IoT Based Vehicle Accident Detection & Rescue Information System

Submitted by

SAYMUM AHMMED SANY

2013-2-60-042

MD. AL-MAMUN RIYADH

2013-2-60-060

Supervised by

SURAJIT DAS BARMAN

Senior Lecturer

Department of Computer Science & Engineering (CSE)

East West University

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science & Engineering (CSE)

East West University, Dhaka, Bangladesh

August, 2017

DECLARATION

We, hereby, declare that the work presented in this thesis is the outcome of the project performed by us under the supervision of Surajit Das Barman, Lecturer, Department of Computer Science and engineering, East West University, Dhaka, Bangladesh. We also declare that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Candidate's Signature

. . . . . .

SAYMUM AHMMED SANY

ID: 2013-2-60-042

. . . . . .

MD. AL-MAMUN RIYADH

ID: 2013-2-60-060

LETTER OF ACCEPTANCE

This Thesis Project entitled **"IoT Based Vehicle Accident Detection and Rescue Information System"** submitted by Saymum Ahmmed Sany (ID: 2013-2-60-042) and MD. AL-MAMUN RIYADH (ID: 2013-2-60-060) to the Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted by the department in partial fulfillment of requirements for the Award of the Degree of Bachelor of Science and Engineering on August, 2017.

Supervisor

. . . . . .

SURAJIT DAS BARMAN

Lecturer, Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

Chairperson

. . . . . .

DR. MD. MOZAMMEL HUQ AZAD KHAN

Chairperson and Professor, Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

ABSTRACT

In this project, an IoT based vehicle accident detection and rescue information system is developed in order to detect vehicle accident and send the location information of the accident place to vehicle owner, nearest hospital and police station via a web service. The communication between the web server and hardware device is established via GSM/GPRS shield, and the location is traced by using the GPS shield. The accident is detected through vibration sensors, keypad and buzzer. The project is developed for real time data fetching form the hardware device using sensors and store in the web server, and send notification to different users either through web application, android mobile application or SMS. This project approximately provides the accurate detection of the location of accident occurred, and send notification to the nearest police station and hospital.

ACKNOWLEDGMENTS

First of all, we would like to thank almighty Allah for giving us strength, patience and knowledge to complete this project work.

We would like to express our sincere gratitude to our supervisor Surajit Das Barman for the continuous support, for his patience, motivation, and immense knowledge. His guidance helped us in all the time of work and writing of this thesis. We could not have imagined having a better supervisor and mentor for this project.

We would like to thank our friends for their feedback, cooperation and of course friendship.

In addition, we would like to express our gratitude to the faculty members of the Department of Computer Science and Engineering (CSE), East West University, Dhaka, Bangladesh who helped us with their feedback.

Last but not least, we would like to thank our family, our parents and to our brothers and sisters for supporting us.

SAYMUM AHMMED SANY & MD AL-MAMUN RIYADH

August, 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

GPRS	General Packet Radio Service
GSM	Global System For Mobile Communication
GPS	Global Positioning System
VTS	Vehicle Tracking Syetem
HTTP	Hypertext Tranfer Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 **Background**

The Internet of Things (IoT) is an arrangement of interrelated computing gadgets, mechanical and digital machines, objects, animals or individuals that are given one kind of an identifiers and the capacity to exchange information over a system without requiring human-to-human or human-to-PC communication. IoT is a new concept that has evolved from the convergence of wireless technologies. Wireless communication is the transfer of information or signal between two or more points that are not connected by an electrical conductor. In IoT devices equipped with Wi-Fi allow the machine-to-machine communication.

Using this from of industrial machines to wearable or wireless devices, using built-in sensors to gather data and take action on that data across a network. The sensor and actuator can be setup in different place but they are working together over an internet network.

Using IoT technique a vehicle tracking system (VTS) can be built. A vehicle tracking system combines the use of automatic vehicle location of individual vehicles with software that collects these fleet data for a comprehensive picture of vehicle locations. Modern vehicle tracking systems commonly use GPS or GLONASS technology for locating the vehicle, but other types of automatic vehicle location technology can also be used. Vehicle information can be viewed on electronic maps via internet with specialized software. The history of vehicle tracking dates to the beginning of GPS technology in 1978. In the early years, the technology was not yet operational, due to an insufficient number of satellites orbiting the earth. On Jan. 17, 1994, after years of gradual growth, the final of the first 24 satellites was launched, and the GPS system was considered fully operational. Early GPS was designed primarily only for military but in 1996, President Bill Clinton determined that the system would be an asset to civilians as well as the military. This policy change made GPS technology available to the average individual, including fleet managers, who could see the benefit of using the technology to keep tabs on their vehicles. In the early days of fleet tracking, in order to properly track a fleet, each vehicle had to be enabled with a costly GPS device. The company was required to pay a typically high monthly fee to use the satellite tracking system. While helpful, these early systems were difficult to implement, costly to use and sometimes inconvenient for drivers and fleet management alike. Thus it took several

years for the concept to catch on. In the earliest days, only large, wealthy fleets took advantage of the technology. The modern fleet tracking system provides the necessary data to fleet managers allowing them to run their operations more efficiently. Reports on driver behavior, vehicle performance and fuel use all make it easier for the fleet manager to cut costs and increase efficiencies. These systems go beyond simple reporting of each vehicle's location, offering fleet managers a wealth of information about their vehicles and their drivers [1].

Major constituents of the GPS-based tracking are:

1. **GPS tracking:** The device fits into the vehicle and captures the GPS location information apart from other vehicle information at regular intervals to a central server. Other vehicle information can include fuel amount, engine temperature, altitude, reverse geocoding, door open/close, tire pressure, cut off fuel, turn off ignition, turn on headlight, turn on taillight, battery status, GSM area code/cell code decoded, number of GPS satellites in view, glass open/close, fuel amount, emergency button status, cumulative idling, computed odometer, engine RPM, throttle position, GPRS status and a lot more. Capability of these devices actually decide the final capability of the whole tracking system; most vehicle tracking systems, in addition to providing the vehicle's location data, feature a wide range of communication ports that can be used to integrate other on board systems, allowing to check their status and control or automate their operation.
2. **GPS tracking server:** The tracking server has three responsibilities: receiving data from the GPS tracking unit, securely storing it, and serving this information on demand to the user.
3. **User interface:** The UI determines how one will be able to access information, view vehicle data, and elicit important details from it [2].

In Bangladesh this VTS is available. There are some company like GP, ROBI provides Vehicle Tracking Service (VTS) which has some common features like tracking the vehicle using satellite GPS & GSM communication. But there is no system which can detect accident and also give the service of VTS. Here comes carsafe project which can detect speed of a car, location of a car, and if there is any accident occur it can communicate automatically to the nearest police station, hospital and owner to reduce instant loss or damage.

1.2 **Objective**

Here the following objectives are set, in the view of above mentioned research background for the present work in VTS, accident detection and rescue information system.

1. To design a vehicle accident detection and rescue information system based on GSM/GPRS/GPS technology.
2. To implement a web service for the vehicle owner, nearest police station and hospital to receive notification about the accident occurance and its location.

1.3 **Motivation**

In Bangladesh, there are some company provides vehicle tracking system. Every VTS do the same thing like speed check, track vehicle. Some VTS can detect an accident. But there is no VTS that gives notification to the nearest hospital and police station if any accident occurs. In Bangladesh, where a person who faces an accident get less facility in treatment there need a system that can communicate to the nearest hospital and police station. Statistics shows, there are more than 3,000 people die on Bangladesh's roads every year. The country has one of the highest rates in the world, with more than 85 deaths for every 10,000 registered motor vehicles. That's around 50 times higher than the rate in most western countries. According to the World Health Organization (WHO), road traffic injuries cause a loss of about 2% of GDP in Bangladesh, or about £1.2bn annually. This is almost equal to the total foreign aid received in a fiscal year. The losses include direct and indirect expenses, such as medical costs, insurance loss, property damage, family income losses and traffic congestion [3].

Table 1.1: Road Accident and Casualties Statistics (2009-2016) (Source: Bangladesh Police)

Year	Number Of Accident	Death	Injury
2009	3381	2958	2686
2010	2827	2646	1803
2011	2667	2546	1641

2012	2636	2538	2134
2013	2029	1957	1396
2014	2027	2067	1535
2015	2394	2376	1958
2016 (Up to July)	1489	1422	1289

Now-a-days lots of accidents happen on highways due to increase in traffic and also due to rash driving of the drivers. And in many situations the family members or the ambulance and police authority is not informed in time. This result in delaying the help reached to the person suffered due to an accident. A serious accident occurs at nightmare and it might be not caught sight of other people then the victim is unable to call some emergency services by himself don't get help at the right time. If other may see the accident but they don't have the number of hospital or police station. So It take's lots of time to inform a nearest hospital or police station. That increase the vulnerability of life of the people who got into accident. Besides, Many people died on the way to the hospital due to lack of information about nearest hospital or delay for waiting for the ambulance. This project will implement a system that can detect accident and show the nearest hospital to the car passengers, also a hospital and police station will get notified about any accident occurred near to the hospital and police station. So that, the hospital and police station can send rescue team in very short time. This project will help to reduce the greater loss and damage for any accident.

1.4 **Outline of the report**

The outline of the rest of this report has been structured as follows.

Chapter 2 presents the background study for the project and hardware specification of the components used in the project it also describes the technical previous related works on the topic of VTS and IoT..

Chapter 3 explains methodology of the project that how the project being implemented.

Chapter 4 presents the real implementation of the project. The hardware, web services including website and mobile application.

Chapter 5 outlines conclusion of this work precisely and also describes the scope of future work for possible extended application of the project.

CHAPTER 2: BACKGROUND STUDY

2.1 **IoT : The revolution of Technology**

The Internet of things (IoT) is the between systems administration of physical gadgets, vehicles, structures, and different things installed with hardware, programming, sensors, actuators, and system availability which empower these items to gather and exchange data. The IoT enables items to be detected or controlled remotely across existing system infrastructure creating open doors for more straightforward of the physical world into computer-based systems and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. Only IoT can connect physical world to the web.

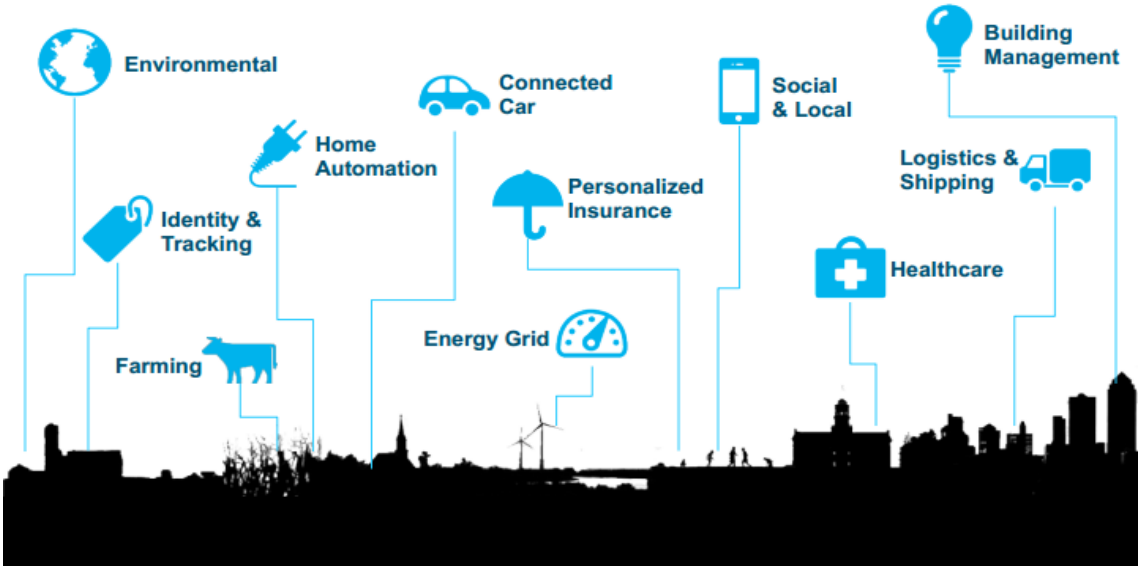

Figure 2.1: IoT connecting the physical world to the web

The IoT is more than internet connected consumer gadgets. Sooner or later every IT organization will need to create an framework to support it. Energy companies already use networked sensors to measure vibrations in turbines. They feed that data through the network to computing systems that analyse it to predict when machines will need maintenance and when they will fail. Jet engine manufacturers embed sensors that measure temperature, pressure, and other conditions to improve their products. Even a gift basket business can deploy sensors to constantly monitor the temperature of perishable products [4].

