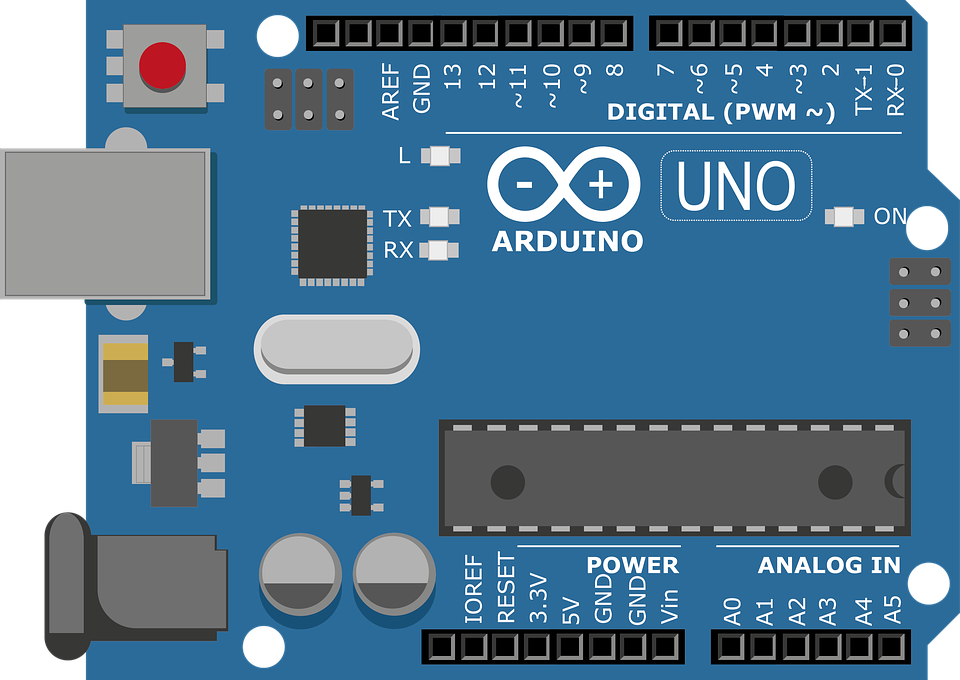
**INTERNET OF THINGS**

**PHASE-5**

**SMART WATER MANAGEMENT**



PET ENGINEERING COLLEGE

# PROJECT OBJECTIVITIES:

# Real-time Water Consumption Monitoring:

Real-time water consumption monitoring refers to the continuous and immediate tracking of water usage in a given area or system. It involves the collection, analysis, and reporting of water consumption data in real-time or near-real-time, allowing for better insights into current usage patterns and the ability to respond promptly to changes or anomalies.

# Public Awareness (in the context of water conservation):

Public awareness in the context of water conservation refers to efforts and campaigns aimed at educating and informing the general public about the importance of water conservation practices. This objective involves raising awareness about the finite nature of water resources, the need to reduce wasteful consumption, and the adoption of water-saving behaviors in homes, businesses, and communities.

# Water Conservation:

Water conservation is the practice of using water resources efficiently and reducing unnecessary water waste. It involves the implementation of strategies and technologies to minimize water consumption while maintaining essential needs and services. The goal of water conservation is to ensure the sustainable use of water resources, protect ecosystems, and meet the needs of present and future generations.

# Sustainable Resource Management (in the context of water):

Sustainable resource management refers to the responsible and balanced utilization of water resources to meet current needs without compromising the ability of future generations to meet their own needs. This objective involves managing water sources, ecosystems, and infrastructure in a way that ensures long-term sustainability, considers environmental impacts, and factors in social and economic considerations.

## Define Objectives and Requirements

### Objectives:

Real- time monitoring of water consumption.

Identifying and addressing water waste.

Pr0moting water conservation.

insuring the sustainable use of water resources.

### Requirements:

Sensor types( flow meters, pressure sensors, temperature sensors, etc.).

Data transmission and communication protocols( Wi- Fi, cellular, LoRa, etc.).

Power source( battery, solar, or wired).

Data storage and analysis platform.

User interface for data visualization.

Alarm system for abnormal water usage.

## Step -2: Sensor Selection

### Sensor Types:

Choose appropriate sensors for measuring water flow, pressure, and temperature.

