VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING (SCOPE)

COVID19 Live Data Tracker and Alert System

Harshit Mishra
(19BCE0799)

Pankaj Shivnani
(19BCE0821)

Alokam Nikhitha
(19BCE2555)

Galla Kiran
(19BCE2584)

Project Report

of

CSE2006 – Microprocessor and Interfacing

Fall Semester 2021-22

Submitted To:

Faculty: Dr Abdul Majeed KK

Date:

Slot: E2

Abstract:

COVID19 is a novel coronavirus disease that causes illnesses ranging from the common cold to more severe diseases. It had a devastative effect on humanity worldwide resulting in the complete disruption of normal life. Due to its effect on humanity COVID19 was declared a pandemic by World Health Organisation in March 2020. The pandemic is not over yet and we still need a cautious and systematic approach in order to minimize its effect in near future.

In order to aid the process of systematic governance, we are trying to implement a COVID19 Live tracker and alert system that informs a user about the current situation of the pandemic in his/her locality. While we display the real time data to our users, we also alert them to strengthen their precautionary measures when the number of cases exceeds a given total. We have also demonstrated the working of a social distance detector which could come in handy to enforce the globally accepted 6-feet distance rule of minimizing COVID spread.

Introduction:

Our proposed model for live tracking the COVID data involves using the real time data as uploaded by common and reliable website sources. We are scraping the data of use, that involves counts such as total cases of a particular state or active cases of a particular state, etc. We then pass this data to our LCD using a Wi-Fi module of NodeMCU(ESP8266) and display it to our users.

The alert system works on a simple if and else condition. Similar to a tracking system, we parse in the active number of active cases, of a particular state, as our conditional statement. We also fix a threshold value that acts as an indicator to our alert system and if the threshold limit is breached by the active number of cases, we activate the alert systems.

The social distance detector uses an ultrasonic distance sensor to find the range of another being within 6-feet and it comes with an added buzzer that turns on when the distance is less than 6-feet.

Literature Survey:

S.No.	Paper Title	Name of the Conference/Journal, Year	Technology Used
1	Prediction and Effective Monitoring of Flood Using Arduino System Controller and ESP8266 Wi-Fi Module	International Journal of Communication and Networking System, 2019	The authors of this paper use Arduino Uno Controllers to control all the operations of the system. Wireless sensors are used to measure Temperature, Wind, Humidity, etc. All the collected data is stored on cloud and is accessed using ThingSpeak website and ESP8266 wifi module. Embedded C language is used for developing the proposed system. The authors use a C library called “wiring” for basic input and output functions. Main advantage of this model is to reduce the hardware component dependency of this system.
2	Covid-19 fever symptom detection based on IoT cloud	International Journal of Electrical and Computer Engineering, 2020	The authors use a Human Body Temperature Sensor(MAX30205) to detect Covid Symptoms. This data is then uploaded on the cloud platform using ThingSpeak with the help of ESP8266 node MCU. When any data value crosses a threshold, an alert is sent to the monitoring manager with the help of an SMS. Any device like Mobile Screens or Television Screens can be used to display visualizations of the collected data.
3	Design of Real-Time Weather Monitoring System Based on Mobile Application Using ESP8266	Journal Of Engineering Sciences, 2019	The authors use several sensors to measure weather conditions like Temperature sensor(LM35), Humidity Sensor, Gas Sensor, LCD, ESP8266 Wifi Module.

			Raspberry Pi is used as the software tool. Node MCU ESP8266 platform allows sending the data collected by weather sensors to the server. This continuous process gives a real-time analysis and monitoring of data. The advantage of this approach is that it uses hardware tools that are cheaper and easily available and can be expanded to use more parameters.
4	IoT-cloud based healthcare model for COVID-19 detection: an enhanced k-Nearest Neighbour classifier based approach	Springer, 2021	The authors of this paper propose a classification based algorithm called Enhanced K-Nearest Neighbors(e-KNN) that is very similar to the traditional K-Nearest Neighbors algorithm. The major difference is that e-KNN computes the value of K dynamically and it also uses ACO (Ant Colony Optimization) based feature selection mechanism. Several performance metrics like Accuracy, Precision, Recall and F1-Score proved that the e-KNN performs much better than the traditional K-Nearest Neighbors algorithm. This system can work as a backend to an IoT based frontend. Similar methods like ESP8266 can be used to connect to a ThingSpeak cloud server and store and fetch data.
5.	IOT based smart patient monitoring system with emphasis on covid and associated respiratory diseases diagnosis	International journal of progressive research in science and engineering, vol.2, no.6, june 2021.	The authors of this paper Proposes a design to monitor the patient’s health conditions such as body temperature, pulse rate, respiration rate and send the message to the guardian using IOT

			platform. He also conducted an extensive survey to analyze how the usage of edge computing progresses the performance of Systems . The performance of edge computing is studied by comparing delay of network, occupation of bandwidth, Power utilization, and many other characteristics. An ardino Uno, heart rate-spo2 monitor, temperature sensor, Esp8266 and an LCD module are used to make the hardware module.
6.	Thing speak Based Sensing and Monitoring System for IOT with Matlab Analysis	International Journal of New Technology and Research, Volume-2, Issue-6, June 2016	The author proposes a project which can provide and prove the strength of IOT using the Thing speak API that is capable to contribute the services for the purpose of building vast number of IOT applications and help to implement them on the public platform. From the project he concludes that Microcontrollers will get minimized and vanish into the environment, and IOT Leads to become universal and in every prospect and the Thing speak IOT Web service is definitely a fascinating web based technology that encompasses the ability to form the expectations of the engineers.
7.	Live tracking system	International Journal of Engineering Research & Technology, Vol. 9 Issue 06, June-2020	The authors designs a GPS tracking device that sends location to the end user at a consistently high frequency. It offers user's real-time location updates, every few seconds which can be critical for the safety of the loved ones. Smart phone at the parent side and the tracking module at child side

			that supports GPS and SMS as a minimum are the basic requirements. The Android App to be used is a digital dashboard where a graphic interface can be build. The tracked location is decided by the latitude and longitude. Using this parameter the user can verify the location by putting received values on the Google of latitude and longitude. The advantage of the proposed system is here the user get one more option to track the device i.e. Android Application which also include the map option for user to observe
8.	Internet of things - Based photovoltaic parameter monitoring System using nodemcu ESP8266	International Journal of Electrical and Computer Engineering (IJECE) Vol.11, No.6, December 2021	The authors of this paper propose an IOT-based PV parameter monitoring system for parameters such as solar irradiance, ambient temperature, PV output voltage, PV output current, and PV output power. They are measured using photodiode, DHT22, impedance divider, and ACS712, respectively. PV output power is obtained from the product of PV voltage and PV current. The main controller is the Node MCU V3 ESP8266, It has the ability to create a WiFi enabled gateway so that the calculated parameters can be transferred wirelessly from the sensor to the cloud without the need for an external WiFi module. Finally , the acquired parameters are tracked on the Cloud server using Thing Speak.

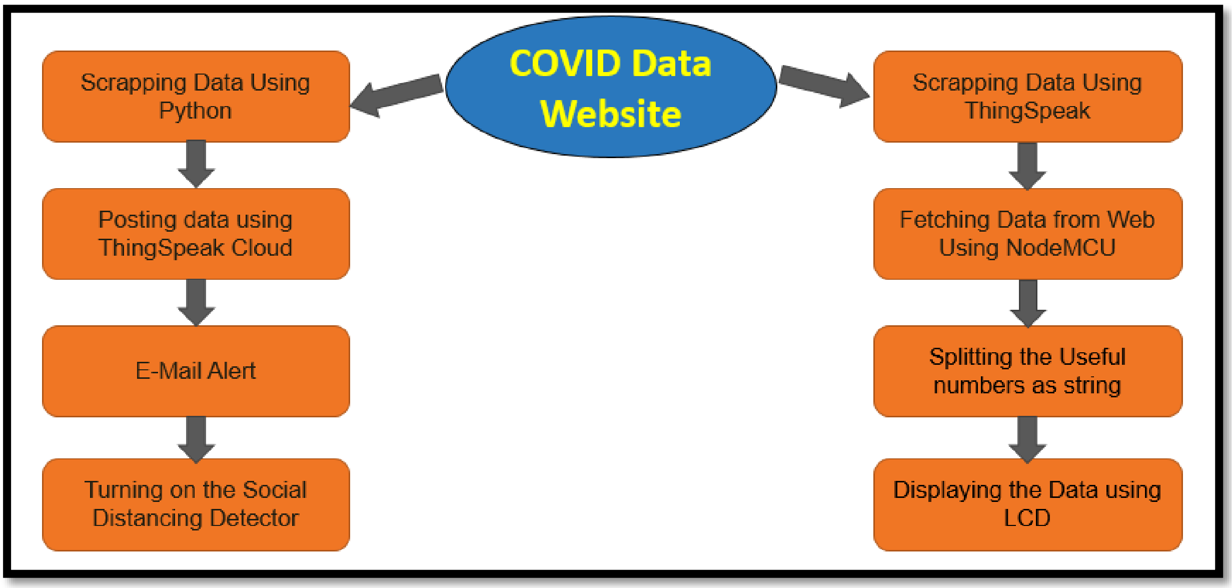
Drawbacks in Existing Work:

The existing approaches of this particular type of project includes live weather tracking that update once a day. Our project is more robust as it shows us the live count as soon as it is updated in the source webpage. Not only this, the existing system use cloud servers which renders the data with some delay. We have tried to minimize the data fetching by using ThingSpeak API. We have thus tried to improve the speed of data rendering in our project. The pre-existing project which involved health monitoring system needed a manual update once in a while through their equipment, while our project being robust fetches real time data and is fully automated with the help of python script that automatically scrapes a particular page for us. The papers proposed above involves various parameter checks which make the project expensive. The parameter checks are not necessary in our idea as we are scraping a free web page and need no input reading from microcontroller's environment and hence offer a very low price for the same service offered in the papers referred.

