Application of Smart IoT Technology in Project Management Scenarios

Kumeel Rasheed
Dept. of Management Sciences
Bahria University
Islamabad, Pakistan
Kumeelrash161@gmail.com

Syed Ammad
Dept. of Civil Engineering
Monash University
Melbourne, Australia
Ammad.syed@monash.edu

Akhunzada Younis Said
Dept. of Computer Science
COMSATS University
Islamabad, Pakistan
Younus.said@gmail.com

Mohammed Balbehaith
Dept. of Business Management and Law
University of Portsmouth
Portsmouth, United Kingdom
Mohammed.Balbehaith@myport.ac.uk

Vipin Kumar Oad
Dept. of Civil and Envr. Engg
Gdańsk University of Technology
Gdańsk, Poland
Vipin.oad@pg.edu.pl

Aimal Khan
Dept. of Computer Science
COMSATS University
Islamabad, Pakistan
Akhan0178@gmail.com

***Abstract*— The report presents a low-cost and flexible solution to control and monitor home and industrial appliances. The popularity of automation devices has increased greatly in recent years due to higher affordability and simplicity through IoT phone and tablet connectivity. For this purpose, we have developed the IoT Outlet: a stand-alone communication unit, used to connect outlets to the internet. The IoT Plug is a power switch which can be accessed via Wi-Fi connection. Users can plug devices into the IoT Plug to remotely switch power on/off, get information of device's power consumption and local motion status, using wireless commands from the web and Android application. To access IoT Plug, it is connected with a router in Wi-Fi access point mode and control the Plug. An Android-based IoT phone application and Website having the control panels, employs standard operations such as Get and Post requests that return responses to communicate between the remote user and the Plug using PHP MySQL Databases. The main purpose of the paper is for IoT application to control and save power by setting the time scheduling through the controlling interface. Through this IoT device we can control and get feedback from any plugged device. There is a lot of need on large scale industries, laboratories, military fields and real-life situation, where one wants to know the status of the electric operated devices. This project is the building block for the application such as wireless controlling of any appliance that can be plugged in with local voltages i.e. 220V.**

***Index Terms*—IoT Plug, Wi-Fi, PHP My SQL, Phone.**

I. INTRODUCTION

The Internet of Things (IoT) is a networked ecosystem in which objects, animals, and humans are given unique IDs and the capacity to send data without the need for human-to-human or human-to-computer interaction. The Internet of Things (IoT) is a network of physical things, gadgets, automobiles, buildings, and other items that are equipped with electronics, software, sensors, and network connectivity to gather and share data [1]. The Internet of Things (IoT) may be used to make our household and industrial appliances IoT. It provides some fresh ideas and a lot of room for growth, all with the goal of increasing the quality of life and making it more pleasant and intelligent. A phone or tablet may be used to operate a range of network-connected devices, such as a IoT home system could monitor the energy consumption of household appliances [2]. IoT will have a wide-ranging influence on our work and lives if it is generally adopted, allowing people to accomplish things in a few easy and convenient ways while conserving energy and decreasing waste.

IoT Plug allows the user to create several and diverse timer schedules, such as turning lights on and off at different times throughout the day when you're not home. The IoT plug will also identify the power use of the electrical equipment attached to it, allowing the user to monitor and reduce energy consumption. Wireless IoT Plug eliminates wiring issues, allowing for more scalability and mobility. The IoT Plug application allows a person to operate several IoT devices at once with a single button press. The project is divided into two parts: hardware and software. IoT Plug will be an easy-to-use product. It may be used with any household equipment that runs on 220V alternating current. Once inserted into a regular outlet, the plug connects to the home router and can be accessed and managed via a Web-Android interface from anywhere at any time.

Following are features that IoT plug offers:

1) Control Function

When a household appliance is hooked in, the owner can use the power outlet to switch it on or off. The IoT Plug can operate a variety of household appliances and gadgets, including lights, air conditioners, televisions, and more.

2) Communication Function

To guarantee the regular operation of the IoT home system, the IoT socket needs communicate with the master controller to send data to it or reply to an order from the master controller. The master controller and the IoT socket form an effective communications system in which communication should be steady and quick.

3) Monitor Function

Users may check the status of Web servers and other devices, as well as voltage, current, and power usage. House owners can assess the electrical appliance load, which serves as a baseline for electrical usage, in order to decrease power consumption during peak hours. Users may save money and contribute to a cleaner environment by moving the bulk of their power consumption to off-peak hours.

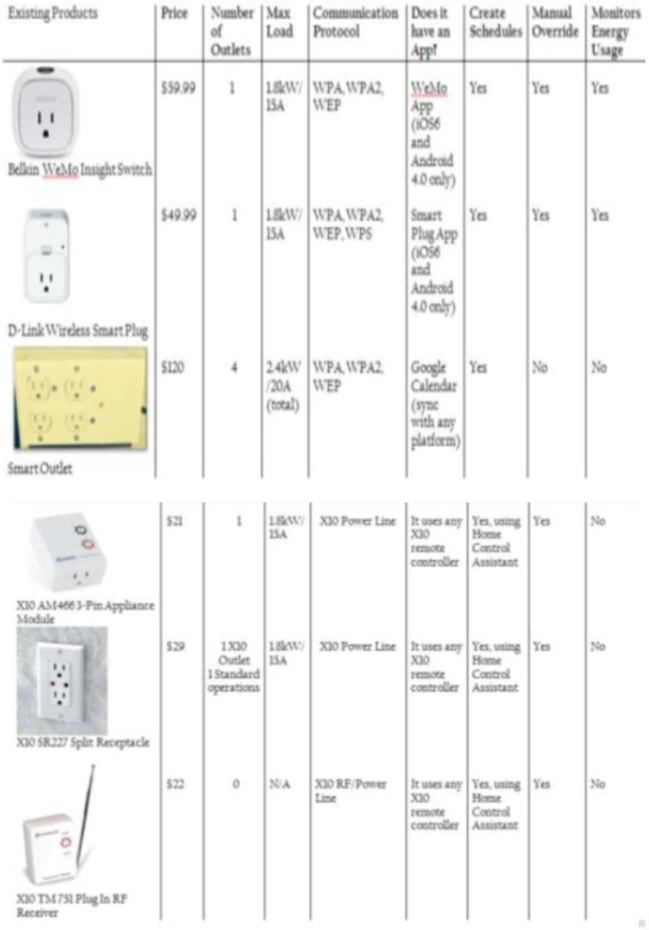
978-1-6654-9058-0/22/$31.00 ©2022 IEEE

4) Timing Function

The capability was primarily created to allow customers to turn on equipment such as electric water heaters and washing machines during off-peak hours, therefore conserving power. With a IoT home application, user can control the gadgets remotely and schedule them to turn off while no one is at home.

