EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
Midterm Assessment Project, Fall 2023 Semester

Course:	**CSE407 Green Computing, Section: 03**
Student ID:	2020-2-60-173
Student Name:	Md. Naimul Islam

Project Title:
IoT based real-time energy monitoring and dashboard development.

Checklist:

- ✔ I used IoT technology to monitor the energy consumption of the appliance for a week.
- ✔ I have prepared an energy monitoring dashboard/webpage with detailed and summarized information.
- ✔ I have considered the financial and business aspects.
- ✔ I did PESTLE analysis on the project.

Declaration

I declare that the project was done by myself, and the submitted information is free from any unfair means. I complied with all the relevant legal and organizational requirements.

Signature

Name: Md. Naimul Islam

Date: 9/12/2023

1. Executive Summary

Through this IoT project, we were able to gather valuable insights into the energy consumption patterns of smart devices. Our selected device for the project was a smart TV and a DTH By measuring voltage and current characteristics, we were able to provide practical insights into their energy usage. Our project utilized a wireless sensor for remote monitoring, making it convenient for users to keep track of their device's energy consumption. The user interface of our web application was designed with the user's convenience in mind, ensuring it was user-friendly and intuitive. Safety features were also implemented to protect both the devices and users. We prioritized performance, scalability, and reliability in optimizing the system, ensuring its effectiveness and longevity. Overall, this project provided a practical solution for monitoring the energy consumption of connected devices, which can lead to significant cost savings and environmental benefits.

2. Brief description of the work

Planning: planning was to find the overall costing and wattage of the desired devices for measurement. After knowing the wattage of the devices which we want to measure our next plan was to find the suitable smart plug or any other device. After some analysis we went on to select the device for measurement of this particular IoT project was a Smart TV and a DTH device.

Researching: Now we selected our device, our next task is to research to select the correct measuring device. We had some choice like automatic approcah of buying a simple smart plug or going on a manual made device with voltage and current sensors. But there was another problem which is to show the real time data in a webpage. To show that data we got to know that we would need an API to conduct this. By the automated smart plug we tried the way of cracking the api but we failed. After that we went on to go on a manual approach to buy voltage current sensors to conduct the project according to the instruction given to us.

Determine objective and Budget:

Our project sets out with a few key objectives:

Accurate Measurements: We want to develop a straightforward yet precise method for measuring the voltage and current drawn by different electronic devices. This involves using readily available tools and instruments to gather real-time data without imposing a significant burden on the devices under study.

Device-Specific Insights: By profiling the voltage and current characteristics of devices like laptops, desktops, and other common electronics, we aim to provide practical insights into their energy consumption patterns. This information can guide users and manufacturers toward more energy-efficient choices.

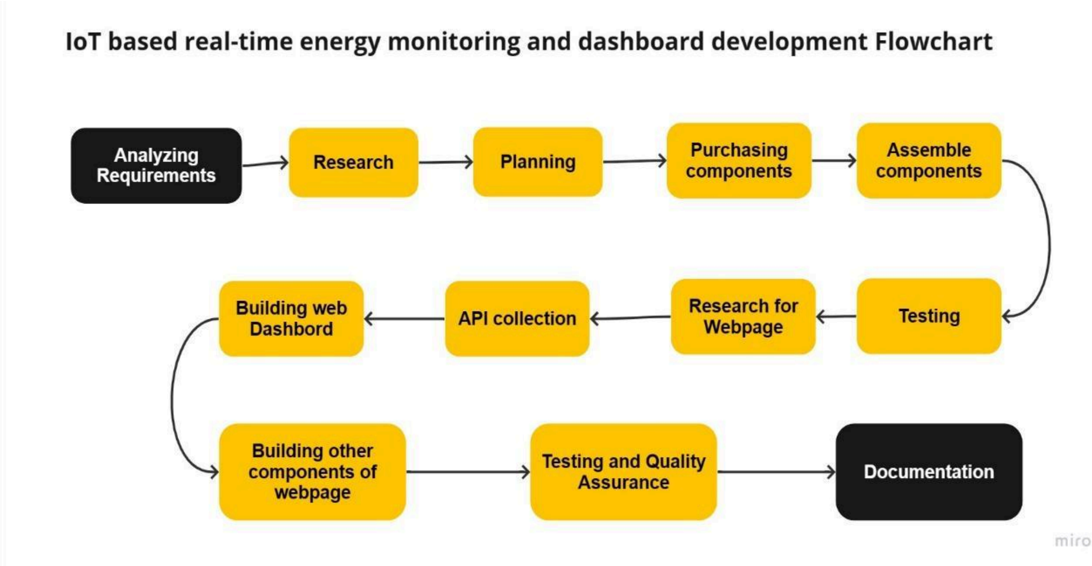
Budget: We had to buy some device so we definitely need some monetary fund in order to conduct this IoT project. After doing the market research, we made the budget.As there are only two members in our group. We split the total budget by two in order to purchase all the equipment.

