**A**

**Mini Project Report**

**On**

**“WATER LOGGING PREVENTION AND DETECTION ”**

Submitted in partial fulfilment of the Requirements for the award of the degree of

**Bachelor of Technology**

**In**

**Computer Science Engeneering**

**By**

**20R21A05D7 - Danduboina Revanth**

**20R21A05C3 - Sai Satwika**

**20R21A05E9 - Gonuguntla Gayathri**

**21R25A0515 - Kasina Anjali**

Under the guidance of

**Dr E.Anu priya**

**PROFESSER**

**&**

**HEAD**

**Department of Computer Science & Engineering**

**2022**



**Department of Computer Science & Engineering**

**CERTIFICATE**

This is to certify that the project entitled **“Water logging prevention and detection ”** has been submitted by **Danduboina Revanth**(**20R21A05D7), Alagandula Sai Satwika(20R21A05C3), Gonguntula Gayathri(20R21A0E9), Kasina Anjali(21R25A0515)** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

**Internal Guide Head of the Department**

**External Examiner**

**i**



**Department of Computer Science & Engineering**

**DECLARATION**

We hereby declare that the project entitled **“Water Logging Prevention And Detection”** is the work done during the period from **August 2022 to December 2022** and is submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

**Danduboina Revanth 20R21A05D7**

**Alagandula Sai Satwika 20R21A05C3**

**Gonuguntla Gayathri 20R21A05E9**

**Kasina Anjali 21R25A0515**

**ii**

**Department of Computer Science & Engineering**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express our guidance for all of them.

First of all, we would like to express our deep gratitude towards our internal guide **Dr. E. ANUPRIYA, Assistant Professor, HOD, Department of CSE** for her support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. E. ANUPRIYA,** HOD, Dept. of CSE for providing the facilities to complete the dissertation.

We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

**Danduboina Revanth 20R21A05D7**

**Alagandula Sai Satwika 20R21A05C3**

**Gonuguntla Gayathri 20R21A05E9**

**Kasina Anjali 21R25A0515**

**iii**



**Department of Computer Science & Engineering**

**ABSTRACT**

In present times water logging has become one of the biggest problem in traffic management. Due to poor planning during the construction of roads this problem has been affecting many people of the society. Even after this problem is recognized by the government, no proper measurement were taken.

This project is used to overcome the problem of water logging in cities during

rains. It is a combination of IOT and web application.We collect the level of water logging on a particular road using sensors, then according to this data, pump will get triggered and send the rain water to near by water reservoir for purification.This data will also be used to create a map application in which user can check out the details of water logging from any place and avoid the traffic.

iv

**LIST OF FIGURES & TABLES**

|  |  |  |
| --- | --- | --- |
| **Figure Number** | **Name of the Figure** | **Page**  **Number** |
| 1 | Ardiuno UNO componets | 8 |
| 2 | Block diagram of working model | 9 |
| 3 | System Architecture | 11 |
| 4 | Use Case Diagram | 12 |
| 5 | Sequence Diagram | 13 |

**v**

**INDEX**

|  |  |
| --- | --- |
| **Certificate** | i |
| **Declaration** | ii |
| **Acknowledgement** | iii |
| **Abstract** | iv |
| **List of Figures and Tables** | v |
| **Chapter 1** |  |
| **Introduction** | 1 |
| 1.1 Overview | 1 |
| 1.2 Purpose of the project | 1 |
| 1.3 Motivation | 2 |
| **Chapter 2** |  |
| **Literature Survey** | 3 |
| 2.1 Existing System | 3 |
| 2.2 Limitations of Existing System | 5 |
| **Chapter 3** |  |
| **Proposed System** |  |
| 3.1 Proposed System | 6 |
| 3.2 Objectives of Proposed System | 6 |
| 3.3 System Requirements | 6 |
| 3.3.1 Software Requirements | 6 |
| 3.3.2 Hardware Requirements | 7 |
| 3.3.3 Functional Requirements | 7 |
| 3.3.4 Non-Functional Requirements | 7 |
| **Chapter 4** |  |
| **System Design** | 10 |
| 4.1 Components/ Users in the Proposed System | 10 |
| 4.2 Proposed System Architecture | 11 |
| 4.3 UML Diagrams | 12 |
| 4.3.1 Use Case Diagram | 12 |
| 4.3.2 Sequence Diagram | 13 |
| 4.4 Module Diagram | 15 |
|  |  |
|  |  |

**Chapter 5**

**5.1 Source Code**

**Chapter 6**

**6.1 Results**

**6.2 Verification and Validation**

**Conclusion**

**References**

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

People travels on the roads, in rainy seasons the roads are blocked with traffic because rain water logging on roads and manholes are full with water. So to overcome these problems our project will help to prevent these problems we are using sensor at manhole when manholes are filled with rain water then the rain water goes to reservoirs through underground pipe lines at the nearest lakes rivers etc.. and by taking all these we pass information to the people in google maps in which area traffic is more by that information they can decide their way to reach their destination.

