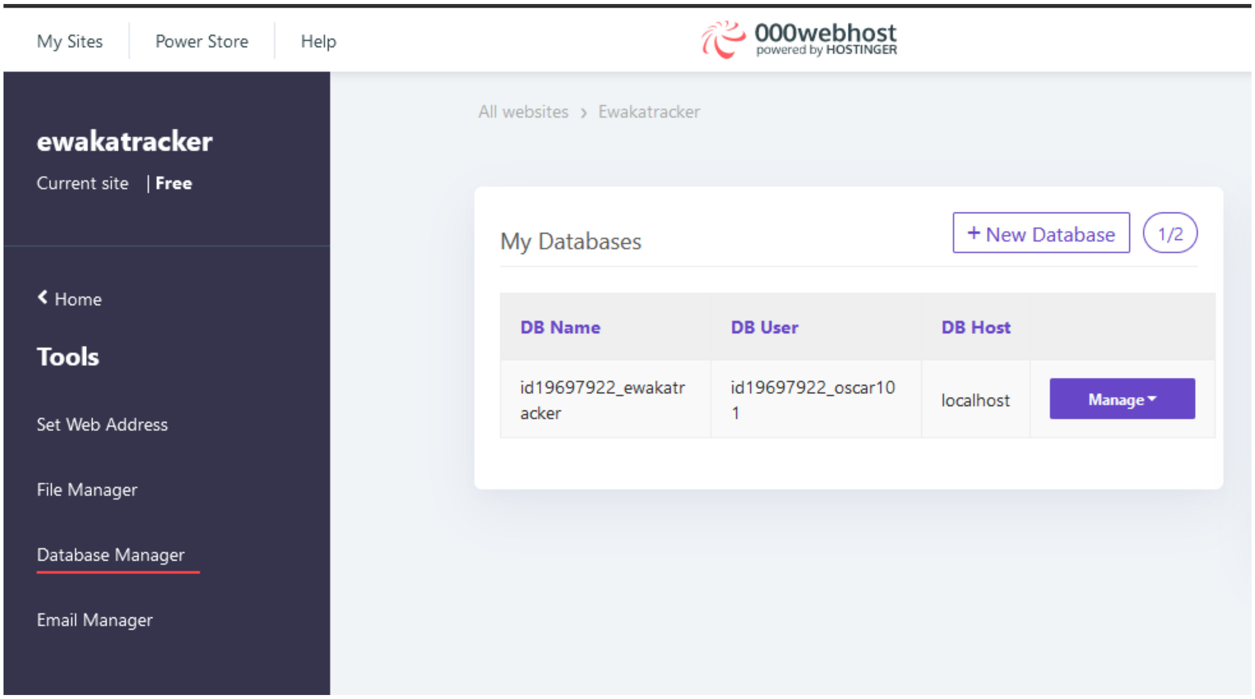
chrome

The official browser from Google

NB: Copy the link on the created website and add on Arduino sketch above. Also use http instead of https

```
String url, temp;
url = "add your link here/gpsdata.php?lat=";
url += latitude;
url += "&lng=";
url += longitude;
url += "&source=";
url += "gprs";
url += "&device_id=";
url += "ewaka_2";
```

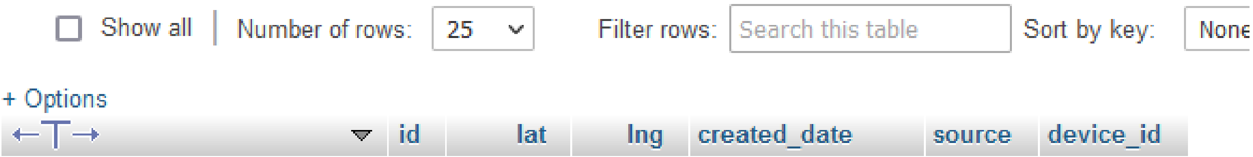
On the top right corner, click on the “Create New Site” button. You can ignore the Go Premium for now. Follow the steps until you have a registered website. The platform only allows one website for the trial version. Click on manage site to access the website you have created. On the left side bar, click tool then database manager. The page has the credentials we need to append to the code. **Note down the DB Name, DB use, and DB host**.