II. Literature Review

In addition to fast switching, automated plug-in modules allow users to set their gadgets on a schedule. These modules allow us to choose when we want our gadgets to turn on, avoiding the need for standby power and saving money on utilities. Standby power, often known as vampire power or phantom load, consumes energy that many people are powerless to control and costs them money [3]. When plugged in, it happens in practically every power source. The integration of IoT plug-in modules into consumers' homes and the elimination of vampire power is our secondary goal in putting gadgets on a timetable. Many of these gadgets are currently on the market and can be purchased through internet sellers or straight from home hardware stores like eBay. Their commonalities include a single input-to-output ratio, IoT phone compatibility, and Wi-Fi, 3G, or 4G controllability. However, each of these devices can only operate one appliance at a time and only work with their own calendar apps.



Existing Products	Price	Number of Outlets	Max Load	Communication Protocol	Does it have an App?	Create Schedules	Manual Override	Monitors Energy Usage
Belkin WeMo Insight Switch	$59.99	1	1.8kW/ 15A	WPA, WPA2, WEP	WeMo App (iOS6 and Android 4.0 only)	Yes	Yes	Yes
D-Link Wireless Smart Plug	$49.99	1	1.8kW/ 15A	WPA, WPA2, WEP, WPS	Smart Plug App (iOS6 and Android 4.0 only)	Yes	Yes	Yes
Smart Outlet	$120	4	2.4kW /20A (total)	WPA, WPA2, WEP	Google Calendar (sync with any platform)	Yes	No	No
X10 AM466 3-Pin Appliance Module	$21	1	1.8kW/ 15A	X10 Power Line	It uses any X10 remote controller	Yes, using Home Control Assistant	Yes	No
X10 SR227 Split Receptacle	$29	1 X10 Outlet 1 Standard operations	1.8kW/ 15A	X10 Power Line	It uses any X10 remote controller	Yes, using Home Control Assistant	Yes	No
X10 TM751 Plug In RF Receiver	$22	0	N/A	X10 RF/Power Line	It uses any X10 remote controller	Yes, using Home Control Assistant	Yes	No

Figure 1: Existing Products

Monitoring energy use and regulating appliances are two of the benefits of utilizing these devices. These devices are designed to be plug-and-play, requiring no difficult setup, so users with little understanding of home networking or electrical wiring may install them in their own houses. With Wi-Fi or cell service, most of these gadgets may be used at any time and in any location. They all have specific phone applications and a manual override option. Users of phones can receive notifications directly on their phones. This is a great method to keep track of user's home state. Different wireless communication strategies are used by these devices. A few of them are listed below.

Table 1: Wireless Connectivity Techniques

Protocol	Bluetooth	Zigbee	Wi-Fi 802.11
Data Rate	1 Mbit/s	20,40,250 Kbit/s	1154 Mbit/s
Range	10 m	10 ~ 100 m	>100
Network Topology	Ad-hoc, small systems	Ad-hoc, peer to peer, star	Point to hub
Frequency	2.4 GHz	2.4 GHz	2.4 and 5 GHz
Power Consumption	Low	Very Low	High
Applications	Inter-device wireless connectivity	Industrial control and monitoring, sensor networks	WLAN Connectivity

Bluetooth modules have a shorter range and are more expensive, they are not recommended [4]. Furthermore, the data speeds are really poor. On the other side, we can use the ZigBee module to solve the aforesaid issues, however ZigBee modules are prohibitively expensive.

III. SPECIFICATION

To design and make the IoT plug functional, both hardware and software components are employed. The required hardware and software specifications are mentioned below.

A. Hardware Components

The hardware components will be the most important for the IoT plug to function properly. To ensure appropriate operation, the components will be interfaced with one another and with the program.

1) *ACS 712 Current Sensor:* The module used for the purpose of measuring the consumption of related or plugged appliance is ACS 712 as shown in Figure 2. The ACS 712 sensor is compact and suited for usage in IoT plugs [5]. A signal conditioning circuit is also connected to the Hall Effect sensor in order to condition and smooth the signal so that the controller can easily measure the AC waves. Motor control, load detection and management, switched-mode power supply, and overcurrent fault prevention are examples of common uses. The gadget is made up of a linear Hall sensor circuit with a copper conduction route near the surface that is precise and low-offset. The magnetic field generated by the applied current flowing via this copper conduction line is measured by the integrated Hall IC and transformed into a proportional voltage.

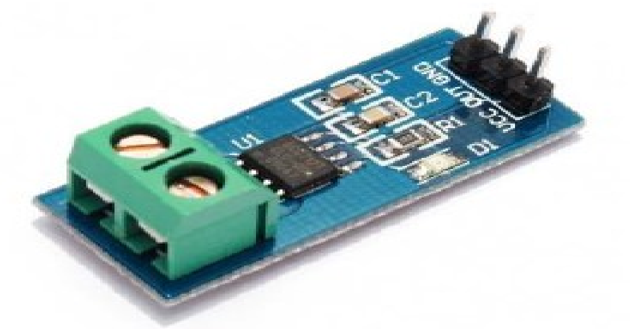


Figure 2: ACS Current Sensor

2) *Relay Module:* A relay is a switch that is controlled by electricity. Many relays employ an electromagnet to mechanically activate a switch, however solid-state relays and other working principles are also used [6]. For the IoT plug, we utilize an SPDT relay to switch on/off the connector using a microcontroller input as shown in Figure 3.

