DOMAIN NAME: INTERNET OF THINGS
PROJECT NAME: SMART WATER SYSTEM

Contents:

1. ABSTRACT:

Introduction:

Water scarcity poses a pressing global challenge. The "Smart Water System" project represents an innovative approach to tackle this issue by harnessing the power of IoT technology and design thinking. Our mission is to monitor and manage water consumption in public spaces, promoting conservation, raising awareness, and ensuring sustainable resource management.

Project Definition and Design Thinking:

Project Definition:

This initiative centers on the deployment of IoT sensors in public areas, such as parks and gardens, to enable real-time water consumption monitoring. Our core objectives encompass enhancing public awareness, reducing water waste, and facilitating sustainable water resource management. The project's structure encompasses defining objectives, designing the IoT sensor network, developing the data-sharing platform, and executing seamless integration through IoT technology and Python.

Design Thinking:

To achieve our goals, we adopt a design thinking approach. We begin by empathizing with water users and stakeholders, comprehending their needs, and identifying pain points. We proceed to define the core problem areas related to water waste and resource inefficiency. Through ideation, we brainstorm creative solutions. Prototyping and testing refine these ideas. Finally, we implement the most effective solutions to address our defined objectives.

IoT Devices Utilized:

1. **Flow Sensors**: These devices measure water flow rates, offering real-time consumption data.
2. **Water Level Sensors**: Utilized for monitoring water levels in reservoirs and containers.
3. **IoT Microcontrollers**: Raspberry Pi.
4. **Wireless Communication Modules**: Ensuring seamless data transmission.
5. **Power Supply**: For reliable and continuous operation.
6. **Data Storage and Analysis**: Utilizing cloud platforms for efficient data management.

In a smart water system project focused on water conservation, some specific targets and objectives could include:

Leak Detection and Reduction: Implement technology to detect and minimize water leaks in the distribution system to reduce non-revenue water loss.

Real-time Monitoring: Develop a system for real-time monitoring of water usage at various points in the distribution network to identify areas of high consumption and potential waste.

Consumption Education: Creating an application to educate consumers about water conservation and provide them with tools to monitor and reduce their water usage.

Pressure Management: Optimize water pressure to reduce unnecessary water flow and energy consumption in the distribution network.

Rainwater Harvesting: Promote and facilitate rainwater harvesting systems for residential and commercial properties to reduce reliance on municipal water sources.

Irrigation Efficiency: Implement smart irrigation systems that adjust watering schedules based on weather conditions and soil moisture levels.

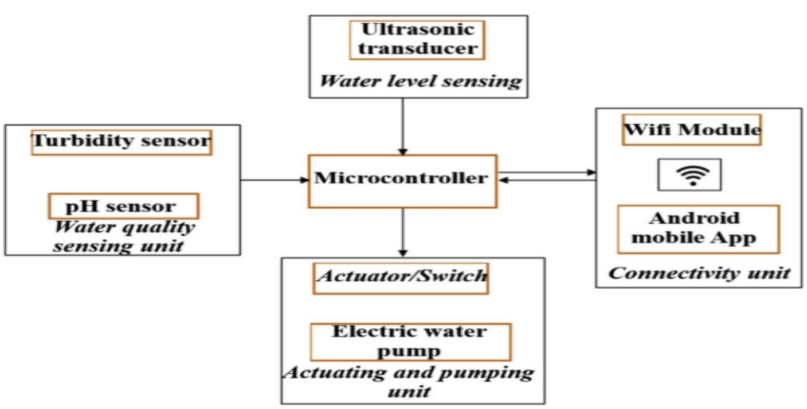
Water Quality Management: Ensure the quality of water in the system, preventing contamination that might lead to unnecessary flushing or waste.

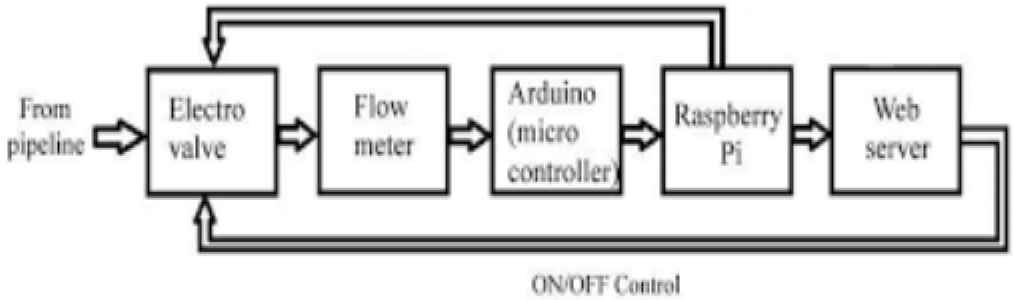
Demand Forecasting: Use data analytics to predict water demand patterns, helping utilities allocate resources more efficiently.

