```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include<SoftwareSerial.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
static const int RXPin = D6, TXPin = D5;
static const uint32_t GPSBaud = 9600;
// The TinyGPS++ object
TinyGPSPlus gps;
// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);
#define BLYNK_TEMPLATE_ID "TMPLsnntq-8q"
#define BLYNK_DEVICE_NAME "ACCIDENT DETECTION"
#define BLYNK_AUTH_TOKEN "6mEwzzDqTN5XgArXdYOOevb3lYju7NYN"

#include "Wire.h"
#include <MPU6050_light.h>
#include<LiquidCrystal_I2C.h>




#include <Wire.h>
int motorpin = D8;
int BUZZER_PIN =D7;
String msg1,msg2;
LiquidCrystal_I2C lcd(0X3F,16,2);
MPU6050 mpu(Wire);
unsigned long timer = 0;


char auth[] = "6mEwzzDqTN5XgArXdYOOevb3lYju7NYN";
char ssid []= "vivo";
char pass[] = "samvedh2020";

void setup() {

  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  ss.begin(GPSBaud);
  pinMode(motorpin,OUTPUT);
  pinMode(BUZZER_PIN,OUTPUT);
  digitalWrite(motorpin,HIGH);



 Wire.begin();
  byte status = mpu.begin();
  Serial.print(F("MPU6050 status: "));
  Serial.println(status);
  while(status!=0){ } // stop everything if could not connect to MPU6050

  Serial.println(F("Calculating offsets, do not move MPU6050"));
  delay(1000);
  mpu.calcOffsets(true,true); // gyro and accelero
  Serial.println("Done!\n");
  Blynk.begin(auth,ssid,pass, "blynk.cloud", 80);

}

void loop()
{
  Blynk.run();
  //digitalWrite(motorpin,HIGH);

  lcd.setCursor(0,0);
  mpu.update();

  if(millis() - timer > 1000)
  { // print data every second


   Serial.print(F("TEMPERATURE: "));Serial.println(mpu.getTemp());
    Serial.print(F("ACCELERO  X: "));Serial.print(mpu.getAccX());
    Serial.print("\tY: ");Serial.print(mpu.getAccY());
    Serial.print("\tZ: ");Serial.println(mpu.getAccZ());

    Serial.print(F("GYRO      X: "));Serial.print(mpu.getGyroX());
    Serial.print("\tY: ");Serial.print(mpu.getGyroY());
    Serial.print("\tZ: ");Serial.println(mpu.getGyroZ());

    Serial.print(F("ACC ANGLE X: "));Serial.print(mpu.getAccAngleX());
    Serial.print("\tY: ");Serial.println(mpu.getAccAngleY());

    Serial.print(F("ANGLE     X: "));Serial.print(mpu.getAngleX());
    Serial.print("\tY: ");Serial.print(mpu.getAngleY());
    Serial.print("\tZ: ");Serial.println(mpu.getAngleZ());
    Serial.println(F("==================================================\n"));
    timer = millis();
     if(mpu.getAngleY()>60)

     Serial.println("ACCIDENTDETECTED");
     lcd.setCursor(0,0);
     lcd.print("ACCIDENT DETECTED");
     for (int positionCounter = 0; positionCounter < 15; positionCounter++)
     {


    lcd.scrollDisplayLeft();
```

```
     delay(150);
     }
    digitalWrite(motorpin,LOW);
    digitalWrite(BUZZER_PIN,HIGH);

    gpss();

    }
   else
   {
    Serial.println("VEHICLE MOVE");
     lcd.setCursor(0,0);
     lcd.print("VEHICLE MOVE");
     for (int positionCounter = 0; positionCounter < 12; positionCounter++)
     {
     lcd.scrollDisplayLeft();
     delay(150);
     }
     digitalWrite(motorpin,HIGH);
     digitalWrite(BUZZER_PIN,LOW);
    //Blynk.virtualWrite(V6,"NOT MOVED");
   }
  }


}
void gpss()
 {
  while (ss.available() > 0)
  {
    gps.encode(ss.read());
    if (gps.location.isUpdated())
    {
      String longitude=String(gps.location.lat(), 6);
      String latitude=String(gps.location.lng(), 6);
      String msg1="lat"+longitude;
      String msg2="lng"+latitude;
      Blynk.virtualWrite(V6,msg1);
      Blynk.virtualWrite(V7,msg2);
      lcd.setCursor(0,0);
      lcd.print(msg1);
      lcd.setCursor(1,0);
      lcd.print(msg2);

     }
  }
 }
/////////////////////////
```