SMART WATER MANAGEMENT

TEAM MEMBER

920321205005- Fomitha Fathima M

PHASE 4 - Development Part 2

Project Title: Smart water Management

Introduction:

Determine the purpose and features of your smart water Management.

Decide on the sensors, actuators, and connectivity options.

Create a platform to display real time water consumption data :

HTML (index.html):

<!DOCTYPE html>

<html>

<head>

<title>Real-time Water Consumption</title>

</head>

<body>

<h1>Water Consumption Data</h1>

<div id="waterConsumption">

<p>Real-time data will be displayed here.</p>

</div>

<script src="script.js"></script>

</body>

</html>

JavaScript (script.js):

// Assuming you have a WebSocket server providing real-time data

const socket = new WebSocket("ws://your-websocket-server-url");

// Function to update the water consumption data on the page

function updateWaterConsumption(data) {

const waterConsumptionDiv = document.getElementById("waterConsumption");

waterConsumptionDiv.innerHTML = `<p>Current Water Consumption: ${data} gallons</p>`;

}

// Handle messages from the WebSocket server

socket.addEventListener("message", (event) => {

const data = JSON.parse(event.data);

updateWaterConsumption(data.consumption);

});

// Handle WebSocket connection errors

socket.addEventListener("error", (error) => {

console.error(`WebSocket Error: ${error.message}`);

});

* WebSocket Server:

You will need a server (e.g., Node.js with the ws library) to provide real-time water consumption data. This server should send updates to the connected clients when new data is available.

This is a simplified example, and in a real-world scenario, you would replace the WebSocket URL with the actual server URL that provides real-time water consumption data. Additionally, you should secure your connection (e.g., using WSS for WebSocket over SSL) and handle server-side data collection and distribution.

Remember that displaying real-time data involves complex considerations for data collection, security, scalability, and user authentication, depending on the specific use case. This example provides a basic structure to get you started.

Design the platform to receive and display water consumption data from IOT sensorsensor :

Designing a platform to receive and display water consumption data from IoT sensors and promote water conservation efforts involves several components and considerations. Here's a high-level design:

* 1. IoT Sensors:
* Deploy IoT sensors at various water usage points (e.g., faucets, showers, irrigation systems).
* Sensors should collect real-time water consumption data and send it to a central server.
* 2. Data Collection Server:
* Set up a central server to receive data from IoT sensors.
* The server should process and store incoming data in a database for historical analysis.
* 3. Database:
* Use a database (e.g., SQL or NoSQL) to store water consumption data.
* Organize the data by location, time, and device for efficient retrieval.
* 4. Web Application:
* Create a web application using HTML, CSS, and JavaScript for users to access the data.
* Provide user-friendly dashboards to display real-time and historical water consumption information.
* 5. User Registration and Authentication:
* Implement user registration and login functionality to personalize the experience.
* Users can track their water usage and set conservation goals.
* 6. Real-Time Data Display:
* Use WebSocket or server-sent events to display real-time data on the web application.
* Visualize water consumption data using charts and graphs for easy understanding.
* 7. Alerts and Notifications:
* Implement alerts and notifications to inform users of unusual water usage or when they're close to exceeding their conservation goals.
* 8. Conservation Tips:
* Provide a section with water conservation tips and best practices to educate users.
* 9. Gamification:
* Gamify the platform by introducing challenges and rewards for water conservation achievements.
* 10. Mobile App:
* Develop a mobile app for users to monitor water consumption on the go.
* 11. Community Features:
* Allow users to share their conservation efforts and achievements on social media.
* Create a community forum for users to discuss water-saving strategies.
* 12. Integration with Smart Home Systems:
* Integrate the platform with smart home systems (e.g., IoT-based home automation) to control water-using devices remotely.
* 13. Reporting and Analytics:
* Provide detailed reports and analytics to help users track their progress and make informed decisions.
* 14. Partnerships and Outreach:
* Collaborate with local utilities, environmental organizations, and government agencies to promote water conservation efforts.
* 15. Sustainability and Data Security:
* Ensure the platform's sustainability, including data security, backup, and disaster recovery plans.
* 16. Marketing and Education:
* Develop marketing campaigns and educational materials to raise awareness about water conservation and the benefits of the platform.
* Remember that a successful water conservation platform should be user-friendly, data-driven, and focused on promoting responsible water usage. It should also adapt to changing user needs and environmental conditions while contributing to long-term sustainability.