Incentive Programs: Establish incentive programs for water-efficient appliances, fixtures, and landscaping to encourage consumers to adopt water-saving technologies.

Water Recycling: Explore opportunities for wastewater treatment and recycling to reduce the demand on fresh water sources.

2. ARCHITECTURE:





EXPLANATION:

A smart water system, often referred to as a "smart water infrastructure" or "smart water management system," is an advanced and technology-driven approach to managing and optimizing water resources. It integrates various components such as sensors, data analytics, and communication technology to monitor, control, and improve the efficiency of water supply and distribution.

1. Sensors and Data Collection:

- **Flow Sensors**: They monitor the flow of water in pipes to detect leaks or anomalies.
- **Water Level Sensors**: These sensors track water levels in reservoirs, tanks, and rivers.

2. Data Transmission:

- Sensor data is collected and transmitted to a central server using various communication technologies such as cellular networks, Wi-Fi, LoRaWAN, or satellite communication.

3. Data Storage:

- Data from sensors is stored in databases and cloud platforms, making it accessible for analysis and decision-making.

4. User Interface and Visualization:

- Dashboards and user interfaces are designed for water utility managers, operators, and even consumers to visualize the data. This can include web applications and mobile apps.
- Real-time information on water quality, water levels, consumption, and more is made available to users.

5. Control and Actuation:

- Based on the data and analysis, the system can automatically control certain aspects of the water infrastructure. For example, it can adjust the flow of water in pipes to balance supply and demand.
- Automated valves and pumps can be controlled to optimize water distribution.

6. Demand Management:

- By analyzing consumption patterns, the system can encourage water conservation and provide incentives to reduce water usage during peak demand times.

7.Regulatory Compliance:

Smart water systems often need to comply with local and national regulations regarding water quality and distribution. Compliance monitoring and reporting are often integrated into the system.

8.Communication with Consumers:

Consumers can receive real-time information about their water usage, quality, and conservation tips through apps or online portals.

In summary, a smart water system leverages sensors, data analytics, and automation to efficiently manage water resources, reduce water loss, and ensure sustainable water distribution while enhancing the overall operation and maintenance of water infrastructure.

3. IoT SENSOR SETUP:

Flow sensor:

Flow sensors in a Smart Water System project play a critical role in monitoring and managing the flow of water within the distribution network. Here are the main points regarding flow sensors:

1.Flow Measurement: Flow sensors are used to measure the rate of water flow through pipelines or channels in the distribution network. They provide real-time data on the volume of water passing through a specific point, allowing for accurate tracking of water consumption and distribution.

2.Leak Detection: Flow sensors are essential for leak detection in the water distribution system. Sudden changes in flow rates that deviate from expected values can indicate the presence of a leak. Early detection of leaks is crucial for minimizing water loss and preventing damage to infrastructure.

3.Pressure Monitoring: Flow sensors often work in conjunction with pressure sensors to monitor the pressure of water within the distribution network. Pressure changes can affect flow rates, and monitoring both parameters helps in maintaining the system's operational efficiency and identifying anomalies.

4.Data for Resource Allocation: Flow sensor data is valuable for optimizing the allocation of water resources. It enables utilities to allocate water based on demand patterns and adjust the flow rates to ensure a reliable and consistent water supply to consumers.

5.System Performance Optimization: Flow sensors provide data that can be used to assess and optimize the overall performance of the water distribution system. By analyzing flow data, utilities can identify areas of inefficiency, plan for infrastructure improvements, and enhance the management of water resources to meet the project's objectives.

LEVEL SENSORS:

Level sensors in a Smart Water System project are vital for monitoring and managing the water levels in reservoirs, tanks, and other water storage facilities. Here are the main points regarding level sensors:

1. **Water Level Measurement:** Level sensors are designed to measure and report the water levels in storage tanks, reservoirs, and other water holding structures. They provide real-time data on the current water levels, allowing system operators to track the availability of water resources.

2. **Resource Management:** The data collected by level sensors is critical for managing water resources effectively. By continuously monitoring water levels, utilities and operators can make informed decisions about water storage, distribution, and usage. This ensures a reliable water supply to consumers and helps prevent water shortages.

