In this we have used ThingSpeak Software as data sharing platform for our smart water management IOT project.

ThingSpeak:

ThingSpeak is an IoT analytics platform service that allows you to collect, store, analyse, visualize, and act on data from sensors or other devices.

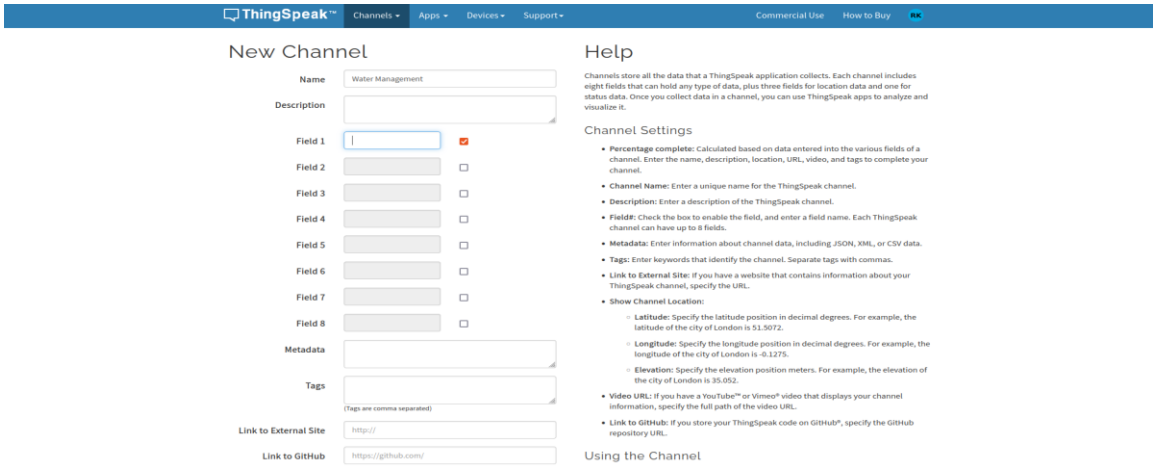
Key Features:

- Real-time data collection.
- Ability to store and retrieve historical data.
- Integration with MATLAB for advanced analytics.
- Support for MQTT and REST APIs for data exchange.

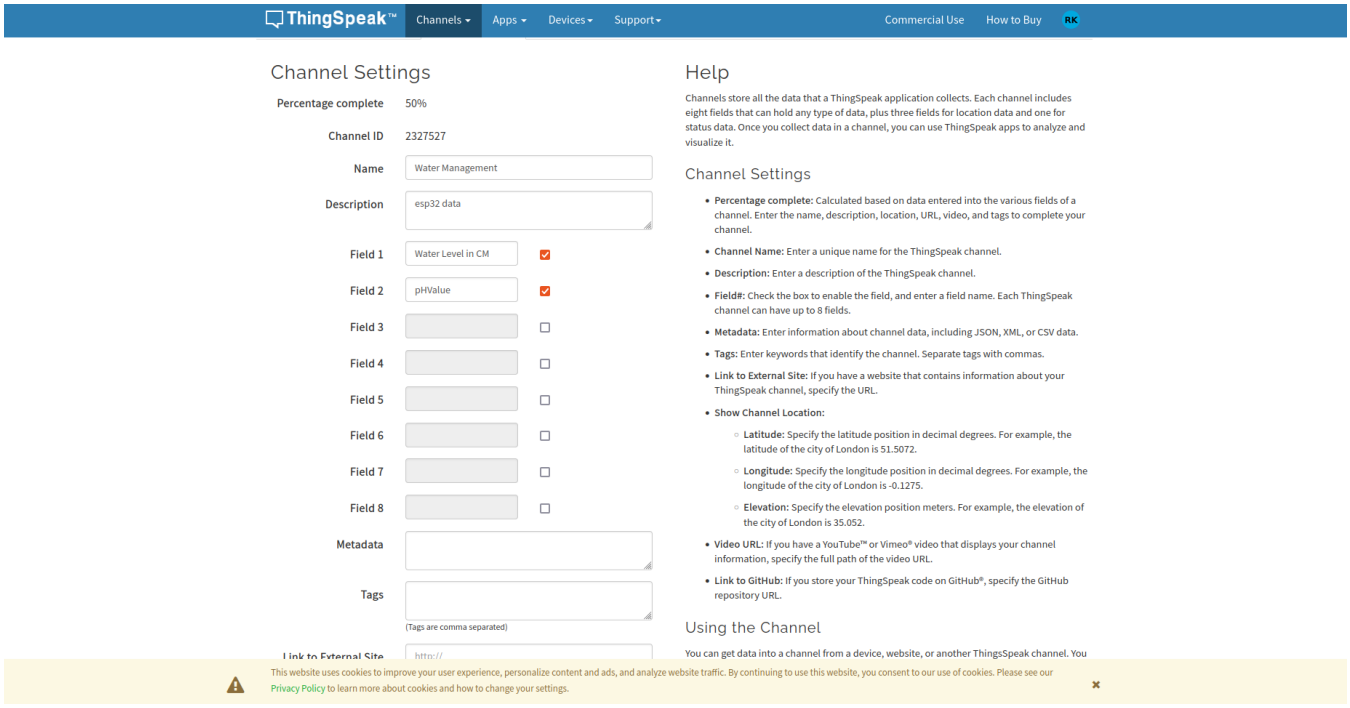
Account Creation: Start by creating a ThingSpeak account on the official website.