Purchasing: After deciding what to purchase for our project again we had to do some research like where to find those devices definitely we had the costs in our mind and also the availability in the market. After that we purchased our desired equipment's which will be given in the detailed description part.

Configuration of Hardware: Now the devices are being purchased, now we need to configure the purchased device to connect those with the electronic devices to get the data. As mentioned before we were going on a manual approach, so that we made a manual device by the purchased device and kept them in a box for safety and connected the device with our made device for getting the data. Data from the device going directly to the cloud and are being shown in our own webpage which would be a real time data.

Webpage Data configuration: Our custom made IOT socket monitors data with per minute interval . We keep monitoring our data from our webpage with 5 minutes of interval. Starting from a certain date for the next 7 days. Except the load shedding times, we were getting the data from the electronic devices.

3. **Flowchart of the work**



4. **Detailed description of each of the steps**

Analyzing Requirements: This step involves understanding the specific needs and goals of the energy monitoring system. This includes identifying the types of energy to be monitored, the required level of accuracy, and the desired data storage and analysis capabilities.
Research Planning: Once the requirements have been analyzed, a research plan can be developed to identify the best possible hardware, software, and communication technologies for the system.
Purchasing Components: Once the research planning phase is complete, the necessary components was purchased. This includes energy sensors, data acquisition hardware, communication devices, and software development tools.

Components:

- Esp8266 nodeMCU
- ZMPT101B AC Voltage Sensor
- SCT013 Current Sensor
- 10K ohm Resistor
- 10 micro–F Capacitor
- Breadboard, wires

Assembling Components: Once the components have been purchased, they need to be assembled into a working system. This step may involved wiring the sensors to the data acquisition hardware, configuring the communication devices, and installing the software.