2.2 **Architecture of IoT**

To do any project or research about IoT having knowledge of architecture of IoT is required. There are four layers of IoT architecture. They are Interface layer, Service layer, Networking layer, Service layer.

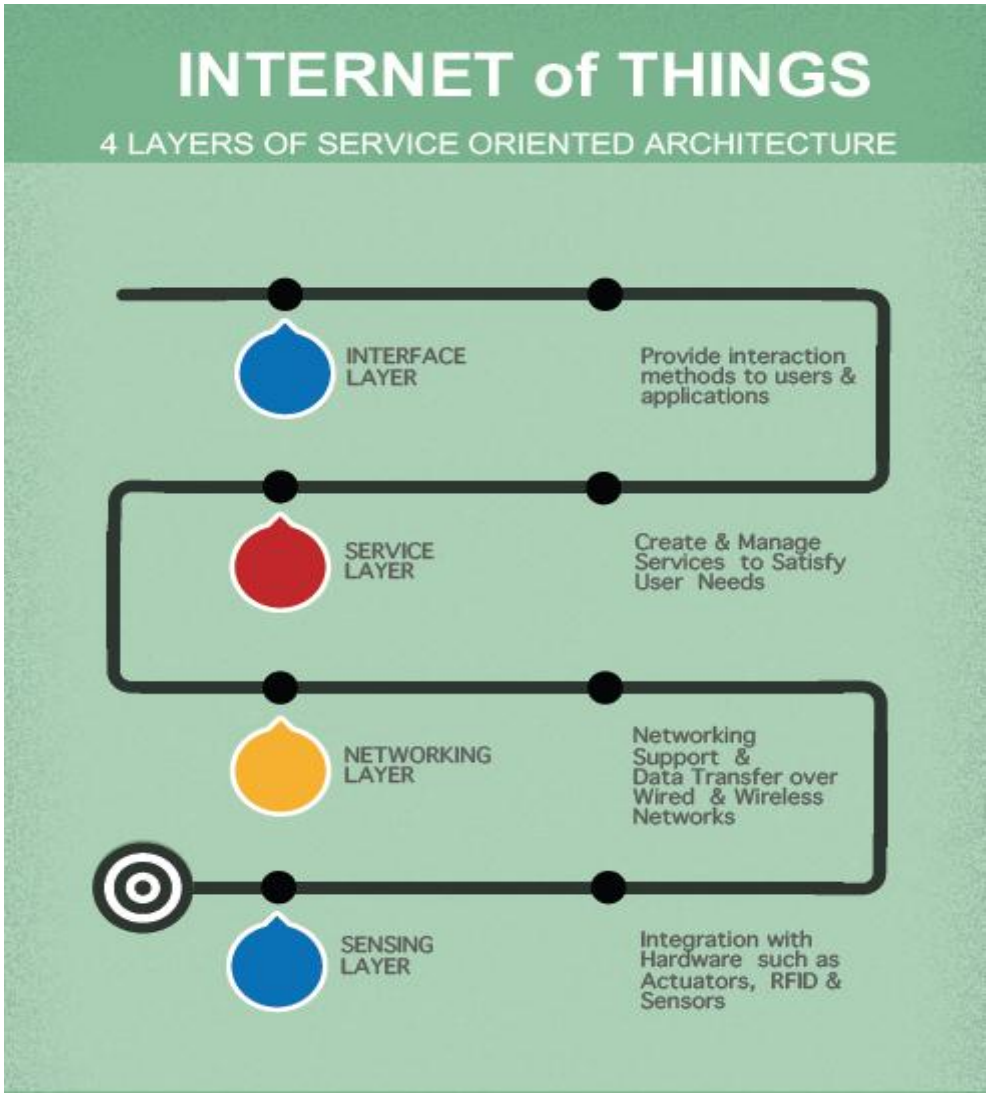

Figure 2.2: IoT Basic layer architecture

2.2.1 **Interface Layer**

The first layer of IoT is interface layer. This layer provides the interaction methods between users and application. This section looks how user can easily used the system. This includes three main approaches. Firstly, we need the ability to create web-based front-ends and portals that interact with devices and with the event-processing layer. Secondly, we need the ability to create dashboards that offer views into analytics and

event processing. Finally, we need to be able to interact with systems outside this network using machine-to-machine communications (APIs).

The recommended approach to building the web front end is to utilize a modular front-end architecture, Web server-side technology, such as Java Servlets/, JSP, PHP, Python, Ruby, etc.

2.2.2 **Service layer**

This layer is used to create and manage services to satisfy users needs. To do so, it process data deep processing. To make more userfrindly application, it provides database with different data and devides work.

This is an important layer for three reasons:

1. The ability to support an HTTP server and/or an MQTT broker to talk to the devices;

2. The ability to aggregate and combine communications from different sensing devices and to route communications to a specifc device (possibly via GSM/GPRS).

3.The ability to bridge and transform between different protocols that is to offer HTTP based APIs that are mediated into an MQTT message going to the device.

2.2.3 **Networking or Communication Layer**

The Networking or Communication layer supports the connectivity of the devices. There are multiple potential protocols for communication between the devices and the cloud. The most wellknown three potential protocols are

- HTTP/HTTPS (and RESTful approaches on those)
- MQTT 3.1/3.1.1
- Constrained application protocol (CoAP)

Let’s take a quick look at each of these protocols in turn

HTTP is well known, and there are many libraries that support it. Because it is a simple textbased protocol, many small devices such as 8-bit controllers can only partially support the protocol – for example enough code to POST or GET a resource. The larger

32-bit based devices can utilize full HTTP client libraries that properly implement the whole protocol.

There are several protocols optimized for IoT use. The two best known are MQTT6 and CoAP7. **MQTT** was invented in 1999 to solve issues in embedded systems and SCADA. It has been through some iterations and the current version (3.1.1) is undergoing standardization in the OASIS MQTT Technical Committee8. MQTT is a publish-subscribe messaging system based on a broker model. The protocol has a very small overhead (as little as 2 bytes permessage), and was designed to support lossy and intermittently connected networks.MQTT was designed to flow over TCP. In addition there is an associated specification designed for ZigBee-style networks called MQTT-SN (Sensor Networks).

CoAP is a protocol from the IETF that is designed to provide a RESTful application protocol modeled on HTTP semantics, but with a much smaller footprint and a binary rather than a text-based approach. CoAP is a more traditional client-server approach rather than a brokered approach. CoAP is designed to be used over UDP [5].

2.2.4 **Sensing Layer**

Sensors collect data from the environment or object under measurement and turn it into useful data. This layer covers everything from legacy industrial devices to robotic camera systems, water-level detectors, air quality sensors, accelerometers, and heart rate monitors. And the scope of the IoT is expanding rapidly, thanks in part to low-power wireless sensor network technologies and Power over Ethernet, which enable devices on a wired LAN to operate without the need for an A/C power source.

2.3 **IoT Platforms and Security**

Even with the recent attention given to security for IoT devices, it can be easy to overlook the need for end-to-end security for an IoT platform. Every part of a platform should be analysed for security prospects. From internet connections to the applications and devices to the transmitted and stored data, there is a potential for an attack vector. Without question, the single most important non-functional requirement of an IoT platform is that it offers robust security.

2.4 **Hardware specification**

2.4.1 **Arduino UNO REV3**

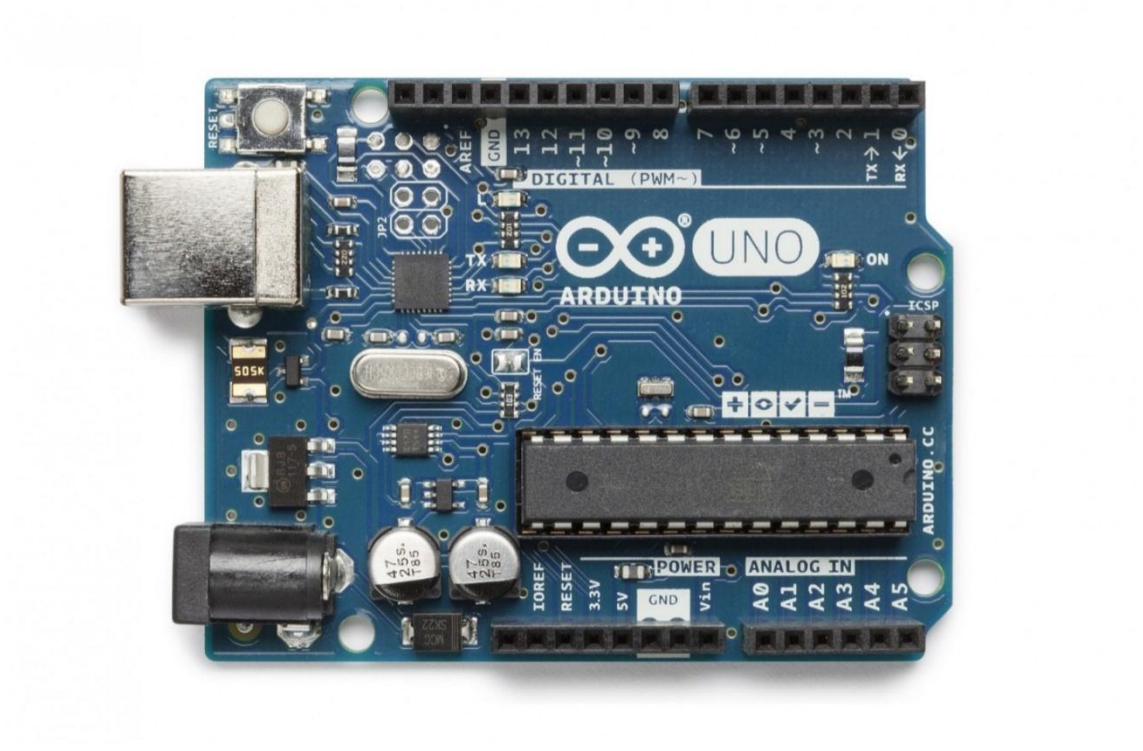

Figure 2.3: Arduino UNO

Table 3.1: Arduino Uno Hardware specification [6].

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6

DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

2.4.2 **GSM, GPS, GSRM Shied (SIM808)**

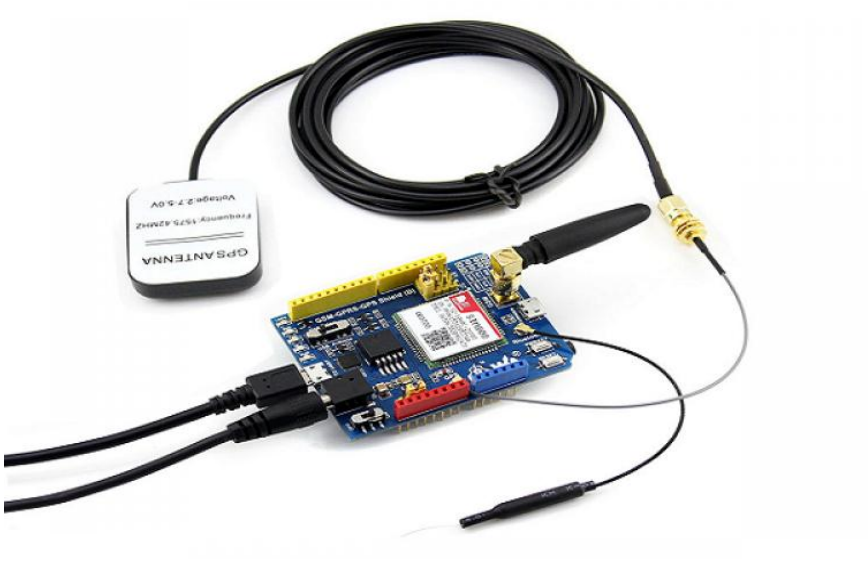

Figure 2.4: GSM/GPRS/GPS/Bluetooth SIM808 Shield (B) [7]

2.4.2.1 **Features**

General

- Arduino connectivity, compatible with UNO, Leonardo, NUCLEO, XNUCLEO
- On board USB TO UART converter CP2102 for UART debugging
- 5 x LEDs for indicating the module working status
- On board voltage level converter, supports both 3.3V and 5V systems
- SIM card slot for 1.8V/3V SIM card
- Baud rate auto detection (1200bps ~115200bps)
- Bluetooth 3.0, supports data transferring through Bluetooth
- RTC with power supply interface
- Firmware upgradable via USB
- Control via AT commands (3GPP TS 27.007,27.005, and SIMCOM enhanced AT Commands)
- Supports SIM application toolkit: GSM 11.14 Release 99
- Operating voltage: 6 ~ 12V
- Operation temperature: -40 °C ~ +85 °C
- Storage temperature: -45 °C ~ +90 °

Table 2.1: Technical Specification on GSM/GPRS of GSM/GPRS/GPS/Bluetooth SIM808 Shield

Band	o GSM 850/EGSM 900/DCS 1800/PCS 1900 MHz o Quad-band auto search o Compliant to GSM phase 2/2+
Emitting power	o Class 4 (2W @ GSM 850/EGSM 900 MHz) o Class 1 (1W @ DCS 1800/PCS 1900 MHz)
GPRS connectivity	o GPRS multi-slot class 12 (default) o GPRS multi-slot class 1~12 (configurable)
GPRS data feature	o Downlink speed: max 85.6kbps o Uplink speed: max 85.6kbps o Coding schemes: CS-1\CS-2\CS-3\CS-4 o Supports PAP (Password Authentication Protocol) for PPP connection o Embedded TCP/IP protocol o Supports PBCCH o Supports CSD transmission speed: 2.4/4.8/9.6/14.4 kbps o Supports USSD
SMS	o Supports: MT/MO/CB/Text/PDU mode o SMS storage: SIM card
Audio	o Voice encode/decode mode: Half Rate\Full Rate\Enhanced Full Rate\Adaptive multi rate o Supports echo cancellation o Supports noise reduction

Table 2.2: Technical Specification on GPS of GSM/GPRS/GPS/Bluetooth SIM808 Shield

Receiver type	○ 22 tracking channels ○ 66 acquisition channels ○ GPS L1 C/A code
Sensitivity	○ Tracking: -165 dBm ○ Cold starts : -148 dBm
Time-To-First-Fix	○ Cold starts : 30s (typ.) ○ Hot starts : < 1s ○ Warm starts: 28s
Accuracy	○ Horizontal position : <2.5m CEP

2.4.2.2 **Board Description [12]**

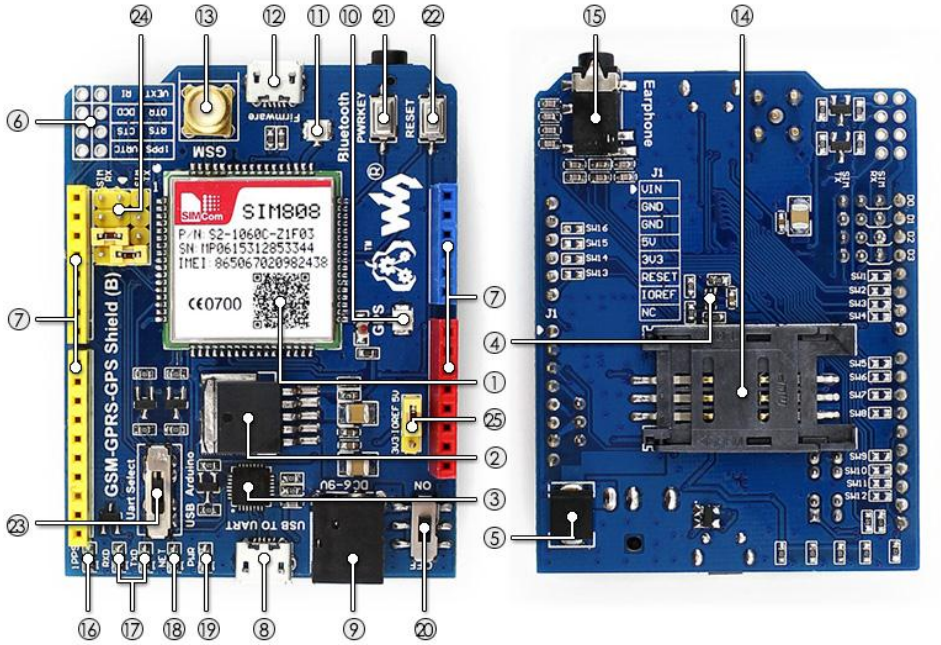

Figure 2.5: GSM/GPRS/GPS/Bluetooth SIM808 Shield (B) Board Description

1. SIM808 module
2. MIC29302 power chip
3. CP2102: USB TO UART converter

4. SMF05C: TVS diode
5. 1N5408: on board rectifier
6. SIM808 functional pins
7. Arduino expansion connector
8. USB TO UART interface
9. DC power jack
10. GPS antenna connector
11. Bluetooth antenna connector
12. Firmware upgrade interface
13. GSM antenna connector
14. SIM card slot
15. 3.5mm earphone/mic jack
16. GPS status indicator
17. CP2102 UART Tx/Rx indicator
18. NET indicator:
    a. flashes fast when the module starts up
    b. flashes slowly after GSM register succeed
19. Power indicator
20. Power switch
21. SIM808 control button: press the button and hold for 1s, to startup/shutdown the SIM808
22. Reset button
23. UART selection switch, select controlling the SIM808 via:
    a. CP2102
    b. UART pins of Arduino interface
24. SIM808 UART configuration:
    a. SIM_TX: SIM808 UART TX
    b. SIM_RX: SIM808 UART RX
25. IOREF power selection: configure the UART voltage level.

2.4.3 **IIC / I2C 1602 Blue Backlight LCD Display Module:**

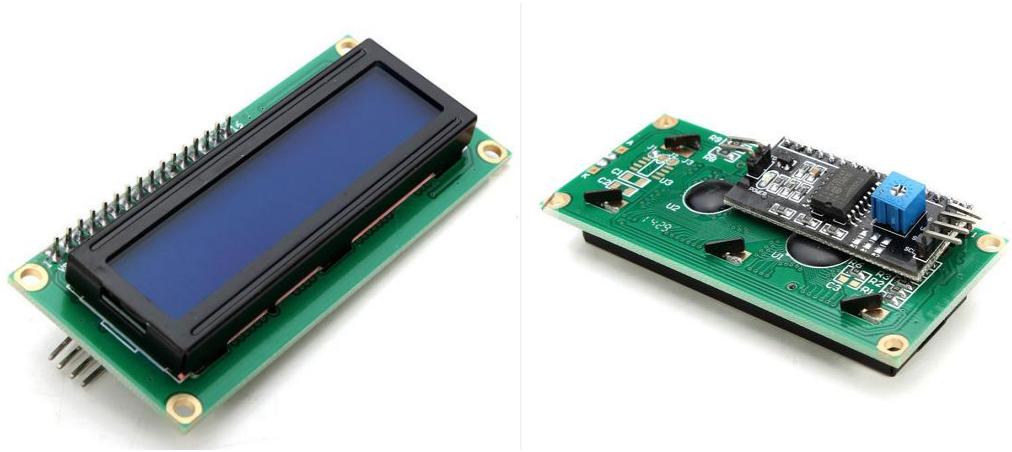

Figure 2.6: IIC / I2C 1602 Blue Backlight LCD Display Module[8]

- LCD display module with blue backlight.
- Wide viewing angle and high contrast.
- Built-in industry standard HD44780 equivalent LCD controller.
- Commonly used in: copiers, fax machines, laser printers, industrial test equipment, networking equipment such as routers and storage devices.
- LCM type: Characters
- Can display 2-lines X 16-characters.
- Voltage: 5V DC.
- Module dimension: 80mm x 35mm x 11mm.
- Viewing area size: 64.5mm x 16mm.

2.4.4 **SW-420 NC Type Vibration Sensor Module Vibration Switch**

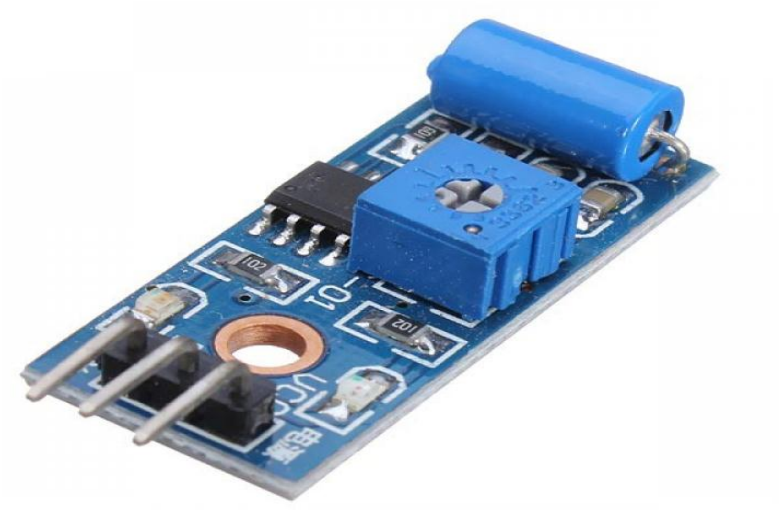

Figure 2.7: SW-420 NC Type Vibration Sensor Module

If the module does not vibrate, the vibration switch was closed on state, output of low output, the green indicator light. The product vibrates, vibration switch momentary disconnect, output is driven high, the green light does not shine. The output can be directly connected with the microcontroller, which to detect high and low level, so as to detect whether the environment exist vibration, play a role in the alarm [9].

- Using SW-420 normally closed type vibration sensor
- Comparator output, clean signal, good waveform, strong driving ability, >15mA
- Working voltage 3.3V ~ 5V
- Output format: digital switching output (0 and 1)
- Using a wide voltage LM393 comparator
- With bolt holes for easy installation
- Small PCB size: 3.2 x 1.4cm.

2.5 **Previous Related Works**

In many research IoT has been used in vehicle tracking system to find the vehicle location. In [10], the hardware and software of the GPS and GSM network were developed. The proposed GPS/GSM based System has the two sections, first is a mobile unit and another is controlling station. The system forms, interfaces, connections, data transmission and gathering of information among the mobile unit and control stations are working effectively. These outcomes are good with GPS technologies.

In [11], a vehicle tracking system is an electronic device, installed in a vehicle to enable the owner or a third party to track the vehicle's place. This paper proposed to design a vehicle tracking system that works using GPS and GSM technology. This system built based on embedded system, used for tracking and positioning of any vehicle by using Global Positioning System (GPS) and Global system for mobile communication (GSM). This design will continuously watch a moving Vehicle and report the status of the Vehicle on demand.

In [12], Face Detection System used to detect the face of the driver, and compare with the predefined face. The car owner is sleeping during the night time and someone theft the car. At that point Face Detection System obtains pictures by one little web camera, which is hidden easily in somewhere in the car. Face Detection System compared the obtained images with the stored images. If the images don't match, then the information sends to the owner through MMS. The owners get the pictures of the criminal in cell phone and follow the place through GPS. The place of the car and its speed displayed to the owner through SMS. The owner can recognize the thief images as well as the place of the car and can easily find out the hijacker’s image. This system applied in day-to-day life.

In [13], this system provided vehicle cabin safety, security based on embedded system by modifying the existing modules. This technique screens the level of the poisonous gasses, for example, CO, LPG and alcohol inside the vehicle provided alert information as alarm during the dangerous situations. The SMS sends to the authorized person through the GSM. In this technique, the IR Sensor used to identify the static obstacle in front of the vehicle and the vehicle stopped if any obstacle detected. This is avoiding accidents due to collision of vehicles with any static obstacles.

