GOVERNMENT COLLEGE OF ENGINEERING ERODE

PROJECT: FLOOD MONITORING AND EARLY WARNING USING BOLT IOT

TEAM MEMBERS	ROLLNO
1.PREETHA S	au731121106036
2.SOWMIYA R	au731121106045
3.MANJARI M	au731121106030
4.SHAHANA V	au731121106044

INTRODUCTION:

Flood is one of the natural disasters that occurs every year . It destroys the infrastructure and causes fatalities. Flood Monitoring and Early Warning System (FMEWS) is a project designed to mitigate the impact of floods. The system relies on various data sources, including weather forecasts, river gauges, rainfall sensors, and historical flood data. Unlike the existing systems, this project intends to develop a more robust and durable system which can withstand the wet weather condition. In order to do this, the system needs to have the basic information such as water conditions, water level and precipitation level to detect the increase of water level during flood.

OBJECTIVES:

A smart computer system for the exploitation of hydrometeorological and weather data captured to generate warnings and notifications for events that may involve a flood risk situation.

- To develop a sensor network device for sensing water level.
- To develop processing and transmission units using GSM.
- To develop algorithms and criteria for triggering early warnings when flood risks are detected.

The ultimate goal of flood warning systems is to minimize flood damages and provide reliable information for flood risk reduction measures, policy making, and agricultural insurance.

IOT SENSOR DEPLOYMENT:

Ultrasonic level sensors are used to detect and range objects in air with high accuracy. A wireless alert is sent if the measured distance is over or under the configured thresholds.

Deploying an IoT system for an ultrasonic sensor involves several steps:

Sensor Selection: An appropriating Ultrasonic sensor is to be chosen by considering factors like range, accuracy, and connectivity options (e.g., Wi-Fi, LoRa, Bluetooth).

Microcontroller: A microcontroller board is selected (e.g., Arduino, Raspberry Pi, ESP8266/ESP32) to interface with the ultrasonic sensor and manage IoT communication.

Power Supply: Ensure a stable power source for both the sensor and microcontroller, which may include batteries, solar panels, or AC power.

Connectivity: Microcontroller board is connected to the internet via Wi-Fi, cellular, or other appropriate means.

Data Processing: Microcontroller is programmed to read data from the ultrasonic sensor and process it as needed (e.g., distance measurement) before sending it to the cloud.

Cloud Platform: An IOT platform is chosen (e.g., AWS IoT, Azure IoT, Google Cloud IoT) to securely manage and store sensor data.

Data Transmission: The code is impelemented to transmit sensor data to the cloud by using the chosen communication protocol and data transmission with appropriate encryption and authentication will be secured.

Alerts and Automation: Set up alerts or triggers based on sensor data thresholds and implement automation rules if necessary.