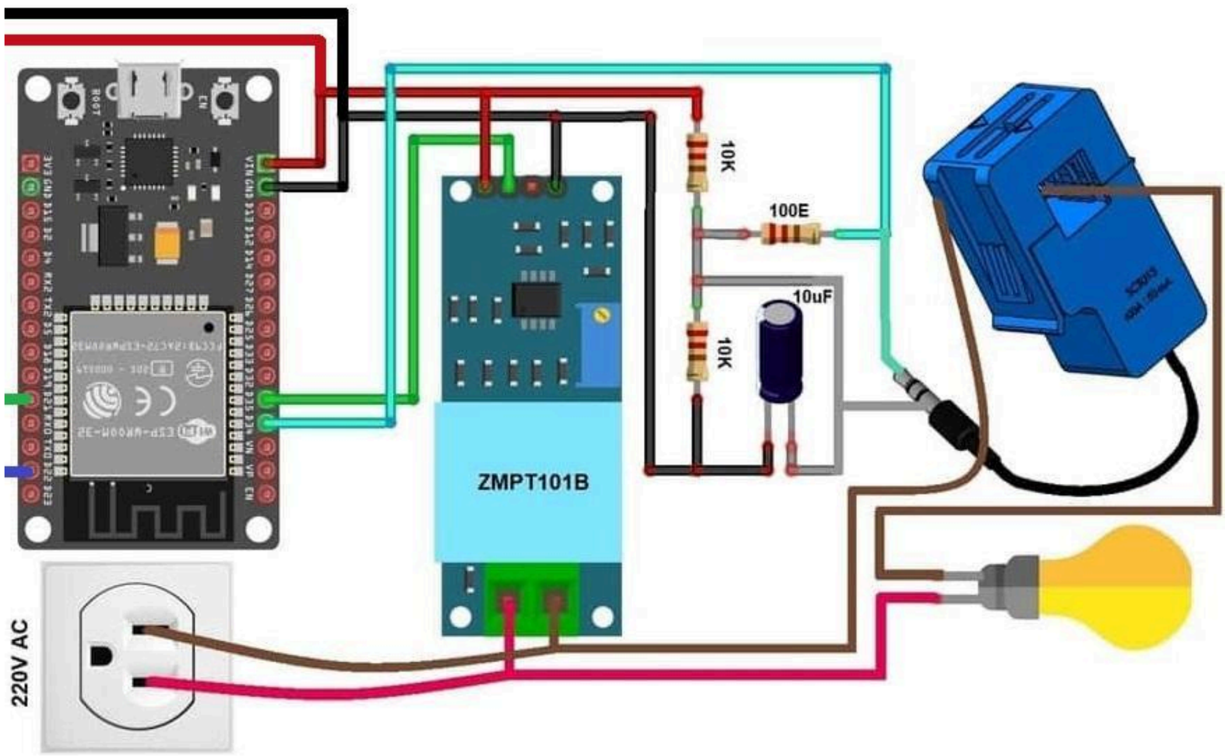


Figure: Circuit Diagram

Testing of Assembled Components: Once the system has been assembled, it needs to be thoroughly tested to ensure that it is working properly. This includes testing the accuracy of the energy sensors, the reliability of the communication devices, and the functionality of the software.

Documentation: Once the system has been tested and verified, we wrote documentation. This documentation include a description of the system architecture, component specifications, assembly instructions, and operating procedures.

Building Web Dashboard: Once the hardware and software components of the energy monitoring system are in place, a web dashboard is developed to provide users with real-time data visualization and analysis capabilities. The dashboard should be designed to be user-friendly and easy to navigate.

API Collection: To integrate the energy monitoring system with real time data we needed google sheet API (Application Programming Interface). The API will allow the dashboard to collect real-time data from the system and display it to users.

Research for Webpage: Once the API has been developed, the web dashboard can be built. This step may involved researching different web development frameworks and libraries to create a responsive and user-friendly interface.

Building Other Components of Webpage: In addition to the real-time energy monitoring dashboard, the web application may also include other components such as history, graphs, and reporting features.

Testing and Quality Assurance: Once the web dashboard has been developed, it needs to be thoroughly tested to ensure that it is working properly. This includes testing the accuracy of the data displayed, the security of system, and the functionality of the reporting features.

