Smart Water Management

NAAN MUDHALVAN PHASE 5 ASSESSMENT

Course Name : Internet of Things

Project Title : Smart Water Management

Team Name : Tech Enthusiast

GitHub link : https://github.com/soundar98

TeamMembers:

Name	Registernumber	Naan Mudhalvan ID
Senthilnathan JR	721221106093	au721221106093
Vimal N	721221106119	au721221106119
Soundar S	721221106098	au721221106098

1.Objective:

The objective of this project is to create a Smart Water Management system that leverages Internet of Things (IoT) technology to monitor and manage water resources efficiently. The project aims to reduce water wastage, prevent leaks, and ensure the sustainable use of water in urban and rural environments.

IoT Device Setup:

The project involves the deployment of various IoT devices for data collection, control, and communication. Here are the key components of the IoT device setup:

Water Quality Sensors:

These sensors are installed in water sources (e.g., reservoirs, rivers) to monitor water quality parameters such as pH, turbidity, and contamination levels.

Flow Sensors:

Flow sensors are placed in water distribution pipelines to measure water flow rates and detect any irregularities or leaks.

Water Level Sensors:

These sensors are deployed in water storage tanks and reservoirs to monitor water levels and trigger refill requests when necessary.

IoT Gateway:

A central IoT gateway collects data from all the sensors, preprocesses the data, and sends it to a cloud-based platform for analysis and control.

2.Platform Development:

The project's platform development involves creating a cloud-based system for data analysis, visualization, and control. The platform consists of the following components

Cloud Data Storage:

Data collected from the IoT devices is stored in a cloud database for real-time and historical analysis. Services like AWS, Azure, or Google Cloud can be used for this purpose.

Data Analysis and Prediction:

Machine learning models are developed to analyze the data and predict water quality, consumption trends, and leak detection. These models help in making informed decisions.

Control System:

The platform can send commands back to the IoT devices to control water flow, shut off supply in case of emergencies, or trigger maintenance alerts.

Dashboard:

User-friendly dashboards are created for water authorities, environmental agencies, and consumers to access real-time information about water quality, consumption, and alerts.

3.Code Implementation:

The code for this project will be written in various programming languages, depending on the component:

IoT Device Code:

Each IoT device has its own code to read sensor data and transmit it to the IoT gateway. This code may be written in languages like C, Python, or platforms like Arduino.

IoT Gateway Code:

The gateway has code to aggregate data, perform preprocessing, and securely transmit it to the cloud. Communication protocols like MQTT or HTTP may be used.

Cloud Data Storage Code:

Setting up and managing databases can be done using cloud providers' services and APIs.

Data Analysis and Prediction Code:

Machine learning models are implemented using Python, along with libraries like TensorFlow, scikit-learn, or specific water quality analysis tools.

Dashboard Code:

Web-based dashboards can be developed using HTML, CSS, and JavaScript, along with frameworks like React or Angular.

4.Explanation in Detail:

