

POWERSYNC: OPTIMIZING ENERGY MONITORING THROUGH ARDUINO AND RASPBERRY PI SERVER BASED CONTROL SYSTEM

Euder Wayne P. Dugay, Gabrielle C. Rivera, Lloyd Vince Angelo B. Tadeo, and Kenosis John B. Wajchina, RCpE

ABSTRACT

This study focused on developing an integrated energy monitoring and control system consisting of a Windows application and a private website for the TTBD0. The Windows application was designed to monitor and control electrical loads within the facility, while the private website displays statistical reports of energy consumption over specific time periods. The project incorporated an IoT system using Arduino and Raspberry Pi, along with a scale model of the TTBD0 for demonstration purposes. The primary objective was to create an integrated system that connects hardware components to software controls via LAN communication protocols, providing users with real-time status updates on energy consumption via the website. The proposed solution aimed to effectively monitor and control active loads at a miniature scale within the TTBD0, while tracking and displaying energy consumption data.

Keywords: Windows App, Website, Energy Monitoring and Control, Raspberry Pi, Arduino

INTRODUCTION

Electricity is an essential but expensive resource. As technology and infrastructure continue to develop, energy demand is rapidly increasing. To ensure sustainable use, it is crucial to implement systems that monitor, control, and optimize energy consumption. One of the largest energy consumers in buildings is the HVAC (heating, ventilation, and air conditioning) system, which can account for 40–70% of energy use in commercial spaces. Effective monitoring helps optimize HVAC performance, lowering energy costs and improving operational efficiency (DEE, 2013).

Globally, it's estimated that 66% of generated energy is wasted, primarily due to inefficient electrical devices, outdated insulation, and poorly optimized industrial systems. This waste significantly contributes to greenhouse gas emissions. Reducing this inefficiency can lead to substantial financial and environmental benefits (Ukpanah, 2024).

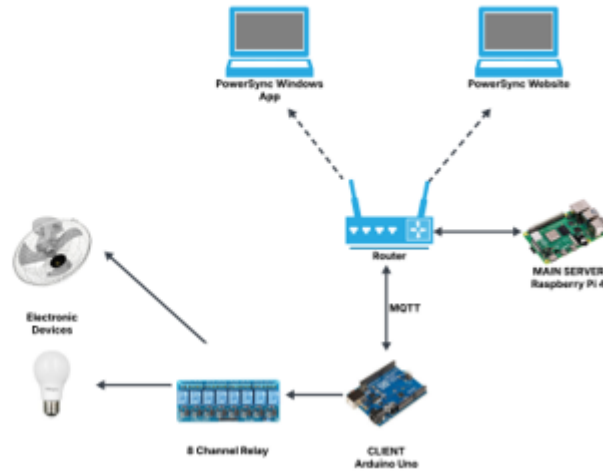
In the Philippines, rising electricity rates further emphasize the need for energy-saving measures. As of September 2024, Meralco increased its electricity rate by ₱0.1543 per kWh due to higher transmission costs (GMA News Online, 2024).

ISO 50001 is an international standard that provides a framework for developing energy policies, setting measurable goals, and improving energy performance through data-driven decisions. It promotes continuous improvement in energy efficiency and is applicable across industries (ISO, 2023).

A study by Akbar et al. (2023) showcased a web-based electricity monitoring system implemented in a university using a waterfall development approach. The system followed ISO 50001 principles and highlighted the benefits of real-time energy data for managing consumption and reducing costs.

Conceptual Framework

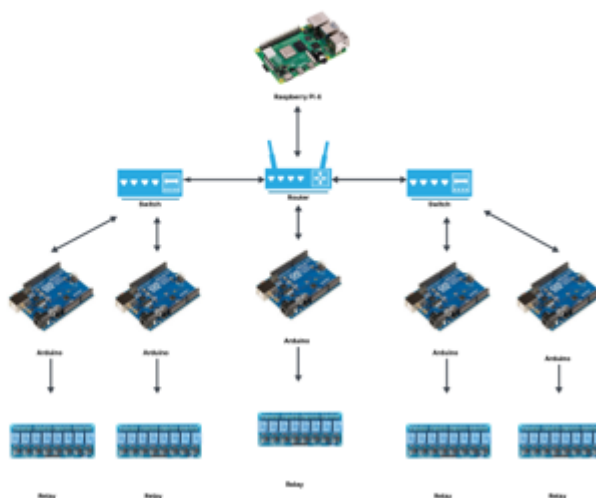
Figure 1
Conceptual Framework



Researchers develop software to process MQTT messages sent via Arduino ethernet shield and Raspberry Pi as a broker connection. The Arduino Uno controls electrical loads in a building, while Raspberry Pi handles web hosting and database management. The LAMP stack (Linux, Apache MySQL, PHP) is used for the software, with Node Red as the bridge between hardware and software. The frontend website uses HTML, JavaScript, and CSS, while the Windows app displays energy consumption and statistics. Data is relayed in Node Red via publish and subscribe protocol and saved on a micro-SD card for future access.