Consider the reliability, accuracy, and durability of sensors.

### Communication:

Select communication protocols and technologies suitable for the deployment environment( e. g., Wi- Fi for indoor, LoRa for long- range outdoor).

### Power Source:

Determine the power source based on deployment location( battery- powered for remote areas, solar- powered for outdoor locations).

## Step -3: Sensor Deployment

### Location Identification:

Identify strategic locations for sensor placement, such as water pipes, faucets, toilets, and irrigation systems.

Ensure accessibility for maintenance.

### Installation:

Install sensors securely and according to manufacturer specifications.

Connect sensors to the chosen power source and communication network.

### Calibration:

Calibrate sensors to ensure accurate measurements.

Regularly check and recalibrate sensors to maintain accuracy.

## Step -4: Data Collection and Transmission

### Data Collection:

Collect data from the sensors in real- time or at regular intervals.

Ensure data integrity and quality control.

### Data Transmission:

Transmit data securely to a central data storage and analysis platform.

Use encryption and authentication to protect data during transmission.

## Step -5: Data Storage and Analysis

### Data Storage:

Store collected data in a secure and scalable database.

Implement data retention policies.

### Data Analysis:

Analyze water consumption patterns to detect anomalies and identify potential water waste.

Use machine learning algorithms for predictive maintenance and optimization.

## Step -6: Visualization and Reporting

### User Interface:

Develop a user- friendly interface for data visualization and reporting.

Provide access to real- time and historical consumption data.

### Alerts and Notifications:

Set up alert systems to notify stakeholders of abnormal water usage or system issues.

## Step 7: Maintenance and Optimization

### Regular Maintenance:

Establish a maintenance schedule for sensor inspection, cleaning, and replacement.

Address any issues promptly to ensure continuous operation.

### Optimization:

Use the insights from data analysis to optimize water usage and conservation efforts.

Adjust sensor placement or configuration as needed.

## Step 8: Public Awareness

### Outreach Campaigns:

Use the collected data to engage the public in water conservation efforts.

Implement educational campaigns and incentives to promote water- saving behaviors.

## Step 9: Compliance and Regulation

### Compliance:

Ensure that the IoT sensor deployment complies with local regulations and data privacy laws.

### Reporting:

Prepare reports and data for regulatory agencies as required.

## Step 10: Evaluation and Continuous Improvement

### Evaluation:

Regularly assess the effectiveness of the IoT sensor system in meeting objectives.

# **PRINCIPLE:**

A smart water-based system based on Arduino collects data from sensors, processes it using Arduino microcontrollers, and then uses this information to automate and optimize water-related functions such as irrigation, water level control, and leak detection. Through real-time monitoring and intelligent decision-making, it encourages efficient water usage and conservation.

# **DESIGN PRINCIPLE:**

The design concepts for an Arduino-based smart water-based system should focus on building a dependable, efficient, and user-friendly solution. Consider the following fundamental design principles:

1. **Scalability:** Make the system scalable so that it can be easily expanded or adapted to handle different sizes and types of water management jobs.
2. Modular components and standardized interfaces are used to facilitate maintenance, upgrades, and repairs.
3. **Energy Efficiency:** Optimize power consumption to ensure the system performs efficiently, particularly if it is powered by a battery or solar panels.
4. **Data Accuracy:** Select high-quality sensors and assure calibration in order to obtain accurate data for informed decision-making.
5. **Redundancy:** To improve system reliability, include redundancy wherever possible, such as backup power sources or multiple sensors.
6. Create an easy user interface, whether through a mobile app or a web portal, to allow users to quickly monitor and control the system.
7. Implement security measures to safeguard data and prevent unwanted access.