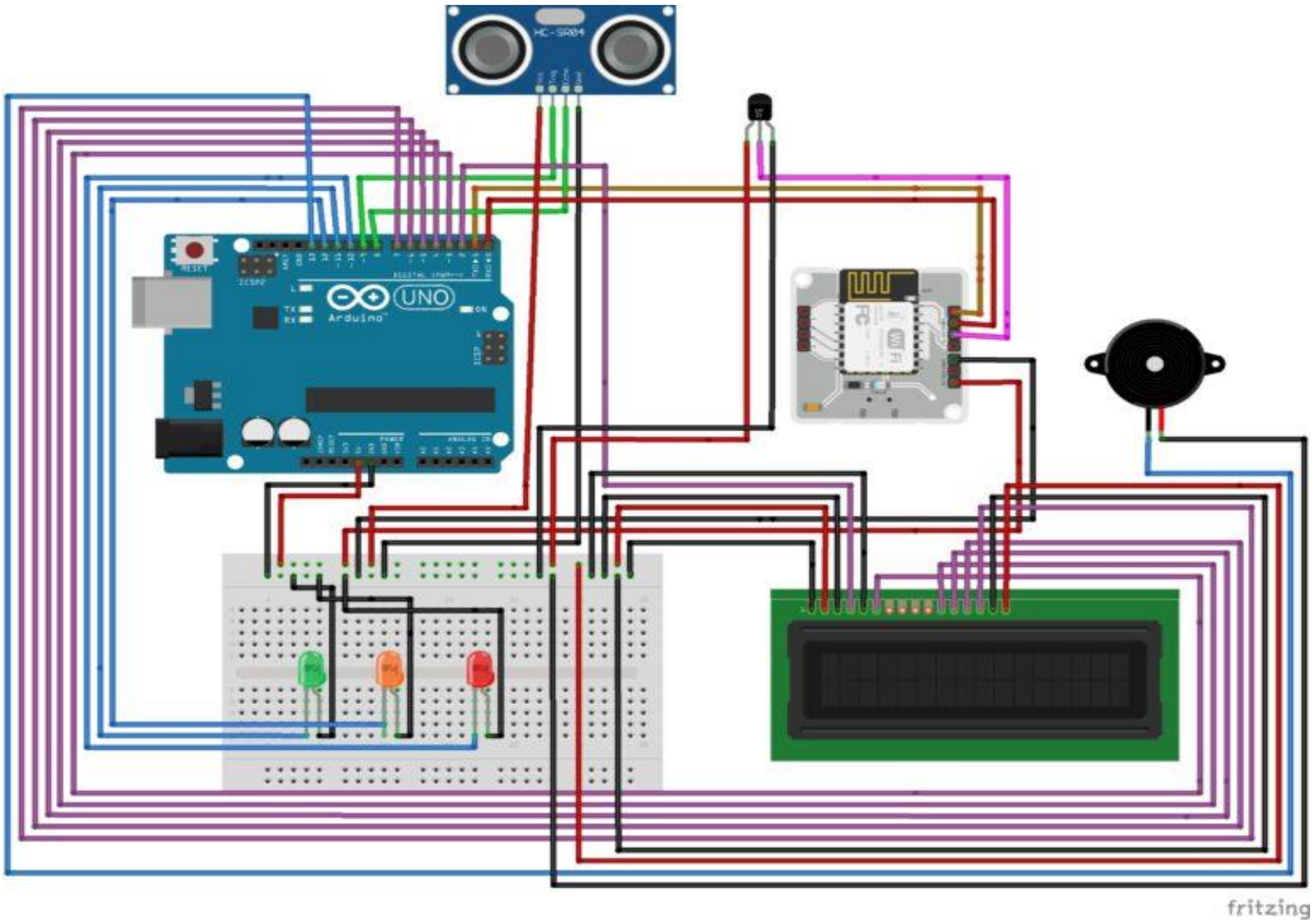
Testing and Calibration: Thoroughly test the system to ensure accurate sensor readings and proper data transmission. Calibrate the sensor if necessary.

PLATFORM DEVELOPMENT:

Steps to develop a platform for flood monitoring system:

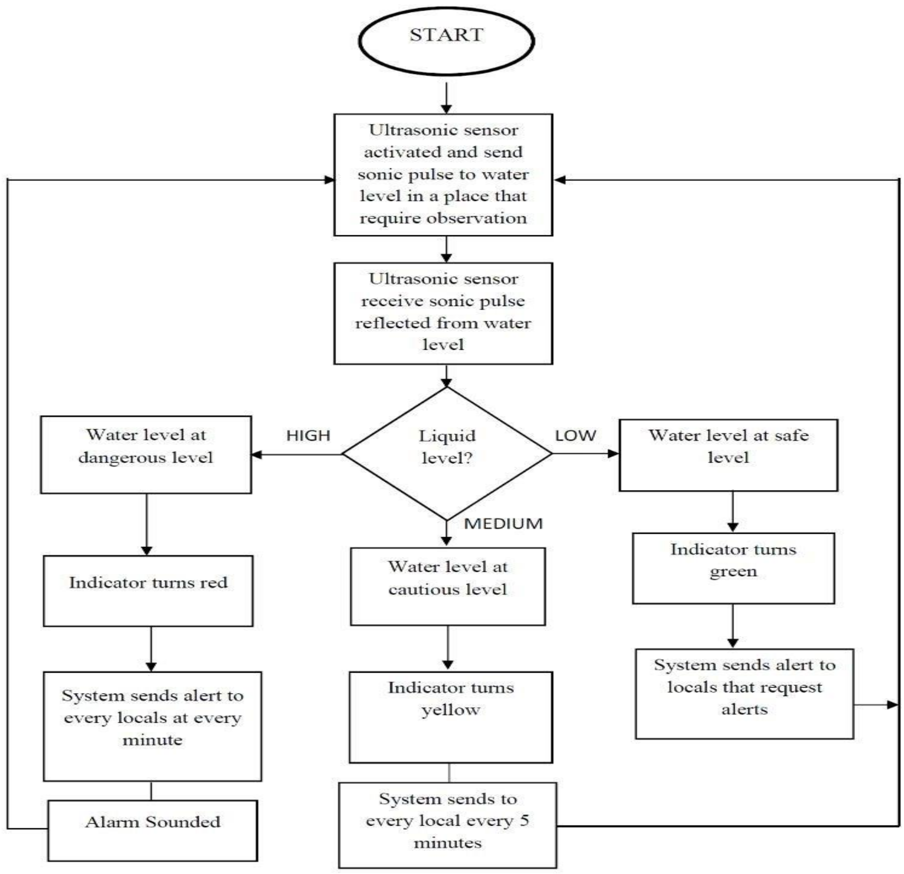
- Sensor deployment
- Data collection and aggregation
- Data analysis and modelling
- Communication infrastructure
- Early warning system
- Geo spatial information
- Public outreach
- Collaboration with authorities
- Testing and validation