My Databases

DB Name	DB User	DB Host	
id19697922_ewakatracker	id19697922_oscar101	localhost	Manage

Click on manage and select PhpMyAdmin. It should take you to PhpMyAdmin page where you will need to create a table to receive the incoming data from the sensor node. Create a six-column table for id, lat, lng, created_date. Source, and device_id.



□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	lat	lng	created_date	source	device_id

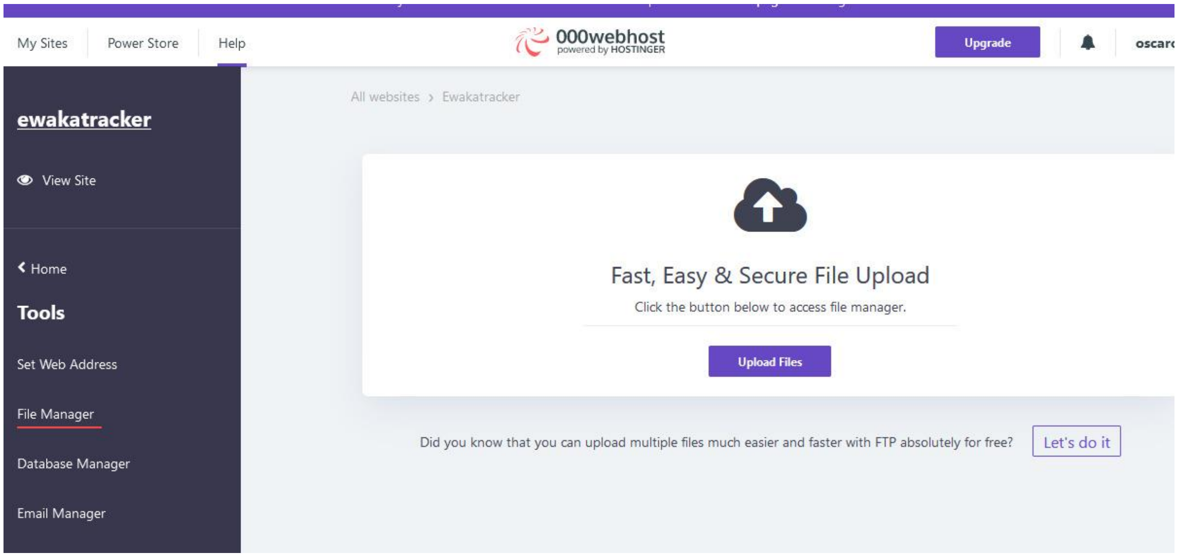
Now go back to the 000webhost platform. We need to add the PhP script and configuration file to allow it receive data from the sensor data. Still under Tool, select File Manager. Download the sketches from this LINK and make the following adjustments before uploading them to the server.

i. Config File- Change the DB_HOST, DB_USERNAME, DB_PASSWORD, AND DB_NAME, based on the parameters you saved while creating the website. You can access the same my clicking on Manage Website.

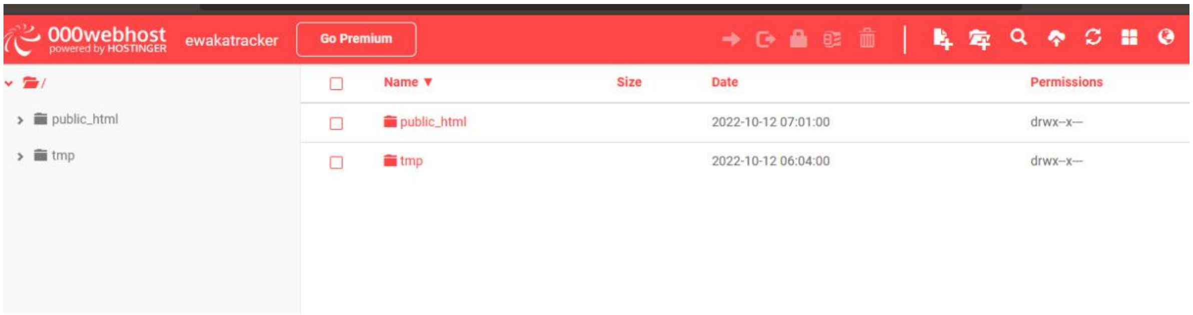
ii. We prepare the PhP script needed to link the server to the firebase. You need to access your firebase and change the url. This is available by going to the specific project created for the tracker, selecting Realtime Database. You can copy the link directly under the data tab.