Arduino controllers and relay boards control electrical loads across buildings, while scalability allows for remote management without extensive reconfiguration, and new buildings can be added to the Windows application control panel.

Figure 2
Conceptual Framework for Scalability



Statement of the Problem

This study sought to address unmonitored energy consumption by implementing an Arduino-based monitoring and control system, accessible through software developed by the researchers. To achieve this, the following questions were posed:

1. How can an Arduino and a Raspberry Pi server-based monitoring and control system be effectively integrated with software?
2. Is it feasible to establish a connection between software controls, Arduino, Raspberry-Pi and the associated hardware?
3. How will the system communicate information about the status of the electrical components to the user?

Scope and Delimitations

The system used a Raspberry Pi and an Arduino as core platforms, along with an 8-channel relay board as panelboard controllers, for seamless control of electrical components. A Windows app allows university administration and guards to monitor loads, and a private website for energy consumption monitoring. The prototype focused on developing a Windows app and a private website using a Raspberry Pi to monitor and control electrical loads and display a weekly consumption summary within the TTBD0 building.

METHODOLOGY

Research Design

This study used a software prototyping methodology using the Systems Development Life Cycle (SLDC), specifically the AGILE Methodology, as the research design. AGILE was selected for its formalized yet flexible approach and emphasis on collaboration, making it particularly appropriate for the project's needs. Since this study represents a collaboration between Computer Engineering and Electronics and Communication Engineering teams, the AGILE methodology provided a logical framework that prioritized integrity and functionality. The researchers thus, followed AGILE's distinct phases or stages to develop and assemble the software system systematically.

Figure 3
AGILE Development



Research Locale

The research was conducted at Saint Mary's University in Bayombong, Nueva Vizcaya, with specific implementation on the TTBD0 of the School of Engineering, Architecture, and Information Technology (SEAIT). The prototype was designed to monitor electrical loads and

provide control through a Windows application, while also featuring a website that displays statistical graphs of power, voltage, current, and electrical consumption.

Data Gathering

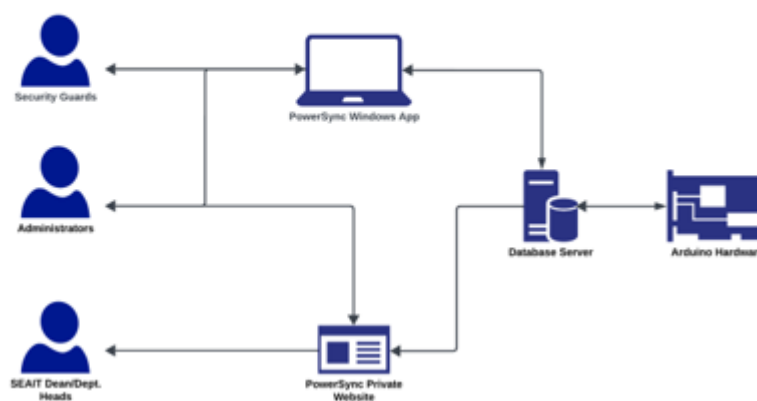
For data collection, the researchers conducted online surveys to evaluate the usability and user-friendliness of both the Windows application and website. The primary data sources were sensor readings from the ECE group's Wattcher device, which used Arduino to monitor energy consumption, and server logs. The Wattcher connects to a router via LAN, transmitting data to a Raspberry Pi that stores information in a MySQL database. System performance was assessed through transmission accuracy logs and response time measurements, while the online surveys provided qualitative feedback on usability and functionality to guide final system adjustments.

