**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY**

**DEPARTMENT OF TELECOMMUNICATIONS ENGINEERING**

**--------------------------**



**DIGITAL COMMUNICATIONS**

**GPS DETECTER PROJECT**

**Students : Vương Minh Phú-1551112**

**Lê Hoàng Giang-1551032**

*HCM City 2019*

**Contents**

1. Abstract 3
2. Install frameworks 3

1. Visual Studio Code Insider 4

2. Create Google map API account 4

3. Create Firebase account 4

4. Internet information services manager 4

5. Android Studio 4

1. Local website 3
2. GPS detector android app 4
3. Results 4
4. Applications 16
5. Reference 20

**I/ Abstract:**

In this report we describe how to build a system to locate a phone run on android operating system. Our system include an android app and a website include Google API. First, by using the basic library in and the GPS chip in android device, we can take longitude and latitude from where it stand, then put its location into JSON Object and send it to our local web. In our server, Google map API requires longitude and latitude to return device’s location.

**II/ Install Framework:**

**1/ Visual studio code insider:**

**1.1/ What is visual studio code insider?**

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). It is a free framework provided by Microsoft.

1.2/ How to download and install Visual studio code insiders?

First, go to this link:

<https://code.visualstudio.com/insiders/>



Choose the biggest purple rectangle which has text “Download for Windows” for download the “.exe” file.

After finish, run the download file and install normally.

**2/Create Google map API account:**

**2.1/ What is Google map API? Why Google map API? What for?**

\_ Google maps is an free application provided by Google company through the URL “map.google.com” or mobile app. People can easily take their location and find the best road in every hours.

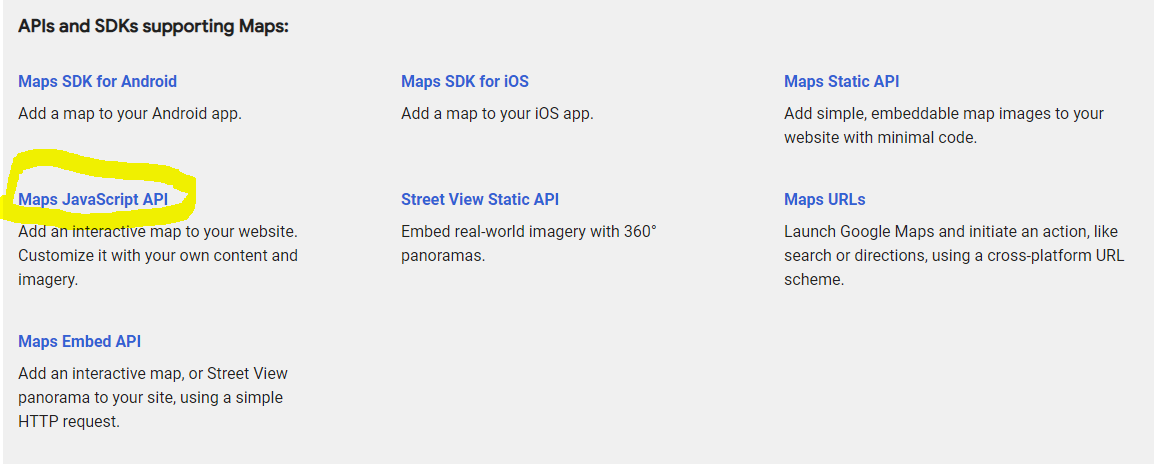
\_ Google maps API is an application programming interface with Google map and approach the giant data of Google in order to create a third app for economic or research.

\_ Google maps have data of more than 200 countries and cover 99% earth area. Over 25 millions location’s updates from collaborators and Google Local Guide teams around the world everyday. About one billion users each month.

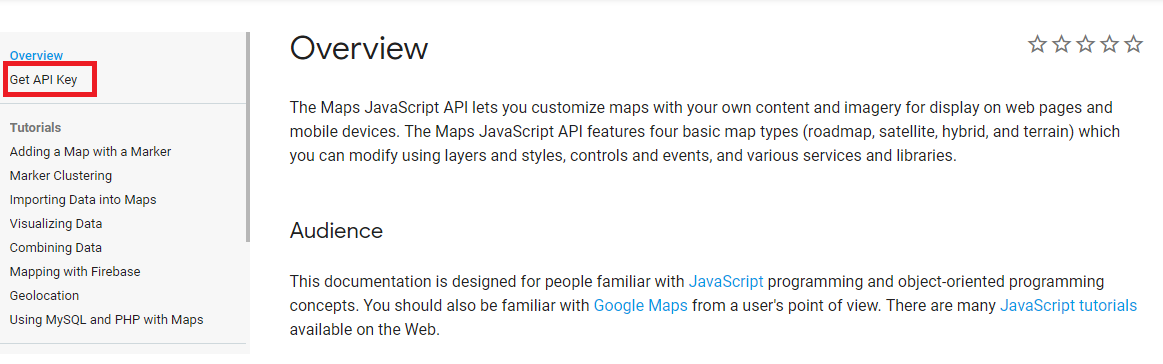
\_ Google API support for web, app and webservice developer.

**2.2/ How to create Google Maps API Keys:**

\_ First, go to the URL: <https://developers.google.com/maps/documentation/>



\_ Choose Maps JavaScript API



\_ Choose Get API Key.

\_ Then just follow the guide in order get API Key.

**3/ Create Firebase Accounts:**

**3.1/ What is firebase? Why we use Firebase?**

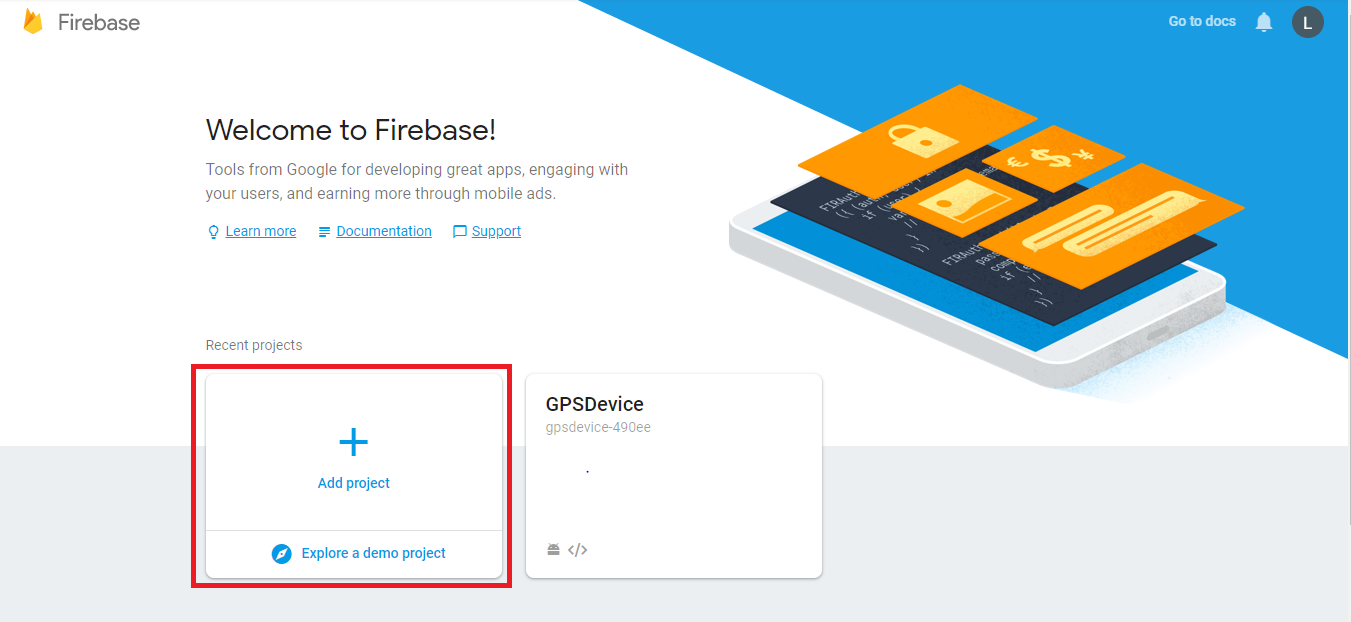
\_ Firebase is a Backend-as-a-Service — BaaS — that started as a YC11 startup and grew up into a next-generation app-development platform on Google Cloud Platform.

\_ Firebase frees developers to focus crafting fantastic user experiences. You don’t need to manage servers. You don’t need to write APIs. Firebase is your server, your API and your datastore, all written so generically that you can modify it to suit most needs. Yeah, you’ll occasionally need to use other bits of the Google Cloud for your advanced applications. Firebase can’t be everything to everybody. But it gets pretty close.

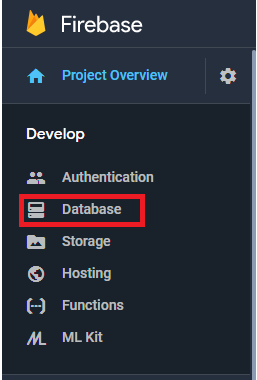
\_ It’s realtime database.

