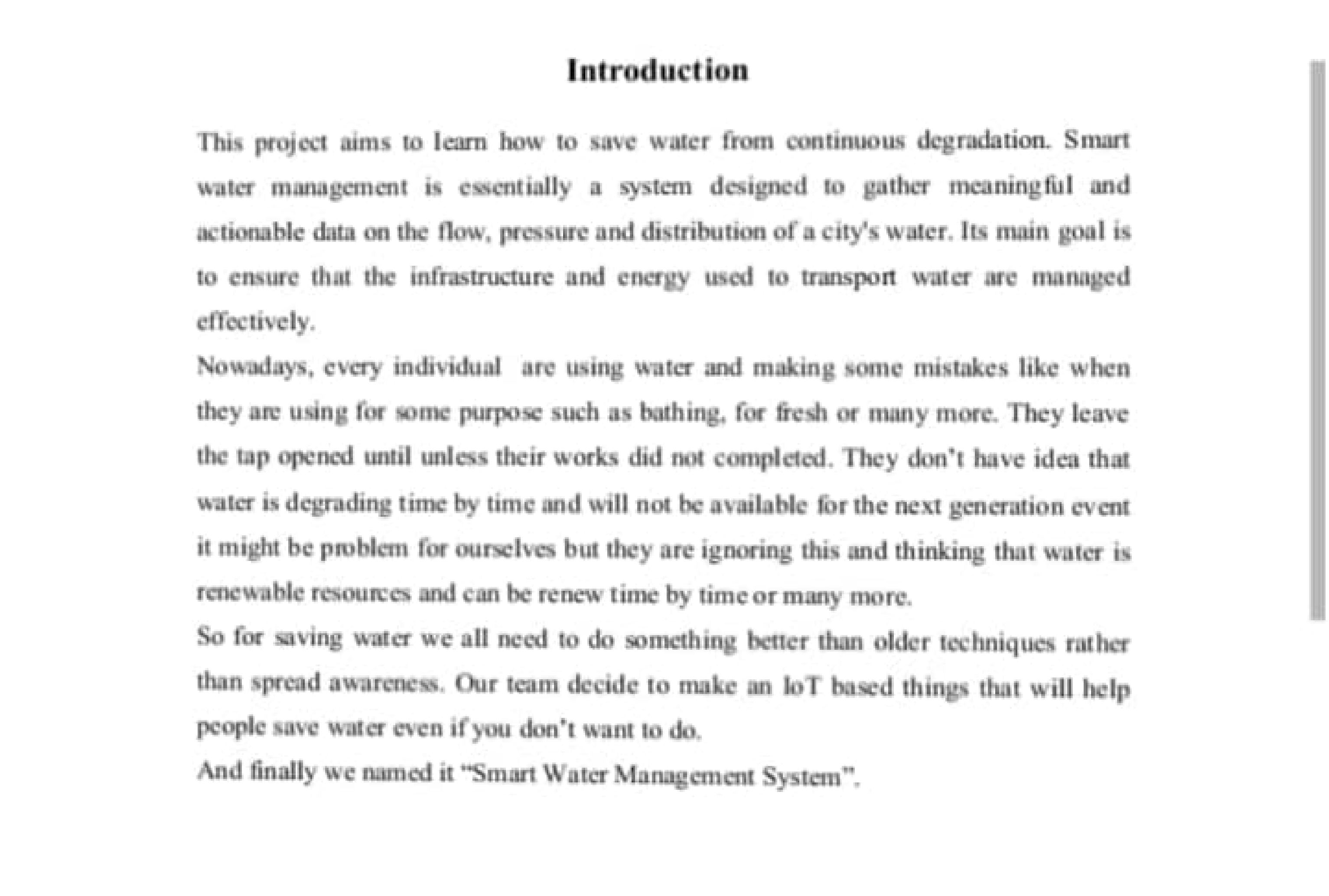


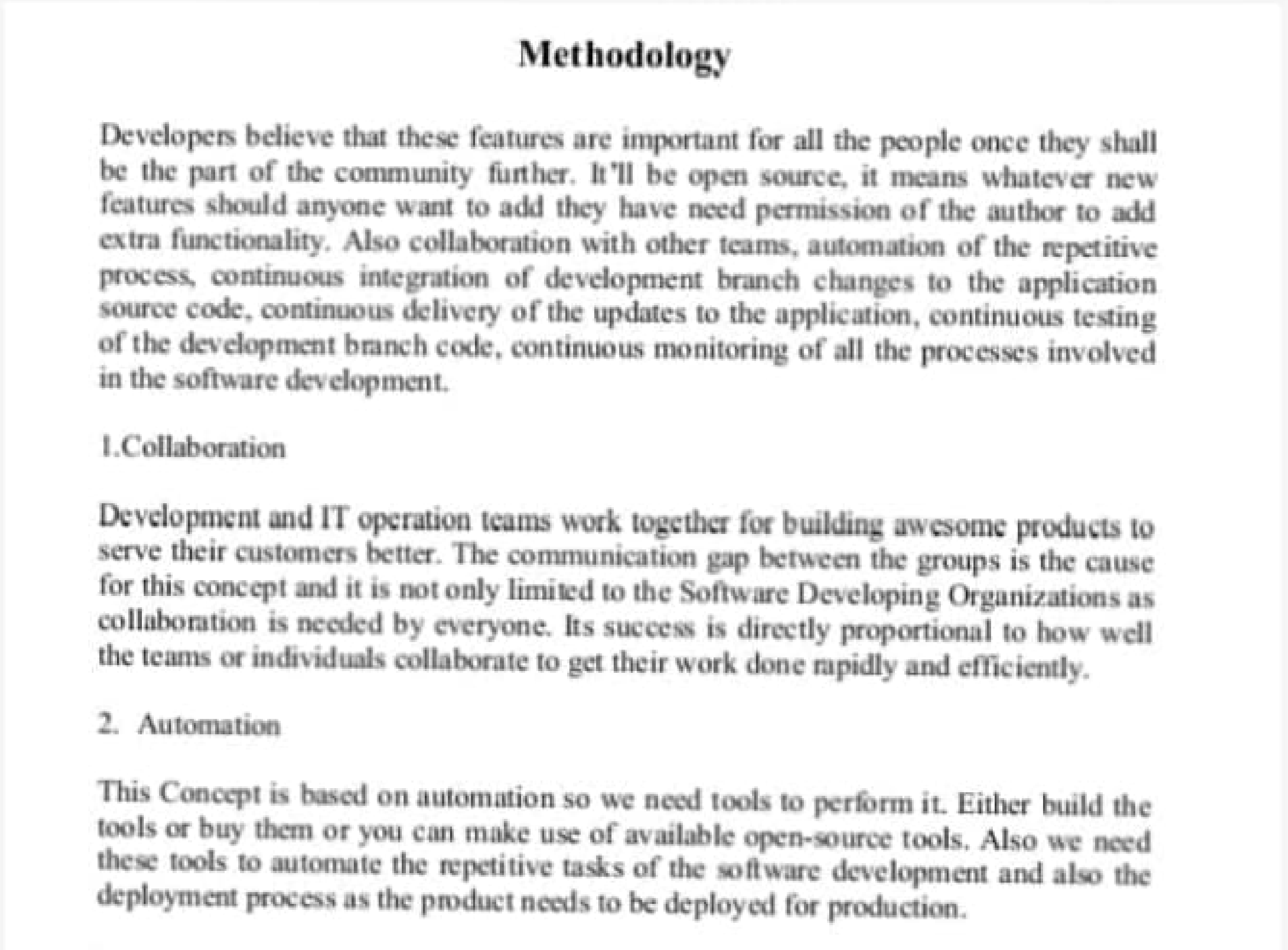
Introduction

This project aims to learn how to save water from continuous degradation. Smart water management is essentially a system designed to gather meaningful and actionable data on the flow, pressure and distribution of a city's water. Its main goal is to ensure that the infrastructure and energy used to transport water are managed effectively.

Nowadays, every individual are using water and making some mistakes like when they are using for some purpose such as bathing, for fresh or many more. They leave the tap opened until unless their works did not completed. They don't have idea that water is degrading time by time and will not be available for the next generation event it might be problem for ourselves but they are ignoring this and thinking that water is renewable resources and can be renew time by time or many more.

So for saving water we all need to do something better than older techniques rather than spread awareness. Our team decide to make an IoT based things that will help people save water even if you don't want to do.

And finally we named it "Smart Water Management System".



Methodology

Developers believe that these features are important for all the people once they shall be the part of the community further. It'll be open source, it means whatever new features should anyone want to add they have need permission of the author to add extra functionality. Also collaboration with other teams, automation of the repetitive process, continuous integration of development branch changes to the application source code, continuous delivery of the updates to the application, continuous testing of the development branch code, continuous monitoring of all the processes involved in the software development.

1.Collaboration

Development and IT operation teams work together for building awesome products to serve their customers better. The communication gap between the groups is the cause for this concept and it is not only limited to the Software Developing Organizations as collaboration is needed by everyone. Its success is directly proportional to how well the teams or individuals collaborate to get their work done rapidly and efficiently.

2. Automation

This Concept is based on automation so we need tools to perform it. Either build the tools or buy them or you can make use of available open-source tools. Also we need these tools to automate the repetitive tasks of the software development and also the deployment process as the product needs to be deployed for production.

3. Continuous Integration

Continuous Integration is a technique for integrating the source code updates from all developers working on the project into the main branch regularly and automatic build checks for errors. The continuous integration of code prevents developers from merge conflicts.

4. Continuous Testing

The testing process is easy till cost starts rising exponentially, impact of software failures is also very dangerous, no one wants to make a release that may affect the user experience of its customers, introduction of new features may expose the organization to a security threat, affects reliability, and compliance-related risks.
It is not only a Quality Assurance function but it starts from the development phase.