Software Description

The energy monitoring system was designed with dual capabilities: controlling electrical loads in the TTBD0 facility and visualizing power consumption data collected from sensors. Key features included graphical data displays and historical analysis capabilities that allow users to review past energy consumption patterns. The system consisted of a web-based interface for monitoring and a Windows application for control, both incorporating user management at different access levels. The architecture followed a client-server model, with PHP handling server-side logic and database interactions using MySQL within a LAMP web server setup. The front-end dashboard utilized JavaScript, HTML, and CSS, with Charts JS providing web visualization tools while Scott Plot handled visualizations for the desktop application built in C#. The system leverages Node-Red's built-in functions for scalability, while the Raspberry Pi implements encryption to secure data transmission between clients and sensors. This infrastructure was designed to be scalable, allowing for potential expansion to monitor and control multiple buildings.

System Architecture

Figure 4
System Architecture

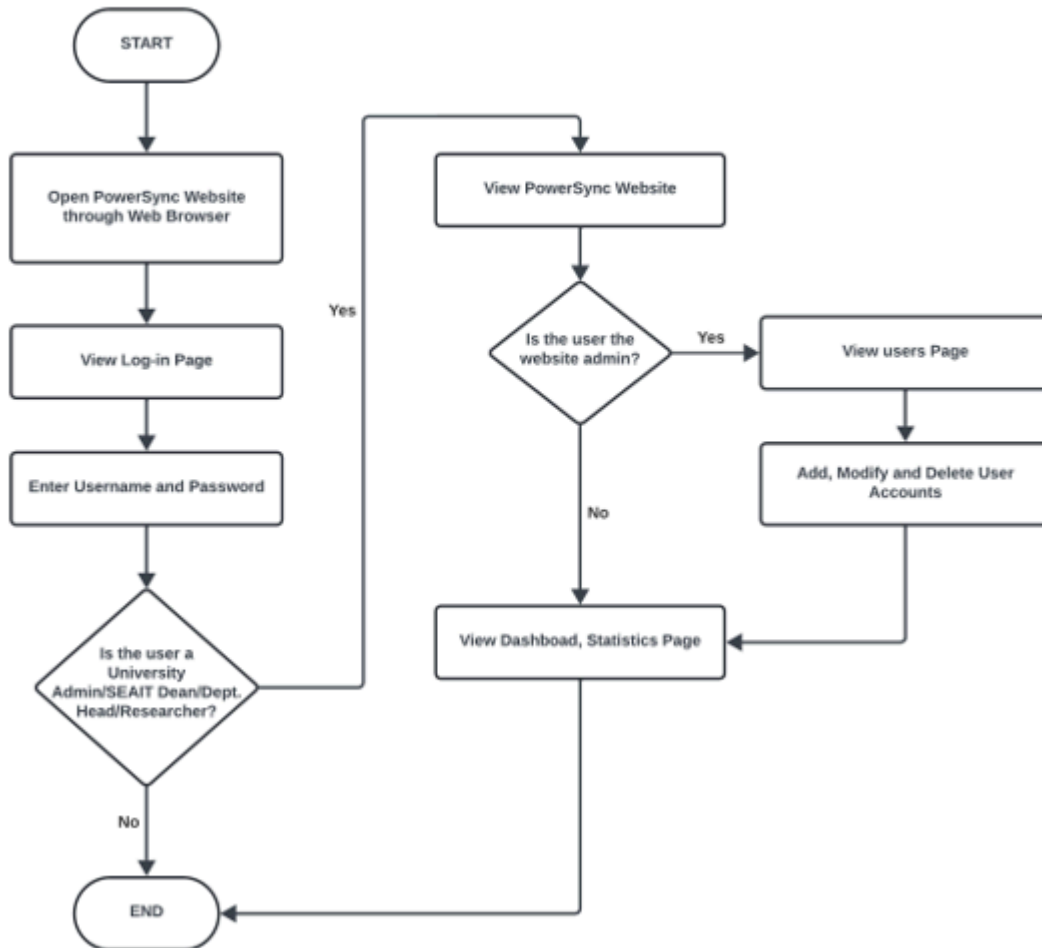


This system architecture outlines the workflow for different user types interacting with the energy monitoring and control system. Administrators and Security Guards can actively control energy consumption at the TTBD0 (Technology Transfer and Business Development Office) through the Windows application. The SEAIT Dean and Department Heads, along with administrators, can access the PowerSync private website to view energy consumption data. The entire system is designed exclusively for use within Saint Mary's University, specifically for the

TTBDO facility. The technical infrastructure is divided between two platforms: the PowerSync Windows application operates over the Local Area Network (LAN), while the private website is hosted on a Raspberry Pi system.