SYSTEMATIC DIAGRAM:

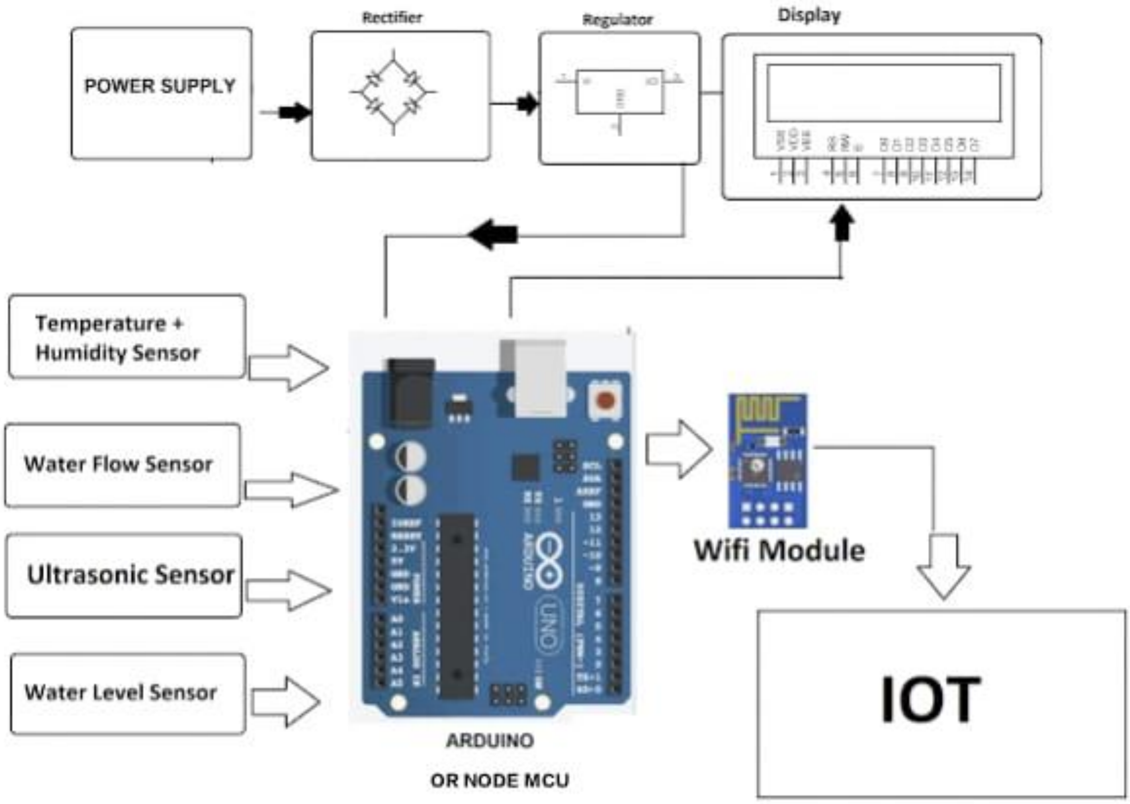


FLOW CHART:

Based on flowchart , Smart IoT Flood Monitoring System is developed to alert the public closest to the area when there is upcoming flood. The process is starting when ultrasonic sensor measures level of water in the river. The collected data from the sensor are gathered and will be forwarded to microcontroller and data will be displayed at web server. Then, data will be analysed and compared. A user can control the stepper motor and buzzer wirelessly. Flood status dangerous will be determined based on that collected data. Thus, water level status will display on LCD and web server. LED will be turn on to indicate the water level. Furthermore, the stepper motor will be turn on for the passage of excessive flood when it reached at the highest threshold value and the alarm will be triggered immediately to alert the public. Hence, the citizens will be well prepared for evacuation before the flood.