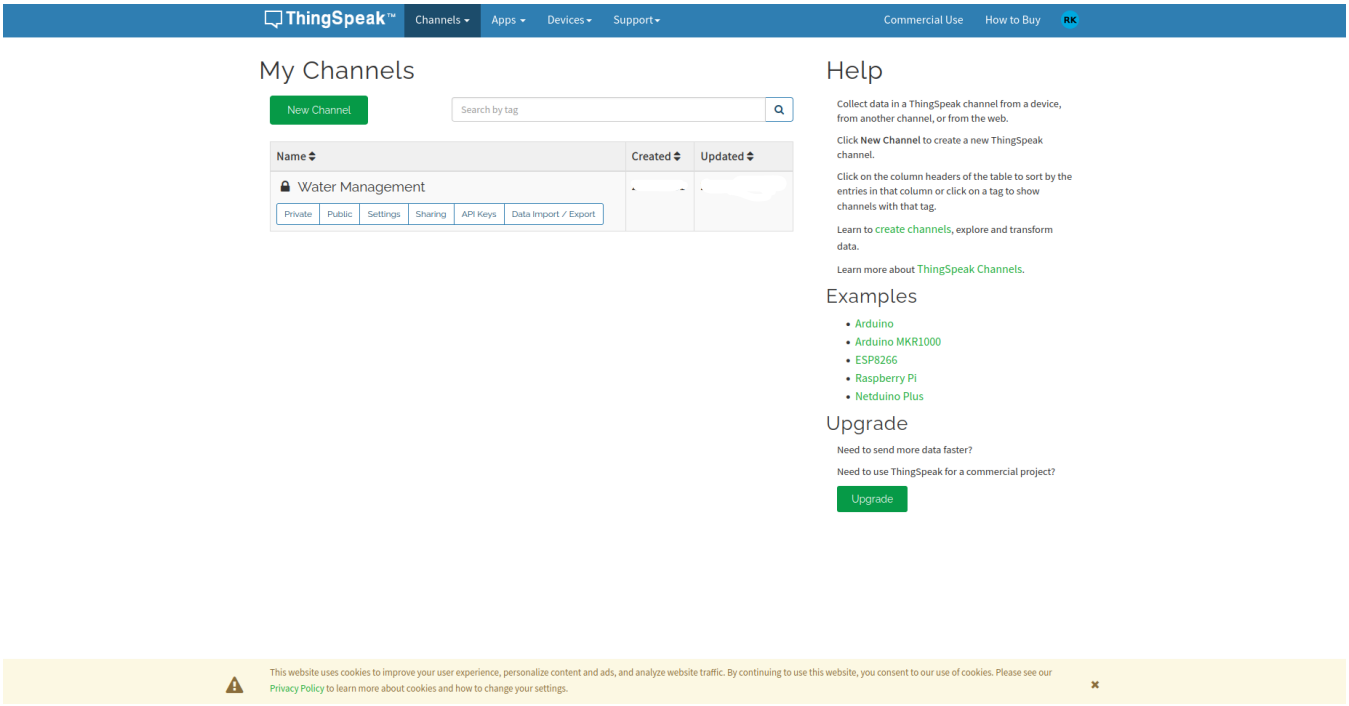
Create a Channel: In ThingSpeak, data is organized into channels. Create a new channel for each type of data you

want to collect.



