

SMART WATER MANAGEMENT SYSTEM USING IOT

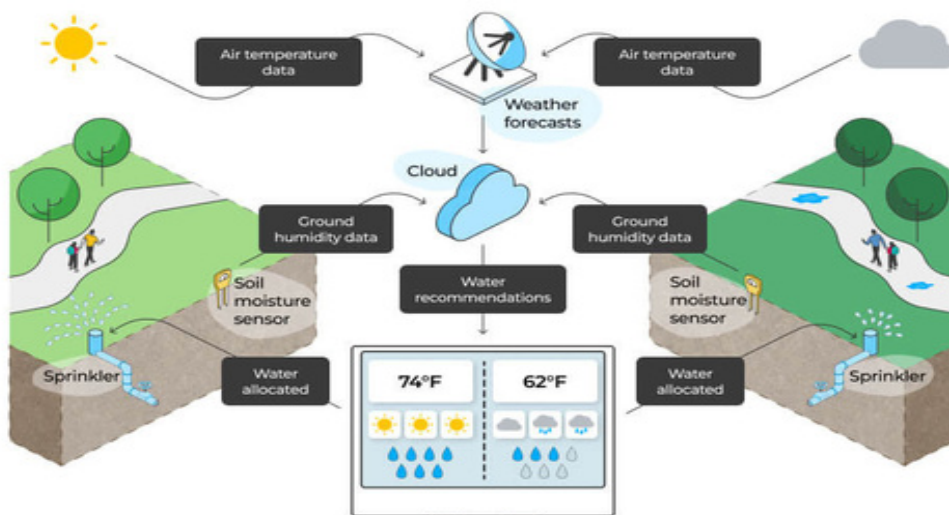
DEVELOPMENT OF SMART WATER :

- IoT water management systems leverage numerous sensors that collect real-time insights on how resources are used.
- These devices transmit the gathered data to the user's application online.
- This information empowers analysis of consumption patterns and encourages more rational water consumption.

SMART CITY WATER MANAGEMENT :

- With the help of sensors, a smart city water management system can enable you to collect real-time data—information that helps you visualize water distribution across the network. Residents with smart meters can make more informed decisions.
-

A smart park irrigation system relies on the soil state, weather forecasts, and current weather conditions



- Water waste and disrupted water supply chains are a drain on the city's budget. IoT can help you watch the health of water equipment and detect problems, like leaks in pipes.

REAL WORLD EXAMPLE :

SMART IRRIGATION OF CITY PARKS :

- The solution calculates the amount of water each area needs depending on the state of the soil, weather forecast, and irrigation calendar. If something goes wrong, such as a leak, the authorities are alerted right away and they're even shown the location.

MAIN BENEFITS :

1. Better transparency in water

2. Fewer incidents
3. Enhanced control over the water supply
4. Saved city budget
5. Improved city sustainability management

SMART DAM MONITORING :

- IoT sensors remotely watch water levels at each dam site. The system sends instant alerts if flooding is possible. With smart capabilities, there's no need to send staff to measure water levels on site. The system even has pre-programmed formulas that replace manual calculations.

MAIN BENEFITS :

- Real-time water level monitoring
- Better dam functionality
- Enhanced dam reliability
- Faster decision-making
- Saved time and resources

PYTHON SCRIPT :

Import necessary libraries

Import time

Import random

Import requests

Simulated sensor data for water level

Def get_water_level():

Return random.uniform(0, 100)

Function to send data to a server

Def send_data_to_server(data):

Server_url = http://example.com/api/water-level

Response = requests.post(server_url, json={"water_level": data})

If response.status_code == 200:

Print("Data sent successfully!")

Else:

```
Print("Failed to send data. Status code:", response.status_code)
```

```
# Main function for smart water management
```

```
Def main():
```

```
    Try:
```

```
        While True:
```

```
            # Read water level data from sensors
```

```
            Water_level = get_water_level()
```

```
            Print("Water Level:", water_level)
```

```
            # Send data to the server
```

```
            Send_data_to_server(water_level)
```

```
            # Wait for a specified interval (e.g., 1 hour)
```

```
            Time.sleep(3600) # 3600 seconds = 1 hour
```

```
    Except KeyboardInterrupt:
```

```
        Print("Smart water management system stopped.")
```

```
# Run the main function
```

```
If __name__ == "__main__":
```

```
    Main()
```

NAME : B.Seethalakshmi

PROJECT TOPIC : SMART WATER MANAGEMENT SYSTEM USING IOT

REGISTER NUMBER :713921106049

COLLEGE CODE :7139

NM ID :au713921106049

EMAIL ID : seethalakshmi0610@gmail.com