* 1. **PURPOSE OF THE PROJECT**

the purpose of this project to reduce traffics on roads and to keep the roads clean and also to convey the traffic information to people.

* 1. **MOTIVATION**

This problems has affected many people so far. Water logging has also gave a lot of trouble for ambulance and emergency services.

This has given us the idea of making this project which we this would be very useful to the people.

**1**

**CHAPTER 2**

**LITERATURE SURVEY**

We conducted a thorough literature survey by reviewing existing systems for detecting fake and clone accounts. Research papers, journals and publications have also been referred in order to prepare this survey.

**2.1 EXISTING SYSTEM**

There is no proper system or solution for this problem.

This project would be the first solution for this problem

**2.2 LIMITATIONS OF EXISTING SYSTEM**

Concisely summarizing the disadvantages of the above implementations:

* In present scenario due to heavy rainfalls, roads are blocked. This water is sent directly into manholes and all of this water is getting wasted.
* As there are no proper measures taken to prevent this problem roads are blocked and it causes heavy traffic jams.
* In current situation water logging detection on map is done by satellite imaging which is a big process. Even in satellite imaging it can only detect a large amount of water logging like floods etc , small water loggings on city roads cannot be detected.

**2**

**CHAPTER 3**

**PROPOSED SYSTEM**

**3.1 PROPOSED SYSTEM**

We can overcome this problem using IOT. We use components like Arduino UNO, ultrasonic sensor, pump etc.Using the above data we provide a mapping application for the user to know locations of water logging from any place.This project uses IOT to detect even small water loggings(like city roads etc.) and time consumption is less.

**3.2 OBJECTIVES OF PROPOSED SYSTEM**

The objectives of the proposed system include the following:

* To reduce traffic.
* To reduce water logging on roads in rainy season.
* To give information to people about traffic.
* To show the information of traffic in maps using HTML,CSS.

**3.3 SYSTEM REQUIREMENTS**

Here are the requirements for developing and deploying the application.

**3.3.1 SOFTWARE REQUIREMENTS**

Below are the software requirements for the application development:

* Heroku to keep the server running,
* developer.here for maps API,
* Arduino IDE,
* Atom,
* Terminal,
* git and git-hub.

**3.3.2 HARDWARE REQUIREMENTS**

Below are the hardware requirements for the application development:

* Arduino UNO: The Arduino hardware and software was designed for artists,

**3**

* designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV!
* Ultrasonic Sensor : Ultrasonic sensors are used primarily as **proximity sensors**. They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology
* DC pump : The relay is the intermediate component between the Arduino board and the water pump. It allows the Arduino board to control the water pump. The role of the relay is to start or stop the pump responsible for filling the bottles. the role of the push button is the activation or deactivation of the relay module
* ESP8266 : What is the ESP8266? The ESP8266 module **enables microcontrollers to connect to 2.4 GHz Wi-Fi**, using IEEE 802.11 bgn. It can be used with ESP-AT firmware to provide Wi-Fi connectivity to external host MCUs, or it can be used as a self-sufficient MCU by running an RTOS-based SDK
* System with basic configuration like Windows 7, 32-bit.

**3.3.3 FUNCTIONAL REQUIREMENTS**

* Efficient map rendering - the API is built for WebGL and HTML5-capable environments to maximize map and map object rendering efficiency on mobile devices and desktops. It also includes support for high-DPI devices to show the best maps wherever possible.
* HERE Maps map data - the API provides full access to world-leading map data and map images, with a choice of view modes and customization options, including three main map types: map, terrain and hybrid.
* Geocoding - the API provides full access to geocoding and reverse geocoding services.
* Routing - the API supports route calculation and display, with a variety of customization options such as public transport routing and others.
* Enterprise routing - the API supports advanced enterprise routing use cases such as isoline routing or truck routing.
* Places - Places API allows application developers to build POI and Address search experiences, giving users access to rich information about tens of millions of places around the globe.
* Custom map objects - the API supports the creation of both interactive and non-interactive map objects: markers with SVG, HTML or bitmap images geo shapes, including polygons, polylines, circles