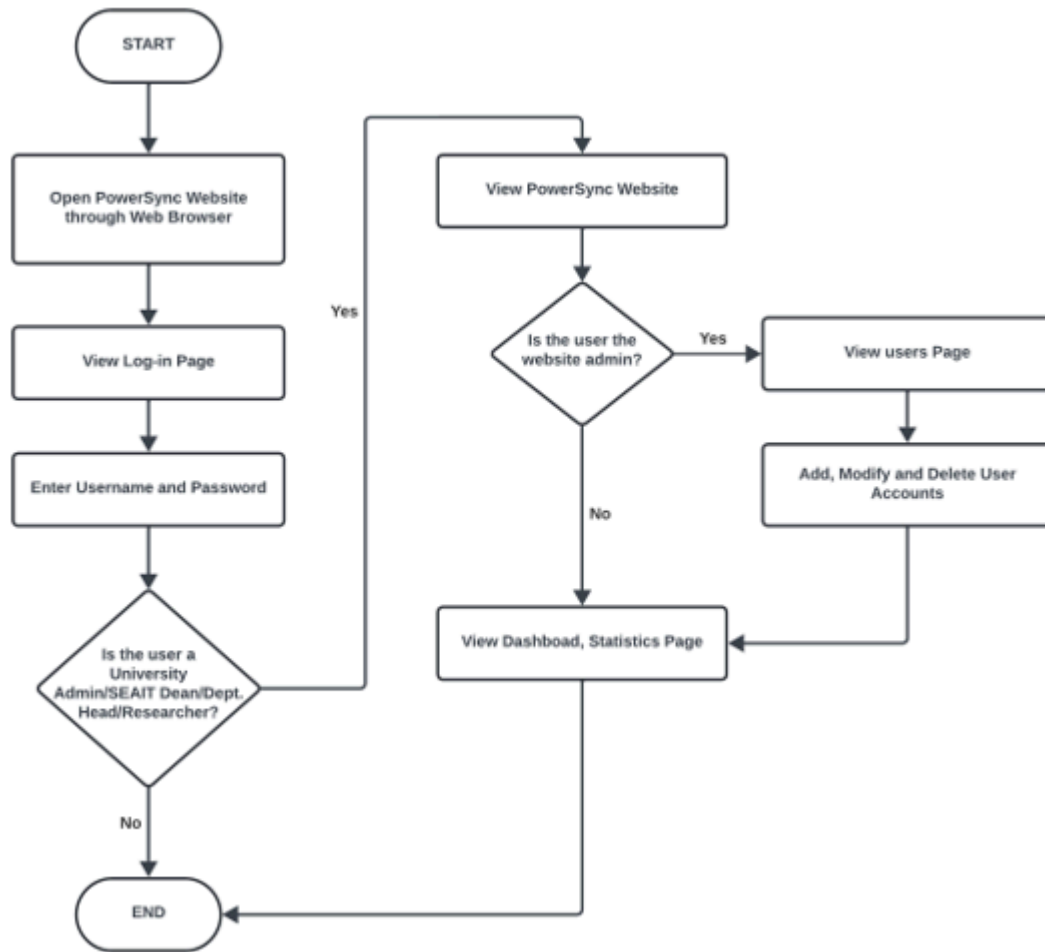
PowerSync Website

Figure 5
Website Flowchart



The PowerSync Website interface is designed for use by university administration, the SEAIT dean, and department heads. System administrators (the researchers) are responsible for creating user profiles and providing login credentials to these stakeholders. To access the website, users must open a web browser, navigate to the PowerSync Website login interface, and enter their unique account name and password. After successful authentication, they are redirected to the main website interface. Both administrators and stakeholders can view the dashboard and statistics pages, which display energy consumption data. However, access to the user management page is restricted exclusively to system administrators to maintain security and ensure only authorized personnel can access the website.

PowerSync Windows App



The PowerSync Windows App is accessible to both university administrators and Security Guards, with administrators using their existing accounts and Security Guards provided with separate login credentials. Once logged in, users are directed to the dashboard, which displays key statistical data, including current active electrical loads, weekly consumption, and an hourly consumption graph. Additionally, users can access the control interface, which displays a visual layout of the building, highlighting rooms with active electrical loads and enabling remote control of those components. Access is restricted to registered users, and unregistered individuals cannot bypass the login screen on either the app or the website.