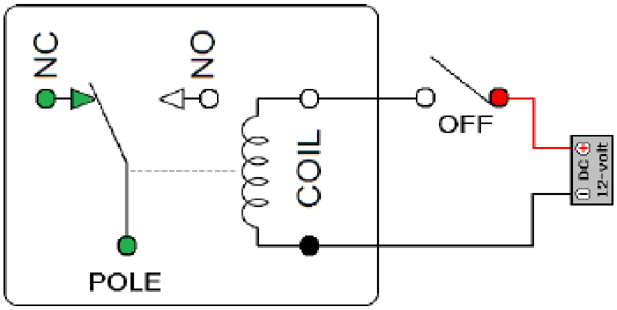


Figure 3: Relay Circuit Diagram

3) *Wi-Fi Module:* The ESP8266 Wi-Fi module will be utilized to link the IoT plug to the internet and local network. The ESP8266 is a highly integrated chip created to meet the demands of a connected world. It provides a comprehensive and self-contained Wi-Fi networking solution, allowing it to host programs or offload all Wi-Fi networking activities to another CPU [7]. The ESP8266 features robust on-board processing and storage capabilities, allowing it to be utilized with sensors and other application-specific devices via its GPIOs with minimum development and runtime loading. The complete solution, including the module, is designed to take up small on the PCB as illustrated in Figure 4.

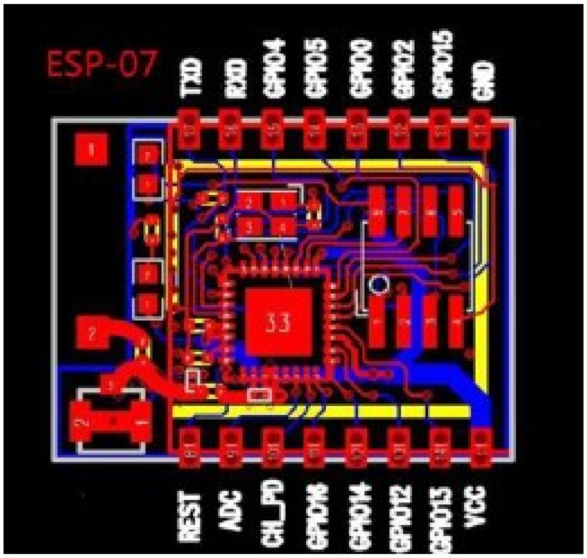


Figure 4: ESP8266 Wi-Fi Module Circuit Diagram

4) *Arduino Module:* Arduino is a programming language that allows you to create computers that can perceive and control more of the physical environment than your typical desktop computer [8]. Arduino is a tiny circuit board that contains an Atmel Microcontroller chip as well as additional components shown in Figure 5. Arduino may be used to quickly build Interactive Objects that take input from a number of switches or sensors, make decisions, and operate a range of outputs including lights, motors, and noises.

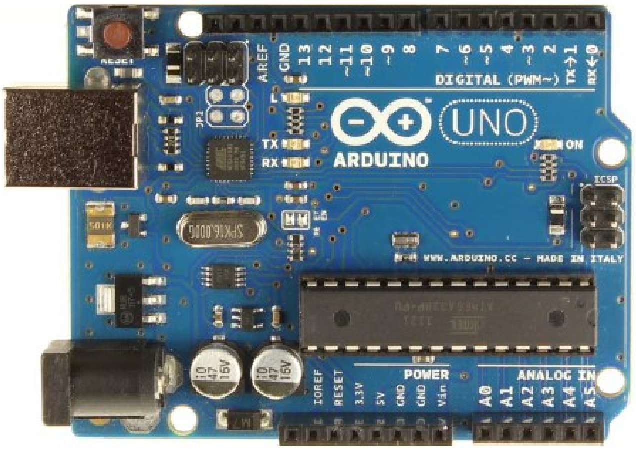
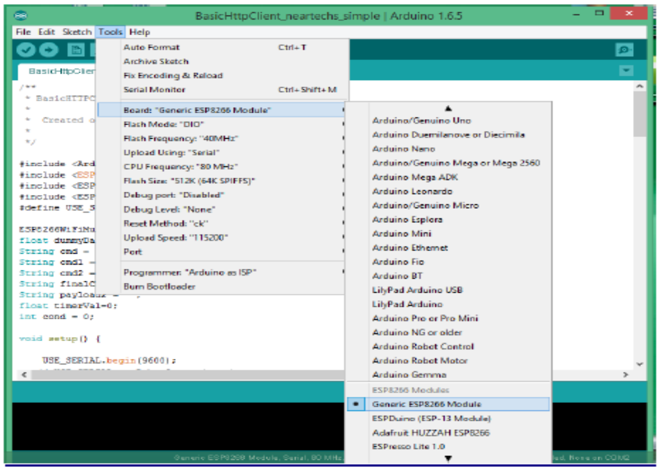


Figure 5: Arduino Uno

B. Software Components

Different sorts of software have been developed to allow for the easy interfacing of many algorithms at the same time. The software is utilized for coding and software-to-hardware interface, which will finally conduct the IoT plug operation.

1) *Arduino 1.6.5*: Arduino 1.6.5 is an IDE (Integrated Development Environment) that tends to make writing code and uploading it to the board very simple. Windows, Mac OS X, and Linux are all supported. The environment is written in Java and uses open-source software like Processing. 1.6.5 is the version that is used. Because the implementation works with both Arduino UNO and ESP, it was originally used for programming Arduino and running esp8266 in AT command mode. It is primarily used for coding the Arduino so that it could transmit AT commands to the ESP module that could be performed by the ESP [8]. Also, we have used Arduino as a TTL converter for the ESP, since every device needs a USB to TTL converter so that it can communicate with the computer as shown in Figure 6.