Once these set-ups are done, upload the three files on the public HTML folder. You need to double click to access this folder before uploading the files. The procedure is as follows.

1. Select tools> File Manage>Upload Files

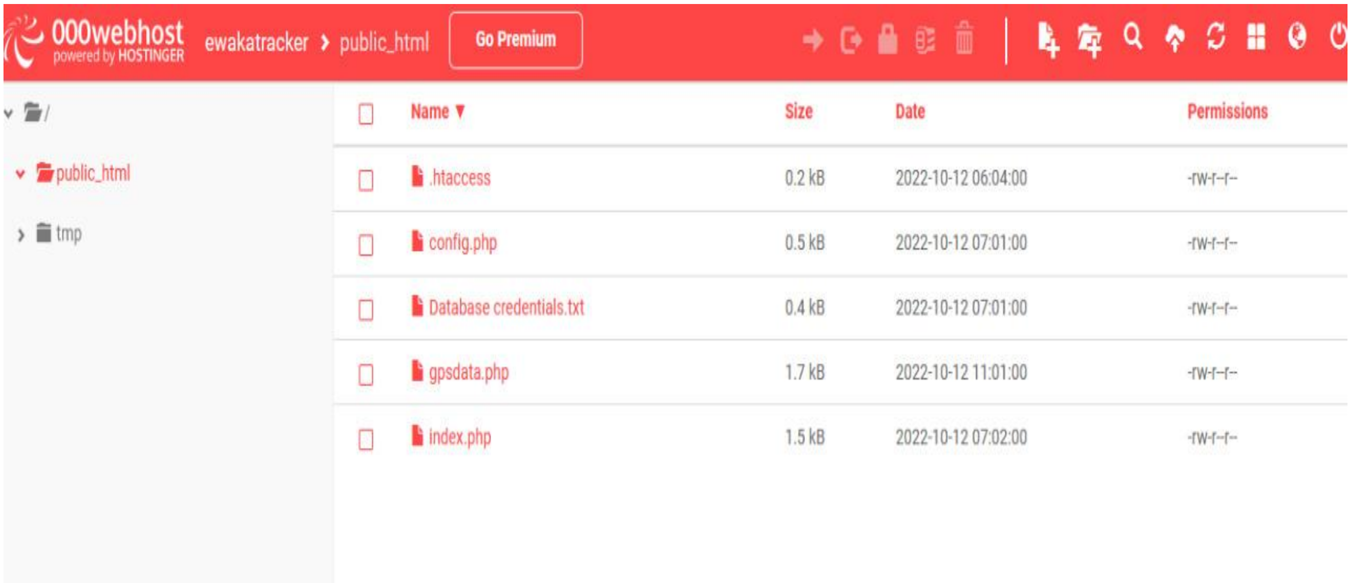

My Sites
Power Store
Help
000webhost powered by HOSTINGER
Upgrade
oscar
ewakatracker
View Site
Home
Tools
Set Web Address
File Manager
Database Manager
Email Manager
All websites > Ewakatracker
Fast, Easy & Secure File Upload
Click the button below to access file manager.
Upload Files
Did you know that you can upload multiple files much easier and faster with FTP absolutely for free?
Let's do it

2. Double click on public HTML folder



□	Name ▼	Size	Date	Permissions
□	public_html		2022-10-12 07:01:00	drwx--x---
□	tmp		2022-10-12 06:04:00	drwx--x---