Treatment of Data

The study used qualitative methods, particularly online surveys, to assess the user-friendliness of PowerSync’s interface. Due to time constraints, Python-generated test data was used in place of actual data from Wattcher’s sensors. This simulated data was sent to a Raspberry Pi server for energy tracking and stored in a MySQL database. User feedback on both the private website and Windows application interfaces guided the qualitative analysis, with survey results informing UI improvements. The website dashboard presented data visually through graphs, offering stakeholders insights into energy consumption trends.

RESULTS AND DISCUSSION

This section contains the results and insights from the development of the Windows App and website, as well as the survey results. It also describes the prototype development of the Windows App and website. The prototype also consists of a miniature scale model of the TTBD0 mounted on a plywood board.

Hardware Development

Creating the miniature model of the TTBD0

1. Connect the dedicated pins from the relay module to the Arduino board.
2. Connect the LED Bulbs to the relay board (Refer to figure 7).
3. Place the hardware components on a tabletop scale model (Refer to Figure 8).

Figure 7

LED connection to relay board

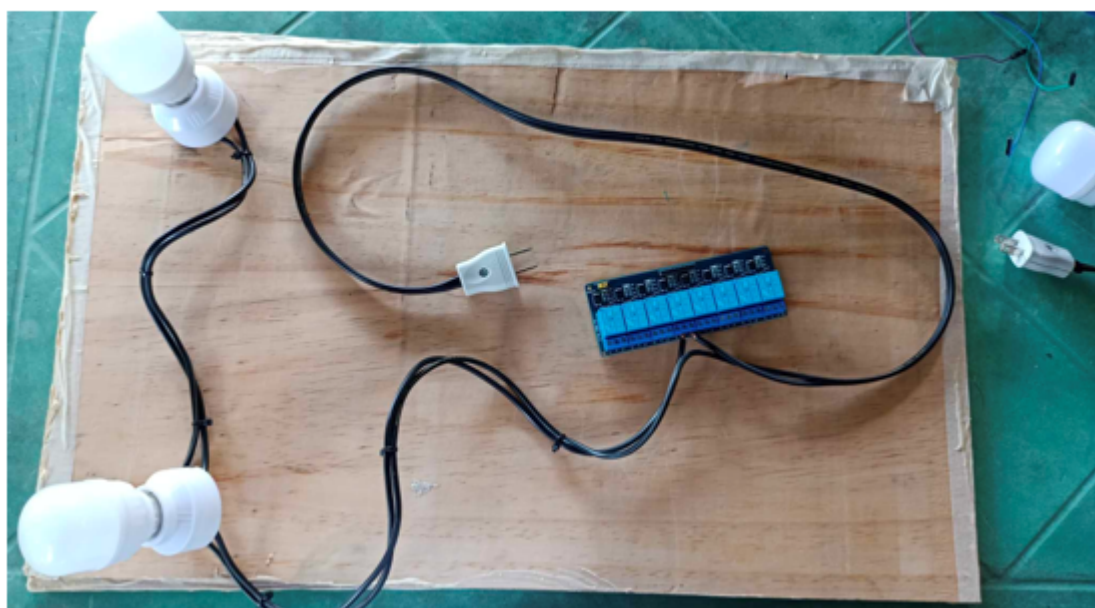
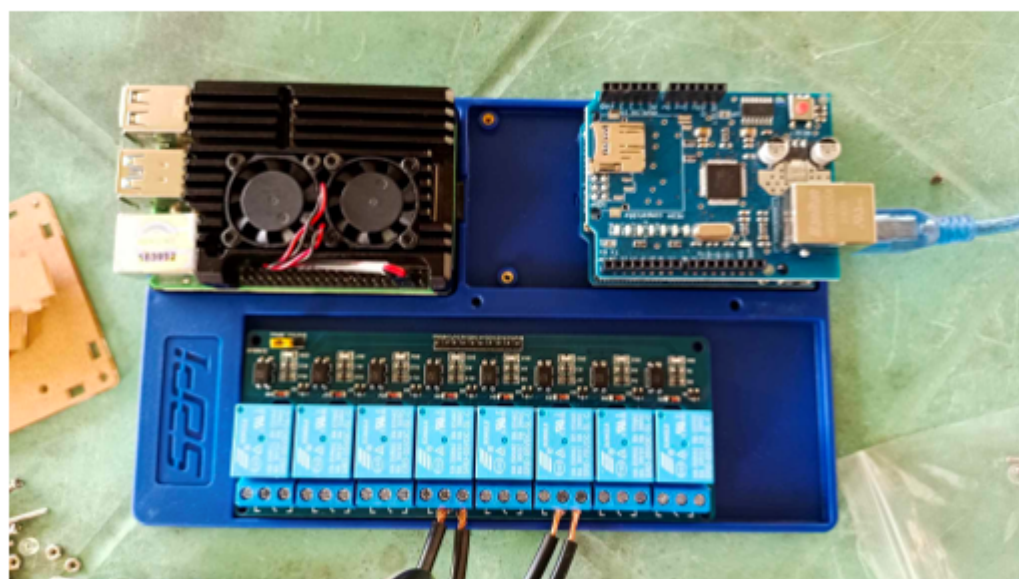


