**DISASTER MANAGEMENT AND WOMEN SAFETY**

**BY**

**SRUTI VERMA-18BCE1005**

**G.NITHESH GURUDAAS-18BCE1101**

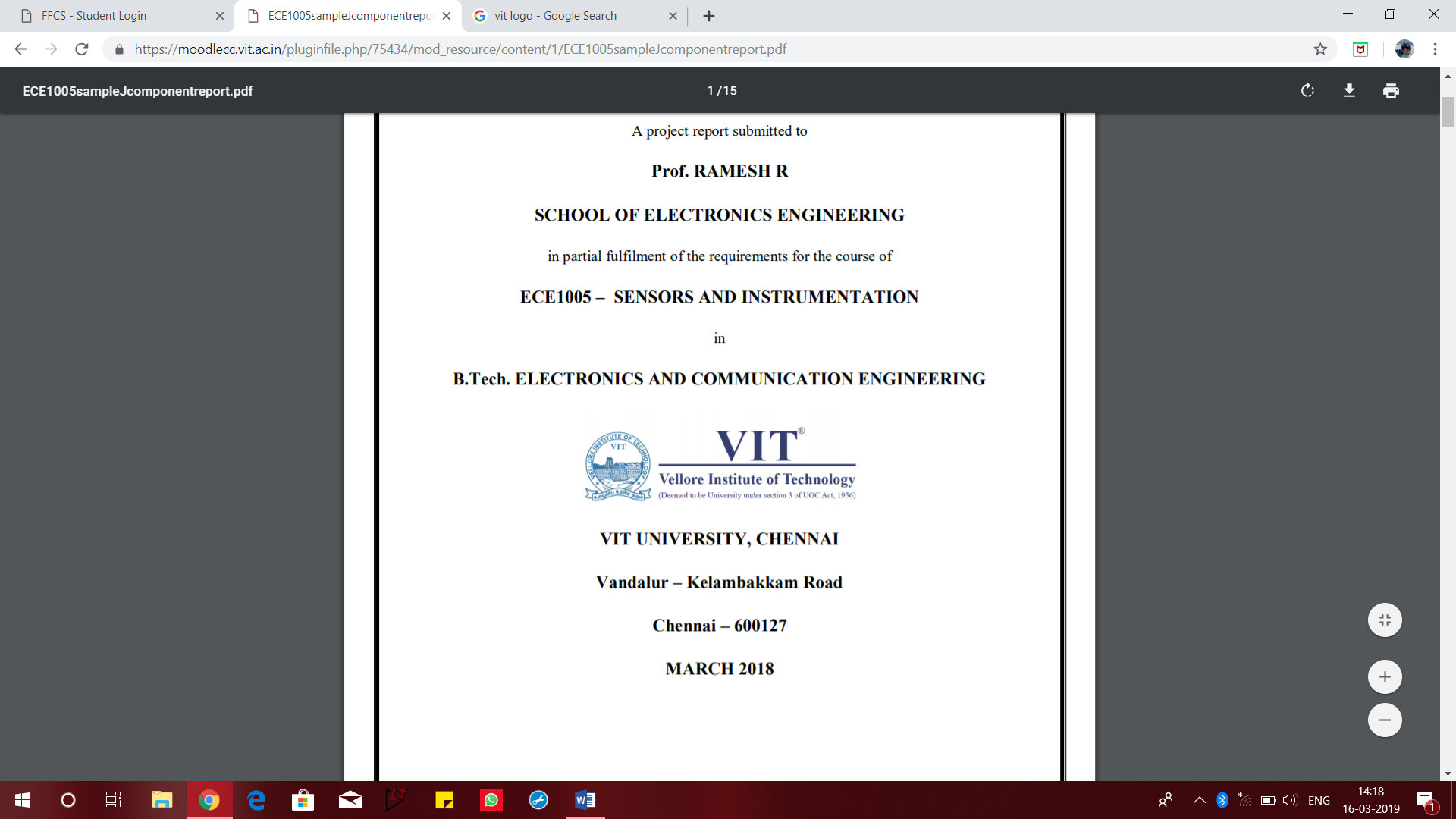
A project report submitted to

**Prof. Dr. D. Vydeki**

**SCHOOL OF ELECTRONICS ENGINEERING**

BY

**B.tech COMPUTER SCIENCE ENGINEERING**



**VIT UNIVERSITY, CHENNAI**

**VANDALUR KELAMBAKAM ROAD**

**CHENNAI-600127**

**MARCH-2020**

**ABSTRACT:**

Our project deals with ensuring women safety and helping people as soon as possible during such crimes or during disasters.