**4**

**3.4 CONCEPTS USED IN THE PROPOSED SYSTEM**

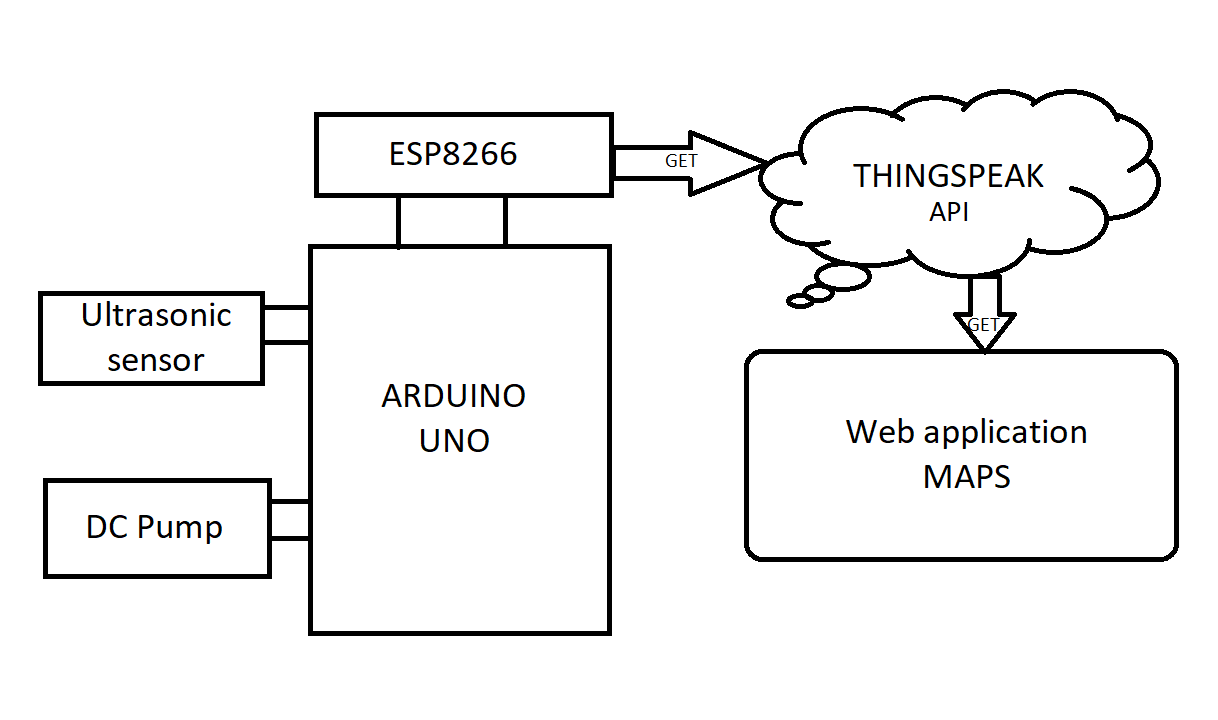
**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 COMPONENTS OR USERS IN THE PROPOSED SYSTEM**

**4.2 PROPOSED SYSTEM ARCHITECTURE**

* First ultrasonic sensor,ESP8266 , pump and relay are connected to Arduino UNO and trigger the pump when a certain water level is reached.
* This water level data is sent to “thingspeak” server using ESP-01 by their API.
* A mapping application is developed to show the water logging location to the user.
* This is achieved by making a GET request to “thingspeak” server and getting the data sent by the Arduino and place markers on the map accordingly.



**5**

**CHAPTER: 5**

**SET UP AND CONNECTIONS**

**ULTRA SONIC ARDUINO UNO:**

GND – GND

echo – 7

trig – 8

Vcc – 5v

**ESP2688 ARDUINO UNO**

1 – 2

2 – GND

3 – 3.3V

7 – 3.3V

8 – 3

**SETUP:**

Attach the ultrasonic sensor to top of the setup facing downward. DC pump to the bottom of the setup and connect a outlet pipe to the pump.