In [14], Kai-Tai Song and Chih-Chieh Yang have a designed and built on a real-time visual tracking system for vehicle safety applications. In this paper manufactured a novel element based vehicle-tracking algorithm, consequently identify and track a few moving articles, like cars and motorcycles, in front of the following vehicle. Joint with the concept of focus of expansion (FOE) and view investigation, the built system can fragment elements of moving articles from moving background and offer a collision word of warning on real-time. The proposed algorithm using a CMOS image sensor and NMOS embedded processor architecture. The constructed stand-alone visual tracking system validated in real road tests. The developed remain solitary visual following framework approved in genuine street tests. The results provided information of

collision warning in urban artery with speed about 60 km/hour both at night and day times.

In [15], the remote monitoring system based on SMS and GSM was implemented. In view of the total design of the system, the hardware and software designed. In this paper, the GSM network is a medium for transmitting the remote signal. This consists of two sections that are the checking focus and the remote observing station. The observing centers consists of a PC and correspondence module of GSM. The software monitoring center and the remote observing station executed by utilizing VB. The result of this demonstration shows that the system can watch and control the remote communication between the monitoring center and the remote checking station.

In [16], the proposed tracking system based on cloud computing infrastructure. The sensors are used to monitor the fuel level, driver conditions, and speed of the vehicle. All data exchanged to cloud server using GSM empowered device. All the vehicles equipped with GPS antenna to find the place. To stay away from the alcoholic and drive, the alcohol sensor introduced to monitor the driver status. The proposed technology significantly avoids the accident in highways.

CHAPTER 3: DESIGN AND METHODOLOGY

3.1 **Decisions of design specification and hardware**

The project is combined an accident detection mechanism with a VTS. Hardware used in this project are listed below:

1. Arduino UNO REV3
2. GSM, GPS, GSRM Shied (SIM808)
3. Vibration Sensor (sw420)
4. LCD Display

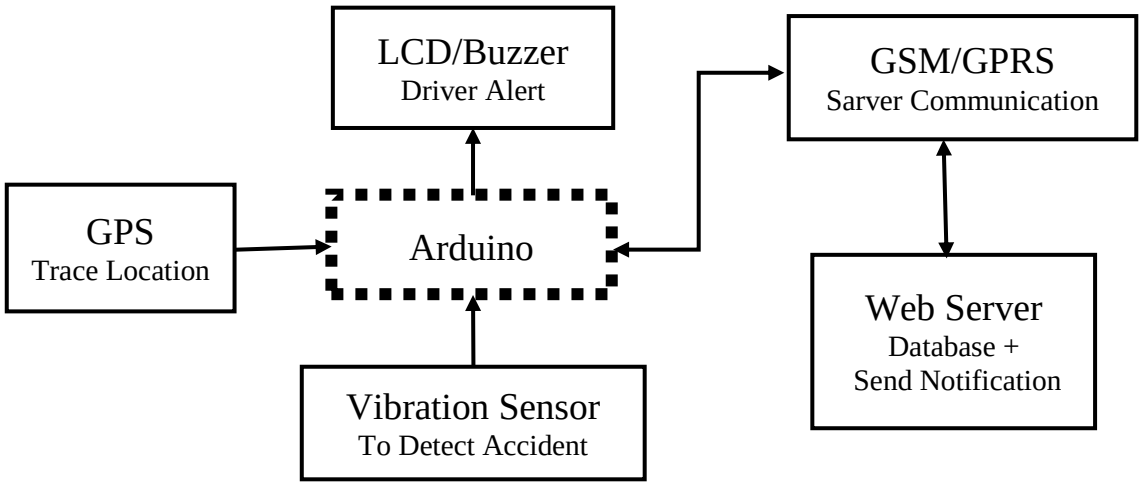

Figure 3.1: Block Diagram of the project

The above block diagram explains the working procedure of the system which can be designed for this project. An Arduino UNO is used here for automation and controlling of the other supporting devices those are GPS, GSM, vibration sensor, LCD display, puss switch, buzzer etc. Actually this paper gives a practical model of a vehicle accident detection and rescue information system which can do routing, tracking of moving vehicle as well as detect accident in large area.

Actually this system consists of two section, the first one is tracking location which is done by GPS in it and as the car moves the location of the car change continuously, the GPS finds the location in terms of two co-ordinates that are longitude and latitude. These two co-ordinates communicate with GSM modem which is shown in the block diagram.

The second one is detection of accident through vibration sensor. To detect accident, a threshold is set to a highest vibration value. If the vibration value is greater than the threshold value, then it will consider that accident occur and wait 60 second for a confirmation. After detection of an accident the system send the accident location to the web server. The web server then informs the car owner and nearest police station and hospital through web service using web application/ mobile application/ mobile SMS.

3.2 **Hardware connection, Program for hardware &web service**

A hardware and web service is designed to implement the project. The hardware will be installed in the vehicle and any user can monitor their vehicle by using the web service. Also any police station and hospital can get the notification of accident occur through the website.

3.2.1 Hardware Connection:

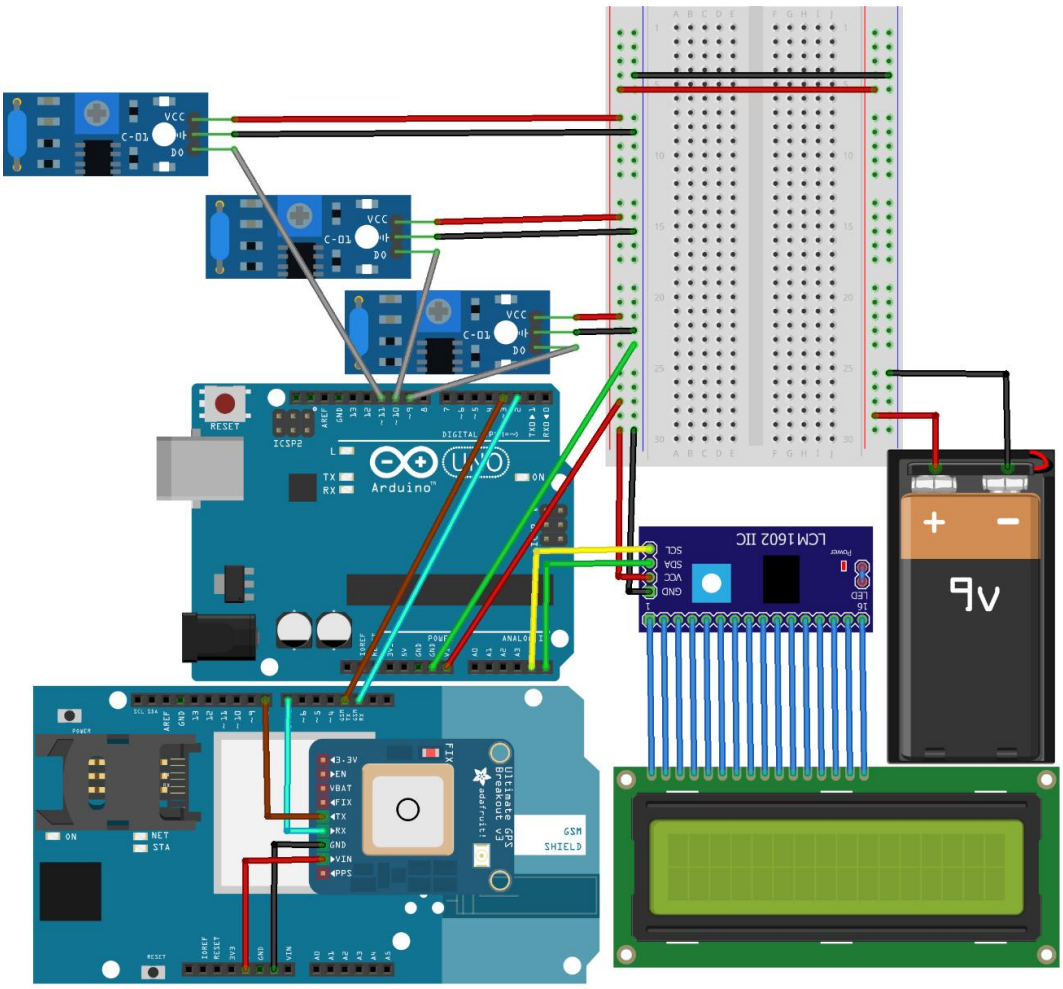

Figure 3.2: Block Diagram of the Hardware.

3.2.2 Web services:

To interact with users, a website has been developed where a user with the hardware can create an account and monitor all the vehicle installed this system. User will get notification if any vehicle gets into accident through the website account, mobile application and mobile SMS with the exact GPS location of accident. Also any police station and hospital can open an account from the website and will get notification through website and mobile SMS about an accident with the accident location and direction towards the accident location using google map.

3.3 **Algorithm for Accident Detection**

Flow chart for accident detection is given below:

Step 1: Start the program.

Step 2: Read vibration sensor data.

Step 3: If sensor value is more than limit go to step 4, otherwise go to step 2.

Step 4: Ask Driver for confirm accident. Set wait time 60 second.

Step 5: If Driver confirm accident go to step 6, otherwise go to step 7.

Step 6: Send notification to web server owner account, nearest police station, hospital also send SMS. Go to step 2.

Step 7: Decrement wait time each second.

Step 8: if wait time = 0 second go to step 9, otherwise go to step 5.

Step 9: go to step 6.

Step 10: Stop.

3.4 **Communication with web server and sending notification**

The system communicates with the web server through GPRS communication via a GSM, GPRS Shield. It will send the vehicle location’s latitude and longitude data to web server upon user request or after detection of accident. The web application will forward necessary notification to the nearest police station and hospital with the website, mobile application and mobile SMS.

3.5 **Calculation of shortest distance:**

If an accident occurs in any place this system can communicate with nearest police station and hospital. To find out the nearest police station and hospital 'haversine' approximation method is used.

It is a formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth's surface – giving an 'as-the-crow-flies' distance between the points.

Haversine formula:

$$a = sin^2(\Delta\varphi/2) + cos\,\varphi 1 \times cos\,\varphi 2 \times sin^2(\Delta\lambda/2)$$

$$c = 2 \times atan2(\sqrt{a}, \sqrt{(1-a)})$$

$$d = R \times c$$

Where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km);

note that angles need to be in radians to pass to trig functions.

The 'haversine' formula remains particularly well-conditioned for numerical computation even at small distances unlike calculations based on the spherical law of cosines. The '(re)versed sine' is $1-\cos\theta$, and the 'half-versed-sine' is $(1-\cos\theta)/2$ or $\sin^2(\theta/2)$ as used above. Once widely used by navigators, it was described by Roger Sinnott in Sky & Telescope magazine in 1984 ("Virtues of the Haversine"): Sinnott explained that the angular separation between Mizar and Alcor in Ursa Major – 0°11′49.69″ – could be accurately calculated on a TRS-80 using the haversine [17].

3.6 **Project flow chart:**

The complete design flow of the project can be depicted by Figure 3.33.

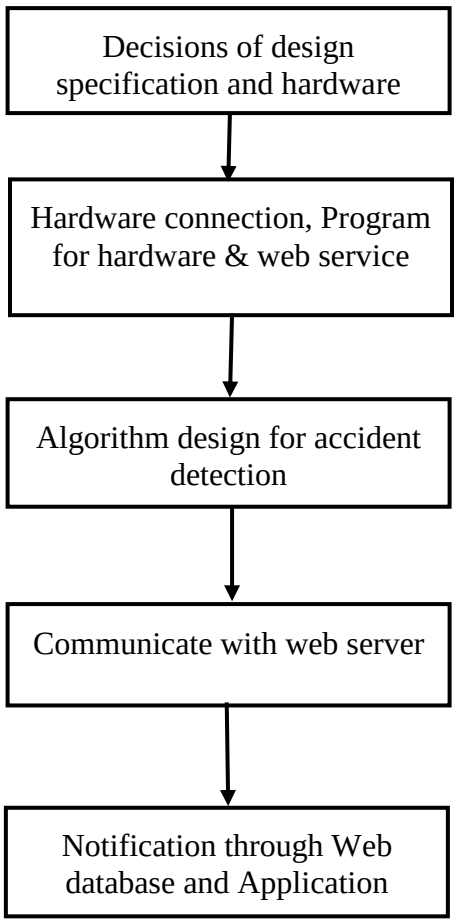

Figure 3.3: Flow Chart for the complete design methodology.