**3.2/ How to get firebase key:**

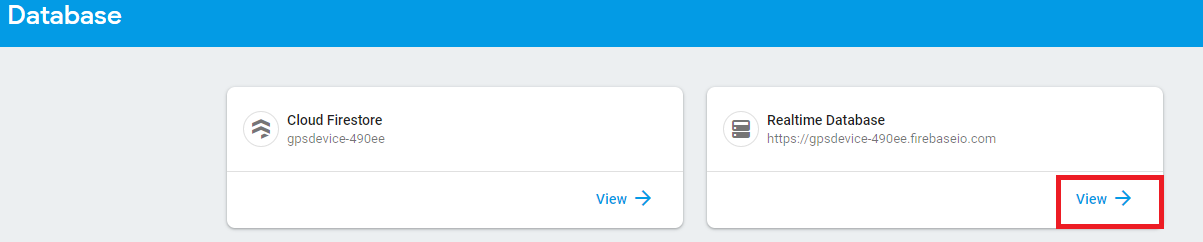
\_ First, log in to Firebase home website by link: <https://console.firebase.google.com/u/0/>



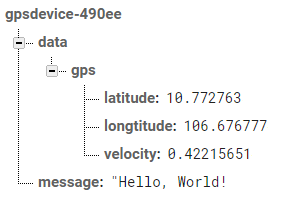
\_ Then choose Add Project



\_ Choose Database



\_ Choose Realtime Database



\_ Create your variable.

\_ Then we can push or pull data from Firebase in realtime.

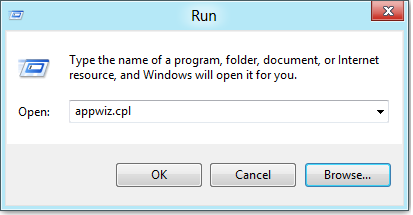
**4/ Internet information service manager (IIS)**

**4.1/ What is it?**

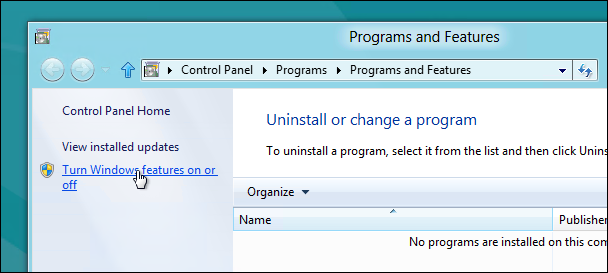
\_ IIS (which stands for Internet Information Services or Internet Information Server) also known as Windows web server is available on most versions of Microsoft Windows operating systems and takes second place in overall usage behind Apache HTTP Server on the internet. It will host websites, web applications and services needed by users or developers. Many versions have shipped as far back as IIS 1 on Windows 3 and with nearly every new Windows OS a new IIS version follows.

**4.2/ How to install and use IIS?**

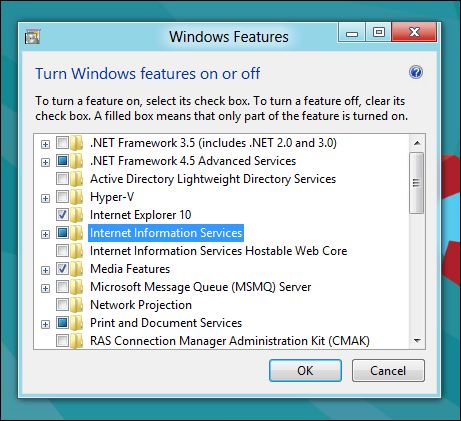
\_ Keeping with Microsoft modular design of, uhm, everything these days, IIS in Windows is still an optional “Windows Feature”. To install it, press the Windows + R key combination to bring up a run box, then type appwiz.cpl and press enter.



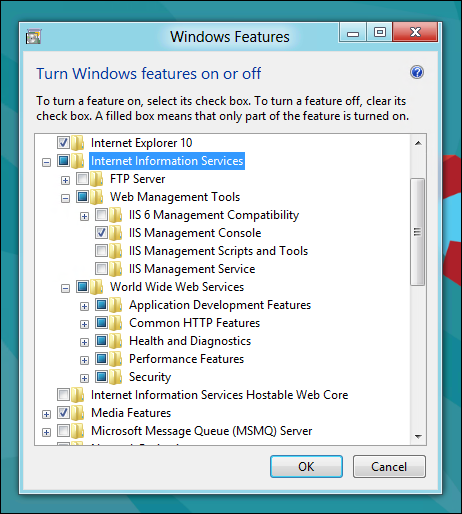
\_ This will open the Program and Features part of Control Panel, on the left hand side click on the “Turn Windows features on or off” link.



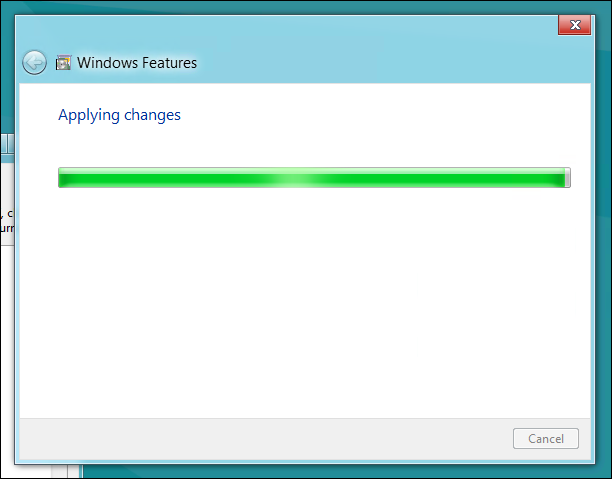
\_ Now click on the Internet Information Services check box.



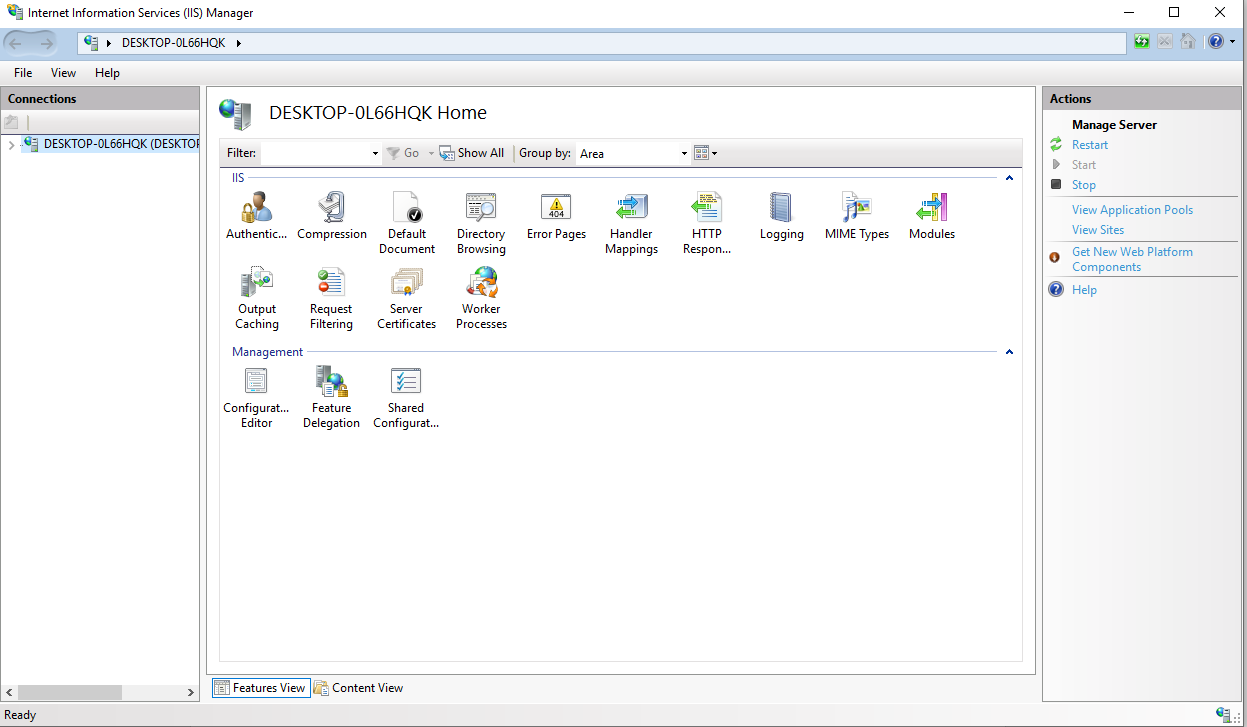
\_ If you’re a developer you are going to want to expand it and explore the sub-components as well. By default it installs all the stuff needed to host a website, and you are probably going to need some of the more developer centric components as well.



\_ After clicking OK, this dialog will appear on your screen for a while.