***We have used concepts of sensors to detect disasters such as flood and gas leak and with PIR sensors we can help disabled ML algorithm for ensuring women safety and detecting fires and INTERNET AND WEB DEVELOPMENT PROGRAMMING to develop a website which sends immediate notification to police along with crime or disaster location using google maps and we have a chatting forum for people to complain about crimes or ask for help during natural disaster.***

***Our website will also send mail to people about any disaster which has happened in their area.***

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No.** | **Title** | | **Page No.** |
|  | **Abstract** | | 2 |
| **1** | **Introduction** | | **3** |
| **2** | **Design/ Implementation** | | **4** |
|  |  | CODE | **5** |
|  |  | Economic Feasibility | **13** |
|  |  | Operational feasibility | **13** |
|  | 2.4 | Hardware Specification | **14** |
|  | 2.5 | Software Requirement | **14** |
| **3** | **Result and Analysis/Testing** | | **15** |
| **4** | **Conclusion** | | **17** |

# 

# 1 INTRODUCTION

It is a shame that rapes take place everyday. Rapes are present in every nook and corner of the world. It doesn’t differentiate between a 3-year-old kid and an 80-year-old lady. From parties to workplaces to our homes, rape and harassment have become a norm. The survivors of these heinous crimes are then left to be humiliated throughout their life.

*For predicting disasters and immediately informing to the police we will use the same camera installed for women safety which will also consist* ***a black box with all sensors fitted and the camera.****These sensors will* ***detect flood,natural gas leak,humidity to predict rain and accelerometer is used to detect earthquake.****The camera will c****atch images of fire and with our ML algorithm*** *we can again know that fire has occurred in the place and location of any disaster with help of sensors and ML will be sent to nearby police station through our* ***google map feature*** *in our website which also consists* ***portal for complaining or asking help to the police.Our website will also send mail to people about any disaster which has happened in their area.***

***Our hackathon model also deals with helping out women as soon as they are facing such heinous crimes.****Main idea is to* ***install cameras*** *almost on every street such that the camera will keep on* ***capturing videos and with our ML algorithm if anyone especially women are being harassed then immediately the camera captures images and depending upon the distance between the victims and accused people,ML algorithm will calculate if victims are harassed or not.****If harassed then the location information will immediately be sent to nearby police station through our* ***google map feature*** *in our website which also consists* ***portal for complaining or asking help to the police.***

# 2 DESIGN/IMPLEMENTATION

**2.1Modules:**

Our project consists of the following modules:

1. Website about disaster info, such as location, distress chat
2. Sensors and connection with IoT
3. ML for recognising harassment or forest fires
4. **Website:**

The website consists of a home page, which has some of the emergency contacts, such as police, ambulance, fire service, and has a map that shows where disaster has occurred. In the map, areas marked with a red circle imply that a disaster has occurred there. This will get updated periodically, based on the input from the sensors that we have placed. We can also sign up for notifications from this page. Based on location info from the device, we can notify the people that they are in the disaster zone or close to the disaster zone. There is also a distress chat option, which redirects you to a forum. In this forum anyone can post messages to ask for help or offer help. For example, if someone is stuck in some debris or some forest fire has happened, then he/she can send a distress message in the forum, saying that in this location such a disaster has happened, and I request help, and other people who are willing will help the concerned person. There is also a distress signal, which people who are in dire need of help can use. Upon clicking this option, the website will retrieve location data from the device and send an email to the authorities saying send help to this location.

The website was made using HTML, CSS, JavaScript, and PHP. Chat app only predominantly used PHP; others mostly used HTML, JavaScript and CSS. JavaScript functions were used to retrieve information on location data, and to display the map. Distress signal actually used python functions to send the signal to the authorities.

1. **Sensors and IoT connection:**

The black box consists of the following sensors:

1. Arduino
2. Gas sensor
3. Ultrasonic water level sensor
4. Camera module integrated with AI algorithms
5. Humidity sensor
6. PIR Sensor
7. Accelerometer

Arduino acts as the central hub for all connections and is interfaced with a laptop, and is connected to the internet. We use IoT to upload all the data related to the sensors onto a MQTT Server. Gas sensor is used to detect the presence of harmful gases. Due to time and budget restrictions, we used a MQ4 Methane gas sensor; but with a higher budget and longer time limits, we can incorporate a costlier, better version which can detect almost all harmful gases. We have fitted an ultrasonic water level sensor to a small container which will collect rain water. Once the water in the rain water container reaches a certain level within a specified time period, it will send an alarm notifying the possibility of a flood. The water in the container has to be emptied every day manually. The humidity sensor is to improve the prediction of weather. The camera module is used to detect forest fires, and harassment. The details about this module are explained further in the next part. An accelerometer is also fitted inside to detect earthquakes. There will always be some amount of seismic activity or some other factor, such as gusts of wind which make it move, so there may be faint vibrations, so we have incorporated a threshold for it, above which the accelerometer will notify the possibility of an earthquake.