CHAPTER 4: IMPLEMENTATION

4.1 **Embedded device**

4.1.1 **Circuit connection:**

The project is developed by using GSM/GPRS/GPS/Bluetooth Shield SIM808 directly connected to all the pins of Arduino. Pin 2 of Arduino is used for RX and pin 3 for TX. 3 vibration sensors are connected with Arduino in pin number 8, 9 and 10. A LCD display is connected at pin number A5, A4. A confirmation switch is connected in pin 6, and a warning buzzer is added in pin 7.

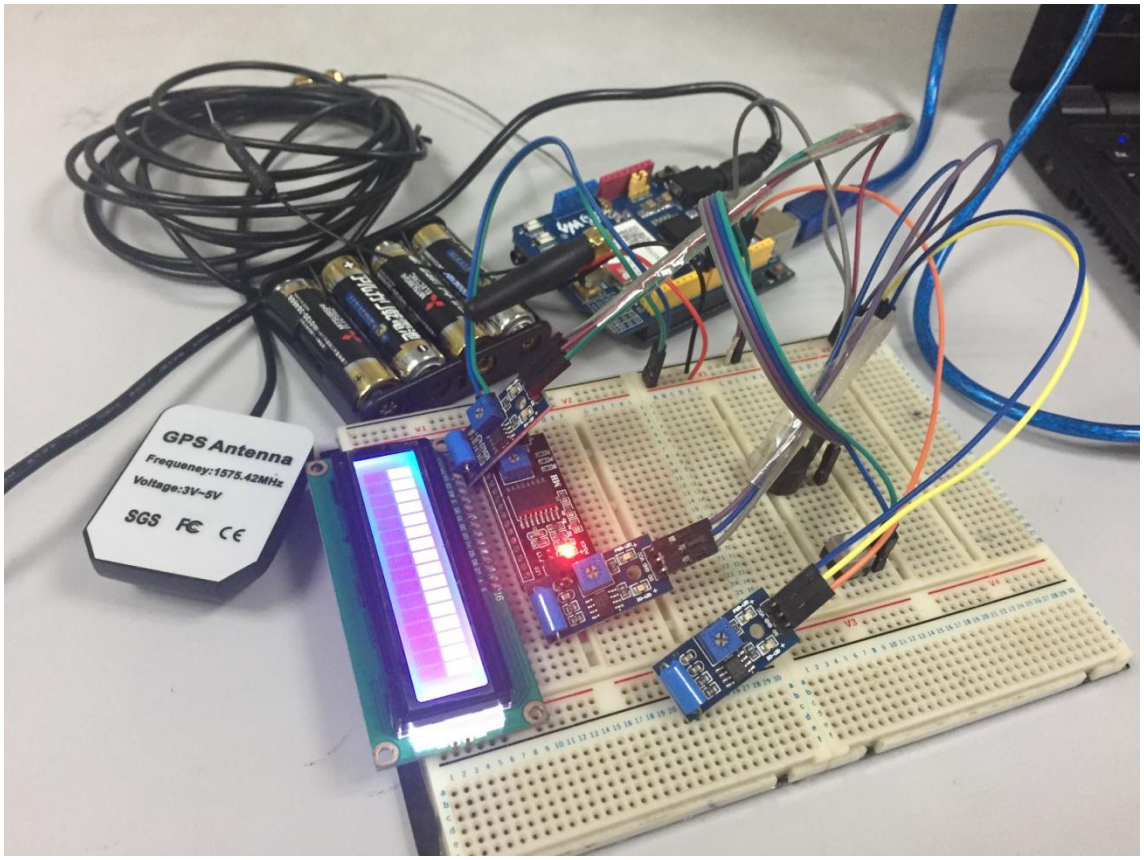

Figure 4.1: Actual Picture of the Embedded Device

4.1.2 **Arduino Programming:**

Arduino IDE is used to write program for Arduino Uno Board and to upload the program to the board.

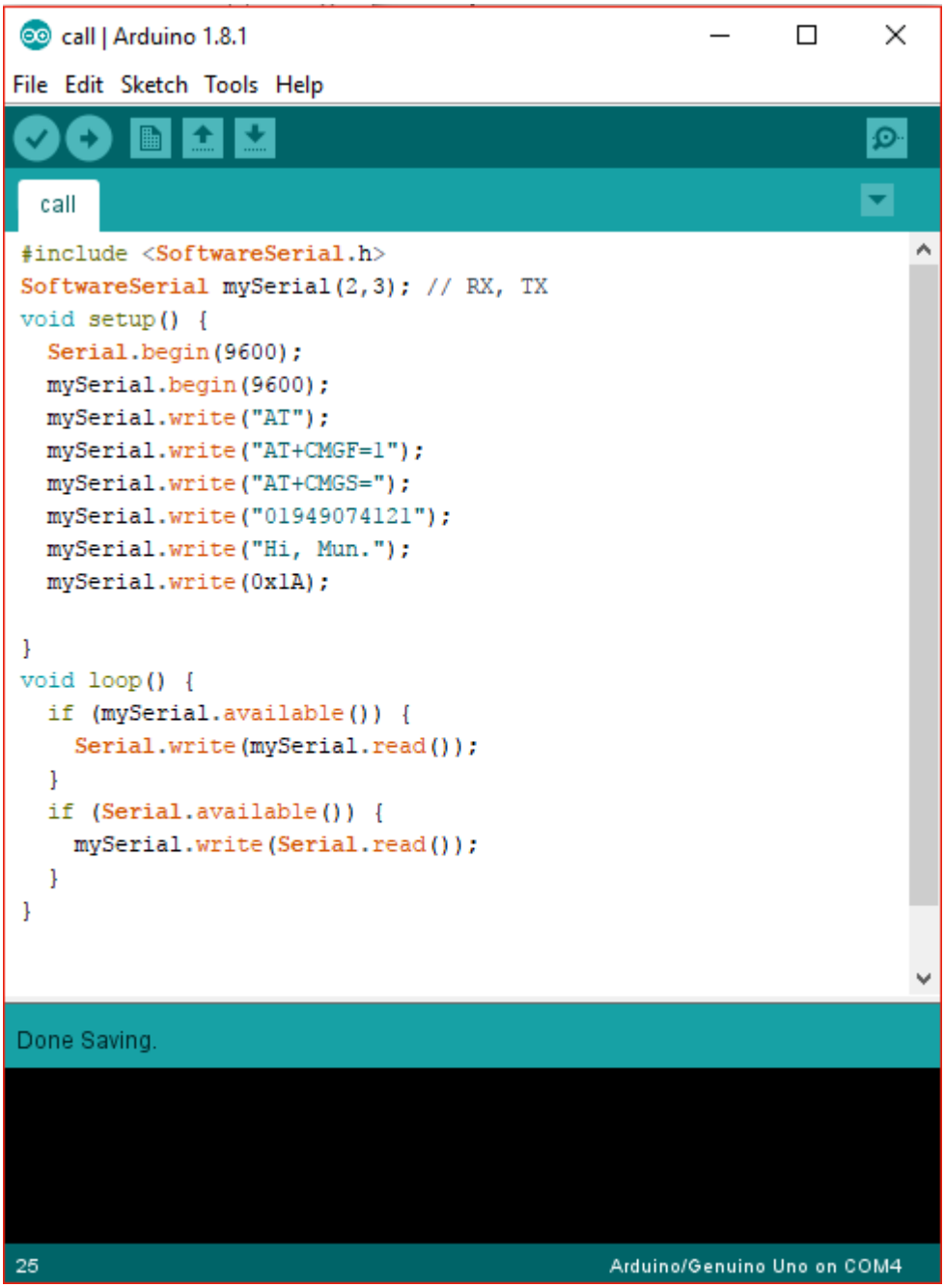

Figure 4.2: Arduino IDE & Program

4.2 **Configuration of Database:**

A web database is configured to store the data that are being sent from the embedded devices. phpMyAdmin application is used to configure the MySQL database. There are 7 tables:

1. user: Store user information.
2. policestation: Store Police Station information, location.
3. hospital: Store Hospital information, location.
4. vehicle: Store each device installed in different vehicle.
5. location: Store vehicle location on demand of user.
6. accident: Store location information after accident detection.
7. rescue: Store which police station and hospital is involving in rescue.

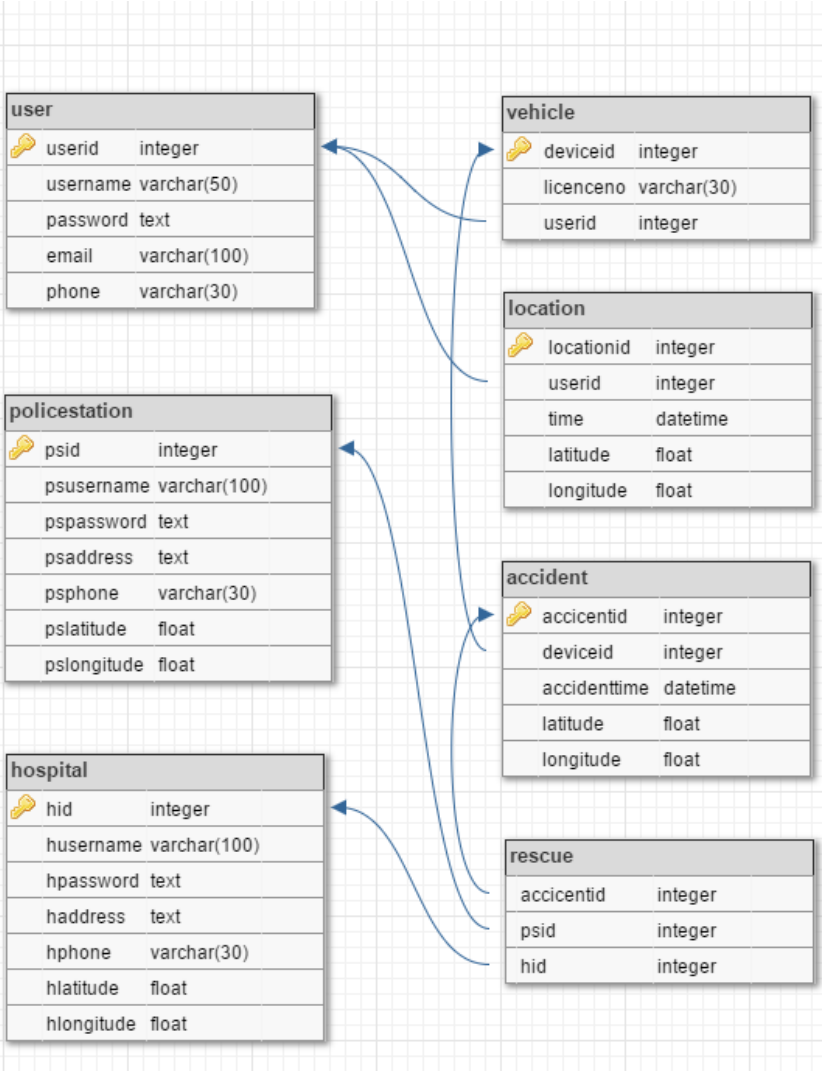

Figure 4.3: Schema Diagram of Database

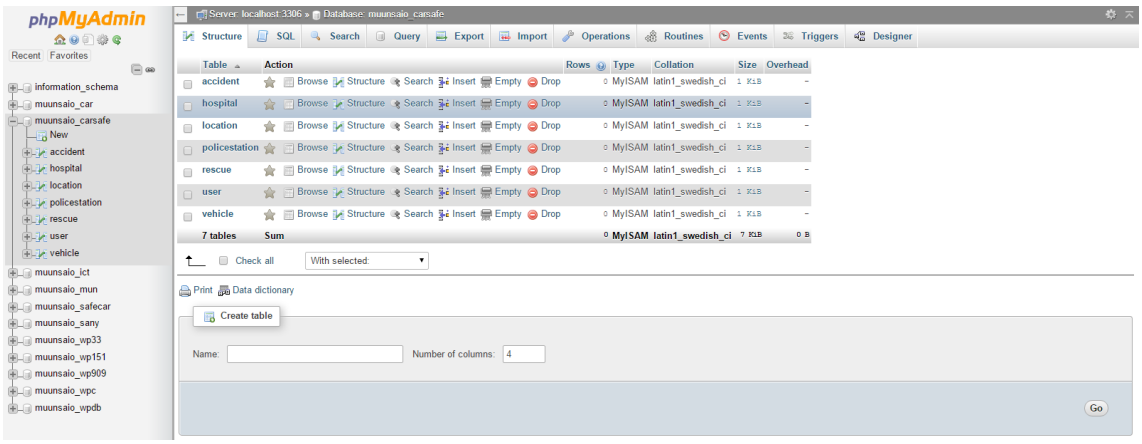

Figure 4.4: MySQL Database in webserver

4.3 **Web Application**

A web application has been developed to receive data from embedded device through GSM shield using GPRS connection. Using this application one user can monitor his/her vehicle. Using google API the web app can show the location of a vehicle. If an accident occurs, this system will notify the owner and show the accident spot as well as the nearest hospital or police station and a way of nearest police station and find three nearest hospital using google map. Also police station and hospital will get notification about the accident and get the location and direction towards the accident spot. All the notification will send by using website account, android mobile application and mobile SMS.