\_ Then find IIS in start menu, you will see



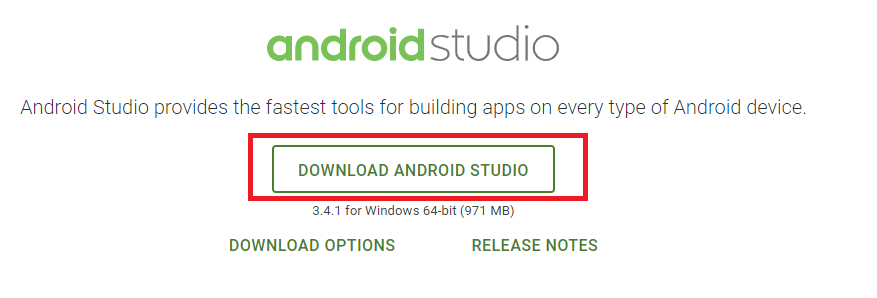
**5/ Android Studio**

**5.1/ What is Android Studio**

\_ Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

**5.2/ Install Android Studio**

\_ Go to Android Studio home page: <https://developer.android.com/studio>



Choose Download android studio then follow the guild to install its.

**III/ Local website**

III/ Build a local website

In this project we will build a website in order to show the result of our GPS Tracker. Our website will contain the estimate velocity, traveled path, or the current position on the map. This web will show the results in real time and update simultaneously by the Android app.

Firstly, we need to create a website using HTML, CSS and Java-script. The HTML part will the base of our web, CSS will be the front and Java-script will handle the what will be display on the real time.

Begin with CSS, we first declare some variables for the theme and style.

|  |
| --- |
| <head>  <title>  GPS Tracker  </title>  <style>  body{  background-color:darkslategrey  }  button{  font-family:sans-serif;  display:block;  position: absolute;  margin-left: 44%;  margin-right: 40%;  color: blue;  padding: 15px 32px;  text-align: center;  }  div{  margin-left: auto;  margin-right: auto;  margin-top: 6%;  margin-bottom: auto;    }  add {  color: red;  position: absolute;  top: 80px;  left: 20px;    }  </style> |

Then we have to include the js for Google API and Firebase, it will follow their syntax as follow, where key is added during the registration from google API

|  |
| --- |
| <script src="http://maps.googleapis.com/maps/api/js?key=#AIzaSyCGR0AEjSkP-22MPQIiG33LfUsnUhUDNQ0&sensor=false"> </script> <script src="https://www.gstatic.com/firebasejs/5.5.3/firebase.js"></script> |

After that we continue to the body of the website which content our result

|  |
| --- |
| <body>  <h1 style="color:white; text-align: center">  MY LOCATION  </h1>   <button onclick="getInfo()" type="button", text-align: center>  Get Location  </button>  <add>  <h2 id="demo"></h2>  </add>      <div id="googleMap" style="width:1000px;height:580px;",></div>  </body> |

As you can see in this codeblock we use style <add>, <div> and <button> that declared on the previous cell. There are two very important functions which are getInfor() and showMap() will be explained below.

Before getting into the functions we need to initialize some syntax for the Firebase first and these codes will be in the script cell.

|  |
| --- |
| *// create firebase config*  const firebaseConfig = {   apiKey: "AIzaSyArEPcIs8WAFnV91hYjNGdgRGKzkVvWFPc",  authDomain: "gpsdevice-490ee.firebaseapp.com",  databaseURL: "https://gpsdevice-490ee.firebaseio.com",  projectId: "gpsdevice-490ee",  storageBucket: "gpsdevice-490ee.appspot.com",  messagingSenderId: "918934474657",  appId: "1:918934474657:web:328d3cce22eda7d9" };  *// get global variable ref ,data from firebase*  var ref;  *// the list for append the way device move*  var direction = [];  *// Syntax firebase*  window.onload=function(){  *// check whether it is the correct firebase*  console.log("Initilize database")  firebase.initializeApp(firebaseConfig);  ref = firebase.database().ref();   console.log("Firebase", ref)   }; |

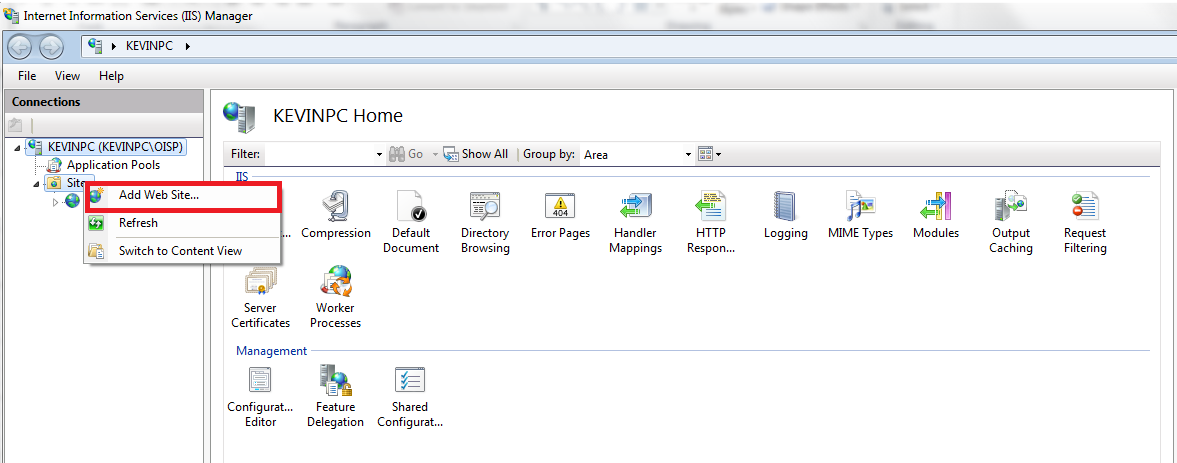
Finally is the code for getInfor() and showMap() functions

|  |
| --- |
| function getInfo() {  *// make sure real time update*  ref.on("value", function (snapshot) {  var gps = snapshot.val().data.gps;  var data = {  "lat": gps.latitude,  "long": gps.longitude,  "velocity": gps.velocity  }  *// call the below function*  showMap(data)  *// draw it down*    }, function (error) {  console.log("Error: " + error.code);  });    }  function showMap(data)  {  var mycenter = {lat: data.lat, lng: data.long};  var mapProp =  {  center:new google.maps.LatLng(data.lat,data.long),  zoom:20,  mapTypeId:google.maps.MapTypeId.ROADMAP  };  var map=new google.maps.Map(document.getElementById("googleMap"), mapProp);  var marker=new google.maps.Marker({  position:mycenter,  map:map  });  direction.push({lat: data.lat, lng: data.long})  *// print on consle for debug*  console.log("velocity: "+ data.velocity)  console.log(direction)  document.getElementById("demo").innerHTML = 'Your velocity: ' + data.velocity + 'm/s';  *// get a new var into Google API*  var devicePath = new google.maps.Polyline({  path: direction,  geodesic: true,  strokeColor: '#FF0000',  strokeOpacity: 1.0,  strokeWeight: 2  });  *// draw*  devicePath.setMap(map);  }; |

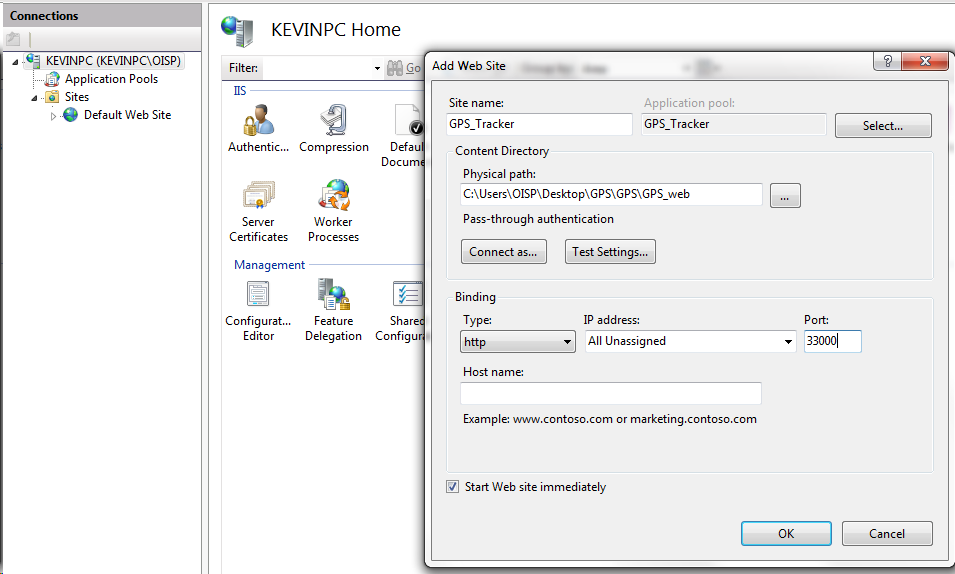
In getInfor() we go to the Firebase cloud and then take the variables from the database including latitude, longitude and velocity and call the showMap() function to show on the website. The showMap() will take data argument and then mycenter, myprop and marker take it. mycenter will be the center of the map, myprop contains the the google api and marker shown the current our position. And finally the devicePath display the path that we traveled on the map.