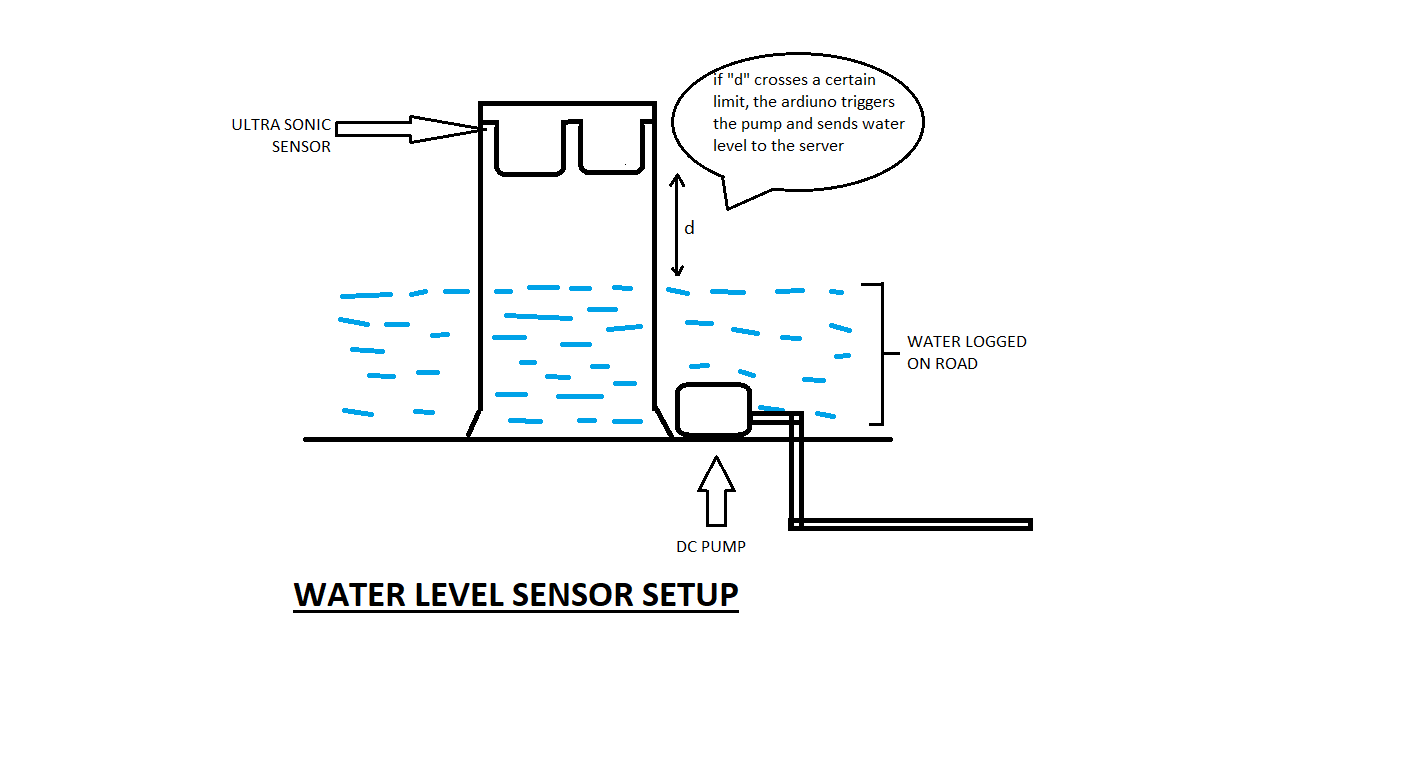
Connect the ESP2688 to the Arduino uno in order to connect to the internet and send the data from this to server.

Place the whole setup on the road sides where water logging.

There is no need for a external server because whole application runs on API calls on developer.here.com and thingspeak.com

The setup is arranged as shown in next figure.

**6**



This setup should be made water resistance since this will be set on road sides and it will be under heavy rainfall.

**7**

**Chapter : Source Code**

**HTML**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta name="viewport" content="initial-scale=1.0, width=device-width" />

<title>Water logging</title>

<!-- map core -->

<script src="https://js.api.here.com/v3/3.1/mapsjs-core.js" type="text/javascript" charset="utf-8"></script>

<script src="https://js.api.here.com/v3/3.1/mapsjs-service.js" type="text/javascript" charset="utf-8"></script>

<!-- basic UI -->

<script src="https://js.api.here.com/v3/3.1/mapsjs-ui.js" type="text/javascript" charset="utf-8"></script>

<link rel="stylesheet" type="text/css" href="https://js.api.here.com/v3/3.1/mapsjs-ui.css" />

<!-- Map event -->

<script src="https://js.api.here.com/v3/3.1/mapsjs-mapevents.js" type="text/javascript" charset="utf-8"></script>