Everything is connected via the Arduino to the IoT, and we can see the output of every action in the MQTT server. For example, if we shake the box too hard, then it will display a message in the MQTT server that earthquake has been sensed, or if we pour water into the container, it will say water level is high, possibility of floods.

1. **Recognising harassment and forest fire:**

The third module of our project is the camera module integrated with ML recognition patterns. It consists of a simple web camera which gives input to a Python server in the laptop. A ML recognition program is used here, and we made it be able to recognise fires and upon showing images or video of fires, the software was able to recognize that it was a fire and sent a message to the MQTT server saying that fire detected. We also made Python send an e-mail to one of our accounts when it found fire. To detect women harassment, we came up with the idea to check if two people are too close and any one of the persons shows emotions of discomfort or anything. But due to time restrictions, we had to scale down as the amount of Machine learning needed for it to understand reactions was very high, so we settled for detecting if two people are standing or sitting very close together. We used face recognition to detect people, and we checked the distance between the two faces. If the amount of distance between them was less than a threshold value, then it would display harassment on the MQTT server.

**Code:**

Forest fire IoT + ML code:

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <Servo.h>

#define LED 16

#define LED1 2

Servo servo;

Servo servoupper;

//Enter your wifi credentials

const char\* ssid = "IOT";

const char\* password = "90trupen";

//Enter your mqtt server configurations

const char\* mqttServer = "m15.cloudmqtt.com"; //Enter Your mqttServer address

const int mqttPort = 1679; //Port number

const char\* mqttUser = "omnrio"; //User

const char\* mqttPassword ="oFNgmBDXP"; //Password

WiFiClient espClient;

PubSubClient client(espClient);

void setup() {

delay(1000);

pinMode(LED,OUTPUT);

pinMode(LED1,OUTPUT);

servo.attach(2); //D4

servoupper.attach(0); // D3

servo.write(0);

servoupper.write(0);

delay(2000);

Serial.begin(115200);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.println("Connecting to WiFi..");

}

Serial.print("Connected to WiFi :");

Serial.println(WiFi.SSID());

client.setServer(mqttServer, mqttPort);

client.setCallback(MQTTcallback);

while (!client.connected()) {

Serial.println("Connecting to MQTT...");

if (client.connect("ESP8266", mqttUser, mqttPassword )) {

Serial.println("connected");

} else {

Serial.print("failed with state ");

Serial.println(client.state()); //If you get state 5: mismatch in configuration

delay(2000);

}

}

client.subscribe("python/test");

}

void MQTTcallback(char\* topic, byte\* payload, unsigned int length) {

Serial.print("Message arrived in topic: ");

Serial.println(topic);

Serial.print("Message:");

String message;

for (int i = 0; i < length; i++) {

message = message + (char)payload[i]; //Conver \*byte to String

}

Serial.print(message); // Frame no 1

if(message == "1") {

servo.write(30);

servoupper.write(0);

delay(2000);

servoupper.write(0);

servo.write(0);

delay(500);

}

Serial.print(message); // Frame no 2

if(message == "2") {

servo.write(60);

servoupper.write(0);

delay(2000);

servoupper.write(0);

servo.write(0);

delay(500);

}

Serial.print(message); // Frame no 3

if(message == "3") {

servo.write(90);

servoupper.write(0);

delay(5000);

servoupper.write(0);

servo.write(0);

delay(500);

}

Serial.print(message); // Frame no 4

if(message == "4") {

servo.write(30);

servoupper.write(45);

delay(2000);

servoupper.write(0);

servo.write(0);

delay(500);

}

Serial.print(message); // Frame no 5

if(message == "5") {

servo.write(60);

servoupper.write(45);

delay(2000);

servoupper.write(0);

servo.write(0);

delay(500);

}

Serial.print(message); // Frame no 6

if(message == "6") {

servo.write(90);

servoupper.write(45);

delay(2000);

servoupper.write(0);

servo.write(0);

delay(500);

}

Serial.println();

Serial.println("-----------------------");

}

void loop() {

client.loop();

