JP COLLEGE OF ENGINEERING

COLLEGE NAME :JP COLLEGE OF ENGINEERING COLLEGECODE:9512
PROJECT NAME:SMART WATER MANAGEMENT
PROJECT ID:Proj_211932_Team_1

TEAMS NUMBERS:
•M.SAMYDURAI,951221106040
Email:samydurai2004@gmail.com
• S.SUNDARAN, 951221106304
Email:sundaranvkp25@gamil.com
• M.KALAISEVAN,951221106016
Email:junga4736@gmail.com

SMART WATER MANAGEMENT SYSTEMS

OBJECTIVE:

Designing a smart water management project involves integrating technology to monitor, control, and optimize water usage. Consider implementing sensor networks for real-time data collection on water quality and consumption. Use analytics to identify patterns and anomalies, enabling efficient resource allocation. Implement automated control systems for valves and pumps based on data insights. Ensure cybersecurity measures to protect the infrastructure. Additionally, incorporate user-friendly interfaces for stakeholders to access and understand the data, promoting awareness and responsible water usage.

Main benefits

- Better transparency in water management
- Fewer incidents
- Enhanced control over the water supply
- Saved city budget
- Improved city sustainability.

Smart water management system IoT design :

PROJECT DEFINITION AND DESIGN:

1.Project Definition:

- **Objectives:** Clearly state the goals of the project, such as improving water efficiency, reducing waste, or ensuring water quality.

- **Scope:** Define the geographical area or specific water systems the project will cover.

- **Stakeholders:** Identify key stakeholders, including government bodies, communities, and industries.

2.Research and Analysis:

- **Water Usage Assessment:** Analyze current water usage patterns and identify areas for improvement.

- **Technology Review:** Explore available technologies for monitoring, automation, and data analytics in water management.

3.System Components:

- .**Sensor Networks:** Specify types of sensors (e.g., flow meters, quality sensors) and their strategic placement.

- **Automation Systems:** Design automated processes for tasks like irrigation, leak detection, and water distribution.

- **Data Analytics Platform:** Choose or design a platform for processing and analyzing collected data.

4.Integration:

- **Interconnected Systems:** Ensure seamless communication between sensors, automation systems, and the analytics platform.

- **Compatibility:** Confirm compatibility with existing infrastructure and technologies.

5.Risk Assessment:

- **Identify Risks:** Anticipate potential challenges, such as technical issues, data security concerns, or resistance from stakeholders.

- **Risk Mitigation:** Develop strategies to mitigate or manage identified risks.

6.Regulatory Compliance:

- Legal Requirements: Understand and comply with local and national regulations related to water management and technology use.

7.Budget and Resources:

- **Cost Estimation:** Provide a detailed breakdown of costs, including equipment, technology implementation, and ongoing maintenance.

- **Resource Allocation:** Define the human resources and expertise required for the project.

8.Timeline:

- **Project Phases:** Divide the project into manageable phases with specific milestones.

- **Implementation Schedule:** Develop a timeline for each phase, considering dependencies and potential delays.

9.Monitoring and Evaluation:

- **Performance Metrics:** Establish measurable indicators to assess the success of the project.

- **Monitoring Plan:** Develop a plan for continuous monitoring and evaluation of the smart water management system.

10.Documentation and Communication:

- **Document Design Choices:** Maintain clear documentation of the chosen technologies, processes, and design decisions.

- **Communication Plan:** Establish a plan for keeping stakeholders informed about project progress.

Iot design system:

1. Water Sensors:

- **These sensors are deployed in key locations, such as reservoirs, pipelines, and distribution points.**

- **They monitor water quality, quantity, and potential issues like leaks.**

2. Microcontrollers:

- **Connected to the water sensors, microcontrollers process and transmit data.**

- **They might use platforms like Arduino or Raspberry Pi.**

3. Communication Module:

- **Facilitates the transfer of sensor data to the central control system.**

- **Common communication protocols include Wi-Fi, LoRa, or cellular networks.**

4. Gateway:

- Aggregates data from multiple microcontrollers and manages communication.

- Acts as a bridge between the local network and the wider internet.

5. Cloud Platform:

- Receives, stores, and processes data from the gateways.

- Allows for remote access, analysis, and control of the water management system.

6. Data Storage:

- Long-term storage of historical data for analysis and trend identification.

7. Data Analytics:

- Utilizes machine learning algorithms or statistical methods to derive insights from the collected data.

- Identifies patterns, anomalies, and potential issues.

8. User Interface:

- A dashboard or user interface accessible through web or mobile applications.

- Provides real-time information, historical trends, and control options for users.

9. Control System:

- Enables automated control actions based on analyzed data.

- For example, adjusting water flow, detecting and isolating leaks, or activating alarms.

10. Actuators:

- Devices that execute control commands, such as valves, pumps, or alarms.

- Connected to the control system to implement necessary actions.

