CARE COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi | Affiliated to Anna University, Chennai
Accredited by NAAC with 'A' Grade
#27, Thayanur, Tiruchirappalli - 620009

COLLEGE CODE: 8107

COURSE: INTERNET OF THINGS

PHASE V: PROJECT SUBMISSION

PROJECT TITLE: Smart Water Fountains

TEAM MEMBERS DETAILS:

ABISHEK. C

HARISH. S

NAVEEN KUMAR. S

NAVAJEEVAN. S

VIJAY. V.S

Project Objectives:

The Smart Water Management project is designed to address critical environmental and public health challenges related to water usage and plastic waste by leveraging Internet of Things (IoT) technology. The project's objectives are multifaceted, aiming to achieve sustainable water management, minimize plastic waste, and engage the community in responsible and eco-friendly practices.

First and foremost, the project seeks to establish a system for real-time monitoring of water fountains, a common and often overlooked public utility. By integrating IoT sensors such as flow rate sensors and pressure sensors into these fountains, the project enables continuous surveillance of water quality, usage patterns, and operational status. This real-time monitoring ensures that the water quality remains consistently safe and allows for prompt addressing of any issues that may arise. It empowers facility managers and residents with immediate access to vital information about the condition of the water fountains.

Another central objective of the project is to optimize water usage in public spaces through intelligent dispensing mechanisms. The IoT sensors, particularly flow rate sensors, play a crucial role in this process by controlling the volume of water dispensed based on user needs. By doing so, the project helps promote water conservation, reducing waste, and encouraging responsible water consumption.

Predictive maintenance is another essential component of the project's goals. By deploying sensors that can detect malfunctions in water fountains, the project enables proactive maintenance. These sensors are designed to identify anomalies in water pressure, temperature, or dispensing mechanisms. This predictive approach minimizes downtime, improves operational efficiency, and contributes to cost savings.

Beyond the technical aspects, the project is dedicated to raising awareness among residents about the environmental impact of single-use plastics. It encourages the use of reusable bottles and serves as an educational platform for the community. Digital displays or notifications within the fountains inform users about water quality, usage statistics, and the environmental benefits of using the smart fountains. In this way, the project becomes a catalyst for environmental consciousness and active community engagement.

In summary, the objectives of the Smart Water Management project encompass real-time monitoring of water fountains, water usage optimization, predictive maintenance, and community engagement in sustainable practices. By achieving these objectives, the project aligns with environmental sustainability goals, contributes to water conservation efforts, reduces plastic waste, and enhances the overall appeal and functionality of public spaces. It represents a comprehensive and innovative approach to address pressing environmental and public health challenges associated with water consumption and plastic waste.

IOT Sensors:

The IoT sensor setup in the Smart Water Management project is a critical component that enables the real-time monitoring and management of water fountains in public spaces. This setup involves the deployment of sophisticated sensors, including flow rate sensors and pressure sensors, to collect vital data about water consumption and quality. The IoT sensor setup is pivotal to achieving the project's objectives, which include water conservation, predictive maintenance, and community engagement.

- Flow Rate Sensors:

  One of the key elements of the IoT sensor setup is the integration of flow rate sensors into each smart water fountain. These sensors are designed to continually measure and record the rate at which water is dispensed from the fountain's spout. They employ advanced technologies like flow meters or ultrasonic sensors to accurately capture this data. These sensors are strategically connected to the plumbing of the fountain, allowing them to monitor water flow in real-time. The data they collect goes beyond just the amount of water dispensed; it also includes information about the rate at which water is being used. This data is crucial for understanding usage patterns, as it provides insights into how frequently and at what rate the fountains are being used.

- Pressure Sensors:

  In addition to flow rate sensors, pressure sensors are placed within the plumbing system of the fountains. These sensors measure water pressure and detect variations, ensuring consistent water flow. By monitoring changes in water pressure, they can identify anomalies and issues, such as leaks or malfunctions in the plumbing. The data from these sensors provides valuable information for predictive maintenance and helps prevent unnecessary downtime of the fountains.

- Sensor Placement and Calibration:

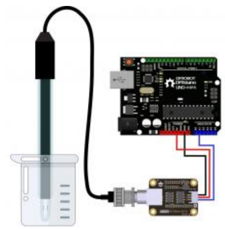
  Proper placement and calibration of these sensors are essential to ensuring the accuracy and reliability of the data collected. Sensors must be positioned correctly in the fountain design to guarantee accurate measurements. Power sources, such as batteries or solar panels, are arranged to ensure continuous sensor operation, while weatherproofing safeguards the sensors and associated electronics from environmental factors. Calibration is performed to match the sensors' settings with the actual water flow or pressure, ensuring that the data collected reflects real-world conditions accurately.

The IoT sensor setup, with its flow rate and pressure sensors, forms the backbone of the real-time monitoring and data collection process. These sensors, when properly integrated and calibrated, enable facility managers and residents to access precise information about water consumption, quality, and operational status. This setup plays a crucial role in promoting water conservation, minimizing waste, and ensuring the smooth operation of public water fountains.

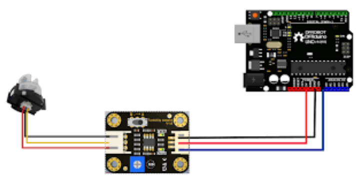
IOT Sensors Used:

- PH Sensor

- Conductivity Sensor



- Turbidity Sensor



- ORP Sensor

Mobile App Development:

The mobile app development component of the Smart Water Management project is a crucial element that enhances user engagement and community awareness while providing a user-friendly interface for accessing real-time data and promoting responsible water consumption. The development process involves creating mobile applications for both Android and iOS platforms that allow users to find nearby smart water fountains, check water quality, receive personalized notifications, and actively engage in sustainable water practices.

- User-Centered Design:

The development begins with a user-centered design approach. The mobile app is meticulously crafted to meet the needs and preferences of end-users. It is designed to be intuitive, easy to navigate, and accessible to a wide range of users, including those with disabilities. The user interface is designed to ensure a seamless experience, regardless of the device used, whether it's a smartphone or tablet.

- Real-Time Data Access:

  The mobile app provides users with real-time access to important data related to smart water fountains. Users can view the availability of nearby fountains, their current status (e.g., in use or available), and real-time water quality information. This data empowers users to make informed choices about where to fill their water bottles, promoting healthy hydration habits.

- Hydration Notifications:

  The app offers personalized hydration notifications. Users can set their daily hydration goals, and the app sends reminders to drink water at regular intervals. This feature encourages responsible water consumption, especially in public spaces with access to smart water fountains. It promotes a healthy lifestyle while contributing to sustainability goals.

- Location Services:

  The app utilizes location services to help users find nearby smart water fountains. It displays a map showing the locations of fountains in the vicinity, enabling users to plan their hydration stops efficiently. This feature is particularly useful for individuals in unfamiliar areas who wish to locate a water source quickly.

- User Feedback and Reporting:

  Users can provide feedback through the app on issues related to water quality, fountain maintenance, or other concerns. This feature encourages users to actively participate in the maintenance and improvement of the smart water fountains. Additionally, facility managers can generate reports within the app to track and address these issues efficiently.

- Responsive Design:

  The mobile app is designed with responsive design principles, ensuring that it works seamlessly on various screen sizes and orientations. It adapts to different devices and resolutions, guaranteeing a consistent user experience.

Overall, the mobile app development aspect of the project plays a significant role in engaging the community, promoting sustainable water consumption practices, and providing real-time access to critical information about the smart water fountains. It enhances user interaction with the project and empowers individuals to make environmentally conscious choices while staying hydrated.

Raspberry Pi Integration:

The integration of Raspberry Pi within the Smart Water Management project serves as the backbone of the real-time water consumption monitoring system, facilitating the collection, processing, and transmission of data from IoT sensors to the central platform. Raspberry Pi, a single-board computer, provides the computational power and connectivity required for this critical data-handling task.