3. **Alerts and Preventive Actions:** Level sensors are often integrated with alerting systems. When water levels fall below or exceed certain predefined thresholds, the system can trigger alerts, enabling rapid response and preventive actions. For example, when water levels in a reservoir are too low, the system can initiate actions to refill it or reduce water supply to avoid running dry.

4. APP DEVELOPMENT:

FRONT END:

1. USER INTERFACE:

- **HTML:** HTML, which stands for Hyper Text Markup Language, is the backbone of the World Wide Web. It serves as the fundamental building block for creating and structuring web content. HTML is a markup language that uses tags to define elements within a web page, such as headings, paragraphs, images, links, and more.

- **CSS:** Cascading Style Sheets, commonly known as CSS, play a pivotal role in the visual aesthetics and design of websites and web applications. CSS is a powerful styling language that works in conjunction with HTML to control the layout, formatting, and overall presentation of web content. It enables web developers and designers to define colors, fonts, spacing, and positioning, ensuring a consistent and visually appealing user experience across various devices and screen sizes.

- **JAVASCRIPT:** JavaScript is a versatile and widely-used programming language primarily known for its role in web development. It is a client-side scripting language, meaning it runs in a user's web browser, enabling dynamic and interactive functionality on websites.

2. DATA VISUALIZATION:

CHART.JS

- Use libraries like Chart.js to create interactive charts and graphs that visually represent water consumption data.
- Display trends and patterns that help users understand their usage.

3. USER ENGAGEMENT:

- Add features like notifications and alerts to inform users of high consumption or water-saving tips.
- Provide a feedback mechanism for users to report leaks or water-saving initiatives.
- This can be done using HTML, CSS and JavaScript.

4. MIT APP INVENTOR:

MIT App Inventor primarily focuses on the development of mobile apps' front-end.

It is designed for creating the user interface, logic, and functionality of mobile applications without requiring extensive knowledge of traditional programming languages.
However, to create mobile apps with server-side or backend functionalities, you typically need to complement MIT App Inventor with external server technologies, such as databases, web services, or cloud-based platforms.

BACK END:

1) **Cloud-Based Database and Backend-as-a-Service (BaaS):**

Firebase: Google's Firebase provides a comprehensive backend solution, offering real-time database, authentication, and cloud functions. It's suitable for managing data, user accounts, and real-time updates for your Smart Water Management System app.

2) **Custom Backend Development:**

Node.js with Express: Node.js, with the Express.js framework, is a popular choice for building custom backend servers. It's flexible and can handle data processing, RESTful APIs, and database integration.

Python with Django or Flask: Python is a versatile language for building backend. Django and Flask are web frameworks suitable for building custom backend for your Smart Water Management System.

3) **Databases:**

MongoDB: database MongoDB is useful for handling unstructured or semi-structured data that may be generated by IoT sensors.

4) **Cloud platforms**:

Amazon Web Services (AWS): offer a wide range of services for building and scaling backend, including server less computing and data storage.

5. RASPBERRY PI INTEGRATION:

Integrating Raspberry Pi with smart water system sensors can be a great project for monitoring and managing water-related data. Here are the basic steps to get started:

1. Select the Sensors: Choose the appropriate sensors for your smart water system. We have selected water level sensors and flow sensors. Then ensure whether they are compatible with Raspberry Pi.

2. Raspberry Pi Setup: Set up your Raspberry Pi with the required operating system (like Raspbian) and connect it to your local network. Make sure it has internet connectivity.

3. GPIO Interface: Connect the sensors to the Raspberry Pi's GPIO pins or via compatible interfaces (e.g., I2C or SPI). Ensure you have the necessary libraries and drivers installed to interact with these sensors.

4. Data Acquisition: Write code on the Raspberry Pi to read data from the sensors. You can use Python, for example, to interface with the sensors and collect data.

5. Data Processing and Analysis: Process the data as needed. You can perform data analysis or apply algorithms for specific purposes, such as leak detection or water quality analysis.

6. Real-time Monitoring: Set up a system to provide real-time monitoring and alerts if sensor readings go beyond predefined thresholds.

7. User Interface: Create a user interface to visualize and control the system. This can be a web-based dashboard accessible from any device on your network.