BLOCK DIAGRAM:



From the block diagram , Ultrasonic HC-SR04 module sensor is used to detect the water level of river. The mbed NXP LPC1768 is a microcontroller that collects all the data in this system. Buzzer is acting as an alarm to alert the public and authority when there is upcoming flood and data will be updated to web server. The function of LCD and LED are to display and indicate the level of water.

CODE IMPLEMENTATION:

```
//Early Flood Detection Using IOT ~ A project by Sabyasachi Ghosh

//<LiquidCrystal.h> is the library for using the LCD 16x2

#include <LiquidCrystal.h>


LiquidCrystal lcd(2, 3, 4, 5, 6, 7);  // Create an instance of the LiquidCrystal library

const int in = 8;                       // This is the ECHO pin of The Ultrasonic sensor HC-SR04

const int out = 9;                       // This is the TRIG pin of the ultrasonic Sensor HC-SR04

// Define pin numbers for various components

const int green = 10;

const int orange = 11;

const int red = 12;

const int buzz = 13;


void setup()

{

 // Start serial communication with a baud rate of 9600

 Serial.begin(9600);

 // Initialize the LCD with 16 columns and 2 rows

 lcd.begin(16, 2);

 // Set pin modes for various components

 pinMode(in, INPUT);

 pinMode(out, OUTPUT);

 pinMode(green, OUTPUT);

 pinMode(orange, OUTPUT);

 pinMode(red, OUTPUT);

 pinMode(buzz, OUTPUT);

 // Display a startup message on the LCD

 lcd.setCursor(0, 0);

 lcd.print("Flood Monitoring");
```

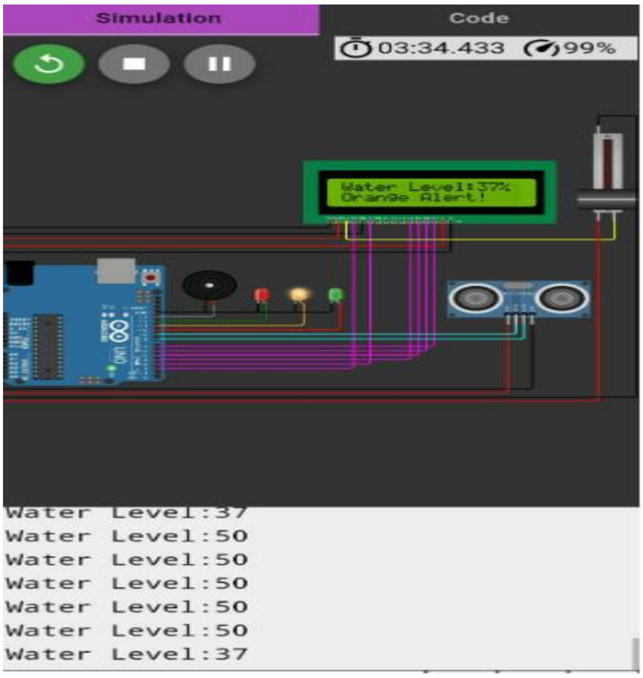
```
  lcd.setCursor(0, 1);
  lcd.print("Alerting System");
  // Wait for 5 seconds and then clear the LCD
  delay(5000);
  lcd.clear();
}

void loop()
{
  // Read distance from the ultrasonic sensor (HC-SR04)
  long dur;
  long dist;
  long per;
  digitalWrite(out, LOW);
  delayMicroseconds(2);
  digitalWrite(out, HIGH);
  delayMicroseconds(10);
  digitalWrite(out, LOW);
  dur = pulseIn(in, HIGH);
  dist = (dur * 0.034) / 2;
  // Map the distance value to a percentage value
  per = map(dist, 10.5, 2, 0, 100);
  // Ensure that the percentage value is within bounds
  if (per < 0)
  {
    per = 0;
  }
  if (per > 100)
  {
    per = 100;
  }
```