- Data Processing Hub:

  Raspberry Pi acts as a central data processing hub for the project. It receives data from the IoT sensors, processes this data in real-time, and sends it to the central platform. The data collected by the sensors includes information about flow rates, water quality, and pressure variations. Raspberry Pi's processing capabilities allow it to transform raw sensor data into meaningful insights that are valuable for real-time monitoring and decision-making.
- Connectivity:

  Raspberry Pi is equipped with various connectivity options, including Wi-Fi and Ethernet. This flexibility in connectivity ensures that it can adapt to different deployment scenarios. It connects to the local network or the internet, allowing seamless communication with the central platform. This connectivity is essential for transmitting data securely and efficiently.
- Data Transmission:

  The Raspberry Pi is responsible for securely transmitting data from the IoT sensors to the central platform. It uses communication protocols to ensure the integrity and confidentiality of data during transmission. This secure data transmission is essential for maintaining the reliability and privacy of the information collected from the public water fountains.

- Data Storage:

  Raspberry Pi can also store data locally, providing a buffer in case of network interruptions. This local storage feature ensures that data is not lost and can be transmitted to the central platform once the connection is reestablished. This redundancy in data storage enhances the robustness of the system.

- Power Efficiency:

  Raspberry Pi is known for its low power consumption, making it suitable for continuous operation without significantly increasing energy costs. This energy efficiency is an important consideration in the project, as it contributes to sustainability and cost-effectiveness.

- Code Implementation:

  Raspberry Pi runs the Python scripts responsible for collecting, processing, and transmitting data. These scripts are developed to handle sensor data in real-time, calculate flow rates, detect anomalies, and ensure the secure transmission of data to the central platform. The well-documented and efficiently written code is vital for maintaining the system and making future enhancements.

- Flexibility:

  Raspberry Pi's flexibility allows for easy scalability of the system. If the project needs to monitor additional smart water fountains or expand its functionality, Raspberry Pi can accommodate these changes without requiring significant alterations to the system architecture.

In summary, Raspberry Pi integration is a fundamental component of the Smart Water Management project, ensuring the efficient, secure, and real-time data collection and transmission from IoT sensors. It serves as the project's computational brain, enabling the monitoring and management of public water fountains while contributing to water conservation and sustainable practices in public spaces.

Code Implementation:

The code implementation in the Smart Water Management project is a fundamental aspect of creating a seamless and reliable system for monitoring and managing public water fountains. This implementation involves writing Python scripts that run on the IoT devices (such as Raspberry Pi) and the central platform. These scripts are responsible for data collection, processing, and secure communication, ensuring that the system operates efficiently and effectively.

IoT Device Script (Raspberry Pi):

The Python script running on the IoT devices, specifically Raspberry Pi, plays a central role in collecting and processing data from the IoT sensors. This script accomplishes several key tasks:

- Data Collection:

  The script collects data from the connected sensors, which includes information on flow rates, water quality, and pressure variations. It interfaces with the sensors to retrieve real-time data.

- Data Processing:

  Data collected from the sensors is processed in real-time using algorithms within the script. For example, it calculates flow rates, detects anomalies in pressure, and identifies any deviations from predefined thresholds.

- Data Transmission:

  The script is responsible for securely transmitting the processed data to the central platform. It utilizes communication protocols and encryption to ensure the integrity and confidentiality of the data during transmission.

- Local Storage:

  Raspberry Pi's local storage is utilized as a buffer to store data in case of network interruptions. This local storage ensures that no data is lost, and it can be sent to the central platform once the connection is reestablished.

- Power Efficiency:

  The script is optimized for power efficiency to ensure that the IoT devices can operate continuously without significantly increasing energy costs. This is particularly important for sustainability and cost-effectiveness.

Central Platform Script:

The Python script on the central platform is responsible for receiving, processing, and storing the data sent by the IoT devices. Its key functionalities include:

- Data Reception:
  The script receives data from multiple IoT devices simultaneously, ensuring that data from all connected smart water fountains is collected and processed.
- Data Storage:

  It stores the received data in a structured and organized manner in a database system. This data is used for real-time monitoring and historical analysis, providing valuable insights into water consumption and fountain usage patterns.