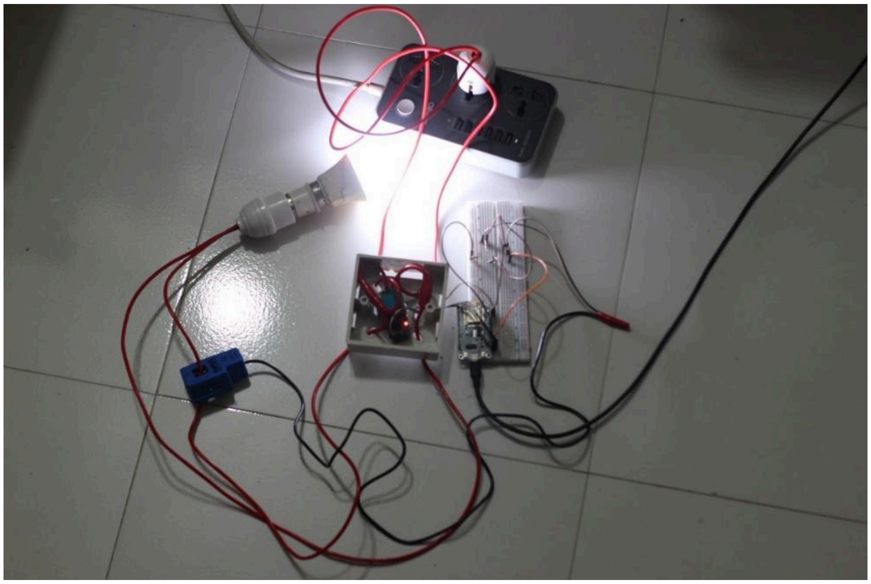
Documentation: Once the web dashboard has been tested and verified. We created detailed documentation. This documentation should include a description of the dashboard features, user instructions, and troubleshooting tips.
Challenges encountered during our IoT project primarily revolved around sourcing essential components such as ESP32 sensors and microcontrollers, given our adoption of a manual approach. The significant hurdle, however, lay in integrating these devices with electronic counterparts, as compatibility issues could potentially lead to safety hazards. To mitigate this risk, we secured the measuring device within a wooden box, prioritizing safety.

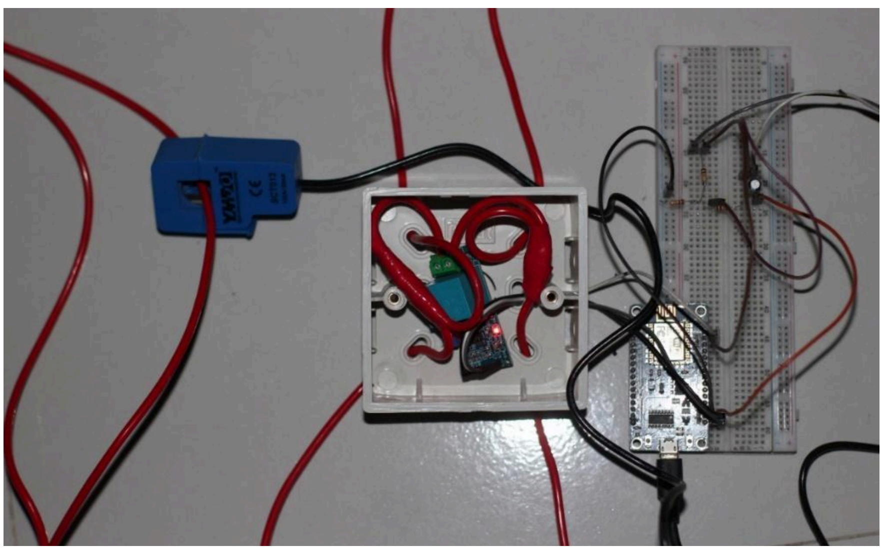
5. **Challenges and hiccups**

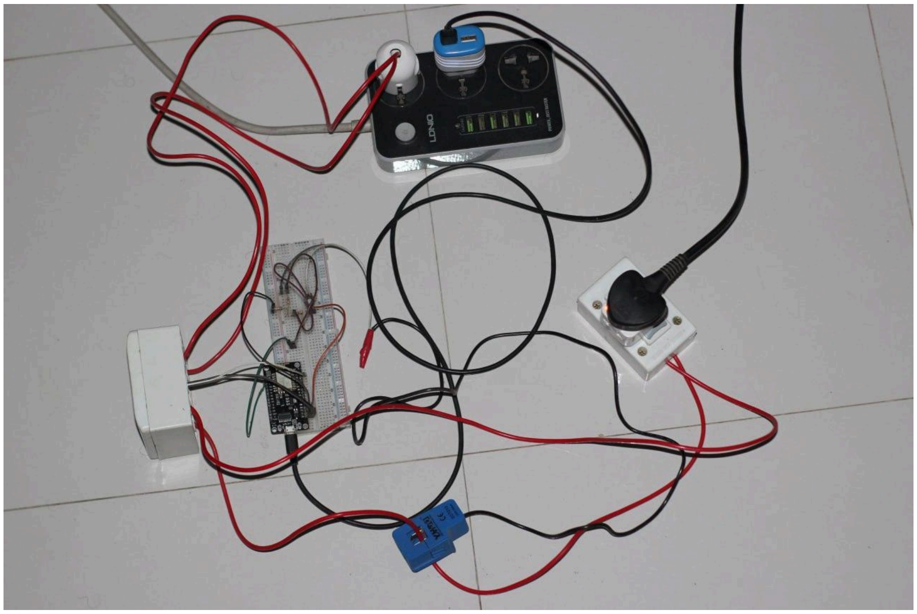
Following the setup, the subsequent challenge involved ensuring the proper functionality of the devices. We encountered initial data inconsistencies that required resolution to proceed. The 7-day monitoring period commenced two days post-device setup. While the data collection spanned the entire duration, we observed occasional inaccuracies, acknowledging the inherent imperfections in achieving efficiency. Notably, our device operated successfully for over 100% of the total time.