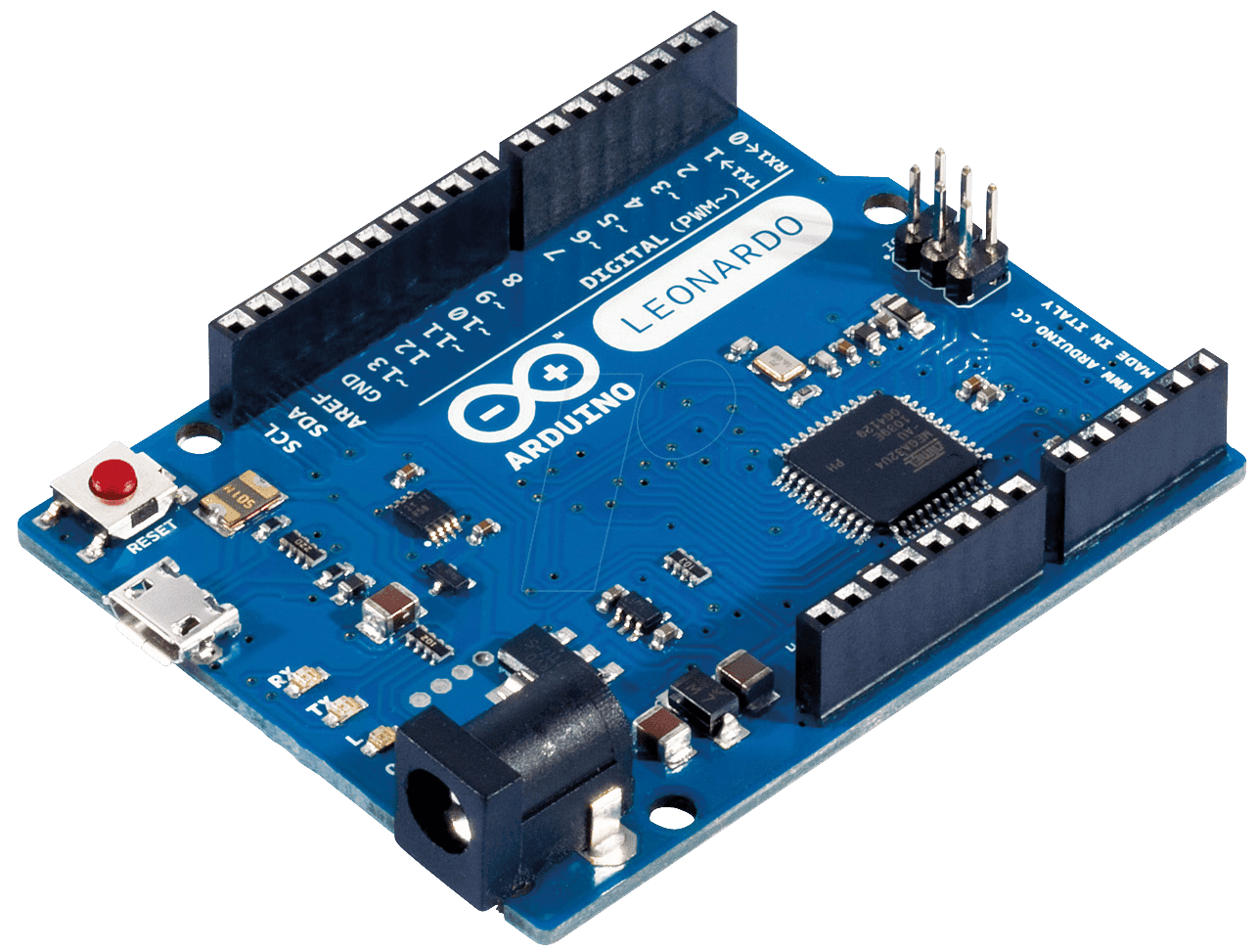
# **BLOCK DIAGRAM**:

The developed system is utilized for water monitoring and quality control. The sensor in the base tank first detects the presence of water. When there is water, the pump starts automatically and begins pumping water to the overhead tank. Different water levels in the over head tank are monitored. When the water reaches the chosen threshold, the user is notified. When the water level reaches the maximum level, the pump shuts down automatically. If water flows continuously for longer than predicted, the water flow sensor will notice it and send the data to the IOT server.



The user can view real-time data in the IOT server, and in this case, one notification will be transmitted from the IOT server to the mobile app. If the PH level of the water or the dirt level is not acceptable, the water will not be pumped into the tank. The user can manage the flow of water by engaging with the server via the mobile app.