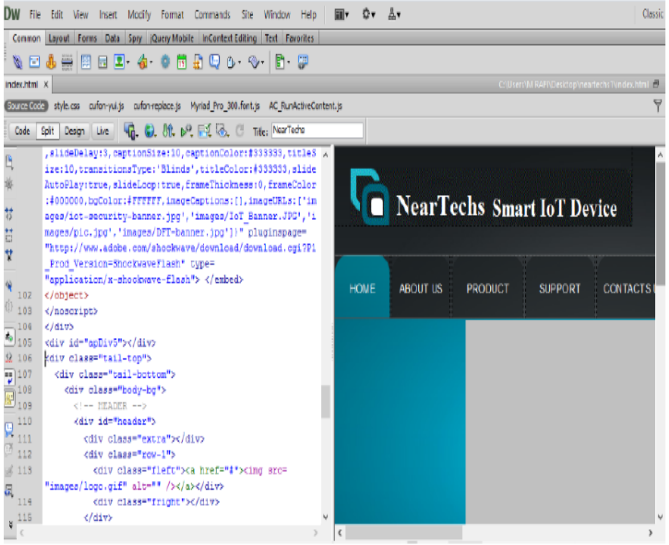
Board: "Generic ESP8266 Module"
Flash Mode: "DIO"
Flash Frequency: "40MHz"
Upload Using: "Serial"
CPU Frequency: "80 MHz"
Flash Size: "512K (64K SPIFFS)"
Debug port: "Disabled"
Debug Level: "None"
Reset Method: "ck"
Upload Speed: "115200"
Port
Programmer: "Arduino as ISP"

```
/**
 * BasicHTTPC
 *
 *  Created o
 *
 */

#include <Ard
#include <ESP
#include <ESP
#include <ESP
#define USE_S

ESP8266WiFiMu
float dummyDa
String cmd =
String cmd1 =
String cmd2 =
String finalC
String payload
float timerVal=0;
int cond = 0;

void setup() {

    USE_SERIAL.begin(9600);
```

Generic ESP8266 Module, Serial, 80 MHz ... ed, None on COM2

Figure 6: Arduino 1.6.5 IDE

2) Adobe Dreamweaver CS 6: Dreamweaver is a high-level language and a sort of text editor used for web creation. It enables the user to visit the site using a locally installed browser and make any necessary adjustments. Dreamweaver, unlike any other program, includes several built-in capabilities such as a built-in debugger, code aid, remote website, free templates and split view. All parts of website maintenance become easier and more trustworthy with Adobe Dreamweaver.


DW
File Edit View Insert Modify Format Commands Site Window Help
Classic
Common Layout Forms Data Spry jQuery Mobile InContext Editing Text Favorites
index.html
Source Code style.css cufon-yui.js cufon-replace.js Myriad_Pro_300.font.js AC_RunActiveContent.js
Code Split Design Live
Title: NearTechs
NearTechs Smart IoT Device
HOME ABOUT US PRODUCT SUPPORT CONTACTS

```
,slideDelay:3,captionSize:10,captionColor:#333333,titleS
ize:10,transitionsType:'Blinds',titleColor:#333333,slide
AutoPlay:true,slideLoop:true,frameThickness:0,frameColor
:#000000,bgColor:#FFFFFF,imageCaptions:[],imageURLs:['im
ages/iot-security-banner.jpg','images/IoT_Banner.JPG','i
mages/pic.jpg','images/DFT-banner.jpg']}" pluginspage=
"http://www.adobe.com/shockwave/download/download.cgi?P1
_Prod_Version=ShockwaveFlash" type=
"application/x-shockwave-flash"> </embed>
</object>
</noscript>
</div>
<div id="apDiv5"></div>
<div class="tail-top">
  <div class="tail-bottom">
    <div class="body-bg">
      <!-- HEADER -->
      <div id="header">
        <div class="extra"></div>
        <div class="row-1">
          <div class="fleft"><a href="#"><img src=
"images/logo.gif" alt="" /></a></div>
          <div class="fright"></div>
        </div>
```

Figure 7: Adobe Dream Viewer

3) Android Studio*:* Android Studio is an all-in-one programming environment for the Android Platform. Java is the platform on which Android apps are built. Android Studio is a simple to learn development environment. The Android Studio was used to create the android app that we use to operate the device via the internet using phones. To create the app, we employed a variety of languages such as JAVA, XML and others, all of which were compiled and run in the Android Studio.

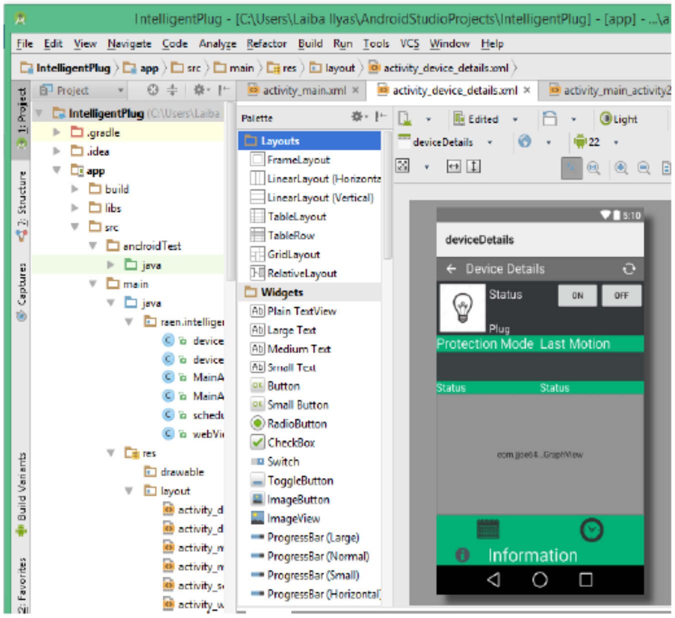


Figure 8: Android Studio

IV. SYSTEM DESIGN

In a broader sense, the system consists of the Internet, the Remote, and the Plug. The physical process is often tracked and managed by integrated computers and networks using a feedback mechanism, which influences the calculations of the physical process of IoT devices [9]. The plug is connected to the internet via the router, which is internet enabled and is a standard internet router used in a basic home for internet connectivity provided by the ISP. The plug connects to the router and accesses the internet, where it is connected to the data cloud of www.neartechs.com where user can login to their account which they had registered already on the website before getting the device, navigate to the dashboard and use the dashboard to update the control data such as On, OFF, Timer ON in the remote cloud database as shown in Figure 8. It always reads the most up-to-date information from the cloud database. On the other side, there are distant users who are linked to the internet via a mobile data connection such as 3G, 4G, or even 2G, either through the same or a separate router. These distant users are the originators of the information stored in the cloud database. The plug receives the current data from the database and acts appropriately, while the remote user updates the control data in the database.

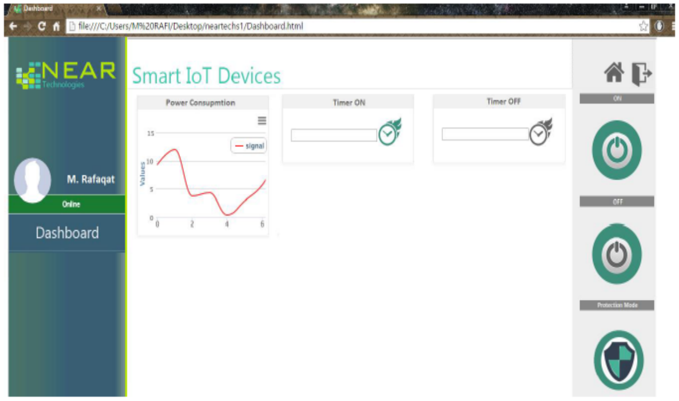


Figure 9: Website Control Panel

The android application, which can be installed on the phone and thus have all of the basic functionality of the plug, is another way to post data to the cloud database or to basically control the device, illustrated in Figure 10. Whenever a button is clicked, it updates or inserts some data into the database, which is then retrieved by the plug and the plug acts accordingly.

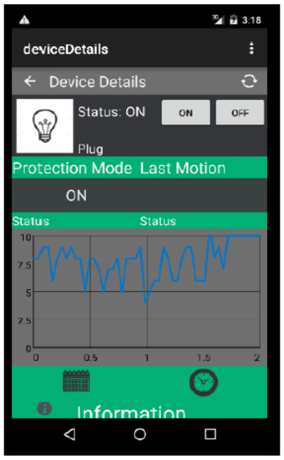
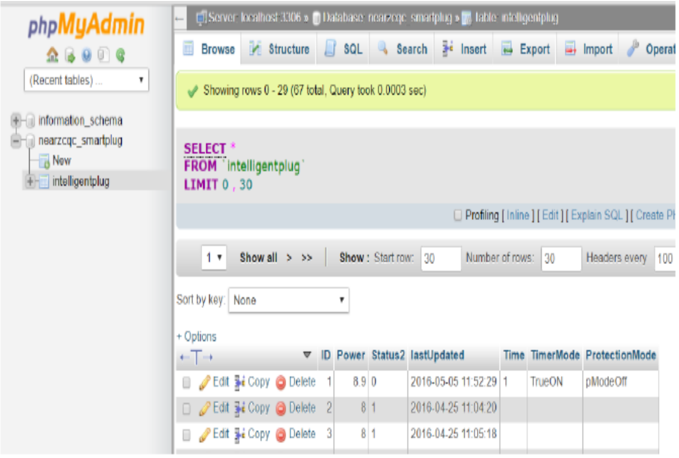


Figure 10: Android Application Software

The database is the other key component, and it is used to store all of the data and information that has to be transmitted between the plug website and the android application shown in Figure 11. The hardware assembly, which comprises of multiple devices joined together to provide a certain desired functionality, is the final and most significant aspect of the project. It consists of Power Measuring devices, switching devices, Wi-Fi connection devices and power supply devices.



Showing rows 0 - 29 (67 total, Query took 0.0003 sec)

```
SELECT *
FROM `intelligentplug`
LIMIT 0 , 30
```

Show : Start row: 30 Number of rows: 30 Headers every 100

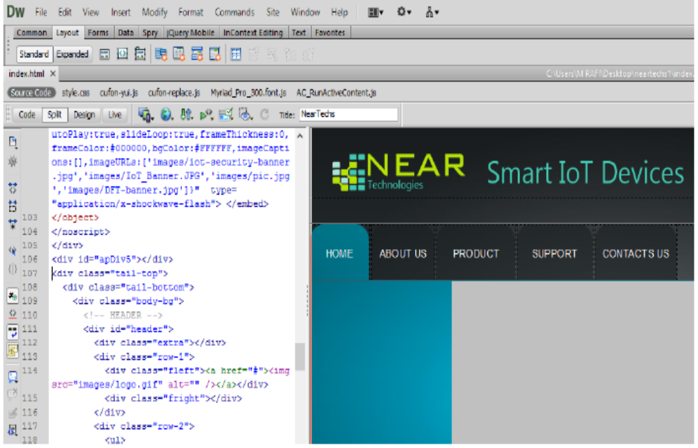
Sort by key: None

	ID	Power	Status2	lastUpdated	Time	TimerMode	ProtectionMode
Edit Copy Delete	1	8.9	0	2016-05-05 11:52:29	1	TrueON	pModeOff
Edit Copy Delete	2	8	1	2016-04-25 11:04:20			
Edit Copy Delete	3	8	1	2016-04-25 11:05:18			

Figure 11: Online Cloud Database

A database server, such as MySQL, is a software that can store vast quantities of data in an orderly fashion that can be retrieved using queries and other languages such as PHP. You may instruct PHP to search the database for a list of Power Consumption that can be used to create a graph or serve another function on the site. In the project, all data from the Plug, the app, and the website is saved exclusively in the database, and the website, app, and hardware are all built to get data from the database and do various actions on that basis. The database stores all of the data exchanged between the Android application, the Website, and the Plug. The database has seven columns: ID is the main key that identifies each new row entered, and Power holds the data logged by the ESP. Following the current power value and the insert, the ESP will query the API. The PHP program will be run, and the data will be inserted into the Power column through a database insert. The database simple comprises of Insert, Update and Select Queries which are execute through the PHP MySQL.

Adobe Dreamweaver is utilized for all tasks and operations related to website layout and interface. First of all, type div is selected and picture of our logo is inserted in that specific div. In another div, title menu is inserted as shown on the front page of the web in Figure 12. Different types of pages are developed based on the menu bar, and these pages are then connected and allocated to the appropriate locations in the menu bar. The main goal is to connect the necessary sites to certain menu bar icons and link them together.


DW File Edit View Insert Modify Format Commands Site Window Help
Common Layout Forms Data Spry jQuery Mobile InContext Editing Text Favorites
Standard Expanded
index.html
C:\Users\M RAFI\Desktop\neartechs1\index.
Source Code style.css cufon-yui.js cufon-replace.js Myriad_Pro_300.font.js AC_RunActiveContent.js
Code Split Design Live
Title: NearTechs
NEAR Technologies
Smart IoT Devices
HOME
ABOUT US
PRODUCT
SUPPORT
CONTACTS US

```
utoPlay:true,slideLoop:true,frameThickness:0,
frameColor:#000000,bgColor:#FFFFFF,imageCapti
ons:[],imageURLs:['images/iot-security-banner
.jpg','images/IoT_Banner.JPG','images/pic.jpg
','images/DFT-banner.jpg']}"  type=
"application/x-shockwave-flash"> </embed>
</object>
</noscript>
</div>
<div id="apDiv5"></div>
<div class="tail-top">
  <div class="tail-bottom">
    <div class="body-bg">
      <!-- HEADER -->
      <div id="header">
        <div class="extra"></div>
        <div class="row-1">
          <div class="fleft"><a href="#"><img
src="images/logo.gif" alt="" /></a></div>
          <div class="fright"></div>
        </div>
        <div class="row-2">
          <ul>
```

Figure 12: GUI Website

The primary login screen webpage via which users may access the dashboard, which contains the module's control functionality as shown in Figure 13. This is the dashboard's security access protocol; if the user enters the required login and password, they will be granted access; otherwise, they will not.

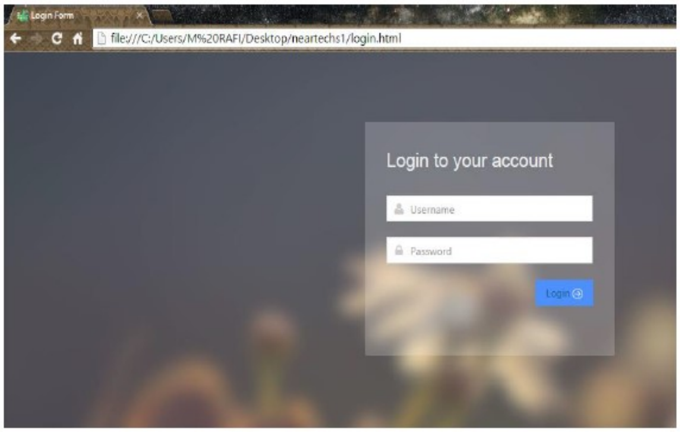


Figure 13: Login Screen

The android application, which will be loaded on a distant user's mobile phone and will be able to manage the plug remotely through the internet, will be described in full. Figure 14 depicts the application's main activity, which includes the primary log in page, from which the user may log in and use the plug's functionality. This code validates whether the username and password given are accurate or wrong.

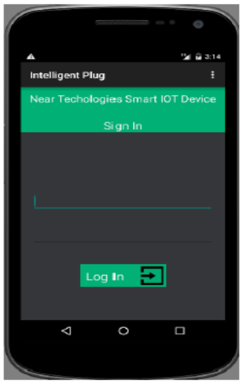
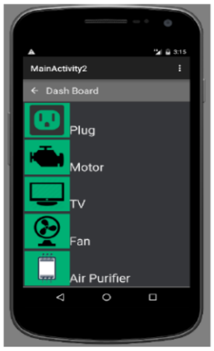


Figure 14: Login Panel Android App

The second activity of the application is the device list, which contains all of the devices that have been registered with a certain user. By selecting any of the device icons, we may browse to the device's control panel, from where we can control the plug's functionality. After going to the control panel activities, we can control the plug using several functions such as protection mode, which activates the motion sensor that is attached to the plug and notifies the user if there is any motion when the protection mode is on as shown in Figure 15.



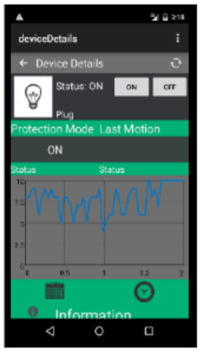
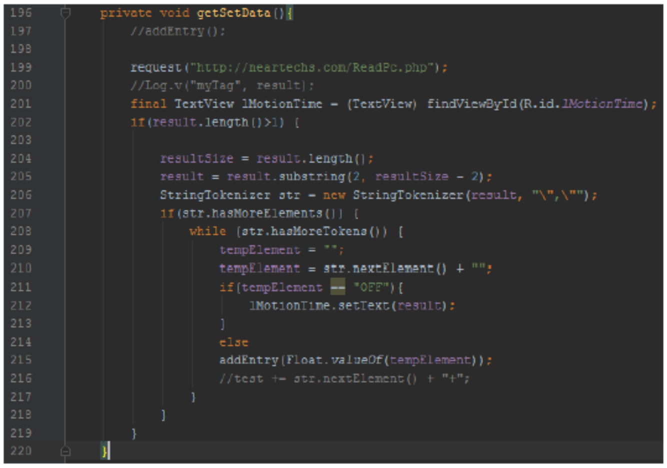


Figure 15: Device Activity List

First, we initialize certain variables in the control panel activity's JAVA file, which will be utilized in subsequent functions. When the button is clicked, the request function accesses the ON OFF API, which sends a request to the server, which then transmits the Database query created by the On.

PHP file to the database, which updates or inserts new status information in the database as shown in Figure 16. In Android, we utilize the GraphView library to generate the graph. We create a GraphView object and then specify some of the fundamental methods that are required. Finally, the request function, which is the primary method of accessing a website's API, establishes an HTTP connection, connects to the server, and passes the data in the API to the server; after the server responds to the application, the responded data is saved to the Buffered Reader, where it is stored as a string in Java Object Notation Format in the result variable.



```
private void getSetData(){
    //addEntry();

    request("http://neartechs.com/ReadPc.php");
    //Log.v("myTag", result);
    final TextView lMotionTime = (TextView) findViewById(R.id.lMotionTime);
    if(result.length()>1) {

        resultSize = result.length();
        result = result.substring(2, resultSize - 2);
        StringTokenizer str = new StringTokenizer(result, "\",\"");
        if(str.hasMoreElements()) {
            while (str.hasMoreTokens()) {
                tempElement = "";
                tempElement = str.nextElement() + "";
                if(tempElement == "OFF"){
                    lMotionTime.setText(result);
                }
                else
                addEntry(Float.valueOf(tempElement));
                //test += str.nextElement() + "+";
            }
        }
    }
}
```

Figure 16: Code

V. IMPLEMENTATION

The system's comprehensive operation is described, including how it works and what kind of queries the plug sends to the server, as well as how the server reacts to them, shown in Figure 17. The most crucial factor that allows the system to work is the command flow. The command flow begins with the remote user, which comprises of a website and an Android application. The android application changes data in the database and serves as a command source, generating commands. It also serves as a source of information for the plug, displaying a graphical depiction of the power used [10].

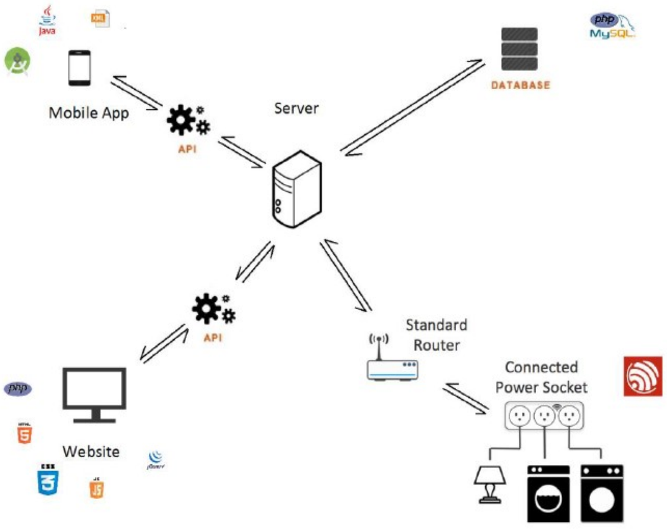


Figure 17: System Overview

Since the ESP accesses the Read.php API to retrieve data from the database every few time frames, when the server receives a request from the ESP, the server executes the request, and because it's a query to get current data from the database, the request is forwarded to the Database System, which executes the query, and the database returns the desired results to the server, which are then responded to the ESP. If the request was successful, the ESP will return 200, and then it will retrieve the payload data and turn on or off the module based on the data. For example, if the payload is ONOFF:1, the ESP will turn on relay pin 12 and so the device linked to the pug will turn on.

Figure 18 depicts the inside construction of the plug's Hardware. The power source supplies electricity to all of the plug's components, while the connection devices offer a means of communication with the internet, from which the data is received, and the plug switching mechanism is regulated based on the data. While the plug is performing all of this, the current measuring sensor receives the current value from the sensor and conducts certain mathematical operations on it before sending it to the server, which records it in the database by sending the data and query to the database system to execute.

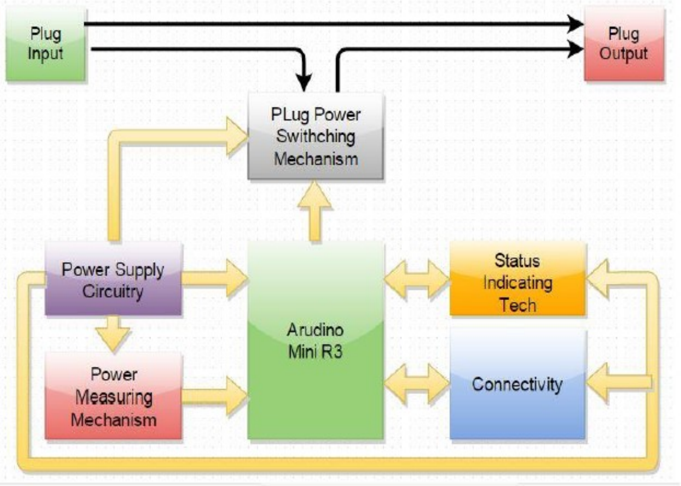
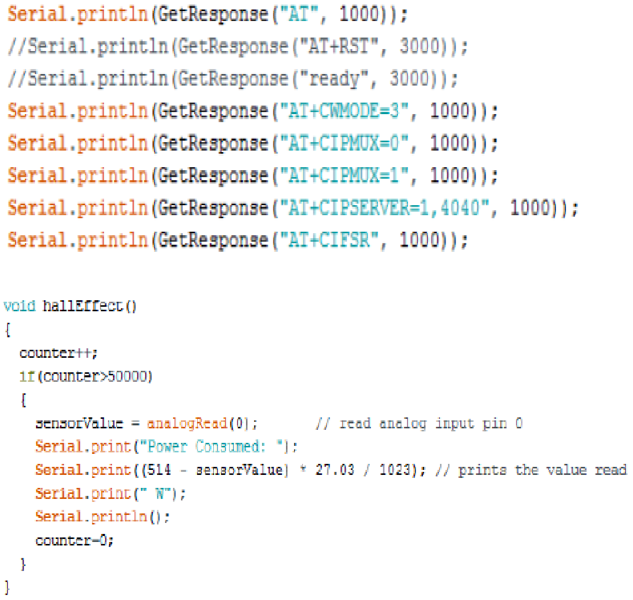


Figure 18: Hardware Architecture

VI. TESTING & EVALUATION

The first Arduino code we used to interact with the ESP8266 module is shown in Figure 19, for the first effort on the Arduino interfaced with the ESP8266 module, and communicated over the serial interface. First and foremost, we created a software serial interface for connecting with the esp module through the serial port and serial monitor. In addition, several variables are set up to hold data from the current sensor as well as the timer value received from the server. When the module is in server mode and there is a client request to the server, we check the serial for any data from the client, and if the data is the data from the button, we read it and write it to the variable server Data.

We adjust the pins accordingly after receiving the value from the server. And it collects data from the power module and displays it on the Arduino's serial monitor by calling the hallEffect method [11]. Basically, this function sends commands to the ESP module, then receives the ESP's answer and prints it to the serial monitor. When the ESP module is working in server mode then the JQUERY GET command is used sent data to the ESP server and where it is shown in the ESP serial [12]. The response is generated by the ESP server and send to the Arduino through the ESP serial. In the ESP serial, the Arduino searches for the string IPD.



```
Serial.println(GetResponse("AT", 1000));
//Serial.println(GetResponse("AT+RST", 3000));
//Serial.println(GetResponse("ready", 3000));
Serial.println(GetResponse("AT+CWMODE=3", 1000));
Serial.println(GetResponse("AT+CIPMUX=0", 1000));
Serial.println(GetResponse("AT+CIPMUX=1", 1000));
Serial.println(GetResponse("AT+CIPSERVER=1,4040", 1000));
Serial.println(GetResponse("AT+CIFSR", 1000));


void hallEffect()
{
  counter++;
  if(counter>50000)
  {
    sensorValue = analogRead(0);       // read analog input pin 0
    Serial.print("Power Consumed: ");
    Serial.print((514 - sensorValue) * 27.03 / 1023); // prints the value read
    Serial.print(" W");
    Serial.println();
    counter=0;
  }
}
```

Figure 19: Execution Code

In the second attempt, we changed the website's structure by including timer functionality, which includes the timer value and mode selection. After successfully operating the IoT plug in server and local area mode, we went on to making it operate over the internet. To do so, we employed a variety of platforms, the first of which was ThingSpeak, which is essentially an IoT platform [13], shown in Figure 20. We have added all of the variables that are being utilized, and we've also created the ESP8266 WiFi's object, sensor value, which will contain the data that is read from the current sensor. We construct an HTTP client and establish a connection with the server. Once the connection is established, the API key is used to gain access to the field data that will be sent to the database. When the ESP connects to the API, it obtains the power value from the ADC pin [14] and sends it to the ThingSpeak database, where it is presented in a graph and readily seen. The data can also be exported to an excel file and utilized for many applications.