6. **Demonstration**

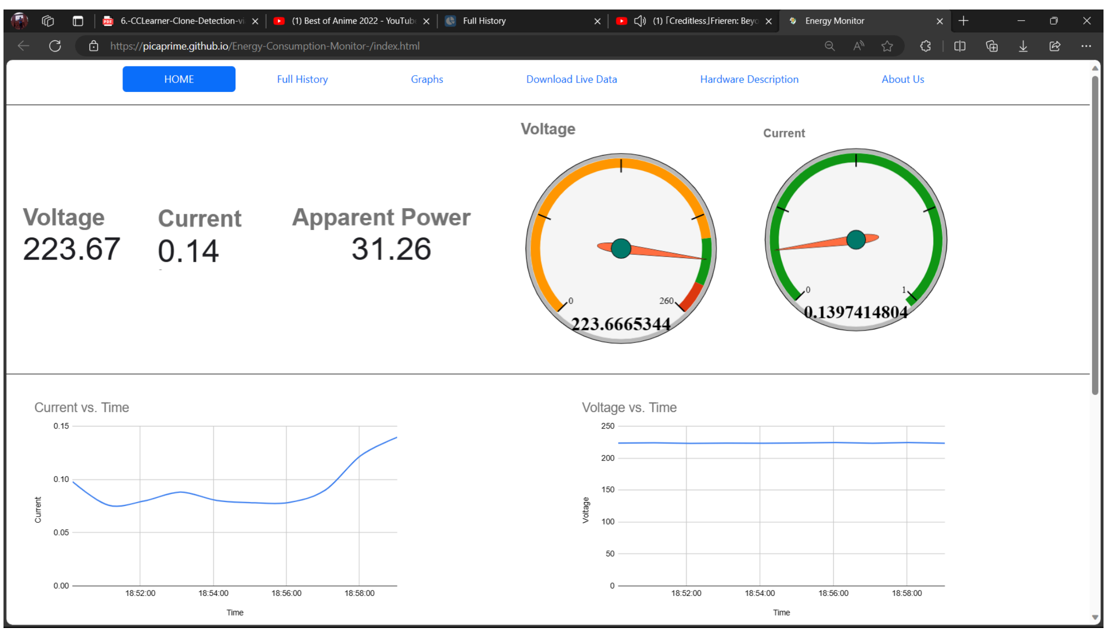
- Link to energy dashboard: [Energy Monitor (picaprime.github.io)](picaprime.github.io)
- Pictures:

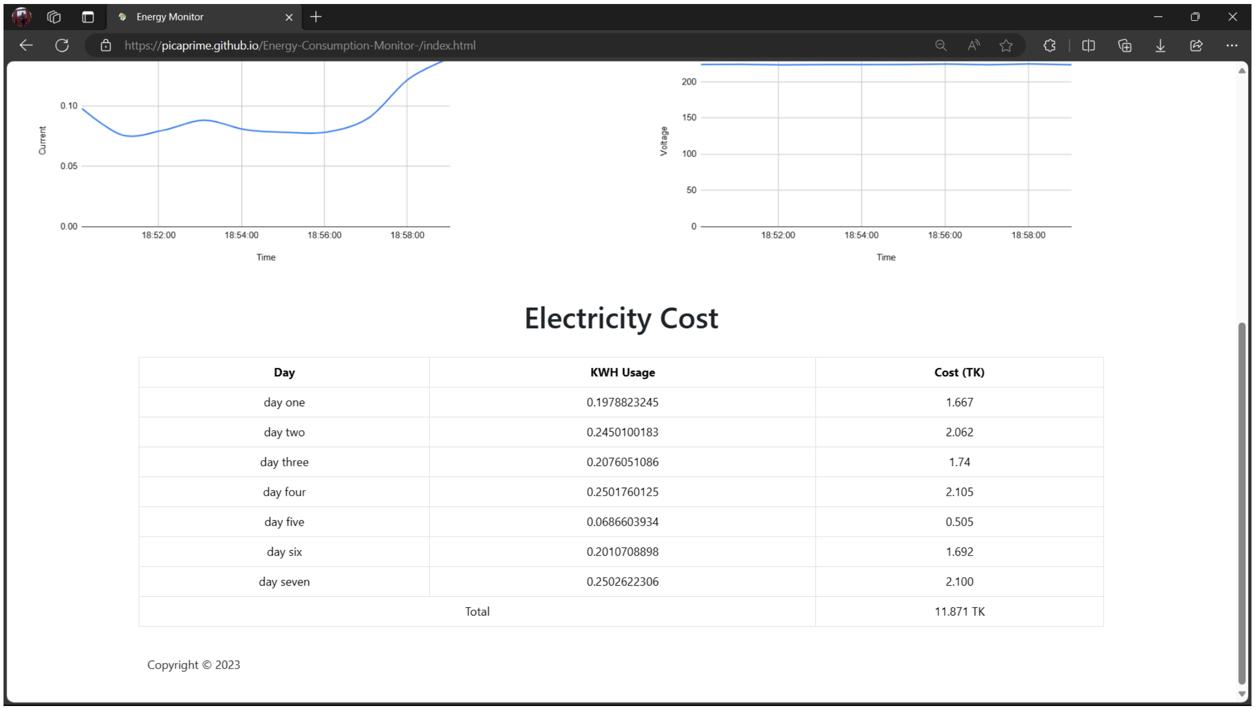






Web Dashboard:





Day	KWH Usage	Cost (TK)
day one	0.1978823245	1.667
day two	0.2450100183	2.062
day three	0.2076051086	1.74
day four	0.2501760125	2.105
day five	0.0686603934	0.505
day six	0.2010708898	1.692
day seven	0.2502622306	2.100
Total		11.871 TK

Copyright © 2023

7. Discussion on issues:

The creation of a real-time energy monitoring system, rooted in IoT technology and accompanied by an intuitive dashboard, marks a noteworthy advancement toward fostering sustainability and operational efficiency. Leveraging the capabilities of IoT enables organizations to not just observe and streamline energy usage but also actively participate in global initiatives aimed at cultivating an environmentally aware future. Ongoing enhancements and advancements in both hardware and software elements will be pivotal to ensuring the enduring effectiveness of these systems within the dynamic landscape of technology..