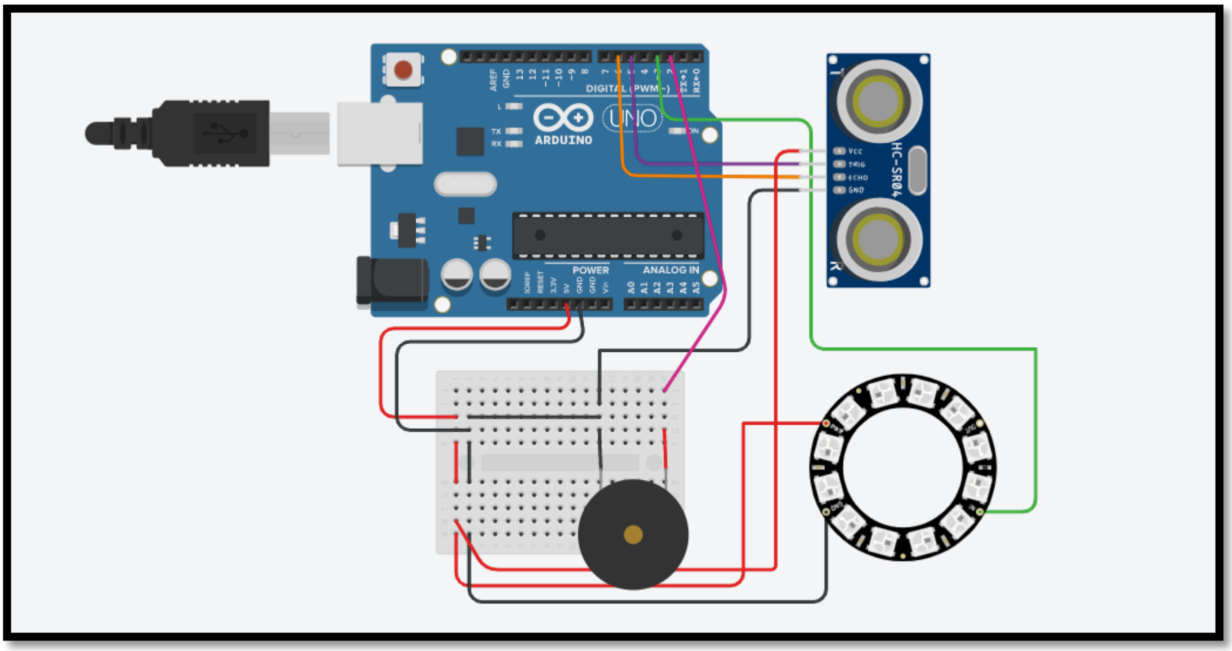
Proposed Work (Novelty):

There is exists no paper that proposes the data tracking of Coronavirus cases. The existing approaches included health monitoring system or weather applications. This clearly suggests that the idea of our project is relatively novel and new. The other thing that we have added on to our project is the alert system. The alert generally finds its use in burglary alarm system, but we have used the same idea to be implemented in this case of detecting coronavirus cases. Finally, we have also demonstrated the working of social distance detector which is completely new to this field. The distance detector is used in automated robotic systems like self-driving cars or industrial robots. We have used the same approach to make our social distance detector by adding a limit of 6-feet and adding a buzzer to it. This gives a completely new dimension to our COVID19 Live tracker and alert system.

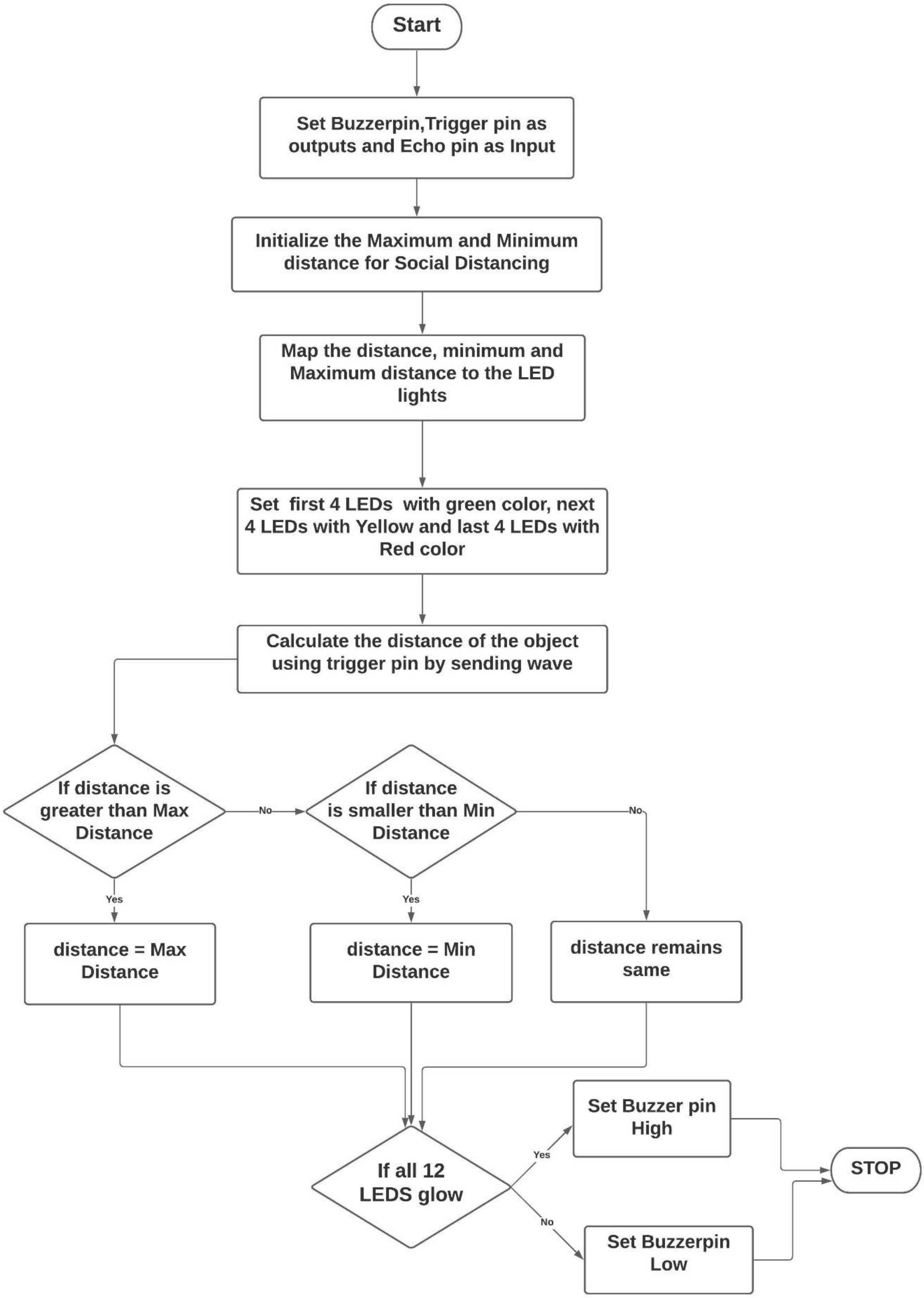
Block Diagram:


COVID Data Website
Scrapping Data Using Python
Posting data using ThingSpeak Cloud
E-Mail Alert
Turning on the Social Distancing Detector
Scrapping Data Using ThingSpeak
Fetching Data from Web Using NodeMCU
Splitting the Useful numbers as string
Displaying the Data using LCD

Pin Diagram:



Flow Chart:


Start
Set Buzzerpin,Trigger pin as outputs and Echo pin as Input
Initialize the Maximum and Minimum distance for Social Distancing
Map the distance, minimum and Maximum distance to the LED lights
Set first 4 LEDs with green color, next 4 LEDs with Yellow and last 4 LEDs with Red color
Calculate the distance of the object using trigger pin by sending wave
If distance is greater than Max Distance
No
If distance is smaller than Min Distance
No
Yes
Yes
distance = Max Distance
distance = Min Distance
distance remains same
Set Buzzer pin High
If all 12 LEDS glow
Yes
No
Set Buzzerpin Low
STOP

Implementation:

A. Live Data Tracker:

- **a.** The live count is fetched from Zee news’ official website.
- **b.** The Scrapped data is auto-updated in ThingSpeack Cloud.
- **c.** The Arduino is coded in an online editor and then is burned with our NodeMCU.
- **d.** The LCD is connected with our NodeMCU to display our live data.

B. E-mail Alert System:

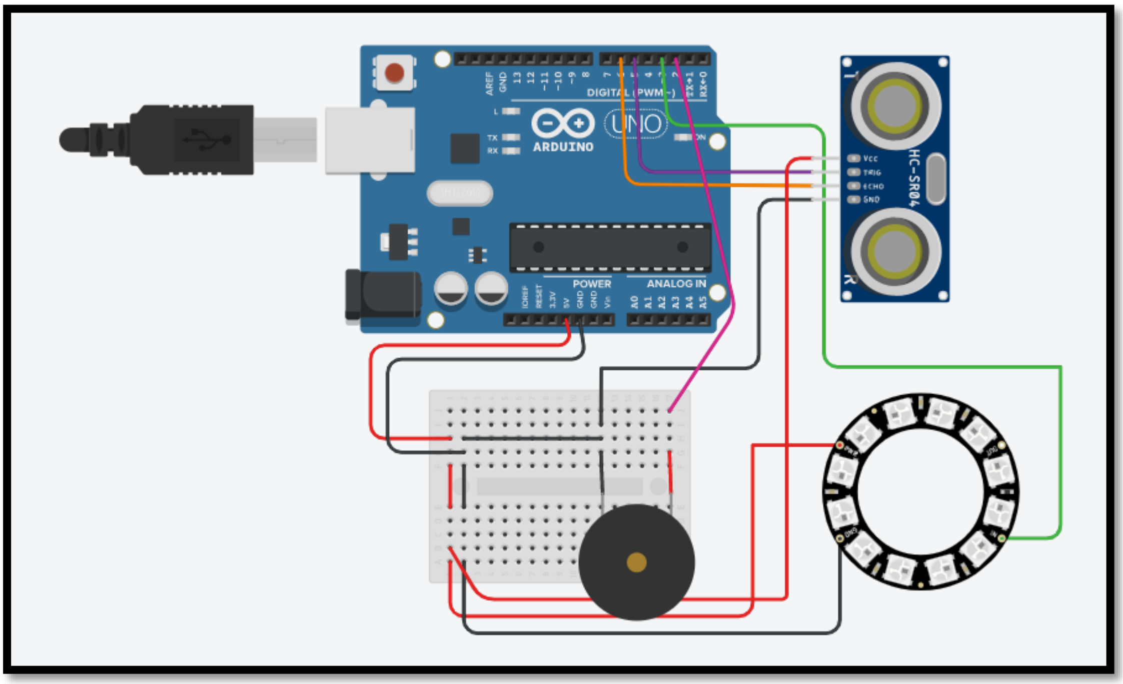
- **a.** The electronic mail system employs coding in MATLAB.
- **b.** The implementation is done using conditional branching in MATLAB.
- **c.** The number of active cases is considered and based on it an alert is sent to the fed email address of user.