One more important step to make our web online is IIS by Microsoft. We follow the ínstructions below:

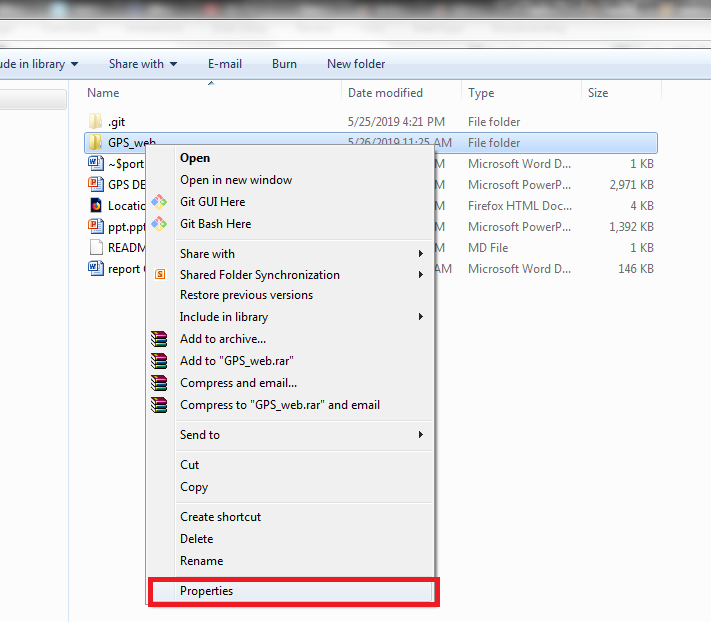
Step 1 Open Internet Information Service and add new rule



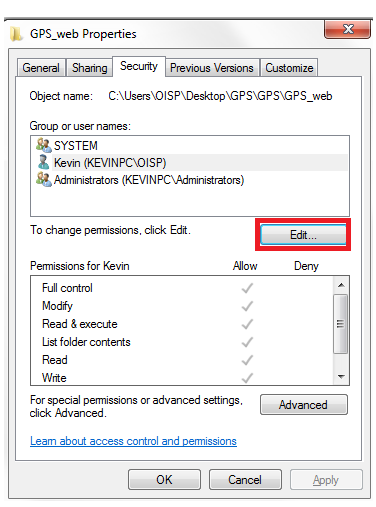
Step 2 specify the site name and add the physical path of our website and port which is 33000 for our website



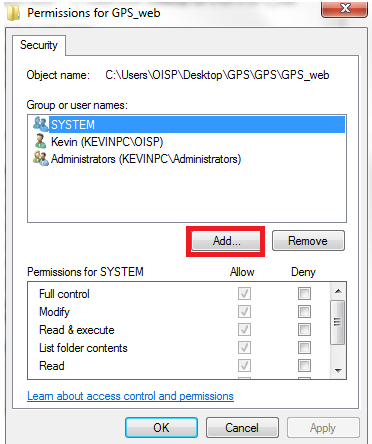
Step 3 we go to the folder containing our website and go to its properties.

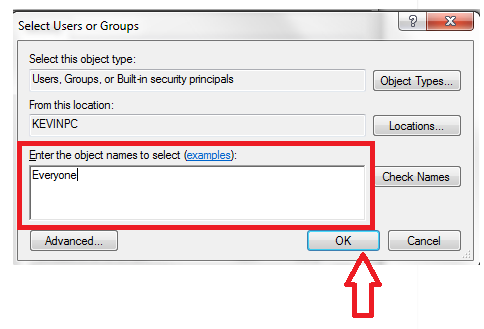


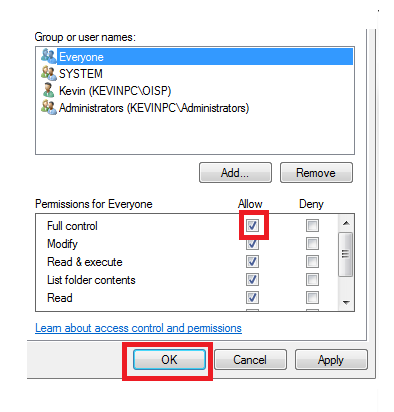
Step 4 Go to security tab and click edit