Fill in the Field What should we needed, Here we use Water Level in CM and pHvalue





Setting Up IoT Devices:

1)**Choose Your Platform**: ThingSpeak is compatible with a variety of IoT platforms such as Arduino, Raspberry Pi, ESP8266, and others.

2)**Install Libraries**: Depending on your IoT platform, install the necessary libraries to enable communication with ThingSpeak.

Sending Data to ThingSpeak:

1)**Generate API Keys**: Create API keys for your ThingSpeak

account to authorize your devices.

2)**Configure IoT Devices**: Set up your devices to send data to ThingSpeak using the provided API keys.

3)**Use REST API or MQTT**: ThingSpeak supports both RESTful API and MQTT for data communication.

Visualizing Data:

1)**Channel Views**: ThingSpeak provides customizable channel views where you can see your data in tabular or graphical formats.

2)**MATLAB Visualizations**: Advanced users can leverage MATLAB Visualizations for more sophisticated data analysis and plotting.

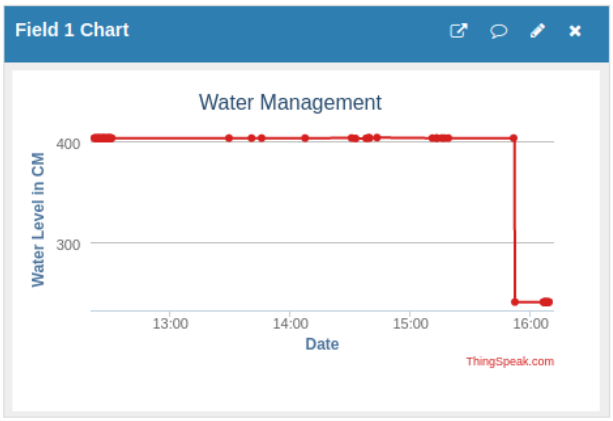
Program:

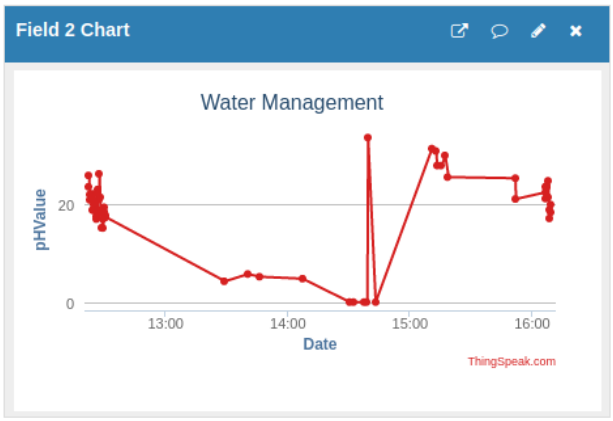
```
#include <Wire.h>
#include <WiFi.h>
#include <ThingSpeak.h>


const char *ssid = "Wokwi-GUEST";
const char *password = "";


#define TRIG_PIN 26
#define ECHO_PIN 25
#define POTENTIOMETER_PIN A0
const float contaminationThreshold = 7.0;


unsigned long channelID = 2327527; // Use Your ThingSpeak Channel ID
const char *writeAPIKey = "3K1OPPG28L2BIA3E"; // Use Your ThingSpeak Write API Key
```

```
WiFiClient client;


void setup() {
Serial.begin(115200);
connectToWiFi();
ThingSpeak.begin(client);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
}


void loop() {
float distance = readUltrasonicDistance();
float pHValue = analogRead(POTENTIOMETER_PIN) / 100.0;
Serial.print("Ultrasonic Distance: ");
Serial.print(distance);
Serial.println(" cm");


Serial.print("pH Value: ");
Serial.println(pHValue);


if (pHValue < contaminationThreshold) {
Serial.println("Water is contaminated!");
} else {
Serial.println("Water is clean.");
}


ThingSpeak.setField(1, distance);
ThingSpeak.setField(2, pHValue);


int updateSuccess = ThingSpeak.writeFields(channelID, writeAPIKey);


if (updateSuccess) {
Serial.println("ThingSpeak update successful");
} else {
Serial.println("Error updating ThingSpeak");
}


delay(2000);
```