8. Testing and Calibration: Thoroughly test the system and calibrate the sensors to ensure accurate measurements.

9. Maintenance: Regularly maintain and update your system to ensure its reliability and accuracy.

6. CODE IMPLEMENTATION:

FRONT END:

LEVEL SENSOR:

```
C: > Users > Raj > <> water.html > ...
<!DOCTYPE html>
<html>
<head>
    <title>Smart Water Management System</title>
    <style>
        /* CSS styles for your page */
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            margin: 0;
            padding: 0;
        }

        header {
            background-color: #3498db;
            color: #fff;
            text-align: center;
            padding: 10px;
        }

        .container {
            max-width: 800px;
            margin: 0 auto;
            padding: 20px;
            background-color: #fff;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
        }
```

```
        }

        #waterConsumption {
            font-size: 24px;
            margin-bottom: 20px;
        }

        button {
            background-color: #3498db;
            color: #fff;
            border: none;
            padding: 10px 20px;
            cursor: pointer;
        }

        button:hover {
            background-color: #2186c8;
        }
    </style>
</head>
<body>
    <header>
        <h1>Smart Water Management System</h1>
    </header>
    <div class="container">
        <div id="waterConsumption">
            Current Water Consumption: 0 gallons
        </div>
        <button id="updateButton">Update Consumption</button>
    </div>

    <script>
```

```html
        <button id="updateButton">Update Consumption</button>
    </div>

    <script>
        // JavaScript code for fetching and updating water consumption
        let currentConsumption = 0;

        function updateWaterConsumption() {
            // Simulate data update (replace with real data retrieval logic)
            currentConsumption += Math.floor(Math.random() * 10);
            document.getElementById('waterConsumption').textContent = `Current Water Consumption: ${currentConsu
        }

        document.getElementById('updateButton').addEventListener('click', updateWaterConsumption);

        // Initial data fetch
        updateWaterConsumption();
    </script>
</body>
</html>
```

FLOWSENSOR:

```
<!DOCTYPE html>
<html>
<head>
    <title>Flow Sensor</title>
    <style>
        /* Add your CSS styles here */
        body {
            font-family: Arial, sans-serif;
        }
        #sensorData {
            font-size: 24px;
        }
        #startButton, #stopButton {
            padding: 10px 20px;
            font-size: 16px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <h1>Flow Sensor Dashboard</h1>
    <p id="sensorData">Flow Rate: <span id="flowRate">0</span> L/min</p>
    <button id="startButton">Start</button>
    <button id="stopButton">Stop</button>

    <script>
        // JavaScript code to interact with the flow sensor
        let isRunning = false;
        let flowRate = 0;
        let intervalId;

        function startFlowSensor() {
```

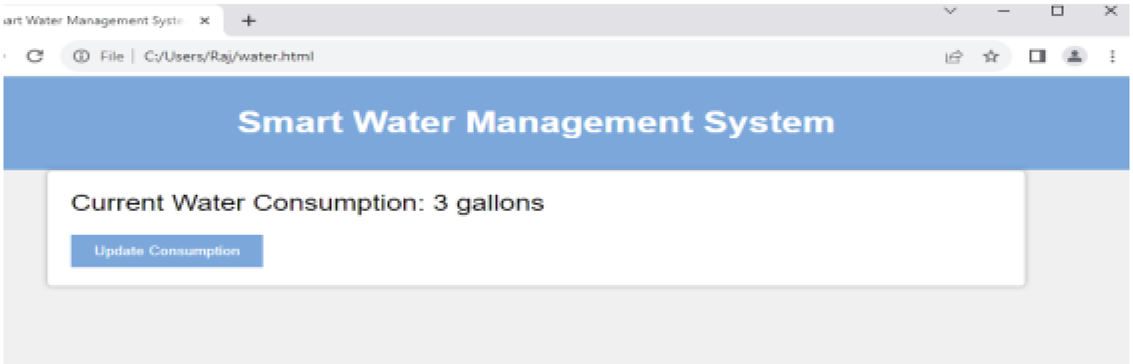
```
        // JavaScript code to interact with the flow sensor
        let isRunning = false;
        let flowRate = 0;
        let intervalId;

        function startFlowSensor() {
            if (!isRunning) {
                intervalId = setInterval(updateFlowRate, 1000);
                isRunning = true;
            }
        }

        function stopFlowSensor() {
            clearInterval(intervalId);
            isRunning = false;
        }

        function updateFlowRate() {
            // Simulate flow rate data (replace with actual sensor data)
            flowRate = Math.random() * 10;
            document.getElementById("flowRate").innerText = flowRate.toFixed(2);
        }

        document.getElementById("startButton").addEventListener("click", startFlowSensor);
        document.getElementById("stopButton").addEventListener("click", stopFlowSensor);
    </script>
</body>
</html>
```

sensor water.py - C:\Users\Raj\AppData\Local\Programs\Python\Python310\sensor water.py (3.10.0)