# **Components Required:**

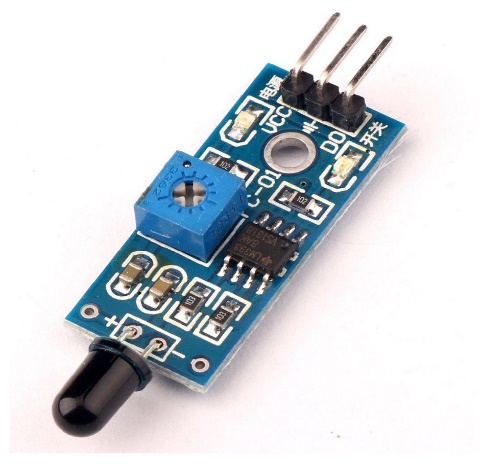


[This Photo](http://www.electronicaivanespinoza.com/2019/05/que-es-arduino-conceptos-basicos-y.html) by Unknown Author is licensed under [CC BY-SA-NC](https://creativecommons.org/licenses/by-nc-sa/3.0/)

1.Arduino Microcontroller:

As the system's central processing unit, select an Arduino board (e.g., Arduino Uno, Arduino Nano, or Arduino Mega).

## 2. Sensors:

 Soil Moisture Sensor: Measures soil moisture content for efficient irrigation.

Water Level Sensor: Monitors water levels in tanks, reservoirs, or bodies of water.

Flow Meter: Measures the flow rate of water in pipes or irrigation systems.

Temperature and Humidity Sensor: Provides data for weather and environmental conditions.

## 3. Actuators:

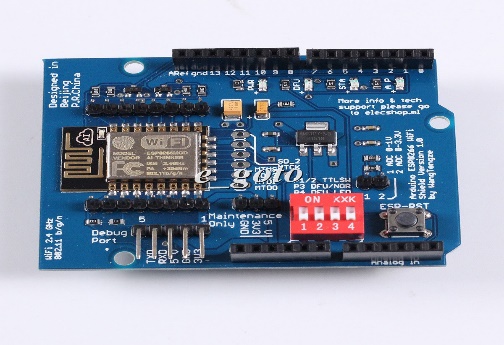
Water Pumps: Control water flow for irrigation or distribution.

Solenoid Valves: Regulate the flow of water in pipes or irrigation systems.

Relays: Switch high-voltage devices, like pumps or valves, on and off.

## 4. Communication Module:

Wi-Fi, GSM, LoRa, or Bluetooth module to enable data transmission and remote control.



[This Photo](http://arduino.stackexchange.com/questions/24919/how-to-connect-wi-fi-shield-esp-12e-esp8266-uart-wifi-wireless-shield-with-ardui) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

## 5.Power Supply:

Depending on your application, you may need a power source, such as a battery, solar panels, or an AC power supply.

# **PLATFORM DEVELOPMENT:**

## WHAT DOES THE PLATFORM DO?

1. Monitoring and Data Collection:

A network of sensors and data gathering tools is used by smart water systems to keep an eye on a variety of water resource-related metrics. This covers measures of temperature, flow rates, water level, water quality, and more. These sensors can be installed in a variety of places, including residences, businesses, treatment plants, distribution networks, and reservoirs.

1. Data Analysis and Management:

To handle the enormous volume of data gathered, the platform makes use of advanced data analytics techniques. The examination of this data offers insightful information about the state of the water and usage trends. The availability and demand for water may be forecasted and optimized using machine learning techniques.

3. Real-time Alerts and Notifications:

In the event of leaks, abnormalities in the water quality, or any other concern, the platform may notify and warn water utility operators and customers, guaranteeing prompt action and mitigation.

1. User Interface and Accessibility:

Decision-makers, water utility staff, and customers can all readily access and understand data thanks to user-friendly interfaces and mobile applications.

1. Demand Forecasting:

Smart water systems can forecast patterns in water demand by examining historical data and current information. This makes it possible for utilities to allocate resources more effectively and to guarantee a steady supply of water.

## **TECHNOLOGIES USED:**

The following technologies used in this project are web based platforms of

1. HTML, CSS
2. MIT APP INVENTOR

## **WEB PLATFORM:**

Because HTML and CSS are primarily used for structuring and styling web pages, respectively, it is not possible to create a complete smart water monitoring system using just these two languages. Additional technologies, such as JavaScript for functionality and data communication, as well as a backend server to handle data from sensors, are required to create a smart water monitoring system. However, I can provide you with a simple HTML and CSS template for a water monitoring system's user interface. Here's a demo of the following project with source code