Figure 8

Hardware components to scale model



Establishing the Raspberry Pi 4 as the server

1. Use Node-RED script to establish a connection for the sensor to the database connection (Refer to Figure 9).
2. Determine if the router can detect the Raspberry Pi (Refer to Figure 10).
3. Input the Linux commands through the Raspberry Pi terminal, this is where most of the applications and API's to be hosted on the Raspberry Pi are installed (Refer to Figure 11).
4. Create the database using phpMyAdmin.

Figure 9
Deployed Node-RED Flow Diagram

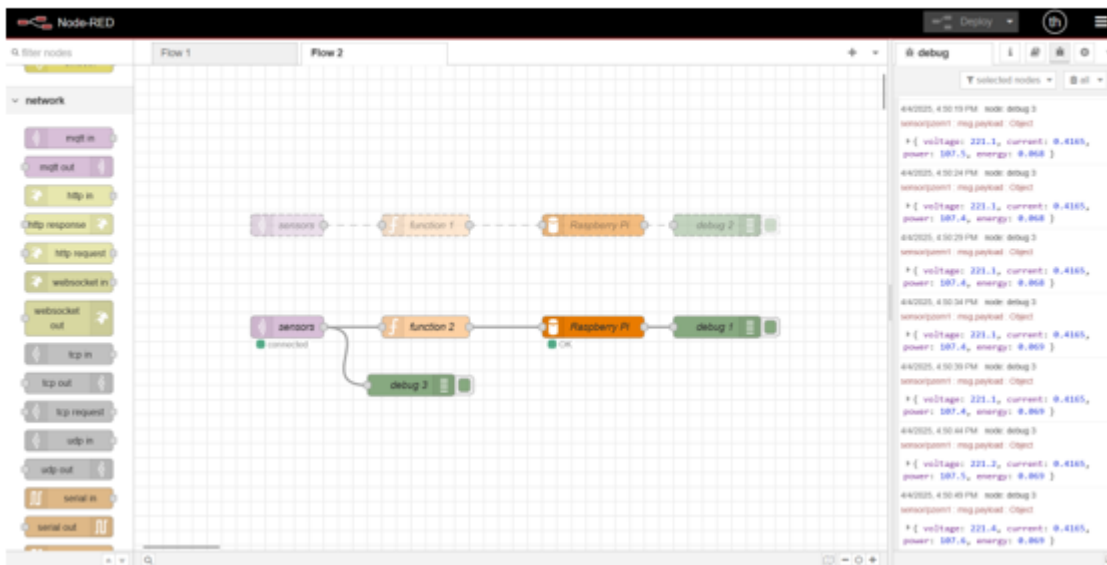


Figure 10
Detecting the Raspberry Pi within the router

DHCP Clients List

This page displays information of all DHCP clients on the network.

ID	Client Name	MAC Address	Assigned IP	Lease Time
1	LAPTOP-DD11BKLU	50:FE:0C:02:F7:77	192.168.0.100	00:00:43
2	raspberrypi	D8:3A:DD:EB:83:50	192.168.0.101	00:00:59