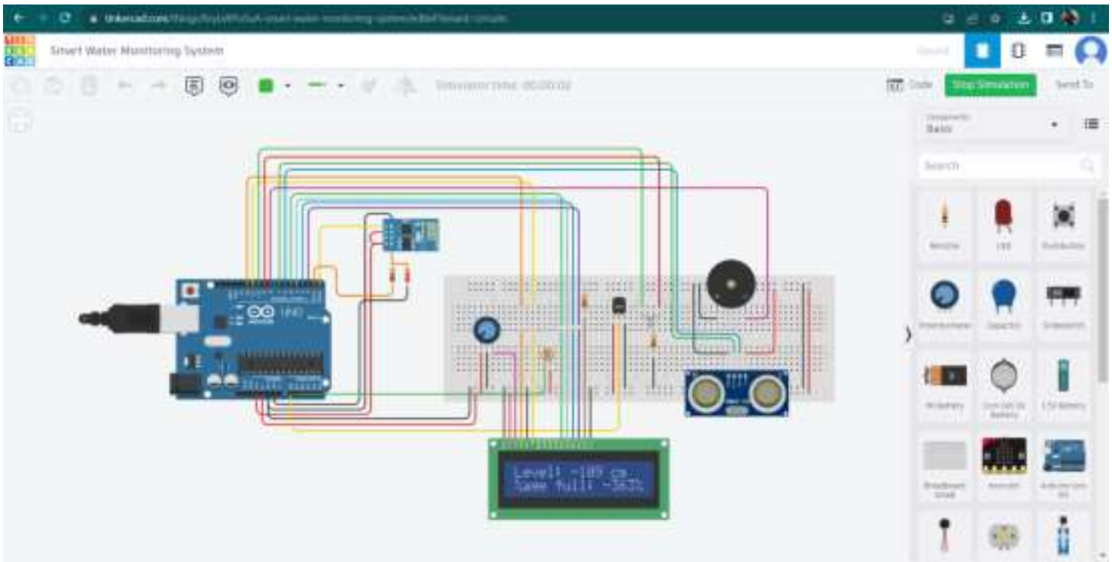
The project monitors water quality, flow rates, and levels to ensure efficient water management. Machine learning models help in early detection of water quality issues and leaks, allowing for preventive actions. Dashboards provide easy access to real-time data and alerts for decision-makers, enabling them to take timely actions to conserve water resources, prevent contamination, and ensure sustainable water usage.

This project's success contributes to better water quality, reduced water wastage, and more efficient water distribution, benefiting both the environment and the community.

URL for Tinkercad Project of Smart water Management:

https://www.tinkercad.com/things/6ryLvRPoSuA-smart-water-monitoring-system/editel?tenant=circuits

Screenshot of Smart water Management Circuit(Tinkercad):



Code for Running above Circuit:

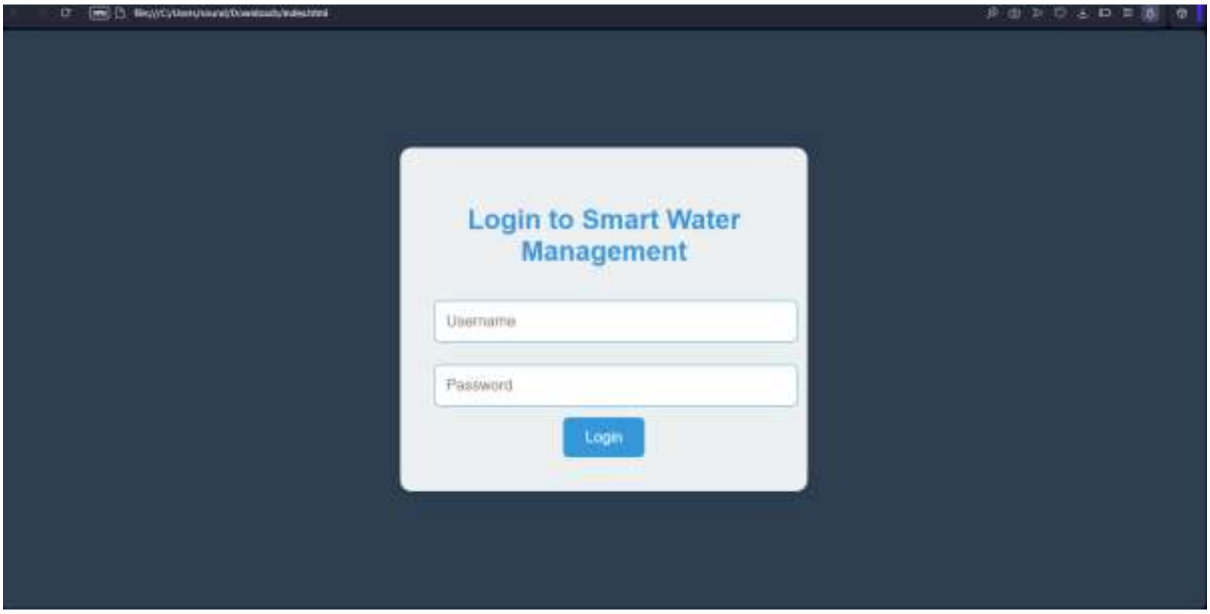
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int redLed = 9;