C. Social Distance Detector:

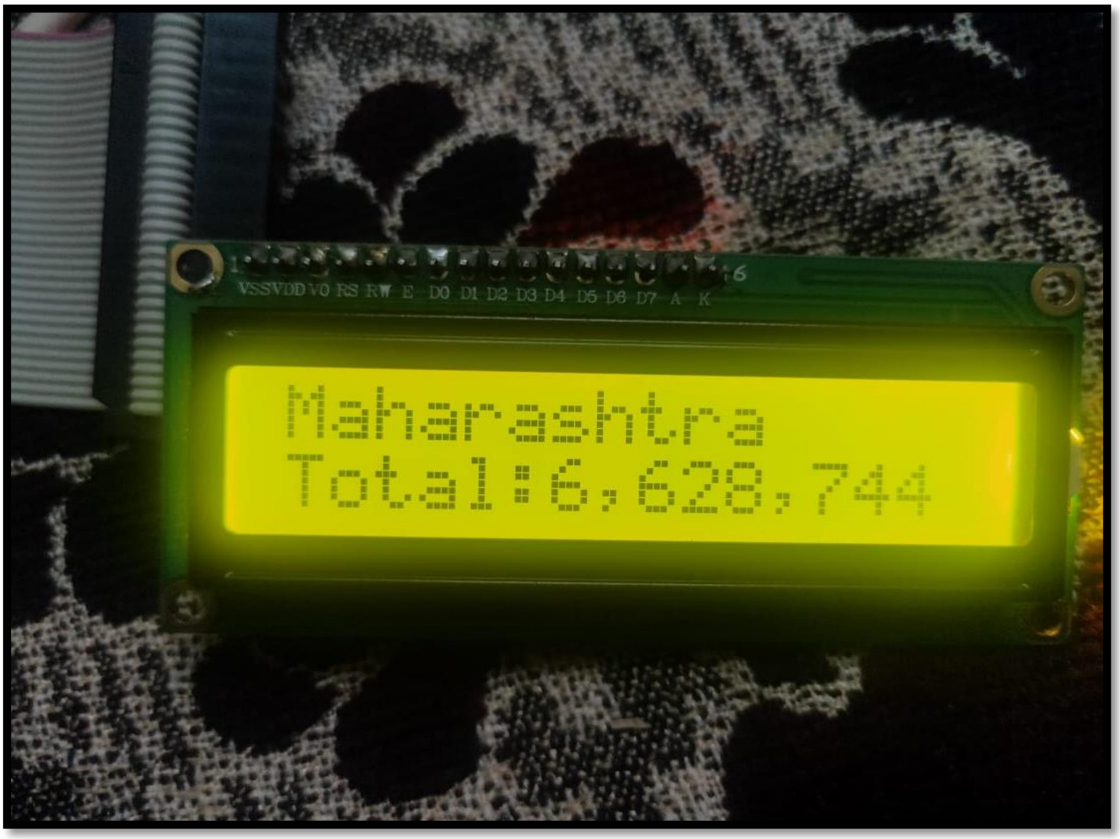
- **a.** The software simulation of social distance detector is demonstrated using Tinkercad.
- **b.** It uses ultraviolet sensors that detect the distance if a particular object enters its range.
- **c.** It is added with a buzzer which starts when the range is less than the specified limit.

Screenshots of the Prototype:

A. Social Distance Detector:

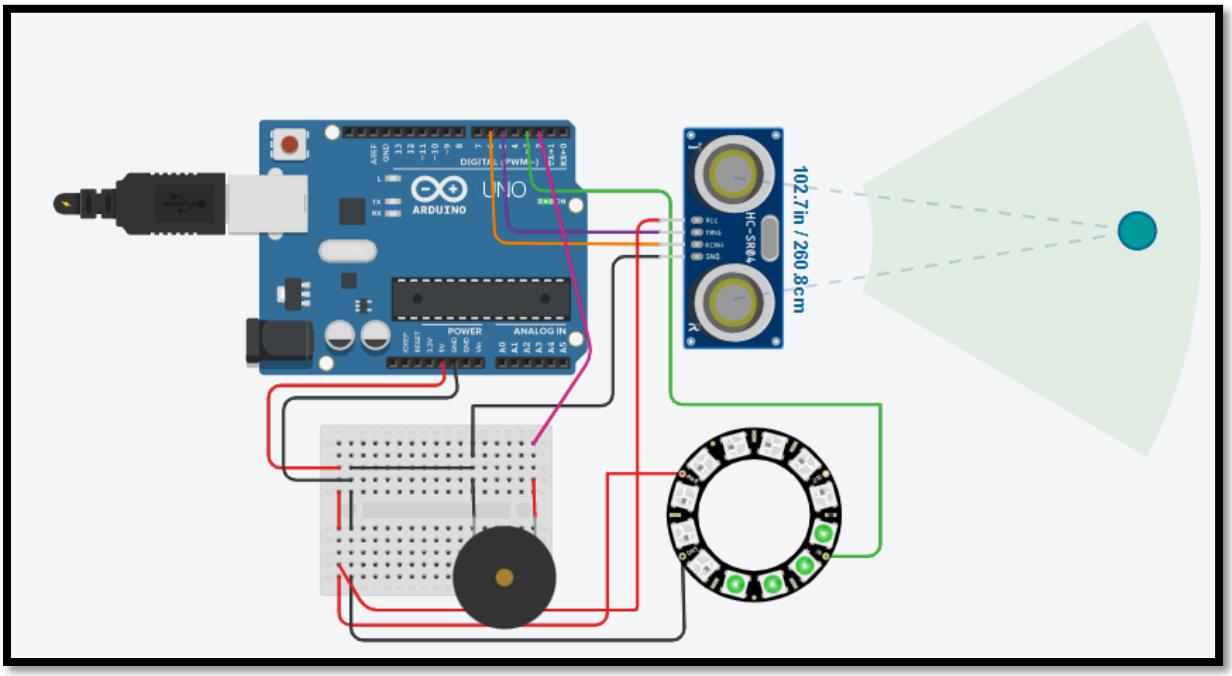


B. Live Data Tracker:

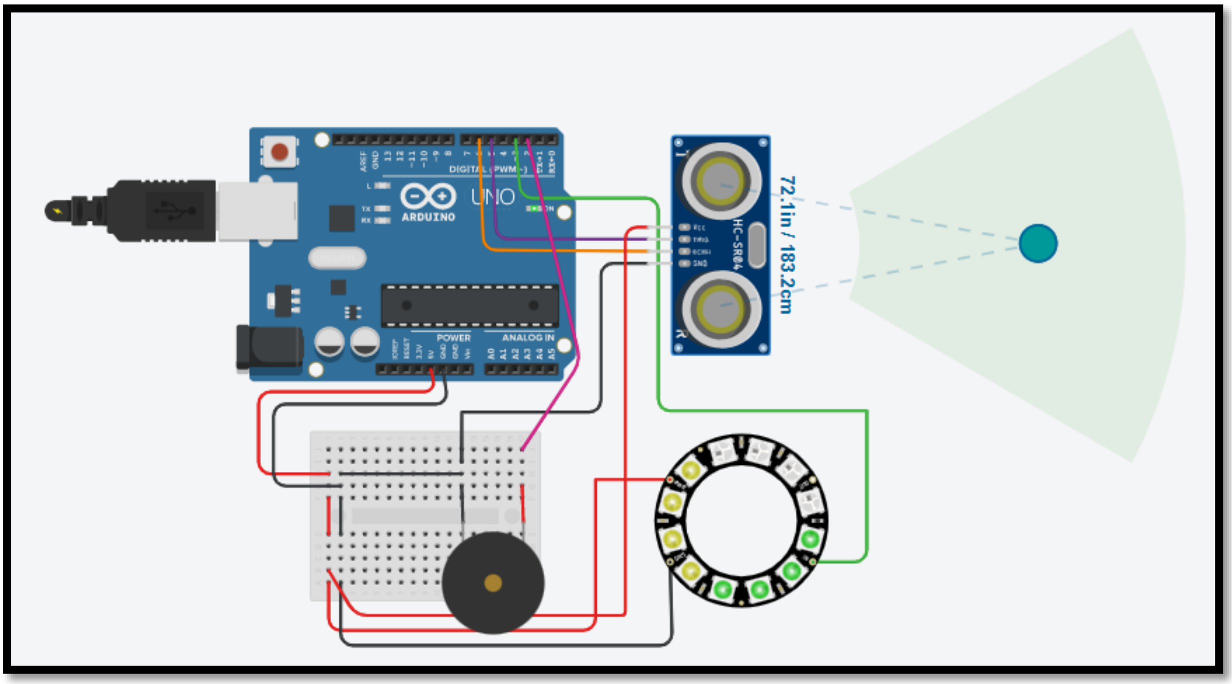


Results:

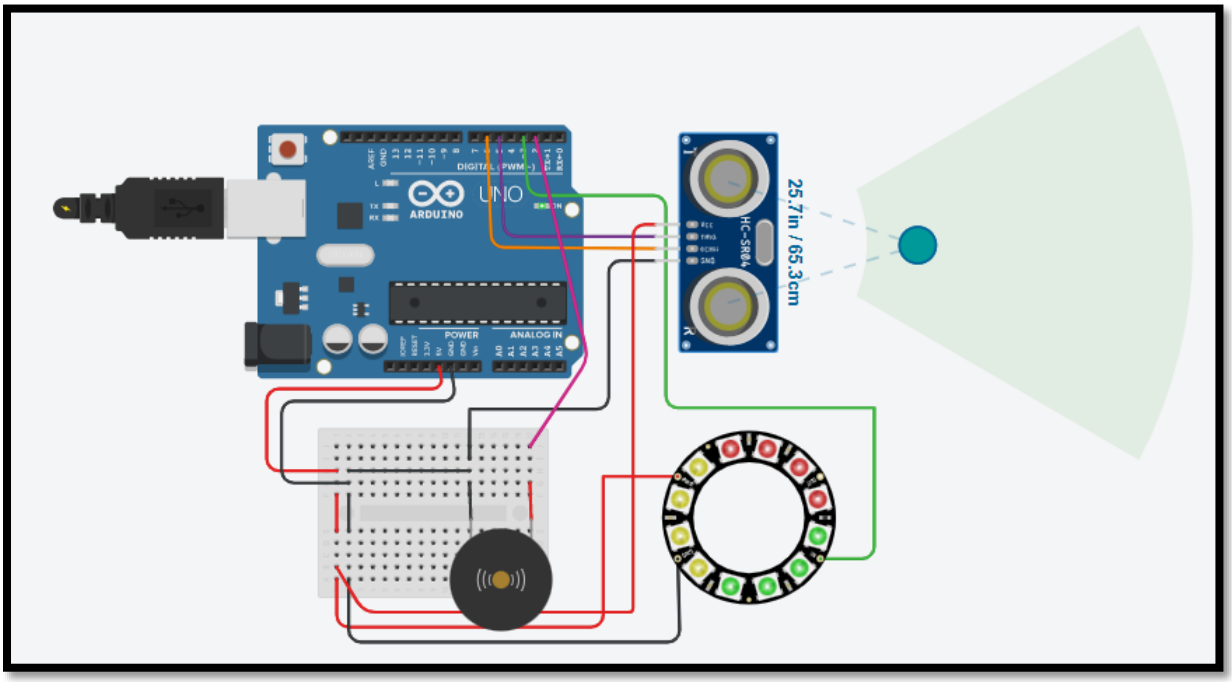
A) Social Distance Detector:



When social distance is maintained

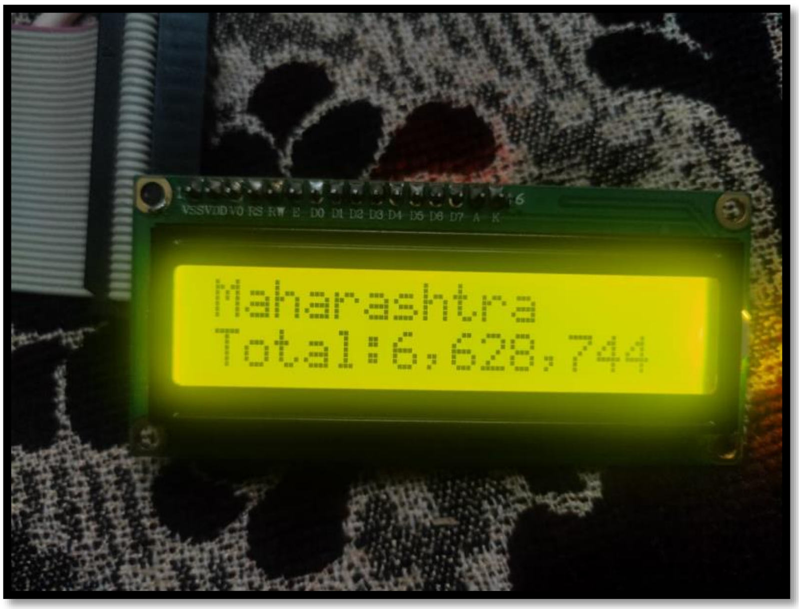


When the distance is still fine but just in range of 6-feet



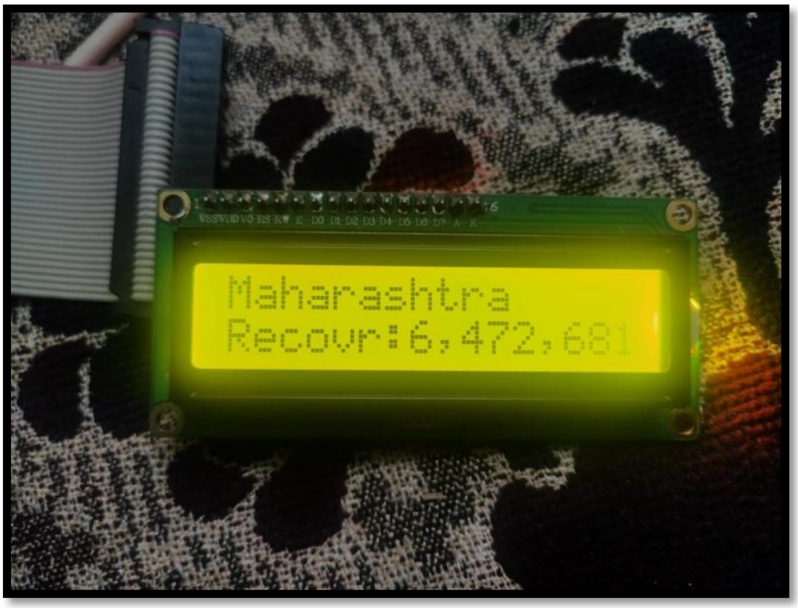
When distance is less than 6-feet

B) Live Counter:



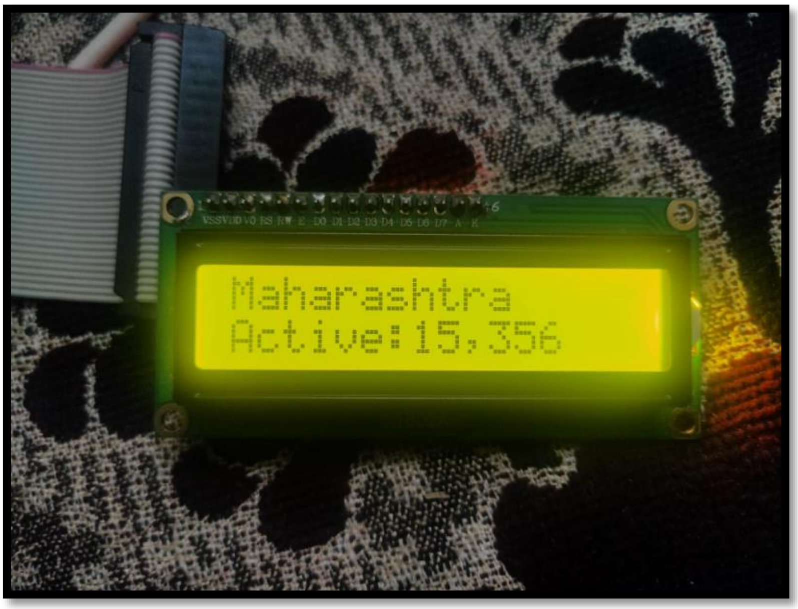
```
Maharashtra
Total:6,628,744
```

Total Cases in Maharashtra



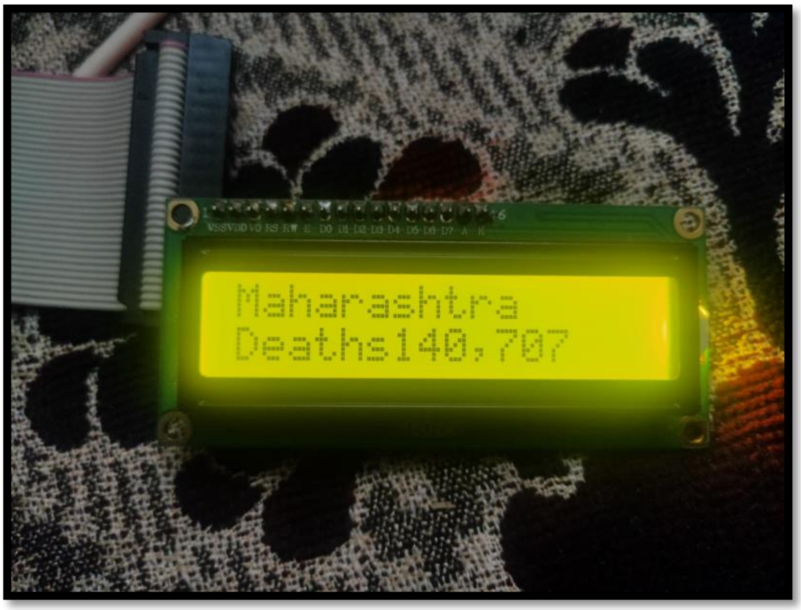
```
Maharashtra
Recovr:6,472,681
```

Total Recovered Cases in Maharashtra



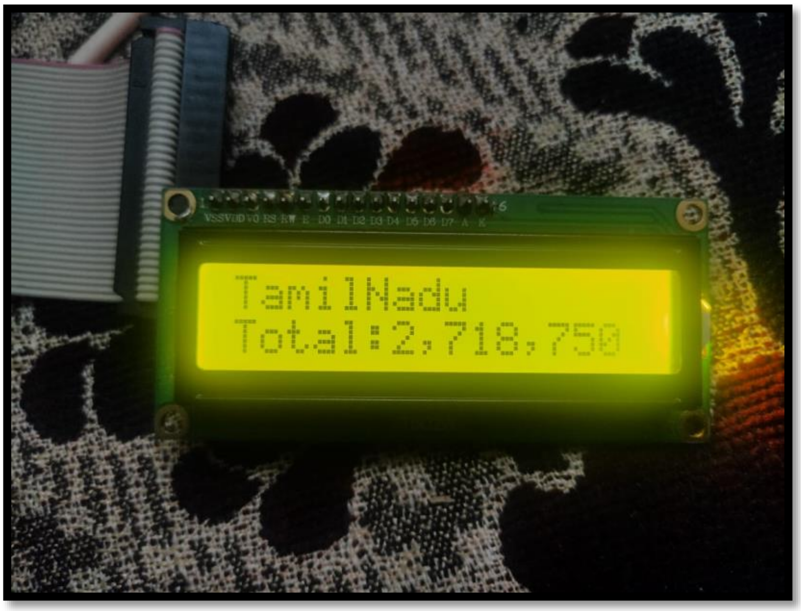
```
Maharashtra
Active:15,356
```

Number of Active Cases in Maharashtra



```
Maharashtra
Deaths140,707
```

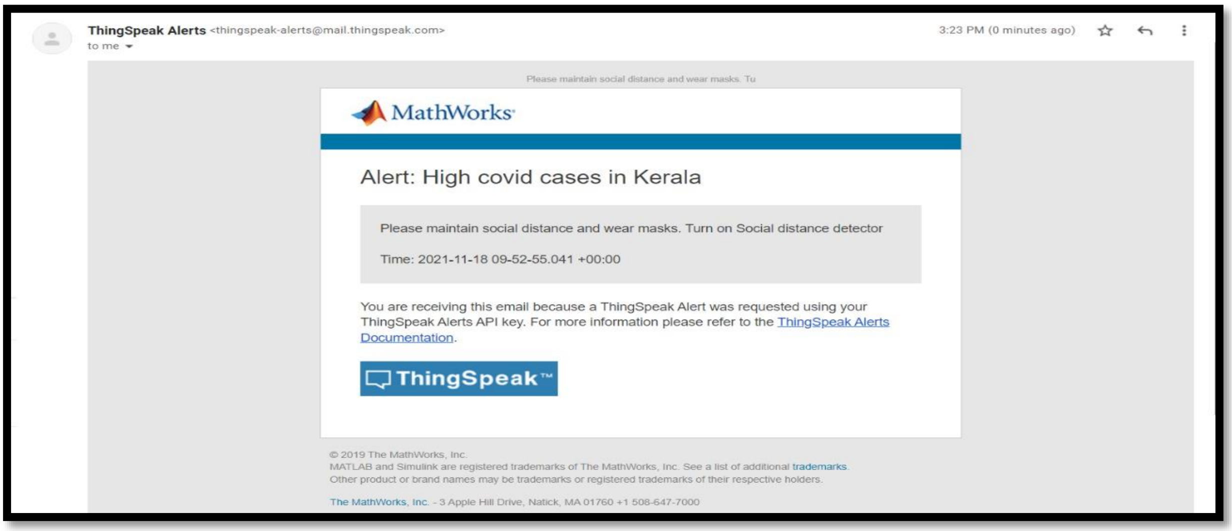
Total Deaths in Maharashtra



```
TamilNadu
Total:2,718,750
```

Total Cases in Tamil Nadu

C) E-Mail Alert:



ThingSpeak Alerts <thingspeak-alerts@mail.thingspeak.com>
to me

3:23 PM (0 minutes ago)

Please maintain social distance and wear masks. Tu

Alert: High covid cases in Kerala

Please maintain social distance and wear masks. Turn on Social distance detector

Time: 2021-11-18 09-52-55.041 +00:00

You are receiving this email because a ThingSpeak Alert was requested using your ThingSpeak Alerts API key. For more information please refer to the ThingSpeak Alerts Documentation.

© 2019 The MathWorks, Inc.
MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See a list of additional trademarks.
Other product or brand names may be trademarks or registered trademarks of their respective holders.

The MathWorks, Inc. - 3 Apple Hill Drive, Natick, MA 01760 +1 508-647-7000

E-mail Alert

Conclusions:

Hence, We have successfully Implemented a Live Covid Tracker with an Email Alert System . And also showed the live count as soon as it is updated in the source webpage in a LCD which is connected with our NodeMCU to display our live data. We also implemented a Social Distance Detector which detects if social distance is maintained or not and glow LEDs based on the distance and also give an alert buzzer if the distance goes less than min distance.

References:

1. S. L. Jurj, R. Rotar, F. Opritoiu and M. Vladutiu, "White-Box Testing Strategy for a Solar Tracking Device Using NodeMCU Lua ESP8266 Wi-Fi Network Development Board Module," 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), 2018, pp. 53-60, doi: 10.1109/SIITME.2018.8599250.

2. Ganorkar, Ankur. (2020). Live Tracking System. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS060770.

3. Menni, C., Valdes, A.M., Freidin, M.B. et al. Real-time tracking of self-reported symptoms to predict potential COVID-19. Nat Med 26, 1037–1040 (2020).

4. Tole Sutikno, Hendril Satrian Purnama, Anggit Pamungkas, Abdul Fadlil,Ibrahim Mohd Alsofyani, Mohd Hatta Jopri:Internet of things-based photovoltaics parameter monitoring system using NodeMCU ESP8266 International Journal of Electrical and

Computer Engineering (IJECE) Vol. 11, No. 6, December 2021, pp. 5578~5587 ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5578-5587

5. Md Khaja Pasha, G Aruna:DESIGN OF REAL-TIME WEATHER MONITORING SYSTEM BASED ON MOBILE APPLICATION USING ESP8266Vol 10,Issue 11, NOV/2019 ISSN NO:0377-9254

6. S. Saha and A. Majumdar, "Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system," 2017 Devices for Integrated Circuit (DevIC), 2017, pp. 307-310, doi: 10.1109/DEVIC.2017.8073958.
7. Škraba, A. Koložvari, D. Kofjač, R. Stojanović, V. Stanovov and E. Semenkin, "Prototype of group heart rate monitoring with NODEMCU ESP8266," 2017 6th Mediterranean Conference on Embedded Computing (MECO), 2017, pp. 1-4, doi: 10.1109/MECO.2017.7977151.

8. S. Halder and G. Sivakumar, "Embedded based remote monitoring station for live streaming of temperature and humidity," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017, pp. 284-287, doi: 10.1109/ICEECCOT.2017.8284683.

9. T. Thaker, "ESP8266 based implementation of wireless sensor network with Linux based web-server," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), 2016, pp. 1-5, doi: 10.1109/CDAN.2016.7570919.

10. Sharmad Pasha : Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis International Journal of New Technology and Research (IJNTR) ISSN: 2454-4116, Volume-2, Issue-6, June 2016

11. Hasan, M. W. (2021). Covid-19 fever symptom detection based on IoT cloud. International Journal of Electrical and Computer Engineering, 11(2), 1823.

12. Mukherjee, R., Kundu, A., Mukherjee, I., Gupta, D., Tiwari, P., Khanna, A., & Shorfuzzaman, M. (2021). IoT-cloud based healthcare model for COVID-19 detection: an enhanced k-Nearest Neighbour classifier based approach. Computing, 1-21.

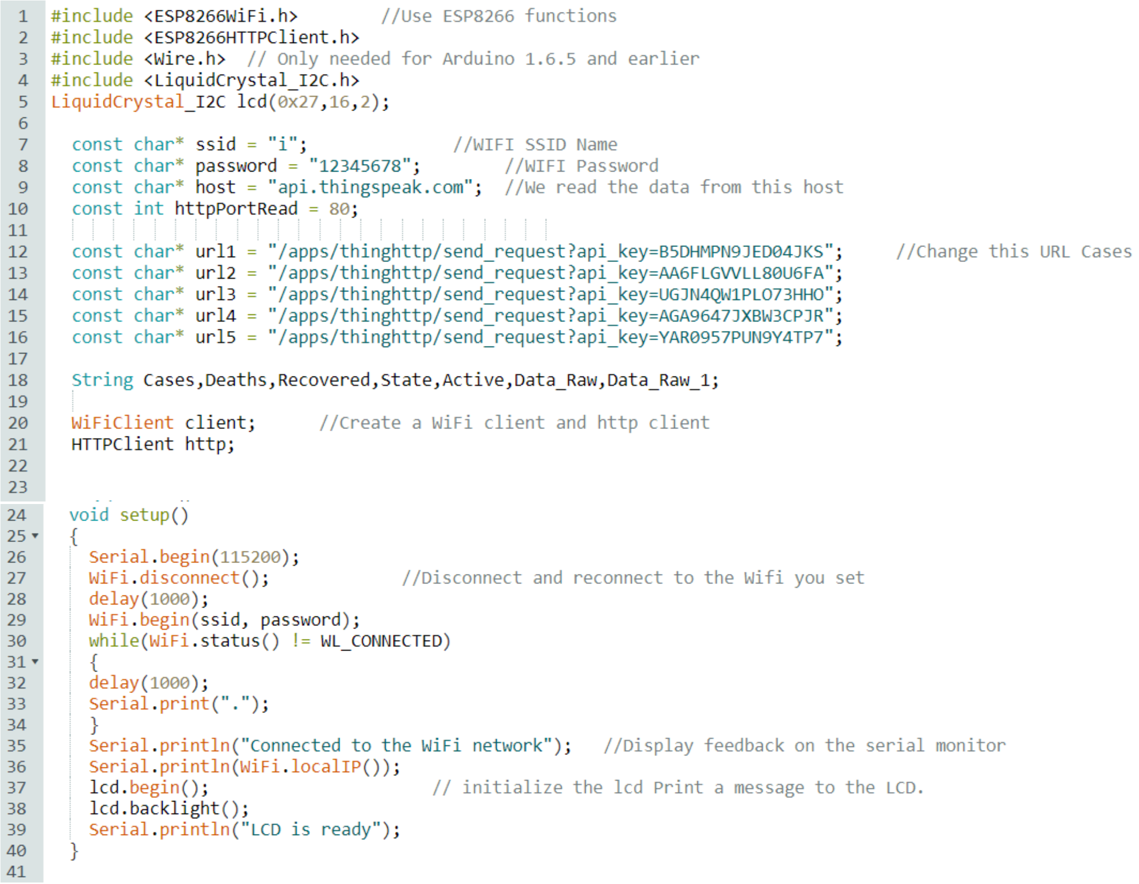
13. Fernandez, L. G. V., Gonçalves, R. T., dos Santos, J. C., Moreira, T. C. P., Nery, J. F. P., de Carvalho, L. M., & de Souza, F. H. B. (2021). Internet of Things versus Covid-19: Integrated Low-Cost Proposal for Oximetry Collection and Data Availability in Cloud for Strategic Management of Population in Isolation. Proceedings of The 3rd International Co.

14. PRIYANKA B. POPHALKAR: IoTBased Smart Patient Monitoring System withEmphasis onCOVID andAssociatedRespiratory Diseases DiagnosisPriyanka BPophalkar1,Vasif AhmedINTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN SCIENCE AND ENGINEERING, VOL.2, NO.6, JUNE 2021.