11. Security Measures:

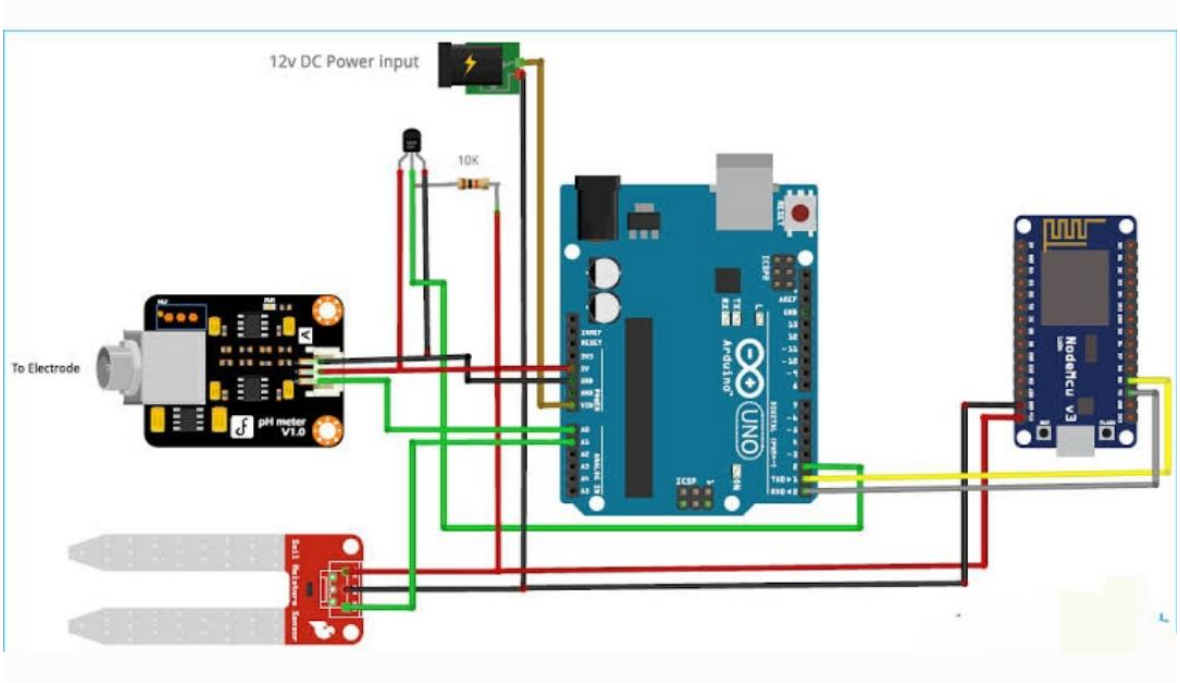
- **Encryption and authentication protocols to secure data transmission.**

- **Access controls to prevent unauthorized**

Project definition:

A smart water management project aims to implement advanced technologies and data-driven solutions to optimize the use of water resources. This could involve deploying sensors to monitor water quality, consumption, and detect leaks. Automation systems can be integrated for efficient irrigation in agriculture or smart metering in urban areas. The project would likely include data analytics to provide insights, improve decision-making, and contribute to sustainable .

Circuit Design:



Program

```
#include <ThingerESP8266.h>
#include <ESP8266WiFi.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET LED_BUILTIN
```

```
#define USERNAME "test123"
#define DEVICE_ID "SWM"
#define DEVICE_CREDENTIAL "ABCDEFGHIJ"


#define SSID "test123"
#define SSID_PASSWORD "test123"
Adafruit_SSD1306 display(OLED_RESET);


byte indikator = 13;


byte sensorInt = 0;
byte flowsensor = D3;


float konstanta = 4.5; //konstanta flow meter


volatile byte pulseCount;


float debit;
float harga;
unsigned int flowmlt;
unsigned long totalmlt;


unsigned long oldTime;


ThingerESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);


void setup()
```

```
        {
        display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
        display.clearDisplay();
        display.display();
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.setCursor(0, 0);


        // Inisialisasi port serial
        Serial.begin(9600);


        pinMode(flowsensor, INPUT);


        pulseCount = 0;
        debit = 0.0;
        flowmlt = 0;
        totalmlt = 0;
        oldTime = 0;
        harga = 0.0;


        // digital pin control example (i.e. turning on/off a light, a relay,
configuring aparameter, etc)
        thing["sensor"] >> [](pson& out){


        digitalWrite(flowsensor, HIGH);


        attachInterrupt(digitalPinToInterrupt(D3), pulseCounter, FALLING);


        out["debit"] = debit;
        out["volume"] = totalmlt;
        out["harga"] = harga;
```

```
};
}


void loop()
{
thing.handle();
display.clearDisplay();
if((millis() - oldTime) > 1000)
{
detachInterrupt(sensorInt);
debit = ((1000.0 / (millis() - oldTime)) * pulseCount) / konstanta;
oldTime = millis();
flowmlt = (debit / 60) * 1000;
totalmlt += flowmlt;
harga = totalmlt*0.002;


unsigned int frac;


Serial.print("Debit air: ");
Serial.print(int(debit));
Serial.print("L/min");
Serial.print("\t");
display.setCursor(0, 0);
display.print("Debit air: ");
display.setCursor(60, 0);
display.print(int(debit));
display.setCursor(85, 0);
display.print("L/min");




Serial.print("Volume: ");
```

```
Serial.print(totalmlt);
Serial.print("mL");
Serial.print("\t");
display.setCursor(0, 12);
display.print("Volume: ");
display.setCursor(50, 12);
display.print(totalmlt);
display.setCursor(100, 12);
display.print("mL");
display.print("\t");


Serial.print("Harga: ");
Serial.print("Rp ");
Serial.println(harga);
display.setCursor(0, 24);
display.print("Harga: ");
display.setCursor(45, 24);
display.print("Rp ");
display.setCursor(70, 24);
display.println(harga);
display.display();




pulseCount = 0;


attachInterrupt(digitalPinToInterrupt(D3), pulseCounter, FALLING);
}
}




void pulseCounter()
{
```

```
        // Increment the pulse counter
        pulseCount++;
        }
```

Reference github link:
https://github.com/thefloatman/smart-water-meter-project/commit/c5824c51870d1ae0b8efb3e400e1a27814352b00#diff-e2eb93a61ffd7877ea5c751abcb3a618e8e2e9a2073a27f66d4114fe10819f86