Figure 11
Raspberry Pi Terminal

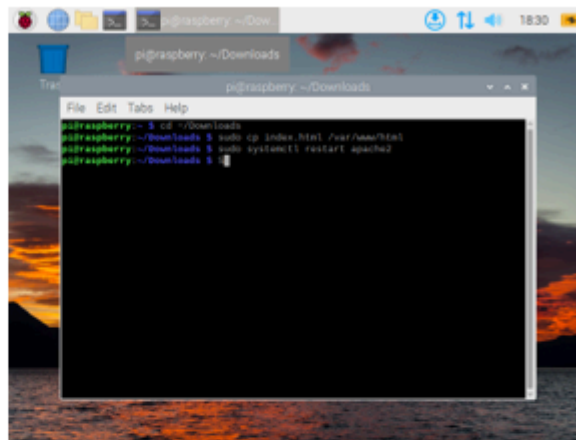


Figure 12
Login Page



Figure 13
Dashboard

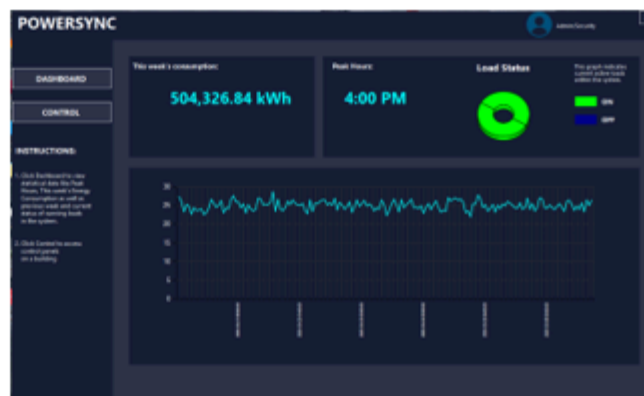


Figure 14
Control Page



Figure 15
Login Page

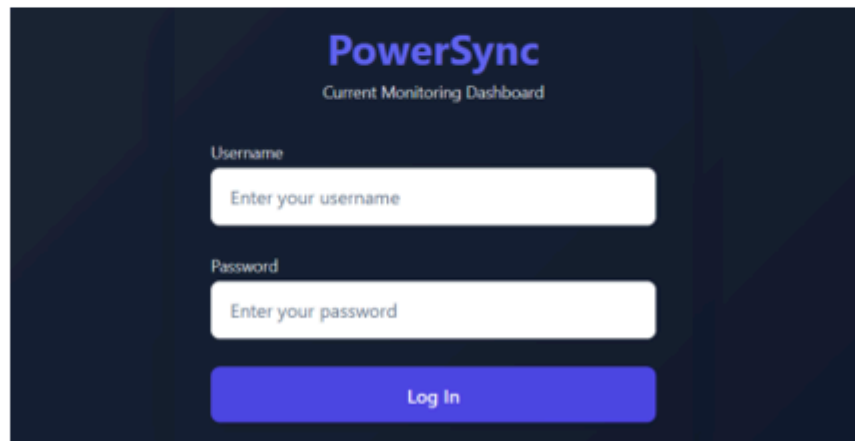


Figure 16
Dashboard for Users (University Administration, SEAIT Dean and Department Heads)



Figure 17
Statistics Page

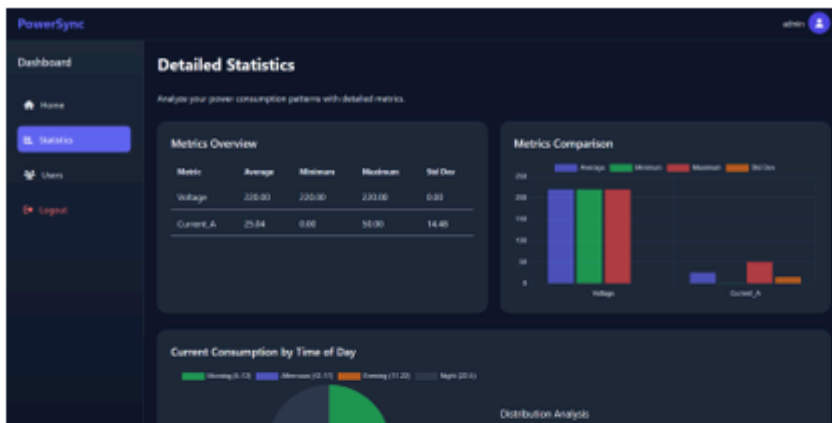
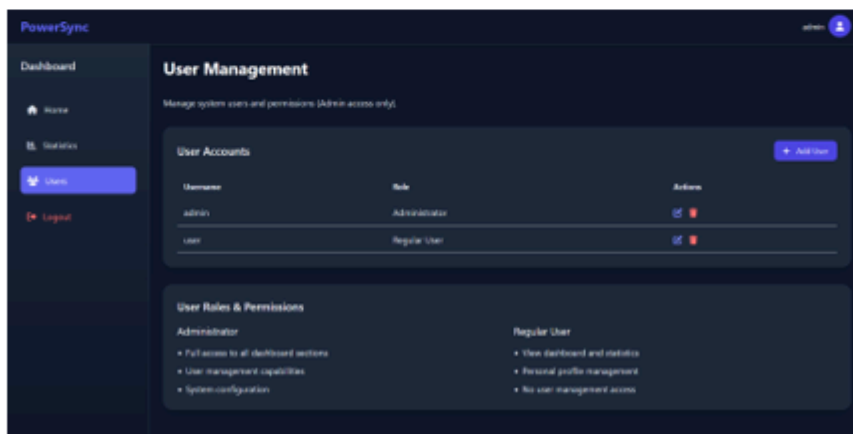