System Components & Functionalities:

Arduino: Arduino is a prototype platform (open source) based on an easy to use hardware and software. It consists of a circuit board, which can be programmed (referred to as a micro controller) and a ready-made software called Arduino IDE(Integrated Development Environment), which is used to write and upload the computer code to the physical board.

computer code to the physical board.

ABSTRACT

The efforts required in achieving required output can be effectively and economically be decrease by the implementations of better designs. If you design well of the project then will easily be able to create your project in less time with respect to others. So it is very important to create your designs or patterns first.

Over the past few years, IoT has become one of the most important technologies of the 21st century. Now that we can connect everyday objects Kitchen appliances, cars, thermostats, baby monitors to the internet via embedded devices, seamless communication is possible between people, process, and things.By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyper connected world, digital systems can record, monitor and adjust each interaction between connected things. The physical world meets the digital world -- and they cooperate.IoT (Internet of Things) is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.IoT systems have applications across industries through their unique flexibility and ability to be suitable in any environment. They enhance data collection, automation, operations, and much more through smart devices and powerful enabling technology.IoT systems allow users to achieve deeper automation, analysis, and integration within a system. They improve the reach of these areas and their accuracy. IoT utilizes existing and emerging technology for sensing, networking, and robotics.IoT exploits recent advances in software, falling hardware prices, and modern attitudes towards technology. Its new and advanced elements bring major changes in the delivery of products, goods, and services; and the social, economic, and political impact of those changes.

Here in my project "Smart Water Management System" it's all about IoT based project where its entire process is done using Arduino uno in which it is connected with Relay switch and main electricity and water pump. They are interconnected in such a way so that it will take an input from the users/consumers and will give the proper output.

Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Arduino has many key features:

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on or off, connect to the cloud and many other actions can be performed.
- You can control your board functions by sending a set of instructions to the micro controller on the board via Arduino IDE.
- Arduino software does not need another piece of hardware in order to load a new code onto the board. You can simply use a USB cable with comparison of other circuit boards programming.
- The Arduino uses a simplified version of C++, making it easier to learn to program.

It makes IOT works easier and breaks the functions of the microcontroller.

Relay Board:

Relay boards are the computer boards with an array switches. They have input and output terminals and are design to control the voltage supply or also can say that it's working is as similar as the transformer. Means both can worked for voltage control. Relay boards independently programmable and provide a real control for each several onboard relay channels.

Following are the test a relay:

1. Keep the multi meter in the continuity check mode.
2. Check for continuity between the N/C contacts and pole.
3. Check for discontinuity between between N/O contacts and the pole.
4. Now energize the relay using the rated voltage.
5. Now check for continuity between N/O contacts and pole.

Water Pump:

Water Pump is so a common type of pumps that they can be found at home, in fields, on farms and other places. They are exclusively used for displacing water. Water pumps run on different sources of power.Also in water cooler you can see water pump and how the whole process are carry on. Firstly it is connected to the electricity and whole part is dipped into the water and having some inner process where all the pumping steps done and is able to pull water and pour into some other places.

Results

My study found that when someone enter any input like for my project if they enter how much amount of water he/she wants it will give instructions to the Arduino that you have to provide such amount of water only and and through wire it'll give desired output to the consumers and will take such an amount of water. Like if you choose 1 litre of water it'll give you the same amount of water, if you choose 20 litres of water it'll give you 20 litres of water .

And after that you'll have been provided total amount of costs per litre and also will be able to see the amount that consumers used in a given period of time.

Conclusion

Performance Estimation

It may help to collect and providing help flawlessly in detail. In a very short time, the collection will be obvious, simple, and helpful. It will help any person looking for intricate guidance to plan a small business or startup. It also helps in managing all the current works relative to the user's needs. It will also contribute to the reduction of the general cost of software maintenance by 80%.

6.2 Usability of System

Our project briefly aims at computerizing and digitalizing the process of startup planning and setting some ground rules of the budget and success of an application. It also creates a user-friendly environment to make the website more appealing and helping to the user who visits later using Software Testing.

- In a computer system, it is not necessary to create copies of any paperwork as all the required details are filled and managed online only.
- To assist the staff in capturing the effort spent on their respective working areas.
- In a computer system, the person has to fill in the required personal to make the results efficient and meeting their needs.
- To utilize the resources efficiently by increasing their productivity through automation.
- It satisfies the user requirement working in any field for complete guidance.
- Be easy to understand by the user and the operator.

Figure 1: Example of the Grid-Based Bulk Supply System in Purba Medinipur District

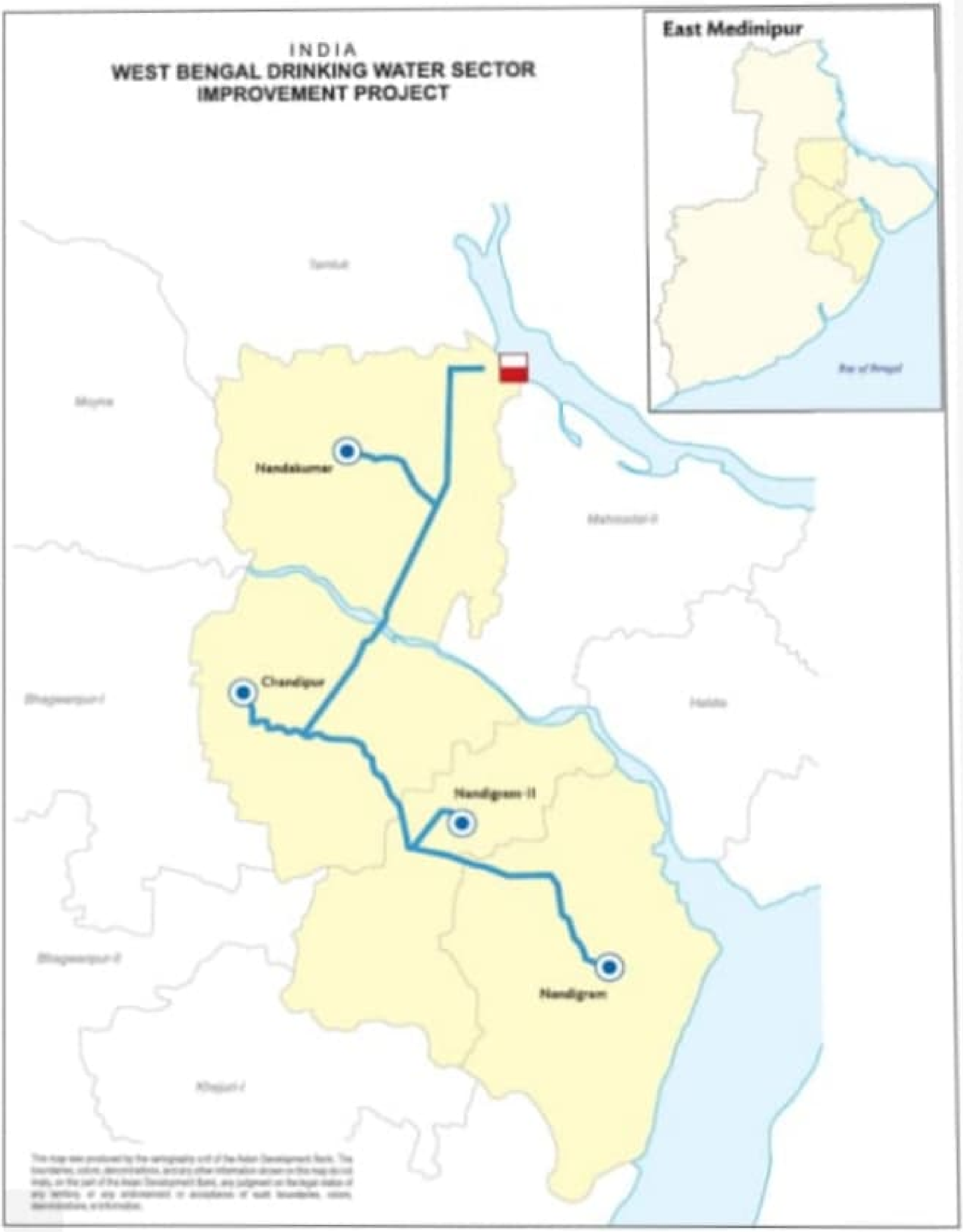

INDIA
WEST BENGAL DRINKING WATER SECTOR IMPROVEMENT PROJECT
East Medinipur
Bay of Bengal
Tamluk
Moyna
Nandakumar
Mahisadal-II
Bhagwanpur-I
Chandipur
Haldia
Nandigram-II
Bhagwanpur-II
Nandigram
Khejuri-I
This map was produced by the cartography unit of the Asian Development Bank. The boundaries, colors, denominations, and any other information shown on this map do not imply, on the part of the Asian Development Bank, any judgment on the legal status of any territory, or any endorsement or acceptance of such boundaries, colors, denominations, or information.