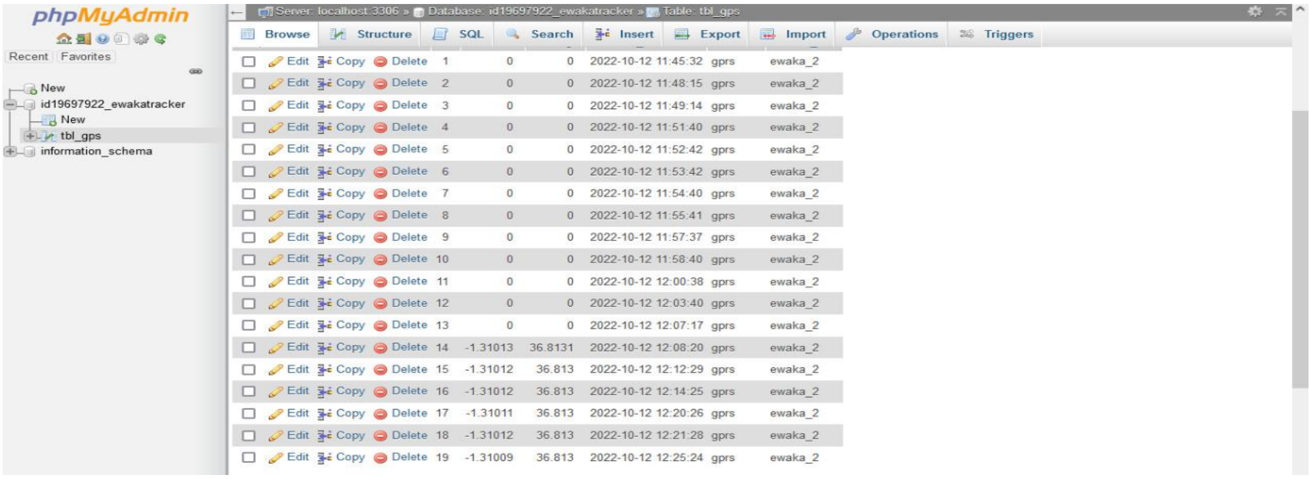
Once inside the folder, on the top right corner, select upload files and select the files from the location you saved them in your computer. The uploaded files will appear as follows.



□	Name ▾	Size	Date	Permissions
□	.htaccess	0.2 kB	2022-10-12 06:04:00	-rw-r--r--
□	config.php	0.5 kB	2022-10-12 07:01:00	-rw-r--r--
□	Database credentials.txt	0.4 kB	2022-10-12 07:01:00	-rw-r--r--
□	gpsdata.php	1.7 kB	2022-10-12 11:01:00	-rw-r--r--
□	index.php	1.5 kB	2022-10-12 07:02:00	-rw-r--r--

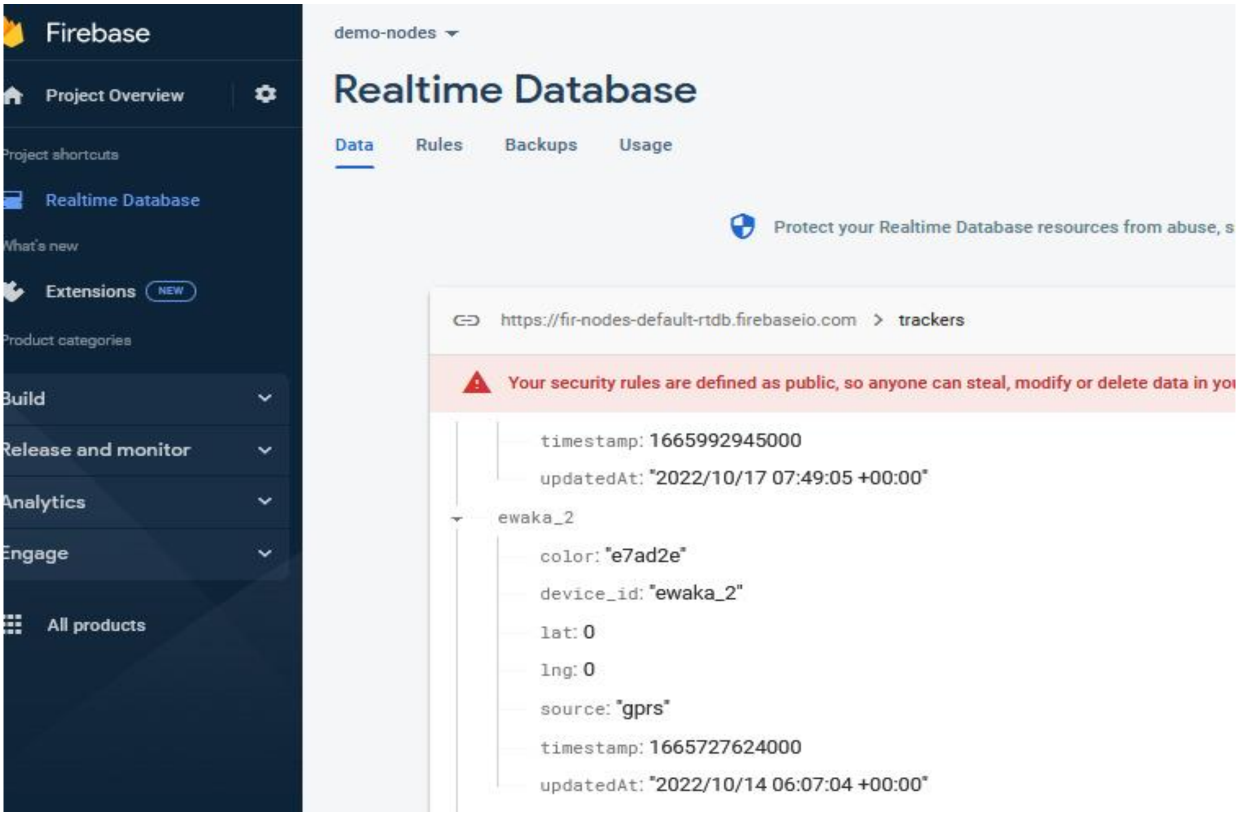
Once this step is done, the platform should be ready to receive data from the sensor node. You can visualize the same data by accessing the PhpMyAdmin page. The same data will be transmitted to the firebase. It is quite a complex procedure the results should be as shown in the figure below. Both pages give the latitudes and longitudes received from the tracker at different times.

