College name: JP COLLEGE OF ENGINEERING
College code: 9512

Project ID :Proj_211932_Team_2

TEAM MEMBERS:

1. GEETHA.N (au951221106011)
2. LOGA SRI.N (au951221106018)
3. PRATHIKA.T (au951221106029)
4. JENIFER.A (au9512211060302)
5. SUBHA.I (au951221106049)

Phase 4 : Development Part 2

Topic:Smart Water Management

Front-End:

1. User Interface (UI): Create a user-friendly web interface using HTML, CSS, and JavaScript to display the real-time data. Consider using a framework like React or Angular for a more interactive UI.

2. Dashboard: Design a dashboard where users can view the level of water in a tank. You can use charts or graphs to present the information visually.

3. Real-time Updates: Implement WebSocket technology to enable real-time updates without the need for constant page refresh. Libraries like Socket.io can help with this.

Back-End:

4. IoT Devices: Set up IoT devices (e.g., sensors) to collect level of water. These devices should send data to your platform periodically.

5. Data Collection: Create an API or server that can receive and process data from IoT devices. Use a suitable programming language like Node.js, Python, or Java.

6. Database: Store the received data in a database (e.g., MySQL, MongoDB) for historical records and analytics.

7. Authentication and Authorization: Implement user authentication to ensure that only authorized users can access the data.

8. APIs: Design RESTful or GraphQL APIs to handle data retrieval and ensure that your front-end can fetch data from the back-end.

9. Real-time Data Processing: Use a real-time data processing framework or message broker like Apache Kafka or MQTT to handle incoming data streams efficiently.

10. Data Presentation: Convert the received data into appropriate formats (e.g., JSON) and send it to the front-end for real-time display.

Security:

11. Security Measures: Implement security measures such as HTTPS for data transmission, secure authentication, and proper data encryption to protect user and device data.

Scalability:

12. Scalability: Ensure that your platform is scalable to accommodate a growing number of IoT devices and users. Consider containerization and cloud hosting for scalability.

Maintenance and Monitoring:

13. Monitoring: Set up monitoring tools to track the health and performance of your system and receive alerts for issues.

14. Maintenance: Regularly update and maintain your platform to ensure it continues to function smoothly.

Data Processing and Analytics:

15.The collected data is processed and analysed to derive valuable insights. Advanced analytics techniques may be used to detect trends, anomalies, and patterns within the data.

16.Creating a real-time smart water Management platform involves a combination of front end and back-end technologies. Here's a simplified outline using C and C++ and python programming with Wi-Fi connection for the front end and Node.js for the back end.

C++ code:

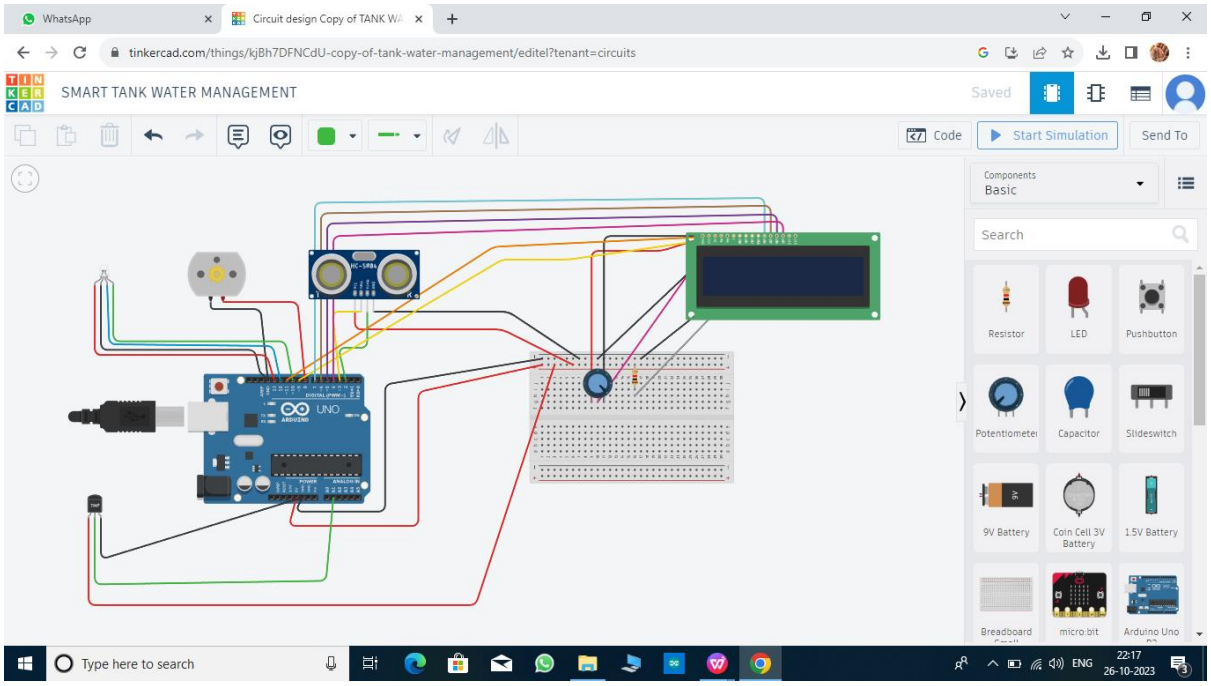
```
#include <LiquidCrystal.h>
```

```
# define echoPin 2 // Echo Pin
# define trigPin 3 // Trigger Pin
int maxRange = 200; // Maximum range
int minRange = 10; // Minimum range
long dur, dist; // Duration used to calculate distance
int val;
int tempPin = 1;
int red_light_pin= 13;
int green_light_pin = 10;
int blue_light_pin = 12;
LiquidCrystal lcd(11, 9, 7, 6, 5, 4);


void setup()
{
 pinMode(LED_BUILTIN, OUTPUT);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(8,OUTPUT);
 pinMode(red_light_pin, OUTPUT);
 pinMode(green_light_pin, OUTPUT);
 pinMode(blue_light_pin, OUTPUT);
 lcd.begin(16, 2);
 // Print a message to the LCD.


}

void loop()
{
```

```
        digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  dur = pulseIn(echoPin, HIGH);
  dist = dur / 58.2;

//temperature
        val = analogRead(tempPin);
        float mv = ( val/1024.0)*5000;
        float cel = mv/10;
        float farh = (cel*9)/5 + 32;
        if(cel>48){
        RGB_color(255, 0, 0);
    lcd.setCursor(0, 0);
    lcd.print("HOT");
    lcd.setCursor(1,1);
    lcd.print(cel);
         }
else if(cel<16){
  RGB_color(0, 0, 255);
  lcd.setCursor(0, 0);
  lcd.print("COLD");
  lcd.setCursor(1,1);
  lcd.print(cel);

}
```

```
  if (dist <= minRange)
  {
    digitalWrite(8, LOW);
  }
        else if(dist >= maxRange){
        digitalWrite(8,HIGH);
    delay(10000);
  }
  else
  {
    digitalWrite(8, LOW);
  }
  delay(50);

}


void RGB_color(int red_light_value, int green_light_value, int blue_light_value)
 {
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
  analogWrite(blue_light_pin, blue_light_value);
}
```

BEFORE SIMULATION:


WhatsApp
Circuit design Copy of TANK WA
tinkercad.com/things/kjBh7DFNCdU-copy-of-tank-water-management/editel?tenant=circuits
SMART TANK WATER MANAGEMENT
Saved
Code
Start Simulation
Send To
Components
Basic
Search
Resistor
LED
Pushbutton
Potentiometer
Capacitor
Slideswitch
9V Battery
Coin Cell 3V Battery
1.5V Battery
Breadboard
micro:bit
Arduino Uno
Type here to search
ENG
22:17
26-10-2023

AFTER SIMULATION: