**NADAR SARASWATHI COLLEGE OF ENGINEERING AND TECHNOLOGY**

**VADAPUDUPATTI, THENI**

(Approved by AICTE, New Delhi and Affiliated to Anna University Chennai)

****

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**SUBMITTED BY**

**GUIDED BY**

DR.VENISH KUMAR M.INDHUMATHI-921019106011

S.KARTHIKA-921019106018

K.NITHYATHARSHINI-921019106501

M.SHARMILA-921019106034

**TOPIC**

REAL TIME RIVER WATER MONITORING AND CONTROL SYSTEM

**FOR**

PROFESIONAL READLINES FOR INNOVATION EMPLOABILITY AND ENTREPRENEURSHIP (Hx8001)

**INTRODUCTION**

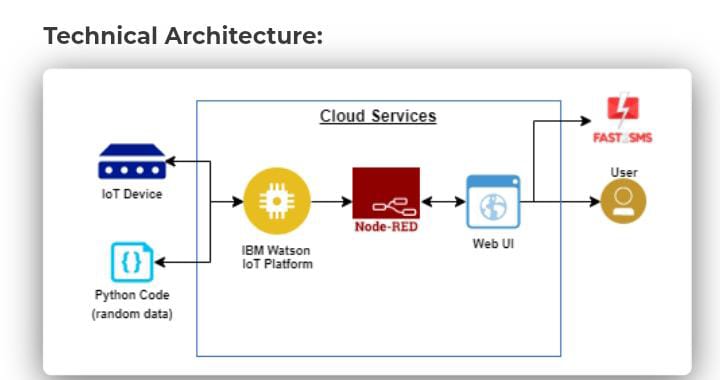
**1.1Project Overview**

The main aim of this project is to detect the quality of river water and quantity of pollutants present in water and so that river water quality is monitored and effective measures can be taken accordingly . To develop a system for real time quality assessment for river water health at residential places using Raspberry Pi. Sensors are used to gather different parameters in River water to monitor water health in real time. IoT is appeared to be a better solution as reliability, scalability, speed, and persistence can be provided

**1.2Purpose**

This instrucatable is used to document the development of an automated river water monitoring system. The monitored parameters are the water level and water temperature. The goal of this project was to develop a low-cost and independent logger that is connected to a database. The water level is monitored with a special QR-level. The QR-codes in this picture get decoded by software and are indicative to the actual water level, Water temperature ,humidity,ph value and turbidity.

**TECHNICAL ARCHITECTURE**

****

**PROJECT FLOW**

* Sending random pH values and turbidity values will be sent to the IBM IoT platform
* Sensors values can be viewed in the Web Application
* Notifies the admin the random values cross the threshold value

To accomplish this, we have to complete all the activities and tasks listed below:

* Create and configure IBM Cloud Services
  + Create IBM Watson IoT Platform
  + Create a device & configure the IBM IoT Platform
  + Create Node-RED service
  + Create a database in Cloudant DB to store location data
* Develop a web Application using Node-RED Service.
  + Develop the web application using Node-RED
* Develop a python script to publish the location details to the IBM IoT platform

SOFTWARE AND PACKAGE

Following software,concept and packages are used in this project.

* Node red
* MIT app inventer

SOFTWARE INSTALLATION

Install the required python libraries:

* Install Watson IoT Python SDK to connect to IBM Watson IoT Platform using python code:

give the following command in command prompt: pip install wiotp-sdk

PROJECT CONCEPT

import time  
import sys  
import ibmiotf.application  
import ibmiotf.device  
import random  
  
  
#Provide your IBM Watson Device Credentials  
organization = "16s88b"  
deviceType = "weather\_device"  
deviceId = "weather\_today"  
authMethod = "token"  
authToken = "Ue7RQMEaYP-h4\_uCM6"  
  
# 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(0,100)  
        Humid=random.randint(0,100)  
          
        data = { 'temp' : temp, 'Humid': Humid }  
        #print data  
        def myOnPublishCallback():  
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM Watson")  
  
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)  
        if not success:  
            print("Not connected to IoTF")  
        time.sleep(1)  
          
        deviceCli.commandCallback = myCommandCallback  
  
# Disconnect the device and application from the cloud  
deviceCli.disconnect()

ADVANTAGES

**Ease and convenience of usageInstantaneous data**

**Improved accuracy of measurement**

**Contact Guardian Water Treatment today**

**DISADVANTAGES**

**Water Cleansing May Not Eliminate PesticidesWater**

**Cleansing Requires Standard Support**