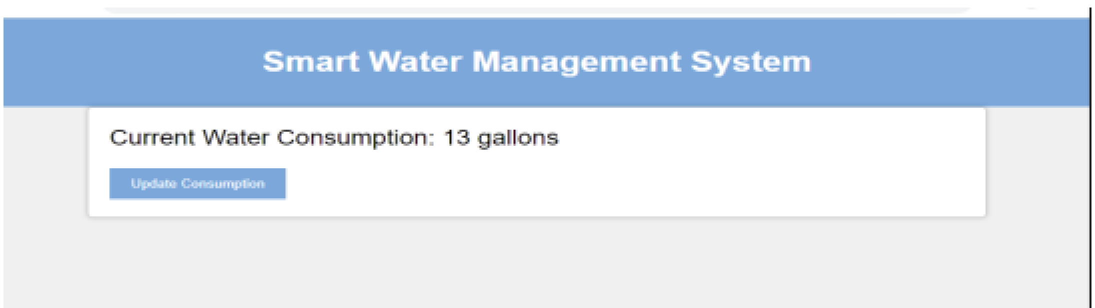
File Edit Format Run Options Window Help

```python
import paho.mqtt.client as mqtt
import random
import time

# Define MQTT broker and topic information
mqtt_broker = "your_mqtt_broker_url"
mqtt_port = 1883
mqtt_topic = "water_consumption"

# Function to simulate water consumption data (replace with actual sensor data)
def simulate_water_consumption():
    return random.uniform(0.1, 2.0)  # Simulated water consumption data in liters

# Callback when the client connects to the MQTT broker
def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe(mqtt_topic)

# Create an MQTT client instance
client = mqtt.Client()
client.on_connect = on_connect

# Connect to the MQTT broker
client.connect(mqtt_broker, mqtt_port, 60)

# Publish water consumption data at regular intervals
while True:
    water_consumption = simulate_water_consumption()
    client.publish(mqtt_topic, payload=f"Water consumption: {water_consumption} liters")
    print(f"Published: Water consumption - {water_consumption} liters")
    time.sleep(5)  # Adjust the interval as needed

# Keep the script running
client.loop_forever()
```

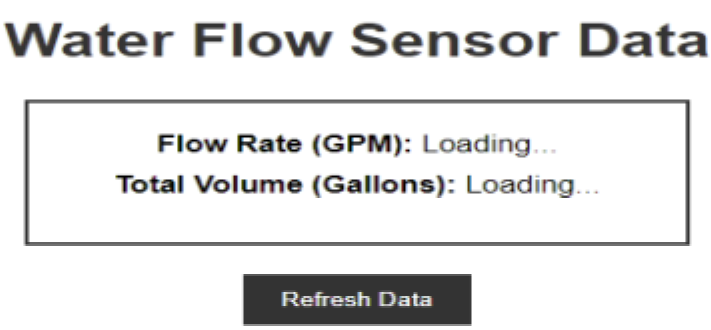
OUTPUT:



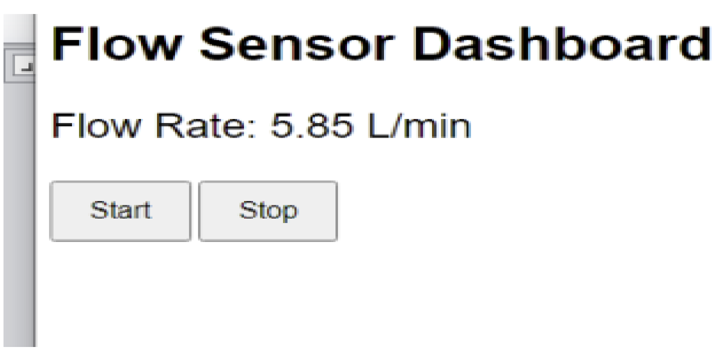
C:/Users/Raj/water.html
Smart Water Management System
Current Water Consumption: 3 gallons
Update Consumption



Smart Water Management System
Current Water Consumption: 13 gallons
Update Consumption



Water Flow Sensor Data
Flow Rate (GPM): Loading...
Total Volume (Gallons): Loading...
Refresh Data