</head>

<body style="padding:0; margin:0">

<!-- <h1 style="text-align: center; font-style: italic;">Water logging Detection</h1> -->

<div align="center" id="mapContainer" style=" margin: auto; width: 100vw; height: 100vh "></div>

<script>

**6**

// const https = require('https');

if(navigator.geolocation){

const url = "https://api.thingspeak.com/channels/1941457/feeds.json?api\_key=9V61BTD80W69MSBC&results=1";

const request = new XMLHttpRequest();

navigator.geolocation.getCurrentPosition(position => {

//console.log(position)

let browserPosition={lat: position.coords.latitude,lng:position.coords.longitude}

var platform = new H.service.Platform({

'apikey': 'B9km\_iOVZ3-Rj76EmGowGl4KyRjMMU5Q17DH1hR4eoU'

});

var maptypes = platform.createDefaultLayers();

// Instantiate (and display) the map

const map = new H.Map(

document.getElementById('mapContainer'),

maptypes.vector.normal.map,

{

zoom: 15,

center: { lng:position.coords.longitude , lat:position.coords.latitude }

});

// Create the default UI:

var ui = H.ui.UI.createDefault(map, maptypes);

**7**

// Event insilization

var mapEvents = new H.mapevents.MapEvents(map);

// Add event listener:

map.addEventListener('tap', function(evt) {

console.log(evt.type, evt.currentPointer.type);

});

var behavior = new H.mapevents.Behavior(mapEvents);

let marker= new H.map.Marker(browserPosition,);

map.addObject(marker);

let view = map.getViewPort();

window.addEventListener('resize',function(){

view.resize();

});

//let m=map;

//api requestto get water level from thingspeak

request.open("GET" , url);

request.send();

request.onload = () => {

if(request.status==200){

var query = {id:0,lat:0,lng:0,waterLevel:0}

console.log(request.response);

var data= JSON.parse(request.response);

query.id = data.channel.last\_entry\_id;

query.lat = data.channel.latitude;

**8**

query.lng = data.channel.longitude;

query.waterLevel=data.feeds[0].field1;

console.log(query);

// Add the first marker

//var waterIcon = new H.map.Icon('images/temp.jpg');

// var svgIcon = '<svg height="100" width="100"><circle cx="50" cy="50" r="40"style="fill:#03c2fc;stroke:black;stroke-width:4;fill-opacity:0.4;stroke-opacity=0.9" /></svg> ';

// var waterIcon = new H.map.Icon(svgIcon);

let waterMarker = new H.map.Circle({lat:query.lat,lng:query.lng},100,{style: {strokeColor: 'rgba(0, 0, 0, 0.7)',lineWidth: 2,fillColor: 'rgba(255, 0, 0, 0.6)'}});

if(query.waterLevel<=500 ){

//addMark({lat:query.lat,lng:query.lng});

map.addObject(waterMarker);

}

}else{

console.log("Error in feching api data");

}}

//console.log(query);

});

}else{

alert("Location is not supported by browser");

}

</script>

</body>

</html>

**9**

**JS express server:**

const express = require('express');

const https = require('https');

const app = express();

app.use(express.static("static"));

app.get("/",function(req,res){

res.sendFile(\_\_dirname+"/index.html");

})

app.listen(3000,function(){

console.log("server is running on port 3000");

});

**Arduino IDE**

#include <SoftwareSerial.h>

#define RX 2

#define TX 3

String AP = "realme 7";       // AP NAME

String PASS = "revanth04"; // AP PASSWORD

String API = "JX9B4DHLYWYSVN4K";   // Write API KEY

String HOST = "api.thingspeak.com";

String PORT = "80";

String field = "field1";

int countTrueCommand;

int countTimeCommand;

boolean found = false;

int valSensor = 1;

SoftwareSerial esp8266(RX,TX);

const int trigPin=8;

const int echoPin=7;

void setup() {

  Serial.begin(9600);

  esp8266.begin(115200);

  sendCommand("AT",5,"OK");

  sendCommand("AT+CWMODE=1",5,"OK");

  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");

}

**10**

void loop() {

 valSensor = getSensorData();

 String getData = "GET /update?api\_key="+ API +"&"+ field +"="+String(valSensor);

sendCommand("AT+CIPMUX=1",5,"OK");

 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");

 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");

 esp8266.println(getData);delay(1500);countTrueCommand++;

 sendCommand("AT+CIPCLOSE=0",5,"OK");