}

ML Code:

import paho.mqtt.client as mqttClient

import paho.mqtt.client as mqttClient

import numpy as np

import cv2

import time

flag=0

'''def on\_connect(client, userdata, flags, rc):

if rc == 0:

print("Connected to broker")

global Connected #Use global variable

Connected = True #Signal connection

else:

print("Connection failed")

broker\_address= "soldier.cloudmqtt.com"

port = 10094

user = "hplnqnys"

password = "JPTXsA\_ocNVY"

client = mqttClient.Client("Python") #create new instance

client.username\_pw\_set(user, password=password) #set username and password

client.on\_connect= on\_connect #attach function to callback

client.connect(broker\_address, port=port) #connect to broker

client.loop\_start()'''

fire\_cascade = cv2.CascadeClassifier("C:\\Users\\Sivaraman\\AppData\\Local\\Programs\\Python\\Python37-32\\Scripts\\haarcascade\_frontalface\_default.xml")

cap = cv2.VideoCapture(0)

while True:

ret, img = cap.read()

gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

fire = fire\_cascade.detectMultiScale(gray, 1.3, 5)

print(len(fire))

h1=fire[0:1]

h2=fire[1:]

cv2.rectangle(img,(0,0),(128,480),(0,255,255),2)

cv2.rectangle(img,(128,0),(256,480),(0,255,255),2)

cv2.rectangle(img,(256,0),(382,480),(0,255,255),2)

cv2.rectangle(img,(382,0),(510,480),(0,255,255),2)

cv2.rectangle(img,(510,0),(640,480),(0,255,255),2)

for (x1,y1,w1,h1) in h1:

cv2.rectangle(img,(x1,y1),(x1+w1,y1+h1),(0,0,255),2)

roi\_gray = gray[y1:y1+h1, x1:x1+w1]

roi\_color = img[y1:y1+h1, x1:x1+w1]

print(x1)

for (x2,y2,w2,h2) in h2:

cv2.rectangle(img,(x2,y2),(x2+w2,y2+h2),(0,0,255),2)

roi\_gray = gray[y2:y2+h2, x2:x2+w2]

roi\_color = img[y2:y2+h2, x2:x2+w2]

print(x2)

if abs((x2+w2)-(x1+w1))<128:

print('Harass')

value1 = "Harass"

client.publish("python/test",value1)

elif (x+w)< 320 and (y+h)>240:

print('Quad-3')

elif (x+w)>320 and (y+h)<240:

print('Quad-2')

elif (x+w)>320 and (y+h)>240:

print('Quad-4')

cv2.imshow('img',img)

k = cv2.waitKey(1)

if k == ord('s'):

break

print("frame over")

cap.release()

cv2.destroyAllWindows()

Ultrasonic Sensor code:

int const trigPin = 10;

int const echoPin = 9;

int const buzzPin = 2;

void setup()

{

pinMode(trigPin, OUTPUT); // trig pin will have pulses output

pinMode(echoPin, INPUT); // echo pin should be input to get pulse width

pinMode(buzzPin, OUTPUT); // buzz pin is output to control buzzering

}

void loop()

{

// Duration will be the input pulse width and distance will be the distance to the obstacle in centimeters

int duration, distance;

// Output pulse with 1ms width on trigPin

digitalWrite(trigPin, HIGH);

delay(1);

digitalWrite(trigPin, LOW);

// Measure the pulse input in echo pin

duration = pulseIn(echoPin, HIGH);

// Distance is half the duration devided by 29.1 (from datasheet)

distance = (duration/2) / 29.1;

// if distance less than 0.5 meter and more than 0 (0 or less means over range)

if (distance <= 50 && distance >= 0) {

// Buzz

digitalWrite(buzzPin, HIGH);

} else {

// Don't buzz

digitalWrite(buzzPin, LOW);

}

// Waiting 60 ms won't hurt any one

delay(60);

}

**2.2 ECONOMIC FEASIBILITY**

**Each black box with all sensors,arduino and camera will cost about 1000 rupees but if we want to use more advanced and powerful sensors then cost is 5000 rupees.Therefore our model is cheap and many black boxes can be installed in various streets.**

**2.3 OPERATIONAL FEASIBILITY**

**Easy to operate and does not require much maintenance.Once code is written and integrated with sensors and cameras,model will work efficiently.**

**2.4 HARDWARE SPECIFICATION**

A **humidity sensor** is a device that detects and measures water vapor. ... Based on our robust capacitive technology, these **humidity sensors** provide accurate measurement of dew point and absolute **humidity** by combining relative **humidity** (RH) and temperature (T) measurements.**Here it is used for predicting rain.**

**Arduino** is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. **Arduino is main cpu which connects with IOT and processes all data.**

**Gas sensors (also known as gas detectors)** are electronic devices that detect and identify different types of gasses. They are commonly used to detect toxic or explosive gasses and measure gas concentration.