int greenLed = 10;

int trigPin = 6;

int echoPin = 7;

long duration;

int dist;

int percentfull;

int h = 30;
```

```
const int piezoPin = 8;
void setup() {
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  lcd.begin(16, 2);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  lcd.setCursor(0, 0);
  int value = analogRead(A1);
  dist = findDist();
percentfull = (h - dist) * 100 / h;
  lcd.setCursor(0, 0);
  lcd.print("Level: ");
  lcd.print(h - dist);
  lcd.print(" cm");
  delay(10);
  lcd.setCursor(0, 1);
  lcd.print("%age full: ");
  lcd.print(percentfull);
  lcd.print("%");
  delay(1000);
```

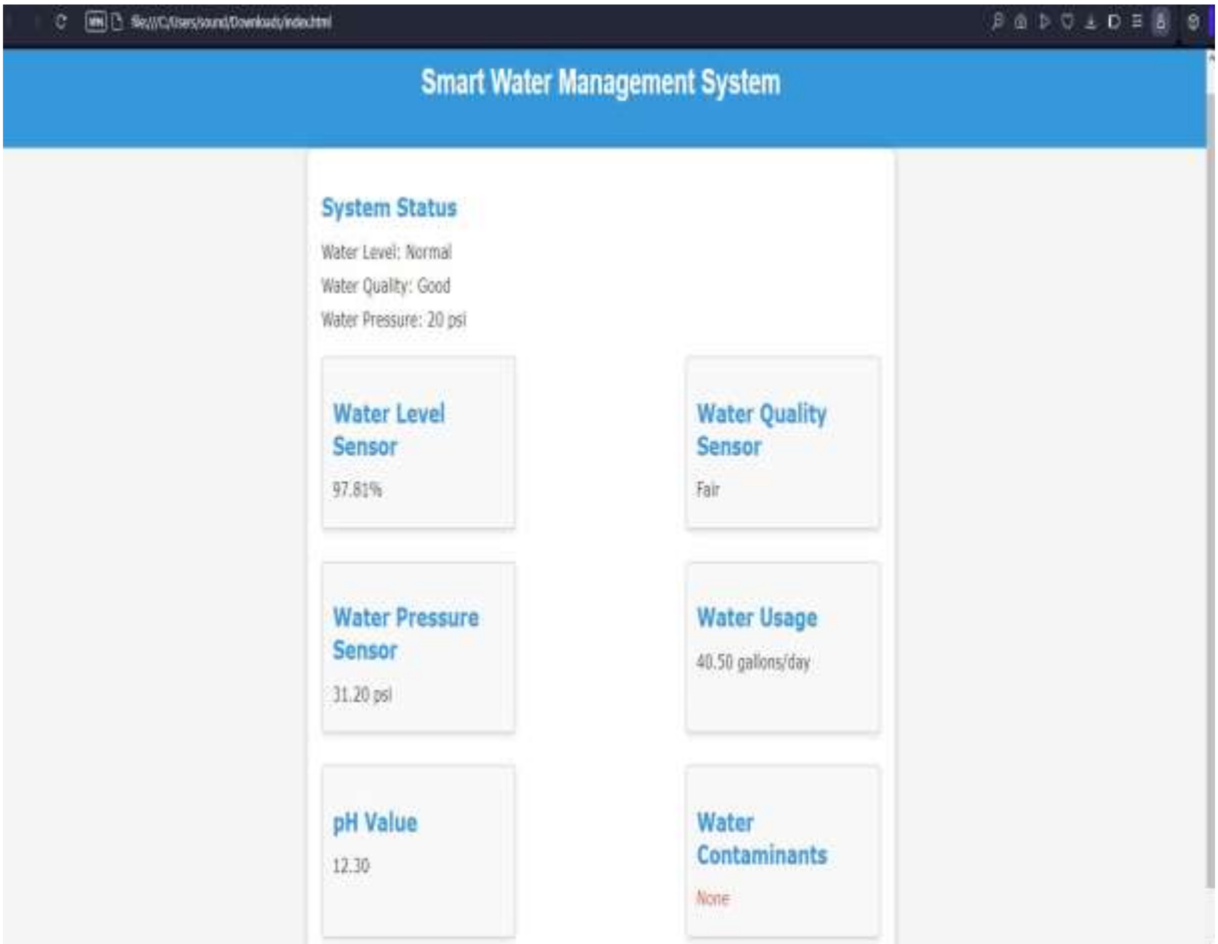
```
if (percentfull >= 90) {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(piezoPin, 500, 500);
    delay(200);
  } else {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    delay(500);
    digitalWrite(greenLed, LOW);
    delay(200); }
  delay(5000);
  lcd.clear();
}
int findDist() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  int distanceCm = duration * 0.034 / 2;
  return distanceCm;
}
```

Webpage of Smart Water Management:

Login page:



Main page:



Analysis page:

Platform UI code for Smart Water Management:

HTML :

We have a simple interface for viewing real-time Smart Water Management.

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Smart Water Management</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>
```

```html
<div id="header">
<h1>Smart Water Management System</h1>
</div>
<div id="container">
<div id="status">
<h2>System Status</h2>
<p>Water Level: <span id="waterLevel">Normal</span></p>
<p>Water Quality: <span id="waterQuality">Good</span></p>
<p>Water Pressure: <span id="waterPressure">20 psi</span></p>
</div>
<div id="sensors">
<div class="sensor-reading">
<h2>Water Level Sensor</h2>
<p><span id="waterLevelSensor">75%</span></p>
</div>
<div class="sensor-reading">
<h2>Water Quality Sensor</h2>
<p><span id="waterQualitySensor">Excellent</span></p>
</div>
<div class="sensor-reading">
<h2>Water Pressure Sensor</h2>
<p><span id="waterPressureSensor">20 psi</span></p>
</div>
<div class="sensor-reading">
<h2>Water Usage</h2>
```

```html
<p><span id="waterUsage">500 gallons/day</span></p>
</div>
<div class="sensor-reading">
<h2>pH Value</h2>
<p><span id="pHValue">7.0</span></p>
</div>
<div class="sensor-reading">
<h2>Water Contaminants</h2>
<p><span id="waterContaminants">None</span></p>
</div>
</div>
</div>
<div id="graphic-picture">

</div>
<script src="script.js"></script>
</body>
</html>
```