Figure 2: Schematic of Smart Water Management under the Project

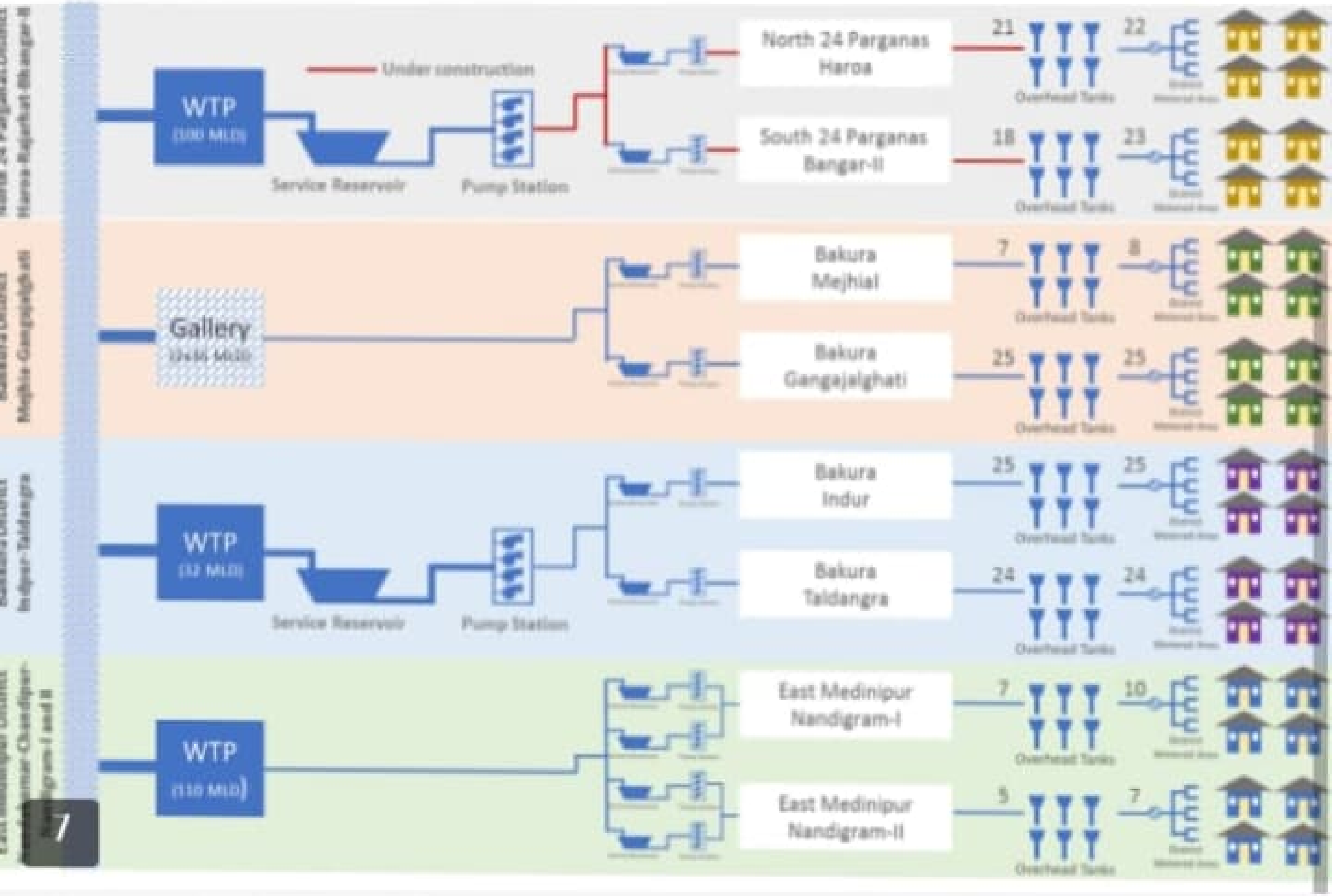


Figure 3: Data Integration under the Project's Smart Water Management

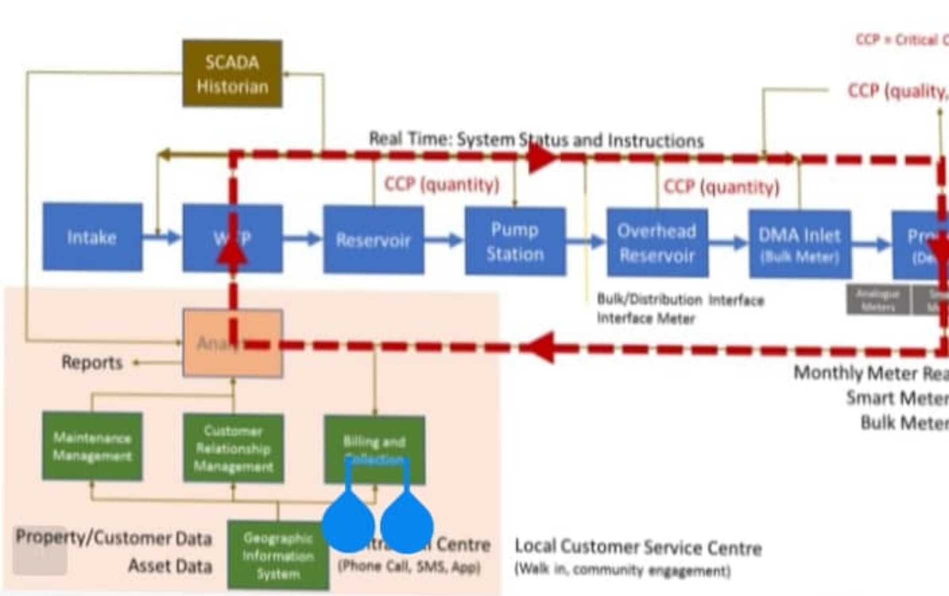
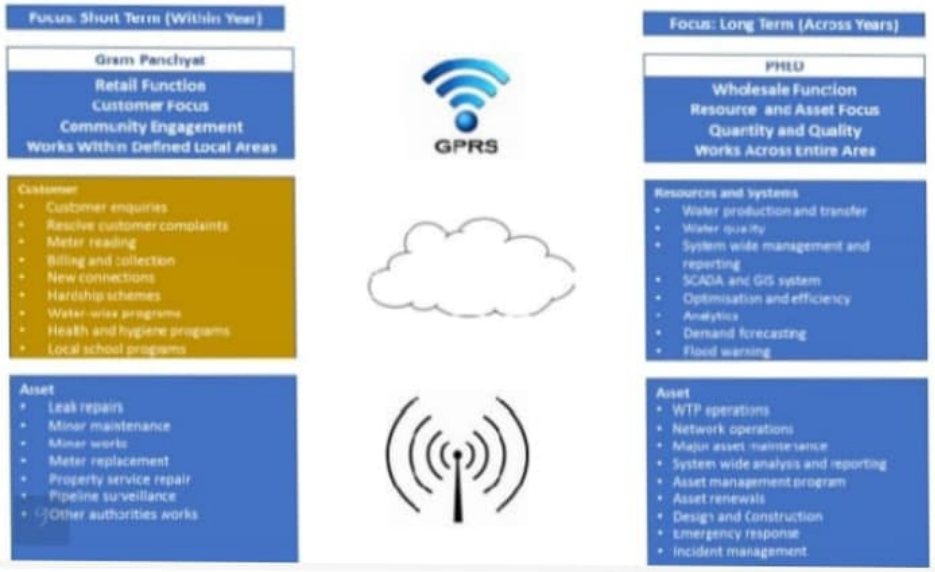


Figure 4: Smart Water Management Functions under the Project



6.3 Limitations

- **Ambiguous:**

  Since IoT is new and unclear on its definition, goals of adoption, and lack of understanding of its concepts.

- **Management Structure:**

  Between developers and operators, there is a lack of management structure as IoT yet is not systematically managed and organized.

- **Training:** As it is new most operators and Developers lack proper understanding or training of its operations and Principles hence new technology tools and methods of IoT are not clearly understood.

- **Experience:**Shortage of IoT experienced individuals hence the whole concept is learned and practices to obtain experience at present time and apply in the project, hence it may result in a long time of its implementation.

Scope of Improvements

- We will try to make it compatible with all types of deployment. Example – When we need immediate patches or changes to be applied.
- We can secure the apps by adopting Industry Standard Best Practices.

Streamline the Testing Process after the development is completed.

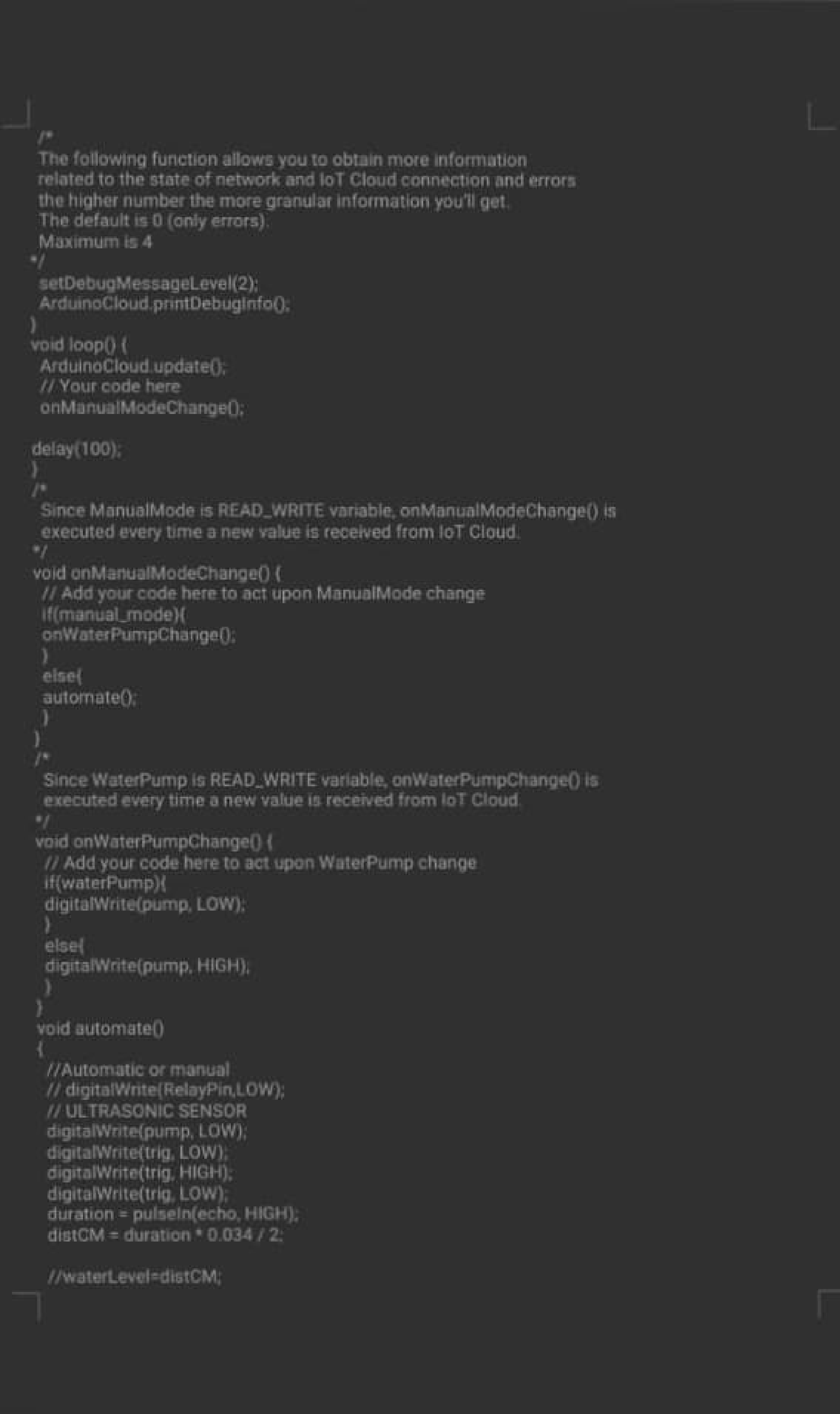
supervisory control and data acquisition (SCADA) and geographic information system (GIS) monitoring at the PHED. The new technology is designed to be implemented across the entire water supply system using advanced practices and computer technology. It is based on a whole-of-system approach that treats the water supply system as a single operating entity rather than a series of separate but connected components. The distribution network has been designed on district-metering area-based approach. Figures 2, 3, and 4 show the schematic of the smart water management approach and design under the project and how data integration will work to improve operational efficiency;