**PIR sensors** allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out.

**Camera module integrated with AI modules.**

**Sensors for water level flood detection.**

An **accelerometer** is an electromechanical device used to measure acceleration forces. Such forces may be static, like the continuous force of gravity or, as is the case with many mobile devices, dynamic to sense movement or vibrations. Acceleration is the measurement of the change in velocity, or speed divided by time.**Used here for detection of earthquake.**

**2.5 SOFTWARE SPECIFICATION**

**XAMPP** is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

**WE HAVE USED XAMPP FOR RUNNING OUR WEBSITE.**

**Notepad** is a simple text editor for Microsoft Windows and a basic text-editing program which enables computer users to create documents. It was first released as a mouse-based MS-DOS program in 1983, and has been included in all versions of Microsoft Windows since Windows 1.0 in 1985.

**WE HAVE USED NOTEPAD FOR WRITING HTML, CSS, JS, AND SUBLIME TEXT EDITOR FOR PHP CODE.**

**TensorFlow** is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for **machine learning applications** such as neural networks.

**MQTT** is an open OASIS and ISO standard lightweight, publish-subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP; however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT.

**2.6OVERVIEW OF LANGUAGES USED**

**Hypertext Markup Language (HTML)** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

**Cascading Style Sheets (CSS)** is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

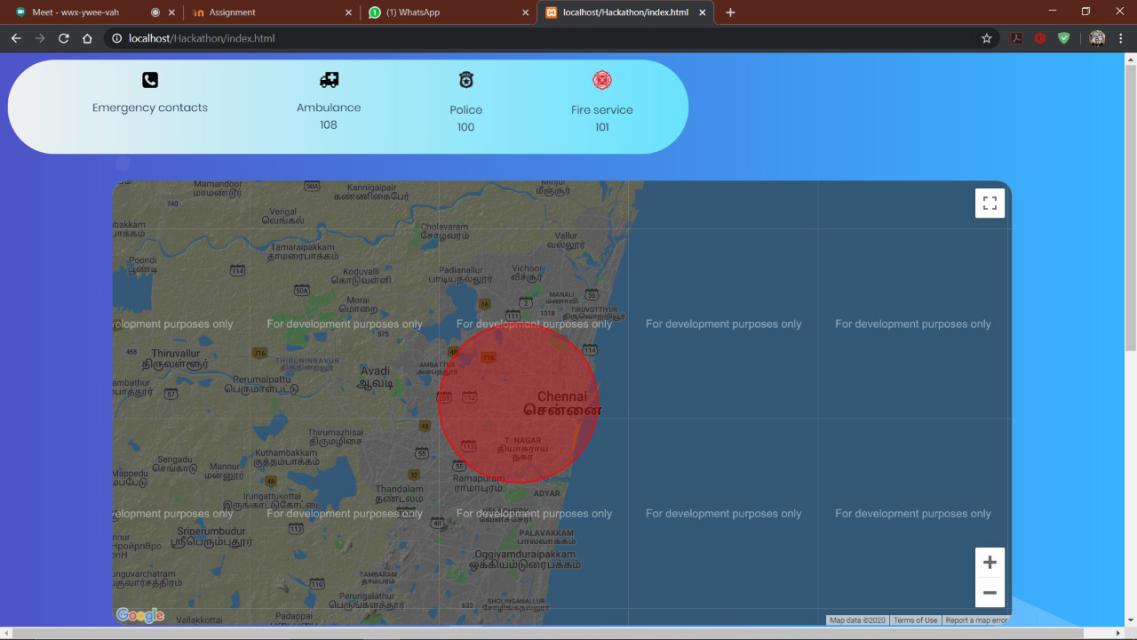
**JavaScript, often abbreviated as JS,** is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