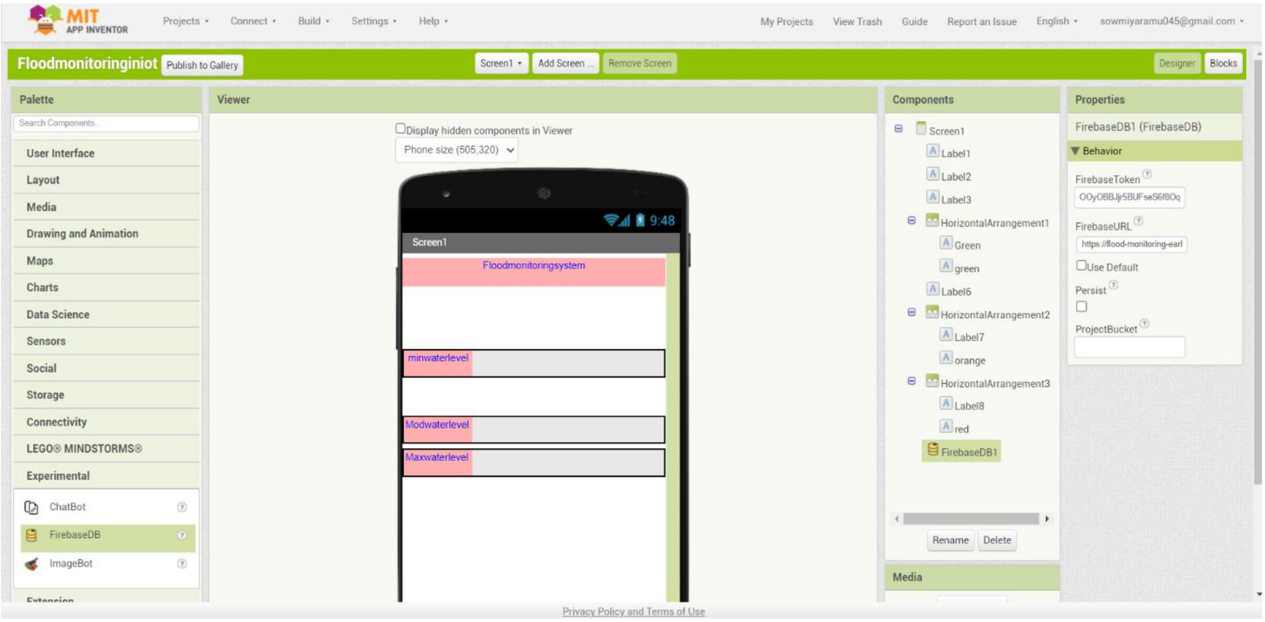
```
// Print water level data to serial
Serial.print("Water Level:");
Serial.println(String(per));
lcd.setCursor(0, 0);
lcd.print("Water Level:");
lcd.print(String(per));
lcd.print("%  ");
// Check water level and set alert levels
if (dist <= 3)
{
  lcd.setCursor(0, 1);
  lcd.print("Red Alert!   ");
  digitalWrite(red, HIGH);
  digitalWrite(green, LOW);
  digitalWrite(orange, LOW);
  digitalWrite(buzz, HIGH);
  delay(2000);
  digitalWrite(buzz, LOW);
  delay(2000);
  digitalWrite(buzz, HIGH);
  delay(2000);
  digitalWrite(buzz, LOW);
  delay(2000);
}
else if (dist <= 10)
{
  lcd.setCursor(0, 1);
  lcd.print("Orange Alert!  ");
  digitalWrite(orange, HIGH);
  digitalWrite(red, LOW);
  digitalWrite(green, LOW);
```