Figure 18
Users Page (Only for Admins)



Analysis of the UI satisfaction survey reveals compelling results: 68% of users rated the system interface as "Excellent," while 22.67% considered it "Commendable." Only 8% of respondents assessed the UI as merely "Competent," with a minimal 1.3% finding it "Challenging."

Analysis of the user experience data reveals exceptional satisfaction levels, with 69% of respondents awarding the system interface the highest rating of "Excellent." An additional 22% classified the UI as "Commendable," while the remaining 9% evaluated it as "Competent."

CONCLUSION AND RECOMMENDATIONS

Conclusion

Both the website and the Windows app exhibit full functionality based on prototype testing. The study's goals are demonstrated by the evidence that follows.

1. The project successfully linked Arduino and Raspberry Pi communication to the hardware components through a LAN connection, utilizing Node-RED and the MQTT protocol.

2. The researchers successfully developed a fully operational and scalable monitoring and control system for electrical loads.
3. The system was able to establish a connection with Wattcher. However, due to time constraints, the researchers still opted to use generated data, as data readings from Wattcher were not yet available.
4. Data from online surveys indicated remarkable user approval, with more than 90% of respondents evaluating the interface as exceeding typical standards, and notably, no participants provided negative feedback.

Recommendations

The following recommendations are for future researchers and studies aiming to enhance the current prototype. These suggestions are also directed towards the beneficiaries of this study.

1. Enhance UI design and functionality of the Windows App and local website.
2. Make more use of the Raspberry Pi instead of only utilizing it as a server.
3. Use ESP32 or ESP8266 rather than Arduino UNO and Wiznet 5100 since it is more cost effective and does twice the same job with just one component rather than two.
4. Improve scalability of the project by giving access to monitor and control more buildings.
5. Implement a timer indicator that displays the remaining time until electrical loads will be remotely turned off in a room.

REFERENCES

- Akbar, F., Pujani, V., & Nazir, R. (2023). Design of real-time electricity consumption monitoring system using campus web. *TEM Journal*, 12(3), 840–849.
- Cirrus Link Solutions. (n.d.). Energy company maximizing MQTT for control and efficient image data transfer. *Cirrus Link*. Retrieved March 30, 2025.
- Degha, H. E., Laallam, F. Z., Seggar, M., & Chaima, A. O. (2019). *Implementation of an energy monitoring system based on Arduino. Artificial Intelligence and Its Applications*.
- Department of the Environment and Energy (DEE). (2013, September 1). *HVAC factsheet—Energy breakdown*.
- Ferencz, K., & Domokos, J. (2020). Using Node-RED platform in an industrial environment. In XXXV. *Jubileumi Kandó Konferencia. Sapientia Hungarian University of Transilvania, Romania*.
- International Organization for Standardization (ISO). (2023, May 30). ISO 50001 — *Energy management*. ISO. <https://www.iso.org/iso-50001-energy-management.html>
- News, T. C., & GMA Integrated. (2024, September 11). *Meralco hikes electricity rate anew in September 2024*. GMA News Online. <https://www.gmanetwork.com/news/money/companies/913508/meralco-hikes-electricity-rate-anew-in-september-2024/story/>
- Ravikiran, A. S. (2024, July 23). *What is Raspberry Pi?* <https://www.raspberrypi.com>
- Steve Cope. (2021, February 12). MQTT — *The standard for IoT messaging*. <http://www.steves-internet-guide.com/>
- Ukpanah, I. (2024, July 10). *Energy waste: A deep dive into environmental consequences*. GreenMatch.co.uk. <https://www.greenmatch.co.uk/blog/energy-waste>