### **SOURCE CODE:**

<!DOCTYPE *html*>

<html>

  <head>

    <title>Smart Water Monitoring System</title>

    <img *src*="../../images/images.jpeg" *alternate*=" picture" />

    <link *rel*="stylesheet" *type*="text/css" *href*="../IoT Project/style.css" />

  </head>

  <body>

    <header>

      <h1>Water Monitoring Dashboard</h1>

    </header>

    <section *class*="sensor">

      <h2>Water Level</h2>

      <div *class*="sensor-value">75%</div>

    </section>

    <section *class*="sensor">

      <h2>Water Purity</h2>

      <div *class*="sensor-value">92%</div>

    </section>

    <section *class*="sensor">

      <h2>Water Temperature</h2>

      <div *class*="sensor-value">28°C</div>

    </section>

  </body>

</html>

CSS CODE:

body {

  font-family: Arial, sans-serif;

  margin: *0*;

  padding: *0*;

  background-color: #f0f0f0;

}

header {

  background-color: #333;

  color: #fff;

  text-align: center;

  padding: *20px*;

}

.sensor {

  background-color: #fff;

  border: *1px* solid #ccc;

  margin: *20px*;

  padding: *20px*;

  border-radius: *5px*;

  box-shadow: *0* *0* *5px* #ccc;

}

.sensor h2 {

  font-size: *24px*;

  margin: *0*;

}

.sensor-value {

  font-size: *36px*;

  margin: *10px* *0*;

}

## **SYSTEM WORKFLOW:**

The overall workflow of the system in the public places is given as



# **PYTHON SCRIPT IN AURDUINO:**

A python script for the sensor and acutators to perform operation:

import RPi.GPIO as GPIO

import time

# Define GPIO pin numbers

trigPin = 18

echoPin = 24

# Set GPIO mode to BCM

GPIO.setmode(GPIO.BCM)

# Set GPIO pin directions

GPIO.setup(trigPin, GPIO.OUT)

GPIO.setup(echoPin, GPIO.IN)

def measure\_distance():

# Send a 10us pulse to trigger the ultrasonic sensor

GPIO.output(trigPin, GPIO.HIGH)

time.sleep(0.00001)

GPIO.output(trigPin, GPIO.LOW)

# Measure the time it takes for the echo to return

pulse\_start = time.time()

while GPIO.input(echoPin) == 0:

pulse\_start = time.time()

while GPIO.input(echoPin) == 1:

pulse\_end = time.time()

# Calculate the distance based on the echo time

pulse\_duration = pulse\_end - pulse\_start

distance = pulse\_duration \* 17150

distance = round(distance, 2)

return distance

# Define water level thresholds

empty\_level = 10

low\_level = 20

full\_level = 30

# Continuously monitor the water level

while True:

distance = measure\_distance()

water\_level = 30 - distance

# Check water level status

if water\_level <= empty\_level:

print("Water tank is empty.")

elif water\_level <= low\_level:

print("Water level is low.")

elif water\_level <= full\_level:

print("Water level is sufficient.")

else:

print("Water tank is full.")

# Add water if level is low

if water\_level <= low\_level:

print("Adding water...")

# Implement water pump activation here

time.sleep(5) # Simulate water filling for 5 seconds

time.sleep(1) # Delay between measurements

# Clean up GPIO resources

GPIO.cleanup()

Additional for adding wifi module through HTTPS protocol

import serial

import time

import requests

# Define the serial port for communication with Arduino

serial\_port = 'COM3' # Change this to the correct port

# Define the IP address and port of the ESP8266 module

esp8266\_ip = '192.168.1.100' # Change this to the IP address of your ESP8266

esp8266\_port = 80 # Change this to the port your ESP8266 is listening on

# Create a serial connection to the Arduino

arduino = serial.Serial(serial\_port, 9600, timeout=1)

def read\_soil\_moisture():

# Send a request to the ESP8266 to get soil moisture reading

url = f'http://{esp8266\_ip}:{esp8266\_port}/moisture'

response = requests.get(url)

moisture\_reading = response.text

return int(moisture\_reading)

def control\_valve(valve\_state):

# Send a command to the ESP8266 to control the solenoid valve

url = f'http://{esp8266\_ip}:{esp8266\_port}/{valve\_state}'

response = requests.get(url)

try:

while True:

# Read soil moisture

moisture = read\_soil\_moisture()

print(f"Soil Moisture: {moisture}")