Step 5 Add the new user name which is everyone and allow it full control

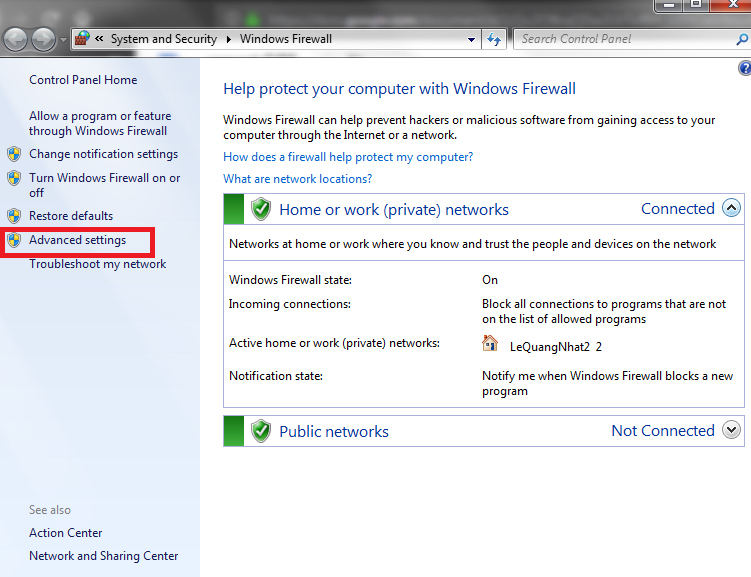




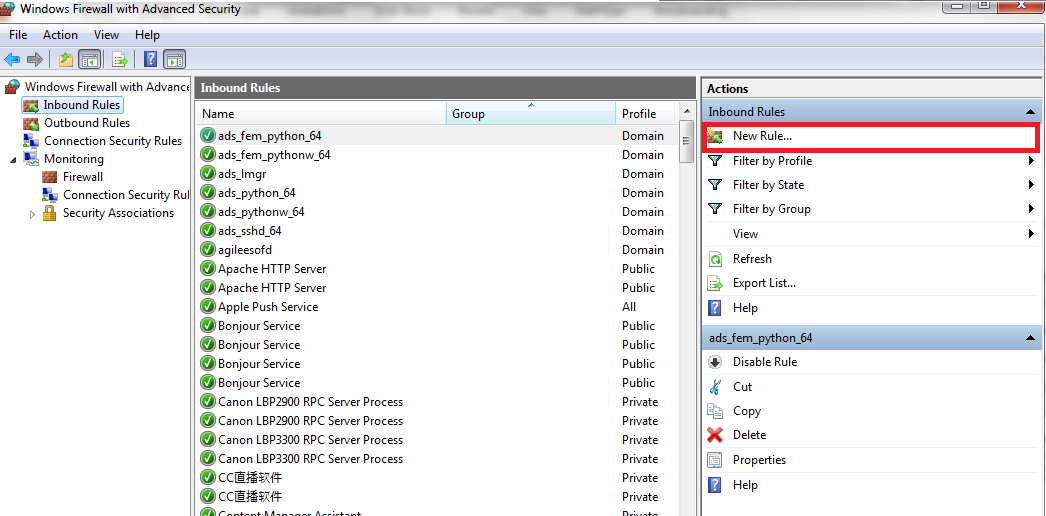


Step 6 We need to specify our firewall to allow our port to be online

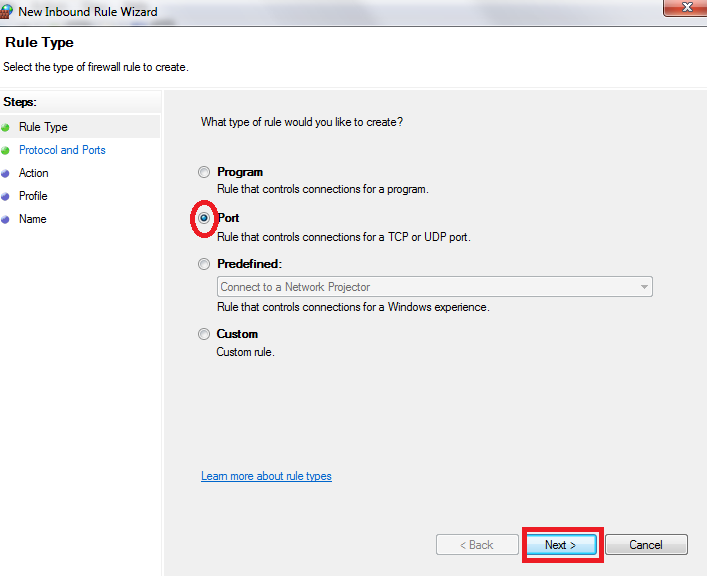
click on advanced settings

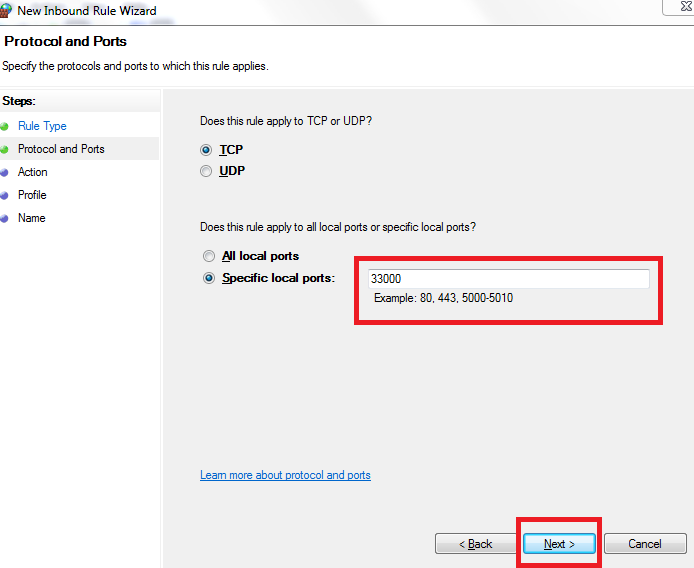


Then new rule

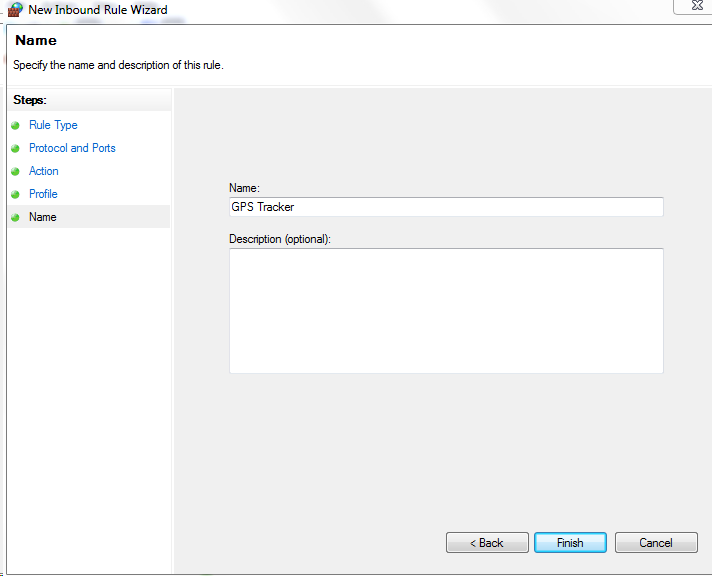


Specify port





We click until the final page appear and click finish.



Now our website is online and can be accessed through internet by its port.

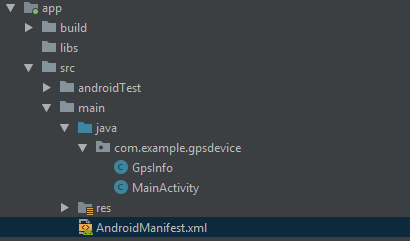
|  |
| --- |
| <!DOCTYPE html> <html>      <head>          <title>              GPS Tracker          </title>          <style>            body{              background-color:darkslategrey            }            button{                font-family:sans-serif;                display:block;                position: absolute;                margin-left: 44%;                margin-right: 40%;                color: blue;                padding: 15px 32px;                text-align: center;            }            div{                margin-left: auto;                margin-right: auto;                margin-top: 6%;                margin-bottom: auto;                           }            add {                color: red;                position: absolute;                top: 80px;                left: 20px;                           }          </style>        <script src="http://maps.googleapis.com/maps/api/js?key=#AIzaSyCGR0AEjSkP-22MPQIiG33LfUsnUhUDNQ0&sensor=false">        </script>        <script src="https://www.gstatic.com/firebasejs/5.5.3/firebase.js"></script>      </head>       <body>          <h1 style="color:white; text-align: center">            MY LOCATION          </h1>           <button onclick="getInfo()" type="button", text-align: center>            Get Location          </button>          <add>            <h2 id="demo"></h2>          </add>                    <div id="googleMap" style="width:1000px;height:580px;",></div>      </body> </html> <script>  *// create firebase config*  const firebaseConfig = {         apiKey: "AIzaSyArEPcIs8WAFnV91hYjNGdgRGKzkVvWFPc",      authDomain: "gpsdevice-490ee.firebaseapp.com",      databaseURL: "https://gpsdevice-490ee.firebaseio.com",      projectId: "gpsdevice-490ee",      storageBucket: "gpsdevice-490ee.appspot.com",      messagingSenderId: "918934474657",      appId: "1:918934474657:web:328d3cce22eda7d9" };  *// get global variable ref*  var ref;  *// the list for append the way device move*  var direction = [];  *// Syntax*  window.onload=function(){      console.log("Initilize database")      firebase.initializeApp(firebaseConfig);      ref = firebase.database().ref();          console.log("Firebase", ref)   };   function getInfo() {      *// make sure real time update*      ref.on("value", function (snapshot) {          var gps = snapshot.val().data.gps;          var data = {          "lat": gps.latitude,          "long": gps.longtitude,          "velocity": gps.velocity          }          *// call the below function*          showMap(data)          *// draw it down*               }, function (error) {          console.log("Error: " + error.code);      });   }    function showMap(data)    {      var mycenter = {lat: data.lat, lng: data.long};      var mapProp =      {        center:new google.maps.LatLng(data.lat,data.long),        zoom:20,        mapTypeId:google.maps.MapTypeId.ROADMAP      };      var map=new google.maps.Map(document.getElementById("googleMap"), mapProp);      var marker=new google.maps.Marker({          position:mycenter,          map:map          });      direction.push({lat: data.lat, lng: data.long})      console.log("velocity: "+ data.velocity)      console.log(direction)      document.getElementById("demo").innerHTML = 'Your velocity: ' + data.velocity + 'm/s';      *// get a new var into Google API*      console.log(direction)      var devicePath = new google.maps.Polyline({          path: direction,          geodesic: true,          strokeColor: '#FF0000',          strokeOpacity: 1.0,          strokeWeight: 2          });      devicePath.setMap(map);    };    *//google.maps.event.addDomListener(window, 'load', showMap);*   </script> |

**IV/ GPS detector android app:**

**1/ In Manifest:**

****

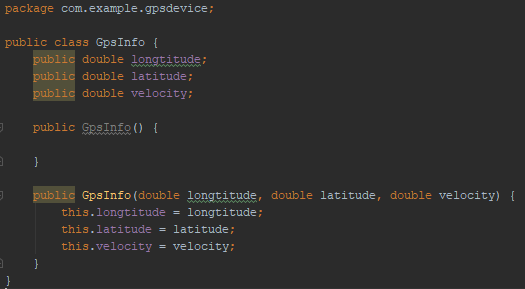
\_ We declare that we will use this package for project, in that will have the main java (The MainActivity).





\_ After that we need permission for our app to get location.

**2/ Initial java function:**

****

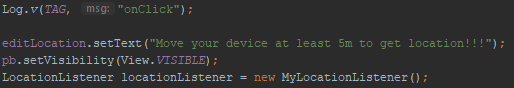
Create new class for define the variable we will push on Firebase.

**3/ Main java function:**



\_ Declare timestamp in order to get the time in miliseconds whenever we run the app.

\_ length and time is two global variable that will be explained later.



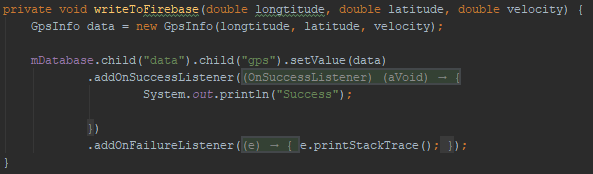
\_ When we click on the icon “Get location”, it will ask use “Move your device at least 5m to get location!!!”

\_ Then set it visible.

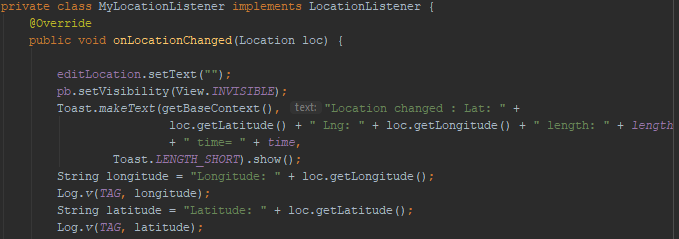
\_ The variable locationListener is declare by MyLocationListener (an basic class in android studio which already have in any Android OS from Android 6.0 to above).



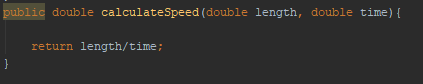
\_ requestLocationUpdates is a loop function of Android 6.0 allow us to call the function which get location again and again with some conditions. It has 4 parameters, first is syntax open Location, second is minimum time we will take update, third is minimum distance that we will take update, fourth is the function that we call again when the condition is satisfying.



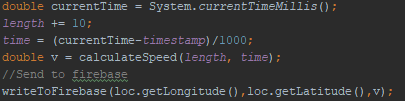
\_ This function helps us to push the data onto Firebase, with the variable GpsInfo that we declared on top.



\_ This class help us get location and show the message whenever the location change.



\_ It help calculate speed by get length divide to time.



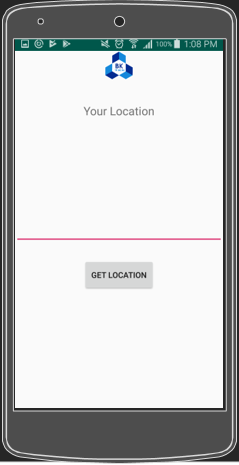
\_ Declare currentTime is a variable take the current time whenever the class is called.