PhpMyAdmin Snippet



□	Edit	Copy	Delete	1	0	0	2022-10-12 11:45:32	gprs	ewaka_2
□	Edit	Copy	Delete	2	0	0	2022-10-12 11:48:15	gprs	ewaka_2
□	Edit	Copy	Delete	3	0	0	2022-10-12 11:49:14	gprs	ewaka_2
□	Edit	Copy	Delete	4	0	0	2022-10-12 11:51:40	gprs	ewaka_2
□	Edit	Copy	Delete	5	0	0	2022-10-12 11:52:42	gprs	ewaka_2
□	Edit	Copy	Delete	6	0	0	2022-10-12 11:53:42	gprs	ewaka_2
□	Edit	Copy	Delete	7	0	0	2022-10-12 11:54:40	gprs	ewaka_2
□	Edit	Copy	Delete	8	0	0	2022-10-12 11:55:41	gprs	ewaka_2
□	Edit	Copy	Delete	9	0	0	2022-10-12 11:57:37	gprs	ewaka_2
□	Edit	Copy	Delete	10	0	0	2022-10-12 11:58:40	gprs	ewaka_2
□	Edit	Copy	Delete	11	0	0	2022-10-12 12:00:38	gprs	ewaka_2
□	Edit	Copy	Delete	12	0	0	2022-10-12 12:03:40	gprs	ewaka_2
□	Edit	Copy	Delete	13	0	0	2022-10-12 12:07:17	gprs	ewaka_2
□	Edit	Copy	Delete	14	-1.31013	36.8131	2022-10-12 12:08:20	gprs	ewaka_2
□	Edit	Copy	Delete	15	-1.31012	36.813	2022-10-12 12:12:29	gprs	ewaka_2
□	Edit	Copy	Delete	16	-1.31012	36.813	2022-10-12 12:14:25	gprs	ewaka_2
□	Edit	Copy	Delete	17	-1.31011	36.813	2022-10-12 12:20:26	gprs	ewaka_2
□	Edit	Copy	Delete	18	-1.31012	36.813	2022-10-12 12:21:28	gprs	ewaka_2
□	Edit	Copy	Delete	19	-1.31009	36.813	2022-10-12 12:25:24	gprs	ewaka_2

Firebase Snippet



Protect your Realtime Database resources from abuse, s

https://fir-nodes-default-rtdb.firebaseio.com > trackers

Your security rules are defined as public, so anyone can steal, modify or delete data in you

```
      timestamp: 1665992945000
      updatedAt: "2022/10/17 07:49:05 +00:00"
  ewaka_2
      color: "e7ad2e"
      device_id: "ewaka_2"
      lat: 0
      lng: 0
      source: "gprs"
      timestamp: 1665727624000
      updatedAt: "2022/10/14 06:07:04 +00:00"
```