# Control the solenoid valve based on moisture level

if moisture < 300: # Adjust the threshold as needed

control\_valve('open')

else:

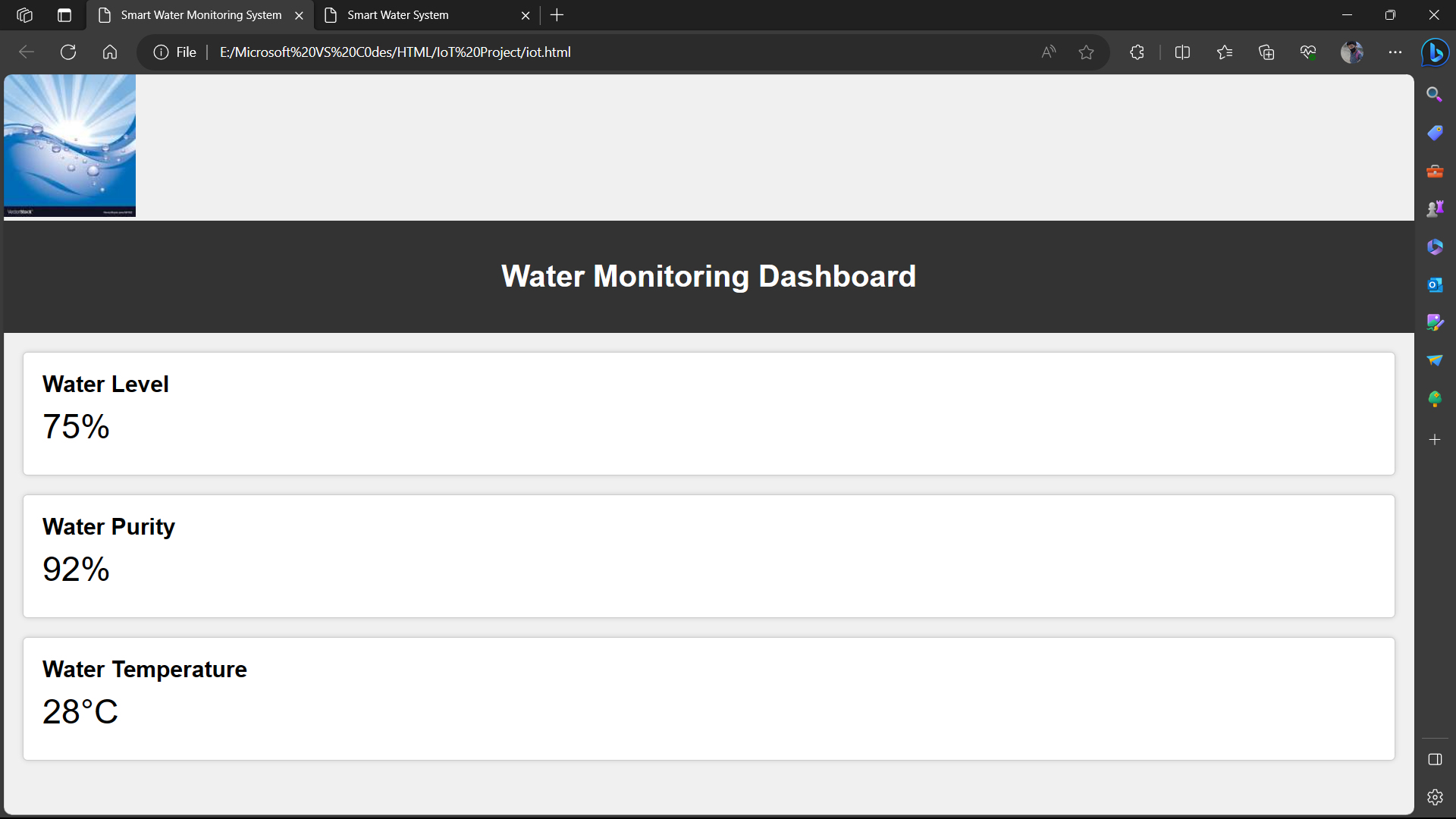
control\_valve('close')

time.sleep(3600) # Check moisture level every hour

except KeyboardInterrupt:

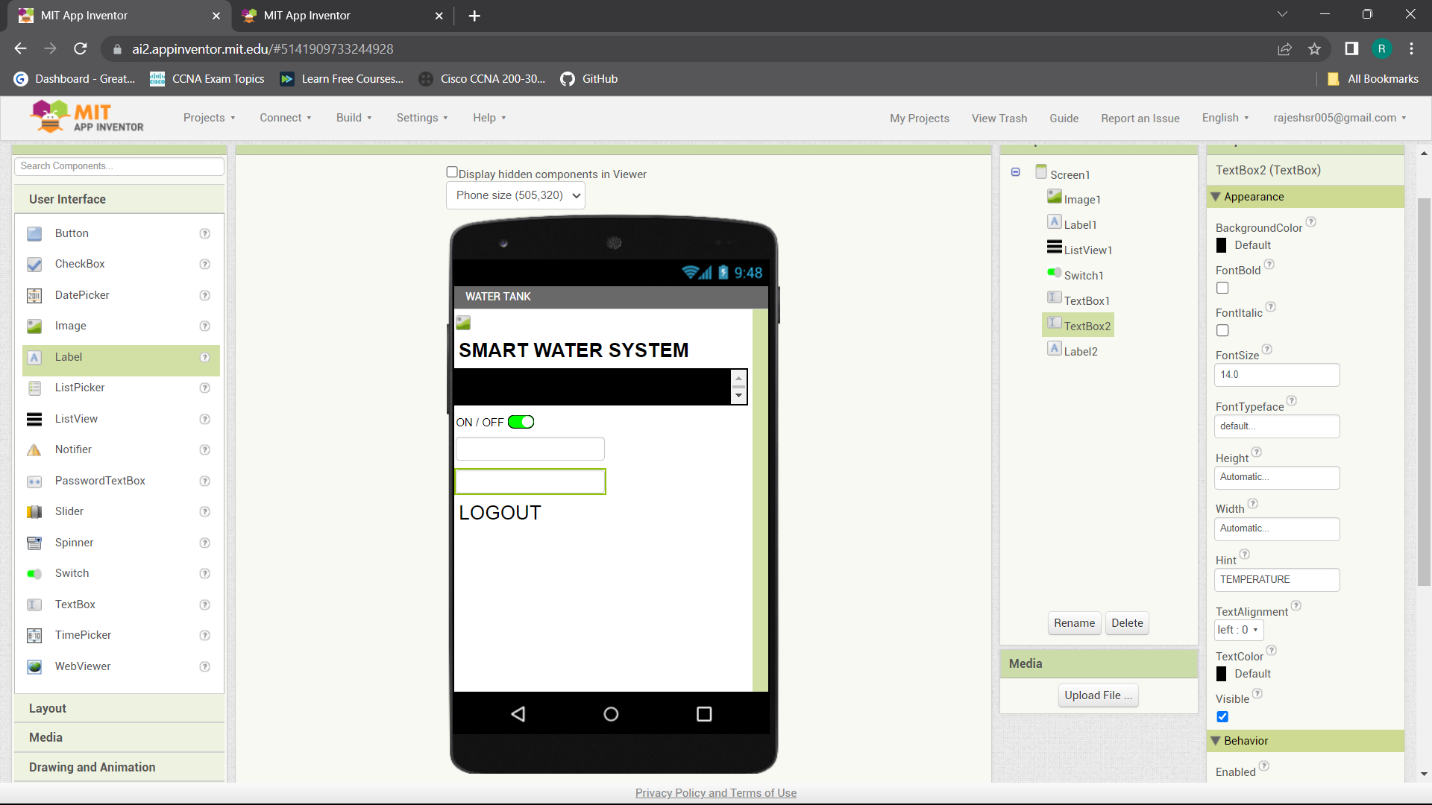
arduino.close()

## OUTPUT:



## MIT APP INVENTOR:

MIT App Inventor is primarily intended for developing mobile applications for Android devices, and it may not be appropriate for creating a full water monitoring system. However, you can create a simple mobile app that interacts with external sensors or data sources to display water monitoring information.



### **WATER LEVEL SENSOR MEASUREMENT:**

## **CONCLUSION:**

Smart water systems are an important step toward more efficient and sustainable water resource management. They play an important role in addressing water-related challenges such as water scarcity, infrastructure maintenance, and environmental protection, all while improving people's and communities' quality of life. As technology advances, these systems' capabilities will only become more sophisticated, making them an essential part of modern water management.