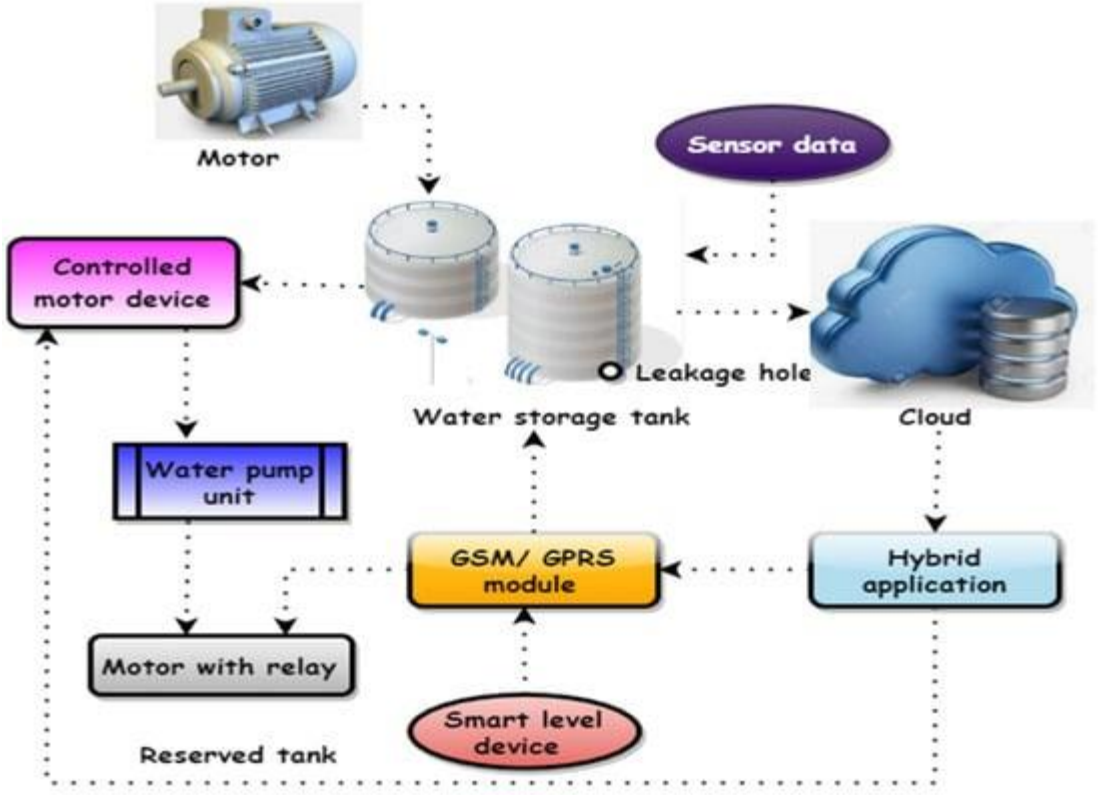
# IoT- Smart Water Management Systems:




## 1. Introduction of Smart Water Management Systems:

One of the essential elements in the universe is water. Nowadays, consumers continuously seek methods to simplify their lives. Monitoring water quality is critical to ensuring the planet's health and long-term viability. Water is the source of many infectious illnesses, and garbage thrown by residents and environmental disasters from industrial enterprises pollute most of the nearby freshwater supplies in SA. Drinking water can be stored in an overhead tank. The principal causes of water quality deterioration in residential buildings are the development of microbes in overhead tanks and distribution networks, corrosion of pipe material, and the non-replacement of existing pipes. To avoid catastrophic health implications, it is necessary to continuously and remotely check the quality parameters of the water system in real-time.

The main contributions of this paper are:

*Smart water management gives a greater understanding of the water system, including flaw detection, preservation, and water management.

*A comprehensive database of regions with water losses or unlawful connections can be built with the introduction of smart water system technology by public service corporations.

*****Smart water grids can save costs by conserving water and energy while improving the quality of service to consumers. Wireless data transfer allows consumers to assess their water use to reduce water costs in other circumstances.

## 2. **Relevant Survey:**

Thermal characteristics have significantly impacted the environmental footprint and life-cycle evaluation of buildings. It is proposed in this study that the opacity of a façade element could be adjusted year-round in response to seasonal variations using the smart water-filled glass (SWFG) control technology described in this research. As a result of SWFG and WFG's climate-based design strategy, glass buildings are no longer seen as climate change liabilities and instead as opportunities for sustainability .According to a new study, the Internet of Things (IoT) could monitor renters' and landlords' electricity and water use (IoT). One technique has made it simpler to read meters and water meters through the internet, and renters and landlords can see the data gathered through an Android application.Tenants and landlords have found the gadget and IoT-based Android app dependable, accurate, useful, and easy to use.

## 3. **Proposed Method**:

In Saudi Arabia, water supply and sanitation are marked by problems and successes. Water shortage is a major issue. Water desalination, water distribution, sewerage, and wastewater treatment have received significant investments to combat water shortages .Today, desalination provides roughly half of the country's drinking water, groundwater mining provides 40%, and surface water supplies in the mountainous southwest account for 10% .A desalinated water pipeline runs from the Arabian Gulf to Riyadh, Saudi Arabia's capital and largest city. Consumers get water nearly at no cost. Despite these advances, the quality of service remains low, for instance, in terms of supply continuity. The public sector in Saudi Arabia is characterized by a widespread lack of institutional capability and governance. The Saudi vision 2030 addresses this challenge to be resolved.

## 4. **Numerical Outcome:**

Internet of Things-based smart water systems can assist prevent these scenarios from occurring and repair the harm that has already been done due to the careless use of water resources. From a freshwater reservoir to the collection and recycling of wastewater, smart water technology makes the entire water supply chain more transparent and controllable. This section includes applications for water management, IoT devices for water management, and smart water treatment methods. Finally, a list of several vital qualities of these systems is framed based on the survey, all of which should be integrated into future designs. As shown in Figure .modern IoT-based water quality monitoring solutions have been thoroughly surveyed. All articles focused on meeting the minimum WHO/USEPA standards for potable water, which need systems to be low-cost, portable, low-power, Internet-enabled, real-time, and compliant with these standards.

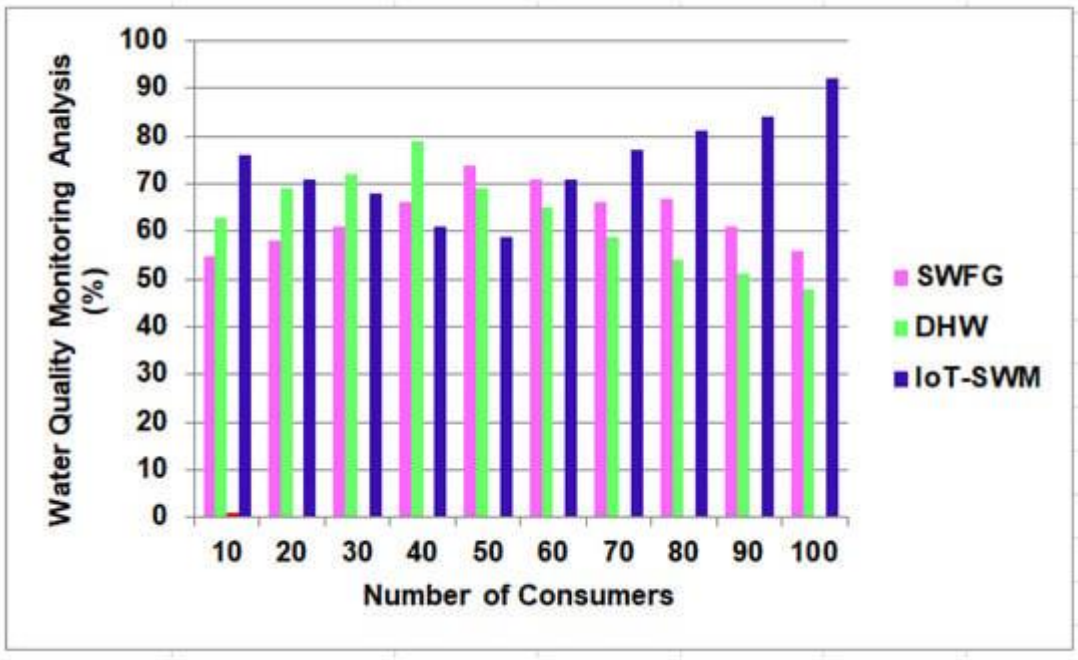



**Figure :**Water quality monitoring analysis.

## 5. **Conclusions:**

The water sector has been grappling with creating an efficient and long-lasting water system. It is included in the IoT-SWM. People intend to broadcast more data to the cloud and analyze it further to construct some algorithm to determine the tank's lifespan and the proper aspects of leaking. Procedures and actions are determined depending on the threshold, capital cost, and the accessibility of equipment and materials. Even though statistically minimal water savings can be achieved using in-line flow restrictors, they can be much more cost-effective than water-efficient taps in certain situations. If they have been installed as part of normal maintenance visits, the expenses would be lower. They are a low-cost alternative to outdated toilets and are unlikely to save a lot of water. When it the time comes to renovate restrooms, installing water-saving toilets should be considered.

**Coding**:

In every route which requires login , just put if logged_in:

# Logout button that send request to /logout

```
from flask import Flask, render_template, url_for
from flask import jsonify, request
from flask import flash, redirect,abort

import datetime as d
app = Flask(__name__)
```

```
lastOffTime = d.datetime.now()
netlitres = 0
prevtank1 = 10
diff = 0
x = 0
logged_in = False

power =0
tank1data = 40
tank2data = 40
current =  "OFF"
option = ""
ontime = 0
stat1 = ""
stat2 = ""
stat3 = ""

'''@app.route('/',  methods=['GET'])
def index():

    global tank1_level
    return render_template('index_gauge.html')  '''

@app.route('/')
def home():
    if logged_in:
        return render_template('home.html')
    else:
        return redirect('/login')
    lastOffTime = d.datetime.now()

@app.route('/login', methods=['GET'])
```

```
def login_page():
    return render_template('login.html')
@app.route('/logout', methods=['GET'])
def logout():
    global logged_in
    logged_in = False
    return render_template('login.html')
@app.route('/login', methods=['POST'])
def check_login():
    global logged_in
    if request.form['password'] == 'password' and request.form['username'] == 'admin':
        logged_in = True
        return redirect('/')
    else:
        return render_template('login.html')

@app.route('/water', methods=['GET'])
def water():
    if logged_in:
        return render_template('index_gauge.html')
    else:
        return redirect('/login')

@app.route('/energy', methods=['GET'])
def energy():
    if logged_in:
        return render_template('energy.html')
    else:
        return redirect('/login')

'''@app.route('/power/<int:p>', methods=['GET'])
def power():
```

```
    global power
    power = p
    return 'ok' '"

@app.route('/deptho/<int:depth_cm1>', methods=['GET'])
def show_post1(depth_cm1):
    global tank1data
    global netlitres
    global prevtank1
    global diff
    if depth_cm1 < prevtank1:
        diff = (prevtank1 - depth_cm1)
    tank1data = depth_cm1
    prevtank1 = depth_cm1
    netlitres = netlitres + diff
    return 'ok'

@app.route('/power/<int:p>', methods=['GET'])
def power(p):
    global power
    power = p
    return 'ok'



@app.route('/change/<string:switch>', methods=['GET','POST'])
def change(switch):
    global option
    option = switch
    return option

@app.route('/stat/<int:p>', methods=['GET'])
def status(p):
```

```
global ontime
global tank1data
global tank2data
global current
global option
global stat1,stat2,stat3
global power
power = p
s1 =""
s2 =""
s3 = ""

if tank1data<20 and tank2data>20 and option == "auto":
    s1 = "xyz"
    stat1="a"
    current = "On"
elif option == "on":
    s2 = "abc"
    stat2 = "a"
elif option == "off":
    s2 = ""
    stat2 = ""
elif option == "a2on":
    s3 = "pqr"
    stat3 = "a"
elif option == "a2off":
    s3 = ""
    stat3 = ""
elif option == "a1on":
    s1 = "xyz"
    stat1 = "a"
    current = "On"
```

```
    elif option == "a1off":
        s1 = ""
        stat1  = ""
        current = "Off"
    pfinal = s1 + s2 + s3
    if stat1 =="a":
        pfinal = pfinal + "xyz"
    if stat2 == "a":
        pfinal = pfinal + "abc"
    if stat3 == "a":
        pfinal = pfinal + "pqr"
    final = pfinal + "$"
    return final


@app.route('/depths/<int:depth_cm2>', methods=['GET'])
def show_post2(depth_cm2):
    global tank2data
    tank2data = depth_cm2
    return 'ok'

@app.route('/return_global', methods=['GET'])
def return_global():
    global tank1data
    global tank2data
    global current
    global ontime
    global lastOffTime
    global netlitres
    global x
    global power
```

```
    x = x+ 1
    if current == "OFF":
        lastOffTime = d.datetime.now()
    elif current == "On":
        if  0==0:
            f= 0
            diff = d.datetime.now() - lastOffTime
            lastOffTime = d.datetime.now()
            f = diff.microseconds
            ontime = ontime + f

    return jsonify(tank1 = tank1data , tank2 = tank2data, stat = current, time = ontime/1000000, net= netlitres)

if __name__ == "__main__":

            app.run(host='0.0.0.0', port=8080, debug=True)
```