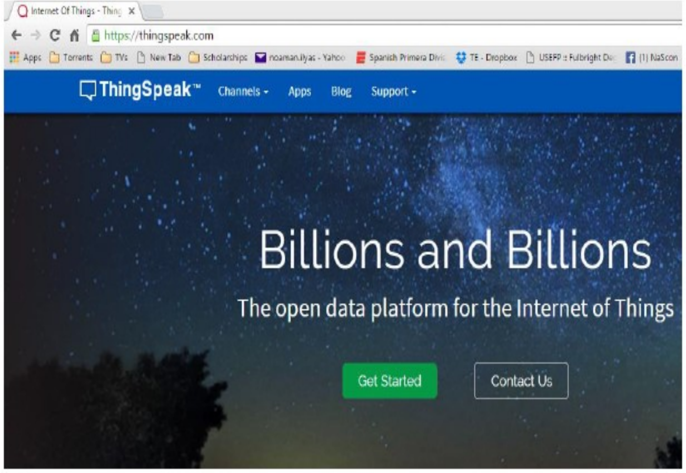


Figure 20: ThingSpeak Website

VII. CONCLUSION

In conclusion, an IoT device is demonstrated that is utilize to manage and control various appliances in the house and in industries. It has the potential to be a fantastic automation solution. Although there are many alternative IoT plug options, it is considerably superior owing to its plug and play, low cost, timer, and graph display characteristics. As the IoT plug can be operated wirelessly over the internet, we can use it to link any appliance to the Internet of Things. The senses with which we interact with technology will diversify and develop in the next decades, and IoT Plug will be there to help the transition. It's a fully wire-free solution enabling immersive, hands-on engagement with a workstation. The IoT Plug has proved to be a very useful tool for enabling true IoT connection. We believe the IoT Plug technology will benefit and enable the development of various IoT interface applications due to its compact size, high precision, and low purchasing cost.

ACKNOWLEDGMENT

The study praises the dedication of all of the contributing researchers in completion of this paper.

REFERENCES

[1] A. H. Mohammed, R. M. KHALEEFAH, and I. A. Abdulateef, “A review software defined networking for internet of things,” in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–8.

[2] K. Rasheed, S. Saad, L. Shahzad, S. Ammad, A. Ali, and I. Badshah, “Application of Gesture Data Recognition in a Human-Interactive Leap Motion Sensor Chair,” in *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, 2021, pp. 567–571. doi: 10.1109/ICDABI53623.2021.9655819.

[3] E. Mariyappan and C. Balakrishnan, “Power draining prevention in Ad-Hoc Sensor networks using sensor network encryption protocol,” in *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 2014, pp. 1–5.

[4] A. S. Alon and J. A. B. Susa, “Wireless Hand Gesture Recognition for an Automatic Fan Speed Control System: Rule-Based Approach,” in *2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, 2020, pp. 250–254.

[5] U. Khair, A. J. Lubis, I. Agustha, and M. Zulfin, “Modeling and simulation of electrical prevenion system using Arduino Uno, GSM modem, and ACS712 current sensor,” in *Journal of Physics: Conference Series*, 2017, vol. 930, no. 1, p. 12049.

[6] W. S. Alaloul, A. H. Qureshi, M. A. Musarat, and S. Saad, “Evolution of close-range detection and data acquisition technologies towards automation in construction progress monitoring,” *J. Build. Eng.*, vol. 43, no. June, p. 102877, 2021, doi: 10.1016/j.jobe.2021.102877.

[7] P. Srivastava, M. Bajaj, and A. S. Rana, “Overview of ESP8266 Wi-Fi module based smart irrigation system using IOT,” in *2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 2018, pp. 1–5.

[8] K. Rasheed, L. Shahzad, S. Saad, H. A. Khan, W. Ahmed, and T. Sadiq, “Parking Guidance System Using Wireless Sensor Networks,” in *2021 International Conference on Decision Aid Sciences and Application (DASA)*, 2021, pp. 573–577. doi: 10.1109/DASA53625.2021.9682213.

[9] S. Saad, W. S. Alaloul, K. Rasheed, and S. Ammad, “Modelling and Simulation of Construction Cyber-Physical Systems,” *Cyber-Physical Syst. Constr. Sect.*, no. April, pp. 88–110, 2022, doi: 10.1201/9781003190134-6.

[10] S. Saad, W. S. Alaloul, N. Shafiq, S. Ammad, A. H. Qureshi, and A. A. M. Mohammed, “Visually Programming Automated Slab Positioning Tool (ASPT) using Evolutionary Solvers,” in *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, 2021, pp. 527–530. doi: 10.1109/ICDABI53623.2021.9655914.

[11] D. H. Petersen, O. Hansen, R. Lin, and P. F. Nielsen, “Micro-four-point probe Hall effect measurement method,” *J. Appl. Phys.*, vol. 104, no. 1, p. 13710, 2008.

[12] B. Kotiyal and M. Muzamil, “Home automation using arduino WiFi module ESP8266,” 2016.

[13] D. Parida, A. Behera, J. K. Naik, S. Pattanaik, and R. S. Nanda, “Real-time environment monitoring system using ESP8266 and ThingSpeak on internet of things platform,” in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 225–229.

[14] R. I. S. Pereira, S. C. S. Jucá, and P. C. M. Carvalho, “IoT embedded systems network and sensors signal conditioning applied to decentralized photovoltaic plants,” *Measurement*, vol. 142, pp. 195–212, 2019.