(iii) ADB's project preparatory team brought in international and local expertise on smart water management during project preparations to assist with developing the requirements and specifications of the new systems in close consultation with the stakeholders;

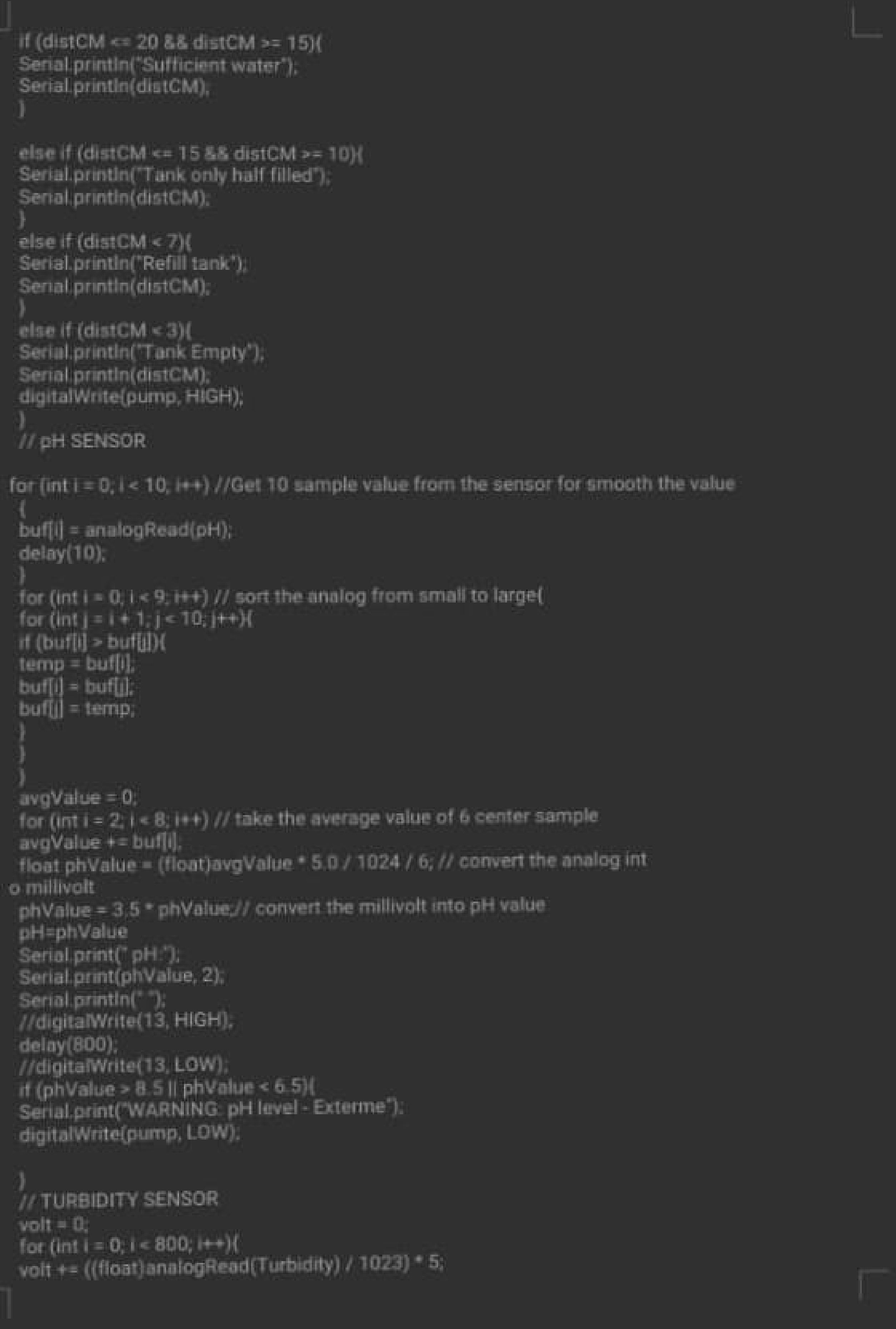
(iv) PHED (state level) will be responsible for the bulk supply of water into the system up to the boundary of the *gram panchayats* and for major repairs. PHED will also assist the *gram panchayats* in continued technical and utility management training, regulate the services, and assemble and integrate the expertise required to develop the smart systems that will deliver operational efficiency and reduce failure risk;

(v) The *gram panchayats* will be responsible for the management of the distribution network and minor repairs of the new system. This will help in fostering local ownership while providing opportunities for local employment and skills to sustain the benefits of the water supply systems and improved health impacts; and

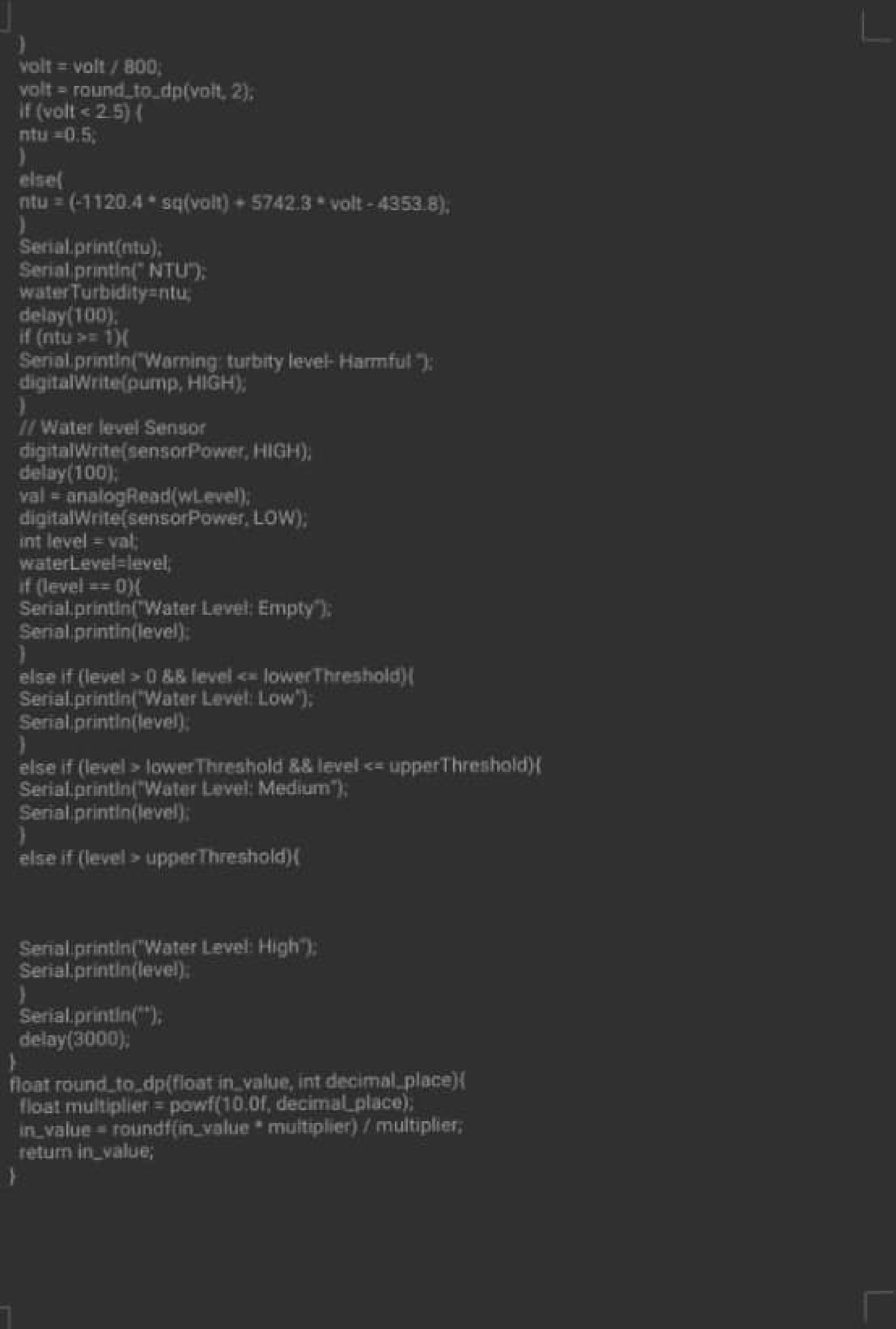
(vi) Training and capacity building programs are developed and will be delivered to target the exact skills that will be required at both PHED and *gram panchayat* levels for smart water management. The project will strengthen the integrated management information systems at central, state, and urban local body or village levels for project implementation and monitoring of water quality and benefits realized.