}

int getSensorData(){

  long time, dist;

  digitalWrite(trigPin, HIGH);

  delay(100);

  digitalWrite(trigPin,LOW);

  time = pulseIn(echoPin,HIGH);

  dist = (time/2)/29.1;

  delay(1000);

  Serial.println(dist);

  return dist;

}

void sendCommand(String command, int maxTime, char readReplay[]) {

  Serial.print(countTrueCommand);

  Serial.print(". at command => ");

  Serial.print(command);

  Serial.print(" ");

  while(countTimeCommand < (maxTime\*1))

  {

    esp8266.println(command);//at+cipsend

    if(esp8266.find(readReplay))//ok

    {

      found = true;

      break;

    }

    countTimeCommand++;

  }

  if(found == true)

  {

    Serial.println("OYI");

    countTrueCommand++;

    countTimeCommand = 0;

  }

**11**

  if(found == false)

  {

    Serial.println("Fail");

    countTrueCommand = 0;

    countTimeCommand = 0;

  }

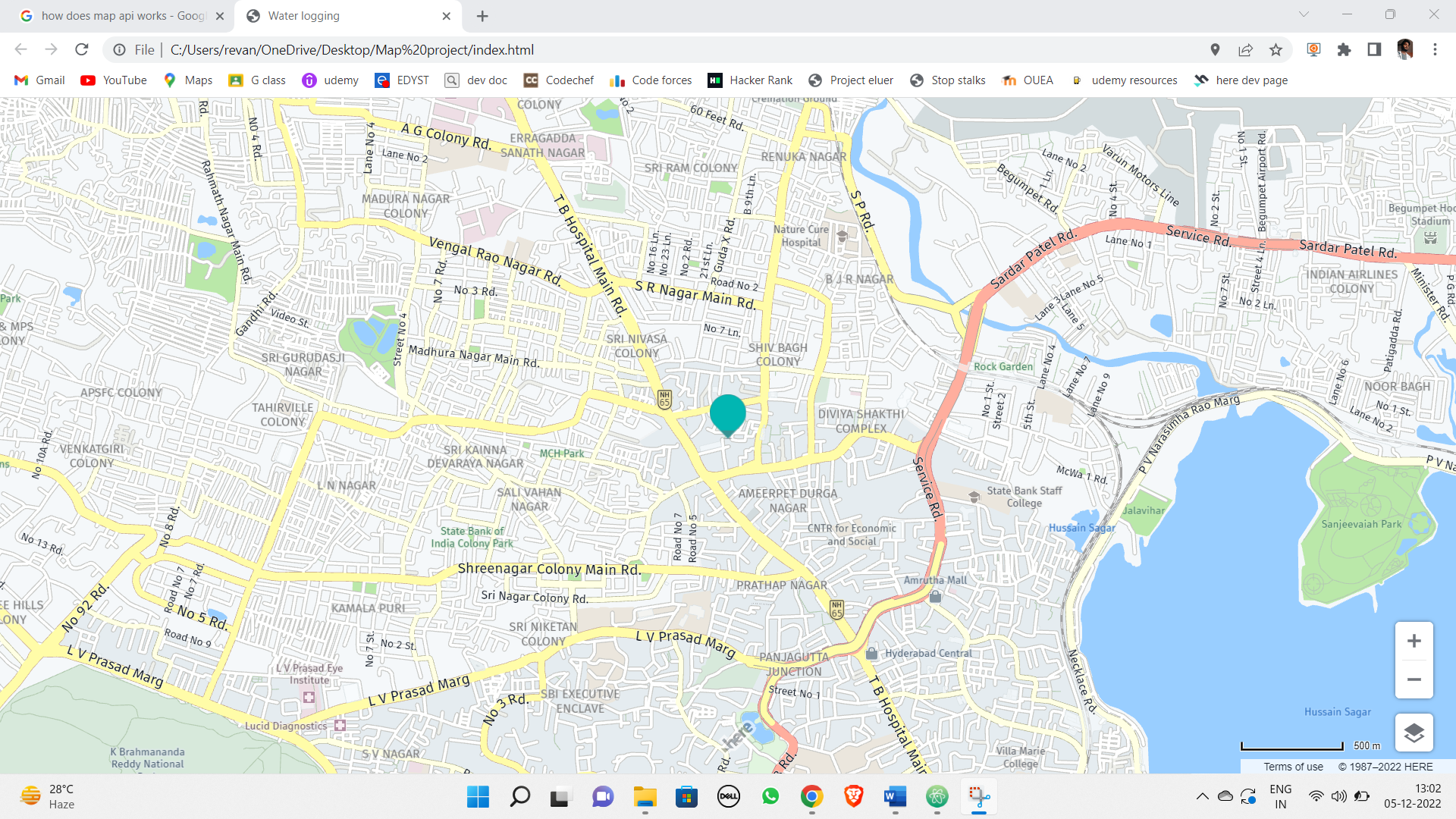
  found = false;