Flow Sensor Dashboard
Flow Rate: 5.85 L/min
Start
Stop

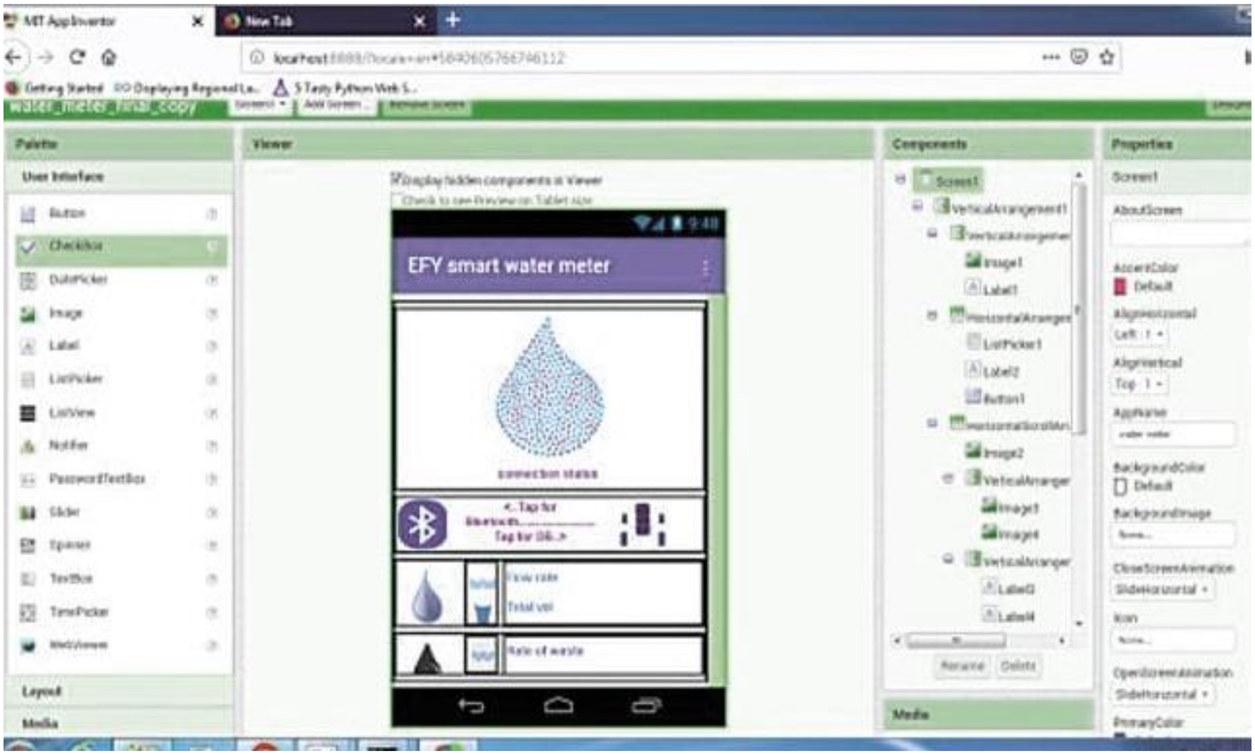
BLUETOOTH CONNECTION (for data transmission):

```
    Meter2.tick(3000);
Serial.print(String(Meter1.getCurrentFlowrate()));
Serial.print("l/m");
Serial.print(":");
Serial.print( String(Meter1.getTotalVolume()));
Serial.print("L total");
Serial.print(":");
Serial.print(String(Meter2.getCurrentFlowrate()));
Serial.print("l/m");
Serial.print(":");
Serial.print( String(Meter2.getTotalVolume()));
Serial.print("L total:");
Serial.println(" ");
```

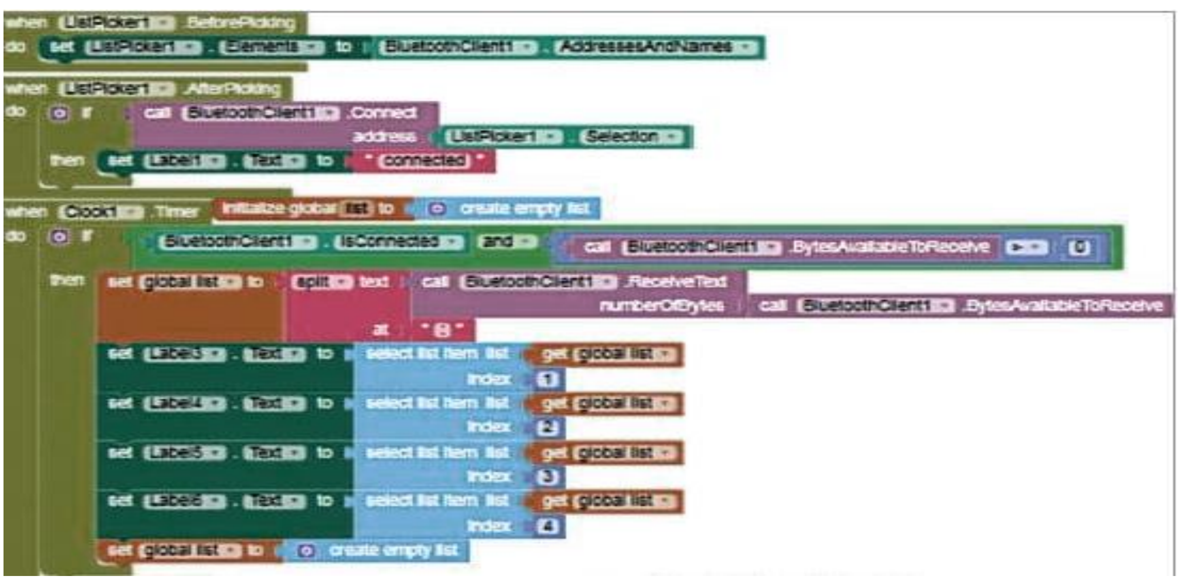
```
oled.clearDisplay();
oled.setTextSize(1);
oled.setTextColor(1);
oled.setCursor(0, 0);
oled.println("Flow rate");
oled.setTextSize(2);
oled.setCursor(10, 12);
oled.println(Meter2.getCurrentFlowrate());
oled.setTextSize(2);
oled.setCursor(80, 12);
oled.println("l/m");
```