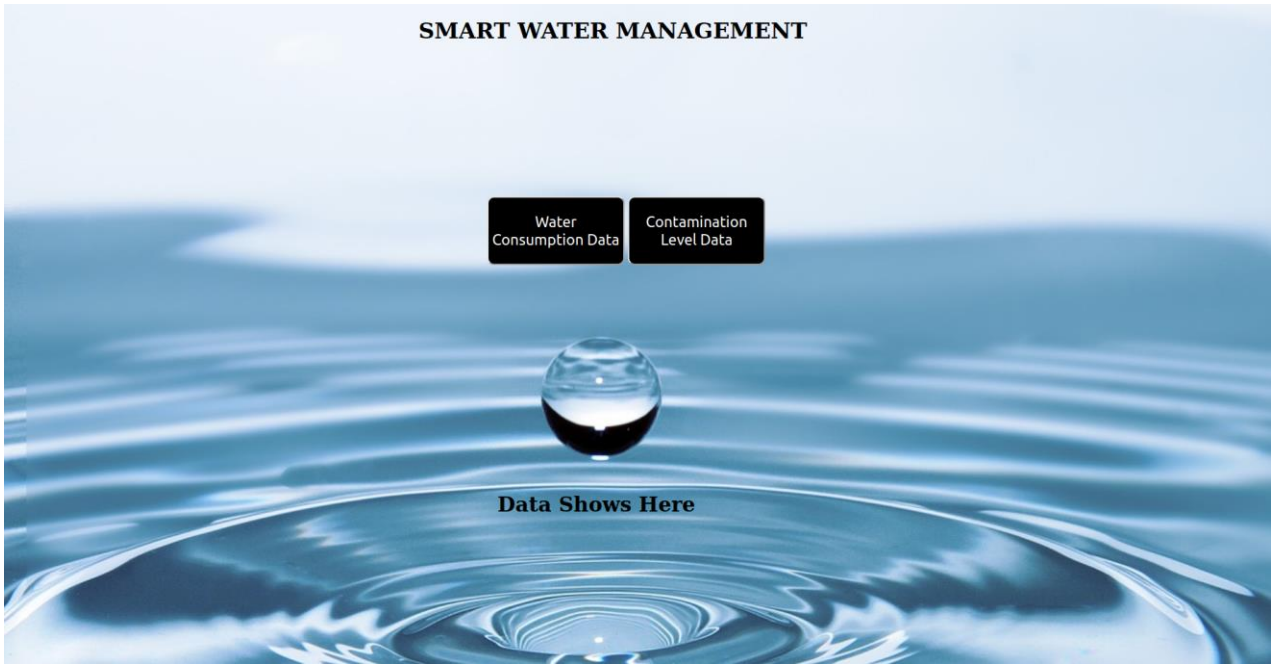
```
}

void connectToWiFi() {
Serial.print("Connecting to WiFi");
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
delay(1000);
Serial.print(".");
}

Serial.println("\nConnected to WiFi");
}

float readUltrasonicDistance() {
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);

return pulseIn(ECHO_PIN, HIGH) * 0.0343 / 2;
}
```

Output:





Website for viewing the real time data for smart water management:



If we click the Water Consumption Data Button then it will show the output where it says Data Shows Here:

If we click the Contamination Level Data Button then it will show the output where it says Data Shows Here:


SMART WATER MANAGEMENT
Water Consumption Data
Contamination Level Data
Contamination Level: 19.82000, Status: Contaminated