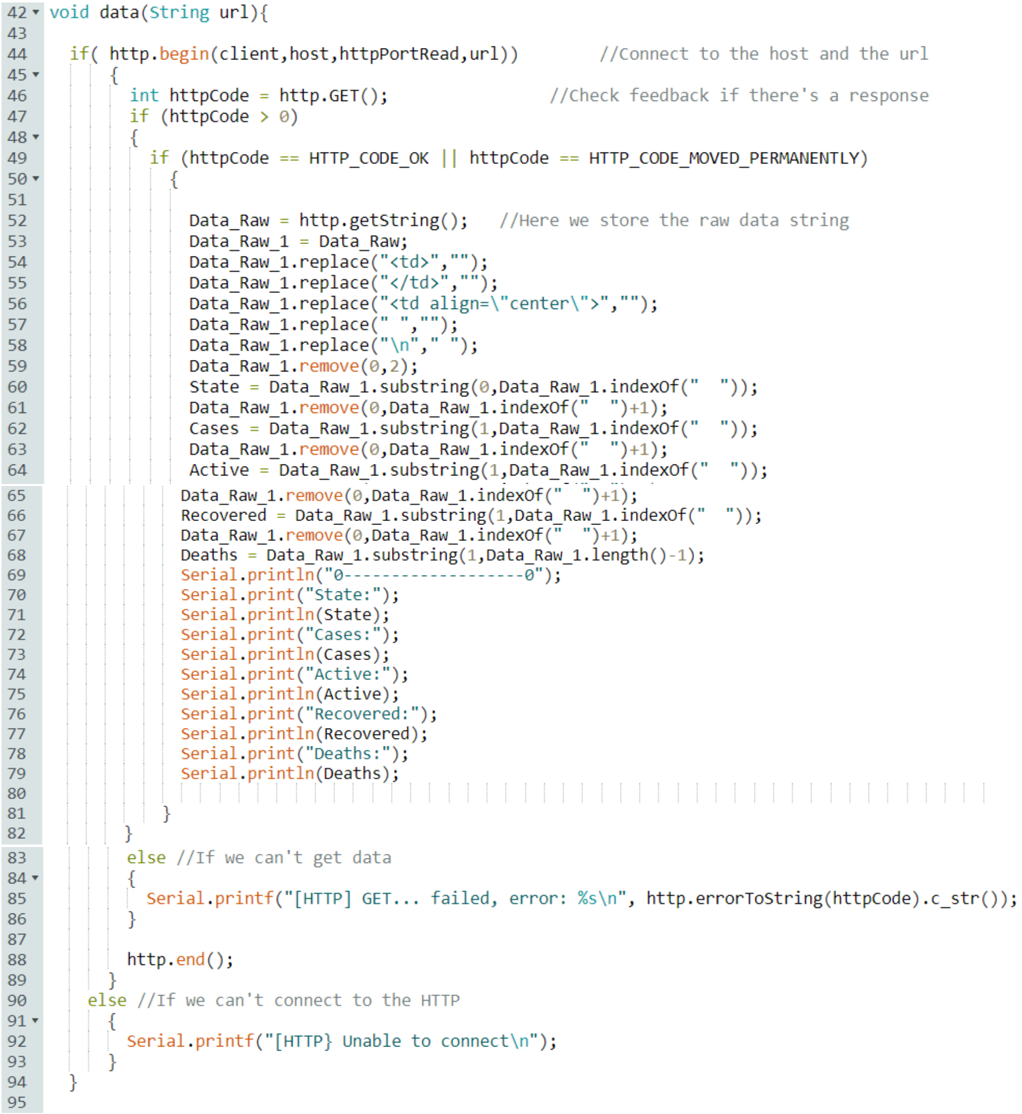
15. D.Dinesh , I.Anette Regina Department of Computer Science, Muthurangam Government Arts College (Autonomous), Vellore:Prediction and Effective Monitoring of Flood Using Arduino System Controller and ESP8266 Wi-Fi Module International Journal of Communication and Networking System Issue: 01, June 2019

Appendix (Code):

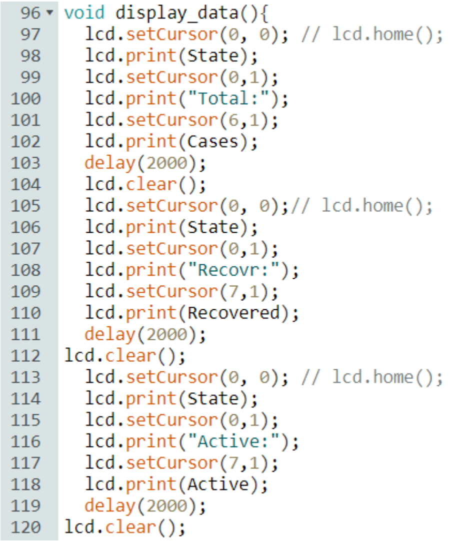
A) COVID Live Tracker



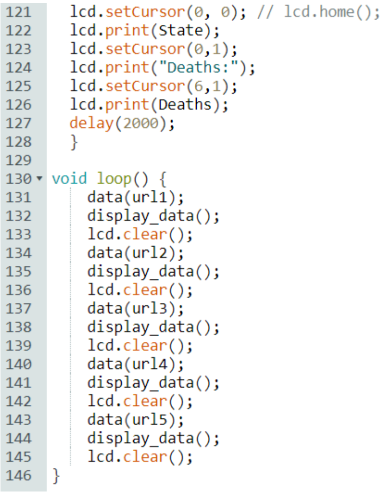
```
#include <ESP8266WiFi.h>        //Use ESP8266 functions
#include <ESP8266HTTPClient.h>
#include <Wire.h>  // Only needed for Arduino 1.6.5 and earlier
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

  const char* ssid = "i";              //WIFI SSID Name
  const char* password = "12345678";        //WIFI Password
  const char* host = "api.thingspeak.com";  //We read the data from this host
  const int httpPortRead = 80;

  const char* url1 = "/apps/thinghttp/send_request?api_key=B5DHMPN9JED04JKS";     //Change this URL Cases
  const char* url2 = "/apps/thinghttp/send_request?api_key=AA6FLGVVLL80U6FA";
  const char* url3 = "/apps/thinghttp/send_request?api_key=UGJN4QW1PLO73HHO";
  const char* url4 = "/apps/thinghttp/send_request?api_key=AGA9647JXBW3CPJR";
  const char* url5 = "/apps/thinghttp/send_request?api_key=YAR0957PUN9Y4TP7";

  String Cases,Deaths,Recovered,State,Active,Data_Raw,Data_Raw_1;

  WiFiClient client;      //Create a WiFi client and http client
  HTTPClient http;


  void setup()
  {
    Serial.begin(115200);
    WiFi.disconnect();             //Disconnect and reconnect to the Wifi you set
    delay(1000);
    WiFi.begin(ssid, password);
    while(WiFi.status() != WL_CONNECTED)
    {
    delay(1000);
    Serial.print(".");
    }
    Serial.println("Connected to the WiFi network");   //Display feedback on the serial monitor
    Serial.println(WiFi.localIP());
    lcd.begin();                   // initialize the lcd Print a message to the LCD.
    lcd.backlight();
    Serial.println("LCD is ready");
  }

```



```cpp
void data(String url){

  if( http.begin(client,host,httpPortRead,url))        //Connect to the host and the url
      {
        int httpCode = http.GET();                  //Check feedback if there's a response
        if (httpCode > 0)
        {
          if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY)
            {

              Data_Raw = http.getString();   //Here we store the raw data string
              Data_Raw_1 = Data_Raw;
              Data_Raw_1.replace("<td>","");
              Data_Raw_1.replace("</td>","");
              Data_Raw_1.replace("<td align=\"center\">","");
              Data_Raw_1.replace(" ","");
              Data_Raw_1.replace("\n"," ");
              Data_Raw_1.remove(0,2);
              State = Data_Raw_1.substring(0,Data_Raw_1.indexOf("  "));
              Data_Raw_1.remove(0,Data_Raw_1.indexOf("  ")+1);
              Cases = Data_Raw_1.substring(1,Data_Raw_1.indexOf("  "));
              Data_Raw_1.remove(0,Data_Raw_1.indexOf("  ")+1);
              Active = Data_Raw_1.substring(1,Data_Raw_1.indexOf("  "));
             Data_Raw_1.remove(0,Data_Raw_1.indexOf("  ")+1);
             Recovered = Data_Raw_1.substring(1,Data_Raw_1.indexOf("  "));
             Data_Raw_1.remove(0,Data_Raw_1.indexOf("  ")+1);
             Deaths = Data_Raw_1.substring(1,Data_Raw_1.length()-1);
             Serial.println("0------------------0");
             Serial.print("State:");
             Serial.println(State);
             Serial.print("Cases:");
             Serial.println(Cases);
             Serial.print("Active:");
             Serial.println(Active);
             Serial.print("Recovered:");
             Serial.println(Recovered);
             Serial.print("Deaths:");
             Serial.println(Deaths);

           }
       }
       else //If we can't get data
       {
         Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
       }

       http.end();
     }
     else //If we can't connect to the HTTP
     {
       Serial.printf("[HTTP} Unable to connect\n");
     }
  }

```

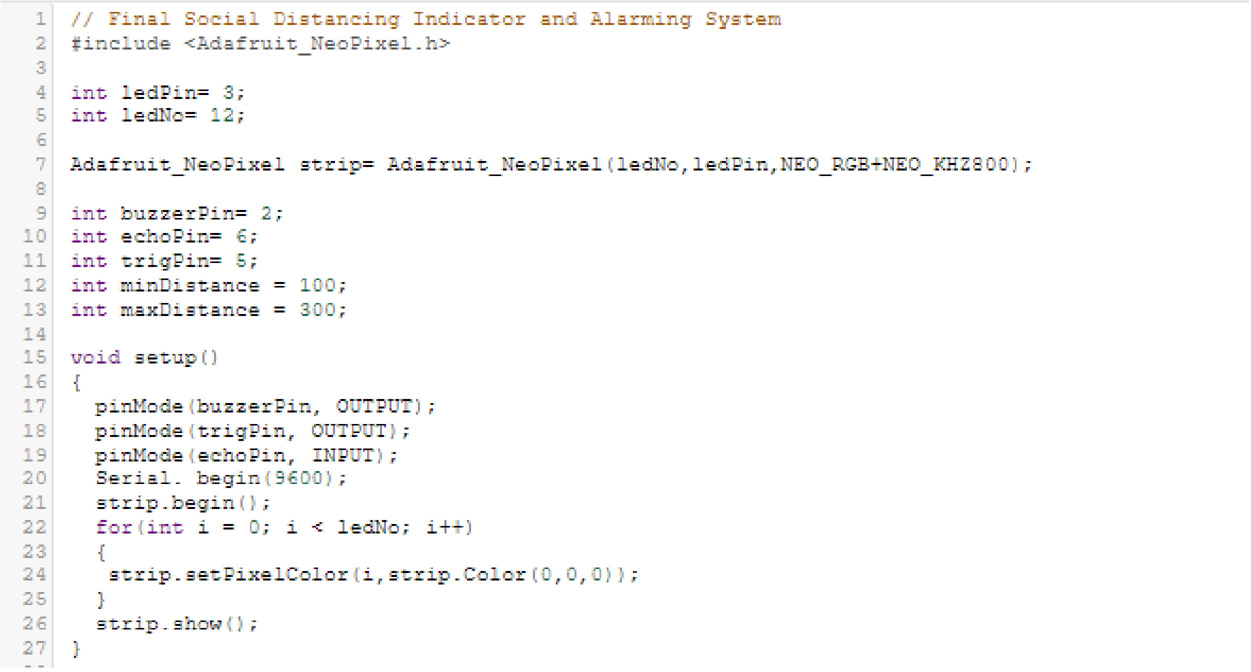


```
void display_data(){
  lcd.setCursor(0, 0); // lcd.home();
  lcd.print(State);
  lcd.setCursor(0,1);
  lcd.print("Total:");
  lcd.setCursor(6,1);
  lcd.print(Cases);
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);// lcd.home();
  lcd.print(State);
  lcd.setCursor(0,1);
  lcd.print("Recovr:");
  lcd.setCursor(7,1);
  lcd.print(Recovered);
  delay(2000);
lcd.clear();
  lcd.setCursor(0, 0); // lcd.home();
  lcd.print(State);
  lcd.setCursor(0,1);
  lcd.print("Active:");
  lcd.setCursor(7,1);
  lcd.print(Active);
  delay(2000);
lcd.clear();
```

