KGiSL Institute of Technology

(Affiliated to ANNA University, Chennai and Approved by AICTE, New Delhi)

365, KGiSL Campus, Thudiyalur Road, Saravanampatti
Coimbatore – 641035

Department of Artificial Intelligence and Data Science

Name	**: Mathew Fedrick I**
Register Number	**: 711721243060**
Regulation	**: R-2021**
Branch	**: B.Tech -Artificial Intelligence and Data Science**
Project Title	**: Smart Water System**
Semester/ Year	**: V / III**

Project Title: Smart Water System

Phase 4: Development Part 2 - Building the Data-Sharing Platform

Introduction:

The goal of this phase is to create a real-time data-sharing platform for monitoring water consumption using IoT sensors. The platform provides a web-based interface for users to view and track real-time water consumption data.

Technology Stack:

The platform leverages several technologies to achieve real-time data sharing:

- **Flask:** A lightweight Python web framework for building web applications.
- **Flask-SocketIO:** An extension for Flask that simplifies WebSocket integration for real-time communication.
- **HTML and JavaScript:** Used for building the front-end interface.
- **Socket.IO:** A real-time, bidirectional communication library for web applications.

Data Sources:

The platform is designed to receive real-time water consumption data from IoT sensors. These sensors could include flow sensors, smart water meters, ultrasonic sensors and

Server-Side (Python) Code:

The Flask server-side code is responsible for:

Creating a Flask application and configuring it with a secret key.

Setting up a SocketIO instance associated with the Flask app.

Defining a function get_realtime_water_consumption() to retrieve real-time water consumption data from the IoT sensors. This function may involve interfacing with the sensors and returning the data.

Handling the root route ("/") to serve the HTML template to clients.

Setting up a SocketIO event handler for 'connect', which continuously emits real-time water consumption data to clients every 5 seconds.

Python code :

```
from flask import Flask, render_template
from flask_socketio import SocketIO

app = Flask(__name__)
app.config['SECRET_KEY'] = 'your-secret-key'
socketio = SocketIO(app)

# Function to retrieve real-time water consumption data from IoT sensors
def get_realtime_water_consumption():
    # Replace this with your code to connect to the sensors and retrieve data
    # You may need to use a library or API provided by your sensor hardware or platform.
    # For demonstration purposes, we'll use a placeholder value.
    return 5.3  # Replace this with actual data retrieval logic.

@app.route('/')
def index():
    return render_template('index.html')

@socketio.on('connect')
def handle_connect():
    print('Client connected')
    while True:
        water_consumption = get_realtime_water_consumption()
        socketio.emit('water_consumption', {'water_consumption':
water_consumption})
        socketio.sleep(5)  # Update data every 5 seconds (adjust as needed)

if __name__ == '__main__':
    socketio.run(app, debug=True)
```

Front-End (HTML and JavaScript) Code:

The front-end code is responsible for:
Loading the Socket.IO library for WebSocket communication.
Establishing a WebSocket connection to the server upon page load.
Listening for 'water_consumption' events from the server, which contain real-time water consumption data.
Updating the displayed water consumption value in real-time on the web page.

HTML Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Real-time Water Consumption</title>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.0.1/socket.io.min.js
"></script>
    <script>
        document.addEventListener('DOMContentLoaded', () => {
            var socket = io.connect('http://' + document.domain + ':' +
location.port);
            socket.on('water_consumption', function(msg) {
                var consumptionElement =
document.getElementById('consumption');
                consumptionElement.innerHTML =
msg.water_consumption.toFixed(2) + ' gallons';
            });
        });
    </script>
</head>
<body>
    <h1>Real-time Water Consumption</h1>
    <p>Current Water Consumption: <span id="consumption">0.00
gallons</span></p>
    <!-- Add elements for water conservation tips and challenges here -->
</body>
</html>
```

Data Flow:

When a user accesses the platform in a web browser, the server sends the HTML template to the client.
The client-side JavaScript code establishes a WebSocket connection to the server using Socket.IO.
The server periodically retrieves real-time water consumption data from the IoT sensors using the get_realtime_water_consumption() function.
The server emits this data to all connected clients using the 'water_consumption' event.
Clients receive the data and update the displayed water consumption value in real-time.

Conclusion :

In conclusion, our real-time water consumption monitoring platform, built on Flask, Flask-SocketIO, HTML, and JavaScript, offers a robust solution for tracking water usage from diverse IoT sensors. This platform bridges the gap between IoT data sources and end-users through a user-friendly web interface, ensuring immediate access to vital water consumption data. Its adaptability to various sensor types makes it highly versatile. By implementing real-time communication with the help of Socket.IO, we enable users to stay informed and take proactive measures for efficient water management. This platform serves as a solid foundation for enhancing water conservation efforts and driving sustainability.