**jQuery** is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License.

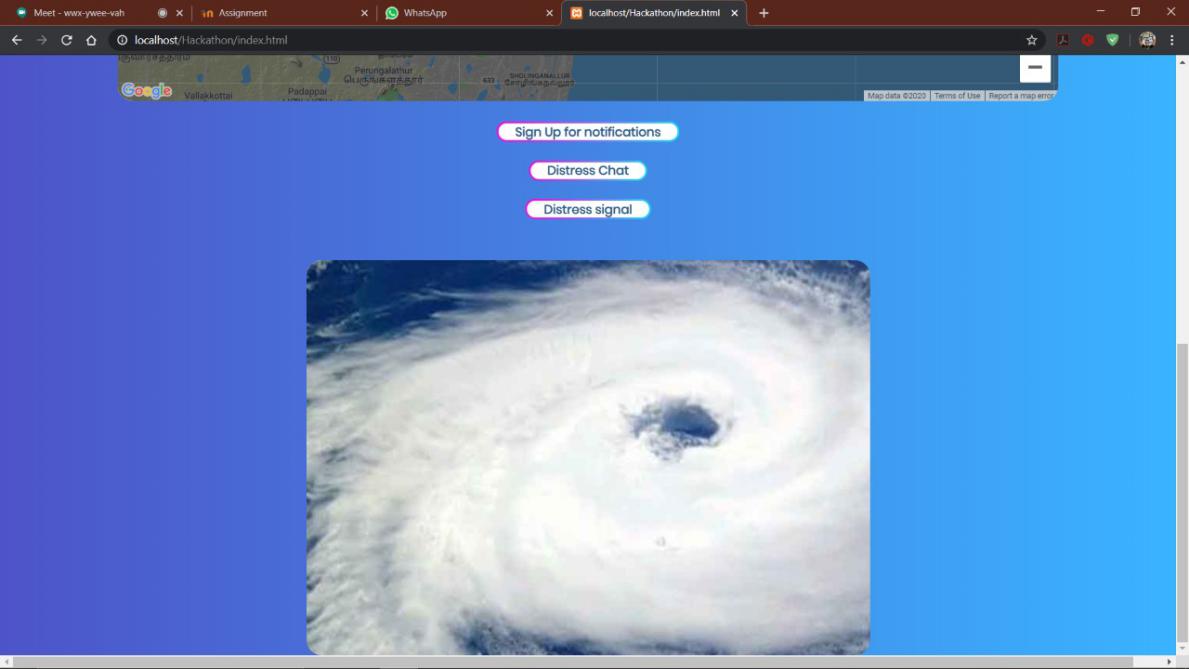
**PHP** is a popular general-purpose scripting language that is especially suited to web development for **back end.** It was originally created by Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group.

**3 RESULTS**

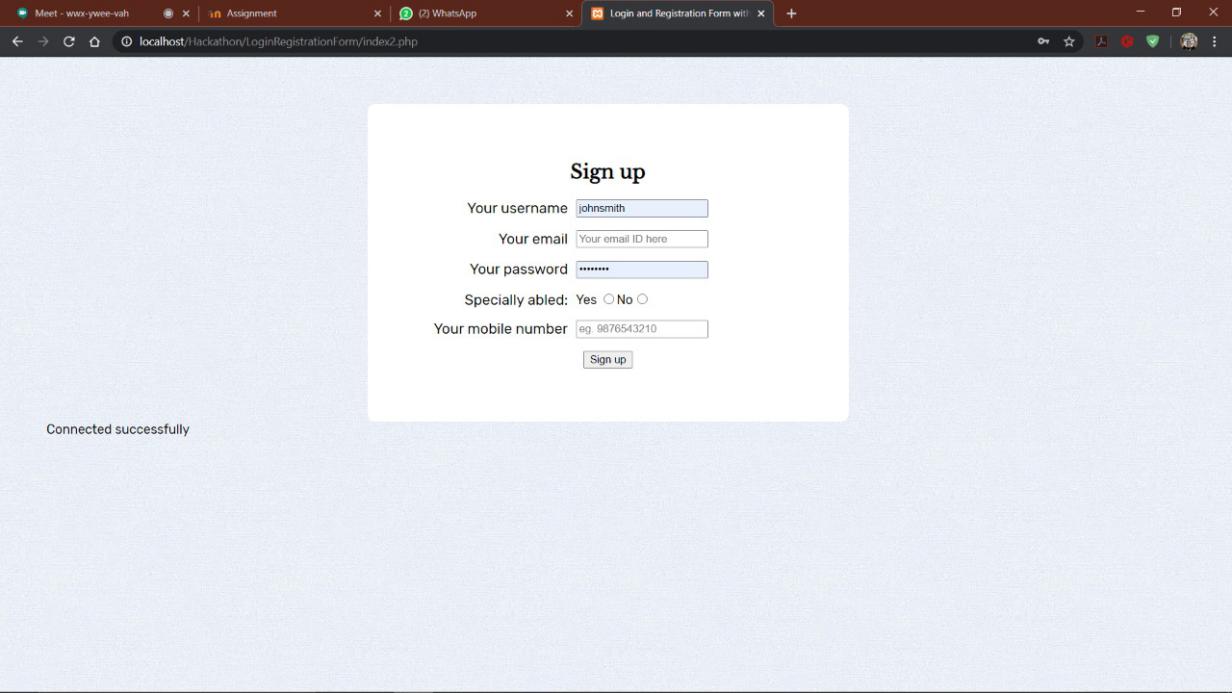
1. **GOGLE MAP IN HOME PAGE OF WEBSITE**

