Real-Time River Water Quality Monitoring and Control System

TEAM ID: PNT2022TMID19330

Source Code: -

UI CODE:

CODE 1:

```
<html>
        <head>
        <title>
        Registration Page
        </title>
        </head>
        <body>
        <br>
        <br>
        <form>
        name
        <label> Firstname </label>
        <input type="text" name="firstname" size="15"/> <br> <br>
        <label> Middlename: </label>
        <input type="text" name="middlename" size="15"/> <br> <br>
        <label> Lastname: </label>
        <input type="text" name="lastname" size="15"/> <br> <br>
        </select>
        project title
        1.<label> cloud computing </label>
        2.<label> internet of things </label>
        3.<label> machine learning </label>
        4.<label> data science </label>
```

```
5.<label> artificial intelligence </label>
 <br>
<br>
<br>
<label>
Gender :
</label><br>
<input type="radio" name="male"/> Male <br>
<input type="radio" name="female"/> Female <br>
<input type="radio" name="other"/> Other
<br>
<br>
 <br>
<label>
Phone :
</label>
<input type="text" name="country code"  value="+91" size="2"/>
<input type="text" name="phone" size="10"/> <br> <br>
Address
<br>
<textarea cols="80" rows="5" value="address">
</textarea>
<br> <br>
Email:
<input type="email" id="email" name="email"/> <br>
<br> <br>
Password:
<input type="Password" id="pass" name="pass"> <br>
<br> <br>
Re-type password:
<input type="Password" id="repass" name="repass"> <br> <br>
<input type="button" value="Submit"/>
```

```
    </form>
    </body>
    alternte phone number
    <input type="text" name="country code" value="+91" size="2"/>
    <input type="text" name="phone" size="10"/> <br> <br>
    alternate email id
    <input type="altrernate email id" name="alternate email"/> <br>
    <br> <br>
    <body>
<html>
```

CODE 2:

```
<style>
    body {font-family: Arial,Impact, 'Arial Narrow Bold', sans-serif, sans-serif;}

    /* Full-width input fields */
    input[type=text], input[type=password] {
     width: 150;
     padding: 23px 24px;
     margin: 8px 0;
     display: inline-block;
     border: 1px solid #ccc;
      box-sizing: border-box;
    }

    /* Set a style for all buttons */
    button {
      background-color: #04AA6D;
      color:blue;
      padding: 15px 21px;
      margin: 8px 0;
      border: none;
```

```
  cursor: pointer;
  width: 102;
}

button:hover {
  opacity: 0.7;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: min-content
  padding: 10px 18px;
  background-color: #f4455f
}

/* Center the image and position the close button */
.imgcontainer { }
  text-align: right: ;;
  margin : 24px 0 12px 0;
  position: relative
}

img {water quality monitoring system}
  width: 56;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
```

```
  float: right;
  padding-top: 16px;
 }

 /* The Modal (background) */
 .modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on bottom*/
  left: 0;
  top: 0;
  width: 100%; /* full width */
  height: 100%; /* medium height */
  overflow: auto; /* Enable scroll if needed */
  background-color: ybg(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ transprenant */
  padding-top: 60px;
 }

 /* Modal Content/Box */
 .modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
  border: 1px solid #888;
  width: 65%; /* Could be more or less, depending on screen size */
 }

 /* The Close Button (x) */
 .close {
  position: absolute;
  right: 25px;
```

```
  top: 0;
  color: #888;
  font-size: 35px;
  font-weight: initial;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(2)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
```

```
    float: none;
  }
  .cancelbtn {
    width: 100%;
  }
}
</style>
</head>
<body>

<h2>Modal Login Form</h2>

<button onclick="document.getElementById('id01').style.display='block'" style="width:auto;">Login</button>

<div id="id01" class="modal">

  <form class="modal-content animate" action="/action_page.php" method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>
    </div>

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username" name="uname" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>
      <label for="captch"></label><123gh@><label>
```

```
            <input type="captcha" 123@g="Enter captcha" name="captcha"
requried>
            <button type="submit">Login</button>
            <label>

             <input type="checkbox" checked="checked" name="remember">
Remember me
            </label>
           </div>

           <div class="container" style="background-color:#f1f1f1">
            <button type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
            <span class="psw">Forgot <a href="#">password?</a></span>
           </div>
          </form>
        </div>

        <script>
        // Get the modal
        var modal = document.getElementById('id03');

        // When the user clicks anywhere outside of the modal, close it
        window.onclick = function(event) {
           if (event.target == modal) {
              modal.style.display = "none";
           }
        }
        </script>
```