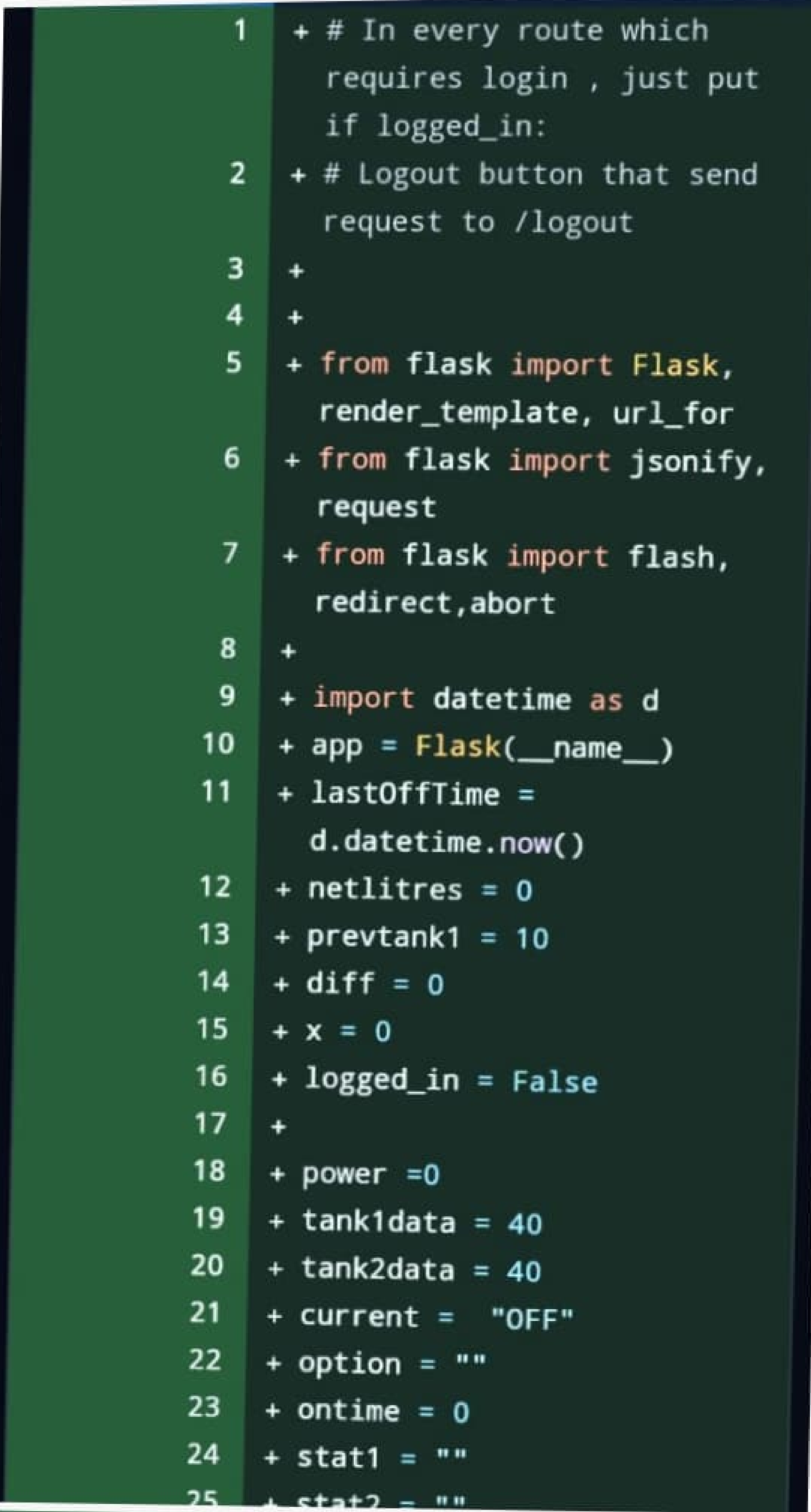
```
 /*
 The following function allows you to obtain more information
 related to the state of network and IoT Cloud connection and errors
 the higher number the more granular information you'll get.
 The default is 0 (only errors).
 Maximum is 4
*/
 setDebugMessageLevel(2);
 ArduinoCloud.printDebugInfo();
}
void loop() {
 ArduinoCloud.update();
 // Your code here
 onManualModeChange();

delay(100);
}
/*
 Since ManualMode is READ_WRITE variable, onManualModeChange() is
 executed every time a new value is received from IoT Cloud.
*/
void onManualModeChange() {
 // Add your code here to act upon ManualMode change
 if(manual_mode){
 onWaterPumpChange();
 }
 else{
 automate();
 }
}
/*
 Since WaterPump is READ_WRITE variable, onWaterPumpChange() is
 executed every time a new value is received from IoT Cloud.
*/
void onWaterPumpChange() {
 // Add your code here to act upon WaterPump change
 if(waterPump){
 digitalWrite(pump, LOW);
 }
 else{
 digitalWrite(pump, HIGH);
 }
}
void automate()
{
  //Automatic or manual
  // digitalWrite(RelayPin,LOW);
  // ULTRASONIC SENSOR
  digitalWrite(pump, LOW);
  digitalWrite(trig, LOW);
  digitalWrite(trig, HIGH);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distCM = duration * 0.034 / 2;

  //waterLevel=distCM;
```



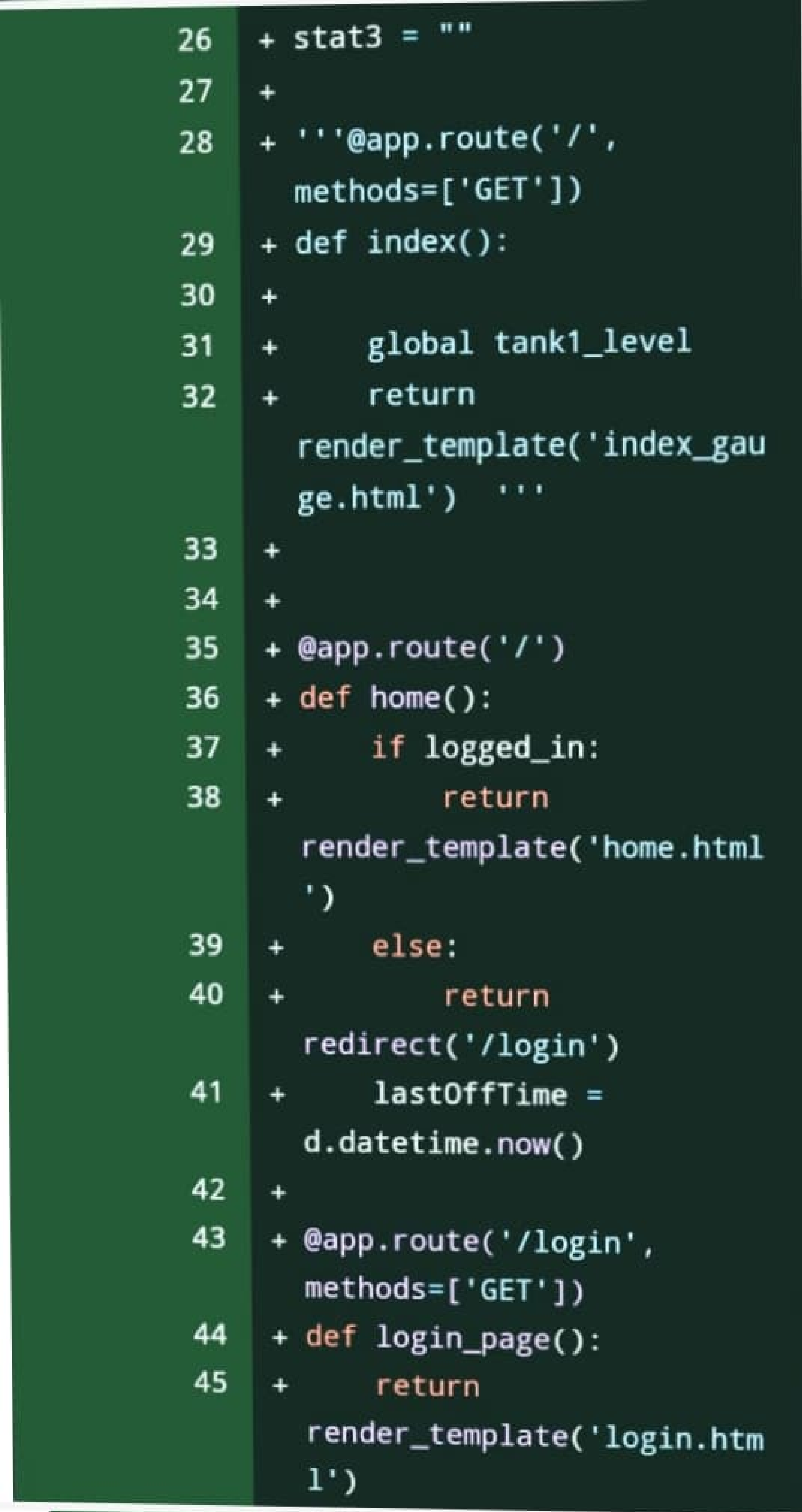
```
  if (distCM <= 20 && distCM >= 15){
  Serial.println("Sufficient water");
  Serial.println(distCM);
  }

  else if (distCM <= 15 && distCM >= 10){
  Serial.println("Tank only half filled");
  Serial.println(distCM);
  }
  else if (distCM < 7){
  Serial.println("Refill tank");
  Serial.println(distCM);
  }
  else if (distCM < 3){
  Serial.println("Tank Empty");
  Serial.println(distCM);
  digitalWrite(pump, HIGH);
  }
  // pH SENSOR

for (int i = 0; i < 10; i++) //Get 10 sample value from the sensor for smooth the value
  {
  buf[i] = analogRead(pH);
  delay(10);
  }
  for (int i = 0; i < 9; i++) // sort the analog from small to large{
  for (int j = i + 1; j < 10; j++){
  if (buf[i] > buf[j]){
  temp = buf[i];
  buf[i] = buf[j];
  buf[j] = temp;
  }
  }
  }
  avgValue = 0;
  for (int i = 2; i < 8; i++) // take the average value of 6 center sample
  avgValue += buf[i];
  float phValue = (float)avgValue * 5.0 / 1024 / 6; // convert the analog int
o millivolt
  phValue = 3.5 * phValue;// convert the millivolt into pH value
  pH=phValue
  Serial.print(" pH:");
  Serial.print(phValue, 2);
  Serial.println(" ");
  //digitalWrite(13, HIGH);
  delay(800);
  //digitalWrite(13, LOW);
  if (phValue > 8.5 || phValue < 6.5){
  Serial.print("WARNING: pH level - Exterme");
  digitalWrite(pump, LOW);

  }
  // TURBIDITY SENSOR
  volt = 0;
  for (int i = 0; i < 800; i++){
  volt += ((float)analogRead(Turbidity) / 1023) * 5;
```



