 **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**

**:**

**Haris Nihaal.S.A**

**Register Number**

**:**

**711721243701**

**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 5:**Development Part 3 - Building the Data-Sharing Platform

**Introduction:**

The Smart Water Management project aims to develop a real-time water consumption monitoring system using IoT sensors, a Raspberry Pi, and a mobile app. The system will help users to track their water usage and identify areas where they can conserve water. It will also provide insights into water consumption trends and patterns, which can be used to develop sustainable water management practices.

**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.

**IoT Sensor Setup:**

The IoT sensor setup for the Smart Water Management project consists of the following components:

Flow sensor: Measures the rate of water flow in a pipe.

Water level sensor: Measures the water level in a tank or reservoir.

Temperature sensor: Measures the temperature of the water.

pH sensor: Measures the pH level of the water.

The sensors are connected to a Raspberry Pi using a variety of communication interfaces, such as I2C, SPI, and UART. The Raspberry Pi is responsible for collecting and processing the sensor data.

**Mobile App Development:**

The mobile app for the Smart Water Management project is developed using a cross-platform development framework, such as React Native or Flutter. The app provides users with a real-time view of their water consumption data. It also allows users to set water conservation goals and track their progress.

**Raspberry Pi Integration:**

The Raspberry Pi in the Smart Water Management project plays a central role in collecting, processing, and transmitting the sensor data. The Raspberry Pi also hosts the mobile app backend API, which provides the app with access to the water consumption data.

**Code Implementation:**

The code for the Smart Water Management project is implemented using Python. The Python code is responsible for the following tasks:

Collecting and processing the sensor data.

Transmitting the sensor data to the mobile app backend API.

Hosting the mobile app backend API.

Generating real-time water consumption reports.

**Python code :**

import board

import busio

import adafruit\_ina219

# Create a bus object using the default I2C pins

i2c = busio.I2C()

# Create an INA219 object using the I2C bus

ina219 = adafruit\_ina219.INA219(i2c)

# Get the current water consumption data

current\_consumption = ina219.current

# Print the current water consumption data

print("Current water consumption:", current\_consumption, "A")

import requests

# Set the URL of the mobile app backend API

api\_url = "https://api.example.com/water\_consumption"

# Create a JSON object containing the sensor data

json\_data = {

"current\_consumption": current\_consumption

}

# Send a POST request to the mobile app backend API with the sensor data

response = requests.post(api\_url, json=json\_data)

# Check the response status code to make sure the data was successfully transmitted

if response.status\_code == 200:

print("Sensor data successfully transmitted to mobile app backend API")

else:

print("Error transmitting sensor data to mobile app backend API:", response.status\_code)

from flask import Flask, request, jsonify

app = Flask(\_name\_)

@app.route("/water\_consumption", methods=["POST"])

def water\_consumption():

# Get the sensor data from the request body

current\_consumption = request.json["current\_consumption"]

# Save the sensor data to a database or other data storage system

# Return a JSON response to the client

return jsonify({

"message": "Sensor data successfully received"

})

if \_name\_ == "\_main\_":

app.run(host="0.0.0.0", port=5000)

import time

# Set the interval at which to generate the water consumption reports

report\_interval = 60 \* 5 # 5 minutes

# Create a list to store the water consumption data

water\_consumption\_data = []

# Start a loop to continuously generate water consumption reports

while True:

# Get the current water consumption data

current\_consumption = ina219.current

# Add the current water consumption data to the list

water\_consumption\_data.append(current\_consumption)

# Check if the interval has passed

if time.time() - report\_time >= report\_interval:

# Calculate the average water consumption over the interval

average\_consumption = sum(water\_consumption\_data) / len(water\_consumption\_data)

# Generate a water consumption report

water\_consumption\_report = {

"average\_consumption": average\_consumption

}

# Save the water consumption report to a file or database

with open("water\_consumption\_report.json", "w") as f:

json.dump(water\_consumption\_report, f)

# Reset the list of water consumption data

water\_consumption\_data = []

# Reset the report time

report\_time = time.time()

# Sleep for the remainder of the interval

time.sleep(report\_interval - (time.time() - report\_time))

**The real-time water consumption monitoring system can promote water conservation and sustainable practices in the following ways:**

Awareness: The system provides users with real-time visibility into their water usage. This awareness can help users to identify areas where they can conserve water.

Goal setting: The system allows users to set water conservation goals. This helps users to track their progress and stay motivated to conserve water.

Insights: The system provides insights into water consumption trends and patterns. This information can be used to develop sustainable water management practices.

Conclusion

The Smart Water Management project is a comprehensive solution for monitoring water consumption using IoT sensors, a Raspberry Pi, and a mobile app. The system can help users to conserve water and develop sustainable water management practices

**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.