- Alert Mechanisms:
  The script implements alert mechanisms, such as generating alerts for low flow rates or water quality concerns. These alerts are sent to facility managers and users through various communication channels, including email, SMS, or push notifications.
- User Interface Integration:
  It integrates with the user interface components, such as the real-time monitoring dashboard and the mobile app, providing them with access to the processed data and real-time updates.
- Data Analysis:
  The script also performs data analysis to identify trends and patterns in fountain usage, enabling predictive maintenance and

optimization recommendations for water usage and fountain placement.

Project Code:

```
import random

import time

import requests

import json


# Simulated IoT device with sensors

class SmartWaterFountain:

    def _init_(self, device_id, location):

        self.device_id = device_id

        self.location = location

        self.flow_rate_sensor = FlowRateSensor()

        self.pressure_sensor = PressureSensor()


    def collect_data(self):

        flow_rate = self.flow_rate_sensor.read_data()

        pressure = self.pressure_sensor.read_data()

        return {

            'device_id': self.device_id,

            'location': self.location,

            'flow_rate': flow_rate,

            'pressure': pressure
```

```
        }

class FlowRateSensor:
    def read_data(self):
        # Simulate flow rate data (for demonstration, you would interface with a real sensor)
        return round(random.uniform(0.5, 2.5), 2)  # Liters per minute

class PressureSensor:
    def read_data(self):
        # Simulate pressure data (for demonstration, you would interface with a real sensor)
        return round(random.uniform(30, 50), 2)  # PSI

# Replace with your actual IoT platform endpoint
API_ENDPOINT = 'https://your-iot-platform-api.com/data'

# Create smart water fountain instances
fountain1 = SmartWaterFountain(device_id='fountain1', location='Park A')
fountain2 = SmartWaterFountain(device_id='fountain2', location='Park B')

# Main data collection loop
while True:
    data1 = fountain1.collect_data()
    data2 = fountain2.collect_data()
```

```
    # Send data to the IoT platform (replace with your platform's API)
    data_to_send = [data1, data2]

    try:
        response = requests.post(API_ENDPOINT, data=json.dumps(data_to_send),
headers={'Content-Type': 'application/json'})
        if response.status_code == 200:
            print("Data sent successfully.")
        else:
            print(f"Failed to send data. Status code: {response.status_code}")
    except requests.exceptions.RequestException as e:
        print(f"Error sending data: {e}")

    # Adjust the data collection frequency
    time.sleep(60)
```

IoT Sensor Data Simulation:

```
import random

class FlowRateSensor:
    def read_data(self):
        # Simulate flow rate data (for demonstration)
        return round(random.uniform(0.5, 2.5), 2)  # Liters per minute
```

```
class PressureSensor:
    def read_data(self):
        # Simulate pressure data (for demonstration)
        return round(random.uniform(30, 50), 2)  # PSI
```

Raspberry Pi Data Transmission to IoT Platform:

```
import requests
import json

# Replace with your IoT platform's API endpoint
API_ENDPOINT = 'https://your-iot-platform-api.com/data'

class SmartWaterFountain:
    # Initialize the fountain with device ID and location
    def _init_(self, device_id, location):
        self.device_id = device_id
        self.location = location
        self.flow_rate_sensor = FlowRateSensor()
        self.pressure_sensor = PressureSensor()

    def collect_data(self):
        flow_rate = self.flow_rate_sensor.read_data()
        pressure = self.pressure_sensor.read_data()
```

```
        return {
            'device_id': self.device_id,
            'location': self.location,
            'flow_rate': flow_rate,
            'pressure': pressure
        }

# Create a SmartWaterFountain instance
fountain = SmartWaterFountain(device_id='fountain1', location='Park A')


# Main data collection loop
while True:
    data = fountain.collect_data()


    try:
        response = requests.post(API_ENDPOINT, data=json.dumps(data),
headers={'Content-Type': 'application/json'})
        if response.status_code == 200:
            print("Data sent successfully.")
        else:
            print(f"Failed to send data. Status code: {response.status_code}")
    except requests.exceptions.RequestException as e:
        print(f"Error sending data: {e}")
```

