*SMART WATER FOUNTAIN USING IOT*

PHASE 4 : DEVELOPMENT PART-2

Abstract :

Smart water fountains are innovative and technologically advanced water dispensing systems that incorporate various sensors, connectivity, and automation to improve water accessibility, conservation, and user experience. These fountains leverage IoT technology to monitor water quality, track consumption, and provide real-time data for both users and maintenance teams. Additionally, they often include features such as touchless operation, water purification, and customizable settings. Smart water fountains not only enhance the efficiency and hygiene of public water sources but also contribute to sustainability efforts by reducing water wastage and promoting responsible water usage. This abstract explores the concept and potential benefits of smart water fountains in the context of modern urban environments.

SYSTEM DESIGN AND ARCHITECHTURE

Raspberry Pi Pico:

This will be the processor of your project

Water pump :

To control the flow of water

Sensors:

you can use sensors like IR to detect when a user approaches the fountain and water level sensors to monitor the water level

Power supply:

Ensure you have an appropriate power supply for your components

Relay or Transistors:

To control the water pump based on sensor inputs.

MOBILE APP DEVELOPMENT:

You can create a mobile app for both android and ios platforms. You have several options for app development , including:

* Native Development : use the platform specific languages and development environments (swift / objectives-c for ios ,java/kotlin for android).
* Cross platform development: use frameworks like react native ,flutter, or Xamarin to develop a single codebase that works on both ios and android

CREATE USER INTERFACE:

Design the mobile app user interface to control your smart water fountain . this can include buttons or silders to start /stop the fountain , adjust water flow, and monitor water level .

MOBLIE APP -SERVER COMMUNICATIONS:

You need a way for your mobile app to communicate with your raspberry pi,which acts as the server for your iot project

* This can be achieved through a few different methods:
* HTTP REQUESTS: you can set up a restful api on your raspberry pi,and the mobile app sends http requests to control the fountain
* MQTT: implement MQTT client functionality in your mobile app to send and receive messages: from your raspberry pi
* WEB SOCKET : use web sockets for real time communication between the mobile app the raspberry pi.

MOBLIE APP DEVELOPMENT:

Write the code for your mobile app, including the logic for sending commands to the raspberry pi,receiving updates , and displaying the fountains status .ensure that the app can handle different states , such as starting ,stopping ,and error handling.

TEST YOUR APP:

Thoroghly test your mobile app, both locally and with the raspberry pi,ensure that it can connect to your raspberry pi and control the smart water fountain as intended.

DEPLOY THE MOBLIE APP:

Publish your mobile app on the respective app stores(google play store for android and apple app store for ios ) . this will make it available to users for download.

**\*Once you’ve completed these steps ,users can download the app , connect it to your raspberry pi,and control the water fountain remotely .make sure to maintain and update your app and iot system as needed to provide a seamless user experience.**

MOBILE APP CODE :

import requests

import RPi.GPIO as GPIO

# Define the Raspberry Pi's IP address

raspberry\_pi\_ip = 'your\_pi\_ip\_address'

ir\_sensor\_pin = 17

GPIO.setmode(GPIO.BCM)

GPIO.setup(ir\_sensor\_pin, GPIO.IN)

try:

ir\_sensor\_state = GPIO.input(ir\_sensor\_pin)

if ir\_sensor\_state == GPIO.HIGH:

print('IR sensor detected an object')

else:

print('IR sensor did not detect an object')

except KeyboardInterrupt:

pass

finally:

GPIO.cleanup()

WEBDEVELOPMENT TECHNOLOGIES USING JAVASCRIPT CODE:

<html>

<head>

<title>Smart Water Fountain</title>

<style>

/\* CSS for styling the fountain (same as before) \*/

/\* ... \*/

.water-flow.stopped {

animation-play-state: paused;

}

</style>

</head>

<body>

<h1>Smart Water Fountain</h1>

<div class="fountain">

<div class="water-flow"></div>

</div>

<button id="startButton">Start Fountain</button>

<button id="stopButton">Stop Fountain</button>

<input type="range" id="flowRate" min="1" max="10" value="5">

<p>Flow Rate: <span id="flowRateValue">5</span></p>

<script>

const waterFlow = document.querySelector(".water-flow");

const startButton = document.getElementById("startButton");

const stopButton = document.getElementById("stopButton");

const flowRateSlider = document.getElementById("flowRate");

const flowRateValue = document.getElementById("flowRateValue");

// Simulate an IR sensor

let irSensorState = false;

function updateIRSensorState() {

// You can replace this with your actual IR sensor reading logic

// For demonstration, we're alternating between true and false.

irSensorState = !irSensorState;

if (irSensorState) {

startButton.disabled = false;

stopButton.disabled = false;

} else {

startButton.disabled = true;

stopButton.disabled = true;

}

}

// Call the function to update the IR sensor state periodically

setInterval(updateIRSensorState, 1000); // Adjust the interval as needed

let currentFlowRate = 5; // Default flow rate

startButton.addEventListener("click", () => {

if (irSensorState) {

waterFlow.style.animationPlayState = "running";

}

});

stopButton.addEventListener("click", () => {

waterFlow.style.animationPlayState = "paused";

});

flowRateSlider.addEventListener("input", () => {

currentFlowRate = flowRateSlider.value;

flowRateValue.textContent = currentFlowRate;

const animationDuration = 10 / currentFlowRate; // Inverse of flow rate

waterFlow.style.animationDuration = `${animationDuration}s`;

});

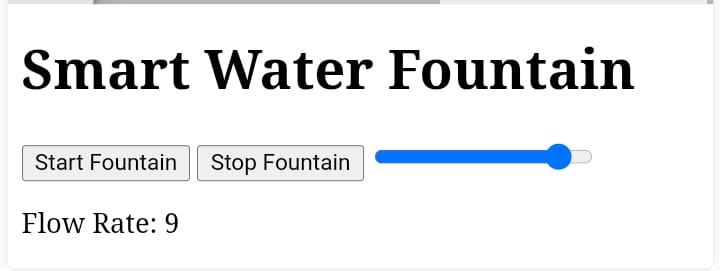
</script>

</body>

</html>

OUTPUT :

The coding is in the language of java script



The code is executed successfully .development of the mobile app and web development technologies is successfully implemented.