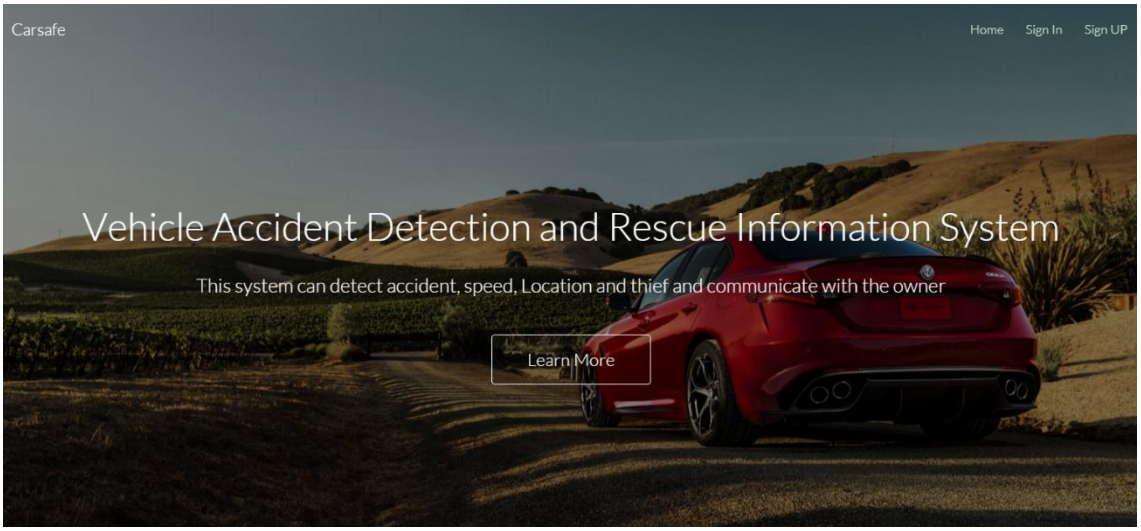

Figure 4.5: Website carsafe.xyz

Figure 4.6: Website Screenshot

Car Information

Device_Id	license No.	Date & Time	Longitude	Latitude	Map Show	Delete
1	y677	2017-07-30 00:10:40	90.425781	23.784145	Show Map	✖
2	bn677	2017-07-30 00:10:45	90.425919	23.783834	Show Map	✖
3	lop677	2017-07-30 00:10:13	90.441437	23.761362	Show Map	✖

Figure 4.7: User panel for multiple car installed the device. (website screenshot)

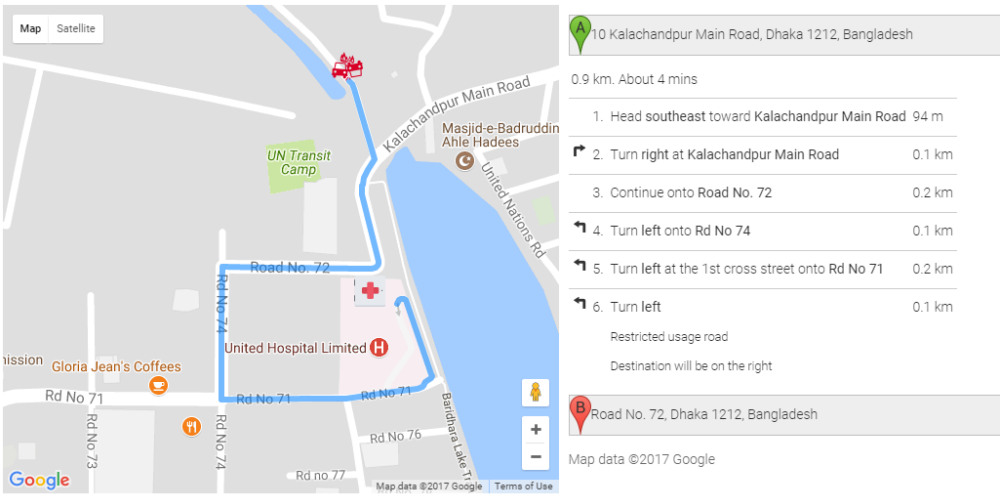

Figure 4.8: Hospital get notification about accident and direction towards the spot.

2nd and 3rd nearest hospital

Nearest serial	Hospital Name	latitude	Longitude	Map Show
2nd	Prescription Point	23.7831400	90.4259470	Show Map
3rd	Ibn Sina Diagnostic & Consultation Center	23.7831630	90.4257170	Show Map

Nearest Police station

Police Station Name	latitude	Longitude	Map Show
Badda Police Station	23.7713860	90.4273590	Show Map

Figure 4.9: Information about three nearest hospital from the accident spot.

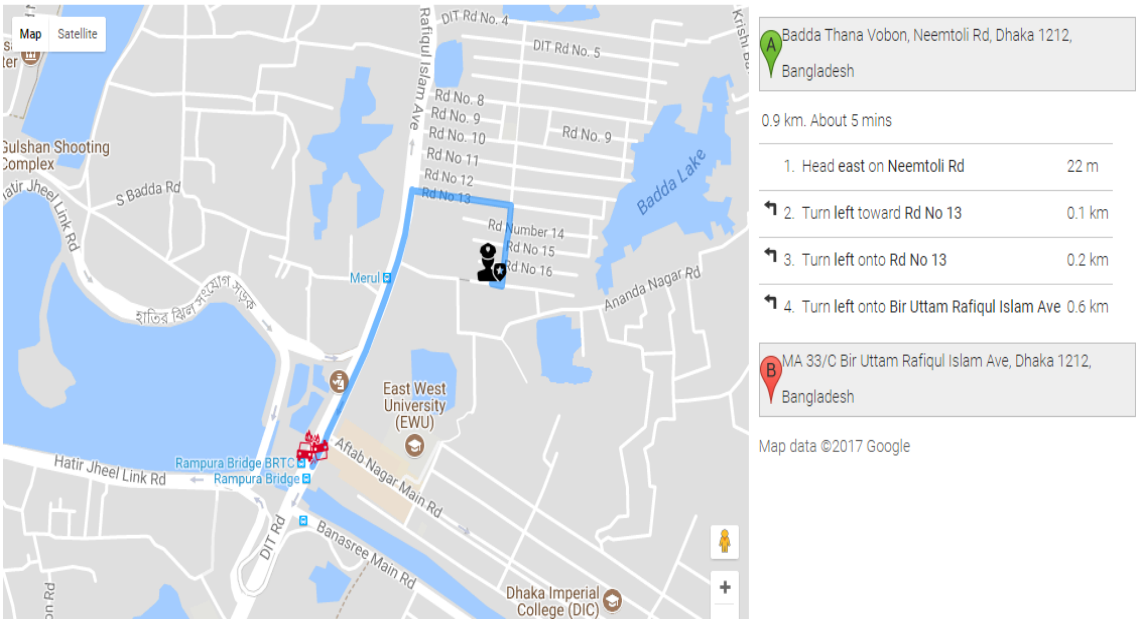

Figure 4.9: Police satation get notification about accident and direction towards the spot.

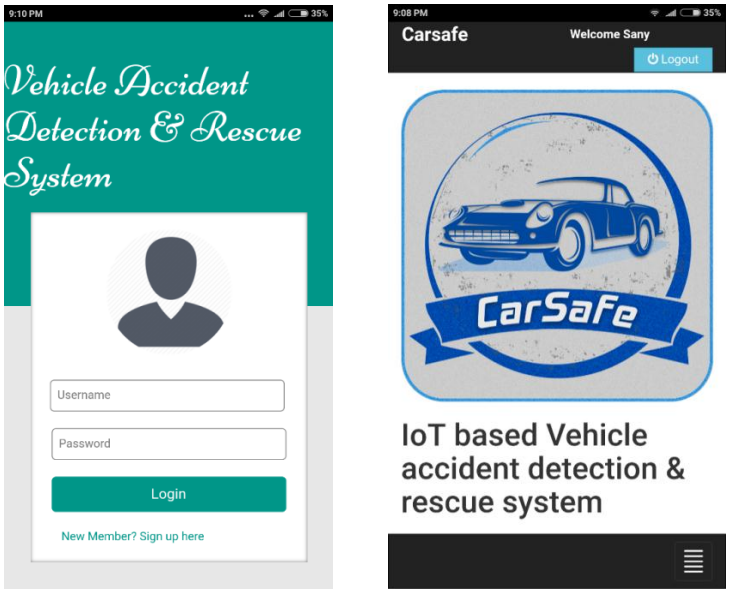

Figure 4.10: Android Application screenshot

4.4 **Mobile Application**

An android application has been developed for android mobile phones that receive data from the web server. This android application serves all the feature available in the website.

4.5 **Security**

4.5.1 **User Authentication for Login**

To access the website, the user has to go through a login page. There is a table “user” stored in the database with username and password. After the user input the username and password, it got matched with the fetched data from the database and if matches, it will log the user in and a new session will be started. This username also enables the data to load on the site. So, even if someone got access to the main webpage no data will be loaded until the correct username and password is given.

4.5.2 **Session Security**

A session can be defined as a server-side storage of information that is desired to persist throughout the user’s interaction with the web site or web application.

Instead of storing large and constantly changing information via cookies in the user’s browser, only a unique identifier is stored on the client side (called a ‘session id’). This session id is passed to the web server every time the browser makes an HTTP request.

The web application pairs this session id with its internal database and retrieves the stored variables for use by the requested page.

On the login page and home page of the website a session security has been added. After arriving on the login page, the user session starts by session_start().

It automatically creates a session id and puts it in the browser as a cookie. Then the login variables get stored in the session and got retrieve each time after cross checking the session id. If the user logs out, the session id will expire and a new one will be generated for each new login sessions.

CHAPTER 5: CONCLUSION

5.1 **Overall Conclusion**

The IoT Based Vehicle Accident detection and rescue system is successfully implemented using database server and API and fulfils all the requirements to be an IoT based framework. This device is capable of reading and collecting the required data and sends them securely to the database stored in server. This system can do tracking of a vehicle which have this device. Besides, if an accident occur this system can communicate nearest hospital and police station. Police station and hospital's authority can see the shortest route to reach the accident spot using this system which have a web application and mobile application. Web based real time data visualization makes this system more convenient to see all the data in a clean, formatted and user friendly way.

5.2 **Future Work**

So far, it has been implemented that the system can collect data successfully from sensors and communicate with web server. In near future, the system can be improved by using more sensor to detect accident with more accuracy and more different way of accident detection. This system will try to communicate at least three nearest hospitals if any major accident occurs and show the shortest path to reach the accident spot. Moreover, the system will integrate with other system. For example, an insurance company can use the project database to inquiry about an accident and provide money to the owner in time. In Bangladesh, traffic jam is very popular word. If people use this system, the system can collect traffic data and notify the driver about traffic and find out a way which have less traffic jam.

