**SMART WATER FOUNTAINS**

**PROBLEM DEFINITION**

An IoT based smart water fountain refers to a network of interconnected devices equipped with sensors designed to control waterflow and detect malfunctions. The fountain is equipped with sensors that monitor various aspects of its operation, such as water level, water quality (e.g., pH and turbidity), temperature. Data collected from the fountain can be analyzed to provide insights into water usage, maintenance needs, and overall performance. This information can be valuable for optimizing the fountain's operation. The primary objective is to provide real-time information about water fountain status to residents through a public platform.

**PROBLEM STATEMENT**

Conventional water fountains have manual control, energy wastage, maintenance burden, lack of data insights, and environmental impact due to their inefficient water usage and energy consumption, making them unsuitable for optimizing operations. There exists a pressing need for a technologically advanced solution leveraging the Internet of Things (IoT) to establish efficient and convenient remote control, automation, and real-time monitoring, optimizing resource usage and user experience. This system combines traditional water features with advanced technology, enabling remote control, monitoring, and data-driven insights.

**DESIGN THINKING:**

1. Hardware Setup:

- Water Level Sensor: Use a water level sensor to monitor the water level in the fountain.

- Pump Control: Connect a water pump to a relay module that can be controlled electronically.

- Microcontroller: Use a microcontroller like Arduino, Raspberry Pi, or an ESP8266/ESP32 for processing and control.

2. Programming the Microcontroller:

- Write code to read data from the water level sensor.

- Implement logic to turn the water pump on and off based on the water level.

- Ensure the microcontroller can communicate with your chosen IoT platform.

3. IoT Platform Integration:

- Choose an IoT platform like AWS IoT, Google Cloud IoT, or IoT platforms like Blynk or Thing speak.

- Set up your IoT platform account and create a new "device" to represent your fountain.

- Configure your microcontroller to send water level data and control commands to the IoT platform using MQTT, HTTP, or another protocol.

4. Remote Control and Monitoring:

- Create a mobile app or web interface that connects to your IoT platform.

- Allow users to monitor the water level and control the fountain remotely.

- Implement security measures to protect the IoT system.

5. Alerts and Notifications:

- Set up alerts and notifications in your IoT platform to notify users if the water level is too low or if there are any issues with the fountain.

6. Data Storage and Analysis:

- Configure data storage in your IoT platform to store historical water level data.

- You can use this data for analytics, trend analysis, or future optimizations.

7. Testing and Debugging:

- Thoroughly test your IoT system to ensure it works reliably.

- Debug any issues that arise during testing.

8. Deployment:

- Install your smart water fountain in its intended location.

- Ensure it has a stable internet connection.

9. Maintenance:

- Regularly monitor the system for any issues.

- Keep the software and firmware up to date.