\_ time will equal to currentTime minus with timestamp which we declared on top, then because it is on miliseconds so I devide it by 1000 to convert to second.

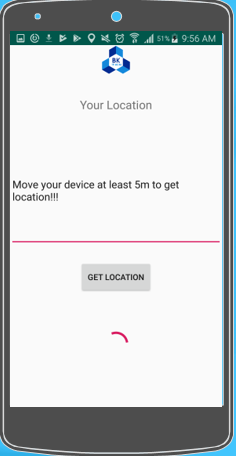
\_ On top, we set after 10 meters the function will call 1 time so everytime the function is called, length plus 10 meters

\_ Finally, last line help we call writeToFirebase and push all data on it.

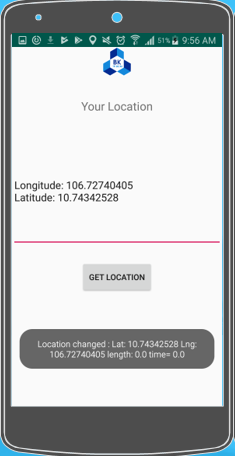
**V/ Results:**



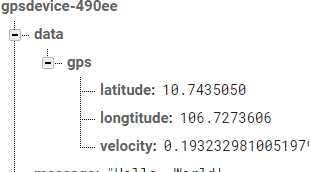
Our App User Interface



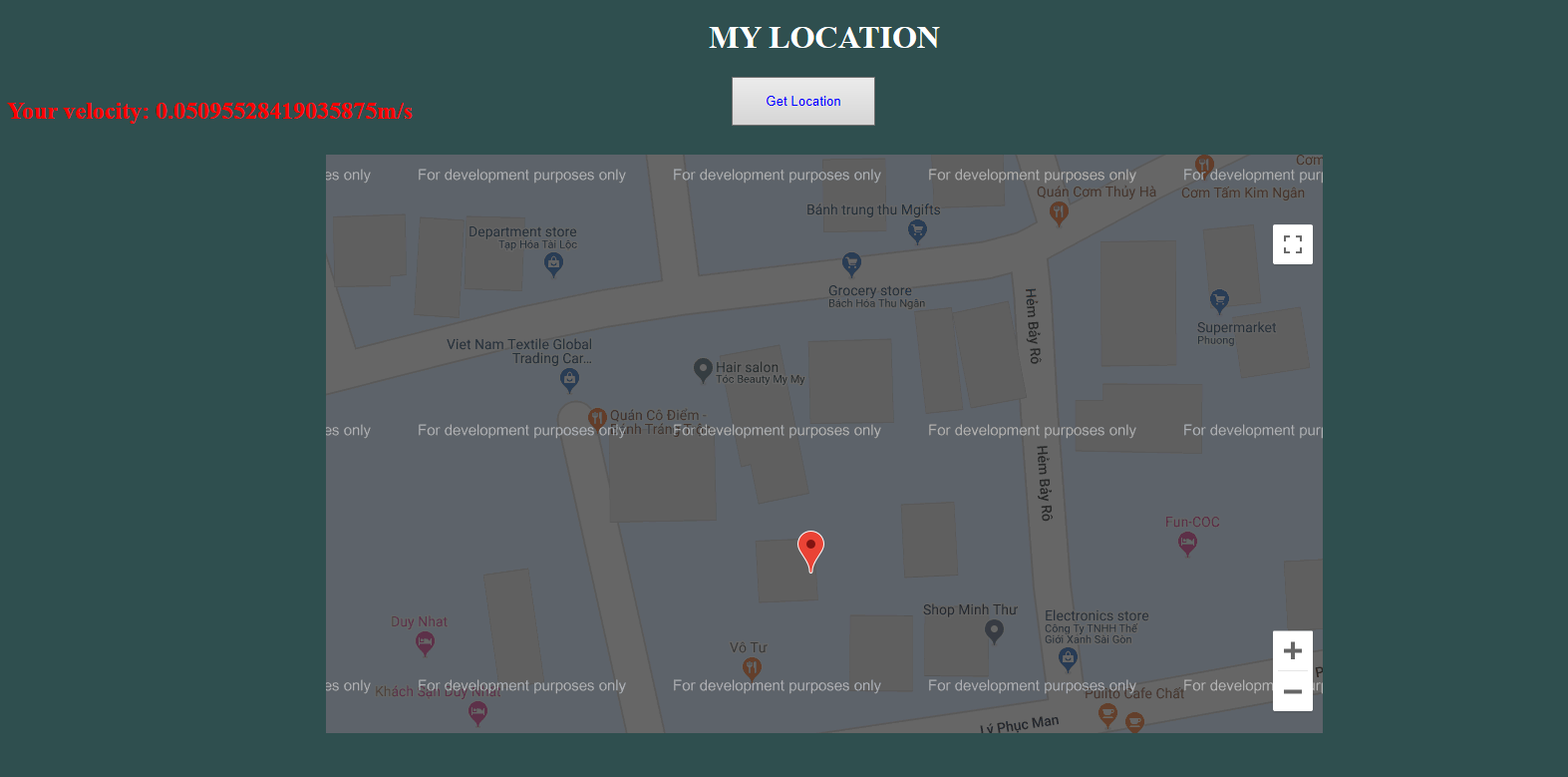
\_ When we push “Get Location”



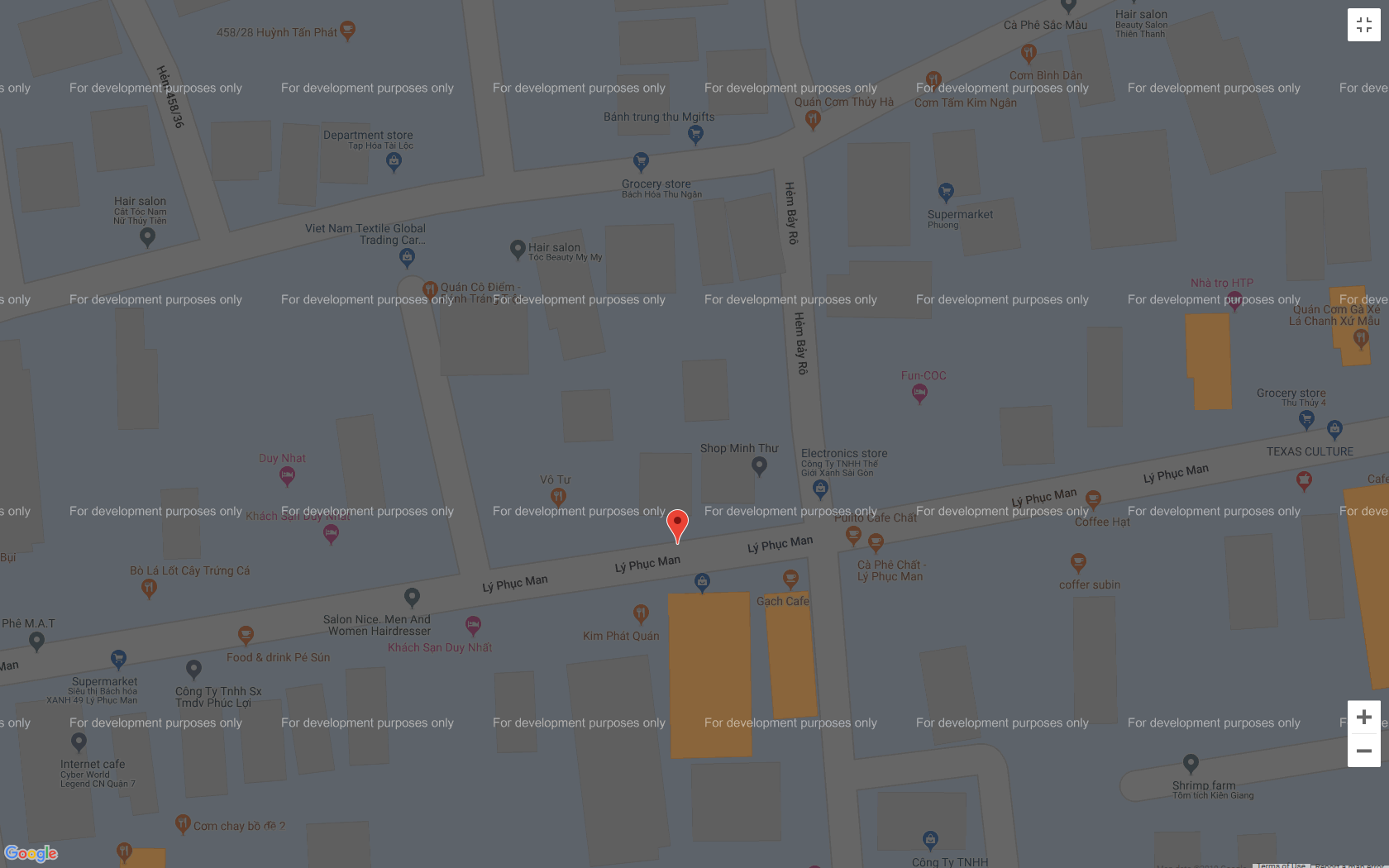
\_ Then, when it take successful, it will push on Firebase