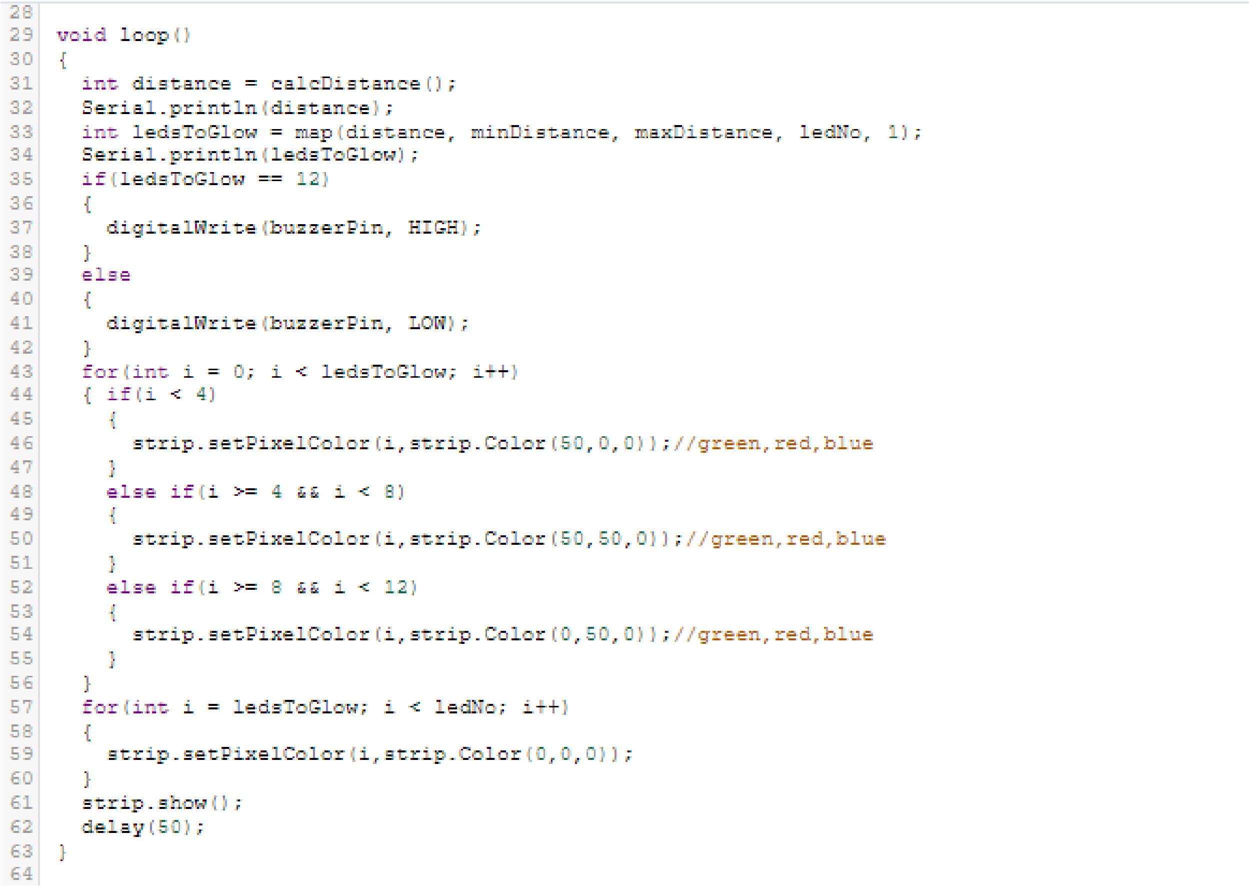


```
 lcd.setCursor(0, 0); // lcd.home();
 lcd.print(State);
 lcd.setCursor(0,1);
 lcd.print("Deaths:");
 lcd.setCursor(6,1);
 lcd.print(Deaths);
 delay(2000);
 }

void loop() {
    data(url1);
    display_data();
    lcd.clear();
    data(url2);
    display_data();
    lcd.clear();
    data(url3);
    display_data();
    lcd.clear();
    data(url4);
    display_data();
    lcd.clear();
    data(url5);
    display_data();
    lcd.clear();
}
```

B) Social Distance Detector:



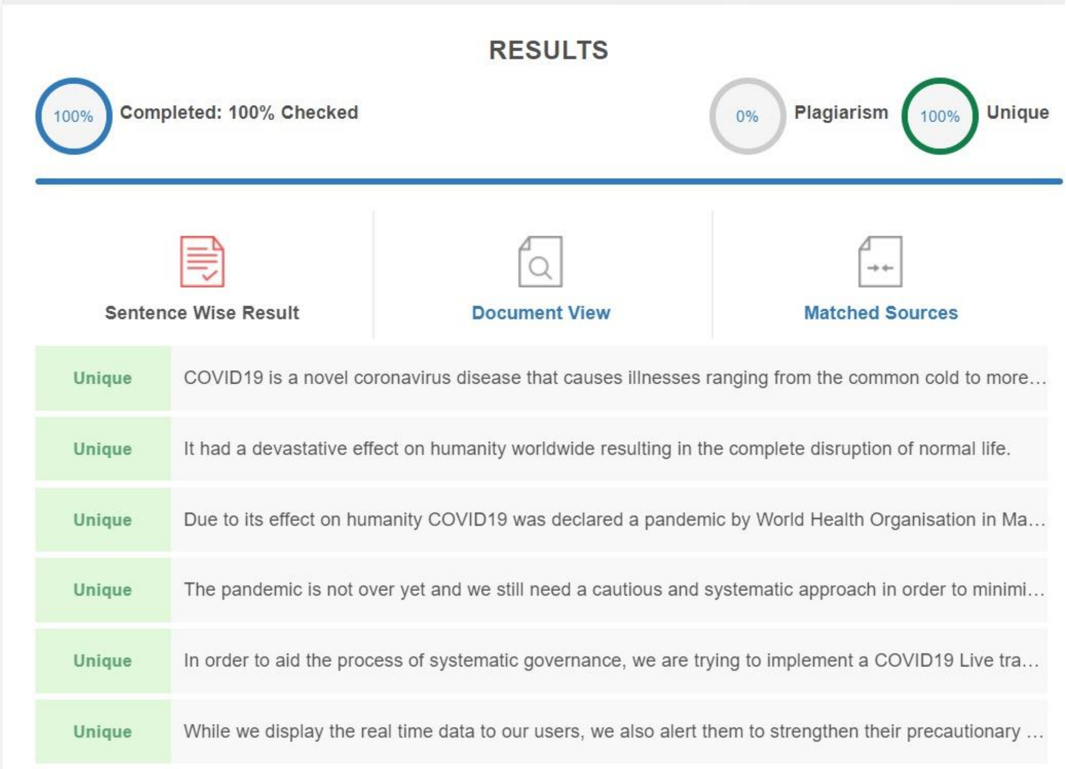
```
// Final Social Distancing Indicator and Alarming System
#include <Adafruit_NeoPixel.h>

int ledPin= 3;
int ledNo= 12;

Adafruit_NeoPixel strip= Adafruit_NeoPixel(ledNo,ledPin,NEO_RGB+NEO_KHZ800);

int buzzerPin= 2;
int echoPin= 6;
int trigPin= 5;
int minDistance = 100;
int maxDistance = 300;

void setup()
{
  pinMode(buzzerPin, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial. begin(9600);
  strip.begin();
  for(int i = 0; i < ledNo; i++)
  {
   strip.setPixelColor(i,strip.Color(0,0,0));
  }
  strip.show();
}
```



```

void loop()
{
  int distance = calcDistance();
  Serial.println(distance);
  int ledsToGlow = map(distance, minDistance, maxDistance, ledNo, 1);
  Serial.println(ledsToGlow);
  if(ledsToGlow == 12)
  {
    digitalWrite(buzzerPin, HIGH);
  }
  else
  {
    digitalWrite(buzzerPin, LOW);
  }
  for(int i = 0; i < ledsToGlow; i++)
  { if(i < 4)
    {
      strip.setPixelColor(i,strip.Color(50,0,0));//green,red,blue
    }
    else if(i >= 4 && i < 8)
    {
      strip.setPixelColor(i,strip.Color(50,50,0));//green,red,blue
    }
    else if(i >= 8 && i < 12)
    {
      strip.setPixelColor(i,strip.Color(0,50,0));//green,red,blue
    }
  }
  for(int i = ledsToGlow; i < ledNo; i++)
  {
    strip.setPixelColor(i,strip.Color(0,0,0));
  }
  strip.show();
  delay(50);
}

```

Plagarism report

Abstract:



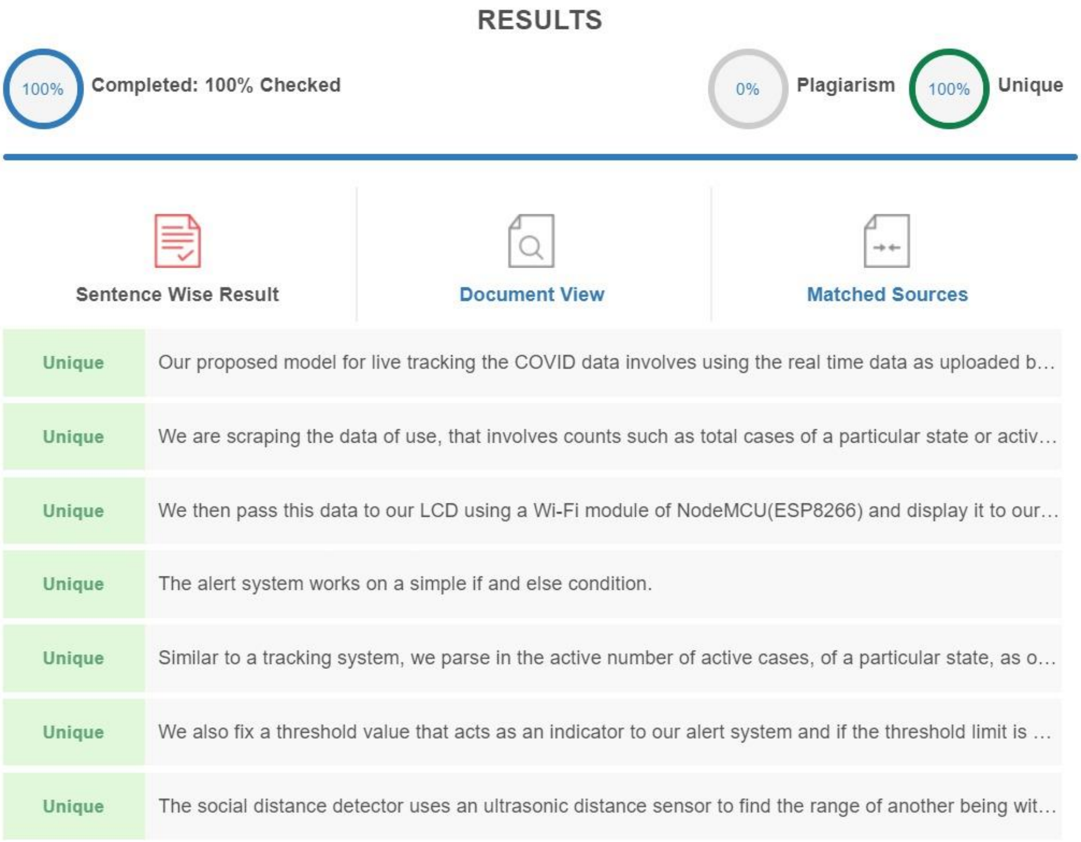
RESULTS

100% Completed: 100% Checked

0% Plagiarism 100% Unique

Unique	COVID19 is a novel coronavirus disease that causes illnesses ranging from the common cold to more…
Unique	It had a devastative effect on humanity worldwide resulting in the complete disruption of normal life.
Unique	Due to its effect on humanity COVID19 was declared a pandemic by World Health Organisation in Ma…
Unique	The pandemic is not over yet and we still need a cautious and systematic approach in order to minimi…
Unique	In order to aid the process of systematic governance, we are trying to implement a COVID19 Live tra…
Unique	While we display the real time data to our users, we also alert them to strengthen their precautionary …

Introduction:



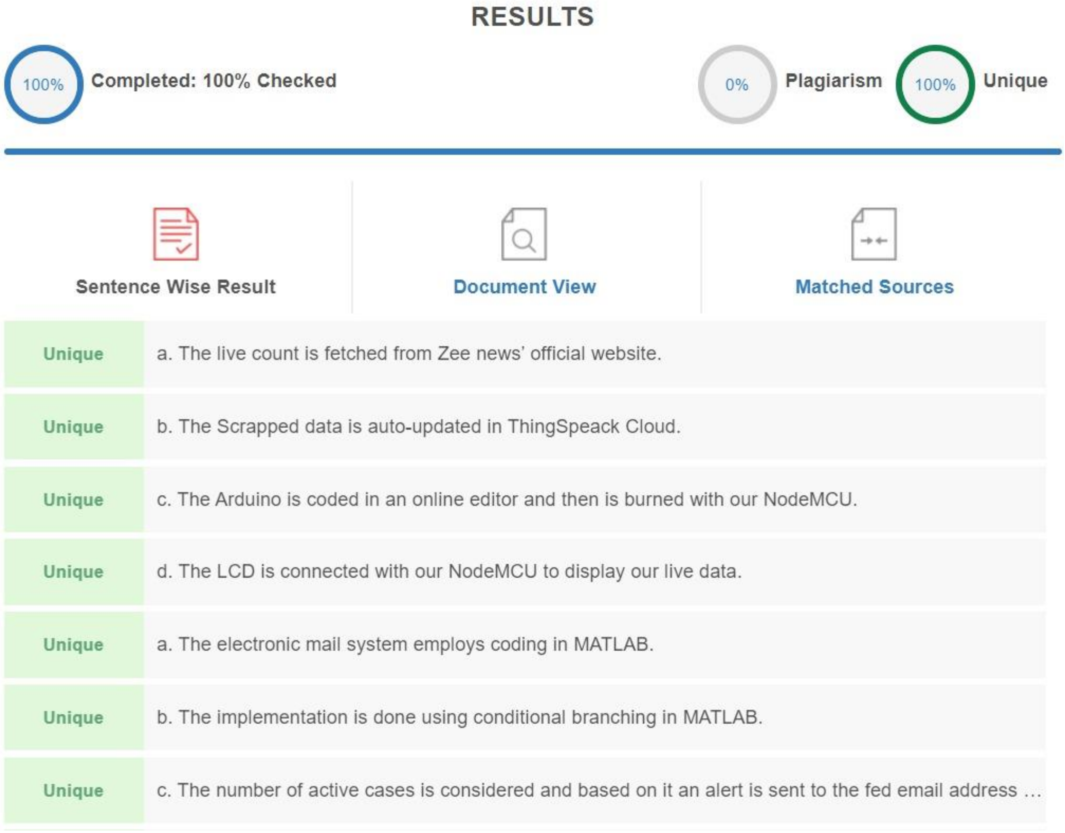
RESULTS

100% Completed: 100% Checked

0% Plagiarism 100% Unique

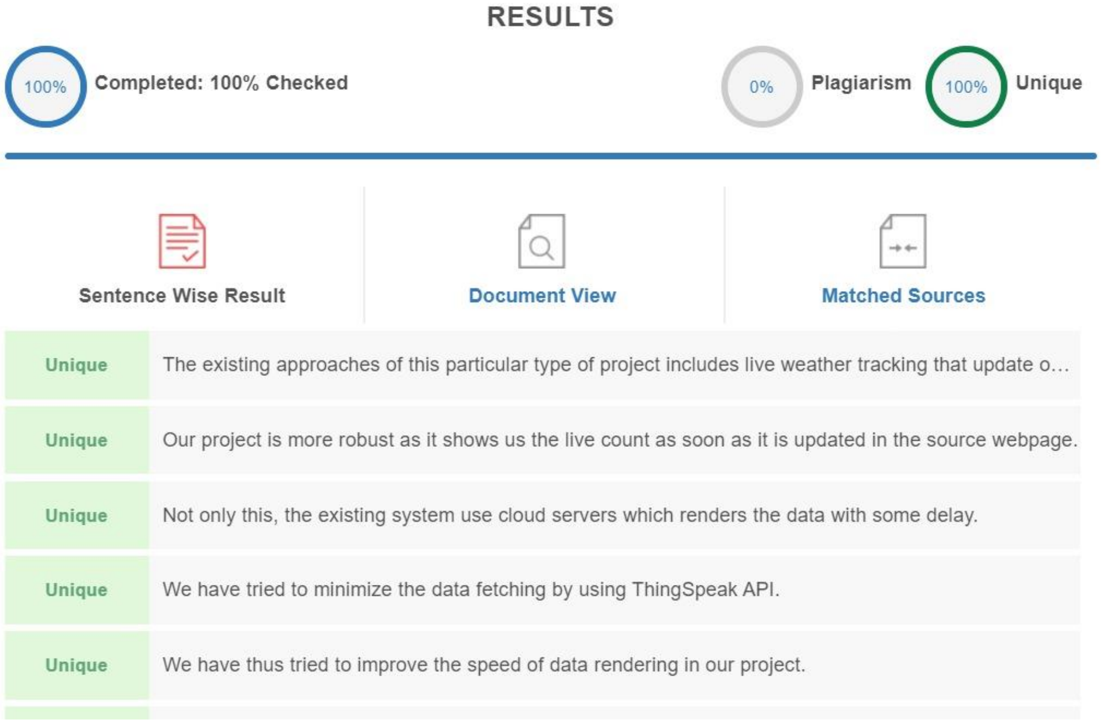
Unique	Our proposed model for live tracking the COVID data involves using the real time data as uploaded b…
Unique	We are scraping the data of use, that involves counts such as total cases of a particular state or activ…
Unique	We then pass this data to our LCD using a Wi-Fi module of NodeMCU(ESP8266) and display it to our…
Unique	The alert system works on a simple if and else condition.
Unique	Similar to a tracking system, we parse in the active number of active cases, of a particular state, as o…
Unique	We also fix a threshold value that acts as an indicator to our alert system and if the threshold limit is …
Unique	The social distance detector uses an ultrasonic distance sensor to find the range of another being wit…

Implementation



Unique	a. The live count is fetched from Zee news' official website.
Unique	b. The Scrapped data is auto-updated in ThingSpeack Cloud.
Unique	c. The Arduino is coded in an online editor and then is burned with our NodeMCU.
Unique	d. The LCD is connected with our NodeMCU to display our live data.
Unique	a. The electronic mail system employs coding in MATLAB.
Unique	b. The implementation is done using conditional branching in MATLAB.
Unique	c. The number of active cases is considered and based on it an alert is sent to the fed email address …

Drawbacks and Novelity



Unique	The existing approaches of this particular type of project includes live weather tracking that update o…
Unique	Our project is more robust as it shows us the live count as soon as it is updated in the source webpage.
Unique	Not only this, the existing system use cloud servers which renders the data with some delay.
Unique	We have tried to minimize the data fetching by using ThingSpeak API.
Unique	We have thus tried to improve the speed of data rendering in our project.