

SMART WATER MANAGEMENT SYSTEM

NAME: Gnana Vasanth P

ID: au962921104008

EMAIL: vasanthsanjay02@gmail.com

PHASE: 4

INTRODUCTION

Smart water management systems can provide a more resilient and efficient water supply system, reducing costs and improving sustainability. High-technology solutions for the water sector include digital meters and sensors, supervisory control and data acquisition (SCADA) systems, and geographic information systems (GIS).

SMART WATER MANAGEMENT

This explainer is adapted from proceedings of a workshop conducted by the Asian Development Bank (ADB) in Tashkent, Uzbekistan for the water sector. The workshop introduced smart systems and focused on remote monitoring of water networks using smart meters and other instruments.

The IOT has the ability to lessen this worrying picture. Smart Water Monitoring and Management Systems, based on the combination of sensors, big data and AI technologies, can provide to water utility operators, farmers and companies the ability to measure, monitor and control their water distribution networks as well as the quality of the water distributed. Less waste, less consumption, and a better management of the water's quality can improve dramatically the preservation of our planet's resources.

Let's take a look at how Smart water Management systems can help addressing the growing lack of available fresh water...

What are Smart Water Management systems and how do they work?

Smart Water Management is the activity of planning, developing, distributing and managing the use of water resources using an array of IOT technologies which are designed to increase transparency, and make more reasonable and sustainable usage of these water resources.

PLATFORM U1 CODE FOR SMART FOUNDATION

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Smart Water Foundation</title>
```

```
<style>
```

```

/* Style for the water level display */

#water-level {
  Font-size: 24px;
  Font-weight: bold;
}

/* Style for the submit button */
#submit-button {
  Padding: 10px 20px;
  Font-size: 18px;
}

</style>
</head>
<body>
<h1>Smart Water Foundation</h1>
<p>Water Level: <span id="water-level">Loading...</span> cm</p>
<button id="submit-button" onclick="sendDataToServer()">Submit
Data</button>
<script>
// Function to update water level data from the server
Function updateWaterLevel() {
// You can use AJAX or fetch to get data from your server
// Replace the URL with the actual endpoint that provides water level
Data
Fetch('/getWaterLevelData')

.then(response => response.json())
.then(data => {
Document.getElementById('water-level').textContent =

```

```

Data.waterLevel + " cm";

})

.catch(error => {
  Console.error('Error fetching water level data:', error);
});
}

// Function to send data to the server (e.g., to trigger data collection)
Function sendDataToServer() {
  // You can use AJAX or fetch to send data to your server
  // Replace the URL with the actual endpoint that handles data
  Submission
  Fetch('/submitData', { method: 'POST' })
  .then(response => {
    If (response.status === 200) {
      Console.log('Data submitted successfully');
    } else {
      Console.error('Data submission failed with status:',

      Response.status);
    }
  })

  .catch(error => {
    Console.error('Error submitting data:', error);
  });
}

// Update water level initially and then at regular intervals

```

```
updateWaterLevel();
setInterval(updateWaterLevel, 10000); // Update every 10 seconds
</script>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<title>Smart Water Foundation</title>
<style>
/* Style for the water level display */

#water-level {
font-size: 24px;
font-weight: bold;
}
/* Style for the submit button */
#submit-button {
padding: 10px 20px;
font-size: 18px;
}
</style>
</head>
<body>
<h1>Smart Water Foundation</h1>
<p>Water Level: <span id="water-level">Loading...</span> cm</p>
<button id="submit-button" onclick="sendDataToServer()">Submit
Data</button>
<script>
```

```

// Function to update water level data from the server
function updateWaterLevel() {
  // You can use AJAX or fetch to get data from your server
  // Replace the URL with the actual endpoint that provides water level
  data
  fetch('/getWaterLevelData')

  .then(response => response.json())
  .then(data => {
    document.getElementById('water-level').textContent =

    data.waterLevel + " cm";

  })
  .catch(error => {
    console.error('Error fetching water level data:', error);
  });
}

// Function to send data to the server (e.g., to trigger data collection)
function sendDataToServer() {
  // You can use AJAX or fetch to send data to your server
  // Replace the URL with the actual endpoint that handles data
  submission
  fetch('/submitData', { method: 'POST' })
  .then(response => {
    if (response.status === 200) {
      console.log('Data submitted successfully');
    } else {
      console.error('Data submission failed with status:',

```

```
response.status);
```

```
}
```

```
}}
```

```
.catch(error => {
```

```
console.error('Error submitting data:', error);
```

```
});
```

```
}
```

```
// Update water level initially and then at regular intervals
```

```
updateWaterLevel();
```

```
setInterval(updateWaterLevel, 10000); // Update every 10 seconds
```

```
</script>
```

```
</body>
```

```
</html>
```

OUTPUT

Circuit design CoPython code forAPI Keys - ThingFwd: Reset MathMathWorks AccountGitHubThingspeak channelSmart Water Foundation

File | C:/Users/dhine/OneDrive/Documents/Smart%20water%20foundation.html

GmailMapsAll Bookmarks

Smart Water Foundation

Water Level: **20** cm

Submit Data

30°C
Partly sunny

Search

ENG
IN18:23
27-10-2023