****

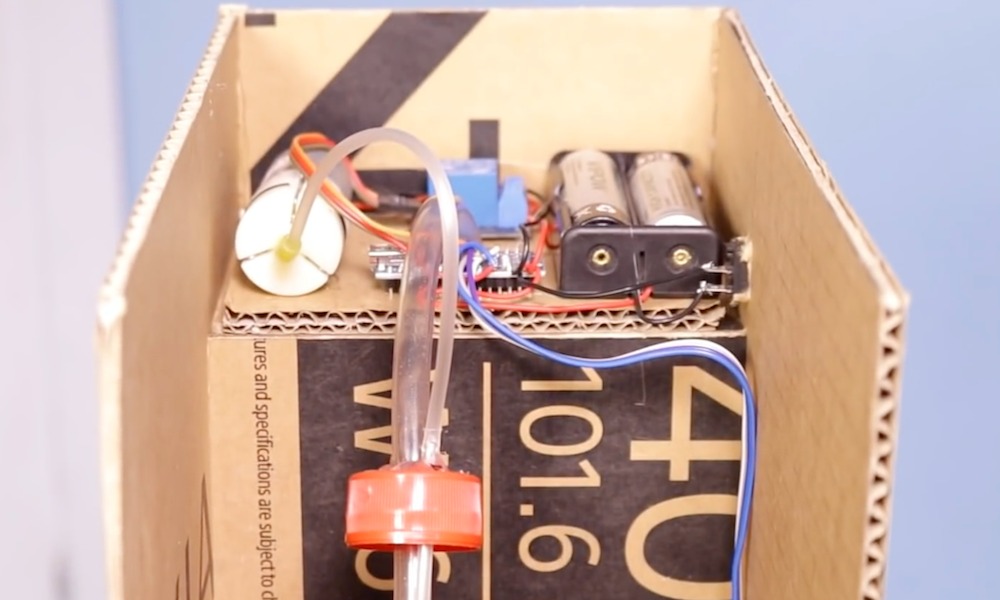
1. **HOME PAGE FOR GETTING NOTIFICATIONS THROUGH MAIL,CHAT AND DISTRESS SIGNAL**