REFERENCE

[1] "A Brief History Of GPS Vehicle Tracking", Trackyourtruck.com, 2017. [Online]. Available: http://www.trackyourtruck.com/blog/brief-history-gps-vehicle-tracking/. [Accessed: 16- Jul- 2017]

[2] "Vehicle tracking system", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Vehicle_tracking_system. [Accessed: 16- Jul- 2017]

[3] S. Al-Mahmood, "Bangladesh's road accidents take heavy toll on poor – and on economy", the Guardian, 2017. [Online]. Available: https://www.theguardian.com/global-development/2012/aug/22/bangladesh-road-accidents-poor-economy. [Accessed: 15- Jul- 2017]

[4] "How to design an IoT-ready infrastructure: The 4-stage architecture", *TechBeacon*, 2017. [Online]. Available: https://techbeacon.com/4-stages-iot-architecture. [Accessed: 03- Aug- 2017].

[5] 2017. [Online]. Available: http://wso2.com/wso2_resources/wso2_whitepaper_a-reference-architecture-for-the-internet-of-things.pdf. [Accessed: 09- Aug- 2017]

[6] "Cite a Website - Cite This For Me", Arduino.org, 2017. [Online]. Available:http://www.arduino.org/products/boards/arduino-uno. [Accessed: 02- Aug- 2017]

[7] "GSM/GPRS/GPS Shield (B), Arduino Shield Based on SIM808", *Waveshare.com*, 2017. [Online]. Available: http://www.waveshare.com/gsm-gprs-gps-shield-b.htm. [Accessed: 02- Aug- 2017]

[8] 2017. [Online]. Available: https://opencircuit.nl/ProductInfo/1000061/I2C-LCD-interface.pdf. [Accessed: 02- Aug- 2017]

[9] S. Module, S. Arduino. and T. If the module does not vibrate, "SW420", *Bdspeedytech.com*, 2017. [Online]. Available: http://bdspeedytech.com/index.php?route=product/product&product_id=1388&search=sw+420. [Accessed: 02- Aug- 2017]

[10] Asaad M. J. Al-Hindawi, Ibraheem Talib,“Experimentally Evaluation of GPS/GSM Based System Design”, Journal of Electronic Systems Volume 2 Number 2 June 2012

[11] Kunal Maurya , Mandeep Singh, Neelu Jain, “ Real Time Vehicle Tracking System using GSM and GPS Technology- An Anti-theft Tracking System,” International Journal of Electronics and Computer Science Engineering. ISSN 2277-1956/V1N3-1103-1107

[12] Vikram Kulkarni & Viswaprakash Babu, “embedded smart car security system on face detection’, special issue of IJCCT, ISSN(Online):2231-0371, ISSN(Print):0975-7449,volume-3, issue-1

[13] V.Ramya, B. Palaniappan, K. Karthick, “Embedded Controller for Vehicle In-Front Obstacle Detection and Cabin Safety Alert System”, International Journal of Computer Science & Information Technology (IJCSIT) Vol.4, No 2, April 2012.

[14] Kai-Tai Song, Chih-Chieh Yang, of National Chiao Tung University, Taiwan, “Front Vehicle Tracking Using Scene Analysis”, Proceedings of the IEEE International Conference on Mechatronics & Automation 2005.

[15] Chen Peijiang, Jiang Xuehua, “Design and Implementation of Remote monitoring system based on GSM,” vol.42, pp.167-175. 2008.

[16] Albert Alexe, R.Ezhilarasie, “Cloud Computing Based Vehicle Tracking Information Systems”, ISSN: 2229 - 4333 (Print) | ISSN: 0976 – 8491 (Online) IJCST Vol. 2, Iss ue 1, March 2011

[17] w. Chris Veness, "Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript", *Movable-type.co.uk*, 2017. [Online]. Available: http://www.movable-type.co.uk/scripts/latlong.html. [Accessed: 02- Aug- 2017]

APPENDIX

Essential Source Code Of This Project

/**************** Arduino Programing ***************/**

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include<String.h>
#include <Wire.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#define warningTime 5
#define threshold 20000
#define buzzer A3
#define vibrationSensorLeft A0
#define vibrationSensorMiddel A1
#define vibrationSensorRight A2
char keypressed;
LiquidCrystal_I2C lcd(0x3F, 16, 2);

long lat,lon;
SoftwareSerial GPS(2,3);
TinyGPS gps;

const byte numRows= 4;        //number of rows on the keypad
const byte numCols= 4;        //number of columns on the keypad

char keymap[numRows][numCols]=
{
```

```
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
byte rowPins[numRows] = {11,10,9,8};        //Rows 0 to 3
byte colPins[numCols]= {7,6,5,4};           //Columns 0 to 3
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);

void setup() {
  Serial.begin(9600);
  GPS.begin(9600);
  pinMode(vibrationSensorLeft, INPUT);
  pinMode(vibrationSensorMiddel, INPUT);
  pinMode(vibrationSensorRight, INPUT);
  pinMode(buzzer, OUTPUT);
  lcd.begin();
  lcd.backlight();
  startup();
  GPS.println("AT+CGATT?");
  delay(500);
  GPS.println("AT+CSTT=\"internet\"");      //APN for Robi SIM
  delay(500);
  GPS.println("AT+CIICR");
  delay(500);
  GPS.println("AT+CIFSR");
  delay(500);
}
void loop() {
  //clocation();
  while(vibration()>threshold){
    accidentDetected();
    startup();
```

```
  }
  //clocation();
  //Serial.println("C Location");
}
void startup(){
  popup("::: CAR SAFE :::",3);
  lcd.setCursor(0, 1);
  lcd.print("   Save Lives  ");
}
void popup(String s,int times){
  int i;
  for(i=0;i<times;i++){
    lcd.print(s);
    delay(500);
    lcd.clear();
    delay(400);
  }
  lcd.print(s);
}
long vibration(){
  long vibrationSensorValueTotal = 0;
  long vibrationSensorValueLeft = pulseIn (vibrationSensorLeft, HIGH);
  long vibrationSensorValueMiddel = pulseIn (vibrationSensorMiddel, HIGH);
  long vibrationSensorValueRight = pulseIn (vibrationSensorRight, HIGH);
  vibrationSensorValueTotal = vibrationSensorValueLeft +
vibrationSensorValueMiddel + vibrationSensorValueRight;
  Serial.println(vibrationSensorValueTotal);
  delay(100);
  return vibrationSensorValueTotal;
}
void accidentDetected(){
  int flag=0;
  int i=warningTime,j;
  while(i>=0){
```

```
  lcd.clear();
  lcd.print(i);
  flag = keyInput(700);
  if(flag==1||flag==2){
    break;
  }
  analogWrite(buzzer,200);
  flag = keyInput(300);
  if(flag==1||flag==2){
    break;
  }
  analogWrite(buzzer,0);
  i--;
}
analogWrite(buzzer,0);
lcd.clear();
if(flag==0){
  lcd.print("Accident Occured");
  for(int i=0;i<6;i++){
    analogWrite(buzzer,225);
    delay(200);
    analogWrite(buzzer,0);
    delay(200);
  }
  location();
  delay(2000);
  lcd.clear();
  delay(500);
}
else if(flag==1){
  popup("No Accident.",5);
}
else if(flag==2){
```

```
    lcd.print("    Accident");
    lcd.setCursor(0, 1);
    lcd.print("   Confirmed");
    for(int i=0;i<6;i++){
      analogWrite(buzzer,225);
      delay(200);
      analogWrite(buzzer,0);
      delay(200);
    }
    location();
    delay(2000);
    lcd.clear();
    delay(500);
  }
}
int keyInput(int times){
  int flag=0,i=0;
  while(i<times){
    keypressed = myKeypad.getKey();
    if (keypressed == '1'){
      flag=1;
      break;
    }
    else if (keypressed == '2'){
      flag=2;
      break;
    }
    i++;
    delay(1);
    }
    return flag;
}
void location(){
```

```
 byte a;
 if(GPS.available()){
   a=GPS.read();
   while(gps.encode(a)){
     gps.get_position(&lat,&lon);
     message(lat,lon);
     Serial.println(lat);
     Serial.println(lon);
     for(int i=0;i<=10;i++){
       Senddata(lat,lon);
       Serial.println("Send");
     }
     return;
   }
 }
 location();
}
void clocation(){
 byte a;
 if(GPS.available()){
   a=GPS.read();
   while(gps.encode(a)){
     gps.get_position(&lat,&lon);
     Serial.println(lat);
     Serial.println(lon);
     Sendcdata(lat,lon);
     return;
   }
 }
 clocation();
}
void message(long lat, long lon){
```

```
  GPS.println("AT+CMGF=1");
  delay(500);
  GPS.println("AT+CMGS=\"01826989995\"");
  delay(500);
  GPS.println("Accident Occured!!\nMap Location:\n");
  delay(500);
  GPS.print("https://maps.google.com/maps?q=");
  GPS.print((lat*0.000001),6);
  delay(200);
  GPS.print(",");
  delay(200);
  GPS.print((lon*0.000001),6);
  delay(200);
  GPS.write(26);
  delay(1000);
}
void Senddata(long lat, long lon)
{
  char end_c[2];
  end_c[0]=0x1a;
  end_c[1]='\0';
  GPS.println("AT+CIFSR");
  delay(500);
  GPS.println("AT+CIPSPRT=0");
  delay(500);
  GPS.println("AT+CIPSTART=\"tcp\",\"carsafe.xyz\",\"80\"");
  delay(500);
  GPS.println("AT+CIPSEND");                    //begin send data to remote server
  delay(500);
  GPS.print("POST ");
  delay(500);
  GPS.print("/accident.php");
  delay(500);
```

```
  GPS.print(" HTTP/1.1\r\nHost: ");
  delay(500);
  GPS.print("carsafe.xyz");
  delay(500);
  GPS.print("\r\n");
  delay(500);
  GPS.print("User-Agent: Arduino\r\n");
  delay(500);
  GPS.print("Content-Type: application/x-www-form-urlencoded\r\n");
  delay(500);
  GPS.print("Content-Length: ");
  delay(500);
  GPS.print("50");

  delay(500);
  GPS.print("\r\n\r\n");
  delay(500);
  GPS.print("lat=");
  delay(500);
  GPS.print((lat*0.000001),6);
  delay(500);
  GPS.print("&lon=");
  delay(500);
  GPS.print((lon*0.000001),6);
  delay(500);
  GPS.print("&did=1234");
  delay(500);
  GPS.print("\r\n\r\n");
  delay(500);
  GPS.print(end_c);
  GPS.print((char)26);
  GPS.println();
}
```

```
void Sendcdata(long lat, long lon)
{
  char end_c[2];
  end_c[0]=0x1a;
  end_c[1]='\0';
  GPS.println("AT+CIFSR");
  delay(500);
  GPS.println("AT+CIPSPRT=0");
  delay(500);
  GPS.println("AT+CIPSTART=\"tcp\",\"carsafe.xyz\",\"80\"");
  delay(500);
  GPS.println("AT+CIPSEND");                    //begin send data to remote server
  delay(500);
  GPS.print("POST ");
  delay(500);
  GPS.print("/location.php");
  delay(500);
  GPS.print(" HTTP/1.1\r\nHost: ");
  delay(500);
  GPS.print("carsafe.xyz");
  delay(500);
  GPS.print("\r\n");
  delay(500);
  GPS.print("User-Agent: Arduino\r\n");
  delay(500);
  GPS.print("Content-Type: application/x-www-form-urlencoded\r\n");
  delay(500);
  GPS.print("Content-Length: ");
  delay(500);
  GPS.print("50");
  delay(500);
  GPS.print("\r\n\r\n");
```

```
  delay(500);
  GPS.print("lat=");
  delay(500);
  GPS.print((lat*0.000001),6);
  delay(500);
  GPS.print("&lon=");
  delay(500);
  GPS.print((lon*0.000001),6);
  delay(500);
  GPS.print("&did=1234");
  delay(500);
  GPS.print("\r\n\r\n");
  delay(500);
  GPS.print(end_c);
  GPS.print((char)26);
  GPS.println();
}
```

/**************** Code for website beckend***************/**

```
<?php
include("include/Database.php");
// user login
session_start();
if(isset($_POST['login'])){
$username=$_POST['username'];
$password=md5($_POST['password']);
// to prevent mysql injection
$username=stripcslashes($username);
$password=stripcslashes($password);
$username=mysqli_real_escape_string($connect,$username);
$password=mysqli_real_escape_string($connect,$password);
$query= "SELECT * FROM user_info WHERE username= '$username' AND
password= '$password'";
```