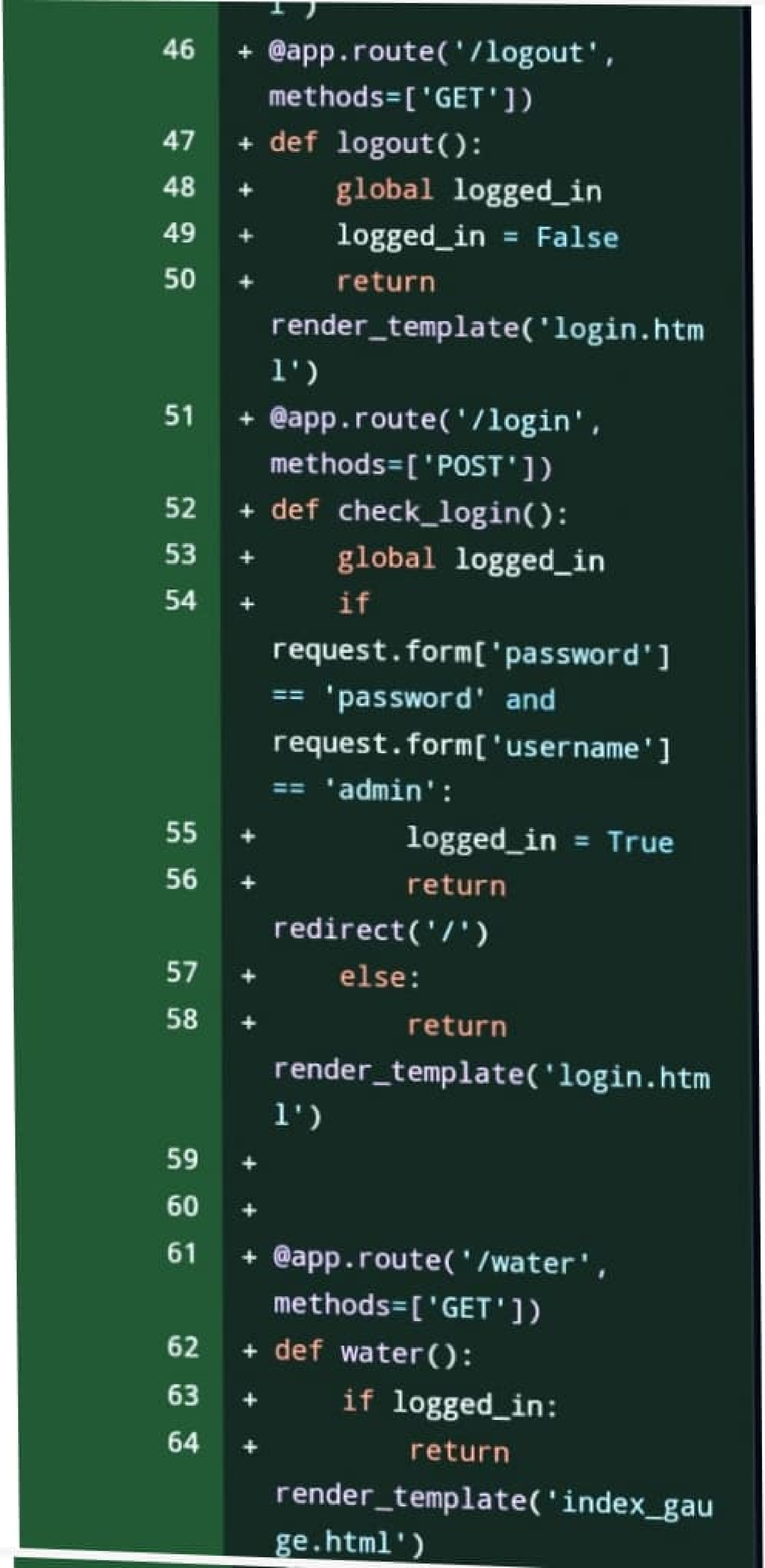
```
 }
 volt = volt / 800;
 volt = round_to_dp(volt, 2);
 if (volt < 2.5) {
 ntu =0.5;
 }
 else{
 ntu = (-1120.4 * sq(volt) + 5742.3 * volt - 4353.8);
 }
 Serial.print(ntu);
 Serial.println(" NTU");
 waterTurbidity=ntu;
 delay(100);
 if (ntu >= 1){
 Serial.println("Warning: turbity level- Harmful ");
 digitalWrite(pump, HIGH);
 }
 // Water level Sensor
 digitalWrite(sensorPower, HIGH);
 delay(100);
 val = analogRead(wLevel);
 digitalWrite(sensorPower, LOW);
 int level = val;
 waterLevel=level;
 if (level == 0){
 Serial.println("Water Level: Empty");
 Serial.println(level);
 }
 else if (level > 0 && level <= lowerThreshold){
 Serial.println("Water Level: Low");
 Serial.println(level);
 }
 else if (level > lowerThreshold && level <= upperThreshold){
 Serial.println("Water Level: Medium");
 Serial.println(level);
 }
 else if (level > upperThreshold){



 Serial.println("Water Level: High");
 Serial.println(level);
 }
 Serial.println("");
 delay(3000);
}
float round_to_dp(float in_value, int decimal_place){
 float multiplier = powf(10.0f, decimal_place);
 in_value = roundf(in_value * multiplier) / multiplier;
 return in_value;
}
```



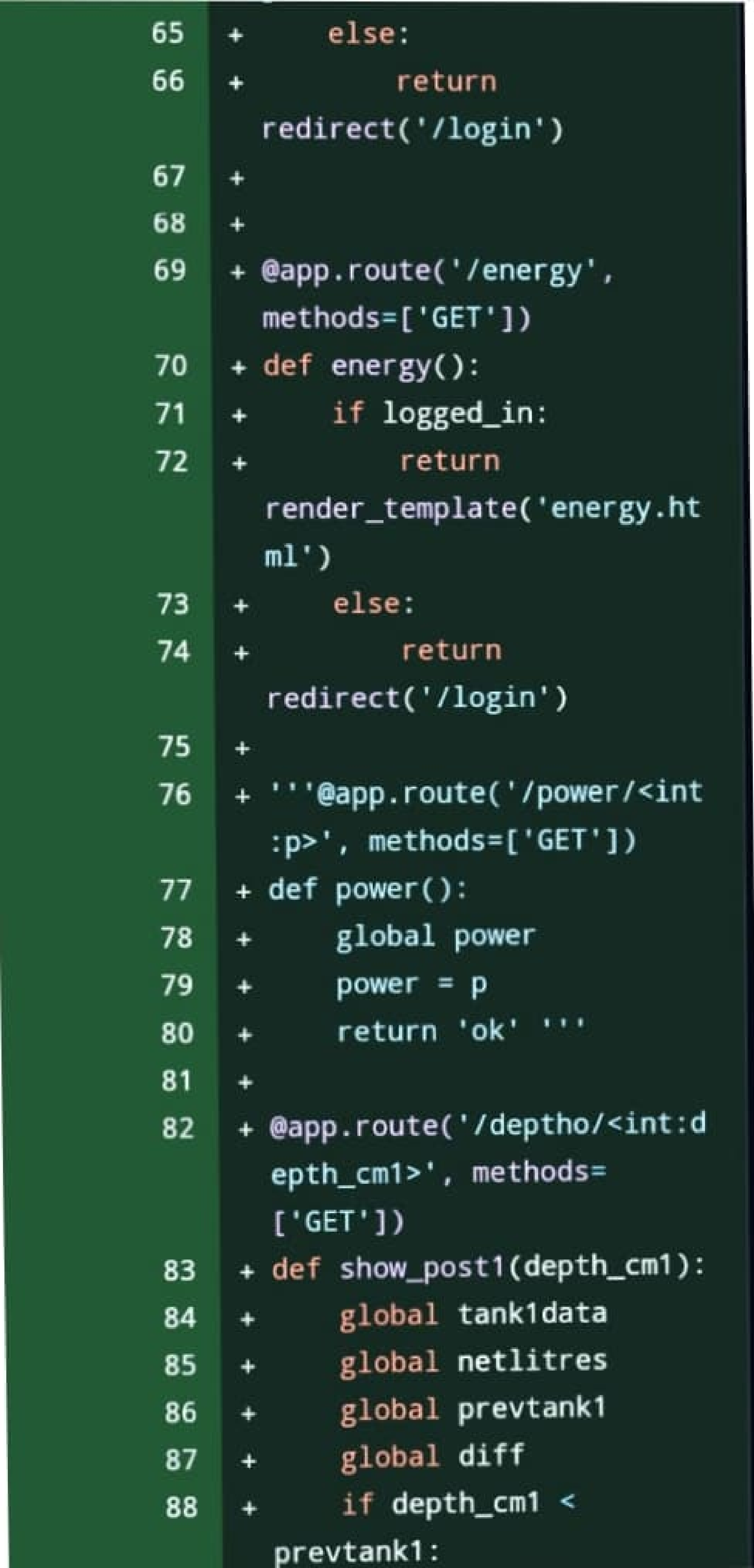
```
+ # In every route which requires login , just put if logged_in:
+ # Logout button that send request to /logout
+
+
+ from flask import Flask, render_template, url_for
+ from flask import jsonify, request
+ from flask import flash, redirect,abort
+
+ import datetime as d
+ app = Flask(__name__)
+ lastOffTime = d.datetime.now()
+ netlitres = 0
+ prevtank1 = 10
+ diff = 0
+ x = 0
+ logged_in = False
+
+ power =0
+ tank1data = 40
+ tank2data = 40
+ current =  "OFF"
+ option = ""
+ ontime = 0
+ stat1 = ""
+ stat2 = ""
```