CSS:

Create a stylesheet (styles.css) to define the layout and styling of your webpage.

```css
body {
  font-family: 'Verdana', sans-serif;
```

```
    background-color: #ffffff;
    color: #333;
    margin: 0;
    padding: 0;
}
#header {
    background-color: #050146;
    color: #fffefe;
    text-align: center;
    padding: 20px;
}
#container {
    max-width: 800px;
    margin: 0 auto;
    background-color: #a7acec;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
    padding: 20px;
    border-radius: 10px;
}
#status, #sensors {
    font-size: 18px;
}
#sensors {
    display: flex;
```

```
    flex-wrap: wrap;
    justify-content: space-between;
}
.sensor-reading {
    flex: 0 0 calc(33.333% - 20px);
    padding: 15px;
    margin: 15px 0;
    background-color: #a7acec;
    border: 1px solid #09021d;
    border-radius: 5px;
    box-shadow: 0 3px 6px rgba(0, 0, 0, 0.1);
}
h2 {
    font-size: 24px;
    color: #170a61;
    margin-bottom: 15px;
}
p {
    font-size: 18px;
    margin: 10px 0;
    color: #333;
}
#waterContaminants {
    color: #c72512;
```

```
}
#graphic-picture {
   text-align: center;
}
img {
   max-width: 100%;
   height: auto;
}
```

JAVASCRIPT:

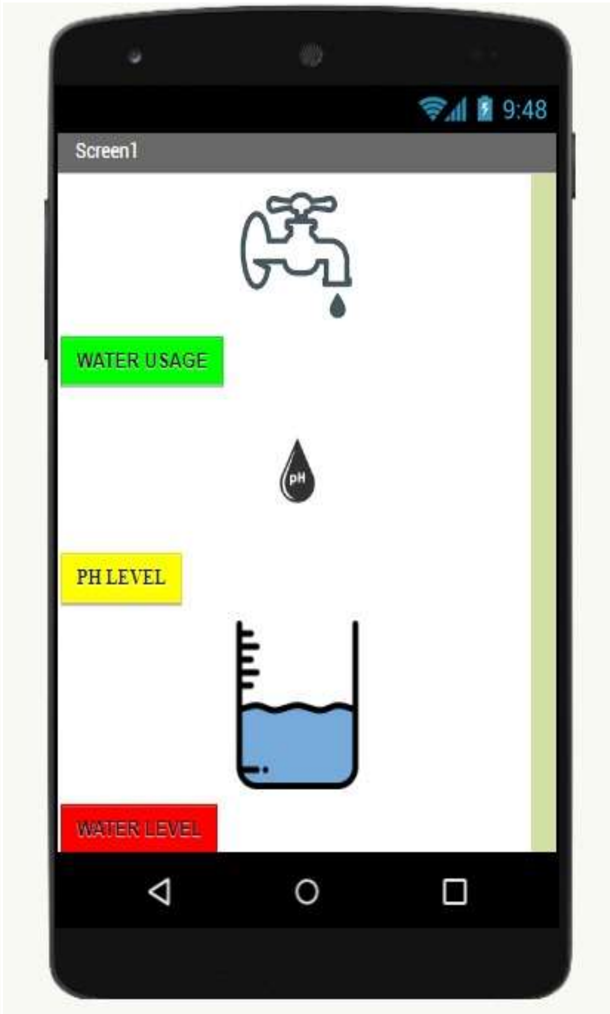
Use JavaScript (app.js) to simulate real-time Smart Water Management updates.

```
const simulatedData = {
   waterLevel: "Normal",
   waterQuality: "Good",
   waterPressure: "20 psi",
   waterLevelSensor: "75%",
   waterQualitySensor: "Excellent",
   waterPressureSensor: "20 psi",
   waterUsage: "500 gallons/day",
   pHValue: "7.0",
   waterContaminants: "None",
};
function updateSensorData() {
```

```
    document.getElementById("waterLevel").textContent =
simulatedData.waterLevel;
    document.getElementById("waterQuality").textContent =
simulatedData.waterQuality;
    document.getElementById("waterPressure").textContent =
simulatedData.waterPressure;
    document.getElementById("waterLevelSensor").textContent =
simulatedData.waterLevelSensor;
    document.getElementById("waterQualitySensor").textContent =
simulatedData.waterQualitySensor;
    document.getElementById("waterPressureSensor").textContent =
simulatedData.waterPressureSensor;
    document.getElementById("waterUsage").textContent =
simulatedData.waterUsage;
    document.getElementById("pHValue").textContent =
simulatedData.pHValue;
    document.getElementById("waterContaminants").textContent =
simulatedData.waterContaminants;
}
updateSensorData();
setInterval(() => {simulatedData.waterLevelSensor = (Math.random()
* 100).toFixed(2) + "%";
    simulatedData.waterQualitySensor = ["Excellent", "Good", "Fair",
"Poor"][Math.floor(Math.random() * 4)];
    simulatedData.waterPressureSensor = (Math.random() * 50 +
10).toFixed(2) + " psi";
```

```
    simulatedData.waterUsage = (Math.random() * 1000).toFixed(2) +
" gallons/day";
    simulatedData.pHValue = (Math.random() * 14).toFixed(2);
    simulatedData.waterContaminants = ["None", "Chlorine", "Lead",
"Bacteria"][Math.floor(Math.random() * 4)];
    updateSensorData();
}, 5000);
```

Mobile app development:



Conclusion:

The Smart Water Management project employs technology and data analyticsto enhance water resource sustainability. Real-time data collection, user-friendly web interfaces, and mobile applications provide valuable insights intowater quality, usage, and conservation. Security measures and complianceensure data protection, while ongoing improvements reflect the project water management.