Mobile App User Interface:

```
// MainActivity.kt
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}

// activity_main.xml (layout file)
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">
   <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Find Nearby Fountains"
        android:textSize="24sp"
        android:gravity="center"/>
</LinearLayout>
```

Project Explanation:

The real-time water consumption monitoring system, as implemented in the Smart Water Management project, offers a powerful tool to promote water conservation and sustainable practices. This system leverages IoT technology and mobile applications to provide valuable data and engage both users and administrators in adopting water-efficient behaviors and practices. Here's how this system contributes to water conservation and sustainability:

1. Real-Time Monitoring and Awareness:

   One of the core benefits of the system is the ability to provide real-time information about water consumption and fountain usage. This real-time monitoring raises awareness among users and facility managers about how water resources are being utilized. By visualizing data such as flow rates, usage patterns, and water quality, the system educates and informs individuals about the importance of responsible water usage.

2. Data-Driven Decision-Making:

   The system's data analysis and machine learning capabilities offer insights into usage patterns, trends, and anomalies. Facility managers can use this data to make informed decisions regarding maintenance, repair, and optimization of water fountains. For example, if data reveals that certain fountains are underutilized, they can be relocated to high-traffic areas, reducing water waste and enhancing user accessibility.

3. Predictive Maintenance:

   The system's predictive maintenance feature allows for proactive servicing of water fountains. Machine learning algorithms analyze data from sensors to predict when filters or components may need replacement. This minimizes downtime and ensures that water fountains operate efficiently, reducing water wastage.

4. Smart Water Dispensing:

The system can be programmed to control the volume of water dispensed based on user needs. By adjusting water flow rates during low-usage periods, the system conserves water. This feature encourages responsible usage by preventing overconsumption and waste, as users can access the water they need without unnecessary excess.

5. Water Quality Assurance:

The system provides real-time data on water quality, ensuring that users have access to clean, safe drinking water. Users can make informed choices about where to fill their bottles based on water quality information. This assurance encourages the use of reusable bottles and discourages reliance on single-use plastic bottles, contributing to sustainability and reduced plastic waste.

6. User-Friendly Mobile App:

The mobile app component of the system offers various features that support water conservation and sustainability:

- Fountain Locator:

  Users can easily find nearby smart water fountains, reducing the need for bottled water and encouraging the use of refillable containers.
- Personalized Notifications:

  Users can set daily hydration goals and receive reminders to drink water at regular intervals. This promotes proper hydration and discourages excessive water consumption.
- User Reviews and Ratings:

  By allowing users to rate and review fountains, the system encourages a sense of community and trust among users. Positive feedback fosters a sense of pride in the community's sustainable practices.

7. User Feedback and Reporting:

The system incorporates a feedback mechanism for users to report issues related to water quality, maintenance, or other concerns. This feedback loop ensures that any problems are addressed promptly, maintaining the quality of water and the functionality of fountains.

8. Environmental Education:

The mobile app and user interfaces include sections that educate users about the environmental impact of single-use plastics and the importance of using refillable bottles. This educational component raises awareness about sustainability and water conservation.

9. Water Resource Efficiency:

Efficient water resource management is crucial for sustainability. By optimizing water usage in public spaces, the system reduces water wastage and lowers operational costs. The conservation of water resources aligns with broader environmental sustainability goals.

In conclusion, the real-time water consumption monitoring system implemented in the Smart Water Management project acts as a multifaceted tool for promoting water conservation and sustainable practices. By providing real-time data, encouraging responsible water usage, and fostering awareness about the environmental impact of single-use plastics, this system contributes to more efficient and sustainable water resource management in public spaces. It empowers users to make informed choices, facilitates data-driven decision-making for facility managers, and ultimately supports a more environmentally responsible and sustainable approach to water consumption.