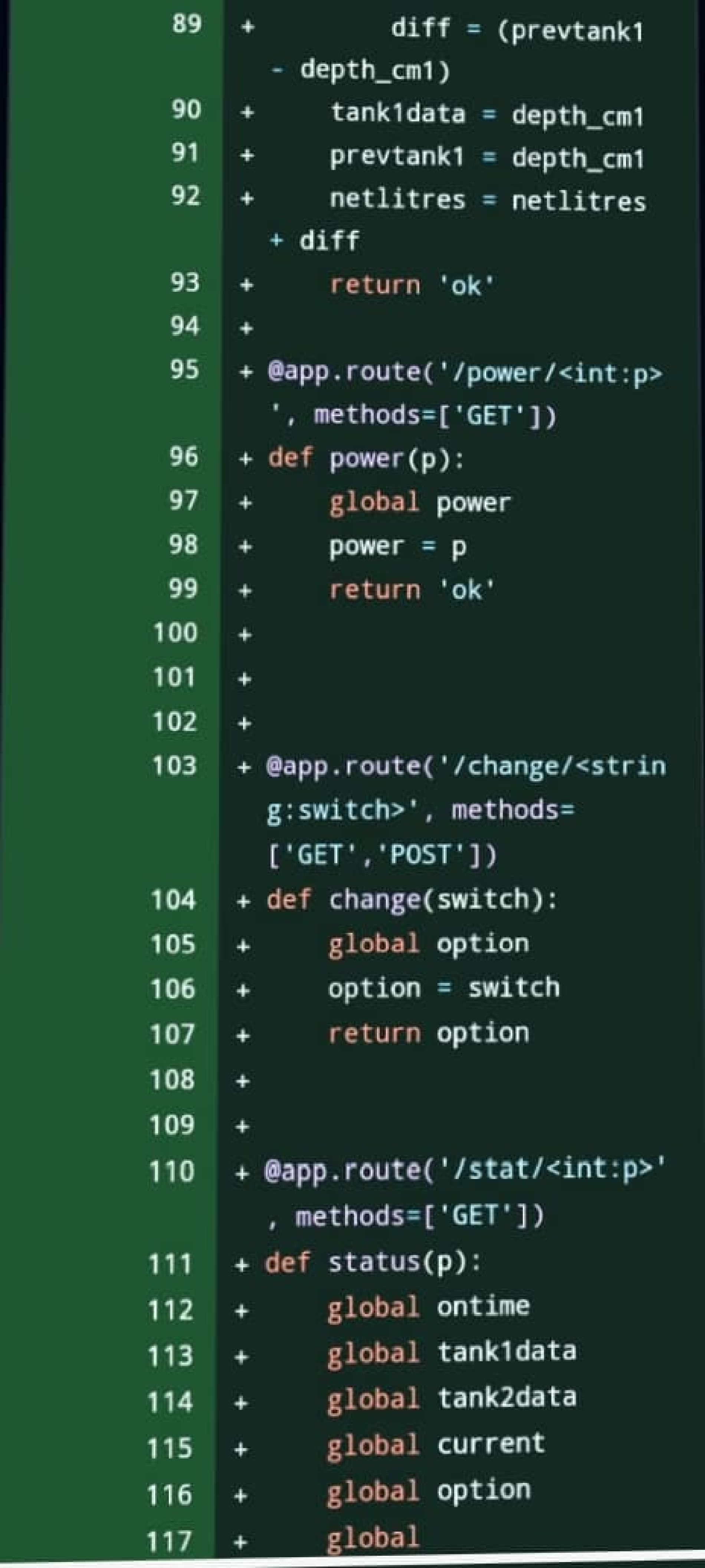
```
+ stat3 = ""
+
+ '''@app.route('/', methods=['GET'])
+ def index():
+
+     global tank1_level
+     return render_template('index_gauge.html')  '''
+
+
+ @app.route('/')
+ def home():
+     if logged_in:
+         return render_template('home.html')
+     else:
+         return redirect('/login')
+     lastOffTime = d.datetime.now()
+
+ @app.route('/login', methods=['GET'])
+ def login_page():
+     return render_template('login.html')
```



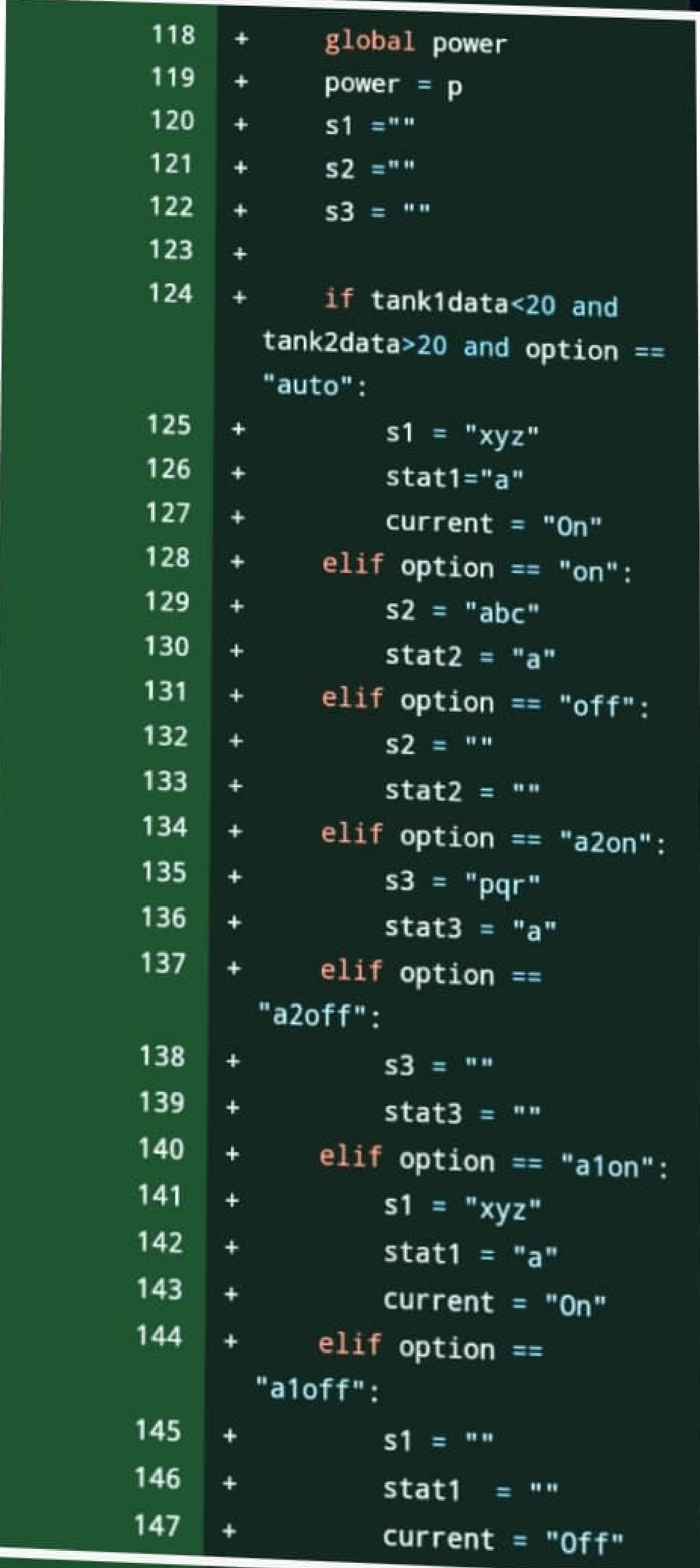
```
l')
+ @app.route('/logout', methods=['GET'])
+ def logout():
+     global logged_in
+     logged_in = False
+     return render_template('login.html')
+ @app.route('/login', methods=['POST'])
+ def check_login():
+     global logged_in
+     if request.form['password'] == 'password' and request.form['username'] == 'admin':
+         logged_in = True
+         return redirect('/')
+     else:
+         return render_template('login.html')
+
+
+ @app.route('/water', methods=['GET'])
+ def water():
+     if logged_in:
+         return render_template('index_gauge.html')
```