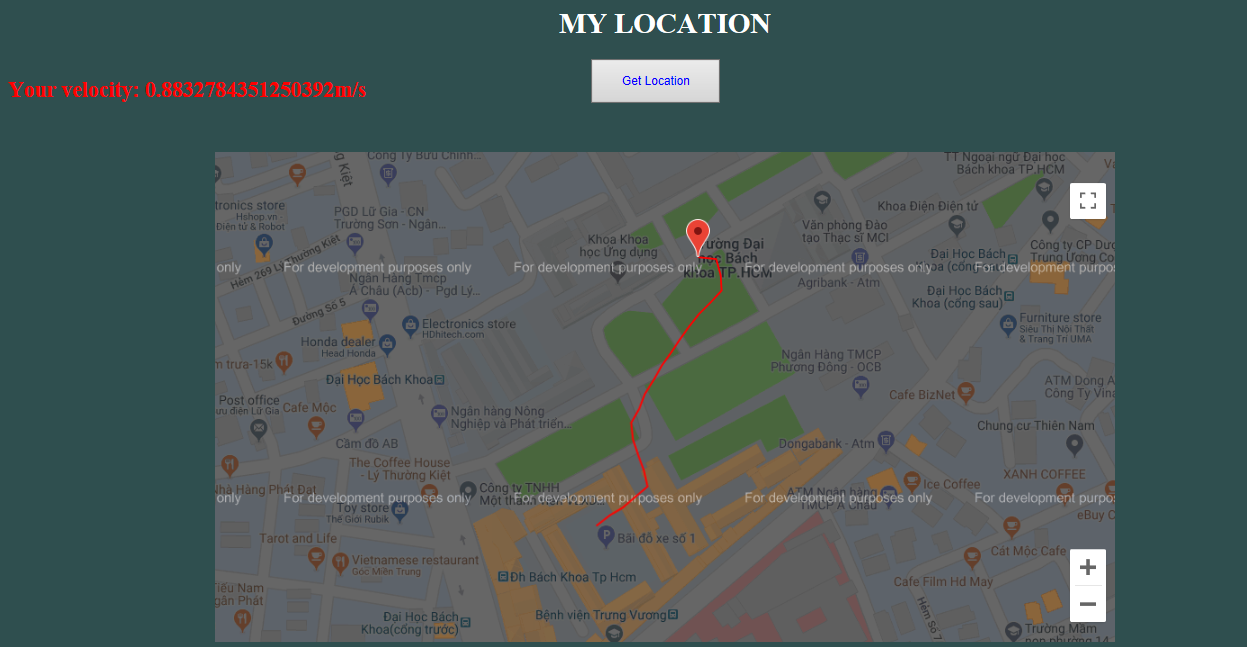
\_ The Firebase get the result



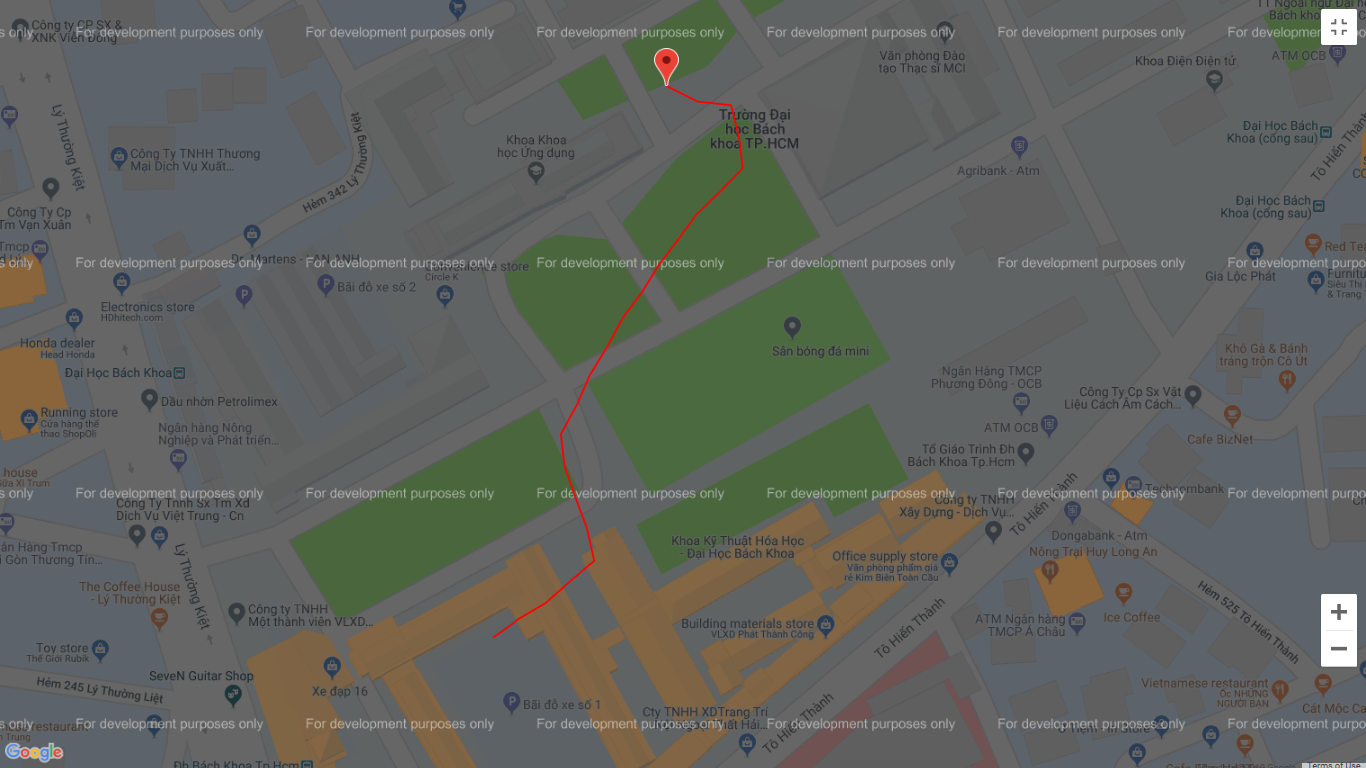
\_ The web will take these data to show.



\_ Get the current location of device.



\_ Tracking when we move



**Full code:**

**+ Manifest:**

|  |  |
| --- | --- |
| |  | | --- | | <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android"    package="com.example.gpsdevice">     <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />    <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />     <application        android:allowBackup="true"        android:icon="@mipmap/ic\_launcher"        android:label="@string/app\_name"        android:roundIcon="@mipmap/ic\_launcher\_round"        android:supportsRtl="true"        android:theme="@style/AppTheme">        <activity android:name=".MainActivity">            <intent-filter>                <action android:name="android.intent.action.MAIN" />                 <category android:name="android.intent.category.LAUNCHER" />            </intent-filter>        </activity>    </application>  </manifest> | |

**+ User Interface (activity\_main.xml):**

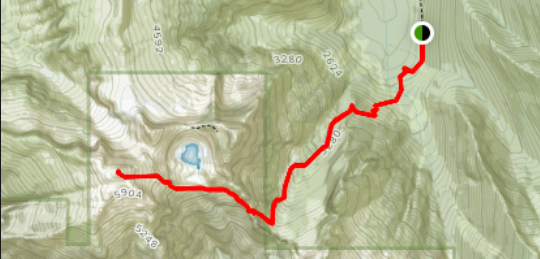
|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout    xmlns:android="http://schemas.android.com/apk/res/android"    xmlns:app="http://schemas.android.com/apk/res-auto"    xmlns:tools="http://schemas.android.com/tools"    android:orientation="vertical"    android:layout\_width="fill\_parent"    android:layout\_height="fill\_parent"    android:weightSum="1">     <ImageView        android:id="@+id/imageView"        android:layout\_width="50dp"        android:layout\_height="50dp"        android:layout\_weight="0.0"        android:layout\_gravity="center\_horizontal"        android:src="@drawable/bk" />     <TextView        android:layout\_width="fill\_parent"        android:layout\_height="wrap\_content"        android:text="Your Location"        android:layout\_weight="0.20"        android:gravity="center"        android:textSize="20sp"        tools:ignore="HardcodedText" />    <EditText        android:layout\_width="match\_parent"        android:layout\_height="wrap\_content"        android:layout\_weight="0.33"        android:id="@+id/editTextLocation"        android:editable="false"        android:autofillHints="">         <requestFocus />    </EditText>    <LinearLayout        android:id="@+id/layButtonH"        android:layout\_height="wrap\_content"        android:layout\_width="fill\_parent"        android:gravity="center"        android:layout\_weight="0.15">         <Button            android:layout\_width="wrap\_content"            android:layout\_height="wrap\_content"            android:text="Get Location"            android:id="@+id/btnLocation"            tools:ignore="HardcodedText" />    </LinearLayout>    <LinearLayout        android:id="@+id/layloadingH"        android:layout\_height="wrap\_content"        android:layout\_weight="0.20"        android:layout\_width="fill\_parent"        android:gravity="center">         <ProgressBar            android:layout\_width="wrap\_content"            android:id="@+id/progressBar1"            android:layout\_height="wrap\_content" />    </LinearLayout> </LinearLayout> |