```
$run= mysqli_query($connect,$query) or die("Failed to query
Database".mysqli_error());
$row= mysqli_num_rows($run);


if($row==1){
$_SESSION["login_user"]=$username;
echo "<script>location='homepage.php'</script>";
}else{
echo "<script>alert('Failed to login')</script>";
echo "<script>location='login.php'</script>";
}}
// logout
if (isset($_GET["login_user"])) {
session_start();
session_destroy();
header("location: index.php");
}
// users car
if (isset($_POST["add"])) {
$license=$_POST["license"];
$username=$_SESSION["login_user"];
$query= "INSERT INTO user_cars VALUES(DEFAULT,'$username','$license')";
$run= mysqli_query($connect,$query);
if (!$run) {
echo "<script>alert('Failed to add')</script>";
echo "<script>location='user_cars.php'</script>";
}else{
echo "<script>alert('Successfully added')</script>";
echo "<script>location='user_cars.php'</script>";
}}
// delete routing data
if(isset($_GET['car_rou_id'])){
$id = $_GET['car_rou_id'];
$car_rou_del ="DELETE FROM cars_info WHERE id = '$id'";
```

```
$run= mysqli_query($connect,$car_rou_del);
 if(!$run){
 echo "<script>alert('Not Deleted')</script>";
 echo "<script>location = 'car_routing_information.php'</script>";
 } else{
  echo "<script>alert('Successfully Deleted..')</script>";
  echo "<script>location = 'car_routing_information.php'</script>";
}
}
// delete cars data
if(isset($_GET['car_id'])){
 $id = $_GET['car_id'];
$car_del ="DELETE FROM user_cars WHERE id = '$id'";
$run= mysqli_query($connect,$car_del);
if(!$run){
 echo "<script>alert('Not Deleted')</script>";
 echo "<script>location = 'your_cars.php'</script>";
 }else{
  echo "<script>alert('Successfully Deleted..')</script>";
  echo "<script>location = 'your_cars.php'</script>";
}
}
if(isset($_GET['license_no'])){
$license=$_GET['license_no'];
$sql="SELECT * FROM cars_info WHERE license_no='$license'";
$run=mysqli_query($connect,$sql);
while($row=$run->fetch_array()){
$lat=$row['latitude'];
$lng=$row['longitude'];
$qry="INSERT INTO show_map VALUES(DEFAULT,'$license','$lat','$lng')";
$result=mysqli_query($connect,$qry);
if(!$result){
echo "<script>Not inserted</script>";
```

```
}}}?>
```

/**Code for google map and finding nearest hospital and police station****/**

```
<?php
   include("header.php");
   include("include/Database.php");
   $qry="SELECT * FROM accident_info";
   $run= mysqli_query($connect,$qry);
   while($row=$run->fetch_array()){
      $lat1=$row["lat"];
      $lng1=$row["lon"];
      $devid=$row["deviceid"];
   }?>
<div class="col-md-12">
  <div class="well">
    <h2>Notifications</h2>
  </div>
</div>
   <marquee style="color: red; font-weight: bold; font-size: 20px; margin-bottom: 10px;">An accident occur</marquee>
<div class="col-sm-12 col-xs-12 col-md-12" >
  <div class=" col-sm-10 col-xs-10 col-md-10" id="direction_map" style="margin-bottom: 20px;">
  </div>
  <div class="col-sm-2 col-xs-3 col-md-2" id="right-panel">
  </div>
</div>
<?php
   $min=95999999;
   $min2=9599999;
   $min3=9599999;
```

```
$timezonedate= new DateTIme("now", new DateTimeZone('Asia/Dhaka'));
$datetime= $timezonedate-> format('Y-m-d h:i:sa');
$qry="SELECT * FROM hospital_information";
$run= mysqli_query($connect,$qry);
if(!$run){
   echo "1query not run";
}
while($row=$run->fetch_array()){
   $hpid=$row["id"];
   $lat2=$row["lat"];
   $lng2=$row["lng"];
   $dlat=($lat2-$lat1);
   $pi=3.141592654/180;
   $dmslat=$dlat*$pi;
   $dlon=($lng2-$lng1);
   $dmslon=$dlon*$pi;
   $dlat1=$lat1*$pi;
   $dlat2=$lat2*$pi;
   $a = sin($dlat/2) * sin($dlat/2) + sin($dlon/2)*sin($dlon/2)*
cos($dlat1)*cos($dlat2);
   $c = 2* atan2(sqrt($a),sqrt(1-$a));
   $distance= 6371*$c;

   // find minimum
   if($distance<$min){
      $min3=$min2;
      $min2=$min;
      $min=$distance;
      $minlat=$lat2;
      $minlon=$lng2;
      $minhpid=$hpid;
      }elseif($distance<$min2){
      $min3=$min2;
      $min2=$distance;
```

```
            $minlat2=$lat2;
            $minlon2=$lng2;
            $minhpid2=$hpid;
            }elseif($distance<$min3){
            $min3=$distance;
            $minlat3=$lat2;
            $minlon3=$lng2;
            $minhpid3=$hpid;
            }
    }
    $qr="SELECT name FROM hospital_information WHERE id= '$minhpid'";
    $re= mysqli_query($connect,$qr);
    if(!$re){
        echo "final query not run";
    }
    while($ro=$re->fetch_array()){
        $hpname= $ro['name'];
    }?>
<script>
    // map with text and  direction
    function initMap() {
        var location= new google.maps.LatLng(<?php echo $lat1; ?>,<?php echo $lng1;
?>);
        var directionsService = new google.maps.DirectionsService;
        // var directionsDisplay = new google.maps.DirectionsRenderer;
        var directionsDisplay = new google.maps.DirectionsRenderer({suppressMarkers:
true});  //to hide the default icons
        var map = new google.maps.Map(document.getElementById('direction_map'), {
          zoom: 9,
          center: location,
        });
        var accident = new google.maps.MarkerImage('images/car-accident32.png');
        var help = new google.maps.MarkerImage('images/medicine32.png');
            //you set your icon for each of the direction points Origin
```

```
        var marker1 = new google.maps.Marker({
            position: new google.maps.LatLng(<?php echo $lat1; ?>,<?php echo $lng1; ?>),
            map: map,
            icon: accident
         });
            //you set your icon for each of the direction points Destination,
        var marker2 = new google.maps.Marker({
            position: new google.maps.LatLng(<?php echo $minlat; ?>,<?php echo
$minlon; ?>),
            map: map,
            icon: help
        });


        directionsDisplay.setMap(map);
        directionsDisplay.setPanel(document.getElementById('right-panel'));
        calculateAndDisplayRoute(directionsService, directionsDisplay);
      }
      function calculateAndDisplayRoute(directionsService, directionsDisplay) {
        directionsService.route({
          origin: new google.maps.LatLng(<?php echo $lat1; ?>,<?php echo $lng1; ?>),
          destination: new google.maps.LatLng(<?php echo $minlat; ?>,<?php echo
$minlon; ?>),
          travelMode: google.maps.TravelMode.DRIVING
        }, function(response, status) {
          if (status === 'OK') {
            directionsDisplay.setDirections(response);
          } else {
            window.alert('Directions request failed due to ' + status);
          }
        });
      }
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDDEVXrd8GO2V3js17wnn
ldIu0sQLA3w7M&callback=initMap"></script>
```

```
<div class="col-md-12">
   <h2>2nd and 3rd nearest hospital</h2>
  <div class="table-responsive">
   <table class="table table-striped table-hover text-center">
     <tr class="info">
       <th class="text-center">Nearest serial</th>
       <th class="text-center">Hospital Name</th>
       <th class="text-center">latitude</th>
       <th class="text-center">Longitude</th>
       <th class="text-center">Map Show</th>
     </tr>
     <?php

       $ql="SELECT name FROM hospital_information WHERE id= '$minhpid2'";
       $res= mysqli_query($connect,$ql);
       if(!$res){
          echo "mquery not run!!";
       }
       while($r=$res->fetch_array()){
          $hpname2= $r["name"];
       ?>
         <tr class="danger">
          <td><?php echo "2nd" ?></td>
          <td><?php echo $hpname2 ?></td>
          <td><?php echo $minlat2 ?></td>
          <td><?php echo $minlon2 ?></td>
          <td><a href="2ndnearesthospitalmap.php?lat2=<?php echo $minlat2
?>&lon2=<?php echo $minlon2 ?>&lat1=<?php echo $lat1?>&lon1=<?php echo
$lng1?>"" class="btn btn-success no-rad">Show Map</a></td>
         </tr>
       <?php } ?>
       <?php
          $ql="SELECT name FROM hospital_information WHERE id= '$minhpid3'";
          $result= mysqli_query($connect,$ql);
```

```
            if(!$result){
               echo "<h3>There is no 3rd nearest hospital found !!!!!</h3>"."<br>";


            }
            while($r=$result->fetch_array()){
               $hpname3= $r['name'];
            ?>
            <tr class="danger">
            <td><?php echo "3rd" ?></td>
            <td><?php echo $hpname3 ?></td>
            <td><?php echo $minlat3 ?></td>
            <td><?php echo $minlon3 ?></td>
            <td><a href="3rdnearesthospitalmap.php?lat3=<?php echo $minlat3; ?>&lon3=<?php echo $minlon3; ?>&lat1=<?php echo $lat1; ?>&lon1=<?php echo $lng1; ?>"" class="btn btn-success no-rad">Show Map</a></td>
            </tr>
           <?php } ?>
    </table>
  </div>
</div>
<div class="col-md-12">
    <h2>Nearest Police station</h2>
  <div class="table-responsive">
    <table class="table table-striped table-hover text-center">
      <tr class="info">
        <th class="text-center">Police Station Name</th>
        <th class="text-center">latitude</th>
        <th class="text-center">Longitude</th>
        <th class="text-center">Map Show</th>
      </tr>
      <?php
        $psmin=999999;
        $qry="SELECT * FROM policestation_information";
        $run= mysqli_query($connect,$qry);
```

```
if(!$run){
   echo "ps query not run";
}

while($row=$run->fetch_array()){
   $psid=$row["id"];
   $lat2=$row["lat"];
   $lng2=$row["lng"];
   $dlat=($lat2-$lat1);
   $pi=3.141592654/180;
   $dmslat=$dlat*$pi;
   $dlon=($lng2-$lng1);
   $dmslon=$dlon*$pi;
   $dlat1=$lat1*$pi;
   $dlat2=$lat2*$pi;
   $a = sin($dlat/2) * sin($dlat/2) + sin($dlon/2)*sin($dlon/2)*
cos($dlat1)*cos($dlat2);
   $c = 2* atan2(sqrt($a),sqrt(1-$a));
   $distance= 6371*$c;

   // find minimum
   if($distance<$psmin){
      $psmin=$distance;
            $minpsid=$psid;
            $minpslat=$lat2;
               $minpslon=$lng2;
   }
}
// echo $minpslat.$minpslon;
$q="SELECT name FROM policestation_information WHERE id= $minpsid";
$res= mysqli_query($connect,$q);
if(!$res){
   echo "psm query not run";
}
```

```
        while($r=$res->fetch_array()){
           $psname= $r['name'];


        }
        ?>
          <tr class="danger">
            <td><?php echo $psname ?></td>
            <td><?php echo $minpslat ?></td>
            <td><?php echo $minpslon ?></td>
            <td><a href="nearestpolicestationmap.php?lat2=<?php echo $minpslat ?>&lon2=<?php echo $minpslon ?>&lat1=<?php echo $lat1?>&lon1=<?php echo $lng1?>&psname=<?php echo $psname; ?>"" class="btn btn-success no-rad">Show Map</a></td>
          </tr>
      </table>
    </div>
</div>


<?php
  $sqlqry= "INSERT INTO rescue_info VALUES (DEFAULT,'$devid','$lat1','$lng1','$minlat','$minlon','$hpname','$datetime')";
    $exec= mysqli_query($connect,$sqlqry) or die ("not insert in rescue information");
    mysqli_close($connect);
    include("footer.php");
?>
```