 }

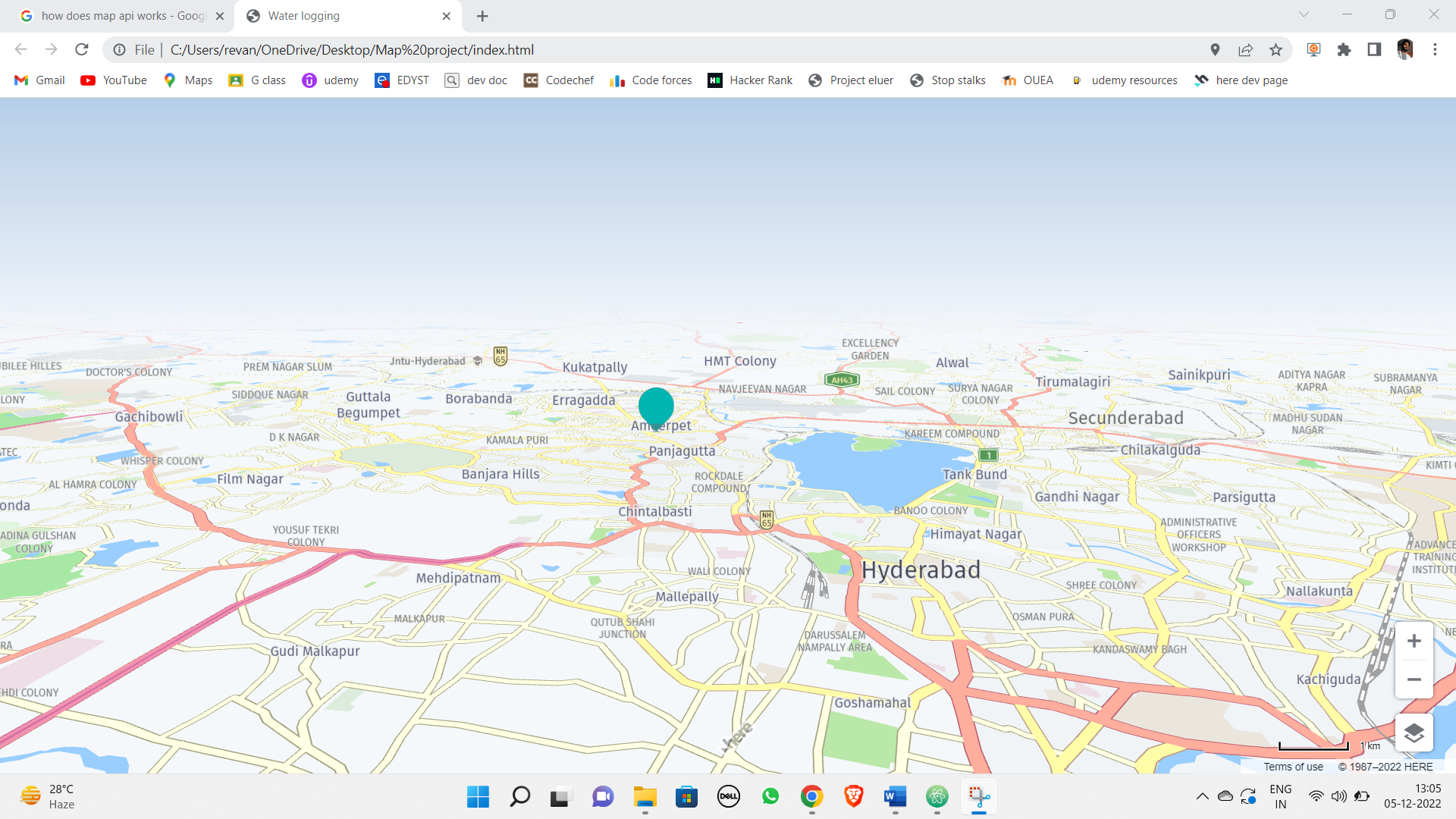
12

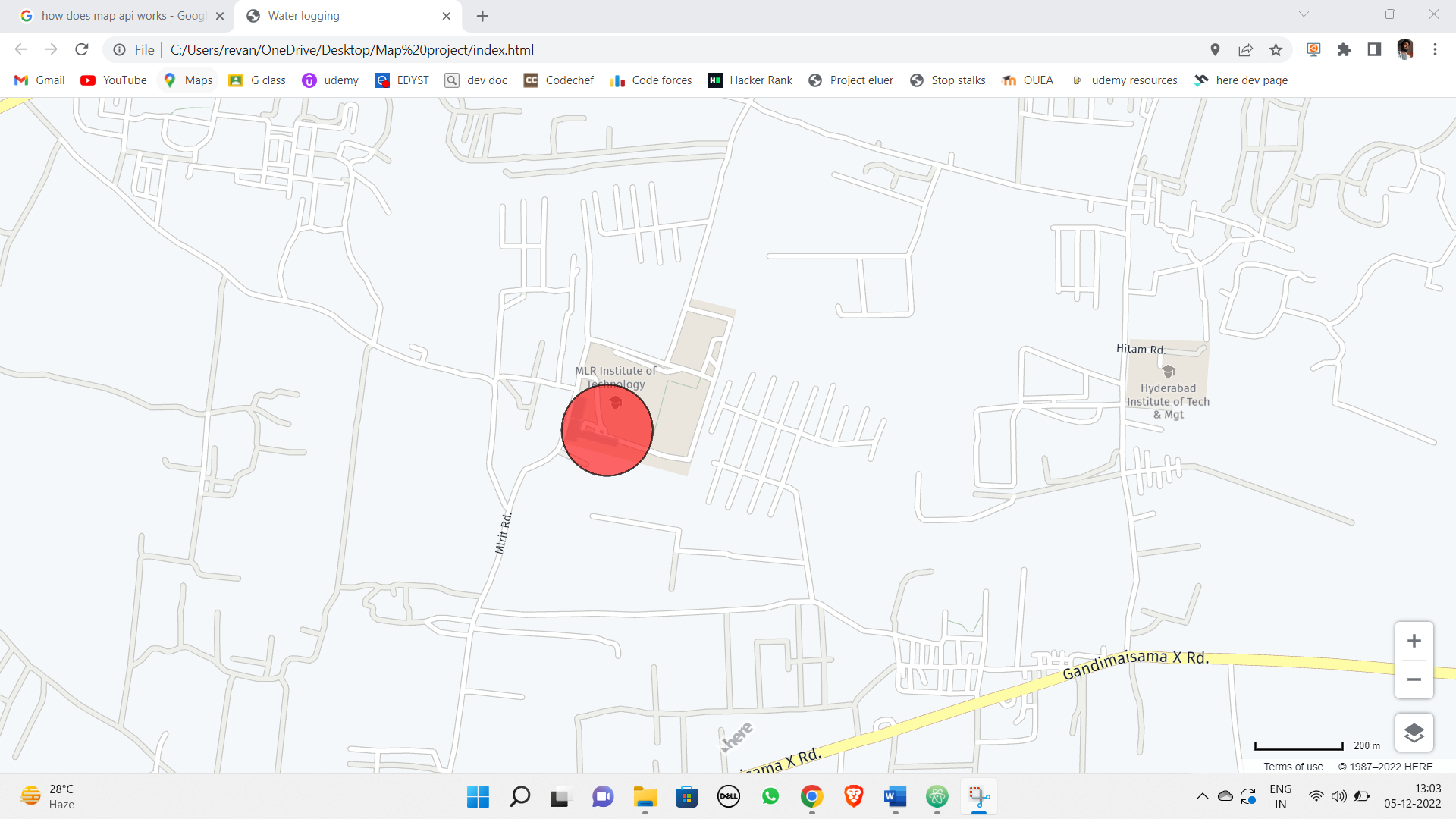
**Chapter :**

**Results and Implementation**



* Maps developed using here maps api.





The red marked location the place of water log on the road.

**14**

**Chapter :**

**Conclusion:**

* People from inclined areas, rural areas, are most affected from this problem. So this will be very
* This project will be very useful to reduces the water logging and traffic jams.
* In this we have learned about web technologies , IOT, REST API. ‘

REFERENCES:

<https://developer.here.com/documentation/>

<https://thingspeak.com/login?skipSSOCheck=true>

<https://docs.arduino.cc/hardware/uno-rev3>

**15**