**+ Main java Function:**

|  |
| --- |
| package com.example.gpsdevice;   import android.Manifest; import android.annotation.SuppressLint; import android.app.Activity; import android.app.AlertDialog; import android.content.ContentResolver; import android.content.Context; import android.content.DialogInterface; import android.content.Intent; import android.content.pm.ActivityInfo; import android.content.pm.PackageManager; import android.location.Address; import android.location.Geocoder; import android.location.Location; import android.location.LocationListener; import android.location.LocationManager; import android.os.Bundle; import android.provider.Settings; import android.support.annotation.NonNull; import android.support.v4.app.ActivityCompat; import android.util.Log; import android.view.View; import android.view.View.OnClickListener; import android.widget.Button; import android.widget.EditText; import android.widget.ProgressBar; import android.widget.Toast;  import com.google.android.gms.tasks.OnFailureListener; import com.google.android.gms.tasks.OnSuccessListener; import com.google.android.gms.tasks.Task; import com.google.firebase.database.DatabaseReference; import com.google.firebase.database.FirebaseDatabase;  import java.io.IOException; import java.util.List; import java.util.Locale;   @SuppressLint("Registered") public class MainActivity extends Activity        implements OnClickListener {     private LocationManager locationMangaer = null;     private EditText editLocation = null;    private ProgressBar pb = null;    private DatabaseReference mDatabase;    private static final String TAG = "Debug";    private static double timestamp = System.currentTimeMillis();    private static double length = 0;    private static double time = 0;    @SuppressLint("CutPasteId")    @Override    public void onCreate(Bundle savedInstanceState) {        super.onCreate(savedInstanceState);        setContentView(R.layout.activity\_main);        *//firebase init*        mDatabase = FirebaseDatabase.getInstance().getReference();          *//if you want to lock screen for always Portrait mode*        setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_PORTRAIT);         pb = (ProgressBar) findViewById(R.id.progressBar1);        pb.setVisibility(View.INVISIBLE);         editLocation = (EditText) findViewById(R.id.editTextLocation);         Button btnGetLocation = (Button) findViewById(R.id.btnLocation);        btnGetLocation.setOnClickListener(this);         locationMangaer = (LocationManager)                getSystemService(Context.LOCATION\_SERVICE);     }     @SuppressLint("SetTextI18n")    @Override    public void onClick(View v) {         Boolean flag = displayGpsStatus();         if (flag) {             Log.v(TAG, "onClick");             editLocation.setText("Move your device at least 5m to get location!!!");            pb.setVisibility(View.VISIBLE);            LocationListener locationListener = new MyLocationListener();             if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED                    && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION)                    != PackageManager.PERMISSION\_GRANTED) {                *// TODO: Consider calling*                *//    ActivityCompat#requestPermissions*                *// here to request the missing permissions, and then overriding*                *//   public void onRequestPermissionsResult(int requestCode, String[] permissions,*                *//                                          int[] grantResults)*                *// to handle the case where the user grants the permission. See the documentation*                *// for ActivityCompat#requestPermissions for more details.*                return;            }             locationMangaer.requestLocationUpdates(LocationManager                    .GPS\_PROVIDER, 10, 10, locationListener);        } else {            alertbox();        }     }     */\*----Method to Check GPS is enable or disable ----- \*/*    private Boolean displayGpsStatus() {        ContentResolver contentResolver = getBaseContext()                .getContentResolver();        return Settings.Secure                .isLocationProviderEnabled(contentResolver,                        LocationManager.GPS\_PROVIDER);    }     */\*----------Method to create an AlertBox ------------- \*/*    protected void alertbox() {        AlertDialog.Builder builder = new AlertDialog.Builder(this);        builder.setMessage("Your Device's GPS is Disable")                .setCancelable(false)                .setTitle("\*\* Gps Status \*\*")                .setPositiveButton("Gps On",                        new DialogInterface.OnClickListener() {                            public void onClick(DialogInterface dialog, int id) {                                *// finish the current activity*                                *// AlertBoxAdvance.this.finish();*                                Intent myIntent = new Intent(                                        Settings.ACTION\_SECURITY\_SETTINGS);                                startActivity(myIntent);                                dialog.cancel();                            }                        })                .setNegativeButton("Cancel",                        new DialogInterface.OnClickListener() {                            public void onClick(DialogInterface dialog, int id) {                                *// cancel the dialog box*                                dialog.cancel();                            }                        });        AlertDialog alert = builder.create();        alert.show();    }     private void writeToFirebase(double longtitude, double latitude, double velocity) {        GpsInfo data = new GpsInfo(longtitude, latitude, velocity);         mDatabase.child("data").child("gps").setValue(data)                .addOnSuccessListener(new OnSuccessListener<Void>() {                    @Override                    public void onSuccess(Void aVoid) {                        System.out.println("Success");                     }                })                .addOnFailureListener(new OnFailureListener() {                    @Override                    public void onFailure(@NonNull Exception e) {                        e.printStackTrace();                    }                });    }     */\*----------Litener class to get coordinates ------------- \*/*    private class MyLocationListener implements LocationListener {        @Override        public void onLocationChanged(Location loc) {             editLocation.setText("");            pb.setVisibility(View.INVISIBLE);            Toast.makeText(getBaseContext(), "Location changed : Lat: " +                            loc.getLatitude() + " Lng: " + loc.getLongitude() + " length: " + length                            + " time= " + time,                    Toast.LENGTH\_SHORT).show();            String longitude = "Longitude: " + loc.getLongitude();            Log.v(TAG, longitude);            String latitude = "Latitude: " + loc.getLatitude();            Log.v(TAG, latitude);            *//TODO*            *//calculte speed*            double currentTime = System.currentTimeMillis();            length += 10;            time = (currentTime-timestamp)/1000;            double v = calculateSpeed(length, time);            *//Send to firebase*            writeToFirebase(loc.getLongitude(),loc.getLatitude(),v);            *// This registration token comes from the client FCM SDKs.*            *// Write a message to the database*            FirebaseDatabase database = FirebaseDatabase.getInstance();              */\*----------to get City-Name from coordinates ------------- \*/*            String cityName = null;            Geocoder gcd = new Geocoder(getBaseContext(),                    Locale.getDefault());            List<Address> addresses;            try {                addresses = gcd.getFromLocation(loc.getLatitude(), loc                        .getLongitude(), 1);                if (addresses.size() > 0)                    System.out.println(addresses.get(0).getLocality());                cityName = addresses.get(0).getLocality();            } catch (IOException e) {                e.printStackTrace();            }             String s = longitude + "\n" + latitude; *//                    +"\n\nMy Currrent City is: " + cityName;*            editLocation.setText(s);        }         @Override        public void onProviderDisabled(String provider) {            *// TODO Auto-generated method stub*        }         @Override        public void onProviderEnabled(String provider) {            *// TODO Auto-generated method stub*        }         @Override        public void onStatusChanged(String provider,                                    int status, Bundle extras) {            *// TODO Auto-generated method stub*        }        public double calculateSpeed(double length, double time){             return length/time;        }    }  } |

**VI/ Applications:**

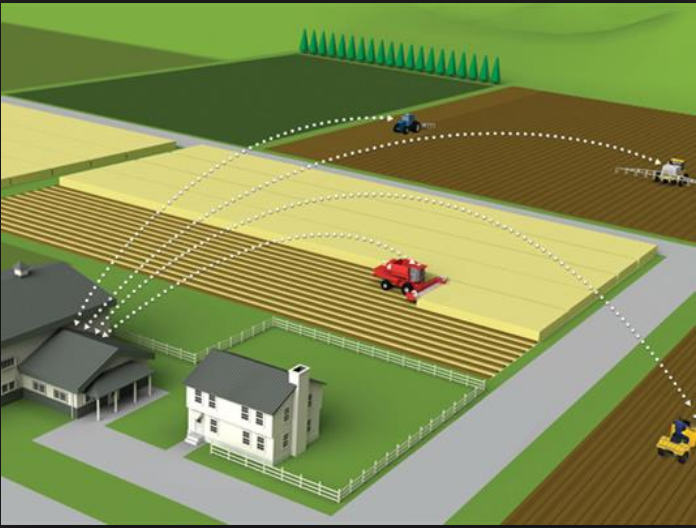
\_ Climbing: when following GPS route in the field, also keep our map app handy with your route and waypoints marked and named on it, so that you can understand better where you are at any moment and also verify that you are indeed headed for correct next point and can easy find the way back.



\_ Security: find your phone, by our app you can find your phone on everywhere.



\_ Farming: Control machine on farm always move in correct way that you want.



**VII/ References:**

* Stack Overflow: <https://stackoverflow.com/>
* Wikimedia: <https://en.wikipedia.org/wiki/Global_Positioning_System>
* W3School: <https://www.w3schools.com/>
* Android Studio: <https://developer.android.com/docs>
* Devside: <https://www.devside.net/wamp-server/accessing-websites-on-a-local-network-lan-web-server>
* Github: <https://github.com/>