****

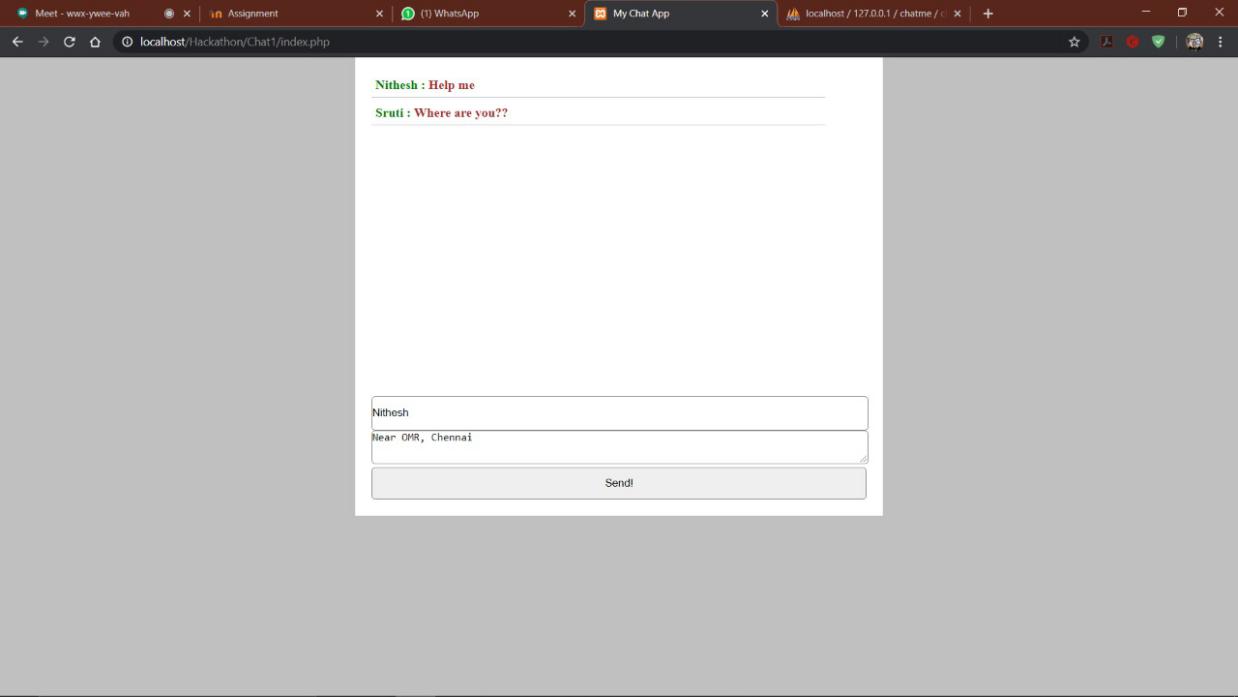
1. **SIGN UP FOR GETTING MAIL ABOUT ANY DISASTER**

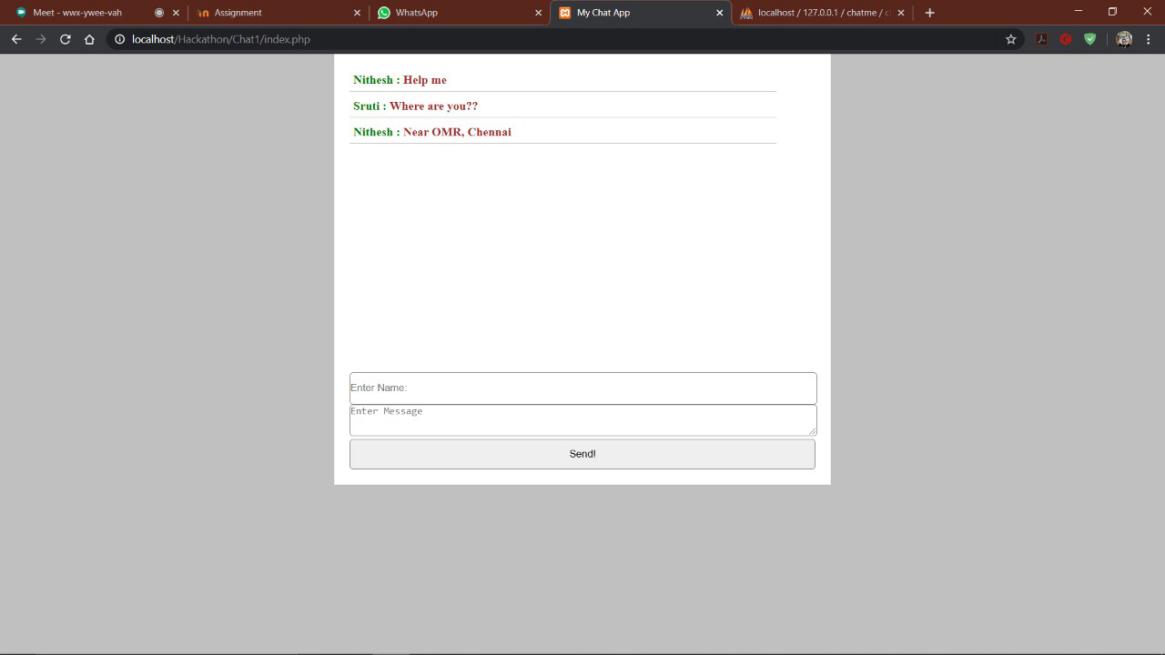
****

1. **BLACK BOX WITH SENSORS**

****

1. **ASKING HELP FORUM**

****

****

**4 CONCLUSION**

Our team of five consisting of one student from ECM,one from ECE and three from CSE immensely used our knowledge of subjects we studied and made a website with the help of knowledge from INTERNET AND WEB DEVELOPMENT course,IOT and ML and won the second prize in the hackathon organised by OWASP club on 14th February,2020 and each member of our team won six hundred rupees cash.We are immensely proud of ourselves that we utilized our knowledge to help women,disabled people and people in trouble because of natural calamities.

**BIO DATA**

1. **SRUTI VERMA(18BCE1005)**

Phone NO.: 6380886773

Email-id: sruti.verma2018@vitstudent.ac.in

1. **K.G. NITHESH GURUDAAS(18BCE1101)**

Phone NO.: 9994709300

Email-id: kg.nitheshgurudass2018@vitstudent.ac.in