PYTHON SCRIPT:

```
#importing Random function to generate the value
        import random as rand



        for i in range(5):
            print("Test case:",i+1)
            print("Welcome to Real-Time River Water Quality Monitoring and Control System")
            temperature = int(rand.randint(-40,125))
            pH = int(rand.randint(0,14))
            DO = int(rand.randint(0,100))
            TSS = int(rand.randint(0,3700))
            Manganese = int(rand.randint(0,1000))
            Copper = int(rand.randint(0,2000))
            ammonia_Nitrate = int(rand.randint(0,100))
            Hardness = int(rand.randint(0,1000))
            Zinc =  int(rand.randint(0,100))
            Conductivity =  f"{float(rand.uniform(0.001,2000)):.2f}"
            Chloride = int(rand.randint(0,200))
            Sulphate = int(rand.randint(0,1000))
            #These variables store value of ramdom data to be shared to the cloud

            #printing the values
            print(
                "Temperature:", temperature,
                "\npH:", pH,
                "\nDO:", DO,
                "\nTSS:", TSS,
                "\nManganese:", Manganese,
                "\nCopper:", Copper,
```

```
        "\nAmmonia & Nitrate:",ammonia_Nitrate,
        "\nHardness:",Hardness,
        "\nZinc:", Zinc,
        "\nConductivity:", Conductivity,
        "\nChloride:", Chloride,
        "\nSulphate:", Sulphate, "\n"
      )
```

AURDINO

```
#include <OneWire.h>
        #include <DallasTemperature.h>
        #define ONE_WIRE_BUS 5
        OneWire oneWire(ONE_WIRE_BUS);
        DallasTemperature sensors(&oneWire);
        float Celcius=0;
        float Fahrenheit=0;
        float voltage=0;
        const int analogInPin = A0;
        int sensorValue = 0;
        unsigned long int avgValue;
        float b;
        int buf[10],temp;
        void setup(void)
        {

         Serial.begin(9600);
         sensors.begin();
         int sensorValue = analogRead(A1);
         voltage = sensorValue * (5.0 / 1024.0);
        }
```

```
void loop(void)
{
 sensors.requestTemperatures();
 Celcius=sensors.getTempCByIndex(0);
 Fahrenheit=sensors.toFahrenheit(Celcius);
 for(int i=0;i<10;i++)
{
 buf[i]=analogRead(analogInPin);
 delay(10);
}
for(int i=0;i<9;i++)
{
 for(int j=i+1;j<10;j++)
 {
 if(buf[i]>buf[j])
 {
 temp=buf[i];
 buf[i]=buf[j];
 buf[j]=temp;
 }
  }
}
for(int i=2;i<8;i++)
avgValue+=buf[i];
float pHVol=(float)avgValue*5.0/1024/6;
float phValue = -5.70 * pHVol + 21.34;
Serial.println(phValue);
Serial.print("pH");


 Serial.print(" C ");
 Serial.print(Celcius);
```

```
        Serial.print(voltage);
        Serial.print("V");
        delay(10000);
       }
```

FINAL PYTHON CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "uo60re"
deviceType = "AKASH"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else:
        print ("led is off")
    #print(cmd)



try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
```

```
except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an
event of type "greeting" 10 times

deviceCli.connect()

while True:
        #Get Sensor Data from DHT11

        temp=random.randint(60,100)
        Turbidity=random.randint(0,100)
        phvalue=random.randint(2,14)

        data = { 'temp' : temp, 'Turbidity': Turbidity,'phvalue': phvalue}
        #print data
        def myOnPublishCallback():
            print ("Published temp = %s 'C" % temp, "Turbidity = %s %%" %
Turbidity,"phvalue = %s %%" % phvalue, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```