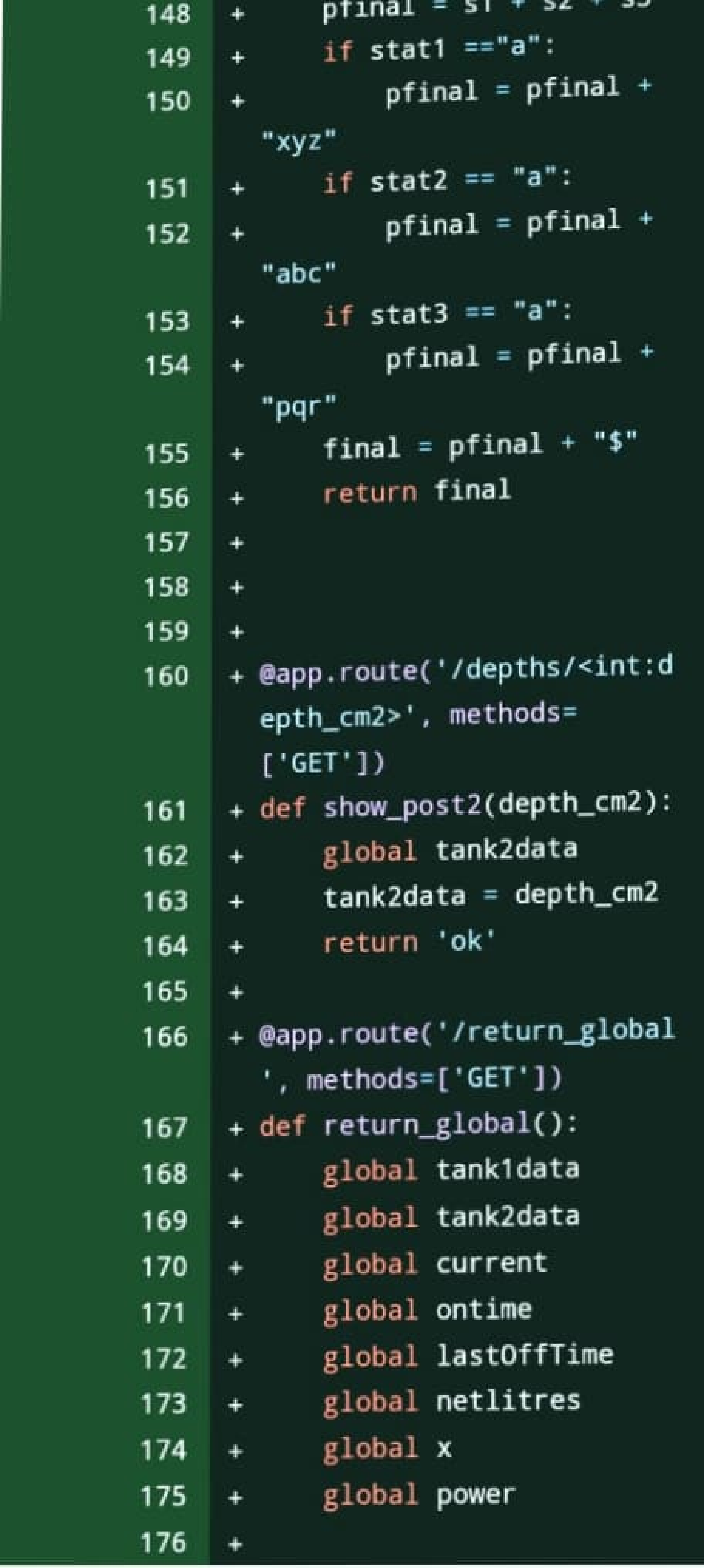
```
+     else:
+         return redirect('/login')
+
+
+ @app.route('/energy', methods=['GET'])
+ def energy():
+     if logged_in:
+         return render_template('energy.html')
+     else:
+         return redirect('/login')
+
+ '''@app.route('/power/<int:p>', methods=['GET'])
+ def power():
+     global power
+     power = p
+     return 'ok' '''
+
+ @app.route('/deptho/<int:depth_cm1>', methods=['GET'])
+ def show_post1(depth_cm1):
+     global tank1data
+     global netlitres
+     global prevtank1
+     global diff
+     if depth_cm1 < prevtank1:
```



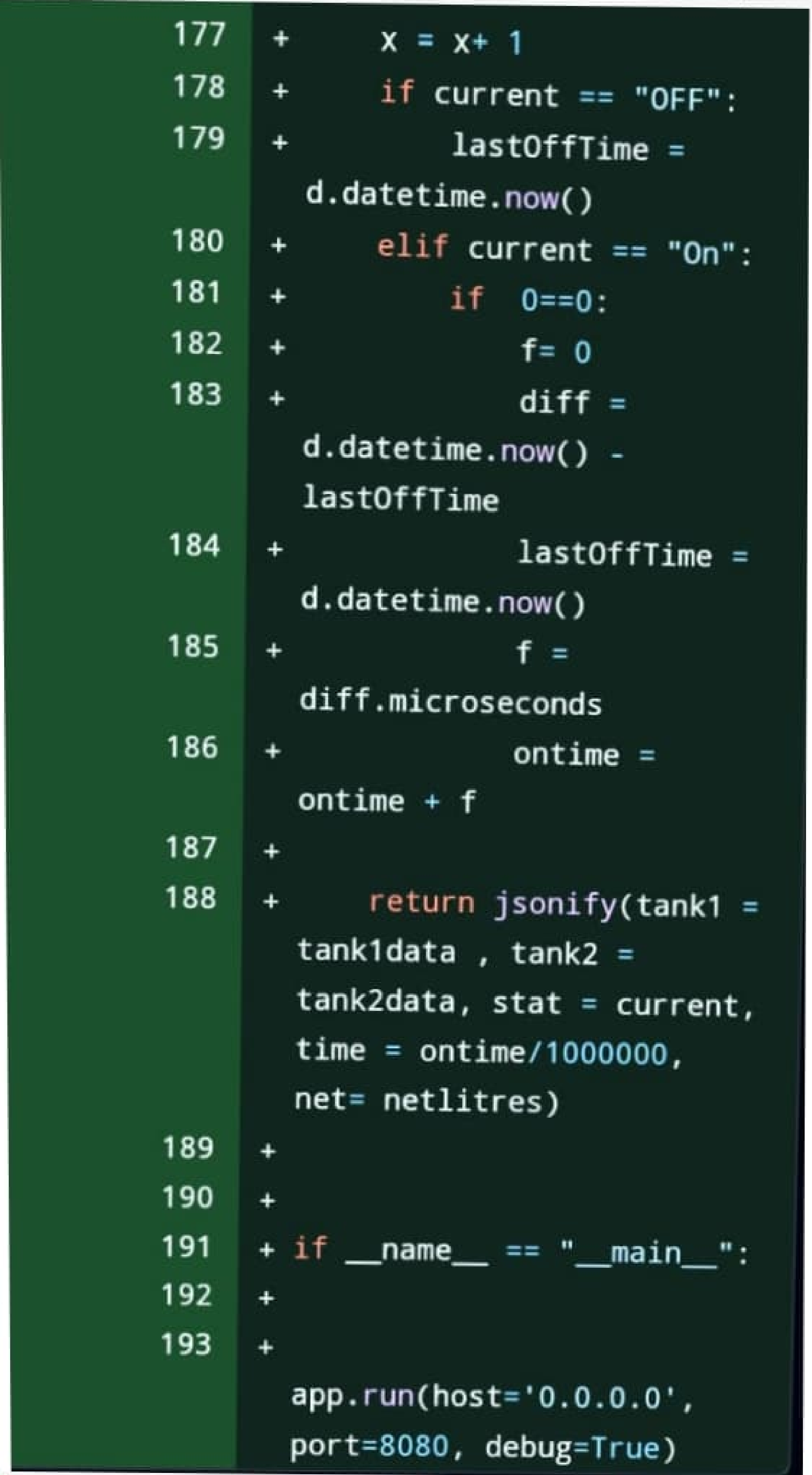
```
+         diff = (prevtank1 - depth_cm1)
+     tank1data = depth_cm1
+     prevtank1 = depth_cm1
+     netlitres = netlitres + diff
+     return 'ok'
+
+ @app.route('/power/<int:p>', methods=['GET'])
+ def power(p):
+     global power
+     power = p
+     return 'ok'
+
+
+
+ @app.route('/change/<string:switch>', methods=['GET','POST'])
+ def change(switch):
+     global option
+     option = switch
+     return option
+
+
+ @app.route('/stat/<int:p>', methods=['GET'])
+ def status(p):
+     global ontime
+     global tank1data
+     global tank2data
+     global current
+     global option
+     global
```



```
+    global power
+    power = p
+    s1 =""
+    s2 =""
+    s3 = ""
+
+    if tank1data<20 and tank2data>20 and option == "auto":
+        s1 = "xyz"
+        stat1="a"
+        current = "On"
+    elif option == "on":
+        s2 = "abc"
+        stat2 = "a"
+    elif option == "off":
+        s2 = ""
+        stat2 = ""
+    elif option == "a2on":
+        s3 = "pqr"
+        stat3 = "a"
+    elif option == "a2off":
+        s3 = ""
+        stat3 = ""
+    elif option == "a1on":
+        s1 = "xyz"
+        stat1 = "a"
+        current = "On"
+    elif option == "a1off":
+        s1 = ""
+        stat1  = ""
+        current = "Off"
```



```
+    pfinal = s1 + s2 + s3
+    if stat1 =="a":
+        pfinal = pfinal + "xyz"
+    if stat2 == "a":
+        pfinal = pfinal + "abc"
+    if stat3 == "a":
+        pfinal = pfinal + "pqr"
+    final = pfinal + "$"
+    return final
+
+
+
+ @app.route('/depths/<int:depth_cm2>', methods=['GET'])
+ def show_post2(depth_cm2):
+    global tank2data
+    tank2data = depth_cm2
+    return 'ok'
+
+ @app.route('/return_global', methods=['GET'])
+ def return_global():
+    global tank1data
+    global tank2data
+    global current
+    global ontime
+    global lastOffTime
+    global netlitres
+    global x
+    global power
+
```



```
+        x = x+ 1
+        if current == "OFF":
+            lastOffTime = d.datetime.now()
+        elif current == "On":
+            if  0==0:
+                f= 0
+                diff = d.datetime.now() - lastOffTime
+                lastOffTime = d.datetime.now()
+                f = diff.microseconds
+                ontime = ontime + f
+
+    return jsonify(tank1 = tank1data , tank2 = tank2data, stat = current, time = ontime/1000000, net= netlitres)
+
+
+if __name__ == "__main__":
+
+    app.run(host='0.0.0.0', port=8080, debug=True)
```