Smart water fountains using machine learning algorithm

ALGORITHM:

1. **Sensor Data Acquisition**:

 - Gather data from various sensors, including water level sensors, temperature sensors, motion sensors, and water quality sensors.

2. **Data Preprocessing**:

 - Filter and preprocess sensor data to eliminate noise and ensure data accuracy.

 - Convert analog sensor data to digital format for analysis.

3. **Data Analysis and Decision Making**:

 - Analyze the data to make decisions, such as:

  - Checking if the water level is below a certain threshold to prevent pump damage.

  - Monitoring water quality for contaminants.

- Adjusting water temperature based on user preferences.

- Detecting motion or proximity of users to activate the fountain.

4. **User Interaction**:

- Implement user interfaces for controlling the fountain, such as mobile apps or web interfaces.

- Allow users to set preferences like water temperature or fountain activation schedules.

5. **Control Mechanisms**:

- Control the fountain's pump, heating/cooling elements, and lighting based on the data analysis and user input.

- Ensure that the water level is maintained within safe limits.

6. **Security**:

- Implement robust security measures to protect the IoT system from unauthorized access and data breaches.

7. **Communication**:

- Establish communication protocols, like MQTT or HTTP, for transmitting data between the fountain and a central server or cloud platform.

8. **Remote Monitoring**:

- Enable remote monitoring and control of the water fountain through the internet.

- Allow users to receive alerts or notifications regarding any issues with the fountain.

9. **Energy Efficiency**:

- Implement power-saving mechanisms to conserve energy, such as scheduling the fountain's operation during specific hours.

10. **Data Storage**:

- Store historical data for analysis, troubleshooting, and future improvements.

11. **Maintenance and Self-Diagnostics**:

- Implement self-diagnostic routines to detect and report faults in sensors or components.

- Provide maintenance alerts when the fountain requires cleaning, refilling, or other maintenance tasks.

12. **Machine Learning and Predictive Maintenance (Optional)**:

- Employ machine learning algorithms to predict fountain maintenance needs based on historical data.

- Implement predictive maintenance routines to reduce downtime.

13. **Scalability**:

- Design the system to be scalable, allowing for the addition of more fountains and sensors as needed.

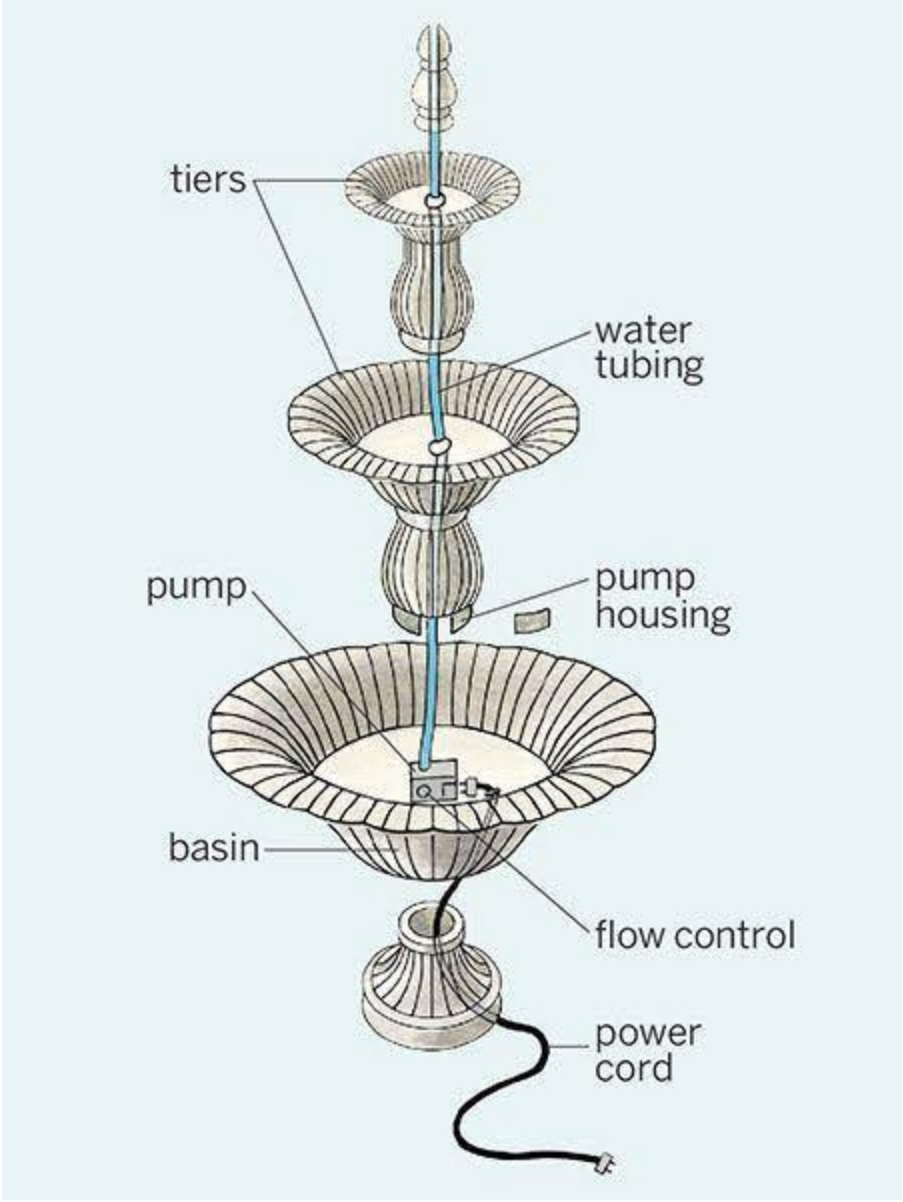
```
Step 1 → Connect Raspberry PI to Portal
Step 2 → User device should be on the same VLAN
Step 3 → Open Google Chrome browser as https://192.168.43.140:1880/ui
Step 4 → Enter Profile-based User ID/Password

SY = Water Monitoring System such that SY= { Ip, Op, Fn}
where Ip= set of input, Op= set of output, Fn= set of function

Start
Input: IoT Sensor Values Ip= {Ip1, Ip2, Ip3, Ip4}
Ip1= Water pH sensor
Ip2= Temperature sensor
Ip3= Water Flow sensor
Ip4= Turbidity sensor

Function: Fn= {Fn1, Fn2, Fn3}
Fn1= Gather IoT sensor data
Fn2= Store Input data on cloud
Fn3= Display data on website

Output: Op= {Op1, Op2}
Op1= if Ip ≥ Threshold (Th)
        Water contaminated → Water tank outlet to be closed
      else
          Op1= if Ip < Threshold (Th)
          Water NOT contaminated → Water tank outlet kept open
Op2= show data on webpage
End
```


tiers
water tubing
pump
pump housing
basin
flow control
power cord