OUTPUT:

FLOW METER:



CODE BLOCK:



7. SCHEMATICS OF SENSORS AND APP:

Schematic Flow:

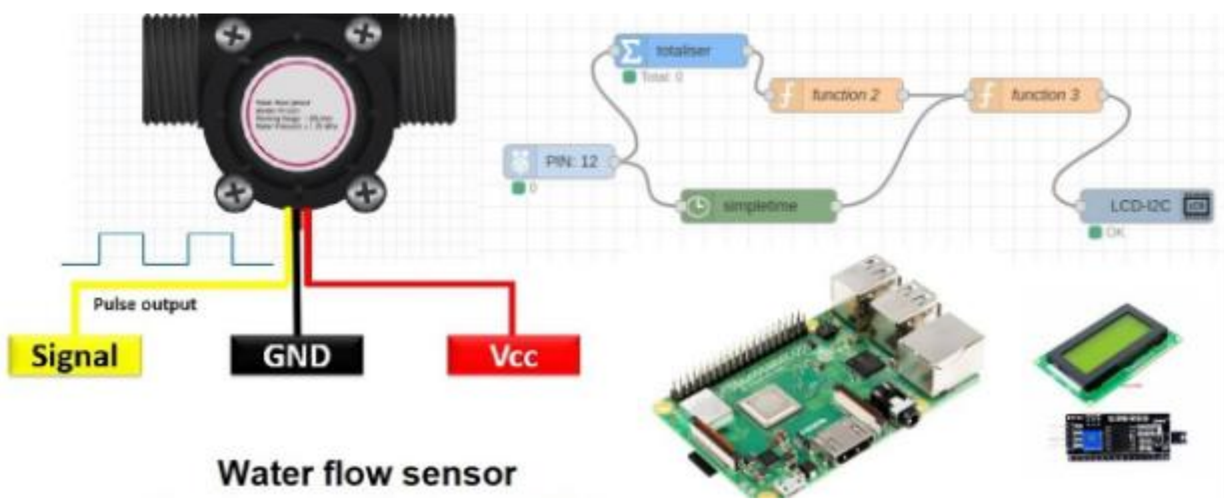
1. Sensors collect data (e.g., water quality, flow rate, water levels).
2. The microcontroller processes the data and sends it to the communication module.
3. The communication module transmits the data to the central server.
4. The central server stores and processes the data, potentially running algorithms to detect anomalies or trends.
5. Data is sent to the cloud platform for further analysis, storage, and remote access.
6. The mobile app connects to the cloud platform and retrieves data.
7. Users can access the app to monitor water parameters and control the water management system.

IoT Devices Utilized:

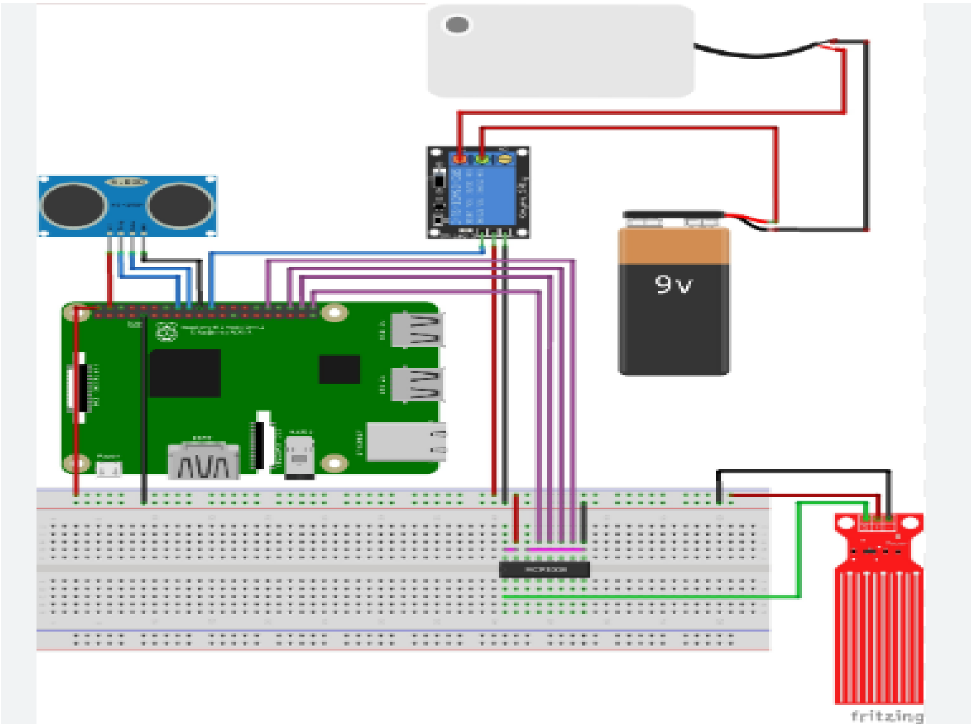
1. **Flow Sensors**: These devices measure water flow rates, offering real-time consumption data.
2. **Water Level Sensors**: Utilized for monitoring water levels in reservoirs and containers.
3. **IoT Microcontrollers**: Raspberry Pi
4. **Wireless Communication Modules**: Ensuring seamless data transmission.
5. **Power Supply**: For reliable and continuous operation.
6. **Data Storage and Analysis**: Utilizing cloud platforms for efficient data management.

8. SENSOR DIAGRAMS:

FLOW SENSOR:



LEVEL SENSOR:



9. USE OF WATER CONSERVATION BY SENSORS:

Sensors and mobile apps play a crucial role in water conservation through smart water management systems. These technologies help monitor, control, and optimize water usage in various ways. Here are some key applications and benefits:

1. **Real-time Monitoring:**

- **Water Quality Monitoring:** Sensors can detect contaminants and changes in water quality, ensuring that water is safe for consumption and industrial processes.
- **Water Flow Monitoring:** Flow sensors measure the rate of water flow in pipes, helping detect leaks, abnormal consumption, and inefficiencies in distribution systems.

2. Leak Detection and Prevention:

- Sensors can identify leaks in water distribution systems and alert users in real-time. Mobile apps can send notifications and provide location details, enabling quick repairs and preventing water wastage.

3. **Water Usage Tracking:**

- Mobile apps allow users to track their water consumption and receive detailed insights into their water usage patterns. This encourages more responsible water use.

4. **Remote Control and Automation:**

- Mobile apps enable users to remotely control water management devices like irrigation systems, pumps, and valves. Users can schedule watering times and adjust settings based on real-time data.

10. CONCLUSION:

In conclusion, smart water systems are a transformative approach to water management that leverage advanced technologies, such as sensors and mobile apps, to enhance the efficiency, sustainability, and reliability of water resources. These systems offer a wide range of benefits for individuals, communities, and the environment:

1. **Water Conservation:** By providing users with insights into their water consumption, enabling remote control of water devices, and offering leak detection capabilities, smart water systems encourage water conservation and reduce water usage.
2. **Environmental Sustainability:** These systems contribute to environmental sustainability by promoting responsible water management, reducing energy consumption, and minimizing water pollution through real-time quality monitoring.
3. **Cost Savings:** Efficient water usage and timely maintenance reduce operational costs for both individuals and utilities, leading to potential cost savings.
4. **Compliance and Regulation**: Smart water systems help ensure compliance with water regulations and standards, reducing the risk of legal and environmental consequences.
5. **Improved Quality of Life:** Access to real-time water data and remote control features through mobile apps provides convenience and peace of mind for users while supporting a higher quality of life.

Smart water systems are a significant step toward addressing the challenges of water scarcity, pollution, and inefficiency. They empower individuals, communities, and utilities to make a positive impact on water resources and the environment. As technology continues to advance, smart water systems are likely to play an increasingly critical role in the sustainable management of one of our planet's most precious resources: water.