1. Appendices

Arduino code:

```
// EmonLibrary examples openenergymonitor.org, Licence GNU GPL V3
#include "EmonLib.h"              // Include Emon Library
EnergyMonitor emon;              // Create an instance
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#define ON_Board_LED 2  //--> Defining an On Board LED, used for indicators when the process of
connecting to a wifi router
```

```
const char* ssid = "********"; //--> Your wifi name or SSID.
const char* password = "********"; //--> Your wifi password.
//----------------------------------------Host & httpsPort
const char* host = "script.google.com";
const int httpsPort = 443;
//----------------------------------------
WiFiClientSecure client; //--> Create a WiFiClientSecure object
String GAS_ID = "AKfycbzaloNi2OtJywzYhBU2pzdkh-JRDE0w_wkgmDVBZLKQLQ6sSLh9zN60APseuqLew_2AWw";
//--> spreadsheet script ID
// Variables for energy calculation
float kWh = 0.0;
unsigned long lastMillis = millis();
void setup()
{
  Serial.begin(115200);

  WiFi.begin(ssid, password); //--> Connect to your WiFi router
  Serial.println("");

  pinMode(ON_Board_LED,OUTPUT); //--> On Board LED port Direction output
  digitalWrite(ON_Board_LED, HIGH); //--> Turn off Led On Board

  //----------------------------------------Wait for connection
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    //----------------------------------------Make the On Board Flashing LED on the process of
connecting to the wifi router.
    digitalWrite(ON_Board_LED, LOW);
    delay(250);
    digitalWrite(ON_Board_LED, HIGH);
    delay(250);
    //----------------------------------------
  }
  //----------------------------------------
  digitalWrite(ON_Board_LED, HIGH); //--> Turn off the On Board LED when it is connected to the
wifi router.
  Serial.println("");
  Serial.print("Successfully connected to : ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  //----------------------------------------
  client.setInsecure();
  emon.voltage(D0, 137.57, 1.7);  // Voltage: input pin, calibration, phase_shift
  emon.current(D1, 0.0941);     // Current: input pin, calibration.
}
void loop()
{
  emon.calcVI(20, 400); // Calculate all. No.of half wavelengths (crossings), time-out

   //Calculate energy consumed in kWh
  unsigned long currentMillis = millis();
  kWh += emon.apparentPower * (currentMillis - lastMillis) / 3600000000.0;
  lastMillis = currentMillis;

  // Print data to Serial for debugging
  float v = emon.Vrms;
  float i = emon.Irms;
  float p = emon.apparentPower;
  Serial.printf("Vrms: %.2fV\tIrms: %.4fA\tPower: %.4fW\tkWh: %.5fkWh\n",
```

```
              v, i, p, kWh);
  sendData(v, i, p, kWh);
  delay(55200);
}

// Subroutine for sending data to Google Sheets
void sendData(float v, float i, float p, float kwh) {
  Serial.println("==========");
  Serial.print("connecting to ");
  Serial.println(host);

  //----------------------------------------Connect to Google host
  if (!client.connect(host, httpsPort)) {
    Serial.println("connection failed");
    return;
  }
  //----------------------------------------

  //----------------------------------------Processing data and sending data
  String string_v =  String(v, DEC);
  String string_i =  String(i, DEC);
  String string_p =  String(p, DEC);
  String string_kwh =  String(kwh, DEC);
  String url = "/macros/s/" + GAS_ID + "/exec?voltage=" + string_v + "&current=" + string_i +
"&power=" + string_p + "&kwh=" + string_kwh;
  Serial.print("requesting URL: ");
  Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
         "Host: " + host + "\r\n" +
         "User-Agent: BuildFailureDetectorESP8266\r\n" +
         "Connection: close\r\n\r\n");
  Serial.println("request sent");
  //----------------------------------------
  //----------------------------------------Checking whether the data was sent successfully or not
  while (client.connected()) {
    String line = client.readStringUntil('\n');
    if (line == "\r") {
      Serial.println("headers received");
      break;
    }
  }
  String line = client.readStringUntil('\n');
  if (line.startsWith("{\"state\":\"success\"")) {
    Serial.println("esp8266/Arduino CI successfull!");
  } else {
    Serial.println("esp8266/Arduino CI has failed");
  }
  Serial.print("reply was : ");
  Serial.println(line);
  Serial.println("closing connection");
  Serial.println("==========");
  Serial.println();
  //----------------------------------------
}
```