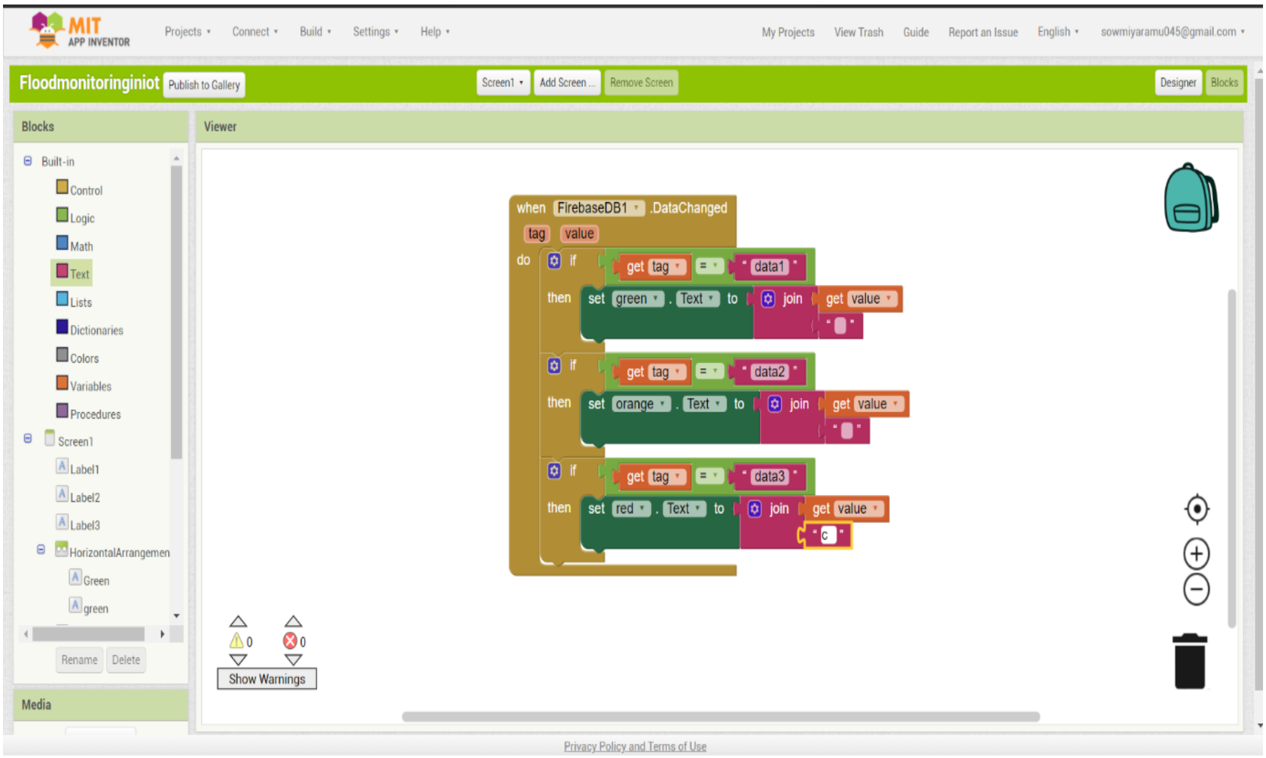
```
  digitalWrite(buzz, HIGH);
  delay(3000);
  digitalWrite(buzz, LOW);
  delay(3000);
 }
 else
 {
  lcd.setCursor(0, 1);
  lcd.print("Green Alert!  ");
  digitalWrite(green, HIGH);
  digitalWrite(orange, LOW);
  digitalWrite(red, LOW);
  digitalWrite(buzz, LOW);
 }
}
```

OUTPUT:



FIREBASE SETUP AND APP DEVELOPMENT:





ENHANCEMENT FOR PUBLIC SAFETY:

The ultimate goal of flood warning systems is to minimize flood damages and provide reliable information for flood risk reduction measures, policy making, and agricultural insurance. issuing early warning alerts to the public and relevant authorities when there's a high probability of flooding. These alerts can be delivered through various channels like mobile apps, SMS, sirens, and broadcast media.

CONCLUSION:

By integrating sensor data collection into flood monitoring and early warning systems, it is possible to provide timely and accurate information to help mitigate the impact of floods and protect lives and property.

Cloud Platform: Choose an IoT platform (e.g., AWS IoT, Azure IoT, Google Cloud IoT) to securely manage and store sensor data. Set up a device registry, topic, or equivalent.

Data Transmission: Implement code to transmit sensor data to the cloud using the chosen communication protocol. Secure the data transmission with appropriate encryption and authentication.

Data Storage: Configure cloud services to store incoming data and set up a database or data storage solution for historical records.

Remote Monitoring: Develop a dashboard or application to visualize and monitor the data in real-time, allowing users to access sensor readings from anywhere.

Alerts and Automation: Set up alerts or triggers based on sensor data thresholds and implement automation rules if necessary.

Power Management: Implement power-saving measures to optimize energy usage, especially if using battery power.

Security: Ensure the IoT system is secure by following best practices, such as firmware updates, secure authentication, and access control.

Scalability: Consider the scalability of your IoT deployment, especially if you plan to add more sensors or expand the system in the future.

Testing and Calibration: Thoroughly test the system to ensure accurate sensor readings and proper data transmission. Calibrate the sensor if necessary.

Maintenance and Support: Plan for ongoing maintenance, updates, and user support as part of your IoT deployment strategy.

IoT Ultrasonic Sensor Deployment

Default (GPT-3.5)

Deploying an IoT system for an ultrasonic sensor involves several steps:

Sensor Selection: Choose an appropriate ultrasonic sensor for your application, considering factors like range, accuracy, and connectivity options (e.g., Wi-Fi, LoRa, Bluetooth).