Google Apps Script code:

```
function doGet(e) {
Logger.log( JSON.stringify(e) );
var result = 'Ok';
if (e.parameter == 'undefined') {
```

```
result = 'No Parameters';
}
else {
var sheet_id = '12GAxTVI8JvMG_B71rtJl_eSdNZhgzzD35s_SWKSugIo'; // Spreadsheet ID
var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
var newRow = sheet.getLastRow() + 1;
var rowData = [];
var Curr_Date = new Date();
rowData[0] = Curr_Date; // Date in column A
var Curr_Time = Utilities.formatDate(Curr_Date, "Asia/Dhaka", 'HH:mm:ss');
rowData[1] = Curr_Time; // Time in column B
for (var param in e.parameter) {
Logger.log('In for loop, param=' + param);
var value = stripQuotes(e.parameter[param]);
Logger.log(param + ':' + e.parameter[param]);
switch (param) {
case 'voltage':
rowData[2] = value; // Voltage in column C
result = 'Voltage Written on column C';
break;
case 'current':
rowData[3] = value; // Current in column D
result += ' ,Current Written on column D';
break;
case 'power':
rowData[4] = value; // Power in column E
result += ' ,Power Written on column E';
break;
case 'kwh':
rowData[5] = value; // kWh in column F
result += ' ,kWh Written on column F';
break;
default:
result = "unsupported parameter";
}
}
Logger.log(JSON.stringify(rowData));
var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
newRange.setValues([rowData]);
}
return ContentService.createTextOutput(result);
}
function stripQuotes( value ) {
return value.replace(/^["']|['"]$/g, "");
}
```

Datasheet link : powerConsumption - Google Sheets

Explanation of issues to be considered:

Challenges and Resolutions:

1. Planning and Research:

We conducted thorough research to ensure the compatibility of purchased devices with the electronic devices from which we collected data.

2. Wattage:
The TV's wattage is 102 W, while the DTH ranges between 8 to 25 watts.

3. Safety:
Safety was paramount, leading us to encase our devices in a wooden box to prioritize security.

4. API Integration:
The API integration process was relatively smooth, with minimal issues encountered.

5. User Interface and Experience (UI/UX):
The dashboard lacked a kWh sector, representing an area for potential improvement.

6. User Manual:
Currently, there is no user manual on the interface; however, we plan to incorporate it in the future for enhanced user guidance.

7. Future Work and Limitations:
Future enhancements could involve monitoring specific devices to gauge their individual energy consumption. Additionally, adding a user manual to the web interface is a priority for improved user understanding.
8. Real-time Data or Stored Data:

Our system acquires real-time data, which is subsequently stored, and users have the option to download data in CSV format.

9. Reliability:
The data is reliable over 95% of the time, with occasional instances of incongruent values. Continuous improvements aim to enhance overall reliability.

Checklist of Issues (see explanations here)	**Remarks (if any)**
✅ *Planning and researching?	
✅ Data Collected?	
✅ Data Stored?	
✅ Data Displayed in the dashboard?	
✅ *Realtime or stored data?	
✅ *Wattage of the chosen device?	
✅ *Wattage of the measuring equipment?	

✅ *AC Power: Why/not apparent power? Why/not Instantaneous Power? AC power vs. DC power?	AC power was converted to DC by laptops battery
✅ *Documentation	
✅ *Safety: Electrical Insulation and Isolation?	
✅ *Caution with overclocking/flushing	
✅ *API Issues: Did you get it? How? If not, how did you solve this problem?	We used google sheet api which was free and available
✅ *UI/UX issues? Standard components of an energy dashboard?	No issues
✅ *User Manual?	IOT device was developed by us
✅ *Future Extensions and Limitations?	
✅ *Installation, Operation and Maintenance?	
✅ *Recurring costs	
✅ *Cost Accounting?	
*Business Aspects? Cost savings and ROI? Value of this product/service? Justification?	
✅ *Reliability? Never failed? Any fail-safe mechanisms?	
✅ *Accuracy? Calibration?	Accuracy was tested with multimeter
✅ *Data quality? ✅ Sampling rate? Crosstalk and interference? Accuracy and calibration?	
✅ *Scalability?	
✅ *Interoperability?	

✅ *Data Security? Important or not in this case?	
✅ *Compliant with regulations?	
✅ *Environmental Impacts? PESTLE analysis?	No Environmental Effects where detected